RM0434
参考手册

基于 Arm® Cortex®-M4 无线多协议的 32 位 MCU，配有 FPU、Bluetooth® 低功耗和 802.15.4 无线解决方案

简介

本参考手册面向应用开发人员，提供有关使用 STM32WB55xx 微控制器存储器与外设的完整信息。

STM32WB55xx 无线多协议超低功耗器件嵌入了功能强大的超低功耗无线电功能，符合蓝牙® 低功耗 SIG 规范 v5.0 和 IEEE 802.15.4-2011 的要求，其中包含专用 Arm® Cortex® -M0+ 内核，可执行所有实时底层操作。

STM32WB55xx 系一个微控制器系列，各产品具有不同的存储器大小、封装和外设。

相关文档

意法半导体网站 *www.st.com* 提供以下文档：

- STM32WB55xx 数据手册

有关 Arm® Cortex®-M4 和 Cortex®-M0+ 内核的信息，请分别参见 www.arm.com 网站上相应的技术参考手册。

有关 802.15.4 的信息，请参见 IEEE 网站 (www.ieee.org)。

有关蓝牙® 的信息，请参见 www.bluetooth.com。

目录

表格索引

图片索引

1 文档约定

1.1 概述

STM32WB55xx 器件内嵌 Arm®(a) CPU1 Cortex®-M4 内核。

1.2 寄存器相关缩写词列表

寄存器说明中使用以下缩写词(b)：

读/写 (rw)	软件可以读写该位。
只读 (r)	软件只能读取该位。
只写 (w)	软件只能写入该位。读取该位时将返回复位值。
读取/写入 0 清零 (rc_w0)	软件可以通过读取该位或通过写入 0 将该位清零。写入 1 对该位的值无影响。
读取/写入 1 清零 (rc_w1)	软件可以通过读取该位或通过写入 1 将该位清零。写入 0 对该位的值无影响。
读取/写入清零 (rc_w)	软件可以通过读取该位或通过写入寄存器将该位清零。写入该位的值并不重要。
读取/读取清零 (rc_r)	软件可以读取该位。读取该位时，将自动清零。写入该位对其值无影响。
读取/读取置 1 (rs_r)	软件可以读取该位。读取该位时，将自动置 1。写入该位对其值无影响。
读取/置 1 (rs)	软件可以读取该位，也可将其置 1。写入 0 对该位的值无影响。
读/仅可写入一次 (rwo)	软件仅可写入一次该位，但可随时读取该位。只能通过复位将该位返回到复位值。
切换 (t)	软件可以通过写入 1 来切换该位。写入 0 无影响。
只读，写触发 (rt_w1)	软件可以读取该位。写入 1 时，将触发事件，但不会影响该位的值。
保留 (Res.)	保留位，必须保持复位值。

a. Arm 是 Arm Limited（或其子公司）在美国和/或其他地区的注册商标。

b. 这是一个涵盖适用于 STM 微控制器的所有缩写词的详尽列表，其中一些缩写词在当前文档中可能并未使用。

1.3 词汇表

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

- **字**：32 位数据。
- **半字**：16 位数据。
- **字节**：8 位数据。
- **选项字节**：存储于 Flash 中的产品配置位。
- **AHB**：高级高性能总线。

1.4 外设可用性

有关各型号产品所支持的外设类型及数量，请参见相关器件数据手册。

2 系统和存储器概述

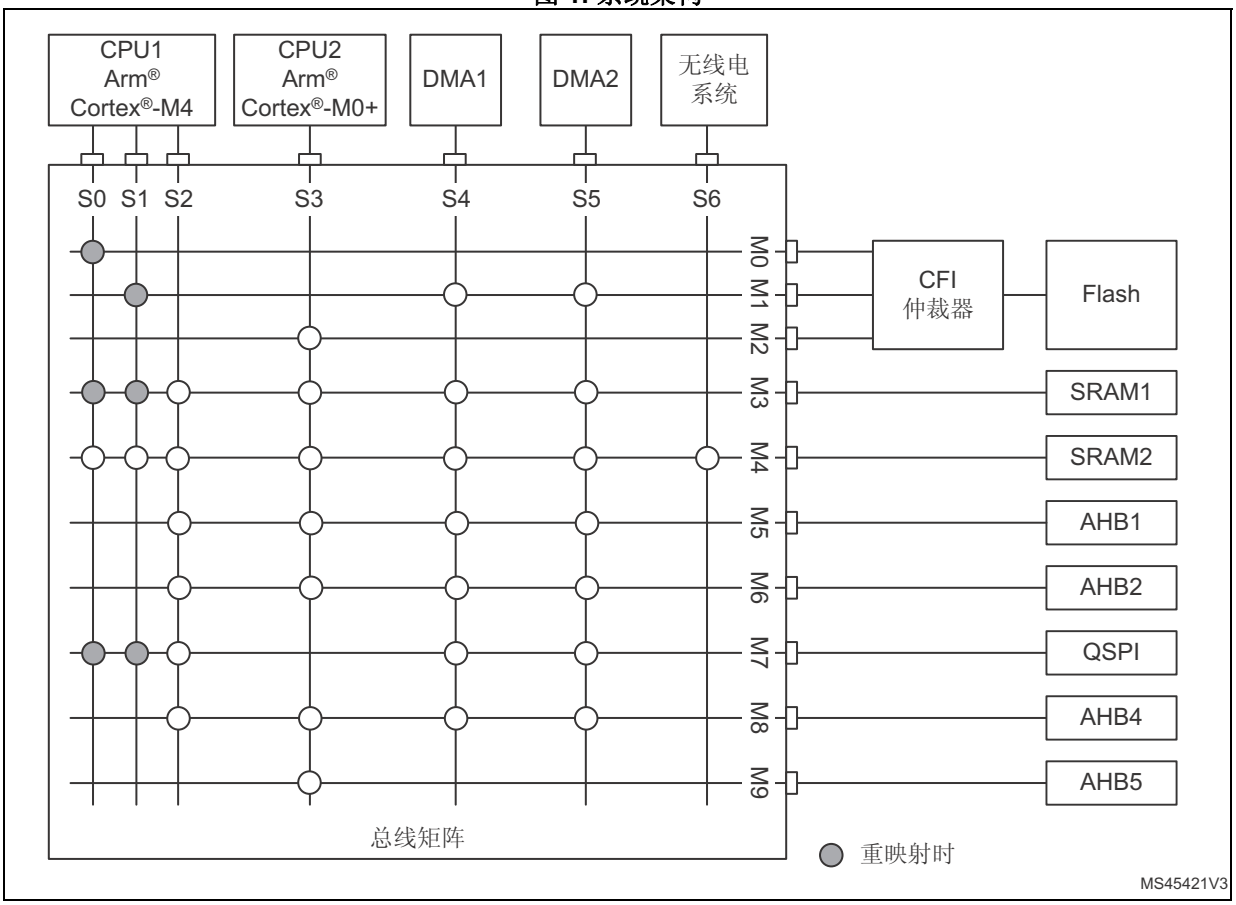
2.1 系统架构

主系统由 32 位多层 AHB 总线矩阵构成，可实现以下部分的互连：

- 七条主控总线：
  - CPU1（CPU1 Cortex®-M4 带 FPU）内核 I 总线
  - CPU1（CPU1 Cortex®-M4 带 FPU）内核 D 总线
  - CPU1（CPU1 Cortex®-M4 带 FPU）内核 S 总线
  - CPU2 (Cortex®-M0+) 内核 S 总线
  - DMA1
  - DMA2
  - 无线电系统
- 十条被控总线：
  - CPU1 (CPU1 Cortex®-M4) ICode 总线上的内部 Flash
  - CPU1 (CPU1 Cortex®-M4) DCode 总线上的内部 Flash
  - CPU2 (Cortex®-M0+) S 总线上的内部 Flash
  - 内部 SRAM1 (192 KB)
  - 内部 SRAM2a (32 KB) + SRAM2b (32 KB)
  - AHB1 外设（包括 AHB-APB 总线桥和 APB 外设（连接到 APB1 和 APB2））
  - AHB2 外设
  - Quad SPI 存储器接口
  - AHB4 共享外设
  - AHB5 外设（包括 AHB-APB 总线桥和无线电外设（连接到 APB3））

借助总线矩阵，可以实现主控总线到被控总线的访问，这样即使在多个高速外设同时运行期间，系统也可以实现并发访问和高效运行。此架构如*图 1* 所示：

图 1. 系统架构



2.1.1 S0：CPU1 (CPU1 Cortex®-M4) I 总线

此总线用于将 CPU1 内核的指令总线连接到总线矩阵。内核通过此总线获取指令。此总线的目标连接位置是内部 Flash、SRAM1、SRAM2a（备份）、SRAM2b（非备份）和外部存储器（通过 QUADSPI）。

2.1.2 S1：CPU1 (CPU1 Cortex®-M4) D 总线

此总线用于将 CPU1 内核的数据总线连接到总线矩阵。内核通过此总线进行立即数加载和调试访问。此总线的目标连接位置是内部 Flash、SRAM1、SRAM2a（备份）、SRAM2b（非备份）和外部存储器（通过 QUADSPI）。

2.1.3 S2：CPU1 (CPU1 Cortex®-M4) S 总线

此总线用于将 CPU1 内核的系统总线连接到总线矩阵。内核使用此总线访问位于外设或 SRAM 区域中的数据。此总线的目标连接位置是 SRAM1、SRAM2a（备份）、SRAM2b（非备份）、AHB1 外设（包括 APB1 和 APB2 外设）、AHB2 外设、AHB4 外设和外部存储器（通过 QUADSPI）。

2.1.4 S3：CPU2 (Cortex®-M0+) S 总线

此总线用于将 CPU2 内核的系统总线连接到总线矩阵。内核使用此总线取指令、进行立即数加载和调试访问、以及访问位于外设或 SRAM 区域中的数据。此总线的目标连接位置是内部 Flash、SRAM1、SRAM2a（备份）、SRAM2b（非备份）、AHB1 外设（包括 APB1 和 APB2 外设）、AHB2 外设、AHB4 外设和 AHB5 外设（包括 APB3 外设）。

2.1.5 S4、S5：DMA 总线

此总线用于将 DMA 的 AHB 主接口连接到总线矩阵。此总线的目标连接位置是 SRAM1、SRAM2a（备份）、SRAM2b（非备份）、AHB1 外设（包括 APB1 和 APB2 外设）、AHB2 外设、AHB4 外设和外部存储器（通过 QUADSPI）。

2.1.6 S6：无线电系统总线

此总线用于将无线电系统的 AHB 主接口连接到总线矩阵。此总线的目标连接位置是 SRAM2a（备份）、SRAM2b（非备份）。

2.1.7 总线矩阵

总线矩阵用于主控总线之间的访问仲裁管理。仲裁采用循环调度算法。总线矩阵由七条主控总线（CPU1：系统总线、DCode 总线、ICode 总线，CPU2：系统总线、DMA1 总线和 DMA2 总线，以及无线电系统总线）和十条被控总线（3 x FLASH、SRAM1、SRAM2a（备份）和 SRAM2b（非备份）、AHB1（包括 APB1 和 APB2）、AHB2、QUADSPI、AHB4 和 AHB5）构成。

AHB/APB 总线桥

借助 AHB-APB1 和 AHB-ABP2 两个总线桥，可在 AHB 总线与两个 APB 总线之间实现完全同步的连接，从而灵活选择外设频率。

借助 AHB-APB3 总线桥，可在 AHB 和 APB 总线之间实现同步连接，从而灵活选择 AHB 和外设之间的频率。

有关连接到此总线桥的外设的地址映射，请参见*第 2.2 节：存储器构成*。

每次芯片复位后，所有外设时钟都被关闭（SRAM1/2 和 Flash 接口除外）。使用外设前，必须在 RCC_AHBxENR 和 RCC_APBxENR 寄存器中使能其时钟。

注： *对 APB 寄存器执行 16 位或 8 位访问时，该访问将转换为 32 位访问：总线桥将 16 位或 8 位数据复制后提供给 32 位向量。*

2.2 存储器构成

2.2.1 简介

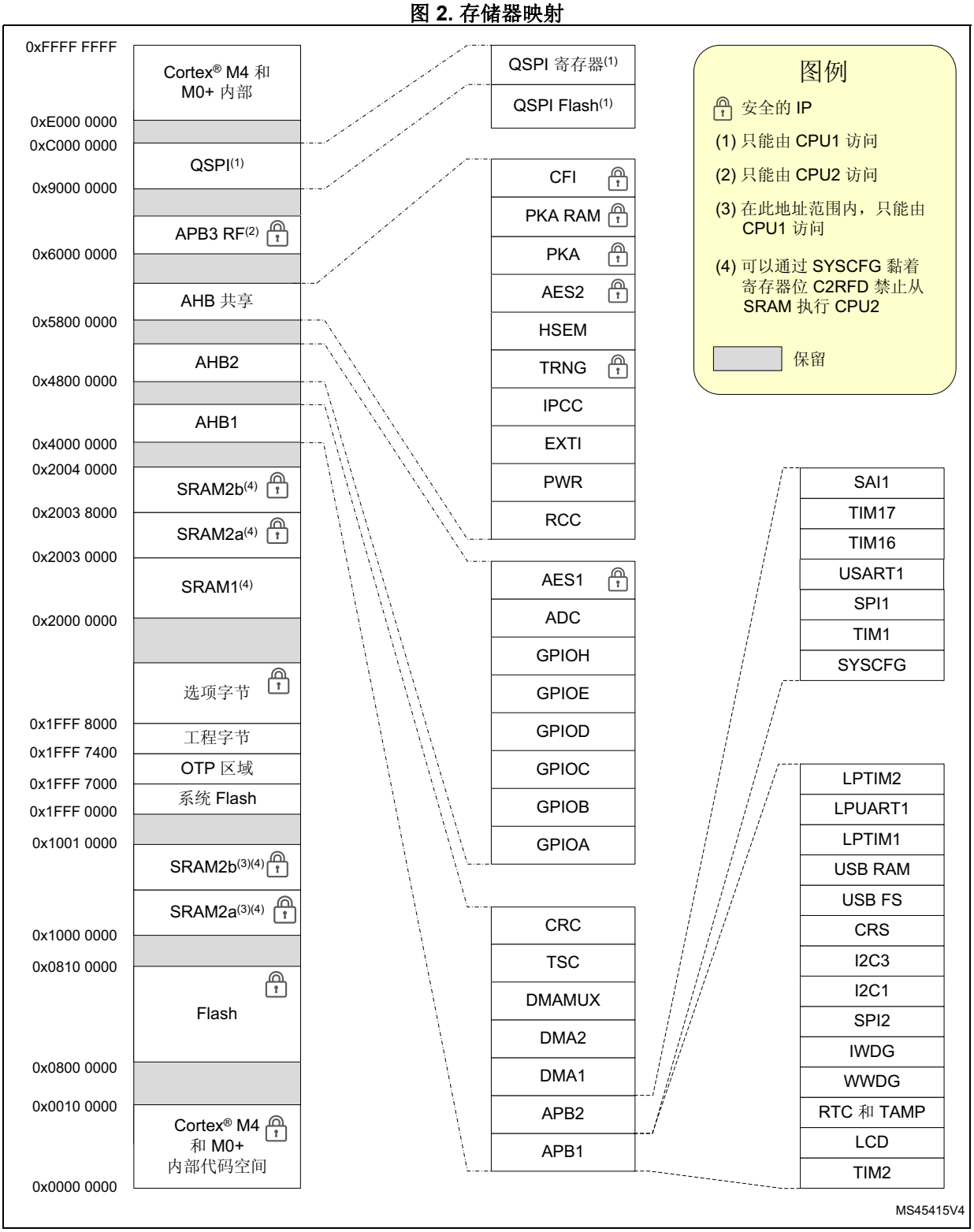
程序存储器、数据存储器、寄存器和 I/O 端口排列在同一个线性（即地址连续）的 4 GB 地址空间内。

各字节按小端格式在存储器中编码。字中编号最低的字节被视为该字的最低有效字节，而编号最高的字节被视为最高有效字节。

可寻址的存储空间分为 8 个主要块，每个块为 512 MB。

2.2.2 存储器映射和寄存器边界地址

图 2. 存储器映射



1. QUADSPI 外设若没有正确配置并使能，禁止访问 QUADSPI Flash 的存储区域。

未分配给片上存储器和外设的所有存储映射区域均视为“保留区”。有关可用存储器和寄存器区域的详细映射，请参见下表。

下表给出了器件中可用外设的边界地址。

表 1. STM32WB55xx 存储器映射和外设寄存器边界地址

总线	边界地址	大小（字节）	外设	外设寄存器映射
AHB3	0xA000 1000 - 0xA000 13FF	1 KB	QUADSPI	*第 401 页的第 15.5.14 节：QUADSPI 寄存器映射*
	0xA000 0000 - 0xA000 0FFF	4 KB	保留	-
	0x9000 0000 - 0x9FFF FFFF	256 KB	QUADSPI Flash	-
-	0x6000 2000 - 0x8FFF FFFF	-	保留	-
APB3	0x6000 1000 - 0x6000 1FFF	4 KB	802.15.4 CTRL	-
	0x6000 0800 - 0x6000 0FFF	2 KB	保留	-
	0x6000 0400 - 0x6000 07FF	1 KB	无线电 CTRL	-
	0x6000 0000 - 0x6000 03FF	1 KB	BLE CTRL	-
AHB4	0x5800 4400 - 0x5FFF FFFF	-	保留	-
	0x5800 4000 - 0x5800 43FF	1 KB	FLASH	*第 120 页的第 3.10.20 节：FLASH 寄存器映射*
	0x5800 3400 - 0x5800 3FFF	-	保留	-
	0x5800 2400 - 0x5800 33FF	5 KB	PKA RAM	*第 644 页的第 23.7.5 节：PKA 寄存器映射和复位值*
	0x5800 2000 - 0x5800 23FF		PKA	
	0x5800 1C00 - 0x5800 1FFF	1 KB	保留	-
	0x5800 1800 - 0x5800 1BFF	1 KB	AES2	*第 613 页的第 22.7.18 节：AES 寄存器映射*
	0x5800 1400 - 0x5800 17FF	1 KB	HSEM	*第 1243 页的第 38.4.9 节：HSEM 寄存器映射*
	0x5800 1000 - 0x5800 13FF	1 KB	真 RNG	*第 561 页的第 21.7.4 节：RNG 寄存器映射*
	0x5800 0C00 - 0x5800 0FFF	1 KB	IPCC	*第 1230 页的第 37.4.9 节：IPCC 寄存器映射和复位值表*
	0x5800 0800 - 0x5800 0BFF	1 KB	EXTI	*第 375 页的第 14.5.17 节：EXTI 寄存器映射*
	0x5800 0400 - 0x5800 07FF	1 KB	PWR	*第 183 页的第 6.6.24 节：PWR 寄存器映射和复位值表*
	0x5800 0000 - 0x5800 03FF	1 KB	RCC	*第 274 页的第 8.4.49 节：RCC 寄存器映射*

表 1. STM32WB55xx 存储器映射和外设寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
AHB2	0x5006 0400 - 0x57FF FFFF	-	保留	-
	0x5006 0000 - 0x5006 03FF	1 KB	AES1	*第 613 页的第 22.7.18 节：AES 寄存器映射*
	0x5004 0000 - 0x5004 03FF	1 KB	ADC	*第 483 页的第 16.6.3 节：ADC 寄存器映射*
	0x4800 1C00 - 0x4800 1FFF	1 KB	GPIOH	*第 297 页的第 9.4.12 节：GPIO 寄存器映射*
	0x4800 1400 - 0x4800 1BFF	3 KB	保留	-
	0x4800 1000 - 0x4800 13FF	1 KB	GPIOE	*第 297 页的第 9.4.12 节：GPIO 寄存器映射*
	0x4800 0C00 - 0x4800 0FFF	1 KB	GPIOD	
	0x4800 0800 - 0x4800 0BFF	1 KB	GPIOC	
	0x4800 0400 - 0x4800 07FF	1 KB	GPIOB	
	0x4800 0000 - 0x4800 03FF	1 KB	GPIOA	
AHB1	0x4002 4400 - 0x47FF FFFF	-	保留	-
	0x4002 4000 - 0x4002 43FF	1 KB	TSC	*第 549 页的第 20.6.11 节：TSC 寄存器映射*
	0x4002 3400 - 0x4002 3FFF	3 KB	保留	-
	0x4002 3000 - 0x4002 33FF	1 KB	CRC	*第 130 页的第 5.4.6 节：CRC 寄存器映射*
	0x4002 0C00 - 0x4002 2FFF	9 KB	保留	-
	0x4002 0800 - 0x4002 0BFF	1 KB	DMAMUX	*第 349 页的第 12.6.7 节：DMAMUX 寄存器映射*
	0x4002 0400 - 0x4002 07FF	1 KB	DMA2	*第 334 页的第 11.6.7 节：DMA 寄存器映射和复位值*
	0x4002 0000 - 0x4002 03FF	1 KB	DMA1	
-	0x4001 5800 - 0x4001 FFFF	42 KB	保留	-

表 1. STM32WB55xx 存储器映射和外设寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
APB2	0x4001 5400 - 4001 57FF	1 KB	SAI1	*第 1216 页的第 36.5.19 节：SAI 寄存器映射*
	0x4001 4C00 - 4001 53FF	2 KB	保留	-
	0x4001 4800 - 4001 4BFF	1 KB	TIM17	*第 856 页的第 26.4.23 节：TIM16/TIM17 寄存器映射*
	0x4001 4400 - 4001 47FF	1 KB	TIM16	*第 856 页的第 26.4.23 节：TIM16/TIM17 寄存器映射*
	0x4001 3C00 - 4001 43FF	2 KB	保留	-
	0x4001 3800 - 4001 3BFF	1 KB	USART1	*第 1073 页的第 33.7.15 节：USART 寄存器映射*
	0x4001 3400 - 4001 37FF	1 KB	保留	-
	0x4001 3000 - 4001 33FF	1 KB	SPI1	*第 1156 页的第 35.6.8 节：SPI 寄存器映射*
	0x4001 2C00 - 4001 2FFF	1 KB	TIM1	*第 739 页的第 24.4.30 节：TIM1 寄存器映射*
	0x4001 0400 - 4001 2BFF	10 KB	保留	-
	0x4001 0200 - 4001 03FF	1 KB	COMP	*第 501 页的第 18.6.3 节：COMP 寄存器映射*
	0x4001 0100 - 4001 01FF		SYSCFG	*第 314 页的第 10.2.17 节：SYSCFG 寄存器映射*
	0x4001 0030 - 4001 00FF		VREFBUF	*第 489 页的第 17.3.3 节：VREFBUF 寄存器映射*
	0x4001 0000 - 4001 002F		SYSCFG	*第 314 页的第 10.2.17 节：SYSCFG 寄存器映射*

表 1. STM32WB55xx 存储器映射和外设寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
APB1	0x4000 9800 - 4000 FFFF	26 KB	保留	-
	0x4000 9400 - 04000 97FF	1 KB	LPTIM2	*第 879 页的第 27.7.11 节：LPTIM 寄存器映射*
	0x4000 8400 - 04000 93FF	4 KB	保留	-
	0x4000 8000 - 04000 83FF	1 KB	LPUART1	*第 1121 页的第 34.5.13 节：LPUART 寄存器映射*
	0x4000 7C00 - 04000 7FFF	1 KB	LPTIM1	*第 879 页的第 27.7.11 节：LPTIM 寄存器映射*
	0x4000 7000 - 04000 7BFF	3 KB	保留	-
	0x4000 6C00 - 04000 6FFF	1KB	USB SRAM	*第 1272 页的第 39.6.3 节：USB 寄存器映射*
	0x4000 6800 - 04000 6BFF	1 KB	USB FS	*第 1272 页的第 39.6.3 节：USB 寄存器映射*
	0x4000 6400 - 04000 67FF	1 KB	保留	-
	0x4000 6000 - 04000 63FF	1 KB	CRS	*第 1284 页的第 40.6.5 节：CRS 寄存器映射*
	0x4000 5C00 - 04000 5FFF	1 KB	I2C3	*第 993 页的第 32.7.12 节：I2C 寄存器映射*
	0x4000 5800 - 04000 5BFF	1 KB	保留	-
	0x4000 5400 - 04000 57FF	1 KB	I2C1	*第 993 页的第 32.7.12 节：I2C 寄存器映射*
	0x4000 3C00 - 04000 53FF	6 KB	保留	-
	0x4000 3800 - 04000 3BFF	1 KB	SPI2	*第 1156 页的第 35.6.8 节：SPI 寄存器映射*
	0x4000 3400 - 0x4000 37FF	1 KB	保留	-
	0x4000 3000 - 0x4000 33FF	1 KB	IWDG	*第 924 页的第 30.4.6 节：IWDG 寄存器映射*
	0x4000 2C00 - 0x4000 2FFF	1 KB	WWDG	*第 930 页的第 31.4.4 节：WWDG 寄存器映射*
	0x4000 2800 - 0x4000 2BFF	1 KB	RTC 和 TAMP	*第 915 页的第 29.6.21 节：RTC 寄存器映射*
	0x4000 2400 - 0x4000 27FF	1 KB	LCD	*第 531 页的第 19.6.6 节：LCD 寄存器映射*
	0x4000 0400 - 0x4000 23FF	8 KB	保留	-
	0x4000 0000 - 0x4000 03FF	1 KB	TIM2	*第 807 页的第 25.4.25 节：TIMx 寄存器映射*
AHB4	0x2003 8000 - 0x2003 FFFF	32 KB	SRAM2b	-
	0x2003 0000 - 0x2003 7FFF	32 KB	SRAM2a	-
AHB1	0x2000 0000 - 0x2002 FFFF	196 KB	SRAM1	-

表 1. STM32WB55xx 存储器映射和外设寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
AHB4	0x1FFF 8000 - 0x1FFF 807F	128 B	Flash 选项	*第 120 页的第 3.10.20 节：FLASH 寄存器映射*
	0x1FFF 7000 - 0x1FFF 73FF	1 KB	Flash OTP	-
	0x1FFF 0000 - 0x1FFF 6FFF	28 KB	Flash 启动程序	-
	0x1000 0000 - 0x1000 FFFF	64 KB	SRAM2a/b CPU1 镜像	-
	0x0800 0000 - 0x080F FFFF	1 MB	用户 Flash	-
(1)	0x0000 0000 - 0x000F FFFF	1 MB	CPUn 启动区域	-

1. 总线取决于所选 CPUn 启动区域。

2.2.3 位段

CPU1 映射包括两个位段区域。这些区域将存储器别名区域中的每个字映射到存储器位段区域中的相应位。在别名区域写入字时，相当于对位段区域的目标位执行读-修改-写操作。

在 STM32WB55xx 器件中，AHB1、APB1、APB2 外设寄存器以及 SRAM1、SRAM2a 和 SRAM2b 均映射到一个位段区域，这样可实现单个位段的读写操作。这些操作仅适用于 CPU1 访问，对于其他总线主接口（如 DMA）无效。

外设位段别名位于地址 0x4200 0000 到 0x42FF FFFF 范围内。

SRAM 位段别名位于地址 0x2200 0000 到 0x227F FFFF 范围内。

可通过一个映射公式说明别名区域中的每个字与位段区域中各个位之间的对应关系。映射公式为：

bit_word_addr = bit_band_base + (byte_offset * 32) + (bit_number * 4)，其中：

- bit_word_addr 代表别名区域中将映射到目标位的字的地址
- bit_band_base 代表别名区域的起始地址
- byte_offset 代表目标位所在位段区域中的字节编号
- bit_number 代表目标位的位位置 (0-7)

示例

下例说明如何将 SRAM1 地址 0x2000 0300 处字节的位 [2] 映射到别名区域。

0x2200 6008 = 0x2200 0000 + 0x0300 * 32 + 2 * 4

对地址 0x2200 6008 执行写操作相当于在 SRAM1 地址 0x2000 0300 处字节的位 [2] 执行读-修改-写操作。

对地址 0x2200 6008 执行读操作将返回 SRAM1 地址 0x2000 0300 处字节的位 [2] 的值 0x01 或 0x00。

有关位段的详细信息，请参见 Cortex®-M4 编程手册。

2.3 启动配置

在 STM32WB55xx 中，可通过 BOOT0 引脚和用户选项中的 nBOOT1 位来选择三种不同的 CPU1 启动模式，如*表 2* 所示。

表 2. 启动模式

启动模式选择		启动模式	别名
BOOT1(1)	**BOOT0**		
x	0	主 Flash	选择主 Flash 作为启动空间
0	1	系统存储器	选择系统 Flash 作为启动空间
1	1	嵌入式 SRAM	选择 SRAM 存储器作为启动空间

1. BOOT1 的值为 nBOOT1 用户选项的相反值

复位后，锁存 BOOT0 和 BOOT1 的值。用户可以通过提供适当的值来选择需要的启动模式。

退出待机模式时，还会对 BOOT0 和 BOOT1 进行重新采样。因此，这些引脚必须保持在所需的启动模式。启动延迟结束后，CPU1 将从地址 0x0000 0000 获取栈顶值，然后从始于 0x0000 0004 的启动存储器开始执行代码。

根据所选的启动模式，主 Flash、系统 Flash 或 SRAM1 存储器可如下访问：

- 从主 Flash 启动：主 Flash 在 CPU1 启动存储器空间 (0x0000 0000) 中有别名，但仍可从其物理地址 0x0800 0000 访问。换句话说，Flash 内容可从地址 0x0000 0000 或 0x0800 0000 开始访问。
- 从系统 Flash 启动：系统 Flash 在 CPU1 启动存储器空间 (0x0000 0000) 中有别名，但仍可从其物理地址 0x1FFF 0000 访问。
- 从 SRAM 存储器启动：SRAM 存储器在 CPU1 启动存储器空间 (0x0000 0000) 中有别名，但仍可从其物理地址 0x2000 0000 访问。

CPU1 物理重映射

在 CPU1 启动后，应用软件可修改地址 0x0000 0000 处的存储器映射。这样的修改通过在 SYSCFG 控制器中编程 *SYSCFG 存储器重映射寄存器 (SYSCFG_MEMRMP)* 来实现。

可重映射以下存储器：

- 主 Flash
- 系统 Flash
- SRAM
- Quad SPI 存储器

嵌入式启动程序

嵌入式启动程序位于系统 Flash 中，由 ST 在生产阶段编程。它用于通过以下器件接口编程 Flash：

- 引脚 PA9 和 PA10 上的 USART1
- 引脚 PB6 和 PB7 上的 I2C1
- 引脚 PC0 和 PC1 上的 I2C3
- 引脚 PA4、PA5、PA6 和 PA7 上的 SPI1
- 引脚 PB12、PB13、PB14 和 PB15 上的 SPI2
- 引脚 PA11 和 PA12 上的 USB FS

2.4 CPU2 启动

器件复位后，CPU2 仅在 CPU1 将 *PWR 控制寄存器 4 (PWR_CR4)* 的 C2BOOT 位置 1 后启动。C2BOOT 值在待机模式下将得以保持，当退出待机模式时，CPU2 将使用该值启动。

CPU2 将从其启动复位向量（如 Flash 用户选项 C2OPT 和 SBRV 定义）启动。

CPU2 可从用户 Flash 或 SRAM1/SRAM2a/SRAM2b 中的任意位置启动。

CPU2 安全启动

复位后，如果用户选项无效且 BOOT0 和 BOOT1 选择 CPU1 从主 Flash 启动，则 CPU2 将从主 Flash 中的安全启动向量启动，地址为 0x080F F000。

安全启动可用于从映像副本恢复最近一次已知的用户选项。

2.5 CPU2 SRAM 取指令禁止

可通过 SYSCFG 寄存器中的 C2RFD 位来禁止 CPU2 从 SRAM 执行指令。禁止 CPU2 从 SRAM 执行指令可提高 CPU2 软件的稳健性。

3 嵌入式 Flash (FLASH)

3.1 简介

Flash 接口可管理 CPU1 (CPU1 Cortex®-M4) 通过 AHB ICode 和 DCode 对 Flash 进行的访问，以及 CPU2 (Cortex®-M0+) 通过 AHB 对 Flash 进行的访问。该接口可为这两个 CPU 的访问操作执行仲裁、可针对 Flash 执行擦除和编程操作，并且可实施读写保护和安全机制。

Flash 接口通过指令预取和缓存机制加速代码执行。

3.2 Flash 主要特性

- 高达 1 MB 的 Flash，采用单存储区架构
- 存储器构成：1 个存储区
  - 主存储器：高达 1 MB
  - 页大小：4 KB
- 数据读取宽度为 72 位（64 位 + 8 个 ECC 位）
- 数据写入宽度为 72 位（64 位 + 8 个 ECC 位）
- 页擦除 (4 KB) 和批量擦除

Flash 接口特性：

- Flash 读操作
- Flash 编程/擦除操作
- 由选项字节激活的读保护 (RDP)
- 由选项字节选择的两个写保护区域 (WRP)
- 由选项字节选择的两个专有代码读保护区域 (PCROP)
- CPU2 安全区域
- Flash 为空检查
- 编程和擦除暂停功能
- CPU1 ICODE 和 CPU2 S 总线上的预取操作
- CPU1 指令缓存：ICode 上的 32 个缓存行，4 x 64 位宽 (1 KB RAM)
- CPU1 数据缓存：DCode 上的 8 个缓存行，4 x 64 位宽（256 字节 RAM）
- CPU2 指令缓存：S 总线上的 4 个缓存行，1 x 64 位宽（32 字节 RAM）
- 误码校正 (ECC)：8 位/64 位
- 选项字节加载器

3.3 Flash 功能说明

3.3.1 Flash 构成

Flash 为 72 位宽的存储单元（64 位 + 8 个 ECC 位），可用于存储代码和数据常量。

Flash 构成如下：

- 一个主存储器块，其中包含 256 页，每页 8 行（共 4 KB），每行 512 字节。
- 一个信息块，其中包含：
  - 系统存储器，CPU1 在系统存储器启动模式下从该存储器启动。此区域为保留区域，其中包含启动程序，用以通过以下接口之一对 Flash 进行重新编程：USART1、USB、I2C1、I2C3、SPI1、SPI2。它由意法半导体在器件制造期间编程并加以保护，以防止误写/误擦除操作。更多详细信息，请参见 *www.st.com* 上提供的 AN2606。
  - 1 KB（128 个双字）OTP（一次性可编程）字节，用于存储用户数据。OTP 数据不能擦除，只能执行一次写操作。如果仅一位为 0，则即使值为 0x0000 0000 0000 0000，整个双字（64 位）也无法再写入。
    当 RDP 级别为 1 且启动源不是 Flash 用户区域时，无法读取 OTP 区域。
  - 用于用户配置的选项字节。

存储器由一个主区域和一个信息块构成，如*表 3* 所示。

表 3. Flash——单存储区构成

区域	地址	大小（字节）	名称
主存储器	0x0800 0000 - 0x0800 0FFF	4 K	第 0 页
	0x0800 1000 - 0x0800 1FFF	4 K	第 1 页
	0x0800 2000 - 0x0800 2FFF	4 K	第 2 页
	0x0800 3000 - 0x0800 3FFF	4 K	第 3 页
	. . .	. . .	. . .
	0x080F E000 - 0x080F EFFF	4 K	第 254 页
	0x080F F000 - 0x080F FFFF	4 K	第 255 页
信息块	0x1FFF 0000 - 0x1FFF 6FFF	28 K	系统存储器
	0x1FFF 7000 - 0x1FFF 73FF	1 K	OTP 区域
	0x1FFF 8000 - 0x1FFF 807F	128	选项字节

3.3.2 空数据检查

在 OBL 阶段，加载所有选项后，Flash 接口会检查主存储器的第一个存储单元是否已编程。此检查结果与 boot0 和 boot1 信息一起用于确定系统从哪个存储器区域启动。它可阻止系统从 Flash 主存储器区域启动，例如在尚未编写任何用户代码时。

可从 *Flash 访问控制寄存器 (FLASH_ACR)* 的 EMPTY 位读取 Flash 主存储器空检查状态。可通过软件写入 EMPTY 位，以修改 Flash 主存储器空状态。

3.3.3 误码校正 (ECC)

Flash 中的数据字宽为 72 位：每个双字（64 位）增加了 8 位。ECC 机制支持：

- 单个错误检测和校正
- 两个错误检测

当检测到一个错误并对其进行校正时，*Flash ECC 寄存器 (FLASH_ECCR)* 中的标志 ECCC（ECC 校正）置 1。如果 ECCCIE 置 1，则会生成中断。

检测到两个错误时，*Flash ECC 寄存器 (FLASH_ECCR)* 中的标志 ECCD（ECC 检测）置 1。在这种情况下会生成 NMI。

检测到一个 ECC 错误时，出错双字的地址会保存在 FLASH_ECCR 寄存器的 ADDR_ECC[16:0] 中。ADDR_ECC[2:0] 始终清零。访问地址的 CPU 的总线 ID 保存在 CPUID[2:0] 中。

ECCC 或 ECCD 置 1 时，如果出现新的 ECC 错误，则 FLASH_ECCR 不会更新。仅当 ECC 标志清零后，FLASH_ECCR 才会更新。

注： *对于初始数据：0xFF FFFF FFFF FFFF FFFF，可检测并校正一个错误，但不支持两个错误检测。*

报告 ECC 错误时，如果数据仍存在于当前缓冲区中，则即使 ECCC 和 ECCD 清零，在失效地址处的新读取操作也可能不会生成 ECC 错误。如果这并非期望的行为，则用户必须复位缓存。

3.3.4 读访问延迟

为了准确读取 Flash 数据，必须根据 Flash 时钟 (HCLK4) 频率和器件 V_{CORE} 内部电压范围在 *Flash 访问控制寄存器 (FLASH_ACR)* 中正确编程等待状态数 (LATENCY)。请参见*第 6.1.6 节：动态电压调节管理*。*表 4* 所示为等待状态与 Flash 时钟频率之间的对应关系。

表 4. Flash 时钟 (HCLK4) 频率对应的等待状态数

等待状态 (WS) (LATENCY)	HCLK4 (MHz)	
	V_{CORE} 范围 1	V_{CORE} 范围 2
0 WS（1 个 HCLK 周期）	≤ 18	≤ 6
1 WS（2 个 HCLK 周期）	≤ 36	≤ 12
2 WS（3 个 HCLK 周期）	≤ 54	≤ 16
3 WS（4 个 HCLK 周期）	≤ 64	N/A

POR 后，HCLK4 时钟频率为 4 MHz（范围 1），FLASH_ACR 寄存器中的等待状态 (WS) 为 0。

从待机模式唤醒后，HCLK4 时钟频率为 16 MHz（范围 1），FLASH_ACR 寄存器中的等待状态 (WS) 为 0。

更改 Flash 时钟频率或范围时，必须应用下述软件顺序来调节访问 Flash 所需的等待状态数。

需要提高 CPU 频率时的操作步骤

1. 将新的等待状态数编程到 *Flash 访问控制寄存器 (FLASH_ACR)* 中的 LATENCY 位。
2. 通过回读 *Flash 访问控制寄存器 (FLASH_ACR)* 的 LATENCY，检查用于访问 Flash 的新等待状态数是否设置成功。如果尚未设置成功，则需等待，直至读取到编程的新等待状态数为止。
3. 通过改写 RCC_CFGR 寄存器中的 SW 位来修改系统时钟源。
4. 如有需要，可通过改写 RCC_EXTCFGR 中的 SHDHPRE 位来修改 CPU 时钟预分频比。
5. 此外，还可通过读取 RCC_CFGR 寄存器中相应的时钟源状态（SWS 位）和/或 RCC_EXTCFGR 寄存器中的 AHB 预分频值（SHDHPREF 位），检查新的系统时钟源和/或新的 Flash 时钟预分频值是否设置成功。

需要降低 CPU 频率时的操作步骤

1. 通过改写 RCC_CFGR 寄存器中的 SW 位来修改系统时钟源。
2. 如有需要，可通过改写 RCC_EXTCFGR 中的 SHDHPRE 位来修改 Flash 时钟预分频比。
3. 可通过读取 RCC_CFGR 寄存器中相应的时钟源状态（SWS 位）和/或 RCC_EXTCFGR 寄存器中的 AHB 预分频值（SHDHPREF 位），检查新的系统时钟源和/或新的 Flash 时钟预分频值是否设置成功。如果尚未设置成功，则需等待，直至读取到编程的新系统时钟源和/或新的 Flash 时钟预分频值为止。
4. 将新的等待状态数编程到 *Flash 访问控制寄存器 (FLASH_ACR)* 中的 LATENCY 位。
5. 此外，还可通过回读 *Flash 访问控制寄存器 (FLASH_ACR)* 的 LATENCY，检查是否使用新的等待状态数来访问 Flash。

3.3.5 自适应实时存储器加速器 (ART Accelerator™)

特有的自适应实时 (ART) 存储器加速器对 STM32（符合工业标准，带有 FPU 的 Arm® Cortex®-M4 处理器）进行了优化。该加速器很好地体现了带有 FPU 的 Arm® Cortex®-M4 在 Flash 技术方面的固有性能优势，克服了通常条件下，高速的处理器在运行中需要经常等待 Flash 读取的情况。

为了发挥处理器的全部性能，该加速器将实施指令预取队列和分支缓存，从而提高了 64 位 Flash 的程序执行速度。根据 CoreMark® 基准测试，凭借 ART Accelerator™ 所获得的性能相当于 Flash 在 CPU 频率高达 64 MHz 时以 0 个等待状态执行程序。

指令预取

CPU1 通过 ICode 总线获取指令，通过 DCode 总线获取缓冲池（常数/数据）。预取块用于提高 ICode 总线访问的效率。

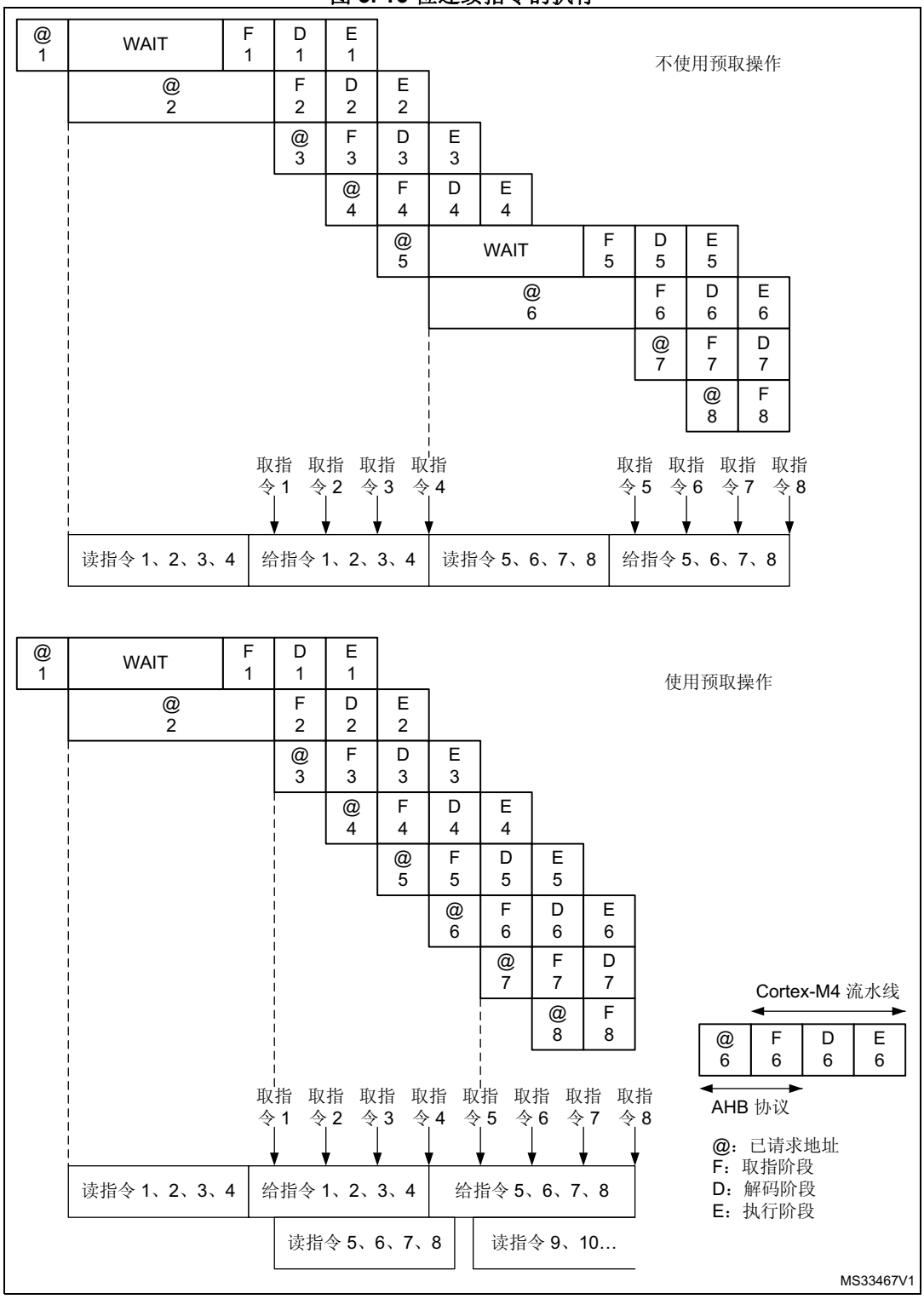
CPU2 通过 S 总线获取指令和缓冲池（常数/数据）。预取块用于提高 S 总线访问的效率。

每个 Flash 读操作可读取 64 位，可以是 2 条 32 位指令，也可以是 4 条 16 位指令，具体取决于启动的程序。此 64 位当前指令行保存在当前缓冲区中。因此对于顺序执行的代码，至少需要 2 个 CPU 周期来执行前一指令行的读取操作。在 CPU 请求当前指令行时，可使用 CPU1 ICode 总线或 CPU2 S 总线的预取操作读取 Flash 中的下一个连续存放的指令行。

可通过将 PRFTEN 位（对于 CPU1，该位处于 *Flash 访问控制寄存器 (FLASH_ACR)* 中；对于 CPU2，该位处于 *Flash CPU2 访问控制寄存器 (FLASH_C2ACR)* 中）置 1 来使能预取功能。当访问 Flash 至少需要一个等待状态时，此功能非常有用。

图 3 所示为需要 3 WS（3 个等待状态）访问 Flash 时连续 16 位指令的执行过程，图中分别介绍了使用和不使用预取操作两种情况。

图 3. 16 位连续指令的执行



处理非顺序执行的代码（有分支）时，指令可能并不存在于当前使用的或预取的指令行中。这种情况下，CPU 等待时间至少等于等待状态数。

如果当前缓冲区中存在循环，则不会执行新的访问。

CPU1 指令缓存存储器 (I-Cache)

为了减少因指令跳转而产生的 CPU1 时间损耗，可将 32 行 4 x 64 位 (1 KB) 的指令保持在指令缓存存储器中。可通过将 *Flash 访问控制寄存器 (FLASH_ACR)* 中的指令缓存使能 (ICEN) 位置 1，为 CPU1 使能此特性。每当出现指令缺失（即请求的指令未存在于当前使用的指令行、预取指令行或指令缓存存储器中）时，系统会将新读取的行复制到指令缓存存储器中。如果 CPU 请求的指令已存在于指令缓存存储器中，则无需任何延时即可立即获取。指令缓存存储器存满后，可采用 LRU（最近最少使用）策略确定指令缓存存储器中待替换的指令行。此特性非常适用于包含循环的代码。

系统复位后指令缓存存储器使能。

CPU1 数据缓存存储器 (D-Cache)

在 CPU 流水线执行阶段，将通过 DCode 总线访问 Flash 中的 CPU1 缓冲池。每次 CPU1 DCode 总线读访问都会获取保存在当前缓冲区中的 64 位。因此，直到提供了请求的数据后，CPU 流水线才会继续执行。为了减少因此而产生的时间损耗，通过 AHB 数据总线 DCode 进行的访问优先于通过 AHB 指令总线 ICode 进行的访问。

如果频繁使用某些缓冲池，可将 *Flash 访问控制寄存器 (FLASH_ACR)* 中的数据缓存使能 (DCEN) 位置 1，来使能 CPU1 数据缓存存储器。此特性的工作原理与指令缓存存储器类似，但保留的数据大小限制在 8 行 4 x 64 位（256 字节）以内。

系统复位后数据缓存存储器使能。

注： *仅当 CPU（而非 DMA）请求数据后 D-Cache 才会激活。*
选项字节块中的数据不可缓存。

CPU2 缓存存储器（S 总线）

为了减少因指令跳转而产生的 CPU2 时间损耗，可将 4 行 1 x 64 位（32 字节）的指令保持在指令缓存存储器中。可通过将 *Flash CPU2 访问控制寄存器 (FLASH_C2ACR)* 中的指令缓存使能 (ICEN) 位置 1，为 CPU2 使能此特性。每当出现指令缺失（即请求的指令未存在于当前使用的指令行、预取指令行或指令缓存存储器中）时，系统会将新读取的行复制到指令缓存存储器中。如果 CPU 请求的指令已存在于指令缓存存储器中，则无需任何延时即可立即获取。指令缓存存储器存满后，可采用 LRU（最近最少使用）策略确定指令缓存存储器中待替换的指令行。此特性非常适用于包含循环的代码。

系统复位后指令缓存存储器使能。

在 CPU 流水线执行阶段，将通过 S 总线访问 Flash 中的 CPU2 缓冲池。每次 CPU2 S 总线读访问都会获取保存在当前缓冲区中的 64 位。因此，直到提供了请求的数据后，CPU 流水线才会继续执行。

Cortex®-M0+ 不支持数据缓存。

3.3.6 Flash 编程和擦除操作

STM32WB55xx 嵌入式 Flash 可采用在线编程或在应用中编程两种方式。

在线编程 (ICP) 方式适用于更新 Flash 的所有内容，更新时使用 JTAG、SWD 协议或系统启动程序支持的接口将用户应用程序（针对 CPU1 和 CPU2）加载到微控制器。ICP 可实现快速而高效的设计迭代，并且避免了不必要的器件封装处理或插接。

与 ICP 方法相比，**在应用中编程 (IAP)** 可通过微控制器支持的任何通信接口（I/O、USB、UART、I^2C 和 SPI 等）将编程数据下载到存储器。IAP 允许用户在应用运行时重新编程 Flash。但是，部分应用程序必须事先通过 ICP 方式编程到 Flash。

如果在 Flash 操作期间发生器件复位，无法保证 Flash 中的内容。

对 Flash 执行编程/擦除操作期间，如果尝试读取 Flash，总线会停止工作。在完成编程/擦除操作后，会正确执行读操作。

注： *在多 CPU 系统中，最好使用信号量来管理 Flash 的编程和擦除操作，以及防止多个 CPU 同时进行操作。*

安全系统 Flash 编程

安全 CPU2 应用程序只能通过在安全 CPU2 上运行的**在应用中编程 (IAP)** 进行编程。只有安全 CPU2 能够将编程数据下载到存储器的安全部分。安全 IAP 允许用户在应用运行时重新编程 Flash。但是，部分应用程序必须事先通过 ICP 方式编程到 Flash。

在线编程 (ICP) 系统启动程序能够与安全 CPU2 IAP 进行通信，从而将编程数据下载到安全存储器中。

解锁 Flash

复位后，*Flash 控制寄存器 (FLASH_CR)* 或 *Flash CPU2 控制寄存器 (FLASH_C2CR)* 不允许执行写操作，以防因电气干扰等原因出现对 Flash 的意外操作。上述寄存器的解锁顺序如下：

1. 在 *Flash 密钥寄存器 (FLASH_KEYR)* 中写入 KEY1 = 0x4567 0123。
2. 在 *Flash 密钥寄存器 (FLASH_KEYR)* 中写入 KEY2 = 0xCDEF 89AB。

如果操作顺序不正确，会锁定 FLASH_CR 寄存器，下次系统复位才会解锁。如果密钥操作顺序不正确，会检测到总线错误并生成硬性故障中断。

也可通过软件将 FLASH_CR 寄存器中的 LOCK 位置 1 来锁定 FLASH_CR 寄存器。

注： *Flash 状态寄存器 (FLASH_SR) 或 Flash CPU2 状态寄存器 (FLASH_C2SR) 中的 BSY 位置 1 时，FLASH_CR 寄存器无法写入。BSY 位置 1 时，对上述寄存器尝试进行写操作会导致 AHB 总线阻塞，直到 BSY 位清零。*

3.3.7 Flash 主存储器擦除顺序

Flash 擦除操作可在页级别（页擦除）或在整个存储器（批量擦除）执行。批量擦除不影响信息块（系统 Flash、OTP 和选项字节）。

Flash 页擦除

CPU1 只能对用户 Flash 的非安全部分进行页擦除。

而安全 CPU2 对用户 Flash 的安全部分和非安全部分均能够进行页擦除。

仅在 *Flash 状态寄存器 (FLASH_SR)* 和 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 的 PESD 位允许的情况下才会启动页擦除。

当页受 PCROP 或 WRP 保护时，将不会被擦除。

表 5. 页擦除概述

页	PCROP	WRP	PCROP_RDP	备注	WRPERR	CPU1 总线错误	CPU2 总线错误
非安全	无	无	x	启动页擦除	无	无	无
		有		中止页擦除（不启动页擦除）	有		
	有	无					
		有					
安全	无	无		由 CPU2 请求，擦除安全页	无	N/A	无
				由 CPU1 请求，中止安全页擦除（不启动安全页擦除）		有	N/A
		有		中止页擦除（不启动页擦除）	有(1)		无
	有	无					
		有					

1. 仅当 CPU2 (Cortex-M0+) 请求 PER 时才生成 WRPERR。当 CPU1 请求 PER 时，不会生成 WRPERR。

页 (4 KB) 擦除的具体步骤如下：

1. 检查 *Flash 状态寄存器 (FLASH_SR)* 或 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 BSY 位，以确认当前未执行任何 Flash 操作。
   检查 *Flash 状态寄存器 (FLASH_SR)* 或 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 PESD 位，以确认是否允许 Flash 编程和擦除操作（即使状态可能会因其他 CPU 发出 Flash 操作请求而发生变化，也仍建议进行这两项检查，以便降低在启动页擦除时接收到总线错误的风险）。
2. 检查并清零之前的编程所导致的全部错误编程标志。否则，PGSERR 将置 1。
3. 在 *Flash 控制寄存器 (FLASH_CR)* 或 *Flash CPU2 控制寄存器 (FLASH_C2CR)* 中，将 PER 位置 1 并选择要擦除的页 (PNB)。
4. 将 FLASH_xxCR 寄存器中的 STRT 位置 1。
5. 等待 FLASH_xxSR 寄存器中的 BSY 位清零。

注： *STRT 位置 1 时，内部振荡器 HSI16 (16 MHz) 自动使能，STRT 位清零时其会自动禁止，但 HSI16 之前已通过 RCC_CR 寄存器中的 HSION 使能的情况除外。*

Flash 批量擦除

CPU1 请求的 Flash 批量擦除操作会被忽略，同时会产生总线错误。

当 PCROP 或 WRP 使能时，将中止所有 Flash 批量擦除操作，并且不会启动任何擦除操作。

表 6. 批量擦除概述

PCROP	WRP	PCROP_RDP	备注	WRPERR	CPU1 总线错误	CPU2 总线错误
无	无		由安全 CPU2 请求，启动 Flash 批量擦除	无	N/A	无
无	有		由安全 CPU2 请求，中止批量擦除（不启动擦除）	有		
有	无					
有	有					
x	x		由 CPU1 请求，中止批量擦除（不启动擦除）	无	有	N/A

批量擦除的具体步骤如下：

1. 检查 *Flash 状态寄存器 (FLASH_SR)* 或 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 BSY 位，以确认当前未执行任何 Flash 操作。
2. 检查并清零之前的编程所导致的全部错误编程标志。否则，PGSERR 将置 1。
3. 将 *Flash 控制寄存器 (FLASH_CR)* 或 *Flash CPU2 控制寄存器 (FLASH_C2CR)* 中的 MER 位置 1。
4. 将 FLASH_xxCR 寄存器中的 STRT 位置 1。
5. 等待 FLASH_xxSR 中的 BSY 位清零。

注： *STRT 位置 1 时，内部振荡器 HSI16 (16 MHz) 自动使能，STRT 位清零时其会自动禁止，但 HSI16 之前已通过 RCC_CR 寄存器中的 HSION 使能的情况除外。*

3.3.8 Flash 主存储器编程顺序

Flash 一次编程 72 位（64 位双字 + 8 位 ECC）。

在之前已编程的双字中，只允许编程全 0 值，尝试编程任何其他值都会使 *Flash 状态寄存器 (FLASH_SR) 或 Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 PROGERR 标志置 1，但双字之前即编程为全 0 值的情况除外。

只能编程双字（2 x 32 位数据）。

- 尝试写入字节（8 位）或半字（16 位）将使 FLASH_xxSR 寄存器中的 SIZERR 标志置 1。
- 尝试写入与双字地址不匹配的双字时，将使 FLASH_xxSR 寄存器中的 PGAERR 标志置 1。

只有安全 CPU2 能够将编程数据下载到存储器的安全部分。CPU1 在安全 Flash 区域进行的 Flash 编程会被忽略，同时会产生总线错误。

标准编程

标准模式下，Flash 编程顺序如下：

1. 检查 *Flash 状态寄存器 (FLASH_SR)* 或 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 BSY 位，以确认当前未执行任何 Flash 主存储器操作。
   检查 *Flash 状态寄存器 (FLASH_SR)* 或 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 PESD 位，以确认是否允许 Flash 编程和擦除操作（即使状态可能会因其他 CPU 发出的 Flash 操作请求而发生变化，也仍建议进行这两项检查，以便降低在开始编程时接收到总线错误的风险）。
2. 检查并清零之前的编程所导致的全部错误编程标志。否则，PGSERR 将置 1。
3. 将 *Flash 控制寄存器 (FLASH_CR)* 或 *Flash CPU2 控制寄存器 (FLASH_C2CR)* 中的 PG 位置 1。
4. 针对所需存储器地址（主存储器块或 OTP 区域内）执行数据写入操作。只能编程双字（64 位）。
   a) 在与双字匹配的地址中写入第一个字。
   b) 写入第二个字。
5. 等待，直到 FLASH_xxSR 寄存器中的 BSY 位清零。
6. 检查 FLASH_xxSR 寄存器中的 EOP 标志是否置 1（表示已成功执行编程操作），并用软件将其清零。
7. 如果不再有编程请求，则将 FLASH_xxSR 寄存器中的 PG 位清零。

注： *Flash 接口接收到适当的序列（双字）后，自动启动编程且 BSY 位置 1。PG 位置 1 时，内部振荡器 HSI16 (16 MHz) 自动使能，PG 位清零时其会自动禁止，但 HSI16 之前已通过 RCC_CR 寄存器中的 HSION 使能的情况除外。*

如果用户需要仅编程一个字，则必须使用擦除值 0xFFFF FFFF 补全双字以启动自动编程。

ECC 由双字计算得出以供编程。

快速编程

此模式允许编程一行（64 个双字，512 字节），无需在编程之前验证 Flash 存储单元，从而缩短了页编程时间，同时可针对每个双字避免高压的上升和下降时间。快速编程过程中，Flash 时钟频率 (HCLK4) 必须至少为 8 MHz。

只有主存储器可编程为快速编程模式。

若要快速实施行编程，必须从 SRAM 执行软件，并且在不重置 CPU 中断向量表时禁止中断。如果请求行编程的 CPU 执行读访问，将导致总线错误。在行编程完成之前，来自任何其他源（其他 CPU 或 DMA）的读访问都将停止（标准双字编程不会导致请求 CPU 发生总线错误，但在标准编程完成之前将停止所有读取操作）。

标准模式下，Flash 主存储器编程顺序如下：

1. 执行批量擦除。否则，PGSERR 将置 1。
2. 检查 *Flash 状态寄存器 (FLASH_SR)* 或 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 BSY 位，以确认当前未执行任何 Flash 主存储器操作。
   检查 *Flash 状态寄存器 (FLASH_SR)* 或 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 PESD 位，以确认是否允许 Flash 编程和擦除操作（即使状态可能会因其他 CPU 发出 Flash 操作请求而发生变化，也仍建议进行这两项检查，以便降低在开始编程时接收到总线错误的风险）。

3. 检查并清零之前的编程所导致的全部错误编程标志。
4. 将 *Flash 控制寄存器 (FLASH_CR)* 或 *Flash CPU2 控制寄存器 (FLASH_C2CR)* 中的 FSTPG 位置 1。
5. 写入 64 个双字以编程一行（512 字节）。
6. 等待，直到 FLASH_xxSR 寄存器中的 BSY 位清零。
7. 检查 FLASH_xxSR 寄存器中的 EOP 标志是否置 1（表示已成功执行编程操作），并用软件将其清零。
8. 如果不再有编程请求，则将 FLASH_xxSR 寄存器中的 FSTPG 位清零。

注： *当正在执行读操作时，如果尝试在快速编程模式下进行写操作，则会中止编程，且不会发出任何系统通知（无错误标志置 1）。*

Flash 接口接收到第一个双字后，编程自动启动。对第一个双字施加高压时，BSY 位置 1，最后一个双字已编程或发生错误时，该位清零。FSTPG 位置 1 时，内部振荡器 HSI16 (16 MHz) 自动使能，FSTPG 位清零时其会自动禁止，但 HSI16 之前已通过 RCC_CR 寄存器中的 HSION 使能的情况除外。

64 个双字必须连续写入。对于所有编程，Flash 均保持高压。两个双字写入请求之间的最长时间为编程时间（约为 20 µs）。如果第二个双字在此编程时间过后传到，则快速编程被中断，MISSERR 置 1。

在 2 次擦除之间，对于一整行，高压持续时间不得超过 8 ms。可通过基于大于等于 8 MHz 的时钟系统连续写入的 64 个双字的序列来保证这一点。设置快速编程后内部超时计数器计数 7 ms，并在超时后会停止编程。此时，FASTERR 位将置 1。

如果发生错误，则高压停止，不会对下一个要编程的双字进行编程。总之，之前所有的双字均已正确编程。

由标志指示编程错误

可检测到多种错误。若发生错误，Flash 操作（编程或擦除）会中止。

- **PROGERR**：编程错误

  在标准编程过程中：如果要写入非全 0 值的字之前未擦除，则 PROGERR 置 1（要编程的值全为 0 时除外）。如果发生任何其他错误（SIZERR、PAGERR、PGSERR、WRPERR），即使重新编程字时无擦除错误，PROGRERR 可能也不会置 1。

- **SIZERR**：大小编程错误

  在标准编程或快速编程过程中：只能编程双字且只能写入 32 位数据。如果写入一个字节或半字，则 SIZERR 置 1。

- **PGAERR**：编程对齐错误

  如果发生以下事件，PGAERR 位会置 1：

  – 每个 CPU 单独检查对齐编程错误。如果同时多个 CPU 编程，则所有 CPU 均不执行检查。应使用 HSEM 或其他软件机制来防止同时多个 CPU 编程。

  – 在标准编程过程中：要编程的第一个字与双字地址不匹配，或第二个字不属于同一双字地址。

  – 在快速编程过程中：要编程的数据与之前编程的双字不属于同一行，或者要编程的地址小于或等于之前的地址。

- **PGSERR**：编程顺序错误

  如果发生以下事件，PGSERR 位会置 1：

  – 对于标准编程顺序或快速编程顺序：PG 和 FSTPG 清零时写入数据。
  – 对于标准编程顺序或快速编程顺序：PG 或 FSTPG 置 1 后，MER 和 PER 不清零。
  – 对于快速编程顺序：将 FSTPG 位置 1 后执行批量擦除操作。
  – 对于批量擦除顺序：MER 置 1 后，PG、FSTPG 和 PER 不清零。
  – 对于页擦除顺序：PER 置 1 后，PG、FSTPG 和 MER 不清零。
  – 如果由于之前发生编程错误，导致 PROGERR、SIZERR、PGAERR、WRPERR、MISSERR、FASTERR 或 PGSERR 置 1，则 PGSERR 也会置 1。
  – 对于快速编程顺序：要编程的行与先前擦除的行位于不同的页。如果已完成批量擦除，则允许对高位页上的行进行编程，但不允许在低位页上进行编程。

- **WRPERR**：写保护错误

  如果发生以下事件，WRPERR 位会置 1：

  – 尝试在写保护区域 (WRP) 或 PCROP 区域执行编程或擦除操作。
  – 尝试在一页或多页受 WRP 或 PCROP 保护时，执行批量擦除操作。
  – 读保护 (RDP) 设为级别 1 时，已连接调试功能或从 SRAM 或系统 Flash 进行启动。
  – 尝试在读保护 (RDP) 设置为级别 2 时修改选项字节，由安全 CPU2 发出请求的情况除外。

- **MISSERR**：快速编程数据丢失错误

  在快速编程过程中：所有数据必须连续写入。如果之前的数据编程已完成，并且要编程的下一个数据尚未写入，则 MISSERR 置 1。

- **FASTERR**：快速编程错误

  在快速编程过程中：如果发生以下事件，FASTERR 会置 1：

  – FSTPG 位置 1 的时间超过 7 µs，这会生成超时检测。
  – 行快速编程被 MISSERR、PGAERR、WRPERR 或 SIZERR 中断。

如果在编程或擦除操作期间出现错误，则 *Flash 状态寄存器 (FLASH_SR)* 或 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的以下错误标志之一将置 1：

- PROGERR、SIZERR、PGAERR、PGSERR、MISSERR（编程错误标志）
- WRPERR（保护错误标志）

这种情况下，FLASH_xxSR 寄存器中的操作错误标志 OPERR 置 1，并且如果 *Flash 控制寄存器 (FLASH_CR)* 或 *Flash CPU2 控制寄存器 (FLASH_C2CR)* 中的错误中断使能位 ERRIE 置 1，则将产生一个中断。

注： *如果检测到多个连续错误（例如，在对 Flash 进行 DMA 传输期间），则直到连续写操作请求结束，这些错误标志才会清零。*

导致总线错误的编程错误

下列错误条件不会生成错误标志，但会导致总线错误。

- 在 RDP 级别为 1 且从系统 Flash 或 SRAM1 启动时对任意页执行 AHB 写操作
- 在 Flash 断电时执行 AHB 写操作
- 通过调试器读写 Flash
- 请求对 Flash 进行快速行编程的源在 Flash 正在进行快速行编程时对 Flash 进行读操作
- 在前一个请求尚未完成时发起新的编程请求
- 在相邻两次双字编程访问之间对 FLASH_CR 寄存器进行写操作
- 在 PESDx 有效（置 1）时对 FLASH_CR 寄存器进行写操作
- 在 FLASH_KEYR 或 FLASH_OPTKEYR 寄存器中写入错误密钥
- 在解锁相应功能后对 FLASH_KEYR 或 FLASH_OPTKEYR 进行任何后续写操作

基于页的行编程中的 PGSERR 和 PGAERR

表 7 介绍了执行快速编程时 PGSERR 和 PGAERR 的处理方式。

表 7. 基于页的行编程中的错误

最后一页/行	当前页/行	MER 有效	PER 有效
页 [x]/行 [y]	页 [x]/行 [y-n]	PGAERR	PGAERR
页 [x]/行 [y]	页 [x-n]/行 [任意]	PGAERR 和 PGSERR	PGAERR 和 PGSERR
页 [x]/行 [y]	页 [x+n]/行 [任意]	无错误	PGAERR

在系统复位后，如果既不执行 MER，也不执行 PER，则尝试执行任何编程操作都会导致 PGAERR 和 PGSERR 错误。

编程与缓存

如果 Flash 写访问影响数据缓存中的数据，Flash 写访问将修改存储器中的数据和缓存中的数据。

如果 Flash 中的擦除操作也涉及数据缓存或指令缓存中的数据，则用户必须确保在代码执行期间访问这些数据之前先将其重新写入缓存。执行擦除操作后，缓存内容将失效。

注： *I/D 缓存只有在被禁止 (I/DCEN=0) 的情况下才能刷新。*

3.4 FLASH 选项字节

3.4.1 选项字节说明

选项字节由最终用户根据具体的应用要求进行配置。以如下配置为例：可在硬件或软件模式下选择看门狗（参见*第 3.4.2 节：选项字节编程*）。

一个双字基于选项字节进行拆分，如*表 8* 所示。

表 8. 选项字节格式

63-56	55-48	47-40	39-32	31-24	23-16	15 -8	7-0
补码 选项字节 3	补码 选项字节 2	补码 选项字节 1	补码 选项字节 0	选项 字节 3	选项 字节 2	选项 字节 1	选项 字节 0

表 9 显示了这些字节在信息块中的构成。选项字节可从*表 9* 列出的存储单元或从选项字节寄存器中读取：

- *Flash 选项寄存器 (FLASH_OPTR)*
- *Flash PCROP 区域 A 起始地址寄存器 (FLASH_PCROP1ASR)*
- *Flash PCROP 区域 A 结束地址寄存器 (FLASH_PCROP1AER)*
- *Flash PCROP 区域 B 起始地址寄存器 (FLASH_PCROP1BSR)*
- *Flash PCROP 区域 B 结束地址寄存器 (FLASH_PCROP1BER)*
- *Flash WRP 区域 A 地址寄存器 (FLASH_WRP1AR)*
- *Flash WRP 区域 B 地址寄存器 (FLASH_WRP1BR)*
- *Flash IPCC 邮箱数据缓冲区地址寄存器 (FLASH_IPCCBR)*
- *安全 Flash 起始地址寄存器 (FLASH_SFR)*
- *Flash 安全 SRAM2 起始地址和 CPU2 复位向量寄存器 (FLASH_SRRVR)*

表 9. 选项字节构成

地址[1]	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
1FFF8000	AGC_TRIM			未使用	nBOOT0	nSWBOOT0	SRAM2_RST	SRAM2_PE	nBOOT1	未使用			WWDG_SW	IWDG_STBY	IWDG_STOP	IWDG_SW	未使用	nRST_SHDW	nRST_STBY	nRST_STOP	BOR_LEV			ESE	RDP							
1FFF8008	未使用																							PCROP1A_STRT								
1FFF8010	PCROP_RDP	未使用																						PCROP1A_END								
1FFF8018	未使用								WRP1A_END								未使用							WRP1A_STRT								

表 9. 选项字节构成（续）

地址[1]	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
1FFF8020	未使用								WRP1B_END								未使用								WRP1B_STRT							
1FFF8028	未使用																								PCROP1B_STRT							
1FFF8030	未使用																								PCROP1B_END							
1FFF8038 到 1FFF8060	未使用																															
1FFF8068	未使用																IPCCDBA															
1FFF8070	未使用																			DDS	未使用				FSD	SFSA						
1FFF8078	C2OPT	NBRSD	SNBRSA						未使用	BRSD	SBRSA						SBRV															
1FFF8FF8	OPVAL																															

1. 双字地址的高 32 位包含低 32 位的反转数据。

用户和读保护选项字节

Flash 地址：0x1FFF 8000

ST 生产值：0x3FFF F1AA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AGC_TRIM[2:0]			Res.	n BOOT0	nSW BOOT0	SRAM2 _RST	SRAM2 _PE	n BOOT1	Res.	Res.	Res.	WWDG _SW	IWGD _STDBY	IWDG _STOP	IWDG _SW
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
Res.	nRST _SHDW	nRST _STDBY	nRST _STOP	BOR_LEV[2:0]			ESE	RDP[7:0]							
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:29 **AGC_TRIM**：自动增益控制微调 (Automatic Gain Control trimming)
默认值为 0b001

位 28 未使用

位 27 **nBOOT0**：nBOOT0 选项位 (nBOOT0 option bit)
0：nBOOT0=0
1：nBOOT0=1

位 26 **nSWBOOT0**：软件 BOOT0 (Software BOOT0)
0：BOOT0 取自选项位 nBOOT0
1：BOOT0 取自 PH3/BOOT0 引脚

位 25 **SRAM2_RST**：发生系统复位时擦除 SRAM2 和 PKA RAM (SRAM2 and PKA RAM erase when system reset)
0：发生系统复位时擦除 SRAM2 和 PKA RAM
1：发生系统复位时不擦除 SRAM2 和非安全 PKA RAM

位 24 **SRAM2_PE**：SRAM2 奇偶校验使能 (SRAM2 parity check enable)
0：使能 SRAM2 奇偶校验
1：禁止 SRAM2 奇偶校验

位 23 **nBOOT1**：启动配置 (Boot configuration)
该位与 BOOT0 引脚或选项位 nBOOT0（具体取决于 nSWBOOT0 选项位配置）搭配使用，用于选择从 Flash 主存储器、SRAM1 或系统存储器启动模式。请参见*第 2.3 节：启动配置*。

位 22:20 未使用

位 19 **WWDG_SW**：窗口看门狗选择 (Window watchdog selection)
0：硬件窗口看门狗
1：软件窗口看门狗

位 18 **IWDG_STDBY**：在待机模式下冻结独立看门狗计数器 (Independent watchdog counter freeze in Standby mode)
0：在待机模式下冻结独立看门狗计数器
1：在待机模式下运行独立看门狗计数器

位 17 **IWDG_STOP**：在停止模式下冻结独立看门狗计数器 (Independent watchdog counter freeze in stop mode)
0：在停止模式下冻结独立看门狗计数器
1：在停止模式下运行独立看门狗计数器

位 16 **IDWG_SW**：独立看门狗选择 (Independent watchdog selection)
0：硬件独立看门狗
1：软件独立看门狗

位 15 未使用

位 14 **nRST_SHDW**
0：进入关断模式时产生复位
1：进入关断模式时不产生复位

位 13 **nRST_STDBY**
0：进入待机模式时产生复位
1：进入待机模式时不产生复位

位 12 **nRST_STOP**
0：进入停机模式时产生复位
1：进入停机模式时不产生复位

位 11:9 **BOR_LEV**：BOR 复位级别 (BOR reset Level)
这些位包含激活/释放复位信号所需达到的 VDD 供电电压阈值。
000：BOR 0 级。复位阈值电压约 1.7 V
001：BOR 1 级。复位阈值电压约 2.0 V
010：BOR 2 级。复位阈值电压约 2.2 V
011：BOR 3 级。复位阈值电压约 2.5 V
100：BOR 4 级。复位阈值电压约 2.8 V

位 8 **ESE**：系统安全使能标志 (System security enabled flag)
0：禁止安全功能
1：使能安全功能

位 7:0 **RDP**：读保护级别 (Read protection level)
0xAA：级别 0，未激活读保护
0xCC：级别 2，激活芯片读保护
其他：级别 1，激活存储器读保护

PCROP1A 起始地址选项字节

Flash 地址：0x1FFF 8008

ST 生产值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1A_STRT[8:0]								
							r	r	r	r	r	r	r	r	r

位 31:9 未使用

位 8:0 **PCROP1A_STRT**：PCROP1A 区域起始偏移 (PCROP1A area start offset)
PCROP1A_STRT 包含 PCROP1A 区域的第一个 2 KB 页。

PCROP1A 结束地址选项字节

Flash 地址：0x1FFF 8010

ST 生产值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PCROP_RDP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1A_END[8:0]								
							r	r	r	r	r	r	r	r	r

位 31 **PCROP_RDP**：RDP 级别下降时，PCROP 区域保持 (PCROP area preserved when RDP level decreased)

该位只置 1。执行由 RDP 从级别 1 更改到级别 0 导致的完全批量擦除后，该位复位。

0：RDP 级别从级别 1 降低到级别 0 时，不擦除 PCROP 区域。

1：RDP 级别从级别 1 降低到级别 0 时，擦除 PCROP 区域（完全批量擦除）。

位 30:9 未使用

位 8:0 **PCROP1A_END**：PCROP1A 区域结束偏移 (PCROP1A area end offset)
PCROP1A_END 包含 PCROP1A 区域的最后一个 2 KB 页。

WRP 区域 A 地址选项字节

Flash 地址：0x1FFF 8018

ST 生产值：0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_END[7:0]							
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_STRT[7:0]							
								r	r	r	r	r	r	r	r

位 31:24 未使用

位 23:16 **WRP1A_END**：WRP 区域“A”结束偏移 (WRP area “A” end offset)
WRPA1_END 包含 WRP 区域“A”的最后一个 4 KB 页。

位 15:8 未使用

位 7:0 **WRP1A_STRT**：WRP 区域“A”起始偏移 (WRP area “A” start offset)
WRPA1_STRT 包含 WRP 区域“A”的第一个 4 KB 页。

WRP 区域 B 地址选项字节

Flash 地址：0x1FFF 8020

ST 生产值：0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_END[7:0]							
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_STRT[7:0]							
								r	r	r	r	r	r	r	r

位 31:24 未使用

位 23:16 **WRP1B_END**：WRP 区域“B”结束偏移 (WRP area “B” end offset)
WRPB1_END 包含 WRP 区域“B”的最后一个 4 KB 页。

位 15:8 未使用

位 7:0 **WRP1B_STRT**：WRP 区域“B”起始偏移 (WRP area “B” start offset)
WRPB1_STRT 包含 WRP 区域“B”的第一个 4 KB 页。

PCROP1B 起始地址选项字节

Flash 地址：0x1FFF 8028

ST 生产值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1B_STRT[8:0]								
							r	r	r	r	r	r	r	r	r

位 31:9 未使用

位 8:0 **PCROP1B_STRT**：PCROP1B 区域起始偏移 (PCROP1B area start offset)
PCROP1B_STRT 包含 PCROP1B 区域的第一个 2 KB 页。

PCROP1B 结束地址选项字节

Flash 地址：0x1FFF 8030

ST 生产值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1B_END[8:0]								
							r	r	r	r	r	r	r	r	r

位 31:9 未使用

位 8:0 **PCROP1B_END**：PCROP1B 区域结束偏移 (PCROP1B area end offset)
PCROP1B_END 包含 PCROP1B 区域的最后一个 2 KB 页。

IPCC 邮箱数据缓冲区地址选项字节

Flash 地址：0x1FFF 8068

ST 生产值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	IPCCDBA[13:0]													
		r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:14 未使用

位 13:0 **IPCCDBA**：IPCC 邮箱数据缓冲区基址偏移 (IPCC mailbox data buffer base address offset)
IPCCDBA 包含 SRAM2 中 IPCC 邮箱数据缓冲区的第一个双字偏移。

安全 Flash 起始地址选项字节

Flash 地址：0x1FFF 8070

ST 生产值：0xFFFF FEXX

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DDS	Res.	Res.	Res.	FSD	SFSA[7:0]							
			r				r	r	r	r	r	r	r	r	r

位 31:13 未使用

位 12 **DDS**：禁止 CPU2 调试访问 (Disable CPU2 Debug access)
0：使能 CPU2 调试访问。
1：禁止 CPU2 调试访问。

位 11:9 未使用

位 8 **FSD**：Flash 安全禁止 (Flash security disable)
FSD=1：系统和 Flash 非安全
FSD=0：系统和 Flash 安全。SFSA[7:0] 包含安全 Flash 区域的第一个 4K 页的起始地址。

位 7:0 **SFSA**：安全 Flash 起始地址 (Secure Flash start address)
系统和 Flash 安全。SFSA[7:0] 包含安全 Flash 区域的第一个 4K 页的起始地址。

安全 SRAM2 起始地址和 CPU2 复位向量选项字节

Flash 地址：0x1FFF 8078

ST 生产值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C2OPT	NBRSD	SNBRSA[4:0]					Res.	BRSD	SBRSA[4:0]					SBRV[17:16]	
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
SBRV[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 **C2OPT**：CPU2 启动复位向量存储器选择 (CPU2 boot reset vector memory selection)
0：SBRV 偏移将以 0x2000 0000 为起始地址对 SRAM1 或 SRAM2 进行寻址（请注意，SBRV 值将保持在 SRAM 区域内）。
1：SBRV 偏移将以 0x0800 0000 为起始地址对 Flash 进行寻址。

位 30 **NBRSD**：非备份 SRAM2b 安全禁止 (non-backup SRAM2b security disable)
NBRSD=1：SRAM2b 非安全。
NBRSD=0：SRAM2b 安全。SNBRSA[4:0] 包含安全非备份 SRAM2b 区域的第一个 1K 页的起始地址。

位 29:25 **SNBRSA**：安全非备份 SRAM2b 起始地址 (Secure non-backup SRAM2b start address)
NBRSD=0：SRAM2b 安全。SNBRSA[4:0] 包含安全非备份 SRAM2b 区域的第一个 1K 页的起始地址。

位 24 未使用

位 23 **BRSD**：备份 SRAM2a 安全禁止 (backup SRAM2a security disable)
BRSD=1：SRAM2a 非安全。
BRSD=0：SRAM2a 安全。SBRSA[4:0] 包含安全备份 SRAM2a 区域的第一个 1K 页的起始地址。

位 22:18 **SBRSA**：安全备份 SRAM2a 起始地址 (Secure backup SRAM2a start address)
BRSD=0：SRAM2a 安全。SBRSA[4:0] 包含安全备份 SRAM2a 区域的第一个 1K 页的起始地址。

位 17:0 **SBRV**：CPU2 启动复位向量 (CPU2 boot reset vector)
包含 C2OPT 所选择的存储器区域内按字对齐的 CPU2 启动复位起始地址偏移。

3.4.2 选项字节编程

复位后，*Flash 控制寄存器 (FLASH_CR)* 和 *Flash CPU2 控制寄存器 (FLASH_C2CR)* 中的选项相关位受写保护。要针对选项字节页执行任何操作，*Flash 控制寄存器 (FLASH_CR)* 中的选项锁定位 OPTLOCK 必须清零。此寄存器的解锁顺序如下：

1. 使用 LOCK 清零序列解锁 FLASH_CR（参见*解锁 Flash*）
2. 在 *Flash 选项密钥寄存器 (FLASH_OPTKEYR)* 中写入 OPTKEY1=0x08192A3B
3. 在 *Flash 选项密钥寄存器 (FLASH_OPTKEYR)* 中写入 OPTKEY2=0x4C5D6E7F

如果操作顺序不正确，会锁定 Flash 选项寄存器，下次系统复位才会解锁。如果密钥操作顺序不正确，会检测到总线错误并生成硬性故障中断。

通过软件将 OPTLOCK 位置 1 后，可防止用户选项发生意外的擦除/编程操作。

注： *如果 LOCK 由软件置 1，则 OPTLOCK 也自动置 1。*

注： *在多 CPU 系统中，最好使用信号量来管理选项编程，以及防止多个 CPU 同时进行选项编程。*

修改用户选项

选项字节与主存储器用户地址的编程方式不同。

要修改用户选项值，请执行以下步骤：

1. 按照上述清零序列将 OPTLOCK 选项锁定位清零。
2. 在选项寄存器中写入所需选项值。
3. 检查 *Flash 状态寄存器 (FLASH_SR)* 或 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 BSY 位，以确认当前未执行任何 Flash 操作。
   检查 *Flash 状态寄存器 (FLASH_SR)* 或 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 PESD 位，以确认是否允许 Flash 编程和擦除操作（即使状态可能会因其他 CPU 发出 Flash 操作请求而发生变化，也仍建议进行这两项检查，以便降低在修改用户选项时接收到总线错误的风险）。
4. 将 *Flash 控制寄存器 (FLASH_CR)* 中的选项起始位 OPTSTRT 置 1。
5. 等待 BSY 位清零。

注： *先擦除用户选项字节页，然后以 Flash 选项寄存器中包含的值对所有选项字节进行编程，从而可自动修改一个选项值。*

安全用户选项

安全选项字节 *安全 Flash 起始地址寄存器 (FLASH_SFR)* 和 *Flash 安全 SRAM2 起始地址和 CPU2 复位向量寄存器 (FLASH_SRRVR)* 只能由安全 CPU2 写入。

选项字节加载

BSY 位清零后，所有新选项都将更新到 Flash，但不应用到系统。读取选项寄存器仍将返回上次加载的选项字节值，新选项仅在加载后才对系统有影响。

在以下两种情况下执行选项字节加载：

- *Flash 控制寄存器 (FLASH_CR)* 中的 OBL_LAUNCH 位置 1 时
- 电源复位后（BOR 复位或退出待机/关断模式）

选项字节加载器读取选项块并将数据存储到内部选项寄存器。这些内部寄存器配置系统，可由软件读取。将 OBL_LAUNCH 置 1 会产生复位，因此在系统复位下执行选项字节加载。

每个选项位在同一双字中也有各自的补码。选项加载期间，验证选项位及其补码可检查是否已正确进行加载。

选项字节加载期间，选项由双字读取。选项字的 ECC 在 OBL 期间不予考虑，其仅在选项区域的直接软件读取期间予以考虑。

如果字与其补码匹配，则将选项字/字节复制到选项寄存器。

如果字与其补码不匹配，则 OPTVERR 状态位置 1。不匹配值被强制放入选项寄存器：

- 对于 USR OPT 选项，不匹配值为所有选项均为“1”，为“000”的 BOR_LEV（最低阈值）除外
- 对于 WRP 选项，不匹配值为默认值“无保护”
- 对于 RDP 选项，不匹配值为默认值“级别 1”
- 对于 PCROP，不匹配值为“所有存储器均受保护”
- 对于 FSD 和 SFSA 选项，不匹配值为“所有存储器（Flash、SRAM2a 和 SRAM2b）均安全”
- 对于 BRSD、SBRSA 和 NBRSD、SNBRASA 选项，不匹配值为“SRAM2 存储器部分安全”
- 对于 DDS 选项，不匹配值为“禁止 CPU2 调试”
- 对于 C2OPT 和 SBRV 选项，不匹配值为“CPU2 从最后一个 Flash 页的起始地址启动（安全启动）”
- 对于 OPTVAL 选项，不匹配值为“无效”

如果 OPTVAL 选项指示“无效”，则所有存储器（Flash、SRAM2a 和 SRAM2b）均安全。

表 10. 选项加载控制

OPTVERR	OPTNV	说明
0	0	选项正确加载，OPTVAL 为“有效”。 – 根据选项来应用安全功能。
0	1	不会发生
1	0	OPTVAL 选项正确加载为“有效”，但部分或全部其他选项和工程位损坏，加载了不匹配值。 – 正确加载安全选项后，将根据加载的安全选项值来应用安全功能。 – 当安全选项损坏时，安全功能将应用于整个存储器，如加载的不匹配值所示。
1	1	部分或全部选项和工程位损坏，加载了不匹配值。OPTVAL 正确加载为“无效”。无论加载的安全选项值如何，安全功能都应用于整个存储器。

在系统复位（上升沿）时，将内部选项寄存器复制到可由软件读写的选项寄存器中：

- FLASH_OPTR
- FLASH_PCROPxySR（x = 1，y = A 或 B）
- FLASH_PCROPxyER（x = 1，y = A 或 B）
- FLASH_WRPxyR（x = 1，y = A 或 B）
- FLASH_IPCCDBA
- FLASH_SFR
- FLASH_SRRVA

上述寄存器也用于修改选项。如果这些寄存器未由用户修改，则会反映系统的选项状态。请参见*修改用户选项*以获得更多信息。

3.5 FLASH UID64

64 位唯一器件标识 (UID64) 存储在 Flash 中，可由 CPU 访问。

表 11. UID64 构成

地址	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
0x1FFF 7580	唯一器件编号																															
0x1FFF 7584	ST 公司 ID																								器件 ID							

UID64 在器件生产期间编程，为每个器件提供唯一代码

- 24 位 ST 公司 ID 将返回 0x00 80 E1
- STM32WB55xx 的 8 位器件 ID 将返回 0x25
- 32 位唯一器件编号是一个序号，因每个 STM32WB55xx 器件而异

可通过固件使用 64 位 UID64 来获取 BLE 48 位器件地址 EUI-48 或 802.15.4 64 位器件地址 EUI-64。

3.6 Flash 保护

可对 Flash 主存储器进行保护，使其具有读保护 (RDP)，以防其遭受外部访问。也可对页进行保护，以防因程序指针错乱而发生意外的写操作 (WRP)。写保护 WRP 粒度为 4 KB。除了 RDP 和 WRP 之外，还可防止 Flash 遭受第三方的读取和写入 (PCROP)。PCROP 粒度为 2 KB。

可为 Flash 主存储器的部分区域提供保护，并向 CPU2 授予对这部分存储区域的独占访问权限。

3.6.1 读保护 (RDP)

通过将 RDP 选项字节置 1，然后应用系统复位来重载新的 RDP 选项字节，可激活读保护。读保护可对 Flash 主存储器、选项字节、备份寄存器（RTC 中的 RTC_BKPxR）和 SRAM2 进行保护。

注： *如果在调试器仍通过 JTAG/SWD 连接时设置读保护，则应用 POR（上电复位），而非系统复位。*

有三种读保护级别：无保护（级别 0）到最大保护或禁止调试（级别 2）。

RDP 选项字节及其补码包含成对值时，Flash 受保护，如*表 12* 所示。

表 12. Flash 读保护状态

RDP 字节值	RDP 补码值	读保护级别
0xAA	0x55	级别 0
除 0xAA 或 0xCC 外的任意值	除 0x55 和 0x33 外的任意值（不一定是补码值）	级别 1（默认值）
0xCC	0x33	级别 2

无论保护级别如何，系统存储器区域均可进行读访问。对于编程/擦除操作，则不可访问。

级别 0：无保护

可对 Flash 主存储器区域执行读取、编程和擦除操作。也可对选项字节、SRAM2 和备份寄存器进行所有操作。

级别 1：读保护

RDP 选项字节被擦除时，此为默认保护级别。RDP 值是除 0xAA 和 0xCC 外的任意值时，或者即使补码不正确，也会定义此级别。

- **用户模式：**在用户模式（**启动 Flash**）下执行的代码可访问 Flash 主存储器、选项字节、SRAM2 和备份寄存器，可对其进行所有操作。
- **调试、启动 RAM 和启动程序模式：**在调试模式下或从启动 RAM 或启动程序运行代码时，Flash 主存储器、备份寄存器（RTC 中的 RTC_BKPxR）和 SRAM2 均不可访问。在上述模式下，对 Flash 进行读访问或写访问会生成总线错误和硬性故障中断。

注意： 如果配置了级别 1 且未定义 PCROP 区域，则必须将 PCROP_RDP 位置 1（RDP 级别从级别 1 降到级别 0 时，进行完全批量擦除）。如果配置了级别 1 且定义了 PCROP 区域，则在需要通过 RDP（而非 PCROP）保护用户代码时，不得将用户代码置于包含 PCROP 区域的页。

级别 2：禁止调试

在此级别下，可保证保护级别 1。此外，CPU1 和 CPU2 调试端口、从 RAM 启动（启动 RAM 模式）和从系统存储器启动（启动程序模式）不再可用。在用户执行模式（启动 FLASH 模式）下，允许对 Flash 主存储器执行所有操作。相反，只能对选项字节执行读取操作和安全写入操作。选项字节只能由安全 CPU2 进行编程和擦除。

无法从非安全应用侧删除级别 2：此为不可逆操作。尝试修改选项字节时，FLASH_xxxSR 寄存器中的保护错误标志 WRPERR 置 1，并且可能会生成中断。

注： *在复位时调试功能也被禁止。*

注： *意法半导体无法对设为保护级别 2 的器件做失效分析。*

更改读保护级别

通过将 RDP 字节的值更改为除 0xCC 外的任意值，可轻松从级别 0 更改为级别 1。通过在 RDP 字节中编程 0xCC 值，可直接从级别 0 或级别 1 更改为级别 2。更改为级别 2 后，无法再更改读保护级别。

将 RDP 编程为值 0xAA 来从级别 1 更改为级别 0 时，如果 *Flash PCROP 区域 A 结束地址寄存器 (FLASH_PCROP1AER)* 中的 PCROP_RDP 置 1，则会对 Flash 主存储器执行批量擦除。也会擦除备份寄存器（RTC 中的 RTC_BKPxR）、SRAM2 和 PKA SRAM。除 PCROP 保护外的用户选项设置为之前从 FLASH_OPTR 和 FLASH_WRPxyR（x = 1，y = A 或 B）复制的值。PCROP 被禁止。OTP 区域不受批量擦除操作的影响，同样保持不变。

如果 FLASH_PCROP1AER 中的 PCROP_RDP 位清零，则完全批量擦除由部分批量擦除所代替，该部分批量擦除是连续页擦除（PCROP 保护的页除外）。这样做是为了保持 PCROP 代码。只有擦除 Flash 后，才会使用之前的选项值对选项重新编程。这同样适用于 FLASH_PCROPxySR 和 FLASH_PCROPxyER 寄存器（x = 1，y = A 或 B）。

请求的批量擦除将执行部分批量擦除，即连续页擦除，由 CPU2 安全 (SFSA) 保护的页除外。这样做是为了保持 CPU2 安全代码。

表 13. RDP 从级别 1 降为级别 0 以及存储器擦除

SFSA	PCROP	PCROP_RDP	备注
部分	无	x	所有非安全页的 Flash 多页擦除、SRAM2 和 PKA RAM 以及备份寄存器擦除。（安全 Flash 页保留）。
	部分	1	
		0	所有非 PCROP 页和非安全页的 Flash 多页擦除、SRAM2 和 PKA RAM 以及备份寄存器擦除（PCROP Flash 页和安全 Flash 页保留）。
	完整非安全		保留 Flash、SRAM2 和 PKA RAM 以及备份寄存器。
完整 Flash	x	x	保留 Flash、SRAM2 和 PKA RAM 以及备份寄存器。

注： *只有在已激活级别 1 并请求级别 0 时，才会执行部分批量擦除。当提高保护级别（0→1、1→2、0→2）时，不会执行批量擦除。*

为验证保护级别更改，必须通过 Flash 控制寄存器中的 OBL_LAUNCH 位重载选项字节。

图 4. 更改读保护 (RDP) 级别

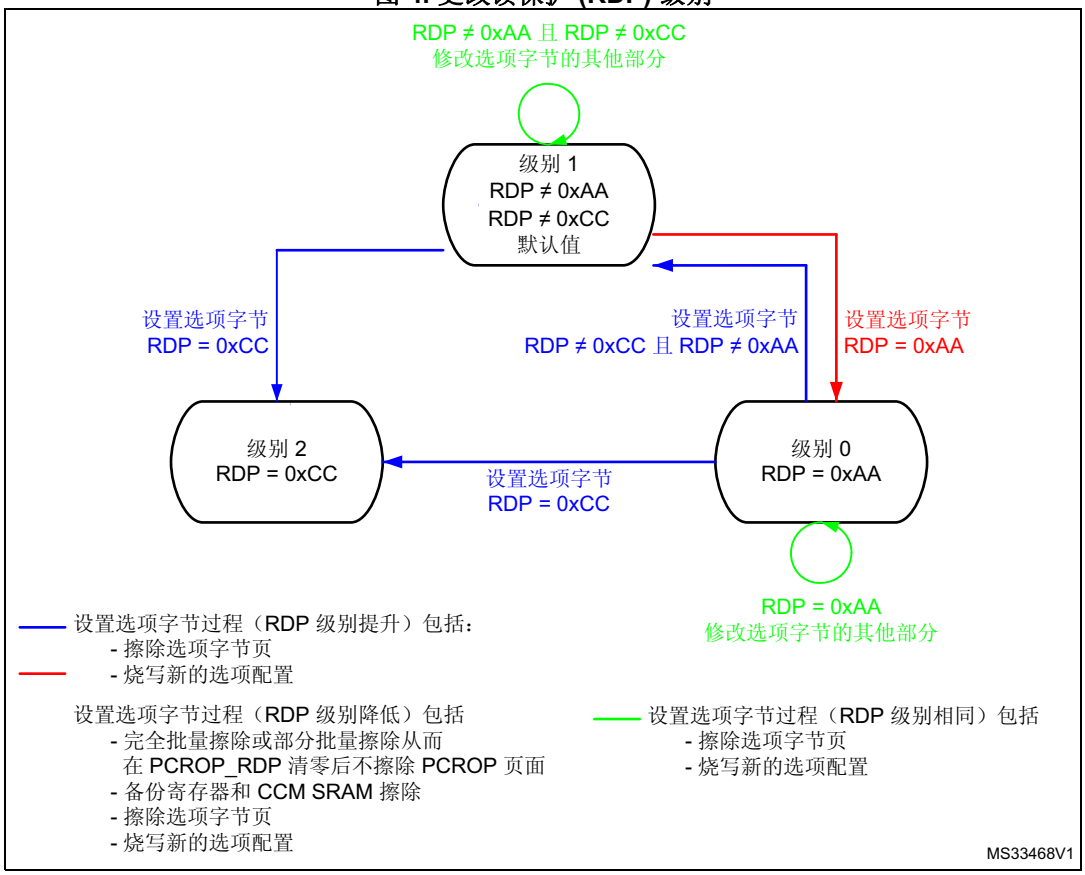


表 14. 访问状态 vs 保护级别和执行模式

区域	保护级别	用户执行（从 Flash 启动）			调试/从 RAM 启动/从加载程序启动		
		读取	写	擦除	读取	写	擦除
Flash 主存储器	1	有	有	有	无	无	无(4)
	2	有	有	有	N/A(1)	N/A(1)	N/A(1)
系统存储器(2)	1	有	无	无	有	无	无
	2	有	无	无	N/A(1)	N/A(1)	N/A(1)
选项字节	1	有	有(3)	有	有	有(3)	有
	2	有	CPU1 和 CPU2 非安全——否	CPU1 和 CPU2 非安全——否	N/A(1)	N/A(1)	N/A(1)
			CPU2 安全——有	CPU2 安全——有			
备份寄存器	1	有	有	N/A	无	无	无(4)
	2	有	有	N/A	N/A(1)	N/A(1)	N/A(1)
SRAM2	1	有	有	N/A	无	无	无(5)
	2	有	有	N/A	N/A(1)	N/A(1)	N/A(1)

1. 保护级别 2 激活后，将禁止调试端口、从 RAM 启动和从系统存储器启动。
2. 无论保护级别（0、1 或 2）和执行模式如何，都只能对系统存储器进行读访问。
3. 在禁止所有级别保护 (0xAA) 的情况下编程 RDP 选项字节时，会擦除 Flash 非安全主存储器。此外，Flash 安全主存储器在以下情况下也将被擦除：SFSA 选项字节编程为非安全，同时 RDP 选项字节编程为禁止所有级别保护 (0xAA)。
4. 当 RDP 从级别 1 更改为级别 0 时擦除备份寄存器。
5. 当 RDP 从级别 1 更改为级别 0 时擦除 SRAM2。

3.6.2 专有代码读保护 (PCROP)

可防止 Flash 的两个部分遭受第三方的读取和写入。

受保护区域为仅执行区域：只能由 STM32 CPU 访问（作为指令代码），所有其他访问（DMA、调试和 CPU 数据读取、写入和擦除）被严格禁止。PCROP 区域的粒度为 2 KB。另有一个选项位 (PCROP_RDP) 可用来选择在 RDP 保护从级别 1 变为级别 0 时 PCROP 区域是否被擦除（参见*更改读保护级别*）。

在 Flash 中，每个 PCROP 区域由起始页偏移和结束页偏移定义。上述偏移在以下 PCROP 地址寄存器中定义：*Flash PCROP 区域 A 起始地址寄存器 (FLASH_PCROP1ASR)*、*Flash PCROP 区域 A 结束地址寄存器 (FLASH_PCROP1AER)*、*Flash PCROP 区域 B 起始地址寄存器 (FLASH_PCROP1BSR)* 和 *Flash PCROP 区域 B 结束地址寄存器 (FLASH_PCROP1BER)*。

PCROP 区域定义为从地址 *Flash 基址 + [PCROPxy_STRT x 0x800]*（包含）到地址 *Flash 基址 + [(PCROPxy_END+1) x 0x800]*（不包含）。最小 PCROP 区域大小为两个 PCROP 页 (4 KB) *PCROPxy_END=PCROPxy_STRT + 1*。

当 *PCROPxy_END = PCROPxy_STRT* 时，整个 Flash 均受 PCROP 保护。

例如，通过从地址 0x0806 2F80（包含）到地址 0x0807 0004（包含）的 PCROP 进行保护：

- 如果选择在 Flash 中启动，则必须对 FLASH_PCROPxySR 或 FLASH_PCROPxyER 寄存器（x = 1，y = A 或 B）进行如下编程：
  - PCROPxy_STRT=0xC5（PCROP 区域第一个地址 0x0806 2800）
  - PCROPxy_END=0xE0（PCROP 区域最后一个地址 0x0807 07FF）

对 PCROP 保护区域执行数据读访问时，将触发 RDERR 标志错误。

任何 PCROP 保护地址同时也受写保护，对其中任一地址执行任何写访问都将触发 WRPERR。

任何 PCROP 区域也受擦除保护。因此，无法对此区域的页进行擦除（包括含有该区域起始地址和结束地址的页）。此外，如果某区域受 PCROP 保护，则无法执行软件批量擦除操作。

在上述示例中，由于采用按页擦除，而从 0xC5 到 0xE0 的所有页在页擦除时均受保护，因此 0x0806 2800 到 0x0807 07FF 的所有地址均无法擦除。

只有当 RDP 从级别 1 切换到级别 0 时，PCROP 才会无效。如果用户选项修改尝试清除 PCROP 或缩小 PCROP 区域，则会启动选项编程，但 PCROP 区域保持不变。相反，可能会增大 PCROP 区域。

选项位 PCROP_RDP 清零，且 RDP 从级别 1 切换到级别 0 时，为保持 PCROP 区域，完全批量擦除由部分批量擦除所代替（参见*更改读保护级别*）。此时，不会擦除 PCROPxy_STRT 和 PCROPxy_END（x = 1，y = A 或 B）。

表 15：PCROP 保护

PCROP 寄存器值（x = 1，y = A 或 B）	PCROP 保护区
PCROPxy_STRT = PCROPxy_END	整个 Flash 均受 PCROP 保护
PCROPxy_STRT > PCROPxy_END	无 PCROPxy，不受保护
PCROPxy_STRT < PCROPxy_END	从 PCROPxy_STRT 到 PCROPxy_END 页均受保护

注： *建议在使用 PCROP_RDP 时将 PCROP 区域与页面大小匹配，或留出 PCROP 区域起始或结束页的剩余部分。*

3.6.3 写保护 (WRP)

可对 Flash 中的用户区域实施保护，以防其遭受意外的写操作。可定义两个写保护 (WRP) 区域，页面大小为 4 KB。每个区域由与物理 Flash 基址相关的起始页偏移和结束页偏移定义。上述偏移在 *Flash WRP 区域 A 地址寄存器 (FLASH_WRP1AR)* 和 *Flash WRP 区域 B 地址寄存器 (FLASH_WRP1BR)* WRP 地址寄存器中定义。

WRP “y” 区域（y=A，B）定义为从地址 *Flash 基址 + [WRP1y_STRT x 0x1000]*（包含）到地址 *Flash 基址 + [(WRP1y_END+1) x 0x1000]*（不包含）。最小 WRP 区域大小为一个 WRP 页 (4 KB)，*WRP1y_END=WRP1y_STRT*。

例如，通过从地址 0x0806 2000（包含）到地址 0x0807 1FFF（包含）的 WRP 进行保护：

- 如果选择在 Flash 中启动，则必须对 FLASH_WRP1AR 寄存器进行如下编程：
  - WRP1A_STRT = 0x62。
  - WRP1A_END = 0x71。

  也可使用 FLASH_WRP1BR 中的 WRP1B_STRT 和 WRP1B_END（Flash 中的区域 “B”）。

WRP 有效时，无法对其执行擦除或编程操作。因此，如果某区域受写保护，则无法执行软件批量擦除操作。

如果尝试对 Flash 的写保护区域执行擦除/编程操作，则 FLASH_SR 寄存器中的写保护错误标志 (WRPERR) 将置 1。对以下区域进行写访问时，此标志也会置 1：

- OTP 区域
- Flash 中永远不能执行写操作的部分（例如 ICP）
- PCROP 区域

注： *选择 Flash 读保护级别（RDP 级别 = 1）后，如果已连接 CPU 调试功能（JTAG 调试或单线调试）或者正在从 RAM 或系统 Flash 执行启动代码，则即使未激活 WRP，也无法对存储器执行编程或擦除操作。尝试执行任何相关操作都会产生硬性故障 (BusFault)。*

表 16：WRP 保护

WRP 寄存器值（x = 1，y = A 或 B）	WRP 保护区
WRPxy_STRT = WRPxy_END	页 WRPxy 受保护
WRPxy_STRT > WRPxy_END	无 WRP，不受保护
WRPxy_STRT < WRPxy_END	从 WRPxy_STRT 到 WRPxy_END 的页均受保护。

注： *为验证 WRP 选项，必须通过 Flash 控制寄存器中的 OBL_LAUNCH 位重载选项字节。*

3.6.4 CPU2 安全 (ESE)

可对 Flash 以及 SRAM2a 和 SRAM2b 存储器的全部或部分区域进行保护，并授予 CPU2 对此区域的独占访问权限，以此防止第三方进行代码执行、读取和写入操作。仅 CPU2 可对这些区域进行代码执行、读取和写入操作。仅限 CPU2 访问，所有其他访问（来自 CPU1 和 DMA）都将被严格禁止。

更改 CPU2 安全模式

可通过安全 CPU2 加载新的用户选项 SFSA，以修改 CPU2 安全起始地址。

CPU2 安全 Flash 区域

在 Flash 中，CPU2 安全 Flash 区域的粒度为一个扇区 (4 KB)，由安全 Flash 起始页偏移用户选项 (SFSA) 定义。此偏移通过*安全 Flash 起始地址寄存器 (FLASH_SFR)* 中的 SFSA 位域控制。

CPU2 安全 Flash 区域定义如下：*Flash 基址 + [SFSA x 0x1000]（包含）*到最后一个 Flash 地址。当使能 CPU2 安全功能时，最小 CPU2 安全区域大小为一个扇区 (4 KB)。

例如，从地址 0x080E 7000（包含）到地址 0x080F FFFF（包含）的 CPU2 安全区域：

- 必须对 FLASH_SFR 寄存器进行如下编程：
  - SFSA=0x0E7。

Flash 选项寄存器 (FLASH_OPTR) 中的标志 ESE 用于指示已使能 CPU2 安全功能。

如果 CPU1 对 CPU2 安全区域进行访问，则会触发 RDERR 或 WRPERR 标志错误。

CPU2 安全 SRAM2 区域

在 Flash 中，CPU2 安全 SRAM2a 和 SRAM2b 区域的粒度为 1 KB，由安全备份 RAM (SRAM2a) 起始地址用户选项（BRSD 和 SBRSA）和安全非备份 RAM (SRAM2b) 起始地址

用户选项（NBRSD 和 SNBRSA）定义。上述偏移通过 *Flash 安全 SRAM2 起始地址和 CPU2 复位向量寄存器 (FLASH_SRRVR)* 中的 SBRSA 和 SNBRSA 位域控制。

CPU2 安全 SRAM2a 区域定义如下：*备份 SRAM2a 基址 + [SBRSA x 0x0400]*（包含）到最后一个 SRAM2a 地址。

例如，从地址 0x080E 5000（包含）到地址 0x2003 7FFF（包含）的 CPU2 安全 SRAM2a 区域：

- 必须对 FLASH_SRRVR 寄存器进行如下编程：
  - SBRSA=0x14。

任何 CPU1 读访问都将返回零数据，而且将丢弃对 CPU2 安全 SRAM2a 区域的写访问并触发总线错误。

当 BRSD 置 1 时，SRAM2a 非安全。

CPU2 安全非备份 SRAM2b 区域定义如下：*非备份 SRAM2b 基址 + [SNBRSA x 0x0400]*（包含）到最后一个 SRAM2b 地址。

例如，从地址 0x2003 EC00（包含）到地址 0x2003 FFFF（包含）的 CPU2 安全 SRAM2b 区域：

- 必须对 FLASH_SRRVR 寄存器进行如下编程：
  - SNBRSA=0x1B。

任何 CPU1 读访问都将返回零数据，而且将忽略对 CPU2 安全 SRAM2b 区域的写访问并触发总线错误。

当 NBRSD 置 1 时，SRAM2b 非安全。

CPU2 调试访问

禁止对 CPU2 进行调试访问，如 *安全 Flash 起始地址寄存器 (FLASH_SFR)* 中的 DDS 位域所指示。调试器无法访问 CPU2 以及安全外设和存储器区域。

3.7 FLASH 编程/擦除暂停

可通过将 *Flash 访问控制寄存器 (FLASH_ACR)* 或 *Flash CPU2 访问控制寄存器 (FLASH_C2ACR)* 中的 PES 位置 1 来暂停 Flash 编程和擦除操作。当 CPU 执行时间关键部分时，此功能非常有用。借助此功能可暂停任何新的编程或擦除操作，防止 CPU 指令和数据获取操作遭到阻止。

至少有一个 PES 位置 1 时：

- 任何正在进行的编程或擦除操作均将完成。
  - Flash 编程/擦除暂停的最大延迟为完成一次编程或擦除操作所用的最长时间（有关 Flash 编程和擦除时序的更多信息，请参见 STM32WB55xx 数据手册）。
- 所有新请求的编程和擦除操作均将暂停，而不会启动。
  - 无论编程/擦除操作当前是否暂停，只要有 PES 置 1，*Flash 状态寄存器 (FLASH_SR)* 和 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 PESD 位就会置 1。这样，CPU 便可先测试 PESD，然后再请求执行编程或擦除操作。

所有 PES 位均复位为 0 时：

- 将启动暂停的编程或擦除操作。
  - *Flash 状态寄存器 (FLASH_SR)* 和 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 PESD 位均将清零。

3.8 Flash 中断

表 17. Flash 中断请求

中断事件	事件标志	事件标志/中断清除方法	中断使能控制位
操作结束	EOP(1)	写入 EOP=1	EOPIE
操作错误	OPERR(2)	写入 OPERR=1	ERRIE
读保护错误	RDERR	写入 RDERR=1	RDERRIE
写保护错误	WRPERR	写入 WRPERR=1	N/A
大小错误	SIZERR	写入 SIZERR=1	N/A
编程顺序错误	PROGERR	写入 PROGERR=1	N/A
编程对齐错误	PGAERR	写入 PGAERR=1	N/A
编程顺序错误	PGSERR	写入 PGSERR=1	N/A
快速编程期间数据丢失错误	MISSERR	写入 MISSERR=1	N/A
快速编程错误	FASTERR	写入 FASTERR=1	N/A
ECC 错误校正	ECCC	写入 ECCC=1	ECCCIE
ECC 双重错误 (NMI)	ECCD	写入 ECCD=1	N/A

1. 仅当 EOPIE 置 1 后，EOP 才会置 1。
2. 仅当 ERRIE 置 1 后，OPERR 才会置 1。

3.9 寄存器访问保护

可通过安全功能保护用户选项寄存器。

FLASH 安全用户选项寄存器（FSD、SFSA、BRSD、SBRSA、NBRSD、SNBRSA、SBRV、C2OPT 和 DDS）为安全寄存器，只能由安全 CPU2 进行写入操作，但允许任何 CPU（安全或非安全）进行读取操作。当 CPU1 尝试进行写入操作时，将被忽略。

3.10 FLASH 寄存器

3.10.1 Flash 访问控制寄存器 (FLASH_ACR)

Flash memory access control register

偏移地址：0x000

复位值：0x0000 0600

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

位 31:17 保留，必须保持复位值。

位 16 **EMPTY**：Flash 用户数据为空 (Flash memory User area empty)
读取时，该位指示用户 Flash 的第一个存储单元是已擦除还是具有编程值。
0：读取：用户 Flash 已编程
1：读取：用户 Flash 为空
写入时，该位将被写入的值覆盖

位 15 **PES**：CPU1 编程/擦除暂停请求 (CPU1 program / erase suspend request)
0：允许进行 Flash 编程和擦除操作。
1：任何新的 Flash 编程和擦除操作都将暂停，直到此位和 *Flash CPU2 访问控制寄存器 (FLASH_C2ACR)* 中的相同位清零。FLASH_ACR 或 FLASH_C2ACR 中至少有一个 PES 位置 1 时，*Flash 状态寄存器 (FLASH_SR)* 和 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 PESD 位将置 1。

位 14:13 保留，必须保持清零

位 12 **DCRST**：CPU1 数据缓存复位 (CPU1 Data cache reset)
0：CPU1 数据缓存不复位
1：CPU1 数据缓存复位
只有在禁止数据缓存后才能写入该位。

位 11 **ICRST**：CPU1 指令缓存复位 (CPU1 Instruction cache reset)
0：CPU1 指令缓存不复位
1：CPU1 指令缓存复位
只有在禁止指令缓存后才能写入该位。

位 10 **DCEN**：CPU1 数据缓存使能 (CPU1 Data cache enable)
0：禁止 CPU1 数据缓存
1：使能 CPU1 数据缓存

位 9 **ICEN**：CPU1 指令缓存使能 (CPU1 Instruction cache enable)
0：禁止 CPU1 指令缓存
1：使能 CPU1 指令缓存

位 8 **PRFTEN**：CPU1 预取使能 (CPU1 Prefetch enable)
0：禁止 CPU1 预取
1：使能 CPU1 预取

位 7:3 保留，必须保持复位值。

位 2:0 **LATENCY[2:0]**：延迟 (Latency)
这些位表示 Flash HCLK 时钟周期与 Flash 访问时间之比。
000：0 个等待状态
001：1 个等待状态
010：2 个等待状态
011：3 个等待状态
其他值：保留

3.10.2 Flash 密钥寄存器 (FLASH_KEYR)

Flash memory key register

偏移地址：0x008

复位值：0x0000 0000

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

位 31:0 **KEY**：Flash 密钥 (Flash memory key)
要将 *Flash 控制寄存器 (FLASH_CR)* 和 *Flash CPU2 控制寄存器 (FLASH_C2CR)* 解锁以允许执行编程/擦除操作，必须顺序写入以下值：
KEY1：0x4567 0123
KEY2：0xCDEF 89AB

3.10.3 Flash 选项密钥寄存器 (FLASH_OPTKEYR)

Flash memory option key register

偏移地址：0x00C

复位值：0x0000 0000

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

位 31:0 **OPTKEYR**：选项字节密钥 (Option byte key)

要将 Flash 选项寄存器解锁以允许执行选项字节编程/擦除操作，必须顺序写入以下值：

KEY1：0x0819 2A3B

KEY2：0x4C5D 6E7F

3.10.4 Flash 状态寄存器 (FLASH_SR)

Flash memory status register

偏移地址：0x010

复位值：0x000X 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PESD	CFGBSY	Res.	BSY
												r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTV ERR	RD ERR	OPTNV	Res.	Res.	Res.	FAST ERR	MISS ERR	PGS ERR	SIZ ERR	PGA ERR	WRP ERR	PROG ERR	Res.	OP ERR	EOP
rc_w1	rc_w1	r				rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1

位 31:20 保留，必须保持复位值。

位 19 **PESD**：暂停编程/擦除操作 (Programming / erase operation suspended)

此位由硬件置 1 和复位。

Flash 访问控制寄存器 (FLASH_ACR) 或 *Flash CPU2 访问控制寄存器 (FLASH_C2ACR)* 中至少有一个 PES 位置 1 时，该位置 1。

FLASH_ACR 和 FLASH_C2ACR 中的 PES 位均清零时，该位清零。

置 1 时，不会启动新的编程或擦除操作。

位 18 **CFGBSY**：编程或擦除配置繁忙 (Programming or erase configuration busy)

此位由硬件置 1 和复位。（在发送第一个字时置 1，在编程操作完成或被错误中断时复位。）

置 1 时，*Flash 控制寄存器 (FLASH_CR)* 请求的 FLASH_PG 和 FLASH_PNB 中的编程和擦除设置将被使用（繁忙），不能更改（正在进行编程或擦除操作）。

复位为 0 时，可修改 *Flash 控制寄存器 (FLASH_CR)* 中 FLASH_PG 和 FLASH_PNB 的编程和擦除设置。

位 17 保留，必须保持复位值。

位 16 **BSY**：繁忙 (Busy)

指示 *Flash 控制寄存器 (FLASH_CR)* 请求的 Flash 操作正在进行。该位在 Flash 操作开始时置 1，在操作结束或出现错误时复位。

位 15 **OPTVERR**：选项和工程位加载有效性错误 (Option and Engineering bits loading validity error)

读取的选项和工程位可能不是用户配置或生产过程中配置的位时，该位由硬件置 1。如果选项和工程位尚未正确加载，则在每次系统复位后，OPTVERR 将再次置 1。加载失败的选项字节将被强制设为安全值，请参见*第 3.4.2 节：选项字节编程*。

写入 1 即可将该位清零。

位 14 **RDERR**：PCROP 读取错误 (PCROP read error)

通过 D 总线读取的地址属于 Flash 的读保护区域（PCROP 保护）时，此位由硬件置 1。如果 FLASH_CR 中的 RDERRIE 置 1，将生成中断。

写入 1 即可将该位清零。

位 13 **OPTNV**：用户选项 OPTVAL 指示 (User Option OPTVAL indication)

此位由硬件置 1 和复位。

0：OBL 用户选项 OPTVAL 指示“有效”。

1：OBL 用户选项 OPTVAL 指示“无效”。

位 12:10 保留，必须保持复位值。

位 9 **FASTERR**：快速编程错误 (Fast programming error)

因错误（对齐、大小、写保护或数据丢失）导致快速编程顺序（由 FSTPG 激活）中断时，此位由硬件置 1。同时，相应状态位（PGAERR、SIZERR、WRPERR 或 MISSERR）也会置 1。

写入 1 即可将该位清零。

位 8 **MISERR**：快速编程数据丢失错误 (Fast programming data miss error)

在快速编程模式下，必须将 64 个双字（512 字节）连续发送到 Flash，并且必须在当前数据完全编程之前将新数据发送到逻辑控制。如果新数据未及时出现，则 MISSERR 由硬件置 1。

写入 1 即可将该位清零。

位 7 **PGSERR**：编程顺序错误 (Programming sequence error)

如果代码在 PG 或 FSTPG 先前未置 1 的情况下对 Flash 执行写访问，则该位由硬件置 1。由于之前的编程错误而导致 PROGERR、SIZERR、PGAERR、WRPERR、MISSERR 或 FASTERR 置 1 时，该位也由硬件置 1。

写入 1 即可将该位清零。

位 6 **SIZERR**：大小错误 (Size error)

在编程或快速编程顺序中，访问大小为字节或半字时，该位由硬件置 1。只允许双字编程（即按字访问）。

写入 1 即可将该位清零。

位 5 **PGAERR**：编程对齐错误 (Programming alignment error)

如果在标准编程期间要编程的数据无法包含在同一双字（64 位）Flash 中，或快速编程时存在页面更改，该位将由硬件置 1。

写入 1 即可将该位清零。

位 4 **WRPERR**：写保护错误 (Write protection error)

如果要擦除/编程的地址属于 Flash 中受写保护（受 WRP、PCROP 或 RDP 等级 1 保护）的区域，则该位由硬件置 1。

写入 1 即可将该位清零。

位 3 **PROGERR**：编程错误 (Programming error)

编程前，要编程的双字地址包含不同于“0xFFFF FFFF FFFF FFFF”的值时，该位由硬件置 1（要写入的数据为 0x0000 0000 0000 0000 时除外）。

写入 1 即可将该位清零。

位 2 保留，必须保持复位值。

位 1 **OPERR**：操作错误 (Operation error)

当 Flash 操作（编程/擦除）失败时，该位由硬件置 1。

只有在使能错误中断 (ERRIE=1) 后，该位才会置 1。

写入“1”即可将该位清零。

位 0 **EOP**：操作结束 (End of operation)

当成功完成一个或多个 Flash 操作（编程/擦除）时，该位由硬件置 1。

只有在使能操作结束中断 (EOPIE=1) 后，该位才会置 1。

写入 1 即可将该位清零。

3.10.5 Flash 控制寄存器 (FLASH_CR)

Flash memory control register

偏移地址：0x014

复位值：0xC000 0000

访问：当前未执行任何 Flash 操作时无等待状态，按字、半字和字节访问

Flash 状态寄存器 (FLASH_SR) 中的 CFGBSY 置 1 时，无法修改该寄存器。

- *Flash 状态寄存器 (FLASH_SR)* 中的 PESD 位清零时，将暂停寄存器写访问，直至 CFGBSY 位清零（由其他 CPU）。
- 如果在 *Flash 状态寄存器 (FLASH_SR)* 中的 PESD 位置 1 时正在进行编程或擦除操作，则寄存器写访问会导致总线错误。
- 如果在 *Flash 状态寄存器 (FLASH_SR)* 中的 PESD 位置 1 时未在进行编程或擦除操作，则会完成寄存器写访问。请求的编程或擦除操作将暂停，BSY/CFGBSY 将置 1 并保持为 1，直到暂停功能失效为止。之后，PESD 位会恢复为 0 并会完成之前暂停的操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OPT LOCK	Res.	Res.	OBL_ LAUNCH	RD ERRIE	ERRIE	EOPIE	Res.	Res.	Res.	Res.	Res.	FSTPG	OPT STRT	STRT
rs	rs			rc_w1	rw	rw	rw						rw	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	PNB[7:0]								MER	PER	PG
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **LOCK**：FLASH_CR 锁定 (FLASH_CR Lock)

该位只置 1。置 1 后，FLASH_CR 寄存器被锁定。当检测到解锁序列时，由硬件将该位清零。如果解锁操作失败，该位仍保持置 1，直到下一次系统复位。

位 30 **OPTLOCK**：选项锁定 (Options Lock)

该位只置 1。置 1 后，FLASH_CR 寄存器中所有与用户选项相关的位以及选项页都被锁定。当检测到解锁序列时，由硬件将该位清零。对 OPTLOCK 位执行解锁序列之前，必须将 LOCK 位清零。

如果解锁操作失败，该位仍保持置 1，直到下一次复位。

位 29:28 保留，必须保持复位值。

位 27 **OBL_LAUNCH**：强制选项字节加载 (Forces the option byte loading)

此位置 1 时，会强制进行选项字节重载。只有在选项字节加载完成时此位才会清零。如果 OPTLOCK 置 1，则此位无法写入。

0：选项字节加载完成
1：已请求选项字节加载

位 26 **RDERRIE**：PCROP 读取错误中断使能 (PCROP read error interrupt enable)

当 FLASH_SR 中的 RDERR 位置 1 后，可通过该位使能中断产生功能。

0：禁止 PCROP 读取错误中断
1：使能 PCROP 读取错误中断

位 25 **ERRIE**：错误中断使能 (Error interrupt enable)

当 FLASH_SR 中的 OPERR 位置 1 后，可通过该位使能中断产生功能。

0：禁止 OPERR 错误中断
1：使能 OPERR 错误中断

位 24 **EOPIE**：操作结束中断使能 (End of operation interrupt enable)
当 FLASH_SR 中的 EOP 位置 1 后，可通过该位使能中断产生功能。
0：禁止 EOP 中断
1：使能 EOP 中断

位 23:19 保留，必须保持复位值

位 18 **FSTPG**：快速编程 (Fast programming)
0：禁止快速编程
1：使能快速编程

位 17 **OPTSTRT**：启动选项修改 (Options modification start)
该位置 1 后可触发选项操作。
该位只能通过软件置 1，并在 FLASH_SR 中的 BSY 位清零后随之清零。

位 16 **STRT**：启动 (Start)
该位置 1 后可触发擦除操作。如果 MER 和 PER 位均复位并且 STRT 位置 1，可能会发生无法预知的行为而不会生成任何错误标志。应禁止此情况发生。
该位只能通过软件置 1，并在 FLASH_SR 中的 BSY 位清零后随之清零。
当 CPU1 请求操作（涉及安全 Flash 页）时，将会拒绝启动并产生总线错误。

位 15:11 保留，必须保持复位值。

位 10:3 **PNB[7:0]**：页编号选择 (Page number selection)
这些位用于选择要擦除的页：
0x00： 页 0
0x01： 页 1
...
0xFF： 页 255

位 2 **MER**：批量擦除 (Mass Erase)
该位置 1 时，会触发批量擦除（所有用户页）。

位 1 **PER**：页擦除 (Page erase)
0：禁止页擦除
1：使能页擦除

位 0 **PG**：编程 (Programming)
0：禁止 Flash 编程
1：使能 Flash 编程

3.10.6 Flash ECC 寄存器 (FLASH_ECCR)

Flash memory ECC register

偏移地址：0x018

复位值：0x0000 0000

访问：当前未执行任何 Flash 操作时无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCD	ECCC	Res.	CPUID[2:0]			Res.	ECCCIE	Res.	Res.	Res.	SYSF_ECC	Res.	Res.	Res.	ADDR_ECC[16]
rc_w1	rc_w1		r	r	r		rw				r				r

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

位 31 **ECCD**：ECC 检测 (ECC detection)
检测到两个 ECC 错误时，该位由硬件置 1。该位置 1 时，会生成 NMI。
写入 1 即可将该位清零。

位 30 **ECCC**：ECC 校正 (ECC correction)
检测到一个 ECC 错误并对其进行校正时，该位由硬件置 1。如果 ECCIE 置 1，则会生成中断。
写入 1 即可将该位清零。

位 29 保留，必须保持复位值。

位28:26 **CPUID**：CPU 标识 (CPU identification)
由硬件置 1，指示导致 ECC 故障的 CPU 访问的总线 ID。

位 25 保留，必须保持复位值。

位 24 **ECCIE**：ECC 校正中断使能 (ECC correction interrupt enable)
0：禁止 ECCC 中断
1：使能 ECCC 中断

位 23:21 保留，必须保持复位值。

位 20 **SYSF_ECC**：系统 Flash ECC 失效 (System Flash memory ECC fail)
该位指示位于系统 Flash 中的 ECC 错误校正或双重 ECC 错误检测。

位 19:17 保留，必须保持复位值。

位 16:0 **ADDR_ECC**：ECC 失效双字地址 (ECC fail double-word address)
该位指示 ECC 错误校正涉及的双字地址或导致双重 ECC 错误检测的双字地址。

3.10.7 Flash 选项寄存器 (FLASH_OPTR)

Flash memory option register

偏移地址：0x020

复位值：bXXX1 XXXX X111 XXXX 1XXX XXX1 XXXX XXXX。复位信号释放时，硬件将 Flash 中的值加载到这些选项位。

访问：当前未执行任何 Flash 操作时无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AGC_TRIM[2:0]			Res.	n BOOT0	nSW BOOT0	SRAM2 _RST	SRAM2 _PE	n BOOT1	Res.	Res.	Res.	WWDG _SW	IWGD _STDBY	IWDG _STOP	IWDG _SW
rw	rw	rw		rw	rw	rw	rw	rw				rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	nRST _SHDW	nRST _STDBY	nRST _STOP	BOR_LEV[2:0]			ESE	RDP[7:0]							
	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 **AGC_TRIM**：无线电自动增益控制微调 (Radio automatic gain control trimming)
默认值为 0b001

位 28 保留，必须保持复位值。

位 27 **nBOOT0**：nBOOT0 选项位 (nBOOT0 option bit)

如果 nSWBOOT0 位配置选择 BOOT0 取自位 nBOOT0，则该位将与位 nBOOT1 搭配用于选择从用户 Flash、SRAM1 或系统 Flash 启动。请参见*第 2.3 节：启动配置*。

0：nBOOT0=0
1：nBOOT0=1

位 26 **nSWBOOT0**：软件 BOOT0 选择 (Software BOOT0 selection)

0：BOOT0 取自选项位 nBOOT0
1：BOOT0 取自 PH3/BOOT0 引脚

位 25 **SRAM2_RST**：发生系统复位时擦除 SRAM2 和 PKA RAM (SRAM2 and PKA RAM Erase when system reset)

0：发生系统复位时擦除 SRAM2 和 PKA RAM
1：发生系统复位时不擦除 SRAM2 和非安全 PKA RAM

位 24 **SRAM2_PE**：SRAM2 奇偶校验使能 (SRAM2 parity check enable)

0：使能 SRAM2 奇偶校验
1：禁止 SRAM2 奇偶校验

位 23 **nBOOT1**：启动配置 (Boot configuration)

该位与 BOOT0 引脚或选项位 nBOOT0（具体取决于 nSWBOOT0 选项位配置）搭配使用，用于选择从用户 Flash、SRAM1 或系统存储器启动模式。请参见*第 2.3 节：启动配置*。

位 22:20 保留，必须保持复位值

位 19 **WWDG_SW**：窗口看门狗选择 (Window watchdog selection)

0：硬件窗口看门狗
1：软件窗口看门狗

位 18 **IWDG_STDBY**：在待机模式下冻结独立看门狗计数器 (Independent watchdog counter freeze in Standby mode)

0：在待机模式下冻结独立看门狗计数器
1：在待机模式下运行独立看门狗计数器

位 17 **IWDG_STOP**：在停止模式下冻结独立看门狗计数器 (Independent watchdog counter freeze in stop mode)

0：在停止模式下冻结独立看门狗计数器
1：在停止模式下运行独立看门狗计数器

位 16 **IDWG_SW**：独立看门狗选择 (Independent watchdog selection)

0：硬件独立看门狗
1：软件独立看门狗

位 15 保留，必须保持清零

位 14 **nRST_SHDW**

0：进入关断模式时产生复位
1：进入关断模式时不产生复位

位 13 **nRST_STDBY**

0：进入待机模式时产生复位
1：进入停机模式时不产生复位

位 12 **nRST_STOP**

0：进入停机模式时产生复位
1：进入停机模式时不产生复位

位 11:9 **BOR_LEV**：BOR 复位级别 (BOR reset Level)
这些位包含激活/释放复位信号所需达到的 VDD 供电电压阈值。
000：BOR 0 级。复位阈值电压约 1.7 V
001：BOR 1 级。复位阈值电压约 2.0 V
010：BOR 2 级。复位阈值电压约 2.2 V
011：BOR 3 级。复位阈值电压约 2.5 V
100：BOR 4 级。复位阈值电压约 2.8 V

位 8 **ESE**：系统安全使能标志 (System security enabled flag)
指示是否使能系统安全功能（用户 Flash FSD=0）。
0：禁止安全功能。
1：使能安全功能。

位 7:0 **RDP**：读保护级别 (Read protection level)
0xAA：级别 0，未激活读保护
0xCC：级别 2，激活芯片读保护
其他：级别 1，激活存储器读保护
注：　请注意级别 1 下的 PCROP_RDP 配置。更多详细信息，请参见级别 1：读保护。

3.10.8 Flash PCROP 区域 A 起始地址寄存器 (FLASH_PCROP1ASR)

Flash memory PCROP zone A Start address register

偏移地址：0x024

复位值：b1111 1111 1111 1111 1111 111X XXXX XXXX

访问：当前未执行任何 Flash 操作时无等待状态；按字、半字访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1A_STRT[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:9 保留，必须保持清零

位 8:0 **PCROP1A_STRT**：PCROP1A 区域起始偏移 (PCROP1A area start offset)
PCROP1A_STRT 包含 PCROP1A 区域的第一个 2 KB 页。

3.10.9 Flash PCROP 区域 A 结束地址寄存器 (FLASH_PCROP1AER)

Flash memory PCROP zone A End address register

偏移地址：0x028

复位值：bX111 1111 1111 1111 1111 111X XXXX XXXX

访问：当前未执行任何 Flash 操作时无等待状态；按字、半字访问。可通过字节访问的形式访问 PCROP_RDP 位

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PCROP_RDP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1A_END[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **PCROP_RDP**：RDP 级别下降时，PCROP 区域保持 (PCROP area preserved when RDP level decreased)

该位只置 1。执行由 RDP 从级别 1 更改到级别 0 导致的完全批量擦除后，该位复位。

0：RDP 级别从级别 1 降低到级别 0 时，不擦除 PCROP 区域。

1：RDP 级别从级别 1 降低到级别 0 时，擦除 PCROP 区域（完全批量擦除）。

位 30:9 保留，必须保持清零

位 8:0 **PCROP1A_END**：PCROP1A 区域结束偏移 (PCROP1A area end offset)

PCROP1A_END 包含 PCROP1A 区域的最后一个 2 KB 页。

3.10.10 Flash WRP 区域 A 地址寄存器 (FLASH_WRP1AR)

Flash memory WRP area A address register

偏移地址：0x02C

复位值：0xFFXX FFXX

访问：当前未执行任何 Flash 操作时无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_END[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_STRT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持清零

位 23:16 **WRP1A_END**：WRP 第一个区域“A”结束偏移 (WRP first area “A” end offset)

包含 WRP 第一个区域的最后一个 4 KB 页。

位 15:8 保留，必须保持清零

位 7:0 **WRP1A_STRT**：WRP 第一个区域“A”起始偏移 (WRP first area “A” start offset)

包含 WRP 第一个 区域的第一个 4 KB 页。

3.10.11 Flash WRP 区域 B 地址寄存器 (FLASH_WRP1BR)

Flash memory WRP area B address register

偏移地址：0x030

复位值：0xFFXX FFXX

访问：当前未执行任何 Flash 操作时无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_END[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_STRT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持清零。

位 23:16 **WRP1B_END**：WRP 第二个区域“B”结束偏移 (WRP second area “B” end offset)
WRPB1_END 包含 WRP 第二个区域的最后一个 4 KB 页。

位 15:8 保留，必须保持清零。

位 7:0 **WRP1B_STRT**：WRP 第二个区域“B”起始偏移 (WRP second area “B” start offset)
WRPB1_END 包含 WRP 第二个区域的第一个 4 KB 页。

3.10.12 Flash PCROP 区域 B 起始地址寄存器 (FLASH_PCROP1BSR)

Flash memory PCROP zone B Start address register

偏移地址：0x034

复位值：b1111 1111 1111 1111 1111 111X XXXX XXXX

访问：当前未执行任何 Flash 操作时无等待状态；按字、半字访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1B_STRT[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:9 保留，必须保持清零

位 8:0 **PCROP1B_STRT**：PCROP1B 区域起始偏移 (PCROP1B area start offset)
包含 PCROP1B 区域的第一个 2 KB 页。

3.10.13 Flash PCROP 区域 B 结束地址寄存器 (FLASH_PCROP1BER)

Flash memory PCROP zone B End address register

偏移地址：0x038

复位值：b1111 1111 1111 1111 1111 111X XXXX XXXX

访问：当前未执行任何 Flash 操作时无等待状态；按字、半字访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1B_END[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:9 保留，必须保持清零

位 8:0 **PCROP1B_END**：PCROP1B 区域结束偏移 (PCROP1B area end offset)
包含 PCROP1B 区域的最后一个 2 KB 页。

3.10.14 Flash IPCC 邮箱数据缓冲区地址寄存器 (FLASH_IPCCBR)

Flash memory IPCC mailbox data buffer address register

偏移地址：0x03C

复位值：b1111 1111 1111 1111 11XX XXXX XXXX XXXX

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

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

位 31:14 保留，必须保持清零

位 13:0 **IPCCDBA**：IPCC 邮箱数据缓冲区基址偏移 (IPCC mailbox data buffer base address offset)
包含 SRAM2 中 IPCC 邮箱数据缓冲区的第一个双字偏移。

3.10.15 安全 Flash 起始地址寄存器 (FLASH_SFR)

Secure Flash memory start address register

偏移地址：0x080

复位值：b1111 1111 1111 1111 1111 1110 XXXX XXXX

该寄存器提供写访问安全功能，仅允许 CPU2 进行写入操作。来自 CPU1 的写访问将被忽略，并会产生总线错误。任何读访问都会返回寄存器值。

写入的值仅在 OBL 之后才会采用。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DDS	Res.	Res.	Res.	FSD	SFSA[7:0]							
			rw				rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:13 保留，必须保持清零

位 12 **DDS**：禁止 CPU2 调试访问 (Disable CPU2 Debug access)
0：使能 CPU2 调试访问
1：禁止 CPU2 调试访问

位 11:9 保留，必须保持清零

位 8 **FSD**：禁止 Flash 安全功能 (Flash memory security disabled)
1：系统和 Flash 非安全
0：系统和 Flash 安全（Flash 的安全区域由 SFSA 指定）

位 7:0 **SFSA**：安全 Flash 起始地址 (Secure Flash memory start address)
SFSA[7:0] 包含安全 Flash 区域的第一个 4 KB 页的起始地址。

3.10.16 Flash 安全 SRAM2 起始地址和 CPU2 复位向量寄存器 (FLASH_SRRVR)

Flash memory secure SRAM2 start address and CPU2 reset vector register

偏移地址：0x084

复位值：bXXXX XXX1 XXXX XXXX XXXX XXXX XXXX XXXX

该寄存器提供写访问安全功能，仅允许 CPU2 进行写入操作。来自 CPU1 的写访问将被忽略，并会产生总线错误。任何读访问都会返回寄存器值。

写入的值仅在 OBL 之后才会采用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C2OPT	NBRSD	SNBRSA[4:0]					Res.	BRSD	SBRSA[4:0]					SBRV[17:16]	
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
SBRV[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **C2OPT**：CPU2 启动复位向量存储器选择 (CPU2 boot reset vector memory selection)
0：SBRV 偏移将以 0x2000 0000 为起始地址对 SRAM1 或 SRAM2 进行寻址。SBRV 值将保持在 SRAM 区域内。
1：SBRV 偏移将以 0x0800 0000 为起始地址对 Flash 进行寻址。

位 30 **NBRSD**：非备份 SRAM2b 安全禁止 (non-backup SRAM2b security disable)
NBRSD=1：SRAM2b 非安全。
NBRSD=0：SRAM2b 安全。SNBRSA[4:0] 包含安全非备份 SRAM2b 区域的第一个 1K 页的起始地址。

位 29:25 **SNBRSA**：安全非备份 SRAM2b 起始地址 (Secure non-backup SRAM2b start address)
NBRSD=0：SRAM2b 安全。SNBRSA[4:0] 包含安全非备份 SRAM2b 区域的第一个 1K 页的起始地址。

位 24 保留，必须保持清零

位 23 **BRSD**：备份 SRAM2a 安全禁止 (backup SRAM2a security disable)
BRSD=1：SRAM2a 非安全。
BRSD=0：SRAM2a 安全。SBRSA[4:0] 包含安全备份 SRAM2a 区域的第一个 1K 页的起始地址。

位 22:18 **SBRSA**：安全备份 SRAM2a 起始地址 (Secure backup SRAM2a start address)
BRSD=0：SRAM2a 安全。SBRSA[4:0] 包含安全备份 SRAM2a 区域的第一个 1K 页的起始地址。

位 17:0 **SBRV**：CPU2 启动复位向量 (CPU2 boot reset vector)
包含 C2OPT 所选择的存储器区域内按字对齐的 CPU2 启动复位起始地址偏移。

3.10.17 Flash CPU2 访问控制寄存器 (FLASH_C2ACR)

Flash memory CPU2 access control register

偏移地址：0x05C

复位值：0x0000 0600

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

位 31:16 保留，必须保持清零

位 15 **PES**：CPU2 编程/擦除暂停请求 (CPU2 program / erase suspend request)
0：允许进行 Flash 编程和擦除操作。
1：任何新的 Flash 编程和擦除操作都将暂停，直到此位和 *Flash 访问控制寄存器 (FLASH_ACR)* 中的相同位清零。FLASH_ACR 或 FLASH_C2ACR 中至少有一个 PES 位置 1 时，*Flash 状态寄存器 (FLASH_SR)* 和 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 PESD 位将置 1。

位 14:12 保留，必须保持清零

位 11 **ICRST**：CPU2 指令缓存复位 (CPU2 Instruction cache reset)
0：CPU2 指令缓存不复位
1：CPU2 指令缓存复位
只有在禁止指令缓存后才能写入该位。

位 10 保留，必须保持清零

位 9 **ICEN**：CPU2 指令缓存使能 (CPU2 Instruction cache enable)
0：禁止 CPU2 指令缓存
1：使能 CPU2 指令缓存

位 8 **PRFEN**：CPU2 预取使能 (CPU2 Prefetch enable)
0：禁止 CPU2 预取操作
1：使能 CPU2 预取操作

位 7:0 保留，必须保持清零

3.10.18 Flash CPU2 状态寄存器 (FLASH_C2SR)

Flash memory CPU2 status register

偏移地址：0x060

复位值：0x000X 0000

访问：无等待状态，按字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PESD	CFGBSY	Res.	BSY
												r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RD ERR	Res.	Res.	Res.	Res.	FAST ERR	MISS ERR	PGS ERR	SIZ ERR	PGA ERR	WRP ERR	PROG ERR	Res.	OP ERR	EOP
	rc_w1					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1

位 31:20 保留，必须保持复位值。

位 19 **PESD**：暂停编程/擦除操作 (Programming / erase operation suspended)

此位由硬件置 1 和复位。

Flash 访问控制寄存器 (FLASH_ACR) 或 *Flash CPU2 访问控制寄存器 (FLASH_C2ACR)* 中至少有一个 PES 位置 1 时，该位置 1。

FLASH_ACR 和 FLASH_C2ACR 中的 PES 位均清零时，该位清零。

置 1 时，不会启动新的编程或擦除操作。

位 18 **CFGBSY**：编程或擦除配置繁忙 (Programming or erase configuration busy)

此位由硬件置 1 和复位。（在发送第一个字时置 1，在编程操作完成或被错误中断时复位。）

置 1 时，编程和擦除设置（位于 *Flash CPU2 控制寄存器 (FLASH_C2CR)* 的 FLASH_PG 和 FLASH_PNB 中）将被使用（繁忙），不能更改。此时正在进行 Flash 编程或擦除操作。

复位为 0 时，可修改编程和擦除设置（位于 *Flash CPU2 控制寄存器 (FLASH_C2CR)* 的 FLASH_PG 和 FLASH_PNB 中）。

位 17 保留，必须保持复位值。

位 16 **BSY**：繁忙 (Busy)

该位指示 *Flash CPU2 控制寄存器 (FLASH_C2CR)* 请求的 Flash 操作正在进行。该位在 Flash 操作开始时置 1，在操作结束或出现错误时复位。

位 15 保留，必须保持复位值。

位 14 **RDERR**：PCROP 读取错误 (PCROP read error)

通过 D 总线读取的地址属于 Flash 的读保护区域（PCROP 保护）时，此位由硬件置 1。如果 FLASH_CR 中的 RDERRIE 置 1，将生成中断。

写入 1 即可将该位清零。

位 13:10 保留，必须保持复位值。

位 9 **FASTERR**：快速编程错误 (Fast programming error)

因错误（对齐、大小、写保护或数据丢失）导致快速编程顺序（由 FSTPG 激活）中断时，此位由硬件置 1。同时，相应状态位（PGAERR、SIZERR、WRPERR 或 MISSERR）也会置 1。

写入 1 即可将该位清零。

位 8 **MISERR**：快速编程数据丢失错误 (Fast programming data miss error)

在快速编程模式下，必须将 64 个双字（512 字节）连续发送到 Flash，并且必须在当前数据完全编程之前将新数据发送到 Flash 逻辑控制。如果新数据未及时出现，则 MISSERR 由硬件置 1。

写入 1 即可将该位清零。

位 7 **PGSERR**：编程顺序错误 (Programming sequence error)

如果代码在 PG 或 FSTPG 先前未置 1 的情况下对 Flash 执行写访问，则该位由硬件置 1。由于之前的编程错误而导致 PROGERR、SIZERR、PGAERR、WRPERR、MISSERR 或 FASTERR 置 1 时，该位也由硬件置 1。

写入 1 即可将该位清零。

位 6 **SIZERR**：大小错误 (Size error)

在编程或快速编程顺序中，访问大小为字节或半字时，该位由硬件置 1。只允许双字编程（即按字访问）。

写入 1 即可将该位清零。

位 5 **PGAERR**：编程对齐错误 (Programming alignment error)

如果在标准编程期间要编程的数据无法包含在同一双字（64 位）Flash 中，或快速编程时存在页面更改，该位将由硬件置 1。

写入 1 即可将该位清零。

位 4 **WRPERR**：写保护错误 (Write protection error)

如果要擦除/编程的地址属于 Flash 中受写保护（受 WRP、PCROP 或 RDP 等级 1 保护）的区域，则该位由硬件置 1。

写入 1 即可将该位清零。

位 3 **PROGERR**：编程错误 (Programming error)

编程前，要编程的双字地址包含不同于“0xFFFF FFFF”的值时，该位由硬件置 1（要写入的数据为 0x0000 0000 0000 0000 时除外）。

写入 1 即可将该位清零。

位 2 保留，必须保持复位值。

位 1 **OPERR**：操作错误 (Operation error)

当 Flash 操作（编程/擦除）失败时，该位由硬件置 1。

只有在使能错误中断 (ERRIE=1) 后，该位才会置 1。

写入“1”即可将该位清零。

位 0 **EOP**：操作结束 (End of operation)

当成功完成一个或多个 Flash 操作（编程/擦除）时，该位由硬件置 1。

只有在使能操作结束中断 (EOPIE=1) 后，该位才会置 1。

写入 1 即可将该位清零。

3.10.19 Flash CPU2 控制寄存器 (FLASH_C2CR)

Flash memory CPU2 control register

偏移地址：0x064

复位值：0xC000 0000

访问：当前未执行任何 Flash 操作时无等待状态，按字、半字和字节访问

Flash CPU2 状态寄存器 (FLASH_C2SR) 中的 CFGBSY 置 1 时，无法修改该寄存器。

- *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 PESD 位清零时，将暂停寄存器写访问，直至 CFGBSY 位清零（由其他 CPU）。
- *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 PESD 位置 1 时，寄存器写访问会导致总线错误。
- 如果在 *Flash CPU2 状态寄存器 (FLASH_C2SR)* 中的 PESD 位置 1 时未在进行编程或擦除操作，则会完成寄存器写访问。请求的编程或擦除操作将暂停，BSY/CFGBSY 将置 1 并保持为 1，直到暂停功能失效为止。之后，PESD 位会恢复为 0 并会完成之前暂停的操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	RD ERRIE	ERR IE	EOP IE	Res.	Res.	Res.	Res.	Res.	FSTPG	Res.	STRT
					rw	rw	rw						rw		rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	PNB[7:0]								MER	PER	PG
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:27 保留，必须保持复位值。

位 26 **RDERRIE**：PCROP 读取错误中断使能 (PCROP read error interrupt enable)
当 FLASH_SR 中的 RDERR 位置 1 后，可通过该位使能中断产生功能。
0：禁止 PCROP 读取错误中断
1：使能 PCROP 读取错误中断

位 25 **ERRIE**：错误中断使能 (Error interrupt enable)
当 FLASH_SR 中的 OPERR 位置 1 后，可通过该位使能中断产生功能。
0：禁止 OPERR 错误中断
1：使能 OPERR 错误中断

位 24 **EOPIE**：操作结束中断使能 (End of operation interrupt enable)
当 FLASH_SR 中的 EOP 位置 1 后，可通过该位使能中断产生功能。
0：禁止 EOP 中断
1：使能 EOP 中断

位 23:19 保留，必须保持复位值

位 18 **FSTPG**：快速编程 (Fast programming)
0：禁止快速编程
1：使能快速编程

位 17 保留，必须保持复位值

位 16 **STRT**：启动 (Start)
该位置 1 后可触发擦除操作。如果 MER 和 PER 位均复位并且 STRT 位置 1，可能会发生无法预知的行为而不会生成任何错误标志。应禁止此情况发生。
该位只能通过软件置 1，并在 FLASH_SR 中的 BSY 位清零后随之清零。
当 CPU1 请求操作（涉及安全 Flash 页）时，将会拒绝启动并产生总线错误。

位 15:11 保留，必须保持复位值。

位 10:3 **PNB[7:0]**：页编号选择 (Page number selection)
这些位用于选择要擦除的页：
0x00： 页 0
0x01： 页 1
...
0xFF： 页 255

位 2 **MER**：批量擦除 (Mass erase)
该位置 1 时，会触发批量擦除（所有用户页）。

位 1 **PER**：页擦除 (Page erase)
0：禁止页擦除
1：使能页擦除

位 0 **PG**：编程 (Programming)
0：禁止 Flash 编程
1：使能 Flash 编程

3.10.20 FLASH 寄存器映射

表 18. Flash 接口寄存器映射和复位值

偏移	寄存器	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
0x000	FLASH_ACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EMPTY	PES	Res.	Res.	DCRST	ICRST	DCEN	ICEN	PRFTEN	Res.	Res.	Res.	Res.	Res.	LATENCY [2:0]		
	Reset value																0	0			0	0	1	1	0						0	0	0
0x004	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x008	FLASH_KEYR	KEYR[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
0x00C	FLASH_OPT KEYR	OPTKEYR[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
0x010	FLASH_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PESD	CFGBSY	Res.	BSY	OPTVERR	RDERR	OPTNV	Res.	Res.	Res.	FASTERR	MISERR	PGSERR	SIZERR	PGAERR	WRPERR	PROGERR	Res.	OPERR	EOP
	Reset value													0	0		0	X	0	X				0	0	0	0	0	0	0		0	0
0x014	FLASH_CR	LOCK	OPTLOCK	Res.	Res.	OBL_LAUNCH	RDERRIE	ERRIE	EOPIE	Res.	Res.	Res.	Res.	Res.	FSTPG	OPTSTRT	STRT	Res.	Res.	Res.	Res.	Res.	PNB[7:0]								MER	PER	PG
	Reset value	1	1			0	0	0	0						0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x018	FLASH_ECCR	ECCD	ECCC	Res.	CPUID [2:0]			Res.	ECCCIE	Res.	Res.	Res.	SYSF_ECC	Res.	Res.	Res.	ADDR_ECC[16:0]																	
	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
0x020	FLASH_OPTR	AGC_TRIM [2:0]			Res.	nBOOT0	nSWBOOT0	SRAM2_RST	SRAM2_PE	nBOOT1	Res.	Res.	Res.	WWDG_SW	IWDG_STBY	IWDG_STOP	IWDG_SW	Res.	nRST_SHDW	nRST_STDBY	nRST_STOP	BOR_LEV [2:0]			ESE	RDP[7:0]							
	Reset value	X	X	X		X	X	X	X	X				X	X	X	X		X	X	X	X	X	X	1	X	X	X	X	X	X	X	X
0x024	FLASH_ PCROP1ASR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1A_STRT[8:0]								
	Reset value																								X	X	X	X	X	X	X	X	X
0x028	FLASH_ PCROP1AER	PCROP_RDP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1A_END[8:0]								
	Reset value	x																							X	X	X	X	X	X	X	X	X
0x02C	FLASH_ WRP1AR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_END[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_STRT[7:0]							
	Reset value									X	X	X	X	X	X	X	X									X	X	X	X	X	X	X	X

表 18. Flash 接口寄存器映射和复位值（续）

偏移	寄存器	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
0x030	FLASH_ WRP1BR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_END[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_STRT[7:0]							
	Reset value									X	X	X	X	X	X	X	X									X	X	X	X	X	X	X	X
0x034	FLASH_ PCROP1BSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1B_STRT[8:0]								
	Reset value																								X	X	X	X	X	X	X	X	X
0x038	FLASH_ PCROP1BER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1B_END[8:0]								
	Reset value																								X	X	X	X	X	X	X	X	X
0x03C	FLASH_ IPCCBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IPCCDBA[13:0]													
	Reset value																			X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x05C	FLASH_C2ACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PES	Res.	Res.	Res.	ICRST	Res.	ICEN	PRFTEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																	0				0		1	0								
0x060	FLASH_C2SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PESD	CFGBSY	Res.	BSY	Res.	RDERR	Res.	Res.	Res.	Res.	FASTERR	MISERR	PGSERR	SIZERR	PGAERR	WRPERR	PROGERR	Res.	OPERR	EOP
	Reset value													0	0		0		0					0	0	0	0	0	0	0		0	0
0x064	FLASH_C2CR	Res.	Res.	Res.	Res.	Res.	RDERRIE	ERRIE	EOPIE	Res.	Res.	Res.	Res.	Res.	FSTPG	Res.	STRT	Res.	Res.	Res.	Res.	Res.	PNB[7:0]								MER	PER	PG
	Reset value						0	0	0						0		0						0	0	0	0	0	0	0	0	0	0	0
0x080	FLASH_SFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DDS	Res.	Res.	Res.	FSD	SFSA[7:0]							
	Reset value																				1				0	X	X	X	X	X	X	X	X
0x084	FLASH_SRRVR	C2OPT	NBRSD	SNBRSA[4:0]					Res.	BRSD	SBRSA[4:0]					SBRV[17:0]																	
	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

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

4 无线电系统

4.1 简介

无线电系统具有超低功耗特性，符合蓝牙® 核心规范 BLE5.0 和 IEEE 802.15.4 标准。

无线电系统由以下部分构成：2.4 GHz RF 前端和蓝牙® 低功耗 (BLE) 以及 IEEE 802.15.4 物理层控制器。无线电系统由包含无线电较低协议软件层的 CPU2 控制。可通过邮箱消息系统实现与 CPU2 上运行的应用程序的接口。

4.2 无线电系统主要特性

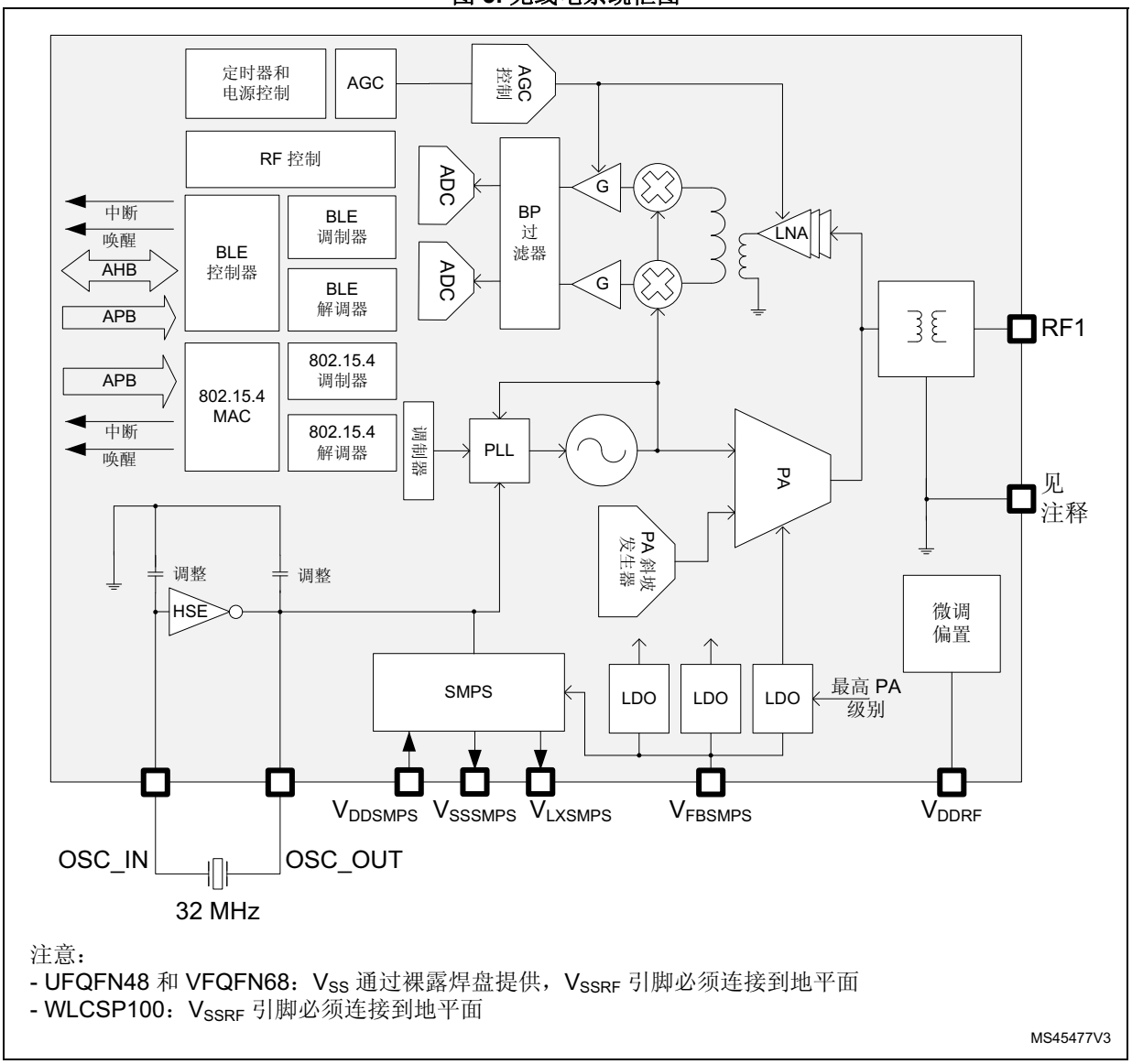
- 2.4 GHz RF 收发器支持：
  - 蓝牙® BLE5.0 (2 Mbps) 标准
  - IEEE 802.15.4-2011 标准
- 可编程输出功率
- RSSI
- 集成巴伦
- 蓝牙 BLE5.0 特性：
  - GAP：中央设备、外围设备、观察者和广播者角色
  - 同时支持多种角色
  - 主/从支持
  - ATT/GATT：客户端和服务器

4.3 无线电系统功能描述

4.3.1 概述

无线电系统的框图如*图 5* 所示。

图 5. 无线电系统框图



注意：
- UFQFN48 和 VFQFN68：V_{SS} 通过裸露焊盘提供，V_{SSRF} 引脚必须连接到地平面
- WLCSP100：V_{SSRF} 引脚必须连接到地平面

MS45477V3

对于默认 V_{FBSMPS} 电源电压，可支持最大默认发送输出功率。要实现更高的输出功率，应增大 V_{FBSMPS} 电源电压（更多相关信息，请参见 STM32WB55xx 数据手册）。

5 循环冗余校验计算单元 (CRC)

5.1 简介

CRC（循环冗余校验）计算单元使用多项式发生器从一个 8 位/16 位/32 位的数据字中产生 CRC 码。

在众多的应用中，基于 CRC 的技术还常用来验证数据传输或存储的完整性。根据功能安全标准的规定，这些技术提供了验证 Flash 完整性的方法。CRC 计算单元有助于在运行期间计算软件的签名，并将该签名与链接时生成并存储在指定存储单元的参考签名加以比较。

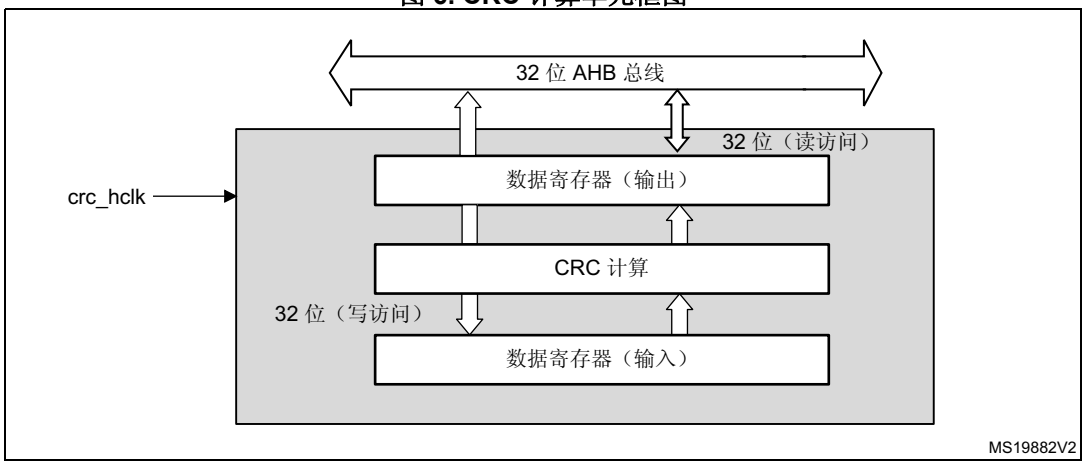
5.2 CRC 主要特性

- 使用 CRC-32（以太网）多项式：0x4C11DB7
  $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 或者，使用位数可编程的（7 位、8 位、16 位和 32 位）的完全可编程多项式
- 处理 8 位、16 位、32 位数据大小
- 可编程 CRC 初始值
- 单输入/输出 32 位数据寄存器
- 输入缓冲器可避免计算期间发生总线阻塞
- 对于 32 位数据大小，CRC 计算在 4 个 AHB 时钟周期 (HCLK) 内完成
- 8 位通用寄存器（可用于临时存储）
- I/O 数据的可逆性选项

5.3 CRC 功能说明

5.3.1 CRC 框图

图 6. CRC 计算单元框图



5.3.2 CRC 内部信号

表 19. CRC 内部输入/输出信号

信号名称	信号类型	说明
crc_hclk	数字输入	AHB 时钟

5.3.3 CRC 操作

CRC 计算单元具有单个 32 位读/写数据寄存器 (CRC_DR)。它用于输入新数据（写访问）和保存之前 CRC 计算的结果（读访问）。

对数据寄存器的每个写操作都会对之前的 CRC 值（存储在 CRC_DR 中）和新值再做一次 CRC 计算。CRC 计算针对整个 32 位数据字或逐个字节完成，具体取决于数据的写入格式。

CRC_DR 寄存器可按字、右对齐半字和右对齐字节进行访问。对于其他寄存器，只允许进行 32 位访问。

计算时间取决于数据宽度：

- 32 位数据需要 4 个 AHB 时钟周期
- 16 位数据需要 2 个 AHB 时钟周期
- 8 位数据需要 1 个 AHB 时钟周期

输入缓冲器中可立即写入第二个数据，无需因之前的 CRC 计算而等待任何等待状态。

可动态调整数据大小，从而能最大程度地减少给定字节数的写访问次数。例如，对 5 个字节进行 CRC 计算时，可先写入字，然后写入字节。

输入数据的顺序可反转，以管理各种数据存放方式（双字/单字/字节，大端/小端等等）。可对 8 位、16 位和 32 位数据执行反转操作，具体取决于 CRC_CR 寄存器中的 REV_IN[1:0] 位。

例如，输入数据 0x1A2B3C4D 在 CRC 计算中用作：

- 按字节执行位反转的 0x58D43CB2
- 按半字执行位反转的 0xD458B23C
- 按全字执行位反转的 0xB23CD458

通过将 CRC_CR 寄存器中 REV_OUT 位置 1 也可以将输出数据反转。

该操作按位进行：例如，输出数据 0x11223344 将转换为 0x22CC4488。

使用 CRC_CR 寄存器中的 RESET 控制位可将 CRC 计算器初始化为可编程值（默认值为 0xFFFFFFFF）。

可使用 CRC_INIT 寄存器对 CRC 初始值进行编程。对 CRC_INIT 寄存器进行写访问时会自动初始化 CRC_DR 寄存器。

CRC_IDR 寄存器可用于保存与 CRC 计算相关的临时值。它不受 CRC_CR 寄存器中的 RESET 位影响。

多项式可编程

多项式系数可完全通过 CRC_POL 寄存器进行编程，通过编程 CRC_CR 寄存器中的 POLYSIZE[1:0] 位可将多项式大小配置为 7 位、8 位、16 位或 32 位。不支持偶数多项式。

如果 CRC 数据小于 32 位，可从 CRC_DR 寄存器的最低有效位读取 CRC 的值。

为实现可靠的 CRC 计算，CRC 计算期间不能实时更改多项式的值或大小。因此，如果正在执行 CRC 计算，则在更改多项式前，应用程序必须先复位，或者先执行 CRC_DR 读操作。

默认的多项式值为 CRC-32（以太网）多项式：0x4C11DB7。

5.4 CRC 寄存器

5.4.1 CRC 数据寄存器 (CRC_DR)

CRC data register

偏移地址：0x00

复位值：0xFFFF FFFF

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

位 31:0 **DR[31:0]**：数据寄存器位 (Data register bits)

该寄存器用于向 CRC 计算器写入新数据。

读取寄存器时可读出之前的 CRC 计算结果。

如果数据大小小于 32 位，则最低有效位可用于写入/读取正确值。

5.4.2 CRC 独立数据寄存器 (CRC_IDR)

CRC independent data register

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

位 31:0 **IDR[31:0]**：通用的 32 位数据寄存器位 (General-purpose 32-bit data register bits)

这些位可用作四个字节的临时存储单元。

此寄存器不受 CRC_CR 寄存器中 RESET 位产生的 CRC 复位影响。

5.4.3 CRC 控制寄存器 (CRC_CR)

CRC control register

偏移地址：0x08

复位值：0x0000 0000

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

位 31:8 保留，必须保持复位值。

位 7 **REV_OUT**：反转输出数据 (Reverse output data)

该位用于控制输出数据位顺序的反转。

0：不影响位顺序

1：位反转输出格式

位 6:5 **REV_IN[1:0]**：反转输入数据 (Reverse input data)

这些位用于控制输入数据位顺序的反转。

00：不影响位顺序

01：按字节执行位反转

10：按半字执行位反转

11：按字执行位反转

位 4:3 **POLYSIZE[1:0]**：多项式大小 (Polynomial size)

这些位用于控制多项式的大小。

00：32 位多项式

01：16 位多项式

10：8 位多项式

11：7 位多项式

位 2:1 保留，必须保持复位值。

位 0 **RESET**：RESET 位 (RESET bit)

此位由软件置 1，用于复位 CRC 计算单元并将数据寄存器设置为存储在 CRC_INIT 寄存器中的值。此位只能置 1，将由硬件自动进行清零

5.4.4 CRC 初始值 (CRC_INIT)

CRC initial value

偏移地址：0x10

复位值：0xFFFF FFFF

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

位 31:0 **CRC_INIT[31:0]**：*可编程 CRC 初始值 (Programmable initial CRC value)*
此寄存器用于写入 CRC 初始值。

5.4.5 CRC 多项式 (CRC_POL)

CRC polynomial

偏移地址：0x14

复位值：0x04C1 1DB7

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

位 31:0 **POL[31:0]**：可编程多项式 (Programmable polynomial)
此寄存器用于写入要用于 CRC 计算的多项式系数。
如果多项式大小小于 32 位，则必须使用最低有效位编程正确值。

5.4.6 CRC 寄存器映射

表 20. CRC 寄存器映射和复位值

偏移	寄存器名称	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
0x00	**CRC_DR**	DR[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
0x04	**CRC_IDR**	IDR[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
0x08	**CRC_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_OUT	REV_IN[1:0]		POLYSIZE[1:0]		Res.	Res.	RESET
	Reset value																									0	0	0	0	0			0
0x10	**CRC_INIT**	CRC_INIT[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
0x14	**CRC_POL**	Polynomial coefficients																															
	Reset value	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	1	1	1	0	1	1	0	1	1	0	1	1	1

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

6 电源控制 (PWR)

6.1 电源

STM32WB55xx 器件的工作电压 V_{DD} 要求介于 1.71 V 到 3.6 V 之间。多个独立电源（V_{DDSMPS}、V_{FBSMPS}、V_{DDA}、V_{DDRF}、V_{DDUSB} 和 V_{LCD}）可用于特定外设：

- V_{DD} = 1.71 V 至 3.6 V

  V_{DD} 是为 I/O 和系统模拟模块（如复位、电源管理、内部时钟和低功耗稳压器）供电的外部电源。通过 VDD 引脚从外部提供。

- V_{DDSMPS} = 1.71 V 到 3.6 V

  V_{DDSMPS} 是为 SMPS 降压转换器供电的外部电源。它通过 VDDSMPS 电源引脚从外部提供，应连接到与 V_{DD} 相同的电源。

- V_{FBSMPS} = 1.40 V 到 3.6 V

  V_{FBSMPS} 是为主稳压器和 RF 系统稳压器供电的外部电源。它通过 VFBSMPS 引脚从外部提供，可以通过 SMPS 降压转换器供电或连接到与 V_{DD} 相同的电源。

- V_{DDA} = 1.62 V (ADC/COMP) 到 2.4 V (VREFBUF) 到 3.6 V。

  V_{DDA} 是为 A/D 转换器、D/A 转换器、电压参考缓冲器、运算放大器和比较器供电的外部模拟电源。V_{DDA} 电压与 V_{DD} 电压无关，不使用这些外设时，最好连接到 V_{DD}。

- V_{DDRF} = 1.71 V 到 3.6 V

  V_{DDRF} 是为无线电供电的外部电源。它通过 VDDRF 引脚从外部提供，应连接到与 V_{DD} 相同的电源。

- V_{DDUSB} = 3.0 V 到 3.6 V

  V_{DDUSB} 是为 USB 收发器供电的外部独立电源。V_{DDUSB} 电压与 V_{DD} 电压无关，不使用 USB 时，最好连接到 V_{DD}。

- V_{LCD} = 2.5 V 到 3.6 V

  LCD 控制器可由 VLCD 引脚外部供电，或由嵌入式升压转换器产生的内部电压内部供电，如果 V_{DD} 高于 2.0 V，则可产生高达 3.6 V 的电压。
  VLCD 与 PC3（当 LCD 不使用时，PC3 可用作 GPIO）复用。

- V_{BAT} = 1.55 V 到 3.6 V

  当 V_{DD} 掉电时，V_{BAT} 作为 RTC、32 kHz 外部时钟振荡器和备份寄存器的电源（通过电源开关供电）。

- V_{REF-} 和 V_{REF+}

  V_{REF+} 是 ADC 的输入参考电压，也是内部电压参考缓冲器（使能时）的输出。

  – 当 $V_{DDA} < 2$ V 时：V_{REF+} 必须等于 V_{DDA}

  – 当 $V_{DDA} \geq 2$ V 时：V_{REF+} 必须介于 2 V 到 V_{DDA} 之间

  当 ADC 不工作时，V_{REF+} 可接地。

  内部电压参考缓冲器支持两种输出电压值，可利用 VREFBUF_CSR 寄存器中的 VRS 位进行配置：

  – $V_{REF+} \approx 2.048$ V：这要求 V_{DDA} 大于等于 2.4 V

  – $V_{REF+} \approx 2.5$ V：这要求 V_{DDA} 大于等于 2.8 V

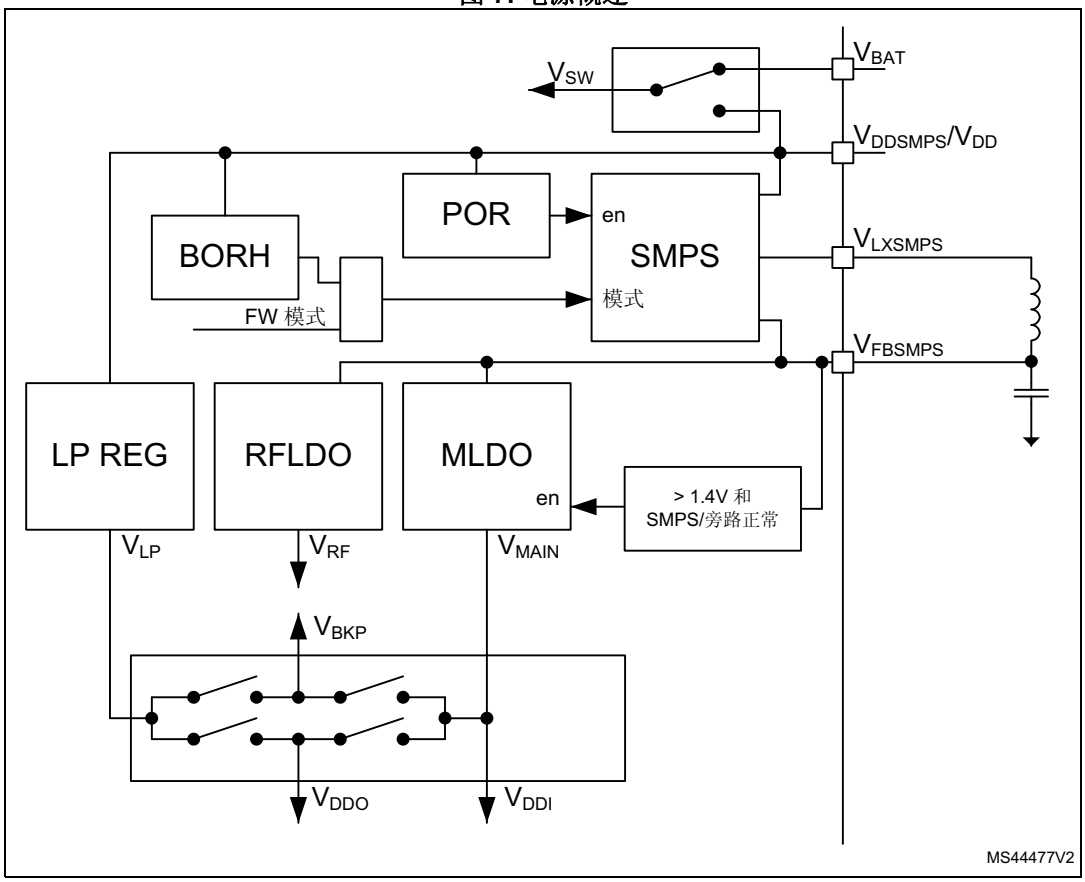
  VREF+ 引脚并非在所有封装上都可用，不可用时，将在内部绑定到 VDDA。当 VREF+ 与 VDDA 在封装中互相绑定时，内部电压参考缓冲器不可用且必须禁止（关于引脚分配说明，请参见数据手册）。

在上电和掉电期间，需要以下电源序列：

- 当 V_{DD} < 1 V 时，其他电源（V_{DDA}、V_{DDUSB} 和 V_{LCD}）必须保持低于 V_{DD} + 0.3 V。掉电期间，只有当提供给 MCU 的能量保持在 1 mJ 以下时，V_{DD} 才能暂时低于其他电源。
- 当 V_{DD} ≥ 1 V 时，所有电源均独立。

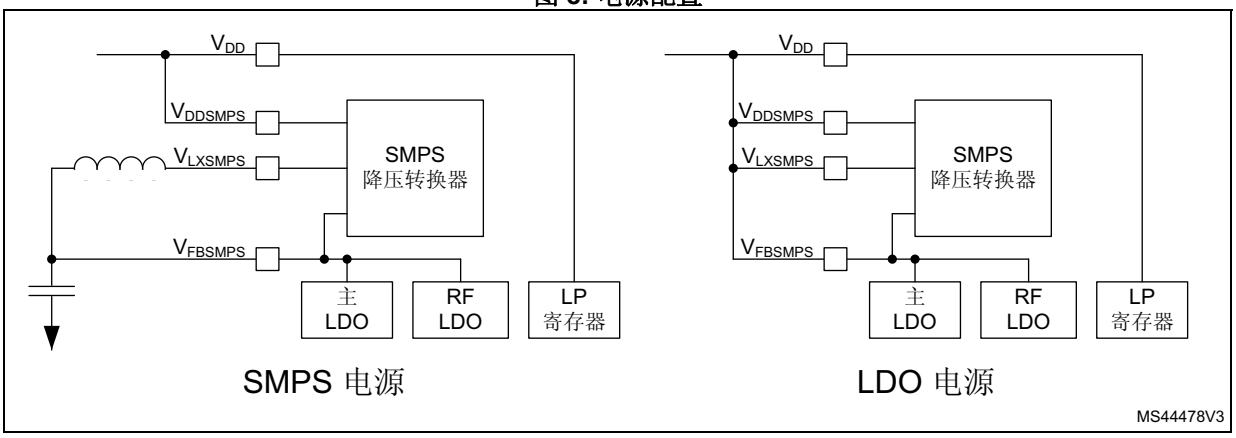
使用一个嵌入式线性调压器来为内部数字电源 V_{CORE} 供电。V_{CORE} 是为数字外设、SRAM1 和 SRAM2 供电的电源。Flash 由 V_{CORE} 和 V_{DD} 供电。

图 7. 电源概述



如*图 8* 所示，STM32WB55xx 支持不同的电源配置，按照*表 21* 的方案控制。

图 8. 电源配置



SMPS 降压转换器工作模式取决于 *PWR 控制寄存器 5 (PWR_CR5)* 中的 SMPSEN 设置以及系统工作模式“运行、停止 0、停止 1、停止 2、待机和关断”。

表 21. 电源配置控制

电源配置	SMPSEN	系统工作模式	说明
默认	0	NA	SMPS 处于旁路模式
SMPS 电源	1(1)	运行	SMPS 处于 SMPS 模式
		停止 0	
		停止 1	SMPS 处于开路模式
		停止 2	
		待机和关断	
LDO 电源	0	任意	SMPS 处于旁路模式

1. 当使能 SMPS 降压转换器 SMPS 模式时，最好使能 BORH 来监控电源。

POR 复位后，SMPS 降压转换器处于旁路模式。

在停止 1、停止 2 和待机期间，SMPS 降压转换器由硬件置于开路模式（请参见*表 21*）。当退出低功耗模式（关断除外）时，SMPS 降压转换器将由硬件设置为 SMPSEN 位选择的模式（SMPS 模式或旁路）。*PWR 控制寄存器 5 (PWR_CR5)* 中的 SMPSEN 位保持待机模式。

当 SMPS 降压转换器处于 SMPS 模式时，可以通过 *PWR 控制寄存器 5 (PWR_CR5)* 中的 BORHC 配置 BORH，以便在电源降至 SMPS 降压转换器 SMPS 模式工作电源电压以下时，能够实时接通。当 V_{DD} 降至所选 BORH 阈值以下时，SMPS 降压转换器被强制进入旁路模式，寄存器位 SMPSEN 被清零，同时会生成 SMPSFBF 中断（如果已使能）。当 V_{DD} 电源升至 BORH 阈值以上时，将生成 BORHF 中断（如果已使能）。由软件将 SMPS 降压转换器切换回 SMPS 模式。

当禁止 SMPS 降压转换器时，SMPSEN 为 0，SMPS 降压转换器处于旁路模式，可以使用应用 LDO 模式。

6.1.1 独立模拟外设电源

为了提高 ADC 转换精度、扩展供电的灵活性，模拟外设配有独立电源，可以单独滤波并屏蔽 PCB 上的噪声。

- 模拟外设电压源从单独的 V_{DDA} 引脚输入。
- V_{SSA} 引脚提供了独立的电源接地连接。

V_{DDA} 电源电压可与 V_{DD} 不同。在使能任一由 V_{DDA} 供电的模拟外设（A/D 转换器、比较器和电压参考缓冲器）之前，必须检查是否存在 V_{DDA}。

V_{DDA} 电源可由外设电压监测功能监测，并与一个阈值（对于 PVM3，阈值为 1.65 V）进行比较，有关详细信息，请参见*第 6.2.3 节*。

当使用单一供电时，可通过外部滤波电路将 V_{DDA} 外部连接至 V_{DD}，以获得无噪声的 V_{DDA} 参考电压。

模拟信号的噪声性能可能受集成 SMPS 切换的影响。为了防止这种情况，在测量模拟信号时可以实时关闭 SMPS。

ADC 参考电压

为确保低电压输入和输出上实现更高精度，用户可将 V_{REF+} 连接至一个低于 V_{DDA} 的独立参考电压。对于模拟输入 (ADC) 信号，V_{REF+} 为最高电压，以满量程值表示。

V_{REF+} 可由外部参考或内部缓冲的电压参考 (VREFBUF) 来提供。

通过将 *VREFBUF 控制和状态寄存器 (VREFBUF_CSR)* 中的 ENVR 位置 1 来使能内部电压参考。当 VRS 位置 1 时，电压参考设为 2.5 V，当 VRS 位清零时，电压参考设为 2.048 V。内部电压参考还可通过 V_{REF+} 引脚为外部元件供电。有关更多信息，请参见器件数据手册或*第 17 节：参考电压缓冲器 (VREFBUF)*。

6.1.2 独立的 USB 收发器电源

USB 收发器通过一个单独的电源引脚 V_{DDUSB} 供电。V_{DDUSB} 范围从 3.0 V 到 3.6 V，完全独立于 V_{DD} 或 V_{DDA}。

复位后，由 V_{DDUSB} 供电的 USB 功能是逻辑隔离且电隔离的，因而不可用。当 V_{DDUSB} 电源存在时，使用 USB FSOTG 外设前，必须通过将 *PWR 控制寄存器 2 (PWR_CR2)* 中的 USV 位置 1，解除此隔离。

V_{DDUSB} 电源可由外设电压监测功能 (PVM1) 监测，并与内部参考电压（V_{REFINT}，1.2 V 左右）进行比较，有关更多详细信息，请参见*第 6.2.3 节*。

6.1.3 独立 LCD 供电

VLCD 引脚用于控制玻璃 LCD 的对比度。可用两种方法使用这一引脚：

- 它可从外部电路接收所需的最大电压，由微控制器通过区段和公用线供给玻璃 LCD。
- 还可用它连接外部电容，微控制器将该电容用于内部的升压转换器。通过软件控制此升压转换器，以向玻璃 LCD 的区段和公用线提供所需电压。

向区段和公用线提供的电压定义了玻璃 LCD 像素的对比度。当用户在帧间配置了死区时，可降低此对比度。

- 向 VLCD 引脚提供外部电源时，该电源范围应为 2.5 V 至 3.6 V。它不依赖于 V_{DD}。
- 当 LCD 基于内部升压转换器时，VLCD 引脚应当连接一个电容（更多信息，请参见产品数据手册）。

6.1.4 电池备份域

为了在 V_{DD} 掉电时，还能保留备份寄存器的内容，且让 RTC 继续工作，可将 V_{BAT} 引脚连接到由电池或者其他电源提供的可选备用电源上。

V_{BAT} 引脚为 RTC 单元、LSE 振荡器和 PC13 到 PC15 I/O 供电，允许 RTC 在主电源关闭时也可工作。V_{BAT} 电源的开关由复位模块中内置的掉电复位电路进行控制。

警告： **在 $t_{RSTTEMPO}$（V_{DD} 启动后的一段延迟）期间或检测到 PDR 后，V_{BAT} 与 V_{DD} 之间的电源开关仍连接到 V_{BAT}。**
在启动阶段，如果 V_{DD} 的建立时间小于 $t_{RSTTEMPO}$（有关 $t_{RSTTEMPO}$ 的值，请参见数据手册）且 $V_{DD} > V_{BAT} + 0.6$ V，会有电流经由 V_{DD} 和电源开关 (V_{BAT}) 之间连接的内部二极管注入 V_{BAT} 引脚。
如果连接到 V_{BAT} 引脚的电源/电池无法承受此注入电流，则强烈建议在该电源与 V_{BAT} 引脚之间连接一个低压降二极管。

如果应用中没有使用任何外部电池，建议将该 V_{BAT} 外部连接到带有 100 nF 外部去耦电容的 V_{DD} 上。

通过 V_{DD} 对备份域供电时（模拟开关连接到 V_{DD}），以下引脚可用：

- PC13、PC14 和 PC15，可用作 GPIO 引脚
- PC13、PC14 和 PC15，可由 RTC 或 LSE 配置（请参见*第 29.3 节：RTC 功能说明*）
- PA0/RTC_TAMP2 和 PC12/RTC_TAMP3（通过 RTC 配置为入侵引脚时）

注： *由于模拟开关仅能传递有限的电流 (3 mA)，因此使用输出模式的 GPIO PC13 到 PC15 是受限的：速率必须限制在 2 MHz，最大负载为 30 pF，且这些 I/O 不能作为电流源使用（如，驱动 LED）。*

通过 V_{BAT} 对备份域供电时（由于不存在 V_{DD}，内部电源开关连接到 V_{BAT}），可实现以下功能：

- PC13、PC14 和 PC15 仅受 RTC 或 LSE 控制（请参见*第 29.3 节：RTC 功能说明*）
- PA0/RTC_TAMP2 和 PC12/RTC_TAMP3（通过 RTC 配置为入侵引脚时）

备份域访问

系统复位后，备份域（RTC 寄存器和备份寄存器）将受到保护，以防止意外的写访问。要使能对备份域的访问，请按以下步骤进行操作：

1. 将 *PWR 控制寄存器 1 (PWR_CR1)* 中的 DBP 位置 1，使能对备份域的访问

VBAT 电池充电

当 VDD 存在时，可通过内部电阻为 VBAT 上的外部电池充电。

VBAT 充电可通过一个 5 kΩ 的电阻或 1.5 kΩ 的电阻来实现，取决于 *PWR 控制寄存器 4 (PWR_CR4)* 中 VBRS 位的值。

可将 *PWR 控制寄存器 4 (PWR_CR4)* 中的 VBE 位置 1 来使能电池充电，电池充电在 VBAT 模式下自动禁止。

6.1.5 稳压器

两个嵌入式线性稳压器为待机电路和备份域以外的所有数字电路供电。根据系统的最大工作频率，主稳压器输出电压 (V_{CORE}) 可由软件编程为两种不同的电源范围（范围 1 和范围 2），以优化功耗（请参见*第 8.2.10 节：时钟源频率与电压调节*和*第 3.3.4 节：读访问延迟*）。

稳压器在复位后始终处于使能状态。根据应用模式，V_{CORE} 电源由主稳压器 (MR) 或低功耗稳压器 (LPR) 提供。

- 在运行模式、睡眠模式和停止 0 模式中，两个稳压器使能，主稳压器 (MR) 为 V_{CORE} 域（内核、存储器和数字外设）提供全功率。
- 在低功耗运行模式和低功耗睡眠模式中，主稳压器关闭，低功耗稳压器 (LPR) 为 V_{CORE} 域提供低功率，保持寄存器及内部 SRAM1 和 SRAM2 的内容。
- 在停止 1 和停止 2 模式中，主稳压器关闭，低功耗稳压器 (LPR) 为 V_{CORE} 域提供低功率，保持寄存器及内部 SRAM1 和 SRAM2 的内容。
- 在待机模式且 SRAM2a 内容保持时（*PWR 控制寄存器 3 (PWR_CR3)* 中的 RRS 位置 1），主稳压器 (MR) 关闭，低功耗稳压器 (LPR) 仅为 SRAM2a 供电。内核和数字外设（待机电路和备份域除外）、SRAM1 和 SRAM2b 掉电。
- 在待机模式中，两个稳压器均掉电。除待机电路和备份域外，寄存器、SRAM1 和 SRAM2 的内容都将丢失。
- 在关断模式中，两个稳压器均掉电。退出关断模式时，产生一个上电复位。因此，除备份域外，寄存器、SRAM1 和 SRAM2 的内容都将丢失。

6.1.6 动态电压调节管理

动态电压调节是一种电源管理技术，包括根据应用性能和功耗要求，增大或减小用于数字外设的电压 (V_{CORE})。

通过动态电压调节增大 V_{CORE} 称作“过压”，用于提高器件性能。

通过动态电压调节减小 V_{CORE} 称为“欠压”，用于节省功率，尤其是用于笔记本电脑和有限电能来自电池的其他移动设备。

- 范围 1：高性能范围。

  主稳压器提供了 1.2 V 的典型输出电压。系统时钟频率可达 64 MHz。读访问的 Flash 访问时间最小，可进行写入和擦除操作。
- 范围 2：低功耗范围。

  主稳压器提供了 1 V 的典型输出电压。系统时钟频率可达 16 MHz。相比于范围 1，读访问的 Flash 访问时间增加；可进行写入和擦除操作。

通过 *PWR 控制寄存器 1 (PWR_CR1)* 中的 VOS 位选择电压调节。

从范围 1 进入范围 2 的时序为：

1. 降低系统频率，使其值小于或等于 16 MHz。
2. 根据范围 2 中新的频率目标来调整等待状态的数量（FLASH_ACR 中的 LATENCY 位）。
3. 在 *PWR 控制寄存器 1 (PWR_CR1)* 的 VOS 位中选择范围 2。

从范围 2 进入范围 1 的时序为：

1. 在 *PWR 控制寄存器 1 (PWR_CR1)* 的 VOS 位中选择范围 1。
2. 等待 *PWR 状态寄存器 2 (PWR_SR2)* 中的 VOSF 标志清零。
3. 根据范围 1 中新的频率目标来调整等待状态的数量（FLASH_ACR 中的 LATENCY 位）。
4. 增加系统频率。

6.2 电源监控器

6.2.1 上电复位 (POR)/掉电复位 (PDR)/欠压复位 (BOR)

该器件具有一个集成的上电复位/掉电复位电路，与欠压复位电路耦合。

通过选项字节，可对 5 个 BOR 阈值进行选择。

BOR 可以在两种工作模式下使用，具体通过 *PWR 控制寄存器 5 (PWR_CR5)* 中的 BORHC 位配置：

- 复位模式，V_{DD} 电源低于所选的 BOR 阈值时，生成系统复位。
- 当 V_{DD} 电源低于所选 BOR 阈值时，强制 SMPS 降压转换器进入旁路模式。当 V_{DD} 电源低于最低 BOR 阈值时，始终生成系统复位。

除关断模式外，BOR 在所有功耗模式下均激活，且不可禁用。需要使能 BOR 机制，并且可以在需要时随时禁止。

复位模式

上电期间，BOR 将使器件保持复位状态，直到电源电压 V_{DD} 达到指定的 V_{BORx} 阈值。当 V_{DD} 降至所选阈值以下时，将使器件复位。当 V_{DD} 高于 V_{BORx} 上限时，释放器件复位，系统可以启动。

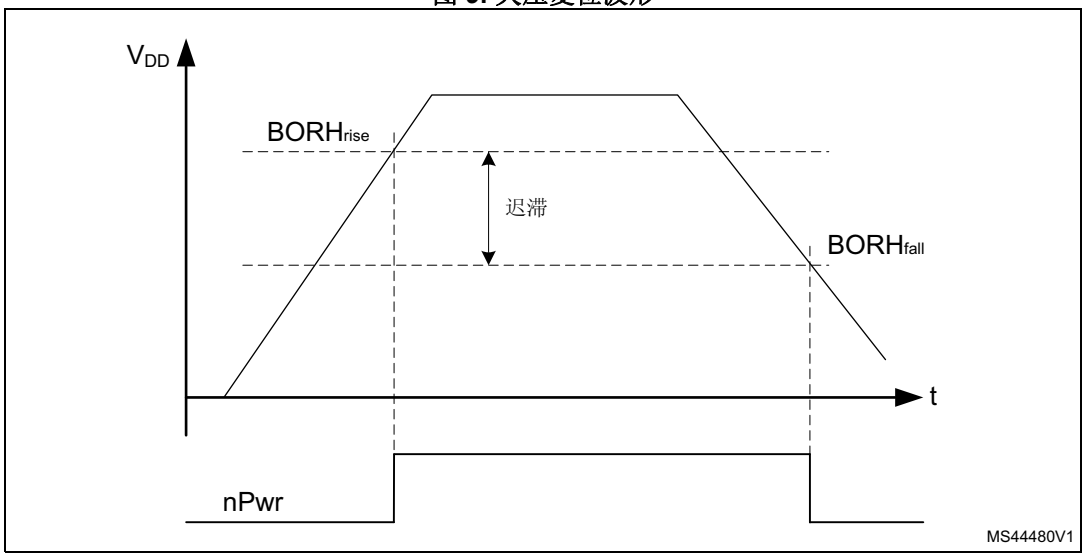
强制 SMPS 降压转换器旁路模式

上电期间，BOR 将使器件保持复位状态，直到电源电压 V_{DD} 达到指定的 V_{BOR0} 阈值。当 V_{DD} 降至此阈值以下时，生成器件复位。当 V_{DD} 高于 V_{BOR0} 上限时，释放器件复位，系统可以启动。

在 SMPS 降压转换器要进入 SMPS 模式时或处于 SMPS 模式期间，使用选定的 BORx（x = 1、2、3 和 4）阈值确定是否使用 SMPS 模式或是否强制 SMPS 降压转换器进入旁路模式。当 V_{DD} 降至所选阈值以下时，进入或处于 SMPS 模式的 SMPS 降压转换器将被强制进入旁路模式。如果使能，可生成 SMPS 降压转换器强制旁路中断 (SMPSFBF)。当 V_{DD} 升至所选 BOR 上限值以上时，如果使能，可生成 BOR 中断 (BORHF)。由软件将 SMPS 降压转换器设置为 SMPS 模式。

有关欠压复位阈值的相关详细信息，请参见数据手册的电气特性部分。

图 9. 欠压复位波形



1. 复位持续时间 $t_{RSTTEMPO}$ 仅针对 BOR 最低阈值 (V_{BOR0}) 存在。

6.2.2 可编程电压检测器 (PVD)

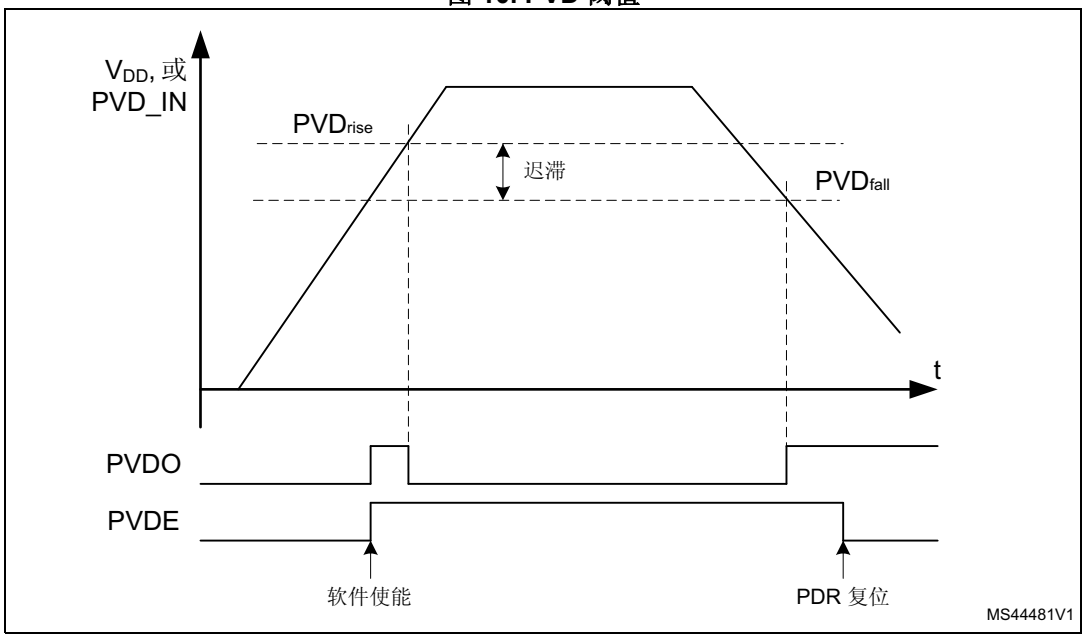
可以使用 PVD 监视 V_{DD} 电源，方法是将其与 *PWR 控制寄存器 2 (PWR_CR2)* 中的 PLS[2:0] 位所选的阈值进行比较。

也可以使用 PVD 来监视 PVD_IN 引脚上的电压级别。在这种情况下，会将 PVD_IN 电压与内部 VREFINT 值进行比较。

通过设置 PVDE 位来使能 PVD。

PWR 状态寄存器 2 (PWR_SR2) 中提供了 PVDO 标志，用于指示 V_{DD} 或 PVD_IN 电压是大于还是小于 PVD 阈值。该事件内部连接到 EXTI 线 16，如果通过 EXTI 寄存器使能，则可以产生中断。当 V_{DD} 或 PVD_IN 电压降至 PVD 阈值以下以及/或者当 V_{DD} 或 PVD_IN 电压升至 PVD 阈值以上时，可以产生 PVD 输出中断，具体取决于 EXTI 线 16 上升沿/下降沿的配置。该功能的用处之一就是可以在中断服务程序中执行紧急关闭系统的任务。

图 10. PVD 阈值



6.2.3 外设电压监测 (PVM)

默认情况下仅监测 V_{DD}，因为它是唯一与所有系统功能相关的电源。其他电源（V_{DDA} 和 V_{DDUSB}）可以独立于 V_{DD}，并可以通过外设电压监测 (PVM) 功能进行监测。

每个 PVM 都是一个比较器，用于比较固定阈值 V_{PVMx} 和所选电源。PVMOx 标志用于表示独立电源是高于还是低于 PVMx 阈值：当电源电压高于 PVMx 阈值时，PVMOx 标志清零，当电源低于 PVMx 阈值时，PVMOx 标志置 1。

每个 PVM 输出都连接到 EXTI 线，如果通过 EXTI 寄存器使能，则可以产生中断。当独立电源降至 PVMx 阈值以下以及/或者升至 PVMx 阈值以上时，会产生 PVMx 输出中断，具体取决于 EXTI 线上升/下降沿的配置。

在停止 0、停止 1 和停止 2 模式下，每个 PVM 都可以保持有效，并且可以从停止模式唤醒 PVM 中断。

表 22. PVM 功能

PVM	电源	PVM 阈值	EXTI 线
PVM1	V_{DDUSB}	V_{PVM1}（1.2 V 左右）	31
PVM2	未使用	-	-
PVM3	V_{DDA}	V_{PVM3}（1.65 V 左右）	33
PVM4	未使用	-	-

默认情况下认为独立电源（V_{DDA} 和 V_{DDUSB}）不存在，并应用逻辑隔离和电气隔离以忽略由这些专用电源供电的外设发出的任何信息。

- 如果这些电源外部短接到 V_{DD}，应用应假定它们可以在不使能任何外设电压监测的情况下使用。
- 如果这些电源独立于 V_{DD}，则可以使能外设电压监测 (PVM)，确认是否存在电源。

使用 USB_FS 外设前，必须按以下顺序进行操作：

1. 如果 V_{DDUSB} 独立于 V_{DD}：
   a) 通过将 *PWR 控制寄存器 2 (PWR_CR2)* 中的 PVME1 位置 1，使能 PVM1。
   b) 等待 PVM1 唤醒时间。
   c) 等待 *PWR 状态寄存器 2 (PWR_SR2)* 中的 PVMO1 位清零。
   d) 可选：禁止 PVM1 以节约功耗。
2. 将 *PWR 控制寄存器 2 (PWR_CR2)* 中的 USV 位置 1，以移除 V_{DDUSB} 电源隔离。

使用以下任一模拟外设前，必须按照以下顺序进行操作：模数转换器、数模转换器、比较器、运算放大器和电压参考缓冲器：

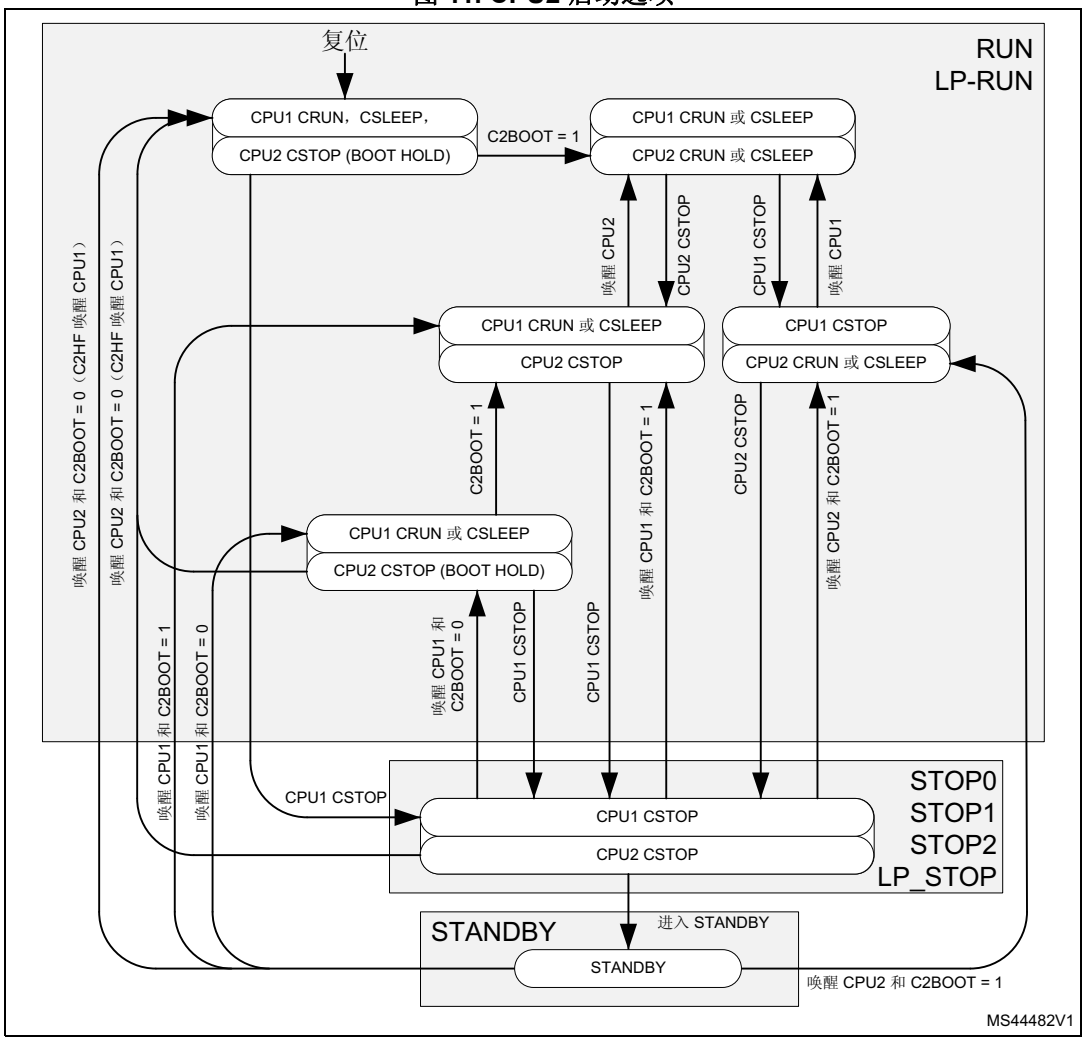
1. 如果 V_{DDA} 独立于 V_{DD}：
   a) 通过将 *PWR 控制寄存器 2 (PWR_CR2)* 中的 PVME3 位置 1，使能 PVM3。
   b) 等待 PVM3 唤醒时间。
   c) 等待 *PWR 状态寄存器 2 (PWR_SR2)* 中的 PVMO3 位清零。
   d) 可选：禁止 PVM3 以节约功耗。
2. 使能模拟外设，此操作自动移除 V_{DDA} 隔离。

6.3 CPU2 启动

CPU2 的启动由 *PWR 控制寄存器 4 (PWR_CR4)* 中的 C2BOOT 位控制。这样一来，在启动 CPU2 之前，CPU1 可以在复位后初始化系统或从系统低功耗模式唤醒。

- 复位后，C2BOOT 位阻止 CPU2 启动。只有在 CPU1 将 C2BOOT 位置 1 后，CPU2 才会启动。
- 当退出系统停止 0、停止 1、停止 2 或待机低功耗模式时，可以通过 C2BOOT 位控制 CPU2 的启动。
  – 当 C2BOOT = 1 时，在系统低功耗模式之后，CPU2 将在通过唤醒源唤醒时启动。
  – 当 C2BOOT = 0 时，在系统低功耗模式之后，CPU2 被阻止启动。CPU1 通过 C2HF 而非 CPU2 唤醒源唤醒。CPU1 通过将 C2BOOT 位置 1 来启动 CPU2。

图 11. CPU2 启动选项



当阻止 CPU2 启动时，从低功耗模式唤醒的启动过程如下：

- 在 CPU1 进入 CSTOP 模式之前，它会清零 C2BOOT 位。
- 当 CPU1 退出 CSTOP 时：
  - 当系统保持运行模式时，它会将 C2BOOT 位置 1，然后处理唤醒事件。
  - 当系统通过 CPU1 唤醒源退出低功耗模式时，它将初始化系统并将 C2BOOT 位置 1，然后处理唤醒事件。
  - 当系统通过 C2HF 唤醒源退出低功耗模式时，它将初始化系统并将 C2BOOT 位置 1，然后返回 CSTOP。
- CPU2 没有特殊的唤醒过程。当 CPU2 唤醒时，系统已由 CPU1 初始化。因此 CPU2 可以直接处理唤醒事件。

当系统保持运行模式时（由于无线电系统），CPU2 将独立于 C2BOOT 设置从 CTSTOP 模式唤醒。

6.4 低功耗模式

默认情况下，系统复位或电源复位后，微控制器处于运行模式，并且至少有一个 CPU 处于执行代码的 CRun 模式。系统提供了多个低功耗模式，可在 CPU 不需要运行时（例如等待外部事件时）节省功耗。用户必须选择具体的模式，以在功耗、启动时间和可用唤醒源之间寻求最佳平衡。

各 CPU 具有两种低功耗模式，可在执行 WFI、WFE 或在使能 SLEEPONEXIT 时从异常处理程序返回时通过 CPU 进入。

- CSleep 模式：当 CPU 进入低功耗模式且 SLEEPDEEP 禁止时，Arm® “睡眠模式”。
- CStop 模式：当 CPU 进入低功耗模式且 SLEEPDEEP 使能时，Arm® “深度睡眠模式”。

器件有多种低功耗模式：

- *睡眠模式*：CPU 时钟关闭，包括 CPU 内核外设（例如 NVIC、SysTick 等）在内的所有外设都可以运行，并在发生中断或事件时唤醒 CPU。
- *低功耗运行模式（LP 运行）*：当系统时钟频率减少到 2 MHz 以下时实现此模式。从 SRAM 或 Flash 执行代码。稳压器处于低功耗模式以最大程度降低工作电流。
- *低功耗睡眠模式（LP 睡眠）*：从低功耗运行模式进入此模式：CPU 关闭。
- *停止 0 模式*、*停止 1 模式* 和 *停止 2 模式*：保留 SRAM1、SRAM2 和所有寄存器的内容。V_{CORE} 域中的所有时钟都停止，并禁止 PLL、MSI、HSI16 和 HSE。LSI 和 LSE 可以保持运行。

  RTC 可以保持有效（带 RTC 的停止模式，不带 RTC 的停止模式）。

  一些具有唤醒功能的外设可以在停止模式期间使能 HSI16 RC，以检测它们的唤醒条件。

  在停止 2 模式下，大多数 V_{CORE} 域处于低泄漏模式。

  停止 1 提供最大数量的有效外设和唤醒源，相比停止 2，其唤醒时间较短，但功耗较高。在停止 0 模式下，主稳压器保持开启，可实现最快的唤醒，但功耗高很多。有效外设和唤醒源与停止 1 模式下相同。

  当从停止 0、停止 1 或停止 2 模式退出时，系统时钟为最高 48 MHz 的 MSI 或 HSI16，具体取决于软件配置。
- *待机模式*：V_{CORE} 域断电。但是，可以保留 SRAM2a 内容：
  - 在待机模式下，当 *PWR 控制寄存器 3 (PWR_CR3)* 中的 RRS 位置 1 时，SRAM2a 内容保留。在这种情况下，SRAM2a 由低功耗稳压器供电。
  - 在待机模式下，当 *PWR 控制寄存器 3 (PWR_CR3)* 中的 RRS 位清零时。在这种情况下，主稳压器和低功耗稳压器关闭。

  V_{CORE} 域中的所有时钟都停止，并禁止 PLL、MSI、HSI16 和 HSE。LSI 和 LSE 可以保持运行。

  RTC 可以保持有效（带 RTC 的待机模式，不带 RTC 的待机模式）。

  当退出待机模式时，系统时钟为 HSI16。
- *关断模式*：V_{CORE} 域断电。V_{CORE} 域中的所有时钟都停止，并禁止 PLL、MSI、HSI16、LSI 和 HSE。LSE 可以保持运行。当退出关断模式时，系统时钟为 4 MHz 的 MSI。在该模式下，会禁止电源电压监测，并且电源电压下降时不能保证产品特性。

注： *仅当两个 CPU 均处于 CStop 模式时，才会进入停止、待机和关断模式。*

此外，可以通过减慢系统时钟和/或在 APB 和 AHB 外设未使用时对其进行时钟门控来降低运行模式下的功耗。

系统工作模式取决于 CPU1、CPU2 和无线电子系统工作模式。仅当全部三个子系统都允许时，系统才会进入低功耗模式。

系统复位后，CPU1 处于 CRUN 模式。仅在由 CPU1 通过 C2BOOT 寄存器位使能时，CPU2 才会启动。只要 CPU1 不启动 CPU2，STM32WB55xx 就会作为单 CPU 系统运行。CPU1 可自行进入系统低功耗模式，也可从系统低功耗模式唤醒。

当 CPU2 启动时，CPU1、CPU2 和无线电子系统可以自行进入系统低功耗模式以及从该模式唤醒。*表 23* 详细介绍了不同子系统的不同唤醒源。

表 23. 子系统低功耗唤醒源

唤醒源	CPU1	CPU2	RADIO
EXTI	从停止模式	从停止模式	不可用
RTC	从停止和待机模式	从停止和待机模式	不可用
WKUP	从停止和待机模式	从停止和待机模式	不可用
SMPS	从停止模式	从停止模式	不可用
RADIO	从停止模式	从停止模式	不可用
RFWAKEUP	不可用	从停止和待机模式	从停止和待机模式

要进入的系统低功耗模式取决于两个 CPU 在 *PWR 控制寄存器 1 (PWR_CR1)* 和 *PWR CPU2 控制寄存器 1 (PWR_C2CR1)* 的 LPMS 位中选择的允许模式。在 CPU2 由 C2BOOT 保持时，这同样适用。

图 12 所示为工作模式状态图。CPU1、CPU2 和无线电子系统根据其自身的子系统状态以相互依存的方式运行。每个子系统都有自己的唤醒源，允许其从停止和待机模式唤醒。要使器件处于停止、待机或关断模式，全部三个子系统都需要处于 CStop。当一个子系统进入 CRun 模式时，器件进入运行模式。

图 12. 低功耗模式可能的转换

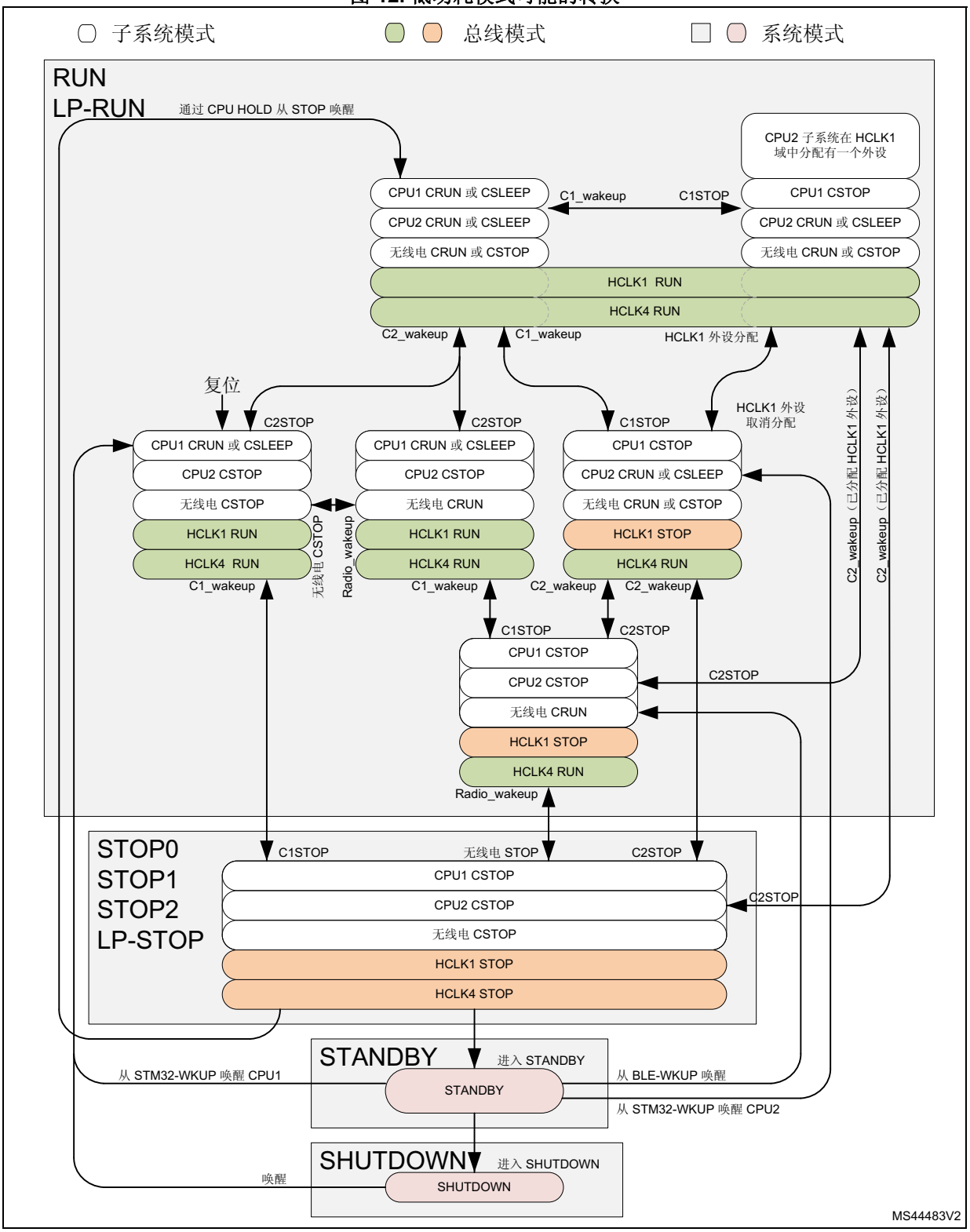


表 24. 低功耗模式汇总

模式名称	条目	唤醒源(1)	唤醒系统时钟	对时钟的影响	稳压器	
					MR	LPR
睡眠（立即睡眠或退出时睡眠）	WFI 或从 ISR 返回	任意中断	与进入睡眠模式之前相同	CPU 时钟关闭 对其他时钟或模拟时钟源无影响	开启	开启
	WFE	唤醒事件				
低功耗运行	将 LPR 位置 1	将 LPR 位清零	与低功耗运行时钟相同	无	关闭	开启
低功耗睡眠	将 LPR 位置 1 + WFI 或从 ISR 返回	任意中断	与进入低功耗睡眠模式之前相同	CPU 时钟关闭 对其他时钟或模拟时钟源无影响	关闭	开启
	将 LPR 位置 1 + WFE	唤醒事件			关闭	开启
停止 0	LPMS=“000”+ SLEEPDEEP 位 + WFI 或从 ISR 或 WFE 返回	任意 EXTI 线（在 EXTI 寄存器中配置） 特定外设事件	当 RCC_CFGR 中 STOPWUCK=1 时为 HSI16 当 STOPWUCK=0 时，是 MSI，频率是进入停止模式前的频率	所有时钟关闭 LSI 和 LSE 除外	开启	开启
停止 1	LPMS=“001”+ SLEEPDEEP 位 + WFI 或从 ISR 或 WFE 返回				关闭	
停止 2	LPMS=“010”+ SLEEPDEEP 位 + WFI 或从 ISR 或 WFE 返回					
待机 +SRAM2a	LPMS=“011”+ 将 RRS 位置 1 + SLEEPDEEP 位 + WFI 或从 ISR 或 WFE 返回	WKUP 引脚边沿、RTC 事件、LSECSS、NRST 引脚的外部复位、IWDG 复位	HSI16			
待机	LPMS=“011”+ 将 RRS 位清零 + SLEEPDEEP 位 + WFI 或从 ISR 或 WFE 返回	WKUP 引脚边沿、RTC 事件、LSECSS、NRST 引脚的外部复位、IWDG 复位			关闭	关闭
关断	LPMS=“1--”+ SLEEPDEEP 位 + WFI 或从 ISR 或 WFE 返回	WKUP 引脚边沿、RTC 事件、NRST 引脚的外部复位	MSI 4 MHz	所有时钟关闭 LSE 除外	关闭	关闭

1. 请参见*表 25*。

表 25. 功能取决于系统工作模式[1]

外设	运行	睡眠	低功耗运行	低功耗睡眠	停止 0		停止 1		停止 2		待机		关断		VBAT
					-	唤醒能力	-	唤醒能力	-	唤醒能力	-	唤醒能力	-	唤醒能力	
CPU1	Y	-	Y	-	-	-	-	-	-	-	-	-	-	-	-
CPU2	Y	-	Y	-	-	-	-	-	-	-	-	-	-	-	-
无线电系统（BLE，802）	Y	Y	-	-	-	Y	-	Y	-	Y	-	Y[2]	-	-	-
Flash（高达 1 MB）	Y	Y	O	O	R	-	R	-	R	-	R	-	R	-	R
SRAM1（高达 192 KB）	Y	O[4]	Y	O[4]	R	-	R	-	R	-	-	-	-	-	-
SRAM2a（高达 32 KB）	Y	O[4]	Y	O[4]	R	-	R	-	R	-	O[3]	-	-	-	-
SRAM2b（高达 32 KB）	Y	O[4]	Y	O[4]	R	-	R	-	R	-	-	-	-	-	-
QUADSPI	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
备份寄存器	Y	Y	Y	Y	R	-	R	-	R	-	R	-	R	-	R
欠压复位 (BOR)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-
欠压 SMPS 强制旁路 (BOR)	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-
可编程电压检测器 (PVD)	O	O	O	O	O	O	O	O	O	O	-	-	-	-	-
外设电压监测器（PVMx；x=1、3）	O	O	O	O	O	O	O	O	O	O	-	-	-	-	-
DMAx（x=1、2）	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
高速内部时钟（HSI16）	O	O	O	O	O[5]	-	O[5]	-	-	-	-	-	-	-	-
振荡器 HSI48	O	O	-	-	-	-	-	-	-	-	-	-	-	-	-
高速外部 (HSE)	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
低速内部 (LSI)	O	O	O	O	O	-	O	-	O	-	O	-	-	-	-
低速外部 (LSE)	O	O	O	O	O	-	O	-	O	-	O	-	O	-	O
多速内部 (MSI)	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
时钟安全系统 (CSS)	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
LSE 上的时钟安全系统	O	O	O	O	O	O	O	O	O	O	O	O	-	-	-
RTC/自动唤醒	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
RTC 入侵引脚的数量	3	3	3	3	3	O	3	O	3	O	3	O	3	O	3
LCD	O	O	O	O	O	O	O	O	O	O	-	-	-	-	-
USB FS	O[8]	O[8]	-	-	-	O	-	O	-	-	-	-	-	-	-
USART1	O	O	O	O	O[6]	O[6]	O[6]	O[6]	-	-	-	-	-	-	-
低功耗 UART (LPUART)	O	O	O	O	O[6]	O[6]	O[6]	O[6]	O[6]	O[6]	-	-	-	-	-

表 25. 功能取决于系统工作模式[1]（续）

外设	运行	睡眠	低功耗运行	低功耗睡眠	停止 0		停止 1		停止 2		待机		关断		VBAT
					-	唤醒能力	-	唤醒能力	-	唤醒能力	-	唤醒能力	-	唤醒能力	
I2C1	O	O	O	O	O[7]	O[7]	O[7]	O[7]	-	-	-	-	-	-	-
I2C3	O	O	O	O	O[7]	O[7]	O[7]	O[7]	O[7]	O[7]	-	-	-	-	-
SPIx（x=1、2）	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
SAI1	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
ADC1	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
VREFBUF	O	O	O	O	O	-	O	-	-	-	-	-	-	-	-
COMPx（x=1、2）	O	O	O	O	O	O	O	O	O	O	-	-	-	-	-
温度传感器	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
定时器 (TIMx)	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
低功耗定时器 1 (LPTIM1)	O	O	O	O	O	O	O	O	O	O	-	-	-	-	-
低功耗定时器 2 (LPTIM2)	O	O	O	O	O	O	O	O	-	-	-	-	-	-	-
独立看门狗 (IWDG)	O	O	O	O	O	O	O	O	O	O	O	O	-	-	-
窗口看门狗 (WWDG)	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
SysTick 定时器	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
触摸感应控制器 (TSC)	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
真随机数发生器 (RNG)	O[8]	O[8]	-	-	-	-	-	-	-	-	-	-	-	-	-
AES 硬件加速器	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
CRC 计算单元	O	O	O	O	-	-	-	-	-	-	-	-	-	-	-
IPCC	O	-	O	-	-	-	-	-	-	-	-	-	-	-	-
HSEM	O	-	O	-	-	-	-	-	-	-	-	-	-	-	-
GPIO	O	O	O	O	O	O	O	O	O	O	(9)	5 个引脚 (10)	(11)	5 个引脚 (10)	-

1. 图例：Y = 支持（使能）。O = 可选（默认情况下禁止。可通过软件使能）。R = 保留数据。
   - =不提供。灰色单元格表示唤醒功能。
2. 需要通过 PWR_CR3.RRS 位保留 SRAM2a 内容。
3. 当 PWR_CR3.RRS 位置 1 时，可以选择保留 SRAM2a 内容。
4. SRAM 时钟可门控打开或关闭。
5. 一些能够从停止模式唤醒的外设可请求将 HSI16 使能。在这种情况下，HSI16 由外设唤醒，并且仅响应请求它的外设。当外设不再需要 HSI16 时，HSI 将自动关闭。
6. UART 和 LPUART 接收是停止模式下的功能，并能在开始、地址匹配或接收到帧事件时产生一个唤醒中断。
7. I2C 地址检测是停止模式下的功能，并能在地址匹配时产生一个唤醒中断。

8. 电压调节仅限范围 1。
9. 待机模式下，I/O 可配置为内部上拉、下拉或浮空。
10. 具有从待机/关断唤醒能力的 I/O 为：PA0、PC13、PC12、PA2、PC5。
11. I/O 在关断模式中可配置为内部上拉、下拉或浮空，但是当退出关断模式时该配置会丢失。

调试模式

默认情况下，如果使用调试功能时应用程序将 MCU 置于停止 0、停止 1、停止 2、待机模式或关断模式，调试连接将丢失。这是因为 CPU1 内核不再受时钟驱动。

不过，通过设置 DBGMCU_CR 寄存器中的一些配置位，即使 MCU 进入低功耗模式，仍可使用软件对其进行调试。有关详细信息，请参见*第 41.4.3 节：调试和低功耗模式*。

6.4.1 运行模式

降低系统时钟速度

在运行模式下，可通过对预分频寄存器编程来降低系统时钟（SYSCLK、HCLK 和 PCLK）速度。进入睡眠模式之前，也可以使用这些预分频器降低外设速度。

有关详细信息，请参见*第 8.4.3 节：RCC 时钟配置寄存器 (RCC_CFGR)*。

外设时钟门控

在运行模式下，可随时停止各外设和存储器的 HCLK 和 PCLK 以降低功耗。

要进一步降低睡眠模式的功耗，可在执行 WFI 或 WFE 指令之前禁止外设时钟。

外设时钟门控由 RCC_AHBxENR 和 RCC_APBxENR 寄存器控制。

在睡眠模式下，复位 RCC_AHBxSMENR 和 RCC_APBxSMENR 寄存器中的对应位可以自动禁止外设时钟。

6.4.2 低功耗运行模式（LP 运行）

当系统处于停止模式时，要进一步降低功耗，可以在低功耗模式下配置稳压器。在此模式下，系统频率不应超过 2 MHz。无线电子系统不能用于低功耗运行模式。

有关稳压器和外设工作条件的详细信息，请参见产品数据手册。

低功耗运行模式下的 I/O 状态

在低功耗运行模式下，所有 I/O 引脚的状态与运行模式下相同。

进入低功耗运行模式

要进入低功耗运行模式，请执行下列操作：

1. 可选：跳转到 SRAM，并通过将 *PWR 控制寄存器 1 (PWR_CR1)* 和 *PWR CPU2 控制寄存器 1 (PWR_C2CR1)* 中的 FPDR 位置 1 使 Flash 掉电。
2. 将系统时钟频率降到 2 MHz 以下。
3. 将 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LPR 位置 1，强制寄存器进入低功耗模式。

有关如何进入低功耗运行模式，请参见*表 26：低功耗运行*。

退出低功耗运行模式

要退出低功耗运行模式，请执行下列操作（请参见*表 26*）：

1. 将 *PWR 控制寄存器 1 (PWR_CR1)* 中的 LPR 位清零，强制寄存器进入主模式。
2. 等待 *PWR 状态寄存器 2 (PWR_SR2)* 中的 REGLPF 位清零。
3. 增加系统时钟频率。

表 26. 低功耗运行

低功耗运行模式	说明
进入模式	将系统时钟频率降到 2 MHz 以下 LPR = 1
退出模式	LPR = 0 等待 REGLPF = 0 增加系统时钟频率
唤醒延迟	稳压器从低功耗模式唤醒的时间

6.4.3 进入低功耗模式

MCU 通过执行 WFI（等待中断）或 WFE（等待事件）指令进入低功耗模式，也可通过在从 ISR 返回时将 CPU1 系统控制寄存器中的 SLEEPONEXIT 位置 1 进入低功耗模式。
在没有中断或事件挂起时，才可通过 WFI 或 WFE 进入低功耗模式。

6.4.4 退出低功耗模式

MCU 根据进入睡眠模式和停止模式的方式退出这两种低功耗模式：

- 如果使用 WFI 指令或从 ISR 返回进入低功耗模式，通过 NVIC 确认的任何外设中断都能唤醒器件。
- 如果使用 WFE 指令进入低功耗模式，MCU 将在有事件发生时立即退出低功耗模式。唤醒事件可以通过 NVIC IRQ 中断或事件生成。
  - 在第一种情况下，当 CPU 系统控制寄存器中的 SEVONPEND = 0 时：在外设控制寄存器和 NVIC 中使能中断。当 MCU 从 WFE 恢复时，必须清零外设中断挂起位和 NVIC 外设 IRQ 通道挂起位（在 NVIC 中断清零挂起寄存器中）。只有当 NVIC 中断的优先级足够高时，才能唤醒和中断 MCU。
    当 CPU 系统控制寄存器中的 SEVONPEND = 1 时：在外设控制寄存器中使能中断，也可选择在 NVIC 中使能中断。当 MCU 从 WFE 恢复时，必须清零外设中断挂起位和使能的 NVIC 外设 IRQ 通道挂起位（在 NVIC 中断清零挂起寄存器中）。所有 NVIC 中断将唤醒 MCU，即使是禁止的亦是如此。只有当已使能 NVIC 中断的优先级足够高时，才能唤醒和中断 MCU。
  - 在第二种情况下，将一个 EXTI 线配置为事件模式。当 CPU 从 WFE 恢复时，因为对应事件线的挂起位没有被置 1，不必清零 EXTI 外设的中断挂起位或 NVIC IRQ 通道挂起位。可能需要清零相应外设中的中断标志。

MCU 通过外部复位（NRST 引脚）、IWDG 复位、已使能 WKUPx 引脚之一上的上升沿或者 RTC 事件（请参见*第 29 节：实时时钟 (RTC)*）或无线电事件（仅用于待机）来退出待机和关断低功耗模式。

从待机或关断模式唤醒后，程序将按照复位（启动引脚采样、选项字节加载和复位向量已获取等）后的方式重新执行。

CPU 从 CStop 模式唤醒时的系统模式可以由 *PWR 扩展状态和状态清除寄存器 (PWR_EXTSCR)* 中的 CnSTOPF 和 CnSBF 确定。

表 27. CPU CSTOP 唤醒与系统工作模式

系统模式	CPU1		CPU2		CPU1 唤醒	CPU2 唤醒
	C1SBF	C1STOPF	C2SBF	C2STOPF		
运行	0	0	0	0	从运行模式唤醒	从运行模式唤醒
	0	1	0	0	从停止模式唤醒，但由于 CPU2，系统已处于运行模式	从运行模式唤醒
	0	0	0	1	从运行模式唤醒	从停止模式唤醒，但由于 CPU1，系统已处于运行模式
	1	0	0	0	从待机模式唤醒，但由于 CPU2 的原因，系统已处于运行模式。	从运行模式唤醒
	0	0	1	0	从运行模式唤醒	从待机模式唤醒，但由于 CPU1 的原因，系统已处于运行模式。
	1	1	0	0	从待机模式唤醒，然后从停止模式唤醒，但由于 CPU2 的原因，系统已处于运行模式。	从运行模式唤醒
	0	0	1	1	从运行模式唤醒	从待机模式唤醒，然后从停止模式唤醒，但由于 CPU1 的原因，系统已处于运行模式。
停止	0	1	0	1	从停止模式唤醒（CPU2 仍处于 CSTOP）。	从停止模式唤醒（CPU1 仍处于 CSTOP）。
	1	1	0	1	在系统处于待机模式后从停止模式唤醒（CPU2 仍处于 CSTOP）。	从停止模式唤醒（CPU1 仍处于 CSTOP）。
	0	1	1	1	从停止模式唤醒（CPU2 仍处于 CSTOP）。	在系统处于待机模式后从停止模式唤醒（CPU1 仍处于 CSTOP）。
待机	1	0	1	0	从待机模式唤醒（CPU2 仍处于 CSTOP）。	从待机模式唤醒（CPU1 仍处于 CSTOP）。
N.A.	其他				无效，不会发生	

6.4.5 睡眠模式

睡眠模式下的 I/O 状态

在睡眠模式下，所有 I/O 引脚的状态与运行模式下相同。

进入睡眠模式

当 CPU 系统控制寄存器的 SLEEPDEEP 位清零时，将根据*进入低功耗模式*进入睡眠模式（请参见*表 28*）。

退出睡眠模式

MCU 按照*退出低功耗模式*所述从睡眠模式退出（请参见*表 28*）。

表 28. 睡眠模式

立即睡眠模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： – SLEEPDEEP = 0 – 没有中断（对于 WFI）或事件（对于 WFE）挂起 请参见 Cortex® 系统控制寄存器。
	从 ISR 返回，且： – SLEEPDEEP = 0 及 – SLEEPONEXIT = 1 – 没有中断挂起 请参见 Cortex® 系统控制寄存器。
退出模式	如果使用 WFI 或从 ISR 返回进入 中断：请参见*表 57：STM32WB55xx CPU1 向量表*和*表 58：STM32WB55xx CPU2 向量表* 如果使用 WFE 进入且 SEVONPEND = 0： 唤醒事件：请参见*第 13.4 节：中断列表* 如果使用 WFE 进入且 SEVONPEND = 1： 中断事件（即使在 NVIC 中禁止时）：请参见*表 57：STM32WB55xx CPU1 向量表*、*表 58：STM32WB55xx CPU2 向量表*或唤醒事件：请参见*第 13.4 节：中断列表*
唤醒延迟	无

6.4.6 低功耗睡眠模式（LP 睡眠）

有关稳压器和外设工作条件的详细信息，请参见产品数据手册。

低功耗睡眠模式下的 I/O 状态

在低功耗睡眠模式下，所有 I/O 引脚的状态与运行模式下相同。

进入低功耗睡眠模式

当 Cortex® 系统控制寄存器的 SLEEPDEEP 位清零时，将根据*第 6.4.3 节*所述从低功耗运行模式进入低功耗睡眠模式。

有关如何进入低功耗睡眠模式的详细信息，请参见*表 29*。

退出低功耗睡眠模式

根据*第 6.4.4 节*所述退出低功耗睡眠模式。当通过发出中断或事件来退出低功耗睡眠模式时，MCU 处于低功耗运行模式。

有关如何退出低功耗睡眠模式的详细信息，请参见*表 29*。

表 29. 低功耗睡眠

低功耗 立即睡眠模式	说明
进入模式	从低功耗运行模式进入低功耗睡眠模式。 WFI（等待中断）或 WFE（等待事件），且： – SLEEPDEEP = 0 – 没有中断（对于 WFI）或事件（对于 WFE）挂起 请参见 Cortex® 系统控制寄存器。
	从低功耗运行模式进入低功耗睡眠模式。 从 ISR 返回，且： – SLEEPDEEP = 0 及 – SLEEPONEXIT = 1 – 没有中断挂起 请参见 Cortex® 系统控制寄存器。
退出模式	如果使用 WFI 或从 ISR 返回进入 中断：请参见*表 57：STM32WB55xx CPU1 向量表*和*表 58：STM32WB55xx CPU2 向量表* 如果使用 WFE 进入且 SEVONPEND = 0： 唤醒事件：请参见*第 13.4 节：中断列表* 如果使用 WFE 进入且 SEVONPEND = 1： 中断（即使在 NVIC 中禁止时）：请参见*表 57：STM32WB55xx CPU1 向量表*和*表 58：STM32WB55xx CPU2 向量表* 唤醒事件：请参见*第 13.4 节：中断列表* 当退出低功耗睡眠模式时，MCU 会处于低功耗运行模式。
唤醒延迟	无

6.4.7 停止 0 模式

停止 0 模式基于 CPU 深度睡眠模式与外设时钟门控。稳压器配置为主稳压器模式。在停止 0 模式下，V_{CORE} 域中的所有时钟都会停止，PLL、MSI、HSI16 和 HSE 振荡器也被禁止。一些带有唤醒功能（I2Cx（x=1、3）、USART1 和 LPUART）的外设可以开启 HSI16 以接收帧，如果该帧不是唤醒帧，也可以在接收到帧后关闭 HSI16。在这种情况下，HSI16 时钟仅传播到请求该时钟的外设中。

SRAM1、SRAM2 和寄存器内容将保留。

停止 0 模式下的 BOR 始终可用。当使用的阈值高于 V_{BOR0} 时，功耗会增加。

停止 0 模式下的 I/O 状态

在停止 0 模式下，所有 I/O 引脚的状态与运行模式下相同。

进入停止 0 模式

当 Cortex 系统控制寄存器的 SLEEPDEEP 位置 1 时，将根据*第 6.4.3 节*进入停止 0 模式（请参见*表 30*）。

如果正在执行 Flash 编程，停止 0 模式的进入将延迟到操作完成后执行。

如果正在访问 APB 域，停止 0 模式的进入则延迟到 APB 访问结束后执行。

在停止 0 模式下，可以通过对各控制位进行编程来选择以下功能：

- 独立的看门狗 (IWDG)：IWDG 通过写入其密钥寄存器或使用硬件选项来启动。而且一旦启动便无法停止，除非复位。请参见*第 30.3 节：IWDG 功能说明*。
- 实时时钟 (RTC)：通过 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 RTCEN 位进行配置。
- 内部 RC 振荡器 (LSI)：通过 *RCC 控制/状态寄存器 (RCC_CSR)* 中的 LSIxON 位进行配置。
- 外部 32.768 kHz 振荡器 (LSE)：通过 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 LSEON 位进行配置。

在停止 0 模式下可以使用多个外设，当这些外设已使能并且由 LSI 或 LSE 提供时钟，或者外设请求 HSI16 时钟 LCD、LPTIM1、LPTIM2、I2Cx（x=1、3）、USART1、LPUART 时，会增加功耗。

在停止 0 模式下，可以使用比较器、PVMx（x=1、3）和 PVD。如果不需要上述功能，可以通过软件禁止功能，以节省其功耗。

在停止 0 模式期间，ADC、温度传感器和 VREFBUF 缓冲区会产生功耗，除非在进入该模式前将它们禁止。

退出停止 0 模式

根据*第 6.4.4 节*所述内容退出停止 0 模式。

有关如何退出停止 0 模式的详细信息，请参见*表 30*。

通过发出中断或唤醒事件退出停止 0 模式时，如果 *RCC 时钟配置寄存器 (RCC_CFGR)* 中的 STOPWUCK 位置 1，则选择 HSI16 振荡器作为系统时钟。如果 STOPWUCK 位清零，则选择 MSI 振荡器作为系统时钟。当选择 HSI16 作为唤醒系统时钟时，唤醒时间更短。选择 MSI 能够以更高频率进行唤醒操作，最高 48 MHz。

当稳压器在低功耗模式下工作时，将器件从使用 HSI16 的停止 0 模式唤醒将需要额外的启动延时。在停止 0 模式下一直开启内部稳压器虽然会增大功耗，但可以缩短启动时间。

当退出停止 0 模式时，MCU 处于运行模式（范围 1 或范围 2，具体取决于 *PWR 控制寄存器 1 (PWR_CR1)* 中的 VOS 位），如果同一寄存器中的 LPR 位置 1，MCU 会处于低功耗运行模式。

表 30. 停止 0 模式

停止 0 模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： – 将 Cortex 系统控制寄存器中的 SLEEPDEEP 位置 1 – 没有中断（对于 WFI）或事件（对于 WFE）挂起 – PWR_CR1 和/或 PWR_C2CR1 中的 LPMS =“000”或更高值
	从 ISR 返回，且： – 将 Cortex 系统控制寄存器中的 SLEEPDEEP 位置 1 – SLEEPONEXIT = 1 – 没有中断挂起 – PWR_CR1 和/或 PWR_C2CR1 中的 LPMS =“000”或更高值
	注： 要进入停止 0 模式，必须清零所有 EXTI 线挂起位（EXTI 挂起寄存器 (EXTI_PR1) 和 EXTI 挂起寄存器 (EXTI_PR2) 中）和生成唤醒中断的外设标志。否则将忽略进入停止 0 模式这一过程，继续执行程序。
退出模式	如果使用 WFI 或从 ISR 返回进入 任何配置为中断模式的 EXTI 线（必须在 NVIC 中使能对应的 EXTI 中断向量）。中断源可以是外部中断或具有唤醒功能的外设。请参见*表 57：STM32WB55xx CPU1 向量表*和*表 58：STM32WB55xx CPU2 向量表*。 如果使用 WFE 进入且 SEVONPEND = 0： 任何配置为事件模式的 EXTI 线。请参见*第 13.4 节：中断列表*。 如果使用 WFE 进入且 SEVONPEND = 1： 任何配置为中断模式的 EXTI 线（即使在 NVIC 中禁止对应的 EXTI 中断向量）。中断源可以是外部中断或具有唤醒功能的外设。请参见*表 57：STM32WB55xx CPU1 向量表*和*表 58：STM32WB55xx CPU2 向量表*。 唤醒事件：请参见*第 13.4 节：中断列表*
唤醒延迟	最长唤醒时间介于 MSI 或 HSI16 唤醒时间和 Flash 从停止 0 模式唤醒的时间之间。

6.4.8 停止 1 模式

在停止 1 模式下，主稳压器关闭，仅低功率稳压器开启，除此之外与停止 0 模式相同。可以从运行模式和低功耗运行模式进入停止 1 模式。

有关如何进入和退出停止 1 模式的详细信息，请参见*表 31*。

表 31. 停止 1 模式

停止 1 模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： – 将 Cortex 系统控制寄存器中的 SLEEPDEEP 位置 1 – 没有中断（对于 WFI）或事件（对于 WFE）挂起 – PWR_CR1 和/或 PWR_C2CR1 中的 LPMS =“001”或更高值
	从 ISR 返回，且： – 将 Cortex 系统控制寄存器中的 SLEEPDEEP 位置 1 – SLEEPONEXIT = 1 – 没有中断挂起 – PWR_CR1 和/或 PWR_C2CR1 中的 LPMS =“001”或更高值
	注：　要进入停止 1 模式，必须清零所有 EXTI 线挂起位（在 EXTI 挂起寄存器 (EXTI_PR1) 和 EXTI 挂起寄存器 (EXTI_PR2) 中）和生成唤醒中断的外设标志。否则将忽略进入停止 1 模式这一过程，继续执行程序。
退出模式	如果使用 WFI 或从 ISR 返回进入 任何配置为中断模式的 EXTI 线（必须在 NVIC 中使能对应的 EXTI 中断向量）。中断源可以是外部中断或具有唤醒功能的外设。请参见*表 57：STM32WB55xx CPU1 向量表*和*表 58：STM32WB55xx CPU2 向量表*。 如果使用 WFE 进入且 SEVONPEND = 0： 任何配置为事件模式的 EXTI 线。请参见*第 13.4 节：中断列表*。 如果使用 WFE 进入且 SEVONPEND = 1： 任何配置为中断模式的 EXTI 线（即使在 NVIC 中禁止对应的 EXTI 中断向量）。中断源可以是外部中断或具有唤醒功能的外设。请参见*表 57：STM32WB55xx CPU1 向量表*和*表 58：STM32WB55xx CPU2 向量表*。 唤醒事件：请参见*第 13.4 节：中断列表*
唤醒延迟	最长唤醒时间介于 MSI 或 HSI16 唤醒时间和稳压器从低功耗模式唤醒的时间 + Flash 从停止 1 模式唤醒的时间之间。

6.4.9 停止 2 模式

停止 2 模式基于 CPU 深度睡眠模式与外设时钟门控。在停止 2 模式下，V_{CORE} 域中的所有时钟都会停止，PLL、MSI、HSI16 和 HSE 振荡器也被禁止。一些带有唤醒能力（I2C3 和 LPUART）的外设可以开启 HSI16 以获取帧，如果该帧不是唤醒帧，也可以在接收到帧后关闭 HSI16。在这种情况下，HSI16 时钟仅传播到请求该时钟的外设中。

SRAM1、SRAM2 和寄存器内容将保留。

停止 2 模式下 BOR 始终可用。当使用的阈值高于 V_{BOR0} 时，功耗会增加。

注：　停止 2 模式期间，会强制比较器、LPUART 输出和 LPTIM1 输出变为低速 (OSPEEDy = 00)。

停止 2 模式下的 I/O 状态

在停止 2 模式下，所有 I/O 引脚的状态与运行模式下相同。

进入停止 2 模式

当 Cortex 系统控制寄存器的 SLEEPDEEP 位置 1 时，将根据*第 6.4.3 节：进入低功耗模式*所述进入停止 2 模式（请参见*表 32*）。

仅可以从运行模式进入停止 2 模式。无法从低功耗运行模式进入停止 2 模式。

如果正在执行 Flash 编程，停止 2 模式的进入将延迟到存储器访问结束后执行。

如果正在访问 APB 域，停止 2 模式的进入则延迟到 APB 访问结束后执行。

在停止 2 模式下，可以通过对各控制位进行编程来选择以下功能：

- 独立的看门狗 (IWDG)：IWDG 通过写入其密钥寄存器或使用硬件选项来启动。而且一旦启动便无法停止，除非复位。请参见*第 30.3 节：IWDG 功能说明*。
- 实时时钟 (RTC)：通过 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 RTCEN 位进行配置。
- 内部 RC 振荡器 (LSI)：通过 *RCC 控制/状态寄存器 (RCC_CSR)* 中的 LSIxON 位进行配置。
- 外部 32.768 kHz 振荡器 (LSE)：通过 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 LSEON 位进行配置。

在停止 2 模式下可以使用多个外设，当这些外设已使能并且由 LSI 或 LSE 提供时钟，或者外设请求 HSI16 时钟 LCD、LPTIM1、I2C3 和 LPUART 时，会增加功耗。

在停止 2 模式下，可以使用比较器、PVMx（x=1、3）和 PVD。如果不需要上述功能，可以通过软件禁止功能，以节省其功耗。

在停止 2 模式期间 ADC、温度传感器和 VREFBUF[(a)] 缓冲区会产生功耗，除非在进入该模式前将它们禁止。

必须通过清零外设自身的使能位来禁止无法在停止 2 模式下使能的所有外设，或者通过将以下寄存器中的相应位置 1 使其处于复位状态。

- *RCC AHB1 外设复位寄存器 (RCC_AHB1RSTR)*
- *RCC AHB2 外设复位寄存器 (RCC_AHB2RSTR)*
- *RCC AHB3 和 AHB4 外设复位寄存器 (RCC_AHB3RSTR)*
- *RCC APB1 外设复位寄存器 1 (RCC_APB1RSTR1)*
- *RCC APB1 外设复位寄存器 2 (RCC_APB1RSTR2)*
- *RCC APB2 外设复位寄存器 (RCC_APB2RSTR)*
- *RCC APB3 外设复位寄存器 (RCC_APB3RSTR)*

a. 仅适用于第 3 类器件。

退出停止 2 模式

根据*第 6.4.4 节：退出低功耗模式*退出停止 2 模式（请参见*表 32*）。

通过发出中断或唤醒事件退出停止 2 模式时，如果 *RCC 时钟配置寄存器 (RCC_CFGR)* 中的 STOPWUCK 位置 1，则选择 HSI16 振荡器作为系统时钟。如果 STOPWUCK 位清零，则选择 MSI 振荡器作为系统时钟。当选择 HSI16 作为唤醒系统时钟时，唤醒时间更短。选择 MSI 能够以更高频率进行唤醒操作，最高 48MHz。

退出停止 2 模式后，MCU 处于运行模式（范围 1 或范围 2，具体取决于 PWR_CR1 中的 VOS 位）。

表 32. 停止 2 模式

停止 2 模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： – 将 Cortex 系统控制寄存器中的 SLEEPDEEP 位置 1 – 没有中断（对于 WFI）或事件（对于 WFE）挂起 – PWR_CR1 和/或 PWR_C2CR1 中的 LPMS = “010”或更高值
	从 ISR 返回，且： – 将 Cortex 系统控制寄存器中的 SLEEPDEEP 位置 1 – SLEEPONEXIT = 1 – 没有中断挂起 – PWR_CR1 和/或 PWR_C2CR1 中的 LPMS = “010”或更高值 *注：　要进入停止 2 模式，必须清零所有 EXTI 线挂起位（在 EXTI 挂起寄存器 (EXTI_PR1) 和 EXTI 挂起寄存器 (EXTI_PR2) 中）和生成唤醒中断的外设标志。否则将忽略进入停止模式这一过程，继续执行程序。*
退出模式	如果使用 WFI 或从 ISR 返回进入： 任何配置为中断模式的 EXTI 线（必须在 NVIC 中使能对应的 EXTI 中断向量）。中断源可以是外部中断或具有唤醒功能的外设。请参见*表 57：STM32WB55xx CPU1 向量表*和*表 58：STM32WB55xx CPU2 向量表*。 如果使用 WFE 进入且 SEVONPEND = 0： 任何配置为事件模式的 EXTI 线。请参见*第 13.4 节：中断列表*。 如果使用 WFE 进入且 SEVONPEND = 1： 任何配置为中断模式的 EXTI 线（即使在 NVIC 中禁止对应的 EXTI 中断向量）。中断源可以是外部中断或具有唤醒功能的外设。请参见*表 57：STM32WB55xx CPU1 向量表*和*表 58：STM32WB55xx CPU2 向量表*。 任何配置为事件模式的 EXTI 线。请参见*第 13.4 节：中断列表*。
唤醒延迟	最长唤醒时间介于 MSI 或 HSI16 唤醒时间和稳压器从低功耗模式唤醒的时间 + Flash 从停止 2 模式唤醒的时间之间。

6.4.10 待机模式

待机模式下可以使用 BOR 达到最低功耗。待机模式基于 CPU 深度睡眠模式，其中稳压器被禁止（保留 SRAM2 内容时除外）。PLL、HSI16、MSI 和 HSE 振荡器也会关闭。

除备份域和待机电路中的寄存器外，SRAM1 和寄存器内容都将丢失（请参见*图 7*）。当 *PWR 控制寄存器 3 (PWR_CR3)* 中的 RRS 位置 1 时，SRAM2 内容保留。在这种情况下，低功耗稳压器开启并仅为 SRAM2 供电。

在待机模式下 BOR 始终可用。当使用的阈值高于 V_{BOR0} 时，功耗会增加。

待机模式下的 I/O 状态

在待机模式下，I/O 可配置为上拉（请参见 PWR_PUCRx 寄存器（x=A、B、C、D、E、F、G、H））或者下拉（请参见 PWR_PDCRx 寄存器（x=A、B、C、D、E、F、G、H）），或者保持模拟状态。

待机模式下可以使用 PC13 的 RTC 输出功能。也可以为 LSE 使用 PC14 和 PC15 功能。5 个唤醒引脚（WKUPx，x=1、2...5）和 3 个 RTC 入侵引脚可用。

进入待机模式

当 Cortex 系统控制寄存器的 SLEEPDEEP 位置 1 时，将根据*第 6.4.3 节：进入低功耗模式*进入待机模式。

有关如何进入待机模式的详细信息，请参见*表 33：待机模式*。

在待机模式下，可以通过对各控制位进行编程来选择以下功能：

- 独立的看门狗 (IWDG)：IWDG 通过写入其密钥寄存器或使用硬件选项来启动。而且一旦启动便无法停止，除非复位。请参见*第 30.3 节：IWDG 功能说明*。
- 实时时钟 (RTC)：通过备份域控制寄存器 (RCC_BDCR) 中的 RTCEN 位进行配置
- 内部 RC 振荡器 (LSI)：通过控制/状态寄存器 (RCC_CSR) 中的 LSIxON 位进行配置。
- 外部 32.768 kHz 振荡器 (LSE)：通过备份域控制寄存器 (RCC_BDCR) 中的 LSEON 位进行配置

退出待机模式

根据*第 6.4.4 节：退出低功耗模式*退出待机模式。*PWR 控制寄存器 3 (PWR_CR3)* 中的 SBF 状态标志指示 MCU 已处于待机模式。从待机模式唤醒后，除 *PWR 控制寄存器 3 (PWR_CR3)* 外，所有寄存器都将复位。

有关如何退出待机模式的详细信息，请参见*表 33：待机模式*。

表 33. 待机模式

待机模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： – 将 Cortex 系统控制寄存器中的 SLEEPDEEP 位置 1 – 没有中断（对于 WFI）或事件（对于 WFE）挂起 – PWR_CR1 和/或 PWR_C2CR1 中的 LPMS =“011”或更高值 – 清零功率状态寄存器 1 (PWR_SR1) 中的 WUFx 位
	从 ISR 返回，且： – 将 Cortex 系统控制寄存器中的 SLEEPDEEP 位置 1 – SLEEPONEXIT = 1 – 没有中断挂起 – PWR_CR1 和/或 PWR_C2CR1 中的 LPMS =“011”或更高值 – 清零功率状态寄存器 1 (PWR_SR1) 中的 WUFx 位 将与所选唤醒源（RTC 闹钟 A、RTC 闹钟 B、RTC 唤醒、入侵或时间戳标志）对应的 RTC 标志清零
退出模式	WKUPx 引脚边沿、RTC 事件、NRST 引脚的外部复位、IWDG 复位、BOR 复位
唤醒延迟	复位阶段

6.4.11 关断模式

关断模式下可达到最低功耗。待机模式基于深度睡眠模式，其中稳压器被禁止。因此 V_{CORE} 域断电。PLL、HSI16、MSI、LSI 和 HSE 振荡器也会关闭。

除备份域中的寄存器之外，SRAM1、SRAM2 和寄存器的内容都将丢失。在关断模式中 BOR 不可用。在此模式中电源电压监测不可用，因此不支持备份域的开启。

关断模式下的 I/O 状态

在关断模式下，I/O 可配置为上拉（请参见 PWR_PUCRx 寄存器（x=A、B、C、D、E、F、G、H））或者下拉（请参见 PWR_PDCRx 寄存器（x=A、B、C、D、E、F、G、H）），或者保持模拟状态。

但是，由于上电复位，退出关断模式时此配置会丢失。

关断模式下可以使用 PC13 的 RTC 输出功能。也可以为 LSE 使用 PC14 和 PC15 功能。5 个唤醒引脚（WKUPx，x=1、2...5）和 3 个 RTC 入侵引脚可用。

进入关断模式

当 Cortex 系统控制寄存器的 SLEEPDEEP 位置 1 时，将根据*进入低功耗模式*进入关断模式。

有关如何进入关断模式的详细信息，请参见*表 34：关断模式*。

在关断模式下，可以通过对各控制位进行编程来选择以下功能：

- 实时时钟 (RTC)：通过备份域控制寄存器 (RCC_BDCR) 中的 RTCEN 位进行配置。注意事项：如果 VDD 断电，则 RTC 内容丢失。
- 外部 32.768 kHz 振荡器 (LSE)：通过备份域控制寄存器 (RCC_BDCR) 中的 LSEON 位进行配置

退出关断模式

根据*退出低功耗模式*退出关断模式。从关断模式退出时会发生上电复位。从关断模式唤醒后会复位所有寄存器（除了备份域中的寄存器）。

有关如何退出关断模式的详细信息，请参见*表 34：关断模式*。

表 34. 关断模式

关断模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： – 将 Cortex 系统控制寄存器中的 SLEEPDEEP 位置 1 – 没有中断（对于 WFI）或事件（对于 WFE）挂起 – PWR_CR1 和 PWR_C2CR1 中的 LPMS = “1XX” – 清零功率状态寄存器 1 (PWR_SR1) 中的 WUFx 位
	从 ISR 返回，且： – 将 Cortex 系统控制寄存器中的 SLEEPDEEP 位置 1 – SLEEPONEXT = 1 – 没有中断挂起 – PWR_CR1 和 PWR_C2CR1 中的 LPMS = “1XX” – 清零功率状态寄存器 1 (PWR_SR1) 中的 WUFx 位 将与所选唤醒源（RTC 闹钟 A、RTC 闹钟 B、RTC 唤醒、入侵或时间戳标志）对应的 RTC 标志清零
退出模式	WKUPx 引脚边沿、RTC 事件、NRST 引脚的外部复位
唤醒延迟	复位阶段

6.4.12 从低功耗模式自动唤醒

RTC 用于在没有外部中断的情况下从低功耗模式唤醒 MCU（自动唤醒模式）。RTC 提供了可编程时基，便于定期从 Stop（0、1 或 2）或待机模式唤醒器件。为此，通过对 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 RTCSEL[1:0] 位进行编程，可以选择三个复用 RTC 时钟源中的其中两个：

- 低功耗 32.768 kHz 外部晶振 (LSE OSC)
  该时钟源能够以极低的功耗提供精确时基。
- 低功耗内部 RC 振荡器 (LSI)
  此时钟源的优势在于可以节省 32.768 kHz 晶振的成本。此内部 RC 振荡器非常省电。

要通过 RTC 闹钟事件从停止模式唤醒，必须：

- 将 EXTI 线 18 配置为上升沿有效
- 配置 RTC 以生成 RTC 闹钟

要从待机模式唤醒，无需配置 EXTI 线 18。

要通过 RTC 唤醒事件从停止模式唤醒，必须：

- 将 EXTI 线 20 配置上升沿有效
- 配置 RTC 以生成 RTC 闹钟

要从待机模式唤醒，则不需要配置 EXTI 线 20。

LCD 帧起始中断也可用作从停止（0、1 或 2）模式定期唤醒的信号。在待机模式下 LCD 不可用。

LCD 时钟源自 RTCSEL[1:0] 选择的 RTC 时钟。

6.5 实时无线电信息

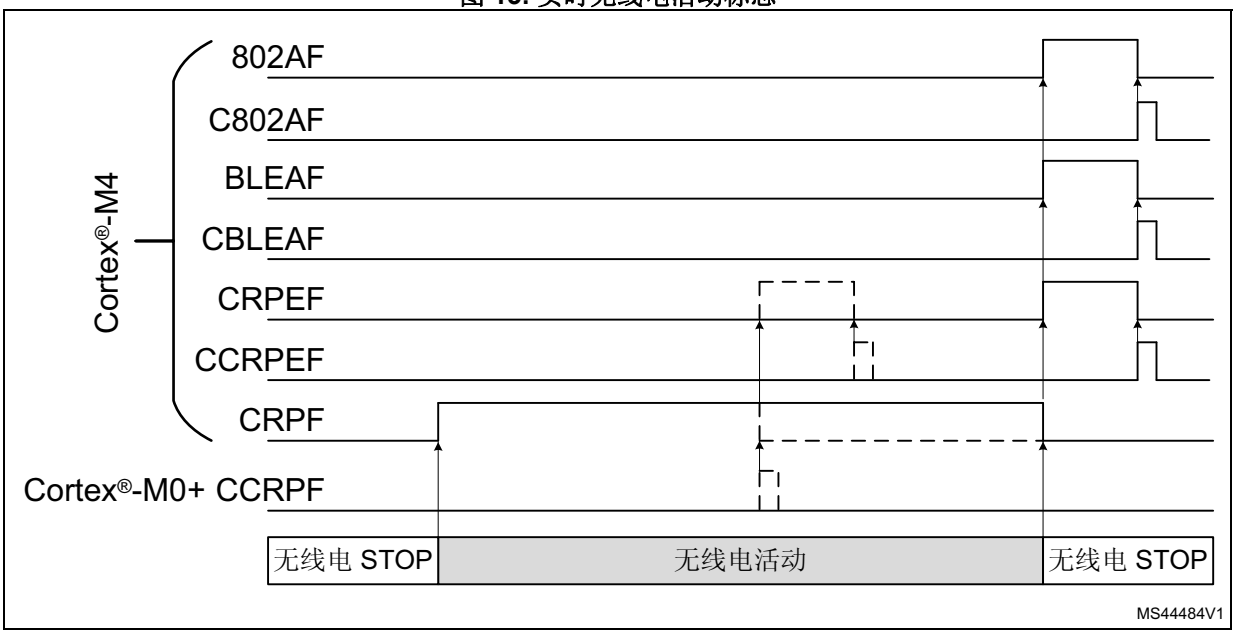
PWR 提供相应标志，指示无线电的实时运行情况：

- IEEE802.15.4 无线电活动结束中断标志
- BLE 无线电活动结束中断标志
- 关键无线电阶段标志
- 关键无线电阶段结束中断标志

CPU1 可以使用这些标志来确定无线电活动。

不同标志的基本时序关系如*图 12* 所示。

图 13. 实时无线电活动标志



使能时，无线电活动结束中断标志 802AF 指示 IEEE802.15.4 无线电活动周期结束，该标志在无线电进入 Cstop 模式时产生。

使能时，无线电活动结束中断标志 BLEAF 指示 BLE 无线电活动周期结束，该标志在无线电进入 Cstop 模式时产生。

关键无线电阶段标志指示无线电的关键实时阶段，其中不得通过擦除或编程操作来阻止对 Flash 的访问（所有擦除和编程操作都由 Flash 接口中的 CPU2 暂停）。关键无线电阶段的结束可以由 CPU2 清零关键无线电阶段标志 CRPF 来触发，并且最终在 IEEE802.15.4 或 BLE 无线电活动结束时清除。

使能时，关键无线电阶段结束中断标志 CRPEF 指示无线电关键阶段结束，该标志在无线电关键阶段结束时产生。

6.6 PWR 寄存器

外设寄存器可支持半字（16 位）或字（32 位）访问。

6.6.1 PWR 控制寄存器 1 (PWR_CR1)

PWR control register 1

偏移地址：0x000

复位值：0x0000 0200 从待机模式唤醒后，此寄存器会复位，但位 [2:0] 除外。

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

位 31:15 保留，必须保持复位值。

位 14 **LPR**：低功耗运行 (Low-power run)
当此位置 1 时，稳压器会从主模式 (MR) 切换为低功耗模式 (LPR)。
注： 当 LPR 位置 1 时无法进入停止 2 模式，而是会进入停止 1 模式。

位 13:11 保留，必须保持复位值。

位 10:9 **VOS**：电压调节范围选择 (Voltage scaling range selection)
00：无法写入（由硬件禁止）
01：范围 1
10：范围 2
11：无法写入（由硬件禁止）

位 8 **DBP**：禁止备份域写保护 (Disable backup domain write protection)
在复位状态下，RTC 和备份寄存器均受到保护，以防止寄生写访问。必须将此位置 1 才能使能对这些寄存器的写访问。
0：禁止对 RTC 和备份寄存器的访问
1：使能对 RTC 和备份寄存器的访问

位 7:6 保留，必须保持复位值。

位 5 **FPDS**：CPU1 处于 LPSleep 期间 Flash 的掉电模式 (Flash memory power down mode during LPSleep for CPU1)

该位选择当 CPU 均处于睡眠模式时，Flash 是处于掉电模式还是空闲模式。只有当系统处于 LPSleep 模式且 CPU2 的 PWR_C2CR1.FPDS 位也允许时，Flash 才能设置为掉电模式。

0：当系统处于 LPSleep 模式时，Flash 处于空闲模式

1：当系统处于 LPSleep 模式时，Flash 处于掉电模式

位 4 **FPDR**：CPU1 处于 LPRun 期间 Flash 的掉电模式 (Flash memory power down mode during LPRun for CPU1)

只有通过首先将（代码 0xC1B0）写入该寄存器（写入代码时，寄存器位不会更新）解锁该寄存器位后，才能向该位写入 1。选择当处于 LPRun 模式时，Flash 是处于掉电模式还是空闲模式。（只有从 SRAM 执行代码时，Flash 才能处于掉电模式）。只有当系统处于 LPRun 模式且 CPU2 的 PWR_C2CR1.FPDR 位也允许时，Flash 才能设置为掉电模式。

0：当系统处于 LPRun 模式时，Flash 处于空闲模式

1：当系统处于 LPRun 模式时，Flash 处于掉电模式

位 3 保留，必须保持复位值。

位 2:0 **LPMS[2:0]**：CPU1 的低功耗模式选择 (Low-power mode selection for CPU1)

退出待机模式时，这些位不会复位。

当 CPU1 进入深度睡眠模式时，使用这些位选择允许的低功耗模式。进入的系统低功耗模式还取决于 CPU2 的 PWR_C2CR1.LPMS 允许的低功耗模式。

000：停止 0 模式

001：停止 1 模式

010：停止 2 模式

011：待机模式

1xx：关断模式

注：　如果 LPR 位置 1，则无法选择停止 2 模式，因此会进入停止 1 模式而不是停止 2 模式。在待机模式下，SRAM2 内容可以保留或丢失，具体取决于 PWR_CR3 中的 RRS 位。

6.6.2 PWR 控制寄存器 2 (PWR_CR2)

PWR control register 2

偏移地址：0x004

复位值：0x0000 0000。退出待机模式时，寄存器会复位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	USV	Res.	Res.	Res.	PVME3	Res	PVME1	PLS[2:0]			PVDE
					rw				rw		rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10 **USV**：V_{DDUSB} USB 电源有效 (V_{DDUSB} USB supply valid)
此位用于验证 V_{DDUSB} 电源是否已应用电气和逻辑隔离。必须将此位置 1 才能使用 USB FS 外设。如果 V_{DDUSB} 并非始终存在于应用中，则可以使用 PVM 确定此电源是否就绪。
0：V_{DDUSB} 不存在。应用逻辑和电气隔离以忽略此电源。
1：V_{DDUSB} 有效。

位 9:7 保留，必须保持复位值。

位 6 **PVME3**：外设电压监测 3 使能：V_{DDA} 与 1.62V (Peripheral voltage monitoring 3 enable: V_{DDA} vs. 1.62V)
0：PVM3（以 1.62V 阈值监测 V_{DDA}）禁止。
1：PVM3（以 1.62V 阈值监测 V_{DDA}）使能。

位 5 保留，必须保持复位值。

位 4 **PVME1**：外设电压监测 1 使能：V_{DDUSB} 与 1.2V (Peripheral voltage monitoring 1 enable: V_{DDUSB} vs. 1.2V)
0：PVM1（以 1.2V 阈值监测 V_{DDUSB}）禁止。
1：PVM1（以 1.2V 阈值监测 V_{DDUSB}）使能。

位 3:1 **PLS[2:0]**：电源电压检测器电平选择 (Power voltage detector level selection)
这些位用于选择电源电压检测器检测的电压阈值：
000：约 2.0 V 的 V_{PVD0}
001：约 2.2 V 的 V_{PVD1}
010：约 2.4 V 的 V_{PVD2}
011：约 2.5 V 的 V_{PVD3}
100：约 2.6 V 的 V_{PVD4}
101：约 2.8 V 的 V_{PVD5}
110：约 2.9 V 的 V_{PVD6}
111：外部输入模拟电压 PVD_IN（内部与 VREFINT 进行比较）
注：　当 SYSCFG_CBR 寄存器中的 PVDL（PVD 锁定）位置 1 时，这些位受到写保护。这些位仅由系统复位进行复位。

位 0 **PVDE**：电源电压检测器使能 (Power voltage detector enable)
0：禁止电源电压检测器。
1：使能电源电压检测器
注：　当 SYSCFG_CBR 寄存器中的 PVDL（PVD 锁定）位置 1 时，该位受到写保护。此位仅由系统复位进行复位。

6.6.3 PWR 控制寄存器 3 (PWR_CR3)

PWR control register 3

偏移地址：0x008

复位值：0x0000 8000。退出待机模式后，此寄存器不会复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIWUL	EC2H	E802A	EBLEA	ECRPE	APC	RRS	EBORHSMPSFB	Res.	Res.	Res.	EWUP5	EWUP4	EWUP3	EWUP2	EWUP1
rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **EIWUL**：使能 CPU1 的内部唤醒线 (Enable internal wakeup line for CPU1)

0：禁止 CPU1 的内部唤醒线中断。

1：使能 CPU1 的内部唤醒线中断。

位 14 **EC2H**：使能 CPU1 的 CPU2 保持中断 (Enable CPU2 Hold interrupt for CPU1)

使能 CPU1 的 CPU2 保持（由于 C2BOOT）中断。

0：禁止 CPU1 中断。

1：使能 CPU1 中断。

位 13 **E802A**：使能 CPU1 的 802.15.4 活动结束中断 (Enable 802.15.4 end of activity interrupt for CPU1)

0：禁止 CPU1 中断。

1：使能 CPU1 中断。

位 12 **EBLEA**：使能 CPU1 的 BLE 活动结束中断 (Enable BLE end of activity interrupt for CPU1)

0：禁止 CPU1 中断。

1：使能 CPU1 中断。

位 11 **ECRPE**：使能 CPU1 的关键无线电阶段活动结束中断 (Enable critical radio phase end of activity interrupt for CPU1)

0：禁止 CPU1 中断。

1：使能 CPU1 中断。

位 10 **APC**：从 CPU1 应用上拉和下拉配置 (Apply pull-up and pull-down configuration from CPU1)

当 CPU1 的该位或 CPU2 的 PWR_C2CR3.APC 位置 1 时，将应用 PWR_PUCRx 和 PWR_PDCRx 寄存器中定义的 I/O 上拉和下拉配置。当两个位清零时，PWR_PUCRx 和 PWR_PDCRx 寄存器不会应用于 I/O。

位 9 **RRS**：待机模式下 SRAM2a 内容保留 (SRAM2a retention in Standby mode)

0：待机模式下 SRAM2a 断电（SRAM2a 内容丢失）。

1：待机模式下使用低功耗稳压器接通 SRAM2a（SRAM2a 内容保留）。

位 8 **EBORHSMPSFB**：使能 CPU1 的强制 BORH 和 SMPS 降压转换器进入旁路模式中断 (Enable BORH and SMPS step-down converter forced in Bypass interrupts for CPU1)

0：禁止 CPU1 的中断 BORHF 和 SMPSFBF。

1：使能 CPU1 的中断 BORHF 和 SMPSFBF。

位 7:5 保留，必须保持复位值。

位 4 **EWUP5**：使能 CPU1 的唤醒引脚 WKUP5 (Enable Wakeup pin WKUP5 for CPU1)
该位置 1 时，会使能外部唤醒引脚 WKUP5，并在 CPU1 出现上升沿或下降沿时触发中断和从停止、待机或关断事件唤醒。通过 *PWR 控制寄存器 4 (PWR_CR4)* 中的 WP5 位配置有效边沿。

位 3 **EWUP4**：使能 CPU1 的唤醒引脚 WKUP4 (Enable Wakeup pin WKUP4 for CPU1)
该位置 1 时，会使能外部唤醒引脚 WKUP4，并在 CPU1 出现上升沿或下降沿时触发中断和从停止、待机或关断事件唤醒。通过 *PWR 控制寄存器 4 (PWR_CR4)* 中的 WP4 位配置有效边沿。

位 2 **EWUP3**：使能 CPU1 的唤醒引脚 WKUP3 (Enable Wakeup pin WKUP3 for CPU1)
该位置 1 时，会使能外部唤醒引脚 WKUP3，并在 CPU1 出现上升沿或下降沿时触发中断和从停止、待机或关断事件唤醒。通过 *PWR 控制寄存器 4 (PWR_CR4)* 中的 WP3 位配置有效边沿。

位 1 **EWUP2**：使能 CPU1 的唤醒引脚 WKUP2 (Enable Wakeup pin WKUP2 for CPU1)
该位置 1 时，会使能外部唤醒引脚 WKUP2，并在 CPU1 出现上升沿或下降沿时触发中断和从停止、待机或关断事件唤醒。通过 *PWR 控制寄存器 4 (PWR_CR4)* 中的 WP2 位配置有效边沿。

位 0 **EWUP1**：使能 CPU1 的唤醒引脚 WKUP1 (Enable Wakeup pin WKUP1 for CPU1)
该位置 1 时，会使能外部唤醒引脚 WKUP1，并在 CPU1 出现上升沿或下降沿时触发中断和从停止、待机或关断事件唤醒。通过 *PWR 控制寄存器 4 (PWR_CR4)* 中的 WP1 位配置有效边沿。

6.6.4 PWR 控制寄存器 4 (PWR_CR4)

PWR control register 4

偏移地址：0x00C

复位值：0x0000 0000。退出待机模式后，此寄存器不会复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2BOOT	Res.	Res.	Res.	Res.	Res.	VBRS	VBE	Res.	Res.	Res.	WP5	WP4	WP3	WP2	WP1
rw						rw	rw				rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **C2BOOT**：复位或从停止或待机模式唤醒后启动 CPU2。
0：复位或从停止或待机模式唤醒后不会启动 CPU2。
1：从停止或待机模式唤醒后将启动 CPU2。

位 14:10 保留，必须保持复位值。

位 9 **VBRS**：V_{BAT} 电池充电电阻选择 (VBAT battery charging resistor selection)
0：通过 5 kΩ 电阻为 V_{BAT} 充电
1：通过 1.5 kΩ 电阻为 V_{BAT} 充电

位 8 **VBE**：V_{BAT} 电池充电使能 (VBAT battery charging enable)
0：禁止 V_{BAT} 电池充电
1：使能 V_{BAT} 电池充电。

位 7:5 保留，必须保持复位值。

位 4 **WP5**：唤醒引脚 WKUP5 极性 (Wakeup pin WKUP5 polarity)
该位用于定义在外部唤醒引脚 WKUP5 上进行事件检测所用的极性
0：在高电平（上升沿）检测
1：在低电平（下降沿）检测

位 3 **WP4**：唤醒引脚 WKUP4 极性 (Wakeup pin WKUP4 polarity)
该位用于定义在外部唤醒引脚 WKUP4 上进行事件检测所用的极性
0：在高电平（上升沿）检测
1：在低电平（下降沿）检测

位 2 **WP3**：唤醒引脚 WKUP3 极性 (Wakeup pin WKUP3 polarity)
该位用于定义在外部唤醒引脚 WKUP3 上进行事件检测所用的极性
0：在高电平（上升沿）检测
1：在低电平（下降沿）检测

位 1 **WP2**：唤醒引脚 WKUP2 极性 (Wakeup pin WKUP2 polarity)
该位用于定义在外部唤醒引脚 WKUP2 上进行事件检测所用的极性
0：在高电平（上升沿）检测
1：在低电平（下降沿）检测

位 0 **WP1**：唤醒引脚 WKUP1 极性 (Wakeup pin WKUP1 polarity)
该位用于定义在外部唤醒引脚 WKUP1 上进行事件检测所用的极性
0：在高电平（上升沿）检测
1：在低电平（下降沿）检测

6.6.5 PWR 状态寄存器 1 (PWR_SR1)

PWR status register 1

偏移地址：0x010

复位值：0x0000 0000。退出待机模式后，此寄存器不会复位。

访问：与标准 APB 读操作相比，读取此寄存器需要 2 个额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WUFI	C2HF	802AF	BLE AF	CRPEF	802 WUF	BLE WUF	BORHF	SMPSFBF	Res.	Res.	WUF5	WUF4	WUF3	WUF2	WUF1
r	r	r	r	r	r	r	r	r			r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15 **WUFI**：内部唤醒中断标志 (Internal wakeup interrupt flag)
在内部唤醒线上检测到唤醒事件时，该位置 1。清除所有内部唤醒源时，该位清零。

位 14 **C2HF**：CPU2 保持中断标志 (CPU2 Hold interrupt flag)
当在 C2BOOT = 0 时检测到 CPU2 唤醒时该位置 1。该位通过 PWR_SCR.CC2HF 清零。

位 13 **802AF**：802.15.4 活动结束中断标志 (802.15.4 end of activity interrupt flag)
802.15.4 活动结束时，该位置 1。该位通过 PWR_SCR.C802AF 清零。

位 12 **BLEAF**：BLE 活动结束中断标志 (BLE end of activity interrupt flag)
BLE 活动结束时，该位置 1。该位通过 PWR_SCR.CBLEAF 清零。

位 11 **CRPEF**：使能关键无线电阶段活动结束中断标志 (Enable critical radio phase end of activity interrupt flag)
无线电阶段活动结束时，该位置 1。该位通过 PWR_SCR.CCRPEF 清零。

位 10 **802WUF**：802.15.4 唤醒中断标志 (802.15.4 wakeup interrupt flag)
在 802.15.4 线上检测到唤醒事件时，该位置 1。该位通过 PWR_SCR.C802WUF 清零。

位 9 **BLEWUF**：BLE 唤醒中断标志 (BLE wakeup interrupt flag)
在 BLE 线上检测到唤醒事件时，该位置 1。该位通过 PWR_SCR.CBLEWUF 清零。

位 8 **BORHF**：BORH 中断标志 (BORH interrupt flag)
当 VDD/ 上升到 BORH 阈值以上时，该位置 1。该位通过 PWR_SCR.CBORHF 清零。

位 7 **SMPSFBF**：强制 SMPS 降压转换器进入旁路模式中断标志 (SMPS step-down converter forced in Bypass interrupt flag)
当 SMPS 降压转换器在 SMPS 模式下使能并由于 BORH 阈值而强制进入旁路模式时，该位置 1。通过 *PWR 控制寄存器 5 (PWR_CR5)* 中的 BORHC 位进行的 BORH 配置应选择用于强制 SMPS 降压转换器进入旁路模式的 BORH。
该位通过 PWR_SCR.CSMPSFBF 清零。

位 6:5 保留，必须保持复位值。

位 4 **WUF5**：唤醒标志 5 (Wakeup flag 5)
当在 WKUP5 唤醒引脚上检测到唤醒事件时该位置 1。通过向 PWR_SCR 寄存器中的 CWUF5 写入 1 进行清零。

位 3 **WUF4**：唤醒标志 4 (Wakeup flag 4)
当在 WKUP4 唤醒引脚上检测到唤醒事件时该位置 1。该位通过向 *PWR 状态清零寄存器 (PWR_SCR)* 中的 CWUF4 位写入 1 进行清零。

位 2 **WUF3**：唤醒标志 3 (Wakeup flag 3)
当在 WKUP3 唤醒引脚上检测到唤醒事件时该位置 1。该位通过向 *PWR 状态清零寄存器 (PWR_SCR)* 中的 CWUF3 位写入 1 进行清零。

位 1 **WUF2**：唤醒标志 2 (Wakeup flag 2)
当在 WKUP2 唤醒引脚上检测到唤醒事件时该位置 1。该位通过向 *PWR 状态清零寄存器 (PWR_SCR)* 中的 CWUF2 位写入 1 进行清零。

位 0 **WUF1**：唤醒标志 1 (Wakeup flag 1)
当在 WKUP1 唤醒引脚上检测到唤醒事件时该位置 1。该位通过向 *PWR 状态清零寄存器 (PWR_SCR)* 中的 CWUF1 位写入 1 进行清零。

6.6.6 PWR 状态寄存器 2 (PWR_SR2)

PWR status register 2

偏移地址：0x014

复位值：0x0000 0002。退出待机/关断模式后，寄存器会部分复位。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PVMO3	Res.	PVMO1	PVDO	VOSF	REGLPF	REGLPS	Res.	Res.	Res.	Res.	Res.	Res.	SMPSF	SMPSBF
	r		r	r	r	r	r							r	r

位 31:15 保留，必须保持复位值。

位 14 **PVMO3**：外设电压监测输出：V_{DDA} 与 1.62 V (Peripheral voltage monitoring output: V_{DDA} vs. 1.62 V)

0：V_{DDA} 电压高于 PVM3 阈值（约 1.62 V）。

1：V_{DDA} 电压低于 PVM3 阈值（约 1.62 V）。

注：禁止 PVM3 时 (PVME3 = 0) 会清零 PVMO3。使能 PVM3 后经过 PVM3 唤醒时间，PVM3 输出开始有效。

位 13 保留，必须保持复位值。

位 12 **PVMO1**：外设电压监测输出：V_{DDUSB} 与 1.2 V (Peripheral voltage monitoring output: V_{DDUSB} vs. 1.2 V)

0：V_{DDUSB} 电压高于 PVM1 阈值（约 1.2 V）。

1：V_{DDUSB} 电压低于 PVM1 阈值（约 1.2 V）。

注：禁止 PVM1 时 (PVME1 = 0) 会清零 PVMO1。使能 PVM1 后经过 PVM1 唤醒时间，PVM1 输出开始有效。

位 11 **PVDO**：电源电压检测器输出 (Power voltage detector output)

0：V_{DD} 或 PVD_IN 的电压高于所选 PVD 阈值

1：V_{DD} 或 PVD_IN 的电压低于所选 PVD 阈值

位 10 **VOSF**：电压调节标志 (Voltage scaling flag)

电压调节变更后，内部稳压器准备好需要一定的延时。VOSF 用于指示稳压器已达到 *PWR 控制寄存器 1 (PWR_CR1)* 的 VOS 位所定义的电压。

0：稳压器在所选电压范围内准备就绪

1：稳压器输出电压更改为所需电平

位 9 **REGLPF**：低功耗稳压器标志 (Low-power regulator flag)

MCU 处于低功耗运行模式时，此位由硬件置 1。当 MCU 从低功耗运行模式退出时，该位保持为 1，直到稳压器在主模式下准备就绪。在提高产品频率之前必须对该位进行轮询。

当稳压器准备就绪后，此位由硬件清零。

0：稳压器在主模式 (MR) 下准备就绪

1：稳压器处于低功耗模式 (LPR)

位 8 **REGLPS**：低功耗稳压器启动 (Low-power regulator started)

该位用于提供上电复位或待机/关断后低功耗稳压器是否准备就绪的信息。如果在 REGLPS 位仍然清零的情况下进入待机模式，则会增加从待机模式唤醒的时间。

0：低功耗稳压器未准备就绪

1：低功耗稳压器已准备就绪

位 7:2 保留，必须保持复位值。

位 1 **SMPSF**：SMPS 降压转换器 SMPS 模式标志 (SMPS step-down converter SMPS mode flag)
该位用于指示 SMPS 降压转换器处于 SMPS 模式。
0：SMPS 降压转换器未就绪。
1：SMPS 降压转换器已就绪。

位 0 **SMPSBF**：SMPS 降压转换器旁路模式标志 (SMPS step-down converter Bypass mode flag)
该位用于指示 SMPS 降压转换器处于旁路模式。
0：SMPS 降压转换器未处于旁路模式。（处于开路模式或 SMPS 模式）
1：SMPS 降压转换器处于旁路模式。

6.6.7 PWR 状态清零寄存器 (PWR_SCR)

PWR status clear register

偏移地址：0x018

复位值：0x0000 0000。

访问：与标准的 APB 写操作相比，写入此寄存器需要 3 个额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CC2HF	C802 AF	CFBLE AF	CCRP EF	C802 WUF	CBLE WUF	CBORH F	CSMPS FBF	Res.	Res.	CWUF5	CWUF4	CWUF3	CWUF2	CWUF1
	w	w	w	w	w	w	w	w			w	w	w	w	w

位 31:15 保留，必须保持复位值。

位 14 **CC2HF**：清零 CPU2 保持中断标志 (Clear CPU2 Hold interrupt flag)
将此位置 1 会清零 PWR_SR1 中的 C2HF 标志。此位始终读为 0。

位 13 **C802AF**：清零 802.15.4 活动结束中断标志 (Clear 802.15.4 end of activity interrupt flag)
将此位置 1 会清零 PWR_SR1 中的 802AF 标志。此位始终读为 0。

位 12 **CBLEAF**：清零 BLE 活动结束中断标志 (Clear BLE end of activity interrupt flag)
将此位置 1 会清零 PWR_SR1 中的 BLEAF 标志。此位始终读为 0。

位 11 **CCRPEF**：清零关键无线电阶段活动结束中断标志 (Clear critical radio phase end of activity interrupt flag)
将此位置 1 会清零 PWR_SR1 中的 CRPEF 标志。此位始终读为 0。

位 10 **C802WUF**：清零 802.15.4 唤醒中断标志 (Clear 802.15.4 wakeup interrupt flag)
将此位置 1 会清零 PWR_SR1 中的 802WUF 标志。此位始终读为 0。

位 9 **CBLEWUF**：清零 BLE 唤醒中断标志 (Clear BLE wakeup interrupt flag)
将此位置 1 会清零 PWR_SR1 中的 BLEWUF 标志。此位始终读为 0。

位 8 **CBORHF**：清零 BORH 中断标志 (Clear BORH interrupt flag)
将此位置 1 会清零 PWR_SR1 中的 SBORHF 标志。此位始终读为 0。

位 7 **CSMPSFBF**：清零强制 SMPS 降压转换器进入旁路模式中断标志 (Clear SMPS step-down converter forced in Bypass interrupt flag)
将此位置 1 会清零 PWR_SR1 中的 SMPSFBF 标志。此位始终读为 0。

位 6:5 保留，必须保持复位值。

位 4 **CWUF5**：唤醒标志 5 清零 (Clear wakeup flag 5)
将此位置 1 会清零 PWR_SR1 寄存器中的 WUF5 标志。

位 3 **CWUF4**：唤醒标志 4 清零 (Clear wakeup flag 4)
将此位置 1 会清零 PWR_SR1 寄存器中的 WUF4 标志。

位 2 **CWUF3**：唤醒标志 3 清零 (Clear wakeup flag 3)
将此位置 1 会清零 PWR_SR1 寄存器中的 WUF3 标志。

位 1 **CWUF2**：唤醒标志 2 清零 (Clear wakeup flag 2)
将此位置 1 会清零 PWR_SR1 寄存器中的 WUF2 标志。

位 0 **CWUF1**：唤醒标志 1 清零 (Clear wakeup flag 1)
将此位置 1 会清零 PWR_SR1 寄存器中的 WUF1 标志。

6.6.8 PWR 控制寄存器 5 (PWR_CR5)

PWR control register 5

偏移地址：0x01C

复位值：0x0000 427X，其中 X 是出厂前编程的。退出待机模式后，此寄存器不会复位。

访问：与标准的 APB 写操作相比，写入此寄存器需要 3 个额外的 APB 周期。

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

位 31:16 保留，必须保持复位值。

位 15 **SMPSEN**：使能 SMPS 降压转换器 SMPS 模式 (Enable SMPS step-down converter SMPS mode enabled)
当使能 SMPS 降压转换器实时开启且 V_{DD} 大小降到 BORH 阈值以下时，该位复位为 0。
0：禁止 SMPS 降压转换器 SMPS 模式。
1：使能 SMPS 降压转换器 SMPS 模式。

注： *模拟信号的噪声性能可能受集成 SMPS 切换的影响。为了防止这种情况，在测量模拟信号时可以实时关闭 SMPS。*

位 14:9 保留，必须保持复位值。

位 8 **BORHC**：BORH 配置选择 (BORH configuration selection)
0：BORH 将产生系统复位。
1：BORH 将强制 SMPS 降压转换器进入旁路模式（BORL 仍会产生系统复位）。

位 7 保留，必须保持复位值。

位 6:4 **SMPSSC**：SMPS 降压转换器电源启动电流选择 (SMPS step-down converter supply startup current selection)

启动电流限制为最高 80 mA + SMPSSC x 20 mA。

位 3:0 **SMPSVOS**：SMPS 降压转换器电压输出调节 (SMPS step-down converter voltage output scaling)

选项字节加载后出厂微调值达到 1.5 V 时，这些位将进行初始化，随后可被固件覆盖。

SMPS 降压输出电压步长为 50 mV。

出厂微调值 - 0x8 对应 1.50 V 的 $V_{FBSMSPS}$ 值，要得到 1.40 V，应将该值减去 0x2。

– 0x0 = 最小电压值
– 0xF = 最大电压值

出厂微调值可以从地址 0x1FFF 7558 位 [11:8] 获得。

6.6.9 PWR 端口 A 上拉控制寄存器 (PWR_PUCRA)

PWR Port A pull-up control register

偏移地址：0x020

复位值：0x0000 0000。退出待机模式后，此寄存器不会复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PUy**：端口 A 上拉位 y (Port A pull-up bit y) (y=0..15)

在 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR CPU2 控制寄存器 3 (PWR_C2CR3)* 中的其中一个 APC 位置 1 时将此位置 1，会激活 Px[y] 的上拉。如果相应的 PDy 位也置 1，则不激活上拉。

6.6.10 PWR 端口 A 下拉控制寄存器 (PWR_PDCRA)

PWR Port A pull-down control register

偏移地址：0x024

复位值：0x0000 0000。退出待机模式后，此寄存器不会复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PDy**：端口 A 下拉位 y (Port A pull-down bit y) (y=0..15)

在 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR CPU2 控制寄存器 3 (PWR_C2CR3)* 中的其中一个 APC 位置 1 时将此位置 1，会激活 PA[y] 的下拉。

6.6.11 PWR 端口 B 上拉控制寄存器 (PWR_PUCRB)

PWR Port B pull-up control register

偏移地址：0x028

复位值：0x0000 0000。退出待机模式后，此寄存器不会复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PUy**：端口 B 上拉位 y (Port B pull-up bit y) (y=0..15)

在 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR CPU2 控制寄存器 3 (PWR_C2CR3)* 中的其中一个 APC 位置 1 时将此位置 1，会激活 Px[y] 的上拉。如果相应的 PDy 位也置 1，则不激活上拉。

6.6.12 PWR 端口 B 下拉控制寄存器 (PWR_PDCRB)

PWR Port B pull-down control register

偏移地址：0x02C

复位值：0x0000 0000。退出待机模式后，此寄存器不会复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PDy**：端口 B 下拉位 y (Port B pull-down bit y) (y=0..15)

在 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR CPU2 控制寄存器 3 (PWR_C2CR3)* 中的其中一个 APC 位置 1 时将此位置 1，会激活 PB[y] 的下拉。

6.6.13 PWR 端口 C 上拉控制寄存器 (PWR_PUCRC)

PWR Port C pull-up control register

偏移地址：0x030

复位值：0x0000 0000。退出待机模式后，此寄存器不会复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PUy**：端口 C 上拉位 y (Port C pull-up bit y) (y=0..15)

在 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR CPU2 控制寄存器 3 (PWR_C2CR3)* 中的其中一个 APC 位置 1 时将此位置 1，会激活 Px[y] 的上拉。如果相应的 PDy 位也置 1，则不激活上拉。

6.6.14 PWR 端口 C 下拉控制寄存器 (PWR_PDCRC)

PWR Port C pull-down control register

偏移地址：0x034

复位值：0x0000 0000。退出待机模式后，此寄存器不会复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PDy**：端口 C 下拉位 y (Port C pull-down bit y) (y=0..15)

在 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR CPU2 控制寄存器 3 (PWR_C2CR3)* 中的其中一个 APC 位置 1 时将此位置 1，会激活 PC[y] 的下拉。

6.6.15 PWR 端口 D 上拉控制寄存器 (PWR_PUCRD)

PWR Port D pull-up control register

偏移地址：0x038

复位值：0x0000 0000。退出待机模式后，此寄存器不会复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PUy**：端口 D 上拉位 y (Port D pull-up bit y) (y=0..15)

在 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR CPU2 控制寄存器 3 (PWR_C2CR3)* 中的其中一个 APC 位置 1 时将此位置 1，会激活 Px[y] 的上拉。如果相应的 PDy 位也置 1，则不激活上拉。

6.6.16 PWR 端口 D 下拉控制寄存器 (PWR_PDCRD)

PWR Port D pull-down control register

偏移地址：0x03C

复位值：0x0000 0000。退出待机模式后，此寄存器不会复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PDy**：端口 D 下拉位 y (Port D pull-down bit y) (y=0..15)

在 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR CPU2 控制寄存器 3 (PWR_C2CR3)* 中的其中一个 APC 位置 1 时将此位置 1，会激活 PD[y] 的下拉。

6.6.17 PWR 端口 E 上拉控制寄存器 (PWR_PUCRE)

PWR Port E pull-up control register

偏移地址：0x020

复位值：0x0000 0000。退出待机模式后，此寄存器不会复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU4	PU3	PU2	PU1	PU0
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4:0 **PUy**：端口 E 上拉位 y (Port E pull-up bit y) (y=0..4)

在 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR CPU2 控制寄存器 3 (PWR_C2CR3)* 中的其中一个 APC 位置 1 时将此位置 1，会激活 Px[y] 的上拉。如果相应的 PDy 位也置 1，则不激活上拉。

6.6.18 PWR 端口 E 下拉控制寄存器 (PWR_PDCRE)

PWR Port E pull-down control register

偏移地址：0x044

复位值：0x0000 0000。退出待机模式后，此寄存器不会复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD4	PD3	PD2	PD1	PD0
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4:0 **PDy**：端口 E 下拉位 y (Port E pull-down bit y) (y=0..4)

在 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR CPU2 控制寄存器 3 (PWR_C2CR3)* 中的其中一个 APC 位置 1 时将此位置 1，会激活 PE[y] 的下拉。

6.6.19 PWR 端口 H 上拉控制寄存器 (PWR_PUCRH)

PWR Port H pull-up control register

偏移地址：0x058

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU3	Res.	PU1	PU0
												rw		rw	rw

位 31:4 保留，必须保持复位值。

位 3 **PU3**：端口 H 上拉位 3 (Port H pull-up bit 3)
在 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR CPU2 控制寄存器 3 (PWR_C2CR3)* 中的其中一个 APC 位置 1 时将此位置 1，会激活 PH[y] 的上拉。
如果相应的 PDy 位也置 1，则不激活上拉。

位 2 保留，必须保持复位值。

位 1:0 **PUy**：端口 H 上拉位 y (Port H pull-up bit y) (y=0..1)
在 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR CPU2 控制寄存器 3 (PWR_C2CR3)* 中的其中一个 APC 位置 1 时将此位置 1，会激活 Px[y] 的上拉。如果相应的 PDy 位也置 1，则不激活上拉。

6.6.20 PWR 端口 H 下拉控制寄存器 (PWR_PDCRH)

PWR Port H pull-down control register

偏移地址：0x05C

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD3	Res.	PD1	PD0
												rw		rw	rw

位 31:4 保留，必须保持复位值。

位 3 **PD3**：端口 H 下拉位 3 (Port H pull-down bit 3)
在 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR CPU2 控制寄存器 3 (PWR_C2CR3)* 中的其中一个 APC 位置 1 时将此位置 1，会激活 PH[y] 的下拉。

位 2 保留，必须保持复位值。

位 1:0 **PDy**：端口 H 下拉位 y (Port H pull-down bit y) (y=0..1)
在 *PWR 控制寄存器 3 (PWR_CR3)* 和 *PWR CPU2 控制寄存器 3 (PWR_C2CR3)* 中的其中一个 APC 位置 1 时将此位置 1，会激活 PH[y] 的下拉。

6.6.21 PWR CPU2 控制寄存器 1 (PWR_C2CR1)

PWR CPU2 control register 1

偏移地址：0x080

复位值：0x0000 0000。从待机模式唤醒后，此寄存器会复位，但位 [2:0] 除外。

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

位 31:16 保留，必须保持复位值。

位 15 **802EWKUP**：802.15.4 外部唤醒信号 (802.15.4 external wakeup signal)

此位置 1 将强制唤醒 802.15.4 控制器。当 802.15.4 控制器退出其睡眠模式时，此位会自动复位。

0：不执行任何操作。

1：将 802.15.4 控制器从其睡眠模式唤醒。

位 14 **BLEEWKUP**：BLE 外部唤醒 (BLE external wakeup)

此位置 1 将强制唤醒 BLE 控制器。当 BLE 控制器退出其睡眠模式时，此位会自动复位。

0：不执行任何操作。

1：将 BLE 控制器从其睡眠模式唤醒。

位 13:6 保留，必须保持复位值。

位 5 **FPDS**：CPU2 处于 LPSleep 期间 Flash 的掉电模式 (Flash memory power down mode during LPSleep for CPU2)

该位选择当 CPU 均处于睡眠模式时，Flash 是处于掉电模式还是空闲模式。只有当系统处于 LPSleep 模式且 CPU1 的 PWR_CR1.FPDS 位也允许时，Flash 才能设置为掉电模式。

0：当系统处于 LPSleep 模式时，Flash 处于空闲模式

1：当系统处于 LPSleep 模式时，Flash 处于掉电模式

位 4 **FPDR**：CPU2 处于 LPRun 期间 Flash 的掉电模式 (Flash memory power down mode during LPRun for CPU2)

只有通过首先将（代码 0xC1B0）写入该寄存器（写入代码时，寄存器位不会更新）解锁该寄存器位后，才能向该位写入 1。选择当处于 LPRun 模式时，Flash 是处于掉电模式还是空闲模式。（只有从 SRAM 执行代码时，Flash 才能处于掉电模式）。只有当系统处于 LPRun 模式且 CPU1 的 PWR_CR1.FPDR 位也允许时，Flash 才能设置为掉电模式。

0：当系统处于 LPRun 模式时，Flash 处于空闲模式

1：当系统处于 LPRun 模式时，Flash 处于掉电模式

位 3 保留，必须保持复位值。

位 2:0 **LPMS[2:0]**：CPU2 的低功耗模式选择 (Low-power mode selection for CPU2)

退出待机模式时，这些位不会复位。

当 CPU2 进入深度睡眠模式时，使用这些位选择进入的低功耗模式。进入的系统低功耗模式还取决于 CPU1 的 PWR_CR1.LPMS 允许的低功耗模式。

000：停止 0 模式

001：停止 1 模式

010：停止 2 模式

011：待机模式

1xx：关断模式

注： *如果 LPR 位置 1，则无法选择停止 2 模式，因此会进入停止 1 模式而不是停止 2 模式。在待机模式下，SRAM2 内容可以保留或不保留，具体取决于 PWR 控制寄存器 3 (PWR_CR3) 中的 RRS 位配置。*

6.6.22 PWR CPU2 控制寄存器 3 (PWR_C2CR3)

PWR CPU2 control register 3

偏移地址：0x084

复位值：0x0000 8000。退出待机模式后，此寄存器不会复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIWUL	Res.	Res.	APC	Res.	E802WUP	EBLEWUP	Res.	Res.	Res.	Res.	EWUP5	EWUP4	EWUP3	EWUP2	EWUP1
rw			rw		rw	rw					rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **EIWUL**：使能 CPU2 的内部唤醒线 (Enable internal wakeup line for CPU2)

0：禁止 CPU2 的内部唤醒线。

1：使能 CPU2 的内部唤醒线。

位 14:13 保留，必须保持复位值。

位 12 **APC**：为 CPU2 应用上拉和下拉配置 (Apply pull-up and pull-down configuration for CPU2)

当 CPU2 的该位或 CPU1 的 PWR_CR3.APC 位置 1 时，将应用 PWR_PUCRx 和 PWR_PDCRx 寄存器中定义的 I/O 上拉和下拉配置。当两个位清零时，PWR_PUCRx 和 PWR_PDCRx 寄存器不会应用于 I/O。

位 11 保留，必须保持复位值。

位 10 **E802WUP**：使能 CPU2 的 802.15.4 主机唤醒中断 (Enable 802.15.4 host wakeup interrupt for CPU2)

0：禁止 CPU2 中断。

1：使能 CPU2 中断。

位 9 **EBLEWUP**：使能 CPU2 的 BLE 主机唤醒中断 (Enable BLE host wakeup interrupt for CPU2)

0：禁止 CPU2 中断。

1：使能 CPU2 中断。

位 8:5 保留，必须保持复位值。

位 4 **EWUP5**：使能 CPU2 的唤醒引脚 WKUP5 (Enable Wakeup pin WKUP5 for CPU2)

该位置 1 时，会使能外部唤醒引脚 WKUP5，并在 CPU2 出现上升沿或下降沿时触发中断和从停止、待机或关断事件唤醒。通过 *PWR 控制寄存器 4 (PWR_CR4)* 中的 WP5 位配置有效边沿。

位 3 **EWUP4**：使能 CPU2 的唤醒引脚 WKUP4 (Enable Wakeup pin WKUP4 for CPU2)

该位置 1 时，会使能外部唤醒引脚 WKUP4，并在 CPU2 出现上升沿或下降沿时触发中断和从停止、待机或关断事件唤醒。通过 *PWR 控制寄存器 4 (PWR_CR4)* 中的 WP4 位配置有效边沿。

位 2 **EWUP3**：使能 CPU2 的唤醒引脚 WKUP3 (Enable Wakeup pin WKUP3 for CPU2)

该位置 1 时，会使能外部唤醒引脚 WKUP3，并在 CPU2 出现上升沿或下降沿时触发中断和从停止、待机或关断事件唤醒。通过 *PWR 控制寄存器 4 (PWR_CR4)* 中的 WP3 位配置有效边沿。

位 1 **EWUP2**：使能 CPU2 的唤醒引脚 WKUP2 (Enable Wakeup pin WKUP2 for CPU2)

该位置 1 时，会使能外部唤醒引脚 WKUP2，并在 CPU2 出现上升沿或下降沿时触发中断和从停止、待机或关断事件唤醒。通过 *PWR 控制寄存器 4 (PWR_CR4)* 中的 WP2 位配置有效边沿。

位 0 **EWUP1**：使能 CPU2 的唤醒引脚 WKUP1 (Enable Wakeup pin WKUP5 for CPU1)

该位置 1 时，会使能外部唤醒引脚 WKUP1，并在 CPU2 出现上升沿或下降沿时触发中断和从停止、待机或关断事件唤醒。通过 *PWR 控制寄存器 4 (PWR_CR4)* 中的 WP1 位配置有效边沿。

6.6.23 PWR 扩展状态和状态清除寄存器 (PWR_EXTSCR)

PWR extended status and status clear register

偏移地址：0x088

复位值：0x0000 0000

访问：与标准的 APB 写操作相比，写入此寄存器需要 3 个额外的 APB 周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2DS	C1DS	CRPF	Res.	C2STOPF	C2SBF	C1STOPF	C1SBF	Res.	Res.	Res.	Res.	Res.	CCRPF	C2CSSF	C1CSSF
r	r	r		r	r	r	r						w	w	w

位 31:16 保留，必须保持复位值。

位 15 **C2DS**：CPU2 深度睡眠模式 (CPU2 deepsleep mode)

当 CPU2 进入深度睡眠模式或由 C2BOOT 保持时，该位由硬件置 1。

0：CPU2 正在运行或处于睡眠状态

1：CPU2 处于深度睡眠状态或由 C2BOOT 保持

位 14 **C1DS**：CPU1 深度睡眠模式 (CPU1 deepsleep mode)

当 CPU1 进入深度睡眠模式时，该位由硬件置 1。

0：CPU1 正在运行或处于睡眠状态

1：CPU1 处于深度睡眠状态

位 13 **CRPF**：关键无线电系统阶段 (Critical Radio system phase)

当无线电系统唤醒时，该位由硬件置 1。该位在无线电系统进入低功耗模式时由硬件复位，在写入 CCRPF 时由软件复位。

0：未处于关键无线电系统阶段。

1：正处于关键无线电系统阶段。

位 12 保留，必须保持复位值。

位 11 **C2STOPF**：CPU2 系统停止标志 (System Stop flag for CPU2)
此位由硬件置 1，并且只能通过任何复位或将 C2CSSF 位置 1 清零。
0：系统未处于停止模式
1：系统已处于停止模式。

位 10 **C2SBF**：CPU2 系统待机标志 (System Standby flag for CPU2)
此位由硬件置 1，并且只能通过 POR 复位或将 C2CSSF 位置 1 清零。
0：系统未处于待机模式
1：系统已处于待机模式。

位 9 **C1STOPF**：CPU1 系统停止标志 (System Stop flag for CPU1)
此位由硬件置 1，并且只能通过任何复位或将 C1CSSF 位置 1 清零。
0：系统未处于停止模式
1：系统已处于停止模式。

位 8 **C1SBF**：CPU1 系统待机标志 (System Standby flag for CPU1)
此位由硬件置 1，并且只能通过 POR 复位或将 C1CSSF 位置 1 清零。
0：系统未处于待机模式
1：系统已处于待机模式。

位 7:3 保留，必须保持复位值。

位 2 **CCRPF**：清除关键无线电系统阶段 (Clear critical Radio system phase)
将该位置 1 会清零 CRPF 位。

位 1 **C2CSSF**：清零 CPU2 停止待机标志 (Clear CPU2 Stop Standby flags)
将此位置 1 会清零 C2STOPF 和 C2SBF 位。

位 0 **C1CSSF**：清零 CPU1 停止待机标志 (Clear CPU1 Stop Standby flags)
将此位置 1 会清零 C1STOPF 和 C1SBF 位。

6.6.24 PWR 寄存器映射和复位值表

表 35. PWR 寄存器映射和复位值

偏移	寄存器	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
0x000	PWR_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPR	Res.	Res.	Res.	VOS [1:0]		DBP	Res.	Res.	FPDS	FPDR	Res.	LPMS [2:0]		
	Reset value																		0				0	1	0			0	0		0	0	0
0x004	PWR_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USV	Res.	Res.	Res.	PVME3	Res.	PVME1	PLS [2:0]			PVDE
	Reset value																						0				0		0	0	0	0	0
0x008	PWR_CR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EIWUL	EC2H	E802A	EBLEA	ECRPE	APC	RRS	EBORHSMPSFB	Res.	Res.	Res.	EWUP5	EWUP4	EWUP3	EWUP2	EWUP1
	Reset value																	1	0	0	0	0	0	0	0				0	0	0	0	0
0x00C	PWR_CR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	C2BOOT	Res.	Res.	Res.	Res.	Res.	VBRS	VBE	Res.	Res.	Res.	WP5	WP4	WP3	WP2	WP1
	Reset value																	0						0	0				0	0	0	0	0
0x010	PWR_SR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUFI	C2HF	802AF	BLEAF	CRPEF	802WUF	BLEWUF	BORHF	SMPSFBF	Res.	Res.	WUF5	WUF4	WUF3	WUF2	WUF1
	Reset value																	0	0	0	0	0	0	0	0	0			0	0	0	0	0
0x014	PWR_SR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PVMO3	Res.	PVMO1	PVDO	VOSF	REGLPF	REGLPS	Res.	Res.	Res.	Res.	Res.	Res.	SMPSF	SMPSBF
	Reset value																		0		0	0	0	0	0							0	0
0x018	PWR_SCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2HF	C802AF	CBLEAF	CCRPEF	C802WUF	CBLEWUF	CBORHF	CSMPSFBF	Res.	Res.	CWUF5	CWUF4	CWUF3	CWUF2	CWUF1
	Reset value																		0	0	0	0	0	0	0	0			0	0	0	0	0
0x01C	PWR_CR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMPSEN	Res.	Res.	Res.	Res.	Res.	Res.	BORHC	Res.	SMPS SC[2:0]			SMPS VOS[3:0]			
	Reset value																	0							0		1	1	1	x	x	x	x
0x020	PWR_PUCRA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	PWR_PDCRA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x028	PWR_PUCRB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x02C	PWR_PDCRB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x030	PWR_PUCRC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 35. PWR 寄存器映射和复位值（续）

偏移	寄存器	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
0x034	PWR_PDCRC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x038	PWR_PUCRD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03C	PWR_PDCRD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x040	PWR_PUCRE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU4	PU3	PU2	PU1	PU0
	Reset value																												0	0	0	0	0
0x044	PWR_PDCRE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD4	PD3	PD2	PD1	PD0
	Reset value																												0	0	0	0	0
0x058	PWR_PUCRH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU3	Res.	PU1	PU0
	Reset value																													0		0	0
0x05C	PWR_PDCRH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD3	Res.	PD1	PD0
	Reset value																													0		0	0
0x080	PWR_C2CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	802EWKUP	BLEEWKUP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FPDS	FPDR	Res.	LPMS [2:0]		
	Reset value																	0	0									0	0		0	0	0
0x084	PWR_C2CR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EIWUL	Res.	Res.	APC	Res.	E802IWUP	EBLEWUP	Res.	Res.	Res.	Res.	EWUP5	EWUP4	EWUP3	EWUP2	EWUP1
	Reset value																	1			0		0	0					0	0	0	0	0
0x088	PWR_EXTSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	C2DS	C1DS	CRPF	Res.	C2STOPF	C2SBF	C1STOPF	C1SBF	Res.	Res.	Res.	Res.	Res.	CCRPF	C2CSSF	C1CSSF
	Reset value																	0	0	0		0	0	0	0						0	0	0

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

7 外设互连矩阵

7.1 简介

多个外设之间直接连接，使得彼此之间可实现自主通信和/或同步。这可节省 CPU 资源，从而降低功耗。此外，这些硬件连接还可消除软件延时，设计可预测性更高的系统。

根据外设的不同，这些互连可以在运行、睡眠、低功耗运行和睡眠、停止 0、停止 1、停止 2 模式下工作。

7.2 连接汇总

表 36. STM32WB55xx 外设互连矩阵(1)(2)

源	目标									
	TIM1	TIM2	TIM16	TIM17	LPTIM1	LPTIM2	ADC1	COMP1	COMP2	IRTIM
TIM1	-	1	-	-	-	-	2	6	6	-
TIM2	1	-	-	-	-	-	2	6	6	-
TIM16	-	-	-	-	-	-	-	-	-	11
TIM17	1	-	-	-	-	-	-	-	-	11
LPTIM1	-	-	-	-	-	-	-	-	-	-
LPTIM2	-	-	-	-	-	-	-	-	-	-
ADC1	3	-	-	-	-	-	-	-	-	-
温度传感器	-	-	-	-	-	-	8	-	-	-
VBAT	-	-	-	-	-	-	8	-	-	-
VREFINT	-	-	-	-	-	-	8	-	-	-
HSE	-	-	-	4	-	-	-	-	-	-
LSE	-	4	4	-	-	-	-	-	-	-
MSI	-	-	-	4	-	-	-	-	-	-
LSI	-	-	4	-	-	-	-	-	-	-
MCO	-	-	-	4	-	-	-	-	-	-
EXTI	-	-	-	-	-	-	2	-	-	-
RTC	-	-	4	-	5	5	-	-	-	-
COMP1	9	9	9	9	5	5	-	-	-	-
COMP2	9	9	9	9	5	5	-	-	-	-
SYST ERR	10	-	10	10	-	-	-	-	-	-
USB	-	7	-	-	-	-	-	-	-	-

1. 表中的编号是*第 7.3 节：互连详细信息*中相应小节的链接。
2. 灰色单元格中的“-”符号表示无互连。

7.3 互连详细信息

7.3.1 从定时器 (TIM1/TIM2/TIM17) 到定时器 (TIM1/TIM2)

目的

某些定时器从内部连接在一起，以实现同步或链接。

当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。*第 24.3.26 节：定时器同步*中对功能进行了说明。

以下章节对同步模式进行了详细说明：

- *第 24.3.26 节：定时器同步*（面向高级控制定时器 (TIM1)）
- *第 25.3.18 节：定时器与外部触发同步*（面向通用定时器 (TIM2)）

触发信号

输出（来自住器件）信号在 TIMx_TRGO（对于 TIM1，在 TIMx_TRGO2 上）上，并伴随一个可配置的定时器事件。输入（到从器件）信号在 TIMx_ITR0/ITR1/ITR2/ITR3 上。

TIM1 的输入和输出信号如*图 148：高级控制定时器框图*所示。

以下各表介绍了可能的主/从连接：

- *表 154：TIM1 内部触发连接*
- *表 158：TIM2 内部触发连接*

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

7.3.2 从定时器 (TIM1/TIM2) 和 EXTI 到 ADC (ADC1)

目的

通用定时器 TIM2、高级控制定时器 TIM1 和 EXTI 可用于生成 ADC 触发事件。

*第 24.3.27 节：ADC 同步*对 TIMx 同步进行了说明。

*第 16.3.18 节：外部触发转换和触发极性（EXTSEL、EXTEN、JEXTSEL、JEXTEN）*对 ADC 同步进行了说明。

触发信号

TIMx_TRGO，TIMx_TRGO2 或 TIMx_CCx 事件的输出（来自定时器）信号。

EXT[15:0] 和 JEXT[15:0] 的输入信号。

以下各表给出了定时器和 ADC 之间的连接：

- *表 73：ADC1——常规通道的外部通道*
- *表 74：ADC1——注入通道的外部通道*

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

7.3.3 从 ADC (ADC1) 到定时器 (TIM1)

目的

ADC1 可通过看门狗信号向高级控制定时器 (TIM1) 提供触发事件。

*第 16.3.28 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*中对 ADC 模拟看门狗设置进行了说明。

*第 24.3.4 节：外部触发输入*中对定时器的触发设置进行了说明。

触发信号

ADC_AWDx_OUT n = 1，2，3（对于 ADC1）输出（来自 ADC）信号，其中 x = 1、2、3（ADC 上有 3 个看门狗）；TIMx_ETR（外部触发）输入（到定时器）信号。

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

7.3.4 从 HSE、LSE、LSI、MSI、MCO、RTC 到定时器 (TIM2/TIM16/TIM17)

目的

外部时钟（HSE 和 LSE）、内部时钟（LSI 和 MSI）、微控制器输出时钟 (MCO)、GPIO 和 RTC 唤醒中断可用作通用定时器 (TIM16/17) 通道 1 的输入。

这可以校准 HSI16/MSI 系统时钟（使用 TIM16 和 LSE）或 LSI（使用 TIM16 和 HSE）。也可用于精确测量 LSI（使用 TIM16 和 HSI16）或 MSI（使用 TIM17 和 HSI16）振荡器频率。

使用低速外部 (LSE) 振荡器时，无需额外的硬件连接。

*第 8.2.21 节：基于 TIM16/TIM17 的内部/外部时钟测量*对此功能进行了说明。

外部时钟 LSE 可用作 TIM2_ETR 引脚上通用定时器 (TIM2) 的输入，请参见*第 8.2.21 节：基于 TIM16/TIM17 的内部/外部时钟测量*。

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

7.3.5 从 RTC、COMP1、COMP2 到低功耗定时器 (LPTIM1/LPTIM2)

目的

RTC 闹钟 A/B、RTC_TAMP1/2/3 输入检测、COMP1/2_OUT 可用作触发信号以启动 LPTIM 计数器 (LPTIM1/2)。

触发信号

第 27.4.6 节：触发多路复用器（以及后续部分）对此触发功能进行了说明。

*表 165：LPTIM1 外部触发器连接*和*表 166：LPTIM2 外部触发器连接*对输入选择进行了说明。

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠、停止 0、停止 1、停止 2（仅限 LPTIM1）。

7.3.6 从定时器 (TIM1/TIM2) 到比较器 (COMP1/COMP2)

目的

高级控制定时器 (TIM1) 和通用定时器 (TIM2) 可用作 COMP1/COMP2 的消隐窗口输入。

*第 18.3.7 节：比较器输出消隐功能*对消隐功能进行了说明。

以下部分给出了消隐源：

- *第 18.6.1 节：比较器 1 控制和状态寄存器 (COMP1_CSR)* 位 20:18 BLANKING[2:0]
- *第 18.6.2 节：比较器 2 控制和状态寄存器 (COMP2_CSR)* 位 20:18 BLANKING[2:0]

触发信号

定时器输出信号 TIMx_Ocx 是 COMP1/COMP2 的消隐源输入。

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

7.3.7 从 USB 到定时器 (TIM2)

目的

USB (OTG_FS SOF) 可以生成面向通用定时器 (TIM2) 的触发信号。

*表 158：TIM2 内部触发连接*对 USB 与 TIM2 的连接进行了说明。

触发信号

由 USB_FS 帧起始生成的内部信号。

有效功耗模式

运行、睡眠。

7.3.8 从内部模拟到 ADC1

目的

内部温度传感器 (VTS)、内部参考电压 (VREFINT) 和 VBAT 监控通道连接到 ADC1 输入通道。

其依据包括：

- *第 16.2 节：ADC 主要特性*
- *第 16.3.11 节：通道选择（SQRx、JSQRx）*
- *第 16.3.30 节：温度传感器*
- *第 16.3.31 节：V_{BAT} 供电监测*
- *第 16.3.32 节：监测内部参考电压*

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

7.3.9 从比较器 (COMP1/COMP2) 到定时器 (TIM1/TIM2/TIM16/TIM17)

目的

比较器 (COMP1/COMP2) 输出值可连接到定时器 TIM1/TIM2/TIM16/TIM17 输入捕获或 TIMx_ETR 信号。

*第 24.3.4 节：外部触发输入*对与 ETR 的连接进行了说明。

比较器 (COMP1/COMP2) 输出值还可使用 I/O 的开漏连接，通过 GPIO 复用功能选择在输入引脚 TIMx_BKIN 或 TIMx_BKIN2 上为定时器 TIM1 生成断路输入信号。

以下部分给出了可能的连接：

- *第 24.4.23 节：TIM1 选择寄存器 1 (TIM1_OR1)*
- *第 24.4.28 节：TIM1 复用功能寄存器 2 (TIM1_AF2)*
- *第 25.4.22 节：TIM2 选择寄存器 1 (TIM2_OR1)*
- *第 25.4.23 节：TIM2 复用功能选择寄存器 1 (TIM2_AF1)*
- *第 26.2 节：TIM16/TIM17 主要特性*

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

7.3.10 从系统错误到定时器 (TIM1/TIM16/TIM17)

目的

CSS、CPU 硬性故障、RAM 奇偶校验错误、FLASH ECC 双重错误检测，PVD 可以向定时器 (TIM1/TIM16/TIM17) 以定时器打断的形式生成系统错误。

打断功能的目的是保护由这些定时器生成的 PWM 信号所驱动的功率器件。

以下部分列出了可能的断路源：

- *第 24.3.16 节：使用刹车功能* (TIM1)
- *第 26.3.11 节：使用刹车功能* (TIM16/TIM17)
- *图 256：TIM16/TIM17 框图*

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

7.3.11 从定时器 (TIM16/TIM17) 到 IRTIM

目的

通用定时器 (TIM16/TIM17) 输出通道 TIMx_OC1 用于生成红外信号输出的波形。

*第 28 节：红外接口 (IRTIM)*对此功能进行了介绍。

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

8 复位和时钟控制 (RCC)

8.1 复位

共有三种类型的复位，即：

1. 系统复位
2. 电源复位
3. 备份域复位

8.1.1 电源复位

只要发生以下事件之一，就会产生电源复位：

1. 欠压复位 (BOR)
2. 退出待机模式时
3. 退出关断模式时

欠压复位，包括上电或掉电复位 (POR/PDR)，将所有寄存器设置为其复位值，备份域除外。

退出待机模式时，V_{CORE} 域的所有寄存器都设置为其复位值。V_{CORE} 域外的寄存器（RTC，WKUP，IWDG，以及待机 / 关断模式控制）不受影响。

退出关断模式时，会产生欠压复位，将除备份域中的寄存器以外的所有寄存器全部复位。

8.1.2 系统复位

除了寄存器 RCC_CSR 和备份域中的寄存器外，系统复位会将其他全部寄存器都复位为复位值。

只要发生以下事件之一，就会产生系统复位：

1. NRST 引脚低电平（外部复位）
2. 窗口看门狗事件（WWDG 复位）
3. 独立看门狗事件（IWDG 复位）
4. 软件复位（SW 复位）（请参见*软件复位*）
5. 低功耗模式安全复位（请参见*低功耗模式安全复位*）
6. 选项字节加载器复位（请参见*选项字节加载器复位*）
7. 欠压复位

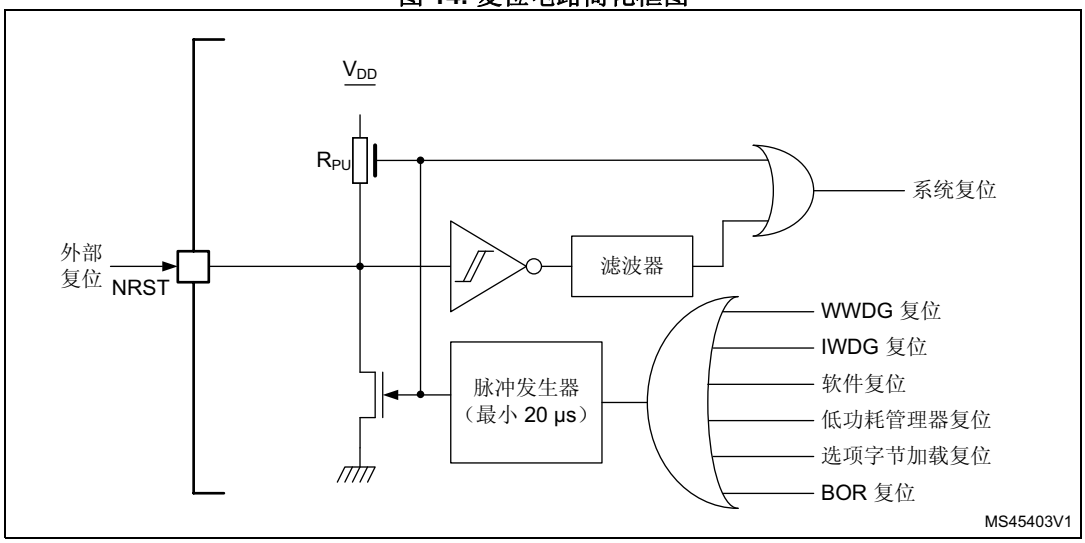
可通过查看控制 / 状态寄存器 (RCC_CSR) 中的复位标志确定复位源（请参见*第 8.4.31 节：RCC 控制 / 状态寄存器 (RCC_CSR)*）。

这些源均作用于 NRST 引脚，该引脚在复位过程中始终保持低电平。CPU1 RESET 复位入口向量通过 BOOT0 和 BOOT1 选择。

芯片内部的复位信号会向 NRST 引脚上输出一个低电平脉冲。脉冲发生器可确保每个内部复位源的复位脉冲都至少持续 20 µs。对于外部复位，在 NRST 引脚处于低电平时产生复位脉冲。

内部复位情况下，内部上拉 R_{PU} 失效，通过上拉电阻节约功耗。

图 14. 复位电路简化框图



软件复位

CPU1 应用中断和复位控制寄存器中的 SYSRESETREQ 位必须设置为强制器件上的软件复位（请参见 STM32F3xx/F4xx/L4xx Cortex®-M4 编程手册 PM0214）。

要强制对器件进行软件复位，必须将 CPU2 应用中断和复位控制寄存器中的 SYSRESETREQ 位置 1。

低功耗模式安全复位

为了防止关键应用错误地进入低功耗模式，提供了两种低功耗模式安全复位。如果在选项字节中使能，则在下列情况下会产生这种复位：

1. 进入待机模式：此复位的使能方式是清零用户选项字节中的 nRST_STDBY 位。使能后，只要成功执行进入待机模式序列，器件就将复位，而非进入待机模式。
2. 进入停止模式：此复位的使能方式是清零用户选项字节中的 nRST_STOP 位。使能后，只要成功执行进入停止模式序列，器件就将复位，而非进入停止模式。
3. 进入关断模式：此复位的使能方式是清零用户选项字节中的 nRST_SHDW 位。使能后，只要成功执行进入关断模式序列，器件就将复位，而非进入关断模式。

有关用户选项字节的详细信息，请参见*第 3.4.1 节：选项字节说明*。

选项字节加载器复位

当 FLASH_CR 寄存器中的 OBL_LAUNCH 位置 1 时，将产生选项字节加载器复位。此位用来通过软件启动选项字节加载。

8.1.3 备份域复位

备份域具有两个特殊复位。

只要发生以下事件之一，就会产生备份域复位：

1. 软件复位，通过将 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 BDRST 位置 1 触发。
2. 在电源 V_{DD} 和 V_{BAT} 都已掉电后，其中任何一个又再上电。

备份域复位仅影响 LSE 振荡器、RTC、备份寄存器和 RCC 备份域控制寄存器。

8.2 时钟

可以使用四种不同的时钟源来驱动系统时钟 (SYSCLK)：

- HSI16（高速内部）16 MHz RC 振荡器时钟
- MSI（多速内部）RC 振荡器时钟，100 kHz 到 48 MHz
- HSE 32 MHz 振荡器时钟
- PLL 时钟

从复位中启动后，MSI 用作系统时钟源，配置为 4 MHz。

器件具有以下附加时钟源：

- LSI1：32 kHz 低速内部 RC，该 RC 可用于驱动独立看门狗，也可选择驱动 RTC（用于停止和待机模式下的自动唤醒，不得用于 RF 系统自动唤醒）。
- LSI2：32 kHz 低速低漂移内部 RC，该 RC 可用于驱动独立看门狗，也可选择驱动 RTC（用于停止和待机模式下的自动唤醒或 RF 系统自动唤醒）。
- LSE：32.768 kHz 低速外部晶振，可选择驱动 RTC（用于停止和待机模式下的自动唤醒或 RF 系统自动唤醒），或实时时钟 (RTCCLK)。
- HSI48：RC 48 MHz 内部时钟源，可用于驱动 USB FS 和真 RNG。

对于每个时钟源来说，在未使用时都可单独开启或者关闭，以降低功耗。

可通过多个预分频器配置 AHB 频率（HCLK1、HCLK2 和 HCLK4）、高速 APB (PCLK2) 和低速 APB (PCLK1) 域。AHB（HCLK1 和 HCLK4）以及 PCLK1 和 PCLK2 域的最大频率为 64 MHz。AHB (HCLK2) 域的最大频率为 32 MHz。

大多数外设时钟均由其总线时钟（HCLK 和 PCLK）提供，但以下时钟除外：

- 48 MHz 时钟，用于 USB FS 和真 RNG。该时钟由以下任一时钟源提供（由软件选择）：
  - PLL VCO（PLLQCLK 仅在运行模式下可用）
  - PLLSAI1 VCO（PLLSAI1QCLK 仅在运行模式下可用）
  - MSI 时钟（仅在运行模式下可用）
  - HSI48 内部振荡器（仅在运行模式下可用）

  当 MSI 时钟利用 LSE 自动调整时，它可用于 USB FS 器件。

  HSI48 时钟可耦合到时钟恢复系统，从而为 USB FS（无晶振型解决方案）提供足够的时钟连接。
- 由以下三个时钟源之一提供的 ADC 时钟（由软件选择）：
  - 系统时钟（SYSCLK 仅在运行模式下可用）
  - PLL VCO（PLLAPCLK 仅在运行模式下可用）
  - PLLSAI1 VCO（PLLSAI1RCLK 仅在运行模式下可用）
- 由以下四个时钟源之一提供的 U(S)ART 时钟（由软件选择）：
  - 系统时钟（SYSCLK 仅在运行模式下可用）
  - HSI16 时钟（仅在运行和停止模式下可用）
  - LSE 时钟（仅在运行和停止模式下可用）
  - APB 时钟（PCLK 取决于 U(S)ART 映射到哪个 APB，仅在 CRun（通过 U(S)ARTxEN 使能时）和 CSleep （同样通过 U(S)ARTxSMEN 使能时）模式下可用）

  仅当时钟为 HSI16 或 LSE 时，支持从停止模式唤醒。
- 由以下三个时钟源之一提供的 I^2C 时钟（由软件选择）：
  - 系统时钟（SYSCLK 仅在运行模式下可用）
  - HSI16 时钟（仅在运行和停止模式下可用）
  - APB 时钟（PCLK 取决于 I^2C 映射到哪个 APB，仅在 CRun（通过 I2CxEN 使能时）和 CSleep（同样通过 I2CxSMEN 使能时）模式下可用）

  仅当时钟为 HSI 时，支持从停止模式唤醒。
- 由以下任一时钟源提供的 SAI1 时钟（由软件选择）：
  - 外部时钟，对于 SAI1，映射到 SAI1_EXTCLK
  - PLL VCO（PLLPCLK 仅在运行模式下可用）
  - PLLSAI1 VCO（PLLSAI1PCLK 仅在运行模式下可用）
  - HSI16 时钟（仅在运行模式下可用）

  SAI1 作为主设备时可以使用 HSI16 时钟，仅在接收模式下检测数据线上的音频活动。这样，在未接收到音频数据流时，不必开启 PLL，从而帮助降低功耗。
- 由以下五个时钟源之一提供的低功耗定时器 (LPTIMx) 时钟（由软件选择）：
  - LSI 时钟（LSI1 或 LSI2 仅在运行和停止模式下可用）
  - LSE 时钟（仅在运行和停止模式下可用）
  - HSI16 时钟（仅在运行模式下可用）
  - APB 时钟（PCLK 取决于 LPTIMx 映射到哪个 APB，仅在 CRun（通过 LPTIMxEN 使能时）和 CSleep（同样通过 LPTIMxSMEN 使能时）模式下可用）
  - 映射到 LPTIMx_IN1 上的外部时钟（仅在运行和停止模式下可用）

  仅当时钟为 LSI 或 LSE，或处于外部时钟模式时，支持停止模式的功能（包括唤醒）。

- 由以下三个时钟源之一提供的 RTC 和 LCD 时钟（由软件选择）：
  - LSE 时钟
  - LSI 时钟（LSI1 或 LSI2）
  - HSE 时钟 32 分频

  仅当时钟为 LSI 或 LSE 时，支持停止模式的功能（包括唤醒）。
- IWDG 的时钟，时钟由 LSI（LSI1 或 LSI2）时钟提供。
- 由以下三个时钟源之一提供的 RF 系统唤醒时钟（由软件选择）：
  - LSE 时钟
  - LSI 时钟（LSI2 和 LSI1 不得用于 RF 系统自动唤醒）
  - HSE 时钟 1024 分频

  仅当时钟为 LSI 或 LSE 时，支持停止模式的功能（包括唤醒）。
- 由以下两个时钟源之一提供的 RF 系统时钟（由硬件选择）：
  - HSI16 时钟
  - HSE 时钟

  仅当时钟为 HSI16（通过 STOPWUCK 选择 HSI16 作为唤醒时钟）时，才支持停止模式下的功能。

RCC 向 CPU1 系统定时器 (SysTick) 外部时钟馈送 8 分频的 AHB 时钟 (HCLK1)。SysTick 可使用此时钟作为时钟源，也可直接使用 CPU1 时钟 (HCLK1) 作为时钟源，具体可在 SysTick 控制和状态寄存器中配置。

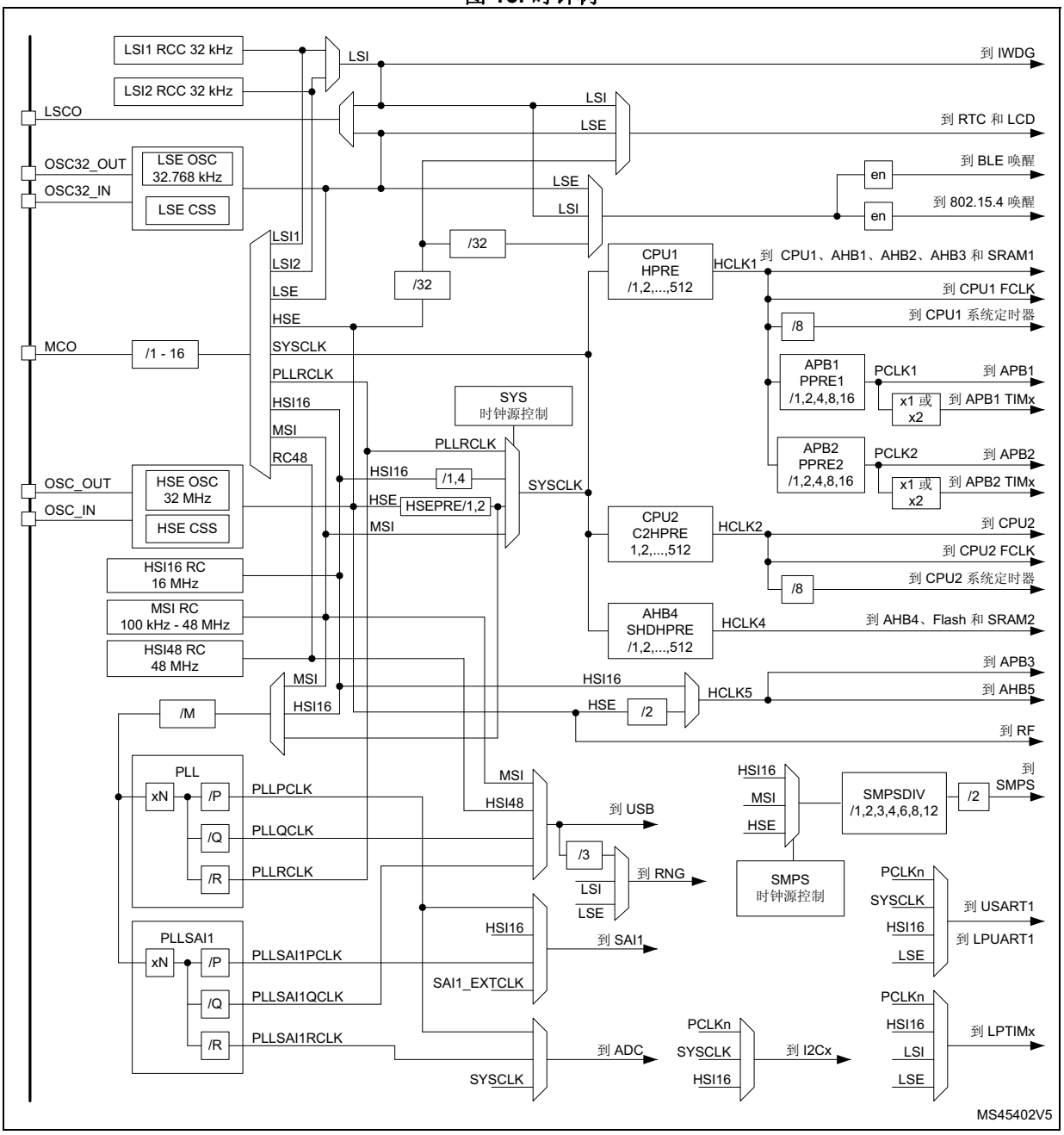
FCLK1 充当 CPU1 的自由运行时钟。有关详细信息，请参见 STM32F3 和 STM32F4 系列 Cortex-M4 编程手册 (PM0214)。

RCC 向 CPU2 系统定时器 (SysTick) 外部时钟馈送 8 分频的 AHB 时钟 (HCLK2)。SysTick 可使用此时钟作为时钟源，也可直接使用 CPU2 时钟 (HCLK2) 作为时钟源，具体可在 SysTick 控制和状态寄存器中配置。

FCLK2 充当 CPU2 的自由运行时钟。

图 15 详细绘制了时钟树，蓝色用于表示系统时钟、CPU 和共用总线的同步时钟，红色用于表示异步无线电外设总线时钟。

图 15. 时钟树



1. 有关内部和外部时钟源特性的所有详细信息，请参见器件数据手册的“电气特性”部分。
2. ADC 时钟可由 ADC 总线接口的 AHB 时钟除以一个可编程的因数（1、2 或 4）来提供。当可编程因数为“1”时，AHB 预分频器必须等于“1”。

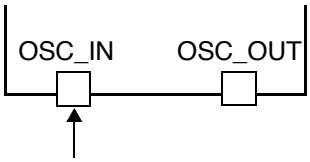
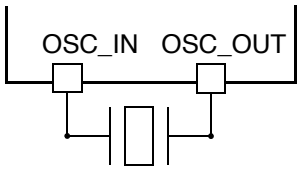
8.2.1 HSE 时钟

高速外部时钟信号 (HSE) 有 2 个时钟源（请参见*图 16*）：

- HSE 外部晶振
- HSE 用户外部时钟

晶振必须尽可能地靠近振荡器的引脚，以尽量减小输出失真和起振稳定时间。负载电容为集成式电容，可根据*频率调整*进行调整。

图 16. HSE 时钟源

时钟源	硬件配置
外部时钟	
晶振	

外部晶振（HSE 晶振）

32 MHz 外部振荡器的优点是主时钟精度非常高，任何无线电操作都必须使用该振荡器。

相关的硬件配置如*图 16* 所示。有关详细信息，请参见*数据手册*的电气特性部分。

RCC 时钟控制寄存器 (RCC_CR) 中的 HSERDY 标志指示 HSE 振荡器是否稳定。在启动时，硬件将此位置 1 后，此时钟才可以使用。如在 *RCC 时钟中断使能寄存器 (RCC_CIER)* 中使能中断，则可产生中断。

HSE 晶振可通过 *RCC 时钟控制寄存器 (RCC_CR)* 中的 HSEON 位开启或关闭。

外部源（HSE 旁路）

在此模式下（可通过将 *RCC 时钟控制寄存器 (RCC_CR)* 中的 HSEBYP 和 HSEON 位置 1 来选择），必须提供外部时钟源，频率为 32 MHz。必须使用占空比约为 45-55% 的外部时钟信号（正弦波）来驱动 OSC_IN 引脚，同时 OSC_OUT 引脚应保持未连接状态（请参见*图 16*）。

注： *有关引脚可用性的详细信息，请参见相应器件数据手册中的引脚排列部分。*

频率调整

由于生产工艺和所使用的晶振不同，不同芯片的 HSE 振荡器频率也不同。用户可以通过写入 *RCC 时钟 HSE 寄存器 (RCC_HSECR)* 中的 HSETUNE 位来调整应用中的 HSE 频率。HSE 频率可通过在 MCO 上输出 HSE 时钟来测量。

HSE 振荡器增益和感测可以通过 *RCC 时钟 HSE 寄存器 (RCC_HSECR)* 中的 HSEGMC 和 HSES 位控制。有关 HSE 校准流程，请参见 AN5042。

8.2.2 HSI16 时钟

HSI16 时钟信号是从 16 MHz 内部振荡器生成的。

HSI16 振荡器的优点是成本较低（无需使用外部元件）。此外，其启动速度也要比 HSE 晶振快，但即使校准后，其精度也不及外部晶振或陶瓷谐振器。

从停止模式（停止 0、停止 1 或停止 2）唤醒后，可选择 HSI16 时钟作为系统时钟，请参见 *第 8.3 节：低功耗模式*。HSI16 时钟还可作为备份时钟源（辅助时钟）使用，以防 HSE 晶振发生故障，请参见 *第 8.2.11 节*。

当使能 RF 系统时，应选择 HSI16 作为从停止模式唤醒后的系统时钟。

HSI16 时钟被用作从待机模式重启唤醒后的系统时钟。

校准

因为生产工艺不同，不同芯片的 RC 振荡器频率也不同，因此 ST 会对每个器件进行出厂校准，达到 T_A= 25°C 时 1% 的精度。

复位后，出厂校准值将载入 *RCC 内部时钟源校准寄存器 (RCC_ICSCR)* 的 HSICAL[7:0] 位中。

RC 振荡器速度会受到电压或温度变化的影响，用户可以使用 *RCC 内部时钟源校准寄存器 (RCC_ICSCR)* 中的 HSITRIM 位微调应用中的 HSI16 频率。

有关如何测量 HSI16 频率变化的更多详细信息，请参见 *第 8.2.21 节*。

RCC 时钟控制寄存器 (RCC_CR) 中的 HSIRDY 标志指示 HSI16 RC 是否稳定。在启动时，硬件将该位置 1 后，HSI16 RC 才可以使用。

HSI16 RC 可通过 *RCC 时钟控制寄存器 (RCC_CR)* 中的 HSION 位开启或关闭。

HSI16 信号还可作为备份时钟源（辅助时钟）使用，以防 HSE 晶振发生故障。请参见 *第 8.2.11 节*。

8.2.3 MSI 时钟

MSI 时钟信号是从内部 RC 振荡器生成的。频率范围可通过软件使用 *RCC 时钟控制寄存器 (RCC_CR)* 中的 MSIRANGE[3:0] 位进行选择。有 12 个频率范围可用：100 kHz、200 kHz、400 kHz、800 kHz、1 MHz、2 MHz、4 MHz（默认值）、8 MHz、16 MHz、24 MHz、32 MHz 和 48 MHz。

在从复位重启、从关断低功耗模式唤醒后，MSI 时钟被用作系统时钟。从复位重启后，MSI 频率被置为其默认值 4 MHz。请参见 *第 8.3 节：低功耗模式*。

从停止模式（停止 0、停止 1 或停止 2）唤醒后，可选择 MSI 时钟作为系统时钟。请参见*第 8.3 节：低功耗模式*。MSI RC 还可作为备份时钟源（辅助时钟）使用，以防 HSE 晶振发生故障。请参见*第 8.2.11 节*。

MSI RC 振荡器的优势在于可提供一个低成本（无外部元件）低功耗的时钟源。此外，当和 LSE 一起用于 PLL 模式时，它可提供一个非常精确的时钟源，该时钟源可用于 USB FS 器件，并且向 PLL 反馈，使系统以最大速率 64 MHz 运行。

RCC 时钟控制寄存器 (RCC_CR) 中的 MSIRDY 标志指示 MSI RC 是否稳定。在启动时，硬件将此位置 1 后，MSI RC 输出时钟才可以使用。MSI RC 可通过 *RCC 时钟控制寄存器 (RCC_CR)* 中的 MSION 位开启或关闭。

利用 LSE 进行硬件自动校准（PLL 模式）

当应用中存在 32.768 kHz 的外部振荡器时，可通过设置 *RCC 时钟控制寄存器 (RCC_CR)* 中的 MSIPLLEN 位将 MSI 配置为 PLL 模式。当配置为 PLL 模式时，MSI 利用 LSE 自动校准。该模式可用于所有 MSI 频率范围。48 MHz 时，处于 PLL 模式的 MSI 可用于 USB FS 器件，不需要外部高速晶振。

软件校准

因为生产工艺不同，不同芯片的 MSI RC 振荡器频率也不同，因此 ST 会对每个器件进行出厂校准，使器件在环境温度 (T_A) 为 25°C 时达到 1% 的精度。复位后，出厂校准值将载入 *RCC 内部时钟源校准寄存器 (RCC_ICSCR)* 的 MSICAL[7:0] 位中。如果应用受到电压或温度变化影响，则这可能也会影响到 RC 振荡器的速度。用户可通过 RCC_ICSCR 寄存器中的 MSITRIM[7:0] 位对应用中的 MSI 频率进行微调。有关如何测量 MSI 频率变化的更多详细信息，请参见*第 8.2.21 节*。

8.2.4 HSI48 时钟

HSI48 时钟信号由内部 48 MHz RC 振荡器生成，可直接用于 USB 和随机数发生器（真 RNG）。

内部 48 MHz RC 振荡器主要用于通过特殊的时钟恢复系统 (CRS) 电路向 USB 外设提供高精度时钟。CRS 可使用 USB SOF 信号、LSE 或外部信号自动而快速地实时调整振荡器频率。系统进入停止或待机模式后，CRS 会立即被禁止。未使用 CRS 时，HSI48 RC 振荡器以其默认频率运行，具体默认频率值取决于生产工艺。

有关如何配置和使用 CRS 外设的更多详细信息，请参见*第 40 节：时钟恢复系统 (CRS)*。

时钟恢复寄存器 (RCC_CRRCR) 中的 HSI48RDY 标志指示 HSI48 RC 振荡器是否稳定。在启动时，硬件将此位置 1 后，HSI48 RC 振荡器才可以使用。

HSI48 可通过时钟恢复寄存器 (RCC_CRRCR) 中的 HSI48ON 位开启或关闭。

8.2.5 PLL

器件嵌入了以下 PLL：PLL 和 PLLSAI1。每个 PLL 提供了多达 3 个独立输出。内部 PLL 可用来倍频 HSI、HSE 或 MSI 输出时钟频率。PLL 输入频率必须介于 4 MHz 到 16 MHz 之间。所选时钟源被可编程因数 PLLM（因数为 1 到 8）分频，提供一个在所需输入范围内的时钟频率。请参见*图 15：时钟树*和 *RCC PLL 配置寄存器 (RCC_PLLCFGR)*。

PLL 配置（对输入时钟和倍频因数进行选择）必须在使能 PLL 之前完成。当 PLL 使能后，这些参数就不能再更改。

要修改 PLL 配置，请按照以下步骤操作：

1. 通过将 *RCC 时钟控制寄存器 (RCC_CR)* 中的 PLLxxxON 位设置为 0 禁止 PLL。
2. 等待，直至 PLLxxxRDY 清零。PLL 现已完全停止。
3. 根据需要更改参数。
4. 通过将 PLLxxxON 设置为 1 再次使能 PLL。
5. 通过配置 *RCC PLL 配置寄存器 (RCC_PLLCFGR)* 或 *RCC PLLSAI1 配置寄存器 (RCC_PLLSAI1CFGR)* 中的 PLLPEN、PLLQEN、PLLREN，使能所需要的 PLL 输出。

如果已在 *RCC 时钟中断使能寄存器 (RCC_CIER)* 中使能中断，则 PLL 就绪时便可产生中断。

PLL 输出频率不可超过 64 MHz。

每个 PLL 输出时钟的使能位（PLLPEN、PLLQEN 和 PLLREN）可随时修改，而不必停止相应的 PLL。如果 PLLRCLK 用作系统时钟，则 PLLREN 不可被清零。

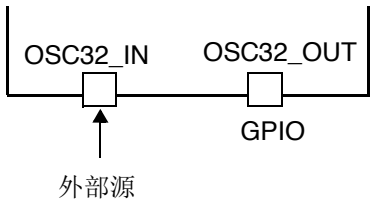
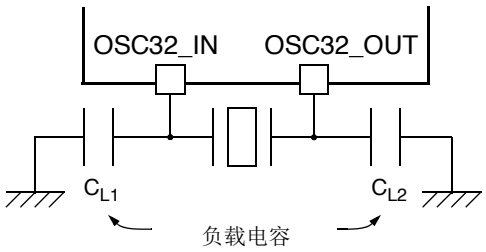
在范围 2 中，PLL M 分频器输入时钟不得超过 16 MHz。*RCC 时钟控制寄存器 (RCC_CR)* 中的 HSEPRE 位应选择 HSE 2 分频。

8.2.6 LSE 时钟

LSE 晶振是 32.768 kHz 低速外部晶振或陶瓷谐振器，可作为实时时钟 (RTC) 的时钟源来提供时钟 / 日历或其他定时功能，具有功耗低且精度高的优点。

谐振器和负载电容必须尽可能地靠近振荡器的引脚，以尽量减小输出失真和起振稳定时间。负载电容值必须根据所选振荡器的不同做适当调整。

图 17. LSE 时钟源

时钟源	硬件配置
外部时钟	
晶振 / 陶瓷谐振器	

LSE 晶振通过 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 LSEON 位开启和关闭。使用 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 LSEDRV[1:0] 位，可在运行时更改晶振驱动强度，以实现稳健性、短启动时间和低功耗之间的最佳平衡。当 LSE 为 ON 时，LSE 驱动可降低为更低的驱动能力 (LSEDRV=00)。但是，当选择了 LSEDRV，如果 LSEON=1 则不能提高驱动能力。

RCC 备份域控制寄存器 (RCC_BDCR) 的 LSERDY 标志指示 LSE 晶振是否稳定。在启动时，硬件将此位置 1 后，LSE 晶振输出时钟信号才可以使用。如在 *RCC 时钟中断使能寄存器 (RCC_CIER)* 中使能中断，则可产生中断。

外部源（LSE 旁路）

在此模式下（可通过将*睡眠模式下的 RCC AHB1 外设时钟使能寄存器 (RCC_AHB1SMENR)* 中的 LSEBYP 和 LSEON 位置 1 来选择），必须提供外部时钟源，频率最高为 1 MHz。必须使用占空比约为 50% 的外部时钟信号（方波、正弦波或三角波）来驱动 OSC32_IN 引脚，同时 OSC32_OUT 引脚可用作 GPIO，请参见*图 17*。

8.2.7 LSI1 时钟

LSI1 RC 可作为低功耗时钟源在停止模式和待机模式下保持运行，供独立看门狗 (IWDG)、RTC、LCD 和 RF 唤醒使用。时钟频率为 32 kHz。有关详细信息，请参见数据手册的电气特性部分。

LSI1 RC 可通过 *RCC 控制 / 状态寄存器 (RCC_CSR)* 中的 LSI1ON 位开启和关闭。

RCC 控制 / 状态寄存器 (RCC_CSR) 中的 LSI1RDY 标志指示 LSI1 振荡器是否稳定。在启动时，硬件将此位置 1 后，此时钟才可以使用。如在 *RCC 时钟中断使能寄存器 (RCC_CIER)* 中使能中断，则可产生中断。

8.2.8 LSI2 时钟

LSI2 RC 可作为低漂移低功耗时钟源在停止模式和待机模式下保持运行，供独立看门狗 (IWDG)、RTC、LCD 和 RF 唤醒使用。时钟频率在 32 kHz 左右。有关详细信息，请参见数据手册的电气特性部分。

LSI2 RC 可通过 *RCC 控制/状态寄存器 (RCC_CSR)* 中的 LSI2ON 位开启和关闭。

RCC 控制/状态寄存器 (RCC_CSR) 中的 LSI2RDY 标志指示 LSI2 振荡器是否稳定。在启动时，硬件将此位置 1 后，此时钟才可以使用。如在 *RCC 时钟中断使能寄存器 (RCC_CIER)* 中使能中断，则可产生中断。

当 NRST 引脚上发生任何系统复位后，在使用 LSI2 之前，应为其振荡器提供微调信息。应通过固件将 LSI2 微调信息从 ST 生产微调 Flash 位置复制到寄存器 RCC_SCR 中的 LSI2TRIM 位置。

LSI2 微调参数

ST 生产 LSI2 微调值可从 Flash 的以下位置获取：

基址：0x1FFF 7548

复位值：0x000X，其中 X 是出厂前编程的

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTLSI2TRIM[3:0]			
												r	r	r	r

8.2.9 系统时钟 (SYSCLK) 选择

可以使用四种不同的时钟源来驱动系统时钟 (SYSCLK)：

- MSI 振荡器
- HSI16 振荡器
- HSE 振荡器
- PLLRCLK

范围 1 中的系统时钟最大频率为 64 MHz。系统复位后，选择 MSI 振荡器（为 4 MHz）作为系统时钟。在直接使用 HSI 或者通过 PLL 使用时钟源来作为系统时钟时，该时钟源无法停止。

只有在目标时钟源已就绪时（时钟在启动延迟或 PLL 锁相后稳定时），才可从一个时钟源切换到另一个。如果选择尚未就绪的时钟源，则切换在该时钟源就绪时才会进行。*RCC 内部时钟源校准寄存器 (RCC_ICSCR)* 中的状态位指示哪个（些）时钟已就绪，以及当前哪个时钟正充当系统时钟。

从待机模式唤醒时，HSI16 被选为系统时钟。

在范围 2 中，系统频率不得超过 16 MHz。*RCC 时钟控制寄存器 (RCC_CR)* 中的 HSEPRE 位应选择 HSE 2 分频。

8.2.10 时钟源频率与电压调节

下表给出了不同产品电压范围下不同的时钟源频率。

表 37. 最大时钟源频率

产品电压范围	时钟频率			
	MSI	HSI	HSE	PLL/PLLSAI1
范围 1	48 MHz	16 MHz	32 MHz	64 MHz （VCO 最大值 = 344 MHz）
范围 2	16 MHz 范围	16 MHz	32 MHz(1)	16 MHz （VCO 最大值 = 128 MHz）

1. HSEPRE 应选择 2 分频。

8.2.11 HSE 时钟安全系统 (CSS)

时钟安全系统可通过软件激活。激活后，时钟监测器将在 HSE 振荡器启动延迟后使能，并在此振荡器停止时被关闭。

如果 HSE 时钟发生故障，HSE 振荡器将自动禁止，一个时钟故障事件将发送到高级控制定时器（TIM1 和 TIM16/17）的断路输入，并且同时还将生成一个中断来向软件通知此故障（时钟安全系统中断 CSSI），以使 MCU 能够执行救援操作。CSSI 与 CPU1 和 CPU2 NMI（不可屏蔽中断）异常向量相链接。

注： *当 HSE CSS 使能后，如果 HSE 时钟发生故障，则会产生 CSS 中断，并会自动产生 NMI。NMI 将无限期执行，除非将 CSS 中断挂起位清零。因此，在 NMI ISR 中，用户必须将 RCC 时钟中断清零寄存器 (RCC_CICR) 中的 CSSC 位置 1，以清除 CSS 中断。*

如果直接或间接使用 HSE 振荡器作为系统时钟（间接是指它用作 PLL 输入时钟，PLL 时钟用作系统时钟），检测到故障时会导致系统时钟切换到 MSI 或 HSI16 振荡器（取决于 *RCC 时钟配置寄存器 (RCC_CFGR)* 中的 STOPWUCK 配置），并禁止 HSE 振荡器。当故障发生时，若 HSE 时钟（分频或不分频）为正在用作系统时钟的 PLL 时钟输入，则也会禁用该 PLL。

8.2.12 LSE 上的时钟安全系统 (LSECSS)

可由软件通过写入 *RCC 控制/状态寄存器 (RCC_CSR)* 中的 LSECSSON 位，来激活 LSE 上的时钟安全系统。仅可通过硬件复位、RTC 软件复位、或在检测到 LSE 故障后禁止该位。LSE 和 LSI 使能（LSEON 和 LSIxON 使能）并就绪（LSERDY 和 LSIRDY 由硬件置 1）后，以及通过 RTCSEL 选择 RTC 时钟后，必须写入 LSECSSON。

LSE 上的 CSS 适用于除 VBAT 外的所有模式，以及除上电复位外的所有系统复位。如果在外部 32 kHz 振荡器上检测到故障，则不会再向 RTC 提供 LSE 时钟，但不会对寄存器进行硬件操作。如果 MSI 处于 PLL 模式，则禁止该模式。

在待机模式下，会产生唤醒。在其他模式下，可发送中断来唤醒软件（请参见 *RCC 时钟中断使能寄存器 (RCC_CIER)*、*RCC 时钟中断标志寄存器 (RCC_CIFR)* 和 *RCC 时钟中断清零寄存器 (RCC_CICR)*）。

软件随后必须禁止 LSECSSON 位并停止出现故障的 32 kHz 振荡器（禁止 LSEON），并更改 RTC 时钟源（无时钟、LSI 或 HSE，通过 RTCSEL），或者采取任何必要措施来确保应用的安全。

8.2.13 LSI 源选择

可以选择 LSI1 或 LSI2 作为系统中的 LSI 源。在通过 LSI2ON 寄存器位开启 LSI2 后，只要 LSI2 就绪就会被选为 LSI 源。

要在不中断 LSI 时钟的情况下从 LSI2 切换到 LSI1，首先应通过 LSI1ON 寄存器位开启 LSI1。在通过 LSI2ON 寄存器位禁止 LSI2 之前，固件应通过 LSI1RDY 寄存器位验证 LSI1 是否就绪。

8.2.14 SMPS 降压转换器时钟

SMPS 降压转换器时钟源可以是 MSI、HSI16 或 HSE 时钟。选择方式是编程 *RCC SMPS 降压转换器控制寄存器 (RCC_SMPSCR)* 中的 SMPSSEL。配置系统时，必须始终确保 SMPS 降压转换器时钟频率介于 4 MHz 和 8 MHz 之间。应在 SMPSDIV 中设置适当的分频系数。在使用时钟驱动 SMPS 降压转换器之前，还需进一步进行固定的 2 分频。

系统复位后，将选择 HSI16 作为 SMPS 降压转换器时钟。

当进入停止 1、停止 2、待机或关断等低功耗模式时，通过 *RCC SMPS 降压转换器控制寄存器 (RCC_SMPSCR)* 中的 SMPSSEL 所选的 SMPS 降压转换器时钟源应与 *RCC 时钟配置寄存器 (RCC_CFGR)* 中的 SW 所选的系统时钟源相同。

从待机模式唤醒并为 V_{CODE} 上电时，将选择 HSI16 作为 SMPS 降压转换器时钟，与 SMPSSEL 中的选择无关。

当无线电系统处于活动状态时，将选择 HSE 作为 SMPS 降压转换器时钟，与 SMPSSEL 中的选择无关。

使用的 SMPS 降压转换器时钟源由 *RCC SMPS 降压转换器控制寄存器 (RCC_SMPSCR)* 中的 SMPSSWS 指示。

只有在目标时钟源已就绪时（时钟在启动延迟后稳定时），才可从一个时钟源切换到另一个。如果选择尚未就绪的时钟源，则切换在该时钟源就绪时才会进行。*RCC 内部时钟源校准寄存器 (RCC_ICSCR)* 中的状态位指示哪个（些）时钟已就绪，SMPSSWS 位指示当前哪个时钟正充当 SMPS 降压转换器时钟。

SMPSDIV 中的分频系数取决于 SMPSSEL 中所选的 SMPS 降压转换器时钟源。当在 SMPSSEL 中选择 MSI 时，还可考虑使用 MSIRANGE 来确定所选的 MSI 频率。这样，便可在保持 SMPS 降压转换器时钟频率不变的情况下从一个时钟源切换到另一个时钟源。当支持的 SMPS 降压转换器时钟 MSIRANGE 置 1 时，仅应选择 MSI 作为 SMPS 降压转换器时钟源。

当 RF 系统处于活动状态时，硬件将自动选择 HSE 源与当前 SMPSDIV 分频比。

表 38. SMPS 降压转换器时钟源的选择和分频

SMPSSEL	HSI16	MSI				HSE
SMPSDIV	16 MHz	16 MHz	24 MHz	32 MHz	48 MHz	32 MHz
00	1 (8 MHz)	1 (8 MHz)	保留	2 (8 MHz)	3 (8 MHz)	2 (8 MHz)
01	2 (4 MHz)	2 (4 MHz)	3 (4 MHz)	4 (4 MHz)	6 (4 MHz)	4 (4 MHz)
10	保留					
11	保留					

8.2.15 ADC 时钟

ADC 时钟由系统时钟或 PLL/PLLSAI1 输出提供。此时钟可达 64 MHz，并可通过配置 ADC1_CCR 寄存器使用以下预分频值进行分频：1、2、4、6、8、10、12、16、32、64、128 或 256。它与 AHB 时钟异步。或者，ADC 时钟可由 ADC 总线接口的 AHB 时钟除以一个可编程的因数（1、2 或 4）来提供。该可编程的因数使用 ADC1_CCR 中的 CKMODE 位域进行配置。

如果可编程因数为“1”，必须将 AHB 预分频器设置为“1”。

8.2.16 RTC 时钟

RTCCLK 时钟源可以是 HSE/32、LSE 或 LSI 时钟。选择方式是编程 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 RTCSEL[1:0] 位。所做的选择只能通过复位备份域的方式修改。配置系统时，必须始终使 PCLK 频率大于或等于 RTCCLK 频率，以便 RTC 正常工作。

LSE 时钟位于备份域中，而 HSE 和 LSI 时钟则不是。因此：

- 如果选择 LSE 作为 RTC 时钟：
  - 只要 V_{BAT} 电源保持工作，即使 V_{DD} 电源关闭，RTC 仍可继续工作。
- 如果选择 LSI 作为 RTC 时钟：
  - 在 V_{DD} 电源掉电时，RTC 的状态将不能保证。
- 如果使用按预分频器进行分频的 HSE 时钟作为 RTC 时钟：
  - 如果 V_{DD} 电源掉电或者内部调压器关闭（切断 V_{CORE} 域的供电），则 RTC 的状态将不能保证。

如果 RTC 时钟为 LSE 或 LSI，则 RTC 在系统复位后仍可获得时钟并保持正常工作。

8.2.17 定时器时钟

定时器时钟频率由硬件自动定义为固定值。分为两种情况：

1. 如果 APB 预分频器等于 1，定时器时钟频率等于 APB 域的频率。
2. 否则，等于 APB 域的频率的两倍 (×2)。

8.2.18 看门狗时钟

如果独立看门狗 (IWDG) 已通过硬件选项或软件访问的方式启动，则将强制开启 LSI 时钟。

启动 IWDG 时，如果 LSI1 振荡器和 LSI2 振荡器均未使能，则将强制开启 LSI1 振荡器。在 LSI1 振荡器稳定后，时钟将提供给 IWDG。

通过 LSI2ON 位开启 LSI2 时，LSI 时钟将从 LSI1 切换到 LSI2。

通过 LSI2ON 禁止 LSI2 时（当 IWDG 运行时），将强制开启 LSI1 振荡器。LSI 时钟会在 LSI1 准备就绪后切换到 LSI1，之后 LSI2 才会停止。

8.2.19 真随机数时钟

真随机数发生器 (RNG) 种子时钟由 USB 选择的时钟提供。

8.2.20 时钟输出功能

- MCO

  微控制器时钟输出 (MCO) 功能可将时钟输出到外部 MCO 引脚上。可选择以下时钟信号之一作为 MCO 时钟：

  – LSI1（在运行和停止模式下可用）
  – LSI2（在运行和停止模式下可用）
  – LSE（在运行和停止模式下可用）
  – MSI（在运行模式下可用）
  – HSI（由 HSION 使能时，在运行模式下可用）
  – HSI48（在运行模式下可用）
  – HSE（在运行模式下可用）
  – PLLRCLK（在运行模式下可用）
  – SYSCLK（在运行模式下可用）

  此选择由 *RCC 时钟配置寄存器 (RCC_CFGR)* 中的 MCOSEL[3:0] 位控制。所选时钟可以通过 *RCC 时钟配置寄存器 (RCC_CFGR)* 的 MCOPRE[2:0] 位域进行分频。

  MCO 上的时钟仅在运行和停止模式下可用，在待机和关断模式下不可用。

- LSCO

  LSCO 输出可将低速时钟输出到外部 LSCO 引脚上：

  – LSI（LSI1 或 LSI2）
  – LSE

  此选择由 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 LSCOSEL 控制，并通过 LSCOEN 使能。

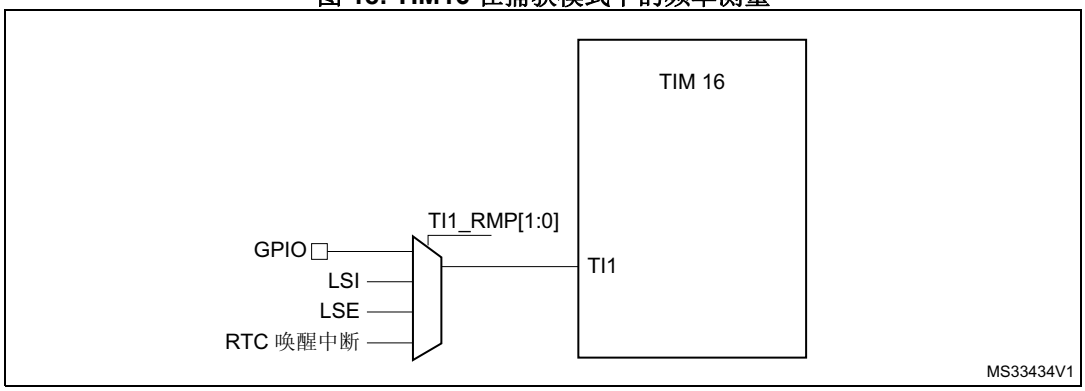
  LSCO 上的时钟在运行和停止模式下可用，其中一个 GPIO 上的时钟在待机和关断模式可用。

必须在复用功能模式下对相应 GPIO 端口的配置寄存器进行编程。

8.2.21 基于 TIM16/TIM17 的内部 / 外部时钟测量

所有板上时钟源的频率都可通过对 TIM16 或 TIM17 通道 1 的输入捕获进行间接测量，如*图 18*和*图 19* 所示。

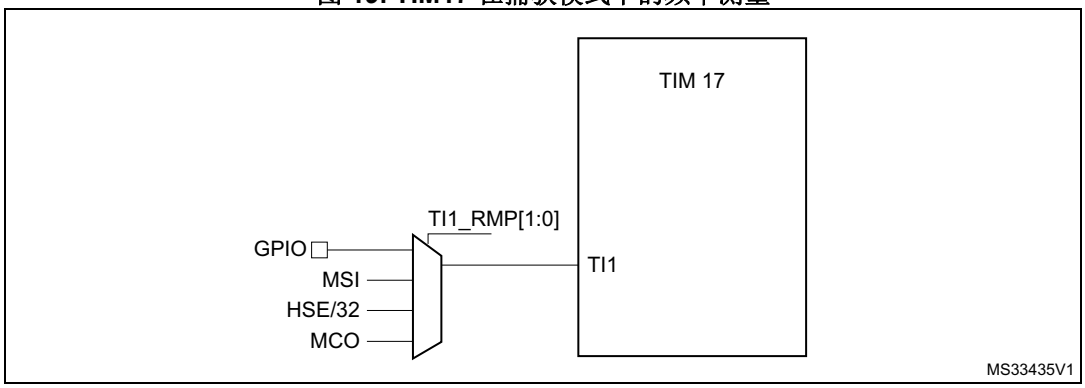
图 18. TIM16 在捕获模式下的频率测量



TIM16 的输入捕获通道可以是 MCU 的 GPIO 线或内部时钟。此选择通过 TIM16_OR 寄存器的 TI1_RMP[1:0] 位执行。可能的情况如下：

- TIM16 通道 1 连接到 GPIO。请参见器件数据手册中的复用功能映射。
- TIM16 通道 1 连接到 LSI 时钟。
- TIM16 通道 1 连接到 LSE 时钟。
- TIM16 通道 1 连接到 RTC 唤醒中断信号。在这种情况下，应使能 RTC 中断。

图 19. TIM17 在捕获模式下的频率测量



定时器 17 的输入捕获通道可以是 MCU 的 GPIO 线或内部时钟。此选择通过 TIM17_OR 寄存器的 TI1_RMP[1:0] 位执行。可能的情况如下：

- TIM17 通道 1 连接到 GPIO。请参见器件数据手册中的复用功能映射。
- TIM17 通道 1 连接到 MSI 时钟。
- TIM17 通道 1 连接到 HSE/32 时钟。
- TIM17 通道 1 连接到微控制器时钟输出 (MCO)，此选择由时钟配置寄存器 (RCC_CFGR) 的 MCOSEL[3:0] 位控制。

HSI16 和 MSI 校准

对于 TIM16，将 LSE 连接到通道 1 输入捕获的主要目的是为了能够精确测量 HSI16 和 MSI 系统时钟（为此，应将 HSI16 或 MSI 用作系统时钟源）。借助 LSE 信号连续边沿之间的 HSI16（或 MSI）时钟计数数量，即可对内部时钟周期进行测量。利用 LSE 的高精度（通常为几十 ppm），用户能以同一分辨率测定内部时钟频率，并可通过对时钟源进行微调来补偿由生产工艺造成的和 / 或与温度和电压相关的频率偏差。

MSI 和 HSI16 振荡器均设有针对此目的的专用校准位，且支持用户访问。

其基本原理是基于相对的测量（例如，HSI/LSE 比）：因此，精度与两个时钟源之比紧密相关。比率越大，测量效果越好。

如果 LSE 不可用，为了尽可能达到最精确的校准，HSE32 将是更好的选择。

但是，如果 MSI 时钟频率较低（通常低于 1 MHz），测量精度会不够好。在这种情况下，建议：

- 累加连续多次捕获的结果
- 使用定时器的输入捕获预分频值（最多每 8 个周期进行 1 次捕获）
- 使用 RTC 唤醒中断信号（如果 RTC 的时钟由 LSE 提供）作为通道 1 输入捕获的输入。这样可提高测量精度。为此，必须使能 RTC 唤醒中断。

LSI 校准

LSI 校准将遵循与 HSI 相同的模式，只是参考时钟有所不同。需要将 LSI 时钟连接到 TIM16 的通道 1 输入捕获。然后将 HSE 定义为系统时钟源，借助 LSI 信号连续边沿之间的时钟计数数量，即可对内部低速时钟周期进行测量。

基本原理基于相对的测量（例如，HSE/LSI 比）。因此，精度与两个时钟源之比紧密相关，比率越大，测量结果越精确。

8.2.22 外设时钟使能

每个 CPU 的大多数外设总线和内核时钟均可单独使能。RCC_AHBxENR 和 RCC_APBxENRy 为 CPU1 使能外设时钟，而 RCC_C2_AHBxENR 和 RCC_C2_APBxENR 为 CPU2 使能外设时钟。外设时钟将遵循相关 CPU 的状态来使能，具体请参见*表 39*。

CPU 睡眠模式期间的外设总线时钟活动由外设时钟 CPU 睡眠模式使能位控制（对于 CPU1，为 RCC_AHBxSMENR 和 RCC_APBxSMENRy 寄存器；对于 CPU2，则为 RCC_C2_AHBxSMENR 和 RCC_C2_APBxSMENRy 寄存器）。睡眠模式期间的外设总线时钟将遵循相关 CPU 的状态来使能，具体请参见*表 39*。

表 39. 外设时钟使能

外设					
xxxEN	**xxxSMEN**	**CPU 模式**	**系统模式**	**总线时钟**	**内核时钟(1)**
0	x	任意	任意	停止	停止
1		CRun	运行	工作	工作
	0	CSleep 和 CStop	运行	停止	工作
	1	CSleep	运行	工作	工作
		CStop	运行	停止	工作
			停止	停止	选择以下时钟源时工作： – HSI16 – LSI – LSE
					选择以下时钟源时停止： – 总线时钟 – SYSCLK – PLL 时钟 – MSI – HSI48 – PLLSAI1 时钟
			待机或关断	停止	停止

1. 只有外设 SAI、I2C、LPTIM、USART、LPUART、USB FS、真 RNG 和 ADC 具有内核时钟。

当外设总线时钟未激活时，不支持外设寄存器进行读写访问。

当外设总线时钟激活时，无论哪个 CPU 在其 CPU xxxEN 位中使能了外设总线时钟，两个 CPU 都可以访问外设。但是，当外设总线时钟仅由一个 CPU 使能，并且该 CPU 进入低功耗模式时，外设总线时钟将停止（也取决于该 CPU xxxSMEN 设置），同时另一个 CPU 将无法再访问外设。因此，最好通过 CPU 专用时钟使能来使能外设总线时钟。

当外设内核时钟未激活时，外设功能将停止。

使能位的同步机制可为外设产生无干扰时钟。使能位置 1 后，在时钟激活前存在时长为 2 个时钟周期的延迟。

注意： 使能外设时钟后，软件必须等待一段延迟才能访问外设寄存器。

注： *当 BLE-IP 激活时，BLE 总线和 SRAM2 总线接口时钟也会一同激活。*

8.3 低功耗模式

- 可通过软件禁止 AHB 和 APB 外设时钟，包括 DMA 时钟。
- 在睡眠和低功耗睡眠模式下，CPU 时钟停止工作。在睡眠模式下，可通过软件停止存储器接口时钟（Flash、SRAM1 和 SRAM2 接口）。当连接到 AHB-APB 总线桥时钟的所有外设时钟均被禁止时，睡眠模式期间将通过硬件禁止 AHB-APB 总线桥时钟。
- 在停止模式（停止 0、停止 1 和停止 2）下，将停止 V_{CORE} 域中的大多数时钟，并禁止 PLL、MSI 和 HSE 振荡器。当允许从停止模式唤醒的 IP（USART1、LPUART1、I2C1 和 I2C3）发出请求时，HSI16 可以保持运行。

  即使 MCU 处于停止模式（如果选择 HSI16 作为该外设的时钟源），所有 U(S)ART、LPUART 和 I^2C 也均可使能 HSI16 振荡器。

  当系统处于停止模式（如果选择 LSE 作为该外设的时钟源）并使能 LSE 振荡器 (LSEON) 时，所有 U(S)ART、LPUART 和 LPTIM 也均可由 LSE 振荡器驱动。在这种情况下，LSE 在停止模式下始终保持开启（它们无法开启 LSE 振荡器）。

  当系统处于停止模式（如果选择 LSI 作为该外设的时钟源）并使能 LSI 振荡器（LSI1ON 或 LSI2ON）时，LPTIM 也可由 LSI 振荡器驱动。
- 在待机和关断模式下，将停止 V_{CORE} 域中的所有时钟，并禁止 PLL、HSI、MSI 和 HSE 振荡器。

通过将 DBGMCU_CR 寄存器中的 DBG_SLEEP、DBG_STOP 或 DBG_STANDBY 位置 1，可以覆盖低功耗模式以进行调试。此外，可使用 EXTI CDBGPWRUPREQ 事件以允许在停止模式下进行调试（请参见*表 40*）。

表 40. 单核低功耗调试配置(1)

模式	CDBGPWRUPREQ	DBGMCU			调试
	CPU1	DBG_STANDBY	DBG_STOP	DBG_SLEEP	CPU1
睡眠	X	X	X	X	使能
停止 0 和停止 1	禁止	X	禁止		禁止
	使能				使能
停止 0、停止 1 和停止 2	X		使能		使能
待机	X	禁止			禁止
		使能			使能

1. X = 不使用。

当退出停止模式（停止 0、停止 1 或停止 2）时，系统时钟为 MSI 或 HSI，具体取决于 RCC_CFGR 寄存器中 STOPWUCK 位的软件配置。使能 RF 系统时，应选择 HSI16 作为 STOPWUCK。如果 STOPWUCK 在退出停止模式时选择 HSI16 时钟，则将复位 C2HPRE 以选择 1 分频。MSI 振荡器的频率（范围和用户微调）是进入停止模式之前配置的频率。HSI16 的用户微调将保留。如果 MSI 在进入停止模式之前处于 PLL 模式，则即使 LSE 在停止模式期间保持开启，也必须在唤醒后等待 PLL 模式稳定时间。在使能 SMPS 降压转换器的情况下进入停止 0 模式之前，必须通过将 HSIKERON 寄存器位置 1 使 HSI16 保持开启状态。

退出待机模式时，系统时钟为 HSI。

退出关断模式时，系统时钟为 MSI。从关断模式唤醒时的 MSI 频率为 4 MHz。用户微调将丢失。

如果正在执行 Flash 编程操作，则将延迟到 Flash 接口访问结束后再进入停止、待机和关断模式。如果正在访问 APB 域，则将延迟到 APB 访问结束后再进入停止、待机和关断模式。

8.4 RCC 寄存器

8.4.1 RCC 时钟控制寄存器 (RCC_CR)

RCC clock control register

偏移地址：0x000

复位值：0x0000 0061（在 POR 复位后），0x0000 0160（从待机模式唤醒复位后）。（HSEBYP 通过 NRST 焊盘复位，而非通过从待机模式唤醒来复位）。

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	PLL SAI1RDY	PLL SAI1ON	PLL RDY	PLLON	Res.	Res.	Res.	HSEP RE	CSS ON	HSE BYP	HSE RDY	HSEON
				r	rw	r	rw				rw	rs	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	HSI KERDY	HSI ASFS	HSI RDY	HSI KERON	HSION	MSIRANGE[3:0]				Res.	MSI PLLEN	MSI RDY	MSION
			r	rw	r	rw	rw	rw	rw	rw	rw		rw	r	rw

位 31:28 保留，必须保持复位值。

位 27 **PLLSAI1RDY**：SAI PLL 时钟就绪标志 (SAI PLL clock ready flag)
由硬件置 1，用于指示 PLLSAI1 已锁定。
0：PLLSAI1 未锁定
1：PLLSAI1 已锁定

位 26 **PLLSAI1ON**：SAI PLL 使能 (SAI PLL enable)
由软件置 1 和清零，用于使能 PLLSAI1。
当进入停止、待机和关断模式时由硬件清零。
0：PLLSAI1 关闭
1：PLLSAI1 开启

位 25 **PLLRDY**：系统 PLL 时钟就绪标志 (System PLL clock ready flag)
由硬件置 1，用于指示系统 PLL 已锁定。
0：PLL 未锁定
1：PLL 已锁定

位 24 **PLLON**：系统 PLL 使能 (System PLL enable)
由软件置 1 和清零，用于使能 PLL。
当进入停止、待机和关断模式时由硬件清零。如果 PLL 时钟用作系统时钟，则此位不可清零。
0：PLL 关闭
1：PLL 开启

位 23:21 保留，必须保持复位值。

位 20 **HSEPRE**：HSE sysclk 和 PLL M 分频器预分频比 (HSE sysclk and PLL M divider prescaler)
由软件置 1 和清零，用于控制选择 HSE 时钟时 sysclk 和 PLL M 分频器输入的分频系数。
0：SYSCLK 和 PLL M 分频器输入时钟未分频 (HSE)
1：SYSCLK 和 PLL M 分频器输入时钟 2 分频 (HSE/2)

位 19 **CSSON**：HSE 时钟安全系统使能 (HSE Clock security system enable)

由软件置 1，用于使能时钟安全系统。当 CSSON 置 1 时，HSE 时钟监测器将在 HSE 振荡器就绪时由硬件使能，并在检出 HSE 时钟故障时由硬件禁止。该位仅置 1，并通过复位清零。

0：HSE 时钟安全系统关闭（时钟监测器关闭）

1：HSE 时钟安全系统开启（如果 HSE 振荡器稳定，则时钟监测器开启；如果不稳定，则关闭）

位 18 **HSEBYP**：HSE 晶振旁路 (HSE crystal oscillator bypass)

由软件置 1 和清零，用于用外部时钟旁路振荡器。外部时钟必须通过将 HSEON 位置 1 的方式使能才能为器件使用。HSEBYP 位只有在 HSE 振荡器已禁止的情况下（HSEON = 0 且 HSERDY = 0）才可写入。

0：不旁路 HSE 晶振

1：外部时钟旁路 HSE 晶振

位 17 **HSERDY**：HSE 时钟就绪标志 (HSE clock ready flag)

由硬件置 1，用于指示 HSE 振荡器已稳定。

0：HSE 振荡器未就绪

1：HSE 振荡器已就绪

注：　在将 HSEON 位清零后，HSERDY 将在 6 个 HSE 时钟周期后转为低电平。

位 16 **HSEON**：HSE 时钟使能 (HSE clock enable)

由软件置 1 和清零。

由硬件清零，用于在进入停止、待机和关断模式时停止 HSE 振荡器。如果 HSE 振荡器直接或间接用作系统时钟，则该位不可复位。

0：HSE 振荡器关闭

1：HSE 振荡器开启

位 15:13 保留，必须保持复位值。

位 12 **HSIKERDY**：针对外设请求的 HSI16 内核时钟就绪标志 (HSI16 kernel clock ready flag for peripherals requests)

由硬件置 1，用于指示 HSI16 振荡器在通过 HSIKERON 或外设内核时钟请求使能时已稳定。当 HSI16 通过软件将 HSION 置 1 的方式使能，或者通过从待机模式唤醒的方式使能时，不会置 1。

0：HSI16 振荡器未就绪

1：HSI16 振荡器已就绪

位 11 **HSIASFS**：HSI16 从停止模式自动启动 (HSI16 automatic start from Stop)

由软件置 1 和清零。当系统唤醒时钟为 MSI 时，该位用于在系统唤醒的同时唤醒 HSI16。

0：在 MSI 作为唤醒时钟的情况下退出停止模式时，不通过硬件使能 HSI16 振荡器。

1：在 MSI 作为唤醒时钟的情况下退出停止模式时，通过硬件使能 HSI16 振荡器。

位 10 **HSIRDY**：HSI16 时钟就绪标志 (HSI16 clock ready flag)（从待机模式唤醒后，一旦 HSI16 就绪，该位将读为 'b1）

由硬件置 1，用于指示 HSI16 振荡器已稳定。仅当 HSI16 通过软件将 HSION 置 1 的方式使能，或者通过从待机模式唤醒的方式使能时，该位才会置 1。当 HSI16 通过 HSIKERON 或 IP 请求使能时，不会置 1。

0：HSI16 振荡器未就绪

1：HSI16 振荡器已就绪

注：　在将 HSION 位清零后，HSIRDY 将在 6 个 HSI16 时钟周期后转为低电平。

位 9 **HSIKERON**：始终为外设内核时钟使能 HSI16 (HSI16 always enable for peripheral kernel clocks)

由软件置 1 和清零，用于强制 HSI16 开启（即使在停止模式下）。通过 HSIKERON 使能的 HSI16 只能为将 HSI16 配置为内核时钟的 USART、LPUART 和 I^2C 外设提供时钟。通过使 HSI16 在停止模式下保持开启，可避免由于 HSI16 启动时间而导致的通信速度变慢。该位对 HSION 值没有任何影响。

0：对 HSI16 振荡器无影响。

1：即使在停止模式下也强制 HSI16 振荡器开启。

位 8 **HSION**：HSI16 时钟使能 (HSI16 clock enable)

由软件置 1 和清零。

由硬件清零，用于在进入停止、待机和关断模式时停止 HSI16 振荡器。

由硬件置 1，用于在退出停止模式时或者在 HSE 晶振发生故障时强制 HSI16 振荡器开启（此时，STOPWUCK=1 或 HSIASFS = 1）。

如果 HSI16 直接或间接用作系统时钟，则此位由硬件置 1。

0：HSI16 振荡器关闭

1：HSI16 振荡器开启

位 7:4 **MSIRANGE[3:0]**：MSI 时钟范围 (MSI clock ranges)

当 MSIRANGE 置 1 时，这些位由软件配置，用于选择 MSI 的频率范围。总共有 12 种频率范围可供选择：

0000：范围 0，100 kHz 左右

0001：范围 1，200 kHz 左右

0010：范围 2，400 kHz 左右

0011：范围 3，800 kHz 左右

0100：范围 4，1 MHz 左右

0101：范围 5，2 MHz 左右

0110：范围 6，4 MHz 左右（复位值）

0111：范围 7，8 MHz 左右

1000：范围 8，16 MHz 左右

1001：范围 9，24 MHz 左右

1010：范围 10，32 MHz 左右

1011：范围 11，48 MHz 左右

其他值：不允许使用（硬件写保护）

注： 警告：当 MSI 关闭 (MSION=0) 或 MSI 就绪 (MSIRDY=1) 时，可以修改 MSIRANGE。当 MSI 开启且未就绪（MSION = 1 且 MSIRDY = 0）时，不得修改 MSIRANGE。

位 3 保留，必须保持复位值。

位 2 **MSIPLLEN**：MSI 时钟 PLL 使能 (MSI clock PLL enable)

由软件置 1 和清零，用于使能 / 禁止 MSI 时钟源的 PLL 部分。

在 LSE 使能（已使能 LSEON）和就绪（由硬件将 LSERDY 置 1）后，MSIPLLEN 必须使能。为了避免在 LSE 未就绪的情况下使能 MSIPLLEN，提供了相应的硬件保护。

当 LSE 禁止 (LSEON = 0) 或 LSE 上的时钟安全系统检测到 LSE 故障时（请参见 RCC_CSR 寄存器），该位由硬件清零。

0：MSI PLL 关闭

1：MSI PLL 开启

位 1 **MSIRDY**：MSI 时钟就绪标志 (MSI clock ready flag)（复位后，一旦 MSI 就绪，该位将读为 'b1）

该位由硬件置 1，用于指示 MSI 振荡器已稳定。

0：MSI 振荡器未就绪

1：MSI 振荡器已就绪

注： 在将 MSION 位清零后，MSIRDY 将在 6 个 MSI 时钟周期后转为低电平。

位 0 **MSION**：MSI 时钟使能 (MSI clock enable)

此位由软件置 1 和清零。

由硬件清零，用于在进入停止、待机和关断模式时停止 MSI 振荡器。

由硬件置 1，用于在退出待机和关断模式时强制 MSI 振荡器开启。

由硬件置 1，用于在退出停止模式时或者在 HSE 振荡器发生故障时强制 MSI 振荡器开启（此时，STOPWUCK=0）。

直接或间接用作系统时钟时，由硬件置 1。

0：MSI 振荡器关闭

1：MSI 振荡器开启

8.4.2 RCC 内部时钟源校准寄存器 (RCC_ICSCR)

RCC internal clock sources calibration register

偏移地址：0x004

复位值：0x40XX 00XX，其中 X 是出厂前编程的。

访问：无等待状态，按字、半字和字节访问

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

位 31 保留，必须保持复位值。

位 30:24 **HSITRIM[6:0]**：HSI16 时钟微调 (HSI16 clock trimming)

通过这些位，可在 HSICAL[7:0] 位基础上实现可由用户编程的微调值。可通过编程使其适应电压和温度的差异，使 HSI16 的频率更为准确。

默认值为 64，加上 HSICAL 值，应能将 HSI16 微调至 16 MHz ± 1 %。

位 23:16 **HSICAL[7:0]**：HSI16 时钟校准 (HSI16 clock calibration)

这些位在启动时初始化为出厂前编程的 HSI16 校准微调值。写入 HSITRIM 后，HSICAL 将更新为 HSITRIM 和出厂微调值之和。

位 15:8 **MSITRIM[7:0]**：MSI 时钟微调 (MSI clock trimming)

通过这些位，可在 MSICAL[7:0] 位基础上实现可由用户编程的微调值。可通过编程使其适应电压和温度的差异，使 MSI 的频率更为准确。

默认值为 0，加上 MSICAL 值，应能将 MSI 微调至其中频。

位 7:0 **MSICAL[7:0]**：MSI 时钟校准 (MSI clock calibration)

这些位在启动时初始化为出厂前编程的 MSI 校准微调值。写入 MSITRIM 后，MSICAL 将更新为 MSITRIM 和出厂微调值之和。

8.4.3 RCC 时钟配置寄存器 (RCC_CFGR)

RCC clock configuration register

偏移地址：0x008

复位值：0x0007 0000（在 POR 复位后），0x0007 0001（从待机模式唤醒复位后）。

访问：0 ≤ 等待状态 ≤ 2，按字、半字和字节访问

只有在时钟源切换期间进行访问时才会插入 1 或 2 个等待状态。

如果在更新 APB 或 AHB 预分频值时进行访问，则插入 0 到 15 个等待状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	MCOPRE[2:0]			MCOSEL[3:0]				Res.	Res.	Res.	Res.	Res.	PPRE2F	PPRE1F	HPREF
	rw	rw	rw	rw	rw	rw	rw						r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP WUCK	Res.	PPRE2[2:0]			PPRE1[2:0]			HPRE[3:0]				SWS[1:0]		SW[1:0]	
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	rw	rw

位 31 保留，必须保持复位值。

位 30:28 **MCOPRE[2:0]**：微控制器时钟输出预分频器 (Microcontroller clock output prescaler)

这些位将由软件置 1 和清零。

强烈建议在 MCO 输出使能之前更改此预分频器。

000：MCO 1 分频

001：MCO 2 分频

010：MCO 4 分频

011：MCO 8 分频

100：MCO 16 分频

其他值：不允许使用

位 27:24 **MCOSEL[3:0]**：微控制器时钟输出 (Microcontroller clock output)

由软件置 1 和清零。

0000：禁止 MCO 输出，MCO 上无时钟

0001：选择 SYSCLK 系统时钟

0010：选择 MSI 时钟

0011：选择 HSI16 时钟

0100：选择 HSE 时钟（稳定后）

0101：选择主 PLLRCLK 时钟

0110：选择 LSI1 时钟

0111：选择 LSI2 时钟

1000：选择 LSE 时钟

1001：选择内部 HSI48 时钟

1100：选择 HSE 时钟（稳定前）

其他值：保留

注： 该时钟输出在启动时或 MCO 时钟源切换期间可能包含一些被截断的周期。

位 23:19 保留，必须保持复位值。

位 18 **PPRE2F**：PCLK2 预分频器标志 (APB2) (PCLK2 prescaler flag (APB2))

由硬件置 1 和复位，用于确认 PCLK2 预分频器编程。

在 PPRE2 中编程新的预分频值时复位。在实际应用编程值时置 1。

0：尚未应用 PCLK2 预分频值。

1：已应用 PCLK2 预分频值。

位 17 **PPRE1F**：PCLK1 预分频器标志 (APB1) (PCLK1 prescaler flag (APB1))

由硬件置 1 和复位，用于确认 PCLK1 预分频器编程。

在 PPRE1 中编程新的预分频值时复位。在实际应用编程值时置 1。

0：尚未应用 PCLK1 预分频值。

1：已应用 PCLK1 预分频值。

位 16 **HPREF**：HCLK1 预分频器标志（CPU1、AHB1、AHB2、AHB3 和 SRAM1）(HCLK1 prescaler flag (CPU1, AHB1, AHB2, AHB3 and SRAM1))

由硬件置 1 和复位，用于确认 HCLK1 预分频器编程。

在 HPRE 中编程新的预分频值时复位。在实际应用编程值时置 1。

0：尚未应用 HCLK1 预分频值。

1：已应用 HCLK1 预分频值。

位 15 **STOPWUCK**：从停止模式唤醒时的时钟以及 CSS 备份时钟选择 (Wakeup from Stop and CSS backup clock selection)

由软件置 1 和清零，用于选择退出停止模式时使用的系统时钟。

所选时钟还用作 HSE 上的时钟安全系统的紧急时钟。警告：HSE 时钟安全系统已通过 *RCC 时钟控制寄存器 (RCC_CR)* 中的 CSSON 使能且系统时钟为 HSE（SWS=“10”）或者请求开启 HSE（SW=“10”）时，不能修改 STOPWUCK。

0：选择 MSI 振荡器作为从停止模式唤醒时的时钟以及 CSS 备份时钟。

1：选择 HSI16 振荡器作为从停止模式唤醒时的时钟以及 CSS 备份时钟。

位 14 保留，必须保持复位值

位 13:11 **PPRE2[2:0]**：PCLK2 高速预分频器 (APB2) (PCLK2 high-speed prescaler (APB2))

由软件置 1 和清零，用于控制 PCLK2 时钟 (APB2) 的分频系数。

可以检查 PPRE2F 标志以确认是否应用了编程的 PPRE2 预分频值。

0xx：HCLK1 不分频

100：HCLK1 2 分频

101：HCLK1 4 分频

110：HCLK1 8 分频

111：HCLK1 16 分频

位 10:8 **PPRE1[2:0]**：PCLK1 低速预分频器 (APB1) (PCLK1 low-speed prescaler (APB1))

由软件置 1 和清零，用于控制 PCLK1 时钟 (APB1) 的分频系数。

可以检查 PPRE1F 标志以确认是否应用了编程的 PPRE1 预分频值。

0xx：HCLK1 不分频

100：HCLK1 2 分频

101：HCLK1 4 分频

110：HCLK1 8 分频

111：HCLK1 16 分频

位 7:4 **HPRE[3:0]**：HCLK1 预分频器（CPU1、AHB1、AHB2、AHB3 和 SRAM1）(HCLK1 prescaler (CPU1, AHB1, AHB2, AHB3, and SRAM1))

由软件置 1 和清零，用于控制 HCLK1 时钟（CPU1、AHB1、AHB2、AHB3 和 SRAM1）的分频系数。

可以检查 HPREF 标志以确认是否应用了编程的 HPRE 预分频值。

注意： 软件必须根据器件电压范围正确设置这些位，以确保系统频率不会超过允许的最大频率（更多详细信息，请参见*第 6.1.6 节：动态电压调节管理*）。在对这些位执行写操作之后、减小电压范围之前，必须读取寄存器位 HPREF 以确保其中的值为新值。

0001：SYSCLK 3 分频
0010：SYSCLK 5 分频
0101：SYSCLK 6 分频
0110：SYSCLK 10 分频
0111：SYSCLK 32 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
其他：SYSCLK 不分频

位 3:2 **SWS[1:0]**：系统时钟切换状态 (System clock switch status)

由硬件置 1 和清零，用于指示用作系统时钟的时钟源。
00：MSI 振荡器用作系统时钟
01：HSI16 振荡器用作系统时钟
10：HSE 用作系统时钟
11：PLL 用作系统时钟

位 1:0 **SW[1:0]**：系统时钟切换 (System clock switch)

由软件置 1 和清零，用于选择系统时钟源 (SYSCLK)。
由硬件配置，用于在退出关断模式时强制选择 MSI 振荡器。由硬件配置，用于在退出待机模式时强制选择 HSI16 振荡器。
由硬件配置，用于在退出停止模式时或者在 HSE 振荡器发生故障时强制选择 MSI 或 HSI16 振荡器（取决于 STOPWUCK 值）。
00：选择 MSI 作为系统时钟
01：选择 HSI16 作为系统时钟
10：选择 HSE 作为系统时钟
11：选择 PLL 作为系统时钟

8.4.4 RCC PLL 配置寄存器 (RCC_PLLCFGR)

RCC PLL configuration register

偏移地址：0x00C

复位值：0x2204 0100

访问：无等待状态，按字、半字和字节访问

此寄存器用于根据以下公式配置 PLL 时钟输出：

- f(VCO 时钟) = f(PLL 时钟输入) × (PLLN / PLLM)
- f(PLL_P) = f(VCO 时钟) / PLLP
- f(PLL_Q) = f(VCO 时钟) / PLLQ
- f(PLL_R) = f(VCO 时钟) / PLLR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLR[2:0]			PLLREN	PLLQ[2:0]			PLLQEN	Res.	Res.	PLLP[4:0]					PLLPEN
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PLLN[7:0]							Res.	PLLM[2:0]			Res.	Res.	PLLSRC[1:0]	
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw			rw	rw

位 31:29 **PLLR[2:0]**：适用于 PLLRCLK 的主 PLL 分频系数 (Main PLL division factor for PLLRCLK)

由软件置 1 和清零，用于控制主 PLL 输出时钟 PLLRCLK 的频率。可以选择该输出作为系统时钟。这些位只能在 PLL 已禁止时写入。

PLLRCLK 输出时钟频率 = VCO 频率 / PLLR，其中，PLLR = 2、3、4... 或 8

[VCO 频率 / (N + 1)]

000：保留

001：PLLR = 2

010：PLLR = 3

011：PLLR = 4

100：PLLR = 5

101：PLLR = 6

110：PLLR = 7

111：PLLR = 8

软件必须正确设置这些位，使其在此域中不超过 64 MHz。

位 28 **PLLREN**：主 PLL PLLRCLK 输出使能 (Main PLL PLLRCLK output enable)

由软件置 1 和复位，用于使能主 PLL 的 PLLRCLK 输出（用作系统时钟）。

当 PLL 的 PLLRCLK 输出用作系统时钟时，该位不可写入。

为实现节能，不使用 PLL 的 PLLRCLK 输出时，PLLREN 的值应为 0。

0：禁止 PLLRCLK 输出

1：使能 PLLRCLK 输出

位 27:25 **PLLQ[2:0]**：适用于 PLLQCLK 的主 PLL 分频系数 (Main PLL division factor for PLLQCLK)

由软件置 1 和清零，用于控制主 PLL 输出时钟 PLLQCLK 的频率。可为 USB 和真随机数时钟选择此输出。这些位只能在 PLL 已禁止时写入。

PLLQCLK 输出时钟频率 = VCO 频率 / PLLQ，其中，PLLQ = 2、3、4... 或 8 [VCO 频率 / (N + 1)]

000：保留

001：PLLQ = 2

010：PLLQ = 3

011：PLLQ = 4

100：PLLQ = 5

101：PLLQ = 6

110：PLLQ = 7

111：PLLQ = 8

软件必须正确设置这些位，使其在此域中不超过 64 MHz。

位 24 **PLLQEN**：主 PLL PLLQCLK 输出使能 (Main PLL PLLQCLK output enable)

由软件置 1 和复位，用于使能主 PLL 的 PLLQCLK 输出（用作系统时钟）。

为实现节能，不使用 PLL 的 PLLQCLK 输出时，PLLQEN 的值应为 0。

0：禁止 PLLQCLK 输出

1：使能 PLLQCLK 输出

位 23:22 保留，必须保持复位值。

位 21:17 **PLLP[4:0]**：适用于 PLLPCLK 的主 PLL 分频系数 (Main PLL division factor for PLLPCLK)

由软件置 1 和清零，用于控制主 PLL 输出时钟 PLLPCLK 的频率。可为 SAI1 和 ADC 选择此输出。这些位只能在 PLL 已禁止时写入。

PLLPCLK 输出时钟频率 = VCO 频率 / PLLP，其中，PLLP = 2、3、4... 或 32 [VCO 频率 / (N + 1)]

0000：保留

00001：PLLP = 2

00010：PLLP = 3

00011：PLLP = 4

00100：PLLP = 5

...

11111：PLLP = 32

注意： 软件必须正确设置这些位，使其在此域中不超过 64 MHz。

位 16 **PLLPEN**：主 PLL PLLPCLK 输出使能 (Main PLL PLLPCLK output enable)

由软件置 1 和复位，用于使能主 PLL 的 PLLPCLK 输出。

为实现节能，不使用 PLL 的 PLLPCLK 输出时，PLLPEN 的值应为 0。

0：禁止 PLLPCLK 输出

1：使能 PLLPCLK 输出

位 15 保留，必须保持复位值。

位 14:8 **PLLN[6:0]**：适用于 VCO 的主 PLL 倍频系数 (Main PLL multiplication factor for VCO)

由软件置 1 和清零，用于控制 VCO 的倍频系数。这些位只能在 PLL 已禁止时写入。

VCO 输出频率 = VCO 输入频率 x PLLN，其中，8 =< PLLN =< 86

0000000：PLLN = 0，错误配置

0000001：PLLN = 1，错误配置

...

0000111：PLLN = 7，错误配置

0001000：PLLN = 8

0001001：PLLN = 9

...

1010101：PLLN = 85

1010110：PLLN = 86

1010111：PLLN = 87，错误配置

...

1111111：PLLN = 127，错误配置

注意： 软件必须正确设置这些位，确保 VCO 输出频率介于 64 MHz 和 344 MHz 之间。

位 7 保留，必须保持复位值。

位 6:4 **PLLM**：主 PLL 和音频 PLLSAI1 输入时钟的分频系数 (Division factor for the main PLL and audio PLLSAI1 input clock)

由软件置 1 和清零，用于在 VCO 之前对 PLL 和 PLLSAI1 输入时钟进行分频。这些位只有在所有 PLL 已禁止时才可写入。

VCO 输入频率 = PLL 输入时钟频率 / PLLM，其中 1 <= PLLM <= 8

000：PLLM = 1

001：PLLM = 2

010：PLLM = 3

011：PLLM = 4

100：PLLM = 5

101：PLLM = 6

110：PLLM = 7

111：PLLM = 8

注意： 软件必须正确设置这些位，确保 VCO 输入频率介于 4 MHz 和 16 MHz 之间。

位 3:2 保留，必须保持复位值。

位 1:0 **PLLSRC**：主 PLL 和音频 PLLSAI1 输入时钟源 (Main PLL and, audio PLLSAI1 entry clock source)

由软件置 1 和清零，用于选择 PLL 和 PLLSAI1 时钟源。这些位只有在 PLL 和 PLLSAI1 已禁止时才可写入。

为实现节能，不使用 PLL 时，PLLSRC 的值应为 00。

00：未向 PLL 和 PLLSAI1 发送任何时钟

01：选择 MSI 时钟作为 PLL 和 PLLSAI1 时钟输入

10：选择 HSI16 时钟作为 PLL 和 PLLSAI1 时钟输入

11：选择 HSE 时钟作为 PLL 和 PLLSAI1 时钟输入

8.4.5 RCC PLLSAI1 配置寄存器 (RCC_PLLSAI1CFGR)

RCC PLLSAI1 configuration register

偏移地址：0x010

复位值：0x2204 0100

访问：无等待状态，按字、半字和字节访问

此寄存器用于根据公式配置 PLLSAI1 时钟输出：

- f(VCOSAI 时钟) = f(PLLSAI1 时钟输入) × (PLLN / PLLM)
- f(PLLSAI1_P) = f(VCOSAI 时钟) / PLLP
- f(PLLSAI1_Q) = f(VCOSAI 时钟) / PLLQ
- f(PLLSAI1_R) = f(VCOSAI 时钟) / PLLR

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

位 31:29 **PLLR[2:0]**：适用于 PLLSAI1RCLK 的音频 PLLSAI1 分频系数 (Audio PLLSAI1 division factor for PLLSAI1RCLK)

由软件置 1 和清零，用于控制音频 PLLSAI1 输出时钟 PLLSAI1RCLK 的频率。可以选择该输出作为系统时钟。这些位只能在 PLLSAI1 已禁止时写入。

PLLSAI1RCLK 输出时钟频率 = VCO 频率 / PLLR，其中，PLLR = 2、3、4... 或 8 [VCO 频率 / (N + 1)]

000：保留

001：PLLR = 2

010：PLLR = 3

011：PLLR = 4

100：PLLR = 5

101：PLLR = 6

110：PLLR = 7

111：PLLR = 8

软件必须正确设置这些位，使其在此域中不超过 64 MHz。

位 28 **PLLREN**：音频 PLLSAI1 PLLSAI1RCLK 输出使能 (Audio PLLSAI1 PLLSAI1RCLK output enable)

由软件置 1 和复位，用于使能音频 PLLSAI1 的 PLLSAI1RCLK 输出（用作系统时钟）。

为实现节能，不使用 PLLSAI1 的 PLLSAI1RCLK 输出时，PLLREN 的值应为 0。

0：禁止 PLLSAI1RCLK 输出

1：使能 PLLSAI1RCLK 输出

位 27:25 **PLLQ[2:0]**：适用于 PLLSAI1QCLK 的音频 PLLSAI1 分频系数 (Audio PLLSAI1 division factor for PLLSAI1QCLK)

由软件置 1 和清零，用于控制音频 PLLSAI1 输出时钟 PLLSAI1QCLK 的频率。可为 USB 或真随机数时钟选择此输出。这些位只能在 PLLSAI1 已禁止时写入。

PLLSAI1QCLK 输出时钟频率 = VCO 频率 / PLLQ，其中，PLLQ = 2、3、4... 或 8 [VCO 频率 / (N + 1)]

000：保留

001：PLLQ = 2

010：PLLQ = 3

011：PLLQ = 4

100：PLLQ = 5

101：PLLQ = 6

110：PLLQ = 7

111：PLLQ = 8

软件必须正确设置这些位，使其在此域中不超过 64 MHz。

位 24 **PLLQEN**：音频 PLLSAI1 PLLSAI1QCLK 输出使能 (Audio PLLSAI1 PLLSAI1QCLK output enable)

由软件置 1 和复位，用于使能音频 PLLSAI1 的 PLLSAI1QCLK 输出（用作系统时钟）。

为实现节能，不使用 PLLSAI1 的 PLLSAI1QCLK 输出时，PLLQEN 的值应为 0。

0：禁止 PLLSAI1QCLK 输出

1：使能 PLLSAI1QCLK 输出

位 23:22 保留，必须保持复位值。

位 21:17 **PLLP[4:0]**：适用于 PLLSAI1PCLK 的音频 PLLSAI1 分频系数 (Audio PLLSAI1 division factor for PLLSAI1PCLK)

由软件置 1 和清零，用于控制音频 PLLSAI1 输出时钟 PLLSAI1PCLK 的频率。可为 SAI1 和 ADC 选择此输出。这些位只能在 PLLSAI1 已禁止时写入。

PLLSAI1PCLK 输出时钟频率 = VCO 频率 / PLLP，其中，PLLP = 2、3、4... 或 32 [VCO 频率 / (N + 1)]

00000：保留

00001：PLLP = 2

00010：PLLP = 3

00011：PLLP = 4

00100：PLLP = 5

...

11111：PLLP = 32

注意： 软件必须正确设置这些位，使其在此域中不超过 64 MHz。

位 16 **PLLPEN**：音频 PLLSAI1 PLLSAI1PCLK 输出使能 (Audio PLLSAI1 PLLSAI1PCLK output enable)

由软件置 1 和复位，用于使能音频 PLLSAI1 的 PLLSAI1PCLK 输出。

为实现节能，不使用 PLL 的 PLLSAI1PCLK 输出时，PLLPEN 的值应为 0。

0：禁止 PLLSAI1PCLK 输出

1：使能 PLLSAI1PCLK 输出

位 15 保留，必须保持复位值。

位 14:8 **PLLN[6:0]**：适用于 VCO 的音频 PLLSAI1 倍频系数 (Audio PLLSAI1 multiplication factor for VCO)
由软件置 1 和清零，用于控制 VCO 的倍频系数。这些位只能在 PLLSAI1 已禁止时写入。
VCO 输出频率 = VCO 输入频率 x PLLN，其中，4=< PLLN =< 86
0000000：PLLN = 0，错误配置
0000001：PLLN = 1，错误配置
...
0000011：PLLN = 3，错误配置
0000100：PLLN = 4
0000101：PLLN = 5
...
1010101：PLLN = 85
1010110：PLLN = 86
1010111：PLLN = 87，错误配置
...
1111111：PLLN = 127，错误配置

注意： 软件必须正确设置这些位，确保 VCO 输出频率介于 64 MHz 和 344 MHz 之间。

位 7:0 保留，必须保持复位值。

8.4.6 RCC 时钟中断使能寄存器 (RCC_CIER)

RCC clock interrupt enable register

偏移地址：0x018

复位值：0x0000 0000

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LSI2 RDYIE	HSI48 RDYIE	LSE CSSIE	Res.	Res.	PLLSAI1 RDYIE	PLL RDYIE	HSE RDYIE	HSI RDYIE	MSI RDYIE	LSE RDYIE	LSI1 RDYIE
				rw	rw	rw			rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11 **LSI2RDYIE**：LSI2 就绪中断使能 (LSI2 ready interrupt enable)
由软件置 1 和清零，用于使能 / 禁止由 LSI2 振荡器稳定所引起的中断。
0：禁止 LSI2 就绪中断
1：使能 LSI2 就绪中断

位 10 **HSI48RDYIE**：HSI48 就绪中断使能 (HSI48 ready interrupt enable)
由软件置 1 和清零，用于使能 / 禁止由内部 HSI48 振荡器所引起的中断。
0：禁止 HSI48 就绪中断
1：使能 HSI48 就绪中断

位 9 **LSECSSIE**：LSE 时钟安全系统中断使能 (LSE clock security system interrupt enable)
由软件置 1 和清零，用于使能 / 禁止由 LSE 上的时钟安全系统引起的中断。
0：禁止因 LSE 时钟故障而引起的时钟安全中断
1：使能因 LSE 时钟故障而引起的时钟安全中断

位 8:7 保留，必须保持复位值。

位 6 **PLLSAI1RDYIE**：PLLSAI1 就绪中断使能 (PLLSAI1 ready interrupt enable)
由软件置 1 和清零，用于使能 / 禁止由 PLLSAI1 锁定引起的中断。
0：禁止 PLLSAI1 锁定中断
1：使能 PLLSAI1 锁定中断

位 5 **PLLRDYIE**：PLL 就绪中断使能 (PLL ready interrupt enable)
由软件置 1 和清零，用于使能 / 禁止由 PLL 锁定引起的中断。
0：禁止 PLL 锁定中断
1：使能 PLL 锁定中断

位 4 **HSERDYIE**：HSE 就绪中断使能 (HSE ready interrupt enable)
由软件置 1 和清零，用于使能 / 禁止由 HSE 振荡器稳定所引起的中断。
0：禁止 HSE 就绪中断
1：使能 HSE 就绪中断

位 3 **HSIRDYIE：HSI16 就绪中断使能 (HSI16 ready interrupt enable)**
由软件置 1 和清零，用于使能 / 禁止由 HSI16 振荡器稳定所引起的中断。
0：禁止 HSI16 就绪中断
1：使能 HSI16 就绪中断

位 2 **MSIRDYIE**：MSI 就绪中断使能 (MSI ready interrupt enable)
由软件置 1 和清零，用于使能 / 禁止由 MSI 振荡器稳定所引起的中断。
0：禁止 MSI 就绪中断
1：使能 MSI 就绪中断

位 1 **LSERDYIE**：LSE 就绪中断使能 (LSE ready interrupt enable)
由软件置 1 和清零，用于使能 / 禁止由 LSE 振荡器稳定所引起的中断。
0：禁止 LSE 就绪中断
1：使能 LSE 就绪中断

位 0 **LSI1RDYIE**：LSI1 就绪中断使能 (LSI1 ready interrupt enable)
由软件置 1 和清零，用于使能 / 禁止由 LSI1 振荡器稳定所引起的中断。
0：禁止 LSI1 就绪中断
1：使能 LSI1 就绪中断

8.4.7 RCC 时钟中断标志寄存器 (RCC_CIFR)

RCC clock interrupt flag register

偏移地址：0x01C

复位值：0x0000 0000

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LSI2 RDYF	HSI48R DYF	LSE CSSF	CSSF	Res.	PLLSAI1 RDYF	PLL RDYF	HSE RDYF	HSI RDYF	MSI RDYF	LSE RDYF	LSI1 RDYF
				r	r	r	r		r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值。

位 11 **LSI2RDYF**：LSI2 就绪中断标志 (LSI2 ready interrupt flag)
当 LSI2 时钟稳定且 LSI2RDYDIE 置 1 时由硬件置 1。
LSI2RDYC 位置 1 时由软件清零。
0：当前未因 LSI2 振荡器引起时钟就绪中断
1：因 LSI2 振荡器引起时钟就绪中断

位 10 **HSI48RDYF**：HSI48 就绪中断标志 (HSI48 ready interrupt flag)
当 HSI48 时钟稳定且 HSI48RDYIE 置 1 时由硬件置 1，以对 HSI48ON 置 1 作出响应（请参见 *RCC 时钟恢复 RC 寄存器 (RCC_CRRCR)*）。
HSI48RDYC 位置 1 时由软件清零。
0：当前未因 HSI48 振荡器引起时钟就绪中断
1：因 HSI48 振荡器引起时钟就绪中断

位 9 **LSECSSF**：LSE 时钟安全系统中断标志 (LSE Clock security system interrupt flag)
当在 LSE 振荡器中检出故障时由硬件置 1。
LSECSSC 位置 1 时由软件清零。
0：当前未因 LSE 时钟故障而引起时钟安全中断
1：因 LSE 时钟故障而引起时钟安全中断

位 8 **CSSF**：HSE 时钟安全系统中断标志 (HSE Clock security system interrupt flag)
当在 HSE 振荡器中检出故障时由硬件置 1。
CSSC 位置 1 时由软件清零。
0：当前未因 HSE 时钟故障而引起时钟安全中断
1：因 HSE 时钟故障而引起时钟安全中断

位 7 保留，必须保持复位值。

位 6 **PLLSAI1RDYF**：PLLSAI1 就绪中断标志 (PLLSAI1 ready interrupt flag)
当 PLLSAI1 锁定并且 PLLSAI1RDYDIE 置 1 时由硬件置 1。
PLLSAI1RDYC 位置 1 时由软件清零。
0：当前未因 PLLSAI1 锁定而引起时钟就绪中断
1：因 PLLSAI1 锁定而引起时钟就绪中断

位 5 **PLLRDYF**：PLL 就绪中断标志 (PLL ready interrupt flag)
当 PLL 锁定并且 PLLRDYDIE 置 1 时由硬件置 1。
PLLRDYC 位置 1 时由软件清零。
0：当前未因 PLL 锁定而引起时钟就绪中断
1：因 PLL 锁定而引起时钟就绪中断

位 4 **HSERDYF**：HSE 就绪中断标志 (HSE ready interrupt flag)
当 HSE 时钟稳定且 HSERDYDIE 置 1 时由硬件置 1。
HSERDYC 位置 1 时由软件清零。
0：当前未因 HSE 振荡器引起时钟就绪中断
1：因 HSE 振荡器引起时钟就绪中断

位 3 **HSIRDYF**：HSI16 就绪中断标志 (HSI16 ready interrupt flag)
当 HSI16 时钟稳定且 HSIRDYDIE 置 1 时由硬件置 1，以对 HSION 置 1 作出响应（请参见 *RCC 时钟控制寄存器 (RCC_CR)*）。当 HSION 未置 1，但外设通过时钟请求使能 HSI16 振荡器时，该位不会置 1，也不会生成中断。
HSIRDYC 位置 1 时由软件清零。
0：当前未因 HSI16 振荡器引起时钟就绪中断
1：因 HSI16 振荡器引起时钟就绪中断

位 2 **MSIRDYF**：MSI 就绪中断标志 (MSI ready interrupt flag)
当 MSI 时钟稳定且 MSIRDYDIE 置 1 时由硬件置 1。
MSIRDYC 位置 1 时由软件清零。
0：当前未因 MSI 振荡器引起时钟就绪中断
1：因 MSI 振荡器引起时钟就绪中断

位 1 **LSERDYF**：LSE 就绪中断标志 (LSE ready interrupt flag)
当 LSE 时钟稳定且 LSERDYDIE 置 1 时由硬件置 1。
LSERDYC 位置 1 时由软件清零。
0：当前未因 LSE 振荡器引起时钟就绪中断
1：因 LSE 振荡器引起时钟就绪中断

位 0 **LSI1RDYF**：LSI1 就绪中断标志 (LSI1 ready interrupt flag)
当 LSI1 时钟稳定且 LSI1RDYDIE 置 1 时由硬件置 1。
LSI1RDYC 位置 1 时由软件清零。
0：当前未因 LSI1 振荡器引起时钟就绪中断
1：因 LSI1 振荡器引起时钟就绪中断

8.4.8 RCC 时钟中断清零寄存器 (RCC_CICR)

RCC clock interrupt clear register

偏移地址：0x020

复位值：0x0000 0000

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LSI2 RDYC	HSI48 RDYC	LSE CSSC	CSSC	Res.	PLLSAI1 RDYC	PLL RDYC	HSE RDYC	HSI RDYC	MSI RDYC	LSE RDYC	LSI1 RDYC
				w	w	w	w		w	w	w	w	w	w	w

位 31:12 保留，必须保持复位值。

位 11 **LSI2RDYC**：LSI2 就绪中断清零 (LSI2 ready interrupt clear)
该位由软件置 1，用于将 LSI2RDYF 标志清零。
0：无影响
1：将 LSI2RDYF 标志清零

位 10 **HSI48RDYC**：HSI48 振荡器就绪中断清零 (HSI48 oscillator ready interrupt clear)
该位由软件置 1，用于将 HSI48RDYF 标志清零。
0：无影响
1：将 HSI48RDYC 标志清零

位 9 **LSECSSC**：LSE 时钟安全系统中断清零 (LSE Clock security system interrupt clear)
此位由软件置 1，用于将 LSECSSF 标志清零。
0：无影响
1：将 LSECSSF 标志清零

位 8 **CSSC**：HSE 时钟安全系统中断清零 (HSE Clock security system interrupt clear)
此位由软件置 1，用于将 HSE CSSF 标志清零。
0：无影响
1：将 HSE CSSF 标志清零

位 7 保留，必须保持复位值。

位 6 **PLLSAI1RDYC**：PLLSAI1 就绪中断清零 (PLLSAI1 ready interrupt clear)
此位由软件置 1，用于将 PLLSAI1RDYF 标志清零。
0：无影响
1：将 PLLSAI1RDYF 标志清零

位 5 **PLLRDYC**：PLL 就绪中断清零 (PLL ready interrupt clear)
该位由软件置 1，用于将 PLLRDYF 标志清零。
0：无影响
1：将 PLLRDYF 标志清零

位 4 **HSERDYC**：HSE 就绪中断清零 (HSE ready interrupt clear)
该位由软件置 1，用于将 HSERDYF 标志清零。
0：无影响
1：将 HSERDYF 标志清零

位 3 **HSIRDYC**：HSI16 就绪中断清零 (HSI16 ready interrupt clear)
此位由软件置 1，用于将 HSIRDYF 标志清零。
0：无影响
1：将 HSIRDYF 标志清零

位 2 **MSIRDYC**：MSI 就绪中断清零 (MSI ready interrupt clear)
该位由软件置 1，用于将 MSIRDYF 标志清零。
0：无影响
1：将 MSIRDYF 标志清零

位 1 **LSERDYC**：LSE 就绪中断清零 (LSE ready interrupt clear)
该位由软件置 1，用于将 LSERDYF 标志清零。
0：无影响
1：将 LSERDYF 标志清零

位 0 **LSI1RDYC**：LSI1 就绪中断清零 (LSI1 ready interrupt clear)
该位由软件置 1，用于将 LSI1RDYF 标志清零。
0：无影响
1：将 LSI1RDYF 标志清零

8.4.9 RCC SMPS 降压转换器控制寄存器 (RCC_SMPSCR)

RCC SMPS step-down converter control register

偏移地址：0x024

复位值：0x0000 0301（在 POR 复位后），0x0000 0300（从待机模式唤醒后）

访问：无等待状态，按字、半字和字节访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	SMPSSWS[1:0]		Res.	Res.	SMPSDIV[1:0]		Res.	Res.	SMPSSEL[1:0]	
						r	r			rw	rw			rw	rw

位 31:10 保留，必须保持复位值。

位 9:8 **SMPSSWS[1:0]**：SMPS 降压转换器时钟切换状态 (SMPS Step Down converter clock switch status)

由硬件置 1 和清零，用于指示当 SMPS 使能时哪个时钟源用作 SMPS 降压转换器时钟。只要 HSE 处于活动状态，就会使用 HSE，而不管 SMPSSEL 中的设置如何。只要在 PWR_CR5.SMPSEN 中禁止 SMPS 降压转换器，就不会使用时钟。

00：将 HSI16 振荡器用作 SMPS 降压转换器时钟

01：将 MSI 振荡器用作 SMPS 降压转换器时钟

10：将 HSE 用作 SMPS 降压转换器时钟

11：不使用时钟

位 7:6 保留，必须保持复位值。

位 5:4 **SMPSDIV[1:0]**：SMPS 降压转换器时钟预分频器 (SMPS Step Down converter clock prescaler)

由软件置 1 和清零，用于控制 SMPS 降压转换器时钟的分频系数。SMPS 降压转换器时钟预分频系数取决于 SMPSDIV[1:0] 和 SMPSSEL[1:0] 设置。请参见*表 38*。

位 3:2 保留，必须保持复位值。

位 1:0 **SMPSSEL[1:0]**：SMPS 降压转换器时钟选择 (SMPS Step Down converter clock selection)

由软件置 1 和清零，用于选择 SMPS 降压转换器时钟源 (SMPSCLK)。

00：选择 HSI16 作为 SMPS 降压转换器时钟

01：选择 MSI 作为 SMPS 降压转换器时钟（应将 MSITRANGE 设置为支持的值，请参见*表 38*）。

10：选择 HSE 作为 SMPS 降压转换器时钟

11：保留

注： *在 POR 复位、NRST 引脚复位或从关断模式唤醒时，由硬件强制 SMPSSEL 选择 MSI（值 b01）。从待机模式唤醒时，由硬件强制 SMPSSEL 选择 HSI（值 b00）。从停止模式唤醒时，由硬件强制 SMPSSEL 选择 RCC_CFGR.STOPWUCK 中定义的 MSI 或 HSI 时钟。*

8.4.10 RCC AHB1 外设复位寄存器 (RCC_AHB1RSTR)

RCC AHB1 peripheral reset register

偏移地址：0x028

复位值：0x00000 0000

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSC RST
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRC RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAMUX1 RST	DMA2RST	DMA1RST
			rw										rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **TSCRST**：触摸感应控制器复位 (Touch Sensing Controller reset)
由软件置 1 和清零。
0：无影响
1：复位 TSC

位 15:13 保留，必须保持复位值。

位 12 **CRCRST**：CRC 复位 (CRC reset)
由软件置 1 和清零。
0：无影响
1：复位 CRC

位 11:3 保留，必须保持复位值。

位 2 **DMAMUX1RST**：DMAMUX 复位 (DMAMUX reset)
由软件置 1 和清零。
0：无影响
1：复位 DMAMUX1

位 1 **DMA2RST**：DMA2 复位 (DMA2 reset)
由软件置 1 和清零。
0：无影响
1：复位 DMA2

位 0 **DMA1RST**：DMA1 复位 (DMA1 reset)
由软件置 1 和清零。
0：无影响
1：复位 DMA1

8.4.11 RCC AHB2 外设复位寄存器 (RCC_AHB2RSTR)

RCC AHB2 peripheral reset register

偏移地址：0x02C

复位值：0x00000 0000

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AES1 RST
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ADC RST	Res.	Res.	Res.	Res.	Res.	GPIOH RST	Res.	Res.	GPIOE RST	GPIOD RST	GPIOC RST	GPIOB RST	GPIOA RST
		rw						rw			rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **AES1RST**：AES1 硬件加速器复位 (AES1 hardware accelerator reset)
由软件置 1 和清零。
0：无影响
1：复位 AES1

位 15:14 保留，必须保持复位值。

位 13 **ADCRST**：ADC 复位 (ADC reset)
由软件置 1 和清零。
0：无影响
1：复位 ADC 接口

位 12:8 保留，必须保持复位值。

位 7 **GPIOHRST**：IO 端口 H 复位 (IO port H reset)
由软件置 1 和清零。
0：无影响
1：复位 IO 端口 H

位 6:5 保留，必须保持复位值。

位 4 **GPIOERST**：IO 端口 E 复位 (IO port E reset)
由软件置 1 和清零。
0：无影响
1：复位 IO 端口 E

位 3 **GPIODRST**：IO 端口 D 复位 (IO port D reset)
由软件置 1 和清零。
0：无影响
1：复位 IO 端口 D

位 2 **GPIOCRST**：IO 端口 C 复位 (IO port C reset)
由软件置 1 和清零。
0：无影响
1：复位 IO 端口 C

位 1 **GPIOBRST**：IO 端口 B 复位 (IO port B reset)
由软件置 1 和清零。
0：无影响
1：复位 IO 端口 B

位 0 **GPIOARST**：IO 端口 A 复位 (IO port A reset)
由软件置 1 和清零。
0：无影响
1：复位 IO 端口 A

8.4.12 RCC AHB3 和 AHB4 外设复位寄存器 (RCC_AHB3RSTR)

RCC AHB3 and AHB4 peripheral reset register

偏移地址：0x030

复位值：0x00000 0000

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	FLASH RST	Res.	Res.	Res.	Res.	IPCC RST	HSEM RST	RNG RST	AES2 RST	PKA RST
						rw					rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	QUADSPI RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								

位 31:26 保留，必须保持复位值。

位 25 **FLASHRST**：Flash 接口复位 (Flash interface reset)
只有在 Flash 掉电时，才能将此位置 1。由软件置 1 和清零。
0：无影响
1：复位 Flash 接口。

位 24:21 保留，必须保持复位值。

位 20 **IPCCRST**：IPCC 接口复位 (IPCC interface reset)
由软件置 1 和清零。
0：无影响
1：复位 IPCC

位 19 **HSEMRST**：HSEM 复位 (HSEM reset)
由软件置 1 和清零。
0：无影响
1：复位 HSEM

位 18 **RNGRST**：真随机数复位 (True RNG reset)
由软件置 1 和清零。
0：无影响
1：复位真随机数

位 17 **AES2RST**：AES2 硬件加速器复位 (AES2 hardware accelerator reset)
由软件置 1 和清零。
0：无影响
1：复位 AES2

位 16 **PKARST**：PKA 硬件加速器复位 (PKA hardware accelerator reset)
由软件置 1 和清零。
0：无影响
1：复位 PKA

位 15:9 保留，必须保持复位值。

位 8 **QUADSPIRST**：四线 SPI 存储器接口复位 (Quad SPI memory interface reset)
由软件置 1 和清零。
0：无影响
1：复位 QUADSPI

位 7:0 保留，必须保持复位值。

8.4.13 RCC APB1 外设复位寄存器 1 (RCC_APB1RSTR1)

RCC APB1 peripheral reset register 1

偏移地址：0x038

复位值：0x0000 0000

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM1 RST	Res.	Res.	Res.	Res.	USB RST	Res.	CRS RST	I2C3 RST	Res.	I2C1 RST	Res.	Res.	Res.	Res.	Res.
rw					rw		rw	rw		rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SPI2 RST	Res.	Res.	Res.	Res.	LCD RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM2 RST
	rw					rw									rw

位 31 **LPTIM1RST**：低功耗定时器 1 复位 (Low Power Timer 1 reset)
由软件置 1 和清零。
0：无影响
1：复位 LPTIM1

位 30:27 保留，必须保持复位值。

位 26 **USBRST**：USB FS 复位 (USB FS reset)
由软件置 1 和清零。
0：无影响
1：复位 USB FS

位 25 保留，必须保持复位值

位 24 **CRSRST**：CRS 复位 (CRS reset)
由软件置 1 和清零。
0：无影响
1：复位 CRS

位 23 **I2C3RST**：I2C3 复位 (I2C3 reset)
由软件置 1 和复位。
0：无影响
1：复位 I2C3

位 22 保留，必须保持复位值

位 21 **I2C1RST**：I2C1 复位 (I2C1 reset)
由软件置 1 和清零。
0：无影响
1：复位 I2C1

位 20:15 保留，必须保持复位值。

位 14 **SPI2RST**：SPI2 复位 (SPI2 reset)
由软件置 1 和清零。
0：无影响
1：复位 SPI2

位 13:10 保留，必须保持复位值。

位 9 **LCDRST**：LCD 接口复位 (LCD interface reset)
由软件置 1 和清零。
0：无影响
1：复位 LCD

位 8:1 保留，必须保持复位值。

位 0 **TIM2RST**：TIM2 定时器复位 (TIM2 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM2

8.4.14 RCC APB1 外设复位寄存器 2 (RCC_APB1RSTR2)

RCC APB1 peripheral reset register 2

偏移地址：0x03C

复位值：0x00000 0000

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM 2RST	Res.	Res.	Res.	Res.	LPUART1 RST
										rw					rw

位 31:6 保留，必须保持复位值。

位 5 **LPTIM2RST**：低功耗定时器 2 复位 (Low-power timer 2 reset)
由软件置 1 和清零。
0：无影响
1：复位 LPTIM2

位 4:1 保留，必须保持复位值。

位 0 **LPUART1RST**：低功耗 UART 1 复位 (Low-power UART 1 reset)
由软件置 1 和清零。
0：无影响
1：复位 LPUART1

8.4.15 RCC APB2 外设复位寄存器 (RCC_APB2RSTR)

RCC APB2 peripheral reset register

偏移地址：0x040

复位值：0x00000 0000

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI1 RST	Res.	Res.	TIM17 RST	TIM16 RST	Res.
										rw			rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1 RST	Res.	SPI1 RST	TIM1 RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw		rw	rw											

位 31:22 保留，必须保持复位值。

位 21 **SAI1RST**：串行音频接口 1 (SAI1) 复位 (Serial audio interface 1 (SAI1) reset)
由软件置 1 和清零。
0：无影响
1：复位 SAI1

位 20:19 保留，必须保持复位值。

位 18 **TIM17RST**：TIM17 定时器复位 (TIM17 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM17 定时器

位 17 **TIM16RST**：TIM16 定时器复位 (TIM16 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM16 定时器

位 16:15 保留，必须保持复位值。

位 14 **USART1RST**：USART1 复位 (USART1 reset)
由软件置 1 和清零。
0：无影响
1：复位 USART1

位 13 保留，必须保持复位值。

位 12 **SPI1RST**：SPI1 复位 (SPI1 reset)
由软件置 1 和清零。
0：无影响
1：复位 SPI1

位 11 **TIM1RST**：TIM1 定时器复位 (TIM1 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM1 定时器

位 10:0 保留，必须保持复位值。

8.4.16 RCC APB3 外设复位寄存器 (RCC_APB3RSTR)

RCC APB3 peripheral reset register

偏移地址：0x044

复位值：0x00000 0000

访问：无等待状态，按字、半字和字节访问

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

位 31:1 保留，必须保持复位值。

位 0 **RFRST**：无线电系统 BLE 和 802.15.4 复位 (Radio system BLE and 802.15.4 reset)
由软件置 1 和清零。
0：无影响
1：复位无线电系统 BLE 和 802.15.4。无线电系统 BLE 和 802.15.4 的复位状态可从 *RCC 控制/状态寄存器 (RCC_CSR)* 中的 RFRSTS 获得。

8.4.17 RCC AHB1 外设时钟使能寄存器 (RCC_AHB1ENR)

RCC AHB1 peripheral clock enable register

偏移地址：0x048

复位值：0x0000 0000

访问：无等待状态，按字、半字和字节访问

注： *当外设时钟未激活时，不支持 CPU1 对外设寄存器进行读写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSCEN
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRCEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAMUX1EN	DMA2EN	DMA1EN
			rw										rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **TSCEN**：CPU1 触摸感应控制器时钟使能 (CPU1 Touch Sensing Controller clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 TSC 时钟
1：为 CPU1 使能 TSC 时钟

位 15:13 保留，必须保持复位值。

位 12 **CRCEN**：CPU1 CRC 时钟使能 (CPU1 CRC clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 CRC 时钟
1：为 CPU1 使能 CRC 时钟

位 11:3 保留，必须保持复位值。

位 2 **DMAMUX1**：CPU1 DMAMUX1 时钟使能 (CPU1 DMAMUX1 clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 DMAMUX1 时钟
1：为 CPU1 使能 DMAMUX1 时钟

位 1 **DMA2EN**：CPU1 DMA2 时钟使能 (CPU1 DMA2 clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 DMA2 时钟
1：为 CPU1 使能 DMA2 时钟

位 0 **DMA1EN**：CPU1 DMA1 时钟使能 (CPU1 DMA1 clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 DMA1 时钟
1：为 CPU1 使能 DMA1 时钟

8.4.18 RCC AHB2 外设时钟使能寄存器 (RCC_AHB2ENR)

RCC AHB2 peripheral clock enable register

偏移地址：0x04C

复位值：0x0000 0000

访问：无等待状态，按字、半字和字节访问

注：　当外设时钟未激活时，不支持 CPU1 对外设寄存器进行读写访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AES1EN
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ADCEN	Res.	Res.	Res.	Res.	Res.	GPIOHEN	Res.	Res.	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
		rw						rw			rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **AES1EN**：CPU1 AES1 加速器时钟使能 (CPU1 AES1 accelerator clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 AES 时钟
1：为 CPU1 使能 AES 时钟

位 15:14 保留，必须保持复位值。

位 13 **ADCEN**：CPU1 ADC 时钟使能 (CPU1 ADC clocks enable)
由软件置 1 和清零。
0：为 CPU1 禁止 ADC 总线和内核时钟
1：为 CPU1 使能 ADC 总线和内核时钟

位 12:8 保留，必须保持复位值。

位 7 **GPIOHEN**：CPU1 IO 端口 H 时钟使能 (CPU1 IO port H clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 IO 端口 H 时钟
1：为 CPU1 使能 IO 端口 H 时钟

位 6:5 保留，必须保持复位值。

位 4 **GPIOEEN**：CPU1 IO 端口 E 时钟使能 (CPU1 IO port E clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 IO 端口 E 时钟
1：为 CPU1 使能 IO 端口 E 时钟

位 3 **GPIODEN**：CPU1 IO 端口 D 时钟使能 (CPU1 IO port D clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 IO 端口 D 时钟
1：为 CPU1 使能 IO 端口 D 时钟

位 2 **GPIOCEN**：CPU1 IO 端口 C 时钟使能 (CPU1 IO port C clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 IO 端口 C 时钟
1：为 CPU1 使能 IO 端口 C 时钟

位 1 **GPIOBEN**：CPU1 IO 端口 B 时钟使能 (CPU1 IO port B clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 IO 端口 B 时钟
1：为 CPU1 使能 IO 端口 B 时钟

位 0 **GPIOAEN**：CPU1 IO 端口 A 时钟使能 (CPU1 IO port A clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 IO 端口 A 时钟
1：为 CPU1 使能 IO 端口 A 时钟

8.4.19 RCC AHB3 和 AHB4 外设时钟使能寄存器 (RCC_AHB3ENR)

RCC AHB3 and AHB4 peripheral clock enable register

偏移地址：0x050

复位值：0x0208 0000

访问：无等待状态，按字、半字和字节访问

注： *当外设时钟未激活时，不支持 CPU1 对外设寄存器进行读写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	FLASH EN	Res.	Res.	Res.	Res.	IPCC EN	HSEM EN	RNG EN	AES2 EN	PKA EN
						rw					rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	QUADSPI EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								

位 31:26 保留，必须保持复位值。

位 25 **FLASHEN**：CPU1 Flash 接口时钟使能 (CPU1 Flash interface clock enable)
只有在 Flash 掉电时，才能将此位清零。由软件置 1 和清零。
0：为 CPU1 禁止 Flash 接口时钟
1：为 CPU1 使能 Flash 接口时钟

位 24:21 保留，必须保持复位值。

位 20 **IPCCEN**：CPU1 IPCC 接口时钟使能 (CPU1 IPCC interface clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 IPCC 时钟
1：为 CPU1 使能 IPCC 时钟

位 19 **HSEMEN**：CPU1 HSEM 时钟使能 (CPU1 HSEM clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 HSEM 时钟
1：为 CPU1 使能 HSEM 时钟

位 18 **RNGEN**：CPU1 真随机数时钟使能 (CPU1 True RNG clocks enable)
由软件置 1 和清零。
0：为 CPU1 禁止真随机数总线和内核时钟
1：为 CPU1 使能真随机数总线和内核时钟

位 17 **AES2EN**：CPU1 AES2 加速器时钟使能 (CPU1 AES2 accelerator clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 AES2 时钟
1：为 CPU1 使能 AES2 时钟

位 16 **PKAEN**：CPU1 PKA 加速器时钟使能 (CPU1 PKA accelerator clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 PKA 时钟
1：为 CPU1 使能 PKA 时钟

位 15:9 保留，必须保持复位值。

位 8 **QUADSPIEN**：CPU1 四线 SPI 存储器接口时钟使能 (CPU1 Quad SPI memory interface clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 QUADSPI 时钟
1：为 CPU1 使能 QUADSPI 时钟

位 7:0 保留，必须保持复位值。

8.4.20 RCC APB1 外设时钟使能寄存器 1 (RCC_APB1ENR1)

RCC APB1 peripheral clock enable register 1

地址：0x058

复位值：0x0000 0400

访问：无等待状态，按字、半字和字节访问

注： *当外设时钟未激活时，不支持 CPU1 对外设寄存器进行读写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM1 EN	Res.	Res.	Res.	Res.	USB EN	Res.	CRS EN	I2C3 EN	Res.	I2C1 EN	Res.	Res.	Res.	Res.	Res.
rw					rw		rw	rw		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SPI2 EN	Res.	Res.	WWDG EN	RTCAPB EN	LCD EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM2 EN
	rw			rs	rw	rw									rw

位 31 **LPTIM1EN**：CPU1 低功耗定时器 1 时钟使能 (CPU1 Low power timer 1 clocks enable)
由软件置 1 和清零。
0：为 CPU1 禁止 LPTIM1 总线和内核时钟
1：为 CPU1 使能 LPTIM1 总线和内核时钟

位 30:27 保留，必须保持复位值。

位 26 **USBEN**：CPU1 USB FS 时钟使能 (CPU1 USB FS clocks enable)
由软件置 1 和清零
0：为 CPU1 禁止 USB FS 总线和内核时钟
1：为 CPU1 使能 USB FS 总线和内核时钟

位 25 保留，必须保持复位值。

位 24 **CRSEN**：CPU1 CRS 时钟使能 (CPU1 CRS clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 CRS 时钟
1：为 CPU1 使能 CRS 时钟

位 23 **I2C3EN**：CPU1 I2C3 时钟使能 (CPU1 I2C3 clocks enable)
由软件置 1 和清零。
0：为 CPU1 禁止 I2C3 总线和内核时钟
1：为 CPU1 使能 I2C3 总线和内核时钟

位 22 保留，必须保持复位值。

位 21 **I2C1EN**：CPU1 I2C1 时钟使能 (CPU1 I2C1 clocks enable)
由软件置 1 和清零。
0：为 CPU1 禁止 I2C1 总线和内核时钟
1：为 CPU1 使能 I2C1 总线和内核时钟

位 20:15 保留，必须保持复位值。

位 14 **SPI2EN**：CPU1 SPI2 时钟使能 (CPU1 SPI2 clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 SPI2 时钟
1：为 CPU1 使能 SPI2 时钟

位 13:12 保留，必须保持复位值。

位 11 **WWDGEN**：CPU1 窗口看门狗时钟使能 (CPU1 Window watchdog clock enable)
由软件置 1，用于使能窗口看门狗时钟。通过硬件系统复位进行复位。如果 WWDG_SW 选项位复位，该位也可由硬件置 1。
0：为 CPU1 禁止窗口看门狗时钟
1：为 CPU1 使能窗口看门狗时钟

位 10 **RTCAPBEN**：CPU1 RTC APB 时钟使能 (CPU1 RTC APB clock enable)
由软件置 1 和清零
0：为 CPU1 禁止 RTC APB 时钟
1：为 CPU1 使能 RTC APB 时钟

位 9 **LCDEN**：CPU1 LCD 时钟使能 (CPU1 LCD clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 LCD 时钟
1：为 CPU1 使能 LCD 时钟

位 8:1 保留，必须保持复位值。

位 0 **TIM2EN**：CPU1 TIM2 定时器时钟使能 (CPU1 TIM2 timer clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 TIM2 时钟
1：为 CPU1 使能 TIM2 时钟

8.4.21 RCC APB1 外设时钟使能寄存器 2 (RCC_APB1ENR2)

RCC APB1 peripheral clock enable register 2

偏移地址：0x05C

复位值：0x00000 0000

访问：无等待状态，按字、半字和字节访问

注： *当外设时钟未激活时，不支持 CPU1 对外设寄存器进行读写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM2EN	Res.	Res.	Res.	Res.	LPUART1EN
															rw

位 31:6 保留，必须保持复位值。

位 5 **LPTIM2EN**：CPU1 低功耗定时器 2 时钟使能 (CPU1 Low power timer 2 clocks enable)
由软件置 1 和清零。
0：为 CPU1 禁止 LPTIM2 总线和内核时钟
1：为 CPU1 使能 LPTIM2 总线和内核时钟

位 4:1 保留，必须保持复位值。

位 0 **LPUART1EN**：CPU1 低功耗 UART 1 时钟使能 (CPU1 Low power UART 1 clocks enable)
由软件置 1 和清零。
0：为 CPU1 禁止 LPUART1 总线和内核时钟
1：为 CPU1 使能 LPUART1 总线和内核时钟

8.4.22 RCC APB2 外设时钟使能寄存器 (RCC_APB2ENR)

RCC APB2 peripheral clock enable register

地址：0x060

复位值：0x0000 0000

访问：按字、半字和字节访问

注： *当外设时钟未激活时，不支持 CPU1 对外设寄存器进行读写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI1 EN	Res.	Res.	TIM17 EN	TIM16 EN	Res.
										rw			rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1 EN	Res.	SPI1 EN	TIM1 EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw		rw	rw											

位 31:22 保留，必须保持复位值。

位 21 **SAI1EN**：CPU1 SAI1 时钟使能 (CPU1 SAI1 clocks enable)
由软件置 1 和清零。
0：为 CPU1 禁止 SAI1 总线和内核时钟
1：为 CPU1 使能 SAI1 总线和内核时钟

位 20:19 保留，必须保持复位值。

位 18 **TIM17EN**：CPU1 TIM17 定时器时钟使能 (CPU1 TIM17 timer clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 TIM17 定时器时钟
1：为 CPU1 使能 TIM17 定时器时钟

位 17 **TIM16EN**：CPU1 TIM16 定时器时钟使能 (CPU1 TIM16 timer clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 TIM16 定时器时钟
1：为 CPU1 使能 TIM16 定时器时钟

位 16:15 保留，必须保持复位值。

位 14 **USART1EN**：CPU1 USART1 时钟使能 (CPU1 USART1 clocks enable)
由软件置 1 和清零。
0：为 CPU1 禁止 USART1 总线和内核时钟
1：为 CPU1 使能 USART1 总线和内核时钟

位 13 保留，必须保持复位值。

位 12 **SPI1EN**：CPU1 SPI1 时钟使能 (CPU1 SPI1 clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 SPI1 时钟
1：为 CPU1 使能 SPI1 时钟

位 11 **TIM1EN**：CPU1 TIM1 定时器时钟使能 (CPU1 TIM1 timer clock enable)
由软件置 1 和清零。
0：为 CPU1 禁止 TIM1 定时器时钟
1：为 CPU1 使能 TIM1 定时器时钟

位 10:0 保留，必须保持复位值。

8.4.23 睡眠模式下的 RCC AHB1 外设时钟使能寄存器 (RCC_AHB1SMENR)

RCC AHB1 peripheral clocks enable in Sleep modes register

偏移地址：0x068

复位值：0x0001 1207

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSC SMEN
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRC SMEN	Res.	Res.	SRAM1 SMEN	Res.	Res.	Res.	Res.	Res.	Res.	DMAMUX1 SMEN	DMA2 SMEN	DMA1 SMEN
			rw			rw							rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **TSCSMEN**：CPU1 CSleep 模式期间的触摸感应控制器时钟使能 (Touch Sensing Controller clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 TSC 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 TSC 时钟由时钟门控使能，CPU1 CStop 模式期间 TSC 时钟由时钟门控禁止

位 15:13 保留，必须保持复位值。

位 12 **CRCSMEN**：CPU1 CSleep 模式期间的 CRC 时钟使能 (CRC clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 CRC 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 CRC 时钟由时钟门控使能，CPU1 CStop 模式期间 CRC 时钟由时钟门控禁止

位 11:10 保留，必须保持复位值。

位 9 **SRAM1SMEN**：CPU1 CSleep 模式期间的 SRAM1 接口时钟使能 (SRAM1 interface clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 SRAM1 接口时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 SRAM1 接口时钟由时钟门控使能，CPU1 CStop 模式期间 SRAM1 接口时钟由时钟门控禁止

位 8:3 保留，必须保持复位值。

位 2 **DMAMUX1SMEN**：CPU1 CSleep 模式期间的 DMAMUX1 时钟使能 (DMAMUX1 clock enable during CPU1 CSleep mode)

睡眠模式期间由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 DMAMUX1 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 DMAMUX1 时钟由时钟门控使能，CPU1 CStop 模式期间 DMAMUX1 时钟由时钟门控禁止

位 1 **DMA2SMEN**：CPU1 CSleep 模式期间的 DMA2 时钟使能 (DMA2 clock enable during CPU1 CSleep mode)

睡眠模式期间由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 DMA2 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 DMA2 时钟由时钟门控使能，CPU1 CStop 模式期间 DMA2 时钟由时钟门控禁止

位 0 **DMA1SMEN**：CPU1 CSleep 模式期间的 DMA1 时钟使能 (DMA1 clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 DMA1 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 DMA1 时钟由时钟门控使能，CPU1 CStop 模式期间 DMA1 时钟由时钟门控禁止

8.4.24 睡眠模式下的 RCC AHB2 外设时钟使能寄存器 (RCC_AHB2SMENR)

RCC AHB2 peripheral clocks enable in Sleep modes register

偏移地址：0x06C

复位值：0x0001 209F

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AES1SMEN
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ADCSMEN	Res.	Res.	Res.	Res.	Res.	GPIOHSMEN	Res.	Res.	GPIOESMEN	GPIODSMEN	GPIOCSMEN	GPIOBSMEN	GPIOASMEN
		rw						rw			rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **AES1SMEN**：CPU1 CSleep 模式期间的 AES1 加速器时钟使能 (AES1 accelerator clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 AES1 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 AES1 时钟由时钟门控使能，CPU1 CStop 模式期间 AES1 时钟由时钟门控禁止

位 15:14 保留，必须保持复位值。

位 13 **ADCSMEN**：CPU1 CSleep 和 CStop 模式期间的 ADC 时钟使能 (ADC clock enable during CPU1 CSleep and CStop modes)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 ADC 总线时钟由时钟门控禁止。

1：CPU1 CSleep 模式期间 ADC 总线时钟由时钟门控使能，CPU1 CStop 模式期间 ADC 总线时钟由时钟门控禁止。

位 12:8 保留，必须保持复位值。

位 7 **GPIOHSMEN**：CPU1 CSleep 模式期间的 IO 端口 H 时钟使能 (IO port H clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 IO 端口 H 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 IO 端口 H 时钟由时钟门控使能，CPU1 CStop 模式期间 IO 端口 H 时钟由时钟门控禁止

位 6:5 保留，必须保持复位值。

位 4 **GPIOESMEN**：CPU1 CSleep 模式期间的 IO 端口 E 时钟使能 (IO port E clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 IO 端口 E 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 IO 端口 E 时钟由时钟门控使能，CPU1 CStop 模式期间 IO 端口 E 时钟由时钟门控禁止

位 3 **GPIODSMEN**：CPU1 CSleep 模式期间的 IO 端口 D 时钟使能 (IO port D clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 IO 端口 D 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 IO 端口 D 时钟由时钟门控使能，CPU1 CStop 模式期间 IO 端口 D 时钟由时钟门控禁止

位 2 **GPIOCSMEN**：CPU1 CSleep 模式期间的 IO 端口 C 时钟使能 (IO port C clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 IO 端口 C 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 IO 端口 C 时钟由时钟门控使能，CPU1 CStop 模式期间 IO 端口 C 时钟由时钟门控禁止

位 1 **GPIOBSMEN**：CPU1 CSleep 模式期间的 IO 端口 B 时钟使能 (IO port B clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 IO 端口 B 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 IO 端口 B 时钟由时钟门控使能，CPU1 CStop 模式期间 IO 端口 B 时钟由时钟门控禁止

位 0 **GPIOASMEN**：CPU1 CSleep 模式期间的 IO 端口 A 时钟使能 (IO port A clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 IO 端口 A 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 IO 端口 A 时钟由时钟门控使能，CPU1 CStop 模式期间 IO 端口 A 时钟由时钟门控禁止

8.4.25 睡眠和停止模式下的 RCC AHB3 和 AHB4 外设时钟使能寄存器 (RCC_AHB3SMENR)

RCC AHB3 and AHB4 peripheral clocks enable in Sleep and Stop modes register

偏移地址：0x070

复位值：0x0307 0100

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	FLASHSMEN	SRAM2SMEN	Res.	Res.	Res.	Res.	Res.	RNGSMEN	AES2SMEN	PKASMEN
						rw	rw						rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	QUADSPISMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								

位 31:26 保留，必须保持复位值。

位 25 **FLASHSMEN**：CPU1 CSleep 模式期间的 Flash 接口时钟使能 (Flash interface clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 Flash 接口时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 Flash 接口时钟由时钟门控使能，CPU1 CStop 模式期间 Flash 接口时钟由时钟门控禁止

位 24 **SRAM2SMEN**：CPU1 CSleep 模式期间的 SRAM2a 和 SRAM2b 存储器接口时钟使能 (SRAM2a and SRAM2b memory interface clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 SRAM2 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 SRAM2 时钟由时钟门控使能，CPU1 CStop 模式期间 SRAM2 时钟由时钟门控禁止

位 23:19 保留，必须保持复位值。

位 18 **RNGSMEN**：CPU1 CSleep 和 CStop 模式期间的真随机数时钟使能 (True RNG clock enable during CPU1 CSleep and CStop modes)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间真随机数总线时钟由时钟门控禁止

1：CPU1 CSleep 模式期间真随机数总线时钟由时钟门控使能，CPU1 CStop 模式期间真 RNG 总线时钟由时钟门控禁止

位 17 **AES2MEN**：CPU1 CSleep 模式期间的 AES2 加速器时钟使能 (AES2 accelerator clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 AES2 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 AES2 时钟由时钟门控使能，CPU1 CStop 模式期间 AES2 时钟由时钟门控禁止

位 16 **PKASMEN**：CPU1 CSleep 模式期间的 PKA 加速器时钟使能 (PKA accelerator clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 PKA 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 PKA 时钟由时钟门控使能，CPU1 CStop 模式期间 PKA 时钟由时钟门控禁止

位 15:9 保留，必须保持复位值。

位 8 **QUADSPISMEN**：CPU1 CSleep 模式期间的四线 SPI 存储器接口时钟使能 (Quad SPI memory interface clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 QUADSPI 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 QUADSPI 时钟由时钟门控使能，CPU1 CStop 模式期间 QUADSPI 时钟由时钟门控禁止

位 7:0 保留，必须保持复位值。

8.4.26 睡眠模式下的 RCC APB1 外设时钟使能寄存器 1 (RCC_APB1SMENR1)

RCC APB1 peripheral clocks enable in Sleep mode register 1

地址：0x078

复位值：0x85A0 4E01

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM1 SMEN	Res.	Res.	Res.	Res.	USB SMEN	Res.	CRS SMEN	I2C3 SMEN	Res.	I2C1 SMEN	Res.	Res.	Res.	Res.	Res.
rw					rw		rw	rw		rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SPI2 SMEN	Res.	Res.	WWDG SMEN	RTCAPB SMEN	LCD SMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM2 SMEN
	rw			rw	rw	rw									rw

位 31 **LPTIM1SMEN**：CPU1 CSleep 和 CStop 模式期间的低功耗定时器 1 时钟使能 (Low power timer 1 clock enable during CPU1 CSleep and CStop mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 LPTIM1 总线时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 LPTIM1 总线时钟由时钟门控使能，CPU1 CStop 模式期间 LPTIM1 总线时钟由时钟门控禁止

位 30:27 保留，必须保持复位值。

位 26 **USBSMEN**：CPU1 CSleep 和 CStop 模式期间的 USB FS 时钟使能 (USB FS clock enable during CPU1 CSleep and CStop modes)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 USB FS 总线时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 USB FS 总线时钟由时钟门控使能，CPU1 CStop 模式期间 USB FS 总线时钟由时钟门控禁止

位 25 保留，必须保持复位值。

位 24 **CRSSMEN**：CPU1 CSleep 模式期间的 CRS 时钟使能 (CRS clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 CRS 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 CRS 时钟由时钟门控使能，CPU1 CStop 模式期间 CRS 时钟由时钟门控禁止

位 23 **I2C3SMEN**：CPU1 CSleep 和 CStop 模式期间的 I2C3 时钟使能 (I2C3 clock enable during CPU1 CSleep and CStop modes)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 I2C3 总线时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 I2C3 总线时钟由时钟门控使能，CPU1 CStop 模式期间 I2C3 总线时钟由时钟门控禁止

位 22 保留，必须保持复位值。

位 21 **I2C1SMEN**：CPU1 CSleep 和 CStop 模式期间的 I2C1 时钟使能 (I2C1 clock enable during CPU1 CSleep and CStop modes)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 I2C1 总线时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 I2C1 总线时钟由时钟门控使能，CPU1 CStop 模式期间 I2C1 总线时钟由时钟门控禁止

位 20:15 保留，必须保持复位值。

位 14 **SPI2SMEN**：CPU1 CSleep 模式期间的 SPI2 时钟使能 (SPI2 clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 SPI2 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 SPI2 时钟由时钟门控使能，CPU1 CStop 模式期间 SPI2 时钟由时钟门控禁止

位 13:12 保留，必须保持复位值。

位 11 **WWDGSMEN**：CPU1 CSleep 模式期间的窗口看门狗时钟使能 (Window watchdog clocks enable during CPU1 CSleep mode)

由软件置 1 和清零。硬件 WWDG_SW 选项复位时，该位由硬件强制置“1”。

0：CPU1 CSleep 和 CStop 模式期间窗口看门狗时钟由时钟门控禁止

1：CPU1 CSleep 模式期间窗口看门狗时钟由时钟门控使能，CPU1 CStop 模式期间窗口看门狗时钟由时钟门控禁止

位 10 **RTCAPBSMEN**：CPU1 CSleep 模式期间的 RTC 总线时钟使能 (RTC bus clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 RTC 总线时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 RTC 总线时钟由时钟门控使能，CPU1 CStop 模式期间 RTC 总线时钟由时钟门控禁止

位 9 **LCDSMEN**：CPU1 CSleep 模式期间的 LCD 时钟使能 (LCD clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 LCD 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 LCD 时钟由时钟门控使能，CPU1 CStop 模式期间 LCD 时钟由时钟门控禁止

位 8:1 保留，必须保持复位值。

位 0 **TIM2SMEN**：CPU1 CSleep 模式期间的 TIM2 定时器时钟使能 (TIM2 timer clock enable during CPU1 CSleep mode)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 TIM2 时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 TIM2 时钟由时钟门控使能，CPU1 CStop 模式期间 TIM2 时钟由时钟门控禁止

8.4.27 睡眠模式下的 RCC APB1 外设时钟使能寄存器 2 (RCC_APB1SMENR2)

RCC APB1 peripheral clocks enable in Sleep mode register 2

偏移地址：0x07C

复位值：0x0000 0021

访问：无等待状态，按字、半字和字节访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM2 SMEN	Res.	Res.	Res.	Res.	LPUART1 SMEN
										rw					rw

位 31:6 保留，必须保持复位值。

位 5 **LPTIM2SMEN**：CPU1 CSleep 和 CStop 模式期间的低功耗定时器 2 时钟使能 (Low power timer 2 clock enable during CPU1 CSleep and CStop modes)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 LPTIM2 总线时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 LPTIM2 总线时钟由时钟门控使能，CPU1 CStop 模式期间 LPTIM2 总线时钟由时钟门控禁止

位 4:1 保留，必须保持复位值。

位 0 **LPUART1SMEN**：CPU1 CSleep 和 CStop 模式期间的低功耗 UART 1 时钟使能 (Low power UART 1 clock enable during CPU1 CSleep and CStop modes)

由软件置 1 和清零。

0：CPU1 CSleep 和 CStop 模式期间 LPUART1 总线时钟由时钟门控禁止

1：CPU1 CSleep 模式期间 LPUART1 总线时钟由时钟门控使能，CPU1 CStop 模式期间 LPUART1 总线时钟由时钟门控禁止

8.4.28 睡眠模式下的 RCC APB2 外设时钟使能寄存器 (RCC_APB2SMENR)

RCC APB2 peripheral clocks enable in Sleep mode register

地址：0x080

复位值：0x0026 5800

访问：按字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI1 SMEN	Res.	Res.	TIM17 SMEN	TIM16 SMEN	Res.
										rw			rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1 SMEN	Res.	SPI1 SMEN	TIM1 SMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw		rw	rw											

位 31:22 保留，必须保持复位值。

位 21 **SAI1SMEN**：CPU1 CSleep 和 CStop 模式期间的 SAI1 时钟使能 (SAI1 clock enable during CPU1 CSleep and CStop modes)

由软件置 1 和清零。
0：CPU1 CSleep 和 CStop 模式期间 SAI1 总线时钟由时钟门控禁止
1：CPU1 CSleep 模式期间 SAI1 总线时钟由时钟门控使能，CPU1 CStop 模式期间 SAI1 总线时钟由时钟门控禁止

位 20:19 保留，必须保持复位值。

位 18 **TIM17SMEN**：CPU1 CSleep 模式期间的 TIM17 定时器时钟使能 (TIM17 timer clock enable during CPU1 CSleep mode)

由软件置 1 和清零。
0：CPU1 CSleep 和 CStop 模式期间 TIM17 定时器时钟由时钟门控禁止
1：CPU1 CSleep 模式期间 TIM17 定时器时钟由时钟门控使能，CPU1 CStop 模式期间 TIM17 定时器时钟由时钟门控禁止

位 17 **TIM16SMEN**：CPU1 CSleep 模式期间的 TIM16 定时器时钟使能 (TIM16 timer clock enable during CPU1 CSleep mode)

由软件置 1 和清零。
0：CPU1 CSleep 和 CStop 模式期间 TIM16 定时器时钟由时钟门控禁止
1：CPU1 CSleep 模式期间 TIM16 定时器时钟由时钟门控使能，CPU1 CStop 模式期间 TIM16 定时器时钟由时钟门控禁止

位 16:15 保留，必须保持复位值。

位 14 **USART1SMEN**：CPU1 CSleep 和 CStop 模式期间的 USART1 时钟使能 (USART1 clock enable during CPU1 CSleep and CStop modes)

由软件置 1 和清零。
0：CPU1 CSleep 和 CStop 模式期间 USART1 总线时钟由时钟门控禁止
1：CPU1 CSleep 模式期间 USART1 总线时钟由时钟门控使能，CPU1 CStop 模式期间 USART1 总线时钟由时钟门控禁止

位 13 保留，必须保持复位值。

位 12 **SPI1SMEN**：CPU1 CSleep 模式期间的 SPI1 时钟使能 (SPI1 clock enable during CPU1 CSleep mode)

由软件置 1 和清零。
0：CPU1 CSleep 和 CStop 模式期间 SPI1 时钟由时钟门控禁止
1：CPU1 CSleep 模式期间 SPI1 时钟由时钟门控使能，CPU1 CStop 模式期间 SPI1 时钟由时钟门控禁止

位 11 **TIM1SMEN**：CPU1 CSleep 模式期间的 TIM1 定时器时钟使能 (TIM1 timer clock enable during CPU1 CSleep mode)

由软件置 1 和清零。
0：CPU1 CSleep 和 CStop 模式期间 TIM1 定时器时钟由时钟门控禁止
1：CPU1 CSleep 模式期间 TIM1 定时器时钟由时钟门控使能，CPU1 CStop 模式期间 TIM1 定时器时钟由时钟门控禁止

位 10:0 保留，必须保持复位值。

8.4.29 RCC 外设独立时钟配置寄存器 (RCC_CCIPR)

RCC peripherals independent clock configuration register

地址：0x088

复位值：0x0000 0000

访问：无等待状态，按字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RNGSEL[1:0]		ADCSEL[1:0]		CLK48SEL[1:0]		Res.	Res.	SAI1SEL[1:0]		LPTIM2SEL[1:0]		LPTIM1SEL[1:0		I2C3SEL[1:0]	
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	I2C1SEL[1:0]		LPUART1SEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USART1SEL[1:0]	
		rw	rw	rw	rw									rw	rw

位 31:30 **RNGSEL[1:0]**：RNG 时钟源选择 (RNG clock source selection)

这些位由软件置 1 和清零，用于选择真 RNG 使用的时钟源。

00：使用由 CLK48SEL 选择的时钟

01：选择 LSI 时钟

10：选择 LSE 时钟

11：保留

位 29:28 **ADCSEL[1:0]**：ADC 时钟源选择 (ADC clock source selection)

这些位由软件置 1 和清零，用于选择 ADC 接口使用的时钟源。

00：没有选择时钟

01：选择 PLLSAI1“R”时钟 (PLLSAI1RCLK) 作为 ADC 时钟

10：选择 PLL“P”时钟 (PLLPCLK) 作为 ADC 时钟

11：选择系统时钟 (SYSCLK) 作为 ADC 时钟

位 27:26 **CLK48SEL[1:0]**：48 MHz 时钟源选择 (48 MHz clock source selection)

这些位由软件置 1 和清零，用于选择 USB 和真随机数使用的 48 MHz 时钟源。真随机数时钟源由 RNGSEL 进一步选择。

00：选择 HSI48 时钟作为 48 MHz 时钟

01：选择 PLLSAI1“Q”时钟 (PLLSAI1QCLK) 作为 48 MHz 时钟

10：选择 PLL“Q”时钟 (PLLQCLK) 作为 48 MHz 时钟

11：选择 MSI 时钟作为 48 MHz 时钟

位 25:24 保留，必须保持复位值。

位 23:22 **SAI1SEL[1:0]**：SAI1 时钟源选择 (SAI1 clock source selection)

这些位由软件置 1 和清零，用于选择 SAI1 时钟源。

00：选择 PLLSAI1“P”时钟 (PLLSAI1PCLK) 作为 SAI1 时钟

01：选择 PLL“P”时钟 (PLLPCLK) 作为 SAI1 时钟

10：选择 HSI16 时钟作为 SAI1 时钟

11：选择外部输入 SAI1_EXTCLK 作为 SAI1 时钟

注意： 当选择外部时钟时，如果外部时钟不存在，将无法切换到另一个时钟。

位 21:20 **LPTIM2SEL[1:0]**：低功耗定时器 2 时钟源选择 (Low power timer 2 clock source selection)
这些位由软件置 1 和清零，用于选择 LPTIM2 时钟源。
00：选择 PCLK 作为 LPTIM2 时钟
01：选择 LSI 时钟作为 LPTIM2 时钟
10：选择 HSI16 时钟作为 LPTIM2 时钟
11：选择 LSE 时钟作为 LPTIM2 时钟

位 19:18 **LPTIM1SEL[1:0]**：低功耗定时器 1 时钟源选择 (Low power timer 1 clock source selection)
这些位由软件置 1 和清零，用于选择 LPTIM1 时钟源。
00：选择 PCLK 作为 LPTIM1 时钟
01：选择 LSI 时钟作为 LPTIM1 时钟
10：选择 HSI16 时钟作为 LPTIM1 时钟
11：选择 LSE 时钟作为 LPTIM1 时钟

位 17:16 **I2C3SEL[1:0]**：I2C3 时钟源选择 (I2C3 clock source selection)
这些位由软件置 1 和清零，用于选择 I2C3 时钟源。
00：选择 PCLK 作为 I2C3 时钟
01：选择系统时钟 (SYSCLK) 作为 I2C3 时钟
10：选择 HSI16 时钟作为 I2C3 时钟
11：保留

位 15:14 保留，必须保持复位值。

位 13:12 **I2C1SEL[1:0]**：I2C1 时钟源选择 (I2C1 clock source selection)
这些位由软件置 1 和清零，用于选择 I2C1 时钟源。
00：选择 PCLK 作为 I2C1 时钟
01：选择系统时钟 (SYSCLK) 作为 I2C1 时钟
10：选择 HSI16 时钟作为 I2C1 时钟
11：保留

位 11:10 **LPUART1SEL[1:0]**：LPUART1 时钟源选择 (LPUART1 clock source selection)
这些位由软件置 1 和清零，用于选择 LPUART1 时钟源。
00：选择 PCLK 作为 LPUART1 时钟
01：选择系统时钟 (SYSCLK) 作为 LPUART1 时钟
10：选择 HSI16 时钟作为 LPUART1 时钟
11：选择 LSE 时钟作为 LPUART1 时钟

位 9:2 保留，必须保持复位值。

位 1:0 **USART1SEL[1:0]**：USART1 时钟源选择 (USART1 clock source selection)
此位由软件置 1 和清零，用于选择 USART1 事件源。
00：选择 PCLK 作为 USART1 时钟
01：选择系统时钟 (SYSCLK) 作为 USART1 时钟
10：选择 HSI16 时钟作为 USART1 时钟
11：选择 LSE 时钟作为 USART1 时钟

8.4.30 RCC 备份域控制寄存器 (RCC_BDCR)

RCC backup domain control register

偏移地址：0x090

复位值：0x0000 0000（除 LSCOSEL、LSCOEN 和 BDRST 只能通过备份域上电复位进行复位以外，其他位均通过备份域复位进行复位，而不是通过从待机模式唤醒和 NRST 焊盘进行复位）。

访问：0 ≤ 等待状态 ≤ 3，按字、半字和字节访问
连续访问该寄存器时，则插入等待状态。

注： *RCC 备份域控制寄存器 (RCC_BDCR) 的位在 V_{CORE} 域之外。因而复位后，这些位受写保护，必须将 PWR 控制寄存器 1 (PWR_CR1) 中的 DBP 位置 1 才能修改这些位。有关详细信息，请参见第 6.1.4 节：电池备份域。只有备份域复位后，这些位（LSCOSEL、LSCOEN 和 BDRST 除外）才能复位。内部复位和外部复位对这些位不会有任何影响。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	LSCOSEL	LSCOEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BDRST
						rw	rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	Res.	Res.	Res.	Res.	Res.	RTCSEL[1:0]		Res.	LSECSSD	LSECSSON	LSEDRV[1:0]		LSEBYP	LSERDY	LSEON
rw						rw	rw		r	rw	rw	rw	rw	r	rw

位 31:26 保留，必须保持复位值。

位 25 **LSCOSEL**：低速时钟输出选择 (Low speed clock output selection)
由软件置 1 和清零。
0：选择 LSI 时钟
1：选择 LSE 时钟

位 24 **LSCOEN**：低速时钟输出使能 (Low speed clock output enable)
由软件置 1 和清零。
0：禁止低速时钟输出 (LSCO)
1：使能低速时钟输出 (LSCO)

位 23:17 保留，必须保持复位值。

位 16 **BDRST**：备份域软件复位 (Backup domain software reset)
由软件置 1 和清零。
0：复位未激活
1：复位整个备份域

位 15 **RTCEN**：RTC 时钟使能 (RTC clock enable)
由软件置 1 和清零。
0：禁止 RTC 时钟
1：使能 RTC 时钟

位 14:10 保留，必须保持复位值。

位 9:8 **RTCSEL[1:0]**：RTC 时钟源选择 (RTC clock source selection)

由软件置 1，用于选择 RTC 的时钟源。选择 RTC 时钟源后，除非备份域复位或在 LSE 上检测到故障（LSECSSD 置 1），否则不可再将其更改。可使用 BDRST 位对其进行复位。

00：无时钟

01：LSE 振荡器时钟用作 RTC 时钟

10：LSI 振荡器时钟用作 RTC 时钟

11：32 分频的 HSE 振荡器时钟用作 RTC 时钟

位 7 保留，必须保持复位值。

位 6 **LSECSSD**：LSE 上的 CSS 故障检测 (CSS on LSE failure Detection)

由硬件置 1，用于指示外部 32 kHz 振荡器 (LSE) 上的
时钟安全系统何时检测到故障。

0：LSE（32 kHz 振荡器）上未检测到故障

1：LSE（32 kHz 振荡器）上检测到故障

位 5 **LSECSSON**：LSE 上的 CSS 使能 (CSS on LSE enable)

由软件置 1，用于使能 LSE（32 kHz 振荡器）上的时钟安全系统。

在 LSE 振荡器使能（已使能 LSEON 位）和就绪（由硬件将 LSERDY 标志置 1）后以及在选择 RTCSEL 位后，LSECSSON 必须使能。

该位一旦使能便不能禁止，但检测到 LSE 故障 (LSECSSD=1) 后除外。此时，软件必须禁止 LSECSSON 位。

0：LSE（32 kHz 外部振荡器）上的 CSS 关闭

1：LSE（32 kHz 外部振荡器）上的 CSS 开启

位 4:3 **LSEDRV[1:0]**：LSE 振荡器驱动能力 (LSE oscillator drive capability)

由软件置 1，用于调整 LSE 振荡器的驱动能力。

00：“Xtal 模式”低驱动能力

01：“Xtal 模式”中低驱动能力

10：“Xtal 模式”中高驱动能力

11：“Xtal 模式”高驱动能力

当振荡器不处于旁路模式时，则处于 Xtal 模式。

位 2 **LSEBYP**：LSE 振荡器旁路 (LSE oscillator bypass)

由软件置 1 和清零，用于旁路振荡器。只有在禁止外部 32 kHz 振荡器（LSEON=0 且 LSERDY=0）后才能写入该位。

0：不旁路 LSE 振荡器

1：旁路 LSE 振荡器

位 1 **LSERDY**：LSE 振荡器就绪 (LSE oscillator ready)

由硬件置 1 和清零，用于指示外部 32 kHz 振荡器已稳定。在 LSEON 位被清零后，LSERDY 将在 6 个外部低速振荡器时钟周期后转为低电平。

0：LSE 振荡器未就绪

1：LSE 振荡器已就绪

位 0 **LSEON**：LSE 振荡器使能 (LSE oscillator enable)

由软件置 1 和清零。

0：LSE 振荡器关闭

1：LSE 振荡器开启

8.4.31 RCC 控制 / 状态寄存器 (RCC_CSR)

RCC control/status register

地址：0x094

复位值：0x0C00 0000。通过 NRST 焊盘来复位（仅与 POR 相关的复位标志除外），而不是通过从待机模式唤醒来复位。

访问：0 ≤ 等待状态 ≤ 3，按字、半字和字节访问

连续访问该寄存器时，则插入等待状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWR RSTF	WWDG RSTF	IWWG RSTF	SFT RSTF	BOR RSTF	PIN RSTF	OB L RSTF	Res.	RMVF	Res.	Res.	Res.	Res.	Res.	Res.	RF RSTS
r	r	r	r	r	r	r		rw							r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFWKPSEL[1:0]		Res.	Res.	LSI2TRIM[3:0]				Res.	Res.	Res.	Res.	LSI2 RDY	LSI2 ON	LSI1 RDY	LSI1 ON
rw	rw			rw	rw	rw	rw					r	rw	r	rw

位 31 **LPWRRSTF**：低功耗复位标志 (Low-power reset flag)
由于进入非法停止、待机或关断模式而发生复位时，由硬件置 1。
通过写入 RMVF 位清零。
0：未发生非法模式复位
1：发生非法模式复位

位 30 **WWDGRSTF**：窗口看门狗复位标志 (Window watchdog reset flag)
发生窗口看门狗复位时，由硬件置 1。
通过写入 RMVF 位清零。
0：未发生窗口看门狗复位
1：发生窗口看门狗复位

位 29 **IWDGRSTF**：独立窗口看门狗复位标志 (Independent window watchdog reset flag)
发生独立看门狗复位时，由硬件置 1。
通过写入 RMVF 位清零。
0：未发生独立看门狗复位
1：发生独立看门狗复位

位 28 **SFTRSTF**：软件复位标志 (Software reset flag)
发生软件复位时，由硬件置 1。
通过写入 RMVF 位清零。
0：未发生软件复位
1：发生软件复位

位 27 **BORRSTF**：BOR 标志 (BOR flag)
发生 BOR 时，由硬件置 1。
通过写入 RMVF 位清零。
0：未发生 BOR
1：发生 BOR

位 26 **PINRSTF**：引脚复位标志 (Pin reset flag)
发生来自 NRST 引脚的复位时，由硬件置 1。
通过写入 RMVF 位清零。
0：未发生来自 NRST 引脚的复位
1：发生来自 NRST 引脚的复位

位 25 **OBLRSTF**：选项字节加载复位标志 (Option byte loader reset flag)
发生来自选项字节加载的复位时，由硬件置 1。
通过写入 RMVF 位清零。
0：未发生来自选项字节加载的复位
1：发生来自选项字节加载的复位

位 24 保留，必须保持复位值。

位 23 **RMVF**：清除复位标志 (Remove reset flag)
由软件置 1，用于将复位标志清零。
0：无影响
1：清除复位标志

位 22:17 保留，必须保持复位值。

位 16 **RFRSTS**：无线电系统 BLE 和 802.15.4 复位状态 (Radio system BLE and 802.15.4 reset status)
由硬件置 1 和清零
0：无线电系统 BLE 和 802.15.4 未处于复位状态，可访问无线电系统。
1：无线电系统 BLE 和 802.15.4 处于复位状态，不可访问无线电系统。

位 15:14 **RFWKPSEL[1:0]**：RF 系统唤醒时钟源选择 (RF system wakeup clock source selection)
由软件置 1，用于选择 RF 系统唤醒逻辑的时钟源。
00：无时钟
01：LSE 振荡器时钟用作 RF 系统唤醒时钟
10：LSI 振荡器时钟用作 RF 系统唤醒时钟
11：1024 分频的 HSE 振荡器时钟用作 RF 系统唤醒时钟

位 13:12 保留，必须保持复位值。

位 11:8 **LSI2TRIM[3:0]**：LSI2 振荡器微调 (LSI2 oscillator trim)
注： 只有当 LSI2 禁止时 (LSI2ON = 0) 才能更改 LSI2TRIM。

位 7:4 保留，必须保持复位值。

位 3 **LSI2RDY**：LSI2 振荡器已就绪 (LSI2 oscillator ready)
由硬件置 1 和清零，用于指示 LSI2 振荡器已稳定。在将 LSI2ON 位清零后，LSI2RDY 将在 3 个 LSI2 振荡器时钟周期后转为低电平。
0：LSI2 振荡器未就绪
1：LSI2 振荡器已就绪

位 2 **LSI2ON**：LSI2 振荡器使能和选择 (LSI2 oscillator enable and selection)
由软件置 1 和清零。
0：LSI2 振荡器关闭（LSI 选择的是 LSI1）
1：LSI2 振荡器开启（LSI2 就绪时，LSI 选择的是 LSI2）

位 1 **LSI1RDY**：LSI1 振荡器已就绪 (LSI1 oscillator ready)
由硬件置 1 和清零，用于指示 LSI1 振荡器已稳定。在将 LSI1ON 位清零后，LSI1RDY 将在 3 个 LSI1 振荡器时钟周期后转为低电平。如果存在 LSE 上的时钟安全系统、独立看门狗或 RTC 发出的 LSI1 请求，即使 LSI1ON = 0，该位也可置 1。
0：LSI1 振荡器未就绪
1：LSI1 振荡器已就绪

位 0 **LSI1ON**：LSI1 振荡器使能 (LSI1 oscillator enable)
由软件置 1 和清零。
0：LSI1 振荡器关闭
1：LSI1 振荡器开启

8.4.32 RCC 时钟恢复 RC 寄存器 (RCC_CRRCR)

RCC Clock recovery RC register

地址：0x098

复位值：0x0000 XXX0，其中 X 是出厂前编程的。

访问：无等待状态，按字、半字和字节访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSI48CAL[8:0]									Res.	Res.	Res.	Res.	Res.	HSI48 RDY	HSI48 ON
r	r	r	r	r	r	r	r	r						r	rw

位 31:16 保留，必须保持复位值

位 15:7 **HSI48CAL[8:0]**：HSI48 时钟校准 (HSI48 clock calibration)

这些位在启动时初始化为出厂前编程的 HSI48 校准微调值。

它们是只读的。

位 6:2 保留，必须保持复位值

位 1 **HSI48RDY**：HSI48 时钟就绪标志 (HSI48 clock ready flag)

由硬件置 1，用以指示 HSI48 振荡器已稳定。仅当通过软件将 HSI48ON 置 1 来使能 HSI48 时，该位才置 1。

0：HSI48 振荡器未就绪

1：HSI48 振荡器已就绪

位 0 **HSI48ON**：HSI48 时钟使能 (HSI48 clock enable)

由软件置 1 和清零。

由硬件清零，用于在进入停止、待机和关断模式时停止 HSI48。

0：HSI48 振荡器关闭

1：HSI48 振荡器开启

8.4.33 RCC 时钟 HSE 寄存器 (RCC_HSECR)

RCC clock HSE register

地址：0x09C

复位值：0x0000 0030（通过 NRST 焊盘来复位，而不会通过从待机模式唤醒来复位）。

访问：具有写访问保护，软件必须在该寄存器中写入一个密钥 (0xCAFE CAFE) 作为字访问密钥来解锁寄存器。解锁后，即可对寄存器执行单次写访问（字、半字或字节）。随后会再次锁定。无等待状态。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	HSETUNE[5:0]						Res.	HSEGMC[2:0]			HSES	Res.	Res.	UNLOCKED
		r	r	r	r	r	r		rw	rw	rw	rw			r

位 31:14 保留，必须保持复位值

位 13:8 **HSETUNE[5:0]**：HSE 电容调节 (HSE capacitor tuning)
可通过软件更改。不得在 HSE 开启时更改。
0x00：最小负载电容。
0x3F：最大负载电容。

位 7 保留，必须保持复位值

位 6:4 **HSEGMC[2:0]**：HSE 电流控制 (HSE current control)
可通过软件更改。不得在 HSE 开启时更改。
000：电流最大限值为 0.18 mA/V。
001：电流最大限值为 0.57 mA/V。
010：电流最大限值为 0.78 mA/V。
011：电流最大限值为 1.13 mA/V。
100：电流最大限值为 0.61 mA/V。
101：电流最大限值为 1.65 mA/V。
110：电流最大限值为 2.12 mA/V。
111：电流最大限值为 2.84 mA/V。

位 3 **HSES**：HSE 感应放大器阈值 (HSE Sense amplifier threshold)
可通过软件更改。不得在 HSE 开启时更改。
0：HSE 偏置电流系数为 1/2。
1：HSE 偏置电流系数为 3/4。

位 2:1 保留，必须保持复位值

位 0 **UNLOCKED**：HSE 时钟控制寄存器解锁 (HSE clock control register unlocked)
由硬件置 1 和清零。
0：寄存器锁定，密钥未编程，或已对寄存器执行写访问。
1：寄存器解锁。已写入密钥 0xCAFE CAFE 解锁该寄存器，因而允许执行单次数据写入操作。

8.4.34 RCC 扩展时钟恢复寄存器 (RCC_EXTCFGR)

RCC extended clock recovery register

地址：0x108

复位值：0x0003 0000。

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFCSS	Res.	Res.	C2HPREF	SHDHPREF
											r			r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	C2HPRE[3:0]				SHDHPRE[3:0]			
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值

位 20 **RFCSS**：无线电系统 HCLK5 和 APB3 所选时钟源指示 (Radio system HCLK5 and APB3 selected clock source indication)
由硬件置 1 和复位，用于指示为无线电系统 HCLK5 和 APB3 时钟选择哪个时钟源。
0：无线电系统 HCLK5 和 APB3 时钟使用 HSI16。
1：无线电系统 HCLK5 和 APB3 时钟使用 2 分频的 HSE。

位 19:18 保留，必须保持复位值

位 17 **C2HPREF**：HCLK2 预分频器标志 (CPU2) (HCLK2 prescaler flag (CPU2))
由硬件置 1 和复位，用于确认 HCLK2 预分频器编程。
在 C2HPRE 中编程新的预分频值时复位。在实际应用编程值时置 1。
0：尚未应用 HCLK2 预分频值。
1：已应用 HCLK2 预分频值。

位 16 **SHDHPREF**：HCLK4 共用预分频器标志（AHB4、Flash 和 SRAM2）(HCLK4 shared prescaler flag (AHB4, Flash memory and SRAM2))
由硬件置 1 和复位，用于确认共用 HCLK4 预分频器编程。
在 SHDHPRE 中编程新的预分频值时复位。在实际应用编程值时置 1。
0：尚未应用 HCLK4 预分频值。
1：已应用 HCLK4 预分频值。

位 15:8 保留，必须保持复位值

位 7:4 **C2HPRE[3:0]**：HCLK2 预分频器 (CPU2) (HCLK2 prescaler (CPU2))
由软件置 1 和清零，用于控制 HCLK2 时钟 (CPU2) 的分频系数。
可以检查 C2HPREF 标志以确认是否应用了编程的 C2HPRE 预分频值。

注意： 软件必须根据器件电压范围正确设置这些位，以确保系统频率不会超过允许的最大频率（更多详细信息，请参见*第 6.1.6 节：动态电压调节管理*）。在对这些位执行写操作之后、减小电压范围之前，必须读取寄存器位 C2HPREF 以确保其中的值为新值。

0001：SYSCLK 3 分频
0010：SYSCLK 5 分频
0101：SYSCLK 6 分频
0110：SYSCLK 10 分频
0111：SYSCLK 32 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
其他：SYSCLK 不分频

位 3:0 **SHDHPRE[3:0]**：HCLK4 共用预分频器（AHB4、Flash 和 SRAM2）(HCLK4 shared prescaler (AHB4, Flash memory and SRAM2))

由软件置 1 和清零，用于控制共用 HCLK4 时钟（AHB4、Flash 和 SRAM2）的分频系数。可以检查 SHDHPREF 标志以确认是否应用了编程的 SHDHPRE 预分频值。

注意： 软件必须根据器件电压范围正确设置这些位，以确保系统频率不会超过允许的最大频率（更多详细信息，请参见*第 6.1.6 节：动态电压调节管理*）。在对这些位执行写操作之后、减小电压范围之前，必须读取寄存器位 SHDHPRE 以确保其中的值为新值。

0001：SYSCLK 3 分频
0010：SYSCLK 5 分频
0101：SYSCLK 6 分频
0110：SYSCLK 10 分频
0111：SYSCLK 32 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
其他：SYSCLK 不分频

8.4.35 RCC CPU2 AHB1 外设时钟使能寄存器 (RCC_C2AHB1ENR)

RCC CPU2 AHB1 peripheral clock enable register

偏移地址：0x148

复位值：0x0000 0000

访问：无等待状态，按字、半字和字节访问

注： *当外设时钟未激活时，不支持 CPU2 对外设寄存器进行读写访问。*

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRCEN	Res.	Res.	SRAM1EN	Res.	Res.	Res.	Res.	Res.	Res.	DMAMUX1EN	DMA2EN	DMA1EN
			rw			rw							rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **TSCEN**：CPU2 触摸感应控制器时钟使能 (CPU2 Touch Sensing Controller clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 TSC 时钟
1：为 CPU2 使能 TSC 时钟

位 15:13 保留，必须保持复位值。

位 12 **CRCEN**：CPU2 CRC 时钟使能 (CPU2 CRC clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 CRC 时钟
1：为 CPU2 使能 CRC 时钟

位 11:10 保留，必须保持复位值。

位 9 **SRAM1EN**：CPU2 SRAM1 时钟使能 (CPU2 SRAM1 clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 SRAM1 时钟
1：为 CPU2 使能 SRAM1 时钟

位 8:3 保留，必须保持复位值。

位 2 **DMAMUX1**：CPU2 DMAMUX1 时钟使能 (CPU2 DMAMUX1 clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 DMAMUX1 时钟
1：为 CPU2 使能 DMAMUX1 时钟

位 1 **DMA2EN**：CPU2 DMA2 时钟使能 (CPU2 DMA2 clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 DMA2 时钟
1：为 CPU2 使能 DMA2 时钟

位 0 **DMA1EN**：CPU2 DMA1 时钟使能 (CPU2 DMA1 clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 DMA1 时钟
1：为 CPU2 使能 DMA1 时钟

8.4.36 RCC CPU2 AHB2 外设时钟使能寄存器 (RCC_C2AHB2ENR)

RCC CPU2 AHB2 peripheral clock enable register

偏移地址：0x14C

复位值：0x0000 0000

访问：无等待状态，按字、半字和字节访问

注： *当外设时钟未激活时，不支持 CPU2 对外设寄存器进行读写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AES1EN
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ADCEN	Res.	Res.	Res.	Res.	Res.	GPIOHEN	Res.	Res.	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
		rw						rw			rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **AES1EN**：CPU2 AES1 加速器时钟使能 (CPU2 AES1 accelerator clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 AES 时钟
1：为 CPU2 使能 AES 时钟

位 15:14 保留，必须保持复位值。

位 13 **ADCEN**：ADC 时钟使能 (ADC clocks enable)
由软件置 1 和清零。
0：为 CPU2 禁止 ADC 总线和内核时钟
1：为 CPU2 使能 ADC 总线和内核时钟

位 12:8 保留，必须保持复位值。

位 7 **GPIOHEN**：CPU2 IO 端口 H 时钟使能 (CPU2 IO port H clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 IO 端口 H 时钟
1：为 CPU2 使能 IO 端口 H 时钟

位 6:5 保留，必须保持复位值。

位 4 **GPIOEEN**：CPU2 IO 端口 E 时钟使能 (CPU2 IO port E clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 IO 端口 E 时钟
1：为 CPU2 使能 IO 端口 E 时钟

位 3 **GPIODEN**：CPU2 IO 端口 D 时钟使能 (CPU2 IO port D clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 IO 端口 D 时钟
1：为 CPU2 使能 IO 端口 D 时钟

位 2 **GPIOCEN**：CPU2 IO 端口 C 时钟使能 (CPU2 IO port C clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 IO 端口 C 时钟
1：为 CPU2 使能 IO 端口 C 时钟

位 1 **GPIOBEN**：CPU2 IO 端口 B 时钟使能 (CPU2 IO port B clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 IO 端口 B 时钟
1：为 CPU2 使能 IO 端口 B 时钟

位 0 **GPIOAEN**：CPU2 IO 端口 A 时钟使能 (CPU2 IO port A clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 IO 端口 A 时钟
1：为 CPU2 使能 IO 端口 A 时钟

8.4.37 RCC CPU2 AHB3 和 AHB4 外设时钟使能寄存器 (RCC_C2AHB3ENR)

RCC CPU2 AHB3 and AHB4 peripheral clock enable register

地址偏移：0x150

复位值：0x0208 0000

访问：无等待状态，按字、半字和字节访问

注： *当外设时钟未激活时，不支持 CPU2 对外设寄存器进行读写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	FLASHEN	Res.	Res.	Res.	Res.	IPCCEN	HSEMEN	RNGEN	AES2EN	PKAEN
						rw					rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:26 保留，必须保持复位值。

位 25 **FLASHEN**：CPU2 Flash 接口时钟使能 (CPU2 Flash interface clock enable)
只有在 Flash 掉电时，才能将此位清零。由软件置 1 和清零。
0：为 CPU2 禁止 Flash 接口时钟
1：为 CPU2 使能 Flash 接口时钟

位 24:21 保留，必须保持复位值。

位 20 **IPCCEN**：CPU2 IPCC 接口时钟使能 (CPU2 IPCC interface clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 IPCC 时钟
1：为 CPU2 使能 IPCC 时钟

位 19 **HSEMEN**：CPU2 HSEM 时钟使能 (CPU2 HSEM clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 HSEM 时钟
1：为 CPU2 使能 HSEM 时钟

位 18 **RNGEN**：CPU2 真随机数时钟使能 (CPU2 True RNG clocks enable)
由软件置 1 和清零。
0：为 CPU2 禁止真随机数总线和内核时钟
1：为 CPU2 使能真随机数总线和内核时钟

位 17 **AES2EN**：CPU2 AES2 加速器时钟使能 (CPU2 AES2 accelerator clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 AES2 时钟
1：为 CPU2 使能 AES2 时钟

位 16 **PKAEN**：CPU2 PKA 加速器时钟使能 (CPU2 PKA accelerator clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 PKA 时钟
1：为 CPU2 使能 PKA 时钟

位 15:0 保留，必须保持复位值。

8.4.38 RCC CPU2 APB1 外设时钟使能寄存器 1 (RCC_C2APB1ENR1)

RCC CPU2 APB1 peripheral clock enable register 1

地址：0x158

复位值：0x0000 0400

访问：无等待状态，按字、半字和字节访问

注： *当外设时钟未激活时，不支持 CPU2 对外设寄存器进行读写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM1 EN	Res.	Res.	Res.	Res.	USB EN	Res.	CRS EN	I2C3 EN	Res.	I2C1 EN	Res.	Res.	Res.	Res.	Res.
rw					rw		rw	rw		rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SPI2 EN	Res.	Res.	Res.	RTCAPB EN	LCD EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM2 EN
	rw				rw	rw									rw

位 31 **LPTIM1EN**：CPU2 低功耗定时器 1 时钟使能 (CPU2 Low power timer 1 clocks enable)
由软件置 1 和清零。
0：为 CPU2 禁止 LPTIM1 总线和内核时钟
1：为 CPU2 使能 LPTIM1 总线和内核时钟

位 30:27 保留，必须保持复位值。

位 26 **USBEN**：CPU2 USB FS 时钟使能 (CPU2 USB FS clocks enable)
由软件置 1 和清零
0：为 CPU2 禁止 USB FS 总线和内核时钟
1：为 CPU2 使能 USB FS 总线和内核时钟

位 25 保留，必须保持复位值。

位 24 **CRSEN**：CPU2 CRS 时钟使能 (CPU2 CRS clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 CRS 时钟
1：为 CPU2 使能 CRS 时钟

位 23 **I2C3EN**：CPU2 I2C3 时钟使能 (CPU2 I2C3 clocks enable)
由软件置 1 和清零。
0：为 CPU2 禁止 I2C3 总线和内核时钟
1：为 CPU2 使能 I2C3 总线和内核时钟

位 22 保留，必须保持复位值。

位 21 **I2C1EN**：CPU2 I2C1 时钟使能 (CPU2 I2C1 clocks enable)
由软件置 1 和清零。
0：为 CPU2 禁止 I2C1 总线和内核时钟
1：为 CPU2 使能 I2C1 总线和内核时钟

位 20:15 保留，必须保持复位值。

位 14 **SPI2EN**：CPU2 SPI2 时钟使能 (CPU2 SPI2 clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 SPI2 时钟
1：为 CPU2 使能 SPI2 时钟

位 13:11 保留，必须保持复位值。

位 10 **RTCAPBEN**：CPU2 RTC APB 时钟使能 (CPU2 RTC APB clock enable)
由软件置 1 和清零
0：为 CPU2 禁止 RTC APB 时钟
1：为 CPU2 使能 RTC APB 时钟

位 9 **LCDEN**：CPU2 LCD 时钟使能 (CPU2 LCD clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 LCD 时钟
1：为 CPU2 使能 LCD 时钟

位 8:1 保留，必须保持复位值。

位 0 **TIM2EN**：CPU2 TIM2 定时器时钟使能 (CPU2 TIM2 timer clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 TIM2 时钟
1：为 CPU2 使能 TIM2 时钟

8.4.39 RCC CPU2 APB1 外设时钟使能寄存器 2 (RCC_C2APB1ENR2)

RCC CPU2 APB1 peripheral clock enable register 2

偏移地址：0x15C

复位值：0x00000 0000

访问：无等待状态，按字、半字和字节访问

注： *当外设时钟未激活时，不支持 CPU2 对外设寄存器进行读写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM2EN	Res.	Res.	Res.	Res.	LPUART1EN
															rw

位 31:6 保留，必须保持复位值。

位 5 **LPTIM2EN**：CPU2 低功耗定时器 2 时钟使能 (CPU2 Low power timer 2 clocks enable)
由软件置 1 和清零。
0：为 CPU2 禁止 LPTIM2 总线和内核时钟
1：为 CPU2 使能 LPTIM2 总线和内核时钟

位 4:1 保留，必须保持复位值。

位 0 **LPUART1EN**：CPU2 低功耗 UART 1 时钟使能 (CPU2 Low power UART 1 clocks enable)
由软件置 1 和清零。
0：为 CPU2 禁止 LPUART1 总线和内核时钟
1：为 CPU2 使能 LPUART1 总线和内核时钟

8.4.40 RCC CPU2 APB2 外设时钟使能寄存器 (RCC_C2APB2ENR)

RCC CPU2 APB2 peripheral clock enable register

地址：0x160

复位值：0x0000 0000

访问：按字、半字和字节访问。

注： *当外设时钟未激活时，不支持 CPU2 对外设寄存器进行读写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI1 EN	Res.	Res.	TIM17 EN	TIM16 EN	Res.
										rw			rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1 EN	Res.	SPI1 EN	TIM1 EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw		rw	rw											

位 31:22 保留，必须保持复位值。

位 21 **SAI1EN**：CPU2 SAI1 时钟使能 (CPU2 SAI1 clocks enable)
由软件置 1 和清零。
0：为 CPU2 禁止 SAI1 总线和内核时钟
1：为 CPU2 使能 SAI1 总线和内核时钟

位 20:19 保留，必须保持复位值。

位 18 **TIM17EN**：CPU2 TIM17 定时器时钟使能 (CPU2 TIM17 timer clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 TIM17 定时器时钟
1：为 CPU2 使能 TIM17 定时器时钟

位 17 **TIM16EN**：CPU2 TIM16 定时器时钟使能 (CPU2 TIM16 timer clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 TIM16 定时器时钟
1：为 CPU2 使能 TIM16 定时器时钟

位 16:15 保留，必须保持复位值。

位 14 **USART1EN**：CPU2 USART1 时钟使能 (CPU1 USART1 clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 USART1 总线和内核时钟
1：为 CPU2 使能 USART1 总线和内核时钟

位 13 保留，必须保持复位值。

位 12 **SPI1EN**：CPU2 SPI1 时钟使能 (CPU2 SPI1 clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 SPI1 时钟
1：为 CPU2 使能 SPI1 时钟

位 11 **TIM1EN**：CPU2 TIM1 定时器时钟使能 (CPU2 TIM1 timer clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 TIM1 定时器时钟
1：为 CPU2 使能 TIM1 定时器时钟

位 10:0 保留，必须保持复位值。

8.4.41 RCC CPU2 APB3 外设时钟使能寄存器 (RCC_C2APB3ENR)

RCC CPU2 APB3 peripheral clock enable register

偏移地址：0x164

复位值：0x00000 0000

访问：无等待状态，按字、半字和字节访问

注：　当外设时钟未激活时，不支持 CPU2 对外设寄存器进行读写访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	802EN	BLEEN
														rw	rw

位 31:2　保留，必须保持复位值。

位 1　**802EN**：CPU2 802.15.4 接口时钟使能 (CPU2 802.15.4 interface clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 802.15.4 时钟
1：为 CPU2 使能 802.15.4 时钟

位 0　**BLEEN**：CPU2 BLE 接口时钟使能 (CPU2 BLE interface clock enable)
由软件置 1 和清零。
0：为 CPU2 禁止 BLE 时钟
1：为 CPU2 使能 BLE 时钟

8.4.42　睡眠模式下的 RCC CPU2 AHB1 外设时钟使能寄存器 (RCC_C2AHB1SMENR)

RCC CPU2 AHB1 peripheral clocks enable in Sleep modes register

偏移地址：0x168

复位值：0x0001 1007

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSC SMEN
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRC SMEN	Res.	Res.	SRAM1 SMEN	Res.	Res.	Res.	Res.	Res.	Res.	DMAMUX1 SMEN	DMA2 SMEN	DMA1 SMEN
			rw			rw							rw	rw	rw

位 31:17　保留，必须保持复位值。

位 16　**TSCSMEN**：CPU2 CSleep 模式期间的触摸感应控制器时钟使能 (Touch Sensing Controller clock enable during CPU2 CSleep mode)
由软件置 1 和清零。
0：CPU2 CSleep 和 CStop 模式期间 TSC 时钟由时钟门控禁止
1：CPU2 CSleep 模式期间 TSC 时钟由时钟门控使能，CPU2 CStop 模式期间 TSC 时钟由时钟门控禁止

位 15:13　保留，必须保持复位值。

位 12 **CRCSMEN**：CPU2 CSleep 模式期间的 CRC 时钟使能 (CRC clock enable during CPU2 CSleep mode)。

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 CRC 时钟由时钟门控禁止

1：CPU2 CSleep 模式期间 CRC 时钟由时钟门控使能，CPU2 CStop 模式期间 CRC 时钟由时钟门控禁止

位 11:10 保留，必须保持复位值。

位 9 **SRAM1SMEN**：CPU2 CSleep 模式期间的 SRAM1 接口时钟使能 (SRAM1 interface clock enabled during CPU2 CSleep mode)

由软件置 1 和清零

0：CPU2 CSleep 和 CStop 模式期间 SRAM1 接口时钟由时钟门控禁止

1：CPU2 CSleep 模式期间 SRAM1 接口时钟由时钟门控使能，CStop 模式期间 SRAM1 接口时钟由时钟门控禁止

位 8:3 保留，必须保持复位值。

位 2 **DMAMUX1SMEN**：CPU2 CSleep 模式期间的 DMAMUX1 时钟使能 (DMAMUX1 clock enable during CPU2 CSleep mode)

睡眠模式期间由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 DMAMUX1 时钟由时钟门控禁止

1：CPU2 CSleep 模式期间 DMAMUX1 时钟由时钟门控使能，CPU2 CStop 模式期间 DMAMUX1 时钟由时钟门控禁止

位 1 **DMA2SMEN**：CPU2 CSleep 模式期间的 DMA2 时钟使能 (DMA2 clock enable during CPU2 CSleep mode)

睡眠模式期间由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 DMA2 时钟由时钟门控禁止

1：CPU2 CSleep 模式期间 DMA2 时钟由时钟门控使能，CPU2 CStop 模式期间 DMA2 时钟由时钟门控禁止

位 0 **DMA1SMEN**：CPU2 CSleep 模式期间的 DMA1 时钟使能 (DMA1 clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 DMA1 时钟由时钟门控禁止

1：CPU2 CSleep 模式期间 DMA1 时钟由时钟门控使能，CPU2 CStop 模式期间 DMA1 时钟由时钟门控禁止

8.4.43 睡眠模式下的 RCC CPU2 AHB2 外设时钟使能寄存器 (RCC_C2AHB2SMENR)

RCC CPU2 AHB2 peripheral clocks enable in Sleep modes register

偏移地址：0x16C

复位值：0x0001 209F

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AES1 SMEN
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ADC SMEN	Res.	Res.	Res.	Res.	Res.	GPIOH SMEN	Res.	Res.	GPIOE SMEN	GPIOD SMEN	GPIOC SMEN	GPIOB SMEN	GPIOA SMEN
		rw						rw			rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **AES1SMEN**：CPU2 CSleep 模式期间的 AES1 加速器时钟使能 (AES1 accelerator clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 AES1 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 AES1 时钟由时钟门控使能，CPU2 CStop 模式期间 AES1 时钟由时钟门控禁止。

位 15:14 保留，必须保持复位值。

位 13 **ADCSMEN**：CPU2 CSleep 和 CStop 模式期间的 ADC 时钟使能 (ADC clock enable during CPU2 CSleep and CStop modes)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 ADC 总线时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 ADC 总线时钟由时钟门控使能，CPU2 CStop 模式期间 ADC 总线时钟由时钟门控禁止。

位 12:8 保留，必须保持复位值。

位 7 **GPIOHSMEN**：CPU2 CSleep 模式期间的 IO 端口 H 时钟使能 (IO port H clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 IO 端口 H 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 IO 端口 H 时钟由时钟门控使能，CPU2 CStop 模式期间 IO 端口 H 时钟由时钟门控禁止。

位 6:5 保留，必须保持复位值。

位 4 **GPIOESMEN**：CPU2 CSleep 模式期间的 IO 端口 E 时钟使能 (IO port E clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 IO 端口 E 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 IO 端口 E 时钟由时钟门控使能，CPU2 CStop 模式期间 IO 端口 E 时钟由时钟门控禁止。

位 3 **GPIODSMEN**：CPU2 CSleep 模式期间的 IO 端口 D 时钟使能 (IO port D clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 IO 端口 D 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 IO 端口 D 时钟由时钟门控使能，CPU2 CStop 模式期间 IO 端口 D 时钟由时钟门控禁止。

位 2 **GPIOCSMEN**：CPU2 CSleep 模式期间的 IO 端口 C 时钟使能 (IO port C clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 IO 端口 C 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 IO 端口 C 时钟由时钟门控使能，CPU2 CStop 模式期间 IO 端口 C 时钟由时钟门控禁止。

位 1 **GPIOBSMEN**：CPU2 CSleep 模式期间的 IO 端口 B 时钟使能 (IO port B clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 IO 端口 B 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 IO 端口 B 时钟由时钟门控使能，CPU2 CStop 模式期间 IO 端口 B 时钟由时钟门控禁止。

位 0 **GPIOASMEN**：CPU2 CSleep 模式期间的 IO 端口 A 时钟使能 (IO port A clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 IO 端口 A 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 IO 端口 A 时钟由时钟门控使能，CPU2 CStop 模式期间 IO 端口 A 时钟由时钟门控禁止。

8.4.44 睡眠模式下的 RCC CPU2 AHB3 和 AHB4 外设时钟使能寄存器 (RCC_C2AHB3SMENR)

RCC CPU2 AHB3 and AHB4 peripheral clocks enable in Sleep mode registe

偏移地址：0x170

复位值：0x0307 0000

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	FLASH SMEN	SRAM2 SMEN	Res.	Res.	Res.	Res.	Res.	RNG SMEN	AES2 SMEN	PKA SMEN
						rw	rw						rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:26 保留，必须保持复位值。

位 25 **FLASHSMEN**：CPU2 CSleep 模式期间的 Flash 接口时钟使能 (Flash interface clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 Flash 接口时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 Flash 接口时钟由时钟门控使能，CPU2 CStop 模式期间 Flash 接口时钟由时钟门控禁止。

位 24 **SRAM2SMEN**：CPU2 CSleep 模式期间的 SRAM2a 和 SRAM2b 存储器接口时钟使能 (SRAM2a and SRAM2b memory interface clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 SRAM2 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 SRAM2 时钟由时钟门控使能，CPU2 CStop 模式期间 SRAM2 时钟由时钟门控禁止。

位 23:19 保留，必须保持复位值。

位 18 **RNGSMEN**：CPU2 CSleep 和 CStop 模式期间的真 RNG 时钟使能 (True RNG clock enable during CPU2 CSleep and CStop mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间真 RNG 总线时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间真 RNG 总线时钟由时钟门控使能，CPU2 CStop 模式期间真 RNG 总线时钟由时钟门控禁止。

位 17 **AES2SMEN**：CPU2 CSleep 模式期间的 AES2 加速器时钟使能 (AES2 accelerator clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 AES2 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 AES2 时钟由时钟门控使能，CPU2 CStop 模式期间 AES2 时钟由时钟门控禁止。

位 16 **PKASMEN**：CPU2 CSleep 模式期间的 PKA 加速器时钟使能 (PKA accelerator clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 PKA 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 PKA 时钟由时钟门控使能，CPU2 CStop 模式期间 PKA 时钟由时钟门控禁止。

位 15:0 保留，必须保持复位值。

8.4.45 睡眠模式下的 RCC CPU2 APB1 外设时钟使能寄存器 1 (RCC_C2APB1SMENR1)

RCC CPU2 APB1 peripheral clocks enable in Sleep mode register 1

地址：0x178

复位值：0x85A0 4601

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM1 SMEN	Res.	Res.	Res.	Res.	USB SMEN	Res.	CRS SMEN	I2C3 SMEN	Res.	I2C1 SMEN	Res.	Res.	Res.	Res.	Res.
rw					rw		rw	rw		rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SPI2 SMEN	Res.	Res.	Res.	RTCAPB SMEN	LCD SMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM2 SMEN
	rw				rw	rw									rw

位 31 **LPTIM1SMEN**：CPU2 CSleep 和 CStop 模式期间的低功耗定时器 1 时钟使能 (Low power timer 1 clock enable during CPU2 CSleep and CStop mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 LPTIM1 总线时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 LPTIM1 总线时钟由时钟门控使能，CPU2 CStop 模式期间 LPTIM1 总线时钟由时钟门控禁止。

位 30:27 保留，必须保持复位值。

位 26 **USBSMEN**：CPU2 CSleep 和 CStop 模式期间的 USB FS 时钟使能 (USB FS clock enable during CPU2 CSleep and CStop modes)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 USB FS 总线时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 USB FS 总线时钟由时钟门控使能，CPU2 CStop 模式期间 USB FS 总线时钟由时钟门控禁止。

位 25 保留，必须保持复位值。

位 24 **CRSSMEN**：CPU2 CSleep 模式期间的 CRS 时钟使能 (CRS clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 CRS 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 CRS 时钟由时钟门控使能，CPU2 CStop 模式期间 CRS 时钟由时钟门控禁止。

位 23 **I2C3SMEN**：CPU2 CSleep 和 CStop 模式期间的 I2C3 时钟使能 (I2C3 clock enable during CPU2 CSleep and CStop modes)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 I2C3 总线时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 I2C3 总线时钟由时钟门控使能，CPU2 CStop 模式期间 I2C3 总线时钟由时钟门控禁止。

位 22 保留，必须保持复位值。

位 21 **I2C1SMEN**：CPU2 CSleep 和 CStop 模式期间的 I2C1 时钟使能 (I2C1 clock enable during CPU2 CSleep and CStop modes)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 I2C1 总线时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 I2C1 总线时钟由时钟门控使能，CPU2 CStop 模式期间 I2C1 总线时钟由时钟门控禁止。

位 20:15 保留，必须保持复位值。

位 14 **SPI2SMEN**：CPU2 CSleep 模式期间的 SPI2 时钟使能 (SPI2 clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 SPI2 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 SPI2 时钟由时钟门控使能，CPU2 CStop 模式期间 SPI2 时钟由时钟门控禁止。

位 13:11 保留，必须保持复位值。

位 10 **RTCAPBSMEN**：CPU2 CSleep 模式期间的 RTC 总线时钟使能 (RTC bus clock enable during CPU2 CSleep mode)

由软件置 1 和清零

0：CPU2 CSleep 和 CStop 模式期间 RTC 总线时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 RTC 总线时钟由时钟门控使能，CPU2 CStop 模式期间 RTC 总线时钟由时钟门控禁止。

位 9 **LCDSMEN**：CPU2 CSleep 模式期间的 LCD 时钟使能 (LCD clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 LCD 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 LCD 时钟由时钟门控使能，CPU2 CStop 模式期间 LCD 时钟由时钟门控禁止。

位 8:1 保留，必须保持复位值。

位 0 **TIM2SMEN**：CPU2 CSleep 模式期间的 TIM2 定时器时钟使能 (TIM2 timer clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 TIM2 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 TIM2 时钟由时钟门控使能，CPU2 CStop 模式期间 TIM2 时钟由时钟门控禁止。

8.4.46 睡眠模式下的 RCC CPU2 APB1 外设时钟使能寄存器 2 (RCC_C2APB1SMENR2)

RCC CPU2 APB1 peripheral clocks enable in Sleep mode register 2

偏移地址：0x17C

复位值：0x0000 0021

访问：无等待状态，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM2 SMEN	Res.	Res.	Res.	Res.	LPUART1 SMEN
										rw					rw

位 31:6 保留，必须保持复位值。

位 5 **LPTIM2SMEN**：CPU2 CSleep 和 CStop 模式期间的低功耗定时器 2 时钟使能 (Low power timer 2 clock enable during CPU2 CSleep and CStop modes)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 LPTIM2 总线时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 LPTIM2 总线时钟由时钟门控使能，CPU2 CStop 模式期间 LPTIM2 总线时钟由时钟门控禁止。

位 4:1 保留，必须保持复位值。

位 0 **LPUART1SMEN**：CPU2 CSleep 和 CStop 模式期间的低功耗 UART 1 时钟使能 (Low power UART 1 clock enable during CPU2 CSleep and CStop mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 LPUART1 总线时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 LPUART1 总线时钟由时钟门控使能，CPU2 CStop 模式期间 LPUART1 总线时钟由时钟门控禁止。

8.4.47 睡眠模式下的 RCC CPU2 APB2 外设时钟使能寄存器 (RCC_C2APB2SMENR)

RCC CPU2 APB2 peripheral clocks enable in Sleep mode register

地址：0x180

复位值：0x0026 5800

访问：按字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI1 SMEN	Res.	Res.	TIM17 SMEN	TIM16 SMEN	Res.
										rw			rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1 SMEN	Res.	SPI1 SMEN	TIM1SM EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw		rw	rw											

位 31:22 保留，必须保持复位值。

位 21 **SAI1SMEN**：CPU2 CSleep 和 CStop 模式期间的 SAI1 时钟使能 (SAI1 clock enable during CPU2 CSleep and CStop mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 SAI1 总线时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 SAI1 总线时钟由时钟门控使能，CPU2 CStop 模式期间 SAI1 总线时钟由时钟门控禁止。

位 20:19 保留，必须保持复位值。

位 18 **TIM17SMEN**：CPU2 CSleep 模式期间的 TIM17 定时器时钟使能 (TIM17 timer clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 TIM17 定时器时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 TIM17 定时器时钟由时钟门控使能，CPU2 CStop 模式期间 TIM17 定时器时钟由时钟门控禁止。

位 17 **TIM16SMEN**：CPU2 CSleep 模式期间的 TIM16 定时器时钟使能 (TIM16 timer clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 TIM16 定时器时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 TIM16 定时器时钟由时钟门控使能，CPU2 CStop 模式期间 TIM16 定时器时钟由时钟门控禁止。

位 16:15 保留，必须保持复位值。

位 14 **USART1SMEN**：CPU2 CSleep 和 CStop 模式期间的 USART1 时钟使能 (USART1 clock enable during CPU2 CSleep and CStop mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 USART1 总线时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 USART1 总线时钟由时钟门控使能，CPU2 CStop 模式期间 USART1 总线时钟由时钟门控禁止。

位 13 保留，必须保持复位值。

位 12 **SPI1SMEN**：CPU2 CSleep 模式期间的 SPI1 时钟使能 (SPI1 clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 SPI1 时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 SPI1 时钟由时钟门控使能，CPU2 CStop 模式期间 SPI1 时钟由时钟门控禁止。

位 11 **TIM1SMEN**：CPU2 CSleep 模式期间的 TIM1 定时器时钟使能 (TIM1 timer clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 TIM1 定时器时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 TIM1 定时器时钟由时钟门控使能，CPU2 CStop 模式期间 TIM1 定时器时钟由时钟门控禁止。

位 10:0 保留，必须保持复位值。

8.4.48 睡眠模式下的 RCC CPU2 APB3 外设时钟使能寄存器 (RCC_C2APB3SMENR)

RCC CPU2 APB3 peripheral clock enable in Sleep mode register

偏移地址：0x184

复位值：0x00000 0003

访问：按字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	802SMEN	BLESMEN
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **802SMEN**：CPU2 CSleep 模式期间的 802.15.4 接口总线时钟使能 (802.15.4 interface bus clock enable during CPU2 CSleep mode)

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 802.15.4 总线时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 802.15.4 总线时钟由时钟门控使能，CPU2 CStop 模式期间 802.15.4 总线时钟由时钟门控禁止。

位 0 **BLEEN**：CPU2 CSleep 模式期间的 BLE 接口总线时钟使能 (BLE interface bus clock enable during CPU2 CSleep mode)。

由软件置 1 和清零。

0：CPU2 CSleep 和 CStop 模式期间 BLE 总线时钟由时钟门控禁止。

1：CPU2 CSleep 模式期间 BLE 总线时钟由时钟门控使能，CPU2 CStop 模式期间 BLE 总线时钟由时钟门控禁止。

注：　　BLE 接口总线时钟也由 BLE IP 本身使能。

8.4.49 RCC 寄存器映射

下表列出了 RCC 寄存器映射和复位值。

表 41. RCC 寄存器映射和复位值

偏移	寄存器	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
0x000	RCC_CR	Res.	Res.	Res.	Res.	PLLSAI1RDY	PLLSAI1ON	PLLRDY	PLLON	Res.	Res.	Res.	HSEPRE	CSSON	HSEBYP	HSERDY	HSEON	Res.	Res.	Res.	HSIKERDY	HSIASFS	HSIRDY	HSIKERON	HSION	MSIRANGE[3:0]				Res.	MSIPLLEN	MSIRDY	MSION
	Reset value					0	0	0	0				0	0	0	0	0				0	0	0	0	0	0	1	1	0		0	0	1
0x004	RCC_ICSCR	Res	HSITRIM[6:0]							HSICAL[7:0]								MSITRIM[7:0]								MSICAL[7:0]							
	Reset value		1	0	0	0	0	0	0	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x
0x008	RCC_CFGR	Res.	MCOPRE[2:0]			MCOSEL[3:0]				Res.	Res.	Res.	Res.	Res.	PPRE2F	PPRE1F	HPREF	STOPWUCK	Res.	PPRE2[2:0]			PPRE1[2:0]			HPRE[3:0]				SWS[1:0]		SW[1:0]	
	Reset value		0	0	0	0	0	0	0						1	1	1	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	RCC_PLLCFGR	PLLR[2:0]			PLLREN	PLLQ[2:0]			PLLQEN	Res.	Res.	PLLP[4:0]					PLLPEN	Res.	PLLN[6:0]							Res.	PLLM[2:0]			Res.	Res.	PLLSRC[1:0]	
	Reset value	0	0	1	0	0	0	1	0			0	0	0	1	0	0		0	0	0	0	0	0	1		0	0	0			0	0
0x010	RCC_PLL SAI1CFGR	PLLR[2:0]			PLLREN	PLLQ[2:0]			PLLQEN	Res.	Res.	PLLP[4:0]					PLLPEN	Res.	PLLN[6:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	1	0	0	0	1	0			0	0	0	1	0	0		0	0	0	0	0	0	1								
0x014	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x018	RCC_CIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSI2RDYIE	HSI48RDYIE	LSECSSIE	Res.	Res.	PLLSAI1RDYIE	PLLRDYIE	HSERDYIE	HSIRDYIE	MSIRDYIE	LSERDYIE	LSI1RDYIE
	Reset value																					0	0	0			0	0	0	0	0	0	0
0x01C	RCC_CIFR	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSI2RDYF	HSI48RDYF	LSECSSF	CSSF	Res.	PLLSAI1RDYF.	PLLRDYF	HSERDYF	HSIRDYF	MSIRDYF	LSERDYF	LSI1RDYF
	Reset value																					0	0	0	0		0	0	0	0	0	0	0
0x020	RCC_CICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSI2RDYC	HSI48RDYC	LSECSSC	CSSC	Res.	PLLSAI1RDYC	PLLRDYC	HSERDYC	HSIRDYC	MSIRDYC	LSERDYC	LSI1RDYC
	Reset value																					0	0	0	0		0	0	0	0	0	0	0
0x024	RCC_SMPSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMPSSWS[1:0]		Res.	Res.	SMPSDIV[1:0]		Res.	Res.	SMPSSEL[1:0]	
	Reset value																							1	1			0	0			0	0

表 41. RCC 寄存器映射和复位值（续）

偏移	寄存器	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
0x028	RCC_AHB1RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSCRST	Res.	Res.	Res.	CRCRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAMUX1RST	DMA2RST	DMA1RST
	Reset value																0				0										0	0	0
0x02C	RCC_AHB2RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AES1RST	Res.	Res.	ADCRST	Res.	Res.	Res.	Res.	Res.	GPIOHRST	Res.	Res.	GPIOERST	GPIODRST	GPIOCRST	GPIOBRST	GPIOARST
	Reset value																0			0						0			0	0	0	0	0
0x030	RCC_AHB3RSTR	Res.	Res.	Res.	Res.	Res.	Res.	FLASHRST	Res.	Res.	Res.	Res.	IPCCRST	HSEMRST	RNGRST	AES2RST	PKARST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	QUADSPIRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value							0					0	0	0	0	0								0								
0x034	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x038	RCC_ APB1RSTR1	LPTIM1RST	Res.	Res.	Res.	Res.	USBRST	Res.	CRSRST	I2C3RST	Res.	I2C1RST	Res.	Res.	Res.	Res.	Res.	Res.	SPI2RST	Res.	Res.	Res.	Res.	Res.	LCDRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM2RST
	Reset value	0					0		0	0		0							0						0								0
0x03C	RCC_ APB1RSTR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM2RST	Res.	Res.	Res.	Res.	LPUART1RST
	Reset value																											0					0
0x040	RCC_ APB2RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI1RST	Res.	Res.	TIM17RST	TIM16RST	Res.	Res.	USART1RST	Res.	SPI1RST	TIM1RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value											0			0	0			0		0	0											
0x044	RCC_APB3RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFRST
	Reset value																																0
0x048	RCC_AHB1ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSCEN	Res.	Res.	Res.	CRCEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAMUX1EN	DMA2EN	DMA1EN
	Reset value																0				0										0	0	0
0x04C	RCC_AHB2ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AES1EN	Res.	Res.	ADCEN	Res.	Res.	Res.	Res.	Res.	GPIOHEN	Res.	Res.	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
	Reset value																0			0						0			0	0	0	0	0
0x050	RCC_AHB3ENR	Res.	Res.	Res.	Res.	Res.	Res.	FLASHEN	Res.	Res.	Res.	Res.	IPCCEN	HSEMEN	RNGEN	AES2EN	PKAEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	QUADSPIEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value							1					0	1	0	0	0								0								

表 41. RCC 寄存器映射和复位值（续）

偏移	寄存器	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
0x054	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x058	RCC_APB1ENR1	LPTIM1EN					USBEN		CRSEN	I2C3EN		I2C1EN							SPI2EN			WWDGEN	RTCAPBEN	LCDEN									TIM2EN
	Reset value	0					0		0	0		0							0			0	1	0									0
0x05C	RCC_APB1ENR2																											LPTIM2EN					LPUART1EN
	Reset value																											0					0
0x060	RCC_APB2ENR											SAI1EN			TIM17EN	TIM16EN			USART1EN		SPI1EN	TIM1EN											
	Reset value											0			0	0			0		0	0											
0x064	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x068	RCC_ AHB1SMENR																TSCSMEN				CRCSMEN			SRAM1SMEN						DMAMUX1SMEN	DMA2SMEN	DMA1SMEN
	Reset value																1				1			1						1	1	1
0x06C	RCC_ AHB2SMENR																AES1SMEN			ADCFSSMEN						GPIOHSMEN		GPIOESMEN	GPIODSMEN	GPIOCSMEN	GPIOBSMEN	GPIOASMEN
	Reset value																1			1						1		1	1	1	1	1
0x070	RCC _AHB3SMENR							FLASHSMEN	SRAM2SMEN						RNGSMEN	AES2SMEN	PKASMEN								QUADSPISMEN							
	Reset value							1	1						1	1	1								1								
0x074	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x078	RCC _APB1SMENR1	LPTIM1SMEN					USBSMEN		CRSSMEN	I2C3SMEN		I2C1SMEN							SPI2SMEN			WWDGSMEN	RTCAPBSMEN	LCDSMEN									TIM2SMEN
	Reset value	1					1		1	1		1							1			1	1	1									1
0x07C	RCC _APB1SMENR2																											LPTIM2SMEN					LPUART1SMEN
	Reset value																											1					1

表 41. RCC 寄存器映射和复位值（续）

偏移	寄存器	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
0x080	RCC_ APB2SMENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI1SMEN	Res.	Res.	TIM17SMEN	TIM16SMEN	Res.	Res.	USART1SMEN	Res.	SPI1SMEN	TIM1SMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value											1			1	1			1		1	1											
0x084	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x088	RCC_CCIPR	RNGSEL [1:0]		ADCSEL [1:0]		CLK48SEL [1:0]		Res.	Res.	SAI1SEL		LPTIM2SEL		LPTIM1SEL		I2C3SEL		Res.	Res.	I2C1SEL		LPUART1SEL		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USART1SEL	
	Reset value	0	0	0	0	0	0			0	0	0	0	0	0	0	0			0	0	0	0									0	0
0x08C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x090	RCC_BDCR	Res.	Res.	Res.	Res.	Res.	Res.	LSCOSEL	LSCOEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BDRST	RTCEN	Res.	Res.	Res.	Res.	Res.	RTC SEL [1:0]		Res.	LSECSSD	LSECSSON	LSE DRV [1:0]		LSEBYP	LSERDY	LSEON
	Reset value							0	0								0	0						0	0		0	0	0	0	0	0	0
0x094	RCC_CSR	LPWRRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	BORRSTF	PINRSTF	OBLRSTF	Res.	RMVF	Res.	Res.	Res.	Res.	Res.	Res.	RFRSTS	RFWKPSEL [1:0]		Res.	Res.	LSI2TRIM [3:0]				Res.	Res.	Res.	Res.	LSI2RDY	LSI2ON	LSI1RDY	LSI1ON
	Reset value	0	0	0	0	1	1	0		0							0	0	0			0	0	0	0					0	0	0	0
0x098	RCC_CRRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSI48CAL[8:0]									Res.	Res.	Res.	Res.	Res.	Res.	HSI48RDY	HSI48ON
	Reset value																	x	x	x	x	x	x	x	x	x						0	0
0x09C	RCC_HSECR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSETUNE[5:0]						Res.	HSEGMC [2:0]			HSES	Res.	Res.	UNLOCKED
	Reset value																			0	0	0	0	0	0	0	0	1	1	0	0	0	0
0x0A0 to 0x104	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x108	RCC_EXTCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFCSS	Res.	Res.	C2HPREF	SHDHPREF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	C2HPRE[3:0]				SHDHPRE [3:0]			
	Reset value												0			1	1									0	0	0	0	0	0	0	0
0x10C to 0x144	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x148	RCC _C2AHB1ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSCEN	Res.	Res.	Res.	CRCEN	Res.	Res.	SRAM1EN	Res.	Res.	Res.	Res.	Res.	Res.	DMAMUX1EN	DMA2EN	DMA1EN
	Reset value																0				0			0							0	0	0

表 41. RCC 寄存器映射和复位值（续）

偏移	寄存器	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
0x14C	RCC _C2AHB2ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AES1EN	Res.	Res.	ADCEN	Res.	Res.	Res.	Res.	Res.	GPIOHEN	Res.	Res.	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
	Reset value																0			0						0			0	0	0	0	0
0x150	RCC _C2AHB3ENR	Res.	Res.	Res.	Res.	Res.	Res.	FLASHEN	Res.	Res.	Res.	Res.	IPCCEN	HSEMEN	RNGEN	AES2EN	PKAEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value							1					0	1	0	0	0																
0x154	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x158	RCC _C2APB1ENR1	LPTIM1EN	Res.	Res.	Res.	Res.	USBEN	Res.	CRSEN	I2C3EN	Res.	I2C1EN	Res.	Res.	Res.	Res.	Res.	Res.	SPI2EN	Res.	Res.	Res.	RTCAPBEN	LCDEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM2EN
	Reset value	0					0		0	0		0							0				1	0									0
0x15C	RCC _C2APB1ENR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM2EN	Res.	Res.	Res.	Res.	LPUART1EN
	Reset value																											0					0
0x160	RCC _C2APB2ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI1EN	Res.	Res.	TIM17EN	TIM16EN	Res.	Res.	USART1EN	Res.	SPI1EN	TIM1EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value											0			0	0			0		0	0											
0x164	RCC _C2APB3ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	802EN	BLEEN
	Reset value																															0	0
0x168	RCC _C2AHB1SMENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSCSMEN	Res.	Res.	Res.	CRCSMEN	Res.	Res.	SRAM1SMEN	Res.	Res.	Res.	Res.	Res.	Res.	DMAMUX1SMEN	DMA2SMEN	DMA1SMEN
	Reset value																1				1			0							1	1	1
0x16C	RCC _C2AHB2SMENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AES1SMEN	Res.	Res.	ADCFSSMEN	Res.	Res.	Res.	Res.	Res.	GPIOHSMEN	Res.	Res.	GPIOESMEN	GPIODSMEN	GPIOCSMEN	GPIOBSMEN	GPIOASMEN
	Reset value																1			1						1			1	1	1	1	1
0x170	RCC _C2AHB3SMENR	Res.	Res.	Res.	Res.	Res.	Res.	FLASHSMEN	SRAM2SMEN	Res.	Res.	Res.	Res.	Res.	RNGSMEN	AES2SMEN	PKASMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value							1	1						1	1	1																
0x174	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 41. RCC 寄存器映射和复位值（续）

偏移	寄存器	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
0x178	RCC_C2 APB1SMENR1	LPTIM1SMEN	Res.	Res.	Res.	Res.	USBSMEN	Res.	CRSSMEN	I2C3SMEN	Res.	I2C1SMEN	Res.	Res.	Res.	Res.	Res.	Res.	SPI2SMEN	Res.	Res.	Res.	RTCAPBSMEN	LCDSMEN	Res.	Res.	Res.	Res.	Res.	Res	Res.	Res.	TIM2SMEN
	Reset value	1					1		1	1		1							1				1	1									1
0x17C	RCC_C2 APB1SMENR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM2SMEN	Res.	Res.	Res.	Res.	LPUART1SMEN
	Reset value																											1					1
0x180	RCC_C2 APB2SMENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI1SMEN	Res.	Res.	TIM17SMEN	TIM16SMEN	Res.	Res.	USART1SMEN	Res.	SPI1SMEN	TIM1SMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	Reset value											1			1	1			1		1	1											
0x184	RCC_C2 APB3SMENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	802SMEN	BLESMEN
	Reset value																															1	1
0x188 .. 03FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

9 通用 I/O (GPIO)

9.1 简介

每个通用 I/O 端口包括 4 个 32 位配置寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR 和 GPIOx_PUPDR）、2 个 32 位数据寄存器（GPIOx_IDR 和 GPIOx_ODR）和 1 个 32 位置 1/复位寄存器 (GPIOx_BSRR)。此外，所有 GPIO 都包括 1 个 32 位锁定寄存器 (GPIOx_LCKR) 和 2 个 32 位复用功能选择寄存器（GPIOx_AFRH 和 GPIOx_AFRL）。

9.2 GPIO 主要特性

- 输出状态：推挽或开漏 + 上拉/下拉
- 从输出数据寄存器 (GPIOx_ODR) 或外设（复用功能输出）输出数据
- 可为每个 I/O 选择不同的速度
- 输入状态：浮空、上拉/下拉、模拟
- 将数据输入到输入数据寄存器（GPIOx_IDR）或外设（复用功能输入）
- 置 1 和复位寄存器（GPIOx_ BSRR），对 GPIOx_ODR 具有按位写权限
- 锁定机制（GPIOx_LCKR），可冻结 I/O 端口配置
- 模拟功能
- 复用功能选择寄存器
- 快速翻转，每次翻转最快只需要两个时钟周期
- 引脚复用非常灵活，允许将 I/O 引脚用作 GPIO 或多种外设功能中的一种

9.3 GPIO 功能描述

根据数据手册中列出的每个 I/O 端口的特性，可通过软件将通用 I/O (GPIO) 端口的各个端口位分别配置为多种模式：

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟
- 具有上拉或下拉功能的开漏输出
- 具有上拉或下拉功能的推挽输出
- 具有上拉或下拉功能的复用功能推挽
- 具有上拉或下拉功能的复用功能开漏

每个 I/O 端口位均可自由编程，但 I/O 端口寄存器必须按 32 位字、半字或字节进行访问。GPIOx_BSRR 寄存器和 GPIOx_BRR 寄存器旨在实现对 GPIOx_ODR 寄存器进行原子读取/修改访问。这样便可确保在读取和修改访问之间发生中断请求也不会有问题。

图 20 和*图 21* 分别显示了标准和 5 V 容限 I/O 端口位的基本结构。*表 42* 给出了可能的端口位配置方案。

图 20. I/O 端口位的基本结构

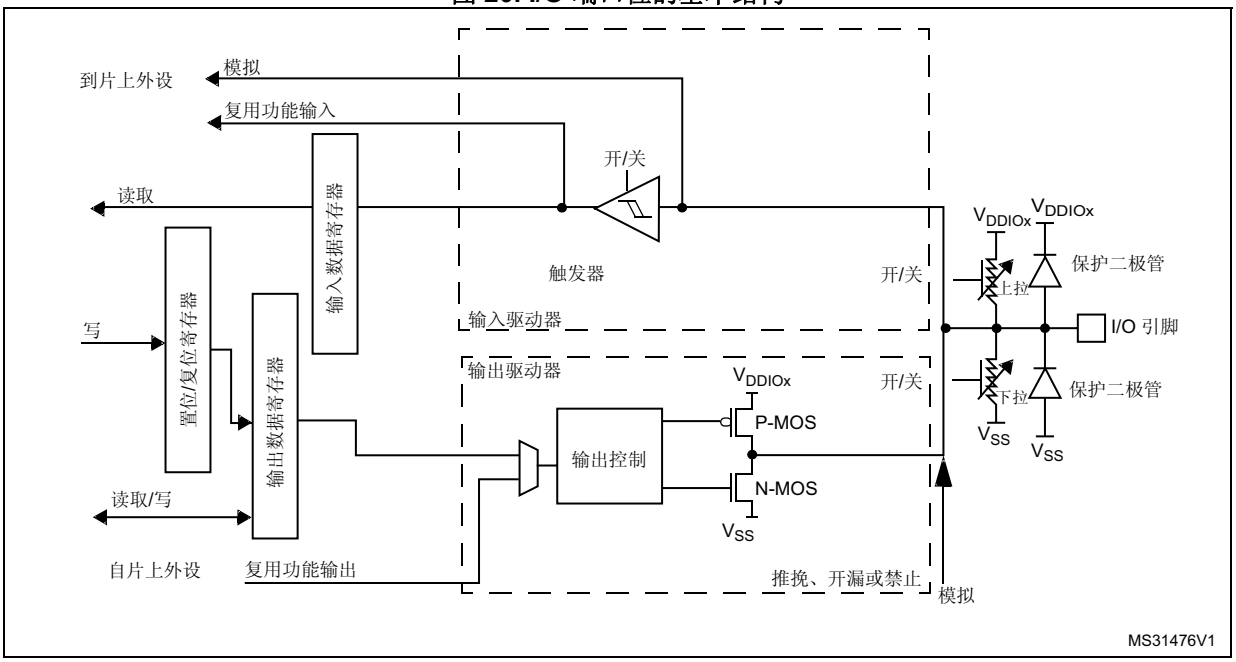
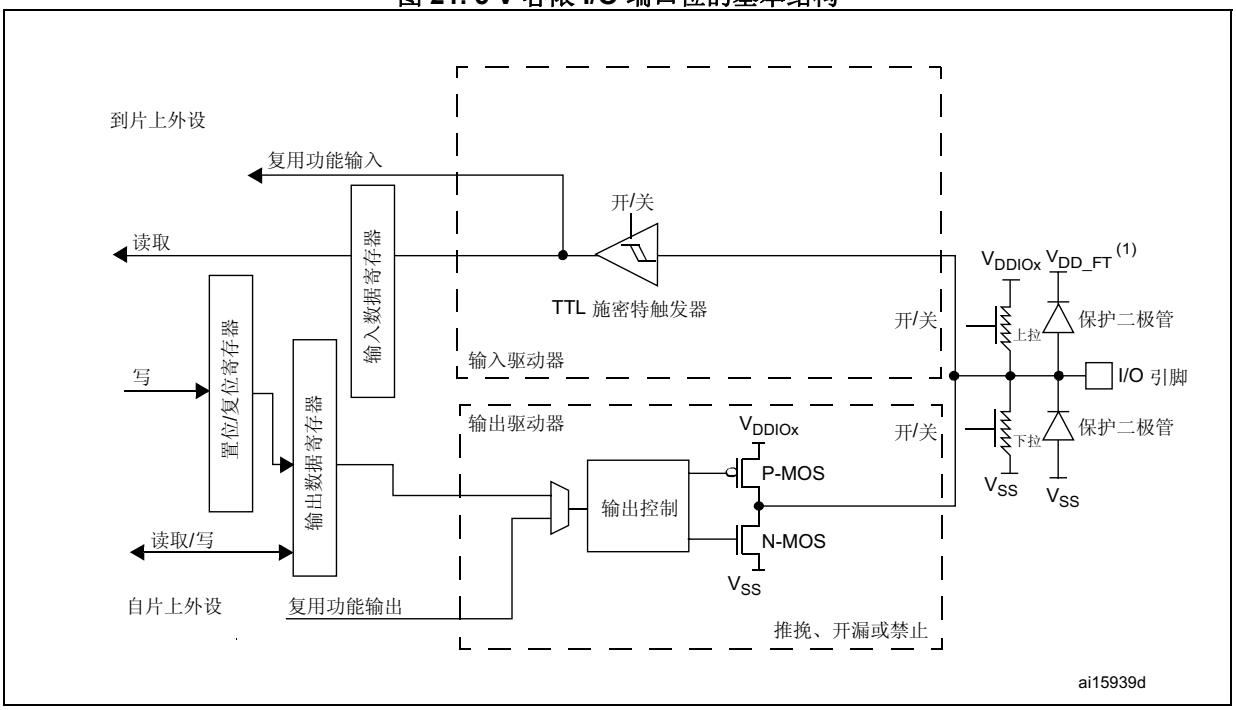


图 21. 5 V 容限 I/O 端口位的基本结构



1. V_{DD_FT} 是和 5 V 容限 I/O 相关的电位，与 V_{DD} 不同。

表 42. 端口位配置表(1)

MODE(i) [1:0]	OTYPER(i)	OSPEED(i) [1:0]		PUPD(i) [1:0]		I/O 配置	
01	0	SPEED [1:0]		0	0	GP 输出	PP
	0			0	1	GP 输出	PP + PU
	0			1	0	GP 输出	PP + PD
	0			1	1	保留	
	1			0	0	GP 输出	OD
	1			0	1	GP 输出	OD + PU
	1			1	0	GP 输出	OD + PD
	1			1	1	保留（GP 输出 OD）	
10	0	SPEED [1:0]		0	0	AF	PP
	0			0	1	AF	PP + PU
	0			1	0	AF	PP + PD
	0			1	1	保留	
	1			0	0	AF	OD
	1			0	1	AF	OD + PU
	1			1	0	AF	OD + PD
	1			1	1	保留	
00	x	x	x	0	0	输入	浮空
	x	x	x	0	1	输入	PU
	x	x	x	1	0	输入	PD
	x	x	x	1	1	保留（输入浮空）	
11	x	x	x	0	0	输入/输出	模拟
	x	x	x	0	1	保留	
	x	x	x	1	0		
	x	x	x	1	1		

1. GP = 通用、PP = 推挽、PU = 上拉、PD = 下拉、OD = 开漏、AF = 复用功能。

9.3.1 通用 I/O (GPIO)

在复位期间及复位刚刚完成后，复用功能尚未激活，大多数 I/O 端口被配置为模拟模式。

复位后，调试引脚处于复用功能上拉/下拉状态：

- PA15：JTDI 处于带上拉的输入模式
- PA14：JTCK/SWCLK 处于带下拉的输入模式
- PA13：JTMS/SWDAT 处于带上拉的输入模式
- PB4：NJTRST 处于带上拉的输入模式
- PB3：JTDO 处于无上拉/下拉的高阻态模式

PH3/BOOT0 在复位期间处于输入模式，并至少持续到选项字节加载阶段结束。请参见 *第 9.3.15 节：将 PH3 用作 GPIO*。

当引脚配置为输出后，写入到输出数据寄存器 (GPIOx_ODR) 的值将在 I/O 引脚上输出。可以在推挽模式下或开漏模式下使用输出驱动器（仅驱动低电平，高电平为高阻态）。

输入数据寄存器 (GPIOx_IDR) 每个 AHB 时钟周期捕获一次 I/O 引脚的数据。

所有 GPIO 引脚都具有内部弱上拉及下拉电阻，可根据 GPIOx_PUPDR 寄存器中的值来打开/关闭。

9.3.2 I/O 引脚复用功能复用器和映射

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

每个 I/O 引脚（PH3 除外）都有一个复用器，该复用器采用多达 16 路复用功能输入（AF0 到 AF15），可通过 GPIOx_AFRL（针对引脚 0 到 7）和 GPIOx_AFRH（针对引脚 8 到 15）寄存器对这些输入进行配置：

- 复位后，复用器选择为复用功能 0 (AF0)。在复用模式下通过 GPIOx_MODER 寄存器配置 I/O。
- 器件数据手册中详细说明了每个引脚的特定复用功能分配。

除了这种灵活的 I/O 复用架构之外，各外设还可以将复用功能映射到不同 I/O 引脚，这可以优化小型封装中可用外设的数量。

要在指定配置下使用 I/O，用户必须按照以下步骤操作：

- **调试功能：**每个器件复位后，立即将这些引脚分配为可由调试主机使用的复用功能引脚。
- **GPIO：**在 GPIOx_MODER 寄存器中将所需 I/O 配置为输出、输入或模拟通道。
- **外设复用功能：**
  - 在 GPIOx_AFRL 或 GPIOx_AFRH 寄存器中，将 I/O 连接到所需的 AFx。
  - 通过 GPIOx_OTYPER、GPIOx_PUPDR 和 GPIOx_OSPEEDER 寄存器，分别选择类型、上拉/下拉以及输出速度。
  - 在 GPIOx_MODER 寄存器中将所需 I/O 配置为复用功能。
- **其他功能：**
  - 对于 ADC 和 COMP，在 GPIOx_MODER 寄存器中将所需 I/O 配置为模拟模式，并在 ADC 和 COMP 寄存器中配置所需功能。
  - 对于 RTC、WKUPx 和振荡器等其他功能，在相关的 RTC、PWR 和 RCC 寄存器中配置所需功能。这些功能优先于标准 GPIO 寄存器中的配置。

有关复用功能 I/O 引脚映射的详细信息，请参见器件数据手册中的“复用功能映射”表。

9.3.3 I/O 端口控制寄存器

每个 GPIO 端口有 4 个 32 位存储器映射的控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR），可配置多达 16 个 I/O。GPIOx_MODER 寄存器用于选择 I/O 模式（输入、输出、AF、模拟）。GPIOx_OTYPER 和 GPIOx_OSPEEDR 寄存器用于选择输出类型（推挽或开漏）和速度。无论采用哪种 I/O 方向，GPIOx_PUPDR 寄存器都用于选择上拉/下拉。

9.3.4 I/O 端口数据寄存器

每个 GPIO 都具有 2 个 16 位数据寄存器：输入和输出数据寄存器（GPIOx_IDR 和 GPIOx_ODR）。GPIOx_ODR 用于存储待输出数据，可对其进行读/写访问。通过 I/O 输入的数据存储到输入数据寄存器 (GPIOx_IDR) 中，它是一个只读寄存器。

有关寄存器说明的详细信息，请参见*第 9.4.5 节：GPIO 端口输入数据寄存器 (GPIOx_IDR)（x = A 到 E 和 H）*和*第 9.4.6 节：GPIO 端口输出数据寄存器 (GPIOx_ODR)（x = A 到 E 和 H）*。

9.3.5 I/O 数据位操作

置 1 / 复位寄存器 (GPIOx_BSRR) 是一个 32 位寄存器，它允许应用程序在输出数据寄存器 (GPIOx_ODR) 中对各个单独的数据位执行置 1 和复位操作。置 1 / 复位寄存器的大小是 GPIOx_ODR 的二倍。

GPIOx_ODR 中的每个数据位对应于 GPIOx_BSRR 中的两个控制位：BS(i) 和 BR(i)。当写入 1 时，BS(i) 位会将对应的 ODR(i) 位**置 1**。当写入 1 时，BR(i) 位会**复位** ODR(i) 对应的位。

在 GPIOx_BSRR 中向任何位写入 0 都不会对 GPIOx_ODR 中的对应位产生任何影响。如果在 GPIOx_BSRR 中同时尝试对某个位执行置 1 和复位操作，则置 1 操作优先。

使用 GPIOx_BSRR 寄存器更改 GPIOx_ODR 中各个位的值是一个“单次”操作，不会锁定 GPIOx_ODR 位。随时都可以直接访问 GPIOx_ODR 位。GPIOx_BSRR 寄存器提供了一种执行原子按位处理的方法。

在对 GPIOx_ODR 进行位操作时，软件无需禁止中断：在一次原子 AHB 写访问中，可以修改一个或多个位。

9.3.6 GPIO 锁定机制

通过将特定的写序列应用到 GPIOx_LCKR 寄存器，可以冻结 GPIO 控制寄存器。冻结的寄存器包括 GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_AFRH。

要对 GPIOx_LCKR 寄存器执行写操作，必须应用特定的写/读序列。当正确的 LOCK 序列应用到此寄存器的第 16 位后，会使用 LCKR[15:0] 的值来锁定 I/O 的配置（在写序列期间，LCKR[15:0] 的值必须相同）。将 LOCK 序列应用到某个端口位后，在执行下一次 MCU 复位或外设复位之前，将无法对该端口位的值进行修改。每个 GPIOx_LCKR 位都会冻结控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_AFRH）中的对应位。

LOCK 序列（参见*第 9.4.8 节：GPIO 端口配置锁定寄存器 (GPIOx_LCKR)（x = A 到 E 和 H）*）只能通过对 GPIOx_LCKR 寄存器进行字（32 位长）访问的方式来执行，因为 GPIOx_LCKR 的第 16 位必须与 [15:0] 位同时置 1。

有关详细信息，请参见*第 9.4.8 节：GPIO 端口配置锁定寄存器 (GPIOx_LCKR)（x = A 到 E 和 H）*中的 LCKR 寄存器说明。

9.3.7 I/O 复用功能输入/输出

有两个寄存器可用来从每个 I/O 可用的复用功能输入/输出中进行选择。借助这些寄存器，用户可根据应用程序的要求将某个复用功能连接到其他某个引脚。

这意味着可使用 GPIOx_AFRL 和 GPIOx_AFRH 复用功能寄存器在每个 GPIO 上复用多个可用的外设功能。这样一来，应用程序可为每个 I/O 选择任何一个可用功能。由于 AF 选择信号由复用功能输入和复用功能输出共用，所以只需为指定 I/O 的复用功能输入/输出选择一个通道即可。

要了解在每个 GPIO 引脚上复用了哪些功能，请参见器件数据手册。

无复用功能映射到 PH3。

9.3.8 外部中断线/唤醒线

所有端口都具有外部中断功能。要使用外部中断线，必须将端口配置为输入模式。

请参见*第 14 节：扩展中断和事件控制器 (EXTI)* 和*第 14.3 节：EXTI 功能说明。*

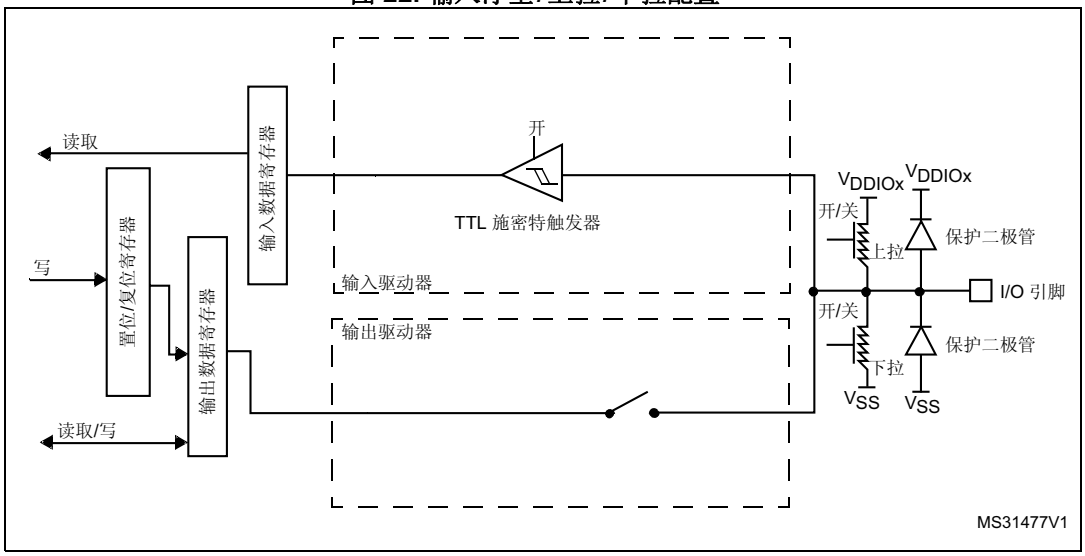
9.3.9 输入配置

对 I/O 端口进行编程作为输入时：

- 输出缓冲器被禁止
- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开上拉和下拉电阻
- 输入数据寄存器每隔 1 个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态

图 22 说明了 I/O 端口位的输入配置。

图 22. 输入浮空/上拉/下拉配置



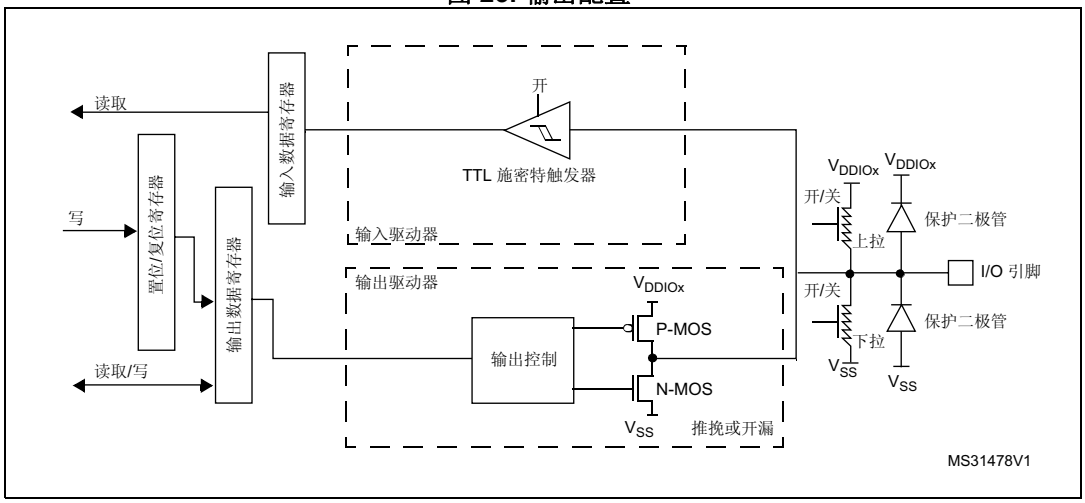
9.3.10 输出配置

对 I/O 端口进行编程作为输出时：

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

图 23 说明了 I/O 端口位的输出配置。

图 23. 输出配置



9.3.11 复用功能配置

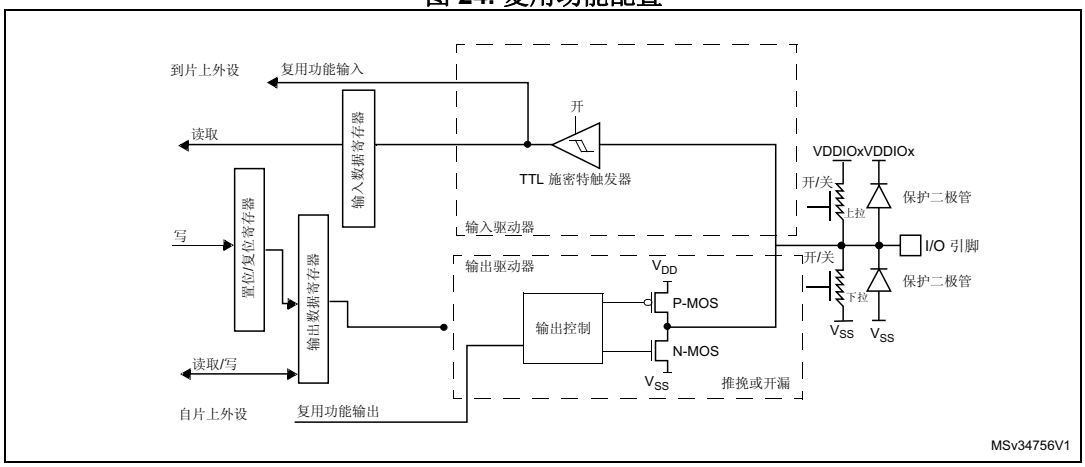
对 I/O 端口进行编程作为复用功能时：

- 可将输出缓冲器配置为开漏或推挽模式
- 输出缓冲器由来自外设的信号驱动（发送器使能和数据）
- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开弱上拉电阻和下拉电阻
- 输入数据寄存器每隔 1 个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态

注： *当所选复用功能为 LCD 功能时，不应用上述复用功能配置。在这种情况下，编程为复用功能输出的 I/O 按模拟配置中所述进行配置。*

图 24 说明了 I/O 端口位的复用功能配置。

图 24. 复用功能配置



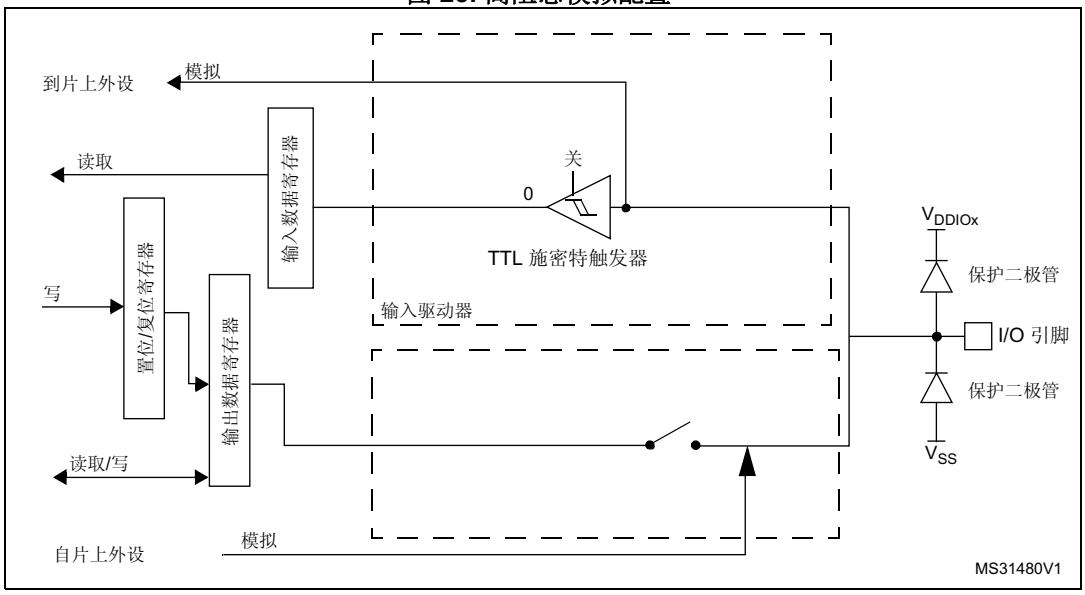
9.3.12 模拟配置

对 I/O 端口进行编程作为模拟配置时：

- 输出缓冲器被禁止。
- 施密特触发器输入停用，I/O 引脚的每个模拟输入的功耗变为零。施密特触发器的输出被强制处理为恒定值 (0)。
- 弱上拉和下拉电阻被硬件关闭。
- 对输入数据寄存器的读访问值为“0”。

图 25 说明了 I/O 端口位的高阻态模拟输入配置。

图 25. 高阻态模拟配置



9.3.13 将 LSE 振荡器引脚用作 GPIO

当 LSE 振荡器关闭（复位后的默认状态）时，可将相关的振荡器引脚用作常规 GPIO。

当 LSE 振荡器打开（通过将 RCC_CSR 寄存器中的 LSEON 位置 1）时，振荡器会控制与其相关联的引脚，这些引脚的 GPIO 配置不起作用。

将振荡器配置为用户外部时钟模式时，仅为时钟输入保留 OSC32_IN 引脚，OSC32_OUT 引脚仍可用作常规 GPIO。

注： *HSE OSC_IN 和 OSC_OUT 引脚是专用振荡器引脚，不能用作 GPIO。*

9.3.14 在 RTC 电源域中使用 GPIO 引脚

当内核电源域掉电时（器件进入待机模式时），PC13/PC14/PC15 GPIO 功能会丢失。在这种情况下，如果不通过 RTC 配置旁路其 GPIO 配置，这些引脚将被设置为模拟输入模式。

有关 RTC 的 I/O 控制的详细信息，请参见*第 29.3 节：RTC 功能说明*。

9.3.15 将 PH3 用作 GPIO

PH3 可用作自举引脚 (BOOT0) 或 GPIO。它可从输入模式切换到模拟输入模式，具体取决于用户选项字节中的 nSWBOOT0 位：

- 如果 nSWBOOT0 = 1，则在选项字节加载阶段之后切换。
- 如果 nSWBOOT0 = 0，则在复位之后切换。

9.4 GPIO 寄存器

本节对 GPIO 寄存器进行了详细介绍。

有关寄存器位、寄存器偏移地址和复位值的汇总，请参见*表 43*。

可按字、半字或字节模式写入外设寄存器。

9.4.1 GPIO 端口模式寄存器 (GPIOx_MODER)(x =A 到 E 和 H）

GPIO port mode register (GPIOx_MODER)

偏移地址：0x00

复位值：

- 0xABFF FFFF（端口 A）
- 0xFFFF FEBF（端口 B）
- 0xFFFF FFFF（端口 C 和 D）
- 0x0000 03FF（端口 E）
- 0x0000 00CF（端口 H）

保留端口 E[31:10]

保留端口 H[31:8 和 5:4]

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

位 31:0 **MODE[15:0][1:0]**：端口 x 配置 I/O 引脚 y (Port x configuration I/O pin y)（y = 15 到 0）
这些位通过软件写入，用于配置 I/O 模式。
00：输入模式
01：通用输出模式
10：复用功能模式
11：模拟模式（复位状态）

9.4.2 GPIO 端口输出类型寄存器 (GPIOx_OTYPER)（x = A 到 E 和 H）

GPIO port output type register

偏移地址：0x04

复位值：0x0000 0000

保留端口 E[31:5]

保留端口 H[31:4 和 2]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **OT[15:0]**：端口 x 配置 I/O 引脚 y (Port x configuration I/O pin y)（y = 15 到 0）
这些位通过软件写入，用于配置 I/O 输出类型。
0：推挽输出（复位状态）
1：开漏输出

9.4.3 GPIO 端口输出速度寄存器 (GPIOx_OSPEEDR)（x = A 到 E 和 H）

GPIO port output speed register

偏移地址：0x08

复位值：0x0C00 0000（端口 A）

复位值：0x0000 00C0（端口 B）

复位值：0x0000 0000（其它端口）

保留端口 E[31:10]

保留端口 H[31:8 和 5:4]

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

位 31:0 **OSPEED[15:0][1:0]**：端口 x 配置 I/O 引脚 y (Port x configuration I/O pin y)（y = 15 到 0）
这些位通过软件写入，用于配置 I/O 输出速度。
00：低速
01：中速
10：快速
11：高速
注：　关于每种速度的频率规范以及电源和负载条件，请参见器件数据手册。

9.4.4 GPIO 端口上拉/下拉寄存器 (GPIOx_PUPDR)（x = A 到 E 和 H）

GPIO port pull-up/pull-down register

偏移地址：0x0C

复位值：0x6400 0000（端口 A）

复位值：0x0000 0100（端口 B）

复位值：0x0000 0000（其它端口）

保留端口 E[31:10]

保留端口 H[31:8 和 5:4]

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

位 31:0 **PUPD[15:0][1:0]**：端口 x 配置 I/O 引脚 y (Port x configuration I/O pin y)（y = 15 到 0）
这些位通过软件写入，用于配置 I/O 上拉或下拉。
00：无上拉或下拉
01：上拉
10：下拉
11：保留

9.4.5 GPIO 端口输入数据寄存器 (GPIOx_IDR)（x = A 到 E 和 H）

GPIO port input data register

偏移地址：0x10

复位值：0x0000 XXXX

保留端口 E[31:5]

保留端口 H[31:4 和 2]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **ID[15:0]**：端口 x 输入数据 I/O 引脚 y (Port x input data I/O pin y)（y = 15 到 0）
这些位为只读。它们包含相应 I/O 端口的输入值。

9.4.6 GPIO 端口输出数据寄存器 (GPIOx_ODR)（x = A 到 E 和 H）

GPIO port output data register

偏移地址：0x14

复位值：0x0000 0000

保留端口 E[31:5]

保留端口 H[31:4 和 2]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **OD[15:0]**：端口输出数据 I/O 引脚 y (Port output data I/O pin y)（y = 15 到 0）
这些位可通过软件读取和写入。

注： 对于原子位置 1/ 复位，通过写入 GPIOx_BSRR 或 GPIOx_BRR 寄存器，可分别置 1 和/或复位 OD 位 (x = A..F)。

9.4.7 GPIO 端口位置 1/ 复位寄存器 (GPIOx_BSRR)（x = A 到 E 和 H）

GPIO port bit set/reset register

偏移地址：0x18

复位值：0x0000 0000

保留端口 E[31:21 和 15:5]

保留端口 H[31:20、18、15:4 和 2]

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

位 31:16 **BR[15:0]**：端口 x 复位 I/O 引脚 y (Port x reset I/O pin y)（y = 15 到 0）
这些位为只写。读取这些位可返回值 0x0000。
0：不会对相应的 ODx 位执行任何操作
1：复位相应的 ODx 位
注： 如果同时对 BSx 和 BRx 置 1，则 BSx 的优先级更高。

位 15:0 **BS[15:0]**：端口 x 位置 1 I/O 引脚 y (Port x set I/O pin y)（y = 15 到 0）
这些位为只写。读取这些位可返回值 0x0000。
0：不会对相应的 ODx 位执行任何操作
1：将相应的 ODx 位置 1

9.4.8 GPIO 端口配置锁定寄存器 (GPIOx_LCKR)（x = A 到 E 和 H）

GPIO port configuration lock register

当正确的写序列应用到第 16 位 (LCKK) 时，此寄存器将用于锁定端口位的配置。位 [15:0] 的值用于锁定 GPIO 的配置。在写序列期间，不能更改 LCKR[15:0] 的值。将 LOCK 序列应用到某个端口位后，在执行下一次 MCU 复位或外设复位之前，将无法对该端口位的值进行修改。

注： 可使用特定的写序列对 GPIOx_LCKR 寄存器执行写操作。在此锁定序列期间只允许使用字访问（32 位长）。

每个锁定位冻结一个特定的配置寄存器（控制寄存器和复用功能寄存器）。

偏移地址：0x1C

复位值：0x0000 0000

保留端口 E[31:17 和 15:5]

保留端口 H[31:17、15:4 和 2]

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

位 31:17 保留，必须保持复位值。

位 16 **LCKK**：锁定键 (Lock key)
可随时读取此位。可使用锁定键写序列对其进行修改。
0：端口配置锁定键未激活
1：端口配置锁定键已激活。GPIOx_LCKR 寄存器被锁定，直到下一次 MCU 复位或外设复位。
锁定键写序列：
WR LCKR[16] = ‘1’ + LCKR[15:0]
WR LCKR[16] = ‘0’ + LCKR[15:0]
WR LCKR[16] = ‘1’ + LCKR[15:0]
RD LCKR
RD LCKR[16] = ‘1’（此读操作为可选操作，但它可确认锁定已激活）

注： 在锁定键写序列期间，不能更改 LCK[15:0] 的值。
锁定序列中的任何错误都将中止锁定操作。
在任一端口位上的第一个锁定序列之后，对 LCKK 位的任何读访问都将返回“1”，直到下一次 MCU 复位或外设复位为止。

位 15:0 **LCK[15:0]**：端口 x 锁定 I/O 引脚 y (Port x lock I/O pin y)（y = 15 到 0）
这些位都是读/写位，但只能在 LCKK 位等于“0”时执行写操作。
0：端口配置未锁定
1：端口配置已锁定

9.4.9 GPIO 复用功能低位寄存器 (GPIOx_AFRL)（x = A 到 E 和 H）

GPIO alternate function low register

偏移地址：0x20

复位值：0x0000 0000

保留端口 E[31:20]

保留端口 H[31:16 和 11:8]

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

位 31:0 **AFSEL[7:0][3:0]**：端口 x I/O 引脚 y 的复用功能选择 (Alternate function selection for port x I/O pin y)（y = 7 到 0）

这些位通过软件写入，用于配置复用功能 I/O。

0000：AF0
0001：AF1
0010：AF2
0011：AF3
0100：AF4
0101：AF5
0110：AF6
0111：AF7
1000：AF8
1001：AF9
1010：AF10
1011：AF11
1100：AF12
1101：AF13
1110：AF14
1111：AF15

9.4.10 GPIO 复用功能高位寄存器 (GPIOx_AFRH)（x = A 到 E 和 H）

GPIO alternate function high register

偏移地址：0x24

复位值：0x0000 0000

保留端口 E[31:0]

保留端口 H[31:0]

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

位 31:0 **AFSEL[15:8][3:0]**：端口 x I/O 引脚 y 的复用功能选择 (Alternate function selection for port x I/O pin y)（y = 15 到 8）

这些位通过软件写入，用于配置复用功能 I/O。

0000：AF0
0001：AF1
0010：AF2
0011：AF3
0100：AF4
0101：AF5
0110：AF6
0111：AF7
1000：AF8
1001：AF9
1010：AF10
1011：AF11
1100：AF12
1101：AF13
1110：AF14
1111：AF15

9.4.11 GPIO 端口位复位寄存器 (GPIOx_BRR)（x = A 到 E 和 H）

GPIO port bit reset register

偏移地址：0x28

复位值：0x0000 0000

保留端口 E[31:5]

保留端口 H[31:4 和 2]

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

位 31:16 保留，必须保持复位值。

位 15:0 **BR[15:0]**：端口 x 复位 IO 引脚 y (Port x reset IO pin y)（y = 15 到 0）

这些位为只写。读取这些位可返回值 0x0000。

0：不会对相应的 ODx 位执行任何操作

1：复位相应的 ODx 位

9.4.12 GPIO 寄存器映射

下表提供了 GPIO 寄存器映射和复位值。

表 43. GPIO 寄存器映射和复位值

偏移	寄存器名称	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
0x00	**GPIOA_MODER**	MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	Reset value	1	0	1	0	1	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
0x00	**GPIOB_MODER**	MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	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	0	1	0	1	1	1	1	1	1
0x00	**GPIOx_MODER** (where x = **C, D**)	MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	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
0x00	**GPIOE_MODER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	1	1	1	1	1	1	1	1	1
0x00	**GPIOH_MODER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE3[1:0]		Res.	Res.	MODE1[1:0]		MODE0[1:0]	
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	1	-	-	1	1	1	1
0x04	**GPIOx_OTYPER** (where x = **A..D**)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**GPIOE_OTYPER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OT4	OT3	OT2	OT1	OT0
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0
0x04	**GPIOH_OTYPER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OT3	Res.	OT1	OT0
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	0	0
0x08	**GPIOA_OSPEEDR**	OSPEED15[1:0]		OSPEED14[1:0]		OSPEED13[1:0]		OSPEED12[1:0]		OSPEED11[1:0]		OSPEED10[1:0]		OSPEED9[1:0]		OSPEED8[1:0]		OSPEED7[1:0]		OSPEED6[1:0]		OSPEED5[1:0]		OSPEED4[1:0]		OSPEED3[1:0]		OSPEED2[1:0]		OSPEED1[1:0]		OSPEED0[1:0]	
	Reset value	0	0	0	0	1	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	0
0x08	**GPIOB_OSPEEDR**	OSPEED15[1:0]		OSPEED14[1:0]		OSPEED13[1:0]		OSPEED12[1:0]		OSPEED11[1:0]		OSPEED10[1:0]		OSPEED9[1:0]		OSPEED8[1:0]		OSPEED7[1:0]		OSPEED6[1:0]		OSPEED5[1:0]		OSPEED4[1:0]		OSPEED3[1:0]		OSPEED2[1:0]		OSPEED1[1:0]		OSPEED0[1:0]	
	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	1	1	0	0	0	0	0	0

表 43. GPIO 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x08	**GPIOx_OSPEEDR** (where x = **C, D**)	OSPEED15[1:0]		OSPEED14[1:0]		OSPEED13[1:0]		OSPEED12[1:0]		OSPEED11[1:0]		OSPEED10[1:0]		OSPEED9[1:0]		OSPEED8[1:0]		OSPEED7[1:0]		OSPEED6[1:0]		OSPEED5[1:0]		OSPEED4[1:0]		OSPEED3[1:0]		OSPEED2[1:0]		OSPEED1[1:0]		OSPEED0[1:0]	
	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
0x08	**GPIOE_OSPEEDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSPEED4[1:0]		OSPEED3[1:0]		OSPEED2[1:0]		OSPEED1[1:0]		OSPEED0[1:0]	
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0
0x08	**GPIOH_OSPEEDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSPEED3[1:0]		Res.	Res.	OSPEED1[1:0]		OSPEED0[1:0]	
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	-	-	0	0	0	0
0x0C	**GPIOA_PUPDR**	PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]		PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
	Reset value	0	1	1	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	0
0x0C	**GPIOB_PUPDR**	PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]		PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
	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	1	0	0	0	0	0	0	0	0
0x0C	**GPIOx_PUPDR** (where x = C, D)	PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]		PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
	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
0x0C	**GPIOE_PUPDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0
0x0C	**GPIOH_PUPDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PUPD3[1:0]		Res.	Res.	PUPD1[1:0]		PUPD0[1:0]	
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	-	-	0	0	0	0
0x10	**GPIOx_IDR** (where x = **A..D**)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
	Reset value																	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x10	**GPIOE_IDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		Res.	ID4	ID3	ID2	ID1	ID0
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	x	x	x	x
0x10	**GPIOH_IDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ID3	Res.	ID1	ID0
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	-	x	x
0x14	**GPIOx_ODR** (where x = **A..D**)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 43. GPIO 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x14	**GPIOE_ODR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OD4	OD3	OD2	OD1	OD0
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0
0x14	**GPIOH_ODR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OD3	Res.	OD1	OD0
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	0	0
0x18	**GPIOx_BSRR** (where x = **A..D**)	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0	BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
	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
0x18	**GPIOE_BSRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BR4	BR3	BR2	BR1	BR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BS4	BS3	BS2	BS1	BS0
	Reset value	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0
0x18	**GPIOH_BSRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BR3	Res.	BR1	BR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BS3	Res.	BS1	BS0
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	0	-	0	0	-	-	-	-	-	-	-	-	-	-	-	-	0	-	0	0
0x1C	**GPIOx_LCKR** (where x = **A..D**)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**GPIOE_LCKR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCK4	LCK3	LCK2	LCK1	LCK0
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0
0x1C	**GPIOH_LCKR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCK3	Res.	LCK1	LCK0
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	0	-	0	0
0x20	**GPIOx_AFRL** (where x = **A..D**)	AFSEL7[3:0]				AFSEL6[3:0]				AFSEL5[3:0]				AFSEL4[3:0]				AFSEL3[3:0]				AFSEL2[3:0]				AFSEL1[3:0]				AFSEL0[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
0x20	**GPIOE_AFRL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AFSEL4 [3:0]				AFSEL3 [3:0]				AFSEL2 [3:0]				AFSEL1 [3:0]				AFSEL0 [3:0]			
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**GPIOH_AFRL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AFSEL3 [3:0]				Res.	Res.	Res.	Res.	AFSEL1 [3:0]				AFSEL0 [3:0]			
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	-	-	-	-	0	0	0	0	0	0	0	0
0x24	**GPIOx_AFRH** (where x = **A..D**)	AFSEL15[3:0]				AFSEL14[3:0]				AFSEL13[3:0]				AFSEL12[3:0]				AFSEL11[3:0]				AFSEL10[3:0]				AFSEL9[3:0]				AFSEL8[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
0x28	**GPIOx_BRR** (where x = **A..D**)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**GPIOE_BRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BR4	BR3	BR2	BR1	BR0
	Reset value																												0	0	0	0	0
0x28	**GPIOH_BRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BR3	Res.	BR1	BR0
	Reset value																													0		0	0

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

10 系统配置控制器 (SYSCFG)

10.1 SYSCFG 主要特性

STM32WB55xx 器件配有一组配置寄存器。系统配置控制器的主要用途如下：

- 重新映射存储区域
- 管理 GPIO 的外部中断线连接
- 管理稳健性功能
- 设置 SRAM2 和 PKA RAM 写保护和软件擦除
- 配置 FPU 中断
- 在一些 I/O 和升压器上使能/禁用 I^2C 超快速模式驱动功能，用于 I/O 模拟开关
- 中断预屏蔽

10.2 SYSCFG 寄存器

10.2.1 SYSCFG 存储器重映射寄存器 (SYSCFG_MEMRMP)

SYSCFG memory remap register

此寄存器用于对存储器重映射进行配置。

偏移地址：0x000

复位值：0x0000 000X（X 是由 BOOT0 引脚和 BOOT1 选项位选择的存储器模式）

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

位 31:3 保留，必须保持复位值。

位 2:0 **MEM_MODE**：存储器映射选择 (Memory mapping selection)

这些位控制在地址 0x0000 0000 的存储器内部映射。它们用于通过软件选择物理重映射，从而旁路 BOOT 模式设置。复位后，这些位将使用 BOOT0 引脚（或选项位，具体取决于 nSWBOOT0 选项位）和 BOOT1 选项位选择的值。

000：将主 Flash 映射到 CPU1 0x00000000
001：将系统 Flash 映射到 CPU1 0x00000000
010：保留
011：将 SRAM1 映射到 CPU1 0x00000000
100：保留
101：保留
110：将 QUADSPI 存储器映射到 CPU1 0x00000000
111：保留

10.2.2 SYSCFG 配置寄存器 1 (SYSCFG_CFGR1)

SYSCFG configuration register 1

偏移地址：0x004

复位值：0x7C00 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FPU_IE[5..0]						Res	Res	Res	I2C3_FMP	Res	I2C1_FMP	I2C_PB9_FMP	I2C_PB8_FMP	I2C_PB7_FMP	I2C_PB6_FMP
rw	rw	rw	rw	rw	rw				rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	BOOSTEN	Res	Res	Res	Res	Res	Res	Res	Res
							rw								

位 31:26 **FPU_IE[5..0]**：CPU1 浮点单元中断使能位 (CPU1 Floating Point Unit interrupts enable bits)

FPU_IE[5]：不精确的中断使能

FPU_IE[4]：输入非常规中断使能

FPU_IE[3]：上溢中断使能

FPU_IE[2]：下溢中断使能

FPU_IE[1]：0 分频中断使能

FPU_IE[0]：无效操作中断使能

位 25:23 保留，必须保持复位值。

位 22 **I2C3_FMP**：I2C3 超快速模式驱动功能激活 (I2C3 Fast-mode Plus driving capability activation)

该位可在通过 AF 选择位选择的 I2C3 引脚上使能 Fm+ 驱动模式。

0：在通过 AF 选择位选择的 I2C3 引脚上未使能 Fm+ 模式。

1：在通过 AF 选择位选择的 I2C3 引脚上使能 Fm+ 模式。

位 21 保留，必须保持复位值。

位 20 **I2C1_FMP**：I2C1 超快速模式驱动功能激活 (I2C1 Fast-mode Plus driving capability activation)

该位可在通过 AF 选择位选择的 I2C1 引脚上使能 Fm+ 驱动模式。

0：在通过 AF 选择位选择的 I2C1 引脚上未使能 Fm+ 模式。

1：在通过 AF 选择位选择的 I2C1 引脚上使能 Fm+ 模式。

位 19 **I2C_PB9_FMP**：PB9 上的超快速模式 (Fm+) 驱动功能激活 (Fast-mode Plus (Fm+) driving capability activation on PB9)

该位可针对 PB9 使能 Fm+ 驱动模式。

0：PB9 引脚在标准模式下工作。

1：在 PB9 引脚上使能 Fm+ 模式并旁路速度控制。

位 18 **I2C_PB8_FMP**：PB8 上的超快速模式 (Fm+) 驱动功能激活 (Fast-mode Plus (Fm+) driving capability activation on PB8)

该位可针对 PB8 使能 Fm+ 驱动模式。

0：PB8 引脚在标准模式下工作。

1：在 PB8 引脚上使能 Fm+ 模式并旁路速度控制。

位 17 **I2C_PB7_FMP**：PB7 上的超快速模式 (Fm+) 驱动功能激活 (Fast-mode Plus (Fm+) driving capability activation on PB7)

该位可针对 PB7 使能 Fm+ 驱动模式。

0：PB7 引脚在标准模式下工作。

1：在 PB7 引脚上使能 Fm+ 模式并旁路速度控制。

位 16 **I2C_PB6_FMP**：PB6 上的超快速模式 (Fm+) 驱动功能激活 (Fast-mode Plus (Fm+) driving capability activation on PB6)

该位可针对 PB6 使能 Fm+ 驱动模式。

0：PB6 引脚在标准模式下工作。

1：在 PB6 引脚上使能 Fm+ 模式并旁路速度控制。

位 15:9 保留，必须保持复位值。

位 8 **BOOSTEN**：I/O 模拟开关升压器使能 (I/O analog switch voltage booster enable)

0：I/O 模拟开关由 V_{DDA} 电压供电。这是将 ADC 用于 V_{DDA} 高电压工作条件时的建议配置。

1：I/O 模拟开关由专用升压器供电（由 V_{DD} 供电）。这是将 ADC 用于 V_{DDA} 低电压工作条件时的建议配置。

位 7:0 保留，必须保持复位值。

10.2.3 SYSCFG 外部中断配置寄存器 1 (SYSCFG_EXTICR1)

SYSCFG external interrupt configuration register 1

偏移地址：0x008

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	EXTI3[2:0]			Res	EXTI2[2:0]			Res	EXTI1[2:0]			Res	EXTI0[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

位 31:15 保留，必须保持复位值。

位 14:12 **EXTI3[2:0]**：EXTI 3 配置位 (EXTI 3 configuration bits)

这些位通过软件写入，以选择 EXTI3 外部中断的源输入。

000：PA[3] 引脚

001：PB[3] 引脚

010：PC[3] 引脚

011：PD[3] 引脚

100：PE[3] 引脚

101：保留

110：保留

111：PH[3] 引脚

位 11 保留，必须保持复位值。

位 10:8 **EXTI2[2:0]**：EXTI 2 配置位 (EXTI 2 configuration bits)

这些位通过软件写入，以选择 EXTI2 外部中断的源输入。

000：PA[2] 引脚

001：PB[2] 引脚

010：PC[2] 引脚

011：PD[2] 引脚

100：PE[2] 引脚

101：保留

110：保留

111：保留

位 7 保留，必须保持复位值。

位 6:4 **EXTI1[2:0]**：EXTI 1 配置位 (EXTI 1 configuration bits)

这些位通过软件写入，以选择 EXTI1 外部中断的源输入。

000：PA[1] 引脚
001：PB[1] 引脚
010：PC[1] 引脚
011：PD[1] 引脚
100：PE[1] 引脚
101：保留
110：保留
111：PH[1] 引脚

位 3 保留，必须保持复位值。

位 2:0 **EXTI0[2:0]**：EXTI 0 配置位 (EXTI 0 configuration bits)

这些位通过软件写入，以选择 EXTI0 外部中断的源输入。

000：PA[0] 引脚
001：PB[0] 引脚
010：PC[0] 引脚
011：PD[0] 引脚
100：PE[0] 引脚
101：保留
110：保留
111：PH[0] 引脚

注： *该寄存器中提及的某些 I/O 引脚可能不适用于小型封装。*

10.2.4 SYSCFG 外部中断配置寄存器 2 (SYSCFG_EXTICR2)

SYSCFG external interrupt configuration register 2

偏移地址：0x00C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	EXTI7[2:0]			Res	EXTI6[2:0]			Res	EXTI5[2:0]			Res	EXTI4[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

位 31:15 保留，必须保持复位值。

位 14:12 **EXTI7[2:0]**：EXTI 7 配置位 (EXTI 7 configuration bits)
这些位通过软件写入，以选择 EXTI7 外部中断的源输入。
000：PA[7] 引脚
001：PB[7] 引脚
010：PC[7] 引脚
011：PD[7] 引脚
100：保留
101：保留
110：保留
111：保留

位 11 保留，必须保持复位值。

位 10:8 **EXTI6[2:0]**：EXTI 6 配置位 (EXTI 6 configuration bits)
这些位通过软件写入，以选择 EXTI6 外部中断的源输入。
000：PA[6] 引脚
001：PB[6] 引脚
010：PC[6] 引脚
011：PD[6] 引脚
100：保留
101：保留
110：保留
111：保留

位 7 保留，必须保持复位值。

位 6:4 **EXTI5[2:0]**：EXTI 5 配置位 (EXTI 5 configuration bits)
这些位通过软件写入，以选择 EXTI5 外部中断的源输入。
000：PA[5] 引脚
001：PB[5] 引脚
010：PC[5] 引脚
011：PD[5] 引脚
100：保留
101：保留
110：保留
111：保留

位 3 保留，必须保持复位值。

位 2:0 **EXTI4[2:0]**：EXTI 4 配置位 (EXTI 4 configuration bits)
这些位通过软件写入，以选择 EXTI4 外部中断的源输入。
000：PA[4] 引脚
001：PB[4] 引脚
010：PC[4] 引脚
011：PD[4] 引脚
100：PE[4] 引脚
101：保留
110：保留
111：保留

注：　该寄存器中提及的某些 I/O 引脚可能在小型封装上不可用。

10.2.5 SYSCFG 外部中断配置寄存器 3 (SYSCFG_EXTICR3)

SYSCFG external interrupt configuration register 3

偏移地址：0x010

复位值：0x0000 0000

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

位 31:15 保留，必须保持复位值。

位 14:12 **EXTI11[2:0]**：EXTI 11 配置位 (EXTI 11 configuration bits)
这些位通过软件写入，以选择 EXTI11 外部中断的源输入。
000：PA[11] 引脚
001：PB[11] 引脚
010：PC[11] 引脚
011：PD[11] 引脚
100：保留
101：保留
110：保留
111：保留

位 11 保留，必须保持复位值。

位 10:8 **EXTI10[2:0]**：EXTI 10 配置位 (EXTI 10 configuration bits)
这些位通过软件写入，以选择 EXTI10 外部中断的源输入。
000：PA[10] 引脚
001：PB[10] 引脚
010：PC[10] 引脚
011：PD[10] 引脚
100：保留
101：保留
110：保留
111：保留

位 7 保留，必须保持复位值。

位 6:4 **EXTI9[2:0]**：EXTI 9 配置位 (EXTI 9 configuration bits)
这些位通过软件写入，以选择 EXTI9 外部中断的源输入。
000：PA[9] 引脚
001：PB[9] 引脚
010：PC[9] 引脚
011：PD[9] 引脚
100：保留
101：保留
110：保留
111：保留

位 3 保留，必须保持复位值。

位 2:0 **EXTI8[2:0]**：EXTI 8 配置位 (EXTI 8 configuration bits)
这些位通过软件写入，以选择 EXTI8 外部中断的源输入。
000：PA[8] 引脚
001：PB[8] 引脚
010：PC[8] 引脚
011：PD[8] 引脚
100：保留
101：保留
110：保留
111：保留

注： *该寄存器中提及的某些 I/O 引脚可能不适用于小型封装。*

10.2.6 SYSCFG 外部中断配置寄存器 4 (SYSCFG_EXTICR4)

SYSCFG external interrupt configuration register 4

偏移地址：0x014

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	EXTI15[2:0]			Res	EXTI14[2:0]			Res	EXTI13[2:0]			Res	EXTI12[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

位 31:15 保留，必须保持复位值。

位 14:12 **EXTI15[2:0]**：EXTI15 配置位 (EXTI15 configuration bits)
这些位通过软件写入，以选择 EXTI15 外部中断的源输入。
000：PA[15] 引脚
001：PB[15] 引脚
010：PC[15] 引脚
011：PD[15] 引脚
100：保留
101：保留
110：保留
111：保留

位 11 保留，必须保持复位值。

位 10:8 **EXTI14[2:0]**：EXTI14 配置位 (EXTI14 configuration bits)
这些位通过软件写入，以选择 EXTI14 外部中断的源输入。
000：PA[14] 引脚
001：PB[14] 引脚
010：PC[14] 引脚
011：PD[14] 引脚
100：保留
101：保留
110：保留
111：保留

位 7 保留，必须保持复位值。

位 6:4 **EXTI13[2:0]**：EXTI13 配置位 (EXTI13 configuration bits)
这些位通过软件写入，以选择 EXTI13 外部中断的源输入。
000：PA[13] 引脚
001：PB[13] 引脚
010：PC[13] 引脚
011：PD[13] 引脚
100：保留
101：保留
110：保留
111：保留

位 3 保留，必须保持复位值。

位 2:0 **EXTI12[2:0]**：EXTI12 配置位 (EXTI12 configuration bits)
这些位通过软件写入，以选择 EXTI12 外部中断的源输入。
000：PA[12] 引脚
001：PB[12] 引脚
010：PC[12] 引脚
011：PD[12] 引脚
100：保留
101：保留
110：保留
111：保留

注： *该寄存器中提及的某些 I/O 引脚可能不适用于小型封装。*

10.2.7 SYSCFG SRAM2 控制和状态寄存器 (SYSCFG_SCSR)

SYSCFG SRAM2 control and status register

偏移地址：0x18

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C2RFD	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SRAM2 BSY	SRAM2 ER
														r	rw

位 31 **C2RFD**：CPU2 SRAM 取指（执行）禁止 (CPU2 SRAM fetch (execution) disable)。
该位可由软件置 1，可通过硬件复位实现复位，其中包括待机后复位。软件写 b0 不起作用。
0：使能 CPU2 从 SRAM1、SRAM2a 和 SRAM2b 取指，以允许 CPU2 从 SRAM 获取和执行代码
1：禁止 CPU2 从 SRAM1、SRAM2a 和 SRAM2b 取指，CPU2 从 SRAM 进行取指会触发总线错误。

位 30:2 保留，必须保持复位值

位 1 **SRAM2BSY**：SRAM2 和 PKA RAM 忙于擦除操作 (SRAM2 and PKA RAM busy by erase operation)

0：未在执行 SRAM2 或 PKA RAM 擦除操作。

1：正在执行 SRAM2 和/或 PKA RAM 擦除操作。

位 0 **SRAM2ER**：SRAM2 和 PKA RAM 擦除 (SRAM2 and PKA RAM Erase)

将此位置 1 可启动硬件 SRAM2 和 PKA RAM 擦除操作。SRAM2 和 PKA RAM 擦除操作结束后，此位自动清零。

注： 此位受写保护：只有在 SYSCFG_SKR 寄存器中写入正确的密钥序列后，才可将此位置 1。

10.2.8 SYSCFG 配置寄存器 2 (SYSCFG_CFGR2)

SYSCFG configuration register 2

偏移地址：0x01C

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	SPF	Res	Res	Res	Res	ECCL	PVDL	SPL	CLL
							rc_w1					rs	rs	rs	rs

位 31:9 保留，必须保持复位值

位 8 **SPF**：SRAM2 奇偶校验错误标志 (SRAM2 parity error flag)

检测到 SRAM2 奇偶校验错误时，该位由硬件置 1。通过软件写入“1”可将该位清零。

0：未检测到 SRAM2 奇偶校验错误。

1：检测到 SRAM2 奇偶校验错误。

位 7:4 保留，必须保持复位值

位 3 **ECCL**：ECC 锁定 (ECC Lock)

此位由软件置 1，只能通过系统复位清零。它可用于使能并锁定 TIM1/16/17 刹车输入的 Flash ECC 错误连接。

0：ECC 错误与 TIM1/16/17 刹车输入断开连接。

1：ECC 错误连接至 TIM1/16/17 刹车输入。

位 2 **PVDL**：PVD 锁定使能位 (PVD lock enable bit)

此位由软件置 1，只能通过系统复位清零。它可用于使能并锁定 TIM1/16/17 刹车输入的 PVD 连接以及 PWR_CR2 寄存器中的 PVDE 和 PLS[2:0]。

0：PVD 中断与 TIM1/16/17 刹车输入断开连接。PVDE 和 PLS[2:0] 位可由应用程序设定。

1：PVD 中断连接至 TIM1/16/17 刹车输入，PVDE 和 PLS[2:0] 位为只读位。

位 1 **SPL**：SRAM2 奇偶校验锁定位 (SRAM2 parity lock bit)

此位由软件置 1，只能通过系统复位清零。它可用于使能并锁定 TIM1/16/17 刹车输入的 SRAM2 奇偶校验错误信号连接。

0：SRAM2 奇偶校验错误信号与 TIM1/16/17 刹车输入断开连接。

1：SRAM2 奇偶校验错误信号连接至 TIM1/16/17 刹车输入。

位 0 **CLL**：CPU1 LOCKUP (Hardfault) 输出使能位 (CPU1 LOCKUP (Hardfault) output enable bit)

此位由软件置 1，只能通过系统复位清零。它可用于使能并锁定 TIM1/16/17 刹车输入的 CPU1 LOCKUP (Hardfault) 输出连接。

0：CPU1 LOCKUP 输出与 TIM1/16/17 刹车输入断开连接。

1：CPU1 LOCKUP 输出连接至 TIM1/16/17 刹车输入。

10.2.9 SYSCFG SRAM2 写保护寄存器 (SYSCFG_SWPR1)

SYSCFG SRAM2 write protection register

偏移地址：0x020

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P31WP	P30WP	P29WP	P28WP	P27WP	P26WP	P25WP	P24WP	P23WP	P22WP	P21WP	P20WP	P19WP	P18WP	P17WP	P16WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15WP	P14WP	P13WP	P12WP	P11WP	P10WP	P9WP	P8WP	P7WP	P6WP	P5WP	P4WP	P3WP	P2WP	P1WP	P0WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:0 **PxWP**（x = 0 到 31）：SRAM2 1 KB 页 x 写保护 (SRAM2 1Kbyte page x write protection)

这些位由软件置 1，只能通过系统复位清零。

0：禁止 SRAM2 1 KB 页 x 的写保护。

1：使能 SRAM2 1 KB 页 x 的写保护。

10.2.10 SYSCFG SRAM2 密钥寄存器 (SYSCFG_SKR)

SYSCFG SRAM2 key register

偏移地址：0x024

系统复位值：0x0000 0000

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

位 31:8 保留，必须保持复位值

位 7:0 **KEY[7:0]**：软件擦除的 SRAM2 写保护密钥 (SRAM2 write protection key for software erase)

要解锁 SYSCFG_CFGR2 寄存器中 SRAM2ER 位的写保护，需要执行以下步骤。

1. 向 Key[7:0] 写入“0xCA”
2. 向 Key[7:0] 写入“0x53”

写入一个错误的关键字会再次激活写保护。

10.2.11 SYSCFG SRAM2 写保护寄存器 2 (SYSCFG_SWPR2)

SYSCFG SRAM2 write protection register 2

偏移地址：0x028

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P63WP	P62WP	P61WP	P60WP	P59WP	P58WP	P57WP	P56WP	P55WP	P54WP	P53WP	P52WP	P51WP	P50WP	P49WP	P48WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P47WP	P46WP	P45WP	P44WP	P43WP	P42WP	P41WP	P40WP	P39WP	P38WP	P37WP	P36WP	P35WP	P34WP	P33WP	P32WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:0 **PxWP**（x = 32 到 63）：SRAM2 1 KB 页 x 写保护 (SRAM2 1Kbyte page x write protection)

这些位由软件置 1，只能通过系统复位清零。

0：禁止 SRAM2 1 KB 页 x 的写保护。

1：使能 SRAM2 1 KB 页 x 的写保护。

10.2.12 SYSCFG CPU1 中断屏蔽寄存器 1 (SYSCFG_IMR1)

SYSCFG CPU1 interrupt mask register 1

偏移地址：0x100

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI15IM	EXTI14IM	EXTI13IM	EXTI12IM	EXTI11IM	EXTI10IM	EXTI9IM	EXTI8IM	EXTI7IM	EXTI6IM	EXTI5IM	Res	Res	Res	Res	Res
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
TIM17IM	TIM16IM	TIM1IM	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
rw	rw	rw													

位 31:21 **xxxIM**：对 CPU1 的外设 xxx 中断屏蔽 (Peripheral xxx interrupt mask to CPU1)

0：将外设 xxx 中断转发至 CPU1

1：对 CPU1 进行外设 xxx 中断屏蔽

位 20:16 保留，必须保持复位值

位 15:13 **xxxIM**：对 CPU1 的外设 xxx 中断屏蔽 (Peripheral xxx interrupt mask to CPU1)

0：将外设 xxx 中断转发至 CPU1

1：对 CPU1 进行外设 xxx 中断屏蔽

位 12:0 保留，必须保持复位值

10.2.13 SYSCFG CPU1 中断屏蔽寄存器 2 (SYSCFG_IMR2)

SYSCFG CPU1 interrupt mask register 2

偏移地址：0x104

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PVDIM	Res	PVM3 IM	Res	PVM1 IM
											rw		rw		rw

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

位 31:21 保留，必须保持复位值

位 20 **xxxIM**：对 CPU1 的外设 xxx 中断屏蔽 (Peripheral xxx interrupt mask to CPU1)
0：将外设 xxx 中断转发至 CPU1
1：对 CPU1 进行外设 xxx 中断屏蔽。

位 19 保留，必须保持复位值

位 18 **xxxIM**：对 CPU1 的外设 xxx 中断屏蔽 (Peripheral xxx interrupt mask to CPU1)
0：将外设 xxx 中断转发至 CPU1
1：对 CPU1 进行外设 xxx 中断屏蔽。

位 17 保留，必须保持复位值

位 16 **xxxIM**：对 CPU1 的外设 xxx 中断屏蔽 (Peripheral xxx interrupt mask to CPU1)
0：将外设 xxx 中断转发至 CPU1
1：对 CPU1 进行外设 xxx 中断屏蔽。

位 15:0 保留，必须保持复位值

10.2.14 SYSCFG CPU2 中断屏蔽寄存器 1 (SYSCFG_C2IMR1)

SYSCFG CPU2 interrupt mask register 1

偏移地址：0x108

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI15 IM	EXTI14 IM	EXTI13 IM	EXTI12 IM	EXTI11 IM	EXTI10 IM	EXTI9 IM	EXTI8 IM	EXTI7 IM	EXTI6 IM	EXTI5 IM	EXTI4 IM	EXTI3 IM	EXTI2 IM	EXTI1 IM	EXTI0 IM
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
Res	Res	Res	ADCIM	COMP IM	AES1 IM	RNG IM	PKA IM	Res	FLASH IM	RCC IM	RTC ALARM IM	RTC WKUP IM	Res	Res	RTC STAMP TAMP LSECSS IM
			rw	rw	rw	rw	rw		rw	rw	rw	rw			rw

位 31:16 **xxxIM**：对 CPU2 的外设 xxx 中断屏蔽 (Peripheral xxx interrupt mask to CPU2)
0：将外设 xxx 中断转发至 CPU2
1：对 CPU2 进行外设 xxx 中断屏蔽

位 15:13 保留，必须保持复位值

位 12:8 **xxxIM**：对 CPU2 的外设 xxx 中断屏蔽 (Peripheral xxx interrupt mask to CPU2)
0：将外设 xxx 中断转发至 CPU2
1：对 CPU2 进行外设 xxx 中断屏蔽

位 7 保留，必须保持复位值

位 6:3 **xxxIM**：对 CPU2 的外设 xxx 中断屏蔽 (Peripheral xxx interrupt mask to CPU2)
0：将外设 xxx 中断转发至 CPU2
1：对 CPU2 进行外设 xxx 中断屏蔽

位 2:1 保留，必须保持复位值

位 0 **xxxIM**：对 CPU2 的外设 xxx 中断屏蔽 (Peripheral xxx interrupt mask to CPU2)
0：将外设 xxx 中断转发至 CPU2
1：对 CPU2 进行外设 xxx 中断屏蔽

10.2.15 SYSCFG CPU2 中断屏蔽寄存器 2 (SYSCFG_C2IMR2)

SYSCFG CPU2 interrupt mask register 2

偏移地址：0x10C

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	LCDIM	TSCIM	PVDIM	Res	PVM3 IM	Res	PVM1 IM
									rw	rw	rw		rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAMUX1 IM	DMA2 CH7 IM	DMA2 CH6 IM	DMA2 CH5 IM	DMA2 CH4 IM	DMA2 CH3 IM	DMA2 CH2 IM	DMA2 CH1 IM	Res	DMA1 CH7 IM	DMA1 CH6 IM	DMA1 CH5 IM	DMA1 CH4 IM	DMA1 CH3 IM	DMA1 CH2 IM	DMA1 CH1 IM
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值

位 22:20 **xxxIM**：对 CPU2 的外设 xxx 中断屏蔽 (Peripheral xxx interrupt mask to CPU2)
0：将外设 xxx 中断转发至 CPU2
1：对 CPU2 进行外设 xxx 中断屏蔽

位 19 保留，必须保持复位值

位 18 **xxxIM**：对 CPU2 的外设 xxx 中断屏蔽 (Peripheral xxx interrupt mask to CPU2)
0：将外设 xxx 中断转发至 CPU2
1：对 CPU2 进行外设 xxx 中断屏蔽

位 17 保留，必须保持复位值

位 16:8 **xxxIM**：对 CPU2 的外设 xxx 中断屏蔽 (Peripheral xxx interrupt mask to CPU2)
0：将外设 xxx 中断转发至 CPU2
1：对 CPU2 进行外设 xxx 中断屏蔽

位 7 保留，必须保持复位值

位 6:0 **xxxIM**：对 CPU2 的外设 xxx 中断屏蔽 (Peripheral xxx interrupt mask to CPU2)
0：将外设 xxx 中断转发至 CPU2
1：对 CPU2 进行外设 xxx 中断屏蔽

10.2.16 SYSCFG 安全 IP 控制寄存器 (SYSCFG_SIPCR)

SYSCFG secure IP control register

偏移地址：0x110

系统复位值：0x0000 0000

该寄存器提供写访问安全功能，仅允许 CPU2 进行写入操作。来自 CPU1 的写访问将被忽略，并会产生总线错误。任何读访问都会返回寄存器值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SRNG	SPKA	SAES2	SAES1
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值

位 3 **SRNG**：使能真 RNG 安全 (Enable True RNG security)。
0：禁止真 RNG 安全。（CPU1 和 CPU2 均可访问 RNG 寄存器）
1：使能真 RNG 安全。（RNG 寄存器只能由 CPU2 访问。CPU1 进行写访问会产生总线错误，读访问会返回零值。）

位 2 **SPKA**：使能 PKA 安全 (Enable PKA security)。
0：禁止 PKA 安全。（CPU1 和 CPU2 均可访问 PKA 寄存器）
1：使能 PKA 安全。PKA 寄存器只能由 CPU2 访问。CPU1 进行写访问会产生总线错误，读访问会返回零值。该位置 1 并发生系统复位（包括因从待机状态唤醒而导致的复位）时，PKA RAM 将被擦除。

位 1 **SAES2**：使能 AES2 安全 (Enable AES2 security)。
0：禁止 AES2 安全。（CPU1 和 CPU2 均可访问 AES2 寄存器）
1：使能 AES2 安全。（AES2 寄存器只能由 CPU2 访问。CPU1 进行写访问会产生总线错误，读访问会返回零值。）

位 0 **SAES1**：使能 AES1 KEY[7:0] 安全 (Enable AES1 KEY[7:0] security)。
0：禁止 AES1 KEY[7:0] 安全。（CPU1 和 CPU2 均可访问 AES1 KEY[7:0] 寄存器）
1：使能 AES1 KEY[7:0] 安全。（AES1 KEY[7:0] 寄存器只能由 CPU2 访问。CPU1 进行写访问会产生总线错误，读访问会返回零值。）

10.2.17 SYSCFG 寄存器映射

下表总结了 SYSCFG 寄存器映射和复位值。

表 44. SYSCFG 寄存器映射和复位值

偏移	寄存器	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
0x00	**SYSCFG_ MEMRMP**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM_MODE [2:0]		
	Reset value																														x	x	x
0x04	**SYSCFG_CFGR1**	FPU_IE[5..0]						Res.	Res.	Res.	I2C3_FMP	Res.	I2C1_FMP	I2C_PB9_FMP	I2C_PB8_FMP	I2C_PB7_FMP	I2C_PB6_FMP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOOSTEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	1	1	1	1	1				0		0	0	0	0	0								0								
0x08	**SYSCFG_EXTICR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI3 [2:0]			Res.	EXTI2 [2:0]			Res.	EXTI1 [2:0]			Res.	EXTI0 [2:0]		
	Reset value																		0	0	0		0	0	0		0	0	0		0	0	0
0x0C	**SYSCFG_EXTICR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI7 [2:0]			Res.	EXTI6 [2:0]			Res.	EXTI5 [2:0]			Res.	EXTI4 [2:0]		
	Reset value																		0	0	0		0	0	0		0	0	0		0	0	0
0x10	**SYSCFG_EXTICR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI11 [2:0]			Res.	EXTI10 [2:0]			Res.	EXTI9 [2:0]			Res.	EXTI8 [2:0]		
	Reset value																		0	0	0		0	0	0		0	0	0		0	0	0
0x14	**SYSCFG_EXTICR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI15 [2:0]			Res.	EXTI14 [2:0]			Res.	EXTI13 [2:0]			Res.	EXTI12 [2:0]		
	Reset value																		0	0	0		0	0	0		0	0	0		0	0	0
0x18	**SYSCFG_SCSR**	C2RFD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM2BSY	SRAM2ER
	Reset value	0																														0	0
0x1C	**SYSCFG_CFGR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPF	Res.	Res.	Res.	Res.	ECCL	PVDL	SPL	CLL
	Reset value																								0					0	0	0	0
0x20	**SYSCFG_SWPR**	P31WP	P30WP	P29WP	P28WP	P27WP	P26WP	P25WP	P24WP	P23WP	P22WP	P21WP	P20WP	P19WP	P18WP	P17WP	P16WP	P15WP	P14WP	P13WP	P12WP	P11WP	P10WP	P9WP	P8WP	P7WP	P6WP	P5WP	P4WP	P3WP	P2WP	P1WP	P0WP
	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
0x24	**SYSCFG_SKR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY							
	Reset value																									0	0	0	0	0	0	0	0
0x28	**SYSCFG_SWPR2**	P63WP	P62WP	P61WP	P60WP	P59WP	P58WP	P57WP	P56WP	P55WP	P54WP	P53WP	P52WP	P51WP	P50WP	P49WP	P48WP	P47WP	P46WP	P45WP	P44WP	P43WP	P42WP	P41WP	P40WP	P39WP	P38WP	P37WP	P36WP	P35WP	P34WP	P33WP	P32WP
	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

表 44. SYSCFG 寄存器映射和复位值（续）

偏移	寄存器	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
0x100	**SYSCFG_IMR1**	EXTI15IM	EXTI14IM	EXTI13IM	EXTI12IM	EXTI11IM	EXTI10IM	EXTI9IM	EXTI8IM	EXTI7IM	EXTI6IM	EXTI5IM	Res.	Res.	Res.	Res.	Res.	TIM17IM	TIM16IM	TIM1IM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0						0	0	0													
0x104	**SYSCFG_IMR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PVDIM	Res.	PVM3IM	Res.	PVM1IM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value												0		0		0																
0x108	**SYSCFG_C2IMR1**	EXTI15IM	EXTI14IM	EXTI13IM	EXTI12IM	EXTI11IM	EXTI10IM	EXTI9IM	EXTI8IM	EXTI7IM	EXTI6IM	EXTI5IM	EXTI4IM	EXTI3IM	EXTI2IM	EXTI1IM	EXTI0IM	Res.	Res.	Res.	ADCIM	COMPIM	AES1IM	RNGIM	PKAIM	Res.	FLASHIM	RCCIM	RTCALARMIM	RTCWKUPIM	Res.	Res.	RTCSTAMPTAMPLSECSSIM
	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
0x10C	**SYSCFG_C2IMR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCDIM	TSCIM	PVDIM	Res.	PVM3IM	Res.	PVM1IM	DMAMUX1IM	DMA2CH7IM	DMA2CH6IM	DMA2CH5IM	DMA2CH4IM	DMA2CH3IM	DMA2CH2IM	DMA2CH1IM	Res.	DMA1CH7IM	DMA1CH6IM	DMA1CH5IM	DMA1CH4IM	DMA1CH3IM	DMA1CH2IM	DMA1CH1IM
	Reset value										0	0	0		0		0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x110	**SYSCFG_SIPCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRNG	SPKA	SAES2	SAES1
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

11 直接存储器访问控制器 (DMA)

11.1 简介

直接存储器访问 (DMA) 控制器是一个主控总线系统外设。

当通过控制为 CPU 减轻负担时，DMA 用于在存储器映射的外设和/或存储器之间进行可编程的数据传输。

DMA 控制器采用单 AHB 主控总线架构。

DMA 有两个实例：DMA1 和 DMA2。

每个通道都专门管理来自一个或更多外设的存储器访问请求。每个 DMA 都有一个仲裁器，用于处理 DMA 请求间的优先级。

11.2 DMA 主要特性

- 单 AHB 主控总线
- 外设到存储器、存储器到外设、存储器到存储器以及外设到外设的数据传输
- 访问（作为源和目标）片上存储器映射的器件，例如 Flash、SRAM、AHB 和 APB 外设
- 所有 DMA 通道均可单独配置：
  - 每个通道均与来自外设的 DMA 请求信号或存储器到存储器传输中的软件触发信号相关联。由软件来完成配置。
  - 各请求之间的优先级可用软件编程（每通道 4 个级别：非常高、高、中、低），在软件优先级相同的情况下可以通过硬件决定优先级（例如，通道 1 请求的优先级高于通道 2 请求的优先级）。
  - 源和目标的传输大小（字节、半字、字）彼此独立，模拟打包和拆包。源和目标的地址必须根据传输数据的大小进行对齐。
  - 支持外设与存储器之间的双向传输，并支持循环缓冲区管理。
  - 可编程的待传输数据数目：0 到 2^{16} - 1
- 每个通道生成一个中断请求。每个中断请求均因以下三个 DMA 事件中的任意一个而引起：传输完成、半传输或传输错误。

11.3 DMA 实现

11.3.1 DMA1 和 DMA2

DMA1 和 DMA2 使用*表 45* 中所示的硬件配置参数实现。

表 45. DMA1 和 DMA2 实现

特性	DMA1	DMA2
通道数	7	7

11.3.2 DMA 请求映射

DMA 控制器通过 DMAMUX 外设连接至来自 AHB/APB 外设的 DMA 请求。

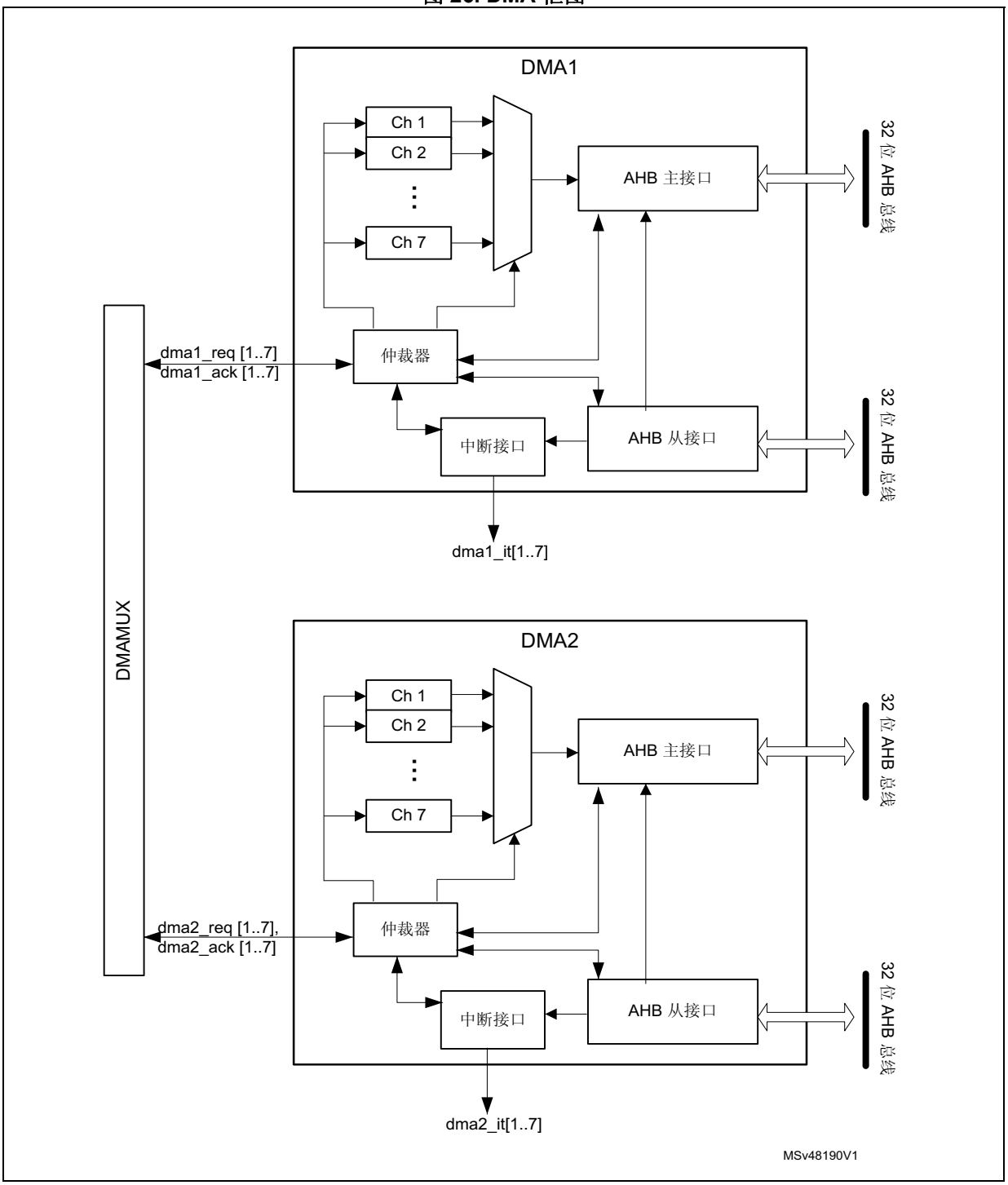
有关不同请求的映射，请参见*第 12.3 节：DMAMUX 实现*。

11.4 DMA 功能说明

11.4.1 DMA 框图

DMA 框图如*图 26* 所示。

图 26. DMA 框图



DMA 控制器通过与其他系统主设备共用 AHB 系统总线来执行直接存储器传输。总线矩阵执行循环调度。当 CPU 与 DMA 访问目标（存储器或外设）相同时，DMA 请求会将 CPU 对系统总线的访问停止数个总线周期。

根据通过 AHB 从接口实现的相应配置，DMA 控制器会在 DMA 通道及其接收的相关请求之间进行仲裁。DMA 控制器还会通过单一 AHB 端口主器件调度 DMA 数据传输。

DMA 控制器按通道向中断控制器生成中断。

11.4.2 DMA 引脚和内部信号

表 46. DMA 内部输入/输出信号

信号名称	信号类型	说明
dma_req[x]	输入	DMA 通道 x 请求
dma_ack[x]	输出	DMA 通道 x 应答
dma_it[x]	输出	DMA 通道 x 中断

11.4.3 DMA 传输

软件在通道级配置 DMA 控制器，以便执行块传输，此类传输由一系列 AHB 总线传输组成。

DMA 块传输可从外设请求，或在进行存储器到存储器传输时由软件触发。

触发该事件后，会按照以下步骤进行单次 DMA 传输：

1. 外设向 DMA 控制器发送单个 DMA 请求信号。
2. DMA 控制器按照与此外设请求相关的通道的优先级来处理该请求。
3. 只要 DMA 控制器授权给外设，DMA 控制器就会向外设发送确认信号。
4. 外设获得 DMA 控制器的确认信号后，便会立即释放其请求。
5. 一旦请求被外设取消，DMA 控制器就会释放确认信号。

外设可继续发送请求，再次启动 DMA 传输。

不论外设是传输源还是传输目标，都使用请求/应答协议。例如，对于存储器到外设传输，外设通过对 DMA 控制器驱动其单个请求信号来启动传输。DMA 控制器随后读取存储器中的单个数据，并将该数据写入外设。

对于给定通道 x，DMA 块传输由以下内容的重复序列组成：

- 单次 DMA 传输，其中封装单个数据的两次 AHB 传输（通过 DMA AHB 总线主控实现）：
  - 通过内部当前外设/存储器地址寄存器进行寻址，从外设数据寄存器或存储器单元中读取单个数据（字节、半字或字）。
    用于首次传输的起始地址是在 DMA_CPARx 或 DMA_CMARx 寄存器中编程的外设或存储器基址。
  - 通过内部当前外设/存储器地址寄存器进行寻址，向外设数据寄存器或存储器单元中写入单个数据（字节、半字或字）。
    用于首次传输的起始地址是在 DMA_CPARx 或 DMA_CMARx 寄存器中编程的外设或存储器基址。
- 编程的 DMA_CNDTRx 寄存器在传输后递减
  该寄存器包含待传输数据项的剩余数目（AHB“先读后写”传输次数）。

重复执行该序列，直至 DMA_CNDTRx 为空。

注： *AHB 主控总线源地址/目标地址必须根据传输至源/目标的单个数据的设定大小进行对齐。*

11.4.4 DMA 仲裁

DMA 仲裁器管理不同通道间的优先级。

当仲裁器为某个有效通道 x 授予优先权后（硬件请求或软件触发），会发起单次 DMA 传输（例如，单个数据的一次 AHB“先读后写”传输）。随后仲裁器会再次对有效通道组进行仲裁，并选择优先级最高的通道。

优先级管理分为两个阶段：

- 软件：每个通道的优先级均在 DMA_CCRx 寄存器中配置为以下四个不同级别之一：
  - 非常高
  - 高
  - 中
  - 低
- 硬件：如果两个请求的软件优先级相同，则索引编号最低的通道优先。例如，通道 2 的优先级高于通道 4。

当通道 x 被编程为在存储器到存储器模式下进行块传输时，在此通道 x 的各个单次 DMA 传输之间将进行重新仲裁。每当存在其他同时带有有效请求的通道时，DMA 仲裁器便会自动授权切换到其他有效请求的通道中具有最高优先级的通道，此通道的优先级可能低于存储器到存储器通道。

11.4.5 DMA 通道

各个通道均能处理外设寄存器（位于固定地址中）与存储器单元之间的 DMA 传输。待传输数据项的数目可编程。每个传输完成后，包含待传输的数据项数目的寄存器都会递减。

DMA 通道以块传输的方式进行编程设置。

数据大小可编程

要传输到外设和存储器的单个数据的大小（字节、半字或字）可分别通过 DMA_CCRx 寄存器的 PSIZE[1:0] 和 MSIZE[1:0] 位域进行编程。

指针递增

根据 DMA_CCRx 寄存器的 PINC 位和 MINC 位的状态，外设和存储器指针在每次传输后可自动递增。

如果使能了**递增模式**（PINC 或 MINC 置 1），则下次传输的地址是前一次传输的地址加上 1、2 或 4，具体取决于 PSIZE[1:0] 或 MSIZE[1:0] 中定义的数据大小。首次传输的地址可在 DMA_CPARx 或 DMA_CMARx 寄存器中进行编程。在传输过程中，这些寄存器将保持初始编程的值。软件无法获得当前传输的地址（位于当前内部外设或存储器地址寄存器中）。

如果将通道 x 配置为**非循环模式**，则在最后一次数据传输完成后（即待传输的单个数据数目达到零后），将不处理任何 DMA 请求。必须禁止 DMA 通道，这样才能将新的数据项数目重新加载到 DMA_CNDTRx 寄存器。

注： *如果禁止通道 x，DMA 寄存器不会被复位。DMA 通道寄存器（DMA_CCRx、DMA_CPARx 和 DMA_CMARx）仍保留在通道配置阶段设置的初始值。*

在**循环模式**下，最后一次数据传输完成后，DMA_CNDTRx 寄存器将自动重新加载初始编程值。当前的内部地址寄存器重新加载 DMA_CPARx 和 DMA_CMARx 寄存器中的基址值。

通道配置流程

配置 DMA 通道 x 时需按照以下步骤操作：

1. 在 DMA_CPARx 寄存器中设置外设寄存器地址。
   在外设事件发生后，或在存储器到存储器模式下使能通道后，会将数据从该地址移至存储器或从存储器移至该地址。
2. 设置 DMA_CMARx 寄存器中的存储器地址。
   在外设事件发生后，或在存储器到存储器模式下使能通道后，会将数据写入存储器或从存储器读取数据。
3. 在 DMA_CNDTRx 寄存器中配置待传输的总数据数。
   每次数据传输后，该值都会递减。
4. 在 DMA_CCRx 寄存器中配置下列参数：
   - 通道优先级
   - 数据传输方向
   - 循环模式
   - 外设和存储器递增模式
   - 外设和存储器数据大小
   - 传输完成一半和/或全部完成以及/或者出现传输错误时的中断使能
5. 将 DMA_CCRx 寄存器中的 EN 位置 1 以激活通道。

通道在使能后可处理来自此通道所连接外设的任意 DMA 请求，或者启动存储器到存储器块传输。

注： *通道配置流程的最后两步可合并为对 DMA_CCRx 寄存器进行单次访问来配置和使能通道。*

通道状态和禁止通道

处于活动状态的通道 x 为已使能通道（读取 DMA_CCRx.EN = 1）。活动通道 x 为必须由软件使能（DMA_CCRx.EN 设置为 1），且之后未发生传输错误 (DMA_ISR.TEIFx = 0) 的通道。如果存在传输错误，通道将由硬件自动禁止 (DMA_CCRx.EN = 0)。

可能会出现以下三种用例：

- 挂起和恢复通道
  这对应于以下两个操作：
  - 活动通道由软件禁止（写入 DMA_CCRx.EN = 0，而 DMA_CCRx.EN = 1）。
  - 软件再次使能通道（DMA_CCRx.EN 设置为 1），而无需重新配置其他通道寄存器（例如 DMA_CNDTRx、DMA_CPARx 和 DMA_CMARx）。

  这种情况不受 DMA 硬件支持，无法保证正确执行剩余数据传输。
- 停止和中止通道
  如果应用程序不再需要此活动通道，则可通过软件将其禁止。通道会停止并中止，但 DMA_CNDTRx 寄存器内容可能无法正确反映相对终止的源和目标缓冲区/寄存器、剩余数据传输的个数。

- 中止并重新启动通道

  这对应于软件序列：禁止活动通道，然后重新配置通道并再次将其使能。

  如果满足以下条件，则硬件支持此用例：

  – 应用程序可保证：在软件禁止通道时，相应的主设备端口的 DMA 传输已经完成。例如，应用程序可先在 DMA 模式下禁止外设，以确保没有来自该外设的待处理硬件 DMA 请求。

  – 软件必须对同一 DMA_CCRx 寄存器进行单独的写访问：首先禁止通道。然后，在需要进行配置更改的情况下，重新配置通道以进行下一次块传输，其中包括 DMA_CCRx。DMA_CCRx.EN=1 时，存在只读 DMA_CCRx 寄存器字段。最后，再次使能通道。

发生通道传输错误时，DMA_CCRx 寄存器的 EN 位由硬件清零。在 DMA_ISR 寄存器的 TEIFx 位置 1 之前，不能通过软件将该 EN 位置 1 来重新激活通道 x。

循环模式（在存储器到外设/外设到存储器传输中）

循环模式可用于处理循环缓冲区和连续数据流（例如 ADC 扫描模式）。可使用 DMA_CCRx 寄存器中的 CIRC 位使能此功能。

注： *在存储器到存储器模式下，不能使用循环模式。在循环模式 (CIRC = 1) 下使能通道前，软件必须将 DMA_CCRx 寄存器的 MEM2MEM 位清零。如果循环模式已激活，则待传输数据的数目将自动重新装载为在通道配置阶段设置的初始值，并继续响应 DMA 请求。*

为停止循环传输，软件需要在禁止 DMA 通道前使外设停止生成 DMA 请求（例如退出 ADC 扫描模式）。
软件必须在启动/使能传输前，以及在停止循环传输后，明确设定 DMA_CNDTRx 值。

存储器到存储器模式

DMA 通道在没有外设请求触发的情况下同样可以工作。该模式被称为存储器到存储器模式，由软件启动。

如果 DMA_CCRx 寄存器的 MEM2MEM 位置 1，则通道（如果使能）会启动传输。DMA_CNDTRx 寄存器达到零后，传输停止。

注： *在循环模式下，不得使用存储器到存储器模式。在存储器到存储器模式 (MEM2MEM = 1) 下使能通道前，软件必须将 DMA_CCRx 寄存器的 CIRC 位清零。*

外设到外设模式

在以下情况下，任意 DMA 通道均可工作在外设到外设模式下：

- 选择来自外设的硬件请求触发 DMA 通道时

  此外设为 DMA 发起方，并且控制自身与属于其他存储器映射外设（此外设未配置为 DMA 模式）的寄存器之间的数据传输。

- 未选择任何外设请求和未将任何外设请求连接至 DMA 通道时

  软件通过将 DMA_CCRx 寄存器的 MEM2MEM 位置 1 来配置寄存器到寄存器的传输。

设定传输方向，分配源/目标

DMA_CCRx 寄存器的 DIR 位值用于设置传输方向，因此会标识源和目标，这与源/目标类型（外设或存储器）无关：

- **DIR = 1** 通常用于定义存储器到外设的传输。一般而言，如果 DIR = 1：
  - **源**属性将由 DMA_MARx 寄存器以及 DMA_CCRx 寄存器的 MSIZE[1:0] 位域和 MINC 位定义。
    无论其常用的命名规则如何，上述“存储器”寄存器、位域和位都用于在外设到外设模式下定义源外设。
  - **目标**属性将由 DMA_PARx 寄存器以及 DMA_CCRx 寄存器的 PSIZE[1:0] 位域和 PINC 位定义。
    无论其常用的命名规则如何，上述“外设”寄存器、位域和位都用于在存储器到存储器模式下定义目标存储器。
- **DIR = 0** 通常用于定义外设到存储器的传输。一般而言，如果 DIR = 0：
  - **源**属性将由 DMA_PARx 寄存器以及 DMA_CCRx 寄存器的 PSIZE[1:0] 位域和 PINC 位定义。
    无论其常用的命名规则如何，上述“外设”寄存器、位域和位都用于在存储器到存储器模式下定义源存储器。
  - **目标**属性将由 DMA_MARx 寄存器以及 DMA_CCRx 寄存器的 MSIZE[1:0] 位域和 MINC 位定义。
    无论其常用的命名规则如何，上述“存储器”寄存器、位域和位都用于在外设到外设模式下定义目标外设。

11.4.6 DMA 数据宽度、对齐和字节序

如果 PSIZE[1:0] 和 MSIZE[1:0] 不相等，则 DMA 控制器将按照*表 47* 所述方式进行数据对齐。

表 47. 可编程的数据宽度和字节序（PINC = MINC = 1 时）

源端口宽度（DIR = 1 时为 MSIZE，否则为 PSIZE）	目标端口宽度（DIR = 1 时为 PSIZE，否则为 MSIZE）	要传输的数据项的数目 (NDT)	源内容：地址或/数据（DIR = 1 时为 DMA_CMARx，否则为 DMA_CPARx）	DMA 传输	目标内容：地址/数据（DIR = 1 时为 DMA_CPARx，否则为 DMA_CMARx）
8	8	8	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1：读取 B0[7:0] @0x0，然后写入 B0[7:0] @0x0 2：读取 B1[7:0] @0x1，然后写入 B1[7:0] @0x1 3：读取 B2[7:0] @0x2，然后写入 B2[7:0] @0x2 4：读取 B3[7:0] @0x3，然后写入 B3[7:0] @0x3	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3
8	16	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1：读取 B0[7:0] @0x0，然后写入 00B0[15:0] @0x0 2：读取 B1[7:0] @0x1，然后写入 00B1[15:0] @0x2 3：读取 B2[7:0] @0x2，然后写入 00B2[15:0] @0x4 4：读取 B3[7:0] @0x3，然后写入 00B3[15:0] @0x6	@0x0/00B0 @0x2/00B1 @0x4/00B2 @0x6/00B3
8	32	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1：读取 B0[7:0] @0x0，然后写入 000000B0[31:0] @0x0 2：读取 B1[7:0] @0x1，然后写入 000000B1[31:0] @0x4 3：读取 B2[7:0] @0x2，然后写入 000000B2[31:0] @0x8 4：读取 B3[7:0] @0x3，然后写入 000000B3[31:0] @0xC	@0x0/000000B0 @0x4/000000B1 @0x8/000000B2 @0xC/000000B3
16	8	4	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6	1：读取 B1B0[15:0] @0x0，然后写入 B0[7:0] @0x0 2：读取 B3B2[15:0] @0x2，然后写入 B2[7:0] @0x1 3：读取 B5B4[15:0] @0x4，然后写入 B4[7:0] @0x2 4：读取 B7B6[15:0] @0x6，然后写入 B6[7:0] @0x3	@0x0/B0 @0x1/B2 @0x2/B4 @0x3/B6
16	16	4	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6	1：读取 B1B0[15:0] @0x0，然后写入 B1B0[15:0] @0x0 2：读取 B3B2[15:0] @0x2，然后写入 B3B2[15:0] @0x2 3：读取 B5B4[15:0] @0x4，然后写入 B5B4[15:0] @0x4 4：读取 B7B6[15:0] @0x6，然后写入 B7B6[15:0] @0x6	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6
16	32	4	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6	1：读取 B1B0[15:0] @0x0，然后写入 0000B1B0[31:0] @0x0 2：读取 B3B2[15:0] @0x2，然后写入 0000B3B2[31:0] @0x4 3：读取 B5B4[15:0] @0x4，然后写入 0000B5B4[31:0] @0x8 4：读取 B7B6[15:0] @0x6，然后写入 0000B7B6[31:0] @0xC	@0x0/0000B1B0 @0x4/0000B3B2 @0x8/0000B5B4 @0xC/0000B7B6
32	8	4	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	1：读取 B3B2B1B0[31:0] @0x0，然后写入 B0[7:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4，然后写入 B4[7:0] @0x1 3：读取 BBBAB9B8[31:0] @0x8，然后写入 B8[7:0] @0x2 4：读取 BFBEBDBC[31:0] @0xC，然后写入 BC[7:0] @0x3	@0x0/B0 @0x1/B4 @0x2/B8 @0x3/BC
32	16	4	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	1：读取 B3B2B1B0[31:0] @0x0，然后写入 B1B0[15:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4，然后写入 B5B4[15:0] @0x2 3：读取 BBBAB9B8[31:0] @0x8，然后写入 B9B8[15:0] @0x4 4：读取 BFBEBDBC[31:0] @0xC，然后写入 BDBC[15:0] @0x6	@0x0/B1B0 @0x2/B5B4 @0x4/B9B8 @0x6/BDBC
32	32	4	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	1：读取 B3B2B1B0[31:0] @0x0，然后写入 B3B2B1B0[31:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4，然后写入 B7B6B5B4[31:0] @0x4 3：读取 BBBAB9B8[31:0] @0x8，然后写入 BBBAB9B8[31:0] @0x8 4：读取 BFBEBDBC[31:0] @0xC，然后写入 BFBEBDBC[31:0] @0xC	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC

解决 AHB 外设不支持字节/半字写传输的问题

如果 DMA 控制器启动 AHB 字节或半字写传输，则 32 位 AHB 主控数据总线 (HWDATA[31:0]) 的未使用数据线上将会重复所传输的数据。

如果 AHB 从外设不支持字节或半字写传输且不会产生任何错误，则 DMA 控制器会对 HWDATA 的 32 个位执行写操作，如下列两个示例所示：

- 要写入半字 0xABCD，DMA 控制器会将 HWDATA 总线设为 0xABCDABCD，并采用半字数据大小（在 AHB 主控总线中，将 HSIZE 设为 HalfWord）。
- 要写入字节 0xAB，DMA 控制器会将 HWDATA 总线设为 0xABABABAB，并采用字节数据大小（在 AHB 主控总线中，将 HSIZE 设为 Byte）。

假设 AHB/APB 桥为 AHB 32 位从外设，且该外设不考虑 HSIZE 数据，则任何 AHB 字节或半字传输都将转换为 32 位 APB 传输，如下所述：

- 将 AHB 字节写传输转换为 APB 字写传输，如向 0x0、0x1、0x2 或 0x3 地址之一写入 0xB0 转换为向 0x0 地址写入 0xB0B0B0B0。
- 将 AHB 半字写传输转换为 APB 字写传输，如向 0x0 或 0x2 地址写入 0xB1B0 转换为向 0x0 地址写入 0xB1B0B1B0。

11.4.7 DMA 错误管理

当对保留的地址空间执行读写操作时，将生成 DMA 传输错误。如果在 DMA 读或写访问过程中生成了 DMA 传输错误，则硬件会将相应 DMA_CCRx 寄存器的 EN 位清零，从而自动禁止出错的通道 x。

DMA_ISR 寄存器的 TEIFx 位置 1。如果 DMA_CCRx 寄存器的 TEIE 位置 1，还将产生中断。

在 DMA_ISR 寄存器的 TEIFx 位清零（通过将 DMA_IFCR 寄存器的 CTEIFx 位置 1）前，DMA_CCRx 寄存器的 EN 位不能再次由软件置 1（通道 x 重新激活）。

软件在收到与外设相关的通道出现传输错误的通知后，首先要停止这个在 DMA 模式下的外设，以禁止任何挂起或后续的 DMA 请求。随后软件可以正常地将 DMA 和外设重新配置为 DMA 模式，以便进行新的传输。

11.5 DMA 中断

对于每个 DMA 通道 x，在发生“半传输”、“传输完成”或“传输错误”时都会生成中断。可以使用单独的中断使能位以提高灵活性。

表 48. DMA 中断请求

中断请求	中断事件	事件标志	中断使能位
通道 x 中断	通道 x 上发生半传输	HTIFx	HTIEx
	通道 x 上传输完成	TCIFx	TCIEx
	通道 x 上发生传输错误	TEIFx	TEIEx
	通道 x 上发生半传输、传输完成或传输错误	GIFx	-

11.6 DMA 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.2 节*。

DMA 寄存器必须按字（32 位）进行访问。

11.6.1 DMA 中断状态寄存器 (DMA_ISR)

DMA interrupt status register

偏移地址：0x00

复位值：0x0000 0000

每个状态位在软件将 DMA_IFCR 寄存器中相应的清零位或相应的全局清零位 CGIFx 置 1 后由硬件清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5
				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
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27 **TEIF7**：通道 7 的传输错误 (TE) 标志 (transfer error (TE) flag for channel 7)
0：无 TE 事件
1：发生 TE 事件

位 26 **HTIF7**：通道 7 的半传输 (HT) 标志 (half transfer (HT) flag for channel 7)
0：无 HT 事件
1：发生 HT 事件

位 25 **TCIF7**：通道 7 的传输完成 (TC) 标志 (transfer complete (TC) flag for channel 7)
0：无 TC 事件
1：发生 TC 事件

位 24 **GIF7**：通道 7 的全局中断标志 (global interrupt flag for channel 7)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

位 23 **TEIF6**：通道 6 的传输错误 (TE) 标志 (transfer error (TE) flag for channel 6)
0：无 TE 事件
1：发生 TE 事件

位 22 **HTIF6**：通道 6 的半传输 (HT) 标志 (half transfer (HT) flag for channel 6)
0：无 HT 事件
1：发生 HT 事件

位 21 **TCIF6**：通道 6 的传输完成 (TC) 标志 (transfer complete (TC) flag for channel 6)
0：无 TC 事件
1：发生 TC 事件

位 20 **GIF6**：通道 6 的全局中断标志 (global interrupt flag for channel 6)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

位 19 **TEIF5**：通道 5 的传输错误 (TE) 标志 (transfer error (TE) flag for channel 5)
0：无 TE 事件
1：发生 TE 事件

位 18 **HTIF5**：通道 5 的半传输 (HT) 标志 (half transfer (HT) flag for channel 5)
0：无 HT 事件
1：发生 HT 事件

位 17 **TCIF5**：通道 5 的传输完成 (TC) 标志 (transfer complete (TC) flag for channel 5)
0：无 TC 事件
1：发生 TC 事件

位 16 **GIF5**：通道 5 的全局中断标志 (global interrupt flag for channel 5)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

位 15 **TEIF4**：通道 4 的传输错误 (TE) 标志 (transfer error (TE) flag for channel 4)
0：无 TE 事件
1：发生 TE 事件

位 14 **HTIF4**：通道 4 的半传输 (HT) 标志 (half transfer (HT) flag for channel 4)
0：无 HT 事件
1：发生 HT 事件

位 13 **TCIF4**：通道 4 的传输完成 (TC) 标志 (transfer complete (TC) flag for channel 4)
0：无 TC 事件
1：发生 TC 事件

位 12 **GIF4**：通道 4 的全局中断标志 (global interrupt flag for channel 4)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

位 11 **TEIF3**：通道 3 的传输错误 (TE) 标志 (transfer error (TE) flag for channel 3)
0：无 TE 事件
1：发生 TE 事件

位 10 **HTIF3**：通道 3 的半传输 (HT) 标志 (half transfer (HT) flag for channel 3)
0：无 HT 事件
1：发生 HT 事件

位 9 **TCIF3**：通道 3 的传输完成 (TC) 标志 (transfer complete (TC) flag for channel 3)
0：无 TC 事件
1：发生 TC 事件

位 8 **GIF3**：通道 3 的全局中断标志 (global interrupt flag for channel 3)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

位 7 **TEIF2**：通道 2 的传输错误 (TE) 标志 (transfer error (TE) flag for channel 2)
0：无 TE 事件
1：发生 TE 事件

位 6 **HTIF2**：通道 2 的半传输 (HT) 标志 (half transfer (HT) flag for channel 2)
0：无 HT 事件
1：发生 HT 事件

位 5 **TCIF2**：通道 2 的传输完成 (TC) 标志 (transfer complete (TC) flag for channel 2)
0：无 TC 事件
1：发生 TC 事件

位 4 **GIF2**：通道 2 的全局中断标志 (global interrupt flag for channel 2)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

位 3 **TEIF1**：通道 1 的传输错误 (TE) 标志 (transfer error (TE) flag for channel 1)
0：无 TE 事件
1：发生 TE 事件

位 2 **HTIF1**：通道 1 的半传输 (HT) 标志 (half transfer (HT) flag for channel 1)
0：无 HT 事件
1：发生 HT 事件

位 1 **TCIF1**：通道 1 的传输完成 (TC) 标志 (transfer complete (TC) flag for channel 1)
0：无 TC 事件
1：发生 TC 事件

位 0 **GIF1**：通道 1 的全局中断标志 (global interrupt flag for channel 1)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

11.6.2 DMA 中断标志清零寄存器 (DMA_IFCR)

DMA interrupt flag clear register

偏移地址：0x04

复位值：0x0000 0000

将该 DMA_IFCR 寄存器中通道 x 的全局清零位 CGIFx 置 1 后，DMA 硬件会清零 DMA_ISR 寄存器中的相应 GIFx 位以及各个 TEIFx、HTIFx 和 TCIFx 标志。

将该 DMA_IFCR 寄存器中的各个 CTEIFx、CHTIFx 和 CTCIFx 清零位均置 1 后，DMA 硬件会清零 DMA_ISR 寄存器中相应的单独标志和全局标志 GIFx，前提是其他两个单独的标志均未置 1。

对任何标志清零位写入 0 均不起作用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5
				w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:28 保留，必须保持复位值。

位 27 **CTEIF7**：通道 7 的传输错误标志清零 (transfer error flag clear for channel 7)

位 26 **CHTIF7**：通道 7 的半传输标志清零 (half transfer flag clear for channel 7)

位 25 **CTCIF7**：通道 7 的传输完成标志清零 (transfer complete flag clear for channel 7)

位 24 **CGIF7**：通道 7 的全局中断标志清零 (global interrupt flag clear for channel 7)

位 23 **CTEIF6**：通道 6 的传输错误标志清零 (transfer error flag clear for channel 6)

位 22 **CHTIF6**：通道 6 的半传输标志清零 (half transfer flag clear for channel 6)

位 21 **CTCIF6**：通道 6 的传输完成标志清零 (transfer complete flag clear for channel 6)

位 20 **CGIF6**：通道 6 的全局中断标志清零 (global interrupt flag clear for channel 6)

位 19 **CTEIF5**：通道 5 的传输错误标志清零 (transfer error flag clear for channel 5)

位 18 **CHTIF5**：通道 5 的半传输标志清零 (half transfer flag clear for channel 5)

位 17 **CTCIF5**：通道 5 的传输完成标志清零 (transfer complete flag clear for channel 5)

位 16 **CGIF5**：通道 5 的全局中断标志清零 (global interrupt flag clear for channel 5)

位 15 **CTEIF4**：通道 4 的传输错误标志清零 (transfer error flag clear for channel 4)

位 14 **CHTIF4**：通道 4 的半传输标志清零 (half transfer flag clear for channel 4)

位 13 **CTCIF4**：通道 4 的传输完成标志清零 (transfer complete flag clear for channel 4)

位 12 **CGIF4**：通道 4 的全局中断标志清零 (global interrupt flag clear for channel 4)

位 11 **CTEIF3**：通道 3 的传输错误标志清零 (transfer error flag clear for channel 3)

位 10 **CHTIF3**：通道 3 的半传输标志清零 (half transfer flag clear for channel 3)

位 9 **CTCIF3**：通道 3 的传输完成标志清零 (transfer complete flag clear for channel 3)

位 8 **CGIF3**：通道 3 的全局中断标志清零 (global interrupt flag clear for channel 3)

位 7 **CTEIF2**：通道 2 的传输错误标志清零 (transfer error flag clear for channel 2)

位 6 **CHTIF2**：通道 2 的半传输标志清零 (half transfer flag clear for channel 2)

位 5 **CTCIF2**：通道 2 的传输完成标志清零 (transfer complete flag clear for channel 2)

位 4 **CGIF2**：通道 2 的全局中断标志清零 (global interrupt flag clear for channel 2)

位 3 **CTEIF1**：通道 1 的传输错误标志清零 (transfer error flag clear for channel 1)

位 2 **CHTIF1**：通道 1 的半传输标志清零 (half transfer flag clear for channel 1)

位 1 **CTCIF1**：通道 1 的传输完成标志清零 (transfer complete flag clear for channel 1)

位 0 **CGIF1**：通道 1 的全局中断标志清零 (global interrupt flag clear for channel 1)

11.6.3 DMA 通道 x 配置寄存器 (DMA_CCRx)

DMA channel x configuration register

偏移地址：0x08 + 0x14 * (x - 1)，（x = 1 到 7）

复位值：0x0000 0000

EN = 1 时，寄存器位域/位 MEM2MEM、PL[1:0]、MSIZE[1:0]、PSIZE[1:0]、MINC、PINC 和 DIR 为只读。

MEM2MEM 和 CIRC 位的状态不能同时置 1。

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

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

位 31:15 保留，必须保持复位值。

位 14 **MEM2MEM**：存储器到存储器模式 (memory-to-memory mode)

0：禁止

1：使能

注： 此位由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后为只读。

位 13:12 **PL[1:0]**：优先级 (priority level)

00：低

01：中

10：高

11：非常高

注： 此位域由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后为只读。

位 11:10 **MSIZE[1:0]**：存储器大小 (memory size)

定义到标识存储器的每次 DMA 传输的数据大小。

在存储器到存储器模式下，如果 DIR = 1，则该位域标识存储器源；如果 DIR = 0，则该位域标识存储器目标。

在外设到外设模式下，如果 DIR = 1，则该位域标识外设源；如果 DIR = 0，则该位域标识外设目标。

00：8 位

01：16 位

10：32 位

11：保留

注： 此位域由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后为只读。

位 9:8 **PSIZE[1:0]**：外设大小 (peripheral size)

定义到标识外设的每次 DMA 传输的数据大小。

在存储器到存储器模式下，如果 DIR = 1，则该位域标识存储器目标；如果 DIR = 0，则该位域标识存储器源。

在外设到外设模式下，如果 DIR = 1，则该位域标识外设目标；如果 DIR = 0 则该位域标识外设源。

00：8 位

01：16 位

10：32 位

11：保留

注： 此位域由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后为只读。

位 7 **MINC**：存储器递增模式 (memory increment mode)

定义到标识存储器的每次 DMA 传输的递增模式。

在存储器到存储器模式下，如果 DIR = 1，则该位域标识存储器源；如果 DIR = 0，则该位域标识存储器目标。

在外设到外设模式下，如果 DIR = 1，则该位域标识外设源；如果 DIR = 0，则该位域标识外设目标。

0：禁止

1：使能

注：　此位由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后为只读。

位 6 **PINC**：外设递增模式 (peripheral increment mode)

定义到标识外设的每次 DMA 传输的递增模式。

在存储器到存储器模式下，如果 DIR = 1，则该位域标识存储器目标；如果 DIR = 0，则该位域标识存储器源。

在外设到外设模式下，如果 DIR = 1，则该位域标识外设目标；如果 DIR = 0 则该位域标识外设源。

0：禁止

1：使能

注：　此位由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后为只读。

位 5 **CIRC**：循环模式 (circular mode)

0：禁止

1：使能

注：　此位由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后并非只读。

位 4 **DIR**：数据传输方向 (data transfer direction)

该位仅在存储器到外设和外设到存储器模式下才能置 1。

0：从外设读取

- 源属性由 PSIZE 和 PINC 以及 DMA_CPARx 寄存器定义。这同样适用于存储器到存储器模式。
- 目标属性由 MSIZE 和 MINC 以及 DMA_CMARx 寄存器定义。这同样适用于外设到外设模式。

1：从存储器读取

- 目标属性由 PSIZE 和 PINC 以及 DMA_CPARx 寄存器定义。这同样适用于存储器到存储器模式。
- 源属性由 MSIZE 和 MINC 以及 DMA_CMARx 寄存器定义。这同样适用于外设到外设模式。

注：　此位由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后为只读。

位 3 **TEIE**：传输错误中断使能 (transfer error interrupt enable)

0：禁止

1：使能

注：　此位由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后并非只读。

位 2 **HTIE**：半传输中断使能 (half transfer interrupt enable)

0：禁止

1：使能

注： 此位由软件置 1 和清零。

使能通道 (EN = 1) 后禁止写入。

使能通道 (EN = 1) 后并非只读。

位 1 **TCIE**：传输完成中断使能 (transfer complete interrupt enable)

0：禁止

1：使能

注： 此位由软件置 1 和清零。

使能通道 (EN = 1) 后禁止写入。

使能通道 (EN = 1) 后并非只读。

位 0 **EN**：通道使能 (channel enable)

发生通道传输错误后，该位由硬件清零。在 DMA_ISR 寄存器的 TEIFx 位清零（通过将 DMA_IFCR 寄存器的 CTEIFx 位置 1）前，该位不能再次由软件置 1（通道 x 重新激活）。

0：禁止

1：使能

注： 此位由软件置 1 和清零。

11.6.4 DMA 通道 x 待传输数据数量寄存器 (DMA_CNDTRx)

DMA channel x number of data to transfer register

偏移地址：0x0C + 0x14 * (x - 1)，（x = 1 到 7）

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 **NDT[15:0]**：待传输的数据数 (number of data to transfer)（0 到 2^{16} - 1）

通道使能后，该位域由硬件更新：

– 在每次 DMA“先读后写”传输后，该位域都会递减，指示剩余的待传输数据项数目。

– 如果通道未处于循环模式（DMA_CCRx 寄存器中的 CIRC = 0），则在达到设定的待传输数据数目时，该位域会保持为零。

– 如果通道处于循环模式 (CIRC = 1)，则在传输完成后该位域会自动重新装载之前设定的值。

如果该位域为零，则无论通道状态如何（使能或未使能），都不会处理任何传输。

注： 此位域由软件置 1 和清零。

使能通道 (EN = 1) 后禁止写入。

使能通道 (EN = 1) 后为只读。

11.6.5 DMA 通道 x 外设地址寄存器 (DMA_CPARx)

DMA channel x peripheral address register

偏移地址：0x10 + 0x14 * (x - 1)，（x = 1 到 7）

复位值：0x0000 0000

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

位 31:0 **PA[31:0]**：外设地址 (peripheral address)

其中包含读/写数据的外设数据寄存器的基址。

PSIZE[1:0] = 01（16 位）时，忽略 PA[31:0] 的位 0。访问将自动对齐到半字地址。

PSIZE = 10（32 位）时，忽略 PA[31:0] 的位 1 和位 0。访问将自动对齐到字地址。

在存储器到存储器模式下，如果 DIR = 1，则该寄存器标识存储器目标地址；如果 DIR = 0，则该寄存器标识存储器源地址。

在外设到外设模式下，如果 DIR = 1，则该寄存器标识外设目标地址；如果 DIR = 0，则该寄存器标识外设源地址。

注：　此寄存器由软件置 1 和清零。

使能通道 (EN = 1) 后禁止写入。

使能通道 (EN = 1) 后并非只读。

11.6.6 DMA 通道 x 存储器地址寄存器 (DMA_CMARx)

DMA channel x memory address register

偏移地址：0x14 + 0x14 * (x - 1)，（x = 1 到 7）

复位值：0x0000 0000

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

位 31:0 **MA[31:0]**：外设地址 (peripheral address)

其中包含读/写数据的存储器的基址。

MSIZE[1:0] = 01（16 位）时，忽略 MA[31:0] 的位 0。访问将自动对齐到半字地址。

MSIZE = 10（32 位）时，忽略 MA[31:0] 的位 1 和位 0。访问将自动对齐到字地址。

在存储器到存储器模式下，如果 DIR = 1，则该寄存器标识存储器源地址；如果 DIR = 0，则该寄存器标识存储器目标地址。

在外设到外设模式下，如果 DIR = 1，则该寄存器标识外设源地址；如果 DIR = 0，则该寄存器标识外设目标地址。

注：　此寄存器由软件置 1 和清零。

使能通道 (EN = 1) 后禁止写入。

使能通道 (EN = 1) 后并非只读。

11.6.7 DMA 寄存器映射和复位值

表 49 给出了 DMA 寄存器映射和复位值。

表 49. DMA 寄存器映射和复位值

偏移	寄存器	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
0x000	DMA_ISR	Res.	Res.	Res.	Res.	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5	TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
	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
0x004	DMA_IFCR	Res.	Res.	Res.	Res.	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5	CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
	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
0x008	DMA_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	DMA_CNDTR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	DMA_CPAR1	PA[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
0x014	DMA_CMAR1	MA[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
0x018	Reserved	Reserved																															
0x01C	DMA_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	DMA_CNDTR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	DMA_CPAR2	PA[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
0x028	DMA_CMAR2	MA[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
0x02C	Reserved	Reserved																															
0x030	DMA_CCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	DMA_CNDTR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x038	DMA_CPAR3	PA[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
0x03C	DMA_CMAR3	MA[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
0x040	Reserved	Reserved																															
0x044	DMA_CCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 49. DMA 寄存器映射和复位值（续）

偏移	寄存器	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
0x048	DMA_CNDTR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04C	DMA_CPAR4	PA[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
0x050	DMA_CMAR4	MA[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
0x054	Reserved	Reserved.																															
0x058	DMA_CCR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x05C	DMA_CNDTR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x060	DMA_CPAR5	PA[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
0x064	DMA_CMAR5	MA[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
0x068	Reserved	Reserved.																															
0x06C	DMA_CCR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x070	DMA_CNDTR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x074	DMA_CPAR6	PA[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
0x078	DMA_CMAR6	MA[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
0x07C	Reserved	Reserved.																															
0x080	DMA_CCR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	DMA_CNDTR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x088	DMA_CPAR7	PA[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
0x08C	DMA_CMAR7	MA[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

有关寄存器边界地址的信息，请参见*第 2.2 节*。

12 DMA 请求复用器 (DMAMUX)

12.1 简介

外设通过设置其 DMA 请求信号来指示存在 DMA 传输请求。DMA 控制器会处理 DMA 请求并生成 DMA 确认信号，而且相应的 DMA 请求信号也将变为无效，但在此之前 DMA 请求一直处于挂起状态。

在本文档中，DMA 请求/确认协议所需的控制信号组并未明确列出和说明，而是称作 DMA 请求线。

DMAMUX 请求复用器可在产品的外设和 DMA 控制器之间重新配置（路由）DMA 请求线。该路由功能通过可编程的多通道 DMA 请求线复用器来确保实现。每条通道可不受限制地选择一个 DMA 请求线，或者按照与来自其 DMAMUX 同步输入的事件相同步的方式选择一个 DMA 请求线。此外，DMAMUX 还可用作其输入触发信号的可编程事件的 DMA 请求发生器。

*第 12.3.1 节*规定了 DMAMUX 实例的数量及其主要特性。

DMAMUX 请求复用器输入到外设 DMA 请求线和到 DMAMUX 请求发生器输出的分配、DMAMUX 请求复用器输出到 DMA 控制器通道的分配以及 DMAMUX 同步和触发输入到内部和外部信号的分配取决于产品实现，具体在*第 12.3.2 节*中进行了介绍。

12.2 DMAMUX 主要特性

- 14 通道可编程 DMA 请求线复用器输出
- 4 通道 DMA 请求发生器
- 20 个 DMA 请求发生器的触发输入
- 20 个同步输入
- 每个 DMA 请求发生器通道均具有：
  - DMA 请求触发输入选择器
  - DMA 请求计数器
  - 所选 DMA 请求触发输入的事件溢出标志
- 每个 DMA 请求线复用器通道输出均具有：
  - 36 个来自外设的输入 DMA 请求线
  - 一个 DMA 请求线输出
  - 同步输入选择器
  - DMA 请求计数器
  - 所选同步输入的事件溢出标志
  - 一个事件输出，用于 DMA 请求链接

12.3 DMAMUX 实现

12.3.1 DMAMUX 实例化

DMAMUX 通过下表中列出的硬件配置参数进行实例化。

表 50. DMAMUX 实例化

特性	DMAMUX
DMAMUX 输出请求通道数	14
DMAMUX 请求发生器通道数	4
DMAMUX 请求触发输入数	20
DMAMUX 同步输入数	20
DMAMUX 外设请求输入数	36

12.3.2 DMAMUX 映射

资源到 DMAMUX 的映射通过硬接线实现。

DMAMUX 与 DMA1 和 DMA2 配合使用：

- DMAMUX 通道 0 到 6 与 DMA1 通道 1 到 7 相连
- DMAMUX 通道 7 到 13 与 DMA2 通道 1 到 7 相连

表 51. DMAMUX：复用器输入到资源的分配

DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源
1	dmamux_req_gen0	22	TIM1_CH2	43	保留
2	dmamux_req_gen1	23	TIM1_CH3	44	保留
3	dmamux_req_gen2	24	TIM1_CH4	45	保留
4	dmamux_req_gen3	25	TIM1_UP	46	保留
5	ADC1	26	TIM1_TRIG	47	保留
6	SPI1_RX	27	TIM1_COM	48	保留
7	SPI1_TX	28	TIM2_CH1	49	保留
8	SPI2_RX	29	TIM2_CH2	50	保留
9	SPI2_TX	30	TIM2_CH3	51	保留
10	I2C1_RX	31	TIM2_CH4	52	保留
11	I2C1_TX	32	TIM2_UP	53	保留
12	I2C3_RX	33	TIM16_CH1	54	保留

表 51. DMAMUX：复用器输入到资源的分配（续）

DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源
13	I2C3_TX	34	TIM16_UP	55	保留
14	USART1_RX	35	TIM17_CH1	56	保留
15	USART1_TX	36	TIM17_UP	57	保留
16	LPUART1_RX	37	AES1_IN	58	保留
17	LPUART1_TX	38	AES1_OUT	59	保留
18	SAI1_A	39	AES2_IN	60	保留
19	SAI1_B	40	AES2_OUT	61	保留
20	QUADSPI	41	保留	62	保留
21	TIM1_CH1	42	保留	63	保留

表 52. DMAMUX：触发输入到资源的分配

触发输入	资源	触发输入	资源
0	EXTI 线 0	16	dmamux_evt0
1	EXTI 线 1	17	dmamux_evt1
2	EXTI 线 2	18	LPTIM1_OUT
3	EXTI 线 3	19	LPTIM2_OUT
4	EXTI 线 4	20	保留
5	EXTI 线 5	21	保留
6	EXTI 线 6	22	保留
7	EXTI 线 7	23	保留
8	EXTI 线 8	24	保留
9	EXTI 线 9	25	保留
10	EXTI 线 10	26	保留
11	EXTI 线 11	27	保留
12	EXTI 线 12	28	保留
13	EXTI 线 13	29	保留
14	EXTI 线 14	30	保留
15	EXTI 线 15	31	保留

表 53. DMAMUX：同步输入到资源的分配

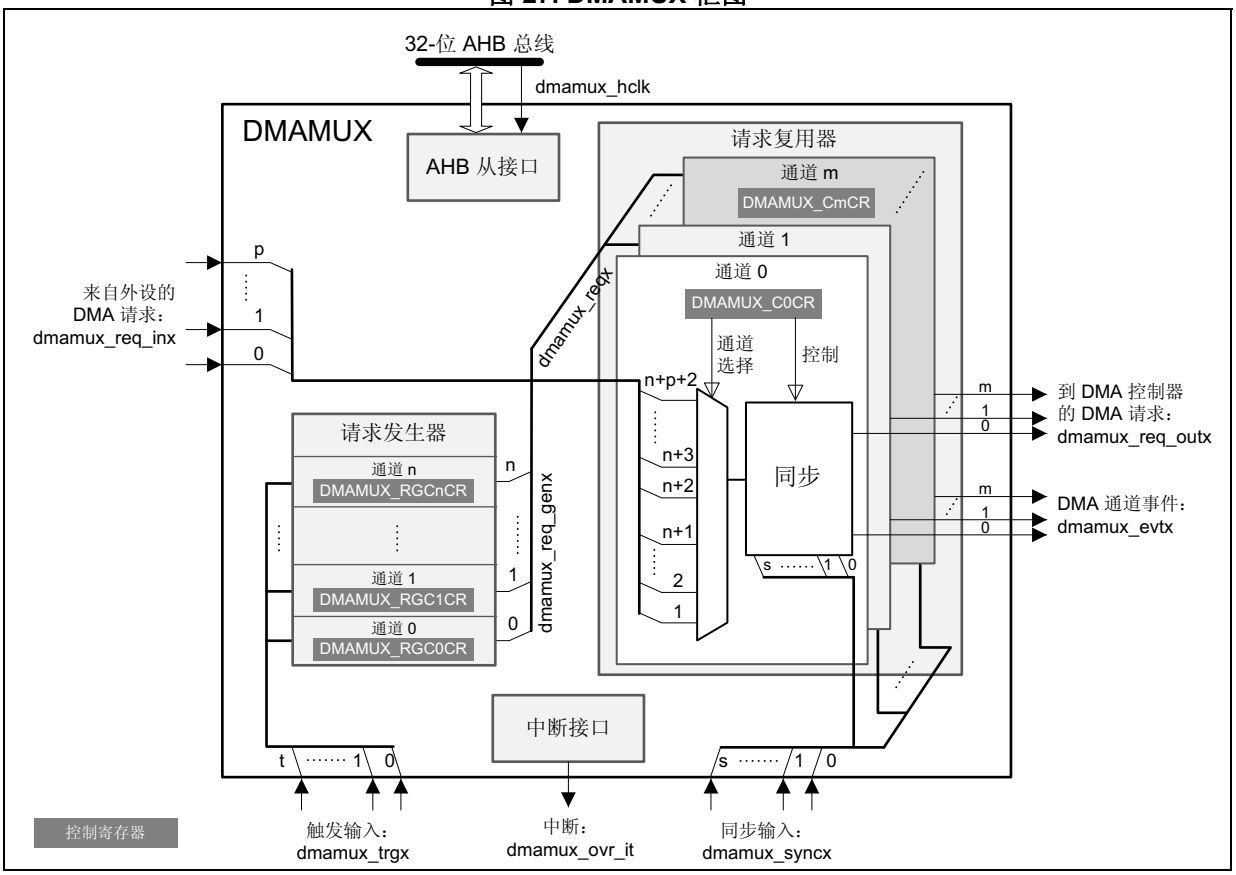
SYNC 输入	资源	SYNC 输入	资源
0	EXTI 线 0	16	dmamux_evt0
1	EXTI 线 1	17	dmamux_evt1
2	EXTI 线 2	18	LPTIM1_OUT
3	EXTI 线 3	19	LPTIM2_OUT
4	EXTI 线 4	20	保留
5	EXTI 线 5	21	保留
6	EXTI 线 6	22	保留
7	EXTI 线 7	23	保留
8	EXTI 线 8	24	保留
9	EXTI 线 9	25	保留
10	EXTI 线 10	26	保留
11	EXTI 线 11	27	保留
12	EXTI 线 12	28	保留
13	EXTI 线 13	29	保留
14	EXTI 线 14	30	保留
15	EXTI 线 15	31	保留

12.4 DMAMUX 功能说明

12.4.1 DMAMUX 框图

图 27 显示了 DMAMUX 框图。

图 27. DMAMUX 框图



DMAMUX 具有两个主要子模块：请求线复用器和请求线发生器。

该实现可：

- 分配 DMAMUX 请求复用器子模块输入 (dmamux_reqx)，这些输入来自外设 (dmamux_req_inx) 和 DMAMUX 请求发生器子模块的通道 (dmamux_req_genx)
- 将 DMAMUX 请求输出分配到 DMA 控制器的通道 (dmamux_req_outx)
- 将内部或外部信号分配到 DMA 请求触发输入 (dmamux_trgx)
- 将内部或外部信号分配到同步输入 (dmamux_syncx)

12.4.2 DMAMUX 信号

表 54 列出了 DMAMUX 信号。

表 54. DMAMUX 信号

信号名称	说明
dmamux_hclk	DMAMUX AHB 时钟
dmamux_req_inx	DMAMUX DMA 请求线输入（来自外设）
dmamux_trgx	DMAMUX DMA 请求触发输入（到请求发生器子模块）
dmamux_req_genx	DMAMUX 请求发生器子模块通道输出
dmamux_reqx	DMAMUX 请求复用器子模块输入（来自外设请求和请求发生器通道）
dmamux_syncx	DMAMUX 同步输入（到请求复用器子模块）
dmamux_req_outx	DMAMUX 请求输出（到 DMA 控制器）
dmamux_evtx	DMAMUX 事件输出
dmamux_ovr_it	DMAMUX 溢出中断

12.4.3 DMAMUX 通道

DMAMUX 通道是一个可能包括额外的 DMAMUX 请求发生器通道的 DMAMUX 请求复用器通道，具体取决于所选的请求复用器输入。

DMAMUX 请求复用器通道专用于连接到 DMA 控制器的一个通道。

通道配置流程

请遵循以下顺序来配置 DMAMUX x 通道和相关的 DMA 通道 y：

1. 对 DMA 通道 y 进行完整的设置和配置，但不要使能通道 y。
2. 对相关 DMAMUX y 通道进行完整的设置和配置。
3. 最后，将 DMA y 通道寄存器中的 EN 位置 1 以激活 DMA 通道。

12.4.4 DMAMUX 请求线复用器

DMAMUX 请求复用器及其多条通道可确保 DMA 请求/确认控制信号（称为 DMA 请求线）的实际路由。

每个 DMA 请求线并行连接到 DMAMUX请求线复用器的所有通道。

DMA 请求来自外设或 DMAMUX 请求发生器。

DMAMUX 请求线复用器通道 x 按照 DMAMUX_CxCR 寄存器中的 DMAREQ_ID 字段的配置来选择 DMA 请求线编号。

注： *字段 DMAREQ_ID 为空值表示未选择任何 DMA 请求线。不允许为 DMAMUX 请求线复用器的两个不同通道配置相同的非零 DMAREQ_ID。*

除了 DMA 请求选择外，还可根据需要配置和使能同步模式和/或事件生成。

同步模式和通道事件生成

每个 DMAMUX 请求线复用器通道 x 可单独同步，方法是将 DMAMUX_CxCR 寄存器中的同步使能 (SE) 位置 1。

DMAMUX 具有多个同步输入。同步输入并行连接到请求复用器的所有通道。

同步输入通过给定通道 x 的 DMAMUX_CxCR 寄存器中的 SYNC_ID 字段选择。

当某个通道处于此同步模式时，如果通过 DMAMUX_CxCR 寄存器的 SPOL[1:0] 字段检测到所选输入同步信号上的可编程上升沿/下降沿，则所选输入 DMA 请求线信号将被传送到复用器通道输出。

此外，DMAMUX 请求复用器内部有一个可编程 DMA 请求计数器，可用于实现通道请求输出生成功能以及事件生成功能。通道 x 输出上的事件生成功能通过 DMAMUX_CxCR 寄存器的 EGE 位（事件生成使能）使能。

如*图 29* 所示，检测到同步输入的边沿后，挂起的所选输入 DMA 请求线将连接到 DMAMUX 复用器通道 x 输出。

注： *如果在不存在挂起的所选输入 DMA 请求线的情况下，发生同步事件，则会将其丢弃。再次发生同步事件之前，后续有效的输入请求线将不会连接至 DMAMUX 复用器通道输出。*

此后，DMA 控制器每次处理连接的 DMAMUX 请求时（已处理请求被禁止），DMAMUX 请求计数器都会递减。下溢时，DMA 请求计数器将自动装入 DMAMUX_CxCR 寄存器的 NBREQ 字段中的值，输入 DMA 请求线将与复用器通道 x 输出断开。

因此，检测到同步事件后传送到复用器通道 x 输出的 DMA 请求数等于 NBREQ 字段中的值加 1。

注： *只能通过软件在相应复用器通道 x 的同步使能位 SE 和事件生成使能位 EGE 均禁止时写入 NBREQ 字段的值。*

图 28. DMAMUX 请求线复用器通道的同步模式

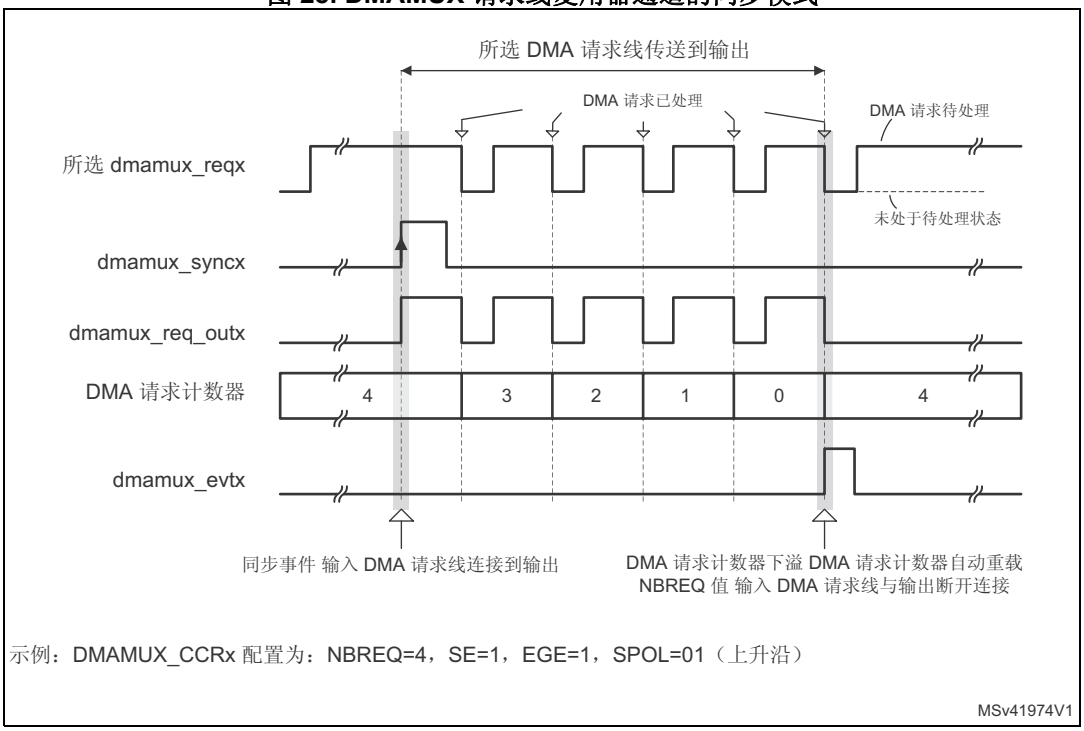
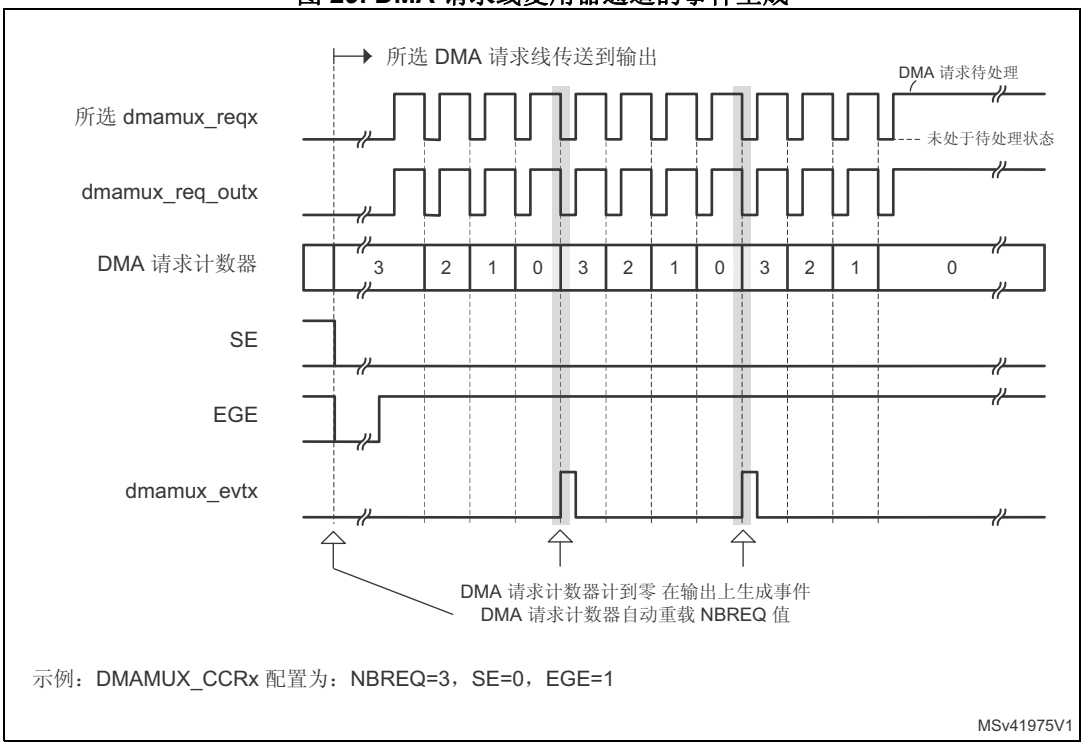


图 29. DMA 请求线复用器通道的事件生成



如果 EGE 使能，当复用器通道的 DMA 请求计数器自动重新装入已编程 NBREQ 字段的值时，复用器通道会生成一个通道事件（即，一个 AHB 时钟周期的脉冲），如*图 28* 和*图 29* 所示。

注： *如果 EGE 使能且 NBREQ = 0，则在每个处理的 DMA 请求后都会生成一个事件。*

注： *如果边沿后的状态保持稳定的时间持续 2 个 AHB 时钟周期以上，则会检测到同步事件（边沿）。*

写入 DMAMUX_CxCR 寄存器后，同步事件将被屏蔽 3 个 AHB 时钟周期。

同步溢出和中断

如果在请求计数器下溢（通过 DMAMUX_CxCR 寄存器的 NBREQ 字段编程的内部请求计数器）之前发生新的同步事件，则 DMAMUX_CSR 状态寄存器中的同步溢出标志位 SOFx 会置 1。

注： *DMA 控制器的相关通道使用结束后，应禁止请求复用器通道 x 同步 (DMAMUX_CxCR.SE = 0)。否则，在新检测到的同步事件后，由于未从 DMA 控制器收到 DMA 确认（即无已处理请求），将会发生同步溢出。*

溢出标志 SOFx 的复位方式是将 DMAMUX_CFR 寄存器中的相关清除同步溢出标志位 CSOFx 置 1。

如果 DMAMUX_CxCR 寄存器中的同步溢出中断使能位 SOIE 置 1，则将同步溢出标志置 1 会产生中断。

12.4.5 DMAMUX 请求发生器

DMAMUX 请求发生器会在其 DMA 请求触发输入上出现触发事件后产生 DMA 请求。

DMAMUX 请求发生器有多个通道。DMA 请求触发输入并行连接到所有通道。

DMAMUX 请求发生器通道的输出是 DMAMUX 请求线复用器的输入。

每个 DMAMUX 请求发生器通道 x 在相应的 DMAMUX_RGxCR 寄存器中都有一个使能位 GE（发生器使能）。

DMAMUX 请求发生器通道 x 的 DMA 请求触发输入通过相应 DMAMUX_RGxCR 寄存器中的 SIG_ID（触发信号 ID）字段选择。

DMA 请求触发输入上的触发事件可以是上升沿、下降沿或任一边沿。有效边沿通过相应 DMAMUX_RGxCR 寄存器中的 GPOL（发生器极性）字段选择。

触发事件后，相应发生器通道开始在其输出上生成 DMA 请求。连接的 DMA 控制器每次处理 DMAMUX 生成的请求时（已处理请求被禁止），内置（DMAMUX 请求发生器内）DMA 请求计数器都会递减。下溢时，请求发生器通道会停止生成 DMA 请求，DMA 请求计数器将在出现下一个触发事件时自动装入其设定值。

因此，触发事件后生成的 DMA 请求的数量为 GNBREQ + 1。

注： *只能通过软件在相应发生器通道 x 的使能位 GE 被禁止时写入 GNBREQ 字段的值。*

如果边沿后的状态保持稳定的时间持续 2 个 AHB 时钟周期以上，则会检测到触发事件（边沿）。

写入 DMAMUX_RGxCR 寄存器后，触发事件将被屏蔽 3 个 AHB 时钟周期。

触发溢出和中断

如果在 DMAMUX 请求发生器计数器下溢（通过 DMAMUX_RGxCR 寄存器的 GNBREQ 字段编程的内部计数器）之前发生新的 DMA 请求触发事件，并且请求发生器通道 x 通过 GE 使能，则状态 DMAMUX_RGSR 寄存器中的请求触发事件溢出标志位 OFx 将通过硬件置为有效。

注： *DMA 控制器的相关通道使用结束后，应禁止请求发生器通道 x (DMAMUX_RGxCR.GE = 0)。否则，在新检测到的触发事件后，由于未从 DMA 收到确认（即无已处理请求），将会发生触发溢出。*

溢出标志 OFx 的复位方式是将 DMAMUX_RGCFR 寄存器中的相关清除溢出标志位 COFx 置 1。

如果 DMAMUX_RGxCR 寄存器中的 DMA 请求触发事件溢出中断使能位 OIE 置 1，则将 DMAMUX 请求触发溢出标志置 1 会产生中断。

12.5 DMAMUX 中断

发生以下事件时会产生中断：

- 每个 DMA 请求线复用器通道发生同步事件溢出
- 每个 DMA 请求发生器通道发生触发事件溢出

对任一事件，每个通道的中断使能、状态和清除标志寄存器位均可单独使用。

表 55. DMAMUX 中断

中断信号	中断事件	事件标志	清除位	使能位
dmamuxovr_it	DMAMUX 请求线复用器的通道 x 上发生同步事件溢出	SOFx	CSOFx	SOIE
	DMAMUX 请求发生器的通道 x 上发生触发事件溢出	OFx	COFx	OIE

12.6 DMAMUX 寄存器

有关 DMAMUX 基址的信息，请参见包括寄存器边界地址的表格。

DMAMUX 寄存器可按（8 位）字节、（16 位）半字或（32 位）字访问。地址应与数据大小对齐。

12.6.1 DMAMUX 请求线复用器通道 x 配置寄存器 (DMAMUX_CxCR)

DMAMUX request line multiplexer channel x configuration register

偏移地址：0x000 + 0x04 * x（x = 0 到 13）

复位值：0x0000 0000

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

位 31:29 保留，必须保持复位值。

位 28:24 **SYNC_ID[4:0]**：同步标识 (Synchronization identification)
选择同步输入（请参见*表 53：DMAMUX：同步输入到资源的分配*）。

位 23:19 **NBREQ[4:0]**：要转发的 DMA 请求数减 1 (Number of DMA requests minus 1 to forward)
定义在同步事件后要转发到 DMA 控制器的 DMA 请求的数量，和/或生成输出事件前的 DMA 请求的数量。
只有当 SE 和 EGE 位均为低电平时，才能写入该字段。

位 18:17 **SPOL[1:0]**：同步极性 (Synchronization polarity)
定义所选同步输入的边沿极性：
00：无事件，即，无同步也无检测。
01：上升沿
10：下降沿
11：上升沿和下降沿

位 16 **SE**：同步使能 (Synchronization enable)
0：禁止同步
1：使能同步

位 15:10 保留，必须保持复位值。

位 9 **EGE**：事件生成使能 (Event generation enable)
0：禁止事件生成
1：使能事件生成

位 8 **SOIE**：同步溢出中断使能 (Synchronization overrun interrupt enable)
0：禁止中断
1：使能中断

位 7:6 保留，必须保持复位值。

位 5:0 **DMAREQ_ID[5:0]**：DMA 请求标识 (DMA request identification)
选择输入 DMA 请求。有关复用器输入到资源的分配的信息，请参见 DMAMUX 表。

12.6.2 DMAMUX 请求线复用器中断通道状态寄存器 (DMAMUX_CSR)

DMAMUX request line multiplexer interrupt channel status register

偏移地址：0x080

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	SOF13	SOF12	SOF11	SOF10	SOF9	SOF8	SOF7	SOF6	SOF5	SOF4	SOF3	SOF2	SOF1	SOF0
		r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:14 保留，必须保持复位值。

位 13:0 **SOF[13:0]**：同步溢出事件标志 (Synchronization overrun event flag)
当 DMA 请求线复用器通道 x 上发生同步事件且 DMA 请求计数器的值小于 NBREQ 时，该标志将置 1。
该标志的清零方式是向 DMAMUX_CFR 寄存器中的相应 CSOFx 位写入 1。

12.6.3 DMAMUX 请求线复用器中断清除标志寄存器 (DMAMUX_CFR)

DMAMUX request line multiplexer interrupt clear flag register

偏移地址：0x084

复位值：0x0000 0000

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

位 31:14 保留，必须保持复位值。

位 13:0 **CSOF[13:0]**：清除同步溢出事件标志 (Clear synchronization overrun event flag)
将 1 写入每个位时，DMAMUX_CSR 寄存器中相应的溢出标志 SOFx 将清零。

12.6.4 DMAMUX 请求发生器通道 x 配置寄存器 (DMAMUX_RGxCR)

DMAMUX request generator channel x configuration register

偏移地址：0x100 + 0x04 * x (x = 0 到 3)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL[1:0]		GE
								rw	rw	rw	rw	rw	rw	rw	rw

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

位 31:24 保留，必须保持复位值。

位 23:19 **GNBREQ[4:0]**：要生成的 DMA 请求数（减 1）(Number of DMA requests to be generated (minus 1))

定义触发事件后生成的 DMA 请求的数量。生成的 DMA 请求的实际数量为 GNBREQ+1。

注： 只有在 GE 位禁止时才能写入此字段。

位 18:17 **GPOL[1:0]**：DMA 请求发生器触发极性 (DMA request generator trigger polarity)

定义所选触发输入的边沿极性

00：无事件。即，无触发检测和生成。

01：上升沿

10：下降沿

11：上升沿和下降沿

位 16 **GE**：DMA 请求发生器通道 x 使能 (DMA request generator channel x enable)

0：禁止 DMA 请求发生器通道 x

1：使能 DMA 请求发生器通道 x

位 15:9 保留，必须保持复位值。

位 8 **OIE**：触发溢出中断使能 (Trigger overrun interrupt enable)

0：禁止在出现触发溢出事件时产生中断

1：使能在出现触发溢出事件时产生中断

位 7:5 保留，必须保持复位值。

位 4:0 **SIG_ID[4:0]**：信号标识 (Signal identification)

选择用于 DMA 请求发生器的通道 x 的 DMA 请求触发输入

12.6.5 DMAMUX 请求发生器中断状态寄存器 (DMAMUX_RGSR)

DMAMUX request generator interrupt status register

偏移地址：0x140

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OF3	OF2	OF1	OF0
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 3:0 **OF[3:0]**：触发溢出事件标志 (Trigger overrun event flag)

在请求计数器下溢（通过 DMAMUX_RGxCR 寄存器的 GNBREQ 字段编程的内部请求计数器）之前，在 DMA 请求发生器通道 x 上发生新触发事件时，该标志将置 1。

该标志的清零方式是向 DMAMUX_RGCFR 寄存器中的相应 COFx 位写入 1。

12.6.6 DMAMUX 请求发生器中断清除标志寄存器 (DMAMUX_RGCFR)

DMAMUX request generator interrupt clear flag register

偏移地址：0x144

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COF3	COF2	COF1	COF0
												w	w	w	w

位 31:4 保留，必须保持复位值。

位 3:0 **COF[3:0]**：清除触发溢出事件标志 (Clear trigger overrun event flag)

将 1 写入每个位时，DMAMUX_RGSR 寄存器中相应的溢出标志 OFx 将清零。

12.6.7 DMAMUX 寄存器映射

下表汇总了 DMAMUX 寄存器和复位值。有关 DMAMUX 寄存器的基址，请参见寄存器边界地址表。

表 56. DMAMUX 寄存器映射和复位值

偏移	寄存器	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
0x000	DMAMUX_C0CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x004	DMAMUX_C1CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x008	DMAMUX_C2CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x00C	DMAMUX_C3CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x010	DMAMUX_C4CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x014	DMAMUX_C5CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x018	DMAMUX_C6CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x01C	DMAMUX_C7CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x020	DMAMUX_C8CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x024	DMAMUX_C9CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x028	DMAMUX_C10CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x02C	DMAMUX_C11CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x030	DMAMUX_C12CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x034	DMAMUX_C13CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	Res.	DMAREQ_ID[5:0]					
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0			0	0	0	0	0	0
0x038 - 0x07C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x080	DMAMUX_CSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOF13	SOF12	SOF11	SOF10	SOF9	SOF8	SOF7	SOF6	SOF5	SOF4	SOF3	SOF2	SOF1	SOF0
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	DMAMUX_CFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSOF13	CSOF12	CSOF11	CSOF10	CSOF9	CSOF8	CSOF7	CSOF6	CSOF5	CSOF4	CSOF3	CSOF2	CSOF1	CSOF0
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 56. DMAMUX 寄存器映射和复位值（续）

偏移	寄存器	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
0x088 - 0x0FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x100	DMAMUX_RG0CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	Reset value									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x104	DMAMUX_RG1CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	Reset value									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x108	DMAMUX_RG2CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	Reset value									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x10C	DMAMUX_RG3CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	Reset value									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x110 - 0x13C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x140	DMAMUX_RGSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OF3	OF2	OF1	OF0
	Reset value																													0	0	0	0
0x144	DMAMUX_RGCFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COF3	COF2	COF1	COF0
	Reset value																													0	0	0	0
0x148 - 0x3FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

13 嵌套向量中断控制器 (NVIC)

13.1 NVIC 主要特性

CPU1 NVIC 特性：

- 63 个可屏蔽中断通道（不包括带 FPU 的 Cortex®-M4 的 16 根中断线）
- 16 个可编程优先级（使用了 4 位中断优先级）
- 低延迟异常中断处理
- 电源管理控制
- 系统控制寄存器的实现

CPU2 NVIC 特性：

- 32 个可屏蔽中断通道（不包括 16 根 Cortex®-M0+ 中断线）
- 4 个可编程优先级（使用了 2 位中断优先级）
- 低延迟异常中断处理
- 电源管理控制

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

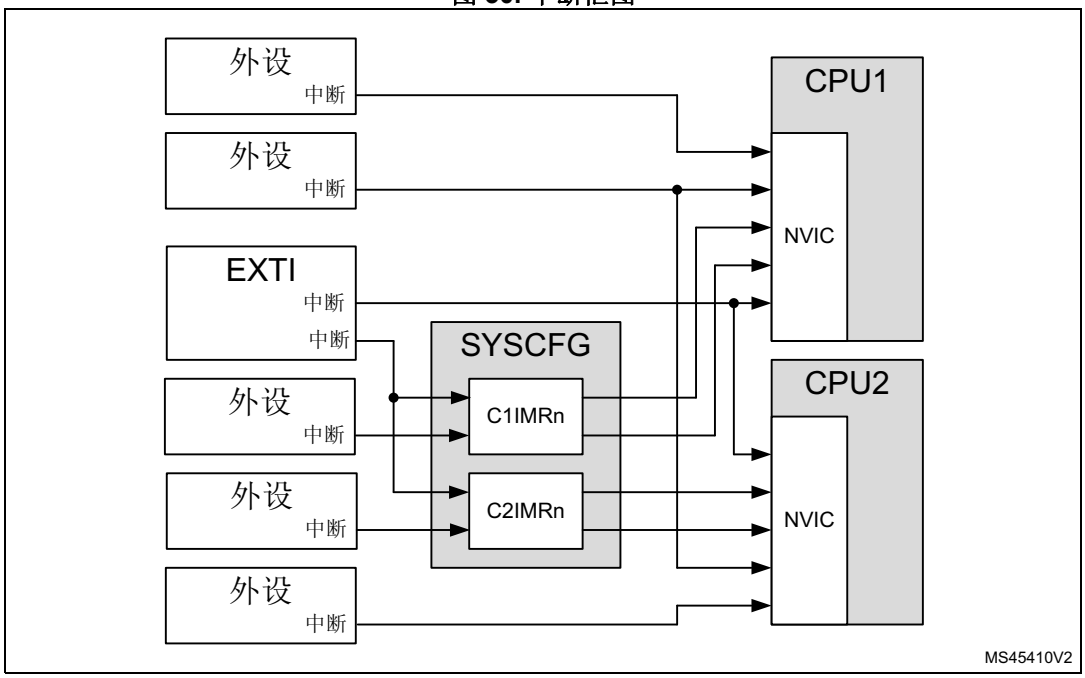
包括内核异常在内的所有中断均通过 NVIC 进行管理。有关异常和 NVIC 编程的更多信息，请参见 Cortex®-M4 的 PM0214 编程手册和 Cortex®-M0+ 的 PM056 编程手册。

13.2 中断框图

对于不同的外设中断，其连接方式也有所不同，具体取决于两个 CPU 之间的共享情况。为防止外设或 EXTI 中断触发两个 CPU，可以在 NVIC 中对其进行屏蔽，或者，对于 NVIC 向量共享多个外设中断的情况，可通过 SYSCFG 寄存器中的预屏蔽实现屏蔽，请参见*第 10 节：系统配置控制器 (SYSCFG)*。

中断框图如*图 30* 所示。

图 30. 中断框图



13.3 中断和异常向量

每个 CPU1 和 CPU2 都有自己的向量表，请分别参见*表 57* 和*表 58*，其中阴影单元格表示处理器异常。

表 57. STM32WB55xx CPU1 向量表

位置	优先级	优先级类型	缩略语	说明	地址
-	-	-	-	保留	0x0000 0000
-	-3	固定	复位	复位	0x0000 0004
-	-2	固定	NMI	不可屏蔽中断 HSE CSS、Flash ECC 和 SRAM2 奇偶校验	0x0000 0008
-	-1	固定	HardFault	所有类型的错误	0x0000 000C
-	0	可设置	MemManager	内存管理器	0x0000 0010
-	1	可设置	BusFault	预取指失败，存储器访问失败	0x0000 0014
-	2	可设置	UsageFault	未定义的指令或非法状态	0x0000 0018
-	-	-	-	保留	0x0000 001C 0x0000 0028

表 57. STM32WB55xx CPU1 向量表（续）

位置	优先级	优先级类型	缩略语	说明	地址
-	3	可设置	SVCall	通过 SWI 指令调用的系统服务	0x0000 002C
-	4	可设置	调试	调试监控器	0x0000 0030
-	-	-	-	保留	0x0000 0034
-	5	可设置	PendSV	可挂起的系统服务请求	0x0000 0038
-	6	可设置	Systick	系统节拍定时器	0x0000 003C
0	7	可设置	WWDG	窗口看门狗提前唤醒	0x0000 0040
1	8	可设置	PVD，PVM[2,0]	PVD（通过 EXTI[16]）(C1IMR2[20]) PVM[0]（通过 EXTI[31]）(C1IMR2[16]) PVM[2]（通过 EXTI[33]）(C1IMR2[18])	0x0000 0044
2	9	可设置	TAMP， RTC_STAMP， LSE_CSS	入侵，时间戳，LSECSS 中断（通过 EXTI[18]）	0x0000 0048
3	10	可设置	RTC_WKUP	RTC 唤醒中断（通过 EXTI[19]）	0x0000 004C
4	11	可设置	Flash	Flash 全局中断和 Flash ECC 单错误中断	0x0000 0050
5	12	可设置	RCC	RCC 全局中断	0x0000 0054
6	13	可设置	EXTI0	EXTI 线 0 中断（通过 EXTI[0]）	0x0000 0058
7	14	可设置	EXTI1	EXTI 线 1 中断（通过 EXTI[1]）	0x0000 005C
8	15	可设置	EXTI2	EXTI 线 2 中断（通过 EXTI[2]）	0x0000 0060
9	16	可设置	EXTI3	EXTI 线 3 中断（通过 EXTI[3]）	0x0000 0064
10	17	可设置	EXTI4	EXTI 线 4 中断（通过 EXTI[4]）	0x0000 0068
11	18	可设置	DMA1_CH1	DMA1 通道 1 中断	0x0000 006C
12	19	可设置	DMA1_CH2	DMA1 通道 2 中断	0x0000 0070
13	20	可设置	DMA1_CH3	DMA1 通道 3 中断	0x0000 0074
14	21	可设置	DMA1_CH4	DMA1 通道 4 中断	0x0000 0078
15	22	可设置	DMA1_CH5	DMA1 通道 5 中断	0x0000 007C
16	23	可设置	DMA1_CH6	DMA1 通道 6 中断	0x0000 0080
17	24	可设置	DMA1_CH7	DMA1 通道 7 中断	0x0000 0084
18	25	可设置	ADC1	ADC1 全局中断	0x0000 0088
19	26	可设置	USB_HP	USB 高优先级中断	0x0000 008C
20	27	可设置	USB_LP	USB 低优先级中断（包括 USB 唤醒）	0x0000 0090

表 57. STM32WB55xx CPU1 向量表（续）

位置	优先级	优先级类型	缩略语	说明	地址
21	28	可设置	C2SEV PWR_C2H	CPU2 SEV（通过 EXTI[40]） PWR CPU2 HOLD 唤醒中断	0x0000 0094
22	29	可设置	COMP	COMP2 和 COMP1 中断（通过 EXTI[21:20]）	0x0000 0098
23	30	可设置	EXTI[9:5]	EXTI 线 [9:5] 中断（通过 EXTI[9:5]）(C1IMR1[25:21])	0x0000 009C
24	31	可设置	TIM1_BRK	定时器 1 刹车中断	0x0000 00A0
25	32	可设置	TIM1_UP， TIM16	定时器 1 更新 (C1IMR1[13]) 定时器 16 全局中断 (C1IMR1[14])	0x0000 00A4
26	33	可设置	TIM1_TRG_COM， TIM17	定时器 1 触发和通信 (C1IMR1[13]) 定时器 17 全局中断 (C1IMR1[15])	0x0000 00A8
27	34	可设置	TIM1_CC	定时器 1 捕获比较中断	0x0000 00AC
28	35	可设置	TIM2	定时器 2 全局中断	0x0000 00B0
29	36	可设置	PKA	私钥加速器中断	0x0000 00B4
30	37	可设置	I2C1_EV	I2C1 事件中断	0x0000 00B8
31	38	可设置	I2C1_ER	I2C1 错误中断	0x0000 00BC
32	39	可设置	I2C3_EV	I2C3 事件中断	0x0000 00C0
33	40	可设置	I2C3_ER	I2C3 错误中断	0x0000 00C4
34	41	可设置	SPI1	SPI 1 全局中断	0x0000 00C8
35	42	可设置	SPI2	SPI 2 全局中断	0x0000 00CC
36	43	可设置	USART1	USART1 全局中断	0x0000 00D0
37	44	可设置	LPUART1	LPUART1 全局中断	0x0000 00D4
38	45	可设置	SAI1	SAI1 A 和 B 全局中断	0x0000 00D8
39	46	可设置	TSC	TSC 全局中断	0x0000 00DC
40	47	可设置	EXTI[15:10]	EXTI 线 [15:10] 中断（通过 EXTI[15:10]） (C1IMR1[31:26])	0x0000 00E0
41	48	可设置	RTC_ALARM	RTC 闹钟（A 和 B）中断（通过 EXTI[17]）	0x0000 00E4
42	49	可设置	CRS_IT	CRS 中断	0x0000 00E8
43	50	可设置	PWR_SOTF PWR_BLEACT PWR_802ACT PWR_RFPHASE	PWR 开启实时中断 PWR BLE 活动结束中断 PWR 802.15.4 活动结束中断 PWR 临界无线电阶段结束中断	0x0000 00EC

表 57. STM32WB55xx CPU1 向量表（续）

位置	优先级	优先级类型	缩略语	说明	地址
44	51	可设置	IPCC_C1_RX_IT	IPCC CPU1 RX 占用中断	0x0000 00F0
45	52	可设置	IPCC_C1_TX_IT	IPCC CPU1 TX 空闲中断	0x0000 00F4
46	53	可设置	HSEM	CPU1 的信号量中断 0	0x0000 00F8
47	54	可设置	LPTIM1	LPtimer 1 全局中断	0x0000 00FC
48	55	可设置	LPTIM2	LPtimer 2 全局中断	0x0000 0100
49	56	可设置	LCD	LCD 全局中断	0x0000 0104
50	57	可设置	QUADSPI	QUADSPI 全局中断	0x0000 0108
51	58	可设置	AES1	AES1 全局中断	0x0000 010C
52	59	可设置	AES2	AES2 全局中断	0x0000 0110
53	60	可设置	真 RNG	真随机数发生器中断	0x0000 0114
54	61	可设置	FPU	浮点单元中断	0x0000 0118
55	62	可设置	DMA2_CH1	DMA2 通道 1 中断	0x0000 011C
56	63	可设置	DMA2_CH2	DMA2 通道 2 中断	0x0000 0120
57	64	可设置	DMA2_CH3	DMA2 通道 3 中断	0x0000 0124
58	65	可设置	DMA2_CH4	DMA2 通道 4 中断	0x0000 0128
59	66	可设置	DMA2_CH5	DMA2 通道 5 中断	0x0000 012C
60	67	可设置	DMA2_CH6	DMA2 通道 6 中断	0x0000 0130
61	68	可设置	DMA2_CH7	DMA2 通道 7 中断	0x0000 0134
62	69	可设置	DMAMUX1_OVR	DMAMUX1 溢出中断	0x0000 0138

表 58. STM32WB55xx CPU2 向量表

位置	优先级	优先级类型	缩略语	说明	地址
-	-	-	-	保留	0x0000 0000
-	-3	固定	复位	复位	0x0000 0004
-14	-2	固定	NMI	不可屏蔽中断 HSE CSS、Flash ECC 和 SRAM2 奇偶校验	0x0000 0008
-13	-1	固定	HardFault	所有类型的错误	0x0000 000C
-	-	-	-	保留	0x0000 0010 0x0000 0028
-5	0	可设置	SVCall	通过 SWI 指令调用的系统服务	0x0000 002C
-	-	-	-	保留	0x0000 0030 0x0000 0034
-2	1	可设置	PendSV	可挂起的系统服务请求	0x0000 0038
-1	2	可设置	Systick	系统节拍定时器	0x0000 003C
0	3	可设置	-	保留	0x0000 0040
1	4	可设置	PVD，PVM[2,0]	PVD（通过 EXTI[16]）(C2IMR2[20]) PVM[0]（通过 EXTI[31]）(C2IMR2[16]) PVM[2]（通过 EXTI[33]）(C2IMR2[18])	0x0000 0044
2	5	可设置	RTC_WKUP， TAMP，RTC_STAMP LSE_CSS， RTC_ALARM	RTC 唤醒中断（通过 EXTI[19]）(C2IMR1[3]) 入侵，时间戳 LSECSS 中断（通过 EXTI[18]）(C2IMR1[0]) RTC 报警（A 和 B）中断（通过 EXTI[17]）(C2IMR1[4])	0x0000 0048
3	6	可设置	USB_HP，USB_LP CRS_IT	USB 高优先级中断，USB 低优先级中断（包括 USB 唤醒）， CRS 中断	0x0000 004C
4	7	可设置	RCC FLASH C1SEV	RCC 全局中断 (C2IMR1[5]) Flash 全局中断和 Flash ECC 单错误中断 (C2IMR1[6]) CPU1 SEV（通过 EXTI[41]）	0x0000 0050
5	8	可设置	EXTI[1:0]	EXTI 线 1:0 中断（通过 EXTI[1:0]）(C2IMR1[17:16])	0x0000 0054
6	9	可设置	EXTI[3:2]	EXTI 线 3:2 中断（通过 EXTI[3:2]）(C2IMR1[19:18])	0x0000 0058
7	10	可设置	EXTI[15:4]	EXTI 线 15:4 中断（通过 EXTI[15:4]）(C2IMR1[31:20])	0x0000 005C
8	11	可设置	TSC 802_IT0	TSC 全局中断 (C2IMR2[21]) 802.15.4 中断 0	0x0000 0060
9	12	可设置	DMA1_CH[3:1]	DMA1 通道 3:1 中断 (C2IMR2[2:0])	0x0000 0064
10	13	可设置	DMA1_CH[7:4]	DMA1 通道 7:4 中断 (C2IMR2[6:3])	0x0000 0068

表 58. STM32WB55xx CPU2 向量表（续）

位置	优先级	优先级类型	缩略语	说明	地址
11	14	可设置	DMA2_CH[7:1] DMAMUX1_OVR	DMA2 通道 7:1 中断 (C2IMR2[14:8]) DMAMUX1 溢出中断 (C2IMR2[15])	0x0000 006C
12	15	可设置	ADC1 COMP	ADC1 全局中断 (C2IMR1[12]) COMP1 和 COMP2 中断（通过 EXTI[21:20]）(C2IMR1[11])	0x0000 0070
13	16	可设置	LPTIM1	LPtimer 1 全局中断	0x0000 0074
14	17	可设置	LPTIM2	LPtimer 2 全局中断	0x0000 0078
15	18	可设置	TIM1_BRK， TIM1_UP， TIM1_TRG_COM， TIM1_CC	定时器 1 刹车，更新，触发和通信，捕获比较中断	0x0000 007C
16	19	可设置	TIM2	定时器 2 全局中断	0x0000 0080
17	20	可设置	TIM16	定时器 16 全局中断	0x0000 0084
18	21	可设置	TIM17	定时器 17 全局中断	0x0000 0088
19	22	可设置	IPCC_C2_RX_IT IPCC_C2_TX_IT HSEM	IPCC CPU2 RX 占用中断 IPCC CPU2 TX 空闲中断 CPU2 的信号量中断 1	0x0000 008C
20	23	可设置	PKA 真 RNG AES1	私钥加速器中断 (C2IMR1[8]) 真随机数发生器中断 (C2IMR1[9]) AES1 全局中断 (C2IMR1[10])	0x0000 0090
21	24	可设置	AES2	AES2 全局中断	0x0000 0094
22	25	可设置	LCD 802_IT1	LCD 全局中断 (C2IMR2[22]) 802.15.4 中断 1	0x0000 0098
23	26	可设置	I2C1_EV I2C1_ER	I2C1 事件中断 I2C1 错误中断	0x0000 009C
24	27	可设置	I2C3_EV I2C3_ER	I2C3 事件中断 I2C3 错误中断	0x0000 00A0
25	28	可设置	SPI1	SPI 1 全局中断	0x0000 00A4
26	29	可设置	SPI2	SPI 2 全局中断	0x0000 00A8
27	30	可设置	USART1	USART1 全局中断	0x0000 00AC
28	31	可设置	LPUART1	LPUART1 全局中断	0x0000 00B0
29	32	可设置	SAI1	SAI1 A 和 B 全局中断	0x0000 00B4
30	33	可设置	BLE_BLUE_IT BLE_RFC_IT BLE_RFFMS_IT BLE_HOST_WKUP	BLE 蓝牙控制器中断 BLE 无线电控制中断 BLE 无线电状态中断 BLE 主机唤醒中断	0x0000 00B8
31	34	可设置	802_IT2 802_HOST_WKUP	802.15.4 中断 2 802.15.4 主机唤醒中断	0x0000 00BC

13.4 中断列表

表 59 中列出了不同的 STM32WB55xx 唤醒源。根据其来源，基于不同类型对唤醒进行处理，有关更多信息，请参见*第 14.4 节：EXTI 功能行为*。

某些唤醒源能够为 CPU 生成事件。请参见*事件*列。

*唤醒*列给出了唤醒 CPU1 和/或 CPU2 的唤醒源能力。

有关 CPU 中断处理，请参见*第 13 节：嵌套向量中断控制器 (NVIC)*。

有关唤醒处理，请参见*第 14 节：扩展中断和事件控制器 (EXTI)*。

表 59. STM32WB55xx 唤醒中断表

EXTI 编号	缩略语	说明	EXTI 类型	事件	唤醒
0	EXTI[0]	EXTI 线 0（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
1	EXTI[1]	EXTI 线 1（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
2	EXTI[2]	EXTI 线 2（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
3	EXTI[3]	EXTI 线 3（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
4	EXTI[4]	EXTI 线 4（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
5	EXTI[5]	EXTI 线 5（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
6	EXTI[6]	EXTI 线 6（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
7	EXTI[7]	EXTI 线 7（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
8	EXTI[8]	EXTI 线 8（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
9	EXTI[8]	EXTI 线 9（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
10	EXTI[10]	EXTI 线 10（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
11	EXTI[11]	EXTI 线 11（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
12	EXTI[12]	EXTI 线 12（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
13	EXTI[13]	EXTI 线 13（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
14	EXTI[14]	EXTI 线 14（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
15	EXTI[15]	EXTI 线 15（来自 SYSCFG）	可配置 A	是	CPU1 和 CPU2
16	PVD	PVD 线	可配置 A	否	CPU1 和 CPU2
17	RTC_ALARM	RTC 报警（A 和 B）中断	可配置 A	是	CPU1 和 CPU2
18	TAMP， RTC_STAMP， LSE_CSS	RTC 入侵中断 RTC 时间戳中断 RCC LSECSS 中断	可配置 A	是	CPU1 和 CPU2
19	RTC_WKUP	RTC 唤醒中断	可配置 A	是	CPU1 和 CPU2
20	COMP1	COMP1 线	可配置 A	是	CPU1 和 CPU2
21	COMP2	COMP2 线	可配置 A	是	CPU1 和 CPU2
22	I2C1 唤醒	I2C1 唤醒	直接 B	否	CPU1 和 CPU2
23	I2C3 唤醒	I2C3 唤醒	直接 B	否	CPU1 和 CPU2

表 59. STM32WB55xx 唤醒中断表（续）

EXTI 编号	缩略语	说明	EXTI 类型	事件	唤醒
24	USART1	USART1 唤醒	直接 B	否	CPU1 和 CPU2
25	LPUART1	LPUART1 唤醒	直接 B	否	CPU1 和 CPU2
26	保留	-	-	-	-
27	保留	-	-	-	-
28	USB 唤醒	USB 唤醒	直接 B	否	CPU1 和 CPU2
29	LPTIM1 唤醒	LPtimer 1 唤醒	直接 B	否	CPU1 和 CPU2
30	LPTIM2 唤醒	LPtimer 2 唤醒	直接 B	否	CPU1 和 CPU2
31	PVM[1]	PVM[1] 线	可配置 A	否	CPU1 和 CPU2
32	保留	-	-	-	-
33	PVM[3]	PVM[3] 线	可配置 A	否	CPU1 和 CPU2
34	保留	-	-	-	-
35	保留	-	-	-	-
36	IPCC CPU1 中断	IPCC CPU1 RX 占用和 TX 空闲中断	直接 C	否	CPU1(1)
37	IPCC CPU2 中断	IPCC CPU2 RX 占用和 TX 空闲中断	直接 C	否	CPU2(2)
38	HSEM 中断 0	CPU1 的信号量中断 0	直接 C	否	CPU1(1)
39	HSEM 中断 1	CPU2 的信号量中断 1	直接 C	否	CPU2(2)
40	C2SEV	CPU2 SEV 线	可配置 A	是	CPU1(3)
41	C1SEV	CPU1 SEV 线	可配置 A	是	CPU2(4)
42	Flash 中断	Flash ECC 和全局中断	直接 C	否	CPU1 和 CPU2
43	LCD 唤醒	LCD 唤醒	直接 B	否	CPU1 和 CPU2
44	HSE CSS 中断	RCC HSE CSS 中断	直接 C	否	CPU1 和 CPU2
45	BLE 中断	BLE，RADIO 和 RF_FSM 中断	直接 B	否	CPU2(2)
46	802 中断	802.15.4 Int0 和 Int1 中断	直接 B	否	CPU2(2)
47	保留	-	-	-	-
48	CDBGPWRUPREQ	调试上电请求唤醒	直接	否	CPU1 和 CPU2

1. 为正确运行，应先将 EXTI 直接事件 C2IMRx.IMn 位设为 0，然后 CPU1 才使用此直接事件。
2. 为正确运行，应先将 EXTI 直接事件 C1IMRx.IMn 位设为 0，然后 CPU2 才使用此直接事件。
3. 为正确运行，应先将 EXTI 可配置事件 C2IMRx.IMn 和 C2EMRx.EMn 位设为 0，然后 CPU1 才使用此可配置事件。
4. 为正确运行，应先将 EXTI 可配置事件 C1IMRx.IMn 和 C1EMRx.EMn 位设为 0，然后 CPU2 才使用此可配置事件。

14 扩展中断和事件控制器 (EXTI)

扩展中断和事件控制器 (EXTI) 通过可配置的事件输入和直接事件输入来管理单独的 CPU 和系统唤醒。它可以针对电源控制提供唤醒请求、针对 CPU NVIC 生成中断请求，以及针对 CPU 事件输入生成事件。对于每个 CPU，均需通过额外的事件生成模块 (EVG) 来生成 CPU 事件信号。

EXTI 唤醒请求可让系统从停止模式唤醒，以及让 CPU 从 CSTOP 和 CSTANDBY 模式唤醒。

此外，还可以在运行模式下生成中断请求和事件请求。

14.1 EXTI 主要特性

EXTI 的主要特性如下：

- 支持 49 个输入事件。
- 支持 2 个 CPU。
- 所有事件输入都允许唤醒系统。
- 对于在外设中没有相关唤醒标志的事件，均在 EXTI 中有一个标志，并且会从 EXTI 生成一个 CPU 中断。
- 一些事件可用于生成 CPU 唤醒事件。

异步事件输入分为 2 组：

- 可配置事件（来自能够生成脉冲的 I/O 或外设的信号）
  - 可配置事件具有以下特性：
    - 可选择的有效触发边沿
    - 中断挂起状态寄存器位
    - 单独的中断和事件生成掩码，用于调节 CPU 唤醒、中断和事件生成
    - 支持软件触发
- 直接事件（来自具有相关标志的外设的中断和唤醒源，需要在外设中清除）
  - 直接事件具有以下特性：
    - 固定上升沿有效触发
    - EXTI 中无中断挂起状态寄存器位（中断挂起状态标志由生成事件的外设提供）
    - 单独的中断和事件生成掩码，用于调节 CPU 唤醒和事件生成
    - 不支持软件触发

14.2 EXTI 框图

EXTI 包含一个通过 AHB 接口访问的寄存器模块、一个事件输入触发模块和一个屏蔽模块，如*图 31* 所示。

寄存器模块包含所有 EXTI 寄存器。

事件输入触发模块提供事件输入边沿触发逻辑。

屏蔽模块为不同的唤醒、中断和事件输出及其屏蔽功能提供事件输入分配。

图 31. EXTI 框图

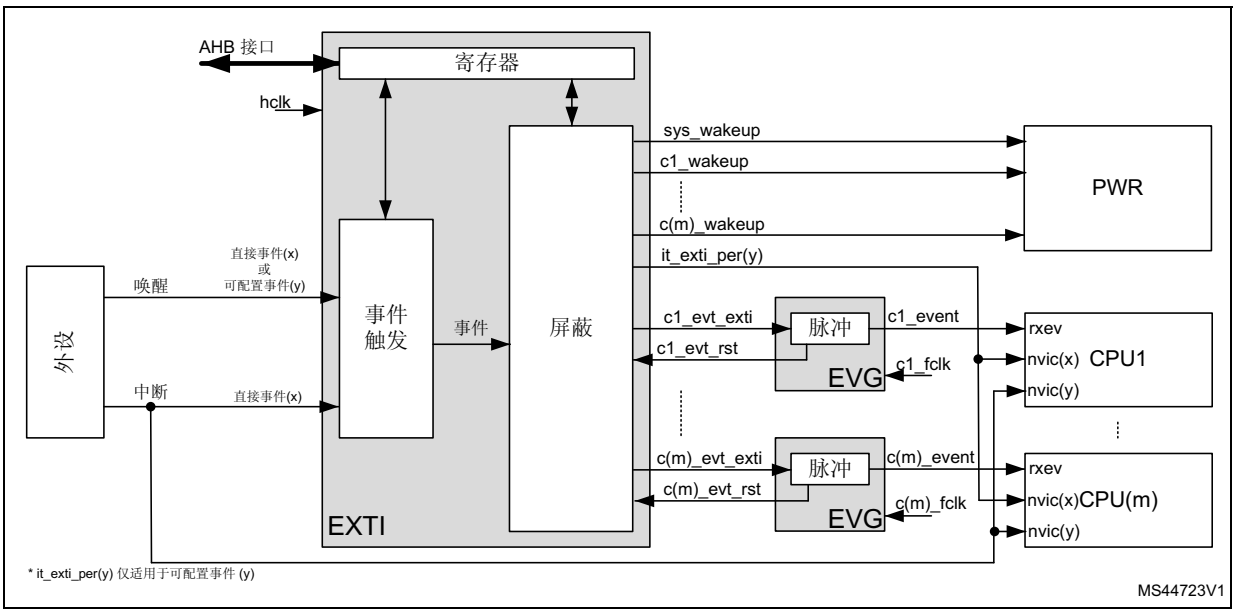


表 60. EXTI 引脚概述

引脚名称	I/O	说明
AHB 接口	I/O	EXTI 寄存器总线接口。当将一个事件配置为允许安全性时，AHB 接口支持安全访问。
hclk	I	AHB 总线时钟和 EXTI 系统时钟。
可配置事件 (y)	I	来自没有相关中断和标志的外设的异步唤醒事件。
直接事件 (x)	I	来自具有相关中断和标志的外设的同步与异步唤醒事件。
it_exti_per (y)	O	与可配置事件 (y) 关联的 CPU1 到 CPU(m) 的中断。
c(m)_evt_exti	O	CPU(m) 的高电平触发事件输出，与 hclk 同步。（m = 1 到 2）
c(m)_evt_rst	I	用于清除 c(m)_evt_exti 的异步复位输入。（m = 1 到 2）
sys_wakeup	O	面向 ck_sys 和 hclk 的 PWR 的异步系统唤醒请求。
c(m)_wakeup	O	面向 CPU(m) 的 PWR 的唤醒请求，与 hclk 同步。（m = 1 到 2）

表 61. EVG 引脚概述

引脚名称	I/O	说明
c_fclk	I	CPU 自由运行时钟。
c_evt_in	I	来自 EXTI 的高电平触发事件输入，与 CPU 时钟异步。
c_event	O	事件脉冲，与 CPU 时钟同步。
c_evt_rst	O	事件复位信号，与 CPU 时钟同步。

14.2.1 外设和 CPU 之间的 EXTI 连接

对于能够在系统处于停止模式或 CPU 处于 CSTOP 模式时生成唤醒或中断事件的外设，将连接至 EXTI。

- 对于生成脉冲或在外设中没有中断状态位的外设唤醒信号，将连接至 EXTI 可配置事件输入。对于这类事件，EXTI 提供的状态挂起位需清零。与状态位相关的 EXTI 中断将中断 CPU。
- 对于在外设中有状态位（需要在外设中清零）的外设中断和唤醒信号，将连接至 EXTI 直接事件输入。EXTI 中无状态挂起位。中断或唤醒由外设中的 CPU 清除。外设中断会直接中断 CPU。

EXTI 可配置事件中断连接到各 CPU(m) 的相应 NVIC(a)。

专用 EXTI/EVG CPU(m) 事件连接至相应的 CPU(m) rxev 输入。

EXTI CPU(m) 唤醒信号连接至 PWR 模块，用于唤醒系统和 CPU(m) 子系统总线时钟。

14.3 EXTI 功能说明

根据 EXTI 事件输入类型和唤醒目标，将使用不同的逻辑实现。适用的功能通过寄存器位进行控制：

- 有效触发边沿使能，具体包括上升沿选择
  EXTI 上升沿触发选择寄存器 (EXTI_RTSR1)、
  EXTI 上升沿触发选择寄存器 (EXTI_RTSR2)，
  和下降沿选择
  EXTI 下降沿触发选择寄存器 (EXTI_FTSR1)、
  EXTI 下降沿触发选择寄存器 (EXTI_FTSR2)。
- 软件触发，具体包括
  EXTI 软件中断事件寄存器 (EXTI_SWIER1)、
  EXTI 软件中断事件寄存器 (EXTI_SWIER2)。
- 中断挂起标志，具体包括
  EXTI 挂起寄存器 (EXTI_PR1)、
  EXTI 挂起寄存器 (EXTI_PR2)。
- CPU 唤醒和中断使能，具体包括
  EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR1)、
  EXTI CPU2 唤醒中断屏蔽寄存器 (EXTI_C2IMR1)、
  EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR2)、
  EXTI CPU2 唤醒中断屏蔽寄存器 (EXTI_C2IMR2)。
- CPU 唤醒和事件使能，具体包括
  EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR1)、
  EXTI CPU2 唤醒事件屏蔽寄存器 (EXTI_C2EMR1)、
  EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR2)、
  EXTI CPU2 唤醒事件屏蔽寄存器 (EXTI_C2EMR2)。

表 62. EXTI 事件输入配置和寄存器控制

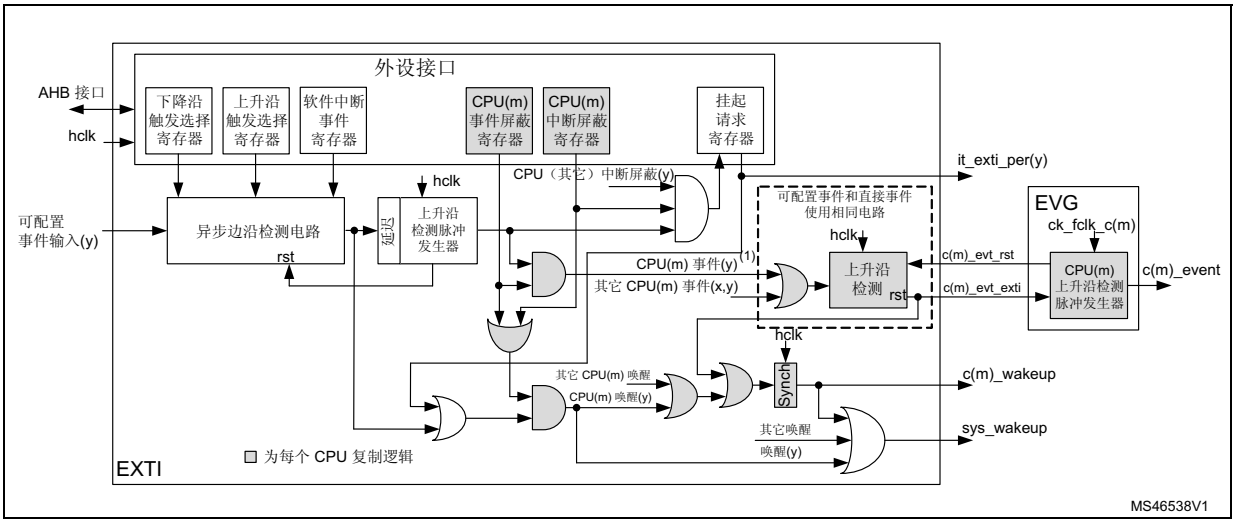
事件输入类型	逻辑实现	EXTI_RTSR	EXTI_FTSR	EXTI_SWIER	EXTI_PR	EXTI_CmIMR	EXTI_CmEMR(1)
可配置	可配置事件输入唤醒逻辑	x	x	x	x	x	x
直接	直接事件输入唤醒逻辑	-	-	-	-	x	x

1. 仅适用于使能了“rxev 生成”配置的输入事件。

14.3.1 EXTI 可配置事件输入唤醒

图 32 所示为具有如下用途的可配置事件输入的相关逻辑详图：唤醒 CPU(m) 子系统总线时钟并生成 CPU(m) 的 EXTI 挂起标志与中断和/或 CPU(m) 唤醒事件。

图 32. 可配置事件触发逻辑 CPU 唤醒



1. 仅适用于支持 CPU rxev 生成 c(n)_event 的输入事件。

软件中断事件寄存器允许通过软件写入相应的寄存器位来触发可配置事件，而边沿选择设置无关紧要。

上升沿和下降沿选择寄存器允许使能和选择可配置事件有效触发边沿或两种边沿。

每个 CPU 都有专用的中断屏蔽寄存器和专用的事件屏蔽寄存器。使能的事件允许在 CPU 上生成一个事件。CPU 的所有事件均在一个 CPU 事件信号中进行逻辑或运算。事件挂起寄存器 (EXTI_PR) 不会针对未屏蔽的 CPU 事件而置位。

可配置事件具有唯一的中断挂起请求寄存器（由各 CPU 共享）。挂起寄存器只会针对未屏蔽的中断而置位。每个可配置事件均提供对所有 CPU 通用的中断。可配置事件中断需要由软件通过 EXTI_PR 寄存器确认。

使能 CPU(m) 中断或 CPU(m) 事件时，异步边沿检测电路由时钟延迟和上升沿检测脉冲发生器复位。这可以确保 EXTI hclk 时钟在异步边沿检测电路复位之前唤醒。

注： *检测到的可配置事件中断挂起请求可由任何 CPU 清除。只要中断挂起请求处于活动状态，系统就无法进入低功耗模式。*

14.3.2 EXTI 直接事件输入唤醒

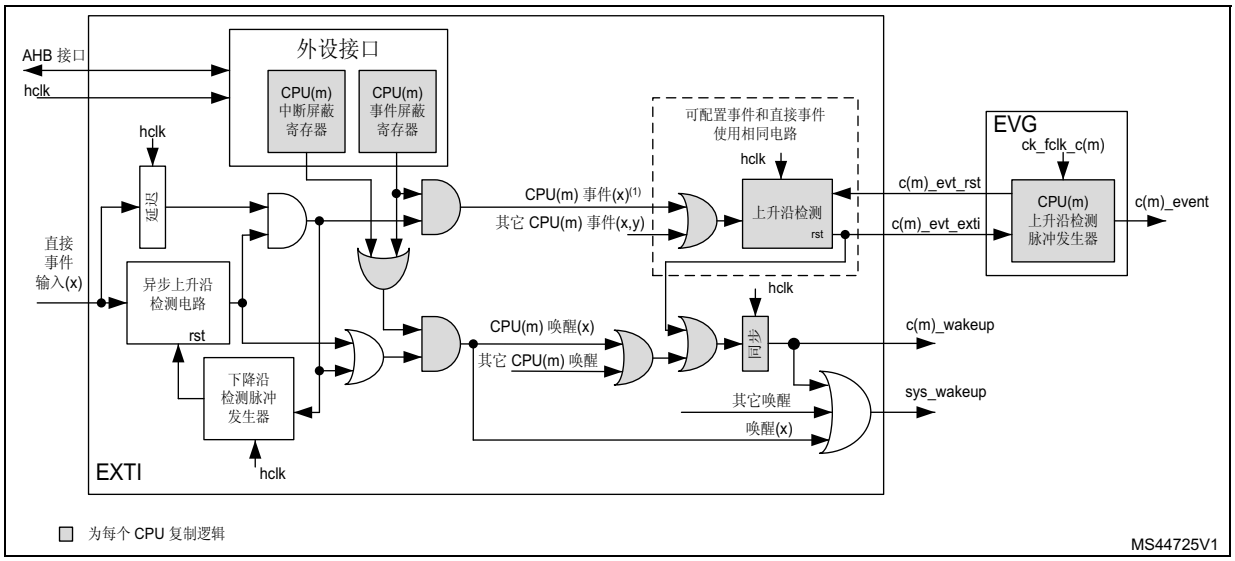
图 33 所示为唤醒系统的直接事件输入的相关逻辑详图。

直接事件没有关联的 EXTI 中断。EXTI 仅唤醒系统和 CPU 子系统时钟，并可能生成 CPU 唤醒事件。与直接唤醒事件相关的外设同步中断会唤醒 CPU。

EXTI 直接事件能够生成 CPU 事件。此 CPU 事件会唤醒 CPU。CPU 事件可能在相关外设中断标志置位之前发生。

注： *在生成事件的外设中清除直接事件。如果 CPU(m) 使能的直接事件输入在 CPU(m) 时钟运行之前被其他 CPU 清除，则 CPU(m) 将不再接收 CPU(m) 中断或 CPU(m) 事件，而且也不会唤醒。但是，系统仍将保持运行模式，进而生成 CPU(m) 时钟。因此，CPU(m) 直接事件不应被其他 CPU 清除。*

图 33. 直接事件触发逻辑 CPU 唤醒



1. 仅适用于支持 CPU rxev 生成 c(n)_event 的输入事件。

14.4 EXTI 功能行为

在生成唤醒事件的相应外设中使能直接事件输入。通过使能至少一个触发沿来使能可配置事件。

使能事件输入后，是否生成 CPU(m) 唤醒取决于 CPU(m) 中断屏蔽和 CPU(m) 事件屏蔽。

表 63. 屏蔽功能

CPU 中断使能 EXTI_CmIMR.IMn	CPU 事件使能 EXTI_CmEMR.EMn	可配置事件输入 EXTI_PR.PIFn	exti(n) 中断(1)	CPU(m) 事件	CPU(m) 唤醒
0（对于所有 CPU）	0	无	已屏蔽	已屏蔽	已屏蔽
	1	无	已屏蔽	有	有
1（对于所有 CPU）	0	状态已锁存	有	已屏蔽	有(2)
	1	状态已锁存	有	有	有

1. 单个 exti(n) 中断将面向所有 CPU。如果 CPU(m) 不需要中断，则应在 CPU(m) NVIC 中屏蔽 exti(n) 中断。
2. 仅限 EXTI_CmIMR.IMn 中已使能 CPU(m) 中断的情况。

对于可配置事件输入，当事件输入上出现使能的边沿时，会生成一个事件请求。当相关 CPU(m) 中断未被屏蔽时，相应的挂起位 EXTI_PR.PIFn 将置 1，CPU(m) 子系统会唤醒，CPU 中断信号将激活。EXTI_PR.PIFn 挂起位应通过软件写入“1”来清零。这会清除 CPU 中断。

对于直接事件输入，当在相关外设中使能时，只会在上升沿生成事件请求。EXTI 中没有相应的 CPU 挂起位。当关联 CPU(m) 中断取消屏蔽时，相应的 CPU 子系统将唤醒。CPU 将由外设同步中断唤醒（中断）。

要生成事件，CPU(m) 事件必须处于未屏蔽状态。当事件输入上出现使能的边沿时，会生成 CPU(m) 事件脉冲。不存在事件挂起位。

对于可配置事件输入，可通过软件在软件中断/事件寄存器 EXTI_SWIER 中写入“1”来生成事件请求，从而允许基于事件生成上升沿。边沿事件挂起位将在 EXTI_PR 中置 1，与 EXTI_RTSR 中的设置无关。

14.5 EXTI 寄存器

EXTI 寄存器映射分为以下几个部分：

表 64. EXTI 寄存器映射的各个部分

地址	说明
0x000 - 0x01C	通用可配置事件 [31:0] 配置
0x020 - 0x03C	通用可配置事件 [63:32] 配置
0x040 - 0x05C	通用可配置事件 [95:64] 配置
0x080 - 0x0BC	CPU1 输入事件配置
0x0C0 - 0x0FC	CPU2 输入事件配置

所有寄存器均可支持按字（32 位）、半字（16 位）或字节（8 位）访问。

14.5.1 EXTI 上升沿触发选择寄存器 (EXTI_RTSR1)

EXTI rising trigger selection register

偏移地址：0x000

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RT31	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RT21	RT20	RT19	RT18	RT17	RT16
rw										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **RT31**：可配置事件输入 31 的上升沿触发事件配置位 (Rising trigger event configuration bit of configurable Event input 31)(1)

0：禁止输入线上升沿触发（事件和中断）
1：允许输入线上升沿触发（事件和中断）

位 30:22 保留，必须保持复位值。

位 21:0 **RT[21:0]**：可配置事件输入 x 的上升沿触发事件配置位 (Rising trigger event configuration bit of configurable event input x)（x = 21 到 0）(1)

0：禁止输入线上升沿触发（事件和中断）
1：允许输入线上升沿触发（事件和中断）

1. 可配置事件输入为边沿触发，在这些输入上不能出现毛刺信号。
如果在对寄存器执行写操作时可配置事件输入上出现上升沿，则相关挂起位不会被置 1。
在同一可配置事件输入上，可同时设置上升沿和下降沿触发。在这种情况下，两种边沿均会生成触发信号。

14.5.2 EXTI 下降沿触发选择寄存器 (EXTI_FTSR1)

EXTI falling trigger selection register

偏移地址：0x004

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FT31	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT21	FT20	FT19	FT18	FT17	FT16
rw										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **FT31**：可配置事件输入 31 的下降沿触发事件配置位 (Falling trigger event configuration bit of configurable event input 31)(1)

0：禁止输入线下降沿触发（事件和中断）
1：允许输入线下降沿触发（事件和中断）

位 30:22 保留，必须保持复位值。

位 21:0 **FT[21:0]**：可配置事件输入 x 的下降沿触发事件配置位 (Falling trigger event configuration bit of configurable event input x)（x = 21 到 0）(1)

0：禁止输入线下降沿触发（事件和中断）
1：允许输入线下降沿触发（事件和中断）

1. 可配置事件输入为边沿触发，在这些输入上不能出现毛刺信号。
如果在对寄存器执行写操作时可配置事件输入上出现下降沿，则相关挂起位不会被置 1。
在同一可配置事件输入上，可同时设置上升沿和下降沿触发。在这种情况下，两种边沿均会生成触发信号。

14.5.3 EXTI 软件中断事件寄存器 (EXTI_SWIER1)

EXTI software interrupt event register

偏移地址：0x008

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWI 31	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWI 21	SWI 20	SWI 19	SWI 18	SWI 17	SWI 16
rw										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWI 15	SWI 14	SWI 13	SWI 12	SWI 11	SWI 10	SWI 9	SWI 8	SWI7	SWI6	SWI5	SWI4	SWI3	SWI2	SWI 1	SWI 0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **SWI31**：事件 31 上的软件中断 (Software interrupt on event 31)
生成的软件中断与 EXTI_RTSR 和 EXTI_FTSR 中的设置无关。读取时将始终返回 0。

0：写入 0 无影响。
1：向此位写入 1 会触发事件 31 上的上升沿事件。此位由硬件自动清零。

位 30:22 保留，必须保持复位值。

位 21:0 **SWI[21:0]**：事件 x 上的软件中断 (Software interrupt on event x)（x = 21 到 0）
生成的软件中断与 EXTI_RTSR 和 EXTI_FTSR 中的设置无关。读取时将始终返回 0。

0：写入 0 无影响。
1：向此位写入 1 会触发事件 x 上的上升沿事件。此位由硬件自动清零。

14.5.4 EXTI 挂起寄存器 (EXTI_PR1)

EXTI pending register

偏移地址：0x00C

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PIF31	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PIF21	PIF20	PIF19	PIF18	PIF17	PIF16
rc_w1										rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIF15	PIF14	PIF13	PIF12	PIF11	PIF10	PIF9	PIF8	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31 **PIF31**：可配置事件输入 31 挂起位 (configurable event inputs 31 Pending bit)
0：未发生触发请求
1：发生了选择的触发请求
当在可配置事件线上发生了选择的边沿事件或 EXTI_SWIER 软件触发时，此位将置 1。向此位写入 1 可将其清零。

位 30:22 保留，必须保持复位值。

位 21:0 **PIF[21:0]**：可配置事件输入 x 挂起位 (configurable event inputs x Pending bit)（x = 21 到 0）
0：未发生触发请求
1：发生了选择的触发请求
当在可配置事件线上发生了选择的边沿事件或 EXTI_SWIER 软件触发时，此位将置 1。向此位写入 1 可将其清零。

14.5.5 EXTI 上升沿触发选择寄存器 (EXTI_RTSR2)

EXTI rising trigger selection register

偏移地址：0x020

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	RT41	RT40	Res.	Res.	Res.	Res.	Res.	Res.	RT33	Res.
						rw	rw							rw	

位 31:10 保留，必须保持复位值。

位 9 **RT41**：可配置事件输入 41 的上升沿触发事件配置位 (1) (Rising trigger event configuration bit of configurable event input 41)
0：禁止输入线上升沿触发（事件和中断）
1：允许输入线上升沿触发（事件和中断）

位 8 **RT40**：可配置事件输入 40 的上升沿触发事件配置位 (1) (Rising trigger event configuration bit of configurable event input 40)
0：禁止输入线上升沿触发（事件和中断）
1：允许输入线上升沿触发（事件和中断）

位 7:2 保留，必须保持复位值。

位 1 **RT33**：可配置事件输入 33 的上升沿触发事件配置位 (Rising trigger event configuration bit of configurable event input 33)(1)

0：禁止输入线上升沿触发（事件和中断）
1：允许输入线上升沿触发（事件和中断）

位 0 保留，必须保持复位值。

1. 可配置事件输入为边沿触发，在这些输入上不能出现毛刺信号。
如果在对寄存器执行写操作时可配置事件输入上出现上升沿，则相关挂起位不会被置 1。
在同一可配置事件输入上，可同时设置上升沿和下降沿触发。在这种情况下，两种边沿均会生成触发信号。

14.5.6 EXTI 下降沿触发选择寄存器 (EXTI_FTSR2)

EXTI falling trigger selection register

偏移地址：0x024

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	FT41	FT40	Res.	Res.	Res.	Res.	Res.	Res.	FT33	Res.
						rw	rw							rw	

位 31:10 保留，必须保持复位值。

位 9 **FT41**：可配置事件输入 41 的下降沿触发事件配置位 (1) (Falling trigger event configuration bit of configurable event input 41)

0：禁止输入线下降沿触发（事件和中断）
1：允许输入线下降沿触发（事件和中断）

位 8 **FT40**：可配置事件输入 40 的下降沿触发事件配置位 (1) (Falling trigger event configuration bit of configurable event input 40)

0：禁止输入线下降沿触发（事件和中断）
1：允许输入线下降沿触发（事件和中断）

位 7:2 保留，必须保持复位值。

位 1 **FT33**：可配置事件输入 33 的下降沿触发事件配置位 (Falling trigger event configuration bit of configurable event input 33)(1)

0：禁止输入线下降沿触发（事件和中断）
1：允许输入线下降沿触发（事件和中断）

位 0 保留，必须保持复位值。

1. 可配置事件输入为边沿触发，在这些输入上不能出现毛刺信号。
如果在对寄存器执行写操作时可配置事件输入上出现下降沿，则相关挂起位不会被置 1。
在同一可配置事件输入上，可同时设置上升沿和下降沿触发。在这种情况下，两种边沿均会生成触发信号。

14.5.7 EXTI 软件中断事件寄存器 (EXTI_SWIER2)

EXTI software interrupt event register

偏移地址：0x028

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	SWI 41	SWI 40	Res.	Res.	Res.	Res.	Res.	Res.	SWI 33	Res.
						rw	rw							rw	

位 31:10 保留，必须保持复位值。

位 9 **SWI41**：事件 41 上的软件中断 (Software interrupt on event 41)
生成的软件中断与 EXTI_RTSR 和 EXTI_FTSR 中的设置无关。读取时将始终返回 0。
0：写入 0 无影响。
1：向此位写入 1 会触发事件 41 上的上升沿事件。此位由硬件自动清零。

位 8 **SWI40**：事件 40 上的软件中断 (Software interrupt on event 40)
生成的软件中断与 EXTI_RTSR 和 EXTI_FTSR 中的设置无关。读取时将始终返回 0。
0：写入 0 无影响。
1：向此位写入 1 会触发事件 40 上的上升沿事件。此位由硬件自动清零。

位 7:2 保留，必须保持复位值。

位 1 **SWI33**：事件 33 上的软件中断 (Software interrupt on event 33)
生成的软件中断与 EXTI_RTSR 和 EXTI_FTSR 中的设置无关。读取时将始终返回 0。
0：写入 0 无影响。
1：向此位写入 1 会触发事件 33 上的上升沿事件。此位由硬件自动清零。

位 0 保留，必须保持复位值。

14.5.8 EXTI 挂起寄存器 (EXTI_PR2)

EXTI pending register

偏移地址：0x02C

复位值：0x0000 0000

仅包含可配置事件的寄存器位。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	PIF41	PIF40	Res.	Res.	Res.	Res.	Res.	Res.	PIF33	Res.
						rc_w1	rc_w1							rc_w1	

位 31:10 保留，必须保持复位值。

位 9 **PIF41**：可配置事件输入 41 挂起位 (configurable event inputs 41 pending bit)

0：未发生触发请求
1：发生了选择的触发请求
当在可配置事件线上发生了选择的边沿事件或 EXTI_SWIER 软件触发时，此位将置 1。向此位写入 1 可将其清零。

位 8 **PIF40**：可配置事件输入 40 挂起位 (configurable event inputs 40 pending bit)

0：未发生触发请求
1：发生了选择的触发请求
当在可配置事件线上发生了选择的边沿事件或 EXTI_SWIER 软件触发时，此位将置 1。向此位写入 1 可将其清零。

位 7:2 保留，必须保持复位值。

位 1 **PIF33**：可配置事件输入 33 挂起位 (configurable event inputs 33 pending bit)

0：未发生触发请求
1：发生了选择的触发请求
当在可配置事件线上发生了选择的边沿事件或 EXTI_SWIER 软件触发时，此位将置 1。向此位写入 1 可将其清零。

位 0 保留，必须保持复位值。

14.5.9 EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR1)

EXTI CPU wakeup with interrupt mask register

偏移地址：0x080

复位值：0x7FC0 0000

包含可配置事件和直接事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IM31	IM30	IM29	IM28	IM27	IM26	IM25	IM24	IM23	IM22	IM21	IM20	IM19	IM18	IM17	IM16
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
IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IM[31:0]**：屏蔽事件输入 x 上的 CPU 唤醒中断 (CPU wakeup with interrupt mask on event input x)（x = 31 到 0）(1)(2)

0：屏蔽来自线 x 的唤醒中断请求
1：取消屏蔽来自线 x 的唤醒中断请求

1. 可配置事件输入的复位值置“0”，以在默认情况下禁止中断。
2. 直接事件输入的复位值置“1”，以在默认情况下使能中断。

14.5.10 EXTI CPU2 唤醒中断屏蔽寄存器 (EXTI_C2IMR1)

EXTI CPU2 wakeup with interrupt mask register

偏移地址：0x0C0

复位值：0x7FC0 0000

包含可配置事件和直接事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IM31	IM30	IM29	IM28	IM27	IM26	IM25	IM24	IM23	IM22	IM21	IM20	IM19	IM18	IM17	IM16
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
IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IM[31:0]**：屏蔽事件输入 x 上的 CPU2 唤醒中断 (CPU2 wakeup with interrupt mask on event input x)（x = 31 到 0）(1)(2)

0：屏蔽来自线 x 的唤醒中断请求

1：取消屏蔽来自线 x 的唤醒中断请求

1. 可配置事件输入的复位值置“0”，以在默认情况下禁止中断。
2. 直接事件输入的复位值置“1”，以在默认情况下使能中断。

14.5.11 EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR1)

EXTI CPU wakeup with event mask register

偏移地址：0x084

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EM21	EM20	EM19	EM18	EM17	Res.
										rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EM15	EM14	EM13	EM12	EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:17 **EM[21:17]**：屏蔽事件输入 x 上的 CPU 唤醒事件生成 (CPU wakeup with event generation mask on event input x)（x = 21 到 17）

0：屏蔽线 x 上的唤醒事件生成

1：取消屏蔽线 x 上的唤醒事件生成

位 16 保留，必须保持复位值。

位 15:0 **EM[15:0]**：屏蔽事件输入 x 上的 CPU 唤醒事件生成 (CPU wakeup with event generation mask on event input x)（x = 15 到 0）

0：屏蔽线 x 上的唤醒事件生成

1：取消屏蔽线 x 上的唤醒事件生成

14.5.12 EXTI CPU2 唤醒事件屏蔽寄存器 (EXTI_C2EMR1)

EXTI CPU2 wakeup with event mask register

偏移地址：0x0C4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EM21	EM20	EM19	EM18	EM17	Res.
										rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EM15	EM14	EM13	EM12	EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:17 **EM[21:17]**：屏蔽事件输入 x 上的 CPU2 唤醒事件生成 (CPU2 wakeup with event generation mask on event input x)（x = 21 到 17）

0：屏蔽线 x 上的唤醒事件生成
1：取消屏蔽线 x 上的唤醒事件生成

位 16 保留，必须保持复位值。

位 15:0 **EM[15:0]**：屏蔽事件输入 x 上的 CPU2 唤醒事件生成 (CPU2 wakeup with event generation mask on event input x)（x = 15 到 0）

0：屏蔽线 x 上的唤醒事件生成
1：取消屏蔽线 x 上的唤醒事件生成

14.5.13 EXTI CPU 唤醒中断屏蔽寄存器 (EXTI_IMR2)

EXTI CPU wakeup with interrupt mask register

偏移地址：0x090

复位值：0x0001 FCFD

包含可配置事件和直接事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IM48
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IM47	IM46	IM45	IM44	IM43	IM42	IM41	IM40	IM39	IM38	IM37	IM36	IM35	IM34	IM33	IM32
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:0 **IM[48:32]**：屏蔽事件输入 x 上的 CPU 唤醒中断 (CPU wakeup with interrupt mask on event input x)（x = 48 到 32）(1)(2)

0：屏蔽来自线 x 的唤醒中断请求
1：取消屏蔽来自线 x 的唤醒中断请求

1. 可配置事件输入的复位值置“0”，以在默认情况下禁止中断。
2. 直接事件输入的复位值置“1”，以在默认情况下使能中断。

14.5.14 EXTI CPU2 唤醒中断屏蔽寄存器 (EXTI_C2IMR2)

EXTI CPU2 wakeup with interrupt mask register

偏移地址：0x0D0

复位值：0x0001 FCFD

包含可配置事件和直接事件的寄存器位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IM48
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IM47	IM46	IM45	IM44	IM43	IM42	IM41	IM40	IM39	IM38	IM37	IM36	IM35	IM34	IM33	IM32
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:0 **IM[48:32]：**屏蔽事件输入 x 上的 CPU 唤醒中断 (CPU wakeup with interrupt mask on event input x)（x = 48 到 32）(1)(2)

0：屏蔽来自线 x 的唤醒中断请求

1：取消屏蔽来自线 x 的唤醒中断请求

1. 可配置事件输入的复位值置“0”，以在默认情况下禁止中断。
2. 直接事件输入的复位值置“1”，以在默认情况下使能中断。

14.5.15 EXTI CPU 唤醒事件屏蔽寄存器 (EXTI_EMR2)

EXTI CPU wakeup with event mask register

偏移地址：0x094

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	EM41	EM40	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
						rw	rw								

位 31:10 保留，必须保持复位值。

位 9 **EM41：**屏蔽事件输入 41 上的 CPU 唤醒事件生成 (CPU wakeup with event generation mask on event input 41)

0：屏蔽线 41 上的唤醒事件生成

1：取消屏蔽线 41 上的唤醒事件生成

位 8 **EM40：**屏蔽事件输入 40 上的 CPU 唤醒事件生成 (CPU wakeup with event generation mask on event input 40)

0：屏蔽线 40 上的唤醒事件生成

1：取消屏蔽线 40 上的唤醒事件生成

位 7:0 保留，必须保持复位值。

14.5.16 EXTI CPU2 唤醒事件屏蔽寄存器 (EXTI_C2EMR2)

EXTI CPU2 wakeup with event mask register

偏移地址：0x0D4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	EM41	EM40	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
						rw	rw								

位 31:10 保留，必须保持复位值。

位 9 **EM41**：屏蔽事件输入 41 上的 CPU2 唤醒事件生成 (CPU2 wakeup with event generation mask on event input 41)

0：屏蔽线 41 上的唤醒事件生成
1：取消屏蔽线 41 上的唤醒事件生成

位 8 **EM40**：屏蔽事件输入 40 上的 CPU2 唤醒事件生成 (CPU2 wakeup with event generation mask on event input 40)

0：屏蔽线 40 上的唤醒事件生成
1：取消屏蔽线 40 上的唤醒事件生成

位 7:0 保留，必须保持复位值。

14.5.17 EXTI 寄存器映射

下表列出了 EXTI 寄存器映射和复位值。

表 65. 异步中断/事件控制器寄存器映射和复位值

偏移	寄存器	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
0x000	**EXTI_RTSR1**	RT[31]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RT[21:0]																					
	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
0x004	**EXTI_FTSR1**	FT[31]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT[21:0]																					
	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
0x008	**EXTI_SWIER1**	SWI[31]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWI[21:0]																					
	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
0x00C	**EXTI_PR1**	PIF[31]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PIF[21:0]																					
	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
0x010-0x01C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x020	**EXTI_RTSR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RT[41:40]		Res.	Res.	Res.	Res.	Res.	Res.	RT33	Res.
	Reset value																							0	0							0	

表 65. 异步中断/事件控制器寄存器映射和复位值（续）

偏移	寄存器	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
0x024	**EXTI_FTSR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT [41:40]		Res.	Res.	Res.	Res.	Res.	Res.	FT33	Res.
	Reset value																							0	0							0	
0x028	**EXTI_SWIER2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWI [41:40]		Res.	Res.	Res.	Res.	Res.	Res.	SWI33	Res.
	Reset value																							0	0							0	
0x02C	**EXTI_PR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PIF [41:40]		Res.	Res.	Res.	Res.	Res.	Res.	PIF33	Res.
	Reset value																							0	0							0	
0x030-0x07C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x080	**EXTI_IMR1**	IM[31:0]																															
	Reset value	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	**EXTI_EMR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EM[21:17]					Res.	EM[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
0x088-0x08C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x090	**EXTI_IMR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IM[48:32]																
	Reset value																1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1
0x094	**EXTI_EMR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EM [41:40]		Res.	Res.	Res.	Res.	Res.	Res.	EM33	Res.
	Reset value																							0	0							0	
0x098-0x0BC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0C0	**EXTI_C2IMR1**	IM[31:0].																															
	Reset value	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C4	**EXTI_C2EMR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EM[21:17]					Res.	EM[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
0x0C8-0x0CC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0D0	**EXTI_C2IMR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IM[48:32]																
	Reset value																1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1
0x0D4	**EXTI_C2EMR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EM [41:40]		Res.	Res.	Res.	Res.	Res.	Res.	EM33	Res.
	Reset value																							0	0							0	

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

15 Quad-SPI 接口 (QUADSPI)

15.1 简介

QUADSPI 是一种专用的通信接口，连接单线、双线或四线 SPI Flash 存储介质。该接口可以在以下三种模式下工作：

- 间接模式：使用 QUADSPI 寄存器执行全部操作
- 状态轮询模式：周期性读取外部 Flash 状态寄存器，而且标志位置 1 时会产生中断（如擦除或烧写完成，会产生中断）
- 内存映射模式：外部 Flash 映射到器件地址空间，从而系统将其视作内部存储器

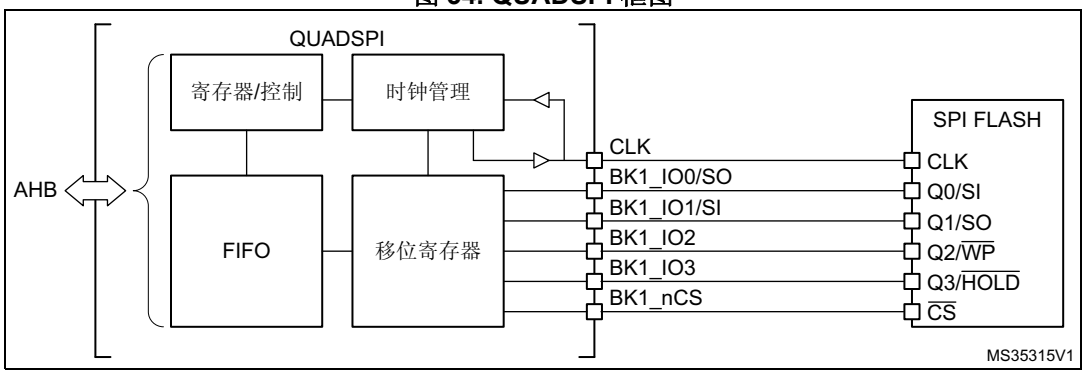
15.2 QUADSPI 主要特性

- 三种功能模式：间接模式、状态轮询模式和内存映射模式
- 支持 SDR 和 DDR 模式
- 针对间接模式和内存映射模式，完全可编程操作码
- 针对间接模式和内存映射模式，完全可编程帧格式
- 集成 FIFO，用于发送和接收
- 允许 8、16 和 32 位数据访问
- 具有适用于间接模式操作的 DMA 通道
- 在达到 FIFO 阈值、超时、操作完成以及发生访问错误时产生中断

15.3 QUADSPI 功能说明

15.3.1 QUADSPI 框图

图 34. QUADSPI 框图



15.3.2 QUADSPI 引脚

表 66 列出了用于连接 Flash 的 QUADSPI 引脚。

表 66. QUADSPI 引脚

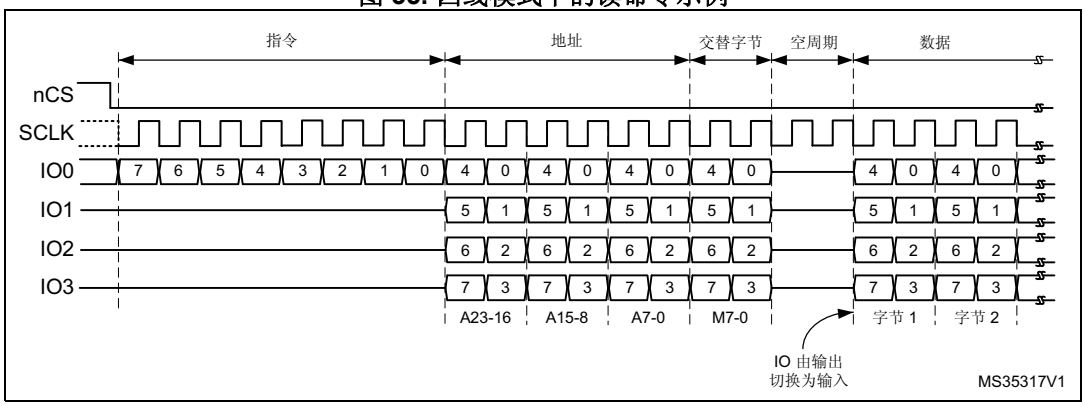
信号名称	信号类型	说明
CLK	数字输出	Flash 时钟
BK1_IO0/SO	数字输入/输出	在双线/四线模式中为双向 IO，在单线模式中为串行输出
BK1_IO1/SI	数字输入/输出	在双线/四线模式中为双向 IO，在单线模式中为串行输入
BK1_IO2	数字输入/输出	在四线模式中为双向 IO
BK1_IO3	数字输入/输出	在四线模式中为双向 IO
BK1_nCS	数字输出	片选（低电平有效）

15.3.3 QUADSPI 命令序列

QUADSPI 通过命令与 Flash 通信。每条命令包括指令、地址、交替字节、空指令和数据这五个阶段。任一阶段均可跳过，但至少要包含指令、地址、交替字节或数据阶段之一。

nCS 在每条指令开始前下降，在每条指令完成后再次上升。

图 35. 四线模式下的读命令示例



指令阶段

这一阶段，将在 QUADSPI_CCR[7:0] 寄存器的 INSTRUCTION 位域中配置的一条 8 位指令发送到 Flash，指定待执行操作的类型。

尽管大多数 Flash 从 IO0/SO 信号（单线 SPI 模式）只能以一次 1 位的方式接收指令，但指令阶段可选择一次发送 2 位（在双线 SPI 模式中通过 IO0/IO1）或一次发送 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3）。这可通过 QUADSPI_CCR[9:8] 寄存器中的 IMODE[1:0] 位域进行配置。

若 IMODE = 00，则跳过指令阶段，命令序列从地址阶段（如果存在）开始。

地址阶段

在地址阶段，将 1-4 字节发送到 Flash，指示操作地址。待发送的地址字节数在 QUADSPI_CCR[13:12] 寄存器的 ADSIZE[1:0] 位域中进行配置。在间接模式和自动轮询模式下，待发送的地址字节在 QUADSPI_AR 寄存器的 ADDRESS[31:0] 中指定。在内存映射模式下，则通过 AHB（来自于 Cortex® 或 DMA）直接给出地址。

地址阶段可一次发送 1 位（在单线 SPI 模式中通过 SO）、2 位（在双线 SPI 模式中通过 IO0/IO1）或 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3）。这可通过 QUADSPI_CCR[11:10] 寄存器中的 ADMODE[1:0] 位域进行配置。

若 ADMODE = 00，则跳过地址阶段，命令序列直接进入下一阶段（如果存在）。

交替字节阶段

在交替字节阶段，将 1-4 字节发送到 Flash，一般用于控制操作模式。待发送的交替字节数在 QUADSPI_CCR[17:16] 寄存器的 ABSIZE[1:0] 位域中进行配置。待发送的字节在 QUADSPI_ABR 寄存器中指定。

交替字节阶段可一次发送 1 位（在单线 SPI 模式中通过 SO）、2 位（在双线 SPI 模式中通过 IO0/IO1）或 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3）。这可通过 QUADSPI_CCR[15:14] 寄存器中的 ABMODE[1:0] 位域进行配置。

若 ABMODE = 00，则跳过交替字节阶段，命令序列直接进入下一阶段（如果存在）。

交替字节阶段存在仅需发送单个半字节而不是一个全字节的情况，比如采用双线模式并且仅使用两个周期发送交替字节时。在这种情况下，固件可采用四线模式 (ABMODE = 11) 并发送一个字节，方法是 ALTERNATE 的位 7 和 3 置“1”（IO3 保持高电平）且位 6 和 2 置“0”（IO2 线保持低电平）。此时，半字节的高 2 位存放在 ALTERNATE 的位 4:3，低 2 位存放在位 1 和位 0 中。例如，如果半字节 2 (0010) 通过 IO0/IO1 发送，则 ALTERNATE 应设置为 0x8A (1000_1010)。

空指令周期阶段

在空指令周期阶段，给定的 1-31 个周期内不发送或接收任何数据，目的是当采用更高的时钟频率时，给 Flash 留出准备数据阶段的时间。这一阶段中给定的周期数在 QUADSPI_CCR[22:18] 寄存器的 DCYC[4:0] 位域中指定。在 SDR 和 DDR 模式下，持续时间被指定为一定个数的全时钟周期。

若 DCYC 为零，则跳过空指令周期阶段，命令序列直接进入数据阶段（如果存在）。

空指令周期阶段的操作模式由 DMODE 确定。

为确保数据信号从输出模式转变为输入模式有足够的“转向”时间，使用双线和四线模式从 Flash 接收数据时，至少需要指定一个空指令周期。

数据阶段

在数据阶段，可从 Flash 接收或向其发送任意数量的字节。

在间接模式和自动轮询模式下，待发送/接收的字节数在 QUADSPI_DLR 寄存器中指定。

在间接写入模式下，发送到 Flash 的数据必须写入 QUADSPI_DR 寄存器。在间接读取模式下，通过读取 QUADSPI_DR 寄存器获得从 Flash 接收的数据。

在内存映射模式下，读取的数据通过 AHB 直接发送回 Cortex 或 DMA。

数据阶段可一次发送/接收 1 位（在单线 SPI 模式中通过 SO/SI）、2 位（在双线 SPI 模式中通过 IO0/IO1）或 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3）。这可通过 QUADSPI_CCR[15:14] 寄存器中的 ABMODE[1:0] 位域进行配置。

若 DMODE = 00，则跳过数据阶段，命令序列在拉高 nCS 时立即完成。这一配置仅可用于仅间接写入模式。

15.3.4 QUADSPI 信号接口协议模式

单线 SPI 模式

传统 SPI 模式允许串行发送/接收单独的 1 位。在此模式下，数据通过 SO 信号（其 I/O 与 IO0 共享）发送到 Flash。从 Flash 接收到的数据通过 SI（其 I/O 与 IO1 共享）送达。

通过将（QUADSPI_CCR 中的）IMODE/ADMODE/ABMODE/DMODE 位域设置为 01，可对不同的命令阶段分别进行配置，以使用此单个位模式。

在每个已配置为单线模式的阶段中：

- IO0 (SO) 处于输出模式
- IO1 (SI) 处于输入模式（高阻抗）
- IO2 处于输出模式并强制置“0”（以禁止“写保护”功能）
- IO3 处于输出模式并强制置“1”（以禁止“保持”功能）

若 DMODE = 01，这对于空指令阶段也同样如此。

双线 SPI 模式

在双线 SPI 模式下，通过 IO0/IO1 信号同时发送/接收两位。

通过将 QUADSPI_CCR 寄存器的 IMODE/ADMODE/ABMODE/DMODE 位域设置为 10，可对不同的命令阶段分别进行配置，以使用双线 SPI 模式。

在每个已配置为双线模式的阶段中：

- IO0/IO1 在数据阶段进行读取操作时处于高阻态（输入），在其他情况下为输出
- IO2 处于输出模式并强制置“0”
- IO3 处于输出模式并强制置“1”

在空指令阶段，若 DMODE = 01，则 IO0/IO1 始终保持高阻态。

四线 SPI 模式

在四线 SPI 模式下，通过 IO0/IO1/IO2/IO3 信号同时发送/接收四位。

通过将 QUADSPI_CCR 寄存器的 IMODE/ADMODE/ABMODE/DMODE 位域设置为 11，可对不同的命令阶段分别进行配置，以使用四线 SPI 模式。

在每个已配置为四线模式的阶段中，IO0/IO1/IO2/IO3 在数据阶段进行读取操作时均处于高阻态（输入），在其他情况下为输出。

在空指令阶段中，若 DMODE = 11，则 IO0/IO1/IO2/IO3 均为高阻态。

IO2 和 IO3 仅用于四线 SPI 模式。如果未配置任何阶段使用四线 SPI 模式，即使 QUADSPI 激活，对应 IO2 和 IO3 的引脚也可用于其他功能。

SDR 模式

默认情况下，DDRM 位 (QUADSPI_CCR[31]) 为 0，QUADSPI 在单倍数据速率 (SDR) 模式下工作。

在 SDR 模式下，当 QUADSPI 驱动 IO0/SO、IO1、IO2、IO3 信号时，这些信号仅在 CLK 的下降沿发生转变。

在 SDR 模式下接收数据时，QUADSPI 假定 Flash 也通过 CLK 的下降沿发送数据。默认情况下（SSHIFT = 0 时），将使用 CLK 后续的边沿（上升沿）对信号进行采样。

DDR 模式

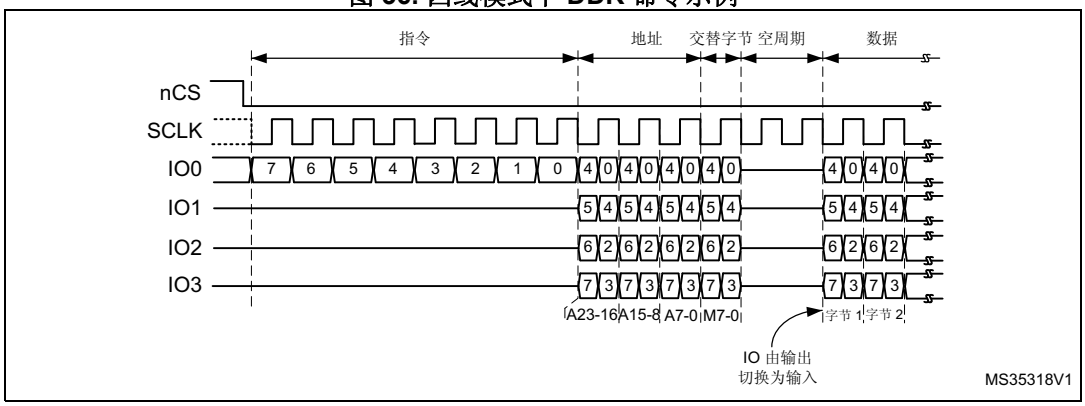
若 DDRM 位 (QUADSPI_CCR[31]) 置 1，则 QUADSPI 在双倍数据速率 (DDR) 模式下工作。

在 DDR 模式下，当 QUADSPI 在地址/交替字节/数据阶段驱动 IO0/SO、IO1、IO2、IO3 信号时，将在 CLK 的每个上升沿和下降沿发送 1 位。

指令阶段不受 DDRM 的影响。始终通过 CLK 的下降沿发送指令。

在 DDR 模式下接收数据时，QUADSPI 假定 Flash 通过 CLK 的上升沿和下降沿均发送数据。若 DDRM = 1，固件必须清零 SSHIFT 位（QUADSPI_CR 的位 4）。因此，在半个 CLK 周期后（下一个反向边沿）对信号采样。

图 36. 四线模式下 DDR 命令示例



15.3.5 QUADSPI 间接模式

在间接模式下，通过写入 QUADSPI 寄存器来触发命令；并通过读写数据寄存器来传输数据，就如同对待其他通信外设那样。

若 FMODE = 00 (QUADSPI_CCR[27:26])，则 QUADSPI 处于间接写入模式，字节在数据阶段中发送到 Flash。通过写入数据寄存器 (QUADSPI_DR) 的方式提供数据。

若 FMODE = 01，则 QUADSPI 处于间接读取模式，在数据阶段中从 Flash 接收字节。通过读取 QUADSPI_DR 来获取数据。

读取/写入的字节数在数据长度寄存器 (QUADSPI_DLR) 中指定。如果 QUADSPI_DLR = 0xFFFF_FFFF（全为“1”），则数据长度视为未定义，QUADSPI 将继续传输数据，直到到达（由 FSIZE 定义的）Flash 的结尾。如果不传输任何字节，DMODE (QUADSPI_CCR[25:24]) 应设置为 00。

如果 QUADSPI_DLR = 0xFFFF_FFFF 并且 FSIZE = 0x1F（最大值指示一个 4 GB 的 Flash），在此特殊情况下，传输将无限继续下去，仅在出现终止请求或 QUADSPI 被禁止后停止。在读取最后一个存储器地址后（地址为 0xFFFF_FFFF），将从地址 = 0x0000_0000 开始继续读取。

当发送或接收的字节数达到编程设定值时，如果 TCIE = 1，则 TCF 置 1 并产生中断。在数据数量不确定的情况下，将根据 QUADSPI_CR 中定义的 Flash 大小，在达到外部 SPI 的限制时，TCF 置 1。

触发命令启动

从本质上讲，在固件给出命令所需的最后一点信息时，命令即会启动。根据 QUADSPI 的配置，在间接模式下有三种触发命令启动的方式。在出现以下情形时，命令立即启动：

1. 对 INSTRUCTION[7:0] (QUADSPI_CCR) 执行写入操作，如果没有地址是必需的（当 ADMODE = 00）并且不需要固件提供数据（当 FMODE = 01 或 DMODE = 00）
2. 对 ADDRESS[31:0] (QUADSPI_AR) 执行写入操作，如果地址是必需的（当 ADMODE = 00）并且不需要固件提供数据（当 FMODE = 01 或 DMODE = 00）
3. 对 DATA[31:0] (QUADSPI_DR) 执行写入操作，如果地址是必需的（当 ADMODE != 00）并且需要固件提供数据（当 FMODE = 00 并且 DMODE != 00）

写入交替字节寄存器 (QUADSPI_ABR) 始终不会触发命令启动。如果需要交替字节，必须预先进行编程。

如果命令启动，BUSY 位（QUADSPI_SR 的位 5）将自动置 1。

FIFO 和数据管理

在间接模式中，数据将通过 QUADSPI 内部的一个 16 字节 FIFO。FLEVEL[4:0] (QUADSPI_SR[12:8]) 指示 FIFO 目前保存了多少字节。

在间接写入模式下 (FMODE = 00)，固件写入 QUADSPI_DR 时，将在 FIFO 中加入数据。字写入将在 FIFO 中增加 4 个字节，半字写入增加 2 个字节，而字节写入仅增加 1 个字节。如果固件在 FIFO 中加入的数据过多（超过 DL[31:0] 指示的值），将在写入操作结束（TCF 置 1）时从 FIFO 中清除超出的字节。

对 QUADSPI_DR 的字节/半字访问必须仅针对该 32 位寄存器的最低有效字节/半字。

FTHRES[3:0] 用于定义 FIFO 的阈值。如果达到阈值，FTF（FIFO 阈值标志）置 1。在间接读取模式下，从 FIFO 中读取的有效字节数超过阈值时，FTF 置 1。从 Flash 中读取最后一个字节后，如果 FIFO 中依然有数据，则无论 FTHRES 的设置为何，FTF 也都会置 1。在间接写入模式下，当 FIFO 中的空字节数超过阈值时，FTF 置 1。

如果 FTIE = 1，则 FTF 置 1 时产生中断。如果 DMAEN = 1，则 FTF 置 1 时启动 DMA 传送。如果阈值条件不再为“真”（CPU 或 DMA 传输了足够的数据后），则 FTF 由硬件清零。

15.3.6 QUADSPI 状态标志轮询模式

在自动轮询模式下，QUADSPI 周期性启动命令以读取一定数量的状态字节（最多 4 个）。可屏蔽接收的字节以隔离一些状态位，从而在所选的位具有定义的值时可产生中断。

对 Flash 的访问开始时与在间接读取模式下相同：如果不需要地址 (AMODE = 00)，则在写入 QUADSPI_CCR 时即开始访问。否则，如果需要地址，则在写入 QUADSPI_AR 时开始第一次访问。BUSY 在此时变为高电平，即使在周期性访问期间也保持不变。

在自动轮询模式下，MASK[31:0] (QUADSPI_PSMAR) 的内容用于屏蔽来自 Flash 的数据。如果 MASK[n] = 0，则屏蔽结果的位 n，从而不考虑此位。如果 MASK[n] = 1 并且位 [n] 的内容与 MATCH[n] (QUADSPI_PSMAR) 相同，说明存在位 n 匹配。

如果轮询匹配模式位（PMM，QUADSPI_CR 的位 23）为 0，将激活“AND”匹配模式。这意味着状态匹配标志 (SMF) 仅在全部未屏蔽位均存在匹配时置 1。

如果 PMM = 1，则激活“OR”匹配模式。这意味着 SMF 在任意未屏蔽位存在匹配时置 1。

如果 SMIE = 1，则在 SMF 置 1 时调用一个中断。

如果自动轮询模式停止 (APMS) 位置 1，则操作停止并且 BUSY 位在检测到匹配时清零。否则，BUSY 位保持为“1”，在发生中止或禁止 QUADSPI (EN = 0) 前继续进行周期性访问。

数据寄存器 (QUADSPI_DR) 包含最新接收的状态字节（FIFO 停用）。数据寄存器的内容不受匹配逻辑所用屏蔽方法的影响。FTF 状态位在新一次状态读取完成后置 1，并且 FTF 在数据读取后清零。

15.3.7 QUADSPI 内存映射模式

在配置为内存映射模式时，外部 SPI 器件被视为是内部存储器。

QUADSPI 外设若没有正确配置并使能，禁止访问 QUADSPI Flash 的存储区域。

即使 Flash 容量更大，寻址空间也无法超过 256 MB。

如果访问的地址超出 FSIZE 定义的范围但仍在 256 MB 范围内，则生成总线错误。此错误的影响具体取决于尝试进行访问的总线主设备：

- 如果为 Cortex® CPU，则会在使能总线故障时发生总线故障异常，在禁止总线故障时发生硬性故障异常。
- 如果为 DMA，则生成 DMA 传输错误，并自动禁用相应的 DMA 通道。

支持字节、半字和字访问类型。

支持芯片内执行 (XIP) 操作，QUADSPI 预期下一次访问并提前加载后续地址中的字节。如果之后访问的是连续地址，由于值已经预取，访问将更快完成。

默认情况下，即便在很长时间内不访问 Flash，QUADSPI 也不会停止预取操作，之前的读取操作将保持激活状态并且 nCS 保持低电平。由于 nCS 保持低电平时，Flash 功耗增加，应用程序可能会激活超时计数器（TCEN = 1，QUADSPI_CR 的位 3），从而在 FIFO 中写满预取的数据后，若在 TIMEOUT[15:0] (QUADSPI_LPTR) 个周期的时长内没有访问，则释放 nCS。

BUSY 在第一次内存映射访问发生时变为高电平。由于进行预取操作，BUSY 在发生超时、中止或外设禁止前不会下降。

15.3.8 QUADSPI Flash 配置

设备配置寄存器 (QUADSPI_DCR) 可用于指定外部 SPI Flash 的特性。

FSIZE[4:0] 位域使用下面的公式定义外部存储器的大小：

Flash 中的字节数 = $2^{[FSIZE+1]}$

FSIZE+1 是对 Flash 寻址所需的地址位数。在间接模式下，Flash 容量最高可达 4 GB（使用 32 位进行寻址），但在内存映射模式下的可寻址空间限制为 256 MB。

QUADSPI 连续执行两条命令时，它在两条命令之间将片选信号 (nCS) 置为高电平默认仅一个 CLK 周期时长。如果 Flash 需要命令之间的时间更长，可使用片选高电平时间 (CSHT) 位域指定 nCS 必须保持高电平的最少 CLK 周期数（最大为 8）。

时钟模式 (CKMODE) 位指示命令之间的 CLK 信号逻辑电平（nCS = 1 时）。

15.3.9 QUADSPI 延迟数据采样

默认情况下，QUADSPI 在 Flash 驱动信号后过半个 CLK 周期才对 Flash 驱动的数据采样。

在外部信号延迟时，这有利于推迟数据采样。使用 SSHIFT 位（QUADSPI_CR 的位 4），可将数据采样移位半个 CLK 周期。

DDR 模式下不支持时钟移位：若 DDRM 位置 1，SSHIFT 位必须清零。

15.3.10 QUADSPI 配置

QUADSPI 配置分两个阶段：

- QUADSPI IP 配置
- QUADSPI Flash 配置

QUADSPI 在配置完毕并使能后，即可在间接模式、状态轮询模式和内存映射模式这三种操作模式之一下工作。

QUADSPI IP 配置。

通过 QUADSPI_CR 配置 QUADSPI IP。用户应配置传入数据的时钟预分频器的分频系数以及采样移位设置。

DDR 模式可通过 DDRM 位进行设置。使能该模式后，在每个时钟的上升沿和下降沿都会发送地址和交替字节以及发送/接收数据。无论 DDRM 位如何设置，都将在 SDR 模式下发送指令。

DMA 请求通过 DMAEN 位置 1 使能。若是用于中断，则相关使能位也可在该阶段置 1。

生成 DMA 请求或生成中断的 FIFO 电平在 FTHRES 位中进行编程。

如果需要超时计数器，则可将 TCEN 位置 1 并在 QUADSPI_LPTR 寄存器中编程超时值。

QUADSPI Flash 配置

与外部目标 Flash 相关的参数通过 QUADSPI_DCR 寄存器进行配置。用户应在 FSIZE 位中编程 Flash 的大小、在 CSHT 位中编程片选保持高电平的最短时间以及在 MODE 位中编程功能模式（模式 0 或模式 3）。

15.3.11 QUADSPI 的用法

使用 FMODE[1:0] (QUADSPI_CCR[27:26]) 选择操作模式。

间接模式的操作步骤

FMODE 编程为 00 可选择间接写入模式，将数据发送到 Flash。FMODE 编程为 01 可选择间接读取模式，读取 Flash 中的数据。

QUADSPI 用于间接模式时，采用以下方式构建帧：

1. 在 QUADSPI_DLR 中指定待读取或写入的数据字节数。
2. 在 QUADSPI_CCR 中指定帧格式、模式和指令代码。
3. 在 QUADSPI_ABR 中指定要在地址阶段后立即发送的可选交替字节。
4. 在 QUADSPI_CR 中指定工作模式。若 FMODE = 00（间接写入模式）并且 DMAEN = 1，则应在 QUADSPI_CR 前指定 QUADSPI_AR。否则在 QUADSPI_AR 更新前（如果已经使能 DMA 控制器），DMA 便可能写入 QUADSPI_DR。
5. 在 QUADSPI_AR 中指定目标地址。
6. 通过 QUADSPI_DR 从 FIFO 读取数据/向 FIFO 写入数据。

在写入控制寄存器 (QUADSPI_CR) 时，用户可指定以下设置：

- 使能位 (EN) 设置为“1”
- DMA 使能位 (DMAEN)，用于向/从 RAM 传输/接收数据
- 超时计数器使能位 (TCEN)
- 采样移位设置 (SSHIFT)
- FIFO 阈值 (FTRHES)，以指示 FTF 标志在何时置 1
- 中断使能
- 自动轮询模式参数：匹配模式和停止模式（在 FMODE = 11 时有效）
- 时钟预分频器

在写入通信配置寄存器 (QUADSPI_CCR) 时，用户指定以下参数：

- 通过 INSTRUCTION 位指定指令字节
- 通过 IMODE 位指定指令发送方式（1/2/4 线）
- 通过 ADMODE 位指定地址发送方式（无/1/2/4 线）
- 通过 ADSIZE 位指定地址长度（8/16/24/32 位）
- 通过 ABMODE 位指定交替字节发送方式（无/1/2/4 线）
- 通过 ABSIZE 位指定交替字节数 (1/2/3/4)
- 通过 DBMODE 位指定是否存在空指令字节
- 通过 DCYC 位指定空指令字节数
- 通过 DMODE 位指定数据发送/接收方式（无/1/2/4 线）

如果无需为某个命令更新地址寄存器 (QUADSPI_AR) 与数据寄存器 (QUADSPI_DR)，则在写入 QUADSPI_CCR 时，该命令序列便立即启动。在 ADMODE 和 DMODE 均为 00 时，或在间接读取模式 (FMODE = 01) 下仅 ADMODE = 00 时，便属于此情况。

在需要地址（ADMODE 不为 00），但无需写入数据寄存器（FMODE = 01 或 DMODE = 00）时，通过写入 QUADSPI_AR 更新地址后，命令序列便立即启动。

在数据传输（FMODE = 00 并且 DMODE ! = 00）中，通过 QUADSPI_DR 写入 FIFO 触发通信启动。

状态标志轮询模式

将 FMODE 位域 (QUADSPI_CCR[27:26]) 设置为 10，使能状态标志轮询模式。在此模式下，将发送编程的帧并周期性检索数据。

每帧中读取的最大数据量为 4 字节。如果 QUADSPI_DLR 请求更多的数据，则忽略多余的部分并仅读取 4 个字节。

在 QUADSPI_PISR 寄存器中指定周期性。

在检索到状态数据后，可在内部进行处理，以达到以下目的：

- 将状态匹配标志位置 1，如果使能，还将产生中断
- 自动停止周期性检索状态字节

接收到的值可通过存储于 QUADSPI_PSMKR 中的值进行屏蔽，并与存储在 QUADSPI_PSMAR 中的值进行或运算或与运算。

若是存在匹配，则状态匹配标志置 1，并且在使能了中断的情况下还将产生中断；如果 AMPS 位置 1，则 QUADSPI 自动停止。

在任何情况下，最新的检索值都在 QUADSPI_DR 中可用。

内存映射模式

在内存映射模式下，外部 Flash 被视为内部存储器，只是存在访问延迟。在该模式下，仅允许对外部 Flash 执行读取操作。

将 QUADSPI_CCR 寄存器中的 FMODE 设置为 11 可进入内存映射模式。

当主器件访问存储器映射空间时,将发送已编程的指令和帧。

FIFO 用作预取缓冲区以接受线性读取。在此模式中，对于 QUADSPI_DR 的任何访问均返回零。

数据长度寄存器 (QUADSPI_DLR) 在内存映射模式中无意义。

15.3.12 指令仅发送一次

一些 Flash（例如 Winbound）能够提供一种模式，指令在该模式中仅通过第一个命令序列进行发送，后续的命令根据地址直接启动。用户通过使用 SIOO 位 (QUADSPI_CCR[28]) 可利用此功能的优势。

SIOO 对于所有功能模式（间接模式、状态轮询模式和内存映射模式）均有效。如果 SIOO 位置 1，仅第一条命令发送指令，接着对 QUADSPI_CCR 执行写入操作。后续命令序列都将跳过指令阶段，直到 QUADSPI_CCR 被写入为止。

在 IMODE = 00（无指令）时，SIOO 不起作用。

15.3.13 QUADSPI 差错管理

在以下情况下可能产生错误：

- 在间接模式或状态标志轮询模式下，如果在 QUADSPI_AR 中编程了错误的地址（根据 QUADSPI_DCR 中 FSIZE[4:0] 定义的 Flash 大小）：TEF 将置 1，如果使能，还将产生中断。
- 另外，在间接模式下，如果地址加数据的长度超过 Flash 的大小，TEF 将在访问被触发时置 1。
- 在内存映射模式下，当主器件执行的访问超出范围或 QUADSPI 被禁止时：将产生总线错误，以响应故障总线主器件请求。
- 当主器件访问存储器映射空间，但内存映射模式被禁止时：将产生总线错误，以响应故障总线主器件请求。

15.3.14 QUADSPI 的繁忙位和中止功能

在 QUADSPI 启动对 Flash 的操作时，QUADSPI_SR 中的 BUSY 位自动置 1。

在间接模式下，在 QUADSPI 完成了请求的命令序列并且 FIFO 为空时，BUSY 位复位。

在自动轮询模式下，仅当最后一次周期性访问完成时（因 APMS = 1 时发生匹配，或因中止），BUSY 位才变为低电平。

在内存映射模式下进行第一次访问后，仅在发生超时事件或中止时 BUSY 位变为低电平。

任何操作都可通过将 QUADSPI_CR 中的 ABORT 位置 1 来中止。在完成中止时，BUSY 位和 ABORT 位自动复位，FIFO 清空。

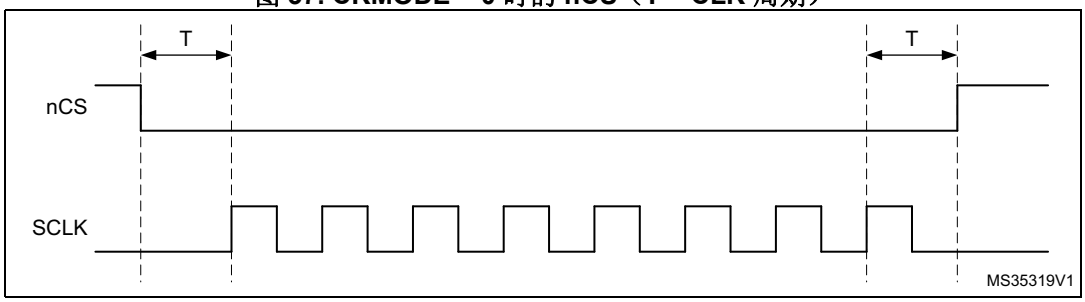
注： *如果中止对状态寄存器的写入操作，有些 Flash 可能发生错误行为。*

15.3.15 nCS 行为

默认情况下，nCS 为高电平，取消选择外部 Flash。nCS 在操作开始前下降，在操作完成时立即上升。

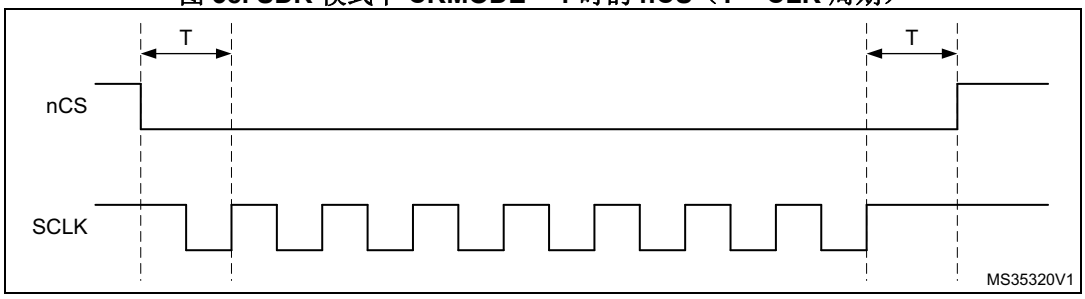
当 CKMODE = 0（“模式 0”，在未进行任何操作时 CLK 保持低电平）时，nCS 在操作首次升高 CLK 边沿时的一个 CLK 周期前降至低电平，在操作最后一次升高 CLK 边沿时的一个 CLK 周期后升至高电平，如*图 37* 所示。

图 37. CKMODE = 0 时的 nCS（T = CLK 周期）



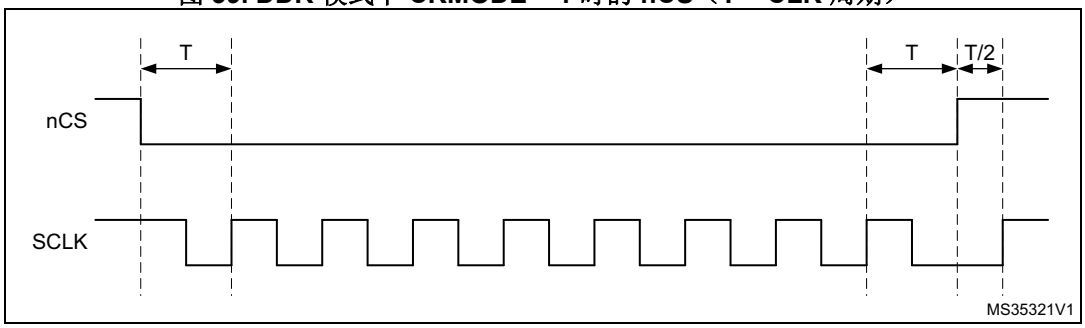
当 CKMODE = 1（“模式 3”，在未进行任何操作时 CLK 升至高电平）且 DDRM = 0（SDR 模式）时，nCS 仍在操作首次升高 CLK 边沿时的一个 CLK 周期前降至低电平，在操作最后一次升高 CLK 边沿时的一个 CLK 周期后升至高电平，如*图 38* 所示。

图 38. SDR 模式下 CKMODE = 1 时的 nCS（T = CLK 周期）



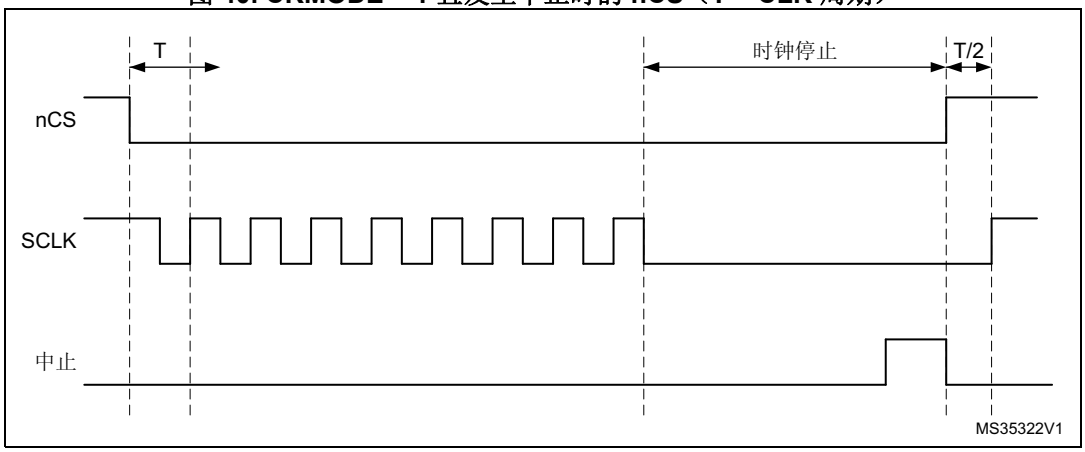
当 CKMODE = 1（模式 3）且 DDRM = 1（DDR 模式）时，nCS 在操作首次升高 CLK 边沿时的一个 CLK 周期前降至低电平，在操作最后一次升高 CLK 边沿时的一个 CLK 周期后升至高电平，如*图 39* 所示。由于 DDR 操作必须伴随下降沿完成，当 nCS 变为高电平时，CLK 为低电平并在经过半个周期后恢复高电平。

图 39. DDR 模式下 CKMODE = 1 时的 nCS（T = CLK 周期）



若 FIFO 在读取操作中保持写满状态或在写入操作中保持为空，则在固件干预 FIFO 前，操作停止并且 CLK 保持低电平。若操作停止时发生中止，则 nCS 在请求中止后即升至高电平，CLK 则在半个周期后升至高电平，如*图 40* 所示。

图 40. CKMODE = 1 且发生中止时的 nCS（T = CLK 周期）



15.4 QUADSPI 中断

发生如下事件时可生成中断：

- 超时
- 状态匹配
- FIFO 阈值
- 传输完成
- 传输错误

可以使用单独的中断使能位以提高灵活性。

表 67. QUADSPI 中断请求

中断事件	事件标志	使能控制位
超时	TOF	TOIE
状态匹配	SMF	SMIE
FIFO 阈值	FTF	FTIE
传输完成	TCF	TCIE
传输错误	TEF	TEIE

15.5 QUADSPI 寄存器

15.5.1 QUADSPI 控制寄存器 (QUADSPI_CR)

QUADSPI control register

偏移地址：0x0000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESCALER[7:0]								PMM	APMS	Res.	TOIE	SMIE	FTIE	TCIE	TEIE
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
Res.	Res.	Res.	Res.	FTHRES[3:0]				Res.	Res.	Res.	SSHIFT	TCEN	DMAEN	ABORT	EN
				rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:24 **PRESCALER[7:0]**：时钟预分频器 (Clock prescaler)

此位域定义基于时钟生成 CLK 所用的分频系数（值 + 1）。

0：F_{CLK} = F，时钟直接用作 QUADSPI CLK（预分频器被旁路）

1：F_{CLK} = F/2

2：F_{CLK} = F/3

...

255：F_{CLK} = F/256

对于奇数时钟分频系数，CLK 的占空比并非 50%。时钟信号的低电平持续时间比高电平持续时间多一个周期。

仅可在 BUSY = 0 时修改此位域。

位 23 **PMM**：轮询匹配模式 (Polling match mode)
此位指示在自动轮询模式期间用来确定是否“匹配”的方法。
0：AND 匹配模式。如果从 Flash 接收的所有未屏蔽位均与匹配寄存器中的对应位相匹配，则 SMF 置 1。
1：OR 匹配模式。如果从 Flash 接收的任意一个未屏蔽位与匹配寄存器中的对应位相匹配，则 SMF 置 1。
仅可在 BUSY = 0 时修改此位。

位 22 **APMS**：自动轮询模式停止 (Automatic poll mode stop)
此位确定在匹配后自动轮询是否停止。
0：仅通过中止或禁用 QUADSPI 停止自动轮询模式。
1：发生匹配时，自动轮询模式停止。
仅可在 BUSY = 0 时修改此位。

位 21 保留，必须保持复位值。

位 20 **TOIE**：超时中断使能 (TimeOut interrupt enable)
此位使能超时中断。
0：禁止中断
1：使能中断

位 19 **SMIE**：状态匹配中断使能 (Status match interrupt enable)
此位使能状态匹配中断。
0：禁止中断
1：使能中断

位 18 **FTIE**：FIFO 阈值中断使能 (FIFO threshold interrupt enable)
此位使能 FIFO 阈值中断。
0：禁止中断
1：使能中断

位 17 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)
此位使能传输完成中断。
0：禁止中断
1：使能中断

位 16 **TEIE**：传输错误中断使能 (Transfer error interrupt enable)
此位使能传输错误中断。
0：禁止中断
1：使能中断

位 15:12 保留，必须保持复位值。

位 11:8 **FTHRES[3:0]**：FIFO 阈值级别 (FIFO threshold level)

定义在间接模式下 FIFO 中将导致 FIFO 阈值标志（FTF，QUADSPI_SR[2]）置 1 的字节数阈值。

在间接写入模式下 (FMODE = 00)：

0：如果 FIFO 中存在 1 个或更多空闲字节可供写入，则 FTF 置 1

1：如果 FIFO 中存在 2 个或更多空闲字节可供写入，则 FTF 置 1

...

15：如果 FIFO 中存在 16 个空闲字节可供写入，则 FTF 置 1

在间接读取模式下 (FMODE = 01)：

0：如果 FIFO 中存在 1 个或更多有效字节可供读取，则 FTF 置 1

1：如果 FIFO 中存在 2 个或更多有效字节可供读取，则 FTF 置 1

...

15：如果 FIFO 中存在 16 个有效字节可供读取，则 FTF 置 1

如果 DMAEN = 1，则在更改 FTHRES 值前，必须禁止相应通道的 DMA 控制器。

位 7:5 保留，必须保持复位值。

位 4 **SSHIFT**：采样移位 (Sample shift)

默认情况下，QUADSPI 在 Flash 驱动数据后过半个 CLK 周期开始采集数据。使用此位，可考虑外部信号延迟，推迟数据采集。

0：不发生移位

1：移位半个周期

在 DDR 模式下 (DDRM = 1)，固件必须确保 SSHIFT = 0。

仅可在 BUSY = 0 时修改此位域。

位 3 **TCEN**：超时计数器使能 (Timeout counter enable)

此位仅在内存映射模式 (FMODE = 11) 下有效。激活此位后，如果在一段时间（通过 TIMEOUT[15:0] (QUADSPI_LPTR) 定义）内一直没有进行访问，将释放片选 (nCS)（从而降低功耗）。

使能超时计数器。

默认情况下，即便在很长时间内不访问 Flash，QUADSPI 也不会停止预取操作，之前的读取操作将保持激活状态并且 nCS 保持低电平。由于 nCS 保持低电平时，Flash 功耗增加，应用程序可能会激活超时计数器（TCEN = 1，QUADSPI_CR 的位 3），从而在 FIFO 中写满预取的数据后，若在 TIMEOUT[15:0] (QUADSPI_LPTR) 个周期的时长内没有访问，则释放 nCS。

0：禁止超时计数器，在内存映射模式中进行访问后，片选 (nCS) 保持激活。

1：使能超时计数器，在内存映射模式下，Flash 持续不活动 TIMEOUT[15:0] 个周期后释放片选 (nCS)。

仅可在 BUSY = 0 时修改此位。

位 2 **DMAEN**：DMA 使能 (DMA enable)

在间接模式下，通过 QUADSPI_DR 寄存器可使用 DMA 输入或输出数据。FIFO 阈值标志 FTF 置 1 时，将触发 DMA 传输。

0：间接模式下禁止 DMA

1：间接模式下使能 DMA

位 1 **ABORT**：中止请求 (Abort request)
此位中止执行中的命令序列。在中止完成时自动复位。
此位可停止当前的传输。
在轮询模式或内存映射模式下，此位也用以复位 APM 位或 DM 位。
0：不请求中止
1：请求中止

位 0 **EN**：使能 (Enable)
使能 QUADSPI。
0：禁止 QUADSPI
1：使能 QUADSPI

15.5.2 QUADSPI 器件配置寄存器 (QUADSPI_DCR)

QUADSPI device configuration register

偏移地址：0x0004

复位值：0x0000 0000

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

位 31:21 保留，必须保持复位值。

位 20:16 **FSIZE[4:0]**：Flash 大小 (Flash memory size)
此位域使用下面的公式定义了外部存储器的大小：
Flash 中的字节数 = $2^{[FSIZE+1]}$
FSIZE+1 是对 Flash 寻址所需的地址位数。在间接模式下，Flash 容量最高可达 4 GB（使用 32 位进行寻址），但在内存映射模式下的可寻址空间限制为 256 MB。
仅可在 BUSY = 0 时修改此位域。

位 15:11 保留，必须保持复位值。

位 10:8 **CSHT[2:0]**：片选高电平时间 (Chip select high time)
CSHT+1 定义片选 (nCS) 在发送至 Flash 的命令之间必须保持高电平的最少 CLK 周期数。
0：nCS 在 Flash 命令之间保持高电平至少 1 个周期
1：nCS 在 Flash 命令之间保持高电平至少 2 个周期
...
7：nCS 在 Flash 命令之间保持高电平至少 8 个周期
仅可在 BUSY = 0 时修改此位域。

位 7:1 保留，必须保持复位值。

位 0 **CKMODE**：模式 0/模式 3 (Mode 0 / mode 3)
此位指示 CLK 在命令之间（nCS = 1 时）的电平。
0：nCS 为高电平（片选释放）时，CLK 必须保持低电平。这称为模式 0。
1：nCS 为高电平（片选释放）时，CLK 必须保持高电平。这称为模式 3。
仅可在 BUSY = 0 时修改此位域。

15.5.3 QUADSPI 状态寄存器 (QUADSPI_SR)

QUADSPI status register

偏移地址：0x0008

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	FLEVEL[4:0]					Res.	Res.	BUSY	TOF	SMF	FTF	TCF	TEF
			r	r	r	r	r			r	r	r	r	r	r

位 31:13 保留，必须保持复位值。

位 12:8 **FLEVEL[4:0]**：FIFO 级别 (FIFO level)

此位域给出 FIFO 中正在保存的字节数量。FIFO 为空时 FLEVEL = 0，写满时 FLEVEL = 16。在内存映射模式和自动状态轮询模式下，FLEVEL 为零。

位 7:6 保留，必须保持复位值。

位 5 **BUSY**：忙 (Busy)

操作进行时，此位置 1。在对 Flash 的操作完成并且 FIFO 为空时，此位自动清零。

位 4 **TOF**：超时标志 (Timeout flag)

发生超时时此位置 1。向 CTOF 写入 1 可将此位清零。

位 3 **SMF**：状态匹配标志 (Status match flag)

在自动轮询模式下，若未屏蔽的接收数据与匹配寄存器 (QUADSPI_PSMAR) 中的对应位相匹配，则此位置 1。向 CSMF 写入 1 可将此位清零。

位 2 **FTF**：FIFO 阈值标志 (FIFO threshold flag)

在间接模式下，若达到 FIFO 阈值，或从 Flash 读取完成后，FIFO 中留有数据时，此位置 1。只要阈值条件不再为“真”，此位就自动清零。

在自动轮询模式下，每次读取状态寄存器时，此位即置 1；读取数据寄存器时，此位清零。

位 1 **TCF**：传输完成标志 (Transfer complete flag)

在间接模式下，当传输的数据数量达到编程设定值，或在任何模式下传输中止时，此位置 1。向 CTCF 写入 1 时，此位清零。

位 0 **TEF**：传输错误标志 (Transfer error flag)

在间接模式下访问无效地址时，此位置 1。向 CTEF 写入 1 可将此位清零。

15.5.4 QUADSPI 标志清零寄存器 (QUADSPI_FCR)

QUADSPI flag clear register

偏移地址：0x000C

复位值：0x0000 0000

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

位 31:5 保留，必须保持复位值。

位 4 **CTOF**：清零超时标志 (Clear timeout flag)
写入 1 可将 QUADSPI_SR 寄存器中的 TOF 标志清零

位 3 **CSMF**：清零状态匹配标志 (Clear status match flag)
写入 1 可将 QUADSPI_SR 寄存器中的 SMF 标志清零

位 2 保留，必须保持复位值。

位 1 **CTCF**：清零传输完成标志 (Clear transfer complete flag)
写入 1 可将 QUADSPI_SR 寄存器中的 TCF 标志清零

位 0 **CTEF**：清零传输错误标志 (Clear Transfer error flag)
写入 1 可将 QUADSPI_SR 寄存器中的 TEF 标志清零

15.5.5 QUADSPI 数据长度寄存器 (QUADSPI_DLR)

QUADSPI data length register

偏移地址：0x0010

复位值：0x0000 0000

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

位 31:0 **DL[31:0]**：数据长度 (Data length)

在间接模式和状态轮询模式下待检索的数据数量（值 + 1）。对状态轮询模式应使用不大于 3 的值（表示 4 字节）。

在间接模式下，所有位置 1 表示未定义长度，QUADSPI 将继续传输数据直到到达由 FSIZE 定义的存储器末尾。

0x0000_0000：传输 1 个字节

0x0000_0001：传输 2 个字节

0x0000_0002：传输 3 个字节

0x0000_0003：传输 4 个字节

...

0xFFFF_FFFD：传输 4,294,967,294 (4G-2) 个字节

0xFFFF_FFFE：传输 4,294,967,295 (4G-1) 个字节

0xFFFF_FFFF：未定义长度 -- 传输所有字节直到到达由 FSIZE 定义的 Flash 的结尾。

如果 FSIZE = 0x1F，则读取无限继续下去。

此位域在内存映射模式 (FMODE = 10) 下不起作用。

仅可在 BUSY = 0 时写入此位域。

15.5.6 QUADSPI 通信配置寄存器 (QUADSPI_CCR)

QUADSPI communication configuration register

偏移地址：0x0014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DDRM	Res.	Res.	SIOO	FMODE[1:0]		DMODE[1:0]		Res.	DCYC[4:0]					ABSIZE[1:0]	
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
ABMODE[1:0]		ADSIZE[1:0]		ADMODE[1:0]		IMODE[1:0]		INSTRUCTION[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **DDRM**：双倍数据速率模式 (Double data rate mode)

此位为地址、交替字节和数据阶段设置 DDR 模式：

0：禁止 DDR 模式

1：使能 DDR 模式

仅可在 BUSY = 0 时写入此位域。

位 30 保留，必须保持复位值。

位 29 保留，必须保持复位值。

位 28 **SIOO**：仅发送指令一次模式 (Send instruction only once mode)

请参见*第 386 页的第 15.3.12 节：指令仅发送一次*。IMODE = 00 时，此位不起作用。

0：在每个事务中发送指令

1：仅为第一条命令发送指令

仅可在 BUSY = 0 时写入此位域。

位 27:26 **FMODE[1:0]**：功能模式 (Functional mode)
此位域定义 QUADSPI 操作的功能模式：
00：间接写入模式
01：间接读取模式
10：自动轮询模式
11：内存映射模式
如果 DMAEN = 1，则在更改 FMODE 值前，必须禁止相应通道的 DMA 控制器。
仅可在 BUSY = 0 时写入此位域。

位 25:24 **DMODE[1:0]**：数据模式 (Data mode)
此位域定义数据阶段的操作模式：
00：无数据
01：单线传输数据
10：双线传输数据
11：四线传输数据
此位域还定义空指令阶段的操作模式。
仅可在 BUSY = 0 时写入此位域。

位 23 保留，必须保持复位值。

位 22:18 **DCYC[4:0]**：空指令周期数 (Number of dummy cycles)
此位域定义空指令阶段的持续时间。在 SDR 和 DDR 模式下，它指定 CLK 周期数 (0-31)。
仅可在 BUSY = 0 时写入此位域。

位 17:16 **ABSIZE[1:0]**：交替字节长度 (Alternate bytes size)
此位定义交替字节长度：
00：8 位交替字节
01：16 位交替字节
10：24 位交替字节
11：32 位交替字节
仅可在 BUSY = 0 时写入此位域。

位 15:14 **ABMODE[1:0]**：交替字节模式 (Alternate bytes mode)
此位域定义交替字节阶段的操作模式：
00：无交替字节
01：单线传输交替字节
10：双线传输交替字节
11：四线传输交替字节
仅可在 BUSY = 0 时写入此位域。

位 13:12 **ADSIZE[1:0]**：地址长度 (Address size)
此位定义地址长度：
00：8 位地址
01：16 位地址
10：24 位地址
11：32 位地址
仅可在 BUSY = 0 时写入此位域。

位 11:10 **ADMODE[1:0]**：地址模式 (Address mode)
此位域定义地址阶段的操作模式：
00：无地址
01：单线传输地址
10：双线传输地址
11：四线传输地址
仅可在 BUSY = 0 时写入此位域。

位 9:8 **IMODE[1:0]**：指令模式 (Instruction mode)
此位域定义指令阶段的操作模式：
00：无指令
01：单线传输指令
10：双线传输指令
11：四线传输指令
仅可在 BUSY = 0 时写入此位域。

位 7:0 **INSTRUCTION[7:0]**：指令 (Instruction)
指定要发送到外部 SPI 设备的指令。
仅可在 BUSY = 0 时写入此位域。

15.5.7 QUADSPI 地址寄存器 (QUADSPI_AR)

QUADSPI address register

偏移地址：0x0018

复位值：0x0000 0000

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

位 31:0 **ADDRESS[31:0]**：地址 (Address)
指定要发送到外部 Flash 的地址。
BUSY = 0 或 FMODE = 11（内存映射模式）时，将忽略写入此位域。

15.5.8 QUADSPI 交替字节寄存器 (QUADSPI_ABR)

QUADSPI alternate bytes registers

偏移地址：0x001C

复位值：0x0000 0000

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

位 31:0 **ALTERNATE[31:0]**：交替字节 (Alternate Bytes)

指定要在地址后立即发送到外部 SPI 器件的可选数据。

仅可在 BUSY = 0 时写入此位域。

15.5.9 QUADSPI 数据寄存器 (QUADSPI_DR)

QUADSPI data register

偏移地址：0x0020

复位值：0x0000 0000

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

位 31:0 **DATA[31:0]**：数据 (Data)

指定要与外部 SPI 器件交换的数据。

在间接写入模式下，写入该寄存器的数据在数据阶段发送到 Flash，在此之前则存储于 FIFO。如果 FIFO 太满，将暂停写入，直到 FIFO 具有足够的空间接受要写入的数据才继续。

在间接模式下，读取该寄存器可获得（通过 FIFO）已从 Flash 接收的数据。如果 FIFO 所含字节数比读取操作要求的字节数少并且 BUSY=1，将暂停读取，直到足够的数据出现或传输完成（不分先后）才继续。

在自动轮询模式下，该寄存器包含最后从 Flash 读取的数据（未进行屏蔽）。

支持对该寄存器进行字、半字以及字节访问。在间接写入模式下，字节写入将在 FIFO 中增加 1 个字节，半字写入增加 2 个，而字写入则增加 4 个。类似地，在间接读取模式下，字节读取将擦除 FIFO 中的 1 个字节，半字读取擦除 2 个，而字读取则擦除 4 个。间接模式下的访问必须与此寄存器的最低位对齐：字节读取必须读取 DATA[7:0] 而半字读取必须读取 DATA[15:0]。

15.5.10 QUADSPI 轮询状态屏蔽寄存器 (QUADSPI _PSMKR)

QUADSPI polling status mask register

偏移地址：0x0024

复位值：0x0000 0000

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

位 31:0 **MASK[31:0]**：状态屏蔽 (Status mask)

对在轮询模式下接收的状态字节进行屏蔽。

对于位 n：

0：屏蔽在自动轮询模式下所接收数据的位 n，在匹配逻辑中不考虑值

1：不屏蔽在自动轮询模式下所接收数据的位 n，在匹配逻辑中考虑值

仅可在 BUSY = 0 时写入此位域。

15.5.11 QUADSPI 轮询状态匹配寄存器 (QUADSPI _PSMAR)

QUADSPI polling status match register

偏移地址：0x0028

复位值：0x0000 0000

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

位 31:0 **MATCH[31:0]**：状态匹配 (Status match)

该值将与屏蔽状态寄存器比较以进行匹配。

仅可在 BUSY = 0 时写入此位域。

15.5.12 QUADSPI 轮询间隔寄存器 (QUADSPI _PIR)

QUADSPI polling interval register

偏移地址：0x002C

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 **INTERVAL[15:0]**：轮询间隔 (Polling interval)
自动轮询阶段读取操作之间的 CLK 周期数。
仅可在 BUSY = 0 时写入此位域。

15.5.13 QUADSPI 低功耗超时寄存器 (QUADSPI_LPTR)

QUADSPI low-power timeout register

偏移地址：0x0030

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 **TIMEOUT[15:0]**：超时时长 (Timeout period)
在内存映射模式下，每次访问结束后，QUADSPI 将预取后续字节并将其存放在 FIFO 中。此位域指示在 FIFO 写满后，QUADSPI 等待多少个 CLK 时钟周期才让 nCS 升至高电平将 Flash 置为低功耗状态。
仅可在 BUSY = 0 时写入此位域。

15.5.14 QUADSPI 寄存器映射

表 68. QUADSPI 寄存器映射和复位值

偏移	寄存器名称	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
0x0000	**QUADSPI_CR**	PRESCALER[7:0]								PMM	APMS	Res.	TOIE	SMIE	FTIE	TCIE	TEIE	Res.	Res.	Res.	Res.	FTHRES [3:0]				Res.	Res.	Res.	SSHIFT	TCEN	DMAEN	ABORT	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
0x0004	**QUADSPI_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSIZE[4:0]					Res.	Res.	Res.	Res.	Res.	CSHT			Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKMODE
	Reset value												0	0	0	0	0						0	0	0								0
0x0008	**QUADSPI_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLEVEL[5:0]						Res.	Res.	BUSY	TOF	SMF	FTF	TCF	TEF
	Reset value																			0	0	0	0	0	0			0	0	0	0	0	0
0x000C	**QUADSPI_FCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTOF	CSMF	Res.	CTCF	CTEF
	Reset value																												0	0		0	0
0x0010	**QUADSPI_DLR**	DL[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
0x0014	**QUADSPI_CCR**	DDRM	Res.	Res.	SIOO	FMODE[1:0]		DMODE[1:0]		Res.	DCYC[4:0]					ABSIZE[1:0]		ABMODE[1:0]		ADSIZE[1:0]		ADMODE[1:0]		IMODE[1:0]		INSTRUCTION[7:0]							
	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
0x0018	**QUADSPI_AR**	ADDRESS[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
0x001C	**QUADSPI_ABR**	ALTERNATE[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
0x0020	**QUADSPI_DR**	DATA[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
0x0024	**QUADSPI_ PSMKR**	MASK[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
0x0028	**QUADSPI_ PSMAR**	MATCH[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
0x002C	**QUADSPI_PIR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTERVAL[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0030	**QUADSPI_ LPTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMEOUT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

16 模数转换器 (ADC)

16.1 简介

ADC 由 12 位逐次逼近模数转换器组成。

ADC 的复用通道多达 19 条。各种不同通道的 A/D 转换可在单次、连续、扫描或不连续采样模式下进行。ADC 的结果存储在一个左对齐或右对齐的 16 位数据寄存器中。

ADC 映射到 AHB 总线，从而可实现快速数据处理。

ADC 具有模拟看门狗特性，允许应用检测输入电压是否超过了用户自定义的阈值上限或下限。

内置硬件过采样器，可提高模拟性能，同时还能减轻 CPU 进行相关计算的负担。

采用了高效的低功耗模式，在低频下可实现极低的功耗。

16.2 ADC 主要特性

- 高性能特性
  - 可配置 12 位、10 位、8 位或 6 位分辨率
  - ADC 转换时间：
    快速通道：12 位分辨率时为 0.234 µs (4.27 Ms/s)
    慢速通道：12 位分辨率时为 0.297 µs (3.37 Ms/s)
  - ADC 转换时间与 AHB 总线时钟频率无关
  - 可通过降低分辨率来缩短转换时间：10 位分辨率时为 0.203 µs
  - 可管理单端或差分输入
  - AHB 从总线接口，可实现快速数据处理
  - 自校准
  - 可独立设置各通道采样时间
  - 多达四条注入通道（对常规通道或注入通道的模拟输入分配完全可配置）
  - 硬件辅助准备注入通道的上下文，从而实现快速上下文切换
  - 数据对齐以保持内置数据一致性
  - 数据可由 DMA 管理，以实现常规通道转换
  - 4 个专用数据寄存器供注入通道使用
- 过采样器
  - 16 位数据寄存器
  - 过采样率可在 2x 到 256x 之间进行调整
  - 可编程数据最多可移位 8 位

- 低功耗特性
  - 速度自适应低功耗模式，可降低 ADC 在低频下工作时的功耗
  - 可在实现慢速总线频率应用的同时保持最佳 ADC 性能（无论 AHB 总线时钟频率为何，快速通道均可保持 0.234 µs 的转换时间）
  - 提供自动控制，可避免 ADC 在低 AHB 总线时钟频率应用中溢出（自动延迟模式）
- 多个外部模拟输入通道
  - GPIO 引脚有多达 5 条快速通道
  - GPIO 引脚有多达 11 条慢速通道
- 此外，还有多条内部专用通道
  - 内部参考电压 (V_{REFINT})
  - 内部温度传感器 (V_{TS})
  - V_{BAT} 监测通道 ($V_{BAT}/3$)
- 可通过以下方式启动转换过程：
  - 通过软件启动常规转换和注入转换
  - 通过极性可配置的硬件触发器（内部定时器事件或 GPIO 输入事件）启动常规转换和注入转换
- 转换模式
  - ADC 可转换单条通道，也可扫描一系列通道
  - 单次模式会在每次触发时对选定的输入执行一次转换
  - 连续模式可连续转换选定的输入
  - 不连续采样模式
- 在 ADC 准备就绪、采样结束、转换结束（常规转换或注入转换）、序列转换结束（常规转换或注入转换）、模拟看门狗 1/2/3 事件或溢出事件时生成中断
- 3 个模拟看门狗
- ADC 输入范围：$V_{REF-} \leq V_{IN} \leq V_{REF+}$

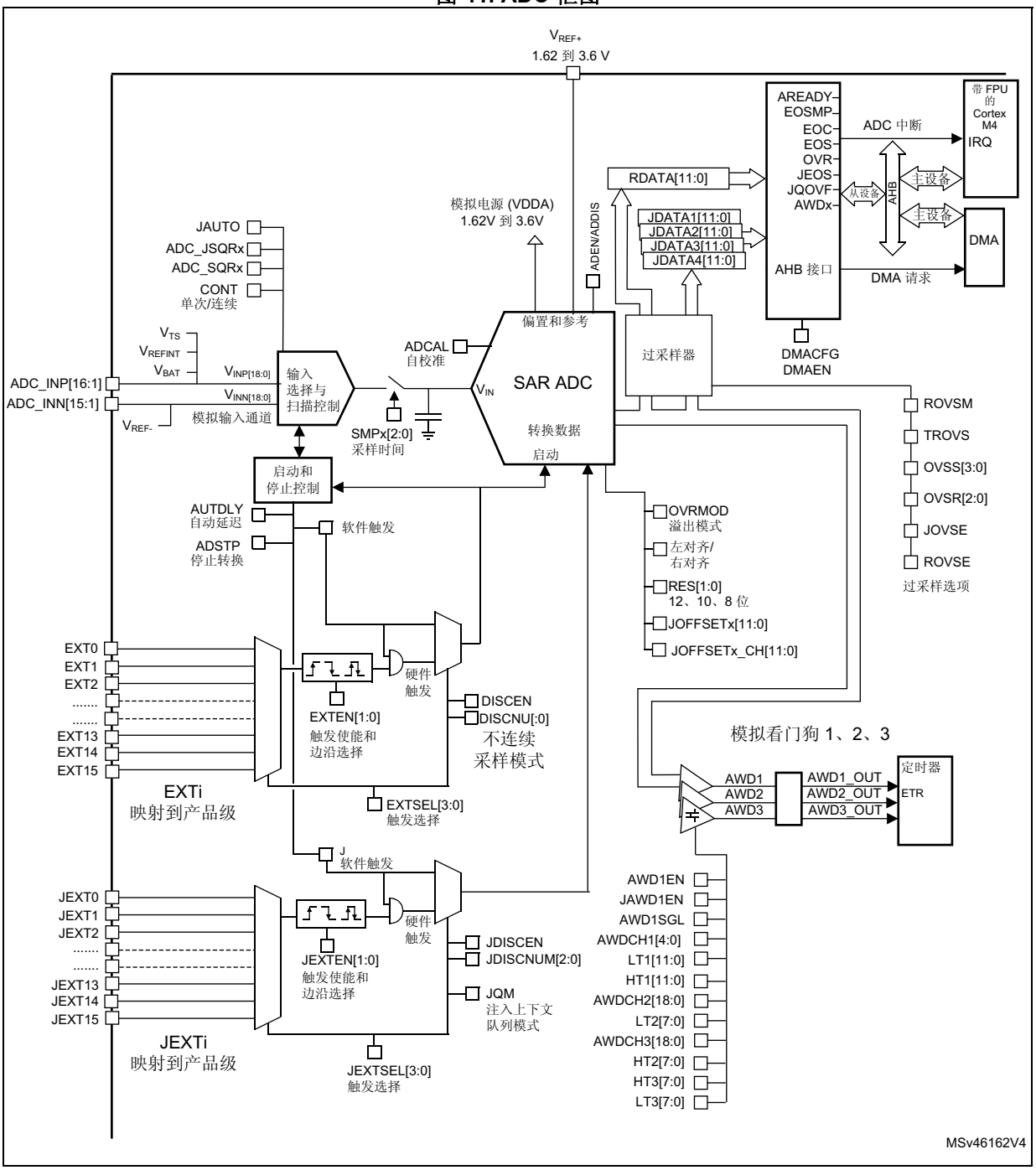
图 41 给出了一个 ADC 的框图。

16.3 ADC 功能说明

16.3.1 ADC 框图

图 41 给出了 ADC 框图，*表 70* 列出了 ADC 的引脚说明。

图 41. ADC 框图



16.3.2 ADC 引脚和内部信号

表 69. ADC 内部输入/输出信号

内部信号名称	信号类型	说明
EXT[15:0]	输入	共有多达 16 个外部触发输入用于常规转换（可连接至片上定时器）。
JEXT[15:0]	输入	共有多达 16 个外部触发输入用于注入转换（可连接至片上定时器）。
ADC_AWDx_OUT	输出	内部模拟看门狗输出信号，连接至片上定时器。（x = 模拟看门狗编号 1、2、3）
VTS	输入	内部温度传感器的输出电压
V_{REFINT}	输入	内部参考电压的输出电压
V_{BAT}	输入电源	外部电池电压

表 70. ADC 输入/输出引脚

引脚名称	信号类型	注释
V_{REF+}	正模拟参考电压输入	ADC 高/正参考电压， 1.62 V ≤ V_{REF+} ≤ V_{DDA}
V_{DDA}	模拟电源输入	模拟电源等于 V_{DDA}： 1.62 V ≤ V_{DDA} ≤ 3.6 V
V_{REF-}	负模拟参考电压输入	ADC 的低/负参考电压。V_{REF-} 在内部连接到 V_{SSA}
V_{SSA}	模拟电源地输入	模拟电源接地引脚。对于无专用 V_{SSA} 引脚的器件封装，V_{SSA} 在内部连接到 V_{SS}。
V_{INP}[18:0]	正模拟输入通道	连接到外部通道：ADC1_INP*i* 或内部通道。
V_{INN}[18:0]	负模拟输入通道	连接到 V_{REF-} 或外部通道：ADC1_INNi 和 ADC1_INP[i+1]
ADC1_INN[15:1]	外部模拟输入信号	最多 15 个模拟输入通道： – 5 个快速通道（ADC1_INN1 到 INN5） – 最多 10 个慢速通道（ADC1_INN6 到 INN15）
ADC1_INP[16:1]	外部模拟输入信号	最多 16 个模拟输入通道： – 5 个快速通道（ADC1_INP1 到 INP5） – 最多 11 个慢速通道（ADC1_INP6 到 INP16）

16.3.3 时钟

双时钟域架构

双时钟域架构意味着 ADC 时钟独立于 AHB 总线时钟。

三个 ADC 的输入时钟相同，可从两个不同的时钟源中选择（请参见*图 42：ADC 时钟方案*）：

1. ADC 时钟可为特定时钟源。它可来自于以下时钟源：
   - 系统时钟
   - PLLSAI1（单 ADC 实现）

   有关如何生成 ADC 专用时钟的更多信息，请参见 RCC 部分。要选择此时钟方案，必须将 ADCx_CCR 寄存器的 CKMODE[1:0] 位复位。
2. ADC 时钟可由 ADC 总线接口的 AHB 时钟除以一个可编程的因数（1、2 或 4）来提供。在这种模式下，可选择可编程的分频系数（根据位 CKMODE[1:0]，选择 1、2 或 4）。

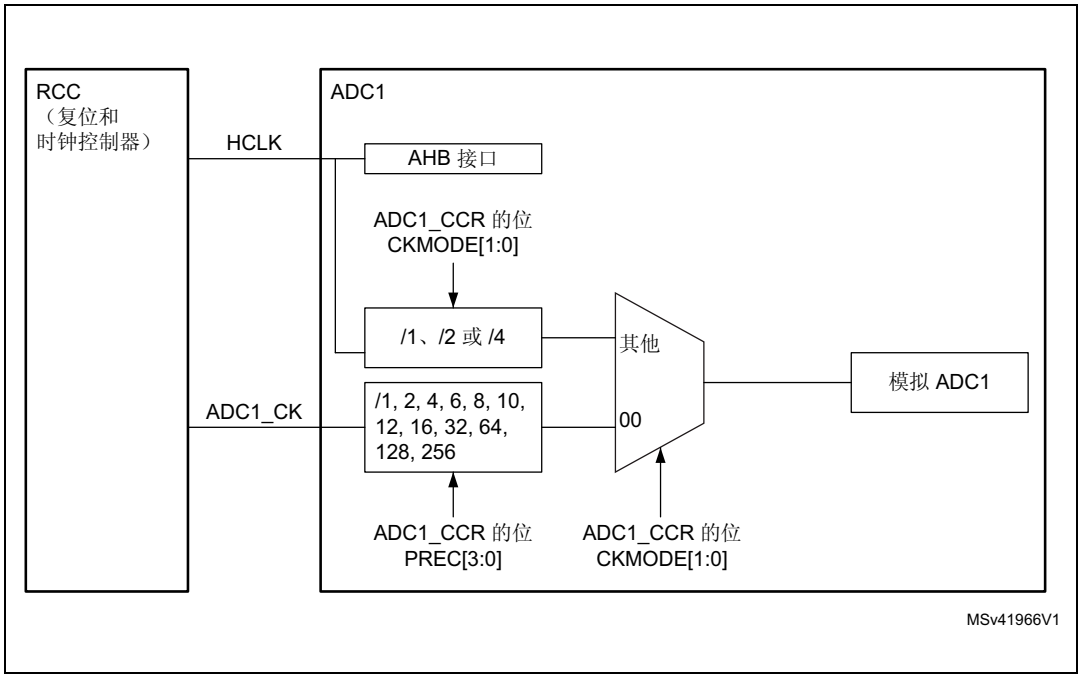
   要选择此时钟方案，ADCx_CCR 寄存器的 CKMODE[1:0] 位不得为“00”。

注： *对于选项 2)，仅当 AHB 预分频器置 1（RCC_CFGR 寄存器中的 HPRE[3:0] = 0xxx）时，才能使用预分频系数 1 (CKMODE[1:0]=01)。*

选项 1) 的优点在于无论选择哪种 AHB 时钟方案，都可以达到最大 ADC 时钟频率。使用通过 ADCx_CCR 寄存器中的 PRESC[3:0] 位配置的预分频器时，ADC 时钟最后会除以以下比率：1、2、4、6、8、12、16、32、64、128、256。

选项 2) 的优势在于绕过了时钟域重新同步。如果 ADC 由定时器触发，并且应用要求 ADC 精确触发（不存在任何不确定性），可使用此选项（否则，重新同步两个时钟域会为触发时刻带来不确定性）。

图 42. ADC 时钟方案



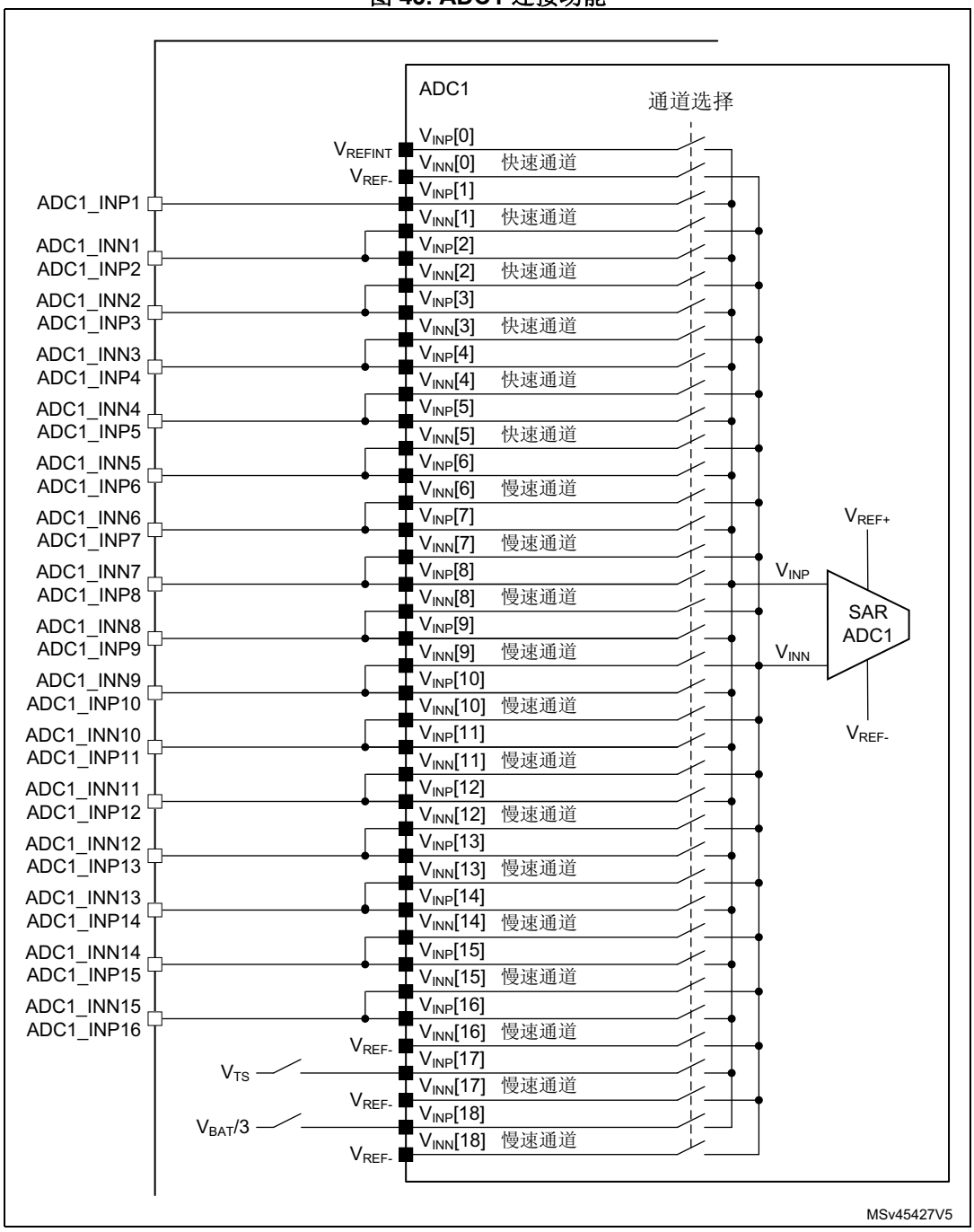
ADC 时钟与 AHB 时钟的时钟比例限制

通常来讲，ADC 时钟与 AHB 时钟之比没有限制，但一些注入通道已编程的情况除外，此时，ADC 时钟与 AHB 时钟之比必须遵循以下规定：

- 如果所有通道的分辨率均为 12 位或 10 位，则 $F_{HCLK} >= F_{ADC}/4$
- 如果一些通道的分辨率为 8 位（且所有通道的分辨率均不低于 8 位），则 $F_{HCLK} >= F_{ADC}/3$
- 如果一些通道的分辨率为 6 位，则 $F_{HCLK} >= F_{ADC} / 2$

16.3.4 ADC1 连接功能

图 43. ADC1 连接功能



16.3.5 从设备 AHB 接口

ADC 利用 AHB 从设备端口实现控制/状态寄存器访问和数据访问。AHB 接口的特性如下：

- 字（32 位）访问
- 单周期响应
- 以零等待状态响应对寄存器的所有读/写访问

AHB 从设备接口不支持分离/重试请求，且绝不会生成 AHB 错误。

16.3.6 ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN)

ADC 默认处于深度掉电模式，在该模式下，其电源会在内部切断以降低泄漏电流（ADC_CR 寄存器中的 DEEPPWD 位的复位状态为 1）。

要开始 ADC 操作，需要先将 DEEPPWD 位清零，使 ADC 退出深度掉电模式。

然后，必须将 ADC_CR 寄存器中的 ADVREGEN 位置 1，以使能 ADC 内部稳压器。软件必须在 ADC 稳压器的启动时间 ($T_{ADCVREG\_STUP}$) 内等待，之后才能开始校准或使能 ADC。该延迟必须通过软件实现。

有关 ADC 稳压器的启动时间，请参见器件数据手册中的 $T_{ADCVREG\_STUP}$ 参数。

ADC 操作完成后，可禁止 ADC (ADEN=0)。此外，还可通过禁止 ADC 稳压器实现节能。这可通过向 ADVREGEN 位写入 0 来实现。

之后，为了通过降低泄漏电流的方式进一步实现节能，还可以将 ADC_CR 寄存器中的 DEEPPWD 位置 1，以使 ADC 再次进入深度掉电模式。进入停止模式之前，这一点特别值得关注。

注： *写入 DEEPPWD=1 会自动禁止 ADC 稳压器，并且 ADVREGEN 位会自动清零。*
如果内部稳压器已禁止 (ADVREGEN=0)，则会保持内部模拟校准。

在 ADC 深度掉电模式下 (DEEPPWD=1)，内部模拟校准会丢失，需要重新启动校准或重新应用之前保存的校准系数（请参见*第 16.3.8 节：校准（ADCAL、ADCALDIF、ADC_CALFACT）*）。

16.3.7 单端输入通道和差分输入通道

可通过写入 ADC_DIFSEL 寄存器中的 DIFSEL[15:1] 位将通道配置为单端输入或差分输入。必须在禁止 ADC (ADEN=0) 时写入此配置。请注意，DIFSEL[18:16,0] 固定为单端通道，始终读为 0。

在单端输入模式下，要通过通道“i”转换的模拟电压是外部电压 $V_{INP[i]}$（正输入）和 V_{REF-}（负输入）之差。

在差分输入模式下，要通过通道“i”转换的模拟电压是外部电压 $V_{INP[i]}$（正输入）与 $V_{INN[i]}$（负输入）之差。

当 ADC 配置为差分模式时，两路输入的偏置电压均应为 $(V_{REF+})/2$。

输入信号应为差分信号（共模电压应固定）。

有关输入通道连接方式的完整说明，请参见*第 16.3.4 节：ADC1 连接功能*。

注意： 将通道“i”配置为差分输入模式时，其负输入电压 $V_{INN[i]}$ 连接至 $V_{INP[i+1]}$。因此，通道“*i+1*”不再可用于单端模式或差分模式下，并且不得配置为进行转换。

注： *由于 ADC 通道 0、16、17、18 连接至单端外部模拟输入或内部通道，因此其强制采用单端配置（相应的 DIFSEL[i] 位始终为零）。*

16.3.8 校准（ADCAL、ADCALDIF、ADC_CALFACT）

ADC 提供有自动校准过程，用于驱动包括 ADC 上电/掉电序列在内的所有校准序列。在校准过程中，ADC 会计算 7 位宽的校准系数，并会在下一次掉电之前在自身内部应用该值。在校准过程中，应用不得使用 ADC，必须等待至校准完成。

执行任何 ADC 操作之前都必须进行校准。校准可消除偏移误差，由于制造工艺或带隙的不同，各芯片的偏移误差也有所不同。

单端输入转换与差分输入转换应用的校准系数有所不同：

- 启动要为单端输入转换应用的校准之前，写入 ADCALDIF=0。
- 启动要为差分输入转换应用的校准之前，写入 ADCALDIF=1。

随后，通过软件将 ADCAL 位置 1 发起校准。仅当 ADC 禁止 (ADEN=0) 后，才能发起校准。ADCAL 位在所有校准序列过程中保持为 1。校准完成后，此位会立即由硬件清零。此时，相关校准系数会存储在模拟 ADC 内部，同时还会存储在 ADC_CALFACT 寄存器的 CALFACT_S[6:0] 或 CALFACT_D[6:0] 位中（具体取决于是单端输入校准还是差分输入校准）

如果禁止 ADC (ADEN=0)，则会保留内部模拟校准。但如果长时间禁止 ADC，建议在重新使能 ADC 之前运行新的校准周期。

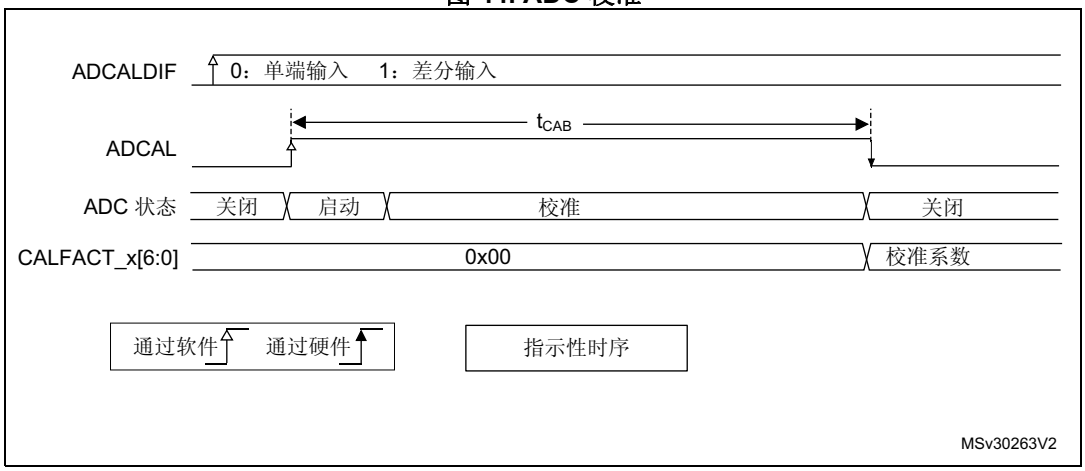
每次 ADC 掉电时，内部模拟校准都会丢失（例如，产品进入 STANDBY 模式或 VBAT 模式时）。这种情况下，为了避免浪费时间重新校准 ADC，可将校准系数重新写入 ADC_CALFACT 寄存器，但前提是软件之前保存了上次校准时得出的校准系数。

如果 ADC 已使能但未进行转换（ADEN=1、ADSTART=0 且 JADSTART=0），可写入校准系数。随后，下次转换启动时，校准系数会自动添加到模拟 ADC 中。这一载入过程是透明的，不会对转换的启动造成延迟。建议在 V_{REF+} 电压变化超过 10% 时重新进行校准。

ADC 校准的软件流程

1. 确保 DEEPPWD=0、ADVREGEN=1，并确保 ADC 稳压器启动时间已过。
2. 确保 ADEN=0。
3. 设置 ADCALDIF=0（单端输入）或 ADCALDIF=1（差分输入），选择此校准的输入模式。
4. 将 ADCAL 置 1。
5. 等待 ADCAL=0。
6. 可从 ADC_CALFACT 寄存器读取校准系数。

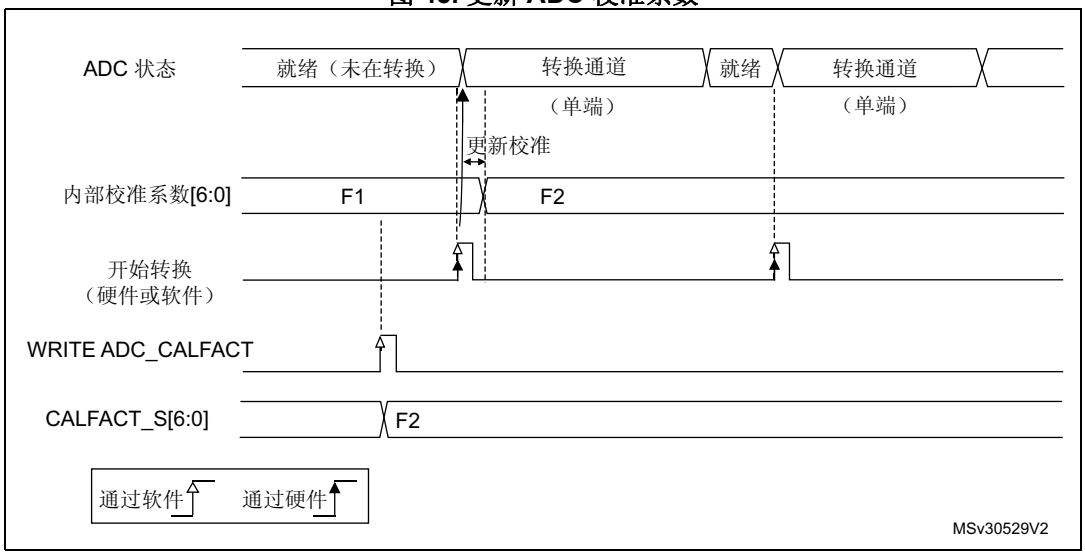
图 44. ADC 校准



将校准系数重新注入 ADC 的软件流程

1. 确保 ADEN=1、ADSTART=0 且 JADSTART=0（ADC 已使能，且未进行任何转换）。
2. 将新的校准系数写入 CALFACT_S 和 CALFACT_D。
3. 启动转换时，仅当内部模拟校准系数与 CALFACT_S 位（单端输入通道）或 CALFACT_D 位（差分输入通道）中存储的值不同时，校准系数才会注入到模拟 ADC。

图 45. 更新 ADC 校准系数

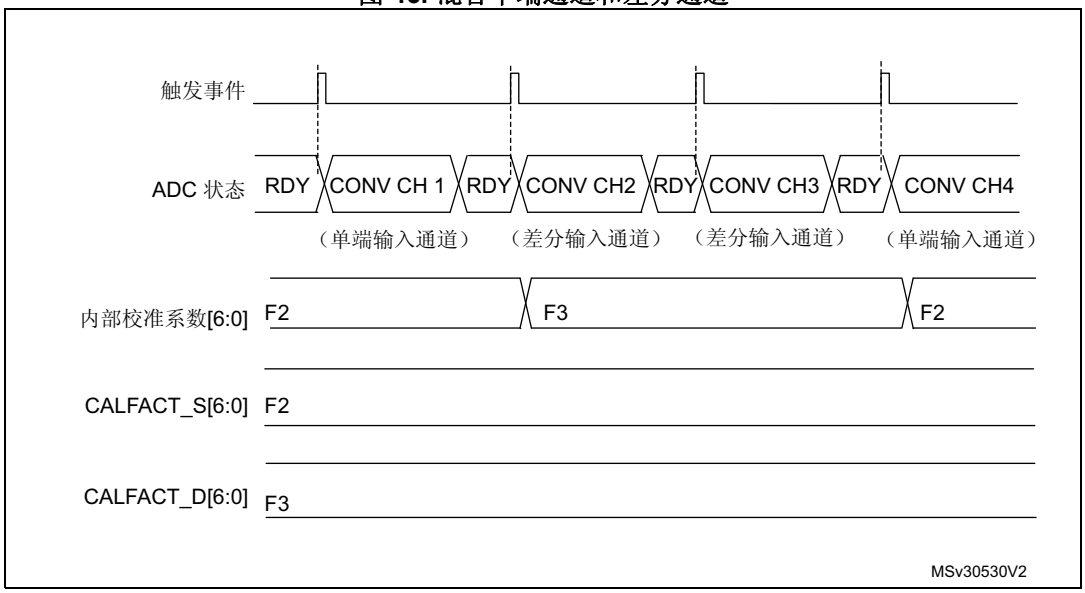


单个 ADC 转换单端模拟输入和差分模拟输入

如果 ADC 既要转换差分输入，也要转换单端输入，则必须执行两次校准，两次校准分别在 ADCALDIF=0 和 ADCALDIF=1 的情况下执行。操作流程如下：

1. 禁止 ADC。
2. 在单端输入模式下 (ADCALDIF=0) 校准 ADC。此操作会更新寄存器 CALFACT_S[6:0]。
3. 在差分输入模式下 (ADCALDIF=1) 校准 ADC。此操作会更新寄存器 CALFACT_D[6:0]。
4. 使能 ADC、配置通道并启动转换。每次从单端输入通道切换为差分输入通道（或进行相反切换）时，校准都将自动注入到模拟 ADC 中。

图 46. 混合单端通道和差分通道



16.3.9 ADC 开关控制（ADEN、ADDIS、ADRDY）

首先按照*第 16.3.6 节：ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN)* 中的流程执行操作。

DEEPPWD=0 且 ADVREGEN=1 后，可使能 ADC，并且 ADC 在开始精确转换之前需要 t_{STAB} 的稳定时间，如*图 47* 所示。以下两个控制位可使能或禁止 ADC：

- 将 ADEN 置 1 可使能 ADC。ADC 准备就绪后，ADRDY 标志会立即置 1。
- 将 ADDIS 置 1 可禁止 ADC。随后，模拟 ADC 被完全禁止后，ADEN 位和 ADDIS 位会自动由硬件清零。

随后可通过将 ADSTART 置 1（请参见*第 16.3.18 节：外部触发转换和触发极性（EXTSEL、EXTEN、JEXTSEL、JEXTEN）*）开始进行常规转换，如果触发器已使能，也可在发生外部触发事件时开始进行常规转换。

可通过将 JADSTART 置 1 开始进行注入转换，如果注入触发器已使能，也可在发生外部注入触发事件时开始进行注入转换。

通过软件使能 ADC 的流程

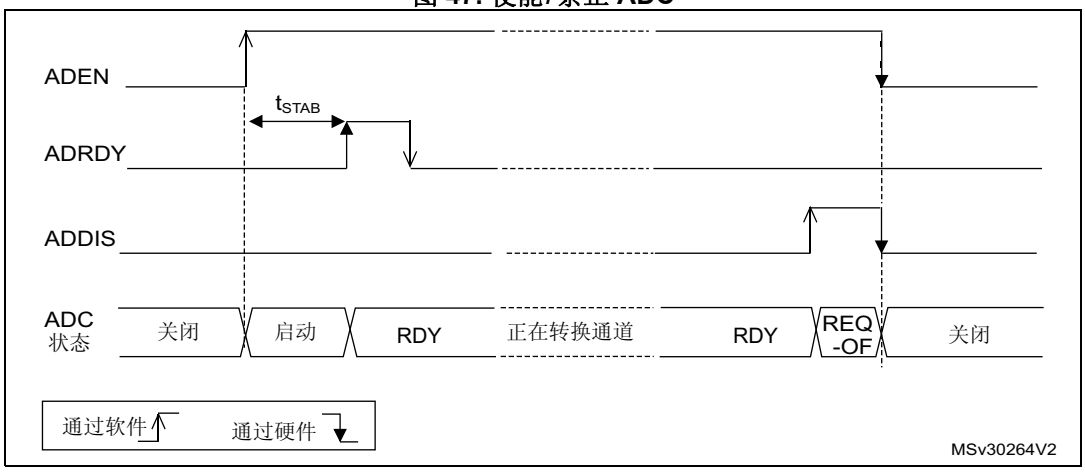
1. 向 ADC_ISR 寄存器中的 ADRDY 位写入“1”，将其清零。
2. 将 ADEN 置 1。
3. 等待，直至 ADRDY=1（ADRDY 会在 ADC 启动时间后置 1）。这可以使用关联中断来实现（将 ADRDYIE 置 1）。
4. 向 ADC_ISR 寄存器中的 ADRDY 位写入“1”，将其清零（可选）。

注意： ADCAL=1 期间和硬件清零 ADCAL 位后（校准结束）的 4 个 ADC 时钟周期内，ADEN 位无法置 1。

通过软件禁止 ADC 的流程

1. 检查 ADSTART 和 JADSTART 是否均为 0，以确保当前未执行任何转换。如有需要，可将 ADSTP 置 1，并将 JADSTP 置 1，随后等待至 ADSTP=0 且 JADSTP=0，以停止任何正在进行的常规转换和注入转换。
2. 将 ADDIS 置 1。
3. 如果应用要求，可等待 ADEN=0，直至模拟 ADC 已有效禁止（ADEN=0 后，ADDIS 将自动复位）。

图 47. 使能/禁止 ADC



16.3.10 写入 ADC 控制位时的限制

仅当 ADC 已禁止（ADEN 必须等于 0）时，软件才可通过写入 RCC 控制位的方式配置和使能 ADC 时钟（请参见 RCC 部分）、ADC_DIFSEL 寄存器中的控制位 DIFSEL、ADC_CR 寄存器中的控制位 ADCAL 和 ADEN。

只有在 ADC 已使能、并且没有待处理的禁止 ADC 的请求（ADEN 必须等于 1 且 ADDIS 必须等于 0）时，才允许软件向 ADC_CR 寄存器的控制位 ADSTART、JADSTART 和 ADDIS 执行写操作。

对于 ADC_CFGR、ADC_SMPRx、ADC_SQRy、ADC_JDRy、ADC_OFRy、ADC_OFCHRy 和 ADC_IER 寄存器的所有其他控制位：

- 对于与常规转换配置相关的控制位，仅当 ADC 已使能 (ADEN=1) 且未进行常规转换（ADSTART 必须等于 0）时，才允许软件对这些位执行写操作。
- 对于与注入转换配置相关的控制位，仅当 ADC 已使能 (ADEN=1) 且未进行注入转换（JADSTART 必须等于 0）时，才允许软件对这些位执行写操作。

仅当 ADC 已使能（可能会进行转换）且没有待处理的禁止 ADC 的请求（ADSTART 或 JADSTART 必须等于 1 且 ADDIS 必须等于 0）时，软件才可以对 ADC_CR 寄存器的 ADSTP 或 JADSTP 控制位执行写操作。

如果 ADC 已使能 (ADEN=1)，软件可随时对 ADC_JSQR 寄存器执行写操作。更多详细信息，请参见*第 16.5.16 节：ADC 注入序列寄存器 (ADC_JSQR)*。

注： *没有硬件保护可以防止执行此类禁止的写操作，ADC 操作可能进入未知状态。要从此状态恢复，必须禁止 ADC（将 ADEN 清零，并且将 ADC_CR 寄存器的所有位都清零）。*

16.3.11 通道选择（SQRx、JSQRx）

复用通道多达 19 条：

- 5 路来自 GPIO 焊盘的快速模拟输入 (ADC1_INP/INN[1..5])
- 多达 10 路来自 GPIO 焊盘的慢速模拟输入 (ADC1_INP/INN[6..15])。并非所有这些输入都连到 GPIO 的焊盘，具体取决于产品。
- ADC 连接到以下内部模拟输入：
  - 内部参考电压 (V_{REFINT}) 连接到 ADC1_INP0。
  - 内部温度传感器 (V_{TS}) 连接到 ADC1_INP17。
  - V_{BAT} 监测通道 ($V_{BAT}/3$) 连接到 ADC1_INP18。

注： *要对其中一条内部模拟通道进行转换，必须先对 ADCx_CCR 寄存器中的 VREFEN、CH17SEL 或 CH18SEL 位进行编程，以使能相应的模拟源。*

可以将转换分为两组：常规转换和注入转换。每个组包含一个转换序列，该序列可按任意顺序在任意通道上完成。例如，可按以下顺序对序列进行转换：ADC1_INP/INN3、ADC1_INP/INN8、ADC1_INP/INN2、ADC1/INN2、ADC1_INP/INN0、ADC1_INP/INN2、ADC1_INP/INN2、ADC1_INP/INN15。

- 一个**常规转换组**最多由 16 个转换构成。必须在 ADC_SQRy 寄存器中选择转换序列的常规通道及其顺序。常规转换组中的转换总数必须写入 ADC_SQR1 寄存器中的 L[3:0] 位。
- 一个**注入转换组**最多由 4 个转换构成。必须在 ADC_JSQR 寄存器中选择转换序列的注入通道及其顺序。注入转换组中的转换总数必须写入 ADC_JSQR 寄存器中的 L[1:0] 位。

不得在进行常规转换时对 ADC_SQRy 寄存器进行修改。因此，必须先写入 ADSTP=1 停止 ADC 常规转换（请参见*第 16.3.17 节：停止正在进行的转换（ADSTP、JADSTP）*）。

仅在使能上下文队列（ADC_CFGR 寄存器中的 JQDIS=0）时，才会在 JADSTART 置 1（正在进行注入转换）的条件下，允许软件实时修改 ADC_JSQR 寄存器。请参见*第 16.3.21 节：注入转换的上下文队列*。

16.3.12 可独立设置各通道采样时间（SMPR1、SMPR2）

开始转换之前，ADC 必须在待测量电压源与 ADC 内置采样电容之间建立直接连接。该采样时间必须足以使输入电压源为嵌入式电容充电至输入电压水平。

对各通道进行采样时可以使用不同的采样时间，采样时间可通过 ADC_SMPR1 和 ADC 寄存器中的 SMP[2:0] 位编程。可选采样时间值如下：

- SMP = 000：2.5 个 ADC 时钟周期
- SMP = 001：6.5 个 ADC 时钟周期
- SMP = 010：12.5 个 ADC 时钟周期
- SMP = 011：24.5 个 ADC 时钟周期
- SMP = 100：47.5 个 ADC 时钟周期
- SMP = 101：92.5 个 ADC 时钟周期
- SMP = 110：247.5 个 ADC 时钟周期
- SMP = 111：640.5 个 ADC 时钟周期

总转换时间的计算公式如下：

T_{CONV} = 采样时间 + 12.5 个 ADC 时钟周期

示例：

如果 F_{ADC_CLK} = 64 MHz，采样时间为 2.5 个 ADC 时钟周期：

T_{CONV} = (2.5 + 12.5) 个 ADC 时钟周期 = 15 个 ADC 时钟周期 = 234.4 ns（对于快速通道）

ADC 通过将状态位 EOSMP 置 1 来指示采样阶段结束（仅限常规转换）。

快速通道和慢速通道的采样时间限制

对于每条通道，必须对 SMP[2:0] 位进行编程，以符合数据手册 ADC 特性部分规定的最短采样时间要求。

I/O 模拟开关升压器

当 V_{DDA} 电压过低时，I/O 模拟开关电阻会增大，此时需要相应地调整采样时间（请参见数据手册中的电气特性）。可通过 SYSCFG_CFGR1 寄存器中的 BOOSTEN 位使能内部升压器，从而在低 V_{DDA} 条件下最大限度减小该电阻值。

16.3.13 单次转换模式 (CONT=0)

在单次转换模式下，ADC 会将通道的所有转换执行一次。CONT 位为 0 时，可通过以下方式启动此模式：

- 将 ADC_CR 寄存器中的 ADSTART 位置 1（适用于常规通道）
- 将 ADC_CR 寄存器中的 JADSTART 位置 1（适用于注入通道）
- 外部硬件触发事件（适用于常规通道或注入通道）

在常规序列中，每次转换完成后：

- 转换数据存储在 16 位 ADC_DR 寄存器中
- EOC（常规转换结束）标志置 1
- EOCIE 位置 1 时将产生中断

在注入序列中，每次转换完成后：

- 转换数据存储在四个 16 位 ADC_JDRy 寄存器的其中一个寄存器中
- JEOC（注入转换结束）标志置 1
- JEOCIE 位置 1 时将产生中断

常规序列完成后：

- EOS（常规序列结束）标志置 1
- EOSIE 位置 1 时将产生中断

注入序列完成后：

- JEOS（注入序列结束）标志置 1
- JEOSIE 位置 1 时将产生中断

随后，ADC 会停止工作，直至发生新的外部常规或注入触发，或者 ADSTART 或 JADSTART 位再次置 1。

注： *要转换单个通道，可将序列长度编程为 1。*

16.3.14 连续转换模式 (CONT=1)

该模式仅适用于常规通道。

在连续转换模式下，如果发生软件或硬件常规触发事件，ADC 会将通道的所有常规转换执行一次，随后会自动重启并持续执行序列的每个转换。CONT 位为 1 时，可通过外部触发或将 ADC_CR 寄存器中的 ADSTART 位置 1 来启动此模式。

在常规序列中，每次转换完成后：

- 转换数据存储在 16 位 ADC_DR 寄存器中
- EOC（转换结束）标志置 1
- EOCIE 位置 1 时将产生中断

转换序列完成后：

- EOS（序列结束）标志置 1
- EOSIE 位置 1 时将产生中断

随后，会立即重启新序列，ADC 会继续重复执行转换序列。

注： *要转换单个通道，可将序列长度编程为 1。*

不能同时使能不连续模式和连续模式：禁止同时将 DISCEN 和 CONT 位置 1。

注入通道不能连续转换，唯一例外的是，在连续转换模式下（使用 JAUTO 位）注入通道配置为在常规通道后的自动转换，请参见自动注入模式一节。

16.3.15 开始转换（ADSTART、JADSTART）

软件通过将 ADSTART 置 1 的方式开始进行 ADC 常规转换。

ADSTART 置 1 后，会开始进行转换：

- 立即开始转换：EXTEN = 0x0 时（软件触发）
- 在所选常规硬件触发的下一有效边沿开始转换：EXTEN 不等于 0x0 时

软件通过将 JADSTART 置 1 的方式开始进行 ADC 注入转换。

JADSTART 置 1 后，会开始进行转换：

- 立即开始转换：JEXTEN = 0x0 时（软件触发）
- 在所选注入硬件触发的下一有效边沿开始转换：JEXTEN 不等于 0x0 时

注： *在自动注入模式下 (JAUTO=1)，使用 ADSTART 位开始常规转换，然后再进行自动注入转换（JADSTART 必须保持清零）。*

ADSTART 位和 JADSTART 位还提供当前是否正在进行 ADC 操作的信息。可以在 ADSTART=0 且 JADSTART=0（指示 ADC 处于空闲状态）时重新配置 ADC。

ADSTART 通过硬件清零：

- 在使用软件常规触发的单次模式下（CONT=0，EXTSEL=0x0）
  - 如果 DISCEN = 1，只要常规转换序列结束（EOS 置 1）或子组处理结束就清零
- 在所有情况下（CONT=x、EXTSEL=x）
  - 执行由软件调用的 ADSTP 程序之后清零

注： *在连续模式下 (CONT=1)，由于序列会自动重新启动，因此，当 EOS 置 1 时，ADSTART 位不会通过硬件清零。*

如果在单次模式下选择了硬件触发（CONT=0 且 EXTSEL 不等于 0x00），当 EOS 置 1 时，ADSTART 位不会通过硬件清零，借此软件无需再次为下一个硬件触发事件复位 ADSTART。这样可确保不会错过任何后续的硬件触发。

JADSTART 通过硬件清零：

- 在使用软件注入触发的单次模式下 (JEXTSEL=0x0)
  - 如果 JDISCEN = 1，只要注入转换序列结束（JEOS 置 1）或子组处理结束就清零
- 在所有情况下 (JEXTSEL=x)
  - 执行由软件调用的 JADSTP 程序之后清零

注：选择软件触发时，如果 EOC 标志仍为高电平，则不应将 ADSTART 位置 1。

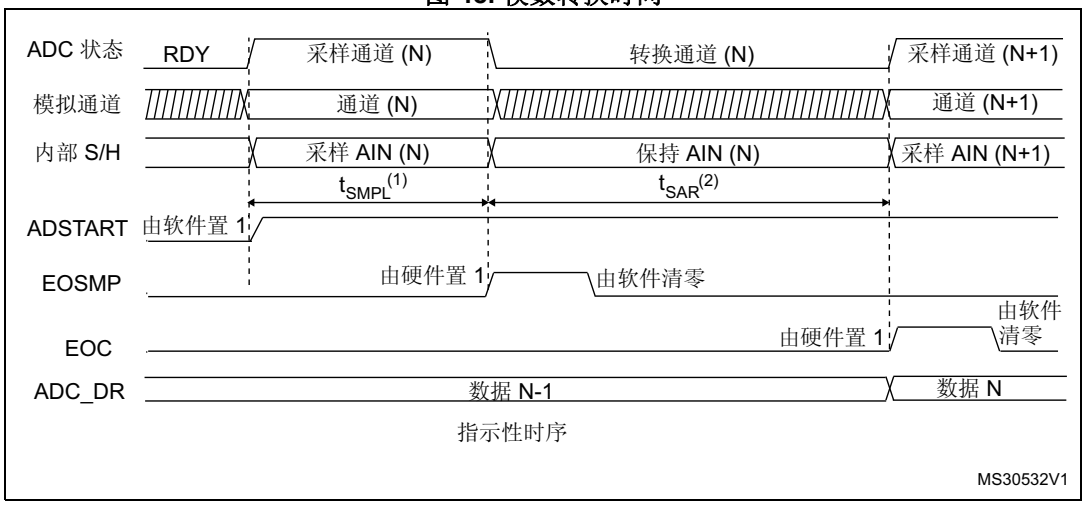
16.3.16 ADC 时序

从转换开始到转换结束所经过的时间是配置的采样时间与逐次逼近时间（具体取决于数据分辨率）的总和：

$T_{CONV} = T_{SMPL} + T_{SAR} = [2.5\ _{|min} + 12.5\ _{|12\ 位}] \times T_{ADC\_CLK}$

$T_{CONV} = T_{SMPL} + T_{SAR} = 39.06\ ns\ _{|min} + 195.31\ ns\ _{|12\ 位} = 234.38\ ns$（对于 $F_{ADC\_CLK} = 64\ MHz$）

图 48. 模数转换时间



1. T_{SMPL} 取决于 SMP[2:0]。
2. T_{SAR} 取决于 RES[2:0]。

16.3.17 停止正在进行的转换（ADSTP、JADSTP）

软件决定是否停止转换，要停止正在进行的常规转换，应将 ADSTP 置 1；要停止正在进行的注入转换，应将 JADSTP 置 1。

停止转换将复位正在进行的 ADC 操作。随后可重新配置 ADC（例如：更改通道选择或触发），为新操作做好准备。

请注意，可以在常规转换仍在执行时停止注入转换，反之亦然。这样便可在常规转换仍在进行时重新配置注入转换序列和触发（反之亦然）。

如果 ADSTP 位由软件置 1，则会中止任何正在进行的常规转换，并会丢弃部分转换结果（ADC_DR 寄存器不会更新为当前转换结果）。

如果 JADSTP 位由软件置 1，则会中止任何正在进行的注入转换，并会丢弃部分转换结果（ADC_JDRy 寄存器不会更新为当前转换结果）。扫描序列也会中止并会复位（这意味着重启 ADC 将重新开始新的序列）。

该程序执行完毕后，ADSTP/ADSTART 位（常规转换时）或 JADSTP/JADSTART 位（注入转换时）会由硬件清零，软件必须轮询 ADSTART（或 JADSTART）直至其复位，然后才能判定 ADC 已完全停止运行。

注： *在自动注入模式下 (JAUTO=1)，将 ADSTP 位置 1 会中止常规转换和注入转换（不得使用 JADSTP）。*

图 49. 停止正在进行的常规转换

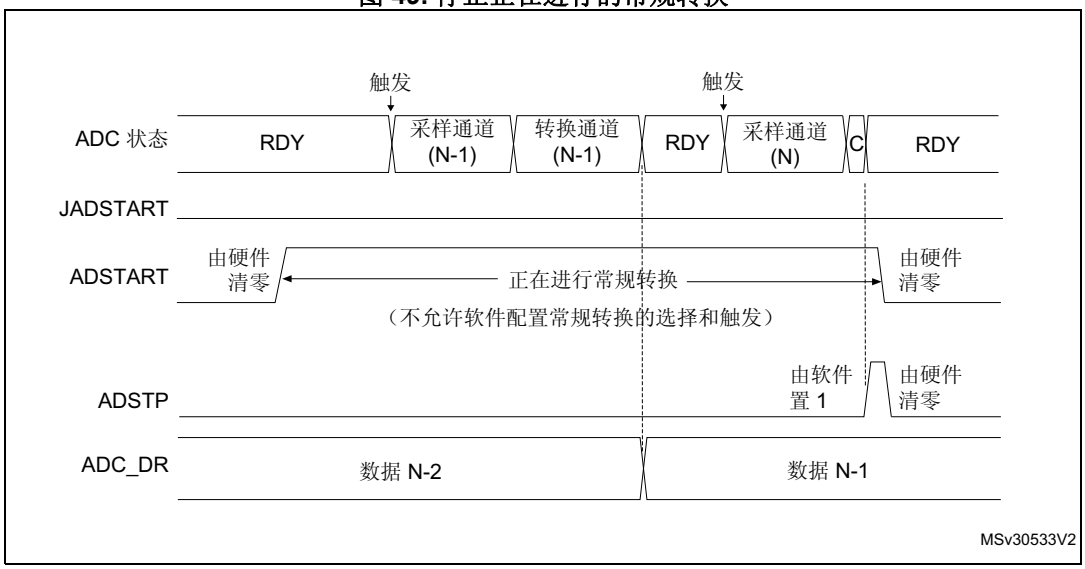
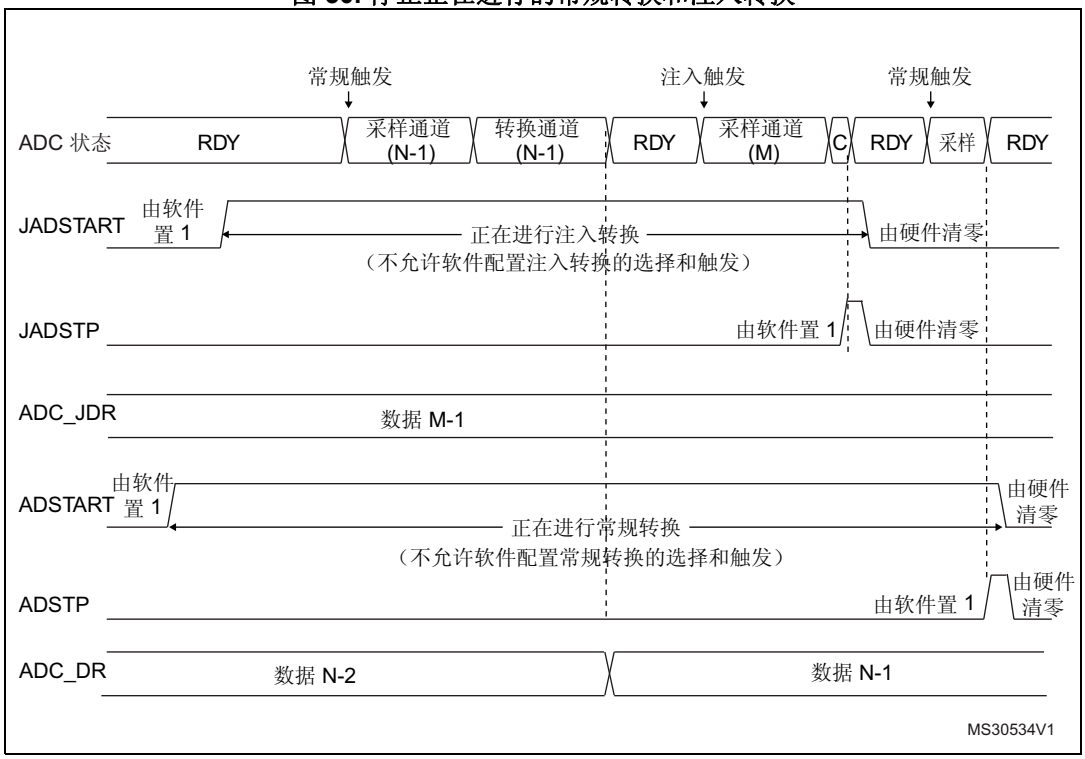


图 50. 停止正在进行的常规转换和注入转换



16.3.18 外部触发转换和触发极性（EXTSEL、EXTEN、JEXTSEL、JEXTEN）

可通过软件或外部事件（例如定时器捕获、输入引脚）触发转换或转换序列。如果 EXTEN[1:0] 控制位（对于常规转换）或 JEXTEN[1:0] 位（对于注入转换）不等于“0b00”，则外部事件能够以所选极性触发转换。

注入队列使能时（位 JQDIS=0），不能实现注入软件触发。

软件将 ADSTART 位置 1 后，常规触发选择有效；软件将 JADSTART 位置 1 后，注入触发选择有效。

在转换进行时发生的硬件触发会被忽略。

- 如果位 ADSTART=0，则会忽略发生的任何常规硬件触发。
- 如果位 JADSTART=0，则会忽略发生的任何注入硬件触发。

表 71 提供了 EXTEN[1:0] 和 JEXTEN[1:0] 值与触发极性之间的对应关系。

表 71. 为常规外部触发配置触发极性

EXTEN[1:0]	源
00	禁止硬件触发检测，使能软件触发检测
01	在上升沿执行硬件触发检测
10	在下降沿执行硬件触发检测
11	在上升沿和下降沿均执行硬件触发检测

注： *不能实时更改常规触发的极性。*

表 72. 为注入外部触发配置触发极性

JEXTEN[1:0]	源
00	– 如果 JQDIS=1（队列禁止）：禁止硬件触发检测，使能软件触发检测 – 如果 JQDIS=0（队列使能）：同时禁止硬件和软件触发检测
01	在上升沿执行硬件触发检测
10	在下降沿执行硬件触发检测
11	在上升沿和下降沿均执行硬件触发检测

注： *如果队列已使能 (JQDIS=0)，可预计并实时更改注入触发的极性。请参见第 16.3.21 节：注入转换的上下文队列。*

EXTSEL 和 JEXTSEL 控制位用于从 16 个可能事件中选择可触发常规组转换和注入组转换的事件。

可通过注入触发中断常规组转换。

注： *不能实时更改常规触发选择。*
可以预计并实时更改注入触发选择。请参见第 423 页的第 16.3.21 节：注入转换的上下文队列。

表 73 到*表 74* 列出了用于常规转换和注入转换的三个 ADC 的所有可能存在的外部触发。

表 73. ADC1——常规通道的外部通道

名称	源	类型	EXTSEL[3:0]
EXT0	TIM1_CC1	片上定时器的内部信号	0000
EXT1	TIM1_CC2	片上定时器的内部信号	0001
EXT2	TIM1_CC3	片上定时器的内部信号	0010
EXT3	TIM2_CC2	片上定时器的内部信号	0011
EXT4	-	-	0100
EXT5	-	-	0101
EXT6	EXTI 线 11	外部引脚	0110
EXT9	TIM1_TRGO	片上定时器的内部信号	1001
EXT10	TIM1_TRGO2	片上定时器的内部信号	1010
EXT11	TIM2_TRGO	片上定时器的内部信号	1011
EXT13	-	-	1101
EXT14	-	-	1110

表 74. ADC1——注入通道的外部通道

名称	源	类型	JEXTSEL[3..0]
JEXT0	TIM1_TRGO	片上定时器的内部信号	0000
JEXT1	TIM1_CC4	片上定时器的内部信号	0001
JEXT2	TIM2_TRGO	片上定时器的内部信号	0010
JEXT3	TIM2_CH1	片上定时器的内部信号	0011
JEXT4	-	-	0100
JEXT5	-	-	0101
JEXT6	EXTI 线 15	外部引脚	0110
JEXT8	TIM1_TRGO2	片上定时器的内部信号	1000
JEXT14	-	-	1110
JEXT15	-	-	1111

16.3.19 注入通道管理

触发注入模式

要使用触发注入，必须将 ADC_CFGR 寄存器中的 JAUTO 位清零。

1. 通过外部触发或将 ADC_CR 寄存器中的 ADSTART 位置 1 来启动常规通道组转换。
2. 如果在常规通道组转换期间出现外部注入触发或者 ADC_CR 寄存器中的 JADSTART 位置 1，则当前的转换会复位，并会启动注入通道序列切换（所有注入通道都会转换一次）。
3. 然后，常规通道组的常规转换会从上次中断的常规转换处恢复。
4. 如果在注入转换期间出现常规事件，注入转换不会中断，但在注入序列结束时会执行常规序列。*图 51* 显示了相应的时序图。

注： *使用触发注入时，必须确保触发事件之间的间隔长于注入序列。例如，如果序列长度为 28 个 ADC 时钟周期（即，采样时间为 1.5 个时钟周期的两次转换），则触发事件的最小间隔不能小于 29 个 ADC 时钟周期。*

自动注入模式

如果将 ADC_CFGR 寄存器中的 JAUTO 位置 1，则注入组中的通道会在常规组通道之后自动转换。这可用于转换最多由 20 个转换构成的序列，这些转换在 ADC_SQRy 和 ADC_JSQR 寄存器中编程。

在该模式下，必须将 ADC_CR 寄存器中的 ADSTART 位置 1 以开始常规转换，然后再进行注入转换（JADSTART 必须保持清零）。将 ADSTP 位置 1 会中止常规转换和注入转换（不得使用 JADSTP 位）。

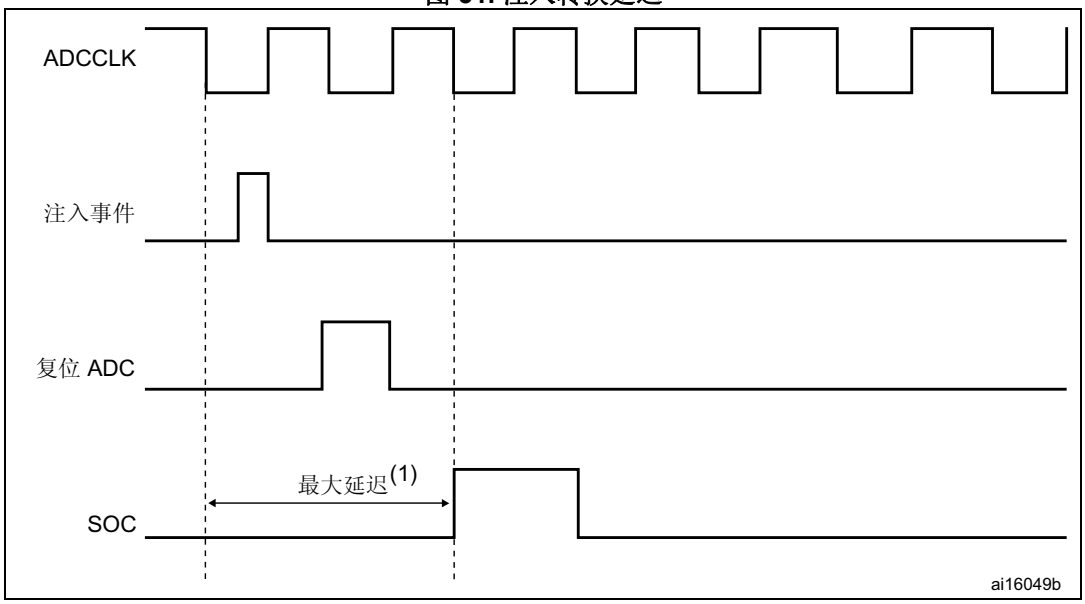
在此模式下，必须禁止注入通道上的外部触发。

如果 CONT 位和 JAUTO 位均已置 1，则常规通道及随后的注入通道会被连续转换。

注： *不能同时使用自动注入和不连续采样模式。*

如果在 JAUTO 模式下使用 DMA 导出常规定序器的数据，需要将其设定为循环模式（将 DMA_CCRx 寄存器中的 CIRC 位置 1）。如果 CIRC 位复位（单发模式），JAUTO 序列将在出现 DMA 传输完成事件时停止。

图 51. 注入转换延迟



1. 有关延迟的最大值，请参见 STM32WB55xx 数据手册中的电气特性部分。

16.3.20 不连续模式（DISCEN、DISCNUM、JDISCEN）

常规组模式

可将 ADC_CFGR 寄存器中的 DISCEN 位置 1 来使能此模式。

该模式用于转换含有 n (n ≤ 8) 个转换的短序列（子组），该短序列是在 ADC_SQRy 寄存器中选择的转换序列的一部分。可通过写入 ADC_CFGR 寄存器中的 DISCNUM[2:0] 位来指定 n 的值。

出现外部触发时，将启动在 ADC_SQRy 寄存器中选择的接下来 n 个转换，直到序列中的所有转换均完成为止。通过 ADC_SQR1 寄存器中的 L[3:0] 位定义总序列长度。

示例：

- DISCEN=1，n = 3，要转换的通道 = 1、2、3、6、7、8、9、10、11
  - 第一次触发：转换的通道为 1、2、3（每次转换时都生成 EOC 事件）。
  - 第二次触发：转换的通道为 6、7、8（每次转换时都生成 EOC 事件）。
  - 第三次触发：转换的通道为 9、10、11（每次转换时都生成 EOC 事件），并会在通道 11 转换完成后生成 EOS 事件。
  - 第四次触发：转换的通道为 1、2、3（每次转换时都生成 EOC 事件）。
  - ...
- DISCEN=0，要转换的通道 = 1、2、3、6、7、8、9、10、11
  - 第一次触发：转换整个序列：通道 1、然后是通道 2、3、6、7、8、9、10 和 11。每次转换都会生成 EOC 事件，最后一次转换还会生成 EOS 事件。
  - 所有后续触发事件都将重启整个序列。

注： *在不连续模式下转换常规组时，不会出现翻转（序列最后一个子组的转换次数少于 n 次）。转换完所有子组后，下一个触发信号将启动第一个子组的转换。在上述示例中，第四次触发重新转换了第一个子组中的通道 1、2 和 3。*

不能同时使能不连续模式和连续模式。如果同时使能两种模式（即 DISCEN=1、CONT=1），ADC 会认定连续模式已禁止并继续执行相关操作。

注入组模式

可将 ADC_CFGR 寄存器中的 JDISCEN 位置 1 来使能此模式。在出现外部注入触发事件之后，该模式会逐通道转换在 ADC_JSQR 寄存器中选择的序列，相当于不连续模式下常规通道“n”固定为 1 的情况。

出现外部触发时，将启动在 ADC_JSQR 寄存器中选择的下一个通道转换，直到序列中的所有转换均完成为止。通过 ADC_JSQR 寄存器中的 JL[1:0] 位定义总序列长度。

示例：

- JDISCEN=1，要转换的通道 = 1、2、3
  - 第一次触发：转换通道 1（生成 JEOC 事件）
  - 第二次触发：转换通道 2（生成 JEOC 事件）
  - 第三次触发：转换通道 3 并生成 JEOC 事件和 JEOS 事件
  - ...

注： *转换完所有注入通道后，下一个触发信号将启动第一个注入通道的转换。在上述示例中，第 4 次触发重新转换了第 1 个注入通道 1。*

不能同时使用自动注入模式和不连续模式：当 JAUTO 置 1 时，DISCEN 和 JDISCEN 位必须通过软件保持清零状态。

16.3.21 注入转换的上下文队列

实现上下文队列可为下一个注入转换序列准备多达 2 个上下文。必须将 ADC_CFGR 寄存器的 JQDIS 位复位才能使能此功能。上下文队列使能时，只能进行硬件触发转换。

该上下文包括：

- 注入触发的配置（ADC_JSQR 寄存器中的 JEXTEN[1:0] 位和 JEXTSEL 位）
- 注入序列的定义（ADC_JSQR 寄存器中的 JSQx[4:0] 位和 JL[1:0] 位）

上下文的所有参数都会在 ADC_JSQR 这一寄存器中定义，该寄存器会实现一个双缓冲区队列，可缓冲多达 2 组参数：

- JSQR 寄存器可随时写入，正在进行注入转换时也不例外。
- 每个写入到 JSQR 寄存器中的数据均会存储在上下文队列中。
- 队列开始时为空，对 JSQR 寄存器进行的第一次写访问时会立即更改上下文，随即 ADC 会准备好接收注入触发。
- 注入序列完成后，队列会被占用，上下文会根据队列中存储的后续 JSQR 参数进行更改。这一新的上下文会用于下一个注入转换序列。
- 如果在队列已满的情况下向 JSQR 寄存器执行写操作，会发生队列溢出。这种溢出情况会通过 JQOVF 标志置为有效来指示。发生溢出时，会忽略造成溢出的 JSQR 寄存器写访问，并且上下文队列保持不变。如果 JQOVFIE 位置 1，可产生中断。

- 队列变空时可能执行两种操作，具体取决于寄存器 ADC_CFGR 的控制位 JQM 的值：
  - 如果 JQM=0，队列刚好在使能 ADC 后为空，但在随后的运行操作期间绝不会变空：队列始终保留上一个有效的上下文，并会根据上一个有效的上下文处理后续的有效注入序列启动。
  - 如果 JQM=1，队列会在注入序列结束后或队列被清空时变空。这种情况下，队列中不存在任何上下文，硬件触发也会被禁止。因此会忽略后续的所有硬件注入触发，直至软件重新向 JSQR 寄存器写入新的注入上下文。
- 读取 JSQR 寄存器会返回当前有效的 JSQR 上下文。如果 JSQR 上下文为空，JSQR 的读出值为 0x0000。
- 如果通过将 JADSTP 置 1 的方式停止注入转换、或者通过将 ADDIS 置 1 的方式禁止 ADC，队列会被清空。
  - 如果 JQM=0，队列会保留上一个有效的上下文。
  - 如果 JQM=1，队列会变空，并会忽略触发。

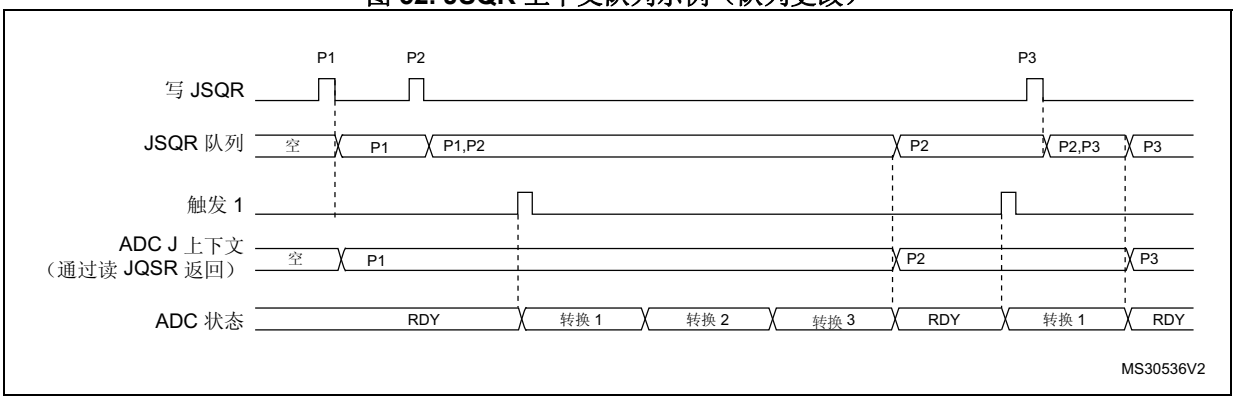
注： *如果配置为不连续模式（位 JDISCEN=1），只有注入序列的最后一次触发会更改上下文并占用队列。第一次触发仅会占用队列，但其他触发仍为有效触发，如下文中的不连续模式示例所示（两个上下文的长度均为 3）：*

- *第一次触发，不连续。序列 1：上下文 1 已占用，已执行第一次转换。*
- *第二次触发，不连续。序列 1：第二次转换。*
- *第三次触发，不连续。序列 1：第三次转换。*
- *第四次触发，不连续。序列 2：上下文 2 已占用，已执行第一次转换。*
- *第五次触发，不连续。序列 2：第二次转换。*
- *第六次触发，不连续。序列 2：第三次转换。*

更改触发或序列上下文时的操作

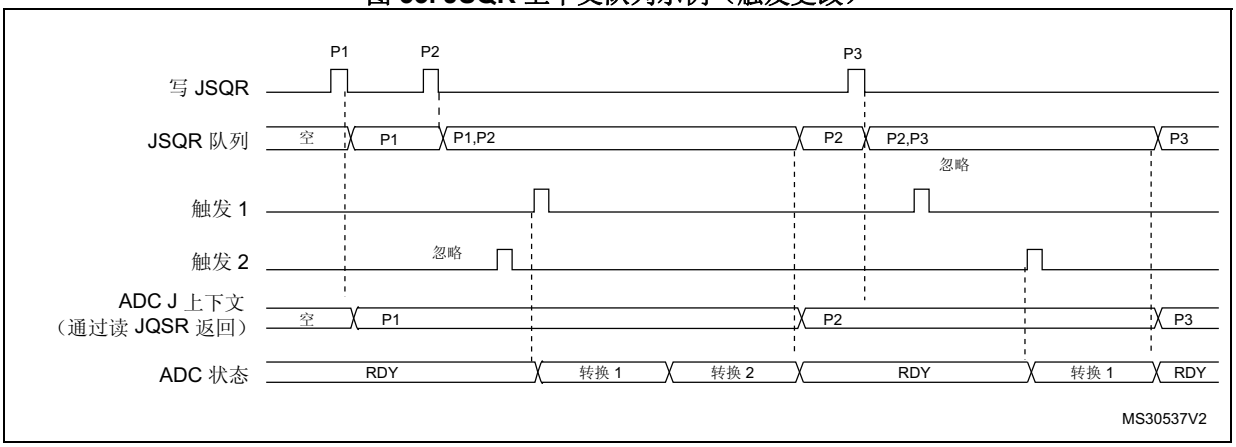
图 52 和*图 53* 显示了更改序列或触发时上下文队列的操作。

图 52. JSQR 上下文队列示例（队列更改）



1. 参数：
   P1：3 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：4 个转换组成的序列，硬件触发 1

图 53. JSQR 上下文队列示例（触发更改）

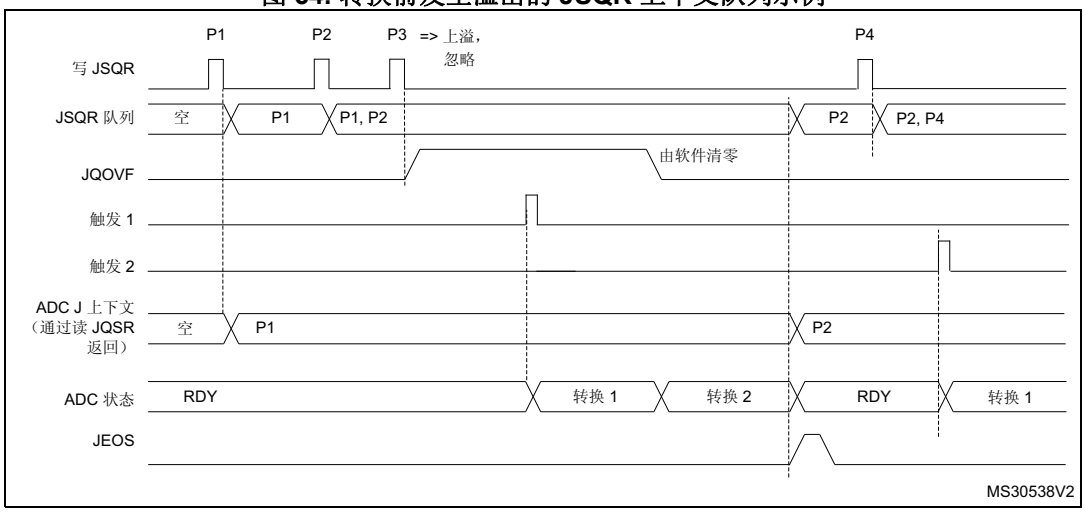


1. 参数：
   P1：2 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 2
   P3：4 个转换组成的序列，硬件触发 1

上下文队列：发生队列溢出时的操作

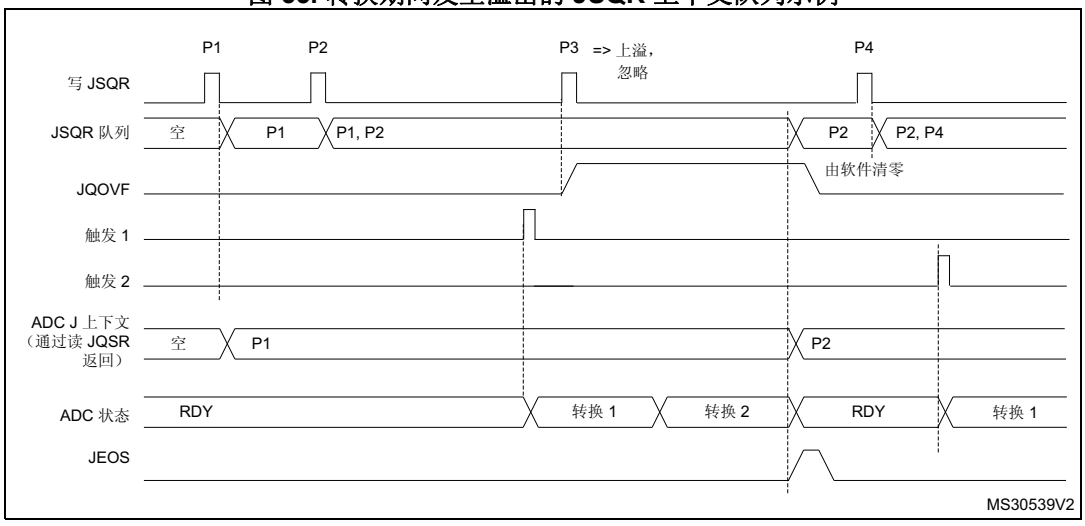
图 54 和*图 55* 显示了转换之前或之后发生溢出时上下文队列的操作。

图 54. 转换前发生溢出的 JSQR 上下文队列示例



1. 参数：
   P1：2 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 2
   P3：3 个转换组成的序列，硬件触发 1
   P4：4 个转换组成的序列，硬件触发 1

图 55. 转换期间发生溢出的 JSQR 上下文队列示例



1. 参数：
   P1：2 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 2
   P3：3 个转换组成的序列，硬件触发 1
   P4：4 个转换组成的序列，硬件触发 1

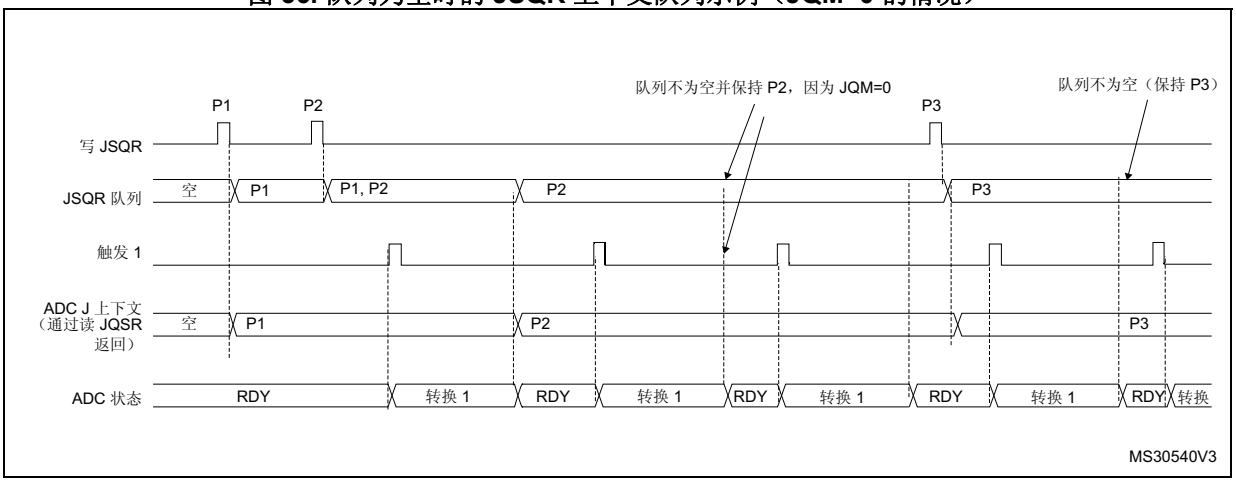
建议按照下述方法管理队列溢出：

- 将每个 P 上下文写入 JSQR 寄存器后，通过标志 JQOVF 指示写操作是否已被忽略（可生成中断）。
- 为避免队列溢出，仅在上一个上下文 P2 的 JEOS 标志置 1 后写入第三个上下文 (P3)。这样可确保上一个上下文已被占用且队列未满。

上下文队列：队列变空时的操作

图 56 和*图 57* 显示了在 JQM=0 或 JQM=1 这两种情况下队列变空时上下文队列的操作。

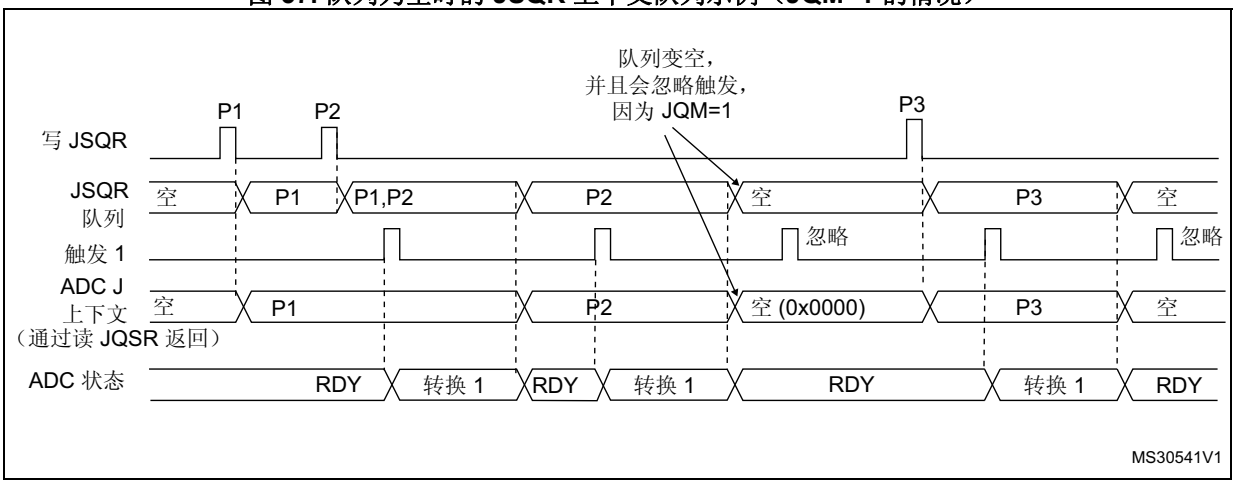
图 56. 队列为空时的 JSQR 上下文队列示例（JQM=0 的情况）



1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

注：　写入 P3 时，上下文会立即改变。但由于内部重新同步的原因，会存在一定的延迟，如果刚好在写入 P3 之后或之前发生触发，可能会发生启动的转换被视为上下文 P2 的情况。为了避免出现这种情况，用户必须确保写入立即应用的新上下文时没有发生 ADC 触发。

图 57. 队列为空时的 JSQR 上下文队列示例（JQM=1 的情况）

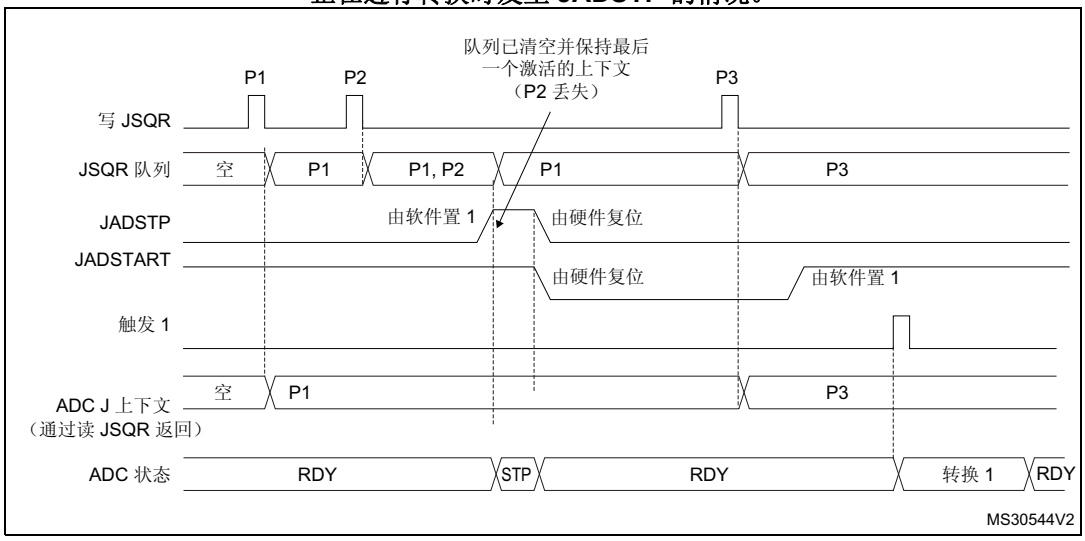


1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

清空上下文队列

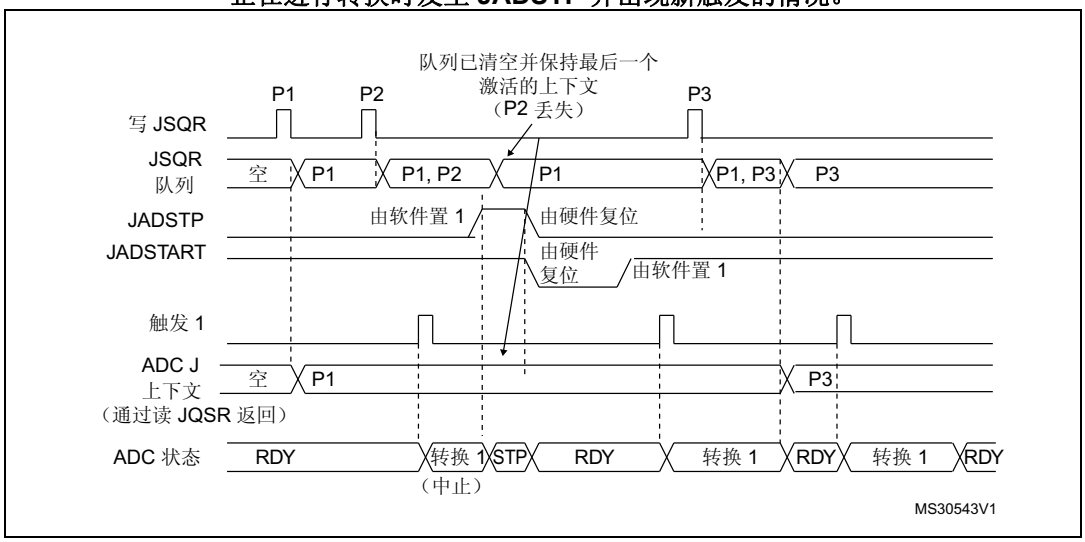
下图显示的是各种情况下清空队列时上下文队列的操作。

图 58. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。正在进行转换时发生 JADSTP 的情况。



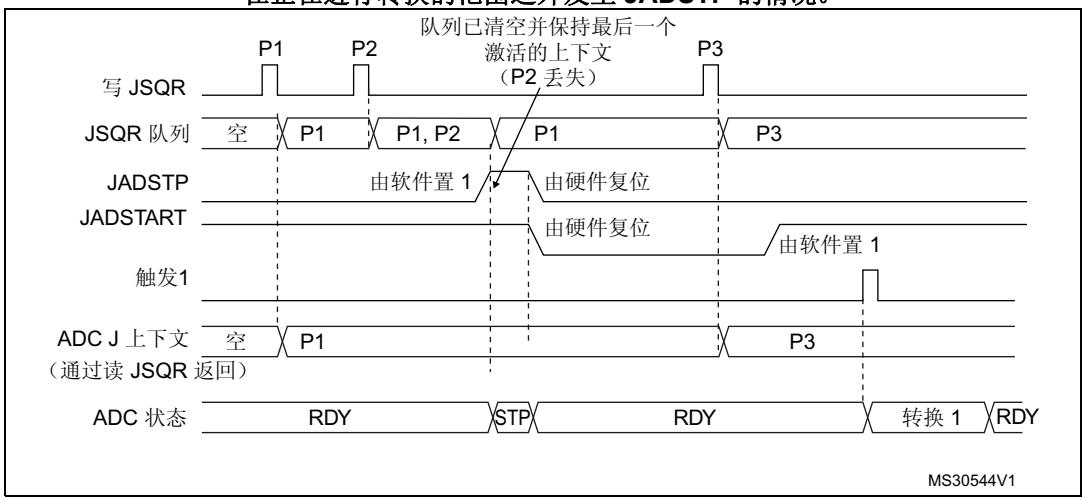
1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 59. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。正在进行转换时发生 JADSTP 并出现新触发的情况。



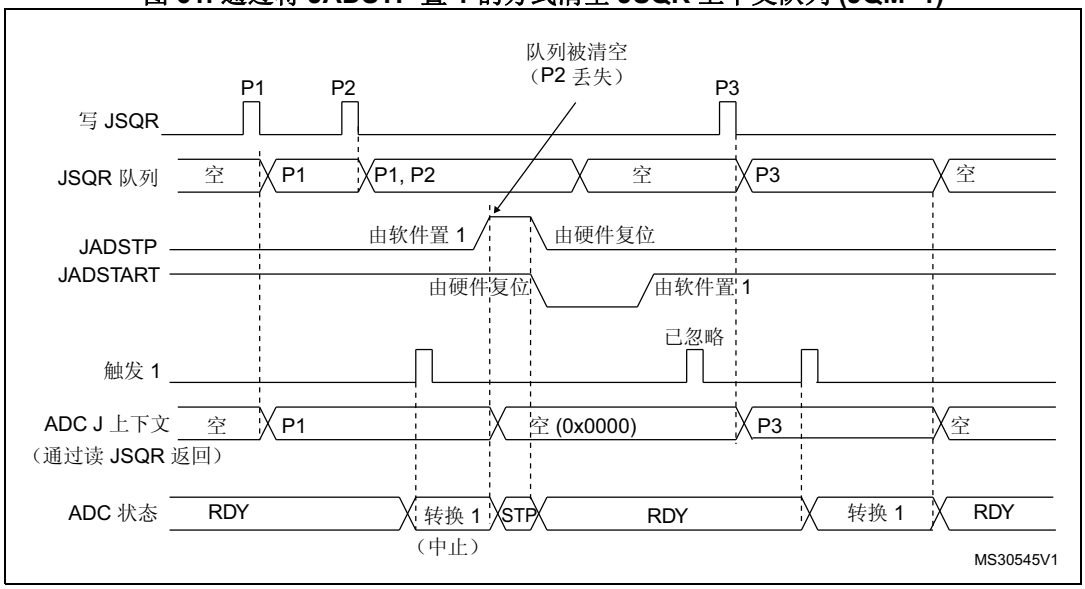
1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 60. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。在正在进行转换的范围之外发生 JADSTP 的情况。



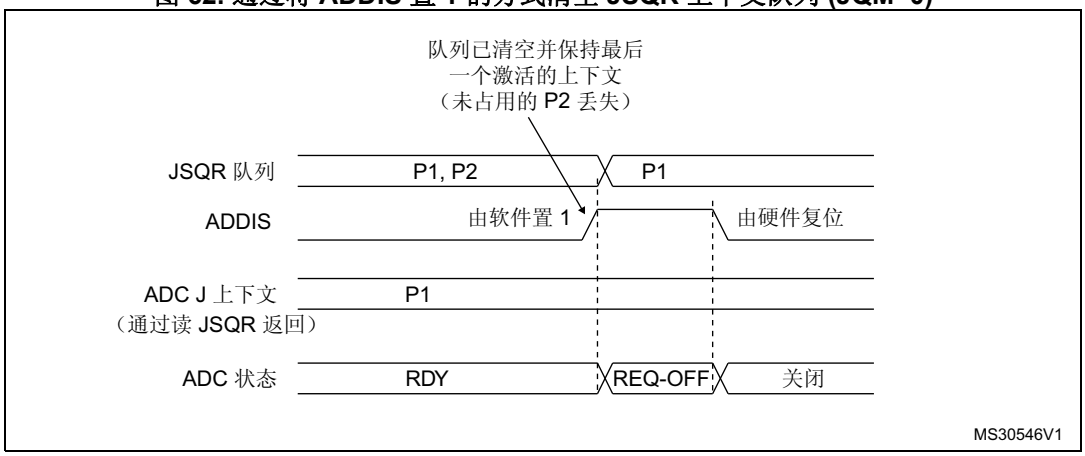
1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 61. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=1)



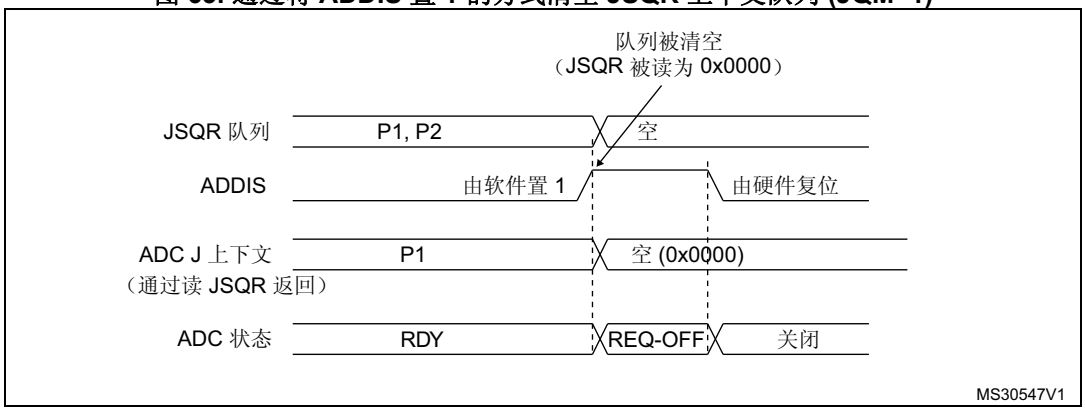
1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 62. 通过将 ADDIS 置 1 的方式清空 JSQR 上下文队列 (JQM=0)



1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 63. 通过将 ADDIS 置 1 的方式清空 JSQR 上下文队列 (JQM=1)



1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

上下文队列：在队列为空时启动 ADC

要在队列为空时启动 ADC 操作，必须按照以下程序进行操作，以免在 ADC 初始化时无法获悉第一个上下文。此程序仅在 JQM 位复位的情况下适用：

5. 写入空 JSQR，JEXTEN 不等于 0（否则会触发软件转换）
6. 将 JADSTART 置 1
7. 将 JADSTP 置 1
8. 等待，直至 JADSTART 复位
9. 将 JADSTART 置 1

禁止队列

可通过将 ADC_CFGR 寄存器中的 JQDIS 位置 1 的方式禁止队列。

16.3.22 可编程分辨率 (RES)——快速转换模式

可通过降低 ADC 分辨率来执行快速转换。

通过对控制位 RES[1:0] 进行编程，可将分辨率配置为 12 位、10 位、8 位或 6 位。*图 68*、*图 69*、*图 70* 和*图 71* 显示了转换结果格式与分辨率和数据对齐方式的对应关系。

对于不要求使用高数据精度的应用，分辨率越低，转换时间越短，降低分辨率可缩短逐次逼近步骤所需的转换时间，如*表 75* 所示。

表 75. T_{SAR} 与分辨率的对应关系

RES（位）	T_{SAR}（ADC 时钟周期）	F_{ADC} = 64 MHz 时的 T_{SAR} (ns)	T_{CONV}（ADC 时钟周期）（采样时间 = 2.5 个 ADC 时钟周期）	F_{ADC} = 64 MHz 时的 T_{CONV} (ns)
12	12.5 个 ADC 时钟周期	195.31 ns	15 个 ADC 时钟周期	234.375 ns
10	10.5 个 ADC 时钟周期	164.06 ns	13 个 ADC 时钟周期	203.125 ns
8	8.5 个 ADC 时钟周期	132.81 ns	11 个 ADC 时钟周期	171.875 ns
6	6.5 个 ADC 时钟周期	101.56 ns	9 个 ADC 时钟周期	140.625 ns

16.3.23 转换结束、采样阶段结束（EOC、JEOC、EOSMP）

每次出现常规转换结束 (EOC) 事件和注入转换结束 (JEOC) 事件时，ADC 都会通知应用。

新的常规转换数据出现在 ADC_DR 寄存器中后，ADC 会立即将 EOC 标志置 1。如果 EOCIE 位置 1，可产生中断。EOC 标志可通过由软件向其写入 1 或读取 ADC_DR 的方式来清零。

新的注入转换数据出现在 ADC_JDRy 寄存器中后，ADC 会立即将 JEOC 标志置 1。如果 JEOCIE 位置 1，可产生中断。JEOC 标志可通过由软件向其写入 1 或读取相应 ADC_JDRy 寄存器的方式来清零。

ADC 还通过将状态位 EOSMP 置 1 来指示采样阶段结束（仅限常规转换）。EOSMP 标志可通过由软件向其写入 1 的方式来清零。如果 EOSMPIE 位置 1，可产生中断。

16.3.24 转换序列结束（EOS、JEOS）

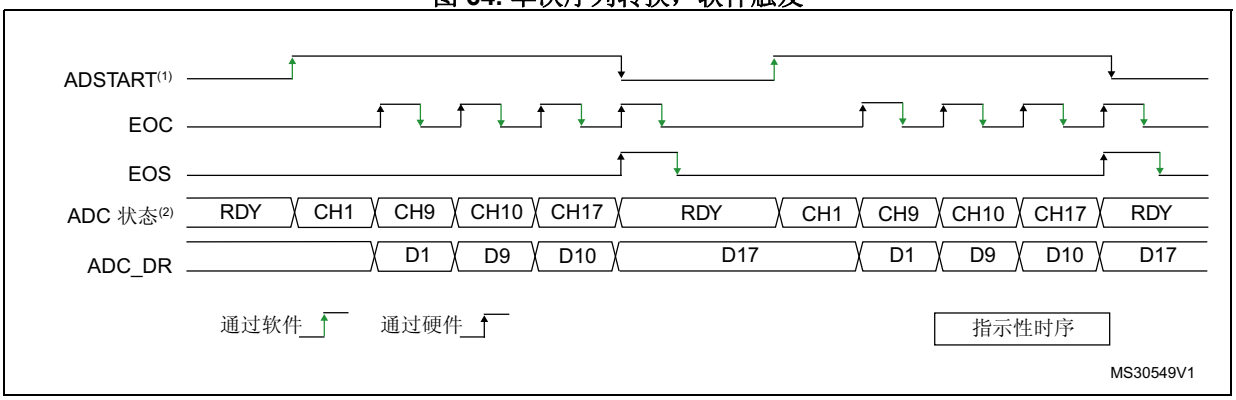
每次出现常规序列结束 (EOS) 事件和注入序列结束 (JEOS) 事件时，ADC 都会通知应用。

常规转换序列的最后一个数据出现在 ADC_DR 寄存器中时，ADC 会立即将 EOS 标志置 1。如果 EOSIE 位置 1，可产生中断。EOS 标志可通过由软件向其写入 1 的方式来清零。

注入转换序列的最后一个数据完成时，ADC 会立即将 JEOS 标志置 1。如果 JEOSIE 位置 1，可产生中断。JEOS 标志可通过由软件向其写入 1 的方式来清零。

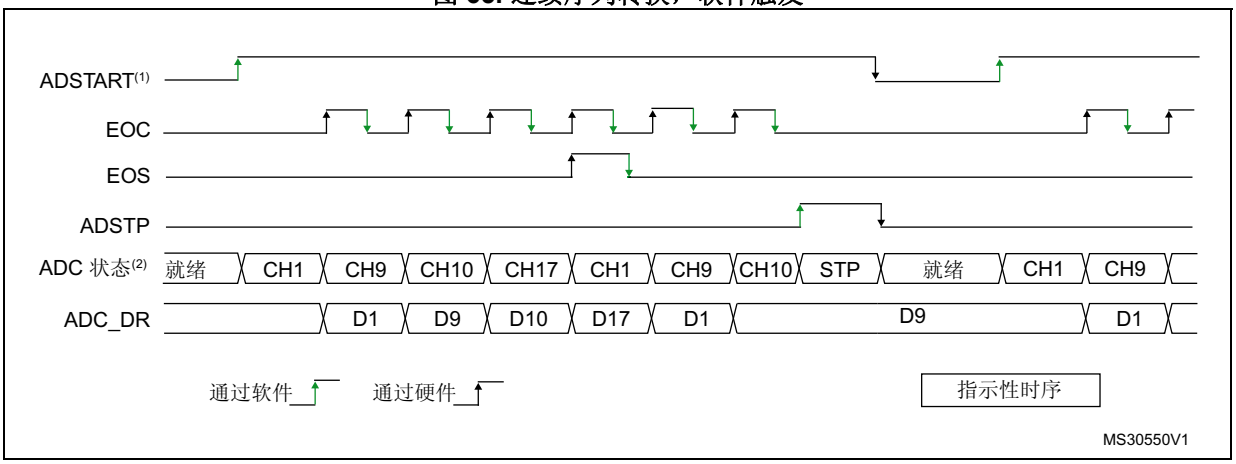
16.3.25 时序图示例（单次模式/连续模式，硬件/软件触发）

图 64. 单次序列转换，软件触发



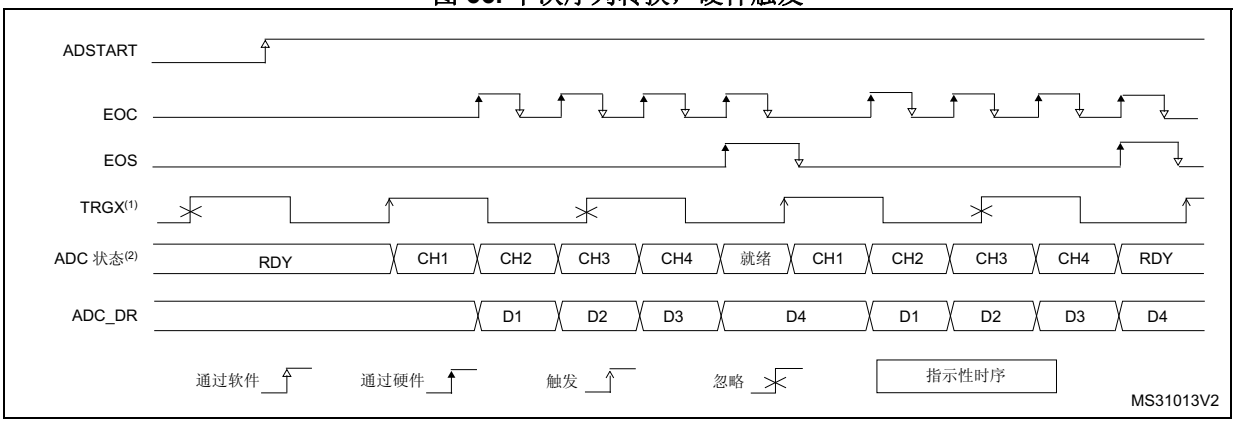
1. EXTEN=0x0，CONT=0
2. 所选通道 = 1、9、10、17；AUTDLY=0

图 65. 连续序列转换，软件触发



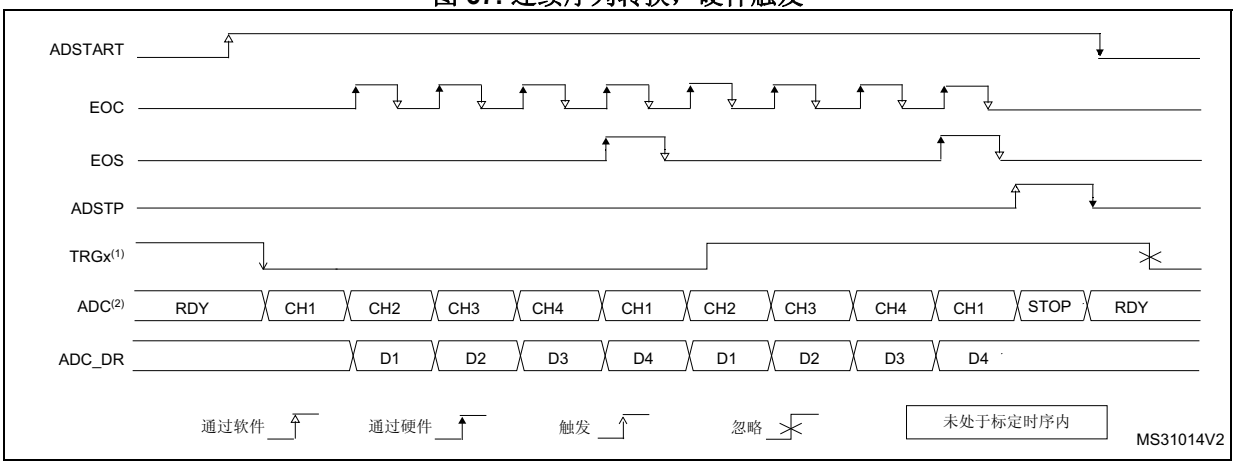
1. EXTEN=0x0，CONT=1
2. 所选通道 = 1、9、10、17；AUTDLY=0

图 66. 单次序列转换，硬件触发



1. 选择 TRGx（过频）作为触发源，EXTEN = 01，CONT = 0
2. 所选通道 = 1、2、3、4；AUTDLY=0

图 67. 连续序列转换，硬件触发



1. 选择 TRGx 作为触发源，EXTEN = 10，CONT = 1
2. 所选通道 = 1、2、3、4；AUTDLY=0

16.3.26 数据管理

数据寄存器、数据对齐和偏移（ADC_DR、OFFSETy、OFFSETy_CH、ALIGN）

数据和对齐

每次常规转换通道结束时（发生 EOC 事件时），转换后数据的结果都会存储在宽度为 16 位的 ADC_DR 数据寄存器中。

每次注入转换通道结束时（发生 JEOC 事件时），转换后数据的结果都会存储在宽度为 16 位的相应 ADC_JDRy 数据寄存器中。

ADC_CFGR 寄存器中的 ALIGN 位用于选择转换后存储的数据的对齐方式。可选择左对齐和右对齐两种方式，如*图 68*、*图 69*、*图 70*、和*图 71* 所示。

特例：采用左对齐时，数据基于半字对齐，分辨率设置为 6 位时除外。当分辨率设置为 6 位时，数据基于字节对齐，如*图 70* 和*图 71* 所示。

注： *过采样模式下不支持左对齐。当 ROVSE 和/或 JOVSE 位置 1 时，忽略 ALIGN 位的值，并且 ADC 仅提供右对齐数据。*

偏移

可以通过将 ADC_OFRy 寄存器中的 OFFSETy_EN 置 1，对通道应用偏移 y（y=1、2、3、4）。应用偏移的通道会编程到 ADC_OFRy 寄存器的 OFFSETy_CH[4:0] 位中。这种情况下，转换后的值会减去写入到 OFFSETy[11:0] 位中的用户自定义偏移。结果可能是负值，因此读取的数据为有符号值，SEXT 位代表扩展符号值。

注： *过采样模式下不支持偏移校正。如果 ROVSE 和/或 JOVSE 位置 1，ADC_OFRy 寄存器中 OFFSETy_EN 位的值会被忽略（视为复位）。*

表 78 介绍了如何对模拟看门狗 1 的所有可能的分辨率进行比较。

表 76. 偏移计算与数据分辨率

分辨率（位 RES[1:0]）	原始转换数据与偏移相减		结果	注释
	原始转换数据，左对齐	偏移		
00：12 位	DATA[11:0]	OFFSET[11:0]	有符号 12 位数据	-
01：10 位	DATA[11:2]，00	OFFSET[11:0]	有符号 10 位数据	用户必须将 OFFSET[1:0] 配置为“00”
10：8 位	DATA[11:4]，0000	OFFSET[11:0]	有符号 8 位数据	用户必须将 OFFSET[3:0] 配置为“0000”
11：6 位	DATA[11:6]，000000	OFFSET[11:0]	有符号 6 位数据	用户必须将 OFFSET[5:0] 配置为“000000”

从对应于通道“i”的 ADC_DR（常规通道）或 ADC_JDRy（注入通道，y=1、2、3、4）读取数据时：

- 如果相应通道的其中一个偏移已使能（位 OFFSETy_EN=1），则读取的数据为有符号数据。
- 如果此通道的四个偏移均未使能，则读取的数据为无符号数据。

图 68、*图 69*、*图 70* 和*图 71* 中显示了有符号数据和无符号数据的对齐方式。

图 68. 右对齐（偏移禁止，无符号值）

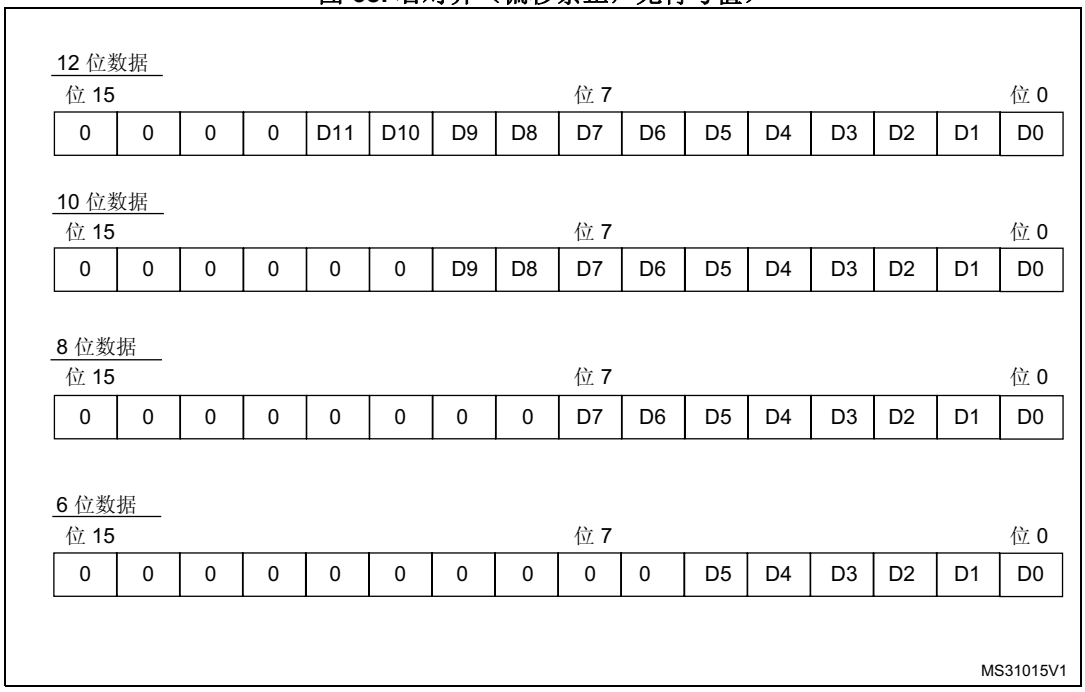


图 69. 右对齐（偏移使能，有符号值）

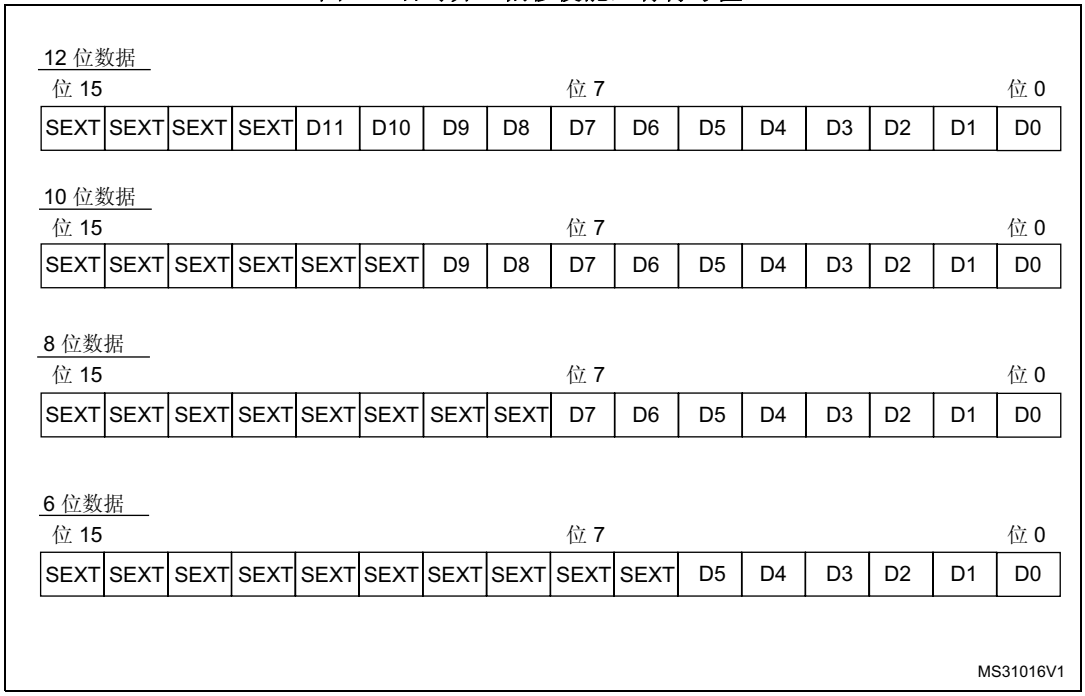


图 70. 左对齐（偏移禁止，无符号值）

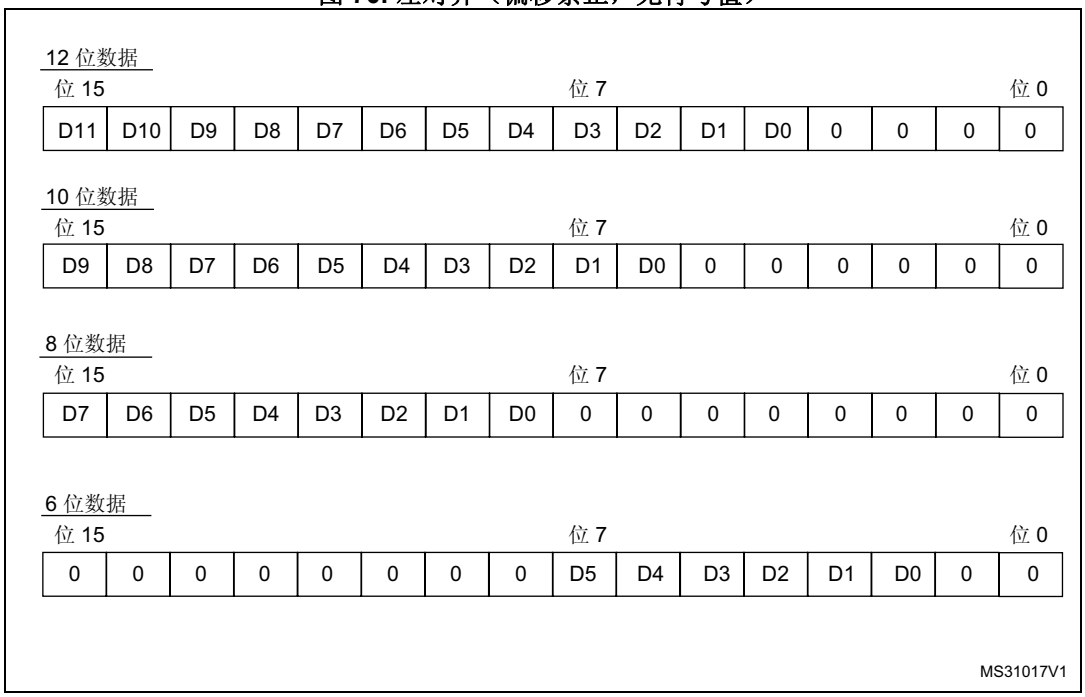
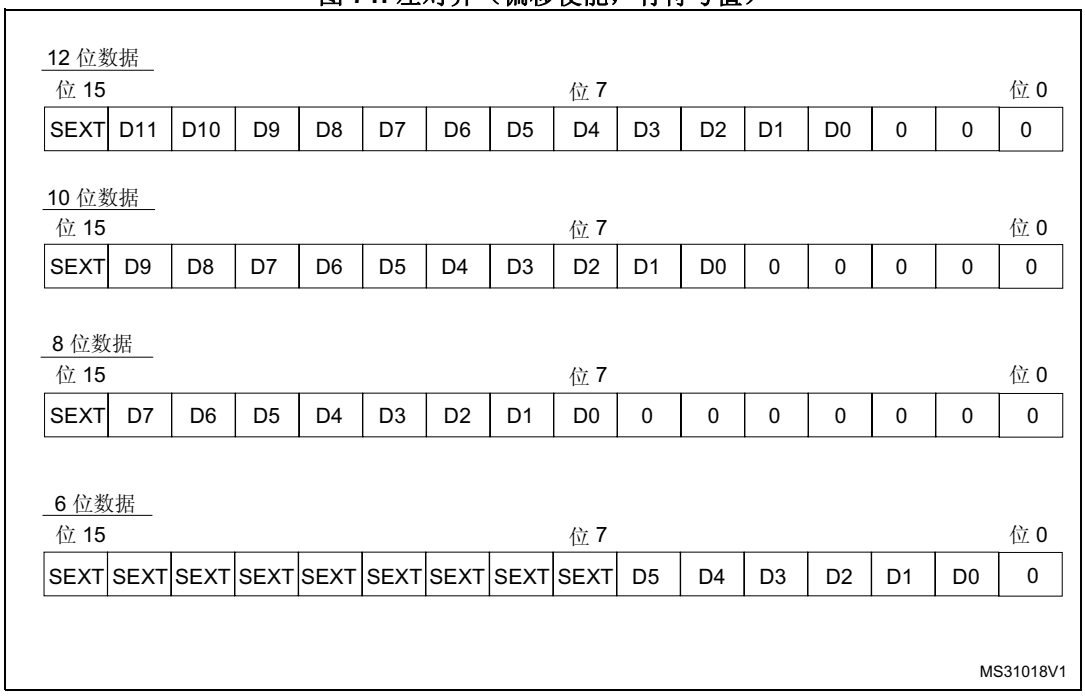


图 71. 左对齐（偏移使能，有符号值）



ADC 溢出（OVR、OVRMOD）

如果常规转换后的数据未在新转换数据可用之前（由 CPU 或 DMA）读取，会由溢出标志 (OVR) 指示缓冲区溢出事件。

如果新转换完成时 EOC 标志仍为 1，则 OVR 标志会置 1。如果 OVRIE 位置 1，可产生中断。

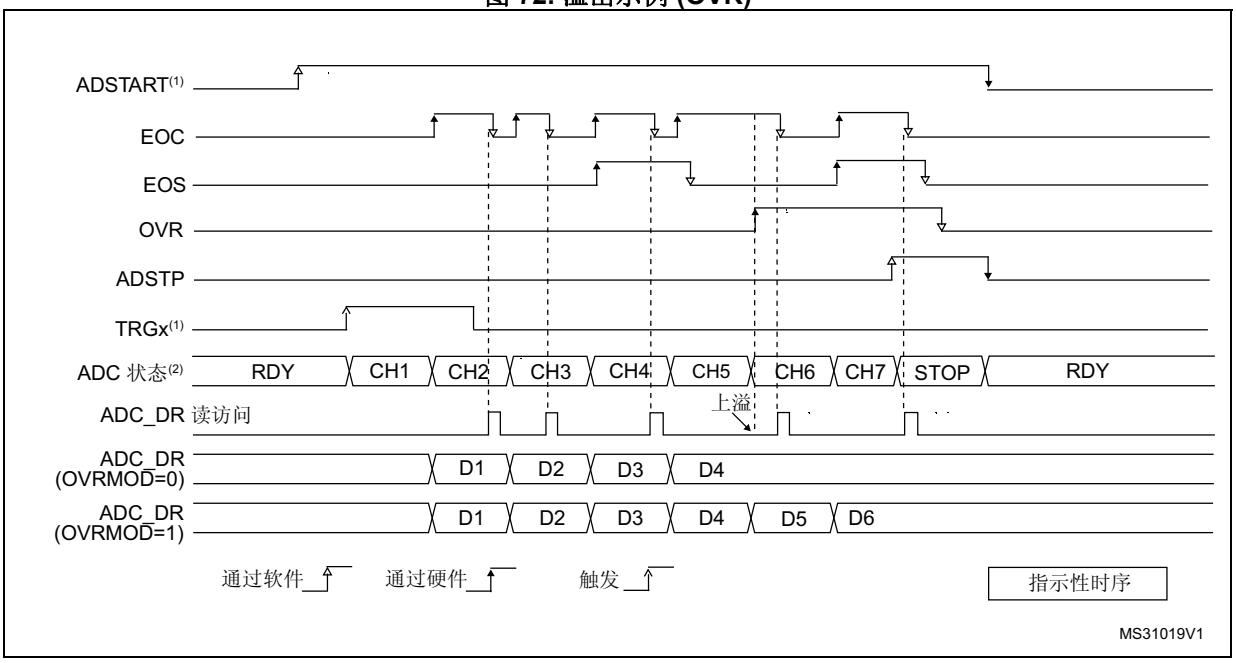
如果发生溢出情况，ADC 仍会保持工作状态并可继续进行转换，除非通过软件将 ADSTP 位置 1，从而停止并复位序列。

OVR 标志可通过由软件向其写入 1 的方式来清零。

可对控制位 OVRMOD 进行编程，从而配置发生溢出事件时是要保留数据还是要覆盖数据：

- OVRMOD=0：溢出事件会保留数据寄存器的数据，防止其被覆盖：会保留原数据，并会丢弃新的转换结果。如果 OVR 保持为 1，可继续进行转换，但还是会丢弃结果数据。
- OVRMOD=1：数据寄存器会由上一次转换结果覆盖，之前未读取的数据会丢失。如果 OVR 保持为 1，可继续正常进行转换，并且 ADC_DR 寄存器将始终包含最新的转换数据。

图 72. 溢出示例 (OVR)



注： *由于四条注入通道均有专用的数据寄存器，因此不会对注入通道进行溢出检测。*

在不使用 DMA 的情况下管理转换序列

如果转换过程足够慢，则可使用软件来处理转换序列。在这种情况下，软件必须使用 EOC 标志及其相关中断来处理各个数据。每当转换结束时，EOC 都会置 1，并且可以读取 ADC_DR 寄存器。OVRMOD 应配置为 0，以便将溢出事件作为错误进行管理。

在不使用 DMA 且不发生溢出的情况下管理转换

ADC 在转换一个或多个通道时不是每次都读取数据的情况下，这可能会很有用（例如，存在模拟看门狗时）。在这种情况下，OVRMOD 位必须配置为 1，OVR 标志应被软件忽略。溢出事件不会阻止 ADC 继续进行转换，ADC_DR 寄存器始终包含最新的转换。

使用 DMA 管理转换

由于转换得出的通道值会存储到唯一的数据寄存器中，因此，对于多个通道的转换，使用 DMA 非常有帮助。这样可以避免丢失在下一次写入之前还未被读出的 ADC_DR 寄存器中的数据。

若 DMA 模式已使能（ADC_CFGR 寄存器中的 DMAEN 位置 1），则会在每个通道转换后生成 DMA 请求。这样便可将转换的数据从 ADC_DR 寄存器传输到用软件选择的目标位置。

尽管如此，如果因 DMA 无法及时处理 DMA 传输请求而发生溢出 (OVR=1)，ADC 会停止生成 DMA 请求，新转换对应的数据不会通过 DMA 进行传输。这意味着可将传输到 RAM 的所有数据都视为有效数据。

根据 OVRMOD 位的配置，可保留或覆盖数据（请参见 *ADC 溢出（OVR、OVRMOD）*一节）。

DMA 传输请求会禁止，直至软件将 OVR 位清零。

根据应用用途的不同，推荐使用两种不同的 DMA 模式，并使用 ADC_CFGR 寄存器的 DMACFG 位配置相应的模式：

- DMA 单次模式 (DMACFG=0)
  如果通过编程将 DMA 设置为传输固定数目的数据，应选择此模式。
- DMA 循环模式 (DMACFG=1)
  如果在循环模式下对 DMA 进行编程，应选择此模式。

DMA 单次模式 (DMACFG=0)

在该模式下，每次出现新的转换数据时，ADC 都会生成 DMA 传输请求，DMA 到达最后一个 DMA 传输操作时（发生 DMA_EOT 中断，请参见 DMA 一段），即使转换已再次开始，ADC 也会停止生成 DMA 请求。

DMA 传输完成后（在 DMA 控制器中配置的所有传输操作均已完成后）：

- ADC 数据寄存器的内容会冻结。
- 任何正在进行的转换都会中止，其部分结果会被丢弃。
- 不会将任何新的 DMA 请求发送到 DMA 控制器。如果仍存在已开始的转换，这样可避免生成溢出错误。
- 扫描序列会停止并复位。
- DMA 会停止。

DMA 循环模式 (DMACFG=1)

在该模式下，每次数据寄存器中出现新的转换数据时，ADC 都会生成 DMA 传输请求，即使 DMA 已到达最后一次 DMA 传输操作也不例外。这样可将处于循环模式的 DMA 配置为处理连续的模拟输入数据流。

16.3.27 动态低功耗特性

自动延迟转换模式 (AUTDLY)

ADC 会执行由 AUTDLY 配置位控制的自动延迟转换模式。自动延迟转换可用于简化软件，并可优化采用低频时钟的应用（此类应用可能存在 ADC 溢出的风险）的性能。

AUTDLY=1 时，仅当同一组内所有之前的数据已处理完毕时，才能开始新的转换：

- 对于常规转换：ADC_DR 寄存器已读取或者 EOC 位已清零时（请参见*图 73*）。
- 对于注入转换：JEOS 位已清零时（请参见*图 74*）。

通过这种方式，可自动调整 ADC 的速度，使其适应系统读取数据的速度。

延时插入到每次常规转换后（无论 DISCEN=0 还是 1）和每个注入转换序列后（无论 JDISCEN=0 还是 1）。

注： *不会在注入转换序列之间插入延时，但会在最后一个序列之后插入延时。*

转换过程中，此延时期间发生的硬件触发事件（针对同一组转换）会被忽略。

注： *如果延时期间发生软件触发，则不会被忽略，仍可在该延时时间内通过将 ADSTART 或 JADSTART 位置 1 的方式重新启动转换；启动新转换之前，会由软件读取数据。*

不会在不同组的转换之间插入延时（常规转换后即进行注入转换，反之亦然）：

- 如果在常规转换的自动延时期间发生注入触发，注入转换会立即开始（请参见*图 74*）。
- 注入序列完成后，ADC 会等待上一常规转换的延时（如果未结束），然后才会启动新的常规转换（请参见*图 76*）。

自动注入模式 (JAUTO=1) 下的操作略有不同，仅当上一注入转换序列的自动延时已结束时（JEOS 已清零时），才能开始进行新的常规转换。这是为了确保软件可在启动新序列之前读取给定序列的所有数据（请参见*图 77*）。

要在连续自动注入和自动延时组合模式下停止转换（JAUTO=1、CONT=1 且 AUTDLY=1），请按照以下程序进行操作：

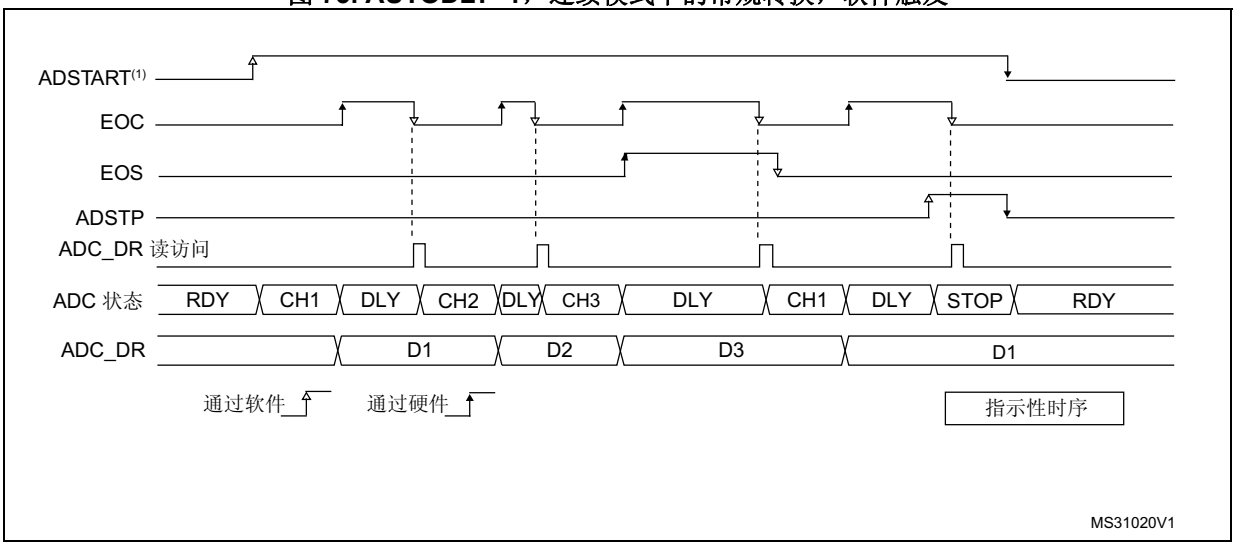
1. 等待，直至 JEOS=1（不会重新启动任何转换）
2. 将 JEOS 清零
3. 将 ADSTP 置 1
4. 读取常规数据

否则，如果 JEOS 在 ADSTP 已置 1 后清零，可能重新启动新的常规序列。

在 AUTDLY 模式下，如果在正在进行的常规序列期间或序列的最后一次常规转换之后的延时期间发生硬件常规触发事件，则会忽略该触发事件。但是，如果触发事件发生在该延时之后，即使发生在随后的注入序列的延时期间，也会将该触发事件视为待处理事件。随后，会在注入序列的延时结束时开始转换。

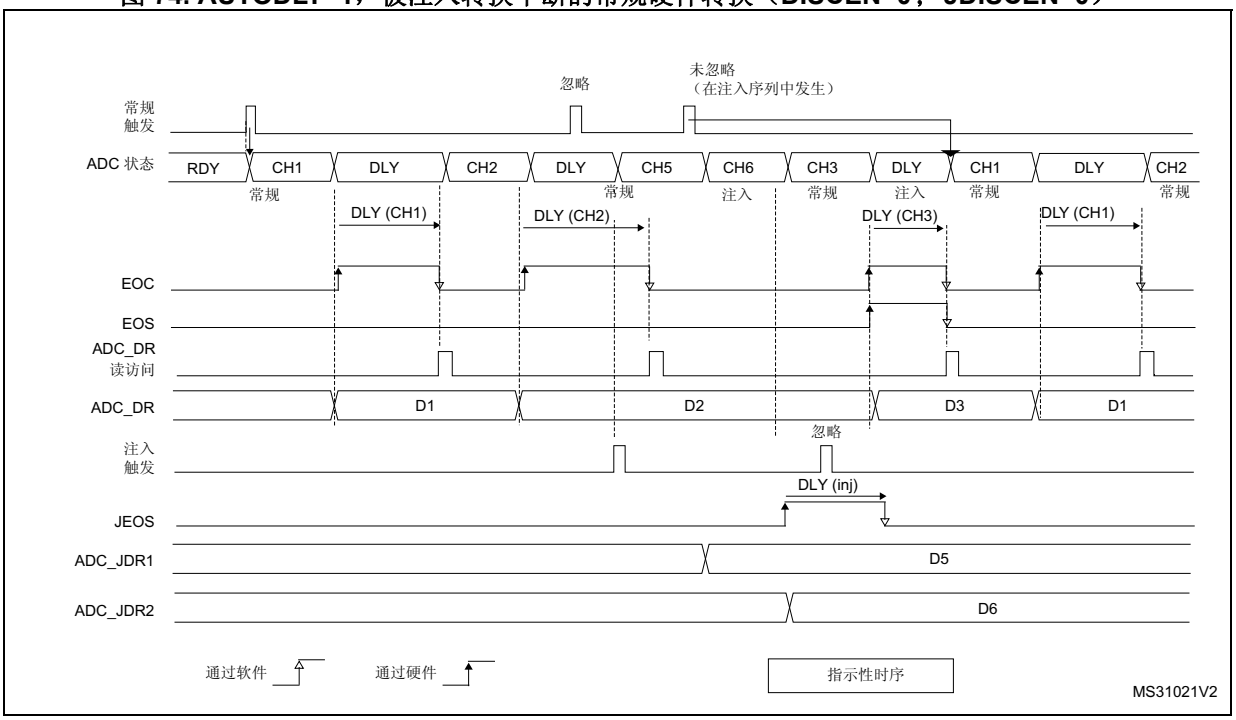
在 AUTDLY 模式下，如果在正在进行的注入序列期间或序列的最后一次注入转换之后的延时期间发生硬件注入触发事件，则会忽略该触发事件。

图 73. AUTODLY=1，连续模式下的常规转换，软件触发



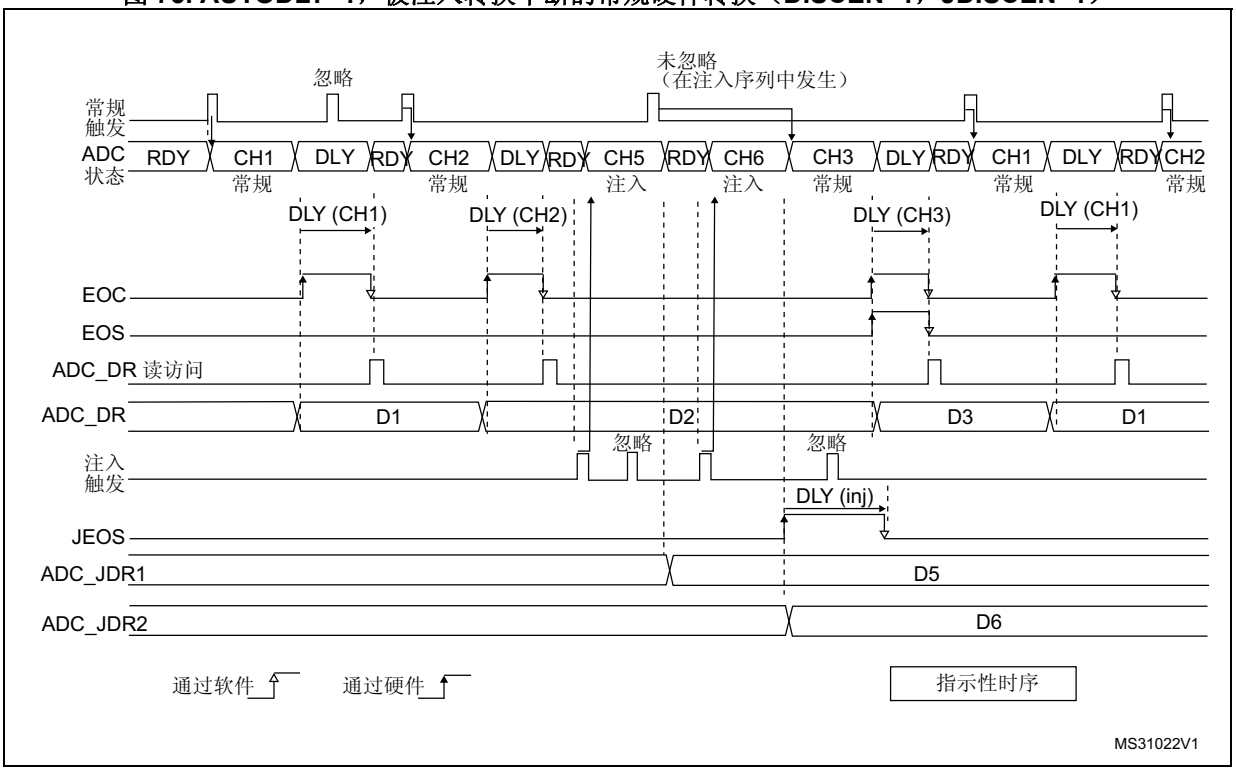
1. AUTDLY=1
2. 常规配置：EXTEN=0x0（软件触发），CONT=1，CHANNELS = 1、2、3
3. 注入配置禁止

图 74. AUTODLY=1，被注入转换中断的常规硬件转换（DISCEN=0；JDISCEN=0）



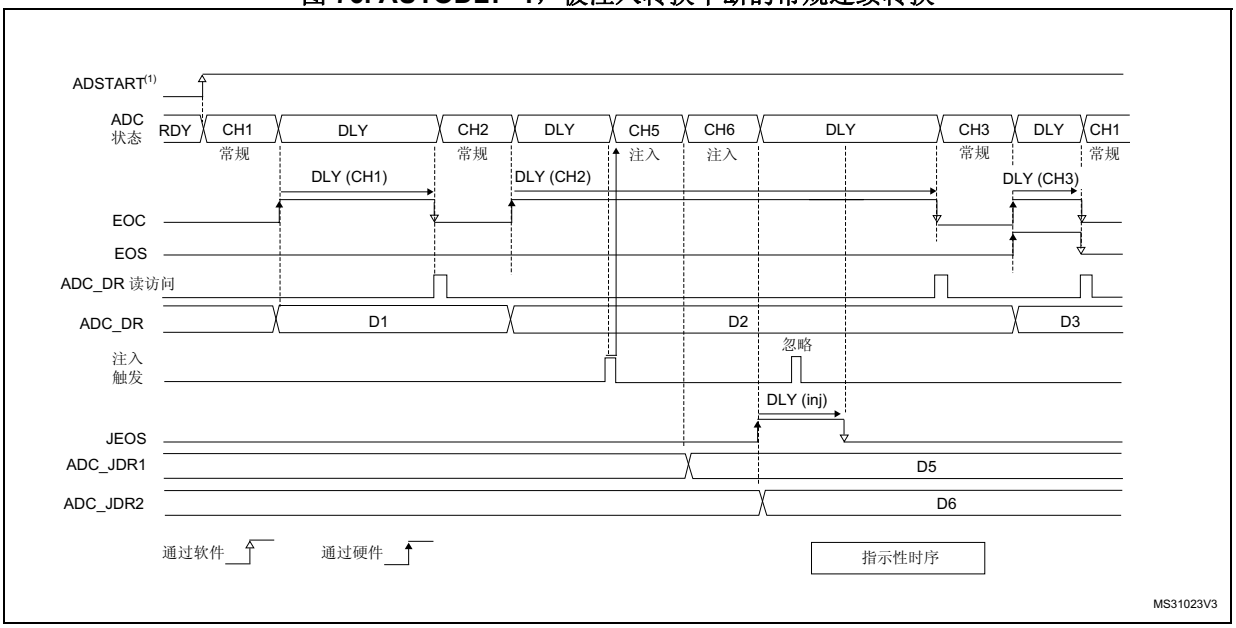
1. AUTDLY=1
2. 常规配置：EXTEN=0x1（硬件触发），CONT=0，DISCEN=0，CHANNELS = 1、2、3
3. 注入配置：JEXTEN=0x1（硬件触发），JDISCEN=0，CHANNELS = 5、6

图 75. AUTODLY=1，被注入转换中断的常规硬件转换（DISCEN=1，JDISCEN=1）



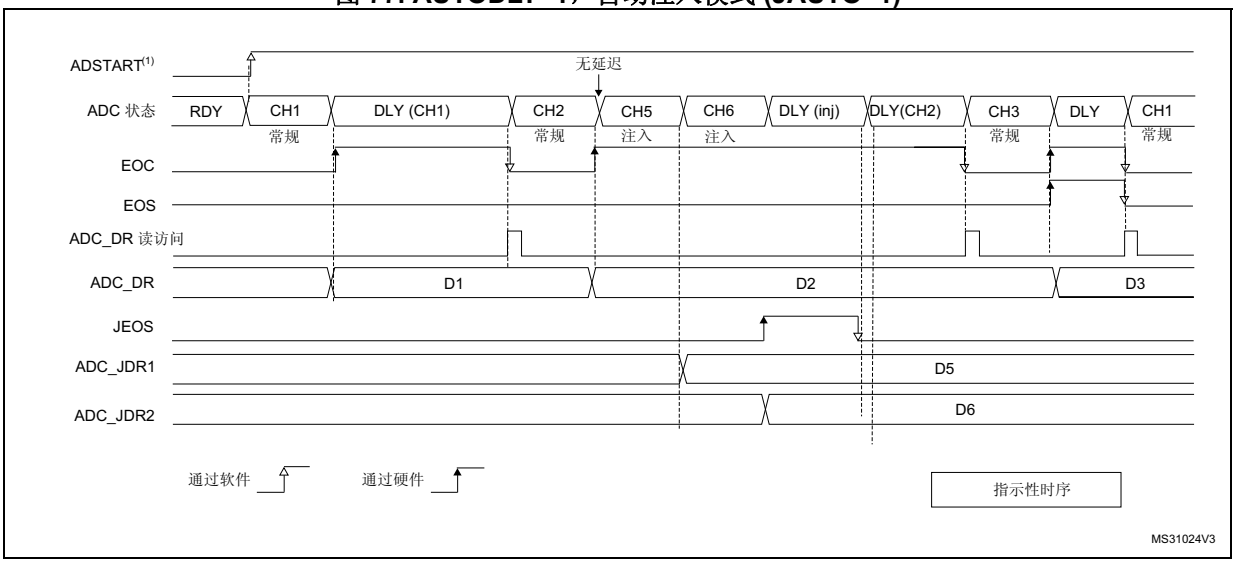
1. AUTDLY=1
2. 常规配置：EXTEN=0x1（硬件触发），CONT=0，DISCEN=1，DISCNUM=1、CHANNELS = 1、2、3
3. 注入配置：JEXTEN=0x1（硬件触发），JDISCEN=1，CHANNELS = 5、6

图 76. AUTODLY=1，被注入转换中断的常规连续转换



1. AUTDLY=1
2. 常规配置：EXTEN=0x0（软件触发），CONT=1，DISCEN=0，CHANNELS = 1、2、3
3. 注入配置：JEXTEN=0x1（硬件触发），JDISCEN=0，CHANNELS = 5、6

图 77. AUTODLY=1，自动注入模式 (JAUTO=1)

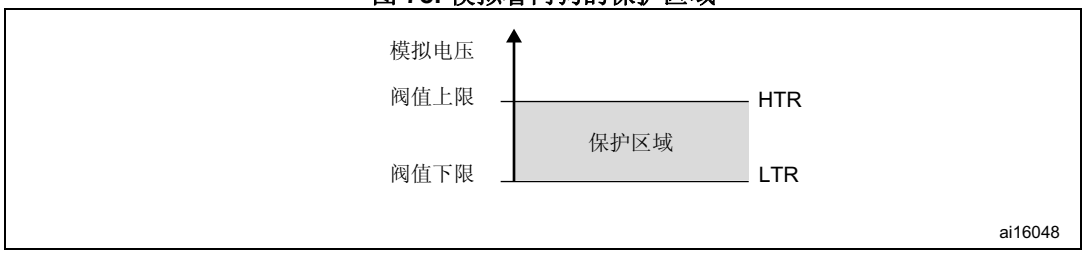


1. AUTDLY=1
2. 常规配置：EXTEN=0x0（软件触发），CONT=1，DISCEN=0，CHANNELS = 1、2
3. 注入配置：JAUTO=1，CHANNELS = 5、6

16.3.28 模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）

三个 AWD 模拟看门狗会监测一些通道是否保持在配置的电压范围（窗口）内。

图 78. 模拟看门狗的保护区域



AWDx 标志和中断

可通过将 ADC_IER 寄存器（x=1、2、3）中的 AWDxIE 置 1 的方式分别为 3 个模拟看门狗使能中断。

AWDx（x=1、2、3）标志可通过由软件向其写入 1 的方式来清零。

在对齐之前，会将 ADC 转换结果与阈值上限和下限进行比较。

模拟看门狗 1 说明

将 ADC_CFGR 寄存器中的 AWD1EN 位置 1，可使能 AWD 模拟看门狗 1。该看门狗监测一条已选通道或所有已使能通道(1) 是否仍在配置的电压范围（窗口）内。

表 77 介绍了应如何配置 ADC_CFGR 寄存器才能在一条或多个通道上使能模拟看门狗。

表 77. 模拟看门狗通道选择

模拟看门狗保护的通道	AWD1SGL 位	AWD1EN 位	JAWD1EN 位
无	x	0	0
所有注入通道	0	0	1
所有常规通道	0	1	0
所有常规通道和注入通道	0	1	1
单个(1)注入通道	1	0	1
单个(1)常规通道	1	1	0
单个(1)常规通道或注入通道	1	1	1

1. 通过 AWD1CH[4:0] 位选择。此外，还必须将通道编程为在合适的常规序列或注入序列中进行转换。

如果 ADC 转换的模拟电压低于阈值下限或高于阈值上限，则 AWD1 模拟看门狗状态位会置 1。

这些阈值编程到模拟看门狗 1 的 ADC_TR1 寄存器的 HT1[11:0] 和 LT1[11:0] 位。如果转换的数据分辨率小于 12 位（取决于 RES[1:0] 位），由于始终会在内部对完整的 12 位原始转换数据进行比较（左对齐），因此编程阈值的 LSB 必须保持清零状态。

表 78 介绍了如何对模拟看门狗 1 的所有可能的分辨率进行比较。

表 78. 模拟看门狗 1 比较

分辨率（RES[1:0] 位）	模拟看门狗比较对象：		注释
	原始转换数据，左对齐	阈值	
00：12 位	DATA[11:0]	LT1[11:0] 和 HT1[11:0]	-
01：10 位	DATA[11:2]，00	LT1[11:0] 和 HT1[11:0]	用户必须将 LT1[1:0] 和 HT1[1:0] 配置为 00
10：8 位	DATA[11:4]，0000	LT1[11:0] 和 HT1[11:0]	用户必须将 LT1[3:0] 和 HT1[3:0] 配置为 0000
11：6 位	DATA[11:6]，000000	LT1[11:0] 和 HT1[11:0]	用户必须将 LT1[5:0] 和 HT1[5:0] 配置为 000000

模拟看门狗 2 和 3 说明

第二个和第三个模拟看门狗更加灵活，可通过编程 AWDCHx[19:0]（x=2、3）中的相应位来保护多条已选通道。

AWDCHx[19:0]（x=2、3）的任何位置 1 时，会使能相应的看门狗。

其分辨率被限制为 8 位，仅高 8 位阈值可编程到 HTx[7:0] 和 LTx[7:0] 中。*表 79* 介绍了如何对所有可能的分辨率进行比较。

表 79. 模拟看门狗 2 和 3 比较

分辨率（RES[1:0] 位）	模拟看门狗比较对象：		注释
	原始转换数据，左对齐	阈值	
00：12 位	DATA[11:4]	LTx[7:0] 和 HTx[7:0]	DATA[3:0] 与比较无关
01：10 位	DATA[11:4]	LTx[7:0] 和 HTx[7:0]	DATA[3:2] 与比较无关
10：8 位	DATA[11:4]	LTx[7:0] 和 HTx[7:0]	-
11：6 位	DATA[11:6]，00	LTx[7:0] 和 HTx[7:0]	用户必须将 LTx[1:0] 和 HTx[1:0] 配置为 00

ADCy_AWDx_OUT 信号输出生成

每个模拟看门狗都关联到一个内部硬件信号 ADCy_AWDx_OUT（y = ADC 编号，x = 看门狗编号），该信号直接连接到一些片上定时器的 ETR 输入（外部触发）。请参见片上定时器部分以了解如何选择 ADCy_AWDx_OUT 信号作为 ETR。

当关联的模拟看门狗使能时，ADCy_AWDx_OUT 会激活：

- 当受保护的转换超出编程阈值时，ADCy_AWDx_OUT 会置 1。
- 在编程阈值范围内的下一受保护转换结束后，ADCy_AWDx_OUT 会复位（如果下一受保护转换仍超出编程阈值范围，此位仍保持置 1）。
- 禁止 ADC 时（将 ADDIS 置 1 时），ADCy_AWDx_OUT 也保持复位状态。请注意，停止常规转换或注入转换（将 ADSTP 或 JADSTP 置 1）对 ADCy_AWDx_OUT 的生成没有任何影响。

注： *AWDx 标志由硬件置 1，并由软件复位：AWDx 标志对 ADCy_AWDx_OUT 的生成没有影响（例如：如果软件未将 AWDx 标志清零，即 AWDx 标志会保持为 1，此时 ADCy_AWDx_OUT 仍可翻转）。*

图 79. ADCy_AWDx_OUT 信号生成（所有常规通道上）

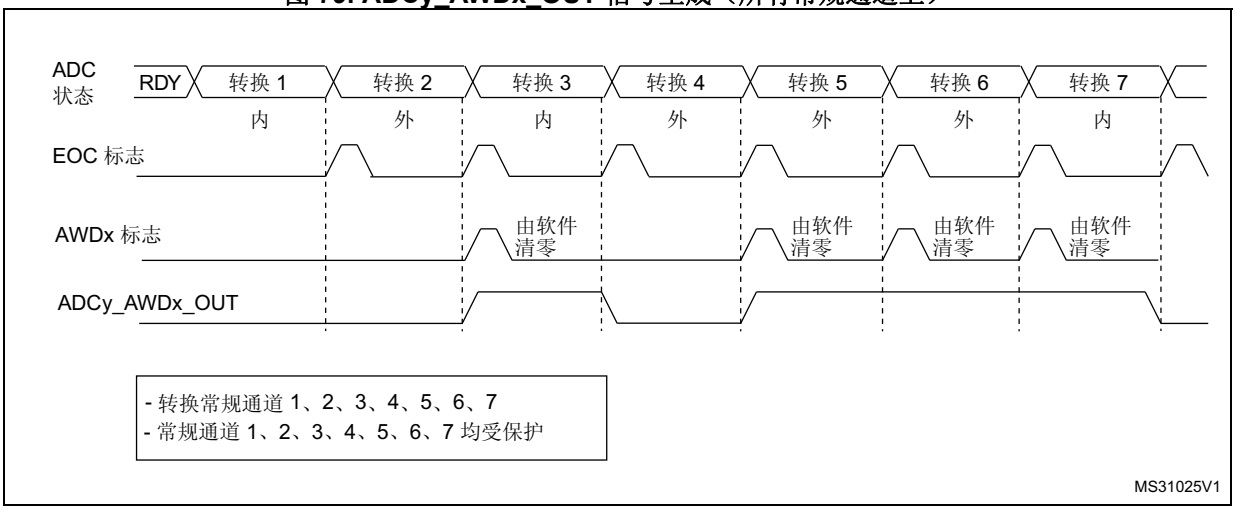


图 80. ADCy_AWDx_OUT 信号生成（AWDx 标志未通过软件清零）

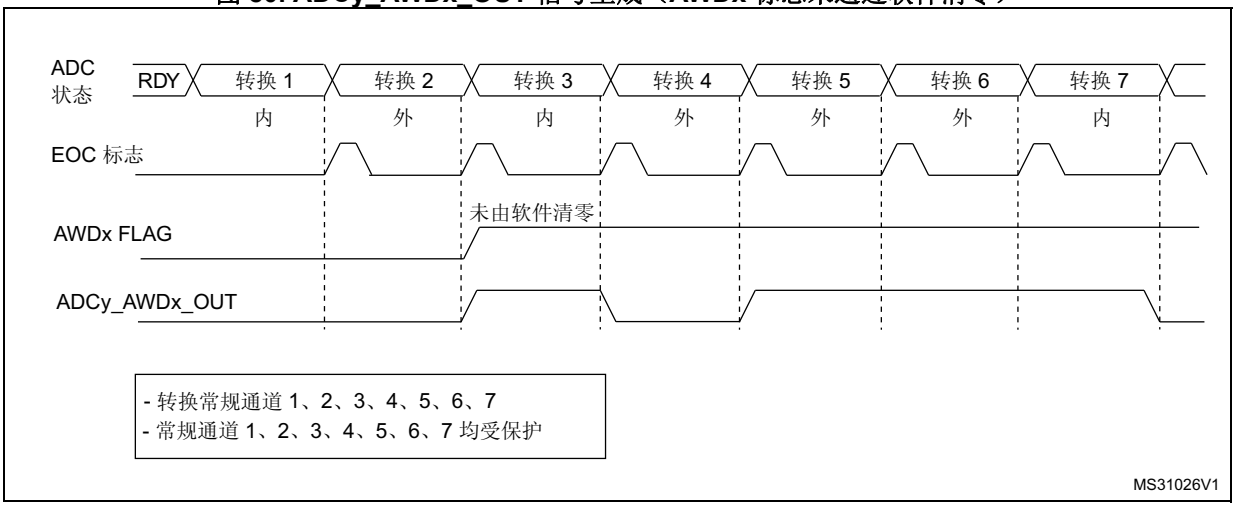


图 81. ADCy_AWDx_OUT 信号生成（单条常规通道上）

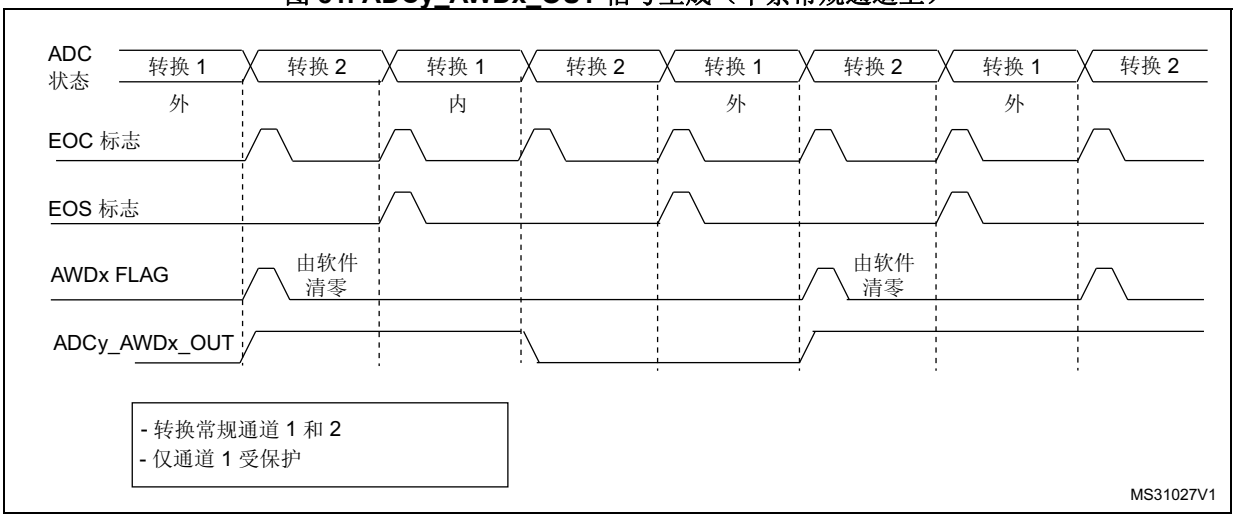
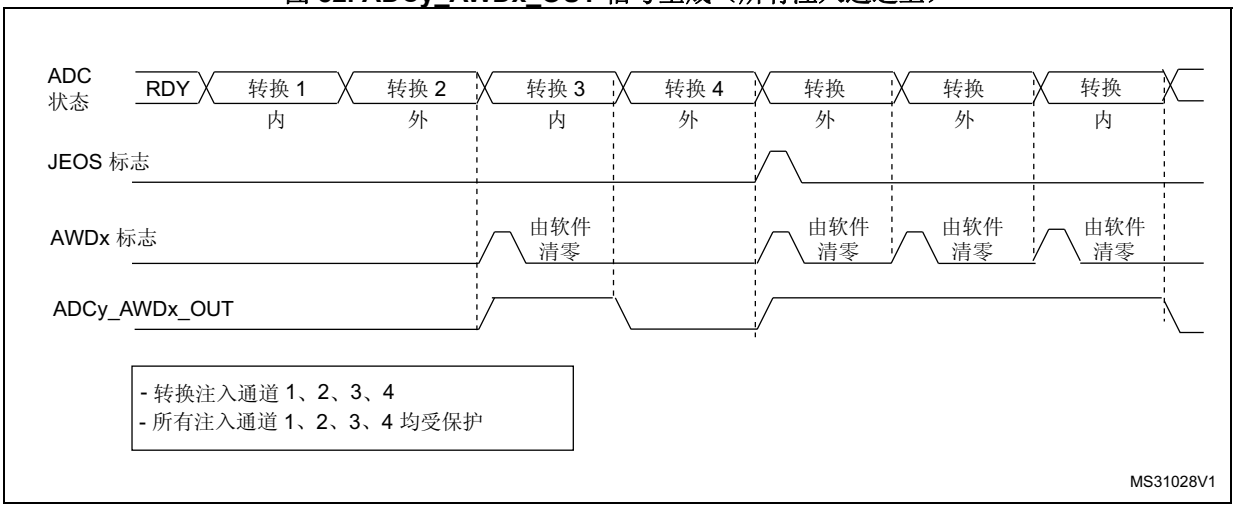


图 82. ADCy_AWDx_OUT 信号生成（所有注入通道上）



16.3.29 过采样器

过采样单元会进行数据预处理，以减轻 CPU 的负担。过采样单元可处理多个转换，并计算多个转换结果的平均值，得到数据宽度增大（高达 16 位）的单个数据。

它提供的结果采用以下形式，其中的 N 和 M 可以进行调整：

$$\text{结果} = \frac{1}{M} \times \sum_{n=0}^{n=N-1} \text{转换}(t_n)$$

允许通过硬件执行以下功能：计算平均值、降低数据速率、改进 SNR 以及基本滤波。

过采样率 N 通过 ADC_CFGR2 寄存器中的 OVFS[2:0] 位进行定义，范围为 2x 到 256x。分频系数 M 通过向右移位来实现（最多可移 8 位），并且通过 ADC_CFGR2 寄存器中的 OVSS[3:0] 位定义。

求和单元可得出多达 20 位的结果（256x 12 位结果），结果会先右移。随后会截断为 16 个最低有效位，这些位使用移位后剩下的最低有效位四舍五入为最接近的数值，然后将最终得到的结果传输到 ADC_DR 数据寄存器中。

注： *如果移位后得到的中间结果超过 16 位，则会将结果按原样截断（无饱和）。*

图 83. 20 位到 16 位结果的截断过程

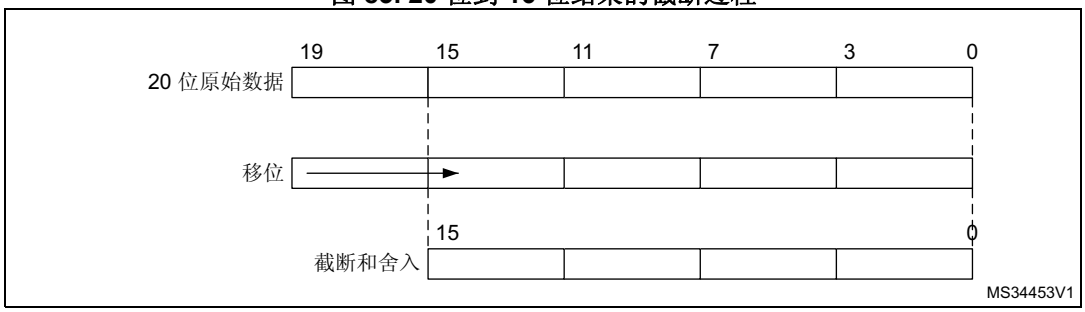


图 84 介绍了从原始的 20 位累加数据到最终 16 位结果的数值处理过程。

图 84. 移 5 位并进行舍入的数值示例

表 80 列出了原始转换数据等于 0xFFF 时对应的各种 N 和 M 组合的数据格式。

表 80. 最大输出结果与 N 和 M 的对应关系（灰色单元格表示截断的部分）

过采样率	最大值原始数据	不移位 OVSS = 0000	移 1 位 OVSS = 0001	移 2 位 OVSS = 0010	移 3 位 OVSS = 0011	移 4 位 OVSS = 0100	移 5 位 OVSS = 0101	移 6 位 OVSS = 0110	移 7 位 OVSS = 0111	移 8 位 OVSS = 1000
2x	0x1FFE	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040	0x020
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

过采样模式下的转换时序不会发生变化：在整个过采样序列中，采样时间保持不变。每完成 N 次转换都会提供新数据，等效延迟等于 N x T_{CONV} = N x (t_{SMPL} + t_{SAR})。各标志的置 1 情况如下：

- 每个采样阶段后都会将采样阶段结束标志 (EOSMP) 置 1
- 如果过采样结果可用，每完成 N 次转换都会发生转换结束事件 (EOC)
- 过采样数据序列完成后（即 N x 序列长度次转换之后），会发生序列结束事件 (EOS)

过采样时支持的 ADC 工作模式

在过采样模式下，大部分 ADC 工作模式都会保留：

- 单次转换或连续模式转换
- 可由软件或触发器启动 ADC 转换
- ADC 在转换过程中停止（中止）
- 通过 CPU 或 DMA 在支持溢出检测的情况下读取数据
- 低功耗模式 (AUTDLY)
- 可编程分辨率：在这种情况下，会按照与 12 位转换相同的方式对分辨率降低的转换值（根据 ADC_CFGR1 寄存器中的 RES[1:0] 位）进行累加、截断、四舍五入和移位

注： *处理过采样数据时，不可使用对齐模式。ADC_CFGR1 中的 ALIGN 位会被忽略，且数据始终采用右对齐格式。*

过采样模式下不支持偏移校正。如果 ROVSE 和/或 JOVSE 位置 1，ADC_OFRy 寄存器中 OFFSETy_EN 位的值会被忽略（视为复位）。

模拟看门狗

模拟看门狗功能会保留（AWDSGL 位和 AWDEN 位），但存在以下区别：

- 会忽略 RES[1:0] 位，始终会使用完整的 12 位值 HT[11:0] 和 LT[11:0] 进行比较
- 会比较 16 位过采样结果 ADC_DR[15:4] 的 12 个最高有效位

注： *移位位数较大时必须多加留意，因为这样会缩小比较范围。例如，如果过采样结果移了 4 位，得到 12 位右对齐数据，那么只能对 8 个数据位执行有效的模拟看门狗比较。比较操作会在 ADC_DR[11:4] 与 HT[0:7]/LT[0:7] 之间进行，并且 HT[11:8]/LT[11:8] 必须保持复位状态。*

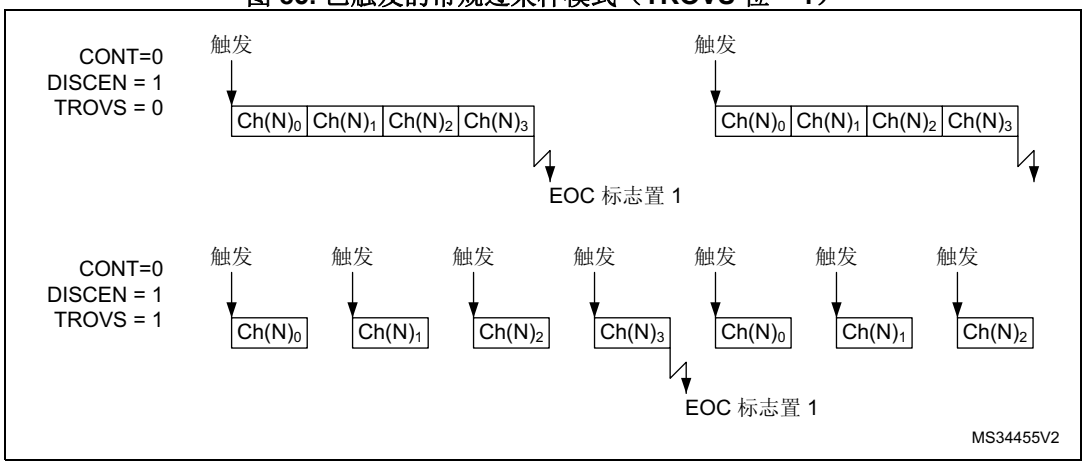
触发模式

平均值计算单元还可用于基本滤波，虽然它不是非常强大的滤波器（衰减缓慢、停止频段衰减受限），但可用作陷波滤波器，用于抑制恒定的寄生频率（通常来自输电线或开关电源）。为此，可使用 ADC_CFGR2 中的 TROVS 位使能特定的不连续模式，以获得由用户定义、且与转换时间本身无关的过采样频率。

图 85 显示了在不连续模式下如何响应触发从而开始转换。

如果 TROVS 位置 1，则会忽略 DISCEN 位的内容，并将此位视为 1。

图 85. 已触发的常规过采样模式（TROVS 位 = 1）



过采样时的注入和常规定序器管理

在过采样模式下，注入定序器和常规定序器可以执行不同操作。如果两个定序器必须同时使用，则可为它们使能过采样并设定一些限制条件（与唯一的累加单元相关）。

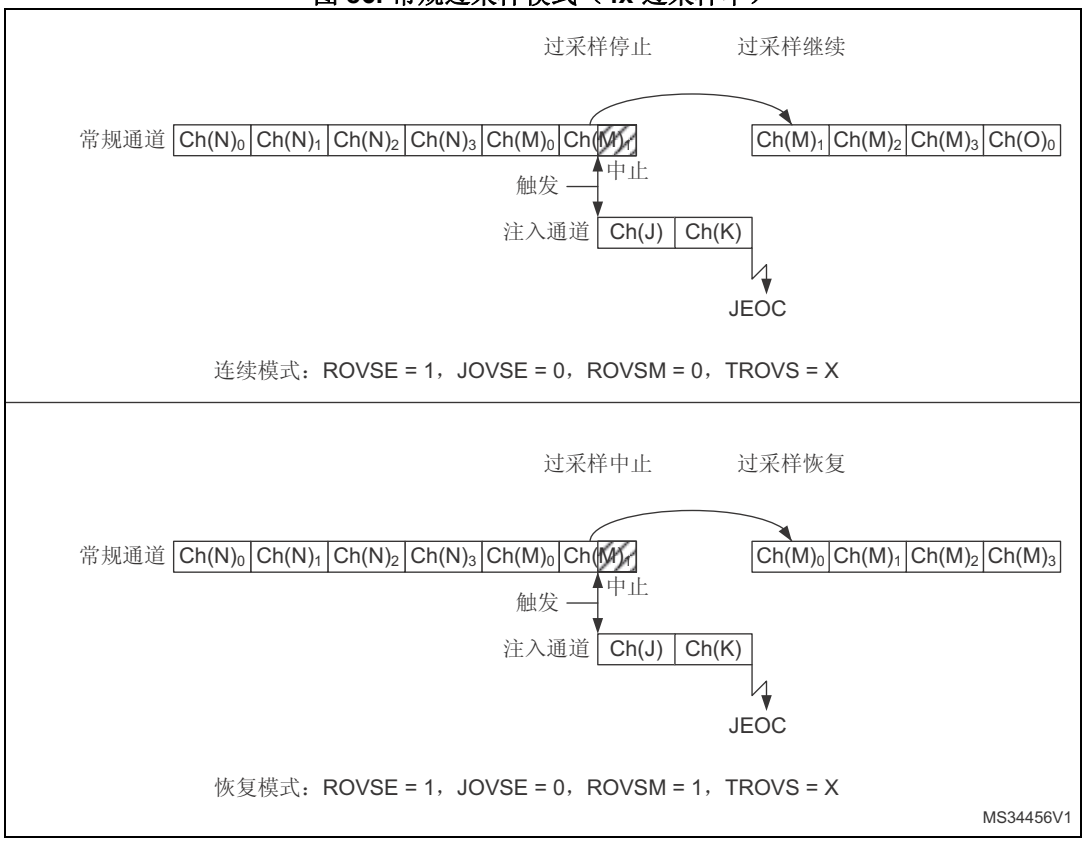
仅对常规通道进行过采样

常规过采样模式位 ROVSM 定义了常规过采样序列在被注入转换中断的情况下如何恢复：

- 在连续模式下，会从上一有效数据开始重新累加（在由于注入触发而发出的转换中止请求之前）。这样可确保在任何注入频率下均可而完成过采样（假设触发之间至少可完成一次常规转换）；
- 在恢复模式下，会从 0 开始重新累加（会忽略之前的转换结果）。该模式可确保所有用于过采样的数据在单个时隙内进行了背靠背转换。需要注意的是，注入触发周期必须超过过采样时长。如果该条件未得到满足，将无法完成过采样，常规定序器将被禁用。

图 86 以 4x 过采样率为例进行了说明。

图 86. 常规过采样模式（4x 过采样率）



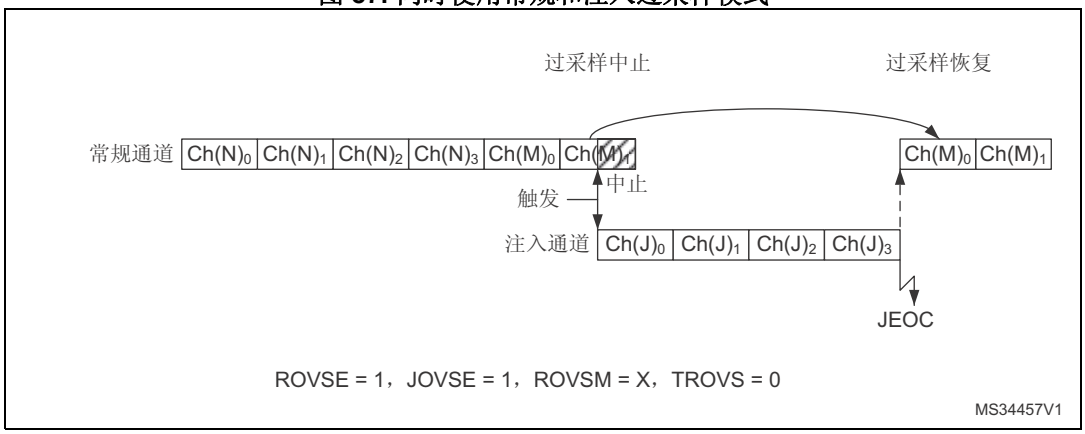
仅对注入通道进行过采样

注入过采样模式位 JOVSE 专为注入定序器中的转换使能过采样。

对常规通道和注入通道进行过采样

可以将 ROVSE 和 JOVSE 位都置 1。在这种情况下，常规过采样模式会强制进入恢复模式（忽略 ROVSM 位），如*图 87* 所示。

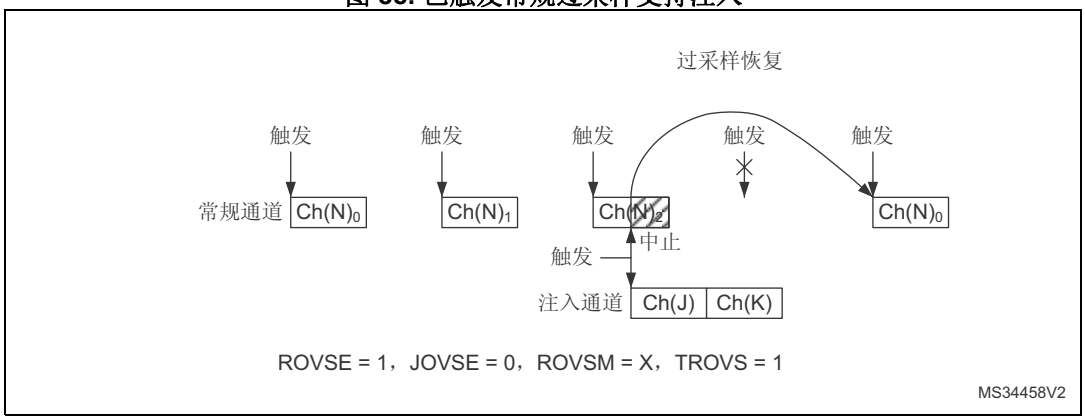
图 87. 同时使用常规和注入过采样模式



已触发常规过采样支持注入转换

已触发常规模式下支持注入转换。在这种情况下，必须禁止注入过采样模式，并且会忽略 ROVSM 位（强制进入恢复模式）。JOVSE 位必须复位。具体操作如*图 88* 所示。

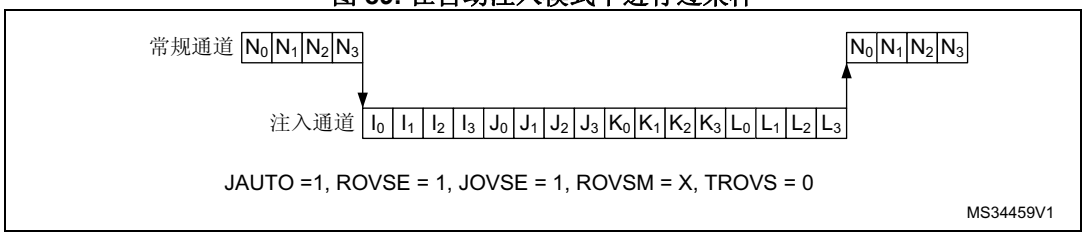
图 88. 已触发常规过采样支持注入



自动注入模式

可以对自动注入序列进行过采样，并将所有转换结果存储在寄存器中以节省 DMA 资源。使用该模式的前提条件是常规和注入过采样均激活：JAUTO = 1、ROVSE = 1 且 JOVSE = 1，不支持其他组合。在自动注入模式下，会忽略 ROVSM 位。*图 89* 显示了转换的排序方式。

图 89. 在自动注入模式下进行过采样



此外，还可以使用 TROVS 位使能触发模式。在这种情况下，ADC 必须进行如下配置：JAUTO = 1、DISCEN = 0、JDISCEN = 0、ROVSE = 1、JOVSE = 1 和 TROVSE = 1。

16.3.30 温度传感器

温度传感器可用于测量器件的结温 (Tj)。温度传感器内部连接到 ADC 输入通道，该通道用于将传感器输出电压转换为数字值。不使用时可将传感器置于掉电模式。支持的温度范围：-40 °C 到 125 °C。

图 90 显示的是温度传感器与 ADC 之间连接的框图。

温度传感器的输出电压随温度线性变化。由于工艺不同，该线的偏移量取决于各个芯片（芯片之间的温度变化可达 45 °C）。

未校准的内部温度传感器更适用于对温度变量而非绝对温度进行测量的应用。为提高温度传感器测量的准确性，ST 在生产过程中将校准值存储在每个器件的系统存储器中。

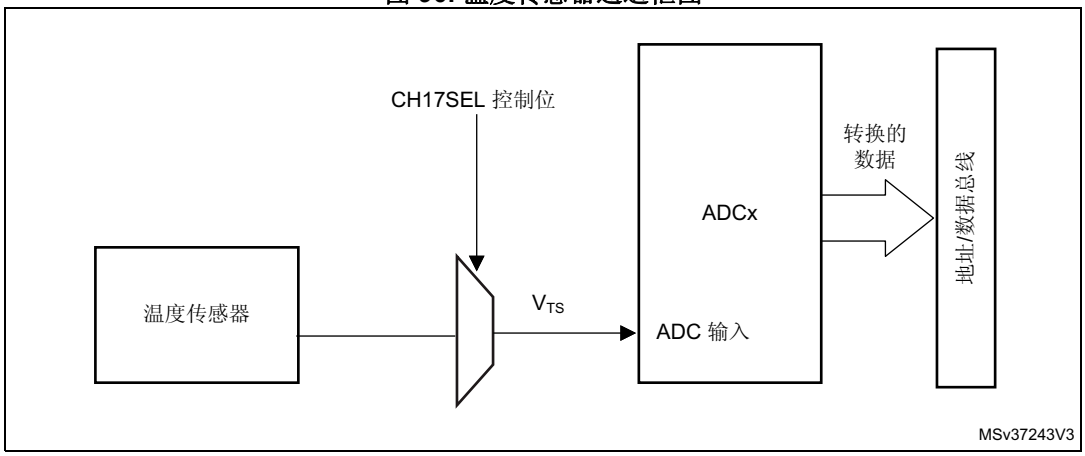
在制造过程中，会将温度传感器的校准数据和内部参考电压的校准数据存储在系统存储区。随后，用户应用可读取这些数据，并使用这些数据提高温度传感器或内部参考的准确性（更多相关信息，请参见数据手册）。

温度传感器在内部连接到 ADC 输入通道，该通道用于将传感器输出电压转换为数字值。有关转换内部温度传感器时要应用的采样时间值，请参见器件数据手册的电气特性部分。

不使用时可将传感器置于掉电模式。

图 90 显示了温度传感器框图。

图 90. 温度传感器通道框图



读取温度

要使用传感器，请执行以下操作：

1. 选择连接到 V_{TS} 的 ADC 输入通道。
2. 设定合适的采样时间（请参见器件数据手册中的电气特性部分）。
3. 在 ADCx_CCR 寄存器中将 CH17SEL 位置 1，以便将温度传感器从掉电模式中唤醒。
4. 开始 ADC 转换。
5. 读取 ADC 数据寄存器中生成的 V_{TS} 数据。

6. 使用以下公式计算实际温度：

$$\text{温度（以°C 为单位）} = \frac{\text{TS\_CAL2\_TEMP} - \text{TS\_CAL1\_TEMP}}{\text{TS\_CAL2} - \text{TS\_CAL1}} \times (\text{TS\_DATA} - \text{TS\_CAL1}) + 30\ °C$$

其中：

- TS_CAL2 是在 TS_CAL2_TEMP 下获得的温度传感器校准值。
- TS_CAL1 是在 TS_CAL1_TEMP 下获得的温度传感器校准值。
- TS_DATA 是由 ADC 转换得到的实际温度传感器输出值。
  更多关于 TS_CAL1 和 TS_CAL2 校准点的信息，请参见器件数据手册。

注： *传感器从掉电模式中唤醒需要一个启动时间，启动时间过后其才能正确输出 V_{TS}。ADC 在上电后同样需要一个启动时间，因此，为尽可能减少延迟，应同时将 ADEN 和 CH17SEL 位置 1。*

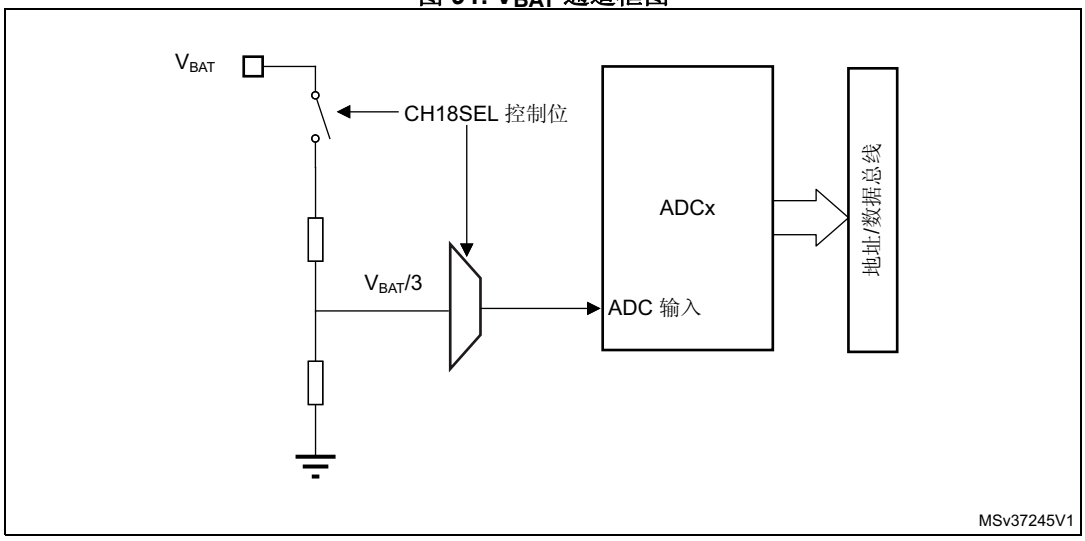
16.3.31 V_{BAT} 供电监测

ADCx_CCR 寄存器中的 CH18SEL 位用于切换到电池电压。由于 V_{BAT} 电压可能高于 V_{DDA}，为确保 ADC 正常工作，V_{BAT} 引脚需要在内部连接到桥接分配器（除以 3）。CH18SEL 置 1 时，会自动使能此桥，以将 $V_{BAT}/3$ 连接到 ADC 输入通道。因此，转换出的数字值为 V_{BAT} 电压的三分之一。为防止电池出现意外的电能消耗，建议仅在必要时为 ADC 转换使能桥接分配器。

有关转换 $V_{BAT}/3$ 电压时要应用的采样时间值，请参见器件数据手册的电气特性部分。

下图显示了 V_{BAT} 传感特性的框图。

图 91. V_{BAT} 通道框图



1. 必须将 CH18SEL 位置 1 才能为 $V_{BAT}/3$ 使能内部通道转换。

16.3.32 监测内部参考电压

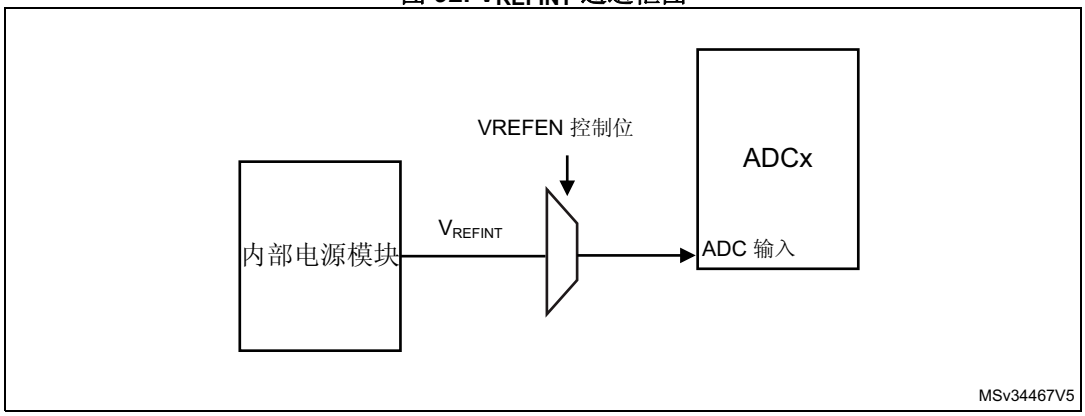
可以通过监测内部参考电压 (V_{REFINT}) 来获得用于评估 ADC V_{REF+} 电压的参考值。

内部参考电压在内部连接到 ADC1 的输入通道 0 (ADC1_INP0)。

有关转换内部参考电压时要应用的采样时间值，请参见产品数据手册的电气特性部分。

图 92 显示了 VREFINT 传感特性的框图。

图 92. V_{REFINT} 通道框图



1. 必须将 ADCx_CCR 寄存器中的 VREFEN 位置 1 才能使能内部通道 (V_{REFINT}) 的转换。

使用内部参考电压计算实际的 V_{DDA} 电压

施加给微控制器的 V_{DDA} 电源电压可能会有变化，或无法获得准确值。在制造过程中由 ADC 在 V_{DDA} = 3.0 V 的条件下获得的内置内部参考电压 (V_{REFINT}) 及其校准数据可用于评估实际的 V_{DDA} 电压水平。

以下公式可求得为器件供电的实际 V_{DDA} 电压：

$$V_{DDA} = 3.0\ V \times VREFINT\_CAL/VREFINT\_DATA$$

其中：

- VREFINT_CAL 是 VREFINT 校准值
- VREFINT_DATA 是由 ADC 转换得到的实际 VREFINT 输出值

将电源相关的 ADC 测量值转换为绝对电压值

ADC 用于提供对应于模拟电源与施加给转换通道的电压之比的数字值。对于大部分应用用例，需要将该比值转换成与 V_{DDA} 无关的电压。对于 V_{DDA} 已知、ADC 转换值进行了右对齐的应用，可使用以下公式得到该绝对值：

$$V_{CHANNELx} = \frac{V_{DDA}}{FULL\_SCALE} \times ADCx\_DATA$$

对于 V_{DDA} 值未知的应用，必须使用内部参考电压，V_{DDA} 可替换为*使用内部参考电压计算实际的 V_{DDA} 电压*部分提供的表达式，从而得出以下公式：

$$V_{CHANNELx} = \frac{3.0\ V \times VREFINT\_CAL \times ADCx\_DATA}{VREFINT\_DATA \times FULL\_SCALE}$$

其中：

- VREFINT_CAL 是 VREFINT 校准值
- ADC_DATA 是由 ADC 在通道 x 上测得的值（右对齐）
- VREFINT_DATA 是由 ADC 转换得到的实际 VREFINT 输出值
- FULL_SCALE 是 ADC 输出的最大数字值。例如，如果分辨率为 12 位，该值为 2^{12} - 1 = 4095，如果分辨率为 8 位，该值为 2^8 - 1 = 255

注： *如果执行 ADC 测量时使用的是输出格式而非 12 位右对齐格式，那么必须先将所有参数转换为兼容格式，然后再进行计算。*

16.4 ADC 中断

在以下情况下会生成中断：

- ADC 就绪后，ADC 上电（ADRDY 标志）
- 常规组的任何转换结束时（EOC 标志）
- 常规组的转换序列结束时（EOS 标志）
- 注入组的任何转换结束时（JEOC 标志）
- 注入组的转换序列结束时（JEOS 标志）
- 发生模拟看门狗检测时（AWD1、AWD2 和 AWD3 标志）
- 采样阶段结束时（EOSMP 标志）
- 发生数据溢出时（OVR 标志）
- 注入序列上下文队列溢出时（JQOVF 标志）

可以使用单独的中断使能位以提高灵活性。

表 81. ADC 中断

中断事件	事件标志	使能控制位
ADC 就绪	ADRDY	ADRDYIE
常规组的转换结束	EOC	EOCIE
常规组的转换序列结束	EOS	EOSIE
注入组的转换结束	JEOC	JEOCIE
注入组的转换序列结束	JEOS	JEOSIE
模拟看门狗 1 状态位置 1	AWD1	AWD1IE
模拟看门狗 2 状态位置 1	AWD2	AWD2IE
模拟看门狗 3 状态位置 1	AWD3	AWD3IE
采样阶段结束	EOSMP	EOSMPIE
上溢	OVR	OVRIE
注入上下文队列溢出	JQOVF	JQOVFIE

16.5 ADC 寄存器

有关寄存器说明中使用的缩写，请参见*第 58 页的第 1.2 节*。

16.5.1 ADC 中断和状态寄存器 (ADC_ISR)

ADC interrupt and status register

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JQOVF	AWD3	AWD2	AWD1	JEOS	JEOC	OVR	EOS	EOC	EOSMP	ADRDY
					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:11 保留，必须保持复位值。

位 10 **JQOVF**：注入上下文队列溢出 (Injected context queue overflow)
当注入上下文队列溢出时，此位会由硬件置 1。通过软件写入 1 可将此位清零。更多信息，请参见*第 16.3.21 节：注入转换的上下文队列*。
0：未发生注入上下文队列溢出（或标志事件已通过软件确认并清零）
1：发生注入上下文队列溢出

位 9 **AWD3**：模拟看门狗 3 标志 (Analog watchdog 3 flag)
当转换电压超过在 ADC_TR3 寄存器的 LT3[7:0] 和 HT3[7:0] 位域中编程的值时，硬件会将此位置 1。通过软件写入 1 可将此位清零。
0：未发生模拟看门狗 3 事件（或标志事件已通过软件确认并清零）
1：发生模拟看门狗 3 事件

位 8 **AWD2**：模拟看门狗 2 标志 (Analog watchdog 2 flag)
当转换电压超过在 ADC_TR2 寄存器的 LT2[7:0] 和 HT2[7:0] 位域中编程的值时，硬件会将此位置 1。通过软件写入 1 可将此位清零。
0：未发生模拟看门狗 2 事件（或标志事件已通过软件确认并清零）
1：发生模拟看门狗 2 事件

位 7 **AWD1**：模拟看门狗 1 标志 (Analog watchdog 1 flag)
当转换电压超过在 ADC_TR1 寄存器的 LT1[11:0] 和 HT1[11:0] 位域中编程的值时，硬件会将此位置 1。但需要通过软件清零。写入 1 可将此位清零。
0：未发生模拟看门狗 1 事件（或标志事件已通过软件确认并清零）
1：发生模拟看门狗 1 事件

位 6 **JEOS**：注入通道序列结束标志 (Injected channel end of sequence flag)
组内所有注入通道转换结束时，硬件将此位置 1。通过软件写入 1 可将此位清零。
0：注入转换序列未完成（或标志事件已通过软件确认并清零）
1：注入转换已完成

位 5 **JEOC**：注入通道转换结束标志 (Injected channel end of conversion flag)

当通道的每次注入转换结束，新数据出现在相应的 ADC_JDRy 寄存器时，会通过硬件将此位置 1。通过软件向此位写入 1，或读取相应的 ADC_JDRy 寄存器都可将此位清零。

0：注入通道转换未完成（或标志事件已通过软件确认并清零）

1：注入通道转换已完成

位 4 **OVR**：ADC 溢出 (ADC overrun)

此位在常规通道上发生溢出事件时由硬件置 1，这意味着在 EOC 标志已置 1 时，新转换已完成。通过软件写入 1 可将此位清零。

0：未发生溢出事件（或标志事件已通过软件确认并清零）

1：发生溢出

位 3 **EOS**：常规序列结束标志 (End of regular sequence flag)

常规通道序列转换结束后，硬件将此位置 1。通过软件写入 1 可将此位清零。

0：常规转换序列未完成（或标志事件已通过软件确认并清零）

1：常规转换序列已完成

位 2 **EOC**：转换结束标志 (End of conversion flag)

当通道的每次常规转换结束，新数据出现在 ADC_DR 寄存器时，会通过硬件将此位置 1。通过软件向此位写入 1，或读取 ADC_DR 寄存器都可将此位清零。

0：常规通道转换未完成（或标志事件已通过软件确认并清零）

1：常规通道转换已完成

位 1 **EOSMP**：采样结束标志 (End of sampling flag)

在任何通道（仅限常规通道）的转换过程中，当采样阶段结束时，会通过硬件将此位置 1。

0：采样阶段未结束（或标志事件已通过软件确认并清零）

1：采样阶段已结束

位 0 ADRDY：ADC 就绪 (ADC ready)

ADC 使能后（位 ADEN=1）以及 ADC 达到准备好接收转换请求的状态时，会通过硬件将此位置 1。通过软件写入 1 可将此位清零。

0：ADC 未准备好开始转换（或标志事件已通过软件确认并清零）

1：ADC 已准备好开始转换

16.5.2 ADC 中断使能寄存器 (ADC_IER)

ADC interrupt enable register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JQOVFIE	AWD3IE	AWD2IE	AWD1IE	JEOSIE	JEOCIE	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10 **JQOVFIE**：注入上下文队列溢出中断使能 (Injected context queue overflow interrupt enable)

此位由软件置 1 和清零，用于使能/禁止注入上下文队列溢出中断。

0：禁止注入上下文队列溢出中断。

1：使能注入上下文队列溢出中断。JQOVF 位置 1 时产生中断。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。

位 9 **AWD3IE**：模拟看门狗 3 中断使能 (Analog watchdog 3 interrupt enable)

此位由软件置 1 和清零，用于使能/禁止模拟看门狗 3 中断。

0：禁止模拟看门狗 3 中断。

1：使能模拟看门狗 3 中断。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 8 **AWD2IE**：模拟看门狗 2 中断使能 (Analog watchdog 2 interrupt enable)

此位由软件置 1 和清零，用于使能/禁止模拟看门狗 2 中断。

0：禁止模拟看门狗 2 中断。

1：使能模拟看门狗 2 中断。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 7 **AWD1IE**：模拟看门狗 1 中断使能 (Analog watchdog 1 interrupt enable)

此位由软件置 1 和清零，用于使能/禁止模拟看门狗 1 中断。

0：禁止模拟看门狗 1 中断。

1：使能模拟看门狗 1 中断。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 6 **JEOSIE**：注入转换序列结束中断使能 (End of injected sequence of conversions interrupt enable)

此位由软件置 1 和清零，用于使能/禁止注入转换序列结束中断。

0：禁止 JEOS 中断。

1：使能 JEOS 中断。JEOS 位置 1 时产生中断。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。

位 5 **JEOCIE**：注入转换结束中断使能 (End of injected conversion interrupt enable)

此位由软件置 1 和清零，用于使能/禁止注入转换结束中断。

0：禁止 JEOC 中断。

1：使能 JEOC 中断。JEOC 位置 1 时产生中断。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 4 **OVRIE**：溢出中断使能 (Overrun interrupt enable)

此位由软件置 1 和清零，用于使能/禁止常规转换的溢出中断。

0：禁止溢出中断。

1：使能溢出中断。OVR 位置 1 时产生中断。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 3 **EOSIE**：常规转换序列结束中断使能 (End of regular sequence of conversions interrupt enable)

此位由软件置 1 和清零，用于使能/禁止常规转换序列结束中断。

0：禁止 EOS 中断。

1：使能 EOS 中断。EOS 位置 1 时产生中断。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 2 **EOCIE**：常规转换结束中断使能 (End of regular conversion interrupt enable)

此位由软件置 1 和清零，用于使能/禁止常规转换结束中断。

0：禁止 EOC 中断。

1：使能 EOC 中断。EOC 位置 1 时产生中断。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 1 **EOSMPIE**：常规转换采样结束中断使能 (End of sampling flag interrupt enable for regular conversions)

此位由软件置 1 和清零，用于使能/禁止常规转换采样阶段结束中断。

0：禁止 EOSMP 中断。

1：使能 EOSMP 中断。EOSMP 位置 1 时产生中断。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 0 **ADRDYIE**：ADC 就绪中断使能 (ADC ready interrupt enable)

此位由软件置 1 和清零，用于使能/禁止 ADC 就绪中断。

0：禁止 ADRDY 中断。

1：使能 ADRDY 中断。ADRDY 位置 1 时产生中断。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

16.5.3 ADC 控制寄存器 (ADC_CR)

ADC control register

偏移地址：0x08

复位值：0x2000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAL	ADCALDIF	DEEPPWD	ADVREGEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JADSTP	ADSTP	JADSTART	ADSTART	ADDIS	ADEN
										rs	rs	rs	rs	rs	rs

位 31 **ADCAL**：ADC 校准 (ADC calibration)

此位由软件置 1，用于开始 ADC 校准。先对 ADCALDIF 位编程，以确定此校准应用于单端输入模式还是差分输入模式。

校准完成后，此位由硬件清零。

0：校准已完成。

1：写入 1 可校准 ADC。读取值为 1 表示正在进行校准。

注：　仅当 ADEN=0 时，才允许通过软件将 ADCAL 置 1 的方式启动校准。

仅当 ADEN=1、ADSTART=0 且 JADSTART=0（ADC 已使能，当前未进行任何转换）时，才允许通过软件对 ADC_CALFACT 执行写操作来更新校准系数。

位 30 **ADCALDIF**：校准的差分模式 (Differential mode for calibration)

此位由软件置 1 和清零，用于为校准配置单端输入模式或差分输入模式。

0：写入 ADCAL 将在单端输入模式下启动校准。

1：写入 ADCAL 将在差分输入模式下启动校准。

注：　仅当 ADC 已禁止、且当前未进行校准时（ADCAL=0、JADSTART=0、JADSTP=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对此位执行写操作。

位 29 **DEEPPWD**：深度掉电使能 (Deep-power-down enable)

此位由软件置 1 和清零，用于使 ADC 进入深度掉电模式。

0：ADC 未处于深度掉电模式

1：ADC 处于深度掉电模式（默认复位状态）

注：　仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、JADSTP=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对此位执行写操作。

位 28 **ADVREGEN**：ADC 稳压器使能 (ADC voltage regulator enable)

此位由软件置 1，用于使能 ADC 稳压器。

执行启动校准或使能 ADC 等操作之前，必须先使能 ADC 稳压器，并且软件必须等待稳压器启动时间。

0：禁止 ADC 稳压器。

1：使能 ADC 稳压器。

有关 ADC 稳压器使能和禁止序列的更多详细信息，请参见*第 16.3.6 节：ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN)*。

仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才能通过软件对此位域进行编程。

位 27:6 保留，必须保持复位值。

位 5 **JADSTP**：ADC 停止注入转换命令 (ADC stop of injected conversion command)

此位由软件置 1，用于停止和丢弃正在进行的注入转换（JADSTP 命令）。

当转换已有效丢弃、并且可重新配置 ADC 注入序列和触发时，会通过硬件将此位清零。随后，ADC 会准备好接收新的开始注入转换命令（JADSTART 命令）。

0：当前未执行 ADC 停止注入转换命令。

1：写入 1 可停止正在进行的注入转换。读取值为 1 表示正在执行 ADSTP 命令。

注： *仅当 JADSTART=1 且 ADDIS=0 时（ADC 已使能、最终会进行注入转换、并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 JADSTP 置 1*

在自动注入模式下 (JAUTO=1)，将 ADSTP 位置 1 会中止常规转换和注入转换（不要使用 JADSTP）

位 4 **ADSTP**：ADC 停止常规转换命令 (ADC stop of regular conversion command)

此位由软件置 1，用于停止和丢弃正在进行的常规转换（ADSTP 命令）。

当转换已有效丢弃、并且可重新配置 ADC 常规序列和触发时，会通过硬件将此位清零。随后，ADC 会准备好接收新的开始常规转换命令（ADSTART 命令）。

0：当前未执行 ADC 停止常规转换命令。

1：写入 1 可停止正在进行的常规转换。读取值为 1 表示正在执行 ADSTP 命令。

注： *仅当 ADSTART=1 且 ADDIS=0 时（ADC 已使能、最终会进行常规转换、并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 ADSTP 置 1。*

在自动注入模式下 (JAUTO=1)，将 ADSTP 位置 1 会中止常规转换和注入转换（不要使用 JADSTP）。

位 3 **JADSTART**：ADC 开始注入转换 (ADC start of injected conversion)

此位由软件置 1，用于开始 ADC 的注入通道转换。根据配置位 JEXTEN 的值，可以立即开始转换（软件触发配置），也可以在发生注入硬件触发事件后开始转换（硬件触发配置）。

此位通过硬件清零：

– 在单次转换模式下，如果选择了软件触发 (JEXTSEL=0x0)：出现注入转换序列结束 (JEOS) 标志时清零。

– 在所有情况下：执行完 JADSTP 命令后，由硬件将 JADSTP 位清零的同时清零。

0：当前未进行 ADC 注入转换。

1：写入 1 可开始注入转换。读取值为 1 表示 ADC 正在运行，最终会转换注入通道。

注： *仅当 ADEN=1 且 ADDIS=0 时（ADC 已使能，并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 JADSTART 置 1。*

在自动注入模式下 (JAUTO=1)，通过将 ADSTART 位置 1 开始常规转换和自动注入转换（JADSTART 必须保持清零）

位 2 **ADSTART**：ADC 开始常规转换 (ADC start of regular conversion)

此位由软件置 1，用于开始 ADC 的常规通道转换。根据配置位 EXTEN 的值，可以立即开始转换（软件触发配置），也可以在发生常规硬件触发事件后开始转换（硬件触发配置）。

此位通过硬件清零：

– 在单次转换模式下，如果选择了软件触发 (EXTSEL=0x0)：出现常规转换序列结束 (EOS) 标志时清零。

– 在所有情况下：执行完 ADSTP 命令后，由硬件将 ADSTP 位清零的同时清零。

0：当前未进行 ADC 常规转换。

1：写入 1 可开始常规转换。读取值为 1 表示 ADC 正在运行，最终会转换常规通道。

注： *仅当 ADEN=1 且 ADDIS=0 时（ADC 已使能，并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 ADSTART 置 1*

在自动注入模式下 (JAUTO=1)，通过将 ADSTART 位置 1 开始常规转换和自动注入转换（JADSTART 必须保持清零）

位 1 **ADDIS**：ADC 禁止命令 (ADC disable command)

此位由软件置 1，用于禁止 ADC（ADDIS 命令）并使其进入掉电状态（OFF 状态）。

ADC 已有效禁止后，会立即通过硬件将此位清零（此时也会通过硬件将 ADEN 清零）。

0：当前未执行 ADDIS 命令。

1：写入 1 可禁止 ADC。读取值为 1 表示正在执行 ADDIS 命令。

注： *仅当 ADEN=1、ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件将 ADDIS 置 1*

位 0 **ADEN**：ADC 使能控制 (ADC enable control)

此位由软件置 1，用于使能 ADC。ADRDY 标志置 1 后，ADC 将立即准备好运行。

如果 ADC 已禁止，则执行 ADDIS 命令后，将通过硬件对此位清零。

0：禁止 ADC（OFF 状态）

1：写入 1 来使能 ADC。

注： *仅当 ADC_CR 寄存器的所有位均为 0 时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0，但 ADVREGEN 位除外，此位必须为 1），才允许通过软件将 ADEN 置 1（并且软件必须等待稳压器的启动时间）*

16.5.4 ADC 配置寄存器 (ADC_CFGR)

ADC configuration register

偏移地址：0x0C

复位值：0x8000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JQDIS	AWD1CH[4:0]					JAUTO	JAWD1EN	AWD1EN	AWD1SGL	JQM	JDISCEN	DISCNUM[2:0]			DISCEN
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
Res.	AUTDLY	CONT	OVRMOD	EXTEN[1:0]		EXTSEL3	EXTSEL2	EXTSEL1	EXTSEL0	ALIGN	RES[1:0]		Res.	DMACFG	DMAEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

位 31 **JQDIS**：注入队列禁止 (Injected Queue disable)

此位由软件置 1 和清零，用于禁止注入队列机制：

0：使能注入队列

1：禁止注入队列

注： *仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行常规转换、也未进行注入转换），才允许通过软件对此位执行写操作。*

将 JQDIS 位置 1 或复位会导致注入队列被清空，JSQR 寄存器也会被清空。

位 30:26 **AWD1CH[4:0]**：模拟看门狗 1 通道选择 (Analog watchdog 1 channel selection)

这些位将由软件置 1 和清零。它们用于选择由模拟看门狗监控的输入通道。

00000：通过 AWD1 监控 ADC 模拟输入通道 0（仅在 ADC1 上可用）

00001：通过 AWD1 监控 ADC 模拟输入通道 1

.....

10010：通过 AWD1 监控 ADC 模拟输入通道 18

其他：保留，不得使用

注： *通过 AWD1CH 选择的通道必须也在 SQRi 或 JSQRi 寄存器中选择。*

仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 25 **JAUTO**：注入组自动转换 (Automatic injected group conversion)

通过软件将此位置 1 和清零可在常规组转换后分别使能/禁止注入组自动转换。

0：禁止注入组自动转换

1：使能注入组自动转换

注： *仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行常规转换、也未进行注入转换），才允许通过软件对此位执行写操作。*

位 24 **JAWD1EN**：注入通道上的模拟看门狗 1 使能 (Analog watchdog 1 enable on injected channels)

此位由软件置 1 和清零

0：在注入通道上禁止模拟看门狗 1

1：在注入通道上使能模拟看门狗 1

注： *仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。*

位 23 **AWD1EN**：常规通道上的模拟看门狗 1 使能 (Analog watchdog 1 enable on regular channels)

此位由软件置 1 和清零

0：在常规通道上禁止模拟看门狗 1

1：在常规通道上使能模拟看门狗 1

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 22 **AWD1SGL**：在单一通道或所有通道上使能看门狗 1 (Enable the watchdog 1 on a single channel or on all channels)

此位由软件置 1 和清零，用于在通过 AWD1CH[4:0] 位确定的通道或所有通道上使能模拟看门狗。

0：在所有通道上使能模拟看门狗 1

1：在单一通道上使能模拟看门狗 1

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 21 **JQM**：JSQR 队列模式 (JSQR queue mode)

此位由软件置 1 和清零。

此位定义空队列的管理方式。

0：JSQR 模式 0：队列从不为空，并会保留上一次写入 JSQR 的配置。

1：JSQR 模式 1：队列可以为空，队列为空时，会在上一个有效注入序列完成后立即在内部禁止注入序列的软件和硬件触发。

更多信息，请参见*第 16.3.21 节：注入转换的上下文队列*。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。

位 20 **JDISCEN**：注入通道的不连续采样模式 (Discontinuous mode on injected channels)

通过软件将此位置 1 和清零可使能/禁止组中注入通道的不连续采样模式。

0：禁止注入通道的不连续采样模式

1：使能注入通道的不连续采样模式

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。

不能同时使用自动注入模式和不连续模式：当 JAUTO 置 1 时，DISCEN 和 JDISCEN 位必须通过软件保持清零状态。

位 19:17 **DISCNUM[2:0]**：不连续采样模式通道计数 (Discontinuous mode channel count)

软件将写入这些位，用于定义在接收到外部触发后于不连续采样模式下转换的常规通道数。

000：1 个通道

001：2 个通道

...

111：8 个通道

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 16 **DISCEN**：常规通道的不连续模式 (Discontinuous mode for regular channels)

此位由软件置 1 和清零，用于使能/禁止常规通道的不连续模式。

0：禁止常规通道的不连续模式

1：使能常规通道的不连续模式

注： *不能同时使能不连续模式和连续模式：禁止同时将 DISCEN 和 CONT 位置 1。*

不能同时使用自动注入模式和不连续模式：当 JAUTO 置 1 时，DISCEN 和 JDISCEN 位必须通过软件保持清零状态。

仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 15 保留，必须保持复位值。

位 14 **AUTDLY**：延迟转换模式 (Delayed conversion mode)

此位由软件置 1 和清零，用于使能/禁止自动延迟转换模式。

0：自动延迟转换模式关闭

1：自动延迟转换模式开启

注： *仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。*

位 13 **CONT**：常规转换的单次/连续转换模式 (Single/continuous conversion mode for regular conversions)

此位由软件置 1 和清零。此位置 1 时，常规转换将持续进行，直到此位清零。

0：单次转换模式

1：连续转换模式

注： *不能同时使能不连续模式和连续模式：禁止同时将 DISCEN 和 CONT 位置 1。*

仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 12 **OVRMOD**：溢出模式 (Overrun Mode)

此位由软件置 1 和清零，用于配置数据溢出的管理方式。

0：如果检测到溢出，ADC_DR 寄存器会保留原有数据。

1：如果检测到溢出，ADC_DR 寄存器会被上一转换结果覆盖。

注： *仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。*

位 11:10 **EXTEN[1:0]**：常规通道的外部触发使能和极性选择 (External trigger enable and polarity selection for regular channels)

通过软件将这些位置 1 和清零可选择外部触发极性和使能常规组的触发。

00：禁止硬件触发检测（可通过软件启动转换）

01：在上升沿执行硬件触发检测

10：在下降沿执行硬件触发检测

11：在上升沿和下降沿都执行硬件触发检测

注： *仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。*

位 9:6 **EXTSEL[3:0]**：常规组的外部触发选择 (External trigger selection for regular group)

这些位可选择用于触发常规组转换的外部事件。

0000：事件 0
0001：事件 1
0010：事件 2
0011：事件 3
0100：事件 4
0101：事件 5
0110：事件 6
0111：事件 7
...
1111：事件 15

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5 **ALIGN**：数据对齐 (Data alignment)

此位由软件置 1 和清零，用于选择右对齐或左对齐。请参见*数据寄存器、数据对齐和偏移（ADC_DR、OFFSETy、 OFFSETy_CH、 ALIGN）*一节。

0：右对齐
1：左对齐

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 4:3 **RES[1:0]**：数据分辨率 (Data resolution)

通过软件写入这些位可选择转换的分辨率。

00：12 位
01：10 位
10：8 位
11：6 位

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 2 保留，必须保持复位值。

位 1 **DMACFG**：直接存储器访问配置 (Direct memory access configuration)

此位由软件置 1 和清零，用于在 DMA 两种工作模式之间进行选择，仅当 DMAEN=1 时，此位才有效。

0：选择 DMA 单次模式
1：选择 DMA 循环模式

更多详细信息，请参见*使用 DMA 管理转换*一节。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 0 **DMAEN**：直接存储器访问使能 (Direct memory access enable)

此位由软件置 1 和清零，用于使能 DMA 请求的生成。这样便可使用 DMA 自动管理转换的数据。有关详细信息，请参见*使用 DMA 管理转换*一节。

0：禁止 DMA
1：使能 DMA

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

16.5.5 ADC 配置寄存器 2 (ADC_CFGR2)

ADC configuration register 2

偏移地址：0x10

复位值：0x0000 0000

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

位 31:28 保留，必须保持复位值。

位 27:17 保留，必须保持复位值。

位 16:11 保留，必须保持复位值。

位 10 **ROVSM**：常规过采样模式 (Regular Oversampling mode)

此位由软件置 1 和清零，用于选择常规过采样模式。

0：连续模式：如果触发注入转换，过采样会暂时停止，并会在注入序列完成后继续（注入序列过程中会保留过采样缓冲区）

1：恢复模式：如果触发注入转换，当前过采样会中止，并会在注入序列完成后从头开始进行过采样（注入序列开始后，过采样缓冲区会清零）

注： 仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 9 **TROVS**：已触发常规过采样 (Triggered Regular Oversampling)

此位由软件置 1 和清零，以使能已触发过采样。

0：会在触发后连续完成某一通道的所有过采样转换

1：某一通道的每个过采样转换都需要重新触发

注： 仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 8:5 **OVSS[3:0]**：过采样移位 (Oversampling shift)

此位域由软件置 1 和清零，用于定义应用到原始过采样结果的右移位数。

0000：不发生移位

0001：移 1 位

0010：移 2 位

0011：移 3 位

0100：移 4 位

0101：移 5 位

0110：移 6 位

0111：移 7 位

1000：移 8 位

其他代码保留

注： 仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 4:2 **OVSR[2:0]**：过采样率 (Oversampling ratio)

此位域由软件置 1 和清零，用于定义过采样率。

000：2x

001：4x

010：8x

011：16x

100：32x

101：64x

110：128x

111：256x

注： 仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 1 **JOVSE**：注入过采样使能 (Injected Oversampling Enable)

此位由软件置 1 和清零，用于使能注入过采样。

0：禁止注入过采样

1：使能注入过采样

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作

位 0 **ROVSE**：常规过采样使能 (Regular Oversampling Enable)

此位由软件置 1 和清零，用于使能常规过采样。

0：禁止常规过采样

1：使能常规过采样

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作

16.5.6 ADC 采样时间寄存器 1 (ADC_SMPR1)

ADC sample time register 1

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:1]	
		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
SMP5_0	SMP4[2:0]			SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			SMP0[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29:0 **SMPx[2:0]**：通道 x 采样时间选择 (Channel x sampling time selection)

通过软件写入这些位可分别为各个通道选择采样时间。在采样周期期间，通道选择位必须保持不变。

000：2.5 个 ADC 时钟周期
001：6.5 个 ADC 时钟周期
010：12.5 个 ADC 时钟周期
011：24.5 个 ADC 时钟周期
100：47.5 个 ADC 时钟周期
101：92.5 个 ADC 时钟周期
110：247.5 个 ADC 时钟周期
111：640.5 个 ADC 时钟周期

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

16.5.7 ADC 采样时间寄存器 2 (ADC_SMPR2)

ADC sample time register 2

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SMP18[2:0]			SMP17[2:0]			SMP16[2:0]			SMP15[2:1]	
					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
SMP15_0	SMP14[2:0]			SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:27 保留，必须保持复位值。

位 26:0 **SMPx[2:0]**：通道 x 采样时间选择 (Channel x sampling time selection)

通过软件写入这些位可分别为各个通道选择采样时间。在采样周期期间，通道选择位必须保持不变。

000：2.5 个 ADC 时钟周期
001：6.5 个 ADC 时钟周期
010：12.5 个 ADC 时钟周期
011：24.5 个 ADC 时钟周期
100：47.5 个 ADC 时钟周期
101：92.5 个 ADC 时钟周期
110：247.5 个 ADC 时钟周期
111：640.5 个 ADC 时钟周期

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

16.5.8 ADC 看门狗阈值寄存器 1 (ADC_TR1)

ADC watchdog threshold register 1

偏移地址：0x20

复位值：0x0FFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	HT1[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LT1[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **HT1[11:0]**：模拟看门狗 1 阈值上限 (Analog watchdog 1 higher threshold)

这些位由软件写入，用于定义模拟看门狗 1 的阈值上限。

请参见*第 16.3.28 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*。

注：　仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 15:12 保留，必须保持复位值。

位 11:0 **LT1[11:0]**：模拟看门狗 1 阈值下限 (Analog watchdog 1 lower threshold)

这些位由软件写入，用于定义模拟看门狗 1 的阈值下限。

请参见*第 16.3.28 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*。

注：　仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

16.5.9 ADC 看门狗阈值寄存器 2 (ADC_TR2)

ADC watchdog threshold register 2

偏移地址：0x24

复位值：0x00FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT2[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT2[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **HT2[7:0]**：模拟看门狗 2 阈值上限 (Analog watchdog 2 higher threshold)
这些位由软件写入，用于定义模拟看门狗 2 的阈值上限。
请参见*第 16.3.28 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*。
注：　仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 15:8 保留，必须保持复位值。

位 7:0 **LT2[7:0]**：模拟看门狗 2 阈值下限 (Analog watchdog 2 lower threshold)
这些位由软件写入，用于定义模拟看门狗 2 的阈值下限。
请参见*第 16.3.28 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*。
注：　仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

16.5.10 ADC 看门狗阈值寄存器 3 (ADC_TR3)

ADC watchdog threshold register 3

偏移地址：0x28

复位值：0x00FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT3[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT3[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **HT3[7:0]**：模拟看门狗 3 阈值上限 (Analog watchdog 3 higher threshold)
这些位由软件写入，用于定义模拟看门狗 3 的阈值上限。
请参见*第 16.3.28 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*。
注：　仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 15:8 保留，必须保持复位值。

位 7:0 **LT3[7:0]**：模拟看门狗 3 阈值下限 (Analog watchdog 3 lower threshold)
这些位由软件写入，用于定义模拟看门狗 3 的阈值下限。
此看门狗将 LT3 的 8 位与转换后数据的高 8 位进行比较。
注：　仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

16.5.11 ADC 常规序列寄存器 1 (ADC_SQR1)

ADC regular sequence register 1

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ4[4:0]					Res.	SQ3[4:0]					Res.	SQ2[4]
			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
SQ2[3:0]				Res.	SQ1[4:0]					Res.	Res.	L[3:0]			
rw	rw	rw	rw		rw	rw	rw	rw	rw			rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **SQ4[4:0]**：常规序列中的第四次转换 (4th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第四次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 23 保留，必须保持复位值。

位 22:18 **SQ3[4:0]**：常规序列中的第三次转换 (3rd conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第三次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 17 保留，必须保持复位值。

位 16:12 **SQ2[4:0]**：常规序列中的第二次转换 (2nd conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第二次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 11 保留，必须保持复位值。

位 10:6 **SQ1[4:0]**：常规序列中的第一次转换 (1st conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第一次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5:4 保留，必须保持复位值。

位 3:0 **L[3:0]**：常规通道序列长度 (Regular channel sequence length)
通过软件写入这些位可定义常规通道转换序列中的转换总数。
0000：1 次转换
0001：2 次转换
...
1111：16 次转换
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

16.5.12 ADC 常规序列寄存器 2 (ADC_SQR2)

ADC regular sequence register 2

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ9[4:0]					Res.	SQ8[4:0]					Res.	SQ7[4]
			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
SQ7[3:0]				Res.	SQ6[4:0]					Res.	SQ5[4:0]				
rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **SQ9[4:0]**：常规序列中的第九次转换 (9th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第九次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 23 保留，必须保持复位值。

位 22:18 **SQ8[4:0]**：常规序列中的第八次转换 (8th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第八次转换

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 17 保留，必须保持复位值。

位 16:12 **SQ7[4:0]**：常规序列中的第七次转换 (7th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第七次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 11 保留，必须保持复位值。

位 10:6 **SQ6[4:0]**：常规序列中的第六次转换 (6th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第六次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5 保留，必须保持复位值。

位 4:0 **SQ5[4:0]**：常规序列中的第五次转换 (5th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第五次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

16.5.13 ADC 常规序列寄存器 3 (ADC_SQR3)

ADC regular sequence register 3

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ14[4:0]					Res.	SQ13[4:0]					Res.	SQ12[4]
			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
SQ12[3:0]				Res.	SQ11[4:0]					Res.	SQ10[4:0]				
rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **SQ14[4:0]**：常规序列中的第十四次转换 (14th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第十四次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 23 保留，必须保持复位值。

位 22:18 **SQ13[4:0]**：常规序列中的第十三次转换 (13th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第十三次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 17 保留，必须保持复位值。

位 16:12 **SQ12[4:0]**：常规序列中的第十二次转换 (12th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第十二次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 11 保留，必须保持复位值。

位 10:6 **SQ11[4:0]**：常规序列中的第十一次转换 (11th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第十一次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5 保留，必须保持复位值。

位 4:0 **SQ10[4:0]**：常规序列中的第十次转换 (10th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第十次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

16.5.14 ADC 常规序列寄存器 4 (ADC_SQR4)

ADC regular sequence register 4

偏移地址：0x3C

复位值：0x0000 0000

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

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

位 31:11 保留，必须保持复位值。

位 10:6 **SQ16[4:0]**：常规序列中的第十六次转换 (16th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第十六次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5 保留，必须保持复位值。

位 4:0 **SQ15[4:0]**：常规序列中的第十五次转换 (15th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第十五次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

16.5.15 ADC 常规数据寄存器 (ADC_DR)

ADC regular data register

偏移地址：0x40

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值。

位 15:0 **RDATA[15:0]**：已转换常规数据 (Regular Data converted)

这些位为只读。其中包含上一转换常规通道的转换结果。数据有左对齐和右对齐两种方式，如*第 16.3.26 节：数据管理*所述。

16.5.16 ADC 注入序列寄存器 (ADC_JSQR)

ADC injected sequence register

偏移地址：0x4C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	JSQ4[4:0]					Res.	JSQ3[4:0]					Res.	JSQ2[4:2]		
	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
JSQ2[1:0]		Res.	JSQ1[4:0]					JEXTEN[1:0]		JEXTSEL[3:0]				JL[1:0]	
rw	rw		rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:26 **JSQ4[4:0]**：注入序列中的第四次转换 (4th conversion in the injected sequence)

通过软件写入这些位，并将通道编号 (0..18) 分配为注入转换序列中的第四次转换。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对这些位执行写操作。

位 25 保留，必须保持复位值。

位 24:20 **JSQ3[4:0]**：注入序列中的第三次转换 (3rd conversion in the injected sequence)

通过软件写入这些位，并将通道编号 (0..18) 分配为注入转换序列中的第三次转换。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对这些位执行写操作。

位 19 保留，必须保持复位值。

位 18:14 **JSQ2[4:0]**：注入序列中的第二次转换 (2nd conversion in the injected sequence)

通过软件写入这些位，并将通道编号 (0..18) 分配为注入转换序列中的第二次转换。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对这些位执行写操作。

位 13 保留，必须保持复位值。

位 12:8 **JSQ1[4:0]**：注入序列中的第一次转换 (1st conversion in the injected sequence)

通过软件写入这些位，并将通道编号 (0..18) 分配为注入转换序列中的第一次转换。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对这些位执行写操作。

位 7:6 **JEXTEN[1:0]**：注入通道的外部触发使能和极性选择 (External Trigger Enable and Polarity Selection for injected channels)

通过软件将这些位置 1 和清零可选择外部触发极性和使能注入组的触发。

00：如果 JQDIS=0（队列使能），同时禁止硬件和软件触发检测

00：如果 JQDIS=1（队列禁止），禁止硬件触发检测（可通过软件启动转换）

01：在上升沿执行硬件触发检测

10：在下降沿执行硬件触发检测

11：在上升沿和下降沿都执行硬件触发检测

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对这些位执行写操作。

如果 JQM=1 且上下文队列为空，则会在内部禁止注入序列的软件和硬件触发（请参见第 16.3.21 节：注入转换的上下文队列）

位 5:2 **JEXTSEL[3:0]**：注入组的外部触发选择 (External Trigger Selection for injected group)

这些位可选择用于触发注入组转换的外部事件：

0000：事件 0
0001：事件 1
0010：事件 2
0011：事件 3
0100：事件 4
0101：事件 5
0110：事件 6
0111：事件 7
...
1111：事件 15

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对这些位执行写操作。

位 1:0 **JL[1:0]**：注入通道序列长度 (Injected channel sequence length)

通过软件写入这些位可定义注入通道转换序列中的转换总数。

00：1 次转换
01：2 次转换
10：3 次转换
11：4 次转换

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对这些位执行写操作。

16.5.17 ADC 偏移 y 寄存器 (ADC_OFRy)

ADC offset y register

偏移地址：0x60 + 0x04 * (y -1)，（y = 1 到 4）

复位值：0x0000 0000

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

位 31 **OFFSETy_EN**：偏移 y 使能 (Offset y Enable)

此位通过软件写入，用于使能和禁止编程到 OFFSETy[11:0] 位中的偏移。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 30:26 **OFFSETy_CH[4:0]**：数据偏移 y 的通道选择 (Channel selection for the Data offset y)

这些位通过软件写入，用于定义编程到 OFFSETy[11:0] 位中的偏移将应用于的通道。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 25:12 保留，必须保持复位值。

位 11:0 **OFFSETy[11:0]**：编程到 OFFSETy_CH[4:0] 位中的通道对应的数据偏移 y (Data offset y for the channel programmed into bits OFFSETy_CH[4:0])

这些位通过软件写入，用于定义在转换通道（可以是常规通道或注入通道）时从原始转换数据中减去的偏移 y。应用数据偏移 y 的通道必须在 OFFSETy_CH[4:0] 位中编程。转换结果可从 ADC_DR（常规转换）或 ADC_JDRyi 寄存器（注入转换）中读取。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

如果多个偏移 (OFFSETy) 指向同一通道，相减时只会考虑使用 x 值最小的偏移。

例如：如果 OFFSET1_CH[4:0]=4 且 OFFSET2_CH[4:0]=4，转换通道 4 时会减去 OFFSET1[11:0]。

16.5.18 ADC 注入通道 y 数据寄存器 (ADC_JDRy)

ADC injected channel y data register

偏移地址：0x80 + 0x04 * (y - 1)，（y = 1 到 4）

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 **JDATA[15:0]**：注入数据 (Injected data)

这些位为只读。它们包括来自注入通道 y 的转换结果。数据有左对齐和右对齐两种方式，如*第 16.3.26 节：数据管理*所述。

16.5.19 ADC 模拟看门狗 2 配置寄存器 (ADC_AWD2CR)

ADC Analog Watchdog 2 Configuration Register

偏移地址：0xA0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD2CH[18:16]		
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD2CH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18:0 **AWD2CH[18:0]**：模拟看门狗 2 通道选择 (Analog watchdog 2 channel selection)
这些位将由软件置 1 和清零。它们用于使能和选择由模拟看门狗 2 监控的输入通道。
AWD2CH[i] = 0：不通过 AWD2 监控 ADC 模拟输入通道 i
AWD2CH[i] = 1：通过 AWD2 监控 ADC 模拟输入通道 i
当 AWD2CH[18:0] = 000..0 时，会禁止模拟看门狗 2

注： 通过 AWD2CH 选择的通道必须也在 SQRi 或 JSQRi 寄存器中选择。
仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

16.5.20 ADC 模拟看门狗 3 配置寄存器 (ADC_AWD3CR)

ADC Analog Watchdog 3 Configuration Register

偏移地址：0xA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD3CH[18:16]		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD3CH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18:0 **AWD3CH[18:0]**：模拟看门狗 3 通道选择 (Analog watchdog 3 channel selection)
这些位将由软件置 1 和清零。它们用于使能和选择由模拟看门狗 3 监控的输入通道。
AWD3CH[i] = 0：不通过 AWD3 监控 ADC 模拟输入通道 i
AWD3CH[i] = 1：通过 AWD3 监控 ADC 模拟输入通道 i
当 AWD3CH[18:0] = 000..0 时，会禁止模拟看门狗 3

注： 通过 AWD3CH 选择的通道必须也在 SQRi 或 JSQRi 寄存器中选择。
仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

16.5.21 ADC 差分模式选择寄存器 (ADC_DIFSEL)

ADC Differential mode Selection Register

偏移地址：0xB0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIFSEL[18:16]		
													r	r	r

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

位 31:19 保留，必须保持复位值。

位 18:16 **DIFSEL[18:16]**：通道 18 到 16 的差分模式 (Differential mode for channels 18 to 16)
这些位为只读。强制这些通道进入单端输入模式（连接到单端 I/O 端口或内部通道）。

位 15:1 **DIFSEL[DIFSEL[15:1]**：通道 15 到 1 的差分模式 (Differential mode for channels 15 to 1)
这些位将由软件置 1 和清零。它们用于选择将通道配置为单端模式或差分模式。
DIFSEL[i] = 0：将 ADC 模拟输入通道 i 配置为单端模式
DIFSEL[i] = 1：将 ADC 模拟输入通道 i 配置为差分模式

注： 仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、JADSTP=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对这些位执行写操作。

位 0 **DIFSEL[0]**：通道 0 的差分模式 (Differential mode for channel 0)
此位为只读。强制该通道进入单端输入模式（连接到内部通道）。

16.5.22 ADC 校准系数 (ADC_CALFACT)

ADC Calibration Factors

偏移地址：0xB4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_D[6:0]						
									rw	rw	rw	rw	rw	rw	rw

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

位 31:23 保留，必须保持复位值。

位 22:16 **CALFACT_D[6:0]**：差分模式下的校准系数 (Calibration Factors in differential mode)

这些位可由硬件或软件写入。

差分输入校准完成后，会立即由硬件更新为校准系数。

软件可向这些位写入新的校准系数。如果新校准系数不同于当前存储于模拟 ADC 中的校准系数，启动新的差分校准后，会立即应用新校准系数。

注： 仅当 ADEN=1、ADSTART=0 且 JADSTART=0 时（ADC 已使能、当前未执行任何校准和转换），才允许通过软件对这些位执行写操作。

位 15:7 保留，必须保持复位值。

位 6:0 **CALFACT_S[6:0]**：单端模式下的校准系数 (Calibration Factors In single-ended mode)

这些位可由硬件或软件写入。

单端输入校准完成后，会立即由硬件更新为校准系数。

软件可向这些位写入新的校准系数。如果新校准系数不同于当前存储于模拟 ADC 中的校准系数，启动新的单端校准后，会立即应用新校准系数。

注： 仅当 ADEN=1、ADSTART=0 且 JADSTART=0 时（ADC 已使能、当前未执行任何校准和转换），才允许通过软件对这些位执行写操作。

16.6 ADC 通用寄存器

这些寄存器定义主 ADC 和从 ADC 共用的控制和状态寄存器：

16.6.1 ADC 通用状态寄存器 (ADC_CSR)

ADC common status register

偏移地址：0x00（该偏移地址与主 ADC 基址 + 0x300 相关）

复位值：0x0000 0000

该寄存器可提供不同 ADC 的状态位图像。但是，它为只读形式且不允许将不同的状态位清零。必须在对应的 ADC_ISR 寄存器中将其写为 0，才能将各个状态位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JQOVF_MST	AWD3_MST	AWD2_MST	AWD1_MST	JEOS_MST	JEOC_MST	OVR_MST	EOS_MST	EOC_MST	EOSMP_MST	ADRDY_MST
					r	r	r	r	r	r	r	r	r	r	r

位 31:11 保留，必须保持复位值。

位 10 **JQOVF_MST**：主 ADC 的注入上下文队列溢出标志 (Injected Context Queue Overflow flag of the master ADC)

此位是相应 ADC_ISR 寄存器中 JQOVF 位的副本。

位 9 **AWD3_MST**：主 ADC 的模拟看门狗 3 标志 (Analog watchdog 3 flag of the master ADC)

此位是相应 ADC_ISR 寄存器中 AWD3 位的副本。

位 8 **AWD2_MST**：主 ADC 的模拟看门狗 2 标志 (Analog watchdog 2 flag of the master ADC)
此位是相应 ADC_ISR 寄存器中 AWD2 位的副本。

位 7 **AWD1_MST**：主 ADC 的模拟看门狗 1 标志 (Analog watchdog 1 flag of the master ADC)
此位是相应 ADC_ISR 寄存器中 AWD1 位的副本。

位 6 **JEOS_MST**：主 ADC 注入序列结束标志 (End of injected sequence flag of the master ADC)
此位是相应 ADC_ISR 寄存器中 JEOS 位的副本。

位 5 **JEOC_MST**：主 ADC 注入转换结束标志 (End of injected conversion flag of the master ADC)
此位是相应 ADC_ISR 寄存器中 JEOC 位的副本。

位 4 **OVR_MST**：主 ADC 的溢出标志 (Overrun flag of the master ADC)
此位是相应 ADC_ISR 寄存器中 OVR 位的副本。

位 3 **EOS_MST**：主 ADC 常规序列结束标志 (End of regular sequence flag of the master ADC)
此位是相应 ADC_ISR 寄存器中 EOS 位的副本。

位 2 **EOC_MST**：主 ADC 常规转换结束标志 (End of regular conversion of the master ADC)
此位是相应 ADC_ISR 寄存器中 EOC 位的副本。

位 1 **EOSMP_MST**：主 ADC 采样阶段结束标志 (End of Sampling phase flag of the master ADC)
此位是相应 ADC_ISR 寄存器中 EOSMP 位的副本。

位 0 **ADRDY_MST**：主 ADC 就绪 (Master ADC ready)
此位是相应 ADC_ISR 寄存器中 ADRDY 位的副本。

16.6.2 ADC 通用控制寄存器 (ADC_CCR)

ADC common control register

偏移地址：0x08（该偏移地址与主 ADC 基址 + 0x300 相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH18SEL	CH17SEL	VREFEN	PRESC[3:0]				CKMODE[1:0]	
							rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:25 保留，必须保持复位值。

位 24 CH18SEL：CH18 选择 (CH18 selection)
通过软件将此位置 1 和清零可控制通道 18
0：禁止 V_{BAT} 通道
1：使能 V_{BAT} 通道

位 23 CH17SEL：CH17 选择 (CH17 selection)
通过软件将此位置 1 和清零可控制通道 17
0：禁止温度传感器通道
1：使能温度传感器通道

位 22 **VREFEN**：V_{REFINT} 使能 (V_{REFINT} enable)
通过软件将此位置 1 和清零可使能/禁止 V_{REFINT} 通道。
0：禁止 V_{REFINT} 通道
1：使能 V_{REFINT} 通道

位 21:18 **PRESC[3:0]**：ADC 预分频系数 (ADC prescaler)
这些位由软件置 1 和清零，用于选择 ADC 的时钟频率。该时钟为所有 ADC 所共用。
0000：输入 ADC 时钟未分频
0001：输入 ADC 时钟 2 分频
0010：输入 ADC 时钟 4 分频
0011：输入 ADC 时钟 6 分频
0100：输入 ADC 时钟 8 分频
0101：输入 ADC 时钟 10 分频
0110：输入 ADC 时钟 12 分频
0111：输入 ADC 时钟 16 分频
1000：输入 ADC 时钟 32 分频
1001：输入 ADC 时钟 64 分频
1010：输入 ADC 时钟 128 分频
1011：输入 ADC 时钟 256 分频
其他：保留

注： 仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对这些位执行写操作。仅当 CKMODE[1:0] = 0b00 时，才会应用 ADC 预分频值。

位 17:16 **CKMODE[1:0]**：ADC 时钟模式 (ADC clock mode)
这些位由软件置 1 和清零，用于定义 ADC 时钟方案（主 ADC 和从 ADC 共用）：
00：CK_ADCx (x=123)（异步时钟模式），在产品级生成（请参见*第 6 节：复位和时钟控制 (RCC)*）
01：HCLK/1（同步时钟模式）。仅当 AHB 时钟预分频器设为 1（RCC_CFGR 寄存器中的 HPRE[3:0] = 0xxx）且系统时钟占空比为 50% 时，才必须使能此配置。
10：HCLK/2（同步时钟模式）
11：HCLK/4（同步时钟模式）

在所有同步时钟模式下，从定时器触发到转换开始的延迟过程中，不存在抖动。

注： 仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对这些位执行写操作。

位 15:0 保留，必须保持复位值。

16.6.3 ADC 寄存器映射

表 82. ADC 寄存器映射和复位值

偏移	寄存器	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
0x00	**ADC_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVF	AWD3	AWD2	AWD1	JEOS	JEOC	OVR	EOS	EOC	EOSMP	ADRDY
	Reset value																						0	0	0	0	0	0	0	0	0	0	0

表 82. ADC 寄存器映射和复位值（续）

偏移	寄存器	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
0x04	**ADC_IER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVFIE	AWD3IE	AWD2IE	AWD1IE	JEOSIE	JEOCIE	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x08	**ADC_CR**	ADCAL	ADCALDIF	DEEPPWD	ADVREGEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JADSTP	ADSTP	JADSTART	ADSTART	ADDIS	ADEN
	Reset value	0	0	1	0																							0	0	0	0	0	0
0x0C	**ADC_CFGR**	JQDIS	AWD1CH[4:0]					JAUTO	JAWD1EN	AWD1EN	AWD1SGL	JQM	JDISCEN	DISCNUM [2:0]			DISCEN	Res.	AUTDLY	CONT	OVRMOD	EXTEN[1:0]		EXTSEL3	EXTSEL2	EXTSEL1	EXTSEL0	ALIGN	RES [1:0]		Res.	DMACFG	DMAEN
	Reset value	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	0	0		0	0
0x0C	**ADC_CFGR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ROVSM	TROVS	OVSS[3:0]			OVSR [2:0]			JOVSE	ROVSE	
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x14	**ADC_SMPR1**	Res.	Res.	SMP9 [2:0]			SMP8 [2:0]			SMP7 [2:0]			SMP6 [2:0]			SMP5 [2:0]			SMP4 [2:0]			SMP3 [2:0]			SMP2 [2:0]			SMP1 [2:0]			SMP0 [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
0x18	**ADC_SMPR2**	Res.	Res.	Res.	Res.	Res.	SMP18 [2:0]			SMP17 [2:0]			SMP16 [2:0]			SMP15 [2:0]			SMP14 [2:0]			SMP13 [2:0]			SMP12 [2:0]			SMP11 [2:0]			SMP10 [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
0x1C	Reserved	Res.																															
0x20	**ADC_TR1**	Res.	Res.	Res.	Res.	HT1[11:0]												Res.	Res.	Res.	Res.	LT1[11:0]												
	Reset value					1	1	1	1	1	1	1	1	1	1	1	1					0	0	0	0	0	0	0	0	0	0	0	0
0x24	**ADC_TR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT2[[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT2[7:0]							
	Reset value									1	1	1	1	1	1	1	1									0	0	0	0	0	0	0	0
0x28	**ADC_TR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT3[[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT3[7:0]							
	Reset value									1	1	1	1	1	1	1	1									0	0	0	0	0	0	0	0
0x2C	Reserved	Res.																															
0x30	**ADC_SQR1**	Res.	Res.	Res.	SQ4[4:0]					Res.	SQ3[4:0]					Res.	SQ2[4:0]					Res.	SQ1[4:0]					Res.	Res.	L[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
0x34	**ADC_SQR2**	Res.	Res.	Res.	SQ9[4:0]					Res.	SQ8[4:0]					Res.	SQ7[4:0]					Res.	SQ6[4:0]					Res.	SQ5[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
0x38	**ADC_SQR3**	Res.	Res.	Res.	SQ14[4:0]					Res.	SQ13[4:0]					Res.	SQ12[4:0]					Res.	SQ11[4:0]					Res.	SQ10[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
0x3C	**ADC_SQR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SQ16[4:0]					Res.	SQ15[4:0]				
	Reset value																						0	0	0	0	0		0	0	0	0	0
0x40	**ADC_DR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	regular RDATA[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44-0x48	Reserved	Res.																															
0x4C	**ADC_JSQR**	Res.	JSQ4[4:0]					Res.	JSQ3[4:0]					Res.	JSQ2[4:0]			Res.	JSQ1[4:0]				JEXTEN[1:0]	JEXTSEL [3:0]				JL[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	0	0	0	0	0
0x50-0x5C	Reserved	Res.																															

表 82. ADC 寄存器映射和复位值（续）

偏移	寄存器	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
0x60	**ADC_OFR1**	OFFSET1_EN	OFFSET1_CH[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET1[11:0]											
	Reset value	0	0	0	0	0	0															0	0	0	0	0	0	0	0	0	0	0	0
0x64	**ADC_OFR2**	OFFSET2_EN	OFFSET2_CH[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET2[11:0]											
	Reset value	0	0	0	0	0	0															0	0	0	0	0	0	0	0	0	0	0	0
0x68	**ADC_OFR3**	OFFSET3_EN	OFFSET3_CH[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET3[11:0]											
	Reset value	0	0	0	0	0	0															0	0	0	0	0	0	0	0	0	0	0	0
0x6C	**ADC_OFR4**	OFFSET4_EN	OFFSET4_CH[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET4[11:0]											
	Reset value	0	0	0	0	0	0															0	0	0	0	0	0	0	0	0	0	0	0
0x70-0x7C	Reserved	Res.																															
0x80	**ADC_JDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x84	**ADC_JDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x88	**ADC_JDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8C	**ADC_JDR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8C-0x9C	Reserved	Res.																															
0xA0	**ADC_AWD2CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD2CH[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA4	**ADC_AWD3CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD3CH[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA8-0xAC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xB0	**ADC_DIFSEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIFSEL[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xB4	**ADC_CALFACT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_D[6:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_S[6:0]							
	Reset value										0	0	0	0	0	0	0									0	0	0	0	0	0	0	0

表 83. ADC 寄存器映射和复位值（主 ADC 和从 ADC 通用寄存器）偏移 = 0x300

偏移	寄存器	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
0x00	**ADC_CSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVF_MST	AWD3_MST	AWD2_MST	AWD1_MST	JEOS_MST	JEOC_MST	OVR_MST	EOS_MST	EOC_MST	EOSMP_MST	ADRDY_MST
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x04	Reserved	Res.																															
0x08	**ADC_CCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH18SEL	CH17SEL	VREFEN	PRESC[3:0]				CKMODE[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value								0	0	0	0	0	0	0	0	0																

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

17 参考电压缓冲器 (VREFBUF)

17.1 简介

STM32WB55xx 器件内置有参考电压缓冲器，既可用作 ADC 的参考电压，也可通过 VREF+ 引脚用作外部元件的参考电压。当 VREF+ 引脚与 VDDA 引脚在封装中互相绑定时，参考电压缓冲器不可用且必须禁止（关于封装引脚分配说明，请参见数据手册）。

17.2 VREFBUF 功能说明

内部参考电压缓冲器支持两种电压值[a]，可利用 VREFBUF_CSR 寄存器中的 VRS 位进行配置：

- VRS = 0：V_{REF_OUT1} 大约为 2.048 V。
- VRS = 1：V_{REF_OUT2} 大约为 2.5 V。

内部参考电压配置为四种不同模式，具体取决于 ENVR 和 HIZ 位的配置。下表列出了这些模式：

表 84. VREF 缓冲器模式

ENVR	HIZ	VREF 缓冲器配置
0	0	VREFBUF 缓冲器关闭： – V_{REF+} 引脚下拉到 V_{SSA}
0	1	外部参考电压模式（默认值）： – VREFBUF 缓冲器关闭 – V_{REF+} 引脚输入模式
1	0	内部参考电压模式： – VREFBUF 缓冲器开启 – V_{REF+} 引脚连接到 VREFBUF 缓冲器输出
1	1	保持模式： – VREFBUF 缓冲器关闭 – V_{REF+} 引脚浮空。借助外部电容来保持电压 – 禁止 VRR 检测，VRR 位保持最后一个状态

通过将 VREFBUF_CSR 寄存器中的 ENVR 位置 1 并将 HIZ 位清零使能 VREFBUF 后，用户必须等待至 VRR 位置 1，指示参考电压输出已达到预期值。

a. 最小 V_{DDA} 电压取决于 VRS 设置，请参见产品数据手册。

17.3 VREFBUF 寄存器

17.3.1 VREFBUF 控制和状态寄存器 (VREFBUF_CSR)

VREFBUF control and status register

偏移地址：0x00

复位值：0x0000 0002

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

位 31:4 保留，必须保持复位值。

位 3 **VRR**：参考电压缓冲器就绪 (Voltage reference buffer ready)
0：参考电压缓冲器输出未就绪。
1：参考电压缓冲器输出已达到请求值。

位 2 **VRS**：参考电压调节 (Voltage reference scale)
此位选择由参考电压缓冲器生成的值。
0：参考电压设为 V_{REF_OUT1}（约 2.048 V）。
1：参考电压设为 V_{REF_OUT2}（约 2.5 V）。

位 1 **HIZ**：高阻态模式 (High impedance mode)
此位控制模拟开关是否连接到 V_{REF+} 引脚。
0：V_{REF+} 从内部连接到参考电压缓冲器输出。
1：V_{REF+} 引脚为高阻态。
关于 ENVR 位配置及模式说明，请参见*表 84：VREF 缓冲器模式*。

位 0 **ENVR**：参考电压缓冲器模式使能 (Voltage reference buffer mode enable)
此位用于使能参考电压缓冲器模式。
0：禁止内部参考电压模式（外部参考电压模式）。
1：使能内部参考电压模式（参考缓冲器使能或保持模式）。

17.3.2 VREFBUF 校准控制寄存器 (VREFBUF_CCR)

VREFBUF calibration control register

偏移地址：0x04

复位值：0x0000 00XX

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

位 31:6 保留，必须保持复位值。

位 5:0 **TRIM[5:0]**：微调代码 (Trimming code)

复位后，这些位将被自动使用生产测试时存储于 FLASH 的微调值初始化。写入这些位可调整内部参考缓冲器电压。

17.3.3 VREFBUF 寄存器映射

下表列出了 VREFBUF 寄存器映射和复位值。

表 85. VREFBUF 寄存器映射和复位值

偏移	寄存器名称	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
0x00	**VREFBUF_CSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VRR	VRS	HIZ	ENVR
	Reset value																													0	0	1	0
0x04	**VREFBUF_CCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM[5:0]					
	Reset value																											x	x	x	x	x	x

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

18 比较器 (COMP)

18.1 简介

器件嵌入 2 个超低功耗比较器 COMP1 和 COMP2。

这两个比较器可用于多种功能，包括：

- 在模拟信号的触发下从低功耗模式唤醒。
- 模拟信号调理。
- 与定时器的 PWM 输出结合使用时，构成逐周期电流控制环路。

18.2 COMP 主要特性

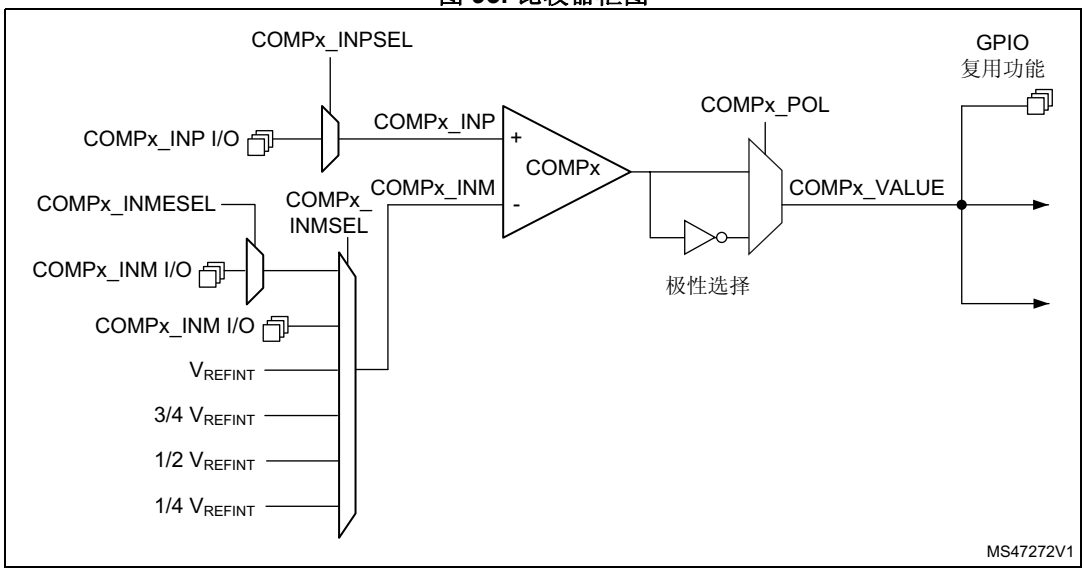
- 每个比较器都具有可配置的正输入和负输入，用于灵活选择电压：
  - 复用 I/O 引脚
  - 通过调节器（缓冲分压器）提供的内部参考电压和三个因数分压值（1/4、1/2、3/4）
- 可编程迟滞
- 可编程速度/功耗
- 输出可以重定向到用于触发以下事件的 I/O 或定时器输入：
  - 断路事件（用于快速 PWM 关断）
- 消隐源比较器输出
- 两个比较器可以组合构成窗口比较器
- 每个比较器都可产生中断，用于使器件从睡眠模式和停止模式唤醒（通过 EXTI 控制器）

18.3 COMP 功能说明

18.3.1 COMP 框图

比较器的框图如*图 93：比较器框图*所示。

图 93. 比较器框图



18.3.2 COMP 引脚和内部信号

用作比较器输入的 I/O 必须在 GPIO 寄存器中配置为模拟模式。

比较器输出可以使用数据手册的“复用功能映射”表中给出的复用功能通道连接到 I/O。

输出也可以在内部重定向到用于以下用途的各种定时器输入：

- 使用 BKIN 和 BKIN2 输入紧急关断 PWM 信号
- 使用 OCREF_CLR 输入进行逐周期电流控制
- 用于时序测量的输入捕捉

可以在内部和外部同时对比较器输出进行重定向。

表 86. COMP1 正输入分配

COMP1_INP	COMP1_INPSEL
PC5	00
PB2	01
PA1	10

表 87. COMP1 负输入分配

COMP1_INM	COMP1_INMSEL[2:0]	COMP1_INMESEL[1:0]
¼ V_{REFINT}	000	N. A.(1)
½ V_{REFINT}	001	N. A.(1)
¾ V_{REFINT}	010	N. A.(1)
V_{REFINT}	011	N. A.(1)
保留	100	N. A.(1)
保留	101	N. A.(1)
PA9	110	N. A.(1)
PC4	111	00
PA0	111	01
PA4	111	10
PA5	111	11

1. N. A.：不受影响。

表 88. COMP2 正输入分配

COMP2_INP	COMP2_INPSEL
PB4	00
PB6	01
PA3	10

表 89. COMP2 负输入分配

COMP2_INM	COMP2_INMSEL[2:0]	COMP2_INMESEL[1:0]
¼ V_{REFINT}	000	N.A.(1)
½ V_{REFINT}	001	N.A.(1)
¾ V_{REFINT}	010	N.A.(1)
V_{REFINT}	011	N.A.(1)
保留	100	N.A.(1)
保留	101	N.A.(1)
PB3	110	N.A.(1)
PB7	111	00
PA2	111	01
PA4	111	10
PA5	111	11

1. N. A.：不受影响。

18.3.3 COMP 复位和时钟

时钟控制器提供的 COMP 时钟与 APB2 时钟同步。

RCC 控制器中不单独提供其时钟使能控制位。COMP 和 SYSCFG 共用复位和时钟使能位。

注： ***重要提示：****极性选择逻辑和到端口的输出重定向独立于 APB2 时钟。因此，即使在停止模式下比较器仍能正常工作。*

18.3.4 比较器锁定机制

这两个比较器可用于过流或热保护等安全用途。对于具有特定功能安全要求的应用，必须保证在发生意外寄存器访问或程序计数器损坏时，不能更改比较器编程。

为此，可以对比较器控制和状态寄存器进行写保护（只读）。

一旦编程完成，COMPx LOCK 位便会置 1。这将导致整个寄存器变成只读，包括 COMPx LOCK 位。

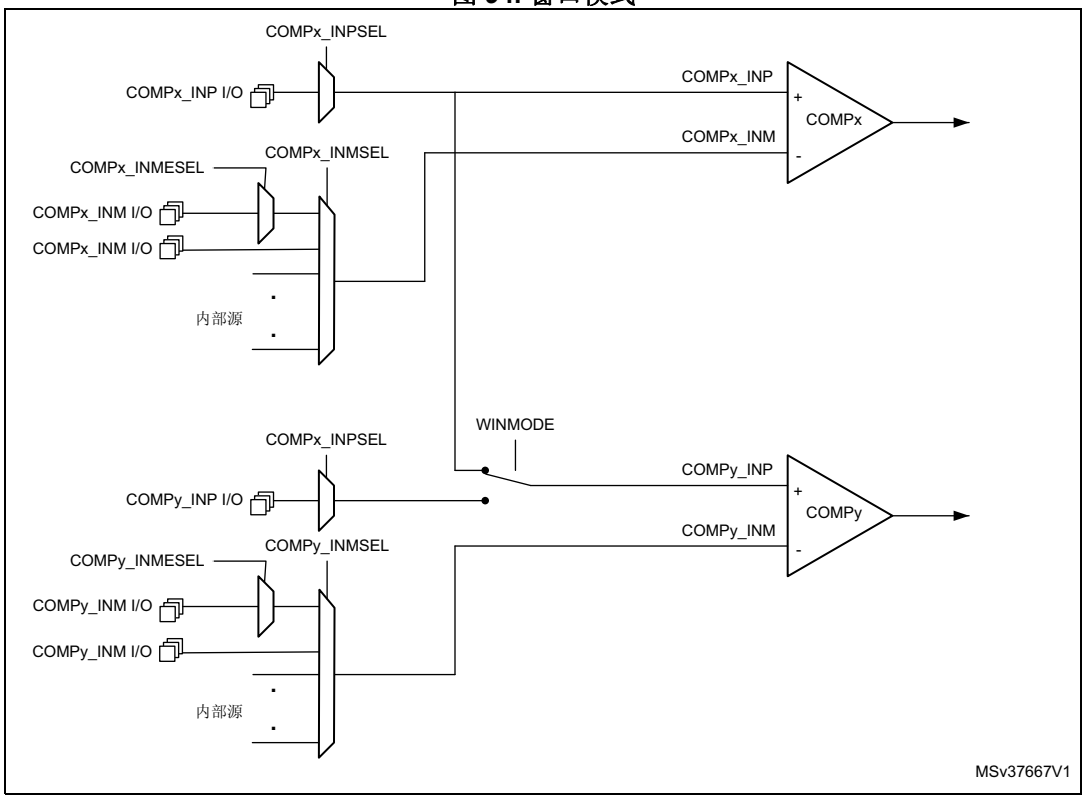
只能通过 MCU 复位来复位写保护。

18.3.5 窗口比较器

窗口比较器用于监视模拟电压是否处于阈值上下限所定义的特定电压范围内。

可使用两个嵌入式比较器创建窗口比较器。受监视的模拟电压连接到连在一起的比较器的非反相 (+) 输入，阈值上下限电压连接到比较器的反相 (-) 输入。可通过使能 WINMODE 位将两个非反相输入在内部连在一起，进而保留一个 IO 用于其他用途。

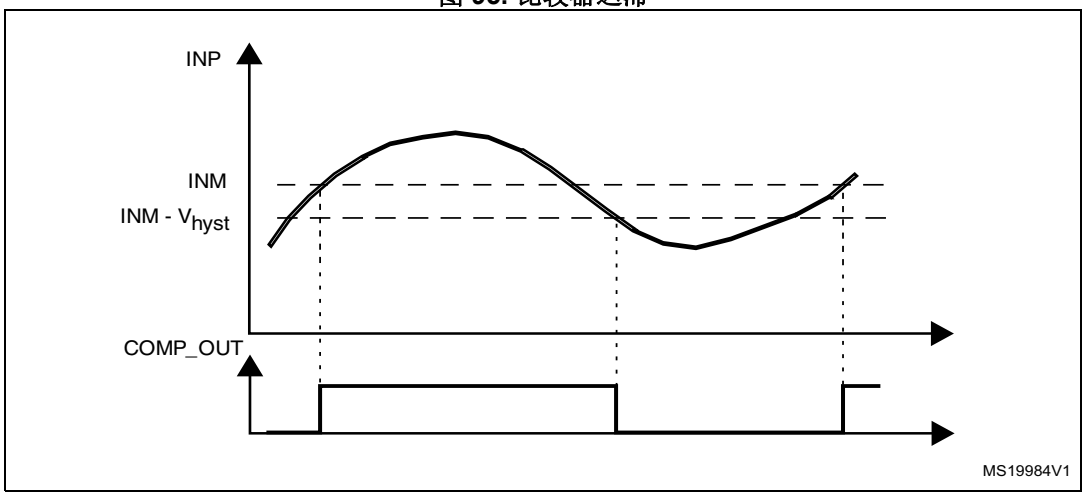
图 94. 窗口模式



18.3.6 迟滞

比较器具有可编程迟滞，可在有噪声信号时避免发生意外输出转换。迟滞可在不需要时（例如，退出低功耗模式时）禁止，以便使用外部组件强制迟滞值。

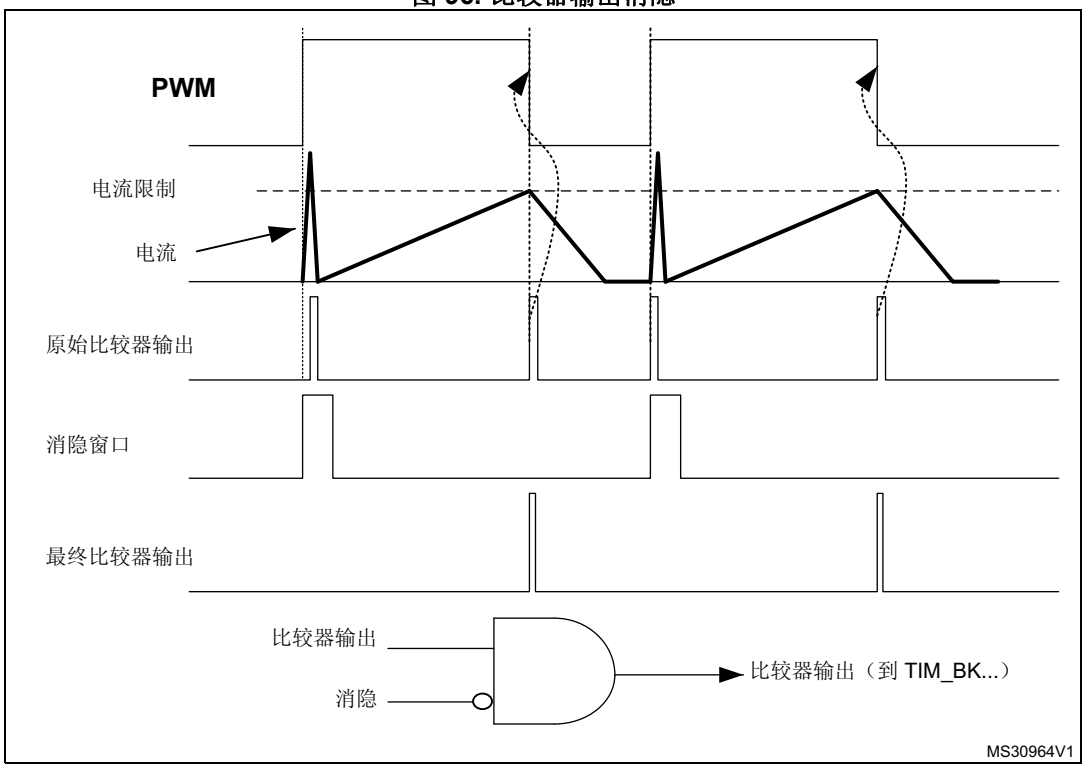
图 95. 比较器迟滞



18.3.7 比较器输出消隐功能

消隐功能的目的是防止电流调节在 PWM 周期开始处出现短暂电流尖峰（通常为功率开关反向并联二极管中的恢复电流）时发生跳闸。其包括选择的消隐窗口，对应定时器比较输出信号。通过软件进行选择（请参见比较器寄存器说明，了解可能的消隐信号）。然后，将消隐信号的补码与比较器输出执行逻辑“与”运算，以提供所需比较器输出。请参见下图所示的示例。

图 96. 比较器输出消隐



18.3.8 COMP 功耗和速度模式

对于给定的应用，可调节 COMP1 和 COMP2 功耗与传播延迟以获得最佳平衡。

COMPx_CSR 寄存器的位 PWRMODE[1:0] 可编程为如下值：

00：高速/全功耗

01 或 10：中速/中等功耗

11：低速/超低功耗

18.4 COMP 低功耗模式

表 90. 低功耗模式下的比较器行为

模式	说明
睡眠	对比较器无影响。 比较器中断可使器件退出睡眠模式。
低功耗运行	无影响。
低功耗睡眠	无影响。COMP 中断可使器件退出低功耗睡眠模式。
停止 0	对比较器无影响。 比较器中断可使器件退出停止模式。
停止 1	
停止 2	
待机	COMP 寄存器掉电，退出待机或关断模式后必须重新初始化。
关断	

18.5 COMP 中断

比较器输出从内部连接到扩展中断和事件控制器。每个比较器都有其各自的 EXTI 线，能够产生中断或事件。该机制还可用于退出低功耗模式。

更多详细信息，请参见中断和事件部分。

要使能 COMPx 中断，需要遵循以下顺序：

1. 将对应于 COMPx 输出事件的 EXTI 线配置为中断模式并将其使能，然后选择上升沿有效、下降沿有效或二者均有效
2. 配置并使能映射到相应 EXTI 线的 NVIC IRQ 通道
3. 使能 COMPx

表 91. 中断控制位

中断事件	事件标志	使能控制位	退出睡眠模式	退出停止模式	退出待机模式
COMP1 输出	COMP1_CSR 中的 VALUE	通过 EXTI	是	是	N/A
COMP2 输出	COMP2_CSR 中的 VALUE	通过 EXTI	是	是	N/A

18.6 COMP 寄存器

18.6.1 比较器 1 控制和状态寄存器 (COMP1_CSR)

Comparator 1 control and status register

COMP1_CSR 为比较器 1 控制/状态寄存器。此寄存器包含与比较器 1 相关的所有位/标志。

偏移地址：0x00

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	VALUE	Res.	Res.	Res.	INMESEL		Res.	SCAL EN	BRG EN	Res.	BLANKING			HYST	
rs	r				rw			rw	rw		rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLA RITY	Res.	Res.	Res.	Res.	Res.	Res.	INP SEL.		INMSEL			PWRMODE		Res.	EN
rw							rw		rw			rw			rw

位 31 **LOCK**：COMP1_CSR 寄存器锁定位 (COMP1_CSR register lock bit)

此位由软件置 1，通过硬件系统复位清零。它将锁定比较器 1 控制寄存器 (COMP1_CSR[31:0]) 的全部内容。

0：比较器 1 的 COMP1_CSR[31:0] 可读/写

1：比较器 1 的 COMP1_CSR[31:0] 只读

位 30 **VALUE**：比较器 1 输出状态位 (Comparator 1 output status bit)

此位为只读。它反映当前比较器 1 输出的状态（受到 POLARITY 位的影响）。

位 29:27 保留，必须保持复位值。

位 26:25 **INMESEL**：比较器 1 负输入扩展选择位 (comparator 1 input minus extended selection bits)

这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它们选择哪个扩展 GPIO 输入连接到比较器的负输入（如果 INMSEL = 111）。

00：PC4

01：PA0

10：PA4

11：PA5

位 24 保留，必须保持复位值。

位 23 **SCALEN**：电压调节器使能位 (Voltage scaler enable bit)

此位由软件置 1 和清零。此位使能比较器 1 负输入上的 V_{REFINT} 分压器输出。

0：禁止带隙调节器（COMP2_CSR 寄存器的 SCALEN 位也复位时）

1：使能带隙调节器

位 22 **BRGEN**：调节器电阻桥使能 (Scaler bridge enable)

此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。此位可使能调节器电阻桥。

0：禁止调节器电阻桥（COMP2_CSR 寄存器的 BRGEN 位也复位时）

1：使能调节器电阻桥

如果 SCALEN 置 1 且 BRGEN 复位，则 BG 参考电压可用，但 1/4 BGAP、1/2 BGAP 和 3/4 BGAP 不可用。发送 BGAP 值，而非 1/4 BGAP、1/2 BGAP、3/4 BGAP。

如果 SCALEN 和 BRGEN 置 1，则 1/4 BGAP、1/2 BGAP、3/4 BGAP 和 BGAP 参考电压可用。

位 21 保留，必须保持复位值

位 20:18 **BLANKING[2:0]**：比较器 1 消隐源选择位 (Comparator 1 blanking source selection bits)
这些位选择哪个定时器输出控制比较器 1 输出消隐。
000：无消隐
001：选择 TIM1 OC5 作为消隐源
010：选择 TIM2 OC3 作为消隐源
所有其他值：保留

位 17:16 **HYST[1:0]**：比较器 1 迟滞选择位 (Comparator 1 hysteresis selection bits)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它们选择比较器 1 的迟滞电压。
00：无迟滞
01：低迟滞
10：中等迟滞
11：高迟滞

位 15 **POLARITY**：比较器 1 极性选择位 (Comparator 1 polarity selection bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它使比较器 1 的极性反相。
0：比较器 1 输出值不反相
1：比较器 1 输出值反相

位 14:9 保留，必须保持复位值。

位 8:7 **INPSEL**：比较器 1 正输入选择位 (Comparator 1 input plus selection bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。
00：外部 I/O - PC5
01：PB2
10：PA1
11：保留

位 6:4 **INMSEL**：比较器 1 负输入选择位 (Comparator 1 input minus selection bits)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它们选择哪个输入连接到比较器 1 的负输入。
000 = 1/4 V_{REFINT}
001 = 1/2 V_{REFINT}
010 = 3/4 V_{REFINT}
011 = V_{REFINT}
100 = 保留
101 = 保留
110 = PA9
111 = 由 INMESEL 位选择的 GPIOx

位 3:2 **PWRMODE[1:0]**：比较器 1 的功耗模式 (Power Mode of the comparator 1)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它们控制比较器 1 的功耗/速度。
00： 高速
01 或 10： 中速
11： 超低功耗

位 1 保留，必须保持清零。

位 0 **EN**：比较器 1 使能位 (Comparator 1 enable bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。可开启比较器 1。
0：关闭比较器 1
1：开启比较器 1

18.6.2 比较器 2 控制和状态寄存器 (COMP2_CSR)

Comparator 2 control and status register

COMP2_CSR 为比较器 2 控制/状态寄存器。此寄存器包含与比较器 2 相关的所有位/标志。

偏移地址：0x04

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	VALUE	Res.	Res.	Res.	INMESEL		Res.	SCAL EN	BRG EN	Res.	BLANKING			HYST	
rs	r				rw			rw	rw		rw			rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLA RITY	Res.	Res.	Res.	Res.	Res.	WIN MODE	INPSEL.		INMSEL			PWRMODE		Res.	EN
rw						rw	rw		rw			rw			rw

位 31 **LOCK**：CSR 寄存器锁定位 (CSR register lock bit)

此位由软件置 1，通过硬件系统复位清零。它将锁定比较器 2 控制寄存器 (COMP2_CSR[31:0]) 的全部内容。

0：比较器 2 的 COMP2_CSR[31:0] 可读/写

1：比较器 2 的 COMP2_CSR[31:0] 只读

位 30 **VALUE**：比较器 2 输出状态位 (Comparator 2 output status bit)

此位为只读。它反映当前比较器 2 输出的状态（受到 POLARITY 位的影响）。

位 29:27 保留，必须保持复位值

位 26:25 **INMESEL**：比较器 2 负输入扩展选择位 (comparator 2 input minus extended selection bits)

这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它们选择哪个扩展 GPIO 输入连接到比较器的负输入（INMSEL = 111 时）。

00：PB7

01：PA2

10：PA4

11：PA5

位 24 保留，必须保持复位值

位 23 **SCALEN**：电压调节器使能位 (Voltage scaler enable bit)

此位由软件置 1 和清零。此位使能比较器 2 负输入上的 V_{REFINT} 分压器输出。

0：禁止带隙调节器（COMP1_CSR 寄存器的 SCALEN 位也复位时）

1：使能带隙调节器

位 22 **BRGEN**：调节器电阻桥使能 (Scaler bridge enable)

此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。此位可使能调节器电阻桥。

0：禁止调节器电阻桥（COMP1_CSR 寄存器的 BRGEN 位也复位时）

1：使能调节器电阻桥

如果 SCALEN 置 1 且 BRGEN 复位，则 BG 参考电压可用，但 1/4 BGAP、1/2 BGAP 和 3/4 BGAP 不可用。发送 BGAP 值，而非 1/4 BGAP、1/2 BGAP、3/4 BGAP。

如果 SCALEN 和 BRGEN 置 1，则 1/4 BGAP、1/2 BGAP、3/4 BGAP 和 BGAP 参考电压可用。

位 21 保留，必须保持复位值

位 20:18 **BLANKING[2:0]**：比较器 2 消隐源选择位 (Comparator 2 blanking source selection bits)
这些位选择哪个定时器输出控制比较器 2 输出消隐。
000：无消隐
001：选择 TIM1 OC5 作为消隐源
010：选择 TIM2 OC3 作为消隐源
所有其他值：保留

位 17:16 **HYST[1:0]**：比较器 2 迟滞选择位 (Comparator 2 hysteresis selection bits)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。选择比较器 2 的迟滞电压。
00：无迟滞
01：低迟滞
10：中等迟滞
11：高迟滞

位 15 **POLARITY**：比较器 2 极性选择位 (Comparator 2 polarity selection bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它使比较器 2 的极性反相。
0：比较器 2 输出值不反相
1：比较器 2 输出值反相

位 14:10 保留，必须保持复位值。

位 9 **WINMODE**：窗口模式选择位 (Windows mode selection bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。此位选择比较器的窗口模式。如果此位置 1，则比较器的两个正输入将连接在一起。
0：比较器 2 的正输入不连接到比较器 1
1：比较器 2 的正输入连接到比较器 1 的正输入

位 8:7 **INPSEL**：比较器 2 正输入选择位 (Comparator 2 input plus selection bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。
00：PB4
01：PB6
10：PA3
11：保留

位 6:4 **INMSEL**：比较器 2 负输入选择位 (Comparator 2 input minus selection bits)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它们选择哪个输入连接到比较器 2 的负输入。
000 = 1/4 V_{REFINT}
001 = 1/2 V_{REFINT}
010 = 3/4 V_{REFINT}
011 = V_{REFINT}
100 = 保留
101 = 保留
110 = PB3
111 = INMESEL 位选择的 GPIOx

位 3:2 **PWRMODE[1:0]**：比较器 2 的功耗模式 (Power Mode of the comparator 2)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它们控制比较器 2 的功耗/速度。
00： 高速
01 或 10： 中速
11： 超低功耗

位 1 保留，必须保持清零。

位 0 **EN**：比较器 2 使能位 (Comparator 2 enable bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。可开启比较器 2。
0：关闭比较器 2
1：开启比较器 2

18.6.3 COMP 寄存器映射

下表对比较器寄存器进行了汇总。

表 92. COMP 寄存器映射和复位值

偏移	寄存器	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
0x00	**COMP1_CSR**	LOCK	VALUE	Res.	Res.	Res.	INMESEL		Res.	SCALEN	BRGEN.	Res.	BLANKING			HYST		POLARITY.	Res.	Res.	Res.	Res.	Res.	Res.	INPSEL		INMSEL			PWRMODE		Res.	EN
	Reset value	0	0				0	0		0	0		0	0	0	0	0	0							0	0	0	0	0	0	0		0
0x04	**COMP2_CSR**	LOCK	VALUE	Res.	Res.	Res.	INMESEL		Res.	SCALEN	BRGEN	Res.	BLANKING			HYST		POLARITY.	Res.	Res.	Res.	Res.	Res.	WINMODE	INPSEL		INMSEL			PWRMODE		Res.	EN
	Reset value	0	0				0	0		0	0		0	0	0	0	0	0						0	0	0	0	0	0	0	0		0

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

19 液晶显示控制器 (LCD)

19.1 简介

LCD 控制器是一款适用于单色被动式液晶显示器 (LCD) 的数字控制器/驱动器，最多具有 8 个公用端子和 44 个区段端子，用以驱动 176 (44x4) 个或 320 (40x8) 个 LCD 图像元素（像素）。端子的确切数量取决于数据手册中所述的器件引脚。

LCD 由若干区段（像素或完整符号）组成，这些区段均可点亮或熄灭。每个区段都包含一层在两根电极之间对齐的液晶分子。当向液晶施加高于阈值电压的电压时，相应的区段显现。区段电压必须为交流，以避免液晶中出现电泳效应（这将影响显示效果）。之后，必须在区段两端生成波形以避免出现直流 (DC)。

词汇表

公用：连接到多个区段（44 个区段）的电气连接端子。

LCD：（液晶显示器）无源显示面板，带有直接引向区段的端子。

偏置：驱动 LCD 时使用的电压等级。它定义为 1/(用于驱动 LCD 显示器的电压等级数 - 1)。

区段：最小可视单元（一个条或点，用于帮助在 LCD 显示器上创建字符）。

升压电路：对比度控制器电路

占空比：定义为 1 /(给定 LCD 显示器上的公用端子数) 的数字。

帧：写入区段的波形的一个周期。

帧速率：每秒帧数，即，每秒激励 LCD 区段的次数。

19.2 LCD 主要特性

- 高度灵活的帧速率控制。
- 支持静态、1/2、1/3、1/4 和 1/8 占空比。
- 支持静态、1/2、1/3 和 1/4 偏置。
- 双缓冲存储器，允许通过应用固件随时更新 LCD_RAM 寄存器中的数据，而不影响所显示数据的完整性。
  - 多达 16 × 32 位寄存器的 LCD 数据 RAM，其中包含像素信息（激活/未激活）。
- 可通过软件选择的 LCD 输出电压（对比度）：V_{LCDmin} 到 V_{LCDmax}。
- 无需外部模拟元件：
  - 内置的升压转换器可生成高于 V_{DD} 的内部 V_{LCD} 电压。
  - 可通过软件在外部和内部 V_{LCD} 电压源之间进行选择。如果选择外部电压源，将禁止内部升压电路以降低功耗。
  - 内置的电阻网络可生成中间 V_{LCD} 电压。
  - 可通过软件配置电阻网络的结构以调整功耗，从而匹配 LCD 面板所需的电容充电。
  - 集成电压输出缓冲器可实现更高的 LCD 驱动能力。
- 可以通过两种不同的方法调整对比度：
  - 当使用内部升压转换器时，软件可以在 V_{LCDmin} 和 V_{LCDmax} 之间调整 V_{LCD}。
  - 可编程的帧间死区（最多 8 个相位周期）。
- 完全支持低功耗模式：LCD 控制器可在睡眠、低功率运行、低功耗睡眠和停止模式下进行显示，也可以完全禁用以降低功耗。
- 内置反相功能以降低功耗和 EMI（电磁干扰）。
- 帧起始中断，在更新 LCD 数据 RAM 时同步软件。
- 闪烁功能：
  - 可以设置 1、2、3、4、8 个或所有像素以可配置的频率闪烁。
  - 可通过软件将闪烁频率调整到大约 0.5 Hz、1 Hz、2 Hz 或 4 Hz。
- 使用的 LCD 区段和公共引脚应配置为 GPIO 复用功能，未使用的区段和公共引脚可用于通用 I/O 或其他外设复用功能。

注： *当 LCD 依赖于内部升压转换器时，VLCD 引脚应通过一个电容连接到 V_{SS}。电容典型值为 1 µF（有关详细信息，请参见产品数据手册中的 C_{EXT} 值）。*

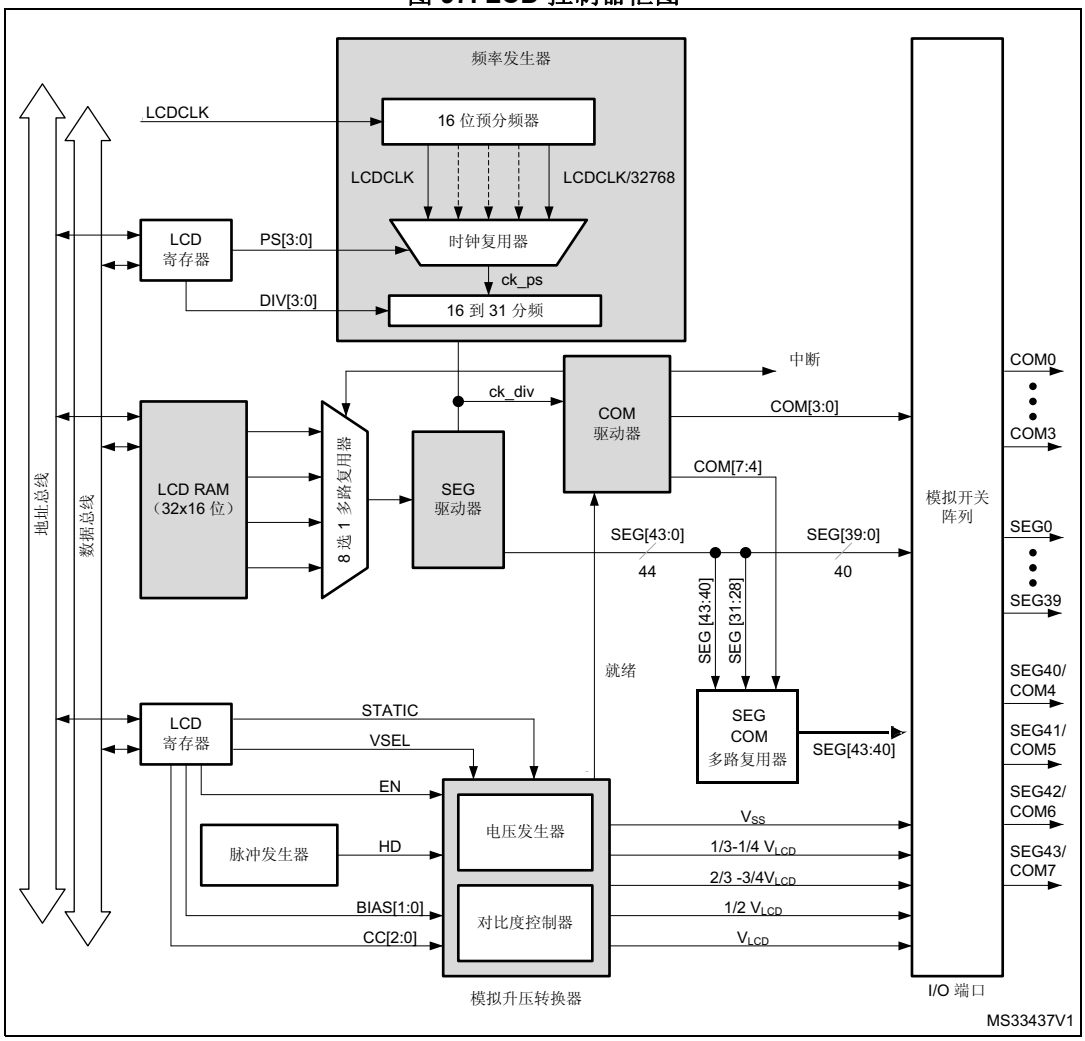
注： *如果不使用 LCD 外设，则 VLCD 引脚应连接到 V_{DDA}。*

19.3 LCD 功能说明

19.3.1 概述

LCD 控制器具有五个主要模块（请参见*图 97*）：

图 97. LCD 控制器框图



注： *LCDCLK 与 RTCCLK 相同。请参考本手册的 RCC 部分中的 RTC/LCD 时钟说明。*

凭借频率发生器，可以通过 LCD 输入时钟频率（LCDCLK，可从 32 kHz 直至 1 MHz）获得各种 LCD 帧速率。

可以使用 3 种不同的时钟源来提供 LCD 时钟 (LCDCLK/RTCCLK)：

- 32 kHz 低速外部 RC (LSE)
- 32 kHz 低速内部 RC (LSI)
- 32 分频的高速外部时钟 (HSE)

19.3.2 频率发生器

此时钟源必须稳定才能获得精确的 LCD 时序并因此最大限度地减少 LCD 区段间的 DC 电压偏移。输入时钟 LCDCLK 可被从 1 到 2^{15}x 31 的任意值分频（请参见*第 525 页的第 19.6.2 节：LCD 帧控制寄存器 (LCD_FCR)*）。频率发生器包含一个预分频器（16 位纹波计数器）和一个 16 到 31 时钟分频器。LCD_FCR 寄存器中的 PS[3:0] 位选择被 $2^{PS[3:0]}$ 分频的 LCDCLK。如果需要更精细的分辨率，LCD_FCR 寄存器中的 DIV[3:0] 位可用于进一步对时钟进行 16 到 31 分频。通过这种方式可以粗调频率，然后通过计数器线性调整时钟以进行微调。频率发生器模块的输出是 f_{ck_div}，它构成整个 LCD 控制器的时基。ck_div 频率与 LCD 相位频率相等，而不等于帧频率（它们仅在静态占空比的情况下相等）。通过将 f_{ck_div} 除以激活的公用端子数（或者将该频率与占空比相乘），可获得帧频率 (f_{frame})。因此，频率发生器的输入时钟频率 (f_{LCDCLK}) 与其输出时钟频率 f_{ck_div} 之间的关系为：

$$f_{ckdiv} = \frac{f_{LCDCLK}}{2^{PS} \times (16 + DIV)}$$

$$f_{frame} = f_{ckdiv} \times 占空比$$

这使得频率发生器非常灵活。*表 93* 中给出了帧速率计算示例。

表 93. 帧速率计算示例

LCDCLK	PS[3:0]	DIV[3:0]	比值	占空比	f_{frame}
32.768 kHz	3	1	136	1/8	30.12 Hz
32.768 kHz	4	1	272	1/4	30.12 Hz
32.768 kHz	4	6	352	1/3	31.03 Hz
32.768 kHz	5	1	544	1/2	30.12 Hz
32.768 kHz	6	1	1088	静态	30.12 Hz
32.768 kHz	1	4	40	1/8	102.40 Hz
32.768 kHz	2	4	80	1/4	102.40 Hz
32.768 kHz	2	11	108	1/3	101.14 Hz
32.768 kHz	3	4	160	1/2	102.40 Hz
32.768 kHz	4	4	320	静态	102.40 Hz
1.00 MHz	6	3	1216	1/8	102.80 Hz
1.00 MHz	7	3	2432	1/4	102.80 Hz
1.00 MHz	7	10	3328	1/3	100.16 Hz
1.00 MHz	8	3	4864	1/2	102.80 Hz
1.00 MHz	9	3	9728	静态	102.80 Hz

选择的帧频率必须在大约 30 Hz 到 100 Hz 的范围内，并在功耗和可接受的刷新速率之间进行平衡。此外，可通过一个专用闪烁预分频器选择闪烁频率。此频率定义为：

$f_{BLINK} = f_{ck\_div}/2^{(BLINKF + 3)}$

其中，BLINKF[2:0] = 0、1、2、... 、7

可实现的闪烁频率为 0.5 Hz、1 Hz、2 Hz 或 4 Hz。

19.3.3 公用驱动器

公用信号由公用驱动器模块产生（请参见*图 97*）。

COM 信号偏置

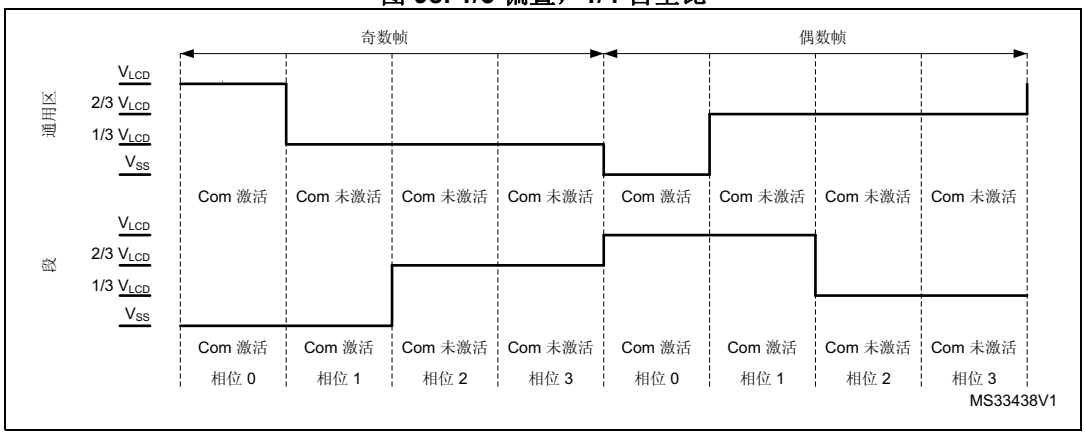
各个 COM 信号的波形相同，但相位不同。这些信号仅在周期的相应相位才具有最大振幅 V_{LCD} 或 V_{SS}，而在其他相位的信号幅值为：

- 1/4 V_{LCD} 或 3/4 V_{LCD}（1/4 偏置时）
- 1/3 V_{LCD} 或 2/3 V_{LCD}（1/3 偏置时）
- 以及 1/2 V_{LCD}（1/2 偏置时）

可以通过 LCD_CR 寄存器中的 BIAS 位在 1/2、1/3 和 1/4 偏置模式之间进行选择。

当某个像素对应的公用线和区段线在相同相位期间激活时，该像素激活，这意味着在此相位期间公用线和区段线之间的电压差最大。对公用信号进行反相以减少 EMI。如*图 98* 所示，通过反相，每个奇数周期结束时的平均电压为 1/2 V_{LCD}。

图 98. 1/3 偏置，1/4 占空比



在 1/2 偏置 (BIAS = 01) 的情况下，VLCD 引脚在奇数帧和偶数帧的节点 b 上产生的中间电压等于 1/2 V_{LCD}（请参见*图 101*）。

COM 信号占空比

根据 LCD_CR 寄存器中的 DUTY[2:0] 位，使用以下占空比产生 COM 信号：静态占空比（见*图 100*）、1/2 占空比（见*图 101*）、1/3 占空比（见*图 102*）、1/4 占空比（见*图 103*）或 1/8 占空比（见*图 104*）。

奇数帧的相位 *n* 期间 COM[*n*] *n*[0 到 7] 激活，因此 COM 引脚被驱动到 V_{LCD}。

偶数帧的相位 *n* 期间 COM 引脚被驱动到 V_{SS}。

在 1/3 或 1/4 偏置的情况下：

- 在 n 以外的相位期间 COM[*n*] 未激活，因此 COM 引脚被驱动到 1/3 (1/4) V_{LCD}（奇数帧期间）和 2/3 (3/4) V_{LCD}（偶数帧期间）

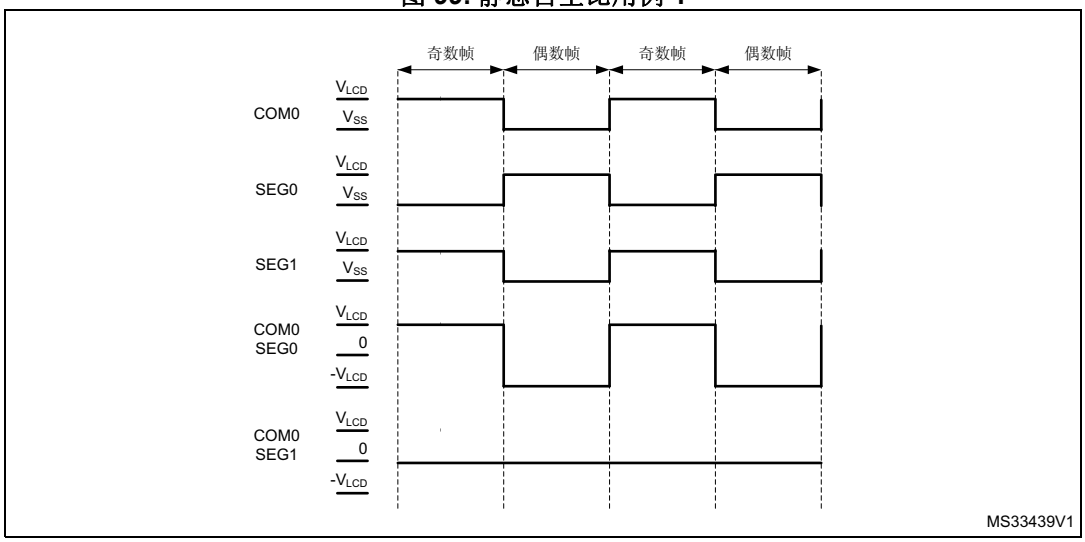
在 1/2 偏置的情况下：

- 如果在 n 以外的相位期间 COM[*n*] 未激活，COM 引脚始终驱动到 1/2 V_{LCD}（奇数帧和偶数帧）。

当选择静态占空比时，区段线不多路复用，这意味着每个区段输出对应于一个像素。通过这种方式，最多只能驱动 44 个像素。未使用 COM[7:1] 时，COM[0] 始终激活且被驱动到 V_{SS}。

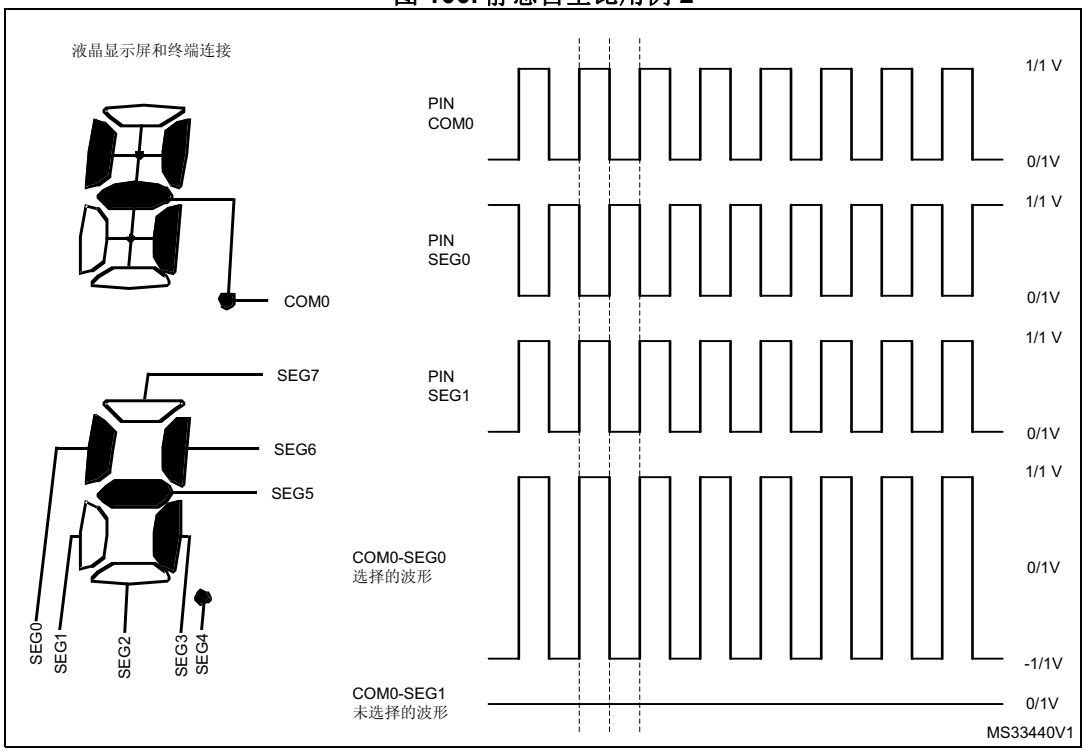
当 LCD_CR 寄存器中的 LCDEN 位复位时，所有公用线均下拉到 V_{SS}，并且 LCD_SR 寄存器中的 ENS 标志变为 0。静态占空比意味着 COM[0] 始终激活，并且只有两个电压电平用于区段和公用线：V_{LCD} 和 V_{SS}。如果像素对应的 SEG 线的电压与 COM 线的电压相反，则像素激活；如果电压相等，则像素未激活。通过这种方式，LCD 可获得最大对比度（请参见*图 99*、*图 100*）。在*图 99* 中，像素 0 激活而像素 1 未激活。

图 99. 静态占空比用例 1



每个帧中只有一个相位，因此 f_{frame} 等于 f_{LCD}。如果选择 1/4 占空比，一个帧中有四个相位，其中，COM[0] 在相位 0 期间激活，COM[1] 在相位 1 期间激活，COM[2] 在相位 2 期间激活，COM[3] 在相位 3 期间激活。

图 100. 静态占空比用例 2

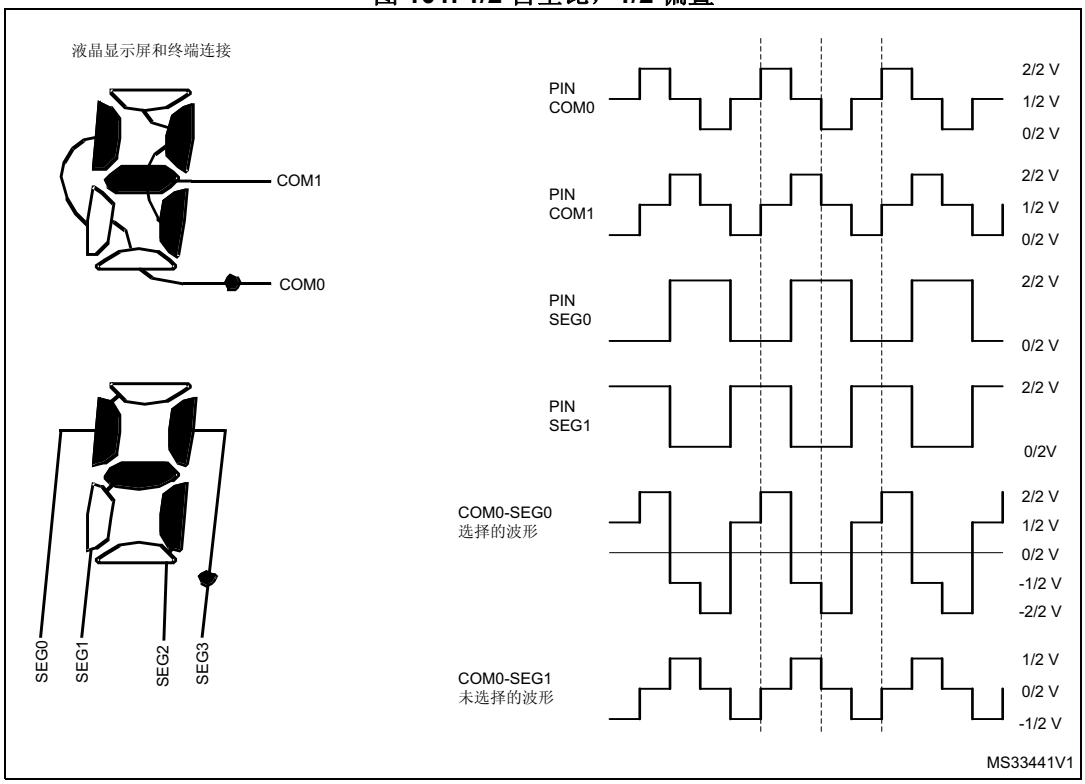


在此模式下，区段端子多路复用并且每个端子控制四个像素。仅当像素对应的 SEG 和 COM 线在相同相位期间激活时，该像素才激活。在 1/4 占空比的情况下，要禁用连接到 COM[0] 的像素 0，当 COM[0] 激活时，SEG[0] 需要在相位 0 期间处于未激活状态。要激活连接到 COM[1] 的像素 0，当 COM[1] 激活时，SEG[0] 需要在相位 1 期间处于激活状态（请参见*图 103*）。要激活连接到 COM[0] 的像素 0 到 43，当 COM[0] 激活时，SEG[0:43] 需要在相位 0 期间处于激活状态。这些注意事项同样适用于其他像素。

8 选 1 多路复用器

当 COM[0] 激活时，公用驱动器模块也将驱动*图 97* 中所示的 8 选 1 多路复用器，以便选择前两个 RAM 寄存器存储单元的内容。当 *COM[7]* 激活时，8 选 1 多路复用器的输出是最后两个 RAM 存储单元的内容。

图 101. 1/2 占空比，1/2 偏置



19.3.4 区段驱动器

区段驱动器模块根据像素数据（来自各相位中由公用驱动器模块驱动的 8 选 1 多路复用器）来控制 SEG 线。

在 1/4 或 1/8 占空比的情况下

当 COM[0] 激活时，与连接到 COM[0]（前两个 LCD_RAM 存储单元的内容）的像素相关的像素信息（激活/未激活）通过 8 选 1 多路复用器。

在奇数帧的相位 0 期间，SEG[*n*] 引脚 *n* [0 到 43] 被驱动到 V_{SS}（指示当 COM[0] 激活时像素 *n* 激活）。

在偶数帧的相位 0 期间，SEG[*n*] 引脚被驱动到 V_{LCD}。如果像素 *n* 未激活，则 SEG[*n*] 引脚被驱动到 2/3 (2/4) V_{LCD}（奇数帧期间）或 1/3 (2/4) V_{LCD}（偶数帧期间）（V_{LCD} 当前引脚电压的反转）（请参见*图 98*）。

在 1/2 偏置的情况下，如果像素未激活，SEG[*n*] 引脚被驱动到 V_{LCD}（奇数帧期间）和 V_{SS}（偶数帧期间）。

当禁止 LCD 控制器（LCD_CR 寄存器中的 LCDEN 位清零）时，SEG 线下拉到 V_{SS}。

图 102. 1/3 占空比，1/3 偏置

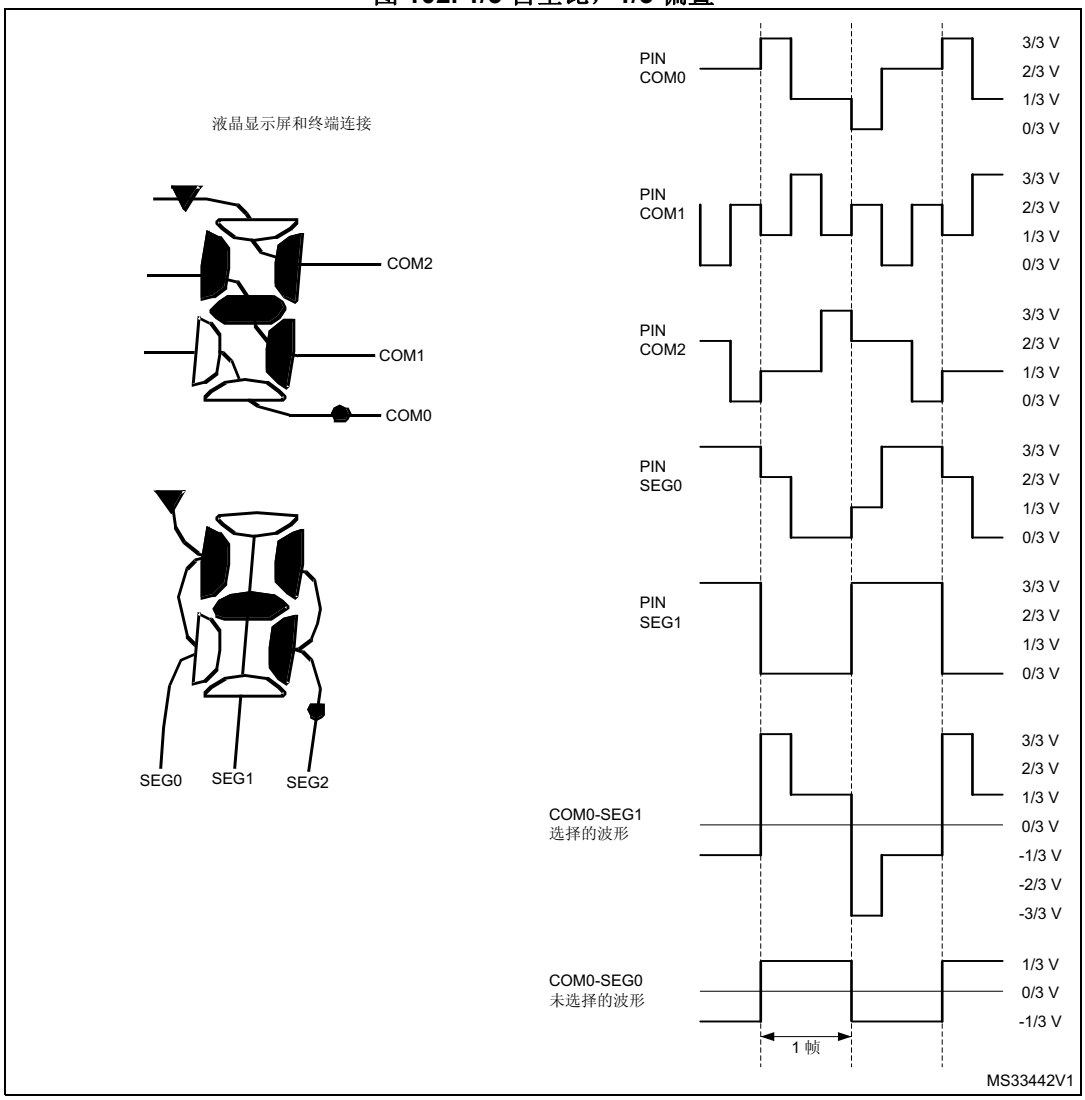


图 103. 1/4 占空比，1/3 偏置

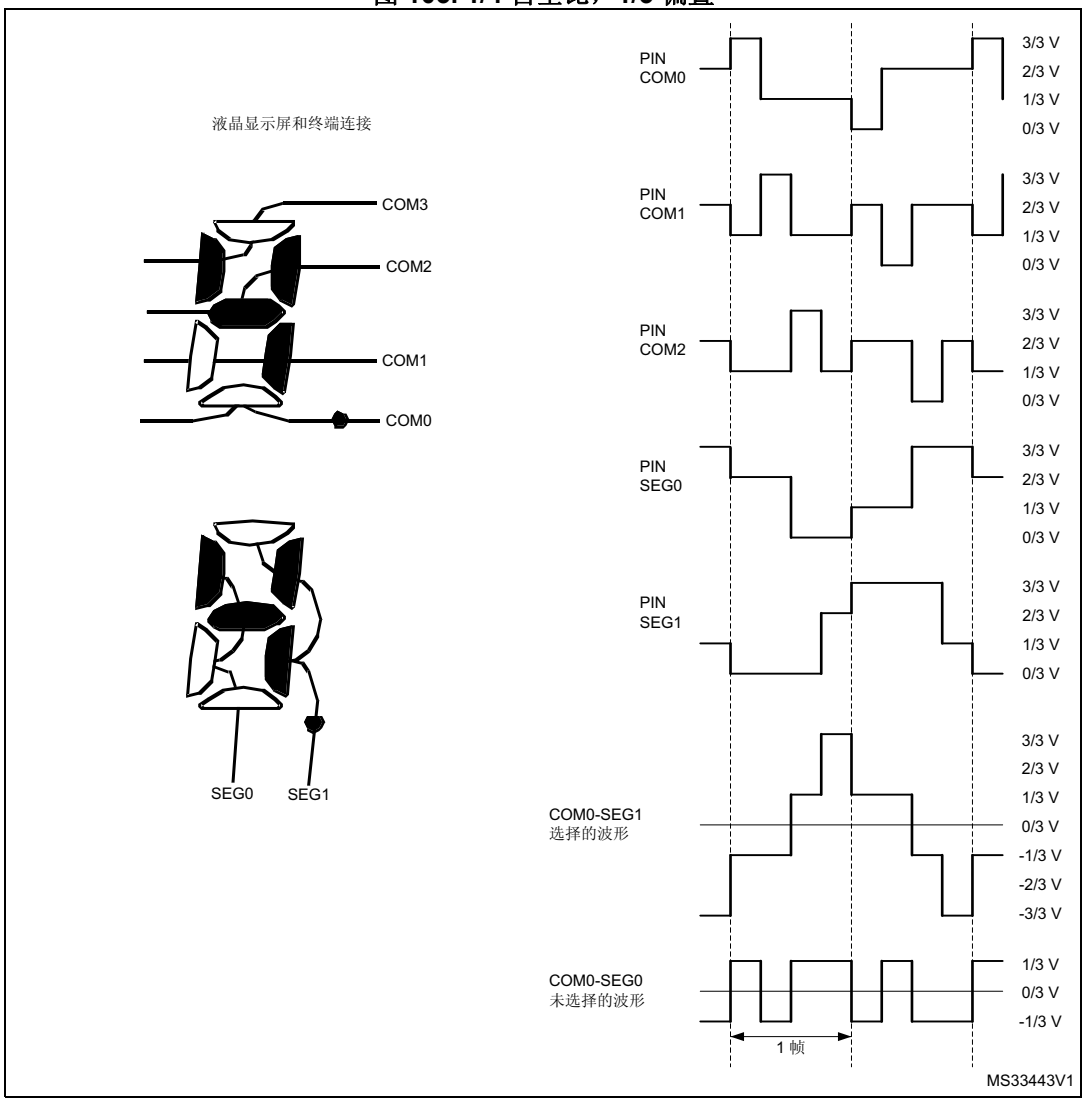
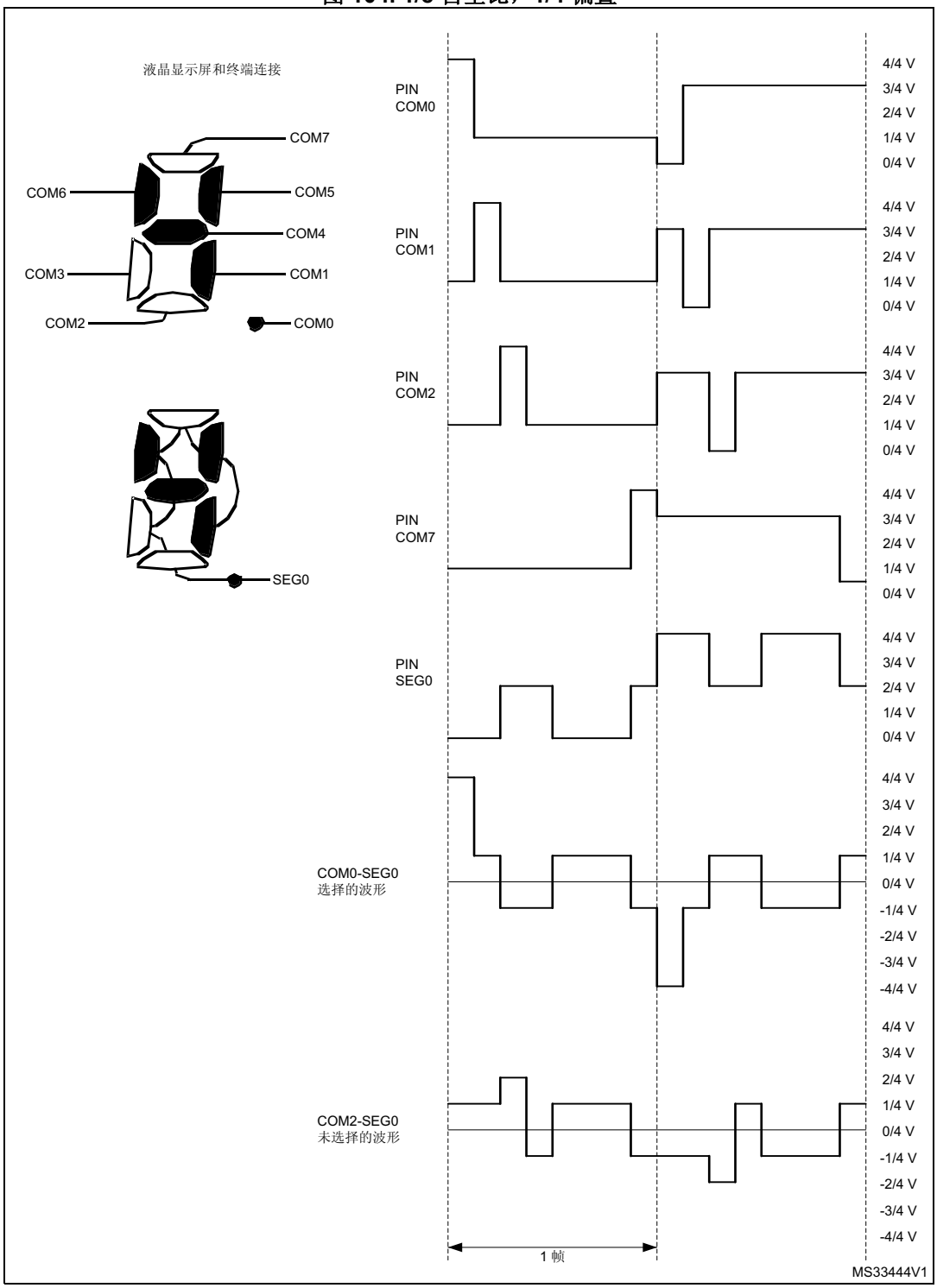


图 104. 1/8 占空比，1/4 偏置



闪烁

区段驱动器还实现了一个可编程的闪烁功能，允许一些像素以特定频率连续地点亮。可通过 LCD_FCR 寄存器中的 BLINK[1:0] 位配置闪烁模式，从而使 1、2、4、8 或所有像素能够同时闪烁（请参见*第 19.6.2 节：LCD 帧控制寄存器 (LCD_FCR)*）。可使用 LCD_FCR 寄存器中 BLINKF[2:0] 位从八个不同值中选择闪烁频率。

表 94 给出了不同闪烁频率的示例（以 ck_div 频率的函数形式）。

表 94. 闪烁频率

BLINKF[2:0] 位			ck_div 频率（LCDCLK 频率为 32.768 kHz）			
			32 Hz	64 Hz	128 Hz	256 Hz
0	0	0	4.0 Hz	N/A	N/A	N/A
0	0	1	2.0 Hz	4.0 Hz	N/A	N/A
0	1	0	1.0 Hz	2.0 Hz	4.0 Hz	N/A
0	1	1	0.5 Hz	1.0 Hz	2.0 Hz	4.0 Hz
1	0	0	0.25 Hz	0.5 Hz	1.0 Hz	2.0 Hz
1	0	1	N/A	0.25 Hz	0.5 Hz	1.0 Hz
1	1	0	N/A	N/A	0.25 Hz	0.5 Hz
1	1	1	N/A	N/A	N/A	0.25 Hz

19.3.5 电压发生器和对比度控制

LCD 电源

LCD 电源可能来自内部升压转换器，或来自施加在 VLCD 引脚上的外部电压。可以使用 LCD_CR 寄存器中的 VSEL 位选择内部或外部电压源。如果选择外部电压源，将禁止内部升压电路（升压转换器）以降低功耗。

当选择升压转换器作为 V_{LCD} 源时，可以通过 LCD_FCR（请参见*第 19.6.2 节*）寄存器中的位 CC[2:0]（对比度控制）在多个值（从 V_{LCDmin} 到 V_{LCDmax}）中选择 V_{LCD} 值。V_{LCD} 的新值在每次新帧开始时生效。

当选择外部电源作为 V_{LCD} 源时，必须在 V_{LCDmin} 到 V_{LCDmax} 的范围内选择 V_{LCD} 电压（请参考数据手册）。然后可以通过编程帧间死区来控制对比度（请参见*第 515 页的死区*）。

必须执行特定的软件序列，根据要使用的 LCD 电源来配置 LCD。这里我们认为 LCD 控制器在配置序列之前处于禁止状态。

使用内部升压转换器时（需要在 VLCD 引脚上使用电容 C_{EXT}）：

- 在 GPIO_AFR 寄存器中将 VLCD 引脚配置为复用功能 LCD
- 等待外部电容 C_{EXT} 充电（C_{EXT} 连接到 VLCD 引脚，C_{EXT} = 1 µF 时大约需要 2 ms）
- 通过复位 LCD_CR 寄存器中的 VSEL 位将电压源设置为内部源
- 通过将 LCD_CR 寄存器中的 LCDEN 位置 1 来使能 LCD 控制器

使用 LCD 外部电源时：

- 通过将 LCD_CR 寄存器中的 VSEL 位置 1 将电压源设置为外部源
- 在 GPIO_AFR 寄存器中将 VLCD 引脚配置为复用功能 LCD
- 通过将 LCD_CR 寄存器中的 LCDEN 位置 1 来使能 LCD 控制器

LCD 中间电压

LCD 中间电压通过内部电阻分压器网络生成，如*图 105* 所示。

LCD 电压发生器发出介于 V_{SS} 和 V_{LCD} 之间的中间电压：

- 1/3 V_{LCD} 和 2/3 V_{LCD}（1/3 偏置时）
- 1/4 V_{LCD}、2/4 V_{LCD} 和 3/4 V_{LCD}（1/4 偏置时）
- 仅 1/2 V_{LCD}（1/2 偏置时）

LCD 驱动器选择

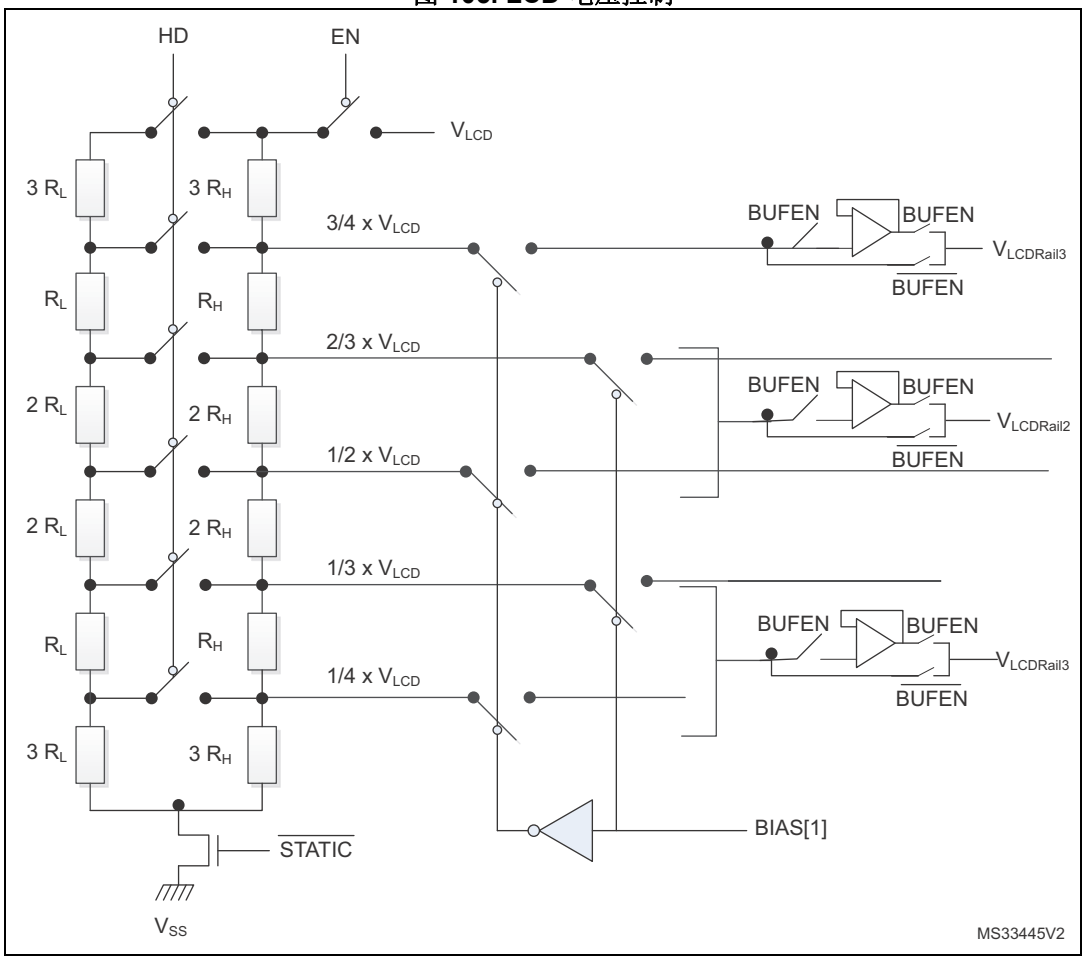
两个电阻网络（一个具有低值电阻 (R_L)，一个具有高值电阻 (R_H)）分别用于在转换过程中增加电流和在静态状态下降低功耗。

EN 开关遵循下述规则（请参见*图 105*）：

- 如果 LCD_CR 寄存器中的 LCDEN 位置 1，EN 开关闭合。
- 当 LCD_CR 寄存器中的 LCDEN 位清零时，EN 开关在偶数帧结束时打开，以避免中间电压不为 0（考虑整个奇数帧加偶数帧）。

LCD_FCR 寄存器中的 PON[2:0]（脉冲打开持续时间）位在 COMMON 线和 SEGMENT 线的电平变化时配置通过 HD（高驱动）接通 R_L 的时间（请参见*图 105*）。较短的驱动时间有助于降低功耗，但内部电阻较高的显示器可能需要更长的驱动时间才能达到令人满意的对比度。

图 105. LCD 电压控制



1. R_{LN} 和 R_{HN} 分别是低值电阻网络和高值电阻网络。

可通过 LCD_FCR 配置寄存器中的 HD 位随时接通 R_{LN} 分频器（请参见*第 19.6.2 节*）。

HD 开关遵循下述规则：

- 如果 LCD_FCR 寄存器中的 HD 位和 PON[2:0] 位复位，则 HD 开关打开。
- 如果 LCD_FCR 寄存器中的 HD 位复位，并且 LCD_FCR 中的 PON[2:0] 位不为 00，则在 PON[2:0] 位中定义的脉冲数期间 HD 开关闭合。
- 如果 LCD_FCR 寄存器中的 HD 位为 1，则 HD 开关常闭。

缓冲模式

当通过将 LCD_CR 寄存器中的 BUFEN 位置 1 使能电压输出缓冲器时，LCD 驱动能力将得到提高，因为缓冲器可防止 LCD 容性负载为电阻桥增加难以接受的负荷以及干扰其电压产生。因此，我们可获得更稳定的中间电压值，从而改善施加到 LCD 像素的 RMS 电压。

在缓冲模式中，高阻值电阻桥 R_{HN} 会产生中间电压来降低功耗，低阻值电阻桥 R_{LN} 会自动禁止，而与 HD 位或 PON 位的配置无关。

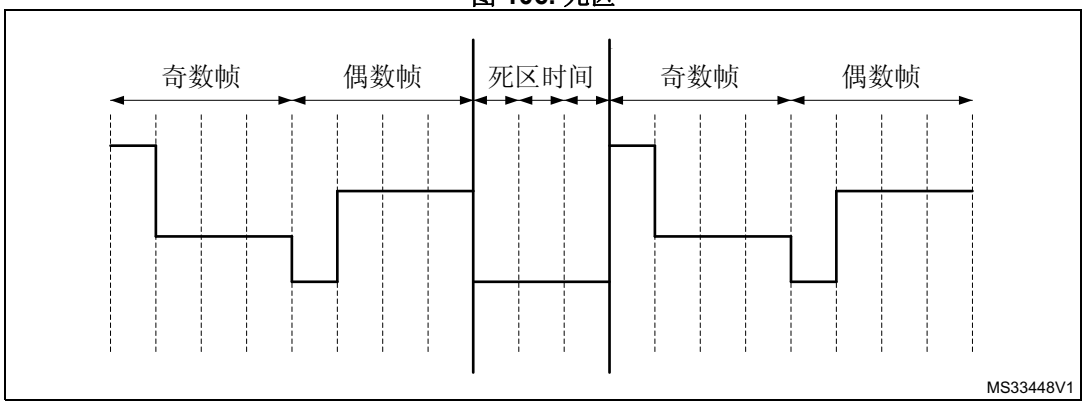
缓冲器可以独立于 V_{LCD} 电源（内部或外部）使用，但只能在未激活 LCD 控制器时使能或禁止。

LCDEN 位激活后，LCD_SR 寄存器中的 RDY 位置 1，以指示电压电平稳定并且 LCD 控制器可以开始工作。

死区

除了使用 CC[2:0] 位外，还可以通过编程帧间死区来控制对比度。在死区内，COM 和 SEG 值置于 V_{SS}。LCD_FCR 寄存器中的 DEAD[2:0] 位可用于编程一段最长为八个相位周期的时间。此死区可降低对比度，而无需修改帧速率。

图 106. 死区



19.3.6 双缓冲存储器

使用此双缓冲存储器，LCD 控制器可保证所显示信息的一致性，而无需使用中断来控制 LCD_RAM 修改。

应用软件可以通过 APB 接口访问一缓冲级 (LCD_RAM)。一旦它修改了 LCD_RAM，便会将 LCD_SR 寄存器中的 UDR 标志置 1。该 UDR 标志（更新显示请求）会请求将更新的信息移到二缓冲级 (LCD_DISPLAY)。

该操作与帧同步进行（在下一帧开始时），直到更新完成，LCD_RAM 处于写保护状态并且 UDR 标志一直保持高电平状态。一旦更新完成，另一个标志（UDD - 更新显示完成）便会置 1 并产生一个中断（如果 LCD_FCR 寄存器中的 UDDIE 位置 1）。

在最坏的情况下，更新 LCD_DISPLAY 所花费的时间为一个奇数帧和一个偶数帧。

使能显示器之前 (LCDEN = 1)，不会发生更新（UDR = 1 且 UDD = 0）。

19.3.7 COM 和 SEG 多路复用

输出引脚与占空比模式

输出引脚包括：

- SEG[43:0]
- COM[3:0]

根据占空比配置的不同，COM 和 SEG 输出引脚会具有不同的功能：

- 在静态、1/2、1/3 和 1/4 占空比模式下，有多达 44 个 SEG 引脚，并分别有 1、2、3 和 4 个 COM 引脚。
- 在 1/8 占空比模式 (DUTY[2:0] = 100) 下，在 SEG[43:40] 引脚上提供 COM[7:4] 输出，可用区段数减少至 40。

小型封装的重新映射功能

此外，可以通过将 LCD_CR 寄存器中 MUX_SEG 位置 1 重新映射 4 个区段。使用外部引脚较少的封装器件时，该功能特别有用。当 MUX_SEG 位置 1 时，输出引脚 SEG[43:40] 与 SEG[31:28] 具有相同功能。

该功能仅当未选择 1/8 占空比模式时才可用。

关于该功能的可用性，请参见产品数据手册的引脚分配部分。

对于小型封装，需检查 SEG/COM 复用引脚的可用性，具体如下：

LCD_SEG[n-1]/LCD_COM7/LCD_SEG[31]

LCD_SEG[n-2]/LCD_COM6/LCD_SEG[30]

LCD_SEG[n-3]/LCD_COM5/LCD_SEG[29]

LCD_SEG[n-4]/LCD_COM4/LCD_SEG[28]

n = 小型封装的区段数。

COM 和 SEG 功能与占空比和重映射的关系汇总

表 95 给出了多路复用 COM 和 SEG 功能的所有可能方式。*图 107* 给出了与外部引脚的信号连接示例。

表 95. 重映射功能

DUTY	MUX_SEG	WLCSP100	VFQFPN68	UFQFPN48	输出引脚	功能
1/8	n.a.	40x8	-	-	COM[3:0]	COM[3:0]
					SEG[43:40]/SEG[31:28]/COM[7:4]	COM[7:4]
					SEG[39:0]	SEG[39:0]
1/4	0	44x4	-	-	COM[3:0]	COM[3:0]
					SEG[43:40]/SEG[31:28]/COM[7:4]	SEG[43:40]
					SEG[39:0]	SEG[39:0]
	1	40x4	-	-	COM[3:0]	COM[3:0]
					SEG[43:40]/SEG[31:28]/COM[7:4]	SEG[31:28]
					SEG[39:32]	SEG[39:32]
					SEG[31:28]	未使用
					SEG[27:0]	SEG[27:0]
	0	-	28x4	-	COM[3:0]	COM[3:0]
					SEG[42:40]/SEG[30:28]/COM[6:4]	SEG[42:40]
					SEG[24:0]	SEG[24:0]
	1	-	28x4	-	COM[3:0]	COM[3:0]
					SEG[42:40]/SEG[30:28]/COM[6:4]	SEG[30:28]
					SEG[24:0]	SEG[24:0]
	n.a.	-	-	13x4	COM[3:0]	COM[3:0]
					SEG[21]	SEG[21]
					SEG[17:16]	SEG[17:16]
					SEG[9:0]	SEG[9:0]

表 95. 重映射功能（续）

DUTY	MUX_SEG	WLCSP100	VFQFPN68	UFQFPN48	输出引脚	功能
1/3	0	44x3	-	-	COM[3]	未使用
					COM[2:0]	COM[2:0]
					SEG[43:40]/SEG[31:28]/COM[7:4]	SEG[43:40]
					SEG[39:0]	SEG[39:0]
	1	40x3	-	-	COM[3]	未使用
					COM[2:0]	COM[2:0]
					SEG[43:40]/SEG[31:28]/COM[7:4]	SEG[31:28]
					SEG[39:32]	SEG[39:32]
					SEG[31:28]	未使用
					SEG[27:0]	SEG[27:0]
	0	-	28x3	-	COM[3]	未使用
					COM[2:0]	COM[2:0]
					SEG[42:40]/SEG[30:28]/COM[6:4]	SEG[42:40]
					SEG[24:0]	SEG[24:0]
	1	-	28x3	-	COM[3]	未使用
					COM[2:0]	COM[2:0]
					SEG[42:40]/SEG[30:28]/COM[6:4]	SEG[30:28]
					SEG[24:0]	SEG[24:0]
	n.a.	-	-	13x3	COM[3]	未使用
					COM[2:0]	COM[2:0]
					SEG[21]	SEG[21]
					SEG[17:16]	SEG[17:16]
					SEG[9:0]	SEG[9:0]

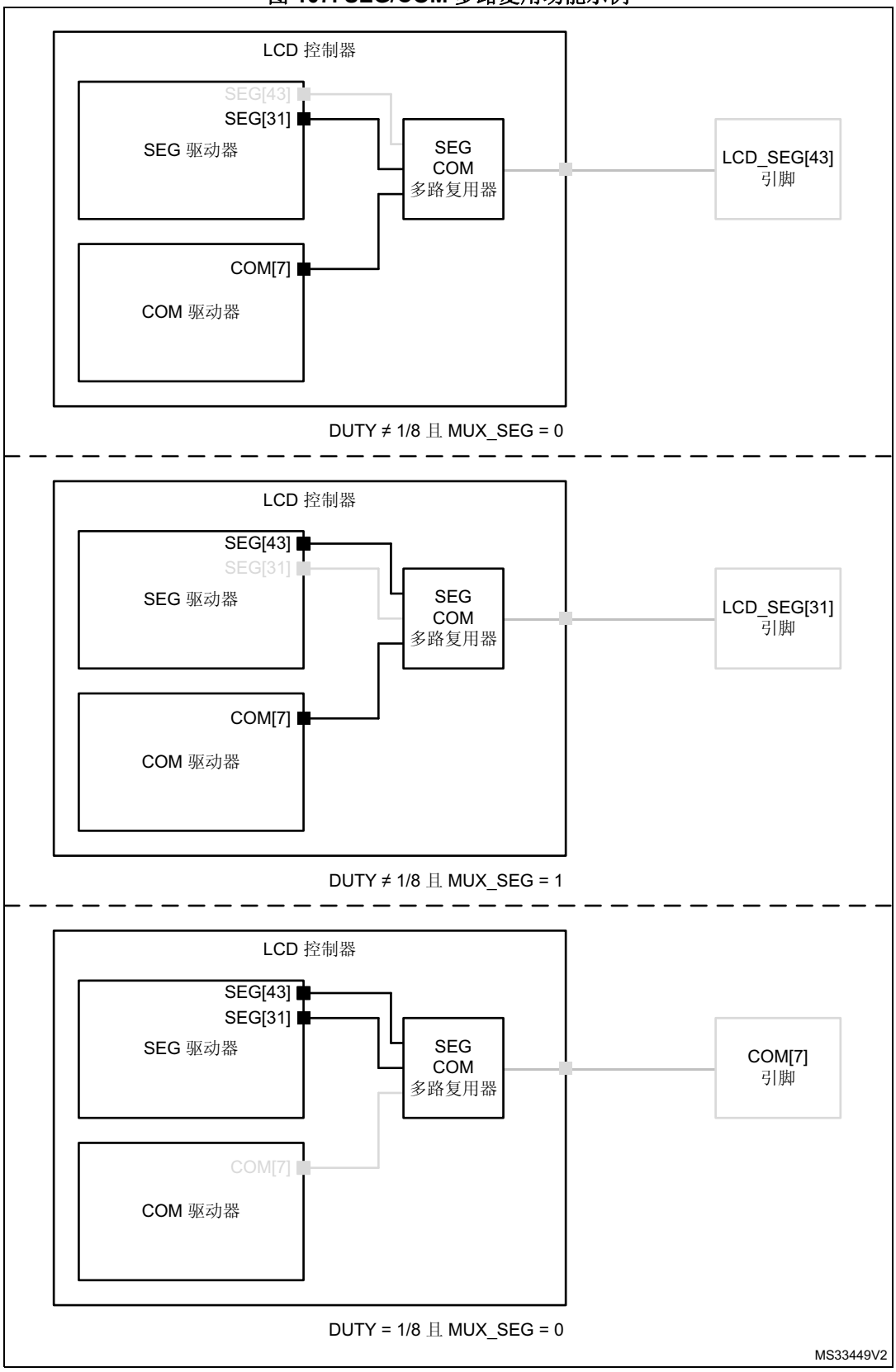
表 95. 重映射功能（续）

DUTY	MUX_SEG	WLCSP100	VFQFPN68	UFQFPN48	输出引脚	功能
1/2	0	44x2	-	-	COM[3:2]	未使用
					COM[1:0]	COM[1:0]
					SEG[43:40]/SEG[31:28]/COM[7:4]	SEG[43:40]
					SEG[39:0]	SEG[39:0]
	1	40x2	-	-	COM[3:2]	未使用
					COM[1:0]	COM[1:0]
					SEG[43:40]/SEG[31:28]/COM[7:4]	SEG[31:28]
					SEG[39:32]	SEG[39:32]
					SEG[31:28]	未使用
					SEG[27:0]	SEG[27:0]
1/2	0	-	28x2	-	COM[3:2]	未使用
					COM[1:0]	COM[1:0]
					SEG[42:40]/SEG[30:28]/COM[6:4]	SEG[42:40]
					SEG[24:0]	SEG[24:0]
	1	-	28x2	-	COM[3:2]	未使用
					COM[1:0]	COM[1:0]
					SEG[42:40]/SEG[30:28]/COM[6:4]	SEG[30:28]
					SEG[24:0]	SEG[24:0]
	n.a.	-	-	13x2	COM[3:2]	未使用
					COM[1:0]	COM[1:0]
					SEG[21]	SEG[21]
					SEG[17:16]	SEG[17:16]
					SEG[9:0]	SEG[9:0]

表 95. 重映射功能（续）

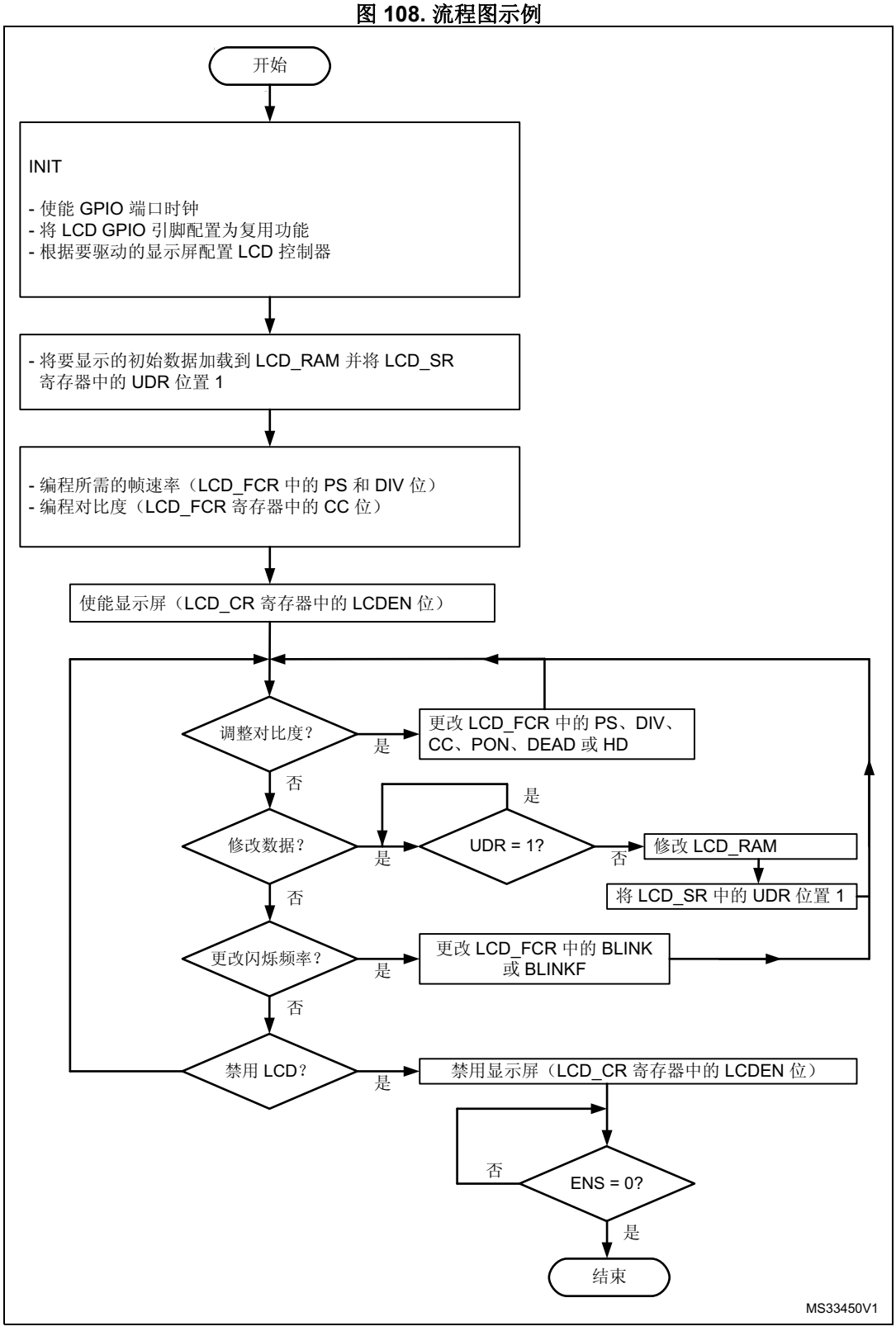
DUTY	MUX_SEG	WLCSP100	VFQFPN68	UFQFPN48	输出引脚	功能
STATIC	0	44x1	-	-	COM[3:1]	未使用
					COM[0]	COM[0]
					SEG[43:40]/SEG[31:28]/COM[7:4]	SEG[43:40]
					SEG[39:0]	SEG[39:0]
	1	40x1	-	-	COM[3:1]	未使用
					COM[0]	COM[0]
					SEG[43:40]/SEG[31:28]/COM[7:4]	SEG[31:28]
					SEG[39:32]	SEG[39:32]
					SEG[31:28]	未使用
					SEG[27:0]	SEG[27:0]
	0	-	28x1	-	COM[3:1]	未使用
					COM[0]	COM[0]
					SEG[42:40]/SEG[30:28]/COM[6:4]	SEG[42:40]
					SEG[24:0]	SEG[24:0]
	1	-	28x1	-	COM[3:1]	未使用
					COM[0]	COM[0]
					SEG[42:40]/SEG[30:28]/COM[6:4]	SEG[30:28]
					SEG[24:0]	SEG[24:0]
STATIC	n.a.	-	-	13x1	COM[3:1]	未使用
					COM[0]	COM[0]
					SEG[21]	SEG[21]
					SEG[17:16]	SEG[17:16]
					SEG[9:0]	SEG[9:0]

图 107. SEG/COM 多路复用功能示例



19.3.8 流程图

图 108. 流程图示例



19.4 LCD 低功耗模式

LCD 控制器可在停止模式下进行显示，也可以完全禁止以降低功耗。

19.5 LCD 中断

下表列出了 LCD 中断请求。

表 96. LCD 中断请求

中断事件	事件标志	事件标志/中断清除方法	中断使能控制位
帧起始 (SOF)	SOF	写入 SOFC = 1	SOFIE
更新显示完成 (UDD)	UDD	写入 UDDC = 1	UDDIE

帧起始 (SOF)

如果 SOFIE（帧起始中断使能）位置 1，则执行 LCD 帧起始中断（请参见*第 19.6.2 节：LCD 帧控制寄存器 (LCD_FCR)*）。执行相应的中断处理向量时在 LCD_CLR 寄存器中将 SOFC 位写为 1，可将 SOF 清除。

更新显示完成 (UDD)

如果 UDDIE（更新显示完成中断使能）位置 1，则执行 LCD 更新显示中断（请参见*第 19.6.2 节：LCD 帧控制寄存器 (LCD_FCR)*）。执行相应的中断处理向量时在 LCD_CLR 寄存器中将 UDDC 位写为 1，可将 UDD 清除。

根据产品实现的不同，上述所有中断事件既可以共享同一个中断向量（LCD 全局中断），也可以分组到 2 个不同的中断向量上（LCD SOF 中断和 LCD UDD 中断）。有关详细信息，请参见*表 57：STM32WB55xx CPU1 向量表*。

要使能 LCD 中断，需按照以下顺序操作：

1. 配置 NVIC 中的 LCD IRQ 通道并将其使能
2. 配置 LCD 以生成中断

19.6 LCD 寄存器

外设寄存器必须按字（32 位）进行访问。

19.6.1 LCD 控制寄存器 (LCD_CR)

LCD control register

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUFEN	MUX_SEG	BIAS[1:0]		DUTY[2:0]			VSEL	LCDEN
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:9 保留，必须保持复位值

位 8 **BUFEN**：电压输出缓冲器使能 (Voltage output buffer enable)
此位用于使能/禁止电压输出缓冲器，以获得更高的驱动能力。
0：禁止输出缓冲器
1：使能输出缓冲器

位 7 **MUX_SEG**：多路复用区段使能 (Mux segment enable)
此位用于使能 SEG 引脚重映射。四个 SEG 引脚可与 SEG[31:28] 多路复用。请参见*第 19.3.7 节*。
0：禁止 SEG 引脚多路复用
1：SEG[31:28] 与 SEG[43:40] 多路复用

位 6:5 **BIAS[1:0]**：偏置选择器 (Bias selector)
这些位决定使用的偏置。禁止使用值 11。
00：偏置 1/4
01：偏置 1/2
10：偏置 1/3
11：保留

位 4:2 **DUTY[2:0]**：占空比选择 (Duty selection)
这些位决定占空比。禁止使用值 101、110 和 111。
000：静态占空比
001：1/2 占空比
010：1/3 占空比
011：1/4 占空比
100：1/8 占空比
101：保留
110：保留
111：保留

位 1 **VSEL**：电压源选择 (Voltage source selection)
VSEL 位决定 LCD 的电压源。
0：内部源（电压升压转换器）
1：外部源（VLCD 引脚）

位 0 **LCDEN**：LCD 控制器使能 (LCD controller enable)
通过软件将此位置 1 可使能 LCD 控制器/驱动器。通过软件将此位清零可在下一帧开始时关闭 LCD。当 LCD 禁止时，所有 COM 和 SEG 引脚均驱动到 V_{SS}。
0：禁止 LCD 控制器
1：使能 LCD 控制器

注： *当使能 LCD 时（LCD_SR 中的 ENS 位置 1），VSEL、MUX_SEG、BIAS、DUTY 和 BUFEN 位处于写保护状态。*

19.6.2 LCD 帧控制寄存器 (LCD_FCR)

LCD frame control register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	PS[3:0]				DIV[3:0]				BLINK[1:0]	
						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
BLINKF[2:0]			CC[2:0]			DEAD[2:0]			PON[2:0]			UDDIE	Res.	SOFIE	HD
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

位 31:26 保留，必须保持复位值

位 25:22 **PS[3:0]**：PS 16 位预分频器 (PS 16-bit prescaler)
这些位通过软件写入，用于定义 PS 16 位预分频器的分频系数。
ck_ps = LCDCLK/(2)。请参见*第 19.3.2 节*。
0000：ck_ps = LCDCLK
0001：ck_ps = LCDCLK/2
0002：ck_ps = LCDCLK/4
...
1111：ck_ps = LCDCLK/32768

位 21:18 **DIV[3:0]**：DIV 时钟分频器 (DIV clock divider)
这些位通过软件写入，用于定义 DIV 分频器的分频系数。请参见*第 19.3.2 节*。
0000：ck_div = ck_ps/16
0001：ck_div = ck_ps/17
0002：ck_div = ck_ps/18
...
1111：ck_div = ck_ps/31

位 17:16 **BLINK[1:0]**：闪烁模式选择 (Blink mode selection)

00：禁止闪烁
01：在 SEG[0]、COM[0] 上启用闪烁（1 个像素）
10：在 SEG[0]、所有 COM 上启用闪烁（最多 8 个像素，取决于编程的占空比）
11：在所有 SEG 和所有 COM 上启用闪烁（所有像素）

位 15:13 **BLINKF[2:0]**：闪烁频率选择 (Blink frequency selection)

000：$f_{LCD}/8$
001：$f_{LCD}/16$
010：$f_{LCD}/32$
011：$f_{LCD}/64$
100：$f_{LCD}/128$
101：$f_{LCD}/256$
110：$f_{LCD}/512$
111：$f_{LCD}/1024$

位 12:10 **CC[2:0]**：对比度控制 (Contrast control)

这些位指定 V_{LCD} 最大电压之一（与 V_{DD} 无关）。范围为 2.60 V 至 3.51 V。

000：V_{LCD0}
001：V_{LCD1}
010：V_{LCD2}
011：V_{LCD3}
100：V_{LCD4}
101：V_{LCD5}
110：V_{LCD6}
111：V_{LCD7}

有关 V_{LCDx} 的值，请参见产品数据手册。

位 9:7 **DEAD[2:0]**：死区持续时间 (Dead time duration)

这些位通过软件写入，用于配置帧间死区的长度。在死区内，COM 和 SEG 电压电平保持为 0 V 以降低对比度，而无需修改帧速率。

000：无死区
001：1 个相位周期死区
010：2 个相位周期死区
.....
111：7 个相位周期死区

位 6:4 **PON[2:0]**：脉冲打开持续时间 (Pulse ON duration)

这些位通过软件写入，用于根据 ck_ps 脉冲定义脉冲持续时间。较短的脉冲有助于降低功耗，但内部电阻较高的显示器可能需要更长的脉冲才能达到令人满意的对比度。

请注意，脉冲长度永远不会超过预分频 LCD 时钟周期的一半。

000：0
001：1/ck_ps
010：2/ck_ps
011：3/ck_ps
100：4/ck_ps
101：5/ck_ps
110：6/ck_ps
111：7/ck_ps

PON 持续时间示例（LCDCLK = 32.768 kHz 且 PS = 0x03）：

000：0 µs
001：244 µs
010：488 µs
011：782 µs
100：976 µs
101：1.22 ms
110：1.46 ms
111：1.71 ms

位 3 **UDDIE**：更新显示完成中断使能 (Update display done interrupt enable)

此位由软件置 1 和清零。

0：禁止 LCD 更新显示完成中断
1：使能 LCD 更新显示完成中断

位 2 保留，必须保持复位值

位 1 **SOFIE**：帧起始中断使能 (Start of frame interrupt enable)

此位由软件置 1 和清零。

0：禁止 LCD 帧起始中断
1：使能 LCD 帧起始中断

位 0 **HD**：高驱动使能 (High drive enable)

此位通过软件写入，用于使能低电阻分频器。内部电阻较高的显示器可能需要更长的驱动时间才能达到令人满意的对比度。在容许额外功耗的情况下，此位非常有用。

0：禁止永久高驱动
1：使能永久高驱动。当 HD=1 时，必须将 PON 位编程为 001

注： *可以随时更新此寄存器中的数据，但仅在下一帧开始时才被使用（能够立即影响器件行为的 UDDIE 和 SOFIE 除外）。*

CC[2:0] 位的新值也会立即应用，但其对器件的作用会在下一帧开始时被电压发生器延迟。

读取该寄存器可获得在寄存器中写入的最后一个值，而不是用于显示当前帧的配置。

注： *当 LCD_CR 寄存器中的 BUFEN 位置 1 时，无论 HD 或 PON[2:0] 位配置如何，低阻值电阻分压器网络都会自动禁止。*

19.6.3 LCD 状态寄存器 (LCD_SR)

LCD status register

偏移地址：0x08

复位值：0x0000 0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCRSF	RDY	UDD	UDR	SOF	ENS
										r	r	r	rs	r	r

位 31:6 保留，必须保持复位值

位 5 **FCRSF**：LCD 帧控制寄存器同步标志 (LCD Frame Control Register Synchronization flag)

每次在 LCDCLK 域中更新 LCD_FCR 寄存器时，此位都会由硬件置 1。对 LCD_FCR 寄存器执行写操作时，此位将由硬件清零。

0：LCD 帧控制寄存器尚未同步

1：LCD 帧控制寄存器已同步

位 4 **RDY**：就绪标志 (Ready flag)

此位通过硬件置 1 和清零。它指示升压转换器的状态。

0：未就绪

1：升压转换器已使能且准备好提供正确的电压

位 3 **UDD**：更新显示完成 (Update Display Done)

此位将由硬件置 1，通过向 LCD_CLR 寄存器中的 UDDC 写入 1 进行清零。位置 1 的优先级高于位清零。

0：无事件

1：更新显示请求完成。如果 LCD_FCR 寄存器中的 UDDIE 位置 1，将产生 UDD 中断

注： 如果器件处于停止模式（未提供 PCLK），即使 UDDIE = 1，UDD 也不会产生中断。如果显示器未使能，将永远不会发生 UDD 中断。

位 2 **UDR**：更新显示请求 (Update display request)

每次软件修改 LCD_RAM 时，必须将 UDR 位置 1 以将更新的数据传送到二级缓冲器。UDR 位在更新结束之前保持置 1，在此期间 LCD_RAM 处于写保护状态。

0：无影响

1：更新显示请求

注： 当禁止显示器时，针对所有 LCD_DISPLAY 存储单元执行更新。当使能显示器时，仅针对公共线激活（取决于 DUTY）的存储单元执行更新。例如，如果 DUTY = 1/2，将仅更新 COM0 和 COM1 的 LCD_DISPLAY。

注： 当此位已为 1 时，写入 0 或写入 1 都无影响。此位只能由硬件清零。仅当 LCDEN=1 时，此位才能清零

位 1 **SOF**：帧起始标志 (Start of frame flag)
此位在新的帧开始时由硬件置 1，同时更新显示数据。此位通过向 LCD_CLR 寄存器中的 SOFC 位写入 1 进行清零。位清零的优先级高于位置 1。
0：无事件
1：发生帧起始事件。如果 SOFIE 位置 1，则产生 LCD 帧起始中断

ENS：LCD 使能状态 (LCD enabled status)
位 0 此位通过硬件置 1 和清零。它指示 LCD 控制器状态。
0：禁止 LCD 控制器
1：使能 LCD 控制器
注： 当 LCD_CR 寄存器中的 LCDEN 位从 0 变为 1 时，ENS 位立即置 1。禁止时，此位反映 LCD 的实际状态，因此在最后显示的帧结束时变为 0。

19.6.4 LCD 清零寄存器 (LCD_CLR)

LCD clear register

偏移地址：0x0C

复位值：0x0000 0000

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

位 31:4 保留，必须保持复位值

位 3 **UDDC**：更新显示完成清零 (Update display done clear)
此位由软件写入，用于将 LCD_SR 寄存器中的 UDD 标志清零。
0：无影响
1：将 UDD 标志清零

位 2 保留，必须保持复位值

位 1 **SOFC**：帧起始标志清零 (Start of frame flag clear)
此位由软件写入，用于将 LCD_SR 寄存器中的 SOF 标志清零。
0：无影响
1：将 SOF 标志清零

位 0 保留，必须保持复位值

19.6.5 LCD 显示器存储器 (LCD_RAM)

LCD display memory

偏移地址：0x14 到 0x50

复位值：0x0000 0000

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

位 31:0 SEGMENT_DATA[31:0]

每个位对应于 LCD 显示器的一个像素。

0：像素未激活

1：像素激活

19.6.6 LCD 寄存器映射

下表对 LCD 寄存器进行了汇总。

表 97. LCD 寄存器映射和复位值

偏移	寄存器	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
0x00	LCD_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUFEN	MUX_SEG	BIAS[1:0]		DUTY [2:0]			VSEL	LCDEN
	Reset value																								0	0	0	0	0	0	0	0	0
0x04	LCD_FCR	Res.	Res.	Res.	Res.	Res.	Res.	PS[3:0]				DIV[3:0]				BLINK[1:0]		BLINKF[2:0]			CC [2:0]			DEAD [2:0]			PON [2:0]			UDDIE	Res.	SOFIE	HD
	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
0x08	LCD_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCRSF	RDY	UDD	UDR	SOF	ENS
	Reset value																											1	0	0	0	0	0
0x0C	LCD_CLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDDC	Res.	SOFC	Res.
	Reset value																													0		0	
0x14	LCD_RAM (COM0)	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S09	S08	S07	S06	S05	S04	S03	S02	S01	S00
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	S43	S42	S41	S40	S39	S38	S37	S36	S35	S34	S33	S32
																						0	0	0	0	0	0	0	0	0	0	0	0
0x1C	LCD_RAM (COM1)	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S09	S08	S07	S06	S05	S04	S03	S02	S01	S00
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	S43	S42	S41	S40	S39	S38	S37	S36	S35	S34	S33	S32
																						0	0	0	0	0	0	0	0	0	0	0	0
0x24	LCD_RAM (COM2)	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S09	S08	S07	S06	S05	S04	S03	S02	S01	S00
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	S43	S42	S41	S40	S39	S38	S37	S36	S35	S34	S33	S32
																						0	0	0	0	0	0	0	0	0	0	0	0
0x2C	LCD_RAM (COM3)	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S09	S08	S07	S06	S05	S04	S03	S02	S01	S00
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	S43	S42	S41	S40	S39	S38	S37	S36	S35	S34	S33	S32
																						0	0	0	0	0	0	0	0	0	0	0	0

表 97. LCD 寄存器映射和复位值（续）

偏移	寄存器	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
0x34	LCD_RAM (COM4)	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S09	S08	S07	S06	S05	S04	S03	S02	S01	S00
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	S39	S38	S37	S36	S35	S34	S33	S32
																										0	0	0	0	0	0	0	0
0x3C	LCD_RAM (COM5)	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S09	S08	S07	S06	S05	S04	S03	S02	S01	S00
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	S39	S38	S37	S36	S35	S34	S33	S32
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	LCD_RAM (COM6)	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S09	S08	S07	S06	S05	S04	S03	S02	S01	S00
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	S39	S38	S37	S36	S35	S34	S33	S32
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	LCD_RAM (COM7)	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13	S12	S11	S10	S09	S08	S07	S06	S05	S04	S03	S02	S01	S00
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	S39	S38	S37	S36	S35	S34	S33	S32
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址表，请参见第 63 页的第 2.2 节。

20 触摸感应控制器 (TSC)

20.1 简介

触摸感应控制器提供了一种简单的解决方案，可用于向任何应用添加电容感应功能。电容感应技术能够检测电极附近的手指，该电极受绝缘体（例如玻璃、塑料）保护以防直接接触。由手指（或任何导电物质）产生的电容变化可基于表面电荷转移采集原理，使用已证实有效的实现方法进行测量。

触摸感应控制器受 STMTouch 触摸感应固件库完全支持，该固件库可供免费使用，用于在最终应用中可靠地实现触摸感应功能。

20.2 TSC 主要特性

触摸感应控制器具有以下主要特性：

- 采用经验证的稳定的表面电荷转移采集原理
- 支持多达 21 路容性感应通道
- 可以并行采集多达 7 路容性感应通道，因此具备极佳的响应时间
- 具有扩频功能，可以提高系统在噪声环境中的稳定性
- 对电荷转移采集序列完全采用硬件管理
- 可编程电荷转移频率
- 可编程采样电容 I/O 引脚
- 可编程通道 I/O 引脚
- 可编程最大计数值，从而避免通道出现故障时采集时间过长
- 具有中断功能的专用采集结束和最大计数错误标志
- 一个采样电容可用于多达 3 路容性感应通道，从而可减少系统组件
- 与接近、触键、线性和旋转触摸传感器实现方案兼容
- 设计为与 STMTouch 触摸感应固件库配合使用

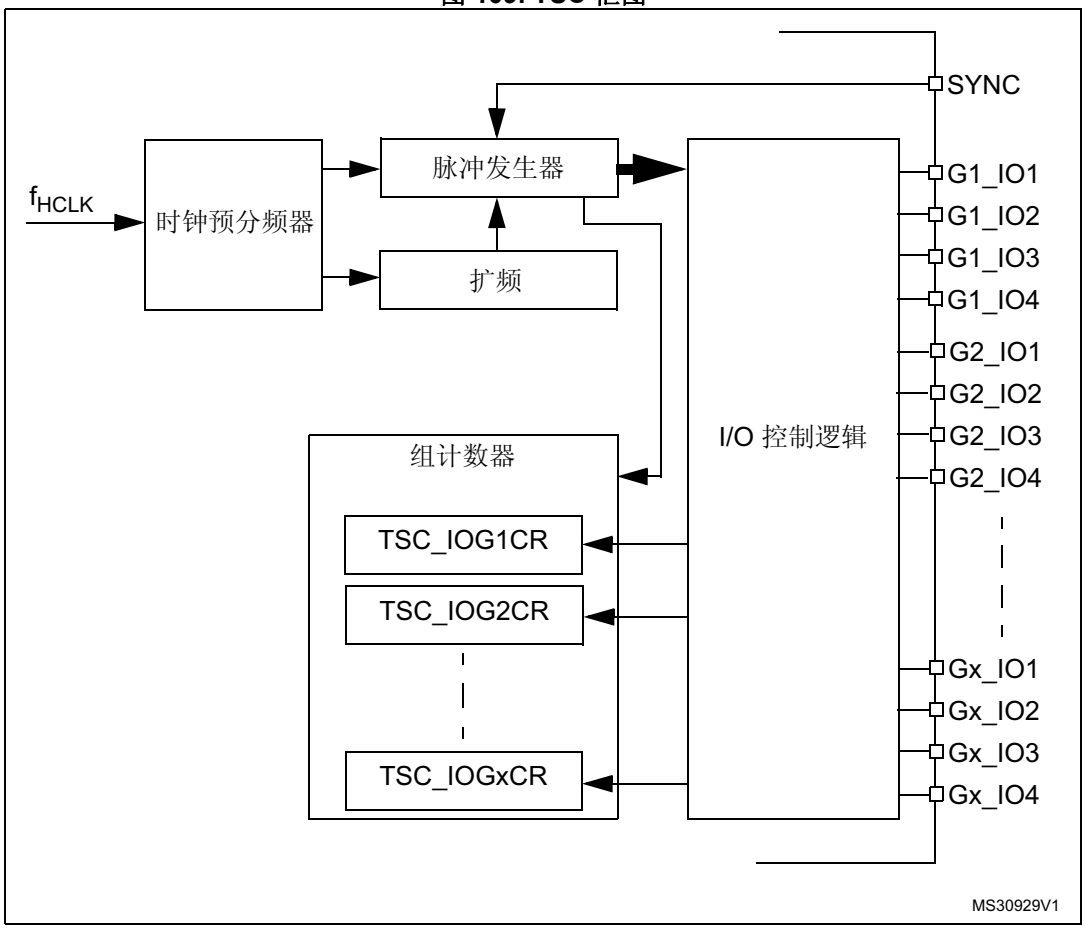
注： *电容感应通道的数量取决于封装尺寸并受 I/O 可用性的限制。*

20.3 TSC 功能说明

20.3.1 TSC 框图

触摸感应控制器的框图如*图 109：TSC 框图*所示。

图 109. TSC 框图



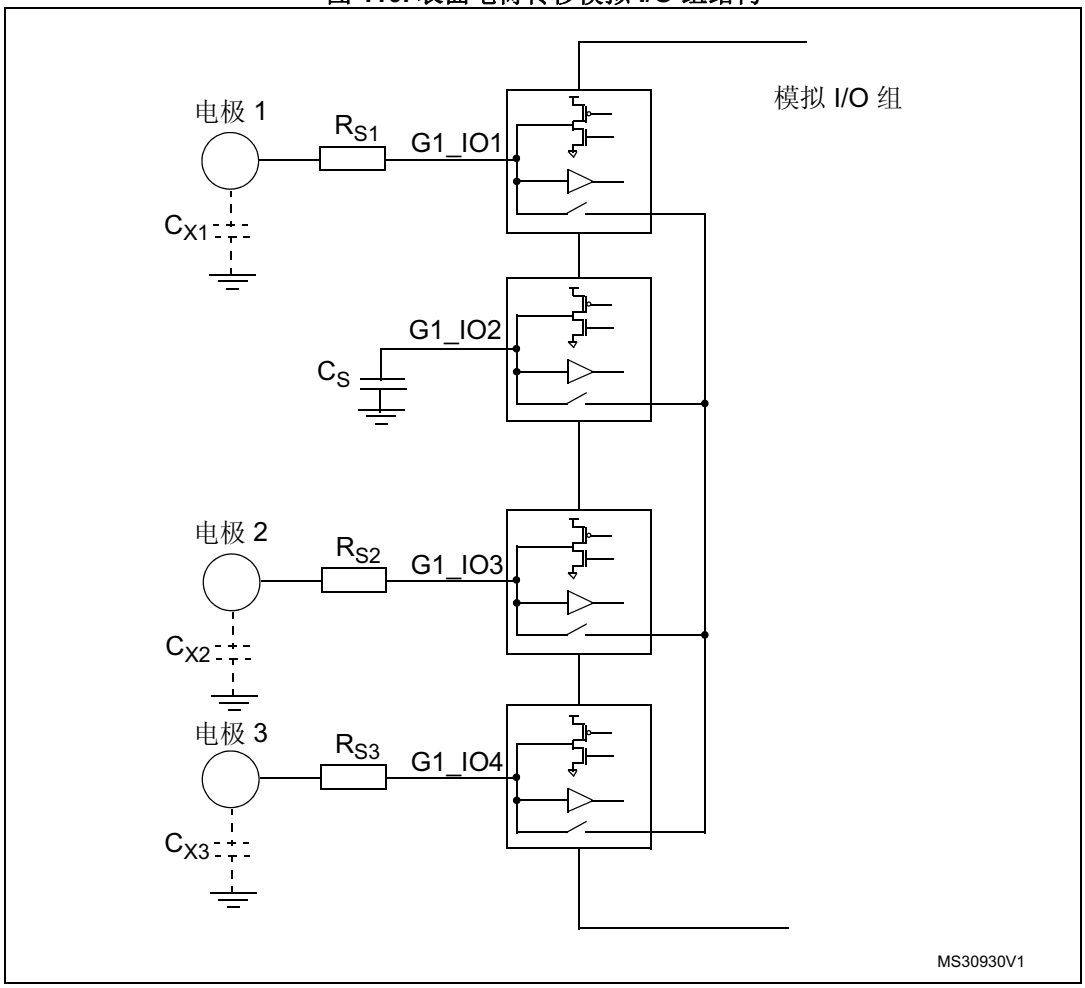
20.3.2 表面电荷转移采集概述

表面电荷转移采集是用于测量电容的一种成熟、稳定且有效的方式。它使用最少数量的外部组件对单端电极类型进行操作。该采集围绕模拟 I/O 组设计，后者由四个 GPIO 组成。（请参见*图 110*）。提供多个模拟 I/O 组，以便可以同时采集多个电容感应通道并支持更多的电容感应通道。在同一个模拟 I/O 组内，电容感应通道的采集操作是连续的。

其中一个 GPIO 专门用于采样电容 C_S。每个模拟 I/O 组在同一时间必须仅使能一个采样电容 I/O。

其余的 GPIO 专用于电极，通常称为通道。对于一些特定需要（如接近检测），每个模拟 I/O 组也可以同时使能多个通道。

图 110. 表面电荷转移模拟 I/O 组结构



注： *Gx_IOy，其中 x 是模拟 I/O 组编号，y 是所选组内的 GPIO 编号。*

表面电荷转移采集原理包括对电极电容 (C_X) 进行充电并将一部分累积电荷转移到采样电容 (C_S)。重复该过程，直到 C_S 两端的电压达到给定的阈值（在本例中为 V_{IH}）。达到阈值所需的电荷转移次数直接表示电极电容的大小。

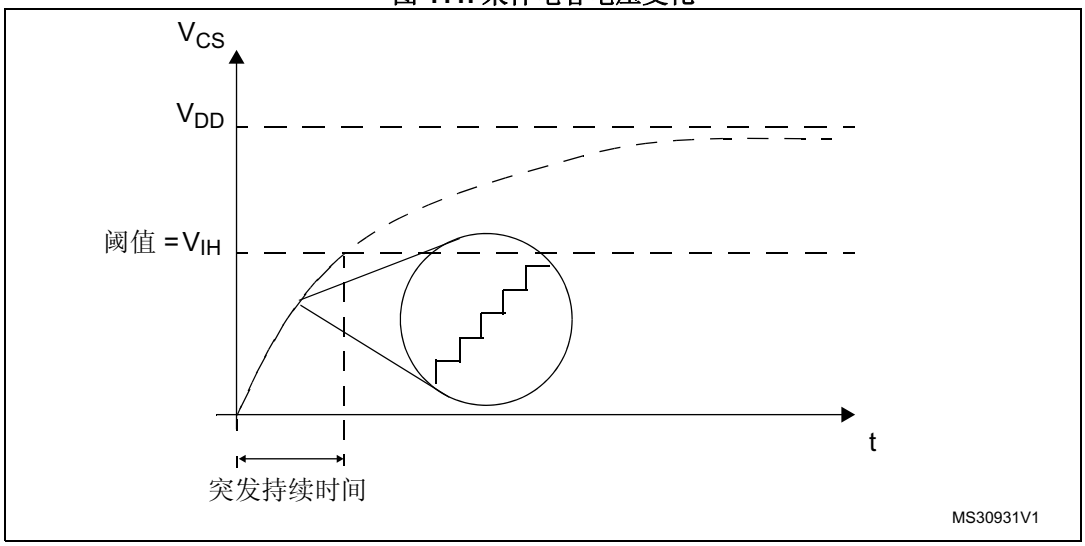
表 98 详细说明了电容感应通道 1 的电荷转移采集序列。重复状态 3 到 7，直到 C_S 两端的电压达到给定的阈值。同一序列适用于其他通道的采集。电极串联电阻 R_S 提高了解决方案的 ESD 抗扰性。

表 98. 采集序列汇总

状态	G1_IO1（通道）	G1_IO2（采样）	G1_IO3（通道）	G1_IO4（通道）	状态说明
#1	输入浮空且模拟开关闭合	开漏输出处于低电平，且模拟开关闭合	输入浮空且模拟开关闭合		将所有 C_X 和 C_S 放电
#2	输入浮空				死区时间
#3	推挽输出处于高电平	输入浮空			将 C_{X1} 充电
#4	输入浮空				死区时间
#5	输入浮空且模拟开关闭合		输入浮空		将电荷从 C_{X1} 转移到 C_S
#6	输入浮空				死区时间
#7	输入浮空				测量 C_S 电压

下图详细描述了采样电容 C_S 的电压随时间的变化情况：

图 111. 采样电容电压变化



20.3.3 复位和时钟

TSC 时钟源是 AHB 时钟 (HCLK)。两个可编程预分频器用于产生脉冲发生器和扩频内部时钟：

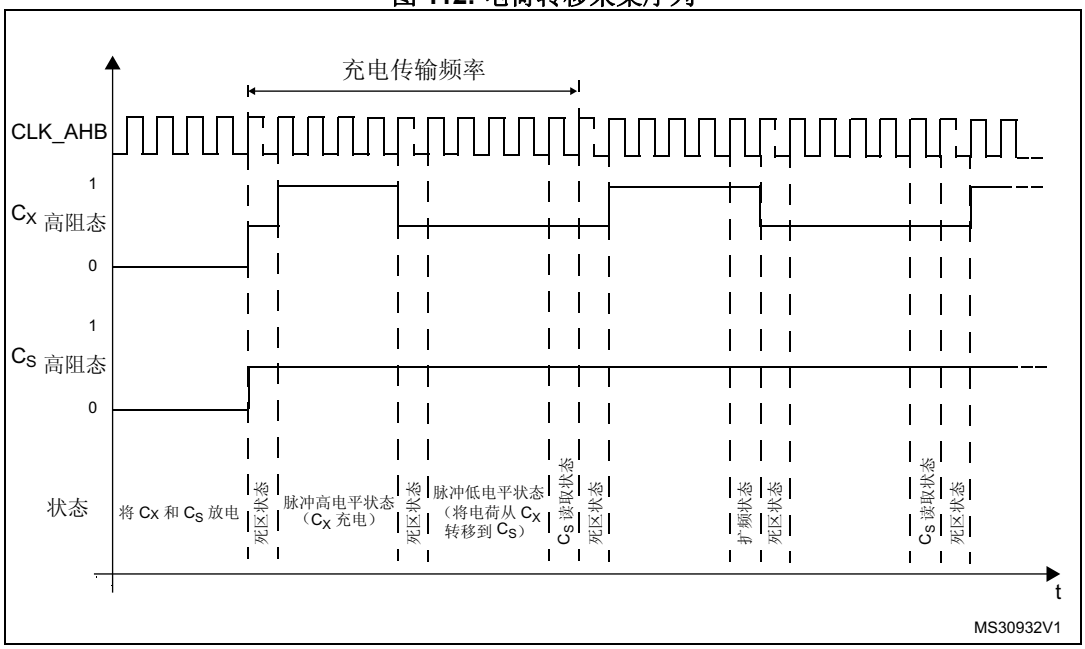
- 脉冲发生器时钟 (PGCLK) 使用 TSC_CR 寄存器的 PGPSC[2:0] 位进行定义
- 扩频时钟 (SSCLK) 使用 TSC_CR 寄存器的 SSPSC 位进行定义

复位和时钟控制器 (RCC) 提供用于使能触摸感应控制器时钟及复位该外设的专用位。有关详细信息，请参见*第 8 节：复位和时钟控制 (RCC)*。

20.3.4 电荷转移采集序列

图 112 详细描述了电荷转移采集序列的示例。

图 112. 电荷转移采集序列



为提高灵活性，可完全配置电荷转移频率。可以使用 TSC_CR 寄存器中的 CTPH[3:0] 和 CTPL[3:0] 位定义脉冲高电平状态（为 C_X 充电）和脉冲低电平状态（将电荷从 C_X 转移到 C_S）的持续时间。脉冲高电平和低电平状态持续时间的标准范围为 500 ns 到 2 µs。为了确保正确测量电极电容，必须设置脉冲高电平状态持续时间以确保 C_X 始终充分充电。

在脉冲高电平和低电平状态之间插入死区时间（在此期间，采样电容 I/O 和通道 I/O 处于输入浮空状态），以确保最优的电荷转移采集序列。死区时间状态的持续时间是 HCLK 的 2 个周期。

如果已启用扩频功能，在脉冲高电平状态结束时，将增加可变数量的 SSCLK 时钟周期。

在脉冲低电平状态的结束时，会读取采样电容 I/O，确定 C_S 两端的电压是否达到给定阈值，读取操作的持续时间是 HCLK 的一个周期。

注： *禁止采用以下 TSC 控制寄存器配置：*

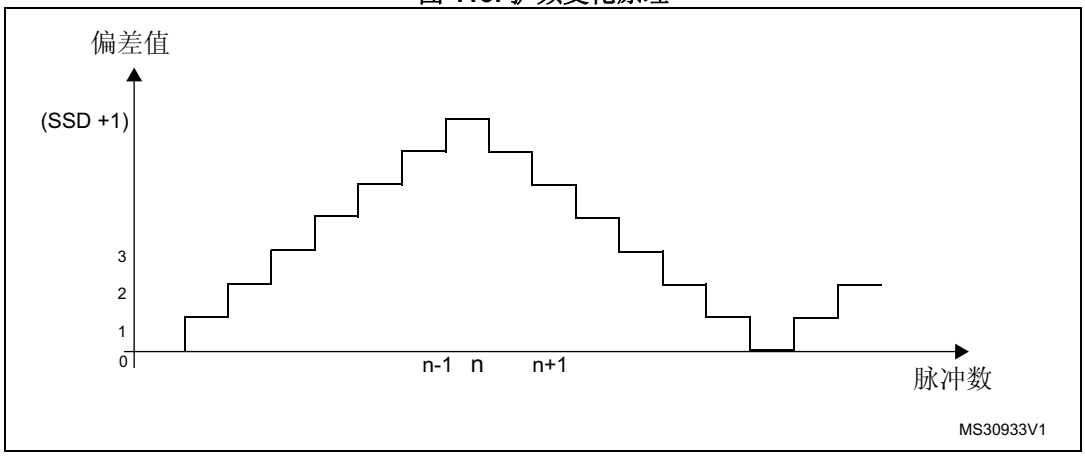
- *PGPSC 位设置为 “000” 且 CTPL 位设置为 “0000”*
- *PGPSC 位设置为 “000” 且 CTPL 位设置为 “0001”*
- *PGPSC 位设置为 “001” 且 CTPL 位设置为 “0000”*

20.3.5 扩频功能

扩频功能用于使电荷转移频率发生变化。这样做是为了改善噪声环境下电荷转移采集的稳定性，并减少由此产生的辐射。最大频率变化范围为标称电荷转移周期的 10% 到 50%。例如，对于 250 kHz (4 µs) 的标称电荷转移频率，典型的扩频偏差为 10% (400 ns)，这将导致出现 227 kHz 的最低电荷转移频率。

实际上，扩频包括使用以下所示的原理将可变数量的 SSCLK 周期添加到脉冲高电平状态：

图 113. 扩频变化原理



下表详细列出了不同 HCLK 设置对应的最大频率偏差：

表 99. 扩频偏差与 AHB 时钟频率之间的关系

f_{HCLK}	扩频步长	最大扩频偏差
32 MHz	31.25 ns	8206 ns

可以使用 TSC_CR 寄存器中的 SSE 位禁止 / 使用扩频功能。也可以根据器件 HCLK 时钟频率和所选电荷转移频率使用 TSC_CR 寄存器中的 SSPSC 和 SSD[6:0] 配置频率偏差。

20.3.6 最大计数错误

最大计数错误可防止因电容感应通道发生故障而导致采集时间较长。它包括为模拟 I/O 组计数器指定最大计数值。最大计数值可使用 TSC_CR 寄存器中的 MCV[2:0] 位进行指定。采集组计数器达到此最大值后，正在进行的采集操作将停止，并且采集结束标志（EOAF 位）和最大计数错误标志（MCEF 位）都会置 1。如果相应的采集结束（EOAIE 位）或 / 和最大计数错误（MCEIE 位）中断使能位都置 1，还可以产生中断。

20.3.7 采样电容 I/O 和通道 I/O 模式选择

要实现由触摸感应控制器控制 GPIO，必须通过标准 GPIO 寄存器和 GPIOxAFR 寄存器使能相应的复用功能。

GPIO 模式由 TSC 控制，并使用 TSC_IOSCR 和 TSC_IOCCR 寄存器进行定义。

当未执行任何采集操作时，由触摸感应控制器控制的所有 I/O 都处于默认状态。当执行采集操作时，仅未使用的 I/O（未定义为采样电容 I/O 或通道 I/O）处于默认状态。TSC_CR 寄存器中的 IODEF 位定义处于默认状态的 I/O 的配置。下表根据 I/O 模式对其配置进行了汇总。

表 100. I/O 状态（取决于其模式和 IODEF 位值）

IODEF 位	采集状态	未使用的 I/O 模式	通道 I/O 模式	采样电容 I/O 模式
0 （推挽输出处于低电平）	未执行	推挽输出 处于低电平	推挽输出 处于低电平	推挽输出 处于低电平
0 （推挽输出处于低电平）	正在执行	推挽输出 处于低电平	-	-
1 （输入浮空）	未执行	输入浮空	输入浮空	输入浮空
1 （输入浮空）	正在执行	输入浮空	-	-

未使用的 I/O 模式

未使用的 I/O 对应于由 TSC 外设控制的 GPIO，但未定义为电极 I/O 或采样电容 I/O。

采样电容 I/O 模式

要实现由 TSC 外设控制采样电容 I/O，必须首先将相应的 GPIO 设置为复用输出开漏模式，然后必须将 TSC_IOSCR 寄存器中相应的 Gx_IOy 位置 1。

每个模拟 I/O 组在同一时间必须仅使能一个采样电容。

通道 I/O 模式

要实现由 TSC 外设控制通道 I/O，必须首先将相应的 GPIO 设置为复用输出推挽模式，然后必须将 TSC_IOCCR 寄存器中的相应 Gx_IOy 位置 1。

对于要求更高等效电极表面的接近检测或为了加快采集过程，可以使能属于同一模拟 I/O 组的多个通道并同时进行采集。

注： *在采集阶段，即使未使能 TSC 外设复用功能，只要 TSC_IOSCR 或 TSC_IOCCR 位置 1，相应的 GPIO 模拟开关就由触摸感应控制器自动进行控制。*

20.3.8 采集模式

触摸感应控制器具有两种采集模式：

- 正常采集模式：TSC_CR 寄存器中的 START 位置 1 后，采集会立即开始。
- 同步采集模式：通过将 TSC_CR 寄存器中的 START 位置 1 来使能采集，但仅当检测到下降沿或上升沿并且 SYNC 输入引脚处于高电平时，采集才会开始。此模式可用于将电容感应通道采集与外部信号同步，而不会额外增加 CPU 负载。

TSC_IOGCSR 寄存器中的 GxE 位用于指定使能哪个模拟 I/O 组（相对计数器正在计数）。已禁止模拟 I/O 组的 C_S 电压不会受到监视，而且该组不会参与触发采集结束标志。但是，如果已禁止模拟 I/O 组包含一些通道，这些通道将被脉冲化。

已使能的模拟 I/O 组的 C_S 电压达到给定阈值时，TSC_IOGCSR 寄存器中相应的 GxS 位置 1。当已使能的所有模拟 I/O 组的采集完成时（已使能的所有模拟 I/O 组的所有 GxS 位都置 1），TSC_ISR 寄存器中的 EOAF 标志会置 1。如果 TSC_IER 寄存器中的 EOAIE 位置 1，则会产生中断请求。

如果检测到最大计数错误，正在进行的采集将会停止，并且 TSC_ISR 寄存器中的 EOAF 和 MCEF 标志都置 1。如果相应位（TSCIER 寄存器的 EOAIE 和 MCEIE 位）置 1，可以针对这两种事件产生中断请求。注意，当检测到最大计数错误时，已使能模拟 I/O 组中的其余 GxS 位不会置 1。

要清除中断标志，TSC_ICR 寄存器中相应的 EOAIC 和 MCEIC 位必须置 1。

当开始新的采集时，模拟 I/O 组计数器清零。采集完成时，这些计数器会更新为在相应通道上产生的电荷转移周期数。

20.3.9 I/O 滞后和模拟开关控制

为提高灵活性，触摸感应控制器还可用于控制每个 Gx_IOy 的施密特触发器滞后和模拟开关。在触摸感应控制器使能后，无论 I/O 控制模式是由标准 GPIO 寄存器还是由其他外设进行控制，此控制均可用。该控制还可用于执行不同的采集序列或用于其他用途。

为提高系统的抗扰性，必须通过复位 TSC_IOHCR 寄存器中相应的 Gx_IOy 位来禁止由 TSC 控制的 GPIO 施密特触发器滞后。

20.4 TSC 低功耗模式

表 101. 低功耗模式对 TSC 的作用

模式	说明
睡眠	无影响。外设中断可使器件退出睡眠模式。
低功耗运行	无影响。
低功耗睡眠	无影响。外设中断可使器件退出低功耗睡眠模式。
停止 0/停止 1	外设寄存器内容仍被保持。
停止 2	
待机	掉电。在退出待机或关断模式后，必须重新初始化外设。
关断	

20.5 TSC 中断

表 102. 中断控制位

中断事件	使能控制位	事件标志	标志清零位	退出睡眠模式	退出停止模式	退出待机模式
采集结束	EOAIE	EOAIF	EOAIC	是	否	否
最大计数错误	MCEIE	MCEIF	MCEIC	是	否	否

20.6 TSC 寄存器

有关寄存器说明中使用的缩写，请参见参考手册中的*第 58 页的第 1.2 节*。

外设寄存器可按字（32 位）进行访问。

20.6.1 TSC 控制寄存器 (TSC_CR)

TSC control register

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
CTPH[3:0]				CTPL[3:0]				SSD[6:0]							SSE
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
SSPSC	PGPSC[2:0]			Res.	Res.	Res.	Res.	MCV[2:0]			IODEF	SYNC POL	AM	START	TSCE
rw	rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 **CTPH[3:0]**：电荷转移脉冲处于高电平 (Charge transfer pulse high)

这些位将由软件置 1 和清零。它们定义电荷转移脉冲处于高电平状态的持续时间（C_X 充电）。

0000：1x t_{PGCLK}
0001：2x t_{PGCLK}
...
1111：16x t_{PGCLK}

注： 正在进行采集时不得修改这些位。

位 27:24 **CTPL[3:0]**：电荷转移脉冲处于低电平 (Charge transfer pulse low)

这些位将由软件置 1 和清零。它们定义电荷转移脉冲处于低电平状态的持续时间（将电荷从 C_X 转移到 C_S）。

0000：1x t_{PGCLK}
0001：2x t_{PGCLK}
...
1111：16x t_{PGCLK}

注： 正在进行采集时不得修改这些位。

注： 有些配置禁止采用。有关详细信息，请参见第 20.3.4 节：电荷转移采集序列。

位 23:17 **SSD[6:0]**：扩频偏差 (Spread spectrum deviation)

这些位将由软件置 1 和清零。它们定义扩频偏差，扩频偏差是由将可变数量的 SSCLK 时钟周期添加到电荷转移脉冲高电平状态产生的。

0000000：1x t_{SSCLK}
0000001：2x t_{SSCLK}
...
1111111：128x t_{SSCLK}

注： 正在进行采集时不得修改这些位。

位 16 **SSE**：扩频使能 (Spread spectrum enable)

此位由软件置 1 和清零，用于使能/禁止扩频功能。

0：禁止扩频
1：使能扩频

注： 正在进行采集时不得修改此位。

位 15 **SSPSC**：扩频预分频器 (Spread spectrum prescaler)

此位由软件置 1 和清零。它选择用于生成扩频时钟 (SSCLK) 的 AHB 时钟分频器。

0：f_{HCLK}

1：f_{HCLK} /2

注： 正在进行采集时不得修改此位。

位 14:12 **PGPSC[2:0]**：脉冲发生器预分频器 (pulse generator prescaler)

这些位由软件置 1 和清零。它们选择用于生成脉冲发生器时钟 (PGCLK) 的 AHB 时钟分频器。

000：f_{HCLK}

001：f_{HCLK} /2

010：f_{HCLK} /4

011：f_{HCLK} /8

100：f_{HCLK} /16

101：f_{HCLK} /32

110：f_{HCLK} /64

111：f_{HCLK} /128

注： 正在进行采集时不得修改这些位。

注： 有些配置禁止采用。有关详细信息，请参见第 20.3.4 节：电荷转移采集序列。

位 11:8 保留，必须保持复位值。

位 7:5 **MCV[2:0]**：最大计数值 (Max count value)

这些位将由软件置 1 和清零。它们定义在产生最大计数错误之前可以生成的电荷转移脉冲的最大数量。

000：255

001：511

010：1023

011：2047

100：4095

101：8191

110：16383

111：保留

注： 正在进行采集时不得修改这些位。

位 4 **IODEF**：I/O 默认模式 (I/O Default mode)

此位由软件置 1 和清零。它定义当未执行任何采集时所有 TSC I/O 的配置。当存在正在执行的采集时，它定义所有未使用 I/O（未定义为采样电容 I/O 或通道 I/O）的配置。

0：强制 I/O 处于推挽输出低电平状态

1：I/O 处于输入浮空状态

注： 正在进行采集时不得修改此位。

位 3 **SYNCPOL**：同步引脚极性 (Synchronization pin polarity)

此位由软件置 1 和清零，以选择同步输入引脚的极性。

0：仅下降沿

1：上升沿和高电平

位 2 **AM**：采集模式 (Acquisition mode)

此位由软件置 1 和清零，用以选择采集模式。

0：正常采集模式（START 位置 1 后，采集立即开始）

1：同步采集模式（如果 START 位置 1，并在 SYNC 输入引脚上检测到选定的信号，则采集开始）

注： 正在进行采集时不得修改此位。

位 1 **START**：启动新的采集 (Start a new acquisition)

此位由软件置 1，用于启动新的采集。采集完成后，此位会由硬件立即清零，也可由软件将此位清零以取消正在执行的采集。

0：不启动采集

1：启动新的采集

位 0 **TSCE**：触摸感应控制器使能 (Touch sensing controller enable)

此位由软件置 1 和清零，用于使能/禁止触摸感应控制器。

0：禁止触摸感应控制器

1：使能触摸感应控制器

注： 禁止触摸感应控制器后，TSC 寄存器设置不起任何作用。

20.6.2 TSC 中断使能寄存器 (TSC_IER)

TSC interrupt enable register

偏移地址：0x04

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIE	EOAIE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **MCEIE**：最大计数错误中断使能 (Max count error interrupt enable)

此位由软件置 1 和清零，用于使能/禁止最大计数错误中断。

0：禁止最大计数错误中断

1：使能最大计数错误中断

位 0 **EOAIE**：采集结束中断使能 (End of acquisition interrupt enable)

此位由软件置 1 和清零，用于使能/禁止采集结束中断。

0：禁止采集结束中断

1：使能采集结束中断

20.6.3 TSC 中断清零寄存器 (TSC_ICR)

TSC interrupt clear register

偏移地址：0x08

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIC	EOAIC
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **MCEIC**：最大计数错误中断清零 (Max count error interrupt clear)

此位由软件置 1，用于清除最大计数错误标志。该标志复位后，此位由硬件清零。写入“0”则不会带来任何影响。

0：无影响

1：将 TSC_ISR 寄存器中相应的 MCEF 清零

位 0 **EOAIC**：采集结束中断清零 (End of acquisition interrupt clear)

此位由软件置 1，用于清除采集结束标志。该标志复位后，此位由硬件清零。写入“0”则不会带来任何影响。

0：无影响

1：将 TSC_ISR 寄存器的相应 EOAF 清零

20.6.4 TSC 中断状态寄存器 (TSC_ISR)

TSC interrupt status register

偏移地址：0x0C

复位值：0x0000 0000

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

位 31:2 保留，必须保持复位值。

位 1 **MCEF**：最大计数错误标志 (Max count error flag)

模拟 I/O 组计数器达到指定的最大计数值时，此位由硬件置 1。由软件向 TSC_ICR 寄存器的 MCEIC 位写入 1，可将此位清零。

0：未检测到最大计数错误 (MCE)

1：检测到最大计数错误 (MCE)

位 0 **EOAF**：采集结束标志 (End of acquisition flag)

当已使能的所有组的采集完成时（已使能的所有模拟 I/O 组的所有 GxS 位都置 1，或检测到最大计数错误），此位由硬件置 1。由软件向 TSC_ICR 寄存器的 EOAIC 位写入 1，可将此位清零。

0：采集正在执行或未启动

1：采集完成

20.6.5 TSC I/O 滞后控制寄存器 (TSC_IOHCR)

TSC I/O hysteresis control register

偏移地址：0x10

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
				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
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:0 **Gx_IOy**：Gx_IOy 施密特触发器滞后模式 (Gx_IOy Schmitt trigger hysteresis mode)

这些位由软件置 1 和清零，用于使能/禁止 Gx_IOy 施密特触发器滞后。

0：禁止 Gx_IOy 施密特触发器滞后

1：使能 Gx_IOy 施密特触发器滞后

注： 这些位控制 I/O 施密特触发器滞后，无论 I/O 控制模式如何（即使由标准 GPIO 寄存器进行控制也是如此）。

20.6.6 TSC I/O 模拟开关控制寄存器 (TSC_IOASCR)

TSC I/O analog switch control register

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
				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
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:0 **Gx_IOy**：Gx_IOy 模拟开关使能 (Gx_IOy analog switch enable)

这些位由软件置 1 和清零，用于使能/禁止 Gx_IOy 模拟开关。

0：禁止 Gx_IOy 模拟开关（断开）

1：使能 Gx_IOy 模拟开关（闭合）

注： 这些位控制 I/O 模拟开关，无论 I/O 控制模式如何（即使由标准 GPIO 寄存器进行控制也是如此）。

20.6.7 TSC I/O 采样控制寄存器 (TSC_IOSCR)

TSC I/O sampling control register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
				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
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:0 **Gx_IOy**：Gx_IOy 采样模式 (Gx_IOy sampling mode)

这些位由软件置 1 和清零，用于将 Gx_IOy 配置为采样电容 I/O。每个模拟 I/O 组必须仅有一个 I/O 定义为采样电容。

0：Gx_IOy 未使用

1：Gx_IOy 用作采样电容

注： *正在进行采集时不得修改这些位。*

在采集阶段，即使未使能 TSC 外设复用功能，只要 TSC_IOSCR 位置 1，相应的 GPIO 模拟开关就由触摸感应控制器自动进行控制。

20.6.8 TSC I/O 通道控制寄存器 (TSC_IOCCR)

TSC I/O channel control register

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
Res.	Res.	Res.	Res.	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
				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
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:0 **Gx_IOy**：Gx_IOy 通道模式 (Gx_IOy channel mode)

这些位由软件置 1 和清零，用于将 Gx_IOy 配置为通道 I/O。

0：Gx_IOy 未使用

1：Gx_IOy 用作通道

注： *正在进行采集时不得修改这些位。*

在采集阶段，即使未使能 TSC 外设复用功能，只要 TSC_IOCCR 位置 1，相应的 GPIO 模拟开关就由触摸感应控制器自动进行控制。

20.6.9 TSC I/O 组控制状态寄存器 (TSC_IOGCSR)

TSC I/O group control status register

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G7S	G6S	G5S	G4S	G3S	G2S	G1S
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G7E	G6E	G5E	G4E	G3E	G2E	G1E
									rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。

位 22:16 **GxS**：模拟 I/O 组 x 状态 (Analog I/O group x status)

当相应已使能模拟 I/O 组 x 上的采集完成时，这些位由硬件置 1。当新的采集开始时，这些位由硬件清零。

0：模拟 I/O 组 x 上的采集正在执行或未启动

1：模拟 I/O 组 x 上的采集已完成

注： 检测到最大计数错误后，已使能模拟 I/O 组的其余 GxS 位不置 1。

位 15:7 保留，必须保持复位值。

位 6:0 **GxE**：模拟 I/O 组 x 使能 (Analog I/O group x enable)

这些位由软件置 1 和清零，用于使能/禁止相应模拟 I/O 组 x 上的采集（计数器正在计数）。

0：禁止模拟 I/O 组 x 上的采集

1：使能模拟 I/O 组 x 上的采集

20.6.10 TSC I/O 组 x 计数器寄存器 (TSC_IOGxCR)

TSC I/O group x counter register

x 表示模拟 I/O 组编号。

偏移地址：0x30 + 0x04 * x，(x = 1..7)

复位值：0x0000 0000

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

位 31:14 保留，必须保持复位值。

位 13:0 **CNT[13:0]**：计数器值 (Counter value)

这些位表示为完成采集（C_S 两端电压已达到阈值）在模拟 I/O 组 x 上产生的电荷转移周期数。

20.6.11 TSC 寄存器映射

表 103. TSC 寄存器映射和复位值

偏移	寄存器	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
0x0000	**TSC_CR**	CTPH[3:0]				CTPL[3:0]				SSD[6:0]							SSE	SSPSC	PGPSC[2:0]			Res.	Res.	Res.	Res.	MCV [2:0]			IODEF	SYNCPOL	AM	START	TSCE
	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
0x0004	**TSC_IER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIE	EOAIE
	Reset value																															0	0
0x0008	**TSC_ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIC	EOAIC
	Reset value																															0	0
0x000C	**TSC_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEF	EOAF
	Reset value																															0	0
0x0010	**TSC_IOHCR**	Res.	Res.	Res.	Res.	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
	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
0x0014	Reserved																																
0x0018	**TSC_IOASCR**	Res.	Res.	Res.	Res.	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
	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
0x001C	Reserved																																
0x0020	**TSC_IOSCR**	Res.	Res.	Res.	Res.	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
	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
0x0024	Reserved																																
0x0028	**TSC_IOCCR**	Res.	Res.	Res.	Res.	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
	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
0x002C	Reserved																																
0x0030	**TSC_IOGCSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G7S	G6S	G5S	G4S	G3S	G2S	G1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G7E	G6E	G5E	G4E	G3E	G2E	G1E
	Reset value										0	0	0	0	0	0	0										0	0	0	0	0	0	0
0x0034	**TSC_IOG1CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 103. TSC 寄存器映射和复位值（续）

偏移	寄存器	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
0x0038	**TSC_IOG2CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x003C	**TSC_IOG3CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0040	**TSC_IOG4CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	**TSC_IOG5CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	**TSC_IOG6CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004C	**TSC_IOG7CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

21 真随机数发生器 (RNG)

21.1 简介

RNG 是一个真随机数发生器，可向应用程序提供作为 32 位采样的全熵输出，它由一个实时熵源（模拟）和一个内部调节组件构成。

RNG 可作为一个实时熵源用来构建符合 NIST 要求的确定性随机位发生器 (DRBG)。

RNG 真随机数发生器已按照德国 BSI AIS-31（T0 到 T8）统计测试进行了测试。

21.2 RNG 主要特性

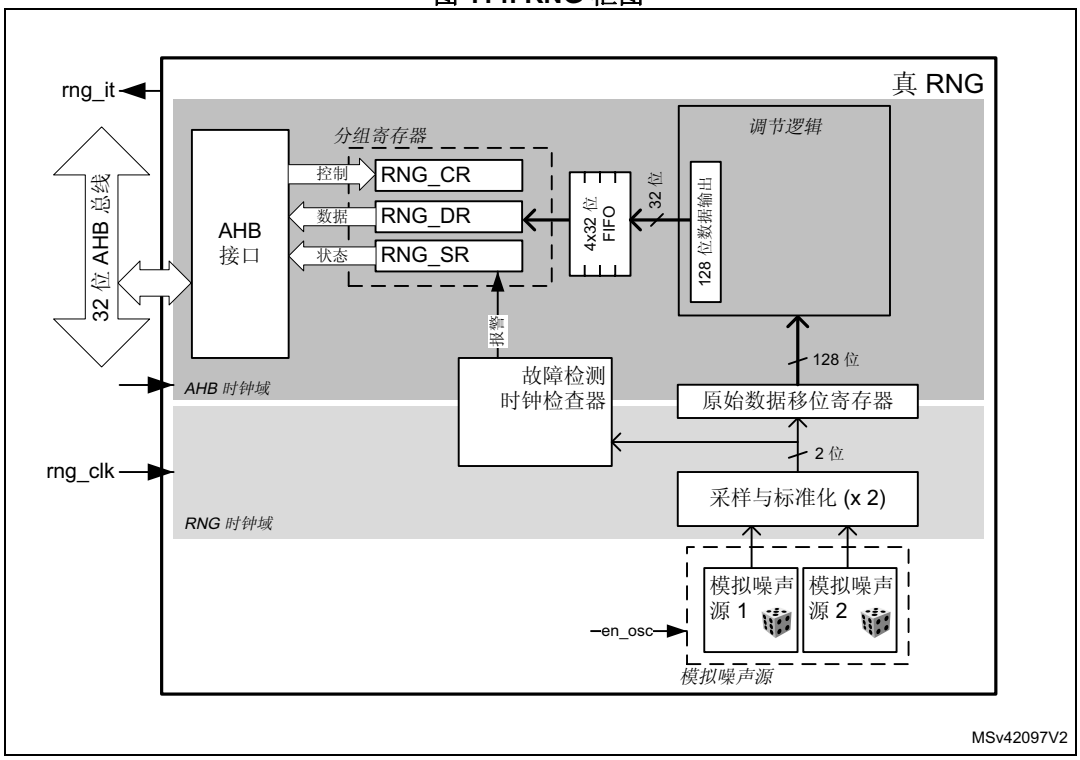
- RNG 提供的 32 位真随机数由模拟熵源生成并通过高质量调节级进行处理。
- 如果值大于 213 个周期（否则为 213 个周期），则每 16x $\frac{f_{AHB}}{f_{RNG}}$ AHB 个时钟周期生成四个 32 位随机采样。
- 支持内置连续基本运行状态测试及相关错误管理。
  - 包括过低采样时钟检测和重复计数测试。
- 可被禁止以降低功耗。
- 它具有 AMBA® AHB 从外设，只能通过 32 位字进行单次访问（否则会生成 AHB 总线错误，并会忽略写访问）。

21.3 RNG 功能说明

21.3.1 RNG 框图

图 114 显示了 RNG 框图。

图 114. RNG 框图



21.3.2 RNG 内部信号

表 104 中列出了 RNG 级而非 STM32 产品级（焊盘上）所提供的内部信号，对这些信号有所了解会很有用。

表 104. RNG 内部输入/输出信号

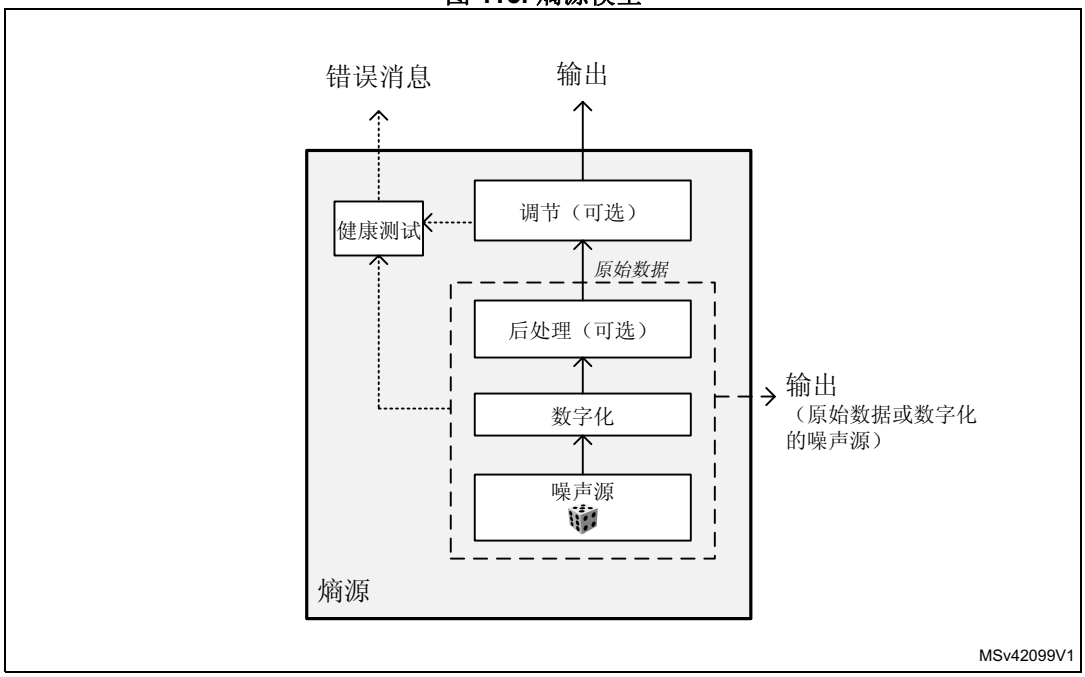
信号名称	信号类型	说明
rng_it	数字输出	RNG 全局中断请求
rng_hclk	数字输入	AHB 时钟
rng_clk	数字输入	RNG 专用时钟，与 rng_hclk 异步

21.3.3 随机数生成

真随机数发生器 (RNG) 按确定的间隔通过其 AHB 接口提供真随机数。RNG 可实现*图 115*所示的熵源模型，并为应用程序提供三个主要功能：

- 采集熵源的位串输出
- 获取噪声源采样，用于验证目的
- 采集连续运行状态测试得到的错误消息

图 115. 熵源模型



RNG 的主要组件包括：

- 物理随机源（模拟噪声源）
- 模拟噪声源的数字化处理级
- 模拟噪声源的加密处理级
- 输出缓冲区，用于缓冲熵源输出和噪声源采样
- 运行状态监视模块，用于对整个熵源执行测试

下文对所有组件进行了详细介绍。

噪声源：

噪声源组件包含不确定性的、能够提供熵值的活动，其最终决定了由熵源产生的输出位串的不确定性。它包括：

- 两个模拟噪声源，每个噪声源由三个自由运行的环形振荡器输出异或 (XOR) 而成。可以禁止这些模拟振荡器以实现节能，如*第 21.3.8 节：RNG 低功耗使用*所述。
- 这些输出的采样级由专用时钟输入 (**rng_clk**) 提供时钟信号，输出 2 位原始数据。

另外，该噪声源采样与 AHB 接口时钟频率 (**rng_hclk**) 无关。

注： *第 21.6 节：RNG 熵源验证中提供了建议使用的 RNG 时钟频率。*

后期处理

从真随机噪声源获取的采样值由 2 位的位串组成。由于该噪声源输出有偏差，因此 RNG 会利用后期处理组件将偏差降至可接受的水平。

更具体来讲，对于每个噪声源产生的 2 位数值,一位是由 RNG 取自采样噪声源,另一位是对取自采样噪声源的值取反。因此，如果噪声源生成的“1”多于“0”（或者相反），则噪声源会被过滤掉。

调节

RNG 中的调节组件是确定性函数，可以提高生成固定长度位串输出（128 位）的熵率。

注： *第 21.5 节：RNG 处理时间中介绍了 RNG 初始化过程中的延时。*

另外请注意，如果已经接收了 32 或更多位原始数据，并且输出 FIFO 需要重新填充，则会触发后期处理计算。因此，当 RNG 128 位 FIFO 在 64 个 RNG 时钟周期后被应用程序清空时，RNG 输出熵为最大值。

*第 21.5 节：RNG 处理时间*中给出了生成两个随机数之间和 RNG 初始化之间所需的时间，以及第一个采样的可用性的说明。

调节组件由更快的 AHB 时钟提供时钟信号。

输出缓冲区

数据输出缓冲区最多可存储四个从调节组件输出的 32 位字。如果已通过 RNG_DR 寄存器从输出 FIFO 读取了四个字，则 128 位调节输出寄存器的内容会被推送到输出 FIFO 中，同时自动启动新的一轮调节。213 个 AHB 时钟周期后，会将四个新字添加到调节输出寄存器。

当通过 RNG_DR 寄存器可获取随机数时，DRDY 标志就从“0”变为“1”。该标志会保持在置 1 状态，直至从 RNG_DR 寄存器读取四个字后输出缓冲区变空。

注： *如果使能了中断，则当该数据就绪标志从“0”变为“1”时，会产生中断。随后，中断会自动由 RNG 清零，如上所述。*

运行状态检查

该组件可确保整个熵源（包括其噪声源）正常启动并按预期运行，能够快速准确地定位故障，从而提供了高可靠性保证。

RNG 可对 RNG_HCTR 寄存器（请参见*第 21.6.2 节*）的推荐值执行以下运行状态检查功能：

1. 连续运行状态测试，无限期地在噪声源输出端运行
   - 重复计数测试，在以下情况时会指示发生错误：
   a) 其中一个噪声源持续不变地输出了 64 位或以上的“0”或“1”，或者 32 个或以上的“01”或“10”。
   b) 两个噪声源都持续不变地输出了 32 位或以上的“0”或“1”，或者 16 个或以上的“01”或“10”。
2. 厂商专用连续测试
   - 实时“过慢”采样时钟检测器，如果一个 RNG 时钟周期小于 AHB 时钟周期的 32 分之一，则该检测器会指示错误。

RNG_SR 寄存器中的 CECS 和 SECS 状态位会在检测到错误条件时进行指示，*第 21.3.7 节：错误管理*对此进行了详细说明。

注： *检测到错误时生成中断。*

21.3.4 RNG 初始化

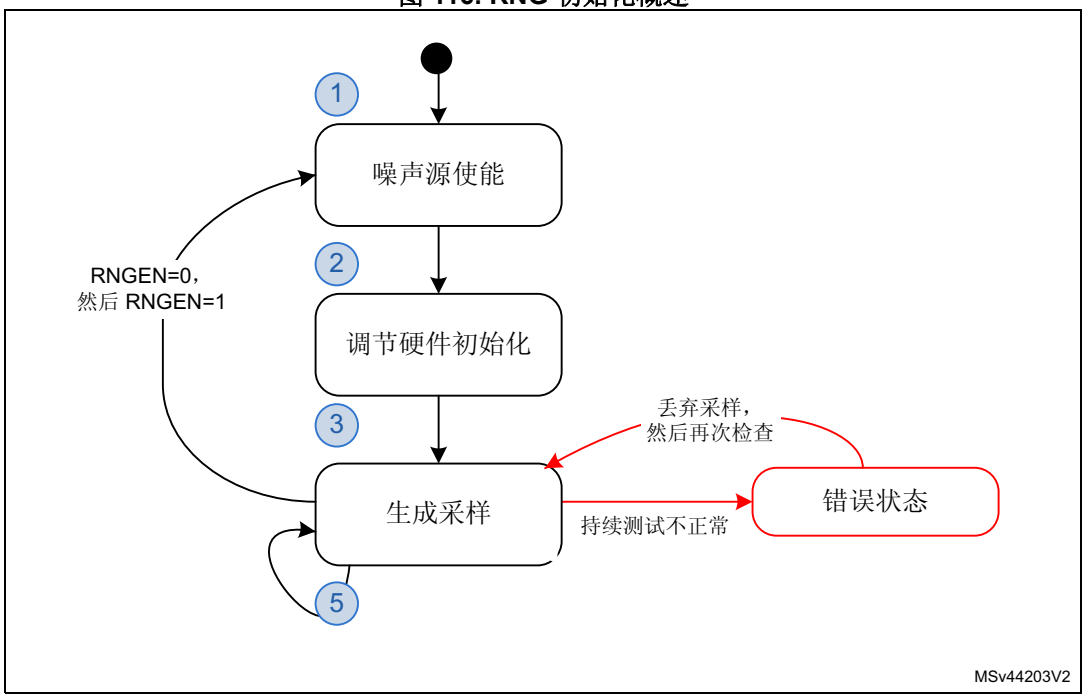
RNG 简化状态机如*图 116* 所示。

在使能 RNG（RNG_CR 中的 RNGEN=1）后，会发生以下一系列事件：

1. 使能模拟噪声源，逻辑系统立即开始采样模拟输出并填充 128 位调节移位寄存器。
2. 使能调节逻辑并且使用两个 128 噪声源位初始化后期处理上下文。
3. 再次用 128 位数据填充调节级内部输入数据缓冲区，并执行一轮调节。然后使用后期处理结果填充输出缓冲区。
4. 输出缓冲区会按照 RNG 使用情况自动重填。

相关初始化时间请参见*第 21.5 节：RNG 处理时间*。

图 116. RNG 初始化概述



21.3.5 RNG 操作

正常工作

要使用中断运行 RNG，建议执行以下步骤：

1. 通过将 RNG_CR 寄存器中的 IE 位置 1 来使能中断。同时通过将 RNGEN 位置 1 来使能 RNG。
2. 这以后，准备好随机数时或出现错误时就产生中断。因此，每次发生中断时，应检查：
   - 未发生错误。RNG_SR 寄存器中的 SEIS 和 CEIS 位应设为 0。
   - 随机数准备就绪。RNG_SR 寄存器中的 DRDY 位必须置 1。
   - 如果满足以上两个条件，最多可连续四次读取 RNG_DR 寄存器的内容。如果调节输出缓冲区中存在有效数据，则应用程序可以额外读取四个字（此时 DRDY 位仍保持为 1）。如果上述条件中有一个或两个都无法满足，则不得读取 RNG_DR 寄存器。如果发生错误，则应使用*第 21.3.7 节*中介绍的错误恢复序列。

要在轮询模式下运行 RNG，建议执行以下步骤：

1. 通过将 RNG_CR 寄存器中的 RNGEN 位置“1”使能随机数生成。
2. 读取 RNG_SR 寄存器并检查：
   – 未发生错误（SEIS 和 CEIS 位应设为 0）
   – 随机数准备就绪（DRDY 位应置 1）
3. 如果满足以上两个条件，最多可连续四次读取 RNG_DR 寄存器的内容。如果调节输出缓冲区中存在有效数据，则应用程序可以额外读取四个字（此时 DRDY 位仍保持为 1）。如果上述条件中有一个或两个都无法满足，则不得读取 RNG_DR 寄存器。如果发生错误，则应使用*第 21.3.7 节*中介绍的错误恢复序列。

注： *如果数据未就绪（DRDY = “0”），则 RNG_DR 会返回 0。*

低功耗运行

如果应用程序比较关注功耗，则可以使用低功耗策略，如*第 21.3.8 节：RNG 低功耗使用*所述。

软件后期处理

无需进行特定的软件后期处理/调节即可满足 AIS-31 认证。如果需要安全强度为 128 位、符合 NIST 要求的 DRBG，则必须基于 RNG 真随机数发生器构建符合要求的随机数发生器软件。

*第 21.3.3 节：随机数生成*介绍了内置的运行状态检查功能。

21.3.6 RNG 时钟

RNG 在两个不同的时钟上运行：AHB 总线时钟和专用 RNG 时钟。

AHB 时钟用于为 AHB 分组寄存器和调节组件提供时钟信号。RNG 时钟用于噪声源采样。*第 21.6 节：RNG 熵源验证*中详细介绍了建议使用的时钟配置。

注： *当 RNG_CR 寄存器中的 CED 位置“0”，那么 RNG 时钟频率应大于 AHB 时钟频率的 32 分之一，否则时钟检查器会始终指示时钟错误（RNG_SR 寄存器中的 CECS = 1）。*

详细信息参见*第 21.3.1 节：RNG 框图*（AHB 和 RNG 时钟域）。

21.3.7 错误管理

运行状态检查模块在生成随机数的同时还验证噪声源行为以及 RNG 源时钟频率是正常的，具体说明见本部分。另外，还描述了相关的错误状态。

时钟错误检测

如果时钟错误检测已使能 (CED = 0) 且 RNG 时钟频率过低，RNG 会将 **CEIS** 和 **CECS** 位都置“1”，指示发生了时钟错误。在这种情况下，应用程序应检查 RNG 时钟是否配置正确（参见*第 21.3.6 节：RNG 时钟*），随后必须将中断标志 CEIS 位清零。当时钟状态正常时，CECS 位自动清零。

注： *时钟错误对生成的随机数没有影响，即应用程序仍然可以读取 RNG_DR 寄存器。*

仅当 RNG 将 CECS 置“1”时，CEIS 才会置 1。

噪声源错误检测

如果发生噪声源（或种子）错误，RNG 会停止生成随机数，并会将 **SEIS** 和 **SECS** 位都置“1”，指示发生了种子错误。如果 RNG_DR 寄存器中有可用值，不能使用该值，因为它可能没有足够的熵。如果在初始化阶段检测到错误，则 RNG 将自动重启整个初始化序列。

要从 RNG 初始化后的种子错误中完全恢复，应使用以下序列：

1. 向 SEIS 位写入“0”，将此位清零。
2. 从 RNG_DR 寄存器读取 12 个字，并将每个字丢弃，将管道清空。
3. 确认 SEIS 仍处于清零状态。随机数生成恢复正常。

21.3.8 RNG 低功耗使用

考虑到功耗问题，可在 DRDY 位置“1”后立即禁止 RNG，具体方法为将 RNG_CR 寄存器中的 RNGEN 位设为“0”。由于 RNGEN =“0”时后期处理逻辑和输出缓冲区仍处于工作状态，因此软件可使用以下功能：

- 如果输出缓冲区中存在有效字，仍可从 RNG_DR 寄存器中读取四个随机数。
- 如果调节输出内部寄存器中存在有效位，仍可从 RNG_DR 寄存器中额外读取四个随机数。如果不属于以上两种情况，则必须由应用程序重新使能 RNG，直至从噪声源采集了至少 32 个新的位并且完成了一轮完整的调节。这对应于 16 个 RNG 时钟周期来采样新的位，216 个 AHB 时钟周期运行一轮调节。

禁止 RNG 时，用户会禁用所有模拟种子发生器，各发生器的功耗列于器件手册中的电气特性部分。用户还会对所有由 RNG 时钟提供时钟信号的逻辑系统进行门控。请注意，由于 RNG 需要一定的时间进行初始化，因此该策略会在一段延时后才可在 RNG_DR 寄存器中提供随机采样。

如果 RNG 模块在初始化期间（也就是刚好在 DRDY 位首次置 1 之前）被禁止，则初始化序列将从 RNGEN 位置“1”时停止的位置恢复运行。

21.4 RNG 中断

在 RNG 中，发生以下事件时会产生中断：

- 数据就绪标志
- 种子错误，请参见*第 21.3.7 节：错误管理*
- 时钟错误，请参见*第 21.3.7 节：错误管理*

可以使用专用的中断使能控制位，如*表 105* 所示。

表 105. RNG 中断请求

中断缩写	中断事件	事件标志	使能控制位	中断清除方法
RNG	数据就绪标志	DRDY	IE	无（自动）
	种子错误标志	SEIS	IE	向 SEIS 写入 0，或向 CONDRST 写入 1 然后写入 0
	时钟错误标志	CEIS	IE	向 CEIS 写入 0

用户可以通过更改 RNG_CR 寄存器中的屏蔽位或通用中断控制位 IE 的方式分别使能或禁止上述中断源。可以从 RNG_SR 寄存器中读取各个中断源的状态。

注： *仅当 RNG 已使能时，才会产生中断。*

21.5 RNG 处理时间

如果值大于 213 个周期（否则为 213 个周期），则每 16x $\frac{f_{AHB}}{f_{RNG}}$ 个时钟周期，调节级可以生成四个 32 位随机数。

设备退出复位状态后需要更长的时间才能生成第一组随机数（请参见*第 21.3.4 节：RNG 初始化*）。事实上，首次使能 RNG 后，会在以下时间后首次提供随机数据：

- 128 RNG 个时钟周期 + 426 个 AHB 周期（$f_{AHB} < f_{threshold}$ 时）
- 192 RNG 个时钟周期 + 213 个 AHB 周期（$f_{AHB} \geq f_{threshold}$ 时）

其中 $f_{threshold} = (213 \times f_{RNG})/ 64$

21.6 RNG 熵源验证

21.6.1 简介

为了评估 RNG 可提供的熵大小，意法半导体按照德国 BSI AIS-31 统计测试（T0 到 T8）对外设进行了测试。测试结果可根据需要提供，客户也可以重现测量结果。

21.6.2 验证条件

意法半导体在以下条件下对 RNG 真随机数发生器进行了测试：

- RNG 时钟 rng_clk= 48 MHz（RNG_CR 寄存器中的 CED 位 =“0”）且 rng_clk = 400 kHz（RNG_CR 寄存器中的 CED 位 =“1”）。

21.6.3 数据采集

为了运行统计测试，需要对熵源的原始数据和输出进行采样。

如果需要为产品检索上述采样，请联系意法半导体。

21.7 RNG 寄存器

RNG 与控制寄存器、数据寄存器和状态寄存器相关联。

21.7.1 RNG 控制寄存器 (RNG_CR)

RNG control register

偏移地址：0x000

复位值：0x0000 0000

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

位 31:6 保留，必须保持复位值。

位 5 **CED**：时钟错误检测 (Clock error detection)

0：使能时钟错误检测

1：禁止时钟错误检测

当使能 RNG 时，不能实时使能或禁止时钟错误检测，即必须禁止 RNG 才能使能或禁止 CED。

位 4 保留，必须保持复位值。

位 3 **IE**：中断使能 (Interrupt enable)

0：禁止 RNG 中断。

1：使能 RNG 中断。当 RNG_SR 寄存器中 DRDY = “1”、SEIS = “1”或 CEIS = “1”，则中断处于挂起状态。

位 2 **RNGEN**：真随机数发生器使能 (True random number generator enable)

0：禁止真随机数发生器。模拟噪声源关闭，并会对由 RNG 时钟提供时钟信号的逻辑系统进行门控。

1：使能真随机数发生器。

位 1:0 保留，必须保持复位值。

21.7.2 RNG 状态寄存器 (RNG_SR)

RNG status register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEIS	CEIS	Res.	Res.	SECS	CECS	DRDY
									rc_w0	rc_w0			r	r	r

位 31:7 保留，必须保持复位值。

位 6 **SEIS**：种子错误中断状态 (Seed error interrupt status)
此位与 SECS 同时置 1，通过写入 0 来清零。写入 1 无效。
0：未检测到错误序列。
1：至少检测到一个错误序列。有关详细信息，请参见 **SECS** 位说明。
如果 RNG_CR 寄存器中 IE = 1，则会挂起中断。

位 5 **CEIS**：时钟错误中断状态 (Clock error interrupt status)
此位与 CECS 同时置 1，通过写入 0 来清零。写入 1 无效。
0：RNG 时钟正确 (fRNGCLK > fHCLK/32)
1：已检测到 RNG 时钟过慢 (fRNGCLK< fHCLK/32)
如果 RNG_CR 寄存器中 IE = 1，则会挂起中断。

位 4:3 保留，必须保持复位值。

位 2 **SECS**：当前状态种子错误 (Seed error current status)
0：目前未检测到错误序列。如果 SEIS 位置 1，则意味着已检测到错误序列并已恢复正常。
1：至少检测到一个以下错误序列：
– 其中一个噪声源持续不变地输出了 64 位或以上的“0”或“1”，或者 32 个或以上的“01”或“10”。
– 两个噪声源都持续不变地输出了 32 位或以上的“0”或“1”，或者 16 个或以上的“01”或“10”。

位 1 **CECS**：当前状态时钟错误 (Clock error current status)
0：RNG 时钟正确 (fRNGCLK > fHCLK/32)。如果 CEIS 位置 1，则意味着已检测到时钟过慢并已恢复正常。
1：RNG 时钟过慢 (fRNGCLK < fHCLK/32)。
注： 只有 RNG_CR 寄存器中的 CED 位置 0 时，CECS 位才有效。

位 0 **DRDY**：数据就绪 (Data Ready)
0：RNG_DR 寄存器尚未有效，无可用随机数据。
1：RNG_DR 寄存器包含有效随机数据。
在输出缓冲区变空之后（读取 RNG_DR 寄存器之后），在生成新的随机值之前，此位返回 0。
注： 当禁止外设时（RNG_CR 寄存器中的 RNGEN = 0），可以将 DRDY 位置 1。
如果 RNG_CR 寄存器中的 IE = 1，则在 DRDY = 1 时生成中断。

21.7.3 RNG 数据寄存器 (RNG_DR)

RNG data register

偏移地址：0x008

复位值：0x0000 0000

RNG_DR 寄存器是只读寄存器，在读取时提供 32 位随机数值。读取该寄存器后，如果输出 FIFO 为空，则该寄存器将在 216 个 AHB 时钟周期之后提供一个新的随机值。

当 DRDY = 1 时，即使 RNGEN = 0，该寄存器的内容也是有效的。

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

位 31:0 **RNDATA[31:0]**：随机数据 (Random data)

当 DRDY = 1 时 32 位随机数据有效。DRDY = 0 时，RNDATA 值为 0。

21.7.4 RNG 寄存器映射

表 106 给出了 RNG 寄存器映射和复位值。

表 106. RNG 寄存器映射和复位值

偏移	寄存器名称	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
0x000	RNG_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CED	Res.	IE	RNGEN	Res.	Res.
	Reset value																											0		0	0		
0x004	RNG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEIS	CEIS	Res.	Res.	SECS	CECS	DRDY
	Reset value																										0	0			0	0	0
0x008	RNG_DR	RNDATA[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

22 AES 硬件加速器 (AES)

22.1 简介

AES 硬件加速器 (AES) 使用完全符合联邦信息处理标准 (FIPS) 出版物 197 中规定的高级加密标准 (AES) 的算法来对数据进行加密或解密。

支持多种链接模式（ECB、CBC、CTR、GCM、GMAC、CCM），支持 128 或 256 位密钥大小。

AES 加速器为 32 位 AHB 外设。它支持对传入和传出数据进行 DMA 单次传输（需要两个 DMA 通道）。

AES 外设可为 STM32 加密库所实现的 AES 加密算法提供硬件加速。

AES 为 AMBA AHB 从外设，仅支持 32 位字单次访问（否则，会产生 AHB 总线错误）。32 位以外的写入可能损坏寄存器内容。

22.2 AES 主要特性

- 符合自 2001 年 11 月起的 NIST *“高级加密标准 (AES)，FIPS 出版物 197”*
- 128 位数据块处理
- 支持 128 位和 256 位密钥长度
- 基于多种链接模式的加密和解密：
  - 电子密码本 (ECB) 模式
  - 密码分组链接 (CBC) 模式
  - 计数器 (CTR) 模式
  - Galois 计数器模式 (GCM)
  - Galois 消息认证码 (GMAC) 模式
  - CBC-MAC 计数器 (CCM) 模式
- ECB 模式处理 128 位数据块的时候会有延迟，使用 128 位密钥或 256 位密钥分别延迟 51 或 75 个时钟周期
- 集成有密钥调度器，包括密钥分散阶段（仅限 ECB 或 CBC 解密）
- AMBA AHB 从外设，仅支持 32 位字单次访问
- 256 位寄存器，用于存储加密密钥（8 个 32 位寄存器）
- 128 位寄存器，用于存储初始化向量（4 个 32 位寄存器）
- 32 位缓冲器，用于数据输入和输出
- 采用自动数据流控制，支持单次传输直接存储器访问 (DMA)（使用两个通道，分别用于传入数据和已处理数据）
- 采用数据交换逻辑，支持 1 位、8 位、16 位或 32 位数据
- 在 AES 需要处理优先级更高的消息时，可通过软件将当前消息挂起，然后恢复原始消息

22.3 AES 实现

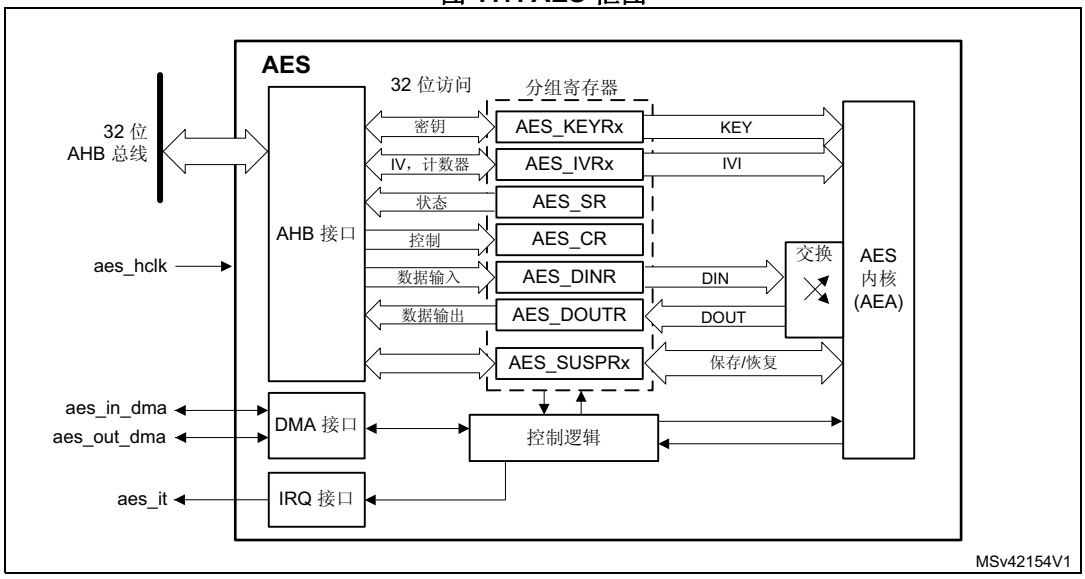
器件有两个 AES 外设实例，AES1 和 AES2。

22.4 AES 功能描述

22.4.1 AES 框图

图 117 显示了 AES 的框图。

图 117. AES 框图



22.4.2 AES 内部信号

表 107 描述了连接 AES 外设的用户相关内部信号。

表 107. AES 内部输入/输出信号

信号名称	信号类型	说明
aes_hclk	数字输入	AHB 总线时钟
aes_it	数字输出	AES 中断请求
aes_in_dma	数字输入/输出	输入 DMA 单次请求/确认
aes_out_dma	数字输入/输出	输出 DMA 单次请求/确认

22.4.3 AES 加密内核

概述

AES 加密内核由以下部分组成：

- AES 算法 (AEA)
- 基于二元 Galois 域的乘法器 (GF2mul)
- 密钥输入
- 初始化向量 (IV) 输入
- 链接算法逻辑（XOR、反馈/计数器、屏蔽）

AES 内核适用于密钥长度为 128 位或 256 位的 128 位数据块（四字）。根据链接模式的不同，AES 可能需要一个 96 位初始化向量 IV（和一个 32 位计数器字段），也可能不需要。

AES 具有以下工作模式：

- **模式 1：**
  使用存储在 AES_KEYRx 寄存器中的密钥实现明文加密。
- **模式 2：**
  ECB 或 CBC 解密密钥准备。在选择 ECB 或 CBC 链接模式的模式 3 之前，必须先使用此模式。准备解密的密钥自动存储在 AES_KEYRx 寄存器中。现在，AES 外设准备好切换到模式 3 以执行数据解密。
- **模式 3：**
  使用存储在 AES_KEYRx 寄存器中的密钥实现密文解密。当选择 ECB 和 CBC 链接模式时，必须通过模式 2 预先准备好密钥。
- 模式 4：
  使用存储在 AES_KEYRx 寄存器中的密钥实现 ECB 或 CBC 密文单次解密（初始密钥自动分散）。

注： *仅在执行 ECB 和 CBC 解密时使用模式 2 和模式 4。*

当选择模式 4 时，只能进行一次解密，因此推荐使用模式 2 和模式 3。

通过对 AES_CR 寄存器中的 MODE[1:0] 位域进行编程来选择工作模式。只能在 AES 外设已禁止时进行。

典型数据处理

AES 的典型用法如*第 569 页的第 22.4.4 节：执行密码操作的 AES 过程*中所述。

注： *中间 AEA 阶段的输出始终不会在加密边界外暴露，IVI 位域除外。*

链接模式

AES 支持以下链接模式，可通过 AES_CR 寄存器的 CHMOD[2:0] 位域选择：

- 电子密码本 (ECB)
- 密码分组链接 (CBC)
- 计数器 (CTR)
- Galois 计数器模式 (GCM)
- Galois 消息认证码 (GMAC)
- CBC-MAC 计数器模式 (CCM)。

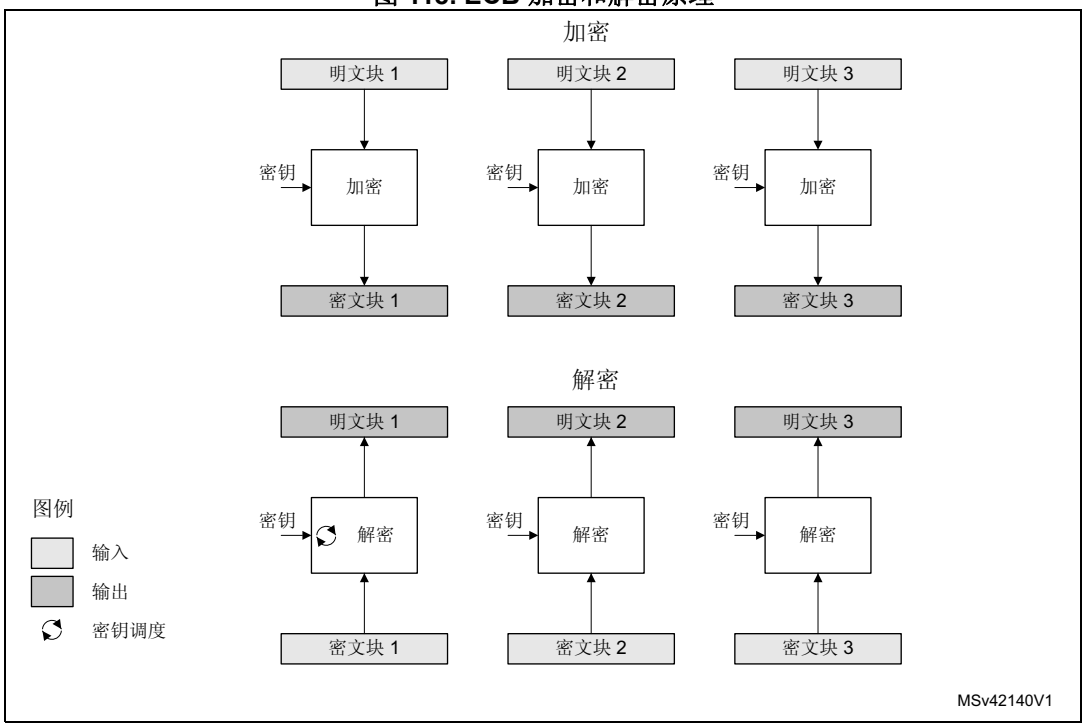
注： *只有在 AES 禁止（AES_CR 寄存器的位 EN 清零）时，才能更改链接模式。*

后续子章节对每个 AES 链接模式的原理进行了介绍。

详细信息请参见从*第 22.4.8 节：AES 基本链接模式（ECB 和 CBC）*开始的专用章节。

电子密码本 (ECB) 模式

图 118. ECB 加密和解密原理

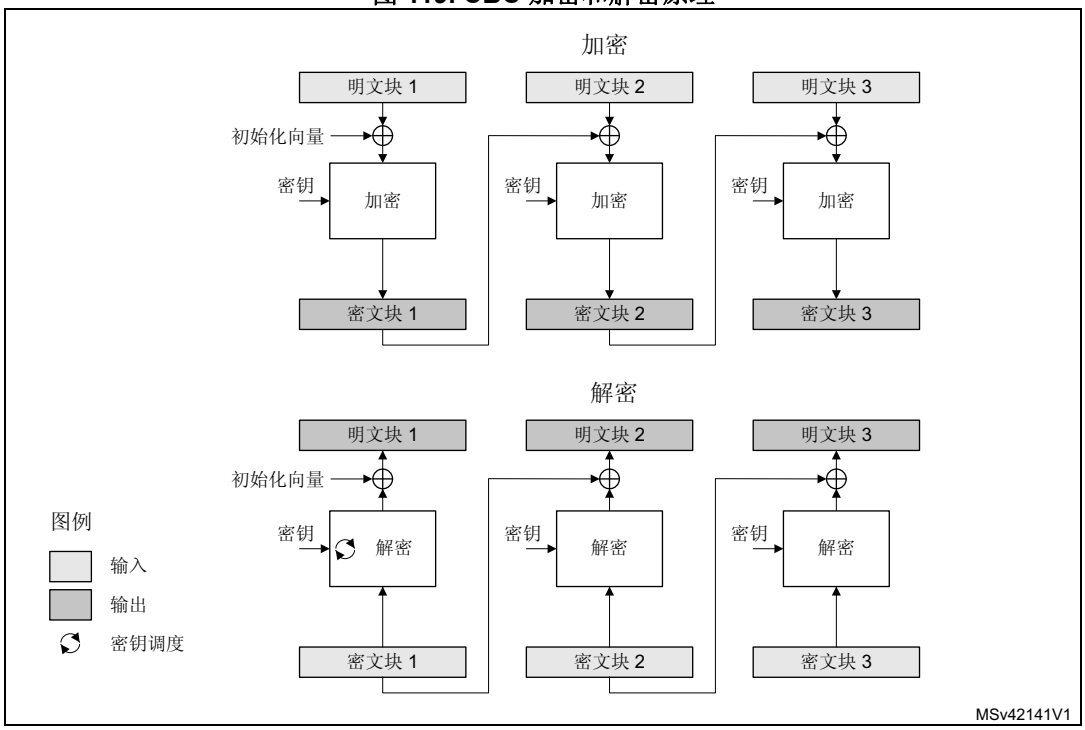


ECB 是最简单的工作模式。无链接操作，且无特殊初始化阶段。消息会划分到各个块中，并对各个块分别进行加密或解密。

注： *对于解密，在处理第一个块之前需要特殊的密钥调度。*

密码分组链接 (CBC) 模式

图 119. CBC 加密和解密原理

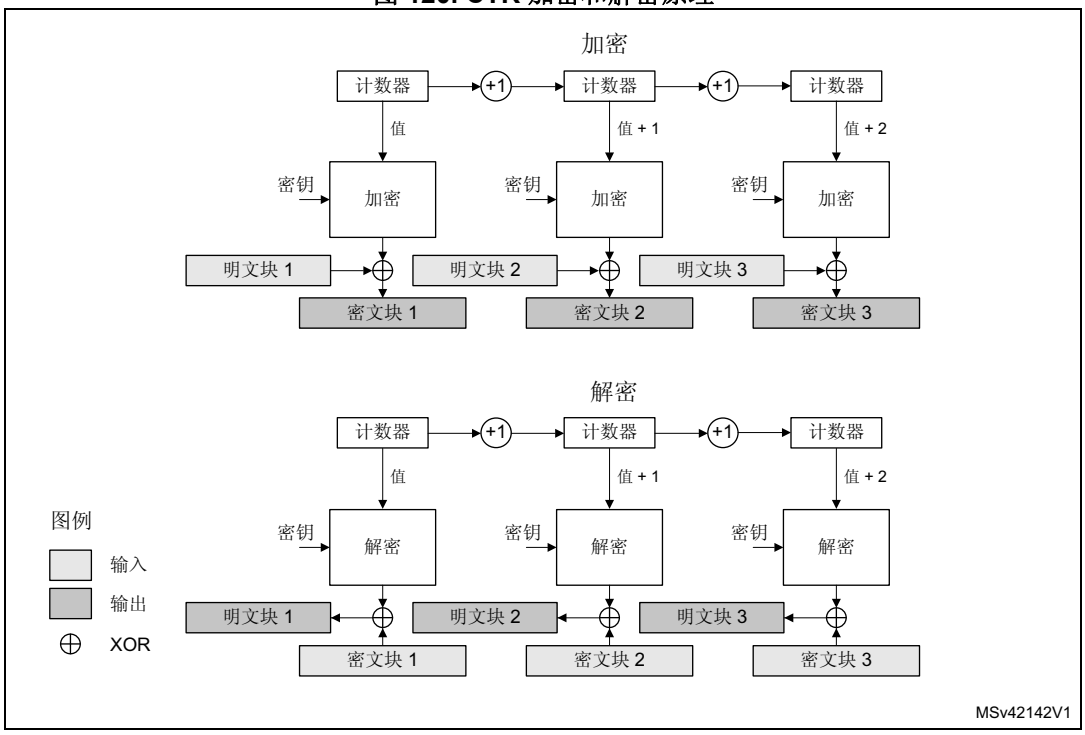


在 CBC 模式下，每个块的输出均与下一个块的输入相连。为了确保每条消息都是唯一的，会在第一个块处理过程中使用初始化向量。

注： *对于解密，在处理第一个块之前需要特殊的密钥调度。*

计数器 (CTR) 模式

图 120. CTR 加密和解密原理

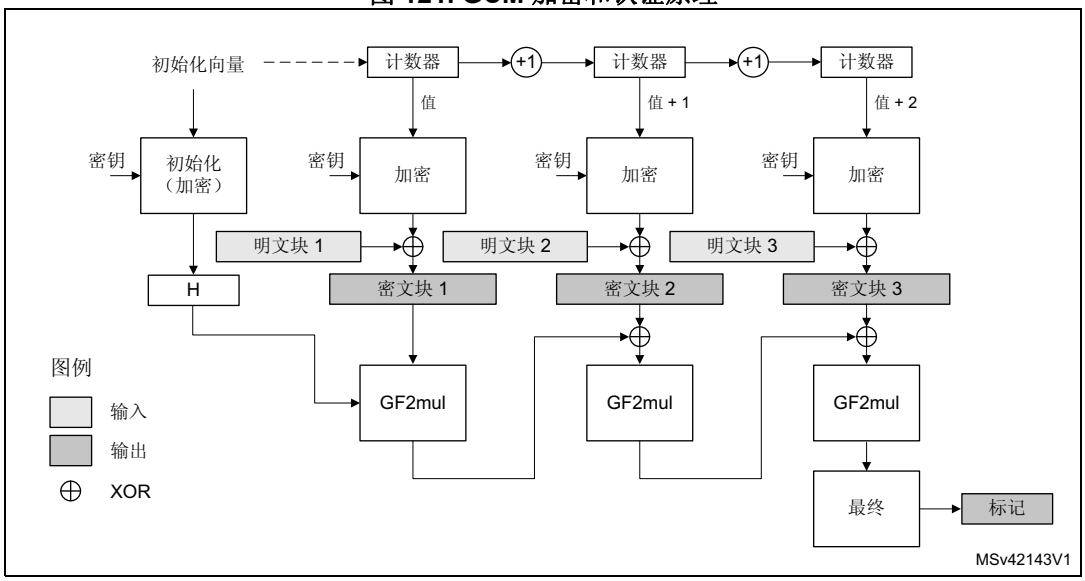


CTR 模式使用 AES 内核生成密匙流。这些密钥将与明文进行逻辑异或运算，以获得密文（如 NIST *特别出版物 800-38A，块密码工作模式的建议*中所述）。

注： *与 ECB 和 CBC 模式不同，CTR 解密无需密钥调度，因为在该链接方案中，AES 内核始终在加密模式下用于生成密钥流或计数器块。*

Galois/计数器模式 (GCM)

图 121. GCM 加密和认证原理

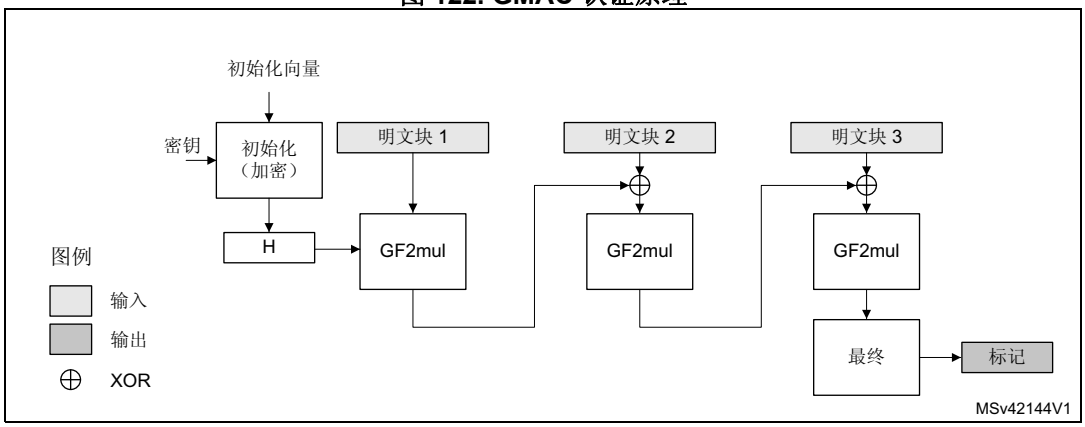


在 Galois/计数器模式 (GCM) 下，将对明文消息进行加密，同时会对消息认证码 (MAC) 进行计算，从而生成相应的密文及其 MAC（也称为认证标记）。NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

GCM 模式基于 AES 计数器模式，可实现机密性。它使用固定有限域乘法器来计算消息认证码。消息末尾处需要一个初始值和一个特定的 128 位块。

Galois 消息认证码 (GMAC) 原理

图 122. GMAC 认证原理

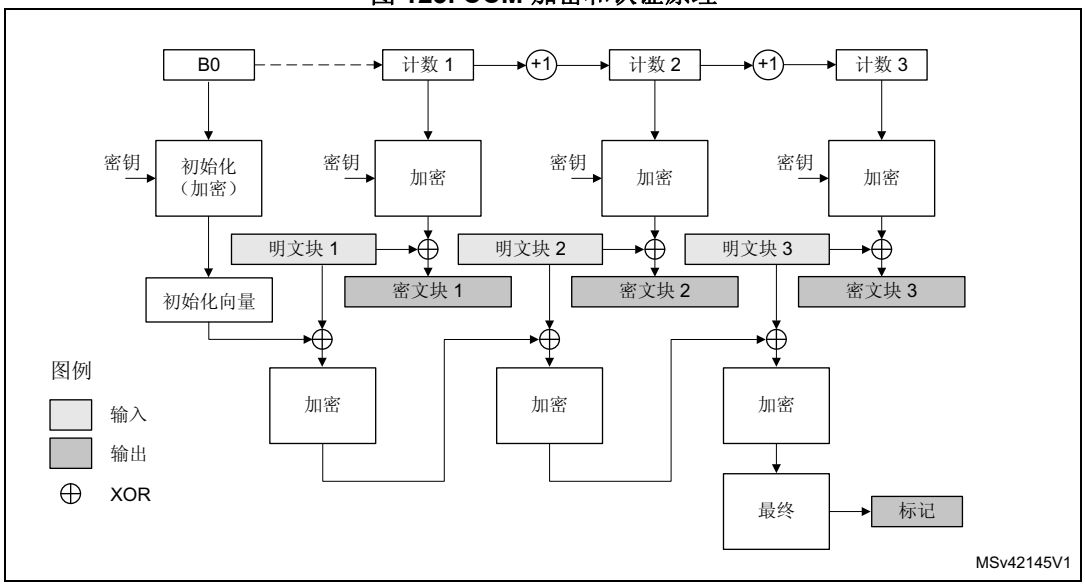


Galois 消息认证码 (GMAC) 允许认证消息并生成相应的消息认证码 (MAC)。NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

GMAC 与 GCM 类似，只不过它应用于仅由明文认证数据组成的消息（即只有标头，无有效负载）。

CBC-MAC 计数器 (CCM) 原理

图 123. CCM 加密和认证原理



在密码块链接-消息认证码计数器 (CCM) 模式下，将对明文消息进行加密，同时会对消息认证码 (MAC) 进行计算，从而生成相应的密文以及相应的 MAC（也称为标记）。NIST *特别出版物 800-38C，块密码工作模式的建议 - CCM 模式实现认证和机密性*中对此进行了介绍。

CCM 模式基于 AES 计数器模式，可实现机密性，并且使用 CBC 来计算消息认证码。它需要一个初始值。

与 GCM 类似，CCM 链接模式可应用于仅由明文认证数据组成的消息（即只有标头，无有效负载）。请注意，这种 CCM 使用方式不被称为 CMAC（它并非类似于 GCM/GMAC），NIST 不推荐这种用途。

22.4.4 执行密码操作的 AES 过程

简介

下面对典型密码操作进行了介绍。详细信息请参见从*第 22.4.8 节：AES 基本链接模式（ECB 和 CBC）*开始的章节。

图 124 和*图 125* 中的流程图介绍了 STM32 加密库如何实现 AES 算法。AES 可在 ECB、CBC、CTR、CCM 和 GCM 工作模式下加速执行 AES-128 和 AES-256 加密算法。

注： *有关加密库的更多详细信息，请参见 UM1924 用户手册“STM32 加密库”（可从 www.st.com 获取）。*

图 124. STM32 加密库 AES 流程图示例

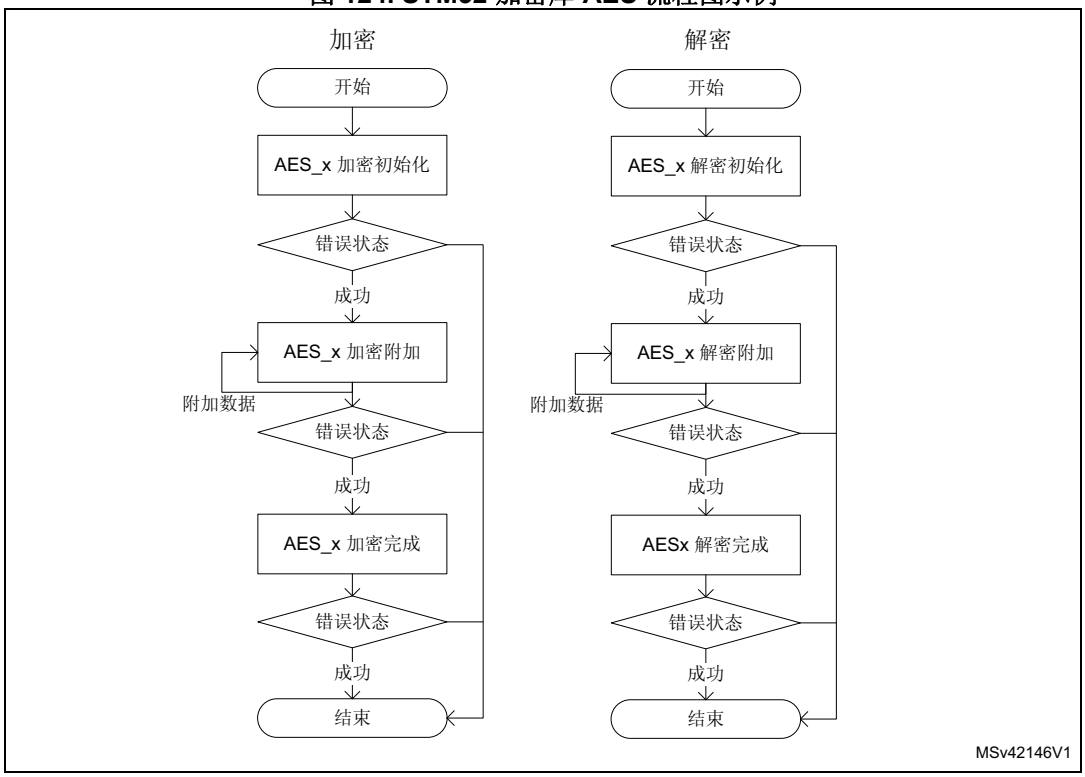
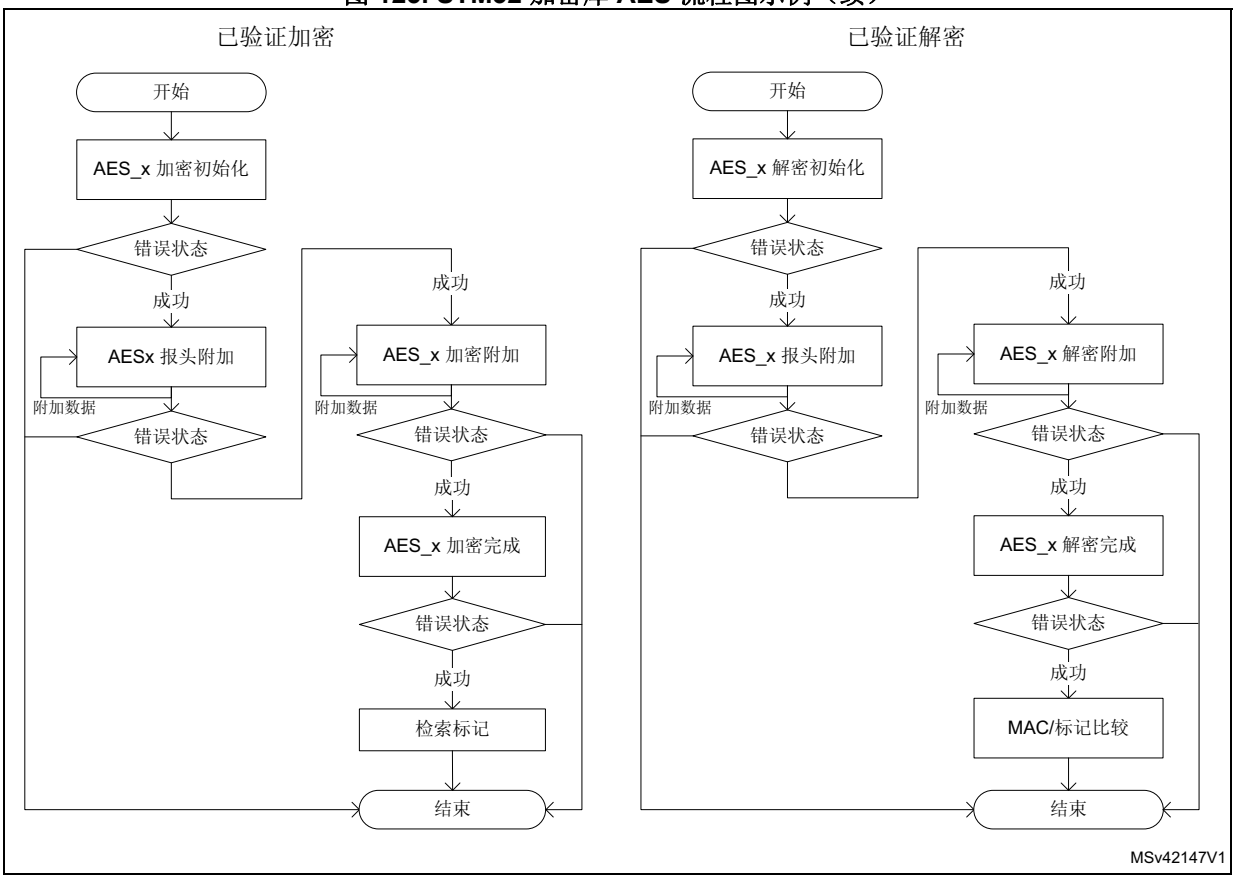


图 125. STM32 加密库 AES 流程图示例（续）



初始化 AES

要初始化 AES，首先通过将 AES_CR 寄存器中的 EN 位清零来禁止它。然后以任何顺序执行以下步骤：

- 通过对 AES_CR 寄存器中的 MODE[1:0] 位域进行编程来配置 AES 模式。
  - 对于加密，必须选择模式 1 (MODE[1:0] = 00)。
  - 对于解密，必须选择模式 3 (MODE[1:0] = 10)，除非使用 ECB 或 CBC 链接模式。在后一种情况下，必须执行加密密钥的初始密钥分散，如*第 22.4.5 节：AES 解密密钥准备*中所述。
- 通过对 AES_CR 寄存器中的 CHMOD[2:0] 位域进行编程来选择链接模式。
- 使用 AES_CR 寄存器中的 KEYSIZE 位域配置密钥大小（128 位或 256 位）。
- 将对称密钥写入 AES_KEYRx 寄存器（4 或 8 个寄存器，具体取决于密钥大小）。
- 使用 AES_CR 寄存器中的 DATATYPE[1:0] 位域配置数据类型（1 位、8 位、16 位或 32 位）。
- 必要时（例如，CBC 或 CTR 链接模式），向 AES_IVRx 寄存器中写入初始化向量。

附加数据

本节介绍附加数据进行处理的不同方式，其中要处理数据的大小不是 128 位的倍数。

有关 ECB 或 CBC 模式，请参见*第 22.4.6 节：AES 密文窃取和数据填充*。这些情况下的最后一个块管理比本节描述序列下的更复杂。

通过轮询附加数据

此方法按照以下序列使用标志轮询来控制附加数据：

1. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
2. 重复以下子序列，直到完全处理有效负载：
   a) 将四个输入数据字写入 AES_DINR 寄存器。
   b) AES_SR 中的状态标志 CCF 置 1 后，从 AES_DOUTR 寄存器读取四个数据字。
   c) 通过将 AES_CR 寄存器中的 CCFC 位置 1，将 CCF 标志清零。
   d) 如果刚刚处理的数据块是消息的倒数第二个块，并且要处理的最后一个块中的有效数据不足 128 位，则用零填充最后一个块的剩余部分，并且在 GCM 有效负载加密或 CCM 有效负载解密的情况下，使用 AES_CR 寄存器的 NPBLB 位域指定无效字节的数量，以便 AES 计算正确的标记。
3. 丢弃不属于有效负载的数据，然后通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。

注： *在对 AES_DINR 寄存器的两次连续写入之间自动插入最多三个等待状态，以便将密钥发送到 AES 处理器。*

NPBLB 不用于 GCM、GMAC 和 CCM 链接模式的标头阶段。

使用中断附加数据

此方法通过以下序列，使用 AES 外设中断来控制附加数据：

1. 通过将 AES_CR 寄存器中的 CCFIE 位置 1 来使能 AES 中断。
2. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
3. 将前四个输入数据字写入 AES_DINR 寄存器。
4. 中断时处理 AES 中断服务程序中的数据：
   a) 从 AES_DOUTR 寄存器读取四个输出数据字。
   b) 通过将 AES_CR 寄存器中的 CCFC 位置 1，将 CCF 标志和挂起中断清零。
   c) 如果刚刚处理的数据块是消息的倒数第二个块，并且要处理的最后一个块中的有效数据不足 128 位，则用零填充最后一个块的其余部分，并且在 GCM 有效负载加密或 CCM 有效负载解密的情况下，使用 AES_CR 寄存器的 NPBLB 位域指定无效字节的数量，以便 AES 计算正确的标记。然后继续进行 4e)。
   d) 如果刚刚处理的数据块是消息的最后一块，则丢弃不属于有效负载的数据，然后通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设，并退出中断服务程序。
   e) 将接下来的四个输入数据字写入 AES_DINR 寄存器，并退出中断服务程序。

注： *AES 允许连续读取或写入操作之间存在延迟，例如在两次 AES 计算之间要响应的另一个外设的中断。*

NPBLB 不用于 GCM、GMAC 和 CCM 链接模式的标头阶段。

使用 DMA 附加数据

借助此方法，所有传输和处理过程均由 DMA 和 AES 管理。要使用此方法，请按照以下步骤操作：

1. 用零填充块的其余部分来准备最后四个字的数据块（如果要处理的数据没有完全填充它）。
2. 将 DMA 控制器配置为将来自存储器的要处理的数据传输到 AES 外设输入以及将来自 AES 外设输出的已处理数据传输到存储器，如*第 22.4.16 节：AES DMA 接口*所述。配置 DMA 控制器，以在传输完成时产生中断。在 GCM 有效负载加密或 CCM 有效负载解密的情况下，如果用零填充了最后一个四字块，DMA 必须不包括最后一个四字块。必须为此最后一个块使用*通过轮询附加数据*中描述的序列，因为处理块之前必须设置 NPBLB 位，以便 AES 计算正确的标记。
3. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
4. 将 AES_CR 寄存器中的 DMAINEN 和 DMAOUTEN 位置 1，以使能 DMA 请求。
5. 当 DMA 中断指示传输完成时，从存储器中获取 AES 处理过的数据。

注： *CCF 标志在这种方法中没有作用，因为读取 AES_DOUTR 寄存器是通过 DMA 自动管理的，在计算过程结束时没有任何软件操作。*

NPBLB 不用于 GCM、GMAC 和 CCM 链接模式的标头阶段。

22.4.5 AES 解密密钥准备

对于 ECB 或 CBC 解密，第一轮解密的密钥必须从最后一轮加密的密钥中分散。因此，在执行解密之前需要完整的加密密钥计划。除 ECB 或 CBC 模式外的 AES 解密**不需要**密钥准备。

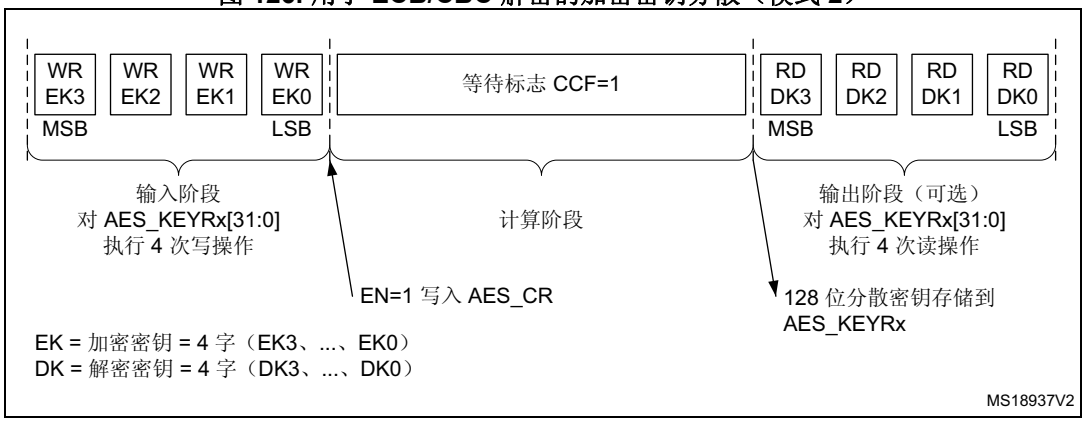
推荐方法是通过将 AES_CR 的 MODE[1:0] 位域设置为 01 来选择模式 2（仅密钥处理），然后通过将 MODE[1:0] 设置为 10 来继续解密（模式 3，仅解密）。模式 2 的使用说明如下：

1. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。
2. 通过将 AES_CR 的 MODE[1:0] 位域设置为 01 来选择模式 2。由于此密钥分散模式独立于所选的链接算法，因此 CHMOD[2:0] 位域在这种情况下并不重要。
3. 通过 AES_CR 寄存器的 KEYSIZE 位，将密钥长度设置为 128 或 256 位。
4. 将加密密钥写入 AES_KEYRx 寄存器（128 或 256 位），如*图 126* 所示。对 AES_IVRx 寄存器执行写入操作不起作用。
5. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
6. 等待 AES_SR 寄存器中的 CCF 标志置 1。
7. 将 CCF 标志清零。分散的密钥在 AES 内核中，可以用于解密。如果需要，应用程序还可以读取 AES_KEYRx 寄存器以获得分散的密钥，如*图 126* 所示（已处理密钥自动加载到 AES_KEYRx 寄存器中）。

注： *分散密钥可用时，将由硬件禁止 AES。*

要重新开始计算分散密钥，请重复步骤 4、5、6 和 7。

图 126. 用于 ECB/CBC 解密的加密密钥分散（模式 2）



如果软件存储了为解密准备的初始密钥，则对于要用给定密钥解密的所有数据，仅执行一次密钥调度操作就已足够。

注： *密钥准备操作持续 59 或 82 个时钟周期，具体取决于密钥大小（128 或 256 位）。*

注： *备用密钥准备是通过将 AES_CR 寄存器中的 MODE[1:0] 位域设置为 11 来选择模式 4。这种情况下，模式 3 不可用。*

22.4.6 AES 密文窃取和数据填充

在 ECB 或 CBC 模式下使用 AES 来管理不是块大小（128 位）倍数的消息时，请使用密文窃取技术，如 NIST *特别出版物 800-38A，块密码工作模式的建议：CBC 模式密文窃取的三种变型*中介绍的窃取技术。由于 AES 外设不支持此类技术，应用程序必须使用倒数第二个块中的数据完成输入数据的最后一个块。

注： *本参考手册中未对密文窃取技术进行介绍。*

类似地，在除 ECB 或 CBC 之外的模式下使用 AES 时，必须先用零填充不完整的输入数据块（即，输入数据短于 128 位的块），然后再加密（即，必须在数据串的尾端附加额外的位）。解密后必须丢弃额外填充的位。由于 AES 不会对**最后一个块**执行自动数据填充操作，因此应用程序必须遵循*第 569 页的第 22.4.4 节：执行密码操作的 AES 过程*中给出的建议来管理大小不是 128 位倍数的消息。

注： *填充数据根据 AES_CR 寄存器中的 DATATYPE[1:0] 位域，以类似于正常数据的方式进行交换（有关详细信息，请参见第 594 页的第 22.4.13 节：AES 数据寄存器和数据交换）。*

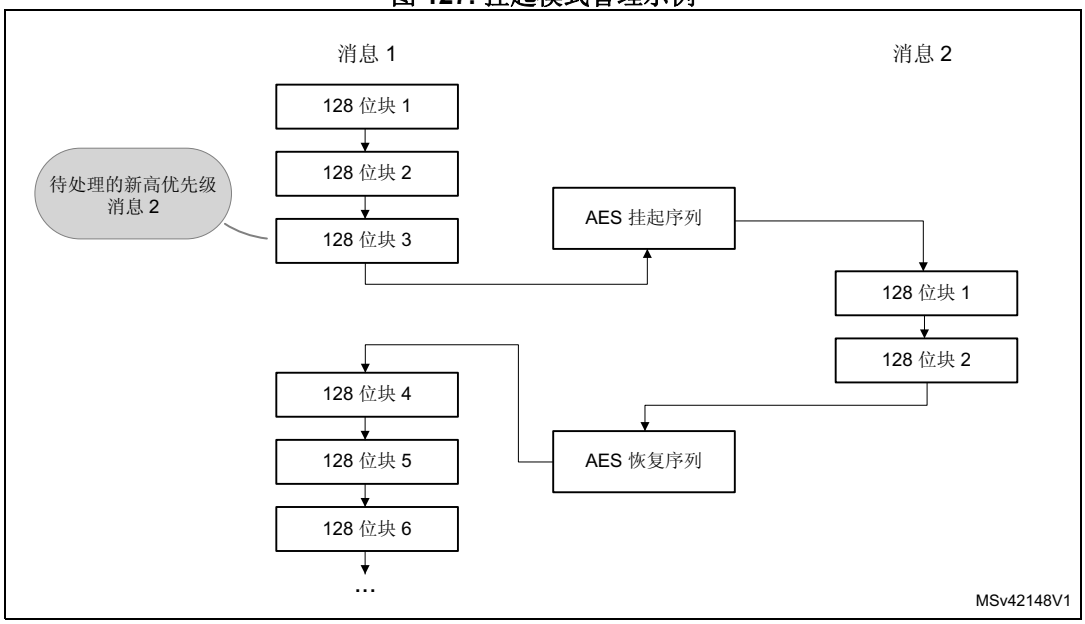
22.4.7 AES 任务挂起和恢复

如果必须处理另一条优先级较高的消息，则可将消息挂起。该最高优先级消息完成发送后，可在加密或解密模式下恢复挂起的消息。

挂起/恢复操作不会破坏链接运算，并且再次使能 AES 以接收下一个数据块时，可立即恢复消息处理。

图 127 所示为挂起/恢复操作示例：为发送长度更短、优先级更高的消息 2 而将消息 1 挂起。

图 127. 挂起模式管理示例



有关挂起/恢复操作的详细说明，请参见每个 AES 模式的相应章节。

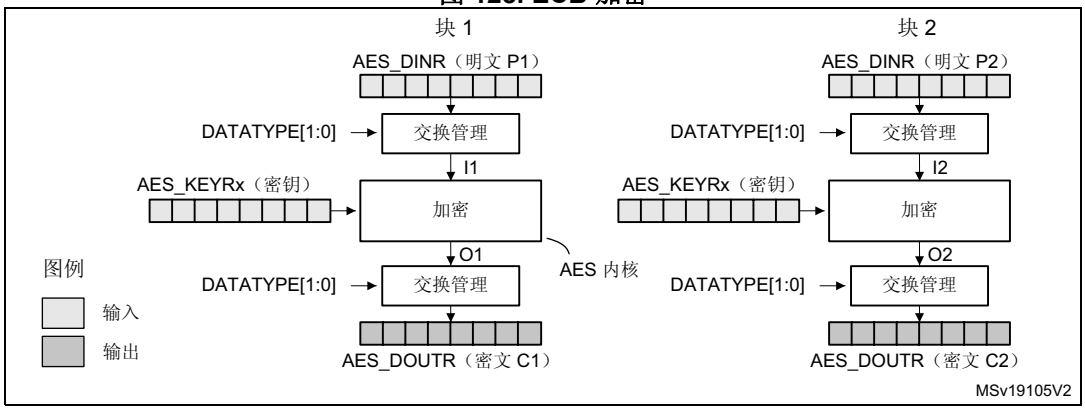
22.4.8 AES 基本链接模式（ECB 和 CBC）

概述

本节简要介绍 AES 计算内核提供的四种基本工作模式：ECB 加密、ECB 解密、CBC 加密和 CBC 解密。有关详细信息，请参见 2001 年 11 月 26 日的 FIPS 出版物 197。

图 128 介绍了电子密码本 (ECB) 加密。

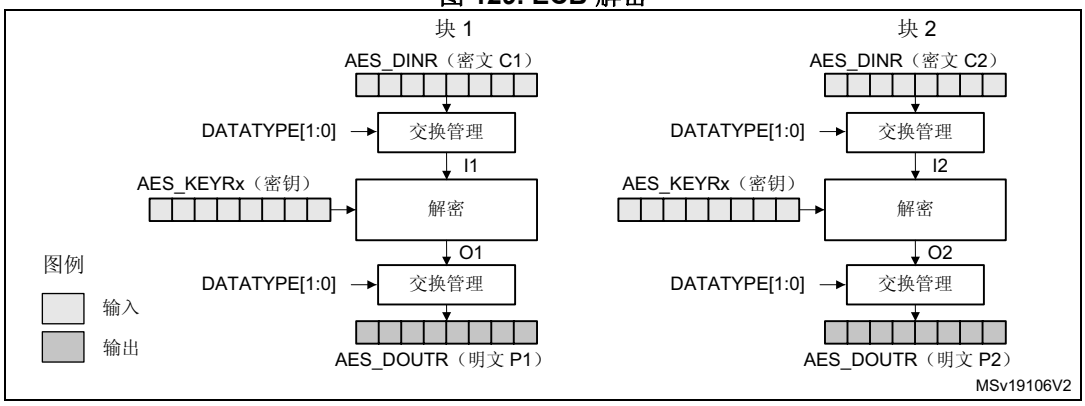
图 128. ECB 加密



在 ECB 加密模式下，AES_DINR 寄存器中的 128 位明文输入数据块 Px 首先进行位/字节/半字交换。使用 128 位或 256 位密钥，通过在 AES 加密模式下设置的 AES 内核对交换结果 Ix 进行处理。加密结果 Ox 经过位/字节/半字交换，然后作为 128 位密文输出数据块 Cx 存储在 AES_DOUTR 寄存器中。ECB 加密以这种方式继续，直到最后一个完整的明文块被加密。

图 129 介绍了电子密码本 (ECB) 解密。

图 129. ECB 解密

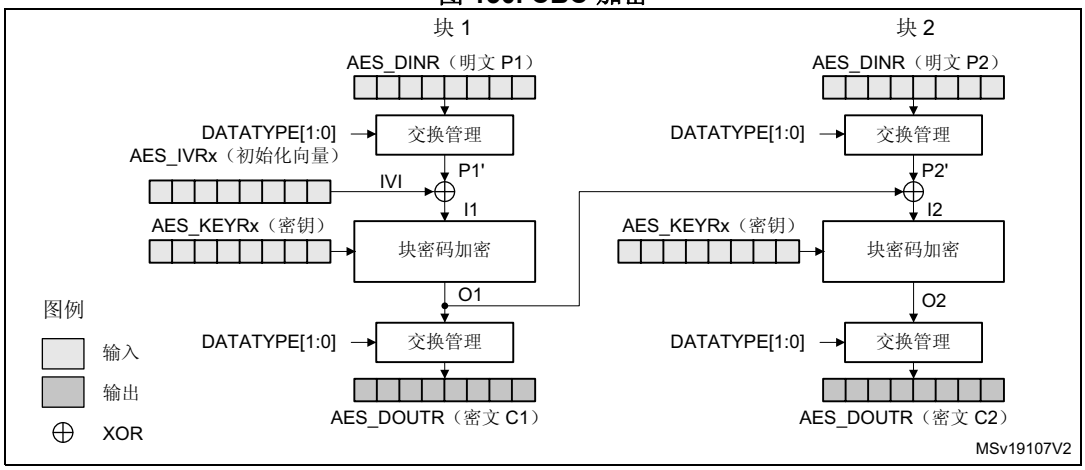


要在 ECB 模式下执行 AES 解密，需要通过收集最后一轮加密密钥准备密钥（需要针对加密首先执行完整的密钥计划），并将其用作解密密文的第一个轮密钥。该准备由 AES 内核提供支持。

在 ECB 解密模式下，AES_DINR 寄存器中的 128 位密文输入数据块 C1 首先进行位/字节/半字交换。该密钥序列与 ECB 加密中的密钥序列相反。使用先前准备的解密密钥，通过在解密模式下设置的 AES 内核对交换结果 I1 进行处理。解密结果经过位/字节/半字交换，然后作为 128 位明文输出数据块 P1 存储在 AES_DOUTR 寄存器中。ECB 解密以这种方式继续，直到最后一个完整的密文块被解密。

图 130 说明了密码块链接 (CBC) 加密模式。

图 130. CBC 加密

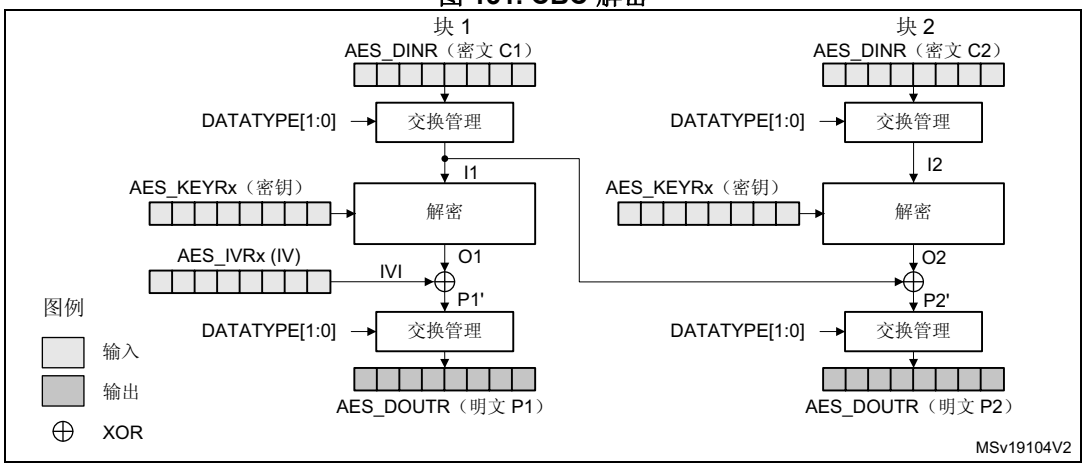


在 CBC 加密模式下，执行位/字节/半字交换 (P1') 后的第一个明文输入块与一个 128 位 IVI 位域（初始化向量和计数器）进行逻辑异或运算，从而使用 128 位或 256 位密钥通过 AES 内核生成用于加密的 I1 输入数据。在交换操作之后，得到的 128 位输出块 O1 用作密文 C1。然后将 O1 数据与第二块明文数据 P2' 进行逻辑异或运算，生成用于 AES 内核的 I2 输入数据以生成第二块密文数据。数据块的链接以这种方式继续，直到消息中的最后一个明文块被加密。

如果消息大小不是 128 位的倍数，则将按照*第 22.4.6 节：AES 密文窃取和数据填充*中说明的方式对最后的不完整数据块进行加密。

图 131 说明了密码块链接 (CBC) 解密。

图 131. CBC 解密



在 CBC 解密模式下（如 ECB 解密模式），必须准备密钥才可以执行 AES 解密。

完成密钥准备过程后，按下述方式执行解密：使用 128 位或 256 位密钥将第一个 128 位密文块（在交换操作之后）直接用作用于解密操作的 AES 核心输入块 I1。其输出 O1 与 128 位 IVI 字段（必须与加密期间使用的字段相同）进行逻辑异或运算以生成第一个明文块 P1。

除了使用来自第一个块的 I1 数据代替初始化向量外，第二个密文块的处理方式与第一个块相同。

解密以这种方式继续，直到最后一个完整的密文块被解密。

如果消息大小不是 128 位的倍数，则将按照*第 22.4.6 节：AES 密文窃取和数据填充*中说明的方式对最后的不完整数据块进行解密。

有关数据交换的更多信息，请参见*第 22.4.13 节：AES 数据寄存器和数据交换*。

ECB/CBC 加密序列

执行 ECB/CBC 加密的事件序列（详见*第 22.4.4 节*）：

1. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。
2. 通过将 AES_CR 寄存器的 MODE[1:0] 位域设置为 00 来选择模式 1，通过将 AES_CR 寄存器的 CHMOD[2:0] 位域设置为 000 或 001 来分别选择 ECB 或 CBC 链接模式。也可以使用 DATATYPE[1:0] 位域定义数据类型。
3. 通过 AES_CR 寄存器中的 KEYSIZE 位选择 128 位或 256 位密钥长度。
4. 将加密密钥写入 AES_KEYRx 寄存器（128 或 256 位）。如果选择了 CBC 模式，则用初始化向量数据填充 AES_IVRx 寄存器。
5. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
6. 对 AES_DINR 寄存器执行四次写入操作，以输入明文（MSB 优先），如*图 132* 所示。
7. 等待 AES_SR 寄存器中的 CCF 标志置 1。
8. 对 AES_DOUTR 寄存器进行四次读取操作，以获取密文（MSB 优先），如*图 132* 所示。然后通过将 AES_CR 寄存器中的 CCFC 位置 1，将 CCF 标志清零。
9. 重复执行步骤 *6*-*7*-*8*，使用相同加密密钥处理所有块。

图 132. ECB/CBC 加密（模式 1）

ECB/CBC 解密序列

执行 AES ECB/CBC 解密的事件序列如下（详见*第 22.4.4 节*）：

1. 按照*第 573 页的第 22.4.5 节：AES 解密密钥准备*中所述的步骤，准备 AES 内核中的解密密钥。
2. 通过将 AES_CR 寄存器的 MODE[1:0] 位域设置为 10 来选择模式 3，并通过将 AES_CR 寄存器的 CHMOD[2:0] 位域设置为 000 或 001 来分别选择 ECB 或 CBC 链接模式。也可以使用 DATATYPE[1:0] 位域定义数据类型。
3. 通过 AES_CR 寄存器的 KEYSIZE 位域，选择 128 或 256 位密钥长度。
4. 将初始化向量写入 AES_IVRx 寄存器（仅在 CBC 模式下需要此步骤）。
5. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES。
6. 对 AES_DINR 寄存器执行四次写入操作，以输入密文（MSB 优先），如*图 133* 所示。
7. 等待 AES_SR 寄存器中的 CCF 标志置 1。
8. 对 AES_DOUTR 寄存器进行四次读取操作，以获取明文（MSB 优先），如*图 133* 所示。然后通过将 AES_CR 寄存器中的 CCFC 位置 1，将 CCF 标志清零。
9. 重复执行步骤 *6-7-8*，使用相同密钥解密所有块。

图 133. ECB/CBC 解密（模式 3）

ECB/CBC 模式下的挂起/恢复操作

要挂起消息处理，请执行以下操作：

1. 如果使用 DMA，则通过将 AES_CR 寄存器中的 DMAINEN 位清零来停止 AES DMA 对 IN FIFO 的数据传输。
2. 如果未使用 DMA，则读取四次 AES_DOUTR 寄存器以保存最后处理的块。如果使用 DMA，等待 AES_SR 寄存器中的 CCF 标志置 1，然后通过将 AES_CR 寄存器中的 DMAOUTEN 输出位清零来停止 DMA 对 OUT FIFO 的数据传输。
3. 如果未使用 DMA，轮询 AES_SR 寄存器的 CCF 标志，直到它变为 1（计算完成）。
4. 通过将 AES_CR 寄存器中的 CCFC 位置 1，将 CCF 标志清零。
5. 保存初始化向量寄存器（仅在 CBC 模式下需要此步骤，因为 AES_IVRx 寄存器在数据处理过程中已发生更改）。
6. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。
7. 将当前 AES 配置保存到存储器中（AES 初始化向量值除外）。如果处理高优先级消息时不需要密钥寄存器，则将其清零。
8. 如果使用 DMA，则保存 DMA 控制器状态（指向 IN 和 OUT 数据传输的指针以及剩余字节数等）。

注： *在第 7 步中，如果中断的进程是解密，则可以选择将存储在 AES_KEYRx 寄存器中的已分散密钥信息保存在存储器中。否则无需保存这些寄存器，因为应用程序已知初始密钥值*

要恢复消息处理，请执行以下操作：

1. 如果使用 DMA，则配置 DMA 控制器以完成 FIFO IN 和 FIFO OUT 传输的其余部分。
2. 确保 AES 禁止（AES_CR 的 EN 位必须为 0）。
3. 使用已保存配置的值恢复 AES_CR 和 AES_KEYRx 寄存器设置。在解密的情况下，可将分散的密钥信息写入 AES_KEYRx 寄存器，而不是原始密钥值。
4. 按*第 22.4.5 节：AES 解密密钥准备*所述准备解密密钥（仅 ECB 或 CBC 解密需要此步骤）。如果已将分散的密钥信息加载到 AES_KEYRx 寄存器中，则此步骤不是必需的。
5. 使用保存的配置恢复 AES_IVRx 寄存器（仅在 CBC 模式下需要此步骤）。
6. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
7. 如果使用 DMA，将 AES_CR 寄存器中的 DMAINEN 和 DMAOUTEN 位置 1，以使能 AES DMA 传输。

使用模式 4 执行备用单次 ECB/CBC 解密

使用模式 4 执行单次 ECB/CBC 解密的事件序列为：

1. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。
2. 通过将 AES_CR 寄存器的 MODE[1:0] 位域设置为 11 来选择模式 4，通过将 AES_CR 寄存器的 CHMOD[2:0] 位域设置为 000 或 001 来分别选择 ECB 或 CBC 链接模式。
3. 通过 AES_CR 寄存器的 KEYSIZE 位域，选择 128 或 256 位密钥长度。
4. 将加密密钥写入 AES_KEYRx 寄存器。如果选择 CBC 模式，则写入 AES_IVRx 寄存器。
5. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。

6. 对 AES_DINR 寄存器执行四次写入操作，以输入密文（MSB 优先）。
7. 等待 AES_SR 寄存器中的 CCF 标志置 1。
8. 对 AES_DOUTR 寄存器进行四次读取操作，以获取明文（MSB 优先）。然后通过将 AES_CR 寄存器中的 CCFC 位置 1，将 CCF 标志清零。

注： *选择模式 4 时，不能使用模式 3。*

在模式 4 下，AES_KEYRx 寄存器包含所有处理阶段的加密密钥。分散密钥不会存储在这些寄存器中。它会存储在 AES 内部。

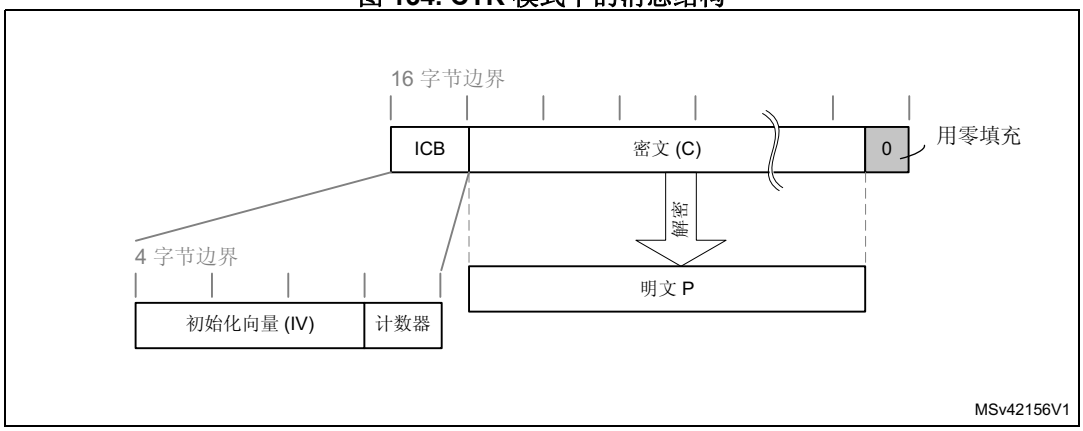
22.4.9 AES 计数器 (CTR) 模式

概述

计数器模式 (CTR) 使用 AES 作为密钥流生成器。然后，生成的密钥与明文进行逻辑异或运算来获得密文。

NIST *特别出版物 800-38A，块密码工作模式的建议*对 CTR 链接进行了相关定义。*图 134* 给出了 CTR 模式下的典型消息结构。

图 134. CTR 模式下的消息结构



该消息的结构为：

- 16 字节的初始计数器块 (ICB)，它由两个不同的字段组成：
  - **初始化向量** (IV)：96 位值，对于给定密钥下每个加密周期，该值都必须唯一。
  - **计数器**：32 位大端模式的整数，随着块的处理次数而递增。应将计数器的初始值设为 1。
- 明文 P 被加密为密文 C（长度已知）。该长度可以不是 16 字节的倍数，在这种情况下需要明文填充。

CTR 加密和解密

图 135 和*图 136* 分别描述了 AES 外设中实现的 CTR 加密和解密过程。通过向 AES_CR 寄存器中的 CHMOD[2:0] 位域写入 010 来选择 CTR 模式。

图 135. CTR 加密

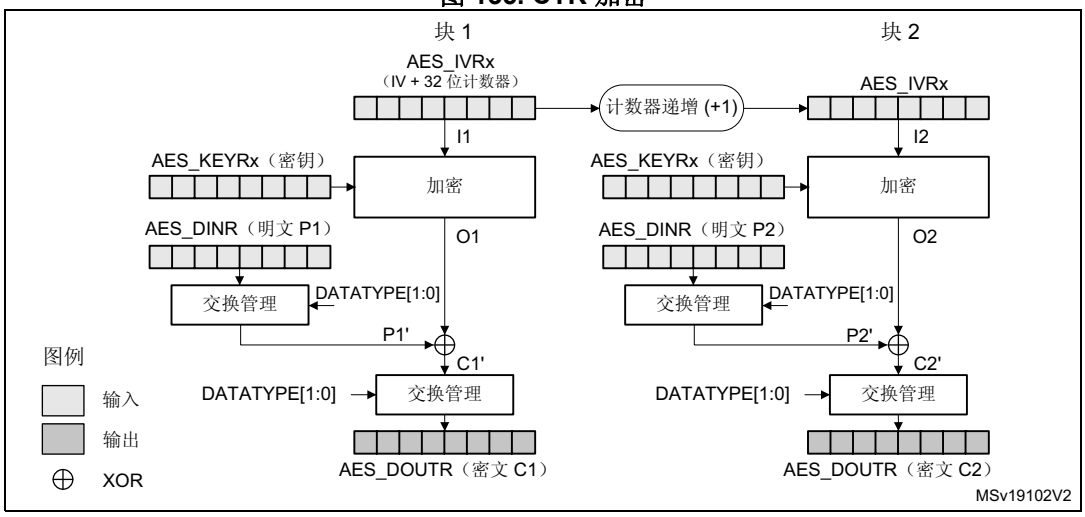
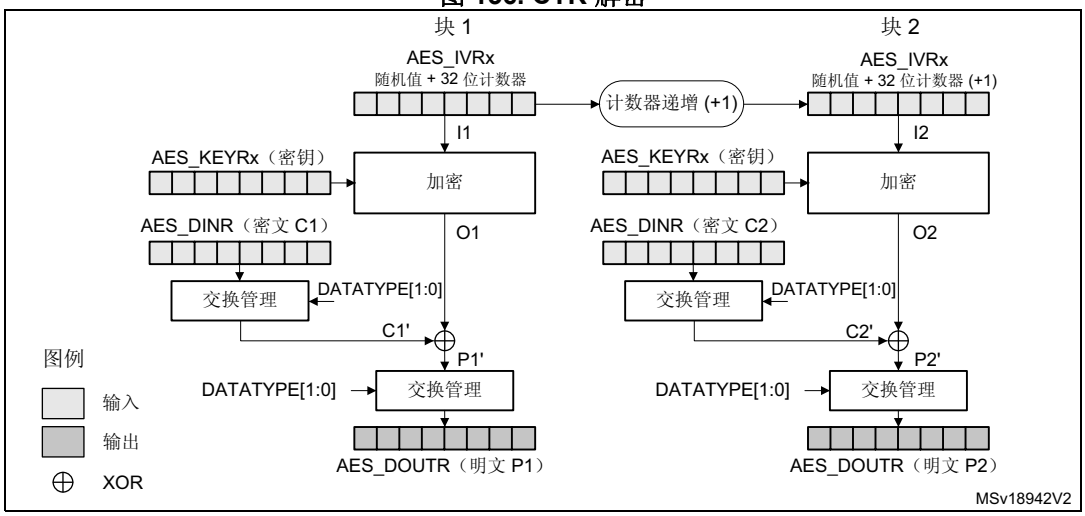


图 136. CTR 解密



在 CTR 模式中，将加密核心输出（也称为密钥流）Ox 与相关输入块（用于加密的 Px'，用于解密的 Cx'）进行逻辑异或运算，以生成正确的输出块（用于加密的 Cx'，用于解密的 Px'）。AES 中的初始化向量必须按*表 108* 所示进行初始化。

表 108. CTR 模式初始化向量定义

AES_IVR3[31:0]	AES_IVR2[31:0]	AES_IVR1[31:0]	AES_IVR0[31:0]
Nonce[31:0]	Nonce[63:32]	Nonce[95:64]	32 位计数器 = 0x0001

在 CBC 模式下，处理第一个数据块时仅使用 AES_IVRx 寄存器一次，而在 CTR 模式下则有所不同，处理每个数据块都使用 AES_IVRx 寄存器，并且 AES 外设会使初始化向量的计数器位递增（随机值位保持不变）。

CTR 解密与 CTR 加密并无不同，因为内核始终会对当前的计数器块加密以产生密钥数据流，该密钥数据流与明文（CTR 加密）或密文（CTR 解密）输入进行异或运算。在 CTR 模式下，MODE[1:0] 位域设置 11、10 或 00 将全部默认为加密模式，同时禁止设置 01（密钥分散）。

在 CTR 链接模式下执行加密或解密的事件序列：

1. 确保 AES 禁止（AES_CR 的 EN 位必须为 0）。
2. 通过将 AES_CR 寄存器中的 CHMOD[2:0] 位域设置为 010 来选择 CTR 链接模式。将 MODE[1:0] 位域设置为除 01 之外的任何值。
3. 初始化 AES_KEYRx 寄存器，并按*表 108* 所述加载 AES_IVRx 寄存器。
4. 将 AES_CR 寄存器的 EN 位置 1，开始加密当前计数器（计算完成时 EN 自动复位）。
5. 如果是最后一个块，可在需要时用零填充数据以获得完整的块。
6. 在 AES 中附加数据并读取结果。*第 22.4.4 节：执行密码操作的 AES 过程*介绍了三种可能的情况。
7. 重复上一步，直到处理完倒数第二个块。对于最后一个块，执行前两步并将不属于有效负载的一部分的位丢弃（如果最后一个输入块中有效数据的大小小于 16 字节）。

CTR 模式下的挂起/恢复操作

该模式与 CBC 模式相似，可以中断消息、发送优先级更高的消息，并可恢复已中断的消息。有关 CBC 挂起/恢复序列的详细信息，请参见*第 22.4.8 节：AES 基本链接模式（ECB 和 CBC）*。

注： *与 CBC 模式类似，恢复操作期间必须重新加载 AES_IVRx 寄存器。*

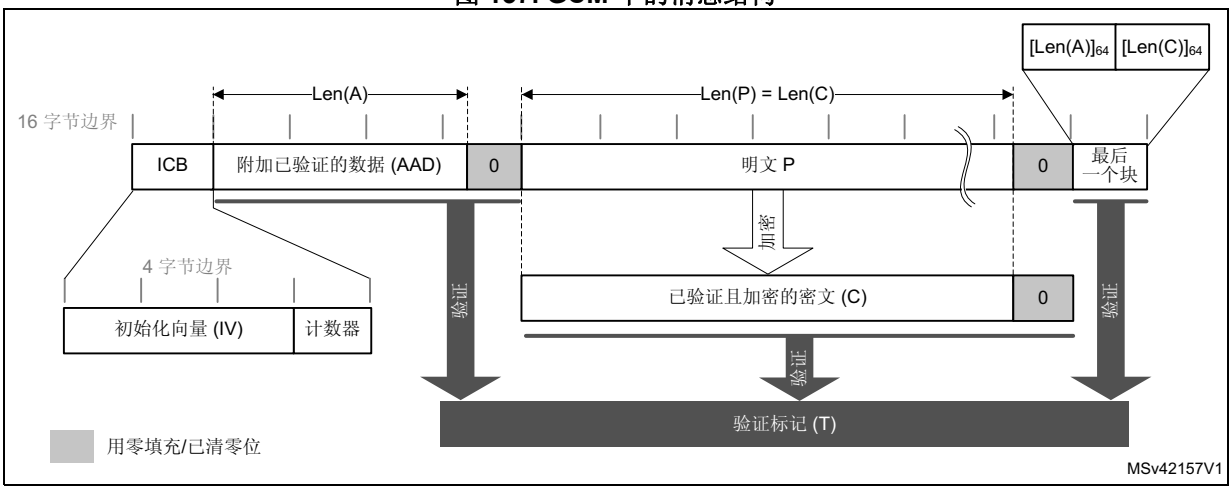
22.4.10 AES Galois/计数器模式 (GCM)

概述

AES Galois/计数器模式 (GCM) 允许将明文消息加密和验证为对应的密文和标记（也称为消息认证码）。为实现保密性，GCM 算法基于 AES 计数器模式。它使用固定有限域乘法器来生成标记。

NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对 GCM 链接进行了相关定义。*图 137* 所示为 GCM 模式下的典型消息结构。

图 137. GCM 中的消息结构



消息具有以下结构：

- **16 字节的初始计数器块 (ICB)**，它由两个不同的字段组成：
  - **初始化向量 (IV)**：96 位值，对于给定密钥下每个加密周期，该值都必须唯一。请注意，GCM 标准支持短于 96 位的 IV，在这种情况下应遵循严格的规则。
  - **计数器**：32 位大端模式的整数，随着块的处理次数而递增。根据 NIST 规范，处理有效负载的第一个块时，计数器值为 0x2。
- **认证的标头 AAD**（也称为附加认证数据）具有已知长度 Len(A)，此长度值可以不是 16 字节的倍数，但不能超过 $2^{64} – 1$ 位。消息的这一部分仅经过认证，未被加密。
- **明文消息 P** 将经过认证且被加密为密文 C，其具有已知长度 Len(P)，该长度值可以不是 16 字节的倍数，但不能超过 $2^{32} - 2$ 个 128 位块。
- **最后一个块**包含 AAD 标头长度（位 [32:63]）和有效负载长度（位 [96:127]）信息，如 *表 109* 所示。

GCM 标准规定密文 C 的位长度与明文 P 的位长度相同。

当消息某一部分（AAD 或 P）的长度不是 16 字节的倍数时，需要采取特殊填充方案。

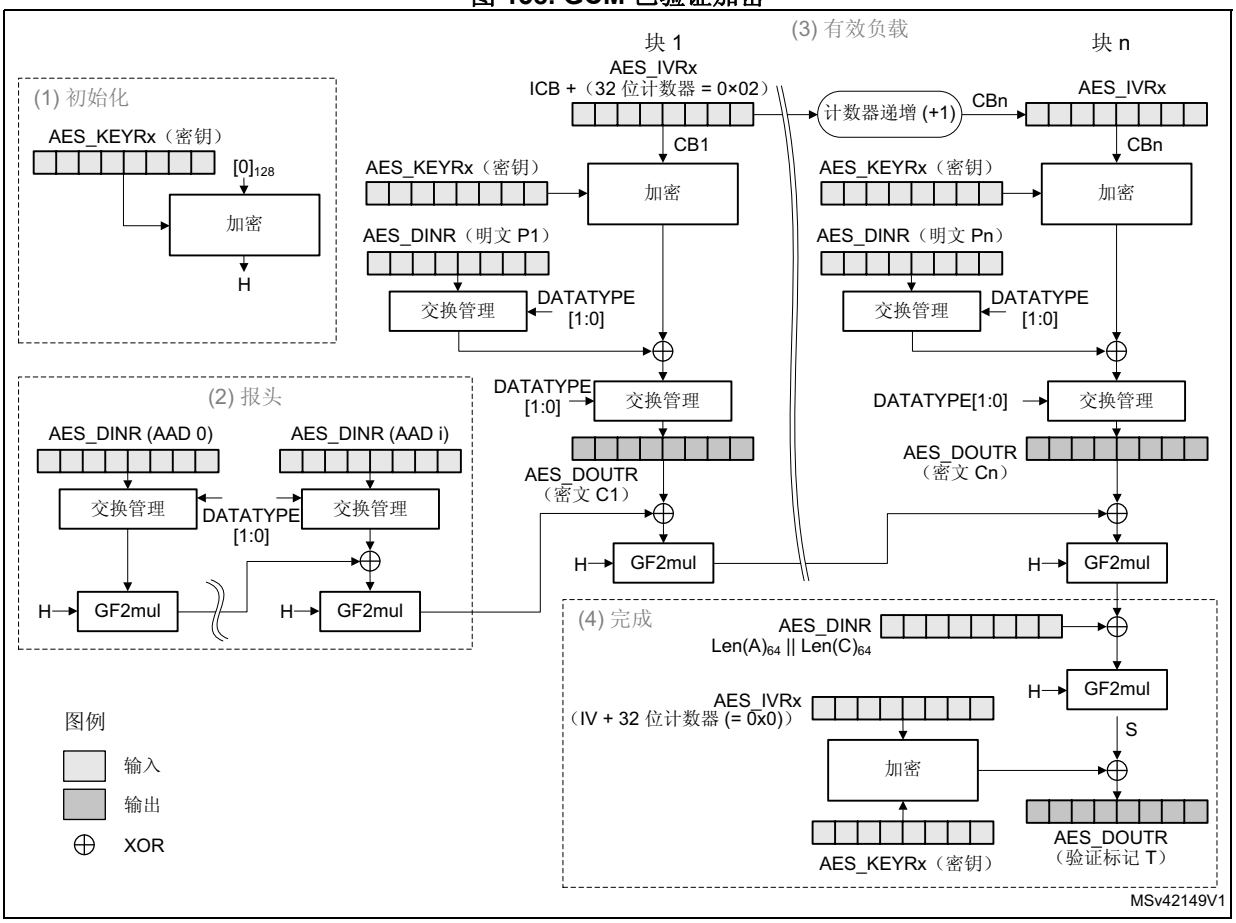
表 109. GCM 最后一个块定义

字节序	位 [0] ---------- 位 [31]	位 [32]---------- 位 [63]	位 [64] -------- 位 [95]	位 [96] --------- 位 [127]
输入数据	0x0	AAD 长度 [31:0]	0x0	有效负载长度 [31:0]

GCM 处理

图 138 列出了 AES 外设中的 GCM 实现。通过向 AES_CR 寄存器中的 CHMOD[2:0] 位域写入 011 来选择 GCM。

图 138. GCM 已验证加密



GCM 模式下明文的保密性机制与计数器模式下的类似，其具有特定递增函数（表示为 32 位递增），可生成输入计数器块序列。

保持数据的**计数器块**的 AES_IVRx 寄存器用于处理每个数据块。AES 外设自动递增计数器 [31:0] 位域。第一个计数器块 (CB1) 由应用软件从初始计数器块 ICB 分散（请参见*表 110*）。

表 110. GCM 模式 IVI 位域初始化

寄存器	AES_IVR3[31:0]	AES_IVR2[31:0]	AES_IVR1[31:0]	AES_IVR0[31:0]
输入数据	ICB[31:0]	ICB[63:32]	ICB[95:64]	Counter[31:0] = 0x2

注： *在 GCM 模式下，禁止将 MODE[1:0] 位域设置为 01 和 11。*

GCM 模式下的认证机制基于散列函数（称为 **GF2mul**），其在二元 Galois 域内执行与固定参数（称为散列子密钥 (H)）的乘法。

GCM 消息通过以下阶段进行处理（将在下一小节中对此进行进一步介绍）：

- **初始化阶段**：AES准备 GCM 散列子密钥 (H)。
- **标头阶段**：AES 处理附加认证数据 (AAD)（仅使用散列计算）。
- **有效负载阶段**：AES 基于散列计算、计数器块加密和数据异或运算处理明文 (P)。其操作方式与密文 (C) 类似。
- **最后阶段**：AES 使用消息的最后一个块生成认证标记 (T)。

GCM 初始化阶段

在此第一步中，将在内部计算和保存 GCM 散列子密钥 (H)，供处理所有块使用。建议序列为：

1. 确保 AES 禁止（AES_CR 的 EN 位必须为 0）。
2. 通过将 AES_CR 寄存器的 CHMOD[2:0] 位域设置为 011 选择 GCM 链接模式，并可选择将 DATATYPE[1:0] 位域置 1。
3. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 00 来指示初始化阶段。
4. 将 AES_CR 寄存器的 MODE[1:0] 位域设置为 00 或 10。尽管此位域仅在有效负载阶段使用，但建议在初始化阶段对其进行设置，并在所有后续阶段保持不变。
5. 使用密钥初始化 AES_KEYRx 寄存器，使用*表 110* 中定义的信息初始化 AES_IVRx 寄存器。
6. 开始通过将 AES_CR 寄存器的 EN 位设置为 1 来计算散列密钥（计算完成时 EN 自动复位）。
7. 等待计算完成，通过将 AES_SR 的 CCF 标志转换为 1 进行指示。或者，使用相应的中断。
8. 通过将 AES_CR 寄存器中的 CCFC 位设置为 1，将 AES_SR 寄存器中的 CCF 标志清零。

GCM 标头阶段

GCM 初始化阶段之后的这个阶段必须在有效负载阶段之前完成。要执行的序列为（加密与解密完全相同）：

1. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 01 来指示标头阶段。请勿修改初始化阶段中设置的 MODE[1:0] 位域。
2. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
3. 如果它是最后一个块，并且块中的 AAD 大小低于 128 位，则用零填充块的其余部分。然后按*第 569 页的第 22.4.4 节：执行密码操作的 AES 过程*中所述的其中一种方式将数据块附加到 AES。
4. 重复步骤 *3*，直到处理完最后一个附加认证数据块。

注： *如无 AAD（即 Len(A) = 0），则可以跳过标头阶段。*

GCM 有效负载阶段

在 GCM 标头阶段之后执行此阶段（加密和解密相同）。在此阶段，加密/解密的有效负载将存储在 AES_DOUTR 寄存器中。要执行的序列为：

1. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 10 来指示有效负载阶段。请勿修改初始化阶段中设置的 MODE[1:0] 位域。
2. 如果跳过了标头阶段，则通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
3. 如果它是最后一个块，并且块中的明文（加密）或密文（解密）大小低于 128 位，则用零填充块的其余部分。
4. 按*第 569 页的第 22.4.4 节：执行密码操作的 AES 过程*中所述的其中一种方式将数据块附加到 AES，并读取结果。
5. 重复上一步，直到倒数第二个明文块被加密或直到最后一个密文块被解密。对于最后一个明文块（仅限加密），执行前两步。对于最后一个块，当其大小小于 16 字节时，将不属于有效负载的一部分的位丢弃。

注： *如无有效负载数据（即 Len(C) = 0），则可以跳过有效负载阶段（请参见 GMAC 模式）。*

GCM 最后阶段

在此最后一个阶段中，AES 外设会生成 GCM 认证标记并将其存储在 AES_DOUTR 寄存器中。要执行的序列为：

1. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 11 来指示最后阶段。
2. 通过连接 AAD 位长度和有效负载位长度来组成块的数据，如*表 109* 中所示。将块写入 AES_DINR 寄存器。
3. 等待计算完成，通过将 AES_SR 的 CCF 标志转换为 1 进行指示。
4. 对 AES_DOUTR 寄存器进行四次读取操作，以获取 GCM 认证标记。
5. 通过将 AES_CR 寄存器中的 CCFC 位设置为 1，将 AES_SR 寄存器中的 CCF 标志清零。
6. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。如果它是经验证的解密，请将生成的标记与通过消息传递的预期标记进行比较。

注： *在最后阶段，必须正常插入数据（无交换）。*

当从报头或有效负载阶段过渡到最后阶段时，不得禁止 AES 外设，否则结果会是错误的。

GCM 模式下的挂起/恢复操作

要挂起消息处理，请执行以下操作：

1. 如果使用 DMA，则通过将 AES_CR 寄存器中的 DMAINEN 位清零来停止 AES DMA 对 IN FIFO 的数据传输。如果未使用 DMA，确保当前计算已完成，完成由 AES_SR 寄存器的 CCF 标志设置为 1 指示。
2. 在有效负载阶段，如果未使用 DMA，则读取四次 AES_DOUTR 寄存器以保存最后处理的块。如果使用 DMA，等待 AES_SR 寄存器中的 CCF 标志置 1，然后通过将 AES_CR 寄存器中的 DMAOUTEN 输出位清零来停止 DMA 对 OUT FIFO 的数据传输。
3. 通过将 AES_CR 寄存器中的 CCFC 位设置为 1，将 AES_SR 寄存器中的 CCF 标志清零。在有效负载阶段（仅限加密模式），验证 AES_SR 寄存器的 BUSY 标志是否已清零，以确保 GF2mul 散列函数已完成。
4. 将 AES_SUSPxR 寄存器保存到存储器中，其中 x 等于 0 到 7。
5. 在有效负载阶段，保存 AES_IVRx 寄存器，因为在数据处理过程中，它们相较于初始值发生更改。在标头阶段，则不需要此步骤。

6. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。
7. 将当前 AES 配置保存到存储器中（初始化向量寄存器 AES_IVRx 除外）。无需保存密钥寄存器，因为应用程序已知初始密钥值。
8. 如果使用 DMA，则保存 DMA 控制器状态 （指向 IN 数据传输的指针以及剩余字节数等）。在有效负载阶段，也必须保存指向 OUT 数据传输的指针。

要恢复消息处理，请执行以下操作：

1. 如果使用 DMA，则重新配置 DMA 控制器以完成 FIFO IN 传输的其余部分。在有效负载阶段，也必须在 DMA 控制器中配置 FIFO OUT 的其余部分。
2. 确保 AES 外设已禁止（AES_CR 寄存器的 EN 位必须为 0）。
3. 将先前保存在存储器中的挂起寄存器值回写到其相应的 AES_SUSPxR 寄存器中，其中 x 等于 0 到 7。
4. 在有效负载阶段，将先前保存在存储器中的初始化向量寄存器值写回到其相应的 AES_IVRx 寄存器中。在标头阶段，将初始化设置值写回到 AES_IVRx 寄存器中。
5. 恢复 AES_CR 和 AES_KEYRx 寄存器中的初始化设置值。
6. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
7. 如果使用 DMA，将 AES_CR 寄存器中的 DMAINEN 位（和 DMAOUTEN 位，如果在有效负载阶段）置 1，以使能 AES DMA 请求。

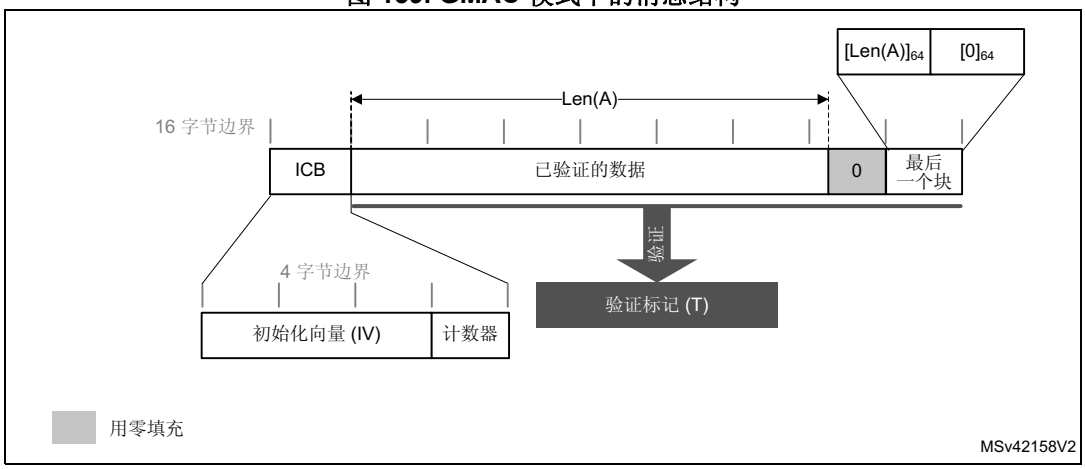
22.4.11 AES Galois 消息认证码 (GMAC)

概述

Galois 消息认证码 (GMAC) 支持认证明文，生成相应的标记信息（也称为消息认证码）。它基于 GCM 算法，NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/ 计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

图 139 给出了 GMAC 下的典型消息结构。

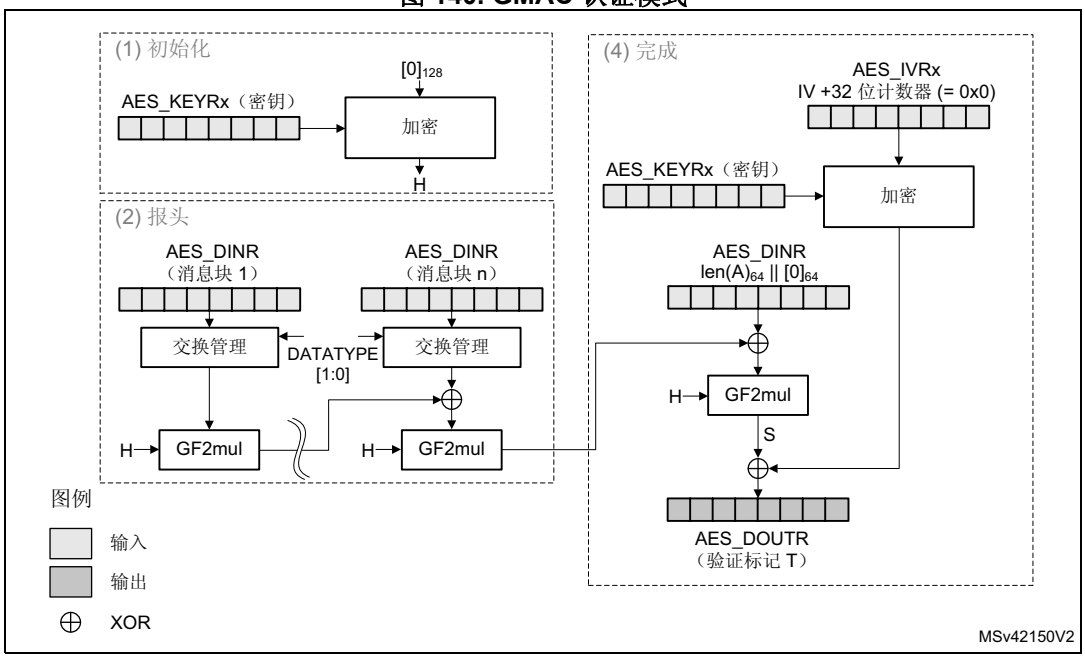
图 139. GMAC 模式下的消息结构



AES GMAC 处理

图 140 列出了 AES 外设中的 GMAC 模式实现。通过向 AES_CR 寄存器中的 CHMOD[2:0] 位域写入 011 来选择此模式。

图 140. GMAC 认证模式



GMAC 算法相当于应用在仅包含标头的消息上的 GCM 算法。因此，除了忽略有效负载阶段之外，所有步骤和设置均与 GCM 相同。

GMAC 模式下的挂起/恢复操作

在 GMAC 模式下，除了只能中断标头阶段外，针对 GCM 描述的序列适用。

22.4.12 AES CBC-MAC 计数器模式 (CCM)

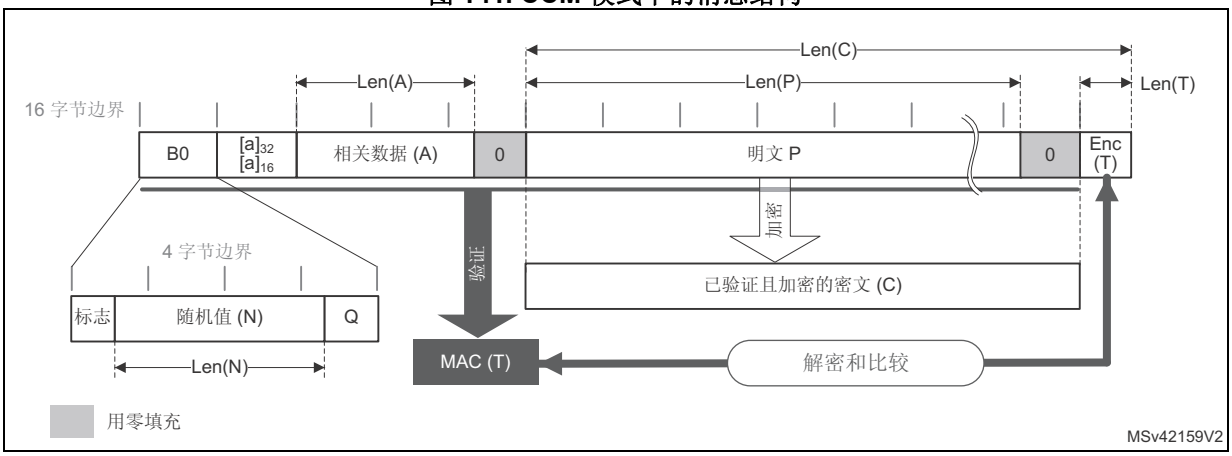
概述

AES **密码块链接-消息认证码计数器模式** (CCM) 算法可用于加密和认证明文，生成对应的密文和标记（也称为消息认证码）。为实现保密性，CCM 算法基于 AES 计数器模式。它使用密码块链接技术来生成消息认证码。这通常被称为 CBC-MAC。

注： *NIST 不允许将该 CBC-MAC 用作 CCM 规范上下文之外的认证模式。*

NIST *特别出版物 800-38C，块密码工作模式的建议 - CCM 模式实现认证和保密性*中对 CCM 链接进行了介绍。*图 141* 给出了 CCM 下的典型消息结构。

图 141. CCM 模式下的消息结构



消息的结构为：

- **16 字节的首次认证块 (B0)**，它由三个不同的字段组成：
  - **Q**：位字符串，表示 P 的八位字节长度 (Len(P))。
  - **随机值 (N)**：大小为 Len(N) 的一次性值（即，应为每一次新的通信分配新的随机值）。Len(N) + Len(P) 的和必须等于 15 个字节。
  - **标志**：最高有效八位字节，包含四个控制信息标志（根据标准规定）。它包含两个 3 位字符串，用于编码值 **t**（MAC 长度，以字节表示）和 **Q**（明文长度，例如 Len(P) < 2^{8q} 个字节）。与 **Q** 相关的计数器块范围为 2^{8Q-4}，即，如果 **Q** 的最大值为 8，则密码中使用的计数器块应为 60 位。
- 与相关数据 (A) 关联的 **16 字节块 (B)**。
  消息的这一部分仅经过认证，未被加密。这部分具有已知长度 Len(A)，此长度值可以不是 16 字节的倍数（请参见*图 141*）。标准还具有以下规定：对于第一个消息块 (B1) 的 MSB 位，以字节表示的相关数据长度 (a) 必须按如下所述进行编码：
  - 如果 $0 < a < 2^{16} - 2^8$，则将其编码为 $[a]_{16}$，即两个字节。
  - 如果 $2^{16} - 2^8 < a < 2^{32}$，则将其编码为 0xff || 0xfe || $[a]_{32}$，即六个字节。
  - 如果 $2^{32} < a < 2^{64}$，则将其编码为 0xff || 0xff || $[a]_{64}$，即十个字节。
- 与明文消息 P 相关的 **16 字节块 (B)**，此明文消息将经过认证并被加密为密文 C（具有已知长度 Len(P)）。此长度可以不是 16 字节的倍数（请参见*图 141*）。
- 长度为 Len(T) 的**加密 MAC (T)** 被附加到总长度为 Len(C) 的密文 C。

当消息某一部分（A 或 P）的长度不是 16 字节的倍数时，需要采取特殊填充方案。

注： *CCM 链接模式同样只能与相关数据搭配使用（即，无有效负载）。*

例如，NIST 特别出版物 800-38C 的 C.1 部分给出了以下值（十六进制数）：

N：10111213 141516（Len(N)= 56 位或 7 字节）

A：00010203 04050607（Len(A)= 64 位或 8 字节）

P：20212223（Len(P)= 32 位或 4 字节）

T：6084341B（Len(T)= 32 位或 t = 4）

B0：4F101112 13141516 00000000 00000004

B1：00080001 02030405 06070000 00000000

B2：20212223 00000000 00000000 00000000

CTR0：0710111213 141516 00000000 00000000

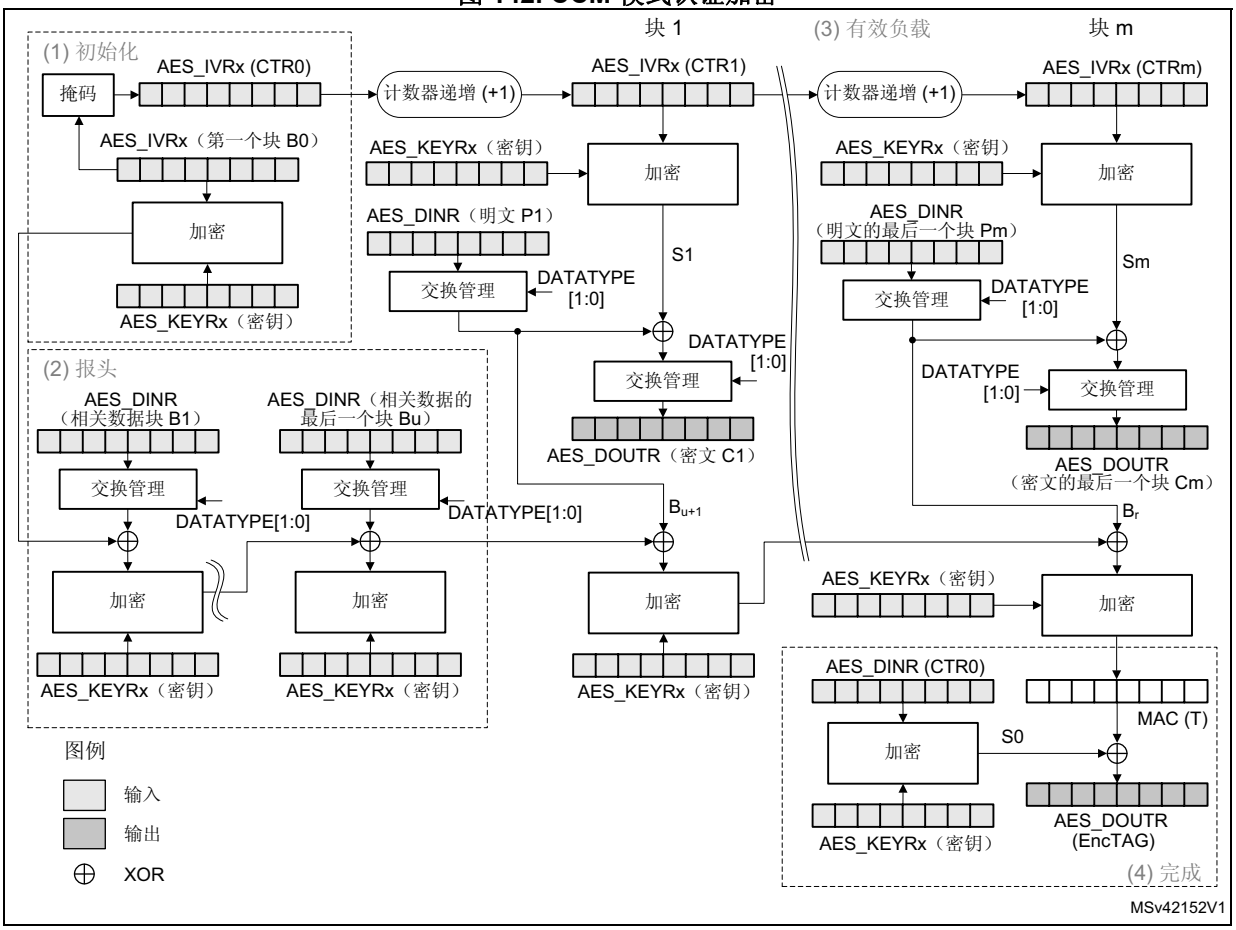
CTR1：0710111213 141516 00000000 00000001

格式化输入数据块 Bx（特别是 B0 和 B1）的生成必须由应用程序管理。

CCM 处理

图 142 列出了 AES 外设中的 CCM 实现（加密示例）。通过向 AES_CR 寄存器中的 CHMOD[2:0] 位域写入 100 来选择此模式。

图 142. CCM 模式认证加密



生成-加密过程的数据输入包括一个有效随机值、一个有效的有效负载字符串和一个有效的相关数据字符串，这些数据均经过正确格式化。对经过格式化的明文数据应用 CBC 链接机制可生成长度已知的 MAC。计数器模式加密需要足够长的计数器块序列作为输入，其将应用于有效负载字符串并单独应用于 MAC。得到的密文 C 是明文 P 的生成-加密过程的输出。

AES_IVRx 寄存器用于处理每个数据块，AES 以第一个块 B0 定义的位长度自动使 CTR 计数器递增。*表 111* 显示了应用程序必须如何加载 B0 数据。

注： *根据 NIST 规定，CCM 模式下的 AES 外设最多支持 64 位计数器。*

表 111. 在 CCM 模式下初始化 AES_IVRx 寄存器

寄存器	AES_IVR3[31:0]	AES_IVR2[31:0]	AES_IVR1[31:0]	AES_IVR0[31:0]
输入数据	B0[31:0]	B0[63:32]	B0[95:64]	B0[127:96]

注： *在 CCM 模式下，禁止将 MODE[1:0] 位域设置为 01 和 11（密钥分散）。*

CCM 消息通过以下阶段进行处理（将在下一小节中对此进行进一步介绍）：

- **初始化阶段**：AES 处理第一个块并准备第一个计数器块。
- **标头阶段**：AES 仅使用标记计算处理相关数据 (A)。
- **有效负载阶段**：IP 基于标记计算、计数器块加密和数据异或运算处理明文 (P)。其操作方式与密文 (C) 类似。
- **最后阶段**：AES 生成消息认证码 (MAC)。

CCM 初始化阶段

在此阶段中，将 CCM 消息的第一个块 B0 写入 AES_IVRx 寄存器中。AES_DOUTR 寄存器不包含任何输出数据。建议序列为：

1. 确保 AES 外设已禁止（AES_CR 的 EN 位必须为 0）。
2. 通过将 AES_CR 寄存器的 CHMOD[2:0] 位域设置为 100 选择 CCM 链接模式，并可选择将 DATATYPE[1:0] 位域置 1。
3. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 00 来指示初始化阶段。
4. 将 AES_CR 寄存器的 MODE[1:0] 位域设置为 00 或 10。尽管此位域仅在有效负载阶段使用，但建议在初始化阶段对其进行设置，并在所有后续阶段保持不变。
5. 使用密钥初始化 AES_KEYRx 寄存器，并如*表 111* 所述使用 B0 数据初始化 AES_IVRx 寄存器。
6. 通过将 AES_CR 寄存器的 EN 位设置为 1 来启动计数器计算（计算完成时 EN 自动复位）。
7. 等待计算完成，通过将 AES_SR 的 CCF 标志转换为 1 进行指示。或者，使用相应的中断。
8. 通过将 AES_CR 寄存器中的 CCFC 位设置为 1，将 AES_SR 寄存器中的 CCF 标志清零。

CCM 标头阶段

GCM 初始化阶段之后的这个阶段必须在有效负载阶段之前完成。在此阶段期间，AES_DOUTR 寄存器不包含任何输出数据。

要执行的序列为（加密与解密完全相同）：

1. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 01 来指示标头阶段。请勿修改初始化阶段中设置的 MODE[1:0] 位域。
2. 如果跳过了标头阶段，则通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
3. 如果它是最后一个块，并且块中的 AAD 大小低于 128 位，则用零填充块的其余部分。然后按*第 569 页的第 22.4.4 节：执行密码操作的 AES 过程*中所述的其中一种方式将数据块附加到 AES。
4. 重复步骤 *3*，直到处理完最后一个附加认证数据块。

注： *如无相关数据（即 Len(A) = 0），则可以跳过标头阶段。*

必须由软件使用相关数据长度格式化相关数据的首个块 (B1)。

CCM 有效负载阶段（加密或解密）

在 CCM 标头阶段之后执行此阶段（加密和解密相同）。在此阶段，加密/解密的有效负载将存储在 AES_DOUTR 寄存器中。要执行的序列为：

1. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 10 来指示有效负载阶段。请勿修改初始化阶段中设置的 MODE[1:0] 位域。
2. 如果跳过了标头阶段，则通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
3. 如果它是最后一个待加密的数据块，并且块中明文大小低于 128 位，则用零填充块的其余部分。
4. 按*第 569 页的第 22.4.4 节：执行密码操作的 AES 过程*中所述的其中一种方式将数据块附加到 AES，并读取结果。
5. 重复上一步，直到倒数第二个明文块被加密或直到最后一个密文块被解密。对于最后一个明文块（仅限加密），应用两步。对于最后一个块，当其大小小于 16 字节时，将不属于有效负载的一部分的位丢弃。

注： *如果没有有效负载数据，即当 Len(P) = 0 或 Len(P) = Len(T) 时，可以跳过有效负载阶段。*

解密密文 C 时，移除 $LSB_{Len(T)}(C)$ 加密标记信息。

CCM 最后阶段

在此最后一个阶段中，AES 外设会生成 GCM 认证标记并将其存储在 AES_DOUTR 寄存器中。要执行的序列为：

1. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 11 来指示最后阶段。
2. 等待 AES_SR 寄存器的计算完成标志 CCF 置 1。
3. 读取 AES_DOUTR 寄存器四次：此输出即为 CCM 认证标记。
4. 通过将 AES_CR 寄存器中的 CCFC 位设置为 1，将 AES_SR 寄存器中的 CCF 标志清零。
5. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。
6. 对于认证解密，将生成的加密标记与密文中填充的加密标记进行比较。

注： *在此最后阶段，正常读取数据（无交换）。*

当从报头阶段过渡到最后阶段时，不得禁止 AES 外设，否则结果会是错误的。

应用程序必须使用标记长度屏蔽认证标记输出以获取有效标记。

CCM 模式下的挂起/恢复操作

要在标头或有效负载阶段挂起消息处理，请执行以下操作：

1. 如果使用 DMA，则通过将 AES_CR 寄存器中的 DMAINEN 位清零来停止 AES DMA 对 IN FIFO 的数据传输。如果未使用 DMA，确保当前计算已完成，完成由 AES_SR 寄存器的 CCF 标志设置为 1 指示。
2. 在有效负载阶段，如果未使用 DMA，则读取四次 AES_DOUTR 寄存器以保存最后处理的块。如果使用 DMA，等待 AES_SR 寄存器中的 CCF 标志置 1，然后通过将 AES_CR 寄存器中的 DMAOUTEN 输出位清零来停止 DMA 对 OUT FIFO 的数据传输。
3. 通过将 AES_CR 寄存器中的 CCFC 位设置为 1，将 AES_SR 寄存器中的 CCF 标志清零。
4. 将 AES_SUSPxR 寄存器（其中 x 等于 0 到 7）保存到存储器中。
5. 保存 AES_IVRx 寄存器，因为在数据处理过程中，它们相较于初始值发生更改。
6. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。
7. 将当前 AES 配置保存到存储器中（初始化向量寄存器 AES_IVRx 除外）。无需保存密钥寄存器，因为应用程序已知初始密钥值。
8. 如果使用 DMA，则保存 DMA 控制器状态 （指向 IN 数据传输的指针以及剩余字节数等）。在有效负载阶段，也必须保存指向 OUT 数据传输的指针。

要恢复消息处理，请执行以下操作：

1. 如果使用 DMA，则重新配置 DMA 控制器以完成 FIFO IN 传输的其余部分。在有效负载阶段，也必须在 DMA 控制器中配置 FIFO OUT 的其余部分。
2. 确保 AES 外设已禁止（AES_CR 寄存器的 EN 位必须为 0）。
3. 将先前保存在存储器中的挂起寄存器值回写到其相应的 AES_SUSPxR 寄存器中（其中 x 等于 0 到 7）。
4. 将先前保存在存储器中的初始化向量寄存器值写回到其相应的 AES_IVRx 寄存器中。
5. 恢复 AES_CR 和 AES_KEYRx 寄存器中的初始化设置值。
6. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
7. 如果使用 DMA，将 AES_CR 寄存器中的 DMAINEN 位置 1（如果在有效负载阶段，还要将 DMAOUTEN 位置 1），以使能 AES DMA 请求。

22.4.13 AES 数据寄存器和数据交换

数据输入和输出

通过将四个连续的 32 位字写入 AES_DINR 寄存器（位域 DIN[127:0]），将 128 位数据块输入到 AES 外设，先最高有效字（位 [127:96]），后最低有效字（位 [31:0]）。

通过从 AES_DOUTR 寄存器（位域 DOUT[127:0]）读取四个连续的 32 位字，从 AES 外设检索 128 位数据块，先最高有效字（位 [127:96]），后最低有效字（位 [31:0]）。

写入 AES_DINR 寄存器或从 AES_DOUTR 寄存器读取的 32 位数据字使用大端模式进行组织，即：

- 要写入 AES_DINR 的字的最高有效字节必须存储在保留要写入的字的四个相邻存储单元的最低地址中，或
- 从 AES_DOUTR 读取的字的最高有效字节存储在接收字的四个相邻存储单元的最低地址中

要使用 DMA 将输入数据块写入 AES，输入块的四个字必须以大端模式连续存储在存储器中，即最高有效字存储在最低地址中。请参见*第 22.4.16 节：AES DMA 接口*。

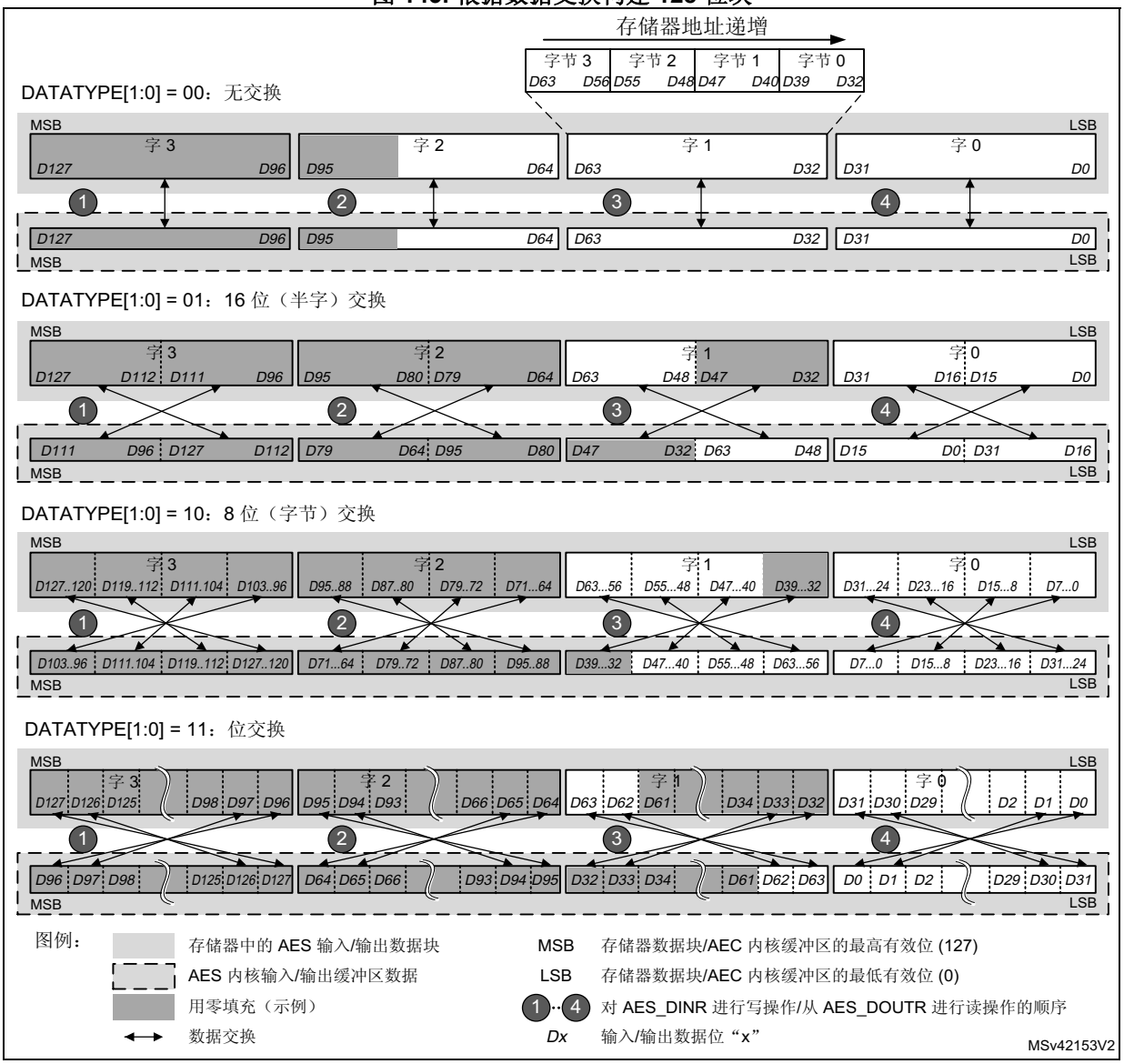
数据交换

可将 AES 外设配置为在将其加载到 AES 处理内核前对 AES_DINR 寄存器中的输入数据字，以及在将其发送到 AES_DOUTR 寄存器前对 AES 处理内核的输出数据，执行位、字节、半字或无交换。选择取决于数据类型。例如，对 ASCII 文本流使用字节交换。

通过 AES_CR 寄存器中的 DATATYPE[1:0] 位域来选择数据交换类型。该选择适用于 AES 内核的输入和输出。

对于不同的数据交换类型，*图 143* 显示了通过 AES_DINR 寄存器输入的输入数据的 AES 处理内核输入缓冲区数据 P127..0 的结构，或通过 AES_DOUTR 寄存器的输出数据的 AES 处理内核输出缓冲区数据 P127..0 的结构。

图 143. 根据数据交换构建 128 位块



注： *AES 密钥寄存器 (AES_KEYRx) 和初始化寄存器 (AES_IVRx) 中的数据对所选交换模式不敏感。*

数据填充

图 143 还给出了一个用零填充存储器数据块的示例，这样数据交换后的补零位在 AES 内核输入缓冲区的 MSB 端形成一个连续区域。该示例显示了输入数据块的填充，包含：

- 48 个消息位，DATATYPE[1:0] = 01
- 56 个消息位，DATATYPE[1:0] = 10
- 34 个消息位，DATATYPE[1:0] = 11

22.4.14 AES 密钥寄存器

AES_KEYRx 寄存器存储加密或解密密钥位域 KEY[127:0] 或 KEY[255:0]。要写入每个寄存器或从中读取的数据以小端模式组织在存储器中，即最高有效字节存储在最高地址中。

密钥分布在八个寄存器中，如*表 112* 中所示。

表 112. AES_KEYRx 寄存器中的密钥字节序（128 位或 256 位密钥长度）

AES_KEYR7 [31:0]	AES_KEYR6 [31:0]	AES_KEYR5 [31:0]	AES_KEYR4 [31:0]	AES_KEYR3 [31:0]	AES_KEYR2 [31:0]	AES_KEYR1 [31:0]	AES_KEYR0 [31:0]
-	-	-	-	KEY[127:96]	KEY[95:64]	KEY[63:32]	KEY[31:0]
KEY[255:224]	KEY[223:192]	KEY[191:160]	KEY[159:128]	KEY[127:96]	KEY[95:64]	KEY[63:32]	KEY[31:0]

AES 外设禁止时，加密或解密的密钥可以写入这些寄存器。

密钥寄存器不受由 AES_CR 寄存器的 DATATYPE[1:0] 位域控制的数据交换影响。

22.4.15 AES 初始化向量寄存器

四个 AES_IVRx 寄存器保持初始化向量输入位域 IVI[127:0]。要写入每个寄存器或从中读取的数据以小端模式组织在存储器中，即最高有效字节存储在最高地址中。寄存器也以最低地址 (AES_IVR0) 到最高地址 (AES_IVR3) 的顺序排序。

位域中数据的有效性取决于选择的链接模式。使用时，位域在 AES 内核的每个计算周期内都会更新。

AES 外设使能时，对 AES_IVRx 寄存器的写入操作对寄存器内容没有影响。要修改 AES_IVRx 寄存器的内容，必须首先清零 AES_CR 寄存器的 EN 位。

读取 AES_IVRx 寄存器会返回最新计数器值（用于管理挂起模式）。

AES_IVRx 寄存器不受由 AES_CR 寄存器的 DATATYPE[1:0] 位域控制的数据交换功能的影响。

22.4.16 AES DMA 接口

AES 外设使用一个接口连接 DMA（直接存储器访问）控制器。DMA 操作通过 AES_CR 寄存器进行控制。

使用 DMA 的数据输入

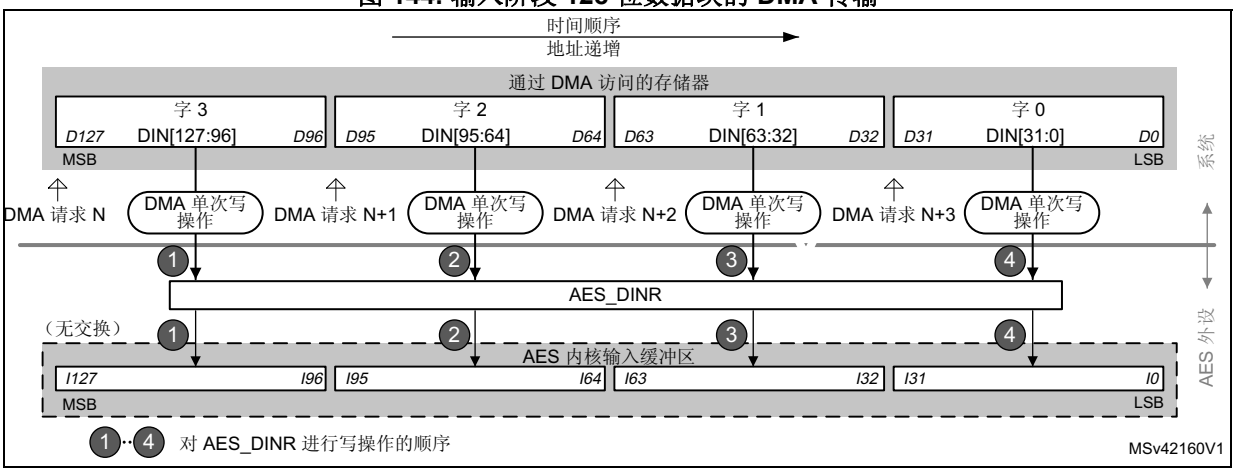
通过将 AES_CR 寄存器的 DMAINEN 位置 1，可使能 DMA 写入 AES。然后，AES 外设每次需要向 AES_DINR 寄存器中写入一个 128 位块（四字）时，都会在输入阶段发出 DMA 请求，如*图 144* 所示。

表 113. 用于存储器到 AES 数据传输的 DMA 通道配置

DMA 通道控制寄存器字段	推荐配置
传输数据量大小	消息长度：128 位的倍数。 根据所选算法和模式，可能需要采用特殊填充/密文窃取技术。例如在 AES GCM 加密或 AES CCM 解密情况下，DMA 传输不得包括最后一个块。有关详细信息，请参见*第 22.4.4 节：执行密码操作的 AES 过程*。
源突发大小（存储器）	单
目标突发大小（外设）	单
DMA FIFO 大小	AES FIFO_size = 4 个字节
源传输宽度（存储器）	32 位字
目标传输宽度（外设）	32 位字
源地址递增（存储器）	是，每次 32 位传输后
目标地址递增（外设）	AES_DINR 的固定地址（无递增）

注： *根据所选算法和模式，可能需要采用特殊填充/密文窃取技术。例如在 AES GCM 加密或 AES CCM 解密情况下，DMA 传输不得包括最后一个块。有关详细信息，请参见第 22.4.4 节：执行密码操作的 AES 过程。*

图 144. 输入阶段 128 位数据块的 DMA 传输



使用 DMA 的数据输出

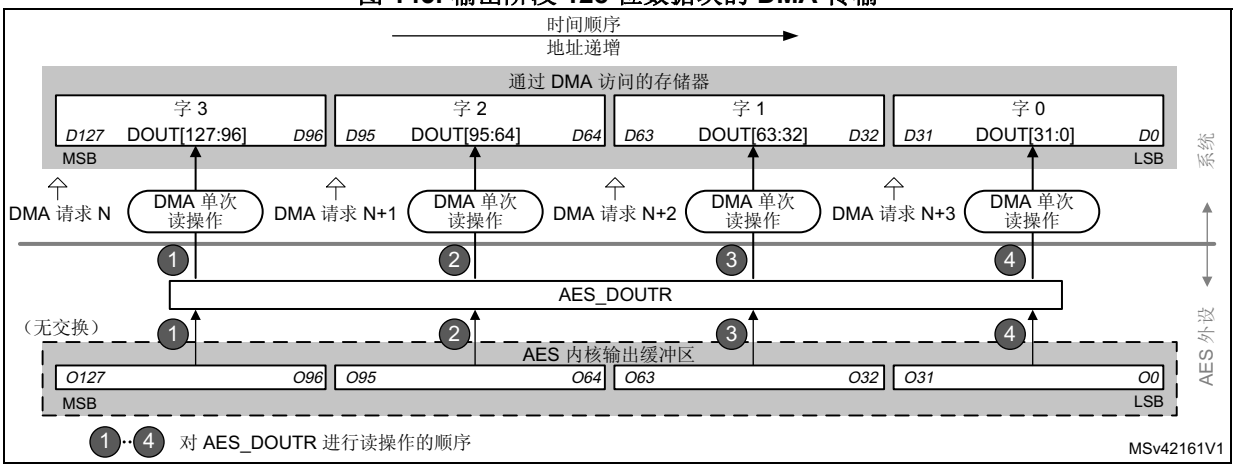
通过将 AES_CR 寄存器的 DMAOUTEN 位置 1，可使能 DMA 从 AES 读取。然后，AES 外设每次需要向 AES_DINR 寄存器中读入一个 128 位块（四字）时，都会在输出阶段发出 DMA 请求，如*图 145* 所示。

表 114. 用于 AES 到存储器数据传输的 DMA 通道配置

DMA 通道控制寄存器字段	推荐配置
传输数据量大小	消息长度，AES 块大小（4 个字）的倍数。根据情况，需要丢弃额外的字节。
源突发大小（外设）	单
目标突发大小（存储器）	单
DMA FIFIO 大小	AES FIFO_size = 4 个字节
源传输宽度（外设）	32 位字
目标传输宽度（存储器）	32 位字
源地址递增（外设）	AES_DINR 的固定地址（无递增）
目标地址递增（存储器）	是，每次 32 位传输后

注：　根据消息大小，应用程序可能需要丢弃最后一个块中的多余字节。

图 145. 输出阶段 128 位数据块的 DMA 传输



不同工作模式下的 DMA 操作

当通过 AES_CR 寄存器的 MODE[1:0] 位域选择模式 1（加密）或模式 3（解密）时，DMA 操作可用。在模式 2（密钥分散）下，AES_KEYRx 寄存器必须通过软件写入，通过 AES_CR 寄存器的 DMAINEN 和 DMAOUTEN 位使能 DMA 传输在该模式下无效。

AES 会一直产生 DMA 单次请求，直到被禁用。因此，128 位数据块处理结束时的数据输出阶段之后，AES 会自动切换到下一个数据块的新数据输入阶段（如果存在）。

当 AES 和存储器之间的数据传输由 DMA 管理时，CCF 标志无意义，可通过软件忽略（置 1）。只有在转换回由软件管理的数据传输时才能清零此标志。请参见*第 22.4.8 节：AES 基本链接模式（ECB 和 CBC）*中的*ECB/CBC 模式下的挂起/恢复操作*作为示例。

22.4.17 AES 错误管理

如果检测到未预期的读取或写入操作，则会将 AES_SR 寄存器中的读取错误标志 (RDERR) 和写入错误标志 (WRERR) 位置 1。如果 AES_CR 寄存器中的错误中断使能 (ERRIE) 位置 1，则会产生中断。有关详细信息，请参见*第 22.5 节：AES 中断*。

注： *检测到错误后，AES 不会禁止，而是会继续处理。*

可随时通过将 AES_CR 寄存器中的 EN 位清零再置 1 来重新初始化 AES。

读取错误标志 (RDERR)

如果在计算阶段或输入阶段检测到未预期的读取操作时，则会将 AES_SR 寄存器中的 AES 读取错误标志 (RDERR) 置 1。如果 AES_CR 寄存器中的 ERRIE 位置 1，则会产生中断。

通过将 AES_CR 寄存器中的相应 ERRC 位置 1 来清零 RDERR 标志。

写入错误标志 (WDERR)

如果在计算阶段或输出阶段检测到未预期的写入操作时，则会将 AES_SR 寄存器中的 AES 写入错误标志 (WRERR) 置 1。如果 AES_CR 寄存器中的 ERRIE 位置 1，则会产生中断。

通过将 AES_CR 寄存器中的相应 ERRC 位置 1 来清零 WDERR 标志。

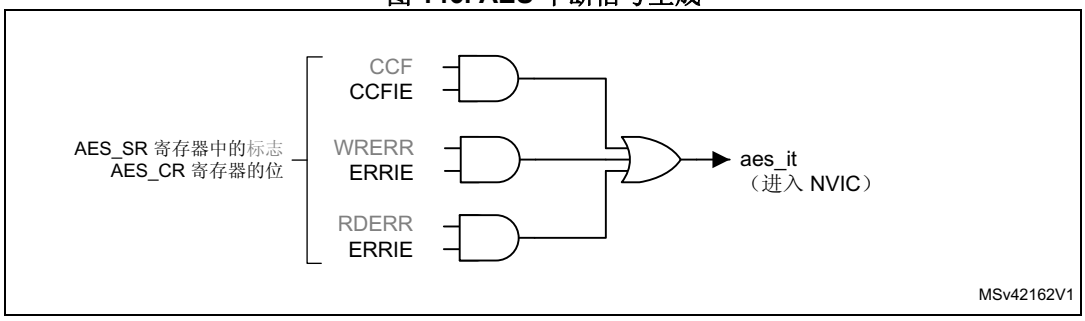
22.5 AES 中断

AES 外设可生成三个独立的可屏蔽中断源，用以通知发生以下事件：

- 计算完成
- 读取错误，请参见*第 22.4.17 节*
- 写入错误，请参见*第 22.4.17 节*

这三个中断源合并为一个连接到 NVIC（嵌套向量中断控制器）的通用中断信号 aes_it。

图 146. AES 中断信号生成



可通过将 AES_CR 寄存器的相应使能位置 1/清零，分别使能/禁止每个 AES 中断源。请参见*图 146*。

可以从 AES_SR 寄存器中读取各个可屏蔽中断源的状态。

表 115 对中断源及其事件标志和使能位进行了总结。

表 115. AES 中断请求

AES 中断事件	事件标志	使能位
计算完成标志	CCF	CCFIE
读取错误标志	RDERR	ERRIE
写入错误标志	WRERR	ERRIE

22.6 AES 处理延迟

下表列出了每种工作模式下处理 128 位块的延迟。

表 116. ECB、CBC 和 CTR 模式下的处理延迟（以时钟周期计）

密钥大小	工作模式	算法	输入阶段 + FSM 集	计算阶段	输出阶段	总计
128 位	模式 1：加密	ECB、CBC、CTR	9	38	4	**51**
	模式 2：密钥分散	-	-	59	-	**59**
	模式 3：解密	ECB、CBC、CTR	9	38	4	**51**
	模式 4：密钥分散和解密	ECB、CBC	9	93	4	**106**
256 位	模式 1：加密	ECB、CBC、CTR	13	58	4	**75**
	模式 2：密钥分散	-	-	82	-	**82**
	模式 3：解密	ECB、CBC、CTR	13	58	4	**75**
	模式 4：密钥分散和解密	ECB、CBC	13	128	4	**145**

表 117. GCM 和 CCM 模式下的处理延迟（以时钟周期计）

密钥大小	工作模式	算法	初始化阶段	标头阶段	有效负载阶段	标记阶段
128 位	模式 1：加密 / 模式 3：解密	GCM	64	35	51	59
		CCM	63	55	114	58
256 位	模式 1：加密 / 模式 3：解密	GCM	88	35	75	75
		CCM	87	79	162	82

注： *数据插入可以包括 AES 在 AHB 总线上强制的等待状态（最多 3 个周期，通常为 1 个周期）。这适用于所有标头 / 有效负载 / 标记阶段。*

22.7 AES 寄存器

22.7.1 AES 控制寄存器 (AES_CR)

AES control register

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NPBLB[3:0]				Res.	KEYSIZE	Res.	CHMOD[2]
								rw	rw	rw	rw		rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	GCMPH[1:0]		DMAOUTEN	DMAINEN	ERRIE	CCFIE	ERRC	CCFC	CHMOD[1:0]		MODE[1:0]		DATATYPE[1:0]		EN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:20 **NPBLB[3:0]**：最后一个块中填充字节的数量 (Number of padding bytes in last block)
此位域用于设置有效负载的最后一个块中填充字节的数量：
0000：所有字节均有效（无填充）
0001：填充最后一个块的一个最低有效字节
...
1111：填充最后一个块的 15 个最低有效字节

位 19 保留，必须保持复位值。

位 18 **KEYSIZE**：密钥大小选择 (Key size selection)
此位域定义了 AES 加密内核中使用的密钥长度（以位为单位）。
0：128
1：256

仅当 AES 禁止时才允许更改位值，以避免不可预测的行为。

位 17 保留，必须保持复位值。

位 16 **CHMOD[2]**：链接模式选择，位 [2] (Chaining mode selection, bit [2])
有关 CHMOD[2:0] 位域的说明，请参见寄存器的位 [5:6]

位 15 保留，必须保持复位值。

位 14:13 **GCMPH[1:0]**：GCM 或 CCM 阶段选择 (GCM or CCM phase selection)
此位域选择 GCM、GMAC 或 CCM 算法的阶段：
00：初始化阶段
01：标头阶段
10：有效负载阶段
11：最后阶段

如果选择 GCM、GMAC 或 CCM 以外的算法（通过 ALGOMODE 位域），则此位域不起作用。

位 12 **DMAOUTEN**：DMA 输出使能 (DMA output enable)

此位在输出阶段使能/禁止使用 DMA 进行数据传输：

0：禁止

1：使能

此位置 1 时，AES 会在输出数据阶段自动生成 DMA 请求。该功能仅在通过 MODE[1:0] 位域选择了模式 1 或模式 3 时才有效。它不适用于模式 2（密钥分散）。

不建议在模式 4 下（单次解密）使用 DMA。

位 11 **DMAINEN**：DMA 输入使能 (DMA input enable)

此位在输入阶段使能/禁止使用 DMA 进行数据传输：

0：禁止

1：使能

此位置 1 时，AES 会在输入数据阶段自动生成 DMA 请求。该功能仅在通过 MODE[1:0] 位域选择了模式 1 或模式 3 时才有效。它不适用于模式 2（密钥分散）。

不建议在模式 4 下（单次解密）使用 DMA。

位 10 **ERRIE**：错误中断使能 (Error interrupt enable)

当 RDERR 和/或 WRERR 置 1 时，此位使能或禁止（屏蔽）AES 中断：

0：禁止（屏蔽）

1：使能

位 9 **CCFIE**：CCF 中断使能 (CCF interrupt enable)

当 CCF（计算完成标志）置 1 时，此位使能或禁止（屏蔽）AES 中断：

0：禁止（屏蔽）

1：使能

位 8 **ERRC**：错误标志清零 (Error flag clear)

写入 1 后，此位将 AES_SR 寄存器中的 RDERR 和 WRERR 错误标志清零：

0：无影响

1：清零 RDERR 和 WRERR 标志

读取标志始终返回零。

位 7 **CCFC**：计算完成标志清零 (Computation complete flag clear)

写入 1 后，此位将 AES_SR 寄存器中的计算完成标志 (CCF) 清零：

0：无影响

1：清零 CCF

读取标志始终返回零。

位 6:5 **CHMOD[1:0]**：链接模式选择，位 [1:0] (Chaining mode selection, bits [1:0])
这些位与位 CHMOD[2]（请参见此寄存器的位 16）一起构成 CHMOD[2:0] 位域，用于选择 AES 链接模式：
000：电子密码本 (ECB)
001：密码块链接 (CBC)
010：计数器模式 (CTR)
011：Galois 计数器模式 (GCM) 和 Galois 消息认证码 (GMAC)
100：CBC-MAC 计数器模式 (CCM)
>100：保留

仅当 AES 禁止时才允许更改位域值，以避免不可预测的行为。

位 4:3 **MODE[1:0]**：AES 工作模式 (AES operating mode)
此位域选择 AES 工作模式：
00：模式 1：加密
01：模式 2：密钥分散（或针对 ECB/CBC 解密准备密钥）
10：模式 3：解密
11：模式 4：密钥分散和单次解密

仅当 AES 禁止时才允许更改位域值，以避免不可预测的行为。在未选择 ECB 或 CBC 链接模式时选择模式 4 的任何尝试，都将默认为有效选择模式 3。在模式 4 之后不能选择模式 3。

位 2:1 **DATATYPE[1:0]**：数据类型选择 (Data type selection)
此位域通过选择数据交换模式，定义了写入 AES_DINR 寄存器或从 AES_DOUTR 寄存器中读取的数据的格式：
00：无
01：半字（16 位）
10：字节（8 位）
11：位

有关详细信息，请参见*第 22.4.13 节：AES 数据寄存器和数据交换*。
仅当 AES 禁止时才允许更改位域值，以避免不可预测的行为。

位 0 **EN**：AES 使能 (AES enable)
此位用于使能/禁止 AES 外设：
0：禁止
1：使能

可随时通过将此位清零再置 1 来重新初始化 AES 外设。
当密钥准备过程结束时（模式 2），此位由硬件自动清零。

22.7.2 AES 状态寄存器 (AES_SR)

AES status register

偏移地址：0x04

复位值：0x0000 0000

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

位 31:4 保留，必须保持复位值。

位 3 **BUSY**：忙 (Busy)

该标志指示 AES 在 GCM 有效负载**加密**阶段是空闲还是繁忙：
0：空闲
1：忙

该标志由硬件控制。当标志指示“空闲”时，可以挂起当前消息处理以处理更高优先级的消息。
该标志仅在 GCM 有效负载加密阶段有效。在其他链接模式下，或在除有效负载加密外的 GCM 阶段，必须忽略该标志。

位 2 **WRERR**：写入错误 (Write error)

此标志指示检测到 AES_DINR 寄存器中发生了未预期的写入操作（计算阶段或数据输出阶段）：
0：未检测到
1：已检测到

该标志由硬件置 1。此位用软件清零，方法是将 AES_CR 寄存器中的 ERRC 位置 1。
标志置 1 时，如果通过 AES_CR 寄存器的 ERRIE 位使能，则会产生中断。
标志置 1 对 AES 操作没有影响。
当选择密钥分散模式或 GCM/CCM 初始化阶段时，该标志无效。

位 1 **RDERR**：读取错误标志 (Read error flag)

此标志指示检测到 AES_DOUTR 寄存器中发生了未预期的读取操作（计算阶段或数据输入阶段）：
0：未检测到
1：已检测到

该标志由硬件置 1。此位用软件清零，方法是将 AES_CR 寄存器中的 ERRC 位置 1。
标志置 1 时，如果通过 AES_CR 寄存器的 ERRIE 位使能，则会产生中断。
标志置 1 对 AES 操作没有影响。
当选择密钥分散模式或 GCM/CCM 初始化/标头阶段时，该标志无效。

位 0 **CCF**：计算完成标志 (Computation completed flag)

该标志指示计算是否完成：
0：未完成
1：已完成

计算完成后，该标志由硬件置 1。此位用软件清零，方法是将 AES_CR 寄存器中的 CCFC 位置 1。
标志置 1 时，如果通过 AES_CR 寄存器的 CCFIE 位使能，则会产生中断。
仅当 DMAOUTEN 位为 0 时，该标志才有效。DMA_EN 为 1 时，此位可能保持为 1。

22.7.3 AES 数据输入寄存器 (AES_DINR)

AES data input register

偏移地址：0x08

复位值：0x0000 0000

仅支持 32 位访问类型。

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

位 31:0 **DIN[x+31:x]**：写入外设的 128 位输入数据块的四个 32 位字之一。

此位域响应一个 32 位输入缓冲区。在输入阶段，对此位域进行 4 倍连续写入，几乎可将一个完整的 128 位输入数据块写入 AES 外设。在每次写入时，数据交换块根据 DATATYPE[1:0] 位域对输入缓冲区的数据进行处理，然后写入 AES 内核 128 位输入缓冲区。

第一次到第四次写入操作分别将“x”替换为：96、64、32 和 0。也就是说，第一次到第四次写入操作的数据为：DIN[127:96]、DIN[95:64]、DIN[63:32] 和 DIN[31:0]。

输入数据块的数据有效性取决于 AES 工作模式：

- **模式 1**（加密）：明文
- **模式 2**（密钥分散）：未使用此位域（将 AES_KEYRx 寄存器用于输入）
- **模式 3**（解密）和**模式 4**（密钥分散和单次解密）：密文

*第 594 页的第 22.4.13 节：AES 数据寄存器和数据交换*对数据交换操作进行了介绍。

22.7.4 AES 数据输出寄存器 (AES_DOUTR)

AES data output register

偏移地址：0x0C

复位值：0x0000 0000

仅支持 32 位访问类型。

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

位 31:0 **DOUT[x+31:x]**：从外设读取的 128 位输出数据块的四个 32 位字之一。

该只读位域获取一个 32 位输出缓冲区。计算完成后（CCF 置 1），对此位域进行 4 倍连续读取，几乎可从 AES 外设读取一个完整的 128 位输出数据块。在到达输出缓冲区之前，数据交换块根据 DATATYPE[1:0] 位域对 AES 内核产生的数据进行处理。

第一次到第四次读取操作分别将 DOUT[x+31:x] 替换为：96、64、32 和 0。也就是说，第一次到第四次读取操作的数据为：DOUT[127:96]、DOUT[95:64]、DOUT[63:32] 和 DOUT[31:0]。

输出数据块的数据有效性取决于 AES 工作模式：

- **模式 1**（加密）：密文
- **模式 2**（密钥分散）：未使用此位域（将 AES_KEYRx 寄存器用于输出）
- **模式 3**（解密）和**模式 4**（密钥分散和单次解密）：明文

*第 594 页的第 22.4.13 节：AES 数据寄存器和数据交换*对数据交换操作进行了介绍。

22.7.5 AES 密钥寄存器 0 (AES_KEYR0)

AES key register 0

偏移地址：0x10

复位值：0x0000 0000

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

位 31:0 **KEY[31:0]**：加密密钥 (Cryptographic key)，位 [31:0]

此位域包含 AES 加密或解密密钥的位 [31:0]，具体取决于工作模式：

- **模式 1**（加密）、**模式 2**（密钥分散）和**模式 4**（密钥分散和单次解密）：写入位域的值是加密密钥。
- **模式 3**（解密）：写入位域的值是在用于解密之前分散的加密密钥。将加密密钥写入位域后，在使能 AES 之前的读取返回相同的值。使能 AES 后且将 CCF 标志置 1 后的读取返回从加密密钥分散的解密密钥。

注： 在模式 4（密钥分散和单次解密）中，位域总是包含加密密钥。

只有在禁止 AES 外设时，才能向 AES_KEYRx 寄存器执行写操作。

更多详细信息，请参见*第 596 页的第 22.4.14 节：AES 密钥寄存器*。

22.7.6 AES 密钥寄存器 1 (AES_KEYR1)

AES key register 1

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[63:48]															
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
KEY[47:32]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[63:32]**：加密密钥 (Cryptographic key)，位 [63:32]
有关 KEY[255:0] 位域的说明，请参见 AES_KEYR0 寄存器。

22.7.7 AES 密钥寄存器 2 (AES_KEYR2)

AES key register 2

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[95:80]															
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
KEY[79:64]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[95:64]**：加密密钥 (Cryptographic key)，位 [95:64]
有关 KEY[255:0] 位域的说明，请参见 AES_KEYR0 寄存器。

22.7.8 AES 密钥寄存器 3 (AES_KEYR3)

AES key register 3

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
KEY[127:112]															
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
KEY[111:96]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[127:96]**：加密密钥 (Cryptographic key)，位 [127:96]
有关 KEY[255:0] 位域的说明，请参见 AES_KEYR0 寄存器。

22.7.9 AES 初始化向量寄存器 0 (AES_IVR0)

AES initialization vector register 0

偏移地址：0x20

复位值：0x0000 0000

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

位 31:0 **IVI[31:0]**：初始化向量输入 (Initialization vector input)，位 [31:0]
有关 IVI[127:0] 位域的说明，请参见 *第 596 页的第 22.4.15 节：AES 初始化向量寄存器*。
初始化向量仅用于除 ECB 以外的链接模式。
只有在禁止 AES 外设时，才能向初始化向量执行写操作。

22.7.10 AES 初始化向量寄存器 1 (AES_IVR1)

AES initialization vector register 1

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[63:48]															
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
IVI[47:32]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[63:32]**：初始化向量输入 (Initialization vector input)，位 [63:32]
有关 IVI[128:0] 位域的说明，请参见 AES_IVR0 寄存器。

22.7.11 AES 初始化向量寄存器 2 (AES_IVR2)

AES initialization vector register 2

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
IVI[95:80]															
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
IVI[79:64]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[95:64]**：初始化向量输入 (Initialization vector input)，位 [95:64]
有关 IVI[128:0] 位域的说明，请参见 AES_IVR0 寄存器。

22.7.12 AES 初始化向量寄存器 3 (AES_IVR3)

AES initialization vector register 3

偏移地址：0x2C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[127:112]															
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
IVI[111:96]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[127:96]**：初始化向量输入 (Initialization vector input)，位 [127:96]
有关 IVI[128:0] 位域的说明，请参见 AES_IVR0 寄存器。

22.7.13 AES 密钥寄存器 4 (AES_KEYR4)

AES key register 4

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[159:144]															
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
KEY[143:128]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[159:128]**：加密密钥 (Cryptographic key)，位 [159:128]
有关 KEY[255:0] 位域的说明，请参见 AES_KEYR0 寄存器。

22.7.14 AES 密钥寄存器 5 (AES_KEYR5)

AES key register 5

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[191:176]															
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
KEY[175:160]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[191:160]**：加密密钥 (Cryptographic key)，位 [191:160]
有关 KEY[255:0] 位域的说明，请参见 AES_KEYR0 寄存器。

22.7.15 AES 密钥寄存器 6 (AES_KEYR6)

AES key register 6

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[223:208]															
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
KEY[207:192]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[223:192]**：加密密钥 (Cryptographic key)，位 [223:192]
有关 KEY[255:0] 位域的说明，请参见 AES_KEYR0 寄存器。

22.7.16 AES 密钥寄存器 7 (AES_KEYR7)

AES key register 7

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[255:240]															
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
KEY[239:224]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[255:224]**：加密密钥 (Cryptographic key)，位 [255:224]
有关 KEY[255:0] 位域的说明，请参见 AES_KEYR0 寄存器。

注： *仅当选择 256 位密钥长度时，才使用密钥寄存器 4 到密钥寄存器 7。当选择 128 位密钥长度时，它们不起作用（此时仅使用密钥寄存器 0 到密钥寄存器 3）。*

22.7.17 AES 挂起寄存器 (AES_SUSPxR)

AES suspend registers

偏移地址：0x040 + x * 0x4，（x = 0 到 7）

复位值：0x0000 0000

当前任务的 AES 处理被挂起以处理更高优先级的任务时，这些寄存器包含 AES 处理器的完整内部寄存器状态。
挂起后，软件在将 AES 处理器用于更高优先级的任务前，读取 AES_SUSPxR 寄存器内容（其中 x 等于 0 到 7）并将其保存到存储器中。完成后，软件在恢复原始任务前将保存的内容恢复到相应的挂起寄存器中。

注： *这些寄存器仅在选择了 GCM、GMAC 或 CCM 链接模式时才可使用。*

仅当 AES 使能时才能读取这些寄存器。当 AES 禁止时读取这些寄存器将返回 0x0000 0000。

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

位 31:0 **SUSPx**：AES 挂起 (AES suspend)
挂起操作后，每个 AES_SUSPxR 寄存器的此位域取内部 AES 寄存器之一的值。

22.7.18 AES 寄存器映射

表 118. AES 寄存器映射和复位值

偏移	寄存器	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
0x0000	AES_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NPBLB				Res.	KEYSIZE	Res.	CHMOD[2]	Res.	GCMPH[1:0]		DMAOUTEN	DMAINEN	ERRIE	CCFIE	ERRC	CCFC	CHMOD[1:0]		MODE[1:0]		DATATYPE[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
0x0004	AES_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	WRERR	RDERR	CCF
	Reset value																													0	0	0	0
0x0008	AES_DINR x=96,64,32,0	DIN[x+31:x]																															
	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
0x000C	AES_DOUTR x=96,64,32,0	DOUT[x+31:x]																															
	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
0x0010	AES_KEYR0	KEY[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
0x0014	AES_KEYR1	KEY[63:32]																															
	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
0x0018	AES_KEYR2	KEY[95:64]																															
	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
0x001C	AES_KEYR3	KEY[127:96]																															
	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
0x0020	AES_IVR0	IVI[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
0x0024	AES_IVR1	IVI[63:32]																															
	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
0x0028	AES_IVR2	IVI[95:64]																															
	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
0x002C	AES_IVR3	IVI[127:96]																															
	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
0x0030	AES_KEYR4	KEY[159:128]																															
	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
0x0034	AES_KEYR5	KEY[191:160]																															
	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
0x0038	AES_KEYR6	KEY[223:192]																															
	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

表 118. AES 寄存器映射和复位值（续）

偏移	寄存器	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
0x003C	AES_KEYR7	KEY[255:224]																															
	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
0x0040	AES_SUSP0R	SUSP0[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
0x0044	AES_SUSP1R	SUSP1[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
0x0048	AES_SUSP2R	SUSP2[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
0x004C	AES_SUSP3R	SUSP3[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
0x0050	AES_SUSP4R	SUSP4[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
0x0054	AES_SUSP5R	SUSP5[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
0x0058	AES_SUSP6R	SUSP6[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
0x005C	AES_SUSP7R	SUSP7[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

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

23 公钥加速器 (PKA)

23.1 简介

公钥加速器 (PKA) 用于计算公钥加解密原语，特别是与 RSA、Diffie-Hellmann 或 GF(p)（Galois 域）上的 ECC（椭圆曲线加密）相关的公钥加解密原语。为了能够以合理的成本实现高性能，上述运算将在 Montgomery 域内执行。

全部必要计算均在此加速器内执行，因此无需其他硬件/软件对输入或输出进行后期处理。

23.2 PKA 主要特性

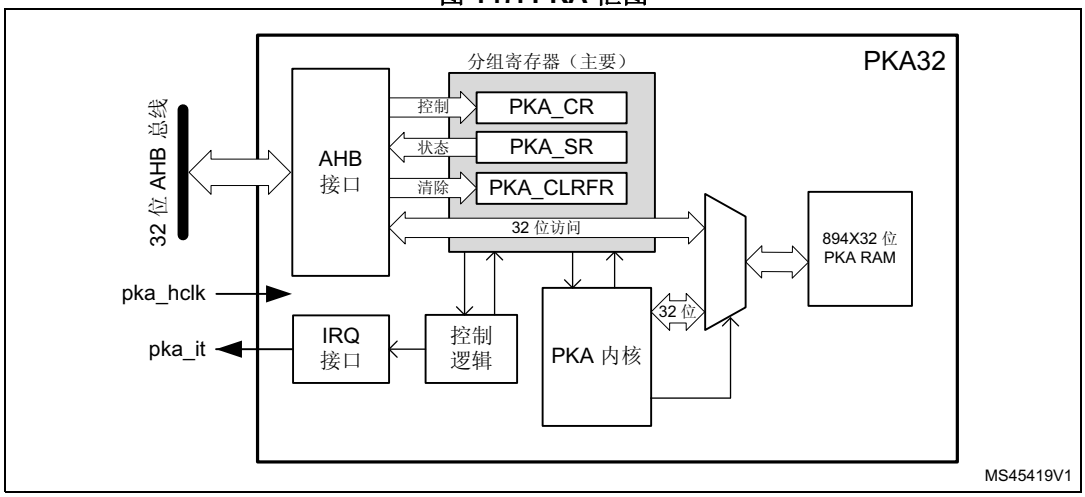
- 加速 RSA、DH 和 GF(p) 上的 ECC 运算，基于 Montgomery 方法实现快速模乘。具体包括：
  - RSA 模幂、RSA 中国剩余定理 (CRT) 求幂
  - ECC 标量乘法、检查点是否在曲线上
  - ECDSA 签名生成与验证
- 能够处理多达 3136 位（针对 RSA/DH）和 640 位（针对 ECC）的操作数。
- 支持算术运算和模运算，例如加法、减法、乘法、模约简、模逆、比较和 Montgomery 乘法。
- 内置 Montgomery 域内向和外向转换。
- AMBA AHB 从外设，仅支持 32 位字单次访问（否则，对于写操作，会生成 AHB 总线错误，并会忽略写访问）。

23.3 PKA 功能说明

23.3.1 PKA 框图

图 147 显示了公钥加速器 PKA 的框图。

图 147. PKA 框图



23.3.2 PKA 内部信号

表 119 列出了 IP 级所提供的内部信号（产品焊盘上不一定提供）。

表 119. 内部输入/输出信号

信号名称	信号类型	说明
pka_hclk	数字输入	AHB 总线时钟
pka_it	数字输出	公钥加速器 IP 全局中断请求

23.3.3 PKA 公钥加速

概述

公钥加速器 (PKA) 用于加速 Rivest、Shamir 和 Adleman (RSA)、Diffie-Hellman (DH) 以及素域上的椭圆曲线加密 (ECC) 运算。支持的操作数大小多达 3136 位（针对 RSA 和 DH）和 640 位（针对 ECC）。

使用一个名为 PKA RAM 的 3576 字节（894 字，每个字 32 位）存储器向 PKA 提供初始数据，并在计算完成后保存结果。通过 PKA AHB 接口进行访问。

PKA 工作模式

表 120 和*表 121* 分别针对整数运算函数和素域 (Fp) 椭圆曲线函数详细列出了 PKA 可执行的运算。

其中的每个运算模式都具有必须写入 PKA_CR 寄存器中的 MODE 位域的相关代码。

表 120. PKA 整数算术函数列表

PKA_CR.MODE[5:0]		执行的运算	参考
十六进制	二进制		
0x01	000001	Montgomery 参数计算 R2 mod n	*第 23.4.2 节*
0x0E	001110	模加 (A+B) mod n	*第 23.4.3 节*
0x0F	001111	模减 (A-B) mod n	*第 23.4.4 节*
0x10	010000	Montgomery 乘法 (AxB) mod n	*第 23.4.5 节*
0x00	000000	模幂 A^e mod n	*第 23.4.6 节*
0x02	000010	模幂 A^e mod n（快速模式）	
0x08	001000	模逆 A-1 mod n	*第 23.4.7 节*
0x0D	001101	模约简 A mod n	*第 23.4.8 节*
0x09	001001	算术加法 A+B	*第 23.4.9 节*
0x0A	001010	算术减法 A-B	*第 23.4.10 节*
0x0B	001011	算术乘法 AxB	*第 23.4.11 节*
0x0C	001100	算术比较（A=B、A>B、A<B）	*第 23.4.12 节*
0x07	000111	RSA CRT 求幂	*第 23.4.13 节*

表 121. PKA 素域 (Fp) 椭圆曲线函数列表

PKA_CR.MODE[5:0]		执行的运算	参考
十六进制	二进制		
0x28	101000	检查点是否在椭圆曲线 Fp 上	*第 23.4.14 节*
0x20	100000	ECC 标量乘法 kP	*第 23.4.15 节*
0x22	100010	ECC 标量乘法 kP（快速模式）	
0x24	100100	ECDSA 签名	*第 23.4.16 节*
0x26	100110	ECDSA 验证	*第 23.4.17 节*

Montgomery 空间与快速模式运算

为了提高效率，PKA 在 Montgomery 域内部执行模乘运算，以便自动执行内向和外向转换。

由于 Montgomery 参数计算十分耗时，因此应用程序可决定使用快速运算模式，在此运算模式下会提供预先计算好的 Montgomery 参数。*第 23.5.2 节：计算时间*对性能改进情况进行了详细介绍。

23.3.4 PKA 的典型应用

简介

PKA 可用于加速多种公钥加密函数。尤其是：

- RSA 加密和解密
- RSA 密钥完成
- CRT-RSA 解密
- DSA 和 ECDSA 签名生成与验证
- DH 和 ECDH 密钥协议

以下出版物给出了上述函数的相关规范：

- FIPS PUB 186-4，数字签名标准 (DSS)，2013 年 7 月，NIST 著
- PKCS #1，RSA 加密标准，v1.5、v2.1 和 v2.2。RSA 实验室著
- IEEE1363-2000，公钥加密的 IEEE 标准规范，2000 年 1 月
- ANSI X9.62-2005，金融服务业用公钥加密：椭圆曲线数字签名算法 (ECDSA)，2005 年 11 月

本节介绍主要函数的原理，有关更详细的说明，请参见上面引用的文档。

RSA 密钥对

对于以下 RSA 运算，公钥和私钥信息定义如下：

- Alice 将其公钥 (*n*, *e*) 传送给 Bob。数字 *n* 和 *e* 均为非常大的正整数。
- Alice 对其私钥 d（也是一个非常大的正整数）进行保密。或者，该私钥也可以由一个五元组 (p, q, dp, dq, qInv) 表示。

有关上述说明的更多信息，请参见 RSA 规范。

RSA 加密/解密原理

根据 PKCS#1 规范的建议，如果 Bob 要使用 Alice 的公钥 (*n*, *e*) 加密消息 M，必须经过以下步骤：

1. 计算编码消息 EM = ENCODE(M)，其中 ENCODE 为编码方法。
2. 将 EM 转换为整数 *m*，$0 \le m < n$，且 (*m*, *n*) 互质。
3. 计算密文 $c = m^e \bmod n$。
4. 将整数 c 转换为字符串密文 C。

如果 Alice 要使用其私钥来解密密文 c，则应按以下步骤操作：

1. 将密文 C 转换为整数密文 c。
2. 还原明文 $m = c^d \bmod n = (m^e)^d \bmod n$。如果私钥是一个五元组 (p, q, dp, dq, qInv)，则通过执行以下运算获得明文 *m*：
   a) $m_1 = c^{dp} \bmod p$
   b) $m_2 = c^{dq} \bmod q$
   c) $h = qInv\ (m_1 - m_2) \bmod p$
   d) $m = m_2 + h\ q$
3. 将整数消息 *m* 转换为编码的消息 EM。
4. 还原消息 M = DECODE(EM)，其中 DECODE 为解码方法。

PKA 可使用 *模幂* $A^e \bmod n$（私钥为 *d* 时）或 *RSA CRT 求幂*（私钥为一个五元组 (p, q, dp, dq, qInv) 时）来加速上述运算。

注： *PKCS#1 标准中规定了消息与整数之间的解码运算和转换运算。*

椭圆曲线选择

对于以下 ECC 运算，曲线参数定义如下：

- 曲线对应于参与方（Alice 和 Bob）之间约定的椭圆曲线域。*第 23.5.1 节：曲线配置*汇总了支持的曲线参数。
- G 是选择的椭圆曲线基点（也称为发生器），具有较大素数阶 n（即，*n* x G = 单位元素 O）。

ECDSA 消息签名生成

ECDSA（椭圆曲线数字签名算法）签名生成函数原理如下：如果 Alice 要使用其私钥整数 d_A 来签名消息 *m*，则需按照以下步骤操作。

1. 计算 e = HASH(*m*)，其中 HASH 为加解密级别的散列函数。
2. 使 z 成为 e 最左边的 L_n 个位，其中 L_n 为群阶 *n* 的位长度。
3. 选择一个经过加解密级别的安全的随机整数 k，其中 $0 < k < n$。
4. 计算曲线点 $(x_1,y_1) = k \times G$。
5. 计算 $r = x_1 \bmod n$。如果 $r = 0$，则返回至步骤 3。
6. 计算 $s = k^{-1} (z + rd_A) \bmod n$。如果 $s = 0$，则返回至步骤 3。
7. 签名为 (*r*, *s*) 对。

PKA 可通过以下方式加速步骤 4 到 7：

- *ECDSA 签名*或
- 以下所有运算：
  - *ECC Fp 标量乘法* k x P
  - *模约简* A mod *n*
  - *模逆* $A^{-1} \bmod n$
  - *模加*和*模数和 Montgomery 乘法*

ECDSA 签名验证

ECDSA（椭圆曲线数字签名算法）签名验证函数原理如下：如果 Bob 要验证 Alice 的签名，则必须拥有其公钥曲线点 Q_A 的副本。

Bob 可通过以下步骤验证 Q_A 是否为有效的曲线点：

1. 检查 Q_A 是否不等于单位元素 O
2. 检查 Q_A 是否处于约定的曲线上
3. 检查 n x Q_A = O

然后，Bob 执行下面详述的步骤：

1. 验证 *r* 和 *s* 是否为 [1, n-1] 内的整数
2. 计算 *e* = HASH(*m*)，其中 HASH 为约定的加密散列函数
3. 使 z 成为 *e* 最左边的 L_n 个位
4. 计算 w = s^{-1} mod *n*
5. 计算 u_1 = zw mod *n* 以及 u_2 = *rw* mod *n*
6. 计算曲线点 (x_1, y_1) = u_1 x G + u_2 x Q_A
7. 如果 *r* = x_1 (mod *n*)，则签名有效，否则无效

PKA 可通过 *ECDSA 验证*加速步骤 4 到 7。

23.3.5 用于执行运算的 PKA 过程

使能/禁止 PKA

将 PKA_CR 寄存器中的 EN 位置 1 可使能 PKA 外设。当 EN=0 时，PKA 外设保持复位状态，应用程序仍可通过 AHB 接口访问 PKA 存储器。

在计算过程中将 EN 位清零会导致运算中止。在这种情况下，无法保证 PKA 存储器的内容。

数据格式

*第 23.4 节*为每种运算规定了 PKA RAM 中的输入数据和结果的格式。

执行 PKA 运算

按照以下步骤执行支持的每种 PKA 运算：

1. 将初始数据装载至 PKA 内部 RAM（位于偏移地址 0x400 处）。
2. 通过对 PKA_CR 寄存器的 MODE 位域进行写操作来指定要执行的运算，然后将 PKA_CR 寄存器中的 START 位置 1。
3. 等待至 PKA_SR 寄存器中的 PROCENDF 位置 1，表示计算已完成。
4. 从 PKA 内部 RAM 读取结果数据，然后通过将 PKA_CLRFR 中的 PROCENDFC 位置 1 来清零 PROCENDF 位。

注： *PKA 忙 (BUSY = 1) 时，将忽略应用程序对 PKA RAM 进行的所有访问，PKA_SR 中的标志 RAMERRF 将置 1。*

使用预先计算的 Montgomery 参数

如*第 23.3.3 节*所述，当执行多个具有相同模数的运算时，如果应用程序只需计算一次相应的 Montgomery 参数，则会十分有益（例如，请参见*第 23.4.5 节*）。这便是所谓的“快速模式”。

若要管理对快速模式的使用，建议采用如下步骤：

1. 在 PKA RAM 中装入模数大小和值信息。相关信息请参见*第 23.5.1 节*。
2. 在 PKA_CR 寄存器中将 PKA 编程为 *Montgomery 参数计算*模式（MODE=“0x1”），然后将 START 位置 1。
3. 等待至 PKA_SR 寄存器中的 PROCENDF 位置 1，然后从 PKA 存储器读回相应的 Montgomery 参数，并通过将 PKA_CLRFR 中的 PROCENDFC 位置 1 来清零 PROCENDF 位。
4. 继续执行所需 PKA 运算，在常规输入数据之上装载 Montgomery 信息 R2 mod m。*第 23.4 节*给出了所有地址。

23.3.6 PKA 错误管理

使用 PKA 时，可能会出现以下错误：

- 访问 PKA RAM 时超出了预期范围。此时，PKA_SR 寄存器中的地址错误标志 (ADDRERRF) 置 1。
- 对正在被 PKA 内核使用的 PKA RAM 发起 AHB 访问。此时，PKA_SR 寄存器中的 RAM 错误标志 (RAMERRF) 置 1，读取 PKA RAM 将返回零，而写操作将被忽略。

对于上述每个错误标志，如果应用程序将 PKA_CR 寄存器中的相应位置 1，则 PKA 会产生一个中断（有关详细信息，请参见*第 23.6 节*）。

通过将 PKA_CLRFR 中的相应位置 1 来清除 ADDRERRF 和 RAMERRF 错误。

可随时通过将 PKA_CR 寄存器中的 EN 位置 0 来重新初始化 PKA。

23.4 PKA 工作模式

23.4.1 简介

以下各小节介绍 PKA 支持的各种运算及其相关输入数据和结果（均存储在 PKA RAM 中）的格式。

以下信息适用于所有 PKA 运算。

- PKA 内核处理 32 位字
- 支持的操作数“大小”如下：
  - ROS（RSA 操作数大小）：数据大小为 (*rsa_size*/32+1) 个字，其中 *rsa_size* 等于所选模数长度。例如，当计算操作数大小为 1024 位的 RSA 时，ROS 等于 33 个字或 1056 位。
  - EOS（ECC 操作数大小）：数据大小为 (*ecc_size*/32+1) 个字，其中 *ecc_size* 等于所选素数模长度。例如，当计算操作数大小为 192 位的 ECC 时，EOS 等于 7 个字或 224 位。

注： *由于 PKA 内核处理 32 位字，如果上述公式的结果为小数，则会舍入为最接近的整数。*

注： *最大 ROS 为 99 个字（最大指数大小为 3136 位），而最大 EOS 为 21 个字（最大操作数大小为 640 位）。*

- 下表中的“存储地址”列指示寄存器 PKA_CR 或 PKA 内的偏移地址（位于 PKA RAM 区域，从 0x400 开始）。要恢复操作数的物理地址，应用程序必须将指定的偏移量与 PKA 的基址相加。
- 关于写入参数（“IN”方向）
  - 将元素作为输入写入 PKA 的存储器时，必须添加一个所有位均等于零的附加字。例如，使用 ECC P256 时和装载输入（以 256 位或 8 个字表示）时，PKA 需要一个附加字，并且必须全部用零填充。
  - 关于字节顺序，例如要准备进行 ECC Fp 标量乘法运算，当应用程序写 ECC P256 曲线（EOS = 9 个字）的 x_p 坐标时，最低有效位在位 0（偏移地址为 0x55C）；最高有效位在位 31（偏移地址为 0x578）。然后，如上所述，还应在偏移地址 0x57C 处写入字 0x00000000。
  - 除非另有说明，否则表中的所有操作数均为整数。
- 关于 PKA 输出（“OUT”方向）
  - 除非表中另有说明，否则当应用程序写入错误参数时，PKA 不会提供错误输出。

注意： 在发起任何 PKA 运算之前，必须检查 PKA 的所有输入参数的有效性。事实上，PKA 假设所有输入参数均有效且彼此一致。

23.4.2 Montgomery 参数计算

此函数用于计算 PKA 使用的 Montgomery 参数 (R^2 mod *n*)，以将操作数转换为 Montgomery 剩余系统表示形式。

注： *此运算也可与 ECC 曲线搭配使用。此时，应使用素数模长度和 EOS 大小。*

表 122 汇总了 Montgomery 参数计算的运算说明。

表 122. Montgomery 参数计算

带方向的参数		值（注）	存储地址	大小
IN	模式	0x01	PKA_CR	6 位
	模数长度	（以位为单位，0 ≤ 值 < 3136 位）	RAM@0x404	32 位
	模数值 n	（仅限奇数，$n < 2^{3136}$）	RAM@0xD5C	ROS
OUT	结果：R^2 n	-	RAM@0x594	

23.4.3 模加

模加运算的计算公式为 A + B mod *n*。*表 123* 汇总了运算说明。

表 123. 模加

带方向的参数		值（注）	存储地址	大小
IN	模式	0x0E	PKA_CR	6 位
	操作数长度	（以位为单位，非空）	RAM@0x404	32 位
	操作数 A	(0 ≤ A < n)	RAM@0x8B4	ROS
	操作数 B	(0 ≤ B < n)	RAM@0xA44	
	模数值 n	(n < 2^{3136})	RAM@0xD5C	
OUT	结果：A+B mod n	（0 ≤ 结果 < n）	RAM@0xBD0	

23.4.4 模减

模减运算的计算如下：

- 如果 A ≥ B，则结果等于 A - B mod *n*
- 如果 A < B，则结果等于 A + n - B mod *n*

表 124 汇总了运算说明。

表 124. 模减

带方向的参数		值（注）	存储地址	大小
IN	模式	0x0F	PKA_CR	6 位
	操作数长度	（以位为单位，非空）	RAM@0x404	32 位
	操作数 A	(0 ≤ A < n)	RAM@0x8B4	ROS
	操作数 B	(0 ≤ B < n)	RAM@0xA44	
	模数值 n	(n < 2^{3136})	RAM@0xD5C	
OUT	结果：A-B mod n	（0 ≤ 结果 < n）	RAM@0xBD0	

23.4.5 模数和 Montgomery 乘法

为提高乘法序列的执行效率，PKA 会使用 Montgomery 域的输入和输出进行乘法加速。此运算的两个主要用途如下：

- 将值从自然域映射到 Montgomery 域，或反之
- 执行模乘运算 A x B mod *n*

下面介绍了执行上述运算的方法。请注意，“x”函数表示此运算，A、B、C 操作数处于自然域中。

1. Montgomery 域的内向和外向转换
   a) 假设 A 是自然域中的一个整数
      使用 *Montgomery 参数计算*计算 *r2modn*
      结果 AR = A x *r2modn* mod *n* 是 Montgomery 域中的 A
   b) 假设 BR 是 Montgomery 域中的一个整数
      结果 B = BR x 1 mod *n* 为自然域中的 B
      类似地，通过计算 A = AR x 1 mod *n*，可以将 a) 中计算出的上述值 AR 转换为自然域
2. 单模乘法运算 A x B mod *n*
   a) 使用 *Montgomery 参数计算*计算 *r2modn*
   b) 计算 AR = A x *r2modn* mod *n*。输出位于 Montgomery 域
   c) 计算 AB = AR x B mod *n*。输出位于自然域
3. 多模乘法运算 A x B x C mod *n*
   a) 使用 *Montgomery 参数计算*计算 *r2modn*
   b) 计算 AR = A x *r2modn* mod *n*。输出位于 Montgomery 域
   c) 计算 BR = B x *r2modn* mod *n*。输出位于 Montgomery 域
   d) 计算 ABR = AR x BR mod *n*。输出位于 Montgomery 域
   e) 计算 CR = C x *r2modn* mod *n*。输出位于 Montgomery 域
   f) 计算 ABCR = ABR x CR mod *n*。输出位于 Montgomery 域
   g) （可选）如果需要对更多操作数进行乘法运算，请重复上述两个步骤
   h) 计算 ABC = ABCR x 1 mod *n* 以获取自然域中的结果

表 125 汇总了 Montgomery 乘法的运算说明。

表 125. Montgomery 乘法

带方向的参数		值（注）	存储地址	大小
IN	模式	0x10	PKA_CR	6 位
	操作数长度	（以位为单位，非空）	RAM@0x404	32 位
	操作数 A	(0 ≤ A < n)	RAM@0x8B4	ROS
	操作数 B	(0 ≤ B < n)	RAM@0xA44	
	模数值 n	（仅限奇数，$n < 2^{3136}$）	RAM@0xD5C	
OUT	结果：AxB mod n	-	RAM@0xBD0	

23.4.6 模幂

模幂运算通常用于执行单步 RSA 运算。其计算公式为 A^e mod *n*。

表 126（正常模式）和*表 127*（快速模式）汇总了模幂运算的运算说明。*第 23.3.5 节*介绍了快速模式的使用方法。

表 126. 模幂运算（正常模式）

带方向的参数		值（注）	存储地址	大小
IN	模式	0x00	PKA_CR	6 位
IN	指数长度	（以位为单位，非空）	RAM@0x400	32 位
	操作数长度	（以位为单位，非空）	RAM@0x404	
IN/OUT	操作数 A（幂的基数）	(0 ≤ A < n)	RAM@0xA44	ROS
IN	指数 e	(0 ≤ e < n)	RAM@0xBD0	
	模数值 n	（仅限奇数，$n < 2^{3136}$）	RAM@0xD5C	
OUT	结果：$A^e \bmod n$	（0 ≤ 结果 < n）	RAM@0x724	

表 127. 模幂运算（快速模式）

带方向的参数		值（注）	存储地址	大小
IN	模式	0x02	PKA_CR	6 位
IN	指数长度	（以位为单位，非空）	RAM@0x400	32 位
	操作数长度	（以位为单位，非空）	RAM@0x404	
IN/OUT	操作数 A（幂的基数）	(0 ≤ A < n)	RAM@0xA44	ROS
IN	指数 e	(0 ≤ e < n)	RAM@0xBD0	
	模数值 n	（仅限奇数，$n < 2^{3136}$）	RAM@0xD5C	
IN/OUT	Montgomery 参数 R2 mod n	（强制）	RAM@0x594	
OUT	结果：$A^e \bmod n$	（0 ≤ 结果 < n）	RAM@0x724	

23.4.7 模逆

模逆运算的计算公式为乘法逆元 $A^{-1} \bmod n$。如果模数 *n* 为素数，则对于 A 的所有值 (1 ≤ A < *n*)，模逆运算输出均有效。如果模数 *n* 不是素数，则只有当 A 和 *n* 之间的最大公约数为 1 时，A 才可逆。

如果操作数 A 是模数 *n* 的约数，则结果将是 *n* 的因子的倍数。

表 128 汇总了模逆运算的运算说明。

表 128. 模逆

带方向的参数		值（注）	存储地址	大小
IN	模式	0x08	PKA_CR	6 位
	操作数长度	（以位为单位，非空）	RAM@0x404	32 位
	操作数 A	(0 ≤ A < n)	RAM@0x8B4	ROS
	模数值 n	（仅限奇数，$n < 2^{3136}$）	RAM@0xA44	
OUT	结果：$A^{-1} \bmod n$	0 < 结果 < n	RAM@0xBD0	

23.4.8 模约简

模约简运算计算的是 A 被 n 除所得的余数。*表 129* 汇总了运算说明。

表 129. 模约简

带方向的参数		值（注）	存储地址	大小
IN	模式	0x0D	PKA_CR	6 位
	操作数长度	（以位为单位，非空）	RAM@0x400	32 位
	模数长度	（以位为单位，8 < 值 < 3136）	RAM@0x404	
	操作数 A	$(0 \le A < 2n < 2^{3136})$	RAM@0x8B4	ROS
	模数值 n	（仅限奇数，$n < 2^{3136}$）	RAM@0xA44	
OUT	结果：A mod n	（0 < 结果 < n）	RAM@0xBD0	

23.4.9 算术加法

算术加法运算的计算公式为 A + B。*表 130* 汇总了运算说明。

表 130. 算术加法

带方向的参数		值（注）	存储地址	大小
IN	模式	0x09	PKA_CR	6 位
	操作数长度 M	（以位为单位，非空）	RAM@0x404	32 位
	操作数 A	$(0 \le < A < 2^{M})$	RAM@0x8B4	ROS
	操作数 B	$(0 \le < B < 2^{M})$	RAM@0xA44	
OUT	结果：A+B	（$0 \le$ 结果 $< 2^{M+1}$）	RAM@0xBD0	ROS + 1

23.4.10 算术减法

算术减法运算的计算如下：

- 如果 A ≥ B，则结果等于 A - B
- 如果 A < B 且 M/32 余数 > 0，则结果等于 $A + 2^{int(M/32)*32+1} - B$
- 如果 A < B 且 M/32 余数为 0，则结果等于 $A + 2^{int(M/32)*32} - B$

表 131 汇总了运算说明。

表 131. 算术减法

带方向的参数		值（注）	存储地址	大小
IN	模式	0x0A	PKA_CR	6 位
	操作数长度 M	（以位为单位，非空）	RAM@0x404	32 位
	操作数 A	$(0 \le < A < 2^{M})$	RAM@0x8B4	ROS
	操作数 B	$(0 \le < B < 2^{M})$	RAM@0xA44	
OUT	结果：A-B	（$0 \le$ 结果 $< 2^{M}$）	RAM@0xBD0	

23.4.11 算术乘法

算术乘法运算的计算公式为 AxB。*表 132* 汇总了运算说明。

表 132. 算术乘法

带方向的参数		值（注）	存储地址	大小
IN	模式	0x0B	PKA_CR	6 位
	操作数长度 M	（以位为单位，非空）	RAM@0x404	32 位
	操作数 A	$(0 \leq < A < 2^M)$	RAM@0x8B4	ROS
	操作数 B	$(0 \leq < B < 2^M)$	RAM@0xA44	
OUT	结果：AxB	（$0 \leq$ 结果 $< 2^M$）	RAM@0xBD0	2xROS

23.4.12 算术比较

算术比较运算的计算如下：

- 如果 A=B，则结果=0x0
- 如果 A>B，则结果=0x1
- 如果 A<B，则结果=0x2

表 133 汇总了算术比较的运算说明。

表 133. 算术比较

带方向的参数		值（注）	存储地址	大小
IN	模式	0x0C	PKA_CR	6 位
	操作数长度 M	（以位为单位，非空）	RAM@0x404	32 位
	操作数 A	$(0 \leq < A < 2^M)$	RAM@0x8B4	ROS
	操作数 B	$(0 \leq < B < 2^M)$	RAM@0xA44	
OUT	结果：A?B	0x0、0x1 或 0x02	RAM@0xBD0	32 位

23.4.13 RSA CRT 求幂

为了提高效率，许多常用加密库（如 OpenSSL RSA）均根据中国剩余定理 (CRT) 在解密和签名方面进行了以下优化：

- *p* 和 *q* 是预先计算的素数，作为私钥的一部分进行存储
- $d_P = d \bmod (p-1)$
- $d_Q = d \bmod (q-1)$
- $q_{inv} = q^{-1} \bmod p$

这些值有助于接收方更加高效地求幂 $m = A^d \pmod{pq}$，具体如下：

- $m_1 = A^{dP} \bmod p$
- $m_2 = A^{dQ} \bmod p$
- $h = q_{inv} (m_1 - m_2) \bmod p$，其中 $m_1 > m_2$
- $m = m_2 + hq$

表 134 汇总了用于计算 CRT 求幂 $A^d \bmod pq$ 的运算说明。

表 134. CRT 求幂

带方向的参数		值（注）	存储地址	大小
IN	模式	0x07	PKA_CR	6 位
IN	操作数长度	（以位为单位，非空）	RAM@0x404	32 位
IN	操作数 d_P	$(0 \le d_P < 2^{M/2})$	RAM@0x65C	ROS/2
	操作数 d_Q	$(0 \le d_Q < 2^{M/2})$	RAM@0xBD0	
	操作数 q_{inv}	$(0 \le q_{inv} < 2^{M/2})$	RAM@0x7EC	
	素数 p(1)	$(0 \le p < 2^{M/2})$	RAM@0x97C	
	素数 q(1)	$(0 \le q < 2^{M/2})$	RAM@0xD5C	
IN	操作数 A	$(0 \le A < 2^{M/2})$	RAM@0xEEC	ROS
OUT	结果：$A^d \bmod pq$	（$0 \le$ 结果 $< pq$）	RAM@0x724	

1. 不得为 2。

23.4.14 检查点是否在椭圆曲线 Fp 上

此运算检查给定点 P (x, y) 是否在素域曲线上，计算公式为 $y^2 = (x^3 + ax + b) \bmod p$，其中 *a* 和 *b* 是曲线的元素。

表 135 汇总了关于检查点是否在椭圆曲线 Fp 上的运算说明。

表 135. 检查点是否在椭圆曲线 Fp 上

带方向的参数		值（注）	存储地址	大小
IN	模式	0x28	PKA_CR	6 位
	模数长度	（以位为单位，非空， 8 < 值 < 640）	RAM@0x404	32 位
	曲线系数 *a* 符号	0x0：正 0x1：负	RAM@0x408	
	曲线系数 |a|	（绝对值，|a| < p）	RAM@0x40C	EOS
	曲线系数 *b*	(|b| < p)	RAM@0x7FC	
	曲线模数值 *p*	（奇素数，0< p < 2640）	RAM@0x460	
	点 P 坐标 *x*	(x < *p*)	RAM@0x55C	
	点 P 坐标 *y*	(y < *p*)	RAM@0x5B0	
OUT	结果：P 在曲线上	0x0：点在曲线上 非 0x0：点不在曲线上	RAM@0x400	32 位

23.4.15 ECC Fp 标量乘法

此运算的计算公式为 k x P (x_P, y_P)，其中 P 表示素域曲线上的一个点，“x”表示椭圆曲线标量点乘法。计算结果为属于同一曲线的点或无穷远的点。

表 136（正常模式）和*表 137*（快速模式）汇总了 ECC Fp 标量乘法的运算说明。*第 23.3.5 节*介绍了快速模式的使用方法。

表 136. ECC Fp 标量乘法

带方向的参数		值（注）	存储地址	大小
IN	模式	0x20	PKA_CR	6 位
IN	标量乘数 k 长度	（以位为单位，非空，8 < 值 < 640）	RAM@0x400	32 位
	模数长度	（以位为单位，非空，8 < 值 < 640）	RAM@0x404	
	曲线系数 *a* 符号	0x0：正 0x1：负	RAM@0x408	
IN	曲线系数 |a|	（绝对值，|a| < p）	RAM@0x40C	EOS
	曲线模数值 *p*	（奇素数，0< p < 2^{640}）	RAM@0x460	
	标量乘数 k	(0 ≤ k < 2^{640})	RAM@0x508	
	点 P 坐标 x_P	(x < *p*)	RAM@0x55C	
	点 P 坐标 y_P	(y < *p*)	RAM@0x5B0	
OUT	结果：k x P 坐标 *x*	（结果 < *p*）	RAM@0x55C	
	结果：k x P 坐标 *y*	（结果 < *p*）	RAM@0x5B0	

表 137. ECC Fp 标量乘法（快速模式）

带方向的参数		值（注）	存储地址	大小
IN	模式	0x22	PKA_CR	6 位
IN	标量乘数 k 长度	（以位为单位，非空，8 < 值 < 640）	RAM@0x400	32 位
	模数长度	（以位为单位，非空，8 < 值 < 640）	RAM@0x404	
	曲线系数 *a* 符号	0x0：正 0x1：负	RAM@0x408	
IN	曲线系数 |a|	（绝对值，|a| < p）	RAM@0x40C	EOS
	曲线模数值 *p*	（奇素数，0< p < 2^{640}）	RAM@0x460	
	标量乘数 k	(0 ≤ k < 2^{640})	RAM@0x508	
	点 P 坐标 x_P	(x < *p*)	RAM@0x55C	
	点 P 坐标 y_P	(y < *p*)	RAM@0x5B0	
IN	Montgomery 参数 R^2 mod *p*	（强制）	RAM@0x4B4	
OUT	结果：k x P 坐标 *x*	（结果 < *p*）	RAM@0x55C	
	结果：k x P 坐标 *y*	（结果 < *p*）	RAM@0x5B0	

执行此运算时，应注意以下特殊情况：

- 对于 k = 0 的情况，此函数返回无穷远处的点，如果曲线参数 b 非零则返回 (0, 0)，否则返回 (0, 1)。对于 k 不为 0 的情况，可能会返回无穷远处的点。当应用程序检测到此行为时，应重新执行计算。
- 对于 k < 0（即，需要负标量乘法）的情况，应向 PKA 提供乘数绝对值 k = |-k|。计算完成后，可使用公式 -P = (x, -y) 计算有效最终结果的 y 坐标（x 坐标保持不变）。

23.4.16 ECDSA 签名

表 138（输入参数）和*表 139*（输出参数）汇总了 ECDSA 签名运算（在*第 23.3.4 节*进行了概要介绍）。

应用程序应检查输出错误是否等于零，如果不等于零，则应生成新的 k 并应重新执行 ECDSA 签名运算。

表 138. ECDSA 签名——输入

带方向的参数		值（注）	存储地址	大小
IN	模式	0x24	PKA_CR	6 位
	曲线素数阶 n 长度	（以位为单位，非空）	RAM@0x400	32 位
	曲线模数 p 长度	（以位为单位，8 < 值 < 640）	RAM@0x404	
	曲线系数 a 符号	0x0：正 0x1：负	RAM@0x408	
	曲线系数 |a|	（绝对值，|a| < p）	RAM@0x40C	EOS
	曲线模数值 *p*	（奇素数，0< p < 2^{640}）	RAM@0x460	
	整数 k(1)	(0 ≤ k < 2^{640})	RAM@0x508	
	曲线基点 G 坐标 x	(x < *p*)	RAM@0x55C	
	曲线基点 G 坐标 y	(y < *p*)	RAM@0x5B0	
	消息 z 的散列值	(z < 2M)	RAM@0xDE8	
	私钥 d	（正整数）	RAM@0xE3C	
	曲线素数阶 n	（素数）	RAM@0xE94	

1. 此整数通常是加解密级别的安全的随机数，但在某些情况下也可以是生成的确定值 k。

表 139. ECDSA 签名——输出

带方向的参数		值（注）	存储地址	大小
OUT	签名部分 *r*	(0 < *r* < n)	RAM@0x700	EOS
	签名部分 *s*	(0 < *s* < n)	RAM@0x754	
ERROR	签名结果	– 0x0：无错误 – 0x1：签名部分 r 等于 0 – 0x2：签名部分 s 等于 0	RAM@0xEE8	32 位

注： *如果错误输出不为零，则应清除 PKA 存储器的内容，以免泄漏私钥的相关信息。*

扩展 ECDSA 支持

PKA 还支持扩展 ECDSA 签名，对应的输入和输出具有相同的 ECDSA 签名（分别参见*表 138* 和*表 139*），此外还添加了点 kG 的坐标。有关此额外输出的定义，请参见*表 140*。

表 140. 扩展 ECDSA 签名（额外输出）

带方向的参数		值（注）	存储地址	大小
OUT	曲线点 kG 坐标 x_1	($0 \le x_1 < p$)	RAM@0x103C	EOS
	曲线点 kG 坐标 y_1	($0 \le y_1 < p$)	RAM@0x1090	

23.4.17 ECDSA 验证

表 141（输入参数）和*表 142*（输出参数）汇总了 ECDSA 验证运算（在*第 23.3.4 节*进行了概要介绍）。

应用程序应检查输出错误是否等于零，如果不等于零，则不会验证签名。

表 141. ECDSA 验证（输入）

带方向的参数		值（注）	存储地址	大小
IN	模式	0x26	PKA_CR	6 位
	曲线素数阶 n 长度	（以位为单位，非空）	RAM@0x404	32 位
	曲线模数 p 长度	（以位为单位，非空， 8 < 值 < 640）	RAM@0x4B4	
	曲线系数 a 符号	0x0：正 0x1：负	RAM@0x45C	
	曲线系数 |a|	（绝对值，|a| < p）	RAM@0x460	EOS
	曲线模数值 *p*	（奇素数，$0< p < 2^{640}$）	RAM@0x4B8	
	曲线基点 G 坐标 x	($x < p$)	RAM@0x5E8	
	曲线基点 G 坐标 y	($y < p$)	RAM@0x63C	
	公钥曲线点 Q 坐标 x_Q	($x_Q < p$)	RAM@0xF40	
	公钥曲线点 Q 坐标 y_Q	($y_Q < p$)	RAM@0xF94	
	签名部分 *r*	($0 < r < n$)	RAM@0x1098	
	签名部分 *s*	($0 < s < n$)	RAM@0xA44	
	消息 *z* 的散列值	($z < 2^M$)	RAM@0xFE8	
	曲线素数阶 *n*	（素数）	RAM@0xD5C	

表 142. ECDSA 验证（输出）

带方向的参数		值（注）	存储地址	大小
OUT	结果：ECDSA 验证	0x0：有效签名 非 0x0：无效签名	RAM@0x5B0	32 位

23.5 配置和处理时间示例

23.5.1 曲线配置

表 143 给出了供 PKA 用于 ECC 域计算的模数值：

- p 为用作所有点算术的模数的素数
  - *plen* 为 *p* 的长度（位）
- n 为由 G（曲线的基点）生成的群的（通常是素数）阶数。此变量也称为 *q*，请参见 RFC 5639
  - *nlen* 为 *n* 的长度（位）
- *a* 和 *b* 为用于定义椭圆曲线的公式 $y^2 = x^3 + a*x + b \pmod{p}$ 的系数。对于 NIST 曲线，*a* 始终等于 -3
- $G = (x_G, y_G)$ 为基点

上述模数是针对符合最新安全建议的曲线给出的：

- 名为 *nist* 的曲线由 NIST 在联邦信息处理标准出版物 FIPS PUB 186-4 中发布。
- 名为 *brainpool* 的曲线由互联网工程任务组 (IETF) 在征求意见 RFC 5639 中发布。

表 143. ECC 曲线参数

曲线 ID	变量	值
nist P-192	p_{len}, n_{len}	192, 192
	p	0xFFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFFFF FFFFFFFF
	n	0xFFFFFFFF FFFFFFFF FFFFFFFF 99DEF836 146BC9B1 B4D22831
	a	-3
	b	0x64210519 E59C80E7 0FA7E9AB 72243049 FEB8DEEC C146B9B1
	x_G	0x188DA80E B03090F6 7CBF20EB 43A18800 F4FF0AFD 82FF1012
	y_G	0x07192B95 FFC8DA78 631011ED 6B24CDD5 73F977A1 1E794811
nist P-224	p_{len}, n_{len}	224, 224
	p	0xFFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 00000000 00000001
	n	0xFFFFFFFF FFFFFFFF FFFFFFFF FFFF16A2 E0B8F03E 13DD2945 5C5C2A3D
	a	-3
	b	0xB4050A85 0C04B3AB F5413256 5044B0B7 D7BFD8BA 270B3943 2355FFB4
	x_G	0xB70E0CBD 6BB4BF7F 321390B9 4A03C1D3 56C21122 343280D6 115C1D21
	y_G	0xBD376388 B5F723FB 4C22DFE6 CD4375A0 5A074764 44D58199 85007E34

表 143. ECC 曲线参数（续）

曲线 ID	变量	值
brainpool P224r1	p_{len}, n_{len}	224, 224
	p	0xD7C134AA 26436686 2A183025 75D1D787 B09F0757 97DA89F5 7EC8C0FF
	n	0xD7C134AA 26436686 2A183025 75D0FB98 D116BC4B 6DDEBCA3 A5A7939F
	a	0x68A5E62C A9CE6C1C 299803A6 C1530B51 4E182AD8 B0042A59 CAD29F43
	b	0x2580F63C CFE44138 870713B1 A92369E3 3E2135D2 66DBB372 386C400B
	x_G	0x0D9029AD 2C7E5CF4 340823B2 A87DC68C 9E4CE317 4C1E6EFD EE12C07D
	y_G	0x58AA56F7 72C0726F 24C6B89E 4ECDAC24 354B9E99 CAA3F6D3 761402CD
brainpool P224t1	p_{len}, n_{len}	224, 224
	p	0xD7C134AA 26436686 2A183025 75D1D787 B09F0757 97DA89F5 7EC8C0FF
	n	0xD7C134AA 26436686 2A183025 75D0FB98 D116BC4B 6DDEBCA3 A5A7939F
	a	0xD7C134AA 26436686 2A183025 75D1D787 B09F0757 97DA89F5 7EC8C0FC
	b	0x4B337D93 4104CD7B EF271BF6 0CED1ED2 0DA14C08 B3BB64F1 8A60888D
	x_G	0x6AB1E344 CE25FF38 96424E7F FE14762E CB49F892 8AC0C760 29B4D580
	y_G	0x0374E9F5 143E568C D23F3F4D 7C0D4B1E 41C8CC0D 1C6ABD5F 1A46DB4C
nist P-256 ECDSA-256	p_{len}, n_{len}	256, 256
	p	0xFFFFFFFF 00000001 00000000 00000000 00000000 FFFFFFFF FFFFFFFF FFFFFFFF
	n	0xFFFFFFFF 00000000 FFFFFFFF FFFFFFFF BCE6FAAD A7179E84 F3B9CAC2 FC632551
	a	-3
	b	0x5AC635D8 AA3A93E7 B3EBBD55 769886BC 651D06B0 CC53B0F6 3BCE3C3E 27D2604B
	x_G	0x6B17D1F2 E12C4247 F8BCE6E5 63A440F2 77037D81 2DEB33A0 F4A13945 D898C296
	y_G	0x4FE342E2 FE1A7F9B 8EE7EB4A 7C0F9E16 2BCE3357 6B315ECE CBB64068 37BF51F5
brainpool P256r1	p_{len}, n_{len}	256, 256
	p	0xA9FB57DB A1EEA9BC 3E660A90 9D838D72 6E3BF623 D5262028 2013481D 1F6E5377
	n	0xA9FB57DB A1EEA9BC 3E660A90 9D838D71 8C397AA3 B561A6F7 901E0E82 974856A7
	a	0x7D5A0975 FC2C3057 EEF67530 417AFFE7 FB8055C1 26DC5C6C E94A4B44 F330B5D9
	b	0x26DC5C6C E94A4B44 F330B5D9 BBD77CBF 95841629 5CF7E1CE 6BCCDC18 FF8C07B6
	x_G	0x8BD2AEB9 CB7E57CB 2C4B482F FC81B7AF B9DE27E1 E3BD23C2 3A4453BD 9ACE3262
	y_G	0x547EF835 C3DAC4FD 97F8461A 14611DC9 C2774513 2DED8E54 5C1D54C7 2F046997

表 143. ECC 曲线参数（续）

曲线 ID	变量	值
brainpool P256t1	p_{len}, n_{len}	256, 256
	p	0xA9FB57DB A1EEA9BC 3E660A90 9D838D72 6E3BF623 D5262028 2013481D 1F6E5377
	n	0xA9FB57DB A1EEA9BC 3E660A90 9D838D71 8C397AA3 B561A6F7 901E0E82 974856A7
	a	0xA9FB57DB A1EEA9BC 3E660A90 9D838D72 6E3BF623 D5262028 2013481D 1F6E5374
	b	0x662C61C4 30D84EA4 FE66A773 3D0B76B7 BF93EBC4 AF2F4925 6AE58101 FEE92B04
	x_G	0xA3E8EB3C C1CFE7B7 732213B2 3A656149 AFA142C4 7AAFBC2B 79A19156 2E1305F4
	y_G	0x2D996C82 3439C56D 7F7B22E1 4644417E 69BCB6DE 39D02700 1DABE8F3 5B25C9BE
brainpool P320r1	p_{len}, n_{len}	320, 320
	p	0xD35E4720 36BC4FB7 E13C785E D201E065 F98FCFA6 F6F40DEF 4F92B9EC 7893EC28 FCD412B1 F1B32E27
	n	0xD35E4720 36BC4FB7 E13C785E D201E065 F98FCFA5 B68F12A3 2D482EC7 EE8658E9 8691555B 44C59311
	a	0x3EE30B56 8FBAB0F8 83CCEBD4 6D3F3BB8 A2A73513 F5EB79DA 66190EB0 85FFA9F4 92F375A9 7D860EB4
	b	0x52088394 9DFDBC42 D3AD1986 40688A6F E13F4134 9554B49A CC31DCCD 88453981 6F5EB4AC 8FB1F1A6
	x_G	0x43BD7E9A FB53D8B8 5289BCC4 8EE5BFE6 F20137D1 0A087EB6 E7871E2A 10A599C7 10AF8D0D 39E20611
	y_G	0x14FDD055 45EC1CC8 AB409324 7F77275E 0743FFED 117182EA A9C77877 AAAC6AC7 D35245D1 692E8EE1
brainpool P320t1	p_{len}, n_{len}	320, 320
	p	0xD35E4720 36BC4FB7 E13C785E D201E065 F98FCFA6 F6F40DEF 4F92B9EC 7893EC28 FCD412B1 F1B32E27
	n	0xD35E4720 36BC4FB7 E13C785E D201E065 F98FCFA5 B68F12A3 2D482EC7 EE8658E9 8691555B 44C59311
	a	0xA7F561E0 38EB1ED5 60B3D147 DB782013 064C19F2 7ED27C67 80AAF77F B8A547CE B5B4FEF4 22340353
	b	0xA7F561E0 38EB1ED5 60B3D147 DB782013 064C19F2 7ED27C67 80AAF77F B8A547CE B5B4FEF4 22340353
	x_G	0x925BE9FB 01AFC6FB 4D3E7D49 90010F81 3408AB10 6C4F09CB 7EE07868 CC136FFF 3357F624 A21BED52
	y_G	0x63BA3A7A 27483EBF 6671DBEF 7ABB30EB EE084E58 A0B077AD 42A5A098 9D1EE71B 1B9BC045 5FB0D2C3

表 143. ECC 曲线参数（续）

曲线 ID	变量	值
nist P-384 ECDSA-384	p_{len}, n_{len}	384, 384
	p	0xFFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFFFF 00000000 00000000 FFFFFFFF
	n	0xFFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF C7634D81 F4372DDF 581A0DB2 48B0A77A ECEC196A CCC52973
	a	-3
	b	0xB3312FA7 E23EE7E4 988E056B E3F82D19 181D9C6E FE814112 0314088F 5013875A C656398D 8A2ED19D 2A85C8ED D3EC2AEF
	x_G	0xAA87CA22 BE8B0537 8EB1C71E F320AD74 6E1D3B62 8BA79B98 59F741E0 82542A38 5502F25D BF55296C 3A545E38 72760AB7
	y_G	0x3617DE4A 96262C6F 5D9E98BF 9292DC29 F8F41DBD 289A147C E9DA3113 B5F0B8C0 0A60B1CE 1D7E819D 7A431D7C 90EA0E5F
brainpool P384r1	p_{len}, n_{len}	384, 384
	p	0x8CB91E82 A3386D28 0F5D6F7E 50E641DF 152F7109 ED5456B4 12B1DA19 7FB71123 ACD3A729 901D1A71 87470013 3107EC53
	n	0x8CB91E82 A3386D28 0F5D6F7E 50E641DF 152F7109 ED5456B3 1F166E6C AC0425A7 CF3AB6AF 6B7FC310 3B883202 E9046565
	a	0x7BC382C6 3D8C150C 3C72080A CE05AFA0 C2BEA28E 4FB22787 139165EF BA91F90F 8AA5814A 503AD4EB 04A8C7DD 22CE2826
	b	0x04A8C7DD 22CE2826 8B39B554 16F0447C 2FB77DE1 07DCD2A6 2E880EA5 3EEB62D5 7CB43902 95DBC994 3AB78696 FA504C11
	x_G	0x1D1C64F0 68CF45FF A2A63A81 B7C13F6B 8847A3E7 7EF14FE3 DB7FCAFE 0CBD10E8 E826E034 36D646AA EF87B2E2 47D4AF1E
	y_G	0x8ABE1D75 20F9C2A4 5CB1EB8E 95CFD552 62B70B29 FEEC5864 E19C054F F9912928 0E464621 77918111 42820341 263C5315
brainpool P384t1	p_{len}, n_{len}	384, 384
	p	0x8CB91E82 A3386D28 0F5D6F7E 50E641DF 152F7109 ED5456B4 12B1DA19 7FB71123 ACD3A729 901D1A71 87470013 3107EC53
	n	0x8CB91E82 A3386D28 0F5D6F7E 50E641DF 152F7109 ED5456B3 1F166E6C AC0425A7 CF3AB6AF 6B7FC310 3B883202 E9046565
	a	0x8CB91E82 A3386D28 0F5D6F7E 50E641DF 152F7109 ED5456B4 12B1DA19 7FB71123 ACD3A729 901D1A71 87470013 3107EC50
	b	0x7F519EAD A7BDA81B D826DBA6 47910F8C 4B9346ED 8CCDC64E 4B1ABD11 756DCE1D 2074AA26 3B88805C ED70355A 33B471EE
	x_G	0x18DE98B0 2DB9A306 F2AFCD72 35F72A81 9B80AB12 EBD65317 2476FECD 462AABFF C4FF191B 946A5F54 D8D0AA2F 418808CC
	y_G	0x25AB0569 62D30651 A114AFD2 755AD336 747F9347 5B7A1FCA 3B88F2B6 A208CCFE 46940858 4DC2B291 2675BF5B 9E582928

表 143. ECC 曲线参数（续）

曲线 ID	变量	值
brainpool P512r1	p_{len}, n_{len}	512, 512
	p	0xAADD9DB8 DBE9C48B 3FD4E6AE 33C9FC07 CB308DB3 B3C9D20E D6639CCA 70330871 7D4D9B00 9BC66842 AECDA12A E6A380E6 2881FF2F 2D82C685 28AA6056 583A48F3
	n	0xAADD9DB8 DBE9C48B 3FD4E6AE 33C9FC07 CB308DB3 B3C9D20E D6639CCA 70330870 553E5C41 4CA92619 41866119 7FAC1047 1DB1D381 085DDADD B5879682 9CA90069
	a	0x7830A331 8B603B89 E2327145 AC234CC5 94CBDD8D 3DF91610 A83441CA EA9863BC 2DED5D5A A8253AA1 0A2EF1C9 8B9AC8B5 7F1117A7 2BF2C7B9 E7C1AC4D 77FC94CA
	b	0x3DF91610 A83441CA EA9863BC 2DED5D5A A8253AA1 0A2EF1C9 8B9AC8B5 7F1117A7 2BF2C7B9 E7C1AC4D 77FC94CA DC083E67 984050B7 5EBAE5DD 2809BD63 8016F723
	x_G	0x81AEE4BD D82ED964 5A21322E 9C4C6A93 85ED9F70 B5D916C1 B43B62EE F4D0098E FF3B1F78 E2D0D48D 50D1687B 93B97D5F 7C6D5047 406A5E68 8B352209 BCB9F822
	y_G	0x7DDE385D 566332EC C0EABFA9 CF7822FD F209F700 24A57B1A A000C55B 881F8111 B2DCDE49 4A5F485E 5BCA4BD8 8A2763AE D1CA2B2F A8F05406 78CD1E0F 3AD80892
brainpool P512t1	p_{len}, n_{len}	512, 512
	p	0xAADD9DB8 DBE9C48B 3FD4E6AE 33C9FC07 CB308DB3 B3C9D20E D6639CCA 70330871 7D4D9B00 9BC66842 AECDA12A
	n	0xAADD9DB8 DBE9C48B 3FD4E6AE 33C9FC07 CB308DB3 B3C9D20E D6639CCA 70330870 553E5C41 4CA92619 41866119 7FAC1047 1DB1D381 085DDADD B5879682 9CA90069
	a	0xAADD9DB8DBE9C48B3FD4E6AE33C9FC07CB308DB3B3C9D20ED6639CCA703308717D4D9B009BC66842AECDA12AE6A380E62881FF2F2D82C68528AA6056583A48F0
	b	0x7CBBBCF9441CFAB76E1890E46884EAE321F70C0BCB4981527897504BEC3E36A62BCDFA2304976540F6450085F2DAE145C22553B465763689180EA257 1867423E
	x_G	0x640ECE5C12788717B9C1BA06CBC2A6FEBA85842458C56DDE9DB1758D39C0313D82BA51735CDB3EA499AA77A7D6943A64F7A3F25FE26F06B51BAA2696FA9035DA
	y_G	0x5B534BD595F5AF0FA2C892376C84ACE1BB4E3019B71634C01131159CAE03CEE9D9932184BEEF216BD71DF2DADF86A627306ECFF96DBB8BACE198B61E00F8B332

表 143. ECC 曲线参数（续）

曲线 ID	变量	值
nist P-521 ECDSA-521	p_{len}, n_{len}	521, 521
	p (2^{521}-1)	0x01FFFFFF FFFF
	n	0x01FFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFA5186 8783BF2F 966B7FCC 0148F709 A5D03BB5 C9B8899C 47AEBB6F B71E9138 6409
	a	-3
	b	0x051 953EB961 8E1C9A1F 929A21A0 B68540EE A2DA725B 99B315F3 B8B48991 8EF109E1 56193951 EC7E937B 1652C0BD 3BB1BF07 3573DF88 3D2C34F1 EF451FD4 6B503F00
	x_G	0x0C6 858E06B7 0404E9CD 9E3ECB66 2395B442 9C648139 053FB521 F828AF60 6B4D3DBA A14B5E77 EFE75928 FE1DC127 A2FFA8DE 3348B3C1 856A429B F97E7E31 C2E5BD66
	y_G	0x118 39296A78 9A3BC004 5C8A5FB4 2C7D1BD9 98F54449 579B4468 17AFBD17 273E662C 97EE7299 5EF42640 C550B901 3FAD0761 353C7086 A272C240 88BE9476 9FD16650

23.5.2 计算时间

下表汇总了 PKA 计算时间（以时钟周期表示）。

表 144. 模幂

指数长度（位）	模式	操作数长度（位）		
		1024	2048	3072
3	正常	152000	407000	864000
	快速	23000	82000	178000
17	正常	163000	448000	955000
	快速	34000	123000	267000
$2^{16}+1$	正常	208000	611000	1308000
	快速	79000	286000	622000
1024	正常	5832000	-	-
	快速	5640000	-	-
2048	正常	-	41917000	-
	快速	-	41023000	-
3072	正常	-	-	137477000
	快速	-	-	136761000

表 145. ECC 标量乘法(1)

模式	模数长度（位）						
	160	192	256	320	384	512	521
正常	817000	1250000	2462000	4254000	6821000	14445000	16580000
快速	815000	1247000	2458000	4247000	6807000	14421000	16579000

1. 这些时间取决于标量参数中包含的“1”的数目。

表 146. ECDSA 签名平均计算时间(1)(2)

模数长度（位）						
160	192	256	320	384	512	521
880000	1332000	2645000	4508000	7298000	15309000	17770000

1. 这些值是给定长度的随机模数的平均执行时间，因为它们取决于模数的长度和值。
2. 定义 NIST 椭圆曲线有限域的模数所需的执行时间比用于 Brainpool 椭圆曲线的模数或相同大小的随机模数所需的执行时间短。

表 147. ECDSA 验证平均计算时间

模数长度（位）						
160	**192**	**256**	**320**	**384**	**512**	**521**
1750000	2675000	5249000	9063000	14559000	30673000	35794000

表 148. Montgomery 参数平均计算时间(1)

操作数长度（位）									
160	**192**	**256**	**320**	**384**	**512**	**521**	**1024**	**2048**	**3072**
2259	3923	5924	7451	10841	17506	32000	59768	233073	552321

1. 计算时间取决于模数的长度和值，因此这些值为给定长度的随机模数的平均执行时间。

23.6 PKA 中断

公钥加速器可生成三个独立的可屏蔽中断源，用以通知发生以下事件：

1. 访问未映射的地址 (ADDRERRF)，请参见*第 23.3.6 节*
2. 在执行 PKA 运算时访问 PKA RAM (RAMERRF)，请参见*第 23.3.6 节*
3. PKA 运算结束 (PROCENDF)

这三个中断源连接到同一全局中断请求信号 pka_it。

用户可通过更改 *PKA 控制寄存器 (PKA_CR)* 中的屏蔽位，单独使能或禁止上述中断源。将相应的屏蔽位置 1 可使能中断。各个中断事件的状态可从 PKA 状态寄存器 (PKA_SR) 中读取，并在 PKA_CLRFR 寄存器中清除。

表 149 对可用功能进行了总结。

表 149. PKA 中断请求

中断事件	事件标志	使能控制位
访问未映射的地址错误	ADDRERRF	ADDRERRIE
PKA RAM 访问错误	RAMERRF	RAMERRIE
PKA 运算结束	PROCENDF	PROCENDIE

23.7 PKA 寄存器

23.7.1 PKA 控制寄存器 (PKA_CR)

PKA control register

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRERRIE	RAMERRIE	Res.	PROCENDIE	Res.
											rw	rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	MODE[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	START	EN
		rw	rw	rw	rw	rw	rw							rw	rw

位 31:21 保留，必须保持复位值。

位 20 **ADDRERRIE**：地址错误中断使能 (Address error interrupt enable)
0：PKA_SR 中的 ADDRERRF 标志置 1 时，不生成中断。
1：PKA_SR 中的 ADDRERRF 标志置 1 时，生成中断。

位 19 **RAMERRIE**：RAM 错误中断使能 (RAM error interrupt enable)
0：PKA_SR 中的 RAMERRF 标志置 1 时，不生成中断。
1：PKA_SR 中的 RAMERRF 标志置 1 时，生成中断。

位 18 保留，必须保持复位值。

位 17 **PROCENDIE**：运算结束中断使能 (End of operation interrupt enable)
0：PKA_SR 中的 PROCENDF 标志置 1 时，不生成中断。
1：PKA_SR 中的 PROCENDF 标志置 1 时，生成中断。

位 16:14 保留，必须保持复位值。

位 13:8 **MODE[5:0]**：PKA 运算模式 (PKA operation code)
000000：先进行 Montgomery 参数计算，然后进行模幂运算
000001：仅进行 Montgomery 参数计算
000010：仅进行模幂运算（必须先载入 Montgomery 参数）
100000：先进行 Montgomery 参数计算，然后进行 ECC 标量乘法
100010：仅进行 ECC 标量乘法（必须先载入 Montgomery 参数）
100100：ECDSA 签名
100110：ECDSA 验证
101000：检查点是否在椭圆曲线 Fp 上
000111：RSA CRT 求幂
001000：模逆
001001：算术加法
001010：算术减法
001011：算术乘法
001100：算术比较
001101：模约简
001110：模加
001111：模减
010000：Montgomery 乘法
其他值：保留

位 7:2 保留，必须保持复位值。

位 1 **START**：开始运算 (start the operation)
向此位写入 1 后，将使用已写入到 PKA RAM 的操作数和数据开始进行 MODE[5:0] 选择的运算。此位始终读为 0。

注：　如果 PKA 忙，则忽略 START。

位 0 **EN**：PKA 使能 (PKA enable)
0：禁止 PKA
1：使能 PKA

注：　EN=0 时，应用程序仍可访问 PKA RAM。

23.7.2 PKA 状态寄存器 (PKA_SR)

PKA status register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDR ERRF	RAM ERRF	Res.	PROC ENDF	BUSY
											r	r		r	r

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

位 31:21 保留，必须保持复位值。

位 20 **ADDRERRF**：地址错误标志 (Address error flag)

0：无地址错误

1：地址访问超出范围（未映射的地址）

位 19 **RAMERRF**：PKA RAM 错误标志 (PKA RAM error flag)

0：无 PKA RAM 访问错误

1：当 PKA 内核正在计算和使用其内部 RAM 时发生了对 PKA RAM 的 AHB 访问（在进行 PKA 运算时不允许 AHB PKA_RAM 访问）。

位 18 保留，必须保持复位值。

位 17 **PROCENDF**：PKA 运算结束标志 (PKA End of Operation flag)

0：正在进行运算

1：PKA 运算已完成。当 BUSY 位清零时，该标志置 1。

位 16 **BUSY**：PKA 运算正在进行中 (PKA operation is in progress)

PKA_CR 中的 START 位置 1 时，此位会置 1。此位在计算完成后会自动清零，这意味着可以安全访问 PKA RAM 并开始执行新的运算。

0：当前未进行运算（默认值）

1：正在进行运算

如果 PKA 以错误的操作码开始，则 IP 将忙几个周期，然后将自动中止运算并返回到就绪状态（BUSY 位置 0）。

位 15:0 保留，必须保持复位值。

23.7.3 PKA 清零标志寄存器 (PKA_CLRFR)

PKA clear flag register

偏移地址：0x08

复位值：0x0000 0000

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

位 31:21 保留，必须保持复位值。

位 20 **ADDRERRFC**：清零地址错误标志 (Clear Address error flag)

0：不执行任何操作

1：将 ADDRERRF 标志清零

位 19 **RAMERRFC**：清零 PKA RAM 错误标志 (Clear PKA RAM error flag)

0：不执行任何操作

1：将 RAMERRF 标志清零

位 18 保留，必须保持复位值。

位 17 **PROCENDFC**：清零 PKA 运算结束标志 (Clear PKA End of Operation flag)
0：不执行任何操作
1：将 PKA_SR 中的 PROCENDF 标志清零

位 16:0 保留，必须保持复位值。

注： *读取 PKA_CLRFR 将返回全零。*

23.7.4 PKA RAM

PKA RAM 映射到偏移地址 0x0400（相对于 PKA 基址）。仅支持 32 位字单次访问（通过 PKA.AHB 接口）。

RAM 大小为 3576 字节（最大字偏移：0x11F4）。

23.7.5 PKA 寄存器映射和复位值

表 150. PKA 寄存器映射和复位值

偏移	寄存器名称 复位值	寄存器大小																													
		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
0x000	PKA_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRERRIE	RAMERRIE	Res.	PROCENDIE	Res.	Res.	Res.	MODE[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	START	EN
	Reset value												0	0		0				0	0	0	0	0	0							0	0
0x004	PKA_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRERRF	RAMERRF	Res.	PROCENDF	BUSY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value												0	0		0	0																
0x008	PKA_CLRFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRERRFC	RAMERRFC	Res.	PROCENDFC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value												0	0		0																

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

24 高级控制定时器 (TIM1)

在本章中，应将“TIMx”理解为“TIM1”，因为本参考手册所适用的产品中只有一个此类定时器的实例。

24.1 TIM1 简介

高级控制定时器 (TIM1) 包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器可用于多种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、PWM 和带死区插入的互补 PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

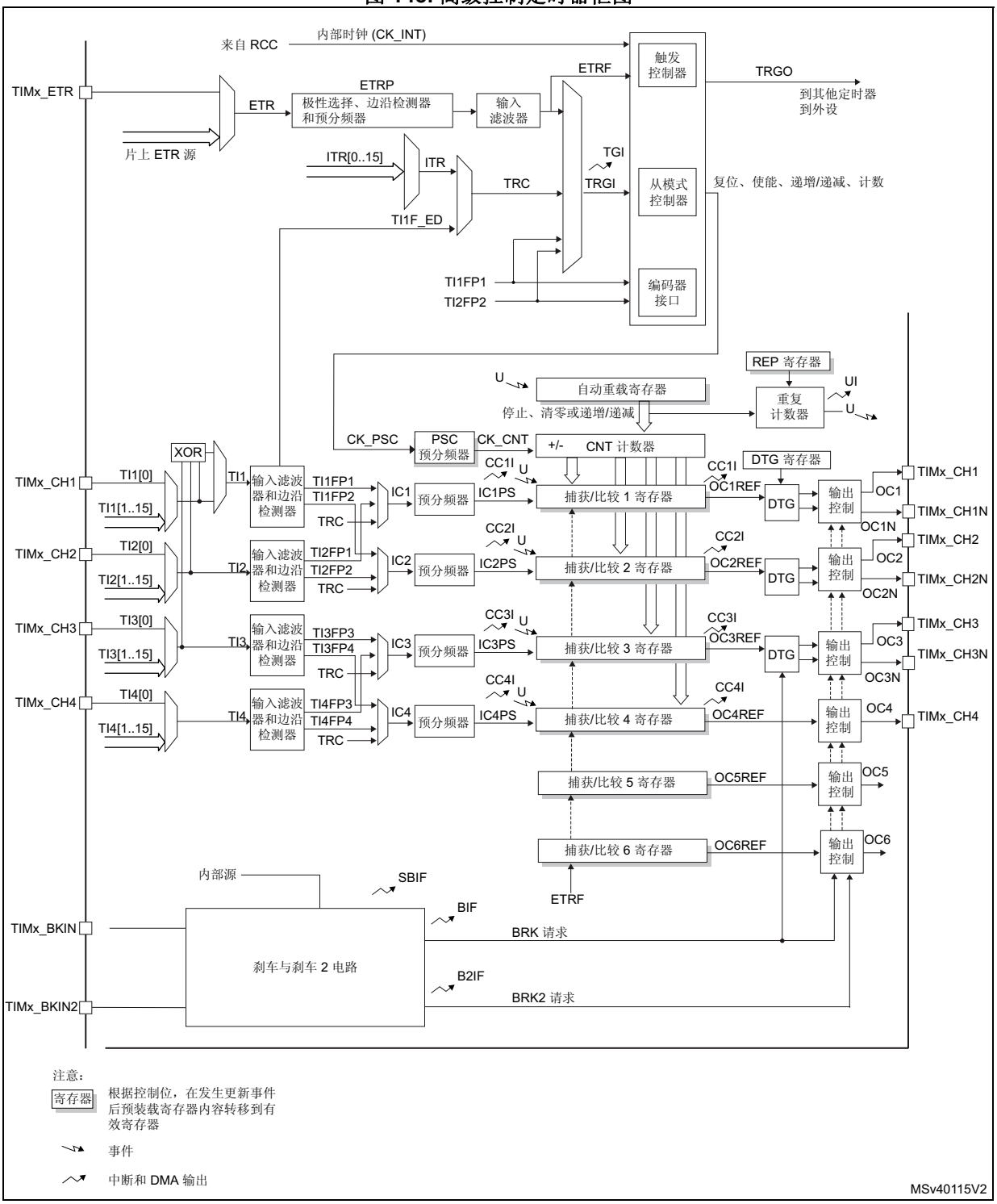
高级控制定时器 (TIM1) 和通用定时器 (TIMy) 彼此完全独立，不共享任何资源。
如*第 24.3.26 节：定时器同步*中所述，它们可以一起同步操作。

24.2 TIM1 主要特性

TIM1 定时器主要特性：

- 16 位递增、递减、递增/递减自动重载计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 到 65536 之间。
- 多达 6 个独立通道，可用于：
  - 输入捕获（但通道 5 和通道 6 除外）
  - 输出比较
  - PWM 生成（边沿和中心对齐模式）
  - 单脉冲模式输出
- 带可编程死区的互补输出。
- 同步电路用来实现外部信号对定时器的控制以及多个定制器互联。
- 重复计数器，用于仅在给定数目的计数器周期后更新定时器寄存器。
- 2 个刹车输入，用于将定时器的输出信号置于用户可选的安全配置中。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较
- 支持定位用增量（正交）编码器和霍尔传感器电路。
- 触发输入用作外部时钟或逐周期电流管理。

图 148. 高级控制定时器框图



1. 内部刹车事件源可以是：
   - 由 CSS 生成的时钟故障事件。有关 CSS 的更多信息，请参见*第 8.2.11 节：HSE 时钟安全系统 (CSS)*
   - PVD 输出
   - SRAM 奇偶校验错误信号
   - CPU1 Cortex®-M4LOCKUP (Hardfault) 输出
   - COMPx 输出，x = 1、2

24.3 TIM1 功能描述

24.3.1 时基单元

可编程高级控制定时器的主要模块是一个 16 位计数器及其相关的自动重载寄存器。计数器可递增计数、递减计数或交替进行递增和递减计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)
- 重复计数器寄存器 (TIMx_RCR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 TIMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器说明

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 TIMx_PSC 寄存器中的 16 位寄存器所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 149 和*图 150* 以一些示例说明在预分频比实时变化时计数器的行为：

图 149. 预分频器分频由 1 变为 2 时的计数器时序图

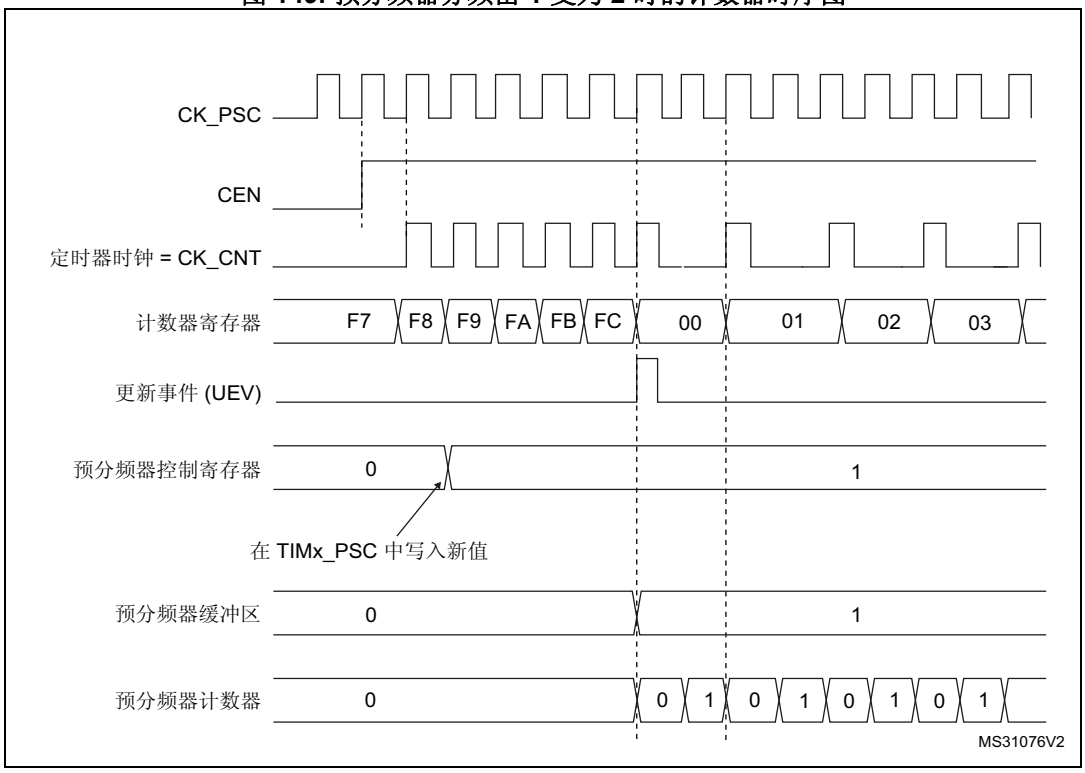
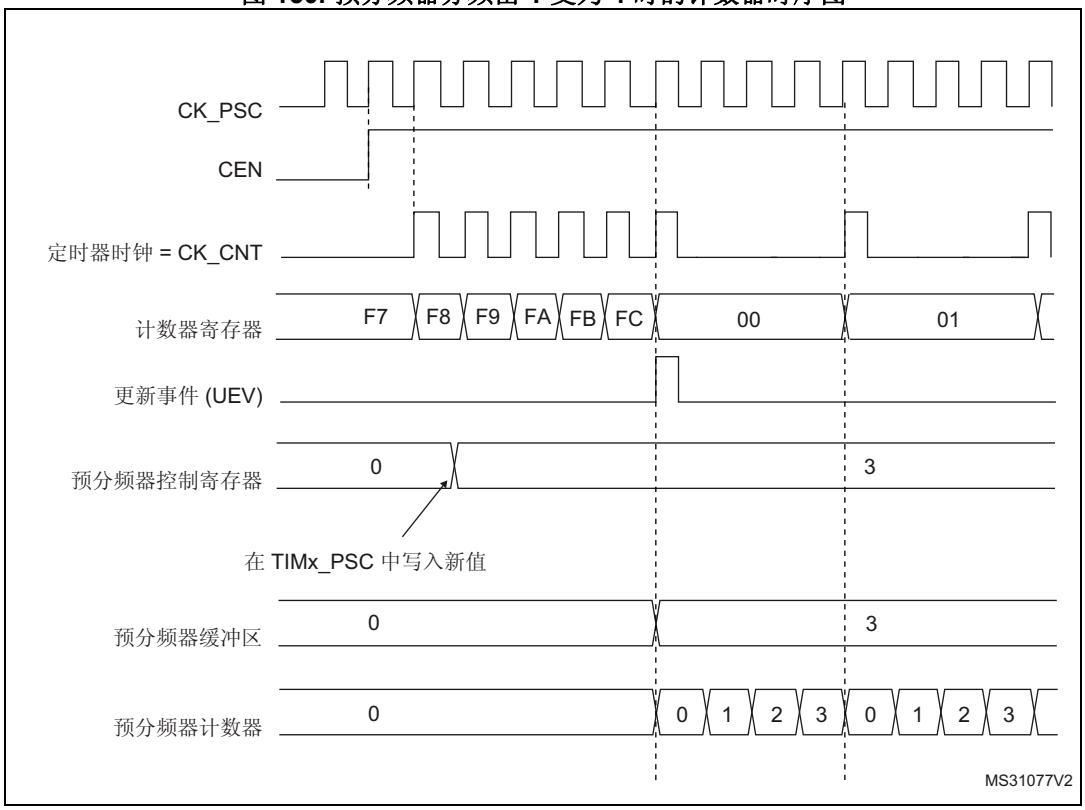


图 150. 预分频器分频由 1 变为 4 时的计数器时序图



24.3.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中设定的次数加一次（(TIMx_RCR) + 1）后，将生成更新事件 (UEV)。否则，将在每次计数器上溢时产生更新事件。

将 TIMx_EGR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容。
- 自动重载影子寄存器将以预装载值 (TIMx_ARR) 进行更新。
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 151. 计数器时序图，1 分频内部时钟

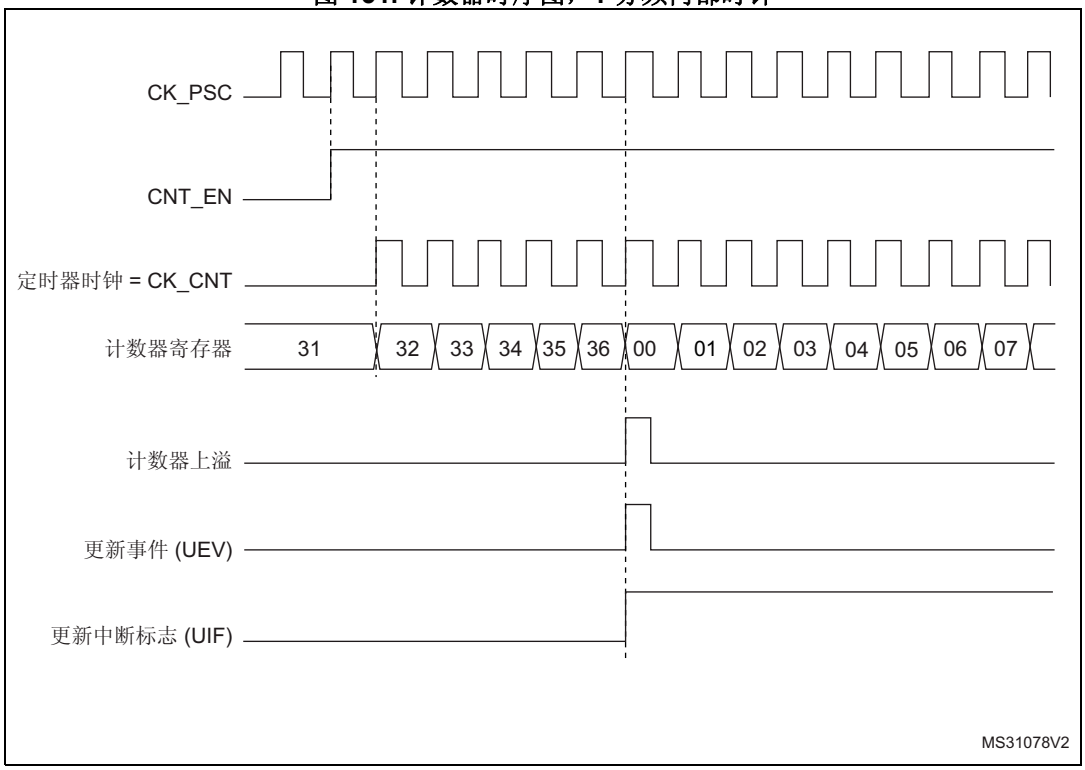


图 152. 计数器时序图，2 分频内部时钟

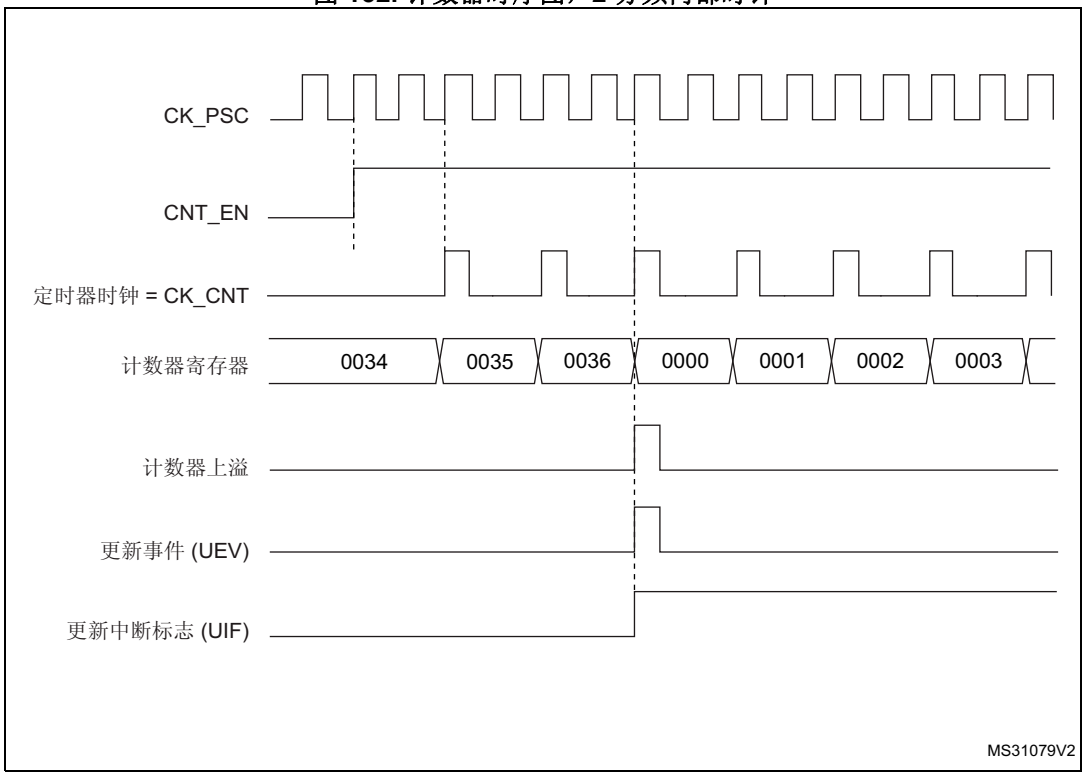


图 153. 计数器时序图，4 分频内部时钟

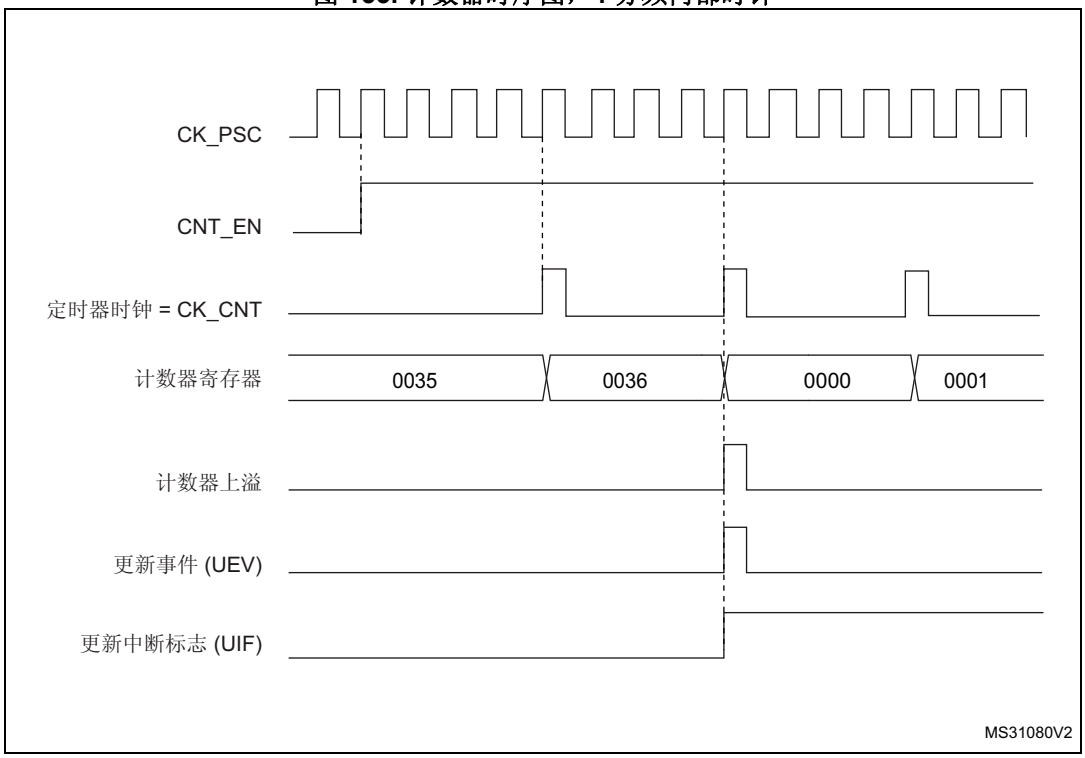


图 154. 计数器时序图，N 分频内部时钟

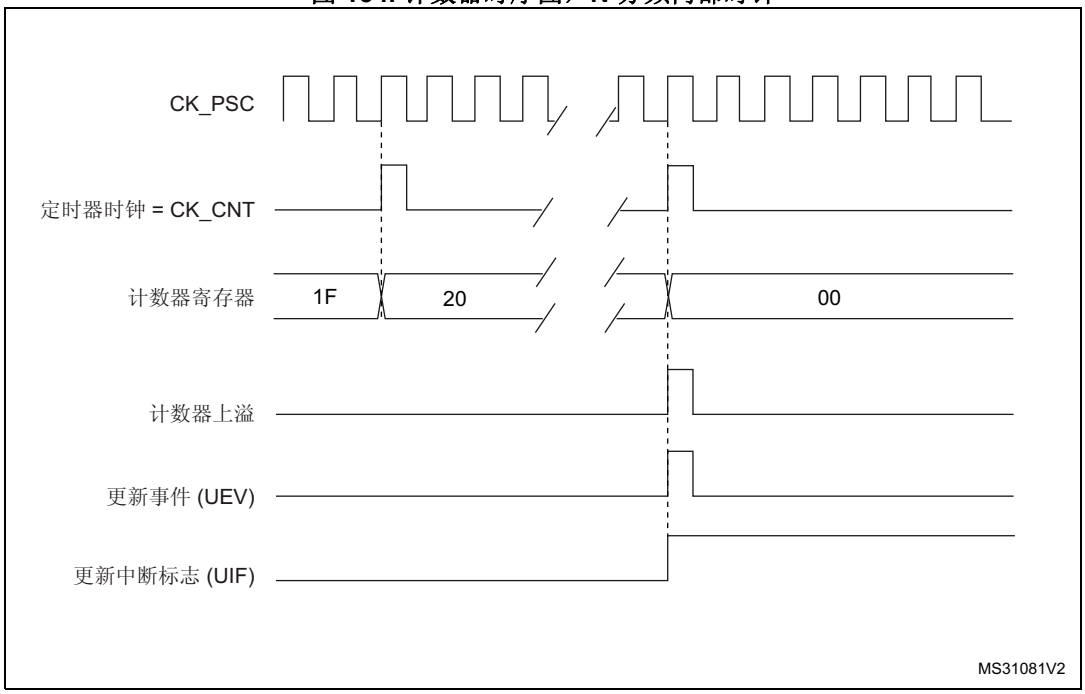


图 155. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

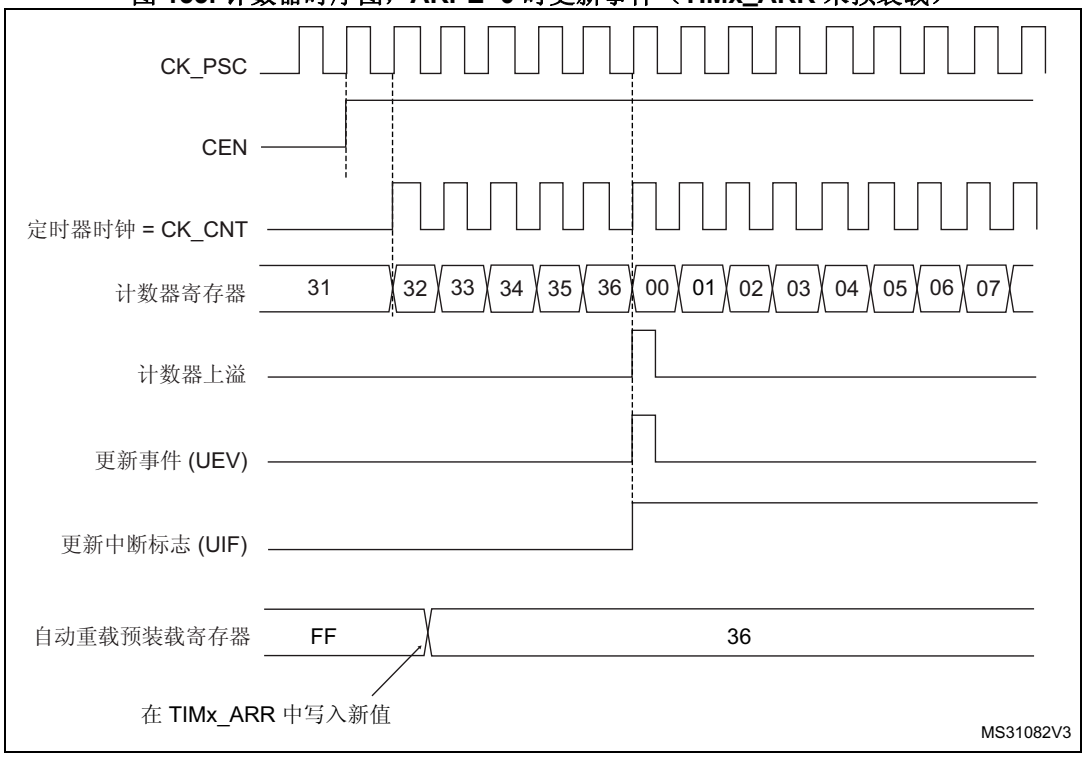
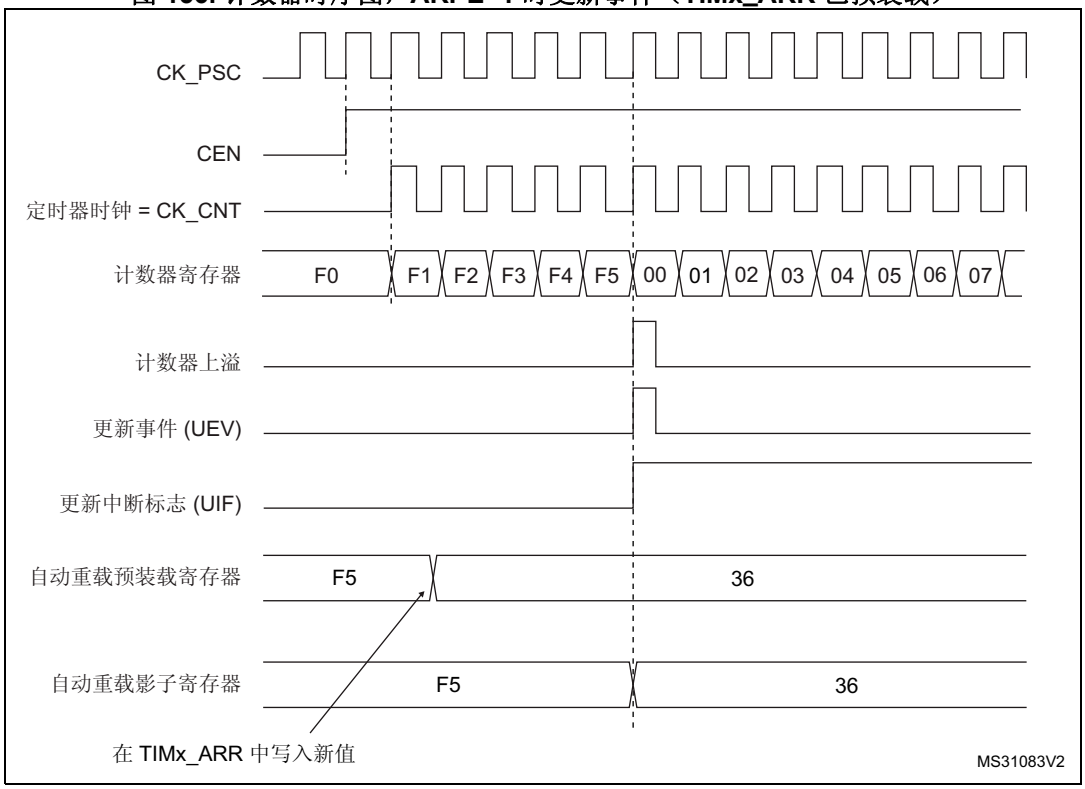


图 156. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



递减计数模式

在递减计数模式下，计数器从自动重载值（TIMx_ARR 寄存器的内容）开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

如果使用重复计数器，则当递减计数的重复次数达到重复计数器寄存器中编程的次数加一次（(TIMx_RCR) + 1）后，将产生更新事件 (UEV)。否则，将在每次计数器下溢时产生更新事件。

将 TIMx_EGR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分频比保持不变）。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容。
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载有效寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，ARR 寄存器更新在计数器重载之前被更新，因此下一个周期就是预期的值。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 157. 计数器时序图，1 分频内部时钟

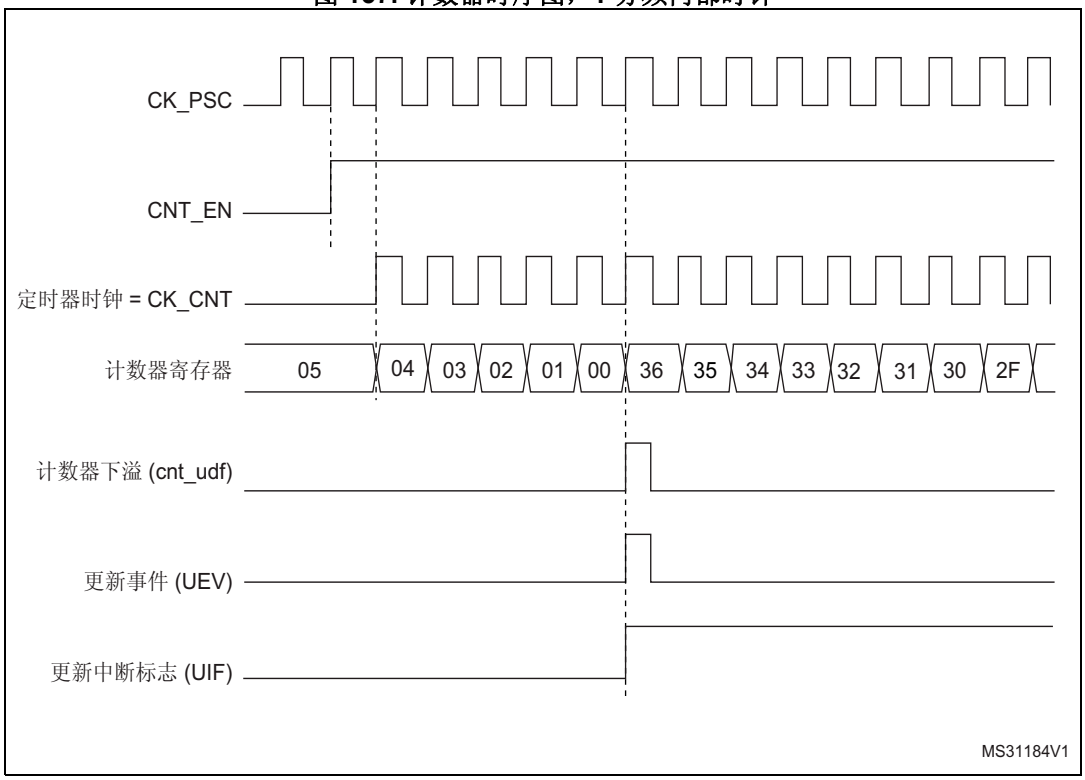


图 158. 计数器时序图，2 分频内部时钟

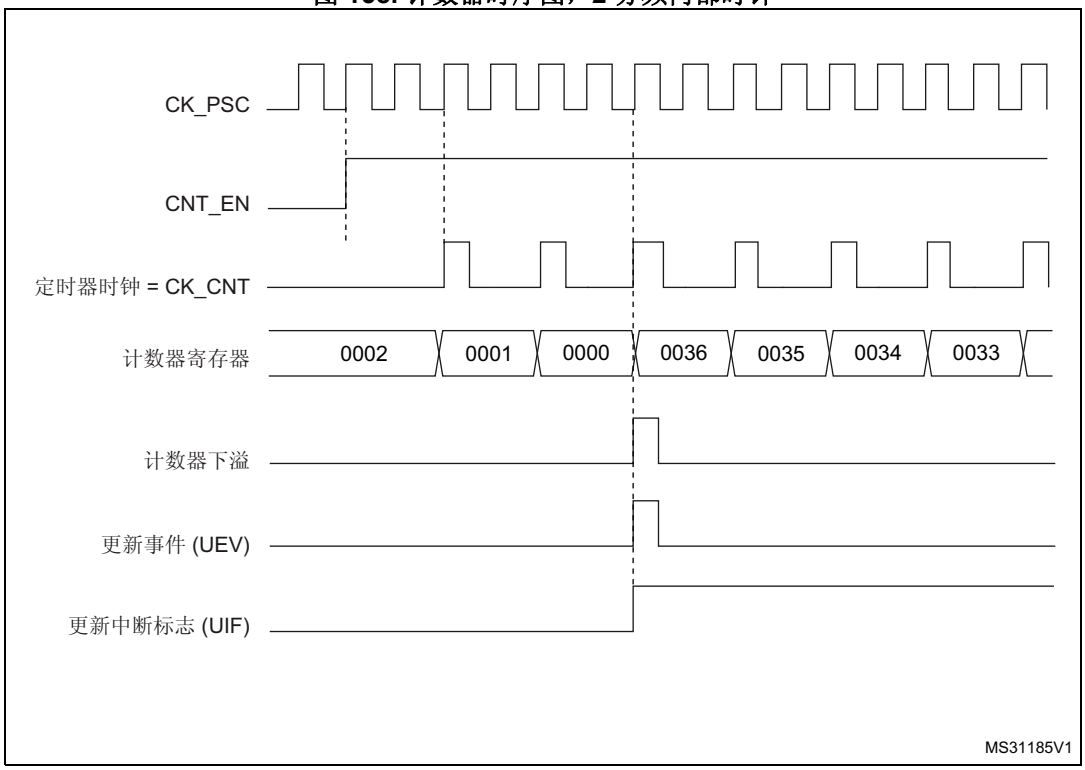


图 159. 计数器时序图，4 分频内部时钟

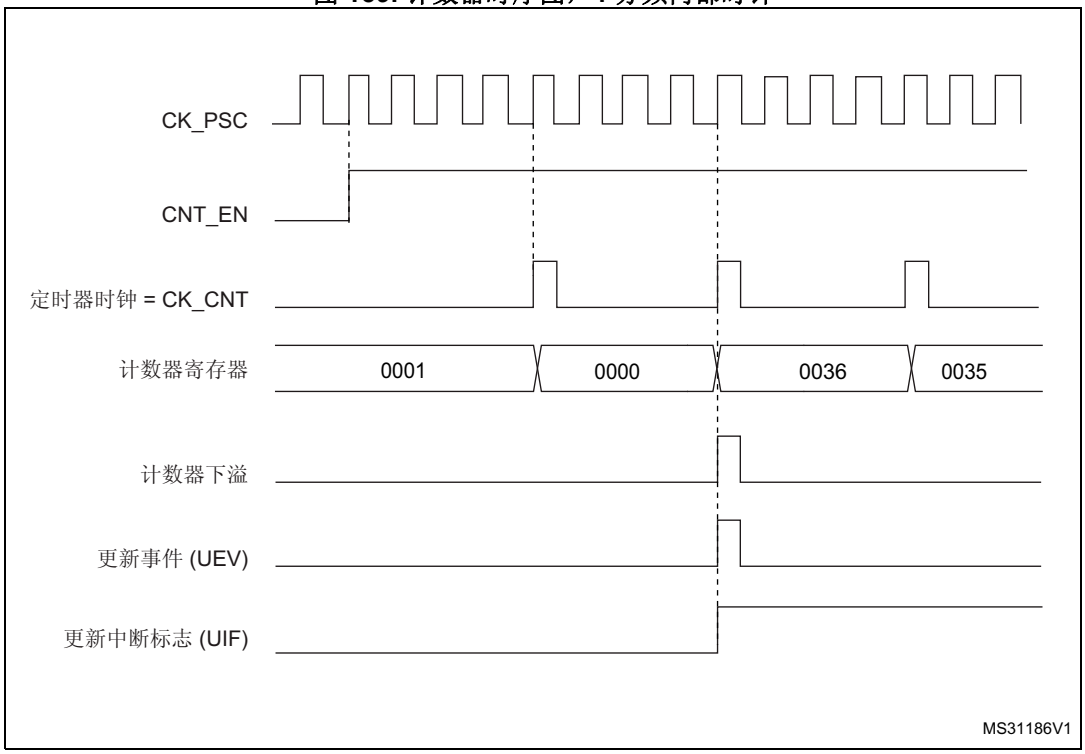


图 160. 计数器时序图，N 分频内部时钟

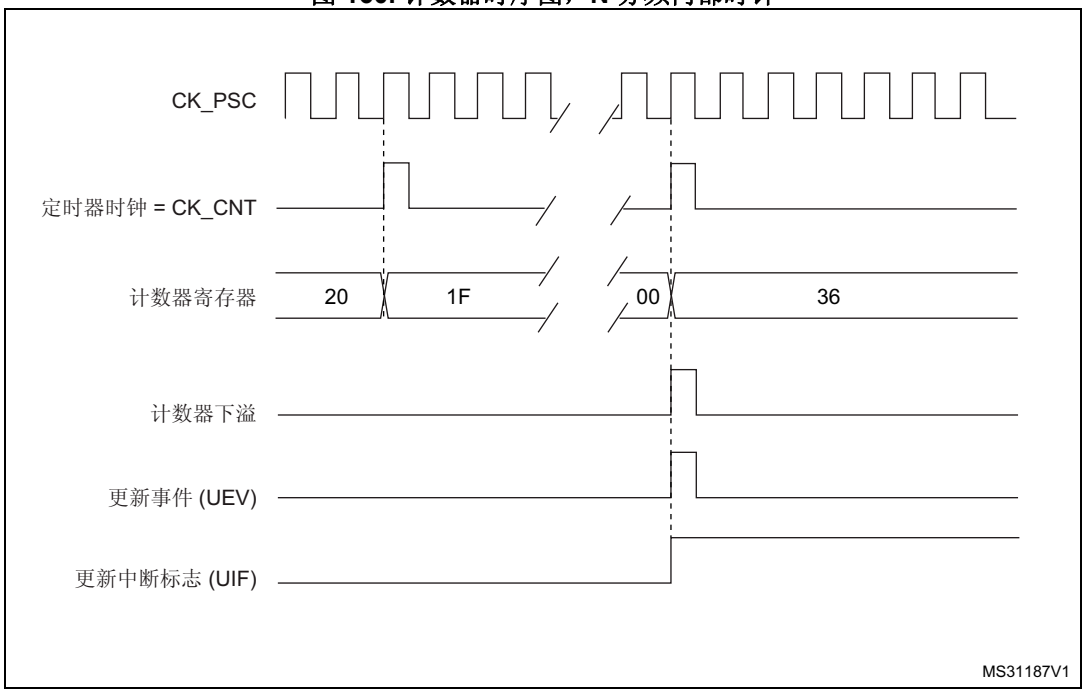
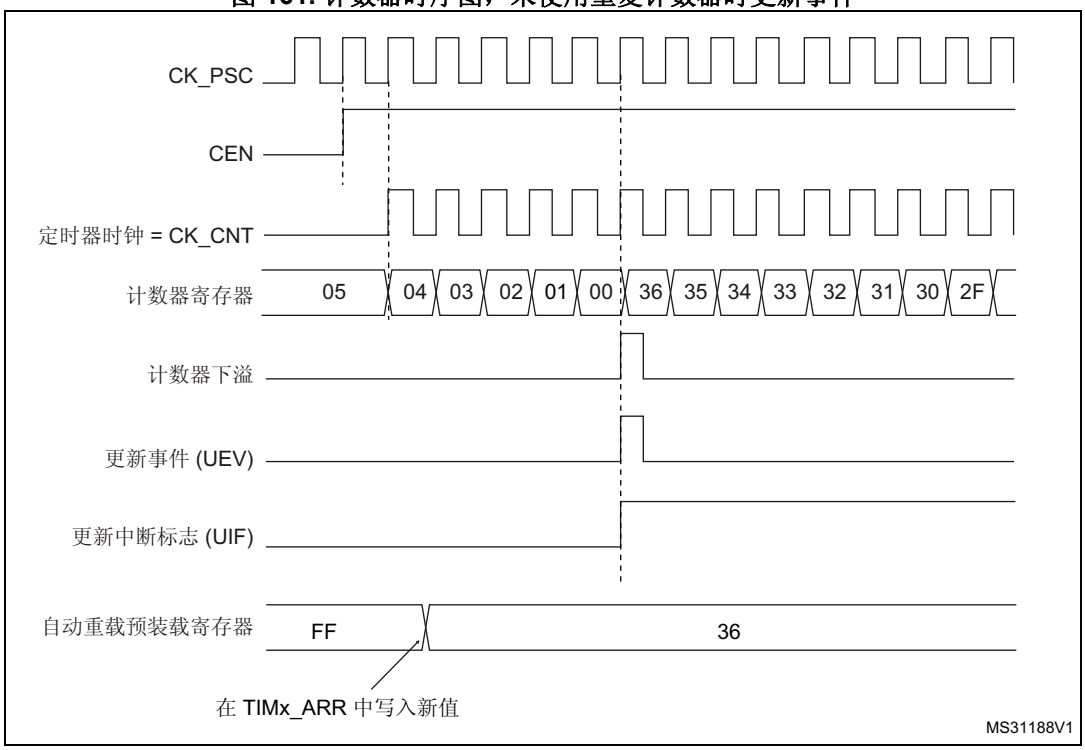


图 161. 计数器时序图，未使用重复计数器时更新事件



中心对齐模式（递增/递减计数）

在中心对齐模式下，计数器从 0 开始计数到自动重载值（TIMx_ARR 寄存器的内容）– 1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中心对齐模式 1，CMS =“01”）、计数器递增计数（中心对齐模式 2，CMS =“10”）以及计数器递增/递减计数（中心对齐模式 3，CMS =“11”）。

在此模式下，TIMx_CR1 寄存器的 DIR 方向位不可写入值，而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

UEV 更新事件可通过软件禁止，只需将 TIMx_CR1 寄存器中的 UDIS 位置 1。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

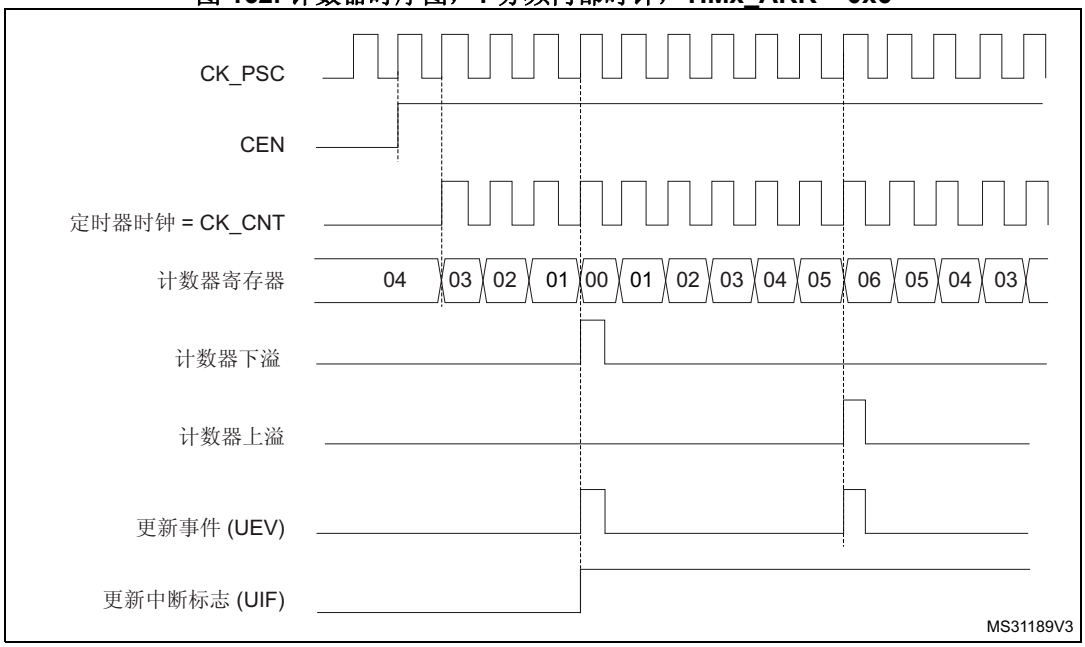
此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会产生 UEV 更新事件，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容。
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载有效寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，如果更新操作是由计数器上溢触发的，则 ARR 寄存器在计数器重载之前更新，因此，下一个计数周期就是我们所希望的新的周期长度（计数器被重载新的值）。

以下各图以一些示例说明不同时钟频率下计数器的行为。

图 162. 计数器时序图，1 分频内部时钟，TIMx_ARR = 0x6



1. 此处使用中心对齐模式 1（有关详细信息，请参见*第 24.4 节：TIM1 寄存器*）。

图 163. 计数器时序图，2 分频内部时钟

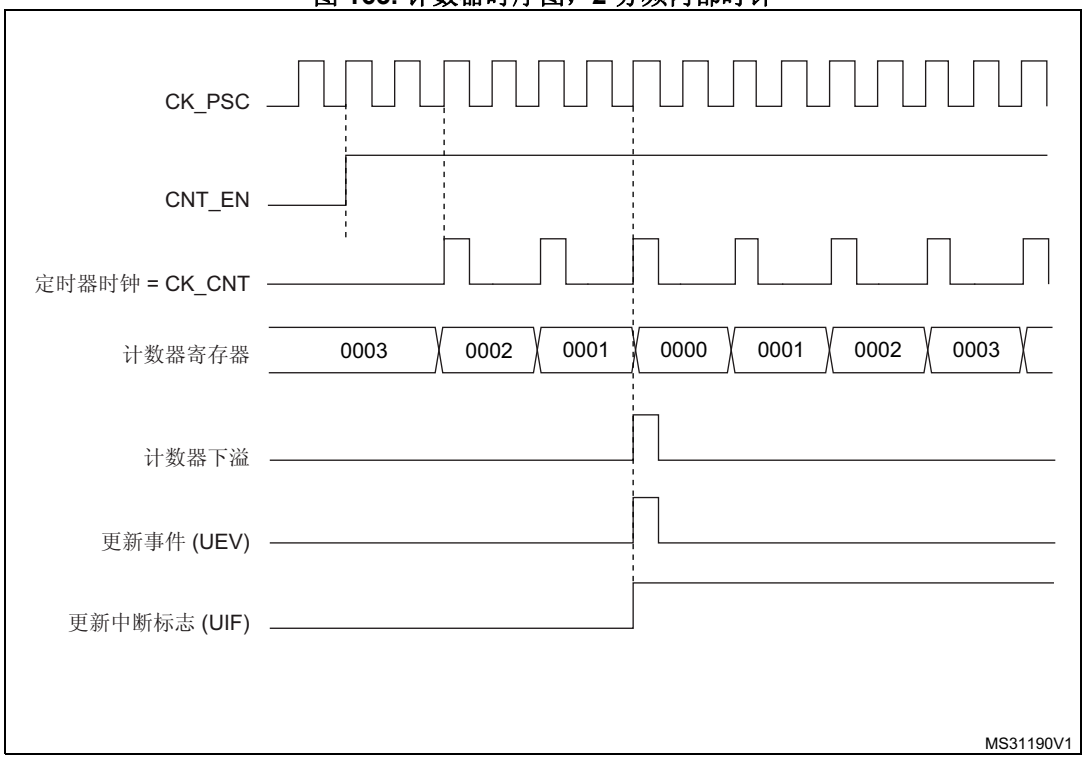


图 164. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36

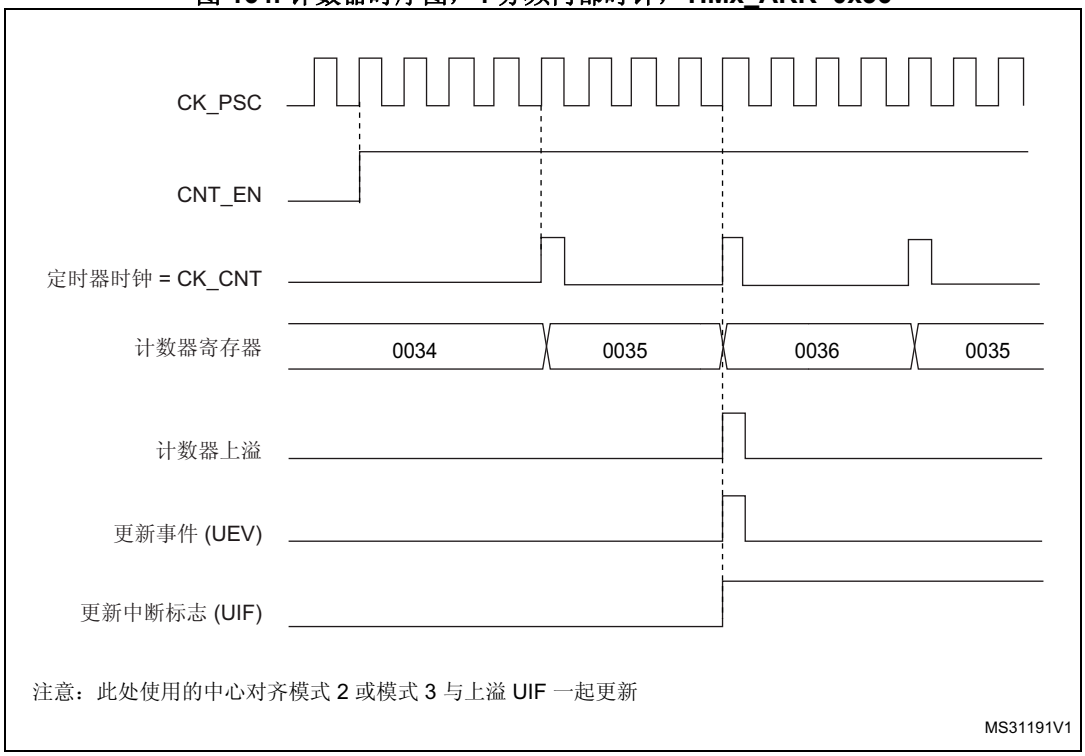


图 165. 计数器时序图，N 分频内部时钟

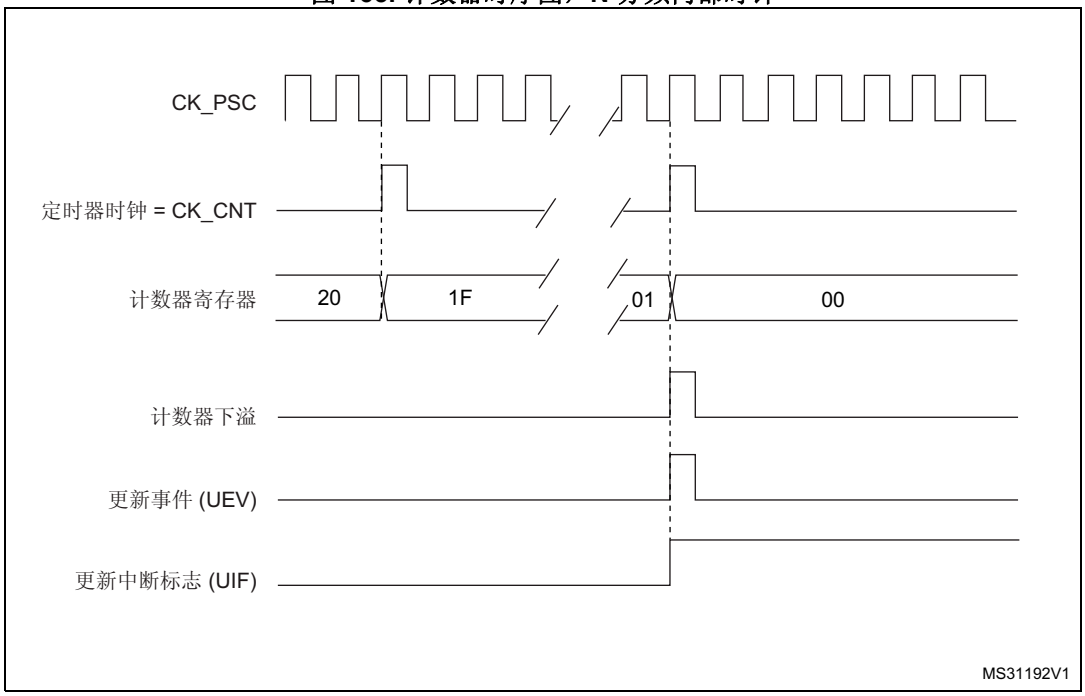


图 166. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

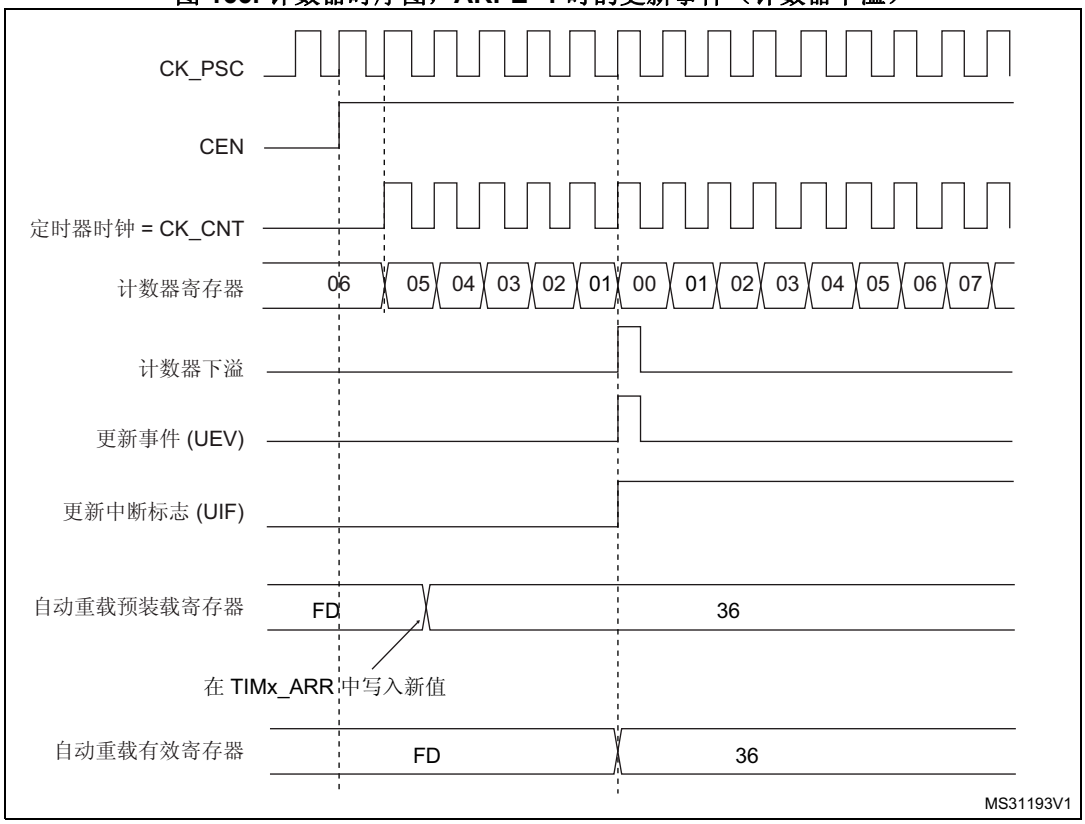
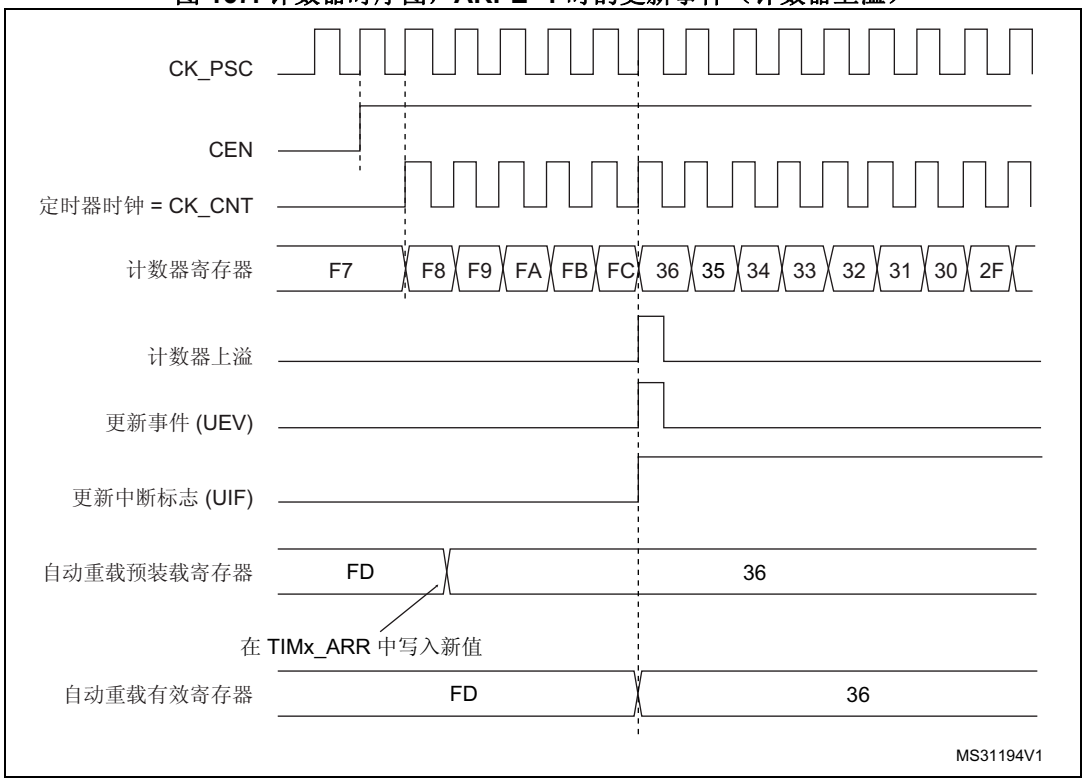


图 167. 计数器时序图，ARPE=1 时的更新事件（计数器上溢）



24.3.3 重复计数器

*第 24.3.1 节：时基单元*介绍如何因计数器上溢/下溢而生成更新事件 (UEV)。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

这意味着，每当发生 N+1 个计数器上溢或下溢（其中，N 是 TIMx_RCR 重复计数器寄存器中的值），数据就将从预装载寄存器转移到影子寄存器（TIMx_ARR 自动重载寄存器、TIMx_PSC 预分频器寄存器以及比较模式下的 TIMx_CCRx 捕获/比较寄存器）。

重复计数器在下列情况下递减：

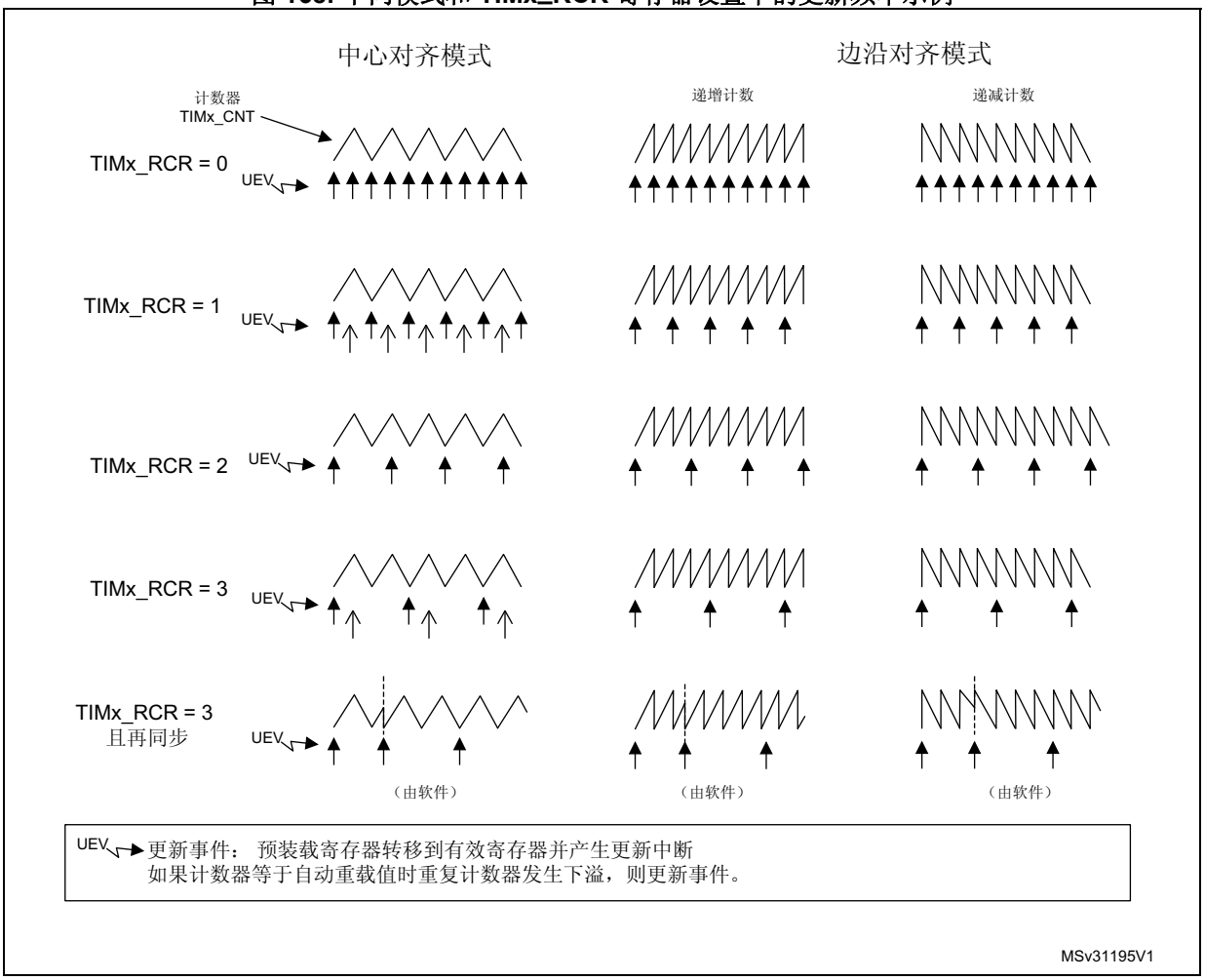
- 递增计数模式下的每个计数器上溢。
- 递减计数模式下的每个计数器下溢。
- 中心对齐模式下每个计数器上溢和计数器下溢。尽管这使得最大重复次数不超过 32768 个 PWM 周期，但在每个 PWM 周期内可更新占空比两次。当在中心对齐模式下，每个 PWM 周期仅刷新一次比较寄存器时，由于模式的对称性，最大分辨率为 $2xT_{ck}$。

重复计数器是自动重载类型；其重复率为 TIMx_RCR 寄存器所定义的值（请参见*图 168*）。当更新事件由软件（通过将 TIMx_EGR 寄存器的 UG 位置 1）或硬件（通过从模式控制器）生成时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TIMx_RCR 寄存器的内容。

在中心对齐模式下，如果 RCR 值为奇数，更新事件将在上溢或下溢时发生，这取决于何时写入 RCR 寄存器以及何时启动计数器：如果在启动计数器前写入 RCR，则 UEV 在上溢时发生。如果在启动计数器后写入 RCR，则 UEV 在下溢时发生。

例如，如果 RCR = 3，UEV 将在每个周期的第四个上溢或下溢事件时产生（取决于何时写入 RCR）。

图 168. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例



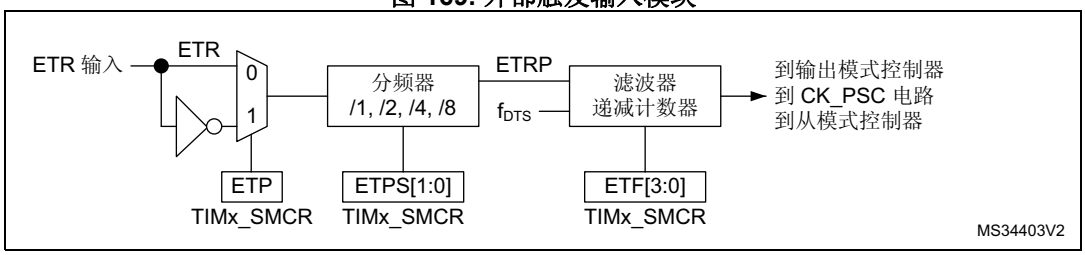
24.3.4 外部触发输入

定时器具有一个外部触发输入 ETR，它可用作：

- 外部时钟（外部时钟模式 2，请参见*第 24.3.5 节*）
- 用于从模式的触发信号（请参见*第 24.3.26 节*）
- 用于逐周期电流调节的 PWM 复位输入（请参见*第 24.3.7 节*）

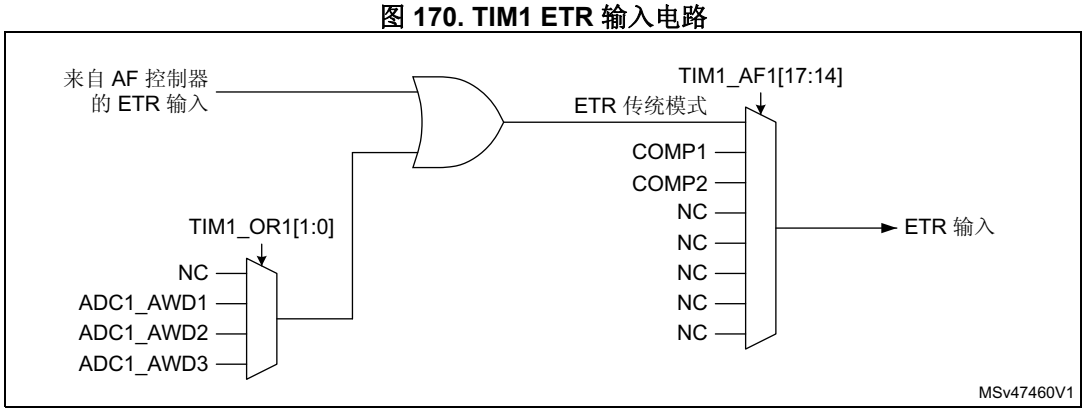
下面的*图 169* 介绍了 ETR 输入的调节过程。输入极性通过 TIMx_SMCR 寄存器中的 ETP 位定义。触发信号可通过 ETPS[1:0] 位域编程的分频比进行预分频，然后通过 ETF[3:0] 位域设置的滤波方式进行数字滤波。

图 169. 外部触发输入模块



ETR 输入来自多个源：输入引脚（默认配置）、比较器输出和模拟看门狗。使用 ETRSEL[3:0] 和 TIM1_OR1[1:0] 位域进行选择。

图 170. TIM1 ETR 输入电路



24.3.5 时钟选择

计数器时钟可由下列时钟源提供：

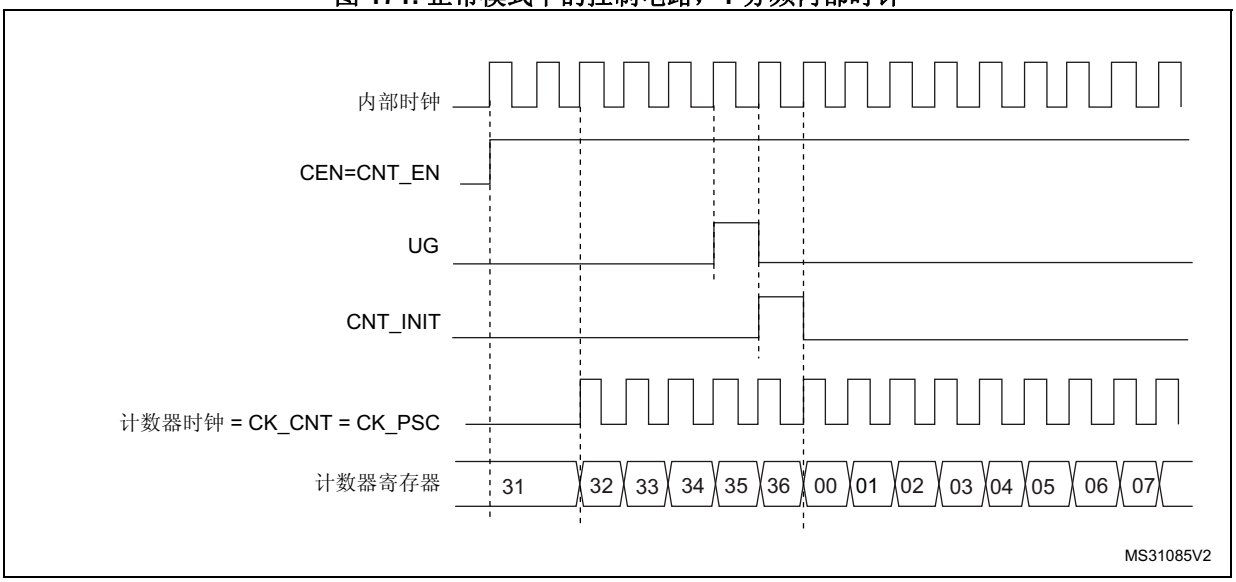
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 编码器模式

内部时钟源 (CK_INT)

如果禁止从模式控制器 (SMS=000)，则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 171 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

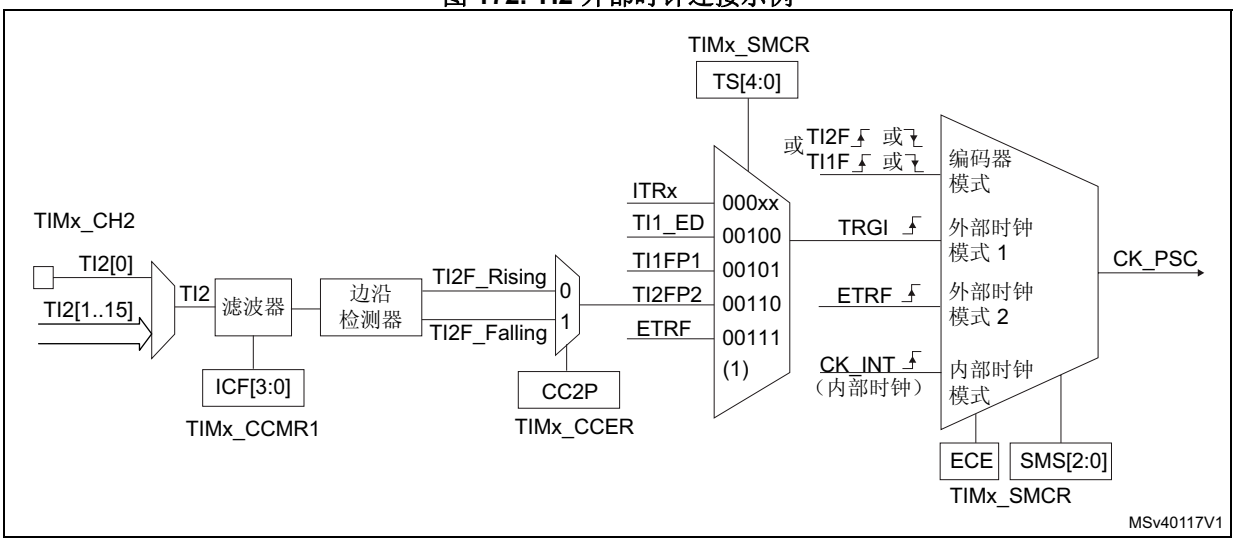
图 171. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 172. TI2 外部时钟连接示例



1. 保留从 01000 到 11111 的代码

例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

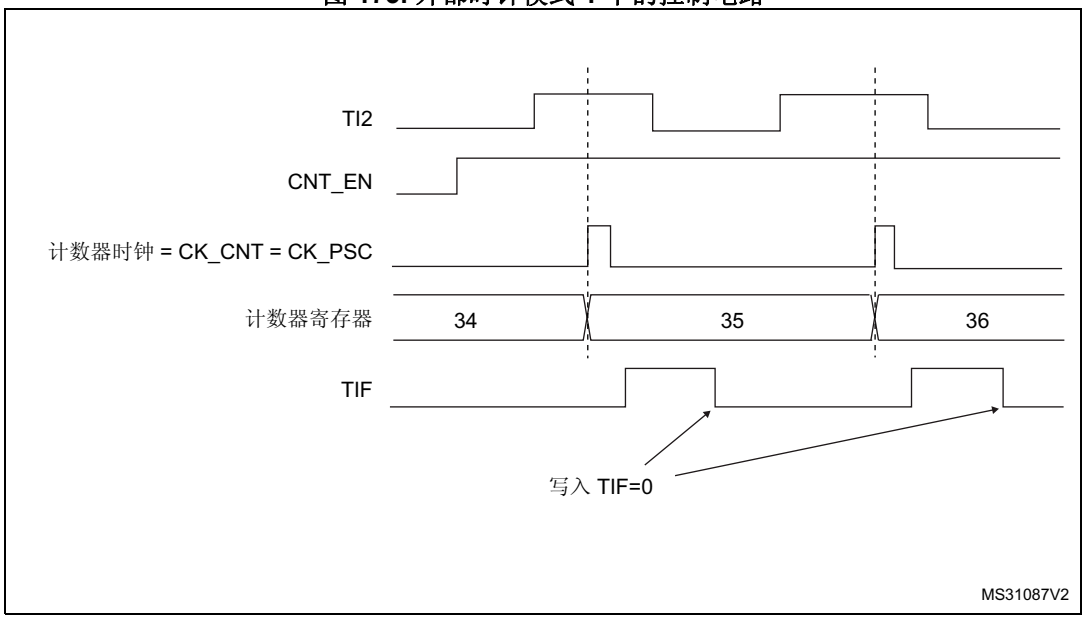
1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2x 源（内部或外部）。
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S = “01” 来配置通道 2，使其能够检测 TI2 输入的上升沿。
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。
4. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为触发输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

注： *由于捕获预分频器不用于触发操作，因此用户无需对其进行配置。*

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 173. 外部时钟模式 1 下的控制电路



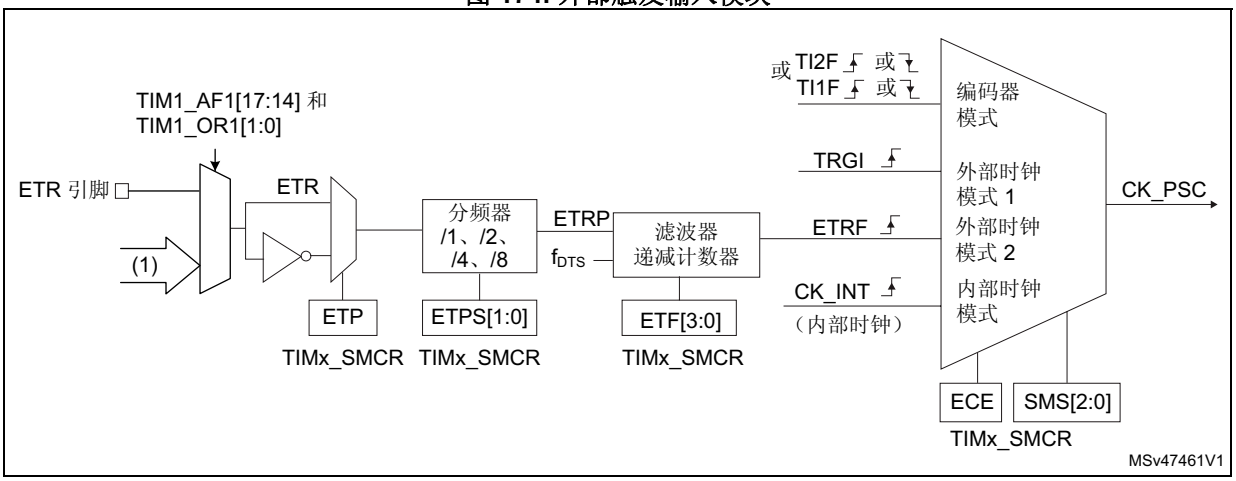
外部时钟源模式 2

通过在 TIMx_SMCR 寄存器中写入 ECE=1 可选择此模式。

计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

图 174 简要介绍了外部触发输入模块。

图 174. 外部触发输入模块



1. 请参见*图 170：TIM1 ETR 输入电路*。

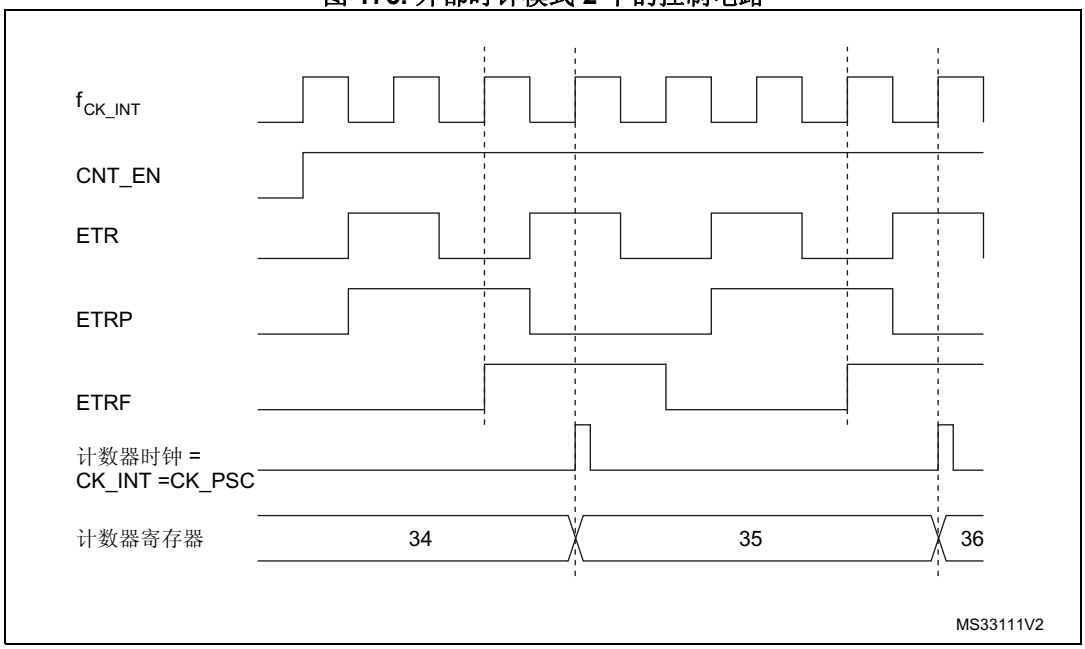
例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF[3:0]=0000。
2. 通过在 TIMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
3. 通过在 TIMx_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
4. 通过在 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
5. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的。

图 175. 外部时钟模式 2 下的控制电路



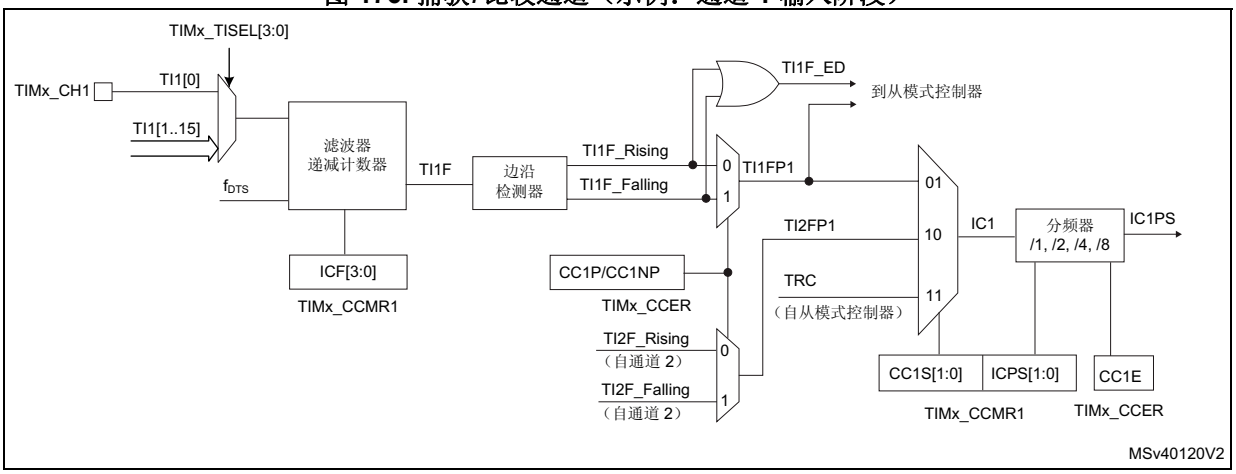
24.3.6 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器，通道 5 和通道 6 除外）和一个输出阶段（比较器和输出控制）构建而成。

图 176 到*图 179* 简要介绍了一个捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 176. 捕获/比较通道（示例：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。输出通道链末端决定最终输出信号的极性。

图 177. 捕获/比较通道 1 主电路

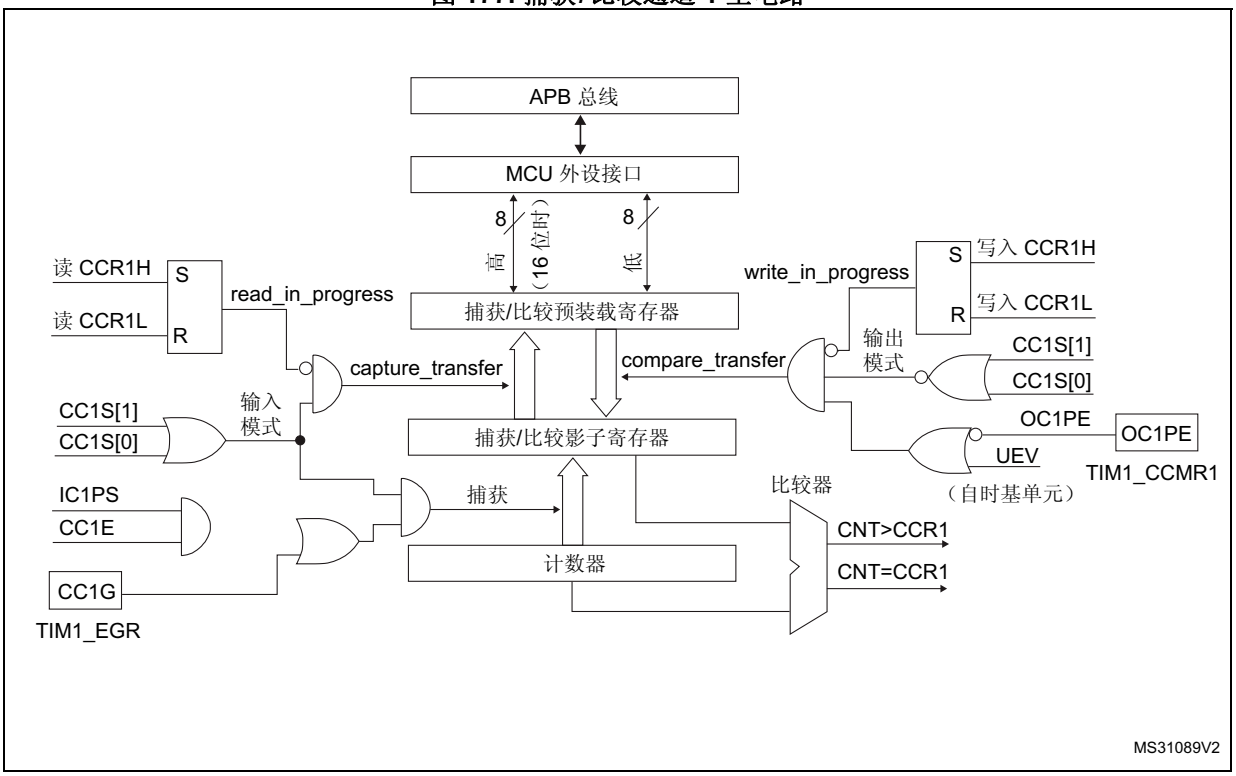
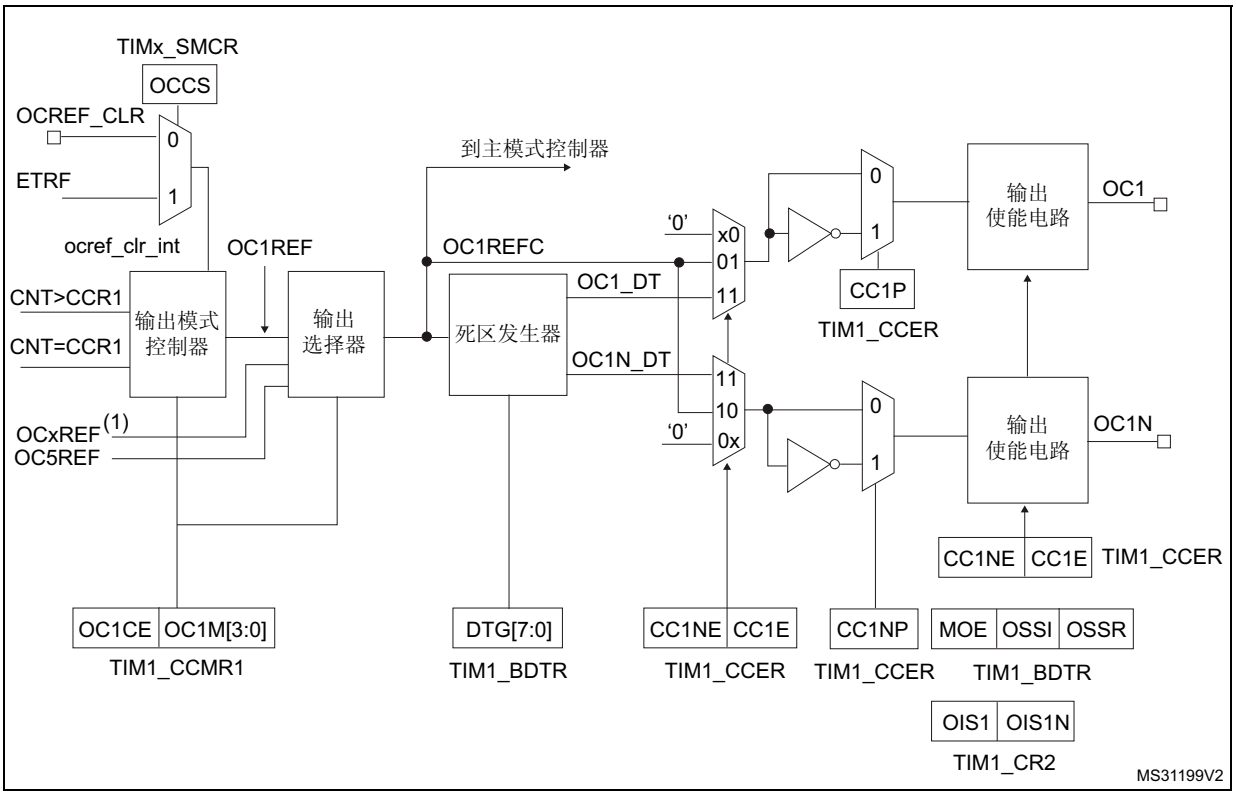


图 178. 捕获/比较通道的输出阶段（通道 1、通道 2 和通道 3）



1. OCxREF，其中 x 为互补通道的序号

图 179. 捕获/比较通道的输出阶段（通道 4）

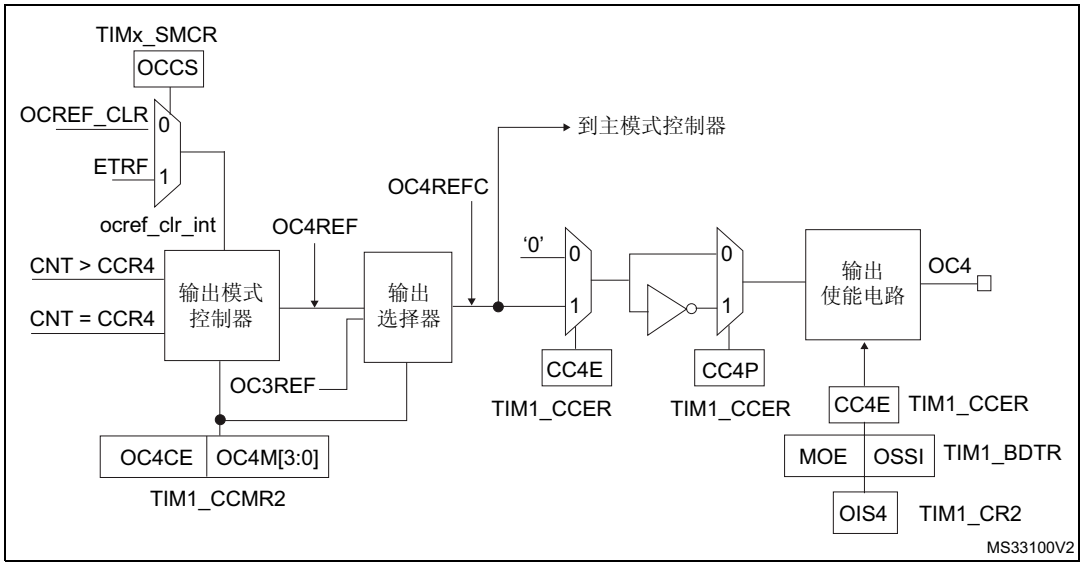
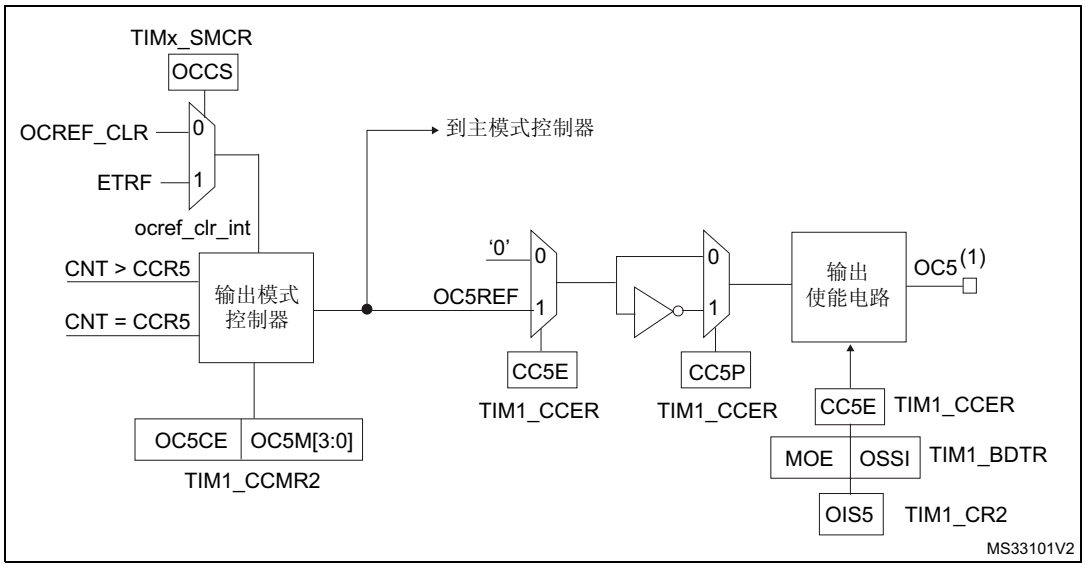


图 180. 捕获/比较通道的输出阶段（通道 5 和通道 6）



1. 不适用于外部。

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。读写操作访问的都是预加载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容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将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

24.3.7 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已置高位，则会将过捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可以通过软件向 CCxIF 写“0”或是读取存储在 TIMx_CCRx 寄存器中的捕获数据将 CCxIF 清零。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据连接到定时器的信号，对相关输入滤波带宽进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设输入信号发生翻转时，信号需要 5 个内部时钟周期才能稳定，则必须设置滤波时间大于 5 个内部时钟周期。若在 TI1 上连续 8 次检测到新电平采样值(以 f_{DST} 频率采样)判定沿跳变合法，则向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。

4. 通过在 TIMx_CCER 寄存器中将 CC1P 位和 CC1NP 位写入 0，选择 TI1 上的有效转换边沿（本例中为上升沿）。
5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理过捕获，建议在读出过捕获标志之前读取数据。这样可以避免丢失在读取过捕获标志后与读取捕获数据前可能发生的过捕获。

注： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

24.3.8 PWM 输入模式

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

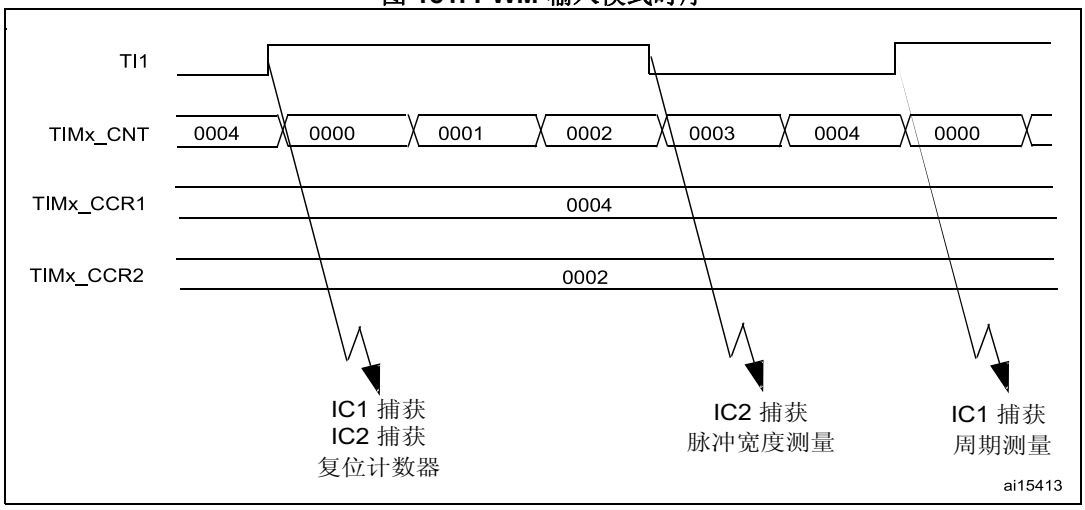
- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，用户可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
3. 选择 TI1FP1 的有效极性（用于在 TIMx_CCR1 中捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
4. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
5. 选择 TI1FP2 的有效极性（用于在 TIMx_CCR2 中捕获）：向 CC2P 位和 CC2NP 位写入 CC2P/CC2NP=“10”（下降沿有效）。

6. 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 00101（选择 TI1FP1）。
7. 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 0100。
8. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 181. PWM 输入模式时序



24.3.9 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx/OCxN）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (OCxREF/OCx) 强制设置为有效电平，用户只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 0101。OCxREF 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=0（OCx 高电平有效）=> 将 OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 0100，可将 OCxREF 信号强制设置为低电平。

无论如何，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。下面的输出比较模式一节对此进行了介绍。

24.3.10 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。通道 1 到通道 4 可用作输出，而通道 5 和通道 6 只能在器件内部使用（例如，用于产生混合波形或触发 ADC）。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCxM=0000)，也可设置为有效电平 (OCxM=0001)、无效电平 (OCxM=0010) 或进行翻转 (OCxM=0011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

使用 TIMx_CCMRx 寄存器中的 OCxPE 位，可将 TIMx_CCRx 寄存器配置为带或不带预装载寄存器。

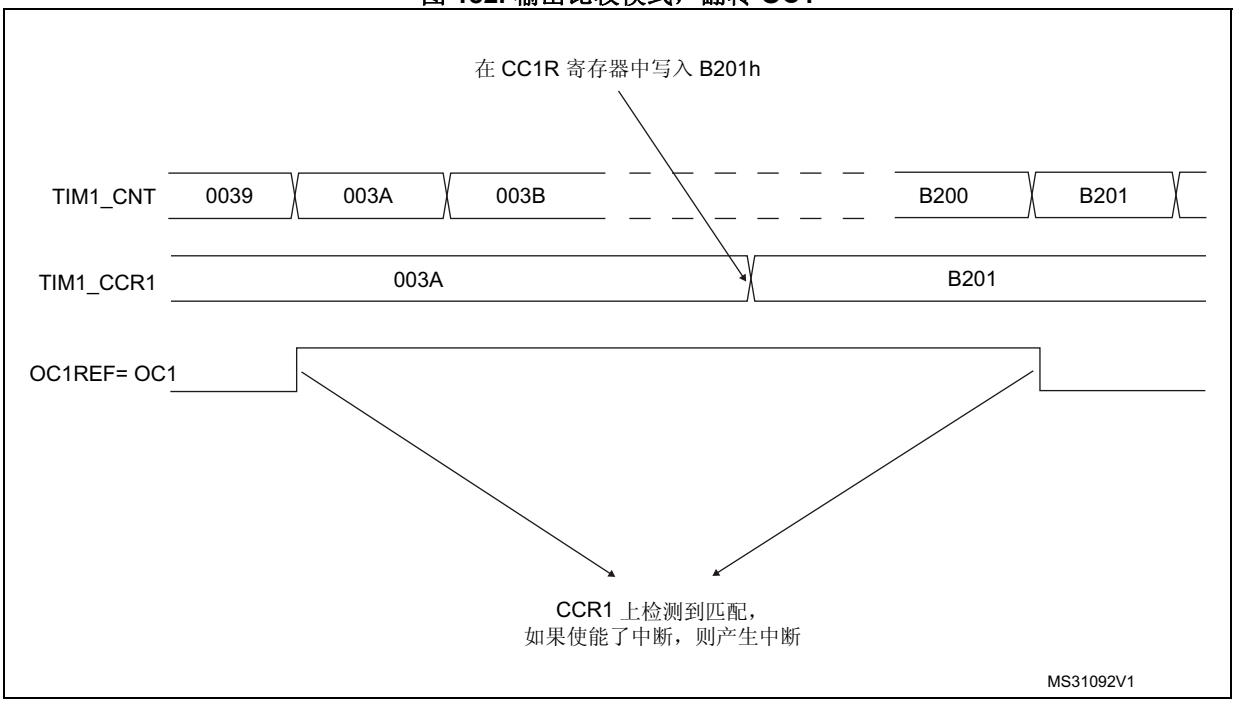
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   - 当 CNT 与 CCRx 匹配时，写入 OCxM = 0011 以翻转 OCx 输出引脚
   - 写入 OCxPE = 0 以禁止预装载寄存器
   - 写入 CCxP = 0 以选择高电平有效极性
   - 写入 CCxE = 1 以使能输出
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可通过软件随时更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=“0”，否则 TIMx_CCRx 影子寄存器仅在下一更新事件 UEV 发生时进行更新）。*图 182* 给出了一个示例。

图 182. 输出比较模式，翻转 OC1



24.3.11 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

各通道可以独立选择 PWM 模式（每个 OCx 输出对应一个 PWM），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“0110”（PWM 模式 1）或“0111”（PWM 模式 2）。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

由于只有在发生更新事件时预装载寄存器才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIMx_CCER 和 TIMx_BDTR 寄存器）的组合使能 OCx 输出。有关详细信息，请参见 TIMx_CCER 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 总是与 TIMx_CCRx 进行比较，以确定是 TIMx_CCRx ≤ TIMx_CNT 还是 TIMx_CNT ≤ TIMx_CCRx（取决于计数器计数方向）。

根据 TIMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

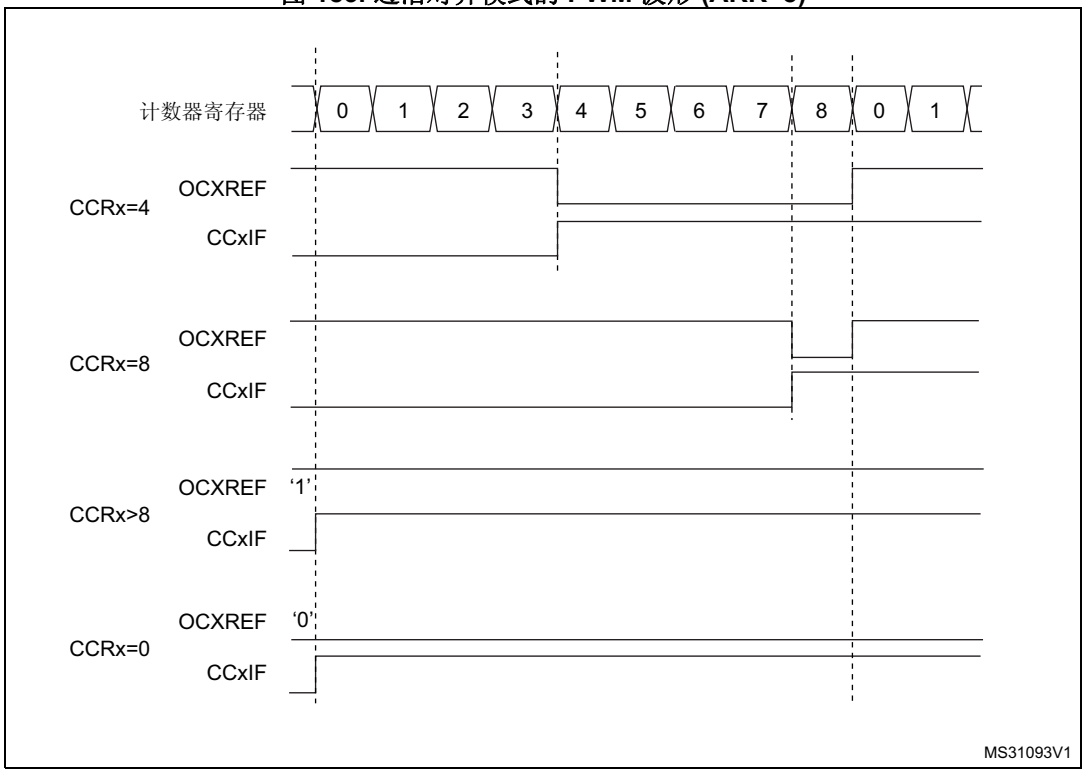
PWM 边沿对齐模式

- 递增计数配置

  当 TIMx_CR1 寄存器中的 DIR 位为低时执行递增计数。请参见*第 649 页的递增计数模式*。

  以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。*图 183* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 183. 边沿对齐模式的 PWM 波形 (ARR=8)



- 递减计数配置

  当 TIMx_CR1 寄存器中的 DIR 位为高时执行递减计数。请参见*第 653 页的递减计数模式*。

  在 PWM 模式 1 下，只要 TIMx_CNT > TIMx_CCRx，参考信号 OCxRef 即为低电平，否则其为高电平。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重载值，则 OCxREF 保持为“1”。此模式下不可能产生 0% 的 PWM 波形。

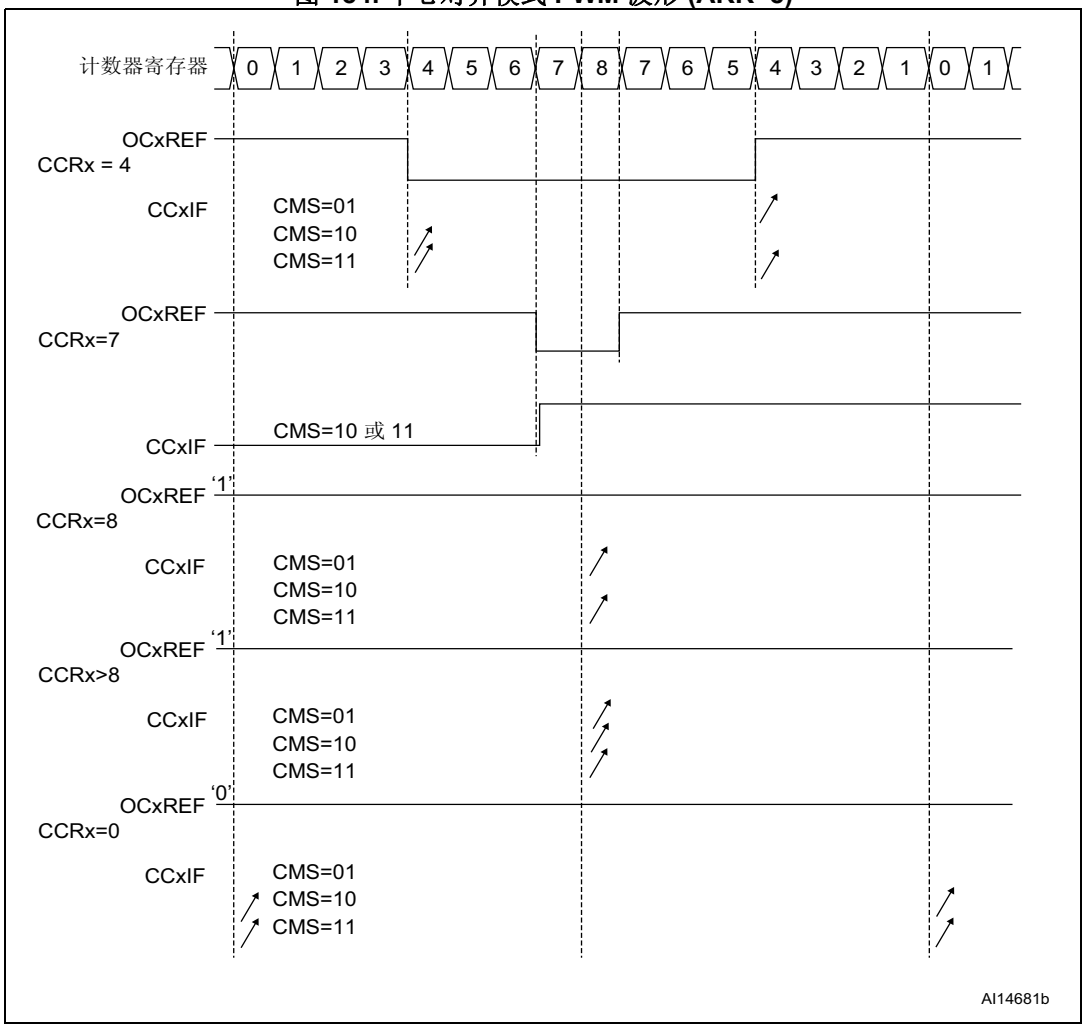
PWM 中心对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为“00”（其余所有配置对 OCxRef/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIMx_CR1 寄存器中的方向位 (DIR) 由硬件更新，不得通过软件更改。请参见*第 656 页的中心对齐模式（递增/递减计数）*。

图 184 显示了中心对齐模式的 PWM 波形，在此例中：

- TIMx_ARR=8。
- PWM 模式为 PWM 模式 1。
- 在根据 TIMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

图 184. 中心对齐模式 PWM 波形 (ARR=8)



中心对齐模式使用建议

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：
  - 如果写入计数器的值大于自动重载值 (TIMx_CNT>TIMx_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
  - 如果向计数器写入 0 或 TIMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

24.3.12 不对称 PWM 模式

在不对称模式下，生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和相移则由一对 TIMx_CCRx 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM，这样每半个 PWM 周期便会调节一次 PWM：

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

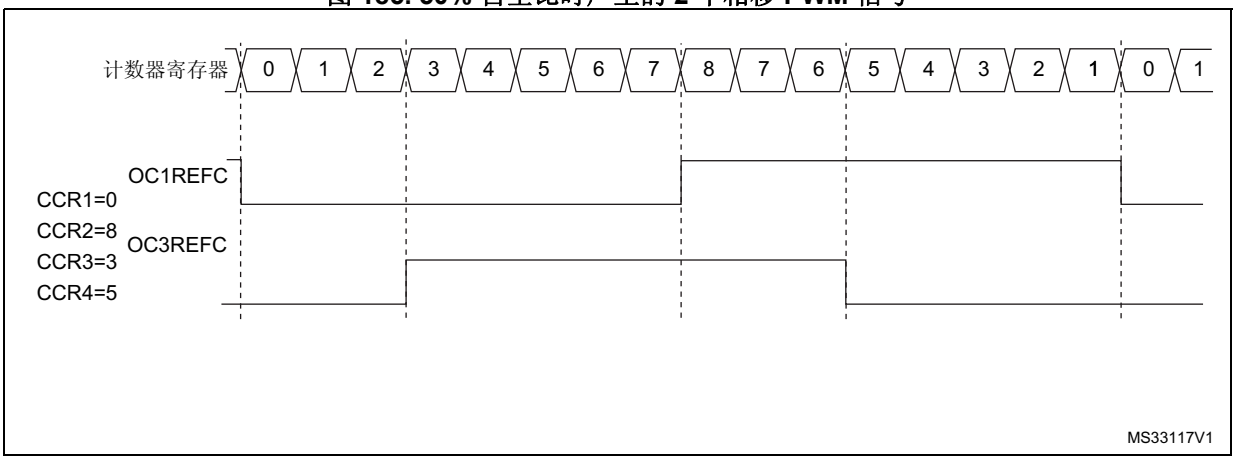
两个通道可以独立选择不对称 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1110”（不对称 PWM 模式 1）或“1111”（不对称 PWM 模式 2）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

给定通道用作不对称 PWM 通道时，也可使用其互补通道。例如，如果通道 1 上产生 OC1REFC 信号（不对称 PWM 模式 1），则由于不对称 PWM 模式 1 的原因，通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

图 185 显示了不对称 PWM 模式下可以产生的信号示例（通道 1 到通道 4 在不对称 PWM 模式 1 下配置）。与死区发生器配合使用时，这可控制相移全桥直流到直流转换器。

图 185. 50% 占空比时产生的 2 个相移 PWM 信号



24.3.13 组合 PWM 模式

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的逻辑或运算或者逻辑与运算组合组成。

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

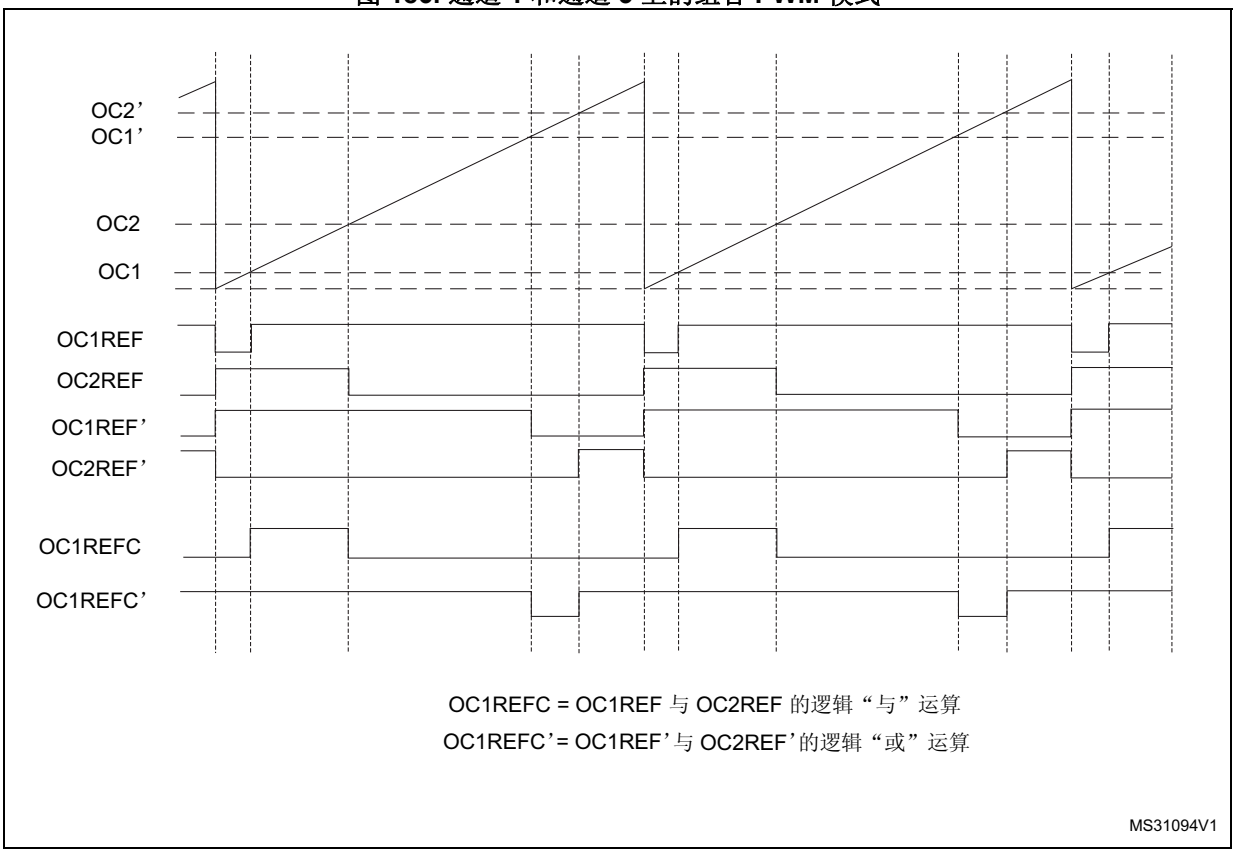
当给定通道用作组合 PWM 通道时，其互补通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 186 显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 在组合 PWM 模式 2 下配置。
- 通道 2 在 PWM 模式 1 下配置。
- 通道 3 在组合 PWM 模式 2 下配置。
- 通道 4 在 PWM 模式 1 下配置。

图 186. 通道 1 和通道 3 上的组合 PWM 模式



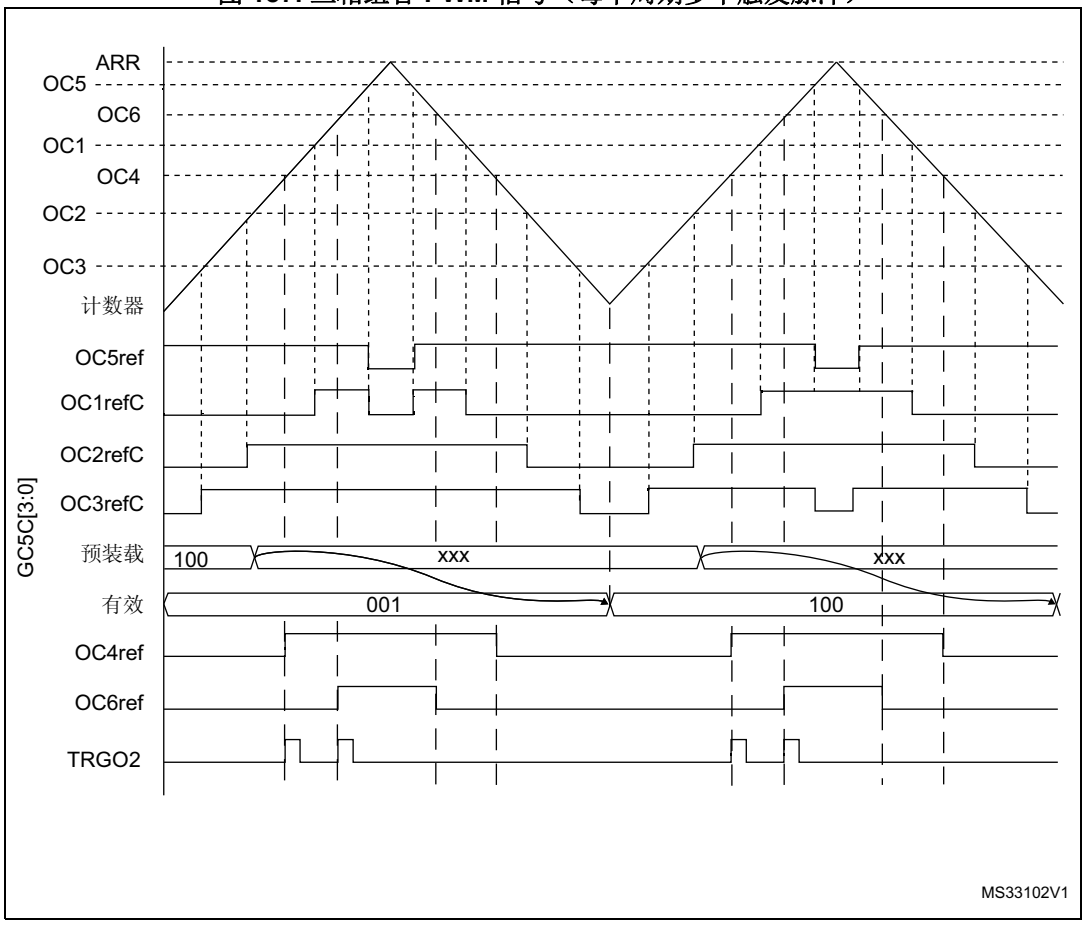
24.3.14 组合三相 PWM 模式

在组合三相 PWM 模式下，其 PWM 信号可由一至三路中心对称的 PWM 信号与另一路可编程信号在脉冲中点进行逻辑与运算产生。OC5REF 信号用于定义产生的组合信号。通过 TIMx_CCR5 中的 3 位 GC5C[3:1]，可以选择 OC5REF 与哪个参考信号组合。产生的信号 OCxREFC 由两个参考 PWM 的逻辑与运算组合组成。

- 如果 GC5C1 置 1，则 OC1REFC 由 TIMx_CCR1 和 TIMx_CCR5 控制
- 如果 GC5C2 置 1，则 OC2REFC 由 TIMx_CCR2 和 TIMx_CCR5 控制
- 如果 GC5C3 置 1，则 OC3REFC 由 TIMx_CCR3 和 TIMx_CCR5 控制

通道 1 到通道 3 可独立选择组合三相 PWM 模式，只需将 3 位 GC5C[3:1] 中的至少一位置 1。

图 187. 三相组合 PWM 信号（每个周期多个触发脉冲）



TRGO2 波形说明了如何根据给定的三相 PWM 信号同步 ADC。更多详细信息，请参见*第 24.3.27 节：ADC 同步*。

24.3.15 互补输出和死区插入

高级控制定时器 (TIM1) 可以输出两路互补信号，并管理输出的关断与接通瞬间。

这段时间通常称为死区，用户必须根据与输出相连接的器件及其特性（电平转换器的固有延迟、开关器件产生的延迟...）来调整死区时间。

每路输出可以独立选择输出极性（主输出 OCx 或互补输出 OCxN）。可通过对 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位执行写操作来完成极性选择。

互补信号 OCx 和 OCxN 通过以下多个控制位的组合进行激活：TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。更多详细信息，请参见*第 723 页的表 155：具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位*。应当注意，切换至空闲状态（MOE 下降到 0）的时刻，死区仍然有效。

CCxE 和 CCxNE 位同时置 1 并且 MOE 位置 1（若存在刹车电路）时，将使能死区插入。每个通道有一个 10 位死区发生器。将基于参考波形 OCxREF 生成 2 个输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效：

- 输出信号 OCx 与参考信号相同，只是其上升沿相对参考上升沿存在延迟。
- 输出信号 OCxN 与参考信号相反，并且其上升沿相对参考下降沿存在延迟。

如果延迟时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。

下图所示为死区发生器的输出信号与参考信号 OCxREF 之间的关系。（在这些示例中，假定 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）

图 188. 带死区插入的互补输出

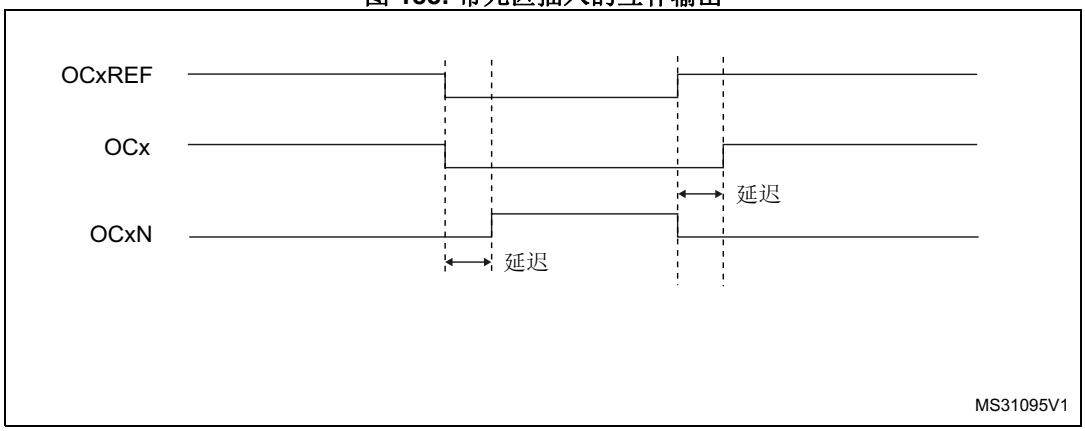


图 189. 延迟时间大于负脉冲宽度的死区波形

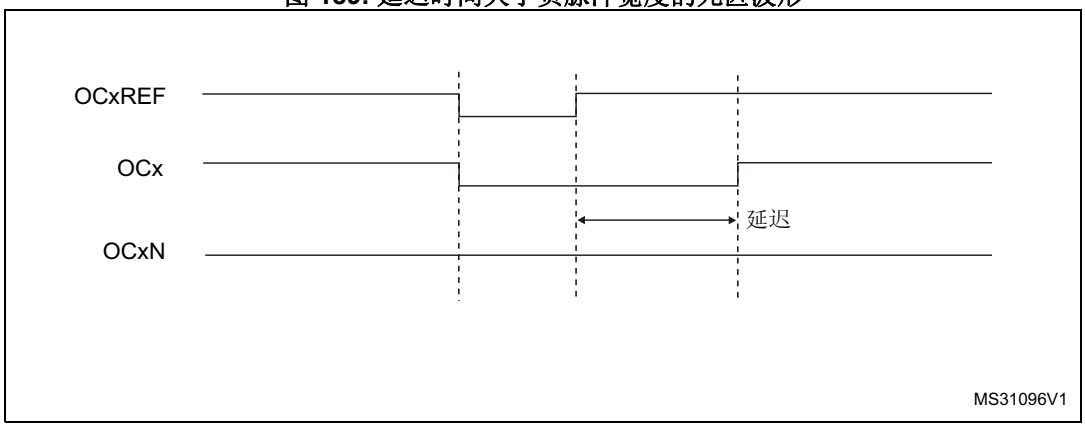
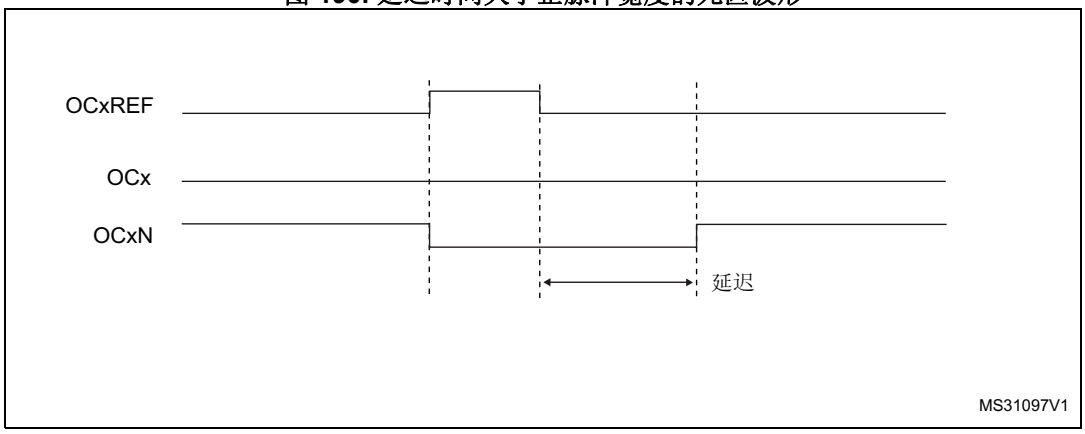


图 190. 延迟时间大于正脉冲宽度的死区波形



死区延迟对于所有通道均相同，可通过 TIMx_BDTR 寄存器中的 DTG 位进行编程。有关延迟时间计算的信息，请参见*第 24.4.20 节：TIM1 刹车和死区寄存器 (TIM1_BDTR)*。

将 OCxREF 重定向到 OCx 或 OCxN

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位，可将 OCxREF 重定向到 OCx 输出或 OCxN 输出。

通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者，使两个输出同时保持无效电平，或者两个输出同时处于有效电平，两者互补并且带死区。

注： *如果仅使能 OCxN (CCxE=0, CCxNE=1)，两者不互补，一旦 OCxREF 为高电平，OCxN 即变为有效。例如，如果 CCxNP=0，则 OCxN=OCxRef。另一方面，如果同时使能 OCx 和 OCxN (CCxE=CCxNE=1)，OCx 在 OCxREF 为高电平时变为有效，而 OCxN 则与之互补，在 OCxREF 为低电平时变为有效。*

24.3.16 使用刹车功能

刹车功能的目的是保护由 TIM1 定时器生成的 PWM 信号所驱动的功率开关。两个刹车输入通常连接到功率级和三相逆变器的故障输出。激活时，刹车电路会关闭 PWM 输出，并将其强制为预定义的安全状态。也可选择一些内部 MCU 事件来触发输出关断。

有两个刹车通道。一个刹车通道收集系统级故障（时钟失效和奇偶校验错误等）和应用故障（来自输入引脚和内置比较器），可以在死区持续时间后将输出强制为预定义的电平（有效或无效）。刹车 2 通道只包括应用故障，能够将输出强制为无效状态。

刹车期间的输出使能信号和输出电平取决于多个控制位：

- TIMx_BDTR 寄存器中的 MOE 位，允许通过软件使能/禁止输出，在发生刹车和刹车 2 事件时复位。
- TIMx_BDTR 寄存器中的 OSSI 位，定义定时器将输出控制在无效状态下，还是将输出控制释放给 GPIO 控制器（通常使其处于高阻态模式）
- TIMx_CR2 寄存器中的 OISx 和 OISxN 位，将输出设置为关断电平（有效或无效）。无论 OISx 和 OISxN 的值为何，均无法在给定时间将 OCx 和 OCxN 输出同时设置为有效电平。更多详细信息，请参见*第 723 页的表 155：具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位*。

退出复位状态后，刹车功能处于禁止状态，MOE 位处于低电平。通过设置 TIMx_BDTR 寄存器中的 BKE 位和 BK2E 位来使能刹车功能。可通过配置同一寄存器中的 BKP 位和 BK2P 位来选择刹车输入的极性。BKE/BK2E 和 BKP/BK2P 位可同时修改。对 BKE/BK2E 和 BKP/BK2P 位执行写操作时，写操作会在 1 个 APB 时钟周期的延迟后生效。因此，执行写操作后，需要等待 1 个 APB 时钟周期，才能准确回读此位。

由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 TIMx_BDTR 寄存器中）之间插入了再同步电路，从而在异步信号与同步信号之间产生延迟。具体而言，如果在 MOE 处于低电平时将其置 1，则必须首先插入延迟（空指令），才能准确进行读取。这是因为写入的是异步信号，而读取的却是同步信号。

通过寄存器 TIMx_AF1 与 TIMx_AF2 可设置多个源来触发刹车，这些源可被单独使能且可设置有效沿。

刹车 (BRK) 通道的源为：

- 连接到 BKIN 引脚的外部源（由 SYSCFG_CFGR2 寄存器设定），具有极性选择和可选的数字滤波
- 内部源：
  - CPU1 Cortex®-M4 LOCKUP 输出
  - PVD 输出
  - SRAM 奇偶校验错误信号
  - Flash ECC 错误
  - CSS 检测器产生时钟故障事件
  - 比较器的输出，具有极性选择和可选的数字滤波

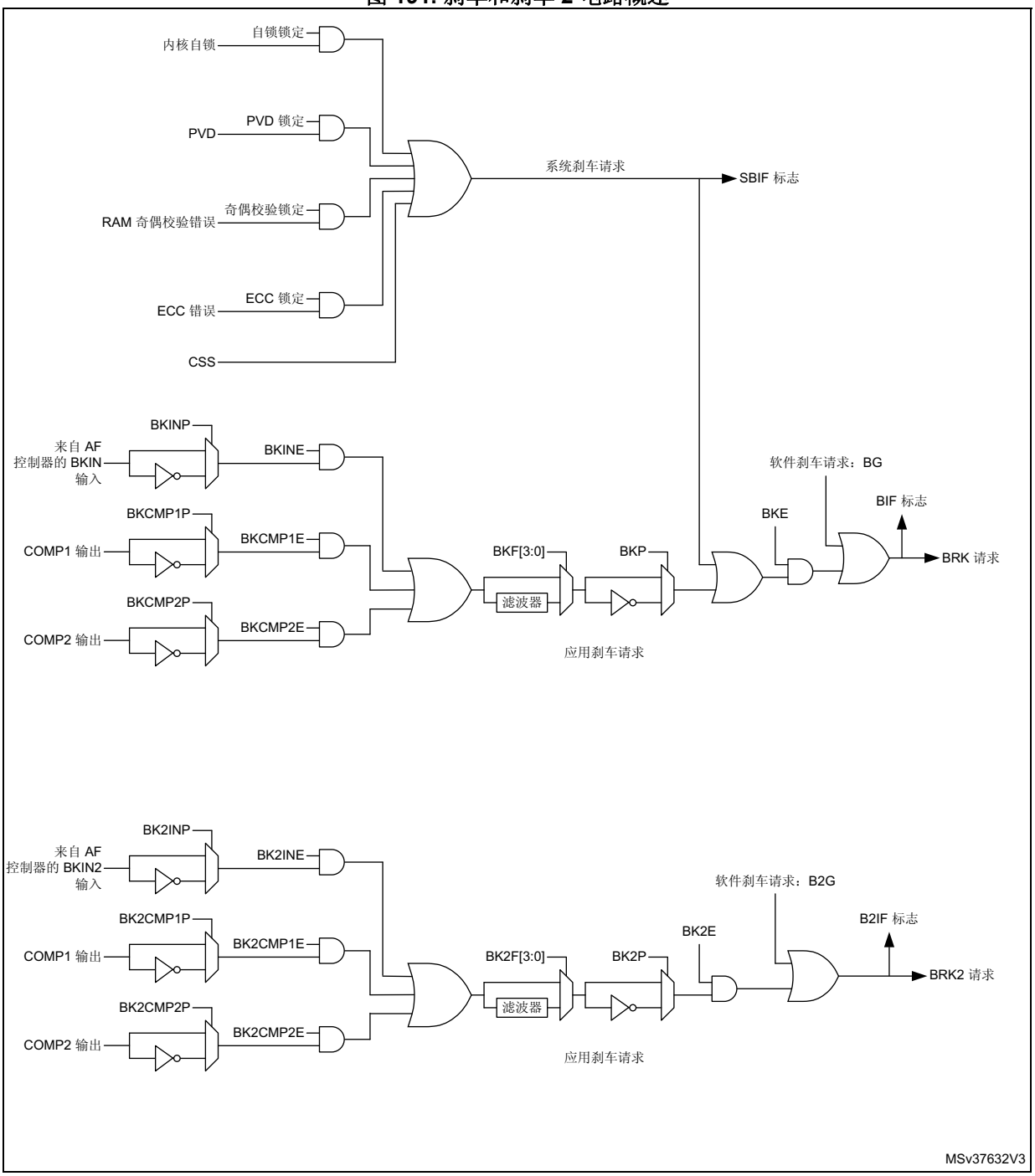
刹车 2 (BRK2) 的源：

- 连接到 BKIN 引脚的外部源（由 SYSCFG_CFGR2 寄存器设定），具有极性选择和可选的数字滤波。
- 来自比较器输出的内部源。

也可由软件通过 TIMx_EGR 寄存器中的 BG 和 B2G 位产生刹车事件。无论 BKE 和 BK2E 使能位的值如何，都可以使用 BG 和 B2G 通过软件生成刹车。

在所有源进入定时器 BRK 或 BRK2 输入之前，对其进行 OR 运算，如以下*图 191* 所示。

图 191. 刹车和刹车 2 电路概述



注： *只有禁止可编程滤波器时才能保证异步（无时钟）操作。如果使能可编程滤波器，必须使用故障安全时钟模式（例如，使用内部 PLL 和/或 CSS）来保证能够处理刹车事件。*

发生刹车之一（其中一个刹车输入上出现所选电平）时：

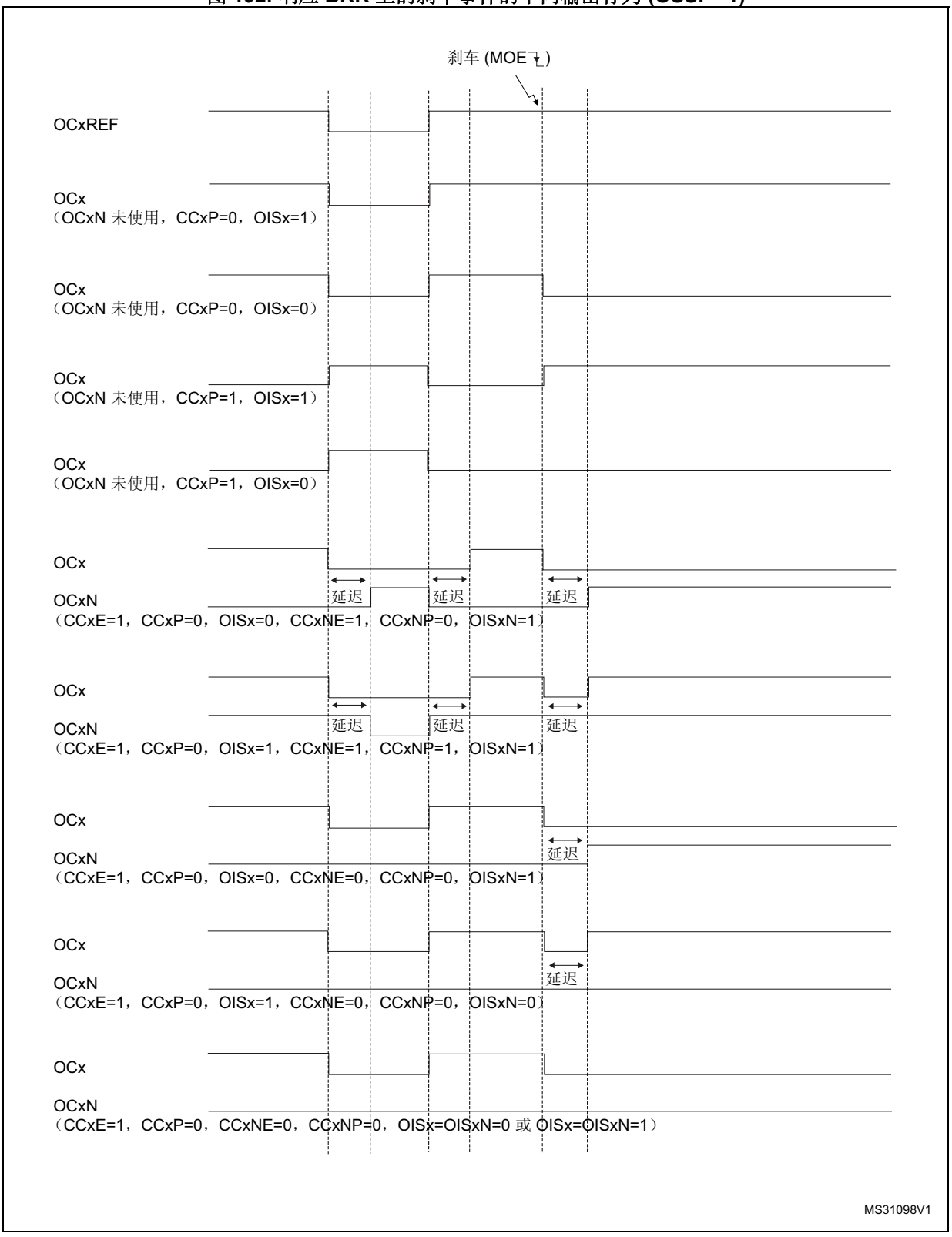
- MOE 位异步清零，使输出处于无效状态、空闲状态甚至释放控制权给 GPIO 控制器（通过 OSSI 位进行选择）。即使 MCU 振荡器关闭，该功能仍然使能。
- MOE=0 时，将以 TIMx_CR2 寄存器 OISx 位中编程的电平驱动每个输出通道。如果 OSSI=0，定时器将释放输出控制（由 GPIO 控制器接管），否则使能输出保持高电平。
- 使用互补输出时：
  - 输出首先置于无效状态（取决于极性）。这是异步操作，因此即使没有为定时器提供时钟，该操作仍有效。
  - 如果定时器时钟仍存在，则将重新激活死区发生器，进而在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，也不能同时将 OCx 和 OCxN 驱动至其有效电平。请注意，MOE 进行再同步，因此死区的持续时间会比通常情况稍长一些（约 2 个 ck_tim 时钟周期）。
  - 如果 OSSI=0，定时器将释放输出控制（由强制高阻态的 GPIO 控制器接管），否则使能输出将保持高电平或在 CCxE 或 CCxNE 位之一为高时立即变为高电平。
- 将刹车状态标志（TIMx_SR 寄存器中的 SBIF、BIF 和 B2IF 位）置 1。如果 TIMx_DIER 寄存器中的 BIE 位置 1，则会产生中断。
- 如果 TIMx_BDTR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件 (UEV) 时自动再次置 1。例如，这可用于执行调节。否则，MOE 将始终保持低电平，直到应用将其再次置 1。这种情况下，这一特性可用于确保安全。可以将刹车输入连接到功率驱动器、温度传感器或任何安全元件的告警。

注： *刹车输入为电平有效。因此，刹车输入有效电平时，不能将 MOE 位置 1（自动或通过软件）。同时，不能将状态标志 BIF 和 B2IF 清零。*

除刹车输入和输出管理外，刹车电路内部还实施了写保护，用以保护应用的安全。通过该功能，用户可冻结多个参数配置（死区持续时间、OCx/OCxN 极性和禁止时的状态、OCxM 配置、刹车使能和极性）。应用可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参见*第 24.4.20 节：TIM1 刹车和死区寄存器 (TIM1_BDTR)*。MCU 复位后只能对 LOCK 位执行一次写操作。

图 192 所示为输出对刹车响应行为的示例。

图 192. 响应 BRK 上的刹车事件的不同输出行为 (OSSI = 1)



两个刹车输入针对定时器输出具有不同的行为：

- BRK 输入可禁止（无效状态）PWM 输出，也可将 PWM 输出强制为预定义的安全状态。
- BRK2 只能禁止（无效状态）PWM 输出。

BRK 输入的优先级高于 BRK2 输入，如*表 151* 所示。

注： *BRK2 必须只在 OSSR = OSSI = 1 时使用。*

表 151. 定时器输出行为与 BRK/BRK2 输入

BRK	BRK2	定时器输出状态	典型用例	
			OCxN 输出（下桥臂开关）	OCx 输出（上桥臂开关）
有效	X	– 无效，之后强制为输出状态（死区后） – 如果 OSSI = 0，则禁止输出（由 GPIO 逻辑接管控制）	死区插入后开启	关闭
无效	有效	无效	关闭	关闭

图 193 给出了 BRK 和 BRK2 输入上出现有效信号时 OCx 和 OCxN 输出行为示例。在这种情况下，两个输出的极性均为高电平有效（TIMx_CCER 寄存器中的 CCxP = CCxNP = 0）。

图 193. BRK 和 BRK2 引脚使能后的 PWM 输出状态 (OSSI=1)

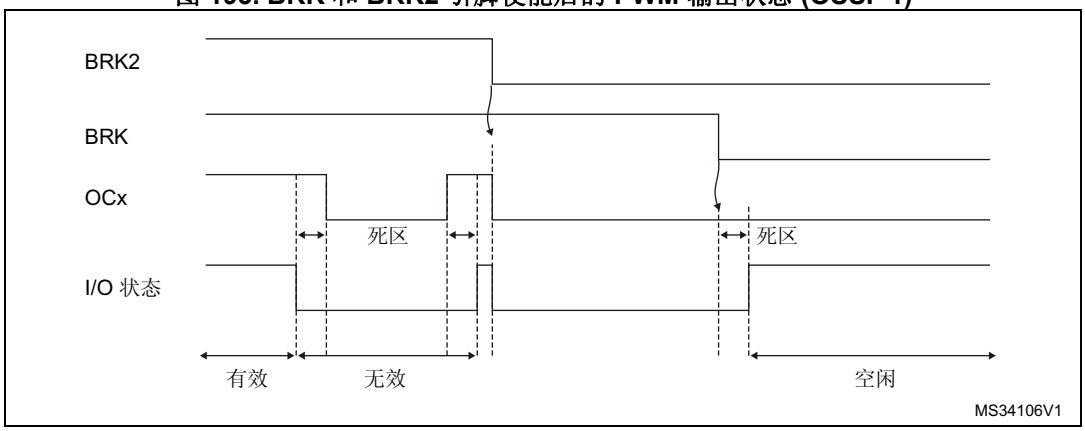
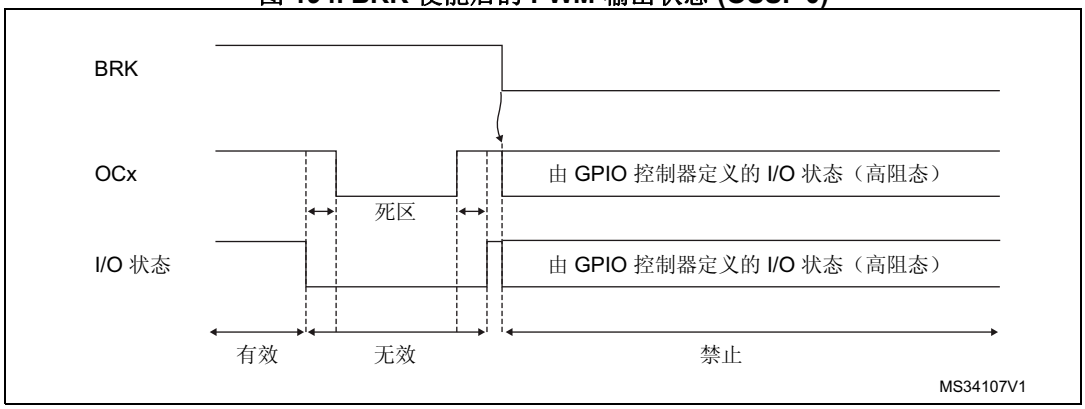


图 194. BRK 使能后的 PWM 输出状态 (OSSI=0)



24.3.17 双向刹车输入

TIM1 具有双向刹车 I/O，如*图 195* 所示。

它们可以：

- 将板级全局刹车信号用于向外部 MCU 或栅极驱动器发送故障信号，唯一的引脚作为输入和输出状态引脚。
- 在必须将多个内部和外部刹车输入合并时，将内部刹车源和多个外部开漏比较器输出“或”连接在一起，触发唯一刹车事件。

使用 TIMxBDTR 寄存器的 BKBID 和 BK2BID 位，将刹车和刹车 2 输入配置为双向模式。可以使用 TIMxBDTR 寄存器中的 LOCK 位，将 BKBID 编程位锁定在只读模式（锁定级别 1 或更高）。

双向模式可以用于刹车和刹车 2 输入，需要将 I/O 配置为低电平有效极性的开漏模式（使用 BKINP、BKP、BK2INP 和 BK2P 位进行配置）。任何来自系统（例如 CSS）、片上外设或刹车输入的刹车请求都会强制将刹车输入置为低电平，以通知发生了故障事件。如果未正确设置极性位（高电平有效极性），则出于安全目的禁止双向模式。

软件刹车事件（BG 和 B2G）也会导致刹车 I/O 被强制置为“0”，从而向外部组件指示定时器已进入刹车状态。但是仅在使能刹车 (BK(2)E = 1) 时有效。当生成软件刹车事件，并且 BK(2)E = 0 时，输出会置于安全状态，并将刹车标志置 1，但对刹车 (2) I/O 无影响。

安全解除机制可防止系统最终锁定（刹车输入上的低电平会触发刹车，进而将相同输入强制置为低电平）。

当 BKDSRM (BK2DSRM) 位置 1 时，会释放刹车输出以清除故障信号，从而使系统能够重新获得保护。

在任何情况下都不能禁止刹车保护电路：

- 刹车输入路径始终有效：即使 BKDSRM (BK2DSRM) 位置 1 且释放开漏控制，刹车事件也仍然有效。这样可以在发生刹车期间防止 PWM 输出重新启动。
- 使能输出（MOE 位置 1）后，BK(2)DSRM 位不能解除刹车保护（请参见*表 152*）。

表 152. 刹车保护解除条件

MOE	BKDIR (BK2DIR)	BKDSRM (BK2DSRM)	刹车保护状态
0	0	X	启动
0	1	0	启动
0	1	1	解除
1	X	X	启动

启动和重新启动刹车电路

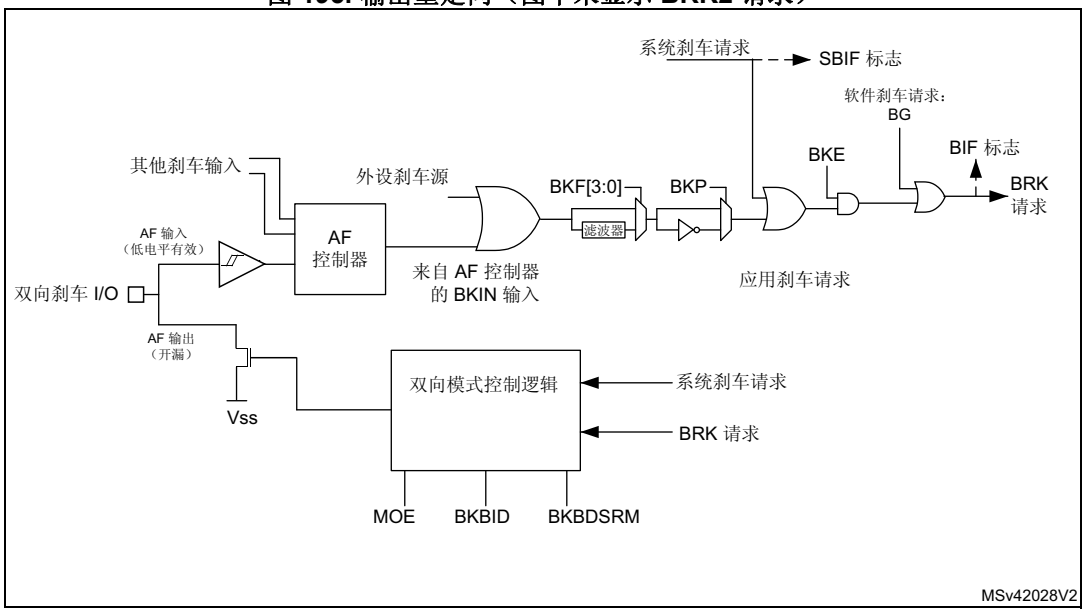
默认情况下（外设复位配置）会启动刹车电路（在输入或双向模式下）。

发生刹车（刹车 2）事件后，必须按照以下步骤重新启动保护：

- 必须将 BKDSRM (BK2DSRM) 位置 1，以释放输出控制
- 软件必须等待系统刹车条件消失（如果有），并清零 SBIF 状态标志（或在重新启动前由系统清零）
- 软件必须轮询 BKDSRM (BK2DSRM) 位，直到此位由硬件清零（当应用程序刹车条件消失时）

此后，刹车电路即启动并激活，可以通过将 MOE 位置 1 来重新使能 PWM 输出。

图 195. 输出重定向（图中未显示 BRK2 请求）



24.3.18 发生外部事件时清除 OCxREF 信号

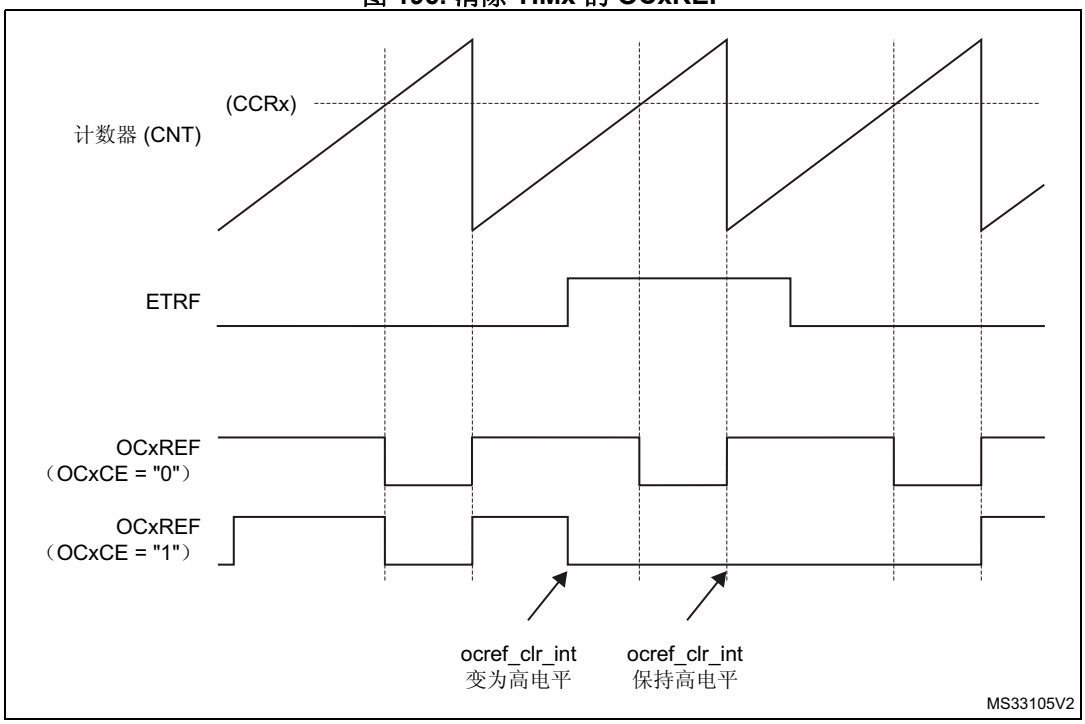
对于给定通道，在 ocref_clr_int 输入上施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号清零。OCxREF 信号将保持低电平，直到发生下一更新事件 (UEV)。该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。通过配置 TIMx_SMCR 寄存器中的 OCCS 位，可在 OCREF_CLR 输入和 ETRF（滤波器后的 ETR）之间选择 ocref_clr_int 输入。

选择 ETRF 时，ETR 必须配置如下：

1. 必须关闭外部触发预分频器：TIMx_SMCR 寄存器中的 ETPS[1:0] 位置“00”。
2. 必须禁止外部时钟模式 2：TIMx_SMCR 寄存器中的 ECE 位置“0”。
3. 外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可根据用户需要进行配置。

图 196 对比了使能位 OCxCE 在不同值下的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 TIMx 编程为 PWM 模式。

图 196. 清除 TIMx 的 OCxREF



注： *如果 PWM 的占空比为 100% (CCRx>ARR)，则下次计数器溢出时会再次使能 OCxREF。*

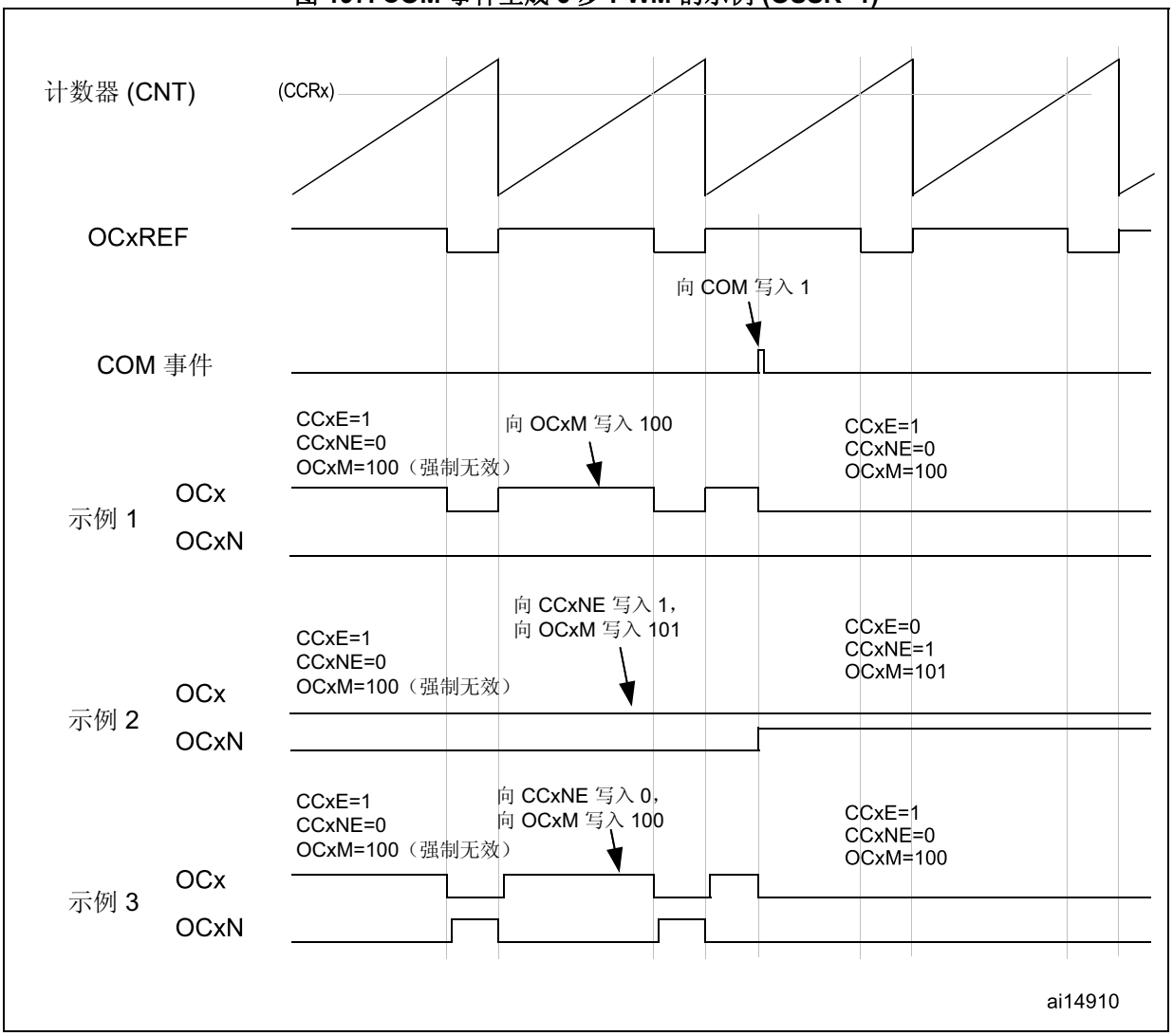
24.3.19 生成 6 步 PWM

当通道使用互补输出时，OCxM、CCxE 和 CCxNE 位上提供预装载位。发生 COM 换向事件时，这些预装载位将传输到影子位。因此，用户可以预先编程下一步骤的配置，并同时更改所有通道的配置。COM 可由软件通过将 TIMx_EGR 寄存器中的 COM 位置 1 而生成，也可以由硬件在 TRGI 上升沿生成。

发生 COM 事件时，某个标志位（TIMx_SR 寄存器中的 COMIF 位）将会置 1。这时，如果 TIMx_DIER 寄存器中的 COMIE 位置 1，将产生中断；如果 TIMx_DIER 寄存器中的 COMDE 位置 1，则将产生 DMA 请求。

图 197 以 3 种不同的编程配置为例，显示了发生 COM 事件时 OCx 和 OCxN 输出的行为。

图 197. COM 事件生成 6 步 PWM 的示例 (OSSR=1)



24.3.20 单脉冲模式

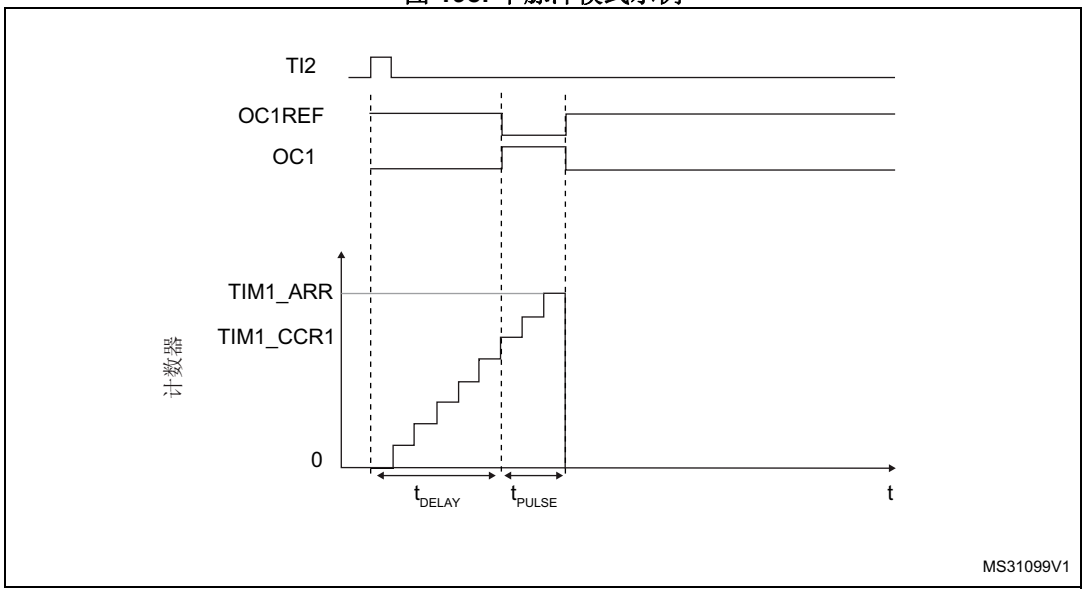
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。通过将 TIMx_CR1 寄存器中的 OPM 位置 1 选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- 递增计数时：CNT < CCRx ≤ ARR（特别注意，0 < CCRx）
- 递减计数时：CNT > CCRx

图 198. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到正沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2x 源（内部或外部）。
2. 在 TIMx_CCMR1 寄存器中写入 CC2S=“01”，以将 TI2FP2 映射到 TI2。
3. 在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
4. 在 TIMx_SMCR 寄存器中写入 TS=00110，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
5. 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），以使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，必须通过在 TIMx_CCMR1 寄存器中写入 OC1M=111，来使能 PWM 模式 2。可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存

器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中， CC1P 的值为 “0”。

在本例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此必须向 TIMx_CR1 寄存器的 OPM 位写入 1，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置 “0” 时，即选择重复模式。

特殊情况：OCx 快速使能：

在单脉冲模式下，TIx 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIMx_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxRef（和 OCx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

24.3.21 可再触发单脉冲模式 (OPM)

该模式允许计数器可以在一个激励信号的触发下启动，并且能产生长度可编程的脉冲，但与如*第 24.3.20 节*所述的不可再触发单脉冲模式间存在以下差别：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

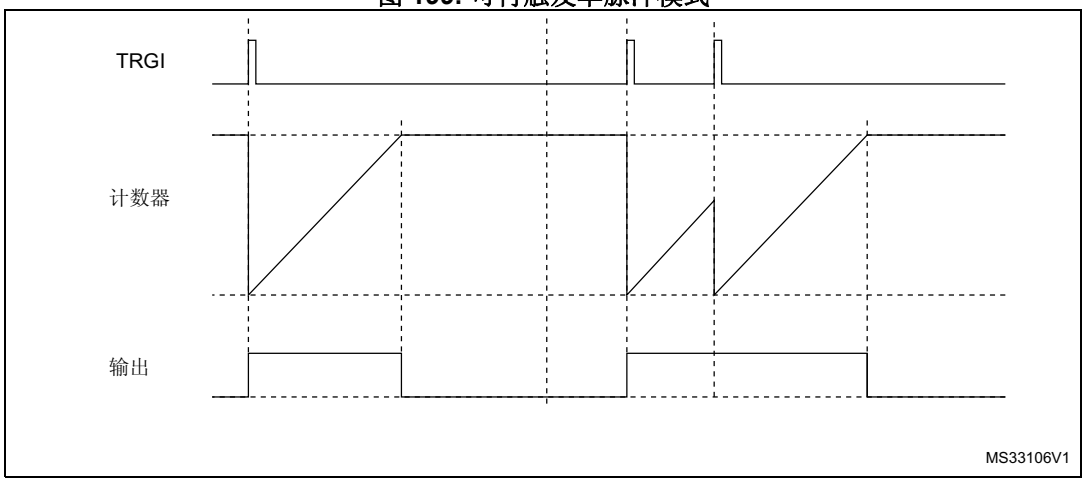
定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] = “1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为 “1000” 或 “1001”。

定时器配置为递增计数模式时，相应的 CCRx 必须置 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCRx 必须高于或等于 ARR。

注： *出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 199. 可再触发单脉冲模式



24.3.22 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 TIMx_SMCR 寄存器中写入 SMS=“001”；如果计数器仅在 TI1 边沿处计数，写入 SMS=“010”；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS=“011”。

通过编程 TIMx_CCER 寄存器的 CC1P 和 CC2P 位，选择 TI1 和 TI2 极性。如果需要，还可对输入滤波器进行编程。CC1NP 和 CC2NP 必须保持低电平。

TI1 和 TI2 两个输入用于连接正交编码器。请参见*表 153*。如果使能计数器（在 TIMx_CR1 寄存器的 CEN 位中写入“1”），则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。TI1FP1 和 TI2FP2 是 TI1 和 TI2 进行输入滤波和极性选择后的信号，如果不进行滤波和反相，则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1 或 TI2 边沿处计数，还是同时在 TI1 和 TI2 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIMx_ARR。同样，捕获、比较、重复计数器和触发输出功能继续正常工作。编码器模式和外部时钟模式 2 不兼容，因此不能同时选择。

注： *使能编码器模式时，预分频器必须设置为零*

在此模式下，计数器会根据正交编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI1 和 TI2 不同时切换）。

表 153. 计数方向与编码器信号的关系

有效边沿	相反信号的电平（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 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

正交编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码器输出可与外部中断输入相连，用以触发计数器复位。

图 200 给出一个计数器工作的实例，说明了计数信号的产生和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S=“01”（TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）。
- CC2S=“01”（TIMx_CCMR2 寄存器，TI1FP2 映射到 TI2 上）。
- CC1P=“0”，CC1NP=“0”（TIMx_CCER 寄存器，TI1FP1 未反相，TI1FP1=TI1）。
- CC2P=“0”，CC2NP=“0”（TIMx_CCER 寄存器，TI1FP2 未反相，TI1FP2= TI2）。
- SMS=“011”（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）。
- CEN=“1”（TIMx_CR1 寄存器，使能计数器）。

图 200. 编码器接口模式下的计数器工作示例

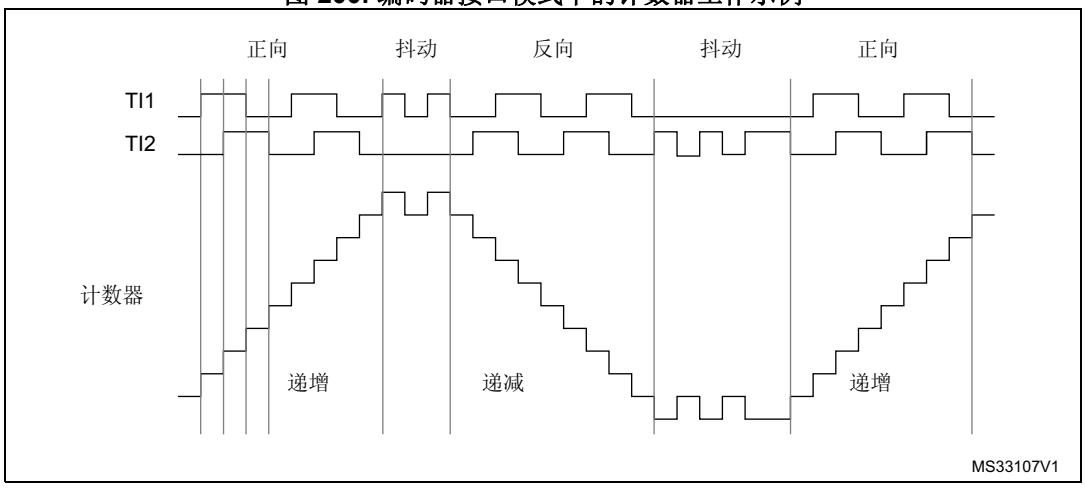
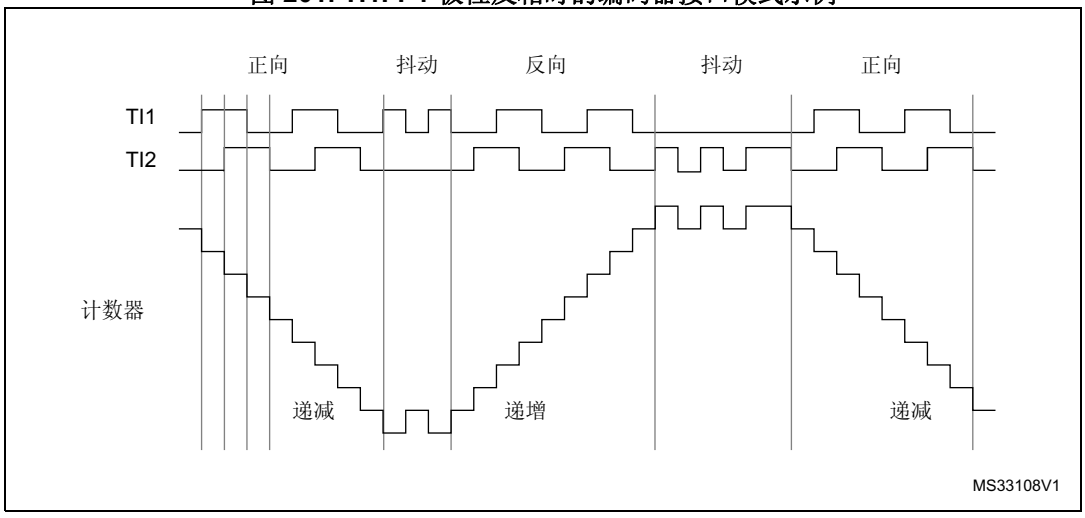


图 201 举例说明 TI1FP1 极性反相时计数器的行为（除 CC1P=“1”外，其他配置与上例相同）。

图 201. TI1FP1 极性反相时的编码器接口模式示例



定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）。还可以通过 DMA 请求读取计数器值。

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 (UIF) 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化角速度的计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

在 32 位定时器实现中，当 IUFREMAP 位置 1 时，计数器的位 31 在读访问时由 UIFCPY 标志覆盖（计数器的最高有效位只能在写模式下访问）。

24.3.23 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时器计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

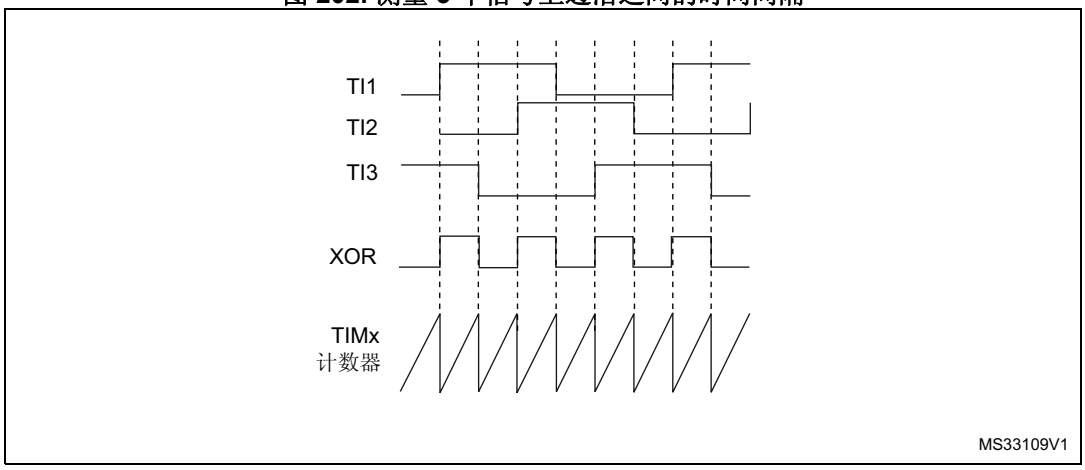
UIF 和 UIFCPY 标志使能之间没有延迟。

24.3.24 定时器输入异或功能

通过 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 到 TIMx_CH3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。这样便于测量两个输入信号上边沿之间的间隔（如下面的*图 202* 所示）。

图 202. 测量 3 个信号上边沿之间的时间间隔



24.3.25 连接霍尔传感器

可通过用于产生电机驱动 PWM 信号的高级控制定时器 (TIM1) 以及*图 203* 中称为“接口定时器”的另一个定时器 TIMx (TIM2)，实现与霍尔传感器的连接。3 个定时器输入引脚（CC1、CC2 和 CC3）通过异或门连接到 TI1 输入通道（通过将 TIMx_CR2 寄存器中的 TI1S 位置 1 来选择），并由“接口定时器”进行捕获。

从模式控制器配置为复位模式；从输入为 TI1F_ED。这样，每当 3 个输入中有一个输入发生切换时，计数器会从 0 开始重新计数。这样将产生由霍尔输入的任何变化而触发的时基。

在“接口定时器”上，捕获/比较通道 1 配置为捕获模式，捕获信号为 TRC（请参见*第 667 页的图 176：捕获/比较通道（示例：通道 1 输入阶段）*）。捕获值对应于输入上两次变化的间隔时间，可提供与电机转速相关的信息。

“接口定时器”可用于在输出模式下产生脉冲，以通过触发 COM 事件更改高级控制定时器 (TIM1) 各个通道的配置。TIM1 定时器用于生成电机驱动 PWM 信号。为此，必须对接口定时器通道进行编程，以便在编程的延迟过后产生正脉冲（在输出比较或 PWM 模式中）。该脉冲通过 TRGO 输出发送到高级控制定时器 (TIM1)。

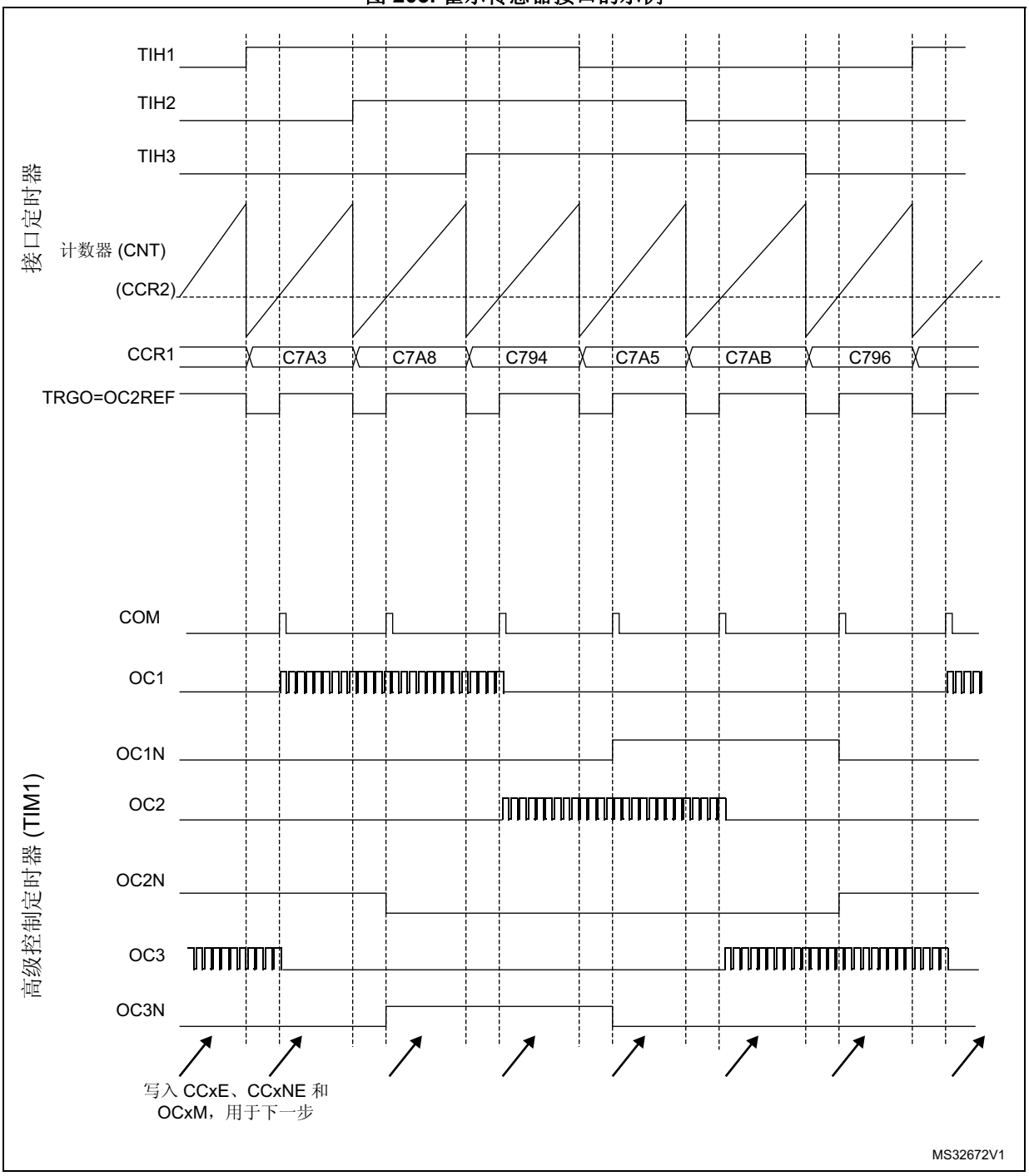
示例：霍尔输入与一个 TIMx 定时器相连接，每当霍尔输入发生更改，需要在所编程的延迟过后更改高级控制定时器 TIM1 的 PWM 配置。

- 向 TIMx_CR2 寄存器的 TI1S 位写入“1”，使 3 个定时器输入经过异或运算后进入 TI1 输入通道。
- 时基编程：向 TIMx_ARR 写入其最大值（计数器必须通过 TI1 的变化清零）。设置预分频器，以得到最大计数器周期，该周期长于传感器上两次变化的间隔时间。
- 将通道 1 编程为捕获模式（选择 TRC）：向 TIMx_CCMR1 寄存器的 CC1S 位写入“01”。如果需要，还可以编程数字滤波器。
- 将通道 2 编程为 PWM 2 模式，并具有所需延迟：向 TIMx_CCMR1 寄存器的 OC2M 位写入“111”，CC2S 位写入“00”。
- 选择 OC2REF 作为 TRGO 上的触发输出：向 TIMx_CR2 寄存器的 MMS 位写入“101”。

在高级控制定时器 TIM1 中，必须选择正确的 ITR 输入作为触发输入，定时器编程为可产生 PWM 信号，捕获/比较控制信号进行预装载（TIMx_CR2 寄存器的 CCPC=1），并且 COM 事件由触发输入控制（TIMx_CR2 寄存器中 CCUS=1）。发生 COM 事件后，在 PWM 控制位（CCxE、OCxM）中写入下一步的配置，此操作可在由 OC2REF 上升沿产生的中断子程序中完成。

图 203 为本示例的示意图。

图 203. 霍尔传感器接口的示例



24.3.26 定时器同步

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。它们可在以下几种模式下实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

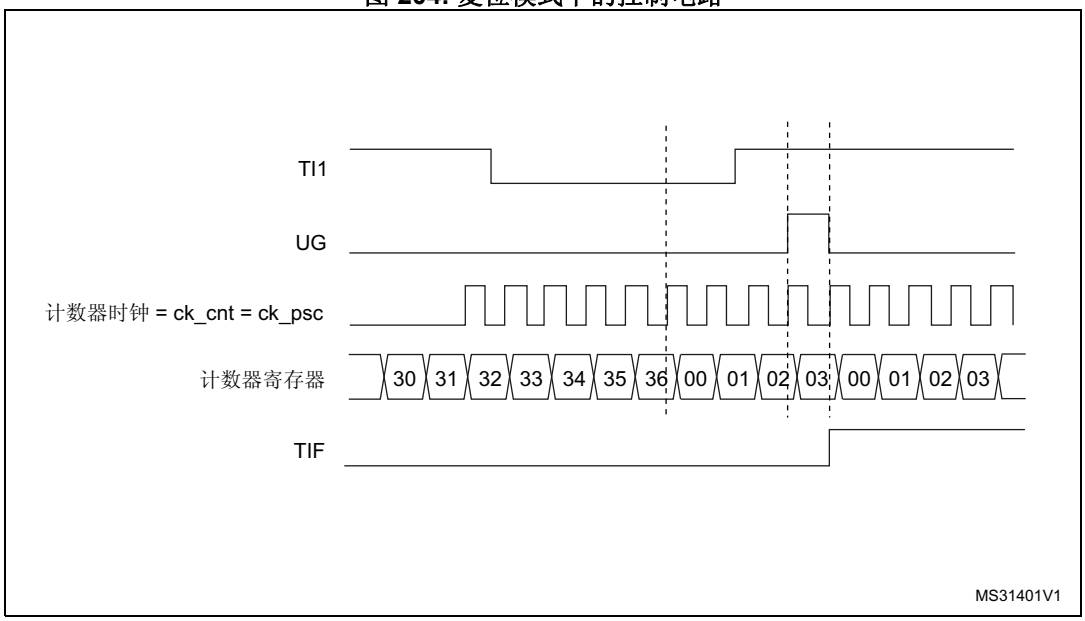
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

- 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP='0'，验证极性（仅检测上升沿）。
- 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 204. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

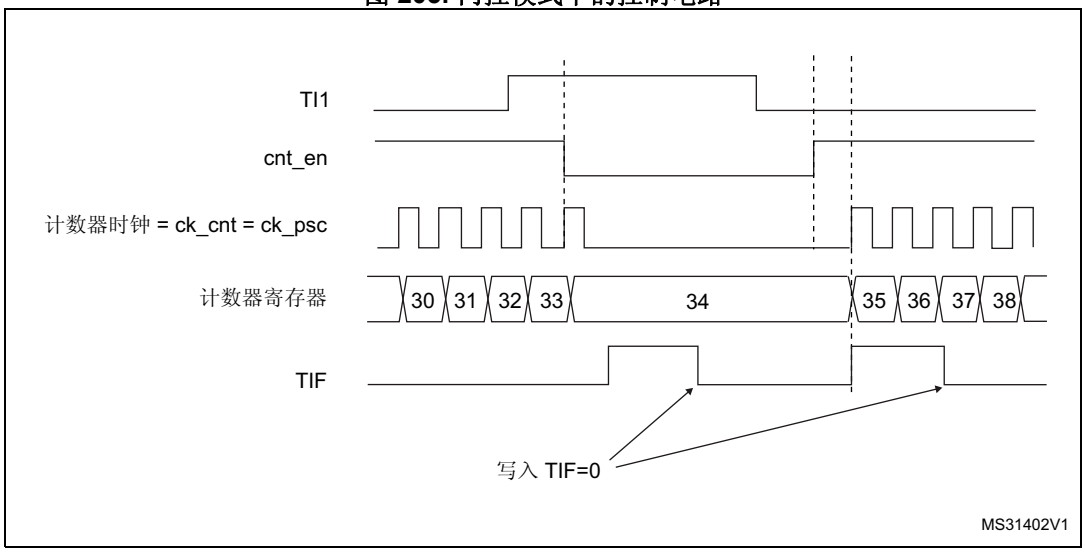
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

- 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1 和 CC1NP=“0”，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 205. 门控模式下的控制电路



从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

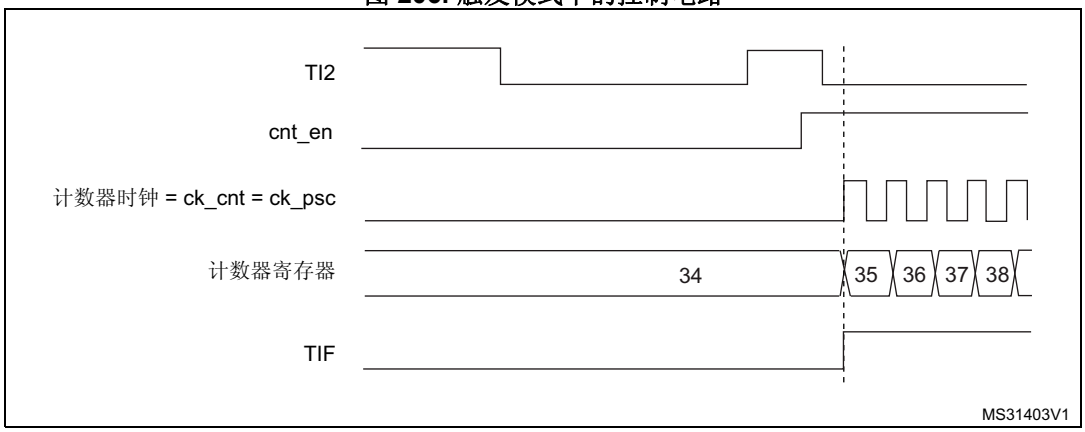
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位配置为只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00110，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 206. 触发模式下的控制电路



从模式：组合复位 + 触发模式

在这种情况下，在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件，并启动计数器。

该模式用于单脉冲模式。

从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

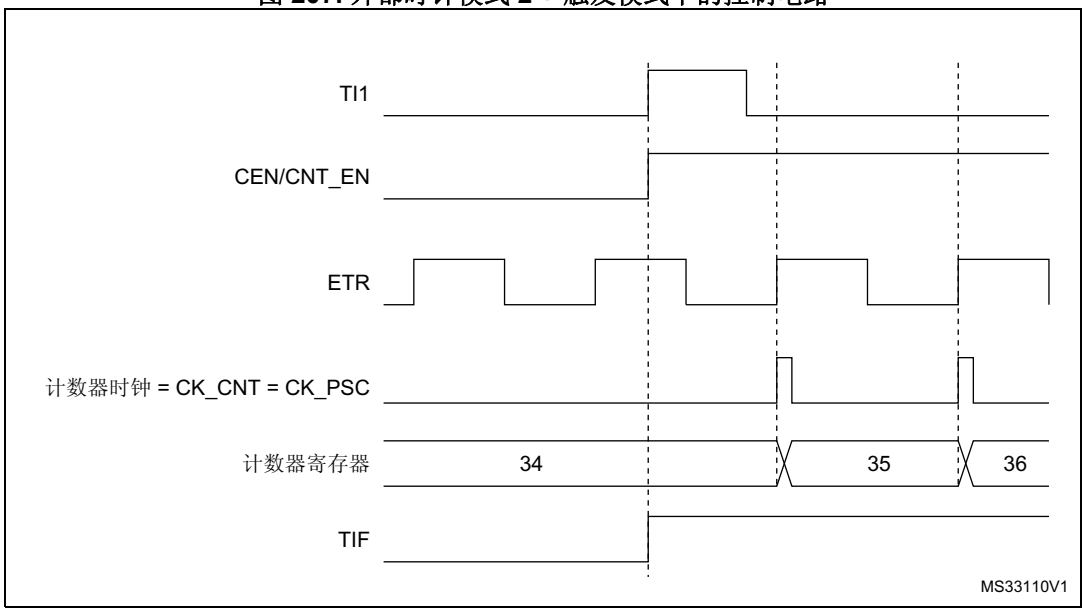
在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

1. 通过对 TIMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
   - ETF=0000：无滤波器。
   - ETPS=00：禁止预分频器。
   - ETP=0：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。
2. 如下配置通道 1，以检测 TI 的上升沿：
   - IC1F=0000：无滤波器。
   - 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
   - TIMx_CCMR1 寄存器中 CC1S=01，只选择输入捕获源。
   - TIMx_CCER 寄存器中 CC1P=0 且 CC1NP=“0”，以确定极性（仅检测上升沿）。
3. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。

ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。

图 207. 外部时钟模式 2 + 触发模式下的控制电路



注：必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

24.3.27 ADC 同步

定时器可通过多种内部信号产生 ADC 触发事件，例如复位、使能或比较事件。也可生成由内部边沿检测器发出的脉冲，例如：

- OC4ref 的上升沿和下降沿
- OC5ref 上的上升沿或 OC6ref 上的下降沿

在重定向到 ADC 的 TRGO2 内部线路上发出触发信号。共有 16 个可能的事件，它们可通过 TIMx_CR2 寄存器中的 MMS2[3:0] 位选择。

第 679 页的图 187 给出了三相电机驱动的应用示例。

注：必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

注：必须先使能 ADC 时钟，才能从主定时器接收事件；从定时器接收触发信号时，不得实时更改 ADC 时钟。

24.3.28 DMA 连续传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的一部分寄存器多次重新编程而无需软件开销，但也可用于定期读取一行中的多个寄存器。

DMA 控制器目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（突发）。每次写入 TIMx_DMAR 寄存器都会重定向到其中一个定时器寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 连续传送长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送，DBL 即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量：

示例：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   - DMA 通道外设地址为 DMAR 寄存器地址。
   - DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   - 要传输的数据量 = 3（参见下文注释）。
   - 禁止循环模式。
2. 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：
   DBL = 3 次传输，DBA = 0xE。
3. 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx。
5. 使能 DMA 通道。

本例适用于每个 CCRx 寄存器只更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。

注： *可以将空值写入保留的寄存器中。*

24.3.29 调试模式

当系统进入调试模式（处理器内核停止）时，TIMx 计数器会根据 DBGMCU 模块中的 DBG_TIM1_STOP 配置位选择继续正常工作或者停止工作。

为了安全起见，当计数器停止 (DBG_TIMx_STOP = 1) 时，输出被禁止（就像 MOE 位被复位一样）。可以将输出强制变为无效状态（OSSI 位 = 1），或者通过 GPIO 控制器（OSSI 位 = 0）来控制输出，通常将其强制为高阻态。

有关详细信息，请参见*第 41.8.7 节：DBGMCU CPU1 APB2 外设冻结寄存器 (DBGMCU_APB2FZR)*。

为了安全起见，当计数器停止 (DBG_TIMx_STOP = 1) 时，输出被禁止（就像 MOE 位被复位一样）。可以将输出强制变为无效状态（OSSI 位 = 1），或者通过 GPIO 控制器（OSSI 位 = 0）来控制输出，以将其强制为高阻态。

24.4 TIM1 寄存器

有关寄存器说明中使用的缩写，请参见相应列表。

24.4.1 TIM1 控制寄存器 1 (TIM1_CR1)

TIM1 control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
				rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)
此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器（ETR、TIx）所使用的死区及采样时钟 (t_{DTS}) 之间的分频比。
00：$t_{DTS}=t_{CK\_INT}$
01：$t_{DTS}=2*t_{CK\_INT}$
10：$t_{DTS}=4*t_{CK\_INT}$
11：保留，不要设置成此值

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:5 **CMS[1:0]**：中心对齐模式选择 (Center-aligned mode selection)
00：边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。
01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。
10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。
11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志都会置 1。
注：　只要计数器处于使能状态 (CEN=1)，就不得从边沿对齐模式切换为中心对齐模式。

位 4 **DIR**：方向 (Direction)
0：计数器递增计数
1：计数器递减计数
注：　当定时器配置为中心对齐模式或编码器模式时，此位为只读状态。

位 3 **OPM**：单脉冲模式 (One pulse mode)
0：计数器在发生更新事件时不会停止计数
1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 **URS**：更新请求源 (Update request source)

此位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会产生更新中断或 DMA 请求。此类事件包括：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 **UDIS**：更新禁止 (Update disable)

此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

然后更新影子寄存器的值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注： 只有事先通过软件将 CEN 位置 1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。

24.4.2 TIM1 控制寄存器 2 (TIM1_CR2)

TIM1 control register 2

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5
								rw	rw	rw	rw		rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 31:24 保留，必须保持复位值。

位 23:20 **MMS2[3:0]**：主模式选择 2 (Master mode selection 2)
这些位可选择将发送到 ADC 以实现同步的信息 (TRGO2)。这些位的组合如下：
0000：**复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO2)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO2 上的信号相比实际复位会有延迟。
0001：**使能**——计数器使能信号 CNT_EN 用作触发输出 (TRGO2)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。当计数器使能信号由触发输入控制时，TRGO2 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。
0010：**更新**——选择更新事件作为触发输出 (TRGO2)。例如，主定时器可用作从定时器的预分频器。
0011：**比较脉冲**——CC1IF 标志置 1 时（即使已为高），只要发生捕获或比较匹配，触发输出 (TRGO2) 都会发送一个正脉冲。
0100：**比较**——OC1REF 信号用作触发输出 (TRGO2)
0101：**比较**——OC2REF 信号用作触发输出 (TRGO2)
0110：**比较**——OC3REF 信号用作触发输出 (TRGO2)
0111：**比较**——OC4REF 信号用作触发输出 (TRGO2)
1000：**比较**——OC5REF 信号用作触发输出 (TRGO2)
1001：**比较**——OC6REF 信号用作触发输出 (TRGO2)
1010：**比较脉冲**——OC4REF 上升沿或下降沿时，TRGO2 上生成脉冲
1011：**比较脉冲**——OC6REF 上升沿或下降沿时，TRGO2 上生成脉冲
1100：**比较脉冲**——OC4REF 或 OC6REF 上升沿时，TRGO2 上生成脉冲
1101：**比较脉冲**——OC4REF 上升沿或 OC6REF 下降沿时，TRGO2 上生成脉冲
1110：**比较脉冲**——OC5REF 或 OC6REF 上升沿时，TRGO2 上生成脉冲
1111：**比较脉冲**——OC5REF 上升沿或 OC6REF 下降沿时，TRGO2 上生成脉冲

注：　必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 19 保留，必须保持复位值。

位 18 **OIS6**：输出空闲状态 6（OC6 输出）(Output Idle state 6 (OC6 output))
请参见 OIS1 位

位 17 保留，必须保持复位值。

位 16 **OIS5**：输出空闲状态 5（OC5 输出）(Output Idle state 5 (OC5 output))
请参见 OIS1 位

位 15 保留，必须保持复位值。

位 14 **OIS4**：输出空闲状态 4（OC4 输出）(Output Idle state 4 (OC4 output))
请参见 OIS1 位

位 13 **OIS3N**：输出空闲状态 3（OC3N 输出）(Output Idle state 3 (OC3N output))
请参见 OIS1N 位

位 12 **OIS3**：输出空闲状态 3（OC3 输出）(Output Idle state 3 (OC3 output))
请参见 OIS1 位

位 11 **OIS2N**：输出空闲状态 2（OC2N 输出）(Output Idle state 2 (OC2N output))
请参见 OIS1N 位

位 10 **OIS2**：输出空闲状态 2（OC2 输出）(Output Idle state 2 (OC2 output))
请参见 OIS1 位

位 9 **OIS1N**：输出空闲状态 1（OC1N 输出）(Output Idle state 1 (OC1N output))
0：当 MOE=0 时，经过死区时间后 OC1N=0
1：当 MOE=0 时，经过死区时间后 OC1N=1

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位即无法修改。

位 8 **OIS1**：输出空闲状态 1（OC1 输出）(Output Idle state 1 (OC1 output))
0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0
1：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=1
注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位即无法修改。

位 7 **TI1S**：TI1 选择 (TI1 selection)
0：TIMx_CH1 引脚连接到 TI1 输入
1：TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合）

位 6:4 **MMS[2:0]**：主模式选择 (Master mode selection)
这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下：
000：**复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。
001：**使能**——计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。
010：**更新**——选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。
011：**比较脉冲**——一旦发生输入捕获或比较匹配事件，当 CC1IF 标志被置 1 时（即使已为高电平），触发输出都会发送一个正脉冲。(TRGO)。
100：**比较**——OC1REF 信号用作触发输出 (TRGO)
101：**比较**——OC2REF 信号用作触发输出 (TRGO)
110：**比较**——OC3REF 信号用作触发输出 (TRGO)
111：**比较**——OC4REF 信号用作触发输出 (TRGO)
注：　必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 3 **CCDS**：捕获/比较 DMA 选择 (Capture/compare DMA selection)
0：发生 CCx 事件时发送 CCx DMA 请求
1：发生更新事件时发送 CCx DMA 请求

位 2 **CCUS**：捕获/比较控制更新选择 (Capture/compare control update selection)
0：如果捕获/比较控制位进行预装载 (CCPC=1)，仅通过将 COMG 位置 1 来对这些位进行更新。
1：如果捕获/比较控制位进行预装载 (CCPC=1)，可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。
注：　此位仅对具有互补输出的通道有效。

位 1 保留，必须保持复位值。

位 0 **CCPC**：捕获/比较预装载控制 (Capture/compare preloaded control)
0：CCxE、CCxNE 和 OCxM 位未进行预装载。
1：CCxE、CCxNE 和 OCxM 位进行了预装载，写入这些位后，仅当发生换向事件 (COM)（COMG 位置 1 或在 TRGI 上检测到上升沿，取决于 CCUS 位）时才会对这些位进行更新。
注：　此位仅对具有互补输出的通道有效。

24.4.3 TIM1 从模式控制寄存器 (TIM1_SMCR)

TIM1 slave mode control register

偏移地址：0x08

复位值：0x0000 0000

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

位 31:22 保留，必须保持复位值。

位 19:17 保留，必须保持复位值。

位 15 **ETP**：外部触发极性 (External trigger polarity)
此位可选择将 ETR 还是 $\overline{\text{ETR}}$ 用于触发操作
0：ETR 未反相，高电平或上升沿有效。
1：ETR 反相，低电平或下降沿有效。

位 14 **ECE**：外部时钟使能 (External clock enable)
此位可使能外部时钟模式 2。
0：禁止外部时钟模式 2。
1：使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。

注： ***1：****将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF（SMS=111 且 TS=00111）具有相同效果。*

2：*外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF（TS 位不得为 00111）。*

3：*如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。*

位 13:12 **ETPS[1:0]**：外部触发预分频器 (External trigger prescaler)
外部触发信号 ETRP 频率不得超过 TIMxCLK 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。
00：预分频器关闭
01：2 分频 ETRP 频率
10：4 分频 ETRP 频率
11：8 分频 ETRP 频率

位 11:8 **ETF[3:0]**：外部触发滤波器 (External trigger filter)

此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器的采样长度。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 7 **MSM**：主/从模式 (Master/slave mode)

0：不执行任何操作。
1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

位 21、20、6、5、4 **TS[4:0]**：触发选择 (Trigger selection)

此位域可选择将要用于同步计数器的触发输入。

00000：内部触发 0 (ITR0)
00001：内部触发 1 (ITR1)
00010：内部触发 2 (ITR2)
00011：内部触发 3 (ITR3)
00100：TI1 边沿检测器 (TI1F_ED)
00101：滤波后的定时器输入 1 (TI1FP1)
00110：滤波后的定时器输入 2 (TI2FP2)
00111：外部触发输入 (ETRF)
其他值：保留

有关各定时器 ITRx 含义的详细信息，请参见*第 709 页的表 154：TIM1 内部触发连接*。

注：　这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

注：　其他位位于同一寄存器的位置 16。

位 3 **OCCS**：OCREF 清零选择 (OCREF clear selection)

此位用于选择 OCREF 清零源。

0：OCREF_CLR_INT 连接到 OCREF_CLR 输入
1：OCREF_CLR_INT 连接到 ETRF

位 16、2、1、0 **SMS[3:0]**：从模式选择 (Slave mode selection)

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

0000：禁止从模式——如果 CEN =“1”，预分频器时钟直接由内部时钟提供。
0001：编码器模式 1——计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。
0010：编码器模式 2——计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。
0011：编码器模式 3——计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个输入的电平。
0100：复位模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。
0101：门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。
0110：触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。
0111：外部时钟模式 1——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。
1000：组合复位 + 触发模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。
1000 以上的代码：保留。

注：　如果将 TI1F_ED 选作触发输入 (TS=00100)，则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

注：　必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

表 154. TIM1 内部触发连接

从 TIM	ITR0 (TS = 00000)	ITR1 (TS = 00001)	ITR2 (TS = 00010)	ITR3 (TS = 00011)
TIM1	-	TIM2	-	TIM17 OC1

24.4.4 TIM1 DMA/中断使能寄存器 (TIM1_DIER)

TIM1 DMA/interrupt enable register

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
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 保留，必须保持复位值。

位 14 **TDE**：触发 DMA 请求使能 (Trigger DMA request enable)
0：禁止触发 DMA 请求
1：使能触发 DMA 请求

位 13 **COMDE**：COM DMA 请求使能 (COM DMA request enable)
0：禁止 COM DMA 请求
1：使能 COM DMA 请求

位 12 **CC4DE**：捕获/比较 4 DMA 请求使能 (Capture/Compare 4 DMA request enable)
0：禁止 CC4 DMA 请求
1：使能 CC4 DMA 请求

位 11 **CC3DE**：捕获/比较 3 DMA 请求使能 (Capture/Compare 3 DMA request enable)
0：禁止 CC3 DMA 请求
1：使能 CC3 DMA 请求

位 10 **CC2DE**：捕获/比较 2 DMA 请求使能 (Capture/Compare 2 DMA request enable)
0：禁止 CC2 DMA 请求
1：使能 CC2 DMA 请求

位 9 **CC1DE**：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)
0：禁止 CC1 DMA 请求
1：使能 CC1 DMA 请求

位 8 **UDE**：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

位 7 **BIE**：刹车中断使能 (Break interrupt enable)
0：禁止刹车中断
1：使能刹车中断

位 6 **TIE**：触发中断使能 (Trigger interrupt enable)
0：禁止触发中断
1：使能触发中断

位 5 **COMIE**：COM 中断使能 (COM interrupt enable)
0：禁止 COM 中断
1：使能 COM 中断

位 4 **CC4IE**：捕获/比较 4 中断使能 (Capture/Compare 4 interrupt enable)
0：禁止 CC4 中断
1：使能 CC4 中断

位 3 **CC3IE**：捕获/比较 3 中断使能 (Capture/Compare 3 interrupt enable)
0：禁止 CC3 中断
1：使能 CC3 中断

位 2 **CC2IE**：捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable)
0：禁止 CC2 中断
1：使能 CC2 中断

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断
1：使能 CC1 中断

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断
1：使能更新中断

24.4.5 TIM1 状态寄存器 (TIM1_SR)

TIM1 status register

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6IF	CC5IF
														rc_w0	rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 31:18 保留，必须保持复位值。

位 17 **CC6IF**：比较 6 中断标志 (Compare 6 interrupt flag)
请参见 CC1IF 说明（注意：通道 6 只能配置为输出）

位 16 **CC5IF**：比较 5 中断标志 (Compare 5 interrupt flag)
请参见 CC1IF 说明（注意：通道 5 只能配置为输出）

位 15:14 保留，必须保持复位值。

位 13 **SBIF**：系统刹车中断标志 (System Break interrupt flag)
只要系统刹车输入变为有效状态，此标志便由硬件置 1。系统刹车输入无效后可通过软件对其清零。
此标志必须复位以使 PWM 重新开始工作。
0：未发生刹车事件。
1：在系统刹车输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

位 12 **CC4OF**：捕获/比较 4 过捕获标志 (Capture/Compare 4 overcapture flag)
请参见 CC1OF 说明

位 11 **CC3OF**：捕获/比较 3 过捕获标志 (Capture/Compare 3 overcapture flag)
请参见 CC1OF 说明

位 10 **CC2OF**：捕获/比较 2 过捕获标志 (Capture/compare 2 overcapture flag)
请参见 CC1OF 说明

位 9 **CC1OF**：捕获/比较 1 过捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将此位清零。
0：未检测到过捕获。
1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8 **B2IF**：刹车 2 中断标志 (Break 2 interrupt flag)
只要刹车 2 输入变为有效状态，此标志便由硬件置 1。刹车 2 输入无效后可通过软件对其清零。
0：未发生刹车事件。
1：在刹车 2 输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

位 7 **BIF**：刹车中断标志 (Break interrupt flag)
只要刹车输入变为有效状态，此标志便由硬件置 1。刹车输入无效后可通过软件对其清零。
0：未发生刹车事件。
1：在刹车输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

位 6 **TIF**：触发中断标志 (Trigger interrupt flag)

在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。

0：未发生触发事件。

1：触发中断挂起。

位 5 **COMIF**：COM 中断标志 (COM interrupt flag)

此标志在发生 COM 事件时（捕获/比较控制位 CCxE、CCxNE 和 OCxM 已更新时）由硬件置 1。但需要通过软件清零。

0：未发生 COM 事件。

1：COM 中断挂起。

位 4 **CC4IF**：捕获/比较 4 中断标志 (Capture/Compare 4 interrupt flag)

请参见 CC1IF 说明

位 3 **CC3IF**：捕获/比较 3 中断标志 (Capture/Compare 3 interrupt flag)

请参见 CC1IF 说明

位 2 **CC2IF**：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)

请参见 CC1IF 说明

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/Compare 1 interrupt flag)

如果通道 CC1 配置为输出：当计数器与比较值匹配时，此标志由硬件置 1，中心对齐模式下除外（请参见 TIMx_CR1 寄存器中的 CMS 位说明）。但需要通过软件清零。

0：不匹配。

1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高。

如果通道 CC1 配置为输入：此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将此位清零。

0：未发生输入捕获事件

1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）

位 0 **UIF**：更新中断标志 (Update interrupt flag)

此位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。此位在以下情况下更新寄存器时由硬件置 1：

– TIMx_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器 = 0 时更新）。

– TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

– TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（请参见 *第 24.4.3 节：TIM1 从模式控制寄存器 (TIM1_SMCR)*）。

24.4.6 TIM1 事件生成寄存器 (TIM1_EGR)

TIM1 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
							w	w	w	w	w	w	w	w	w

位 15:9 保留，必须保持复位值。

位 8 **B2G**：刹车 2 生成 (Break 2 generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作。
1：生成刹车 2 事件。MOE 位清零且 B2IF 标志置 1。使能后可发生相关中断。

位 7 **BG**：刹车生成 (Break generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作。
1：生成刹车事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 6 **TG**：触发生成 (Trigger generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作。
1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 5 **COMG**：捕获/比较控制更新生成 (Capture/Compare control update generation)
此位可通过软件置 1，并由硬件自动清零。
0：不执行任何操作。
1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位。
注： 此位仅对具有互补输出的通道有效。

位 4 **CC4G**：捕获/比较 4 生成 (Capture/Compare 4 generation)
请参见 CC1G 说明

位 3 **CC3G**：捕获/比较 3 生成 (Capture/Compare 3 generation)
请参见 CC1G 说明

位 2 **CC2G**：捕获/比较 2 生成 (Capture/compare 2 generation)
请参见 CC1G 说明

位 1 **CC1G**：捕获/比较 1 生成 (Capture/Compare 1 generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作
1：通道 1 上生成捕获/比较事件：
如果通道 CC1 配置为输出：
使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。
如果通道 CC1 配置为输入：
TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)
此位可通过软件置 1，并由硬件自动清零。
0：不执行任何操作
1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (TIMx_ARR)。

24.4.7 TIM1 捕获/比较模式寄存器 1 [复用] (TIM1_CCMR1)

TIM1 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获模式和输出比较模式下的功能均不同。可独立组合两种模式（例如，输入捕获模式下的通道 1 和输出比较模式下的通道 2）。

输入捕获模式：

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **IC2F[3:0]**：输入捕获 2 滤波器 (Input capture 2 filter)
请参见 IC1F[3:0] 说明。

位 11:10 **IC2PSC[1:0]**：输入捕获 2 预分频器 (Input capture 2 prescaler)
请参见 OC1PSC[1:0] 说明。

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC2 通道配置为输出
01：CC2 通道配置为输入，IC2 映射到 TI2 上
10：CC2 通道配置为输入，IC2 映射到 TI1 上
11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器的采样长度。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC[1:0]**：输入捕获 1 预分频器 (Input capture 1 prescaler)

此位域定义 CC1 输入 (IC1) 的预分频比。只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：CC1 通道配置为输入，IC1 映射到 TI2 上
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

24.4.8 TIM1 捕获/比较模式寄存器 1 [复用] (TIM1_CCMR1)

TIM1 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获模式和输出比较模式下的功能均不同。可独立组合两种模式（例如，输入捕获模式下的通道 1 和输出比较模式下的通道 2）。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		OC1 CE	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24 **OC2M[3]**：输出比较 2 模式——位 3 (Output Compare 2 mode - bit 3)
请参见 OC2M 说明——位 14:12

位 23:17 保留，必须保持复位值。

位 16 **OC1M[3]**：输出比较 1 模式——位 3 (Output Compare 1 mode - bit 3)
请参见 OC1M 说明——位 6:4

位 15 **OC2CE**：输出比较 2 清零使能 (Output Compare 2 clear enable)
请参见 OC1CE 说明。

位 14:12 **OC2M[2:0]**：输出比较 2 模式 (Output Compare 2 mode)
请参见 OC1M[2:0] 说明。

位 11 **OC2PE**：输出比较 2 预装载使能 (Output Compare 2 preload enable)
请参见 OC1PE 说明。

位 10 **OC2FE**：输出比较 2 快速使能 (Output Compare 2 fast enable)
请参见 OC1FE 说明。

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC2 通道配置为输出
01：CC2 通道配置为输入，IC2 映射到 TI2 上
10：CC2 通道配置为输入，IC2 映射到 TI1 上
11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。

位 7 **OC1CE**：输出比较 1 清零使能 (Output Compare 1 clear enable)
0：OC1Ref 不受 ocref_clr_int 信号影响
1：ocref_clr_int 信号（OCREF_CLR 输入或 ETRF 输入）上检测到高电平时，OC1Ref 立即清零

位 6:4 **OC1M[2:0]**：输出比较 1 模式 (Output Compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为无效状态（OC1REF=“0”），否则为有效状态（OC1REF=“1”）。

0111：PWM 模式 2——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。

1000：可再触发 OPM 模式 1——在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。

1001：可再触发 OPM 模式 2——在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。

1010：保留。

1011：保留。

1100：组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101：组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110：不对称 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

1111：不对称 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。

注：　在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

注：　此位域将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成 COM 事件时，OC1M 有效位才会从预装载位获取新值。

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。

*注：　**1：**只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。*

*　　**2：**只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其他情况下则无法保证该行为。*

位 2 **OC1FE**：输出比较 1 快速使能 (Output Compare 1 fast enable)

此位用于加快触发输入事件对 CC 输出的影响。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：CC1 通道配置为输入，IC1 映射到 TI2 上

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

24.4.9 TIM1 捕获/比较模式寄存器 2 [复用] (TIM1_CCMR2)

TIM1 capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获模式和输出比较模式下的功能均不同。可独立组合两种模式（例如，输入捕获模式下的通道 1 和输出比较模式下的通道 2）。

输入捕获模式：

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **IC4F[3:0]**：输入捕获 4 滤波器 (Input capture 4 filter)

请参见 IC1F[3:0] 说明。

位 11:10 **IC4PSC[1:0]**：输入捕获 4 预分频器 (Input capture 4 prescaler)

请参见 IC1PSC[1:0] 说明。

位 9:8 **CC4S[1:0]**：捕获/比较 4 选择 (Capture/Compare 4 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC4 通道配置为输出

01：CC4 通道配置为输入，IC4 映射到 TI4 上

10：CC4 通道配置为输入，IC4 映射到 TI3 上

11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC4E = “0”），才可向 CC4S 位写入数据。

位 7:4 **IC3F[3:0]**：输入捕获 3 滤波器 (Input capture 3 filter)
请参见 IC1F[3:0] 说明。

位 3:2 **IC3PSC[1:0]**：输入捕获 3 预分频器 (Input capture 3 prescaler)
请参见 IC1PSC[1:0] 说明。

位 1:0 **CC3S[1:0]**：捕获/比较 3 选择 (Capture/compare 3 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC3E = “0”），才可向 CC3S 位写入数据。

24.4.10 TIM1 捕获/比较模式寄存器 2 [复用] (TIM1_CCMR2)

TIM1 capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入捕获模式和输出比较模式下的功能均不同。可独立组合两种模式（例如，输入捕获模式下的通道 1 和输出比较模式下的通道 2）。

输出比较模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4 CE	OC4M[2:0]			OC4 PE	OC4 FE	CC4S[1:0]		OC3 CE	OC3M[2:0]			OC3 PE	OC3 FE	CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24 **OC4M[3]**：输出比较 4 模式——位 3 (Output Compare 4 mode - bit 3)
请参见 OC1M 说明。

位 23:17 保留，必须保持复位值。

位 16 **OC3M[3]**：输出比较 3 模式——位 3 (Output Compare 3 mode - bit 3)
请参见 OC1M 说明。

位 15 **OC4CE**：输出比较 4 清零使能 (Output compare 4 clear enable)
请参见 OC1CE 说明。

位 14:12 **OC4M[2:0]**：输出比较 4 模式 (Output compare 4 mode)
请参见 OC4M 说明。

位 11 **OC4PE**：输出比较 4 预装载使能 (Output compare 4 preload enable)
请参见 OC1PE 说明。

位 10 **OC4FE**：输出比较 4 快速使能 (Output compare 4 fast enable)
请参见 OC1FE 说明。

位 9:8 **CC4S[1:0]**：捕获/比较 4 选择 (Capture/Compare 4 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC4E = “0”），才可向 CC4S 位写入数据。

位 7 **OC3CE**：输出比较 3 清零使能 (Output compare 3 clear enable)
请参见 OC1CE 说明。

位 6:4 **OC3M[2:0]**：输出比较 3 模式 (Output compare 3 mode)
请参见 OC1M 说明。

位 3 **OC3PE**：输出比较 3 预装载使能 (Output compare 3 preload enable)
请参见 OC1PE 说明。

位 2 **OC3FE**：输出比较 3 快速使能 (Output compare 3 fast enable)
请参见 OC1FE 说明。

位 1:0 **CC3S[1:0]**：捕获/比较 3 选择 (Capture/Compare 3 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC3E = “0”），才可向 CC3S 位写入数据。

24.4.11 TIM1 捕获/比较使能寄存器 (TIM1_CCER)

TIM1 capture/compare enable register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E
										rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21 **CC6P**：捕获/比较 6 输出极性 (Capture/Compare 6 output polarity)
请参见 CC1P 说明

位 20 **CC6E**：捕获/比较 6 输出使能 (Capture/Compare 6 output enable)
请参见 CC1E 说明

位 19:18 保留，必须保持复位值。

位 17 **CC5P**：捕获/比较 5 输出极性 (Capture/Compare 5 output polarity)
请参见 CC1P 说明

位 16 **CC5E**：捕获/比较 5 输出使能 (Capture/Compare 5 output enable)
请参见 CC1E 说明

位 15 **CC4NP**：捕获/比较 4 互补输出极性 (Capture/Compare 4 complementary output polarity)
请参见 CC1NP 说明

位 14 保留，必须保持复位值。

位 13 **CC4P**：捕获/比较 4 输出极性 (Capture/Compare 4 output Polarity)
请参见 CC1P 说明

位 12 **CC4E**：捕获/比较 4 输出使能 (Capture/Compare 4 output enable)
请参见 CC1E 说明

位 11 **CC3NP**：捕获/比较 3 互补输出极性 (Capture/Compare 3 complementary output polarity)
请参见 CC1NP 说明

位 10 **CC3NE**：捕获/比较 3 互补输出使能 (Capture/Compare 3 complementary output enable)
请参见 CC1NE 说明

位 9 **CC3P**：捕获/比较 3 输出极性 (Capture/Compare 3 output polarity)
请参见 CC1P 说明

位 8 **CC3E**：捕获/比较 3 输出使能 (Capture/Compare 3 output enable)
请参见 CC1E 说明

位 7 **CC2NP**：捕获/比较 2 互补输出极性 (Capture/Compare 2 complementary output polarity)
请参见 CC1NP 说明

位 6 **CC2NE**：捕获/比较 2 互补输出使能 (Capture/Compare 2 complementary output enable)
请参见 CC1NE 说明

位 5 **CC2P**：捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)
请参见 CC1P 说明

位 4 **CC2E**：捕获/比较 2 输出使能 (Capture/Compare 2 output enable)
请参见 CC1E 说明

位 3 **CC1NP**：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)
CC1 通道配置为输出：
0：OC1N 高电平有效。
1：OC1N 低电平有效。
CC1 通道配置为输入：
此位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 且 CC1S=“00”（通道配置为输出），此位立即变为不可写状态。
此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。

位 2 **CC1NE**：捕获/比较 1 互补输出使能 (Capture/Compare 1 complementary output enable)

0：关闭——OC1N 未激活。OC1N 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的函数。

1：开启——在相应输出引脚上输出 OC1N 信号，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位。

此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NE 有效位才会从预装载位获取新值。

位 1 **CC1P**：捕获/比较 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：CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。

00：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。

01：反相/下降沿触发。电路对 TIxFP1 下升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。

10：保留，不使用此配置。

11：非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，此位立即变为不可写状态。

此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。

位 0 **CC1E**：捕获/比较 1 输出使能 (Capture/Compare 1 output enable)

CC1 通道配置为输出：

0：关闭——OC1 未激活。OC1 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的函数。

1：开启——OC1 信号输出到相应的输出引脚上，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位。

CC1 通道配置为输入：此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1 (TIMx_CCR1) 中。

0：禁止捕获。

1：使能捕获。

此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1E 有效位才会从预装载位获取新值。

表 155. 具有刹车功能的互补通道 OCx 和 OCxN 的输出控制位

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	X	0	0	禁止输出（不由定时器驱动：高阻态） OCx=0、OCxN=0	
		0	0	1	禁止输出（不由定时器驱动：高阻态） OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出（不由定时器驱动：高阻态） OCxN=0
		X	1	1	OCREF + 极性 + 死区	OCREF 互补项（对 OCREF 进行“非”运算）+ 极性 + 死区
		1	0	1	关闭状态（输出使能为无效状态） OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		1	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	关闭状态（输出使能为无效状态） OCxN=CCxNP
0	0	X	X	X	禁止输出（不再由定时器驱动）。输出状态由 GPIO 控制器定义，可以是高电平、低电平或高阻态。	
	1		0	0		
			0	1	关闭状态（输出使能为无效状态） 异步：OCx=CCxP、OCxN=CCxNP （如果触发 BRK 或 BRK2）。 随后（仅当触发 BRK 时才有效），如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，假定 OISx 和 OISxN 并没有都设置成 OCx 及 OCxN 的有效电平（否则在半桥配置下驱动开关时可能导致短路）。 **注意**：BRK2 只能在 OSSI = OSSR = 1 时使用。	
			1	0		
			1	1		

1. 如果一个通道的两个输出均未使用（由 GPIO 接管控制控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注： *与互补通道 OCx 和 OCxN 相连的外部 I/O 引脚的状态取决于通道 OCx 和 OCxN 的状态以及 GPIO 寄存器。*

24.4.12 TIM1 计数器 (TIM1_CNT)

TIM1 counter

偏移地址：0x24

复位值：0x0000 0000

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

位 31 **UIFCPY**：UIF 副本 (UIF copy)

此位是 TIMx_ISR 寄存器中 UIF 位的只读副本。如果 TIMxCR1 中的 UIFREMAP 位复位，则位 31 保留，读为 0。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

24.4.13 TIM1 预分频器 (TIM1_PSC)

TIM1 prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 **PSC[15:0]**：预分频值 (Prescaler value)

计数器时钟频率 (CK_CNT) 等于 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。

24.4.14 TIM1 自动重载寄存器 (TIM1_ARR)

TIM1 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的更多详细信息，请参见*第 647 页的第 24.3.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

24.4.15 TIM1 重复计数器寄存器 (TIM1_RCR)

TIM1 repetition counter register

偏移地址：0x30

复位值：0x0000

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

位 15:0 **REP[15:0]**：重复计数器值 (Repetition Counter value)

使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向活动寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。

与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，无论向 TIMx_RCR 寄存器写入何值都无影响。

这意味着 PWM 模式下 (REP+1) 相当于：

边沿对齐模式下的 PWM 周期数。

中心对齐模式下的 PWM 半周期数。

24.4.16 TIM1 捕获/比较寄存器 1 (TIM1_CCR1)

TIM1 capture/compare register 1

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR1[15:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：CCR1 为要装载到有效捕获/比较 1 寄存器的值（预装载值）。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 1）。

实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。

如果通道 CC1 配置为输入：CR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。只能读取 TIMx_CCR1 寄存器，无法对其进行编程。

24.4.17 TIM1 捕获/比较寄存器 2 (TIM1_CCR2)

TIM1 capture/compare register 2

偏移地址：0x38

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR2[15:0]**：捕获/比较 2 值 (Capture/Compare 2 value)

如果通道 CC2 配置为输出：CCR2 为要装载到有效捕获/比较 2 寄存器的值（预装载值）。
如果没有通过 TIMx_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 2）。
有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。
如果通道 CC2 配置为输入：CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。只能读取 TIMx_CCR2 寄存器，无法对其进行编程。

24.4.18 TIM1 捕获/比较寄存器 3 (TIM1_CCR3)

TIM1 capture/compare register 3

偏移地址：0x3C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR3[15:0]**：捕获/比较值 (Capture/Compare value)

如果通道 CC3 配置为输出：CCR3 为要装载到有效捕获/比较 3 寄存器的值（预装载值）。
如果没有通过 TIMx_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 3）。
有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。

如果通道 CC3 配置为输入：CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。只能读取 TIMx_CCR3 寄存器，无法对其进行编程。

24.4.19 TIM1 捕获/比较寄存器 4 (TIM1_CCR4)

TIM1 capture/compare register 4

偏移地址：0x40

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR4[15:0]**：捕获/比较值 (Capture/Compare value)

如果通道 CC4 配置为输出：CCR4 为要装载到有效捕获/比较 4 寄存器的值（预装载值）。如果没有通过 TIMx_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 4）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。

如果通道 CC4 配置为输入：CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。只能读取 TIMx_CCR4 寄存器，无法对其进行编程。

24.4.20 TIM1 刹车和死区寄存器 (TIM1_BDTR)

TIM1 break and dead-time register

偏移地址：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	BK2BID	BKBID	BK2 DSRM	BK DSRM	BK2P	BK2E	BK2F[3:0]				BKF[3:0]			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： *由于可以根据 LOCK 配置锁定位 BK2BID、BKBID、BK2DSRM、BKDSRM、BK2P、BK2E、BK2F[3:0]、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 的写操作，因此必须在第一次对 TIMx_BDTR 寄存器执行写访问时对这些位进行配置。*

位 31:30 保留，必须保持复位值。

位 29 **BK2BID**：刹车 2 双向 (Break2 bidirectional)

请参见 BKBID 说明

位 28 **BKBID**：刹车双向 (Break Bidirectional)

0：刹车输入 BRK 为输入模式

1：刹车输入 BRK 为双向模式

在双向模式下（BKBID 位置 1），刹车输入配置为输入模式和开漏输出模式。任何激活的刹车事件都将使刹车输入上呈逻辑低电平，以向外部器件指示发生了内部刹车事件。

注： *只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。*

注： *对此位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 27 **BK2DSRM**：刹车 2 解除 (Break2 Disarm)
请参见 BKDSRM 说明

位 26 **BKDSRM**：刹车解除 (Break Disarm)
0：启动刹车输入 BRK
1：解除刹车输入 BRK
当刹车源激活后，此位由硬件清零。
必须通过软件将 BKDSRM 位置 1 以释放双向输出控制（开漏输出处于高阻态），然后不断轮询此位，直到其由硬件复位，指示故障条件已消失。
注： 对此位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 25 **BK2P**：刹车 2 极性 (Break 2 polarity)
0：刹车输入 BRK2 为低电平有效
1：刹车输入 BRK2 为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。
注： 对此位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 24 **BK2E**：刹车 2 使能 (Break 2 enable)
注： 必须只在 OSSR = OSSI = 1 时使用。
注： 编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。
注： 对此位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 23:20 **BK2F[3:0]**：刹车 2 滤波器 (Break 2 filter)
此位域可定义 BRK2 输入的采样频率和适用于 BRK2 的数字滤波器的采样长度。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：
0000：无滤波器，BRK2 异步工作
0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8
注： 编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

位 19:16 **BKF[3:0]**：刹车滤波器 (Break filter)

此位域可定义 BRK 输入的采样频率和适用于 BRK 的数字滤波器的采样长度。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，BRK 异步工作
0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

位 15 **MOE**：主输出使能 (Main output enable)

只要刹车输入（BRK 或 BRK2）为有效状态，此位便由硬件异步清零。此位由软件置 1，也可根据 AOE 位状态自动置 1。此位仅对配置为输出的通道有效。

0：响应刹车事件（2 个）。禁止 OC 和 OCN 输出。
响应刹车事件或向 MOE 写入 0 时：OC 和 OCN 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。
1：如果 OC 和 OCN 输出的相应使能位（TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。
有关详细信息，请参见 OC/OCN 使能说明（*第 24.4.11 节：TIM1 捕获/比较使能寄存器 (TIM1_CCER)*）。

位 14 **AOE**：自动输出使能 (Automatic output enable)

0：MOE 只能由软件置 1
1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1（如果刹车输入 BRK 和 BRK2 均无效）

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 13 **BKP**：刹车极性 (Break polarity)

0：刹车输入 BRK 为低电平有效
1：刹车输入 BRK 为高电平有效

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。
注：　对此位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 12 **BKE**：刹车使能 (Break enable)

此位可使能完整的刹车保护（包括连接到 bk_acth 的所有源和相应的 BKIN 源，如*图 191：刹车和刹车 2 电路概述*所示）。

0：禁止刹车功能
1：使能刹车功能

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。
注：　对此位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 11 **OSSR**：运行模式下的关闭状态选择 (Off-state selection for Run mode)

此位在 MOE=1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。

有关详细信息，请参见 OC/OCN 使能说明（*第 24.4.11 节：TIM1 捕获/比较使能寄存器 (TIM1_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管）。

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平（输出仍由定时器控制）。

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。

位 10 **OSSI**：空闲模式下的关闭状态选择 (Off-state selection for Idle mode)

当由于刹车事件或软件写操作而使 MOE=0 时，此位作用于配置为输出的通道。

有关详细信息，请参见 OC/OCN 使能说明（*第 24.4.11 节：TIM1 捕获/比较使能寄存器 (TIM1_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管）。

1：处于无效状态时，首先将 OC/OCN 输出强制为其无效电平，然后在死区后将其强制为空闲电平。定时器始终控制输出。

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。

位 9:8 **LOCK[1:0]**：锁定配置 (Lock configuration)

这些位用于针对软件错误提供写保护。

00：关闭锁定——不对任何位提供写保护。

01：锁定级别 1，此时无法对 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_BDTR 寄存器中的 *BK2BID*、*BKBID*、*BK2DSRM*、*BKDSRM*、*BK2P*、*BK2E*、*BK2F[3:0]*、*BKF[3:0]*、*AOE*、*BKP*、*BKE*、*OSSI*、*OSSR* 和 *DTG[7:0]* 位执行写操作。

10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（TIMx_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。

11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注：　复位后只能对 LOCK 位执行一次写操作。对 TIMx_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。

位 7:0 **DTG[7:0]**：配置死区发生器 (Dead-time generator setup)

此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。

DTG[7:5]=0xx => DT=DTG[7:0]x t_{dtg}，其中 $t_{dtg}=t_{DTS}$。

DTG[7:5]=10x => DT=(64+DTG[5:0])xt_{dtg}，其中 $T_{dtg}=2xt_{DTS}$。

DTG[7:5]=110 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=8xt_{DTS}$。

DTG[7:5]=111 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=16xt_{DTS}$。

示例：如果 T_{DTS}=125ns (8MHz)，则可能的死区值为：

0 到 15875 ns（步长为 125 ns）

16 us 到 31750 ns（步长为 250 ns）

32 us 到 63us（步长为 1 us）

64 us 到 126 us（步长为 2 us）

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位域即无法修改。

24.4.21 TIM1 DMA 控制寄存器 (TIM1_DCR)

TIM1 DMA control register

偏移地址：0x48

复位值：0x0000

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

位 15:13 保留，必须保持复位值。

位 12:8 **DBL[4:0]**：DMA 连续传送长度 (DMA burst length)

该 5 位向量定义了 DMA 的传送长度（当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送），即传送次数。可按半字或字节进行传送（请参见下面的示例）。

00000：1 次传送
00001：2 次传送
00010：3 次传送
...
10001：18 次传送

示例：以下面的传送为例：DBL = 7 字节且 DBA = TIMx_CR1。

– 如果 DBL = 7 字节且 DBA = TIMx_CR1 表示待传送字节的地址，应通过以下公式给出传送的地址：

(TIMx_CR1 地址) + DBA + (DMA 索引)，其中 DMA 索引 = DBL

在本例中，将为 (TIMx_CR1 地址) + DBA 加上 7 个字节，得到将要复制数据的源/目标地址。在这种情况下，将向自以下地址开始的 7 个寄存器传送数据：(TIMx_CR1 地址) + DBA

根据 DMA 数据大小的配置，可能发生下面几种情况：

– 如果按半字配置 DMA 数据大小，则将向 7 个寄存器中的每一个传送 16 位数据。

– 如果按字节配置 DMA 数据大小，也将向 7 个寄存器传送数据：第一个寄存器包含第一个 MSB 字节，第二个寄存器包含第一个 LSB 字节，依此类推。因此，使用传送定时器时，还必须指定 DMA 传送的数据大小。

位 7:5 保留，必须保持复位值。

位 4:0 **DBA[4:0]**：DMA 基址 (DMA base address)

该 5 位向量定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：

00000：TIMx_CR1
00001：TIMx_CR2
00010：TIMx_SMCR
...

24.4.22 TIM1 全传输 DMA 地址 (TIM1_DMAR)

TIM1 DMA address for full transfer

偏移地址：0x4C

复位值：0x0000 0000

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

位 31:0 **DMAB[31:0]**：DMA 连续传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器：(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

24.4.23 TIM1 选择寄存器 1 (TIM1_OR1)

TIM1 option register 1

偏移地址：0x50

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP	Res.	Res.	TIM1_ETR_ADC1_RMP	
											rw			rw	rw

位 31:5 保留，必须保持复位值。

位 4 **TI1_RMP**：输入捕获 1 重映射 (Input Capture 1 remap)

0：TIM1 输入捕获 1 连接到 I/O

1：TIM1 输入捕获 1 连接到 COMP1 输出

位 3:2 保留，必须保持复位值。

位 1:0 **TIM1_ETR_ADC1_RMP[1:0]**：TIM1_ETR_ADC1 重映射功能 (TIM1_ETR_ADC1 remapping capability)

00：TIM1_ETR 未连接到 ADC1 AWDx（当 ETR 来自 ETR 输入引脚时，必须选择此项）

01：TIM1_ETR 连接到 ADC1 AWD1

10：TIM1_ETR 连接到 ADC1 AWD2

11：TIM1_ETR 连接到 ADC1 AWD3

注： ADC1 AWDx 源与 TIM1_ETR 输入信号进行逻辑“或”运算。使用 ADC1 AWDx 时，需要确保复用功能控制器中未使能相应的 TIM1_ETR 输入引脚。

24.4.24 TIM1 捕获/比较模式寄存器 3 (TIM1_CCMR3)

TIM1 capture/compare mode register 3

偏移地址：0x54

复位值：0x0000 0000

通道 5 和通道 6 只能配置为输出。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6 CE	OC6M[2:0]			OC6 PE	OC6FE	Res.	Res.	OC5 CE	OC5M[2:0]			OC5PE	OC5FE	Res.	Res.
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 **OC6CE**：输出比较 6 清零使能 (Output compare 6 clear enable)
请参见 OC1CE 说明。

位 24、14、13、12 **OC6M[3:0]**：输出比较 6 模式 (Output Compare 6 mode)
请参见 OC1M 说明。

位 11 **OC6PE**：输出比较 6 预装载使能 (Output compare 6 preload enable)
请参见 OC1PE 说明。

位 10 **OC6FE**：输出比较 6 快速使能 (Output compare 6 fast enable)
请参见 OC1FE 说明。

位 9:8 保留，必须保持复位值。

位 7 **OC5CE**：输出比较 5 清零使能 (Output compare 5 clear enable)
请参见 OC1CE 说明。

位 16、6、5、4 **OC5M[3:0]**：输出比较 5 模式 (Output Compare 5 mode)
请参见 OC1M 说明。

位 3 **OC5PE**：输出比较 5 预装载使能 (Output Compare 5 preload enable)
请参见 OC1PE 说明。

位 2 **OC5FE**：输出比较 5 快速使能 (Output compare 5 fast enable)
请参见 OC1FE 说明。

位 1:0 保留，必须保持复位值。

24.4.25 TIM1 捕获/比较寄存器 5 (TIM1_CCR5)

TIM1 capture/compare register 5

偏移地址：0x58

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **GC5C3**：通道 5 和通道 3 组合 (Group Channel 5 and Channel 3)

通道 3 输出上失真：

0：OC5REF 对 OC3REFC 无影响

1：OC3REFC 是 OC3REFC 和 OC5REF 的逻辑与运算结果

此位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR2 中选择了预装载功能）。

注： 也可在组合 PWM 信号上应用此失真。

位 30 **GC5C2**：通道 5 和通道 2 组合 (Group Channel 5 and Channel 2)

通道 2 输出上失真：

0：OC5REF 对 OC2REFC 无影响

1：OC2REFC 是 OC2REFC 和 OC5REF 的逻辑与运算结果

此位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR1 中选择了预装载功能）。

注： 也可在组合 PWM 信号上应用此失真。

位 29 **GC5C1**：通道 5 和通道 1 组合 (Group Channel 5 and Channel 1)

通道 1 输出上失真：

0：OC5REF 对 OC1REFC5 无影响

1：OC1REFC 是 OC1REFC 和 OC5REF 的逻辑与运算结果

此位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR1 中选择了预装载功能）。

注： 也可在组合 PWM 信号上应用此失真。

位 28:16 保留，必须保持复位值。

位 15:0 **CCR5[15:0]**：捕获/比较 5 值 (Capture/Compare 5 value)

CCR5 是捕获/比较寄存器 5 的预装载值。

如果没有通过 TIMx_CCMR3 寄存器中的 OC5PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 5）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC5 输出上发出信号的值。

24.4.26 TIM1 捕获/比较寄存器 6 (TIM1_CCR6)

TIM1 capture/compare register 6

偏移地址：0x5C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR6[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR6[15:0]**：捕获/比较 6 值 (Capture/Compare 6 value)

CCR6 是捕获/比较寄存器 6 的预装载值。

如果没有通过 TIMx_CCMR3 寄存器中的 OC6PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 6）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC6 输出上发出信号的值。

24.4.27 TIM1 复用功能选项寄存器 1 (TIM1_AF1)

TIM1 alternate function option register 1

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL[3:2]	
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		Res.	Res.	BK CMP2P	BK CMP1P	BKINP	Res.	Res.	Res.	Res.	Res.	Res.	BK CMP2E	BK CMP1E	BKINE
rw	rw			rw	rw	rw							rw	rw	rw

位 31:18 保留，必须保持复位值。

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)

这些位选择 ETR 输入源。

0000：ETR 传统模式

0001：COMP1 输出

0010：COMP2 输出

其他值：保留

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，这些位即无法修改。

位 13:12 保留，必须保持复位值。

位 11 **BKCMP2P**：BRK COMP2 输入极性 (BRK COMP2 input polarity)

此位选择 COMP2 输入有效电平，必须与 BKP 极性位一起编程。

0：COMP2 输入为高电平有效

1：COMP2 输入为低电平有效

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BKCMP1P**：BRK COMP1 输入极性 (BRK COMP1 input polarity)
此位选择 COMP1 输入有效电平，必须与 BKP 极性位一起编程。
0：COMP1 输入为高电平有效
1：COMP1 输入为低电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
此位选择 BKIN 复用功能输入有效电平，必须与 BKP 极性位一起编程。
0：BKIN 输入为高电平有效
1：BKIN 输入为低电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8:3 保留，必须保持复位值。

位 2 **BKCMP2E**：BRK COMP2 使能 (BRK COMP2 enable)
此位使能定时器 BRK 输入的 COMP2。COMP2 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BKCMP1E**：BRK COMP1 使能 (BRK COMP1 enable)
此位使能定时器 BRK 输入的 COMP1。COMP1 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
此位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

注： 请参见图 170：TIM1 ETR 输入电路和图 191：刹车和刹车 2 电路概述。

24.4.28 TIM1 复用功能寄存器 2 (TIM1_AF2)

TIM1 Alternate function register 2

偏移地址：0x64

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	BK2 CMP2 P	BK2 CMP1 P	BK2 INP	Res.	Res.	Res.	Res.	Res.	Res.	BK2 CMP2E	BK2 CMP1E	BK2INE
				rw	rw	rw							rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11 **BK2CMP2P**：BRK2 COMP2 输入极性 (BRK2 COMP2 input polarity)
此位选择 COMP2 输入有效电平，必须与 BK2P 极性位一起编程。
0：COMP2 输入为低电平有效
1：COMP2 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BK2CMP1P**：BRK2 COMP1 输入极性 (BRK2 COMP1 input polarity)
此位选择 COMP1 输入有效电平，必须与 BK2P 极性位一起编程。
0：COMP1 输入为低电平有效
1：COMP1 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BK2INP**：BRK2 BKIN2 输入极性 (BRK2 BKIN2 input polarity)
此位选择 BKIN2 复用功能输入有效电平，必须与 BK2P 极性位一起编程。
0：BKIN2 输入为低电平有效
1：BKIN2 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8:3 保留，必须保持复位值。

位 2 **BK2CMP2E**：BRK2 COMP2 使能 (BRK2 COMP2 enable)
此位使能定时器 BRK2 输入的 COMP2。COMP2 输出与其他 BRK2 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BK2CMP1E**：BRK2 COMP1 使能 (BRK2 COMP1 enable)
此位使能定时器 BRK2 输入的 COMP1。COMP1 输出与其他 BRK2 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BK2INE**：BRK2 BKIN 输入使能 (BRK2 BKIN input enable)
此位使能定时器 BRK2 输入的 BKIN2 复用功能。BKIN2 输入与其他 BRK2 源进行“或”运算。
0：禁止 BKIN2 输入
1：使能 BKIN2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

注： 请参见图 191：刹车和刹车 2 电路概述。

24.4.29 TIM1 定时器输入选择寄存器 (TIM1_TISEL)

TIM1 timer input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw						rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **TI4SEL[3:0]**：选择 TI4[0] 到 TI4[15] 输入 (selects TI4[0] to TI4[15] input)
0000：TIM1_CH4 输入
其他值：保留

位 23:20 保留，必须保持复位值。

位 19:16 **TI3SEL[3:0]**：选择 TI3[0] 到 TI3[15] 输入 (selects TI3[0] to TI3[15] input)
0000：TIM1_CH3 输入
其他值：保留

位 15:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：选择 TI2[0] 到 TI2[15] 输入 (selects TI2[0] to TI2[15] input)
0000：TIM1_CH2 输入
其他值：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM1_CH1 输入
其他值：保留

24.4.30 TIM1 寄存器映射

TIM1 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 156. TIM1 寄存器映射和复位值

偏移	寄存器名称	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
0x00	**TIM1_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0	0	0	0	0	0	0	0
0x04	**TIM1_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5	Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS [2:0]			CCDS	CCUS	Res.	CCPC
	Reset value									0	0	0	0		0		0		0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x08	**TIM1_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3]		Res.	Res.	Res.	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			OCCS	SMS[2:0]		
	Reset value											0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**TIM1_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**TIM1_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6IF	CC5IF	Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value															0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**TIM1_EGR**	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COM	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																								0	0	0	0	0	0	0	0	0
0x18	**TIM1_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM1_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**TIM1_CCMR2** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	OC4CE	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM1_CCMR2 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4PSC [1:0]		CC4S [1:0]		IC3F[3:0]				IC3PSC [1:0]		CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIM1_CCER**	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E	CC4NP	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Reset value											0	0			0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 156. TIM1 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x24	**TIM1_CNT**	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIM1_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM1_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	**TIM1_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	**TIM1_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIM1_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**TIM1_CCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	**TIM1_CCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIM1_BDTR**	Res.	Res.	BK2BID	BKBID	BK2DSRM	BKDSRM	BK2P	BK2E	BK2F[3:0]				BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]							
	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
0x48	**TIM1_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIM1_DMAR**	DMAB[31:0]																															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	**TIM1_OR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP	Res.	Res.	TIM1_ETR_ADC1_RMP	
	Reset value																												0			0	0

表 156. TIM1 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x54	**TIM1_CCMR3** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]	OC6CE	OC6M [2:0]			OC6PE	OC6FE	Res.	Res.	OC5CE	OC5M [2:0]			OC5PE	OC5FE	Res.	Res.
	Reset value								0								0	0	0	0	0	0	0			0	0	0	0	0	0		
0x58	**TIM1_CCR5**	GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR5[15:0]															
	Reset value	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C	**TIM1_CCR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR6[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM1_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res ETRSEL [3:0]						BKCMP2P	BKCMP1P	BKINP	Res.	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
	Reset value															0	0	0	0			0	0	0							0	0	1
0x64	**TIM1_AF2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BK2CMP2P	BK2CMP1P	BK2INP	Res.	Res.	Res.	Res.	Res.	Res.	BK2CMP2E	BK2CMP1E	BK2INE
	Reset value																					0	0	0							0	0	1
0x68	**TIM1_TISEL**	Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]				Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

25 通用定时器 (TIM2)

25.1 TIM2 简介

通用定时器 TIM2 包含一个 32 位自动重载计数器，该计数器由可编程预分频器驱动。

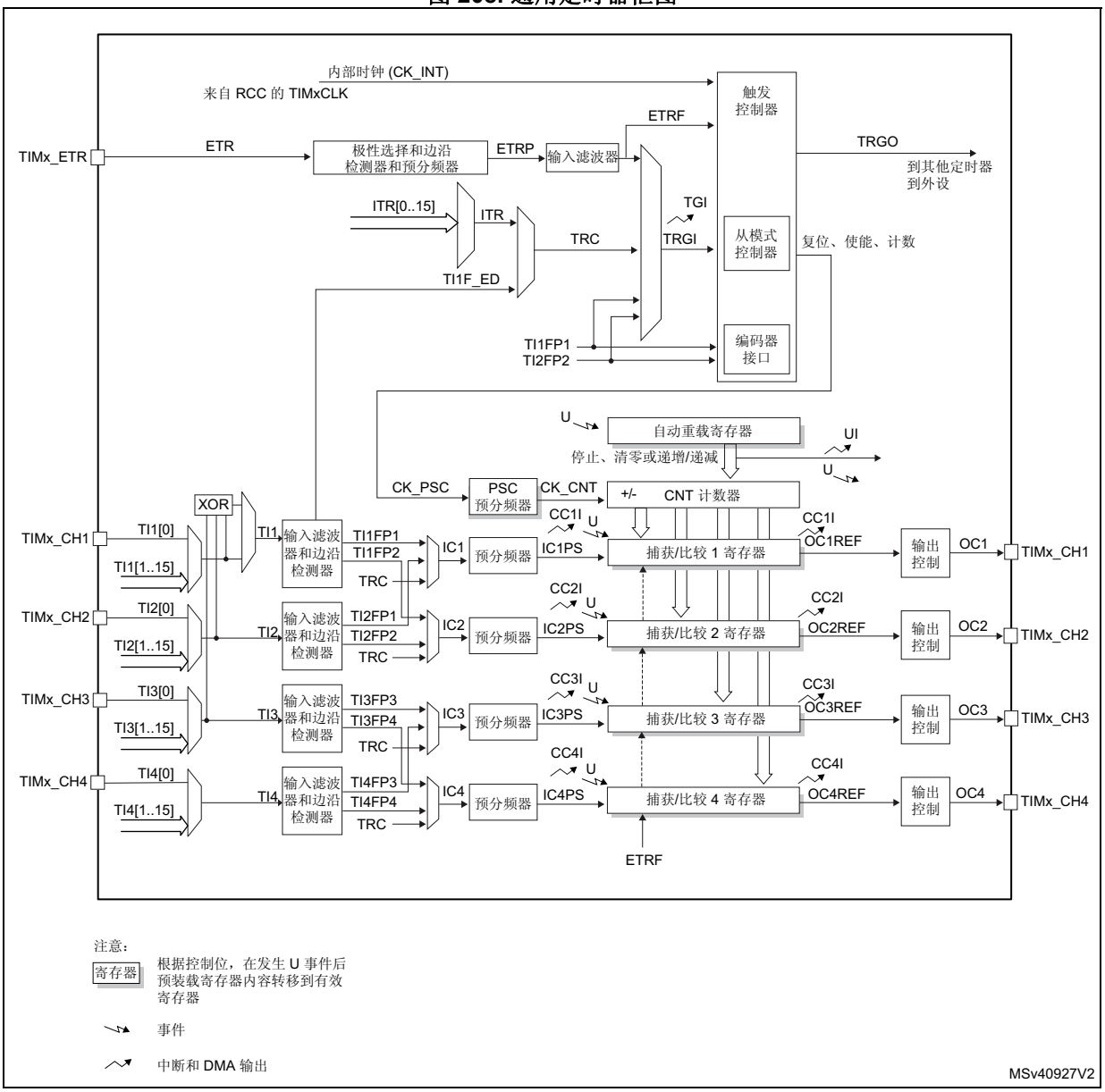
该定时器可用于多种用途，包括测量输入信号的脉冲宽度（*输入捕获*）或生成输出波形（*输出比较和 PWM*）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

25.2 TIM2 主要特性

- 32 位递增、递减、递增/递减自动重载计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 到 65535 之间。
- 多达 4 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿和中心对齐模式）
  - 单脉冲模式输出
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较
- 支持定位用增量（正交）编码器和霍尔传感器电路
- 触发输入用作外部时钟或逐周期电流管理

图 208. 通用定时器框图



25.3 TIM2 功能描述

25.3.1 时基单元

可编程定时器的主要模块由一个 32 位计数器及其相关的自动重载寄存器组成。计数器可递增计数、递减计数或交替进行递增和递减计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

请注意，实际的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

预分频器说明

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 16 位/32 位寄存器（TIMx_PSC 寄存器）所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 209 和*图 210* 以一些示例说明在预分频比实时变化时计数器的行为：

图 209. 预分频器分频由 1 变为 2 时的计数器时序图

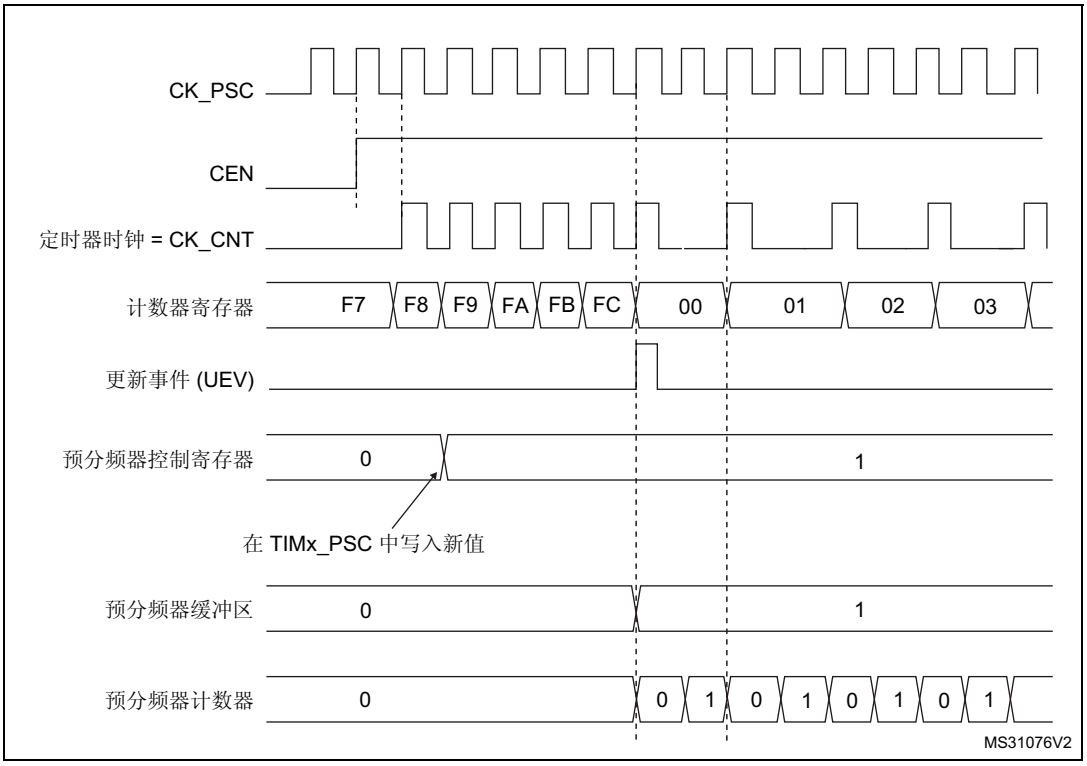
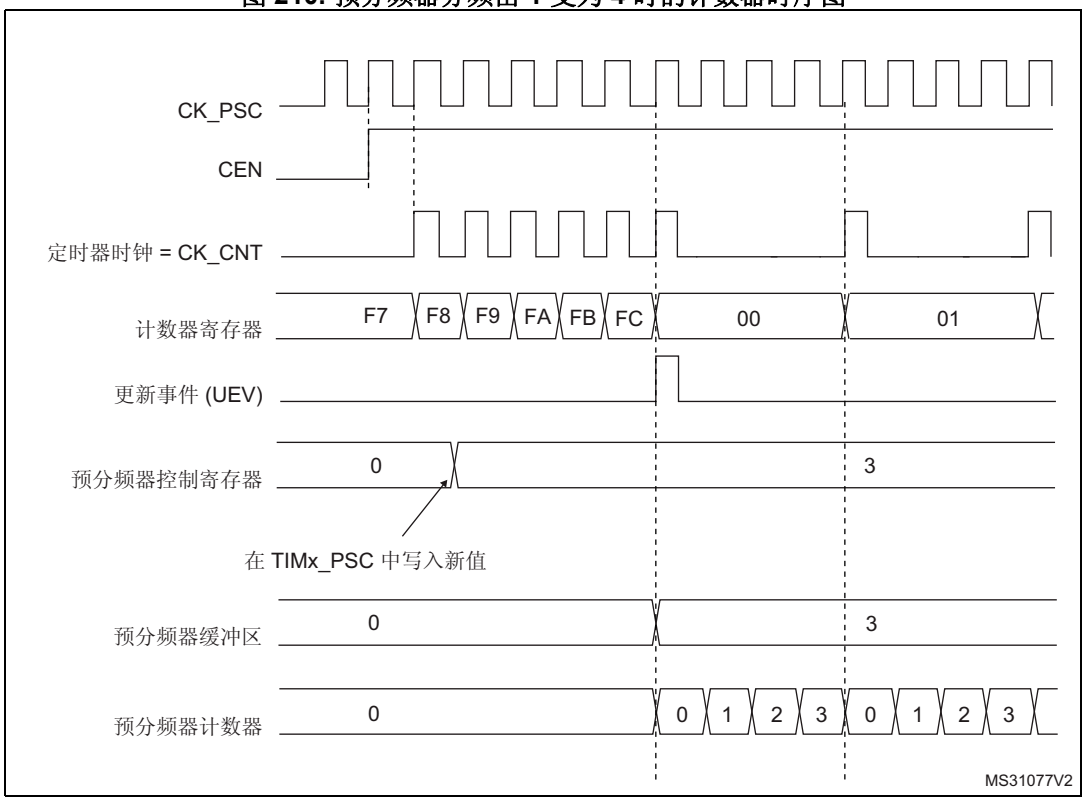


图 210. 预分频器分频由 1 变为 4 时的计数器时序图



25.3.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

每次发生计数器上溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）
- 使用预装载值 (TIMx_ARR) 更新自动重载影子寄存器

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 211. 计数器时序图，1 分频内部时钟

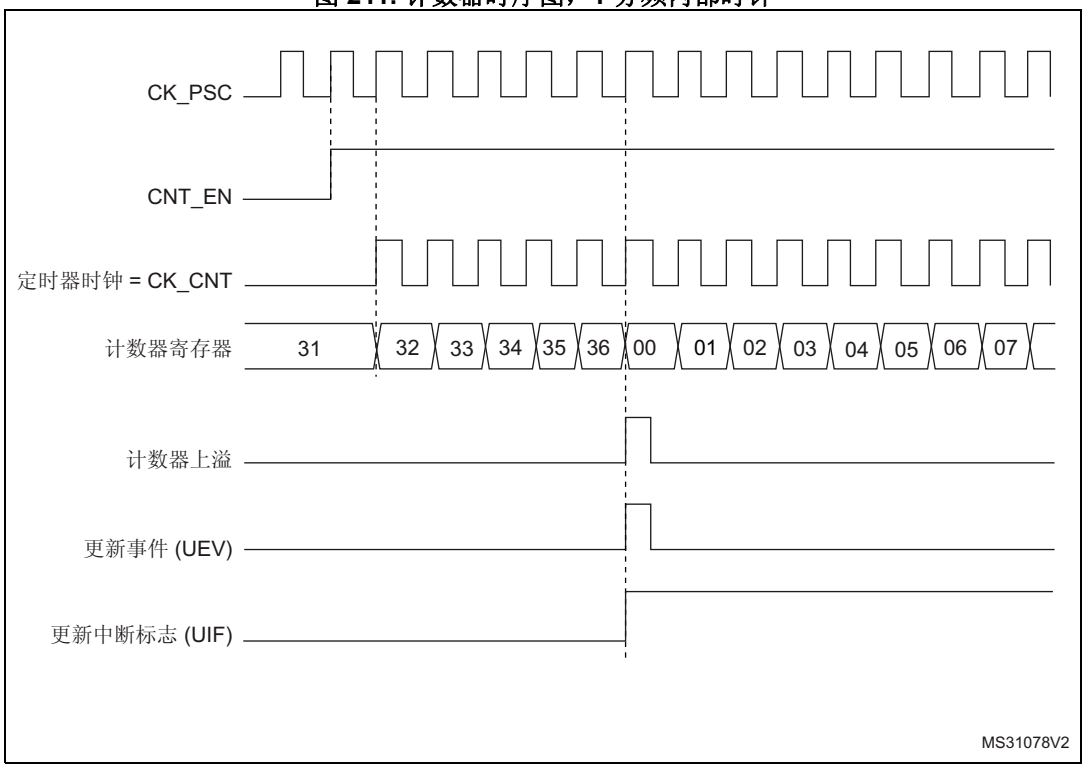


图 212. 计数器时序图，2 分频内部时钟

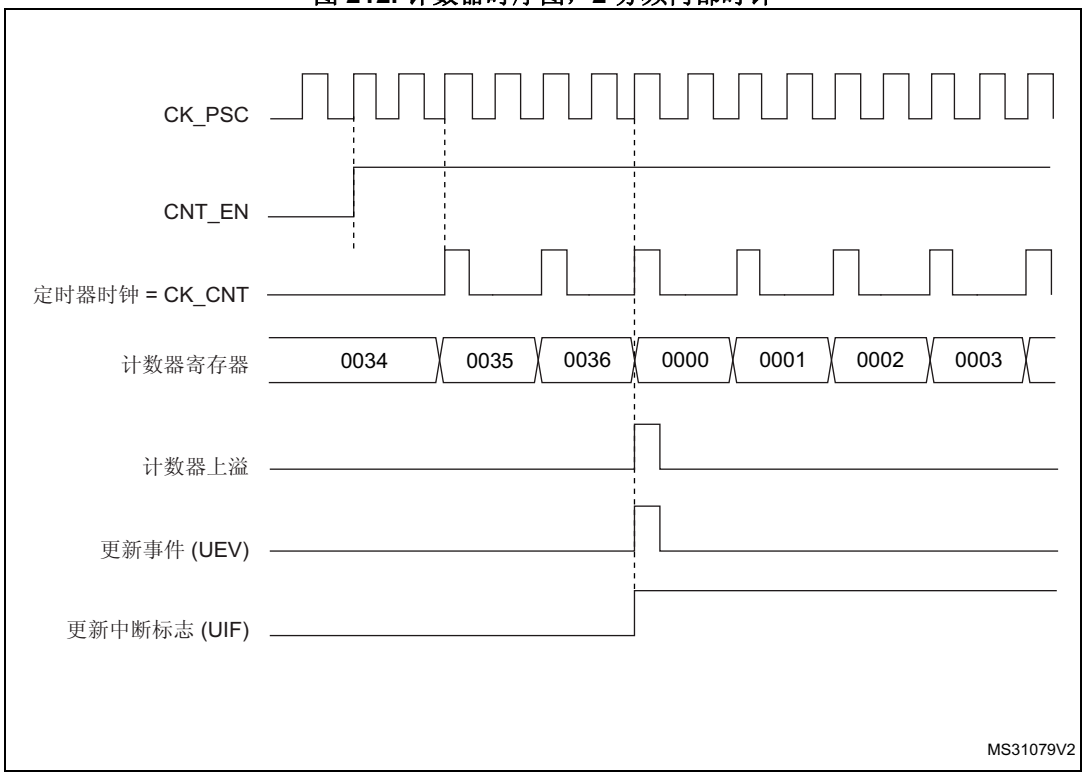


图 213. 计数器时序图，4 分频内部时钟

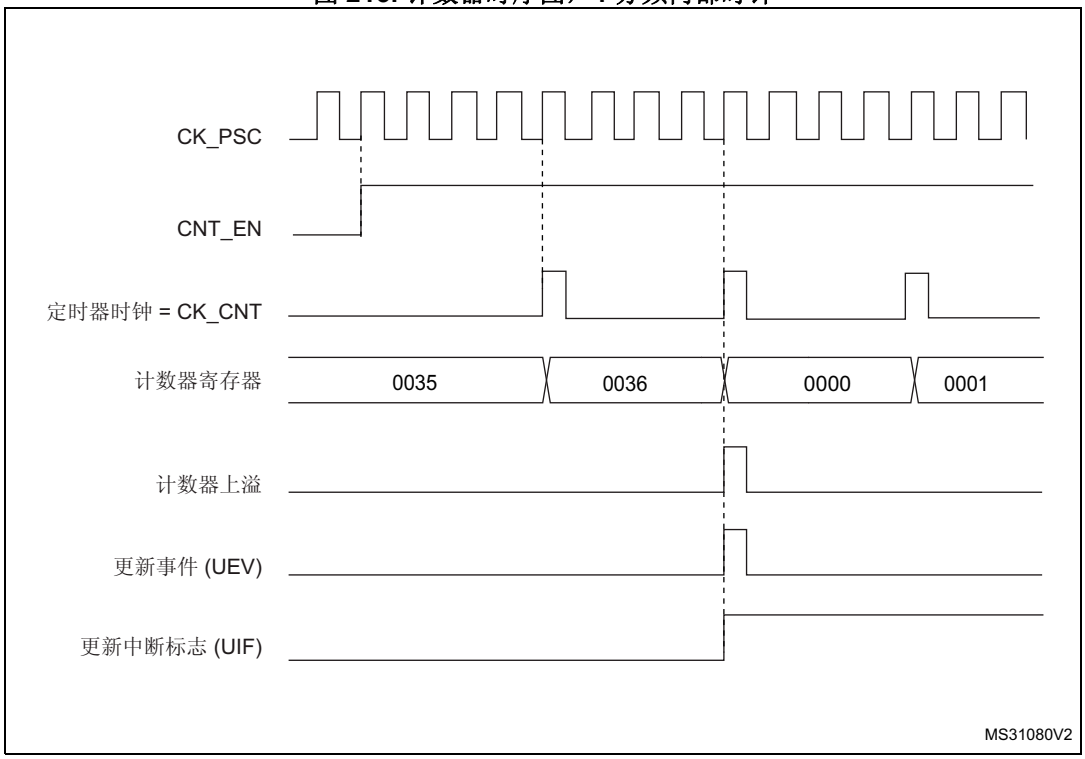


图 214. 计数器时序图，N 分频内部时钟

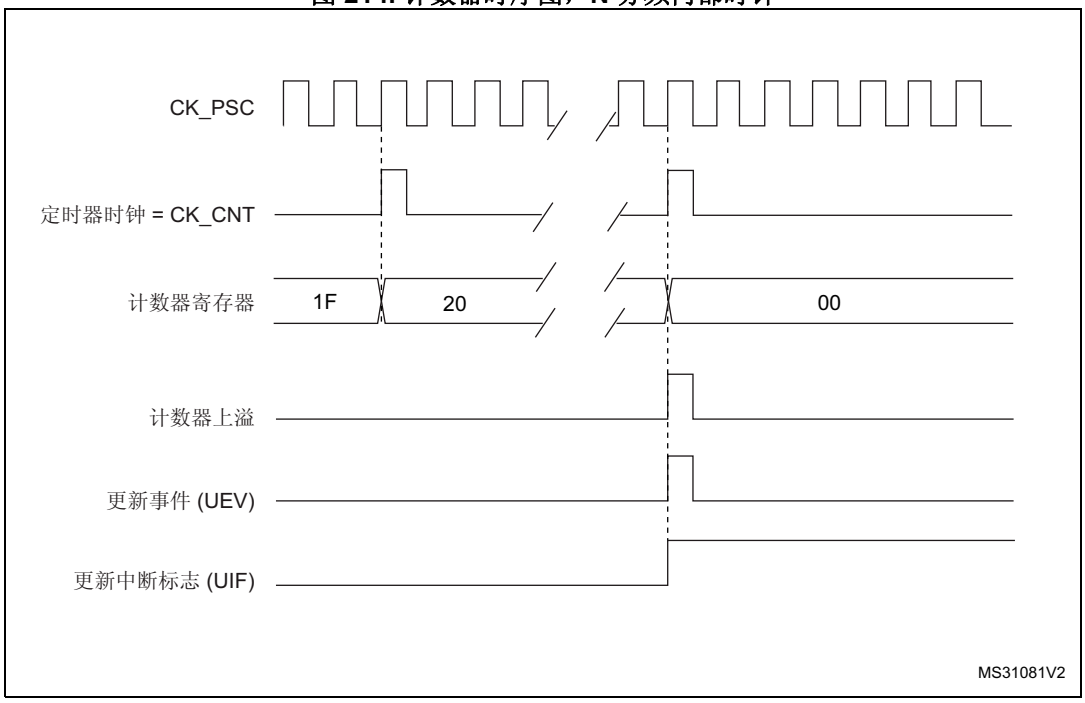


图 215. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

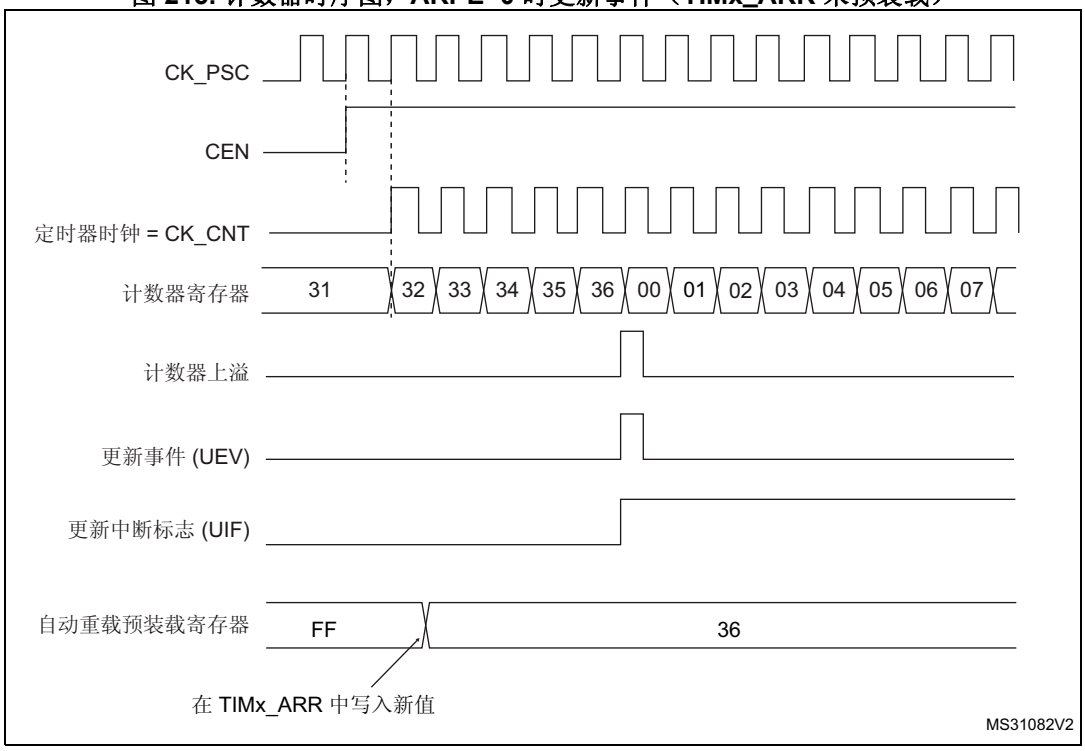
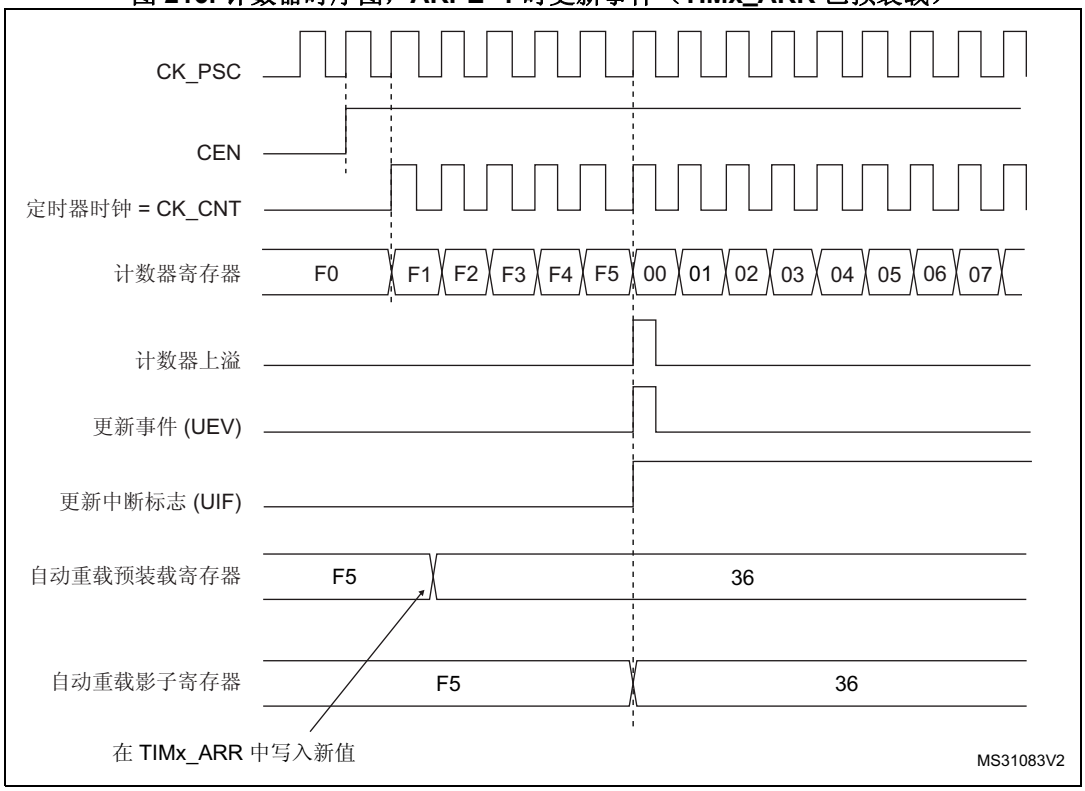


图 216. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



递减计数模式

在递减计数模式下，计数器从自动重载值（TIMx_ARR 寄存器的内容）开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

每次发生计数器下溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分频比保持不变）。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载有效寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，ARR 寄存器更新在计数器重载之前被更新，因此下一个周期就是预期的值。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 217. 计数器时序图，1 分频内部时钟

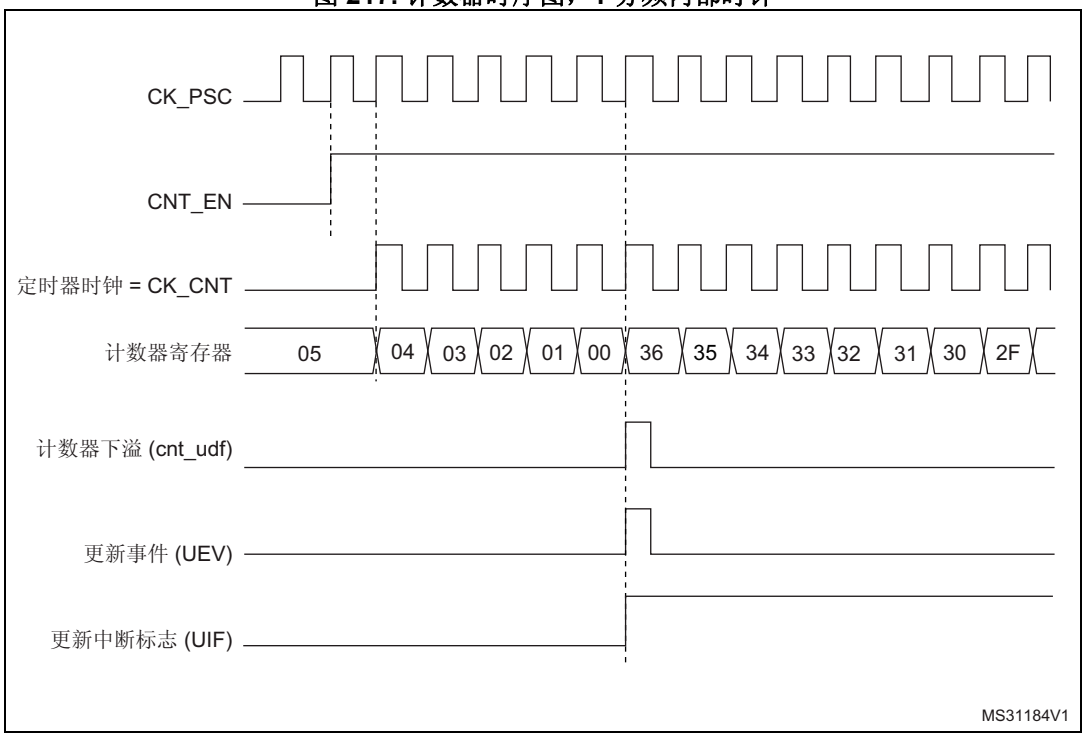


图 218. 计数器时序图，2 分频内部时钟

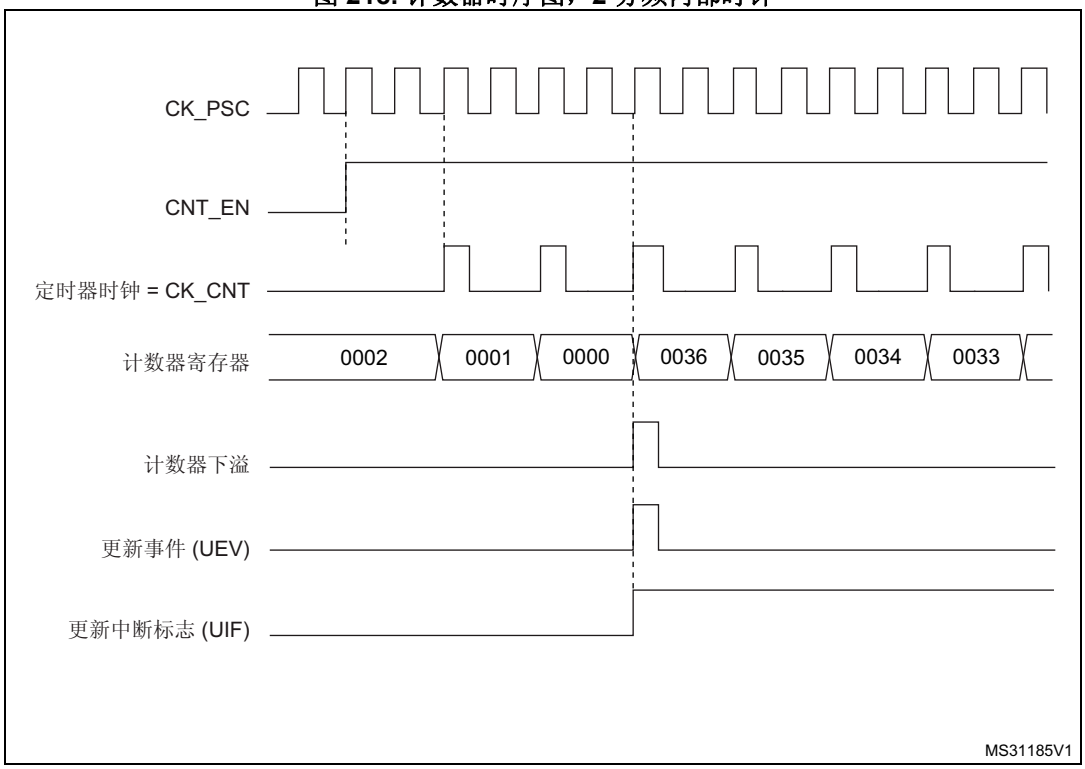


图 219. 计数器时序图，4 分频内部时钟

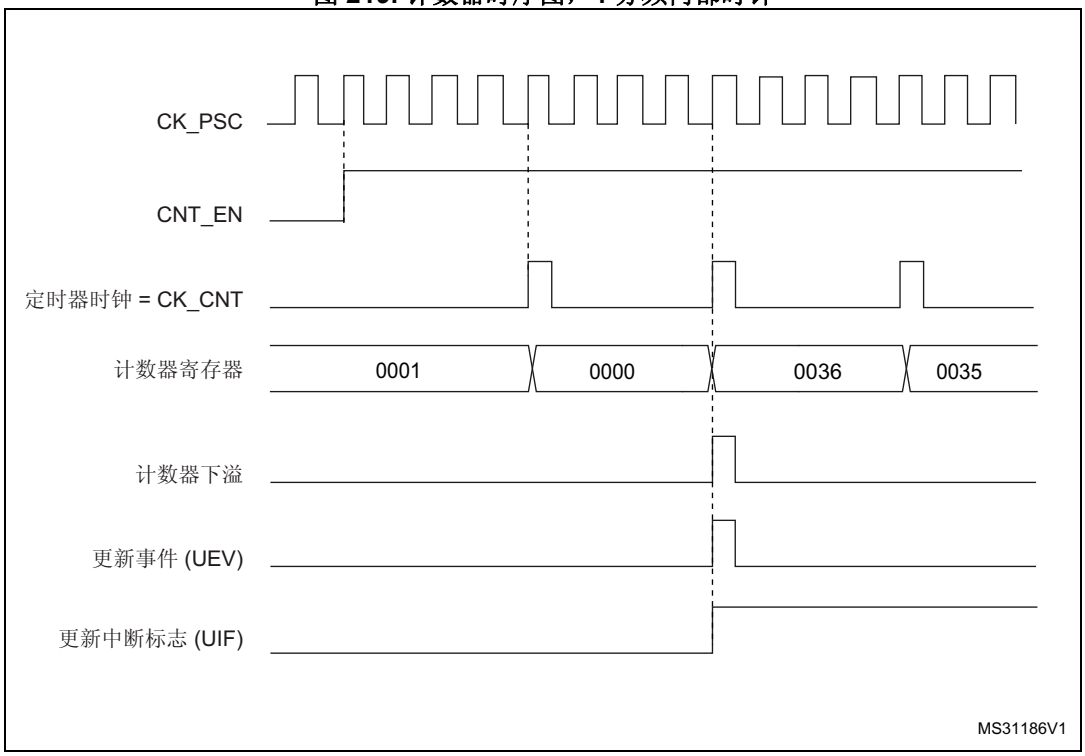


图 220. 计数器时序图，N 分频内部时钟

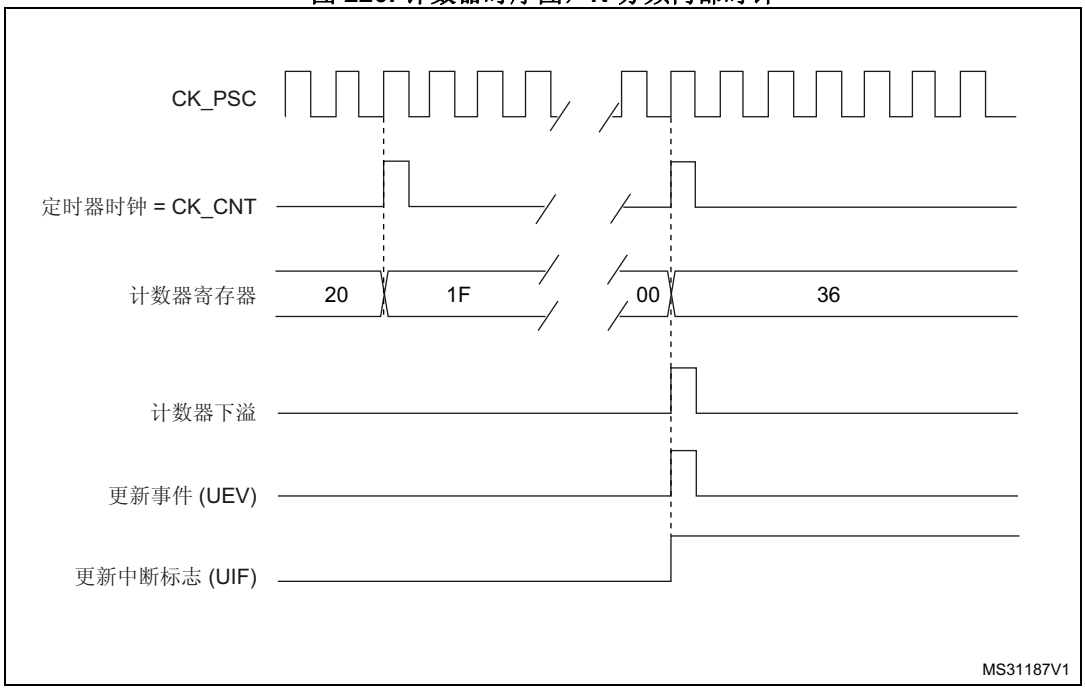
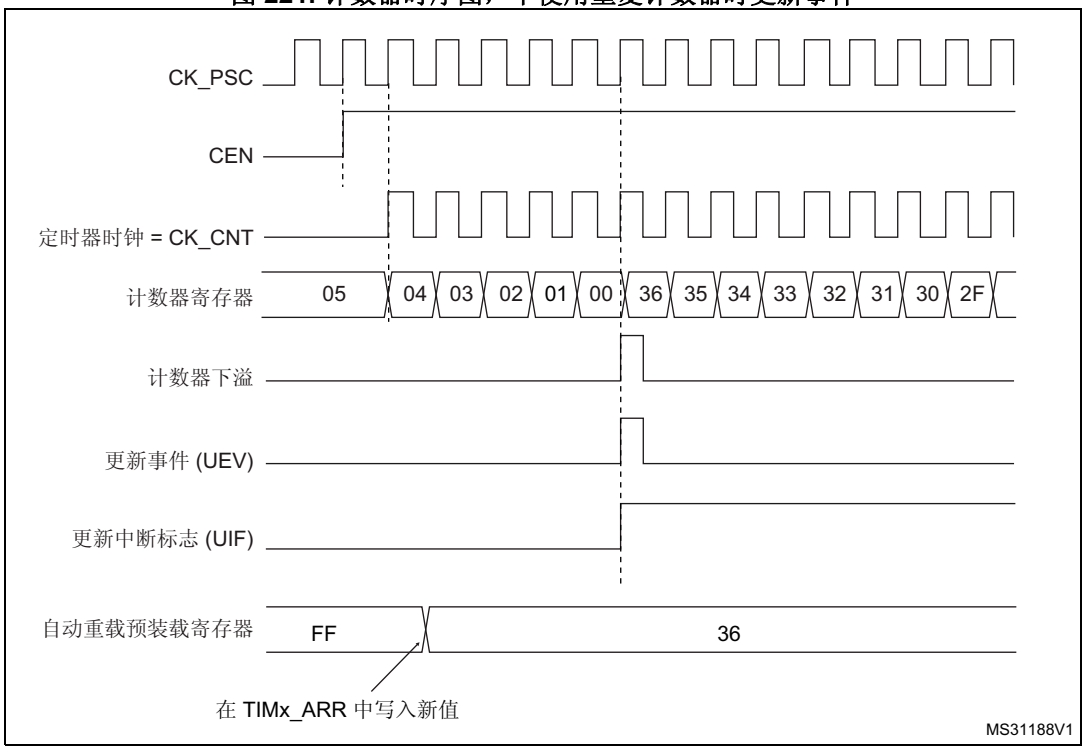


图 221. 计数器时序图，不使用重复计数器时更新事件



中心对齐模式（递增/递减计数）

在中心对齐模式下，计数器从 0 开始计数到自动重载值（TIMx_ARR 寄存器的内容）– 1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中心对齐模式 1，CMS = “01”）、计数器递增计数（中心对齐模式 2，CMS = “10”）以及计数器递增/递减计数（中心对齐模式 3，CMS = “11”）。

此模式下无法写入方向位（TIMx_CR1 寄存器中的 DIR 位）。而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

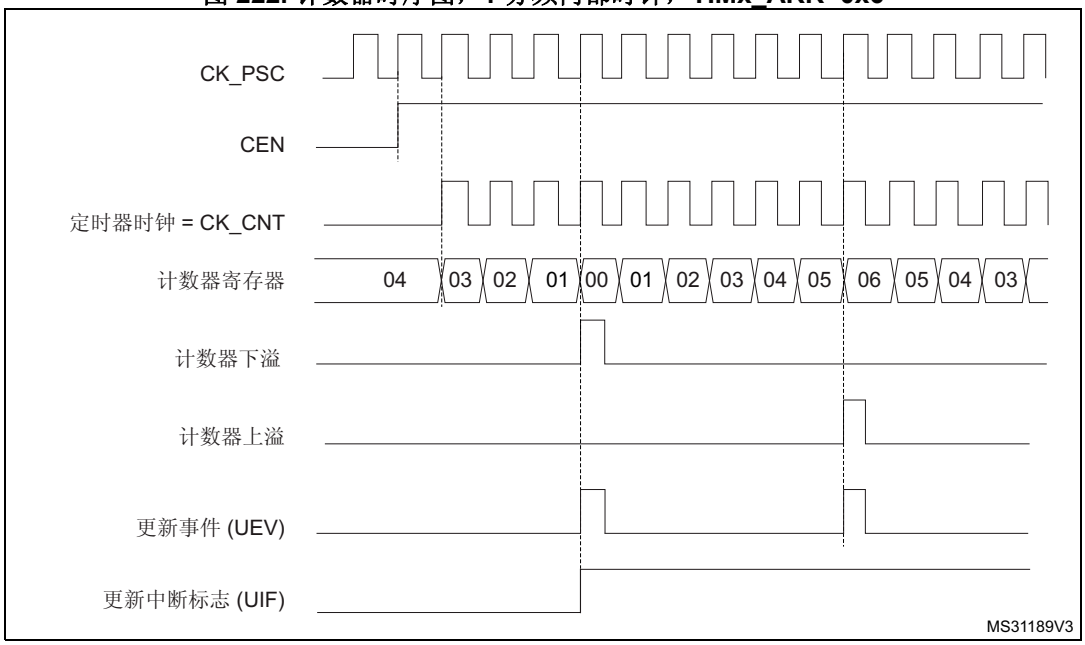
此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载有效寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，如果更新操作是由计数器上溢触发的，则 ARR 寄存器在计数器重载之前更新，因此，下一个计数周期就是我们所希望的新的周期长度（计数器被重载新的值）。

以下各图以一些示例说明不同时钟频率下计数器的行为。

图 222. 计数器时序图，1 分频内部时钟，TIMx_ARR=0x6



1. 此处使用中心对齐模式 1（有关详细信息，请参见*第 784 页的第 25.4.1 节：TIM2 控制寄存器 1 (TIM2_CR1)*）。

图 223. 计数器时序图，2 分频内部时钟

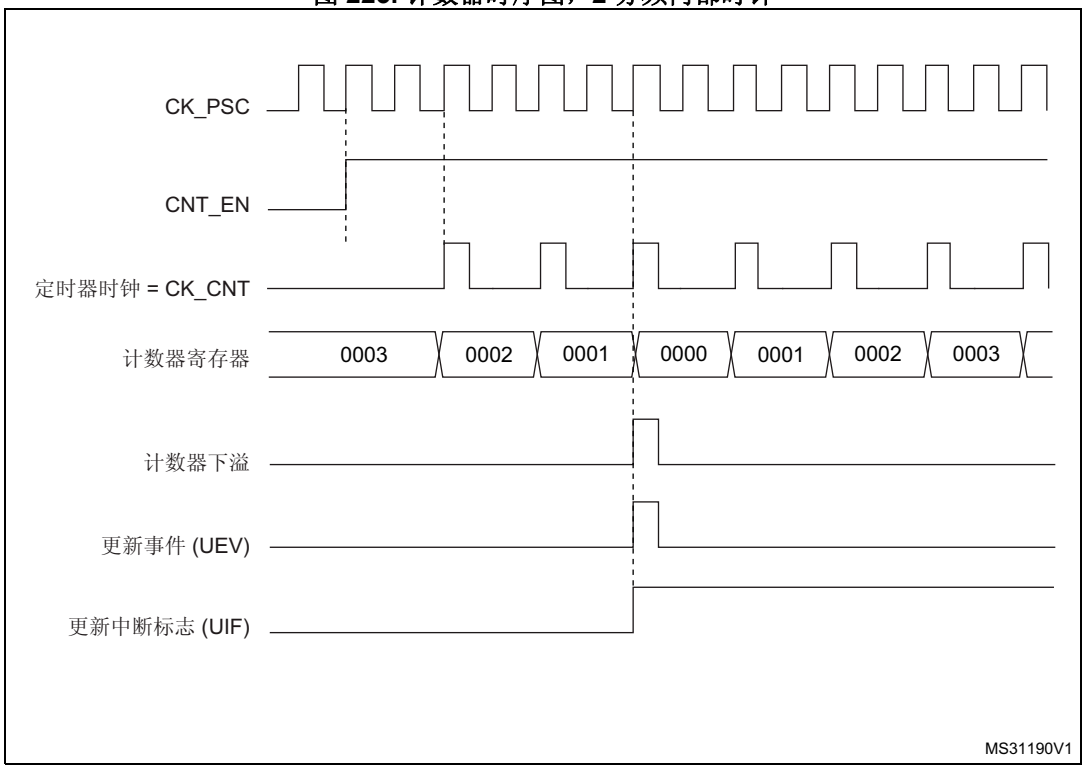
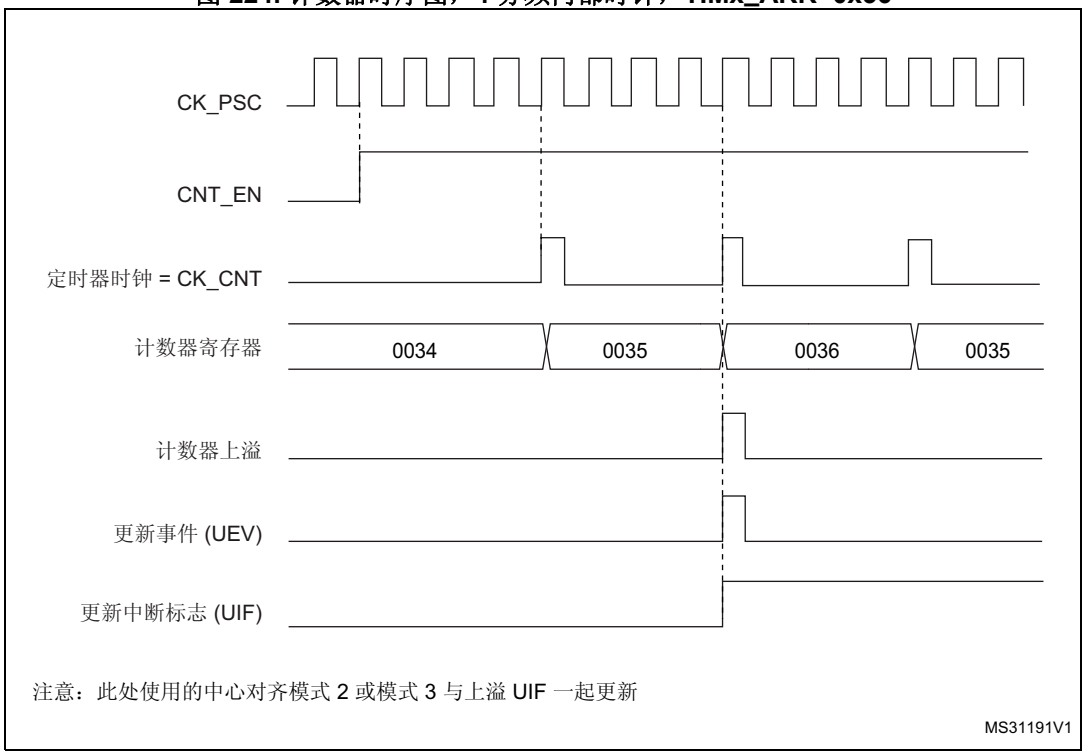


图 224. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36



1. 中心对齐模式 2 或模式 3 与上溢 UIF 结合使用。

图 225. 计数器时序图，N 分频内部时钟

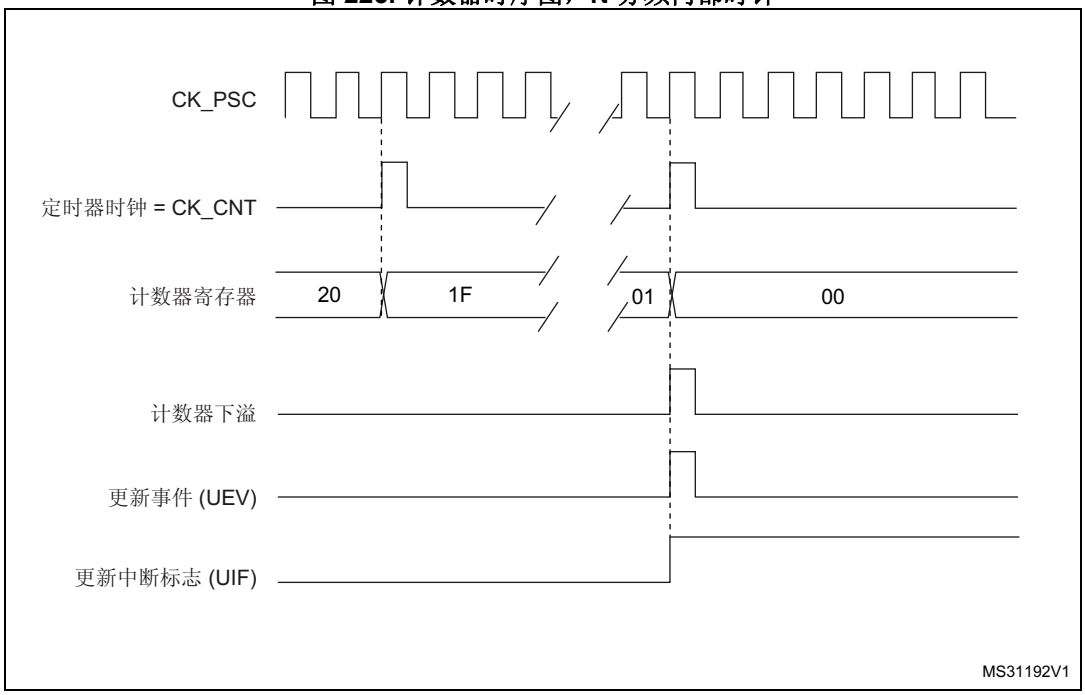


图 226. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

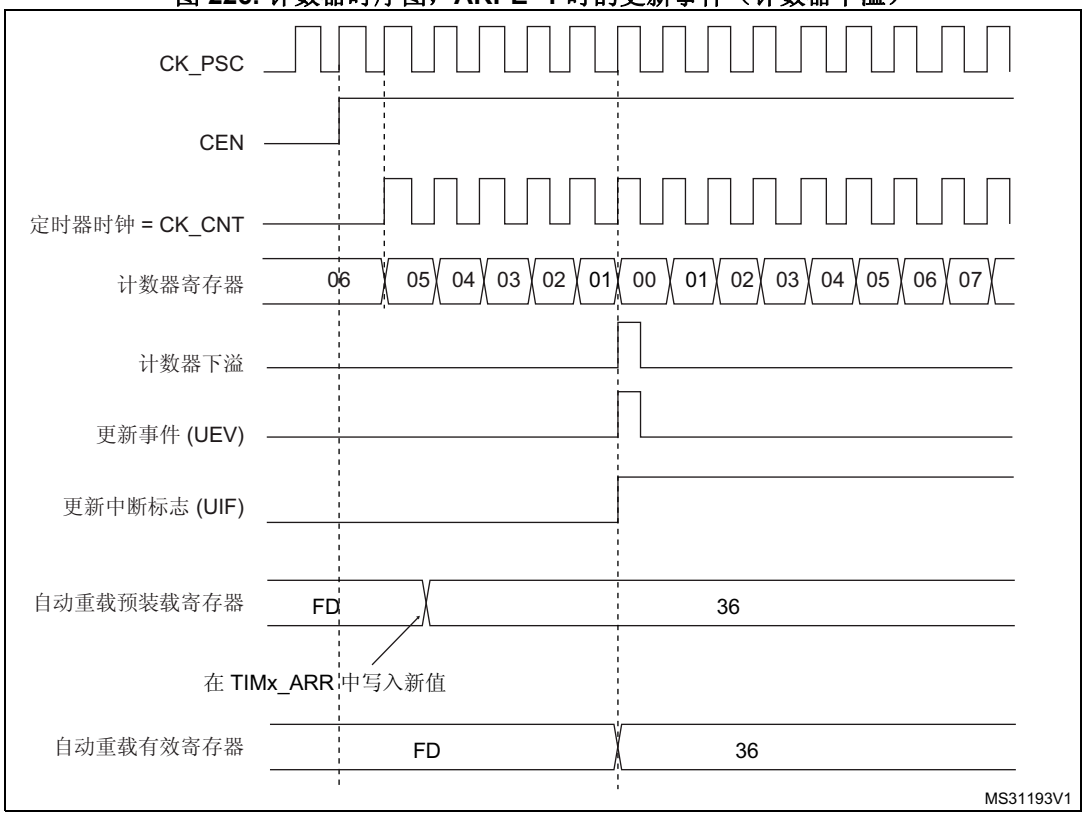
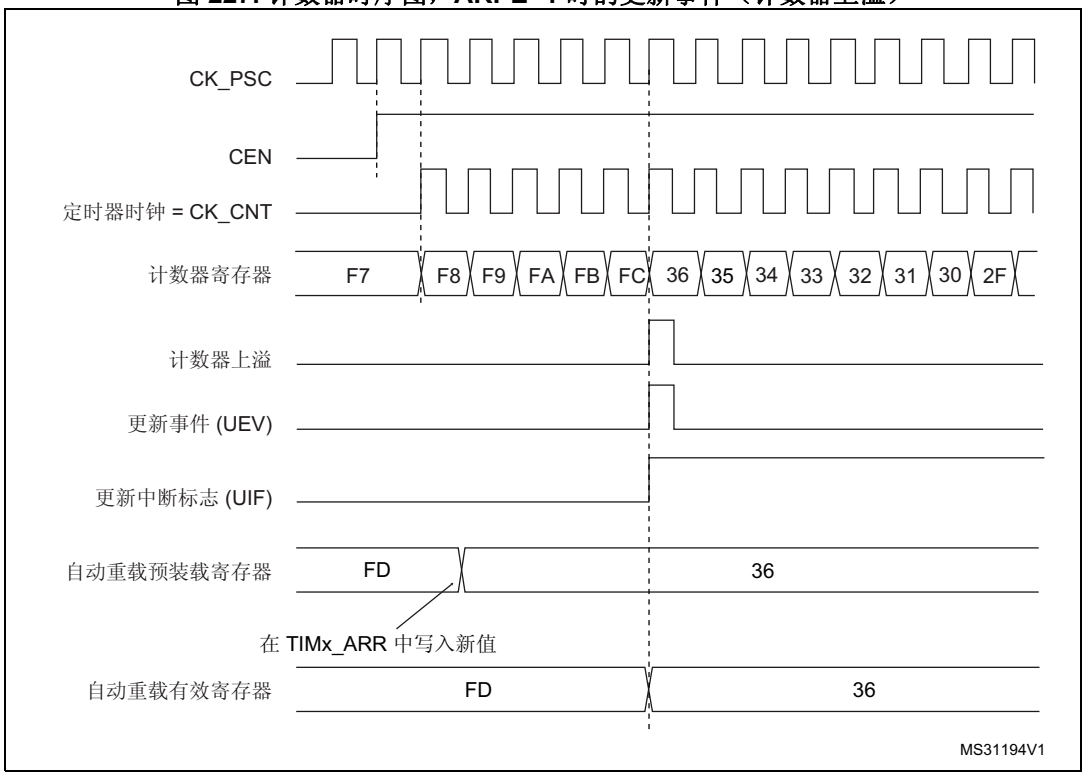


图 227. 计数器时序图，ARPE=1 时的更新事件（计数器上溢）



25.3.3 时钟选择

计数器时钟可由下列时钟源提供：

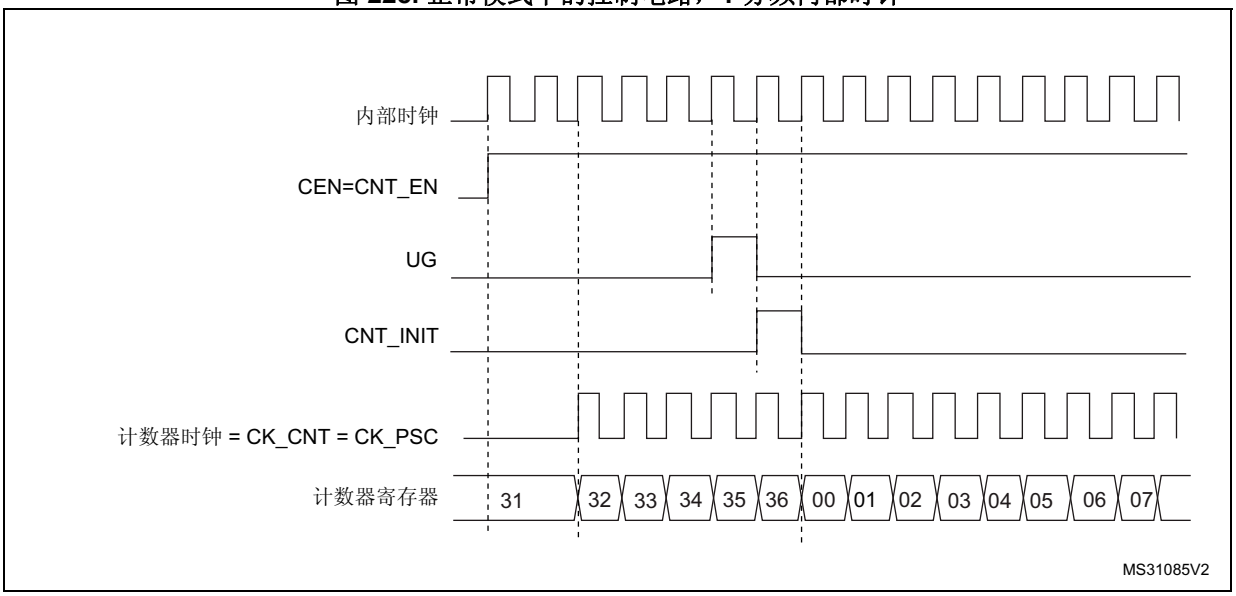
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚 (TIx)
- 外部时钟模式 2：外部触发输入 (ETR)
- 内部触发输入 (ITRx)：使用一个定时器作为另一个定时器的预分频器，例如可以将定时器 X 配置为定时器 Y 的预分频器。更多详细信息，请参见*第 780 页的将一个定时器用作另一个定时器的预分频器*。

内部时钟源 (CK_INT)

如果禁止从模式控制器（TIMx_SMCR 寄存器中 SMS=000），则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 228 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

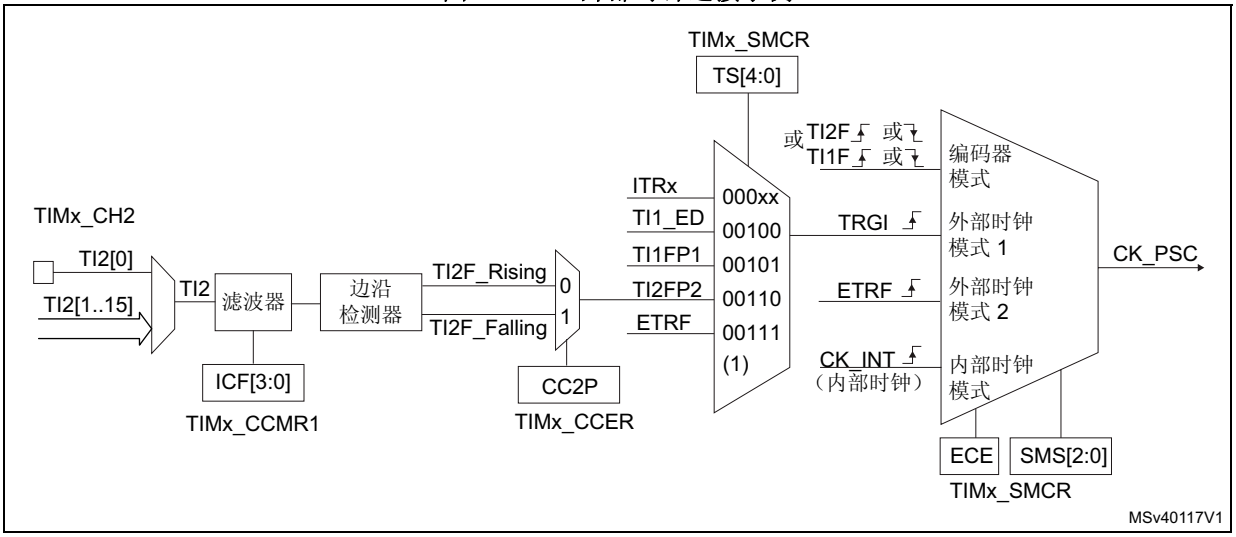
图 228. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 229. TI2 外部时钟连接示例



1. 保留从 01000 到 11111 的代码：ITRy。

例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2x 源（内部或外部）。
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S=“01”来配置通道 2，使其能够检测 TI2 输入的上升沿。
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。

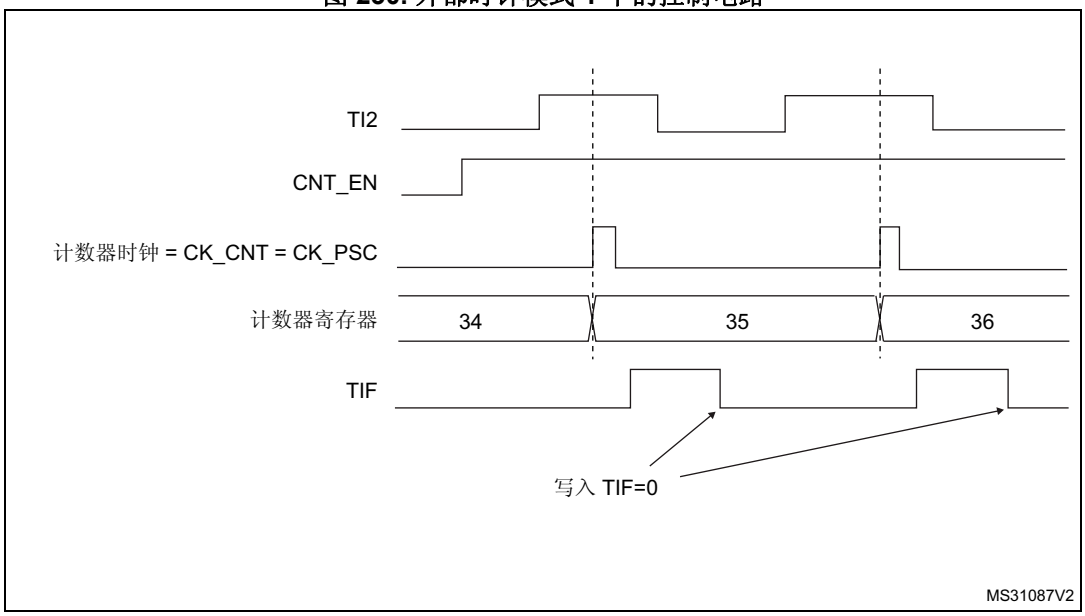
注： *由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

4. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 230. 外部时钟模式 1 下的控制电路



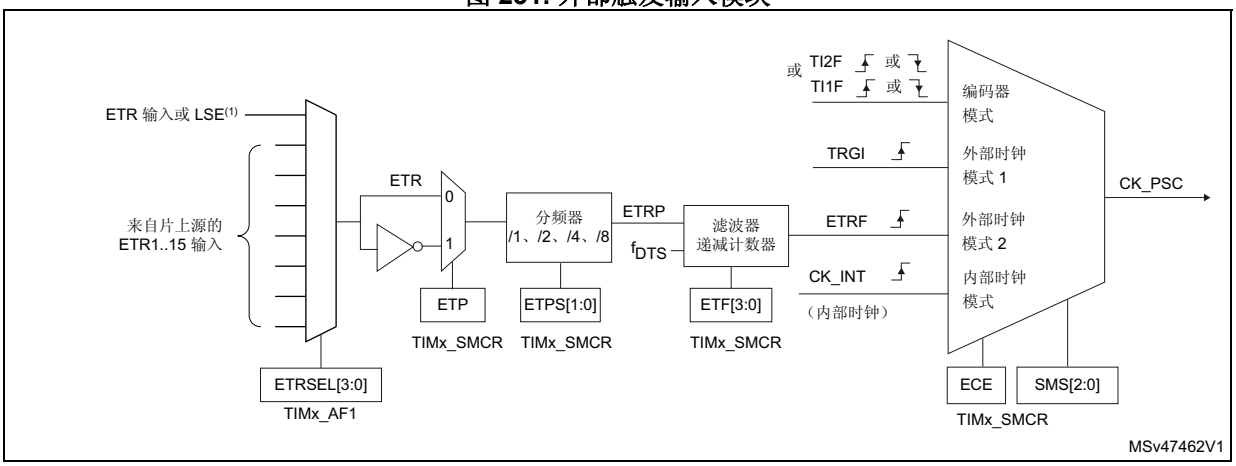
外部时钟源模式 2

通过在 TIMx_SMCR 寄存器中写入 ECE=1 可选择此模式。

计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

图 231 简要介绍了外部触发输入模块。

图 231. 外部触发输入模块



1. 根据 ETR1_RMP 位编程。

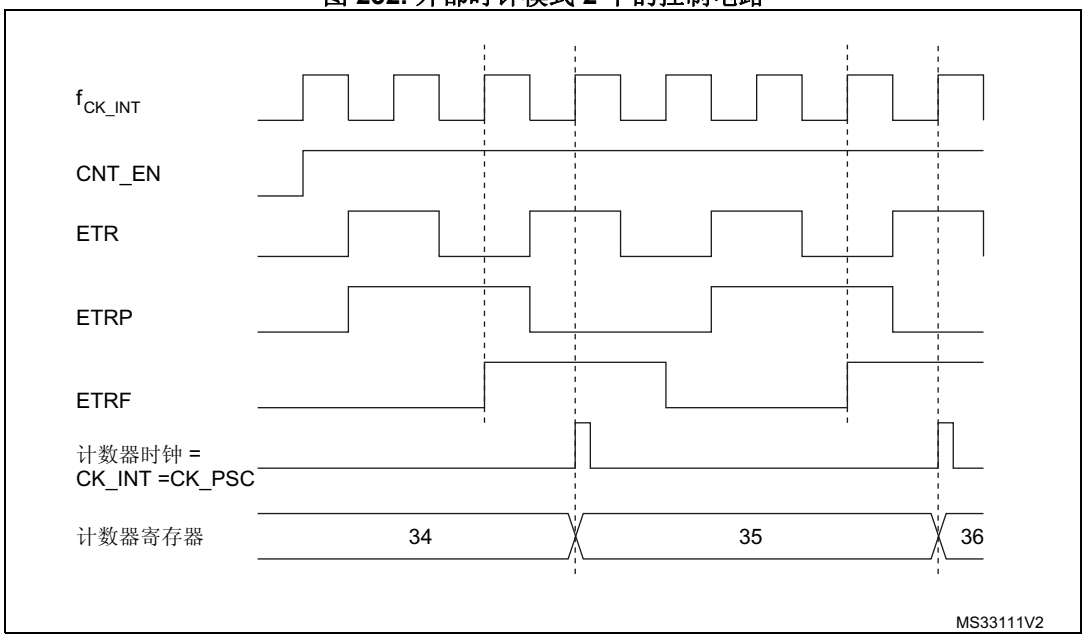
例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 根据 TIMx_AF1 寄存器中的 ETRSEL[3:0] 位和 TIM2_OR1 寄存器中的 ETR1_RMP 位选择正确的 ETR 源（内部或外部）。
2. 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF[3:0]=0000。
3. 通过在 TIMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
4. 通过在 TIMx_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
5. 通过在 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
6. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的。

图 232. 外部时钟模式 2 下的控制电路



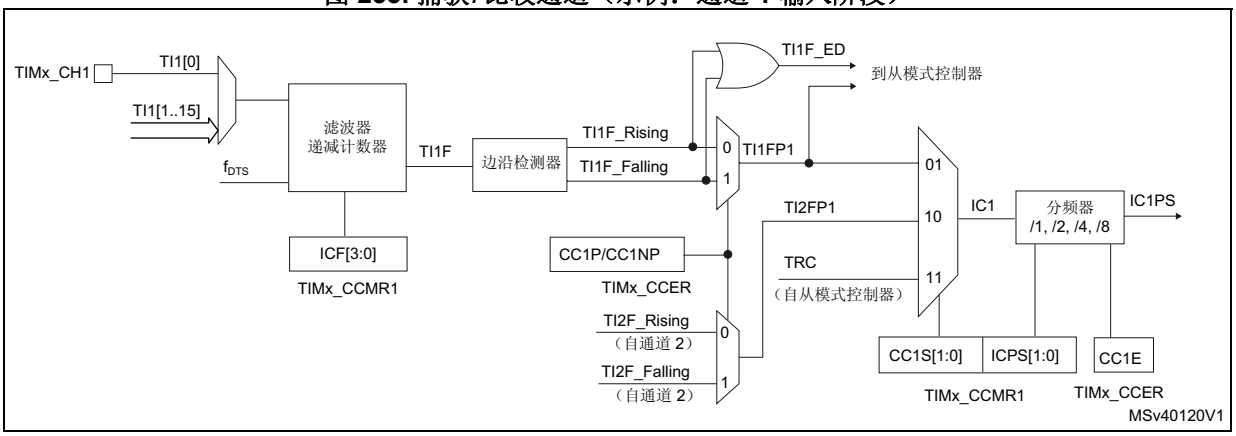
25.3.4 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

下图简要介绍了一路捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 233. 捕获/比较通道（示例：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 234. 捕获/比较通道 1 主电路

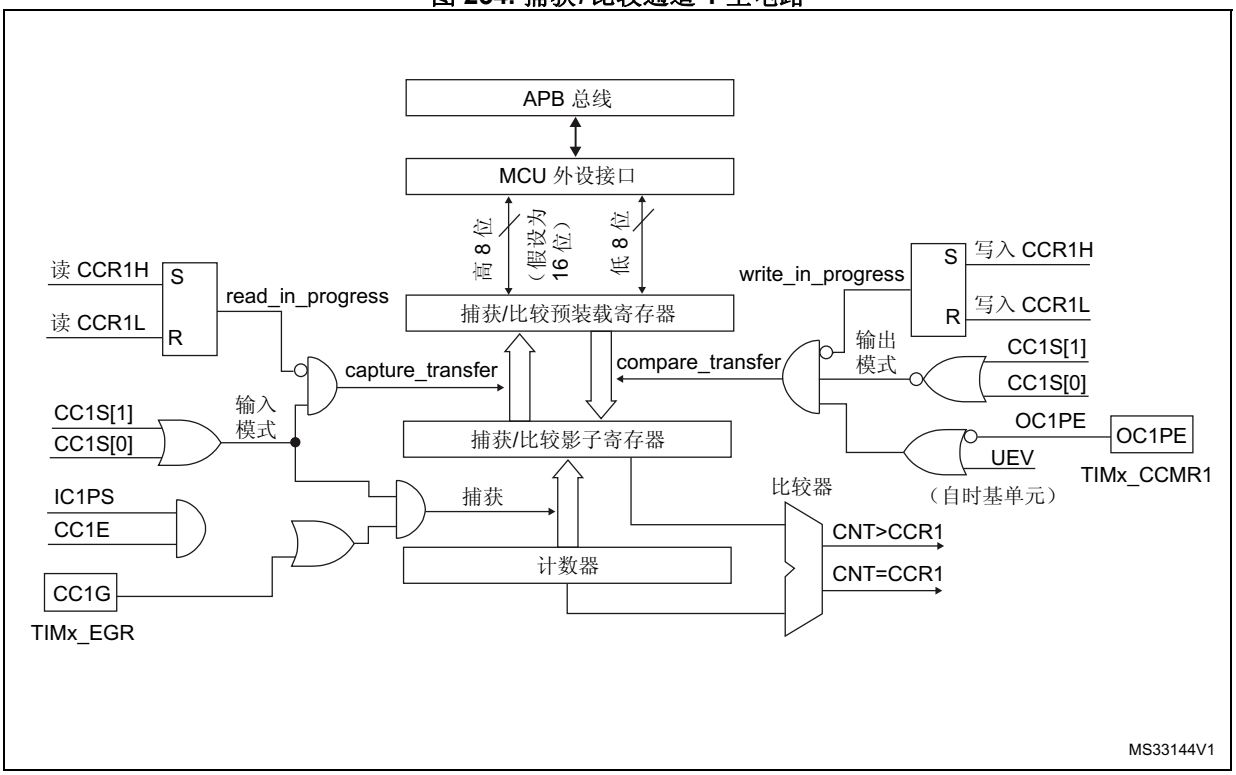
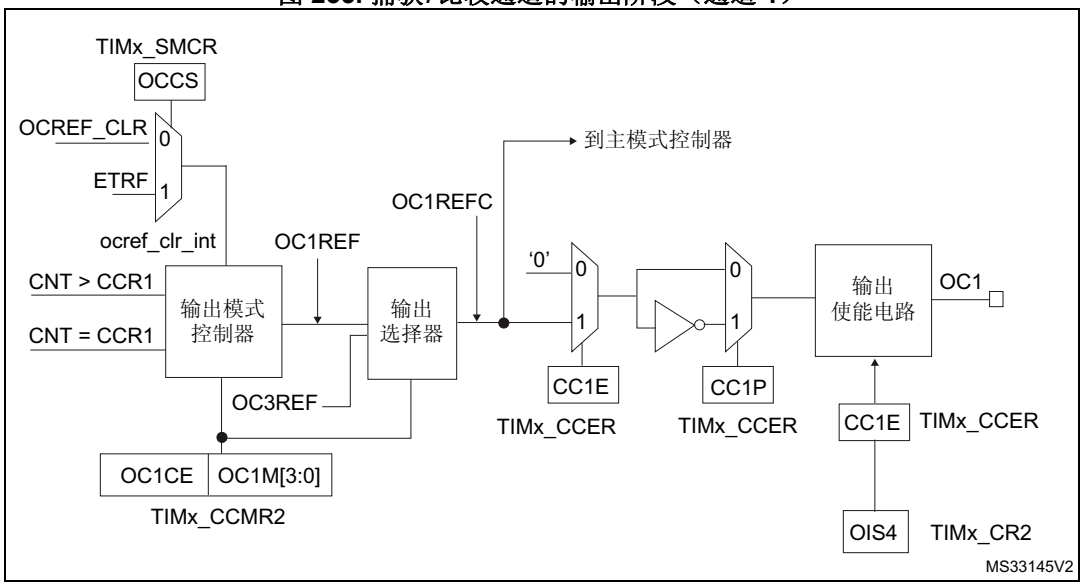


图 235. 捕获/比较通道的输出阶段（通道 1）



捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容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将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

25.3.5 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件方法向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据连接到定时器的信号，对相关输入滤波带宽进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号边沿变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波带宽设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 f_{DTS} 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。

4. 通过向 TIMx_CCER 寄存器中的 CC1P 位和 CC1NP 位写入 000，选择 TI1 通道的有效跳变沿（本例中为上升沿）。
5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入 00）。
6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

注： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

25.3.6 PWM 输入模式

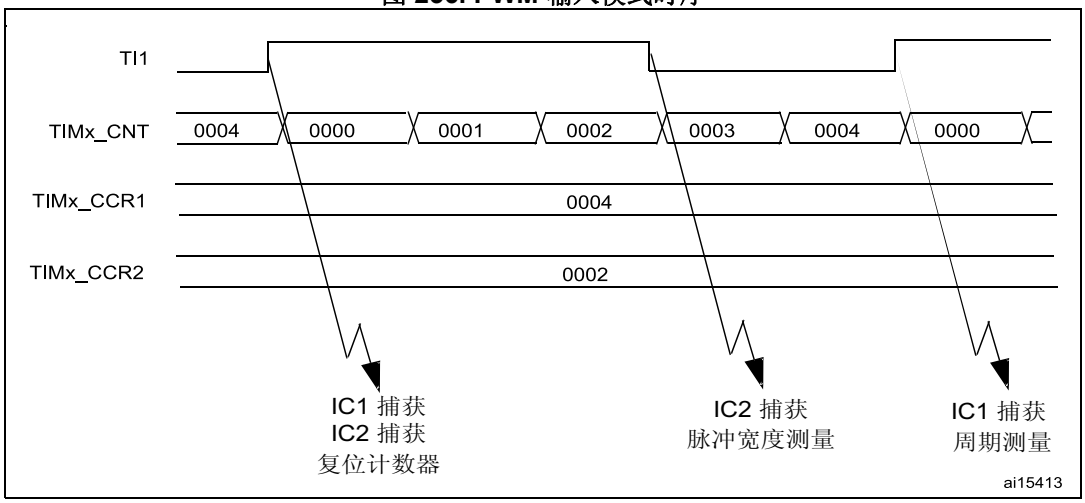
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
3. 选择 TI1FP1 的有效极性（同时用于 TIMx_CCR1 中的捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
4. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
5. 选择 TI1FP2 的有效极性（用于 TIMx_CCR2 中的捕获）：向 CC2P 位写入“1”，向 CC2NP 位写入“0”（下降沿有效）。
6. 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 00101（选择 TI1FP1）。
7. 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 100。
8. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 236. PWM 输入模式时序



1. PWM 输入模式只能与 TIMx_CH1/TIMx_CH2 信号配合使用，因为只有 TI1FP1 和 TI2FP2 与从模式控制器相连。

25.3.7 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (ocxref/OCx) 强制设置为有效电平，只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 101。ocxref 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=0（OCx 高电平有效）=> OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 100，可将 ocxref 信号强制设置为低电平。

无论如何，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。输出比较模式一节对此进行了介绍。

25.3.8 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCXM=000)，也可设置为有效电平 (OCXM=001)、无效电平 (OCXM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

使用 TIMx_CCMRx 寄存器中的 OCxPE 位，可将 TIMx_CCRx 寄存器配置为带或不带预装载寄存器。

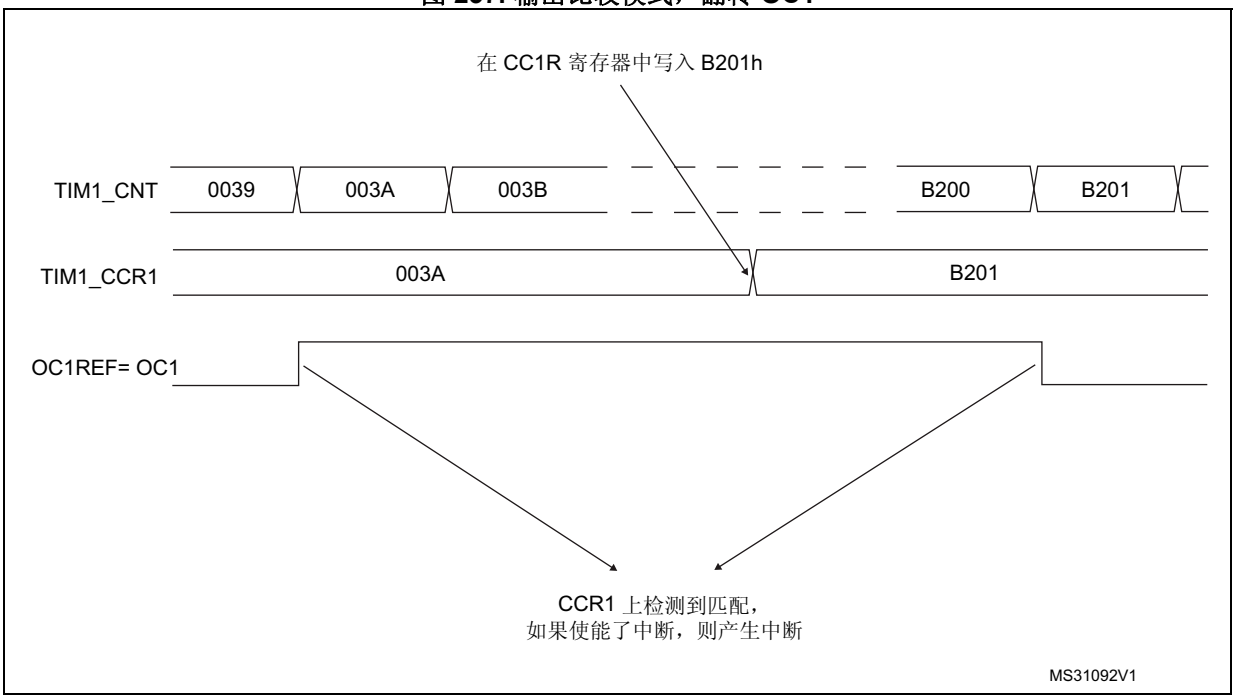
在输出比较模式下，更新事件 UEV 对 ocxref 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断和/或 DMA 请求，将 CCxIE 位和/或 CCxDE 位置 1。
4. 选择输出模式。例如，当 CNT 与 CCRx 匹配、未使用预装载 CCRx 并且 OCx 使能且为高电平有效时，必须写入 OCxM=011、OCxPE=0、CCxP=0 和 CCxE=1 来翻转 OCx 输出引脚。
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIMx_CCRx 影子寄存器）。*图 237* 给出了一个示例。

图 237. 输出比较模式，翻转 OC1



25.3.9 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 110（PWM 模式 1）或 111（PWM 模式 2），可以独立选择各通道（每个 OCx 输出对应一个 PWM）的 PWM 模式。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

由于只有在发生更新事件时预装载寄存器才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。OCx 输出通过将 TIMx_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 TIMx_CCERx 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 总是与 TIMx_CCRx 进行比较，以确定是 TIMx_CCRx ≤ TIMx_CNT 还是 TIMx_CNT ≤ TIMx_CCRx（取决于计数器计数方向）。不过，为符合 OCREF_CLR 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 OCREF），OCREF 信号仅在以下情况下变为有效状态：

- 比较结果发生改变，或
- 输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）从“冻结”配置（不进行比较，OCxM=“000”）切换为任一 PWM 模式（OCxM=“110”或“111”）。

定时器运行期间，可以通过软件强制 PWM 输出。

根据 TIMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

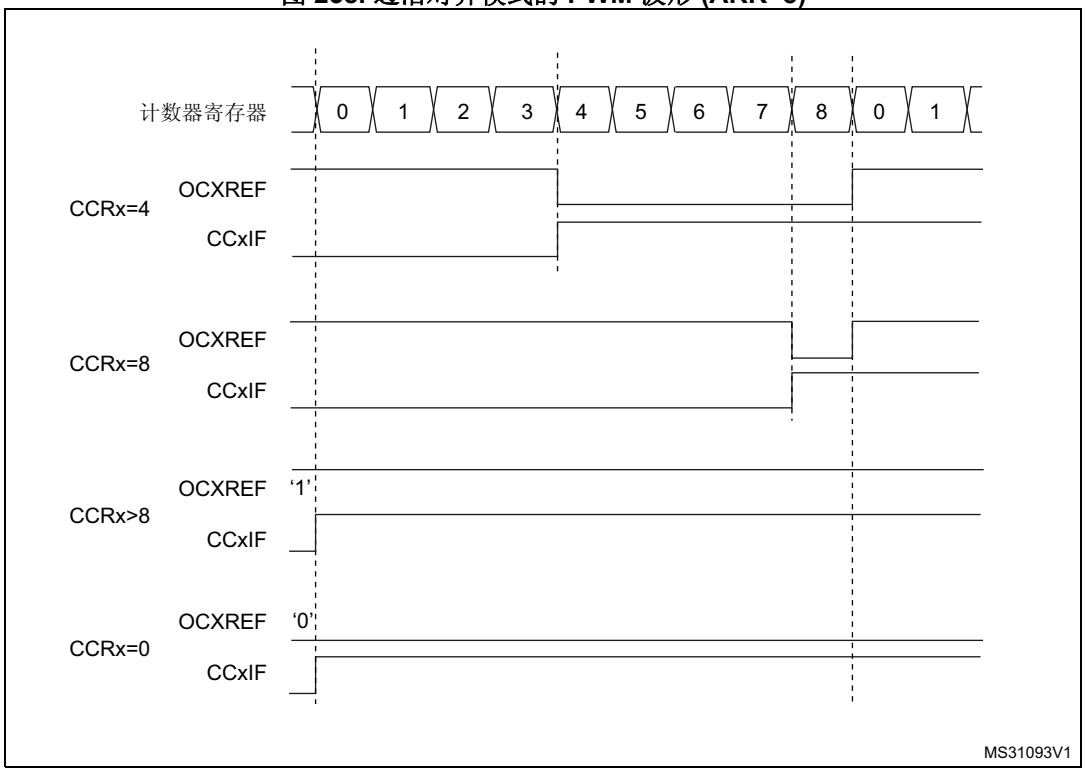
PWM 边沿对齐模式

递增计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低时执行递增计数。请参见*第 746 页的递增计数模式*。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。*图 238* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 238. 边沿对齐模式的 PWM 波形 (ARR=8)



递减计数配置

当 TIMx_CR1 寄存器中的 DIR 位为高时执行递减计数。请参见*第 749 页的递减计数模式*。

在 PWM 模式 1 下，只要 TIMx_CNT>TIMx_CCRx，参考信号 ocxref 便为低电平，否则为高电平。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重载值，则 ocxref 保持为 100%。此模式下不可能产生 PWM。

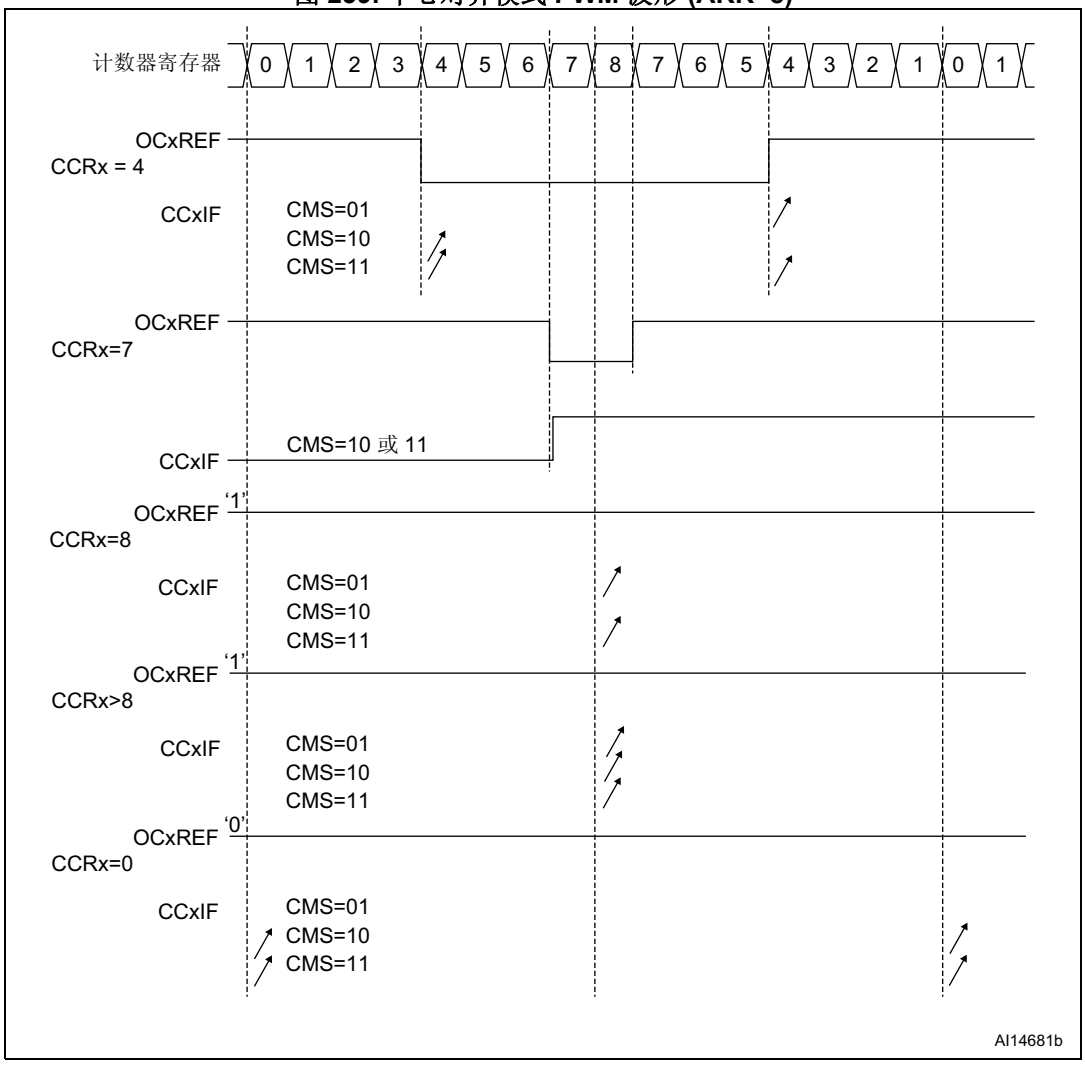
PWM 中心对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时（其余所有配置对 ocxref/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIMx_CR1 寄存器中的方向位 (DIR) 由硬件更新，不得通过软件更改。请参见*第 752 页的中心对齐模式（递增/递减计数）*。

图 239 显示了中心对齐模式的 PWM 波形，在此例中：

- TIMx_ARR=8。
- PWM 模式为 PWM 模式 1。
- 在根据 TIMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

图 239. 中心对齐模式 PWM 波形 (ARR=8)



中心对齐模式使用建议：

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：
  - 如果写入计数器的值大于自动重载值 (TIMx_CNT>TIMx_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
  - 如果向计数器写入 0 或 TIMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

25.3.10 不对称 PWM 模式

在不对称模式下，生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和相移则由一对 TIMx_CCRx 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM，这样每半个 PWM 周期便会调节一次 PWM：

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

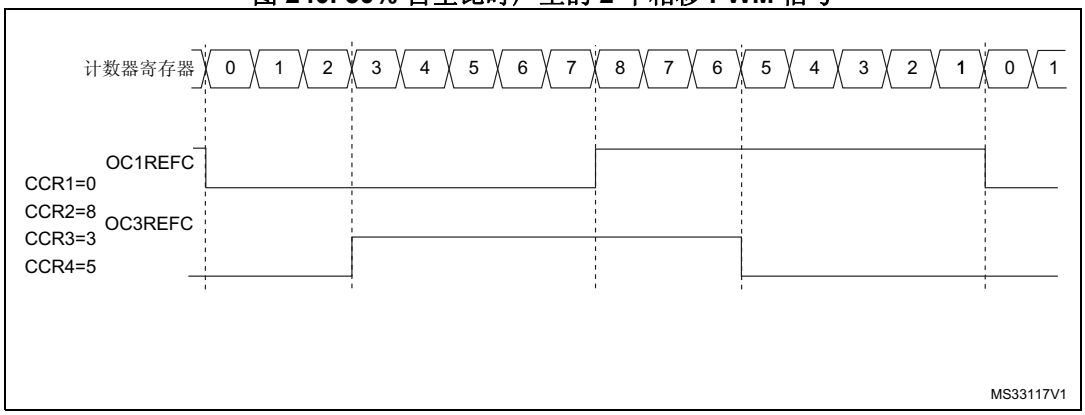
两个通道可以独立选择不对称 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1110”（不对称 PWM 模式 1）或“1111”（不对称 PWM 模式 2）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

给定通道用作不对称 PWM 通道时，也可使用其辅助通道。例如，如果通道 1 上产生 OC1REFC 信号（不对称 PWM 模式 1），则由于不对称 PWM 模式 2 的原因，通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

图 240 显示了不对称 PWM 模式下可以产生的信号示例（通道 1 到通道 4 在不对称 PWM 模式 1 下配置）。

图 240. 50% 占空比时产生的 2 个相移 PWM 信号



25.3.11 组合 PWM 模式

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的逻辑或运算或者逻辑与运算组合组成。

——OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制

——OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

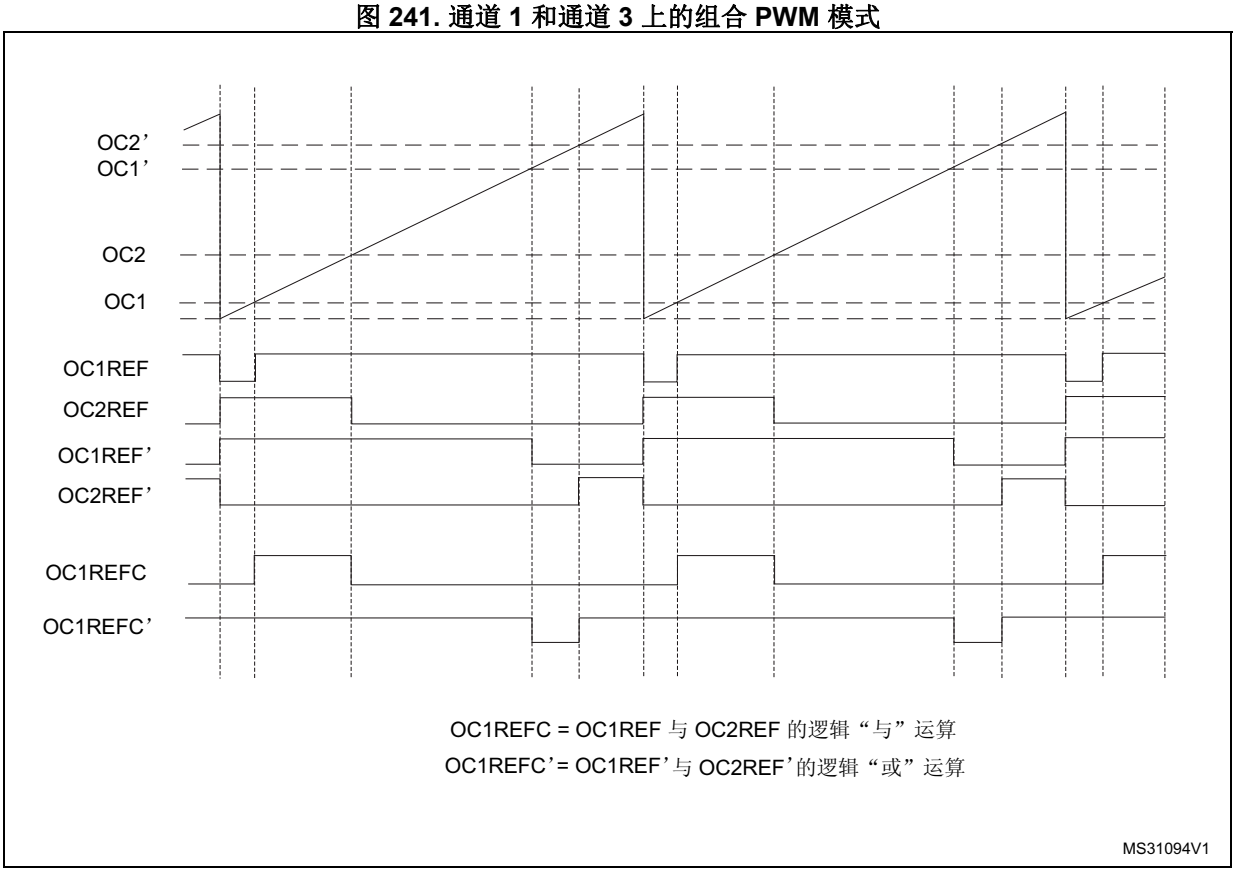
当给定通道用作组合 PWM 通道时，其辅助通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 241 显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 在组合 PWM 模式 2 下配置。
- 通道 2 在 PWM 模式 1 下配置。
- 通道 3 在组合 PWM 模式 2 下配置。
- 通道 4 在 PWM 模式 1 下配置。

图 241. 通道 1 和通道 3 上的组合 PWM 模式



25.3.12 发生外部事件时清除 OCxREF 信号

对于给定通道，在 ocref_clr_int 输入上施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号清零。OCxREF 信号将保持低电平，直到发生下一更新事件 (UEV)。该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。

通过配置 TIMx_SMCR 寄存器中的 OCCS 位，可在 OCREF_CLR 输入和 ETRF（滤波器后的 ETR）之间选择 OCREF_CLR_INPUT。

对于给定通道，在 ETRF 输入施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号复位。OCxREF 信号将保持低电平，直到发生下一更新事件 (UEV)。

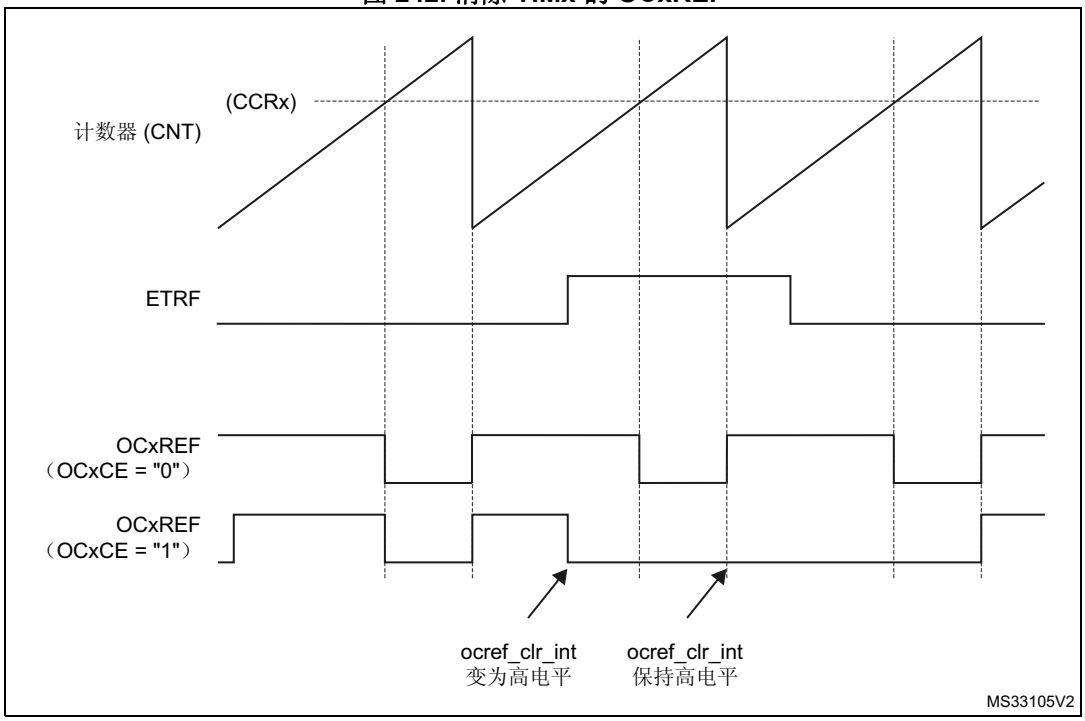
该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。

例如，OCxREF 信号可以连接到比较器的输出，用于控制电流。此时，ETR 必须配置如下：

1. 必须关闭外部触发预分频器：TIMx_SMCR 寄存器中的 ETPS[1:0] 位清为 00。
2. 必须禁止外部时钟模式 2：TIM1_SMCR 寄存器中的 ECE 位清为 0。
3. 外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可根据应用需要进行配置。

图 242 对比了不同使能位 OCxCE 值的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 TIMx 编程为 PWM 模式。

图 242. 清除 TIMx 的 OCxREF



注： *如果 PWM 的占空比为 100% (CCRx>ARR)，则下次计数器上溢时会再次使能 OCxREF。*

25.3.13 单脉冲模式

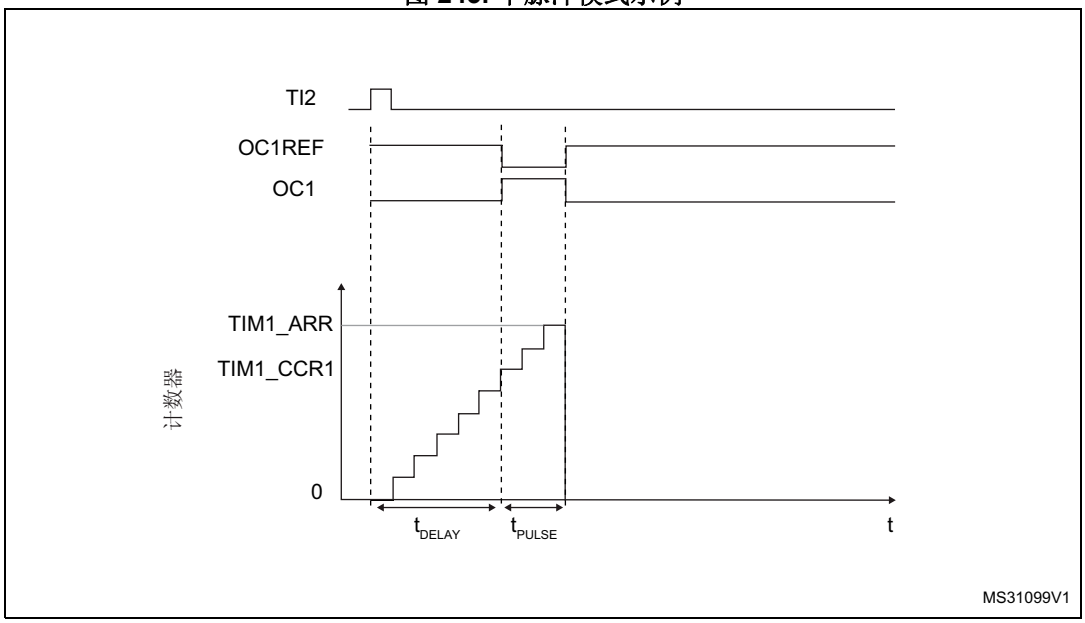
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。通过将 TIMx_CR1 寄存器中的 OPM 位置 1 选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- CNT<CCRx ≤ ARR（特别注意，0 < CCRx）

图 243. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到正沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2x 源（内部或外部）。
2. 在 TIMx_CCMR1 寄存器中写入 CC2S = 01，将 TI2FP2 映射到 TI2。
3. 在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
4. 在 TIMx_SMCR 寄存器中写入 TS=00110，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
5. 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从 “0” 变为 “1”，在计数器达到自动重载值时由 “1” 变为 “0”。为此，必须通过在 TIMx_CCMR1 寄存器中写入 OC1M=111，来使能 PWM 模式 2。可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入 “1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中， CC1P 的值为 “0”。

在本例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为低电平。

由于仅需要 1 个脉冲（单脉冲模式），因此必须向 TIMx_CR1 寄存器的 OPM 位写入 1，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置 “0” 时，即选择重复模式。

特殊情况：OCx 快速使能：

在单脉冲模式下，TIx 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIMx_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxRef（和 OCx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

25.3.14 可再触发单脉冲模式 (OPM)

该模式允许计数器可以在一个激励信号的触发下启动，并且能产生长度可编程的脉冲，但与不可再触发单脉冲模式间存在以下差别，如*第 25.3.13 节*所述：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] = “1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为 “1000” 或 “1001”。

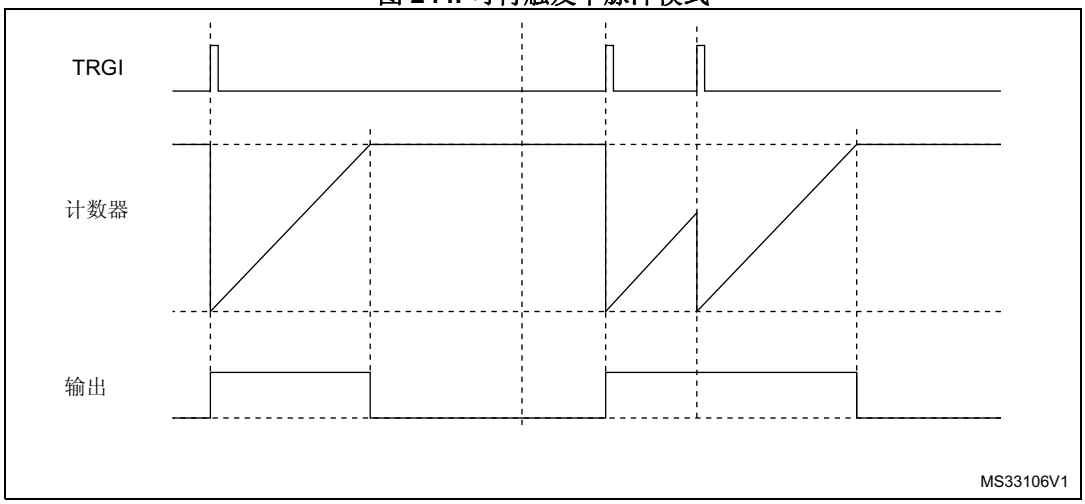
定时器配置为递增计数模式时，相应的 CCRx 必须置 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCRx 必须高于或等于 ARR。

注： *在可再触发单脉冲模式下，CCxIF 标志没有意义。*

出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 244. 可再触发单脉冲模式



25.3.15 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 TIMx_SMCR 寄存器中写入 SMS=001；如果计数器仅在 TI1 边沿处计数，写入 SMS=010；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS=011。

通过编程 TIMx_CCER 寄存器的 CC1P 和 CC2P 位，选择 TI1 和 TI2 极性。CC1NP 和 CC2NP 必须保持清零。如果需要，还可对输入滤波器进行编程。CC1NP 和 CC2NP 必须保持低电平。

TI1 和 TI2 两个输入用于连接增量编码器。请参见 *表 157*。如果使能计数器（在 TIMx_CR1 寄存器的 CEN 位中写入“1”），则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。TI1FP1 和 TI2FP2 是进行输入滤波器和极性选择后 TI1 和 TI2 的信号，如果不进行滤波和反相，则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1 或 TI2 边沿处计数，还是同时在 TI1 和 TI2 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIMx_ARR。同样，捕获、比较、预分频器、触发输出功能继续正常工作。

在此模式下，计数器会根据正交编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI1 和 TI2 不同时切换）。

表 157. 计数方向与编码器信号的关系

有效边沿	相反信号的电平（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 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

外部增量编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码器输出可与外部中断输入相连，用以触发计数器复位。

图 245 以计数器工作为例，说明了计数信号的生成和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S=01（TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）
- CC2S=01（TIMx_CCMR2 寄存器，TI2FP2 映射到 TI2 上）
- CC1P 和 CC1NP = “0”（TIMx_CCER 寄存器，TI1FP1 未反相，TI1FP1=TI1）
- CC2P 和 CC2NP = “0”（TIMx_CCER 寄存器，TI2FP2 未反相，TI2FP2= TI2）
- SMS=011（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）
- CEN = 1（TIMx_CR1 寄存器，使能计数器）

图 245. 编码器接口模式下的计数器工作示例

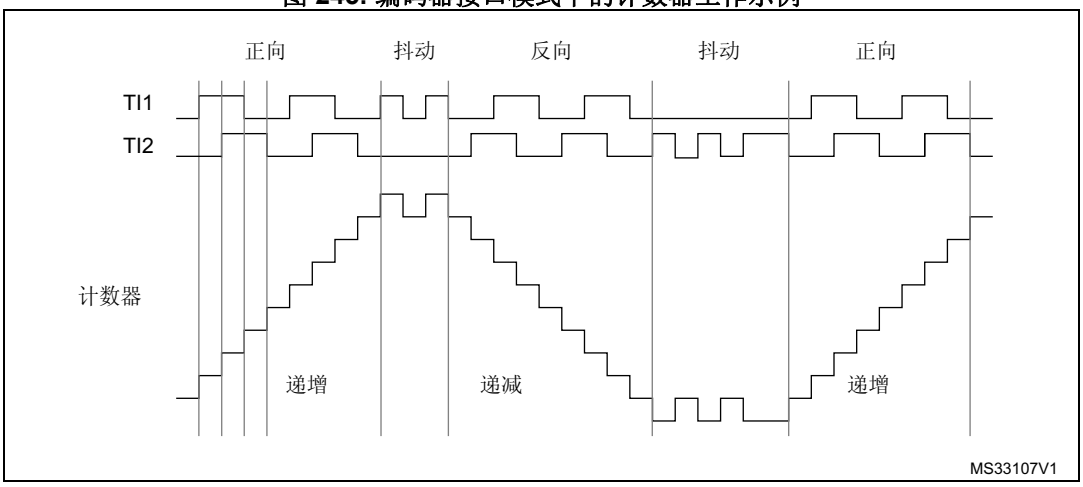
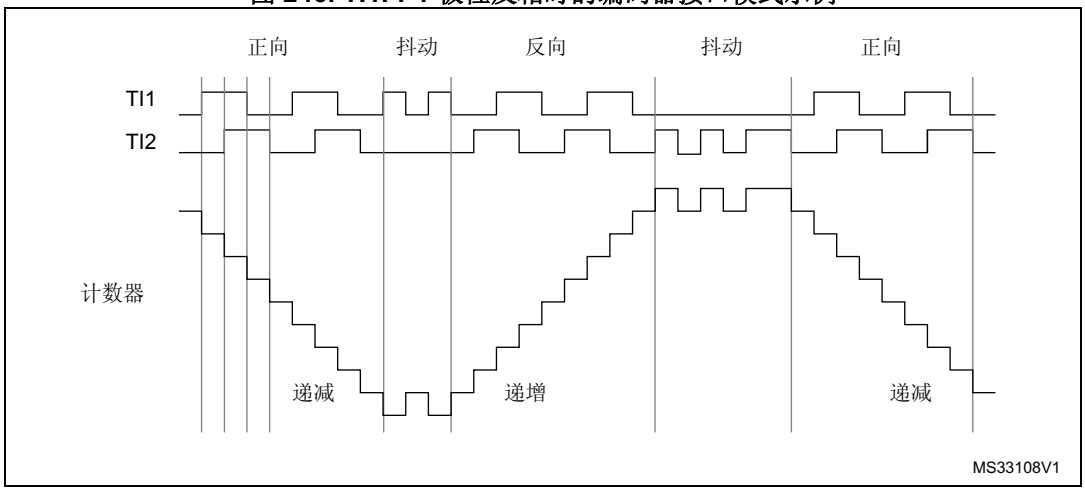


图 246 举例说明了 TI1FP1 极性反相时的计数器行为（除 CC1P=1 外，其他配置与上例相同）。

图 246. TI1FP1 极性反相时的编码器接口模式示例



定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）。此外，还可以通过实时时钟产生的 DMA 请求读取计数器值。

25.3.16 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 (UIF) 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化角速度的计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

在 32 位定时器实现中，当 IUFREMAP 位置 1 时，计数器的位 31 在读访问时由 UIFCPY 标志覆盖（计数器的最高有效位只能在写模式下访问）。

25.3.17 定时器输入异或功能

借助 TIM1xx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 到 TIMx_CH3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。

*第 696 页的第 24.3.25 节：连接霍尔传感器*以连接霍尔传感器为例介绍了此功能。

25.3.18 定时器与外部触发同步

TIMx 定时器可与外部触发以下列模式实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

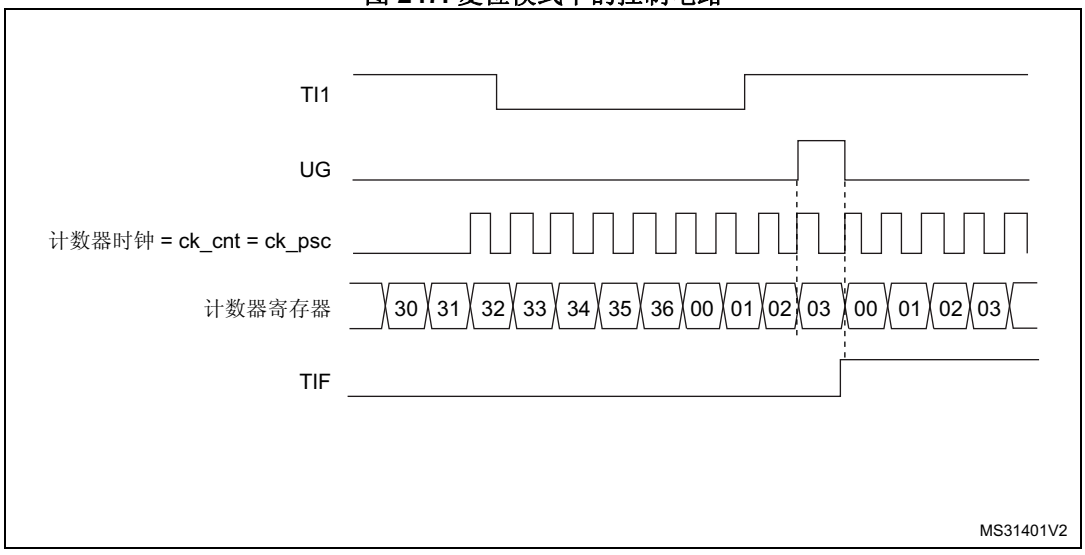
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

1. 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，验证极性（仅检测上升沿）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断，还可发送中断请求（取决于 TIMx_DIER 寄存器中的 TIE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 247. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

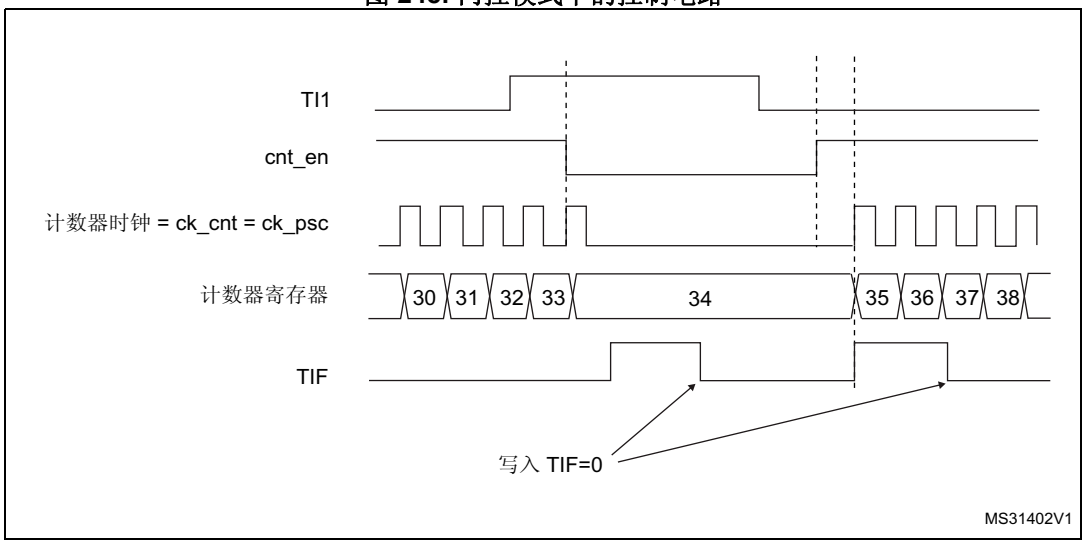
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

1. 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1 和 CC1NP=0，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 248. 门控模式下的控制电路



1. 由于门控模式作用于电平而非边沿，因此在门控模式下，“CCxP=CCxNP=1”（同时检测上升沿和下降沿）不发挥任何作用。

从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

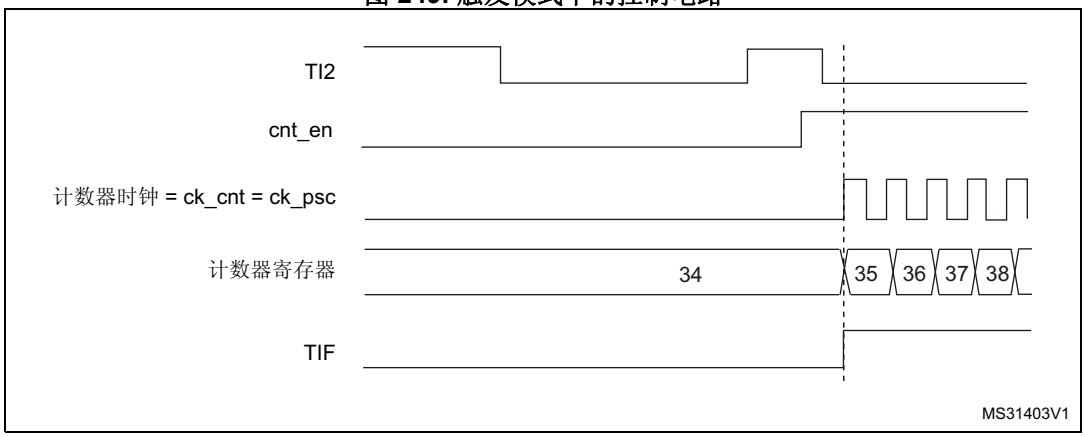
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

1. 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00110，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 249. 触发模式下的控制电路



从模式：外部时钟模式 2 + 触发模式

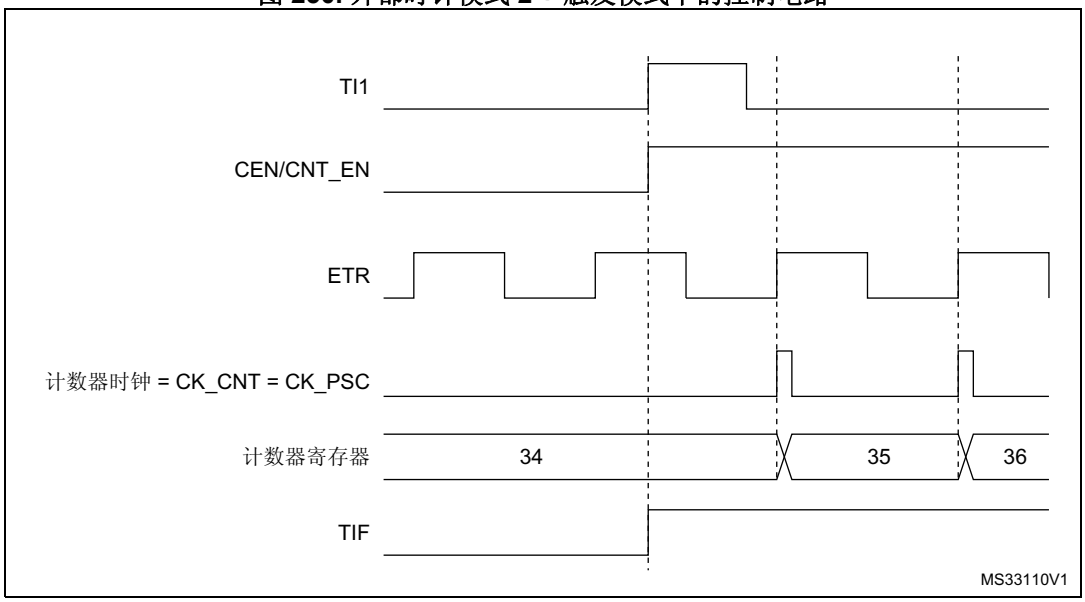
外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

1. 通过对 TIMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
   - ETF = 0000：无滤波器。
   - ETPS = 00：禁止预分频器。
   - ETP = 0：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。
2. 如下配置通道 1，以检测 TI 的上升沿：
   - IC1F=0000：无滤波器。
   - 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
   - TIMx_CCMR1 寄存器中 CC1S=01，只选择输入捕获源。
   - TIMx_CCER 寄存器中 CC1P=0 且 CC1NP= 0，以确定极性（仅检测上升沿）。
3. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。

图 250. 外部时钟模式 2 + 触发模式下的控制电路

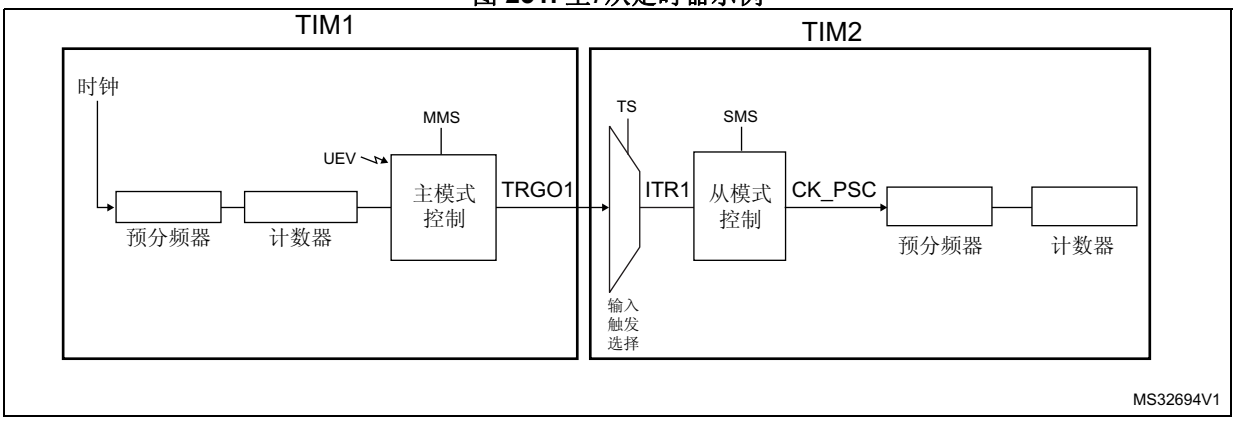


25.3.19 定时器同步

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

*图 251：主/从定时器示例*简要介绍了触发选择和主模式选择框图。

图 251. 主/从定时器示例



将一个定时器用作另一个定时器的预分频器

例如，可以将 TIM1 配置为 TIM2 的预分频器。请参见*图 251*。为此：

1. 将 TIM1 配置为主模式，以便每次发生更新事件 UEV 时都输出一个周期性触发信号。如果在 TIM1_CR2 寄存器中写入 MMS=010，则每次生成更新事件时，TRGO 都会输出一个上升沿。
2. 要将 TIM1 的 TRGO 输出连接到 TIM2，必须将 TIM2 配置为从模式，使用 ITR0 作为内部触发。通过 TIM2_SMCR 寄存器中的 TS 位（写入 TS=00000）可对此进行选择。
3. 然后必须将从模式控制器设为外部时钟模式 1（在 TIM2_SMCR 寄存器中写入 SMS=111）。这样一来，TIM2 的时钟将由 TIM1 周期性触发信号的上升沿（与 TIM1 的计数器上溢对应）提供。
4. 最后必须通过将这两个定时器的相应 CEN 位（TIMx_CR1 寄存器）置 1 同时使能二者。

注： *如果选择 TIM1 的 OCx 信号作为触发输出 (MMS=1xx)，该信号的上升沿将用于驱动 TIM2 的计数器。*

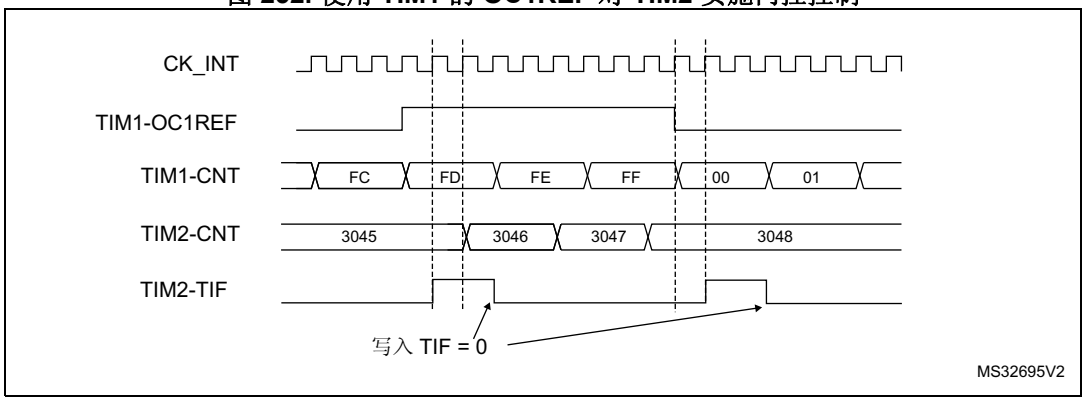
使用一个定时器使能另一个定时器

本例中通过定时器 1 的输出比较 1 来控制 TIM2 的使能。相关连接图，请参见*图 251*。仅当 TIM1 的 OC1REF 为高电平时，TIM2 才根据分频后的内部时钟进行计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 ($f_{CK\_CNT} = f_{CK\_INT}/3$)。

1. 将 TIM1 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出（TIM1_CR2 寄存器中的 MMS=100）。
2. 配置 TIM1 的 OC1REF 波形（TIM1_CCMR1 寄存器）。
3. 配置 TIM2 以接收来自 TIM1 的输入触发（TIM2_SMCR 寄存器中的 TS=00000）。
4. 将 TIM2 配置为门控模式（TIM2_SMCR 寄存器中的 SMS=101）。
5. 通过向 CEN 位（TIM2_CR1 寄存器）写入“1”使能 TIM2。
6. 通过向 CEN 位（TIM1_CR1 寄存器）写入“1”启动 TIM1。

注： *计数器 2 的时钟与计数器 1 不同步，此模式仅影响 TIM2 的计数器使能信号。*

图 252. 使用 TIM1 的 OC1REF 对 TIM2 实施门控控制

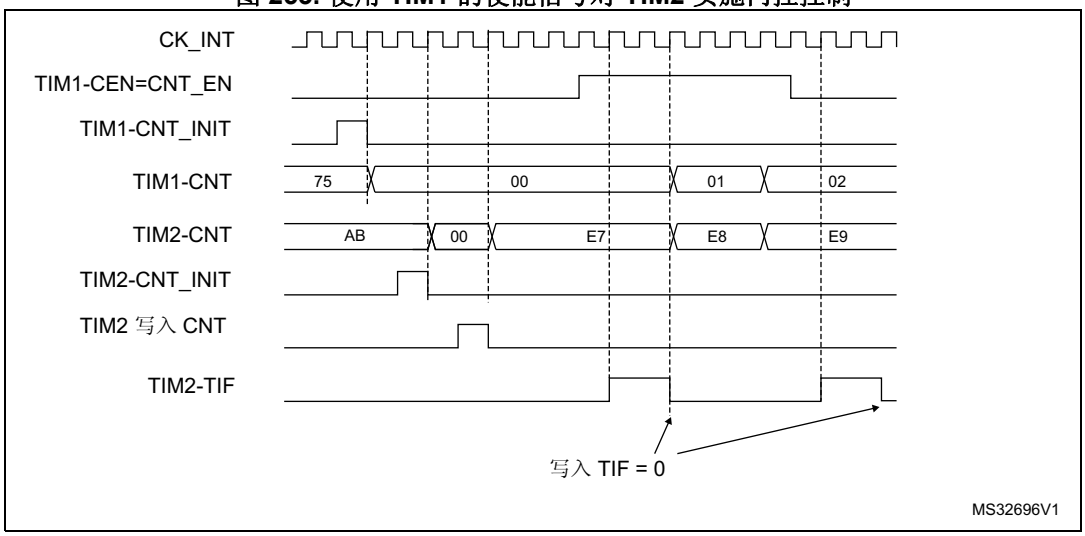


在*图 252* 的示例中，TIM2 的计数器和预分频器在启动前未进行初始化。因此从各自的当前值开始计数。启动定时器 TIM1 之前，通过复位这两个定时器可以从指定值开始计数。然后便可以在定时器计数器中写入任意值。两个定时器都可通过软件使用 TIMx_EGR 寄存器中的 UG 位轻松复位。

在下一示例中（请参见*图 253*），TIM1 与 TIM2 同步。TIM1 为主模式，从 0 开始计数。TIM2 为从模式，从 0xE7 开始计数。两个定时器的预分频比相同。通过向 TIM1_CR1 寄存器中的 CEN 位写入“0”来禁止 TIM1 时，TIM2 将停止：

1. 将 TIM1 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出（TIM1_CR2 寄存器中的 MMS=100）。
2. 配置 TIM1 的 OC1REF 波形（TIM1_CCMR1 寄存器）。
3. 配置 TIM2 以接收来自 TIM1 的输入触发（TIM2_SMCR 寄存器中的 TS=00000）。
4. 将 TIM2 配置为门控模式（TIM2_SMCR 寄存器中的 SMS=101）。
5. 通过向 UG 位（TIM1_EGR 寄存器）写入“1”复位 TIM1。
6. 通过向 UG 位（TIM2_EGR 寄存器）写入“1”复位 TIM2。
7. 通过在 TIM2 的计数器 (TIM2_CNTL) 中写入“0xE7”使 TIM2 初始化为 0xE7。
8. 通过向 CEN 位（TIM2_CR1 寄存器）写入“1”使能 TIM2。
9. 通过向 CEN 位（TIM1_CR1 寄存器）写入“1”启动 TIM1。
10. 通过向 CEN 位（TIM1_CR1 寄存器）写入“0”停止 TIM1。

图 253. 使用 TIM1 的使能信号对 TIM2 实施门控控制

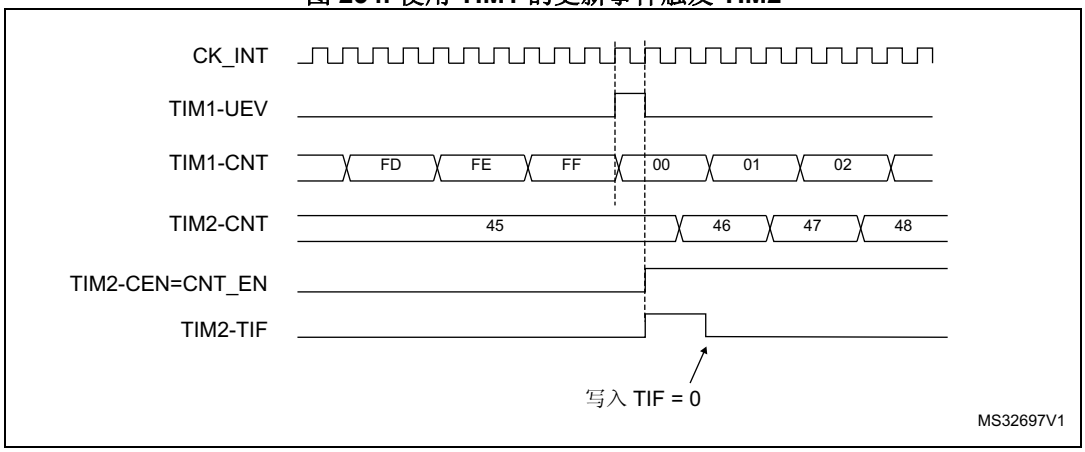


使用一个定时器启动另一个定时器

本例中使用定时器 1 的更新事件使能定时器 2。相关连接图，请参见*图 251*。只要定时器 1 生成更新事件，定时器 2 便根据分频后的内部时钟从当前值（可以不为 0）开始计数。定时器 2 收到触发信号时，其 CEN 位自动置 1，并且计数器开始计数，直到向 TIM2_CR1 寄存器的 CEN 位写入“0”后停止计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 ($f_{CK\_CNT} = f_{CK\_INT}/3$)。

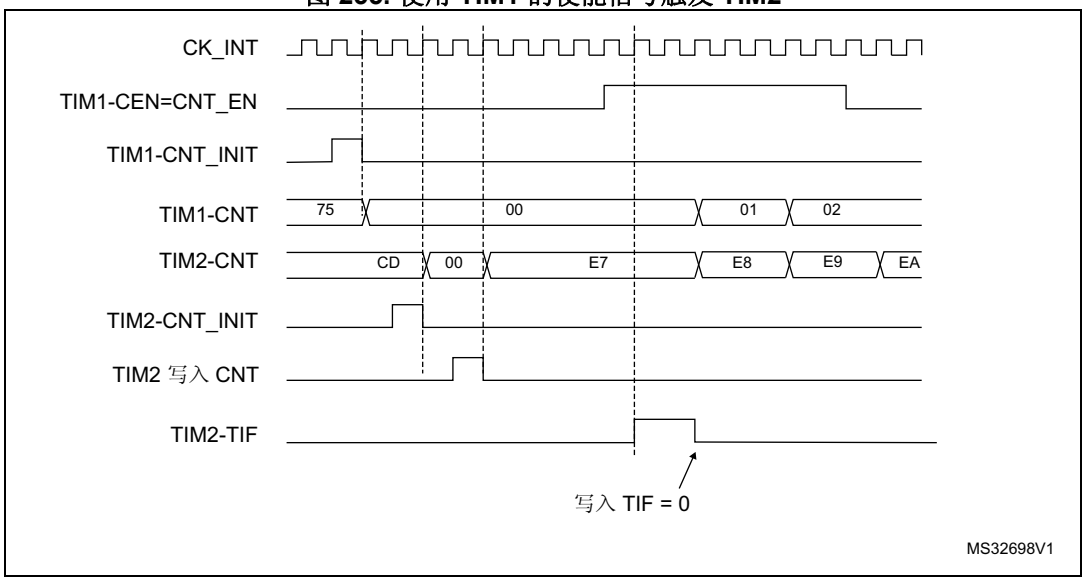
1. 将 TIM1 配置为主模式，发送其更新事件 (UEV) 作为触发输出（TIM1_CR2 寄存器中的 MMS=010）。
2. 配置 TIM1 的周期（TIM1_ARR 寄存器）。
3. 配置 TIM2 以接收来自 TIM1 的输入触发（TIM2_SMCR 寄存器中的 TS=00000）。
4. 将 TIM2 配置为触发模式（TIM2_SMCR 寄存器中的 SMS=110）。
5. 通过向 CEN 位（TIM1_CR1 寄存器）写入“1”启动 TIM1。

图 254. 使用 TIM1 的更新事件触发 TIM2



如上述示例所示，用户可以在开始计数之前初始化两个计数器。*图 255* 显示了与*图 254* 具有相同配置，只不过处于触发模式（TIM2_SMCR 寄存器中的 SMS=110）而非门控模式的计数行为。

图 255. 使用 TIM1 的使能信号触发 TIM2



注： *必须先使能接收 TRGO 信号的从外设（定时器、ADC 等），才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。*

25.3.20 DMA 连续传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的一部分多次重新编程而无需软件开销，但也可用于定期读取一行中的多个寄存器。

DMA 控制器目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（突发）。每次写入 TIMx_DMAR 寄存器都会重定向到其中一个定时器寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 连续传送长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送，即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量：

示例：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   - DMA 通道外设地址为 DMAR 寄存器地址。
   - DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   - 要传输的数据量 = 3（参见下文注释）。
   - 禁止循环模式。
2. 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：
   DBL = 3 次传输，DBA = 0xE。
3. 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx。
5. 使能 DMA 通道。

本例适用于每个 CCRx 寄存器必须更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。

注： *可以将空值写入保留的寄存器中。*

25.3.21 调试模式

当系统进入调试模式（处理器内核停止）时，TIMx 计数器会根据 DBGMCU 模块中的 DBG_TIM2_STOP 配置位选择继续正常工作或者停止工作。有关详细信息，请参见 *第 41.8.3 节：DBGMCU CPU1 APB1 外设冻结寄存器 1 (DBGMCU_APB1FZR1)*。

25.4 TIM2 寄存器

在本章中，应将“TIMx”理解为“TIM2”，因为本参考手册所适用的产品中只有一个此类定时器的实例。

有关寄存器说明中使用的缩写，请参见*第 1.2 节*。

外设寄存器可支持半字（16 位）或字（32 位）访问。

25.4.1 TIM2 控制寄存器 1 (TIM2_CR1)

TIM2 control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
				rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)
此位域指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟（ETR、TIx）之间的分频比
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

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:5 **CMS[1:0]**：中心对齐模式选择 (Center-aligned mode selection)
00：边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。
01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。
10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。
11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志都会置 1。
注：　只要计数器处于使能状态 (CEN=1)，就不得从边沿对齐模式切换为中心对齐模式。

位 4 **DIR**：方向 (Direction)

0：计数器递增计数

1：计数器递减计数

注： 当定时器配置为中心对齐模式或编码器模式时，此位为只读状态。

位 3 **OPM**：单脉冲模式 (One-pulse mode)

0：计数器在发生更新事件时不会停止计数

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 **URS**：更新请求源 (Update request source)

此位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会产生更新中断或 DMA 请求。此类事件包括：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 **UDIS**：更新禁止 (Update disable)

此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

然后更新影子寄存器的值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注： 只有事先通过软件将 CEN 位置 1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。

在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。

25.4.2 TIM2 控制寄存器 2 (TIM2_CR2)

TIM2 control register 2

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			CCDS	Res.	Res.	Res.
								rw	rw	rw	rw	rw			

位 15:8 保留，必须保持复位值。

位 7 **TI1S**：TI1 选择 (TI1 selection)

0：TIMx_CH1 引脚连接到 TI1 输入

1：TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合）。另请参见 *第 696 页的第 24.3.25 节：连接霍尔传感器*

位 6:4 **MMS[2:0]**：主模式选择 (Master mode selection)

这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下：

000：**复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。

001：**使能**——计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。

当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。

010：**更新**——选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。

011：**比较脉冲**——一旦发生输入捕获或比较匹配事件，当 CC1IF 标志被置 1 时（即使已为高电平），触发输出都会发送一个正脉冲。(TRGO)

100：**比较**——OC1REF 信号用作触发输出 (TRGO)

101：**比较**——OC2REF 信号用作触发输出 (TRGO)

110：**比较**——OC3REF 信号用作触发输出 (TRGO)

111：**比较**——OC4REF 信号用作触发输出 (TRGO)

注： 必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 3 **CCDS**：捕获/比较 DMA 选择 (Capture/compare DMA selection)

0：发生 CCx 事件时发送 CCx DMA 请求

1：发生更新事件时发送 CCx DMA 请求

位 2:0 保留，必须保持复位值。

25.4.3 TIM2 从模式控制寄存器 (TIM2_SMCR)

TIM2 slave mode control register

偏移地址：0x08

复位值：0x0000 0000

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

位 31:22 保留，必须保持复位值。

位 19:17 保留，必须保持复位值。

位 16 **SMS[3]**：从模式选择——位 3 (Slave mode selection - bit 3)
请参见 SMS 说明——位 2:0

位 15 **ETP**：外部触发极性 (External trigger polarity)
此位可选择将 ETR 还是 $\overline{ETR}$ 用于触发操作
0：ETR 未反相，高电平或上升沿有效
1：ETR 反相，低电平或下降沿有效

位 14 **ECE**：外部时钟使能 (External clock enable)
此位可使能外部时钟模式 2。
0：禁止外部时钟模式 2。
1：使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。
1：将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF（SMS=111 且 TS=00111）具有相同效果。
2：外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF（TS 位不得为 00111）。
3：如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。

位 13:12 **ETPS[1:0]**：外部触发预分频器 (External trigger prescaler)
外部触发信号 ETRP 频率不得超过 CK_INT 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。
00：预分频器关闭
01：2 分频 ETRP 频率
10：4 分频 ETRP 频率
11：8 分频 ETRP 频率

位 11:8 **ETF[3:0]**：外部触发滤波器 (External trigger filter)

此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器的采样长度。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 7 **MSM**：主/从模式 (Master/Slave mode)

0：不执行任何操作
1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

位 21、20、6、5、4 **TS[4:0]**：触发选择 (Trigger selection)（请参见 TS[4:3] 的位 21:20）

此位域可选择将要用于同步计数器的触发输入。

00000：内部触发 0 (ITR0)
00001：内部触发 1 (ITR1)
00010：内部触发 2 (ITR2)
00011：内部触发 3 (ITR3)
00100：TI1 边沿检测器 (TI1F_ED)
00101：滤波后的定时器输入 1 (TI1FP1)
00110：滤波后的定时器输入 2 (TI2FP2)
00111：外部触发输入 (ETRF)
01000：内部触发 4 (ITR4)
01001：内部触发 5 (ITR5)
01010：内部触发 6 (ITR6)
01011：内部触发 7 (ITR7)
01100：内部触发 8 (ITR8)
其他值：保留

有关各定时器 ITRx 含义的详细信息，请参见*第 789 页的表 158：TIM2 内部触发连接*。

注：　这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

位 3 **OCCS**：OCREF 清零选择 (OCREF clear selection)

此位用于选择 OCREF 清零源

0：OCREF_CLR_INT 连接到 OCREF_CLR 输入
1：OCREF_CLR_INT 连接到 ETRF

位 16、2、1、0 **SMS[3:0]**：从模式选择 (Slave mode selection)

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

0000：禁止从模式——如果 CEN = “1”，预分频器时钟直接由内部时钟提供。

0001：编码器模式 1——计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。

0010：编码器模式 2——计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。

0011：编码器模式 3——计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个输入的电平。

0100：复位模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。

0101：门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。

0110：触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。

0111：外部时钟模式 1——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。

1000：组合复位 + 触发模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。

注：　如果将 TI1F_ED 选作触发输入 (TS=00100)，则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

注：　必须先使能接收 TRGO 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

表 158. TIM2 内部触发连接

从 TIM	ITR0	ITR1	ITR2 - ITR8
TIM2	TIM1	USB_FS SOF(1)	-

1. 此连接仅在 TIM2_OR1 寄存器中的 ITR_RMP 位置 1 时有效。

25.4.4 TIM2 DMA/中断使能寄存器 (TIM2_DIER)

TIM2 DMA/Interrupt enable register

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
Res.	Res.	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
			rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

位 15:14 保留，必须保持复位值。

位 13 保留，必须保持复位值。

位 12 **CC4DE**：捕获/比较 4 DMA 请求使能 (Capture/Compare 4 DMA request enable)

0：禁止 CC4 DMA 请求。

1：使能 CC4 DMA 请求。

位 11 **CC3DE**：捕获/比较 3 DMA 请求使能 (Capture/Compare 3 DMA request enable)
0：禁止 CC3 DMA 请求。
1：使能 CC3 DMA 请求。

位 10 **CC2DE**：捕获/比较 2 DMA 请求使能 (Capture/Compare 2 DMA request enable)
0：禁止 CC2 DMA 请求。
1：使能 CC2 DMA 请求。

位 9 **CC1DE**：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)
0：禁止 CC1 DMA 请求。
1：使能 CC1 DMA 请求。

位 8 **UDE**：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。

位 7 保留，必须保持复位值。

位 6 **TIE**：触发中断使能 (Trigger interrupt enable)
0：禁止触发中断。
1：使能触发中断。

位 5 保留，必须保持复位值。

位 4 **CC4IE**：捕获/比较 4 中断使能 (Capture/Compare 4 interrupt enable)
0：禁止 CC4 中断。
1：使能 CC4 中断。

位 3 **CC3IE**：捕获/比较 3 中断使能 (Capture/Compare 3 interrupt enable)
0：禁止 CC3 中断。
1：使能 CC3 中断。

位 2 **CC2IE**：捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable)
0：禁止 CC2 中断。
1：使能 CC2 中断。

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断。
1：使能 CC1 中断。

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断。
1：使能更新中断。

25.4.5 TIM2 状态寄存器 (TIM2_SR)

TIM2 status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 15:13 保留，必须保持复位值。

位 12 **CC4OF**：捕获/比较 4 重复捕获标志 (Capture/Compare 4 overcapture flag)
请参见 CC1OF 说明

位 11 **CC3OF**：捕获/比较 3 重复捕获标志 (Capture/Compare 3 overcapture flag)
请参见 CC1OF 说明

位 10 **CC2OF**：捕获/比较 2 重复捕获标志 (Capture/compare 2 overcapture flag)
请参见 CC1OF 说明

位 9 **CC1OF**：捕获/比较 1 重复捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将此位清零。
0：未检测到重复捕获。
1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8:7 保留，必须保持复位值。

位 6 **TIF**：触发中断标志 (Trigger interrupt flag)
在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。
0：未发生触发事件。
1：触发中断挂起。

位 5 保留，必须保持复位值。

位 4 **CC4IF**：捕获/比较 4 中断标志 (Capture/Compare 4 interrupt flag)
请参见 CC1IF 说明

位 3 **CC3IF**：捕获/比较 3 中断标志 (Capture/Compare 3 interrupt flag)
请参见 CC1IF 说明

位 2 **CC2IF**：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)
请参见 CC1IF 说明

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/compare 1 interrupt flag)
如果通道 CC1 配置为输出：当计数器与比较值匹配时，此标志由硬件置 1，中心对齐模式（请参见 TIMx_CR1 寄存器中的 CMS 位说明）和可再触发单脉冲模式下除外。但需要通过软件清零。
0：不匹配。
1：TIMx_CNT 计数器的内容与 TIMx_CCR1 寄存器的内容匹配。
如果通道 CC1 配置为输入：此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将此位清零。
0：未发生输入捕获事件。
1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）。

位 0 **UIF**：更新中断标志 (Update interrupt flag)
此位在发生更新事件时通过硬件置 1。但需要通过软件清零。
0：未发生更新
1：更新中断挂起。此位在以下情况下更新寄存器时由硬件置 1：
上溢或下溢并且当 TIMx_CR1 寄存器中 UDIS = 0 时。
TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。
TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（参见同步控制寄存器说明）。

25.4.6 TIM2 事件生成寄存器 (TIM2_EGR)

TIM2 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

位 15:7 保留，必须保持复位值。

位 6 **TG**：触发生成 (Trigger generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作。
1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 5 保留，必须保持复位值。

位 4 **CC4G**：捕获/比较 4 生成 (Capture/Compare 4 generation)
请参见 CC1G 说明

位 3 **CC3G**：捕获/比较 3 生成 (Capture/Compare 3 generation)
请参见 CC1G 说明

位 2 **CC2G**：捕获/比较 2 生成 (Capture/compare 2 generation)
请参见 CC1G 说明

位 1 **CC1G**：捕获/比较 1 生成 (Capture/Compare 1 generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作。
1：通道 1 上生成捕获/比较事件：
如果通道 CC1 配置为输出：
使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。
如果通道 CC1 配置为输入：
TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)
此位可通过软件置 1，并由硬件自动清零。
0：不执行任何操作
1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (TIMx_ARR)。

25.4.7 TIM2 捕获/比较模式寄存器 1 [复用] (TIM2_CCMR1)

TIM2 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **IC2F[3:0]**：输入捕获 2 滤波器 (Input capture 2 filter)

位 11:10 **IC2PSC[1:0]**：输入捕获 2 预分频器 (Input capture 2 prescaler)

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/compare 2 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出。

01：CC2 通道配置为输入，IC2 映射到 TI2 上。

10：CC2 通道配置为输入，IC2 映射到 TI1 上。

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效。

注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器的采样长度。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样

0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC[1:0]**：输入捕获 1 预分频器 (Input capture 1 prescaler)

此位域定义 CC1 输入 (IC1) 的预分频比。只要 CC1E=0（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：CC1 通道配置为输入，IC1 映射到 TI2 上
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

25.4.8 TIM2 捕获/比较模式寄存器 1 [复用] (TIM2_CCMR1)

TIM2 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 **OC2CE**：输出比较 2 清零使能 (Output Compare 2 clear enable)

位 24、14:12 **OC2M[3:0]**：输出比较 2 模式 (Output compare 2 mode)

请参见 OC1M 说明——位 6:4

位 11 **OC2PE**：输出比较 2 预装载使能 (Output Compare 2 preload enable)

位 10 **OC2FE**：输出比较 2 快速使能 (Output Compare 2 fast enable)

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出

01：CC2 通道配置为输入，IC2 映射到 TI2 上

10：CC2 通道配置为输入，IC2 映射到 TI1 上

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

位 7 **OC1CE**：输出比较 1 清零使能 (Output Compare 1 clear enable)

0：OC1Ref 不受 ETRF 输入影响

1：ETRF 输入上检测到高电平时，OC1Ref 立即清零

位 16，6:4 **OC1M[3:0]**：输出比较 1 模式 (Output compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为无效状态 (OC1REF=0)，否则为有效状态 (OC1REF=1)。

0111：PWM 模式 2——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。

1000：可再触发 OPM 模式 1——在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。

1001：可再触发 OPM 模式 2——在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。

1010：保留。

1011：保留。

1100：组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101：组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110：不对称 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

1111：不对称 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

注： 在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。

注： 只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其他情况下则无法保证该行为。

位 2 **OC1FE**：输出比较 1 快速使能 (Output Compare 1 fast enable)

此位用于加快触发输入事件对 CC 输出的影响。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出。

01：CC1 通道配置为输入，IC1 映射到 TI1 上。

10：CC1 通道配置为输入，IC1 映射到 TI2 上。

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

25.4.9 TIM2 捕获/比较模式寄存器 2 [复用] (TIM2_CCMR2)

TIM2 capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:12 **IC4F[3:0]**：输入捕获 4 滤波器 (Input capture 4 filter)

位 11:10 **IC4PSC[1:0]**：输入捕获 4 预分频器 (Input capture 4 prescaler)

位 9:8 **CC4S[1:0]**：捕获/比较 4 选择 (Capture/Compare 4 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注：　仅当通道关闭时（TIMx_CCER 中的 CC4E = 0），才可向 CC4S 位写入数据。

位 7:4 **IC3F[3:0]**：输入捕获 3 滤波器 (Input capture 3 filter)

位 3:2 **IC3PSC[1:0]**：输入捕获 3 预分频器 (Input capture 3 prescaler)

位 1:0 **CC3S[1:0]**：捕获/比较 3 选择 (Capture/Compare 3 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注：　仅当通道关闭时（TIMx_CCER 中的 CC3E = 0），才可向 CC3S 位写入数据。

25.4.10 TIM2 捕获/比较模式寄存器 2 [复用] (TIM2_CCMR2)

TIM2 capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M [3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 **OC4CE**：输出比较 4 清零使能 (Output compare 4 clear enable)

位 24、14:12 **OC4M[3:0]**：输出比较 4 模式 (Output compare 4 mode)
请参见 OC1M 说明（TIMx_CCMR1 寄存器中的位 6:4）

位 11 **OC4PE**：输出比较 4 预装载使能 (Output compare 4 preload enable)

位 10 **OC4FE**：输出比较 4 快速使能 (Output compare 4 fast enable)

位 9:8 **CC4S[1:0]**：捕获/比较 4 选择 (Capture/Compare 4 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC4E = 0），才可向 CC4S 位写入数据。

位 7 **OC3CE**：输出比较 3 清零使能 (Output compare 3 clear enable)

位 16，6:4 **OC3M[3:0]**：输出比较 3 模式 (Output compare 3 mode)
请参见 OC1M 说明TIMx_CCMR1 寄存器中的位 6:4）

位 3 **OC3PE**：输出比较 3 预装载使能 (Output compare 3 preload enable)

位 2 **OC3FE**：输出比较 3 快速使能 (Output compare 3 fast enable)

位 1:0 **CC3S[1:0]**：捕获/比较 3 选择 (Capture/Compare 3 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC3E = 0），才可向 CC3S 位写入数据。

25.4.11 TIM2 捕获/比较使能寄存器 (TIM2_CCER)

TIM2 capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

位 15 **CC4NP**：*捕获/比较 4 互补输出极性 (Capture/Compare 4 complementary output Polarity)*
请参见 CC1NP 说明

位 14 保留，必须保持复位值。

位 13 **CC4P**：*捕获/比较 4 输出极性 (Capture/Compare 4 output Polarity)*
请参见 CC1P 说明

位 12 **CC4E**：*捕获/比较 4 输出使能 (Capture/Compare 4 output enable)*
请参见 CC1E 说明

位 11 **CC3NP**：*捕获/比较 3 互补输出极性 (Capture/Compare 3 complementary output Polarity)*
请参见 CC1NP 说明

位 10 保留，必须保持复位值。

位 9 **CC3P**：*捕获/比较 3 输出极性 (Capture/Compare 3 output polarity)*
请参见 CC1P 说明

位 8 **CC3E**：*捕获/比较 3 输出使能 (Capture/Compare 3 output enable)*
请参见 CC1E 说明

位 7 **CC2NP**：*捕获/比较 2 互补输出极性 (Capture/Compare 2 complementary output Polarity)*
请参见 CC1NP 说明

位 6 保留，必须保持复位值。

位 5 **CC2P**：*捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)*
请参见 CC1P 说明

位 4 **CC2E**：*捕获/比较 2 输出使能 (Capture/Compare 2 output enable)*
请参见 CC1E 说明

位 3 **CC1NP**：*捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output Polarity)*
CC1 通道配置为输出：在这种情况下，CC1NP 必须保持清零。
CC1 通道配置为输入：此位与 CC1P 配合使用可定义 TI1FP1/TI2FP1 极性。请参见 CC1P 说明。

位 2 保留，必须保持复位值。

位 1 **CC1P**：*捕获/比较 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：CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的极性。
00：未反相/上升沿触发
电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。
01：反相/下降沿触发
电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。
10：保留，不使用此配置
11：未反相/上升沿和下降沿均触发
电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。

位 0 **CC1E**：*捕获/比较 1 输出使能 (Capture/Compare 1 output enable)*
CC1 通道配置为输出：
0：关闭——OC1 无效
1：开启——在相应输出引脚上输出 OC1 信号
CC1 通道配置为输入：此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1 (TIMx_CCR1) 中。
0：禁止捕获
1：使能捕获

表 159. 标准 OCx 通道的输出控制位

CCxE 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： *与标准 OCx 通道相连的外部 IO 引脚的状态取决于 OCx 通道的状态以及 GPIO 和 AFIO 寄存器。*

25.4.12 TIM2 计数器 [复用] (TIM2_CNT)

TIM2 counter

该寄存器的位 31 有两种可能的定义，具体取决于 TIMx_CR1 寄存器中 UIFREMAP 的值：

- 本节内容对应于 UIFREMAP = 0 的情况
- 下一节内容对应于 UIFREMAP = 1 的情况

偏移地址：0x24

复位值：0x0000 0000

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

位 31 **CNT[31]**：计数器值的最高有效位 (Most significant bit of counter value)
其他定时器上保留。

位 30:0 **CNT[30:0]**：计数器值 (Counter value)

25.4.13 TIM2 计数器 [复用] (TIM2_CNT)

TIM2 counter

该寄存器的位 31 有两种可能的定义，具体取决于 TIMx_CR1 寄存器中 UIFREMAP 的值：

- 上一节内容对应于 UIFREMAP = 0 的情况
- 本节内容对应于 UIFREMAP = 1 的情况

偏移地址：0x24

复位值：0x0000 0000

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

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
此位是 TIMx_ISR 寄存器中 UIF 位的只读副本

位 30:0 **CNT[30:0]**：计数器值 (Counter value)

25.4.14 TIM2 预分频器 (TIM2_PSC)

TIM2 prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 **PSC[15:0]**：预分频值 (Prescaler value)

计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。

25.4.15 TIM2 自动重载寄存器 (TIM2_ARR)

TIM2 auto-reload register

偏移地址：0x2C

复位值：0x0000 FFFF

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

位 31:0 **ARR[31:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的更多详细信息，请参见*第 744 页的第 25.3.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

25.4.16 TIM2 捕获/比较寄存器 1 (TIM2_CCR1)

TIM2 capture/compare register 1

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CCR1[31:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 是捕获/比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 1）。

实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。只能读取 TIMx_CCR1 寄存器，无法对其进行编程。

25.4.17 TIM2 捕获/比较寄存器 2 (TIM2_CCR2)

TIM2 capture/compare register 2

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CCR2[31:0]**：捕获/比较 2 值 (Capture/Compare 2 value)

如果通道 CC2 配置为输出：

CCR2 是捕获/比较寄存器 2 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 2）。

实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。

如果通道 CC2 配置为输入：

CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。只能读取 TIMx_CCR2 寄存器，无法对其进行编程。

25.4.18 TIM2 捕获/比较寄存器 3 (TIM2_CCR3)

TIM2 capture/compare register 3

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CCR3[31:0]**：捕获/比较值 (Capture/Compare value)

如果通道 CC3 配置为输出：

CCR3 是捕获/比较寄存器 3 的预装载值。

如果没有通过 TIMx_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 3）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。

如果通道 CC3 配置为输入：

CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。只能读取 TIMx_CCR3 寄存器，无法对其进行编程。

25.4.19 TIM2 捕获/比较寄存器 4 (TIM2_CCR4)

TIM2 capture/compare register 4

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **CCR4[31:0]**：捕获/比较值 (Capture/Compare value)

1. 如果 CC4 通道配置为输出（CC4S 位）：
   CCR4 是捕获/比较寄存器 4 的预装载值。
   如果没有通过 TIMx_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 4）。
   有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。
2. 如果 CC4 通道配置为输入（TIMx_CCMR4 寄存器中的 CC4S 位）：
   CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。只能读取 TIMx_CCR4 寄存器，无法对其进行编程。

25.4.20 TIM2 DMA 控制寄存器 (TIM2_DCR)

TIM2 DMA control register

偏移地址：0x48

复位值：0x0000

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

位 15:13 保留，必须保持复位值。

位 12:8 **DBL[4:0]**：DMA 连续传送长度 (DMA burst length)

该 5 位向量定义了 DMA 的传送次数（当对 TIMx_DMAR 寄存器进行读或写时，定时器进行一次连续传送）。

00000：1 次传送
00001：2 次传送
00010：3 次传送
...
10001：18 次传送

位 7:5 保留，必须保持复位值。

位 4:0 **DBA[4:0]**：DMA 基址 (DMA base address)

该 5 位向量定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：
00000：TIMx_CR1
00001：TIMx_CR2
00010：TIMx_SMCR
...

示例：以下面的传送为例：DBL = 7 次传送且 DBA = TIMx_CR1。这种情况下将向/从自 TIMx_CR1 地址开始的 7 个寄存器传输数据。

25.4.21 TIM2 全传输 DMA 地址 (TIM2_DMAR)

TIM2 DMA address for full transfer

偏移地址：0x4C

复位值：0x0000

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

位 15:0 **DMAB[15:0]**：DMA 连续传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器

(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

25.4.22 TIM2 选择寄存器 1 (TIM2_OR1)

TIM2 option register 1

偏移地址：0x50

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI4_RMP		ETR1_RMP	ITR_RMP
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:2 **TI4_RMP**：定时器输入 4 重映射 (Timer input 4 remap)
由软件置 1 和清零。
00：TIM2 TI4 连接到 GPIO（请参见复用功能映射）
01：TIM2 TI4 连接到 COMP1_OUT
10：TIM2 TI4 连接到 COMP2_OUT
11：TIM2 TI4 连接到 COMP1_OUT 和 COMP2_OUT 的逻辑或运算结果

位 1 **ETR1_RMP**：外部触发 1 重映射 (External trigger 1 remap)
由软件置 1 和清零。
0：TIM2 ETR1 连接到 GPIO（请参见复用功能映射）
1：LSE 内部时钟连接到 TIM2_ETR1 输入

位 0 **ITR_RMP**：内部触发重映射 (Internal trigger remap)
0：TIM2 内部触发 ITR2 未连接
1：TIM2 内部触发 ITR2 连接至 USB_FS_SOF

25.4.23 TIM2 复用功能选择寄存器 1 (TIM2_AF1)

TIM2 alternate function option register 1

偏移地址：0x60

复位值：0x0000 0000

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

位 31:18 保留，必须保持复位值。

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)
这些位选择 ETR 输入源。
0000：GPIO 或 LSE 内部时钟，具体取决于 TIM2_OR1 中的 ETR1_RMP 位
0001：COMP1
0010：COMP2
其他值：保留

位 13:0 保留，必须保持复位值。

25.4.24 TIM2 定时器输入选择寄存器 (TIM2_TISEL)

TIM2 timer input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：TI2[0] 到 TI2[15] 输入选择 (TI2[0] to TI2[15] input selection)
这些位选择 TI2[0] 到 TI2[15] 输入源。
0000：TIM2_CH2 输入
其他值：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：TI1[0] 到 TI1[15] 输入选择 (TI1[0] to TI1[15] input selection)
这些位选择 TI1[0] 到 TI1[15] 输入源。
0000：TIM2_CH1 输入
其他值：保留

25.4.25 TIMx 寄存器映射

TIMx 寄存器按照下表所述方式映射：

表 160. TIM2 寄存器映射和复位值

偏移	寄存器名称	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
0x00	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0	0	0	0	0	0	0	0
0x04	**TIMx_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			CCDS	Res.	Res.	Res.
	Reset value																									0	0	0	0	0			
0x08	**TIMx_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3]		Res.	Res.	Res.	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value											0	0				0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																				0	0	0	0	0		0		0	0	0	0	0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value																				0	0	0	0			0		0	0	0	0	0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																										0		0	0	0	0	0
0x18	**TIMx_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**TIMx_CCMR2** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	O24CE	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR2 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4 PSC [1:0]		CC4S [1:0]		IC3F[3:0]				IC3 PSC [1:0]		CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIMx_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
	Reset value																	0		0	0	0		0	0	0		0	0	0		0	0

表 160. TIM2 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x24	**TIMx_CNT**	CNT[31] or UIFCPY	CNT[30:0]																														
	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
0x28	**TIMx_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	ARR[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
0x30	Reserved																																
0x34	**TIMx_CCR1**	CCR1[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
0x38	**TIMx_CCR2**	CCR2[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
0x3C	**TIMx_CCR3**	CCR3[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
0x40	**TIMx_CCR4**	CCR4[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
0x44	Reserved																																
0x48	**TIMx_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIMx_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	**TIM2_OR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI4_RMP		ETR1_RMP	ITR_RMP
	Reset value																													0	0	0	0
0x60	**TIM2_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value															0	0	0	0														

表 160. TIM2 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x68	**TIM2_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																					0	0	0	0					0	0	0	0

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

26 通用定时器 (TIM16/TIM17)

26.1 TIM16/TIM17 简介

TIM16/TIM17 定时器包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器可用于各种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、PWM 和带死区插入的互补 PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

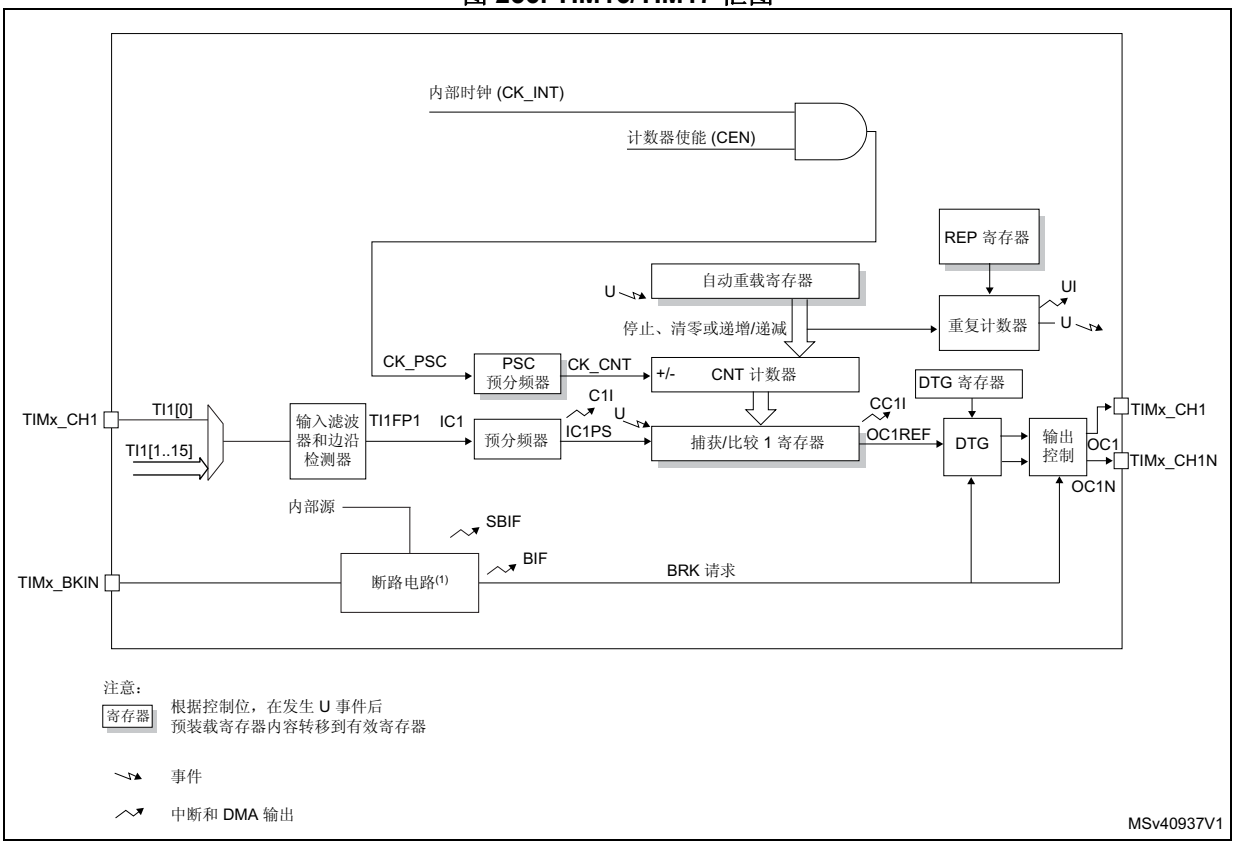
TIM16/TIM17 定时器彼此完全独立，不共享任何资源。

26.2 TIM16/TIM17 主要特性

TIM16/TIM17 定时器包括下列特性：

- 16 位自动重载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65535 之间
- 一个具有以下用途的通道：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐模式）
  - 单脉冲模式输出
- 带可编程死区的互补输出
- 重复计数器，用于仅在给定数目的计数器周期后更新定时器寄存器
- 用于将定时器的输出信号置于复位状态或已知状态的断路输入
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢
  - 输入捕获
  - 输出比较
  - 断路输入

图 256. TIM16/TIM17 框图



26.3 TIM16/TIM17 功能描述

26.3.1 时基单元

可编程高级控制定时器的主要模块是一个 16 位递增计数器及其相关的自动重载寄存器。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)
- 重复计数器寄存器 (TIMx_RCR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 TIMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器说明

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 TIMx_PSC 寄存器中的 16 位寄存器所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 257 和*图 258* 以一些示例说明在预分频比实时变化时计数器的行为：

图 257. 预分频器分频由 1 变为 2 时的计数器时序图

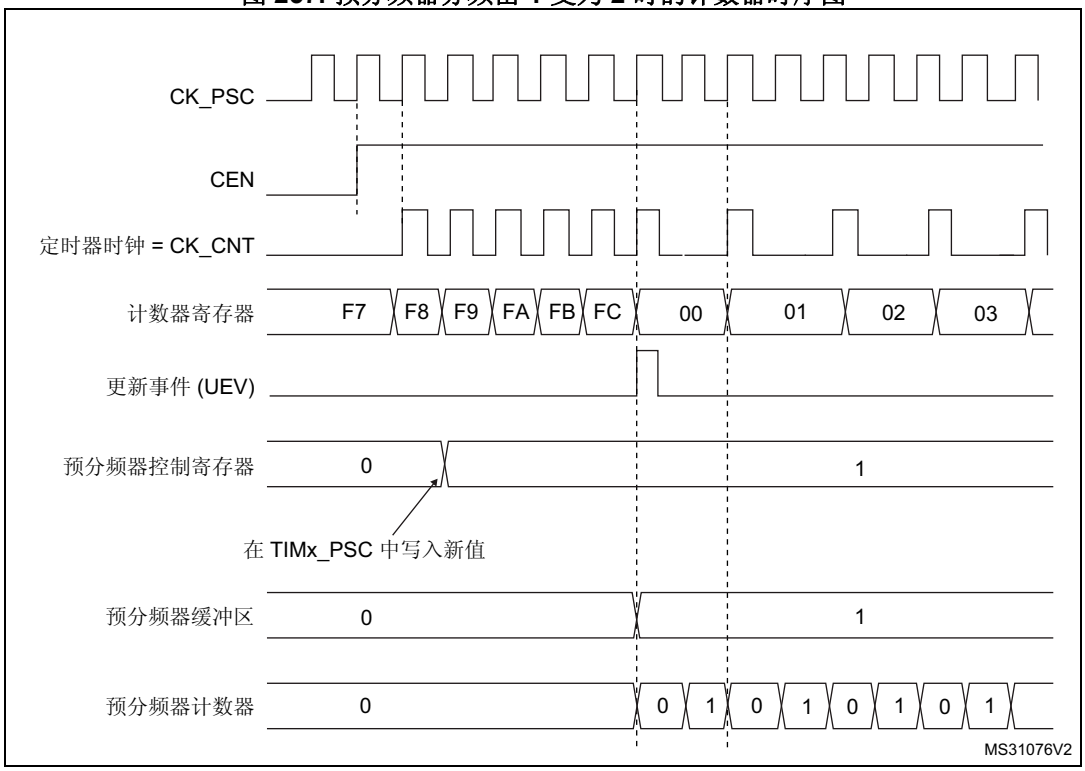
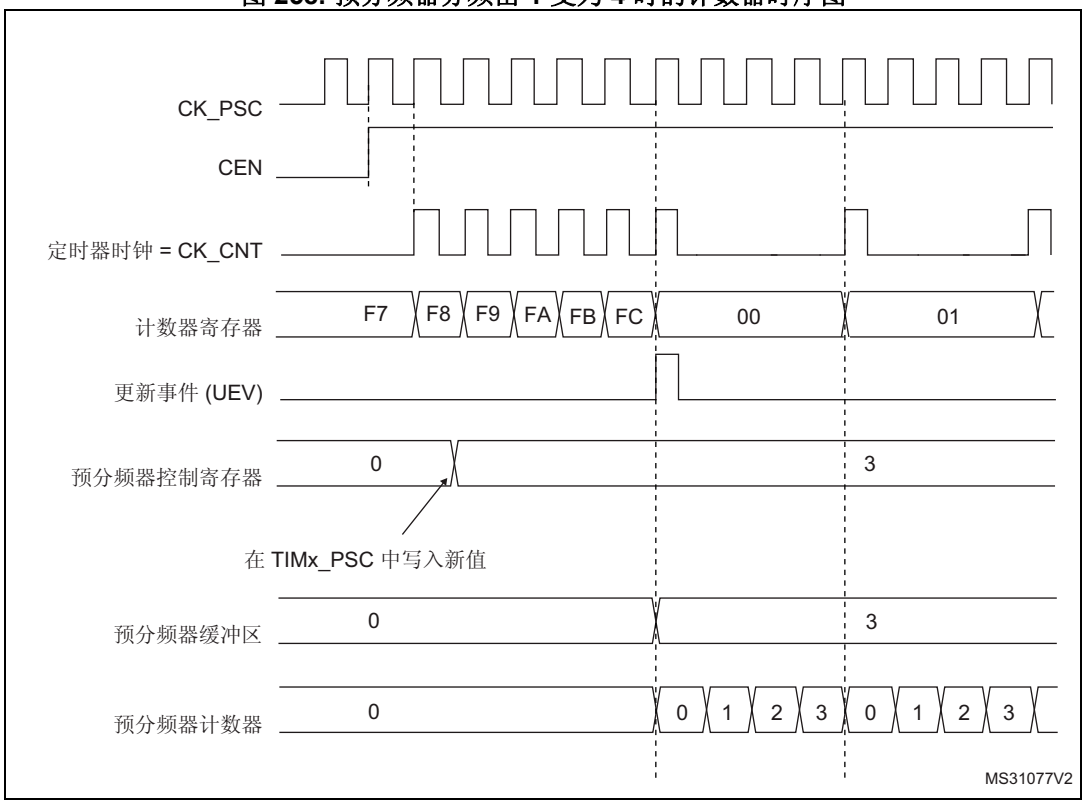


图 258. 预分频器分频由 1 变为 4 时的计数器时序图



26.3.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中编程的次数 (TIMx_RCR) 后，将生成更新事件 (UEV)。否则，将在每次计数器上溢时产生更新事件。

将 TIMx_EGR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容。
- 自动重载影子寄存器将以预装载值 (TIMx_ARR) 进行更新。
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 259. 计数器时序图，1 分频内部时钟

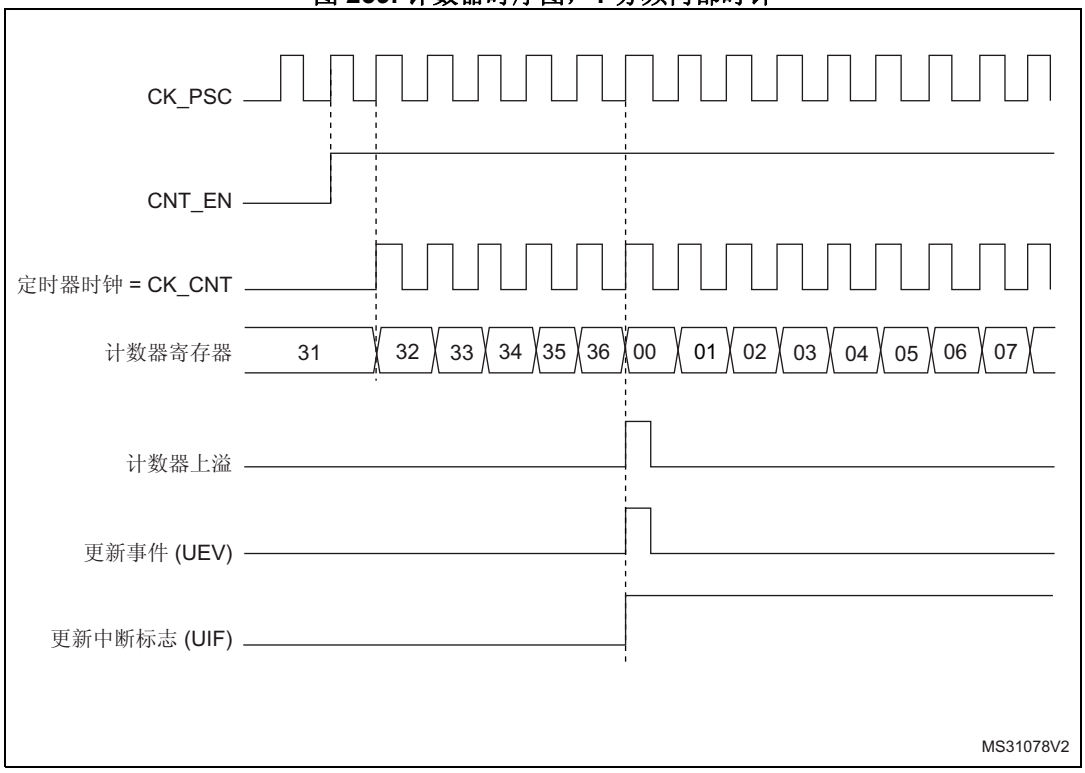


图 260. 计数器时序图，2 分频内部时钟

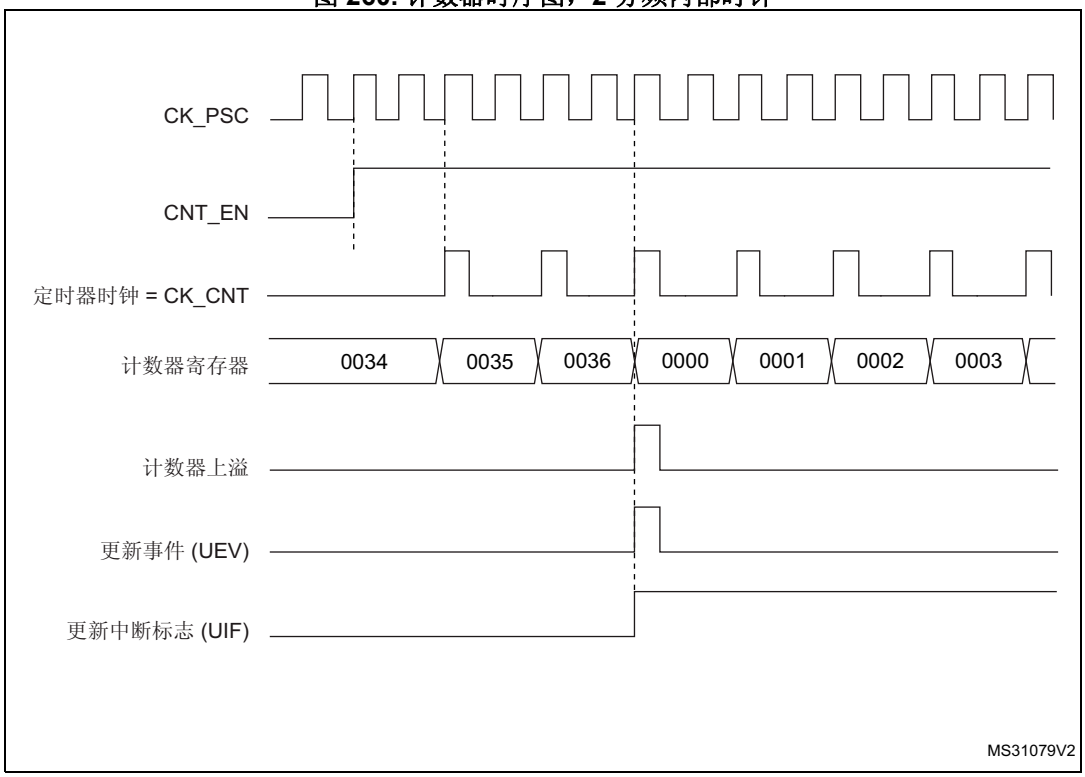


图 261. 计数器时序图，4 分频内部时钟

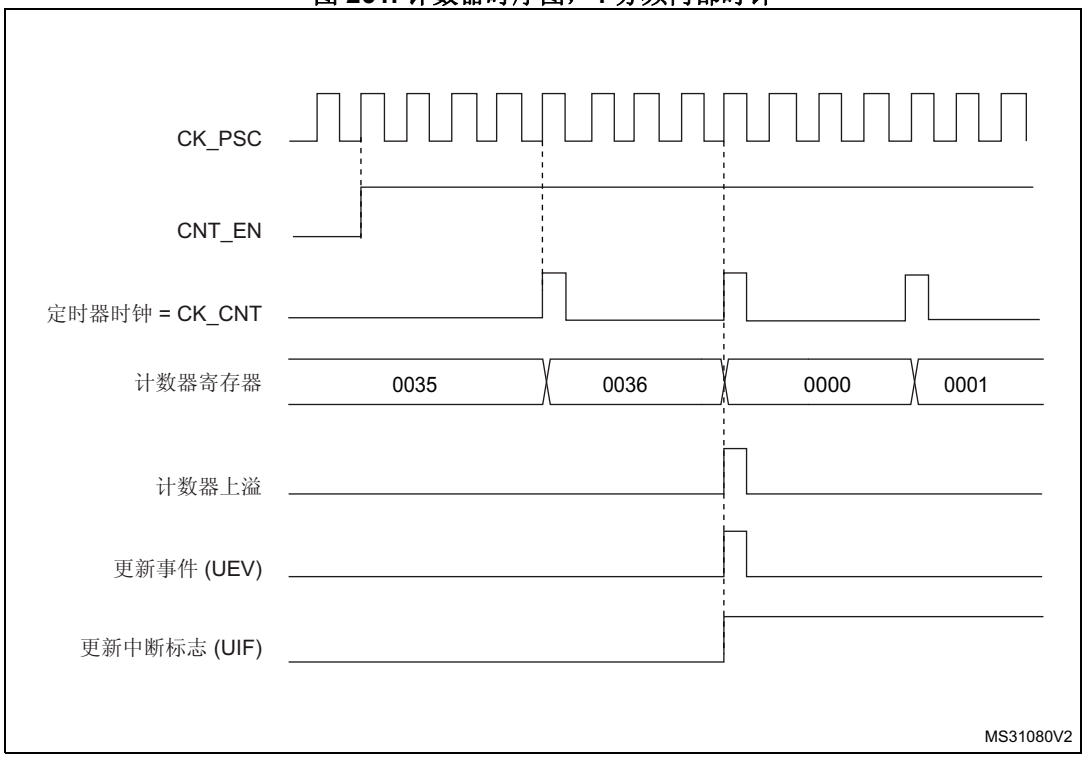


图 262. 计数器时序图，N 分频内部时钟

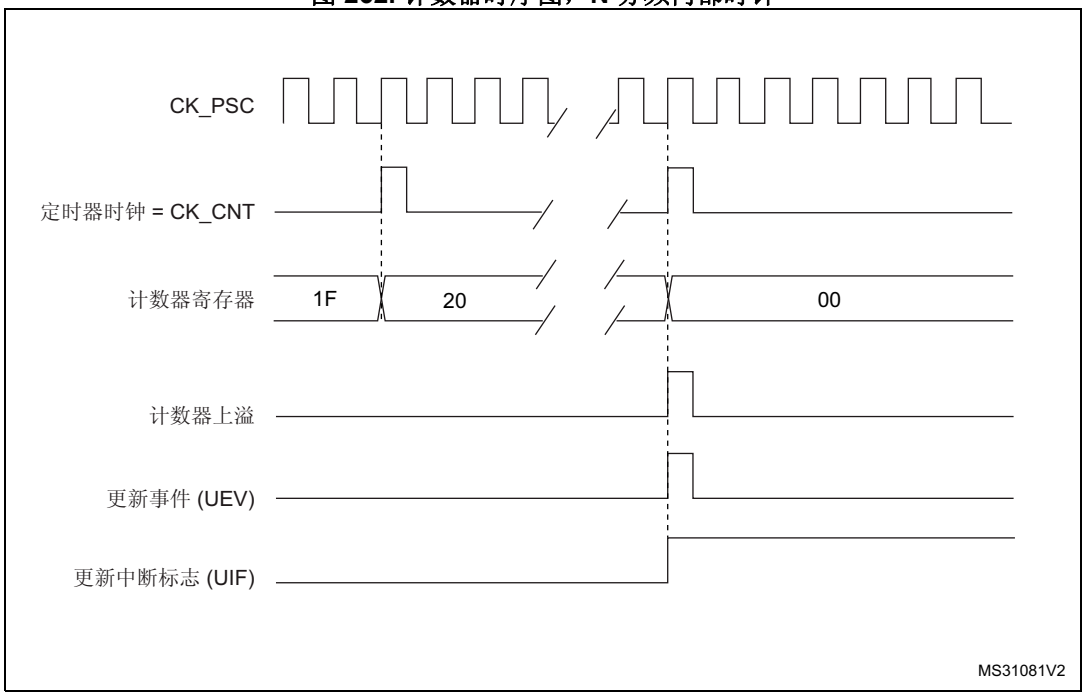


图 263. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

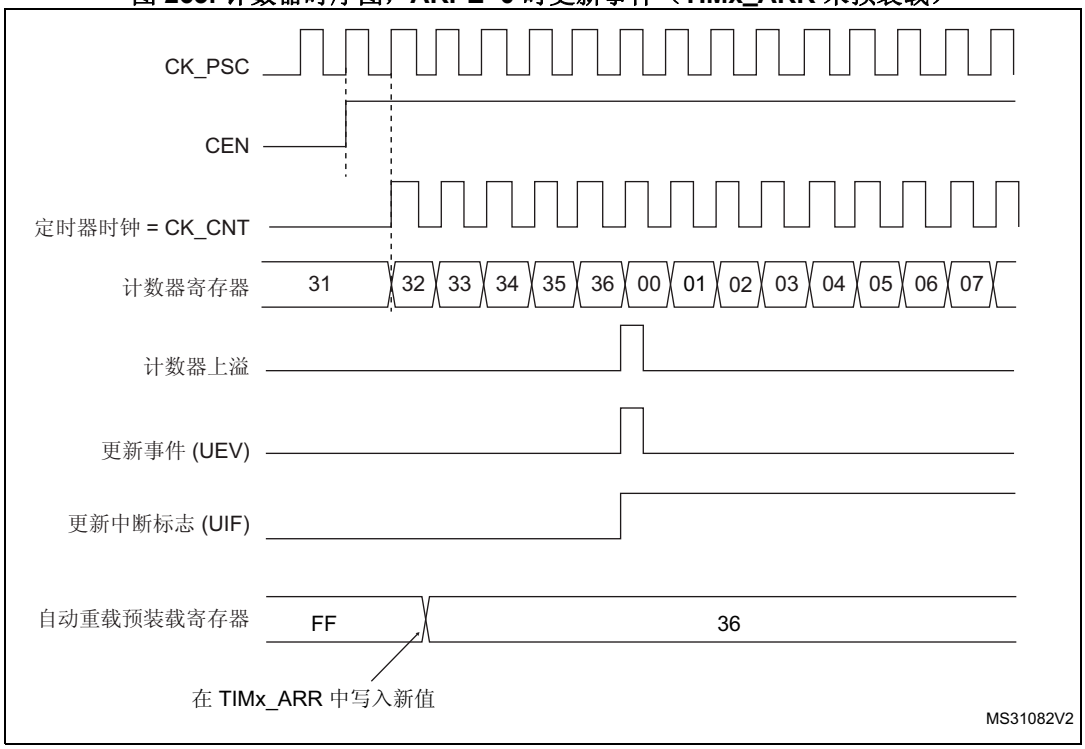
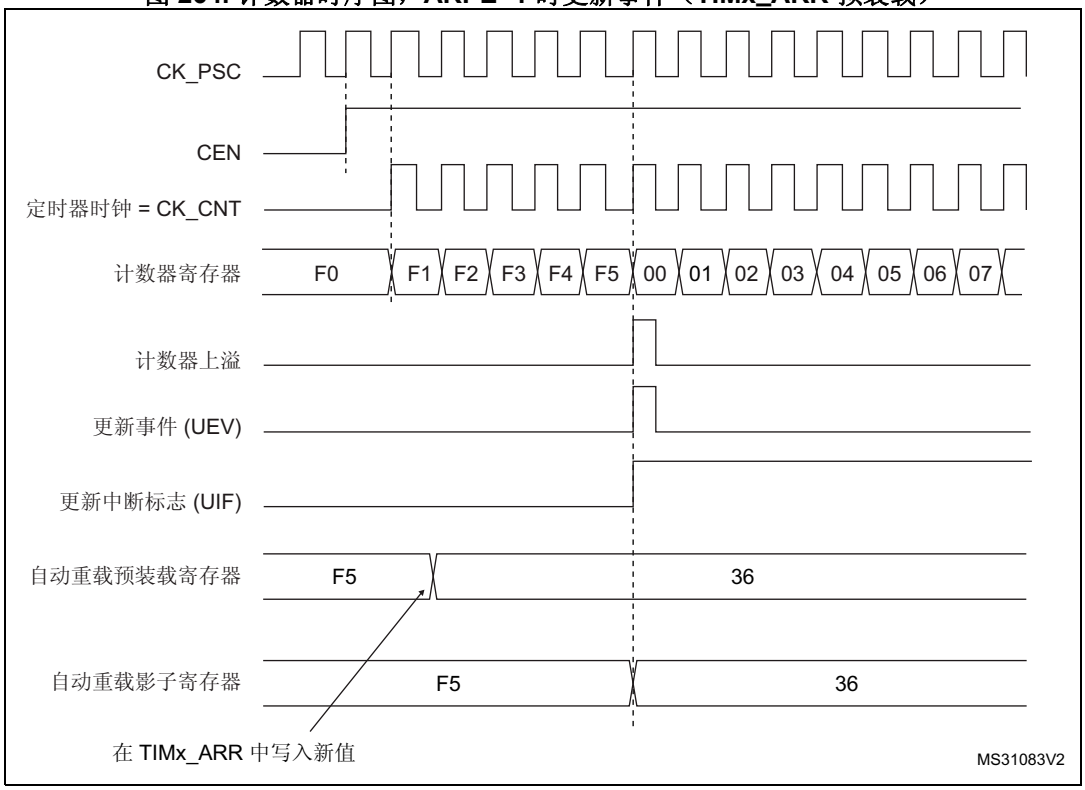


图 264. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 预装载）



26.3.3 重复计数器

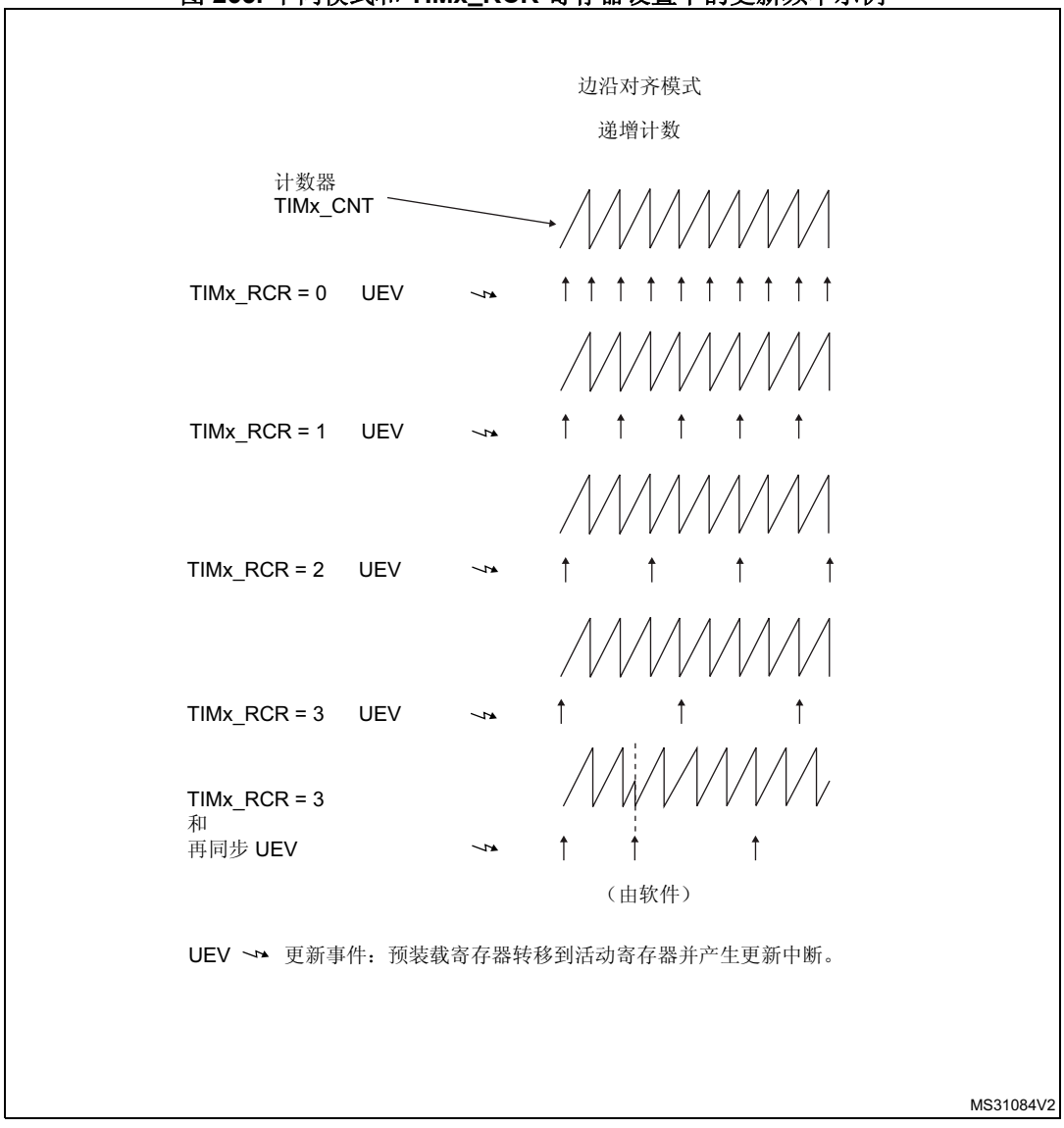
*第 26.3.1 节：时基单元*介绍如何因计数器上溢而生成更新事件 (UEV)。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

这意味着，每当发生 N 个计数器上溢（其中，N 是 TIMx_RCR 重复计数器寄存器中的值），数据就将从预装载寄存器转移到影子寄存器（TIMx_ARR 自动重载寄存器、TIMx_PSC 预分频器寄存器以及比较模式下的 TIMx_CCRx 捕获/比较寄存器）。

重复计数器在每个计数器上溢时递减。

重复计数器是自动重载类型；其重复率为 TIMx_RCR 寄存器所定义的值（请参见*图 265*）。当更新事件由软件（通过将 TIMx_EGR 寄存器的 UG 位置 1）或硬件（通过从模式控制器）生成时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TIMx_RCR 寄存器的内容。

图 265. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例



26.3.4 时钟选择

计数器时钟可由下列时钟源提供：

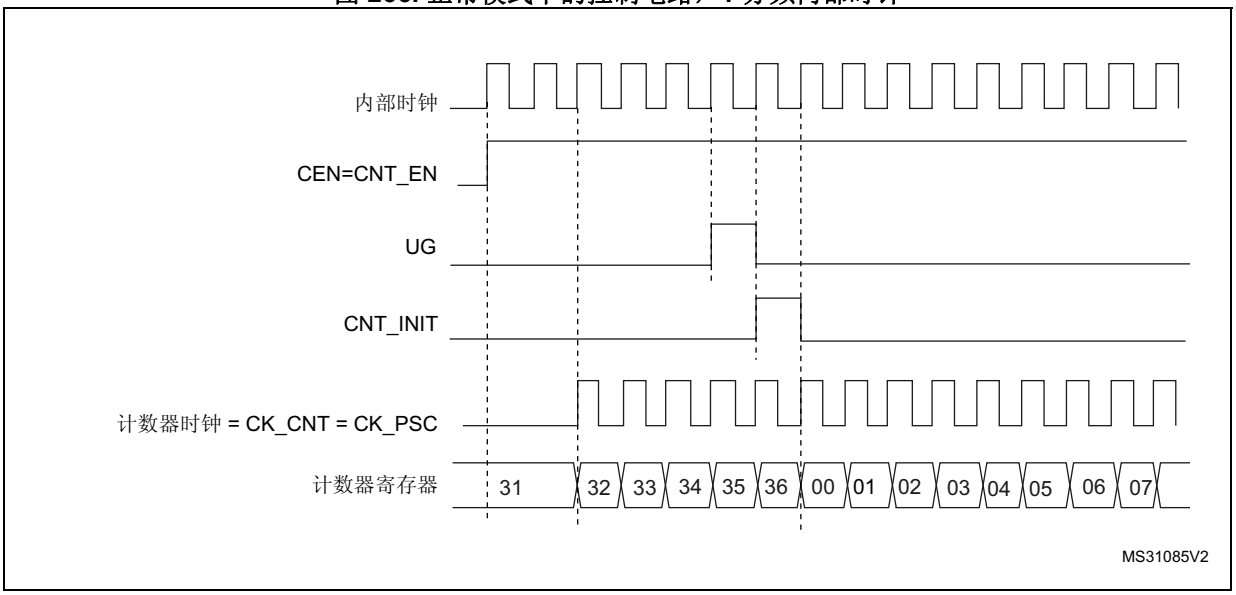
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚

内部时钟源 (CK_INT)

如果禁止从模式控制器 (SMS=000)，则 CEN 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 266 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

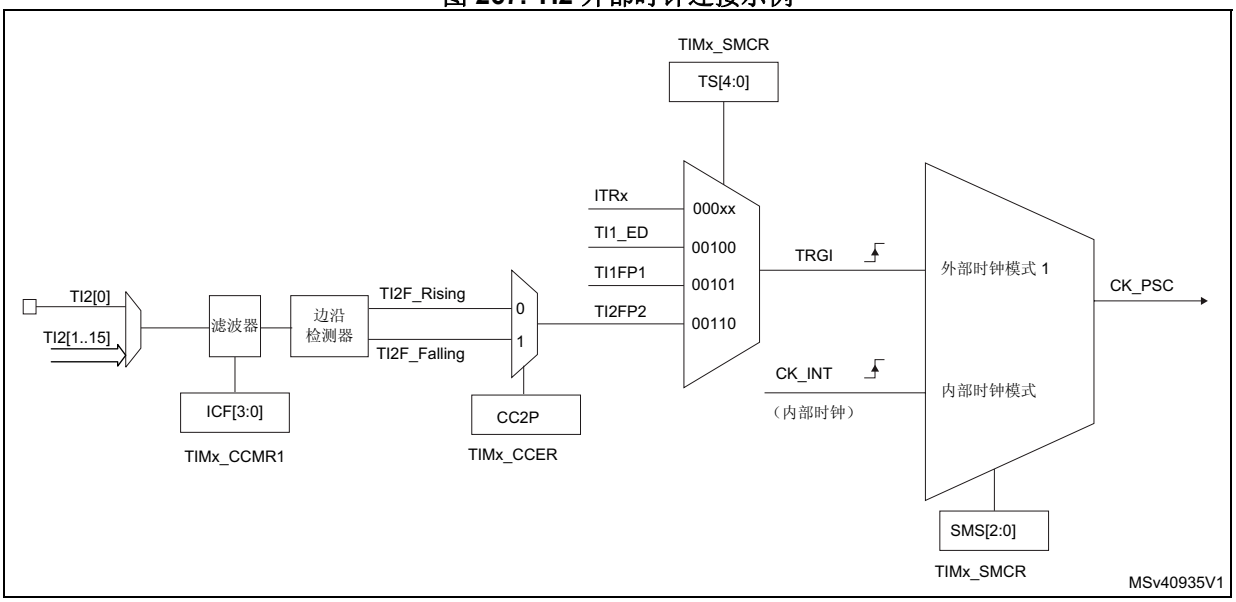
图 266. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 267. TI2 外部时钟连接示例



例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

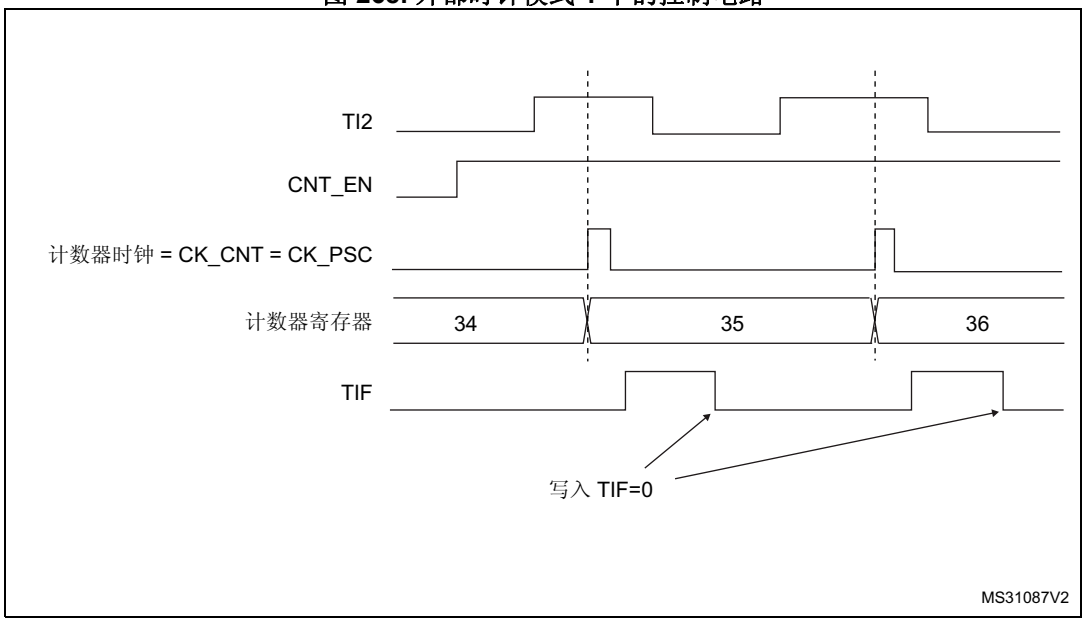
1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2[x] 源（内部或外部）。
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S = “01” 来配置通道 2，使其能够检测 TI2 输入的上升沿。
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。
4. 通过在 TIMx_CCER 寄存器中写入 CC2P = 0 来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为触发输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

注： *由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 268. 外部时钟模式 1 下的控制电路



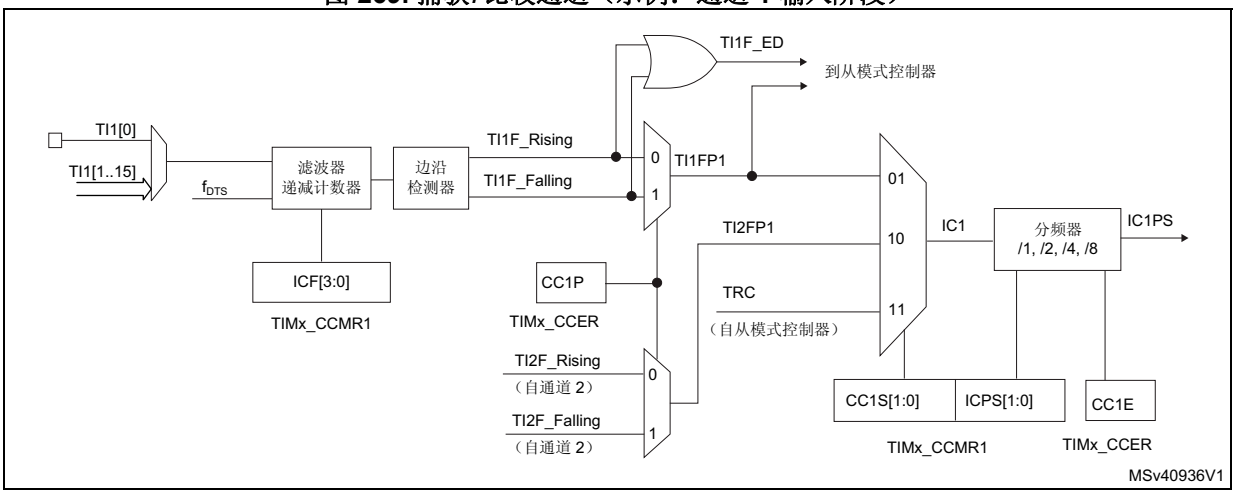
26.3.5 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

图 269 到*图 271* 简要介绍了一路捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 269. 捕获/比较通道（示例：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 270. 捕获/比较通道 1 主电路

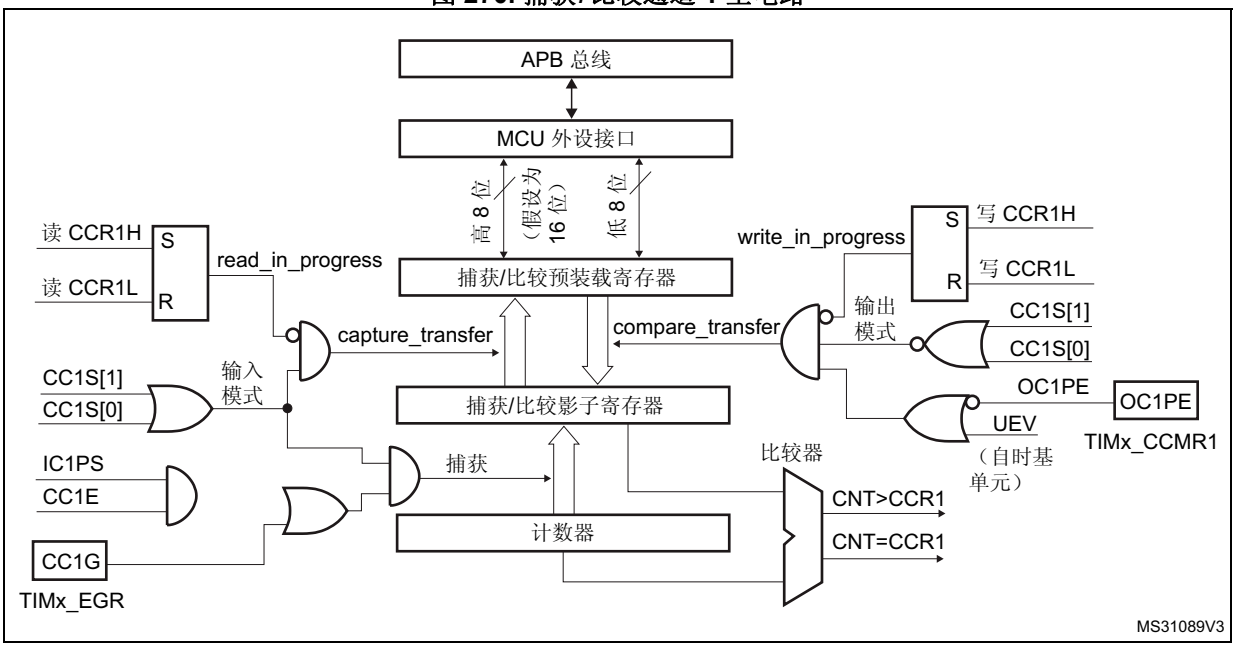
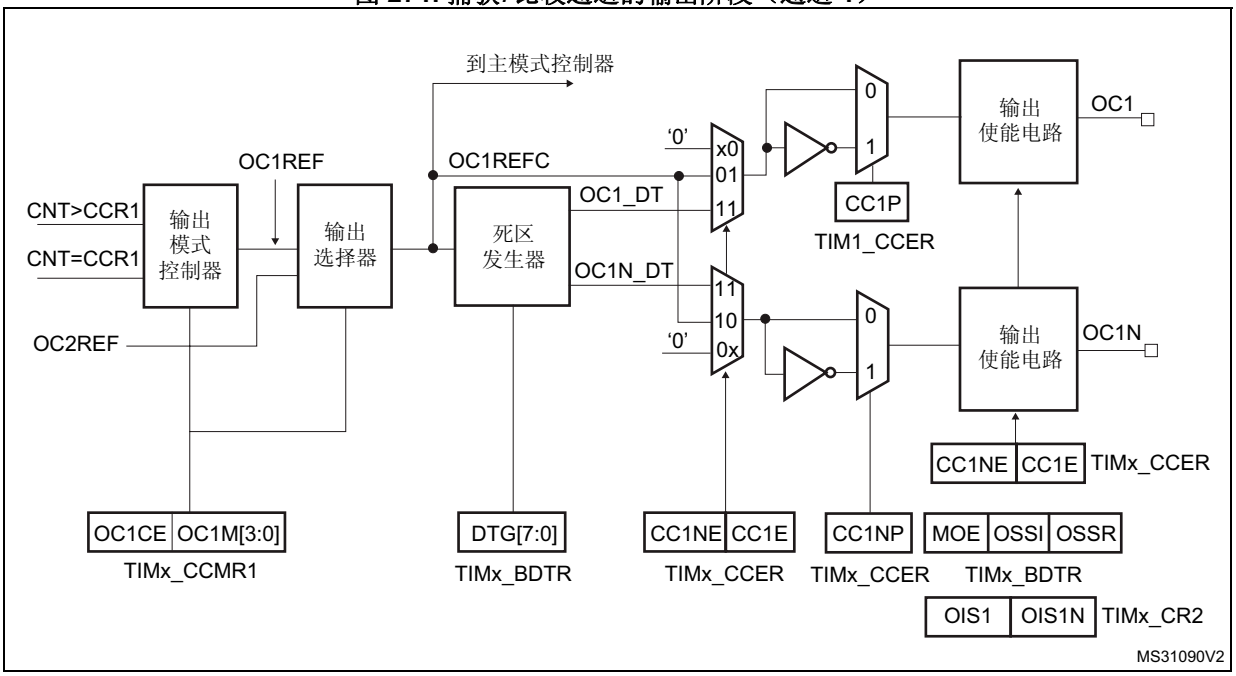


图 271. 捕获/比较通道的输出阶段（通道 1）



捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容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将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

26.3.6 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件将 CCxIF 清零，方法是：向 CCxIF 写入“0”，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据连接到定时器的信号，对相关输入滤波带宽进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号边沿变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波带宽设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 f_{DTS} 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
4. 通过在 TIMx_CCER 寄存器中将 CC1P 位写入 0，选择 TI1 上的有效转换边沿（本例中为上升沿）。
5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

注： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

26.3.7 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（OCxREF 和 OCx/OCxN）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (OCXREF/OCx) 强制设置为有效电平，只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 101。OCxREF 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=0（OCx 高电平有效）=> 将 OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 100，可将 OCxREF 信号强制设置为低电平。

无论如何，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。下面的输出比较模式一节对此进行了介绍。

26.3.8 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCXM=000)，也可设置为有效电平 (OCXM=001)、无效电平 (OCXM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

使用 TIMx_CCMRx 寄存器中的 OCxPE 位，可将 TIMx_CCRx 寄存器配置为带或不带预装载寄存器。

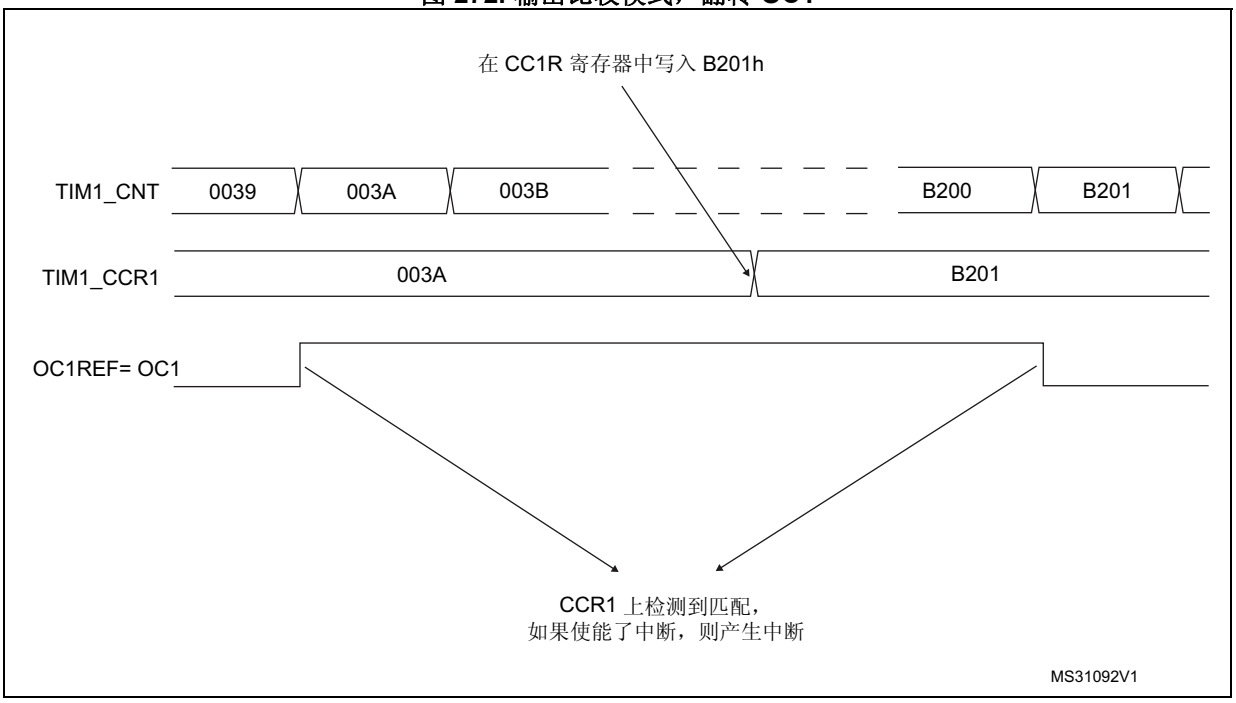
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   - 当 CNT 与 CCRx 匹配时，写入 OCxM = 011 以翻转 OCx 输出引脚
   - 写入 OCxPE = 0 以禁止预装载寄存器
   - 写入 CCxP = 0 以选择高电平有效极性
   - 写入 CCxE = 1 以使能输出
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可通过软件随时更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=“0”，否则 TIMx_CCRx 影子寄存器仅在下一更新事件 UEV 发生时进行更新）。*图 272* 给出了一个示例。

图 272. 输出比较模式，翻转 OC1



26.3.9 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

各通道可以独立选择 PWM 模式（每个 OCx 输出对应一个 PWM），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“110”（PWM 模式 1）或“111”（PWM 模式 2）。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

由于只有在发生更新事件时预装载寄存器才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

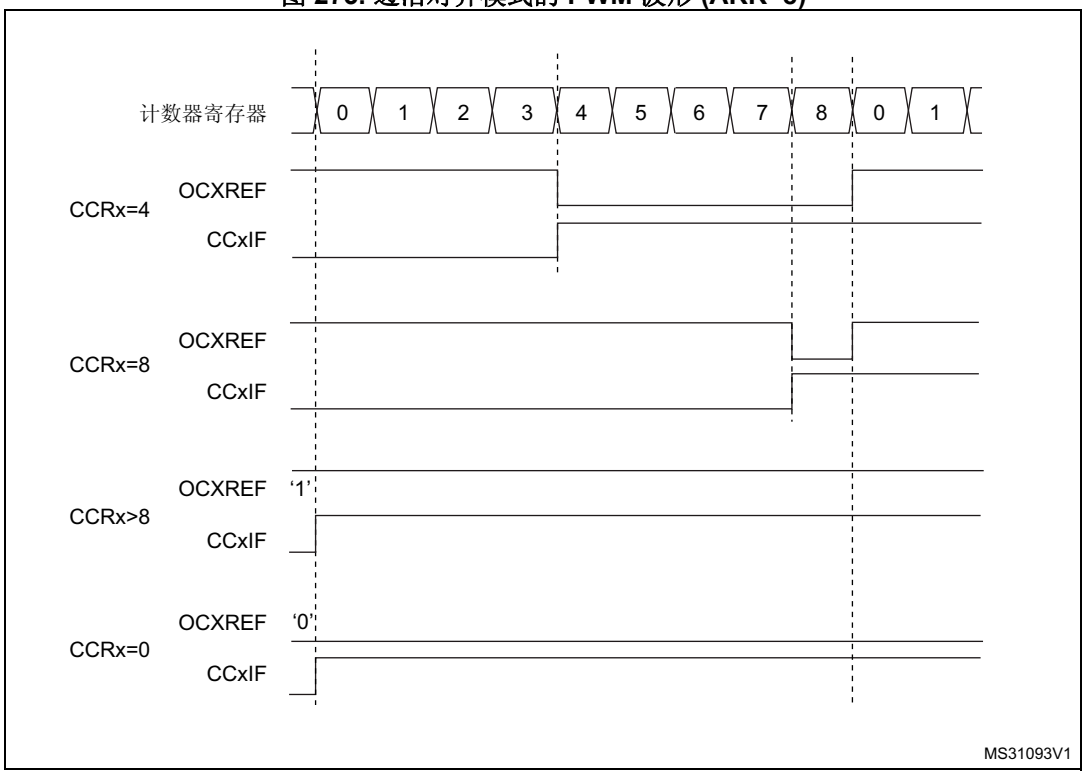
OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIMx_CCER 和 TIMx_BDTR 寄存器）的组合使能 OCx 输出。有关详细信息，请参见 TIMx_CCER 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 总是与 TIMx_CCRx 进行比较，以确定是 TIMx_CCRx ≤ TIMx_CNT 还是 TIMx_CNT ≤ TIMx_CCRx（取决于计数器计数方向）。

TIM16/TIM17 只能递增计数。请参见*第 814 页的递增计数模式*。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。*图 273* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 273. 边沿对齐模式的 PWM 波形 (ARR=8)



26.3.10 互补输出和死区插入

TIM16/TIM17 通用定时器可以输出一路互补信号，并管理输出的关断和接通。

这段时间通常称为死区，用户必须根据与输出相连接的器件及其特性（电平转换器的固有延迟、开关器件产生的延迟...）来调整死区时间

每路输出可以独立选择输出极性（主输出 OCx 或互补输出 OCxN）。可通过对 TIMx_CCER 寄存器中的 CCxP 和 CCxNP 位执行写操作来完成极性选择。

互补信号 OCx 和 OCxN 通过以下多个控制位的组合进行激活：TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。更多详细信息，请参见 *第 845 页的表 162：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17)*。应当注意，切换至空闲状态（MOE 下降到 0）的时刻，死区仍然有效。

CCxE 和 CCxNE 位同时置 1 并且 MOE 位置 1（如果存在断路）时，将使能死区插入。每个通道有一个 10 位死区发生器。将基于参考波形 OCxREF 生成 2 个输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效：

- 输出信号 OCx 与参考信号相同，只是其上升沿相对参考上升沿存在延迟。
- 输出信号 OCxN 与参考信号相反，并且其上升沿相对参考下降沿存在延迟。

如果延迟时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。

下图所示为死区发生器的输出信号与参考信号 OCxREF 之间的关系。（在这些示例中，假定 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）

图 274. 带死区插入的互补输出

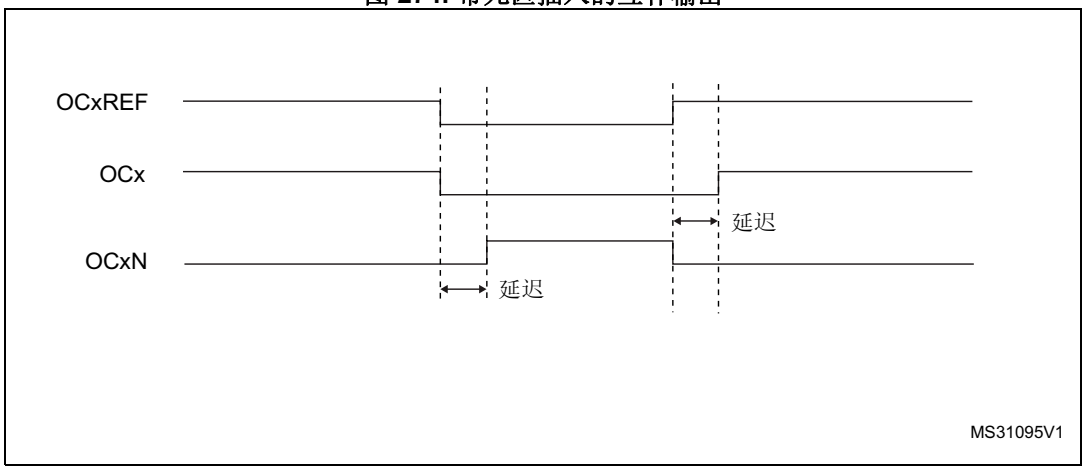


图 275. 延迟时间大于负脉冲宽度的死区波形

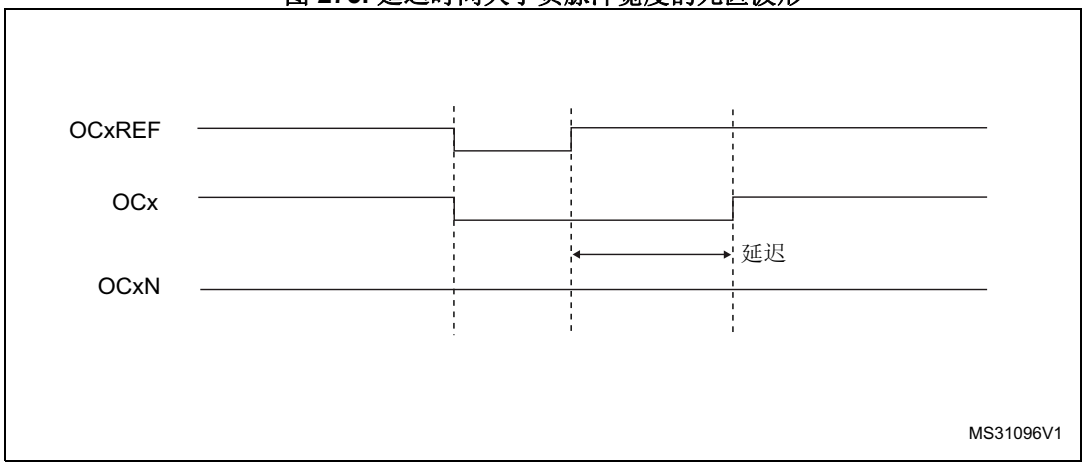
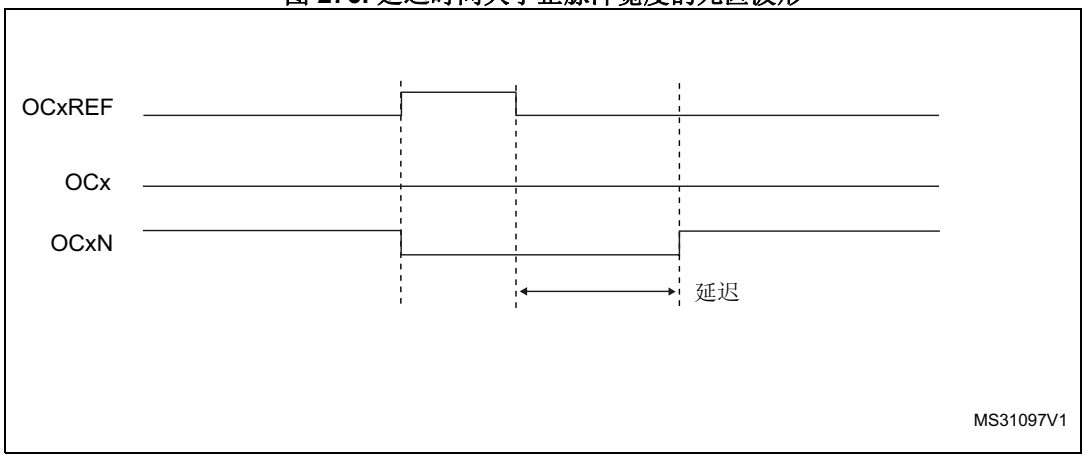


图 276. 延迟时间大于正脉冲宽度的死区波形



死区延迟对于所有通道均相同，可通过 TIMx_BDTR 寄存器中的 DTG 位进行编程。有关延迟时间计算的信息，请参见*第 848 页的第 26.4.14 节：TIMx 断路和死区寄存器 (TIMx_BDTR)（x = 16 到 17）*。

将 OCxREF 重定向到 OCx 或 OCxN

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位，可将 OCxREF 重定向到 OCx 输出或 OCxN 输出。

通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者，使两个输出同时保持无效电平，或者两个输出同时处于有效电平，两者互补并且带死区。

注： *如果仅使能 OCxN (CCxE=0, CCxNE=1)，两者不互补，一旦 OCxREF 为高电平，OCxN 即变为有效。例如，如果 CCxNP=0，则 OCxN=OCxRef。另一方面，如果同时使能 OCx 和 OCxN (CCxE=CCxNE=1)，OCx 在 OCxREF 为高电平时变为有效，而 OCxN 则与之互补，在 OCxREF 为低电平时变为有效。*

26.3.11 使用刹车功能

断路功能的目的是保护由 TIM16/TIM17 定时器生成的 PWM 信号所驱动的电源开关。断路输入通常被连接到功率级和三相逆变器的故障输出。激活时，断路电路会关闭 PWM 输出，并将其强制为预定义的安全状态。

退出复位状态后，断路功能处于禁止状态，MOE 位处于低电平。通过设置 TIMx_BDTR 寄存器中的 BKE 位来使能断路功能。断路输入的极性可通过该寄存器中的 BKP 位来选择。BKE 和 BKP 位可同时修改。对 BKE 和 BKP 位执行写操作时，写操作会在 1 个 APB 时钟周期的延迟后生效。因此，执行写操作后，需要等待 1 个 APB 时钟周期，才能准确回读此位。

由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 TIMx_BDTR 寄存器中）之间插入了再同步电路，从而在异步信号与同步信号之间产生延迟。具体而言，如果在 MOE 处于低电平时将其置 1，则必须首先插入延迟（空指令），才能准确进行读取。这是因为写入的是异步信号，而读取的却是同步信号。

发生断路（断路输入上出现所选电平）时：

- MOE 位异步清零，使输出处于无效状态、空闲状态甚至释放对 AFIO 控制器的控制（通过 OSSI 位进行选择）。即使 MCU 振荡器关闭，该功能仍然有效。
- MOE=0 时，将以 TIMx_CR2 寄存器 OISx 位中编程的电平驱动每个输出通道。如果 OSSI = 0，定时器将释放输出控制（由 AFIO 控制器接管），否则使能输出保持高电平。
- 使用互补输出时：
  - 输出首先置于复位状态或无效状态（取决于极性）。这是异步操作，因此即使没有为定时器提供时钟，该操作仍有效。
  - 如果定时器时钟仍存在，则将重新激活死区发生器，进而在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，也不能同时将 OCx 和 OCxN 驱动至其有效电平。请注意，MOE 进行再同步，因此死区的持续时间会比通常情况长一些（约 2 个 ck_tim 时钟周期）。
  - 如果 OSSI = 0，定时器将释放使能输出（由强制高阻态的 AFIO 控制器接管），否则使能输出将保持高电平或在 CCxE 或 CCxNE 位之一为高电平时立即变为高电平。
- 将断路状态标志（TIMx_SR 寄存器中的 BIF 位）置 1。如果 TIMx_DIER 寄存器中的 BIE 位置 1，可产生中断。
- 如果 TIMx_BDTR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件 (UEV) 时自动再次置 1。这一特性有许多用处，比如，可用于实现调节器的功能。否则，MOE 将始终保持低电平，直到再次向此位写入 1。这种情况下，这一特性可用于确保安全。可以将断路输入连接到功率驱动器的警报、温度传感器或任何安全元件。

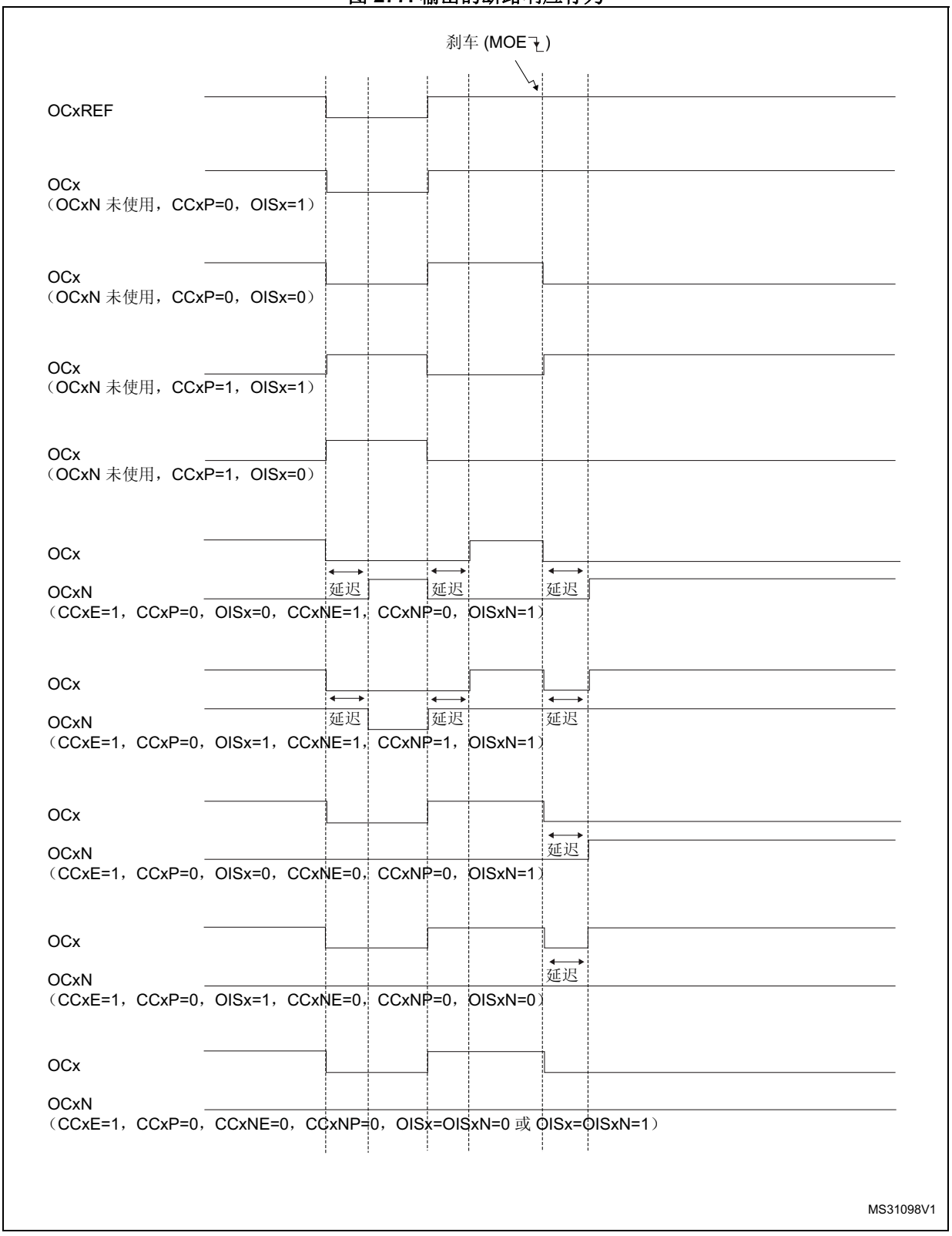
注： *断路输入为电平有效。因此，当断路输入有效电平时，不能将 MOE 位置 1（自动或通过软件）。同时，不能将状态标志 BIF 清零。*

断路可由 BRK 输入生成，该输入具有可编程极性，其使能位 BKE 位于 TIMx_BDTR 寄存器中。

除断路输入和输出管理外，断路电路内部还实施了写保护，用以保护应用的安全。通过该功能，用户可冻结多个参数配置（死区持续时间、OCx/OCxN 极性和禁止时的状态、OCxM 配置、断路使能和极性）。可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参见*第 848 页的第 26.4.14 节：TIMx 断路和死区寄存器 (TIMx_BDTR)（x = 16 到 17）*。MCU 复位后只能对 LOCK 位执行一次写操作。

图 277 所示为输出对断路响应行为的示例。

图 277. 输出的断路响应行为



26.3.12 双向断路输入

TIM16/TIM17 具有双向断路 I/O，如*图 278* 所示。

它们可以：

- 将板级全局断路信号用于向外部 MCU 或栅极驱动器发送故障信号，唯一的引脚作为输入和输出状态引脚。
- 在必须将多个内部和外部断路输入合并时，将内部断路源和多个外部开漏比较器输出“或”连接在一起，触发唯一断路事件。

使用 TIMxBDTR 寄存器的 BKBID 位将断路输入配置为双向模式。可以使用 TIMxBDTR 寄存器中的 LOCK 位，将 BKBID 编程位锁定在只读模式（处于锁定级别 1 或更高级别）。

双向模式需要将 I/O 配置为开漏模式且使极性低电平有效（使用 BKINP 和 BKP 位）。任何来自系统（例如 CSS）、片上外设或断路输入的断路请求都会强制将断路输入置为低电平，以通知发生了故障事件。如果未正确设置极性位（高电平有效极性），则出于安全目的禁止双向模式。

软件断路事件 (BG) 也会导致断路 I/O 被强制为“0”，从而向外部组件指示定时器已进入断路状态。但是仅在断路使能时 (BKE = 1) 有效。当生成软件断路事件且 BKE = 0 时，输出将被置于安全状态，并且断路标志置 1，但对断路 I/O 无影响。

安全解除机制可防止系统最终锁定（断路输入上的低电平会触发断路，进而将相同输入强制置为低电平）。

当 BKDSRM 位置 1 时，会释放断路输出以清除故障信号，从而使系统能够重新启动。

在任何情况下都不能禁止断路保护电路：

- 断路输入路径始终有效：即使 BKDSRM 位置 1 且释放开漏控制，断路事件也仍然有效。这样可以在发生断路期间防止 PWM 输出重新启动。
- 使能输出（MOE 位置 1）后，BKDSRM 位不能解除断路保护（请参见*表 161*）。

表 161. 断路保护解除条件

MOE	BKDIR	BKDSRM	断路保护状态
0	0	X	启动
0	1	0	启动
0	1	1	解除
1	X	X	启动

启动和重新启动断路电路

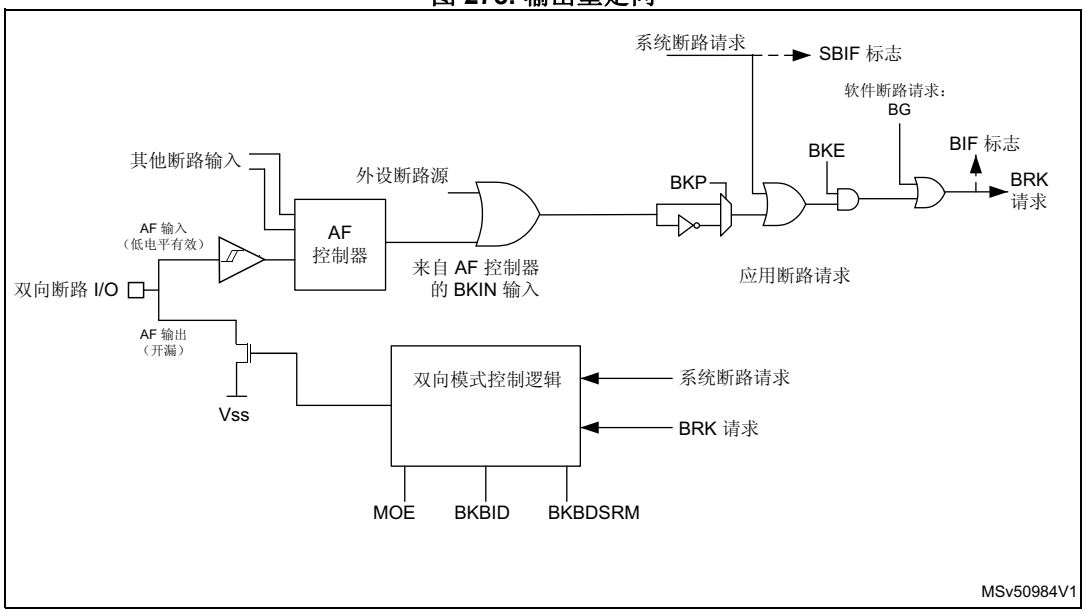
默认情况下（外设复位配置）会启动断路电路（在输入或双向模式下）。

发生断路事件后，必须按照以下步骤重新启动保护：

- 必须将 BKDSRM 位置 1，以释放输出控制
- 软件必须等待系统断路条件消失（如果有），并清零 SBIF 状态标志（或在重新启动前由系统清零）
- 软件必须轮询 BKDSRM 位，直到此位由硬件清零（当应用断路条件消失时）

此后，断路电路即启动并激活，可以通过将 MOE 位置 1 来重新使能 PWM 输出。

图 278. 输出重定向



26.3.13 单脉冲模式

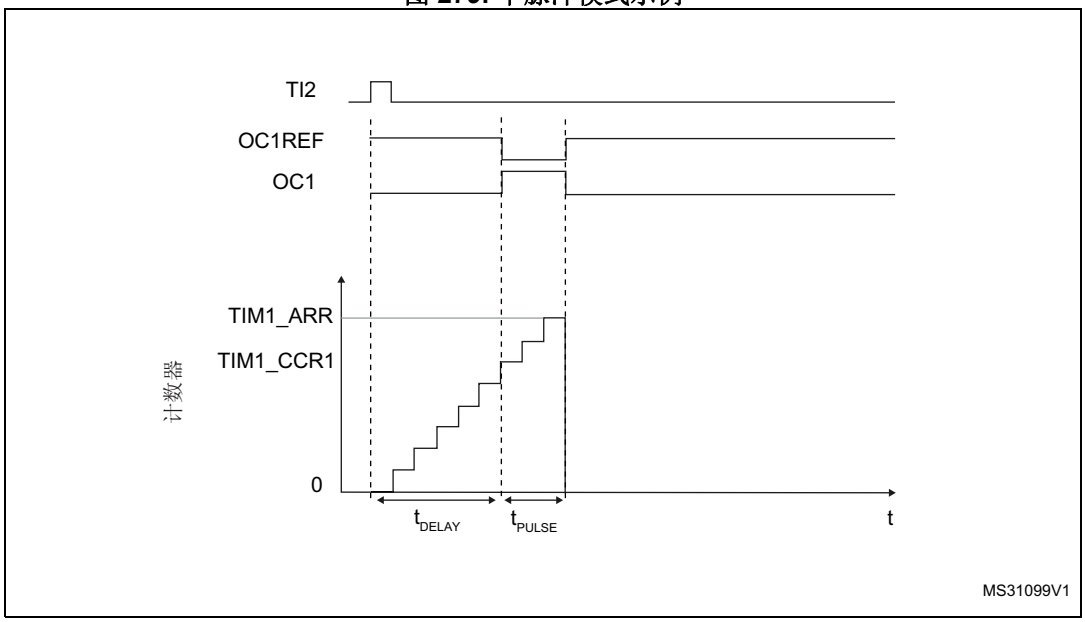
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。通过将 TIMx_CR1 寄存器中的 OPM 位置 1 选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- CNT < CCRx ≤ ARR（特别注意，0 < CCRx）

图 279. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到正沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2[x] 源（内部或外部）。
2. 在 TIMx_CCMR1 寄存器中写入 CC2S=“01”，以将 TI2FP2 映射到 TI2。
3. 在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
4. 在 TIMx_SMCR 寄存器中写入 TS=“00110”，以将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
5. 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），以使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，必须通过在 TIMx_CCMR1 寄存器中写入 OC1M=111，来使能 PWM 模式 2。可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

由于仅需要 1 个脉冲，因此必须向 TIMx_CR1 寄存器的 OPM 位写入 1，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。

特殊情况：OCx 快速使能

在单脉冲模式下，TIx 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIMx_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxRef（和 OCx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

26.3.14 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

26.3.15 从模式——组合复位 + 触发模式

在这种情况下，在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件，并启动计数器。

该模式用于单脉冲模式。

26.3.16 DMA 连续传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对几个定时器寄存器多次重新编程而无需软件开销，但也可用于定期读取一行中的多个寄存器。

DMA 控制器目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（突发）。每次写入 TIMx_DMAR 寄存器都会重定向到其中一个定时器寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 连续传送长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送，即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   - DMA 通道外设地址为 DMAR 寄存器地址。
   - DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   - 要传输的数据量 = 3（参见下文注释）。
   - 禁止循环模式。
2. 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：
   DBL = 3 次传输，DBA = 0xE。
3. 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx。
5. 使能 DMA 通道。

本例适用于每个 CCRx 寄存器只更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。

注： *可以将空值写入保留的寄存器中。*

26.3.17 调试模式

当微控制器进入调试模式（CPU1 Cortex®-M4 内核停止）时，TIMx 计数器会根据 DBG 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。有关详细信息，请参见 *第 41.8.7 节：DBGMCU CPU1 APB2 外设冻结寄存器 (DBGMCU_APB2FZR)*。

为了安全起见，当计数器停止 (DBG_TIMx_STOP = 1) 时，输出被禁止（就像 MOE 位被复位一样）。可以将输出强制变为未激活状态（OSSI 位 = 1），或者通过 GPIO 控制器（OSSI 位 = 0）来控制输出，以将其强制为高阻态。

26.4 TIM16/TIM17 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.2 节*。

26.4.1 TIMx 控制寄存器 1 (TIMx_CR1)（x = 16 到 17）

TIMx control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rw		rw	rw	rw				rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)

0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)

此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器 (TIx) 所使用的死区及采样时钟 (t_{DTS}) 之间的分频比。

00：$t_{DTS}=t_{CK\_INT}$
01：$t_{DTS}=2*t_{CK\_INT}$
10：$t_{DTS}=4*t_{CK\_INT}$
11：保留，不要设置成此值

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)

0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:4 保留，必须保持复位值。

位 3 **OPM**：单脉冲模式 (One pulse mode)

0：计数器在发生更新事件时不会停止计数
1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 **URS**：更新请求源 (Update request source)

此位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会产生更新中断或 DMA 请求。此类事件包括：
- 计数器上溢/下溢
- 将 UG 位置 1
- 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 **UDIS**：更新禁止 (Update disable)

此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

然后更新影子寄存器的值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注： 只有事先通过软件将 CEN 位置 1，才可以使用外部时钟和门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

26.4.2 TIMx 控制寄存器 2 (TIMx_CR2)（x = 16 到 17）

TIMx control register 2

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC
						rw	rw					rw	rw		rw

位 15:10 保留，必须保持复位值。

位 9 **OIS1N**：输出空闲状态 1（OC1N 输出）(Output Idle state 1 (OC1N output))

0：当 MOE=0 时，经过死区时间后 OC1N=0

1：当 MOE=0 时，经过死区时间后 OC1N=1

注： 只要编程了 LOCK（TIMx_BKR 寄存器中的 LOCK 位）级别 1、2 或 3，此位即无法修改。

位 8 **OIS1**：输出空闲状态 1（OC1 输出）(Output Idle state 1 (OC1 output))

0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0

1：当 MOE=1 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0

注： 只要编程了 LOCK（TIMx_BKR 寄存器中的 LOCK 位）级别 1、2 或 3，此位即无法修改。

位 7:4 保留，必须保持复位值。

位 3 **CCDS**：捕获/比较 DMA 选择 (Capture/compare DMA selection)

0：发生 CCx 事件时发送 CCx DMA 请求

1：发生更新事件时发送 CCx DMA 请求

位 2 **CCUS**：捕获/比较控制更新选择 (Capture/compare control update selection)

0：如果捕获/比较控制位进行预装载 (CCPC=1)，仅通过将 COMG 位置 1 来对这些位进行更新。

1：如果捕获/比较控制位进行预装载 (CCPC=1)，可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。

注： 此位仅对具有互补输出的通道有效。

位 1 保留，必须保持复位值。

位 0 **CCPC**：捕获/比较预装载控制 (Capture/compare preloaded control)
0：CCxE、CCxNE 和 OCxM 位未进行预装载。
1：CCxE、CCxNE 和 OCxM 位在写入后被预装载，只有当 COM 位置 1 时才进行更新。
注： 此位仅对具有互补输出的通道有效。

26.4.3 TIMx DMA/中断使能寄存器 (TIMx_DIER)（x = 16 到 17）

TIMx DMA/interrupt enable register

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
Res.	Res.	Res.	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE
						rw	rw	rw		rw				rw	rw

位 15:10 保留，必须保持复位值。

位 9 **CC1DE**：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)
0：禁止 CC1 DMA 请求
1：使能 CC1 DMA 请求

位 8 **UDE**：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

位 7 **BIE**：断路中断使能 (Break interrupt enable)
0：禁止断路中断
1：使能断路中断

位 6 保留，必须保持复位值。

位 5 **COMIE**：COM 中断使能 (COM interrupt enable)
0：禁止 COM 中断
1：使能 COM 中断

位 4:2 保留，必须保持复位值。

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断
1：使能 CC1 中断

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断
1：使能更新中断

26.4.4 TIMx 状态寄存器 (TIMx_SR)（x = 16 到 17）

TIMx status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF
						rc_w0		rc_w0		rc_w0				rc_w0	rc_w0

位 15:10 保留，必须保持复位值。

位 9 **CC1OF**：捕获/比较 1 重复捕获标志 (Capture/Compare 1 overcapture flag)

仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将此位清零。

0：未检测到重复捕获。

1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8 保留，必须保持复位值。

位 7 **BIF**：断路中断标志 (Break interrupt flag)

只要断路输入变为有效状态，此标志便由硬件置 1。断路输入无效后可通过软件对其清零。

0：未发生断路事件。

1：在断路输入上检测到有效电平。

位 6 保留，必须保持复位值。

位 5 **COMIF**：COM 中断标志 (COM interrupt flag)

发生一个 COM 事件时，会由硬件将该标志置 1（一旦捕获/比较控制位——CCxE、CCxNE、OCxM——已更新）。但需要通过软件清零。

0：未发生 COM 事件。

1：COM 中断挂起。

位 4:2 保留，必须保持复位值。

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/Compare 1 interrupt flag)

如果通道 CC1 配置为输出：

当计数器与比较值匹配时，此标志由硬件置 1。但需要通过软件清零。

0：不匹配。

1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢时变为高电平。

如果通道 CC1 配置为输入：

此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将此位清零。

0：未发生输入捕获事件

1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）

位 0 **UIF**：更新中断标志 (Update interrupt flag)

此位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。此位在以下情况下更新寄存器时由硬件置 1：

– TIMx_CR1 寄存器中的 UDIS = 0，并且重复计数器值上溢时（重复计数器 = 0 时更新）。

– TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

26.4.5 TIMx 事件生成寄存器 (TIMx_EGR)（x = 16 到 17）

TIMx event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	Res.	COMG	Res.	Res.	Res.	CC1G	UG
								w		w				w	w

位 15:8 保留，必须保持复位值。

位 7 **BG**：断路生成 (Break generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作。
1：生成断路事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 6 保留，必须保持复位值。

位 5 **COMG**：捕获/比较控制更新生成 (Capture/Compare control update generation)
此位可通过软件置 1，并由硬件自动清零。
0：不执行任何操作
1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位
注： 此位仅对具有互补输出的通道有效。

位 4:2 保留，必须保持复位值。

位 1 **CC1G**：捕获/比较 1 生成 (Capture/Compare 1 generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作。
1：通道 1 上生成捕获/比较事件：
如果通道 CC1 配置为输出：
使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。
如果通道 CC1 配置为输入：
TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)
此位可通过软件置 1，并由硬件自动清零。
0：不执行任何操作。
1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。

26.4.6 TIMx 捕获/比较模式寄存器 1 [复用] (TIMx_CCMR1)（x = 16 到 17）

TIMx capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输入捕获模式（本节）或输出比较模式（下一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器的采样长度。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样

0001：$f_{SAMPLING}=f_{CK\_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK\_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK\_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC[1:0]**：输入捕获 1 预分频器 (Input capture 1 prescaler)

此位域定义 CC1 输入 (IC1) 的预分频比。

只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获

01：每发生 2 个事件便执行一次捕获

10：每发生 4 个事件便执行一次捕获

11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：CC1 通道配置为输入，IC1 映射到 TI2 上

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

26.4.7 TIMx 捕获/比较模式寄存器 1 [复用] (TIMx_CCMR1)（x = 16 到 17）

TIMx capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

同一寄存器可用于输出比较模式（本节）或输入捕获模式（上一节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其他位在输入模式和输出模式下的功能均不同。

输出比较模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
									rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 15： 保留，必须保持复位值。

位 16，6:4 **OC1M[3:0]**：输出比较 1 模式 (Output Compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——只要 TIMx_CNT<TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。

0111：PWM 模式 2——只要 TIMx_CNT<TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。

所有其他值：保留

*注： **1**：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。*

***2**：在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。*

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。

*注： **1**：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。*

***2**：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其他情况下则无法保证该行为。*

位 2 **OC1FE**：输出比较 1 快速使能 (Output Compare 1 fast enable)

此位用于加快触发输入事件对 CC 输出的影响。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OC1FE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：CC1 通道配置为输入，IC1 映射到 TI2 上

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

26.4.8 TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 16 到 17）

TIMx capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
												rw	rw	rw	rw

位 15:4 保留，必须保持复位值。

位 3 **CC1NP**：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)

CC1 通道配置为输出：

0：OC1N 高电平有效。

1：OC1N 低电平有效。

CC1 通道配置为输入：

此位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。

*注： **1.** 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 且 CC1S= “00”（通道配置为输出），此位立即变为不可写状态。*

***2.** 此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。*

位 2 **CC1NE**：捕获/比较 1 互补输出使能 (Capture/Compare 1 complementary output enable)

0：关闭——OC1N 未激活。OC1N 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的函数。

1：开启——在相应输出引脚上输出 OC1N 信号，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位。

位 1 **CC1P**：捕获/比较 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：

CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的极性。

00：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。

01：反相/下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式下执行触发操作）。

10：保留，不使用此配置。

11：未反相/边沿触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。

注： ***1.*** *只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，此位立即变为不可写状态。*

2. *此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。*

位 0 **CC1E**：捕获/比较 1 输出使能 (Capture/Compare 1 output enable)

CC1 通道配置为输出：

0：关闭——OC1 未激活。OC1 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的函数。

1：开启——OC1 信号输出到相应的输出引脚上，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位。

CC1 通道配置为输入：

此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1 (TIMx_CCR1) 中。

0：禁止捕获

1：使能捕获

表 162. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17)

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	X	0	0	禁止输出（不由定时器驱动：高阻态） OCx=0 OCxN=0、OCxN_EN=0	
		0	0	1	禁止输出（不由定时器驱动：高阻态） OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出（不由定时器驱动：高阻态） OCxN=0
		X	1	1	OCREF + 极性 + 死区	OCREF 互补项（对 OCREF 进行“非”运算）+ 极性 + 死区
		1	0	1	关闭状态（输出使能为无效状态） OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		1	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP、OCx_EN=1	关闭状态（输出使能为无效状态） OCxN=CCxNP、OCxN_EN=1
0	0	X	X	X	禁止输出（不再由定时器驱动）。输出状态由 GPIO 控制器定义，可以是高电平、低电平或高阻态。	
	1		0	0		
			0	1	关闭状态（输出使能为无效状态） 异步：OCx = CCxP、OCxN = CCxNP 那么如果时钟存在：在死区后 Ocx = OISx 且 OCxN = OISxN，从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应	
			1	0		
			1	1		

1. 如果一个通道的两个输出均未使用（由 GPIO 控制器接管控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注： *与互补通道 OCx 和 OCxN 相连的外部 I/O 引脚的状态取决于通道 OCx 和 OCxN 的状态以及 AFIO 寄存器。*

26.4.9 TIMx 计数器 (TIMx_CNT)（x = 16 到 17）

TIMx counter

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

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

位 31 **UIFCPY**：UIF 副本 (UIF Copy)

此位是 TIMx_ISR 寄存器中 UIF 位的只读副本。如果 TIMx_CR1 中的 UIFREMAP 位复位，则位 31 保留，读为 0。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT**[15:0]：计数器值 (Counter value)

26.4.10 TIMx 预分频器 (TIMx_PSC)（x = 16 到 17）

TIMx prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 **PSC[15:0]**：预分频值 (Prescaler value)

计数器时钟频率 (CK_CNT) 等于 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。

26.4.11 TIMx 自动重载寄存器 (TIMx_ARR)（x = 16 到 17）

TIMx auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的更多详细信息，请参见*第 812 页的第 26.3.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

26.4.12 TIMx 重复计数器寄存器 (TIMx_RCR)（x = 16 到 17）

TIMx repetition counter register

偏移地址：0x30

复位值：0x0000

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

位 15:8 保留，必须保持复位值。

位 7:0 **REP[7:0]**：重复计数器值 (Repetition Counter value)

使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向活动寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。

与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，无论向 TIMx_RCR 寄存器写入何值都无影响。

这意味着在 PWM 模式 (REP+1)下 对应于边沿对齐模式的 PWM 周期数。

26.4.13 TIMx 捕获/比较寄存器 1 (TIMx_CCR1)（x = 16 到 17）

TIMx capture/compare register 1

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR1[15:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 是捕获/比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 1）。

实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

26.4.14 TIMx 断路和死区寄存器 (TIMx_BDTR)（x = 16 到 17）

TIMx break and dead-time register

偏移地址：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	BKBID	Res.	BK DSRM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
			rw		rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： *由于 LOCK 配置决定了 BKBID、BKDSRM、AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 的位可能被写锁定，因此必须在第一次对 TIMx_BDTR 寄存器执行写访问时对这些位进行配置。*

位 31:29 保留，必须保持复位值。

位 28 **BKBID**：断路双向 (Break Bidirectional)

0：断路输入 BRK 为输入模式

1：断路输入 BRK 为双向模式

在双向模式下（BKBID 位置 1），断路输入配置为输入模式和开漏输出模式。任何激活的断路事件都将使断路输入上呈逻辑低电平，以向外部器件指示发生了内部断路事件。

注： *只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。*

注： *对此位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 27 保留，必须保持复位值。

位 26 **BKDSRM**：断路解除 (Break Disarm)

0：启动断路输入 BRK

1：解除断路输入 BRK

当断路源激活后，此位由硬件清零。

必须通过软件将 BKDSRM 位置 1 以释放双向输出控制（开漏输出处于高阻态），然后不断轮询此位，直到其由硬件复位，指示故障条件已消失。

注： *对此位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 25:20 保留，必须保持复位值。

位 19:16 保留，必须保持复位值。

位 15 **MOE**：主输出使能 (Main output enable)

只要断路输入变为有效状态，此位便由硬件异步清零。此位由软件置 1，也可根据 AOE 位状态自动置 1。此位仅对配置为输出的通道有效。

0：OC 和 OCN 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。

1：如果 OC 和 OCN 输出的相应使能位（TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。

有关详细信息，请参见 OC/OCN 使能说明（*第 843 页的第 26.4.8 节：TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 16 到 17）*）。

位 14 **AOE**：自动输出使能 (Automatic output enable)

0：MOE 只能由软件置 1

1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1（如果断路输入无效）

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 13 **BKP**：断路极性 (Break polarity)

0：断路输入 BRK 为低电平有效

1：断路输入 BRK 为高电平有效

*注：　**1.** 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。*

*　　**2.** 对此位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 12 **BKE**：断路使能 (Break enable)

0：禁止断路输入（BRK 和 CCS 时钟故障事件）

1：使能断路输入（BRK 和 CCS 时钟故障事件）

*注：　**1.** 编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。*

*　　**2.** 对此位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 11 **OSSR**：运行模式下的关闭状态选择 (Off-state selection for Run mode)

此位在 MOE = 1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。

有关详细信息，请参见 OC/OCN 使能说明（*第 843 页的第 26.4.8 节：TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 16 到 17）*）。

0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 AFIO 逻辑接管）。

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平（输出仍由定时器控制）。

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。

位 10 **OSSI**：空闲模式下的关闭状态选择 (Off-state selection for Idle mode)

此位在 MOE=0 时作用于配置为输出的通道。

有关详细信息，请参见 OC/OCN 使能说明（*第 843 页的第 26.4.8 节：TIMx 捕获/比较使能寄存器 (TIMx_CCER)（x = 16 到 17）*）。

0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号 = 0）

1：处于无效状态时，一旦 CCxE = 1 或 CCxNE = 1，便将 OC/OCN 输出首先强制为其空闲电平。然后设置 OC/OCN 使能输出信号=1

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。

位 9:8 **LOCK[1:0]**：锁定配置 (Lock configuration)

这些位用于针对软件错误提供写保护。

00：关闭锁定——不对任何位提供写保护。

01：锁定级别 1，此时无法对 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_BDTR 寄存器中的 BKE/BKP/AOE 位执行写操作。

10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（TIMx_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。

11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注：　复位后只能对 LOCK 位执行一次写操作。对 TIMx_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。

位 7:0 **DTG[7:0]**：配置死区发生器 (Dead-time generator setup)

此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。
DTG[7:5]=0xx => DT=DTG[7:0]x t_{dtg}，其中 $t_{dtg}=t_{DTS}$。
DTG[7:5]=10x => DT=(64+DTG[5:0])xt_{dtg}，其中 $T_{dtg}=2xt_{DTS}$。
DTG[7:5]=110 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=8xt_{DTS}$。
DTG[7:5]=111 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=16xt_{DTS}$。
示例：如果 T_{DTS}=125ns (8MHz)，则可能的死区值为：
0 到 15875 ns（步长为 125 ns）
16 µs 到 31750 ns（步长为 250 ns）
32 µs 到 63 µs（步长为 1 µs）
64 µs 到 126 µs（步长为 2 µs）

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位域即无法修改。

26.4.15 TIMx DMA 控制寄存器 (TIMx_DCR)（x = 16 到 17）

TIMx DMA control register

偏移地址：0x48

复位值：0x0000

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

位 15:13 保留，必须保持复位值。

位 12:8 **DBL[4:0]**：DMA 连续传送长度 (DMA burst length)

该 5 位向量定义了 DMA 的传送长度（当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送），即一个 DMA 要连续传送的次数。可按半字或字节进行传送（请参见下面的示例）。
00000：1 次传送
00001：2 次传送
00010：3 次传送
...
10001：18 次传送

位 7:5 保留，必须保持复位值。

位 4:0 **DBA[4:0]**：DMA 基址 (DMA base address)

该 5 位域定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。
示例：
00000：TIMx_CR1
00001：TIMx_CR2
00010：TIMx_SMCR
...

示例：以下面的传送为例：DBL = 7 次传送且 DBA = TIMx_CR1。这种情况下将向/从自 TIMx_CR1 地址开始的 7 个寄存器传输数据。

26.4.16 TIMx 全传输 DMA 地址 (TIMx_DMAR)（x = 16 到 17）

TIMx DMA address for full transfer

偏移地址：0x4C

复位值：0x0000

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

位 15:0 **DMAB[15:0]**：DMA 连续传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器

(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

26.4.17 TIM16 选项寄存器 1 (TIM16_OR1)

TIM16 option register 1

偏移地址：0x50

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **TI1_RMP[1:0]**：定时器 16 输入 1 连接 (Timer 16 input 1 connection)

此位由软件置 1 和清零。

00：TIM16 TI1 连接到 GPIO

01：TIM16 TI1 连接到 LSI

10：TIM16 TI1 连接到 LSE

11：TIM16 TI1 输入连接到 RTC 唤醒中断

26.4.18 TIM16 复用功能寄存器 1 (TIM16_AF1)

TIM16 alternate function register 1

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	Res.	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
				rw	rw	rw							rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11 **BKCMP2P**：BRK COMP2 输入极性 (BRK COMP2 input polarity)
此位选择 COMP2 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP2 输入为低电平有效
1：COMP2 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BKCMP1P**：BRK COMP1 输入极性 (BRK COMP1 input polarity)
此位选择 COMP1 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP1 输入为低电平有效
1：COMP1 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
此位选择 BKIN 复用功能输入灵敏度，必须与 BKP 极性位一起编程。
0：BKIN 输入为低电平有效
1：BKIN 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8:3 保留，必须保持复位值。

位 2 **BKCMP2E**：BRK COMP2 使能 (BRK COMP2 enable)
此位使能定时器 BRK 输入的 COMP2。COMP2 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BKCMP1E**：BRK COMP1 使能 (BRK COMP1 enable)
此位使能定时器 BRK 输入的 COMP1。COMP1 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
此位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

26.4.19 TIM16 输入选择寄存器 (TIM16_TISEL)

TIM16 input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM16_CH1 输入
其他值：保留

26.4.20 TIM17 选项寄存器 1 (TIM17_OR1)

TIM17 option register 1

偏移地址：0x50

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **TI1_RMP[1:0]**：定时器 17 输入 1 连接 (Timer 17 input 1 connection)
此位由软件置 1 和清零。
00：TIM17 TI1 连接到 GPIO
01：TIM17 TI1 连接到 MSI
10：TIM17 TI1 连接到 HSE/32
11：TIM17 TI1 连接到 MCO

26.4.21 TIM17 复用功能寄存器 1 (TIM17_AF1)

TIM17 alternate function register 1

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	Res.	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
				rw	rw	rw							rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11 **BKCMP2P**：BRK COMP2 输入极性 (BRK COMP2 input polarity)
此位选择 COMP2 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP2 输入为低电平有效
1：COMP2 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BKCMP1P**：BRK COMP1 输入极性 (BRK COMP1 input polarity)
此位选择 COMP1 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP1 输入为低电平有效
1：COMP1 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
此位选择 BKIN 复用功能输入灵敏度，必须与 BKP 极性位一起编程。
0：BKIN 输入为低电平有效
1：BKIN 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8:3 保留，必须保持复位值。

位 2 **BKCMP2E**：BRK COMP2 使能 (BRK COMP2 enable)
此位使能定时器 BRK 输入的 COMP2。COMP2 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BKCMP1E**：BRK COMP1 使能 (BRK COMP1 enable)
此位使能定时器 BRK 输入的 COMP1。COMP1 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
此位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

26.4.22 TIM17 输入选择寄存器 (TIM17_TISEL)

TIM17 input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)

0000：TIM17_CH1 输入

其他值：保留

26.4.23 TIM16/TIM17 寄存器映射

TIM16/TIM17 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 163. TIM16/TIM17 寄存器映射和复位值

偏移	寄存器名称	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
0x00	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0				0	0	0	0
0x04	**TIMx_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC
	Reset value																							0	0					0	0		0
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE
	Reset value																							0	0	0		0				0	0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF
	Reset value																							0		0		0				0	0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	Res.	COMG	Res.	Res.	Res.	CC1G	UG
	Reset value																									0		0				0	0
0x18	**TIMx_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																0										0	0	0	0	0	0	0
	TIMx_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC [1:0]		CC1S [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0x20	**TIMx_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
	Reset value																													0	0	0	0
0x24	**TIMx_CNT**	UIFCPY or Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIMx_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 163. TIM16/TIM17 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x30	**TIMx_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x34	**TIMx_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIMx_BDTR**	Res.	Res.	Res.	BKBID	Res.	BKDSRM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]							
	Reset value				0		0											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**TIMx_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIMx_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	**TIM16_OR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP [1:0]	
	Reset value																															0	0
0x50	**TIM17_OR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP [1:0]	
	Reset value																															0	0
0x60	**TIM16_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	Res.	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
	Reset value																					0	0	0							0	0	1
0x60	**TIM17_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	Res.	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
	Reset value																					0	0	0							0	0	1
0x68	**TIM16_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																													0	0	0	0
0x68	**TIM17_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

27 低功耗定时器 (LPTIM)

27.1 简介

LPTIM 是一个 16 位定时器，可从降低功耗的最终发展中受益。由于 LPTIM 的时钟源具有多样性，因此 LPTIM 能够在所有电源模式（待机模式除外）下保持运行状态。即使没有内部时钟源，LPTIM 也能运行，鉴于这一点，可将其用作“脉冲计数器”，这种脉冲计数器在某些应用中十分有用。此外，LPTIM 还能将系统从低功耗模式唤醒，因此非常适合实现“超时功能”，在这种功能模式下系统功耗极低。

LPTIM 引入了一个灵活的时钟方案，该方案能够提供所需的功能和性能，同时还能最大程度地降低功耗。

27.2 LPTIM 主要特性

- 16 位递增计数器
- 3 位预分频器，可采用 8 种分频系数（1、2、4、8、16、32、64 和 128）
- 可选时钟
  - 内部时钟源：LSE、LSI、HSI16 或 APB 时钟
  - LPTIM 输入的外部时钟源（在没有 LP 振荡器运行的情况下工作，可在使用脉冲计数器应用场景中使用）
- 16 位 ARR 自动重载寄存器
- 16 位比较寄存器
- 连续/单触发模式
- 可选软件/硬件输入触发
- 可编程数字防抖动干扰滤波器
- 可配置输出：脉冲和 PWM
- 可配置 I/O 极性
- 编码器模式

27.3 LPTIM 实现

表 164 介绍了 STM32WB55xx 器件上的 LPTIM 实现：LPTIM1 支持所有特性。LPTIM2 支持的特性略少，但在其余方面与 LPTIM1 完全相同。

表 164. STM32WB55xx LPTIM 特性

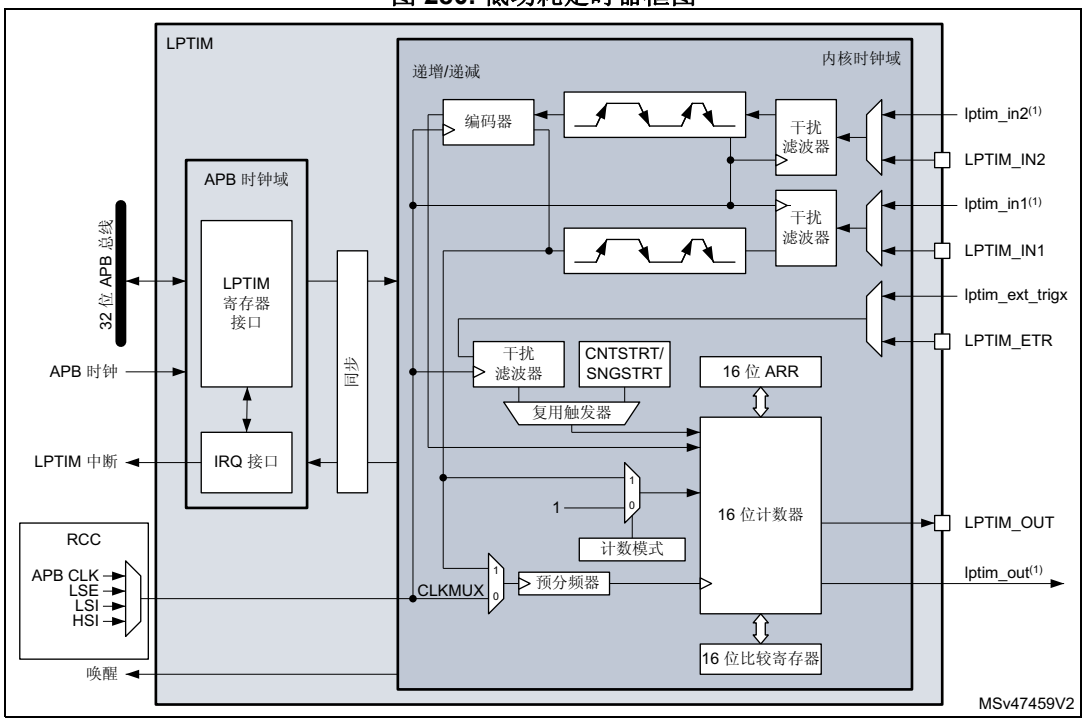
LPTIM 模式/特性(1)	LPTIM1	LPTIM2
编码器模式	X	-

1. X = 支持。

27.4 LPTIM 功能说明

27.4.1 LPTIM 框图

图 280. 低功耗定时器框图



1. lptim_in1 和 lptim_in2 分别为内部 LPTIM 输入 1 和输入 2 的信号，可以连接到内部外设。lptim_out 是内部 LPTIM 输出信号，可以连接到内部外设。

27.4.2 LPTIM 触发映射

下面详细介绍了 LPTIM 外部触发连接：

表 165. LPTIM1 外部触发器连接

TRIGSEL	外部触发信号
lptim_ext_trig0	GPIO
lptim_ext_trig1	RTC 闹钟 A
lptim_ext_trig2	RTC 闹钟 B
lptim_ext_trig3	RTC_TAMP1 输入检测
lptim_ext_trig4	RTC_TAMP2 输入检测
lptim_ext_trig5	RTC_TAMP3 输入检测
lptim_ext_trig6	COMP1_OUT
lptim_ext_trig7	COMP2_OUT

表 166. LPTIM2 外部触发器连接

TRIGSEL	外部触发信号
lptim_ext_trig0	GPIO
lptim_ext_trig1	RTC 闹钟 A
lptim_ext_trig2	RTC 闹钟 B
lptim_ext_trig3	RTC_TAMP1 输入检测
lptim_ext_trig4	RTC_TAMP2 输入检测
lptim_ext_trig5	RTC_TAMP3 输入检测
lptim_ext_trig6	COMP1_OUT
lptim_ext_trig7	COMP2_OUT

27.4.3 LPTIM 复位和时钟

LPTIM 可通过多个时钟源提供时钟。它可以由内部时钟信号提供时钟，内部时钟信号可通过复位和时钟控制器 (RCC) 在 APB、LSI、LSE 或 HSI16 时钟源中进行选择。此外，LPTIM 还可通过注入到其外部 Input1 上的外部时钟信号提供时钟。当通过外部时钟源提供时钟时，LPTIM 可以在下述两种可能配置中的其中一种配置下运行：

- 第一种配置是，LPTIM 通过外部信号提供时钟，但同时通过 APB 或 LSE、LSI 和 HSI16 等任何其他内置振荡器为 LPTIM 提供内部时钟信号。
- 第二种配置是，LPTIM 仅由外部时钟源通过外部 Input1 提供时钟。此配置可在进入低功耗模式后所有内置振荡器关闭时，用于实现超时功能或脉冲计数器功能。

对 CKSEL 和 COUNTMODE 位进行编程，可控制 LPTIM 使用外部时钟源还是内部时钟源。

当使用外部时钟源时，可使用 CKPOL 位选择外部时钟信号的有效边沿。如果上升沿和下降沿均为有效边沿，则还应提供内部时钟信号（第一种配置）。在这种情况下，内部时钟信号频率应至少为外部时钟信号频率的五倍。

27.4.4 干扰滤波器

外部（映射到 GPIO）或内部（映射到芯片级或其他嵌入式外设，例如嵌入式比较器）LPTIM 输入由数字滤波器保护，避免任何毛刺和噪声干扰在 LPTIM 内部传播，从而防止产生意外计数或触发。

在激活数字滤波器之前，首先应向 LPTIM 提供内部时钟源，这是保证滤波器正常工作的必要条件。

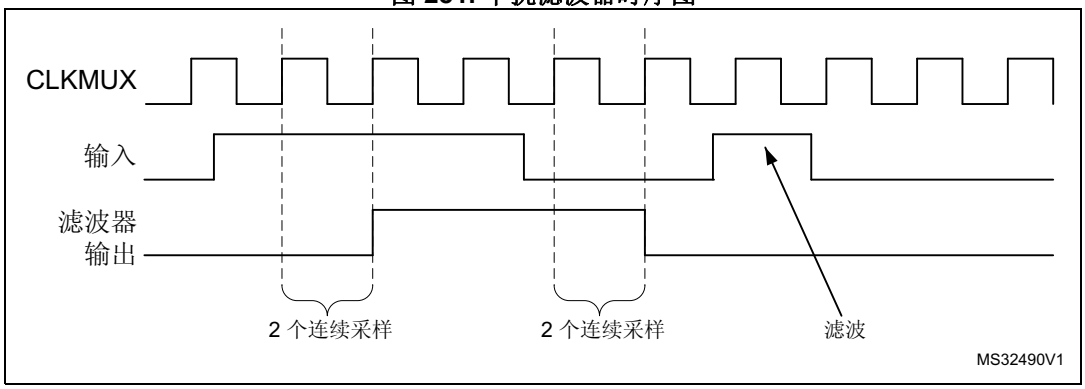
数字滤波器分为两组：

- 第一组数字滤波器保护 LPTIM 外部输入。数字滤波器的敏感性由 CKFLT 位控制。
- 第二组数字滤波器保护 LPTIM 内部触发输入。数字滤波器的敏感性由 TRGFLT 位控制。

注： *数字滤波器的敏感性以组为单位进行控制。无法单独配置同一组内各个数字滤波器的敏感性。*

滤波器的敏感性会影响相同的连续采样的数量，在其中一个 LPTIM 输入上检测到此类连续采样时，才能将某信号电平变化视为有效跳变。*图 281* 给出了编程 2 个连续采样时，干扰滤波器行为的示例。

图 281. 干扰滤波器时序图



注： *不提供内部时钟信号时，必须通过将 CKFLT 和 TRGFLT 位设为 0 来停用数字滤波器。在这种情况下，可使用外部模拟滤波器来防止 LPTIM 外部输入产生干扰。*

27.4.5 预分频器

LPTIM 16 位计数器前面要有一个可配置的 2 次幂预分频器。预分频器的分频比由 PRESC[2:0] 3 位域进行控制。下表列出了所有可能的分频比：

表 167. 预分频器的分频比

编程	分频系数
000	/1
001	/2
010	/4
011	/8
100	/16
101	/32
110	/64
111	/128

27.4.6 触发多路复用器

LPTIM 计数器可通过软件启动，也可以在 8 个触发输入之一上检测到有效边沿后启动。

TRIGEN[1:0] 用于确定 LPTIM 触发源：

- TRIGEN[1:0] 等于“00”时，LPTIM 计数器会在通过软件将 CNTSTRT 位或 SNGSTRT 位其中之一置 1 后立即启动。TRIGEN[1:0] 的其余三个可能的值用于配置触发输入使用的有效边沿。LPTIM 计数器会在检测到有效边沿后立即启动。
- TRIGEN[1:0] 不等于“00”时，TRIGSEL[2:0] 用于选择使用 8 个触发输入中的哪一个来启动计数器。

外部触发信号视为 LPTIM 的异步信号。因此，检测到触发信号后，由于同步问题，需要延迟两个计数器时钟周期，定时器才能开始运行。

如果在定时器已启动时发生新的触发事件，则此事件将被忽略（除非已使能超时功能）。

注： *必须使能定时器，才能将 SNGSTRT/CNTSTRT 位置 1。当定时器禁止时，对这些位执行的任何写操作都将被硬件丢弃。*

27.4.7 工作模式

LPTIM 支持以下两种工作模式：

- 连续模式：定时器自由运行，由触发事件启动并且直到被禁止才会停止。
- 单触发模式：定时器由触发事件启动，当达到 ARR 值时停止。

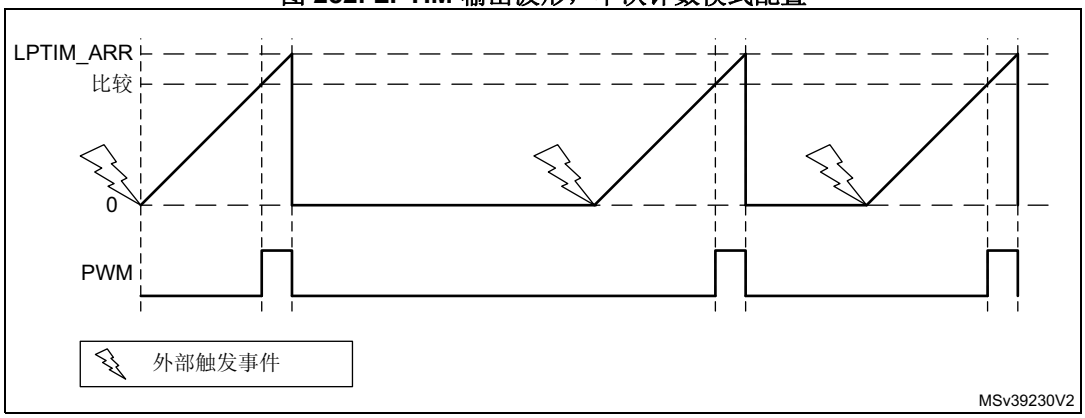
单触发模式

要使能单触发计数，必须将 SNGSTRT 位置 1。

新的触发事件将重新启动定时器。从计数器启动到计数器达到 ARR，这段时间内发生的任何触发事件均将被丢弃。

选择外部触发时，在 SNGSTRT 位置 1 后以及计数器寄存器停止后（包含零值）到达的每个外部触发事件都将为计数器启动新的单触发计数周期，如*图 282* 所示。

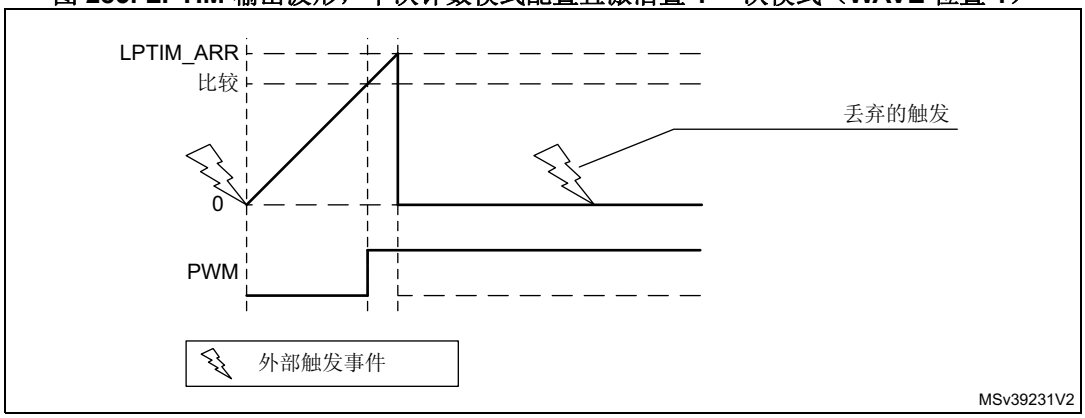
图 282. LPTIM 输出波形，单次计数模式配置



- 置 1 一次模式激活：

应注意，当 LPTIM_CFGR 寄存器中的 WAVE 位域置 1 时，将激活置 1 一次模式。在这种情况下，计数器仅会在第一个触发事件后启动一次，任何后续触发事件都将被丢弃，如*图 283* 所示。

图 283. LPTIM 输出波形，单次计数模式配置且激活置 1 一次模式（WAVE 位置 1）



若通过软件启动（TRIGEN[1:0] = “00”），将 SNGSTRT 置 1 会使计数器进行单触发计数。

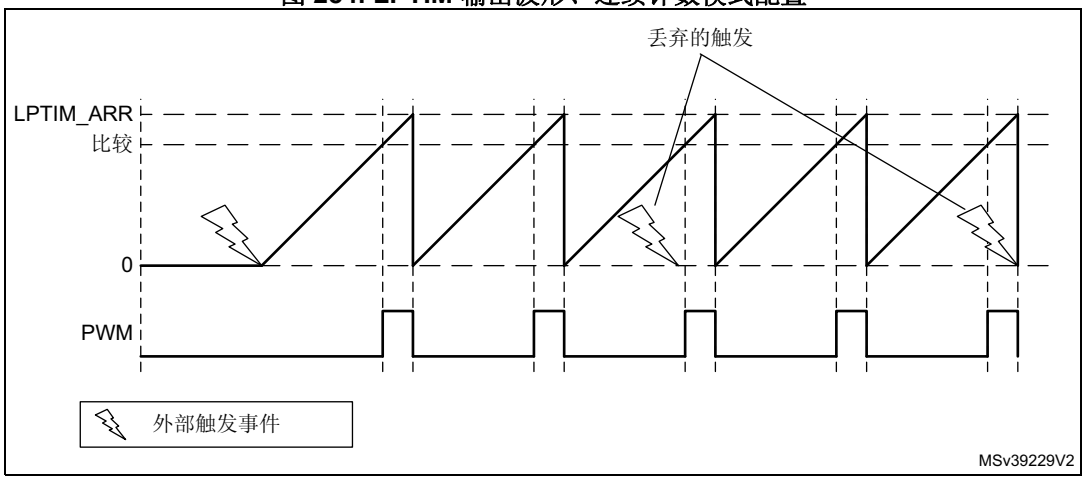
连续模式

要使能连续计数，必须将 CNTSTRT 位置 1。

若选择外部触发，则在 CNTSTRT 置 1 后到达的外部触发事件将启动计数器进行连续计数。任何后续的外部触发事件都将被丢弃，如*图 284* 所示。

若通过软件启动（TRIGEN[1:0] = “00”），将 CNTSTRT 置 1 会使计数器开始连续计数。

图 284. LPTIM 输出波形、连续计数模式配置



SNGSTRT 和 CNTSTRT 位只能在定时器使能时（ENABLE 位置 1）置 1。可以“实时”从单触发模式切换为连续模式。

如果之前选择的是连续模式，则将 SNGSTRT 置 1 会使 LPTIM 切换为单触发模式。计数器（激活时）将在达到 ARR 后立即停止。

如果之前选择的是单触发模式，则将 CNTSTRT 置 1 会使 LPTIM 切换为连续模式。计数器（激活时）将在达到 ARR 后立即重新启动。

27.4.8 超时功能

若在一个选定的触发输入上检测到有效边沿，则可用于复位 LPTIM 计数器。该功能通过 TIMOUT 位进行控制。

第一个触发事件将启动定时器，任何后续的触发事件将复位计数器，且定时器将重新启动。

可实现低功耗超时功能。超时值对应于比较值；如果在预期的时间帧内未发生触发事件，MCU 将由比较匹配事件唤醒。

27.4.9 生成波形

两个 16 位寄存器、LPTIM_ARR（自动重载寄存器）和 LPTIM_CMP（比较寄存器）用于在 LPTIM 输出上生成多个不同的波形

定时器可生成以下波形：

- PWM 模式：只要 LPTIM_CNT 中的计数器值超过 LPTIM_CMP 中的比较值，LPTIM 输出就会置位。只要 LPTIM_ARR 和 LPTIM_CNT 寄存器之间发生匹配，LPTIM 输出就会复位。
- 单脉冲模式：对于第一个脉冲，输出波形与 PWM 模式输出波形类似，随后输出将永久复位。
- 置 1 一次模式：除输出保持最后一个信号电平外（取决于配置的输出极性），输出波形与单脉冲模式输出波形类似。

上述模式要求 LPTIM_ARR 寄存器的值严格大于 LPTIM_CMP 寄存器的值。

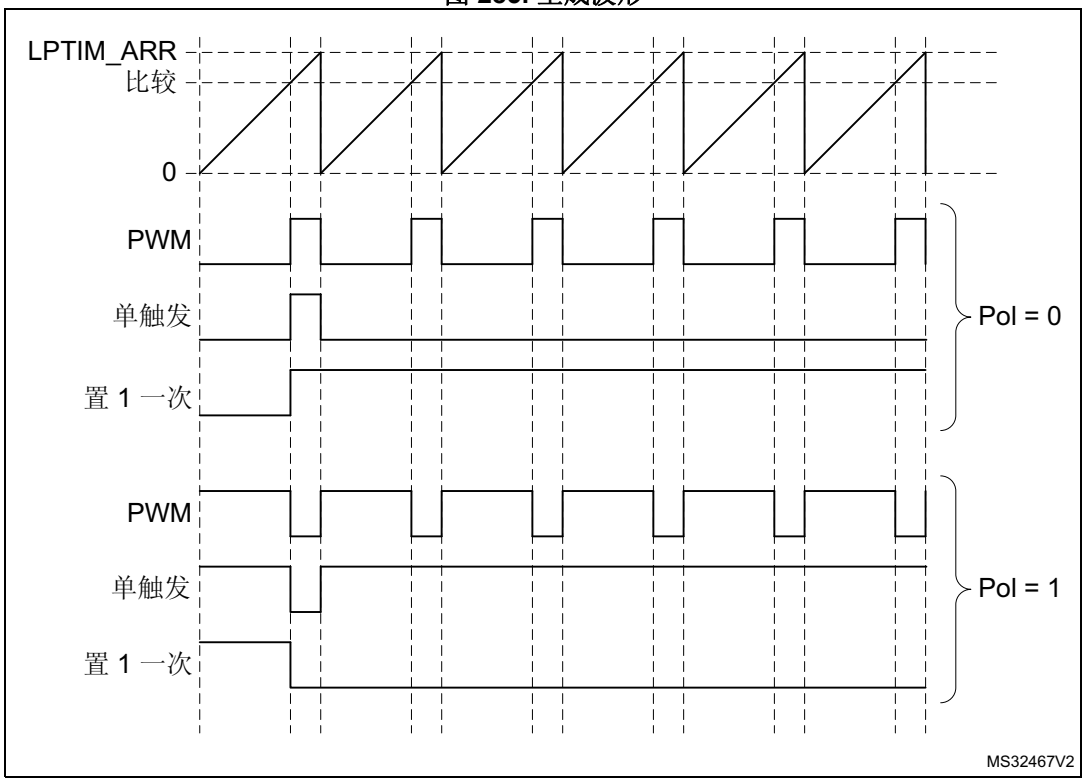
LPTIM 输出波形可通过 WAVE 位配置，具体如下：

- 若将 WAVE 位复位为 0，则会强制 LPTIM 生成 PWM 波形或单脉冲波形，具体取决于将哪个位（CNTSTRT 或 SNGSTRT）置 1。
- 若将 WAVE 位置 1，则会强制 LPTIM 生成置 1 一次模式波形。

WAVPOL 位控制 LPTIM 输出极性。更改立即生效，因此输出默认值将在极性重新配置后立即更改，甚至会在定时器使能前进行更改。

生成的信号的频率高达 LPTIM 时钟频率 2 分频。*图 285* 给出了可能在 LPTIM 输出上生成的三种波形。此外，此图还显示了通过 WAVPOL 位更改极性所产生的效果。

图 285. 生成波形



27.4.10 寄存器更新

LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器在 APB 总线写操作后会立即更新，若定时器已启动，也可在当前周期结束时更新。

PRELOAD 位控制 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的更新方式：

- 当 PRELOAD 位复位为 0 时，LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器会在写访问后立即更新。
- 当 PRELOAD 位置 1 时，若定时器已启动，LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器会在当前周期结束时更新。

LPTIM APB 接口和 LPTIM 内核逻辑使用的时钟不同，因此在 APB 写操作后，需要经过一定的延迟，写入值才能用于计数器比较器。在此延迟期间，必须避免向这些寄存器执行其他写操作。

LPTIM_ISR 寄存器中的 ARROK 标志和 CMPOK 标志分别指示 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的写操作已完成。

向 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器执行写操作后，只有在前一次写操作完成后，才能对同一寄存器执行新的写操作。在 ARROK 标志或 CMPOK 标志置 1 前执行连续的写操作将造成无法预知的结果。

27.4.11 计数器模式

LPTIM 计数器可用于对 LPTIM Input1 上的外部事件进行计数，也可用于对内部时钟周期进行计数。CKSEL 位和 COUNTMODE 位用于控制将使用哪些源更新计数器。

若使用 LPTIM 对 Input1 上的外部事件进行计数，计数器可在上升沿、下降沿或两种边沿进行更新，具体取决于写入 CKPOL[1:0] 位的值。

根据 CKSEL 和 COUNTMODE 值，可选择以下计数模式：

- CKSEL = 0：LPTIM 由内部时钟源提供时钟
  - COUNTMODE = 0

    LPTIM 配置为由内部时钟源提供时钟，LPTIM 计数器配置为在每个内部时钟脉冲后进行更新。
  - COUNTMODE = 1

    LPTIM 外部 Input1 通过提供给 LPTIM 的内部时钟进行采样。

    因此，为了不丢失任何事件，外部 Input1 信号变化的频率决不应超过提供给 LPTIM 的内部时钟的频率。此外，不得对提供给 LPTIM 的内部时钟进行预分频 (PRESC[2:0] = 000)。
- CKSEL = 1：LPTIM 由外部时钟源提供时钟

  COUNTMODE 值不相关。

  在这种配置下，LPTIM 无需内部时钟源（已使能干扰滤波器时除外）。注入到 LPTIM 外部 Input1 的信号用作 LPTIM 的系统时钟。此配置适合未使能任何内置振荡器的工作模式。

  对于这种配置，LPTIM 计数器可以在 input1 时钟信号的上升沿或下降沿进行更新，但不可在上升沿和下降沿均更新。

  由于注入到 LPTIM 外部 Input1 的信号也可用于为 LPTIM 内核逻辑提供时钟，计数器递增计数前存在一些初始延时（使能 LPTIM 后）。更确切地说，LPTIM 外部 Input1 的前五个有效边沿将丢失（使能 PTIM 后）。

27.4.12 定时器使能

LPTIM_CR 寄存器中的 ENABLE 位用于使能/禁止 LPTIM 内核逻辑。将 ENABLE 位置 1 后，需要延迟两个计数器时钟周期，才能真正使能 LPTIM。

LPTIM_CFGR 和 LPTIM_IER 寄存器必须在禁止 LPTIM 后才能修改。

27.4.13 定时器计数器复位

为了将 LPTIM_CNT 寄存器的内容复位为零，实现了两种复位机制：

- 同步复位机制：同步复位由 LPTIM_CR 寄存器中的 COUNTRST 位控制。在将 COUNTRST 位域置 1 后，复位信号在 LPTIM 内核时钟域中传播。因此，重要的是要注意，要在经历几个 LPTIM 内核逻辑时钟脉冲之后再考虑复位。这将使 LPTIM 计数器在复位触发和生效之间额外计数几个脉冲。由于 COUNTRST 位位于 APB 时钟域中，并且 LPTIM 计数器位于 LPTIM 内核时钟域中，因此当将 1 写入到 COUNTRST 位时，内核时钟需要 3 个时钟周期的延迟用以同步由 APB 时钟域发出的复位信号。
- 异步复位机制：异步复位由位于 LPTIM_CR 寄存器中的 RSTARE 位控制。当该位置 1 时，对 LPTIM_CNT 寄存器的任何读访问都会将其内容复位为零。应在不提供 LPTIM 内核时钟的时间范围内触发异步复位。例如，当 LPTIM Input1 管脚为外部时钟输入管脚时，只有当足够保证 LPTIM Input1 不会发生反转时，才应用异步复位。
  应注意的是，为了实现可靠的 LPTIM_CNT 寄存器内容读取，必须执行两次连续的读访问并进行比较。当两次读访问的值相等时，可认为读访问可靠。然而，当使能了异步复位时，不可能两次读取 LPTIM_CNT 寄存器。

警告： LPTIM 内没有防止两个复位机制同时使用的机制。所以开发人员应该确保这两个机制是排斥地使用。

27.4.14 编码器模式

此模式用于处理来自正交编码器的信号，此正交编码器用于检测旋转元件的角度位置。编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 LPTIM_ARR 寄存器中编程的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此必须在启动之前配置 LPTIM_ARR。通过两个外部输入信号 Input1 和 Input2 生成时钟信号作为 LPTIM 计数器时钟。这两个信号间的相位确定计数方向。

仅当 LPTIM 由内部时钟源提供时钟时才可使用编码器模式。Input1 和 Input2 输入上的信号频率不得超过 LPTIM 内部时钟频率 4 分频。必须满足此条件才能确保 LPTIM 正常工作。

方向变化由 LPTIM_ISR 寄存器中的两个递减和递增标志指示。此外，如果通过 DOWNIE 位使能，还可为两种方向变化事件产生中断。

要激活编码器模式，必须将 ENC 位置 1。LPTIM 必须首先配置为连续模式。

当编码器模式激活时，LPTIM 计数器按照增量编码器的速度和方向自动修改。因此，其内容始终代表编码器的位置。计数方向由递增和递减标志指示，对应于编码器转子的旋转方向。

根据使用 CKPOL[1:0] 位配置的边沿敏感性，可得几种不同的计数方案。下表汇总了可能的组合（假设 Input1 和 Input2 不同时切换）。

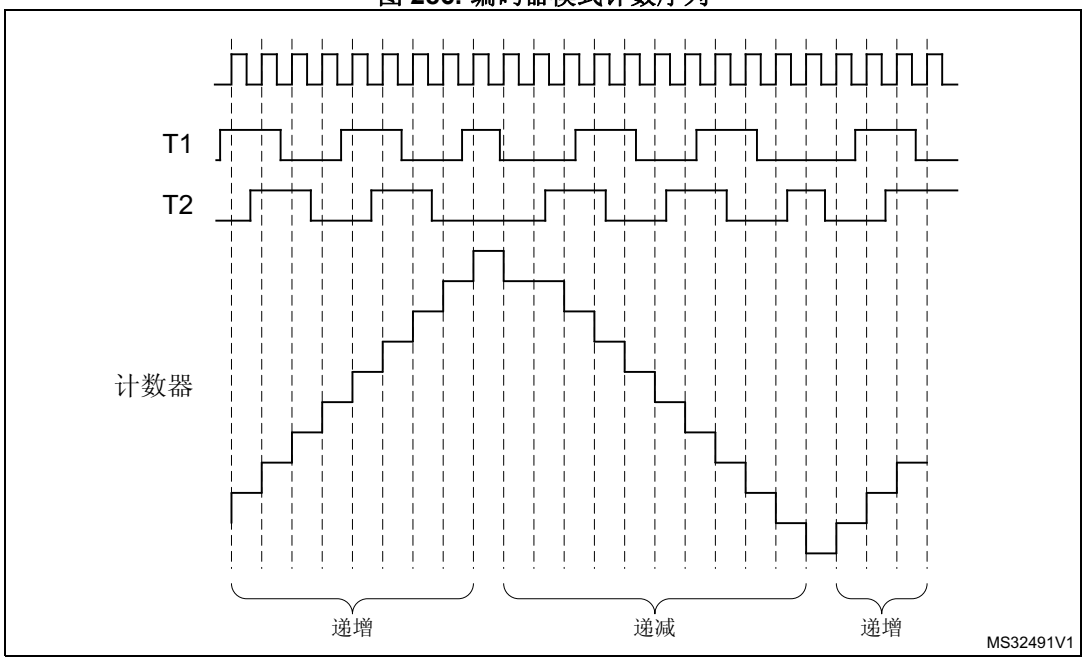
表 168. 编码器计数方案

有效边沿	相反信号的电平（Input1 对应 Input2，Input2 对应 Input1）	Input1 信号		Input2 信号	
		上升	下降	上升	下降
上升沿	高	递减	不计数	递增	不计数
	低	递增	不计数	递减	不计数
下降沿	高	不计数	递增	不计数	递减
	低	不计数	递减	不计数	递增
两种边沿	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

下图所示为编码器模式下配置了两种边沿敏感性的计数序列。

注意： 在此模式下，LPTIM 必须由内部时钟源提供时钟，因此 CKSEL 位必须保持其复位值 0。另外，预分频器分频比必须等于其复位值 1（PRESC[2:0] 位必须为“000”）。

图 286. 编码器模式计数序列



27.4.15 调试模式

当微控制器进入调试模式时（内核停止），LPTIM 计数器会根据 DBG 模块中的 DBG_LPTIM_STOP 配置位选择继续正常工作或者停止工作。

27.5 LPTIM 低功耗模式

表 169. 低功耗模式对 LPTIM 的影响

模式	说明
睡眠	无影响。LPTIM 中断可使器件退出睡眠模式。
低功耗运行	无影响。
低功耗睡眠	无影响。LPTIM 中断可使器件退出低功耗睡眠模式。
停止 0/停止 1	当 LPTIM 由 LSE 或 LSI 提供时钟信号时无影响。LPTIM 中断会导致器件退出停止 0 和停止 1。
停止 2	当 LPTIM1 由 LSE 或 LSI 提供时钟信号时对 LPTIM1 无影响。LPTIM1 中断会导致器件退出停止 2。LPTIM2 寄存器的内容会保留，但是在进入停止 2 之前必须禁止 LPTIM2。
待机	LPTIM 外设掉电，退出待机模式或关断模式后必须重新初始化。
关断	

27.6 LPTIM 中断

若以下事件通过 LPTIM_IER 寄存器使能，则这些事件会生成中断/唤醒事件：

- 比较匹配
- 自动重载匹配（编码器模式下无论哪种方向）
- 外部触发事件
- 自动重载寄存器写操作完成
- 比较寄存器写操作完成
- 方向变化（编码器模式），可编程（递增/递减/同时递增和递减）

注： *只要 LPTIM_IER 寄存器（中断使能寄存器）中的位在 LPTIM_ISR 寄存器（状态寄存器）中相应标志置 1 后置 1，就不会触发中断。*

表 170. 中断事件

中断事件	说明
比较匹配	当计数器寄存器 (LPTIM_CNT) 的内容与比较寄存器 (LPTIM_CMP) 的内容匹配时，生成中断标志。
自动重载匹配	当计数器寄存器 (LPTIM_CNT) 的内容与自动重载寄存器 (LPTIM_ARR) 的内容匹配时，生成中断标志。
外部触发事件	当检测到外部触发事件时，会生成中断标志。
自动重载寄存器更新成功	当对 LPTIM_ARR 寄存器的写操作完成时，生成中断标志。
比较寄存器更新成功	当对 LPTIM_CMP 寄存器的写操作完成时，生成中断标志。
方向更改	用于编码器模式。嵌入两个中断标志以发出方向更改的信号： – UP 标志发出递增计数方向更改的信号。 – DOWN 标志发出递减计数方向更改的信号。

27.7 LPTIM 寄存器

27.7.1 LPTIM 中断和状态寄存器 (LPTIM_ISR)

LPTIM interrupt and status register

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWN	UP	ARR OK	CMP OK	EXT TRIG	ARRM	CMPM
									r	r	r	r	r	r	r

位 31:7 保留，必须保持复位值。

位 6 **DOWN**：计数方向从递增变为递减 (Counter direction change up to down)

在编码器模式下，由硬件将 DOWN 位置 1 时，会通知应用计数方向由递增变为递减。可以通过向 LPTIM_ICR 寄存器中的 DOWNCF 位写入 1 来将 DOWN 标志清零。

注： 如果 LPTIM 不支持编码器模式功能，则保留该位。请参见第 27.3 节：LPTIM 实现。

位 5 **UP**：计数方向从递减变为递增 (Counter direction change down to up)

在编码器模式下，由硬件将 UP 位置 1 时，会通知应用计数方向由递减变为递增。可以通过向 LPTIM_ICR 寄存器中的 UPCF 位写入 1 来将 UP 标志清零。

注： 如果 LPTIM 不支持编码器模式功能，则保留该位。请参见第 27.3 节：LPTIM 实现。

位 4 **ARROK**：自动重载寄存器更新成功 (Autoreload register update OK)

由硬件将 ARROK 置 1 时，会通知应用 LPTIM_ARR 寄存器的 APB 总线写操作已成功完成。可以通过向 LPTIM_ICR 寄存器中的 ARROKCF 位写入 1 将 ARROK 标志清零。

位 3 **CMPOK**：比较寄存器更新成功 (Compare register update OK)

由硬件将 CMPOK 置 1 时，会通知应用 LPTIM_CMP 寄存器的 APB 总线写操作已成功完成。

位 2 **EXTTRIG**：外部触发边沿事件 (External trigger edge event)

由硬件将 EXTTRIG 置 1 时，会通知应用所选的外部触发输入上产生有效边沿。如果由于定时器已启动而忽略触发事件，则不会将此标志置 1。可以通过向 LPTIM_ICR 寄存器中的 EXTTRIGCF 位写入 1 来将 EXTTRIG 标志清零。

位 1 **ARRM**：自动重载匹配 (Autoreload match)

由硬件将 ARRM 置 1 时，会通知应用 LPTIM_CNT 寄存器的值已达到 LPTIM_ARR 寄存器的值。可以通过向 LPTIM_ICR 寄存器中的 ARRMCF 位写入 1 来将 ARRM 标志清零。

位 0 **CMPM**：比较匹配 (Compare match)

由硬件将 CMPM 位置 1 时，会通知应用 LPTIM_CNT 寄存器的值已达到 LPTIM_CMP 寄存器的值。

27.7.2 LPTIM 中断清零寄存器 (LPTIM_ICR)

LPTIM interrupt clear register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNCF	UPCF	ARROKCF	CMPOKCF	EXTTRIGCF	ARRMCF	CMPMCF
									w	w	w	w	w	w	w

位 31:9 保留，必须保持复位值。

位 8:7 保留，必须保持复位值。

位 6 **DOWNCF**：方向变为递减清零标志 (Direction change to down clear flag)
将 1 写入此位时，LPTIM_ISR 寄存器中的 DOWN 标志将清零。
注： 如果 LPTIM 不支持编码器模式功能，则保留该位。请参见第 27.3 节：LPTIM 实现。

位 5 **UPCF**：方向变为递增清零标志 (Direction change to UP clear flag)
将 1 写入此位时，LPTIM_ISR 寄存器中的 UP 标志将清零。
注： 如果 LPTIM 不支持编码器模式功能，则保留该位。请参见第 27.3 节：LPTIM 实现。

位 4 **ARROKCF**：自动重载寄存器更新成功清零标志 (Autoreload register update OK clear flag)
将 1 写入此位时，LPTIM_ISR 寄存器中的 ARROK 标志将清零。

位 3 **CMPOKCF**：比较寄存器更新成功清零标志 (Compare register update OK clear flag)
将 1 写入此位时，LPTIM_ISR 寄存器中的 CMPOK 标志将清零。

位 2 **EXTTRIGCF**：外部触发有效边沿清零标志 (External trigger valid edge clear flag)
将 1 写入此位时，LPTIM_ISR 寄存器中的 EXTTRIG 标志将清零。

位 1 **ARRMCF**：自动重载匹配清零标志 (Autoreload match clear flag)
将 1 写入此位时，LPTIM_ISR 寄存器中的 ARRM 标志将清零。

位 0 **CMPMCF**：比较匹配清零标志 (Compare match clear flag)
将 1 写入此位时，LPTIM_ISR 寄存器中的 CMP 标志将清零。

27.7.3 LPTIM 中断使能寄存器 (LPTIM_IER)

LPTIM interrupt enable register

偏移地址：0x008

复位值：0x0000 0000

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

位 31:9 保留，必须保持复位值。

位 8:7 保留，必须保持复位值。

位 6 **DOWNIE**：方向变为递减中断使能 (Direction change to down Interrupt Enable)
0：禁止 DOWN 中断
1：使能 DOWN 中断
注： 如果 LPTIM 不支持编码器模式功能，则保留该位。请参见第 27.3 节：LPTIM 实现。

位 5 **UPIE**：方向变为递增中断使能 (Direction change to UP Interrupt Enable)
0：禁止 UP 中断
1：使能 UP 中断
注： 如果 LPTIM 不支持编码器模式功能，则保留该位。请参见第 27.3 节：LPTIM 实现。

位 4 **ARROKIE**：自动重载寄存器更新成功中断使能 (Autoreload register update OK Interrupt Enable)
0：禁止 ARROK 中断
1：使能 ARROK 中断

位 3 **CMPOKIE**：比较寄存器更新成功中断使能 (Compare register update OK Interrupt Enable)
0：禁止 CMPOK 中断
1：使能 CMPOK 中断

位 2 **EXTTRIGIE**：外部触发有效边沿中断使能 (External trigger valid edge Interrupt Enable)
0：禁止 EXTTRIG 中断
1：使能 EXTTRIG 中断

位 1 **ARRMIE**：自动重载匹配中断使能 (Autoreload match Interrupt Enable)
0：禁止 ARRM 中断
1：使能 ARRM 中断

位 0 **CMPMIE**：比较匹配中断使能 (Compare match Interrupt Enable)
0：禁止 CMPM 中断
1：使能 CMPM 中断

注意： 必须在 LPTIM 已禁止时（ENABLE 位复位为 0）才能修改 LPTIM_IER 寄存器

27.7.4 LPTIM 配置寄存器 (LPTIM_CFGR)

LPTIM configuration register

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENC	COUNT MODE	PRELOAD	WAVPOL	WAVE	TIMOUT	TRIGEN[1:0]		Res.
							rw	rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIGSEL[2:0]			Res.	PRESC[2:0]			Res.	TRGFLT[1:0]		Res.	CKFLT[1:0]		CKPOL[1:0]		CKSEL
rw	rw	rw		rw	rw	rw		rw	rw		rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29 保留，必须保持复位值。

位 28:25 保留，必须保持复位值。

位 24 **ENC**：编码器模式使能 (Encoder mode enable)

ENC 位控制编码器模式

0：禁止编码器模式

1：使能编码器模式

注： 如果 LPTIM 不支持编码器模式功能，则保留该位。请参见第 27.3 节：LPTIM 实现。

位 23 **COUNTMODE**：计数器模式使能 (counter mode enabled)

COUNTMODE 位用于选择 LPTIM 使用哪个时钟源来为计数器提供时钟：

0：计数器在每个内部时钟脉冲后递增

1：计数器在 LPTIM 外部 Input1 上的每个有效时钟脉冲后递增

位 22 **PRELOAD**：寄存器更新模式 (Registers update mode)

PRELOAD 位控制 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的更新方式

0：寄存器在每次 APB 总线写访问后更新

1：寄存器在当前 LPTIM 周期结束时更新

位 21 **WAVPOL**：波形极性 (Waveform shape polarity)

WAVEPOL 位控制输出极性

0：LPTIM 输出反映 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器之间的比较结果

1：LPTIM 输出反映与 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器之间的比较结果相反的值

位 20 **WAVE**：波形 (Waveform shape)

WAVE 位控制输出波形

0: 停用置 1 一次模式，并输出 PWM 或单脉冲波形，具体波形取决于定时器的启动方式，即 CNTSTRT（对于 PWM 波形）或 SNGSTRT （对于单脉冲波形）。

1： 激活置 1 一次模式

位 19 **TIMOUT**：超时使能 (Timeout enable)

TIMOUT 位控制超时功能

0： 定时器已启动时到达的触发事件将被忽略

1： 定时器已启动时到达的触发事件将复位并重新启动计数器

位 18:17 **TRIGEN[1:0]**：触发使能和极性 (Trigger enable and polarity)

TRIGEN 位控制 LPTIM 计数器是否由外部触发信号启动。如果已选择由外部触发信号启动，触发有效边沿的配置有以下三种：

00：软件触发（由软件启动计数）

01：上升沿为有效边沿

10：下降沿为有效边沿

11：上升沿和下降沿均为有效边沿

位 16 保留，必须保持复位值。

位 15:13 **TRIGSEL[2:0]**：触发源选择器 (Trigger selection)

TRIGSEL 位用于选择作为 LPTIM 触发事件的触发源，可用触发源包括以下 8 种：

000：lptim_ext_trig0

001：lptim_ext_trig1

010：lptim_ext_trig2

011：lptim_ext_trig3

100：lptim_ext_trig4

101：lptim_ext_trig5

110：lptim_ext_trig6

111：lptim_ext_trig7

详细信息，请参见*第 27.4.2 节：LPTIM 触发映射*。

位 12 保留，必须保持复位值。

位 11:9 **PRESC[2:0]**：时钟预分频器 (Clock prescaler)

PRESC 位配置预分频器的分频系数。分频系数可从以下分频系数中选择：

000： /1

001： /2

010： /4

011： /8

100： /16

101： /32

110： /64

111： /128

位 8 保留，必须保持复位值。

位 7:6 **TRGFLT[1:0]**：触发信号的可配置数字滤波器 (Configurable digital filter for trigger)

TRGFLT 值用于设置连续相同采样的数量，若在内部触发信号电平发生变化时检测到此类连续采样，才会将此电平变化视为有效电平切换。必须存在内部时钟源才能使用此功能。

00：任何触发信号有效电平变化均视为有效触发。

01：触发信号有效电平变化必须至少稳定 2 个时钟周期，才能将其视为有效触发。

10：触发信号有效电平变化必须至少稳定 4 个时钟周期，才能将其视为有效触发。

11：触发信号有效电平变化必须至少稳定 8 个时钟周期，才能将其视为有效触发。

位 5 保留，必须保持复位值。

位 4:3 **CKFLT[1:0]**：外部时钟的可配置数字滤波器 (Configurable digital filter for external clock)

CKFLT 值用于设置连续相同采样的数量，若在外部时钟信号电平发生变化时检测到此类连续采样，才会将此电平变化视为有效电平切换。必须存在内部时钟源才能使用此功能。

00：任何外部时钟信号电平变化均视为有效切换。

01：外部时钟信号电平变化必须至少稳定 2 个时钟周期，才能将其视为有效切换。

10：外部时钟信号电平变化必须至少稳定 4 个时钟周期，才能将其视为有效切换。

11：外部时钟信号电平变化必须至少稳定 8 个时钟周期，才能将其视为有效切换。

位 2:1 **CKPOL[1:0]**：时钟极性 (Clock Polarity)

如果 LPTIM 由外部时钟源提供时钟：

当 LPTIM 由外部时钟源提供时钟时，CKPOL 位用于配置计数所使用的有效边沿：

00：上升沿为用于计数的有效边沿。
如果将 LPTIM 配置为编码器模式（ENC 位置 1），则编码器子模式 1 有效。

01：下降沿为用于计数的有效边沿。
如果将 LPTIM 配置为编码器模式（ENC 位置 1），则编码器子模式 2 有效。

10：上升沿和下降沿均为有效边沿。当外部时钟信号的上升沿和下降沿均视为有效边沿时，LPTIM 还必须由内部时钟源提供时钟，且内部时钟源频率至少等于外部时钟频率的四倍。
如果将 LPTIM 配置为编码器模式（ENC 位置 1），则编码器子模式 3 有效。

11：不允许

有关编码器模式子模式的更多详细信息，请参见*第 27.4.14 节：编码器模式*。

位 0 **CKSEL**：时钟选择器 (Clock selector)

CKSEL 位选择 LPTIM 将使用的时钟源：

0：LPTIM 由内部时钟源（APB 时钟或任意内置振荡器）提供时钟

1：LPTIM 由外部时钟源通过 LPTIM 外部 Input1 提供时钟

注意： 必须在 LPTIM 已禁止时（ENABLE 位复位为 0）才能修改 LPTIM_CFGR 寄存器。

27.7.5 LPTIM 控制寄存器 (LPTIM_CR)

LPTIM control register

偏移地址：0x010

复位值：0x0000 0000

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

位 31:5 保留，必须保持复位值。

位 4 **RSTARE**：读操作后复位使能 (Reset after read enable)

此位由软件置 1 和清零。当 RSTARE 置 1 时，对 LPTIM_CNT 寄存器的任何读取访问都将异步复位 LPTIM_CNT 寄存器的内容。

注意： 只能对该位域执行写操作。这意味着该位不能被读回以验证已写入的值。例如，如果该位设置为 1，即使使能了“读操作后复位”功能（由于该位域以前已写入 1），尝试读取它也将返回 0。要关闭“读操作后复位”或确保已经关闭，应（通过将其编程为 0）将该位复位，即使其已经为 0。

位 3 **COUNTRST**：计数器复位 (Counter reset)

此位通过软件置 1，通过硬件清零。当设置为 1 时，该位将触发 LPTIM_CNT 计数器寄存器的同步复位。由于该复位的同步性质，它仅在 3 个 LPTimer 内核时钟周期（LPTimer 内核时钟可能不同于 APB 时钟）的同步延迟之后发生。

注意： COUNTRST 在已由硬件清零之前，不得由软件置“1”。因此，软件应在尝试将 COUNTRST 位置“1”之前检查其是否已清零。

位 2 **CNTSTRT**：定时器以连续模式启动 (Timer start in Continuous mode)

此位通过软件置 1，通过硬件清零。

若通过软件启动（TRIGEN[1:0] =“00”），将此位置 1 会使 LPTIM 以连续模式启动。

如果禁止软件启动（TRIGEN[1:0] 不等于“00”），将此位置 1 会使定时器在检测到外部触发信号后立即以连续模式启动。

如果在进行单脉冲模式计数时将此位置 1，则在 LPTIM_ARR 寄存器和 LPTIM_CNT 寄存器下一次匹配时定时器不会停止，LPTIM 计数器将继续以连续模式计数。

只有在使能 LPTIM 时才能将此位置 1。此位由硬件自动复位。

位 1 **SNGSTRT**：LPTIM 以单脉冲模式启动 (LPTIM start in Single mode)

此位通过软件置 1，通过硬件清零。

若通过软件启动（TRIGEN[1:0] =“00”），将此位置 1 会使 LPTIM 以单脉冲模式启动。

如果禁止软件启动（TRIGEN[1:0] 不等于“00”），将此位置 1 会使 LPTIM 在检测到外部触发信号后立即以单脉冲模式启动。

如果在 LPTIM 处于连续计数模式时将此位置 1，LPTIM 将在 LPTIM_ARR 寄存器和 LPTIM_CNT 寄存器下一次匹配时停止。

只有在使能 LPTIM 时才能将此位置 1。此位由硬件自动复位。

位 0 **ENABLE**：LPTIM 使能 (LPTIM enable)

ENABLE 位由软件置 1 和清零。

0：禁止 LPTIM

1：使能 LPTIM

27.7.6 LPTIM 比较寄存器 (LPTIM_CMP)

LPTIM compare register

偏移地址：0x014

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 **CMP[15:0]**：比较值 (Compare value)

CMP 为 LPTIM 所使用的比较值。

注意： 必须在 LPTIM 已使能时（ENABLE 位置 1）才能修改 LPTIM_CMP 寄存器。

27.7.7 LPTIM 自动重载寄存器 (LPTIM_ARR)

LPTIM autoreload register

偏移地址：0x018

复位值：0x0000 0001

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

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

位 31:16 保留，必须保持复位值。

位 15:0 **ARR[15:0]**：自动重载值 (Auto reload value)

ARR 为 LPTIM 的自动重载值。

此值必须严格大于 CMP[15:0] 的值。

注意： 必须在 LPTIM 已使能时（ENABLE 位置 1）才能修改 LPTIM_ARR 寄存器。

27.7.8 LPTIM 计数器寄存器 (LPTIM_CNT)

LPTIM counter register

偏移地址：0x01C

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

当 LPTIM 通过异步时钟运行时，读取 LPTIM_CNT 寄存器会返回不可靠的值。因此在这种情况下，必须连续执行读访问两次，并验证两次返回的值是否相同。

应注意的是，为了实现可靠的 LPTIM_CNT 寄存器读访问，必须执行两次连续的读访问并进行比较。当两次连续读访问的值相等时，可认为读访问可靠。

27.7.9 LPTIM1 选项寄存器 (LPTIM1_OR)

LPTIM1 option register

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OR_1	OR_0
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **OR_1**：选项寄存器位 1 (Option register bit 1)
0： LPTIM1 输入 2 连接到 I/O
1： LPTIM1 输入 2 连接到 COMP2_OUT

位 0 **OR_0**：选项寄存器位 0 (Option register bit 0)
0： LPTIM1 输入 1 连接到 I/O
1： LPTIM1 输入 1 连接到 COMP1_OUT

27.7.10 LPTIM2 选项寄存器 (LPTIM2_OR)

LPTIM2 option register

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OR_1	OR_0
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **OR_1**：选项寄存器位 1 (Option register bit 1)
0： LPTIM2 输入 1 连接到 I/O
1： LPTIM2 输入 1 连接到 COMP2_OUT

位 0 **OR_0**：选项寄存器位 0 (Option register bit 0)
0： LPTIM2 输入 1 连接到 I/O
1： LPTIM2 输入 1 连接到 COMP1_OUT

27.7.11 LPTIM 寄存器映射

下表对 LPTIM 寄存器进行了汇总。

表 171. LPTIM 寄存器映射和复位值

偏移	寄存器名称	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
0x000	LPTIM_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWN(1)	UP(1)	ARROK	CMPOK	EXTTRIG	ARRM	CMPM
	Reset value																										0	0	0	0	0	0	0
0x004	LPTIM_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNCF(1)	UPCF(1)	ARROKCF	CMPOKCF	EXTTRIGCF	ARRMCF	CMPMCF
	Reset value																										0	0	0	0	0	0	0
0x008	LPTIM_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNIE(1)	UPIE(1)	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE
	Reset value																										0	0	0	0	0	0	0
0x00C	LPTIM_CFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENC(1)	COUNTMODE	PRELOAD	WAVEPOL	WAVE	TIMOUT	TRIGEN		Res.	TRIGSEL[2:0]			Res.	PRESC			Res.	TRGFLT		Res.	CKFLT		CKPOL		CKSEL
	Reset value								0	0	0	0	0	0	0	0		0	0	0		0	0	0		0	0		0	0	0	0	0
0x010	LPTIM_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSTARE	COUNTRST	CNTSTRT	SNGSTRT	ENABLE
	Reset value																												0	0	0	0	0
0x014	LPTIM_CMP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMP[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	LPTIM_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x01C	LPTIM_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	LPTIM_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OR_1	OR_0
	Reset value																															0	0

1. 如果 LPTIM 不支持编码器模式功能，则保留该位。请参见*第 27.3 节：LPTIM 实现*。

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

28 红外接口 (IRTIM)

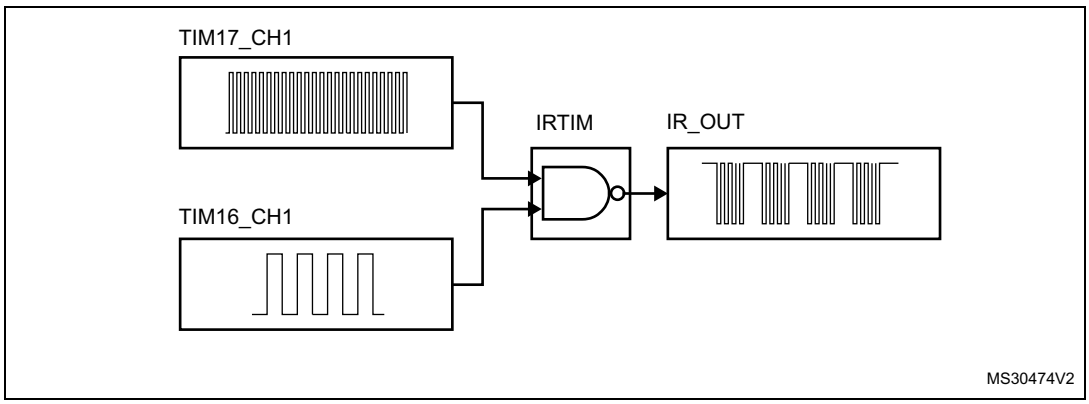
器件具有用于遥控的红外接口 (IRTIM)。此接口与红外 LED 一起用于实现遥控功能。

此接口内部连接到 TIM16 和 TIM17，如*图 287* 所示。

要生成红外遥控信号，必须使能 IR 接口并正确配置 TIM16 通道 1 (TIM16_OC1) 和 TIM17 通道 1 (TIM17_OC1)，以生成正确波形。

通过基本输入捕获模式可以轻松实现红外接收器。

图 287. IRTIM 与 TIM16 和 TIM17 的内部硬件连接



所有标准 IR 脉冲调制模式都可通过编程两个定时器输出比较通道获得。

TIM17 用于生成高频载波信号，而 TIM16 生成调制包络。

在 IR_OUT 引脚上输出红外功能。通过 GPIOx_AFRx 寄存器使能相关复用功能位来激活此功能。

高灌电流 LED 驱动能力（仅在 PB9 引脚上可用）可以通过 SYSCFG_CFGR1 寄存器中的 I2C_PB9_FMP 位激活，并用于吸收直接控制红外 LED 所需的高电流。

29 实时时钟 (RTC)

29.1 简介

实时时钟 (RTC) 提供用于管理所有低功耗模式的自动唤醒单元。

实时时钟 (RTC) 是一个独立的 BCD 定时器/计数器。RTC 提供具有可编程闹钟中断功能的日历时钟/日历。

RTC 还包含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 BCD 格式的秒、分钟、小时（12 或 24 小时制）、星期几、日期、月份和年份。此外，还可提供二进制格式的亚秒值。

系统可以自动将月份的天数补偿为 28、29（闰年）、30 和 31 天。并且还可以进行夏令时补偿。

其他 32 位寄存器还包含可编程的闹钟亚秒、秒、分钟、小时、星期几和日期。

此外，还可以使用数字校准功能对晶振精度的偏差进行补偿。

Backup 域复位后，所有 RTC 寄存器都会受到保护，以防止可能的非正常写访问。

无论器件状态如何（运行模式、低功耗模式或处于复位状态），只要电源电压保持在工作范围内，RTC 便不会停止工作。

29.2 RTC 主要特性

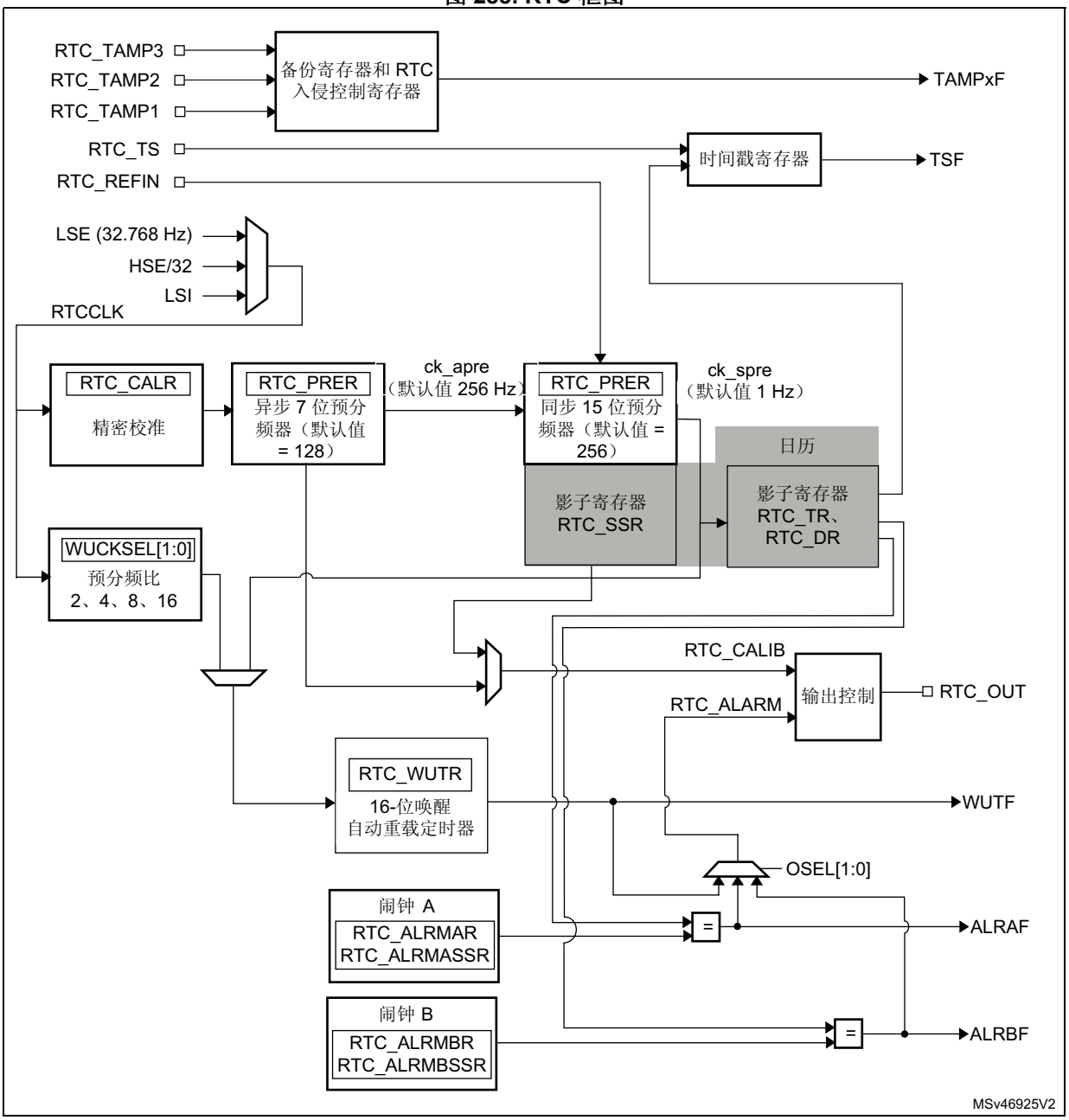
RTC 单元的主要特性如下（参见*图 288：RTC 框图*）：

- 包含亚秒、秒、分钟、小时（12/24 小时制）、星期几、日期、月份和年份的日历。
- 软件可编程的夏令时补偿。
- 具有中断功能的可编程闹钟。可通过任意日历字段的组合触发闹钟。
- 自动唤醒单元，可周期性地生成标志以触发自动唤醒中断。
- 参考时钟检测：可使用更加精确的第二时钟源（50 Hz 或 60 Hz）来提高日历的精确度。
- 利用亚秒级移位特性与外部时钟实现精确同步。
- 数字校准电路（周期性计数器调整）：精度为 0.95 ppm，在数秒钟的校准窗口中获得。
- 用于事件保存的时间戳功能。
- 带可配置过滤器和内部上拉的入侵检测事件。
- 可屏蔽中断/事件：
  - 闹钟 A
  - 闹钟 B
  - 唤醒中断
  - 时间戳
  - 入侵检测
- 20 个备份寄存器。

29.3 RTC 功能说明

29.3.1 RTC 框图

图 288. RTC 框图



RTC 包含：

- 入侵检测擦除备份寄存器
- 来自 I/O 的一个时间戳事件
- 篡改事件检测可生成一个时间戳事件
- 切换到 V_{BAT} 时会生成时间戳

29.3.2 时钟和预分频器

RTC 时钟源 (RTCCLK) 通过时钟控制器从 LSE 时钟、LSI 振荡器时钟以及 HSE 时钟三者中选择。有关 RTC 时钟源配置的更多信息，请参见*第 8 节：复位和时钟控制 (RCC)*。

可编程的预分频器阶段可生成 1 Hz 的时钟，用于更新日历。为最大程度地降低功耗，预分频器分为 2 个可编程的预分频器（参见*图 288：RTC 框图*）：

- 一个通过 RTC_PRER 寄存器的 PREDIV_A 位配置的 7 位异步预分频器。
- 一个通过 RTC_PRER 寄存器的 PREDIV_S 位配置的 15 位同步预分频器。

注： *使用两个预分频器时，推荐将异步预分频器配置为较高的值，以最大程度降低功耗。*

要使用频率为 32.768 kHz 的 LSE 获得频率为 1 Hz 的内部时钟 (ck_spre)，需要将异步预分频系数设置为 128，并将同步预分频系数设置为 256。

分频系数的最小值为 1，最大值为 2^{22}。

这对应于约为 4 MHz 的最大输入频率。

f_{ck_apre} 可根据以下公式得出：

$$f_{CK\_APRE} = \frac{f_{RTCCLK}}{PREDIV\_A + 1}$$

ck_apre 时钟用于为二进制 RTC_SSR 亚秒递减计数器提供时钟。当该计数器计数到 0 时，会使用 PREDIV_S 的内容重载 RTC_SSR。

f_{ck_spre} 可根据以下公式得出：

$$f_{CK\_SPRE} = \frac{f_{RTCCLK}}{(PREDIV\_S + 1) \times (PREDIV\_A + 1)}$$

ck_spre 时钟既可以用于更新日历，也可以用作 16 位唤醒自动重载定时器的时基。为获得较短的超时周期，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（有关详细信息，请参见*第 29.3.5 节：周期性自动唤醒*）。

29.3.3 实时时钟和日历

RTC 日历时间和日期寄存器可通过与 PCLK（APB 时钟）同步的影子寄存器来访问。这些时间和日期寄存器也可以直接访问，这样可避免等待同步的持续时间。

- RTC_SSR 对应于亚秒
- RTC_TR 对应于时间
- RTC_DR 对应于日期

每个 RTCCLK 周期，都将当前日历值复制到影子寄存器，并将 RTC_ISR 寄存器的 RSF 位置 1（请参见*第 29.6.4 节：RTC 初始化和状态寄存器 (RTC_ISR)*）。在停机和待机模式下不会执行复制操作。退出这两种模式时，影子寄存器会在最长 1 个 RTCCLK 周期后进行更新。

当应用读取日历寄存器时，它会访问影子寄存器的内容。也可以通过将 RTC_CR 寄存器的 BYPSHAD 控制位置 1 来直接访问日历寄存器。默认情况下，该位被清零，用户访问影子寄存器。

在 BYPSHAD=0 模式下读取 RTC_SSR、RTC_TR 或 RTC_DR 寄存器时，APB 时钟频率 (f_{APB}) 必须至少为 RTC 时钟频率 (f_{RTCCLK}) 的 7 倍。

影子寄存器通过系统复位来复位。

29.3.4 可编程闹钟

RTC 单元提供两个可编程闹钟，即闹钟 A 和闹钟 B。以下说明针对闹钟 A，但同样适用于闹钟 B。

可通过 RTC_CR 寄存器中的 ALRAE 位来使能可编程闹钟功能。如果日历亚秒、秒、分钟、小时、日期或日与闹钟寄存器 RTC_ALRMASSR 和 RTC_ALRMAR 中编程的值相匹配，则 ALRAF 标志会被置为 1。可通过 RTC_ALRMAR 寄存器的 MSKx 位以及 RTC_ALRMASSR 寄存器的 MASKSSx 位单独选择各日历字段。可通过 RTC_CR 寄存器中的 ALRAIE 位来使能闹钟中断。

注意： 如果选择秒字段（RTC_ALRMAR 中的 MSK1 位复位），则 RTC_PRER 寄存器中设置的同步预分频器分频系数必须至少为 3，才能确保闹钟正确地运行。

闹钟 A 和闹钟 B（如果已通过 RTC_CR 寄存器中的位 OSEL[1:0] 使能）可连接到 RTC_ALARM 输出。可通过 RTC_CR 寄存器的 POL 位配置 RTC_ALARM 输出极性。

29.3.5 周期性自动唤醒

周期性唤醒标志由 16 位可编程自动重载递减计数器生成。唤醒定时器范围可扩展至 17 位。

可通过 RTC_CR 寄存器中的 WUTE 位来使能此唤醒功能。

唤醒定时器的时钟输入可以是：

- 2、4、8 或 16 分频的 RTC 时钟 (RTCCLK)。

  当 RTCCLK 为 LSE (32.768 kHz) 时，可配置的唤醒中断周期介于 122 µs 和 32 s 之间，且分辨率低至 61 µs。
- ck_spre（通常为 1 Hz 内部时钟）

  当 ck_spre 频率为 1 Hz 时，可得到的唤醒时间为 1s 到 36h 左右，分辨率为 1 秒。这一较大的可编程时间范围分为两部分：
  - WUCKSEL [2:1] = 10 时为 1s 到 18h
  - WUCKSEL [2:1] = 11 时约为 18h 到 36h。在后一种情况下，会将 2^{16} 添加到 16 位计数器当前值。完成初始化序列后（请参见*第 886 页的编程唤醒定时器*），定时器开始递减计数。在低功耗模式下使能唤醒功能时，递减计数保持有效。此外，当计数器计数到 0 时，RTC_ISR 寄存器的 WUTF 标志会置 1，并且唤醒计数器会使用其重载值（RTC_WUTR 寄存器值）自动重载。

之后必须用软件清零 WUTF 标志。

通过将 RTC_CR 寄存器中的 WUTIE 位置 1 来使能周期性唤醒中断时，它会使器件退出低功耗模式。

如果已通过 RTC_CR 寄存器的位 OSEL[1:0] 使能周期性唤醒标志，则该标志可连接到 RTC_ALARM 输出。可通过 RTC_CR 寄存器的 POL 位配置 RTC_ALARM 输出极性。

系统复位以及低功耗模式（睡眠、停机和待机）对唤醒定时器没有任何影响。

29.3.6 RTC 初始化和配置

RTC 寄存器访问

RTC 寄存器为 32 位寄存器。除了当 BYPSHAD=0 时对日历影子寄存器执行的读访问之外，APB 接口会在访问 RTC 寄存器时引入 2 个等待周期。

RTC 寄存器写保护

系统复位后，可通过对 PWR_CR 寄存器中的 DBP 位清零来保护 RTC 寄存器以防止非正常的写访问（请参见电源控制部分）。必须将 DBP 位置 1 才能使能 RTC 寄存器的写访问。

Backup 域复位后，所有 RTC 寄存器均受到写保护。通过向写保护寄存器 (RTC_WPR) 写入一个密钥来使能对 RTC 寄存器的写操作。

要解锁所有 RTC 寄存器（RTC_TAMPCR、RTC_BKPxR、RTC_OR 和 RTC_ISR[13:8] 除外）的写保护，需要执行以下步骤：

1. 将“0xCA”写入 RTC_WPR 寄存器。
2. 将“0x53”写入 RTC_WPR 寄存器。

写入一个错误的关键字会再次激活写保护。

保护机制不受系统复位影响。

日历初始化和配置

要编程包括时间格式和预分频器配置在内的初始时间和日期日历值，需按照以下顺序操作：

1. 将 RTC_ISR 寄存器中的 INIT 位置 1 以进入初始化模式。在此模式下，日历计数器将停止工作并且其值可更新。
2. 轮询 RTC_ISR 寄存器中的 INITF 位。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）。
3. 要为日历计数器生成 1 Hz 时钟，应编程 RTC_PRER 寄存器中的两个预分频系数。
4. 在影子寄存器（RTC_TR 和 RTC_DR）中加载初始时间和日期值，然后通过 RTC_CR 寄存器中的 FMT 位配置时间格式（12 或 24 小时制）。
5. 通过清零 INIT 位退出初始化模式。随后，自动加载实际日历计数器值，在 4 个 RTCCLK 时钟周期后重新开始计数。

当初始化序列完成之后，日历开始计数。

注： *系统复位后，应用可读取 RTC_ISR 寄存器中的 INITS 标志，以检查日历是否已初始化。如果该标志为 0，表明自系统复位以来，日历还尚未初始化过，其年份字段一直还保持着 Backup 域复位默认值 (0x00)。*

要在初始化之后读取日历，必须首先用软件检查 RTC_ISR 寄存器的 RSF 标志是否置 1。

夏令时

可通过 RTC_CR 寄存器的 SUB1H、ADD1H 和 BKP 位管理夏令时。

利用 SUB1H 或 ADD1H，软件只需单次操作便可在日历中减去或增加一个小时，无需执行整个初始化步骤。

此外，软件还可以使用 BKP 位来记录是否曾经执行过此操作。

编程闹钟

要对可编程的闹钟进行编程或更新，必须执行类似的步骤。以下步骤针对闹钟 A，但同样适用于闹钟 B。

1. 将 RTC_CR 中的 ALRAE 位清零以禁止闹钟 A。
2. 编程闹钟 A 寄存器 (RTC_ALRMASSR/RTC_ALRMAR)。
3. 将 RTC_CR 寄存器中的 ALRAE 位置 1 以再次使能闹钟 A。

注： *由于时钟同步缘故，RTC_CR 寄存器的每次更改需要在大约 2 个 RTCCLK 时钟周期后执行。*

编程唤醒定时器

要配置或更改唤醒定时器的自动重载值（RTC_WUTR 中的 WUT[15:0]），需要按照以下顺序操作：

1. 清零 RTC_CR 中的 WUTE 以禁止唤醒定时器。
2. 轮询 RTC_ISR 中的 WUTWF，直到该位置 1，以确保可以访问唤醒自动重载定时器和 WUCKSEL[2:0] 位。大约需要 2 个 RTCCLK 时钟周期（由于时钟同步）。
3. 编程唤醒自动重载值 WUT[15:0]，并选择唤醒时钟（RTC_CR 中的 WUCKSEL[2:0] 位）。将 RTC_CR 寄存器中的 WUTE 位置 1 以再次使能定时器。唤醒定时器重新开始递减计数。由于时钟同步的缘故，在 WUTE 清零后，WUTWF 位也会清零，但需要花费多达 2 个 RTCCLK 时钟周期。

29.3.7 读取日历

当 RTC_CR 寄存器中的 BYPSHAD 控制位清零时

要正确读取 RTC 日历寄存器（RTC_SSR、RTC_TR 和 RTC_DR），APB 时钟频率 (f_{PCLK}) 必须等于或大于 RTC 时钟频率 (f_{RTCCLK}) 的七倍。这可以确保同步机制的安全性。

如果 APB 时钟频率低于 RTC 时钟频率的七倍，则软件必须分两次读取日历时间寄存器和日期寄存器。这样，当两次读取的 RTC_TR 结果相同时，才能确保数据正确。否则必须执行第三次读访问。任何情况下，APB 的时钟频率都不能低于 RTC 的时钟频率。

每次将日历寄存器中的值复制到 RTC_SSR、RTC_TR 和 RTC_DR 影子寄存器时，RTC_ISR 寄存器中的 RSF 位都会置 1。每个 RTCCLK 周期执行一次复制。为确保这 3 个值来自同一时刻点，读取 RTC_SSR 或 RTC_TR 时会锁定高阶日历影子寄存器中的值，直到读取 RTC_DR。为避免软件对日历执行读访问的时间间隔小于 1 个 RTCCLK 周期：第一次读取日历之后必须通过软件将 RSF 清零，并且软件必须等待到 RSF 置 1 之后才可再次读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

从低功耗模式（停机模式或待机模式）唤醒之后，必须通过软件将 RSF 清零。之后，软件必须等待至 RSF 再次置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

RSF 位必须在唤醒之后而不是进入低功耗模式之前进行清零。

系统复位之后，软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。实际上，系统复位会将影子寄存器复位为其默认值。

初始化之后（请参见*第 885 页的日历初始化和配置*），软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

同步之后（请参见*第 29.3.9 节：RTC 同步*），软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

当 RTC_CR 寄存器中的 BYPSHAD 控制位置 1 时（旁路影子寄存器）

读取日历寄存器时会直接从日历计数器获取值，这样便无需等待至 RSF 位置 1。这对于从低功耗模式（停机模式或待机模式）退出后的情况特别有用，因为影子寄存器在这些模式下不更新。

当 BYPSHAD 位置 1 时，如果在对寄存器的两次读访问之间出现 RTCCLK 沿，则不同寄存器的结果彼此可能不一致。此外，如果在读操作期间出现 RTCCLK 沿，则可能导致其中一个寄存器的值不正确。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： *当 BYPSHAD=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。*

29.3.8 复位 RTC

日历影子寄存器（RTC_SSR、RTC_TR 和 RTC_DR）以及 RTC 状态寄存器 (RTC_ISR) 的某些位通过所有可用的系统复位源复位为各自的默认值。

相反，以下寄存器则通过 Backup 域复位来复位为各自的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_CR)、预分频器寄存器 (RTC_PRER)、RTC 校准寄存器 (RTC_CALR)、RTC 移位寄存器 (RTC_SHIFTR)、RTC 时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）、RTC 入侵配置寄存器 (RTC_TAMPCR)、RTC 备份寄存器 (RTC_BKPxR)、唤醒定时器寄存器 (RTC_WUTR)、闹钟 A 和闹钟 B 寄存器（RTC_ALRMASSR/RTC_ALRMAR 和 RTC_ALRMBSSR/RTC_ALRMBR），以及选项寄存器 (RTC_OR)。

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。

29.3.9 RTC 同步

RTC 可与高精度的远程时钟同步。在读取亚秒字段后（RTC_SSR 或 RTC_TSSSR），即可计算远程时钟的时间与 RTC 之间的精准偏差。之后，可使用 RTC_SHIFTR 对 RTC 的时钟进行零点几秒的“平移”，经过调整后可消除此偏差。

RTC_SSR 包含同步预分频器计数器的值。这样，便可计算分辨率低至 1/(PREDIV_S + 1) 秒的 RTC 的准确时间。因此，可通过增大同步预分频器的值 (PREDIV_S[14:0]) 来提高分辨率。将 PREDIV_S 设置为 0x7FFF 时，可得到允许的最大分辨率（30.52 µs，时钟频率为 32768 Hz）。

但是，提高 PREDIV_S 意味着必须降低 PREDIV_A 才能将同步预分频器的输出维持在 1 Hz。这样，异步预分频器的输出频率会增大，RTC 的动态功耗也会相应增加。

可以使用 RTC 平移控制寄存器 (RTC_SHIFTR) 对 RTC 进行微调。可以用大小为 1/(PREDIV_S + 1) 秒的分辨率对 RTC_SHIFTR 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 SUBFS[14:0] 值加到同步预分频器计数器 SS[15:0] 中：这将使时钟产生延迟。如果同时将 ADD1S 位置 1，则会增加一秒，与此同时减去的时间为零点几秒，因此将使时钟提前。

注意： 初始化平移操作前，用户必须检查确认 SS[15] = 0，以确保不会发生上溢。

对 RTC_SHIFTR 寄存器执行写操作以启动平移操作时，硬件会将 SHPF 标志置 1 以指示平移操作挂起。完成平移操作时，硬件会将该位清零。

注意： 该同步功能与参考时钟检测功能不兼容：当 REFCKON=1 时，固件不能对 RTC_SHIFTR 执行写操作。

29.3.10 RTC 参考时钟检测

RTC 日历更新可与参考时钟 RTC_REFIN（通常为市电频率，50 Hz 或 60 Hz）同步。RTC_REFIN 参考时钟的精度应高于 32.768 kHz LSE 时钟。使能 RTC_REFIN 检测时（将 RTC_CR 的 REFCKON 位置 1），日历仍由 LSE 提供时钟，而 RTC_REFIN 用于补偿不准确的日历更新频率 (1 Hz)。

每个 1 Hz 时钟边沿都与最近的 RTC_REFIN 时钟边沿进行比较（如果在给定的时间窗口内发现一个边沿）。在大多数情况下，两个时钟边沿恰好对齐。当 1 Hz 时钟由于 LSE 时钟不精确而发生偏离时，RTC 会稍微偏移 1 Hz 时钟，以便后续的 1 Hz 时钟边沿能够对齐。利用这种机制，可使日历像参考时钟一样精确。

RTC 使用 32.768 kHz 石英产生的 256 Hz 时钟 (ck_apre) 检测是否存在参考时钟源。大约在日历每次更新时（每 1 秒钟），便会在时间窗口期间执行一次检测。检测到第一个参考时钟边沿时，该窗口等于 7 个 ck_apre 周期。随后的日历更新使用长度为 3 个 ck_apre 周期的较小窗口。

每次在窗口中检测到参考时钟时，都会行强制输出 ck_spre 时钟的同步预分频器进行重载。当参考时钟与 1 Hz 时钟对齐时，此操作不起作用，因为预分频器会在同一时刻重载。当时钟不对齐时，重载操作会微调后续的 1 Hz 时钟边沿，使其与参考时钟对齐。

如果参考时钟停止（在 3 个 ck_apre 窗口内未出现参考时钟边沿），日历将仅根据 LSE 时钟进行连续更新。RTC 随后使用 ck_spre 边沿上居中的大检测窗口（7 个 ck_apre 周期）等待参考时钟。

使能 RTC_REFIN 检测后，必须将 PREDIV_A 和 PREDIV_S 设置为各自的默认值：

- PREDIV_A = 0x007F
- PREVID_S = 0x00FF

注： *RTC_REFIN 时钟检测在待机模式下不可用。*

29.3.11 RTC 精密数字校准

RTC 频率可采用约 0.954 ppm 的分辨率进行数字校准，校准范围为 -487.1 ppm 到 +488.5 ppm。使用一系列微调 （增加和/或减少单独的 RTCCLK 脉冲）进行频率校正。这些微调的分布非常均匀，因此 RTC 的校准效果相当好，即使在短时间内持续观察也是如此。

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。

精密数字校准寄存器 (RTC_CALR) 可指定 32 秒周期内要减少的 RTCCLK 时钟周期数：

- 将位 CALM[0] 置 1 时，32 秒周期内将只减少 1 个脉冲。
- 将 CALM[1] 置 1 时，将减少 2 个周期。
- 将 CALM[2] 置 1 时，将减少 4 个周期
- 依此类推，将 CALM[8] 置 1 时，将减少 256 个时钟。

注： *CALM[8:0] (RTC_CALR) 可指定 32 秒周期内要减少的 RTCCLK 脉冲数。将位 CALM[0] 置 1 时，32 秒周期内将只减少 1 个脉冲（当 cal_cnt[19:0] = 0x80000 时）；将 CALM[1] 置 1 时，将减少 2 个周期（当 cal_cnt = 0x40000 和 0xC0000 时）；将 CALM[2] 置 1 时，将减少 4 个周期（当 cal_cnt = 0x20000/0x60000/0xA0000/0xE0000 时）；依此类推，将 CALM[8] 置 1 时，将减少 256 个时钟（当 cal_cnt = 0xXX800 时）。*

使用适当分辨率时，CALM 可使 RTC 频率减少最多 487.1 ppm，而 CALP 可用于使频率增加 488.5 ppm。将 CALP 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 CALM 和 CALP 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})：

$$F_{CAL} = F_{RTCCLK} \times [1 + (CALP \times 512 - CALM) / (2^{20} + CALM - CALP \times 512)]$$

PREDIV_A<3 条件下的校准

当异步预分频器值（RTC_PRER 寄存器中的 PREDIV_A 位）小于 3 时，不能将 CALP 位置 1。如果 CALP 已置 1 并且 PREDIV_A 位的值小于 3，则会忽略 CALP，即假定 CALP 等于 0 而执行校准。

要在 PREDIV_A 小于 3 的条件下执行校准，应降低同步预分频器值 (PREDIV_S) 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 CALM 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 PREDIV_A 等于 1 时（分频系数为 2），应将 PREDIV_S 设置为 16379 而不是 16383（少 4）。唯一相关的其他情况是，当 PREDIV_A 等于 0 时，应将 PREDIV_S 设置为 32759 而不是 32767（少 8）。

如果以这种方式减少 PREDIV_S，则采用以下公式计算校准

输入时钟的有效频率：

$$F_{CAL} = F_{RTCCLK} \times [1 + (256 - CALM) / (2^{20} + CALM - 256)]$$

在这种情况下，如果 RTCCLK 恰好为 32768.00 Hz，则当 CALM[7:0] 等于 0x100 时（CALM 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，计算正确的 CALM 和 CALP 值以确保 RTC 精度。此外，还为应用提供了一个可选的 1 Hz 输出，用来测量和验证 RTC 精度。

如果在有限的间隔内测量 RTC 的精确频率，则会导致测量期间产生最多 2 个 RTCCLK 时钟周期的测量误差，具体取决于数字校准周期与测量周期的对齐方式。

但是，如果测量周期与校准周期的长度相同，则可以消除此测量误差。在这种情况下，观测到的唯一误差是由数字校准的分辨率导致的误差。

- 默认情况下，校准周期为 32 秒。

在此模式下，测量整个 32 秒内 1 Hz 输出的精度，可确保测量误差在 0.477 ppm 内（32 秒内为 0.5 个 RTCCLK 周期，受校准分辨率限制）。

- 可将 RTC_CALR 寄存器的 CALW16 位置 1，以强制 16 秒的校准周期。

此时，可在 16 秒内测量 RTC 精度，产生的最大误差为 0.954 ppm（16 秒内为 0.5 个 RTCCLK 周期）。但是，由于校准分辨率降低，长期的 RTC 精度也会降到 0.954 ppm：将 CALW16 置 1 时，CALM[0] 位将始终保持为 0。

- 可将 RTC_CALR 寄存器的 CALW8 位置 1，以强制 8 秒的校准周期。

此时，可在 8 秒内测量 RTC 精度，产生的最大误差为 1.907 ppm（8 秒内为 0.5 个 RTCCLK 周期）。长期的 RTC 精度也会降到 1.907 ppm：将 CALW8 置 1 时，CALM[1:0] 位将始终保持为 00。

动态重校准

当 RTC_ISR/INITF=0 时，可动态更新校准寄存器 (RTC_CALR)，具体步骤如下：

1. 轮询 RTC_ISR/RECALPF（重新校准挂起标志）。
2. 如果该标志为 0，则可以根据需要向 RTC_CALR 写入新值。随后 RECALPF 位会被自动置为 1。
3. 新校准设置将在对 RTC_CALR 执行写操作之后的三个 ck_apre 周期内生效。

29.3.12 时间戳功能

将 RTC_CR 寄存器的 TSE 位或 ITSE 位置 1 可使能时间戳。

将 TSE 置 1 时：

当在 RTC_TS 引脚上检测到时间戳事件时，日历会保存到时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）中。

将 ITSE 置 1 时：

当检测到内部时间戳事件时，日历会保存到时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）中。切换至 VBAT 电源可生成内部时间戳事件。

由于内部事件或外部事件而发生时间戳事件时，RTC_ISR 寄存器中的时间戳标志位 (TSF) 将置 1。如果是内部事件，RTC_ISR 寄存器中的 ITSF 标志也将置 1。

通过将 RTC_CR 寄存器中的 TSIE 位置 1，可在发生入侵检测事件时生成中断。

如果在时间戳标志 (TSF) 已置 1 的条件下检测到新的时间戳事件，则时间戳上溢标志 (TSOVF) 将置 1，而时间戳寄存器（RTC_TSTR 和 RTC_TSDR）将保持上一事件的结果。

注： *在发生由同步过程引发的时间戳事件后，TSF 在 2 个 ck_apre 周期置为 1。*

TSOVF 的产生中不存在延迟。也就是说如果两个时间戳事件接连发生，TSOVF 可能为“1”而 TSF 为“0”。因此，建议只在检测到 TSF 为“1”后再轮询 TSOVF。

注意： 如果在 TSF 位清零后紧接着发生时间戳事件，则 TSF 和 TSOVF 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 TSF 位读取为“1”，否则应用程序不得将“0”写入 TSF 位。

此外，入侵事件可能导致时间戳被记录。有关 TAMPTS 控制位的说明，请参见*第 29.6.16 节：RTC 入侵配置寄存器 (RTC_TAMPCR)*。

29.3.13 入侵检测

RTC_TAMPx 入侵输入事件既可配置为边沿检测，也可配置为带过滤的电平检测。

入侵检测可配置用于以下目的：

- 擦除 RTC 备份寄存器（默认配置）
- 生成中断，能够从停止模式和待机模式唤醒
- 为低功耗定时器生成硬件触发

RTC 备份寄存器

备份寄存器 (RTC_BKPxR) 不会通过系统复位来复位，也不会在器件从待机模式唤醒时复位。

备份寄存器在发生入侵检测事件（请参见*第 29.6.20 节：RTC 备份寄存器 (RTC_BKPxR)* 和*第 891 页的入侵检测初始化*）时复位，TAMPxNOERASE 位置 1 时或者 RTC_TAMPCR 寄存器中的 TAMPxMF 置 1 时除外。

入侵检测初始化

可通过将 RTC_TAMPCR 寄存器中相应的 TAMPxE 位置 1 来使能各输入。

每个 RTC_TAMPx 入侵检测输入分别与 RTC_ISR 寄存器中的标志 TAMPxF 相关联。

将 TAMPxMF 清零时：

TAMPxF 标志将在引脚上出现入侵事件后使能，并存在下述延迟：

- 当 TAMPFLT 不为 0x0 时（带过滤的电平检测），延迟为 3 个 ck_apre 周期
- 当 TAMPTS=1 时（入侵事件的时间戳），延迟为 3 个 ck_apre 周期
- 当 TAMPFLT=0x0（边沿检测）且 TAMPTS=0 时，无延迟

在此期间，只要 TAMPxF 置 1，就无法检测到同一引脚上出现的新入侵。

将 TAMPxMF 置 1 时：

在上述延迟期间以及在 2.5 个 ck_rtc 附加周期内，无法检测到同一引脚上出现的新入侵。

通过将 RTC_TAMPCR 寄存器中的 TAMPIE 位置 1，可在发生入侵检测事件时（当 TAMPxF 置 1 时）生成中断。当一个或多个 TAMPxMF 置 1 时，不允许将 TAMPIE 置 1。

将 TAMPIE 清零时，可通过将 RTC_TAMPCR 寄存器中相应的 TAMPxIE 位置 1 来分别使能各个入侵引脚事件中断。当相应的 TAMPxMF 置 1 时，不允许将 TAMPxIE 置 1。

出现入侵事件时生成触发输出

低功耗定时器可将入侵事件检测用作触发输入。

将 RTC_TAMPCR 寄存器中的 TAMPxMF 位清零时，必须通过软件将 TAMPxF 标志清零以便在同一引脚上检测新入侵。

将 TAMPxMF 位置 1 时，TAMPxF 标志会被屏蔽，并在 RTC_ISR 寄存器中保持清零。此配置允许在停止模式下自动触发低功耗定时器，无需通过系统唤醒来执行 TAMPxF 清零。在这种情况下，备份寄存器不清零。

入侵事件的时间戳

当 TAMPTS 置“1”时，任何入侵事件都会导致时间戳事件发生。在这种情况下，如同发生正常时间戳事件一样，RTC_ISR 中的 TSF 位或 TSOVF 位会置 1。在 TSF 或 TSOVF 置 1 的同时，受影响的入侵标志寄存器 TAMPxF 也会随之置 1。

对入侵输入的边沿检测

如果 TAMPFLT 位设置为“00”，当相应的 TAMPxTRG 位上出现上升沿或下降沿时，RTC_TAMPx 引脚将生成入侵检测事件。选择边沿检测时，会禁止 RTC_TAMPx 输入上的内部上拉电阻。

注意： 使用边沿检测时，建议在使能入侵检测后（通过读取 GPIO 寄存器）以及向备份寄存器中写入敏感值前立即通过软件检查入侵引脚电平，以确保使能入侵事件检测后才出现有效边沿。
当 TAMPFLT = “00”且 TAMPxTRG = 0（上升沿检测）时，如果在使能入侵检测前入侵输入已处于高电平，则可通过硬件来检测入侵事件。

检测到入侵事件并清零后，应当在重新编程备份寄存器 (RTC_BKPxR) 之前禁止 RTC_TAMPx，然后再重新使能（TAMPxE 置 1）。这可防止应用在 RTC_TAMPx 输入值仍指示入侵检测时，对备份寄存器执行写操作。这相当于对 RTC_TAMPx 输入的电平检测。

注： *当 V_{DD} 电源关闭时，入侵检测仍有效。要避免意外复位备份寄存器，应将 RTC_TAMPx 映射到的引脚从外部连接到正确的电平。*

对 RTC_TAMPx 输入的带过滤电平检测

通过将 TAMPFLT 设置为非零值可执行带过滤的电平检测。在 TAMPxTRG 位指定的电平连续出现 2、4 或 8 个（取决于 TAMPFLT 值）采样时生成入侵检测事件。

除非通过将 TAMPPUDIS 置 1 禁止入侵输入，否则在入侵输入的状态被采样前，将通过 I/O 内部上拉电阻对 RTC_TAMPx 输入预充电。预充电的持续时间由 TAMPPRCH 位确定，允许增大 RTC_TAMPx 输入上的电容。

可使用 TAMPFREQ 确定用于电平检测的采样频率，以便使入侵检测延迟与上拉电阻功耗之间达到最佳平衡。

注： *有关上拉电阻的电气特性，请参见数据手册。*

29.3.14 校准时钟输出

将 RTC_CR 寄存器中的 COE 位置 1 时，会在 RTC_CALIB 器件输出上提供一个参考时钟。

如果 RTC_CR 寄存器中的 COSEL 位复位且 PREDIV_A = 0x7F，则 RTC_CALIB 频率为 $f_{RTCCLK/64}$。这相当于 RTCCLK 频率为 32.768 kHz 时，512 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 COSEL 置 1 且“PREDIV_S+1”为 256 的非零整数倍（比如：PREDIV_S[7:0] = 0xFF），则 RTC_CALIB 频率为 $f_{RTCCLK}/(256 * (PREDIV\_A+1))$。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（PREDIV_A = Ox7F、PREDIV_S = 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 (SHPF = 1)。

注： *选择 RTC_CALIB 或 RTC_ALARM 输出时，RTC_OUT 引脚会自动配置为输出。*

COSEL 位清零时，RTC_CALIB 输出为异步预分频器的第 6 级输出。

COSEL 位置 1 时，RTC_CALIB 输出为同步预分频器的第 8 级输出。

29.3.15 闹钟输出

RTC_CR 寄存器中的 OSEL[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_ISR 寄存器中相应标志的内容。

输出的极性由 RTC_CR 中的 POL 控制位确定，这样当 POL 置 1 时会输出选定标志位的相反值。

闹钟输出

使用 RTC_OR 寄存器中的控制位 RTC_ALARM_TYPE 可将 RTC_ALARM 引脚配置为输出开漏或输出上拉。

注： *使能 RTC_ALARM 输出之后，其优先级高于 RTC_CALIB（与 COE 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。

29.4 RTC 低功耗模式

29.5 RTC 中断

所有 RTC 中断均与 EXTI 控制器相连。请参见*第 14 节：扩展中断和事件控制器 (EXTI)*。

29.6 RTC 寄存器

有关寄存器说明中使用的缩写，请参见参考手册中的*第 58 页的第 1.2 节*。

外设寄存器可按字（32 位）进行访问。

29.6.1 RTC 时间寄存器 (RTC_TR)

RTC time register

RTC_TR 是日历时间影子寄存器。只能在初始化模式下对该寄存器执行写操作。请参见*第 885 页的日历初始化和配置*和*第 886 页的读取日历*。

此寄存器受写保护。*第 885 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x00

Backup 域复位值：0x0000 0000

系统复位：当 BYPSHAD = 0 时为 0x0000 0000。当 BYPSHAD = 1 时不受影响。

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

位 31-23 保留，必须保持复位值

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 保留，必须保持复位值。

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 保留，必须保持复位值。

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)

29.6.2 RTC 日期寄存器 (RTC_DR)

RTC date register

RTC_DR 是日历日期影子寄存器。只能在初始化模式下对该寄存器执行写操作。请参见*第 885 页的日历初始化和配置*和*第 886 页的读取日历*。

此寄存器受写保护。*第 885 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x04

Backup 域复位值：0x0000 2101

系统复位：当 BYPSHAD = 0 时为 0x0000 2101。当 BYPSHAD = 1 时不受影响。

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

位 31:24 保留，必须保持复位值

位 23:20 **YT[3:0]**：年份的十位（BCD 格式）(Year tens in BCD format)

位 19:16 **YU[3:0]**：年份的个位（BCD 格式）(Year units in BCD format)

位 15:13 **WDU[2:0]**：星期几的个位 (Week day units)

000：禁止
001：星期一
...
111：星期日

位 12 **MT**：月份的十位（BCD 格式）(Month tens in BCD format)

位 11:8 **MU**：月份的个位（BCD 格式）(Month units in BCD format)

位 7:6 保留，必须保持复位值。

位 5:4 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 3:0 **DU[3:0]**：日期的个位（BCD 格式）(Date units in BCD format)

29.6.3 RTC 控制寄存器 (RTC_CR)

RTC control register

偏移地址：0x08

Backup 域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSE	COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H
							rw	rw	rw	rw	rw	rw	rw	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	Res.	FMT	BYPSHAD	REFCKON	TSEDGE	WUCKSEL[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24 **ITSE**：内部事件时间戳使能 (timestamp on internal event enable)
0：禁止内部事件时间戳
1：使能内部事件时间戳

位 23 **COE**：校准输出使能 (Calibration output enable)
该位使能 RTC_CALIB 输出
0：禁止校准输出
1：使能校准输出

位 22:21 **OSEL[1:0]**：输出选择 (Output selection)
这些位用于选择要连接到 RTC_ALARM 输出的标志
00：禁止输出
01：使能闹钟 A 输出
10：使能闹钟 B 输出
11：使能唤醒输出

位 20 **POL**：输出极性 (Output polarity)
该位用于配置 RTC_ALARM 输出的极性
0：当 ALRAF/ALRBF/WUTF 置 1 时（取决于 OSEL[1:0]），该引脚为高电平
1：当 ALRAF/ALRBF/WUTF 置 1 时（取决于 OSEL[1:0]），该引脚为低电平

位 19 **COSEL**：校准输出选择 (Calibration output selection)
当 COE=1 时，该位可选择 RTC_CALIB 上输出的信号。
0：校准输出为 512 Hz（采用默认预分频器设置）
1：校准输出为 1 Hz（采用默认预分频器设置）
在 RTCCLK 为 32.768 kHz 且预分频器为其默认值（PREDIV_A=127 且 PREDIV_S=255）的条件下，这些频率有效。请参见*第 29.3.14 节：校准时钟输出*。

位 18 **BKP**：备份 (Backup)
用户可对此位执行写操作以记录是否已对夏令时进行更改。

位 17 **SUB1H**：减少 1 小时（冬季时间更改）(Subtract 1 hour (winter time change))
该位置 1 时，如果当前小时不是 0，则日历时间将减少 1 小时。此位始终读为 0。
当前小时为 0 时，将此位置 1 没有任何作用。
0：无影响。
1：将当前时间减少 1 小时。这可用于冬季时间更改（在初始化模式以外）。

位 16 **ADD1H**：增加 1 小时（夏季时间更改）(Add 1 hour (summer time change))
该位置 1 时，日历时间将增加 1 小时。此位始终读为 0。
0：无影响。
1：将当前时间增加 1 小时。这可用于夏季时间更改（在初始化模式以外）。

位 15 **TSIE**：时间戳中断使能 (Time-stamp interrupt enable)
0：禁止时间戳中断
1：使能时间戳中断

位 14 **WUTIE**：唤醒定时器中断使能 (Wakeup timer interrupt enable)
0：禁止唤醒定时器中断
1：使能唤醒定时器中断

位 13 **ALRBIE**：*闹钟 B 中断使能 (Alarm B interrupt enable)*
0：禁止闹钟 B 中断
1：使能闹钟 B 中断

位 12 **ALRAIE**：闹钟 A 中断使能 (Alarm A interrupt enable)
0：禁止闹钟 A 中断
1：使能闹钟 A 中断

位 11 **TSE**：时间戳使能 (timestamp enable)
0：禁止时间戳
1：使能时间戳

位 10 **WUTE**：唤醒定时器使能 (Wakeup timer enable)
0：禁止唤醒定时器
1：使能唤醒定时器
注：　唤醒定时器禁止时，需要等到 WUTWF=1 后才能重新使能。

位 9 **ALRBE**：*闹钟 B 使能 (Alarm B enable)*
0：禁止闹钟 B
1：使能闹钟 B

位 8 **ALRAE**：闹钟 A 使能 (Alarm A enable)
0：禁止闹钟 A
1：使能闹钟 A

位 7 保留，必须保持复位值。

位 6 **FMT**：小时格式 (Hour format)
0：24 小时 / 天格式
1：AM/PM 小时格式

位 5 **BYPSHAD**：旁路影子寄存器 (Bypass the shadow registers)

0：日历值（从 RTC_SSR、RTC_TR 和 RTC_DR 读取时）取自影子寄存器，该影子寄存器每两个 RTCCLK 周期更新一次。

1：日历值（从 RTC_SSR、RTC_TR 和 RTC_DR 读取时）直接取自日历计数器。

注： 如果 APB 时钟的频率低于 7 倍的 RTCCLK 频率，则必须将 BYPSHAD 置“1”。

位 4 **REFCKON**：RTC_REFIN 参考时钟检测使能（50 Hz 或 60 Hz）(RTC_REFIN reference clock detection enable (50 or 60 Hz))

0：禁止 RTC_REFIN 检测

1：使能 RTC_REFIN 检测

注： PREDIV_S 必须为 0x00FF。

位 3 **TSEDGE**：时间戳事件有效边沿 (Timestamp event active edge)

0：RTC_TS 输入上升沿生成时间戳事件

1：RTC_TS 输入下降沿生成时间戳事件

TSEDGE 发生更改时，必须复位 TSE 以避免将 TSF 意外置 1。

位 2:0 **WUCKSEL[2:0]**：唤醒时钟选择 (Wakeup clock selection)

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 时钟（通常为 1 Hz）

11x：选择 ck_spre 时钟（通常为 1 Hz）并将 WUT 计数器值增加 2^{16}（见下面的注释）

注： *只能在初始化模式下 (RTC_ISR/INITF = 1) 对该寄存器的位 7、6 和 4 执行写操作。*

WUT = 唤醒单元计数器值。当 WUCKSEL[2:1 = 11] 时，WUT = (0x0000 到 0xFFFF) + 0x10000（增加的值）。

只能在 RTC_CR WUTE 位 = 0 且 RTC_ISR WUTWF 位 = 1 时对该寄存器的位 2 到 0 执行写操作。

建议不要在日历小时递增时更改小时，因为这样做会屏蔽日历小时的增量。

ADD1H 和 SUB1H 的更改在下一秒生效。

此寄存器受写保护。第 885 页的RTC 寄存器写保护中介绍了写访问的过程。

注意： TSEDGE 发生更改时，必须复位 TSE 以避免对 TSF 的误操作。

29.6.4 RTC 初始化和状态寄存器 (RTC_ISR)

RTC initialization and status register

此寄存器受写保护（RTC_ISR[13:8] 位除外）。*第 885 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x0C

Backup 域复位值：0x0000 0007

系统复位：不受影响（INIT、INITF 和 RSF 位除外，它们在复位时被清零）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSF	RECALPF
														rc_w0	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAMP3F	TAMP2F	TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUTWF	ALRBWF	ALRAWF
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r

位 31:18 保留，必须保持复位值

位 17 **ITSF**：内部时间戳标志 (Internal time-stamp flag)
发生内部时间戳事件时，由硬件将此标志置 1。
该标志由软件写零清零，且必须通过向两个位写零将此标志与 TSF 位一起清零。

位 16 **RECALPF**：重新校准挂起标志 (Recalibration pending Flag)
当软件对 RTC_CALR 寄存器执行写操作时，RECALPF 状态标志将自动置“1”，指示 RTC_CALR 寄存器已屏蔽。当采用新的校准设置时，该位恢复为 “0”。请参见*动态重校准*。

位 15 **TAMP3F**：RTC_TAMP3 检测标志 (RTC_TAMP3 detection flag)
在 RTC_TAMP3 输入上检测到入侵检测事件时，由硬件将此标志置 1。
该标志由软件写零清除。

位 14 **TAMP2F**：RTC_TAMP2 检测标志 (RTC_TAMP2 detection flag)
在 RTC_TAMP2 输入上检测到入侵检测事件时，由硬件将此标志置 1。
该标志由软件写零清除。

位 13 **TAMP1F**：RTC_TAMP1 检测标志 (RTC_TAMP1 detection flag)
在 RTC_TAMP1 输入上检测到入侵检测事件时，由硬件将此标志置 1。
该标志由软件写零清除。

位 12 **TSOVF**：时间戳溢出标志 (Time-stamp overflow flag)
当在 TSF 已置 1 的情况下发生时间戳事件时，由硬件将此标志置 1。
该标志由软件写零清除。建议仅在 TSF 位清零之后再检查并清零 TSOVF 位。否则，如果时间戳事件恰好在清零 TSF 位之前刚刚发生，则溢出事件可能会被漏掉。

位 11 **TSF**：时间戳标志 (Time-stamp flag)
发生时间戳事件时，由硬件将此标志置 1。
该标志由软件写零清除。如果 ITSF 标志已置 1，必须通过向两个位写零将 TSF 与 ITSF 一起清零。

位 10 **WUTF**：唤醒定时器标志 (Wakeup timer flag)
当唤醒自动重载计数器计数到 0 时，由硬件将此标志置 1。
该标志由软件写零清除。
软件必须在 WUTF 再次置 1 的 1.5 个 RTCCLK 周期之前将该标志清零。

位 9 **ALRBF**：闹钟 B 标志 (Alarm B flag)

当时间/日期寄存器（RTC_TR 和 RTC_DR）与闹钟 B 寄存器 (RTC_ALRMBR) 匹配时，由硬件将该标志置 1。

该标志由软件写零清除。

位 8 **ALRAF**：闹钟 A 标志 (Alarm A flag)

当时间/日期寄存器（RTC_TR 和 RTC_DR）与闹钟 A 寄存器 (RTC_ALRMAR) 匹配时，由硬件将该标志置 1。

该标志由软件写零清除。

位 7 **INIT**：初始化模式 (Initialization mode)

0：自由运行模式

1：初始化模式，用于编程时间和日期寄存器（RTC_TR 和 RTC_DR）以及预分频器寄存器 (RTC_PRER)。计数器停止计数，当 INIT 被复位后，计数器从新值开始计数。

位 6 **INITF**：初始化标志 (Initialization flag)

当此位置 1 时，RTC 处于初始化状态，此时可更新事件、日期和预分频器寄存器。

0：不允许更新日历寄存器

1：允许更新日历寄存器

位 5 **RSF**：寄存器同步标志 (Registers synchronization flag)

每次将日历寄存器的值复制到影子寄存器（RTC_SSRx、RTC_TRx 和 RTC_DRx）时，都会由硬件将此位置 1。在初始化模式下、平移操作挂起时 (SHPF=1) 或在旁路影子寄存器模式 (BYPSHAD=1) 下，该位由硬件清零。该位还可由软件清零。

在初始化模式下，该位可由软件或硬件清零。

0：日历影子寄存器尚未同步

1：日历影子寄存器已同步

位 4 **INITS**：初始化状态标志 (Initialization status flag)

当日历年份字段不为 0 时（Backup 域复位状态），由硬件将该位置 1。

0：日历尚未初始化

1：日历已经初始化

位 3 **SHPF**：平移操作挂起 (Shift operation pending)

0：没有平移操作挂起

1：某个平移操作挂起

只要通过对 RTC_SHIFTR 寄存器执行写操作来启动平移操作，此标志便由硬件置 1。执行完相应的平移操作后，此标志由硬件清零。对 SHPF 位执行写入操作不起作用。

位 2 **WUTWF**：唤醒定时器写标志 (Wakeup timer write flag)

此位在 RTC_CR 中的 WUTE 位清零后，并在 2 个 RTCCLK 周期后由硬件置 1，在 WUTE 位置 1 后并在 2 个 RTCCLK 周期后清零。当 WUTE 位清零且 WUTWF 位置 1 时，可更改唤醒定时器的值。

0：不允许更新唤醒定时器配置

1：允许更新唤醒定时器配置

位 1 **ALRBWF**：闹钟 B 写标志 (Alarm B write flag)

在 RTC_CR 寄存器中的 ALRBE 位置 0 之后，当闹钟 B 的值可更改时，由硬件将该位置 1。

该位在初始化模式下由硬件清零。

0：不允许更新闹钟 B

1：允许更新闹钟 B

位 0 **ALRAWF**：闹钟 A 写标志 (Alarm A write flag)

在 RTC_CR 寄存器中的 ALRAE 位置 0 后，当闹钟 A 的值可更改时，由硬件将该位置 1。

该位在初始化模式下由硬件清零。

0：不允许更新闹钟 A

1：允许更新闹钟 A

注：　将 ALRAF、ALRBF、WUTF 和 TSF 位编程为 0 之后 2 个 APB 时钟周期，清零生效。

29.6.5 RTC 预分频器寄存器 (RTC_PRER)

RTC prescaler register

只能在初始化模式下对该寄存器执行写操作。必须通过两次独立的写访问执行初始化。请参见*第 885 页的日历初始化和配置*。

此寄存器受写保护。*第 885 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x10

Backup 域复位值：0x007F 00FF

系统复位：不受影响

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

位 31:23 保留，必须保持复位值

位 22:16 **PREDIV_A[6:0]**：异步预分频系数 (Asynchronous prescaler factor)
下面是异步分频系数的公式：
ck_apre 频率 = RTCCLK 频率/(PREDIV_A+1)

位 15 保留，必须保持复位值。

位 14:0 **PREDIV_S[14:0]**：同步预分频系数 (Synchronous prescaler factor)
下面是同步分频系数的公式：
ck_spre 频率 = ck_apre 频率/(PREDIV_S+1)

29.6.6 RTC 唤醒定时器寄存器 (RTC_WUTR)

RTC wakeup timer register

仅当 RTC_ISR 中的 WUTWF 置 1 时才可对该寄存器执行写操作。

此寄存器受写保护。*第 885 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x14

Backup 域复位值：0x0000 FFFF

系统复位：不受影响

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

位 31:16 保留，必须保持复位值

位 15:0 **WUT[15:0]**：唤醒自动重载值位 (Wakeup auto-reload value bits)

当使能唤醒定时器时（WUTE 置 1），每 (WUT[15:0] + 1) 个 ck_wut 周期将 WUTF 标志置 1 一次。ck_wut 周期通过 RTC_CR 寄存器的 WUCKSEL[2:0] 位进行选择。

当 WUCKSEL[2] = 1 时，唤醒定时器变为 17 位，WUCKSEL[1] 等效为 WUT[16]，即要重载到定时器的最高有效位。

WUTF 第一次置 1 发生在 WUTE 置 1 之后 (WUT+1) 个 ck_wut 周期。

禁止在 WUCKSEL[2:0]=011(RTCCLK/2) 时将 WUT[15:0] 设置为 0x0000。

29.6.7 RTC 闹钟 A 寄存器 (RTC_ALRMAR)

RTC alarm A register

仅当 RTC_ISR 中的 ALRAWF 置 1 时或在初始化模式下，才可以对该寄存器执行写操作。

此寄存器受写保护。*第 885 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x1C

Backup 域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **MSK4**：闹钟 A 日期掩码 (Alarm A date mask)
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

位 30 **WDSEL**：星期几选择 (Week day selection)
0：DU[3:0] 代表日期的个位
1：DU[3:0] 代表星期几 DT[1:0] 为无关位

位 29:28 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 27:24 **DU[3:0]**：日期的个位或日（BCD 格式）(Date units or day in BCD format)

位 23 **MSK3**：闹钟 A 小时掩码 (Alarm A hours mask)
0：如果小时匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，小时无关

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 **MSK2**：闹钟 A 分钟掩码 (Alarm A minutes mask)
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

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 **MSK1**：闹钟 A 秒掩码 (Alarm A seconds mask)
0：如果秒匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，秒无关

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)

29.6.8 RTC 闹钟 B 寄存器 (RTC_ALRMBR)

RTC alarm B register

仅当 RTC_ISR 中的 ALRBWF 置 1 时或在初始化模式下，才可以对该寄存器执行写操作。

此寄存器受写保护。*第 885 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x20

Backup 域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **MSK4**：闹钟 B 日期掩码 (Alarm B date mask)
0：如果日期和日匹配，则闹钟 B 置 1
1：在闹钟 B 比较中，日期和日无关

位 30 **WDSEL**：星期几选择 (Week day selection)
0：DU[3:0] 代表日期的个位
1：DU[3:0] 代表星期几 DT[1:0] 为无关位

位 29:28 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 27:24 **DU[3:0]**：日期个位或日（BCD 格式）(Date units or day in BCD format)

位 23 **MSK3**：闹钟 B 小时掩码 (Alarm B hours mask)
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

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 **MSK2**：闹钟 B 分钟掩码 (Alarm B minutes mask)
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

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 **MSK1**：闹钟 B 秒掩码 (Alarm B seconds mask)
0：如果秒匹配，则闹钟 B 置 1
1：在闹钟 B 比较中，秒无关

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)

29.6.9 RTC 写保护寄存器 (RTC_WPR)

RTC write protection register

偏移地址：0x24

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **KEY**：写保护关键字 (Write protection key)
可通过软件对该字节执行写操作。
读取该字节时，始终返回 0x00。
有关如何解锁 RTC 寄存器写保护的介绍，请参见 *RTC 寄存器写保护*。

29.6.10 RTC 亚秒寄存器 (RTC_SSR)

RTC sub second register

偏移地址：0x28

Backup 域复位值：0x0000 0000

系统复位：当 BYPSHAD = 0 时为 0x0000 0000。当 BYPSHAD = 1 时不受影响。

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

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

位 31:16 保留，必须保持复位值

位 15:0 **SS**：亚秒值 (Sub seconds value)
SS[15:0] 是同步预分频器计数器的值。此亚秒值可根据以下公式得出：
亚秒值 = (PREDIV_S - SS) / (PREDIV_S + 1)
注： 仅当执行平移操作之后，SS 才能大于 PREDIV_S。在这种情况下，正确的时间/日期比 RTC_TR/RTC_DR 所指示的时间/日期慢一秒钟。

29.6.11 RTC 平移控制寄存器 (RTC_SHIFTR)

RTC shift control register

此寄存器受写保护。*第 885 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x2C

Backup 域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SUBFS[14:0]														
	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31 **ADD1S**：增加一秒钟 (Add one second)

0：无影响

1：对时钟/日历增加一秒钟

此位为只写位且始终读为 0。当平移操作挂起（RTC_ISR 中的 SHPF=1）时，对该位执行写操作无作用。

此函数应与 SUBFS 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。

位 30:15 保留，必须保持复位值

位 14:0 **SUBFS**：减少亚秒值 (Subtract a fraction of a second)

此位为只写位且始终读为 0。当平移操作挂起（RTC_ISR 中的 SHPF=1）时，对该位执行写操作无作用。

写入 SUBFS 的值将加到同步预分频器计数器中。由于该计数器递减计数，此操作可有效地从时钟减去（延迟）以下时间：

延迟（秒）= SUBFS / (PREDIV_S + 1)

当 ADD1S 函数与 SUBFS 结合使用时，可有效地将亚秒值增加到时钟（提前时钟），使时钟提前以下时间：

提前（秒）= (1 - (SUBFS / (PREDIV_S + 1)))。

注：　对 SUBFS 执行写操作将使 RSF 清零。软件随后会等待至 RSF=1 以确定影子寄存器已更新为平移后的时间。

29.6.12 RTC 时间戳时间寄存器 (RTC_TSTR)

RTC timestamp time register

仅当 RTC_ISR 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

偏移地址：0x30

Backup 域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]			
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	r	r	r	r	r	r	r		r	r	r	r	r	r	r

位 31:23 保留，必须保持复位值

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 保留，必须保持复位值

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 保留，必须保持复位值

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)

29.6.13 RTC 时间戳日期寄存器 (RTC_TSDR)

RTC timestamp date register

仅当 RTC_ISR 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

偏移地址：0x34

Backup 域复位值：0x0000 0000

系统复位：不受影响

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[1:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
r	r	r	r	r	r	r	r			r	r	r	r	r	r

位 31:16 保留，必须保持复位值

位 15:13 **WDU[1:0]**：星期几的个位 (Week day units)

位 12 **MT**：月份的十位（BCD 格式）(Month tens in BCD format)

位 11:8 **MU[3:0]**：月份的个位（BCD 格式）(Month units in BCD format)

位 7:6 保留，必须保持复位值

位 5:4 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 3:0 **DU[3:0]**：日期的个位（BCD 格式）(Date units in BCD format)

29.6.14 RTC 时间戳亚秒寄存器 (RTC_TSSSR)

RTC time-stamp sub second register

仅当 RTC_ISR/TSF 置 1 时，该寄存器的内容才有效。当 RTC_ISR/TSF 位复位时，清零该寄存器。

偏移地址：0x38

Backup 域复位值：0x0000 0000

系统复位：不受影响

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

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

位 31:16 保留，必须保持复位值

位 15:0 **SS**：亚秒值 (Sub seconds value)

当发生时间戳事件时，SS[15:0] 是同步预分频器计数器的值。

29.6.15 RTC 校准寄存器 (RTC_CALR)

RTC calibration register

此寄存器受写保护。*第 885 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x3C

Backup 域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALP	CALW8	CALW 16	Res.	Res.	Res.	Res.	CALM[8:0]								
rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值

位 15 **CALP**：将 RTC 的频率增加 488.5 ppm (Increase frequency of RTC by 488.5 ppm)
0：不增加 RTCCLK 脉冲。
1：每 2^{11} 个脉冲有效插入一个 RTCCLK 脉冲（将频率增加 488.5 ppm）。
此功能应与 CALM 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 * CALP) - CALM。
请参见*第 29.3.11 节：RTC 精密数字校准*。

位 14 **CALW8**：使用 8 秒校准周期 (Use an 8-second calibration cycle period)
当 CALW8 置“1”时，选择 8 秒校准周期。
注： *CALW8=“1”时，CALM[1:0] 将始终保持为“00”。请参见第 29.3.11 节：RTC 精密数字校准。*

位 13 **CALW16**：使用 16 秒校准周期 (Use a 16-second calibration cycle period)
当 CALW16 置“1”时，选择 16 秒校准周期。如果 CALW8 = 1，则不能将该位置“1”。
注： *当 CALW16=“1”时，CALM[0] 将始终保持为“0”。请参见第 29.3.11 节：RTC 精密数字校准。*

位 12:9 保留，必须保持复位值

位 8:0 **CALM[8:0]**：负校准 (Calibration minus)
在 2^{20} 个 RTCCLK 脉冲内屏蔽 CALM 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 CALP 结合使用。请参见*第 888 页的第 29.3.11 节：RTC 精密数字校准*。

29.6.16 RTC 入侵配置寄存器 (RTC_TAMPCR)

RTC tamper configuration register

偏移地址：0x40

Backup 域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP3 MF	TAMP3 NO ERASE	TAMP3 IE	TAMP2 MF	TAMP2 NO ERASE	TAMP2 IE	TAMP1 MF	TAMP1 NO ERASE	TAMP1 IE
							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
TAMP PUDIS	TAMPPRCH [1:0]		TAMPFLT[1:0]		TAMPFREQ[2:0]			TAMP TS	TAMP3 TRG	TAMP3 E	TAMP2 TRG	TAMP2 E	TAMPI E	TAMP1 TRG	TAMP1 E
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24 **TAMP3MF**：入侵 3 屏蔽标志 (Tamper 3 mask flag)

0：入侵 3 事件生成触发事件，必须通过软件将 TAMP3F 清零来允许下一次入侵事件检测。

1：入侵 3 事件生成触发事件。屏蔽 TAMP3F 并由硬件在内部将其清零。不擦除备份寄存器。

注： TAMP3MF 置 1 时，不得使能入侵 3 中断。

位 23 **TAMP3NOERASE**：入侵 3 不擦除 (Tamper 3 no erase)

0：入侵 3 事件擦除备份寄存器。

1：入侵 3 事件不擦除备份寄存器。

位 22 **TAMP3IE**：入侵 3 中断使能 (Tamper 3 interrupt enable)

0：TAMPIE = 0 时禁止入侵 3 中断。

1：使能入侵 3 中断。

位 21 **TAMP2MF**：入侵 2 屏蔽标志 (Tamper 2 mask flag)

0：入侵 2 事件生成触发事件，如果要允许下一次入侵事件检测，必须通过软件将 TAMP2F 清零。

1：入侵 2 事件生成触发事件。屏蔽 TAMP2F 并由硬件在内部将其清零。不擦除备份寄存器。

注： TAMP2MF 置 1 时，不得使能入侵 2 中断。

位 20 **TAMP2NOERASE**：入侵 2 不擦除 (Tamper 2 no erase)

0：入侵 2 事件擦除备份寄存器。

1：入侵 2 事件不擦除备份寄存器。

位 19 **TAMP2IE**：入侵 2 中断使能 (Tamper 2 interrupt enable)

0：TAMPIE = 0 时禁止入侵 2 中断。

1：使能入侵 2 中断。

位 18 **TAMP1MF**：入侵 1 屏蔽标志 (Tamper 1 mask flag)

0：入侵 1 事件生成触发事件，必须通过软件将 TAMP1F 清零来允许下一次入侵事件检测。

1：入侵 1 事件生成触发事件。屏蔽 TAMP1F 并由硬件在内部将其清零。不擦除备份寄存器。

注： TAMP1MF 置 1 时，不得使能入侵 1 中断。

位 17 **TAMP1NOERASE**：入侵 1 不擦除 (Tamper 1 no erase)

0：入侵 1 事件擦除备份寄存器。

1：入侵 1 事件不擦除备份寄存器。

位 16 **TAMP1IE**：入侵 1 中断使能 (Tamper 1 interrupt enable)

0：TAMPIE = 0 时禁止入侵 1 中断。

1：使能入侵 1 中断。

位 15 **TAMPPUDIS**：RTC_TAMPx 上拉禁止 (RTC_TAMPx pull-up disable)

该位决定在每次采样之前是否对每个 RTC_TAMPx 引脚都进行预充电。

0：采样之前对 RTC_TAMPx 引脚进行预充电（使能内部上拉）

1：禁止对 RTC_TAMPx 引脚进行预充电。

位 14:13 **TAMPPRCH[1:0]**：RTC_TAMPx 预充电持续时间 (RTC_TAMPx precharge duration)

这些位决定了在每次采样之前激活上拉的持续时间。TAMPPRCH 对每个 RTC_TAMPx 输入都有效。

0x0：1 个 RTCCLK 周期

0x1：2 个 RTCCLK 周期

0x2：4 个 RTCCLK 周期

0x3：8 个 RTCCLK 周期

位 12:11 **TAMPFLT[1:0]**：RTC_TAMPx 过滤器计数 (RTC_TAMPx filter count)

这些位决定了为激活入侵事件所需的指定电平 (TAMP*TRG) 上的连续采样次数。TAMPFLT 对每个 RTC_TAMPx 输入都有效。

0x0：在 RTC_TAMPx 输入转变为有效电平的边沿激活入侵事件（RTC_TAMPx 输入上无内部上拉）。

0x1：在有效电平上连续执行 2 次采样后激活入侵事件。

0x2：在有效电平上连续执行 4 次采样后激活入侵事件。

0x3：在有效电平上连续执行 8 次采样后激活入侵事件。

位 10:8 **TAMPFREQ[2:0]**：入侵采样频率 (Tamper sampling frequency)

决定对每个 RTC_TAMPx 输入进行采样时的频率。

0x0：RTCCLK/32768（RTCCLK = 32768 Hz 时为 1 Hz）

0x1：RTCCLK/16384（RTCCLK = 32768 Hz 时为 2 Hz）

0x2：RTCCLK/8192（RTCCLK = 32768 Hz 时为 4 Hz）

0x3：RTCCLK/4096（RTCCLK = 32768 Hz 时为 8 Hz）

0x4：RTCCLK/2048（RTCCLK = 32768 Hz 时为 16 Hz）

0x5：RTCCLK/1024（RTCCLK = 32768 Hz 时为 32 Hz）

0x6：RTCCLK/512（RTCCLK = 32768 Hz 时为 64 Hz）

0x7：RTCCLK/256（RTCCLK = 32768 Hz 时为 128 Hz）

位 7 **TAMPTS**：发生入侵检测事件时激活时间戳 (Activate timestamp on tamper detection event)

0：发生入侵检测事件时不保存时间戳

1：发生入侵检测事件时保存时间戳

即便 RTC_CR 寄存器中的 TSE=0，TAMPTS 仍有效。

位 6 **TAMP3TRG**：RTC_TAMP3 输入的有效电平 (Active level for RTC_TAMP3 input)
如果 TAMPFLT ≠ 00：
0：RTC_TAMP3 输入保持低电平会触发入侵检测事件。
1：RTC_TAMP3 输入保持高电平会触发入侵检测事件。
如果 TAMPFLT != 00：
0：RTC_TAMP3 输入上升沿会触发入侵检测事件。
1：RTC_TAMP3 输入下降沿会触发入侵检测事件。

位 5 **TAMP3E**：RTC_TAMP3 检测使能 (RTC_TAMP3 detection enable)
0：禁止 RTC_TAMP3 输入检测
1：使能 RTC_TAMP3 输入检测

位 4 **TAMP2TRG**：RTC_TAMP2 输入的有效电平 (Active level for RTC_TAMP2 input)
如果 TAMPFLT != 00：
0：RTC_TAMP2 输入保持低电平会触发入侵检测事件。
1：RTC_TAMP2 输入保持高电平会触发入侵检测事件。
如果 TAMPFLT != 00：
0：RTC_TAMP2 输入上升沿会触发入侵检测事件。
1：RTC_TAMP2 输入下降沿会触发入侵检测事件。

位 3 **TAMP2E**：RTC_TAMP2 输入检测使能 (RTC_TAMP2 input detection enable)
0：禁止 RTC_TAMP2 检测
1：使能 RTC_TAMP2 检测

位 2 **TAMPIE**：入侵中断使能 (Tamper interrupt enable)
0：禁止入侵中断
1：使能入侵中断
注： 该位使能所有入侵引脚事件的中断，无论 TAMPxIE 电平如何。如果将该位清零，可以通过将 TAMPxIE 置 1 分别使能每个入侵事件中断。

位 1 **TAMP1TRG**：RTC_TAMP1 输入的有效电平 (Active level for RTC_TAMP1 input)
如果 TAMPFLT != 00
0：RTC_TAMP1 输入保持低电平会触发入侵检测事件。
1：RTC_TAMP1 输入保持高电平会触发入侵检测事件。
如果 TAMPFLT != 00：
0：RTC_TAMP1 输入上升沿会触发入侵检测事件。
1：RTC_TAMP1 输入下降沿会触发入侵检测事件。

位 0 **TAMP1E**：RTC_TAMP1 输入检测使能 (RTC_TAMP1 input detection enable)
0：禁止 RTC_TAMP1 检测
1：使能 RTC_TAMP1 检测

注意： 如果 TAMPFLT = 0，则更改 TAMPxTRG 时必须复位 TAMPxE，以避免将 TAMPxF 意外置 1。

29.6.17 RTC 闹钟 A 亚秒寄存器 (RTC_ALRMASSR)

RTC alarm A sub second register

仅当 RTC_CR 中的 ALRAE 复位时或在初始化模式下，才可以对该寄存器执行写操作。

此寄存器受写保护。*第 885 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x44

Backup 域复位值：0x0000 0000

系统复位：不受影响

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

位 31:28 保留，必须保持复位值。

位 27:24 MASKSS[3:0]：屏蔽从此位开始的最高有效位 (Mask the most-significant bits starting at this bit)

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，SS[14:1] 为无关位。仅比较 SS[0]。
2：在闹钟 A 比较中，SS[14:2] 为无关位。仅比较 SS[1:0]。
3：在闹钟 A 比较中，SS[14:3] 为无关位。仅比较 SS[2:0]。
...
12：在闹钟 A 比较中，SS[14:12] 为无关位。比较 SS[11:0]。
13：在闹钟 A 比较中，SS[14:13] 为无关位。比较 SS[12:0]。
14：在闹钟 A 比较中，SS[14] 为无关位。比较 SS[13:0]。
15：所有 15 个 SS 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。

位 23:15 保留，必须保持复位值。

位 14:0 SS[14:0]：亚秒值 (Sub seconds value)

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 MASKSS-1。

29.6.18 RTC 闹钟 B 亚秒寄存器 (RTC_ALRMBSSR)

RTC alarm B sub second register

仅当 RTC_CR 中的 ALRBE 复位时或在初始化模式下，才可以对该寄存器执行写操作。

该寄存器受写保护。*RTC 寄存器写保护*一节中介绍了写访问的过程。

偏移地址：0x48

Backup 域复位值：0x0000 0000

系统复位：不受影响

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

位 31:28 保留，必须保持复位值。

位 27:24 **MASKSS[3:0]**：屏蔽从此位开始的最高有效位 (Mask the most-significant bits starting at this bit)

0x0：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）。
0x1：在闹钟 B 比较中，SS[14:1] 为无关位。仅比较 SS[0]。
0x2：在闹钟 B 比较中，SS[14:2] 为无关位。仅比较 SS[1:0]。
0x3：在闹钟 B 比较中，SS[14:3] 为无关位。仅比较 SS[2:0]。
...
0xC：在闹钟 B 比较中，SS[14:12] 为无关位。比较 SS[11:0]。
0xD：在闹钟 B 比较中，SS[14:13] 为无关位。比较 SS[12:0]。
0xE：在闹钟 B 比较中，SS[14] 为无关位。比较 SS[13:0]。
0xF：所有 15 个 SS 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。

位 23:15 保留，必须保持复位值。

位 14:0 **SS[14:0]**：亚秒值 (Sub seconds value)

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 MASKSS-1。

29.6.19 RTC 选项寄存器 (RTC_OR)

RTC option register

偏移地址：0x4C

Backup 域复位值：0x0000 0000

系统复位：不受影响

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

位 31:2 保留，必须保持复位值。

位 1 **RTC_OUT_RMP**：RTC_OUT 重映射 (RTC_OUT remap)

将此位置 1 可将 RTC 输出重映射到 PB2，具体如下：

RTC_OUT_RMP = “0”：

如果 OSEL/= “00”：RTC_ALARM 输出到 PC13

如果 OSEL= “00”且 COE = “1”：RTC_CALIB 输出到 PC13

RTC_OUT_RMP = “1”：

如果 OSEL /= “00”且 COE = “0”：RTC_ALARM 输出到 PB2

如果 OSEL = “00”且 COE = “1”：RTC_CALIB 输出到 PB2

如果 OSEL /= “00”且 COE = “1”：RTC_CALIB 输出到 PB2 且 RTC_ALARM 输出到 PC13

位 0 **RTC_ALARM_TYPE**：PC13 上的 RTC_ALARM 输出类型 (RTC_ALARM output type on PC13)

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

29.6.20 RTC 备份寄存器 (RTC_BKPxR)

RTC backup registers

偏移地址：0x50 到 0x9C

Backup 域复位值：0x0000 0000

系统复位：不受影响

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

位 31:0 BKP[31:0]

应用可向/从这些寄存器写入/读取数据。

当 V_{DD} 关闭时，这些寄存器由 V_{BAT} 供电，因而系统复位时，这些寄存器不会复位，并且当器件在低功耗模式下工作时，寄存器的内容仍然有效。

发生入侵检测事件时该寄存器会被复位，并且只要 TAMPxF=1，该寄存器就一直保持复位。

29.6.21 RTC 寄存器映射

表 172. RTC 寄存器映射和复位值

偏移	寄存器名称	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
0x00	**RTC_TR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]				Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[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
0x04	**RTC_DR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]				YU[3:0]				WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[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	1
0x08	**RTC_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSE	COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H	TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	Res.	FMT	BYPSHAD	REFCKON	TSEDGE	WUCKSEL[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
0x0C	**RTC_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSF	RECALPF	TAMP3F	TAMP2F	TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUT WF	ALRBWF	ALRAWF
	Reset value															0	0		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x10	**RTC_PRER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV_A[6:0]							PREDIV_S[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
0x14	**RTC_WUTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUT[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x1C	**RTC_ALRMAR**	MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]				MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[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
0x20	**RTC_ALRMBR**	MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]				MSK2	MNT[2:0]			MNU[3:0]				MSK2	ST[2:0]			SU[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
0x24	**RTC_WPR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY							
	Reset value																									0	0	0	0	0	0	0	0
0x28	**RTC_SSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**RTC_SHIFTR**	ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBFS[14:0]														
	Reset value	0																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	**RTC_TSTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]				Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[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

表 172. RTC 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x34	**RTC_TSDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDU[1:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
	Reset value																	0	0	0	0	0	0	0	0			0	0	0	0	0	0
0x38	**RTC_TSSSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**RTC_ CALR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALP	CALW8	CALW16	Res.	Res.	Res.	Res.	CALM[8:0]								
	Reset value																	0	0	0					0	0	0	0	0	0	0	0	0
0x40	**RTC_TAMPCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP3MF	TAMP3NOERASE	TAMP3IE	TAMP2MF	TAMP2NOERASE	TAMP2IE	TAMP1MF	TAMP1NOERASE	TAMP1IE	TAMPPUDIS	TAMPPRCH[1:0]		TAMPFLT[1:0]		TAMPFREQ[2:0]			TAMPTS	TAMP3TRG	TAMP3E	TAMP2TRG	TAMP2E	TAMPIE	TAMP1TRG	TAMP1E
	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
0x44	**RTC_ ALRMASSR**	Res.	Res.	Res.	Res.	MASKSS [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]														
	Reset value					0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**RTC_ ALRMBSSR**	Res.	Res.	Res.	Res.	MASKSS [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]														
	Reset value					0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	**RTC_ OR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTC_OUT_RMP	RTC_ALARM_TYPE
	Reset value																															0	0
0x50 to 0x9C	**RTC_BKP0R**	BKP[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
	to **RTC_BKP19R**	BKP[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

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

30 独立看门狗 (IWDG)

30.1 简介

此器件具有一个嵌入式看门狗外设，具有安全性高、定时准确及使用灵活的优点。此独立看门狗外设可检测并解决由软件错误导致的故障，并在计数器达到给定的超时值时触发系统复位。

独立看门狗 (IWDG) 由其专用低速时钟 (LSI) 驱动，因此即便在主时钟发生故障时仍然保持工作状态。

IWDG 最适合应用于需要看门狗作为一个在主程序之外，能够完全独立工作，并且对时间精度要求较低的应用。有关窗口看门狗的详细信息，请参见*第 925 页的第 31 节*。

30.2 IWDG 主要特性

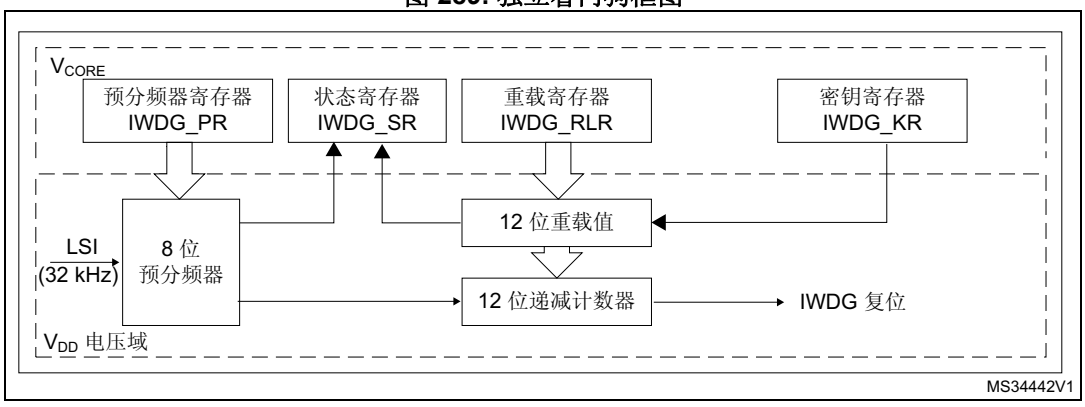
- 自由运行递减计数器
- 时钟由独立 RC 振荡器提供（可在待机和停止模式下运行）
- 复位条件
  - 当递减计数器值小于 0x000 时复位（如果看门狗已激活）
  - 在窗口之外重载递减计数器时复位（如果看门狗已激活）

30.3 IWDG 功能说明

30.3.1 IWDG 框图

图 289 给出了独立看门狗模块的功能框图。

图 289. 独立看门狗框图



1. 寄存器接口位于 V_{CORE} 电压域。看门狗功能位于 V_{DD} 电压域，在停止模式和待机模式下仍能工作。

通过向 *IWDG 键寄存器 (IWDG_KR)* 中写入值 0x0000 CCCC 来启动独立看门狗时，计数器开始从复位值 0xFFF 递减计数。当计数器计数到终值 (0x000) 时会产生一个复位信号（IWDG 复位）。

任何时候将键值 0x0000 AAAA 写到 *IWDG 键寄存器 (IWDG_KR)* 中，IWDG_RLR 的值就会被重载到计数器，从而避免产生看门狗复位。

一旦运行，IWDG 便无法停止。

30.3.2 窗口选项

通过在 *IWDG 窗口寄存器 (IWDG_WINR)* 中设置合适的窗口，IWDG 也可以用作窗口看门狗。

当计数器值大于 *IWDG 窗口寄存器 (IWDG_WINR)* 中存储的值时，如果执行重载操作，则会产生复位。

IWDG 窗口寄存器 (IWDG_WINR) 的默认值为 0x0000 0FFF，因此，如果不更新此默认值，将禁止窗口选项。

窗口值一经更改，便执行重载操作，以便将递减计数器复位为 *IWDG 重载寄存器 (IWDG_RLR)* 值，并方便计算周期数以生成下一次重载。

使能窗口选项时配置 IWDG

1. 通过在 *IWDG 键寄存器 (IWDG_KR)* 中写入 0x0000 CCCC 来使能 IWDG。
2. 通过在 *IWDG 键寄存器 (IWDG_KR)* 中写入 0x0000 5555 来使能寄存器访问。
3. 通过将 *IWDG 预分频器寄存器 (IWDG_PR)* 编程为 0~7 中的数值来配置 IWDG 预分频器。
4. 写 *IWDG 重载寄存器 (IWDG_RLR)*。
5. 等待寄存器更新 (IWDG_SR = 0x0000 0000)。
6. 写 *IWDG 窗口寄存器 (IWDG_WINR)*。这会自动刷新 *IWDG 重载寄存器 (IWDG_RLR)* 中的计数器值。

注： *当 IWDG 状态寄存器 (IWDG_SR) 设置为 0x0000 0000 时，写入窗口值允许刷新计数器值为 RLR 的值。*

禁止窗口选项时配置 IWDG

不使用窗口选项时，可按以下步骤配置 IWDG：

1. 通过在 *IWDG 键寄存器 (IWDG_KR)* 中写入 0x0000 CCCC 来使能 IWDG。
2. 通过在 *IWDG 键寄存器 (IWDG_KR)* 中写入 0x0000 5555 来使能寄存器访问。
3. 通过将 *IWDG 预分频器寄存器 (IWDG_PR)* 编程为 0~7 中的数值来配置预分频器。
4. 写 *IWDG 重载寄存器 (IWDG_RLR)*。
5. 等待寄存器更新 (IWDG_SR = 0x0000 0000)。
6. 刷新计数器值为 IWDG_RLR 的值 (IWDG_KR = 0x0000 AAAA)。

30.3.3 硬件看门狗

如果通过器件选项位使能“硬件看门狗”功能，上电时将自动使能看门狗；如果在计数器计数结束前，若软件没有向 *IWDG 键寄存器 (IWDG_KR)* 写入相应的值，或者在窗口内部重载了递减计数器，则系统会产生复位。

30.3.4 低功耗冻结

根据 IWDG_STOP 和 IWDG_STBY 选项配置，IWDG 可分别在停止模式和待机模式期间继续计数或停止计数。如果停止模式或待机模式期间 IWDG 保持运行，它可从此模式唤醒器件。更多详细信息，请参见*用户和读保护选项字节*。

30.3.5 寄存器访问保护

IWDG 预分频器寄存器 (IWDG_PR)、*IWDG 重载寄存器 (IWDG_RLR)* 和 *IWDG 窗口寄存器 (IWDG_WINR)* 具有写访问保护。若要对其进行修改，用户必须首先对 *IWDG 键寄存器 (IWDG_KR)* 写入代码 0x0000 5555。而写入其他值则会破坏该序列，从而使寄存器访问保护再次生效。这表示重载操作（即写入 0x0000 AAAA）也会启动写保护功能。

状态寄存器指示预分频值、递减计数器重载值或窗口值是否正在被更新。

30.3.6 调试模式

当器件进入调试模式时（内核停止），IWDG 计数器会根据 DBG 模块中的 DBG_IWDG_STOP 配置位选择继续正常工作或者停止工作。

30.4 IWDG 寄存器

有关寄存器说明中使用的缩写，请参见*第 58 页的第 1.2 节*。

外设寄存器可支持半字（16 位）或字（32 位）访问。

30.4.1 IWDG 键寄存器 (IWDG_KR)

IWDG key register

偏移地址：0x00

复位值：0x0000 0000（待机模式时复位）

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

位 31:16 保留，必须保持复位值。

位 15:0 **KEY[15:0]**：键值 (Key value)（只能写，读为 0x0000）

必须每隔一段时间便通过软件对这些位写入键值 0xAAAA，否则当计数器计数到 0 时，看门狗会产生复位。

写入键值 0x5555 可使能对 IWDG_PR、IWDG_RLR 和 IWDG_WINR 寄存器的访问（请参见*第 30.3.5 节：寄存器访问保护*）

写入键值 0xCCCC 可启动看门狗（选中硬件看门狗选项的情况除外）

30.4.2 IWDG 预分频器寄存器 (IWDG_PR)

IWDG prescaler register

偏移地址：0x04

复位值：0x0000 0000

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

位 31:3 保留，必须保持复位值。

位 2:0 **PR[2:0]**：预分频系数 (Prescaler divider)

这些位受写访问保护，请参见*第 30.3.5 节：寄存器访问保护*。通过软件设置这些位来选择计数器时钟的预分频因子。若要更改预分频器的分频系数，*IWDG 状态寄存器 (IWDG_SR)* 的 PVU 位必须为 0。

000：4 分频
001：8 分频
010：16 分频
011：32 分频
100：64 分频
101：128 分频
110：256 分频
111：256 分频

注： *读取该寄存器会返回 V_{DD} 电压域的预分频器值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/有效的。因此，只有在 IWDG 状态寄存器 (IWDG_SR) 中的 PVU 位为 0 时，从寄存器读取的值才有效。*

30.4.3 IWDG 重载寄存器 (IWDG_RLR)

IWDG reload register

偏移地址：0x08

复位值：0x0000 0FFF（待机模式时复位）

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

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

位 31:12 保留，必须保持复位值。

位 11:0 **RL[11:0]**：看门狗计数器重载值 (Watchdog counter reload value)

这些位受写访问保护，请参见*寄存器访问保护*。这个值由软件设置，每次对 *IWDG 键寄存器 (IWDG_KR)* 写入值 0xAAAA 时，这个值就会重装载到看门狗计数器中。之后，看门狗计数器便从该装载的值开始递减计数。超时周期由该值和时钟预分频器共同决定。有关超时信息，请参见数据手册。

若要更改重载值，*IWDG 状态寄存器 (IWDG_SR)* 中的 RVU 位必须为 0。

注： *读取该寄存器会返回 V_{DD} 电压域的重载值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/有效的。因此，只有在 IWDG 状态寄存器 (IWDG_SR) 中的 RVU 位为 0 时，从寄存器读取的值才有效。*

30.4.4 IWDG 状态寄存器 (IWDG_SR)

IWDG status register

偏移地址：0x0C

复位值：0x0000 0000（待机模式时不复位）

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

位 31:3 保留，必须保持复位值。

位 2 **WVU**：看门狗计数器窗口值更新 (Watchdog counter window value update)
该位由硬件置 1 以指示窗口值正在更新。当在 V_{DD} 电压域下完成重载值更新操作后（需要多达 5 个周期），会通过硬件将该位复位。
窗口值只有在 WVU 位为 0 时才可更新。
此位只有在通用“窗口”= 1 时才生成。

位 1 **RVU**：看门狗计数器重载值更新 (Watchdog counter reload value update)
该位由硬件置 1 以指示重载值正在更新。当在 V_{DD} 电压域下完成重载值更新操作后（需要多达 5 个周期），会通过硬件将该位复位。
重载值只有在 RVU 位为 0 时才可更新。

位 0 **PVU**：看门狗预分频器值更新 (Watchdog prescaler value update)
该位由硬件置 1 以指示预分频器值正在更新。当在 V_{DD} 电压域下完成预分频器值更新操作后（需要多达 5 个周期），会通过硬件将该位复位。
预分频器值只有在 PVU 位为 0 时才可更新。

注： *如果应用使用多个重载值、预分频器值或窗口值，则必须等到 RVU 位被复位后才能更改重载值，等到 PVU 位被复位后才能更改预分频器值，而且必须等到 WVU 位被复位后才能更改窗口值。但是，在更新预分频器和/或重载/窗口值之后，则无需等到 RVU、PVU 或 WVU 复位后再继续执行代码（进入低功耗模式时除外）。*

30.4.5 IWDG 窗口寄存器 (IWDG_WINR)

IWDG window register

偏移地址：0x10

复位值：0x0000 0FFF（待机模式时复位）

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

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

位 31:12 保留，必须保持复位值。

位 11:0 **WIN[11:0]**：看门狗计数器窗口值 (Watchdog counter window value)
这些位受写访问保护，请参见*第 30.3.5 节*，它们包含用于与递减计数器进行比较的窗口值上限。
为防止发生复位，当递减计数器的值低于窗口寄存器值且大于 0x0 时必须重载。
若要更改重载值，*IWDG 状态寄存器 (IWDG_SR)* 中的 WVU 位必须为 0。
注：读取该寄存器会返回 V_{DD} 电压域的重载值。如果正在对该寄存器执行写操作，则读取的值可能无效。因此，只有在 IWDG 状态寄存器 (IWDG_SR) 中的 WVU 位为 0 时，从寄存器读取的值才有效。

30.4.6 IWDG 寄存器映射

下表提供了 IWDG 寄存器映射和复位值。

表 173. IWDG 寄存器映射和复位值

偏移	寄存器名称	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
0x00	**IWDG_KR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**IWDG_PR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[2:0]		
	Reset value																														0	0	0
0x08	**IWDG_RLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RL[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x0C	**IWDG_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WVU	RVU	PVU
	Reset value																														0	0	0
0x10	**IWDG_WINR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WIN[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

31 系统窗口看门狗 (WWDG)

31.1 简介

系统窗口看门狗 (WWDG) 通常被用来监测，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。除非程序在 T6 位变成 0 前刷新递减计数器的值，否则看门狗电路在达到预置的时间周期时，会产生一个 MCU 复位。如果在递减计数器达到窗口寄存器值之前刷新控制寄存器中的 7 位递减计数器值，也会产生 MCU 复位。这意味着必须在限定的时间窗口内刷新计数器。

WWDG 时钟由 APB 时钟经预分频后提供，通过可配置的时间窗口来检测应用程序非正常的过迟或过早的操作。

WWDG 最适合那些要求看门狗在精确计时窗口起作用的应用程序。

31.2 WWDG 主要特性

- 可编程的自由运行递减计数器
- 复位条件
  - 当递减计数器值小于 0x40 时复位（如果看门狗已激活）
  - 在窗口之外重载递减计数器时复位（如果看门狗已激活）（请参见*图 291*）
- 提前唤醒中断 (EWI)：当递减计数器等于 0x40 时触发（如果已使能且看门狗已激活）

31.3 WWDG 功能说明

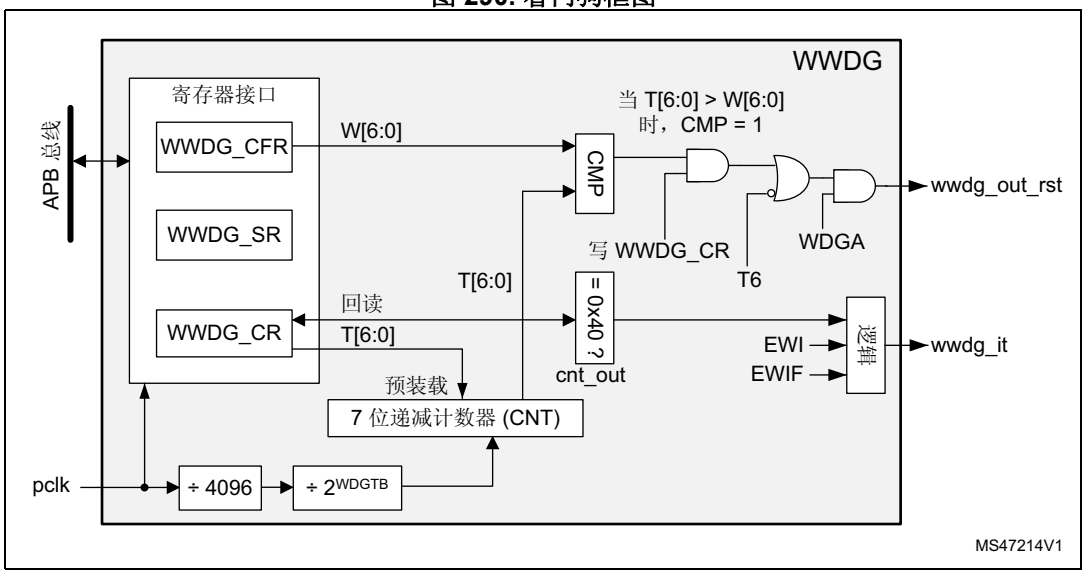
如果激活看门狗（WWDG_CR 寄存器中的 WDGA 位置 1），则当 7 位递减计数器（T[6:0] 位）从 0x40 递减到 0x3F（T6 已清零）时会引发复位。当计数器值大于窗口寄存器中所存储的值时，如果软件重载计数器，则会产生复位。

应用程序在正常运行过程中必须定期地写入 WWDG_CR 寄存器以防止 MCU 发生复位。只有当计数器值低于窗口寄存器值且高于 0x3F 时，才能执行此操作。要存储到 WWDG_CR 寄存器中的值必须介于 0xFF 和 0xC0 之间。

请参见*图 290* 了解 WWDG 框图。

31.3.1 WWDG 框图

图 290. 看门狗框图



31.3.2 使能看门狗

当用户选项 WWDG_SW 选择“软件窗口看门狗”时，看门狗在复位后总处于关闭状态。可通过设置 WWDG_CR 寄存器中的 WDGA 位来使能看门狗，之后除非执行复位操作，否则不能再次关闭。

当用户选项 WWDG_SW 选择“硬件窗口看门狗”时，看门狗在复位后始终使能，无法禁止。

31.3.3 控制递减计数器

递减计数器处于自由运行状态，即使禁止看门狗，递减计数器仍继续递减计数。当使能看门狗时，必须将 T6 位置 1，以防止立即复位。

T[5:0] 位包含了看门狗产生复位之前的计时数目；复位前的延时时间在一个最小值和一个最大值之间变化，这是因为写入 WWDG_CR 寄存器时，预分频器的状态是未知的（请参见 *图 291*）。配置寄存器 (WWDG_CFR) 包含窗口的上限：为防止发生复位，当递减计数器的值低于窗口寄存器值且大于 0x3F 时必须重载。*图 291* 介绍了窗口看门狗的工作过程。

注： *可使用 T6 位产生软件复位（将 WDGA 位置 1 并将 T6 位清零）。*

31.3.4 看门狗中断高级特性

如果在产生实际复位之前必须执行特定的安全操作或数据记录，则可使用提前唤醒中断 (EWI)。通过设置 WWDG_CFR 寄存器中的 EWI 位使能 EWI 中断。当递减计数器的值为 0x40 时，将生成 EWI 中断。在复位器件之前，可以使用相应的中断服务程序 (ISR) 来触发特定操作（例如通信或数据记录）。

在某些应用中，可以使用 EWI 中断来管理软件系统检查和/或系统恢复/功能退化，而不会生成 WWDG 复位。在这种情况下，相应的中断服务程序 (ISR) 可用来重载 WWDG 计数器以避免 WWDG 复位，然后再触发所需操作。

通过将 0 写入 WWDG_SR 寄存器中的 EWIF 位来清除 EWI 中断。

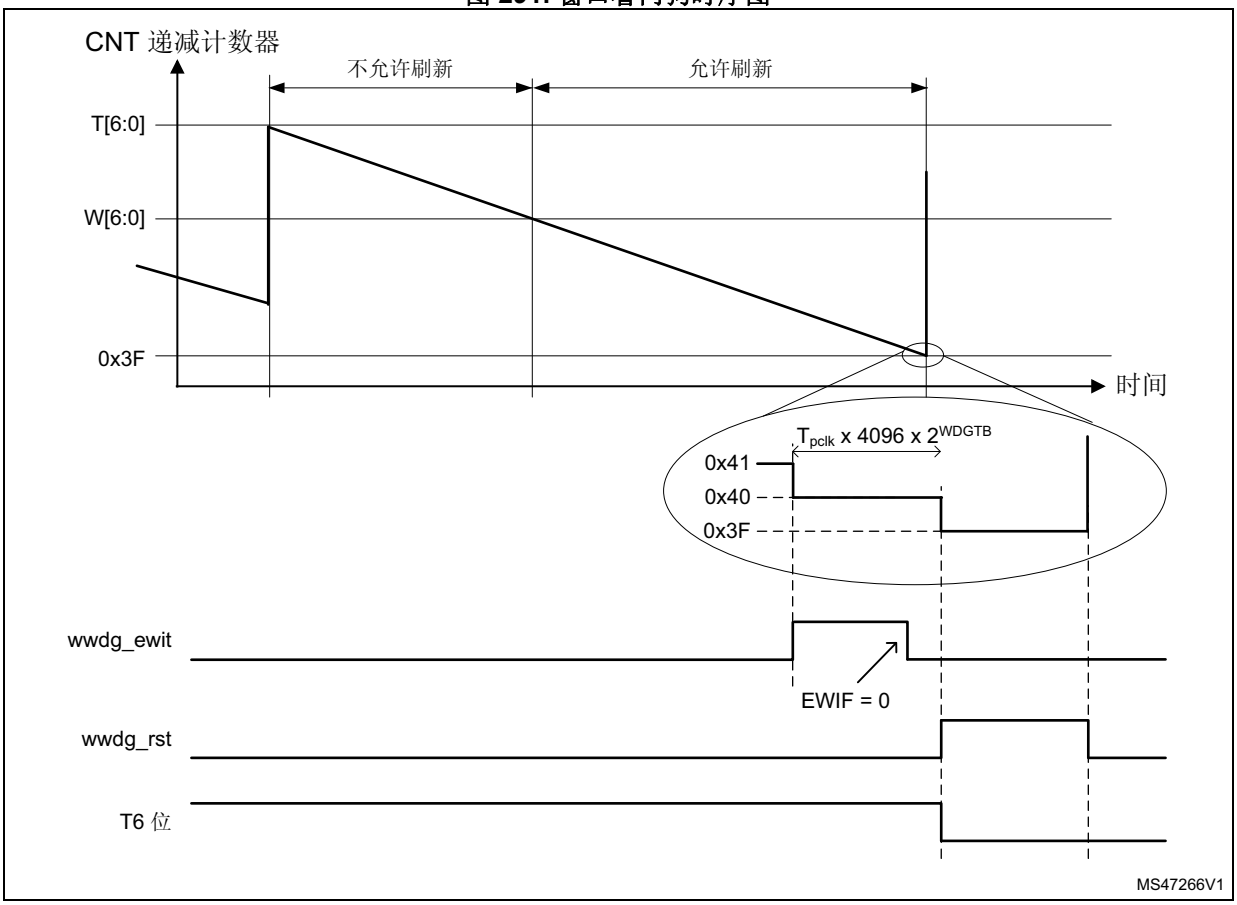
注： *当由于在更高优先级任务中有系统锁定而无法使用 EWI 中断时，最终会产生 WWDG 复位。*

31.3.5 如何设置看门狗超时

使用*图 291* 中的公式来计算 WWDG 超时。

警告： **写入 WWDG_CR 寄存器时，始终将 1 写入 T6 位，以避免生成立即复位。**

图 291. 窗口看门狗时序图



超时值的计算公式如下：

$$t_{WWDG} = t_{PCLK} \times 4096 \times 2^{WDGTB[2:0]} \times (T[5:0] + 1) \quad (ms)$$

其中：

t_{WWDG}：WWDG 超时

t_{PCLK}：APB 时钟周期，以 ms 为测量单位

4096：对应于内部分频器的值

例如，假设 APB 频率等于 48 MHz，将 WDGTB[2:0] 设置为 3 并将 T[5:0] 设置为 63：

$$t_{WWDG} = (1/48000) \times 4096 \times 2^{3} \times (63 + 1) = 43.69ms$$

有关 t_{WWDG} 的最小值和最大值，请参见数据手册。

31.3.6 调试模式

当器件进入调试模式时（处理器停止），WWDG 计数器会根据 DBG 模块中的配置位选择继续正常工作或者停止工作。有关详细信息，请参见*第 41 节：调试支持 (DBG)*。

31.4 WWDG 寄存器

有关寄存器说明中使用的缩写，请参见*第 58 页的第 1.2 节*。

外设寄存器可支持半字（16 位）或字（32 位）访问。

31.4.1 控制寄存器 (WWDG_CR)

Control register

偏移地址：0x000

复位值：0x0000 007F

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

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

位 31:8 保留，必须保持复位值。

位 7 **WDGA**：激活位 (Activation bit)
此位由软件置 1，只有复位后才由硬件清零。当 WDGA = 1 时，看门狗可产生复位。
0：禁止看门狗
1：使能看门狗

位 6:0 **T[6:0]**：7 位计数器 (7-bit counter)（MSB 到 LSB）
这些位用来存储看门狗计数器的值，每隔 (4096 x $2^{WDGTB[1:0]}$) PCLK 个周期递减一次。当它从 0x40 递减到 0x3F（T6 清零）时会产生复位。

31.4.2 配置寄存器 (WWDG_CFR)

Configuration register

偏移地址：0x004

复位值：0x0000 007F

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

位 31:14 保留，必须保持复位值。

位 13:11 **WDGTB[2:0]**：定时器时基 (Timer base)
可按如下方式修改预分频器的时基：
000：CK 计数器时钟 (PCLK div 4096) 分频器 1
001：CK 计数器时钟 (PCLK div 4096) 分频器 2
010：CK 计数器时钟 (PCLK div 4096) 分频器 4
011：CK 计数器时钟 (PCLK div 4096) 分频器 8
100：CK 计数器时钟 (PCLK div 4096) 分频器 16
101：CK 计数器时钟 (PCLK div 4096) 分频器 32
110：CK 计数器时钟 (PCLK div 4096) 分频器 64
111：CK 计数器时钟 (PCLK div 4096) 分频器 128

位 10 保留，必须保持复位值。

位 9 **EWI**：提前唤醒中断 (Early wakeup interrupt)
置 1 后，只要计数器值达到 0x40 就会产生中断。此中断只有在复位后才由硬件清零。

位 8:7 保留，必须保持复位值。

位 6:0 **W[6:0]**：7 位窗口值 (7-bit window value)
这些位包含用于与递减计数器进行比较的窗口值。

31.4.3 状态寄存器 (WWDG_SR)

Status register

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF
															rc_w0

位 31:1 保留，必须保持复位值。

位 0 **EWIF**：提前唤醒中断标志 (Early wakeup interrupt flag)

当计数器值达到 0x40 时此位由硬件置 1。它必须由软件通过写入 0 来清零。写入“1”无影响。如果不使能中断，此位也会被置 1。

31.4.4 WWDG 寄存器映射

下表提供了 WWDG 寄存器映射和复位值。

表 174. WWDG 寄存器映射和复位值

偏移	寄存器	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
0x000	**WWDG_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGA	T[6:0]						
	Reset value																									0	1	1	1	1	1	1	1
0x004	**WWDG_CFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGTB [2:0]			Res.	EWI	Res.	Res.	W[6:0]							
	Reset value																		0	0	0		0			1	1	1	1	1	1	1
0x008	**WWDG_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF
	Reset value																																0

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

32 内部集成电路 (I2C) 接口

32.1 简介

I^2C（内部集成电路）总线接口处理微控制器与串行 I^2C 总线间的通信。它提供多主模式功能，可以控制所有 I^2C 总线特定的序列、协议、仲裁和时序。它支持标准模式 (Sm)、快速模式 (Fm) 和超快速模式 (Fm+)。

它还与 SMBus（系统管理总线）和 PMBus（电源管理总线）兼容。

可使用 DMA 来减轻 CPU 的工作量。

32.2 I2C 主要特性

- 兼容 I^2C 总线规范第 03 版：
  - 从模式和主模式
  - 多主模式功能
  - 标准速度模式（高达 100 kHz）
  - 快速模式（高达 400 kHz）
  - 超快速模式（高达 1 MHz）
  - 7 位和 10 位寻址模式
  - 多个 7 位从地址（2 个从设备地址寄存器，1 个具有可配置的掩码位段）
  - 所有 7 位地址应答模式
  - 广播呼叫
  - 总线上的数据建立和保持时间可软件配置
  - 方便易用的事件管理
  - 可选的时钟延长
  - 软件复位
- 带 DMA 功能的 1 字节缓冲
- 可编程模拟和数字噪声滤波器

还可额外提供以下特性，具体取决于产品实现（请参见*第 32.3 节：I2C 特性实现*）：

- 兼容 SMBus 规范第 3.0 版：
  - 具有 ACK 控制的硬件 PEC（数据包错误校验）生成和验证
  - 命令和数据应答控制
  - 支持地址解析协议 (ARP)
  - 支持主机和从设备
  - SMBus 报警
  - 超时和空闲条件检测
- 兼容 PMBus 第 1.3 版标准
- 独立时钟：选择独立时钟源可使 I2C 通信速度不受 PCLK 时钟频率更改的影响
- 地址匹配时从停止模式唤醒

32.3 I2C 特性实现

表 175. STM32WB55xx I2C 实现

I2C 特性(1)	I2C1	I2C3
7 位寻址模式	X	X
10 位寻址模式	X	X
标准模式（高达 100 kb/s）	X	X
快速模式（高达 400 kb/s）	X	X
超快速模式，20 mA 输出驱动 I/O（高达 1 Mb/s）	X	X
独立时钟	X	X
SMBus/PMbus	X	-
地址匹配时从停止 0/停止 1 模式唤醒	X	X
地址匹配时从停止 2 模式唤醒	-	X

1. X = 支持。

32.4 I2C 功能说明

除了接收和发送数据之外，此接口还可以从串行格式转换为并行格式，反之亦然。中断由软件使能或禁止。该接口通过数据引脚 (SDA) 和时钟引脚 (SCL) 连接到 I^2C 总线。它可以连接到标准速度（高达 100 kHz）、快速（高达 400 kHz）或超快速（高达 1 MHz）I^2C 总线。

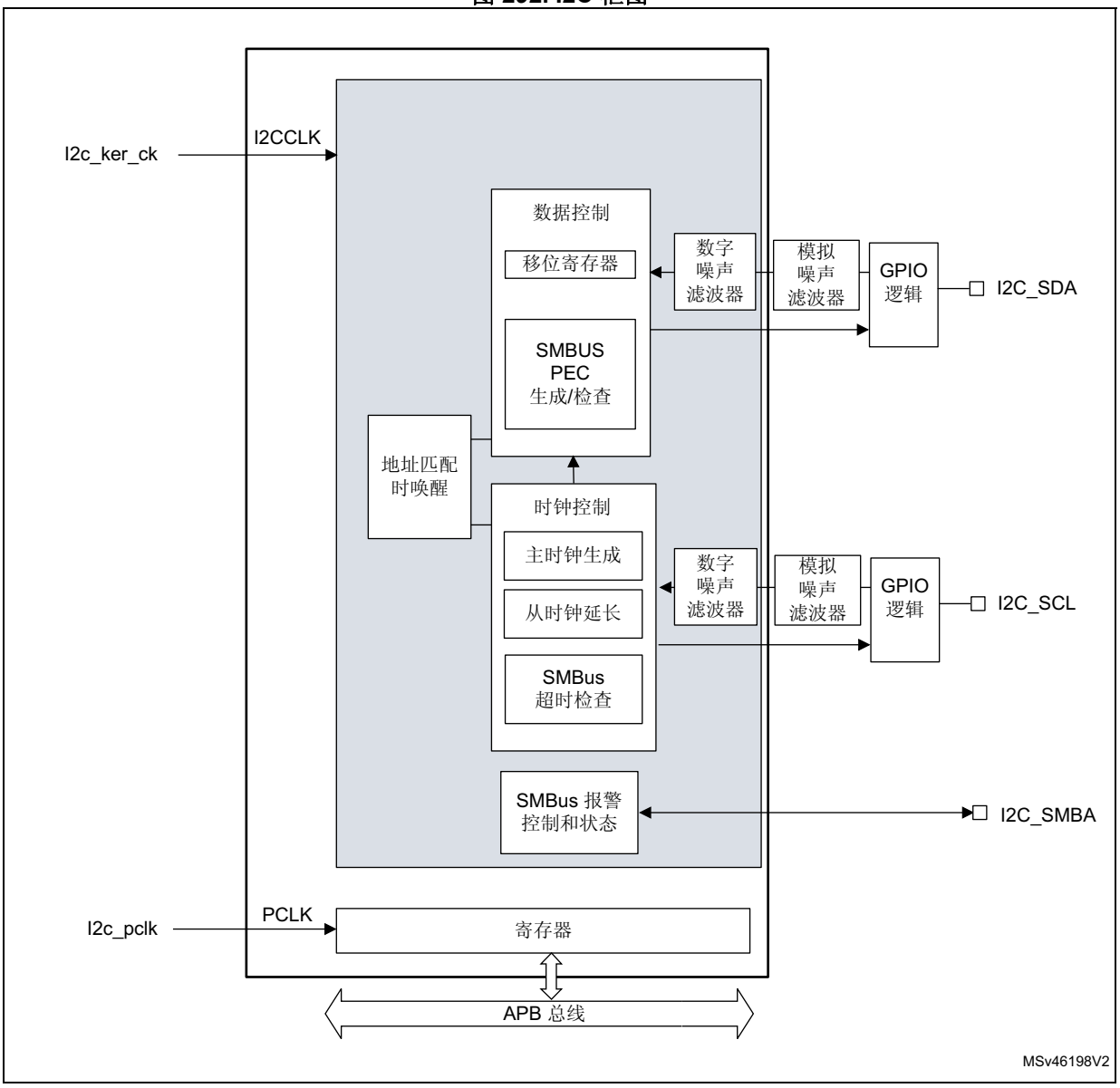
该接口也可通过数据引脚 (SDA) 和时钟引脚 (SCL) 连接到 SMBus。

如果支持 SMBus 功能：还可使用额外的可选 SMBus 报警引脚 (SMBA)。

32.4.1 I2C 框图

I2C 接口如*图 292* 所示。

图 292. I2C 框图



I2C 的时钟由独立时钟源提供，这使得 I2C 能够独立于 PCLK 频率工作。

对于支持以 20 mA 输出电流驱动超快速模式操作的 I2C I/O，可以通过系统配置控制器 (SYSCFG) 中的控制位使能驱动功能。*请参见第 32.3 节：I2C 特性实现。*

32.4.2 I2C 引脚和内部信号

表 176. I2C 输入/输出引脚

引脚名称	信号类型	说明
I2C_SDA	双向	I2C 数据
I2C_SCL	双向	I2C 时钟
I2C_SMBA	双向	SMBus 报警

表 177. I2C 内部输入/输出信号

内部信号名称	信号类型	说明
i2c_ker_ck	输入	I2C 内核时钟，在本文档中也称为 I2CCLK
i2c_pclk	输入	I2C APB 时钟
i2c_it	输出	I2C 中断，有关中断源的完整列表，请参见 *表 190：I2C 中断请求*
i2c_rx_dma	输出	I2C 接收数据 DMA 请求 (I2C_RX)
i2c_tx_dma	输出	I2C 发送数据 DMA 请求 (I2C_TX)

32.4.3 I2C 时钟要求

I2C 内核的时钟由 I2CCLK 提供。

I2CCLK 周期 t_{I2CCLK} 必须遵循以下条件：

$t_{I2CCLK} < (t_{LOW} - t_{filters}) / 4$ 且 $t_{I2CCLK} < t_{HIGH}$

其中：

t_{LOW}：SCL 低电平时间；t_{HIGH}：SCL 高电平时间

$t_{filters}$：滤波器使能时，该值为模拟滤波器和数字滤波器引入的延时总和。

模拟滤波器延时最大值为 260 ns。数字滤波器延时为 DNF x t_{I2CCLK}。

PCLK 时钟周期 t_{PCLK} 必须遵循以下条件：

$t_{PCLK} < 4/3\ t_{SCL}$

其中，t_{SCL}：SCL 周期

注意： 当 I2C 内核的时钟由 PCLK 提供时，该时钟必须遵循 t_{I2CCLK} 的条件。

32.4.4 模式选择

该接口在工作时可选用以下四种模式之一：

- 从发送器
- 从接收器
- 主发送器
- 主接收器

默认情况下，它以从模式工作。接口在生成起始位后会自动由从模式切换为主模式，并在出现仲裁丢失或生成停止位时从主模式切换为从模式，从而实现多主模式功能。

通信流程

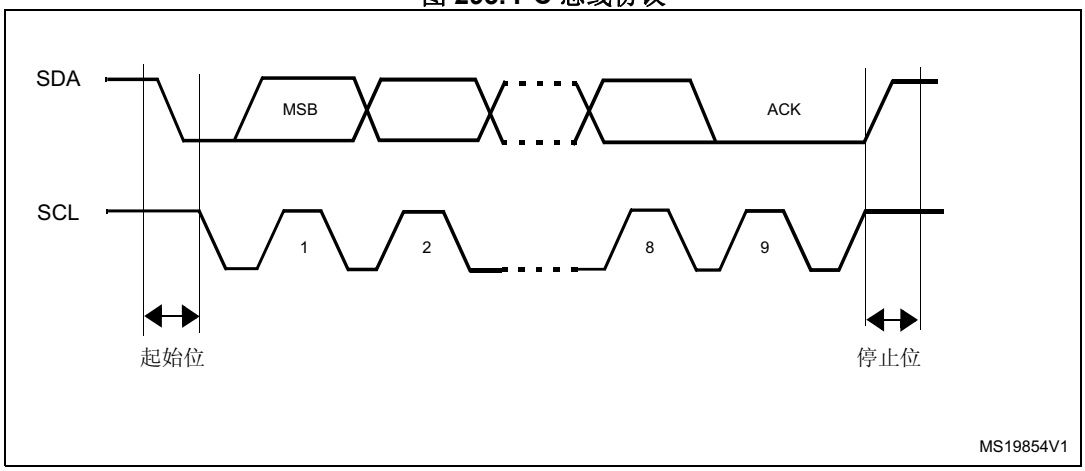
在主模式下，I2C 接口会启动数据传输并生成时钟信号。串行数据传输始终是在出现起始位时开始，在出现停止位时结束。起始位和停止位均在主模式下由软件生成。

在从模式下，该接口能够识别其自身地址（7 或 10 位）以及广播呼叫地址。广播呼叫地址检测可由软件使能或禁止。保留的 SMBus 地址也可由软件使能。

数据和地址均以 8 位字节传输，MSB 在前。起始位后紧随地址字节（7 位地址占据一个字节；10 位地址占据两个字节）。地址始终在主模式下传送。

在字节传输 8 个时钟周期后是第 9 个时钟脉冲，在此期间接收器必须向发送器发送一个应答位。请参见下图。

图 293. I^2C 总线协议



应答位可由软件使能或禁止。I2C 接口地址可通过软件进行选择。

32.4.5 I2C 初始化

使能和禁止外设

I2C 外设时钟必须在时钟控制器中进行配置和使能。

然后可通过将 I2C_CR1 寄存器中的 PE 位置 1 使能 I2C。

当禁止 I2C (PE=0) 时，I^2C 将执行软件复位。更多详细信息，请参见*第 32.4.6 节：软件复位*。

噪声滤波器

通过将 I2C_CR1 寄存器中的 PE 位置 1 来使能 I2C 外设之前，如有必要，用户必须配置噪声滤波器。默认情况下，SDA 和 SCL 输入上集成了模拟噪声滤波器。该模拟滤波器符合 I^2C 规范，此规范要求在快速模式和超快速模式下对脉宽在 50 ns 以下的脉冲都要抑制。用户可通过将 ANFOFF 位置 1 来禁止该模拟滤波器，通过配置 I2C_CR1 寄存器中的 DNF[3:0] 位来选择数字滤波器。

使能数字滤波器时，SCL 或 SDA 线的电平只有在电平稳定时间超过 DNF x I2CCLK 个周期后才会发生内部变化。从而可抑制的尖峰脉宽在 1 到 15 个 I2CCLK 周期可编程。

表 178. 模拟滤波器与数字滤波器对比

-	模拟滤波器	数字滤波器
抑制的脉冲宽度	≥ 50 ns	从 1 到 15 个 I2C 外设时钟的可编程长度
优点	停止模式中仍可用	– 长度可编程：额外的滤波能力与标准要求 – 稳定长度
缺点	随温度、电压和过程变化会发生变化	当使能数字滤波器后，无法在地址匹配时从停止模式唤醒

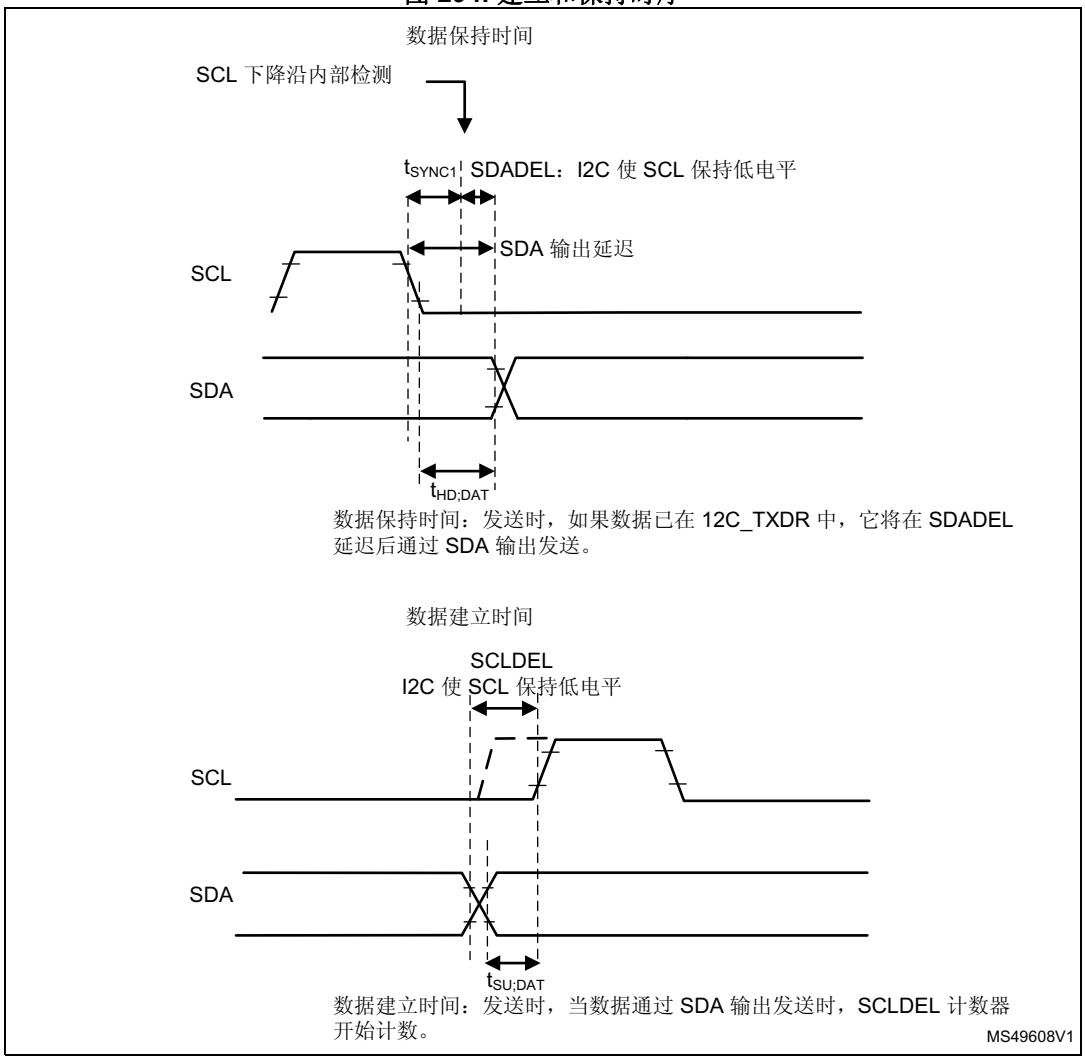
注意： 使能 I2C 时，不允许更改滤波器配置。

I2C 时序

必须配置时序，以便保证主模式和从模式下使用正确的数据保持和建立时间。配置方法是编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLDEL[3:0] 和 SDADEL[3:0] 位。

STM32CubeMX 工具进行计算并将 I2C_TIMINGR 内容提供在 I2C 配置窗口中

图 294. 建立和保持时序



- 当内部检测到 SCL 下降沿时，会在发送 SDA 输出之前插入一段延时。该延时为 t_{SDADEL} = SDADEL x t_{PRESC} + t_{I2CCLK}，其中 t_{PRESC} = (PRESC+1) x t_{I2CCLK}。
  T_{SDADEL} 会影响保持时间 $t_{HD;DAT}$。

SDA 总输出延时为：

t_{SYNC1} + {[SDADEL x (PRESC+1) + 1] x t_{I2CCLK} }

t_{SYNC1} 持续时间取决于以下参数：

- SCL 下降斜率
- 模拟滤波器（使能时）引入的输入延时：$t_{AF(min)} < t_{AF} < t_{AF(max)}$
- 数字滤波器（使能时）引入的输入延时：t_{DNF} = DNF x t_{I2CCLK}
- SCL 与 I2CCLK 时钟建立同步而产生的延时（2 到 3 个 I2CCLK 周期）

为了桥接 SCL 下降沿的未定义区域，用户编程 SDADEL 时必须遵循以下条件：

$\{t_{f\ (max)} + t_{HD;DAT\ (min)} - t_{AF(min)} - [(DNF + 3) \times t_{I2CCLK}]\} / \{(PRESC + 1) \times t_{I2CCLK}\} \leq SDADEL$

$SDADEL \leq \{t_{HD;DAT\ (max)} - t_{AF(max)} - [(DNF+4) \times t_{I2CCLK}]\} / \{(PRESC + 1) \times t_{I2CCLK}\}$

注： *只有使能模拟滤波器时，公式中才包含 $t_{AF(min)}$ / $t_{AF(max)}$。有关 t_{AF} 值的信息，请参见器件数据手册。*

标准模式、快速模式和超快速模式下的 $t_{HD;DAT}$ 最大值分别可达 3.45 µs、0.9 µs 和 0.45 µs，但在一次转变时间内必须小于 $t_{VD;DAT}$ 的最大值。只有器件未延长 SCL 信号的低电平周期 (t_{LOW}) 时，才必须满足该最大值条件。如果时钟延长 SCL，数据必须在建立时间内保持有效，之后才能释放时钟。

SDA 上升沿通常为最坏情况，因此在这种情况下，上述公式变成如下形式：

$SDADEL \leq \{t_{VD;DAT\ (max)} - t_{r\ (max)} - 260\ ns - [(DNF+4) \times t_{I2CCLK}]\} / \{(PRESC + 1) \times t_{I2CCLK}\}$。

注： *NOSTRETCH=0 时会违反该条件，这是因为器件会根据 SCLDEL 值来延长 SCL 低电平时间，以保证建立时间。*

有关 t_f、t_r、$t_{HD;DAT}$ 和 $t_{VD;DAT}$ 标准值的信息，请参见*表 179：I2C-SMBUS 规范数据建立和保持时间*。

- 在 t_{SDADEL} 延时后，或在因数据未写入 I2C_TXDR 寄存器而导致从器件必须延长时钟的情况下发送 SDA 输出后，SCL 线会在建立时间内保持低电平。该建立时间为 t_{SCLDEL} = (SCLDEL+1) x t_{PRESC}，其中 t_{PRESC} = (PRESC+1) x t_{I2CCLK}。
  t_{SCLDEL} 会影响建立时间 $t_{SU;DAT}$。

为了桥接 SDA 跳变（上升沿通常为最坏情况）的未定义区域，编程 SCLDEL 时必须遵循以下条件：

$\{[t_{r\ (max)} + t_{SU;DAT\ (min)}] / [(PRESC+1)] \times t_{I2CCLK}]\} - 1 <= SCLDEL$

有关 t_r 和 $t_{SU;DAT}$ 标准值的信息，请参见*表 179：I2C-SMBUS 规范数据建立和保持时间*。

将使用的 SDA 和 SCL 跳变时间值就是应用中的值。使用最大值而非标准值会增加 SDADEL 和 SCLDEL 计算的约束条件，但能够确保任意应用的特性。

注： *在发送和接收模式下，对于每个时钟脉冲，检测到 SCL 下降沿后，I2C 主器件或从器件会至少在 [(SDADEL+SCLDEL+1) x (PRESC+1) + 1] x t_{I2CCLK} 期间内延长 SCL 低电平时间。在发送模式下，如果 SDADEL 计数器计数结束后数据还未写入 I2C_TXDR，则 I2C 会继续延长 SCL 低电平时间，直到写入下一个数据。随后，会将新数据 MSB 发送到 SDA 输出，SCLDEL 计数器将开始计数，同时会继续延长 SCL 低电平时间以确保提供充足的数据建立时间。*

如果从模式下 NOSTRETCH = 1，则 SCL 不会延长。因此，编程 SDADEL 时还必须确保提供充足的建立时间。

表 179. I²C-SMBUS 规范数据建立和保持时间

符号	参数	标准模式 (Sm)		快速模式 (Fm)		超快速模式 (Fm+)		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
$t_{HD;DAT}$	数据保持时间	0	-	0	-	0	-	0.3	-	µs
$t_{VD;DAT}$	数据有效时间	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	数据建立时间	250	-	100	-	50	-	250	-	ns
t_r	SDA 和 SCL 信号的上升时间	-	1000	-	300	-	120	-	1000	
t_f	SDA 和 SCL 信号的下降时间	-	300	-	300	-	120	-	300	

此外，在主模式下，必须通过编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLH[7:0] 和 SCLL[7:0] 位来配置 SCL 时钟的高电平和低电平。

- 当内部检测到 SCL 下降沿时，会在释放 SCL 输出之前插入一段延时。该延时为 t_{SCLL} = (SCLL+1) x t_{PRESC}，其中 t_{PRESC} = (PRESC+1) x t_{I2CCLK}。

  t_{SCLL} 会影响 SCL 低电平时间 t_{LOW}。
- 当内部检测到 SCL 上升沿时，会在将 SCL 输出强制为低电平之前插入一段延时。该延时为 t_{SCLH} = (SCLH+1) x t_{PRESC}，其中 t_{PRESC} = (PRESC+1) x t_{I2CCLK}。t_{SCLH} 会影响 SCL 高电平时间 t_{HIGH}。

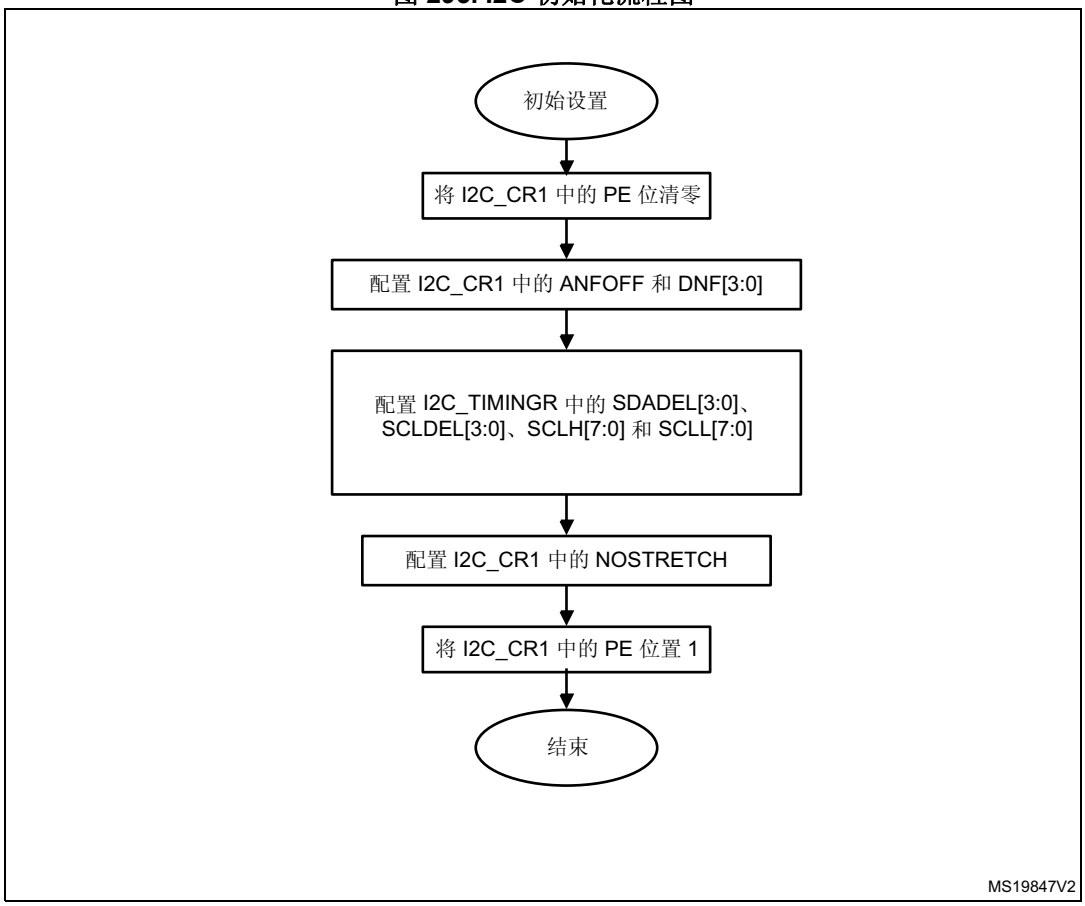
更多详细信息，请参见 *I2C 主模式初始化*。

注意： 使能 I2C 后，不允许更改时序配置。

此外，还必须在使能I2C 从设备前，对 NOSTRETCH 进行配置。更多详细信息，请参见 *I2C 从模式初始化*。

注意： 使能 I2C 后，不允许更改 NOSTRETCH 配置。

图 295. I2C 初始化流程图



32.4.6 软件复位

可通过将 I2C_CR1 寄存器中的 PE 位清零来执行软件复位。在这种情况下，I2C 线 SCL 和 SDA 被释放。内部状态机复位，通信控制位和状态位恢复为其复位值。配置寄存器不受影响。

下面列出了受影响的寄存器位：

1. I2C_CR2 寄存器：START、STOP 和 NACK
2. I2C_ISR 寄存器：BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、STOPF、BERR、ARLO 和 OVR

支持 SMBus 功能时还会影响到以下寄存器位：

1. I2C_CR2 寄存器：PECBYTE
2. I2C_ISR 寄存器：PECERR、TIMEOUT 和 ALERT

必须使 PE 保持低电平持续至少 3 个 APB 时钟周期，才能成功执行软件复位。使用以下软件写序列可确保这一点：- 写入 PE=0 - 检查 PE=0 - 写入 PE=1

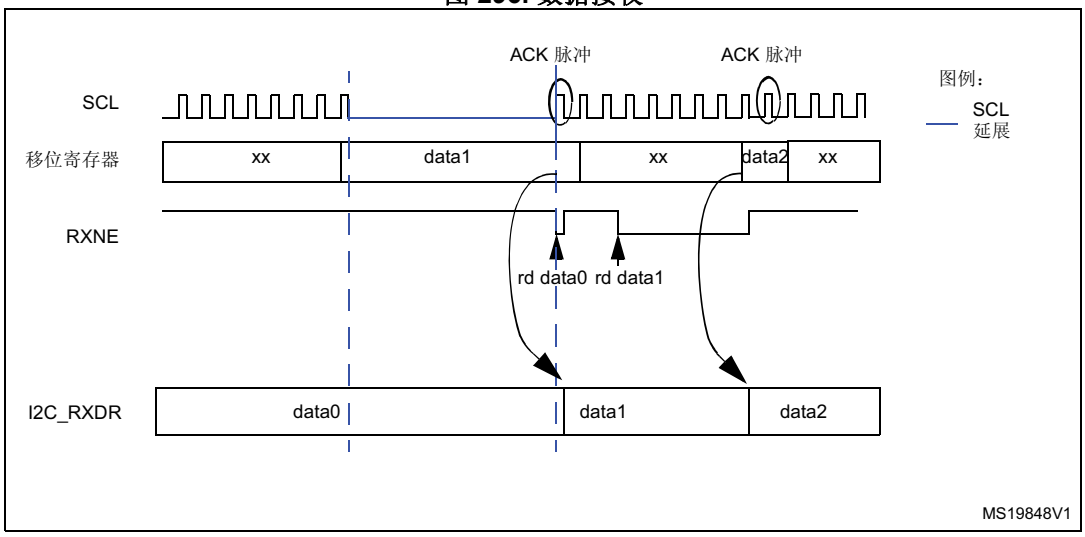
32.4.7 数据传输

数据传输由发送和接收数据寄存器以及移位寄存器来管理。

接收

SDA 输入填充移位寄存器。在第 8 个 SCL 脉冲后（接收到完整的数据字节时），如果 I2C_RXDR 寄存器为空 (RXNE=0)，则移位寄存器的内容会复制到其中。如果 RXNE=1（意味着尚未读取上一次接收到的数据字节），则将延长 SCL 线的低电平时间，直到读取了 I2C_RXDR 为止。在第 8 个和第 9 个 SCL 脉冲之间（应答脉冲之前）插入一段延长的时间。

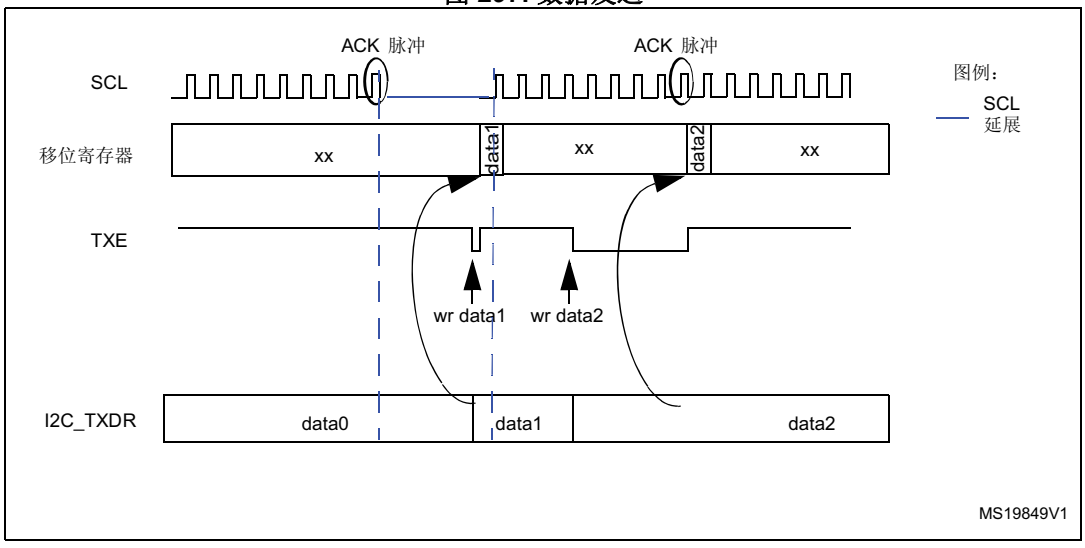
图 296. 数据接收



发送

如果 I2C_TXDR 寄存器不为空 (TXE=0)，则其内容会在第 9 个 SCL 脉冲（应答脉冲）后复制到移位寄存器中。然后移位寄存器的内容会移出到 SDA 线上。如果 TXE=1（意味着 I2C_TXDR 内尚未写入任何数据），则将延长 SCL 线的低电平时间，直到写入了 I2C_TXDR 为止。在第 9 个 SCL 脉冲后进行延长。

图 297. 数据发送



硬件传输管理

I2C 在硬件中内置了字节计数器，以便在下列各种模式下管理字节传输和结束通信：

- 主模式下生成 NACK、STOP 和 ReSTART
- 从接收器模式下控制 ACK 是否发出
- SMBus 模式下生成/校验 PEC

字节计数器通常在主模式下使用。在从模式下，字节计数器默认为禁止状态，但可以通过软件来使能，方法是将 I2C_CR2 寄存器中的 SBC（从字节控制）位置 1。

待传输的字节数在 I2C_CR2 寄存器的 NBYTES[7:0] 位域中进行编程。如果待传输的字节数 (NBYTES) 大于 255，或者接收方希望控制是否对接收到的数据字节进行应答，则必须选择重载模式，方法是将 I2C_CR2 寄存器的 RELOAD 位置 1。在该模式下，完成 NBYTES 中所编程字节数的数据传输之后，TCR 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TCR 标志置 1，SCL 便会延长。当往 NBYTES 写入一个非零值时，TCR 由软件清零。

在往 NBYTE 中设置最后一次传输的字节数前，必须把 RELOAD 位清零。

当主模式下 RELOAD=0 时，可在以下 2 种模式下使用计数器：

- **自动结束模式**（I2C_CR2 寄存器中的 AUTOEND = “1”）。在该模式下，一旦完成 NBYTES[7:0] 位域中所编程字节数的数据传输，主器件便会自动发送停止位。
- **软件结束模式**（I2C_CR2 寄存器中的 AUTOEND = “0”）在该模式下，一旦完成 NBYTES[7:0] 位域中所编程字节数的数据传输，TC 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TC 标志置 1，SCL 信号便会延长，需要软件介入操作。当软件把 I2C_CR2 寄存器中的起始位或停止位置 1 时，TC 标志将被清零。当主器件要发送重复起始位时，必须使用该模式。

注意： 当 RELOAD 位置 1 时，AUTOEND 位将不起作用。

表 180. I2C 配置

功能	SBC 位	RELOAD 位	AUTOEND 位
主 Tx/Rx NBYTES + STOP	x	0	1
主 Tx/Rx + NBYTES + RESTART	x	0	0
从 Tx/Rx 接收的所有字节都要回复应答	0	x	x
具有 ACK 控制的从 Rx	1	1	x

32.4.8 I2C 从模式

I2C 从模式初始化

要在从模式下工作，用户必须至少使能一个从地址。可使用 I2C_OAR1 和 I2C_OAR2 这两个寄存器来编程自身从地址 OA1 和 OA2。

- OA1 既可配置为 7 位寻址模式（默认），也可通过将 I2C_OAR1 寄存器的 OA1MODE 位置 1 配置为 10 位寻址模式。

  通过将 I2C_OAR1 寄存器中的 OA1EN 位置 1 来使能 OA1。

- 如果需要额外的从地址，可配置第 2 个从地址 OA2。将 I2C_OAR2 寄存器的 OA2MSK[2:0] 位置 1 最多可屏蔽 7 个 OA2 LSB。因此，当 OA2MSK 配置为 1 到 6 时，将分别只有 OA2[7:2]、OA2[7:3]、OA2[7:4]、OA2[7:5]、OA2[7:6] 或 OA2[7] 与接收到的地址作比较。只要 OA2MSK 不等于 0， OA2 的地址比较器便会排除 I2C 保留地址 （0000 XXX 和 1111 XXX），这些地址将不会得到应答。如果 OA2MSK=7，接收到的所有 7 位地址（保留地址除外）均得到应答。 OA2 始终为 7 位地址。

  如果这些保留地址在 I2C_OAR1 或 I2C_OAR2 寄存器中进行了编程并且 OA2MSK=0，则它们可以在通过特定使能位使能后得到应答。

  通过将 I2C_OAR2 寄存器中的 OA2EN 位置 1 来使能 OA2。
- 通过将 I2C_CR1 寄存器中的 GCEN 位置 1 来使能广播呼叫地址。

当通过 I2C 的其中一个使能地址来寻址到该 I2C 设备时，ADDR 中断状态标志将置 1，并且 ADDRIE 位置 1 时将生成中断。

默认情况下，从器件使用其时钟延长功能（即必要时延长 SCL 信号的低电平时间）来为软件操作的执行提供时机。如果主器件不支持时钟延长，则必须对 I2C 进行如下配置：将 I2C_CR1 寄存器中 NOSTRETCH 位置 1。

接收到 ADDR 中断后，如果使能多个地址，则用户必须读取 I2C_ISR 寄存器中的 ADDCODE[6:0] 位，以确定是哪个地址匹配。还必须检查 DIR 标志，以获悉传输方向。

带时钟延长的从模式 (NOSTRETCH = 0)

在默认模式下，I2C 从器件会在以下情况下延长 SCL 时钟：

- ADDR 标志置 1 时：接收到的地址与其中一个使能的从地址匹配。通过软件将 ADDRCF 位置 1 以清零 ADRR 标志时，将释放该时钟延展。
- 发送时，前一次数据传输已完成但 I2C_TXDR 寄存器中未写入任何新数据，或者 ADDR 标志清零 (TXE=1) 时未写入第一个数据字节。往 I2C_TXDR 寄存器中写入数据时，将释放该时钟延展。
- 接收时，尚未读取 I2C_RXDR 寄存器但新的数据接收已完成。读取 I2C_RXDR 时，将释放该时钟延展。
- 当从器件字节控制模式和重载模式（SBC=1 且 RELOAD=1）下 TCR = 1 时，这意味着最后一个数据字节已完成传输。通过向 NBYTES[7:0] 字段写入一个非零值以将 TCR 清零时，将释放该时钟延展。
- 在 SCL 下降沿检测之后，I2C 会延长 SCL 的低电平时间（不超过 $[(SDADEL+SCLDEL+1) \times (PRESC+1) + 1] \times t_{I2CCLK}$）。

不带时钟延长的从模式 (NOSTRETCH = 1)

当 I2C_CR1 寄存器中的 NOSTRETCH = 1 时，I2C 从器件不会延长 SCL 信号。

- ADDR 标志置 1 时，不会延长 SCL 时钟。
- 发送时，必须在与发送数据对应的第一个 SCL 脉冲出现之前，向 I2C_TXDR 寄存器写入数据。否则，会发生下溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。当第一次数据发送开始而 STOPF 位仍置 1（尚未清零）时，OVR 标志也将置 1。因此，如果写入下一次传输要发送的第一个数据后才清零上一次传输的 STOPF 标志，则会出现 OVR 状态，甚至对于待发送的第一个数据也是如此。
- 接收时，必须在下一个数据字节的第 9 个 SCL 脉冲（ACK 脉冲）出现之前，从 I2C_RXDR 寄存器读取数据。否则，会发生上溢， I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

从器件字节控制模式

要在从接收模式下实现字节 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从器件字节控制模式。这样符合 SMBus 标准。

要在从接收模式下实现字节 ACK 控制，必须选择重载模式 (RELOAD=1)。要控制每个字节，必须在 ADDR 中断子程序中将 NBYTES 初始化为 0x1，并在每接收一个字节后将 NBYTE 重载为 0x1。接收到字节后，TCR 位将置 1，从而延长 SCL 信号的第 8 个和第 9 个脉冲之间的低电平时间。用户可以从 I2C_RXDR 寄存器中读取数据，然后通过配置 I2C_CR2 寄存器中的 ACK 位来决定是否应答。通过将 NBYTES 编程为非零值来释放 SCL 延长：发送应答或不应答信号，然后可继续接收下一个字节。

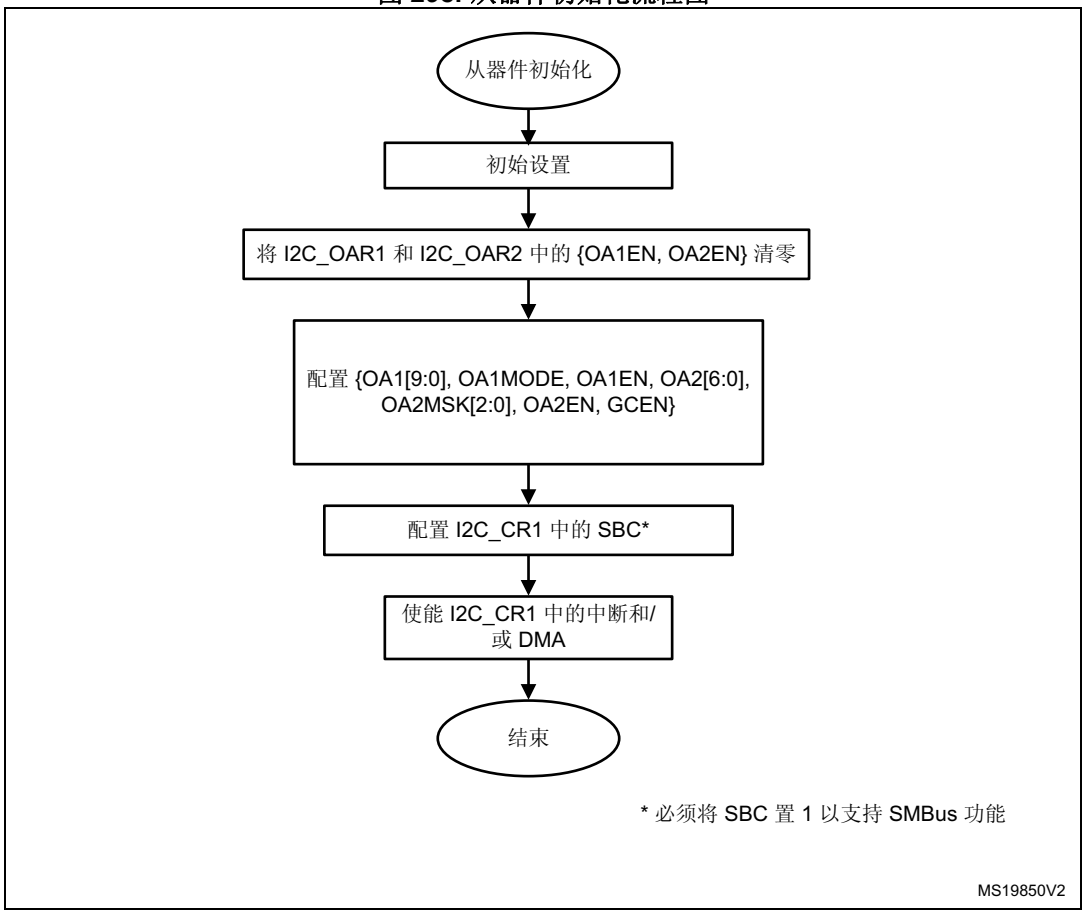
NBYTES 可加载大于 0x1 的值，在这种情况下，接收流在 NBYTES 个数据接收期间是连续的。

注： *SBC 位只能在 I2C 被禁止时、从器件不被寻址时或 ADDR=1 时配置。*

ADDR=1 或 TCR=1 时，可以更改 RELOAD 位的值。

注意： 从器件字节控制模式与 NOSTRETCH 模式不兼容。不允许在 NOSTRETCH=1 时将 SBC 位置 1。

图 298. 从器件初始化流程图



从发送器

当 I2C_TXDR 寄存器为空时，将生成发送中断状态 (TXIS)。如果 I2C_CR1 寄存器中的 TXIE 位置 1，将生成中断。

I2C_TXDR 寄存器中写入待发送的下一个数据字节时，TXIS 位将被清零。

接收到 NACK 时，I2C_ISR 寄存器中的 NACKF 位将置 1，如果 I2C_CR1 寄存器中的 NACKIE 位置 1，还将生成中断。从器件自动释放 SCL 和 SDA 线，以使主器件执行停止或重复起始位的发送。收到 NACK 时，TXIS 位不会置 1。

当接收到停止位且 I2C_CR1 寄存器中的 STOPIE 位置 1 时，I2C_ISR 寄存器中的 STOPF 标志将置 1 并且会生成中断。在大多数应用中，SBC 位通常编程为“0”。在这种情况下，如果接收到从地址 (ADDR=1) 时 TXE = 0，用户可以选择发送 I2C_TXDR 寄存器的内容作为第一个数据字节，也可以选择通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器以编程新的数据字节。

在从器件字节控制模式 (SBC=1) 下，必须在地址匹配中断子程序 (ADDR=1) 中向 NBYTE 写入待发送数据的个数。在这种情况下，传输期间 TXIS 事件的数量对应于 NBYTES 中编程的值。

注意： 如果 NOSTRETCH = 1，当 ADDR 标志置 1 时不会延长 SCL 时钟，因此用户无法在 ADDR 子程序中刷新 I2C_TXDR 寄存器的内容，从而编程第一个数据字节。必须在 I2C_TXDR 寄存器中预编程待发送的第一个数据字节：

- 该数据可以是前一个传输消息的最后一个 TXIS 事件中写入的数据。
- 如果该数据字节不是待发送的数据字节，可通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器，从而编程新的数据字节。必须仅在执行完这些操作后再清零 STOPF 位，以确保在地址应答之后第一次数据传输之前执行这些操作。

  如果第一次数据传输开始时 STOPF 仍置 1，则将生成下溢错误（OVR 标志置 1）。

  如果需要 TXIS 事件（发送中断或发送 DMA 请求），用户必须将 TXE 位和 TXIS 位均置 1，以便生成 TXIS 事件。

图 299. I2C 从发送器的传输序列流程图 (NOSTRETCH= 0)

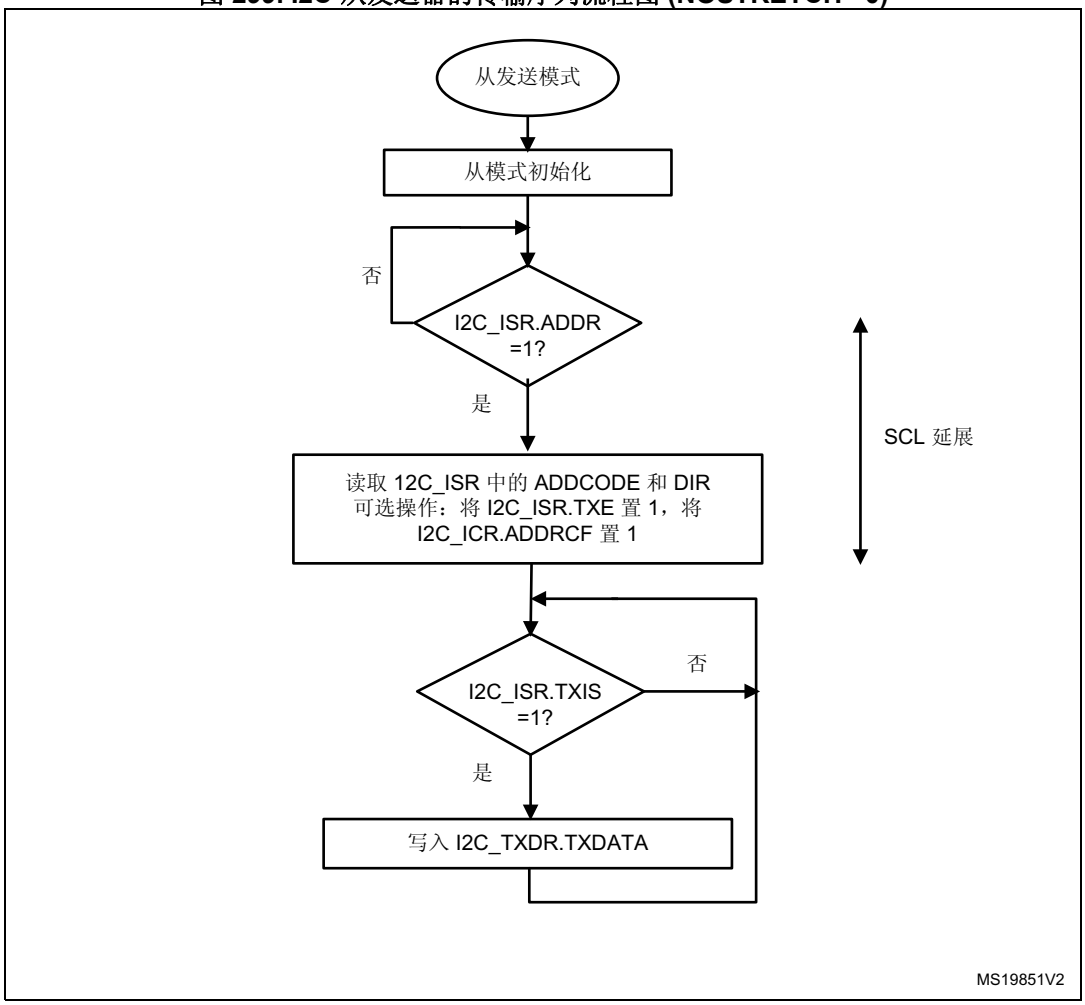


图 300. I2C 从发送器的传输序列流程图 (NOSTRETCH= 1)

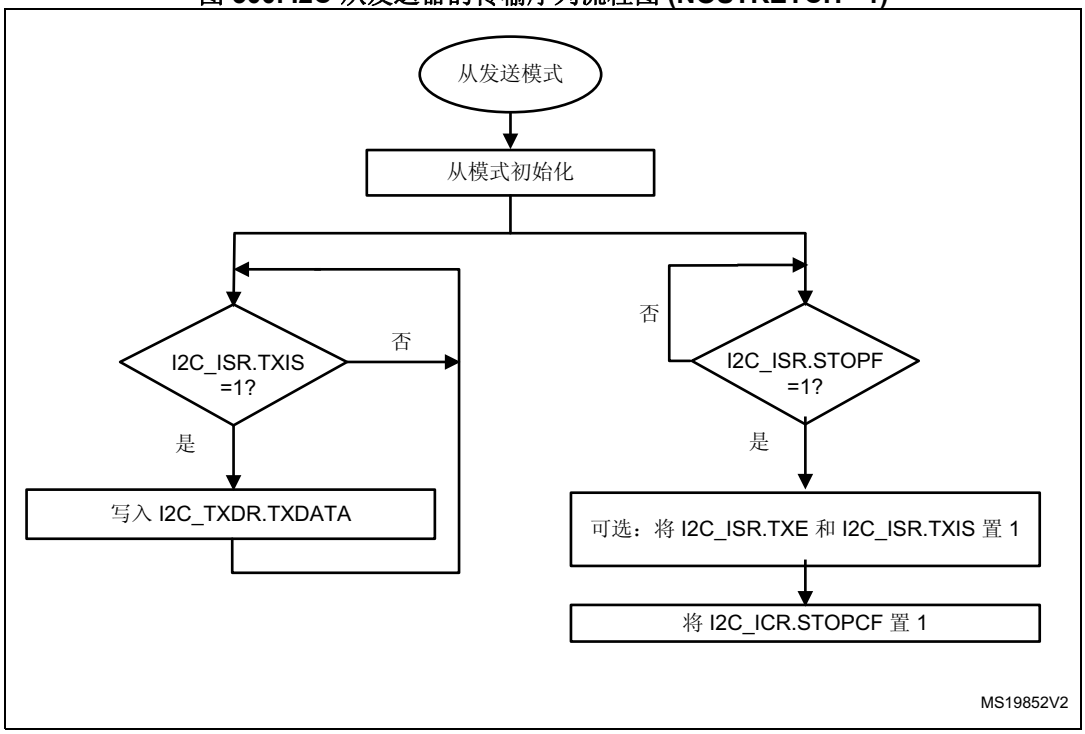
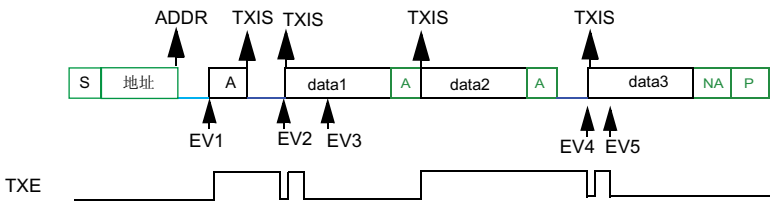


图 301. I2C 从发送器的传输总线图

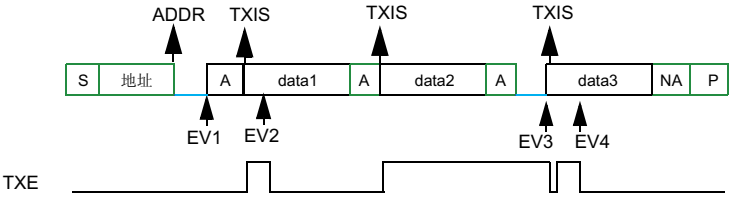
示例：I2C 从器件发送 3 个字节，刷新第 1 个数据，NOSTRETCH=0：



图注：发送 / 接收 / SCL 延长

EV1：ADDR ISR：检查 ADDCODE 和 DIR，将 TXE 置 1，将 ADDRCF 置 1
EV2：TXIS ISR：写入 data1
EV3：TXIS ISR：写入 data2
EV4：TXIS ISR：写入 data3
EV5：TXIS ISR：写入 data4（不发送）

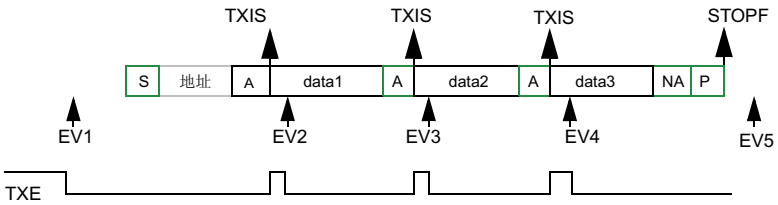
示例：I2C 从器件发送 3 个字节，不刷新第 1 个数据，NOSTRETCH=0：



图注：发送 / 接收 / SCL 延长

EV1：ADDR ISR：检查 ADDCODE 和 DIR，将 ADDRCF 置 1
EV2：TXIS ISR：写入 data2
EV3：TXIS ISR：写入 data3
EV4：TXIS ISR：写入 data4（不发送）

示例：I2C 从器件发送 3 个字节，NOSTRETCH=1：



图注：发送 / 接收 / SCL 延长

EV1：写入 data1
EV2：TXIS ISR：写入 data2
EV3：TXIS ISR：写入 data3
EV4：TXIS ISR：写入 data4（不发送）
EV5：STOPF ISR：（可选操作：将 TXE 和 TXIS 置 1），将 STOPCF 置 1

MS19853V1

从接收器

当 I2C_RXDR 满时，I2C_ISR 中的 RXNE 将置 1，如果 I2C_CR1 中的 RXIE 置 1，还将生成中断。读取 I2C_RXDR 时，将清零 RXNE。

接收到停止条件且 I2C_CR1 寄存器中的 STOPIE 置 1 时，I2C_ISR 中的 STOPF 将置 1 并且会生成中断。

图 302. 从接收器的传输序列流程图 (NOSTRETCH=0)

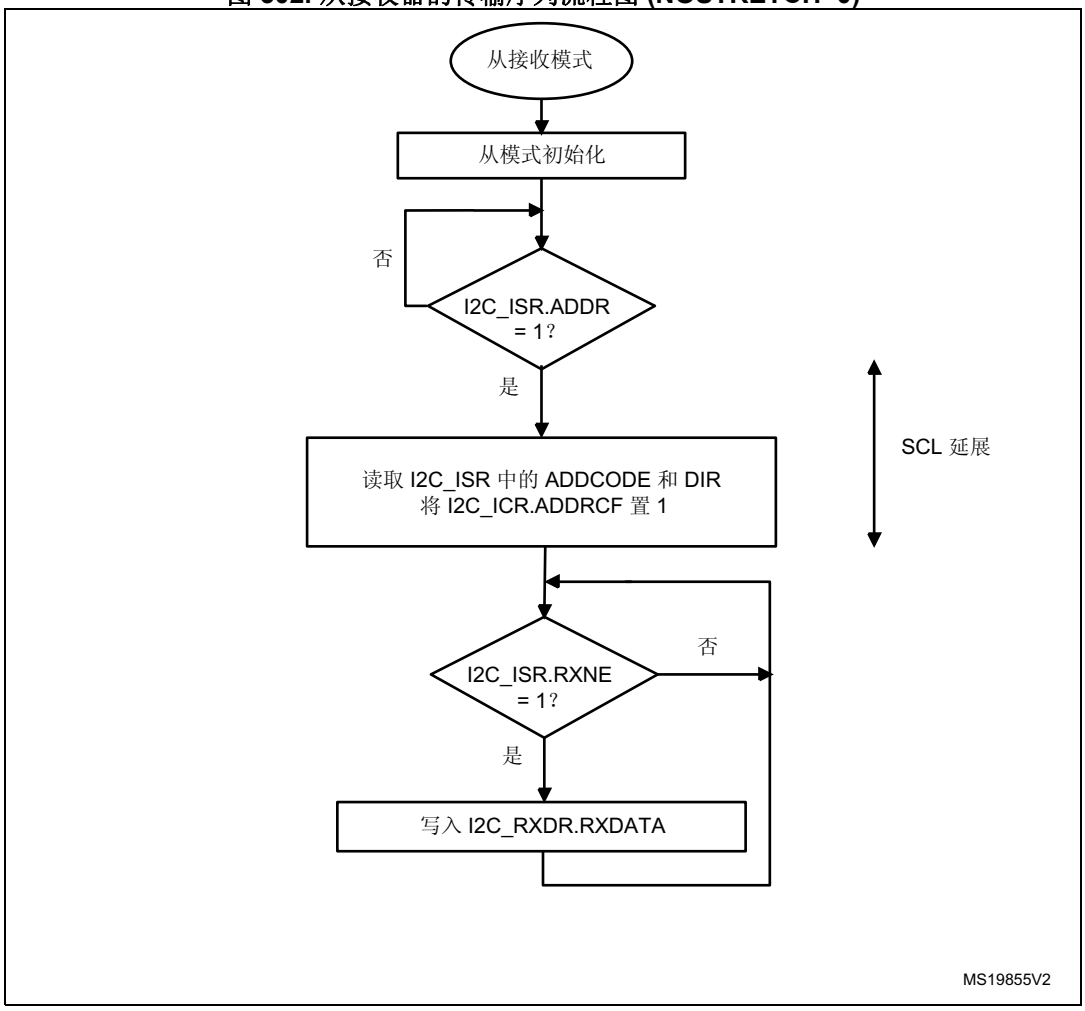


图 303. 从接收器的传输序列流程图 (NOSTRETCH=1)

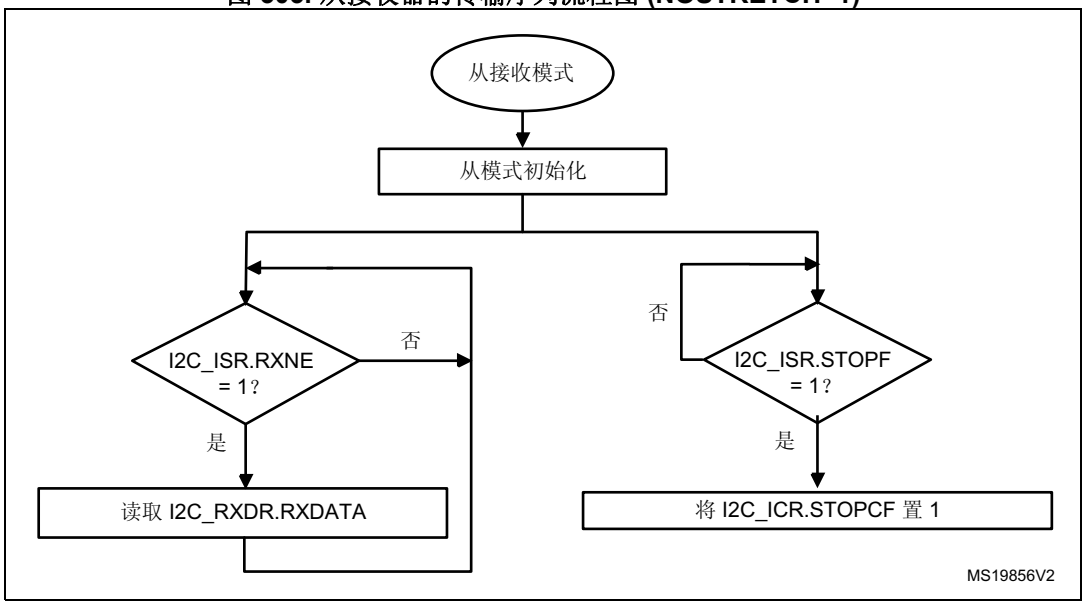
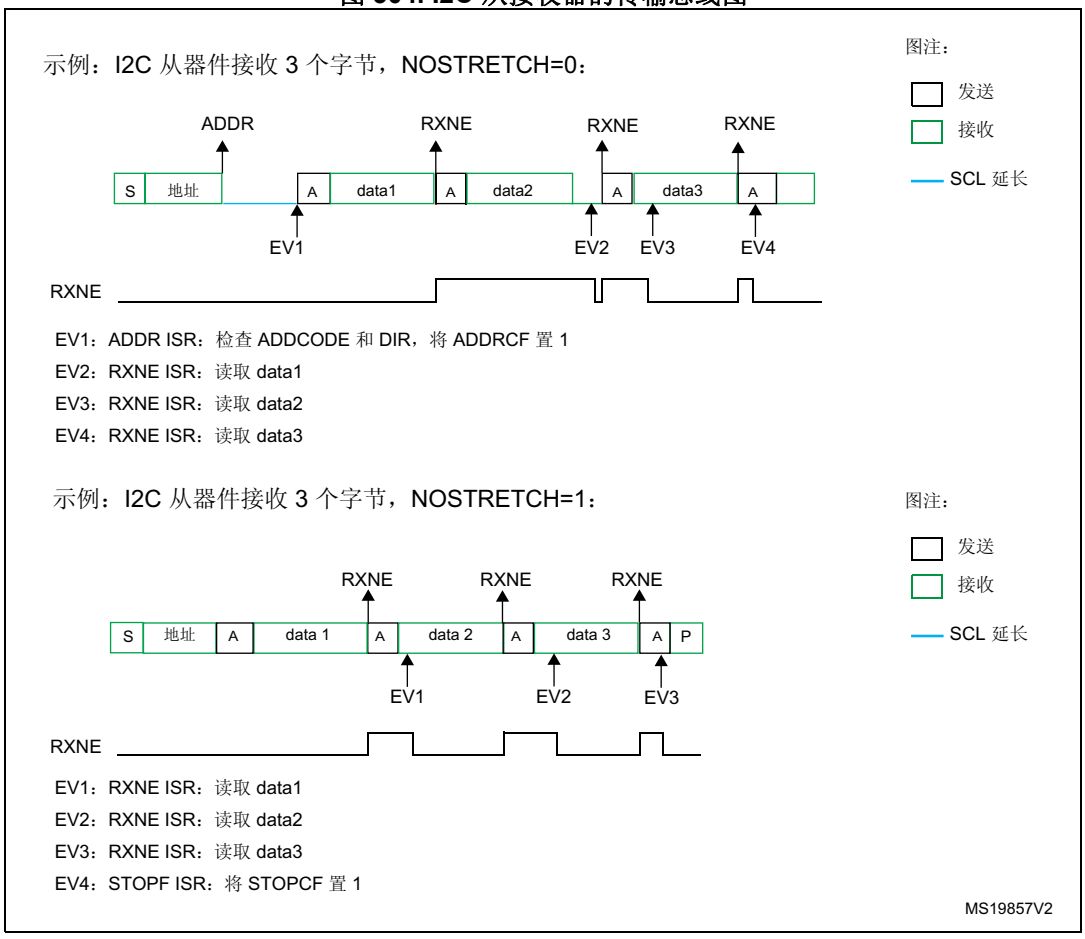


图 304. I2C 从接收器的传输总线图



32.4.9 I2C 主模式

I2C 主模式初始化

使能外设前，必须通过设置 I2C_TIMINGR 寄存器中的 SCLH 和 SCLL 位来配置 I2C 主时钟。

STM32CubeMX 工具进行计算并将 I2C_TIMINGR 内容提供在 I2C 配置窗口中。

为了支持多主环境和从时钟延长，I2C 实现了时钟同步机制。

为了实现时钟同步，需执行以下操作：

- 使用 SCLL 计数器从 SCL 低电平内部检测开始对时钟的低电平进行计数。
- 使用 SCLH 计数器从 SCL 高电平内部检测开始对时钟的高电平进行计数。

I2C 经过 t_{SYNC1} 延时后检测其自身的 SCL 低电平，该延时取决于 SCL 下降沿、SCL 输入噪声滤波器（模拟 + 数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLL 计数器达到 I2C_TIMINGR 寄存器的 SCLL[7:0] 位中编程的值，I2C 便会将 SCL 释放为高电平。

I2C 经过 t_{SYNC2} 延时后检测其自身的 SCL 高电平，该延时取决于 SCL 上升沿、SCL 输入噪声滤波器（模拟 + 数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLH 计数器达到 I2C_TIMINGR 寄存器的 SCLH[7:0] 位中编程的值，I2C 便会使 SCL 变为低电平。

因此，主时钟周期为：

$$t_{SCL} = t_{SYNC1} + t_{SYNC2} + \{[(SCLH+1) + (SCLL+1)] \times (PRESC+1) \times t_{I2CCLK}\}$$

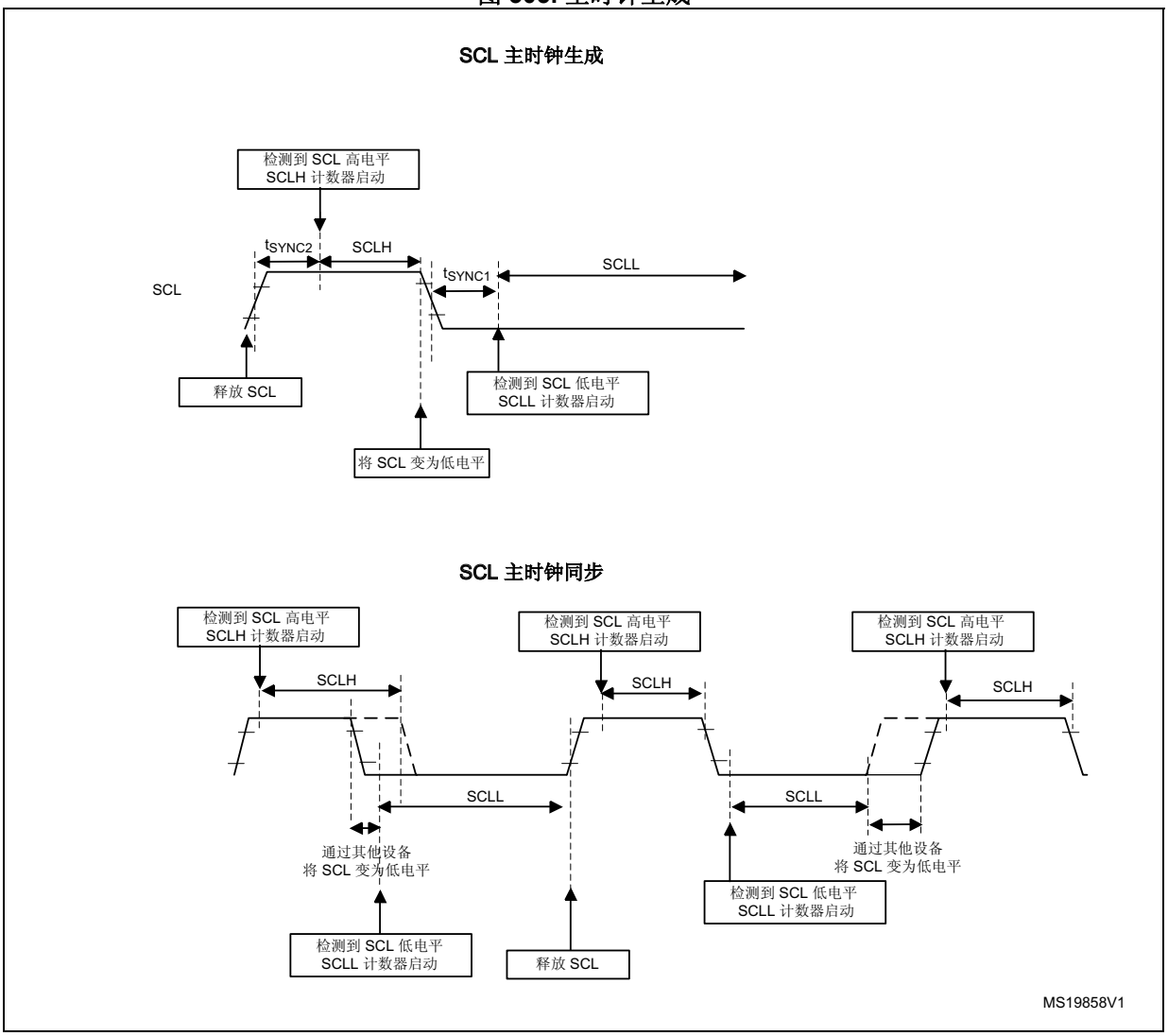
t_{SYNC1} 的持续时间取决于以下参数：

- SCL 下降斜率
- 模拟滤波器（使能时）引入的输入延时
- 数字滤波器（使能时）引入的输入延时：DNF x t_{I2CCLK}
- SCL 与 I2CCLK 时钟建立同步而产生的延时（2 到 3 个 I2CCLK 周期）

t_{SYNC2} 的持续时间取决于以下参数：

- SCL 上升斜率
- 模拟滤波器（使能时）引入的输入延时
- 数字滤波器（使能时）引入的输入延时：DNF x t_{I2CCLK}
- SCL 与 I2CCLK 时钟建立同步而产生的延时（2 到 3 个 I2CCLK 周期）

图 305. 主时钟生成



注意： 为了符合 I^2C 或 SMBus 规范，主时钟必须遵循下表中给出的时序：

表 181. I²C-SMBUS 规范时钟时序

符号	参数	标准模式 (Sm)		快速模式 (Fm)		超快速模式 (Fm+)		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
f_{SCL}	SCL 时钟频率	-	100	-	400	-	1000	-	100	kHz
$t_{HD:STA}$	（重复）起始条件的保持时间	4.0	-	0.6	-	0.26	-	4.0	-	µs
$t_{SU:STA}$	重复起始条件的建立时间	4.7	-	0.6	-	0.26	-	4.7	-	µs
$t_{SU:STO}$	停止条件的建立时间	4.0	-	0.6	-	0.26	-	4.0	-	µs
t_{BUF}	停止条件和起始条件之间的总线空闲时间	4.7	-	1.3	-	0.5	-	4.7	-	µs
t_{LOW}	SCL 时钟的低电平周期	4.7	-	1.3	-	0.5	-	4.7	-	µs
t_{HIGH}	SCL 时钟的高电平周期	4.0	-	0.6	-	0.26	-	4.0	50	µs
t_r	SDA 和 SCL 信号的上升时间	-	1000	-	300	-	120	-	1000	ns
t_f	SDA 和 SCL 信号的下降时间	-	300	-	300	-	120	-	300	ns

注： *SCLL 还用于生成 t_{BUF} 和 $t_{SU:STA}$ 时序。*

SCLH 还用于生成 $t_{HD:STA}$ 和 $t_{SU:STO}$ 时序。

有关 I2C_TIMINGR 设置与 I2CCLK 频率的示例，请参见*第 32.4.10 节：I2C_TIMINGR 寄存器配置示例*。

主模式通信初始化（地址阶段）

要发起通信，用户必须在 I2C_CR2 寄存器中为寻址的从器件编程以下参数：

- 寻址模式（7 位或 10 位）：ADD10
- 待发送的从地址：SADD[9:0]
- 传输方向：RD_WRN
- 读取 10 位地址时：HEAD10R 位。必须对 HEAD10R 进行相应配置，以指示传输方向变化时必须发送完整的地址序列，还是只发送地址头。
- 待传输的字节数：NBYTES[7:0]。如果字节数等于或大于 255，则初始化时必须将 NBYTES[7:0] 填充为 0xFF。

然后，用户必须将 I2C_CR2 寄存器中的 START 位置 1。START 位置 1 时，不允许更改上述所有位。

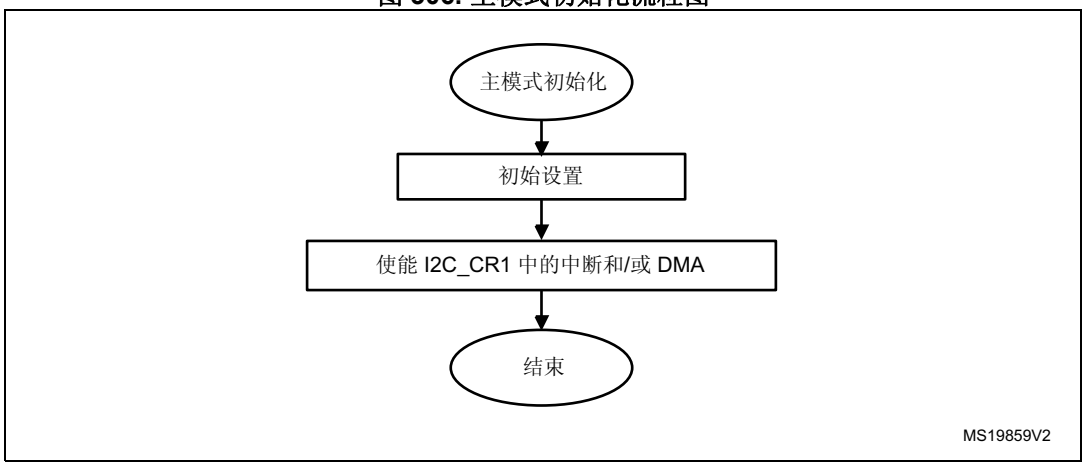
之后，当主器件检测到总线空闲 (BUSY = 0) 时，它会在经过 t_{BUF} 的延时后自动发送起始位，随后发出从器件地址。

仲裁丢失时，主器件将自动切换回从模式，如果作为从器件被寻址，还可对其自身地址进行应答。

注： *无论接收到的应答值为何，只要已在总线上发送从地址，START 位便会由硬件复位。如果仲裁丢失，START 位也会由硬件复位。*
在 10 位寻址模式下，如果从器件不对从地址的前 7 位进行应答，则主器件将自动重新启动从地址发送，直至接收到 ACK。在这种情况下，如果从从器件接收到 NACK，则必须将 ADDRCF 置 1，以停止发送从地址。
如果当 START 位置 1 时，I2C 作为从器件 (ADDR=1) 被寻址，则 I2C 将切换为从模式，START 位将在 ADDRCF 位置 1 时清零。

注： *该步骤同样适用于重复起始位。在这种情况下，BUSY=1。*

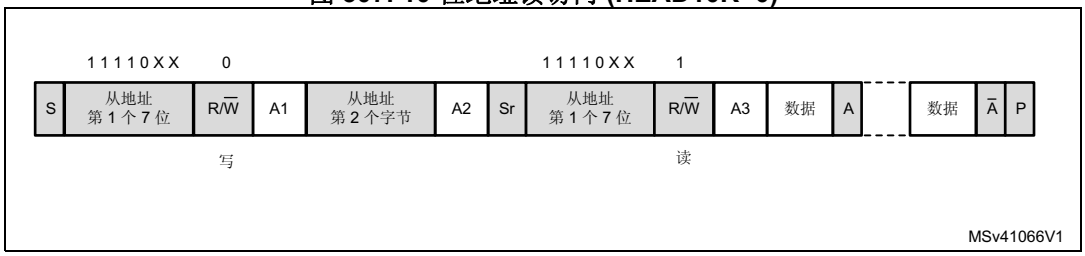
图 306. 主模式初始化流程图



主接收器寻址 10 位地址从器件的初始化过程

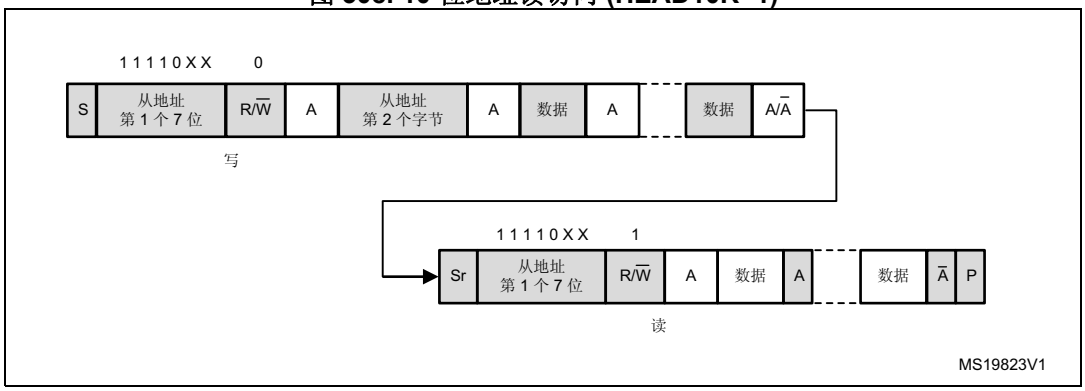
- 如果从地址采用 10 位格式，用户可选择将 I2C_CR2 寄存器中的 HEAD10R 位清零来发送完整的读序列。在这种情况下，主器件会在 START 位置 1 后自动发送以下完整序列：（重复）起始位 + 带写方向的从器件 10 位地址头字节 + 从器件地址第 2 个字节 + 重复起始位 + 带读方向的从器件 10 位地址头字节

图 307. 10 位地址读访问 (HEAD10R=0)



- 如果主器件对 10 位地址从器件进行寻址、向该从器件发送数据、然后再从该从器件读取数据，则必须首先完成主器件发送过程。然后，重复起始位置 1，10 位从地址配置为 HEAD10R=1。在这种情况下，主器件发送以下序列：重复起始位 + 从地址 10 位头读取

图 308. 10 位地址读访问 (HEAD10R=1)



主发送器

写传输时，在发送完每个字节（即第 9 个 SCL 脉冲（接收到 ACK 时））后，TXIS 标志将置 1。

如果 I2C_CR1 寄存器中的 TXIE 位置 1，TXIS 事件将生成中断。当 I2C_TXDR 寄存器中写入待发送的下一个数据字节时，该标志将被清零。

传输期间的 TXIS 事件的数量对应于 NBYTES[7:0] 中编程的值。如果待发送的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES 数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延展，直到 NBYTES[7:0] 被写入非零值。

收到 NACK 时，TXIS 标志不会置 1。

- 当 RELOAD=0 且 NBYTES 数据传输完成时：
  - 在自动结束模式 (AUTOEND=1) 下，将自动发送停止位。
  - 在软件结束模式 (AUTOEND=0) 下，TC 标志将置 1 且 SCL 线的低电平将被延展，以便执行以下软件操作：

    可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从地址和待传输字节数来请求发送重复起始位。将 START 位置 1 会将 TC 标志清零，并在总线上发送起始位。

    可通过将 I2C_CR2 寄存器中的 STOP 位置 1 来请求停止位。将 STOP 位置 1 会将 TC 标志清零，并在总线上发送停止位。
- 如果接收到 NACK：TXIS 标志不会置 1，并且接收到 NACK 后会自动发送停止位。I2C_ISR 寄存器中的 NACKF 标志置 1，如果 NACKIE 位置 1，还将生成中断。

图 309. I2C 主发送器的传输序列流程图（N≤255 字节）

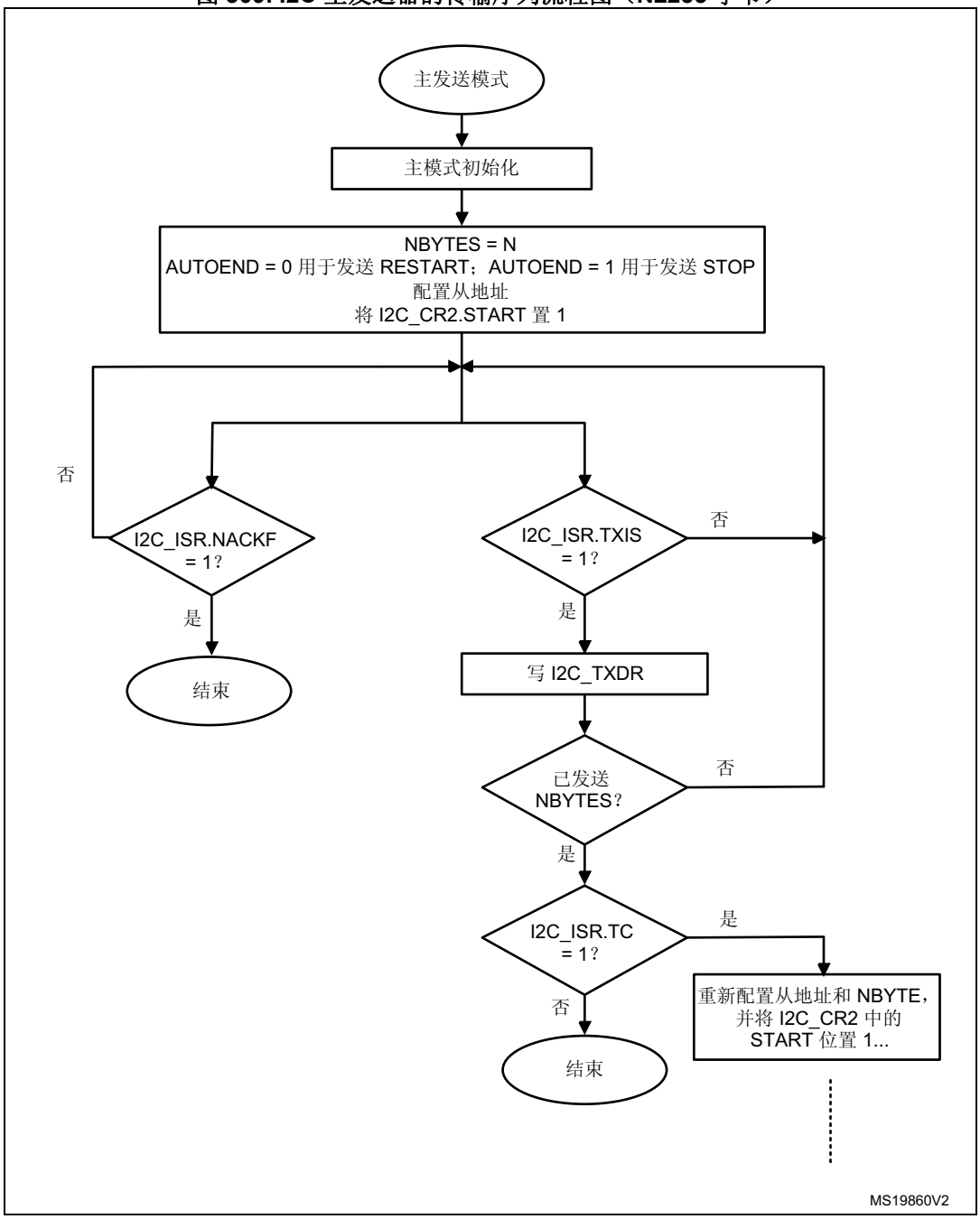


图 310. I2C 主发送器的传输序列流程图（N>255 字节）

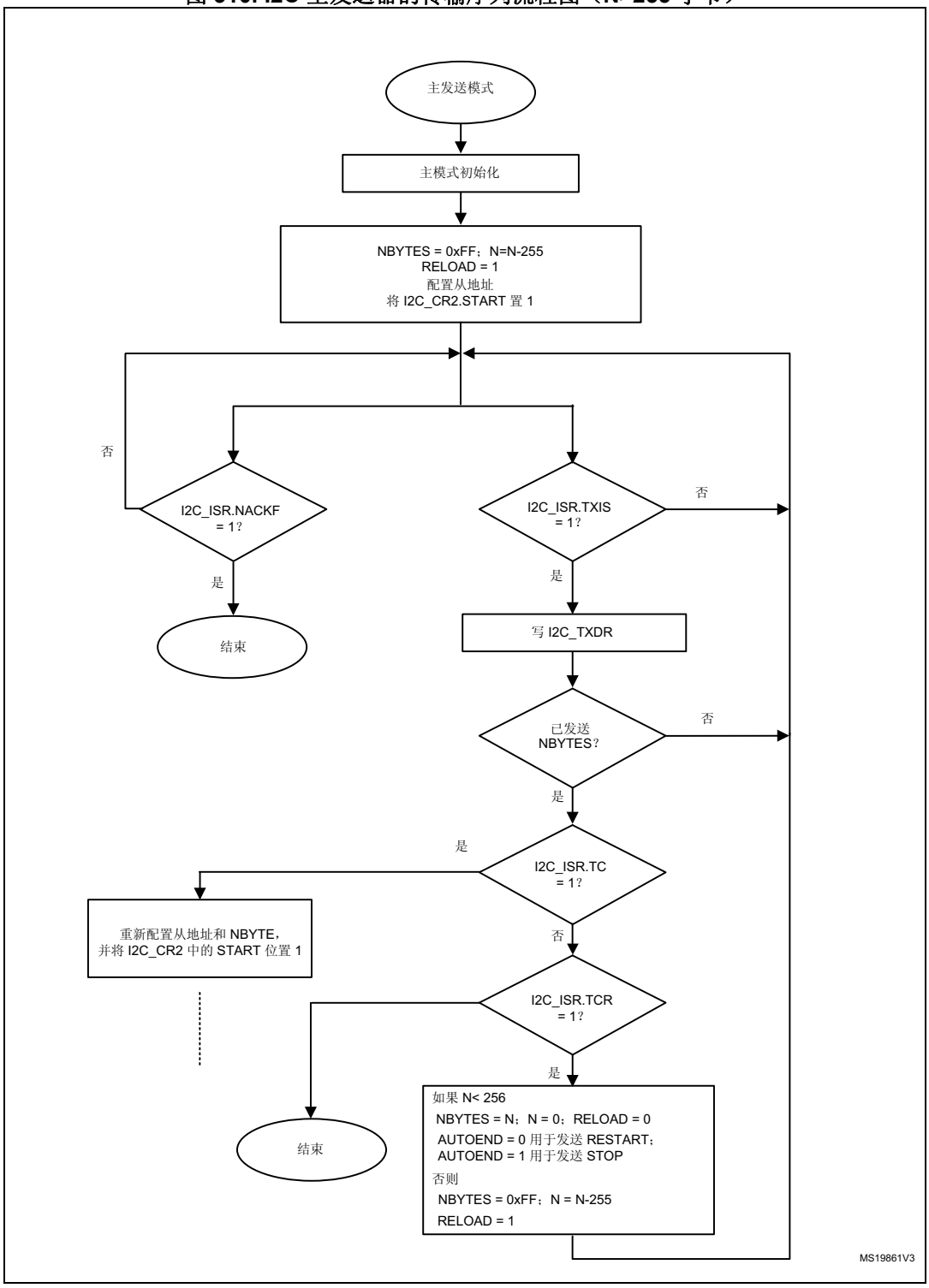
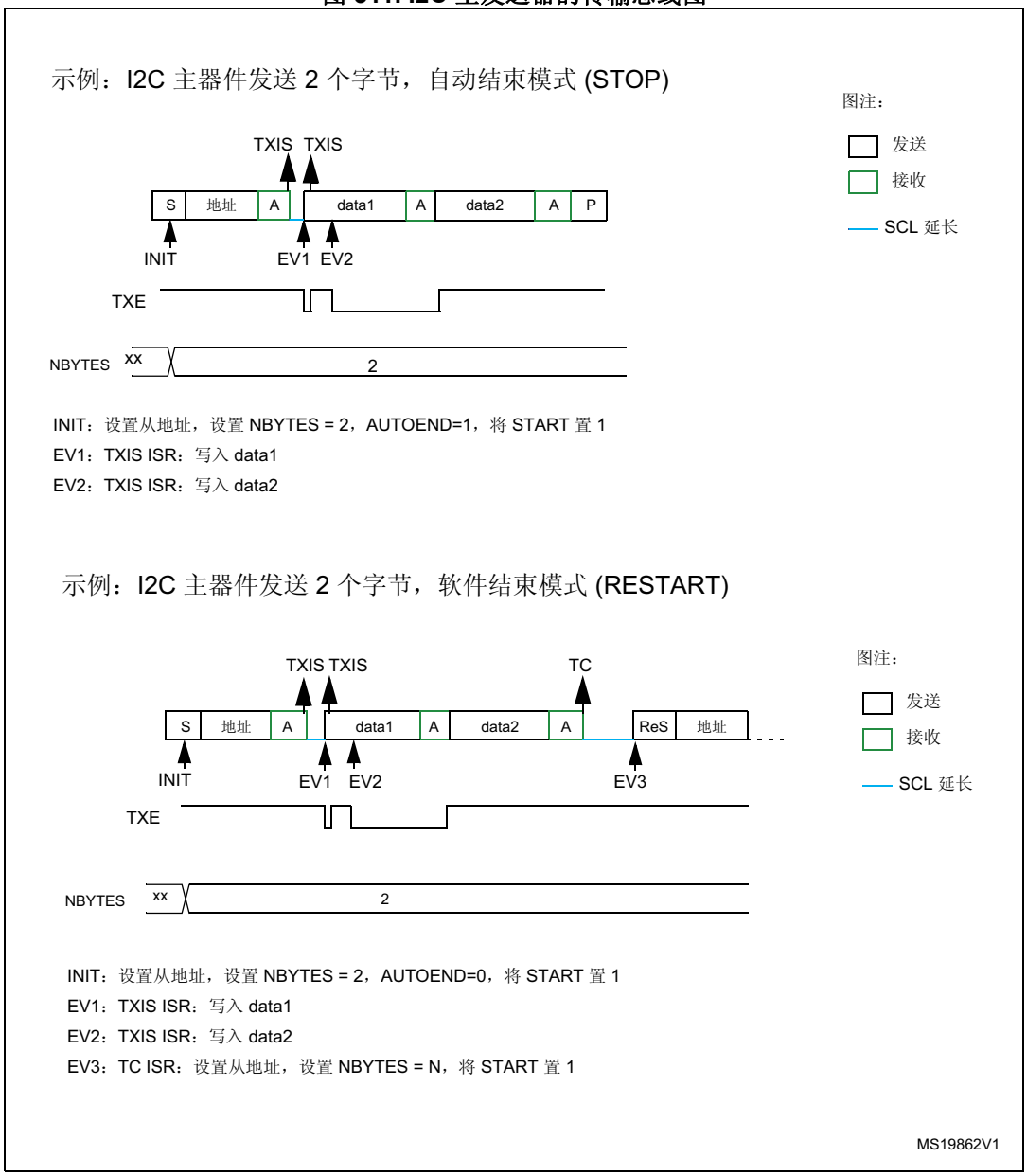


图 311. I2C 主发送器的传输总线图



主接收器

读传输时，在接收到每个字节（即第 8 个 SCL 脉冲）后，RXNE 标志将置 1。如果 I2C_CR1 寄存器中的 RXIE 位置 1，RXNE 事件将生成中断。读取 I2C_RXDR 时，将清零该标志。

如果待接收的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES[7:0] 数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延展，直到 NBYTES[7:0] 被写入非零值。

- 当 RELOAD=0 且 NBYTES[7:0] 数据传输完成时：
  - 在自动结束模式 (AUTOEND=1) 下，接收到最后一个字节后，将自动发送 NACK 和停止位。
  - 在软件结束模式 (AUTOEND=0) 下，接收到最后一个字节后，将自动发送 NACK，TC 标志将置 1 且 SCL 线的低电平将被延展，以便执行以下软件操作：

    可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从地址和待传输字节数来请求发送重复起始位。将 START 位置 1 会将 TC 标志清零，并在总线上发送起始位，后跟从地址。

    可通过将 I2C_CR2 寄存器中的 STOP 位置 1 来请求停止位。将 STOP 位置 1 会将 TC 标志清零，并在总线上发送停止位。

图 312. I2C 主接收器的传输序列流程图（N≤255 字节）

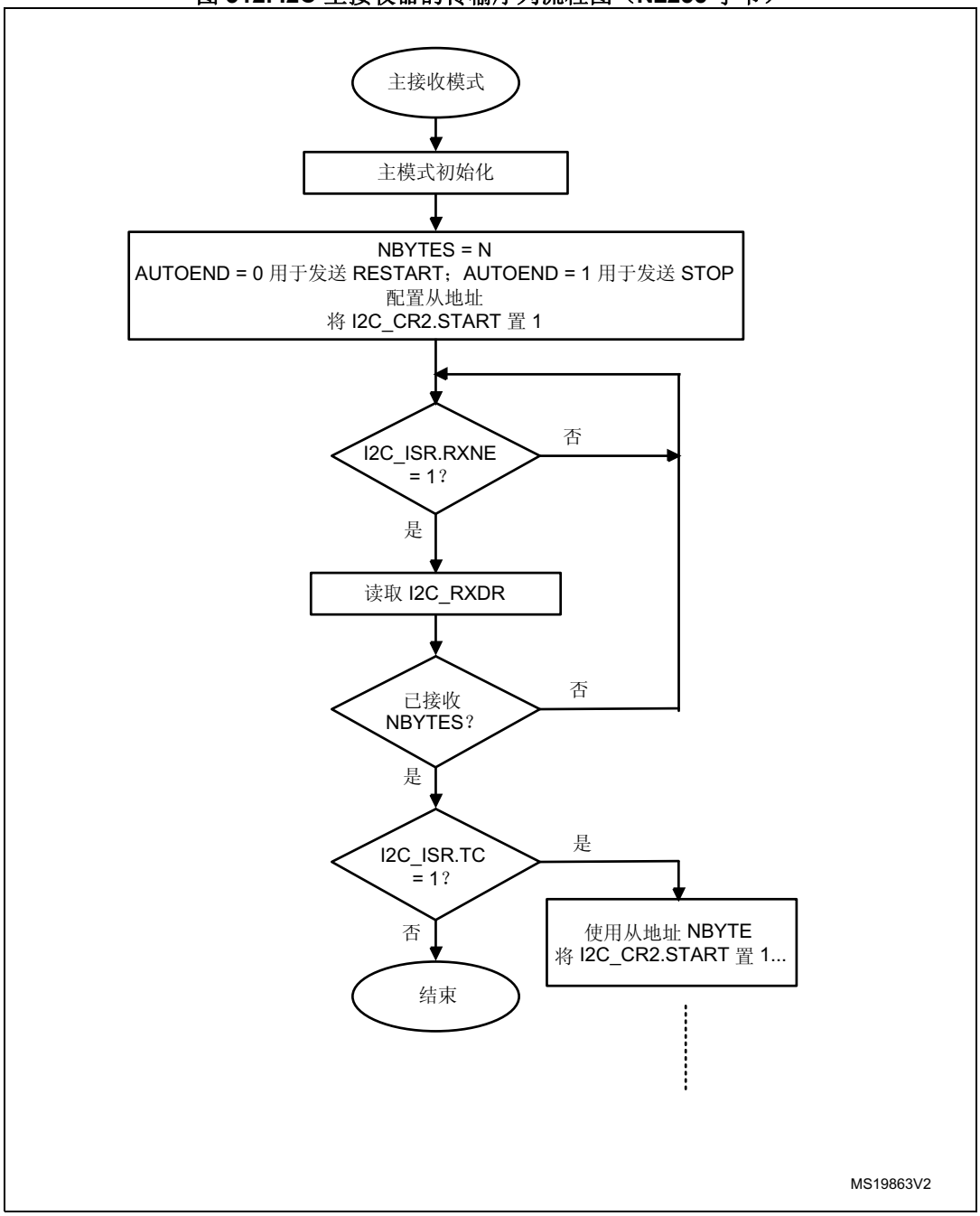


图 313. I2C 主接收器的传输序列流程图（N>255 字节）

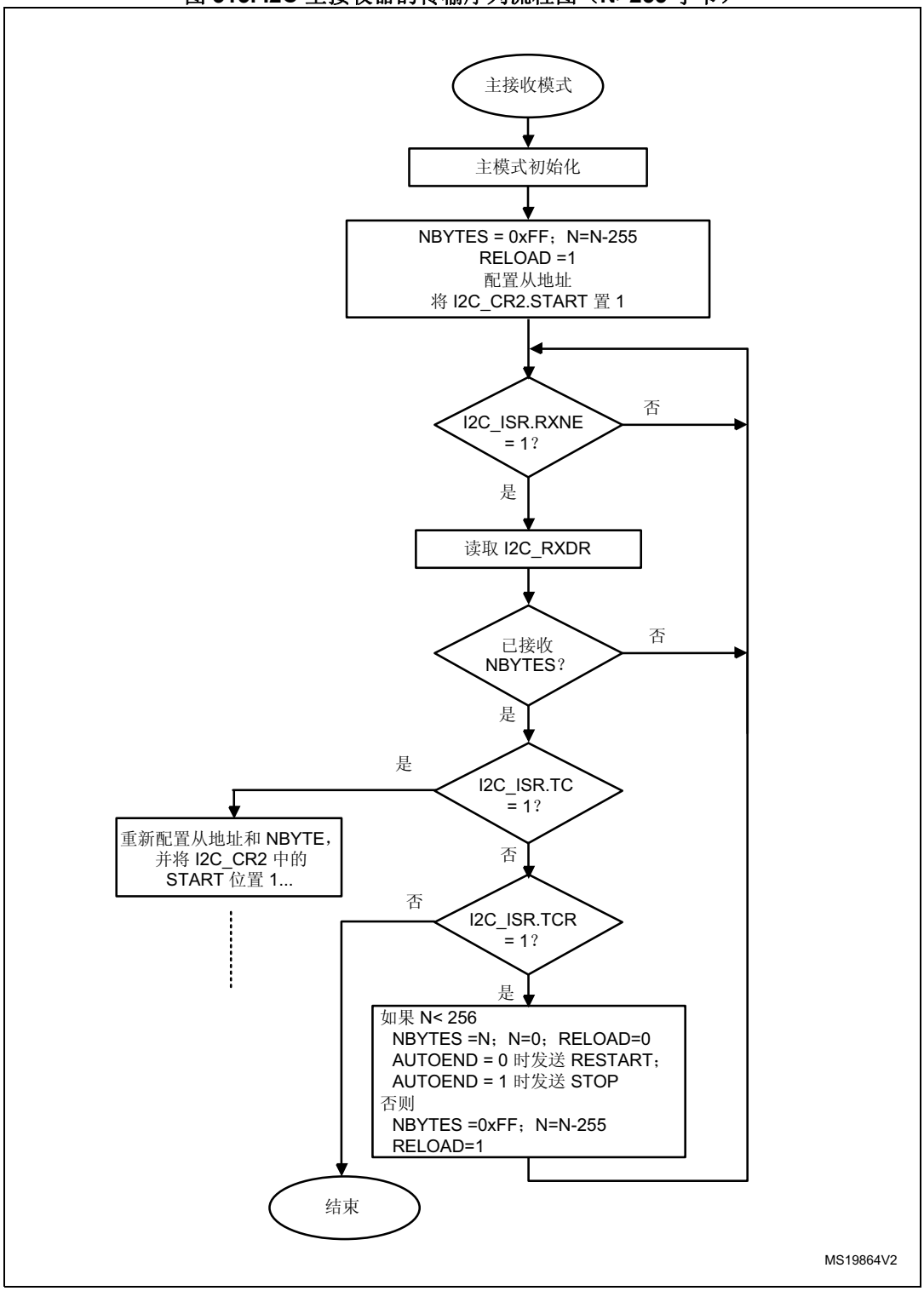
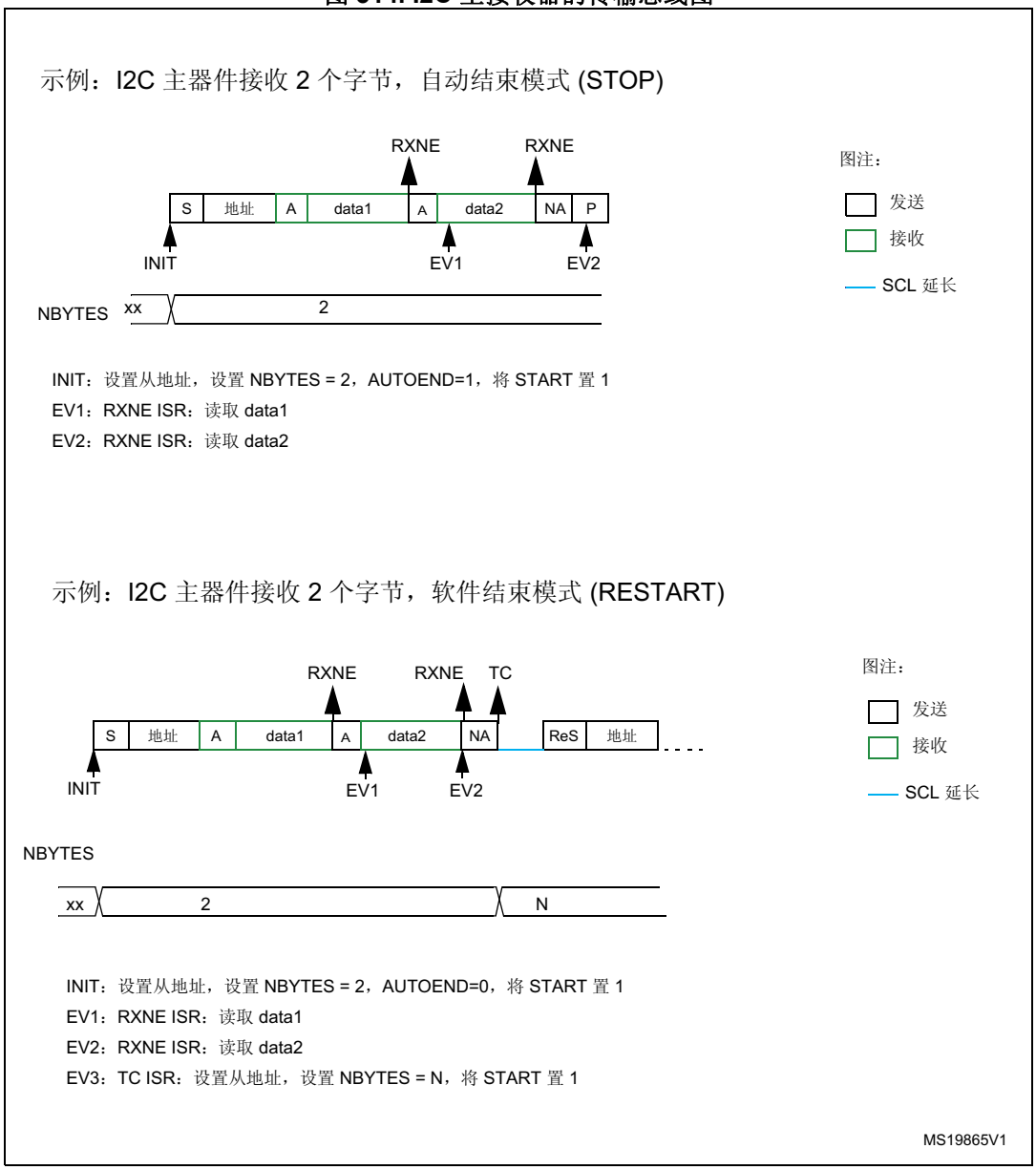


图 314. I2C 主接收器的传输总线图



32.4.10 I2C_TIMINGR 寄存器配置示例

下文各表提供了相应示例，以介绍如何编程 I2C_TIMINGR 才能获得符合 I^2C 规范的时序。要获取更准确的配置值，必须使用 STM32CubeMX 工具（I2C 配置窗口）。

表 182. f_{I2CCLK} = 8 MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	500 kHz
PRESC	1	1	0	0
SCLL	0xC7	0x13	0x9	0x6
t_{SCLL}	200x250 ns = 50 µs	20x250 ns = 5.0 µs	10x125 ns = 1250 ns	7x125 ns = 875 ns
SCLH	0xC3	0xF	0x3	0x3
t_{SCLH}	196x250 ns = 49 µs	16x250 ns = 4.0µs	4x125ns = 500ns	4x125 ns = 500 ns
t_{SCL}(1)	约 100 µs(2)	约 10 µs(2)	约 2500 ns(3)	约 2000 ns(4)
SDADEL	0x2	0x2	0x1	0x0
t_{SDADEL}	2x250 ns = 500 ns	2x250 ns = 500 ns	1x125 ns = 125 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5x250 ns = 1250 ns	5x250 ns = 1250 ns	4x125 ns = 500 ns	2x125 ns = 250 ns

1. 由于 SCL 内部检测存在延时，SCL 周期 t_{SCL} 大于 t_{SCLL} + t_{SCLH}。为 t_{SCL} 提供的值仅用于举例说明。
2. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 500 ns。t_{SYNC1} + t_{SYNC2} = 1000 ns 时的示例。
3. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 500 ns。t_{SYNC1} + t_{SYNC2} = 750 ns 时的示例。
4. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 500 ns。t_{SYNC1} + t_{SYNC2} = 655 ns 时的示例。

表 183. f_{I2CCLK} = 16 MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	3	3	1	0
SCLL	0xC7	0x13	0x9	0x4
t_{SCLL}	200 x 250 ns = 50 µs	20 x 250 ns = 5.0 µs	10 x 125 ns = 1250 ns	5 x 62.5 ns = 312.5 ns
SCLH	0xC3	0xF	0x3	0x2
t_{SCLH}	196 x 250 ns = 49 µs	16 x 250 ns = 4.0 µs	4 x 125 ns = 500 ns	3 x 62.5 ns = 187.5 ns
t_{SCL}(1)	约 100 µs(2)	约 10 µs(2)	约 2500 ns(3)	约 1000 ns(4)
SDADEL	0x2	0x2	0x2	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	2 x 125 ns = 250 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x2
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	3 x 62.5 ns = 187.5 ns

1. 由于 SCL 内部检测存在延时，SCL 周期 t_{SCL} 大于 t_{SCLL} + t_{SCLH}。为 t_{SCL} 提供的值仅用于举例说明。
2. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 250 ns。t_{SYNC1} + tSYNC2 = 1000 ns 时的示例。
3. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 250 ns。t_{SYNC1} + tSYNC2 = 750 ns 时的示例。
4. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 250 ns。t_{SYNC1} + tSYNC2 = 500 ns 时的示例。

32.4.11 SMBus 特性

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 32.3 节：I2C 特性实现*。

简介

系统管理总线 (SMBus) 是一个双线制接口，各器件可通过它在彼此之间或者与系统的其余部分进行通信。它以 I^2C 的工作原理为基础。SMBus 可针对系统和电源管理相关的任务提供控制总线。

该外设与 SMBUS 规范兼容 (http://smbus.org)。

系统管理总线规范涉及三类器件。

- 从器件，用于接收或响应命令。
- 主器件，用于发出命令、生成时钟和中止传输。
- 主机，专用的主器件，可提供连接系统 CPU 的主接口。主机必须具有主-从器件功能，并且必须支持 SMBus 主机通知协议。系统中只允许存在一个主机。

该外设可配置为主器件或从器件，也可配置为主机。

总线协议

任何给定器件都有十一种可用命令协议。器件既可以在这十一种协议中任选其一，也可以使用全部十一种协议进行通信。这十一种协议分别为快速命令、发送字节、接收字节、写入字节、写入字、读取字节、读取字、过程调用、块读取、块写入以及块写入-块读取过程调用。这些协议应通过用户软件实施。

有关这些协议的详细信息，请参见 SMBus 规范 (http://smbus.org)。

地址解析协议 (ARP)

通过为各个从器件动态分配一个新的唯一地址可解决 SMBus 从地址冲突的问题。为了提供一种机制来针对地址分配隔离各个器件，各器件必须具有唯一的器件标识符 (UDID)。该 128 位数字由软件实现。

该外设支持地址解析协议 (ARP)。通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 器件默认地址 (0b1100 001)。ARP 命令应通过用户软件实现。

此外，还将在从模式下执行仲裁以支持 ARP。

有关 SMBus 地址解析协议的详细信息，请参见 SMBus 规范 (http://smbus.org)。

接收的命令和数据应答控制

SMBus 接收器必须能够对接收到的每个命令或数据进行否定应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。更多详细信息，请参见*第 944 页的从器件字节控制模式*。

主机通知协议

该外设通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来支持主机通知协议。在这种情况下，主机将应答 SMBus 主机地址 (0b0001 000)。

使用该协议时，器件用作主器件，而主机用作从器件。

SMBus 报警

器件支持 SMBus ALERT 可选信号。只具备从功能的器件可通过 SMBALERT# 引脚向主机发出信号，指示它想要通信。主机会处理该中断并通过报警响应地址 (0b0001 100) 同时访问所有 SMBALERT# 器件。只有那些将 SMBALERT# 拉到低电平的器件会应答报警响应地址。

如果配置为从器件 (SMBHEN=0)，则通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来将 SMBA 引脚拉为低电平。这同时还会使能报警响应地址。

如果配置为主机 (SMBHEN=1)，则当 SMBA 引脚上检测到下降沿且 ALERTEN=1 时，I2C_ISR 寄存器中的 ALERT 标志置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，将生成中断。当 ALERTEN=0 时，即使外部 SMBA 引脚为低电平，ALERT 线也将被视为高电平。

如果无需 SMBus ALERT 引脚，则当 ALERTEN=0 时，SMBA 引脚可用作标准 GPIO。

数据包错误校验

SMBus 规范中引入了数据包错误校验机制来提高可靠性和通信稳定性。数据包错误校验的实施方式是在每次消息传输结束时附加数据包错误代码 (PEC)。PEC 的计算方式是对所有消息字节（包括地址和读/写位）使用 CRC-8 多项式 $C(x) = x_8 + x^2 + x + 1$。

外设内置了硬件 PEC 计算器，可在接收到的字节与硬件计算的 PEC 不匹配时自动发送否定应答信号。

超时

该外设内置了硬件定时器，以便符合 SMBus 规范中定义的 3 个超时。

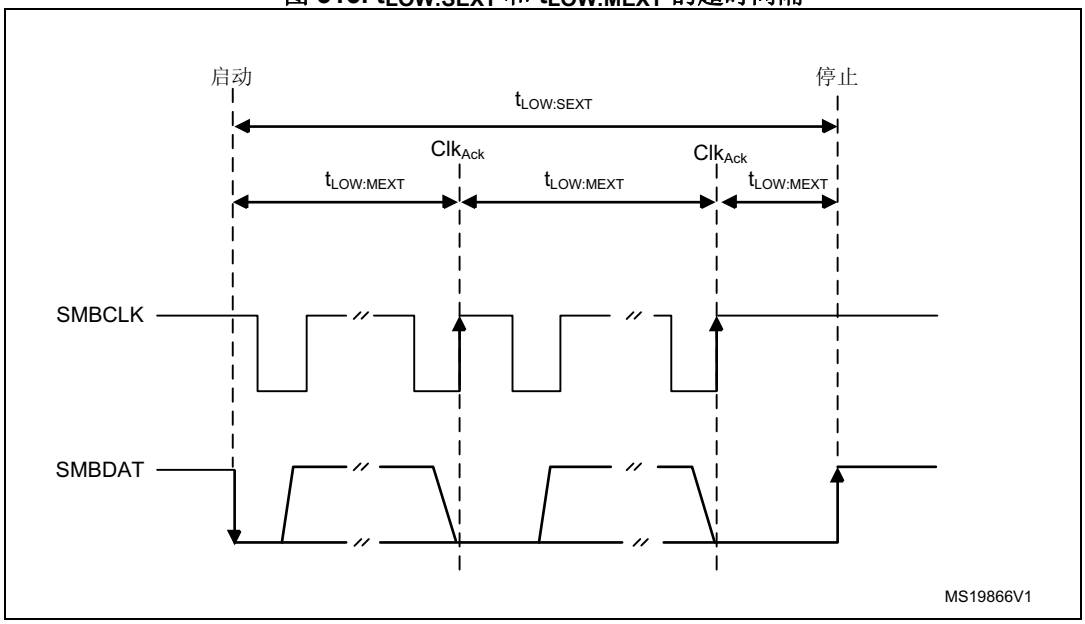
表 184. SMBus 超时规范

符号	参数	限值		单位
		最小值	最大值	
$t_{TIMEOUT}$	检测时钟低电平超时	25	35	ms
$t_{LOW:SEXT}$(1)	累积时钟低电平延长时间（从器件）	-	25	ms
$t_{LOW:MEXT}$(2)	累积时钟低电平延长时间（主器件）	-	10	ms

1. $t_{LOW:SEXT}$ 是一段累积时间，即给定从器件在一条消息的最初起始到停止期间时钟信号可延展的时间。其它从器件或主器件也可能延长时钟，进而导致时钟低电平总延长时间超过 $t_{LOW:SEXT}$。因此，测量该参数时该器件应该是全速主器件寻址的唯一器件。

2. $t_{LOW:MEXT}$ 是一段累积时间，即主器件在消息的每个字节（定义为 START 到 ACK、ACK 到 ACK 或 ACK 到 STOP）内时钟信号可延展的时间。从器件或其它主器件也可能延长时钟，进而导致时钟低电平总时间超过 $t_{LOW:MEXT}$（针对给定字节）。因此，测量该参数时该全速主器件只寻址一个从器件。

图 315. $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的超时间隔



总线空闲检测

如果主器件检测到时钟和数据信号的高电平时间已达 t_{IDLE}（超过 $t_{HIGH,MAX}$），则认为总线空闲（请参见*表 179：I2C-SMBUS 规范数据建立和保持时间*）。

该时序参数已考虑如下情况：主器件已动态添加至总线，但可能尚未检测到 SMBCLK 或 SMBDAT 线上的状态转换。在这种情况下，主器件必须等待足够长的时间，以确定当前未进行传输。外设支持硬件总线空闲检测。

32.4.12 SMBus 初始化

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 32.3 节：I2C 特性实现*。

除了 I2C 初始化之外，还必须进行一些其他的特定初始化，以便执行 SMBus 通信：

接收的命令和数据应答控制（从模式）

SMBus 接收器必须能够对接收到的每个命令或数据进行否定应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。更多详细信息，请参见*第 944 页的从器件字节控制模式*。

特定地址（从模式）

必要时必须使能特定的 SMBus 地址。更多详细信息，请参见*第 966 页的总线空闲检测*。

- 通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 器件默认地址 (0b1100 001)。
- 通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来使能 SMBus 主机地址 (0b0001 000)。
- 通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来使能报警响应地址 (0b0001100)。

数据包错误校验

通过将 I2C_CR1 寄存器中的 PECEN 位置 1 来使能 PEC 的计算。然后，借助硬件字节计数器（I2C_CR2 寄存器中的 NBYTES[7:0]）来管理 PEC 传输。使能 I2C 之前，必须配置 PECEN 位。

PEC 传输由硬件字节计数器来管理，因此在从模式下连接 SMBus 时必须将 SBC 位置 1。当 PECBYTE 位置 1 且 RELOAD 位清零时，传输完 NBYTES-1 字节的数据后会传输 PEC。如果 RELOAD 置 1，PECBYTE 将不起作用。

注意： 使能 I2C 时，不允许更改 PECEN 配置。

表 185. 带 PEC 的 SMBUS 配置

模式	SBC 位	RELOAD 位	AUTOEND 位	PECBYTE 位
主 Tx/Rx NBYTES + PEC+ STOP	x	0	1	1
主 Tx/Rx NBYTES + PEC + ReSTART	x	0	0	1
从 Tx/Rx + PEC	1	0	x	1

超时检测

将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 和 TEXTEN 位置 1 来使能超时检测。定时器必须按如下方式编程：即在 SMBus 规范规定的时间最大值之前检测出超时情况。

- $t_{TIMEOUT}$ 检查
  要使能 $t_{TIMEOUT}$ 检查，必须将 12 位 TIMEOUTA[11:0] 位编程为定时器重载值，以检查 $t_{TIMEOUT}$ 参数。必须将 TIDLE 位配置为“0”，以检测 SCL 低电平超时。
  然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。
  如果 SCL 的低电平持续时间超过 (TIMEOUTA+1) x 2048 x t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。
  请参见*表 186：不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 $t_{TIMEOUT}$ = 25 ms）*。

注意： TIMEOUTEN 位置 1 时，不允许更改 TIMEOUTA[11:0] 位和 TIDLE 位的配置。

- $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 检查
  必须根据外设配置为主器件还是从器件来配置 TIMEOUTB 定时器，以便为从器件校验 $t_{LOW:SEXT}$，为主器件校验 $t_{LOW:MEXT}$。由于标准只规定了最大值，用户可以为这两个参数选择相同的值。
  然后，通过将 I2C_TIMEOUTR 寄存器中的 TEXTEN 位置 1 来使能定时器。
  如果 SMBus 外设延展 SCL 的累积时间超过 (TIMEOUTB+1) x 2048 x t_{I2CCLK}，并且达到*第 966 页的总线空闲检测*一节给出的超时间隔，则 I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。
  请参见*表 187：不同 I2CCLK 频率下的 TIMEOUTB 设置示例*。

注意： TEXTEN 位置 1 时，不允许更改 TIMEOUTB 配置。

总线空闲检测

要使能 t_{IDLE} 检查，必须将 12 位 TIMEOUTA[11:0] 字段编程为定时器重载值，以获取 t_{IDLE} 参数。必须将 TIDLE 位配置为“1”，以检测 SCL 和 SDA 高电平超时。

然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。

如果 SCL 和 SDA 线的高电平持续时间超过 (TIMEOUTA+1) x 4 x t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

请参见*表 188：不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 t_{IDLE} = 50 µs）*。

注意： TIMEOUTEN 置 1 时，不允许更改 TIMEOUTA 和 TIDLE 配置。

32.4.13 SMBus：I2C_TIMEOUTR 寄存器配置示例

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 32.3 节：I2C 特性实现*。

- 将 $t_{TIMEOUT}$ 的最大持续时间配置为 25 ms：

表 186. 不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 $t_{TIMEOUT}$ = 25 ms）

f_{I2CCLK}	TIMEOUTA[11:0] 位	TIDLE 位	TIMEOUTEN 位	$t_{TIMEOUT}$
8 MHz	0x61	0	1	98 x 2048 x 125 ns = 25 ms
16 MHz	0xC3	0	1	196 x 2048 x 62.5 ns = 25 ms

- 将 $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的最大持续时间配置为 8 ms：

表 187. 不同 I2CCLK 频率下的 TIMEOUTB 设置示例

f_{I2CCLK}	TIMEOUTB[11:0] 位	TEXTEN位	$t_{LOW:EXT}$
8 MHz	0x1F	1	32 x 2048 x 125 ns = 8 ms
16 MHz	0x3F	1	64 x 2048 x 62.5 ns = 8 ms

- 将 t_{IDLE} 的最大持续时间配置为 50 µs

表 188. 不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 t_{IDLE} = 50 µs）

f_{I2CCLK}	TIMEOUTA[11:0] 位	TIDLE 位	TIMEOUTEN 位	t_{TIDLE}
8 MHz	0x63	1	1	100 x 4 x 125 ns = 50 µs
16 MHz	0xC7	1	1	200 x 4 x 62.5 ns = 50 µs

32.4.14 SMBus 从模式

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 32.3 节：I2C 特性实现*。

除了 I2C 从模式传输管理（请参见*第 32.4.8 节：I2C 从模式*）之外，还提供了一些额外的软件流程图来支持 SMBus。

SMBus 从发送器

在 SMBus 模式下作为从发送器时，必须将 SBC 编程为“1”，以便在完成已编程数据字节数的传输后进行 PEC 传输。当 PECBYTE 位置 1 时，NBYTES[7:0] 中编程的字节数包含 PEC 传输。在这种情况下，总 TXIS 中断数为 NBYTES-1，如果主器件在完成 NBYTES-1 字节的数据传输后请求传输额外的字节，则将自动发送 I2C_PECR 寄存器的内容。

注意： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 316. SMBus 从发送器的传输序列流程图（N 字节 + PEC）

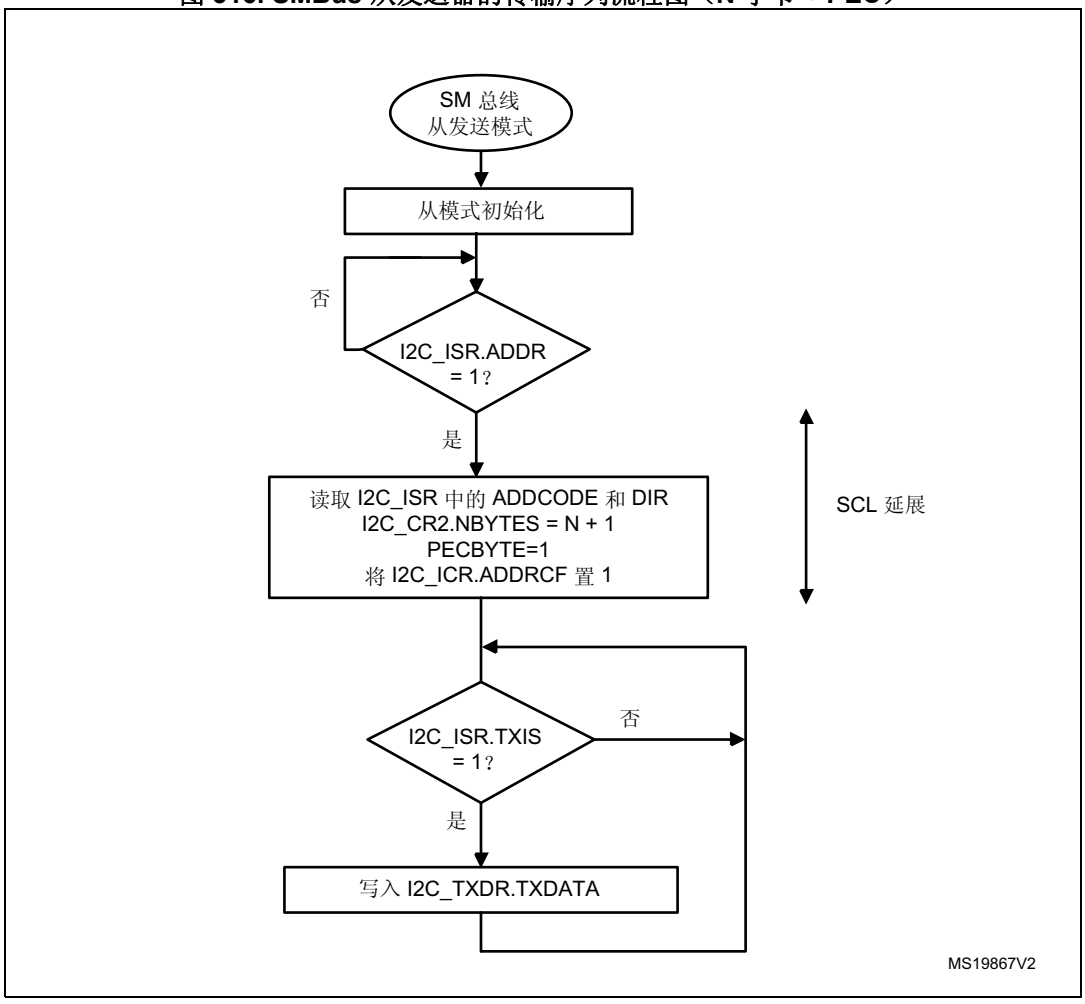
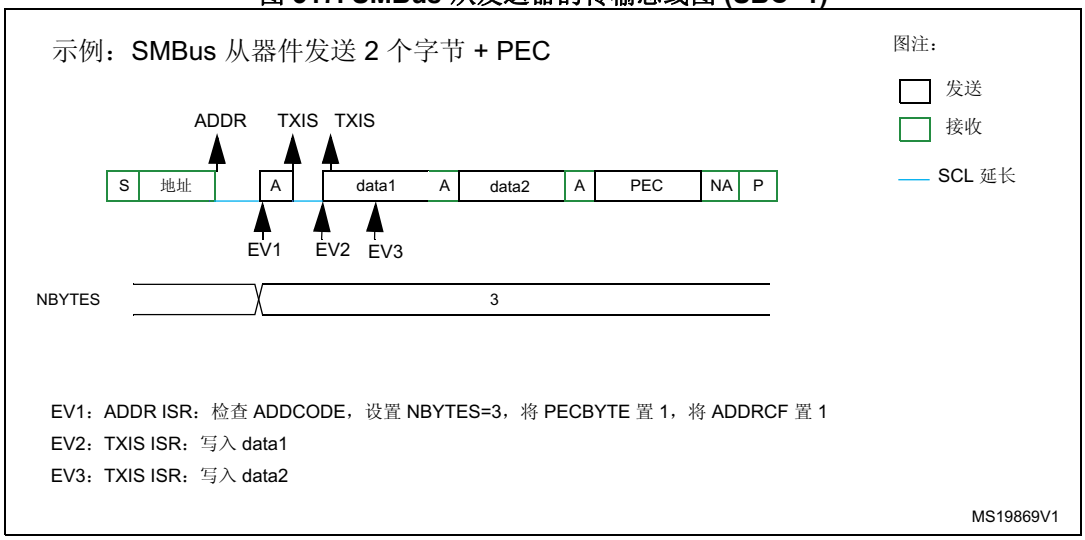


图 317. SMBus 从发送器的传输总线图 (SBC=1)



SMBus 从接收器

在 SMBus 模式下使用 I2C 时，必须将 SBC 编程为“1”，以便在完成已编程数据字节数的传输后进行 PEC 校验。要对每个字节进行 ACK 控制，必须选择重载模式 (RELOAD=1)。更多详细信息，请参见*第 944 页的从器件字节控制模式*。

要校验 PEC 字节，必须将 RELOAD 位清零并将 PECBYTE 位置 1。在这种情况下，当接收到 NBYTES-1 字节的数据后，接收的下一个字节将与内部 I2C_PECR 寄存器的内容作比较。如果比较不匹配，则将自动生成 NACK 信号；如果比较匹配，则将自动生成 ACK 信号，而与 ACK 位的值无关。PEC 字节一经接收，便会像任何其他数据一样复制到 I2C_RXDR 寄存器中，并且 RXNE 标志将置 1。

当 PEC 不匹配时，PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

如果无需 ACK 软件控制，用户可编程 PECBYTE=1，在同一写操作下，将 NBYTES 编程为连续接收的字节数。接收到 NBYTES-1 字节的数据后，会将接收的下一个字节视为 PEC 进行校验。

注意： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 318. SMBus 从接收器的传输序列流程图（N 字节 + PEC）

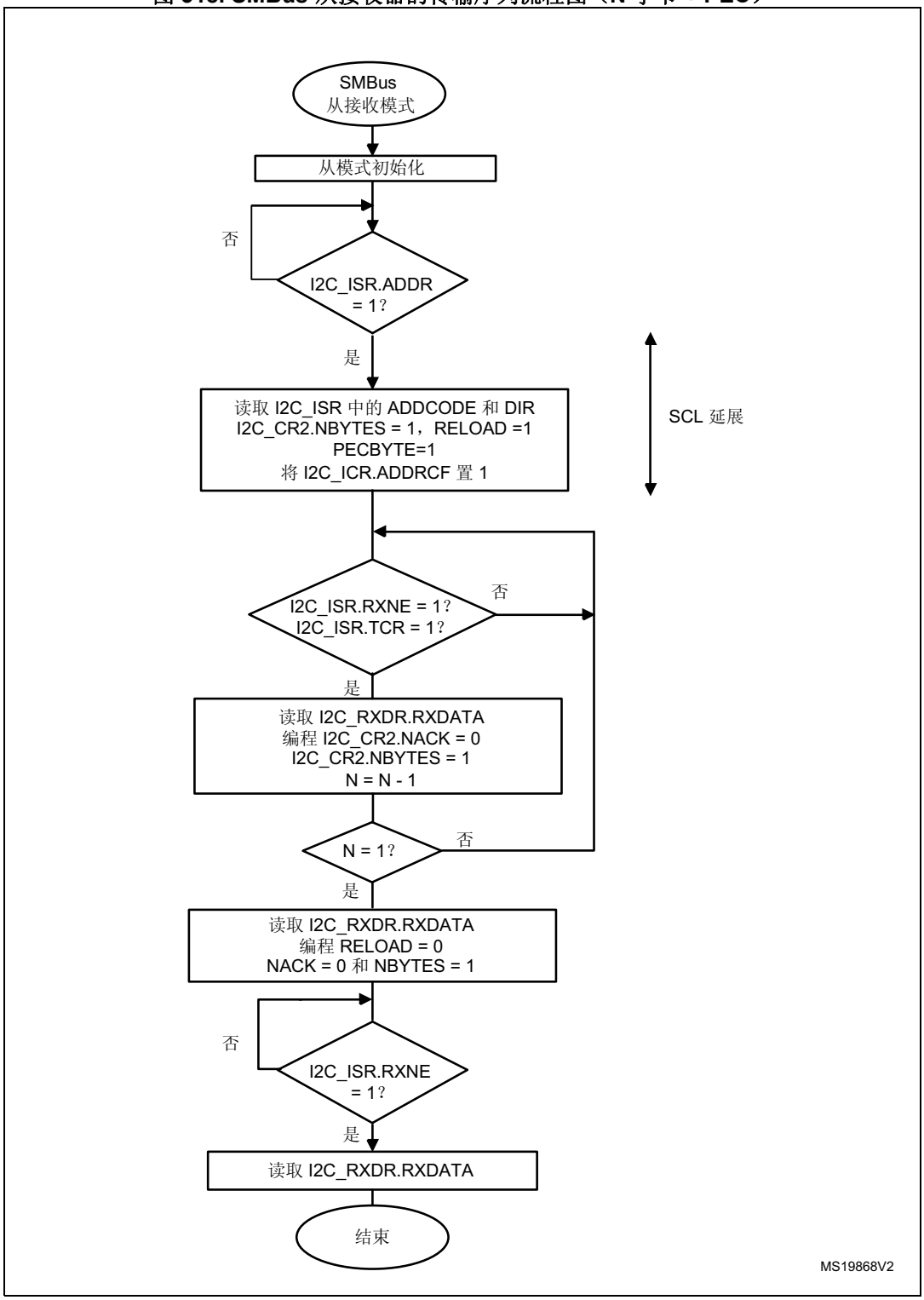
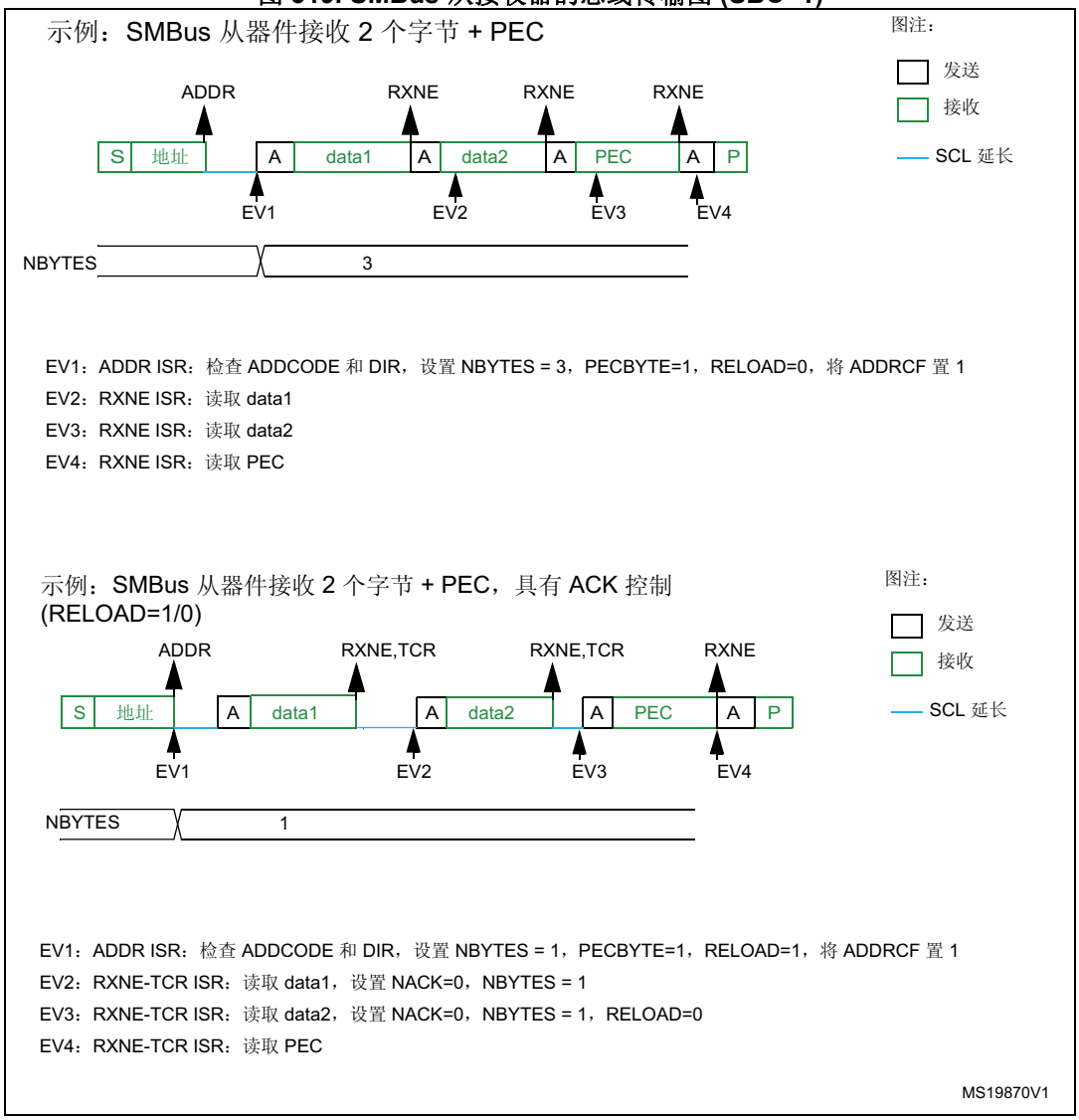


图 319. SMBus 从接收器的总线传输图 (SBC=1)



仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 32.3 节：I2C 特性实现*。

除了 I2C 主模式传输管理（请参见*第 32.4.9 节：I2C 主模式*）之外，还提供了一些额外的软件流程图来支持 SMBus。

SMBus 主发送器

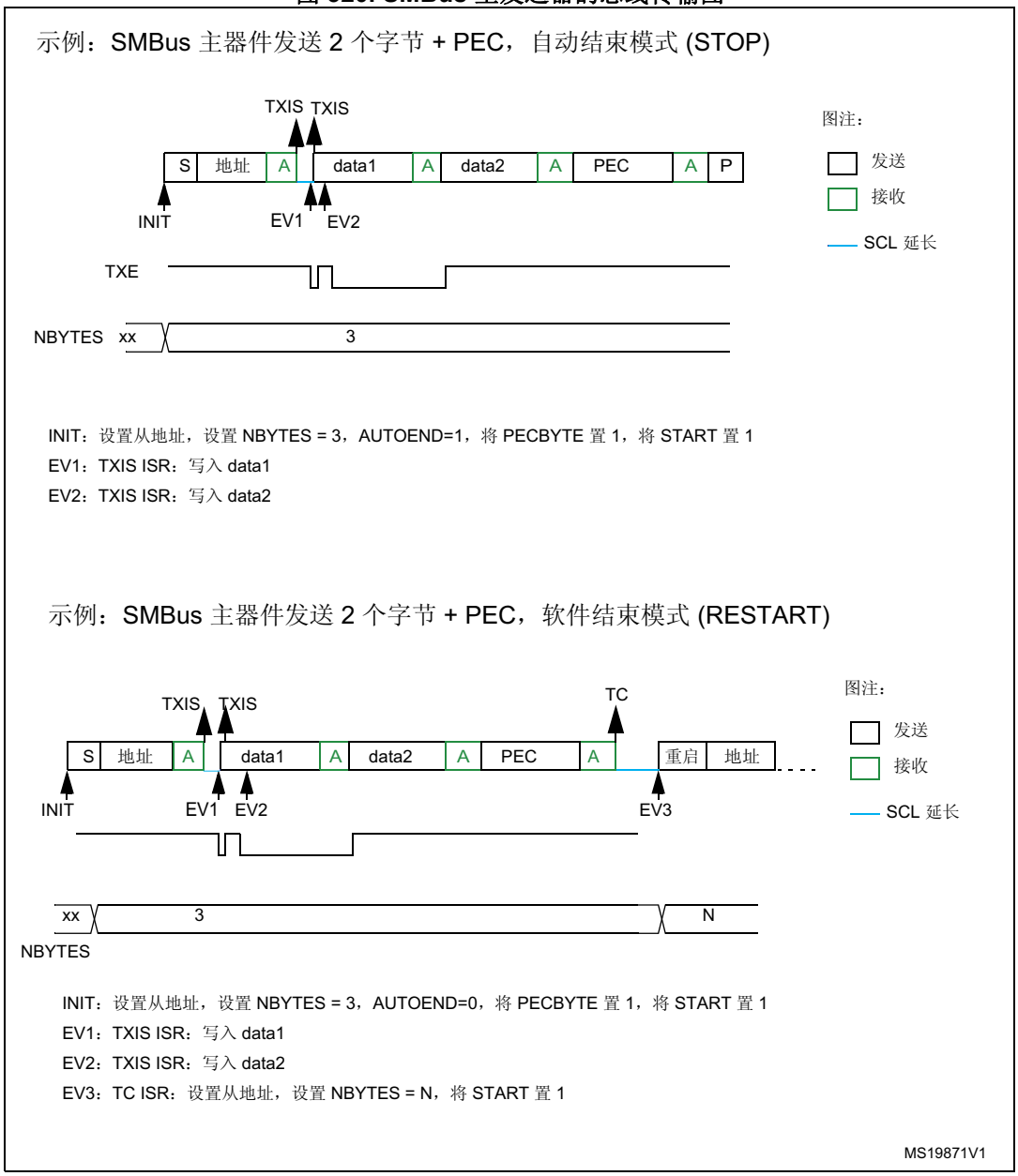
当 SMBus 主器件想要发送 PEC 时，必须在 START 位置 1 前，将 PECBYTE 位置 1 并在 NBYTES[7:0] 字段中设置字节数。在这种情况下，总 TXIS 中断数为 NBYTES-1。因此，如果 PECBYTE 位在 NBYTES=0x1 时置 1，则将自动发送 I2C_PECR 寄存器的内容。

如果 SMBus 主器件想要在 PEC 后发送停止位，则必须选择自动结束模式 (AUTOEND=1)。在这种情况下，传输 PEC 后将自动发送停止位。

如果 SMBus 主器件想要在 PEC 后发送重复起始位，则必须选择软件模式 (AUTOEND=0)。在这种情况下，发送 NBYTES-1 字节的数据后，将发送 I2C_PECR 寄存器的内容，TC 标志将在传输完 PEC 之后置 1，SCL 线的低电平时间将延长。必须在 TC 中断子程序中设置重复起始位。

注意： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 320. SMBus 主发送器的总线传输图



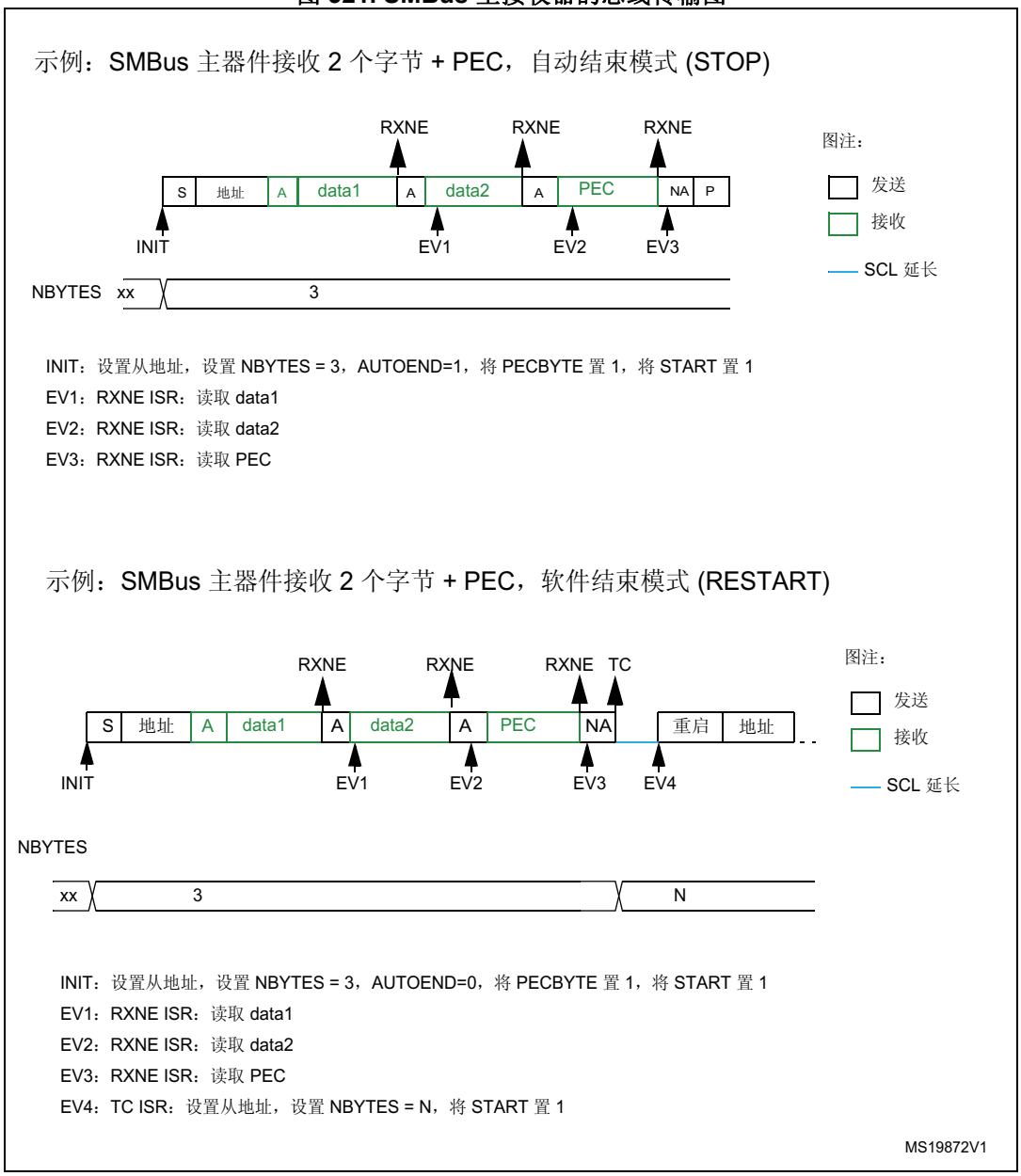
SMBus 主接收器

当 SMBus 主器件想要接收 PEC，并在传输结束后发送 STOP 时，可选择自动结束模式 (AUTOEND=1)。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从地址。在这种情况下，当接收到 NBYTES-1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。接收 PEC 字节后，给出 NACK 响应和停止位。

当 SMBus 主接收器想要接收 PEC 字节，并且在传输结束后发送重复起始位时，必须选择软件模式 (AUTOEND=0)。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从地址。在这种情况下，当接收到 NBYTES-1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。接收到 PEC 字节后，TC 标志将置 1，SCL 线的低电平时间将延长。可以在 TC 中断子程序中设置重复起始位。

注意： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 321. SMBus 主接收器的总线传输图



32.4.15 地址匹配时从停止模式唤醒

仅当支持从停止模式唤醒功能时，才涉及本节内容。请参见*第 32.3 节：I2C 特性实现*。

被寻址时，I2C能够从停止模式中唤醒MCU（APB时钟关断）。支持所有寻址模式。

将 I2C_CR1 寄存器中的 WUPEN 位置 1，可以使能从停止模式唤醒功能。对于 I2CCLK，必须选择 HSI 振荡器作为时钟源，以便从停止模式唤醒。

在停止模式期间，HSI 关闭。当检测到 START 时，I2C 接口将 HSI 接通，并延长 SCL 使其处于低电平直到唤醒 HSI。

HSI 随后用来接收地址。

地址匹配的情况下，MCU 唤醒时间内，I2C 延长 SCL 使其处于低电平。当软件清除 ADDR 标志时，此延长被释放，传输正常进行。

如果地址不匹配，HSI 再次关断，MCU 不被唤醒。

注： *如果 I2C 时钟是系统时钟，或者 WUPEN = 0，则接收到 START 后，HSI 不会接通。*

只有 ADDR 中断能够唤醒 MCU。因此，当 I2C 以主器件身份或在 ADDR 标志置 1 后以被寻址从器件身份执行传输时，不要进入停止模式。通过在 ADDR 中断程序中清除 SLEEPDEEP 位，然后仅在 STOPF 标志置 1 后再将其置 1，来对此进行管理。

注意： 数字滤波器与从停止模式唤醒功能不兼容。如果 DNF 位不等于 0，则将 WUPEN 位置 1 将不起任何作用。

注意： 只有当 I2C 时钟源为 HSI 振荡器时，该功能才可用。

注意： 必须使能时钟延长 (NOSTRETCH=0) 才能确保从停止模式唤醒功能正常工作。

注意： 如果禁止从停止模式唤醒 (WUPEN=0)，则在进入停止模式前必须禁止 I2C 外设 (PE=0)。

32.4.16 错误条件

以下错误条件可能导致通信失败。

总线错误 (BERR)

当检测到起始位或停止位但不位于第 9N 个 SCL 时钟脉冲之后时，会检测到总线错误。当 SDA 边沿出现且 SCL 为高电平时，会检测到起始或停止位。

只有当 I2C 在传输过程中用作主器件或被寻址为从器件时（即未处于从模式下的地址阶段），才会将总线错误标志置 1。

在从模式下检测到错位的起始位或重复起始位时，I2C 会像接收到正确的起始位一样进入地址识别状态。

检测到总线错误时，I2C_ISR 寄存器中的 BERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

仲裁丢失 (ARLO)

当 SDA 线上发送高电平但在 SCL 上升沿却采样到低电平时，会检测到仲裁丢失。

- 在主模式下，将在地址阶段、数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，SDA 线和 SCL 线被释放，起始控制位由硬件清零，主器件自动切换为从模式。
- 在从模式下，将在数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，传输停止，SCL 和 SDA 线被释放。

检测到仲裁丢失时，I2C_ISR 寄存器中的 ARLO 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

上溢/下溢错误 (OVR)

当满足 NOSTRETCH=1 和以下条件时，将在从模式下检测到上溢或下溢错误：

- 接收过程中接收到一个新字节但尚未读取 RXDR 寄存器。接收的新字节丢失，自动发送 NACK 来响应新字节。
- 在发送过程中：
  - 当 STOPF=1 且应发送第一个数据字节时。TXE=0 时发送 I2C_TXDR 寄存器的内容，否则发送 0xFF。
  - 必须发送一个新字节但尚未向 I2C_TXDR 寄存器写入数据时，将发送 0xFF。

检测到上溢或下溢错误时，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

数据包错误校验错误 (PECERR)

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 32.3 节：I2C 特性实现*。

当接收到的 PEC 字节与 I2C_PECR 寄存器的内容不匹配时，将检测到 PEC 错误。接收到错误的 PEC 后，将自动发送 NACK。

检测到 PEC 错误时，I2C_ISR 寄存器中的 PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

超时错误 (TIMEOUT)

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 32.3 节：I2C 特性实现*。

满足以下任何条件均会出现超时错误：

- TIDLE=0 且 SCL 的低电平持续时间达到 TIMEOUTA[11:0] 位中定义的时间：这用于检测 SMBus 超时。
- TIDLE=1 且 SDA 和 SCL 的高电平持续时间达到 TIMEOUTA[11:0] 位中定义的时间：这用于检测总线空闲情况。
- 主器件的累积时钟低电平延长时间达到了 TIMEOUTB[11:0] 位中定义的时间（SMBus $t_{LOW:MEXT}$ 参数）
- 从器件的累积时钟低电平延长时间达到了 TIMEOUTB[11:0] 位中定义的时间（SMBus $t_{LOW:SEXT}$ 参数）

当在主模式下检测到超时时，将自动发送停止位。

当在从模式下检测到超时时，将自动释放 SDA 和 SCL 线。

检测到超时错误时，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

报警 (ALERT)

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 32.3 节：I2C 特性实现*。

当 I2C 接口配置为主机 (SMBHEN=1)、使能了报警引脚检测 (ALERTEN=1) 并且在 SMBA 引脚上检测到下降沿时，ALERT 标志将置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，将生成中断。

32.4.17 DMA 请求

使用 DMA 进行发送

将 I2C_CR1 寄存器中的 TXDMAEN 位置 1 可以使能 DMA（直接存储器访问）进行发送。当 TXIS 位置 1 时，数据将从由 DMA 外设配置的 SRAM 区（请参见）装载进 I2C_TXDR 寄存器。

只有数据字节采用 DMA 进行传输。

- 在主模式下：初始化、从地址、方向、字节数和起始位均由软件编程（发送的从地址无法通过 DMA 传输）。当所有数据均通过 DMA 传输时，必须在起始位置 1 之前初始化 DMA。传输结束由 NBYTES 计数器来管理。请参见*第 955 页的主发送器*。
- 在从模式下：
  - 当 NOSTRETCH=0 时，如果所有数据均通过 DMA 传输，则必须在地址匹配事件之前（或清零 ADDR 之前在 ADDR 中断子程序中）初始化 DMA。
  - 当 NOSTRETCH=1 时，必须在地址匹配事件之前初始化 DMA。
- 支持 SMBus 时：PEC 传输由 NBYTES 计数器管理。请参见*第 969 页的SMBus 从发送器*和*第 972 页的SMBus 主发送器*。

注：如果使用 DMA 进行发送，则无需使能 TXIE 位。

使用 DMA 进行接收

将 I2C_CR1 寄存器中的 RXDMAEN 位置 1 可以使能 DMA（直接存储器访问）进行接收。当 RXNE 位置 1 时，数据将从 I2C_RXDR 寄存器装载进由 DMA 外设配置的 SRAM 区（请参见*第 11 节：直接存储器访问控制器 (DMA)*）。只有数据字节（包括 PEC）采用 DMA 进行传输。

- 在主模式下：初始化、从地址、方向、字节数和起始位均由软件编程。当所有数据均通过 DMA 传输时，必须在起始位置 1 之前初始化 DMA。传输结束由 NBYTES 计数器来管理。
- 在从模式下，当 NOSTRETCH=0 时，如果所有数据均通过 DMA 传输，则必须在地址匹配事件之前（或清零 ADDR 标志之前在 ADDR 中断子程序中）初始化 DMA。
- 如果支持 SMBus（请参见*第 32.3 节：I2C 特性实现*）：PEC 传输由 NBYTES 计数器管理。请参见*第 970 页的SMBus 从接收器*和*第 973 页的SMBus 主接收器*。

注：如果使用 DMA 进行接收，则无需使能 RXIE 位。

32.4.18 调试模式

当微控制器进入调试模式时（内核停止），SMBus 超时定时器会根据 DBG 模块中的 DBG_I2Cx_ 配置位选择继续正常工作或者停止工作。

32.5 I2C 低功耗模式

表 189. 低功耗模式对 I2C 的影响

模式	说明
睡眠	无影响。I2C 中断可使器件退出睡眠模式。
停止(1)	I2C 模块的寄存器内容仍被保持。如果 WUPEN = 1 并且 I2C 的时钟由内部振荡器 (HSI) 提供：地址识别功能正常。I2C 地址匹配条件会导致器件退出停止模式。如果 WUPEN=0：必须在进入停止模式之前禁止 I2C。
待机	I2C 外设掉电，退出待机模式后必须重新初始化。

1. 有关每个实例支持的停止模式的信息，请参见 I2C 实现表。如果不支持从特定停止模式唤醒，则必须在进入此停止模式之前禁止实例。

32.6 I2C 中断

下表给出了 I2C 中断请求列表。

表 190. I2C 中断请求

中断缩写		中断事件	事件标志	使能控制位	中断清除方法	退出睡眠模式	退出 Stop 模式	退出 Standby 模式
I2C	I2C_EV	接收缓冲区非空	RXNE	RXIE	读取 I2C_RXDR 寄存器	是	否	否
		发送缓冲区中断状态	TXIS	TXIE	写入 I2C_TXDR 寄存器			
		停止位检测中断标志	STOPF	STOPIE	写入 STOPCF=1			
		传输完成等待重载	TCR	TCIE	写入 I2C_CR2 (NBYTES[7:0] ≠ 0)			
		传输完成	TC		写入 START=1 或 STOP=1			
		地址匹配	ADDR	ADDRIE	写入 ADDRCF=1		是(1)	
		接收到 NACK 应答	NACKF	NACKIE	写入 NACKCF=1		否	
	I2C_ER	总线错误	BERR	ERRIE	写入 BERRCF=1	是	否	否
		仲裁丢失	ARLO		写入 ARLOCF=1			
		上溢/下溢	OVR		写入 OVRCF=1			
		PEC 错误	PECERR		写入 PECERRCF=1			
		超时/t_{LOW} 错误	TIMEOUT		写入 TIMEOUTCF=1			
		SMBus 报警	ALERT		写入 ALERTCF=1			

1. 仅当 I2C 实例支持从停止模式唤醒功能时，ADDR 匹配事件才能将器件从停止模式唤醒。请参见*第 32.3 节：I2C 特性实现*。

32.7 I2C 寄存器

有关寄存器说明中使用的缩写，请参见*第 58 页的第 1.2 节*。

外设寄存器按字（32 位）进行访问。

32.7.1 I2C 控制寄存器 1 (I2C_CR1)

I2C control register 1

偏移地址：0x00

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	WUPEN	NOSTRETCH	SBC
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDMAEN	TXDMAEN	Res.	ANFOFF	DNF[3:0]				ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23 **PECEN**：PEC 使能 (PEC enable)

0：禁止 PEC 计算

1：使能 PEC 计算

注：　如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 32.3 节：I2C 特性实现。

位 22 **ALERTEN**：SMBus 报警使能 (SMBus alert enable)

从机模式 (SMBHEN=0)：

0：将 SMBA 引脚释放为高电平并禁止报警响应地址头：0001100x 后跟 NACK。

1：将 SMBA 引脚驱动为低电平并使能报警响应地址头：0001100x 后跟 ACK。

主机模式 (SMBHEN=1)：

0：不支持 SMBus 报警引脚 (SMBA)。

1：支持 SMBus 报警引脚 (SMBA)。

注：　当 ALERTEN=0 时，SMBA 引脚可用作标准 GPIO。

如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 32.3 节：I2C 特性实现。

位 21 **SMBDEN**：SMBus 器件默认地址使能 (SMBus Device Default address enable)

0：禁止器件默认地址。不对地址 0b1100001x 应答。

1：使能器件默认地址。对地址 0b1100001x 应答。

注：　如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 32.3 节：I2C 特性实现。

位 20 **SMBHEN**：SMBus 主机地址使能 (SMBus Host address enable)

0：禁止主机地址。不对地址 0b0001000x 应答。

1：使能主机地址。对地址 0b0001000x 应答。

注：　如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 32.3 节：I2C 特性实现。

位 19 **GCEN**：广播呼叫使能 (General call enable)

0：禁止广播呼叫。不对地址 0b00000000 应答。

1：使能广播呼叫。对地址 0b00000000 应答。

位 18 **WUPEN**：从停止模式唤醒使能 (Wakeup from Stop mode enable)

0：禁止从停止模式唤醒。

1：使能从停止模式唤醒。

注：　如果不支持从停止模式唤醒功能，该位保留并由硬件强制清零。请参见第 32.3 节：I2C 特性实现。

注：　只有当 DNF = “0000”时，WUPEN 才能置 1。

位 17 **NOSTRETCH**：时钟延长禁止 (Clock stretching disable)

该位用于在从模式下禁止时钟延长。它在主模式下必须保持清零。

0：使能时钟延长。

1：禁止时钟延长。

注：　该位只能在 I2C 禁止时 (PE = 0) 编程。

位 16 **SBC**：从设备模式下的字节控制 (Slave byte control)

该位用于在从设备模式下使能硬件字节控制。

0：禁止从设备模式下的字节控制。

1：使能从设备模式下的字节控制。

位 15 **RXDMAEN**：DMA 接收请求使能 (DMA reception requests enable)

0：禁止 DMA 接收请求。

1：使能 DMA 接收请求。

位 14 **TXDMAEN**：DMA 发送请求使能 (DMA transmission requests enable)

0：禁止 DMA 发送请求。

1：使能 DMA 发送请求。

位 13 保留，必须保持复位值。

位 12 **ANFOFF**：模拟噪声滤波器关闭 (Analog noise filter OFF)

0：使能模拟噪声滤波器。

1：禁止模拟噪声滤波器。

注：　该位只能在 I2C 禁止时 (PE = 0) 编程。

位 11:8 **DNF[3:0]**：数字噪声滤波器 (Digital noise filter)

这些位用于配置 SDA 和 SCL 输入端的数字噪声滤波器。数字滤波器可滤除脉宽 DNF[3:0] * t_{I2CCLK} 以下的尖峰

0000：禁止数字滤波器

0001：使能数字滤波器，可滤除的噪声尖峰脉宽可达 1 t_{I2CCLK}

...

1111：使能数字滤波器，可滤除的噪声尖峰脉宽可达 15 t_{I2CCLK}

注：　如果模拟滤波器也已使能，数字滤波将叠加在模拟滤波之上。

该滤波器只能在 I2C 禁止时 (PE = 0) 编程。

位 7 **ERRIE**：错误中断使能 (Error interrupts enable)
0：禁止错误检测中断
1：使能错误检测中断
注： *以下任一错误均会生成中断：*
仲裁丢失 (ARLO)
总线错误检测 (BERR)
上溢/ 下溢 (OVR)
超时检测 (TIMEOUT)
PEC 错误检测 (PECERR)
报警引脚事件检测 (ALERT)

位 6 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)
0：禁止传输完成中断
1：使能传输完成中断
注： *以下任一事件均会生成中断：*
传输完成 (TC)
传输完成等待重载 (TCR)

位 5 **STOPIE**：停止位检测中断使能 (Stop detection Interrupt enable)
0：禁止停止位检测 (STOPF) 中断
1：使能停止位检测 (STOPF) 中断

位 4 **NACKIE**：接收到否定应答中断使能 (Not acknowledge received Interrupt enable)
0：禁止接收到否定应答 (NACKF) 中断
1：使能接收到否定应答 (NACKF) 中断

位 3 **ADDRIE**：地址匹配中断使能（仅从模式）(Address match Interrupt enable (slave only))
0：禁止地址匹配 (ADDR) 中断
1：使能地址匹配 (ADDR) 中断

位 2 **RXIE**：RX 中断使能 (RX Interrupt enable)
0：禁止接收 (RXNE) 中断
1：使能接收 (RXNE) 中断

位 1 **TXIE**：TX 中断使能 (TX Interrupt enable)
0：禁止发送 (TXIS) 中断
1：使能发送 (TXIS) 中断

位 0 **PE**：外设使能 (Peripheral enable)
0：禁止外设
1：使能外设
注： *当 PE=0 时，将释放 I2C SCL 线和 SDA 线。内部状态机和状态位均恢复为复位值。清零时，PE 必须保持低电平状态至少 3 个 APB 时钟周期。*

32.7.2 I2C 控制寄存器 2 (I2C_CR2)

I2C control register 2

偏移地址：0x04

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PEC BYTE	AUTOEND	RELOAD	NBYTES[7:0]							
					rs	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
NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:0]									
rs	rs	rs	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:27 保留，必须保持复位值。

位 26 **PECBYTE**：数据包错误校验字节 (Packet error checking byte)

此位由软件置 1，并可在 PEC 传输完成时、接收到停止位或匹配地址时、或者 PE=0 时由硬件清零。

0：不传输 PEC。

1：请求 PEC 发送/接收。

注： *向该位写入“0”不起作用。*

当 RELOAD 置 1 时，该位不起作用。

当 SBC=0 时，该位在从模式下不起作用。

如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 32.3 节：I2C 特性实现。

位 25 **AUTOEND**：自动结束模式（主模式）(Automatic end mode (master mode))

此位由软件置 1 和清零。

0：软件结束模式：当 NBYTES 数据传输完成时，TC 标志将置 1，SCL 的低电平时间将延长直到相应软件操作结束。

1：自动结束模式：当 NBYTES 数据传输完成时，将自动发送停止位。

注： *在从模式下，或当 RELOAD 位置 1 时，该位不起作用。*

位 24 **RELOAD**：NBYTES 重载模式 (NBYTES reload mode)

此位由软件置 1 和清零。

0：传输 NBYTES 数据（后跟停止位或重复起始位）之后即完成传输。

1：传输 NBYTES 数据之后未完成传输（将重载 NBYTES）。当 NBYTES 数据传输完成时，TCR 标志将置 1，SCL 的低电平时间将延长直到相应软件操作结束。

位 23:16 **NBYTES[7:0]**：字节数 (Number of bytes)

在此设置待发送/接收的字节数。在从模式下，当 SBC=0 时，该字段为无关字段。

注： *START 位置 1 时，不允许更改这些位。*

位 15 **NACK**：NACK 生成（从模式）(NACK generation (slave mode))

此位由软件置 1，并可在发送 NACK 时、接收到停止位或匹配地址时、或者 PE=0 时由硬件清零。

0：在当前接收的字节后发送 ACK。

1：在当前接收的字节后发送 NACK。

注：　向该位写入“0”不起作用。

该位仅在从模式下使用：在主接收器模式下，无论 NACK 位的值为何，最后一个字节（后跟停止位或重复起始位）后都将自动生成 NACK。

当从接收器 NOSTRETCH 模式下发生上溢时，无论 NACK 位的值为何，都将自动生成 NACK。

使能硬件 PEC 校验时 (PECBYTE=1)，PEC 应答值与 NACK 值无关。

位 14 **STOP**：停止位生成（主模式）*(*Stop generation (master mode))

该位由软件置 1，并可在检测到停止位时或 PE = 0 时由硬件清零。

在主模式下：

0：不生成停止位。

1：在当前字节传输完成后生成停止位。

注：　向该位写入“0”不起作用。

位 13 **START**：起始位生成 (Start generation)

该位由软件置 1，并可在发送起始位（后跟地址序列）之后、发生仲裁丢失时、出现超时错误时、或者 PE = 0 时由硬件清零。它也可由软件清零，方法是向 I2C_ICR 寄存器中的 ADDRCF 位写入“1”。

0：不生成起始位。

1：生成重复起始/起始位：

如果 I2C 已处于主模式下且 AUTOEND = 0，则将该位置 1 会在 NBYTES 传输结束后且 RELOAD=0 的情况下生成重复起始位。

否则，将该位置 1 会在总线释放后立即生成起始位。

注：　向该位写入“0”不起作用。

即使总线繁忙或 I2C 处于从模式，也可将 START 位置 1。

当 RELOAD 置 1 时，该位不起作用。

位 12 **HEAD10R**：读方向传输时，只发送 10 位地址的前 7 位地址头字节（主接收器模式）(10-bit address header only read direction (master receiver mode))

0：主器件发送完整的 10 位从地址读序列：起始位 + 带写方向的 2 字节 10 位地址 + 重复起始位 + 带读方向的 10 位地址的前 7 位。

1：主器件只发送 10 位地址的前 7 位,后跟读方向。

注：　START 位置 1 时，不允许更改此位。

位 11 **ADD10**：10 位寻址模式（主模式）(10-bit addressing mode (master mode))

0：主器件工作在 7 位寻址模式下

1：主器件工作在 10 位寻址模式下

注：　START 位置 1 时，不允许更改此位。

位 10 **RD_WRN**：传输方向（主模式）(Transfer direction (master mode))
0：主器件请求写传输。
1：主器件请求读传输。
注： START 位置 1 时，不允许更改此位。

位 9:0 **SADD[9:0]**：从地址（主模式）(Slave address (master mode))
在 7 位寻址模式 (ADD10 = 0) 下：
SADD[7:1] 应写入待发送的 7 位从地址。SADD[9]、SADD[8] 和 SADD[0] 位为无关位。
在 10 位寻址模式 (ADD10 = 1) 下：
SADD[9:0] 应写入待发送的 10 位从地址。
注： START 位置 1 时，不允许更改这些位。

32.7.3 I2C 自身地址 1 寄存器 (I2C_OAR1)

I2C own address 1 register

偏移地址：0x08

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA1EN	Res.	Res.	Res.	Res.	OA1 MODE	OA1[9:0]									
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **OA1EN**：设备自身地址 1 使能 (Own Address 1 enable)
0：禁止设备自身地址 1。不对接收的从地址 OA1 应答。
1：使能设备自身地址 1。对接收的从地址 OA1 应答。

位 14:11 保留，必须保持复位值。

位 10 **OA1MODE**：设备自身地址 1 10 位模式 (Own Address 1 10-bit mode)
0：设备自身地址 1 为 7 位地址。
1：设备自身地址 1 为 10 位地址。
注： 仅可在 OA1EN=0 时写入该位。

位 9:0 **OA1[9:0]**：接口自身从地址 (Interface own slave address)
7 位寻址模式：OA1[7:1] 包含 7 位自身从地址。OA1[9]、OA1[8] 和 OA1[0] 位为无关位。
10 位寻址模式：OA1[9:0] 包含 10 位自身从地址。
注： 仅可在 OA1EN=0 时写入这些位。

32.7.4 I2C 自身地址 2 寄存器 (I2C_OAR2)

I2C own address 2 register

偏移地址：0x0C

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2EN	Res.	Res.	Res.	Res.	OA2MSK[2:0]			OA2[7:1]							Res.
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位 31:16 保留，必须保持复位值。

位 15 **OA2EN**：设备自身地址 2 使能 (Own Address 2 enable)
0：禁止设备自身地址 2。不对接收的从地址 OA2 应答。
1：使能设备自身地址 2。对接收的从地址 OA2 应答。

位 14:11 保留，必须保持复位值。

位 10:8 **OA2MSK[2:0]**：设备自身地址 2 屏蔽位 (Own Address 2 masks)
000：无屏蔽
001：OA2[1] 被屏蔽，为无关位。仅比较 OA2[7:2]。
010：OA2[2:1] 被屏蔽，为无关位。仅比较 OA2[7:3]。
011：OA2[3:1] 被屏蔽，为无关位。仅比较 OA2[7:4]。
100：OA2[4:1] 被屏蔽，为无关位。仅比较 OA2[7:5]。
101：OA2[5:1] 被屏蔽，为无关位。仅比较 OA2[7:6]。
110：OA2[6:1] 被屏蔽，为无关位。仅比较 OA2[7]。
111：OA2[7:1] 被屏蔽，为无关位。不进行比较，对接收到的全部 7 位地址（保留位除外）应答。

注： *仅可在 OA2EN=0 时写入这些位。*
只要 OA2MSK 不等于 0，即使比较匹配，也不会对保留的 I2C 地址（0b0000xxx 和 0b1111xxx）应答。

位 7:1 **OA2[7:1]**：接口地址 (Interface address)
7 位寻址模式：7 位地址

注： *仅可在 OA2EN=0 时写入这些位。*

位 0 保留，必须保持复位值。

32.7.5 I2C 时序寄存器 (I2C_TIMINGR)

I2C timing register

偏移地址：0x10

复位值：0x0000 0000

访问：无等待周期

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

位 31:28 **PRESC[3:0]**：时序预分频因子 (Timing prescaler)

该字段用于对 I2CCLK 进行预分频，以生成用于数据建立和保持计数器（请参见*第 937 页的 I2C 时序*）以及 SCL 高电平和低电平计数器（请参见*第 951 页的I2C 主模式初始化*）的时钟周期 t_{PRESC}。

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

位 27:24 保留，必须保持复位值。

位 23:20 **SCLDEL[3:0]**：数据建立时间 (Data setup time)

该字段用于在 SDA 边沿和 SCL 上升沿之间生成延时 t_{SCLDEL}。在主模式和从模式下，如果 NOSTRETCH = 0，则 SCL 线的低电平时间将在 t_{SCLDEL} 期间延长。

$t_{SCLDEL} = (SCLDEL+1) \times t_{PRESC}$

注： t_{SCLDEL} 用于生成 $t_{SU:DAT}$ 时序。

位 19:16 **SDADEL[3:0]**：数据保持时间 (Data hold time)

该字段用于在 SCL 下降沿和 SDA 边沿之间生成延时 t_{SDADEL}。在主模式和从模式下，如果 NOSTRETCH = 0，则 SCL 线的低电平时间将在 t_{SDADEL} 期间延长。

$t_{SDADEL} = SDADEL \times t_{PRESC}$

注： SDADEL 用于生成 $t_{HD:DAT}$ 时序。

位 15:8 **SCLH[7:0]**：SCL 高电平周期（主模式）(SCL high period (master mode))

在主模式下，该字段用于生成 SCL 高电平周期。

$t_{SCLH} = (SCLH+1) \times t_{PRESC}$

注： SCLH 还用于生成 $t_{SU:STO}$ 和 $t_{HD:STA}$ 时序。

位 7:0 **SCLL[7:0]**：SCL 低电平周期（主模式）(SCL low period (master mode))

在主模式下，该字段用于生成 SCL 低电平周期。

$t_{SCLL} = (SCLL+1) \times t_{PRESC}$

注： SCLL 还用于生成 t_{BUF} 和 $t_{SU:STA}$ 时序。

注： 该寄存器必须在 I2C 禁止时 (PE = 0) 进行配置。

注： STM32CubeMX 工具进行计算并将 I2C_TIMINGR 内容提供在 I2C 配置窗口中。

32.7.6 I2C 超时寄存器 (I2C_TIMEOUTR)

I2C timeout register

偏移地址：0x14

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]											
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
TIMOUTEN	Res.	Res.	TIDLE	TIMEOUTA[11:0]											
rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **TEXTEN**：时钟信号延展超时使能 (Extended clock timeout enable)

0：禁止时钟信号延展超时检测

1：使能时钟信号延展超时检测 当 I2C 接口执行 SCL 延展的累积时间超过 $t_{LOW:EXT}$ 时，将检测到超时错误 (TIMEOUT=1)。

位 30:28 保留，必须保持复位值。

位 27:16 **TIMEOUTB[11:0]**：总线超时 B (Bus timeout B)

该字段用于配置累积时钟延展超时：

在主模式下，将检测主器件的累积时钟低电平延展时间 ($t_{LOW:MEXT}$)

在从模式下，将检测从器件的累积时钟低电平延展时间 ($t_{LOW:SEXT}$)

$t_{LOW:EXT}$ = (TIMEOUTB+1) x 2048 x t_{I2CCLK}

注： 仅可在 TEXTEN=0 时写入这些位。

位 15 **TIMOUTEN**：时钟超时使能 (Clock timeout enable)

0：禁止 SCL 超时检测

1：使能 SCL 超时检测：当 SCL 的低电平时间超过 $t_{TIMEOUT}$ (TIDLE=0)，或 SCL 的高电平时间超过 t_{IDLE} (TIDLE=1) 时，将检测到超时错误 (TIMEOUT=1)。

位 14:13 保留，必须保持复位值。

位 12 **TIDLE**：空闲时钟超时检测 (Idle clock timeout detection)

0：TIMEOUTA 用于检测 SCL 低电平超时

1：TIMEOUTA 用于检测 SCL 和 SDA 高电平超时（总线空闲条件）

注： 仅可在 TIMOUTEN=0 时写入该位。

位 11:0 **TIMEOUTA[11:0]**：总线超时 A (Bus Timeout A)

该字段用于配置：

SCL 低电平超时条件 $t_{TIMEOUT}$（当 TIDLE=0 时）

$t_{TIMEOUT}$= (TIMEOUTA+1) x 2048 x t_{I2CCLK}

总线空闲条件，即 SCL 和 SDA 高电平（当 TIDLE=1 时）

t_{IDLE}= (TIMEOUTA+1) x 4 x t_{I2CCLK}

注： 仅可在 TIMOUTEN=0 时写入这些位。

注： 如果不支持 SMBus 功能，该寄存器保留，并由硬件强制为“0x00000000”。请参见第 32.3 节：I2C 特性实现。

32.7.7 I2C 中断和状态寄存器 (I2C_ISR)

I2C interrupt and status register

偏移地址：0x18

复位值：0x0000 0001

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR
								r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.	ALERT	TIME OUT	PEC ERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r		r	r	r	r	r	r	r	r	r	r	r	r	rs	rs

位 31:24 保留，必须保持复位值。

位 23:17 **ADDCODE[6:0]**：地址匹配代码（从模式）(Address match code (Slave mode))

发生地址匹配事件时 (ADDR = 1)，这些位更新为接收到的地址。

在 10 位地址的情况下，ADDCODE 提供 10 位地址的头字节，后跟地址的 2 个 MSB。

位 16 **DIR**：传输方向（从模式）(Transfer direction (Slave mode))

该标志在发生地址匹配事件时 (ADDR=1) 更新。

0：写传输，从器件进入接收器模式。

1：读传输，从器件进入发送器模式。

位 15 **BUSY**：总线繁忙 (Bus busy)

该标志用于指示总线上正在进行通信。当检测到起始位时，该位由硬件置 1。当检测到停止位或 PE = 0 时，该位由硬件清零。

位 14 保留，必须保持复位值。

位 13 **ALERT**：SMBus 报警 (SMBus alert)

当 SMBHEN=1（SMBus 主机配置）、ALERTEN=1 且在 SMBA 引脚上检测到 SMBALERT 事件（下降沿）时，该标志由硬件置 1。该位由软件清零，方法是将 ALERTCF 位置 1。

注： *当 PE=0 时，该位由硬件清零。*

如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 32.3 节：I2C 特性实现。

位 12 **TIMEOUT**：超时或 t_{LOW} 检测标志 (Timeout or t_{LOW} detection flag)

发生超时或延长时钟超时时，该标志由硬件置 1。该位由软件清零，方法是将 TIMEOUTCF 位置 1。

注： *当 PE=0 时，该位由硬件清零。*

如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 32.3 节：I2C 特性实现。

位 11 **PECERR**：接收期间的 PEC 错误 (PEC Error in reception)

当接收到的 PEC 与 PEC 寄存器的内容不匹配时，该标志由硬件置 1。接收到错误的 PEC 后，将自动发送 NACK。该标志由软件清零，方法是将 PECCF 位置 1。

注：　当 PE=0 时，该位由硬件清零。

如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 32.3 节：I2C 特性实现。

位 10 **OVR**：上溢/下溢（从模式）(Overrun/Underrun (slave mode))

在从模式下且 NOSTRETCH=1 时，如果发生上溢/下溢错误，该标志由硬件置 1。该标志由软件清零，方法是将 OVRCF 位置 1。

注：　当 PE=0 时，该位由硬件清零。

位 9 **ARLO**：仲裁丢失 (Arbitration lost)

发生仲裁丢失时，该标志由硬件置 1。该标志由软件清零，方法是将 ARLOCF 位置 1。

注：　当 PE=0 时，该位由硬件清零。

位 8 **BERR**：总线错误 (Bus error)

当检测到错位的起始位或停止位，而外设也参与传输时，该标志由硬件置 1。在从模式下的地址阶段，该标志不会置 1。该标志由软件清零，方法是将 *BERRCF 位置 1*。

注：　当 PE=0 时，该位由硬件清零。

位 7 **TCR**：传输完成等待重载 (Transfer Complete Reload)

当 RELOAD=1 且 NBYTES 数据传输完成时，该标志由硬件置 1。当 NBYTES 写入一个非零值时，该标志由软件清零。

注：　当 PE=0 时，该位由硬件清零。

该标志单独用于主模式，SBC 位置 1 时单独用于从模式。

位 6 **TC**：传输完成（主模式）(Transfer Complete (master mode))

当 RELOAD=0、AUTOEND=0 且 NBYTES 数据传输完成时，该标志由硬件置 1。当 START 位或 STOP 位置 1 时，该标志由软件清零。

注：　当 PE=0 时，该位由硬件清零。

位 5 **STOPF**：停止位检测标志 (Stop detection flag)

当在总线上检测到停止位，且外设也参与本次传输时，该标志由硬件置 1：

– 外设作为主器件，该位置位的前提是外设已经发出停止位。

– 外设作为从器件，该位置位的前提条件是此次传输的寻址对象就是该外设。

该标志由软件清零，方法是将 STOPCF 位置 1。

注：　当 PE=0 时，该位由硬件清零。

位 4 **NACKF**：接收到否定应答标志 (Not Acknowledge received flag)

传输完字节后接收到 NACK 时，该标志由硬件置 1。该标志由软件清零，方法是将 NACKCF 位置 1。

注：　当 PE=0 时，该位由硬件清零。

位 3 **ADDR**：地址匹配（从模式）(Address matched (slave mode))

接收到的地址与使能的从设备地址之一匹配时，该位由硬件置 1。该位由软件清零，方法是将 *ADDRCF 位置 1*。

注：　当 PE=0 时，该位由硬件清零。

位 2 **RXNE**：接收数据寄存器不为空（接收器）(Receive data register not empty (receivers))
当接收到的数据已复制到 I2C_RXDR 寄存器且准备就绪可供读取时，该位由硬件置 1。读取 I2C_RXDR 时，将清零该位。
注：　当 PE=0 时，该位由硬件清零。

位 1 **TXIS**：发送中断状态（发送器）(Transmit interrupt status (transmitters))
当 I2C_TXDR 寄存器为空时，该位由硬件置 1，待发送的数据必须写入 I2C_TXDR 寄存器。下一个待发送的数据写入 I2C_TXDR 寄存器时，该位被清零。
该位只能在 NOSTRETCH=1 时由软件写入“1”，以生成 TXIS 事件（TXIE=1 时为中断，TXDMAEN=1 时为 DMA 请求）。
注：　当 PE=0 时，该位由硬件清零。

位 0 **TXE**：发送数据寄存器为空（发送器）(Transmit data register empty (transmitters))
当 I2C_TXDR 寄存器为空时，该位由硬件置 1。下一个待发送的数据写入 I2C_TXDR 寄存器时，该位被清零。
该位可由软件写入“1”，以刷新发送数据寄存器 I2C_TXDR。
注：　当 PE=0 时，该位由硬件置 1。

32.7.8 I2C 中断清零寄存器 (I2C_ICR)

I2C interrupt clear register

偏移地址：0x1C

复位值：0x0000 0000

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ALERTCF	TIMOUTCF	PECCF	OVRCF	ARLOCF	BERRCF	Res.	Res.	STOPCF	NACKCF	ADDRCF	Res.	Res.	Res.
		w	w	w	w	w	w			w	w	w			

位 31:14 保留，必须保持复位值。

位 13 **ALERTCF**：报警标志清零 (Alert flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 ALERT 标志将清零。
注：　如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 32.3 节：I2C 特性实现。

位 12 **TIMOUTCF**：超时检测标志清零 (Timeout detection flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 TIMEOUT 标志将清零。
注：　如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 32.3 节：I2C 特性实现。

位 11 **PECCF**：PEC 错误标志清零 (PEC Error flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 PECERR 标志将清零。
注：　如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 32.3 节：I2C 特性实现。

位 10 **OVRCF**：上溢/下溢标志清零 (Overrun/Underrun flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 OVR 标志将清零。

位 9 **ARLOCF**：仲裁丢失标志清零 (Arbitration lost flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 ARLO 标志将清零。

位 8 **BERRCF**：总线错误标志清零 (Bus error flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 BERRF 标志将清零。

位 7:6 保留，必须保持复位值。

位 5 **STOPCF**：停止位检测标志清零 (STOP detection flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 STOPF 标志将清零。

位 4 **NACKCF**：否定应答标志清零 (Not Acknowledge flag clear)
向此位写入 1 时，I2C_ISR 寄存器中的 NACKF 标志将清零。

位 3 **ADDRCF**：地址匹配标志清零 (Address Matched flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 ADDR 标志将清零。将 1 写入此位时，I2C_CR2 寄存器中的 START 位也将清零。

位 2:0 保留，必须保持复位值。

32.7.9 I2C PEC 寄存器 (I2C_PECR)

I2C PEC register

偏移地址：0x20

复位值：0x0000 0000

访问：无等待周期

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PEC[7:0]**：数据包错误校验寄存器 (Packet error checking register)
当 PECEN=1 时，此字段包含内部 PEC。
当 PE=0 时，PEC 由硬件清零。

注： *如果不支持 SMBus 功能，该寄存器保留，并由硬件强制为“0x00000000”。请参见第 32.3 节：I2C 特性实现。*

32.7.10 I2C 接收数据寄存器 (I2C_RXDR)

I2C receive data register

偏移地址：0x24

复位值：0x0000 0000

访问：无等待周期

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **RXDATA[7:0]**：8 位接收数据 (8-bit receive data)
从 I^2C 总线接收的数据字节。

32.7.11 I2C 发送数据寄存器 (I2C_TXDR)

I2C transmit data register

偏移地址：0x28

复位值：0x0000 0000

访问：无等待周期

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **TXDATA[7:0]**：8 位发送数据 (8-bit transmit data)
待发送到 I^2C 总线的数据字节
注： 仅可在 TXE=1 时写入这些位。

32.7.12 I2C寄存器映射

下表提供了 I2C 寄存器映射和复位值。

表 191. I2C 寄存器映射和复位值

偏移	寄存器名称	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
0x0	**I2C_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	WUPEN	NOSTRETCH	SBC	RXDMAEN	TXDMAEN	Res.	ANFOFF	DNF[3:0]				ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
	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
0x4	**I2C_CR2**	Res.	Res.	Res.	Res.	Res.	PECBYTE	AUTOEND	RELOAD	NBYTES[7:0]								NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:0]									
	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
0x8	**I2C_OAR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA1EN	Res.	Res.	Res.	Res.	OA1MODE	OA1[9:0]									
	Reset value																	0					0	0	0	0	0	0	0	0	0	0	0
0xC	**I2C_OAR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA2EN	Res.	Res.	Res.	Res.	OA2MSK [2:0]	OA2[7:1]									Res.
	Reset value																	0					0	0	0	0	0	0	0	0	0	0	
0x10	**I2C_TIMINGR**	PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL[3:0]				SDADEL[3:0]				SCLH[7:0]								SCLL[7:0]							
	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
0x14	**I2C_TIMEOUTR**	TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]												TIMOUTEN	Res.	Res.	TIDLE	TIMEOUTA[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	0	0
0x18	**I2C_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR	BUSY	Res.	ALERT	TIMEOUT	PECERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
	Reset value									0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x1C	**I2C_ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALERTCF	TIMOUTCF	PECCF	OVRCF	ARLOCF	BERRCF	Res.	Res.	STOPCF	NACKCF	ADDRCF	Res.	Res.	Res.
	Reset value																			0	0	0	0	0	0			0	0	0			
0x20	**I2C_PECR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x24	**I2C_RXDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

表 191. I2C 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x28	**I2C_TXDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

33 通用同步/异步收发器 (USART/UART)

本节介绍通用同步/异步收发器 (USART)。

33.1 USART 简介

USART 能够灵活地与外部设备进行全双工数据交换，满足外部设备对工业标准 NRZ 异步串行数据格式的要求。USART 通过小数波特率发生器实现了多种波特率。

USART 不仅支持同步单向通信和半双工单线通信，也支持 LIN（局域互连网络）、智能卡协议、IrDA（红外线数据协会）SIR ENDEC 规范和调制解调器操作 (CTS/RTS)，还支持多处理器通信。

通过配置多个缓冲区使用 DMA（直接存储器访问）可实现高速数据通信。

33.2 USART 主要特性

- 全双工异步通信
- NRZ 标准格式（标记/空格）
- 可配置为 16 倍过采样或 8 倍过采样，从而在速度容差与时钟容差之间取得最佳平衡
- 波特率发生器系统
- 两个用于收发数据的内部 FIFO
  每个 FIFO 均可由软件使能/禁止，并且均带有一个状态标志
- 通用可编程收发波特率
- 双时钟域，带有独立于 PCLK 的外设专用内核时钟
- 自动波特率检测
- 数据字长度可编程（7 位、8 位或 9 位）
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 停止位可配置（支持 1 个或 2 个停止位）
- 用于同步通信的同步主/从模式和时钟输出/输入
- SPI 从模式发送下溢错误标志
- 单线半双工通信
- 使用 DMA 实现连续通信
- DMA 能使用预留的 SRAM 来缓存收/发的字节
- 发射器和接收器有单独的使能位
- 发送和接收的信号极性能单独控制
- Tx/Rx 引脚配置可交换
- 调制解调器和 RS-485 收发器的硬件流控制
- 通信控制/错误检测标志
- 奇偶校验控制：
  - 发送奇偶校验位
  - 检查接收的数据字节的奇偶性
- 具有标志的中断源标志
- 多处理器通信：从静默模式唤醒（通过空闲线检测或地址标记检测）

33.3 USART 扩展特性

- LIN 主模式同步停止符号发送功能和 LIN 从模式停止符号检测功能
  - 对 USART 进行 LIN 硬件配置时可生成 13 位停止符号和检测 10/11 位停止符号
- IrDA SIR 编解码器在正常模式下，支持 3/16 位持续时间
- 智能卡模式
  - 对于 ISO/IEC 7816-3 标准中定义的智能卡，支持 T=0 和 T=1 异步协议
  - 智能卡工作模式下，支持 0.5 和 1.5 个停止位
- 支持 ModBus 通信
  - 超时功能
  - CR/LF 字符识别

33.4 USART 实现

表 192 介绍了 STM32WB55xx 器件上的 USART 实现。

表 192. USART/LPUART 功能

USART 模式/特性(1)	USART1	LPUART
调制解调器的硬件流控制	X	X
使用 DMA 进行连续通信	X	X
多处理器通信	X	X
同步模式（主/从）	X	-
智能卡模式	X	-
单线半双工通信	X	X
IrDA SIR ENDEC 模块	X	-
LIN 模式	X	-
双时钟域和从低功耗模式唤醒	X	X
接收器超时中断	X	-
Modbus 通信	X	-
自动波特率检测	X	-
驱动器使能	X	X
USART 数据长度	7 位、8 位和 9 位	
Tx/Rx FIFO	X	X
Tx/Rx FIFO 大小	8	

1. X = 支持。

33.5 USART 功能说明

33.5.1 USART 框图

图 322. USART 框图

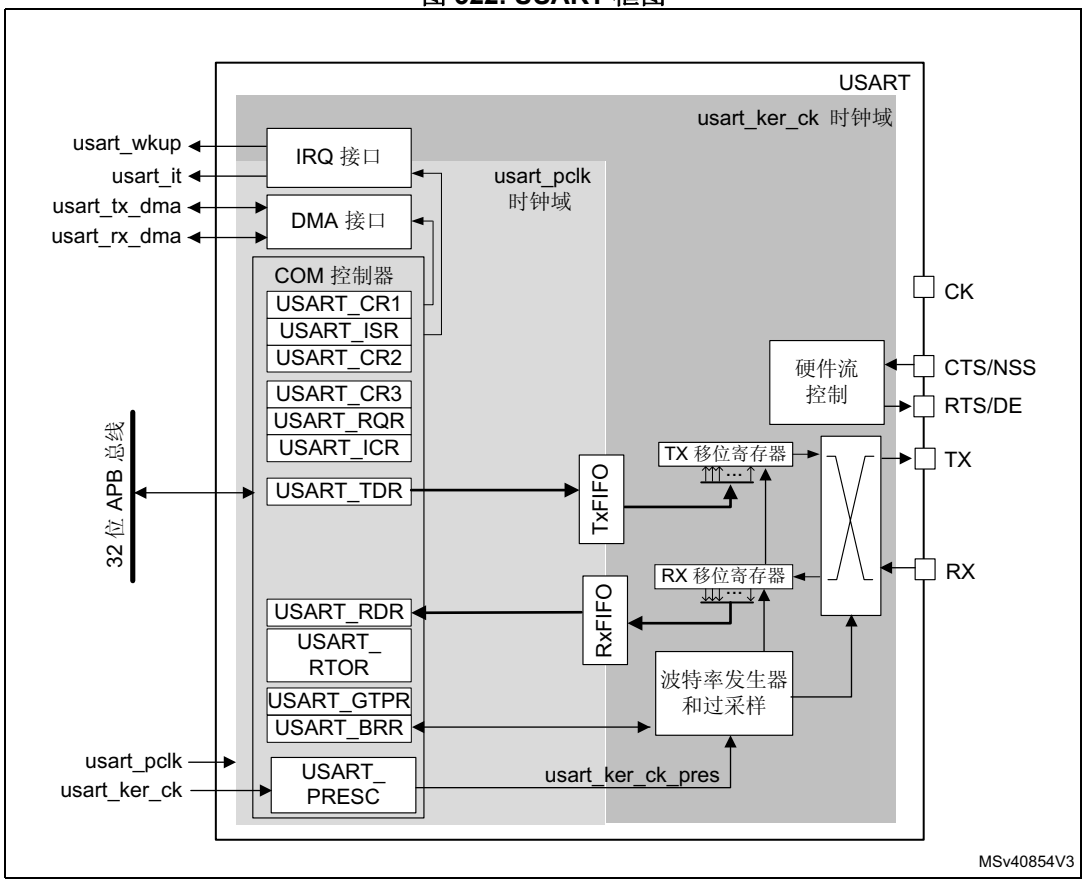


图 322 中的简化框图显示的是两个完全独立的时钟域：

- **usart_pclk** 时钟域

  usart_pclk 时钟信号驱动外设总线接口。需要访问 USART 寄存器时，该信号必须有效。

- **usart_ker_ck** 内核时钟域。

  usart_ker_ck 是 USART 时钟源。它独立于 **usart_pclk**，由 RCC 提供。因此，即使 **usart_ker_ck** 时钟停止，也可以对 USART 寄存器进行读/写操作。

  禁用双时钟域功能时，**usart_ker_ck** 时钟与 **usart_pclk** 时钟相同。

usart_pclk 和 **usart_ker_ck** 之间无任何限制：**usart_ker_ck** 既可快于也可慢于 **usart_pclk**。唯一的限制是软件以足够快的速度管理通信的能力。

USART 工作在 SPI 从模式下时，会使用源自外部 SCLK 信号（由外部主 SPI 器件提供）的串行接口时钟来处理数据流。**usart_ker_ck** 时钟的速度必须至少 3 倍于 CK 输入时钟。

33.5.2 USART 信号

USART 双向通信

USART 双向通信需要至少两个引脚：接收数据输入引脚 (RX) 和发送数据输出引脚 (TX)：

- **RX**（接收数据输入引脚）

  RX 为串行数据输入引脚。采用过采样技术进行数据恢复，即区分有效输入数据和噪声。

- **TX**（发送数据输出引脚）

  如果关闭发送器，该输出引脚模式由其 I/O 端口配置决定。如果使能了发送器但没有需要发送的数据，则 TX 引脚处于高电平。在单线和智能卡模式下，该 I/O 用于发送和接收数据。

RS232 硬件流控制模式

在 RS232 硬件流控制模式下需要以下引脚：

- **CTS**（清除以发送）

  如果驱动为高电平，则该信号用于在当前传输结束时阻止数据发送。

- **RTS**（请求以发送）

  如果为低电平，则该信号用于指示 USART 已准备好接收数据。

RS485 硬件控制模式

在 RS485 硬件控制模式下需要以下引脚：

- **DE** 驱动器使能

  该信号用于激活外部收发器的发送模式。

注： *DE 和 RTS 共用同一个引脚。*

同步主/从模式和智能卡模式

在同步主/从模式和智能卡模式下需要以下引脚：

- CK

  该引脚在同步主模式和智能卡模式下用作时钟输出。

  它在同步从模式下用作时钟输入。

  在同步主模式下，该引脚用于输出发送器数据时钟，对应于 SPI 主模式的同步传输（起始位和结束位上无时钟脉冲，可软件配置是否在最后一个数据位发送时钟脉冲）。同时，RX 引脚可同步接收数据。该机制可用于控制带移位寄存器的外设（如 LCD 驱动器）。时钟相位和极性可通过软件编程。

  在智能卡模式下，CK 输出向智能卡提供时钟。

- NSS

  该引脚在同步从模式下用作从器件选择输入。

注： *NSS 和 CTS 共用同一个引脚。*

33.5.3 USART 字符说明

可通过对 USART_CR1 寄存器中的 M 位（M0：位 12，M1：位 28）进行编程来将字长设置为 7 位、8 位或 9 位（请参见*图 323*）。

- 7 位字符长度：M[1:0] = “10”
- 8 位字符长度：M[1:0] = “00”
- 9 位字符长度：M[1:0] = “01”

注： *7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。*

在默认情况下，信号（TX 或 RX）在起始位工作期间处于低电平状态。在停止位工作期间处于高电平状态。

通过极性配置控制，可以单独针对每个信号对这些值取反。

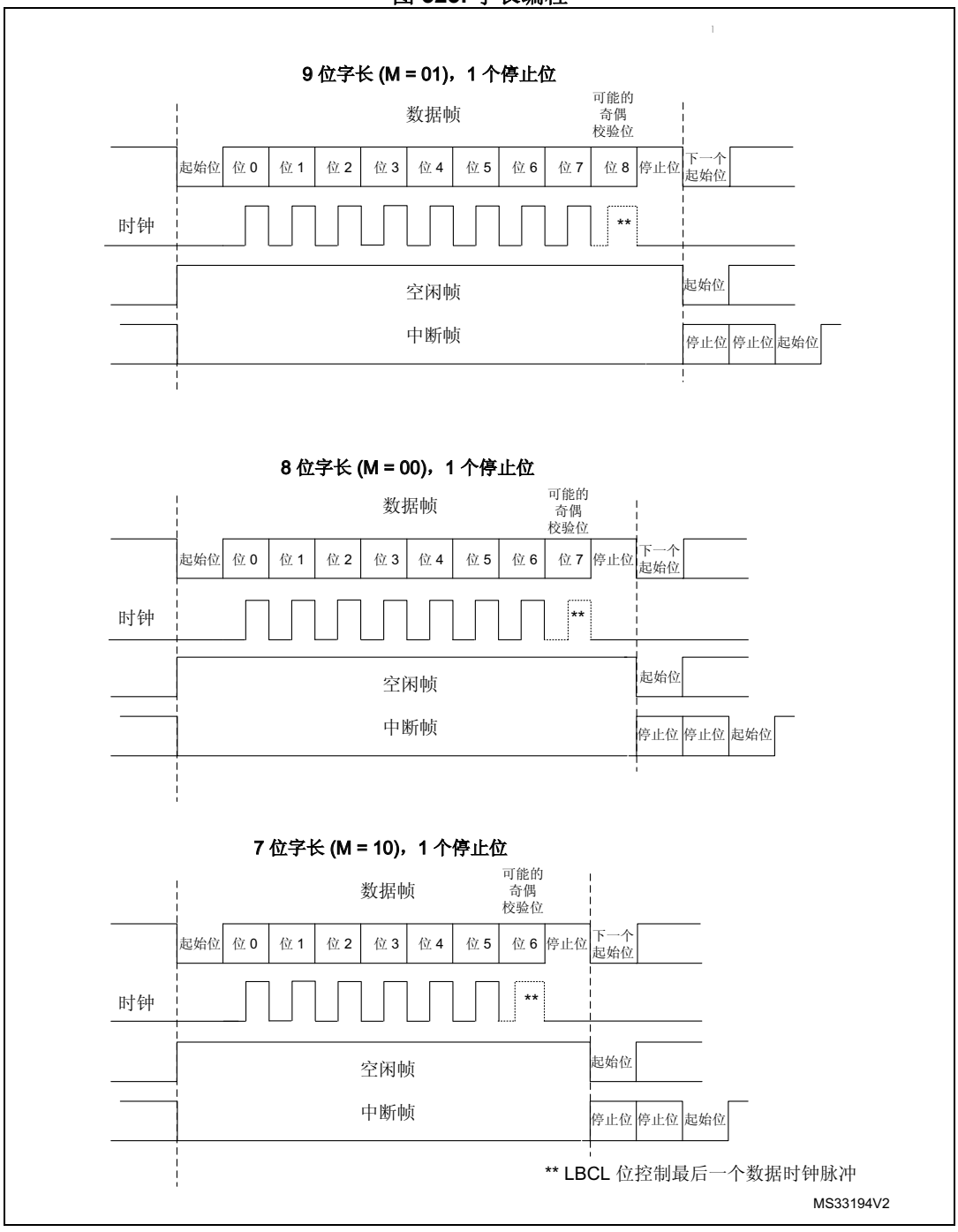
空闲字符可理解为整个帧周期内电平均为“1”（停止位的电平也是“1”）。

停止字符可理解为在一个帧周期内接收到的电平均为“0”。发送器在中断帧的末尾插入 2 个停止位。

发送和接收操作由通用波特率发生器驱动。当发送器和接收器的使能位置 1 时，将分别生成发送时钟和接收时钟。

下面给出了各个块的详细说明。

图 323. 字长编程


9 位字长 (M = 01)，1 个停止位
数据帧
可能的
奇偶
校验位
起始位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
位 7
位 8
停止位
下一个
起始位
时钟
**
空闲帧
起始位
中断帧
停止位
停止位
起始位
8 位字长 (M = 00)，1 个停止位
数据帧
可能的
奇偶
校验位
起始位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
位 7
停止位
下一个
起始位
时钟
**
空闲帧
起始位
中断帧
停止位
停止位
起始位
7 位字长 (M = 10)，1 个停止位
数据帧
可能的
奇偶
校验位
起始位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
停止位
下一个
起始位
时钟
**
空闲帧
起始位
中断帧
停止位
停止位
起始位
** LBCL 位控制最后一个数据时钟脉冲
MS33194V2

33.5.4 USART FIFO 和阈值

USART 可工作在 FIFO 模式下。

USART 具有一个发送 FIFO (TXFIFO) 和一个接收 FIFO (RXFIFO)。可通过将 USART_CR1 寄存器中的 FIFOEN（位 29）置 1 使能 FIFO 模式。仅 UART、SPI 和智能卡模式下支持该模式。

最大数据字长度为 9 位，因此 TXFIFO 为 9 位宽。不过，RXFIFO 的默认宽度为 12 位。这是因为接收器不仅在 FIFO 中存储数据，而且还存储与每个字符相关的错误标志（奇偶校验错误、噪声错误和帧错误标志）。

注： *接收的数据与相应的标志一起存储在 RXFIFO 中，但读取 RDR 时仅读取数据。*

状态标志位于 USART_ISR 寄存器中。

可以配置触发 Tx 和 RX 中断的 TXFIFO 和 RXFIFO 阈值。这些阈值通过 USART_CR3 控制寄存器中的 RXFTCFG 和 TXFTCFG 位域进行编程。

在这种情况下：

- 当 RXFIFO 中接收的数据量达到 RXFTCFG 位域中编程的阈值时，USART_ISR 寄存器中的 RXFT 标志置 1 并生成相应中断（如果使能）。

  这意味着 RXFIFO 被填充，直到 RXFIFO 中的数据量等于编程的阈值。

  已接收到 RXFTCFG 数据：USART_RDR 中有 1 个数据，RXFIFO 中有 (RXFTCFG - 1) 个数据。例如，如果将 RXFTCFG 编程为“101”，则在接收到对应于 FIFO 大小的数据量（RXFIFO 中有（FIFO 大小 - 1）个数据，USART_RDR 中有 1 个数据）时，RXFT 标志将置 1。因此，下一个接收到的数据不会将上溢标志置 1。
- 当 TXFIFO 中空单元数量达到 TXFTCFG 位域中编程的阈值时，USART_ISR 寄存器中的 TXFT 标志置 1 并生成相应中断（如果使能）。

  这意味着 TXFIFO 被清空，直到 TXFIFO 中的空存储单元量等于编程的阈值。

33.5.5 USART 发送器

发送器可发送 7 位、8 位或 9 位的数据字，具体取决于 M 位的状态。要激活发送器功能，必须将发送使能位 (TE) 置 1。发送移位寄存器中的数据在 TX 引脚输出，相应的时钟脉冲在 SCLK 引脚输出。

字符发送

USART 发送期间，首先通过 TX 引脚移出数据的最低有效位（默认配置）。在该模式下，USART_TDR 寄存器的缓冲区 (TDR) 位于内部总线和发送移位寄存器之间。

使能 FIFO 模式时，写入到发送数据寄存器 (USART_TDR) 中的数据会在 TXFIFO 中排队。

每个字符前面都有一个起始位，其对应于一个位周期的逻辑低电平。字符由可配置数量的停止位终止。

停止位的数量可配置为 0.5、1、1.5 或 2。

注： *向 USART_TDR 中写入要发送的数据前，TE 位必须先置 1。*

数据发送期间不应复位 TE 位。发送期间复位 TE 位会冻结波特率计数器，进而损坏 TX 引脚上的数据。当前发送的数据随即丢失。

使能 TE 位时，将发送空闲帧。

可配置的停止位

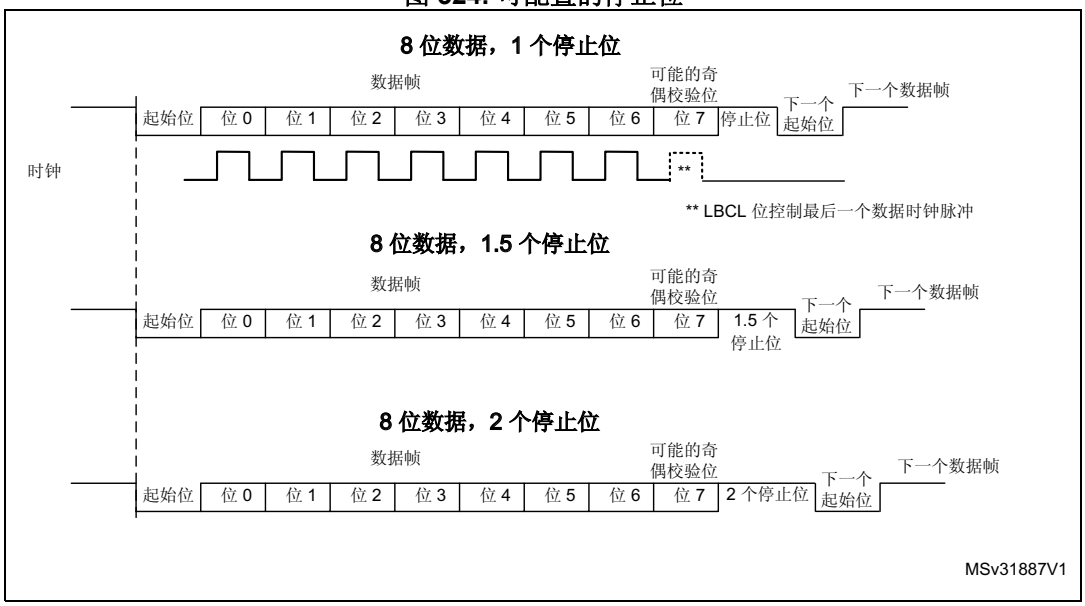
可以在 USART_CR2 的位 13 和位 12 中编程将随各个字符发送的停止位的数量。

- ***1 个停止位***：这是停止位数量的默认值。
- ***2 个停止位***：正常 USART 模式、单线模式和调制解调器模式支持该值。
- ***1.5 个停止位***：用于智能卡模式。

空闲帧发送将包括停止位。

中断发送是 10 个低电平位（M[1:0] = “00”时）、11 个低电平位（M[1:0] = “01”时）或 9 个低电平位（M[1:0] = “10”时），然后是 2 个停止位（请参见*图 324*）。无法传送长中断（中断长度大于 9/10/11 个低电平位）。

图 324. 可配置的停止位



字符发送步骤

要发送字符，需遵循以下步骤：

1. 对 USART_CR1 中的 M 位进行编程以定义字长。
2. 使用 USART_BRR 寄存器选择所需波特率。
3. 对 USART_CR2 中的停止位数量进行编程。
4. 通过向 USART_CR1 寄存器中的 UE 位写入 1 使能 USART。
5. 如果必须进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAT)。按照*第 33.5.10 节：USART 多处理器通信*中的说明配置 DMA 寄存器。
6. 将 USART_CR1 中的 TE 位置 1 以便在首次发送时发送一个空闲帧。
7. 在 USART_TDR 寄存器中写入要发送的数据。为每个要在单缓冲区模式下发送的数据重复这一步骤。
   - 禁止 FIFO 模式时，向 USART_TDR 写入数据会将 TXE 标志清零。
   - 使能 FIFO 模式时，向 USART_TDR 写入数据，数据会添加到 TXFIFO 中。当 TXFNF 标志置 1 时，写 USART_TDR 的操作会被执行。该标志会保持置 1，直到 TXFIFO 已满。
8. 将最后一个数据写入 USART_TDR 寄存器后，等待 TC = 1。
   - 禁止 FIFO 模式时，这表示最后一个帧的发送已完成。
   - 使能 FIFO 模式时，这表示 TXFIFO 和移位寄存器均为空。

   当 USART 被禁止或进入暂停模式时，需要执行此检查来避免损坏最后一次发送。

单字节通信

- 禁止 FIFO 模式时

  对发送数据寄存器进行写操作始终会清零 TXE 位。TXE 标志由硬件置 1。它表示：
  - 数据已从 USART_TDR 寄存器移到移位寄存器中且数据发送已开始；
  - USART_TDR 寄存器为空；
  - USART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。

  TXEIE 位置 1 时该标志位会生成中断。

  发送时，写 USART_TDR 寄存器，会将数据存储在 TDR 缓冲区。该数据随后会在当前发送结束时复制到移位寄存器中。

  未发送时，写 USART_TDR 寄存器，会将数据写入移位寄存器，数据发送开始时，TXE 位置 1。
- 使能 FIFO 模式时，TXFNF（TXFIFO 未满）标志由硬件置 1，以指示：
  - TXFIFO 未满；
  - USART_TDR 寄存器为空；
  - USART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。发送时，对 USART_TDR 寄存器的写操作会将数据存储在 TXFIFO 中。该数据将在当前发送结束时从 TXFIFO 复制到移位寄存器中。

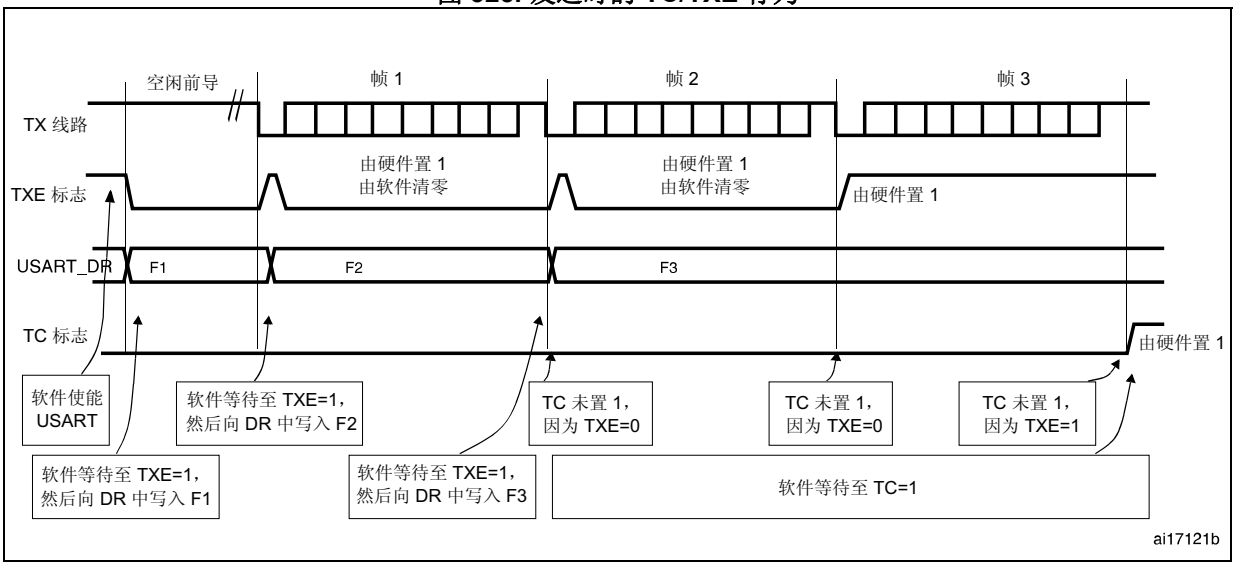
  TXFIFO 未满时，即使在对 USART_TDR 寄存器执行完写操作后，TXFNF 标志也保持为“1”。该标志在 TXFIFO 已满时清零。TXFNFIE 位置 1 时该标志位会生成中断。

  或者，当达到 TXFIFO 阈值时，会生成中断并将数据写入 FIFO。在这种情况下，CPU 可写入程序设定的阈值大小的数据块。

  如果帧已发送（停止位后）且 TXE 标志（FIFO 模式下的 TXFE）置 1，TC 标志将变为高电平。如果 USART_CR1 寄存器中的 TCIE 位置 1，将生成中断。

向 USART_TDR 寄存器中写入最后一个数据后，必须等待至 TC 置 1，之后才可禁止 USART 或使微控制器进入低功耗模式（请参见*图 325：发送时的 TC/TXE 行为*）。

图 325. 发送时的 TC/TXE 行为



注： *使能 FIFO 管理时，TXFNF 标志将用于数据发送。*

中断字符

将 SBKRQ 位置 1 将发送一个中断字符。中断帧的长度取决于 M 位（请参见*图 323*）。

如果将“1”写入 SBKRQ 位，则当前字符发送完成后，将在 TX 线路上发送一个中断字符。通过写操作将 SBKF 位置 1 并在中断字符发送完成时（发送中断字符后的停止位期间），该位由硬件复位。USART 在中断帧末尾的两位持续时间内插入一个逻辑“1”信号 (STOP)，以确保识别下个帧的起始位。

如果 SBKRQ 位置 1，则在当前发送结束时，会发送一个中断字符。

使能 FIFO 模式时，即使 TXFIFO 已满，发送中断字符的优先级也仍高于发送数据的优先级。

空闲字符

将 TE 位置 1 会驱动 USART 在第一个数据帧之前发送一个空闲帧。

33.5.6 USART 接收器

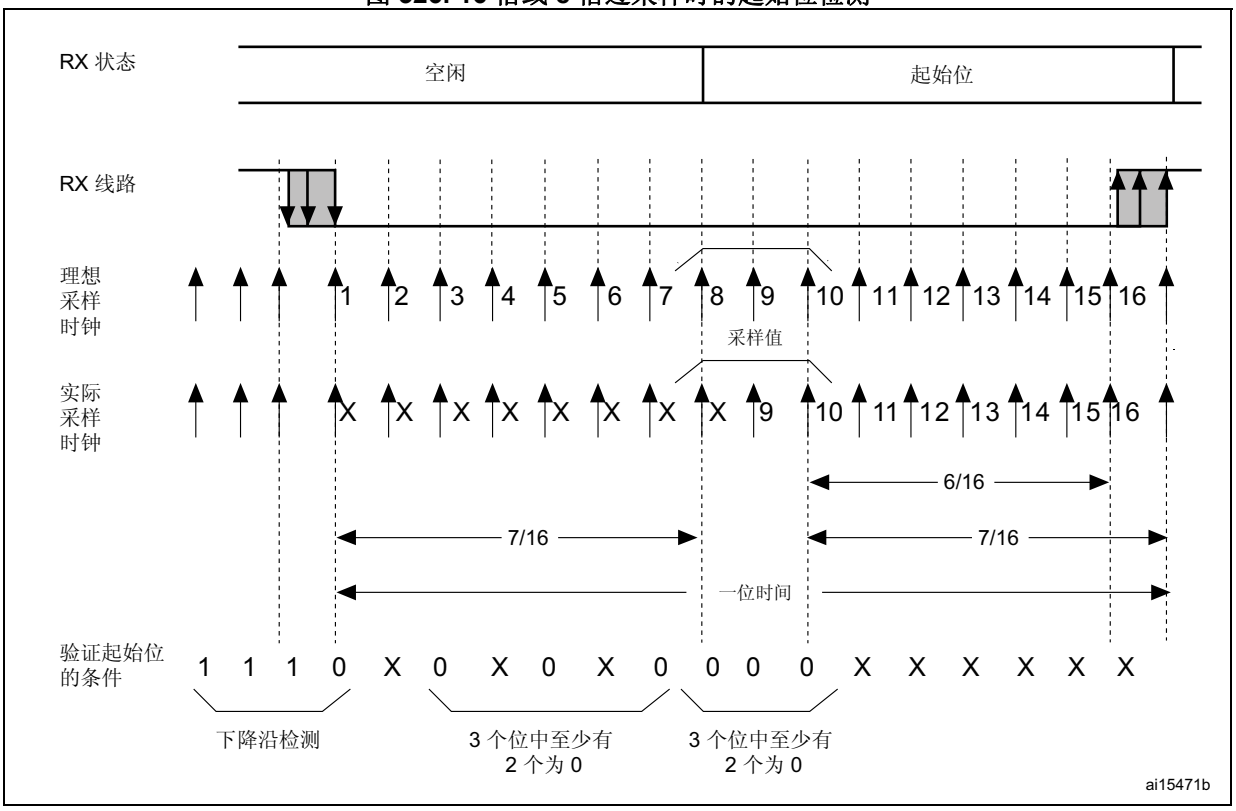
USART 可接收 7 位、8 位或 9 位的数据字，具体取决于 USART_CR1 寄存器中的 M 位。

起始位检测

16 倍或 8 倍过采样时，起始位检测序列相同。

在 USART 中，识别出特定序列的采样时会检测到起始位。此序列为：1 1 1 0 X 0 X 0X 0X 0 X 0X 0。

图 326. 16 倍或 8 倍过采样时的起始位检测



注： *如果序列不完整，起始位检测将中止，接收器将返回空闲状态（无标志位置 1）等待下降沿。*

如果 3 个采样位均为“0”（针对第 3 位、第 5 位和第 7 位进行首次采样时检测到这 3 位均为“0”；针对第 8 位、第 9 位和第 10 位进行第二次采样时仍检测到这 3 位均为“0”），可确认起始位（RXNE 标志置 1 且 RXNEIE = 1 时生成中断；如果使能 FIFO 模式，则 RXFNE 标志置 1 且 RXFNEIE = 1 时生成中断）。

满足以下条件时，可验证起始位但 NE 噪声标志置 1：

a) 对于两次采样，3 个采样位中有 2 位为“0”（针对第 3 位、第 5 位和第 7 位进行采样；针对第 8 位、第 9 位和第 10 位采样）

或

b) 如果其中一次采样时（对第 3 位、第 5 位和第 7 位进行采样或对第 8 位、第 9 位和第 10 位进行采样），3 个采样位中有 2 个为“0”。

如果上述条件均不满足，则启动检测中止，接收器返回空闲状态（无标志置 1）。

字符接收

USART 接收期间，首先通过 RX 引脚移入数据的最低有效位（默认配置）。

字符接收步骤

要接收字符，需遵循以下步骤：

1. 对 USART_CR1 中的 M 位进行编程以定义字长。
2. 使用波特率寄存器 USART_BRR 选择所需波特率。
3. 对 USART_CR2 中的停止位数量进行编程。
4. 通过向 USART_CR1 寄存器中的 UE 位写入“1”使能 USART。
5. 如果将进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAR)。按照*第 33.5.10 节：USART 多处理器通信*中的说明配置 DMA 寄存器。
6. 将 RE 位 USART_CR1 置 1。这一操作将使能接收器开始搜索起始位。

接收到字符时：

- 如果已禁止 FIFO 模式，则 RXNE 位置 1，这表明移位寄存器的内容已传送到 RDR。也就是说，已接收到并可读取数据（及其相应的错误标志）。
- 如果已使能 FIFO 模式，则 RXFNE 位置 1，这表示 RXFIFO 非空。读取 USART_RDR 会返回输入到 RXFIFO 中的最早数据。接收到数据时，数据以及相应的错误位将一起被存储在 RXFIFO 中。
- 如果 RXNEIE（使能 FIFO 模式时为 RXFNEIE）位置 1，则会生成中断。
- 如果接收期间已检测到帧错误、噪声错误、奇偶校验错误或上溢错误，错误标志会置 1。
- 在多缓冲区通信模式下：
  - 如果禁止 FIFO 模式，则 RXNE 标志会在每次接收到字节后置 1。该标志在 DMA 读取接收数据寄存器时被清零。
  - 如果使能 FIFO 模式，则 RXFNE 标志在 RXFIFO 非空时置 1。每次收到 DMA 请求后，都会从 RXFIFO 检索数据。当 RXFIFO 非空时（即，当存在要从 RXFIFO 中读取的数据时），会触发 DMA 请求。
- 在单缓冲区模式下：
  - 如果禁止 FIFO 模式，则通过软件对 USART_RDR 寄存器进行读操作来将 RXNE 标志清零。也可以通过将 USART_RQR 寄存器中的 RXFRQ 位编程为“1”来清零 RXNE 标志。RXNE 标志必须在结束接收下一个字符前清零，以避免发生上溢错误。
  - 如果使能 FIFO 模式，则 RXFNE 在 RXFIFO 非空时置 1。每次对 USART_RDR 执行完读操作后，都会从 RXFIFO 中检索数据。RXFIFO 为空时，RXFNE 标志将清零。也可以通过将 USART_RQR 中的 RXFRQ 位编程为“1”来清零 RXFNE 标志。RXFIFO 已满时，必须在结束接收下一个字符前读取 RXFIFO 中的第一个条目，以避免发生上溢错误。当 RXFNEIE 位置 1 时，RXFNE 标志会生成中断。或者，当达到 RXFIFO 阈值时，会生成中断并从 RXFIFO 中读取数据。在这种情况下，CPU 可读取由编程的阈值定义的数据块。

中断字符

接收到中断字符时，USART 将会按照帧错误对其进行处理。

空闲字符

检测到空闲帧时，除了在 IDLEIE 位置 1 时会生成中断外，处理步骤与接收到数据字符的情况相同。

上溢错误

- 禁止 FIFO 模式

  如果在 RXNE 未复位时接收到字符，则会发生上溢错误。

  RXNE 位清零前，数据无法从移位寄存器传送到 RDR 寄存器。每接收到一个字节后，RXNE 标志都将置 1。

  当 RXNE 标志位置 1 时，如果在接收到下一个数据或尚未处理上一个 DMA 请求，则会发生上溢错误。发生上溢错误时：

  – ORE 位置 1。
  – RDR 中的内容不会丢失。可通过读取 USART_RDR 寄存器获得之前的数据。
  – 移位寄存器将被覆盖。之后，上溢期间接收到的任何数据都将丢失。
  – 如果 RXNEIE 或 EIE 位置 1，则会生成中断。

- 使能 FIFO 模式

  移位寄存器准备好传送并且接收 FIFO 已满时，会发生上溢错误。

  在 RXFIFO 中出现一个空闲位置之前，数据无法从移位寄存器传送到 USART_RDR 寄存器。当 RXFIFO 非空时，RXFNE 标志置 1。

  如果 RXFIFO 已满且移位寄存器已准备好传送，则会发生上溢错误。发生上溢错误时：

  – ORE 位置 1。
  – RXFIFO 中的第一个条目不会丢失。通过读取 USART_RDR 寄存器可获得此条目。
  – 移位寄存器将被覆盖。之后，上溢期间接收到的任何数据都将丢失。
  – 如果 RXFNEIE 或 EIE 位置 1，则会生成中断。

通过将 USART_ICR 寄存器中的 ORECF 位置 1 来复位 ORE 位。

注： *ORE 位置 1 时表示至少 1 个数据丢失。*

禁止 FIFO 模式时，有以下两种可能

- *如果 RXNE = 1，则最后一个有效数据存储于接收寄存器 (RDR) 中并且可进行读取，*
- *如果 RXNE = 0，则最后一个有效数据已被读取，因此 RDR 寄存器中没有要读取的数据。接收到新（丢失）数据的同时已读取 RDR 寄存器中的最后一个有效数据时，会发生该情况。*

选择时钟源和合适的过采样方法

通过时钟控制系统选择时钟源（请参见复位和时钟控制 (RCC) 部分）。在使能 USART 之前，必须通过 UE 位选择时钟源。

必须遵循以下两个条件选择时钟源：

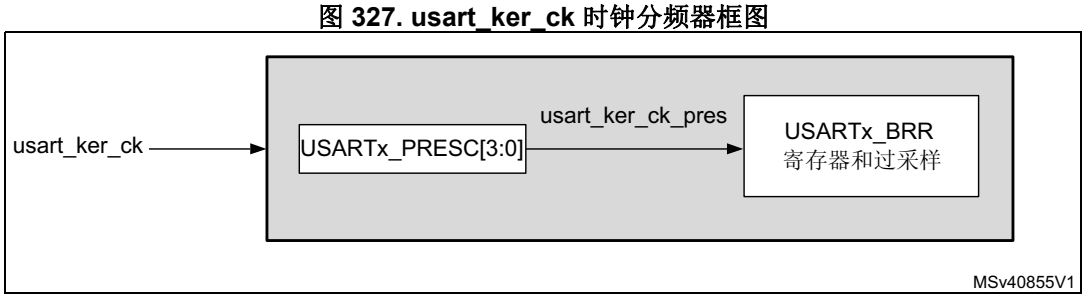
- 可在低功耗模式下使用 USART。
- 通信速度。

时钟源频率为 usart_ker_ck。

如果支持双时钟域和从低功耗模式唤醒功能，则 usart_ker_ck 时钟源可在 RCC 中进行配置（请参见复位和时钟控制 (RCC) 部分）。否则，usart_ker_ck 时钟与 usart_pclk 相同。

usart_ker_ck 时钟可根据可编程系数（在 USART_PRESC 寄存器中定义）进行分频。

图 327. usart_ker_ck 时钟分频器框图



某些 usart_ker_ck 时钟源允许 USART 在 MCU 处于低功耗模式时接收数据。需要时，USART 可基于所接收的数据和选择的唤醒模式来唤醒 MCU，以通过用软件读取 USART_RDR 寄存器的方式或通过 DMA 的方式传输已接收数据。

对于其他时钟源，系统必须在正常工作模式才能进行 USART 通信。

时钟源还决定通信速度范围（尤其是最大通信速度）。

接收器采用不同的用户可配置过采样技术（除了同步模式下），进行数据恢复，可以区分有效数据与噪声。这可在最大通信速度与抗噪声/时钟误差性能之间实现最佳平衡。

可通过编程 USART_CR1 寄存器中的 OVER8 位来选择采样方法，且采样时钟可以是波特率时钟的 16 倍或 8 倍（请参见*图 328* 和*图 329*）。

根据应用：

- 选择 8 倍过采样（OVER8 = 1）以获得更高的速度（高达 usart_ker_ck_pres/8）。这种情况下接收器对时钟偏差的最大容差将会降低（请参见 *第 1013 页的第 33.5.8 节：USART 接收器对时钟偏差的容差*）
- 选择 16 倍过采样 (OVER8=0) 以增加接收器对时钟偏差的容差。在这种情况下，最大速度被限制为最大 usart_ker_ck_pres/16（其中，usart_ker_ck_pres 为 USART 输入时钟通过预分频器进行分频得到的值）。

可通过编程 USART_CR3 寄存器中的 ONEBIT 位选择用于评估逻辑电平的方法。有两种选择可供使用：

- 在已接收位的中心进行三次采样，从而进行多数表决。这种情况下，如果用于多数表决的 3 次采样结果不相等，NE 位置 1。
- 在已接收位的中心进行单次采样

  根据应用：

  – 在噪声环境下工作时，请选择三次采样的多数表决法 (ONEBIT = 0)；在检测到噪声时请拒绝数据（请参见*图 193*），因为这表示采样过程中产生了干扰。

  – 线路无噪声时请选择单次采样法 (ONEBIT=1) 以增加接收器对时钟偏差的容差（请参见*第 1013 页的第 33.5.8 节：USART 接收器对时钟偏差的容差*）。这种情况下 NE 位始终不会置 1。

帧中检测到噪声时：

- 在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）的上升沿时 NE 位置 1。
- 无效数据从移位寄存器传送到 USART_RDR 寄存器。
- 单字节通信时无中断产生。然而，在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）生成中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时将发出中断。

通过将 USART_ICR 寄存器中的 NECF 位置 1 复位 NE 位。

注： *SPI 模式不支持噪声错误。*

智能卡、IrDA 和 LIN 模式下不可采用 8 倍过采样。在这些模式下，OVER8 位由硬件强制清零。

图 328. 16 倍过采样时的数据采样

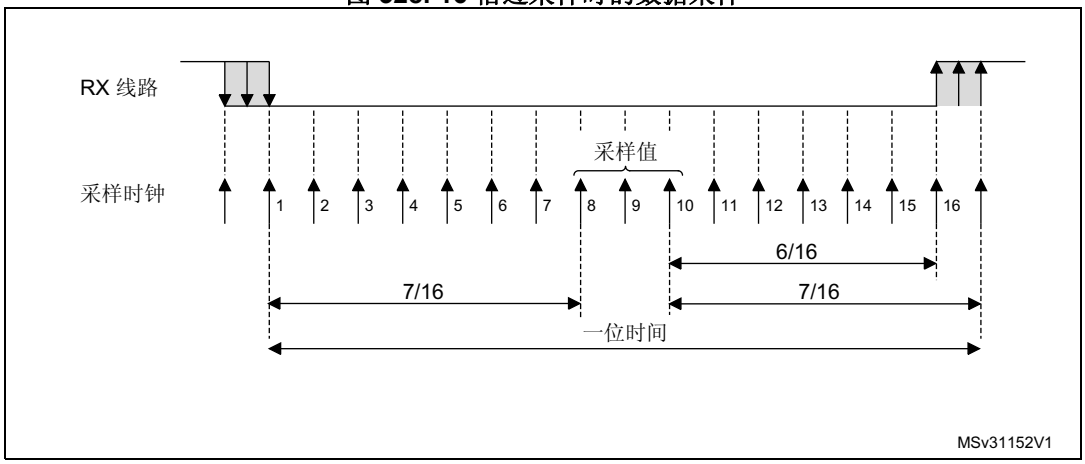


图 329. 8 倍过采样时的数据采样

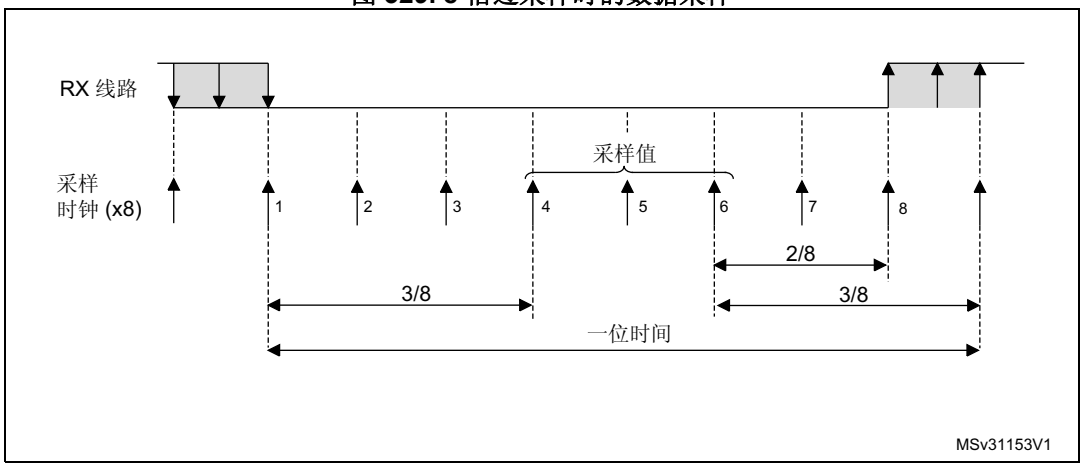


表 193. 通过采样数据进行噪声检测

采样值	NE 状态	接收的位值
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

帧错误

如果接收数据时未在预期时间内识别出停止位，从而出现同步失效或过度的噪声，则会检测到帧错误。

检测到帧错误时：

- FE 位由硬件置 1。
- 无效数据从移位寄存器传送到 USART_RDR 寄存器（使能 FIFO 模式时为 RXFIFO）。
- 单字节通信时无中断产生。然而，在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）生成中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时将发出中断。

通过将“1”写入 USART_ICR 寄存器中的 FECF 位来复位 FE 位。

注： *SPI 模式不支持帧错误。*

接收期间可配置的停止位

可通过 USART_CR 的控制位配置要接收的停止位的数量：可以是 1 或 2 个（正常模式下），也可以是 0.5 或 1.5 个（智能卡模式下）。

- **0.5 个停止位（在智能卡模式下接收时）**：不会对 0.5 个停止位进行采样。结果，选择 0.5 个停止位时，无法检测到帧错误和中断帧。
- **1 个停止位**：将在第 8、第 9 和第 10 次采样时对 1 个停止位进行采样。
- 1.5 个停止位（在智能卡模式下）

  在智能卡模式下发送时，设备必须检查数据是否正确发送。因此，必须使能接收器块（USART_CR1 中的 RE = 1）并检查停止位，以测试智能卡是否已检测到奇偶校验错误。

  发生奇偶校验错误时，智能卡会在采样时将数据信号强制为低电平（即 NACK 信号），该信号被标记为帧错误。之后，FE 标志在 1.5 个停止位的末尾由 RXNE 标志（使能 FIFO 模式时为 RXFNE）置 1。在第 16、第 17 和第 18 次采样时对 1.5 个停止位进行采样（停止位采样开始后维持 1 个波特时钟周期）。1.5 个停止位可分为 2 个部分：0.5 个波特时钟周期（未发生任何动作），然后是 1 个正常的停止位周期（一半时间处进行采样）（更多详细信息，请参见*第 1025 页的第 33.5.16 节：USART 接收器超时*）。
- 2 个停止位

  采样 2 个停止位时在第 8、第 9 和第 10 次采样时对第一个停止位进行采样。

  如果在第一个停止位期间检测到帧错误，则帧错误标志会置 1。

  发生帧错误时不检测第 2 个停止位。RXNE 标志（使能 FIFO 模式时为 RXFNE）将在第一个停止位结束时置 1。

33.5.7 USART 波特率生成

接收器和发送器（Rx 和 Tx）的波特率均设置为 USART_BRR 寄存器中编程的值。

公式 1：适用于标准 USART（包括 SPI 模式）的波特率（OVER8 = “0”或“1”）

在 16 倍过采样的情况下，波特率通过以下公式得出：

$$\text{Tx/Rx 波特} = \frac{\text{usart\_ker\_ckpres}}{\text{USARTDIV}}$$

在 8 倍过采样的情况下，波特率通过以下公式得出：

$$\text{Tx/Rx 波特} = \frac{2 \times \text{usart\_ker\_ckpres}}{\text{USARTDIV}}$$

公式 2：智能卡、LIN 和 IrDA 模式下的波特率 (OVER8 = 0)

波特率可根据以下公式得出：

$$\text{Tx/Rx 波特} = \frac{\text{usart\_ker\_ckpres}}{\text{USARTDIV}}$$

USARTDIV 是一个存放在 USART_BRR 寄存器中的无符号定点数。

- 当 OVER8 = 0 时，BRR = USARTDIV。
- 当 OVER8 = 1 时
  - BRR[2:0] = USARTDIV[3:0]，右移 1 位。
  - BRR[3] 必须保持清零。
  - BRR[15:4] = USARTDIV[15:4]

注： *对 USART_BRR 执行写操作后，波特率计数器更新为波特率寄存器中的新值。因此，波特率寄存器的值不应在通信时发生更改。*

16 倍和 8 倍过采样时，USARTDIV 必须大于或等于 16。

如何从 USART_BRR 寄存器中获取 USARTDIV

示例 1

要通过 usart_ker_ck_pres = 8 MHz 获得 9600 波特：

- 16 倍过采样时：
  USARTDIV = 8 000 000/9600
  BRR = USARTDIV = 833d = 0341h
- 8 倍过采样时：
  USARTDIV = 2 * 8 000 000/9600
  USARTDIV = 1666,66 (1667d = 683h)
  BRR[3:0] = 3h >> 1 = 1h
  BRR = 0x681

示例 2

要通过 usart_ker_ck_pres = 48 MHz 获得 921.6 K 波特：

- 16 倍过采样时：
  USARTDIV = 48 000 000/921 600
  BRR = USARTDIV = 52d = 34h
- 8 倍过采样时：
  USARTDIV = 2 * 48 000 000/921 600
  USARTDIV = 104 (104d = 68h)
  BRR[3:0] = USARTDIV[3:0] >> 1 = 8h >> 1 = 4h
  BRR = 0x64

33.5.8 USART 接收器对时钟偏差的容差

仅当总时钟系统偏差小于 USART 接收器的容差时，USART 异步接收器才能正常工作。

影响总偏差的因素包括：

- DTRA：发送器误差引起的偏差（其中还包括发送器本地振荡器的偏差）
- DQUANT：接收器的波特率量化引起的误差
- DREC：接收器本地振荡器的偏差
- DTCL：传输线路引起的偏差（通常是由于收发器所引起，它可能会在低电平到高电平转换时序与高电平到低电平转换时序之间引入不对称）

$$DTRA + DQUANT + DREC + DTCL + DWU < USART\ 接收器偏差$$

其中

DWU 是使用从低功耗模式唤醒时因采样点偏差而产生的误差。

M[1:0] = 01 时：

$$DWU = \frac{t_{WUUSART}}{11 \times Tbit}$$

M[1:0] = 00 时：

$$DWU = \frac{t_{WUUSART}}{10 \times Tbit}$$

M[1:0] = 10 时：

$$DWU = \frac{t_{WUUSART}}{9 \times Tbit}$$

$t_{WUUSART}$ 是检测到起始位下降沿与时钟（由外设请求）就绪、达到外设且调压器就绪之间的时间。

USART 接收器在*表 194* 和*表 195* 中指定的最大容许偏差下可正确接收数据，具体取决于以下设置：

- 由 USART_CR1 寄存器中的 M 位定义的 9 位、10 位或 11 位字符长度。
- 由 USART_CR1 寄存器中的 OVER8 位定义的 8 倍或 16 倍过采样。
- USART_BRR 寄存器的 BRR[3:0] 位等于或不等于 0000。
- 使用 1 位或 3 位对数据进行采样，取决于 USART_CR3 寄存器中 ONEBIT 位的值。

表 194. BRR [3:0] = 0000 时的 USART 接收器容差

M 位	OVER8 位 = 0		OVER8 位 = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
00	3.75%	4.375%	2.50%	3.75%
01	3.41%	3.97%	2.27%	3.41%
10	4.16%	4.86%	2.77%	4.16%

表 195. BRR[3:0] 不等于 0000 时的 USART 接收器容差

M 位	OVER8 位 = 0		OVER8 位 = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
00	3.33%	3.88%	2%	3%
01	3.03%	3.53%	1.82%	2.73%
10	3.7%	4.31%	2.22%	3.33%

注： *当接收的帧恰好包含 10 个（M 位 = 00）、11 个（M 位 = 01）或 9 个（M 位 = 10）位时间的空闲帧时，表 194 和表 195 中指定的数据可能与特例中的数据略微不同。*

33.5.9 USART 自动波特率检测

USART 可根据接收一个字符检测并自动设置 USART_BRR 寄存器的值。自动波特率检测在以下两种情况下非常有用：

- 事先不知道系统的通信速度。
- 系统正在使用精确度相对较低的时钟源且该机制允许在不测量时钟偏差的情况下获得正确的波特率。

时钟源频率必须与预期通信速度兼容。

- 16 倍过采样时，波特率范围为 usart_ker_ck_pres/65535 到 usart_ker_ck_pres/16。
- 8 倍过采样时，波特率范围为 usart_ker_ck_pres/65535 到 usart_ker_ck_pres/8。

在激活自动波特率检测之前，必须通过 USART_CR2 寄存器中的 ABRMOD[1:0] 字段选择自动波特率检测模式。根据不同的字符模式，存在四种检测模式。在这些自动波特率模式下，波特率在同步接收数据期间被多次测量，每次测量的结果都与前一次进行比较。

这些模式如下：

- **模式 0**：以“1”位开头的任意字符。

  这种情况下，USART 会测量起始位的持续时间（下降沿到上升沿）。
- **模式 1**：以 10xx 位模式开头的任意字符。

  这种情况下，USART 会测量起始位和第一个数据位的持续时间。测量在下降沿到下降沿期间完成，可在信号斜率较小时确保较高的精度。
- **模式 2**：0x7F 字符帧（可以是 LSB 在前模式下的 0x7F 字符，也可以是 MSB 在前模式下的 0xFE 字符）。

  这种情况下，先在起始位结束时更新波特率 (BR)，然后在位 6 结束时更新波特率（根据从下降沿到下降沿执行的测量：BR6）。以 BR 对位 0 到位 6 进行采样，而以 BR6 对字符的其它位进行采样。
- **模式 3**：0x55 字符帧。

  这种情况下，先在起始位结束时更新波特率 (BR)，然后在位 0 结束时更新波特率（根据从下降沿到下降沿执行的测量：BR0），最后在位 6 结束时更新波特率 (BR6)。以 BR 对位 0 进行采样，以 BR0 对位 1 到位 6 进行采样，以 BR6 对字符的其它位进行采样。同时，对 RX 线路的各个中间转换执行其他检查。如果 RX 上的转换与接收器（基于根据位 0 计算的波特率的接收器）未充分同步，则生成错误。

激活自动波特率检测之前，必须先通过向 USART_BRR 寄存器写入非零的波特率值来初始化该寄存器。

通过将 USART_CR2 寄存器中的 ABREN 位置 1 来激活自动波特率检测。之后 USART 将等待 RX 线路上的第一个字符。通过将 USART_ISR 寄存器中的 ABRF 标志置 1 来指示自动波特率操作完成。如果线路干扰较大，则无法保证正确的波特率检测。这种情况下，BRR 值可能会损坏，ABRE 错误标志位将置 1。如果通信速度不在自动波特率检测范围（位持续时间不在 16 个和 65536 个时钟周期（16 倍过采样时）之间，也不在 8 个和 65536 个时钟周期（8 倍过采样时）之间）内，也会出现这种情况。

稍后，可通过复位 ABRF 标志（通过写入“0”）重新启动自动波特率检测。

禁止 FIFO 管理且发生自动波特率错误时，ABRE 标志通过 RXNE 和 FE 位置 1。

使能 FIFO 管理且发生自动波特率错误时，ABRE 标志通过 RXFNE 和 FE 位置 1。

如果使能 FIFO 模式，则应使用第一个 RXFIFO 位置上的数据进行自动波特率检测。因此，在启动自动波特率检测之前，请通过检查 USART_ISR 寄存器的 RXFNE 标志来确保 RXFIFO 为空。

注： *如果在自动波特率操作期间禁止 USART (UE=0)，则可能损坏 BRR 值。*

33.5.10 USART 多处理器通信

可以执行 USART 多处理器通信（多个 USART 连接在一个网络中）。例如，其中一个 USART 可以是主 USART，其 TX 输出与其他 USART 的 RX 输入相连；而其他 USART 为从 USART，其各自的 TX 输出在逻辑上通过与运算连在一起，并与主 USART 的 RX 输入相连。

在多处理器配置中，理想情况下通常只有预期的消息接收方主动接收完整的消息内容，从而减少由所有未被寻址的接收器造成的冗余 USART 服务开销。

可通过静默功能将未被寻址的器件置于静默模式下。为了使用静默模式功能，必须将 USART_CR1 寄存器中的 MME 位置 1。

注： *使能 FIFO 管理且 MME 已置 1 时，不得清零 MME 位再快速将其置 1（在两个 usart_ker_ck 周期内），否则静默模式可能保持有效。*

使能静默模式时：

- 不得将接收状态位置 1。
- 禁止任何接收中断。
- USART_ISR 寄存器中的 RWU 位置“1”。在某些情况下，RWU 可以由硬件或软件通过 USART_RQR 寄存器中的 MMRQ 位自动控制。

根据 USART_CR1 寄存器中 WAKE 位的设置，USART 可使用以下两种方法进入或退出静音模式：

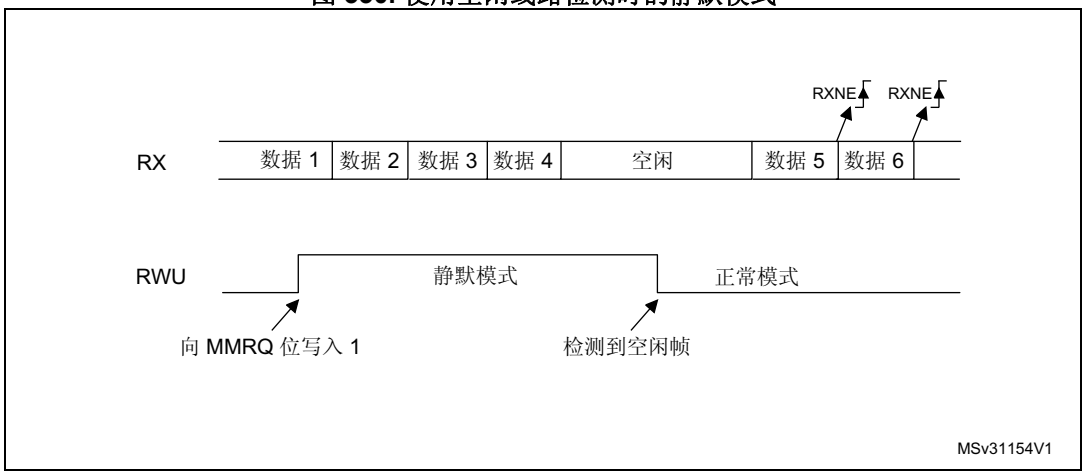
- 如果 WAKE 位被复位，则进行空闲线路检测。
- 如果 WAKE 位置 1，则进行地址标记检测。

空闲线路检测 (WAKE=0)

向 MMRQ 位写入“1”且 RWU 位自动置 1 时，USART 进入静默模式。

检测到空闲帧时，USART 将唤醒。此时 RWU 位会由硬件清零，但 USART_ISR 寄存器中的 IDLE 位不会置 1。*图 330* 中给出了使用空闲线路检测时静默模式行为的示例。

图 330. 使用空闲线路检测时的静默模式



注： *如果在 IDLE 字符已经过去时将 MMRQ 位置 1，将不会进入静默模式（RWU 未置 1）。*

如果在线路处于空闲状态时激活 USART，在一个 IDLE 帧持续时间后（不只在接收一个字符帧后）会检测到空闲状态。

4 位/7 位地址标记检测 (WAKE=1)

在此模式下，如果字节的 MSB 为 1，则将这些字节识别为地址，否则将其识别为数据。在地址字节中，目标接收器的地址位于 4 个或 7 个 LSB 中。7 位或 4 位地址检测通过 ADDM7 位来选择。接收器会将此 4 位/7 位字与其地址进行比较，该接收器的地址在 USART_CR2 寄存器的 ADD 位中进行设置。

注： *在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。*

当接收到与其编程地址不匹配的地址字符时，USART 会进入静默模式。此时，RWU 位将由硬件置 1。USART 进入静默模式后，RXNE 标志不会针对此地址字节置 1，也不会发出中断或 DMA 请求。使能 FIFO 管理时，软件应确保在进入静默模式之前 RXFIFO 中至少有一个空位置。

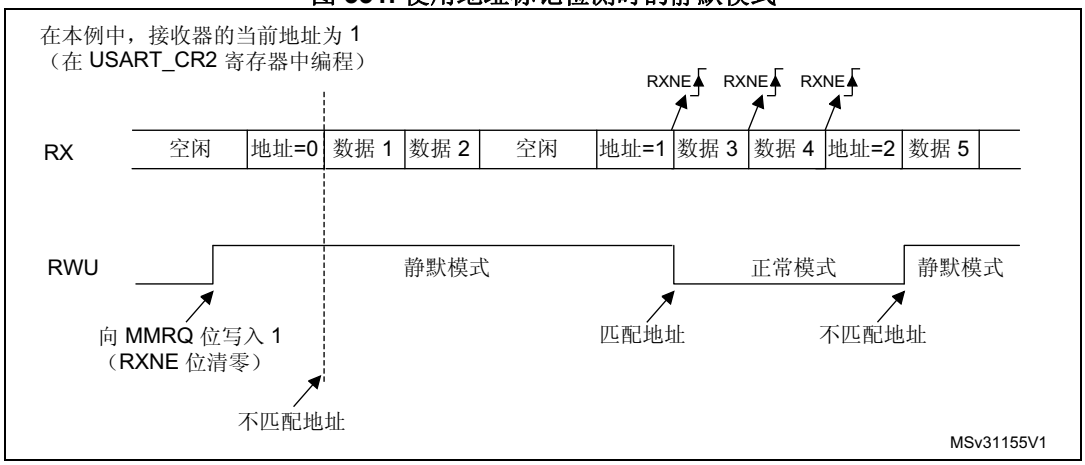
当向 MMRQ 位写入 1 时，USART 也会进入静默模式。这种情况下，RWU 位也自动置 1。

当接收到与编程地址匹配的地址字符时，USART 会退出静默模式。然后 RWU 位被清零，可以开始正常接收后续字节。由于 RWU 位已清零，RXNE/RXFNE 位会针对地址字符置 1。

注： *使能 FIFO 管理时，如果在接收器对数据的最后一位进行采样时 MMRQ 置 1，则可在有效进入静默模式之前接收该数据*

图 331 中给出了使用地址标记检测时静默模式行为的示例。

图 331. 使用地址标记检测时的静默模式



33.5.11 USART Modbus 通信

USART 为 Modbus/RTU 和 Modbus/ASCII 协议的实现提供基本支持。Modbus/RTU 是一个半双工块传输协议。该协议的控制部分（地址识别、块完整性控制和命令解析）必须用软件实现。

USART 为块结束检测提供基本支持，无需软件开销或其它资源。

Modbus/RTU

在此模式下，一个块的结束通过超过 2 个字符时间的“静默”（空闲线路）来识别。此功能通过可编程的超时功能实现。

超时功能和中断必须分别通过 USART_CR2 寄存器中的 RTOEN 位和 USART_CR1 寄存器中的 RTOIE 位激活。与 2 个字符时间（例如 22 个位时间）的超时相对应的值必须在 RTO 寄存器中编程。如果在此期间接收线路空闲，则在接收到最后一个停止位后，将生成中断，同时通知软件当前块接收已完成。

Modbus/ASCII

在此模式下，块结束通过特定 (CR/LF) 字符序列识别。USART 通过字符匹配功能管理此机制。

通过在 ADD[7:0] 字段中编程 LF ASCII 码以及激活字符匹配中断 (CMIE=1)，软件可在接收到 LF 时获得通知并检查 DMA 缓存区中的 CR/LF。

33.5.12 USART 极性控制

将 USART_CR1 寄存器中的 PCE 位置 1，可以使能奇偶校验控制（发送时生成奇偶校验位，接收时进行奇偶校验检查）。根据 M 位定义的帧长度，*表 196* 中列出了可能的 USART 帧格式。

表 196. USART 帧格式

M 位	PCE 位	USART 帧(1)
00	0	| SB | 8 位数据 | STB |
00	1	| SB | 7 位数据 | PB | STB |
01	0	| SB | 9 位数据 | STB |
01	1	| SB | 8 位数据 PB | STB |
10	0	| SB | 7 位数据 | STB |
10	1	| SB | 6 位数据 | PB | STB |

1. 图注：SB：起始位，STB：停止位，PB：奇偶校验位。在数据寄存器中，PB 始终位于 MSB 位置（第 8 位或第 7 位，具体取决于 M 位的值）。

偶校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为偶数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据=00110101 且 4 个位置 1，则在选择偶校验（USART_CR1 寄存器中的 PS 位 = 0）时，校验位为 0。

奇校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为奇数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据=00110101 且 4 个位置 1，则在选择奇校验（USART_CR1 寄存器中的 PS 位 = 1）时，校验位为 1。

接收时进行奇偶校验检查

如果奇偶校验检查失败，则 USART_ISR 寄存器中的 PE 标志置 1；如果 USART_CR1 寄存器中 PEIE 位置 1，则会生成中断。通过软件将 1 写入 USART_ICR 寄存器中的 PECF 位来清零 PE 标志。

发送时的奇偶校验生成

如果 USART_CR1 寄存器中的 PCE 位置 1，则在数据寄存器中所写入数据的 MSB 位会进行传送，但是会由奇偶校验位进行更改（如果选择偶校验 (PS=0)，则“1”的数量为偶数；如果选择奇校验 (PS=1)，则“1”的数量为奇数）。

33.5.13 USART LIN（局域互连网络）模式

仅在支持 LIN 模式时才与本节相关。请参见*第 996 页的第 33.4 节：USART 实现*。

通过将 USART_CR2 寄存器中的 LINEN 位置 1 来选择 LIN 模式。在 LIN 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 CLKEN 位。
- USART_CR3 寄存器中的 STOP[1:0]、SCEN、HDSEL 和 IREN 位。

LIN 发送

与正常的 USART 发送相比，在 LIN 主器件中发送时必须采用*第 33.5.4 节*中介绍的步骤，同时还具有以下区别：

- M 位清零以配置 8 位字长度。
- LINEN 位置 1 以进入 LIN 模式。此时，将 SBKRQ 位置 1 会发送 13 个“0”位作为中断字符。然后会发送值为“1”的 2 个位以进行下一启动检测。

LIN 接收

使能 LIN 模式后，将激活中断检测电路。该检测完全独立于正常的 USART 接收器。在空闲状态或某个帧期间，只要发生中断即可检测出来。

接收器（USART_CR1 寄存器中 RE=1）使能后，电路便开始监测 RX 输入上的启动信号。检测起始位的方法与搜索中断字符或数据的方法相同。检测到起始位后，电路会对接下来的位进行采样，方法与数据采样相同（第 8、第 9 和第 10 次采样）。如果 10 个（USART_CR2 中 LBDL = 0 时）或 11 个（USART_CR2 中 LBDL=1 时）连续位均检测为“0”，且其后跟随分隔符，则 USART_ISR 寄存器中的 LBDF 标志将置 1。如果 LBDIE 位=1，则会生成中断。在验证中断前，会对分隔符进行检查，因为它表示 RX 线路已恢复到高电平。

如果在第 10 或第 11 次采样前已对“1”采样，则中断检测电路会取消当前检测，并重新搜索起始位。

如果禁止 LIN 模式 (LINEN=0)，接收器会作为正常的 USART 继续工作，不会再进行断路检测。

如果使能 LIN 模式 (LINEN=1)，只要发生帧错误（例如，在“0”处检测到停止位，这种情况可能出现在任何断路帧中），接收器即会停止，直到断路检测电路接收到“1”（断路字不完整时）或接收到分隔符（检测到断路时）为止。

*第 1020 页的图 332：LIN 模式下的中断检测（11 位中断长度——LBDL 位置 1）*中显示了中断检测器状态机和中断标志的行为。

*第 1021 页的图 333：LIN 模式下的中断检测与帧错误检测*中列出了中断帧的示例。

图 332. LIN 模式下的中断检测（11 位中断长度——LBDL 位置 1）

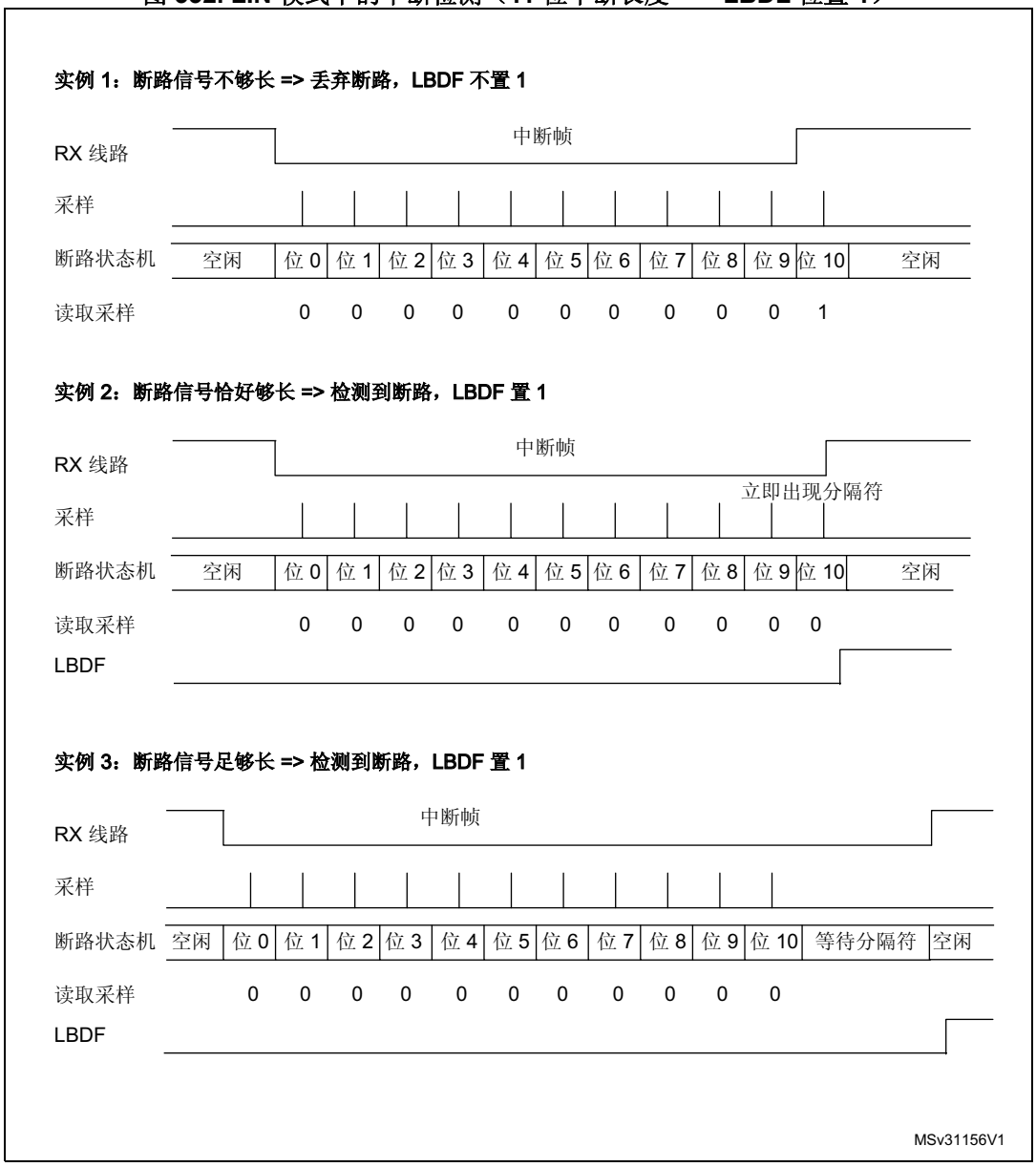
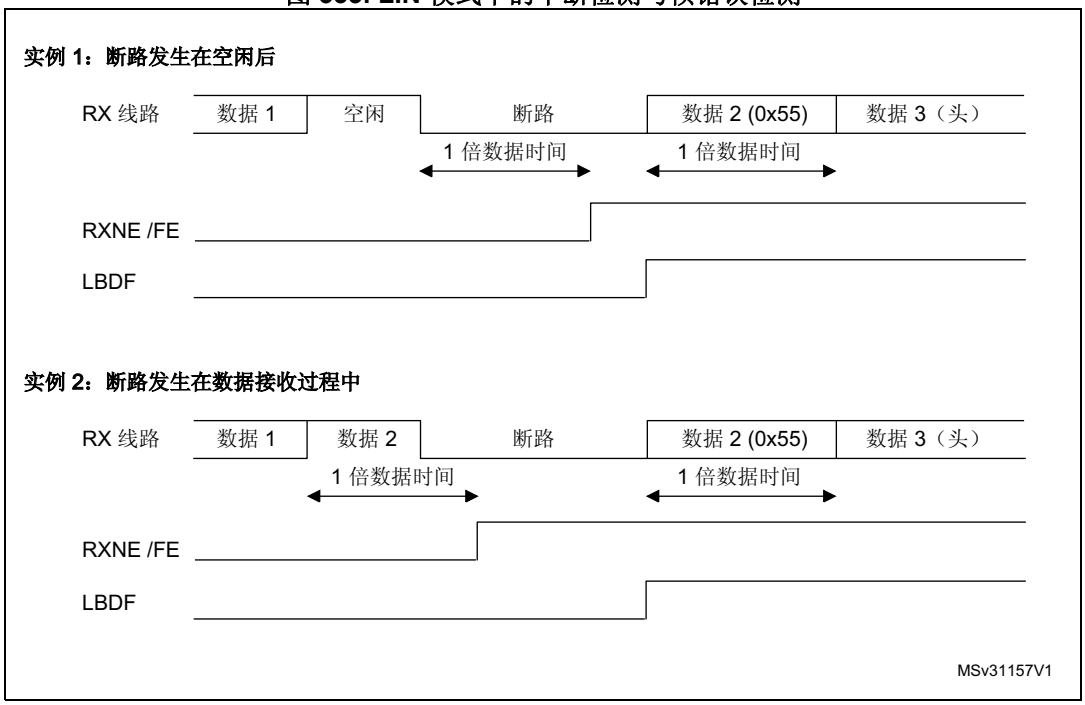


图 333. LIN 模式下的中断检测与帧错误检测



33.5.14 USART 同步模式

主模式

通过将 USART_CR2 寄存器中的 CLKEN 位编程为“1”来选择同步主模式。在同步模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位。
- USART_CR3 寄存器中的 SCEN、HDSEL 和 IREN 位。

在此模式下，USART 可用于在主模式下控制双向同步串行通信。SCLK 引脚是 USART 发送器时钟的输出。在起始位或停止位期间，不会向 SCLK 引脚发送时钟脉冲。在最后一个有效数据位（地址标记）期间，会（也可能不会）生成时钟脉冲，这取决于 USART_CR2 寄存器中 LBCL 位的状态。USART_CR2 寄存器中的 CPOL 位用于选择时钟极性；USART_CR2 寄存器中的 CPHA 位用于选择外部时钟的相位（请参见*图 334*、*图 335* 和*图 336*）。

在空闲状态、报头模式和发送中断期间，外部 SCLK 时钟处于未激活状态。

在同步主模式下，USART 发送器的工作方式与异步模式下完全相同。但是，由于 SCLK 与 TX 同步（根据 CPOL 和 CPHA），因此 TX 上的数据是同步的。

在同步主模式下，USART 接收器的工作方式与异步模式下不同。如果 RE 置 1，则数据在 SCLK 上采样（上升或下降沿，具体取决于 CPOL 和 CPHA），而不会进行任何过采样。此时必须确保给定建立时间和保持时间（取决于波特率：1/16 位时间）符合要求。

注： *在主模式下，SCLK 引脚可与 TX 引脚结合使用。因此，仅当使能发送器 (TE=1) 且正在发送数据时（USART_TDR 数据寄存器已写入），才会提供时钟。这意味着，没有发送数据的情况下无法接收同步数据。*

图 334. USART 同步主发送示例

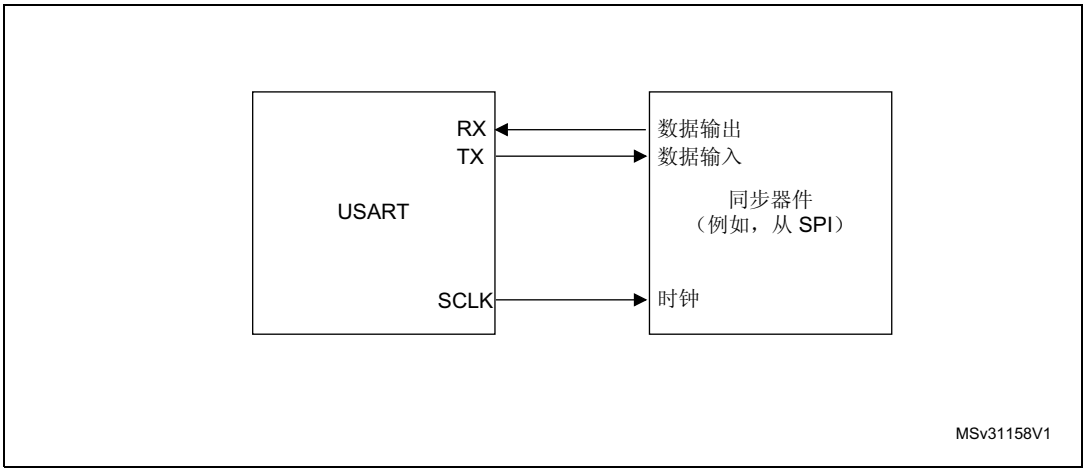


图 335. USART 在同步主模式下的数据时钟时序图（M 位 = 00）

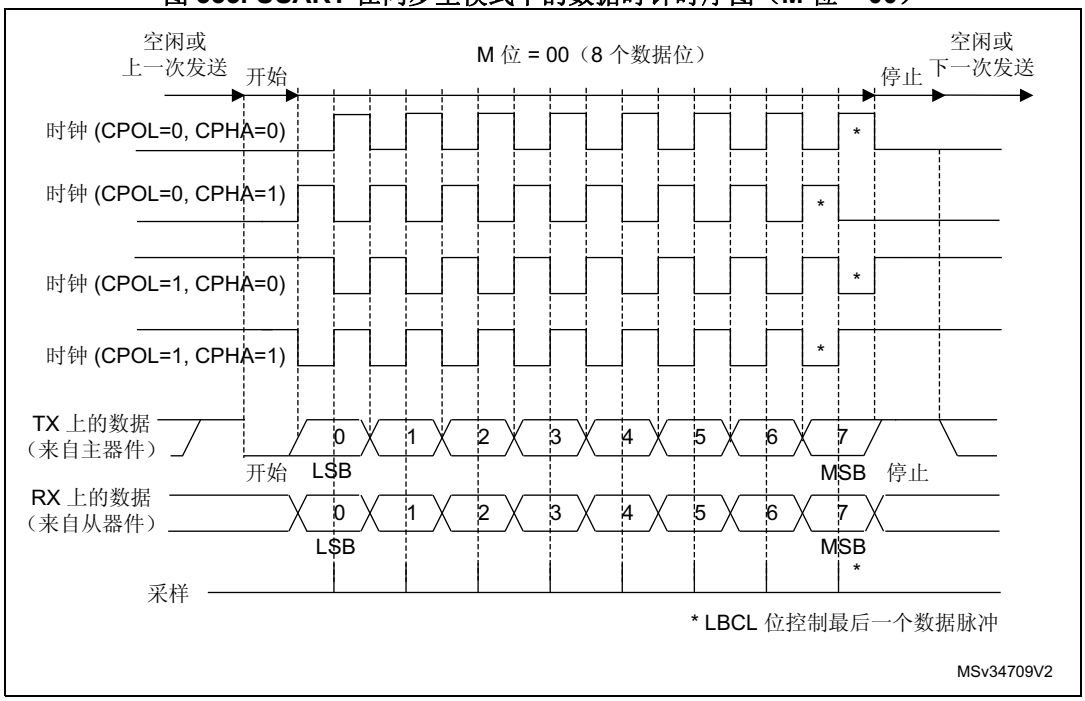
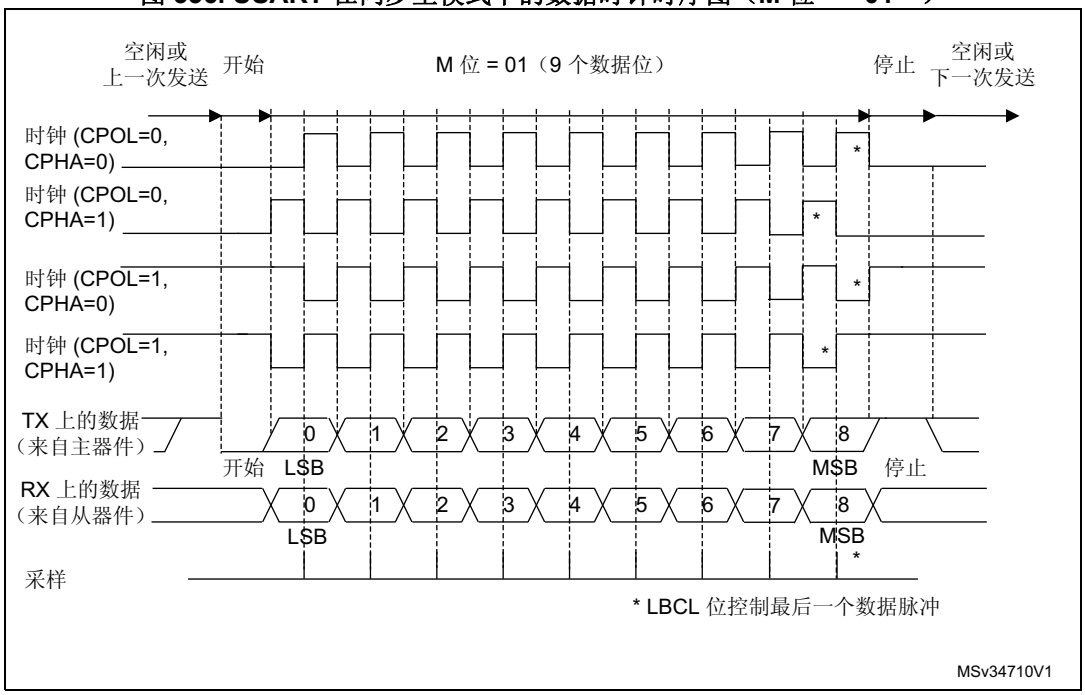


图 336. USART 在同步主模式下的数据时钟时序图（M 位 =“01”）



从模式

通过将 USART_CR2 寄存器中的 SLVEN 位编程为“1”来选择同步从模式。在同步从模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 和 CLKEN 位。
- USART_CR3 寄存器中的 SCEN、HDSEL 和 IREN 位。

在此模式下，USART 可用于在从模式下控制双向同步串行通信。在从模式下，SCLK 引脚是 USART 的输入。

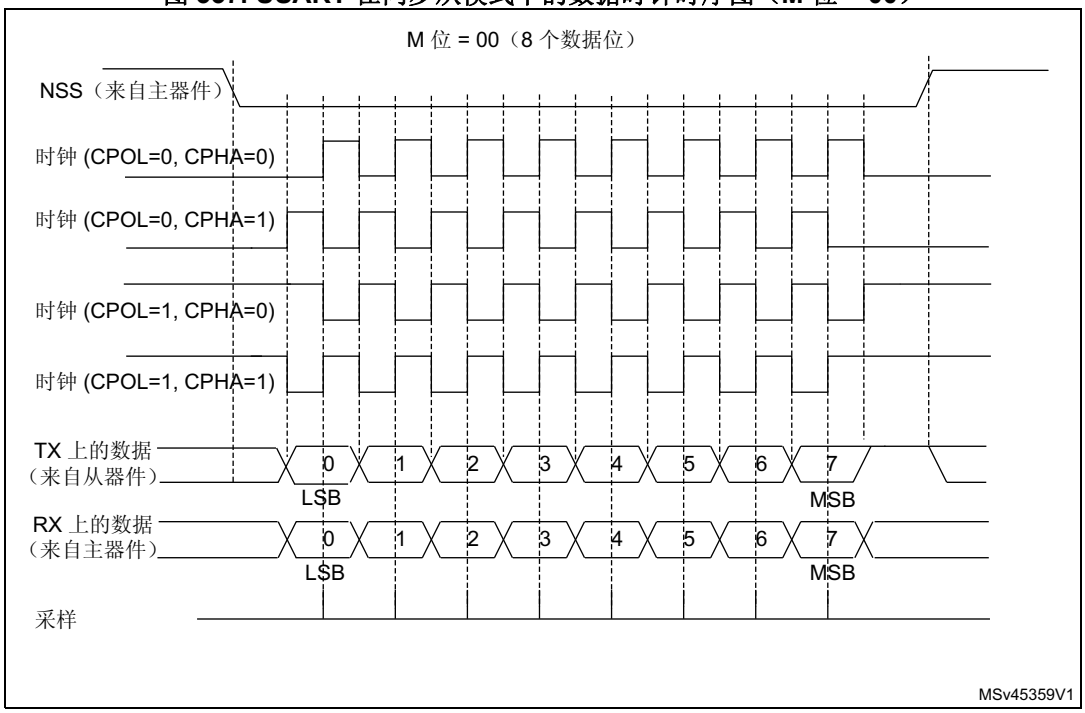
注： *当外设用于 SPI 从器件模式时，外设时钟源 (usart_ker_ck_pres) 的频率必须是 CK 输入频率的 3 倍以上。*

USART_CR2 寄存器中的 CPOL 位和 CPHA 位分别用于选择时钟极性和外部时钟的相位（请参见*图 337*）。

从发送模式支持下溢错误标志。如果在软件尚未将任何值加载到 USART_TDR 之前出现第一个数据发送时钟脉冲，则此标志将置 1。

从器件支持硬件和软件 NSS 管理。

图 337. USART 在同步从模式下的数据时钟时序图（M 位 = 00）



从器件选择 (NSS) 引脚管理

可通过 USART_CR2 寄存器中的 DIS_NSS 位设置硬件或软件从器件选择管理：

- 软件 NSS 管理 (DIS_NSS = 1)

  将始终选择 SPI 从器件并忽略 NSS 输入引脚。

  外部 NSS 引脚空闲，可供其它应用使用。
- 硬件 NSS 管理 (DIS_NSS = 0)

  SPI 从器件选择取决于 NSS 输入引脚。NSS 为低电平时选择从器件，NSS 为高电平时取消选择从器件。

注： *禁止 USART 时 (UE=0)，必须选择 LBCL（仅用于 SPI 主器件模式）、CPOL 和 CPHA 位以确保时钟脉冲正常工作。*

在 SPI 从器件模式下，必须在启动主器件通信之前（或时钟稳定时在相邻帧之间）使能 USART。否则，如果 USART 从器件在主器件处于某个帧中间时使能，则它将与主器件失去同步。在通信时钟的第一个边沿到来之前或者正在进行的通信结束之前，从器件的数据寄存器就需要准备就绪，否则 SPI 从器件会发送零。

SPI 从器件下溢错误

发生下溢错误时，USART_ISR 寄存器中的 UDR 标志会置 1，SPI 从器件继续发送最后一个数据，直到下溢错误标志由软件清零。

下溢标志在帧开始时置 1。如果 USART_CR3 寄存器中的 EIE 位置 1，则会触发下溢错误中断。

通过将 USART_ICR 寄存器中的位 UDRCF 置 1 来清零下溢错误标志。

发生下溢错误时，仍然可以对 TDR 寄存器进行写操作。清除下溢错误将允许发送新数据。

如果发生下溢错误且没有新数据被写入 TDR 中，则 TC 标志在帧结束时置 1。

注： *如果向 USART_TDR 写入数据的时间点过于接近第一个 SCLK 发送边沿，则可能会发生下溢错误。为避免发生此下溢错误，应在第一个 SCLK 边沿的 3 个 usart_ker_ck 周期之前对 USART_TDR 进行写操作。*

33.5.15 USART 单线半双工通信

通过将 USART_CR3 寄存器中的 HDSEL 位置 1 来选择单线半双工模式。在此模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 和 CLKEN 位。
- USART_CR3 寄存器中的 SCEN 和 IREN 位。

USART 可以配置为遵循单线半双工协议，其中 TX 和 RX 线路从内部相连接。使用 USART_CR3 寄存器中的控制位 HDSEL 可在半双工通信和全双工通信间进行选择。

向 HDSEL 位写入“1”后：

- TX 和 RX 线路从内部相连接。
- 不再使用 RX 引脚。
- 无数据传输时，TX 引脚始终处于释放状态。因此，它在空闲状态或接收过程中用作标准 I/O。这意味着，必须对 I/O 进行配置，以便将 TX 配置为复用功能开漏并外接上拉电阻。

除此之外，通信协议与正常 USART 模式下的通信协议相似。此线路上的任何冲突都必须由软件管理（例如，使用中央仲裁器）。尤其要注意，发送过程永远不会被硬件封锁，只要数据是在 TE 位置 1 的情况下写入，发送就会持续进行。

33.5.16 USART 接收器超时

接收器超时功能可通过将 USART_CR2 控制寄存器中的 RTOEN 位置 1 来使能。

超时间隔通过 USART_RTOR 寄存器中的 RTO 位域进行编程。

接收器超时计数器遵循以下规则开始计数：

- STOP =“00”或 STOP =“11”时从停止位结束时开始计数。
- STOP =“10”时从第二个停止位结束时开始计数。
- STOP =“01”时从停止位开始时开始计数。

经过超时间隔后，USART_ISR 寄存器中的 RTOF 标志置 1。如果 USART_CR1 寄存器中的 RTOIE 位置 1，则会产生超时中断。

33.5.17 USART 智能卡模式

仅在支持智能卡模式时才涉及本节内容。请参见*第 996 页的第 33.4 节：USART 实现*。

通过将 USART_CR3 寄存器中的 SCEN 位置 1 选择智能卡模式。在智能卡模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位。
- USART_CR3 寄存器中的 HDSEL 和 IREN 位。

此外，还可将 CLKEN 位置 1，以便为智能卡提供时钟。

智能卡接口支持符合 ISO 7816-3 标准的异步智能卡协议。同时支持 T=0（字符模式）和 T=1（块模式）。

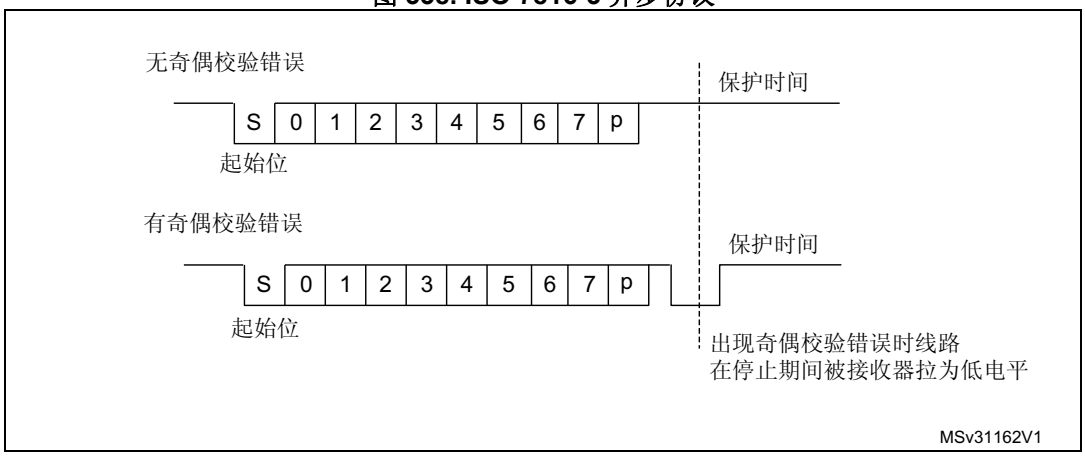
USART 应如下所示进行配置：

- 8 个位加奇偶校验：当 USART_CR1 寄存器中的 M=1 且 PCE=1 时。
- 发送和接收数据时使用 1.5 个停止位：当 USART_CR2 寄存器中的 STOP=“11”时。接收时也可以选择 0.5 个停止位。

在 T=0（字符）模式下，奇偶校验错误在保护时间周期内的每个字符结束时指示。

图 338 显示了有奇偶校验错误和无奇偶校验错误时数据线上情况的示例。

图 338. ISO 7816-3 异步协议



连接到智能卡时，USART 的 TX 输出会驱动一条双向线（它也由智能卡驱动）。必须将 TX 引脚配置为开漏引脚。

智能卡模式采用单线半双工通信协议。

- 从发送移位寄存器发送数据会经过至少 1/2 个时钟周期的延迟。正常工作时，已满的发送移位寄存器会在下一个波特时钟边沿开始移位。在智能卡模式下，此发送过程还会进一步经过 1/2 波特时钟周期的延迟。
- 发送时，如果智能卡检测到奇偶校验错误，它会通过将线路驱动为低电平 (NACK) 告知 USART 此状态。此 NACK 信号（将发送线拉低 1 个时钟周期）会导致发送器端（配置为 1.5 个停止位）出现帧错误。USART 可根据协议自动重新发送数据。重试次数在 SCARCNT 位域中编程。如果经过编程次数的重试后，USART 继续收到 NACK，USART 会停止发送并将该错误以帧错误形式发出。TXE 位（使能 FIFO 模式时为 TXFNF 位）可使用 USART_RQR 寄存器中的 TXFRQ 位置 1。
- 发送时智能卡自动重试：在 USART 检测 NACK 与重复字符的起始位之间插入 2.5 个波特率周期的延迟。接收最后一个重复字符后，立即将 TC 位置 1（无保护时间）。如果软件要重复此操作，必须确保标准要求的最短 2 个波特率周期。

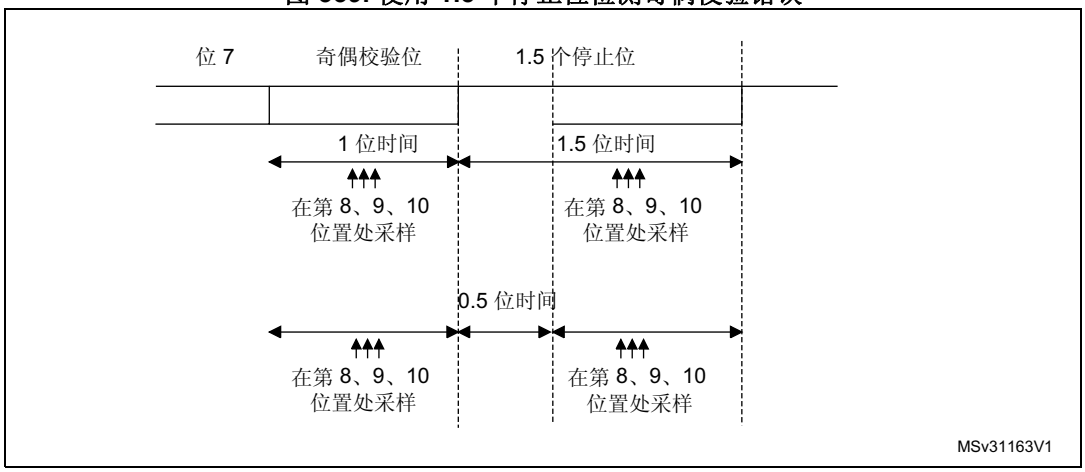
- 如果在接收一个使用 1.5 个停止位编程的帧期间检测到奇偶校验错误，则在完成接收帧后，发送线会被拉低一个时钟周期。这是为了向智能卡指出发送到 USART 的数据尚未正确接收。如果 NACK 控制位置 1，接收器会发送“NACK”信号表示奇偶校验错误；否则不会发送 NACK 信号（将在 T=1 模式下会使用）。如果接收到的字符错误，则不会激活 RXNE（使能 FIFO 模式时为 RXFNE）/接收 DMA 请求。根据协议规范，智能卡必须重新发送相同的字符。如果经过 SCARCNT 位域中指定的最大重试次数后接收到的字符仍然错误，USART 会停止发送 NACK 信号，并将错误以奇偶校验错误的形式发出。
- 接收时智能卡自动重试：如果 USART 向智能卡发送 NACK 信号，但智能卡不重复字符，则 BUSY 标志将保持置 1。
- 发送时，USART 会在两个连续字符之间插入保护时间（按照保护时间寄存器中编程的值）。由于保护时间在前一个字符的停止位后测量，因此必须将 GT[7:0] 寄存器编程为所需 CGT（字符保护时间，如 7816-3 规范所定义）减去 12（一个字符的持续时间）。
- 通过对保护时间寄存器进行编程，可以延迟 TC 标志的置位。正常工作时，当发送移位寄存器为空且没有新的发送请求出现时，会对 TC 标志进行置位。在智能卡模式下，空的发送移位寄存器会触发保护时间计数器，使其递增计数至保护时间寄存器中的值。在此期间，TC 标志被强制为低电平。当保护时间计数器达到设置值时，TC 置位为高电平。TCBGT 标志可用于检测数据传输是否结束，而无需等待保护时间结束。该标志在帧发送结束之后且未从智能卡接收到 NACK 时置 1。
- TC 标志的释放不受智能卡模式的影响。
- 如果在发送端检测到帧错误（由来自接收器的 NACK 信号引起），则发送端的接收块不会将 NACK 作为起始位进行检测。根据 ISO 协议，接收到的 NACK 信号的持续时间可以是 1 或 2 个时钟周期。
- 在接收端，如果检测到奇偶校验错误并发送了 NACK 信号，则接收端不会将 NACK 作为起始位进行检测。

注： *中断字符在智能卡模式下无效。带有帧错误的 0x00 数据被视为数据，而非中断。*

当翻转 TE 位时，不会发送空闲帧。空闲帧（在其它配置中进行了定义）在 ISO 协议中未进行定义。

图 339 详细介绍了 USART 如何对 NACK 信号采样。在本例中，USART 正在发送数据并配置了 1.5 个停止位。USART 的接收部分已被使能，以检查数据的完整性和 NACK 信号。

图 339. 使用 1.5 个停止位检测奇偶校验错误



USART 可以通过 SCLK 输出为智能卡提供时钟。在智能卡模式下，SCLK 仅通过一个 5 位预分频器由内部外设输入时钟提供。分频比在 USART_GTPR 寄存器中进行配置。SCLK 频率可在 usart_ker_ck_pres/2 到 usart_ker_ck_pres/62 之间进行编程，其中 usart_ker_ck_pres 为经过编程的预分频器分频的外设输入时钟。

块模式 (T=1)

在 T=1（块）模式下，通过将 USART_CR3 寄存器中的 NACK 位清零可停用奇偶校验错误发送。

在块模式下请求从智能卡执行读操作时，软件必须将 RTOR 寄存器编程为 BWT（块等待时间）-11。如果在经过此时间段后未从智能卡接收到应答，将生成超时中断。如果在该时间段超时之前接收到第一个字符，则通过 RXNE/RXFNE 中断发出信号指示。

注： *即使在使用 DMA 模式下的 USART 从块模式下的智能卡读取时，也必须使能 RXNE/RXFNE 中断。同时，只有在接收到第一个字节后才可使能 DMA。*

接收到第一个字符（RXNE/RXFNE 中断）后，为允许自动检查两个连续字符间的最长等待时间，必须将 RTO 寄存器编程为 CWT（字符等待时间 - 值 11）。此时间以波特率时间单位表示。前一个字符结束后，如果智能卡未在小于 CWT 的时间段内发送新字符，USART 将通过 RTOF 标志和中断（当 RTOIE 位置 1 时）向软件指示此情况。

注： *按照智能卡协议定义，BWT/CWT 的值应从最后一个字符开始（起始位）时定义。必须将 RTO 寄存器分别编程为 BWT -11 或 CWT -11，并考虑最后一个字符本身的长度。*

块长度计数器用于对 USART 接收到的所有字符进行计数。当 USART 进行发送时，此计数器复位。块长度由智能卡在块的第三个字节（起始字段）中传达。必须将此值编程到 USART_RTOR 寄存器中的 BLEN 字段。使用 DMA 模式时，在块开始之前，必须将此寄存器字段编程为最小值 (0x0)。对于该值，在接收到第四个字符后生成中断。软件必须读取 LEN 字段（第三个字节），其值必须从接收缓冲区中读取。

在中断驱动接收模式下，块长度可以由软件或者通过编程 BLEN 值来检查。但是在块开始前，可以编程 BLEN 的最大值 (0xFF)。接收到第三个字符后，将编程实际值。

如果块使用 LRC（纵向冗余校验，1 个结尾字节），则 BLEN=LEN。如果块使用 CRC 机制（2 个结尾字节），则必须编程 BLEN=LEN+1。块总长度（包括起始字段、结尾字段和信息字段）等于 BLEN+4。块结束的信号通过 EOBF 标志和中断（EOBIE 位置 1 时）发送给软件。

如果块长度出现错误，则通过 RTO 中断（字符等待时间上溢）发送块结束信号。

注： *错误检查代码 (LRC/CRC) 必须通过软件计算/验证。*

正向约定和反向约定

智能卡协议定义了两种约定：正向约定和反向约定。

正向约定定义为：LSB 在前，逻辑位值 1 对应于线路的 H 状态，奇偶校验为偶校验。要使用此约定，必须编程以下控制位：MSBFIRST=0，DATAINV=0（默认值）。

反向约定定义为：MSB 在前，逻辑位值 1 对应于信号线路的 L 状态，奇偶校验为偶校验。要使用此约定，必须编程以下控制位：MSBFIRST=1，DATAINV=1。

注： *将逻辑数据值取反（0=H，1=L）时，奇偶校验位将同样取反。*

为识别智能卡约定，智能卡会将初始字符 TS 作为 ATR（复位应答）的第一个字符发送。TS 支持两种格式：LHHL LLL LLH 和 LHHL HHH LLH。

- (H) LHHL LLL LLH 建立反向约定：状态 L 编码为值 1，时间分量 2 传送最高有效位（MSB 在前）。按反向约定解码时，传送的字节等于“3F”。
- (H) LHHL HHH LLH 建立正向约定：状态 H 编码为值 1，时间分量 2 传送最低有效位（LSB 在前）。按正向约定解码时，传送的字节等于“3B”。

在 2 到 10 的九个时间分量中，如果有偶数个位设置为 1，则字符的奇偶校验正确。

由于 USART 不了解智能卡使用哪种约定，因此 USART 需要能够识别任意一种模式并相应操作。模式识别不在硬件中完成，而是通过软件序列完成。此外，假设以正向约定配置 USART（默认）而智能卡以反向约定 (TS = LHHL LLL LLH) 应答，则 USART 接收到的字节将为“03”，奇偶校验将为奇校验。

因此，有两种方法可用于识别 TS 模式：

方法 1

将 USART 编程为标准智能卡模式/正向约定。这种情况下，TS 模式接收会向智能卡生成奇偶校验错误中断和错误信号。

- 奇偶校验错误中断通知软件智能卡未以正向约定正确应答。之后，软件重新将 USART 编程为反向约定
- 为响应错误信号，智能卡会重试同一 TS 字符，此时重新编程后的 USART 将正确接收该字符

或者，为应答奇偶校验错误中断，软件可决定重新编程 USART，同时向智能卡生成新的复位命令，然后再次等待 TS。

方法 2

将 USART 编程为 9 位/无奇偶校验模式，无位反向。在此模式下，按如下方式接收两种 TS 模式中的任一种：

(H) LHHL LLL LLH = 0x103 -> 选择反向约定

(H) LHHL HHH LLH = 0x13B -> 选择正向约定

软件根据这两种模式检查接收到的字符，如果其中任意一种匹配，则相应编程 USART 以接收下一个字符。

如果两种都未被识别，则可能复位智能卡以重新开始协商。

33.5.18 USART IrDA SIR ENDEC 模块

仅在支持 IrDA 模式时才涉及本节内容。请参见*第 996 页的第 33.4 节：USART 实现*。

通过将 USART_CR3 寄存器中的 IREN 位置 1 来选择 IrDA 模式。在 IrDA 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN、STOP 和 CLKEN 位。
- USART_CR3 寄存器中的 SCEN 和 HDSEL 位。

IrDA SIR 物理层规定使用反相归零 (RZI) 调制方案，它以红外光脉冲表示逻辑 0（参见*图 340*）。

SIR 发送编码器用于调制 USART 发出的非归零 (NRZ) 位流。输出脉冲流会发送到外部输出驱动器和红外线 LED。USART 支持的 SIR ENDEC 比特率最高为 115.2 Kbps。在正常模式下，所发送的脉冲宽度规定为一个位周期的 3/16。

SIR 接收解码器用于解调由红外探测器发出的归零位流，并将接收到的 NRZ 串行位流输出到 USART。在空闲状态下，解码器输入通常为高电平（标记状态）。发送编码器输出的极性与解码器输入相反。当解码器输入为低电平时，会检测到起始位。

- IrDA 是一个半双工通信协议。如果发送器忙（USART 正在向 IrDA 编码器发送数据时），则 IrDA 解码器会忽略 IrDA 接收线上的所有数据；如果接收器忙（USART 正在接收来自 USART 的解码数据时），则 IrDA 不会对 USART 发送到 IrDA 的 TX 上的数据进行编码。在接收数据时，应避免同时进行发送，因为这样做可能会破坏要发送的数据。
- “0”作为高电平脉冲发送，而“1”作为“0”发送。在正常模式下，脉冲宽度规定为所选位周期的 3/16（参见*图 341*）。
- SIR 解码器用于将兼容 IrDA 的接收信号转换为 USART 的位流。
- SIR 接收逻辑将高电平状态视为逻辑“1”，将低电平脉冲视为逻辑“0”。
- 发送编码器输出的极性与解码器输入相反。SIR 输出在空闲时处于低电平状态。
- IrDA 规范要求脉冲容忍值要大于 1.41 µs。可接受的脉冲宽度可通过寄存器设置。接收器端的干扰检测逻辑会滤除宽度小于 2 个 PSC 周期的脉冲（PSC 是在 USART_GTPR 中编程的预分频器值）。宽度小于 1 个 PSC 周期的脉冲都将被拒绝，但宽度大于 1 个而小于 2 个周期的脉冲可能被接受也可能被拒绝，而宽度大于 2 个周期的脉冲将被接受作为有效脉冲。当 PSC=0 时，IrDA 编码器/解码器不工作。
- 接收器能够与低功耗发送器进行通信。
- 在 IrDA 模式下，USART_CR2 寄存器中的停止位必须配置为“1 个停止位”。

IrDA 低功耗模式

- 发送器

  在低功耗模式下，脉冲宽度不再保持为位周期的 3/16。此时的脉冲宽度为低功耗波特率的 3 倍，最小可为 1.42 MHz。通常此值是 1.8432 MHz (1.42 MHz < PSC < 2.12 MHz)。低功耗模式下的可编程分频器会对系统时钟进行分频，以达到此值。
- 接收器

  在低功耗模式下接收与在正常模式下接收类似。为进行干扰检测，USART 应丢弃持续时间短于 1/PSC 的脉冲。只有当持续时间长于 2 个 IrDA 低功耗波特时钟周期（USART_GTPR 的 PSC 值）时，才是有效低电平。

注： *宽度小于两个但大于一个 PSC 周期的脉冲可能被接受，也可能被拒绝。*

接收器的建立时间应由软件进行管理。IrDA 物理层规范规定发送和接收之间至少要经过 10 ms 的延迟（IrDA 是一个半双工协议）。

图 340. IrDA SIR ENDEC 框图

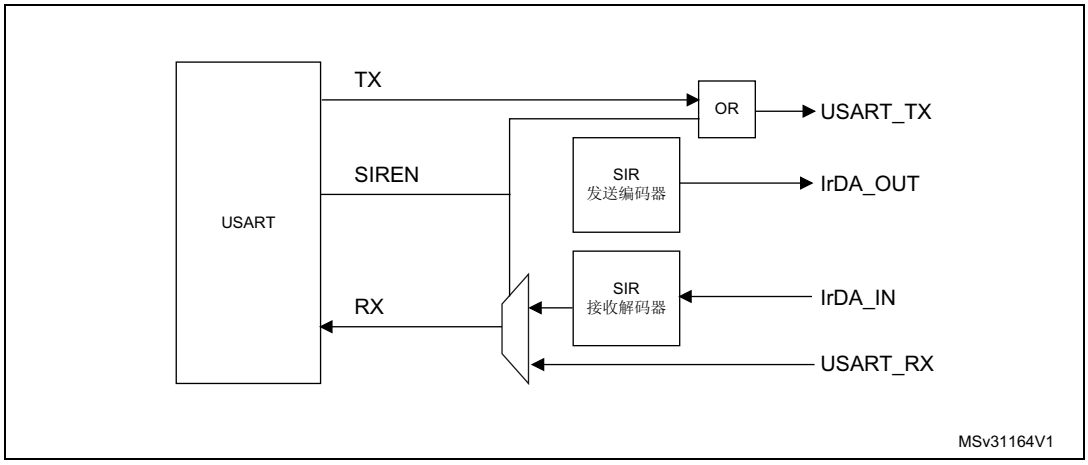
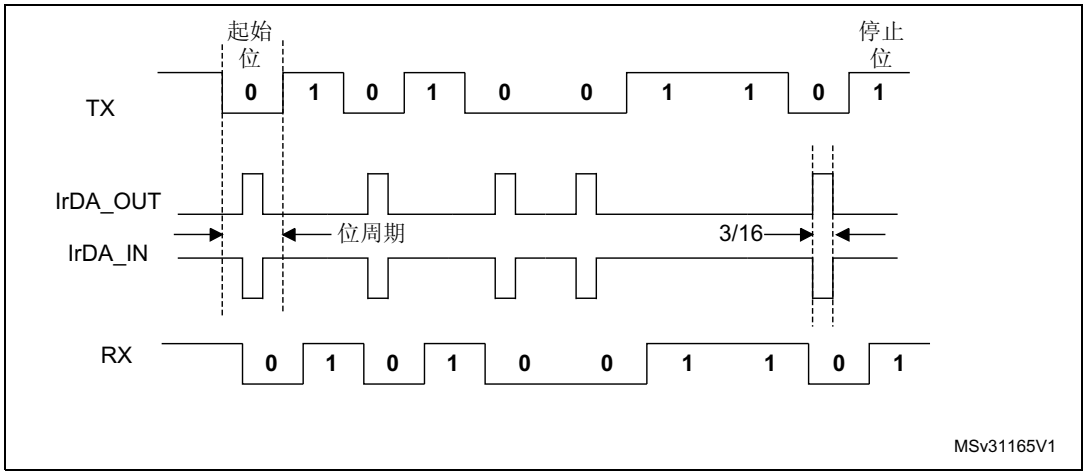


图 341. IrDA 数据调制 (3/16)——正常模式



33.5.19 使用 USART 和 DMA 进行连续通信

USART 能够使用 DMA 进行连续通信。接收缓冲区和发送缓冲区的 DMA 请求是独立生成的。

注： *要确定是否支持 DMA 模式，请参见第 996 页的第 33.4 节：USART 实现。如果不支持 DMA 模式，请按照第 33.5.6 节中的说明使用 USART。可以将 USART_ISR 寄存器中的 TXE/RXNE 标志清零，从而在禁止 FIFO 时实现连续通信。*

使用 DMA 进行发送

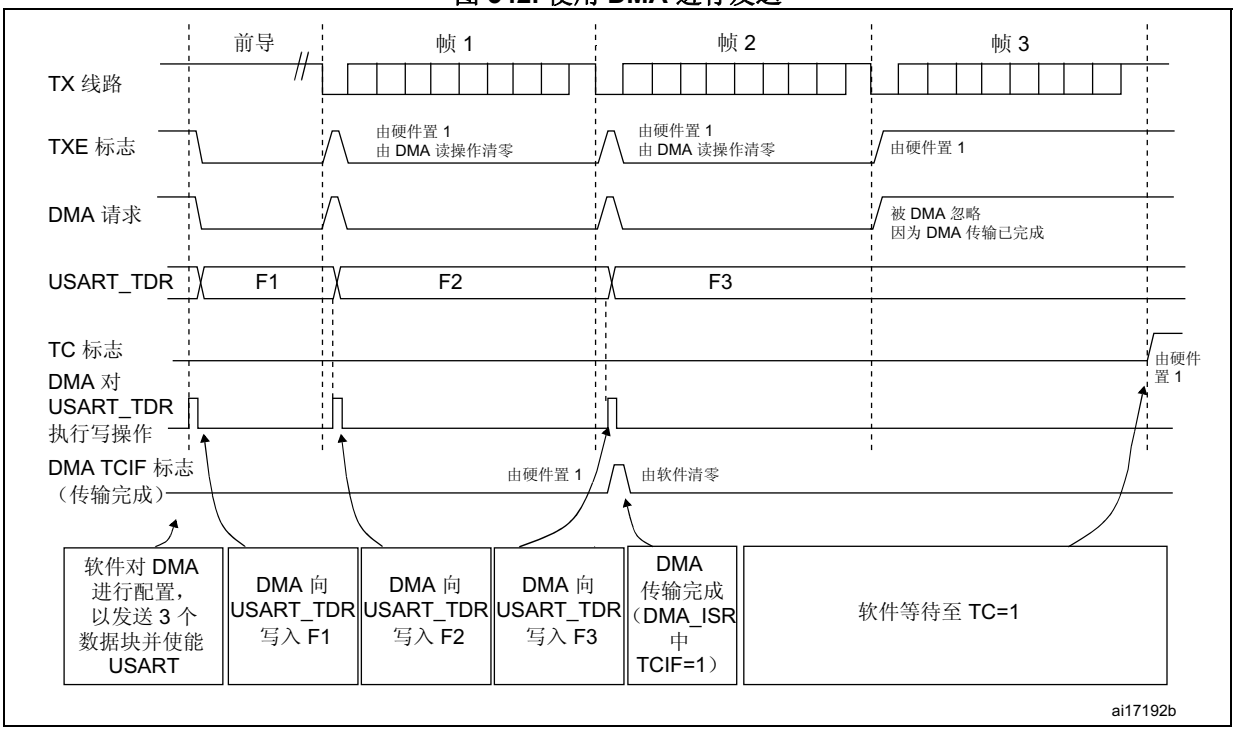
将 USART_CR3 寄存器中的 DMAT 位置 1 可以使能 DMA 模式进行发送。当 TXE 标志（使能 FIFO 模式时为 TXFNF 标志）置 1 时，可使用 DMA 外设（请参见相应的*直接存储器访问控制器*部分）将数据从配置的 SRAM 区域加载到 USART_TDR 寄存器。要映射一个 DMA 通道以进行 USART 发送，请按以下步骤操作（x 表示通道编号）：

1. 在 DMA 控制寄存器中写入 USART_TDR 寄存器地址，将其配置为传输的目标地址。每次发生 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据都从存储器移动到此地址。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的源地址。每次发生 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据都从该存储区域加载到 USART_TDR 寄存器中。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生 DMA 中断。
6. 通过将 USART_ICR 寄存器中的 TCCF 位置 1，将 USART_ISR 寄存器中的 TC 标志清零。
7. 在 DMA 寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

在发送模式下，DMA 对所有要发送的数据执行了写操作（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 TC 标志进行监视，以确保 USART 通信已完成。在禁止 USART 之前或系统进入低功耗模式之前（禁止外设时钟时）必须执行此步骤，以避免损坏最后一次发送。软件必须等待直到 TC=1。TC 标志在所有数据发送期间都保持清零状态，然后在最后一帧发送结束时由硬件置 1。

图 342. 使用 DMA 进行发送



注： *使能 FIFO 管理时，DMA 请求由发送 FIFO 未满（即 TXFNF = 1）触发。*

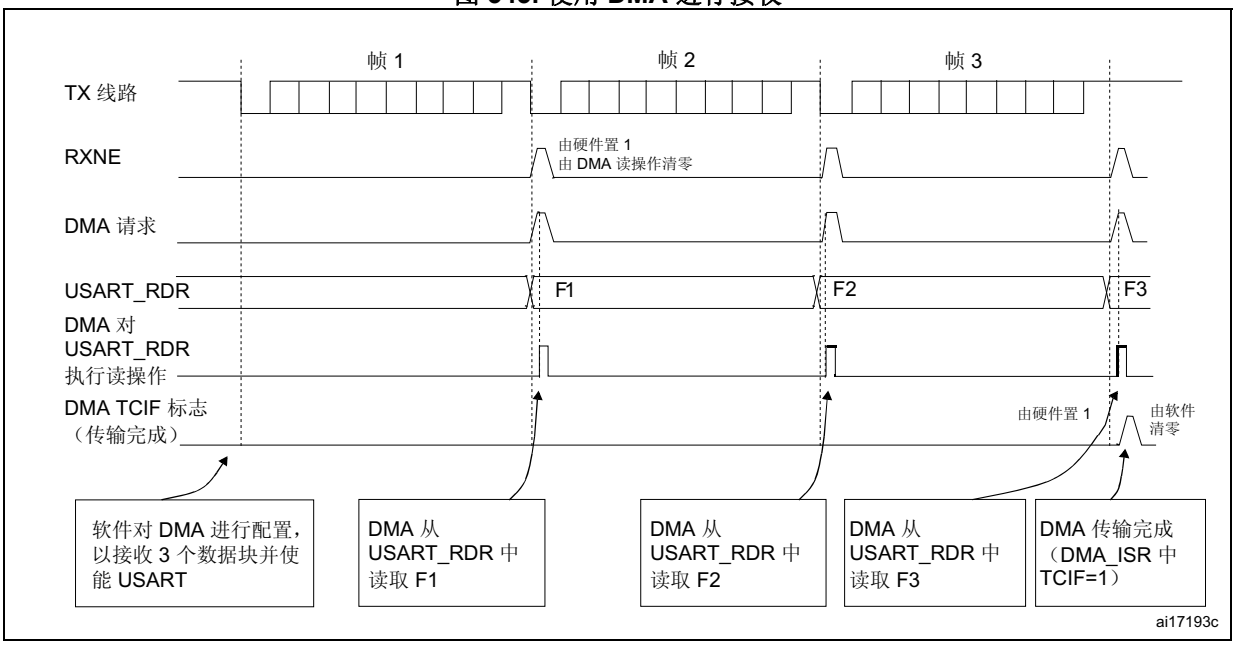
使用 DMA 进行接收

将 USART_CR3 寄存器中的 DMAR 位置 1 可以使能 DMA 模式进行接收。接收数据字节时，可使用 DMA 外设（请参见相应的*直接存储器访问控制器*部分）将数据从 USART_RDR 寄存器加载到配置的 SRAM 区域中。要映射一个 DMA 通道以进行 USART 接收，请按以下步骤操作：

1. 在 DMA 控制寄存器中写入 USART_RDR 寄存器地址，将其配置为传输的源地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都从该地址移动到存储器。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的目标地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都从 USART_RDR 加载到此存储区域。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 控制寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生中断。
6. 在 DMA 控制寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

图 343. 使用 DMA 进行接收



注： *使能 FIFO 管理时，DMA 请求由接收 FIFO 非空（即 RXFNE = 1）触发。*

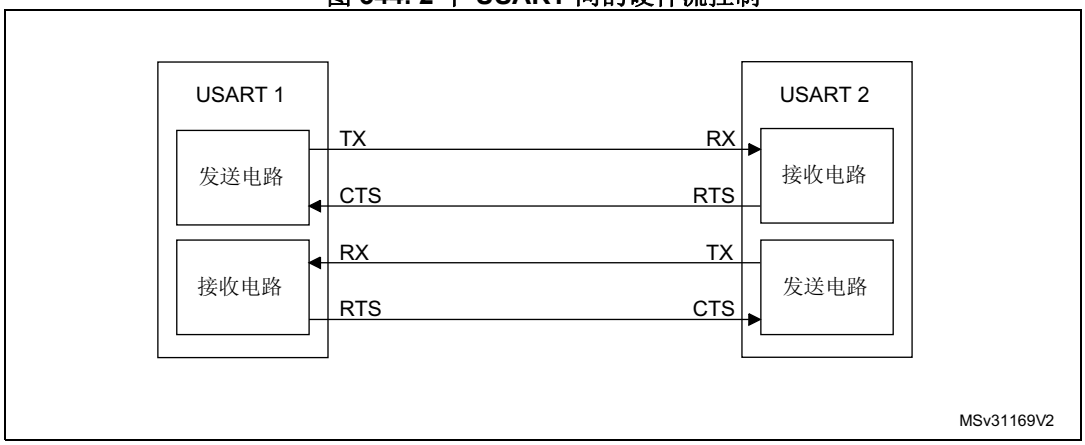
多缓冲区通信中的错误标志和中断生成

在多缓冲区通信模式下，如果事务中发生任何错误，都会在当前字节后将相应错误标志置 1。如果中断使能标志置 1，则会产生中断。在单字节接收过程中，与 RXNE（使能 FIFO 模式时为 RXFNE）一同置位的帧错误、上溢错误和噪声标志具有单独的错误标志中断使能位（USART_CR3 寄存器中的 EIE 位）；如果该位置 1，在发生其中任何一个错误时，都会在当前字节后使能中断。

33.5.20 RS232 硬件流控制和 RS485 驱动器使能

使用 nCTS 输入和 nRTS 输出可以控制 2 个器件间的串行数据流。*图 344* 显示了在这种模式下如何连接 2 个器件：

图 344. 2 个 USART 间的硬件流控制

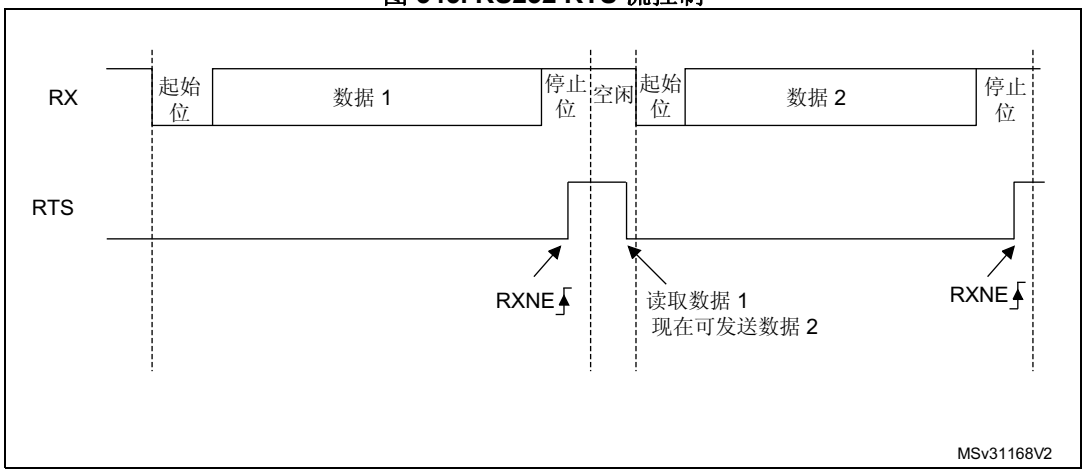


向 USART_CR3 寄存器中的 RTSE 位和 CTSE 位写入“1”，可分别使能 RS232 RTS 和 CTS 流控制。

RS232 RTS 流控制

如果使能 RTS 流控制 (RTSE=1)，只要 USART 接收器准备好接收新数据，便会将 nRTS 变为有效（连接到低电平）。当接收寄存器已满时，会将 nRTS 变为无效，表明发送过程会在当前帧结束后停止。*图 345* 显示了在使能 RTS 流控制的情况下进行通信的示例。

图 345. RS232 RTS 流控制



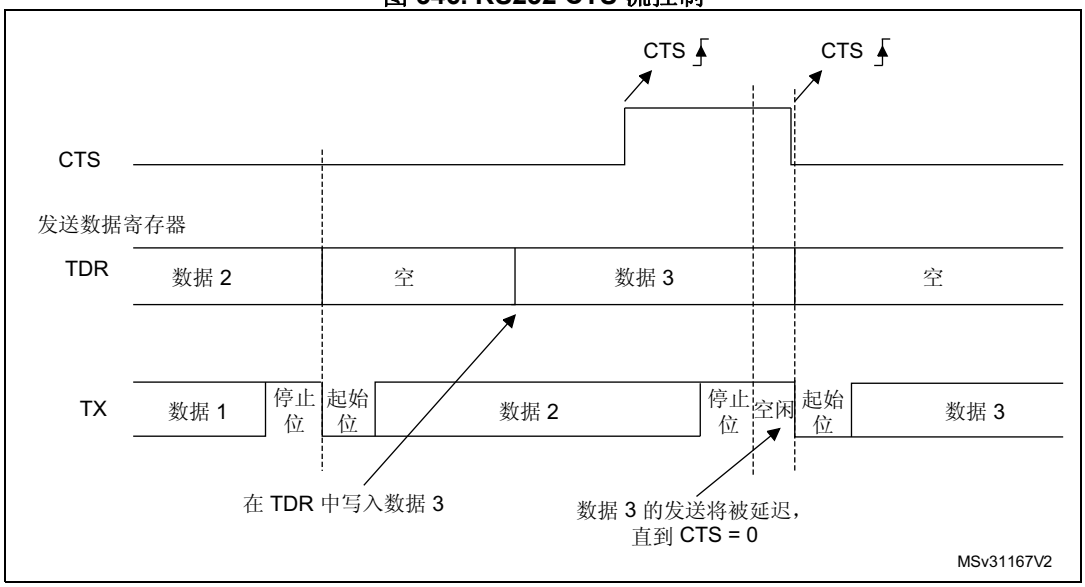
注： *如果使能 FIFO 模式，则仅当 RXFIFO 已满时，nRTS 才会变为无效。*

RS232 CTS 流控制

如果使能 CTS 流控制 (CTSE=1)，则发送器会在发送下一帧前检查 nCTS。如果 nCTS 有效（连接到低电平），则会发送下一数据（假设数据已准备好发送，即 TXE/TXFE=0）；否则不会进行发送。如果在发送过程中 nCTS 变为无效，则当前发送完成之后，发送器停止。

当 CTSE=1 时，只要 nCTS 发生变化，CTSIF 状态位便会由硬件自动置 1。这指示接收器是否已准备好进行通信。如果 USART_CR3 寄存器中的 CTSIE 位置 1，则会产生中断。*图 346* 显示了在使能 CTS 流控制的情况下进行通信的示例。

图 346. RS232 CTS 流控制



注： *为正常运行，必须在当前字符结束前至少 3 个 USART 时钟源周期内使能 nCTS。此外还应注意，当脉冲短于 2 个 PCLK 周期时无法将 CTSCF 标志置 1。*

RS485 驱动器使能

驱动器使能功能可通过将 USART_CR3 控制寄存器中的 DEM 位置 1 使能。这样用户便可通过 DE（驱动器使能）信号激活外部收发器控制。使能时间为激活 DE 信号与起始位开始间的时间。可以通过 USART_CR1 控制寄存器中的 DEAT [4:0] 位域编程禁止时间。禁止时间为发送的消息中最后一个停止位结束与取消激活 DE 信号间的时间。可以通过 USART_CR1 控制寄存器中的 DEDT [4:0] 位域编程禁止时间。DE 信号的极性可使用 USART_CR3 控制寄存器中的 DEP 位配置。

在 USART 中，DEAT 和 DEDT 以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

33.5.21 USART 低功耗管理

USART 具有高级低功耗模式功能，即使禁止 usart_pclk 时钟，也能正常传输数据。

UESM 位置 1 时，USART 能够将 MCU 从低功耗模式唤醒。

对 usart_pclk 进行门控时，如果某些特定操作需要激活 **usart_pclk** 时钟，则 USART 会提供唤醒中断 (**usart_wkup**)：

- 禁止 FIFO 模式时

  必须激活 usart_pclk 时钟以清空 USART 数据寄存器。

  在这种情况下，usart_wkup 中断源为 RXNE 置“1”。RXNEIE 位必须在进入低功耗模式之前置 1。

- 使能 FIFO 模式时

  必须激活 usart_pclk 时钟以：

  – 填充 TXFIFO

  – 或清空 RXFIFO

  在这种情况下，usart_wkup 中断源可以为：

  – RXFIFO 非空。此时，RXFNEIE 位必须在进入低功耗模式之前置 1。

  – RxFIFO 已满。此时，RXFFIE 位必须在进入低功耗模式之前置 1，接收到的数据量对应于 RXFIFO 大小，并且 RXFF 标志未置 1。

  – TXFIFO 为空。此时，TXFEIE 位必须在进入低功耗模式之前置 1。

  这允许在低功耗模式下发送/接收 TXFIFO/RXFIFO 中的数据。

  为了避免发生上溢/下溢错误以及在低功耗模式下发送/接收数据，usart_wkup 中断源可以是以下事件之一：

  – 达到 TXFIFO 阈值。此时，TXFTIE 位必须在进入低功耗模式之前置 1。

  – 达到 RXFIFO 阈值。此时，RXFTIE 位必须在进入低功耗模式之前置 1。

  例如，如果唤醒时间短于在线路上接收单个字节所需的时间，则应用可将阈值设为最大 RXFIFO 大小。

  使用 RXFIFO 已满、TXFIFO 为空、RXFIFO 非空和 RXFIFO/TXFIFO 阈值中断将 MCU 从低功耗模式唤醒时，可在低功耗模式下尽可能多地进行 USART 传输，这样有助于优化功耗。

或者，也可通过 WUS 位域选择特定 **usart_wkup** 中断。

检测到唤醒事件后，WUF 标志会由硬件置 1 并在 WUFIE 位置 1 时生成一个 **usart_wkup** 中断。在这种情况下，无需 **usart_wkup** 中断，将 WUF 置 1 便足以将 MCU 从低功耗模式唤醒。

注： *在进入低功耗模式之前，请确保未进行任何 USART 传输。检查 BUSY 标志不能确保在进行数据接收时绝不会进入低功耗模式。*

WUF 标志在检测到唤醒事件时置 1，而与 MCU 处于低功耗模式还是工作模式无关。

如果在初始化和使能接收器后立即进入低功耗模式，则必须检查 REACK 位以确保 USART 确实已使能。

当 DMA 用于接收时，它必须在进入低功耗模式前禁止，并在退出低功耗模式后重新使能。

如果使能 FIFO，则只有在使能静默模式的情况下才能在地址匹配时从低功耗模式唤醒。

使用静默模式和低功耗模式

如果 USART 在进入低功耗模式前处于静默模式：

- 不得使用空闲检测时从静默模式唤醒，因为空闲检测无法在低功耗模式下工作。
- 如果使用地址匹配时从静默模式唤醒，则低功耗模式唤醒源也必须是地址匹配。如果 RXNE 标志在进入低功耗模式时置 1，则接口将在地址匹配时和从低功耗模式唤醒时保持静默模式。

注： *使能 FIFO 管理时，静默模式可与从低功耗模式唤醒搭配使用，而且没有任何限制（即，上述关于静默和低功耗模式的两点仅在 FIFO 管理禁止时有效）。*

USART 内核时钟 (usart_ker_ck) 在低功耗模式下关闭时从低功耗模式唤醒

在低功耗模式下，如果在 USART 接收线上检测到下降沿时 usart_ker_ck 时钟处于关闭状态，则 USART 接口会借助 usart_ker_ck_req 信号请求开启 usart_ker_ck 时钟。usart_ker_ck 随后用来接收帧。

如果唤醒事件通过验证，则 MCU 将从低功耗模式唤醒，并会正常进行数据接收。

如果唤醒事件未通过验证，则 usart_ker_ck 会再次关闭，MCU 不会被唤醒并保持在低功耗模式下，且内核时钟请求被释放。

以下示例显示了将唤醒事件编程为“地址匹配检测”并禁止 FIFO 管理的情况。

图 347 所示为唤醒事件通过验证时的 USART 行为。

图 347. 唤醒事件通过验证（唤醒事件 = 地址匹配，禁止 FIFO）

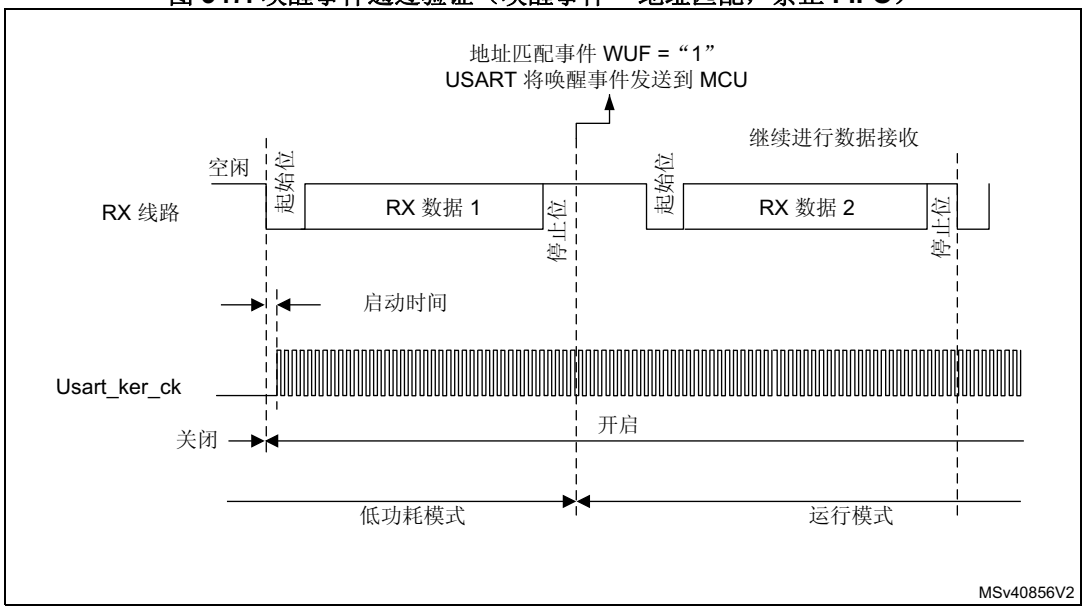
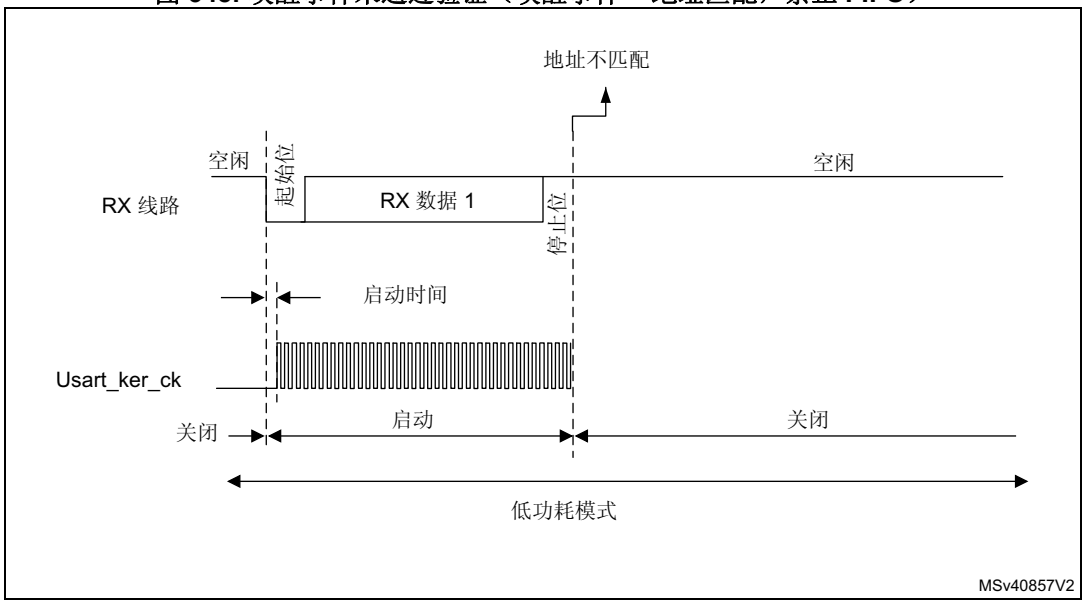


图 348 所示为唤醒事件未通过验证时的 USART 行为。

图 348. 唤醒事件未通过验证（唤醒事件 = 地址匹配，禁止 FIFO）



注： *将地址匹配或任一接收的帧用作唤醒事件时，如上两图适用。如果唤醒事件为起始位检测，则 USART 会在起始位结束时向 MCU 发送唤醒事件。*

确定允许从低功耗模式正确唤醒微控制器的最大 USART 波特率

允许从低功耗模式正确唤醒微控制器的最大波特率取决于唤醒时间参数（请参见器件数据手册）和 USART 接收器容差（请参见*第 33.5.8 节：USART 接收器对时钟偏差的容差*）。

举例来说：OVER8 = 0，M 位 = “01”，ONEBIT = 0，BRR [3:0] = 0000。

在这些条件下，根据*表 194：BRR [3:0] = 0000 时的 USART 接收器容差*，USART 接收器的容差为 3.41%。

DTRA + DQUANT + DREC + DTCL + DWU < USART 接收器的容差

$D_{WUmax} = t_{WUUSART} / (11 \times T_{bit\ Min})$

$T_{bit\ Min} = = t_{WUUSART} / (11 \times D_{WUmax})$

其中 $t_{WUUSART}$ 为从低功耗模式唤醒的时间。

考虑一种理想情况：参数 DTRA、DQUANT、DREC 和 DTCL 为 0%，则 DWU 的最大值为 3.41%。实际上，我们至少需要考虑 usart_ker_ck 不精确的情况。

例如，如果将 HSI 用作 usart_ker_ck，HSI 的不精确度为 1%，则可以得到：

$t_{WUUSART}$ = 3 µs（仅提供数值作为参考；有关正确数值，请参见器件数据手册）

D_{WUmax} = 3.41% - 1% = 2.41%

$T_{bit\ min}$ = 3 µs/ (11 x 2.41%) = 11.32 µs

结果，允许从低功耗模式正确唤醒的最大波特率为：1/11.32 µs = 88.36 K 波特。

33.6 USART 中断

在 USART 通信过程中，中断 (usart_it) 可由不同事件生成。USART 模块也可生成唤醒中断 (usart_wkup)。

有关所有 USART 中断请求的详细说明，请参见*表 197*。

表 197. USART 中断请求

中断事件	事件标志	使能控制位	中断清除方法	中断已激活	
				usart_it	usart_wkup
发送数据寄存器为空	TXE	TXEIE	TXE 在 TDR 中被写入数据时清零。	是	否
发送 FIFO 未满	TXFNF	TXFNFIE	TXFNF 在 TXFIFO 已满时清零。	是	否
发送 FIFO 为空	TXFE	TXFEIE	TXFE 在 TXFIFO 包含至少一个数据时清零，或通过将 TXFRQ 位置 1 来清零。	是	是
达到发送 FIFO 阈值	TXFT	TXFTIE	TXFT 在 TXFIFO 内容少于编程的阈值时由硬件清零	是	是
CTS 中断	CTSIF	CTSIE	CTSIF 由软件通过将 CTSCF 位置 1 来清零。	是	否
发送完成	TC	TCIE	TC 在 TDR 中被写入数据时清零，或通过将 TCCF 位置 1 来清零。	是	否
保护时间前发送完成	TCBGT	TCBGTIE	TCBGT 在 TDR 中被写入数据时清零，或通过将 TCBGTCF 位置 1 来清零。	是	否
接收数据寄存器不为空（已准备好读取数据）	RXNE	RXNEIE	RXNE 通过读取 RDR 或通过将 RXFRQ 位置 1 来清零。	是	是
接收 FIFO 非空	RXFNE	RXFNEIE	RXFNE 在 RXFIFO 为空时清零，或通过将 RXFRQ 位置 1 来清零。	是	是
接收 FIFO 已满	RXFF(1)	RXFFIE	RXFF 在 RXFIFO 至少包含一个数据时清零。	是	是
达到接收 FIFO 阈值	RXFT	RXFTIE	RXFT 在 RXFIFO 内容少于编程的阈值时由硬件清零	是	是
检测到溢出错误	ORE	RXNEIE/RXFNEIE	ORE 通过将 ORECF 位置 1 来清零。	是	否
检测到空闲线路	IDLE	IDLEIE	IDLE 通过将 IDLECF 位置 1 来清零。	是	否
奇偶校验错误	PE	PEIE	PE 通过将 PECF 位置 1 来清零。	是	否
LIN 断路	LBDF	LBDIE	LBDF 通过将 LBDCF 位置 1 来清零。	是	否

表 197. USART 中断请求（续）

中断事件	事件标志	使能控制位	中断清除方法	中断已激活	
				usart_it	usart_wkup
多缓冲区通信中的噪声错误、上溢错误和帧错误。	NE 或 ORE 或 FE	EIE	NE 通过将 NECF 位置 1 来清零。 ORE 通过将 ORECF 位置 1 来清零。 FE 标志通过将 FECF 位置 1 来清零。	是	否
字符匹配	CMF	CMIE	CMF 通过将 CMCF 位置 1 来清零。	是	否
接收器超时	RTOF	RTOFIE	RTOF 通过将 RTOCCF 位置 1 来清零。	是	否
块结束	EOBF	EOBIE	EOBF 通过将 EOBCF 位置 1 来清零。	是	否
从低功耗模式唤醒	WUF	WUFIE	WUF 通过将 WUCF 位置 1 来清零。	是	是
SPI 从器件下溢错误	UDR	EIE	UDR 通过将 UDRCF 位置 1 来清零。	是	否
达到发送 FIFO 阈值	TXFT	TXFTIE	TXFT 在 TXFIFO 内容少于编程的阈值时由硬件清零。	是	是
达到接收 FIFO 阈值	RXFT	RXFTIE	RXFT 在 RXFIFO 内容少于编程的阈值时由硬件清零。	是	是

1. 如果 USART 接收 n+1 个数据（n 表示 RXFIFO 大小，RXFIFO 中有 n 个数据，USART_RDR 中有 1 个数据），则 RXFF 标志置位。在停止模式下，未向 USART_RDR 提供时钟。因此，将不会对该寄存器进行写操作，一旦有 n 个数据被接收并写入到 RXFIFO，RXFF 中断将生效（RXFF 标志未置 1）。

33.7 USART 寄存器

有关寄存器说明中使用的缩写，请参见*第 58 页的第 1.2 节*。

33.7.1 USART 控制寄存器 1 [备用] (USART_CR1)

USART control register 1

偏移地址：0x00

复位值：0x0000 0000

同一寄存器可用于使能 FIFO 模式（本节）和禁止 FIFO 模式（下一节）的情况。

使能 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXFFIE	TXFEIE	FIFOEN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXFNFIE	TCIE	RXFNEIE	IDLEIE	TE	RE	UESM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **RXFFIE**：RXFIFO 变满时中断使能 (RXFIFO Full interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 RXFF=1 时，生成 USART 中断

位 30 **TXFEIE**：TXFIFO 为空时中断使能 (TXFIFO empty interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TXFE=1 时，生成 USART 中断

位 29 **FIFOEN**：FIFO 模式使能 (FIFO mode enable)

此位由软件置 1 和清零。

0：禁止 FIFO 模式。

1：使能 FIFO 模式。

只有在禁止 USART (UE=0) 时才能写入此位域。

注： *FIFO 模式只能在标准 UART 通信、SPI 主/从器件模式和智能卡模式下使用，不得在 IrDA 和 LIN 模式下使能。*

位 28 **M1**：字长 (Word length)

此位必须与位 12 (M0) 搭配使用来确定字长度。该位由软件置 1 或清零。

M[1:0] = “00”：1 个起始位，8 个数据位，n 个停止位

M[1:0] = “01”：1 个起始位，9 个数据位，n 个停止位

M[1:0] = “10”：1 个起始位，7 个数据位，n 个停止位

只有在禁止 USART (UE=0) 时才能写入此位。

注： *在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。*

位 27 **EOBIE**：块结束中断使能 (End of Block interrupt enable)

此位由软件置 1 和清零。

0：禁止中断。

1：USART_ISR 寄存器中的 EOBF 标志置 1 时生成 USART 中断。

注：　如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 26 **RTOIE**：接收器超时中断使能 (Receiver timeout interrupt enable)

此位由软件置 1 和清零。

0：禁止中断。

1：USART_ISR 寄存器中的 RTOF 位置 1 时生成 USART 中断。

注：　如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。第 996 页的第 33.4 节：USART 实现。

位 25:21 **DEAT[4:0]**：驱动器使能使能时间 (Driver Enable assertion time)

该 5 位值用于定义激活 DE（启动器使能）信号与起始位开始间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

只有在禁止 USART (UE=0) 时才能写入此位域。

注：　如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 20:16 **DEDT[4:0]**：驱动器使能禁止时间 (Driver Enable deassertion time)

该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE（驱动器使能）信号间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

如果在 DEDT 时间内对 USART_TDR 寄存器执行写操作，则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。

只有在禁止 USART (UE=0) 时才能写入此位域。

注：　如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 15 **OVER8**：过采样模式 (Oversampling mode)

0：16 倍过采样

1：8 倍过采样

只有在禁止 USART (UE=0) 时才能写入此位。

注：　在 LIN、IrDA 和智能卡模式下，此位必须保持清零。

位 14 **CMIE**：字符匹配中断使能 (Character match interrupt enable)

此位由软件置 1 和清零。

0：禁止中断。

1：USART_ISR 寄存器中的 CMF 位置 1 时生成 USART 中断。

位 13 **MME**：静默模式使能 (Mute mode enable)

此位可使能 USART 静默模式功能。此位置 1 时，USART 会按照 WAKE 位的定义在工作模式与静默模式之间切换。该位由软件置 1 和清零。

0：接收器永久处于活动模式。

1：接收器可在静默模式和活动模式之间切换。

位 12 **M0**：字长 (Word length)

此位与位 28 (M1) 搭配使用来确定字长度。它由软件置 1 或清零（请参见位 28 (M1) 的说明）。

只有在禁止 USART (UE=0) 时才能写入此位。

位 11 **WAKE**：接收器唤醒方法 (Receiver wakeup method)

此位用于确定 USART 静默模式的唤醒方法。该位由软件置 1 或清零。

0：空闲线路

1：地址标记

只有在禁止 USART (UE=0) 时才能写入此位域。

位 10 **PCE**：奇偶校验控制使能 (Parity control enable)

该位选择硬件奇偶校验控制（生成和检测）。使能奇偶校验控制时，计算出的奇偶校验位被插入到 MSB 位置（如果 M=1，则为第 9 位；如果 M=0，则为第 8 位），并对接收到的数据检查奇偶校验位。此位由软件置 1 和清零。一旦该位置 1，PCE 在当前字节的后面处于活动状态（在接收和发送时）。

0：禁止奇偶校验控制

1：使能奇偶校验控制

只有在禁止 USART (UE=0) 时才能写入此位域。

位 9 **PS**：奇偶校验选择 (Parity selection)

该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。将在当前字节的后面选择奇偶校验。

0：偶校验

1：奇校验

只有在禁止 USART (UE=0) 时才能写入此位域。

位 8 **PEIE**：PE 中断使能 (PE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 PE=1 时，生成 USART 中断

位 7 **TXFNFIE**：TXFIFO 未满中断使能 (TXFIFO not full interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TXFNF = 1 时，生成 USART 中断

位 6 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TC=1 时，生成 USART 中断

位 5 **RXFNEIE**：RXFIFO 非空中断使能 (RXFIFO not empty interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 ORE=1 或 RXFNE=1 时，生成 USART 中断

位 4 **IDLEIE**：IDLE 中断使能 (IDLE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 IDLE=1 时，生成 USART 中断

位 3 **TE**：发送器使能 (Transmitter enable)

该位使能发送器。该位由软件置 1 和清零。

0：禁止发送器

1：使能发送器

注： *除了在智能卡模式下以外，传送期间 TE 位上的低电平脉冲（“0”后紧跟的是“1”）会在当前字的后面发送一个报头（空闲线路）。为生成空闲字符，TE 不能立即写入“1”。为确保所需的持续时间，软件可轮询 USART_ISR 寄存器中的 TEACK 位。*

在智能卡模式下，当 TE 置 1 时，在发送开始前存在 1 位的时间延迟。

位 2 **RE**：接收器使能 (Receiver enable)

该位使能接收器。该位由软件置 1 和清零。

0：禁止接收器

1：使能接收器并开始搜索起始位

位 1 **UESM**：低功耗模式下的 USART 使能 (USART enable in low-power mode)

当此位清零时，USART 无法将 MCU 从低功耗模式唤醒。

当此位置 1 时，USART 可将 MCU 从低功耗模式唤醒。

此位由软件置 1 和清零。

0：USART 无法将 MCU 从低功耗模式唤醒。

1：USART 能够将 MCU 从低功耗模式唤醒。

注： *建议在进入低功耗模式前将 UESM 位置 1，并在退出低功耗模式时将其清零。*

如果 USART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 0 **UE**：USART 使能 (USART enable)

此位清零后，USART 预分频器和输出将立即停止，并丢弃所有当前操作。USART 配置会保留，而所有的 USART_ISR 状态标志均会复位。此位由软件置 1 和清零。

0：禁止 USART 预分频器和输出，低功耗模式

1：使能 USART

注： *为进入低功耗模式而不在线路上生成错误，之前必须复位 TE 位，并且软件必须等待 USART_ISR 中的 TC 位置 1 后才能复位 UE 位。*

UE = 0 时也会复位 DMA 请求，因必须在复位 UE 位前禁止 DMA 通道。

在智能卡模式下 (SCEN = 1)，无论 UE 位值为何，当 CLKEN = 1 时，SCLK 始终可用。

33.7.2 USART 控制寄存器 1 [备用] (USART_CR1)

USART control register 1

偏移地址：0x00

复位值：0x0000 0000

同一寄存器可用于使能 FIFO 模式（上一节）和禁止 FIFO 模式（本节）的情况。

禁止 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	FIFO EN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]				
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29 **FIFOEN**：FIFO 模式使能 (FIFO mode enable)

此位由软件置 1 和清零。

0：禁止 FIFO 模式。

1：使能 FIFO 模式。

只有在禁止 USART (UE=0) 时才能写入此位域。

注： FIFO 模式只能在标准 UART 通信、SPI 主/从器件模式和智能卡模式下使用，不得在 IrDA 和 LIN 模式下使能。

位 28 **M1**：字长 (Word length)

此位必须与位 12 (M0) 搭配使用来确定字长度。该位由软件置 1 或清零。

M[1:0] = “00”： 1 个起始位，8 个数据位，n 个停止位

M[1:0] = “01”： 1 个起始位，9 个数据位，n 个停止位

M[1:0] = “10”： 1 个起始位，7 个数据位，n 个停止位

只有在禁止 USART (UE=0) 时才能写入此位。

注： 在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。

位 27 **EOBIE**：块结束中断使能 (End of Block interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：USART_ISR 寄存器中的 EOBF 标志置 1 时生成 USART 中断

注： 如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 26 **RTOIE**：接收器超时中断使能 (Receiver timeout interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：USART_ISR 寄存器中的 RTOF 位置 1 时生成 USART 中断

注： 如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。第 996 页的第 33.4 节：USART 实现。

位 25:21 **DEAT[4:0]**：驱动器使能使能时间 (Driver Enable assertion time)

该 5 位值用于定义激活 DE（启动器使能）信号与起始位开始间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

只有在禁止 USART (UE=0) 时才能写入此位域。

注：　如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 20:16 **DEDT[4:0]**：驱动器使能禁止时间 (Driver Enable deassertion time)

该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE（驱动器使能）信号间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

如果在 DEDT 时间内对 USART_TDR 寄存器执行写操作，则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。

只有在禁止 USART (UE=0) 时才能写入此位域。

注：　如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 15 **OVER8**：过采样模式 (Oversampling mode)

0：16 倍过采样

1：8 倍过采样

只有在禁止 USART (UE=0) 时才能写入此位。

注：　在 LIN、IrDA 和智能卡模式下，此位必须保持清零。

位 14 **CMIE**：字符匹配中断使能 (Character match interrupt enable)

此位由软件置 1 和清零。

0：禁止中断。

1：USART_ISR 寄存器中的 CMF 位置 1 时生成 USART 中断。

位 13 **MME**：静默模式使能 (Mute mode enable)

此位可使能 USART 静默模式功能。此位置 1 时，USART 会按照 WAKE 位的定义在工作模式与静默模式之间切换。该位由软件置 1 和清零。

0：接收器永久处于活动模式。

1：接收器可在静默模式和活动模式之间切换。

位 12 **M0**：字长 (Word length)

此位与位 28 (M1) 搭配使用来确定字长度。它由软件置 1 或清零（请参见位 28 (M1) 的说明）。

只有在禁止 USART (UE=0) 时才能写入此位。

位 11 **WAKE**：接收器唤醒方法 (Receiver wakeup method)

此位用于确定 USART 静默模式的唤醒方法。该位由软件置 1 或清零。

0：空闲线路

1：地址标记

只有在禁止 USART (UE=0) 时才能写入此位域。

位 10 **PCE**：奇偶校验控制使能 (Parity control enable)

该位选择硬件奇偶校验控制（生成和检测）。使能奇偶校验控制时，计算出的奇偶校验位被插入到 MSB 位置（如果 M=1，则为第 9 位；如果 M=0，则为第 8 位），并对接收到的数据检查奇偶校验位。此位由软件置 1 和清零。一旦该位置 1，PCE 在当前字节的后面处于活动状态（在接收和发送时）。

0：禁止奇偶校验控制

1：使能奇偶校验控制

只有在禁止 USART (UE=0) 时才能写入此位域。

位 9 **PS**：奇偶校验选择 (Parity selection)

该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。将在当前字节的后面选择奇偶校验。

0：偶校验

1：奇校验

只有在禁止 USART (UE=0) 时才能写入此位域。

位 8 **PEIE**：PE 中断使能 (PE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 PE=1 时，生成 USART 中断

位 7 **TXEIE**：发送数据寄存器为空 (Transmit data register empty)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TXE = 1 时，生成 USART 中断

位 6 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TC=1 时，生成 USART 中断

位 5 **RXNEIE**：接收数据寄存器非空 (Receive data register not empty)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 ORE=1 或 RXNE=1 时，生成 USART 中断

位 4 **IDLEIE**：IDLE 中断使能 (IDLE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 IDLE=1 时，生成 USART 中断

位 3 **TE**：发送器使能 (Transmitter enable)

该位使能发送器。该位由软件置 1 和清零。

0：禁止发送器

1：使能发送器

注：　除了在智能卡模式下以外，传送期间 TE 位上的低电平脉冲（“0”后紧跟的是“1”）会在当前字的后面发送一个报头（空闲线路）。为生成空闲字符，TE 不能立即写入“1”。为确保所需的持续时间，软件可轮询 USART_ISR 寄存器中的 TEACK 位。

在智能卡模式下，当 TE 置 1 时，在发送开始前存在 1 位的时间延迟。

位 2 **RE**：接收器使能 (Receiver enable)

该位使能接收器。该位由软件置 1 和清零。

0：禁止接收器

1：使能接收器并开始搜索起始位

位 1 **UESM**：低功耗模式下的 USART 使能 (USART enable in low-power mode)

当此位清零时，USART 无法将 MCU 从低功耗模式唤醒。

当此位置 1 时，USART 可将 MCU 从低功耗模式唤醒。

此位由软件置 1 和清零。

0：USART 无法将 MCU 从低功耗模式唤醒。

1：USART 能够将 MCU 从低功耗模式唤醒。

注：　建议在进入低功耗模式前将 UESM 位置 1，并在退出低功耗模式时将其清零。

如果 USART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 0 **UE**：USART 使能 (USART enable)

此位清零后，USART 预分频器和输出将立即停止，并丢弃所有当前操作。USART 配置会保留，而所有的 USART_ISR 状态标志均会复位。此位由软件置 1 和清零。

0：禁止 USART 预分频器和输出，低功耗模式

1：使能 USART

注： 为进入低功耗模式而不在线路上生成错误，之前必须复位 TE 位，并且软件必须等待 USART_ISR 中的 TC 位置 1 后才能复位 UE 位。

UE = 0 时也会复位 DMA 请求，因必须在复位 UE 位前禁止 DMA 通道。

在智能卡模式下 (SCEN = 1)，无论 UE 位值为何，当 CLKEN = 1 时，SCLK 始终可用。

33.7.3 USART 控制寄存器 2 (USART_CR2)

USART control register 2

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
ADD[7:0]								RTOEN	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV
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
SWAP	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	DIS_NSS	Res.	Res.	SLVEN
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw			rw

位 31:24 **ADD[7:0]**：USART 节点的地址 (Address of the USART node)

ADD[7:4]：

这些位用于指定 USART 节点的地址或要识别的字符代码。

它们在多处理器通信时于静默模式或低功耗模式下使用，以通过 7 位地址标记检测唤醒 MCU。发送器发送字符的 MSB 应为 1。它们还可用于正常接收和静默模式无效时的字符检测（例如，ModBus 协议中的块结束检测）。这种情况下，接收到的整个字符（8 位）将与 ADD[7:0] 值进行比较，如果匹配，CMF 标志将置 1。

仅在禁止接收 (RE = 0) 或禁止 USART (UE=0) 时才能写入这些位。

ADD[3:0]：

这些位用于指定 USART 节点的地址或要识别的字符代码。

它们在多处理器通信时于静默模式或低功耗模式下使用，以通过地址标记检测进行唤醒。

仅在禁止接收 (RE = 0) 或禁止 USART (UE=0) 时才能写入这些位。

位 23 **RTOEN**：接收器超时使能 (Receiver timeout enable)

此位由软件置 1 和清零。

0：禁止接收器超时功能。

1：使能接收器超时功能。

使能此功能后，如果 RX 线路在 RTOR（接收器超时寄存器）中编程的持续时间内处于空闲状态（无接收），则 USART_ISR 寄存器中的 RTOF 标志置 1。

注： 如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 22:21 **ABRMOD[1:0]**：自动波特率模式 (Auto baud rate mode)

这些位将由软件置 1 和清零。

00：通过测量起始位检测波特率。

01：下降沿到下降沿的测量（接收到的帧必须以一个等于 1 的位开头，即帧 = 10xxxxxx）。

10：0x7F 帧检测。

11：0x55 帧检测。

仅在 ABREN = 0 时或禁止 USART (UE=0) 时才能写入该位域。

注：　如果 DATAINV=1 且/或 MSBFIRST=1，这些模式必须与在线路上时相同，例如 MSBFIRST 的 0xAA。

如果 USART 不支持自动波特率功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 20 **ABREN**：自动波特率使能 (Auto baud rate enable)

此位由软件置 1 和清零。

0：禁止自动波特率检测。

1：使能自动波特率检测。

注：　如果 USART 不支持自动波特率功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 19 **MSBFIRST**：最高有效位在前 (Most significant bit first)

此位由软件置 1 和清零。

0：发送/接收数据时位 0 在前，后跟起始位。

1：发送/接收数据时 MSB（位 7/8）在前，后跟起始位。

只有在禁止 USART (UE=0) 时才能写入此位域。

位 18 **DATAINV**：二进制数据反向 (Binary data inversion)

此位由软件置 1 和清零。

0：按正/正向逻辑发送/接收数据寄存器中的逻辑数据。（1=H，0=L）。

1：按负/反向逻辑发送/接收数据寄存器中的逻辑数据。（1=L，0=H）。奇偶校验位也取反。

只有在禁止 USART (UE=0) 时才能写入此位域。

位 17 **TXINV**：TX 引脚有效电平反向 (TX pin active level inversion)

此位由软件置 1 和清零。

0：TX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作。

1：对 TX 引脚信号值取反。（V_{DD} = 0/标记，Gnd=1/空闲）。

允许在 TX 线路上使用外部反相器。

只有在禁止 USART (UE=0) 时才能写入此位域。

位 16 **RXINV**：RX 引脚有效电平反向 (RX pin active level inversion)

此位由软件置 1 和清零。

0：RX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作。

1：对 RX 引脚信号值取反。（V_{DD} = 0/标记，Gnd=1/空闲）。

允许在 RX 线路上使用外部反相器。

只有在禁止 USART (UE=0) 时才能写入此位域。

位 15 **SWAP**：交换 TX/RX 引脚 (Swap TX/RX pins)

此位由软件置 1 和清零。

0：按标准引脚排列定义使用 TX/RX 引脚。

1：交换 TX 和 RX 引脚功能。允许在与另一个 UART 的交叉连接时工作。

只有在禁止 USART (UE=0) 时才能写入此位域。

位 14 **LINEN**：LIN 模式使能 (LIN mode enable)

此位由软件置 1 和清零。

0：禁止 LIN 模式

1：使能 LIN 模式

LIN 模式可以使用 USART_CR1 寄存器中的 SBKRQ 位发送 LIN 同步断路（13 个低位），并可检测 LIN 同步断路。

只有在禁止 USART (UE=0) 时才能写入此位域。

注：　如果 USART 不支持 LIN 模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 13:12 **STOP[1:0]**：停止位 (STOP bits)

这些位用于编程停止位。

00：1 个停止位

01：0.5 个停止位

10：2 个停止位

11：1.5 个停止位

只有在禁止 USART (UE=0) 时才能写入此位域。

位 11 **CLKEN**：时钟使能 (Clock enable)

该位允许用户使能 SCLK 引脚。

0：禁止 SCLK 引脚

1：使能 SCLK 引脚

只有在禁止 USART (UE=0) 时才能写入此位。

注：　如果既不支持同步模式，也不支持智能卡模式，则该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

在智能卡模式下，为了向智能卡正确提供 SCLK 时钟，必须按以下步骤操作：

UE = 0

SCEN = 1

GTPR 配置

CLKEN= 1

UE = 1

位 10 **CPOL**：时钟极性 (Clock polarity)

该位允许用户在同步模式下选择 SCLK 引脚上时钟输出的极性。它与 CPHA 位结合使用可获得所需的时钟/数据关系。

0：空闲时 SCLK 引脚为低电平

1：空闲时 SCLK 引脚为高电平

只有在禁止 USART (UE=0) 时才能写入此位。

注：　如果不支持同步模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 9 **CPHA**：时钟相位 (Clock phase)

此位用于在同步模式下选择 SCLK 引脚上时钟输出的相位。它与 CPOL 位结合使用可获得所需的时钟/数据关系（请参见*图 328* 和*图 329*）

0：从第一个时钟边沿开始采样数据

1：从第二个时钟边沿开始采样数据

只有在禁止 USART (UE=0) 时才能写入此位。

注：　如果不支持同步模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 8 **LBCL**：最后一个位时钟脉冲 (Last bit clock pulse)

此位用于在同步模式下选择与发送的最后一个数据位 (MSB) 关联的时钟脉冲是否必须在 SCLK 引脚上输出。

0：最后一个数据位的时钟脉冲不在 SCLK 引脚上输出

1：最后一个数据位的时钟脉冲在 SCLK 引脚上输出

注意： 最后一位为发送的第 7 个、第 8 个或第 9 个数据位，具体取决于 USART_CR1 寄存器中 M 位所选择的7位、8 位或 9 位格式。

只有在禁止 USART (UE=0) 时才能写入此位。

注： 如果不支持同步模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 7 保留，必须保持复位值。

位 6 **LBDIE**：LIN 断路检测中断使能 (LIN break detection interrupt enable)

断路中断屏蔽（使用中断分隔符进行中断检测）

0：禁止中断

1：当 USART_ISR 寄存器中 LBDF = 1 时，生成中断

注： 如果不支持 LIN 模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 5 **LBDL**：LIN 断路检测长度 (LIN break detection length)

该位用于选择 11 位中断检测或 10 位中断检测。

0：10 位中断检测

1：11 位中断检测

只有在禁止 USART (UE=0) 时才能写入此位。

注： 如果不支持 LIN 模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 4 **ADDM7**：7 位地址检测/4 位地址检测 (7-bit Address Detection/4-bit Address Detection)

此位用于选择 4 位地址检测或 7 位地址检测。

0：4 位地址检测

1：7 位地址检测（在 8 位数据模式下）

只有在禁止 USART (UE=0) 时才能写入该位

注： 在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。

位 3 **DIS_NSS**：

当 DIS_NSS 位置 1 时，忽略 NSS 引脚输入。

0：SPI 从器件选择取决于 NSS 输入引脚。

1：始终选择 SPI 从器件，忽略 NSS 输入引脚。

注： 不支持 SPI 从器件模式时，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 2:1 保留，必须保持复位值。

位 0 **SLVEN**：同步从模式使能 (Synchronous Slave mode enable)

SLVEN 位置 1 时，使能同步从模式。

0：禁止从模式。

1：使能从模式。

注： 不支持 SPI 从器件模式时，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

注： 使能发送器时不应对 CPOL、CPHA 和 LBCL 位进行写操作。

33.7.4 USART 控制寄存器 3 (USART_CR3)

USART control register 3

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXFTCFG[2:0]			RXF TIE	RXFTCFG[2:0]			TCBG TIE	TXFTIE	WUFIE	WUS[1:0]		SCARCNT[2:0]			Res.
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
DEP	DEM	DDRE	OVR DIS	ONE BIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HD SEL	IRLP	IREN	EIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 **TXFTCFG[2:0]**：TXFIFO 阈值配置 (TXFIFO threshold configuration)

000：TXFIFO 达到其深度的 1/8
001：TXFIFO 达到其深度的 1/4
010：TXFIFO 达到其深度的 1/2
011：TXFIFO 达到其深度的 3/4
100：TXFIFO 达到其深度的 7/8
101：TXFIFO 变空
其余组合：保留

位 28 **RXFTIE**：RXFIFO 阈值中断使能 (RXFIFO threshold interrupt enable)

此位由软件置 1 和清零。
0：禁止中断。
1：当接收 FIFO 达到 RXFTCFG 中编程的阈值时，生成 USART 中断。

位 27:25 **RXFTCFG[2:0]**：接收 FIFO 阈值配置 (Receive FIFO threshold configuration)

000：接收 FIFO 达到其深度的 1/8
001：接收 FIFO 达到其深度的 1/4
010：接收 FIFO 达到其深度的 1/2
011：接收 FIFO 达到其深度的 3/4
100：接收 FIFO 达到其深度的 7/8
101：接收 FIFO 已满
其余组合：保留

位 24 **TCBGTIE**：保护时间前发送完成中断使能 (Transmission Complete before guard time, interrupt enable)

此位由软件置 1 和清零。
0：禁止中断
1：当 USART_ISR 寄存器中的 TCBGT=1 时，生成 USART 中断

注：　如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 23 **TXFTIE**：TXFIFO 阈值中断使能 (TXFIFO threshold interrupt enable)

此位由软件置 1 和清零。
0：禁止中断。
1：当 TXFIFO 达到 TXFTCFG 中编程的阈值时，生成 USART 中断。

位 22 **WUFIE**：从低功耗模式唤醒中断使能 (Wakeup from low-power mode interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 WUF=1 时，生成 USART 中断

注： *WUFIE 必须在进入低功耗模式前置 1。*

如果 USART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 21:20 **WUS[1:0]**：从低功耗模式唤醒中断标志选择 (Wakeup from low-power mode interrupt flag selection)

该位域用于指定激活 WUF（从低功耗模式唤醒标志）的事件。

00：WUF 在地址匹配时激活（按 ADD[7:0] 和 ADDM7 所定义）

01：保留

10：WUF 在起始位检测时激活

11：WUF 在 RXNE/RXFNE 时激活

只有在禁止 USART (UE=0) 时才能写入此位域。

如果 USART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 19:17 **SCARCNT[2:0]**：智能卡自动重试计数 (Smartcard auto-retry count)

此位域用于指定智能卡模式下发送和接收的重试次数。

在发送模式下，此位域用于指定生成发送错误（FE 位置 1）前自动重新发送的重试次数。

在接收模式下，此位域用于指定生成接收错误（RXNE/RXFNE 位和 PE 位置 1）前错误接收尝试的次数。

只有在禁止 USART (UE=0) 时才能编程此位域。

使能 USART (UE=1) 时，此位域只能写入 0x0，以停止重新发送。

0x0：禁止重新发送——发送模式下不会自动重新发送。

0x1 到 0x7：自动重新发送的尝试次数（发出错误信号前）

注： *如果不支持智能卡模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。*

位 16 保留，必须保持复位值。

位 15 **DEP**：驱动器使能极性选择 (Driver enable polarity selection)

0：DE 信号高电平有效。

1：DE 信号低电平有效。

只有在禁止 USART (UE=0) 时才能写入此位。

注： *如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。*

位 14 **DEM**：驱动器使能模式 (Driver enable mode)

此位用于通过 DE 信号激活外部收发器控制。

0：禁止 DE 功能。

1：使能 DE 功能。DE 信号在 RTS 引脚上输出。

只有在禁止 USART (UE=0) 时才能写入此位。

注： *如果不支持驱动器使能功能，该位保留且必须保持复位值。第 996 页的第 33.4 节：USART 实现。*

位 13 **DDRE**：接收出错时的 DMA 禁止 (DMA Disable on Reception Error)

0：接收出错时不禁止 DMA。相应的错误标志置 1，但 RXNE 保持为 0 以防止上溢。因此，将不使能 DMA 请求，从而不会传送错误数据（无 DMA 请求），但会传送接收到的下一个正确数据。（用于智能卡模式）

1：接收出错后禁止 DMA。相应的错误标志以及 RXNE 均置 1。屏蔽 DMA 请求，直到错误标志清零。这意味着软件必须首先禁止 DMA 请求 (DMAR = 0) 或者将 RXNE（使能 FIFO 模式时为 RXFNE）清零，然后再将错误标志清零。

只有在禁止 USART (UE=0) 时才能写入此位。

注：　接收错误包括：奇偶校验错误、帧错误或噪声错误。

位 12 **OVRDIS**：上溢禁止 (Overrun Disable)

此位用于禁止接收上溢检测。

0：接收新数据前未读取已接收的数据时，上溢错误标志 ORE 置 1。

1：禁止上溢功能。如果在 RXNE 标志仍置 1 时接收到新数据，

则 ORE 标志不会置 1，且新接收的数据会覆盖 USART_RDR 寄存器之前的内容。使能 FIFO 模式时，RXFIFO 将被旁路，数据将直接写入 USART_RDR 寄存器中。即使在使能 FIFO 管理时，也将使用 RXNE 标志。

只有在禁止 USART (UE=0) 时才能写入此位。

注：　此控制位用于检查通信流而不会读取数据。

位 11 **ONEBIT**：一个采样位方法使能 (One sample bit method enable)

该位允许用户选择采样方法。选择一个采样位方法后，将禁止噪声检测标志 (NE)。

0：三个采样位方法

1：一个采样位方法

只有在禁止 USART (UE=0) 时才能写入此位。

位 10 **CTSIE**：CTS 中断使能 (CTS interrupt enable)

0：禁止中断

1：当 USART_ISR 寄存器中 CTSIF = 1 时，生成中断

注：　如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 9 **CTSE**：CTS 使能 (CTS enable)

0：禁止 CTS 硬件流控制。

1：使能 CTS 模式，仅当 nCTS 输入有效（连接到 0）时才发送数据。如果在发送数据时使 nCTS 输入无效，会在停止之前完成发送。如果使 nCTS 有效时数据已写入数据寄存器，则将延迟发送，直到 nCTS 有效。

只有在禁止 USART (UE=0) 时才能写入该位。

注：　如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 8 **RTSE**：RTS 使能 (RTS enable)

0：禁止 RTS 硬件流控制。

1：使能 RTS 输出，仅当接收缓冲区中有空间时才会请求数据。发送完当前字符后应停止发送数据。可以接收数据时使 nRTS 输出有效（拉至 0）。

只有在禁止 USART (UE=0) 时才能写入此位。

注：　如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 7 **DMAT**：DMA 使能发送器 (DMA enable transmitter)

该位由软件置 1/复位。

1：针对发送使能 DMA 模式

0：针对发送禁止 DMA 模式

位 6 **DMAR**：DMA 使能接收器 (DMA enable receiver)

该位由软件置 1/复位。

1：针对接收使能 DMA 模式

0：针对接收禁止 DMA 模式

位 5 **SCEN**：智能卡模式使能 (Smartcard mode enable)

该位用于使能智能卡模式。

0：禁止智能卡模式

1：使能智能卡模式

只有在禁止 USART (UE=0) 时才能写入此位域。

注：　如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 4 **NACK**：智能卡 NACK 使能 (Smartcard NACK enable)

0：出现奇偶校验错误时禁止 NACK 发送

1：出现奇偶校验错误时使能 NACK 发送

只有在禁止 USART (UE=0) 时才能写入此位域。

注：　如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 3 **HDSEL**：半双工选择 (Half-duplex selection)

选择单线半双工模式

0：未选择半双工模式

1：选择半双工模式

只有在禁止 USART (UE=0) 时才能写入此位。

位 2 **IRLP**：IrDA 低功耗模式 (IrDA low-power)

该位用于选择正常模式和低功耗 IrDA 模式

0：正常模式

1：低功耗模式

只有在禁止 USART (UE=0) 时才能写入此位。

注：　如果不支持 IrDA 模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 1 **IREN**：IrDA 模式使能 (IrDA mode enable)

此位由软件置 1 和清零。

0：禁止 IrDA

1：使能 IrDA

只有在禁止 USART (UE=0) 时才能写入此位。

注：　如果不支持 IrDA 模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 0 **EIE**：错误中断使能 (Error interrupt enable)

如果出现帧错误、上溢错误、噪声标志或 SPI 从器件下溢错误（USART_ISR 寄存器中的 FE=1、ORE=1、NE=1 或 UDR = 1），则需要使用错误中断使能位来使能中断生成。

0：禁止中断。

1：USART_ISR 寄存器中的 FE=1、ORE=1、NE=1 或 UDR = 1（在 SPI 从器件模式下）时，生成中断。

33.7.5 USART 波特率寄存器 (USART_BRR)

USART baud rate register

只有在禁止 USART (UE=0) 时才能写入此寄存器。在自动波特率检测模式下，该位由硬件自动更新。

偏移地址：0x0C

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 **BRR[15:0]**：USART 波特率 (USART baud rate)

BRR[15:4]

BRR[15:4] = USARTDIV[15:4]

BRR[3:0]

当 OVER8 = 0 时，BRR[3:0] = USARTDIV[3:0]。

当 OVER8 = 1 时：

BRR[2:0] = USARTDIV[3:0] 右移 1 位。

BRR[3] 必须保持清零。

33.7.6 USART 保护时间和预分频器寄存器 (USART_GTPR)

USART guard time and prescaler register

偏移地址：0x10

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:8 **GT[7:0]**：保护时间值 (Guard time value)

此位域用于编程保护时间值（以波特时钟周期数为单位）。

该位用于智能卡模式。经过此保护时间后，发送完成标志置 1。

只有在禁止 USART (UE=0) 时才能写入此位域。

注： 如果不支持智能卡模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 7:0 **PSC[7:0]**：预分频器值 (Prescaler value)

在 IrDA 低功耗和正常的 IrDA 模式下：

PSC[7:0] = IrDA 正常和低功耗波特率

用于编程预分频器，进行 USART 源时钟分频以获得低功耗频率：

使用寄存器中给出的值（8 个有效位）对源时钟进行分频：

00000000：保留 - 不编程此值

00000001：源时钟 1 分频

00000010：源时钟 2 分频

...

在智能卡模式下：

PSC[4:0]：预分频器值 (Prescaler value)

用于编程预分频器，进行 USART 源时钟分频以提供智能卡时钟。

将寄存器中给出的值（5 个有效位）乘以 2 得出源时钟频率的分频系数：

00000：保留 - 不编程此值

00001：源时钟 2 分频

00010：源时钟 4 分频

00011：源时钟 6 分频

...

只有在禁止 USART (UE=0) 时才能写入此位域。

注： 如果使用智能卡模式，则位 [7:5] 必须保持清零。

不支持智能卡和 IrDA 模式时，该位域保留并由硬件强制清零。请参见第 996 页的第 33.4 节：USART 实现。

33.7.7 USART 接收器超时寄存器 (USART_RTOR)

USART receiver timeout register

偏移地址：0x14

复位值：0x0000 0000

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

位 31:24 **BLEN[7:0]**：块长度 (Block Length)

此位域用于提供智能卡 T=1 接收下的块长度。其值等于信息字符数 + 结尾字段的长度 (1-LEC/2-CRC) - 1。

例如：

BLEN = 0 -> 0 个信息字符 + LEC

BLEN = 1 -> 0 个信息字符 + CRC

BLEN = 255 -> 254 个信息字符 + CRC（总共 256 个字符）

在智能卡模式下，块长度计数器在 TXE=0（使能 FIFO 模式时为 TXFE = 0）时复位。

此位域也可用于其他模式。这种情况下，块长度计数器在 RE=0（禁止接收器）和/或 EOBCF 位写入 1 时复位。

注： 块接收开始后可编程此值（使用起始字段中 LEN 字符中的数据）。每个接收到的块只能对此值编程一次。

位 23:0 **RTO[23:0]**：接收器超时值 (Receiver timeout value)

该位域根据 RX 线上没有活动的位数分配接收器超时值。

在标准模式下，如果在接收到最后一个字符后，在 RTO 值对应的时间内未检测到新的起始位，则 RTOF 标志置 1。

在智能卡模式下，此值用于实施 CWT 和 BWT。有关更多详细信息，请参见智能卡章节。在标准模式下，从接收到的最后一个字符的起始位开始执行 CWT/BWT 测量。

注： 每个接收到的字符只能对此值编程一次。

注： 可以实时写入 RTOR。如果新值小于或等于计数器的值，RTOF 标志置 1。

如果不支持接收器超时功能，此寄存器保留并由硬件强制为“0x00000000”。请参见第 996 页的第 33.4 节：USART 实现。

33.7.8 USART 请求寄存器 (USART_RQR)

USART request register

偏移地址：0x18

复位值：0x0000 0000

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

位 31:5 保留，必须保持复位值。

位 4 **TXFRQ**：发送数据刷新请求 (Transmit data flush request)

禁止 FIFO 模式时，向该位写入“1”会将 TXE 标志置 1。这可丢弃发送数据。由于错误 (NACK) 而未发送数据且 USART_ISR 寄存器中的 FE 标志有效时，只能在智能卡模式下使用此位。如果 USART 不支持智能卡模式，该位保留且必须保持复位值。

使能 FIFO 时，TXFRQ 位置 1 以清空整个 FIFO。这会将 TXFE 标志（发送 FIFO 为空，USART_ISR 寄存器中的位 23）置 1。在 UART 模式和智能卡模式下都支持清空发送 FIFO。

注： 在 FIFO 模式下，TXFNF 标志在清空请求期间复位，直到 TxFIFO 为空，以确保数据寄存器中没有写入数据。

位 3 **RXFRQ**：接收数据刷新请求 (Receive data flush request)

向该位写入 1 将清空整个接收 FIFO（即，将 RXFNE 位清零）。

这可以丢弃接收的数据而不对其执行读取操作，并避免发生上溢情况。

位 2 **MMRQ**：静默模式请求 (Mute mode request)

向此位写入 1 可将 USART 置于静默模式，并将 RWU 标志复位。

位 1 **SBKRQ**：发送中断请求 (Send break request)

向此位写入 1 可将 SBKF 标志置 1 并在发送设备可用后立即请求在线路上发送 BREAK。

注： 如果应用需要在之前插入的所有数据（包括尚未发送的数据）后发送中断字符，软件应等到 TXE 标志使能后将 SBKRQ 位置 1。

位 0 **ABRRQ**：自动波特率请求 (Auto baud rate request)

向此位写入 1 可复位 USART_ISR 中的 ABRF 标志，并请求对下一个接收到的数据帧进行自动波特率测量。

注： 如果 USART 不支持自动波特率功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

33.7.9 USART 中断和状态寄存器 [备用] (USART_ISR)

USART interrupt and status register

偏移地址：0x1C

复位值：0x0XX0 00C0

使能 FIFO/智能卡模式时，XX = 28

使能 FIFO 模式且禁止智能卡模式时，XX = 08

同一寄存器可用于使能 FIFO 模式（本节）和禁止 FIFO 模式（下一节）的情况。

使能 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXFT	RXFT	TCBGT	RXFF	TXFE	RE ACK	TE ACK	WUF	RWU	SBKF	CMF	BUSY
				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
ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXFNF	TC	RXFNE	IDLE	ORE	NE	FE	PE
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27 **TXFT**：TXFIFO 阈值标志 (TXFIFO threshold flag)

当 TXFIFO 达到在 USART_CR3 寄存器的 TXFTCFG 中编程的阈值时（即 TXFIFO 包含 TXFTCFG 个空位置），该位由硬件置 1。如果 USART_CR3 寄存器中的 TXFTIE 位 =1（位 31），则会生成中断。

0：TXFIFO 未达到编程的阈值。

1：TXFIFO 已达到编程的阈值。

位 26 **RXFT**：RXFIFO 阈值标志 (RXFIFO threshold flag)

达到在 USART_CR3 寄存器的 RXFTCFG 中编程的阈值时，该位由硬件置 1。这意味着，接收 FIFO 中有 (RXFTCFG - 1) 个数据，USART_RDR 寄存器中有一个数据。如果 USART_CR3 寄存器中的 RXFTIE 位 = 1（位 27），则会生成中断。

0：接收 FIFO 未达到编程的阈值。

1：接收 FIFO 已达到编程的阈值。

注： 当 RXFTCFG 阈值配置为“101”时，如果存在 16 个数据（即 RXFIFO 中有 15 个数据，USART_RDR 中有 1 个数据），则 RXFT 标志将置 1。因此，接收到的第 17 个数据不会导致上溢错误。接收到第 18 个数据后会发生上溢错误。

位 25 **TCBGT**：保护时间前发送完成标志 (Transmission complete before guard time flag)

当写入到 USART_TDR 中的最后一个数据已正确从移位寄存器中发出时，该位置 1。

如果包含数据的帧完成发送，并且智能卡未发回任何 NACK，则该位在智能卡模式下由硬件置 1。如果 USART_CR3 寄存器中的 TCBGTIE=1，则生成中断。

此位由软件清零，方法是向 USART_ICR 寄存器中的 TCBGTCF 写入 1 或向 USART_TDR 寄存器执行写操作。

0：发送未完成或发送未成功完成（即，从智能卡接收到 NACK）。

1：发送成功完成（在保护时间结束之前完成，智能卡未发送 NACK）。

注： *如果 USART 不支持智能卡模式，该位保留且保持复位值。如果 USART 支持智能卡模式并使能智能卡模式，则 TCBGT 复位值为 “1”。请参见第 996 页的第 33.4 节：USART 实现。*

位 24 **RXFF**：RXFIFO 已满 (RXFIFO Full)

当接收到的数据量对应于 RXFIFO 大小 + 1（RXFIFO 已满 + USART_RDR 寄存器中的 1 个数据）时，该位由硬件置 1。

如果 USART_CR1 寄存器中 RXFFIE 位 = 1，则会生成中断。

0：RXFIFO 未满。

1：RXFIFO 已满。

位 23 **TXFE**：TXFIFO 为空 (TXFIFO empty)

当 TXFIFO 为空时，该位由硬件置 1。当 TXFIFO 包含至少一个数据时，该标志被清零。也可以通过向 USART_RQR 寄存器中的位 TXFRQ（位 4）写入 1 将 TXFE 标志置 1。

如果 USART_CR1 寄存器中的 TXFEIE 位 = 1（位 30），则会生成中断。

0：TXFIFO 非空。

1：TXFIFO 为空。

位 22 **REACK**：接收使能确认标志 (Receive enable acknowledge flag)

USART 采用接收使能值时，通过硬件将此位置 1/复位。

此位可用于验证 USART 是否准备好在进入低功耗模式前接收数据。

注： *如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 996 页的第 33.4 节：USART 实现。*

位 21 **TEACK**：发送使能确认标志 (Transmit enable acknowledge flag)

USART 采用发送使能值时，通过硬件将此位置 1/复位。

通过写入 TE=0 生成空闲帧请求，然后在 USART_CR1 寄存器中写入 TE=1 以遵循 TE=0 最短周期时，可使用此位。

位 20 **WUF**：从低功耗模式唤醒标志 (Wakeup from low-power mode flag)

当检测到唤醒事件时，此位由硬件置 1。事件通过 WUS 位域定义。此位由软件清零，方法是向 USART_ICR 寄存器中的 WUCF 写入 1。

如果 USART_CR3 寄存器中 WUFIE=1，则会生成中断。

注： *当 UESM 清零时，WUF 标志也清零。*

如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 19 **RWU**：*接收器从静默模式唤醒 (Receiver wakeup from Mute mode)*

该位指示 USART 是否处于静默模式。当识别出唤醒/静默序列时，此位由硬件清零/置 1。静默模式控制序列（地址或 IDLE）通过 USART_CR1 寄存器中的 WAKE 位进行选择。

当选择 IDLE 模式下唤醒时，该位只能通过用软件向 USART_RQR 寄存器中的 MMRQ 位写 1 的方式置 1。

0：接收器处于工作模式

1：接收器处于静默模式

注： *如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 996 页的第 33.4 节：USART 实现。*

位 18 **SBKF**：发送中断标志 (Send break flag)

此位指示已请求发送中断字符。通过将 1 写入 USART_CR3 寄存器中的 SBKRQ 位，此位由软件置 1。此位在中断发送的停止位期间由硬件自动复位。

0：不发送中断字符

1：将发送中断字符

位 17 **CMF**：字符匹配标志 (Character match flag)

接收到由 ADD[7:0] 定义的字符后由硬件将此位置 1。通过向 USART_ICR 寄存器中的 CMCF 写入 1，此位由软件清零。

如果 USART_CR1 寄存器中 CMIE=1，则会生成中断。

0：未检测到字符匹配

1：检测到字符匹配

位 16 **BUSY**：忙标志 (Busy flag)

此位由硬件置 1 和复位。当 RX 线路上正在进行通信（成功检测到起始位）时有效。在接收结束（成功或失败）时复位。

0：USART 处于空闲状态（无接收）

1：正在接收

位 15 **ABRF**：自动波特率标志 (Auto baud rate flag)

已设置自动波特率（RXFNE 也将置 1，并在 RXFNEIE = 1 时生成中断），或者自动波特率操作未成功完成时，此位由硬件置 1 (ABRE=1)（此时，ABRE、RXFNE 和 FE 也置 1）

为请求新的自动波特率检测，通过向 USART_RQR 寄存器中的 ABRRQ 写入 1，此位由软件清零。

注： 如果 USART 不支持自动波特率功能，该位保留且保持复位值。

位 14 **ABRE**：自动波特率错误 (Auto baud rate error)

如果波特率测量失败（波特率超出范围或字符比较失败），此位由硬件置 1。

通过将 1 写入 USART_CR3 寄存器中的 ABRRQ 位，此位由软件清零。

注： 如果 USART 不支持自动波特率功能，该位保留且保持复位值。

位 13 **UDR**：SPI 从器件下溢错误标志 (SPI slave underrun error flag)

在从发送模式下，如果在软件尚未将任何值加载到 USART_TDR 时出现第一个数据发送时钟脉冲，此标志将置 1。该标志通过将 USART_ICR 寄存器中的 UDRCF 位置 1 来复位。

0：无下溢错误

1：存在下溢错误

注： 如果 USART 不支持 SPI 从器件模式，该位保留且保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 12 **EOBF**：块结束标志 (End of block flag)

接收到完整块后，此位由硬件置 1（例如 T=1 智能卡模式）。接收到的字节数（从块的起始处，包括起始字段）等于或大于 BLEN + 4 时执行检测。

如果 USART_CR2 寄存器中 EOBIE = 1，则会生成中断。

通过向 USART_ICR 寄存器中的 EOBCF 写入 1，此位由软件清零。

0：未达到块结束

1：已达到块结束（字符数）

注： 如果不支持智能卡模式，该位保留且保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 11 **RTOF**：接收器超时 (Receiver timeout)

已经过在 RTOR 寄存器中编程的超时值后，如果无任何通信，此位由硬件置 1。通过向 USART_ICR 寄存器中的 RTOCF 写入 1，此位由软件清零。

如果 USART_CR2 寄存器中 RTOIE=1，则会生成中断。

在智能卡模式下，该超时对应于 CWT 或 BWT 时间。

0：未达到超值值

1：已达到超时值，未接收到任何数据

注： *如果 RTOR 寄存器中编程的时间值将 2 个字符隔开，则 RTOF 不置 1。如果此时间大于该值 + 2 个采样时间（2/16 或 2/8，具体取决于过采样方法），则 RTOF 标志置 1。*

即使 RE = 0，计数器仍会计数，但 RTOF 仅在 RE = 1 时置 1。如果 RE 置 1 时已经超时，则 RTOF 将置 1。

如果 USART 不支持接收器超时功能，该位保留且保持复位值。

位 10 **CTS**：CTS 标志 (CTS flag)

此位由硬件置 1/复位。此位是对 nCTS 输入引脚的状态取反。

0：nCTS 线置 1

1：nCTS 线复位

注： *如果不支持硬件流控制功能，该位保留且保持复位值。*

位 9 **CTSIF**：CTS 中断标志 (CTS interrupt flag)

如果 CTSE 位置 1，当 nCTS 输入切换时，此位由硬件置 1。通过将 1 写入 USART_ICR 寄存器中的 CTSCF 位，此位由软件清零。

如果 USART_CR3 寄存器中 CTSIE=1，则会生成中断。

0：nCTS 状态线上未发生变化

1：nCTS 状态线上发生变化

注： *如果不支持硬件流控制功能，该位保留且保持复位值。*

位 8 **LBDF**：LIN 中断检测标志 (LIN break detection flag)

检测到 LIN 断路时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 LBDCF 写入 1，此位由软件清零。

如果 USART_CR2 寄存器中 LBDIE = 1，则会生成中断。

0：未检测到 LIN 断路

1：未检测到 LIN 断路

注： *如果 USART 不支持 LIN 模式，该位保留且保持复位值。请参见第 996 页的第 33.4 节：USART 实现。*

位 7 **TXFNF**：TXFIFO 未满 (TXFIFO not full)

TXFNF 会在 TXFIFO 未满时由硬件置 1，表示可向 USART_TDR 中写入数据。每次对 USART_TDR 进行写操作都会将数据置于 TXFIFO 中。该标志保持置 1，直到 TXFIFO 已满。当 TXFIFO 已满时，该标志清零，表示不能向 USART_TDR 中写入数据。

如果 USART_CR1 寄存器中 TXFNFIE 位 = 1，则会生成中断。

0：发送 FIFO 已满

1：发送 FIFO 未满

注： *在清空请求期间，TXFNF 保持复位，直到 TXFIFO 为空。发送清空请求（通过将 TXFRQ 位置 1）后，应先检查 TXFNF 标志，然后再写入 TXFIFO（TXFNF 和 TXFE 将同时置 1）。*

单缓冲区发送期间使用该位。

位 6 **TC**：发送完成 (Transmission complete)

该位指示写入到 USART_TDR 中的最后一个数据已从移位寄存器中发出。

如果已完成对包含数据的帧的发送并且 TXFE 置 1，则该位由硬件置 1。

如果 USART_CR1 寄存器中 TCIE = 1，则会生成中断。

TC 位由软件清零，方法是向 USART_ICR 寄存器中的 TCCF 写入 1 或向 USART_TDR 寄存器执行写操作。

0：传送未完成

1：传送已完成

注： 如果 TE 位复位且无任何发送正在进行，TC 位会立即置 1。

位 5 **RXFNE**：RXFIFO 非空 (RXFIFO not empty)

RXFIFO 非空时，RXFNE 位由硬件置 1，这表示可以从 USART_RDR 寄存器读取数据。每次对 USART_RDR 进行读操作都会在 RXFIFO 中释放一个位置。

RXFNE 在 RXFIFO 为空时清零。也可以通过向 USART_RQR 寄存器中的 RXFRQ 位写入 1 将 RXFNE 标志位清零。

如果 USART_CR1 寄存器中 RXFNEIE=1，则会生成中断。

0：未接收到数据

1：已准备好读取接收到的数据

位 4 **IDLE**：检测到空闲线路 (IDLE line detected)

检测到空闲线路时，该位由硬件置 1。如果 USART_CR1 寄存器中 IDLEIE = 1，则会生成中断。通过向 USART_ICR 寄存器中的 IDLECF 写入 1，此位由软件清零。

0：未检测到空闲线路

1：检测到空闲线路

注： 直到 RXFNE 位已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。

使能静默模式 (MME=1) 后，如果 USART 未静默 (RWU=0)，则 IDLE 置 1，无论是否通过 WAKE 位选择了静默模式。如果 RWU=1，IDLE 不置 1。

位 3 **ORE**：溢出错误 (Overrun error)

在 RXFF = 1 的情况下，当移位寄存器中当前正在接收的数据准备好传输到 USART_RDR 寄存器时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 ORECF 写入 1，此位由软件清零。

如果 USART_CR1 寄存器中 RXFNEIE=1 或 EIE = 1，则会生成中断。

0：无溢出错误

1：检测到溢出错误

注： 当此位置 1 时，USART_RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。EIE 位置 1 后，如果在多缓冲区通信中 ORE 标志置 1，则会生成中断。

USART_CR3 寄存器中的 OVRDIS 位置 1 时，此位将被永久强制清零（无上溢检测）。

位 2 **NE**：噪声检测标志 (Noise detection flag)

当在接收的帧上检测到噪声时，该位由硬件置 1。此位由软件清零，方法是向 USART_ICR 寄存器中的 NECF 位写入 1。

0：未检测到噪声

1：检测到噪声

注： 该位不会生成中断，因为该位出现的时间与本身生成中断的 RXFNE 位出现的时间相同。EIE 位置 1 后，如果在多缓冲区通信中 NE 标志置 1，则会生成中断。

当线路无噪声时，可以通过将 ONEBIT 位编程为 1 提高 USART 对偏差的容差来禁止 NE 标志（请参见第 1013 页的第 33.5.8 节：USART 接收器对时钟偏差的容差）。

此错误与 USART_RDR 中的字符相关联。

位 1 **FE**：帧错误 (Framing error)

当检测到去同步化、过度的噪声或中断字符时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 FECF 写入 1，此位由软件清零。

在智能卡模式下发送数据时，如果在达到最大发送尝试次数后仍未成功（智能卡向数据帧发送 NACK 信号），则此位置 1。

如果 USART_CR1 寄存器中 EIE = 1，则会生成中断。

0：未检测到帧错误

1：检测到帧错误或中断字符

注： 此错误与 USART_RDR 中的字符相关联。

位 0 **PE**：奇偶校验错误 (Parity error)

当在接收器模式下发生奇偶校验错误时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 PECF 写入 1，此位由软件清零。

如果 USART_CR1 寄存器中的 PEIE = 1，则会生成中断。

0：无奇偶校验错误

1：奇偶校验错误

注： 此错误与 USART_RDR 中的字符相关联。

33.7.10 USART 中断和状态寄存器 [备用] (USART_ISR)

USART interrupt and status register

偏移地址：0x1C

复位值：0x0000 00C0

同一寄存器可用于使能 FIFO 模式（上一节）和禁止 FIFO 模式（本节）的情况。

禁止 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	TCBGT	Res.	Res.	RE ACK	TE ACK	WUF	RWU	SBKF	CMF	BUSY
						r			r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:26 保留，必须保持复位值。

位 25 **TCBGT**：保护时间前发送完成标志 (Transmission complete before guard time flag)

当写入到 USART_TDR 中的最后一个数据已正确从移位寄存器中发出时，该位置 1。

如果包含数据的帧完成发送，并且智能卡未发回任何 NACK，则该位在智能卡模式下由硬件置 1。如果 USART_CR3 寄存器中的 TCBGTIE=1，则生成中断。

此位由软件清零，方法是向 USART_ICR 寄存器中的 TCBGTCF 写入 1 或向 USART_TDR 寄存器执行写操作。

0：发送未完成或发送未成功完成（即，从智能卡接收到 NACK）

1：发送成功完成（在保护时间结束之前完成，智能卡未发送 NACK）。

注： 如果 USART 不支持智能卡模式，该位保留且保持复位值。如果 USART 支持智能卡模式并使能智能卡模式，则 TCBGT 复位值为“1”。请参见第 996 页的第 33.4 节：USART 实现。

位 24:23 保留，必须保持复位值。

位 22 **REACK**：接收使能确认标志 (Receive enable acknowledge flag)

USART 采用接收使能值时，通过硬件将此位置 1/复位。

此位可用于验证 USART 是否准备好在进入低功耗模式前接收数据。

注：　如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 21 **TEACK**：发送使能确认标志 (Transmit enable acknowledge flag)

USART 采用发送使能值时，通过硬件将此位置 1/复位。

通过写入 TE=0 生成空闲帧请求，然后在 USART_CR1 寄存器中写入 TE=1 以遵循 TE=0 最短周期时，可使用此位。

位 20 **WUF**：从低功耗模式唤醒标志 (Wakeup from low-power mode flag)

当检测到唤醒事件时，此位由硬件置 1。事件通过 WUS 位域定义。此位由软件清零，方法是向 USART_ICR 寄存器中的 WUCF 写入 1。

如果 USART_CR3 寄存器中 WUFIE=1，则会生成中断。

注：　当 UESM 清零时，WUF 标志也清零。

如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 19 **RWU**：*接收器从静默模式唤醒 (Receiver wakeup from Mute mode)*

该位指示 USART 是否处于静默模式。当识别出唤醒/静默序列时，此位由硬件清零/置 1。静默模式控制序列（地址或 IDLE）通过 USART_CR1 寄存器中的 WAKE 位进行选择。

当选择 IDLE 模式下唤醒时，该位只能通过用软件向 USART_RQR 寄存器中的 MMRQ 位写 1 的方式置 1。

0：接收器处于工作模式

1：接收器处于静默模式

注：　如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 18 **SBKF**：发送中断标志 (Send break flag)

此位指示已请求发送中断字符。通过将 1 写入 USART_CR3 寄存器中的 SBKRQ 位，此位由软件置 1。此位在中断发送的停止位期间由硬件自动复位。

0：不发送中断字符

1：将发送中断字符

位 17 **CMF**：字符匹配标志 (Character match flag)

接收到由 ADD[7:0] 定义的字符后由硬件将此位置 1。通过向 USART_ICR 寄存器中的 CMCF 写入 1，此位由软件清零。

如果 USART_CR1 寄存器中 CMIE=1，则会生成中断。

0：未检测到字符匹配

1：检测到字符匹配

位 16 **BUSY**：忙标志 (Busy flag)

此位由硬件置 1 和复位。当 RX 线路上正在进行通信（成功检测到起始位）时有效。在接收结束（成功或失败）时复位。

0：USART 处于空闲状态（无接收）

1：正在接收

位 15 **ABRF**：自动波特率标志 (Auto baud rate flag)

已设置自动波特率（RXNE 也将置 1，并在 RXNEIE = 1 时生成中断），或者自动波特率操作未成功完成时，此位由硬件置 1 (ABRE=1)（此时，ABRE、RXNE 和 FE 也置 1）

为请求新的自动波特率检测，通过向 USART_RQR 寄存器中的 ABRRQ 写入 1，此位由软件清零。

注：　如果 USART 不支持自动波特率功能，该位保留且保持复位值。

位 14 **ABRE**：自动波特率错误 (Auto baud rate error)

如果波特率测量失败（波特率超出范围或字符比较失败），此位由硬件置 1。

通过将 1 写入 USART_CR3 寄存器中的 ABRRQ 位，此位由软件清零。

注：　如果 USART 不支持自动波特率功能，该位保留且保持复位值。

位 13 **UDR**：SPI 从器件下溢错误标志 (SPI slave underrun error flag)

在从发送模式下，如果在软件尚未将任何值加载到 USART_TDR 时出现第一个数据发送时钟脉冲，此标志将置 1。该标志通过将 USART_ICR 寄存器中的 UDRCF 位置 1 来复位。

0：无下溢错误

1：存在下溢错误

注：　如果 USART 不支持 SPI 从器件模式，该位保留且保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 12 **EOBF**：块结束标志 (End of block flag)

接收到完整块后，此位由硬件置 1（例如 T=1 智能卡模式）。接收到的字节数（从块的起始处，包括起始字段）等于或大于 BLEN + 4 时执行检测。

如果 USART_CR2 寄存器中 EOBIE = 1，则会生成中断。

通过向 USART_ICR 寄存器中的 EOBCF 写入 1，此位由软件清零。

0：未达到块结束

1：已达到块结束（字符数）

注：　如果不支持智能卡模式，该位保留且保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 11 **RTOF**：接收器超时 (Receiver timeout)

已经过在 RTOR 寄存器中编程的超时值后，如果无任何通信，此位由硬件置 1。通过向 USART_ICR 寄存器中的 RTOCF 写入 1，此位由软件清零。

如果 USART_CR2 寄存器中 RTOIE=1，则会生成中断。

在智能卡模式下，该超时对应于 CWT 或 BWT 时间。

0：未达到超值值

1：已达到超时值，未接收到任何数据

注：　如果 RTOR 寄存器中编程的时间值将 2 个字符隔开，则 RTOF 不置 1。如果此时间大于该值 + 2 个采样时间（2/16 或 2/8，具体取决于过采样方法），则 RTOF 标志置 1。

即使 RE = 0，计数器仍会计数，但 RTOF 仅在 RE = 1 时置 1。如果 RE 置 1 时已经超时，则 RTOF 将置 1。

如果 USART 不支持接收器超时功能，该位保留且保持复位值。

位 10 **CTS**：CTS 标志 (CTS flag)

此位由硬件置 1/复位。此位是对 nCTS 输入引脚的状态取反。

0：nCTS 线置 1

1：nCTS 线复位

注：　如果不支持硬件流控制功能，该位保留且保持复位值。

位 9 **CTSIF**：CTS 中断标志 (CTS interrupt flag)

如果 CTSE 位置 1，当 nCTS 输入切换时，此位由硬件置 1。通过将 1 写入 USART_ICR 寄存器中的 CTSCF 位，此位由软件清零。

如果 USART_CR3 寄存器中 CTSIE=1，则会生成中断。

0：nCTS 状态线上未发生变化

1：nCTS 状态线上发生变化

注：　如果不支持硬件流控制功能，该位保留且保持复位值。

位 8 **LBDF**：LIN 中断检测标志 (LIN break detection flag)

检测到 LIN 断路时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 LBDCF 写入 1，此位由软件清零。

如果 USART_CR2 寄存器中 LBDIE = 1，则会生成中断。

0：未检测到 LIN 断路

1：未检测到 LIN 断路

注： 如果 USART 不支持 LIN 模式，该位保留且保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 7 **TXE**：发送数据寄存器为空 (Transmit data register empty)

当 USART_TDR 寄存器的内容已传输到移位寄存器时，TXE 由硬件置 1。通过对 USART_TDR 寄存器执行写操作将该位清零。为丢弃数据（仅在智能卡 T=0 模式下出现发送故障时），也可以通过向 USART_RQR 寄存器中的 TXFRQ 写入 1 来将 TXE 标志置 1。

如果 USART_CR1 寄存器中 TXEIE 位 = 1，则会生成中断。

0：数据寄存器已满

1：数据寄存器未满

位 6 **TC**：发送完成 (Transmission complete)

该位指示写入到 USART_TDR 中的最后一个数据已从移位寄存器中发出。

如果已完成对包含数据的帧的发送并且 TXE 置 1，则此位由硬件置 1。

如果 USART_CR1 寄存器中 TCIE = 1，则会生成中断。

TC 位由软件清零，方法是向 USART_ICR 寄存器中的 TCCF 写入 1 或向 USART_TDR 寄存器执行写操作。

0：传送未完成

1：传送已完成

注： 如果 TE 位复位且无任何发送正在进行，TC 位会立即置 1。

位 5 **RXNE**：读取数据寄存器不为空 (Read data register not empty)

当 USART_RDR 移位寄存器的内容已传输到 USART_RDR 寄存器时，RXNE 位由硬件置 1。通过对 USART_RDR 寄存器执行读取操作将该位清零。也可以通过将 USART_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE 标志位清零。

如果 USART_CR1 寄存器中 RXNEIE = 1，则会生成中断。

0：未接收到数据

1：已准备好读取接收到的数据

位 4 **IDLE**：检测到空闲线路 (IDLE line detected)

检测到空闲线路时，该位由硬件置 1。如果 USART_CR1 寄存器中 IDLEIE = 1，则会生成中断。通过向 USART_ICR 寄存器中的 IDLECF 写入 1，此位由软件清零。

0：未检测到空闲线路

1：检测到空闲线路

注： 直到 RXNE 位已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。

使能静默模式 (MME=1) 后，如果 USART 未静默 (RWU=0)，则 IDLE 置 1，无论是否通过 WAKE 位选择了静默模式。如果 RWU=1，IDLE 不置 1。

位 3 **ORE**：溢出错误 (Overrun error)

在 RXNE = 1 的情况下，当移位寄存器中当前正在接收的数据准备好传输到 USART_RDR 寄存器时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 ORECF 写入 1，此位由软件清零。

如果 USART_CR1 寄存器中 RXNEIE=1 或 EIE = 1，则会生成中断。

0：无溢出错误

1：检测到溢出错误

注： 当此位置 1 时，USART_RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。EIE 位置 1 后，如果在多缓冲区通信中 ORE 标志置 1，则会生成中断。

USART_CR3 寄存器中的 OVRDIS 位置 1 时，此位将被永久强制清零（无上溢检测）。

位 2 **NE**：噪声检测标志 (Noise detection flag)

当在接收的帧上检测到噪声时，该位由硬件置 1。此位由软件清零，方法是向 USART_ICR 寄存器中的 NECF 位写入 1。

0：未检测到噪声

1：检测到噪声

注： *该位不会生成中断，因为该位出现的时间与本身生成中断的 RXNE 位出现的时间相同。EIE 位置 1 后，如果在多缓冲区通信中 NE 标志置 1，则会生成中断。*

当线路无噪声时，可以通过将 ONEBIT 位编程为 1 提高 USART 对偏差的容差来禁止 NE 标志（请参见第 1013 页的第 33.5.8 节：USART 接收器对时钟偏差的容差）。

位 1 **FE**：帧错误 (Framing error)

当检测到去同步化、过度的噪声或中断字符时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 FECF 写入 1，此位由软件清零。

在智能卡模式下发送数据时，如果在达到最大发送尝试次数后仍未成功（智能卡向数据帧发送 NACK 信号），则此位置 1。

如果 USART_CR1 寄存器中 EIE = 1，则会生成中断。

0：未检测到帧错误

1：检测到帧错误或中断字符

位 0 **PE**：奇偶校验错误 (Parity error)

当在接收器模式下发生奇偶校验错误时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 PECF 写入 1，此位由软件清零。

如果 USART_CR1 寄存器中的 PEIE = 1，则会生成中断。

0：无奇偶校验错误

1：奇偶校验错误

33.7.11 USART 中断标志清零寄存器 (USART_ICR)

USART interrupt flag clear register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.
											w			w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	UDRCF	EOBCF	RTOCF	Res.	CTSCF	LBDCF	TCBGTCF	TCCF	TXFECF	IDLECF	ORECF	NECF	FECF	PECF
		w	w	w		w	w	w	w	w	w	w	W	w	w

位 31:21 保留，必须保持复位值。

位 20 **WUCF**：从低功耗模式唤醒清零标志 (Wakeup from low-power mode clear flag)

将 1 写入此位时，USART_ISR 寄存器中 WUF 标志将清零。

注： *如果 USART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。*

位 19:18 保留，必须保持复位值。

位 17 **CMCF**：字符匹配清零标志 (Character match clear flag)

将 1 写入此位时，USART_ISR 寄存器中 CMF 标志将清零。

位 16:14 保留，必须保持复位值。

位 13 **UDRCF**：SPI 从器件下溢清零标志 (SPI slave underrun clear flag)
将 1 写入此位时，USART_ISR 寄存器中 UDRF 标志将清零。
注： 如果 USART 不支持 SPI 从器件模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 12 **EOBCF**：块结束清零标志 (End of block clear flag)
将 1 写入此位时，USART_ISR 寄存器中 EOBF 标志将清零。
注： 如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 11 **RTOCF**：接收器超时清零标志 (Receiver timeout clear flag)
将 1 写入此位时，USART_ISR 寄存器中 RTOF 标志将清零。
注： 如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 10 保留，必须保持复位值。

位 9 **CTSCF**：CTS 清零标志 (CTS clear flag)
将 1 写入此位时，USART_ISR 寄存器中 CTSIF 标志将清零。
注： 如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 8 **LBDCF**：LIN 断路检测清零标志 (LIN break detection clear flag)
将 1 写入此位时，USART_ISR 寄存器中 LBDF 标志将清零。
注： 如果不支持 LIN 模式，该位保留且必须保持复位值。请参见第 996 页的第 33.4 节：USART 实现。

位 7 **TCBGTCF**：保护时间前发送完成清零标志 (Transmission complete before Guard time clear flag)
将 1 写入此位时，USART_ISR 寄存器中 TCBGT 标志将清零。

位 6 **TCCF**：发送完成清零标志 (Transmission complete clear flag)
将 1 写入此位时，USART_ISR 寄存器中 TC 标志将清零。

位 5 **TXFECF**：TXFIFO 为空清零标志 (TXFIFO empty clear flag)
将 1 写入此位时，USART_ISR 寄存器中 TXFE 标志将清零。

位 4 **IDLECF**：检测到空闲线路清零标志 (Idle line detected clear flag)
将 1 写入此位时，USART_ISR 寄存器中 IDLE 标志将清零。

位 3 **ORECF**：上溢错误清零标志 (Overrun error clear flag)
将 1 写入此位时，USART_ISR 寄存器中 ORE 标志将清零。

位 2 **NECF**：检测到噪声清零标志 (Noise detected clear flag)
将 1 写入此位时，USART_ISR 寄存器中 NE 标志将清零。

位 1 **FECF**：帧错误清零标志 (Framing error clear flag)
将 1 写入此位时，USART_ISR 寄存器中 FE 标志将清零。

位 0 **PECF**：奇偶校验错误清零标志 (Parity error clear flag)
将 1 写入此位时，USART_ISR 寄存器中 PE 标志将清零。

33.7.12 USART 接收数据寄存器 (USART_RDR)

USART receive data register

偏移地址：0x24

复位值：0xXXXX XXXX

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
							r	r	r	r	r	r	r	r	r

位 31:9 保留，必须保持复位值。

位 8:0 **RDR[8:0]**：接收数据值 (Receive data value)

包含接收到的数据字符。

RDR 寄存器在输入移位寄存器和内部总线之间提供了并行接口（请参见*图 322*）。

在使能奇偶校验位的情况下进行接收时，从 MSB 位中读取的值为接收到的奇偶校验位。

33.7.13 USART 发送数据寄存器 (USART_TDR)

USART transmit data register

偏移地址：0x28

复位值：0xXXXX XXXX

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

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

位 31:9 保留，必须保持复位值。

位 8:0 **TDR[8:0]**：发送数据值 (Transmit data value)

包含要发送的数据字符。

USART_TDR 寄存器在内部总线和输出移位寄存器之间提供了并行接口（请参见*图 322*）。

在使能奇偶校验位的情况下（USART_CR1 寄存器中的 PCE 位被置 1）进行发送时，由于 MSB 的写入值（位 7 或位 8，具体取决于数据长度）会被奇偶校验位所取代，因此该值不起任何作用。

注：　只能在 TXE/TXFNF=1 时写入此寄存器。

33.7.14 USART 预分频器寄存器 (USART_PRESC)

USART prescaler register

只有在禁止 USART (UE=0) 时才能写入此寄存器。

偏移地址：0x2C

复位值：0x0000 0000

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

位 31:4 保留，必须保持复位值。

位 3:0 **PRESCALER[3:0]**：时钟预分频器 (Clock prescaler)

USART 输入时钟可通过预分频系数进行分频：

0000：输入时钟未分频
0001：输入时钟 2 分频
0010：输入时钟 4 分频
0011：输入时钟 6 分频
0100：输入时钟 8 分频
0101：输入时钟 10 分频
0110：输入时钟 12 分频
0111：输入时钟 16 分频
1000：输入时钟 32 分频
1001：输入时钟 64 分频
1010：输入时钟 128 分频
1011：输入时钟 256 分频
其余组合：保留

注： 为 PRESCALER 编程不允许的值时，编程的预分频值将为“1011”，即输入时钟除以 256。

33.7.15 USART 寄存器映射

下表提供了 USART 寄存器映射和复位值。

表 198. USART 寄存器映射和复位值

偏移	寄存器名称	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
0x00	USART_CR1 FIFO enabled	RXFFIE	TXFEIE	FIFOEN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]					OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXFNFIE	TCIE	RXFNEIE	IDLEIE	TE	RE	UESM	UE
	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
0x00	USART_CR1 FIFO disabled	Res.	Res.	FIFOEN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]					OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
	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
0x04	USART_CR2	ADD[7:0]								RTOEN	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	DIS_NSS	Res.	Res.	SLVEN
	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
0x08	USART_CR3	TXFTCFG[2:0]			RXFTIE	RXFTCFG[2:0]			TCBGTIE	TXFTIE	WUFIE	WUS[1:0]	SCARCNT2:0]				Res.	DEP	DEM	DDRE	OVRDIS	ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
	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
0x0C	USART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	USART_GTPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GT[7:0]								PSC[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	USART_RTOR	BLEN[7:0]								RTO[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	0	0	0	0	0	0	0	0
0x18	USART_RQR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
	Reset value																												0	0	0	0	0

表 198. USART 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x1C	USART_ ISR FIFO mode enabled	Res.	Res.	Res.	Res.	TXFT	RXFT	TCBGT	RXFF	TXFE	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXFNF	TC	RXFNE	IDLE	ORE	NE	FE	PE
	Reset value					0	0	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0x1C	USART_ ISR FIFO mode disabled	Res.	Res.	Res.	Res.	Res.	Res.	TCBGT	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
	Reset value							0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0x20	USART_ ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	Res.	Res.	UDRCF	EOBCF	RTOCF	Res.	CTSCF	LBDCF	TCBGTCF	TCCF	TXFECF	IDLECF	ORECF	NECF	FECF	PECF
	Reset value												0			0				0	0	0		0	0	0	0	0	0	0	0	0	0
0x24	USART_ RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x28	USART_ TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x2C	USART_ PRESC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]			
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 2.2 节：存储器构成*。

34 低功耗通用异步接收器 (LPUART)

本节介绍低功耗通用异步收发器 (LPUART)。

34.1 LPUART 简介

LPUART 是一种 UART，允许在有限功耗下双向 UART 通信。仅需 32.768 kHz LSE 时钟即可进行高达 9600 波特/s 的 UART 通信。当 LPUART 用其他不同于 LSE 的时钟源时，可以达到更高的波特率。

即使当微控制器处于低功耗模式，能耗极低时，LPUART 也会等待 UART 帧的到来。LPUART 包含所有必要的硬件支持，使在最小功耗下可以进行异步串行通信。

它支持半双工单线通信和调制解调器操作 (CTS/RTS)，还支持多处理器通信。

DMA（直接存储器访问）可用于数据发送/接收。

34.2 LPUART 主要特性

- 全双工异步通信
- NRZ 标准格式（标记/空格）
- 可编程波特率
- 使用 32.768 kHz 时钟源时波特率为 300 波特/s 到 9600 波特/s
- 使用高频时钟源可实现更高的波特率
- 两个用于收发数据的内部 FIFO
  每个 FIFO 均可由软件使能/禁止，并且均带有用于指示 FIFO 状态的状态标志
- 双时钟域，带有独立于 PCLK 的外设专用内核时钟
- 数据字长度可编程（7 位、8 位或 9 位）
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 停止位可配置（支持 1 个或 2 个停止位）
- 单线半双工通信
- 使用 DMA 实现连续通信
- 使用中央 DMA 在预留的 SRAM 缓冲区中收/发字节
- 发射器和接收器有单独的使能位
- 发送和接收的单独信号极性控制
- Tx/Rx 引脚配置可交换
- 调制解调器和 RS-485 收发器的硬件流控制
- 传输检测标志：
  - 接收缓冲区已满
  - 发送缓冲区为空
  - BUSY 标志和发送结束标志

- 奇偶校验控制：
  - 发送奇偶校验位
  - 检查接收的数据字节的奇偶性
- 四个错误检测标志：
  - 上溢错误
  - 噪声检测
  - 帧错误
  - 奇偶校验错误
- 具有标志的中断源
- 多处理器通信：从静默模式唤醒（通过空闲线检测或地址标记检测）

34.3 LPUART 功能说明

34.3.1 LPUART 框图

图 349. LPUART框图

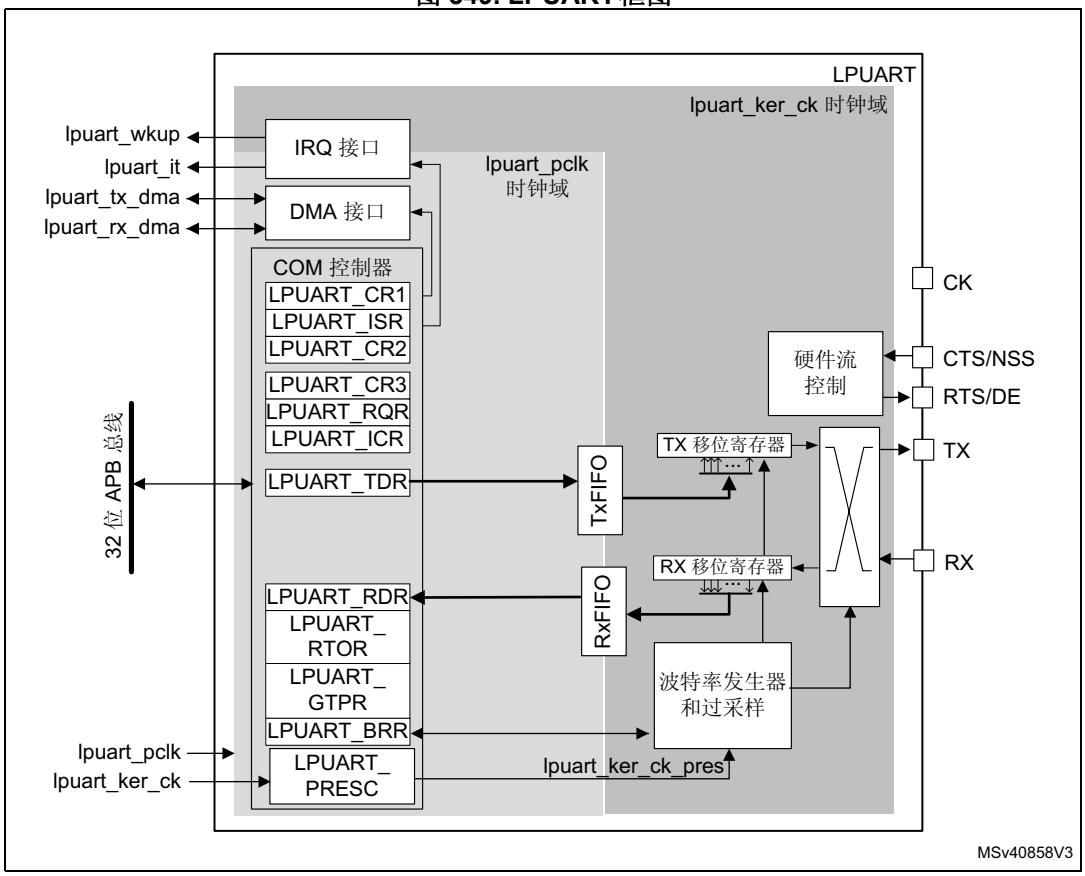


图 349 中的简化框图显示的是两个完全独立的时钟域：

- **lpuart_pclk** 时钟域

  lpuart_pclk 时钟信号馈送外设总线接口。需要访问 LPUART 寄存器时，该信号必须有效。

- **lpuart_ker_ck** 内核时钟域

  lpuart_ker_ck 是 LPUART 时钟源。它独立于 **lpuart_pclk**，由 RCC 提供。因此，即使 **lpuart_ker_ck** 停止，也可以对 LPUART 寄存器进行读/写操作。

  禁用双时钟域功能时，**lpuart_ker_ck** 与 lpuart_pclk 时钟相同。

lpuart_pclk 和 **lpuart_ker_ck** 之间无任何限制：**lpuart_ker_ck** 既可快于也可慢于 **lpuart_pclk**，唯一的限制是软件以足够快的速度管理通信的能力。

34.3.2 LPUART 信号

LPUART 双向通信需要至少两个引脚：接收数据输入引脚 (RX) 和发送数据输出引脚 (TX)：

- **RX**（接收数据输入引脚）

  RX 为串行数据输入引脚。

- **TX**（发送数据输出引脚）

  如果关闭发送器，该输出引脚模式由其 I/O 端口配置决定。如果使能了发送器但没有待发送的数据，则 TX 引脚处于高电平。在单线模式下，该 I/O 用于发送和接收数据。

RS232 硬件流控制模式

在 RS232 硬件流控制模式下需要以下引脚：

- **CTS**（清除以发送）

  如果驱动为高电平，则该信号用于在当前传输结束时阻止数据发送。

- **RTS**（请求以发送）

  如果为低电平，则该信号用于指示 USART 已准备好接收数据。

RS485 硬件流控制模式

在 RS485 硬件控制模式下需要以下引脚：

- **DE** 驱动器使能

  该信号用于激活外部收发器的发送模式。

注： *DE 和 RTS 共用同一个引脚。*

34.3.3 LPUART 字符说明

可通过对 LPUART_CR1 寄存器中的 M 位（M0：位 12，M1：位 28）进行编程来将字长设置为 7 位、8 位或 9 位（请参见*图 323*）。

- 7 位字符长度：M[1:0] = “10”
- 8 位字符长度：M[1:0] = “00”
- 9 位字符长度：M[1:0] = “01”

在默认情况下，信号（TX 或 RX）在起始位工作期间处于低电平状态。在停止位工作期间处于高电平状态。

通过极性配置控制，可以单独针对每个信号对这些值取反。

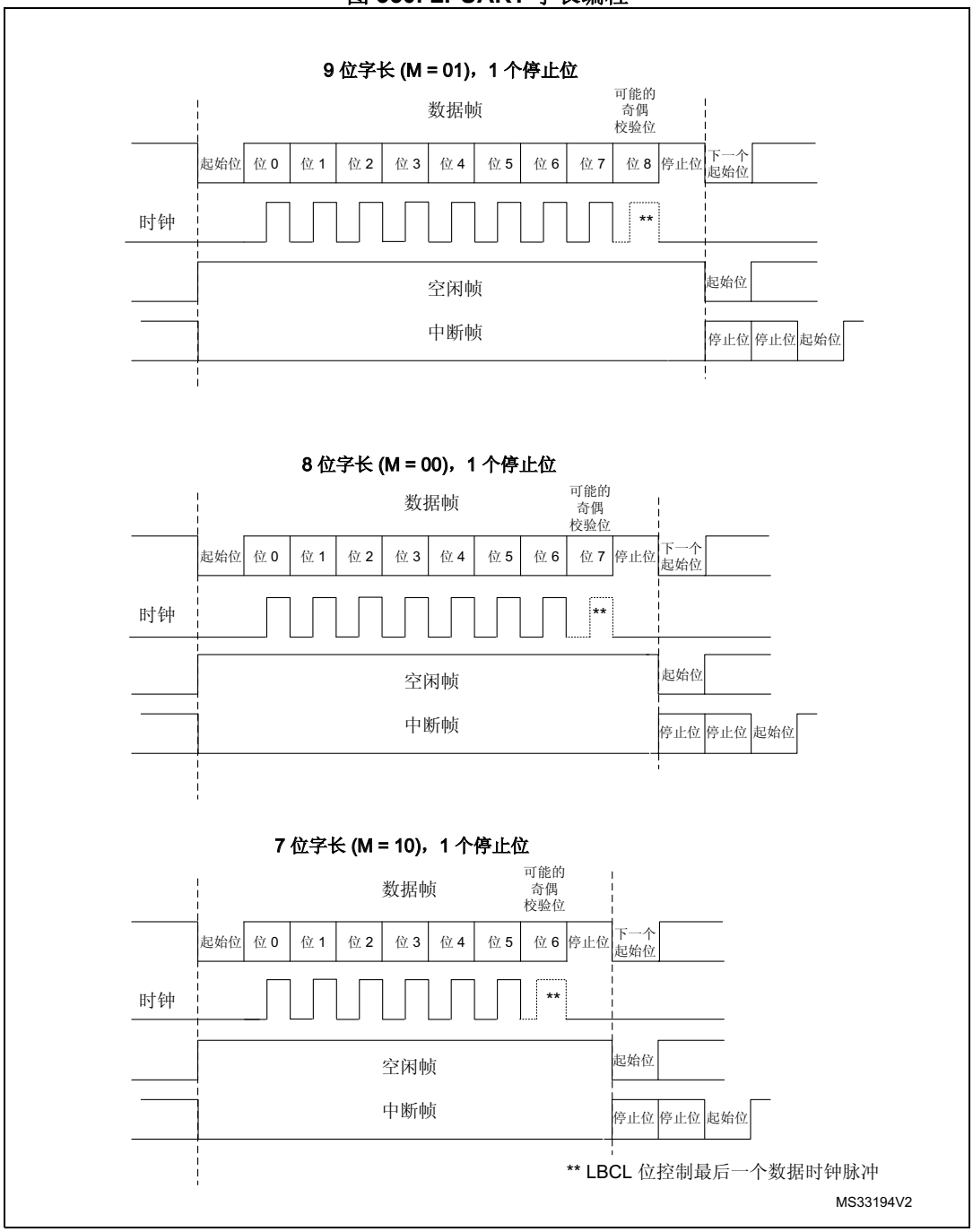
*空闲字符*可理解为整个帧周期内电平均为“1”。（停止位的电平也是“1”）。

*停止字符*可理解为在一个帧周期内接收到的电平均为“0”。发送器在中断帧的末尾插入 2 个停止位。

发送和接收操作由通用波特率发生器驱动。发送器和接收器的使能位置 1 时，将分别生成发送时钟和接收时钟。

下面给出了各个块的详细信息。

图 350. LPUART 字长编程



34.3.4 LPUART FIFO 和阈值

LPUART 可工作在 FIFO 模式下。

LPUART 具有一个发送 FIFO (TXFIFO) 和一个接收 FIFO (RXFIFO)。可通过将 LPUART_CR1 寄存器中的 FIFOEN 位（位 29）置 1 使能 FIFO 模式。

最大数据字长度为 9 位，因此 TXFIFO 为 9 位宽。不过，RXFIFO 的默认宽度为 12 位。这是因为接收器不仅在 FIFO 中存储数据，而且还存储与每个字符相关的错误标志（奇偶校验错误、噪声错误和帧错误标志）。

注： *接收的数据与相应的标志一起存储在 RXFIFO 中，但读取 RDR 时仅读取数据。*

状态标志位于 LPUART_ISR 寄存器中。

可以定义触发 Tx 和 RX 中断的 TXFIFO 和 RXFIFO 阈值。这些阈值通过 LPUART_CR3 控制寄存器中的 RXFTCFG 和 TXFTCFG 位域进行编程。

在这种情况下：

- 当 RXFIFO 中接收的数据量达到 RXFTCFG 位域中编程的阈值时，LPUART_ISR 寄存器中的 RXFT 标志置 1 并生成相应中断（如果使能）。

  这意味着 RXFIFO 被填充，直到 RXFIFO 中的数据量等于编程的阈值。

  已接收到 RXFTCFG 数据：LPUART_RDR 中有 1 个数据，RXFIFO 中有 (RXFTCFG - 1) 个数据。例如，如果将 RXFTCFG 编程为“101”，则在接收到对应于 FIFO 大小的数据量（RXFIFO 中有 FIFO 大小 - 1 个数据，LPUART_RDR 中有 1 个数据）时，RXFT 标志将置 1。因此，下一个接收到的数据不会将上溢标志置 1。
- 当 TXFIFO 中空单元数量达到 TXFTCFG 位域中编程的阈值时，LPUART_ISR 寄存器中的 TXFT 标志置 1 并生成相应中断（如果使能）。

  这意味着 TXFIFO 为空，直到 TXFIFO 中的空存储单元量等于编程的阈值。

34.3.5 LPUART 发送器

发送器可发送 7 位、8 位或 9 位的数据字，具体取决于 M 位的状态。要激活发送器功能，必须将发送使能位 (TE) 置 1。发送移位寄存器中的数据输出到 TX 引脚。

字符发送

LPUART 发送期间，首先通过 TX 引脚移出数据的最低有效位（默认配置）。该模式下，LPUART_TDR 寄存器的缓冲区 (TDR) 位于内部总线和发送移位寄存器之间（请参见*图 349*）。

使能 FIFO 模式时，写入到 LPUART_TDR 寄存器中的数据会在 TXFIFO 中排队。

每个字符前面都有一个起始位，其对应于一个位周期的逻辑低电平。字符由可配置数量的停止位终止。

停止位的数量可为 1 或 2。

注： *向 LPUART_TDR 中写入要发送的数据前，TE 位必须先置 1。*

数据发送期间不应复位 TE 位。发送期间复位 TE 位会冻结波特率计数器，进而损坏 TX 引脚上的数据。当前发送的数据将会丢失。

使能 TE 位后，将会发送空闲帧。

可配置的停止位

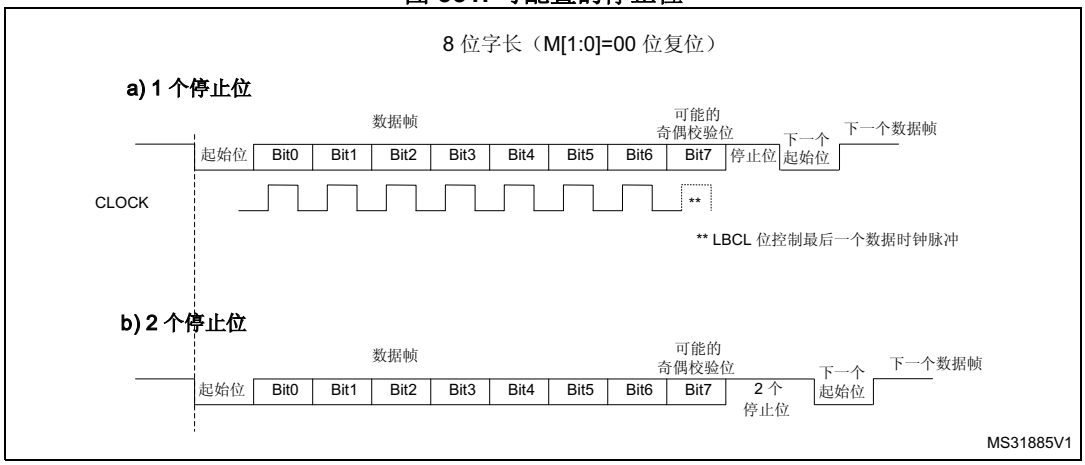
可以在 LPUART_CR2 的位 13 和位 12 中编程将随各个字符发送的停止位的数量。

- ***1 个停止位***：这是停止位数量的默认值。
- ***2 个停止位***：正常 LPUART 模式、单线模式和调制解调器模式支持该值。

空闲帧发送将包括停止位。

中断发送是 10 个低电平位（M[1:0] = “00”时）、11 个低电平位（M[1:0] = “01”时）或 9 个低电平位（M[1:0] = “10”时），然后是 2 个停止位。无法传送长中断（中断长度大于 9/10/11 个低电平位）。

图 351. 可配置的停止位



字符发送步骤

要发送字符，需遵循以下步骤：

1. 对 LPUART_CR1 中的 M 位进行编程以定义字长。
2. 使用 LPUART_BRR 寄存器选择所需波特率。
3. 对 LPUART_CR2 中的停止位数量进行编程。
4. 通过向 LPUART_CR1 寄存器中的 UE 位写入“1”使能 LPUART。
5. 如果将进行多缓冲区通信，请选择 LPUART_CR3 中的 DMA 使能 (DMAT)。按照*第 33.5.10 节：USART 多处理器通信*中的说明配置 DMA 寄存器。
6. 将 LPUART_CR1 中的 TE 位置 1 以便在首次发送时发送一个空闲帧。
7. 在 LPUART_TDR 寄存器中写入要发送的数据。为每个要在单缓冲区模式下发送的数据重复该操作。
   - 禁止 FIFO 模式时，向 LPUART_TDR 写入数据会将 TXE 标志清零。
   - 使能 FIFO 模式时，向 LPUART_TDR 写入数据会为 TXFIFO 增添一个数据。当 TXFNF 标志置 1 时，会对 LPUART_TDR 执行写操作。该标志会保持置 1，直到 TXFIFO 已满。
8. 将最后一个数据写入 LPUART_TDR 寄存器后，等待 TC =1。这表明最后一个帧的传送已完成。
   - 禁止 FIFO 模式时，这表示最后一个帧的发送已完成。
   - 使能 FIFO 模式时，这表示 TXFIFO 和移位寄存器均为空。

   当 LPUART 被禁止或进入暂停模式时，需要执行此检查来避免损坏最后一次发送。

单字节通信

- 禁止 FIFO 模式时：

  对发送数据寄存器进行写操作始终会清零 TXE 位。TXE 标志由硬件置 1 以指示：

  – 数据已从 LPUART_TDR 寄存器移到移位寄存器中且数据发送已开始；
  – LPUART_TDR 寄存器为空；
  – LPUART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。

  当 TXEIE 位置 1 时，TXE 标志会生成中断。

  发送时，要传入 LPUART_TDR 寄存器的写指令中存有 TDR 寄存器中的数据，该数据将在当前发送结束时复制到移位寄存器中。

  未发送时，要传入 LPUART_TDR 寄存器的写指令会将数据置于移位寄存器中，数据发送开始时，TXE 位置 1。

- 使能 FIFO 模式时，TXFNF（TXFIFO 未满）标志由硬件置 1，以指示：
  – TXFIFO 未满；
  – LPUART_TDR 寄存器为空；
  – LPUART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。发送时，对 LPUART_TDR 寄存器的写操作会将数据存储在 TXFIFO 中。该数据将在当前发送结束时从 TXFIFO 复制到移位寄存器中。

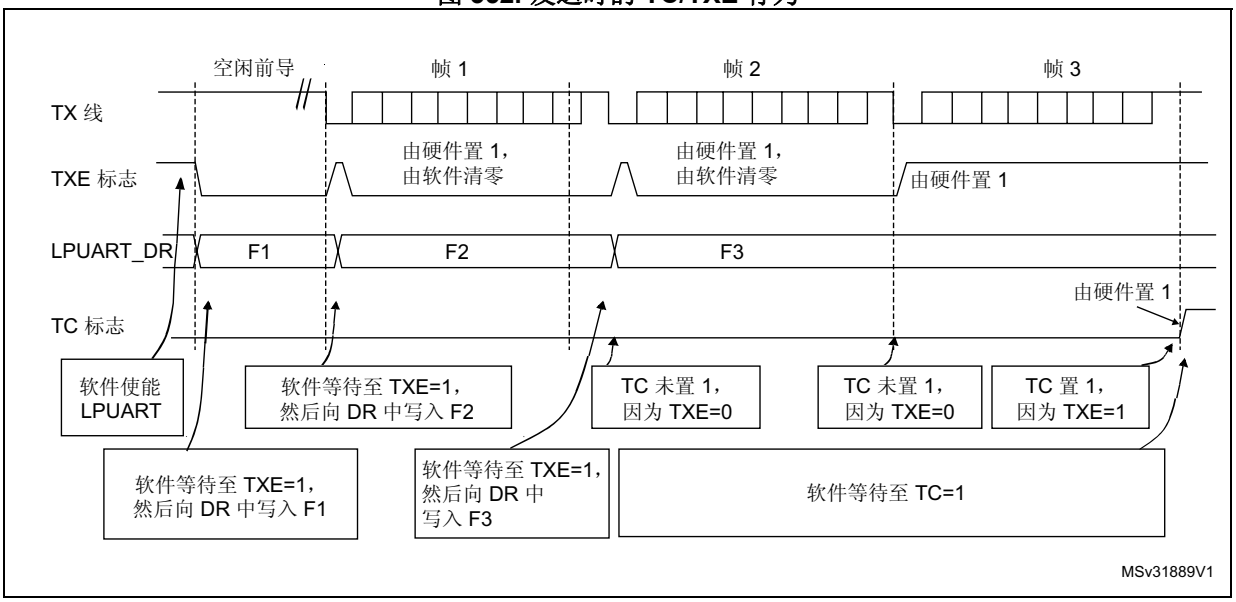
  TXFIFO 未满时，即使在对 LPUART_TDR 执行完写操作后，TXFNF 标志也保持为“1”。该标志在 TXFIFO 已满时清零。TXFNEIE 位置 1 时，该标志会生成中断。

  或者，当达到 TXFIFO 阈值时，会生成中断并将数据写入 TXFIFO。在这种情况下，CPU 可写入由编程的阈值定义的数据块。

  如果帧已发送（停止位后）且 TXE 标志（FIFO 模式下的 TXFE）置 1，TC 位将变为高电平。如果 LPUART_CR1 寄存器中的 TCIE 位置 1，将生成中断。

  向 LPUART_TDR 寄存器中写入最后一个数据后，必须等待至 TC=1，之后才可禁止 LPUART 或使微控制器进入低功率模式（请参见*图 352：发送时的 TC/TXE 行为*）。

图 352. 发送时的 TC/TXE 行为



注：　使能 FIFO 管理时，TXFNF 标志将用于数据发送。

中断字符

将 SBKRQ 位置 1 将发送一个中断字符。中断帧的长度取决于 M 位（请参见*图 350*）。

如果将“1”写入 SBKRQ 位，则当前字符发送完成后，将在 TX 线路上发送一个中断字符。通过写操作将 SBKF 位置 1 并在中断字符发送完成时（发送中断字符后的停止位期间），该位由硬件复位。LPUART 在中断帧末尾的两位持续时间内插入一个逻辑“1”信号 (STOP)，以确保识别下个帧的起始位。

如果 SBKRQ 位置 1，则在当前发送结束时，会发送一个中断字符。

使能 FIFO 模式时，即使 TXFIFO 已满，发送中断字符的优先级也仍高于发送数据的优先级。

空闲字符

将 TE 位置 1 会驱动 LPUART 在第一个数据帧之前发送一个空闲帧。

34.3.6 LPUART 接收器

LPUART 可接收 7 位、8 位或 9 位的数据字，具体取决于 LPUART_CR1 寄存器中的 M 位。

起始位检测

在 LPUART 中，先在 Rx 线路上出现下降沿时检测起始位，然后在起始位中间采样以确认电平是否仍为“0”。如果起始采样为“1”，则噪声错误标志 (NE) 置 1，起始位将被丢弃，接收器将等待新的起始位。否则，接收器将继续正常采样所有传入位。

字符接收

LPUART 接收期间，首先通过 RX 引脚移入数据的最低有效位（默认配置）。该模式下，LPUART_RDR 寄存器的缓冲区 (RDR) 位于内部总线和接收移位寄存器之间。

字符接收步骤

要接收字符，需遵循以下步骤：

1. 对 LPUART_CR1 中的 M 位进行编程以定义字长。
2. 使用波特率寄存器 LPUART_BRR 选择所需波特率。
3. 对 LPUART_CR2 中的停止位数量进行编程。
4. 通过向 LPUART_CR1 寄存器中的 UE 位写入“1”使能 LPUART。
5. 如果将进行多缓冲区通信，请选择 LPUART_CR3 中的 DMA 使能 (DMAR)。按照*第 33.5.10 节：USART 多处理器通信*中的说明配置 DMA 寄存器。
6. 将 RE 位 LPUART_CR1 置 1。这一操作将使能接收器开始搜索起始位。

接收到字符时

- 禁止 FIFO 模式时，RXNE 位置 1。这表明移位寄存器的内容已传送到 RDR。也就是说，已接收到并可读取数据（及其相应的错误标志）。
- 如果已使能 FIFO 模式，则 RXFNE 位置 1，这表示 RXFIFO 非空。读取 LPUART_RDR 会返回输入到 RXFIFO 中的最早数据。接收到数据时，数据以及相应的错误位将一起被存储在 RXFIFO 中。
- 如果 RXNEIE（FIFO 模式下时为 RXFNEIE）位置 1，则会生成中断。
- 如果接收期间已检测到帧错误、噪声错误或上溢错误，错误标志位可置 1。

- 在多缓冲区通信模式下：
  - 禁止 FIFO 模式时，每接收到一个字节后 RXNE 标志均置 1，然后通过 DMA 对接收数据寄存器执行读操作清零。
  - 如果使能 FIFO 模式，则 RXFNE 标志在 RXFIFO 非空时置 1。每次收到 DMA 请求后，都会从 RXFIFO 检索数据。在 RXFIFO 非空时（即，当 RXFIFO 中存在要读取的数据时），会触发 DMA 请求。
- 在单缓冲区模式下：
  - 如果禁止 FIFO 模式，则通过软件对 LPUART_RDR 寄存器进行读操作来将 RXNE 标志清零。也可以通过将 LPUART_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE 标志位清零。RXNE 位必须在结束接收下一个字符前清零，以避免发生上溢错误。
  - 如果使能 FIFO 模式，则 RXFNE 标志在 RXFIFO 非空时置 1。每次对 LPUART_RDR 寄存器执行完读操作后，都会从 RXFIFO 中检索数据。RXFIFO 为空时，RXFNE 标志将清零。也可以通过将 LPUART_RQR 寄存器中的 RXFRQ 位置 1 将 RXFNE 标志清零。RXFIFO 已满时，必须在结束接收下一个字符前读取 RXFIFO 中的第一个条目，以避免发生上溢错误。当 RXFNEIE 位置 1 时，RXFNE 标志会生成中断。或者，当达到 RXFIFO 阈值时，会生成中断并从 RXFIFO 中读取数据。在这种情况下，CPU 可读取由编程的阈值定义的数据块。

中断字符

接收到中断字符时，USART 将会按照帧错误对其进行处理。

空闲字符

检测到空闲帧时，除了在 IDLEIE 位置 1 时会生成中断外，处理步骤与接收到数据字符的情况相同。

上溢错误

- 禁止 FIFO 模式

  如果在 RXNE 未复位时接收到字符，则会发生上溢错误。

  RXNE 位清零前，数据无法从移位寄存器传送到 RDR 寄存器。每接收到一个字节后，RXNE 标志位都将置 1。

  当 RXNE 标志位置 1 时，如果在接收到下一个数据或尚未处理上一个 DMA 请求，则会发生上溢错误。发生上溢错误时：
  - ORE 位置 1。
  - RDR 中的内容不会丢失。对 LPUART_RDR 执行读操作时可使用先前的数据。
  - 移位寄存器将被覆盖。之后，上溢期间接收到的任何数据都将丢失。
  - 如果 RXNEIE 位置 1 或 EIE 位置 1，则会生成中断。
- 使能 FIFO 模式

  移位寄存器准备好传送且接收 FIFO 已满时，会发生上溢错误。

  在 RXFIFO 中存在一个空闲位置之前，数据无法从移位寄存器传送到 LPUART_RDR 寄存器。当 RXFIFO 非空时，RXFNE 标志置 1。

如果 RXFIFO 已满且移位寄存器已准备好传送，则会发生上溢错误。发生上溢错误时：

– ORE 位置 1。
– RXFIFO 中的第一个条目不会丢失。可通过对 LPUART_RDR 执行读操作来获取。
– 移位寄存器将被覆盖。之后，上溢期间接收到的任何数据都将丢失。
– 如果 RXFNEIE 位置 1 或 EIE 位置 1，则会生成中断。

通过将 ICR 寄存器中的 ORECF 位置 1 来复位 ORE 位。

注： *ORE 位置 1 时表示至少 1 个数据丢失。*

禁止 FIFO 模式时，有以下两种可能

- *如果 RXNE = 1，则最后一个有效数据存储于接收寄存器 (RDR) 中并且可进行读取。*
- *如果 RXNE=0，则表示最后一个有效数据已被读取，因此 RDR 中没有要读取的数据。会发生这种情况，已读取 RDR 寄存器中的最后一个有效数据的同时接收到新（并且丢失）的数据。*

选择时钟源

时钟源的选择通过时钟控制系统完成（请参见*复位和时钟控制器 (RCC) 部分*）。在使能 LPUART 之前，必须通过 UE 位选择时钟源。

必须遵循以下两个条件选择时钟源：

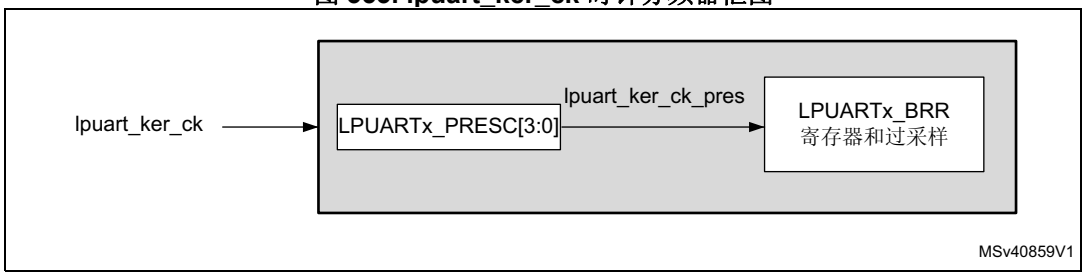
- 可在低功耗模式下使用 LPUART
- 通信速度

时钟源频率为 lpuart_ker_ck。

如果支持双时钟域和从低功耗模式唤醒功能，则 lpuart_ker_ck 时钟源可在 RCC 中进行配置（请参见*复位和时钟控制器 (RCC) 部分*）。否则，lpuart_ker_ck 与 lpuart_pclk 相同。

lpuart_ker_ck 可根据 LPUART_PRESC 寄存器中的可编程系数进行分频。

图 353. lpuart_ker_ck 时钟分频器框图



某些 lpuart_ker_ck 时钟源允许 LPUART 在 MCU 处于低功耗模式时接收数据。必要时，LPUART 可基于所接收的数据和唤醒模式的选择来唤醒 MCU，以通过用软件读取 LPUART_RDR 寄存器的方式或通过 DMA 的方式传输已接收数据。

对于其他时钟源，系统必须激活才能进行 LPUART 通信。

时钟源还决定通信速度范围（尤其是最大通信速度）。

接收器在尽可能靠近波特周期中间的位置采样每个传入波特。每个传入波特仅进行一次采样。

注： *不进行数据噪声检测。*

帧错误

如果接收数据时未在预期时间内识别出停止位，从而出现同步失效或过度的噪声，则会检测到帧错误。

检测到帧错误时：

- FE 位由硬件置 1。
- 无效数据从移位寄存器传送到 LPUART_RDR 寄存器。
- 单字节通信时无中断产生。然而，在 RXNE 位产生中断时，该位出现上升沿。多缓冲区通信时，LPUART_CR3 寄存器中的 EIE 位置 1 时将发出中断。

通过将 1 写入 LPUART_ICR 寄存器中的 FECF 位来复位 FE 位。

接收期间可配置的停止位

可通过 LPUART_CR2 的控制位配置要接收的停止位的数量：可以是 1 个或 2 个（正常模式下）。

- **1 个停止位**：将在第 8、第 9 和第 10 次采样时对 1 个停止位进行采样。
- **2 个停止位**：将在第二个停止位的中间对 2 个停止位进行采样。RXNE 和 FE 标志在此采样期间（例如，在第二个停止位期间）置 1。发生帧错误时不检测第一个停止位。

34.3.7 LPUART 波特率生成

接收器和发送器（Rx 和 Tx）的波特率均设置为 LPUART_BRR 寄存器中编程的值。

$$\text{Tx/Rx 波特} = \frac{256 \times \text{lpuartckpres}}{\text{LPUARTDIV}}$$

LPUARTDIV 在 LPUART_BRR 寄存器中定义。

注： *对 LPUART_BRR 执行写操作后，波特率计数器更新为波特率寄存器中的新值。因此，波特率寄存器的值不应在通信时发生更改。*

禁止在 LPUART_BRR 寄存器中写入小于 0x300 的值。

f_{CK} *必须介于 3 x 波特率到 4096 x 波特率之间。*

LPUART 时钟源为 LSE 时可达到的最大波特率为 9600 波特。当 LPUART 由与 LSE 时钟不同的时钟源计时，可以达到更高的波特率。例如，如果 LPUART 时钟源频率为 100 MHz，则可达到的最大波特率约为 33 M 波特。

表 199. lpuart_ker_ck_pres = 32,768 KHz 时编程的波特率的误差计算

波特率		lpuart_ker_ck_pres = 32,768 KHz		
序号	所需值	实际值	波特率寄存器中编程的值	误差 % = (计算值 - 所需值) / 所需波特率
1	0.3 KBps	0.3 KBps	0x6D3A	0
2	0.6 Kbps	0.6 Kbps	0x369D	0
3	1200 Bps	1200.087 Bps	0x1B4E	0.007
4	2400 Bps	2400.17 Bps	0xDA7	0.007
5	4800 Bps	4801.72 Bps	0x6D3	0.035
6	9600 KBps	9608.94 Bps	0x369	0.093

表 200. 采用 16 倍过采样 (OVER8 = 0) 时，在 f_{CK} = 100 MHz 下

波特率		f_{CK} = 100 MHz		
序号	所需值	实际值	波特率寄存器中编程的值	误差 % = (计算值 - 所需值) / 所需波特率
1	38400 波特	38400,04 波特	A2C2A	0,0001
2	57600 波特	57600,06 波特	6C81C	0,0001
3	115200 波特	115200,12 波特	3640E	0,0001
4	230400 波特	230400,23 波特	1B207	0,0001
5	460800 波特	460804,61 波特	D903	0,001
6	921600 波特	921625,81 波特	6C81	0,0028
7	4000 K 波特	4000000,00 波特	1900	0
8	10000 K 波特	10000000,00 波特	A00	0
9	20000 K 波特	20000000,00 波特	500	0
10	30000 K 波特	33032258,06 波特	307	0,1

34.3.8 LPUART 接收器对时钟偏差的容差

仅当总时钟系统偏差小于 LPUART 接收器的容差时，LPUART 异步接收器才能正常工作。影响总偏差的因素包括：

- DTRA：发送器误差引起的偏差（其中还包括发送器本地振荡器的偏差）
- DQUANT：接收器的波特率量化引起的误差
- DREC：接收器本地振荡器的偏差
- DTCL：传输线路引起的偏差（通常是由于收发器所引起，它可能会在低电平到高电平转换时序与高电平到低电平转换时序之间引入不对称）

DTRA + DQUANT + DREC + DTCL + DWU < LPUART 接收器偏差

其中

DWU 是使用从低功耗模式唤醒时因采样点偏差而产生的误差。

LPUART 接收器在*表 201* 中指定的最大容许偏差下可正确接收数据：

- 通过 LPUART_CR2 寄存器中的 STOP[1:0] 位定义的停止位数
- LPUART_BRR 寄存器值

表 201. LPUART 接收器的容差

M 位	768 < BRR < 1024	1024 < BRR < 2048	2048 < BRR < 4096	4096 ≤ BRR
8 位（M=00），1 个停止位	1.82%	2.56%	3.90%	4.42%
9 位（M=01），1 个停止位	1.69%	2.33%	2.53%	4.14%
7 位（M=“10”），1 个停止位	2.08%	2.86%	4.35%	4.42%
8 位（M=00），2 个停止位	2.08%	2.86%	4.35%	4.42%
9 位（M=01），2 个停止位	1.82%	2.56%	3.90%	4.42%
7 位（M=“10”），2 个停止位	2.34%	3.23%	4.92%	4.42%

注： *当接收的帧恰好包含 10 个（M 位 = “00”）、11 个（M 位 = “01”）或 9 个（M 位 = “10”）位时间的空闲帧时，表 201* 中指定的数据可能与特例中的数据略微不同。

34.3.9 LPUART 多处理器通信

可以执行 LPUART 多处理器通信（多个 LPUART 连接在一个网络中）。例如，其中一个 LPUART 可以是主器件，其 TX 输出与其他 LPUART 的 RX 输入相连接。其他 LPUART 为从器件，其各自的 TX 输出在逻辑上通过与运算连在一起，并与主 LPUART 的 RX 输入相连。

在多处理器配置中，理想情况下通常只有预期的消息接收方主动接收完整的消息内容，从而减少由所有未被寻址的接收器造成的冗余 LPUART 服务开销。

可通过静默功能将未被寻址的器件置于静默模式下。为了使用静默模式功能，必须将 LPUART_CR1 寄存器中的 MME 位置 1。

注： *使能 FIFO 管理且 MME 已置 1 时，不得清零 MME 位再快速将其置 1（在两个 lpuart_ker_ck 周期内），否则静默模式可能保持有效。*

使能静默模式时：

- 不得将接收状态位置 1。
- 禁止任何接收中断。
- LPUART_ISR 寄存器中的 RWU 位置“1”。在某些情况下，RWU 可以由硬件或软件通过 LPUART_RQR 寄存器中的 MMRQ 位自动控制。

根据 LPUART_CR1 寄存器中 WAKE 位的设置，LPUART 可使用以下两种方法进入或退出静默模式：

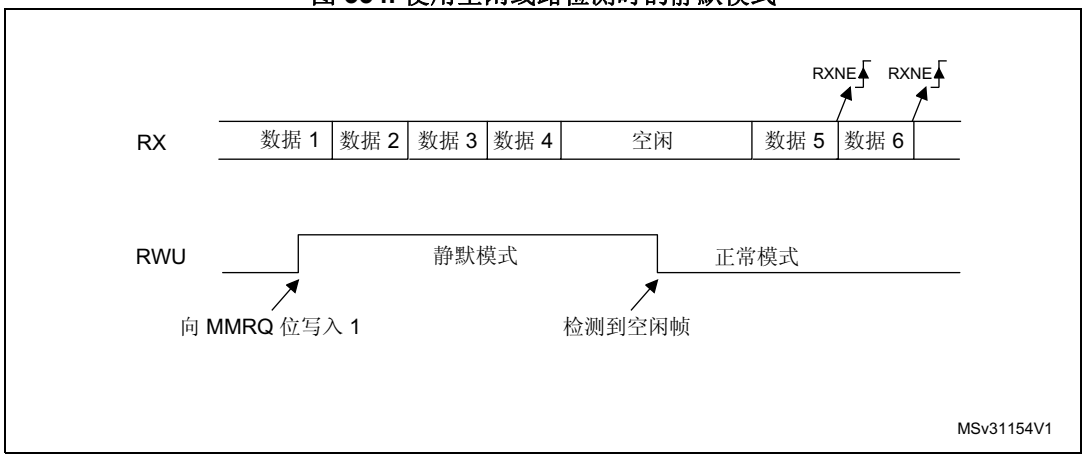
- 如果 WAKE 位被复位，则进行空闲线路检测。
- 如果 WAKE 位置 1，则进行地址标记检测。

空闲线路检测 (WAKE=0)

当向 MMRQ 位写入 1 且 RWU 位自动置 1 时，LPUART 进入静默模式。

检测到空闲帧时，LPUART 将唤醒。此时 RWU 位会由硬件清零，但 LPUART_ISR 寄存器中的 IDLE 位不会置 1。*图 354* 中给出了使用空闲线路检测时静默模式行为的示例。

图 354. 使用空闲线路检测时的静默模式



注： *如果在 IDLE 字符已经过去时将 MMRQ 位置 1，将不会进入静默模式（RWU 未置 1）。*

如果在线路处于空闲状态时激活 LPUART，在一个 IDLE 帧持续时间后（不只在接收一个字符帧后）会检测到空闲状态。

4 位/7 位地址标记检测 (WAKE=1)

在此模式下，如果字节的 MSB 为 1，则将这些字节识别为地址，否则将其识别为数据。在地址字节中，目标接收器的地址位于 4 个或 7 个 LSB 中。7 位或 4 位地址检测通过 ADDM7 位来选择。接收器会将此 4 位/7 位字与其地址进行比较，该接收器的地址在 LPUART_CR2 寄存器的 ADD 位中进行设置。

注： *在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。*

当接收到与其编程地址不匹配的地址字符时，LPUART 会进入静默模式。此时，RWU 位将由硬件置 1。LPUART 进入静默模式后，RXNE 标志不会针对此地址字节置 1，也不会发出中断或 DMA 请求。

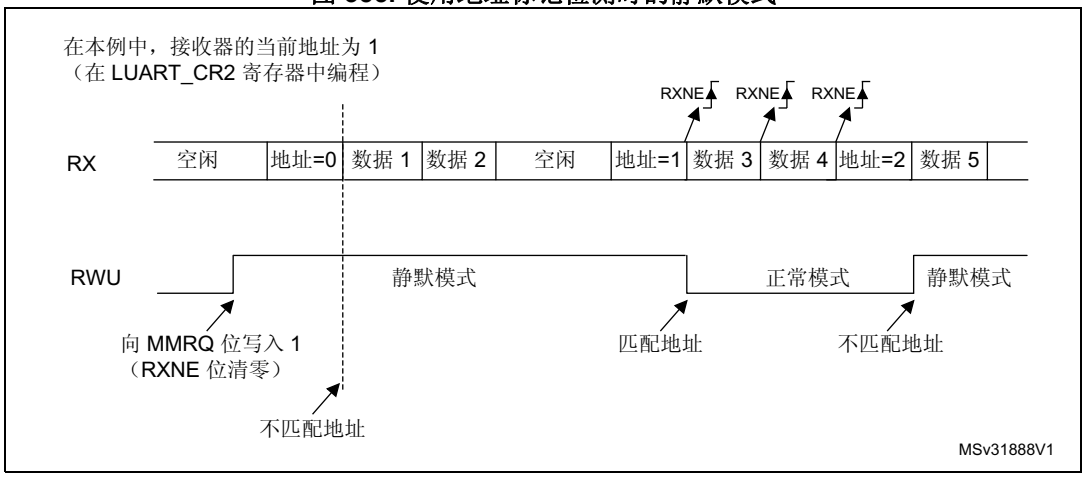
当向 MMRQ 位写入“1”时，LPUART 也会进入静默模式。这种情况下，RWU 位也自动置 1。

当接收到与编程地址匹配的地址字符时，LPUART 会退出静默模式。然后 RWU 位被清零，可以开始正常接收后续字节。由于 RWU 位已清零，RXNE/RXFNE 位会针对地址字符置 1。

注： *使能 FIFO 管理时，如果在接收器对数据的最后一位进行采样时 MMRQ 位置 1，则可在有效进入静默模式之前接收该数据。*

图 355 中给出了使用地址标记检测时静默模式行为的示例。

图 355. 使用地址标记检测时的静默模式



34.3.10 LPUART 极性控制

将 LPUART_CR1 寄存器中的 PCE 位置 1，可以使能奇偶校验控制（发送时生成奇偶校验位，接收时进行奇偶校验检查）。根据 M 位定义的帧长度，*表 202* 中列出了可能的 LPUART 帧格式。

表 202：LPUART 帧格式

M 位	PCE 位	LPUART 帧(1)
00	0	| SB | 8 位数据 | STB |
00	1	| SB | 7 位数据 | PB | STB |
01	0	| SB | 9 位数据 | STB |
01	1	| SB | 8 位数据 PB | STB |
10	0	| SB | 7 位数据 | STB |
10	1	| SB | 6 位数据 | PB | STB |

1. 图注：SB：起始位，STB：停止位，PB：奇偶校验位。
2. 在数据寄存器中，PB 始终位于 MSB 位置（第 8 位或第 7 位，具体取决于 M 位的值）。

偶校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为偶数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据=00110101 且 4 个位置 1，则在选择偶校验（LPUART_CR1 寄存器中的 PS 位 = 0）时，校验位为 0。

奇校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为奇数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据=00110101 且 4 个位置 1，则在选择奇校验（LPUART_CR1 寄存器中的 PS 位 = 1）时，校验位为 1。

接收时进行奇偶校验检查

如果奇偶校验检查失败，则 LPUART_ISR 寄存器中的 PE 标志置 1；如果 LPUART_CR1 寄存器中 PEIE 位置 1，则会生成中断。通过软件将 1 写入 LPUART_ICR 寄存器中的 PECF 位来清零 PE 标志。

发送时的奇偶校验生成

如果 LPUART_CR1 寄存器中的 PCE 位置 1，则在数据寄存器中所写入数据的 MSB 位会进行传送，但是会由奇偶校验位进行更改（如果选择偶校验 (PS=0)，则“1”的数量为偶数；如果选择奇校验 (PS=1)，则“1”的数量为奇数）。

34.3.11 LPUART 单线半双工通信

通过将 LPUART_CR3 寄存器中的 HDSEL 位置 1 来选择单线半双工模式。在此模式下，必须将以下位清零：

- LPUART_CR2 寄存器中的 LINEN 和 CLKEN 位。
- LPUART_CR3 寄存器中的 SCEN 和 IREN 位。

LPUART 可以配置为遵循单线半双工协议，其中 TX 和 RX 线路从内部相连接。使用 LPUART_CR3 寄存器中的控制位 HDSEL 可在半双工通信和全双工通信间进行选择。

向 HDSEL 位写入“1”后：

- TX 和 RX 线路从内部相连接。
- 不能再使用 RX 引脚。
- 无数据传输时，TX 引脚始终处于释放状态。因此，它在空闲状态或接收过程中用作标准 I/O。这意味着，必须对 I/O 进行配置，以便将 TX 配置为复用功能开漏并外接上拉电阻。

除此之外，通信协议与正常 LPUART 模式下的通信协议相似。此线路上的任何冲突都必须由软件管理（例如，使用中央仲裁器）。尤其要注意，发送过程永远不会被硬件封锁，只要数据是在 TE 位置 1 的情况下写入，发送就会持续进行。

注： *在 LPUART 通信中，当进行 1 个停止位配置时，RXNE 标志在停止位中间置 1。*

34.3.12 使用 DMA 和 LPUART 进行连续通信

LPUART 能够使用 DMA 进行连续通信。接收缓冲区和发送缓冲区的 DMA 请求是独立生成的。

注： *要确定是否支持 DMA 模式，请参见第 996 页的第 33.4 节：USART 实现。如果不支持 DMA，请按照第 33.5.6 节中的说明使用 LPUSRT。当 FIFO 禁止时，可以将 LPUART_ISR 寄存器中的 TXE/RXNE 标志清零，从而实现连续通信。*

使用 DMA 进行发送

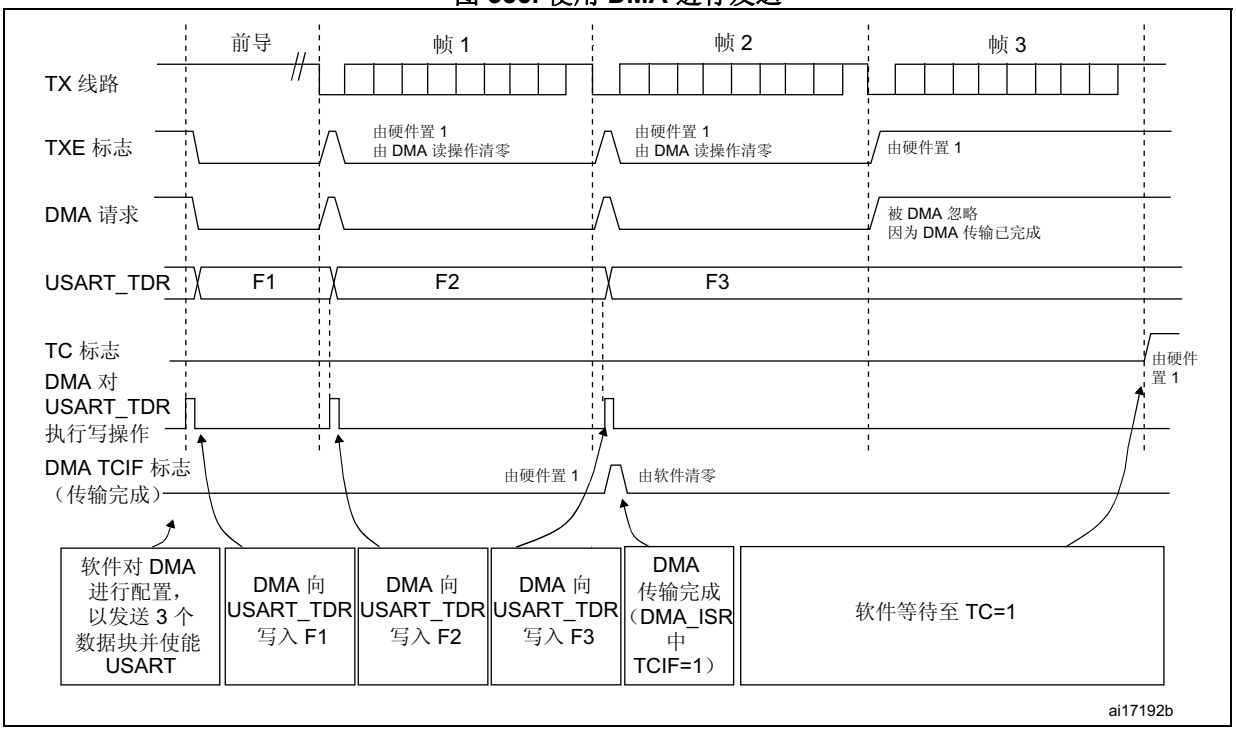
将 LPUART_CR3 寄存器中的 DMAT 位置 1 可以使能 DMA 模式进行发送。当 TXE 标志（使能 FIFO 模式时为 TXFNF 标志）置 1 时，可使用 DMA 外设（请参见相应的*直接存储器访问控制器部分*）将数据从配置的 SRAM 区加载到 LPUART_TDR 寄存器。要映射一个 DMA 通道以进行 LPUART 发送，请按以下步骤操作（x 表示通道编号）：

1. 在 DMA 控制寄存器中写入 LPUART_TDR 寄存器地址，将其配置为传输的目标地址。每次发生 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据都从存储器移动到此地址。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的源地址。每次发生 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据都从该存储区域加载到 LPUART_TDR 寄存器中。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生 DMA 中断。
6. 通过将 LPUART_ICR 寄存器中的 TCCF 位置 1，将 LPUART_ISR 寄存器中的 TC 标志清零。
7. 在 DMA 寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

在发送模式下，DMA 对所有要发送的数据执行了写操作（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 TC 标志进行监视，以确保 LPUART 通信已完成。在禁止 LPUART 或进入低功耗模式前必须执行此步骤，以避免损坏最后一次发送。软件必须等待直到 TC=1。TC 标志在所有数据发送期间都保持清零状态，然后在最后一帧发送结束时由硬件置 1。

图 356. 使用 DMA 进行发送



注：　使能 FIFO 管理时，DMA 请求由发送 FIFO 未满（即 TXFNF = 1）触发。

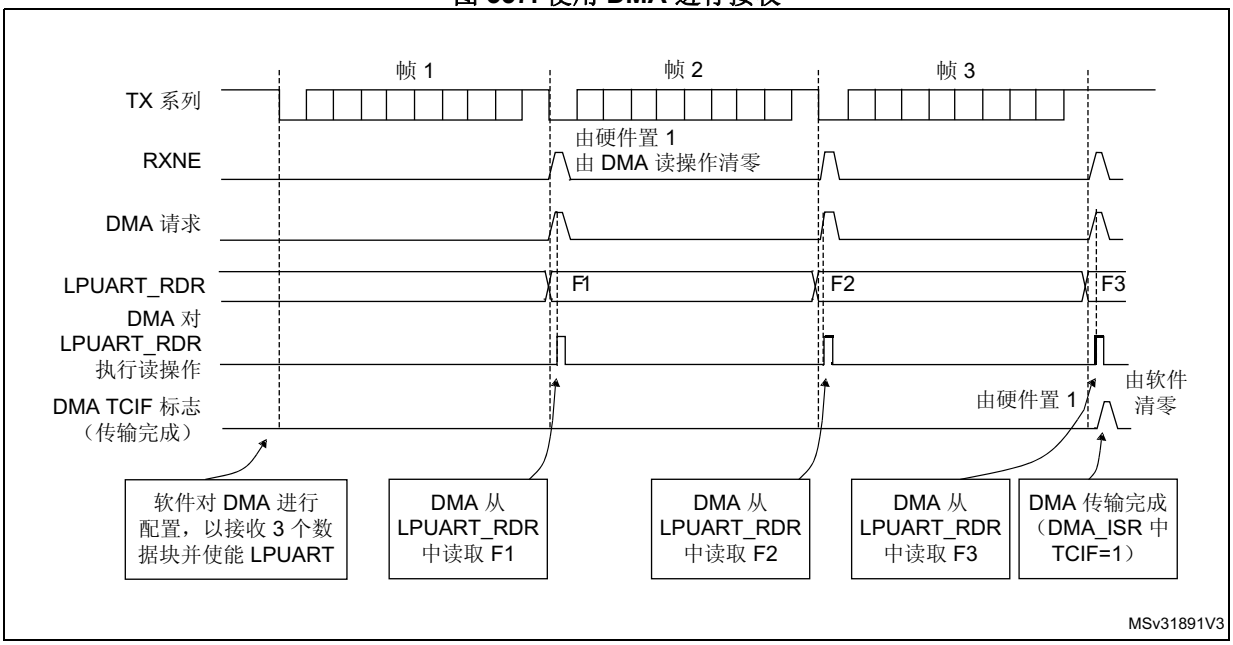
使用 DMA 进行接收

将 LPUART_CR3 寄存器中的 DMAR 位置 1 可以使能 DMA 模式进行接收。接收数据字节时，可使用 DMA 外设将数据从 LPUART_RDR 寄存器加载到配置的 SRAM 区域中（请参见相应的*直接存储器访问控制器 (DMA) 部分*）。要映射一个 DMA 通道以进行 LPUART 接收，请按以下步骤操作：

1. 在 DMA 控制寄存器中写入 LPUART_RDR 寄存器地址，将其配置为传输的源地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都从该地址移动到存储器。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的目标地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都从 LPUART_RDR 加载到此存储区域。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 控制寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生中断。
6. 在 DMA 控制寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

图 357. 使用 DMA 进行接收



注： *使能 FIFO 管理时，DMA 请求由接收 FIFO 非空（即 RXFNE = 1）触发。*

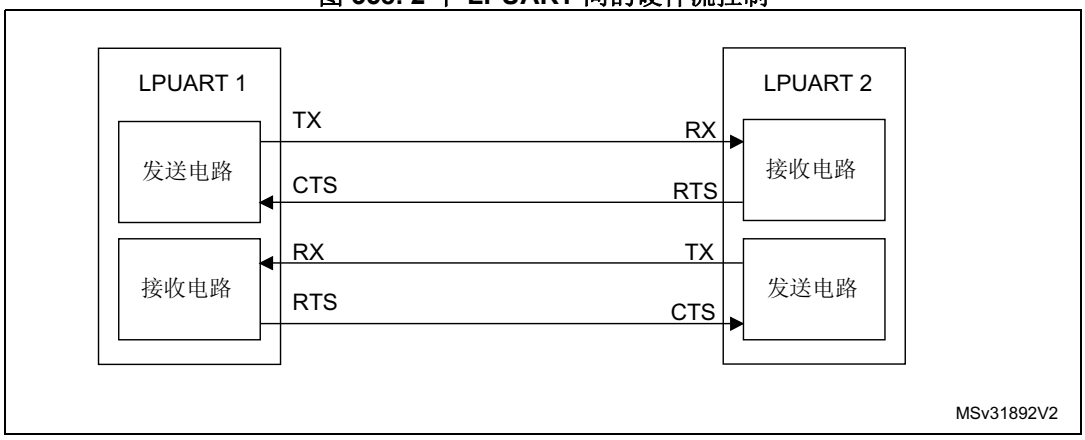
多缓冲区通信中的错误标志和中断生成

在多缓冲区通信模式下，如果事务中发生任何错误，都会在当前字节后将相应错误标志置 1。如果中断使能标志置 1，则会产生中断。在单字节接收过程中，与 RXNE（使能 FIFO 模式时为 RXFNE）一同置位的帧错误、上溢错误和噪声标志具有单独的错误标志中断使能位（LPUART_CR3 寄存器中的 EIE 位）；如果该位置 1，在发生其中任何一个错误时，都会在当前字节后使能中断。

34.3.13 RS232 硬件流控制和 RS485 驱动器使能

使用 nCTS 输入和 nRTS 输出可以控制 2 个器件间的串行数据流。*图 344* 显示了在这种模式下如何连接 2 个器件：

图 358. 2 个 LPUART 间的硬件流控制

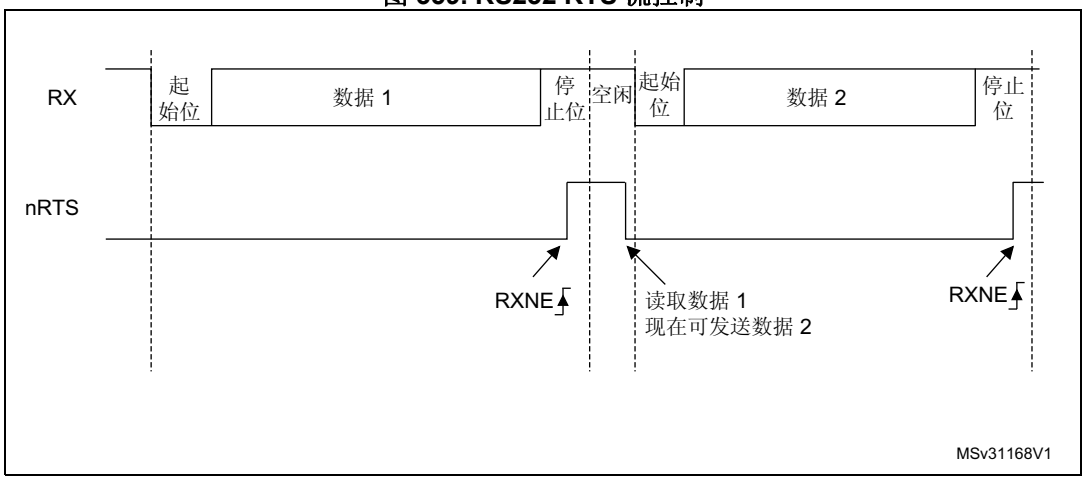


分别向 LPUART_CR3 寄存器中的 RTSE 位和 CTSE 位写入 1，可以分别使能 RS232 RTS 和 CTS 流控制。

RS232 RTS 流控制

如果使能 RTS 流控制 (RTSE=1)，只要 LPUART 接收器准备好接收新数据，便会将 nRTS 变为有效（连接到低电平）。当接收寄存器已满时，会将 nRTS 变为无效，表明发送过程会在当前帧结束后停止。*图 359* 显示了在使能 RTS 流控制的情况下进行通信的示例。

图 359. RS232 RTS 流控制



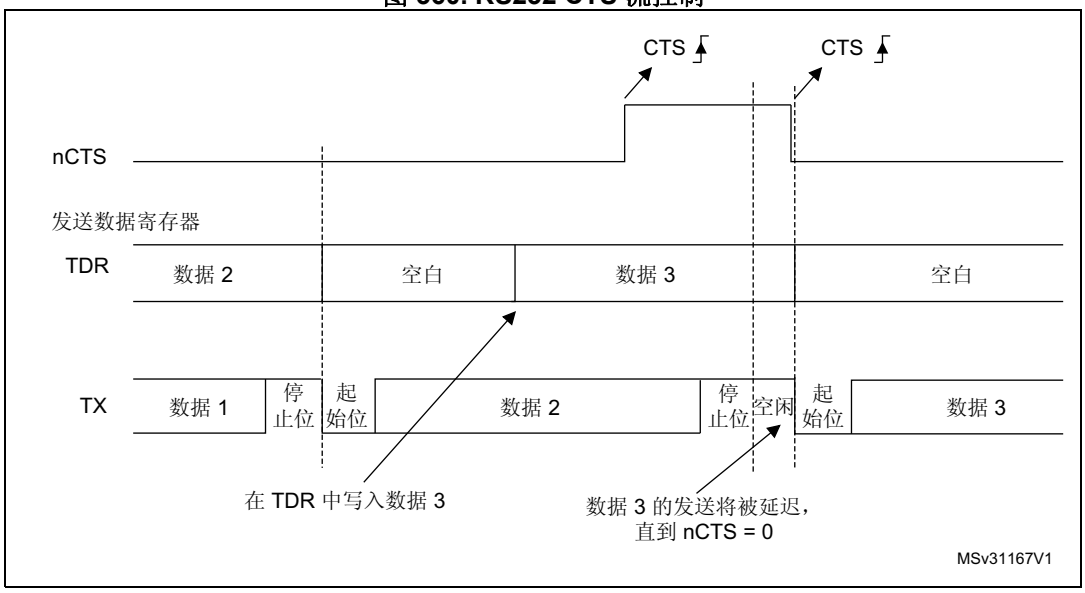
注： *如果使能 FIFO 模式，则仅当 RXFIFO 已满时，nRTS 才会变为无效。*

RS232 CTS 流控制

如果使能 CTS 流控制 (CTSE=1)，则发送器会在发送下一帧前检查 nCTS。如果 nCTS 有效（连接到低电平），则会发送下一数据（假设数据已准备好发送，即 TXE/TXFE=0）；否则不会进行发送。如果在发送过程中 nCTS 变为无效，则当前发送完成之后，发送器停止。

当 CTSE=1 时，只要 nCTS 发生变化，CTSIF 状态位便会由硬件自动置 1。这指示接收器是否已准备好进行通信。如果 LPUART_CR3 寄存器中的 CTSIE 位置 1，则会产生中断。*图 360* 显示了在使能 CTS 流控制的情况下进行通信的示例。

图 360. RS232 CTS 流控制



注： *为正常运行，必须在当前字符结束前至少 3 个 LPUART 时钟源周期内使能 nCTS。此外还应注意，当脉冲短于 2 个 PCLK 周期时无法将 CTSCF 标志置 1。*

RS485 驱动器使能

驱动器使能功能可通过将 LPUART_CR3 控制寄存器中的 DEM 位置 1 使能。这样便可通过 DE（驱动器使能）信号激活外部收发器控制。使能时间为激活 DE 信号与起始位开始间的时间。可以通过 LPUART_CR1 控制寄存器中的 DEAT [4:0] 位域编程禁止时间。禁止时间为发送的消息中最后一个停止位结束与取消激活 DE 信号间的时间。可以通过 LPUART_CR1 控制寄存器中的 DEDT [4:0] 位域编程禁止时间。DE 信号的极性可使用 LPUART_CR3 控制寄存器中的 DEP 位配置。

LPUART 的 DEAT 和 DEDT 用 LPUART 时钟源 (f_{CK}) 周期表示：

- 驱动器使能有效时间等于
  - (1 + (DEAT x P)) x f_{CK}（如果 P # 0）
  - (1 + DEAT) x f_{CK}（如果 P = 0）
- 驱动器使能禁止时间等于
  - (1 + (DEDT x P)) x f_{CK}（如果 P # 0）
  - (1 + DEDT) x f_{CK}（如果 P = 0）

其中 P = BRR[20:11]

34.3.14 LPUART 低功耗管理

LPUART 具有高级低功耗模式功能，即使禁止 lpuart_pclk 时钟，也能正常传输数据。
UESM 位置 1 时，LPUART 能够将 MCU 从低功耗模式唤醒。
对 usart_pclk 进行门控时，如果某些特定操作需要激活 **usart_pclk** 时钟，则 LPUART 会提供唤醒中断 (**usart_wkup**)：

- 禁止 FIFO 模式时

  必须激活 luart_pclk 时钟以清空 LPUART 数据寄存器。

  在这种情况下，lpuart_wkup 中断源为 RXNE 置“1”。RXNEIE 位必须在进入低功耗模式之前置 1。

- 使能 FIFO 模式时

  必须激活 luart_pclk 时钟以

  – 填充 TXFIFO
  – 或清空 RXFIFO

  在这种情况下，lpuart_wkup 中断源可以为：

  – RXFIFO 非空。此时，RXFNEIE 位必须在进入低功耗模式之前置 1。
  – RxFIFO 已满。此时，RXFFIE 位必须在进入低功耗模式之前置 1，接收到的数据量对应于 RXFIFO 大小，并且 RXFF 标志未置 1。
  – TXFIFO 为空。此时，TXFEIE 位必须在进入低功耗模式之前置 1。

  这允许在低功耗模式下发送/接收 TXFIFO/RXFIFO 中的数据。

  为了避免发生上溢/下溢错误以及在低功耗模式下发送/接收数据，lpuart_wkup 中断源可以是以下事件之一：

  – 达到 TXFIFO 阈值。此时，TXFTIE 位必须在进入低功耗模式之前置 1。
  – 达到 RXFIFO 阈值。此时，RXFTIE 位必须在进入低功耗模式之前置 1。

  例如，如果唤醒时间短于在线路上接收单个字节所需的时间，则应用可将阈值设为最大 RXFIFO 大小。

  使用 RXFIFO 已满、TXFIFO 为空、RXFIFO 非空和 RXFIFO/TXFIFO 阈值中断将 MCU 从低功耗模式唤醒时，可在低功耗模式下尽可能多地进行 LPUART 传输，这样有助于优化功耗。

或者，也可通过 WUS 位域选择特定 **lpuart_wkup** 中断。

检测到唤醒事件后，WUF 标志会由硬件置 1 并在 WUFIE 位置 1 时生成 **lpuart_wkup** 中断。在这种情况下，无需 **lpuart_wkup** 中断即可唤醒。将 WUF 置 1 便足以将 MCU 从低功耗模式唤醒。

注： *在进入低功耗模式之前，请确保未进行任何 LPUART 传输。检查 BUSY 标志不能确保在进行数据接收时绝不会进入低功耗模式。*

WUF 标志在检测到唤醒事件时置 1，而与 MCU 处于低功耗模式还是工作模式无关。

如果在初始化和使能接收器后立即进入低功耗模式，则必须校验 REACK 位以确保 LPUART 确实已使能。

当 DMA 用于接收时，它必须在进入低功耗模式前禁止，并在退出低功耗模式后重新使能。

如果使能 FIFO，则只有在使能静默模式的情况下才能在地址匹配时从低功耗模式唤醒。

使用静默模式和低功耗模式

如果 LPUART 在进入低功耗模式前处于静默模式：

- 不得使用空闲检测时从静默模式唤醒，因为空闲检测无法在低功耗模式下工作。
- 如果使用地址匹配时从静默模式唤醒，则低功耗模式唤醒源也必须是地址匹配。如果 RXNE 标志在进入低功耗模式时置 1，则接口将在地址匹配时和从低功耗模式唤醒时保持静默模式。

注： *使能 FIFO 管理时，静默模式可与从低功耗模式唤醒搭配使用，而且没有任何限制（即，上述关于静默和低功耗模式的两点仅在 FIFO 管理禁止时有效）。*

LPUART 内核时钟 lpuart_ker_ck 在低功耗模式下关闭时从低功耗模式唤醒

在低功耗模式下，如果在 LPUART 接收线上检测到下降沿时 lpuart_ker_ck 时钟处于关闭状态，则 LPUART 接口会借助 lpuart_ker_ck_req 信号请求开启 lpuart_ker_ck 时钟。lpuart_ker_ck 随后用来接收帧。

如果唤醒事件通过验证，则 MCU 将从低功耗模式唤醒，并会正常进行数据接收。

如果唤醒事件未通过验证，则 usart_ker_ck 会再次关闭，MCU 不会被唤醒并保持在低功耗模式下，且内核时钟请求被释放。

以下示例显示了将唤醒事件编程为“地址匹配检测”并禁止 FIFO 管理的情况。

图 361 所示为唤醒事件通过验证时的行为。

图 361. 唤醒事件通过验证（唤醒事件 = 地址匹配，禁止 FIFO）

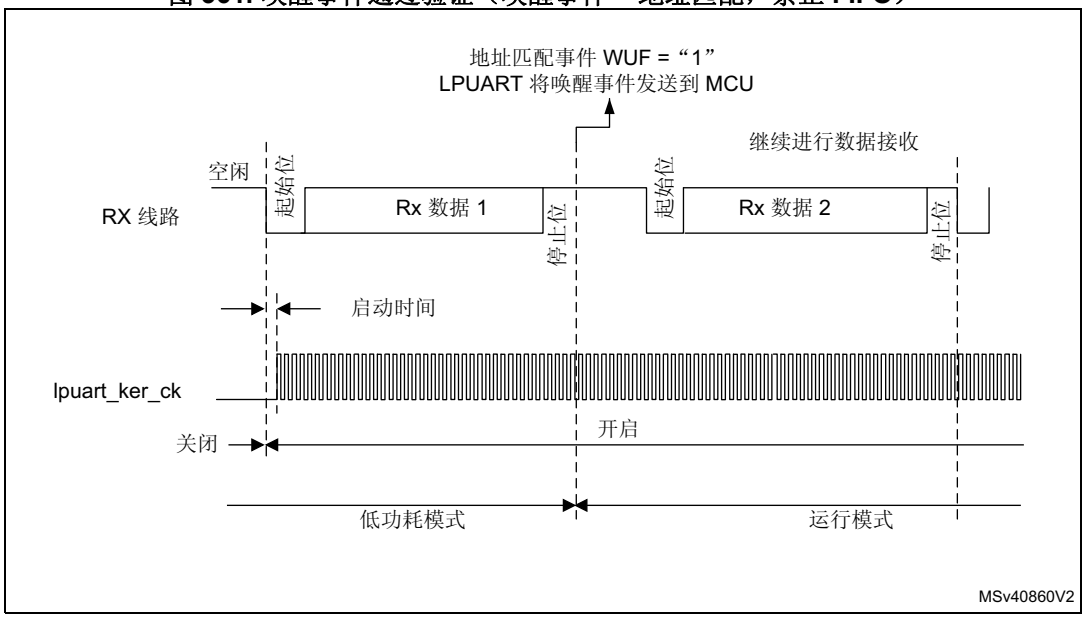
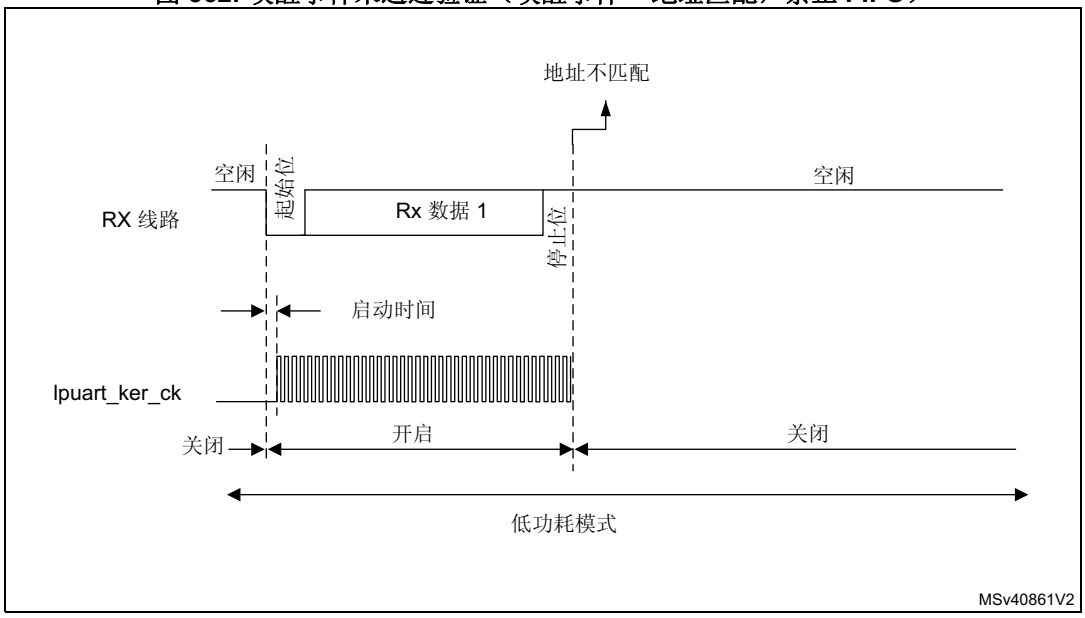


图 362 所示为唤醒事件未通过验证时的行为。

图 362. 唤醒事件未通过验证（唤醒事件 = 地址匹配，禁止 FIFO）



注： *将地址匹配或任一接收的帧用作唤醒事件时，如上两图适用。如果唤醒事件为起始位检测，则 LPUART 会在起始位结束时向 MCU 发送唤醒事件。*

确定允许从低功耗模式正确唤醒 MCU 的最大 LPUART 波特率

允许从低功耗模式正确唤醒 MCU 的最大波特率取决于唤醒时间参数（请参见器件数据手册）和 LPUART 接收器容差（请参见*第 34.3.8 节：LPUART 接收器对时钟偏差的容差*）。

举例来说：OVER8 = 0，M 位 = “01”，ONEBIT = 0，BRR [3:0] = 0000。

在这些条件下，根据*表 201：LPUART 接收器的容差*，LPUART 接收器的容差为 3.41%。

DTRA + DQUANT + DREC + DTCL + DWU < LPUART 接收器的容差

$D_{WUmax} = t_{WULPUART} / (11 \times T_{bit\ Min})$

$T_{bit\ Min} = = t_{WULPUART} / (11 \times D_{WUmax})$

其中 $t_{WULPUART}$ 为从低功耗模式唤醒的时间。

考虑一种理想情况：参数 DTRA、DQUANT、DREC 和 DTCL 为 0%，则 DWU 的最大值为 3.41%。实际上，我们至少需要考虑 lpuart_ker_ck 不精确的情况。

例如，如果将 HSI 用作 lpuart_ker_ck，HSI 的不精确度为 1%，则可以得到：

$t_{WULPUART}$ = 3 µs（仅提供数值作为参考；有关正确数值，请参见器件数据手册）

D_{WUmax} = 3.41% - 1% = 2.41%

$T_{bit\ min}$ = 3 µs/ (11 x 2.41%) = 11.32 µs

结果，允许从低功耗模式正确唤醒的最大波特率为：1/11.32 µs = 88.36 K 波特。

34.4 LPUART 中断

有关所有 LPUART 中断请求的详细说明，请参见*表 197*。

表 203. LPUART 中断请求

中断事件	事件标志	使能控制位	中断清除方法	中断已激活	
				lpuart_it	lpuart_wkup
发送数据寄存器为空	TXE	TXEIE	TXE 在 TDR 中被写入数据时清零。	是	否
发送 FIFO 未满	TXFNF	TXFNFIE	TXFNF 在 TXFIFO 已满时清零。	是	否
发送 FIFO 为空	TXFE	TXFEIE	TXFE 在 TXFIFO 包含至少一个数据时清零，或通过将 TXFRQ 位置 1 来清零。	是	是
达到发送 FIFO 阈值	TXFT	TXFTIE	TXFT 在 TXFIFO 内容少于编程的阈值时由硬件清零	是	是
CTS 中断	CTSIF	CTSIE	CTSIF 由软件通过将 CTSCF 位置 1 来清零。	是	否
发送完成	TC	TCIE	TC 在 TDR 中被写入数据时清零，或通过将 TCCF 位置 1 来清零。	是	否
接收数据寄存器不为空（已准备好读取数据）	RXNE	RXNEIE	RXNE 通过读取 RDR 或通过将 RXFRQ 位置 1 来清零。	是	是
接收 FIFO 非空	RXFNE	RXFNEIE	RXFNE 在 RXFIFO 为空时清零，或通过将 RXFRQ 位置 1 来清零。	是	是
接收 FIFO 已满	RXFF(1)	RXFFIE	RXFF 在 RXFIFO 至少包含一个数据时清零。	是	是
达到接收 FIFO 阈值	RXFT	RXFTIE	RXFT 在 RXFIFO 内容少于编程的阈值时由硬件清零	是	是
检测到溢出错误	ORE	RXNEIE/RXFNEIE	ORE 通过将 ORECF 位置 1 来清零。	是	否

表 203. LPUART 中断请求（续）

中断事件	事件标志	使能控制位	中断清除方法	中断已激活	
				lpuart_it	lpuart_wkup
检测到空闲线路	IDLE	IDLEIE	IDLE 通过将 IDLECF 位置 1 来清零。	是	否
奇偶校验错误	PE	PEIE	PE 通过将 PECF 位置 1 来清零。	是	否
多缓冲区通信中的噪声标志、溢出错误和帧错误。	NE 或 ORE 或 FE	EIE	NE 通过将 NECF 位置 1 来清零。 ORE 通过将 ORECF 位置 1 来清零。 FE 标志通过将 FECF 位置 1 来清零	是	否
字符匹配	CMF	CMIE	CMF 通过将 CMCF 位置 1 来清零。	是	否
从低功耗模式唤醒	WUF	WUFIE	WUF 通过将 WUCF 位置 1 来清零。	是	是
达到发送 FIFO 阈值	TXFT	TXFTIE	TXFT 在 TXFIFO 内容少于编程的阈值时由硬件清零	是	是
达到接收 FIFO 阈值	RXFT	RXFTIE	RXFT 在 RXFIFO 内容少于编程的阈值时由硬件清零	是	是

1. 如果 LPUART 接收 n+1 个数据（n 表示 RXFIFO 大小，RXFIFO 中有 n 个数据，LPUART_RDR 中有 1 个数据），则 RXFF 标志置位。在停止模式下，未向 LPUART_RDR 提供时钟。因此，将不会对该寄存器进行写操作，一旦有 n 个数据被接收并写入到 RXFIFO，RXFF 中断将生效（RXFF 标志未置 1）。

34.5 LPUART寄存器

有关寄存器说明中使用的缩写的列表，请参见*第 58 页的第 1.2 节*。

34.5.1 控制寄存器 1 [备用] (LPUART_CR1)

Control register 1

偏移地址：0x00

复位值：0x0000 0000

同一寄存器可用于使能 FIFO 模式（本节）和禁止 FIFO 模式（下一节）的情况。

使能 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXFFIE	TXFEIE	FIFOEN	M1	Res.	Res.	DEAT[4:0]					DEDT[4:0]				
rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXFNFIE	TCIE	RXFNEIE	IDLEIE	TE	RE	UESM	UE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **RXFFIE**：RXFIFO 变满时中断使能 (RXFIFO full interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 RXFF=1 时，生成 LPUART 中断

位 30 **TXFEIE**：TXFIFO 为空时中断使能 (TXFIFO empty interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 TXFE=1 时，生成 LPUART 中断

位 29 **FIFOEN**：FIFO 模式使能 (FIFO mode enable)

此位由软件置 1 和清零。

0：禁止 FIFO 模式

1：使能 FIFO 模式

位 28 **M1**：字长 (Word length)

此位必须与位 12 (M0) 搭配使用来确定字长度。该位由软件置 1 或清零。

M[1:0] = “00”：1 个起始位，8 个数据位，n 个停止位

M[1:0] = “01”：1 个起始位，9 个数据位，n 个停止位

M[1:0] = “10”：1 个起始位，7 个数据位，n 个停止位

只有在 LPUART 没有使能 (UE=0) 时才能写入此位。

注： 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。

位 27:26 保留，必须保持复位值。

位 25:21 **DEAT[4:0]**：驱动器使能使能时间 (Driver Enable assertion time)

该 5 位值用于定义激活 DE（启动器使能）信号与起始位开始间的时间。以 lpuart_ker_ck 时钟周期数表示。有关详细信息，请参见*第 33.5.20 节：RS232 硬件流控制和 RS485 驱动器使能*。

只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 20:16 **DEDT[4:0]**：驱动器使能禁止时间 (Driver Enable deassertion time)

该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE（驱动器使能）信号间的时间。以 lpuart_ker_ck 时钟周期数表示。有关详细信息，请参见*第 34.3.13 节：RS232 硬件流控制和 RS485 驱动器使能*。

如果在 DEDT 时间内对 LPUART_TDR 寄存器执行写操作，则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。

只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 15 保留，必须保持复位值。

位 14 **CMIE**：字符匹配中断使能 (Character match interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：如果 LPUART_ISR 寄存器中的 CMF 位置 1，则生成 LPUART 中断

位 13 **MME**：静默模式使能 (Mute mode enable)

此位用于激活 LPUART 的静默模式功能，此位置 1 时，LPUART 可按 WAKE 位定义的方式在活动模式与静默模式之间切换。该位由软件置 1 和清零。

0：接收器永久处于活动模式

1：接收器可在静默模式和活动模式之间切换

位 12 **M0**：字长 (Word length)

此位与位 28 (M1) 搭配使用来确定字长度。它由软件置 1 或清零（请参见位 28 (M1) 的说明）。

只有在 LPUART 没有使能 (UE=0) 时才能写入此位。

位 11 **WAKE**：接收器唤醒方法 (Receiver wakeup method)

此位用于确定 LPUART 静默模式的唤醒方法。该位由软件置 1 或清零。

0：空闲线路

1：地址标记

只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 10 **PCE**：奇偶校验控制使能 (Parity control enable)

该位选择硬件奇偶校验控制（生成和检测）。使能奇偶校验控制时，计算出的奇偶校验位被插入到 MSB 位置（如果 M=1，则为第 9 位；如果 M=0，则为第 8 位），并对接收到的数据检查奇偶校验位。此位由软件置 1 和清零。一旦该位置 1，PCE 在当前字节的后面处于活动状态（在接收和发送时）。

0：禁止奇偶校验控制

1：使能奇偶校验控制

只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 9 **PS**：奇偶校验选择 (Parity selection)

该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。将在当前字节的后面选择奇偶校验。

0：偶校验

1：奇校验

只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 8 **PEIE**：PE 中断使能 (PE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 PE=1 时，生成 LPUART 中断

位 7 **TXFNFIE**：TXFIFO 未满中断使能 (TXFIFO not full interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 LPUART_ISR 寄存器中 TXE/TXFNF =1 时，生成 LPUART 中断

位 6 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 LPUART_ISR 寄存器中的 TC=1 时，生成 LPUART 中断

位 5 **RXFNEIE**：RXFIFO 非空中断使能 (RXFIFO not empty interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 LPUART_ISR 寄存器中的 ORE=1 或 RXNE/RXFNE=1 时，生成 LPUART 中断

位 4 **IDLEIE**：IDLE 中断使能 (IDLE interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 LPUART_ISR 寄存器中的 IDLE=1 时，生成 LPUART 中断

位 3 **TE**：发送器使能 (Transmitter enable)
该位使能发送器。该位由软件置 1 和清零。
0：禁止发送器
1：使能发送器
注： *传送期间 TE 位上的低电平脉冲（“0”后紧跟的是“1”）会在当前字的后面发送一个报头（空闲线路）。为生成空闲字符，TE 不能立即写入 1。为确保所需的持续时间，软件可轮询 LPUART_ISR 寄存器中的 TEACK 位。*
当 TE 置 1 时，在发送开始前存在 1 位的时间延迟。

位 2 **RE**：接收器使能 (Receiver enable)
该位使能接收器。该位由软件置 1 和清零。
0：禁止接收器
1：使能接收器并开始搜索起始位

位 1 **UESM**：停止模式下的 LPUART 使能 (LPUART enable in Stop mode)
当此位清零时，LPUART 无法将 MCU 从低功耗模式唤醒。
当此位置 1 时，LPUART 能够将 MCU 从低功耗模式唤醒，前提是 LPUART 时钟选择为 HSI 或 LSE（在 RCC 中）。
此位由软件置 1 和清零。
0：LPUART 无法将 MCU 从低功耗模式唤醒。
1：LPUART 能够将 MCU 从低功耗模式唤醒。当该功能激活时，LPUART 的时钟源必须是 HSI 或 LSE（请参见 RCC 一章）。
注： *建议在进入低功耗模式前将 UESM 位置 1，并在退出低功耗模式时将其清零。*

位 0 **UE**：LPUART 使能 (LPUART enable)
此位清零后，LPUART 预分频器和输出将立即停止，并丢弃当前操作。LPUART 的配置保留，但 LPUART_ISR 中的所有状态标志将被复位。此位由软件置 1 和清零。
0：禁止 LPUART 预分频器和输出，低功耗模式
1：使能 LPUART
注： *为进入低功耗模式而不在线路上生成错误，之前必须复位 TE 位，并且软件必须等待 LPUART_ISR 寄存器中的 TC 位置 1 后才能复位 UE 位。*
UE = 0 时也会复位 DMA 请求，因必须在复位 UE 位前禁止 DMA 通道。

34.5.2 控制寄存器 1 [备用] (LPUART_CR1)

Control register 1

偏移地址：0x00

复位值：0x0000 0000

同一寄存器可用于使能 FIFO 模式（上一节）和禁止 FIFO 模式（本节）的情况。

禁止 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	FIFO EN	M1	Res.	Res.	DEAT[4:0]					DEDT[4:0]				
		rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29 **FIFOEN**：FIFO 模式使能 (FIFO mode enable)
此位由软件置 1 和清零。
0：禁止 FIFO 模式。
1：使能 FIFO 模式。

位 28 **M1**：字长 (Word length)
此位必须与位 12 (M0) 搭配使用来确定字长度。该位由软件置 1 或清零。
M[1:0] = “00”： 1 个起始位，8 个数据位，n 个停止位
M[1:0] = “01”： 1 个起始位，9 个数据位，n 个停止位
M[1:0] = “10”： 1 个起始位，7 个数据位，n 个停止位
只有在 LPUART 没有使能 (UE=0) 时才能写入此位。
注： 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。

位 27:26 保留，必须保持复位值。

位 25:21 **DEAT[4:0]**：驱动器使能使能时间 (Driver Enable assertion time)
该 5 位值用于定义激活 DE（启动器使能）信号与起始位开始间的时间。以 lpuart_ker_ck 时钟周期数表示。有关详细信息，请参见*第 33.5.20 节：RS232 硬件流控制和 RS485 驱动器使能*。
只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 20:16 **DEDT[4:0]**：驱动器使能禁止时间 (Driver Enable deassertion time)
该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE（驱动器使能）信号间的时间。以 lpuart_ker_ck 时钟周期数表示。有关详细信息，请参见*第 34.3.13 节：RS232 硬件流控制和 RS485 驱动器使能*。
如果在 DEDT 时间内对 LPUART_TDR 寄存器执行写操作，则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。
只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 15 保留，必须保持复位值。

位 14 **CMIE**：字符匹配中断使能 (Character match interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：如果 LPUART_ISR 寄存器中的 CMF 位置 1，则生成 LPUART 中断

位 13 **MME**：静默模式使能 (Mute mode enable)

此位用于激活 LPUART 的静默模式功能，此位置 1 时，LPUART 可按 WAKE 位定义的方式在活动模式与静默模式之间切换。该位由软件置 1 和清零。

0：接收器永久处于活动模式

1：接收器可在静默模式和活动模式之间切换

位 12 **M0**：字长 (Word length)

此位与位 28 (M1) 搭配使用来确定字长度。它由软件置 1 或清零（请参见位 28 (M1) 的说明）。

只有在 LPUART 没有使能 (UE=0) 时才能写入此位。

位 11 **WAKE**：接收器唤醒方法 (Receiver wakeup method)

此位用于确定 LPUART 静默模式的唤醒方法。该位由软件置 1 或清零。

0：空闲线路

1：地址标记

只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 10 **PCE**：奇偶校验控制使能 (Parity control enable)

该位选择硬件奇偶校验控制（生成和检测）。只有在 LPUART 没有使能时，使能奇偶校验控制时，计算出的奇偶校验位被插入到 MSB 位置（如果 M=1，则为第 9 位；如果 M=0，则为第 8 位），并对接收到的数据检查奇偶校验位。此位由软件置 1 和清零。一旦该位置 1，PCE 在当前字节的后面处于活动状态（在接收和发送时）。

0：禁止奇偶校验控制

1：使能奇偶校验控制

只有在 LPUART 没有使能 (UE=0) 时才能写入此位。

位 9 **PS**：奇偶校验选择 (Parity selection)

该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。将在当前字节的后面选择奇偶校验。

0：偶校验

1：奇校验

只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 8 **PEIE**：PE 中断使能 (PE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 PE=1 时，生成 LPUART 中断

位 7 **TXEIE**：发送数据寄存器为空 (Transmit data register empty)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中 TXE/TXFNF =1 时，生成 LPUART 中断

位 6 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 TC=1 时，生成 LPUART 中断

位 5 **RXNEIE**：接收数据寄存器非空 (Receive data register not empty)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 ORE=1 或 RXNE/RXFNE=1 时，生成 LPUART 中断

位 4 **IDLEIE**：IDLE 中断使能 (IDLE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 IDLE=1 时，生成 LPUART 中断

位 3 **TE**：发送器使能 (Transmitter enable)

该位使能发送器。该位由软件置 1 和清零。

0：禁止发送器

1：使能发送器

注： 传送期间 TE 位上的低电平脉冲（“0”后紧跟的是“1”）会在当前字的后面发送一个报头（空闲线路）。为生成空闲字符，TE 不能立即写入 1。为确保所需的持续时间，软件可轮询 LPUART_ISR 寄存器中的 TEACK 位。

当 TE 置 1 时，在发送开始前存在 1 位的时间延迟。

位 2 **RE**：接收器使能 (Receiver enable)

该位使能接收器。该位由软件置 1 和清零。

0：禁止接收器

1：使能接收器并开始搜索起始位

位 1 **UESM**：停止模式下的 LPUART 使能 (LPUART enable in Stop mode)

当此位清零时，LPUART 无法将 MCU 从低功耗模式唤醒。

当此位置 1 时，LPUART 能够将 MCU 从低功耗模式唤醒，前提是 LPUART 时钟选择为 HSI 或 LSE（在 RCC 中）。

此位由软件置 1 和清零。

0：LPUART 无法将 MCU 从低功耗模式唤醒。

1：LPUART 能够将 MCU 从低功耗模式唤醒。当该功能激活时，LPUART 的时钟源必须是 HSI 或 LSE（请参见 RCC 一章）。

注： 建议在进入低功耗模式前将 UESM 位置 1，并在退出低功耗模式时将其清零。

位 0 **UE**：LPUART 使能 (LPUART enable)

此位清零后，LPUART 预分频器和输出将立即停止，并丢弃当前操作。LPUART 的配置保留，但 LPUART_ISR 中的所有状态标志将被复位。此位由软件置 1 和清零。

0：禁止 LPUART 预分频器和输出，低功耗模式

1：使能 LPUART

注： 为进入低功耗模式而不在线路上生成错误，之前必须复位 TE 位，并且软件必须等待 LPUART_ISR 寄存器中的 TC 位置 1 后才能复位 UE 位。

UE = 0 时也会复位 DMA 请求，因必须在复位 UE 位前禁止 DMA 通道。

34.5.3 控制寄存器 2 (LPUART_CR2)

Control register 2

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
ADD[7:0]								Res.	Res.	Res.	Res.	MSBFIRST	DATAINV	TXINV	RXINV
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
SWAP	Res.	STOP[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDM7	Res.	Res.	Res.	Res.
rw		rw	rw								rw				

位 31:24 **ADD[7:0]**：LPUART 节点的地址 (Address of the LPUART node)

ADD[7:4]：

这些位用于指定 LPUART 节点的地址或要识别的字符代码。

它们在多处理器通信时于静默模式或停止模式下使用，以通过 7 位地址标记检测唤醒 MCU。发送器发送字符的 MSB 应为 1。它们还可用于正常接收和静默模式无效时的字符检测（例如，ModBus 协议中的块结束检测）。这种情况下，接收到的整个字符（8 位）将与 ADD[7:0] 值进行比较，如果匹配，CMF 标志将置 1。

仅在禁止接收 (RE = 0) 或禁止 LPUART (UE=0) 时才能写入这些位。

ADD[3:0]：

这些位用于指定 LPUART 节点的地址或要识别的字符代码。

它们在多处理器通信时于静默模式或低功耗模式下使用，以通过地址标记检测进行唤醒。

仅在禁止接收 (RE = 0) 或禁止 LPUART (UE=0) 时才能写入这些位。

位 23:20 保留，必须保持复位值。

位 19 **MSBFIRST**：最高有效位在前 (Most significant bit first)

此位由软件置 1 和清零。

0：发送/接收数据时位 0 在前，后跟起始位。

1：发送/接收数据时 MSB（位 7/8）在前，后跟起始位。

只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 18 **DATAINV**：二进制数据反向 (Binary data inversion)

此位由软件置 1 和清零。

0：按正/正向逻辑发送/接收数据寄存器中的逻辑数据。（1=H，0=L）。

1：按负/反向逻辑发送/接收数据寄存器中的逻辑数据。（1=L，0=H）。奇偶校验位也取反。

只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 17 **TXINV**：TX 引脚有效电平反向 (TX pin active level inversion)

此位由软件置 1 和清零。

0：TX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作。

1：对 TX 引脚信号值取反。（V_{DD} = 0/标记，Gnd=1/空闲）。

允许在 TX 线路上使用外部反相器。

只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 16 **RXINV**：RX 引脚有效电平反向 (RX pin active level inversion)

此位由软件置 1 和清零。

0：RX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作。

1：对 RX 引脚信号值取反。（V_{DD} = 0/标记，Gnd=1/空闲）。

允许在 RX 线路上使用外部反相器。

只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 15 **SWAP**：交换 TX/RX 引脚 (Swap TX/RX pins)

此位由软件置 1 和清零。

0：按标准引脚排列定义使用 TX/RX 引脚。

1：交换 TX 和 RX 引脚功能。允许在与另一个 UART 的交叉连接时工作。

只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 14 保留，必须保持复位值。

位 13:12 **STOP[1:0]**：停止位 (STOP bits)

这些位用于编程停止位。

00：1 个停止位

01：保留

10：2 个停止位

11：保留

只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。

位 11:5 保留，必须保持复位值。

位 4 **ADDM7**：7 位地址检测/4 位地址检测 (7-bit Address Detection/4-bit Address Detection)
此位用于选择 4 位地址检测或 7 位地址检测。
0：4 位地址检测
1：7 位地址检测（在 8 位数据模式下）
只有在 LPUART 没有使能 (UE=0) 时才能写入该位
注： 在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。

位 3:0 保留，必须保持复位值。

34.5.4 控制寄存器 3 (LPUART_CR3)

Control register 3

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXFTCFG[2:0]			RXFTIE	RXFTCFG[2:0]			Res.	TXFTIE	WUFIE	WUS[1:0]		Res.	Res.	Res.	Res.
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
DEP	DEM	DDRE	OVRDIS	Res.	CTSIE	CTSE	RTSE	DMAT	DMAR	Res.	Res.	HDSEL	Res.	Res.	EIE
rw	rw	rw	rw		rw	rw	rw	rw	rw			rw			rw

位 31:29 **TXFTCFG[2:0]**：TXFIFO 阈值配置 (TXFIFO threshold configuration)
000：TXFIFO 达到其深度的 1/8。
001：TXFIFO 达到其深度的 1/4。
110：TXFIFO 达到其深度的 1/2。
011：TXFIFO 达到其深度的 3/4。
100：TXFIFO 达到其深度的 7/8。
101：TXFIFO 变空。
其余组合：保留。

位 28 **RXFTIE**：RXFIFO 阈值中断使能 (RXFIFO threshold interrupt enable)
此位由软件置 1 和清零。
0：禁止中断。
1：当接收 FIFO 达到 RXFTCFG 中编程的阈值时，生成 LPUART 中断。

位 27:25 **RXFTCFG[2:0]**：接收 FIFO 阈值配置 (Receive FIFO threshold configuration)
000：接收 FIFO 达到其深度的 1/8。
001：接收 FIFO 达到其深度的 1/4。
110：接收 FIFO 达到其深度的 1/2。
011：接收 FIFO 达到其深度的 3/4。
100：接收 FIFO 达到其深度的 7/8。
101：接收 FIFO 已满。
其余组合：保留。

位 24 保留，必须保持复位值。

位 23 **TXFTIE**：TXFIFO 阈值中断使能 (TXFIFO threshold interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 TXFIFO 达到 TXFTCFG 中编程的阈值时，生成 LPUART 中断

位 22 **WUFIE**：从低功耗模式唤醒中断使能 (Wakeup from low-power mode interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 LPUART_ISR 寄存器中的 WUF=1 时，生成 LPUART 中断
注： *WUFIE 必须在进入低功耗模式前置 1。*
如果 LPUART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 33.4 节：USART 实现。

位 21:20 **WUS[1:0]**：从低功耗模式唤醒中断标志选择 (Wakeup from low-power mode interrupt flag selection)
该位域用于指定激活 WUF（从低功耗模式唤醒标志）的事件。
00：WUF 在地址匹配时激活（按 ADD[7:0] 和 ADDM7 所定义）
01：保留
10：WUF 在起始位检测时激活
11：WUF 在 RXNE 时激活
只有在 LPUART 没有使能 (UE=0) 时才能写入此位域。
注： *如果 LPUART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 33.4 节：USART 实现。*

位 19:16 保留，必须保持复位值。

位 15 **DEP**：驱动器使能极性选择 (Driver enable polarity selection)
0：DE 信号高电平有效。
1：DE 信号低电平有效。
只有在 LPUART 没有使能 (UE=0) 时才能写入此位。

位 14 **DEM**：驱动器使能模式 (Driver enable mode)
此位用于通过 DE 信号激活外部收发器控制。
0：禁止 DE 功能。
1：使能 DE 功能。DE 信号在 RTS 引脚上输出。
只有在 LPUART 没有使能 (UE=0) 时才能写入此位。

位 13 **DDRE**：接收出错时的 DMA 禁止 (DMA Disable on Reception Error)
0：接收出错时不禁止 DMA。相应的错误标志置 1，但 RXNE 保持为 0 以防止上溢。因此，将不使能 DMA 请求，从而不会传送错误数据（无 DMA 请求），但会传送接收到的下一个正确数据。
1：接收出错后禁止 DMA。相应的错误标志以及 RXNE 均置 1。屏蔽 DMA 请求，直到错误标志清零。这意味着软件必须首先禁止 DMA 请求 (DMAR = 0) 或者将 RXNE 清零，然后再将错误标志清零。
只有在 LPUART 没有使能 (UE=0) 时才能写入此位。
注： *接收错误包括：奇偶校验错误、帧错误或噪声错误。*

位 12 **OVRDIS**：上溢禁止 (Overrun Disable)
此位用于禁止接收上溢检测。
0：接收新数据前未读取已接收的数据时，上溢错误标志 ORE 置 1。
1：禁止上溢功能。如果在 RXNE 标志仍置 1 时接收到新数据。
则 ORE 标志不会置 1，且新接收的数据会覆盖 LPUART_RDR 寄存器之前的内容。
只有在 LPUART 没有使能 (UE=0) 时才能写入此位。
注： *此控制位用于检查通信流而不会读取数据。*

位 11 保留，必须保持复位值。

位 10 **CTSIE**：CTS 中断使能 (CTS interrupt enable)
0：禁止中断
1：当 LPUART_ISR 寄存器中 CTSIF = 1 时，生成中断

位 9 **CTSE**：CTS 使能 (CTS enable)
0：禁止 CTS 硬件流控制
1：使能 CTS 模式，仅当 nCTS 输入有效（连接到 0）时才发送数据。如果在发送数据时使 nCTS 输入无效，会在停止之前完成发送。如果使 nCTS 有效时数据已写入数据寄存器，则将延迟发送，直到 nCTS 有效。
只有在 LPUART 没有使能 (UE=0) 时才能写入该位。

位 8 **RTSE**：RTS 使能 (RTS enable)
0：禁止 RTS 硬件流控制
1：使能 RTS 输出，仅当接收缓冲区中有空间时才会请求数据。发送完当前字符后应停止发送数据。可以接收数据时使 nRTS 输出有效（拉至 0）。
只有在 LPUART 没有使能 (UE=0) 时才能写入此位。

位 7 **DMAT**：DMA 使能发送器 (DMA enable transmitter)
该位由软件置 1/复位。
1：针对发送使能 DMA 模式
0：针对发送禁止 DMA 模式

位 6 **DMAR**：DMA 使能接收器 (DMA enable receiver)
该位由软件置 1/复位。
1：针对接收使能 DMA 模式
0：针对接收禁止 DMA 模式

位 5:4 保留，必须保持复位值。

位 3 **HDSEL**：半双工选择 (Half-duplex selection)
选择单线半双工模式
0：未选择半双工模式
1：选择半双工模式
只有在 LPUART 没有使能 (UE=0) 时才能写入此位。

位 2:1 保留，必须保持复位值。

位 0 **EIE**：错误中断使能 (Error interrupt enable)
如果发生帧错误、上溢错误或出现噪声标志（LPUART_ISR 寄存器中 FE = 1 或 ORE = 1 或 NE = 1），则需要使用错误中断使能位来使能中断生成。
0：禁止中断
LPUART_ISR 寄存器中的 FE=1 或 ORE=1 或 NE=1 时生成中断

34.5.5 波特率寄存器 (LPUART_BRR)

Baud rate register

只有在 LPUART 没有使能 (UE=0) 时才能写入此寄存器。在自动波特率检测模式下，该位由硬件自动更新。

偏移地址：0x0C

复位值：0x0000 0000

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

位 31:20 保留，必须保持复位值。

位 19:0 **BRR[19:0]**：LPUART 波特率 (LPUART baud rate)

注： *禁止在 LPUART_BRR 寄存器中写入小于 0x300 的值。*

如果 LPUART_BRR 必须 ≧ 0x300 且 LPUART_BRR 为 20 位，则使用高 fck 值生成高波特率时应十分谨慎。fck 必须在 [3 x 波特率到 4096 x 波特率] 的范围内。

34.5.6 请求寄存器 (LPUART_RQR)

Request register

偏移地址：0x18

复位值：0x0000 0000

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

位 31:5 保留，必须保持复位值。

位 4 **TXFRQ**：发送数据刷新请求 (Transmit data flush request)

该位在 FIFO 模式使能时使用。TXFRQ 位置 1 以清空整个 FIFO。这会将标志 TXFE（TXFIFO 为空，LPUART_ISR 寄存器中的位 23）置 1。

注： *在 FIFO 模式下，TXFNF 标志在清空请求期间复位，直到 TxFIFO 为空，以确保数据寄存器中没有写入数据。*

位 3 **RXFRQ**：接收数据刷新请求 (Receive data flush request)

向该位写入 1 时会将 RXNE 标志清零。

这可以丢弃接收的数据而不对其执行读取操作，并避免发生上溢。

位 2 **MMRQ**：静默模式请求 (Mute mode request)

向此位写入 1 可将 LPUART 置于静默模式，并将 RWU 标志复位。

位 1 **SBKRQ**：发送中断请求 (Send break request)

向此位写入 1 可将 SBKF 标志置 1 并在发送设备可用后立即请求在线路上发送 BREAK。

注： *如果应用需要在之前插入的所有数据（包括尚未发送的数据）后发送中断字符，软件应等到 TXE 标志使能后将 SBKRQ 位置 1。*

位 0 保留，必须保持复位值。

34.5.7 中断和状态寄存器 [备用] (LPUART_ISR)

Interrupt and status register

偏移地址：0x1C

复位值：0x0080 00C0

同一寄存器可用于使能 FIFO 模式（本节）和禁止 FIFO 模式（下一节）的情况。

使能 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXFT	RXFT	Res.	RXFF	TXFE	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY
				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
Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	TXFNF	TC	RXFNE	IDLE	ORE	NE	FE	PE
					r	r		r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27 **TXFT**：TXFIFO 阈值标志 (TXFIFO threshold flag)

当 TXFIFO 达到在 LPUART_CR3 寄存器的 TXFTCFG 中编程的阈值时（即 TXFIFO 包含 TXFTCFG 个空位置），该位由硬件置 1。如果 LPUART_CR3 寄存器中的 TXFTIE 位 =1（位 31），则会生成中断。

0：TXFIFO 未达到编程的阈值。

1：TXFIFO 已达到编程的阈值。

位 26 **RXFT**：RXFIFO 阈值标志 (RXFIFO threshold flag)

当 RXFIFO 达到在 LPUART_CR3 寄存器的 RXFTCFG 中编程的阈值时（即接收 FIFO 包含 RXFTCFG 个数据），该位由硬件置 1。如果 LPUART_CR3 寄存器中的 RXFTIE 位 =1（位 27），则会生成中断。

0：接收 FIFO 未达到编程的阈值。

1：接收 FIFO 已达到编程的阈值。

位 25 保留，必须保持复位值。

位 24 **RXFF**：RXFIFO 已满 (RXFIFO Full)

当接收到的数据量对应于 RXFIFO 大小 + 1（RXFIFO 已满 + LPUART_RDR 寄存器中的 1 个数据）时，该位由硬件置 1。

如果 LPUART_CR1 寄存器中 RXFFIE 位 = 1，则会生成中断。

0：RXFIFO 未满

1：RXFIFO 已满

位 23 **TXFE**：TXFIFO 为空 (TXFIFO empty)

当 TXFIFO 为空时，该位由硬件置 1。当 TXFIFO 包含至少一个数据时，该标志被清零。也可以通过向 LPUART_RQR 寄存器中的位 TXFRQ（位 4）写入 1 将 TXFE 标志置 1。

如果 LPUART_CR1 寄存器中的 TXFEIE 位 =1（位 30），则会生成中断。

0：TXFIFO 非空

1：TXFIFO 为空

位 22 **REACK**：接收使能确认标志 (Receive enable acknowledge flag)

LPUART 采用接收使能值时，通过硬件将此位置 1/复位。

此位可用于验证 LPUART 是否准备好在进入低功耗模式前接收数据。

注： 如果 LPUART 不支持从停止模式唤醒功能，该位保留且保持复位值。

位 21 **TEACK**：发送使能确认标志 (Transmit enable acknowledge flag)

LPUART 采用发送使能值时，通过硬件将此位置 1/复位。

通过写入 TE=0 生成空闲帧请求，然后在 LPUART_CR1 寄存器中写入 TE=1 以遵循 TE=0 最短周期时，可使用此位。

位 20 **WUF**：从低功耗模式唤醒标志 (Wakeup from low-power mode flag)

当检测到唤醒事件时，此位由硬件置 1。事件通过 WUS 位域定义。通过向 LPUART_ICR 寄存器中的 WUCF 写入 1，此位由软件清零。

如果 LPUART_CR3 寄存器中 WUFIE=1，则会生成中断。

注： 当 UESM 清零时，WUF 标志也清零。

如果 LPUART 不支持从停止模式唤醒功能，该位保留且保持复位值。

位 19 **RWU**：*接收器从静默模式唤醒 (Receiver wakeup from Mute mode)*

此位指示 LPUART 是否处于静默模式。当识别出唤醒/静默序列时，此位由硬件清零/置 1。静默模式控制序列（地址或 IDLE）通过 LPUART_CR1 寄存器中的 WAKE 位选择。

当选择 IDLE 模式下唤醒时，该位只能通过用软件向 LPUART_RQR 寄存器中的 MMRQ 位写 1 的方式置 1。

0：接收器处于工作模式

1：接收器处于静默模式

注： 如果 LPUART 不支持从停止模式唤醒功能，该位保留且保持复位值。

位 18 **SBKF**：发送中断标志 (Send break flag)

此位指示已请求发送中断字符。通过将 1 写入 LPUART_CR3 寄存器中的 SBKRQ 位，此位由软件置 1。此位在中断发送的停止位期间由硬件自动复位。

0：不发送中断字符

1：将发送中断字符

位 17 **CMF**：字符匹配标志 (Character match flag)

接收到由 ADD[7:0] 定义的字符后由硬件将此位置 1。通过向 LPUART_ICR 寄存器中的 CMCF 写入 1，此位由软件清零。

如果 LPUART_CR1 寄存器中 CMIE=1，则会生成中断。

0：未检测到字符匹配

1：检测到字符匹配

位 16 **BUSY**：忙标志 (Busy flag)

此位由硬件置 1 和复位。当 RX 线路上正在进行通信（成功检测到起始位）时有效。在接收结束（成功或失败）时复位。

0：LPUART 处于空闲状态（无接收）

1：正在接收

位 15:11 保留，必须保持复位值。

位 10 **CTS**：CTS 标志 (CTS flag)

此位由硬件置 1/复位。此位是对 nCTS 输入引脚的状态取反。

0：nCTS 线置 1

1：nCTS 线复位

注： 如果不支持硬件流控制功能，该位保留且保持复位值。

位 9 **CTSIF**：CTS 中断标志 (CTS interrupt flag)

如果 CTSE 位置 1，当 nCTS 输入切换时，此位由硬件置 1。通过将 1 写入 LPUART_ICR 寄存器中的 CTSCF 位，此位由软件清零。

如果 LPUART_CR3 寄存器中 CTSIE=1，则会生成中断。

0：nCTS 状态线上未发生变化

1：nCTS 状态线上发生变化

注： 如果不支持硬件流控制功能，该位保留且保持复位值。

位 8 保留，必须保持复位值。

位 7 **TXFNF**：TXFIFO 未满 (TXFIFO not full)

TXFNF 在 TXFIFO 未满时由硬件置 1，并因此可向 LPUART_TDR 中写入数据。每次对 LPUART_TDR 进行写操作都会将数据置于 TXFIFO 中。该标志保持置 1，直到 TXFIFO 已满。当 TXFIFO 已满时，该标志清零，表示不能向 LPUART_TDR 写入数据。

在清空请求期间，TXFNF 保持复位，直到 TXFIFO 为空。发送清空请求（通过将 TXFRQ 位置 1）后，应先检查 TXFNF 标志，然后再写入 TXFIFO（TXFNF 和 TXFE 将同时置 1）。

如果 LPUART_CR1 寄存器中 TXFNFIE 位 = 1，则会生成中断。

0：数据寄存器已满/发送 FIFO 已满。

1：数据寄存器/发送 FIFO 未满。

注： 单缓冲区发送期间使用该位。

位 6 **TC**：发送完成 (Transmission complete)

如果已完成对包含数据的帧的发送并且 TXFF 置 1，则该位由硬件置 1。如果 LPUART_CR1 寄存器中 TCIE = 1，则会生成中断。通过向 LPUART_ICR 寄存器中的 TCCF 写入 1 或向 LPUART_TDR 寄存器执行写操作，此位由软件清零。

如果 LPUART_CR1 寄存器中 TCIE = 1，则会生成中断。

0：传送未完成

1：传送已完成

注： 如果 TE 位复位且无任何发送正在进行，TC 位会立即置 1。

位 5 **RXFNE**：RXFIFO 非空 (RXFIFO not empty)

RXFIFO 非空时，RXFNE 位由硬件置 1，并因此可以从 LPUART_RDR 寄存器读取数据。每次对 LPUART_RDR 进行读操作都会在 RXFIFO 中释放一个位置。它在 RXFIFO 为空时清零。

也可以通过向 LPUART_RQR 寄存器中的 RXFRQ 位写入 1 将 RXFNE 标志位清零。

如果 LPUART_CR1 寄存器中 RXFNEIE=1，则会生成中断。

0：未接收到数据

1：已准备好读取接收到的数据

位 4 **IDLE**：检测到空闲线路 (IDLE line detected)

检测到空闲线路时，该位由硬件置 1。如果 LPUART_CR1 寄存器中 IDLEIE=1，则会生成中断。通过向 LPUART_ICR 寄存器中的 IDLECF 写入 1，此位由软件清零。

0：未检测到空闲线路

1：检测到空闲线路

注： 直到 RXFNE 位已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。

使能静默模式 (MME=1) 后，如果 LPUART 未静默 (RWU=0)，则 IDLE 置 1，无论是否通过 WAKE 位选择了静默模式。如果 RWU=1，IDLE 不置 1。

位 3 **ORE**：溢出错误 (Overrun error)

在 RXFF = 1 的情况下，当移位寄存器中当前正在接收的数据准备好传输到 LPUART_RDR 寄存器时，该位由硬件置 1。通过向 LPUART_ICR 寄存器中的 ORECF 写入 1，此位由软件清零。

如果 LPUART_CR1 寄存器中 RXFNEIE=1 或 EIE = 1，则会生成中断。

0：无溢出错误

1：检测到溢出错误

注：　当此位置 1 时，LPUART_RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。EIE 位置 1 后，如果在多缓冲区通信中 ORE 标志置 1，则会生成中断。

LPUART_CR3 寄存器中的 OVRDIS 位置 1 时，此位将被永久强制清零（无上溢检测）。

位 2 **NE**：起始位噪声检测标志 (Start bit noise detection flag)

当在接收的帧的起始位上检测到噪声时，此位由硬件置 1。此位由软件清零，方法是向 LPUART_ICR 寄存器中的 NECF 位写入 1。

0：未检测到噪声

1：检测到噪声

注：　该位不会生成中断，因为该位出现的时间与本身生成中断的 RXFNE 位出现的时间相同。EIE 位置 1 后，如果在多缓冲区通信中 NE 标志置 1，则会生成中断。

此错误与 LPUART_RDR 中的字符相关联。

位 1 **FE**：帧错误 (Framing error)

当检测到去同步化、过度的噪声或中断字符时，该位由硬件置 1。通过向 LPUART_ICR 寄存器中的 FECF 写入 1，此位由软件清零。

在智能卡模式下发送数据时，如果在达到最大发送尝试次数后仍未成功（智能卡向数据帧发送 NACK 信号），则此位置 1。

如果 LPUART_CR1 寄存器中 EIE = 1，则会生成中断。

0：未检测到帧错误

1：检测到帧错误或中断字符

注：　此错误与 LPUART_RDR 中的字符相关联。

位 0 **PE**：奇偶校验错误 (Parity error)

当在接收器模式下发生奇偶校验错误时，该位由硬件置 1。通过向 LPUART_ICR 寄存器中的 PECF 写入 1，此位由软件清零。

如果 LPUART_CR1 寄存器中 PEIE = 1，则会生成中断。

0：无奇偶校验错误

1：奇偶校验错误

注：　此错误与 LPUART_RDR 中的字符相关联。

34.5.8 中断和状态寄存器 [备用] (LPUART_ISR)

Interrupt and status register

偏移地址：0x1C

复位值：0x0000 00C0

同一寄存器可用于使能 FIFO 模式（上一节）和禁止 FIFO 模式（本节）的情况。

禁止 FIFO 模式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
					r	r		r	r	r	r	r	r	r	r

位 31:23 保留，必须保持复位值。

位 22 **REACK**：接收使能确认标志 (Receive enable acknowledge flag)

LPUART 采用接收使能值时，通过硬件将此位置 1/复位。

此位可用于验证 LPUART 是否准备好在进入低功耗模式前接收数据。

注： 如果 LPUART 不支持从停止模式唤醒功能，该位保留且保持复位值。

位 21 **TEACK**：发送使能确认标志 (Transmit enable acknowledge flag)

LPUART 采用发送使能值时，通过硬件将此位置 1/复位。

通过写入 TE=0 生成空闲帧请求，然后在 LPUART_CR1 寄存器中写入 TE=1 以遵循 TE=0 最短周期时，可使用此位。

位 20 **WUF**：从低功耗模式唤醒标志 (Wakeup from low-power mode flag)

当检测到唤醒事件时，此位由硬件置 1。事件通过 WUS 位域定义。通过向 LPUART_ICR 寄存器中的 WUCF 写入 1，此位由软件清零。

如果 LPUART_CR3 寄存器中 WUFIE=1，则会生成中断。

注： 当 UESM 清零时，WUF 标志也清零。

如果 LPUART 不支持从停止模式唤醒功能，该位保留且保持复位值。

位 19 **RWU**：*接收器从静默模式唤醒 (Receiver wakeup from Mute mode)*

此位指示 LPUART 是否处于静默模式。当识别出唤醒/静默序列时，此位由硬件清零/置 1。静默模式控制序列（地址或 IDLE）通过 LPUART_CR1 寄存器中的 WAKE 位选择。

当选择 IDLE 模式下唤醒时，该位只能通过用软件向 LPUART_RQR 寄存器中的 MMRQ 位写 1 的方式置 1。

0：接收器处于工作模式

1：接收器处于静默模式

注： 如果 LPUART 不支持从停止模式唤醒功能，该位保留且保持复位值。

位 18 **SBKF**：发送中断标志 (Send break flag)

此位指示已请求发送中断字符。通过将 1 写入 LPUART_CR3 寄存器中的 SBKRQ 位，此位由软件置 1。此位在中断发送的停止位期间由硬件自动复位。

0：不发送中断字符

1：将发送中断字符

位 17 **CMF**：字符匹配标志 (Character match flag)

接收到由 ADD[7:0] 定义的字符后由硬件将此位置 1。通过向 LPUART_ICR 寄存器中的 CMCF 写入 1，此位由软件清零。

如果 LPUART_CR1 寄存器中 CMIE=1，则会生成中断。

0：未检测到字符匹配

1：检测到字符匹配

位 16 **BUSY**：忙标志 (Busy flag)

此位由硬件置 1 和复位。当 RX 线路上正在进行通信（成功检测到起始位）时有效。在接收结束（成功或失败）时复位。

0：LPUART 处于空闲状态（无接收）

1：正在接收

位 15:11 保留，必须保持复位值。

位 10 **CTS**：CTS 标志 (CTS flag)

此位由硬件置 1/复位。此位是对 nCTS 输入引脚的状态取反。

0：nCTS 线置 1

1：nCTS 线复位

注： 如果不支持硬件流控制功能，该位保留且保持复位值。

位 9 **CTSIF**：CTS 中断标志 (CTS interrupt flag)

如果 CTSE 位置 1，当 nCTS 输入切换时，此位由硬件置 1。通过将 1 写入 LPUART_ICR 寄存器中的 CTSCF 位，此位由软件清零。

如果 LPUART_CR3 寄存器中 CTSIE=1，则会生成中断。

0：nCTS 状态线上未发生变化

1：nCTS 状态线上发生变化

注： 如果不支持硬件流控制功能，该位保留且保持复位值。

位 8 保留，必须保持复位值。

位 7 **TXE**：发送数据寄存器为空/TXFIFO 未满 (Transmit data register empty/TXFIFO not full)

当 LPUART_TDR 寄存器的内容已传输到移位寄存器时，TXE 由硬件置 1。通过对 LPUART_TDR 寄存器执行写入操作将该位清零。

如果 LPUART_CR1 寄存器中 TXEIE 位 = 1，则会生成中断。

0：数据寄存器已满

1：数据寄存器未满

注： 单缓冲区发送期间使用该位。

位 6 **TC**：发送完成 (Transmission complete)

如果已完成对包含数据的帧的发送并且 TXE 置 1，则此位由硬件置 1。如果 LPUART_CR1 寄存器中 TCIE = 1，则会生成中断。通过向 LPUART_ICR 寄存器中的 TCCF 写入 1 或向 LPUART_TDR 寄存器执行写操作，此位由软件清零。

如果 LPUART_CR1 寄存器中 TCIE = 1，则会生成中断。

0：传送未完成

1：传送已完成

注： 如果 TE 位复位且无任何发送正在进行，TC 位会立即置 1。

位 5 **RXNE**：读取数据寄存器不为空 (Read data register not empty)

当 LPUART_RDR 移位寄存器的内容已传输到 LPUART_RDR 寄存器时，RXNE 位由硬件置 1。通过对 LPUART_RDR 寄存器执行读取操作将该位清零。也可以通过将 LPUART_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE 标志位清零。

如果 LPUART_CR1 寄存器中 RXNEIE = 1，则会生成中断。

0：未接收到数据

1：已准备好读取接收到的数据

位 4 **IDLE**：检测到空闲线路 (IDLE line detected)

检测到空闲线路时，该位由硬件置 1。如果 LPUART_CR1 寄存器中 IDLEIE=1，则会生成中断。通过向 LPUART_ICR 寄存器中的 IDLECF 写入 1，此位由软件清零。

0：未检测到空闲线路

1：检测到空闲线路

注： *直到 RXNE 位已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。*

使能静默模式 (MME=1) 后，如果 LPUART 未静默 (RWU=0)，则 IDLE 置 1，无论是否通过 WAKE 位选择了静默模式。如果 RWU=1，IDLE 不置 1。

位 3 **ORE**：溢出错误 (Overrun error)

在 RXFF = 1 的情况下，当移位寄存器中当前正在接收的数据准备好传输到 LPUART_RDR 寄存器时，该位由硬件置 1。通过向 LPUART_ICR 寄存器中的 ORECF 写入 1，此位由软件清零。

如果 LPUART_CR1 寄存器中 RXNEIE=1 或 EIE = 1，则会生成中断。

0：无溢出错误

1：检测到溢出错误

注： *当此位置 1 时，LPUART_RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。EIE 位置 1 后，如果在多缓冲区通信中 ORE 标志置 1，则会生成中断。*

LPUART_CR3 寄存器中的 OVRDIS 位置 1 时，此位将被永久强制清零（无上溢检测）。

位 2 **NE**：起始位噪声检测标志 (Start bit noise detection flag)

当在接收的帧的起始位上检测到噪声时，此位由硬件置 1。此位由软件清零，方法是向 LPUART_ICR 寄存器中的 NECF 位写入 1。

0：未检测到噪声

1：检测到噪声

注： *该位不会生成中断，因为该位出现的时间与本身生成中断的 RXNE 位出现的时间相同。EIE 位置 1 后，如果在多缓冲区通信中 NE 标志置 1，则会生成中断。*

位 1 **FE**：帧错误 (Framing error)

当检测到去同步化、过度的噪声或中断字符时，该位由硬件置 1。通过向 LPUART_ICR 寄存器中的 FECF 写入 1，此位由软件清零。

在智能卡模式下发送数据时，如果在达到最大发送尝试次数后仍未成功（智能卡向数据帧发送 NACK 信号），则此位置 1。

如果 LPUART_CR1 寄存器中 EIE = 1，则会生成中断。

0：未检测到帧错误

1：检测到帧错误或中断字符

位 0 **PE**：奇偶校验错误 (Parity error)

当在接收器模式下发生奇偶校验错误时，该位由硬件置 1。通过向 LPUART_ICR 寄存器中的 PECF 写入 1，此位由软件清零。

如果 LPUART_CR1 寄存器中 PEIE = 1，则会生成中断。

0：无奇偶校验错误

1：奇偶校验错误

34.5.9 中断标志清零寄存器 (LPUART_ICR)

Interrupt flag clear register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.
											w			w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CTSCF	Res.	Res.	TCCF	Res.	IDLECF	ORECF	NECF	FECF	PECF
						w			w		w	w	w	w	w

位 31:21 保留，必须保持复位值。

位 20 **WUCF**：从低功耗模式唤醒清零标志 (Wakeup from low-power mode clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 WUF 标志将清零。

注： 如果 LPUART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 33.4 节：USART 实现。

位 19:18 保留，必须保持复位值。

位 17 **CMCF**：字符匹配清零标志 (Character match clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 CMF 标志将清零。

位 16:10 保留，必须保持复位值。

位 9 **CTSCF**：CTS 清零标志 (CTS clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 CTSIF 标志将清零。

位 8:7 保留，必须保持复位值。

位 6 **TCCF**：发送完成清零标志 (Transmission complete clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 TC 标志将清零。

位 5 保留，必须保持复位值。

位 4 **IDLECF**：检测到空闲线路清零标志 (Idle line detected clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 IDLE 标志将清零。

位 3 **ORECF**：上溢错误清零标志 (Overrun error clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 ORE 标志将清零。

位 2 **NECF**：检测到噪声清零标志 (Noise detected clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 NE 标志将清零。

位 1 **FECF**：帧错误清零标志 (Framing error clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 FE 标志将清零。

位 0 **PECF**：奇偶校验错误清零标志 (Parity error clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 PE 标志将清零。

34.5.10 接收数据寄存器 (LPUART_RDR)

Receive data register

偏移地址：0x24

复位值：0x0000 0000

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

位 31:9 保留，必须保持复位值。

位 8:0 **RDR[8:0]**：接收数据值 (Receive data value)

包含接收到的数据字符。

RDR 寄存器在输入移位寄存器和内部总线之间提供了并行接口（请参见*图 349*）。

在使能奇偶校验位的情况下进行接收时，从 MSB 位中读取的值为接收到的奇偶校验位。

34.5.11 发送数据寄存器 (LPUART_TDR)

Transmit data register

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

位 31:9 保留，必须保持复位值。

位 8:0 **TDR[8:0]**：发送数据值 (Transmit data value)

包含要发送的数据字符。

TDR 寄存器在内部总线和输出移位寄存器之间提供了并行接口（请参见*图 349*）。

在使能奇偶校验位的情况下（LPUART_CR1 寄存器中的 PCE 位被置 1）进行发送时，由于 MSB 的写入值（位 7 或位 8，具体取决于数据长度）会被奇偶校验位所取代，因此该值不起任何作用。

注：　只能在 TXE/TXFNF=1 时写入此寄存器。

34.5.12 预分频器寄存器 (LPUART_PRESC)

Prescaler register

只有在 LPUART 没有使能 (UE=0) 时才能写入此寄存器。

偏移地址：0x2C

复位值：0x0000 0000

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

位 31:4 保留，必须保持复位值。

位 3:0 **PRESCALER[3:0]**：时钟预分频器 (Clock prescaler)

LPUART 输入时钟可通过预分频器进行分频：

0000：输入时钟未分频
0001：输入时钟 2 分频
0010：输入时钟 4 分频
0011：输入时钟 6 分频
0100：输入时钟 8 分频
0101：输入时钟 10 分频
0110：输入时钟 12 分频
0111：输入时钟 16 分频
1000：输入时钟 32 分频
1001：输入时钟 64 分频
1010：输入时钟 128 分频
1011：输入时钟 256 分频
其余组合：保留。

注： 为 PRESCALER 编程不允许的值时，编程的预分频值将为«1011»，即输入时钟除以 256。

34.5.13 LPUART 寄存器映射

下表提供了 LPUART 寄存器映射和复位值。

表 204. LPUART 寄存器映射和复位值

偏移	寄存器名称	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
0x00	LPUART_CR1 FIFO mode enabled	RXFFIE	TXFEIE	FIFOEN	M1	Res.	Res.	DEAT[4:0]					DEDT[4:0]					Res.	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXFNFIE	TCIE	RXFNEIE	IDLEIE	TE	RE	UESM	UE
	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
0x00	LPUART_CR1 FIFO mode disabled	Res.	Res.	FIFOEN	M1	Res.	Res.	DEAT[4:0]					DEDT[4:0]					Res.	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
	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
0x04	LPUART_CR2	ADD[7:0]								Res.	Res.	Res.	Res.	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	Res.	STOP[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDM7	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0					0	0	0	0	0		0	0								0				
0x08	LPUART_CR3	TXFTCFG[2:0]			RXFTIE	RXFTCFG[2:0]			Res.	TXFTIE	WUFIE	WUS[1:0]		Res.	Res.	Res.	Res.	DEP	DEM	DDRE	OVRDIS	Res.	CTSIE	CTSE	RTSE	DMAT	DMAR	Res.	Res.	HDSEL	Res.	Res.	EIE
	Reset value	0	0	0	0	0	0	0		0	0	0	0					0	0	0	0		0	0	0	0	0			0			0
0x0C	LPUART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[19:0]																			
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10-0x14	Reserved																																
0x18	LPUART_RQR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	Res.
	Reset value																												0	0	0	0	
0x1C	LPUART_ISR FIFO mode enabled	Res.	Res.	Res.	Res.	TXFT	RXFT	Res.	RXFF	TXFF	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	TXFNF	TC	RXFNE	IDLE	ORE	NE	FE	PE
	Reset value					0	0		0	1	0	0	0	0	0	0	0						0	0		1	0	0	0	0	0	0	0
0x1C	LPUART_ISR FIFO mode disabled	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
	Reset value										0	0	0	0	0	0	0						0	0		1	0	0	0	0	0	0	0

表 204. LPUART 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x20	LPUART_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTSCF	Res.	Res.	TCCF	Res.	IDLECF	ORECF	NECF	FECF	PECF
	Reset value												0			0								0			0		0	0	0	0	0
0x24	LPUART_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x28	LPUART_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x2C	LPUART_PRESC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]			
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 2.2 节：存储器构成*。

35 串行外设接口 (SPI)

35.1 简介

SPI 接口可用于使用 SPI 协议或与外部器件进行通信。SPI 模式可通过软件进行选择。器件复位后默认选择 SPI Motorola 模式。

串行外设接口 (SPI) 协议支持与外部器件进行半双工、全双工和单工同步串行通信。该接口可配置为主模式，在这种情况下，它可为外部从器件提供通信时钟 (SCK)。该接口还能够在多主模式配置下工作。

35.2 SPI 主要特性

- 主模式或从模式操作
- 基于三条线的全双工同步传输
- 基于双线的半双工同步传输，其中一条可作为双向数据线
- 基于双线的单工同步传输，其中一条可作为单向数据线
- 4 到 16 位数据位宽大小选择
- 多主模式功能
- 8 个主模式波特率预分频器，可达 $f_{PCLK}/2$
- 从模式频率可达 $f_{PCLK}/2$
- 对于主模式和从模式都可通过硬件或软件进行 NSS 管理：动态切换主/从操作
- 可编程的时钟极性和相位
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 可触发中断的专用发送和接收标志
- SPI 总线忙状态标志
- 支持 SPI Motorola 模式
- 用于确保可靠通信的硬件 CRC 功能：
  - 在发送模式下可将 CRC 值作为最后一个字节发送
  - 根据收到的最后一个字节自动进行 CRC 错误校验
- 可触发中断的主模式故障和上溢标志
- CRC 错误标志
- 具有 DMA 功能的两个 32 位内置 Rx 和 Tx FIFO
- 增强型 TI 和 NSS 脉冲模式支持

35.3 SPI 实现

下表列出了器件内置的所有 SPI 外设的特性。

表 205. STM32WB55xx SPI 实现

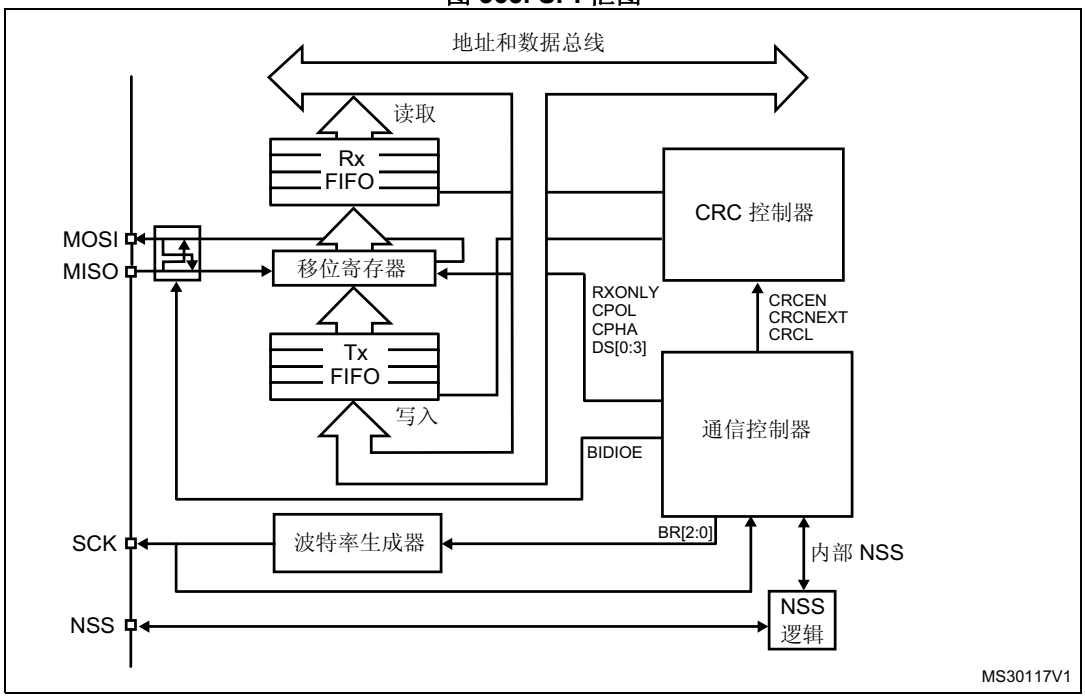
SPI 特性	SPI1	SPI2
增强型 NSSP 和 TI 模式	是	是
I^2S 支持	否	否
硬件 CRC 计算	是	是
数据位宽大小可配置	4 到 16 位	4 到 16 位
Rx/Tx FIFO 大小	32 位	32 位
从低功耗睡眠模式唤醒	是	是

35.4 SPI 功能说明

35.4.1 概述

SPI 支持在 MCU 与外部器件之间进行同步串行通信。应用软件可通过轮询状态标志或使用专用 SPI 中断对通信进行管理。SPI 的主要组件及其交互方式如下面的*图 363* 框图所示。

图 363. SPI 框图



四个 I/O 引脚专用于与外部器件进行 SPI 通信。

- **MISO**：主输入 / 从输出数据。通常情况下，此引脚用于在从模式下发送数据和在主模式下接收数据。
- **MOSI**：主输出 / 从输入数据。通常情况下，此引脚用于在主模式下发送数据和在从模式下接收数据。
- **SCK**：SPI 主器件的串行时钟输出引脚以及 SPI 从器件的串行时钟输入引脚。
- **NSS**：从器件选择引脚。根据 SPI 和 NSS 设置，该引脚可用于：
  - 选择特定的从器件以进行通信
  - 同步数据帧或
  - 检测多个主器件之间是否存在冲突

  详细信息，请参见*第 35.4.5 节：从器件选择 (NSS) 引脚管理*。

SPI 总线支持一个主器件与一个或多个从器件之间进行通信。该总线至少由两条线构成——一条用于时钟信号，另一条用于同步数据传输。其他信号可以根据 SPI 节点间的数据交换及其从器件选择信号管理进行添加。

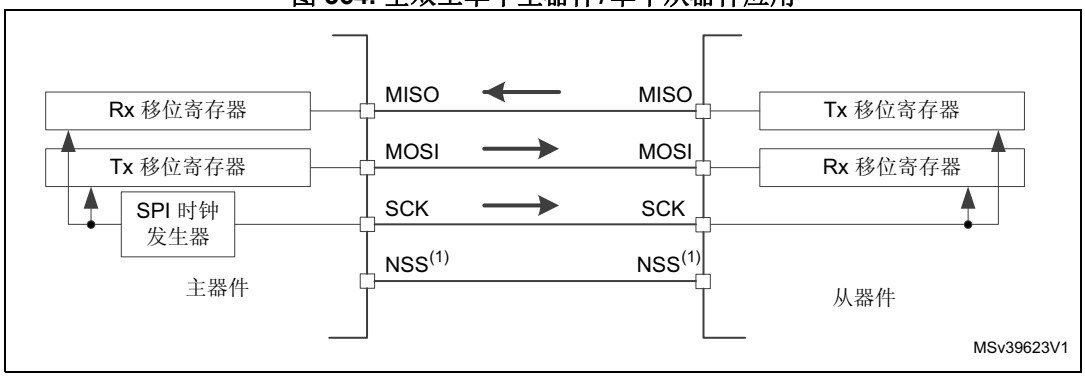
35.4.2 一个主器件和一个从器件之间的通信

根据目标器件和应用要求，SPI 支持 MCU 使用不同的配置与之通信。这些配置使用 2 条或 3 条线（通过软件 NSS 管理），也可以使用 3 条或 4 条线（通过硬件 NSS 管理）。通信始终由主器件发起。

全双工通信

默认情况下，SPI 配置为进行全双工通信。在这种配置下，主器件和从器件的移位寄存器通过 MOSI 和 MISO 引脚之间的两条单向线连接。在 SPI 通信过程中，数据随主器件提供的 SCK 时钟边沿同步移位。主器件通过 MOSI 线将待发送的数据发送给从器件，通过 MISO 线从从器件接收数据。当数据帧传输完成时（所有位均移出），主器件和从器件之间即完成信息交换。

图 364. 全双工单个主器件 / 单个从器件应用

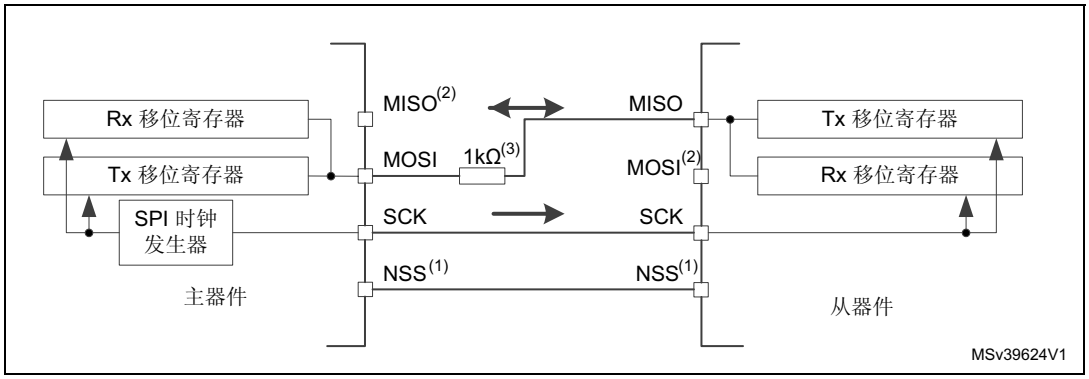


1. NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些引脚。之后，必须在内部为主器件和从器件处理硬件控制流。有关更多详细信息，请参见*第 35.4.5 节：从器件选择 (NSS) 引脚管理*。

半双工通信

通过将 SPIx_CR1 寄存器的 BIDIMODE 位置 1，SPI 可采用半双工模式进行通信。在这种配置下，使用一条交叉连接线将主器件和从器件的移位寄存器连接起来。在此通信过程中，数据随 SCK 时钟边沿在移位寄存器之间进行移位，传输方向由主器件和从器件通过各自 SPIx_CR1 寄存器中的 BDIOE 位进行选择。在这种配置下，主器件的 MISO 引脚和从器件的 MOSI 引脚未被使用，这 2 个引脚可作通用 GPIO 使用。

图 365. 半双工单个主器件/单个从器件应用



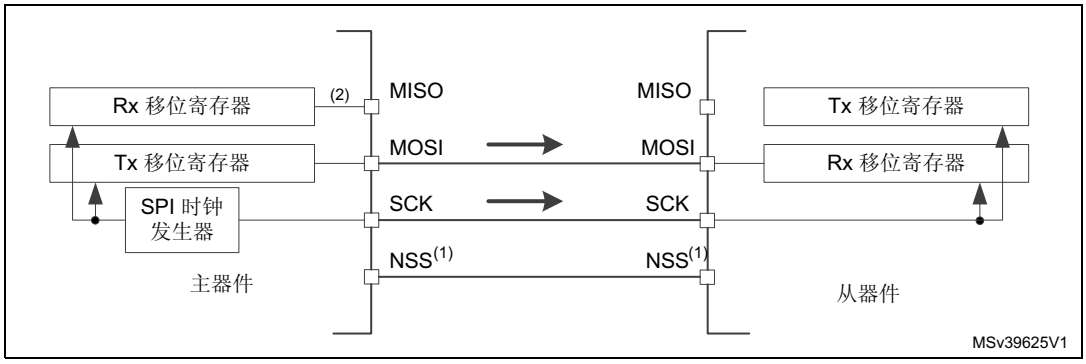
1. NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些引脚。之后，必须在内部为主器件和从器件处理硬件控制流。有关更多详细信息，请参见*第 35.4.5 节：从器件选择 (NSS) 引脚管理*。
2. 在这种配置下，主器件的 MISO 引脚和从器件的 MOSI 引脚可用作 GPIO。
3. 当以双向模式工作的两个节点间的通信方向不是同步变化时，会出现临界情况，新发送器访问共用数据线，而前一个发送器仍保持线路上的相反值（值取决于 SPI 配置和通信数据）。两个节点会出现冲突，在共用线上短暂提供相反的输出电平，直到下一个节点也相应地改变其方向设置。建议此模式下在 MISO 和 MOSI 引脚之间插入串行电阻以在这种情况下保护输出并限制电流在二者之间流过。

单工通信

通过 SPIx_CR2 寄存器中的 RXONLY 位将 SPI 设置为只发送模式或只接收模式，可使 SPI 以单工模式进行通信。在这种配置下，仅使用一条线在主器件和从器件的移位寄存器之间进行传输。通信中未被使用到的一对 MISO 和 MOSI 引脚，可用作标准 GPIO。

- **只发送模式 (RXONLY=0)**：配置设置与全双工设置相同。应用必须忽略在未使用的输入引脚上捕获的信息。该引脚可以用作标准 GPIO。
- **只接收模式 (RXONLY=1)**：应用可通过将 RXONLY 位置 1 来禁止 SPI 输出功能。在从器件配置下，MISO 输出被禁止，该引脚可用作 GPIO。当从器件选择信号有效时，从器件继续从 MOSI 引脚接收数据（请参见 *35.4.5：从器件选择 (NSS) 引脚管理*）。基于数据缓冲区的配置产生接收数据事件。在主器件配置下，MOSI 输出被禁止，该引脚可用作 GPIO。只要 SPI 处于使能状态，便不断生成时钟信号。停止时钟的唯一方式是将 RXONLY 位或 SPE 位清零，直至来自 MISO 引脚的传入模式结束，然后基于相应配置填充数据缓冲区结构。

图 366. 单工单个主器件/单个从器件应用（主器件为只发送模式/从器件为只接收模式）



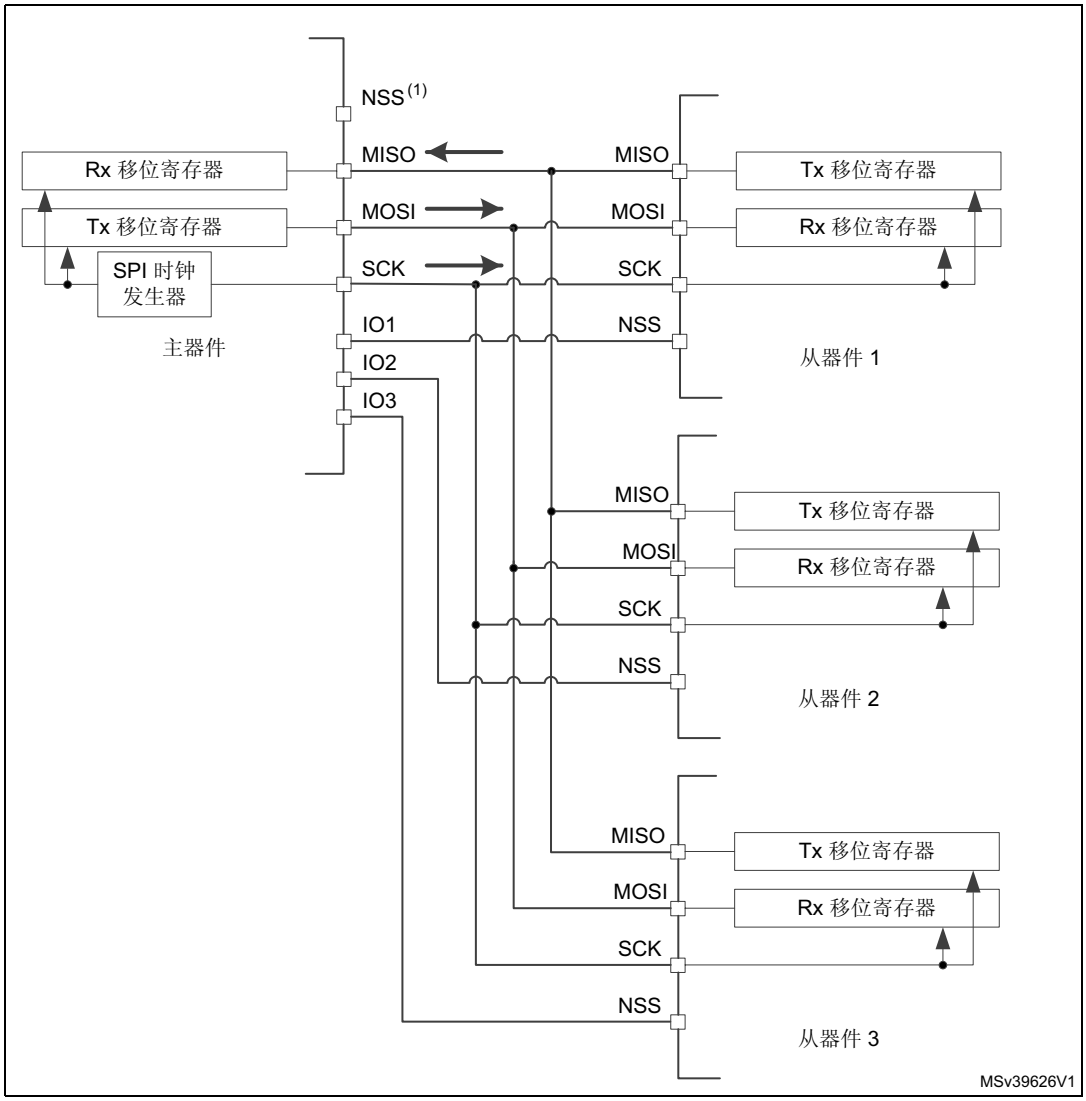
1. NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可配置外设不使用 NSS 引脚，由主机和从机内部自动处理。请参见*第 35.4.5 节：从器件选择 (NSS) 引脚管理*。
2. 当 Rx 移位寄存器输入端发生异常输入时，在标准发送模式下所有发送、接收相关联的事件都必须忽略（例如 OVF 标志位）。
3. 在这种配置下，MASTER 和 SLAVE 二端的 MISO 引脚均可用作 GPIO。

注： *任何单工通信都可以通过把半双工模式中的方向设置固定来实现（使能双向模式，同时 BDIO 位保持不变）。*

35.4.3 标准多从器件通信

在具有两个或多个独立从器件的配置下，主器件使用 GPIO 引脚来管理每个从器件的片选线（请参见*图 367*）。主器件必须通过拉低与从器件 NSS 输入相连的 GPIO 的电平来单独选择一个从器件。执行该操作后，便建立了标准主器件与专用从器件之间的通信。

图 367. 主器件和三个独立的从器件



1. 此配置的主器件侧不使用 NSS 引脚。该引脚必须在内部管理 (SSM = 1，SSI = 1) 以避免任何 MODF 错误。
2. 由于从器件的 MISO 引脚连在一起，所有从器件 MISO 引脚的 GPIO 配置必须设置为复用功能开漏（请参见*第 9.3.7 节：I/O 复用功能输入/输出*）。

35.4.4 多主通信

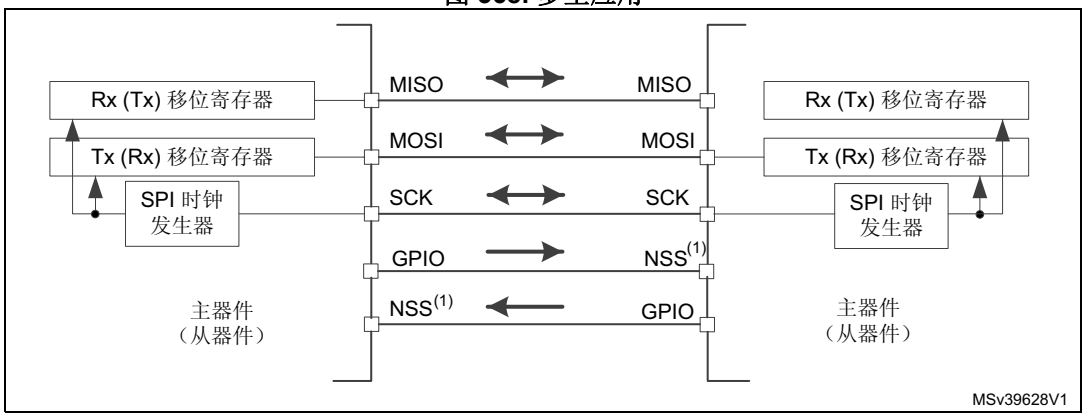
如果 SPI 总线未用于多主功能，用户可使用内置功能来检测尝试同时控制总线的两个节点间是否存在潜在冲突。对于该检测，NSS 引脚配置为硬件输入模式。

由于此时只有一个结点可将其输出施加到公用数据线上，因此无法连接超过两个以此模式工作的 SPI 节点。

当节点无效时，默认情况下均保持从模式。一旦一个节点要接管对总线的控制，它会将自身切换到主模式，然后通过专用 GPIO 引脚向其他节点的从器件选择输入施加有效电平。会话完成后，有效的从器件选择信号将被释放，控制总线的节点会短暂切换回被动从模式，等待下一个会话开始。

如果两个节点同时发出各自的控制请求，则会出现总线冲突（请参见模式故障 MODF 事件）。随后，用户可应用某个简单的仲裁过程（例如，在两个节点上施加不同的预定义超时来推迟下一个尝试）。

图 368. 多主应用



1. 在两个节点上，NSS 引脚配置为硬件输入模式。当无效节点配置为从器件时，其有效电平将使能 MISO 线输出控制。

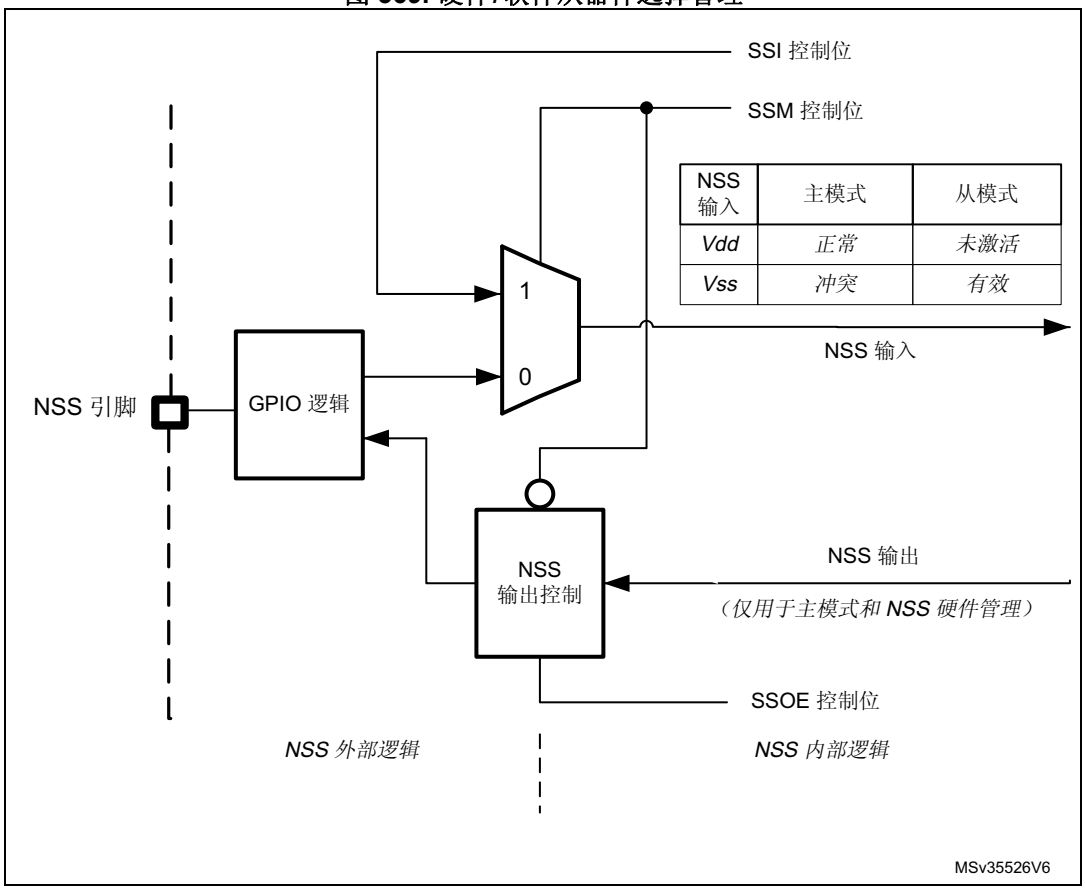
35.4.5 从器件选择 (NSS) 引脚管理

在从模式下，NSS 用作标准的“片选”输入，使从器件与主器件进行通信。在主模式下，NSS 可用作输出或输入。用作输入时，可防止多主模式总线冲突；用作输出时，可驱动单个从器件的从器件选择信号。

可以使用 SPIx_CR1 寄存器中的 SSM 位设置硬件或软件从器件选择管理：

- **软件 NSS 管理 (SSM = 1)**：在这种配置下，由 SPIx_CR1 寄存器中的 SSI 位的值内部驱动从器件选择信息。外部 NSS 引脚空闲，可供其他应用使用。
- **硬件 NSS 管理 (SSM = 0)**：在这种情况下，可行的配置有两种：所用配置取决于 NSS 输出配置（SPIx_CR1 寄存器中的 SSOE 位）。
  - **NSS 输出使能（SSM=0 且 SSOE = 1）**：仅在将 MCU 设置为主器件时才使用该配置。NSS 引脚由硬件管理。只要在主模式下使能 SPI (SPE=1)，NSS 信号便会被驱动为低电平，并且会一直保持低电平状态，直至禁止 SPI (SPE =0)。如果激活 NSS 脉冲模式 (NSSP=1)，连续通信间会生成脉冲。SPI 无法在采用此 NSS 设置的多主模式配置下工作。
  - **NSS 输出禁止（SSM=0 且 SSOE = 0）**：如果微控制器在总线上用作主器件，此配置可实现多主模式功能。如果在该模式下将 NSS 引脚拉至低电平，SPI 将进入主模式故障状态，器件将在从模式下自动进行重新配置。在从模式下，NSS 引脚用作标准的“片选”输入，当 NSS 线为低电平时将选择从器件。

图 369. 硬件/软件从器件选择管理



35.4.6 通信格式

SPI 通信过程中，将同时执行接收和发送操作。串行时钟 (SCK) 对数据线上的信息的移位和采样进行同步。通信格式取决于时钟相位、时钟极性和数据帧格式。为了能够在彼此间进行通信，主器件和从器件必须遵循相同的通信格式。

时钟相位和极性控制

通过 SPIx_CR1 寄存器中的 CPOL 和 CPHA 位，可以用软件选择四种可能的时序关系。CPOL（时钟极性）位控制不传输任何数据时时钟的空闲状态值。此位对主器件和从器件都有作用。如果复位 CPOL，SCK 引脚在空闲状态处于低电平。如果将 CPOL 置 1，SCK 引脚在空闲状态处于高电平。

如果将 CPHA 位置 1，则会在 SCK 引脚的第二个边沿捕获传输的第一个数据位（如果复位 CPOL 位，则为下降沿；如果将 CPOL 位置 1，则为上升沿）。即，在每次出现该时钟边沿时锁存数据。如果将 CPHA 位复位，则会在 SCK 引脚的第一个边沿捕获传输的第一个数据位（如果将 CPOL 位置 1，则为下降沿；如果将 CPOL 位复位，则为上升沿）。即，在每次出现该时钟边沿时锁存数据。

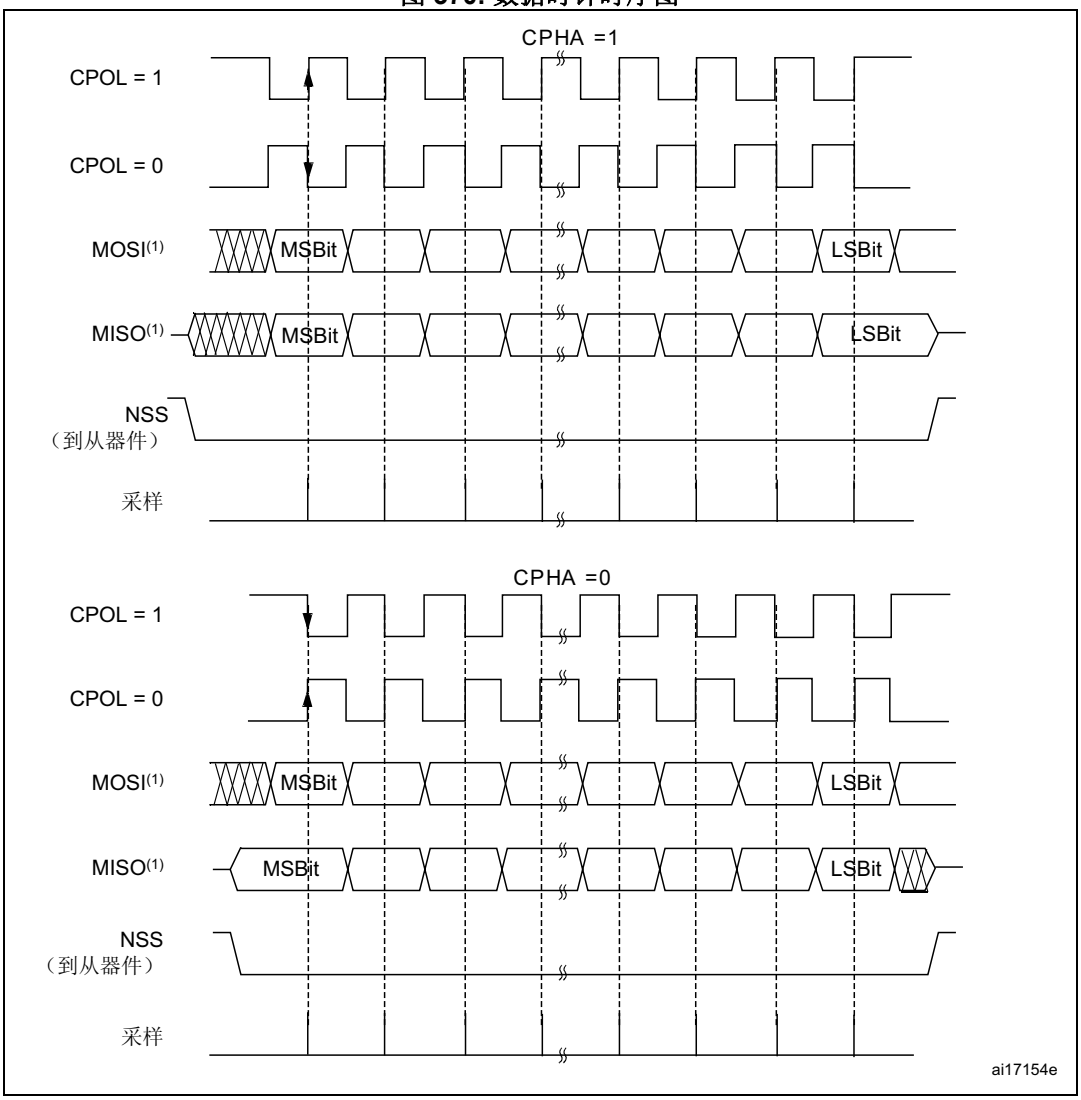
CPOL（时钟极性）和 CPHA（时钟相位）位的组合用于选择数据捕获时钟边沿。

图 370 给出了在 CPHA 和 CPOL 位的四种组合下的 SPI 全双工传输。

注： *在切换 CPOL/CPHA 位之前，必须通过复位 SPE 位来禁止 SPI。*

SCK 的空闲状态必须与 SPIx_CR1 寄存器中选择的极性相对应（如果 CPOL=1，则上拉 SCK；如果 CPOL=0，则下拉 SCK）。

图 370. 数据时钟时序图

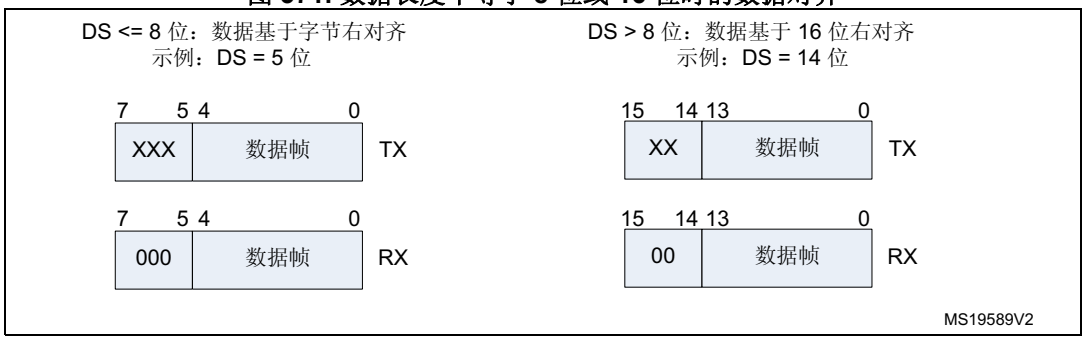


1. 数据位的顺序取决于 LSBFIRST 位的设置。

数据帧格式

SPI 移位寄存器可设置为以 MSB 在前或 LSB 在前的方式移出数据，具体取决于 LSBFIRST 位的值。数据帧的长度通过 DS 位进行选择。数据帧的长度可设置为 4 位到 16 位，且此设置对于发送和接收均适用。无论所选的数据帧长度为何，必须依照 FRXTH 电平对 FIFO 进行读访问。访问 SPIx_DR 寄存器时，数据帧始终按字节（数据不超过一个字节时）或半字进行右对齐（请参见*图 371*）。在通信过程中，只会为数据帧内的位提供时钟并传输这些位。

图 371. 数据长度不等于 8 位或 16 位时的数据对齐



注： *数据长度至少为 4 位。如果所选的数据长度不足 4 位，则会将数据帧长度强制为 8 位。*

35.4.7 SPI 配置

主器件和从器件的配置步骤几乎相同。对于具体的模式设置，请遵从相应部分的内容。若要对标准通信进行初始化，请执行以下步骤：

1. 对适当的 GPIO 寄存器执行写操作：将 MOSI、MISO 和 SCK 引脚配置为 GPIO。
2. 对 SPI_CR1 寄存器执行写操作：
   a) 通过 BR[2:0] 位配置串行时钟波特率（注：4）。
   b) 配置 CPOL 和 CPHA 位的组合，定义数据传输和串行时钟之间的关系（四种关系中的一种）（NSSP 模式下必须将 CPHA 清零）。（注：2——在 TI 模式下使能 CRC 的情况除外）。
   c) 通过配置 RXONLY 或 BIDIMODE 和 BIDIOE 来选择单工或半双工模式（RXONLY 和 BIDIMODE 不可同时置 1）。
   d) 配置 LSBFIRST 位以定义帧格式（注：2）。
   e) 如果需要 CRC，请配置 CRCL 和 CRCEN 位（SCK 时钟信号处于空闲状态时）。
   f) 配置 SSM 和 SSI（注：2 和 3）。
   g) 配置 MSTR 位（在多主模式 NSS 配置下，如果主器件配置为防止发生 MODF 错误，则应避免 NSS 上出现状态冲突）。
3. 对 SPI_CR2 寄存器执行写操作：
   a) 配置 DS[3:0] 位，选择传输的数据长度。
   b) 配置 SSOE（注：1、2 和 3）。
   c) 如果需要使用 TI 协议，请将 FRF 位置 1（TI 模式下将 NSSP 位保持清零状态）。
   d) 如果在两个数据单元之间需要 NSS 脉冲模式，请将 NSSP 位置 1（NSSP 模式下将 CHPA 和 TI 位保持清零状态）。
   e) 配置 FRXTH 位。RXFIFO 阈值必须与 SPIx_DR 寄存器的读访问大小相符。
   f) 如果封装模式下使用 DMA，请初始化 LDMA_TX 和 LDMA_RX 位。
4. 对 SPI_CRCPR 寄存器执行写操作：需要时配置 CRC 多项式。
5. 对相应的 DMA 寄存器执行写操作：如果使用 DMA 数据流，请在 DMA 寄存器中配置 SPI Tx 和 Rx 专用的 DMA 数据流。

注： *(1) 从模式下无需此步骤。*

(2) TI 模式下无需此步骤。

(3) NSSP 模式下无需此步骤。

(4) 从模式下无需此步骤，但从器件在 TI 模式下工作时除外。

35.4.8 使能 SPI 的步骤

建议在主器件发送时钟前使能 SPI 从器件。否则，数据传输可能会不正常。从器件的数据寄存器必须包含待发送的数据才能开始与主器件通信（在通信时钟的第一个边沿；如果时钟信号连续，则是在正在进行的通信结束前）。使能 SPI 从器件前，SCK 信号必须稳定为所选极性对应的空闲状态电平。

当 SPI 处于使能状态，且 TXFIFO 不为空或者对 TXFIFO 执行下一次写操作时，全双工模式（或任何只发送模式）下的主器件开始通信。

在任何主器件只接收模式（RXONLY=1 或 BIDIMODE=1 且 BIDIOE=0）下，使能 SPI 后，主器件立即开始通信且时钟立即开始运行。

要处理 DMA，请遵从相应部分的内容。

35.4.9 数据发送和接收过程

RXFIFO 和 TXFIFO

所有 SPI 数据交互均经由 32 位内置 FIFO。这使得 SPI 能够以连续流工作，并能防止在数据帧长度较短时发生上溢。每个方向都有其自身的 FIFO，称为 TXFIFO 和 RXFIFO。这些 FIFO 可在所有 SPI 模式下使用，但已使能 CRC 计算的只接收模式（从器件或主器件）除外（请参见*第 35.4.14 节：CRC 计算*）。

FIFO 的处理取决于数据交换模式（双工和单工）、数据帧格式（帧中的位数）、FIFO 数据寄存器中的访问大小（8 位或 16 位）以及访问 FIFO 时是否使用数据封装（请参见*第 35.4.13 节：TI 模式*）。

对 SPIx_DR 寄存器执行读访问时，会返回尚未读取的 RXFIFO 中存储的最早的值。对 SPIx_DR 执行写访问时，会在发送队列结束时将写入的数据存储到 TXFIFO 中。读访问必须始终符合由 SPIx_CR2 寄存器中的 FRXTH 位配置的 RXFIFO 阈值。FTLVL[1:0] 和 FRLVL[1:0] 位用于指示两个 FIFO 当前的占用水平。

SPIx_DR 寄存器的读访问必须通过 RXNE 事件进行管理。当数据存储到 RXFIFO 中且达到阈值（由 FRXTH 位定义）时会触发该事件。当 RXNE 清零时，RXFIFO 被视为空。同样地，待发送数据帧的写访问通过 TXE 事件进行管理。当 TXFIFO 占用水平小于或等于其容量的一半时会触发该事件。否则，TXE 清零，TXFIFO 被视为已满。通过这种方式，当数据帧格式不超过 8 位时，RXFIFO 最多可存储四个数据帧，而 TXFIFO 最多只能存储三个数据帧。当软件尝试向 TXFIFO 中写入更多 16 位模式的数据时，这种差异能够防止 TXFIFO 中已存储的 3 个 8 位数据帧出现损坏的情况。TXE 和 RXNE 事件可通过中断来轮询或处理。请参见*图 373* 到*图 376*。

另一种管理数据交换的方式是使用 DMA（请参见*使用 DMA（直接存储器寻址）进行通信*）。

如果在 RXFIFO 已满时收到下一个数据，将发生上溢事件（请参见*第 35.4.10 节：SPI 状态标志*中的 OVR 标志说明）。上溢事件可通过中断来轮询或处理。

BSY 位被置 1 表示正在处理当前数据帧交互。当时钟信号连续运行时，在主器件中，BSY 标志在数据帧之间保持置 1 状态，但在从器件中，BSY 标志在数据帧传输之间变为低电平并持续最短的一段时间（一个 SPI 时钟）。

序列处理

可通过一个序列传送一些数据帧从而完成一条消息。使能发送后，序列即开始，只要主器件的 TXFIFO 中存在数据便一直继续。时钟信号由主器件持续提供，直至 TXFIFO 变为空，之后时钟信号停止，等待其他数据。

在只接收模式、半双工模式（BIDIMODE=1 且 BIDIOE=0）或单工模式（BIDIMODE=0 且 RXONLY=1）下，当使能 SPI 并激活只接收模式后，主器件将立即启动序列。时钟信号由主器件提供，且直至主器件禁止 SPI 或只接收模式才会停止。在此之前，主器件会连续接收数据帧。

当主器件能够以连续模式（SCK 信号连续）提供所有交互时，任何时候都必须根据从器件功能来处理数据流及其内容。必要时，主器件必须降低通信速度，提供较慢的时钟或带有足够延时的单独帧或数据段。请注意，SPI 模式下不存在主器件或从器件的下溢错误信号，来自从器件的数据始终由主器件处理，即使从器件无法及时正确地准备数据也是如此。从器件最好使用 DMA，尤其是数据帧较短而总线速率较高时。

在多从器件系统中，每个序列必须通过 NSS 脉冲进行控制，从而只选择其中一个从器件进行通信。在单个从器件系统中，无需通过 NSS 来控制从器件，但此时提供此脉冲通常会更好，以在每个数据序列开始时同步从器件。NSS 可通过软件和硬件进行管理（请参见 *第 35.4.5 节：从器件选择 (NSS) 引脚管理*）。

当 BSY 位置 1 时，表示正在处理数据帧事务。当所进行的帧交互完成时，RXNE 标志将置 1。最后一位采样后，整个数据帧会存储到 RXFIFO 中。

禁止 SPI 的步骤

当禁止 SPI 时，必须按照本段中介绍的禁止步骤进行操作。当外设时钟停止时，在系统进入低功耗模式前做到这一点是十分重要的。否则会损坏正在进行的交互。在某些模式下，禁止步骤是停止所进行的连续通信的唯一方式。

当处于全双工或只发送模式下的主器件停止提供待发送的数据时，可结束任何事务。在这种情况下，时钟在最后一个数据传输后停止。当处理奇数数量的数据帧时，必须特别注意封装模式以防止交换一些空字节（请参见*数据封装*部分）。在这些模式下禁止 SPI 之前，用户必须按照标准的禁止步骤进行操作。SPI 在主模式传输时，如果在数据帧处理的过程中，或者 TXFIFO 中有待传输的数据帧，把 SPI 禁止，SPI 的此时的状态是不可预测的。所以，必须按照标准的禁止流程进行操作。

只要主器件处于只接收模式，停止连续时钟的唯一方式就是通过 SPE=0 来禁止外设。这必须在最后一个数据帧传输内的特定时间段，即第一位采样与最后一位传输开始之间完成（以便接收全部数量的预期数据帧并防止在最后一个有效数据帧后读取任何其他的“空”数据）。在该模式下禁止 SPI 时必须遵循特定步骤。

禁止 SPI 后，已接收但未读取的数据始终存储在 RXFIFO 中，这些数据必须在下次使能 SPI 后进行处理，然后才能启动新序列。为防止存在未读取的数据，需确保禁止 SPI 时 RXFIFO 为空，可通过正确的禁止步骤来禁止 SPI，也可以通过控制外设复位专用的特定寄存器以软件复位的方式来初始化所有 SPI 寄存器从而禁止 SPI（请参见 RCC_APBiRSTR 寄存器中的 SPIiRST 位）。

标准禁止步骤通过轮询 BSY 状态以及 FTLVL[1:0] 来检查发送会话是否完全结束。还可以在必须识别正在处理的传输是否结束的特定情况下完成这种检查，例如：

- 当 NSS 信号由软件管理且主器件必须为从器件提供 NSS 脉冲结束时，或者
- 最后一个数据帧或 CRC 帧传输仍在外设总线中处理时，来自 DMA 或 FIFO 的传输数据流完成。

正确的禁止步骤如下（使用只接收模式时除外）：

1. 等待至 FTLVL[1:0] = 00（无需发送更多数据）。
2. 等待至 BSY=0（最后一个数据帧已处理完）。
3. 禁止 SPI (SPE = 0)。
4. 读取数据，直至 FRLVL[1:0] = 00（读取接收的所有数据）。

某些只接收模式的正确禁止步骤如下：

1. 当最后一个数据帧正在处理时，通过在特定时间窗口内禁止 SPI (SPE=0) 来中断接收流。
2. 等待至 BSY=0（最后一个数据帧已处理完）。
3. 读取数据，直至 FRLVL[1:0] = 00（读取接收的所有数据）。

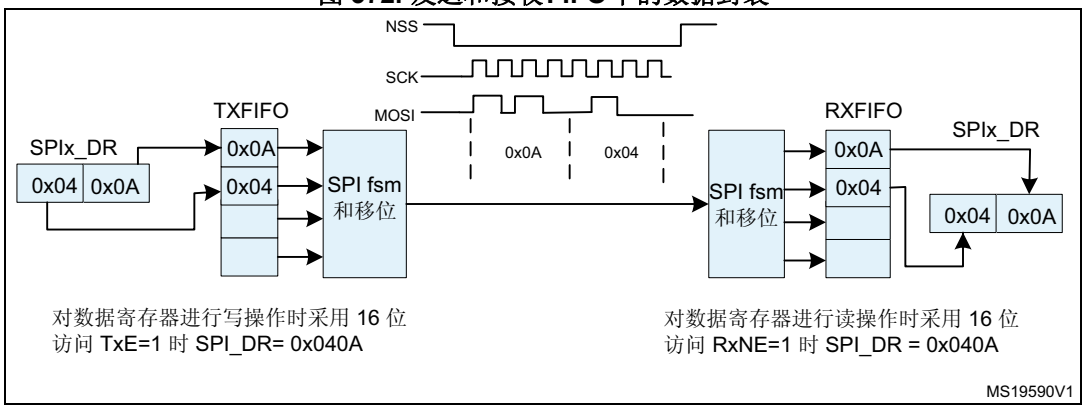
注： *如果使用封装模式并且必须接收奇数数量的数据帧且数据帧格式为小于或等于 8 位（不超过一个字节），则 FRXTH 必须在 FRLVL[1:0] = 01 时置 1，以便生成 RXNE 事件从而读取最后的奇数编号数据帧并且使 FIFO 指针保持正确对齐。*

数据封装

若数据帧长度不足一个字节（小于或等于 8 位），当 SPIx_DR 寄存器上执行任何 16 位读写访问时将自动使用数据封装。在这种情况下将并行处理双数据帧模式。最初，SPI 以所访问字的 LSB 中存储的模式工作，然后以 MSB 中存储的另一种模式来工作。*图 372* 给出了数据封装模式序列处理的示例。在对发送器的 SPIx_DR 寄存器执行单次 16 位访问后发送两个数据帧。如果 RXFIFO 阈值设置为 16 位 (FRXTH=0)，该序列只会在接收器中生成一个 RXNE 事件。接收器随后必须通过对 SPIx_DR 执行单次 16 位读访问来访问这两个数据帧，从而响应该单个 RXNE 事件。RxFIFO 阈值设置和后续读访问必须始终与接收器侧保持一致，因为若不一致，则会丢失数据。

如果必须处理奇数数量的此类“不超过一个字节”的数据帧，则会出现特定问题。在发送器侧，只需将任意奇序列的最后一个数据帧以 8 位访问的方式写入 SPIx_DR 即可。接收器必须针对所接收的奇序列中的最后一个数据帧更改 Rx_FIFO 阈值大小，以生成 RXNE 事件。

图 372. 发送和接收 FIFO 中的数据封装



使用 DMA（直接存储器寻址）进行通信

为了以最大速度工作并且方便避免上溢所需的数据寄存器读/写过程，SPI 提供了 DMA 功能，该功能采用了简单的请求/应答协议。

将 SPIx_CR2 寄存器中的使能位 TXE 或 RXNE 置 1 时，将请求 DMA 访问。必须向发送缓冲区和接收缓冲区发出单独的请求。

- 在发送过程中，每次 TXE 位置 1 都会发出 DMA 请求。然后，DMA 将对 SPIx_DR 寄存器执行写操作。
- 在接收过程中，每次 RXNE 位置 1 都会发出 DMA 请求。然后，DMA 将对 SPIx_DR 寄存器执行读操作。

请参见*图 373* 到*图 376*。

当 SPI 仅用于发送数据时，可以只使能 SPI Tx DMA 通道。在这种情况下，OVR 标志会置 1，因为未读取接收的数据。当 SPI 仅用于接收数据时，可以只使能 SPI Rx DMA 通道。

在发送模式下，DMA 写入所有要发送的数据（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 BSY 标志进行监视，以确保 SPI 通信已完成。在禁止 SPI 或进入停止模式前必须执行此步骤，以避免损坏最后一次发送。软件必须首先等待 FTLVL[1:0]=00，再等待 BSY=0。

通过 DMA 开始通信时，为防止 DMA 通道管理引发错误事件，必须按顺序执行以下步骤：

1. 如果使用 DMA Rx，通过 SPI_CR2 寄存器中的 RXDMAEN 位来使能 DMA 接收缓冲区。
2. 如果使用数据流，通过 DMA 寄存器来使能 Tx 和 Rx 的 DMA 数据流。
3. 如果使用 DMA Tx，通过 SPI_CR2 寄存器中的 TXDMAEN 位来使能 DMA 发送缓冲区。
4. 通过将 SPE 位置 1 使能 SPI。

要关闭通信，必须按顺序执行以下步骤：

1. 如果使能了 DMA，通过 DMA 寄存器来禁止 Tx 和 Rx 的 DMA 数据流。
2. 通过后续 SPI 禁止步骤来禁止 SPI。
3. 如果使用 DMA Tx 和/或 DMA Rx，通过将 SPI_CR2 寄存器中的 TXDMAEN 和 RXDMAEN 位清零来禁止 DMA 发送缓冲区和接收缓冲区。

使用 DMA 时的数据封装

如果由 DMA 管理传输（SPIx_CR2 寄存器中的 TXDMAEN 和 RXDMAEN 位置 1），则将根据为 SPI TX 和 SPI RX 的 DMA 通道配置的 PSIZE 值自动使能/禁止封装模式。如果 DMA 通道的 PSIZE 值等于 16 位，且 SPI 数据大小小于或等于 8 位，则将使能封装模式。然后，DMA 将自动管理对 SPIx_DR 寄存器的写操作。

如果使用数据封装模式，且要传输的数据数量不是 2 的倍数，则 LDMA_TX/LDMA_RX 位必须置 1。然后，SPI 会认为最后一次 DMA 传输时只发送或接收一个数据（有关详细信息，请参见*第 1135 页的数据封装*）。

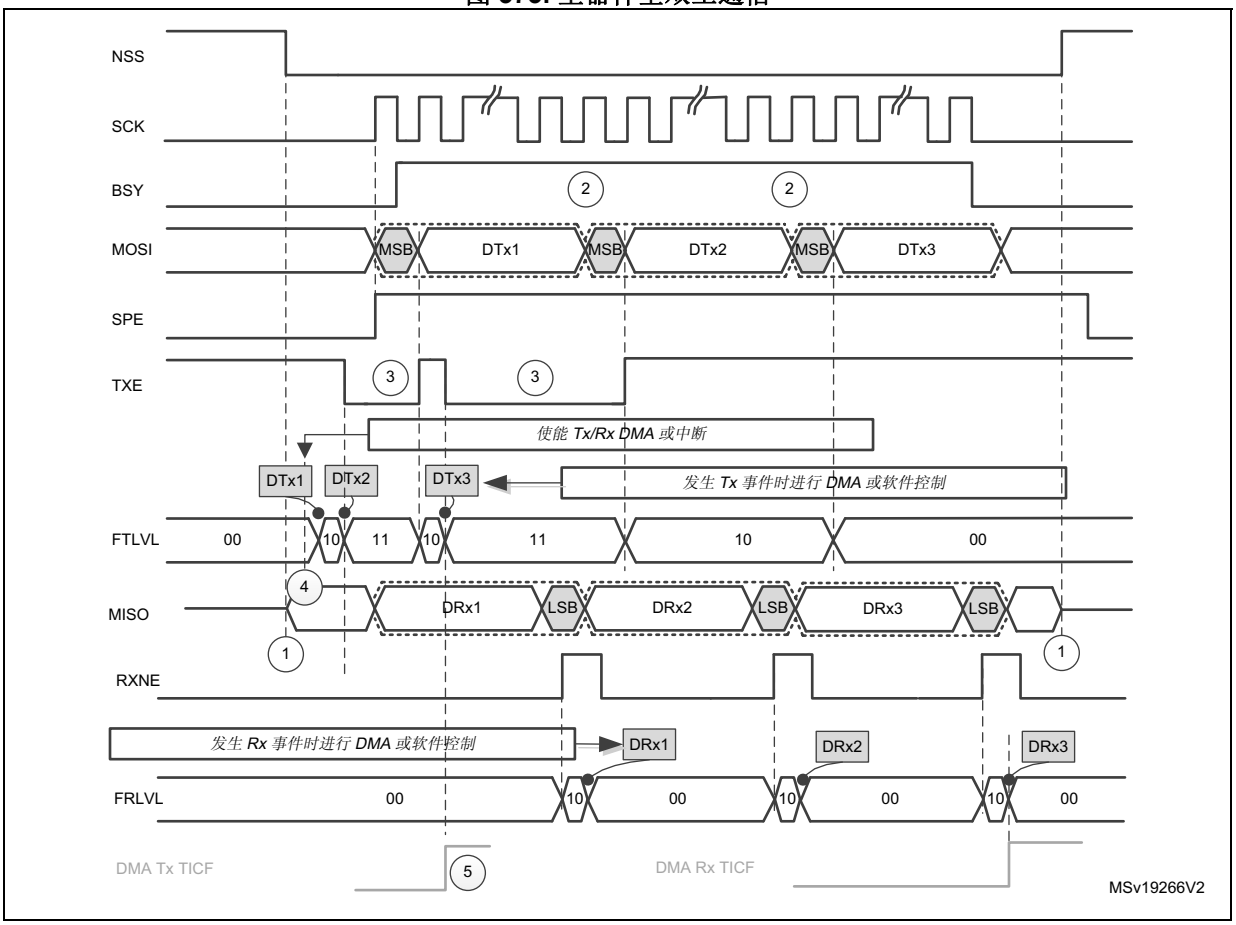
通信图

本部分将介绍一些典型的时序图。无论 SPI 事件是通过轮询、中断还是 DMA 进行处理，这些时序图均有效。为简单起见，此处均假设 LSBFIRST=0、CPOL=0 和 CPHA=1。不提供 DMA 数据流的完整配置。

以下带编号的注释对*第 1138 页的图 373* 到*第 1141 页的图 376* 均适用。

1. 激活 NSS 并使能 SPI 后，从器件开始控制 MISO 线，而当其中一个条件不成立时，从器件将与 MISO 线断开。必须为从器件提供充足的时间，以便在传输开始前准备好主器件专用的数据。
   在主器件上，只有使能 SPI 后，SPI 外设才会控制 MOSI 和 SCK 信号（偶尔还会控制 NSS 信号）。如果禁止 SPI，SPI 外设会与 GPIO 逻辑断开，因此这些线上的电平只取决于 GPIO 设置。
2. 在主器件上，如果通信（时钟信号）连续，BSY 在数据帧之间保持有效。在从器件上，BSY 信号在数据帧之间始终会保持至少一个时钟周期的低电平状态。
3. 只有 TXFIFO 已满时，TXE 信号才会清零。
4. DMA 仲裁过程在 TXDMAEN 位置 1 后立即开始。TXE 中断在 TXEIE 置 1 后立即生成。TXE 信号处于有效电平时，开始向 TxFIFO 传输数据，直至 TxFIFO 已满或 DMA 传输完成。
5. 如果要发送的所有数据可装入 TxFIFO，则 DMA Tx TCIF 标志甚至会在 SPI 总线上的通信开始前置 1。SPI 传输完成前，该标志始终为高电平状态。
6. 数据包的 CRC 值是通过 SPIx_TXCPRCR 和 SPIx_RXCPCR 寄存器一帧一帧连续生成的，完成整个数据封装后，CRC 信息可通过 DMA 自动处理（Tx 通道必须设置为要处理的数据帧数），也可由软件处理（用户必须在处理最后一个数据帧的过程中处理 CRCNEXT 位）。
   在发送端 SPIx_TXCPCR 所计算出的CRC值只是简单的由发送器发送出去，而接收端收到的 CRC 信息会被加载到 RxFIFO 与 SPIx_RXCRCR 寄存器内容进行比较（如果不一致则 CRC 错误标志位会置位）。因此用户必须注意刷新 FIFO 中的相关信息，可以通过软件读出 RxFIFO 中存储的所有内容的方式来实现，若已针对 Rx 通道预设置了适当数量的数据帧（数据帧数 + CRC 帧数），也可通过 DMA 的方式来实现（请参见示例假设中的设置）。
7. 在数据封装模式下，TxE 和 RxNE 事件成对出现，对 FIFO 的每次读/写访问都为 16 位宽，直至数据帧数为偶数。如果 TxFIFO 处于 ¾ 满状态，则 FTLVL 将保持 FIFO 满时对应的状态。因此在 TxFIFO 变为 ½ 满状态前，无法存储最后一个奇数编号的数据帧。该帧可通过软件或自动由 DMA（LDMA_TX 控制置 1 时）对其进行 8 位访问的方式存储在 TxFIFO 中。
8. 要在封装模式下接收最后一个奇数编号的数据帧，必须在处理最后一个数据帧后，将 Rx 阈值更改为 8 位，这可通过软件（设置为 FRXTH=1）实现或由 DMA 内部信号在 LDMA_RX 置 1 时自动实现。

图 373. 主器件全双工通信



主器件全双工通信示例的假设条件如下：

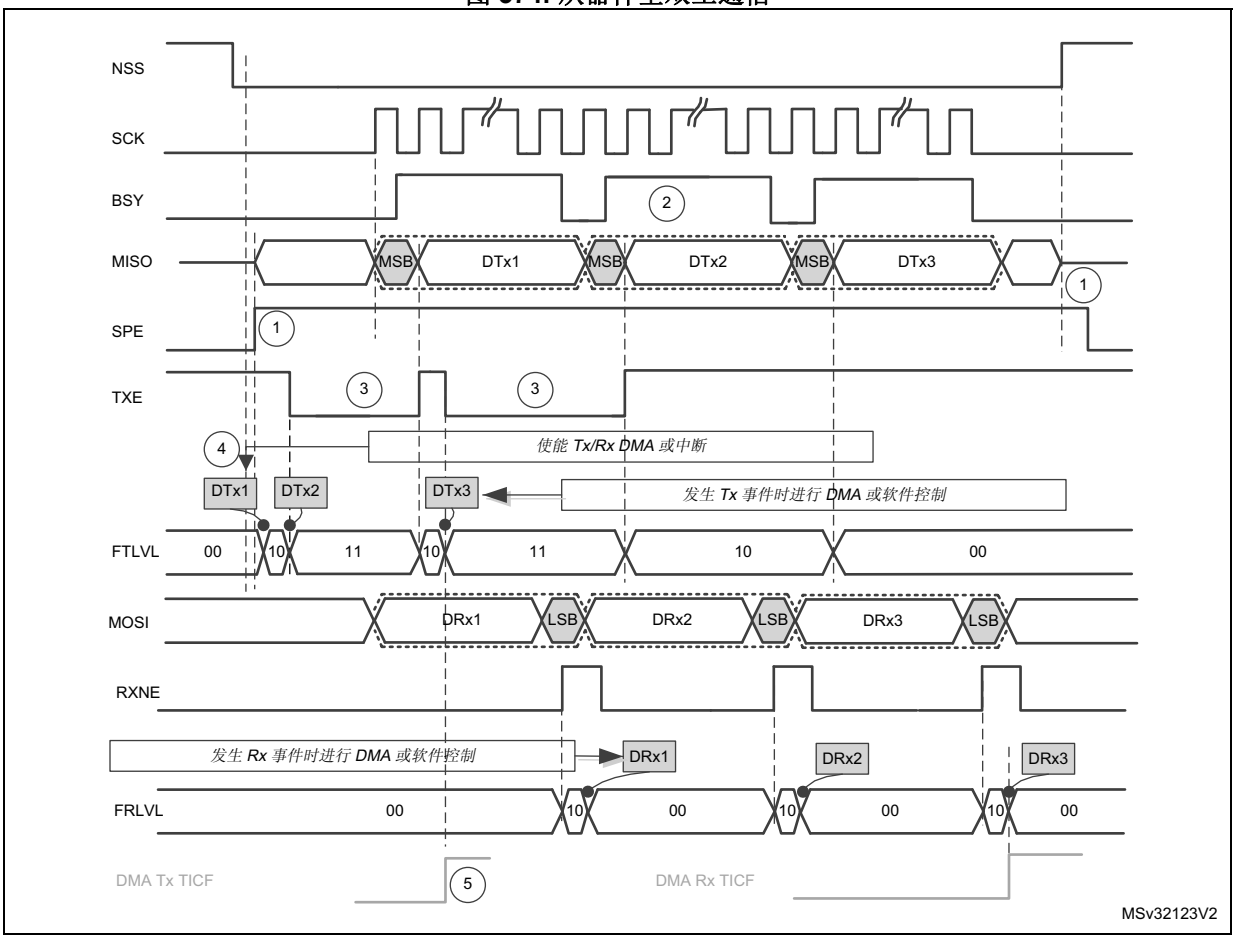
- 数据大小 > 8 位

如果使用 DMA：

- 由 DMA 处理的 Tx 帧的数量设置为 3
- 由 DMA 处理的 Rx 帧的数量设置为 3

有关通用假设条件和注释的详细信息，另请参见*第 1137 页的通信图*。

图 374. 从器件全双工通信



从器件全双工通信示例的假设条件如下：

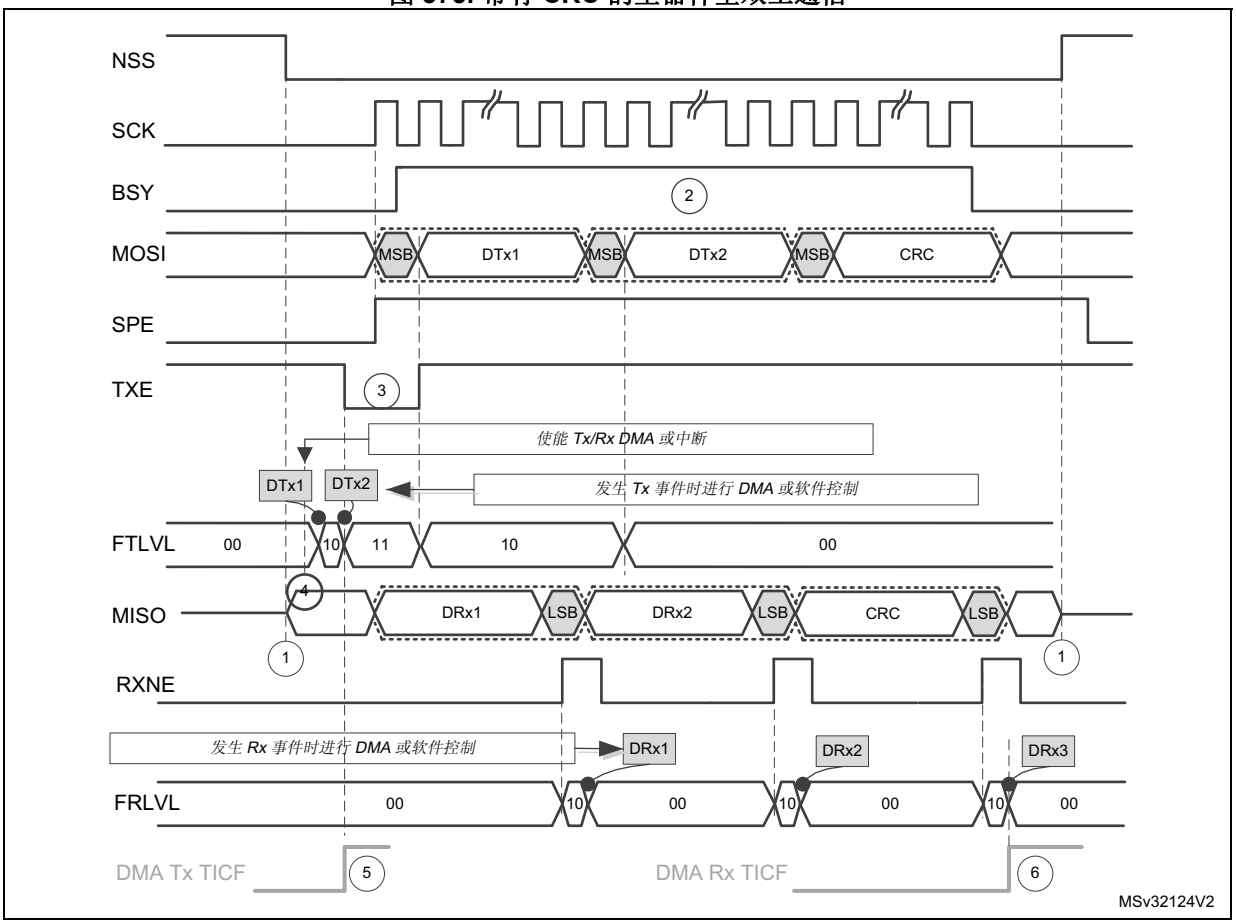
- 数据大小 > 8 位

如果使用 DMA：

- 由 DMA 处理的 Tx 帧的数量设置为 3
- 由 DMA 处理的 Rx 帧的数量设置为 3

有关通用假设条件和注释的详细信息，另请参见*第 1137 页的通信图*。

图 375. 带有 CRC 的主器件全双工通信



带有 CRC 的主器件全双工通信的假设条件如下：

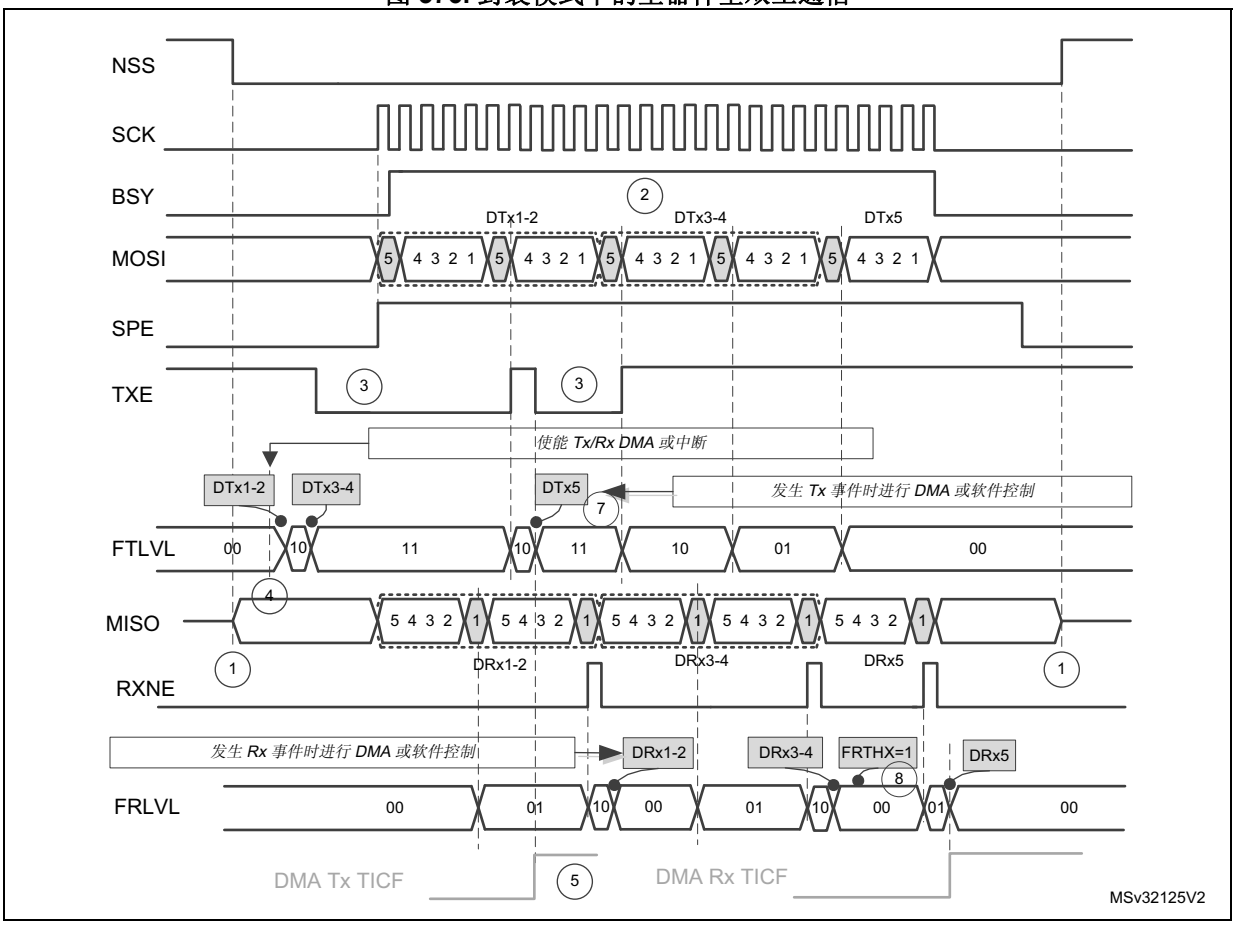
- 数据大小 = 16 位
- CRC 已使能

如果使用 DMA：

- 由 DMA 处理的 Tx 帧的数量设置为 2
- 由 DMA 处理的 Rx 帧的数量设置为 3

有关通用假设条件和注释的详细信息，另请参见*第 1137 页的通信图*。

图 376. 封装模式下的主器件全双工通信



封装模式下的主器件全双工通信示例的假设条件如下：

- 数据大小 = 5 位
- 主要通过 16 位访问的方式来读/写 FIFO
- FRXTH=0

如果使用 DMA：

- 要由 DMA 处理的 Tx 帧的数量设置为 3
- 要由 DMA 处理的 Rx 帧的数量设置为 3
- Tx 和 Rx 的 DMA 通道的 PSIZE 均设置为 16 位
- LDMA_TX=1 且 LDMA_RX=1

有关通用假设条件和注释的详细信息，另请参见*第 1137 页的通信图*。

35.4.10 SPI 状态标志

应用可通过三种状态标志监视 SPI 总线的状态。

发送缓冲区为空 (TXE)

当发送 TXFIFO 有足够的空间来存储要发送的数据时，TXE 标志将置 1。TXE 标志与 TXFIFO 占用水平相关。该标志变为高电平后将一直保持高电平状态，直至 TXFIFO 占用水平小于或等于 FIFO 深度的 1/2。如果 SPIx_CR2 寄存器中的 TXEIE 位置 1，可产生中断。当 TXFIFO 占用水平超过 1/2 时，此位将自动清零。

接收缓冲区非空 (RXNE)

RXNE 标志根据 SPIx_CR2 寄存器中 FRXTH 位的值进行设置：

- 如果 FRXTH 置 1，RXNE 将变为高电平并一直保持高电平状态，直至 RXFIFO 占用水平大于或等于 1/4（8 位）。
- 如果 FRXTH 清零，RXNE 将变为高电平并一直保持高电平状态，直至 RXFIFO 占用水平大于或等于 1/2（16 位）。

如果 SPIx_CR2 寄存器中的 RXNEIE 位置 1，可产生中断。

当上述条件不再为真时，RXNE 将由硬件自动清零。

忙标志 (BSY)

BSY 标志由硬件置 1 和清零（写入此标志没有任何作用）。

当 BSY 置 1 时，表示 SPI 上正在进行数据传输（SPI 总线繁忙）。

某些模式下可以使用 BSY 标志来检测传输是否结束，以便软件在进入低功耗模式（该模式下不提供外设时钟）前禁止 SPI 或其外设时钟。这可避免破坏最后一个数据的传输。

BSY 标志还可用于避免在多主模式系统中发生写冲突。

在以下任意一种条件下，BSY 标志将清零：

- 正确禁止 SPI 时
- 在主模式下检测到故障时（MODF 位置 1）
- 在主模式下，完成了数据发送并且不准备发送任何新数据时
- 在从模式下，BSY 标志在各传输之间的至少一个 SPI 时钟周期内置为“0”时

注： *当主器件可以立即处理下一次发送时（例如，如果主器件处于只接收模式或其发送 FIFO 不为空），在主器件侧的传输之间，通信连续且 BSY 标志始终置“1”。尽管从器件并非如此，但建议始终使用 TXE 和 RXNE 标志（而非 BSY 标志）来处理数据发送或接收操作。*

35.4.11 SPI 错误标志

如果以下其中一个错误标志置 1 且已通过将 ERRIE 位置 1 使能了中断，则将生成 SPI 中断。

上溢标志 (OVR)

当主器件或从器件接收了数据但 RXFIFO 没有足够的空间来存储接收的数据时，将出现上溢的情况。如果软件或 DMA 没有足够的时间来读取之前接收的数据（存储在 RXFIFO 中）或数据存储空间受限 （例如在只接收模式下使能 CRC 时 RXFIFO 不可用，在这种情况下，接收缓冲区便限制为一个数据帧缓冲区），会发生这种情况 （请参见*第 35.4.14 节：CRC 计算*）。

当出现上溢的情况时，新接收的值不会覆盖 RXFIFO 中之前的值。新接收的值将被丢弃，之后发送的所有数据都将丢失。要将 OVR 位清零，应首先对 SPI_DR 寄存器执行读访问，然后再对 SPI_SR 寄存器执行读访问。

模式故障 (MODF)

当主器件的内部 NSS 信号（NSS 硬件模式下为 NSS 引脚，NSS 软件模式下为 SSI 位）被拉低时，将发生模式故障。这会自动将 MODF 位置 1。主模式故障会在以下几方面影响 SPI 接口：

- 如果 ERRIE 位置 1，MODF 位将置 1，并生成 SPI 中断。
- SPE 位清零。这将关闭器件的所有输出，并禁止 SPI 接口。
- MSTR 位清零，从而强制器件进入从模式。

使用以下软件序列将 MODF 位清零：

1. 在 MODF 位置 1 时，对 SPIx_SR 寄存器执行读或写访问。
2. 然后，对 SPIx_CR1 寄存器执行写操作。

为避免包含多个 MCU 的系统中发生多从模式冲突，必须在 MODF 位清零序列期间将 NSS 引脚拉高。在该清零序列后，可以将 SPE 和 MSTR 位恢复到原始状态。安全起见，硬件不允许在 MODF 位置 1 时将 SPE 和 MSTR 位置 1。在从器件中，MODF 位不可置 1，但由前一次多主模式冲突引起时除外。

CRC 错误 (CRCERR)

当 SPIx_CR1 寄存器中的 CRCEN 位置 1 时，此标志用于验证接收数据的有效性。如果移位寄存器中接收的值与 SPIx_RXCRCR 的值不匹配，SPIx_SR 寄存器中的 CRCERR 标志将置 1。该标志由软件清零。

TI 模式帧格式错误 (FRE)

如果 SPI 在从模式下工作，并配置为符合 TI 模式协议，则在通信进行期间出现 NSS 脉冲时，将检测到 TI 模式帧格式错误。出现此错误时，SPIx_SR 寄存器中的 FRE 标志将置 1。发生错误时不会禁止 SPI，但会忽略 NSS 脉冲，并且 SPI 会等待下一个 NSS 脉冲，然后再开始新的传输。由于错误检测可能导致丢失两个数据字节，因此数据可能会损坏。

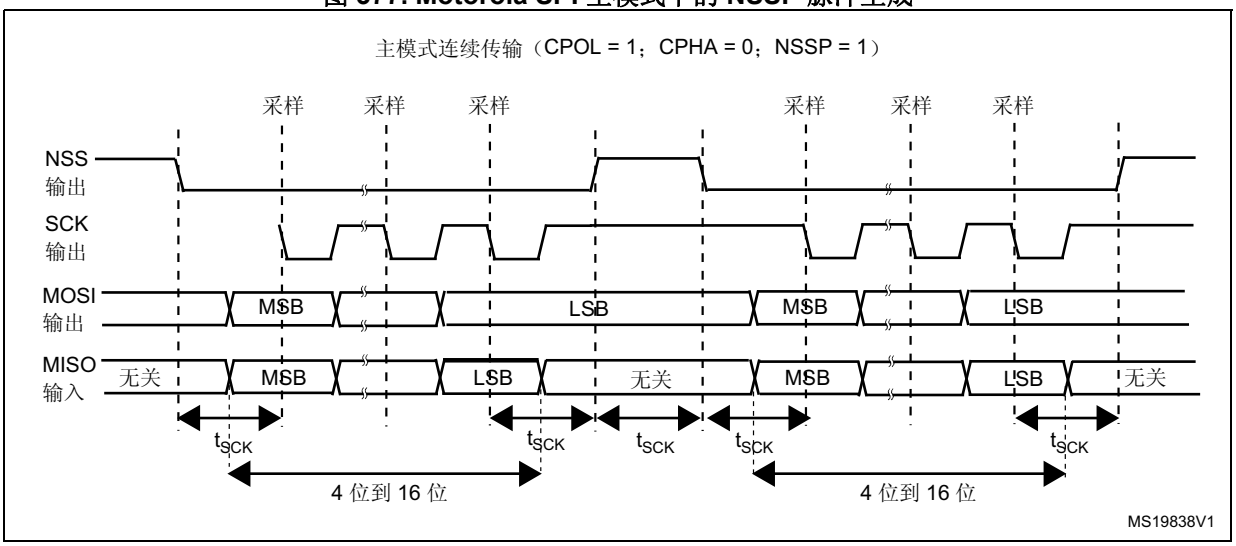
读取 SPIx_SR 寄存器时，将清零 FRE 标志。如果 ERRIE 位置 1，则检测到 NSS 错误时将生成中断。在这种情况下，由于无法保证数据的一致性，应禁止 SPI，并在重新使能从 SPI 后，由主器件重新发起通信。

35.4.12 NSS 脉冲模式

该模式通过 SPIx_CR2 寄存器中的 NSSP 位来激活，只有将 SPI 接口配置为 Motorola SPI 主模式 (FRF=0) 且在第一个边沿捕捉时，该模式才起作用（SPIx_CR1 CPHA = 0，CPOL 设置忽略）。激活后，当 NSS 至少保持一个时钟周期的高电平状态时，两个连续的数据帧传输间将生成 NSS 脉冲。该模式下，从器件可以锁存数据。NSSP 脉冲模式旨在用于具有一个主器件-从器件对的应用。

图 377 给出了使能 NSSP 脉冲模式后的 NSS 引脚管理情况。

图 377. Motorola SPI 主模式下的 NSSP 脉冲生成



注： 当 CPOL = 0 时会出现类似行为。在这种情况下，采样边沿为 SCK 的*上升*沿，NSS 的使能和禁止均参考该采样边沿。

35.4.13 TI 模式

主模式下的 TI 协议

SPI 接口与 TI 协议兼容。可以使用 SPIx_CR2 寄存器的 FRF 位来配置 SPI，以兼容此协议。

时钟极性和相位都被强制为遵循 TI 协议，和 SPIx_CR1 中的设置无关。NSS 管理也特定于 TI 协议，在这种情况下，无法通过 SPIx_CR1 和 SPIx_CR2 寄存器（SSM、SSI 和 SSOE）来对 NSS 管理进行配置。

在从模式下，SPI 波特率预分频器用于控制在当前传输完成时 MISO 引脚切换为高阻态的时刻（请参见*图 378*）。可以使用任意波特率，因此可以非常灵活地确定此时刻。但是，波特率通常设置为外部主时钟波特率。MISO 信号变为高阻态的延时 ($t_{release}$) 取决于内部重新同步以及通过 SPIx_CR1 寄存器的 BR[2:0] 位设置的波特率值。具体公式如下：

$$\frac{t_{baud\_rate}}{2} + 4 \times t_{pclk} < t_{release} < \frac{t_{baud\_rate}}{2} + 6 \times t_{pclk}$$

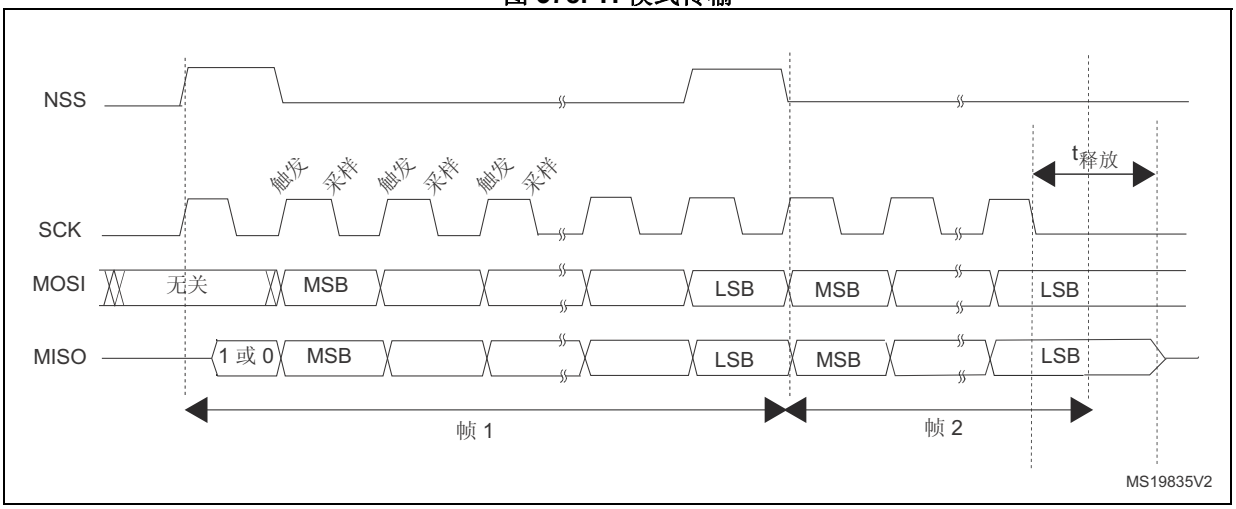
如果从器件在数据帧传输期间检测到错位的 NSS 脉冲，TIFRE 标志将置 1。

如果数据大小等于 4 位或 5 位，处于全双工模式或只发送模式的主器件将使用在 LSB 后再添加一个空数据位的协议。每个周期内，将在该空位（而非 LSB）时钟周期生成 TI NSS 脉冲。

此特性不适用于 Motorola SPI 通信（FRF 位为 0）。

*图 378：TI 模式传输*给出了选择 TI 模式时的 SPI 通信波形。

图 378. TI 模式传输



35.4.14 CRC 计算

为检查发送数据和接收数据的可靠性，使用两个独立的 CRC 计算器。SPI 可提供 CRC8 或 CRC16 计算，而与固定为 8 位或 16 位的数据帧长度无关。对于所有其他的数据帧长度，CRC 均不适用。

CRC 原理

在使能 SPI (SPE = 1) 前，通过将 SPIx_CR1 寄存器中的 CRCEN 位置 1 来使能 CRC 计算。使用值为奇数的可编程多项式对每个位计算 CRC 值。在由 SPIx_CR1 寄存器中的 CPHA 位和 CPOL 位定义的采样时钟边沿进行计算。所计算的 CRC 值在数据块末尾自动进行校验，以及针对由 CPU 或 DMA 管理的传输进行校验。当检测到所接收数据内部计算的 CRC 与发送器发送的 CRC 不匹配时，CRCERR 标志将置 1 以指示数据损坏错误。CRC 计算的正确处理步骤取决于 SPI 配置和所选的传输管理。

注： *多项式值只应为奇数。不支持任何偶数值。*

CPU 管理的 CRC 传输

通信开始后将一直持续到必须发送或接收 SPIx_DR 寄存器中的最后一个数据帧时。之后，SPIx_CR1 寄存器中的 CRCNEXT 位必须置 1，以指示当前处理的数据帧传输后将处理 CRC 帧传输。CRCNEXT 位必须在最后一个数据帧传输结束前置 1。在 CRC 传输期间，CRC 计算将冻结。

所接收的 CRC 以数据字节或数据字的形式存储在 RXFIFO 中。因此仅在 CRC 模式下，接收缓冲区才必须被视为一个 16 位缓冲区且一次仅接收一个数据帧。

CRC 帧的传输通常在数据序列结束时再传输一个数据帧。然而，在设置通过 16 位 CRC 校验的 8 位数据帧时，还需要另外两个帧才能发送完整的 CRC。

接收最后一个 CRC 数据后，将执行自动校验，将接收的值与 SPIx_RXCRC 寄存器中的值进行比较。软件必须校验 SPIx_SR 寄存器中的 CRCERR 标志，以确定数据传输是否损坏。软件通过向 CRCERR 标志写入“0”来将其清零。

接收 CRC 后，CRC 值存储到 RXFIFO 中，且必须在 SPIx_DR 寄存器中进行读取，以将 RXNE 标志清零。

DMA 管理的 CRC 传输

当使能的 SPI 通信支持 CRC 通信和 DMA 模式时，在通信结束时会自动发送和接收 CRC（在只接收模式下读取 CRC 数据时除外）。CRCNEXT 位不是一定要通过软件来处理。SPI 发送 DMA 通道计数器必须设置为要发送的数据帧数，其中不包括 CRC 帧。在接收器侧，接收的 CRC 值在传输结束时通过 DMA 自动处理，但用户必须注意刷新 RXFIFO 中接收的 CRC 信息（因为该信息始终加载到其中）。在全双工模式下，接收 DMA 通道计数器可设置为要接收的数据帧数，其中包括 CRC，这意味着在通过 16 位 CRC 校验的 8 位数据帧的特殊情况下：

DMA_RX = Numb_of_data + 2

在只接收模式下，DMA 接收通道计数器应仅包含已传输的数据量，而不包括 CRC 计算。之后，基于通过 DMA 实现的完整传输，必须通过软件从 FIFO 中读回所有 CRC 值，因为 FIFO 在该模式下用作单个缓冲区。

如果传输过程中出现损坏，则在数据和 CRC 传输结束时，SPIx_SR 寄存器中的 CRCERR 标志将置 1。

如果使用封装模式，则 LDMA_RX 位需要管理数据数是否为奇数。

复位 SPIx_TXCRC 和 SPIx_RXCRC 值

在 CRC 阶段后对新数据进行采样时，SPIx_TXCRC 和 SPIx_RXCRC 值将自动清零。借此，可使用 DMA 循环模式（只接收模式下不适用）从而不间断地传输数据（中间 CRC 校验阶段会涉及一些数据块）。

如果在通信过程中禁止 SPI，必须按以下顺序进行操作：

1. 禁止 SPI
2. 将 CRCEN 位清零
3. 使能 CRCEN 位
4. 使能 SPI

注： *当 SPI 接口配置为从模式时，一旦 CRCNEXT 信号被释放，NSS 内部信号需要在 CRC 阶段事务期间保持低电平。这就是当 NSS 硬件模式正常应用于从器件时，不能在 NSS 脉冲模式下使用 CRC 计算的原因。*

在 TI 模式下，尽管时钟相位和时钟极性设置固定并且与 SPIx_CR1 寄存器无关，但如果应用 CRC，则 SPIx_CR1 寄存器中必须保持相应的设置（CPOL = 0，CPHA = 1）。此外，CRC 计算必须通过 SPI 禁止序列在会话之间复位，方法是在主器件和从器件两侧重新使能上述 CRCEN 位，否则 CRC 计算可能在该特定模式下损坏。

35.5 SPI 中断

在 SPI 通信过程中，中断可由以下事件产生：

- 发送 TXFIFO 准备就绪，可以装载数据
- 接收 RXFIFO 中接收了数据
- 主模式故障
- 上溢错误
- TI 帧格式错误
- CRC 协议错误

中断可分别进行使能和禁止。

表 206. SPI 中断请求

中断事件	事件标志	使能控制位
发送 TXFIFO 准备就绪，可以装载数据	TXE	TXEIE
RXFIFO 中接收了数据	RXNE	RXNEIE
主模式故障	MODF	ERRIE
上溢错误	OVR	
TI 帧格式错误	FRE	
CRC 协议错误	CRCERR	

35.6 SPI 寄存器

外设寄存器可支持半字（16 位）或字（32 位）访问。此外，SPI_DR 可支持 8 位访问。

35.6.1 SPI 控制寄存器 1 (SPIx_CR1)

SPI control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDIMODE	BIDIOE	CRCEN	CRCNEXT	CRCL	RXONLY	SSM	SSI	LSBFIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 **BIDIMODE**：双向通信数据模式使能 (Bidirectional data mode enable)

此位通过一条共用的双向数据线来支持半双工通信。双向模式激活时，使 RXONLY 位保持清零状态。

0：选择双线单向通信数据模式

1：选择单线双向数据模式

位 14 **BIDIOE**：双向模式下的输出使能 (Output enable in bidirectional mode)

此位与 BIDIMODE 位，用于选择双向模式下的传输方向。

0：禁止输出（只接收模式）

1：使能输出（只发送模式）

注： 在主模式下，使用 MOSI 引脚；在从模式下，使用 MISO 引脚。

位 13 **CRCEN**：硬件 CRC 计算使能 (Hardware CRC calculation enable)

0：禁止 CRC 计算

1：使能 CRC 计算

注： 为确保正确操作，只应在禁止 SPI（SPE = “0”）时对此位执行写操作。

位 12 **CRCNEXT**：发送下一个 CRC (Transmit CRC next)

0：下一个发送值来自发送缓冲区。

1：下一个发送值来自发送 CRC 寄存器。

注： 向 SPIx_DR 寄存器写入最后一个数据后，必须立即对此位执行写操作。

位 11 **CRCL**：CRC 长度 (CRC length)

此位由软件置 1 和清零，用以选择 CRC 长度。

0：8 位 CRC 长度

1：16 位 CRC 长度

注： 为确保正确操作，只应在禁止 SPI（SPE = “0”）时对此位执行写操作。

位 10 **RXONLY**：只接收模式使能 (Receive only mode enabled)

此位用于使能通过一条双向线专门接收数据的单工通信。当只接收模式激活时，将 BIDIMODE 位保持清零状态。此位也适用于多从模式系统，在此类系统中，不会访问特定从器件，也不会损坏访问的从器件的输出。

0：全双工（发送和接收）

1：禁止输出（只接收模式）

位 9 **SSM**：软件从器件管理 (Software slave management)

当 SSM 位置 1 时，NSS 引脚输入替换为 SSI 位的值。

0：禁止软件从器件管理

1：使能软件从器件管理

注：　此位不适用于 SPI TI 模式。

位 8 **SSI**：内部从器件选择 (Internal slave select)

仅当 SSM 位置 1 时，此位才有效。此位的值将作用到 NSS 引脚上，并忽略 NSS 引脚的 I/O 值。

注：　此位不适用于 SPI TI 模式。

位 7 **LSBFIRST**：帧格式 (Frame format)

0：发送/接收数据时 MSB 在前

1：发送/接收数据时 LSB 在前

注：　1. 正在通信时不应更改此位。

2. 此位不适用于 SPI TI 模式。

位 6 **SPE**：SPI 使能 (SPI enable)

0：禁止外设

1：使能外设

注：　禁止 SPI 时，请按照第 1134 页的禁止 SPI 的步骤中所述的步骤操作。

位 5:3 **BR[2:0]**：波特率控制 (Baud rate control)

000：$f_{PCLK}/2$

001：$f_{PCLK}/4$

010：$f_{PCLK}/8$

011：$f_{PCLK}/16$

100：$f_{PCLK}/32$

101：$f_{PCLK}/64$

110：$f_{PCLK}/128$

111：$f_{PCLK}/256$

注：　正在通信时不应更改这些位。

位 2 **MSTR**：主模式选择 (Master selection)

0：从配置

1：主配置

注：　正在通信时不应更改此位。

位 1 **CPOL**：时钟极性 (Clock polarity)

0：空闲状态时，CK 为 0

1：空闲状态时，CK 为 1

注： 正在通信时不应更改此位。

除了在 TI 模式下应用 CRC 的情况外，此位不会用于 SPI TI 模式。

位 0 **CPHA**：时钟相位 (Clock phase)

0：从第一个时钟边沿开始采样数据

1：从第二个时钟边沿开始采样数据

注： 正在通信时不应更改此位。

除了在 TI 模式下应用 CRC 的情况外，此位不会用于 SPI TI 模式。

35.6.2 SPI 控制寄存器 2 (SPIx_CR2)

SPI control register 2

偏移地址：0x04

复位值：0x0700

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LDMA_TX	LDMA_RX	FRXTH	DS[3:0]				TXEIE	RXNEIE	ERRIE	FRF	NSSP	SSOE	TXDMAEN	RXDMAEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 保留，必须保持复位值。

位 14 **LDMA_TX**：发送的最后一次 DMA 传输 (Last DMA transfer for transmission)

此位用于数据封装模式，用于定义通过 DMA 发送的数据总数为奇数还是偶数。只有 SPIx_CR2 寄存器中的 TXDMAEN 位置 1 且使用封装模式（数据长度 =< 8 位，对 SPIx_DR 的写访问为 16 位宽）时，此位才有意义。当 SPI 禁止时（SPIx_CR1 寄存器中的 SPE = 0），必须对其执行写操作。

0：待传输数据数量为偶数

1：待传输数据数量为奇数

注： 如果 CRCEN 位置 1，请参见第 1134 页的禁止 SPI 的步骤。

位 13 **LDMA_RX**：接收的最后一次 DMA 传输 (Last DMA transfer for reception)

此位用于数据封装模式下，用于定义通过 DMA 接收的数据总数为奇数还是偶数。只有 SPIx_CR2 寄存器中的 RXDMAEN 位置 1 且使用封装模式（数据长度 =< 8 位，对 SPIx_DR 的写访问为 16 位宽）时，此位才有意义。当 SPI 禁止时（SPIx_CR1 寄存器中的 SPE = 0），必须对其执行写操作。

0：待传输数据数量为偶数

1：待传输数据数量为奇数

注： 如果 CRCEN 位置 1，请参见第 1134 页的禁止 SPI 的步骤。

位 12 **FRXTH**：FIFO 接收阈值 (FIFO reception threshold)

此位用于设置触发 RXNE 事件的 RXFIFO 阈值

0：如果 FIFO 占用水平大于或等于 1/2（16 位），将生成 RXNE 事件。

1：如果 FIFO 占用水平大于或等于 1/4（8 位），将生成 RXNE 事件。

注：

位 11:8 **DS[3:0]**：数据位宽 (Data size)

这些位用于配置 SPI 传输的数据位宽。

0000：未使用
0001：未使用
0010：未使用
0011：4 位
0100：5 位
0101：6 位
0110：7 位
0111：8 位
1000：9 位
1001：10 位
1010：11 位
1011：12 位
1100：13 位
1101：14 位
1110：15 位
1111：16 位

如果软件尝试写入其中一个“未使用”值，这些位将被强制设为“0111”（8 位）。

位 7 **TXEIE**：发送缓冲区空中断使能 (Tx buffer empty interrupt enable)

0：屏蔽 TXE 中断。
1：使能 TXE 中断。TXE 标志置 1 时产生中断请求。

位 6 **RXNEIE**：接收缓冲区非空中断使能 (RX buffer not empty interrupt enable)

0：屏蔽 RXNE 中断。
1：使能 RXNE 中断。RXNE 标志置 1 时产生中断请求。

位 5 **ERRIE**：错误中断使能 (Error interrupt enable)

此位用于控制在错误状况发生时是否产生中断（SPI 模式下的 CRCERR、OVR 和 MODF，TI 模式下的 FRE）。

0：屏蔽错误中断
1：使能错误中断

位 4 **FRF**：帧格式 (Frame format)

0：SPI Motorola 模式
1：SPI TI 模式

注：　只有在禁止 SPI (SPE=0) 后才能对此位执行写操作。

位 3 **NSSP**：NSS 脉冲管理 (NSS pulse management)

此位仅用于主模式。连续传输时，此位允许 SPI 在两个连续数据间生成 NSS 脉冲。单次数据传输时，此位强制 NSS 引脚在传输后变为高电平。

如果 CPHA = “1”或 FRF = “1”，此位无意义。

0：未生成 NSS 脉冲
1：生成 NSS 脉冲

注：　1. 只有在禁止 SPI (SPE=0) 后才能对此位执行写操作。
2. 此位不适用于 SPI TI 模式。

位 2 **SSOE**：SS 输出使能 (SS output enable)
0：在主模式下禁止 SS 输出，SPI 接口可在多主模式配置下工作。
1：在主模式下以及使能 SPI 接口后使能 SS 输出。SPI 接口不能在多主模式环境下工作。
注： 此位不适用于 SPI TI 模式。

位 1 **TXDMAEN**：发送缓冲区 DMA 使能 (Tx buffer DMA enable)
当此位置 1 时，每当 TXE 标志置 1，即产生 DMA 请求。
0：禁止发送缓冲区 DMA
1：使能发送缓冲区 DMA

位 0 **RXDMAEN**：接收缓冲区 DMA 使能 (Rx buffer DMA enable)
当此位置 1 时，每当 RXNE 标志置 1，即产生 DMA 请求。
0：禁止接收缓冲区 DMA
1：使能接收缓冲区 DMA

35.6.3 SPI 状态寄存器 (SPIx_SR)

SPI status register

偏移地址：0x08

复位值：0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	FTLVL[1:0]		FRLVL[1:0]		FRE	BSY	OVR	MODF	CRCERR			TXE	RXNE
			r	r	r	r	r	r	r	r	rc_w0			r	r

位 15:13 保留，必须保持复位值。

位 12:11 **FTLVL[1:0]**：FIFO 发送等级 (FIFO transmission level)
这些位将由硬件置 1 和清零。
00：FIFO 为空
01：1/4 FIFO
10：1/2 FIFO
11：FIFO 为满（当 FIFO 阈值大于 1/2 时即视为满）

位 10:9 **FRLVL[1:0]**：FIFO 接收级别 (FIFO reception level)
这些位将由硬件置 1 和清零。
00：FIFO 为空
01：1/4 FIFO
10：1/2 FIFO
11：FIFO 已满
注： 使能 CRC 计算时，这些位不适用于 SPI 仅接收模式。

位 8 **FRE**：帧格式错误 (Frame format error)
该标志在 TI 从模式下用于 SPI。请参见*第 35.4.11 节：SPI 错误标志*。
此标志由硬件置 1，在读取 SPIx_SR 时由软件复位。
0：未发生帧格式错误
1：发生帧格式错误

位 7 **BSY**：忙标志 (Busy flag)

0：SPI 不忙

1：SPI 忙于通信或者发送缓冲区不为空

此标志由硬件置 1 和清零。

注： BSY 标志必须谨慎使用：请参见第 35.4.10 节：SPI 状态标志和第 1134 页的禁止 SPI 的步骤。

位 6 **OVR**：上溢标志 (Overrun flag)

0：未发生上溢

1：发生上溢

此标志由硬件置 1，由软件序列复位。

位 5 **MODF**：模式故障 (Mode fault)

0：未发生模式故障

1：发生模式故障

此标志由硬件置 1，由软件序列复位。有关软件序列，请参见第 1143 页的模式故障 (MODF) 一节。

位 4 **CRCERR**：CRC 错误标志 (CRC error flag)

0：接收到的 CRC 值与 SPIx_RXCRCR 值匹配

1：接收到的 CRC 值与 SPIx_RXCRCR 值不匹配

注： 此标志由硬件置 1，通过软件写入 0 来清零。

位 1 **TXE**：发送缓冲区为空 (Transmit buffer empty)

0：发送缓冲区非空

1：发送缓冲区为空

位 0 **RXNE**：接收缓冲区非空 (Receive buffer not empty)

0：接收缓冲区为空

1：接收缓冲区非空

35.6.4 SPI 数据寄存器 (SPIx_DR)

SPI data register

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
DR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **DR[15:0]**：数据寄存器 (Data register)

已接收或者要发送的数据

数据寄存器用于连接 Rx 和 Tx FIFO。读取数据寄存器时，将访问 RxFIFO；而写入数据寄存器时，将访问 TxFIFO（请参见第 35.4.9 节：数据发送和接收过程）。

注： 数据始终右对齐。写入寄存器时将忽略未使用位，读取寄存器时会将未使用位读为 0。Rx 阈值设置必须始终与当前使用的读访问相符。

35.6.5 SPI CRC 多项式寄存器 (SPIx_CRCPR)

SPI CRC polynomial register

偏移地址：0x10

复位值：0x0007

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

位 15:0 **CRCPOLY[15:0]**：CRC 多项式寄存器 (CRC polynomial register)

此寄存器包含用于 CRC 计算的多项式。

CRC 多项式 (0x0007) 是此寄存器的复位值。可根据需要配置另一个多项式。

注： 多项式值只应为奇数。不支持任何偶数值。

35.6.6 SPI 接收 CRC 寄存器 (SPIx_RXCRCR)

SPI Rx CRC register

偏移地址：0x14

复位值：0x0000

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

位 15:0 **RXCRC[15:0]**：接收 CRC 寄存器 (Rx CRC register)

使能 CRC 计算后，RXCRC[15:0] 位将包含接收字节序列的 CRC 值。当 SPIx_CR1 寄存器中的 CRCEN 位写入 1 时，此寄存器复位。CRC 通过 SPIx_CRCPR 寄存器中编程的多项式连续计算。

CRC 帧格式设置为 8 位长度（SPIx_CR1 的 CRCL 位清零）时，仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。

选择 16 位 CRC 帧格式（SPIx_CR1 寄存器的 CRCL 位置 1）时，考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。

当 BSY 标志置 1 时，读取此寄存器可能返回一个不正确的值。

35.6.7 SPI 发送 CRC 寄存器 (SPIx_TXCRCR)

SPI Tx CRC register

偏移地址：0x18

复位值：0x0000

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

位 15:0 **TXCRC[15:0]**：发送 CRC 寄存器 (Tx CRC register)

使能 CRC 计算后，TXCRC[7:0] 位将包含发送字节序列的 CRC 值。当 SPIx_CR1 寄存器中的 CRCEN 位写入 1 时，此寄存器复位。CRC 通过 SPIx_CRCPR 寄存器中编程的多项式连续计算。

CRC 帧格式设置为 8 位长度（SPIx_CR1 的 CRCL 位清零）时，仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。

选择 16 位 CRC 帧格式（SPIx_CR1 寄存器的 CRCL 位置 1）时，考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。

当 BSY 标志置 1 时，读取此寄存器可能返回一个不正确的值。

35.6.8 SPI 寄存器映射

表 207 给出了 SPI 寄存器映射和复位值。

表 207. SPI 寄存器映射和复位值

偏移	寄存器	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
0x00	**SPIx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BIDIMODE	BIDIOE	CRCEN	CRCNEXT	CRCL	RXONLY	SSM	SSI	LSBFIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**SPIx_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LDMA_TX	LDMA_RX	FRXTH	DS[3:0]				TXEIE	RXNEIE	ERRIE	FRF	NSSP	SSOE	TXDMAEN	RXDMAEN
	Reset value																		0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
0x08	**SPIx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTLVL[1:0]		FRLVL[1:0]		FRE	BSY	OVR	MODF	CRCERR			TXE	RXNE
	Reset value																				0	0	0	0	0	0	0	0	0			1	0
0x0C	**SPIx_DR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**SPIx_CRCPR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCPOLY[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x14	**SPIx_RXCRCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXCRC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**SPIx_TXCRCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXCRC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

36 串行音频接口 (SAI)

36.1 简介

SAI 接口（串行音频接口）灵活性高、配置多样，可支持多种音频协议。该接口适用于许多立体声或单声道应用。例如，它可配置为支持 I2S 标准、LSB 或 MSB 对齐、PCM/DSP、TDM 和 AC'97 等协议。将音频模块配置为发送器时，SAI 接口可提供 SPDIF 输出。

SAI 通过两个完全独立的音频子模块来实现这种灵活性和可重配置性。每个模块都有自己的时钟发生器和 I/O 线控制器。

SAI 可以配置为主模式或配置为从模式。音频子模块既可作为接收器，又可作为发送器；既可与另一模块同步，又可以不同步。

36.2 SAI 主要特性

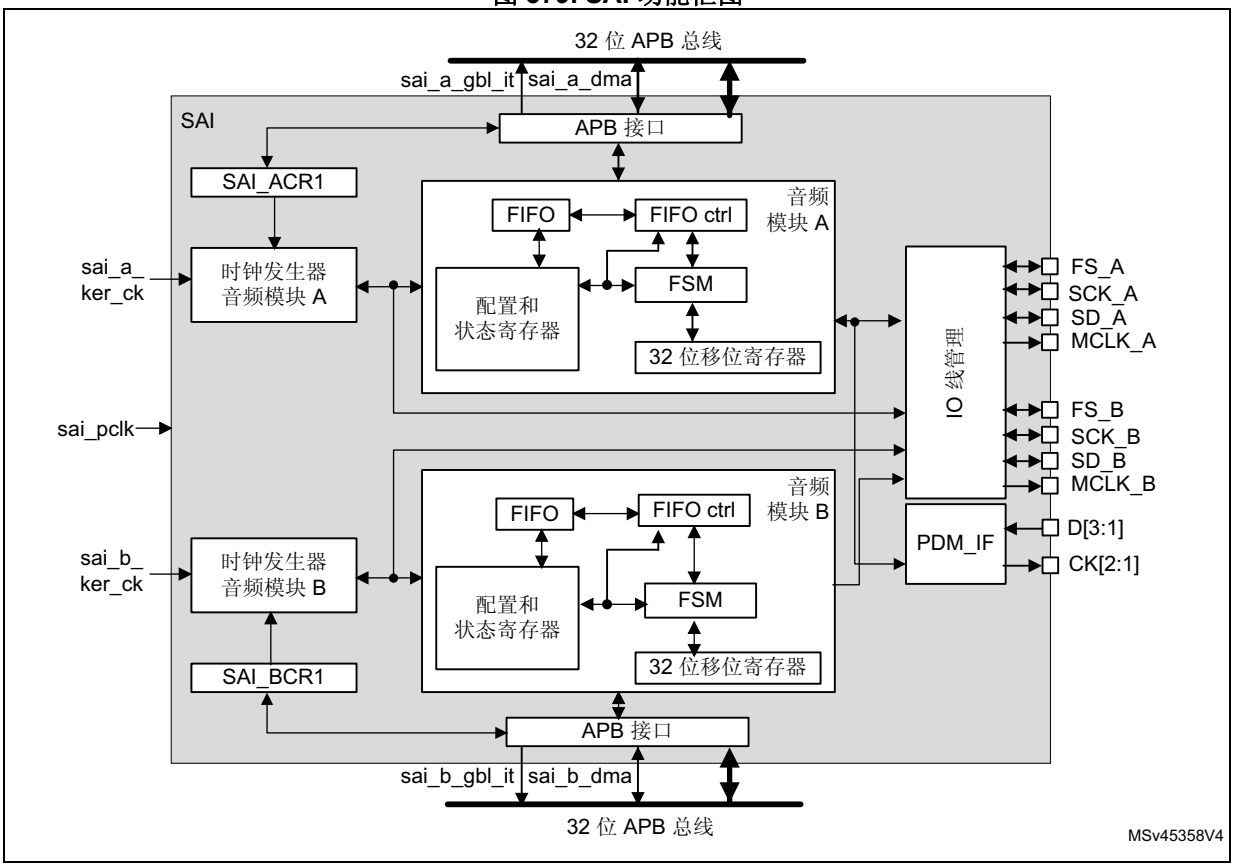
- 具有两个独立的音频子模块，子模块既可作为接收器，也可作为发送器，并带有自身的 FIFO。
- 每个音频子模块集成多达 8 个字的 FIFO。
- 两个音频子模块间可以是同步或异步模式。
- 两个音频子模块的主/从配置相互独立。
- 当两个音频子模块都配置为主模式时，每个子模块的时钟发生器产生独立的音频采样频率。
- 数据大小可配置：8 位、10 位、16 位、20 位、24 位或 32 位。
- 音频协议：I2S、LSB 或 MSB 对齐、PCM/DSP、TDM、AC'97。
- PDM 接口，支持多达 4 对麦克风。
- 如果需要，可提供 SPDIF 输出。
- 高达 16 个大小可配置的 Slot。
- 每帧的位数可配置。
- 帧同步有效电平可配置（偏移、位长、电平）。
- 可配置 Slot 中第一个有效位的位置。
- 支持 LSB 或 MSB 数据传输。
- 支持静音模式。
- 具有立体声/单声道音频帧功能。
- 通信时钟选通边沿可配置 (SCK)。
- 错误标志对应相应中断（分别使能时）。
  - 上溢和下溢检测
  - 从模式下的帧同步信号提前检测
  - 从模式下的帧同步信号滞后检测
  - 接收时编码解码器未针对 AC'97 模式就绪
- 支持如下中断源（使能时）：
  - 错误
  - FIFO 请求
- DMA 接口有 2 个通道。

36.3 SAI 功能说明

36.3.1 SAI 框图

图 379 给出了 SAI 框图，*表 208* 和*表 209* 列出了 SAI 的内部信号和外部信号。

图 379. SAI 功能框图



SAI 主要由两个各自带有时钟发生器的音频子模块组成。每个音频模块集成一个 32 位移位寄存器，该寄存器由模块自身的功能状态机控制。数据存储和读取都是通过专用的 FIFO 来完成。FIFO 可通过 CPU 访问，也可通过 DMA 访问以减轻 CPU 的通信负担。每个音频模块是独立的。这两个音频子模块可彼此同步。

I/O 线控制器管理 SAI 中指定音频模块的 4 个专用引脚（SD、SCK、FS、MCLK）。如果将两个子模块声明为同步模块，则可以共用其中某些引脚，从而留出一些引脚用作通用 I/O。MCLK 引脚是否可用作输出引脚取决于实际应用和解码要求以及音频模块是否配置为主模式。

如果将一个 SAI 配置为与另外那个 SAI 模块同步运行，可以释放更多 I/O（引脚 SD_x 除外）。

可配置功能状态机来处理多种音频协议。一些寄存器用于设置所需协议（音频帧波形发生器）。

音频子模块在主模式或从模式下均可用作发送器或接收器。主模式意味着从 SAI 生成 SCK_x 位时钟和帧同步信号，而从模式则意味着 SCK_x 位时钟和帧同步信号来自另一外部或内部主器件。在特殊情况下，FS 信号方向与主模式或从模式定义不直接相关。在 AC'97 协议中，即使 SAI（链接控制器）设置为消耗 SCK 时钟，FS 信号也会是 SAI 输出（从模式下也是如此）。

注： *为方便阅读此部分，符号 SAI_x 指 SAI_A 或 SAI_B，其中“x”代表 SAI A 子模块或 SAI B 子模块。*

36.3.2 SAI 引脚和内部信号

表 208. SAI 内部输入/输出信号

内部信号名称	信号类型	说明
sai_a_gbl_it/ sai_b_gbl_it	输出	音频模块 A 和 B 全局中断。
sai_a_dma、 sai_b_dma	输入/输出	音频模块 A 和 B DMA 确认与请求。
sai_a_ker_ck/ sai_b_ker_ck	输入	音频模块 A/B 内核时钟。
sai_pclk	输入	APB 时钟。

表 209. SAI 输入/输出引脚

名称	信号类型	注释
SAI_SCK_A/B	输入/输出	音频模块 A/B 位时钟。
SAI_MCLK_A/B	输出	音频模块 A/B 主时钟。
SAI_SD_A/B	输入/输出	用于模块 A/B 的数据线。
SAI_FS_A/B	输入/输出	用于音频模块 A/B 的帧同步线。
SAI_CK[2:1]	输出	PDM 比特流时钟。
SAI_D[3:1]	输入	PDM 比特流数据。

36.3.3 SAI 的主要模式

SAI 的每个音频子模块均可通过所选音频模块的 SAI_xCR1 寄存器中的 MODE 位配置为主模式或从模式。

主模式

在主模式下，SAI 会向连接的外部器件提供时钟信号：

- 位时钟和帧同步分别在引脚 SCK_x 和 FS_x 上输出。
- 如果需要，SAI 也可以在 MCLK_x 引脚上生成主时钟。

SCK_x、FS_x 和 MCLK_x 均配置为输出。

从模式

SAI 从外部器件接收时钟信号。

- 如果将 SAI 子模块配置为异步模式，则 SCK_x 引脚和 FS_x 引脚将被配置为输入。
- 如果将 SAI 子模块配置为与另一个音频子模块同步运行，则会释放相应的 SCK_x 引脚和 FS_x 引脚以用作通用 I/O。

在从模式下，不使用 MCLK_x 引脚，可将其分配给其他功能。

建议在使能主器件前先使能从器件。

配置和使能 SAI 模式

可通过相应音频模块的 SAI_xCR1 寄存器中的 MODE 位将每个音频子模块独立定义为发送器或接收器。为此，SAI_SD_x 引脚将分别被配置为输出或输入。

一个 SAI 外设下的两个子模块若都工作为主模式，此时两个子模块各自的 MCLK 和 SCK 时钟频率各不相同，在这种情况下，必须将这两个音频模块配置为异步模式。

SAI 中的每个音频模块均通过 SAI_xCR1 寄存器中的 SAIEN 位使能。在从模式下，此位一经激活，发送器或接收器便会对时钟线、数据线和同步线上的活动敏感。

在主 TX 模式下，即使 FIFO 中没有数据，使能音频模块也会立即为外部从模块产生位时钟，但 FS 信号的产生受 FIFO 中是否存在数据的控制。FIFO 接收到要发送的第一个数据后，此数据将输出到外部从模块。如果 FIFO 中没有要发送的数据，则随后将在音频帧中传送值 0，并会产生一个下溢标志。

在从模式下，使能音频模块时和检测到 SOF 位时开始音频帧。

在从 TX 模式下，使能音频模块后的第一个帧上不可能出现下溢事件，因为此时的强制操作顺序如下：

1. 通过软件或 DMA 写入 SAI_xDR。
2. 直到等到 FIFO 阈值 (FLH) 非 0（FIFO 空）。
3. 使能音频模块为从发送模式。

36.3.4 SAI 同步模式

SAI 子时钟 A 和 B 可以同步。

内部同步

SAI 中的音频子模块可以配置为和另一个音频子模块同步运行。在这种情况下，二者将共用位时钟和帧同步信号，以减少通信时占用的外部引脚数。配置为同步模式的音频模块将释放其 SCK_x、FS_x 和 MCLK_x 引脚以用作 GPIO，而配置为异步模式的音频模块将使用其 FS_x、SCK_x 和 MCLK_x I/O 引脚（如果该音频模块被配置为主模块）。

通常，同步模式下的音频模块可用于在全双工模式下配置 SAI。两个音频模块中的一个可配置为主模块，另一个为从模块；也可将两个均配置为从模块；一个模块为异步模块（SAI_xCR1 中的相应位 SYNCEN[1:0] 设置为 00），另一个为同步模块（SAI_xCR1 中的相应位 SYNCEN[1:0] 设置为 01）。

注： *由于存在内部重新同步阶段，因此 PCLK APB 频率必须大于位时钟频率的两倍。*

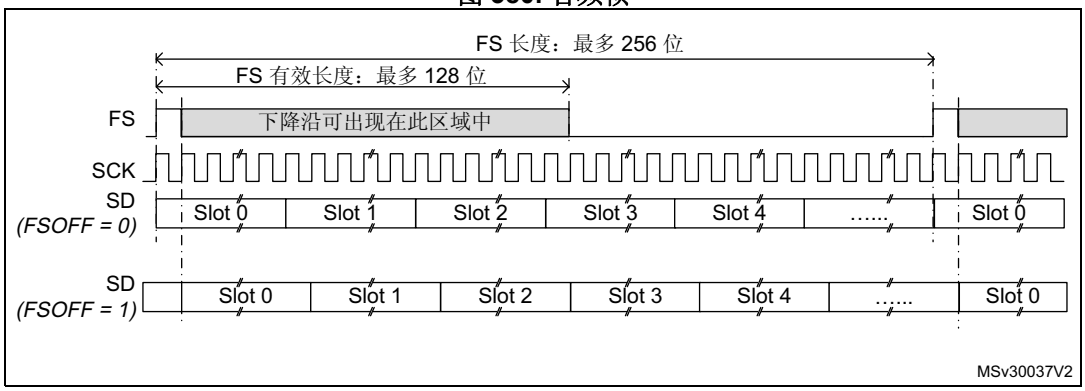
36.3.5 音频数据大小

通过配置 SAI_xCR1 寄存器中的 DS[2:0] 位，配置音频帧的数据大小。数据大小可以是 8 位、10 位、16 位、20 位、24 位或 32 位。在传输期间，将首先发送数据的 MSB 或 LSB，具体取决于 SAI_xCR1 寄存器中的 LSBFIRST 位的配置。

36.3.6 帧同步

FS 信号用作音频帧中的帧同步信号 (SOF)。此信号的波形完全可配置，以在帧同步时，支持各种具有特殊规格的音频协议。这一可重新配置性通过寄存器 SAI_xFRCR 来实现。*图 380* 说明了这种灵活性。

图 380. 音频帧



在 AC'97 模式或 SPDIF 模式下（SAI_xCR1 寄存器中的位 PRTCFG[1:0] = 10 或 PRTCFG[1:0] = 01），帧同步信号的波形被强制配置为支持 AC'97 协议。SAI_xFRCR 寄存器值被忽略。

每个音频模块相互独立，因此均需要特定的配置。

帧长度

- 主模式

  将 SAI_xFRCR 寄存器中的 FRL[7:0] 位域置 1，可将音频帧的长度配置为最长 256 个位时钟周期。

  如果帧长度大于为该帧声明的 Slot 数，则要发送的剩余位将用 0 填充，或者 SD 线将释放为高阻态，具体取决于 SAI_xCR2 寄存器中的位 TRIS 的状态（见 *FS 信号的作用*）。在接收模式下，剩余位被忽略。

  如果将位 NODIV 清零，(FRL+1) 必须等于 2 的几次幂（从 8 到 256）以确保音频帧的每个位时钟周期都包含整数个 MCLK 脉冲。

  如果将位 NODIV 置 1，则 (FRL+1) 字段可采用从 8 到 256 的任意值。请参见*第 36.3.8 节：SAI 时钟发生器*。

- 从模式

  音频帧的长度主要用于指定由外部主模块向从模块发送的每个音频帧的位时钟周期数。它主要用于从主模块中检测音频帧传输期间出现的提前或滞后帧同步信号。这种情况下会出现错误。有关详细信息，请参见*第 36.3.14 节：错误标志*。

  在从模式下，SAI_xFRCR 寄存器中的 FRL[7:0] 位的配置不受任何限制。

帧中的位数等于 FRL[7:0] + 1。

音频帧中要传输的最小位数为 8。

帧同步极性

SAI_xFRCR 寄存器中的 FSPOL 位用于设置 FS 引脚的有效极性，通过该极性来启动帧。SOF 信号对边沿敏感。

在从模式下，音频模块等待一个有效帧来启动发送或接收。SOF 信号与此信号同步。只有通信期间未检测到 SOF 信号并且 SOF 信号与预期 SOF 信号相同时，帧同步极性才有效（请参见*第 36.3.14 节：错误标志*）。

在主模式下，每次音频帧完成时均会发送帧同步信号，直至 SAI_xCR1 寄存器中的 SAIEN 位清零。如果前一个音频帧结束时 FIFO 中不存在数据，则将按照*第 36.3.14 节：错误标志*所述管理下溢条件，但音频通信流不会中断。

帧同步有效电平长度

SAI_xFRCR 寄存器的 FSALL[6:0] 位用于配置帧同步信号的有效电平的长度。该长度可设置为 1 到 128 个位时钟周期。

例如，有效长度在 I2S、LSB 或 MSB 对齐模式下为帧长的一半，在 PCM/DSP 或 TDM 模式下为 1 位。

帧同步偏移

基于应用中支持的音频协议（例如 I2S 标准协议和 MSB 对齐协议），可以在发送音频帧的最后一位或第一位时将帧同步信号置为有效。通过 SAI_xFRCR 寄存器中的 FSOFF 位选择两个配置之一。

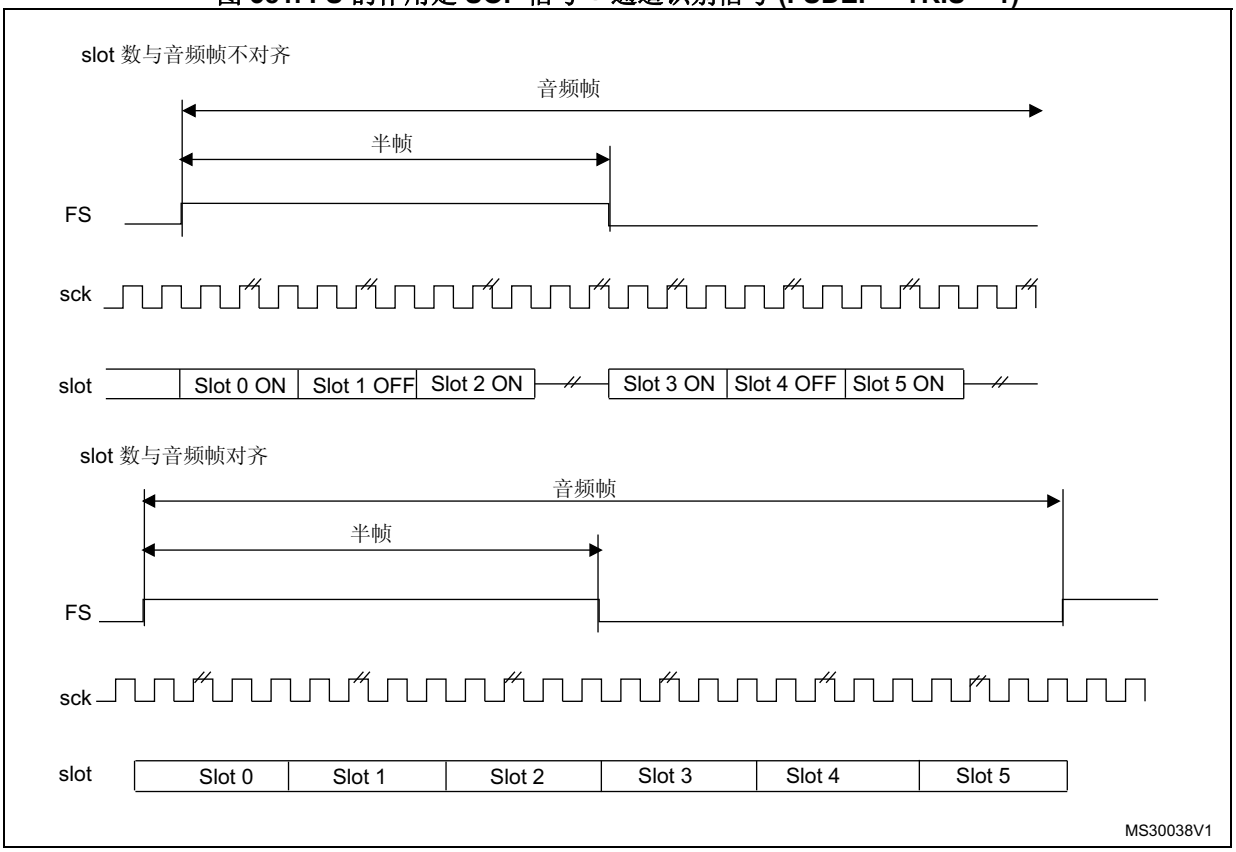
FS 信号的作用

FS 信号可以有不同的含义，具体取决于 FS 的功能。SAI_xFRCR 寄存器中的 FSDEF 位用于选择 FS 信号的含义。

- 0：SOF 信号，例如 PCM/DSP、TDM、AC’97 和音频协议。
- 1：音频帧内的 SOF 信号和通道识别信号，例如 I2S、MSB 或 LSB 对齐协议。

当 FS 信号被视为帧内的 SOF 信号和通道识别信号时，声明的 Slot 数必须是一半用于左通道，一半用于右通道。如果半个音频帧上的位时钟周期数大于某个通道的专用 Slot 数，若 TRIS = 0，则 SAI_xCR2 寄存器中的剩余位时钟周期将发送 0。否则若 TRIS = 1，SD 线将释放为高阻态。接收时，直到通道发生变化，才会考虑剩余位时钟。

图 381. FS 的作用是 SOF 信号 + 通道识别信号 (FSDEF = TRIS = 1)

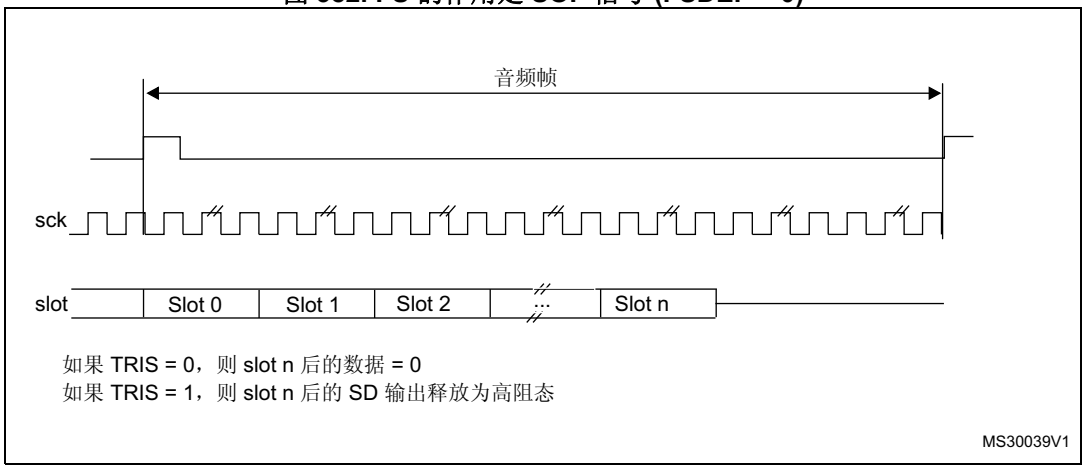


1. 帧长度应为偶数。

如果 SAI_xFRCR 中的 FSDEF 位保持清零状态，则 FS 信号等效于 SOF 信号，如果 SAI_xSLOTR 中的 NBSLOT[3:0] 位定义的 Slot 数乘以 SAI_xSLOTR 中的 SLOTSZ[1:0] 位配置的 Slot 位数所得的结果小于帧大小（SAI_xFRCR 寄存器中的 FRL[7:0] 位），则：

- 如果 SAI_xCR2 寄存器中的 TRIS = 0，则最后一个 Slot 后的剩余位将强制为 0，直至发送器的帧结束，
- 如果 TRIS = 1，则传输这些剩余位时，数据线将释放为高阻态。在接收模式下，这些位被丢弃。

图 382. FS 的作用是 SOF 信号 (FSDEF = 0)



在发送模式下将音频模块配置为获取 SD 线上的 SPDIF 输出时，将不使用 FS 信号。相应的 FS I/O 会被释放，留作他用。

36.3.7 Slot 配置

Slot 是音频帧中的基本元素。音频帧中的 Slot 数等于 NBSLOT[3:0] + 1。

每个音频帧的最大 Slot 数固定为 16。

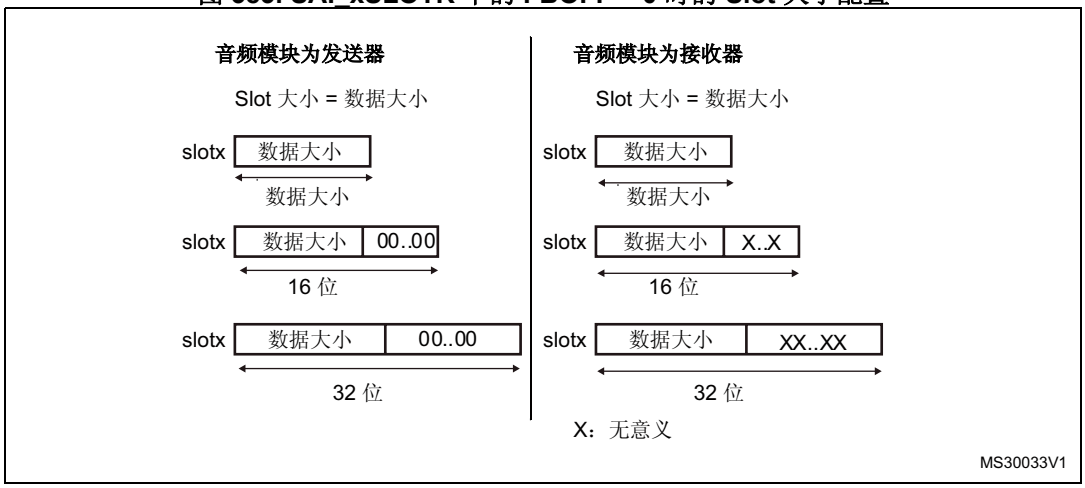
对于 AC'97 协议或 SPDIF 协议（位 PRTCFG[1:0] = 10 或 PRTCFG[1:0] = 01 时），Slot 数自动按照协议规范设置，NBSLOT[3:0] 的值被忽略。

通过设置 SAI_xSLOTR 寄存器的 SLOTEN[15:0] 位，可将各个 Slot 定义为有效 Slot 或无效 Slot。

传输一个无效 Slot 时，SD 数据线将强制为 0 或释放为高阻态，具体取决于 TRIS 位在发送模式下的配置（请参见*无效 Slot 上的输出数据线管理*）。在接收模式下，该 Slot 对应的接收值将被忽略。结果，不会有 FIFO 访问，也不会有与此无效 Slot 状态相关的 FIFO 读/写请求。

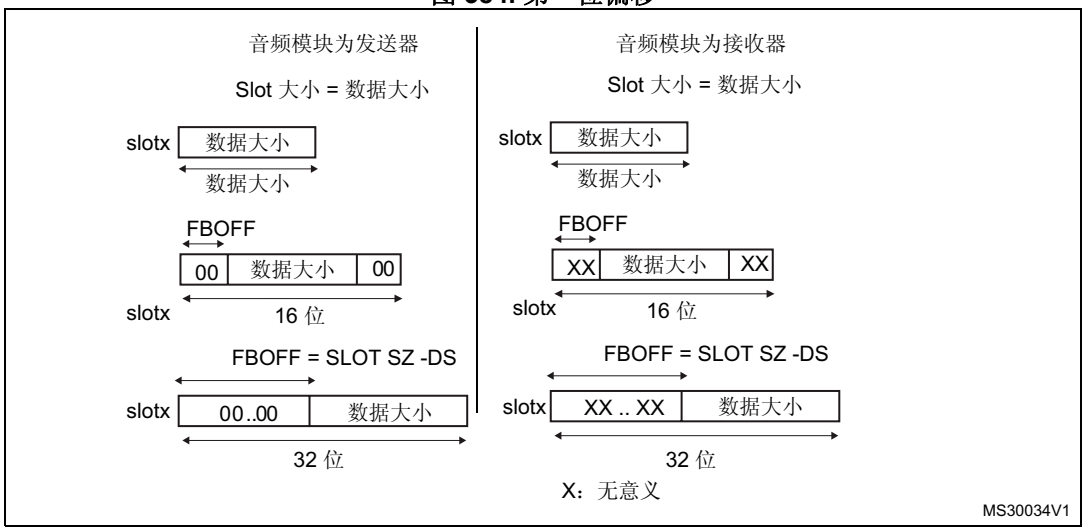
Slot 大小也可配置，如*图 383* 所示。通过 SAI_xSLOTR 寄存器中的 SLOTSZ[1:0] 位域来选择 Slot 的大小，该大小将应用于音频帧中的所有 Slot。

图 383. SAI_xSLOTR 中的 FBOFF = 0 时的 Slot 大小配置



可以选择 Slot 内要传输的第一个数据位的位置，此偏移通过 SAI_xSLOTR 寄存器中的 FBOFF[4:0] 位配置。在发送模式下，将从 Slot 开始时注入 0 值，直至到达此偏移位置。接收时，偏移阶段中的位被忽略。此特性适用于 LSB 对齐协议（如果偏移等于 Slot 大小减去数据大小）。

图 384. 第一位偏移



为避免 SAI 出现错误，强制要求遵循以下条件：

FBOFF ≤ (SLOTSZ - DS)

DS ≤ SLOTSZ

NBSLOT x SLOTSZ ≤ FRL（帧长度）

SAI_xFRCR 寄存器中的 FSDEF 位置 1 时，Slot 数必须为偶数。

在 AC'97 和 SPDIF 协议（位 PRTCFG[1:0] = 10 或 PRTCFG[1:0] = 01）中，Slot 大小按照 *第 36.3.11 节：AC'97 链路控制器*中的定义自动设置。

36.3.8 SAI 时钟发生器

每个音频模块都具有自己的时钟发生器。时钟发生器通过 sai_x_ker_ck 构建主时钟 (MCLK_x) 和位时钟 (SCK_x) 信号。sai_x_ker_ck 时钟由产品的时钟控制器 (RCC) 提供。

主时钟 (MCLK_x) 的生成

当音频模块定义为主模式或从模式时，时钟发生器提供主时钟 (MCLK_x)。即使相应模块的 SAIEN 位置 0，只要 MCKEN 位置 1，就会生成主时钟。如果 MCLK_x 时钟用作外部音频设备的系统时钟，则此功能非常有用，因为它允许在激活音频流之前生成 MCLK_x。

要在传输音频采样之前在 MCLK_x 输出上生成主时钟，用户应用必须遵循以下顺序：

1. 检查 SAIEN 是否为 0。
2. 将 MCKDIV[5:0] 分频器编程为所需的值。
3. 将 MCKEN 位置 1。
4. 稍后，应用程序可以配置 SAI 的其他部分，并将 SAIEN 位置 1 以开始传输音频采样。

为避免对 MCLK_x 输出上生成的时钟造成干扰，建议不要执行以下操作：

- 当 MCKEN = 1 时更改 MCKDIV
- 当 SAIEN = 1 时将 MCKEN 置 0

当通过 MCKEN 位来打开或者关闭 MCLK_x 时 (SAIEN=0)，SAI 此外设能保证 MCLK_x 输出没有杂波。

表 210 列出了 MCLK_x 激活条件。

表 210. MCLK_x 激活条件

MCLKEN	NODIV	模块 x 的 SAIEN	MCLK_x
0	X	0	禁止
1			启用
0	1	1	禁止
1			启用
X	0		启用

注： *当 MCLKEN 置 1 时，MCLK_x 也可以在 AC'97 模式下生成。*

位时钟 (SCK_x) 的生成

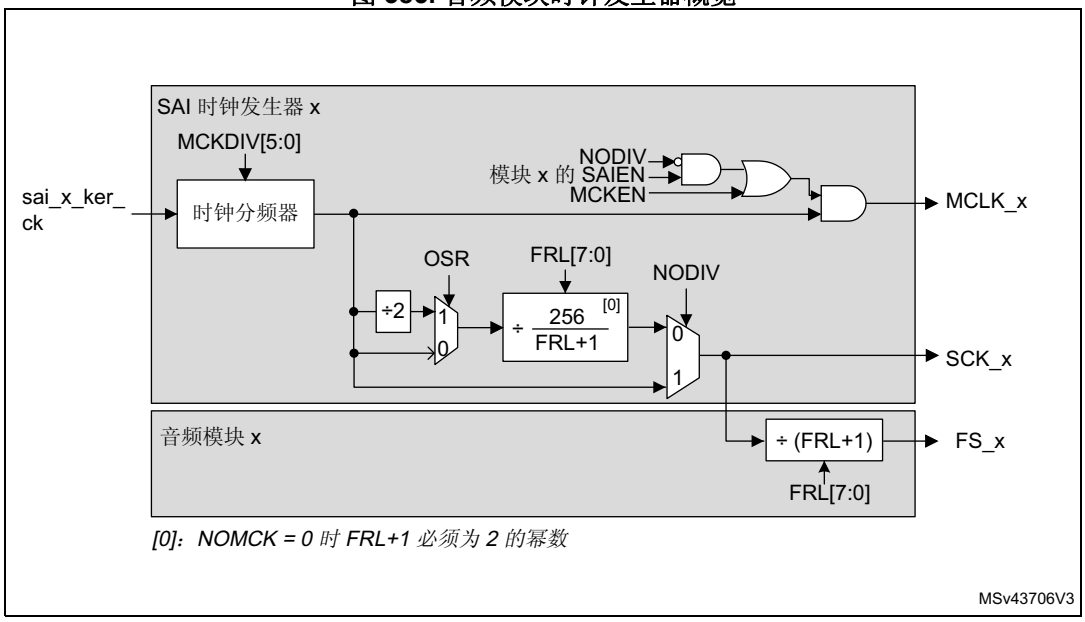
当音频模块定义为主模式时，时钟发生器提供位时钟 (SCK_x)。帧同步 (FS_x) 还可源自时钟发生器提供的信号。

在从模式下，将忽略 NODIV 和 OSR 字段的值，并且不生成 SCK_x 时钟。

SCK_x 的位时钟选通边沿可以通过 CKSTR 字段进行配置，该边沿在主模式和从模式下均可正常工作。

图 385 给出了音频模块时钟发生器的架构。

图 385. 音频模块时钟发生器概览



必须使用 NODIV 位强制主时钟 (MCLK_x) 频率和帧同步 (FS_x) 频率之间的比值为 256 或 512。

- 如果 NODIV 置 0，帧同步与主时钟之间的频率比为固定值 512 或 256（根据 OSR 值），但帧长度必须为 2 的幂数。下面进行了详细说明。
- 如果 NODIV 置 1，则应用可以通过 MCKDIV 调整位时钟 (SCK_x) 的频率。此外，只要帧长度大于或等于 8（即 FRL[7:0] > 6），帧长度值就没有限制。帧同步频率取决于 MCKDIV 和帧长度 (FRL[7:0])。在这种情况下，MCLK_x 的频率等于 SCK_x。

NODIV、MCKEN、SAIEN、OVR、CKSTR 和 MCKDIV[5:0] 位属于 SAI_xCR1 寄存器，而 FRL[7:0] 属于 SAI_xFRCR。

NODIV = 0 时的时钟发生器编程

此时，MCLK_x 频率将为：

- 如果 OSR = 0，$F_{MCLK\_x} = 256 \times F_{FS\_x}$
- 如果 OSR = 1，$F_{MCLK\_x} = 512 \times F_{FS\_x}$

MCKDIV 不为 0 时，MCLK_x 频率由如下公式计算得出：

$$F_{MCLK\_x} = \frac{F_{sai\_x\_ker\_ck}}{MCKDIV}$$

帧同步频率由如下公式计算得出：

$$F_{FS\_x} = \frac{F_{sai\_x\_ker\_ck}}{MCKDIV \times (OSR + 1) \times 256}$$

位时钟频率 (SCK_x) 由如下公式计算得出：

$$F_{SCK\_x} = \frac{F_{sai\_x\_ker\_ck} \times (FRL + 1)}{MCKDIV \times (OSR + 1) \times 256}$$

注： *当 NODIV 等于 0 时，(FRL+1) 必须为 2 的幂数。此外，(FRL+1) 必须介于 8 到 256 之间。(FRL +1) 代表音频帧中的位时钟数。*

MCKDIV 分频比为奇数时，MCLK 的占空比不会为 50%。如果 MCKDIV 为奇数、OSR 等于 0 或 (FRL+1) = 2^8，位时钟信号 (SCK_x) 的占空比也可以不是 50%。

建议将 MCKDIV 编程为偶数值或较大的值（大于 10）。

请注意，MCKDIV = 0 时得到的结果与 MCKDIV = 1 时得到的结果相同。

NODIV = 1 时的时钟发生器编程

MCKDIV 不为 0 时，位时钟 (SCK_x) 的频率由如下公式计算得出：

$$F_{SCK\_x} = F_{MCLK\_x} = \frac{F_{sai\_x\_ker\_ck}}{MCKDIV}$$

帧同步 (FS_x) 的频率由如下公式计算得出：

$$F_{FS\_x} = \frac{F_{sai\_x\_ker\_ck}}{(FRL + 1) \times MCKDIV}$$

注： *NODIV 置 1 时，(FRL+1) 可采用 8 到 256 之间的任意值。*

请注意，MCKDIV = 0 时得到的结果与 MCKDIV = 1 时得到的结果相同。

时钟发生器编程示例

表 211 列出了针对 48 kHz、96 kHz 和 192 kHz 频率的编程示例。

表 211. 时钟发生器编程示例

输入 sai_x_ker_ck 时钟频率	MCLK	F_{MCLK}/F_{FS}	FRL (1)	OSR	NODIV	MCKEN	MCKDIV[5:0]	音频采样频率 (F_{FS})
98.304 MHz	是	512	2^N-1	1	0	1	0 或 1	192 kHz
		512	2^N-1	1	0	1	2	96 kHz
		512	2^N-1	1	0	1	4	48 kHz
		256	2^N-1	0	0	1	2	192 kHz
		256	2^N-1	0	0	1	4	96 kHz
		256	2^N-1	0	0	1	8	48 kHz
	否	-	63	-	1	0	8	192 kHz
		-	63	-	1	0	16	96 kHz
		-	63	-	1	0	32	48 kHz

1. N 表示介于 3 到 8 之间的整数值。

36.3.9 内部 FIFO

SAI 中的每个音频模块都有自己的 FIFO。根据模块是定义为发送器还是接收器，其 FIFO 可相应地用来读取或写入。因此只存在一个 FIFO 请求与 SAI_xSR 寄存器中的 FREQ 位相关。

如果 SAI_xIM 寄存器中的 FREQIE 位使能，将产生中断。这取决于：

- FIFO 阈值设置（SAI_xCR2 中的 FLVL 位）
- 通信方向（发送器或接收器）。请参见*在发送模式下产生中断*和*在接收模式下产生中断*

在发送模式下产生中断

根据发送模式下的 FIFO 配置产生中断：

- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 为空时（FTH[2:0] 置为 0b000），如果 SAI_xDR 寄存器中没有数据（SAI_xSR 中的 FLVL[2:0] 位小于 001b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当 FIFO 不再为空时（SAI_xSR 中的 FLVL[2:0] 位不是 0b000），即 FIFO 中存储一个或多个数据时，此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之一满时（FTH[2:0] 置为 001b），如果不到四分之一的 FIFO 包含数据（SAI_xSR 中的 FLVL[2:0] 位小于 0b010），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当至少四分之一的 FIFO 包含数据时（SAI_xSR 中的 FLVL[2:0] 位大于等于 0b010），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 半满时（FTH[2:0] 置为 0b010），如果不到一半的 FIFO 包含数据（SAI_xSR 中的 FLVL[2:0] 位小于 011b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当至少一半的 FIFO 包含数据时（SAI_xSR 中的 FLVL[2:0] 位大于或等于 011b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。

- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之三满时（FTH[2:0] 置为 011b），如果不到四分之三的 FIFO 包含数据（SAI_xSR 中的 FLVL[2:0] 位小于 0b100），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当至少四分之三的 FIFO 包含数据时（SAI_xSR 中的 FLVL[2:0] 位大于或等于 0b100），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 为满时（FTH[2:0] 置为 0b100），如果 FIFO 不满（SAI_xSR 中的 FLVL[2:0] 位小于 101b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当 FIFO 已满时（SAI_xSR 中的 FLVL[2:0] 位等于值 101b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。

在接收模式下产生中断

根据接收模式下的 FIFO 配置产生中断：

- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 为空时（FTH[2:0] 置为 0b000），如果 SAI_xDR 寄存器中至少有一个数据（SAI_xSR 中的 FLVL[2:0] 位大于或等于 001b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当 FIFO 变为空时（SAI_xSR 中的 FLVL[2:0] 位等于 0b000），即 FIFO 中未存储数据时，此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之一满时（FTH[2:0] 置为 001b），如果至少有四分之一的 FIFO 数据单元可用（SAI_xSR 中的 FLVL[2:0] 位大于或等于 0b010），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当不到四分之一的 FIFO 数据单元可用时（SAI_xSR 中的 FLVL[2:0] 位小于 0b010），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 半满时（FTH[2:0] 置为值 0b010），如果至少有一半的 FIFO 数据单元可用（SAI_xSR 中的 FLVL[2:0] 位大于或等于 011b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当不到一半的 FIFO 数据单元可用时（SAI_xSR 中的 FLVL[2:0] 位小于 011b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之三满时（FTH[2:0] 置为值 011b），如果至少有四分之三的 FIFO 数据单元可用（SAI_xSR 中的 FLVL[2:0] 位大于或等于 0b100），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当不到四分之三的 FIFO 数据单元可用时（SAI_xSR 中的 FLVL[2:0] 位小于 0b100），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 满时（FTH[2:0] 置为 0b100），如果 FIFO 已满（SAI_xSR 中的 FLVL[2:0] 位等于 101b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当 FIFO 不满时（SAI_xSR 中的 FLVL[2:0] 位小于 101b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。

与中断的产生类似，如果 SAI_xCR1 寄存器中的 DMAEN 位置 1，则 SAI 可使用 DMA。FREQ 位的有效机制与上述 FREQIE 的中断产生机制相同。

每个 FIFO 均是一个 8 字 FIFO。无论访问大小为何，每次对 FIFO 进行读写时，均针对 1 个字的 FIFO 单元进行操作。每个 FIFO 字包含一个音频 Slot。每次访问 SAI_xDR 寄存器后，FIFO 指针递增一个字。

数据应以右对齐方式写入 SAI_xDR。

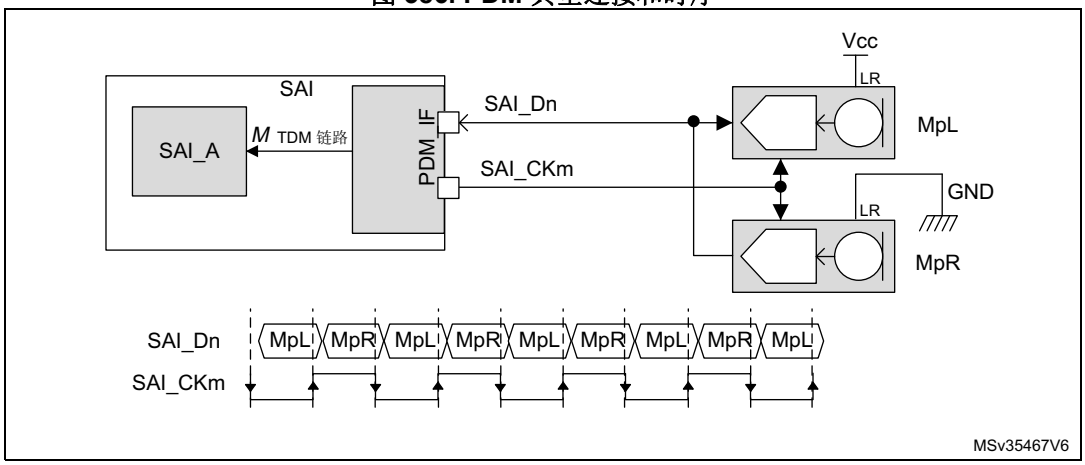
接收到的数据将以右对齐方式存储到 SAI_xDR。

当禁止 SAI 时将 SAI_xCR2 寄存器中的 FFLUSH 位置 1 时，可重新初始化 FIFO 指针。如果 FFLUSH 在 SAI 使能情况下置 1，则 FIFO 中的数据将自动清除。

36.3.10 PDM 接口

PDM（脉冲密度调制）接口旨在支持数字麦克风，最多可并联 4 对数字麦克风。*图 386* 所示为一对数字麦克风通过 PDM 接口实现的典型连接。两个麦克风共用同一比特流时钟和数据线。借助配置引脚 (LR)，其中一个麦克风可在 SAI_CK[m] 上升沿提供有效数据，而另一个麦克风则在 SAI_CK[m] 下降沿提供有效数据（m 表示时钟线的数量）。

图 386. PDM 典型连接和时序



1. **n** 表示数据线的数量，**p** 表示麦克风对的数量。

PDM 功能旨在与配置为 TDM 主模式的 SAI_A 子模块配合使用。它不能与 SAI_B 子模块搭配使用。PDM 接口使用由 SAI_A 的 TDM 接口提供的时序信号，并对其进行调整来生成时钟 (SAI_CK[m])。

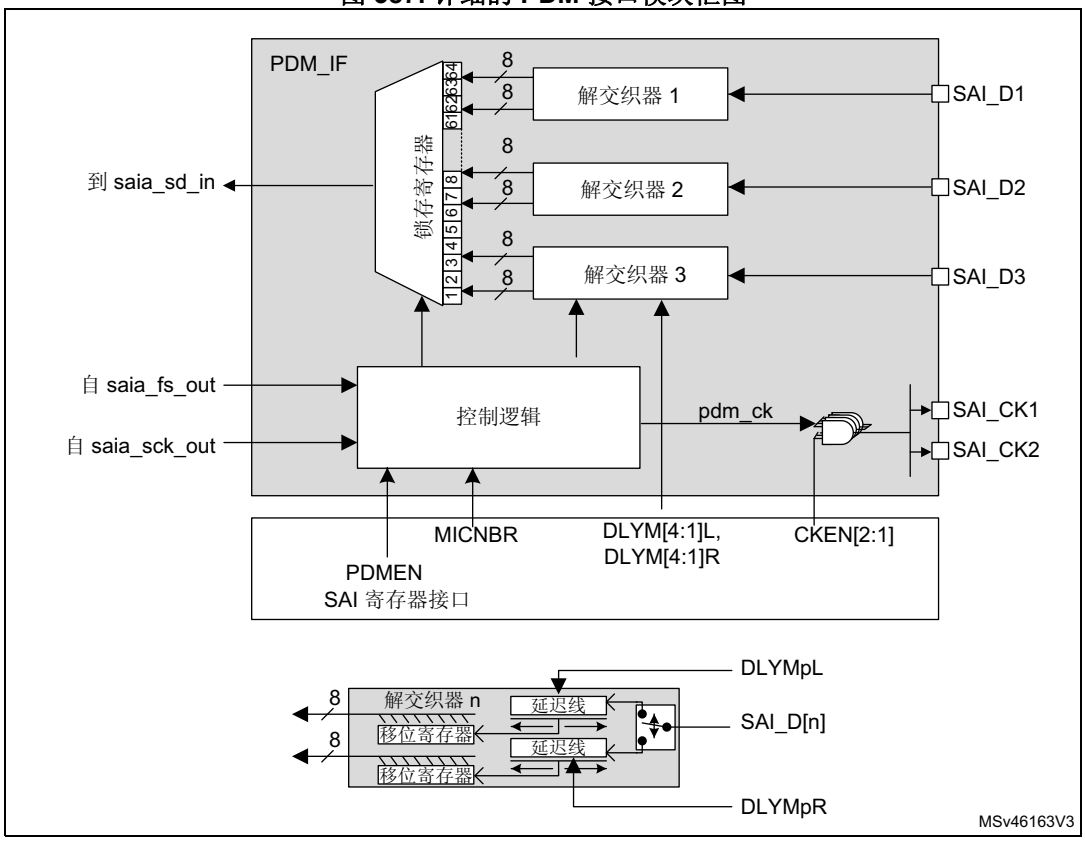
PDM 中的数据处理序列如下：

1. PDM 接口通过从 SAI_A 的 TDM 接口接收的位时钟构建比特流时钟。
2. 接收自麦克风的比特流数据 (SAI_D[n]) 进行解交织 (de-interleaved)，然后经过 7 位延迟线，根据比特流时钟的精度对每个麦克风的延迟进行微调。
3. 移位寄存器将每个串行比特流转换为字节。
4. 最后通过 TDM 接口的串行数据线将得到的字节移出到 SAI_A。

下面的*图 387* 给出了 PDM 接口框图，其中包括解交织 (de-interleaved) 的详细视图。

注： *PDM 接口未嵌入抽样过滤器，因此无法根据比特流构建 PCM 音频采样，需要借助应用软件来执行这一操作。*

图 387. 详细的 PDM 接口模块框图



1. **n** 表示数据线的数量，**p** 表示麦克风对的数量。

可通过 SAI_PDMCR 寄存器中的 PDMEN 位使能 PDM 接口。不过，必须在使能 SAI_A 模块之前使能 PDM 接口。

为减少存储器占用量，用户可根据应用需求选择麦克风的数量。这可以通过 MICNBR[1:0] 位来实现。用户可以选择 2、4、6 或 8 个麦克风。例如，如果应用需要使用 3 个麦克风，则用户必须选择 4 个麦克风。

使能 PDM 接口

要使能 PDM 接口，需遵循以下序列：

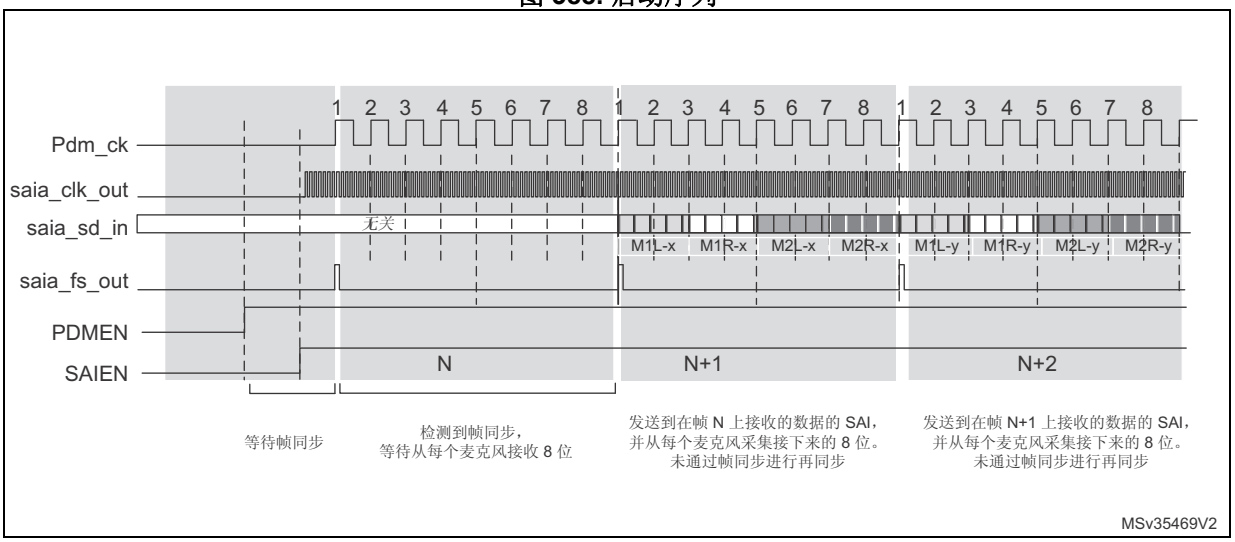
1. 将 SAI_A 配置为 TDM 主模式（请参见*表 212*）。
2. 按以下步骤配置 PDM 接口：
   a) 通过 MICNBR 定义数字麦克风的数量。
   b) 通过将 CKEN 的相应位置 1 来使能应用所需的比特流时钟。
3. 通过 PDMEN 位使能 PDM 接口。
4. 使能 SAI_A。

注： *使能 PDM 接口和 SAI_A 后，在 SAI_ADR 上接收到的前 2 个 TDMA 帧将是无效帧，应将其丢弃。*

启动序列

图 388 显示了启动序列：PDM 接口使能后，会等待帧同步事件，然后再开始采集麦克风采样。经过 8 个 SAI_CK 时钟周期之后，来自每个麦克风的数据字节将变为可用，并会通过 TDM 接口传输到 SAI。

图 388. 启动序列



SAI_ADR 数据格式

在 SAI_ADR 寄存器中，将根据以下参数对来自麦克风的数据进行排列：

- 麦克风的数量
- 所选 Slot 宽度
- 左对齐第一位

Slot 宽度定义了 SAI_ADR 寄存器中每个字的有效位数。

如果选择 32 位的 Slot 宽度，则 SAI_ADR 中的每个数据将包含 32 个有用位。这样可减少存储到存储器中的字数。不过，软件必须相应地执行某些操作来对每个麦克风的数据进行解交织 (de-interleave)。

另一方面，如果将 Slot 宽度设为 8 位，则 SAI_ADR 中的每个数据将包含 8 个有用位。这样会增加存储到存储器中的字数。不过，这样有助于避免额外的处理操作，因为每个字包含一个麦克风的信息。

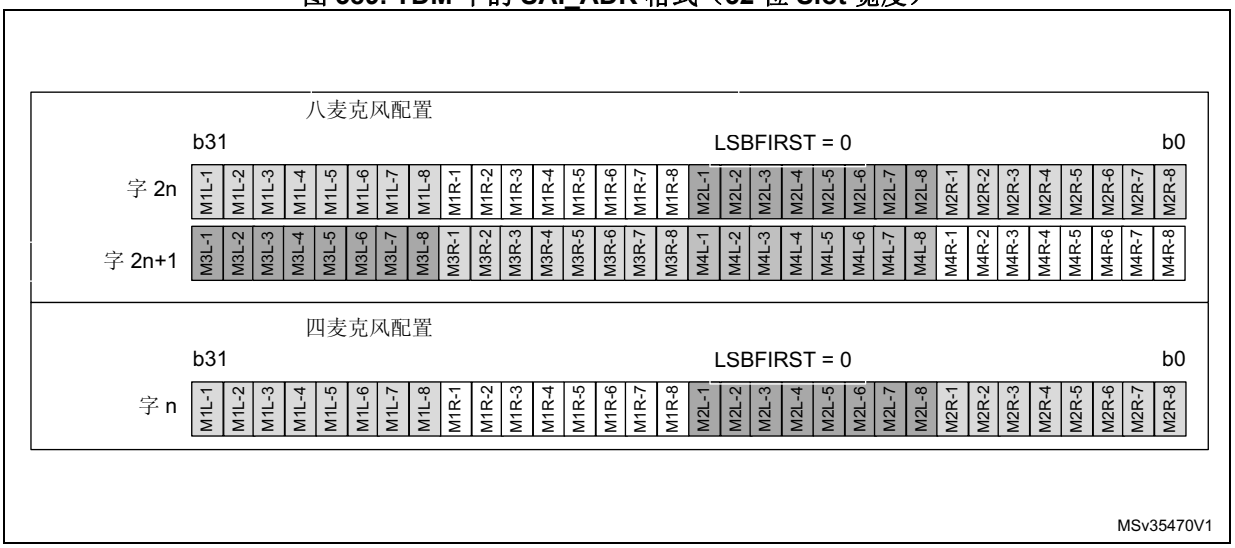
SAI_ADR 数据格式示例

- **32 位 Slot 宽度**（DS = 0b111 且 SLOTSZ = 0）。请参见*图 389*。

  对于 8 麦克风配置，连续对 SAI_ADR 寄存器进行两次字读操作可获取所有麦克风的数据字节。

  对于 4 麦克风配置，每次对 SAI_ADR 寄存器进行字读操作均可获取所有麦克风的数据字节。

图 389. TDM 中的 SAI_ADR 格式（32 位 Slot 宽度）

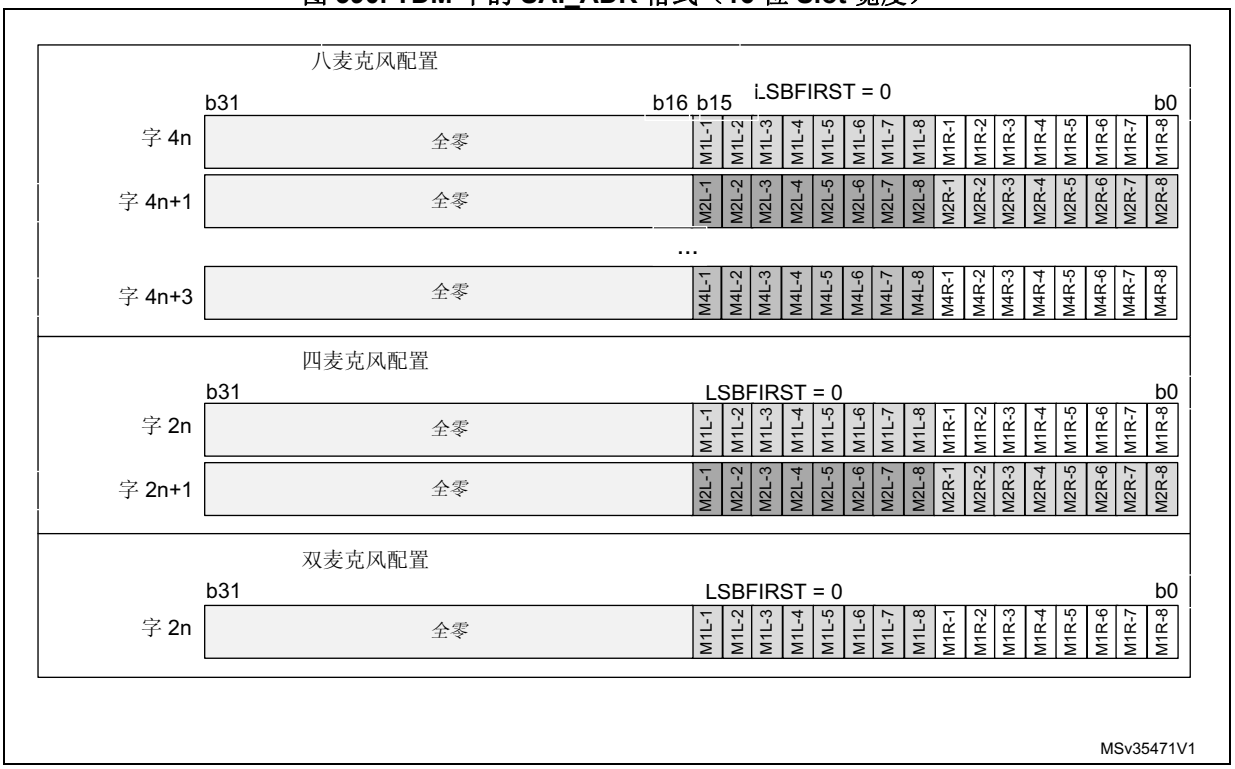


- **16 位 Slot 宽度**（DS = 0b100 且 SLOTSZ = 0）。请参见*图 390*。

  对于 8 麦克风配置，连续对 SAI_ADR 寄存器进行四次字读操作可获取所有麦克风的数据字节。请注意，SAI_ADR 的 16 位数据为右对齐。

  对于 4 麦克风配置或 2 麦克风配置，SAI 行为类似于 8 麦克风配置。获取 4 个麦克风的数据字节需要多达 2 个 16 位字；而获取 2 个麦克风的数据字节需要 1 个 16 位字。

图 390. TDM 中的 SAI_ADR 格式（16 位 Slot 宽度）

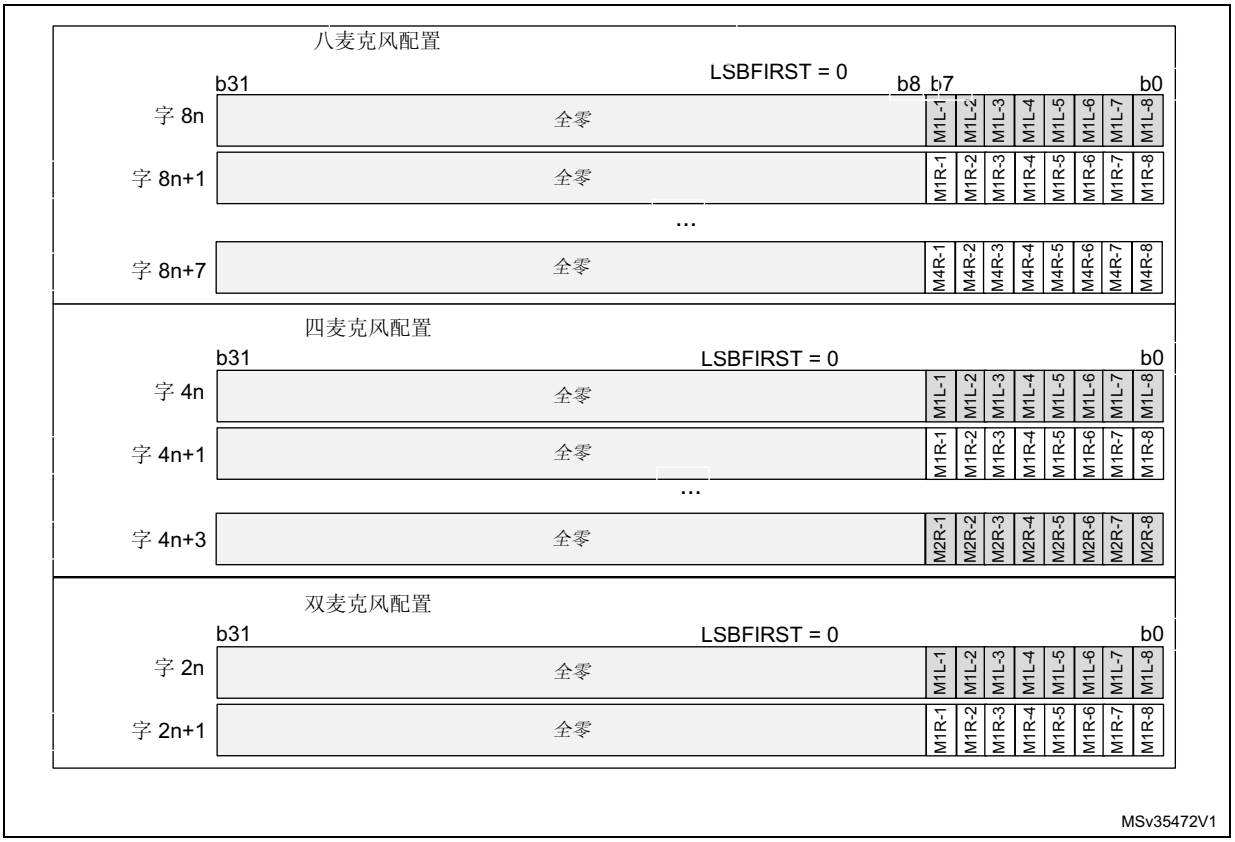


- **使用 8 位 Slot 宽度**（DS = 0b010 且 SLOTSZ = 0）。请参见*图 391*。

  对于 8 麦克风配置，连续对 SAI_ADR 寄存器进行 8 次字读操作可获取所有麦克风的数据字节。请注意，SAI_ADR 的 8 位数据为右对齐。

  对于 4 麦克风配置或 2 麦克风配置，SAI 行为类似于 8 麦克风配置。获取 4 个麦克风的数据字节需要多达 4 个 8 位字；而获取 2 个麦克风的数据字节需要 2 个 8 位字。

图 391. TDM 中的 SAI_ADR 格式（8 位 Slot 宽度）



PDM 接口的 TDM 配置

SAI_A TDM 接口从内部连接到 PDM 接口，以获取麦克风采样。用户应用程序必须按照*表 212* 所示配置 PDM 接口，以确保与 PDM 接口的连接良好。

表 212. TDM 设置

位域	值	注释
MODE	0b01	必须为主接收模式
PRTCFG	0b00	TDM 的自由协议
DS	X	根据所需数据格式，通过帧长度和 Slot 数（FRL 和 NBSLOT）进行调整。请参见*表 213*。
LSBFIRST	X	可根据所需数据格式使用该参数
CKSTR	0	信号在 SCK_A 位时钟的上升沿发生跳变，在位时钟的下降沿达到稳定状态。
MONO	0	立体声模式
FRL	X	根据麦克风数量 (MICNBR) 进行调整。请参见*表 213*。
FSALL	0	脉宽为一个位时钟周期
FSDEF	0	FS 信号为帧起始信号

表 212. TDM 设置（续）

位域	值	注释
FSPOL	1	FS 为高电平有效
FSOFF	0	在 Slot 0 的第一位上使能 FS
FBOFF	0	Slot 上无偏移
SLOTSZ	0	Slot 大小 = 数据大小
NBSLOT	X	根据所需数据格式，通过 Slot 大小和帧长度（FRL 和 DS）进行调整。请参见*表 213*。
SLOTEN	X	根据 NBSLOT 进行调整
NODIV	1	无需生成主时钟 MCLK
MCKDIV	X	取决于提供给 sai_a_ker_ck 输入的频率。 应对该参数进行调整，以生成合适的比特流时钟频率。请参见*表 213*。

调整比特流时钟频率

为正确编程 SAI TDM 接口，用户应用程序必须考虑*表 212* 中给出的设置，并遵循以下规则：

1. 使用以下公式，根据 PDM 比特流时钟所需的频率调整位时钟频率 (F_{SCK_A})：

$$F_{SCK\_A} = F_{PDM\_CK} \times (MICNBR + 1) \times 2$$

MICNBR 可以是 0、1、2 或 3（0 = 2 个麦克风，请参见*第 36.5.17 节*）

2. 使用以下公式设置帧长度 (FRL)

$$FRL = (16 \times (MICNBR + 1)) - 1$$

3. 将 Slot 大小 (DS) 配置为 (FRL+1) 的倍数。

表 213. 允许的 TDM 帧配置(1)

麦克风采样率	麦克风数量	所需 SAI_CK[m] 频率 (2)	位时钟 (SCK_A) 频率	帧同步 (FS_A) 频率	FRL	DS	NBSLOT	注释
48 kHz	最多 8 个	3.072 MHz	24.576 MHz	384 kHz	63	0b100	3	每个帧 4 个 16 位 Slot
		3.072 MHz	24.576 MHz	384 kHz	63	0b010	7	每个帧 8 个 8 位 Slot
	最多 6 个	3.072 MHz	18.432 MHz	384 kHz	47	0b110	1	每个帧 2 个 24 位 Slot
		3.072 MHz	18.432 MHz	384 kHz	47	0b100	2	每个帧 3 个 16 位 Slot
		3.072 MHz	18.432 MHz	384 kHz	47	0b010	5	每个帧 6 个 8 位 Slot
	最多 4 个	3.072 MHz	12.288 MHz	384 kHz	31	0b111	0	每个帧 1 个 32 位 Slot
		3.072 MHz	12.288 MHz	384 kHz	31	0b100	1	每个帧 2 个 16 位 Slot
		3.072 MHz	12.288 MHz	384 kHz	31	0b010	3	每个帧 4 个 8 位 Slot
	最多 2 个	3.072 MHz	6.144 MHz	384 kHz	15	0b100	0	每个帧 1 个 16 位 Slot
		3.072 MHz	6.144 MHz	384 kHz	15	0b010	1	每个帧 2 个 8 位 Slot
16 kHz	最多 8 个	1.024 MHz	8.192 MHz	128 kHz	63	0b100	3	每个帧 4 个 16 位 Slot
		1.024 MHz	8.192 MHz	128 kHz	63	0b010	7	每个帧 8 个 8 位 Slot
	最多 6 个	1.024 MHz	6.144 MHz	128 kHz	47	0b110	1	每个帧 2 个 24 位 Slot
		1.024 MHz	6.144 MHz	128 kHz	47	0b010	5	每个帧 6 个 8 位 Slot
	最多 4 个	1.024 MHz	4.096 MHz	128 kHz	31	0b111	0	每个帧 1 个 32 位 Slot
		1.024 MHz	4.096 MHz	128 kHz	31	0b100	1	每个帧 2 个 16 位 Slot
		1.024 MHz	4.096 MHz	128 kHz	31	0b010	3	每个帧 4 个 8 位 Slot
	最多 2 个	1.024 MHz	2.048 MHz	128 kHz	15	0b100	0	每个帧 1 个 16 位 Slot
		1.024 MHz	2.048 MHz	128 kHz	15	0b010	1	每个帧 2 个 8 位 Slot

1. 有关 TDM 配置的更多信息，请参见*表 212：TDM 设置*。提供给 SAI 的 sai_a_ker_ck 时钟频率应是 SCK_A 频率的倍数，并且应相应地编程 MCKDIV。
2. 上表给出了抽取率为 64 时所允许的设置。

调整延迟线

使能 PDM 接口时，应用程序可通过 SAI_PDMDLY 寄存器实时调整每个麦克风输入的延迟单元数。

新的延迟值将在两个 TDM 帧后生效。

36.3.11 AC’97 链路控制器

SAI 可用作 AC’97 链路控制器。在此协议中：

- Slot 数和 Slot 大小固定。
- 帧同步信号定义完善并且波形固定。

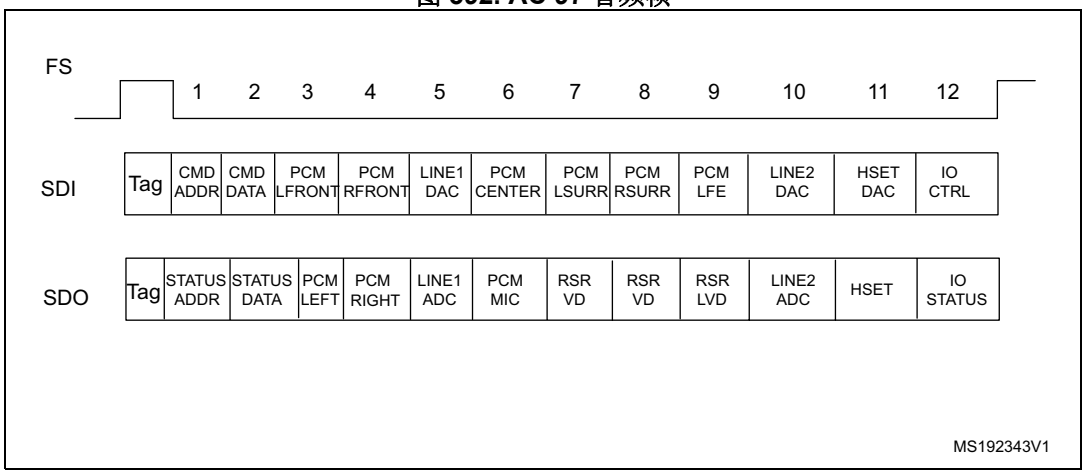
要选择此协议，可将 SAI_xCR1 寄存器中的 PRTCFG[1:0] 位置为 10。选择 AC’97 模式时，只能使用 16 位或 20 位的数据大小，否则将无法保证 SAI 运行正常。

- 因此，NBSLOT[3:0] 和 SLOTSZ[1:0] 位将被忽略。
- Slot 数固定为 13。第一个 Slot 为 16 位宽，所有其他 Slot 均为 20 位宽（数据 Slot）。
- SAI_xSLOTR 寄存器中的 FBOFF[4:0] 位被忽略。
- SAI_xFRCR 寄存器被忽略。
- 未使用 MCLK。

无论采用主配置还是从配置，由于 AC’97 链路控制器驱动 FS 信号，FS 信号自动被配置成异步输出。

图 392 给出了 AC’97 音频帧的结构。

图 392. AC’97 音频帧



注： *在 AC’97 协议中，TAG 的位 2 将保留（始终为 0），因此无论 SAI FIFO 中写入何值，TAG 的位 2 均强制为 0。*

有关 TAG 表示的详细信息，请参见 AC’97 协议标准。

可将一个 SAI 用于 AC’97 点对点通信。

在接收器模式下，用作 AC’97 链路控制器的 SAI 无需 FIFO 请求，因此当 Slot 0 中的编码解码器就绪位解码为低电平时，FIFO 中不存储任何数据。如果 SAI_xIM 寄存器中的 CNRDYIE 位使能，则 SAI_xSR 寄存器中的标志 CNRDY 将置 1 并会产生一个中断。此标志专用于 AC’97 协议。

AC’97 模式下的时钟发生器编程

在 AC’97 模式下，帧长度固定为 256 位，其频率应设置为 48 kHz。*第 36.3.8 节：SAI 时钟发生器*中给出的规则应在 FRL = 255 时使用，以生成适当的帧速率 (F_{FS_x})。

36.3.12 SPDIF 输出

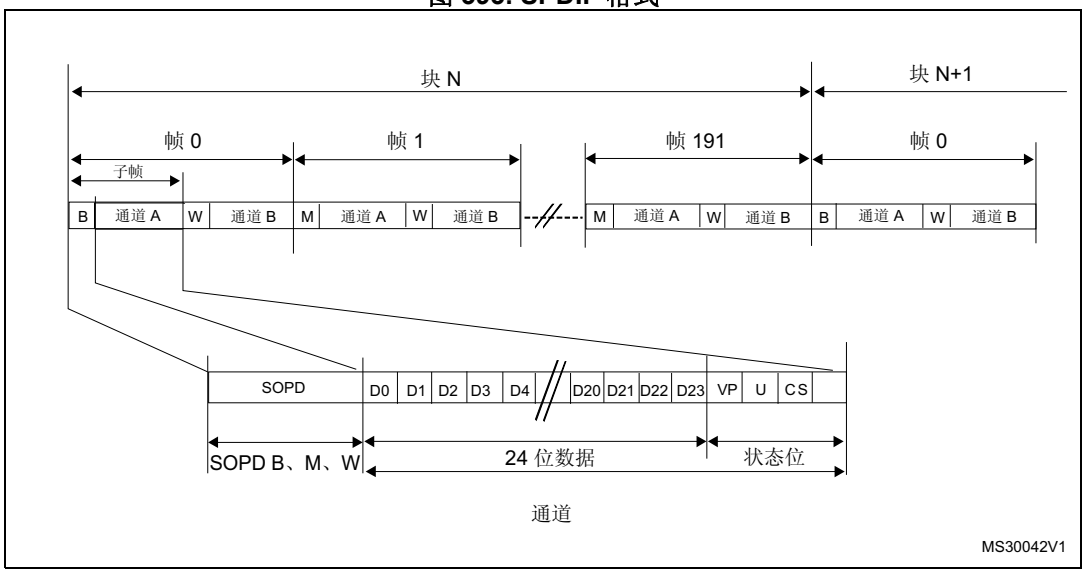
SPDIF 接口仅在发送模式下可用。它支持 IEC60958 音频标准。

要选择 SPDIF 模式，需将 SAI_xCR1 寄存器中的 PRTCFG[1:0] 位置为 01。

对于 SPDIF 协议：

- 仅使能 SD 数据线。
- 释放 FS、SCK、MCLK I/O 引脚。
- 为使能 SAI 的时钟发生器并管理 SD 线上的数据速率，将 MODE[1] 位强制设置为 0 以选择主模式。
- 数据大小强制设置为 24 位。忽略在 SAI_xCR1 寄存器的 DS[2:0] 位中设置的值。
- 由于位时钟是符号率的两倍，由此必须配置时钟发生器来定义符号率。数据采用曼彻斯特协议进行编码。
- 忽略 SAI_xFRCR 和 SAI_xSLOTR 寄存器。在内部配置 SAI 使其符合 SPDIF 协议要求，如*图 393* 所示。

图 393. SPDIF 格式



SPDIF 块包含 192 个帧。每个帧由两个 32 位的子帧构成，通常一个子帧用于左通道，一个用于右通道。每个子帧由一个 SOPD 式样（4 位）构成，用于指定该子帧是否为块的开始（并用于识别通道 A），或者用于识别块中某处的通道 A，或者通道 B（请参见*表 214*）。接下来的 28 位是通道信息，由 24 个数据位和 4 个状态位组成。

表 214. SOPD 式样

SOPD	前导码		说明
	最后一位是 0	最后一位是 1	
B	11101000	00010111	块开始处的通道 A 数据
W	11100100	00011011	块中某处的通道 B 数据
M	11100010	00011101	通道 A 数据

必须按如下方式填充 SAI_xDR 中存储的数据：

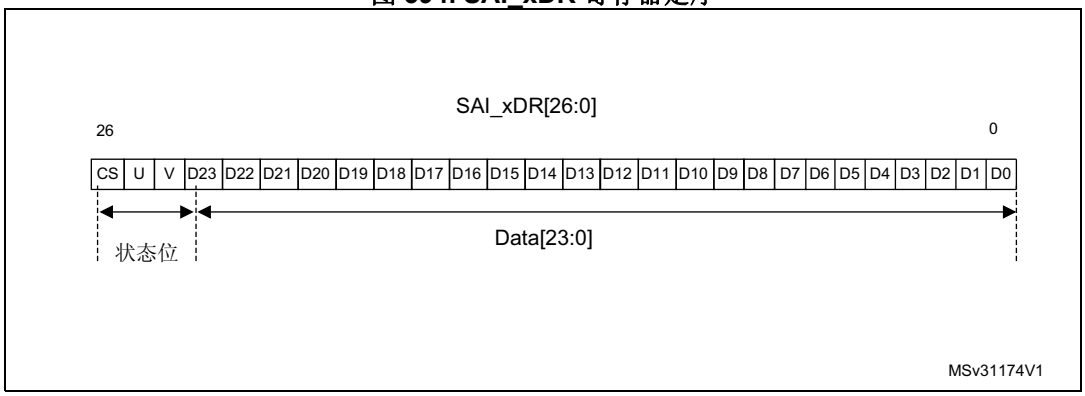
- SAI_xDR[26:24] 包含通道状态位、用户位和有效性位。
- SAI_xDR[23:0] 包含所考虑通道的 24 位数据。

如果数据大小为 20 位，应将数据映射到 SAI_xDR[23:4] 上。

如果数据大小为 16 位，应将数据映射到 SAI_xDR[23:8] 上。

SAI_xDR[23] 始终代表 MSB。

图 394. SAI_xDR 寄存器定序



注： *执行传输时，LSB 始终优先。*

SAI 首先发送一个块子帧的对应前导码，随后在 SD 线上发送 SAI_xDR（以曼彻斯特协议进行编码）。SAI 通过传输按 *表 215* 所述计算的奇偶校验位来结束子帧。

表 215. 奇偶校验位计算

SAI_xDR[26:0]	传输奇偶校验位 P 值
奇数个 0	0
奇数个 1	1

在 SPDIF 模式下，下溢是在 SAI_xSR 寄存器唯一个用到的错误标志。因为 SAI 只能在发送模式下工作。因此，要从通过下溢中断或下溢状态位检测到的下溢错误中恢复，应按顺序执行以下步骤：

1. 如果已使用 DMA，则禁止 DMA 流（通过 DMA 外设）。
2. 禁止 SAI 并通过轮询 SAI_xCR1 寄存器中的 SAIEN 位确认已从物理上禁止外设。
3. 将 SAI_xCLRFR 寄存器中的 COVRUNDR 标志清零。
4. 通过将 SAI_xCR2 寄存器中的 FFLUSH 位置 1 来清空 FIFO。
   软件需要指向与新块开始（前导码 B 的数据）相对应的后续数据的地址。如果已使用 DMA，应相应地更新 DMA 源起始地址指针。
5. 如果用于管理数据的 DMA 根据新的源起始地址来传输数据，则再次使能 DMA 流（DMA 外设）。
6. 通过将 SAI_xCR1 寄存器中的 SAIEN 位置 1 再次使能 SAI。

SPDIF 发生器模式下的时钟发生器编程

对于 SPDIF 发生器，SAI 应提供等于符号率两倍的位时钟。下表给出了符号率相对于音频采样率的通常示例。

表 216. 音频采样频率与符号率

音频采样频率 (F_S)	符号率
44.1 kHz	2.8224 MHz
48 kHz	3.072 MHz
96 kHz	6.144 MHz
192 kHz	12.288 MHz

通常，音频采样频率 (F_S) 与位时钟速率 (F_{SCK_X}) 之间的关系由以下公式给出：

$$F_S = \frac{F_{SCK\_x}}{128}$$

按如下公式计算位时钟速率：

$$F_{SCK\_x} = \frac{F_{SAI\_CK\_x}}{MCKDIV}$$

注： *仅当在 SAI_ACR1 寄存器中将 NODIV 置 1 时，上述公式才有效。*

36.3.13 特性

根据所选的音频协议，SAI 接口内嵌了一些特定的实用功能。这些功能可通过 SAI_xCR2 寄存器的特定位来访问。

静音模式

当音频子模块用作发送器或接收器时，可使用静音模式。

发送模式下的音频子模块

在发送模式下，可随时选择静音模式。静音模式对于全部音频帧均有效。在帧传输过程中将 SAI_xCR2 寄存器的 MUTE 置 1，将使能静音模式。

该静音模式位仅在帧结束时选通。如果帧结束时置 1，静音模式将在新的音频帧开始时激活，并持续整个帧长度，直至下次帧结束； 然后将选通此位，以确定下一帧是否仍为静音帧。

如果通过 SAI_xSLOTR 寄存器的 NBSLOT[3:0] 位设置的 Slot 数小于或等于 2，可指定静音模式下发送的值是否为 0 或该值是否为每个 Slot 的最后一个值。通过 SAI_xCR2 寄存器中的 MUTEVAL 位进行选择。

如果在 SAI_xSLOTR 寄存器的 NBSLOT[3:0] 位中设置的 Slot 数大于 2，由于在各 Slot 的每位上都发送值 0，因此 SAI_xCR2 中的 MUTEVAL 位没有意义。

在静音模式下，FIFO 指针仍递增，这意味着将丢弃 FIFO 中请求在静音模式下传输的数据。

接收模式下的音频子模块

在接收模式下，对于给定数量的连续音频帧（SAI_xCR2 寄存器中的 MUTECNT[5:0] 位），如果在音频帧所有声明的有效 Slot 接收到 0，则可检测到从外部发送器发来的静音模式。

检测到相应数量的静音帧时，SAI_xSR 寄存器中的 MUTEDET 标志置 1 并会在 SAI_xCR2 中的 MUTEDETIE 位置 1 情况下产生中断。

在音频子模块禁止时或在有效 Slot 内至少接收到音频帧中的一个数据时，静音帧计数器清零。计数器达到 MUTECNT[5:0] 位中指定的值时，仅产生一次中断。随后中断事件在计数器清零时重新初始化。

注： *静音模式不可用于 SPDIF 音频模块。*

单声道/立体声模式

在发送器模式下，如果 Slot 数等于 2（SAI_xSLOTR 中的 NBSLOT[3:0] = 0001），则可寻址单声道模式，而无需在存储器中进行任何数据预处理。在这种情况下，由于发送时 Slot 0 的数据被复制到数据 Slot 1 中，FIFO 的访问时间将减少一半。

要使能单声道模式，

1. 将 SAI_xCR1 寄存器中的 MONO 位置 1。
2. 将 SAI_xSLOTR 中的 NBSLOT 置 1，并将 SLOTEN 设置为 3。

在接收模式下，MONO 位可置 1 并且仅在 Slot 数等于 2（与发送模式下相同）时才有意义。当此位置 1 时，只有 Slot 0 的数据将存储到 FIFO 中。Slot 1 的数据由于被认为与前一个 Slot 的数据相同而被丢弃。如果接收模式下的数据流量是左声道数据和右声道数据明显不同的真立体声音频流，则 MONO 位没有意义。由软件完成从输出立体声文件到等效单声道文件的转换。

压扩模式

移动通信应用可能需要通过数据压扩算法处理待发送或待接收的数据。

应用软件可根据 SAI_xCR2 寄存器中的 COMP[1:0] 位（仅当选择自由协议模式时使用），选择在 SD 串行输出线（压缩）发送数据前是否处理数据，以及是否在 SD 串行输入线（扩展）接收数据后扩展数据，如*图 395* 所示。所支持的两个压扩模式是 μ-Law 和 A-Law，二者是 CCITT G.711 推荐标准的一部分。

美国和日本采用的压扩标准是 μ-Law，该标准允许 14 位动态范围（SAI_xCR2 寄存器中的 COMP[1:0] = 10）。

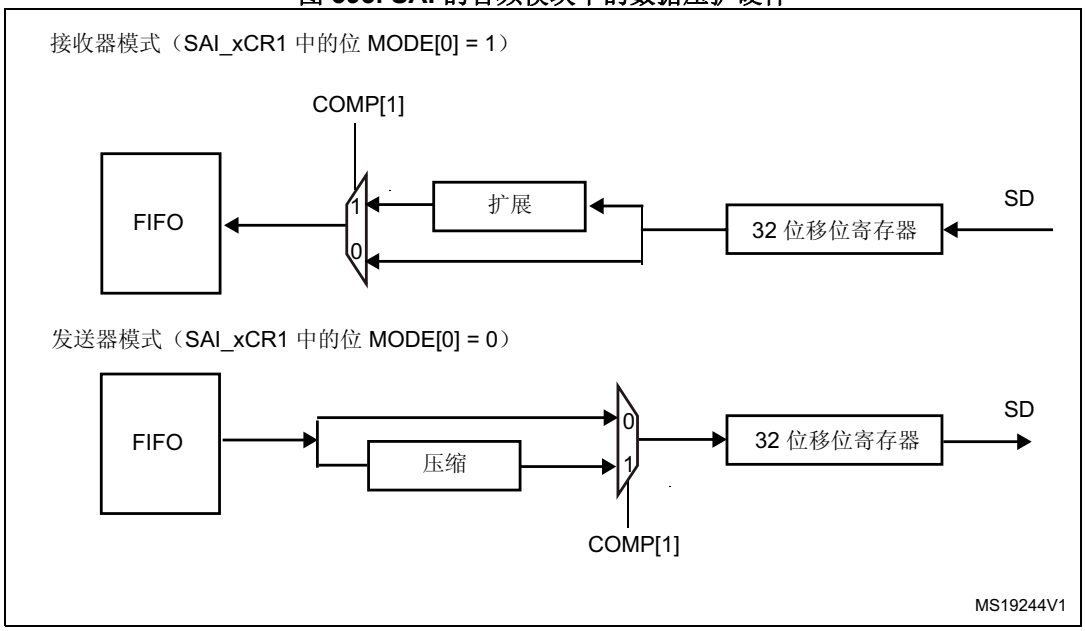
欧洲压扩标准是 A-Law，该标准支持 13 位动态范围（SAI_xCR2 寄存器中的 COMP[1:0] = 11）。

可根据 1 的补码或 2 的补码表示来计算 μ-Law 或 A-Law 压扩标准，具体取决于 SAI_xCR2 寄存器中的 CPL 位设置。

在 μ-Law 和 A-Law 标准中，数据将编码为采用 MSB 对齐的 8 位。压扩数据始终为 8 位宽。因此，当 SAI 音频模块使能（SAI_xCR1 寄存器中的位 SAIEN = 1）并且通过 COMP[1:0] 位选择这两个压扩模式之一时，SAI_xCR1 寄存器中的 DS[2:0] 位将强制为 010。

如果无需压扩处理，则 COMP[1:0] 位应保持清零。

图 395. SAI 的音频模块中的数据压扩硬件



1. 选择 AC'97 或 SPDIF 时不适用。

通过 SAI_xCR2 自动选择扩展模式和压缩模式：

- 如果 SAI 音频模块配置为发送器，且 SAI_xCR2 寄存器中的 COMP[1] 位置 1，将采用压缩模式。
- 如果 SAI 音频模块声明为接收器，将采用扩展算法。

无效 Slot 上的输出数据线管理

在发送模式下，当在数据线上发送无效 Slot 时，可选择 SD 线输出的行为（通过 TRIS 位实现）。

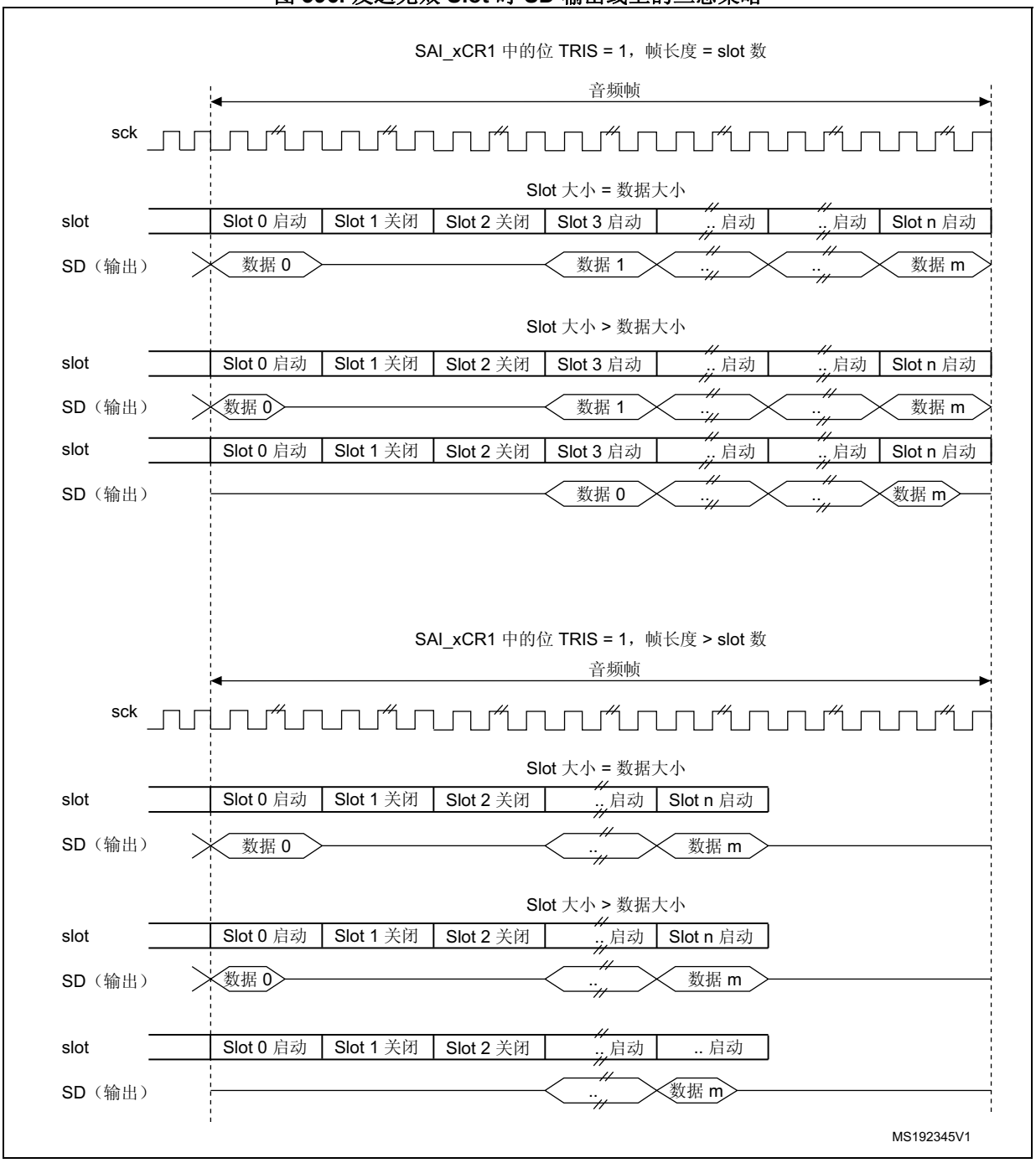
- 发送无效 Slot 时，SAI 将 SD 输出线强制为 0。
- 该输出线在最后一个数据位传输结束时释放为高阻态，为另一个连接此节点的发送器释放该数据线。

切记不要让两个发送器同时驱动同一个 SD 输出引脚，否则会导致短路。为确保存在发送间隙，如果数据低于 32 位，可通过在 SAI_xSLOTR 寄存器中设置 SLOTSZ[1:0] = 10 将数据扩展到 32 位。随后，如果下一 Slot 声明为无效，则 SD 输出引脚将在有效 Slot 的 LSB 结束时（将数据扩展到 32 位的填 0 阶段）置为三态。

此外，如果 Slot 数乘以 Slot 大小所得结果小于帧长度，则在通过填 0 来补充音频帧结束时，SD 输出线置为三态。

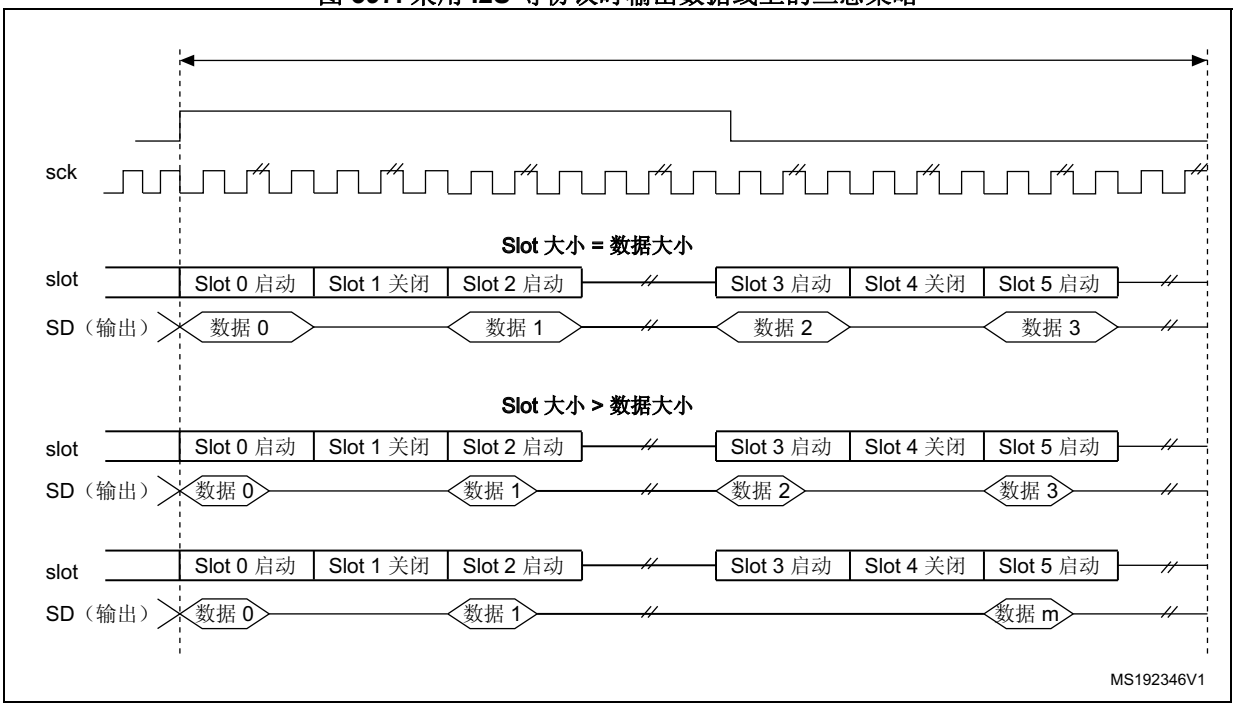
图 396 说明了这些行为。

图 396. 发送无效 Slot 时 SD 输出线上的三态策略



当所选音频协议使用 FS 信号作为 SOF 信号或通道识别信号（SAI_xFRCR 寄存器中的位 FSDEF = 1）时，将按照*图 397* 管理三态模式（其中，SAI_xCR1 寄存器中的位 TRIS = 1，FSDEF = 1，半帧长大于 Slot 数/2 且 NBSLOT = 6）。

图 397. 采用 I2S 等协议时输出数据线上的三态策略



如果 SAI_xCR2 寄存器中的 TRIS 位清零，*图 396* 和*图 397* 上的 SD 输出线上的所有高阻态将替换为使用值 0 驱动。

36.3.14 错误标志

SAI 使用以下错误标志：

- FIFO 上溢/下溢
- 帧同步提前检测
- 帧同步滞后检测
- 编解码器未就绪（仅限 AC'97）
- 主模式时钟配置错误

FIFO 上溢/下溢 (OVRUDR)

FIFO 上溢/下溢位是 SAI_xSR 寄存器中的 OVRUDR 位。

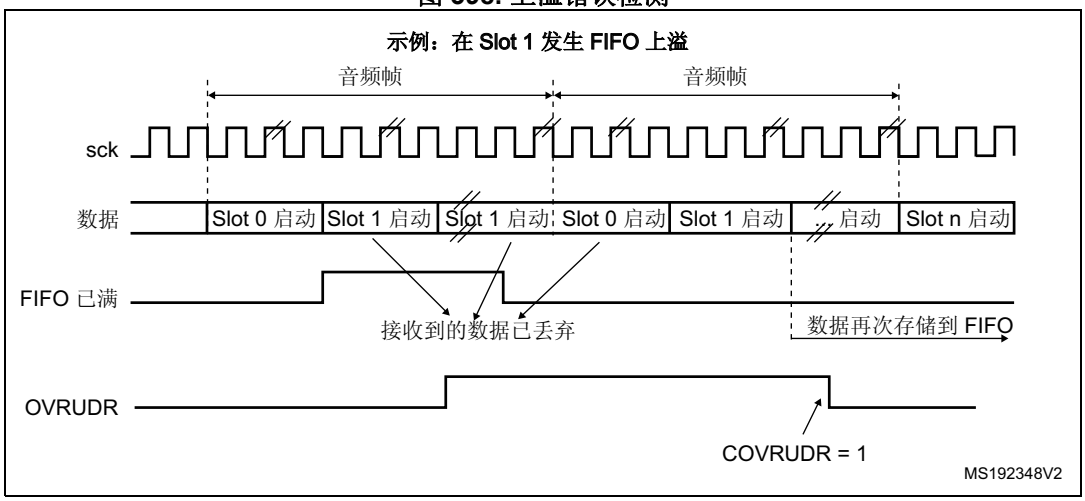
由于音频模块既可作为接收器，又可作为发送器，并且指定 SAI 中的每个音频模块都具有自己的 SAI_xSR 寄存器，因此上溢或下溢错误共用同一位。

上溢

若音频模块配置为接收器，则在 FIFO 已满且无法再存储接收数据的情况下又收到音频帧数据时，将出现上溢情况。这种情况下，接收数据将丢失，SAI_xSR 寄存器中的 OVRUDR 标志置 1；如果 SAI_xIM 寄存器中的 OVRUDRIE 位置 1，还将生成中断。内部将存储发生上溢时的 Slot 编号。FIFO 无法再存储更多数据，直至释放出空间存储新数据为止。在 FIFO 释放了至少一个数据的空间时，SAI 音频模块接收器将从检测到上溢后内部记录的 Slot 编号开始接收来自新音频帧的新数据，这样可避免目标存储器中出现数据 Slot 不对齐的情况（请参见*图 398*）。

SAI_xCLRFR 寄存器中的 COVRUDR 位置 1 时将清除 OVRUDR 标志。

图 398. 上溢错误检测



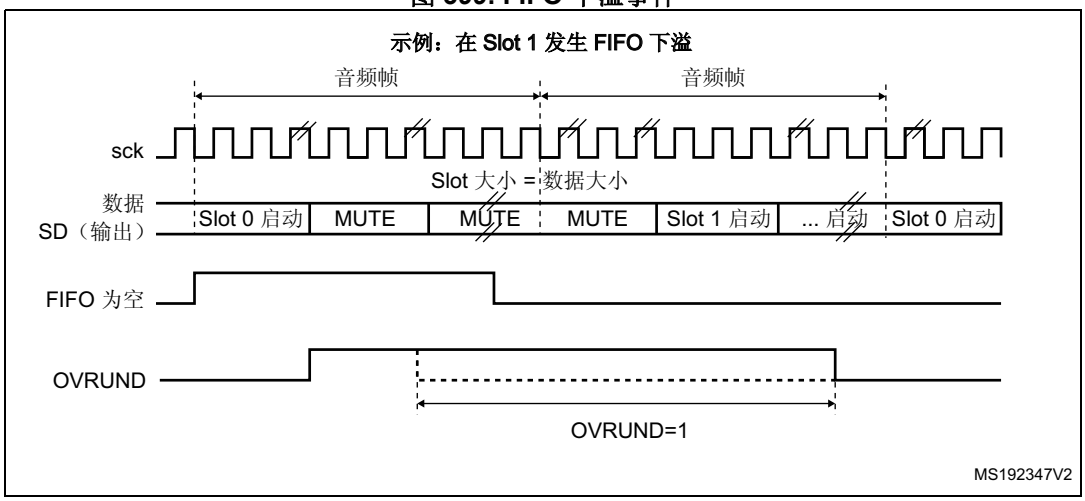
下溢

当 SAI 中的音频模块用作发送器时，如果需要发送数据时 FIFO 为空，则可能出现下溢。如果检测到下溢，则将存储发生事件的 Slot 编号并发送 MUTE 值 (00)，直到 FIFO 准备好发送与检测到下溢的 Slot 对应的数据（请参见*图 399*）。这样可避免存储器指针与音频帧中的 Slot 之间发生同步失效。

下溢事件会使 SAI_xSR 寄存器中的 OVRUDR 标志置 1，如果 SAI_xIM 寄存器中的 OVRUDRIE 位置 1，还将生成中断。要清除该标志，可将 SAI_xCLRFR 寄存器中的 COVRUDR 位置 1。

将音频子模块配置为主模式或从模式时，会发生下溢事件。

图 399. FIFO 下溢事件



帧同步提前检测 (AFSDET)

AFSDET 标志仅在从模式下使用。主模式下不会使能该标志。由于帧长度、帧极性和帧偏移已定义且已知，该标志用于指示是否比预期更早检测到帧同步 (FS) 信号。

出现帧检测提前时，SAI_xSR 寄存器中的 AFSDET 标志将置 1。

对 FS 提前不敏感的当前音频帧不受该检测影响。也就是说 FS 信号的“寄生”事件将被标记但不干扰当前音频帧。

如果 SAI_xIM 寄存器中的 AFSDETIE 位置 1，将生成中断。要清除 AFSDET 标志，必须将 SAI_xCLRFR 寄存器中的 CAFSDET 位置 1。

为了在出现帧检测提前错误后重新与主模块同步，需要执行以下四个步骤：

1. 通过复位 SAI_xCR1 寄存器中的 SAIEN 位禁止 SAI 模块。为确保禁止 SAI，读回 SAIEN 位并检查其是否已设置为 0。
2. 通过 SAI_xCR2 寄存器的 FFLUS 位清空 FIFO。
3. 重新使能 SAI 外设（SAIEN 位置 1）。
4. SAI 模块将等待 FS 使能以重新开始与主模块同步。

注： *AC’97 模式下不使能 AFSDET 标志，原因是 SAI 音频模块作为链路控制器，即使在声明为从模块时也会生成 FS 信号。由于未使用 FS 信号，因此它在 SPDIF 模式下无意义。*

帧同步滞后检测

只有当 SAI 音频模块用作从模块时，SAI_xSR 寄存器中的 LFSDET 标志才可置 1。SAI_xFRCR 寄存器中，帧长度、帧极性和帧偏移配置均已知。

如果外部主模块未在预定时间发送 FS 信号生成过晚，LFSDET 标志将置 1，如果 SAI_xIM 寄存器中的 LFSDETIE 位置 1，还将生成中断。

SAI_xCLRFR 寄存器中的 CLFSDET 位置 1 时将清除 LFSDET 标志。

在检测到相应错误时帧同步滞后检测标志置 1，SAI 需要重新与主模块同步（请参见 *帧同步提前检测 (AFSDET)* 中所述的顺序）。

在噪声环境中，音频模块的状态机可能会错误检测到对 SCK 时钟的干扰，并将 SAI 数据移位到错误的帧位置。SAI 会检测到此事件，并将其报告为帧同步滞后检测错误。

如果外部主模块不是在连续模式下管理音频数据帧发送，则不会对帧造成破坏，大多数应用都不会出现这种情况。这种情况下 LFSDET 标志将置 1。

注： *AC’97 模式下不使能 LFSDET 标志，原因是 SAI 音频模块作为链路控制器，即使在声明为从模块时也会生成 FS 信号。由于协议未使用 FS 信号，因此此位在 SPDIF 模式下无意义。*

编解码器未就绪 (CNRDY AC’97)

仅当 SAI 音频模块配置为在 AC’97 模式下工作时（SAI_xCR1 寄存器中的 PRTCFG[1:0] = 10），SAI_xSR 寄存器中的 CNRDY 标志才有意义。如果 SAI_xIM 寄存器中的 CNRDYIE 位置 1，则在 CNRDY 标志置 1 时将生成中断。

在接收 AC’97 音频帧的 TAG 0 (slot0) 期间，当编解码器未准备好进行通信时，将使能 CNRDY。这种情况下，在 TAG 0 指示编解码器就绪之前，数据都不会自动存储到 FIFO，原因是编解码器未就绪。编解码器就绪后将捕获 SAI_xSLOTR 寄存器中定义的所有有效 Slot 。

要清除 CNRDY 标志，必须将 SAI_xCLRFR 寄存器中的 CCNRDY 位置 1。

主模式时钟配置错误 (NODIV = 0)

当音频模块在主模式下工作 (MODE[1] = 0) 且 NODIV 位等于 0 时，如果满足以下条件，则只要使能 SAI，WCKCFG 标志便会置 1：

- (FRL+1) 不是 2 的几次幂，并且
- (FRL+1) 不在 8 和 256 之间

MODE 位、NODIV 位和 SAIEN 位属于 SAI_xCR1 寄存器，FRL 位属于 SAI_xFRCR 寄存器。

如果 WCKCFGIE 位置 1，则当 SAI_xSR 寄存器中的 WCKCFG 标志置 1 时将生成中断。要清除该标志，可将 SAI_xCLRFR 寄存器中的 CWCKCFG 位置 1。

当 WCKCFG 位置 1 时，音频模块会自动禁止，此时 SAIEN 位也将由硬件自动清零。

36.3.15 禁止 SAI

可随时通过清零 SAI_xCR1 寄存器中的 SAIEN 位禁止 SAI 音频模块。所有已开始的帧将在 SAI 停止工作前自动完成。SAIEN 位将保持高电平，直到当前音频帧传输结束时 SAI 完全关闭。

如果 SAI 中有与另一个音频模块同步工作的音频模块，则必须先禁止以主模式工作的音频模块。

36.3.16 SAI DMA 接口

为减轻 CPU 负担和优化总线带宽，每个 SAI 音频模块都具有独立的 DMA 接口以便对 SAI_xDR 寄存器进行读/写操作（访问内部 FIFO）。每个音频子模块都有一个支持基本 DMA 请求/应答协议的 DMA 通道。

要为 DMA 传输配置音频子模块，可将 SAI_xCR1 寄存器中的 DMAEN 位置 1。DMA 请求直接由 FIFO 控制器管理，具体取决于 FIFO 阈值（更多详细信息，请参见*第 36.3.9 节：内部 FIFO*）。DMA 传输方向与 SAI 音频子模块配置相关：

- 如果音频模块用作发送器，则音频模块的 FIFO 控制器将输出 DMA 请求以向 FIFO 加载 SAI_xDR 寄存器中写入的数据。
- 如果音频模块用作接收器，则 DMA 请求与来自 SAI_xDR 寄存器的读取操作相关。

按照下面的顺序将 SAI 接口配置为 DMA 模式：

1. 配置 SAI 和 FIFO 阈值以指定何时启动 DMA 请求。
2. 配置 SAI DMA 通道。
3. 使能 DMA。
4. 使能 SAI 接口。

注： *配置 SAI 模块前，必须禁止 SAI DMA 通道。*

36.4 SAI 中断

SAI 支持 7 个中断源，如*表 217* 所示。

表 217. SAI 中断源

中断源	中断组	音频模块模式	中断使能	中断清零
FREQ	FREQ	主或从 接收器或发送器	SAI_xIM 寄存器中的 FREQIE	取决于： – FIFO 阈值设置（SAI_xCR2 中的 FLVL 位） – 通信方向（发送器或接收器） 更多详细信息，请参见*第 36.3.9 节：内部 FIFO*
OVRUDR	ERROR	主或从 接收器或发送器	SAI_xIM 寄存器中的 OVRUDRIE	SAI_xCLRFR 寄存器中的 COVRUDR = 1
AFSDET	ERROR	从 （不适用于 AC’97 模式和 SPDIF 模式）	SAI_xIM 寄存器中的 AFSDETIE	SAI_xCLRFR 寄存器中的 CAFSDET = 1
LFSDET	ERROR	从 （不适用于 AC’97 模式和 SPDIF 模式）	SAI_xIM 寄存器中的 LFSDETIE	SAI_xCLRFR 寄存器中的 CLFSDET = 1
CNRDY	ERROR	从 （仅限 AC’97 模式）	SAI_xIM 寄存器中的 CNRDYIE	SAI_xCLRFR 寄存器中的 CCNRDY = 1
MUTEDET	MUTE	主或从 仅限接收模式	SAI_xIM 寄存器中的 MUTEDETIE	SAI_xCLRFR 寄存器中的 CMUTEDET = 1
WCKCFG	ERROR	主模式且 SAI_xCR1 寄存器中的 NODIV = 0	SAI_xIM 寄存器中的 WCKCFGIE	SAI_xCLRFR 寄存器中的 CWCKCFG = 1

按照以下顺序使能中断：

1. 禁止 SAI 中断。
2. 配置 SAI。
3. 配置 SAI 中断源。
4. 使能 SAI。

36.5 SAI 寄存器

36.5.1 配置寄存器 1 (SAI_ACR1)

Configuration register 1

偏移地址：0x004

复位值：0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	MCKEN	OSR	MCKDIV[5:0]						NODIV	Res.	DMAEN	SAIEN
				rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	OUTDRIV	MONO	SYNCEN[1:0]		CKSTR	LSBFIRST	DS[2:0]			Res.	PRTCFG[1:0]		MODE[1:0]	
		rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27 **MCKEN**：主时钟生成使能 (Master clock generation enable)
0：不生成主时钟
1：独立于 SAIEN 位生成主时钟

位 26 **OSR**：主时钟的过采样率 (Oversampling ratio for master clock)
只有在 NODIV 位置 0 时，此位才有意义。
0：主时钟频率 = F_{FS} x 256
1：主时钟频率 = F_{FS} x 512

位 25:20 **MCKDIV[5:0]**：主时钟分频器 (Master clock divider)。
这些位将由软件置 1 和清零。
000000：内核时钟输入 (sai_x_ker_ck) 1 分频。
否则，主时钟频率将根据*第 36.3.8 节：SAI 时钟发生器*中的公式计算。
音频模块为从模块时，这些位没有意义。
必须在音频模块禁止的情况下配置这些位。

位 19 **NODIV**：无分频器 (No divider)
此位由软件置 1 和清零。
0：主时钟发生器和帧同步之间的比值固定为 256 或 512
1：主时钟发生器和帧同步之间的比值取决于 FRL[7:0]

位 18 保留，必须保持复位值。

位 17 **DMAEN**：DMA 使能 (DMA enable)
此位由软件置 1 和清零。
0：禁止 DMA
1：使能 DMA

注： *在接收模式下，必须在 DMAEN 位置 1 前配置 MODE[1:0] 位，以避免 DMA 请求，原因是复位后音频模块将默认以发送模式工作。*

位 16 **SAIEN**：音频模块使能 (Audio block enable)

此位由软件置 1。

要关闭音频模块，必须由应用软件将此位编程为 0 并轮询此位，直到此位读回 0，这表示该模块已完全禁止。将此位置 1 前，先检查此位是否已置 0，否则不会执行此使能命令。

此位可用于控制 SAI 音频模块的状态。如果在音频帧传输期间禁止此位，则正在进行的传输将继续，直到传输结束时 SAI 音频模块才完全禁止。

0：禁止 SAI 音频模块

1：使能 SAI 音频模块

注： 当 SAI 模块（A 或 B）配置为主模式时，SAI 模块输入中必须有时钟，然后才能将 SAIEN 位置 1。

位 15:14 保留，必须保持复位值。

位 13 **OUTDRIV**：输出驱动 (Output drive)

此位由软件置 1 和清零。

0：当 SAIEN 置 1 时驱动音频模块输出。

1：在此位置 1 后立即驱动音频模块输出。

注： 此位必须在音频模块配置后的使能前置 1。

位 12 **MONO**：单声道模式 (Mono mode)

此位由软件置 1 和清零。仅当 Slot 数为 2 时此位才有意义。如果选择了单声道模式，则当音频模块用作发送器时，Slot 0 的数据将复制到 Slot 1 上。在接收模式下，将丢弃 Slot 1 并仅存储从 Slot 0 接收的数据。更多详细信息，请参见 *单声道/立体声模式*一节。

0：立体声模式

1：单声道模式

位 11:10 **SYNCEN[1:0]**：同步使能 (Synchronization enable)

这些位将由软件置 1 和清零。必须在音频子模块禁止的情况下配置这些位。

00：音频子模块处于异步模式。

01：音频子模块与另一个内部音频子模块同步。这种情况下，必须将该音频子模块配置为从模式。

10：保留

11：保留

注： 使能 SPDIF 模式后，应将音频子模块配置为异步。

位 9 **CKSTR**：时钟选通边沿 (Clock strobing edge)

此位由软件置 1 和清零。必须在音频模块禁止的情况下配置此位。SPDIF 音频协议下此位没有意义。

0：在 SCK 上升沿更改 SAI 生成的信号，而在 SCK 下降沿对 SAI 接收的信号进行采样。

1：在 SCK 下降沿更改 SAI 生成的信号，而在 SCK 上升沿对 SAI 接收的信号进行采样。

位 8 **LSBFIRST**：最低有效位优先 (Least significant bit first)

此位由软件置 1 和清零。必须在音频模块禁止的情况下配置此位。AC’97 音频协议下此位没有意义，原因是传输 AC’97 数据时，数据的 MSB 位优先。SPDIF 音频协议下此位没有意义，原因是传输 SPDIF 数据时，数据的 LSB 位优先。

0：优先传输数据的 MSB

1：优先传输数据的 LSB

位 7:5 **DS[2:0]**：数据大小 (Data size)

这些位将由软件置 1 和清零。选择 SPDIF 协议时（位 PRTCFG[1:0]），将忽略这些位，原因是在这种情况下帧和数据大小是固定的。通过 COMP[1:0] 位选择压扩模式时，将忽略 DS[1:0]，原因是算法已将数据大小固定为 8 位。

必须在音频模块禁止的情况下配置这些位。

000：保留
001：保留
010：8 位
011：10 位
100：16 位
101：20 位
110：24 位
111：32 位

位 4 保留，必须保持复位值。

位 3:2 **PRTCFG[1:0]**：协议配置 (Protocol configuration)

这些位将由软件置 1 和清零。必须在音频模块禁止的情况下配置这些位。

00：自由协议 通过设置大部分配置寄存器位以及帧配置寄存器，自由协议允许用户使用音频模块这一强大的配置功能来处理特定的音频协议（如 I2S、LSB/MSB 对齐、TDM、PCM/DSP...）。
01：SPDIF 协议
10：AC'97 协议
11：保留

位 1:0 **MODE[1:0]**：SAIx 音频模块模式 (SAIx audio block mode)

这些位将由软件置 1 和清零。必须在 SAIx 音频模块禁止的情况下配置这些位。

00：主发送器
01：主接收器
10：从发送器
11：从接收器

注： 如果将音频模块配置为 SPDIF 模式，则将强制设置主发送模式 (MODE[1:0] = 00)。

36.5.2 配置寄存器 1 (SAI_BCR1)

Configuration register 1

偏移地址：0x024

复位值：0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	MCKEN	OSR	MCKDIV[5:0]						NODIV	Res.	DMAEN	SAIEN
				rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	OUTDRIV	MONO	SYNCEN[1:0]		CKSTR	LSBFIRST	DS[2:0]			Res.	PRTCFG[1:0]		MODE[1:0]	
		rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27 **MCKEN**：主时钟生成使能 (Master clock generation enable)

0：不生成主时钟
1：独立于 SAIEN 位生成主时钟

位 26 **OSR**：主时钟的过采样率 (Oversampling ratio for master clock)
只有在 NODIV 位置 0 时，此位才有意义。
0：主时钟频率 = F_{FS} x 256
1：主时钟频率 = F_{FS} x 512

位 25:20 **MCKDIV[5:0]**：主时钟分频器 (Master clock divider)
这些位将由软件置 1 和清零。
000000：内核时钟输入 (sai_x_ker_ck) 1 分频。
否则，主时钟频率将根据*第 36.3.8 节：SAI 时钟发生器*中的公式计算。
音频模块为从模块时，这些位没有意义。
必须在音频模块禁止的情况下配置这些位。

位 19 **NODIV**：无分频器 (No divider)
此位由软件置 1 和清零。
0：主时钟发生器和帧同步之间的比值固定为 256 或 512
1：主时钟发生器和帧同步之间的比值取决于 FRL[7:0]

位 18 保留，必须保持复位值。

位 17 **DMAEN**：DMA 使能 (DMA enable)
此位由软件置 1 和清零。
0：禁止 DMA
1：使能 DMA
注：　在接收模式下，必须在 DMAEN 位置 1 前配置 MODE[1:0] 位，以避免 DMA 请求，原因是复位后音频模块将默认以发送模式工作。

位 16 **SAIEN**：音频模块使能 (Audio block enable)
此位由软件置 1。
要关闭音频模块，必须由应用软件将此位编程为 0 并轮询此位，直到此位读回 0，这表示该模块已完全禁止。将此位置 1 前，先检查此位是否已置 0，否则不会执行此使能命令。
此位可用于控制 SAI 音频模块的状态。如果在音频帧传输期间禁止此位，则正在进行的传输将继续，直到传输结束时 SAI 音频模块才完全禁止。
0：禁止 SAI 音频模块
1：使能 SAI 音频模块
注：　当 SAI 模块（A 或 B）配置为主模式时，SAI 模块输入中必须有时钟，然后才能将 SAIEN 位置 1。

位 15:14 保留，必须保持复位值。

位 13 **OUTDRIV**：输出驱动 (Output drive)
此位由软件置 1 和清零。
0：当 SAIEN 置 1 时驱动音频模块输出。
1：在此位置 1 后立即驱动音频模块输出。
注：　此位必须在音频模块配置后的使能前置 1。

位 12 **MONO**：单声道模式 (Mono mode)
此位由软件置 1 和清零。仅当 Slot 数为 2 时此位才有意义。如果选择了单声道模式，则当音频模块用作发送器时，Slot 0 的数据将复制到 Slot 1 上。在接收模式下，将丢弃 Slot 1 并仅存储从 Slot 0 接收的数据。更多详细信息，请参见*单声道/立体声模式*一节。
0：立体声模式
1：单声道模式

位 11:10 **SYNCEN[1:0]**：同步使能 (Synchronization enable)

这些位将由软件置 1 和清零。必须在音频子模块禁止的情况下配置这些位。

00：音频子模块处于异步模式。

01：音频子模块与另一个内部音频子模块同步。这种情况下，必须将该音频子模块配置为从模式。

10：保留

11：保留

注： 使能 SPDIF 模式后，应将音频子模块配置为异步。

位 9 **CKSTR**：时钟选通边沿 (Clock strobing edge)

此位由软件置 1 和清零。必须在音频模块禁止的情况下配置此位。SPDIF 音频协议下此位没有意义。

0：在 SCK 上升沿更改 SAI 生成的信号，而在 SCK 下降沿对 SAI 接收的信号进行采样。

1：在 SCK 下降沿更改 SAI 生成的信号，而在 SCK 上升沿对 SAI 接收的信号进行采样。

位 8 **LSBFIRST**：最低有效位优先 (Least significant bit first)

此位由软件置 1 和清零。必须在音频模块禁止的情况下配置此位。AC'97 音频协议下此位没有意义，原因是传输 AC'97 数据时，数据的 MSB 位优先。SPDIF 音频协议下此位没有意义，原因是传输 SPDIF 数据时，数据的 LSB 位优先。

0：优先传输数据的 MSB

1：优先传输数据的 LSB

位 7:5 **DS[2:0]**：数据大小 (Data size)

这些位将由软件置 1 和清零。选择 SPDIF 协议时（位 PRTCFG[1:0]），将忽略这些位，原因是在这种情况下帧和数据大小是固定的。通过 COMP[1:0] 位选择压扩模式时，将忽略 DS[1:0]，原因是算法已将数据大小固定为 8 位。

必须在音频模块禁止的情况下配置这些位。

000：保留

001：保留

010：8 位

011：10 位

100：16 位

101：20 位

110：24 位

111：32 位

位 4 保留，必须保持复位值。

位 3:2 **PRTCFG[1:0]**：协议配置 (Protocol configuration)

这些位将由软件置 1 和清零。必须在音频模块禁止的情况下配置这些位。

00：自由协议 通过设置大部分配置寄存器位以及帧配置寄存器，自由协议允许用户使用音频模块这一强大的配置功能来处理特定的音频协议（如 I2S、LSB/MSB 对齐、TDM、PCM/DSP...）。

01：SPDIF 协议

10：AC'97 协议

11：保留

位 1:0 **MODE[1:0]**：SAIx 音频模块模式 (SAIx audio block mode)

这些位将由软件置 1 和清零。必须在 SAIx 音频模块禁止的情况下配置这些位。

00：主发送器

01：主接收器

10：从发送器

11：从接收器

注： 如果将音频模块配置为 SPDIF 模式，则将强制设置主发送模式 (MODE[1:0] = 00)。在主发送模式下，音频模块将立即开始生成 FS 和时钟。

36.5.3 配置寄存器 2 (SAI_ACR2)

Configuration register 2

偏移地址：0x008

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:14 **COMP[1:0]**：压扩模式 (Companding mode)

这些位将由软件置 1 和清零。μ-Law 和 A-Law 算法是 CCITT G.711 建议的一部分，要使用何种补码类型取决于 *CPL 位*。

数据扩展还是数据压缩由 MODE[0] 位的状态确定。

如果将音频模块配置为发送器，则应用数据压缩。

如果将音频模块配置为接收器，则自动应用数据扩展。

更多详细信息，请参见*压扩模式*一节。

00：不支持压扩算法

01：保留

10：μ-Law 算法

11：A-Law 算法

注： 仅当选择了自由协议模式时才能使用压扩模式。

位 13 **CPL**：补码位 (Complement bit)

此位由软件置 1 和清零。

此位定义用于压扩模式的补码类型。

0：1 的补码表示。

1：2 的补码表示。

注： 仅当压扩模式为 μ-Law 算法或 A-Law 算法时此位才有效。

位 12:7 **MUTECNT[5:0]**：静音计数器 (Mute counter)

这些位将由软件置 1 和清零。这些位仅用于接收模式。

这些位中所设置的值将与接收模式下检测到的连续静音帧数量进行比较。当静音帧数量与该值相等时，MUTEDET 标志置 1，并且在 MUTEDETIE 位置 1 的情况下，还将生成中断。

更多详细信息，请参见*静音模式*一节。

位 6 **MUTEVAL**：静音值 (Mute value)

此位由软件置 1 和清零。必须在使能音频模块 (SAIEN) 前写入。仅当音频模块用作发送器、Slot 数小于或等于 2 并且 MUTE 位置 1 时，此位才有意义。

如果声明了 2 个以上的 Slot，则无论 MUTEVAL 位的值为何，静音模式下发送的位值都将等于 0。

如果 Slot 数小于或等于 2 且 MUTEVAL = 1，则为每个 Slot 发送的 MUTE 值将是上一帧期间发送的值。

更多详细信息，请参见*静音模式*一节。

0：静音模式期间发送位值 0。

1：静音模式期间发送上一个值。

注： 此位对 SPDIF 音频模块无意义，从而也不使用。

位 5 **MUTE**：静音 (Mute)

此位由软件置 1 和清零。仅当音频模块用作发送器时此位才有意义。Slot 数小于或等于 2 时，MUTE 值与 MUTEVAL 值相关；Slot 数大于 2 时，MUTE 值等于 0。

更多详细信息，请参见*静音模式*一节。

0：禁止静音模式。

1：使能静音模式。

注： 此位对 SPDIF 音频模块无意义，从而也不使用。

位 4 **TRIS**：数据线的三态管理 (Tristate management on data line)

此位由软件置 1 和清零。仅当将音频模块配置为发送器时此位才有意义。当音频模块配置为 SPDIF 模式时不使用此位。应在 SAI 禁止时配置此位。

更多详细信息，请参见*无效 Slot 上的输出数据线管理*一节。

0：Slot 无效时，SD 输出线仍由 SAI 驱动。

1：SD 输出线将在上一个有效 Slot（下一个 Slot 无效）的最后一个数据位传输结束时释放（高阻态）。

位 3 **FFLUSH**：FIFO 清空 (FIFO flush)

此位由软件置 1，始终读为 0。应在 SAI 禁止时配置此位。

0：不清空 FIFO。

1：FIFO 清空。将此位编程为 1 可触发 FIFO 清空。所有的内部 FIFO 指针（读和写）将清零。这种情况下，仍存留在 FIFO 中的数据丢失（发送或接收数据不会继续丢失）。FIFO 清空前，必须禁止 DMA 数据流 / 中断。

位 2:0 **FTH[2:0]**：FIFO 阈值 (FIFO threshold)

此位由软件置 1 和清零。

000：FIFO 为空

001：¼ FIFO

010：½ FIFO

011：¾ FIFO

100：FIFO 已满

101：保留

110：保留

111：保留

36.5.4 配置寄存器 2 (SAI_BCR2)

Configuration register 2

偏移地址：0x028

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:14 **COMP[1:0]**：压扩模式 (Companding mode)

这些位将由软件置 1 和清零。μ-Law 和 A-Law 算法是 CCITT G.711 建议的一部分，要使用何种补码类型取决于 *CPL 位*。

数据扩展还是数据压缩由 MODE[0] 位的状态确定。

如果将音频模块配置为发送器，则应用数据压缩。

如果将音频模块配置为接收器，则自动应用数据扩展。

更多详细信息，请参见*压扩模式*一节。

00：不支持压扩算法

01：保留。

10：μ-Law 算法

11：A-Law 算法

注： 仅当选择了自由协议模式时才能使用压扩模式。

位 13 **CPL**：补码位 (Complement bit)

此位由软件置 1 和清零。

此位定义用于压扩模式的补码类型。

0：1 的补码表示。

1：2 的补码表示。

注： 仅当压扩模式为 μ-Law 算法或 A-Law 算法时此位才有效。

位 12:7 **MUTECNT[5:0]**：静音计数器 (Mute counter)

这些位将由软件置 1 和清零。这些位仅用于接收模式。

这些位中所设置的值将与接收模式下检测到的连续静音帧数量进行比较。当静音帧数量与该值相等时，MUTEDET 标志置 1，并且在 MUTEDETIE 位置 1 的情况下，还将生成中断。

更多详细信息，请参见*静音模式*一节。

位 6 **MUTEVAL**：静音值 (Mute value)。

此位由软件置 1 和清零。必须在使能音频模块 (SAIEN) 前写入。仅当音频模块用作发送器、Slot 数小于或等于 2 并且 MUTE 位置 1 时，此位才有意义。

如果声明了 2 个以上的 Slot，则无论 MUTEVAL 位的值为何，静音模式下发送的位值都将等于 0。

如果 Slot 数小于或等于 2 且 MUTEVAL = 1，则为每个 Slot 发送的 MUTE 值将是上一帧期间发送的值。

更多详细信息，请参见*静音模式*一节。

0：静音模式期间发送位值 0。

1：静音模式期间发送上一个值。

注： 此位对 SPDIF 音频模块无意义，从而也不使用。

位 5 **MUTE**：静音 (Mute)。

此位由软件置 1 和清零。仅当音频模块用作发送器时此位才有意义。Slot 数小于或等于 2 时，MUTE 值与 MUTEVAL 值相关；Slot 数大于 2 时，MUTE 值等于 0。

更多详细信息，请参见*静音模式*一节。

0：禁止静音模式。

1：使能静音模式。

注： 此位对 SPDIF 音频模块无意义，从而也不使用。

位 4 **TRIS**：数据线的三态管理 (Tristate management on data line)。

此位由软件置 1 和清零。仅当将音频模块配置为发送器时此位才有意义。当音频模块配置为 SPDIF 模式时不使用此位。应在 SAI 禁止时配置此位。

更多详细信息，请参见*无效 Slot 上的输出数据线管理*一节。

0：Slot 无效时，SD 输出线仍由 SAI 驱动。

1：SD 输出线将在上一个有效 Slot（下一个 Slot 无效）的最后一个数据位传输结束时释放（高阻态）。

位 3 **FFLUSH**：FIFO 清空 (FIFO flush)。

此位由软件置 1，始终读为 0。应在 SAI 禁止时配置此位。

0：不清空 FIFO。

1：FIFO 清空。将此位编程为 1 可触发 FIFO 清空。所有的内部 FIFO 指针（读和写）将清零。这种情况下，仍存留在 FIFO 中的数据丢失（发送或接收数据不会继续丢失）。FIFO 清空前，必须禁止 DMA 数据流 / 中断。

位 2:0 **FTH[2:0]**：FIFO 阈值 (FIFO threshold)。

此位由软件置 1 和清零。

000：FIFO 为空

001：¼ FIFO

010：½ FIFO

011：¾ FIFO

100：FIFO 已满

101：保留

110：保留

111：保留

36.5.5 帧配置寄存器 (SAI_AFRCR)

Frame configuration register

偏移地址：0x00C

复位值：0x0000 0007

注： *该寄存器对于 AC'97 和 SPDIF 音频协议无意义*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSOFF	FSPOL	FSDEF
													rw	rw	r

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

位 31:19 保留，必须保持复位值。

位 18 **FSOFF**：帧同步偏移 (Frame synchronization offset)

此位由软件置 1 和清零。此位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。必须在音频模块禁止的情况下配置此位。

0：在 Slot 0 的第一位上使能 FS。

1：在 Slot 0 第一位的前一位上使能 FS。

位 17 **FSPOL**：帧同步极性 (Frame synchronization polarity)

此位由软件置 1 和清零。此位用于配置 FS 信号上的 SOF 电平。此位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。

必须在音频模块禁止的情况下配置此位。

0：FS 为低电平有效（下降沿）

1：FS 为高电平有效（上升沿）

位 16 **FSDEF**：帧同步定义 (Frame synchronization definition)

此位由软件置 1 和清零。

0：FS 信号为起始帧信号

1：FS 信号为 SOF 信号 + 通道识别信号

此位置 1 时，SAI_xSLOTR 寄存器中定义的 Slot 数必须为偶数。这意味着有半数 Slot 将用于左通道，其他 Slot 用于右通道（例如，对于 I2S 或 MSB/LSB 对齐等协议，此位必须置 1）。

此位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。必须在音频模块禁止的情况下配置此位。

位 15 保留，必须保持复位值。

位 14:8 **FSALL[6:0]**：帧同步有效电平长度 (Frame synchronization active level length)

这些位将由软件置 1 和清零。这些位用于指定音频帧中 FS 信号的有效电平长度，以位时钟数 (SCK) + 1 (FSALL[6:0] + 1) 表示。

这些位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。

必须在音频模块禁止的情况下配置这些位。

位 7:0 **FRL[7:0]**：帧长度 (Frame length)。

这些位将由软件置 1 和清零。这些位用于定义以 SCK 时钟周期数表示的音频帧长度：帧中的位数等于 FRL[7:0] + 1。

音频帧中发送的位数必须大于或等于 8，否则音频模块将出现操作异常。数据大小为 8 位且在 SAI_xSLOTR 寄存器的 NBSLOT[4:0] 中只定义了一个 Slot 0 (NBSLOT[3:0] = 0000) 时便属于这种情况。

在主模式下，如果使用主时钟（MCLK_x 引脚上提供），则帧长度应为 8 到 256 之间的一个等于 2 的几次幂的数。不使用主时钟 (NODIV = 1) 时，建议将帧长度编程为 8 到 256 之间的值。

这些位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。必须在音频模块禁止的情况下配置这些位。

36.5.6 帧配置寄存器 (SAI_BFRCR)

Frame configuration register

偏移地址：0x02C

复位值：0x0000 0007

注： *该寄存器对于 AC'97 和 SPDIF 音频协议无意义*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSOFF	FSPOL	FSDEF
													rw	rw	r

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

位 31:19 保留，必须保持复位值。

位 18 **FSOFF**：帧同步偏移 (Frame synchronization offset)

此位由软件置 1 和清零。此位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。必须在音频模块禁止的情况下配置此位。

0：在 Slot 0 的第一位上使能 FS。

1：在 Slot 0 第一位的前一位上使能 FS。

位 17 **FSPOL**：帧同步极性 (Frame synchronization polarity)

此位由软件置 1 和清零。此位用于配置 FS 信号上的 SOF 电平。此位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。

必须在音频模块禁止的情况下配置此位。

0：FS 为低电平有效（下降沿）

1：FS 为高电平有效（上升沿）

位 16 **FSDEF**：帧同步定义 (Frame synchronization definition)

此位由软件置 1 和清零。

0：FS 信号为起始帧信号

1：FS 信号为 SOF 信号 + 通道识别信号

此位置 1 时，SAI_xSLOTR 寄存器中定义的 Slot 数必须为偶数。这意味着有半数 Slot 将用于左通道，其他 Slot 用于右通道（例如，对于 I2S 或 MSB/LSB 对齐等协议，此位必须置 1）。

此位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。必须在音频模块禁止的情况下配置此位。

位 15 保留，必须保持复位值。

位 14:8 **FSALL[6:0]**：帧同步有效电平长度 (Frame synchronization active level length)
这些位将由软件置 1 和清零。这些位用于指定音频帧中 FS 信号的有效电平长度，以位时钟数 (SCK) + 1 (FSALL[6:0] + 1) 表示。
这些位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。
必须在音频模块禁止的情况下配置这些位。

位 7:0 **FRL[7:0]**：帧长度 (Frame length)
这些位将由软件置 1 和清零。这些位用于定义以 SCK 时钟周期数表示的音频帧长度：帧中的位数等于 FRL[7:0] + 1。
音频帧中发送的位数必须大于或等于 8，否则音频模块将出现操作异常。数据大小为 8 位且在 SAI_xSLOTR 寄存器的 NBSLOT[4:0] 中只定义了一个 Slot 0 (NBSLOT[3:0] = 0000) 时便属于这种情况。
在主模式下，如果使用主时钟（MCLK_x 引脚上提供），则帧长度应为 8 到 256 之间的一个等于 2 的几次幂的数。不使用主时钟 (NODIV = 1) 时，建议将帧长度编程为 8 到 256 之间的值。
这些位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。

36.5.7 Slot 寄存器 (SAI_ASLOTR)

Slot register

偏移地址：0x010

复位值：0x0000 0000

注： *该寄存器对于 AC'97 和 SPDIF 音频协议无意义*

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

位 31:16 **SLOTEN[15:0]**：Slot 使能 (Slot enable)
这些位将由软件置 1 和清零。
每个 SLOTEN 位都对应于从 0 到 15 的一个 Slot 位置（最多 16 个 Slot）。
0：无效 Slot。
1：有效 Slot。
必须在音频模块禁止的情况下使能 Slot。
在 AC'97 或 SPDIF 模式下会忽略这些位。

位 15:12 保留，必须保持复位值。

位 11:8 **NBSLOT[3:0]**：音频帧中的 Slot 数 (Number of slots in an audio frame)
这些位将由软件置 1 和清零。
此位域中设置的值表示音频帧中的 Slot 数 + 1（包括无效 Slot 数）。Slot 数最大值为 16。
SAI_xFRCR 寄存器中的 FSDEF 位置 1 时，Slot 数应为偶数。
必须在音频模块禁止的情况下配置 Slot 数。
在 AC'97 或 SPDIF 模式下会忽略这些位。

位 7:6 **SLOTSZ[1:0]**：Slot 大小 (Slot size)
此位由软件置 1 和清零。
Slot 大小必须大于或等于数据大小。如果不满足该条件，SAI 的行为将不确定。
有关如何驱动 SD 线的信息，请参见*无效 Slot 上的输出数据线管理*一节。
必须在音频模块禁止的情况下配置这些位。
在 AC'97 或 SPDIF 模式下会忽略这些位。
00：Slot 大小与数据大小（在 SAI_xCR1 寄存器的 DS[3:0] 位中指定）相当。
01：16 位
10：32 位
11：保留

位 5 保留，必须保持复位值。

位 4:0 **FBOFF[4:0]**：第一个位偏移 (First bit offset)
这些位将由软件置 1 和清零。
此位域中设置的值定义 Slot 中第一个数据传输位的位置。它表示一个偏移值。在发送模式下，此数据位域以外的位将强制清零。在接收模式下，将丢弃额外接收的位。
必须在音频模块禁止的情况下配置这些位。
在 AC'97 或 SPDIF 模式下会忽略这些位。

36.5.8 Slot 寄存器 (SAI_BSLOTR)

Slot register

偏移地址：0x030

复位值：0x0000 0000

注： *该寄存器对于 AC'97 和 SPDIF 音频协议无意义*

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

位 31:16 **SLOTEN[15:0]**：Slot 使能 (Slot enable)
这些位将由软件置 1 和清零。
每个 SLOTEN 位都对应于从 0 到 15 的一个 Slot 位置（最多 16 个 Slot）。
0：无效 Slot。
1：有效 Slot。
必须在音频模块禁止的情况下使能 Slot。
在 AC'97 或 SPDIF 模式下会忽略这些位。

位 15:12 保留，必须保持复位值。

位 11:8 **NBSLOT[3:0]**：音频帧中的 Slot 数 (Number of slots in an audio frame)
这些位将由软件置 1 和清零。
此位域中设置的值表示音频帧中的 Slot 数 + 1（包括无效 Slot 数）。Slot 数最大值为 16。
SAI_xFRCR 寄存器中的 FSDEF 位置 1 时，Slot 数应为偶数。
必须在音频模块禁止的情况下配置 Slot 数。
在 AC'97 或 SPDIF 模式下会忽略这些位。

位 7:6 **SLOTSZ[1:0]**：Slot 大小 (Slot size)
此位由软件置 1 和清零。
Slot 大小必须大于或等于数据大小。如果不满足该条件，SAI 的行为将不确定。
有关如何驱动 SD 线的信息，请参见*无效 Slot 上的输出数据线管理*一节。
必须在音频模块禁止的情况下配置这些位。
在 AC'97 或 SPDIF 模式下会忽略这些位。
00：Slot 大小与数据大小（在 SAI_xCR1 寄存器的 DS[3:0] 位中指定）相当。
01：16 位
10：32 位
11：保留

位 5 保留，必须保持复位值。

位 4:0 **FBOFF[4:0]**：第一个位偏移 (First bit offset)
这些位将由软件置 1 和清零。
此位域中设置的值定义 Slot 中第一个数据传输位的位置。它表示一个偏移值。在发送模式下，此数据位域以外的位将强制清零。在接收模式下，将丢弃额外接收的位。
必须在音频模块禁止的情况下配置这些位。
在 AC'97 或 SPDIF 模式下会忽略这些位。

36.5.9 中断屏蔽寄存器 (SAI_AIM)

Interrupt mask register

偏移地址：0x014

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET IE	AFSDET IE	CNRDY IE	FREQ IE	WCKCFG IE	MUTEDET IE	OVRUDR IE
									rw	rw	rw	rw	rw	rw	rw

位 31:7 保留，必须保持复位值。

位 6 **LFSDETIE**：帧同步滞后检测中断使能 (Late frame synchronization detection interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
此位置 1 时，若 SAI_xSR 寄存器中的 LFSDET 位置 1，则生成中断。
此位对于 AC'97、SPDIF 模式无意义；若音频模块为主模块，此位也无意义。

位 5 **AFSDETIE**：帧同步提前检测中断使能 (Anticipated frame synchronization detection interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
此位置 1 时，若 SAI_xSR 寄存器中的 AFSDET 位置 1，则生成中断。
此位对于 AC'97、SPDIF 模式无意义；若音频模块为主模块，此位也无意义。

位 4 **CNRDYIE**：编解码器未就绪中断使能 (Codec not ready interrupt enable) (AC'97)

此位由软件置 1 和清零。

0：禁止中断

1：使能中断

若使能该中断，音频模块将在 AC'97 帧的 Slot 0 (tag0) 中检测连接到该线路的编解码器是否就绪。如果未就绪，SAI_xSR 寄存器中的 CNRDY 标志置 1，并生成中断。

仅当通过 PRTCFG[1:0] 位选择了 AC'97 模式且音频模块用作接收器时，此位才有意义。

位 3 **FREQIE**：FIFO 请求中断使能 (FIFO request interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：使能中断

此位置 1 时，若 SAI_xSR 寄存器中的 FREQ 位置 1，则生成中断。

在接收模式下，必须在 FREQIE 位置 1 前配置 MODE 位，以避免寄生中断，原因是复位后音频模块将默认用作发送器。

位 2 **WCKCFGIE**：时钟配置错误中断使能 (Wrong clock configuration interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：使能中断

仅在将音频模块配置为主模式 (MODE[1] = 0) 且 NODIV = 0 时才执行此位。

此位在 SAI_xSR 寄存器中的 WCKCFG 标志置 1 时生成中断。

注：　此位仅用于自由协议模式，其他模式下没有意义。

位 1 **MUTEDETIE**：静音检测中断使能 (Mute detection interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：使能中断

此位置 1 时，若 SAI_ASR 寄存器中的 MUTEDET 位置 1，则生成中断。

仅当音频模块配置为以发送模式工作时此位才有意义。

位 0 **OVRUDRIE**：上溢/下溢中断使能 (Overrun/underrun interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：使能中断

此位置 1 时，若 SAI_ASR 寄存器中的 OVRUDR 位置 1，则生成中断。

36.5.10 中断屏蔽寄存器 (SAI_BIM)

Interrupt mask register

偏移地址：0x034

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET IE	AFSDET IE	CNRDY IE	FREQ IE	WCKCFG IE	MUTEDET IE	OVRUDR IE
									rw	rw	rw	rw	rw	rw	rw

位 31:7 保留，必须保持复位值。

位 6 **LFSDETIE**：帧同步滞后检测中断使能 (Late frame synchronization detection interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
此位置 1 时，若 SAI_xSR 寄存器中的 LFSDET 位置 1，则生成中断。
此位对于 AC'97、SPDIF 模式无意义；若音频模块为主模块，此位也无意义。

位 5 **AFSDETIE**：帧同步提前检测中断使能 (Anticipated frame synchronization detection interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
此位置 1 时，若 SAI_xSR 寄存器中的 AFSDET 位置 1，则生成中断。
此位对于 AC'97、SPDIF 模式无意义；若音频模块为主模块，此位也无意义。

位 4 **CNRDYIE**：编解码器未就绪中断使能 (Codec not ready interrupt enable) (AC'97)
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
若使能该中断，音频模块将在 AC'97 帧的 Slot 0 (tag0) 中检测连接到该线路的编解码器是否就绪。
如果未就绪，SAI_xSR 寄存器中的 CNRDY 标志置 1，并生成中断。
仅当通过 PRTCFG[1:0] 位选择了 AC'97 模式且音频模块用作接收器时，此位才有意义。

位 3 **FREQIE**：FIFO 请求中断使能 (FIFO request interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
此位置 1 时，若 SAI_xSR 寄存器中的 FREQ 位置 1，则生成中断。
在接收模式下，必须在 FREQIE 位置 1 前配置 MODE 位，以避免寄生中断，原因是复位后音频模块将默认用作发送器。

位 2 **WCKCFGIE**：时钟配置错误中断使能 (Wrong clock configuration interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
仅在将音频模块配置为主模式 (MODE[1] = 0) 且 NODIV = 0 时才执行此位。
此位在 SAI_xSR 寄存器中的 WCKCFG 标志置 1 时生成中断。
注： 此位仅用于自由协议模式，其他模式下没有意义。

位 1 **MUTEDETIE**：静音检测中断使能 (Mute detection interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
此位置 1 时，若 SAI_ASR 寄存器中的 MUTEDET 位置 1，则生成中断。
仅当音频模块配置为以发送模式工作时此位才有意义。

位 0 **OVRUDRIE**：上溢/下溢中断使能 (Overrun/underrun interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
此位置 1 时，若 SAI_ASR 寄存器中的 OVRUDR 位置 1，则生成中断。

36.5.11 状态寄存器 (SAI_ASR)

Status register

偏移地址：0x018

复位值：0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLVL[2:0]		
													r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR
									r	r	r	r	r	r	r

位 31:19 保留，必须保持复位值。

位 18:16 **FLVL[2:0]**：FIFO 阈值电压 (FIFO level threshold)

此位为只读位。FIFO 阈值标志只通过硬件管理，其设置取决于 SAI 模块的配置（发送器或接收器模式）。

如果将 SAI 模块配置为发送器：

000：FIFO 为空

001：FIFO <= ¼，但非空

010：¼ < FIFO <= ½

011：½ < FIFO <= ¾

100：¾ < FIFO，但未满

101：FIFO 已满

如果将 SAI 模块配置为接收器：

000：FIFO 为空

001：FIFO < ¼，但非空

010：¼ <= FIFO < ½

011：½ =< FIFO < ¾

100：¾ =< FIFO，但未满

101：FIFO 已满

位 15:7 保留，必须保持复位值。

位 6 **LFSDET**：帧同步滞后检测 (Late frame synchronization detection)

此位为只读位。

0：无错误。

1：帧同步信号未在正确的时刻出现。

仅当音频模块配置为以从模式工作时，此标志才能置 1。

不适用于 AC’97 或 SPDIF 模式。

它可在 SAI_xIM 寄存器中的 LFSDETIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CLFSDET 位置 1 时清除该标志。

位 5 **AFSDET**：帧同步提前检测 (Anticipated frame synchronization detection)

此位为只读位。

0：无错误。

1：提前检测到帧同步信号。

仅当音频模块配置为以从模式工作时，此标志才能置 1。

不适用于 AC’97 或 SPDIF 模式。

它可在 SAI_xIM 寄存器中的 AFSDETIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CAFSDET 位置 1 时清除该标志。

位 4 **CNRDY**：编解码器未就绪 (Codec not ready)

此位为只读位。

0：外部 AC'97 编解码器已就绪

1：外部 AC'97 编解码器未就绪

仅当在 SAI_xCR1 寄存器中选择了 AC'97 音频模块并且音频模块配置为接收器模式时，才使用此位。

它可在 SAI_xIM 寄存器中的 CNRDYIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CCNRDY 位置 1 时清除该标志。

位 3 **FREQ**：FIFO 请求 (FIFO request)

此位为只读位。

0：无 FIFO 请求。

1：FIFO 请求读取或写入 SAI_xDR。

请求内容取决于音频模块的配置：

– 如果模块配置为发送模式，则 FIFO 请求与向 SAI_xDR 中写入相关。

– 如果音频模块配置为接收模式，则 FIFO 请求与从 SAI_xDR 中读取相关。

此标志可在 SAI_xIM 寄存器中的 FREQIE 位置 1 时生成中断。

位 2 **WCKCFG**：时钟配置错误标志 (Wrong clock configuration flag)

此位为只读位。

0：时钟配置正确

1：时钟配置不符合*第 36.3.6 节：帧同步*中定义的帧长度规范（SAI_xFRCR 寄存器中 FRL[7:0] 位的配置）

此位仅在音频模块工作在主模式 (MODE[1] = 0) 下且 NODIV = 0 时使用。

它可在 SAI_xIM 寄存器中的 WCKCFGIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CWCKCFG 位置 1 时清除该标志。

位 1 **MUTEDET**：静音检测 (Mute detection)

此位为只读位。

0：SD 输入线上未检测到 MUTE 值

1：在SD 输入线上检测到指定数量的连续音频帧中的 MUTE 值（0 值）

如果在指定音频帧的每个 Slot 或在一定数量（在 SAI_xCR2 寄存器中的 MUTECNT 位中设置）的连续音频帧中接收到连续的 0 值，则该标志置 1。

它可在 SAI_xIM 寄存器中的 MUTEDETIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CMUTEDET 位置 1 时清除该标志。

位 0 **OVRUDR**：上溢/下溢 (Overrun/underrun)

此位为只读位。

0：无上溢/下溢错误。

1：检测到上溢/下溢错误。

仅当音频模块分别被配置为接收器和发送器时才会发生上溢和下溢情况。

它可在 SAI_xIM 寄存器中的 OVRUDRIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 COVRUDR 位置 1 时清除该标志。

36.5.12 状态寄存器 (SAI_BSR)

Status register

偏移地址：0x038

复位值：0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLVL[2:0]		
													r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR
									r	r	r	r	r	r	r

位 31:19 保留，必须保持复位值。

位 18:16 **FLVL[2:0]**：FIFO 阈值电压 (FIFO level threshold)

此位为只读位。FIFO 阈值标志只通过硬件管理，其设置取决于 SAI 模块的配置（发送器或接收器模式）。

如果将 SAI 模块配置为发送器：

000：FIFO 为空

001：FIFO <= ¼，但非空

010：¼ < FIFO <= ½

011：½ < FIFO <= ¾

100：¾ < FIFO，但未满

101：FIFO 已满

如果将 SAI 模块配置为接收器：

000：FIFO 为空

001：FIFO < ¼，但非空

010：¼ <= FIFO < ½

011：½ =< FIFO < ¾

100：¾ =< FIFO，但未满

101：FIFO 已满

位 15:7 保留，必须保持复位值。

位 6 **LFSDET**：帧同步滞后检测 (Late frame synchronization detection)

此位为只读位。

0：无错误。

1：帧同步信号未在正确的时刻出现。

仅当音频模块配置为以从模式工作时，此标志才能置 1。

不适用于 AC'97 或 SPDIF 模式。

它可在 SAI_xIM 寄存器中的 LFSDETIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CLFSDET 位置 1 时清除该标志。

位 5 **AFSDET**：帧同步提前检测 (Anticipated frame synchronization detection)

此位为只读位。

0：无错误。

1：提前检测到帧同步信号。

仅当音频模块配置为以从模式工作时，此标志才能置 1。

不适用于 AC'97 或 SPDIF 模式。

它可在 SAI_xIM 寄存器中的 AFSDETIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CAFSDET 位置 1 时清除该标志。

位 4 **CNRDY**：编解码器未就绪 (Codec not ready)

此位为只读位。

0：外部 AC'97 编解码器已就绪

1：外部 AC'97 编解码器未就绪

仅当在 SAI_xCR1 寄存器中选择了 AC'97 音频模块并且音频模块配置为接收器模式时，才使用此位。

它可在 SAI_xIM 寄存器中的 CNRDYIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CCNRDY 位置 1 时清除该标志。

位 3 **FREQ**：FIFO 请求 (FIFO request)

此位为只读位。

0：无 FIFO 请求。

1：FIFO 请求读取或写入 SAI_xDR。

请求内容取决于音频模块的配置：

– 如果模块配置为发送模式，则 FIFO 请求与向 SAI_xDR 中写入相关。

– 如果音频模块配置为接收模式，则 FIFO 请求与从 SAI_xDR 中读取相关。

此标志可在 SAI_xIM 寄存器中的 FREQIE 位置 1 时生成中断。

位 2 **WCKCFG**：时钟配置错误标志 (Wrong clock configuration flag)

此位为只读位。

0：时钟配置正确

1：时钟配置不符合*第 36.3.6 节：帧同步*中定义的帧长度规范（SAI_xFRCR 寄存器中 FRL[7:0] 位的配置）

此位仅在音频模块工作在主模式 (MODE[1] = 0) 下且 NODIV = 0 时使用。

它可在 SAI_xIM 寄存器中的 WCKCFGIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CWCKCFG 位置 1 时清除该标志。

位 1 **MUTEDET**：静音检测 (Mute detection)

此位为只读位。

0：SD 输入线上未检测到 MUTE 值

1：在SD 输入线上检测到指定数量的连续音频帧中的 MUTE 值（0 值）

如果在指定音频帧的每个 Slot 或在一定数量（在 SAI_xCR2 寄存器中的 MUTECNT 位中设置）的连续音频帧中接收到连续的 0 值，则该标志置 1。

它可在 SAI_xIM 寄存器中的 MUTEDETIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CMUTEDET 位置 1 时清除该标志。

位 0 **OVRUDR**：上溢/下溢 (Overrun/underrun)

此位为只读位。

0：无上溢/下溢错误。

1：检测到上溢/下溢错误。

仅当音频模块分别被配置为接收器和发送器时才会发生上溢和下溢情况。

它可在 SAI_xIM 寄存器中的 OVRUDRIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 COVRUDR 位置 1 时清除该标志。

36.5.13 清除标志寄存器 (SAI_ACLRFR)

Clear flag register

偏移地址：0x01C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLFSDET	CAFSDET	CCNRDY	Res.	CWCKCFG	CMUTEDET	COVRUDR
									w	w	w		w	w	w

位 31:7 保留，必须保持复位值。

位 6 **CLFSDET**：清除帧同步滞后检测标志 (Clear late frame synchronization detection flag)
此位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 LFSDET 标志。
此位不适用于 AC’97 或 SPDIF 模式
读取此位将始终返回值 0。

位 5 **CAFSDET**：清除帧同步提前检测标志 (Clear anticipated frame synchronization detection flag)
此位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 AFSDET 标志。
不适用于 AC’97 或 SPDIF 模式。
读取此位将始终返回值 0。

位 4 **CCNRDY**：清除编解码器未就绪标志 (Clear codec not ready flag)
此位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 CNRDY 标志。
仅当在 SAI_xCR1 寄存器中选择了 AC’97 音频协议时，才使用此位。
读取此位将始终返回值 0。

位 3 保留，必须保持复位值。

位 2 **CWCKCFG**：清除时钟配置错误标志 (Clear wrong clock configuration flag)
此位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 WCKCFG 标志。
仅当音频模块设置为主模块时 (MODE[1] = 0) 且 SAI_xCR1 寄存器中的 NODIV = 0 时，才使用此位。
读取此位将始终返回值 0。

位 1 **CMUTEDET**：清除静音检测标志 (Clear mute detection flag)
此位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 MUTEDET 标志。
读取此位将始终返回值 0。

位 0 **COVRUDR**：清除上溢/下溢标志 (Clear overrun/underrun)
此位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 OVRUDR 标志。
读取此位将始终返回值 0。

36.5.14 清除标志寄存器 (SAI_BCLRFR)

Clear flag register

偏移地址：0x03C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLFSDET	CAFSDET	CCNRDY	Res.	CWCKCFG	CMUTEDET	COVRUDR
									w	w	w		w	w	w

位 31:7 保留，必须保持复位值。

位 6 **CLFSDET**：清除帧同步滞后检测标志 (Clear late frame synchronization detection flag)
此位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 LFSDET 标志。
此位不适用于 AC’97 或 SPDIF 模式。
读取此位将始终返回值 0。

位 5 **CAFSDET**：清除帧同步提前检测标志 (Clear anticipated frame synchronization detection flag)
此位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 AFSDET 标志。
不适用于 AC’97 或 SPDIF 模式。
读取此位将始终返回值 0。

位 4 **CCNRDY**：清除编解码器未就绪标志 (Clear codec not ready flag)
此位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 CNRDY 标志。
仅当在 SAI_xCR1 寄存器中选择了 AC’97 音频协议时，才使用此位。
读取此位将始终返回值 0。

位 3 保留，必须保持复位值。

位 2 **CWCKCFG**：清除时钟配置错误标志 (Clear wrong clock configuration flag)
此位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 WCKCFG 标志。
仅当音频模块设置为主模块时 (MODE[1] = 0) 且 SAI_xCR1 寄存器中的 NODIV = 0 时，才使用此位。
读取此位将始终返回值 0。

位 1 **CMUTEDET**：清除静音检测标志 (Clear mute detection flag)
此位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 MUTEDET 标志。
读取此位将始终返回值 0。

位 0 **COVRUDR**：清除上溢/下溢标志 (Clear overrun/underrun)
此位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 OVRUDR 标志。
读取此位将始终返回值 0。

36.5.15 数据寄存器 (SAI_ADR)

Data register

偏移地址：0x020

复位值：0x0000 0000

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

位 31:0 **DATA[31:0]**：数据 (Data)

若 FIFO 未满，写入该寄存器可向 FIFO 加载数据。

若 FIFO 非空，读取该寄存器可清空 FIFO 。

36.5.16 数据寄存器 (SAI_BDR)

Data register

偏移地址：0x040

复位值：0x0000 0000

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

位 31:0 **DATA[31:0]**：数据 (Data)

若 FIFO 未满，写入该寄存器可向 FIFO 加载数据。

若 FIFO 非空，读取该寄存器可清空 FIFO 。

36.5.17 PDM 控制寄存器 (SAI_PDMCR)

PDM control register

偏移地址：0x0044

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKEN2	CKEN1	Res.	Res.	MICNBR[1:0]		Res.	Res.	Res.	PDMEN
						rw	rw			rw	rw				rw

位 31:16 保留，必须保持复位值。

位 15:10 保留，必须保持复位值。

位 9 **CKEN2**：2 号比特流时钟使能 (Clock enable of bitstream clock number 2)
此位由软件置 1 和清零。
0：禁止 SAI_CK2 时钟
1：使能 SAI_CK2 时钟
注： 建议不要在 PDMEN = 1 时配置此位。

位 8 **CKEN1**：1 号比特流时钟使能 (Clock enable of bitstream clock number 1)
此位由软件置 1 和清零。
0：禁止 SAI_CK1 时钟
1：使能 SAI_CK1 时钟
注： 建议不要在 PDMEN = 1 时配置此位。

位 7:6 保留，必须保持复位值。

位 5:4 **MICNBR[1:0]**：麦克风数量 (Number of microphones)
此位由软件置 1 和清零。
00：配置有 2 个麦克风
01：配置有 4 个麦克风
10：配置有 6 个麦克风
11：配置有 8 个麦克风
*注： 建议不要在 PDMEN = 1 时配置该字段。**

位 3:1 保留，必须保持复位值。

位 0 **PDMEN**：PDM 使能 (PDM enable)
此位由软件置 1 和清零。此位可用于控制 PDM 接口模块的状态。
在使能 PDM 接口之前，确保 SAI 已工作在 TDM 主模式下。
0：禁止 PDM 接口
1：使能 PDM 接口

36.5.18 PDM 延迟寄存器 (SAI_PDMDLY)

PDM delay register

偏移地址：0x0048

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DLYM4R[2:0]			Res.	DLYM4L[2:0]			Res.	DLYM3R[2:0]			Res.	DLYM3L[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DLYM2R[2:0]			Res.	DLYM2L[2:0]			Res.	DLYM1R[2:0]			Res.	DLYM1L[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

位 31 保留，必须保持复位值。

位 30:28 **DLYM4R[2:0]**：**第 4 对**的第二个麦克风的延迟线 (Delay line for second microphone of **pair 4**)

此位由软件置 1 和清零。

000：无延迟

001：延迟 1 个 T_{SAI_CK} 周期

010：延迟 2 个 T_{SAI_CK} 周期

...

111：延迟 7 个 T_{SAI_CK} 周期

此位域可实时更改。

位 27 保留，必须保持复位值。

位 26:24 **DLYM4L[2:0]**：第 4 对的第一个麦克风的延迟线 (Delay line for first microphone of pair 4)

此位由软件置 1 和清零。

000：无延迟

001：延迟 1 个 T_{SAI_CK} 周期

010：延迟 2 个 T_{SAI_CK} 周期

...

111：延迟 7 个 T_{SAI_CK} 周期

此位域可实时更改。

位 23 保留，必须保持复位值。

位 22:20 **DLYM3R[2:0]**：第 3 对的第二个麦克风的延迟线 (Delay line for second microphone of pair 3)

此位由软件置 1 和清零。

000：无延迟

001：延迟 1 个 T_{SAI_CK} 周期

010：延迟 2 个 T_{SAI_CK} 周期

...

111：延迟 7 个 T_{SAI_CK} 周期

此位域可实时更改。

位 19 保留，必须保持复位值。

位 18:16 **DLYM3L[2:0]**：第 3 对的第一个麦克风的延迟线 (Delay line for first microphone of pair 3)
此位由软件置 1 和清零。
000：无延迟
001：延迟 1 个 T_{SAI_CK} 周期
010：延迟 2 个 T_{SAI_CK} 周期
...
111：延迟 7 个 T_{SAI_CK} 周期

此位域可实时更改。

位 15 保留，必须保持复位值。

位 14:12 **DLYM2R[2:0]**：第 2 对的第二个麦克风的延迟线 (Delay line for second microphone of pair 2)
此位由软件置 1 和清零。
000：无延迟
001：延迟 1 个 T_{SAI_CK} 周期
010：延迟 2 个 T_{SAI_CK} 周期
...
111：延迟 7 个 T_{SAI_CK} 周期

此位域可实时更改。

位 11 保留，必须保持复位值。

位 10:8 **DLYM2L[2:0]**：第 2 对的第一个麦克风的延迟线 (Delay line for first microphone of pair 2)
此位由软件置 1 和清零。
000：无延迟
001：延迟 1 个 T_{SAI_CK} 周期
010：延迟 2 个 T_{SAI_CK} 周期
...
111：延迟 7 个 T_{SAI_CK} 周期

此位域可实时更改。

位 7 保留，必须保持复位值。

位 6:4 **DLYM1R[2:0]**：第 1 对的第二个麦克风的延迟线调整 (Delay line adjust for second microphone of pair 1)
此位由软件置 1 和清零。
000：无延迟
001：延迟 1 个 T_{SAI_CK} 周期
010：延迟 2 个 T_{SAI_CK} 周期
...
111：延迟 7 个 T_{SAI_CK} 周期

此位域可实时更改。

位 3 保留，必须保持复位值。

位 2:0 **DLYM1L[2:0]**：第 1 对的第一个麦克风的延迟线调整 (Delay line adjust for first microphone of pair 1)
此位由软件置 1 和清零。
000：无延迟
001：延迟 1 个 T_{SAI_CK} 周期
010：延迟 2 个 T_{SAI_CK} 周期
...
111：延迟 7 个 T_{SAI_CK} 周期

此位域可实时更改。

36.5.19 SAI 寄存器映射

下表对 SAI 寄存器进行了汇总。

表 218. SAI 寄存器映射和复位值

偏移	寄存器名称	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
0x0000	Reserved																																
0x0004 or 0x0024	SAI_xCR1	Res.	Res.	Res.	Res.	MCKEN	OSR	MCKDIV[5:0]						NODIV	Res.	DMAEN	SAIEN	Res.	Res.	OUTDRIV	MONO	SYNCEN[1:0]		CKSTR	LSBFIRST	DS[2:0]			Res.	PRTCFG[1:0]		MODE[1:0]	
	Reset value					0	0	0	0	0	0	0	0	0		0	0			0	0	0	0	0	0	0	1	0		0	0	0	0
0x0008 or 0x0028	SAI_xCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP[1:0]		CPL	MUTECN[5:0]						MUTE VAL	MUTE	TRIS	FFLUS	FTH[2:0]		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x000C or 0x002C	SAI_xFRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSOFF	FSPOL	FSDEF	Res.	FSALL[6:0]							FRL[7:0]							
	Reset value														0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x0010 or 0x0030	SAI_xSLOTR	SLOTEN[15:0]																Res.	Res.	Res.	Res.	NBSLOT[3:0]				SLOTSZ[1:0]		Res.	FBOFF[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
0x0014 or 0x0034	SAI_xIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDETIE	AFSDETIE	CNRDYIE	FREQIE	WCKCFGIE	MUTEDETIE	OVRUDRIE
	Reset value																										0	0	0	0	0	0	0
0x0018 or 0x0038	SAI_xSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLVL[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR
	Reset value														0	0	0										0	0	0	0	1	0	0
0x001C or 0x003C	SAI_xCLRFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLFSDET	CAFSDET	CCNRDY	Res.	CWCKCFG	CMUTEDET	COVRUDR
	Reset value																										0	0	0		0	0	0
0x0020 or 0x0040	SAI_xDR	DATA[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
0x0044	SAI_PDMCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKEN2	CKEN1	Res.	Res.	MICNBR[1:0]		Res.	Res.	Res.	PDMEN
	Reset value																							0	0			0	0				0

表 218. SAI 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x0048	SAI_PDMDLY	Res.	DLYM4R[2:0]			Res.	DLYM4L[2:0]			Res.	DLYM3R[2:0]			Res.	DLYM3L[2:0]			Res.	DLYM2R[2:0]			Res.	DLYM2L[2:0]			Res.	DLYM1R[2:0]			Res.	DLYM1L[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

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

37 处理器间通信控制器 (IPCC)

37.1 IPCC 简介

处理器间通信控制器 (IPCC) 用于两个处理器之间的数据通信。

IPCC 模块提供了一种非阻塞信号传输机制，能够以原子方式发布和检索通信数据。它可为 12 个通道提供信号传输：

- 6 个通道在从处理器 1 到处理器 2 的方向上
- 6 个通道在从处理器 2 到处理器 1 的方向上

这两个方向上均可实现两种不同类型的通信。

IPCC 通信数据必须位于通用存储器中，该存储器不是 IPCC 模块的一部分。

37.2 IPCC 主要特性

- 为 12 个通道传输状态信号
  - 通道占用/空闲标志，也用作锁定标志
- 每个处理器有两条中断线
  - 一条用于占用的 RX 通道（通过发送处理器发布通信数据）
  - 一条用于空闲的 TX 通道（通过接收处理器检索通信数据）
- 按通道屏蔽中断
  - 通道占用屏蔽
  - 通道空闲屏蔽
- 两种通道工作模式
  - 单工（每个通道都有自己的通信数据存储单元）
  - 半双工（每个通道都与一个双向通信数据信息存储单元相关联）

37.3 IPCC 功能说明

IPCC 通信数据位于通用存储器中，该存储器不是 IPCC 模块的一部分。通信数据的地址单元应是已知的或位于已知的通用区域中，如前文所述，该通用区域不是 IPCC 模块的一部分。

对于每次通信，IPCC 模块均提供通道状态标志 CHnF。

- 通道状态标志 CHnF 为 0 时，表示相关的 IPCC 通道空闲（通信数据已由接收处理器检索），并且可由发送处理器访问。
- 通道状态标志 CHnF 为 1 时，表示相关的 IPCC 通道已占用（即，通信数据已由发送处理器发布），并且可由接收处理器访问。

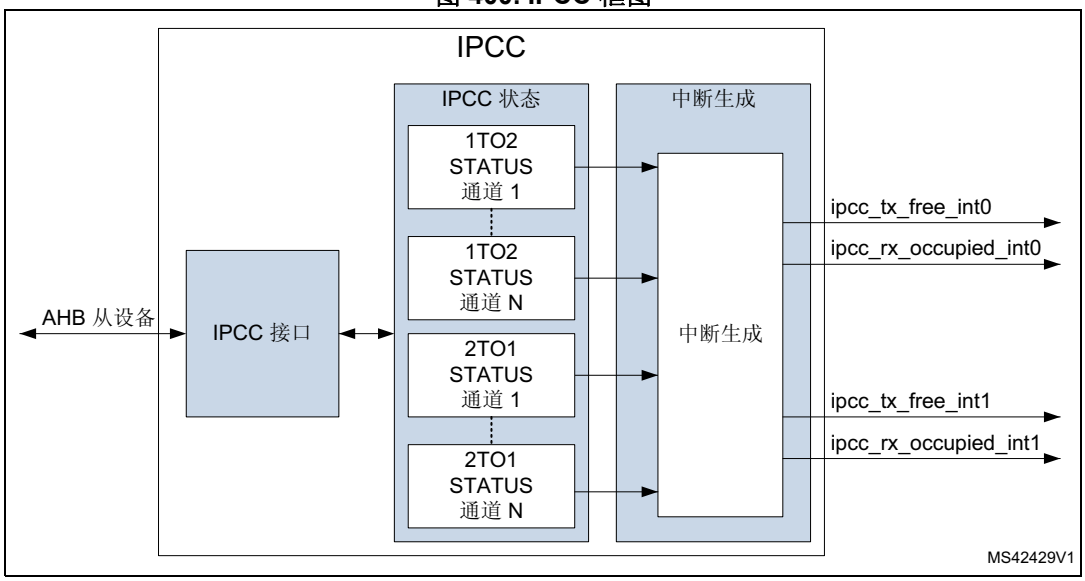
两个处理器均应已知通道工作模式。通用参数可用于指示通道传输模式，并且还应位于已知的通用区域中。IPCC 无法提供此参数。

37.3.1 IPCC 框图

IPCC（请参见*图 400*）模块包括以下子模块：

- 状态模块，包含通道状态
- IPCC 接口模块，提供对通道状态寄存器的 AHB 访问
- 中断接口模块，提供对中断的控制

图 400. IPCC 框图



37.3.2 IPCC 单工通道模式

在单工通道模式下，将为通信数据分配一个专用存储单元（用于在单个方向上传输数据）。相关的通道 N 控制位（请参见*表 219*）用于管理从发送处理器到接收处理器的数据传输。

表 219. 用于通信的位

处理器	A	B
SEND A = 1 RECEIVE B = 2	IPCC_C1CR.TXFIE IPCC_C1MR.CHnFM IPCC_C1SCR.CHnS IPCC_C1TOC2SR.CHnF	IPCC_C2CR.RXOIE IPCC_C2MR.CHnOM IPCC_C2SCR.CHnC
SEND A = 2 RECEIVE B = 1	IPCC_C2CR.TXFIE IPCC_C2MR.CHnFM IPCC_C2SCR.CHnS IPCC_C2TOC1SR.CHnF	IPCC_C1CR.RXOIE IPCC_C1MR.CHnOM IPCC_C1SCR.CHnC

发送处理器将通信数据发布到存储器中后，会立即通过 CHnS 将通道状态标志 CHnF 置 1，以将通道设置为占用状态。

接收处理器从存储器中检索到通信数据后，会立即通过 CHnC 将通道状态标志 CHnF 清零，以将通道恢复为空闲状态。

图 401. IPCC 单工通道模式传输时序

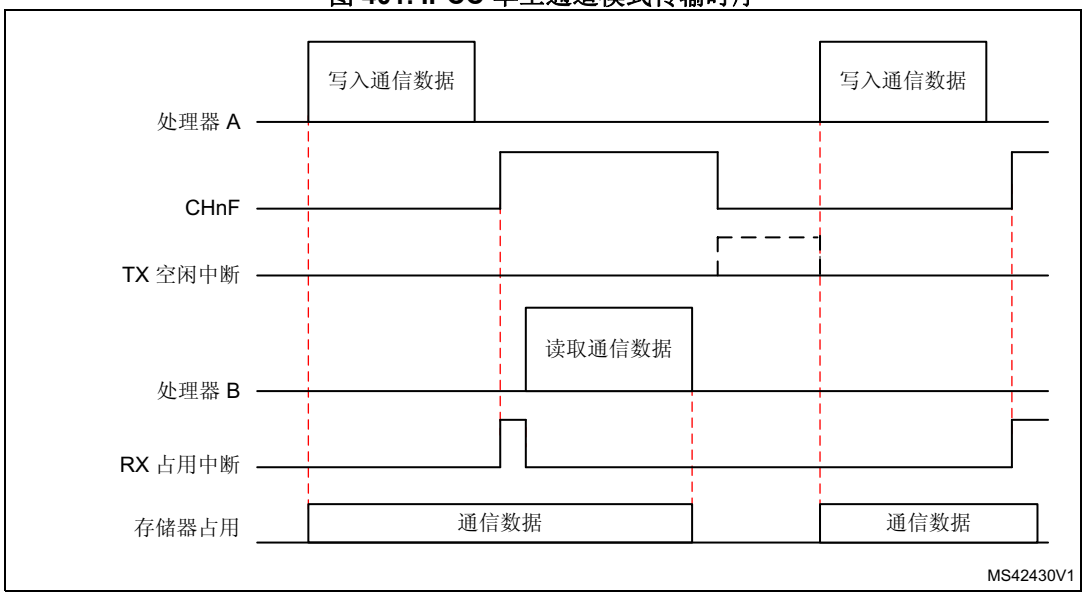
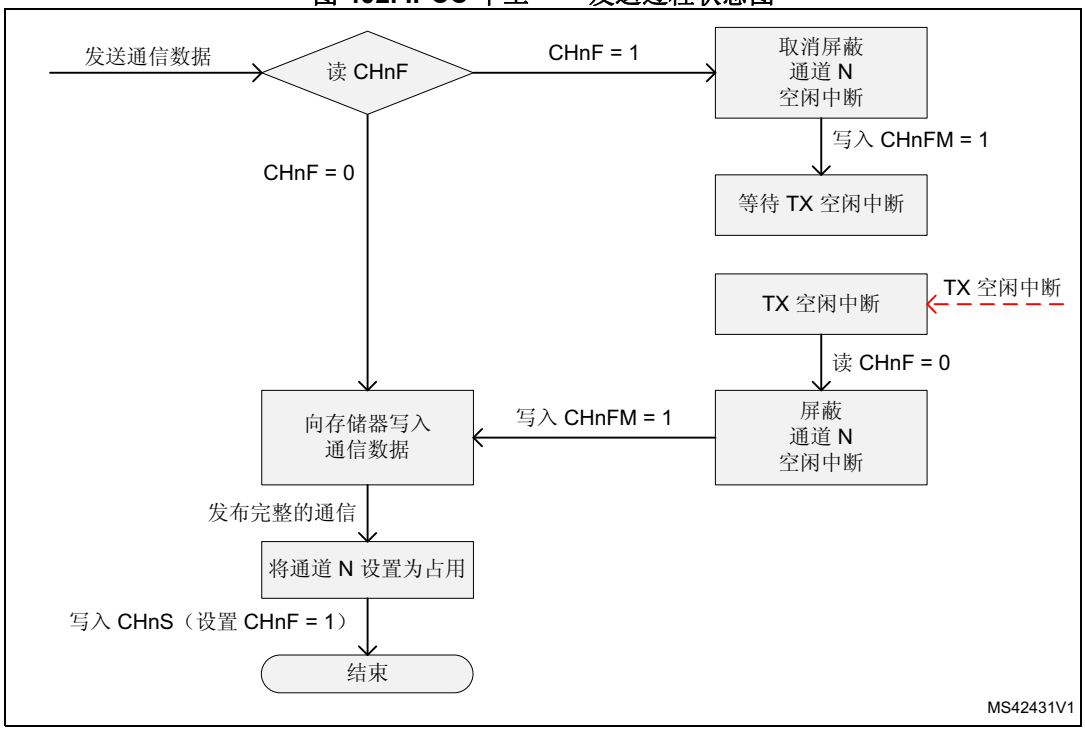


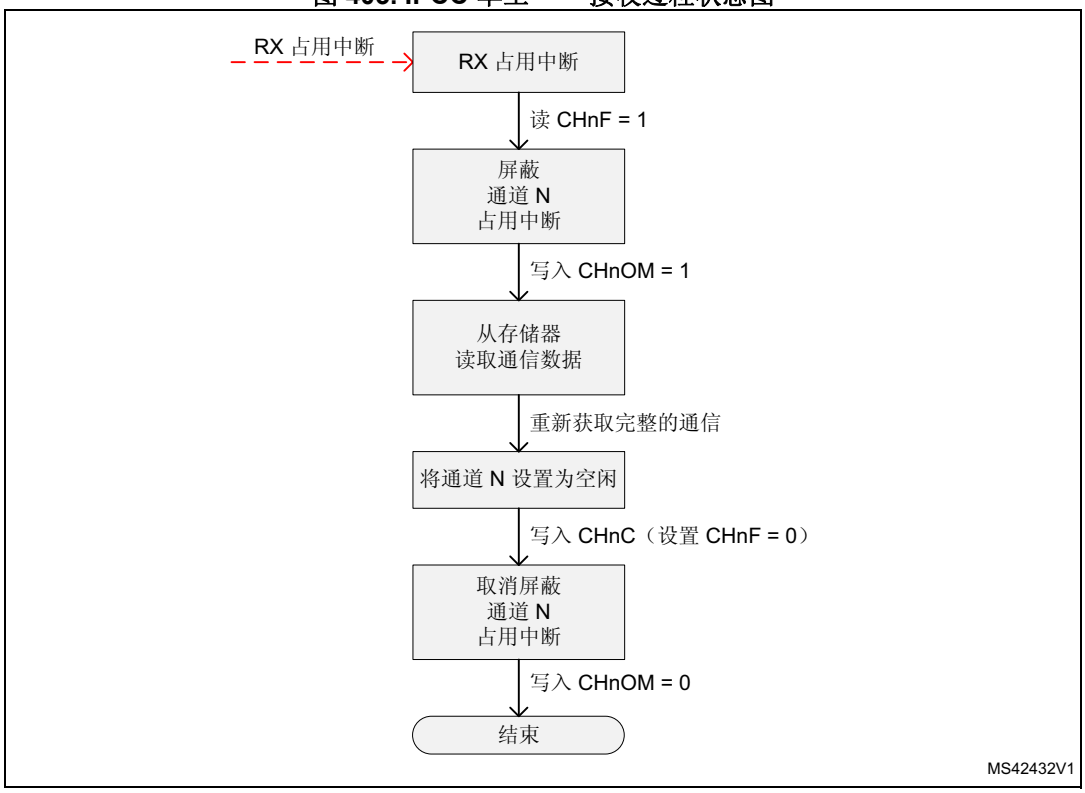
图 402. IPCC 单工——发送过程状态图



发送通信数据：

- 发送处理器将检查通道状态标志 CHnF
  - 该标志为 0 时，通道空闲（接收处理器检索到最后的通信数据），可以写入新的通信数据。
  - 该标志为 1 时，通道被占用（接收处理器未检索到最后的通信数据），发送处理器将取消屏蔽通道空闲中断 (CHnFM = 0)。
  - 发生 TX 空闲中断时，发送处理器将检查哪个通道变为空闲，并屏蔽相关的通道空闲中断 (CHnFM = 1)，然后即可进行新的通信。
- 发布完整的通信数据后，会立即通过 CHnS 将通道设置为占用状态：这会将存储器访问权限授予接收处理器，并产生 RX 占用中断。

图 403. IPCC 单工——接收过程状态图



要接收通信数据，需取消屏蔽通道占用中断 (CHnOM = 0)：

- 发生 RX 占用中断时，接收处理器将检查哪个通道已被占用，并屏蔽相关的通道占用中断 (CHnOM)，然后从存储器读取通信数据。
- 检索完整的通信数据后，会立即通过 CHnC 将通道状态清除为空闲。（这会将存储器访问权限重新授予接收处理器，并会产生 TX 空闲中断）。
- 清除通道状态后，将取消屏蔽通道占用中断 (CHnOM = 0)。

37.3.3 IPCC 半双工通道模式

当一个处理器发送通信数据，而另一个处理器对此发送响应时，使用半双工通道模式（乒乓）。

在半双工通道模式下，将为通信数据和响应分配一个专用存储单元，用于双向数据传输。发送处理器通道状态标志 CHnF 分配给通道，由两个处理器共用（请参见*表 219*）。

处理器 A 将通信数据发布到存储器中后，会立即通过 CHnS 将处理器 A 通道状态标志 CHnF 置 1，以将通道设置为占用状态（将存储器访问权限授予处理器 B）。

处理器 B 从存储器检索到通信数据后，不会改变通道状态标志。处理器 B 保留存储器访问权限以进行响应。

处理器 B 将响应发布到存储器中后，会立即通过 CHnC 将通道状态标志 CHnF 清零，以将通道设置为空闲状态（将存储器访问权限重新授予处理器 A）。

处理器 A 从存储器检索到响应后，不会改变通道状态标志。处理器 A 保留存储器单元访问权限以用于下一次数据通信。

图 404. IPCC 半双工通道模式传输时序

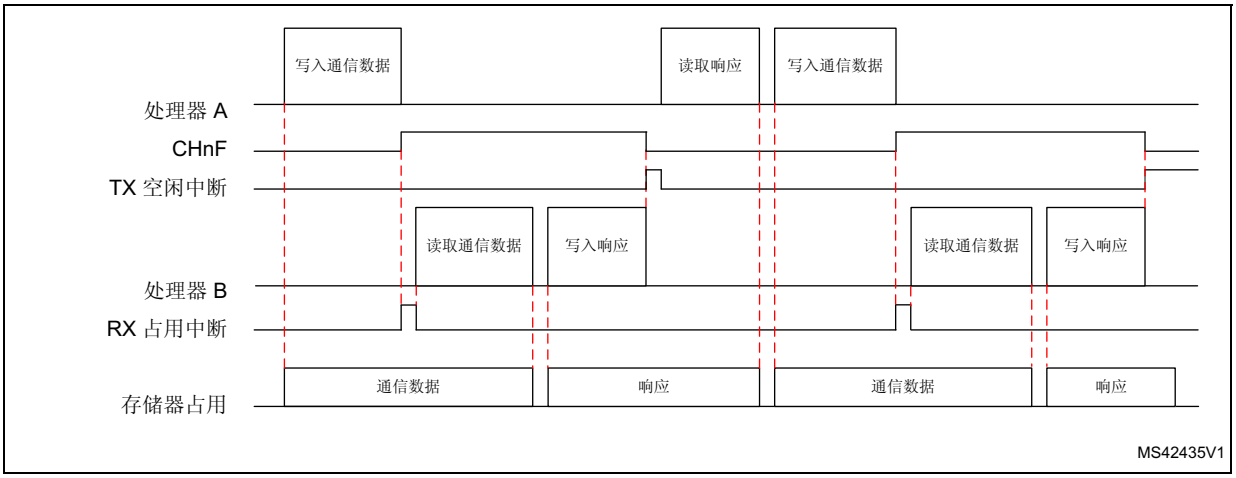
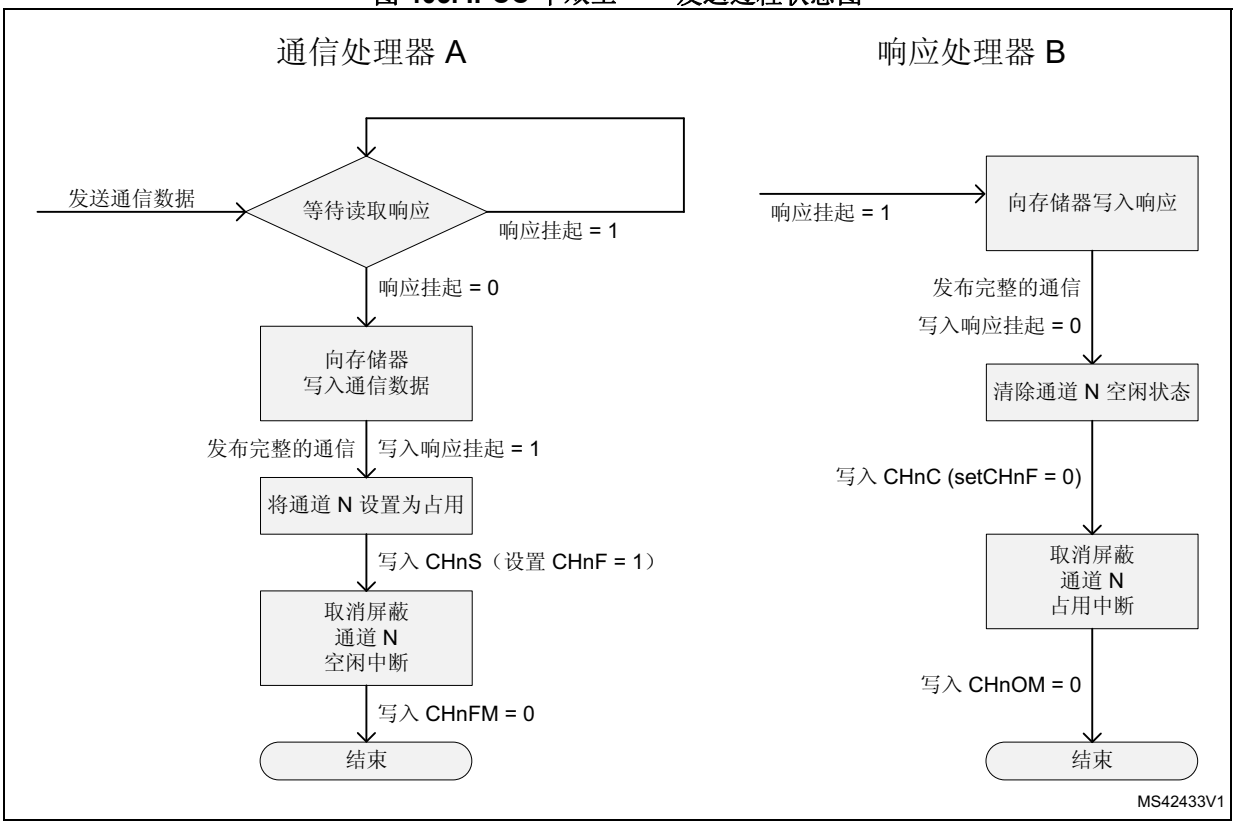


图 405. IPCC 半双工——发送过程状态图



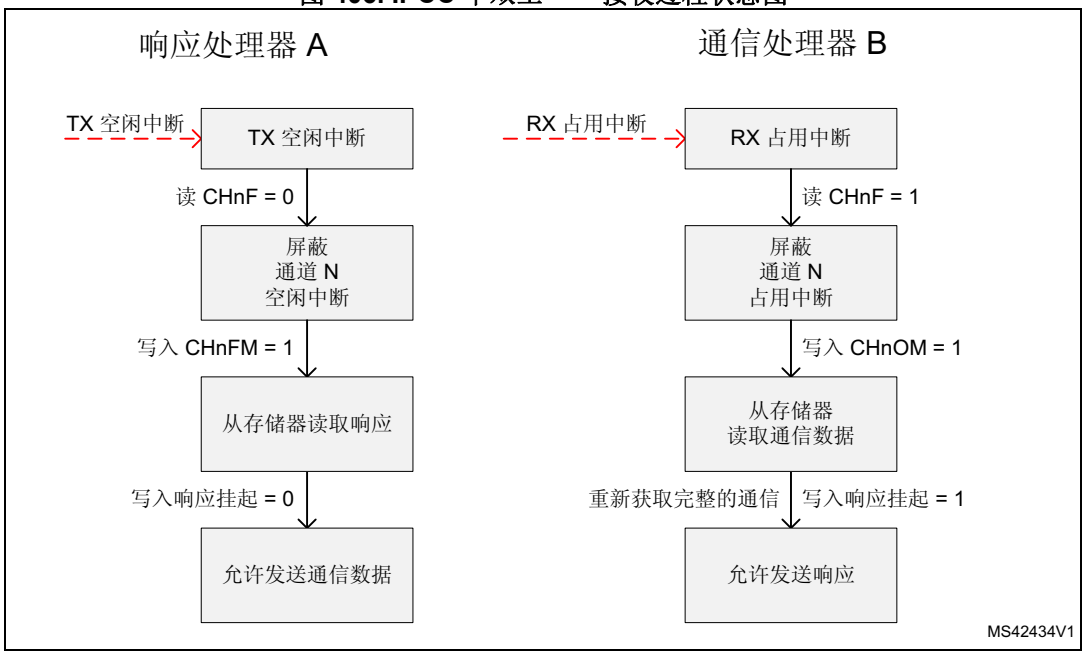
发送通信数据：

- 发送处理器将等待其响应挂起 FW 变量变为 0
  - 响应挂起 FW 变量为 0 后，将发布通信数据。
- 发布完整的通信数据后，会通过 CHnS 将通道状态标志 CHnF 置 1，以将通道设置为占用状态，响应挂起 FW 变量将设置为 1（这会将存储器访问权限授予接收处理器，并产生 RX 占用中断）。
- 通道状态标志 CHnF 置 1 后，将取消屏蔽通道空闲中断 (CHnFM = 0)。

发送响应：

- 接收处理器将等待其响应挂起 FW 变量变为 1。
  - 响应挂起 FW 变量为 1 后，将发布响应。
- 发布完整的响应后，会通过 CHnC 将通道状态标志 CHnF 清零，以将通道设置为空闲状态，响应挂起 FW 变量将设置为 0（这会将存储器访问权限授予发送处理器，并产生 TX 空闲中断）。
- 通道状态标志 CHnF 清零后，将取消屏蔽通道占用中断 (CHnOM = 0)。

图 406. IPCC 半双工——接收过程状态图



要接收通信数据，需取消屏蔽通道占用中断 (CHnOM = 0)：

- 发生 RX 占用中断时，接收处理器将检查哪个通道已被占用，并屏蔽相关的通道占用中断 (CHnOM)，然后从存储器读取通信数据。
- 检索到完整的通信数据后，响应挂起 FW 变量将置 1。接收处理器不会更改通道状态，并会保留存储器访问权限，以便发布后续响应。

要接收响应，需取消屏蔽通道空闲中断 (CHnFM = 0)：

- 发生 TX 空闲中断时，发送处理器将检查哪个通道变为空闲，并屏蔽相关通道空闲中断 (CHnFM)，然后从存储器读取响应。
- 检索到完整的响应后，响应挂起 FW 变量将清零。发送处理器不会更改通道状态，并会保留存储器访问权限，以便发布后续通信数据。

37.3.4 IPCC 中断

共有 4 条中断线：

- 2 个 RX 通道占用中断，每个处理器各 1 个
  - 中断使能 RXOIE（每个处理器各 1 个）
  - 单独屏蔽 CHnOM（每个通道各 1 个）
- 2 个 TX 通道空闲中断，每个处理器各 1 个
  - 中断使能 TXFIE（每个处理器各 1 个）
  - 单独屏蔽 CHnFM（每个通道各 1 个）

RX 占用中断由接收处理器使用，在取消屏蔽的通道状态指示占用 (CHnF = 1) 时出现。

TX 空闲中断由发送处理器使用，在取消屏蔽的通道状态指示空闲 (CHnF = 0) 时出现。

37.4 IPCC 寄存器

外设寄存器必须按字（32 位）进行访问。不允许字节（8 位）和半字（16 位）访问，并且不会产生总线错误。

37.4.1 IPCC 处理器 1 控制寄存器 (IPCC_C1CR)

IPCC Processor 1 control register

偏移地址：0x000

复位值：0x0000 0000

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

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

位 31:17 保留，必须保持复位值。

位 16 **TXFIE**：处理器 1 发送通道空闲中断使能 (Processor 1 Transmit channel free interrupt enable)。与 IPCC_C1TOC2SR 相关。

1：使取消屏蔽的处理器 1 发送通道变为空闲状态，以产生 TX 空闲中断。

0：禁止处理器 1 TX 空闲中断。

位 15:1 保留，必须保持复位值。

位 0 **RXOIE**：处理器 1 接收通道占用中断使能 (Processor 1 Receive channel occupied interrupt enable)。与 IPCC_C2TOC1SR 相关。

1：使取消屏蔽的处理器 1 发送通道变为占用状态，以产生 RX 占用中断。

0：禁止处理器 1 RX 占用中断。

37.4.2 IPCC 处理器 1 屏蔽寄存器 (IPCC_C1MR)

IPCC Processor 1 mask register

偏移地址：0x004

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6FM	CH5FM	CH4FM	CH3FM	CH2FM	CH1FM
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6OM	CH5OM	CH4OM	CH3OM	CH2OM	CH1OM
										rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:16 **CH[6:1]FM**：处理器 1 发送通道 x 状态置 1 (Processor 1 transmit channel x status set)。与 IPCC_C1TOC2SR.CHxF (x = n - 15) 相关。

1：屏蔽发送通道 x 空闲中断。

0：取消屏蔽发送通道 x 空闲中断。

位 15:6 保留，必须保持复位值。

位 5:0 **CH[6:1]OM**：处理器 1 接收通道 x 状态清零 (Processor 1 Receive channel x status clear)。与 IPCC_C2TOC1SR.CHxF (x = n + 1) 相关。

1：屏蔽接收通道 x 占用中断。

0：取消屏蔽接收通道 x 占用中断。

37.4.3 IPCC 处理器 1 状态置位清零寄存器 (IPCC_C1SCR)

IPCC Processor 1 status set clear register

偏移地址：0x008

复位值：0x0000 0000

读取该寄存器将始终返回 0x0000 0000。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6S	CH5S	CH4S	CH3S	CH2S	CH1S
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6C	CH5C	CH4C	CH3C	CH2C	CH1C
										rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:16 **CH[6:1]S**：处理器 1 发送通道 x 状态置 1 (Processor 1 transmit channel x status set)。与 IPCC_C1TOC2SR.CHxF (x = n - 15) 相关。

1：处理器 1 发送通道 x 状态位置 1。

0：不执行任何操作。

位 15:6 保留，必须保持复位值。

位 5:0 **CH[6:1]C**：处理器 1 接收通道 x 状态清零 (Processor 1 Receive channel x status clear)。与 IPCC_C2TOC1SR.CHxF (x = n + 1) 相关。

1：处理器 1 接收通道 x 状态位清零。

0：不执行任何操作。

37.4.4 IPCC 处理器 1 到处理器 2 状态寄存器 (IPCC_C1TOC2SR)

IPCC processor 1 to processor 2 status register

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6F	CH5F	CH4F	CH3F	CH2F	CH1F
										r	r	r	r	r	r

位 31:6 保留，必须保持复位值。

位 5:0 **CH[6:1]F**：屏蔽前处理器 1 发送到处理器 2 接收通道 x 状态标志 (Processor 1 transmit to processor 2 receive channel x status flag before masking) (x = n + 1)

1：通道占用，接收处理器 2 可以读取数据。

0：通道空闲，发送处理器 1 可以写入数据。

37.4.5 IPCC 处理器 2 控制寄存器 (IPCC_C2CR)

IPCC Processor 2 control register

偏移地址：0x010

复位值：0x0000 0000

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

位 31:17 保留，必须保持复位值。

位 16 **TXFIE**：处理器 2 发送通道空闲中断使能 (Processor 2 Transmit channel free interrupt enable)。与 IPCC_C2TOC1SR 相关。

1：使取消屏蔽的处理器 2 发送通道变为空闲状态，以产生 TX 空闲中断。

0：禁止处理器 2 TX 空闲中断。

位 15:1 保留，必须保持复位值。

位 0 **RXOIE**：处理器 2 接收通道占用中断使能 (Processor 2 Receive channel occupied interrupt enable)。与 IPCC_C1TOC2SR 相关。

1：使取消屏蔽的处理器 2 接收通道变为占用状态，以产生 RX 占用中断。

0：禁止处理器 2 RX 占用中断。

37.4.6 IPCC 处理器 2 屏蔽寄存器 (IPCC_C2MR)

IPCC Processor 2 mask register

偏移地址：0x014

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6FM	CH5FM	CH4FM	CH3FM	CH2FM	CH1FM
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6OM	CH5OM	CH4OM	CH3OM	CH2OM	CH1OM
										rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:16 **CH[6:1]FM**：处理器 2 发送通道 x 空闲中断屏蔽 (Processor 2 Transmit channel x free interrupt mask)。与 IPCC_C2TOC1SR.CHxF (x = n - 15) 相关。

1：屏蔽发送通道 x 空闲中断。

0：取消屏蔽发送通道 x 空闲中断。

位 15:6 保留，必须保持复位值。

位 5:0 **CH[6:1]OM**：处理器 2 接收通道 x 占用中断屏蔽 (Processor 2 receive channel x occupied interrupt mask)。与 IPCC_C1TOC2SR.CHxF (x = n + 1) 相关。

1：屏蔽接收通道 x 占用中断。

0：取消屏蔽接收通道 x 占用中断。

37.4.7 IPCC 处理器 2 状态置位清零寄存器 (IPCC_C2SCR)

IPCC Processor 2 status set clear register

偏移地址：0x018

复位值：0x0000 0000

读取该寄存器将始终返回 0x0000 0000。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6S	CH5S	CH4S	CH3S	CH2S	CH1S
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6C	CH5C	CH4C	CH3C	CH2C	CH1C
										rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:16 **CH[6:1]S**：处理器 2 发送通道 x 状态置 1 (Processor 2 transmit channel x status set) 与 IPCC_C2TOC1SR.CHxF (x = n - 15) 相关。

1：处理器 2 发送通道 x 状态位置 1。

0：不执行任何操作。

位 15:6 保留，必须保持复位值。

位 5:0 **CH[6:1]C**：处理器 2 接收通道 x 状态清零 (Processor 2 receive channel x status clear) 与 IPCC_C1TOC2SR.CHxF (x = n + 1) 相关。

1：处理器 2 接收通道 x 状态位清零。

0：不执行任何操作。

37.4.8 IPCC 处理器 2 到处理器 1 状态寄存器 (IPCC_C2TOC1SR)

IPCC processor 2 to processor 1 status register

偏移地址：0x01C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6F	CH5F	CH4F	CH3F	CH2F	CH1F
										r	r	r	r	r	r

位 31:6 保留，必须保持复位值。

位 5:0 **CH[6:1]F**：屏蔽前处理器 2 发送到处理器 1 接收通道 x 状态标志 (Processor 2 transmit to processor 1 receive channel x status flag before masking) (x = n + 1)

1：通道占用，接收处理器 1 可以读取数据。

0：通道空闲，发送处理器 2 可以写入数据。

37.4.9 IPCC 寄存器映射和复位值表

表 220. IPCC 寄存器映射和复位值

偏移	寄存器名称	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
0x0000	**IPCC_C1CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXOIE
	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
0x0004	**IPCC_C1MR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6FM	CH5FM	CH4FM	CH3FM	CH2FM	CH1FM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6OM	CH5OM	CH4OM	CH3OM	CH2OM	CH1OM
	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
0x0008	**IPCC_C1SCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6S	CH5S	CH4S	CH3S	CH2S	CH1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6C	CH5C	CH4C	CH3C	CH2C	CH1C
	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
0x000C	**IPCC_C1TOC2SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6F	CH5F	CH4F	CH3F	CH2F	CH1F
	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
0x0010	**IPCC_C2CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXOIE
	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
0x0014	**IPCC_C2MR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6FM	CH5FM	CH4FM	CH3FM	CH2FM	CH1FM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6OM	CH5OM	CH4OM	CH3OM	CH2OM	CH1OM
	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
0x0018	**IPCC_C2SCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6S	CH5S	CH4S	CH3S	CH2S	CH1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6C	CH5C	CH4C	CH3C	CH2C	CH1C
	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
0x001C	**IPCC_C2TOC1SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH6F	CH5F	CH4F	CH3F	CH2F	CH1F
	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

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

38 硬件信号量 (HSEM)

38.1 硬件信号量简介

硬件信号量模块提供 32 个（32 位）基于寄存器的信号量。

信号量可以用于确保在不同内核间运行的不同进程之间的同步。HSEM 提供一个非阻塞机制以原子方式来锁定信号量。其提供了下列功能：

- 可通过以下 2 种方式锁定信号量：
  - 2 步锁定：将 COREID 和 PROCID 写入信号量，然后进行读取检查
  - 1 步锁定：从信号量读取 COREID
- 当信号量被释放时生成中断
  - 每个信号量可以在其中一条中断线上生成中断
- 信号量清零保护
  - 只有当 COREID 与 PROCID 匹配时，才会将信号量清零
- 根据 COREID 将全局信号量清零

38.2 硬件信号量的主要特性

HSEM 包括下列特性：

- 32 个（32 位）信号量
- 8 位 PROCID
- 4 位 COREID
- 2 条中断线
- 锁定指示

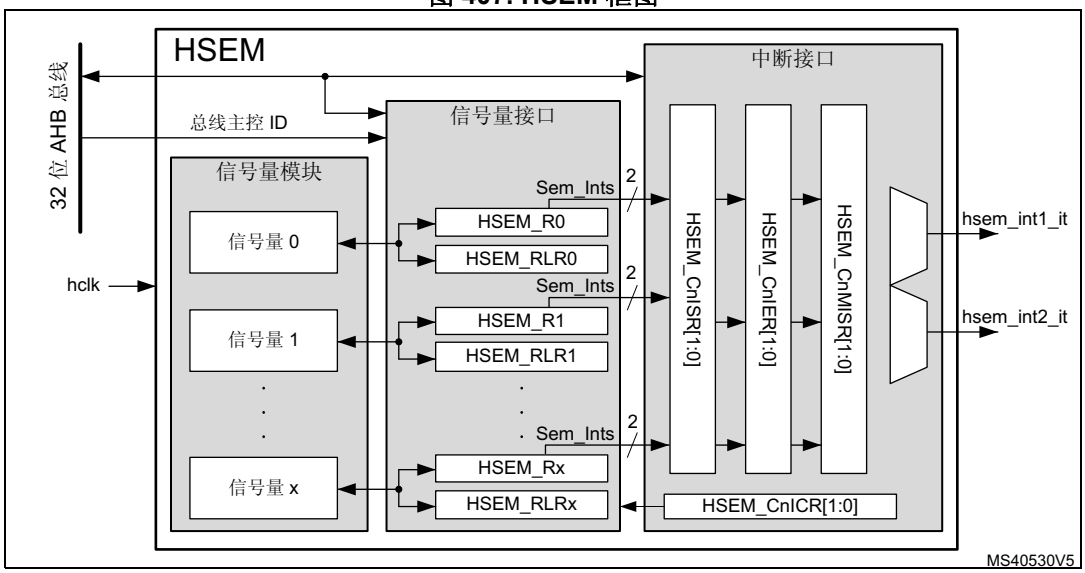
38.3 HSEM 功能说明

38.3.1 HSEM 框图

如*图 407* 所示，HSEM 基于以下三个子模块：

- 包含信号量状态和 ID 的信号量模块
- 通过 HSEM_Rx 和 HSEM_RLRx 寄存器对信号量进行 AHB 访问的信号量接口模块
- 通过 HSEM_CnISR、HSEM_CnIER、HSEM_CnMISR 和 HSEM_CnICR 寄存器提供中断控制的中断接口模块

图 407. HSEM 框图



38.3.2 HSEM 内部信号

表 221. HSEM 内部输入/输出信号

信号名称	信号类型	说明
hsem_hclk	数字输入	AHB 时钟
hsem_int1_it	数字输出	中断 1 线
hsem_int2_it	数字输出	中断 2 线

38.3.3 HSEM 锁定步骤

共有 2 种锁定步骤

- 2 步（写）锁定
- 1 步（读）锁定

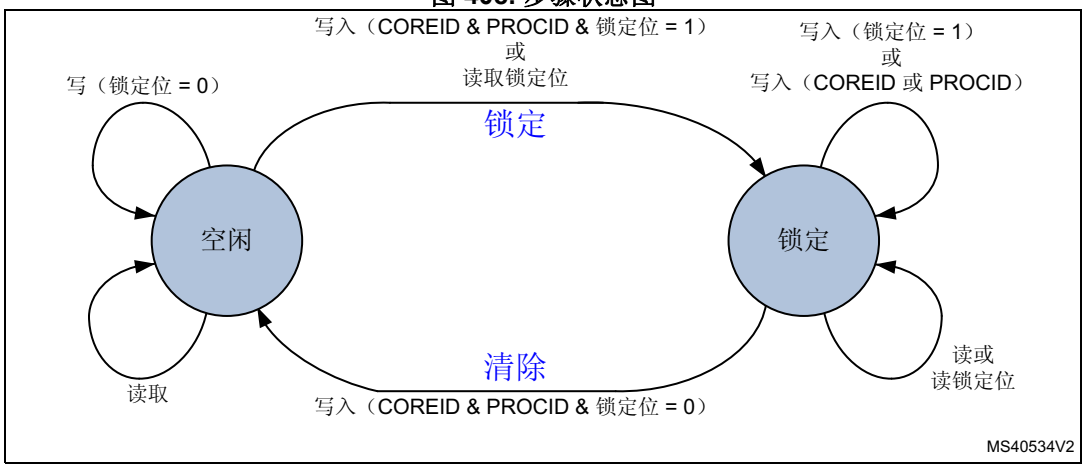
当锁定位为 0 时，信号量空闲，在这种情况下，COREID 与 PROCID 也为 0。当锁定位为 1 时，信号量锁定，COREID 指示是哪一个 AHB 总线主控将其锁定。PROCID 指示是 AHB 总线主控的哪一个进程锁定了信号量。

当“写”锁定信号量时，COREID 取自主控 ID，PROCID 取自“写”数据。当“读”锁定信号量时，COREID 取自 AHB 总线主控 ID，PROCID 将为零。1 步（读）锁定不存在 PROCID。

COREID 取自 AHB 总线主控 ID。PROCID 通过相应 AHB 总线主控的固件写入。每个 AHB 总线主控进程必须有唯一的 PROCID。只有 2 步锁定步骤才存在 PROCID。

这两种步骤（1 步和 2 步）可以同时使用。

图 408. 步骤状态图



2 步（写）锁定步骤

2 步锁定步骤由两步组成，首先通过“写”以锁定信号量，然后通过“读”HSEM_Rx 寄存器以检查是否成功锁定。

- 将 PROCID 和 COREID 以及锁定位 = 1 写入信号量
  （如果在写操作时信号量空闲，则锁定信号量）
- 回读信号量
  （固件检查锁定状态，如果 PROCID 与 COREID 匹配，则锁定被确认）
- 否则重试（信号量已由其他 AHB 总线主控或进程锁定）

信号量只能在空闲时才能锁定。

信号量可在 PROCID 为“0”时锁定。

连续的将锁定位 = 1 写到已锁定信号量的尝试都被忽略。

1 步（读）锁定步骤

1 步步骤由单步组成，通过读取 HSEM_RLRx 寄存器以锁定并检查信号量。

- 读锁定信号量的 COREID。
- 如果读 COREID 匹配且 PROCID = 0，则锁定信号量（如果 COREID 匹配，但 PROCID 不为“0”，则意味着来自同一 COREID 的另一进程使用 2 步（写）步骤锁定了信号量）。
- 否则重试（信号量已由其他 AHB 总线主控或进程锁定）

信号量只能在空闲时才能锁定。当读锁定一个空闲信号量时，PROCID 将为“0”。当读锁定一个已锁定信号量时，将返回锁定它的 COREID 和 PROCID。所有读锁定（包括锁定信号量的第一次读锁定）都将返回锁定信号量的 COREID。

如果同一 AHB 总线主控的多个进程使用 1 步步骤，则所有使用同一信号量的进程都将读到相同的状态。如果只有一个进程锁定信号量，则该 AHB 总线主控的所有进程都将读到信号量由其自己和 COREID 锁定。

38.3.4 HSEM 写/读/读锁定寄存器地址

对于每个信号量，都会提供两个 AHB 寄存器地址，这两个地址位于两个 0x80 存储区中，彼此分开。

在第一个寄存器地址区中，可通过 HSEM_Rx 寄存器对信号量进行写（锁定/清零）和读操作。

在第二个寄存器地址区中，可通过 HSEM_RLRx 寄存器对信号量进行读（锁定）操作。

38.3.5 HSEM 清零步骤

将信号量清零是一个受保护的过程，目的是防止 AHB 总线主控或不具有信号量锁定权限的过程意外将信号量清零。信号量清零步骤由将相应的 COREID 和 PROCID 以及锁定位 = 0 写入信号量组成。清零后，信号量锁定位、COREID 和 PROCID 均为“0”。

清零后，可能生成一个中断以表示该事件。为此，应使能信号量中断。

清零步骤由一个对信号量 HSEM_Rx 寄存器的写操作组成。

- 将 PROCID 和 COREID 以及锁定位 = 0 写入信号量
- 如果 PROCID 与 COREID 匹配，则信号量释放，并且，如果使能了中断，则可能生成一个中断
- 否则，忽略写操作，信号量保持锁定状态并且不会生成任何中断（信号量被其他 AHB 总线主控或进程锁定）

如果同一 AHB 总线主控的多个进程使用 1 步锁定步骤 (PROCID = 0)，则使用同一信号量的所有进程还将为相应 AHB 总线主控的其他进程清零信号量。

38.3.6 HSEM COREID 信号量清零

AHB 总线主控锁定的所有信号量可使用 HSEM_CR 寄存器一次全部清零。

清零 AHB 总线主控锁定的全部信号量的步骤如下：

- 写入 COREID 和正确的键值。具有匹配 COREID 的所有锁定信号量都将清零（设置为释放），并且可生成中断（使能时）。

该步骤可在 AHB 总线主控非正常工作时使用，在这种情况下，其他 AHB 总线主控可释放锁定的信号量，方法是向 HSEM_CR 寄存器写入非正常工作的 COREID 和正确的键值。这将清零具有匹配 COREID 的所有锁定信号量。

可为释放的信号量生成中断。为此，应在 HSEM_CnIER 寄存器中使能信号量中断。

38.3.7 HSEM 中断

中断线有两条，分别为 hsem_int1_it（用于中断 1）和 hsem_int2_it（用于中断 2），允许每个信号量生成一个中断。

这两条中断线均可提供以下功能：

- 每个信号量的中断使能
- 每个信号量的中断清零
- 每个信号量的中断状态
- 每个信号量的屏蔽中断状态

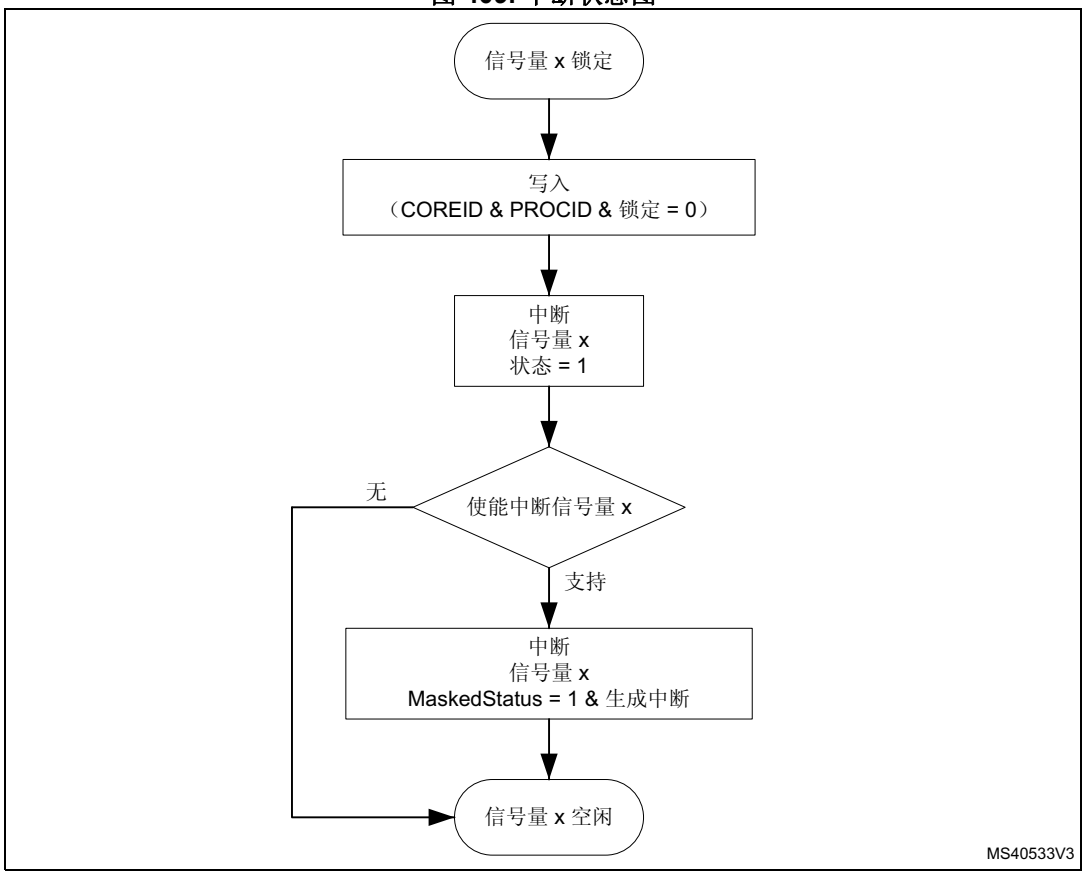
可以通过中断使能 (HSEM_CnIER) 使能信号量影响中断线。已禁止（已屏蔽）信号量中断不会设置该信号量的屏蔽中断状态，也不会在中断线上生成中断。

中断清零 (HSEM_CnICR) 将清零中断线相关信号量的中断状态和屏蔽中断状态。

中断状态 (HSEM_CnISR) 会镜像使能前中断线的信号量中断状态。

屏蔽中断状态 (HSEM_CnMISR) 仅会镜像中断线上已使能信号量中断的信号量中断状态。已使能信号量的所有已屏蔽中断状态均需要清零，才能清零中断线。

图 409. 中断状态图



下文将介绍信号量释放时获得中断的步骤。

尝试锁定信号量 x

- 如果信号量锁定成功，则无需中断。
- 如果信号量锁定失败：
  需在 HSEM_CnICR 中为中断线 hsem_intn_it 清零挂起的信号量 x 的中断状态。
  再次重新尝试锁定信号量 x：
  - 如果信号量锁定成功，则无需中断（在第一次尝试锁定信号量到清零信号量中断状态的过程中，已释放信号量）。
  - 如果信号量锁定失败，需在 HSEM_CnIER 中为中断线 hsem_intn_it 使能信号量 x 的中断。

在出现信号量 x 释放中断后，尝试锁定信号量 x

- 如果信号量锁定成功：
  需在 HSEM_CnIER 中为中断线 hsem_intn_it 禁止信号量 x 的中断，并在 HSEM_CnICR 中为中断线 hsem_intn_it 清零挂起的信号量 x 的中断状态。
- 如果信号量 x 锁定失败：
  需在 HSEM_CnICR 中为中断线 hsem_intn_it 清零挂起的信号量 x 的中断状态。
  再次尝试锁定信号量 x：
  - 如果信号量锁定成功（在第一次尝试锁定信号量到清零信号量中断状态的过程中，已释放信号量）：
    需在 HSEM_CnIER 中为中断线 hsem_intn_it 禁止信号量中断。
  - 如果信号量锁定失败，需等待信号量释放中断。

注： *中断不会锁定信号量。中断后，AHB 总线主控或过程仍必须执行该锁定步骤来锁定信号量。*

可以通过信号量空闲中断通知多个 AHB 总线主控。每个 AHB 总线主控都会获得其中断，第一个作出响应的将锁定信号量。

38.3.8 AHB 总线主控 ID 验证

HSEM 仅允许经授权的 AHB 总线主控 ID 锁定和解锁信号量。

- 对信号量 HSEM_Rx 寄存器进行 AHB 总线主控 2 步锁定写入访问时，将通过有效的总线主控 ID 进行核验。
  - 如果通过未经授权的 AHB 总线主控 ID 进行访问，则相关访问将遭到丢弃，不会锁定信号量。
- 对信号量 HSEM_RLRx 寄存器进行 AHB 总线主控 1 步锁定读取访问时，将通过有效的总线主控 ID 进行核验。
  - 如果对 HSEM_RLRx 进行未经授权的 AHB 总线主控 ID 读取访问，则将返回以下读取数据，具体取决于信号量状态：
    - 当信号量释放时，将返回全“0”。
    - 当信号量之前已锁定时，将返回 HSEM_RLRx 数据。
- 对 HSEM_CR 寄存器进行 COREID 信号量清零写入访问时，将通过有效的总线主控 ID 进行核验。只有有效的总线主控 ID 可写入 HSEM_CR 寄存器，清零任一 COREID 信号量。
  - 如果通过未经授权的 AHB 总线主控 ID 进行访问，则相关访问将遭到丢弃，不会清零 COREID 信号量。

表 222 详细介绍了总线主控/CPU 与 COREID 之间的关系。

表 222. 未经授权的 AHB 总线主控 ID

总线主控 0 (CPU1)	总线主控 1 (CPU2)
COREID = 4	COREID = 8

注： *允许通过未经授权的 AHB 总线主控 ID 对其他寄存器进行访问。*

38.4 HSEM 寄存器

应按字格式访问寄存器。将忽略字节和半字访问，这些访问不会影响信号量。字节和半字读取访问将始终返回 0。字节和半字访问将不会生成总线错误。

38.4.1 HSEM 寄存器信号量 x (HSEM_Rx)

HSEM register semaphore x

偏移地址：0x000 + 0x4 * x（x = 0 到 31）

复位值：0x0000 0000

HSEM_Rx 应用于执行 2 步写入锁定和回读。仅允许通过经授权的 AHB 总线主控 ID 进行写入访问。将丢弃通过未经授权的 AHB 总线主控 ID 进行的写入访问。

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

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

位 31 **LOCK**：锁定指示 (Lock indication)

此位可通过固件写入和读取。

0：写入时释放信号量（仅当 COREID 与 PROCID 匹配时），读取时已释放信号量。

1：写入时尝试锁定信号量，读取时已锁定信号量。

位 30:12 保留，必须保持复位值。

位 11:8 **COREID[3:0]**：信号量 COREID (Semaphore COREID)

通过固件写入，当信号量释放且 LOCK 位写入 1 时，只有在写入信号量的 AHB 总线主控的总线 ID 匹配时，才会写入 COREID。

当信号量清零时（LOCK 位写入 0 且 AHB 总线主控 ID 与 COREID 匹配），COREID 将清零。

当信号量清零时（LOCK 位写入 0 且 AHB 总线主控 ID 与 COREID 不匹配），COREID 将不受影响。

当 LOCK 位已为 1（信号量已锁定）时写入，COREID 将不受影响。

读取操作将返回存储的 COREID 值。

位 7:0 **PROCID[7:0]**：信号量 PROCID (Semaphore PROCID)

通过固件写入，当信号量释放且锁定位写入 1 时，PROCID 将设置为写入的数据。

当信号量清零（LOCK 位写入 0）时，PROCID 将清零。

当 LOCK 位已为 1（信号量已锁定）时写入，PROCID 将不受影响。

读取操作将返回编程的 PROCID 值。

38.4.2 HSEM 读取锁定寄存器信号量 x (HSEM_RLRx)

HSEM read lock register semaphore x

偏移地址：0x080 + 0x004 * x（x = 0 到 31）

复位值：0x0000 0000

访问与 HSEM_Rx 相同的物理位。HSEM_RLRx 应用于执行 1 步读取锁定。仅允许通过经授权的 AHB 总线主控 ID 进行读取访问。将丢弃通过未经授权的 AHB 总线主控 ID 进行的读取访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	COREID[3:0]				PROCID[7:0]							
				r	r	r	r	r	r	r	r	r	r	r	r

位 31 **LOCK**：锁定指示 (Lock indication)

此位仅由固件在该地址处读取。通过有效的总线主控 ID 读取时，将始终返回 1。

当信号量释放且固件执行读取操作时，硬件会将信号量设置为锁定状态。

当信号量锁定且固件执行读取操作时，LOCK 位不受影响。

0：信号量释放。

1：信号量锁定。

位 30:12 保留，必须保持复位值。

位 11:8 **COREID[3:0]**：信号量 COREID (Semaphore COREID)

此位域仅由固件在该地址处读取。

如果在信号量释放时读取，则硬件会将 COREID 设置为读取信号量的 AHB 总线主控 ID。将读取锁定信号量的 AHB 总线主控的 COREID。

如果在信号量锁定时读取，则将返回已锁定信号量的 AHB 总线主控的 COREID。

位 7:0 **PROCID[7:0]**：信号量处理器 ID (Semaphore processor ID)

此位域仅由固件在该地址处读取。

如果在信号量释放时读取，则将返回 0。

如果在信号量锁定时读取，则将返回已锁定信号量的进程的进程 ID。

38.4.3 HSEM 中断使能寄存器 (HSEM_CnIER)（n=1 到 2）

HSEM interrupt enable register

偏移地址：模块 1：0x100

偏移地址：模块 2：0x110

复位值：0x0000 0000

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

位 31:0 **ISE[31:0]**：中断信号量 x 使能位 (Interrupt semaphore x enable bit)

通过固件读取和写入此位。

0：禁止（屏蔽）信号量 x 的中断生成。

1：使能（不屏蔽）信号量 x 的中断生成。

38.4.4 HSEM 中断清零寄存器 (HSEM_CnICR)（n=1 到 2）

HSEM interrupt clear register

偏移地址：模块 1：0x104

偏移地址：模块 2：0x114

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ISC[31:16]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISC[15:0]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:0 **ISC[31:0]**：中断信号量 x 清零位 (Interrupt semaphore x clear bit)

此位通过固件读写，始终读为 0。

0：中断信号量 x 的状态和屏蔽状态不受影响。

1：中断信号量 x 的状态和屏蔽状态清零。

38.4.5 HSEM 中断状态寄存器 (HSEM_CnISR)（n=1 到 2）

HSEM interrupt status register

偏移地址：模块 1：0x108

偏移地址：模块 2：0x118

复位值：0x0000 0000

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

位 31:0 **ISF[31:0]**：使能（屏蔽）前的中断信号量 x 状态位 (Interrupt semaphore x status bit before enable (mask))

此位通过硬件置 1，只能通过固件复位。
此位通过固件写入相应的 HSEM_CnICR 位来清零。
0：中断信号量 x 的状态，中断未挂起。
1：中断信号量 x 的状态，中断挂起。

38.4.6 HSEM 中断状态寄存器 (HSEM_CnMISR)（n=1 到 2）

HSEM interrupt status register

偏移地址：模块 1：0x10C

偏移地址：模块 2：0x11C

复位值：0x0000 0000

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

位 31:0 **MISF[31:0]**：使能（屏蔽）后的屏蔽中断信号量 x 状态位 (Masked interrupt semaphore x status bit after enable (mask))

此位通过硬件置 1，只能通过固件读取。
此位通过固件写入相应的 HSEM_CnICR 位来清零。
0：屏蔽后中断信号量 x 的状态未挂起。
1：屏蔽后中断信号量 x 的状态挂起。

38.4.7 HSEM 清零寄存器 (HSEM_CR)

HSEM clear register

偏移地址：0x140

复位值：0x0000 0000

仅允许通过经授权的 AHB 总线主控 ID 进行写入访问。将丢弃通过未经授权的 AHB 总线主控 ID 进行的写入访问。

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

位 31:16 **KEY[15:0]**：信号量清零键 (Semaphore clear Key)
此位可通过固件写入，将始终读为 0。
键值与 HSEM_KEYR.KEY 不匹配时，信号量不受影响。
键值与 HSEM_KEYR.KEY 匹配时，与 COREID 匹配的所有信号量都将清零为释放状态。

位 15:12 保留，必须保持复位值。

位 11:8 **COREID[3:0]**：要清零的信号量的 COREID (COREID of semaphores to be cleared)
此位域可通过固件写入，始终读为 0。
指示在写入 HSEM_CR 时清零信号量的 COREID。

位 7:0 保留，必须保持复位值。

38.4.8 HSEM 中断清零寄存器 (HSEM_KEYR)

HSEM interrupt clear register

偏移地址：0x144

复位值：0x0000 0000

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

位 31:16 **KEY[15:0]**：信号量清零键 (Semaphore Clear Key)
此位可通过固件写入和读取。
清零信号量时要匹配的键值。

位 15:0 保留，必须保持复位值。

38.4.9 HSEM 寄存器映射

表 223. HSEM 寄存器映射和复位值

偏移	寄存器名称	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
0x000	**HSEM_R0**	LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COREID [3:0]				PROCID[7:0]							
	Reset value	0																				0	0	0	0	0	0	0	0	0	0	0	0
0x004	**HSEM_R1**	LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COREID [3:0]				PROCID[7:0]							
	Reset value	0																				0	0	0	0	0	0	0	0	0	0	0	0
⋮																																	
0x07C	**HSEM_R31**	LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COREID[3:0]				PROCID[7:0]							
	Reset value	0																				0	0	0	0	0	0	0	0	0	0	0	0
0x080	**HSEM_RLR0**	LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COREID [3:0]				PROCID							
	Reset value	0																				0	0	0	0	0	0	0	0	0	0	0	0
0x084	**HSEM_RLR1**	LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COREID [3:0]				PROCID[7:0]							
	Reset value	0																				0	0	0	0	0	0	0	0	0	0	0	0
⋮																																	
0x0FC	**HSEM_RLR31**	LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COREID [3:0]				PROCID[7:0]							
	Reset value	0																				0	0	0	0	0	0	0	0	0	0	0	0
0x100	**HSEM_C1IER**	ISE[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
0x104	**HSEM_C1ICR**	ISC[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
0x108	**HSEM_C1ISR**	ISF[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
0x10C	**HSEM_C1MISR**	MISF[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
0x110	**HSEM_C2IER**	ISE[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
0x114	**HSEM_C2ICR**	ISC[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
0x118	**HSEM_C2ISR**	ISF[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

表 223. HSEM 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x11C	**HSEM_C2MISR**	MISF[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
0x140	**HSEM_CR**	KEY[15:0]																Res.	Res.	Res.	Res.	COREID [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0								
0x144	**HSEM_KEYR**	KEY[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

39 通用串行总线全速设备接口 (USB)

39.1 简介

USB 外设在全速 USB 2.0 总线和 APB1 总线之间实现了一个接口。

此外设支持 USB 挂起/恢复，这样可以停止设备时钟以降低功耗。

39.2 USB 主要特性

- 符合 USB 2.0 全速规范
- 可配置端点数为 1 个到 8 个
- 1024 字节的专用数据包缓冲区存储器 (SRAM)
- 循环冗余校验 (CRC) 生成/校验、非归零反相 (NRZI) 编码/解码以及位填充
- 同步传输支持
- 双缓冲批量/同步端点支持
- USB 挂起/恢复操作
- 帧锁定时钟脉冲生成
- USB 2.0 链路层电源管理支持
- 电池充电规范第 1.2 版支持
- USB 连接/断开功能（USB_DP 线上可控制的内置上拉电阻）

39.3 USB 实现

表 224 列出了器件中的 USB 实现。

表 224. STM32WB55xx USB 实现

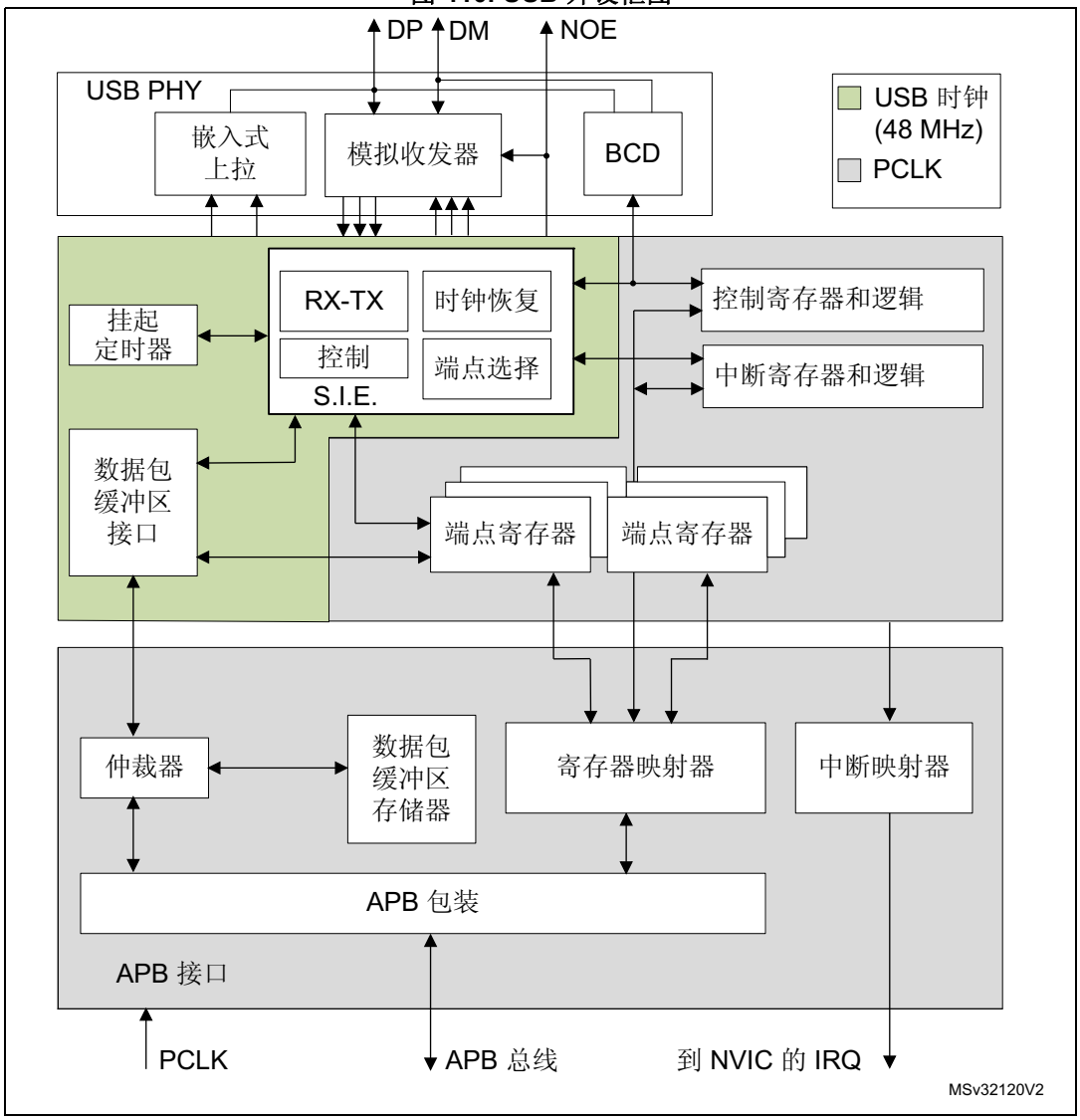
USB 功能(1)	USB
端点数	8
专用数据包缓冲区存储器 SRAM 的大小	1024 字节
专用数据包缓冲区存储器 SRAM 访问模式	2 x 16 位/字
USB 2.0 链路层电源管理 (LPM) 支持	X
电池充电检测 (BCD) 支持	X
USB_DP 线上的内置上拉电阻	X

1. X = 支持

39.4 USB 功能说明

图 410 显示了 USB 外设的框图。

图 410. USB 外设框图



USB 外设在主机 PC 和微控制器实现的功能之间提供符合 USB 规范的连接。主机 PC 和系统存储器之间通过专用数据包缓冲区存储器进行数据传输，此数据包缓冲区存储器通过 USB 外设直接访问。此专用存储器大小为 1024 字节，最多可使用 16 个单向端点或 8 个双向端点。USB 外设与 USB 主机相连，按 USB 标准的要求检测令牌数据包、处理数据发送/接收以及处理握手包。事务格式化通过硬件来执行，其中包括 CRC 生成和校验。

每个端点都与一个缓冲区描述块关联，此描述块指示与端点相关的存储区的位置、大小或必须传送的字节数。当 USB 外设识别到有效功能/端点对的令牌时，将根据需要进行相关的数据传输（如果已配置端点）。USB 外设缓冲的数据装载到内部 16 位寄存器中，并且会对专用缓冲区执行存储器访问。当所有数据传输完成后，必要时，可根据传输方向生成或等待适当的 USB 握手包。

事务结束时将生成端点相关中断，以读取状态寄存器和/或使用不同的中断响应程序。微控制器可确定：

- 必须处理哪个端点
- 出现错误时发生了哪类事务（位填充、格式化、CRC、协议、ACK 丢失和上溢/下溢等）

为同步传输和高吞吐量批量传输提供特殊支持，实现双缓冲区用法，从而确保当微控制器使用其中一个缓冲区时，始终有另一个缓冲区可供 USB 外设使用。

必要时，可通过写入控制寄存器将单元置于低功耗模式（挂起模式）。此时，可避免产生所有静态功耗，USB 时钟会减速或停止。如果在低功耗模式下检测到 USB 输入上存在活动，则会异步唤醒设备。可以将特殊中断源直接连接到唤醒线，从而使系统立即重启正常的时钟生成操作和/或支持直接时钟启动/停止。

39.4.1 USB 模块说明

USB 外设实现了与 USB 连接相关的所有功能，具体包括以下模块：

- USB 物理接口 (USB PHY)：此模块维持与外部 USB 主机间的电气连接。它包含差分模拟收发器本身、可控制的内置上拉电阻（连接到 USB_DP 线）以及电池充电检测 (BCD) 支持，在相同 USB_DP 和 USB_DM 线上复用。模拟收发器的输出使能控制信号（低电平有效）通过 USB_NOE 从外部提供。它可用于驱动某活动 LED 或提供有关其他电路实际通信方向的信息。
- 串行接口引擎 (SIE)：此模块的功能包括同步模式识别、位填充、CRC 生成和校验、PID 验证/生成以及握手评估。它必须与 USB 收发器连接，并使用数据包缓冲区接口提供的虚拟缓冲区来存储本地数据。此单元还会根据 USB 外设事件（例如，帧起始 (SOF)、USB_Reset 和数据错误）和端点相关事件（例如，数据包发送结束或接收成功）生成相应信号，这些信号随后用于生成中断。
- 定时器：此模块生成帧起始锁定时钟脉冲并在已有 3 ms 未接收到通信数据时检测全局挂起（来自主机）。
- 数据包缓冲区接口：此模块管理本地存储器，能够以灵活的方式实现一组用于发送和接收的缓冲区。它可根据来自 SIE 的请求选择合适的缓冲区，并将这些请求置于端点寄存器指向的存储器地址。在数据包交换过程中，每交换一个字节，地址便会递增一次，这样便可跟踪已交换的字节数，避免超出缓冲区的最大容量。
- 端点相关寄存器：每个端点有一个相关的寄存器，其中包含端点类型及其当前状态。对于单向/单缓冲区端点，可使用单个寄存器实现两个不同的端点。这类寄存器有 8 个，最多支持 16 个单向/单缓冲区端点或 7 个双缓冲区端点，并且支持这两种端点的任意组合。例如，可以将 USB 外设编程为具有 4 个双缓冲区端点和 8 个单缓冲区/单向端点。
- 控制寄存器：这类寄存器包含与整个 USB 外设的状态相关的信息，用于强制执行某些 USB 事件（例如，恢复和掉电）。
- 中断寄存器：这类寄存器包含中断屏蔽和事件记录。它们可用于查询中断原因、中断状态或清除挂起中断的状态。

注： ** 端点 0 始终用于单缓冲区模式下控制传输。*

USB 外设通过 APB1 接口连接到 APB1 总线，此接口包含以下模块：

- 数据包存储器：此存储器是在物理上包含数据包缓冲区的本地存储器。可通过数据包缓冲区接口使用此存储器，以创建数据结构并且可通过应用软件直接访问。数据包存储器的大小为 1024 字节，以 512 个 16 位半字的结构存在。
- 仲裁器：此模块接受来自 APB1 总线和 USB 接口的存储器请求。它通过为 APB1 访问分配优先级来解决冲突，同时始终保留一半存储器带宽来完成所有 USB 传输。该时分双工方案实现了虚拟双端口 SRAM，允许在发生 USB 事务时进行存储器访问。此方案还支持任意长度的多字 APB1 传输。
- 寄存器映射器：此模块收集 USB 外设不同字节宽和位宽的寄存器，这些寄存器采用 16 位宽半字组的形式，通过 APB1 寻址。
- APB1 封装器：此模块为存储器和寄存器提供连接 APB1 的接口。它还将整个 USB 外设映射到 APB1 地址空间。
- 中断映射器：此模块用于选择可能的 USB 事件如何生成中断以及将中断映射到 NVIC。

39.5 编程注意事项

以下各节将介绍 USB 外设与应用软件之间的预期交互，帮助简化应用软件开发。

39.5.1 通用 USB 设备编程

本部分将介绍应用软件需要执行哪些主要任务才能获得符合 USB 规范的行为。与大多数常规 USB 事件相关的操作均在考虑范围之内，并且有相关段落专门说明双缓冲端点和同步传输这两个特例。除系统复位外，操作始终在下述其中一个 USB 事件的驱动下由 USB 外设发起。

39.5.2 系统复位和上电复位

系统复位和上电复位后，应用软件应执行的第一个操作是为 USB 外设提供所需的全部时钟信号，随后使其复位信号无效，以便能够访问其寄存器。下文将介绍整个初始化序列。

首先，应用软件需要使用设备时钟管理逻辑所提供的相关控制位激活寄存器宏单元时钟并使宏单元特定的复位信号无效。

随后，必须使用 CNTR 寄存器的 PDWN 位接通设备中与 USB 收发器相关的模拟部分，这一步需要特殊处理。此位旨在接通为端口收发器供电的内部参考电压。此电路有一段定义的启动时间（数据手册中指定的 $t_{STARTUP}$），这段时间内未定义 USB 收发器的行为。因此，在将 CNTR 寄存器中的 PDWN 位置 1 后，必须等待这段时间结束时才能消除 USB 部分的复位条件（通过将 CNTR 寄存器中的 FRES 位清零）。清除 ISTR 寄存器会在任何其他宏单元操作使能之前，消除所有意外的挂起中断。

系统复位时，微控制器必须初始化所需的全部寄存器和数据包缓冲区描述表，以便 USB 外设能够正常地产生中断和进行数据传输。对于非特定于任何端点的所有寄存器，必须按照应用软件的需求（使能中断的选择，所选的数据包缓冲区地址等）进行初始化。对于 USB 复位情况，过程将继续进行（请参见下一段）。

USB 复位（RESET 中断）

发生该事件时，USB 外设的状态与系统复位后的初始化状态（如上一段所述）相同：所有端点寄存器中均禁止通信（USB 外设不会响应任何数据包）。作为对 USB 复位事件的响应，必须使能 USB 功能（USB 地址为 0），仅实现默认控制端点（端点地址也为 0）。具体实现方法是：将 USB_DADDR 寄存器的使能功能 (EF) 位置 1 并相应地初始化 EP0R 寄存器及其相关的数据包缓冲区。在 USB 枚举过程中，主机会为此设备分配一个唯一地址（该地址必须写入 USB_DADDR 寄存器的 ADD[6:0] 位）并配置任何其他必要的端点。

当接收到 RESET 中断时，应用软件负责在触发中断的复位序列结束后的 10 ms 内再次使能 USB 功能 0 的默认端点。

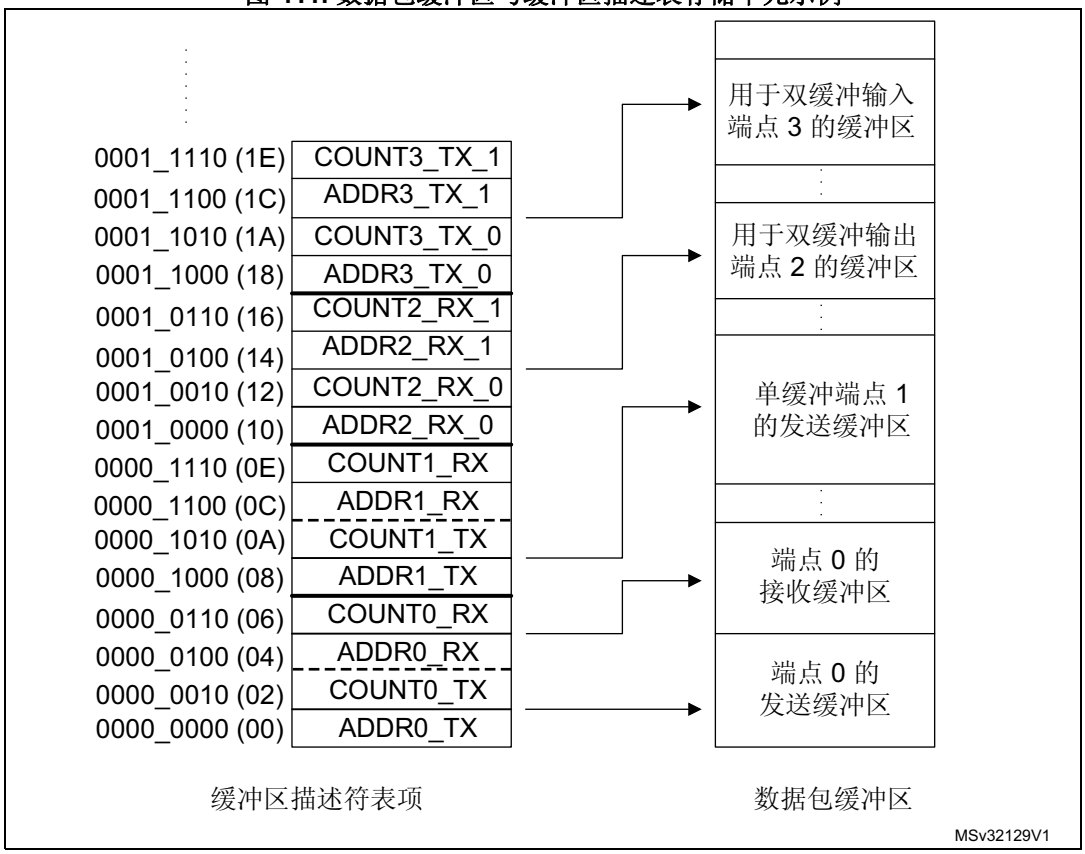
数据包缓冲区的结构和用法

每个双向端点均可从主机接收数据或向主机发送数据。端点接收的数据存储在为该端点保留的专用存储器缓冲区内，而另一个存储器缓冲区则包含该端点要发送的数据。对该存储器的访问通过数据包缓冲区接口模块来执行，该模块会发出存储器访问请求并等待应答。由于数据包缓冲区存储器还必须供微处理器访问，因此需通过仲裁逻辑来处理访问冲突：半个 APB1 周期供微控制器访问，剩下的半个周期供 USB 外设访问。通过这种方式，微控制器和 USB 外设可将数据包存储器视为双端口 SRAM 来执行操作，而无需考虑任何冲突，即使微控制器正在执行背靠背访问也如此。USB 外设逻辑使用专用时钟。此专用时钟的频率根据 USB 标准的要求固定为 48 MHz，这可能与用于 APB1 总线接口的时钟不同。可采用不同的时钟配置，即 APB1 时钟频率可高于或低于 USB 外设的时钟频率。

注： *根据 USB 数据速率和数据包存储器接口要求，必须确保 APB1 时钟的频率不低于 10 MHz，以避免发生数据上溢/下溢问题。*

每个端点与两个数据包缓冲区关联（通常，一个用于发送，另一个用于接收）。缓冲区可置于数据包存储器内的任何位置，因为其位置和大小在缓冲区描述表中指定。此描述表也位于数据包存储器中，地址由 USB_BTABLE 寄存器指定。每个表条目与一个端点寄存器关联并且由四个 16 位半字组成，因此表起始地址必须始终与 8 字节边界对齐（USB_BTABLE 寄存器的低三位始终为“000”）。*第 39.6.2 节：缓冲区描述符表*介绍了缓冲区描述符表条目。对于一个既非同步端点也非双缓冲批量端点的单向端点，只需要一个数据包缓冲区（与所支持的传输方向相关的数据包缓冲区）。与不支持的传输方向或未使用端点相关的其他表存储单元可供用户使用。同步端点和双缓冲批量端点的数据包缓冲区处理方式比较特殊（请分别参见*第 39.5.4 节：同步传输*和*第 39.5.3 节：双缓冲端点*）。*图 411* 给出了缓冲区描述表条目和数据包缓冲区之间的关系。

图 411. 数据包缓冲区与缓冲区描述表存储单元示例



在接收或发送期间，每个数据包缓冲区从底部开始使用。USB 外设始终不会更改与所分配存储器缓冲区相邻的存储单元的内容；如果接收的数据包长度大于所分配缓冲区的长度（缓冲区上溢条件），则到达最后一个可用的存储单元后便停止将数据复制到存储器。

端点初始化

初始化端点的第一步是向 ADDRn_TX/ADDRn_RX 寄存器写入相应值，以使 USB 外设获悉要发送的数据已准备就绪，要接收的数据可进行缓冲。必须根据端点类型设置 USB_EPnR 寄存器中的 EP_TYPE 位，最终使用 EP_KIND 位使能所需的任何特殊功能。在发送端，必须通过 USB_EPnR 寄存器中的 STAT_TX 位使能端点，并且必须初始化 COUNTn_TX。接收时，必须将 STAT_RX 位置 1 以使能接收，并且必须通过 BL_SIZE 和 NUM_BLOCK 位域向 COUNTn_RX 写入所分配的缓冲区大小。单向端点（同步端点和双缓冲批量端点除外）只需初始化与所支持方向相关的位和寄存器。使能发送与/或接收后，不得通过应用软件修改寄存器 USB_EPnR 以及存储单元 ADDRn_TX/ADDRn_RX 和 COUNTn_TX/COUNTn_RX（分别地），因为硬件会实时更改其值。数据传输操作完成时（通过 CTR 中断事件通知），可再次访问这些存储单元以重新使能新操作。

IN 数据包（数据传送）

接收 IN 令牌数据包时，如果所接收的地址与已配置的有效端点匹配，则 USB 外设会访问与所寻址端点相关的缓冲区描述符表条目内的 ADDRn_TX 和 COUNTn_TX 存储单元的内容。这些存储单元的内容存储在其内部 16 位寄存器 ADDR 和 COUNT 中（无法通过软件访问）。数据包存储器会被再次访问以读取要发送的第一个字节（请参见*第 1249 页的数据包缓冲区的结构和用法*），然后会根据 USB_EPnR 位 DTOG_TX 开始发送 DATA0 或 DATA1 PID。当 PID 发送完成时，从缓冲区存储器读取的第一个字节将被装载到输出移位寄存器中以通过 USB 总线发送出去。发送完最后一个数据字节后，将发送计算得出的 CRC。如果寻址的端点无效，将根据 USB_EPnR 寄存器中的 STAT_TX 位发送 NAK 或 STALL 握手包而非数据包。

ADDR 内部寄存器用作当前缓冲区存储单元的指针，而 COUNT 用于对要发送的剩余字节计数。从数据包缓冲区存储器读取的每个半字均通过 USB 总线从最低有效字节开始发送。在发送缓冲区存储器中从 ADDRn_TX 指向的地址开始读取 COUNTn_TX/2 个半字。如果发送的数据包由奇数个字节组成，则仅会使用所访问的最后一个半字的低半部分。

接收到主机的 ACK 后，USB_EPnR 寄存器将通过以下方式更新：DTOG_TX 位翻转、通过设置 STAT_TX=10 (NAK) 将端点置为无效、将 CTR_TX 位置 1。应用软件必须首先识别端点，即请求微控制器关注以检查 USB_ISTR 寄存器中的 EP_ID 和 DIR 位。处理 CTR_TX 事件会开始清零中断位；应用软件随后会准备另一个装满数据的发送缓冲区，用下一次传输期间要发送的字节数更新 COUNTn_TX 表存储单元，最后将 STAT_TX 设置为“11”(VALID) 以重新使能发送。当 STAT_TX 位等于“10”(NAK) 时，寻址到相应端点的任何 IN 请求都得不到应答，并且指示流控制状态：USB 主机将重试事务，直到成功。必须按上述顺序执行操作序列，以避免与触发 CTR 中断的 IN 事务寻址到同一端点且紧随其后的第二个 IN 事务的通知丢失。

OUT 和 SETUP 数据包（数据接收）

USB 外设对这两种令牌的处理方式大致相同，SETUP 数据包不同于 OUT 数据包的处理将在下文有关控制传输的段落中详细说明。接收 OUT/SETUP PID 时，如果地址与有效端点匹配，则 USB 外设会访问与所寻址端点相关的缓冲区描述符表条目内的 ADDRn_RX 和 COUNTn_RX 存储单元的内容。ADDRn_RX 的内容直接存储在其内部寄存器 ADDR 中。而此时 COUNT 复位，BL_SIZE 和 NUM_BLOCK 位域的值（从 COUNTn_RX 内容中读取）用于初始化内部 16 位计数器 BUF_COUNT，该计数器用于检查缓冲区上溢条件（上述所有内部寄存器均无法通过软件访问）。USB 外设后续接收的数据字节均以半字为单位打包（接收的第一个字节作为最低有效字节存储），随后传输到从内部 ADDR 寄存器包含的地址开始的数据包缓冲区，每次传输字节时 BUF_COUNT 递减、COUNT 递增。当检测到 DATA 数据包结束时，将检验所接收的 CRC 是否正确，只有当接收期间没有错误时，才会将 ACK 握手包发送回发送主机。

当出现 CRC 错误或其他类型的错误（位填充超限、帧错误等）时，数据字节仍将在数据包存储器缓冲区中进行复制，至少到错误检测点为止，但不会发送 ACK 数据包且 USB_ISTR 寄存器中的 ERR 位置 1。不过，这种情况下通常不需要软件操作：USB 外设会从接收错误中恢复，并保持就绪状态等待下一个事务的到来。如果寻址的端点无效，将根据 USB_EPnR 寄存器中的 STAT_RX 位发送 NAK 或 STALL 握手包而非 ACK，并且不会向接收存储器缓冲区中写入任何数据。

接收存储器缓冲区存储单元从 ADDRn_RX 中包含的地址开始写入，写入的字节数与接收的数据包长度（包括 CRC，即数据有效负载长度 + 2）相对应，或者一直写到分配的最后一个存储单元（由 BL_SIZE 和 NUM_BLOCK 定义），以先结束者为准。凭借这种方法，USB 外设写入的数据始终不会超出所分配接收存储器缓冲区的容量。如果数据包有效负载的长度（应用软件实际使用的字节数）大于所分配的缓冲区，则 USB 外设会检测到缓冲区上溢条件。在这种情况下，将发送 STALL 握手而非常规的 ACK 来通知主机出现问题，此时不会产生中断，事务将被视作失败。

当事务正确完成后（通过发送 ACK 握手包），内部 COUNT 寄存器将复制回缓冲区描述表条目内的 COUNTn_RX 存储单元。BL_SIZE 和 NUM_BLOCK 位域不受影响，这两个位域通常无需重写，USB_EPnR 寄存器通过以下方式更新：DTOG_RX 位翻转、通过设置 STAT_RX = “10” (NAK) 将端点置为无效，将 CTR_RX 位置 1。如果事务因错误或缓冲区上溢条件而失败，则上面列出的所有操作均不会发生。应用软件必须首先识别端点，即请求微控制器关注以检查 USB_ISTR 寄存器中的 EP_ID 和 DIR 位。首先确定事务类型（USB_EPnR 寄存器中的 SETUP 位）以处理 CTR_RX 事件；应用软件必须清零中断标志位，并通过读取与正在处理的端点相关的缓冲区描述表条目内的 COUNTn_RX 存储单元获得所接收字节的数量。处理完接收的数据后，应用软件应将 USB_EPnR 中的 STAT_RX 位设置为 “11” （有效），从而允许其他事务。当 STAT_RX 位等于 “10” (NAK) 时，寻址到相应端点的任何 OUT 请求都得不到应答，并且指示流控制状态：USB 主机将重试事务，直到成功。必须按上述顺序执行操作序列，以避免丢失与触发 CTR 中断的 OUT 事务寻址到同一端点且紧随其后的第二个 OUT 事务的通知。

控制传输

控制传输包含一个 SETUP 事务，其后跟有零个或多个数据阶段（全部数据阶段均采用相同的方向），再后面是一个状态阶段（一个零字节反向传输）。SETUP 事务仅由控制端点来处理，并且与 OUT 事务十分相似（数据接收），不同之处在于所寻址端点寄存器的 DTOG_TX 和 DTOG_RX 位的值分别设置为 1 和 0 以初始化控制传输，STAT_TX 和 STAT_RX 均设置为 “10” (NAK) 以允许软件决定后续事务必须为 IN 还是 OUT，具体取决于 SETUP 的内容。控制端点必须在每个 CTR_RX 事件时检查 USB_EPnR 寄存器中的 SETUP 位，以区分正常的 OUT 事务与 SETUP 事务。USB 设备可通过解析 SETUP 阶段传输的数据来确定数据阶段的数量和方向，并用于在出现错误时停止事务。为此，在最后一个数据阶段前的所有数据阶段中，应将未使用的方向设置为 STALL，以便在主机过于迅速地改变传输方向时获取 STALL 作为状态阶段。

使能最后一个数据阶段时，应将相反方向设置为 NAK，以便在主机立即改变传输方向（以执行状态阶段）时一直等待控制操作完成。如果控制操作成功完成，软件会将 NAK 更改为 VALID，否则将更改为 STALL。同时，如果状态阶段为 OUT，则应将 STATUS_OUT（USB_EPnR 寄存器中的 EP_KIND）位置 1，以便在执行数据非零的状态事务时生成错误。当处理状态事务时，应用软件会将 STATUS_OUT 位清零，并将 STAT_RX 设置为 VALID（以接受新命令）、将 STAT_TX 设置为 NAK（以在下一次 SETUP 后立即延迟一段可能的状态阶段）。

由于 USB 规范规定 SETUP 数据包不可通过 ACK 以外的握手包来应答（这最终将中止之前发出的命令而启动新命令），USB 逻辑不允许控制端点通过 NAK 或 STALL 握手包来应答从主机接收的 SETUP 令牌。

当 STAT_RX 位设置为 “01” (STALL) 或 “10” (NAK) 且接收到 SETUP 令牌时，USB 将接受数据（以执行所需的数据传输）并发回 ACK 握手包。如果该端点之前发出的 CTR_RX 请求尚未得到应用软件的应答（即 CTR_RX 位仍因之前完成的接收操作而置 1），USB 将丢弃 SETUP 事务且不会通过任何握手包进行应答（无论其状态如何均如此），从而模拟接收错误并强制主机再次发送 SETUP 令牌。这样做的目的在于避免与触发 CTR_RX 中断的 IN 事务寻址到同一端点且紧随其后的第二个 IN 事务的通知丢失。

39.5.3 双缓冲端点

USB 标准定义的所有不同端点类型均代表不同的通信模型，并且说明了各类数据传输操作的典型要求。当主机 PC 和 USB 功能之间要传输大量数据时，批量端点类型是最适合的模型。这是因为主机会调度批量事务以填充帧中的所有可用带宽，因此只要 USB 功能就绪处理寻址到它的批量事务，便可最大程度地提高实际传输速率。如果在下一个事务到达时，USB 功能仍忙于上一个事务，则它将通过 NAK 握手进行应答。主机 PC 将再次发出同一事务，直到 USB 功能准备好处理该事务。由于重复发送占用带宽，因此实际传输速率会降低。为此，可为批量端点搭配使用称为“双缓冲”的专用功能。

激活“双缓冲”功能时，数据翻转序列用于选择 USB 外设将使用哪个缓冲区来执行所需的数据传输，使用“发送”和“接收”数据包存储区在每个事务成功完成时管理缓冲区交换，以便在 USB 外设填充一个缓冲区时，始终有另一个完整的缓冲区可供应用软件使用。例如，在指向“接收”双缓冲批量端点的 OUT 事务期间，当一个缓冲区用来自 USB 主机的新数据填充时，另一个缓冲区可供微控制器软件使用（同样适用于“发送”双缓冲批量端点和 IN 事务）。

由于交换缓冲区管理需要使用全部 4 个缓冲区描述表存储单元来管理地址指针和已分配存储区的长度，因此用于实现双缓冲批量端点的 USB_EPnR 寄存器被强制用作单向寄存器。因此，只有一个 STAT 位对必须设置为“00”（禁止）以外的值：将双缓冲批量端点使能用于接收时为 STAT_RX；将双缓冲批量端点使能用于发送时为 STAT_TX。如果需要将双缓冲批量端点使能同时用于发送和接收，则必须使用两个 USB_EPnR 寄存器。

要利用双缓冲功能达到最高传输速率，必须修改前几章中介绍的端点流控制结构，以便只在 USB 外设和应用软件之间发生缓冲区冲突时（而非每次事务成功结束时），将端点状态切换为 NAK。USB 外设当前使用的存储器缓冲区由与端点方向相关的 DTOG 位定义：DTOG_RX（USB_EPnR 寄存器的位 14）用于“接收”双缓冲批量端点；DTOG_TX（USB_EPnR 寄存器的位 6）用于“发送”双缓冲批量端点。为实现新的流控制方案，USB 外设应知道应用软件当前使用哪个数据包缓冲区，以便获悉是否存在任何冲突。由于 USB_EPnR 寄存器中有两个 DTOG 位，但只有一个位可供 USB 外设用于数据和缓冲区排序（由于双缓冲功能所要求的单向限制），因此应用软件可使用另一个位显示当前使用哪个缓冲区。这种新的缓冲区标志称为 SW_BUF。下表给出了对于“发送”和“接收”双缓冲批量端点，USB_EPnR 寄存器位与 DTOG/SW_BUF 定义之间的对应关系。

表 225. 双缓冲缓冲区标志定义

缓冲区标志	“发送”端点	“接收”端点
DTOG	DTOG_TX（USB_EPnR 位 6）	DTOG_RX（USB_EPnR 位 14）
SW_BUF	USB_EPnR 位 14	USB_EPnR 位 6

USB 外设当前使用的存储器缓冲区由 DTOG 缓冲区标志定义，而应用软件当前使用的缓冲区由 SW_BUF 缓冲区标志定义。对于“发送”和“接收”这两种情况，缓冲区标志值与使用的数据包缓冲区之间的关系是相同的，如下表所列。

表 226. 批量双缓冲存储器缓冲区用法

端点类型	DTOG	SW_BUF	USB 外设使用的数据包缓冲区	应用软件使用的数据包缓冲区
IN	0	1	ADDRn_TX_0 / COUNTn_TX_0 缓冲区描述表存储单元。	ADDRn_TX_1 / COUNTn_TX_1 缓冲区描述表存储单元。
	1	0	ADDRn_TX_1 / COUNTn_TX_1 缓冲区描述表存储单元	ADDRn_TX_0 / COUNTn_TX_0 缓冲区描述表存储单元。
	0	0	无 (1)	ADDRn_TX_0 / COUNTn_TX_0 缓冲区描述表存储单元。
	1	1	无(1)	ADDRn_TX_0 / COUNTn_TX_0 缓冲区描述表存储单元。
OUT	0	1	ADDRn_RX_0 / COUNTn_RX_0 缓冲区描述表存储单元。	ADDRn_RX_1 / COUNTn_RX_1 缓冲区描述表存储单元。
	1	0	ADDRn_RX_1 / COUNTn_RX_1 缓冲区描述表存储单元。	ADDRn_RX_0 / COUNTn_RX_0 缓冲区描述表存储单元。
	0	0	无(1)	ADDRn_RX_0 / COUNTn_RX_0 缓冲区描述表存储单元。
	1	1	无(1)	ADDRn_RX_1 / COUNTn_RX_1 缓冲区描述表存储单元。

1. 端点处于 NAK 状态。

批量端点的双缓冲功能通过以下方式激活：

- USB_EPnR 寄存器的 EP_TYPE 位域中写入“00”，以将端点定义为批量
- 将同一寄存器中的 EP_KIND 位设置为“1”(DBL_BUF)

应用软件负责根据使用的第一个缓冲区初始化 DTOG 和 SW_BUF 位；考虑到这两位具有特殊的仅-位翻转属性，必须执行该操作。如果第一个事务在 DBL_BUF 置 1 后结束，则将触发双缓冲批量端点的特殊流控制，该特殊流控制用于寻址到该端点的所有其他事务，直到 DBL_BUF 保持置 1。每个事务结束时，所寻址端点 USB_EPnR 寄存器的 CTR_RX 或 CTR_TX 位将置 1，具体取决于使能的方向。同时，USB_EPnR 寄存器中受影响的 DTOG 位由硬件位翻转，从而使 USB 外设缓冲区在完全不依赖软件的情况下实现交换。与普通事务以及 DBL_BUF 置 1 后的第一个事务不同，STAT 位对不受事务终止的影响，其值保持为“11”（有效）。不过，当接收到新事务的令牌数据包时，如果检测到 USB 外设和应用软件之间存在缓冲区冲突（此时，DTOG 和 SW_BUF 的值相同，请参见 *第 1254 页的表 226*），则实际的端点状态将被屏蔽为“10”(NAK)。应用软件通过清零中断标志并开始对完成的事务进行任何所需处理来响应 CTR 事件通知。当应用软件数据包缓冲区使用完毕后，软件将翻转 SW_BUF 位，向该位写入“1”以通知 USB 外设该缓冲区当前可用。通过这种方式，不应答事务的数量仅受应用软件准备事务数据的时间限制：如果准备时间短于在 USB 总线上完成一个事务所需的时间，则将不会发生因流控制引起的重新发送，实际传输速率将仅受主机 PC 的限制。

应用软件可始终覆盖为双缓冲批量端点实现的特殊流控制，向相关 USB_EPnR 寄存器的 STAT 位对写入“11”（有效）以外的明确状态。在这种情况下，USB 外设将始终使用编程的端点状态，而与缓冲区使用情况无关。

39.5.4 同步传输

USB 标准支持需要固定且精确的数据生产/消费频率的全速外设，这种通信定义为“同步”。该数据的典型示例包括：音频采样、压缩视频流以及通常对传输频率精度有严苛要求的任何一种采样数据。当端点在枚举阶段定义为“同步”时，主机将在帧中分配所需带宽并在每个帧中严密地传输一个 IN 或 OUT 帧，具体取决于端点方向。为限制带宽需求，同步通信中不允许重新发送失败的事务，这将导致以下情况：同步事务没有握手阶段，数据包之后不等待或发送任何 ACK 包。出于同样的原因，同步传输不支持数据位翻转，始终使用 DATA0 PID 来启动任何数据包。

端点的同步行为通过将其 USB_EPnR 寄存器中的 EP_TYPE 位设置为“10”来选择；由于没有握手阶段，STAT_RX/STAT_TX 位对的合法值只有“00”（禁止）和“11”（有效），任何其他值都将产生不符合 USB 标准的结果。同步端点通过实现双缓冲来简化应用软件的开发，使用“发送”和“接收”数据包存储区在每个事务成功时管理缓冲区交换，以便在 USB 外设填充一个缓冲区时，始终有另一个完整的缓冲区可供应用软件使用。

根据*表 227*，USB 外设当前使用的存储器缓冲区由与端点方向相关的 DTOG 位定义（DTOG_RX 用于“接收”同步端点，DTOG_TX 用于“发送”同步端点，二者均位于相关的 USB_EPnR 寄存器中）。

表 227. 同步存储器缓冲区的用法

端点类型	DTOG 位值	应用软件使用的 数据包缓冲区	应用软件使用的 数据包缓冲区
IN	0	ADDRn_TX_0 / COUNTn_TX_0 缓冲区描述表存储单元。	ADDRn_TX_1 / COUNTn_TX_1 缓冲区描述表存储单元。
	1	ADDRn_TX_1 / COUNTn_TX_1 缓冲区描述表存储单元。	ADDRn_TX_0 / COUNTn_TX_0 缓冲区描述表存储单元。
OUT	0	ADDRn_RX_0 / COUNTn_RX_0 缓冲区描述表存储单元。	ADDRn_RX_1 / COUNTn_RX_1 缓冲区描述表存储单元。
	1	ADDRn_RX_1 / COUNTn_RX_1 缓冲区描述表存储单元。	ADDRn_RX_0 / COUNTn_RX_0 缓冲区描述表存储单元。

与双缓冲批量端点一样，用于实现同步端点的 USB_EPnR 寄存器被强制用作单向寄存器。如果需要将同步端点使能同时用于发送和接收，则必须使用两个 USB_EPnR 寄存器。

应用软件负责根据使用的第一个缓冲区初始化 DTOG 位；考虑到这两位具有特殊的仅-位翻转属性，必须执行该操作。每个事务结束时，所寻址端点 USB_EPnR 寄存器的 CTR_RX 或 CTR_TX 位将置 1，具体取决于使能的方向。同时，USB_EPnR 寄存器中受影响的 DTOG 位由硬件同步，从而使缓冲区在完全不依赖软件的情况下实现交换。STAT 位对不受事务完成的影响，因为同步传输缺少握手阶段，没有可用的流控制，端点始终保持“11”（有效）。同步 OUT 传输期间出现的 CRC 错误或缓冲区上溢条件仍被视为正确事务，它们始终会触发 CTR_RX 事件。但是，CRC 错误仍会将 USB_ISTR 寄存器中的 ERR 位置 1 以向软件通知可能的数据损坏。

39.5.5 挂起/恢复事件

USB 标准定义了一个称为 SUSPEND 的特殊外设状态，在此状态下，USB 总线消耗的平均电流不得超过 2.5 mA。这是对总线供电设备的基本要求，而自供电设备无需遵循这一严苛的功耗限制。在挂起模式下，主机 PC 不在 USB 总线上发送任何通信数据并持续 3 ms 以上的时间，以此发送通知：因为在正常操作期间必须每隔 1 ms 发送一次 SOF 数据包，如果 USB 外设检测到 3 个连续的 SOF 数据包缺失，则会将这种情况视为主机 PC 的挂起请求，然后将 USB_ISTR 寄存器中的 SUSP 位设置为“1”，从而引起中断（如果使能）。设备挂起后，可通过所谓的 RESUME 序列恢复其正常操作，该序列可从主机 PC 或直接从外设本身启动，但始终通过主机 PC 终止。挂起的 USB 外设必须仍能够检测到 RESET 序列，将此事件视为正常的 USB 复位事件并对其做出响应。

挂起 USB 外设所使用的实际步骤取决于设备，因为对于不同的设备构成，降低总功耗所需的操作也会有所不同。

下面给出了典型挂起步骤的简单说明，重点关注与 USB 外设的 SUSP 通知对应的应用软件程序中与 USB 相关的方面：

1. 将 USB_CNTR 寄存器中的 FSUSP 位置 1。此操作将激活 USB 外设中的挂起模式。激活挂起模式后，将立即禁止“接收到 SOF 时检查”功能，以避免在 USB 挂起时发出任何其他 SUSP 中断。
2. 消除或降低 USB 外设以外的模块中的任何静态功耗。
3. 将 USB_CNTR 寄存器中的 LP_MODE 位置 1，以消除模拟 USB 收发器中的静态功耗，但使这些收发器能够检测恢复活动。
4. 可选择关闭外部振荡器或器件 PLL 以停止设备内的任何活动。

如果在设备处于 SUSPEND 模式时发生 USB 事件，则必须调用 RESUME 程序来恢复标称时钟和正常的 USB 行为。当唤醒事件是 USB 复位序列时，必须特别注意确保此过程不超过 10 ms（更多详细信息，请参见“通用串行总线规范”）。在 USB 外设挂起时启动恢复或复位序列会将 USB_CNTR 寄存器中的 LP_MODE 位异步清零。即使此事件能够触发 WKUP 中断（如果使能），也必须仔细评估是否使用中断响应程序，因为系统时钟重启会引入较长的延时；为缩短重新激活标称时钟前的延时，建议紧接着挂起程序末尾放置恢复程序，以便在系统时钟重启时，立即执行相应代码。为防止 ESD 放电或任何其他类型的噪声唤醒系统（退出挂起模式是一个异步事件），挂起期间需在数据线上激活合适的模拟滤波器；滤波器的宽度约为 70 ns。

以下是恢复程序应解决的一系列操作：

1. 可选择接通外部振荡器和/或器件 PLL。
2. 将 USB_CNTR 寄存器中的 FSUSP 位清零。
3. 如果必须识别恢复触发事件，则可根据*表 228* 使用 USB_FNR 寄存器中的位 RXDP 和 RXDM，表中还列出了所有情况下的目标软件操作。如果需要，可通过监视上述位的状态（即检查它们何时达到“10”配置，该配置表示空闲总线状态）来检测恢复或复位序列是否结束；此外，当复位序列结束时，USB_ISTR 寄存器中的 RESET 位将置 1，从而发出中断（如果使能），该中断应照常处理。

表 228. 恢复事件检测

[RXDP,RXDM] 状态	唤醒事件	所需的软件恢复操作
“00”	根复位	无
“10”	无（总线上的噪声）	返回挂起模式
“01”	根恢复	无
“11”	不允许（总线上的噪声）	返回挂起模式

设备可能需要退出挂起模式来响应与 USB 协议不是直接相关的特定事件（例如，通过移动鼠标唤醒整个系统）。在这种情况下，可通过如下方式启动恢复序列：将 USB_CNTR 寄存器中的 RESUME 位置 1，然后在一段介于 1 ms 和 15 ms 之间的时间间隔后将其复位为 0（此时间间隔可使用 ESOF 中断来定时，当系统时钟以标称频率运行时，周期为 1 ms）。RESUME 位清零后，恢复序列将由主机 PC 完成，其结束可再次使用 USB_FNR 寄存器中的 RXDP 和 RXDM 位来监视。

注： *RESUME 位仍必须仅在 USB 外设置于挂起模式后（即 USB_CNTR 寄存器中的 FSUSP 位置 1）使用。*

39.6 USB 和 USB SRAM 寄存器

USB 外设寄存器可分为以下几组：

- 通用寄存器：中断和控制寄存器
- 端点寄存器：端点配置和状态

USB SRAM 寄存器包括：

- 缓冲区描述符表：数据缓冲区所在的数据包存储器的存储单元（有关 USB SRAM 基址，请参见*第 2.2 节：存储器构成*）。

所有寄存器地址均以相对于 USB 外设寄存器基址的偏移量表示，但缓冲区描述符表的存储单元除外，其起始地址位于 USB SRAM 基址，偏移值由 USB_BTABLE 寄存器指定。

有关寄存器说明中使用的缩写，请参见*第 58 页的第 1.2 节*。

外设寄存器可支持半字（16 位）或字（32 位）访问。

39.6.1 通用寄存器

这类寄存器会影响 USB 外设的一般行为，包括定义工作模式、中断处理、设备地址以及访问主机 PC 所更新的当前帧数。

USB 控制寄存器 (USB_CNTR)

USB control register

偏移地址：0x40

复位值：0x0003

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRM	PMAOVRM	ERRM	WKUPM	SUSPM	RESETM	SOFM	ESOFM	L1REQM	Res.	L1RESUME	RESUME	FSUSP	LP_MODE	PDWN	FRES
rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位 15 **CTRM**：正确传输中断屏蔽 (Correct transfer interrupt mask)

0：禁止正确传输 (CTR) 中断。

1：使能 CTR 中断，当 USB_ISTR 寄存器中的相应位置 1 时，将生成中断请求。

位 14 **PMAOVRM**：数据包存储区上溢/下溢中断屏蔽 (Packet memory area over / underrun interrupt mask)

0：禁止 PMAOVR 中断。

1：使能 PMAOVR 中断，当 USB_ISTR 寄存器中的相应位置 1 时，将生成中断请求。

位 13 **ERRM**：错误中断屏蔽 (Error interrupt mask)
0：禁止 ERR 中断。
1：使能 ERR 中断，当 USB_ISTR 寄存器中的相应位置 1 时，将生成中断请求。

位 12 **WKUPM**：唤醒中断屏蔽 (Wakeup interrupt mask)
0：禁止 WKUP 中断。
1：使能 WKUP 中断，当 USB_ISTR 寄存器中的相应位置 1 时，将生成中断请求。

位 11 **SUSPM**：挂起模式中断屏蔽 (Suspend mode interrupt mask)
0：禁止挂起模式请求 (SUSP) 中断。
1：使能 SUSP 中断，当 USB_ISTR 寄存器中的相应位置 1 时，将生成中断请求。

位 10 **RESETM**：USB 复位中断屏蔽 (USB reset interrupt mask)
0：禁止 RESET 中断。
1：使能 RESET 中断，当 USB_ISTR 寄存器中的相应位置 1 时，将生成中断请求。

位 9 **SOFM**：帧起始中断屏蔽 (Start of frame interrupt mask)
0：禁止 SOF 中断。
1：使能 SOF 中断，当 USB_ISTR 寄存器中的相应位置 1 时，将生成中断请求。

位 8 **ESOFM**：预期帧起始中断屏蔽 (Expected start of frame interrupt mask)
0：禁止预期帧起始 (ESOF) 中断。
1：使能 ESOF 中断，当 USB_ISTR 寄存器中的相应位置 1 时，将生成中断请求。

位 7 **L1REQM**：LPM L1 状态请求中断屏蔽 (LPM L1 state request interrupt mask)
0：禁止 LPM L1 状态请求 (L1REQ) 中断。
1：使能 L1REQ 中断，当 USB_ISTR 寄存器中的相应位置 1 时，将生成中断请求。

位 6 保留，必须保持复位值。

位 5 **L1RESUME**：LPM L1 恢复请求 (LPM L1 Resume request)
微控制器可将此位置 1 以向主机发送 LPM L1 恢复信号。信号发送结束后，此位通过硬件清零。

位 4 **RESUME**：恢复请求 (Resume request)
微控制器可将此位置 1 以向主机发送恢复信号。它必须根据 USB 规范激活并持续 1 ms 到 15 ms 的时间，经过这段时间后，主机 PC 便可驱动恢复序列直到结束。

位 3 **FSUSP**：强制挂起 (Force suspend)
软件必须在接收到 SUSP 中断时将此位置 1，当持续 3 ms 未从 USB 外设接收到任何通信数据时，将发出 SUSP 中断。
0：无影响。
1：进入挂起模式。模拟收发器中的时钟和静态功耗不受影响。如果挂起功耗是一项要求（总线供电设备），则应用软件应将在 FSUSP 后将 LP_MODE 位置 1，下面给出了具体说明。

位 2 **LP_MODE**：低功耗模式 (Low-power mode)
当挂起模式功率限制要求避免所有静态功耗时（为外部上拉电阻供电所需的静态功耗除外），使用此模式。当应用软件准备停止所有系统时钟或降低其频率以满足 USB 挂起状态的功耗要求时，应进入此状态。挂起模式期间的 USB 活动（WKUP 事件）会将此位异步复位（此位也可通过软件复位）。
0：无低功耗模式。
1：进入低功耗模式。

位 1 **PDWN**：掉电 (Power down)

当出于某种原因而需要完全禁止 USB 外设时，此位用于完全关闭所有与 USB 相关的模拟部分。此位置 1 时，USB 外设将与收发器断开，并且无法使用。

0：退出掉电模式。

1：进入掉电模式。

位 0 **FRES**：强制 USB 复位 (Force USB Reset)

0：清除 USB 复位。

1：强制 USB 外设复位，其方式与在 USB 上发出 RESET 信号相同。USB 外设保持 RESET 状态，直到软件将此位清零。使能时，将生成“USB-RESET”中断。

USB 中断状态寄存器 (USB_ISTR)

USB interrupt status register

偏移地址：0x44

复位值：0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR	PMA OVR	ERR	WKUP	SUSP	RESET	SOF	ESOF	L1REQ	Res.	Res.	DIR	EP_ID[3:0]			
r	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0			r	r	r	r	r

此寄存器包含所有中断源的状态，允许应用软件确定哪些事件引起了中断请求。

此寄存器的高半部分包含多个单独位，每个位表示一个特定事件。这些位会在发生相关事件时由硬件置 1；如果 USB_CNTR 寄存器中的相应位置 1，将生成通用中断请求。中断程序用于检查每个位，它将执行所有必要的操作并在最终将已处理的位清零。如果其中的任一位未清零，则中断仍被视为处于挂起状态，中断线将再次保持高电平。如果多个位同时置 1，则将仅生成一个中断。

端点事务完成可通过一种不同的方式进行处理，以缩短中断响应延时。端点成功完成一个事务后，硬件会立即将 CTR 位置 1，从而在 USB_CNTR 中的相应位置 1 时生成通用中断请求。端点专用中断状态的激活与 USB_CNTR 寄存器中的 CTRM 位无关。中断状态均保持激活，直到软件将相应 USB_EPnR 寄存器中的挂起位清零为止（CTR 位实际上是一个只读位）。对于与端点相关的中断，软件可使用事务方向 (DIR) 和 EP_ID 只读位来确定哪个端点发出最后一个中断请求以及调用相应的中断服务程序。

用户可通过指定软件在中断服务程序中检查 USB_ISTR 位的顺序来选择同时挂起的多个 USB_ISTR 事件的相对优先级。仅清零与事件相关且已被处理的位。在服务程序结束时，将请求另一个中断来处理剩余中断状态。

为避免意外清零某些位，建议通过加载指令清零相应位，即向所有不得改变的位写入 1，向所有待清零的位写入 0（这些位只能通过软件清零）。由于在读操作和写操作之间，另一个位会被硬件置 1，并将在微处理器有时间处理事件之前由下一次写操作清零，因此应避免读-修改-写周期。

下面详细介绍每个位：

位 15 **CTR**：正确传输 (Correct transfer)

此位由硬件置 1，用于指示端点已成功完成事务；软件可通过 DIR 和 EP_ID 位确定哪个端点请求中断。此位为只读。

位 14 **PMAOVR**：数据包存储区上溢/下溢 (Packet memory area over / underrun)

当微控制器无法及时响应 USB 存储器请求时，此位将置 1。USB 外设通过以下方式处理此事件：接收期间不发送 ACK 握手包，发送期间在发送的流中强制位填充错误；在这两种情况下，主机都将重试事务。正常操作期间，绝对不应出现 PMAOVR 中断。由于主机重试失败的事务，应用软件将有机会在此中断处理期间加快设备操作，以为下一次事务重试做准备；但这在同步传输期间不会发生（无论如何也不会重试任何同步事务），因此会导致数据丢失。此位为读/写位，但只能写入“0”，写入“1”不起作用。

位 13 **ERR**：错误 (Error)

只要出现下列其中一个错误，此标志便会置 1。
NANS：无应答。主机响应已超时。
CRC：循环冗余校验错误。接收的 CRC（令牌或数据中）之一有错误。
BST：位填充错误。在 PID、数据和/或 CRC 中的任一位置上检测到了位填充错误。
FVIO：帧格式违例。接收到了非标准帧（EOP 未处于正确位置、令牌序列错误等）。
由于 USB 外设和 PC 主机在出现错误时采用全透明方式管理重新发送，因此 USB 软件通常可忽略错误。此中断在软件开发阶段会很有用，可用于监视 USB 总线上的传输质量以及向用户指示可能的问题（例如，连接器松动、环境过于嘈杂、USB 电缆中的导线断裂等）。此位为读/写位，但只能写入“0”，写入“1”不起作用。

位 12 **WKUP**：唤醒 (Wakeup)

当在挂起模式期间检测到用于唤醒 USB 外设的活动时，此位将由硬件置 1。此事件将异步清零 CTLR 寄存器中的 LP_MODE 位并激活 USB_WAKEUP 线，这可用于通知设备的其余部分（例如，唤醒单元）恢复过程即将开始。此位为读/写位，但只能写入“0”，写入“1”不起作用。

位 11 **SUSP**：挂起模式请求 (Suspend mode request)

当持续 3 ms 未接收到通信数据时，此位将由硬件置 1，用于指示来自 USB 总线的挂起模式请求。挂起状态检查将在任何 USB 复位后立即使能，并在挂起模式激活时 (FSUSP=1) 由硬件禁止，直到恢复序列结束。此位为读/写位，但只能写入“0”，写入“1”不起作用。

位 10 **RESET**：USB 复位请求 (USB reset request)

当 USB 外设在其输入上检测到有效的 USB RESET 信号时置 1。USB 外设响应 RESET 时，仅会复位其内部协议状态机，从而在 USB_CNTR 寄存器中的 RESETM 使能位置 1 时生成中断。在 RESET 位清零之前，将始终禁止发送和接收。所有配置寄存器均不复位：微控制器必须明确清零这些寄存器（这是为了确保能够安全提供 RESET 中断，以及能够完成 RESET 后紧跟的任何事务）。功能地址和端点寄存器由 USB 复位事件复位。
此位为读/写位，但只能写入“0”，写入“1”不起作用。

位 9 **SOF**：帧起始 (Start of frame)

此位用于指示新的 USB 帧开始，在 SOF 数据包通过 USB 总线到达时置 1。中断服务程序可监视 SOF 事件，以获得 USB 主机的 1 ms 同步事件以及安全读取在接收到 SOF 数据包时更新的 USB_FNR 寄存器（这对于同步应用软件会十分有用）。此位为读/写位，但只能写入“0”，写入“1”不起作用。

位 8 **ESOF**：预期帧起始 (Expected start of frame)

当未接收到预期的 SOF 数据包时，此位由硬件置 1。主机每隔 1 ms 发送一个 SOF 数据包，但如果器件未按预期接收到数据包，则挂起定时器将发出此中断。如果连续生成三个 ESOF 中断（即，三个 SOF 数据包丢失）且两两中断之间没有任何通信数据，则将生成 SUSP 中断。即使在挂起定时器尚未锁定时丢失 SOF 数据包，此位也会置 1。此位为读/写位，但只能写入“0”，写入“1”不起作用。

位 7 **L1REQ**：LPM L1 状态请求 (LPM L1 state request)

当成功接收并应答进入 L1 状态的 LPM 命令时，此位由硬件置 1。此位为读/写位，但只能写入“0”，写入“1”不起作用。

位 6:5 保留，必须保持复位值。

位 4 **DIR**：事务方向 (Direction of transaction)

此位将由硬件根据已生成中断请求的成功事务的方向写入。
如果 DIR 位为 0，则与中断端点相关的 USB_EPnR 寄存器中的 CTR_TX 位置 1。中断事务的类型为 IN（USB 外设将数据发送给主机 PC）。
如果 DIR 位为 1，则与中断端点相关的 USB_EPnR 寄存器中的 CTR_RX 位置 1，或者 CTR_TX 和 CTR_RX 位均置 1。中断事务的类型为 OUT（USB 外设从主机 PC 接收数据），或者两个挂起事务均等待处理。
此信息可供应用软件用于访问与触发事务相关的 USB_EPnR 寄存器的位，因为它表示中断挂起的方向。此位为只读。

位 3:0 **EP_ID[3:0]**：端点标识符 (Endpoint Identifier)

这些位将由硬件根据已生成中断请求的端点编号写入。如果有多个端点事务挂起，则硬件将写入与优先级最高的端点相关的端点标识符，优先级的定义方式如下：定义两个端点组，同步端点和双缓冲批量端点的优先级最高，其他端点次之。如果来自同一组的多个端点请求一个中断，则 USB_ISTR 寄存器中的 EP_ID 位将按照请求端点寄存器的编号升序分配，即 EP0R 的优先级最高，EP1R 次之，依此类推。应用软件可按照此优先级方案为每个端点分配一个寄存器，以便对同时发生的多个端点请求进行适当排序。这些位为只读。

USB 帧数寄存器 (USB_FNR)

USB frame number register

偏移地址：0x48

复位值：0x0XXX，其中 X 未定义

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDP	RXDM	LCK	LSOF[1:0]		FN[10:0]										
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 15 **RXDP**：接收数据 + 线状态 (Receive data + line status)

此位可用于观察所接收的数据正上行端口数据线的状态。可以在挂起结束程序期间使用此位来帮助确定唤醒事件。

位 14 **RXDM**：接收数据 - 线状态 (Receive data - line status)

此位可用于观察所接收的数据负上行端口数据线的状态。可以在挂起结束程序期间使用此位来帮助确定唤醒事件。

位 13 **LCK**：锁定 (Locked)

当在 USB 复位状态结束后或 USB 恢复序列结束后接收到至少两个连续的 SOF 数据包时，此位将由硬件置 1。锁定后，帧定时器将保持此状态，直到发生 USB 复位或 USB 挂起事件。

位 12:11 **LSOF[1:0]**：丢失 SOF (Lost SOF)

这些位在生成 ESOF 中断时由硬件写入，用于对丢失的连续 SOF 数据包进行计数。接收到 SOF 数据包时，这些位将被清零。

位 10:0 **FN[10:0]**：**帧数** (Frame number)

此位域包含最后接收的 SOF 数据包中的 11 位帧数。主机每发送一个帧，帧数便会递增，帧数对于同步传输十分有用。此位域在生成 SOF 中断时更新。

USB 设备地址 (USB_DADDR)

USB device address

偏移地址：0x4C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EF	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
								rw	rw	rw	rw	rw	rw	rw	rw

位 15:8 保留

位 7 **EF**：使能功能 (Enable function)

此位由软件置 1，用于使能 USB 设备。此设备的地址包含在下面的 ADD[6:0] 位中。如果此位为 0，则无论 USB_EPnR 寄存器的设置为何，都不会处理任何事务。

位 6:0 **ADD[6:0]**：设备地址 (Device Address)

这些位包含枚举过程中由主机 PC 分配的 USB 功能地址。此位域和相关 USB_EPnR 寄存器中的端点地址 (EA) 位域均必须与 USB 令牌中包含的信息匹配，才能处理所需端点的事务。

缓冲区表地址 (USB_BTABLE)

Buffer table address

偏移地址：0x50

复位值：0x0000

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

位 15:3 **BTABLE[15:3]**：缓冲区表 (Buffer table)

这些位包含专用数据包存储器中缓冲区分配表的起始地址。此表描述了各个端点缓冲区的位置和大小，它必须与 8 字节边界对齐（低 3 位始终为 0）。在寻址到此设备的每个事务开始时，USB 外设将读取此表中与所寻址端点相关的元素，以获取端点的缓冲区起始位置和缓冲区大小（请参见*第 1249 页的数据包缓冲区的结构和用法*）。

位 2:0 保留，由硬件强制清零。

LPM 控制和状态寄存器 (USB_LPMCSR)

LPM control and status register

偏移地址：0x54

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BESL[3:0]				REM WAKE	Res.	LPM ACK	LPM EN
								r				r		rw	rw

位 15:8 保留，必须保持复位值。

位 7:4 **BESL[3:0]**：BESL 值 (BESL value)
这些位包含通过最后应答的 LPM 令牌接收的 BESL 值

位 3 **REMWAKE**：bRemoteWake 值 (bRemoteWake value)
此位包含通过最后应答的 LPM 令牌接收的 bRemoteWake 值

位 2 保留

位 1 **LPMACK**：LPM 令牌应答使能 (LPM Token acknowledge enable)
0：有效 LPM 令牌将为 NYET。
1：有效 LPM 令牌将为 ACK。
只有成功完成 LPM 事务后才会返回 NYET/ACK：
EXT 令牌和 LPM 令牌中均无错误（否则返回 ERROR）
接收到有效 bLinkState = 0001B (L1)（否则返回 STALL）

位 0 **LPMEN**：LPM 支持使能 (LPM support enable)
该位由软件置 1，用于在 USB 设备内使能 LPM 支持。如果此位为 0，则不会处理任何 LPM 事务。

电池充电检测器 (USB_BCDR)

Battery charging detector

偏移地址：0x58

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPPU	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PS2 DET	SDET	PDET	DC DET	SDEN	PDEN	DCD EN	BCD EN
rw								r	r	r	r	rw	rw	rw	rw

位 15 **DPPU**：DP 上拉控制 (DP pull-up control)

此位由软件置 1，用于使能 DP 线上的内置上拉。必要时，可通过用户软件将此位清零以向主机发出断开信号。

位 14:8 保留，必须保持复位值。

位 7 **PS2DET**：DM 上拉检测状态 (DM pull-up detection status)

此位只在 PD 期间有效，用于指示 DM 电压电平与 V_{LGC} 阈值之间的比较结果。正常情况下，DM 电平应低于此阈值。如果 DM 电平高于此阈值，则意味着 DM 已外部拉为高电平。原因可能是连接到 PS2 端口（同时上拉 DP 和 DM 线）或某些不遵循 BCD 规范的专有充电器。

0：检测到正常端口（连接到 SDP、ACA、CDP 或 DCP）。

1：检测到 PS2 端口或专有充电器。

位 6 **SDET**：二次检测 (SD) 状态 (Secondary detection (SD) status)

此位指示 SD 的结果。

0：检测到 CDP。

1：检测到 DCP。

位 5 **PDET**：初次检测 (PD) 状态 (Primary detection (PD) status)

此位指示 PD 的结果。

0：未检测到 BCD 支持（连接到 SDP 或专有设备）。

1：检测到 BCD 支持（连接到 ACA、CDP 或 DCP）。

位 4 **DCDET**：数据接触点检测 (DCD) 状态 (Data contact detection (DCD) status)

此位指示 DCD 的结果。

0：未检测到数据线接触。

1：检测到数据线接触。

位 3 **SDEN**：二次检测 (SD) 模式使能 (Secondary detection (SD) mode enable)

该位由软件置 1，用于将 BCD 置于 SD 模式。要正确工作，应只选择一个检测模式（DCD、PD、SD 或 OFF）。

位 2 **PDEN**：初次检测 (PD) 模式使能 (Primary detection (PD) mode enable)

该位由软件置 1，用于将 BCD 置于 PD 模式。要正确工作，应只选择一个检测模式（DCD、PD、SD 或 OFF）。

位 1 **DCDEN**：数据接触点检测 (DCD) 模式使能 (Data contact detection (DCD) mode enable)

该位由软件置 1，用于将 BCD 置于 DCD 模式。要正确工作，应只选择一个检测模式（DCD、PD、SD 或 OFF）。

位 0 **BCDEN**：电池充电器检测 (BCD) 使能 (Battery charging detector (BCD) enable)

该位由软件置 1，用于在 USB 设备内使能 BCD 支持。使能后，USB PHY 完全由 BCD 控制，无法用于正常通信。BCD 检测完成后，应通过将该位清零使 BCD 置于 OFF 模式，以实现正常的 USB 操作。

端点相关寄存器

这类寄存器的数量因 USB 外设设计要处理的端点数而异。USB 外设最多支持 8 个双向端点。每个 USB 设备必须支持一个控制端点，端点地址（EA 位）必须设置为 0。当使能多个端点编号值相同的端点时，USB 外设将以一种未定义的方式工作。对于每个端点而言，USB_EPnR 寄存器可用于存储端点特定的信息。

USB 端点 n 寄存器 (USB_EPnR)，n=[0..7]

USB endpoint n register

偏移地址：0x00 到 0x1C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR_RX	DTOG_RX	STAT_RX[1:0]		SETUP	EP TYPE[1:0]		EP_KIND	CTR_TX	DTOG_TX	STAT_TX[1:0]		EA[3:0]			
rc_w0	t	t	t	r	rw	rw	rw	rc_w0	t	t	t	rw	rw	rw	rw

当从 USB 总线接收到 USB 复位或通过 CTLR 寄存器中的 FRES 位强制 USB 复位时，这些位也会复位；但 CTR_RX 和 CTR_TX 位除外，它们将保持不变，以避免丢失紧跟在 USB 复位事件后的数据包通知。每个端点都有自己的 USB_EPnR 寄存器，其中 *n* 是端点标识符。

由于在读操作和写操作之间，一些位会被硬件置 1，并将在 CPU 有时间检测变化之前由下一次写操作修改，因此应避免这类寄存器上的读-修改-写周期。为此，受该问题影响的所有位都有一个“不变量”值，无需修改时必须使用该值。建议通过加载指令来修改这类寄存器，即向只能通过硬件修改的所有位写入其“不变量”值。

位 15 **CTR_RX**：接收传输正确 (Correct Transfer for reception)

当此端点上成功完成 OUT/SETUP 事务时，此位将由硬件置 1；软件只能将此位清零。如果 USB_CNTR 寄存器中的 CTRM 位相应地置 1，则会生成通用中断条件和端点相关中断条件，后者始终有效。已发生事务的类型（OUT 或 SETUP）可通过下述的 SETUP 位来确定。

以 NAK 或 STALL 握手结束的事务不会将此位置 1，因为在出现协议错误或数据翻转不匹配时，实际上并未传输任何数据。

此位为读/写位，但只能写入“0”，写入“1”不起作用。

位 14 **DTOG_RX**：数据翻转，用于接收传输 (Data Toggle, for reception transfers)

如果端点不是同步端点，则此位包含要接收的下一个数据包的数据翻转位的预期值（0=DATA0，1=DATA1）。当 ACK 握手发送给 USB 主机时，硬件会将此位翻转，接着将接收到具有匹配数据 PID 值的数据包；如果端点定义为控制端点，则硬件会在接收到寻址到此端点的 SETUP PID 时将此位清零。

如果端点正在使用双缓冲功能，则此位还用于支持数据包缓冲区交换（请参见*第 39.5.3 节：双缓冲端点*）。

如果端点是同步端点，则此位只用于支持数据包缓冲区交换，因为此类端点不使用数据翻转，仅传输 DATA0 数据包（请参见*第 39.5.4 节：同步传输*）。硬件在数据包接收结束后立即翻转此位，因为同步传输不使用握手。

此位还可通过软件同步以初始化其值（当端点不是控制端点时是强制的）或强制使用特定的数据翻转/数据包缓冲区。当应用软件写入“0”时，DTOG_RX 的值保持不变，而写入“1”时，此位的值将进行翻转。此位为读/写位，但只能通过写入 1 来翻转。

位 13:12 **STAT_RX [1:0]**：状态位，用于接收传输 (Status bits, for reception transfers)

这些位包含有关端点状态的信息，如*第 1267 页的表 229：接收状态编码*中所列。这些位可通过软件翻转以初始化其值。当应用软件写入“0”时，此位的值保持不变，而写入“1”时，此位的值将进行翻转。当寻址到此端点的 OUT 或 SETUP（仅控制）事务已正确传输后 (CTR_RX=1)，硬件会将 STAT_RX 位设置为 NAK，因此软件有时间在应答新事务之前准备接收的数据。

双缓冲批量端点实现了一种特殊的事务流控制，可基于缓冲区可用性条件来控制状态（请参见*第 39.5.3 节：双缓冲端点*）。

如果端点定义为同步端点，则其状态只能是“VALID”或“DISABLED”，在这种情况下，硬件无法在事务成功完成后更改端点的状态。如果软件为同步端点将 STAT_RX 位设置为“STALL”或“NAK”，则 USB 外设的行为是不明确的。这些位为读/写位，但只能通过写入 1 来同步。

位 11 **SETUP**：建立事务已完成 (Setup transaction completed)

此位为只读位，将在最后完成的事务为 SETUP 时由硬件置 1。此位仅为控制端点更改其值。它必须在接收事务成功完成（CTR_RX 事件）时进行检查，以确定所发生事务的类型。为防止中断服务程序因接下来传入的令牌而更改 SETUP 位，此位将在 CTR_RX 位为 1 时保持冻结；其状态将在 CTR_RX 为 0 时更改。此位为只读。

位 10:9 **EP_TYPE[1:0]**：端点类型 (Endpoint type)

这些位配置此端点的行为，如*第 1268 页的表 230：端点类型编码*所述。端点 0 必须始终为控制端点，每个 USB 功能必须至少有一个地址为 0 的控制端点，但必要时也可以有其他控制端点。只有控制端点会处理 SETUP 事务，其他类型的端点会忽略 SETUP 事务。SETUP 事务不能通过 NAK 或 STALL 应答。如果控制端点定义为 NAK，则 USB 外设将不会应答，而会在接收到 SETUP 事务时在接收方向上模拟接收错误。如果控制端点在接收方向上定义为 STALL，则 SETUP 数据包仍将被接受，从而传输数据并发出 CTR 中断。OUT 事务的接收以常规方式处理，即使端点为控制端点也如此。

批量端点和中断端点的行为十分相似，唯一的区别在于使用 EP_KIND 配置位可实现的特殊功能。

*第 39.5.4 节：同步传输*中介绍了同步端点的用法。

位 8 **EP_KIND**：端点种类 (Endpoint kind)

此位的含义取决于 EP_TYPE 位所配置的端点类型。*表 231* 汇总了不同的含义。

DBL_BUF：此位通过软件置 1，用于为此批量端点使能双缓冲功能。*第 39.5.3 节：双缓冲端点*中介绍了双缓冲批量端点的用法。

STATUS_OUT：此位通过软件置 1，用于指示将出现状态输出事务：在这种情况下，所有包含非零字节数据的 OUT 事务均通过“STALL”而非“ACK”来应答。此位可用于提高应用软件针对控制传输期间的协议错误的鲁棒性（仅限控制端点）。当 STATUS_OUT 复位时，OUT 事务可以有任何数量的字节（根据需要）。

位 7 **CTR_TX**：发送传输正确 (Correct Transfer for transmission)

当此端点上成功完成 IN 事务时，此位将由硬件置 1；软件只能将此位清零。如果 USB_CNTR 寄存器中的 CTRM 位相应地置 1，则会生成通用中断条件和端点相关中断条件，后者始终有效。

以 NAK 或 STALL 握手结束的事务不会将此位置 1，因为在出现协议错误或数据翻转不匹配时，实际上并未传输任何数据。

此位为读/写位，但只能写入“0”。

位 6 **DTOG_TX**：数据翻转，用于发送传输 (Data Toggle, for transmission transfers)

如果端点不是同步端点，则此位包含要发送的下一个数据包的数据翻转位的预期值（0=DATA0，1=DATA1）。当从 USB 主机接收到 ACK 握手时，硬件会将此位翻转，接着将发送一个数据包。如果端点定义为控制端点，则硬件会在接收到寻址到此端点的 SETUP PID 时将此位置 1。

如果端点正在使用双缓冲功能，则此位还用于支持数据包缓冲区交换（请参见*第 39.5.3 节：双缓冲端点*）。

如果端点是同步端点，则此位用于支持数据包缓冲区交换，因为此类端点不使用数据同步，仅传输 DATA0 数据包（请参见*第 39.5.4 节：同步传输*）。硬件在数据包发送结束后立即翻转此位，因为同步传输不使用握手。

此位还可通过软件翻转以初始化其值（当端点不是控制端点时十分必要）或强制使用特定的数据翻转/数据包缓冲区。当应用软件写入“0”时，DTOG_TX 的值保持不变，而写入“1”时，此位的值将进行同步。此位为读/写位，但只能通过写入 1 来同步。

位 5:4 **STAT_TX [1:0]**：状态位，用于发送传输 (Status bits, for transmission transfers)

这些位包含有关端点状态的信息，如*表 232* 所列。这些位可通过软件翻转以初始化其值。当应用软件写入“0”时，此位的值保持不变，而写入“1”时，此位的值将进行翻转。当寻址到此端点的 IN 或 SETUP（仅控制）事务已正确传输后 (CTR_TX=1)，硬件会将 STAT_TX 位设置为 NAK。然后，它将等待软件准备下一组要发送的数据。

双缓冲批量端点实现了一种特殊的事务流控制，可基于缓冲区可用性来条件控制状态（请参见*第 39.5.3 节：双缓冲端点*）。

如果端点定义为同步端点，则其状态只能是“VALID”或“DISABLED”。因此，硬件无法在事务成功完成后更改端点的状态。如果软件为同步端点将 STAT_TX 位设置为“STALL”或“NAK”，则 USB 外设的行为是不确定的。这些位为读/写位，但只能通过写入 1 来同步。

位 3:0 **EA[3:0]**：端点地址 (Endpoint address)

软件必须在此位域中写入 4 位地址，用于确定定向到此端点的事务。必须在使能相应端点前写入一个值。

表 229. 接收状态编码

STAT_RX[1:0]	意义
00	**DISABLED**：忽略寻址到此端点的所有接收请求。
01	**STALL**：停止端点，所有接收请求导致 STALL 握手。
10	**NAK**：不应答端点，所有接收请求导致 NAK 握手。
11	**VALID**：将此端点使能用于接收。

表 230. 端点类型编码

EP_TYPE[1:0]	意义
00	BULK
01	CONTROL
10	ISO
11	INTERRUPT

表 231. 端点种类的意义

EP_TYPE[1:0]		EP_KIND 的意义
00	BULK	DBL_BUF
01	CONTROL	STATUS_OUT
10	ISO	未使用
11	INTERRUPT	未使用

表 232. 发送状态编码

STAT_TX[1:0]	意义
00	**DISABLED**：忽略寻址到此端点的所有发送请求。
01	**STALL**：停止端点，所有发送请求导致 STALL 握手。
10	**NAK**：不应答端点，所有发送请求导致 NAK 握手。
11	**VALID**：将此端点使能用于发送。

39.6.2 缓冲区描述符表

注： *缓冲区描述符表位于数据包缓冲区存储器内的独立“USB SRAM”地址空间。*

尽管缓冲区描述符表位于数据包缓冲区存储器（“USB SRAM”区）内，但其条目可视为额外的寄存器，用于配置在 USB 宏单元和设备之间交换数据所使用的数据包缓冲区的位置和大小。

第一个数据包存储单元位于 USB SRAM 基址。下文介绍了与 USB_EPnR 寄存器相关的缓冲区描述符表条目。数据包存储器只应按字节（8 位）或半字（16 位）访问。不允许按字（32 位）访问。

有关数据包缓冲区和缓冲区说明符表用法的详细说明，可参见*第 1249 页的数据包缓冲区的结构和用法*。

发送缓冲区地址 n (USB_ADDRn_TX)

偏移地址：[USB_BTABLE] + n*8

注： *对于 IN 方向上的双缓冲端点或同步端点，此地址单元用于 USB_ADDRn_TX_0。*

对于 OUT 方向上的双缓冲端点或同步端点，此地址单元用于 USB_ADDRn_RX_0。

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

位 15:1 **ADDRn_TX[15:1]**：发送缓冲区地址 (Transmission buffer address)

这些位指向将包含如下内容的数据包缓冲区的起始地址：与 USB_EPnR 寄存器相关的端点将在寻址到它的下一个 IN 令牌中发送的数据。

位 0 必须始终写为 0，因为数据包存储器为半字宽，所有数据包缓冲区必须按半字对齐。

发送字节数 n (USB_COUNTn_TX)

偏移地址：[USB_BTABLE] + n*8 + 2

注： *对于 IN 方向上的双缓冲端点或同步端点，此地址单元用于 USB_COUNTn_TX_0。*

对于 OUT 方向上的双缓冲端点或同步端点，此地址单元用于 USB_COUNTn_RX_0。

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

位 15:10 不使用这些位，因为 USB 规范将数据包大小限制为 1023 字节。USB 外设不会考虑这些位的值。

位 9:0 **COUNTn_TX[9:0]**：发送字节数 (Transmission byte count)

这些位包含与 USB_EPnR 寄存器相关的端点将在寻址到它的下一个 IN 令牌中发送的字节数。

接收缓冲区地址 n (USB_ADDRn_RX)

偏移地址：[USB_BTABLE] + n*8 + 4

注： *对于 OUT 方向上的双缓冲端点或同步端点，此地址单元用于 USB_ADDRn_RX_1。*
对于 IN 方向上的双缓冲端点或同步端点，此地址单元用于 USB_ADDRn_TX_1。

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

位 15:1 **ADDRn_RX[15:1]**：接收缓冲区地址 (Reception buffer address)
这些位指向将包含如下内容的数据包缓冲区的起始地址：与 USB_EPnR 寄存器相关的端点在寻址到它的下一个 OUT/SETUP 令牌中接收的数据。

位 0 此位必须始终写为 0，因为数据包存储器为半字宽，所有数据包缓冲区必须按半字对齐。

接收字节数 n (USB_COUNTn_RX)

偏移地址：[USB_BTABLE] + n*8 + 6

注： *对于 OUT 方向上的双缓冲端点或同步端点，此地址单元用于 USB_COUNTn_RX_1。*
对于 IN 方向上的双缓冲端点或同步端点，此地址单元用于 USB_COUNTn_TX_1。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLSIZE	NUM_BLOCK[4:0]					COUNTn_RX[9:0]									
rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r	r	r

此表存储单元用于存储数据包接收期间需要的两个不同的值。最高有效位包含已分配缓冲区大小的定义，以便进行缓冲区溢出检测；此存储单元的最低有效位由 USB 外设在接收结束时回写，以给出已接收字节的实际数量。由于可用位数受到限制，缓冲区大小使用已分配存储块的数量来表示，其中可通过选择块大小在精细粒度 / 小缓冲区与粗略粒度 / 大缓冲区之间进行折衷。已分配缓冲区的大小是端点描述符的一部分，通常在枚举过程中根据其 maxPacketSize 参数值定义（请参见“通用串行总线规范”）。

位 15 **BL_SIZE**：块大小 (Block size)
此位选择用于定义已分配缓冲区的存储器块的大小。
– 如果 BL_SIZE = 0，则存储器块为 2 字节，这是半字宽存储器中允许的最小块。使用此块大小时，所分配缓冲区的大小在 2 字节到 62 字节范围内。
– 如果 BL_SIZE = 1，则存储器块为 32 字节，可达到 USB 规范定义的最大数据包长度。使用此块大小时，理论上所分配缓冲区的大小在 32 字节到 1024 字节范围内，这是 USB 标准规范允许的最大数据包长度。不过，可用的缓冲区大小受可用缓冲区存储器的限制。

位 14:10 **NUM_BLOCK[4:0]**：块数 (Number of blocks)
这些位定义分配到此数据包缓冲区的存储器块的数量。已分配存储器的实际数量取决于*表 233* 中给出的 BL_SIZE 值。

位 9:0 **COUNTn_RX[9:0]**：接收字节数 (Reception byte count)
这些位包含与 USB_EPnR 寄存器相关的端点在寻址到它的最后一个 OUT/SETUP 事务期间接收的字节数。

表 233. 已分配缓冲区存储器的定义

NUM_BLOCK[4:0] 的值	已分配的存储器 （当 BL_SIZE=0 时）	已分配的存储器 （当 BL_SIZE=1 时）
0 (‘00000)	不允许	32 字节
1 (‘00001)	2 字节	64 字节
2 (‘00010)	4 字节	96 字节
3 (‘00011)	6 字节	128 字节
...	...	...
14 (‘01110)	28 字节	480 字节
15 (‘01111)	30 字节	
16 (‘10000)	32 字节	
...	...	...
29 (‘11101)	58 字节	
30 (‘11110)	60 字节	
31 (‘11111)	62 字节	N/A

39.6.3 USB 寄存器映射

下表提供了 USB 寄存器映射和复位值。

表 234. USB 寄存器映射和复位值

偏移	寄存器	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
0x00	**USB_EP0R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]		SETUP	EP TYPE [1:0]		EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]		EA[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**USB_EP1R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]		SETUP	EP TYPE [1:0]		EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]		EA[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	**USB_EP2R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]		SETUP	EP TYPE [1:0]		EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]		EA[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**USB_EP3R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]		SETUP	EP TYPE [1:0]		EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]		EA[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**USB_EP4R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]		SETUP	EP TYPE [1:0]		EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]		EA[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**USB_EP5R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]		SETUP	EP TYPE [1:0]		EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]		EA[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**USB_EP6R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]		SETUP	EP TYPE [1:0]		EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]		EA[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**USB_EP7R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]		SETUP	EP TYPE [1:0]		EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]		EA[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20-0x3F	Reserved																																
0x40	**USB_CNTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTRM	PMAOVRM	ERRM	WKUPM	SUSPM	RESETM	SOFM	ESOFM	L1REQM	Res.	L1RESUME	RESUME	FSUSP	LP_MODE	PDWN	FRES
	Reset value																	0	0	0	0	0	0	0	0	0		0	0	0	0	1	1
0x44	**USB_ISTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR	PMAOVR	ERR	WKUP	SUSP	RESET	SOF	ESOF	L1REQ	Res.	Res.	DIR	EP_ID[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0			0	0	0	0	0
0x48	**USB_FNR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDP	RXDM	LCK	LSOF [1:0]		FN[10:0]										
	Reset value																	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x
0x4C	**USB_DADDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EF	ADD[6:0]						
	Reset value																									0	0	0	0	0	0	0	0

表 234. USB 寄存器映射和复位值（续）

偏移	寄存器	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
0x50	**USB_BTABLE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BTABLE[15:3]													Res.	Res.	Res.
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x54	**USB_LPMCSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BESL[3:0]				REMWAKE	Res.	LPMACK	LPMEN
	Reset value																									0	0	0	0	0		0	0
0x58	**USB_BCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPPU	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PS2DET	SDET	PDET	DCDET	SDEN	PDEN	DCDEN	BCDEN
	Reset value																	0								0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

40 时钟恢复系统 (CRS)

40.1 简介

时钟恢复系统 (CRS) 是一个作用于内部精细粒度可调 RC 振荡器 HSI48 的高级数字控制器。CRS 提供一种十分强大的方法，可通过与可选同步信号进行比较来评估振荡器输出频率。它能够根据测得的频率误差值自动调整振荡器微调，同时还可以进行手动微调。

CRS 非常适合为 USB 外设提供精密时钟。在这种情况下，同步信号可由 USB 总线上的帧起始 (SOF) 数据包信号提供，USB 主机以 1 ms 的时间间隔精确发送该信号。

同步信号也可由 LSE 振荡器输出提供，或者由用户软件生成。

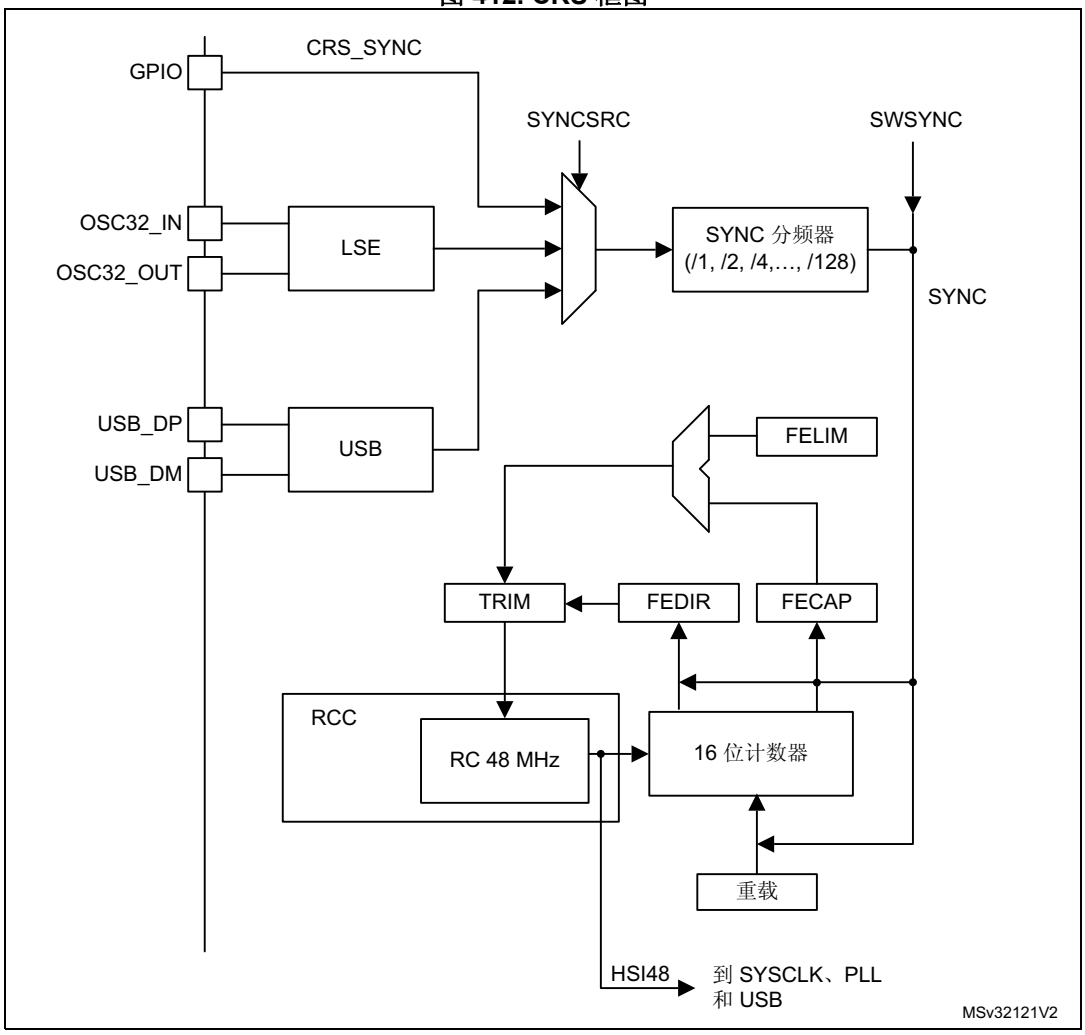
40.2 CRS 主要特性

- 具有可编程预分频器和极性的可选同步源：
  - LSE 振荡器输出
  - 数据包接收
- 可由软件生成同步脉冲
- 振荡器自动微调功能，无需 CPU 操作
- 手动控制选项，可加快启动融合
- 16 位频率误差计数器，用于自动误差值捕捉和重载
- 可编程限值，用于自动频率误差值评估和状态报告
- 可屏蔽中断/事件：
  - 预期同步 (ESYNC)
  - 同步正常 (SYNCOK)
  - 同步警告 (SYNCWARN)
  - 同步或微调错误 (ERR)

40.3 CRS 功能说明

40.3.1 CRS 框图

图 412. CRS 框图



40.3.2 同步输入

有关 CRS 同步源配置的更多信息，请参见*第 40.6.2 节：CRS 配置寄存器 (CRS_CFGR)*。

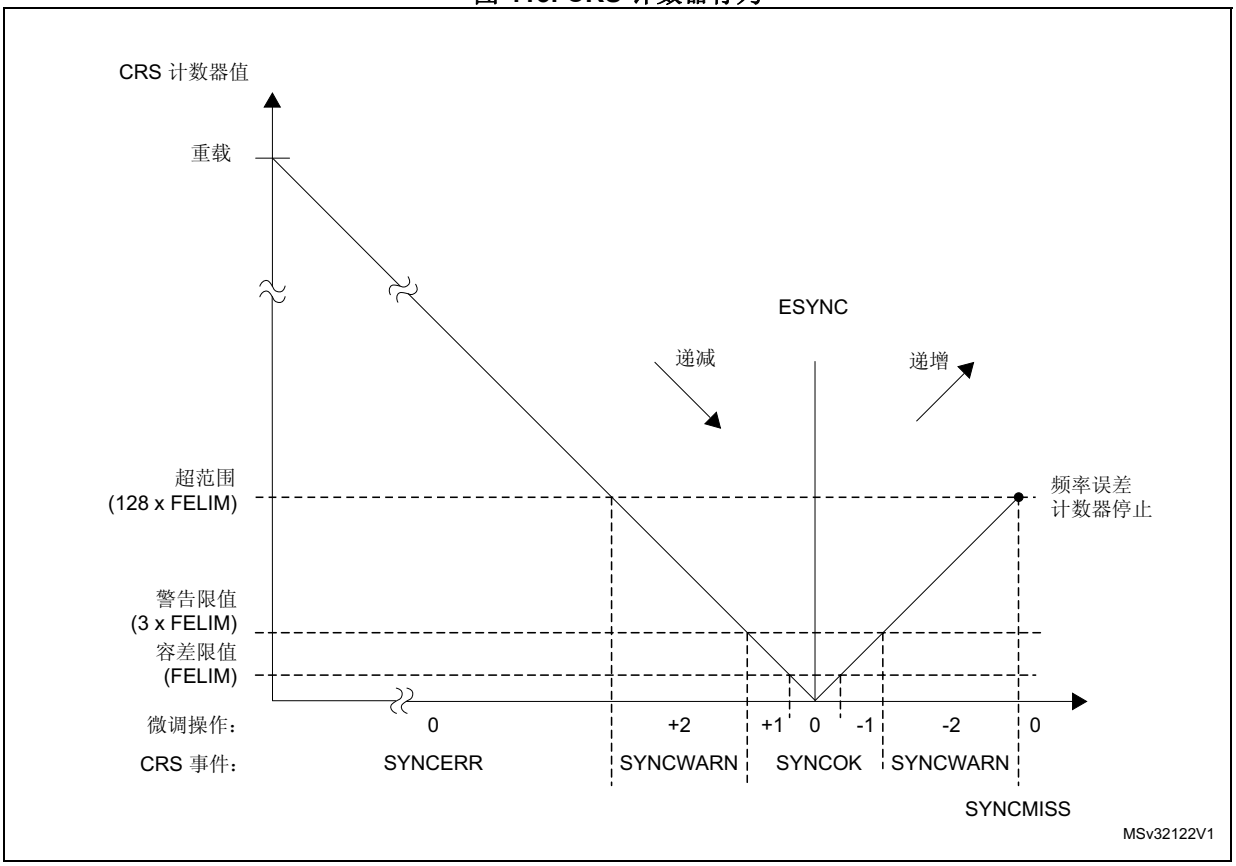
还可通过用软件将 CRS_CR 寄存器中的 SWSYNC 位置 1 的方法来生成同步事件。

40.3.3 频率误差测量

频率误差计数器是一个 16 位递减/递增计数器，会在每个 SYNC 事件发生时重载 RELOAD 值。它开始递减计数，直至达到零值，此时会生成 ESYNC（预期同步）事件。随后它将递增计数到 OUTRANGE 限值，这种情况下最终将停止计数（如果未接收到 SYNC 事件）并生成 SYNCMISS 事件。OUTRANGE 限值定义为频率误差限值（CRS_CFGR 寄存器的 FELIM 字段）乘以 128。

当检测到 SYNC 事件时，频率误差计数器的实际值及其方向存储在 CRS_ISR 寄存器的 FECAP（频率误差捕捉）字段和 FEDIR（频率误差方向）位中。当在递减计数阶段（达到零值之前）期间检测到 SYNC 事件时，意味着实际频率小于目标频率（因此，TRIM 值应递增）；当在递增计数期间检测到 SYNC 事件时，意味着实际频率大于目标频率（因此，TRIM 值应递减）。

图 413. CRS 计数器行为



40.3.4 频率误差评估和自动微调

测得的频率误差通过将频率值与一组限值进行比较进行估算：

- TOLERANCE LIMIT，在 CRS_CFGR 寄存器的 FELIM 字段中直接给出
- WARNING LIMIT，定义为 3 * FELIM 值
- OUTRANGE（误差限值），定义为 128 * FELIM 值

该比较结果用于生成状态指示以及控制自动微调，自动微调通过将 CRS_CR 寄存器中的 AUTOTRIMEN 位置 1 来使能。

- 当频率误差低于容差限值时，意味着 TRIM 字段中的实际微调值是最优值，因此无需任何微调操作。
  - 指示 SYNCOK 状态
  - AUTOTRIM 模式下的 TRIM 值无变化
- 当频率误差低于警告限值但高于或等于容差限值时，意味着需要某种微调操作，但只需一步微调便可达到最优 TRIM 值。
  - 指示 SYNCOK 状态
  - 在 AUTOTRIM 模式下通过一步微调来调整 TRIM 值
- 当频率误差高于或等于警告限值但低于误差限值时，意味着需要更强的微调操作，并且存在下一周期无法达到最优 TRIM 值的风险。
  - 指示 SYNCWARN 状态
  - 在 AUTOTRIM 模式下通过两步微调来调整 TRIM 值
- 当频率误差高于或等于误差限值时，意味着频率超出微调范围。当 SYNC 输入不干净或某个 SYNC 脉冲丢失时（例如，当一个 USB SOF 被破坏时）。也会发生这种情况。
  - 指示 SYNCERR 或 SYNCMISS 状态
  - AUTOTRIM 模式下的 TRIM 值无变化

注： *如果 TRIM 字段的实际值十分接近其限值，则自动微调会强制其上溢或下溢，之后 TRIM 值会恰好设置为限值，此时将指示 TRIMOVF 状态。*

在 AUTOTRIM 模式（CRS_CR 寄存器中的 AUTOTRIMEN 位置 1）下，CRS_CR 的 TRIM 字段由硬件调整并且是只读的。

40.3.5 CRS 初始化和配置

RELOAD 值

RELOAD 值应根据预分频后目标频率与同步源频率之比进行选择。该值随后会减 1，以在零值时达到预期同步。具体公式如下：

$$RELOAD = (f_{TARGET} / f_{SYNC}) - 1$$

RELOAD 字段的复位值对应于 48 MHz 的目标频率和 1 kHz 的同步信号频率（来自 USB 的 SOF 信号）。

FELIM 值

FELIM 值的选择与 HSI48 振荡器的特性及其典型微调步长紧密相关。最优值对应于微调步长的一半，以 HSI48 振荡器时钟节拍数表示。可使用以下公式：

$$FELIM = (f_{TARGET} / f_{SYNC}) * STEP[\%] / 100\% / 2$$

结果应始终舍入为最接近的整数值以获得最佳微调响应。如果应用中不需要频繁的微调操作，可通过稍微增大 FELIM 值来增加微调滞后。

FELIM 字段的复位值对应于 (f_{TARGET} / f_{SYNC}) = 48000 以及 0.14% 的典型微调步长。

注意： 错误配置 RELOAD 和 FELIM 字段无法实现硬件保护，从而导致不定微调响应。预期工作模式需要正确设置 RELOAD 值（根据同步源频率），该值还大于 128 * FELIM 值（OUTRANGE 限值）。

40.4 CRS 低功耗模式

表 235. 低功耗模式对 CRS 的作用

模式	说明
睡眠	无影响。 CRS 中断可使器件退出睡眠模式。
停止	CRS 寄存器被冻结。 CRS 停止工作，直到退出停止或待机模式且 HSI48 振荡器重启。
待机	

40.5 CRS 中断

表 236. 中断控制位

中断事件	事件标志	使能控制位	清零标志位
预期同步	ESYNCF	ESYNCIE	ESYNCC
同步正常	SYNCOKF	SYNCOKIE	SYNCOKC
同步警告	SYNCWARNF	SYNCWARNIE	SYNCWARNC
同步或微调错误 （TRIMOVF、SYNCMISS 和 SYNCERR）	ERRF	ERRIE	ERRC

40.6 CRS 寄存器

有关寄存器说明中使用的缩写，请参见参考手册中的*第 58 页的第 1.2 节*。

外设寄存器可按字（32 位）进行访问。

40.6.1 CRS 控制寄存器 (CRS_CR)

CRS control register

偏移地址：0x00

复位值：0x0000 2000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	TRIM[5:0]						SW SYNC	AUTO TRIMEN	CEN	Res.	ESYNC IE	ERRIE	SYNC WARNIE	SYNC OKIE
		rw	rw	rw	rw	rw	rw	rt_w1	rw	rw		rw	rw	rw	rw

位 31:14 保留，必须保持复位值。

位 13:8 **TRIM[5:0]**：HSI48 振荡器平滑微调 (HSI48 oscillator smooth trimming)

这些位为 HSI48 振荡器提供用户可编程的微调值。可通过编程使其适应电压和温度的差异，使 HSI48 振荡器的频率更为准确。

默认值为 32，该值对应于微调间隔的中间值。微调步在产品数据手册中指定。较高的 TRIM 值对应于较高的输出频率。

当 AUTOTRIMEN 位置 1 时，该字段由硬件控制并且是只读的。

位 7 **SWSYNC**：生成软件 SYNC 事件 (Generate software SYNC event)

此位由软件置 1 以生成软件 SYNC 事件。它由硬件自动清零。

0：不执行任何操作。

1：生成软件 SYNC 事件。

位 6 **AUTOTRIMEN**：自动微调使能 (Automatic trimming enable)

此位根据在两个 SYNC 事件间测得的频率误差使能 TRIM 位的自动硬件调整。如果此位置 1，则 TRIM 位是只读的。TRIM 值可通过硬件一次调整一步或两步，具体取决于测得的频率误差值。更多详细信息，请参见*第 40.3.4 节：频率误差评估和自动微调*。

0：禁止自动微调，TRIM 位可由用户调整。

1：使能自动微调，TRIM 位是只读的且由硬件控制。

位 5 **CEN**：频率误差计数器使能 (Frequency error counter enable)

此位为频率误差计数器使能振荡器时钟。

0：禁止频率误差计数器

1：使能频率误差计数器

当此位置 1 时，CRS_CFGR 寄存器被写保护，无法修改。

位 4 保留，必须保持复位值。

位 3 **ESYNCIE**：预期 SYNC 中断使能 (Expected SYNC interrupt enable)
0：禁止预期 SYNC (ESYNCF) 中断
1：使能预期 SYNC (ESYNCF) 中断

位 2 **ERRIE**：同步或微调错误中断使能 (Synchronization or trimming error interrupt enable)
0：禁止同步或微调错误 (ERRF) 中断
1：使能同步或微调错误 (ERRF) 中断

位 1 **SYNCWARNIE**：SYNC 警告中断使能 (SYNC warning interrupt enable)
0：禁止 SYNC 警告 (SYNCWARNF) 中断
1：使能 SYNC 警告 (SYNCWARNF) 中断

位 0 **SYNCOKIE**：SYNC 事件正常中断使能 (SYNC event OK interrupt enable)
0：禁止 SYNC 事件正常 (SYNCOKF) 中断
1：使能 SYNC 事件正常 (SYNCOKF) 中断

40.6.2 CRS 配置寄存器 (CRS_CFGR)

CRS configuration register

只有禁止频率误差计数器时，才可写入该寄存器（CRS_CR 中的 CEN 位清零）。当计数器使能时，该寄存器被写保护。

偏移地址：0x04

复位值：0x2022 BB7F

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

位 31 **SYNCPOL**：SYNC 极性选择 (SYNC polarity selection)
此位由软件置 1 和清零，以选择 SYNC 信号源的输入极性。
0：SYNC 在上升沿有效（默认）
1：SYNC 在下降沿有效

位 30 保留，必须保持复位值。

位 27 保留，必须保持复位值。

位 26:24 **SYNCDIV[2:0]**：SYNC 分频器 (SYNC divider)
这些位由软件置 1 和清零，以控制 SYNC 信号的分频系数。
000：SYNC 不进行分频（默认）
001：SYNC 2 分频
010：SYNC 4 分频
011：SYNC 8 分频
100：SYNC 16 分频
101：SYNC 32 分频
110：SYNC 64 分频
111：SYNC 128 分频

位 23:16 **FELIM[7:0]**：频率误差限值 (Frequency error limit)

FELIM 包含用于评估 CRS_ISR 寄存器的 FECAP[15:0] 位中锁存的已捕捉频率误差值的值。
有关 FECAP 评估的更多详细信息，请参见*第 40.3.4 节：频率误差评估和自动微调*。

位 15:0 **RELOAD[15:0]**：计数器重载值 (Counter reload value)

RELOAD 是每次发生 SYNC 事件时要装载到频率误差计数器中的值。
有关计数器行为的更多详细信息，请参见*第 40.3.3 节：频率误差测量*。

40.6.3 CRS 中断和状态寄存器 (CRS_ISR)

CRS interrupt and status register

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FECAP[15:0]															
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
FEDIR	Res.	Res.	Res.	Res.	TRIM OVF	SYNC MISS	SYNC ERR	Res.	Res.	Res.	Res.	ESYNCF	ERRF	SYNC WARNF	SYNC OKF
r					r	r	r					r	r	r	r

位 31:16 **FECAP[15:0]**：频率误差捕捉 (Frequency error capture)

FECAP 是最后一次 SYNC 事件发生时锁存的频率误差计数器值。
有关 FECAP 使用的更多详细信息，请参见*第 40.3.4 节：频率误差评估和自动微调*。

位 15 **FEDIR**：频率误差方向 (Frequency error direction)

FEDIR 是最后一次 SYNC 事件发生时锁存的频率误差计数器的计数方向。它显示实际频率低于目标频率还是高于目标频率。
0：递增计数方向，实际频率高于目标频率。
1：递减计数方向，实际频率低于目标频率。

位 14:11 保留，必须保持复位值。

位 10 **TRIMOVF**：微调上溢或下溢 (Trimming overflow or underflow)

当自动微调尝试使 TRIM 值上溢或下溢时，该标志由硬件置 1。如果 CRS_CR 寄存器中的 ERRIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ERRC 位置 1。
0：未发出微调错误信号
1：已发出微调错误信号

位 9 **SYNCMISS**：SYNC 丢失 (SYNC missed)

当频率误差计数器达到值 FELIM * 128 且未检测到 SYNC 时，该标志由硬件置 1，这意味着 SYNC 脉冲丢失或频率误差过大（内部频率过高）而无法通过调整 TRIM 值来补偿，应采取其他措施。此时，频率误差计数器停止（等待下一个 SYNC），并且会在 CRS_CR 寄存器中的 ERRIE 位置 1 时生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ERRC 位置 1。
0：未发出 SYNC 丢失错误信号
1：已发出 SYNC 丢失错误信号

位 8 **SYNCERR**：SYNC 错误 (SYNC error)

当 SYNC 脉冲在 ESYNC 事件之前达到且所测得的频率误差大于或等于 FELIM * 128 时，该标志由硬件置 1。这意味着频率误差过大（内部频率过低）而无法通过调整 TRIM 值进行补偿，应采取其他措施。如果 CRS_CR 寄存器中的 ERRIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ERRC 位置 1。

0：未发出 SYNC 错误信号

1：已发出 SYNC 错误信号

位 7:4 保留，必须保持复位值。

位 3 **ESYNCF**：预期 SYNC 标志 (Expected SYNC flag)

当频率误差计数器达到零值时，此标志由硬件置 1。如果 CRS_CR 寄存器中的 ESYNCIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ESYNCC 位置 1。

0：未发出预期 SYNC 信号

1：已发出预期 SYNC 信号

位 2 **ERRF**：错误标志 (Error flag)

发生同步或微调错误时，该标志由硬件置 1。它是 TRIMOVF、SYNCMISS 和 SYNCERR 的逻辑或运算的结果。如果 CRS_CR 寄存器中的 ERRIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ERRC 位置 1，这将清零 TRIMOVF、SYNCMISS 和 SYNCERR 位。

0：未发出同步或微调错误信号

1：已发出同步或微调错误信号

位 1 **SYNCWARNF**：SYNC 警告标志 (SYNC warning flag)

当所测得的频率误差大于或等于 FELIM * 3 但小于 FELIM * 128 时，该标志由硬件置 1。这意味着要补偿频率误差，必须通过两步或多步来调整 TRIM 值。如果 CRS_CR 寄存器中的 SYNCWARNIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 SYNCWARNC 位置 1。

0：未发出 SYNC 警告信号

1：已发出 SYNC 警告信号

位 0 **SYNCOKF**：SYNC 事件正常标志 (SYNC event OK flag)

当所测得的频率误差小于 FELIM * 3 时，该标志由硬件置 1。这意味着无需调整 TRIM 值或只需一步微调便足以补偿频率误差。如果 CRS_CR 寄存器中的 SYNCOKIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 SYNCOKC 位置 1。

0：未发出 SYNC 事件正常信号

1：已发出 SYNC 事件正常信号

40.6.4 CRS 中断标志清零寄存器 (CRS_ICR)

CRS interrupt flag clear register

偏移地址：0x0C

复位值：0x0000 0000

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

位 31:4 保留，必须保持复位值。

位 3 **ESYNCC**：预期 SYNC 清零标志 (Expected SYNC clear flag)
将 1 写入此位时，CRS_ISR 寄存器中 ESYNCF 标志将清零。

位 2 **ERRC**：错误清零标志 (Error clear flag)
将 1 写入此位将清零 TRIMOVF、SYNCMISS 和 SYNCERR 位，进而会清零 CRS_ISR 寄存器中的 ERRF 标志。

位 1 **SYNCWARNC**：SYNC 警告清零标志 (SYNC warning clear flag)
将 1 写入此位时，CRS_ISR 寄存器中 SYNCWARNF 标志将清零。

位 0 **SYNCOKC**：SYNC 事件正常清零标志 (SYNC event OK clear flag)
将 1 写入此位时，CRS_ISR 寄存器中 SYNCOKF 标志将清零。

40.6.5 CRS 寄存器映射

表 237. CRS 寄存器映射和复位值

偏移	寄存器名	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
0x00	**CRS_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM[5:0]						SWSYNC	AUTOTRIMEN	CEN	Res.	ESYNCIE	ERRIE	SYNCWARNIE	SYNCOKIE
	Reset value																			1	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**CRS_CFGR**	SYNCPOL	Res.	SYNC SRC [1:0]		Res.	SYNC DIV [2:0]			FELIM[7:0]								RELOAD[15:0]															
	Reset value	0		1	0		0	0	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1	1	1
0x08	**CRS_ISR**	FECAP[15:0]																FEDIR	Res.	Res.	Res.	Res.	TRIMOVF	SYNCMISS	SYNCERR	Res.	Res.	Res.	Res.	ESYNCF	ERRF	SYNCWARNF	SYNCOKF
	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
0x0C	**CRS_ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ESYNCC	ERRC	SYNCWARNC	SYNCOKC
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 63 页的第 2.2 节*。

41 调试支持 (DBG)

41.1 简介

为支持软件开发和系统集成，提供了以下一整套调试功能：

- 针对系统中每个 CPU 内核的独立断点调试
- 代码执行跟踪
- 软件指令
- 交叉触发
- 调试功能可以通过 JTAG/串行线调试访问端口，使用行业标准调试工具进行控制。跟踪端口允许捕获数据以进行记录和分析。

调试功能基于 Arm® Coresight™ 组件。

- 通用特性：
  - SWJ-DP：JTAG/串行线调试端口
  - AHB-AP：AHB 访问端口
- CPU2 调试功能（禁止 CPU2 调试，调试器无法访问功能）：
  - ROM 表
  - 系统控制空间 (SCS)
  - 断点单元 (BPU)
  - 数据观察点和跟踪单元 (DWT)
  - 交叉触发接口 (CTI)
- CPU1 调试功能
  - ROM 表
  - 系统控制空间 (SCS)
  - 断点单元 (FPB)
  - 数据观察点和跟踪单元 (DWT)
  - 指令跟踪宏单元 (ITM)
  - 嵌入式跟踪宏单元 (ETM)
  - 交叉触发接口 (CTI)
  - 跟踪端口接口单元 (TPU)

调试器可通过 CPU2 AHB-AP 及其相关的 AHB 总线访问 CPU2 调试功能。

调试器可通过 CPU1 AHB-AP 访问 CPU1 调试功能。

更多相关信息，请参见*第 41.20 节*中引用的 Arm® 文档。

41.2 调试场合

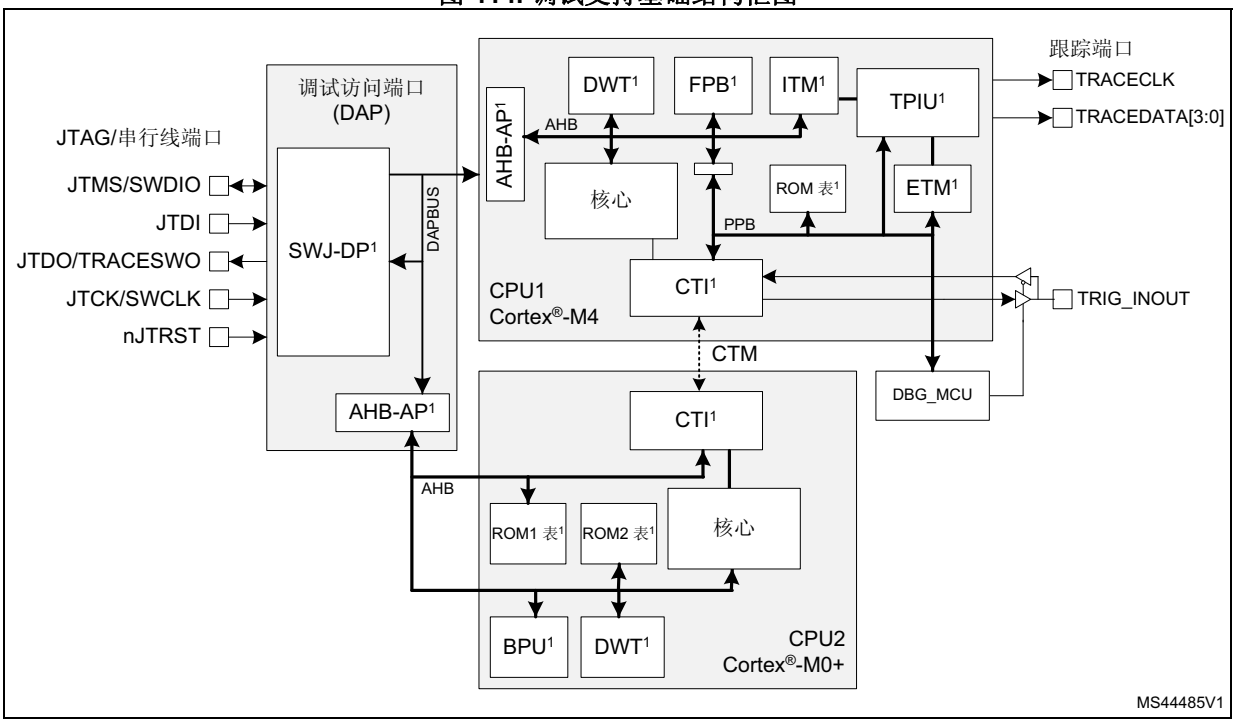
跟踪和调试系统可被应用在多种场合：

- 低成本跟踪
  通过单线调试输出提供有限跟踪功能，可以支持使用“printf”进行代码指令、跟踪数据和地址观察点、中断检测以及程序计数器采样。即使在关闭一个或两个处理器或者其时钟信号时，也可以保持单线跟踪功能。
- 针对每个内核进行独立断点调试
  可使用连接到 JTAG/SWD 调试端口的设备对两个处理器内核同时进行独立调试。这样便可支持断点和观察点设置、逐步执行代码以及存储器访问等。
- 同步调试两个内核
  当其中一个内核因断点或调试器停止命令而停止时，另一个内核也可停止。类似地，两个内核可同时重启。这样，用户便可调试松耦合应用，因为这类应用要求两个处理器保持同步。
- 通过跟踪端口跟踪代码执行
  来自 CPU1 (Cortex®-M4) 的跟踪信息合并为单个跟踪流并实时发送到跟踪端口分析器。跟踪中嵌入的 ID 可使分析器识别每个信息包的来源。

41.3 DBG 功能说明

41.3.1 DBG 框图

图 414. 调试支持基础结构框图



1. Arm® CoreSight™ 组件

41.3.2 DBG 引脚和内部信号

表 238. JTAG/串行线调试端口引脚

引脚名称	JTAG 调试端口		SW 调试端口		引脚分配
	类型	说明	类型	说明	
JTMS/SWDIO	I	JTAG 测试模式选择	IO	串行线数据输入/输出	PA13
JTCK/SWCLK	I	JTAG 测试时钟	I	串行线时钟	PA14
JTDI	I	JTAG 测试数据输入	-	-	PA15
JTDO/TRACESWO	O	JTAG 测试数据输出	-	-	PB3
nJTRST	I	JTAG 测试复位	-	-	PB4

表 239. 跟踪端口引脚

引脚名称	类型	说明	引脚分配
TRACED0	O	跟踪同步数据输出 0	请参见数据手册
TRACED1		跟踪同步数据输出 1	
TRACED2		跟踪同步数据输出 2	
TRACED3		跟踪同步数据输出 3	
TRACECK		跟踪时钟	

表 240. 单线跟踪端口引脚

引脚名称	类型	说明	引脚分配
TRACESWO	O	单线跟踪同步数据输出	PB3(1)

1. TRACESWO 与 JTDO 多路复用。这意味着单线跟踪仅在使用串行线调试接口时可用，而不是在使用 JTAG 时可用。

表 241. 触发引脚

引脚名称	类型	说明	引脚分配
TRIG_INOUT	IO	外部触发输入和输出(1)	请参见数据手册

1. 可通过 DBGMCU 中的 TRGOEN 位将 TRIG_INOUT 配置为输入或输出。

41.4 DBG 功能说明

41.4.1 DBG 电源域

调试组件位于内核电源域中。这意味着，在关断或待机低功耗模式下无法连接调试器。为避免在器件进入待机模式时失去连接，可通过将 DBGMCU 中的相应位置 1 来保持为内核供电。这也将使处理器时钟保持活动状态，并推迟复位，从而维持调试会话。

41.4.2 DBG 时钟

调试器通过调试接口引脚 JTCK/SWCLK 提供调试端口时钟。该时钟用于在串行线和 JTAG 模式下寄存串行输入数据，同时操作状态机和调试端口的内部逻辑。因此，为确保调试端口返回到空闲状态，在访问结束之后，该时钟还必须持续几个周期。

SWJ-DP 包含一个异步接口，用来连接到 DAPCLK 域，该域覆盖 SWJ-DP 的余部和 CPU2 访问端口。

DAPCLK 是通过对系统 HCLK4 进行门控而得到的。

DAPCLK 域可以通过调试器使用调试端口 CTRL/STAT 寄存器中的 CDBGPWRUPREQ 位来使能。必须先使能时钟，然后调试器才能访问器件上的所有调试功能。时钟的可用性由调试端口 CTRL/STAT 寄存器中的 CDBGPWRUPACK 位指示。在上电时、OBL 后和从待机模式唤醒后，将禁止 DAPCLK。在调试器断开连接时也应禁止 DAPCLK，以免浪费能源。

处理器中包含的调试和跟踪组件（ETM ITM、DWG 以及 FPB 等）由相应的内核时钟提供时钟信号。

41.4.3 调试和低功耗模式

STM32WB55xx 器件具有节能特性，允许在不需要的时候关闭或停止内核电源域。如果电源关闭或者内核没有时钟信号，则调试器将无法访问所有调试组件。为了避免这种情况，设计了节能模式仿真这一解决方案。如果对该域使能了仿真功能，则该域仍会进入节能模式，但其时钟和电源保持不变。换句话说，该域的行为就像处于节能模式，但调试器仍然可以对其进行访问。

仿真模式在微控制器调试 (DBGMCU) 单元中编程。有关详细信息，请参见*第 41.8 节：微控制器调试单元 (DBGMCU)*。

41.4.4 DBG 复位

调试端口 (SWJ-DP) 复位条件包括上电复位或 OBL 复位，以及从待机模式唤醒时。

41.4.5 串行线和 JTAG 调试端口 (SWJ-DP)

SWJ-DP 是一个 Coresight™ 组件，提供用来连接调试设备的外部访问端口。

可配置两种类型的接口：

- 一个 5 针标准 JTAG 接口 (JTAG-DP)
- 一个 2 针（时钟 + 数据）“串行线调试”端口 (SW-DP)

这两种模式共用相同的 IO 引脚，因此两者互斥。

默认情况下，系统复位或上电复位后选择 JTAG-DP 模式。五个 IO 引脚由硬件在调试备用功能模式下配置。在 SWJ-DP 模式下，在 JTDI、JTMS/SWDIO 和 nJTRST 上使用了上拉电阻，在 JTCK/SWCLK 上使用了下拉电阻。

调试器可以通过在 JTMS/SWDIO 上传送以下串行数据序列来选择 SW-DP 模式：

... (50 或以上个 1) ..., 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, ... (50 或以上个 1) ...

对于每一数据位都必须循环选通 JTCK/SWCLK。

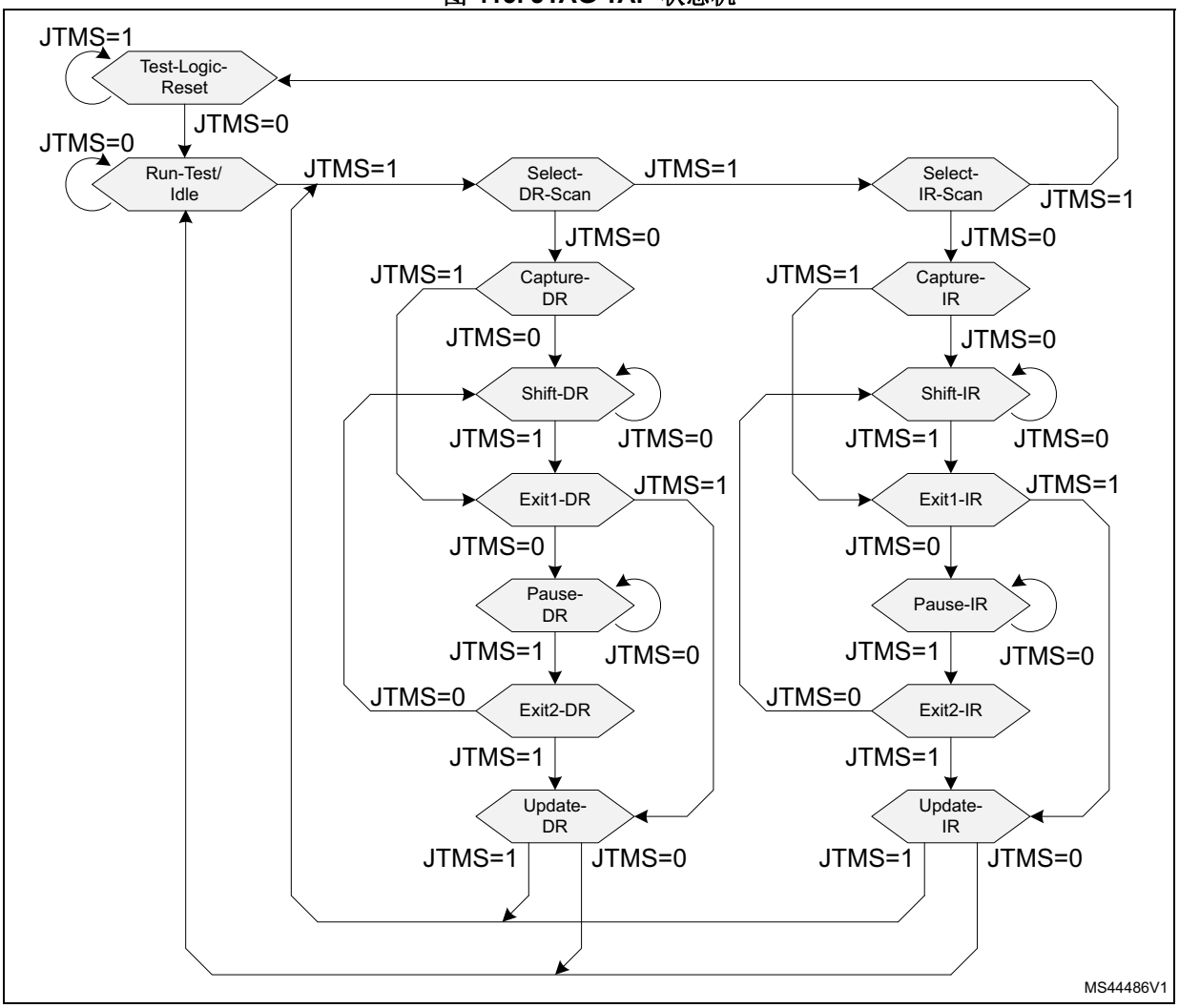
在 SW-DP 模式下，未使用的 JTAG 引脚 JTDI、JTDO 和 nJTRST 可用于其他功能。应注意，所有 SWJ 端口 IO 均可通过软件重新配置为其他功能，但不能再进行调试。

41.4.6 JTAG 调试端口

JTAG 调试端口上有两个 TAP，即 JTAG-DP TAP 和 BSC TAP。

JTAG-DP 设计了一个基于 IEEE 标准 1149.1-1990 的 TAP 状态机 (TAPSM)，如*图 415* 所示。状态机控制两个扫描通道，一个与指令寄存器 (IR) 相关，一个与多个数据寄存器 (DR) 相关。

图 415. JTAG TAP 状态机



JTAG-DP 的工作原理如下：

- 当 TAPSM 通过 Capture-IR 状态时，0b0001 被传送到指令寄存器 (IR) 扫描通道。IR 扫描通道连接在 JTDI 和 JTDO 之间。
- 当 TAPSM 处于 Shift-IR 状态时，IR 扫描通道在 JTCK 的每个上升沿移位一位。这意味，在第一个节拍上：
  - IR 扫描通道的 LSB 是 JTDO 上的输出。
  - IR 扫描通道的位 [n] 被传送到位 [n-1]。
  - JTDI 上的值会传送到 IR 扫描通道的 MSB 中。
- 当 TAPSM 通过 Update-IR 状态时，扫描进 IR 扫描通道中的值被传送到指令寄存器。
- 当 TAPSM 通过 Capture-DR 状态时，值从其中一个数据寄存器传送到其中一个 DR 扫描通道，DR 扫描通道连接在 JTDI 和 JTDO 之间。
- 指令寄存器中保存的值决定了数据寄存器（和相关 DR 扫描通道）的选择。
- 当 TAPSM 处于 Shift-DR 状态时，数据进行移位，与处于 IR 状态时的 IR 移位情形相同。
- 当 TAPSM 通过 Update-DR 状态时，扫描进 DR 扫描通道中的值被传送到所选择的数据寄存器中。
- 当 TAPSM 处于 Run-Test/Idle 状态时，不会发生特殊操作。IDCODE 指令加载在 IR 中。

nJTRST 信号在激活后将状态机异步重置为 Test-Logic-Reset 状态。

表 242 中列出与 4 位 IR 指令对应的数据寄存器。总 IR 指令长度为 9 位。

表 242. JTAG-DP 数据寄存器

IR 指令	DR 寄存器	扫描通道长度	说明
0000 到 0111	(BYPASS)	1	*未设计*：选择 BYPASS
1000	ABORT	35	*ABORT 寄存器* – 位 31:1 = 保留 – 位 0 = APABORT：写入 1 以生成 AP 中止。
1001	(BYPASS)	1	*保留*：选择 BYPASS
1010	DPACC	35	*调试接口寄存器* 初始化调试接口，并允许访问调试接口寄存器。 – 传输 IN 数据时： 位 34:3 = DATA[31:0] = 为写请求传输的 32 位数据 位 2:1 = A[3:2] = 调试端口寄存器的 2 位地址 位 0 = RnW = 读请求 (1) 或写请求 (0) – 传输 OUT 数据时： 位 34:3 = DATA[31:0] = 读请求后读取的 32 位数据 位 2:0 = ACK[2:0] = 3 位确认： 010 = OK/FAULT 001 = WAIT 其他 = 保留

表 242. JTAG-DP 数据寄存器（续）

IR 指令	DR 寄存器	扫描通道长度	说明
1011	APACC	35	*存取接口寄存器* 初始化存取接口并允许访问存取接口寄存器。 – 传输 IN 数据时： 位 34:3 = DATA[31:0] = 为写请求移入的 32 位数据 位 2:1 = A[3:2] = 访问端口寄存器的 2 位子地址 位 0 = RnW = 读请求 (1) 或写请求 (0) – 传输 OUT 数据时： 位 34:3 = DATA[31:0] = 读请求后读取的 32 位数据 位 2:0 = ACK[2:0] = 3 位确认： 010 = OK/FAULT 001 = WAIT 其他 = 保留
1100	(BYPASS)	1	*保留：*选择 BYPASS
1101	(BYPASS)	1	*保留：*选择 BYPASS
1110	IDCODE	32	*ID 代码* 0x6BA0 0477：Arm® JTAG 调试端口 ID 代码
1111	BYPASS	1	*旁路* 在 JTDI 和 JTDO 之间插入单个 JTCK 周期延迟

有关 DR 寄存器的详细说明，请参见《Arm® 调试接口架构规范》*[1]*。

41.4.7 SW 调试端口

串行线调试协议使用两个引脚：

- SWCLK：从主机到目标的时钟
- SWDIO：双向串行数据（需要 100 kΩ 上拉电阻）

串行数据传输与时钟同步，首先传输 LSB。传输过程包括三个阶段：

1. 主机发送的数据包请求（8 位），请参见*表 243*。
2. 目标发送的确认响应（3 位），请参见*表 244*。
3. 主机（写操作时）或目标（读操作时）发送的传输数据（33 位），请参见*表 245*。

只有在确认响应为“OK”时才进行数据传输。

在每一个传输阶段之间，如果数据传输方向发生改变，则插入单个时钟周期的周转时间。

表 243. 数据包请求

位域	名称	说明
0	启动	必须为 1
1	APnDP	– 0：DP 寄存器访问——有关 DP 寄存器的列表，请参见*表 242* – 1：AP 寄存器访问——请参见*第 41.5 节：访问端口*
2	RnW	– 0：写请求 – 1：读请求
4:3	A(3:2)	DP 或 AP 寄存器的地址位域

表 243. 数据包请求（续）

位域	名称	说明
5	奇偶校验	前面几位的单位奇偶校验
6	停止	0
7	驻留	不受主机驱动，必须由目标读为 1

表 244. ACK 响应

位域	名称	说明
2:0	ACK	– 000：FAULT – 010：WAIT – 100：OK

表 245. 数据传输

位域	名称	说明
31:0	WDATA 或 RDATA	写入或读取数据
32	奇偶校验	32 个数据位的单位奇偶校验

如果目标响应为 FAULT 或 WAIT ACK，则数据传输过程将取消，除非使能了上溢检测，在这种情况下，数据会被目标忽略（写操作时）或不接受指令控制（读操作时）。

当引线第一次接通时，必须由主机生成引线复位信号，否则会出现协议错误。引线复位信号由 50 或以上个 SWDIO 为高电平的 SWCLK 周期和紧随的两个 SWDIO 为低电平的 SWCLK 周期组成。

有关串行线调试协议的更多详细信息，请参见《Arm® 调试接口架构规范》*[1]*。

注： *SWJ-DP 适用 SWD 协议版本 2。*

41.4.8 调试端口寄存器

SW-DP 和 JTAG-DP 都可以访问*表 246* 中列出的调试端口 (DP) 寄存器。

调试器可访问 DP 寄存器，如下所示：

- 如果使用 JTAG，则使用存储区中的寄存器地址对 DPACC 寄存器中的 A(3:2) 位域编程。对 RnW 位编程，选择读操作或写操作。如果选择写操作，则使用写入数据对 DATA 位域进行编程。如果使用 SWD，则 A(3:2) 和 RnW 位域包含在发送到 SW-DP 的数据包请求字中，数据包请求字的 APnDP 位复位（请参见*表 243*）。写入数据在数据阶段发送。
- 要访问地址 0x4 处的某个分区 DP 寄存器，必须先将寄存器编号写入地址 0x8 处的 DP_SELECT 寄存器。对地址 0x4 的任何后续读取操作或写入操作都将访问与 DP_SELECT 寄存器的内容相对应的寄存器。

41.4.9 DP 调试端口标识寄存器 (DP_DPIDR)

DP debug port identification register

偏移地址：0x0

复位值：0x5BA0 2477

只读

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REVISION[0:3]				PARTNO[0:7]								Res.	Res.	Res.	MIN
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
VERSION[0:3]				DESIGNER[0:10]											Res.
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

位 31:28 **REVISION**：版本代码 (Revision code)
0x5

位 27:20 **PARTNO**：调试端口的产品编号 (Part number for the debug port)
0xBA

位 19:17 保留，必须保持复位值。

位 16 **MIN**：最少调试端口 (MINDP) 设计 (Minimal debug port (MINDP) implementation)
0x0：未设计 MINDP（支持事务计数器和推入操作）

位 15:12 **VERSION**：DP 架构版本 (DP architecture version)
0x2：DPv2

位 11:1 **DESIGNER**：JEDEC 设计人员身份代码 (JEDEC designer identity code)
0x23B：Arm® JEDEC 代码

位 0 保留，必须保持复位值。

41.4.10 DP 中止寄存器 (DP_ABORTR)

DP abort register

偏移地址：0x0

复位值：0x0000 0000

只写

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ORUNERR CLR	WDERR CLR	STKERR CLR	STKCMP CLR	DAPABORT
											w	r	r	r	

位 31:5 保留，必须保持复位值。

位 4 **ORUNERRCLR**：上溢错误清零 (Overrun error clear)
0：无影响
1：将 CTRL/STAT.STICKYORUN 位清零

位 3 **WDERRCLR**：写入数据错误清零 (Write data error clear)
0：无影响
1：将 CTRL/STAT.WDATAERR 位清零

位 2 **STKERRCLR**：粘滞错误清零 (Sticky error clear)
0：无影响
1：将 CTRL/STAT.STICKYERR 位清零

位 1 **STKCMPCLR**：粘滞比较清零 (Sticky compare clear)
0：无影响
1：将 CTRL/STAT.STICKYCMP 位清零

位 0 **DAPABORT**：在返回的 WAIT 响应数量过多时中止当前 AP 事务，表示该事务被阻塞。
0：无影响
1：中止事务

41.4.11 DP 控制和状态寄存器 (DP_CTRL/STATR)

DP control and status register

偏移地址：0x4，DP_SELECTR.DPBANKSEL = 0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	CDBGPWRUPACK	CDBGPWRUPREQ	Res.	Res.	Res.	Res.	TRNCNT[4:11]							
		r	r					r	r	r	r				r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRNCNT[0:3]				CMASKLANE[0:3]				WDATAERR	READOK	STICKYERR	STICKYCMP	TRNMODE[0:1]		STICKYORUN	ORUNDETECT
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

位 31:30 保留，必须保持复位值。

位 29 **CDBGPWRUPACK**：请参见*第 41.4.2 节：DBG 时钟*中的说明。
0 = 门控 DAPCLK
1 = 使能 DAPCLK

位 28 **CDBGPWRUPREQ**：控制 DAPCLK 使能请求信号。
0 = 请求 DAPCLK 门控
1 = 请求 DAPCLK 使能

位 27:26 保留，必须保持复位值。

位 25:24 保留，必须保持复位值。

位 23:12 **TRNCNT**：事务计数器 (Transaction counter)。若要通过 AP 将事务序列编程为增量地址，TRNCNT 需载有要执行的事务数。每个事务成功完成时，事务数会递减。

位 11:8 **MASKLANE**：指示推入比较和推入验证操作（CTRL/STAT.TRNMODE = 1 或 2）中要屏蔽的字节。在推入操作中，在 AP 写入事务中提供的字与目标 AP 地址中的当前值进行比较。

0b1XXX = 比较中包含字节通道 3
0bX1XX = 比较中包含字节通道 2
0bXX1X = 比较中包含字节通道 1
0bXXX1 = 比较中包含字节通道 0

位 7 **WDATAERR**：SW-DP 模式下的写入数据错误（只读）(Write data error (read only) in SW-DP)。指示：

– 写入操作的数据阶段存在奇偶校验错误或成帧错误，或
– 已被 DP 接受的写入操作被丢弃，而不是被提交给 AP

向 ABORT.WDERRCLR 位写入 1 即可将此位复位。

0：无错误
1：发生错误

在 JTAG-DP 中为保留位。

位 6 **READOK**：SW-DP 模式下的 AP 读取响应（只读）(AP read response (read only) in SW-DP)。指示对最后一次 AP 读访问的响应。

0：读操作不正常
1：读操作正常

在 JTAG-DP 中为保留位。

位 5 **STICKYERR**：事务错误 (Transaction error)（在 SW-DP 模式下，此位只读；在 JTAG-DP 模式下，此位可读/写）。指示在 AP 事务期间发生了错误。

0：无错误
1：发生错误

在 SW-DP 模式下，通过向 ABORT.STKERRCLR 位写入 1 可将此位复位。在 JTAG-DP 模式下，通过向此位写入 1 可将此位复位。

位 4 **STICKYCMP**：比较匹配 (Compares match)（在 SW-DP 模式下，此位只读；在 JTAG-DP 模式下，此位可读/写）。指示在推入操作中发生了匹配。

0：匹配（当 TRNMODE = 0x1 时）；不匹配（当 TRNMODE = 0x2 时）
1：不匹配（当 TRNMODE = 0x1 时）；匹配（当 TRNMODE = 0x2 时）

在 SW-DP 模式下，通过向 ABORT.STKCMPCLR 位写入 1 可将此位复位。在 JTAG-DP 模式下，通过向此位写入 1 可将此位复位。

位 3:2 **TRNMODE**：AP 写操作传输模式 (Transfer mode for AP write operations)（对于读操作，此位域必须设为 0x0）。

0x0：正常工作。AP 事务直接传递到 AP。
0x1：推入验证操作。DP 存储写入数据并执行目标 AP 地址的读取事务。将读取的结果与存储的数据进行比较，如果二者不匹配，则 STICKYCMP 位置 1。
0x2：推入比较操作。DP 存储写入数据并执行目标 AP 地址的读取事务。将读取的结果与存储的数据进行比较，如果二者匹配，则 STICKYCMP 位置 1。
0x3：保留。

在推入操作中，只将 MASKLANE 位域表示的数据字节进行比较。

位 1 **STICKYORUN**：上溢 (Overrun)（在 SW-DP 模式下，此位只读；在 JTAG-DP 模式下，此位可读/写）。指示发生了上溢（在完成上一个事务之前接收到新的事务）。此位仅在 ORUNDETECT 位置 1 时才置 1。

0：无上溢

1：发生上溢

在 SW-DP 模式下，通过向 ABORT.ORUNERRCLR 位写入 1 可将此位复位。在 JTAG-DP 模式下，通过向此位写入 1 可将此位复位。

位 0 **ORUNDETECT**：上溢检测模式使能 (Overrun detection mode enable)

0：禁止上溢检测。

1：使能上溢检测。如果发生上溢，则 STICKYORUN 位将置 1，同时后续事务将被阻止，直到 STICKYORUN 位清零。

41.4.12 DP 数据链路控制寄存器 (DP_DLCR)

DP data link control register

偏移地址：0x4，DP_SELECTR.DPBANKSEL = 1

复位值：0x0000 0040

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

位 31:10 保留，必须保持复位值。

位 9:8 **TURNROUND**：SWDIO 的三态周期 (Tristate period for SWDIO)

0x0：1 个数据位周期

0x1：2 个数据位周期

0x2：3 个数据位周期

0x3：4 个数据位周期

位 7 保留，必须保持复位值。

位 6 保留，必须保持复位值。（设为 1）

位 5:0 保留，必须保持复位值。

41.4.13 DP 目标标识寄存器 (DP_TARGETIDR)

DP target identification register

偏移地址：0x4，DP_SELECTR.DPBANKSEL = 2

复位值：0x0495 0041

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TREVISION[0:3]				TPARTNO[0:11]											
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
TPARTNO[12:15]				TDESIGNER[0:10]											Res.
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

位 31:28 **TREVISION**：目标版本 (Target revision)
0x0：版本 1

位 27:12 **TPARTNO**：目标产品编号 (Target part number)
0x4950：STM32WB55xx

位 11:1 **TDESIGNER**：目标设计人员 JEDEC 代码 (Target designer JEDEC code)
0x020：STMicroelectronics

位 0 保留，必须保持复位值（设为 1）

41.4.14 DP 数据链路协议标识寄存器 (DP_DLPIDR)

DP data link protocol identification register

偏移地址：0x4，DP_SELECTR.DPBANKSEL = 3

复位值：0x0000 0001

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

位 31:28 **TINSTANCE**：目标实例编号 (Target instance number)。用于在多点系统中定义此器件的实例编号。
0x0：实例编号 0

位 27:4 保留，必须保持复位值。

位 3:0 **PROTSVN**：串行线调试协议版本 (Serial Wire Debug protocol version)
0x1：版本 2

41.4.15 DP 重新发送寄存器 (DP_RESENDR)

DP resend register (DP_RESENDR)

偏移地址：0x8

复位值：0x0000 0000

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

位 31:0 **RESEND**：返回由最后一个 AP 读操作或 DP RDBUFF 读操作返回的值。在读传输损坏的情况下使用。

41.4.16 DP 访问端口选择寄存器 (DP_SELECTR)

DP access port select register

偏移地址：0x8

复位值：未知

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

位 31:28 **APSEL**：访问端口选择 (Access port select)。选择下一个事务的访问端口。
0x0：AP0 - CPU1 (Cortex®-M4) 调试访问端口 (AHB-AP)
0x1：AP1 - CPU2 (Cortex®-M0+) 调试访问端口 (AHB-AP)
0x2 到 0xF：保留

位 27:8 保留，必须保持复位值。

位 7:4 **APBANKSEL**：AP 寄存器存储区选择 (AP register bank select)。选择活动 AP 上的 4 字寄存器存储区，以用于下一个事务。

位 3:0 **DPBANKSEL**：DP 寄存器存储区选择 (DP register bank select)。选择调试端口地址为 0x4 的寄存器。
0x0：CTRL/STAT 寄存器
0x1：DLCR 寄存器
0x2：TARGETID 寄存器
0x3：DLPIDR 寄存器
0x4 到 0xF：保留

41.4.17 DP 读缓冲区寄存器 (DP_BUFFR)

DP read buffer register

偏移地址：0xC

复位值：0x0000 0000

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

位 31:0 **RDBUFF**：包含由最后一个 AP 读访问返回的值。AP 读访问返回的值可通过对同一地址进行第二次读访问来获得，这将在相应的总线上启动新的事务，或者还可以从该寄存器读取，在这种情况下不会发生新的 AP 事务。

41.4.18 DP 目标标识寄存器 (DP_TARGETSELR)

DP target identification register

偏移地址：0xC

复位值：未知

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

位 31:28 **TINSTANCE**：目标实例编号 (Target instance number)。用于在多点系统中定义目标器件的实例编号。必须将用于 DLPIDR.TINSTANCE 的值写入这些位才能选择此器件。

位 27:12 **TPARTNO**：目标产品编号 (Target part number)。定义目标器件的产品编号。必须将用于 TARGETID.TPARTNO 的值写入这些位才能选择此器件。

位 11:1 **TDESIGNER**：目标设计人员 JEDEC 代码 (Target designer JEDEC code)。定义目标器件的 JEDEC 代码。必须将用于 TARGETID.TDESIGNER 的值写入这些位才能选择此器件。

位 0 保留，必须保持复位值（设为 1）。

41.4.19 调试端口寄存器映射和复位值

这些寄存器未在 CPU 存储器总线上，只能通过 SW-DP 和 JTAG-DP 调试接口访问。

调试端口地址为 2 位宽，在 JTAG-DP 寄存器 DPACC 或 SW-DP 数据包请求 A[3:2] 位域中定义。

表 246. 调试端口寄存器映射和复位值

偏移	寄存器名称	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
0b00	DP_DPIDR	REVISION [0:3]				PARTNO[0:7]								Res.	Res.	Res.	MIN	VERSION [0:3]				DESIGNER[0:10]											Res.
	Reset value	0	1	0	1	1	0	1	1	1	0	1	0				0	0	0	1	0	0	1	0	0	0	1	1	1	0	1	1	1
0b00	DP_ABORTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ORUNERRCLR	WDERRCLR	STKERRCLR	STKCMPCLR	DAPABORT
	Reset value																												0	0	0	0	0
0b01(1)	DP_CTRL/STATR	Res.	Res.	CDBGPWRUPACK	CDBGPWRUPREQ	Res.	Res.	Res.	Res.	TRNCNT[0:11]												APPSET[3:0]				WDATAERR	READOK	STICKYERR	STICKYCMP	TRNMODE[0:1]		STICKYORUN	ORUNDETECT
	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
0b01(1)	DP_DLCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TURNROUND [0:1]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																							0	0								
0b01(2)	DP_TARGETIDR	TREVISION [0:3]				TPARTNO[0:15]																TDESIGNER[0:10]											Res.
	Reset value	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
0b01(3)	DP_DLPIDR	TINSTANCE [0:3]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PROTSVN [0:3]			
	Reset value	0	0	0	0																									0	0	0	1
0b10	DP_RESENDR	RESEND[0:31]																															
	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
0b10	DP_SELECTR	APSEL[0:3]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APBANKSEL [0:3]				DPBANKSEL [0:3]			
	Reset value	x	x	x	x																					x	x	x	x	x	x	x	x
0b11	DP_BUFFR	RDBUFF[0:31]																															
	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
0b11	DP_TERGETSELR	TINSTANCE [0:3]				TPARTNO[0:15]																TDESIGNER[0:10]											Res.
	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	

1. DP_SELECTR.DPBANKSEL = 1。
2. DP_SELECTR.DPBANKSEL = 2。
3. DP_SELECTR.DPBANKSEL = 3。

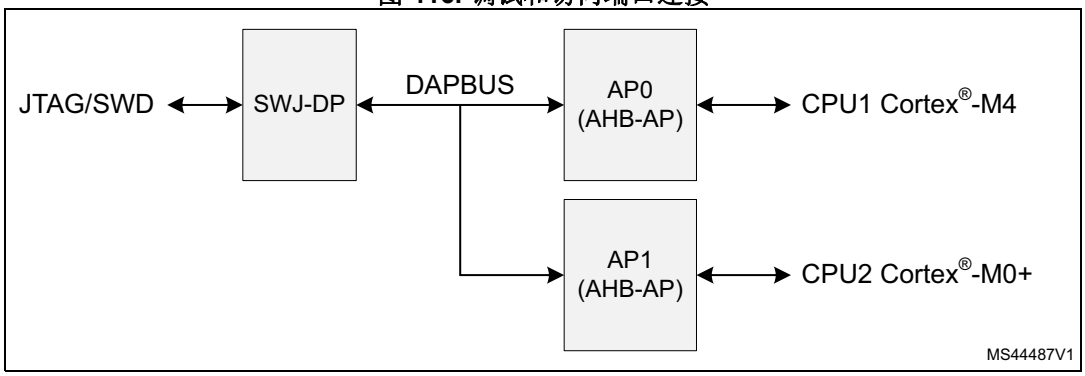
41.5 访问端口

如*图 416* 所示，有两个访问端口 (AP) 连接到 DP：

1. AP0：CPU1 (Cortex®-M4) 访问端口 (AHB-AP)：允许通过其内部 AHB 总线访问 Cortex®-M4 处理器内核中集成的调试和跟踪功能。
2. AP1：CPU2 (Cortex®-M0+) 访问端口 (AHB-AP)：允许通过其内部 AHB 总线访问 Cortex®-M0+ 处理器内核中集成的调试和跟踪功能。

两个访问端口都是 MEM-AP 类型，即调试和跟踪组件寄存器都映射在相关调试总线的地址空间中。调试器将 AP 视为一组 32 位寄存器，每个存储区有四个寄存器。其中一些寄存器用于配置或监视 AP 本身，而其他寄存器则用于执行总线传输。*表 247：访问端口寄存器映射和复位值*中列出了 AP 寄存器。

图 416. 调试和访问端口连接



AP 寄存器的地址包括：

- 位 [7:4]：DP 中 SELECT 寄存器 APBANKSEL 位域的内容（请参见*第 41.4.16 节：DP 访问端口选择寄存器 (DP_SELECTR)*）
- 位 [3:2]：JTAG-DP 模式下 APACC 数据寄存器（请参见*表 246：调试端口寄存器映射和复位值*）或 SW-DP 数据包请求（请参见*表 243：数据包请求*）中的 A(3:2) 位域的内容，具体取决于使用的调试接口
- 位 [1:0]：始终设置为 0

DP 中 SELECT 寄存器 APSEL 位域的内容定义访问的 MEM-AP。

调试器可访问 AP 寄存器，如下所示：

1. 对 DP 中的 SELECT 寄存器 APSEL 位域编程，以选择其中一个 AP；对 APBANKSEL 位域编程，以选择要访问的寄存器存储区（请参见*第 41.4.16 节：DP 访问端口选择寄存器 (DP_SELECTR)*）。
2. 如果使用 JTAG，则使用存储区中的寄存器地址对 APACC 寄存器中的 A(3:2) 位域编程。对 RnW 位编程，选择读操作或写操作。如果选择写操作，则使用写入数据对 DATA 位域进行编程。如果使用 SWD，则 A(3:2) 和 RnW 位域包含在发送到 SW-DP 的数据包请求字当中，数据包请求字的 APnDP 位置 1（请参见*表 243：数据包请求*）。写入数据在数据阶段发送。

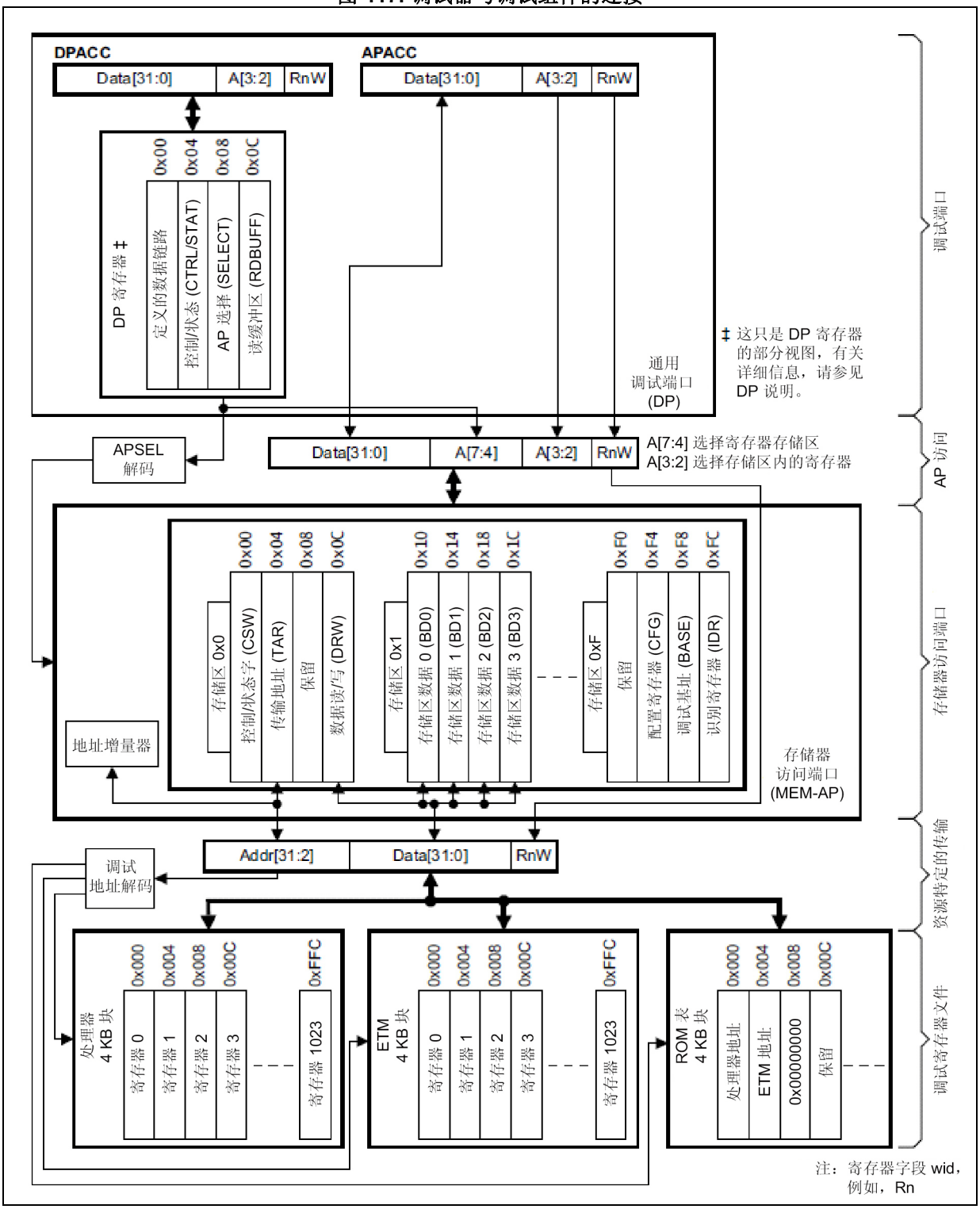
调试器可通过 MEM-AP 寄存器访问存储器映射的调试组件寄存器（即，使用上述 AP 寄存器访问过程），如下所示：

1. 对 TAR 寄存器中的事务目标地址编程。
2. 对 CSW 寄存器编程，必要时，请使用传输参数（如 AddrInc）。
3. 针对 TAR 寄存器中保存的地址，写入或读取 DRW 寄存器，以启动总线事务。或者，读取或写入分区数据寄存器 BDn，以触发访问地址 TAR[31:4] + n（支持无需更改 TAR 寄存器中的地址即可访问多达四个连续的地址）。

图 417 显示了如何使用 MEM-AP 将调试端口连接到调试组件（例如处理器、ETM 或 ROM 表）。

有关 MEM-AP 的更多详细信息，请参见《Arm® 调试接口架构规范》*[1]*。

图 417. 调试器与调试组件的连接



41.5.1 AP 控制/状态字寄存器 (AP_CSWR)

AP control/status word register

偏移地址：0x0

复位值：0x2300 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	SPROT	Res.	PROT[0:4]					SPISTATUS	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	r		r												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	MODE[0:3]				TRINPROG	DEVICEEN	ADDRIN[0:1]		Res.	SIZE[0:2]		
								r	r	r	r		r	r	r

位 31 保留，必须保持复位值。

位 30 **SPROT**：安全传输请求 (Secure transfer request)。在 AHB-AP 中，此位域为总线传输设置保护属性 HPROT[6]。

0：如果 SPIDEN 为高电平，加密传输。如果 SPIDEN 为低电平，非加密传输。

1：非加密传输。

位 29 保留，必须保持复位值。

位 28:24 **PROT**：总线传输保护 (Bus transfer protection)。在 AHB-AP 中，此位域为总线传输设置保护属性 HPROT[4:0]。

0bXXXX0：指令读取

0bXXXX1：数据访问

0bXXX0X：用户模式

0bXXX1X：特权模式

0bXX0XX：不可缓冲

0bXX1XX：可缓冲

0bX0XXX：不可缓存

0bX1XXX：可缓存

0b0XXXX：非独占

0b1XXXX：独占

位 23 **SPISTATUS**：SPIDEN 状态选项位 (Status of SPIDEN option bit)（只读）。该信号确定了调试器是否可访问安全存储器。

0：禁止加密 AHB 传输

1：允许加密 AHB 传输

位 22:12 保留，必须保持复位值。

位 11:8 **MODE**：屏障支持使能 (Barrier support enabled)。定义是否支持存储器屏障操作。

0x0：不支持

位 7 **TRINPROG**：正在进行传输 (Transfer in progress)（只读）。指示是否正在 AP 上进行总线传输。

0x0：无正在进行的传输。

0x1：正在进行总线传输。

位 6 **DEVICEEN**：器件使能 (Device Enabled)（只读）。定义了是否可访问 AP。

0x1：使能 AP 访问。

位 5:4 **ADDRINC**：自动递增模式 (Auto-increment mode)。定义了在事务结束之后 TAR 地址是否会自动递增。

0x0：不自动递增。

0x1：地址按事务字节（SIZE 位域）的大小递增。

0x2：使能打包传输。32 位 AP 访问将产生与编程的事务大小相对应的 1 个 32 位、2 个 16 位或 4 个 8 位总线事务。数据将相应地被打包或解包。

0x3：保留。

位 3 保留，必须保持复位值。

位 2:0 **SIZE**：下一个存储器访问事务的大小 (Size of next memory access transaction)

0x0：字节（8 位）

0x1：半字（16 位）

0x2：字（32 位）

0x3-0x7：保留

41.5.2 AP 传输地址寄存器 (AP_TAR)

AP transfer address register

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

位 31:0 **TA**：当前传输的地址 (Address of current transfer)

41.5.3 AP 数据读/写寄存器 (AP_DRWR)

AP data read/write register

偏移地址：0x0C

复位值：0x0000 0000

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

位 31:0 **TD**：当前传输的数据 (Data of current transfer)

41.5.4 AP 分区数据寄存器 (AP_BD0-3R)

AP banked data registers

偏移地址：0x10 (DB0R)

偏移地址：0x14 (BD1R)

偏移地址：0x18 (BD2R)

偏移地址：0x1C (BD3R)

复位值：0x0000 0000

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

位 31:0 **TBD**：到地址 TAR 的当前传输的分区数据 (Banked data of current transfer to address TAR)。TA+ AP_BDnR 地址 [3:2] + 0b00。不在 AP_BD0-3R 上执行自动地址递增。仅字传输支持分区传输。

41.5.5 AP 基址寄存器 (AP_BASER)

AP base address register

偏移地址：0xF8

复位值：0xE00F F003 (AP0)、0xF000 0003 (AP1)

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

位 31:12 **BASEADDR**：AP ROM 表基址（位 31 到 12）(Base address (bits 31 to 12) of ROM table for the AP)。由于 ROM 表必须与 4 KB 边界对齐，因此 12 个 LSB 均为零。

AP0 CPU1 (Cortex®-M4) AHB-AP：0xE00FF

AP1 CPU2 (Cortex®-M0+) AHB-AP：0xF0000

位 11:2 保留，必须保持复位值。

位 1 **FORMAT**：基址寄存器格式 (Base address register format)

1：Arm® 调试接口 v5。

位 0 **ENTRYPRESENT**：指示访问端口总线上有无调试组件。

1：有调试组件

41.5.6 AP 标识寄存器 (AP_IDR)

AP identification register

偏移地址：0xFC

复位值：0x2477 0011 (AP0)、0x6477 0001 (AP1)

只读

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

位 31:28 **REVISION**：
0x2：r0p3
0x6：r0p7

位 27:24 **JEDECBANK**：JEDEC 存储区 (JEDEC bank)
0x4：Arm®

位 23:17 **JEDECCODE**：JEDEC 代码 (JEDEC code)
0x3B：Arm®

位 16 **MEMAP**：存储器访问端口 (Memory access port)
0x1：标准寄存器映射

位 15:8 保留，必须保持复位值。

位 7:0 **IDENTITY**：标识 AP 的类型。
0x11：CPU1 (Cortex®-M4) AHB-AP (AP0)
0x01：CPU2 (Cortex®-M0+) AHB-AP (AP1)

41.5.7 访问端口寄存器映射和复位值

这些寄存器未在 CPU 存储器总线上，只能通过 SW-DP 和 JTAG-DP 调试接口访问。

访问端口地址为 8 位宽，由调试端口寄存器 DP_SELECTR.APBANKSEL[3:0] 位域和 JTAG-DP 寄存器 DPACC 或 SW-DP 数据包请求 A[3:2] 位域定义。

表 247. 访问端口寄存器映射和复位值

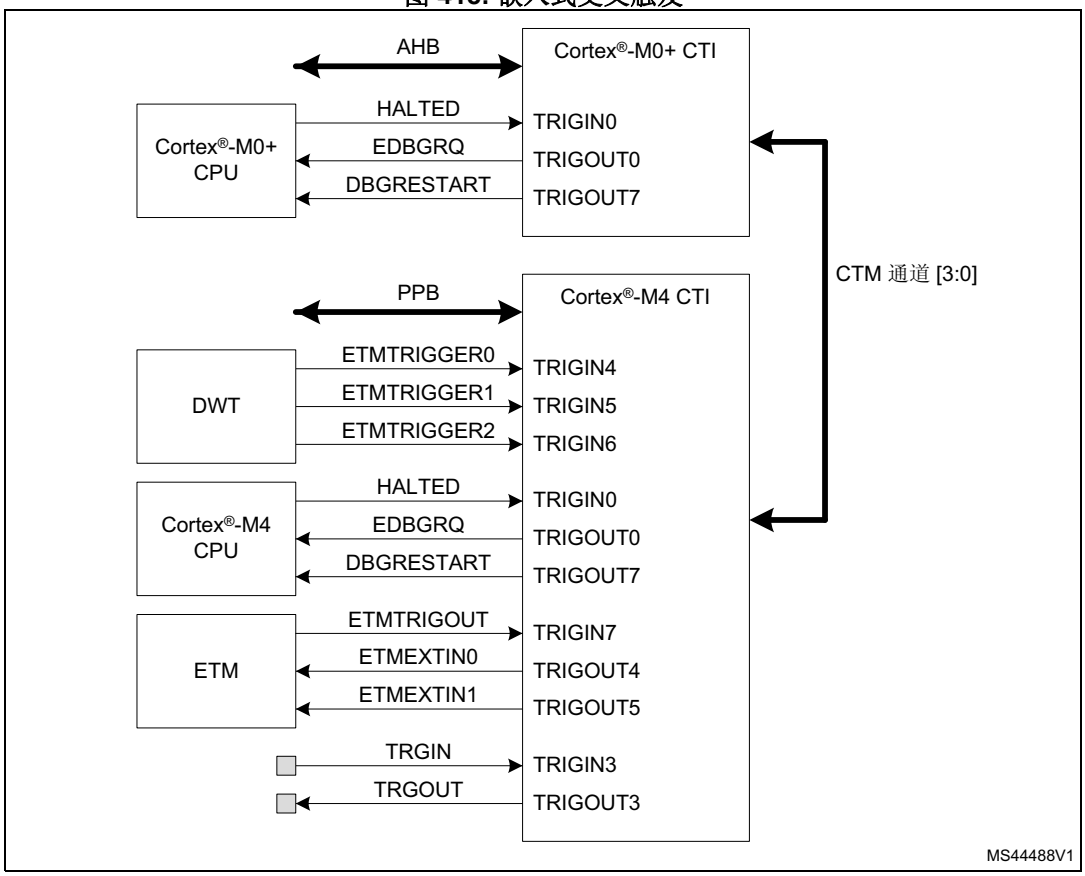
偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	AP_CSWR	Res.	SPROT	Res.	PROT[0:4]					SPISTATUS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[0:3]				TRINPROG [0:1]		ADDRIN [0:1]		Res.	SIZE [0:2]		
	Reset value		0		0	0	0	1	1	0												0	0	0	0	0	1	0	0		0	0	0
0x04	AP_TAR	TA[0:31]																															
	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
0x0C	AP_DRWR	TD[0:31]																															
	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
0x10	AP_BD0R	TBD[0:31]																															
	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
0x14	AP_BD1R	TBD[0:31]																															
	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
0x18	AP_BD2R	TBD[0:31]																															
	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
0x1C	AP_BD3R	TBD[0:31]																															
	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
0xF8	AP_BASER	BASEADDR[0:19]																													FORMAT	ENTRYPRESENT	
	Reset value (AP0)	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1
	Reset value (AP1)	1	1	1	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	0	1	1
0xFC	AP_IDR	REVilSION [0:3]				JEDECBANK [0:3]				JEDECCODE[0:6]							MEMAP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDENTITY[0:7]							
	Reset value (AP0)	0	0	1	0	0	1	0	0	0	1	1	1	0	1	1	1									0	0	0	1	0	0	0	1
	Reset value (AP1)	0	1	1	0	0	1	0	0	0	1	1	1	0	1	1	1									0	0	0	0	0	0	0	1

41.6 交叉触发接口 (CTI) 和矩阵 (CTM)

交叉触发接口 (CTI) 和交叉触发矩阵 (CTM) 一起形成了 CoreSight™ 嵌入式交叉触发功能（请参见*图 418*）。

有两个 CTI 组件，一个专用于 CPU2，另一个专用于 CPU1。CTI 通过 CTM 相互连接。调试器可通过相应的访问端口和相关的 AHB 访问 CTI 寄存器。

图 418. 嵌入式交叉触发



CTI 允许来自各种源的事件触发调试和/或跟踪活动。例如，在其中一个处理器内核中到达的断点可以停止另一个处理器，或者在外部触发输入上检测到的转换可以启动代码跟踪。

每个 CTI 最多拥有 8 个触发输入和 8 个触发输出。任何输入都可以连接到同一个 CTI 的任何输出，也可通过 CTM 连接到另一个 CTI 上的任何输出。

表 *248* 到 *251* 中列出了各个 CTI 的触发输入和输出信号。

表 248. CPU2 CTI 输入

编号	源信号	源组件	注释
0	HALTED	CPU2	CPU2 停止——指示 CPU2 处于调试模式
1	-	-	未使用
2	-	-	未使用

表 248. CPU2 CTI 输入（续）

编号	源信号	源组件	注释
3	-	-	未使用
4	-	-	未使用
5	-	-	未使用
6	-	-	未使用
7	-	-	未使用

表 249. CPU2 CTI 输出

编号	输出信号	目标组件	注释
0	EDBGRQ	CPU2	CPU2 停止请求——将 CPU2 置于调试模式
1	-	-	未使用
2	-	-	未使用
3	-	-	未使用
4	-	-	未使用
5	-	-	未使用
6	-	-	未使用
7	DBGRESTART	CPU2	CPU2 重启请求——CPU2 退出调试模式

表 250. CPU1 CTI 输入

编号	源信号	源组件	注释
0	HALTED	CPU1	CPU1 停止——指示 CPU1 处于调试模式
1	-	-	未使用
2	-	-	未使用
3	-	-	未使用
4	ETMTRIGGER0	CPU1 DWT	跟踪触发——使能 CPU1 执行跟踪
5	ETMTRIGGER1	CPU1 DWT	跟踪触发——使能 CPU1 执行跟踪
6	ETMTRIGGER2	CPU1 DWT	跟踪触发——使能 CPU1 执行跟踪
7	ETMTRIGOUT	CPU1 ETM	ETM 触发——指示 CPU1 跟踪处于活动状态

表 251. CPU1 CTI 输出

编号	源信号	源组件	注释
0	EDBGRQ	CPU1	CPU1 停止请求——将 CPU1 置于调试模式
1	-	-	未使用
2	-	-	未使用
3	-	-	未使用

表 251. CPU1 CTI 输出（续）

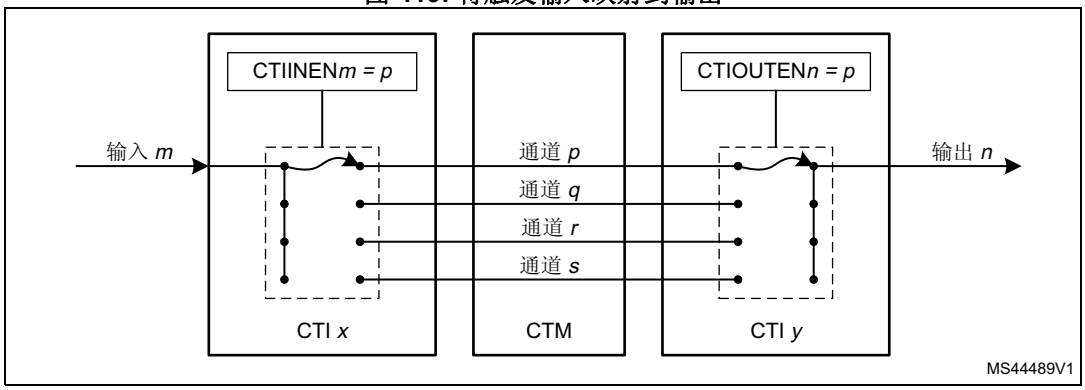
编号	源信号	源组件	注释
4	ETMEXTIN0	CPU1 ETM	ETM 触发 0 请求——使能 CPU1 执行跟踪
5	ETMEXTIN1	CPU1 ETM	ETM 触发 0 请求——使能 CPU1 执行跟踪
6	-	-	未使用
7	DBGRESTART	CPU1	CPU1 重启请求——CPU1 退出调试模式

交叉触发矩阵中有四个事件通道，在不同 CTI 的触发输入和输出之间，最多支持四个并行双向连接。要将 CTI *x* 上的输入编号 *m* 连接到 CTI *y* 上的输出编号 *n*，则必须使用 CTI *x* 的 CTIINEN*m* 寄存器将输入连接到事件通道 *p*。同一通道 *p* 必须使用 CTI *y* 的 CTIOUTEN*n* 寄存器连接到输出。

注： *这同样适用于同一个 CTI 的输入和输出之间的连接。*

一个输入可连接到多个通道（最多四个），因此一个输入可被发送到多个输出。类似地，一个输出可连接到多个输入。还可以将多个输入/输出连接到同一通道。

图 419. 将触发输入映射到输出



配置示例

当任一 CPU 内核遇到断点时，将停止另一内核。同步重启两个内核。

要在其中一个内核停止时也停止另一个内核，两个内核的 HALTED 输出必须均连接到对方的 EDBGRQ 输入。

通过*表 248* 和*表 250* 可知，来自 CPU2 的 HALTED 信号连接到 CPU2 CTI 的输入 0，而来自 CPU1 的 HALTED 信号连接到 CPU1 CTI 上的输入 0。因此，我们在每个 CTI 上编程 CTIIEN0 寄存器，将这些输入连接到 CTM 通道（例如，通道 0）。

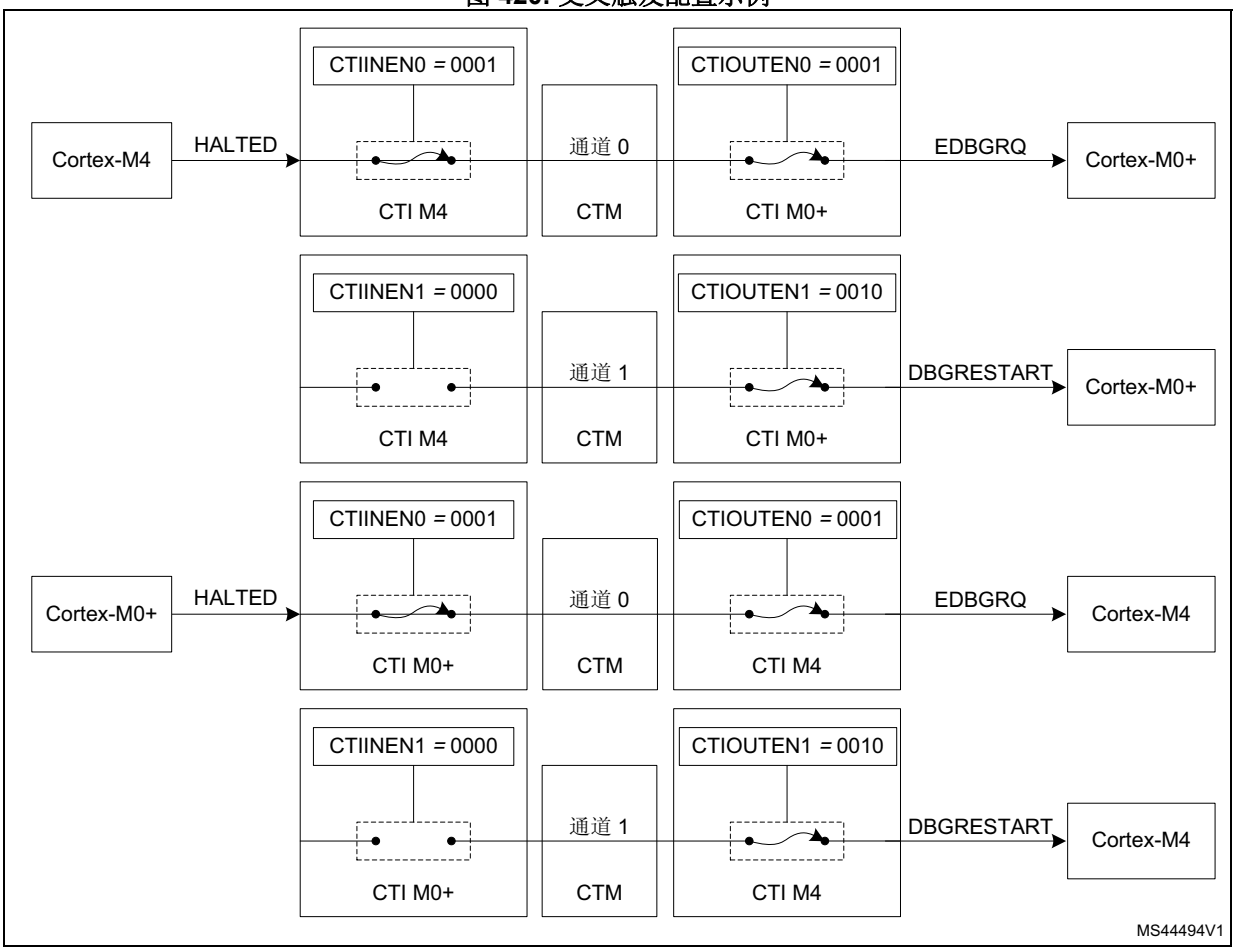
通过*表 249* 和*表 251* 可知，CPU 的 EDBGRQ 信号连接到相应 CTI 的输出 0。因此，我们在每个 CTI 上编程 CTIOUTEN0 寄存器，将这些输出连接到同一 CTM 通道。

要同时重启两个内核，调试器必须使用其中一个 CTI 中的 APPPULSE 寄存器。这样，调试器便可在四个 ETM 通道中的任何一个上生成脉冲。通道必须连接到两个内核的 DBGRESTART 信号。

通过*表 249* 和*表 251* 可知，CPU 的 DBGRESTART 信号连接到相应 CTI 的输出 1。因此，我们在每个 CTI 上编程 CTIOUTEN1 寄存器，将这些输出连接到未使用的 CTM 通道（例如，通道 1）。

图 420 对上述配置进行了说明。

图 420. 交叉触发配置示例



要强制两个处理器同时重启，请按以下步骤操作：

1. 通过依次将 0x01 和 0x00 写入每个 CTI 中的 CTIINTACK 寄存器来清除调试请求。
2. 通过将 0x02 写入任一 CTI 中的 APPPULSE 寄存器，在通道 1 上产生脉冲。这将对两个处理器生成重启请求。

   请注意，调试器还可通过将 0x01 写入任一 CTI 中的 APPPULSE 寄存器来强制两个内核同时停止，这将在通道 0 上生成脉冲。

有关交叉触发接口 CoreSight™ 组件的更多信息，请参见《Arm® CoreSight™ SoC-400 技术参考手册》*[2]*。

41.7 交叉触发接口寄存器

对于 CPU1 CTI，寄存器文件基址为 0xE0043000；对于 CPU2 CTI，寄存器文件基址为 0xF0001000。CPU1 CTI 和 CPU2 CTI 通过不同的访问端口进行访问。每个 CTI 的寄存器是相同的。

41.7.1 CTI 控制寄存器 (CTI_CONTROLR)

CTI control register

偏移地址：0x000

复位值：0x0000 0000

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

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

位 31:1 保留，必须保持复位值。

位 0 **GLBEN**：全局使能 (Global enable)
0：禁止交叉触发
1：使能交叉触发

41.7.2 CTI 触发确认寄存器 (CTI_INTACKR)

CTI trigger acknowledge register

偏移地址：0x010

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **INTACK[7:0]**：触发确认 (Trigger acknowledge)
每个 CTITRIGOUT 输出都使用一个寄存器位。当向该寄存器的某位写入 1 时，则确认相应的 CTITRIGOUT 输出，使其被清零。

41.7.3 CTI 应用程序触发设置寄存器 (CTI_APPSETR)

CTI application trigger set register

偏移地址：0x014

复位值：0x0000 0000

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

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

位 31:4 保留，必须保持复位值。

位 3:0 **APPSET[3:0]**：设置通道事件 (Set channel event)

读：

0bXXX0：通道 0 事件处于未激活状态
0bXXX0：通道 0 事件处于激活状态
0bXX0X：通道 1 事件处于未激活状态
0bXX1X：通道 1 事件处于激活状态
0bX0XX：通道 2 事件处于未激活状态
0bX1XX：通道 2 事件处于激活状态
0b0XXX：通道 3 事件处于未激活状态
0b1XXX：通道 3 事件处于激活状态

写：

0bXXX0：无影响
0bXXX0：设置通道 0 事件
0bXX0X：无影响
0bXX1X：设置通道 1 事件
0bX0XX：无影响
0bX1XX：设置通道 2 事件
0b0XXX：无影响
0b1XXX：设置通道 3 事件

41.7.4 CTI 应用程序触发清零寄存器 (CTI_APPCLEAR)

CTI application trigger clear register

偏移地址：0x018

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPCLEAR[3:0]			
												w			

位 31:4 保留，必须保持复位值。

位 3:0 **APPCLEAR[3:0]**：清除通道事件 (Clear channel event)

0b0000：无影响
0bXXX1：清除通道 0 事件
0bXX1X：清除通道 1 事件
0bX1XX：清除通道 2 事件
0b1XXX：清除通道 3 事件

41.7.5 CTI 应用程序脉冲寄存器 (CTI_APPPULSER)

CTI application pulse register

偏移地址：0x01C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPPULSE[3:0]			
												w			

位 31:4 保留，必须保持复位值。

位 3:0 **APPPULSE**：脉冲通道事件 (Pulse channel event)。该寄存器立即自清零。

0b0000：无影响
0bXXX1：生成通道 0 脉冲
0bXX1X：生成通道 1 脉冲
0bX1XX：生成通道 2 脉冲
0b1XXX：生成通道 3 脉冲

41.7.6 CTI 触发输入 x 使能寄存器 (CTI_INENRx)

CTI trigger In x enable register

偏移地址：0x020 + 4 * x，其中 x = 0 到 7

复位值：0x0000 0000

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

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

位 31:4 保留，必须保持复位值。

位 3:0 **TRIGINEN[3:0]**：当 CTITRIGINx（x = 0 到 7）处于激活状态时，使能或禁止四个通道中每个通道上的交叉触发事件。

0b0000：触发不生成通道事件
0bXXX1：触发 n 生成通道 0 事件
0bXX1X：触发 n 生成通道 1 事件
0bX1XX：触发 n 生成通道 2 事件
0b1XXX：触发 n 生成通道 3 事件

41.7.7 CTI 触发输出 x 使能寄存器 (CTI_OUTENRx)

CTI trigger out x enable register

偏移地址：0x0A0 + 4 * x，其中 x = 0 到 7

复位值：0x0000 0000

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

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

位 31:4 保留，必须保持复位值。

位 3:0 **TRIGOUTEN[3:0]**：对于每个通道，此位域定义了该通道上的事件是否能够生成 CTITRIGOUTx（x = 0 到 7）触发。

0b0000：通道事件不生成触发输出触发
0bXXX1：通道 0 事件生成触发输出 n 触发
0bXX1X：通道 1 事件生成触发输出 n 触发
0bX1XX：通道 2 事件生成触发输出 n 触发
0b1XXX：通道 3 事件生成触发输出 n 触发

41.7.8 CTI 触发输入状态寄存器 (CTI_TRGISTSR)

CTI trigger in status register

偏移地址：0x130

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINSTATUS[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **TRIGINSTATUS[7:0]**：触发输入状态 (Trigger input status)

每个 CTITRIGIN 输入都使用一个寄存器位。如果某位设置为 1，则表示相应触发输入处于激活状态。如果某位设置为 0，则表示相应触发输入处于未激活状态。

41.7.9 CTI 触发输出状态寄存器 (CTI_TRGOSTSR)

CTI trigger out status register

偏移地址：0x134

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTSTATUS[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **TRIGOUTSTATUS[7:0]**：触发输出状态 (Trigger output status)

每个 CTITRIGOUT 输出都使用一个寄存器位。如果某位设置为 1，则表示相应触发输出处于激活状态。如果某位设置为 0，则表示相应触发输出处于未激活状态。

41.7.10 CTI 通道输入状态寄存器 (CTI_CHINSTSR)

CTI channel in status register

偏移地址：0x138

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHINSTATUS[3:0]			
												r			

位 31:4 保留，必须保持复位值。

位 3:0 **CHINSTATUS[3:0]**：通道输入状态 (Channel input status)

每个通道输入都使用一个寄存器位。如果某位设置为 1，则表示相应通道输入处于激活状态。如果某位设置为 0，则表示相应通道输入处于未激活状态。

41.7.11 CTI 通道输出状态寄存器 (CTI_CHOUTSTSR)

CTI channel out status register

偏移地址：0x13C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHOUTSTATUS[3:0]			
												r			

位 31:4 保留，必须保持复位值。

位 3:0 **CHOUTSTATUS[3:0]**：通道输出状态 (Channel output status)

每个通道输出都使用一个寄存器位。如果某位设置为 1，则表示相应通道输出处于激活状态。如果某位设置为 0，则表示相应通道输出处于未激活状态。

41.7.12 CTI 通道门控寄存器 (CTI_GATER)

CTI channel gate register

偏移地址：0x140

复位值：0x0000 000F

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

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

位 31:4 保留，必须保持复位值。

位 3:0 **GATEEN[3:0]**：通道输出使能 (Channel output enable)。对于每个通道，定义了该通道事件能否通过 CTM 传播到其他 CTI。

0b0000：通道事件不能传播
0bXXX1：通道 0 事件能够传播
0bXX1X：通道 1 事件能够传播
0bX1XX：通道 2 事件能够传播
0b1XXX：通道 3 事件能够传播

41.7.13 CTI 声明标记设置寄存器 (CTI_CLAIMSETR)

CTI claim tag set register

偏移地址：0xFA0

复位值：0x0000 000F

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

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

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMSET[3:0]**：设置声明标记位 (Set claim tag bits)

写：

0000：无影响

xxx1：将位 0 置 1

xx1x：将位 1 置 1

x1xx：将位 2 置 1

1xxx：将位 3 置 1

读：

0xF：表示声明标记使用四个位

41.7.14 CTI 声明标记清零寄存器 (CTI_CLAIMCLR)

CTI claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

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

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

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMCLR[3:0]**：重置声明标记位 (Reset claim tag bits)

写：

0000：无影响

xxx1：将位 0 清零

xx1x：将位 1 清零

x1xx：将位 2 清零

1xxx：将位 3 清零

读操作：返回声明标记的当前值

41.7.15 CTI 锁定访问寄存器 (CTI_LAR)

CTI lock access register

偏移地址：0xFB0

复位值：N/A

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

位 31:0 **ACCESS_W[31:0]**：使处理器内核能够对某些 CTI 寄存器进行写访问（调试器无需解锁组件）

0xC5AC CE55：使能写访问

其他值：禁止写访问

41.7.16 CTI 锁定状态寄存器 (CTI_LSR)

CTI lock status register

偏移地址：0xFB4

复位值：0x0000 0003

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

位 31:3 保留，必须保持复位值。

位 2 **LOCKTYPE**：指示 CTI_LAR 寄存器大小

0：32 位

位 1 **LOCKGRANT**：当前锁定状态 (Current status of lock)。此位始终由外部调试器读为零。
0：允许写访问
1：阻止写访问。只允许读取操作。

位 0 **LOCKEXIST**：指示是否存在锁定控制机制。此位始终由外部调试器读为零。
0：不存在锁定控制机制
1：存在锁定控制机制

41.7.17 CTI 认证状态寄存器 (CTI_AUTHSTATR)

CTI authentication status register

偏移地址：0xFB8

复位值：0x0000 000A

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
								r		r		r		r	

位 31:8 保留，必须保持复位值。

位 7:6 **SNID[1:0]**：安全非侵入式调试的安全等级 (Security level for secure non-invasive debug)
0x0：未设计

位 5:4 **SID**[1:0]：安全侵入式调试的安全等级 (Security level for secure invasive debug)
0x0：未设计

位 3:2 **NSNID[1:0]**：非安全非侵入式调试的安全等级 (Security level for non-secure non-invasive debug)
0x2：禁止
0x3：使能

位 1:0 **NSID**[1:0]：非安全侵入式调试的安全等级 (Security level for non-secure invasive debug)
0x2：禁止
0x3：使能

41.7.18 CTI 设备配置寄存器 (CTI_DEVIDR)

CTI device configuration register

偏移地址：0xFC8

复位值：0x0004 0800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NUMCH[3:0]			
												r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMTRIG[7:0]								Res.	Res.	Res.	EXTMUXNUM[4:0]				
r											r				

位 31:20 保留，必须保持复位值。

位 19:16 **NUMCH[3:0]**：可用 ECT 通道数量 (Number of ECT channels available)
0x4：4 个通道

位 15:8 **NUMTRIG[7:0]**：可用 ECT 触发数量 (Number of ECT triggers available)
0x8：8 个触发输入和 8 个触发输出

位 7:5 保留，必须保持复位值。

位 4:0 **EXTMUXNUM[4:0]**：触发输入/输出多路复用器数量 (Number of trigger input/output multiplexers)
0x0：无

41.7.19 CTI 器件类型标识寄存器 (CTI_DEVTYPER)

CTI Device Type Identifier register

偏移地址：0xFCC

复位值：0x0000 0014

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **SUBTYPE[3:0]**：子分类 (Sub-classification)
0x1：表示该组件为交叉触发组件。

位 3:0 **MAJORTYPE[3:0]**：主分类 (Major classification)
0x4：表示该组件允许调试器控制 CoreSight™ SoC-400 系统中的其他组件。

41.7.20 CTI CoreSight 外设标识寄存器 4 (CTI_PIDR4)

CTI CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：Arm® JEDEC 代码

41.7.21 CTI CoreSight 外设标识寄存器 0 (CTI_PIDR0)

CTI CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0006

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (Part number bits [7:0])
0x06：CTI 产品编号

41.7.22 CTI CoreSight 外设标识寄存器 1 (CTI_PIDR1)

CTI CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (Part number bits [11:8])
0x9：CTI 产品编号

41.7.23 CTI CoreSight 外设标识寄存器 2 (CTI_PIDR2)

CTI CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 004B

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x4：r0p5

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x3：Arm® JEDEC 代码

41.7.24 CTI CoreSight 外设标识寄存器 3 (CTI_PIDR3)

CTI CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

41.7.25 CTI CoreSight 组件标识寄存器 0 (CTI_CIDR0)

CTI CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (Component ID bits [7:0])
0x0D：公共 ID 值

41.7.26 CTI CoreSight 外设标识寄存器 1 (CTI_CIDR1)

CTI CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (Component ID bits [15:12])——组件类
0x9：CoreSight™ 组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (Component ID bits [11:8])
0x0：公共 ID 值

41.7.27 CTI CoreSight 组件标识寄存器 2 (CTI_CIDR2)

CTI CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (Component ID bits [23:16])
0x05：公共 ID 值

41.7.28 CTI CoreSight 组件标识寄存器 3 (CTI_CIDR3)

CTI CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (Component ID bits [31:24])
0xB1：公共 ID 值

41.7.29 CTI 寄存器映射和复位值

表 252. CTI 寄存器映射和复位值

偏移	寄存器名称	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
0x000	CTI_CONTROLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GLBEN
	Reset value																																0
0x010	CTI_INTACKR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTACK[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x014	CTI_APPSETR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPSET[3:0]			
	Reset value																													0	0	0	0
0x018	CTI_APPCLEAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPCLEAR [3:0]			
	Reset value																													0	0	0	0
0x01C	CTI_APPPULSER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPPULSE [3:0]			
	Reset value																													0	0	0	0
0x020	CTI_INENR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x024	CTI_INENR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x028	CTI_INENR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x02C	CTI_INENR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x030	CTI_INENR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x034	CTI_INENR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x038	CTI_INENR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x03C	CTI_INENR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x0A0	CTI_OUTENR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN [3:0]			
	Reset value																													0	0	0	0
0x0S4	CTI_OUTENR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN [3:0]			
	Reset value																													0	0	0	0
0x0S8	CTI_OUTENR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN [3:0]			
	Reset value																													0	0	0	0

表 252. CTI 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x0SC	CTI_OUTENR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
	Reset value																													0	0	0	0
0x0B0	CTI_OUTENR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
	Reset value																													0	0	0	0
0x0B4	CTI_OUTENR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
	Reset value																													0	0	0	0
0x0B8	CTI_OUTENR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
	Reset value																													0	0	0	0
0x0BC	CTI_OUTENR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN[3:0]			
	Reset value																													0	0	0	0
0x130	CTI_TRIGISTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINSTATUS[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x134	CTI_TRIGOSTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTSTATUS[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x138	CTI_CHINSTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHISTATUS[3:0]			
	Reset value																													0	0	0	0
0x13C	CTI_CHOUTSTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHOSTATUS[3:0]			
	Reset value																													0	0	0	0
0x140	CTI_GATER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GATEEN[3:0]			
	Reset value																													1	1	1	1
0xFA0	CTI_CLAIMSETR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
	Reset value																													1	1	1	1
0xFA4	CTI_CLAIMCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
	Reset value																													0	0	0	0
0xFB0	CTI_LAR	KEY																															
	Reset value																																
0xFB4	CTI_LSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NTT	SLK	SLI
	Reset value																														0	1	1
0xFB8	CTI_AUTHSTATR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
	Reset value																									0	0	0	0	0	1	0	1
0xFC8	CTI_DEVIDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NUMCH[3:0]				NUMTRIG[7:0]								Res.	Res.	Res.	EXMUXNUM[4:0]				
	Reset value													0	1	0	0	0	0	0	0	1	0	0	0				0	0	0	0	0
0xFCC	CTI_DEVTYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUB[3:0]				MAJOR[3:0]			
	Reset value																									0	0	0	1	0	1	0	0

表 252. CTI 寄存器映射和复位值（续）

偏移	寄存器名称	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
0xFD0	CTI_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFE0	CTI_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	0	0	0	1	1	0
0xFE4	CTI_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	CTI_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	1	0	0	1	0	1	1
0xFEC	CTI_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	CTI_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	CTI_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	CTI_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	CTI_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

有关寄存器边界地址的信息，请参见*第 41.9 节：CPU2 ROM 表*和*第 41.13 节：CPU1 ROM 表*。

41.8 微控制器调试单元 (DBGMCU)

DBGMCU 组件包含多个寄存器，用来在调试模式下控制电源和时钟行为。它允许调试器（或调试软件）：

- 在低功耗模式（休眠、停止或待机）下，保持 CPU1 处理器内核的时钟和电源；在低功耗调试模式下，CPU2 操作不受影响
- 在低功耗模式下，保持系统调试和跟踪组件的时钟和电源
- 当处理器内核在调试模式下停止时，停止某些外设（看门狗、定时器、RTC）的时钟

DBGMCU 寄存器不通过系统复位而复位，只能通过上电复位而复位。调试器可通过基址为 0xE0042000 的 CPU1 AHB 访问端口对其进行访问。

注： *DBGMCU 不是标准 CoreSight™ 组件，因此它不会显示在 CPU1 ROM 表中。*

41.8.1 DBGMCU 标识代码寄存器 (DBGMCU_IDCODE)

DBGMCU Identity Code register

偏移地址：0x000

复位值：0x2001 6495

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DEV_ID[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **REV_ID[15:0]**：版本 (Revision)
0x2001 = 版本 2.1

位 15:12 保留，必须保持复位值。

位 11:0 **DEV_ID[11:0]**：器件 ID (Device ID)
0x495：STM32WB55xx

41.8.2 DBGMCU 配置寄存器 (DBGMCU_CR)

DBGMCU configuration register

偏移地址：0x004

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRACE_IOEN	Res.	Res.	DBG_STANDBY	DBG_STOP	DBG_SLEEP
										rw			rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28 **TRGOEN**：外部触发输出使能 (External trigger output enable)。此位控制双向触发引脚 TRIG_INOUT 的方向。

0：输入。TRIG_INOUT 连接到 TRGIN。

1：输出。TRIG_INOUT 连接到 TRGOUT。

位 27:6 保留，必须保持复位值。

位 5 **TRACE_IOEN**：跟踪端口和时钟使能 (Trace port and clock enable)。此位使能跟踪端口时钟 TRACECLK。

0：禁止。

1：使能。

位 4:3 保留，必须保持复位值。

位 2 **DBG_STANDBY**：允许在 STANDBY 模式下调试 CPU1，对 CPU2 操作无影响。

0：正常工作。在 STANDBY 模式下，将禁止所有时钟，器件将自动掉电。

1：禁止自动停止时钟/掉电。所有活动的 CPU1 时钟和振荡器将在 STANDBY 模式下继续运行，并将保持器件电源，因此支持全面的 CPU1 调试功能。在退出 STANDBY 模式时，执行器件复位。

位 1 **DBG_STOP**：允许在 STOP 模式下调试 CPU1，对 CPU2 操作无影响。

0：正常工作。在 STOP 模式下，将自动禁止所有 CPU1 时钟。

1：禁止自动停止时钟。所有活动的时钟和振荡器将在 STOP 模式下继续运行，因此支持全面的 CPU1 调试功能。退出 STOP 时，时钟设置将被设置为 STOP 模式退出状态。

位 0 **DBG_SLEEP**：允许在 SLEEP 模式下调试 CPU1，对 CPU2 操作无影响。

0：正常工作。在 SLEEP 模式时，将自动停止处理器时钟。

1：禁止自动停止时钟。CPU1 处理器时钟将继续运行，因此支持全面的 CPU1 调试功能。

41.8.3 DBGMCU CPU1 APB1 外设冻结寄存器 1 (DBGMCU_APB1FZR1)

DBGMCU CPU1 APB1 peripheral freeze register 1

偏移地址：0x03C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DBG_ LPTIM1_ STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_ I2C3_ STOP	Res.	DBG_ I2C1_ STOP	Res.	Res.	Res.	Res.	Res.
rw								rw		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBG_ IWDG_ STOP	DBG_ WWDG _STOP	DBG_ RTC_ STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_ TIM2_ STOP
			rw	rw	rw										rw

DBG_LPTIM1_STOP：在 CPU1 调试模式下停止 LPTIM1 (LPTIM1 stop in CPU1 debug)

位 31 0：正常工作。当 CPU1 处于调试模式时，LPTIM1 继续运行。
1：在调试模式下停止。当 CPU1 处于调试模式时，LPTIM1 被冻结。

位 30:24 保留，必须保持复位值。

DBG_I2C3_STOP：在 CPU1 调试模式下停止 I2C3 SMBUS 超时 (I2C3 SMBUS timeout stop in CPU1 debug)

位 23 0：正常工作。当 CPU1 处于调试模式时，I2C3 SMBUS 超时继续运行。
1：在调试模式下停止。当 CPU1 处于调试模式时，I2C3 SMBUS 超时被冻结。

位 22 保留，必须保持复位值。

DBG_I2C1_STOP：在 CPU1 调试模式下停止 I2C1 SMBUS 超时 (I2C1 SMBUS timeout stop in CPU1 debug)

位 21 0：正常工作。当 CPU1 处于调试模式时，I2C1 SMBUS 超时继续运行。
1：在调试模式下停止。当 CPU1 处于调试模式时，I2C1 SMBUS 超时被冻结。

位 20:13 保留，必须保持复位值。

DBG_IWDG_STOP：在 CPU1 调试模式下停止 IWDG (IWDG stop in CPU1 debug)

位 12 0：正常工作。当 CPU1 处于调试模式时，IWDG 继续运行。
1：在调试模式下停止。当 CPU1 处于调试模式时，IWDG 被冻结。

DBG_WWDG_STOP：在 CPU1 调试模式下停止 WWDG (WWDG stop in CPU1 debug)

位 11 0：正常工作。当 CPU1 处于调试模式时，WWDG 继续运行。
1：在调试模式下停止。当 CPU1 处于调试模式时，WWDG 被冻结。

DBG_RTC_STOP：在 CPU1 调试模式下停止 RTC (RTC stop in CPU1 debug)

位 10 0：正常工作。当 CPU1 处于调试模式时，RTC 继续运行。
1：在调试模式下停止。当 CPU1 处于调试模式时，RTC 被冻结。

位 9:1 保留，必须保持复位值。

DBG_TIM2_STOP：在 CPU1 调试模式下停止 TIM2 (TIM2 stop in CPU1 debug)

位 0 0：正常工作。当 CPU1 处于调试模式时，TIM2 继续运行。
1：在调试模式下停止。当 CPU1 处于调试模式时，TIM2 被冻结。

41.8.4 DBGMCU CPU2 APB1 外设冻结寄存器 1 (DBGMCU_C2APB1FZR1)

DBGMCU CPU2 APB1 peripheral freeze register 1

偏移地址：0x040

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DBG_LPTIM1_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_I2C3_STOP	Res.	DBG_I2C1_STOP	Res.	Res.	Res.	Res.	Res.
rw								rw		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBG_IWDG_STOP	Res.	DBG_RTC_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM2_STOP
			rw		rw										rw

DBG_LPTIM1_STOP：在 CPU2 调试模式下停止 LPTIM1 (LPTIM1 stop in CPU2 debug)

位 31 0：正常工作。当 CPU2 处于调试模式时，LPTIM1 继续运行。
1：在调试模式下停止。当 CPU2 处于调试模式时，LPTIM1 被冻结。

位 30:24 保留，必须保持复位值。

DBG_I2C3_STOP：在 CPU2 调试模式下停止 I2C3 SMBUS 超时 (I2C3 SMBUS timeout stop in CPU2 debug)

位 23 0：正常工作。当 CPU2 处于调试模式时，I2C3 SMBUS 超时继续运行。
1：在调试模式下停止。当 CPU2 处于调试模式时，I2C3 SMBUS 超时被冻结。

位 22 保留，必须保持复位值。

DBG_I2C1_STOP：在 CPU2 调试模式下停止 I2C1 SMBUS 超时 (I2C1 SMBUS timeout stop in CPU2 debug)

位 21 0：正常工作。当 CPU2 处于调试模式时，I2C1 SMBUS 超时继续运行。
1：在调试模式下停止。当 CPU2 处于调试模式时，I2C1 SMBUS 超时被冻结。

位 20:13 保留，必须保持复位值。

DBG_IWDG_STOP：在 CPU2 调试模式下停止 IWDG (IWDG stop in CPU2 debug)

位 12 0：正常工作。当 CPU2 处于调试模式时，IWDG 继续运行。
1：在调试模式下停止。当 CPU2 处于调试模式时，IWDG 被冻结。

位 11 保留，必须保持复位值。

DBG_RTC_STOP：在 CPU2 调试模式下停止 RTC (RTC stop in CPU2 debug)

位 10 0：正常工作。当 CPU2 处于调试模式时，RTC 继续运行。
1：在调试模式下停止。当 CPU2 处于调试模式时，RTC 被冻结。

位 9:1 保留，必须保持复位值。

DBG_TIM2_STOP：在 CPU2 调试模式下停止 TIM2 (TIM2 stop in CPU2 debug)

位 0 0：正常工作。当 CPU2 处于调试模式时，TIM2 继续运行。
1：在调试模式下停止。当 CPU2 处于调试模式时，TIM2 被冻结。

41.8.5 DBGMCU CPU1 APB1 外设冻结寄存器 2 (DBGMCU_APB1FZR2)

DBGMCU CPU1 APB1 peripheral freeze register 2

偏移地址：0x044

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_LPTIM2_STOP	Res.	Res.	Res.	Res.	Res.
										rw					

位 31:6 保留，必须保持复位值。

DBG_LPTIM2_STOP：在 CPU1 调试模式下停止 LPTIM2 (LPTIM2 stop in CPU1 debug)

位 5 0：正常工作。当 CPU1 处于调试模式时，LPTIM2 继续运行。
1：在调试模式下停止。当 CPU1 处于调试模式时，LPTIM2 被冻结。

位 4:0 保留，必须保持复位值。

41.8.6 DBGMCU CPU2 APB1 外设冻结寄存器 2 (DBGMCU_C2APB1FZR2)

DBGMCU CPU2 APB1 peripheral freeze register 2

偏移地址：0x048

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_ LPTIM2_ STOP	Res.	Res.	Res.	Res.	Res.
										rw					

位 31:6 保留，必须保持复位值。

DBG_LPTIM2_STOP：在 CPU2 调试模式下停止 LPTIM2 (LPTIM2 stop in CPU2 debug)

位 5 0：正常工作。当 CPU2 处于调试模式时，LPTIM2 继续运行。
1：在调试模式下停止。当 CPU2 处于调试模式时，LPTIM2 被冻结。

位 4:0 保留，必须保持复位值。

41.8.7 DBGMCU CPU1 APB2 外设冻结寄存器 (DBGMCU_APB2FZR)

DBGMCU CPU1 APB2 peripheral freeze register

偏移地址：0x04C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_ TIM17_ STOP	DBG_ TIM16_ STOP	Res.
													rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DBG_ TIM1_ STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
				rw											

位 31:19 保留，必须保持复位值。

DBG_TIM17_STOP：在 CPU1 调试模式下停止 TIM17 (TIM17 stop in CPU1 debug)

位 18
0：正常工作。当 CPU1 处于调试模式时，TIM17 继续运行。
1：在调试模式下停止。当 CPU1 处于调试模式时，TIM17 被冻结。

DBG_TIM16_STOP：在 CPU1 调试模式下停止 TIM16 (TIM16 stop in CPU1 debug)

位 17
0：正常工作。当 CPU1 处于调试模式时，TIM16 继续运行。
1：在调试模式下停止。当 CPU1 处于调试模式时，TIM16 被冻结。

位 16:12 保留，必须保持复位值。

DBG_TIM1_STOP：在 CPU1 调试模式下停止 TIM1 (TIM1 stop in CPU1 debug)

位 11
0：正常工作。当 CPU1 处于调试模式时，TIM1 继续运行。
1：在调试模式下停止。当 CPU1 处于调试模式时，TIM1 被冻结。

位 10:0 保留，必须保持复位值。

41.8.8 DBGMCU CPU2 APB2 外设冻结寄存器 (DBGMCU_C2APB2FZR)

DBGMCU CPU2 APB2 peripheral freeze register

偏移地址：0x048

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM17_STOP	DBG_TIM16_STOP	Res.
													rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DBG_TIM1_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
				rw						rw					

位 31:19 保留，必须保持复位值。

DBG_TIM17_STOP：在 CPU1 调试模式下停止 TIM17 (TIM17 stop in CPU1 debug)

位 18
0：正常工作。当 CPU1 处于调试模式时，TIM17 继续运行。
1：在调试模式下停止。当 CPU1 处于调试模式时，TIM17 被冻结。

DBG_TIM16_STOP：在 CPU1 调试模式下停止 TIM16 (TIM16 stop in CPU1 debug)

位 17
0：正常工作。当 CPU1 处于调试模式时，TIM16 继续运行。
1：在调试模式下停止。当 CPU1 处于调试模式时，TIM16 被冻结。

位 16:12 保留，必须保持复位值。

DBG_TIM1_STOP：在 CPU1 调试模式下停止 TIM1 (TIM1 stop in CPU1 debug)

位 11
0：正常工作。当 CPU1 处于调试模式时，TIM1 继续运行。
1：在调试模式下停止。当 CPU1 处于调试模式时，TIM1 被冻结。

位 10:0 保留，必须保持复位值。

41.8.9 DBGMCU 寄存器映射和复位值

表 253. DBGMCU 寄存器映射和复位值

偏移	寄存器名称	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
0x000	DBGMCU_IDCODE	REV_ID[15:0]																Res.	Res.	Res.	Res.	DEV_ID[11:0]											
	Reset value	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	1	0	0	1	0	1	0	1
0x004	DBGMCU_CR	Res.	Res.	Res.	TRGOEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRACE_IOEN	Res.	Res.	DBG_STANDBY	DBG_STOP	DBG_SLEEP
	Reset value				0																							0			0	0	0
0x008-0x038	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x03C	DBGMCU_APB1FZR1	DBG_LPTIM1_STOP								DBG_I2C3_STOP		DBG_I2C1_STOP								DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP										DBG_TIM2_STOP
	Reset value	0								0		0								0	0	0											0
0x040	DBGMCU_C2APB1FZR1	DBG_LPTIM1_STOP								DBG_I2C3_STOP		DBG_I2C1_STOP								DBG_IWDG_STOP		DBG_RTC_STOP										DBG_TIM2_STOP
	Reset value	0								0		0								0		0											0
0x044	DBGMCU_APB1FZR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_LPTIM2_STOP	Res.	Res.	Res.	Res.	Res.
	Reset value																											0					
0x048	DBGMCU_C2APB1FZR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_LPTIM2_STOP	Res.	Res.	Res.	Res.	Res.
	Reset value																											0					
0x04C	DBGMCU_APB2FZR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM17_STOP	DBG_TIM16_STOP	Res.	Res.	Res.	Res.	Res.	DBG_TIM1_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value														0	0						0											

表 253. DBGMCU 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x050	DBGMCU_ C2APB2FZR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM17_STOP	DBG_TIM16_STOP	Res.	Res.	Res.	Res.	Res.	DBG_TIM1_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value														0	0						0											

有关寄存器边界地址的信息，请参见*第 41.8 节：微控制器调试单元 (DBGMCU)*。

41.9 CPU2 ROM 表

ROM 表是 CoreSight™ 组件，包含可通过 AHBD 访问的所有 Coresight™ 调试组件的基址。这些表允许调试器自动发现 CoreSight™ 系统的拓扑。

CPU2 子系统有两个 ROM 表：

1. ROM1：CPU2 处理器 ROM 表，由 CPU2 AHB-AP 中的 BASE 寄存器指向。包含 CTI 以及 CPU2 ROM 表的基址指针。
2. ROM2：CPU2 ROM 表，包含指向 CPU2 系统控制空间寄存器的指针（以允许调试器识别 CPU 内核）以及指向 CPU2 子系统中其他 CoreSight™ 组件（PBU、DWT）的指针。

CPU2 处理器 ROM 表占用一个 4 KB、32 位宽的 AHB 地址空间块，地址范围从 0xF0000000 到 0xF0000FFC。

表 254. CPU2 处理器 ROM 表

ROM 表中的地址	组件名称	组件基址	组件偏移地址	大小	条目
0xF0000000	CPU2 ROM 表	0xE00FF000	0xF00FF000	4 KB	0xF00FF003
0xF0000004	CTI	0xF0001000	0x00001000	4 KB	0x00001003
0xF0000008	未使用	-	-	-	0x00002002
0xF000000C	未使用	-	-	-	0x10000002
0xF0000010	表顶部	-	-	-	0x00000000
0xF000000C 到 0xF0000FC8	保留	-	-	-	0x00000000
0xF0000FCC 到 0xF0000FFC	ROM 表寄存器	-	-	-	请参见*表 256*

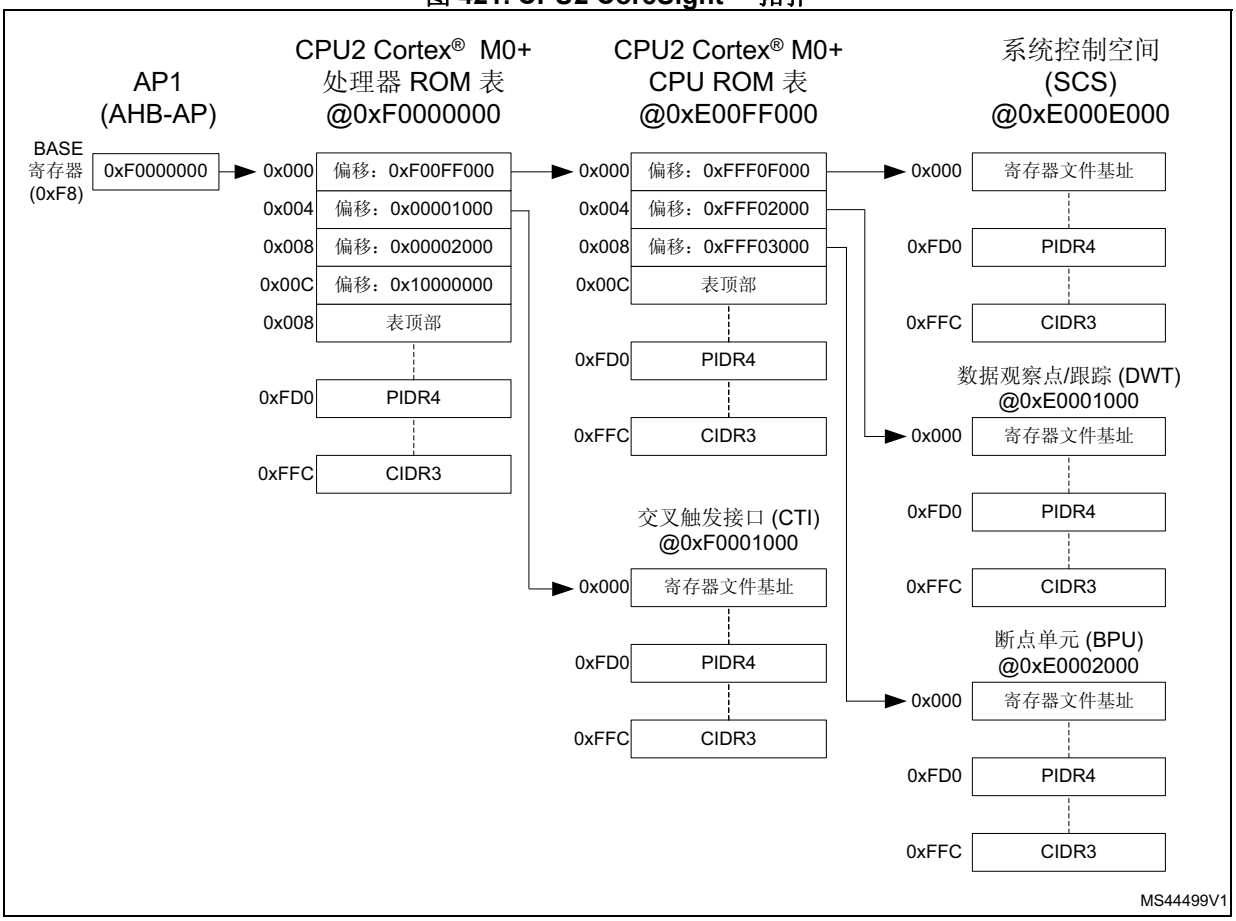
CPU2 ROM 表占用一个 4 KB、32 位宽的 APB-D 地址空间块，地址范围从 0xE00FF000 到 0xE00FFFFC。

表 255. CPU2 ROM 表

ROM 表中的地址	组件名称	组件基址	组件偏移地址	大小	条目
0xE00FF000	SCS	0xE000E000	0xFFF0F000	4 KB	0xFFF0F003
0xE00FF004	DWT	0xE0001000	0xFFF02000	4 KB	0xFFF02003
0xE00FF008	BPU	0xE0002000	0xFFF03000	4 KB	0xFFF03003
0xE00FF00C	表顶部	-	-	-	0x00000000
0xE00FF010 到 0xE00FFFC8	保留	-	-	-	0x00000000
0xE00FFFCC 到 0xE00FFFFC	ROM 表寄存器	-	-	-	请参见*表 257*

CPU2 子系统的 CoreSight™ 组件拓扑如*图 421* 所示。

图 421. CPU2 CoreSight™ 拓扑



41.9.1 CPU2 ROM1 存储器类型寄存器 (C2ROM1_MEMTYPER)

CPU2 ROM1 memory type register

偏移地址：0xFCC

复位值：0x0000 0001

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

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

位 31:1 保留，必须保持复位值。

位 0 **SYSMEM**：系统存储器 (System memory)
0x1：该总线上有系统存储器

41.9.2 CPU2 ROM1 CoreSight 外设标识寄存器 4 (C2ROM1_PIDR4)

CPU2 ROM1 CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：Arm® JEDEC 连续代码

41.9.3 CPU2 ROM1 CoreSight 外设标识寄存器 0 (C2ROM1_PIDR0)

CPU2 ROM1 CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0C0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (Part number bits [7:0])
0xC0：Cortex®-M0+ 处理器 ROM 表

41.9.4 CPU2 ROM1 CoreSight 外设标识寄存器 1 (C2ROM1_PIDR1)

CPU2 ROM1 CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B4

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (Part number bits [11:8])
0x4：Cortex®-M0+ 处理器 ROM 表

41.9.5 CPU2 ROM1 CoreSight 外设标识寄存器 2 (C2ROM1_PIDR2)

CPU2 ROM1 CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 000B

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x0：rev r0p0

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x3：Arm® JEDEC 代码

41.9.6 CPU2 ROM1 CoreSight 外设标识寄存器 3 (C2ROM1_PIDR3)

CPU2 ROM1 CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

41.9.7 CPU2 ROM1 CoreSight 组件标识寄存器 0 (C2ROM1_CIDR0)

CPU2 ROM1 CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (Component ID bits [7:0])
0x0D：公共 ID 值

41.9.8 CPU2 ROM1 CoreSight 外设标识寄存器 1 (C2ROM1_CIDR1)

CPU2 ROM1 CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 0010

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (Component ID bits [15:12])——组件类
0x1：ROM 表组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (Component ID bits [11:8])
0x0：公共 ID 值

41.9.9 CPU2 ROM1 CoreSight 组件标识寄存器 2 (C2ROM1_CIDR2)

CPU2 ROM1 CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (Component ID bits [23:16])
0x05：公共 ID 值

41.9.10 CPU2 ROM1 CoreSight 组件标识寄存器 3 (C2ROM1_CIDR3)

CPU2 ROM1 CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (Component ID bits [31:24])
0xB1：公共 ID 值

41.9.11 CPU2 处理器 ROM 表寄存器和复位值

表 256. CPU2 处理器 ROM 表寄存器映射和复位值

偏移	寄存器名称	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
0xFCC	C2ROM1_ MEMTYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSMEM
	Reset value																																1
0xFD0	C2ROM1_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFE0	C2ROM1_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									1	1	0	0	0	0	0	0
0xFE4	C2ROM1_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	0	1	0	0
0xFE8	C2ROM1_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	0	1	0	1	1
0xFEC	C2ROM1_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	C2ROM1_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	C2ROM1_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									0	0	0	1	0	0	0	0
0xFF8	C2ROM1_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	C2ROM1_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

有关寄存器边界地址的信息，请参见*第 41.9 节：CPU2 ROM 表*。

41.9.12 CPU2 ROM2 存储器类型寄存器 (C2ROM2_MEMTYPER)

CPU2 ROM2 memory type register

偏移地址：0xFCC

复位值：0x0000 0001

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

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

位 31:1 保留，必须保持复位值。

位 0 **SYSMEM**：系统存储器 (System memory)
0x1：该总线上有系统存储器

41.9.13 CPU2 ROM2 CoreSight 外设标识寄存器 4 (C2ROM2_PIDR4)

CPU2 ROM2 CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：Arm® JEDEC 连续代码

41.9.14 CPU2 ROM2 CoreSight 外设标识寄存器 0 (C2ROM2_PIDR0)

CPU2 ROM2 CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0C0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (Part number bits [7:0])
0xC0：CPU2 ROM 表

41.9.15 CPU2 ROM2 CoreSight 外设标识寄存器 1 (C2ROM2_PIDR1)

CPU2 ROM2 CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B4

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (Part number bits [11:8])
0x4：CPU2 ROM 表

41.9.16 CPU2 ROM2 CoreSight 外设标识寄存器 2 (C2ROM2_PIDR2)

CPU2 ROM2 CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 000B

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x0：rev r0p0

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x3：Arm® JEDEC 代码

41.9.17 CPU2 ROM2 CoreSight 外设标识寄存器 3 (C2ROM2_PIDR3)

CPU2 ROM2 CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

41.9.18 CPU2 ROM2 CoreSight 组件标识寄存器 0 (C2ROM2_CIDR0)

CPU2 ROM2 CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (Component ID bits [7:0])
0x0D：公共 ID 值

41.9.19 CPU2 ROM2 CoreSight 外设标识寄存器 1 (C2ROM2_CIDR1)

CPU2 ROM2 CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 0010

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (Component ID bits [15:12])——组件类
0x1：ROM 表组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (Component ID bits [11:8])
0x0：公共 ID 值

41.9.20 CPU2 ROM2 CoreSight 组件标识寄存器 2 (C2ROM2_CIDR2)

CPU2 ROM2 CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (Component ID bits [23:16])
0x05：公共 ID 值

41.9.21 CPU2 ROM2 CoreSight 组件标识寄存器 3 (C2ROM2_CIDR3)

CPU2 ROM2 CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (Component ID bits [31:24])
0xB1：公共 ID 值

41.9.22 CPU2 ROM 表寄存器映射和复位值

表 257. CPU2 ROM 表寄存器映射和复位值

偏移	寄存器名称	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
0xFCC	C2ROM2_ MEMTYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSMEM
	Reset value																																1
0xFD0	C2ROM2_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON[3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFE0	C2ROM2_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									1	1	0	0	0	0	0	0
0xFE4	C2ROM2_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	0	1	0	0
0xFE8	C2ROM2_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	0	1	0	1	1
0xFEC	C2ROM2_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	C2ROM2_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	C2ROM2_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									0	0	0	1	0	0	0	0
0xFF8	C2ROM2_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	C2ROM2_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

有关寄存器边界地址的信息，请参见*第 41.9 节：CPU2 ROM 表*。

41.10 CPU2 数据观察点和跟踪单元 (DWT)

DWT 提供了四个比较器，可用作：

- 观察点
- ETM 触发器
- PC 采样触发器
- 数据地址采样触发器
- 数据比较器（仅适用于比较器 1）
- 时钟周期计数器比较器（仅适用于比较器 0）

还包含以下计数器：

- 时钟周期
- 分支指令
- 加载存储单元 (LSU) 操作
- 睡眠周期
- 每条指令周期数
- 中断开销

DWT 比较器会将其 DWT_COMPR 寄存器中保存的值与下列其中一项进行比较：

- 数据地址
- 指令地址
- 数据值
- 周期计数值（仅适用于比较器 0）

比较器可以使用掩码进行地址匹配，因此可匹配一系列地址。

一旦匹配成功时，比较器会生成以下其中一项：

- 一个或多个 DWT 数据跟踪数据包，包括一个或多个：
  - 引发数据访问的指令的地址
  - 偏移地址，数据访问地址的位 [15:0]
  - 匹配的数据值
- 观察点调试事件（在 PC 值上或者在被访问的数据地址上）。
- 指示匹配超出 DWT 单元的 CMPMATCH[N] 事件。

观察点调试事件会生成 DebugMonitor 异常，或者导致处理器停止执行并进入调试状态。

有关如何使用 DWT 的更多详细信息，请参见《Arm®v7-M 架构参考手册》*[5]*。

41.10.1 DWT 控制寄存器 (DWT_CTRLR)

DWT control register

偏移地址：0x000

复位值：0x4000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMCOMP[3:0].				NOTRCPKT	NOEXTTRIG	NOCYCCNT	NOPRFCNT	Res.	CYCEVTENA	FOLDEVTENA	LSUEVTENA	SLEEPEVTENA	EXCEVTENA	CPIEVTENA	EXCTRCENA
r				r	r	r	r		rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	PCSAMPLENA	SYNCTAP[1:0]		CYCTAP	POSTINIT[3:0]				POSTRESET[3:0]				CYCCNTENA
			rw	rw		rw	rw				rw				rw

位 31:28 **NUMCOMP**：设计的比较器数量 (Number of comparators implemented)，只读值

0x4：四个比较器

位 27 **NOTRCPKT**：支持跟踪采样和异常跟踪 (Trace sampling and exception tracing support)，只读值

0x0：支持

位 26 **NOEXTTRIG**：外部信号匹配 CMPMATCH 支持 (External match signal, CMPMATCH support)，只读值

0x0：支持

位 25 **NOCYCCNT**：循环计数器支持 (Cycle counter support)，只读值

0x0：支持

位 24 **NOPRFCNT**：性能分析计数器支持 (Profiling counter support)，只读值

0x0：支持

位 23 保留，必须保持复位值。

位 22 **CYCEVTENA**：使能 POSTCNT 下溢事件计数器数据包生成 (Enable for POSTCNT underflow event counter packet generation)

0x0：禁止

0x1：使能

位 21 **FOLDEVTENA**：使能折叠指令计数器上溢事件生成 (Enable for folded instruction counter overflow event generation)

0x0：禁止

0x1：使能

位 20 **LSUEVTENA**：使能 LSU 计数器上溢事件生成 (Enable for LSU counter overflow event generation)

0x0：禁止

0x1：使能

位 19 **SLEEPEVTENA**：使能睡眠计数器上溢事件生成 (Enable for sleep counter overflow event generation)

0x0：禁止

0x1：使能

位 18 **EXCEVTENA**：使能异常开销计数器上溢事件生成 (Enable for exception overhead counter overflow event generation)

0x0：禁止

0x1：使能

位 17 **CPIEVTENA**：使能 CPI 计数器上溢事件生成 (Enable for CPI counter overflow event generation)
0x0：禁止
0x1：使能

位 16 **EXCTRCENA**：使能异常跟踪生成 (Enable for exception trace generation)
0x0：禁止
0x1：使能

位 15:13 保留，必须保持复位值。

位 12 **PCSAMPLENA**：使能将 POSTCNT 计数器用作定时器，用于定期生成 PC 样本数据包。
0x0：禁止
0x1：使能

位 11:10 **SYNCTAP[1:0]**：选择同步数据包计数器点击 CYCCNT 计数器的位置。此位域确定同步数据包速率。
0x0：禁止。无同步数据包
0x1：点击位置为 CYCCNT[24]
0x2：点击位置为 CYCCNT[26]
0x3：点击位置为 CYCCNT[28]

位 9 **CYCTAP**：选择 POSTCNT 点击 CYCCNT 计数器的位置。
0x0：点击位置为 CYCCNT[6]
0x1：点击位置为 CYCCNT[10]

位 8:5 **POSTINIT[3:0]**：POSTCNT 计数器初始值 (Initial value of the POSTCNT counter)。如果使能了 POSTCNT 计数器，则忽略对此位域的写操作（即，在写入 POSTINIT 之前 CYCEVTENA 或 PCSAMPLENA 必须复位）。

位 4:1 **POSTPRESET[3:0]**：POSTCNT 计数器的重载值 (Reload value of the POSTCNT counter)

位 0 **CYCCNTENA**：使能 CYCCNT 计数器。
0x0：禁止
0x1：使能

41.10.2 DWT 周期计数寄存器 (DWT_CYCCNTR)

DWT cycle count register

偏移地址：0x004

复位值：0x0000 0000

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

位 31:0 **CYCCNT[31:0]**：处理器时钟周期计数器 (Processor clock cycle counter)

41.10.3 DWT CPI 计数寄存器 (DWT_CPICNTR)

DWT CPI count register

偏移地址：0x008

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **CPICNT[7:0]**：CPI 计数器 (CPI counter)。对执行多周期指令（由 DWT_LSUCNTR 记录的多周期指令除外）所需的其他周期进行计数，并对任何指令读取停止 (Stall) 进行计数。

41.10.4 DWT 异常计数寄存器 (DWT_EXCCNTR)

DWT exception count register

偏移地址：0x00C

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **EXCCNT[7:0]**：异常开销周期计数器 (Exception overhead cycle counter)。对在异常处理中花费的周期数进行计数。

41.10.5 DWT 睡眠计数寄存器 (DWT_SLPCNTR)

DWT sleep count register

偏移地址：0x010

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **SLEEPCNT[7:0]**：睡眠周期计数器 (Sleep cycle counter)。对在睡眠模式（WFI、WFE、退出时睡眠）下消耗的周期数进行计数。

41.10.6 DWT LSU 计数寄存器 (DWT_LSUCNTR)

DWT LSU count register

偏移地址：0x014

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **LSUCNT[7:0]**：加载存储计数器 (Load store counter)。对执行加载和存储指令所需的其他周期进行计数。

41.10.7 DWT 折叠计数寄存器 (DWT_FOLDCNTR)

DWT fold count register

偏移地址：0x018

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **FOLDCNT[7:0]**：折叠指令计数器 (Folded instruction counter)。每出现一条只花费 0 周期的指令就递增计数。

41.10.8 DWT 程序计数器采样寄存器 (DWT_PCSR)

DWT program counter sample register

偏移地址：0x01C

复位值：0x0000 0000

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

位 31:0 **EIASAMPLE[31:0]**：执行指令地址采样值 (Executed Instruction Address sample value)。采样程序计数器的当前值。

41.10.9 DWT 比较器寄存器 x (DWT_COMPxR)

DWT comparator register x

偏移地址：0x020 + x * 0x10（适用于 x = 0 到 3）

复位值：0x0000 0000

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

位 31:0 **COMP[31:0]**：用于比较的参考值 (Reference value for comparison)

41.10.10 DWT 掩码寄存器 x (DWT_MASKxR)

DWT mask register x

偏移地址：0x024 + x * 0x10（适用于 x = 0 到 3）

复位值：0x0000 0000

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

位 31:5 保留，必须保持复位值。

位 4:0 **MASK[4:0]**：比较器掩码大小 (Comparator mask size)。为比较器 n 匹配的地址范围提供访问地址的忽略掩码的大小。调试器可写入 0b11111 到此位域，然后读回寄存器以确定支持的最大掩码大小。

41.10.11 DWT 功能寄存器 x (DWT_FUNCTxR)

DWT function register x

偏移地址：0x028 + x * 0x10（适用于 x = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	DATAVADDR1[3:0]			
							r					rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAVADDR0[3:0]				DATAVSIZE[1:0]		LINK1 ENA	DATAV MATCH	CYC MATCH	Res.	EMIT RANGE	Res.	FUNCTION[3:0]			
rw				rw		rw	rw	rw		rw		rw			

位 31:25 保留，必须保持复位值。

位 24 **MATCHED**：比较器匹配 (Comparator match)，只读值。指示自上次读取寄存器以来是否发生了比较器匹配。

0：不匹配

1：发生了匹配

位 23:20 保留，必须保持复位值。

位 19:16 **DATAVADDR1[3:0]**：当 DATAVMATCH 和 LNK1ENA 位均为 1 时，此位域可存储第二比较器的比较器编号以用于关联地址比较。

位 15:12 **DATAVADDR0[3:0]**：当 DATAVMATCH 和 LNK1ENA 位均为 1 时，此位域可存储比较器的比较器编号以用于关联地址比较。

位 11:10 **DATAVSIZE[1:0]**：对于数据值匹配，指定所需数据比较的大小。

0x0：字节

0x1：半字

0x2：字

0x3：保留

位 9 **LNK1ENA**：指示是否支持使用第二比较器，只读值。

0x1：支持

位 8 **DATAVMATCH**：使能周期比较。

0x0：执行地址比较

0x1：执行数据值比较

位 7 **CYCMATCH**：使能比较器 0 周期计数比较。对于其他比较器，此位域保留。

0x0：无周期计数比较

0x1：将 DWT_COMP0R 与周期计数器 DWT_CYCCNTR 进行比较

位 6 保留，必须保持复位值。

位 5 **EMITRANGE**：使能生成数据跟踪地址偏移数据包（包括数据地址位 0 到 15）

0x0：禁止

0x1：使能

位 4 保留，必须保持复位值。

位 3:0 **FUNCTION[3:0]**：选择发生比较器匹配时采取的操作。此位域的含义取决于 DATAVMATCH 和 CYCMATCH 位域的设置。请参见 *[5]*。

41.10.12 DWT CoreSight 外设标识寄存器 4 (DWT_PIDR4)

DWT CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：Arm® JEDEC 代码

41.10.13 DWT CoreSight 外设标识寄存器 0 (DWT_PIDR0)

DWT CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0002

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (Part number bits [7:0])
0x02：DWT 产品编号

41.10.14 DWT CoreSight 外设标识寄存器 1 (DWT_PIDR1)

DWT CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (Part number bits [11:8])
0x9：DWT 产品编号

41.10.15 DWT CoreSight 外设标识寄存器 2 (DWT_PIDR2)

DWT CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 003B

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x3：r0p4

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x3：Arm® JEDEC 代码

41.10.16 DWT CoreSight 外设标识寄存器 3 (DWT_PIDR3)

DWT CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

41.10.17 DWT CoreSight 组件标识寄存器 0 (DWT_CIDR0)

DWT CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (Component ID bits [7:0])
0x0D：公共 ID 值

41.10.18 DWT CoreSight 外设标识寄存器 1 (DWT_CIDR1)

DWT CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 00E0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (Component ID bits [15:12])——组件类
0xE：跟踪发生器组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (Component ID bits [11:8])
0x0：公共 ID 值

41.10.19 DWT CoreSight 组件标识寄存器 2 (DWT_CIDR2)

DWT CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (Component ID bits [23:16])
0x05：公共 ID 值

41.10.20 DWT CoreSight 组件标识寄存器 3 (DWT_CIDR3)

DWT CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (Component ID bits [31:24])
0xB1：公共 ID 值

41.10.21 CPU2 DWT 寄存器

CPU2 DWT 寄存器的 AHBD 地址范围为 0xE0001000 到 0xE0001FFC。

表 258. CPU2 DWT 寄存器映射和复位值

偏移	寄存器名称	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
0x000	DWT_CTRLR	NUMCOMP[3:0]				NOTRCPKT	NOEXTTRIG	NOCYCCNT	NOPRFCNT	Res.	CYCEVTENA	FOLDEVTENA	LSUEVTENA	SLEEPEVTENA	EXCEVTENA	CPIEVTENA	EXCTRCENA	Res.	Res.	Res.	PCSAMPLENA	SYNCTAP[1:0]		CYCTAP	POSINIT[3:0]				POSTPRESET[3:0]				CYCCNTENA
	Reset value	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	0
0x004	DWT_CYCCNTR	CYCCNT[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
0x008	DWT_CPICNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPICNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x00C	DWT_EXCCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXCCNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x010	DWT_SLPCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLEEPCNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x014	DWT_LSUCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSUCNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x018	DWT_FOLDCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FOLDCNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x01C	DWT_PCSR	EIASAMPLE[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
0x020	DWT_COMP0R	COMP[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
0x024	DWT_MASK0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASK[4:0]				
	Reset value																												0	0	0	0	0
0x028	DWT_FUNCT0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	DATAVADDR1[3:0]				DATAVADDR0[3:0]				DATAVSIZE[1:0]		LNK1ENA	DATAVMATCH	CYCMATCH	Res.	EMITRANGE	Res.	FUNCTION[3:0]			
	Reset value								0					0	0	0	0	0	0	0	0	0	0	0	0	0		0		0	0	0	0
0x030	DWT_COMP1R	COMP[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
0x034	DWT_MASK1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASK[4:0]				
	Reset value																												0	0	0	0	0

表 258. CPU2 DWT 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x038	DWT_FUNCT1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MATCHED	Res.	Res.	Res.	q	DATAVADDR1[3:0]				DATAVADDR0[3:0]				DATAVSIZE[1:0]		LNK1ENA	DATAVMATCH	CYCMATCH	Res.	EMITRANGE	Res.	FUNCTION[3:0]			
	Reset value								0					0	0	0	0	0	0	0	0	0	0	0	0	0		0		0	0	0	0
0x040	DWT_COMP2R	COMP[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
0x044	DWT_MASK2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASK[4:0]			
	Reset value																												0	0	0	0	0
0x048	DWT_FUNCT2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	DATAVADDR1[3:0]				DATAVADDR0[3:0]				DATAVSIZE[1:0]		LNK1ENA	DATAVMATCH	CYCMATCH	Res.	EMITRANGE	Res.	FUNCTION[3:0]			
	Reset value								0					0	0	0	0	0	0	0	0	0	0	0	0	0		0		0	0	0	0
0x050	DWT_COMP3R	COMP[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
0x054	DWT_MASK3R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASK[4:0]			
	Reset value																												0	0	0	0	0
0x058	DWT_FUNCT3R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	DATAVADDR1[3:0]				DATAVADDR0[3:0]				DATAVSIZE[1:0]		LNK1ENA	DATAVMATCH	CYCMATCH	Res.	EMITRANGE	Res.	FUNCTION[3:0]			
	Reset value								0					0	0	0	0	0	0	0	0	0	0	0	0	0		0		0	0	0	0
0xFD0	DWT_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFE0	DWT_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	0	0	0	0	1	0
0xFE4	DWT_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	0	0	0	0
0xFE8	DWT_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	1	1	1	0	1	1
0xFEC	DWT_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	DWT_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	DWT_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
	Reset value																									0	1	1	1	0	0	0	0

表 258. CPU2 DWT 寄存器映射和复位值（续）

偏移	寄存器名称	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
0xFF8	DWT_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	DWT_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

有关寄存器边界地址的信息，请参见*第 41.9 节：CPU2 ROM 表*。

41.11 CPU2 断点单元 (PBU)

BPU 允许设置硬件断点，其含有八个比较器，这些比较器监视指令读取地址，并在检测到匹配时返回断点指令。CPU2 PBU 不支持 Flash 补丁功能。

41.11.1 BPU 控制寄存器 (BPU_CTRLR)

BPU control register

偏移地址：0x000

复位值：0x0000 0080

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	NUM_CODE[6:4]			NUM_LIT[3:0]				NUM_CODE[3:0]				Res.	Res.	KEY	ENABLE
	r			r				r						rw	rw

位 31:15 保留，必须保持复位值。

位 14:12 **NUM_CODE[6:4]**：支持的指令地址比较器数量 (Number of instruction address comparators supported)——最低有效位，只读值。

0x0：支持 8 个指令比较器。

位 11:8 **NUM_LIT[3:0]**：支持的文字地址比较器数量 (Number of literal address comparators supported)，只读值。

0x0：无支持的文字比较器。

位 7:4 **NUM_CODE[3:0]**：支持的指令地址比较器数量 (Number of instruction address comparators supported)——最低有效位，只读值。

0x8：支持 8 个指令比较器

位 1 **KEY**：写保护密钥 (Write protect key)。如果此位未设置为 1，则忽略对 BPU_CTRLR 寄存器的写操作。

位 0 **ENABLE**：BPU 使能 (BPU enable)

0x0：禁止

0x1：使能

41.11.2 BPU 重映射寄存器 (BPU_REMAPR)

BPU remap register

偏移地址：0x004

复位值：0x0000 0000

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

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

位 31:30 保留，必须保持复位值。

位 29 **RMPSPT**：指示是否支持 Flash 补丁重映射，只读值。
0x0：不支持重映射。

位 28:0 保留，必须保持复位值。

41.11.3 BPU 比较器寄存器 (BPU_COMPxR)

BPU comparator registers

偏移地址：0x008 + x * 0x4（适用于 x = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REPLACE[1:0]		Res.	COMP[26:14]												
rw			rw												

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

位 31:30 **REPLACE[1:0]**：定义 COMP 位域和指令读取地址之间发生匹配时的行为。
0x0：预留
0x1：断点在低位半字上，高位半字不受影响。
0x2：断点在高位半字上，低位半字不受影响。
0x3：断点在高位半字和低位半字上。

位 29 保留，必须保持复位值。

位 28:2 **COMP[26:0]**：与指令代码存储器（0x00000000 到 0x1FFFFFFF）的访问地址位 28:2 进行比较的值。如果发生匹配，则由 REPLACE 位域定义要采取的操作。

位 1 保留，必须保持复位值。

位 0 **ENABLE**：比较器使能 (Comparator enable)。仅当此位与 BPU_CTRLR 寄存器中的 BPU ENABLE 位均置 1 时，比较器才使能。
0：禁止
1：使能

41.11.4 BPU CoreSight 外设标识寄存器 4 (BPU_PIDR4)

BPU CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：Arm® JEDEC 代码

41.11.5 BPU CoreSight 外设标识寄存器 0 (BPU_PIDR0)

BPU CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 000C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (Part number bits [7:0])
0x0C：BPU 产品编号

41.11.6 BPU CoreSight 外设标识寄存器 1 (BPU_PIDR1)

BPU CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (Part number bits [11:8])
0x0：BPU 产品编号

41.11.7 BPU CoreSight 外设标识寄存器 2 (BPU_PIDR2)

BPU CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 002B

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x2：r0p3

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x3：Arm® JEDEC 代码

41.11.8 BPU CoreSight 外设标识寄存器 3 (BPU_PIDR3)

BPU CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

41.11.9 BPU CoreSight 组件标识寄存器 0 (BPU_CIDR0)

BPU CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (Component ID bits [7:0])
0x0D：公共 ID 值

41.11.10 BPU CoreSight 外设标识寄存器 1 (BPU_CIDR1)

BPU CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 00E0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (Component ID bits [15:12])——组件类
0xE：跟踪发生器组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (Component ID bits [11:8])
0x0：公共 ID 值

41.11.11 BPU CoreSight 组件标识寄存器 2 (BPU_CIDR2)

BPU CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (Component ID bits [23:16])
0x05：公共 ID 值

41.11.12 BPU CoreSight 组件标识寄存器 3 (BPU_CIDR3)

BPU CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (Component ID bits [31:24])

0xB1：公共 ID 值

41.11.13 CPU2 BPU 寄存器映射和复位值

CPU2 BPU 寄存器的地址范围为 0xE0002000 到 0xE0002FFC。

表 259. CPU2 BPU 寄存器映射和复位值

偏移	寄存器名称	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
0x000	BPU_CTRLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NUM_CODE[6:4]			NUM_LIT[3:0]				NUM_CODE[3:0]				Res.	Res.	KEY	ENABLE
	Reset value																		0	0	0	0	0	0	0	1	0	0	0			0	0
0x004	BPU_REMAPR	Res.	Res.	RMPSPT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value			0																													
0x008 to 0x024	BPU_COMP0-7R	REPLACE[1:0]		Res.	COMP[26:0]																											Res.	ENABLE
	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
0xFD0	BPU_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFE0	BPU_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	0	0	1	1	0	0
0xFE4	BPU_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	0	0	0	0
0xFE8	BPU_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	1	0	1	0	1	1
0xFEC	BPU_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	BPU_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	BPU_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									1	1	1	0	0	0	0	0
0xFF8	BPU_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	BPU_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

有关寄存器边界地址的信息，请参见*第 41.9 节：CPU2 ROM 表*。

41.12 CPU2 交叉触发接口 (CTI)

请参见*第 41.6 节*。

41.13 CPU1 ROM 表

ROM 表是一款 CoreSight™ 组件，包含可通过 AHB-AP 访问的所有 Coresight™ 调试组件的基址。这些表允许调试器自动发现 CoreSight 系统的拓扑。

CPU1 子系统有一个 ROM 表。该表由 CPU1 AHB-AP 中的 BASE 寄存器指向。其包含系统控制空间寄存器的基址指针（允许调试器识别 CPU 内核），以及 FPB、DWT、ITM、ETM 和 CTI 的基址指针。

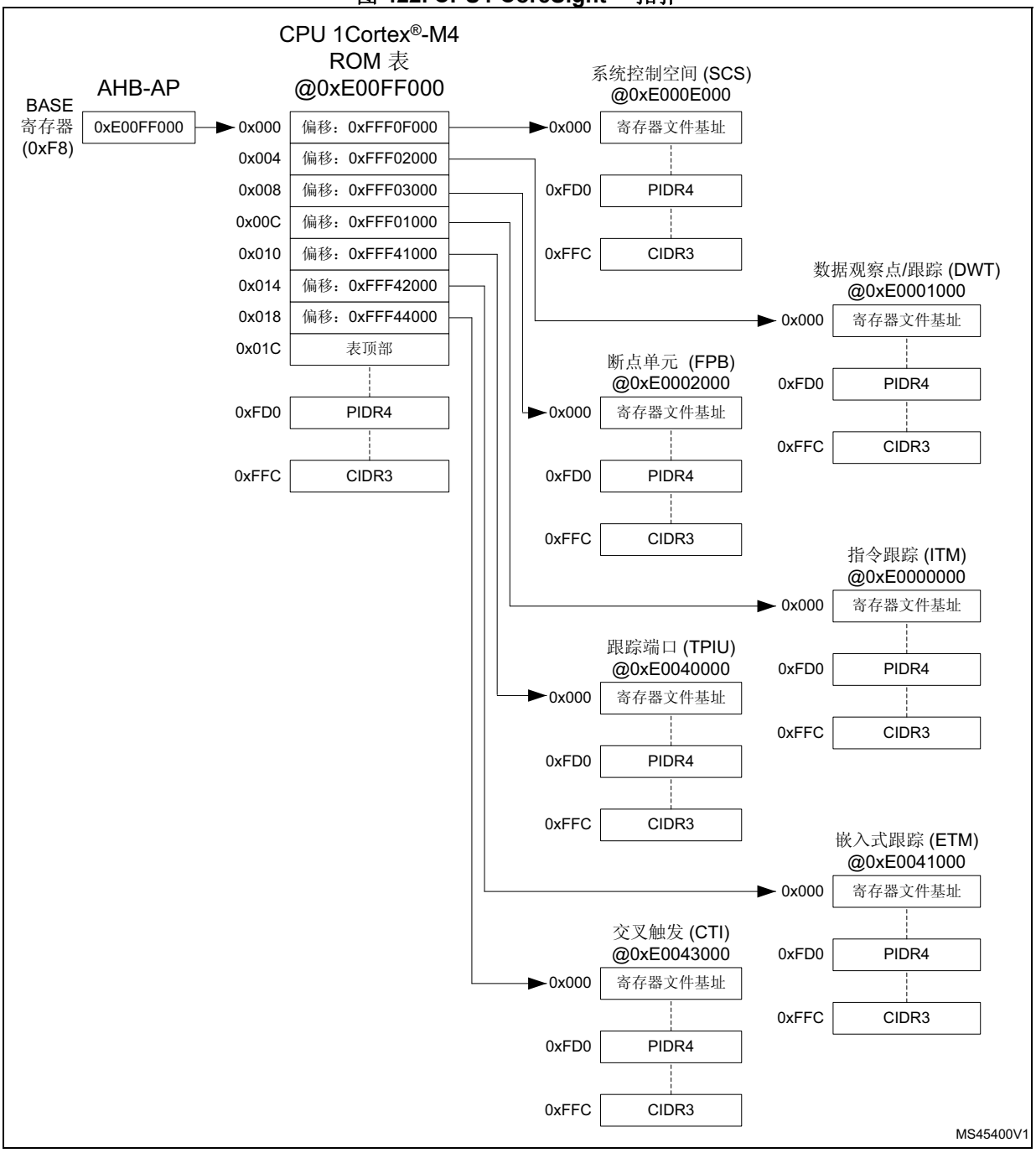
CPU1 ROM 表（请参见*表 260*）占用一个 4 KB、32 位宽的地址空间块，地址范围从 0xE00FF000 到 0xE00FFFFC。

表 260. CPU1 ROM 表

地址（ROM 表中）	组件名称	组件基址	组件偏移地址	大小	条目
0xE00FF000	SCS	0xE000E000	0xFFF0F000	4 KB	0xFFF0F003
0xE00FF004	DWT	0xE0001000	0xFFF02000	4 KB	0xFFF02003
0xE00FF008	FPB	0xE0002000	0xFFF03000	4 KB	0xFFF03003
0xE00FF00C	ITM	0xE0000000	0xFFF01000	4 KB	0xFFF01003
0xE00FF010	TPIU	0xE0040000	0xFFF41000	4 KB	0xFFF41003
0xE00FF014	ETM	0xE0041000	0xFFF42000	4 KB	0xFFF42003
0xE00FF018	CTI	0xE0043000	0xFFF44000	4 KB	0xFFF44003
0xE00FF01C	表顶部	-	-	-	0x00000000
0xE00FF020 到 0xE00FFFC8	保留	-	-	-	0x00000000
0xE00FFFCC 到 0xE00FFFFC	ROM 表寄存器	-	-	-	请参见*表 256*

CPU1 子系统的 CoreSight™ 组件拓扑如*图 422* 所示。

图 422. CPU1 CoreSight™ 拓扑



41.13.1 CPU1 ROM 存储器类型寄存器 (C1ROM_MEMTYPER)

CPU1 ROM memory type register

偏移地址：0xFCC

复位值：0x0000 0001

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

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

位 31:1 保留，必须保持复位值。

位 0 **SYSMEM**：系统存储器 (System memory)
0x1：该总线上有系统存储器

41.13.2 CPU1 ROM CoreSight 外设标识寄存器 4 (C1ROM_PIDR4)

CPU1 ROM CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x0：意法半导体 JEDEC 连续代码

41.13.3 CPU1 ROM CoreSight 外设标识寄存器 0 (C1ROM_PIDR0)

CPU1 ROM CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0095

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (Part number bits [7:0])
0x95：STM32WB55xx

41.13.4 CPU1 ROM CoreSight 外设标识寄存器 1 (C1ROM_PIDR1)

CPU1 ROM CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 0004

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0x0：意法半导体 JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (Part number bits [11:8])
0x4：STM32WB55xx

41.13.5 CPU1 ROM CoreSight 外设标识寄存器 2 (C1ROM_PIDR2)

CPU1 ROM CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 000A

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x0：rev r0p0

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x2：意法半导体 JEDEC 代码

41.13.6 CPU1 ROM CoreSight 外设标识寄存器 3 (C1ROM_PIDR3)

CPU1 ROM CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

41.13.7 CPU1 ROM CoreSight 组件标识寄存器 0 (C1ROM_CIDR0)

CPU1 ROM CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (Component ID bits [7:0])
0x0D：公共 ID 值

41.13.8 CPU1 ROM CoreSight 外设标识寄存器 1 (C1ROM_CIDR1)

CPU1 ROM CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 0010

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (Component ID bits [15:12])——组件类
0x1：ROM 表组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (Component ID bits [11:8])
0x0：公共 ID 值

41.13.9 CPU1 ROM CoreSight 组件标识寄存器 2 (C1ROM_CIDR2)

CPU1 ROM CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (Component ID bits [23:16])
0x05：公共 ID 值

41.13.10 CPU1 ROM CoreSight 组件标识寄存器 3 (C1ROM_CIDR3)

CPU1 ROM CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (Component ID bits [31:24])
0xB1：公共 ID 值

41.13.11 CPU1 ROM 表寄存器映射和复位值

表 261. CPU1 ROM 表寄存器映射和复位值

偏移	寄存器名称	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
0xFCC	C1ROM_ MEMTYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSMEM
	Reset value																																1
0xFD0	C1ROM_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFE0	C1ROM_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									1	0	0	1	0	1	0	1
0xFE4	C1ROM_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									0	0	0	0	0	1	0	0
0xFE8	C1ROM_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	0	1	0	1	0
0xFEC	C1ROM_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	C1ROM_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	C1ROM_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									0	0	0	1	0	0	0	0
0xFF8	C1ROM_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	C1ROM_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

有关寄存器边界地址的信息，请参见*第 41.13 节：CPU1 ROM 表*。

41.14 CPU1 数据观察点和跟踪单元 (DWT)

DWT 提供了四个比较器，可用作：

- 观察点
- ETM 触发器
- PC 采样触发器
- 数据地址采样触发器
- 数据比较器（仅适用于比较器 1）
- 时钟周期计数器比较器（仅适用于比较器 0）

还包含以下计数器：

- 时钟周期
- 分支指令
- 加载存储单元 (LSU) 操作
- 睡眠周期
- 每条指令周期数
- 中断开销

DWT 比较器将其 DWT_COMPR 寄存器中保存的值与下列其中一项进行比较：

- 数据地址
- 指令地址
- 数据值
- 周期计数值（仅适用于比较器 0）

比较器可以使用掩码进行地址匹配，因此可匹配一系列地址。

一旦匹配成功时，比较器会生成以下其中一项：

- 一个或多个 DWT 数据跟踪数据包，包括一个或多个：
  - 引发数据访问的指令的地址
  - 偏移地址，数据访问地址的位 [15:0]
  - 匹配的数据值
- 观察点调试事件（在 PC 值上或者在被访问的数据地址上）。
- 指示匹配超出 DWT 单元的 CMPMATCH[N] 事件。

观察点调试事件会生成 DebugMonitor 异常，或者导致处理器停止执行并进入调试状态。

有关如何使用 DWT 的更多详细信息，请参见《Arm®v7-M 架构参考手册》*[5]*。

41.14.1 DWT 控制寄存器 (DWT_CTRLR)

DWT control register (DWT_CTRLR)

偏移地址：0x000

复位值：0x4000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMCOMP[3:0].				NOTRCPKT	NOEXTTRIG	NOCYCCNT	NOPRFCNT	Res.	CYCEVTENA	FOLDEVTENA	LSUEVTENA	SLEEPEVTENA	EXCEVTENA	CPIEVTENA	EXCTRCENA
r				r	r	r	r		rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	PCSAMPLENA	SYNCTAP[1:0]		CYCTAP	POSTINIT[3:0]				POSTRESET[3:0]				CYCCNTENA
			rw	rw		rw	rw				rw				rw

位 31:28 **NUMCOMP**：设计的比较器数量 (Number of comparators implemented)，只读值

0x4：四个比较器

位 27 **NOTRCPKT**：支持跟踪采样和异常跟踪 (Trace sampling and exception tracing support)，只读值

0x0：支持

位 26 **NOEXTTRIG**：外部信号匹配 CMPMATCH 支持 (External match signal, CMPMATCH support)，只读值

0x0：支持

位 25 **NOCYCCNT**：循环计数器支持 (Cycle counter support)，只读值

0x0：支持

位 24 **NOPRFCNT**：性能分析计数器支持 (Profiling counter support)，只读值

0x0：支持

位 23 保留，必须保持复位值。

位 22 **CYCEVTENA**：使能 POSTCNT 下溢事件计数器数据包生成 (Enable for POSTCNT underflow event counter packet generation)

0x0：禁止

0x1：使能

位 21 **FOLDEVTENA**：使能折叠指令计数器上溢事件生成 (Enable for folded instruction counter overflow event generation)

0x0：禁止

0x1：使能

位 20 **LSUEVTENA**：使能 LSU 计数器上溢事件生成 (Enable for LSU counter overflow event generation)

0x0：禁止

0x1：使能

位 19 **SLEEPEVTENA**：使能睡眠计数器上溢事件生成 (Enable for sleep counter overflow event generation)

0x0：禁止

0x1：使能

位 18 **EXCEVTENA**：使能异常开销计数器上溢事件生成 (Enable for exception overhead counter overflow event generation)

0x0：禁止

0x1：使能

位 17 **CPIEVTENA**：使能 CPI 计数器上溢事件生成 (Enable for CPI counter overflow event generation)
0x0：禁止
0x1：使能

位 16 **EXCTRCENA**：使能异常跟踪生成 (Enable for exception trace generation)
0x0：禁止
0x1：使能

位 15:13 保留，必须保持复位值。

位 12 **PCSAMPLENA**：使能将 POSTCNT 计数器用作定时器，用于定期生成 PC 样本数据包。
0x0：禁止
0x1：使能

位 11:10 **SYNCTAP[1:0]**：选择同步数据包计数器点击 CYCCNT 计数器的位置。此位域确定同步数据包速率。
0x0：禁止。无同步数据包
0x1：点击位置为 CYCCNT[24]
0x2：点击位置为 CYCCNT[26]
0x3：点击位置为 CYCCNT[28]

位 9 **CYCTAP**：选择 POSTCNT 点击 CYCCNT 计数器的位置。
0x0：点击位置为 CYCCNT[6]
0x1：点击位置为 CYCCNT[10]

位 8:5 **POSTINIT[3:0]**：POSTCNT 计数器初始值 (Initial value of the POSTCNT counter)。如果使能了 POSTCNT 计数器，则忽略对此位域的写操作（即，在写入 POSTINIT 之前 CYCEVTENA 或 PCSAMPLENA 必须复位）。

位 4:1 **POSTPRESET[3:0]**：POSTCNT 计数器的重载值 (Reload value of the POSTCNT counter)

位 0 **CYCCNTENA**：使能 CYCCNT 计数器。
0x0：禁止
0x1：使能

41.14.2 DWT 周期计数寄存器 (DWT_CYCCNTR)

DWT cycle count register

偏移地址：0x004

复位值：0x0000 0000

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

位 31:0 **CYCCNT[31:0]**：处理器时钟周期计数器 (Processor clock cycle counter)

41.14.3 DWT CPI 计数寄存器 (DWT_CPICNTR)

DWT CPI count register

偏移地址：0x008

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **CPICNT[7:0]**：CPI 计数器 (CPI counter)。对执行多周期指令（由 DWT_LSUCNTR 记录的多周期指令除外）所需的其他周期进行计数，并对任何指令读取停止 (Stall) 进行计数。

41.14.4 DWT 异常计数寄存器 (DWT_EXCCNTR)

DWT exception count register

偏移地址：0x00C

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **EXCCNT[7:0]**：异常开销周期计数器 (Exception overhead cycle counter)。对在异常处理中花费的周期数进行计数。

41.14.5 DWT 睡眠计数寄存器 (DWT_SLPCNTR)

DWT sleep count register

偏移地址：0x010

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **SLEEPCNT[7:0]**：睡眠周期计数器 (Sleep cycle counter)。对在睡眠模式（WFI、WFE、退出时睡眠）下消耗的周期数进行计数。

41.14.6 DWT LSU 计数寄存器 (DWT_LSUCNTR)

DWT LSU count register

偏移地址：0x014

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **LSUCNT[7:0]**：加载存储计数器 (Load store counter)。对执行加载和存储指令所需的其他周期进行计数。

41.14.7 DWT 折叠计数寄存器 (DWT_FOLDCNTR)

DWT fold count register

偏移地址：0x018

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **FOLDCNT[7:0]**：折叠指令计数器 (Folded instruction counter)。每出现一条只花费 0 周期的指令就递增计数。

41.14.8 DWT 程序计数器采样寄存器 (DWT_PCSR)

DWT program counter sample register

偏移地址：0x01C

复位值：0x0000 0000

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

位 31:0 **EIASAMPLE[31:0]**：执行指令地址采样值 (Executed Instruction Address sample value)。采样程序计数器的当前值。

41.14.9 DWT 比较器寄存器 x (DWT_COMPxR)

DWT comparator register x

偏移地址：0x020 + x * 0x10（适用于 x = 0 到 3）

复位值：0x0000 0000

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

位 31:0 **COMP[31:0]**：用于比较的参考值 (Reference value for comparison)

41.14.10 DWT 掩码寄存器 x (DWT_MASKxR)

DWT mask register x

偏移地址：0x024 + x * 0x10（适用于 x = 0 到 3）

复位值：0x0000 0000

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

位 31:5 保留，必须保持复位值。

位 4:0 **MASK[4:0]**：比较器掩码大小 (Comparator mask size)。为比较器 n 匹配的地址范围提供访问地址的忽略掩码的大小。调试器可写入 0b11111 到此位域，然后读回寄存器以确定支持的最大掩码大小。

41.14.11 DWT 功能寄存器 x (DWT_FUNCTxR)

DWT function register x

偏移地址：0x028 + x * 0x10（适用于 x = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATAVADDR1[3:0]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAVADDR0[3:0]				DATAVSIZE[1:0]		LINK1 ENA	DATAV MATCH	CYC MATCH	Res.	EMIT RANGE	Res.	FUNCTION[3:0]			
rw				rw		rw	rw	rw		rw		rw			

位 31:25 保留，必须保持复位值。

位 24 **MATCHED**：比较器匹配 (Comparator match)，只读值。指示自上次读取寄存器以来是否发生了比较器匹配。

0：不匹配。

1：发生了匹配

位 23:20 保留，必须保持复位值。

位 19:16 **DATAVADDR1[3:0]**：当 DATAVMATCH 和 LNK1ENA 位均为 1 时，此位域可存储第二比较器的比较器编号以用于关联地址比较。

位 15:12 **DATAVADDR0[3:0]**：当 DATAVMATCH 和 LNK1ENA 位均为 1 时，此位域可存储比较器的比较器编号以用于关联地址比较。

位 11:10 **DATAVSIZE[1:0]**：对于数据值匹配，指定所需数据比较的大小。

0x0：字节

0x1：半字

0x2：字

0x3：保留

位 9 **LNK1ENA**：指示是否支持使用第二比较器，只读值。

0x1：支持

位 8 **DATAVMATCH**：使能周期比较。

0x0：执行地址比较

0x1：执行数据值比较

位 7 **CYCMATCH**：使能比较器 0 周期计数比较。对于其他比较器，此位域保留。

0x0：无周期计数比较

0x1：将 DWT_COMP0R 与周期计数器 DWT_CYCCNTR 进行比较

位 6 保留，必须保持复位值。

位 5 **EMITRANGE**：使能生成数据跟踪地址偏移数据包（包括数据地址位 0 到 15）

0x0：禁止

0x1：使能

位 4 保留，必须保持复位值。

位 3:0 **FUNCTION[3:0]**：选择发生比较器匹配时采取的操作。此位域的含义取决于 DATAVMATCH 和 CYCMATCH 位域的设置。请参见 *[5]*。

41.14.12 DWT CoreSight 外设标识寄存器 4 (DWT_PIDR4)

DWT CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：Arm® JEDEC 代码

41.14.13 DWT CoreSight 外设标识寄存器 0 (DWT_PIDR0)

DWT CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0002

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (Part number bits [7:0])
0x02：DWT 产品编号

41.14.14 DWT CoreSight 外设标识寄存器 1 (DWT_PIDR1)

DWT CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (Part number bits [11:8])
0x9：DWT 产品编号

41.14.15 DWT CoreSight 外设标识寄存器 2 (DWT_PIDR2)

DWT CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 003B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x3：r0p4

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x3：Arm® JEDEC 代码

41.14.16 DWT CoreSight 外设标识寄存器 3 (DWT_PIDR3)

DWT CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

41.14.17 DWT CoreSight 组件标识寄存器 0 (DWT_CIDR0)

DWT CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (Component ID bits [7:0])
0x0D：公共 ID 值

41.14.18 DWT CoreSight 外设标识寄存器 1 (DWT_CIDR1)

DWT CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 00E0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (Component ID bits [15:12])——组件类
0xE：跟踪发生器组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (Component ID bits [11:8])
0x0：公共 ID 值

41.14.19 DWT CoreSight 组件标识寄存器 2 (DWT_CIDR2)

DWT CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (Component ID bits [23:16])
0x05：公共 ID 值

41.14.20 DWT CoreSight 组件标识寄存器 3 (DWT_CIDR3)

DWT CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (Component ID bits [31:24])
0xB1：公共 ID 值

41.14.21 CPU1 DWT 寄存器映射和复位值

CPU1 DWT 的地址范围为 0xE0001000 到 0xE0001FFC。

表 262. CPU1 DWT 寄存器映射和复位值

偏移	寄存器名称	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
0x000	DWT_CTRLR	NUMCOMP[3:0]				NOTRCPKT	NOEXTTRIG	NOCYCCNT	NOPRFCNT	Res.	CYCEVTENA	FOLDEVTENA	LSUEVTENA	SLEEPEVTENA	EXCEVTENA	CPIEVTENA	EXCTRCENA	Res.	Res.	Res.	PCSAMPLENA	SYNCTAP[1:0]		CYCTAP	POSINIT[3:0]				POSTPRESET[3:0]				CYCCNTENA
	Reset value	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	0
0x004	DWT_CYCCNTR	CYCCNT[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
0x008	DWT_CPICNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPICNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x00C	DWT_EXCCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXCCNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x010	DWT_SLPCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLEEPCNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x014	DWT_LSUCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSUCNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x018	DWT_FOLDCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FOLDCNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x01C	DWT_PCSR	EIASAMPLE[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
0x020	DWT_COMP0R	COMP[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
0x024	DWT_MASK0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASK[4:0]				
	Reset value																												0	0	0	0	0
0x028	DWT_FUNCT0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	DATAVADDR1[3:0]				DATAVADDR0[3:0]				DATAVSIZE[1:0]		LNK1ENA	DATAVMATCH	CYCMATCH	Res.	EMITRANGE	Res.	FUNCTION[3:0]			
	Reset value								0					0	0	0	0	0	0	0	0	0	0	0	0	0		0		0	0	0	0
0x030	DWT_COMP1R	COMP[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
0x034	DWT_MASK1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASK[4:0]				
	Reset value																												0	0	0	0	0
0x038	DWT_FUNCT1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	DATAVADDR1[3:0]				DATAVADDR0[3:0]				DATAVSIZE[1:0]		LNK1ENA	DATAVMATCH	CYCMATCH	Res.	EMITRANGE	Res.	FUNCTION[3:0]			
	Reset value								0					0	0	0	0	0	0	0	0	0	0	0	0	0		0		0	0	0	0

表 262. CPU1 DWT 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x040	DWT_COMP2R	COMP[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
0x044	DWT_MASK2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASK[4:0]				
	Reset value																												0	0	0	0	0
0x048	DWT_FUNCT2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	DATAVADDR1[3:0]				DATAVADDR0[3:0]				DATAVSIZE[1:0]		LNK1ENA	DATAVMATCH	CYCMATCH	Res.	EMITRANGE	Res.	FUNCTION[3:0]			
	Reset value								0					0	0	0	0	0	0	0	0	0	0	0	0	0		0		0	0	0	0
0x050	DWT_COMP3R	COMP[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
0x054	DWT_MASK3R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASK[4:0]				
	Reset value																												0	0	0	0	0
0x058	DWT_FUNCT3R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	DATAVADDR1 [3:0]				DATAVADDR0 [3:0]				DATAVSIZE [1:0]		LNK1ENA	DATAVMATCH	CYCMATCH	Res.	EMITRANGE	Res.	FUNCTION [3:0]			
	Reset value								0					0	0	0	0	0	0	0	0	0	0	0	0	0		0		0	0	0	0
0xFD0	DWT_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFE0	DWT_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	0	0	0	0	1	0
0xFE4	DWT_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	0	0	0	0
0xFE8	DWT_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	1	1	1	0	1	1
0xFEC	DWT_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	DWT_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	DWT_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									1	1	1	0	0	0	0	0
0xFF8	DWT_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	DWT_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

有关寄存器边界地址的信息，请参见*第 41.13 节：CPU1 ROM 表*。

41.15 CPU1 指令跟踪宏单元 (ITM)

ITM 以数据包形式生成跟踪信息，有四个源可生成数据包。如果多个源同时生成数据包，则 ITM 仲裁数据包输出的顺序。按照优先级降序排列，四个源依次为：

1. 软件跟踪

   软件可直接写入任意 32 x 32 位 ITM 激励寄存器以生成数据包。可以对每个端口的权限级别进行编程。当软件写入使能的激励端口时，ITM 将端口标识、写访问的大小和被写入的数据合并成其写入 FIFO 的数据包。ITM 将 FIFO 中的数据包输出到跟踪总线上。读取激励端口寄存器会返回激励寄存器的状态（空或挂起），通过位 0 指示。

2. 硬件跟踪

   DWT 生成跟踪数据包以响应数据跟踪事件、PC 采样或性能分析计数器回卷。ITM 将这些数据包输出在跟踪总线上。

3. 局域时间戳

   ITM 包含一个 21 位计数器，由（预分频的）处理器时钟提供时钟信号。计数器值以时间戳数据包的形式输出到跟踪总线上。每生成一个时间戳数据包，计数器就复位为零。因此，时间戳指示自从上一个时间戳数据包以来经过的时间。

41.15.1 ITM 激励寄存器 x (ITM_STIMRx)

ITM stimulus register x

偏移地址：0x000 + x * 0x4（x = 0 到 31）

复位值：未知

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

位 31:0 **STIMULUS[31:0]**：写数据以软件事件数据包的形式输出在跟踪总线上。读取时，位 0 为 FIFOREADY 指示符：

0：激励端口缓冲区已满（或端口被禁止）

1：激励端口可接受新的写数据

41.15.2 ITM 跟踪使能寄存器 (ITM_TER)

ITM trace enable register

偏移地址：0x080

复位值：0x00000000

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

位 31:0 **STIMENA[31:0]**：每一位 n (0:31) 使能与 ITM_STIMRn 寄存器关联的激励端口。

0：禁止端口

1：使能端口

41.15.3 ITM 跟踪特权寄存器 (ITM_TPR)

ITM trace privilege register

偏移地址：0xE00

复位值：0x00000000

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

位 31:0 **PRIVMASK[31:0]**：使能非特权访问 ITM 激励端口 (Enable unprivileged access to ITM stimulus ports)。每一位控制八个激励端口。

0bXXX0：允许非特权访问端口 0 到 7

0bXXX1：仅允许特权访问端口 0 到 7

0bXX0X：允许非特权访问端口 8 到 15

0bXX1X：仅允许特权访问端口 8 到 15

0bX0XX：允许非特权访问端口 16 到 23

0bX1XX：仅允许特权访问端口 16 到 23

0b0XXX：允许非特权访问端口 24 到 31

0b1XXX：仅允许特权访问端口 24 到 31

41.15.4 ITM 跟踪控制寄存器 (ITM_TCR)

ITM trace control register

偏移地址：0xE80

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	TRACEBUSID[6:0]						
								rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TSPRESCALE[1:0]		Res.	Res.	Res.	SWOENA	TXENA	SYNCENA	TSENA	ITMENA
						rw					r	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23 **BUSY**：指示 ITM 当前是否正在处理事件，只读值。

0：不忙

1：忙

位 22:16 **TRACEBUSID[6:0]**：多源跟踪流格式化标识符 (Identifier for multi-source trace stream formatting)。如果使用了多源跟踪，则调试器必须将一个非零值写入此位域。注意：系统中每个跟踪源必须使用不同的 ID。

位 15:10 保留，必须保持复位值。

位 9:8 **TSPRESCALE[1:0]**：局部时间戳预分频器 (Local timestamp prescaler)，与跟踪数据包参考时钟搭配使用。可能的值为：

0x0：无预分频。

0x1：4 分频。

0x2：16 分频。

0x3：64 分频。

位 7:5 保留，必须保持复位值。

位 4 **SWOENA**：使能时间戳计数器异步时钟，只读值。

0：时间戳计数器使用处理器时钟

位 3 **TXENA**：使能将硬件事件数据包从 DWT 单元转发到跟踪端口。

0：禁止

1：使能

位 2 **SYNCENA**：使能同步数据包传输。

注：　如果调试器将此位置 1，则还必须配置 DWT 中 DWT_CTRLR 寄存器 SYNCTAP 位域，以获得正确的同步速度。

0：禁止

1：使能

位 1 **TSENA**：使能局部时间戳生成。

0：禁止

1：使能

位 0 **ITMENA**：使能 ITM。

0：禁止

1：使能

41.15.5 ITM CoreSight 外设标识寄存器 4 (ITM_PIDR4)

ITM CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：Arm® JEDEC 代码

41.15.6 ITM CoreSight 外设标识寄存器 0 (ITM_PIDR0)

ITM CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0001

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (Part number bits [7:0])
0x01：ITM 产品编号

41.15.7 ITM CoreSight 外设标识寄存器 1 (ITM_PIDR1)

ITM CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (Part number bits [11:8])
0x0：ITM 产品编号

41.15.8 ITM CoreSight 外设标识寄存器 2 (ITM_PIDR2)

ITM CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 003B

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x3：r0p4

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x3：Arm® JEDEC 代码

41.15.9 ITM CoreSight 外设标识寄存器 3 (ITM_PIDR3)

ITM CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

41.15.10 ITM CoreSight 组件标识寄存器 0 (ITM_CIDR0)

ITM CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (Component ID bits [7:0])
0x0D：公共 ID 值

41.15.11 ITM CoreSight 外设标识寄存器 1 (ITM_CIDR1)

ITM CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 00E0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (Component ID bits [15:12])——组件类
0xE：跟踪发生器组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (Component ID bits [11:8])
0x0：公共 ID 值

41.15.12 ITM CoreSight 组件标识寄存器 2 (ITM_CIDR2)

ITM CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (Component ID bits [23:16])
0x05：公共 ID 值

41.15.13 ITM CoreSight 组件标识寄存器 3 (ITM_CIDR3)

ITM CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (Component ID bits [31:24])
0xB1：公共 ID 值

41.15.14 ITM 寄存器映射和复位值

ITM 寄存器的地址范围为 0xE0000000 到 0xE0000FFC。

表 263. CPU1 ITM 寄存器映射和复位值

偏移	寄存器名称	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
0x000 to 0x07C	ITM_STIM0-31R	STIMULUS[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
0x0E00	ITM_TER	STIMENA[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
0x0E40	ITM_TPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIVMASK[3:0]			
	Reset value																													0	0	0	0
0xE80	ITM_TCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	TRACEBUSID[6:0]							Res.	Res.	Res.	Res.	GTSFREQ[1:0]		TSPRESCALE[1:0]		Res.	Res.	Res.	SWOENA	TXENA	SYNCENA	TSENA	ITMENA
	Reset value									0	0	0	0	0	0	0	0					0	0	0	0				0	0	0	0	0
0xFD0	ITM_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFE0	ITM_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	0	0	0	0	0	1
0xFE4	ITM_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	0	0	0	0
0xFE8	ITM_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	1	1	1	0	1	1
0xFEC	ITM_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	ITM_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	ITM_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									1	1	1	0	0	0	0	0
0xFF8	ITM_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	ITM_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

有关寄存器边界地址的信息，请参见*第 41.13 节：CPU1 ROM 表*。

41.16 CPU1 断点单元 (FPB)

FPB 允许用户设置硬件断点。其包含六个用于监视指令读取地址的比较器和两个文字地址比较器。如果发生匹配，则将地址重新映射到系统存储器中的地址（定义为 FPB_REMAPR 寄存器加上与匹配比较器对应的偏移量）。或者，可将指令比较器配置为生成断点指令。

41.16.1 FPB 控制寄存器 (FPB_CTRLR)

FPB control register

偏移地址：0x000

复位值：0x0000 0260

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	NUM_CODE[6:4]			NUM_LIT[3:0]				NUM_CODE[3:0]				Res.	Res.	KEY	ENABLE
	r			r				r						rw	rw

位 31:15 保留，必须保持复位值。

位 14:12 **NUM_CODE[6:4]**：支持的指令地址比较器数量 (Number of instruction address comparators supported)——最低有效位，只读值。

0x0：支持 6 个指令比较器。

位 11:8 **NUM_LIT[3:0]**：支持的文字地址比较器数量 (Number of literal address comparators supported)，只读值。

0x2：支持两个文字比较器。

位 7:4 **NUM_CODE[3:0]**：支持的指令地址比较器数量 (Number of instruction address comparators supported)——最低有效位，只读值。

0x6：支持 6 个指令比较器

位 1 **KEY**：写保护密钥 (Write protect key)。如果此位未设置为 1，则忽略对 FPB_CTRLR 寄存器的写操作。

位 0 **ENABLE**：FPB 使能 (FPB enable)

0x0：禁止

0x1：使能

41.16.2 FPB 重映射寄存器 (FPB_REMAPR)

FPB remap register

偏移地址：0x004

复位值：0x2000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	RMPSPT	REMAP[23:11]												
		r	rw												

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

位 31:30 保留，必须保持复位值。

位 29 **RMPSPT**：指示是否支持 Flash 补丁重映射，只读值。
0x1：支持重映射。

位 28:5 **REMAP[23:0]**：重映射目标地址 (Remap target address)。SRAM 中基址的位 [28:5]（FPB 向其重映射地址）。重映射基址必须与支持实现的比较器所需的字数（即 (NUM_CODE+NUM_LIT) 个字，至少 8 个字）对齐。由于重映射范围为 SRAM 存储区 0x20000000-0x3FFFFFFF，因此重映射地址的位 [31:29] 为 0b001。

位 4:0 保留，必须保持复位值。

41.16.3 FPB 比较器寄存器 (FPB_COMPxR)

FPB comparator registers

偏移地址：0x008 + x * 0x4（适用于 x = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REPLACE[1:0]		Res.	COMP[26:14]												
rw			rw												

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

位 31:30 **REPLACE[1:0]**：定义 COMP 位域和指令读取地址之间发生匹配时的行为。
0x0：预留
0x1：断点在低位半字上，高位半字不受影响。
0x2：断点在高位半字上，低位半字不受影响。
0x3：断点在高位半字和低位半字上。

位 29 保留，必须保持复位值。

位 28:2 **COMP[26:0]**：与指令代码存储器（0x00000000 到 0x1FFFFFFF）的访问地址位 28:2 进行比较的值。如果发生匹配，则由 REPLACE 位域定义要采取的操作。

位 1 保留，必须保持复位值。

位 0 **ENABLE**：比较器使能 (Comparator enable)。仅当此位与 FPB_CTRLR 寄存器中的 FPB ENABLE 位均置 1 时，比较器才使能。
0：禁止
1：使能

41.16.4 FPB CoreSight 外设标识寄存器 4 (FPB_PIDR4)

FPB CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：Arm® JEDEC 代码

41.16.5 FPB CoreSight 外设标识寄存器 0 (FPB_PIDR0)

FPB CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0003

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (Part number bits [7:0])
0x03：FPB 产品编号

41.16.6 FPB CoreSight 外设标识寄存器 1 (FPB_PIDR1)

FPB CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (Part number bits [11:8])
0x0：FPB 产品编号

41.16.7 FPB CoreSight 外设标识寄存器 2 (FPB_PIDR2)

FPB CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 002B

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x2：r0p3

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x3：Arm® JEDEC 代码

41.16.8 FPB CoreSight 外设标识寄存器 3 (FPB_PIDR3)

FPB CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

41.16.9 FPB CoreSight 组件标识寄存器 0 (FPB_CIDR0)

FPB CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (Component ID bits [7:0])
0x0D：公共 ID 值

41.16.10 FPB CoreSight 外设标识寄存器 1 (FPB_CIDR1)

FPB CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 00E0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (Component ID bits [15:12])——组件类
0xE：跟踪发生器组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (Component ID bits [11:8])
0x0：公共 ID 值

41.16.11 FPB CoreSight 组件标识寄存器 2 (FPB_CIDR2)

FPB CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (Component ID bits [23:16])
0x05：公共 ID 值

41.16.12 FPB CoreSight 组件标识寄存器 3 (FPB_CIDR3)

FPB CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (Component ID bits [31:24])
0xB1：公共 ID 值

41.16.13 FPB 寄存器映射和复位值

CPU1 FPB 的地址范围为 0xE0002000 到 0xE0002FFC。

表 264. CPU1 FPB 寄存器映射和复位值

偏移	寄存器名称	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
0x000	FPB_CTRLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NUM_CODE[6:4]			NUM_LIT[3:0]				NUM_CODE[3:0]				Res.	Res.	KEY	ENABLE
	Reset value																		0	0	0	0	0	1	0	0	1	1	0			0	0
0x004	FPB_REMAPR	Res.	Res.	RMPSPT	REMAP[23:0]																								Res.	Res.	Res.	Res.	Res.
	Reset value			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					
0x008 to 0x024	FPB_COMP0-7R	REPLACE[1:0]		Res.	COMP[26:0]																											Res.	ENABLE
	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
0xFD0	FPB_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFE0	FPB_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	0	0	0	0	1	1
0xFE4	FPB_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	0	0	0	0
0xFE8	FPB_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	1	0	1	0	1	1
0xFEC	FPB_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	FPB_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	FPB_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									1	1	1	0	0	0	0	0
0xFF8	FPB_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	FPB_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

有关寄存器边界地址的信息，请参见*第 41.13 节：CPU1 ROM 表*。

41.17 CPU1 嵌入式跟踪宏单元 (ETM™)

CPU1 ETM™ 是一款与 CPU 紧密结合的 CoreSight™ 组件。ETM™ 可生成跟踪数据包，用以跟踪 CPU1 内核的执行情况。在 STM32WB55xx 中，ETM™ 仅配置为指令跟踪，即跟踪信息中不包含数据访问。

ETM™ 通过处理器跟踪接口从 CPU 接收信息，包括：

- 在同一周期内执行的指令数量
- 程序流的变化
- 当前处理器指令状态
- 由加载和存储指令访问的存储器位置的地址
- 传输的类型、方向和大小
- 条件代码信息
- 异常信息
- 等待中断状态信息

有关更多信息，请参见《Arm® CoreSight™ ETM™-Cortex®-M4 技术参考手册》。

41.17.1 ETM 控制寄存器 (ETM_CR)

ETM control register

偏移地址：0x000

复位值：0x0000 0411

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	ETMEN	PROG	DBRQ	BO	STALL	Res.	Res.	Res.	Res.	Res.	Res.	PDN
				rw	rw	rw	rw	rw							rw

位 31:28 保留，必须保持复位值。

位 27 **TSEN**：时间戳使能 (Timestamp Enable)
0：禁止时间戳
1：使能时间戳

位 26:12 保留，必须保持复位值。

位 11 **ETMEN**：ETM 使能 (ETM enable)
0：禁止跟踪输出
1：使能跟踪输出

位 10 **PROG**：ETM 编程 (ETM programming)。在编程 ETM™ 之前，必须先将此位置 1。此位置 1 时会阻止跟踪。
0x0：ETM 处于工作状态
0x1：ETM 处于编程状态

位 9 **BO**：分支输出 (Branch output)。置 1 后，即使分支是因某个直接分支指令而引起，也会输出所有分支地址。将此位置 1 后，无需访问正在执行的代码的存储器映像，即可重建程序流。
此位置 1 时，将生成更多跟踪数据，这可能会影响跟踪系统的性能。无论此位的状态如何，都会跟踪有关分支执行的信息。

位 8 **STALL**：停止处理器 (Stall processor)。FIFOFULL 输出可用于停止处理器，以防发生溢出。FIFOFULL 输出仅在停止处理器位置 1 时使能。此位为 0 时，FIFOFULL 输出始终保持低电平。如果跟踪数据包过多，FIFO 将发生溢出。如果发生溢出，待 FIFO 清空后，跟踪将恢复工作而不会损坏。

位 6:1 保留，必须保持复位值。

位 0 **PDN**：ETM 掉电 (ETM power down)。此位可通过设计用来控制 ETM 是否处于低功耗状态。必须在调试会话开始时通过跟踪软件工具将此位清零。
此位置 1 时，可能会忽略对某些寄存器和位域的写操作。用户始终可对以下寄存器和位域进行写操作：
– ETMCR 位 [0]
– ETMLAR
– ETMCLAIMSET 寄存器
– ETMCLAIMCLR 寄存器
在此位置 1 的情况下写入 ETMCR 时，可能会忽略除位 [0] 之外的位。

41.17.2 ETM 配置代码寄存器 (ETM_CCR)

ETM configuration code register

偏移地址：0x004

复位值：0x8C80 2000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDPRES	Res.	Res.	Res.	CPMAC	TSPRES	CIDCMP[1:0]		FFPRES	NEXTO[2:0].			NEXTI[2:0].			SEQPRES
r				r	r	r		r	r			r			r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NCNT[2:0].			NMMDEC[4:0].					NDVCMP[3:0]				NADCMP[3:0]			
r			r					r				r			

位 30 **IDPRES**：ETM ID 寄存器存在 (ETM ID register present)
1：ETMIDR 寄存器存在并定义了使用的 ETM 架构版本。

位 30:28 保留，必须保持复位值。

位 27 **CPMAC**：协处理器和存储器访问 (Co-processor and memory access)
1：支持对寄存器进行存储器映射访问。

位 26 **TSPRES**：跟踪开始/停止模块存在 (Trace start/stop block present)
1：跟踪开始/停止模块存在。

位 25:24 **CIDCMP[1:0]**：上下文 ID 比较器数量 (Number of context ID comparators)
0x0：未设计上下文 ID 比较器。

位 23 **FFPRES**：FIFOFULL 逻辑存在 (FIFOFULL logic present)。要使用 FIFOFULL，系统还必须支持该功能，如 ETMSCR 的位 [8] 所示。
1：ETM 中存在 FIFOFULL 逻辑。

位 22:20 **NEXTO[2:0]**：外部输出数量 (Number of external outputs)
0x0：不支持外部输出。

位 19:17 **NEXTI[2:0]**：外部输入数量 (Number of external inputs)
0x2：支持两个外部输入。

位 16 **SEQPRES**：定序器存在 (Sequencer present)
0：ETM 中不存在定序器。

位 15:13 **NCNT[2:0]**：计数器数量 (Number of counters)
0x1：设计了一个计数器。

位 12:8 **NMMDEC[4:0]**：存储器映射解码器数量 (Number of memory map decoders)
0x0：未设计存储器映射解码器输入。

位 7:4 **NDVCMP[3:0]**：数据值比较器数量 (Number of data value comparators)
0x0：未设计数据值比较器。

位 3:0 **NADCMP[3:0]**：地址比较器对数量 (Number of address comparator pairs)
0x0：未设计地址比较器对。

41.17.3 ETM 触发寄存器 (ETM_TRIGGER)

ETM trigger register

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN [1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw		rw							rw						

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)。如果将 A 定义为第一个资源匹配，将 B 定义为第二个资源匹配，将事件定义为 A 和 B 的函数。
0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二个资源标识符 (Second resource identifier)。有关位说明，请参见 RESOURCEA[6:0] 位域。

位 6:0 **RESOURCEA[6:0]**：第一个资源标识符 (First resource identifier)。位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能会导致不可预测的行为。

类型（位 [6:4]）	索引（位 [3:0]）	
0x2	0-3	嵌入式 ICE 观察点比较器 1-4（来自 DWT）
0x4	0	值为 0 的计数器 1
0x5	15	跟踪开始/停止资源
0x6	0-1	外部输入 1-2
	15	硬接线输入，始终为真

41.17.4 ETM 状态寄存器 (ETM_SR)

ETM status register

偏移地址：0x010

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGFL	TSSRSTAT	PROGVAL	UOVFL
												rw	rw	r	r

位 31:4 保留，必须保持复位值。

位 3 **TRIGFL**：触发标志 (Trigger flag)。发生触发时置 1，可防止在再次编程 ETM 之前输出触发信号。

位 2 **TSSRSTAT**：触发开始/停止资源状态 (Trigger start/stop resource status)。保存跟踪开始/停止资源的当前状态。如果置 1，则表示已匹配跟踪开启地址，无相应的跟踪关闭地址匹配。

位 1 **PROGVAL**：编程位值 (Programming bit value)，只读值。ETM 编程位的当前有效值，ETM_CR 的位 [10]。在开始编程 ETM 之前，必须将编程位置 1 并等待此位变为 1。

在 FIFO 为空之前，或者 ETM_OSLSR 寄存器中的 OS 锁定位置 1 时，此位始终保持为 0。

位 0 **UOVFL**：未跟踪溢出 (Untraced overflow)，只读值。如果置 1，则表示存在尚未跟踪的溢出。在重新开始跟踪时，或在 ETM 掉电位（ETM_CR 寄存器的位 [0]）置 1 时，此位清零。

41.17.5 ETM 状态寄存器 (ETM_SCR)

ETM status register

偏移地址：0x014

复位值：0x0002 0D09

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	NSUPPROC[2:0]			PORTMODESUP	PORTSIZESUP	MAXPORTSIZE[3]	FFSUP	Res.	Res.	Res.	Res.	Res.	MAXPORTSIZE[2:0]		
	r			r	r	r	r						r		

位 31:18 保留，必须保持复位值。

位 17 **NOFTCHCOMP**：无取指令比较 (No fetch comparisons)

1：未设计取指令比较。

位 16:15 保留，必须保持复位值。

位 14:12 **NSUPPROC**：支持的处理器数 (Number of supported processors)

0x0：支持一个处理器。

位 11 **PORTMODESUP**：端口模式支持 (Port mode support)

0：不支持当前选择的端口模式。

1：支持当前选择的端口模式。

位 10 **PORTSIZESUP**：端口大小支持 (Port size support)

0：不支持当前选择的端口大小。

1：支持当前选择的端口大小。

位 9 **MAXPORTSIZE[3]**：最大 ETM 端口大小位 [3] (Maximum ETM port size bit [3])。此位与位 [2:0] 搭配使用

位 8 **FFSUP**：FIFOFULL 支持 (FIFOFULL support)

1：支持 FIFOFULL。

位 7:3 保留，必须保持复位值。

位 2:0 **MAXPORTSIZE[2:0]**：最大 ETM 端口大小位 [2:0] (Maximum ETM port size bit [2:0])。这些位与位 9 搭配使用，以指示支持的最大 ETM 端口大小。

0x1：最大端口大小 = 1

41.17.6 ETM 跟踪使能事件寄存器 (ETM_TEEVR)

ETM trace enable event register

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN [1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw		rw							rw						

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)。如果将 A 定义为第一个资源匹配，将 B 定义为第二个资源匹配，将事件定义为 A 和 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二个资源标识符 (Second resource identifier)。有关位说明，请参见 RESOURCEA[6:0] 位域。

位 6:0 **RESOURCEA[6:0]**：第一个资源标识符 (First resource identifier)。位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能会导致不可预测的行为。

类型（位 [6:4]）	索引（位 [3:0]）	
0x2	0-3	嵌入式 ICE 观察点比较器 1-4（来自 DWT）
0x4	0	值为 0 的计数器 1
0x5	15	跟踪开始/停止资源
0x6	0-1 15	外部输入 1-2 硬接线输入，始终为真

41.17.7 ETM 跟踪使能控制 1 寄存器 (ETM_TECR1)

ETM trace enable control 1 register

偏移地址：0x024

复位值：0x0000 0000

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

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

位 31:26 保留，必须保持复位值。

位 25 **ENONOFF**：跟踪开始/停止跟踪使能控制 (Trace start/stop trace enable control)

0：跟踪开始/停止模块不控制跟踪使能。
1：跟踪开始/停止模块控制跟踪使能。

位 24:0 保留，必须保持复位值。

41.17.8 ETM FIFOFULL 级别寄存器 (ETM_FFLR)

ETM FIFOFULL level register

偏移地址：0x028

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **LEVEL[7:0]**：FIFO 中剩余的字节数 (The number of bytes left in the FIFO)，如果低于该字节数，将触发 FIFOFULL 信号。例如，如果将此值设置为 15，而 FIFO 中的空闲字节少于 15 个，则会导致处理器停止（如果已使能）。

41.17.9 ETM 计数器重载值 1 寄存器 (ETM_CNTRLDVR1)

ETM counter reload value 1 register

偏移地址：0x140

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值。

位 15:0 **INICNT[15:0]**：初始计数 (Initial count)。指定计数器重载值。

41.17.10 ETM 同步频率寄存器 (ETM_SYNCFR)

ETM synchronization frequency register

偏移地址：0x1E0

复位值：0x0000 0400

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	FREQUENCY[11:0]											
				r											

位 31:12 保留，必须保持复位值。

位 11:0 **FREQUENCY[11:0]**：此值表示跟踪中相邻同步点的时间间隔（单位为字节）（工具只能在同步点开始解压）。

0x400：ETM 以每跟踪 1024 字节一次的固定频率来生成同步数据包。

41.17.11 ETM ID 寄存器 (ETM_IDR)

ETM ID register

偏移地址：0x1E4

复位值：0x4114 F250

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DESIGNER[7:0]								Res.	Res.	Res.	ABPE	SXSUPP	T32SUPP	Res.	LDPCF
r											r	r	r		r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROCFAM[3:0]				ETMARCHMAJ[3:0]				ETMARCHMIN[3:0]				REVISION[3:0]			
r				r				r				r			

位 31:24 **DESIGNER[7:0]**：设计人员代码 (Implementer code)

0x41：ARM ®

位 23:21 保留，必须保持复位值。

位 20 **ABPE**：备用分支数据包编码 (Alternative branch packet encoding)

1：设计了 ABPE。

位 19 **SXSUPP**：安全扩展支持 (Security extensions support)

0：ETM 的行为如同处理器始终处于安全模式。

位 18 **T32SUPP**：32 位 Thumb 指令跟踪 (32-bit Thumb instruction tracing)

1：将 32 位 Thumb 指令作为单个指令进行跟踪。

位 17 保留，必须保持复位值。

位 16 **LDPCF**：先装载 PC (Load PC first)

0：不支持数据跟踪

位 15:12 **PROCFAM[3:0]**：处理器系列 (Processor family)

0xF：此寄存器中未标识处理器系列

位 11:8 **ETMARCHMAJ[3:0]**：主要 ETM 架构版本 (Major ETM architecture version)

0x2：版本 3

位 7:4 **ETMARCHMIN[3:0]**：次要 ETM 架构版本 (Minor ETM architecture version)

0x5：版本 5

位 3:0 **REVISION[3:0]**：设计版本 (Implementation revision)

0x0：版本 0

41.17.12 ETM 配置代码扩展寄存器 (ETM_CCER)

ETM configuration code extension register

偏移地址：0x1E8

复位值：0x1854 1800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	TSSIZE	TSENC	RFC	Res.	Res.	Res.	Res.	TSIMP	EIIMP	TSSUWP	NUMWPIN[3:0]			
		r	r	r					r	r	r	r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMIR[2:0]			SUPPDAC	RR	XXINBUS[7:0]								XXINSEL[2:0]		
r			r	r	r								r		

位 31:30 保留，必须保持复位值。

位 29 **TSSIZE**：时间戳大小 (Timestamp size)
0：48 位

位 28 **TSENC**：时间戳编码 (Timestamp encoding)
1：时间戳编码为自然二进制数。

位 27 **RFC**：简化功能计数器 (Reduced function counter)
1：计数器为简化功能计数器。

位 26:23 保留，必须保持复位值。

位 22 **TSIMP**：设计时间戳 (Timestamping implemented)
1：设计了时间戳

位 21 **EIIMP**：设计嵌入式 ICE 行为控制 (Embedded ICE behavior control implemented)
0：未设计

位 20 **TSSUWP**：跟踪开始/停止使用嵌入式 ICE 观察点输入。
1：是

位 19:16 **NUMWPIN[3:0]**：嵌入式 ICE 观察点输入 (Embedded ICE watchpoint inputs)。来自 DWT 的输入数量。
0x4：四个输入

位 15:13 **NUMIR[2:0]**：仪表资源 (Instrumentation resources)
0x0：无

位 12 **SUPPDAC**：数据地址比较支持 (Data address comparison support)
1：不支持

位 11 **RR**：可读寄存器 (Readable registers)
1：所有寄存器均可读

位 10:3 **XXINBUS[7:0]**：扩展的外部输入总线 (Extended external input bus)
0x0：未设计

位 2:0 **XXINSEL[2:0]**：扩展的外部输入选择器 (Extended external input selectors)
0x0：未设计

41.17.13 ETM 跟踪使能开始/停止嵌入式 ICE 控制寄存器 (ETM_TESSEICR)

ETM trace enable start/stop Embedded ICE control register

偏移地址：0x1F0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STOPRS[3:0]			
												rw			

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

位 31:20 保留，必须保持复位值。

位 19:16 **STOPRS[3:0]**：停止资源选择 (Stop resource selection)。将这些位中的任何一位置 1 都会选择相应的嵌入式 ICE 观察点输入作为跟踪使能停止资源。位 [16] 对应输入 1，位 [17] 对应输入 2，位 [18] 对应输入 3，位 [19] 对应输入 4。

位 3:0 **STARTRS[3:0]**：开始资源选择 (Start resource selection)。将这些位中的任何一位置 1 都会选择相应的嵌入式 ICE 观察点输入作为跟踪使能开始资源。位 [0] 对应输入 1，位 [1] 对应输入 2，位 [2] 对应输入 3，位 [3] 对应输入 4。

41.17.14 ETM 时间戳事件寄存器 (ETM_TSEVR)

ETM timestamp event register

偏移地址：0x1F8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCN [1:0]		RESOURCEB[6:0]							RESOURCEA[6:0]						
rw		rw							rw						

位 31:17 保留，必须保持复位值。

位 16:14 **FCN[2:0]**：布尔函数 (Boolean function)。如果将 A 定义为第一个资源匹配，将 B 定义为第二个资源匹配，将事件定义为 A 和 B 的函数。

0x0：A
0x1：NOT A
0x2：A AND B
0x3：NOT A AND B
0x4：NOT A AND NOT B
0x5：A OR B
0x6：NOT A OR B
0x7：NOT A OR NOT B

位 13:7 **RESOURCEB[6:0]**：第二个资源标识符 (Second resource identifier)。有关位说明，请参见 RESOURCEA[6:0] 位域。

位 6:0 **RESOURCEA[6:0]**：第一个资源标识符 (First resource identifier)。位 [6:4] 定义资源类型，位 [3:0] 定义索引。下面列出了支持的资源标识符。编程任何其他值都可能会导致不可预测的行为。

类型（位 [6:4]）	索引（位 [3:0]）	
0x2	0-3	嵌入式 ICE 观察点比较器 1-4（来自 DWT）
0x4	0	值为 0 的计数器 1
0x5	15	跟踪开始/停止资源
0x6	0-1 15	外部输入 1-2 硬接线输入，始终为真

41.17.15 ETM 跟踪 ID 寄存器 (ETM_TRACEIDR)

ETM trace ID register

偏移地址：0x200

复位值：0x0000 0000

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

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

位 31:7 保留，必须保持复位值。

位 6:0 **TRACEID[6:0]**：要输出到跟踪总线上的跟踪 ID (Trace ID to output onto the trace bus)。应为其编程唯一值，以便与系统中的其他跟踪源进行区分。

41.17.16 ETM ID 寄存器 2 (ETM_IDR2)

ETM ID register 2

偏移地址：0x208

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWPTO	RFETO
														r	r

位 31:2 保留，必须保持复位值。

位 1 **SWPTO**：标识 SWP 或 SWPB 指令的顺序。

0：在存储传输之前跟踪负载传输

位 0 **RFETO**：标识 RFE 指令的顺序。
0：在 CPSR 传输之前跟踪 PC 传输

41.17.17 ETM 器件掉电状态寄存器 2 (ETM_PDSR)

ETM device power down status register 2

偏移地址：0x314

复位值：0x0000 0001

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

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

位 31:1 保留，必须保持复位值。

位 0 **POWER**：ETM 上电 (ETM powered up)
1：可访问 ETM 跟踪寄存器。

41.17.18 ETM 声明标记设置寄存器 (ETM_CLAIMSETR)

ETM claim tag set register

偏移地址：0xFA0

复位值：0x0000 000F

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

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

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMSET[3:0]**：设置声明标记位 (Set claim tag bits)
写：
0000：无影响
xxx1：将位 0 置 1
xx1x：将位 1 置 1
x1xx：将位 2 置 1
1xxx：将位 3 置 1
读：
0xF：表示声明标记使用四个位

41.17.19 ETM 声明标记清零寄存器 (ETM_CLAIMCLR)

ETM claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

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

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

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMCLR[3:0]**：重置声明标记位 (Reset claim tag bits)

写：

0000：无影响

xxx1：将位 0 清零

xx1x：将位 1 清零

x1xx：将位 2 清零

1xxx：将位 3 清零

读操作：返回声明标记的当前值

41.17.20 ETM 锁定访问寄存器 (ETM_LAR)

ETM lock access register

偏移地址：0xFB0

复位值：N/A

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

位 31:0 **ACCESS_W[31:0]**：使处理器内核能够对某些 ETM 寄存器进行写访问（调试器无需解锁组件）

0xC5ACCE55：使能写访问

其他值：禁止写访问

41.17.21 ETM 锁定状态寄存器 (ETM_LSR)

ETM lock status register

偏移地址：0xFB4

复位值：0x0000 0003

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCK TYPE	LOCK GRANT	LOCK EXIST
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **LOCKTYPE**：指示 ETM_LAR 寄存器的大小。
0：32 位

位 1 **LOCKGRANT**：当前锁定状态 (Current status of lock)。此位始终由外部调试器读为零。
0：允许写访问。
1：阻止写访问。只允许读取操作。

位 0 **LOCKEXIST**：指示是否存在锁定控制机制。此位始终由外部调试器读为零。
0：不存在锁定控制机制。
1：存在锁定控制机制。

41.17.22 ETM 认证状态寄存器 (ETM_AUTHSTATR)

ETM authentication status register

偏移地址：0xFB8

复位值：0x0000 000A

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
								r		r		r		r	

位 31:8 保留，必须保持复位值。

位 7:6 **SNID[1:0]**：安全非侵入式调试的安全等级 (Security level for secure non-invasive debug)
0x0：未设计

位 5:4 **SID**[1:0]：安全侵入式调试的安全等级 (Security level for secure invasive debug)
0x0：未设计

位 3:2 **NSNID[1:0]**：非安全非侵入式调试的安全等级 (Security level for non-secure non-invasive debug)
0x0：未设计

位 1:0 **NSID**[1:0]：非安全侵入式调试的安全等级 (Security level for non-secure invasive debug)
0x0：未设计

41.17.23 ETM CoreSight 器件标识寄存器 (ETM_DEVTYPER)

ETM CoreSight device identity register

偏移地址：0xFCC

复位值：0x0000 0013

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **SUBTYPE**：器件子类型标识符 (Device sub-type identifier)
0x1：处理器跟踪

位 3:0 **MAJORTYPE**：器件主类型标识符 (Device main type identifier)
0x3：跟踪源

41.17.24 ETM CoreSight 外设标识寄存器 4 (ETM_PIDR4)

ETM CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：Arm® JEDEC 代码

41.17.25 ETM CoreSight 外设标识寄存器 0 (ETM_PIDR0)

ETM CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0025

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (Part number bits [7:0])
0x25：ETM 产品编号

41.17.26 ETM CoreSight 外设标识寄存器 1 (ETM_PIDR1)

ETM CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (Part number bits [11:8])
0x9：ETM 产品编号

41.17.27 ETM CoreSight 外设标识寄存器 2 (ETM_PIDR2)

ETM CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 002B

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x0：r0p1

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x3：Arm® JEDEC 代码

41.17.28 ETM CoreSight 外设标识寄存器 3 (ETM_PIDR3)

ETM CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

41.17.29 ETM CoreSight 组件标识寄存器 0 (ETM_CIDR0)

ETM CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (Component ID bits [7:0])
0x0D：公共 ID 值

41.17.30 ETM CoreSight 外设标识寄存器 1 (ETM_CIDR1)

ETM CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (Component ID bits [15:12])——组件类
0x9：跟踪发生器组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (Component ID bits [11:8])
0x0：公共 ID 值

41.17.31 ETM CoreSight 组件标识寄存器 2 (ETM_CIDR2)

ETM CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (Component ID bits [23:16])
0x05：公共 ID 值

41.17.32 ETM CoreSight 组件标识寄存器 3 (ETM_CIDR3)

ETM CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (Component ID bits [31:24])
0xB1：公共 ID 值

41.17.33 ETM 寄存器映射和复位值

ETM 寄存器的地址范围为 0xE0041000 到 0xE0041FFC，调试器可以通过 CPU1 AHB-AP 对其进行访问。

表 265. CPU1 ETM 寄存器映射和复位值

偏移	寄存器名称	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
0x000	ETM_CR	Res.	Res.	Res.	TSEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETMEN	PROG	DBGRQ	BO	STALL	Res.	Res.	Res.	Res.	Res.	Res.	PDN
	Reset value				0																	0	1	0	0	0			1				1
0x004	ETM_CCR	IDPRES	Res.	Res.	Res.	CPMAC	TSPRES	CIDCMP [1:0]		FFPRES	NEXTO [2:0]			NEXTI [2:0]			SEQPRES	NCNT [2:0]			NMMDEC [4:0]					NDVCMP [3:0]				NADCMP [3:0]			
	Reset value	1				1	1	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008	ETM_TRIGGER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]							
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	ETM_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGFL	TSSRSTAT	PROGVAL	UOVFL
	Reset value																													0	0	0	0
0x014	ETM_SCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NOFTCHCOMP	Res.	Res.	NSUPPROC [2:0]			PORTMODESUP	PORTSIZESUP	MAXPORTSIZE[3]	FFSUP	Res.	Res.	Res.	Res.	Res.	MAXPORTSIZE [2:0]		
	Reset value															1			0	0	0	1	1	0	1					0	0	1
0x020	ETM_TEEVR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]							
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	ETM_TECR1	Res.	Res.	Res.	Res.	Res.	Res.	ENONOFF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value							0																									
0x028	ETM_FFLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LEVEL[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x140	ETM_CNTRLDVR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INICNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1E0	ETM_SYNCFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FREQENCY[11:0]											
	Reset value																					0	1	0	0	0	0	0	0	0	0	0	0
0x1E4	ETM_IDR	DESIGNER[7:0]								Res.	Res.	Res.	ABPE	SXSUPP	T32SUPP	Res.	LDPCF	PROCFAM [3:0]				ETMARCHMAJ [3:0]				ETMARCHMIN [3:0]				REVISION [3:0]			
	Reset value	0	1	0	0	0	0	0	1				1	0	1		0	1	1	1	1	0	0	1	0	0	1	0	1	0	0	0	0

表 265. CPU1 ETM 寄存器映射和复位值（续）

偏移	寄存器名称	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
0x1E8	ETM_CCER	Res.	Res.	TSSIZE	TSENC	RFC	Res.	Res.	Res.	Res.	TSIMP	EIIMP	TSSUWP	NUMWPIN [3:0]				NUMIR [2:0]			SUPPDAC	RR	XXINBUS[7:0]								XXINSEL [2:0]		
	Reset value	0	0	0	1	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0x1F0	ETM_TESSEICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STOPRS [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STARTRS [3:0]			
	Reset value													0	0	0	0													0	0	0	0
0x1F8	ETM_TSEVR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCN[2:0]			RESOURCEB[6:0]							RESOURCEA[6:0]							
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x200	ETM_TRACEIDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRACEID[6:0]						
	Reset value																										0	0	0	0	0	0	0
0x208	ETM_IDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWPTO	RFETO
	Reset value																															0	0
0x314	ETM_PDSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	POWER
	Reset value																																1
0xFA0	ETM_CLAIMSETR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SET[3:0]			
	Reset value																													1	1	1	1
0xFA4	ETM_CLAIMCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR[3:0]			
	Reset value																													0	0	0	0
0xFB0	ETM_LAR	KEY[31:0]																															
	Reset value																																
0xFB4	ETM_LSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NTT	SLK	SLI
	Reset value																														0	1	1
0xFB8	ETM_AUTHSTATR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID [1:0]		SID [1:0]		NSNID [1:0]		NSID [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0xFCC	ETM_DEVTYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE [3:0]				MAJORTYP [3:0]			
	Reset value																									0	0	0	1	0	0	1	1
0xFD0	ETM_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFE0	ETM_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	1	0	0	1	0	1
0xFE4	ETM_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	ETM_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	0	1	0	1	1

表 265. CPU1 ETM 寄存器映射和复位值（续）

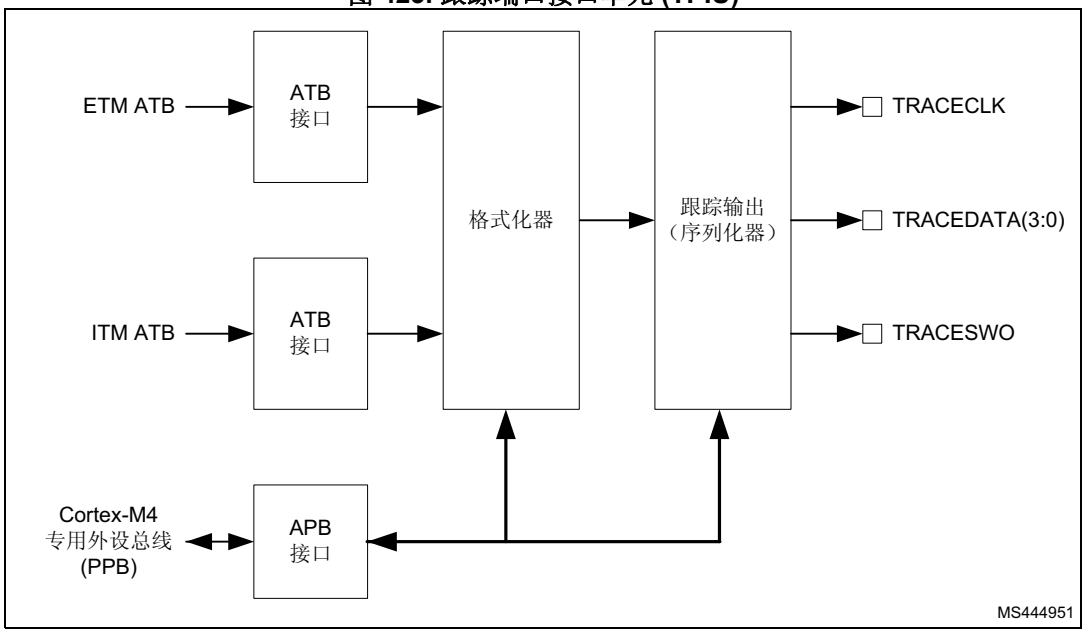
偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFEC	ETM_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND				CMOD			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	ETM_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	ETM_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS				PREAMBLE [11:8]			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	ETM_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	ETM_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

有关寄存器边界地址的信息，请参见*第 41.13 节：CPU1 ROM 表*。

41.18 CPU1 跟踪端口接口单元 (TPIU)

TPIU 用来对跟踪流进行格式化处理并将其输出到外部跟踪端口信号上。如*图 423* 所示，TPIU 有两个 ATB 从端口，分别用于从 ETM 和 ITM 传入的跟踪数据。跟踪端口为同步并行端口，包括一个时钟输出 TRACECLK 和四个数据输出 TRACEDATA(3:0)。跟踪端口宽度的可编程范围为 1 到 4。使用较小的端口宽度可减少所需测试点/连接器引脚的数量，可将腾出的 IO 用于其它用途，但代价是限制了跟踪端口的带宽，从而限制了可以实时输出的跟踪信息的数量。

图 423. 跟踪端口接口单元 (TPIU)



跟踪数据也可以输出到串行线输出 TRACESWO 上。

有关 CPU1 中跟踪端口接口的更多信息，请参见《Arm® Cortex®-M4 技术参考手册》*[2]*。

41.18.1 TPIU 支持的端口大小寄存器 (TPIU_SSPSR)

TPIU supported port size register

偏移地址：0x000

复位值：0x0000 000F

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

位 31:0 **PORTSIZE[31:0]**：指示支持的跟踪端口大小，从 1 到 32 个引脚 (Indicates supported trace port sizes, from 1 to 32 pins)。位 n-1 置 1 指示支持端口大小 n。

0x0000 000F：支持端口大小 1 到 4

41.18.2 TPIU 当前端口大小寄存器 (TPIU_CSPSR)

TPIU current port size register

偏移地址：0x004

复位值：0x0000 0001

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

位 31:0 **PORTSIZE[31:0]**：指示当前跟踪端口大小。位 n-1 置 1 指示当前端口大小为 n 个引脚。n 的值必须在支持的端口大小范围 (1-4) 内。只能将一个位置 1，否则可能导致不可预测的行为。只有当格式化器停止时，才应修改该寄存器。

41.18.3 TPIU 异步时钟预分频器寄存器 (TPIU_ACPR)

TPIU asynchronous clock presecaler register

偏移地址：0x010

复位值：0x0000 0000

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

位 31:13 保留，必须保持复位值。

位 12:0 **PRESCALER[12:0]**：为异步输出 TRACESWO 选择波特率。波特率为 TRACELKIN 频率除以 (PRESCALER +1)。

41.18.4 TPIU 所选引脚协议寄存器 (TPIU_SPPR)

TPIU selected pin protocol register

偏移地址：0x0F0

复位值：0x0000 0001

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

位 31:2 保留，必须保持复位值。

位 1:0 **TXMODE[1:0]**：选择用于跟踪输出的协议。

0x0：并行跟踪端口模式
0x1：使用曼彻斯特编码的异步 SWO
0x2：使用 NRZ 编码的异步 SWO
0x3：保留

41.18.5 TPIU 格式化器和刷新状态寄存器 (TPIU_FFSR)

TPIU formatter and flush status register

偏移地址：0x300

复位值：0x0000 0008

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

位 31:4 保留，必须保持复位值。

位 3 **FTNONSTOP**：指示格式化器是否可停止。

1：不能停止格式化器

位 2 **TCPRESENT**：指示可选的 TRACECTL 输出引脚是否可用。

0：该器件中不存在 TRACECTL 引脚。

位 1 **FTSTOPPED**：格式化器收到了停止请求信号和所有跟踪数据，并发送了后置码。ATB 接口上的任何其他跟踪数据都被忽略。

0：格式化器未停止
1：格式化器已停止

位 0 **FLINPROG**：刷新正在进行 (Flush in progress)。指示 ATB 从端口上是否有正在进行的刷新。此位反映 AFVALIDS 输出的状态。刷新可由 TPIU_FFCR 寄存器中的刷新控制位启动。

0：无正在进行的刷新
1：正在进行刷新

41.18.6 TPIU 格式化器和刷新控制寄存器 (TPIU_FFCR)

TPIU formatter and flush control register

偏移地址：0x304

复位值：0x0000 0102

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGIN	Res.	Res.	Res.	Res.	Res.	Res.	ENFCONT	Res.
							r							rw	

位 31:9 保留，必须保持复位值。

位 8 **TRIGIN**：跟踪流中的触发指示 (Trigger on trigger in)

1：指示当 TRIGIN 输入被置为有效时，跟踪流中存在触发。

位 7:2 保留，必须保持复位值。

位 1 **ENFCONT**：使能连续格式化 (Enable continuous formatting)。在 SWO 模式下将此位清 0 会忽略格式化器，并且仅输出 ITM/DWT 跟踪，ETM 跟踪将被丢弃。

0：禁止连续格式化。

1：使能连续格式化。

位 0 保留，必须保持复位值。

41.18.7 TPIU 格式化器同步计数器寄存器 (TPIU_FSCR)

TPIU formatter synchronization counter register

偏移地址：0x308

复位值：0x0000 0040

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

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

位 31:13 保留，必须保持复位值。

位 12:0 **CYCCOUNT[12:0]**：使能有效使用不同大小的 TPA，而不会浪费捕获器件的大量存储容量。该计数器包含自最上一个 128 位同步数据包以来的格式化帧数。它是一个 12 位计数器，最大计数器值为 4096。相当于每 65536 个字节（即 4096 个数据包 x 16 字节/数据包）同步一次。默认设置为每 1024 个字节（即每 64 个格式化帧）传输一个同步数据包。如果格式化器配置为连续模式，则在正常运行情况下插入全字和半字同步帧。在这些情况下，计数值为完全同步数据包间的最大完整帧数。

41.18.8 TPIU 声明标记设置寄存器 (TPIU_CLAIMSETR)

TPIU claim tag set register

偏移地址：0xFA0

复位值：0x0000 000F

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

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

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMSET[3:0]**：设置声明标记位 (Set claim tag bits)

写：

0000：无影响

xxx1：将位 0 置 1

xx1x：将位 1 置 1

x1xx：将位 2 置 1

1xxx：将位 3 置 1

读：

0xF：表示声明标记使用四个位

41.18.9 TPIU 声明标记清零寄存器 (TPIU_CLAIMCLR)

TPIU claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

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

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

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMCLR[3:0]**：重置声明标记位 (Reset claim tag bits)
写：
0000：无影响
xxx1：将位 0 清零
xx1x：将位 1 清零
x1xx：将位 2 清零
1xxx：将位 3 清零
读操作：返回声明标记的当前值

41.18.10 TPIU 器件配置寄存器 (TPIU_DEVIDR)

TPIU device configuration register

偏移地址：0xFC8

复位值：0x0000 0CA1

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SWO UARTNRZ	SWOMAN	TCLK DATA	FIFO SIZE[2:0]			CLK RELAT	MAXNUM[3:0]				
				r	r	r	r			r	r				

位 31:11 保留，必须保持复位值。

位 11 **SWONRZ**：指示是否支持串行线输出 NRZ。
0x1：支持

位 10 **SWOMAN**：指示是否支持曼彻斯特编码格式的串行线输出。
0x1：支持

位 9 **TCLKDATA**：指示是否支持跟踪时钟加数据
0x0：支持

位 8:6 **FIFOSIZE[2:0]**：以 2 的幂表示的 FIFO 大小 (FIFO size in powers of 2)
0x2：FIFO 大小 = 4 字节

位 5 **CLKRELAT**：指示 ATB 时钟和 TRACECLKIN 之间的关系（同步或异步）
0x1：异步

位 4:0 **MAXNUM[4:0]**：ATB 输入端口多路复用的数量/类型 (Number/type of ATB input port multiplexing)
0x1：两个输入端口

41.18.11 TPIU 器件类型标识寄存器 (TPIU_DEVTYPER)

TPIU device type identifier register

偏移地址：0xFCC

复位值：0x0000 0011

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **SUBTYPE[3:0]**：子分类 (Sub-classification)

0x1：跟踪端口组件

位 3:0 **MAJORTYPE[3:0]**：主分类 (Major classification)

0x1：跟踪 Sink 组件

41.18.12 TPIU CoreSight 外设标识寄存器 4 (TPIU_PIDR4)

TPIU CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)

0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)

0x4：Arm® JEDEC 代码

41.18.13 TPIU CoreSight 外设标识寄存器 0 (TPIU_PIDR0)

TPIU CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 00A1

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号位 [7:0] (Part number bits [7:0])
0xA1：CPU1 TPIU 产品编号

41.18.14 TPIU CoreSight 外设标识寄存器 1 (TPIU_PIDR1)

TPIU CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码位 [3:0] (JEP106 identity code bits [3:0])
0xB：Arm® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号位 [11:8] (Part number bits [11:8])
0x9：CPU1 TPIU 产品编号

41.18.15 TPIU CoreSight 外设标识寄存器 2 (TPIU_PIDR2)

TPIU CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 004B

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x4：r0p5

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
0x1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码位 [6:4] (JEP106 identity code bits [6:4])
0x3：Arm® JEDEC 代码

41.18.16 TPIU CoreSight 外设标识寄存器 3 (TPIU_PIDR3)

TPIU CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

41.18.17 TPIU CoreSight 组件标识寄存器 0 (TPIU_CIDR0)

TPIU CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 位 [7:0] (Component ID bits [7:0])
0x0D：公共 ID 值

41.18.18 TPIU CoreSight 外设标识寄存器 1 (TPIU_CIDR1)

TPIU CoreSight peripheral identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 位 [15:12] (Component ID bits [15:12])——组件类
0x9：CoreSight™ 组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 位 [11:8] (Component ID bits [11:8])
0x0：公共 ID 值

41.18.19 TPIU CoreSight 组件标识寄存器 2 (TPIU_CIDR2)

TPIU CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 位 [23:16] (Component ID bits [23:16])
0x05：公共 ID 值

41.18.20 TPIU CoreSight 组件标识寄存器 3 (TPIU_CIDR3)

TPIU CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 位 [31:24] (Component ID bits [31:24])
0xB1：公共 ID 值

41.18.21 CPU1 TPIU 寄存器映射和复位值

表 266. CPU1 TPIU 寄存器映射和复位值

偏移	寄存器名称	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
0x000	TPIU_SSPSR	PORTSIZE[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	1	1	1	1
0x004	TPIU_CSPSR	PORTSIZE[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	1
0x010	TPIU_ACPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x0F0	TPIU_SPPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXMODE[1:0]	
	Reset value																															0	1
0x300	TPIU_FFSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTNONSTOP	TCPRESENT	FTSTOPPED	FLINPROG
	Reset value																													1	0	0	0
0x304	TPIU_FFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGIN	Res.	Res.	Res.	Res.	Res.	Res.	ENFCONT	Res.
	Reset value																								1							1	
0x308	TPIU_FSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CYCCOUNT[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0xFA0	TPIU_CLAIMSETR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
	Reset value																													1	1	1	1
0xFA4	TPIU_CLAIMCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
	Reset value																													0	0	0	0
0xFC8	TPIU_DEVIDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWONRZ	SWOMAN	TCLKDATA	FIFOSIZE[2:0]			CLKRELAT	MUXNUM[4:0]				
	Reset value																					1	1	0	0	1	0	1	0	0	0	0	1
0xFD0	TPIU_DEVTYPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE [3:0]				MAJORTYPE [3:0]				
	Reset value																								0	0	0	1	0	0	0	0	1
0xFD0	TPIU_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]				
	Reset value																								0	0	0	0	0	1	0	0

表 266. CPU1 TPIU 寄存器映射和复位值（续）

偏移	寄存器名称	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
0xFD4	TPIU_PIDR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	TPIU_PIDR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	TPIU_PIDR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	TPIU_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									1	0	1	0	0	0	0	1
0xFE4	TPIU_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	TPIU_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	0	1	0	1	1
0xFEC	TPIU_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	TPIU_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	TPIU_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	TPIU_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	TPIU_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

41.19 CPU1 交叉触发接口 (CTI)

请参见*第 41.6 节：交叉触发接口 (CTI) 和矩阵 (CTM)*。

41.20 参考

1. IHI 0031C (ID080813)——《Arm® 调试接口架构规范 ADIv5.0 到 ADIv5.2》，发布版本 C，2013 年 8 月 8 日
2. DDI 0480F (ID100313)——《Arm® CoreSight™ SoC-400 r3p2 技术参考手册》，发布版本 G，2015 年 3 月 16 日
3. DDI 0461B (ID010111)——《Arm® CoreSight™ 跟踪存储器控制器 r0p1 技术参考手册》，发布版本 B，2010 年 12 月 10 日
4. DDI 0314H——《Arm® CoreSight™ 组件技术参考手册》，发布版本 H，2009 年 7 月 10 日
5. DDI 0403D (ID100710)——《Arm®v7-M 架构参考手册》，发布版本 E.b，2014 年 12 月 2 日
6. DDI 0494-2a (ID062813)——《Arm® CoreSight™ ETM™-M0+ r0p1 技术参考手册》，发布版本 D，2015 年 7 月 6 日
7. DDI 0440C (ID070610)——《Arm® CoreSight™ ETM™-M4 r0p1 技术参考手册》，发布版本 C，2012 年 6 月 29 日

42 器件电子签名

器件电子签名存储在 Flash 模块的系统存储区中，可以使用调试接口或 CPU 对其进行读取。它包含出厂前编程的标识和校准数据，这些数据允许用户固件或其他外部器件与微控制器的特性自动匹配。

42.1 唯一器件 ID 寄存器（96 位）

唯一器件标识符最适合：

- 用作序列号（USB 字符串序列号或其他终端应用程序）。
- 在对存储器进行编程前将唯一 ID 与软件加密原语和协议结合使用时用作安全密钥的一部分以提高 Flash 中代码的安全性。
- 激活安全启动过程等。

96 位的唯一器件标识符提供了一个对于给定器件和任何上下文都唯一的参考号码。用户不能更改这些位。

基址：0x1FFF 7590

偏移地址：0x00

只读 = 0xXXXX XXXX，其中 X 是出厂前编程的

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID(31:16)															
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
UID(15:0)															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **UID[31:0]**：以 BCD 格式表示的晶圆上的 X 和 Y 坐标 (X and Y coordinates on the wafer expressed in BCD format)

偏移地址：0x04

只读 = 0xXXXX XXXX，其中 X 是出厂前编程的

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[63:48]															
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
UID[47:32]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:8 **UID[63:40]**：LOT_NUM[23:0]
批号（ASCII 编码）

位 7:0 **UID[39:32]**：WAF_NUM[7:0]
晶圆编号（8 位无符号数）

偏移地址：0x08

只读 = 0xXXXX XXXX，其中 X 是出厂前编程的

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[95:80]															
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
UID[79:64]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **UID[95:64]**：LOT_NUM[55:24]
批号（ASCII 编码）

42.2 存储器大小数据寄存器

42.2.1 Flash 大小数据寄存器

Flash size data register

基址：0x1FFF 75E0

偏移地址：0x00

只读 = 0xXXXX，其中 X 是出厂前编程的

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

位 15:0 **FLASH_SIZE[15:0]**：Flash 大小 (Flash memory size)
指示以 KB 表示的器件 Flash 大小。
例如，0x040 对应于 64 KB。

42.3 封装数据寄存器

基址：0x1FFF 7500

偏移地址：0x00

只读 = 0xXXXX，其中 X 是出厂前编程的

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKG[4:0]				
											r	r	r	r	r

位 15:5 保留，必须保持复位值

位 4:0 **PKG[4:0]**：封装类型 (Package type)

10010：WLCSP100
10100：VFQFPN68
10101：UFQPFN84
其他值：保留

43 版本历史

表 267. 文档版本历史

日期	版本	变更
2018 年 1 月 15 日	1	初始版本。
2018 年 4 月 12 日	2	更新了文档标题*简介*和*相关文档*。 更新了*图 1：系统架构*和*图 2：存储器映射*。 更新了*表 1：STM32WB55xx 存储器映射和外设寄存器边界地址*。 更新了*第 2.1.4 节：S3：CPU2 (Cortex®-M0+) S 总线*、*CPU1 物理重映射*和*第 2.33 节：CPU2 启动*，并增加了*第 2.34 节：CPU2 SRAM 取指令禁止*。 更新了*第 3.3.4 节：读访问延迟*、*第 3.3.8 节：Flash 主存储器编程顺序*、*标准编程*、*快速编程*、*由标志指示编程错误*、*基于页的行编程中的 PGSERR 和 PGAERR*、*用户和读保护选项字节*、*安全 SRAM2 起始地址和 CPU2 复位向量选项字节*、*选项字节加载*、*第 3.5 节：FLASH UID64*、*更改读保护级别*、*第 3.6.4 节：CPU2 安全 (ESE)*、*CPU2 安全 SRAM2 区域*、*第 3.10.1 节：Flash 访问控制寄存器 (FLASH_ACR)*、*第 3.10.4 节：Flash 状态寄存器 (FLASH_SR)*、*第 3.10.5 节：Flash 控制寄存器 (FLASH_CR)*、*第 3.10.7 节：Flash 选项寄存器 (FLASH_OPTR)*、*第 3.10.16 节：Flash 安全 SRAM2 起始地址和 CPU2 复位向量寄存器 (FLASH_SRRVR)* 和*第 3.10.18 节：Flash CPU2 状态寄存器 (FLASH_C2SR)*。 增加了*导致总线错误的编程错误*和*基于页的行编程中的 PGSERR 和 PGAERR*。 更新了*表 4：Flash 时钟 (HCLK4) 频率对应的等待状态数*、*表 9：选项字节构成*、*表 13：RDP 从级别 1 降为级别 0 以及存储器擦除*和*表 18：Flash 接口寄存器映射和复位值*。 更新了*第 5.4.2 节：CRC 独立数据寄存器 (CRC_IDR)* 和*表 20：CRC 寄存器映射和复位值*。 更新了*第 6 节：电源控制 (PWR)*、*第 6.4 节：低功耗模式*、*第 6.4.3 节：进入低功耗模式*、*第 6.6.3 节：PWR 控制寄存器 3 (PWR_CR3)*、*第 6.6.5 节：PWR 状态寄存器 1 (PWR_SR1)*、*第 6.6.7 节：PWR 状态清零寄存器 (PWR_SCR)*、*第 6.6.8 节：PWR 控制寄存器 5 (PWR_CR5)*，*第 6.6.21 节：PWR CPU2 控制寄存器 1 (PWR_C2CR1)* 和*第 6.6.22 节：PWR CPU2 控制寄存器 3 (PWR_C2CR3)*。 更新了*表 24：低功耗模式汇总*、*表 30：停止 0 模式*、*表 31：停止 1 模式*、*表 32：停止 2 模式*、*表 33：待机模式*、*表 34：关断模式*和*表 35：PWR 寄存器映射和复位值*。 更新了*图 7：电源概述*和*图 8：电源配置*。 更新了*表 36：STM32WB55xx 外设互连矩阵*和*第 7.3.1 节：从定时器 (TIM1/TIM2/TIM17) 到定时器 (TIM1/TIM2)*。

表 267. 文档版本历史（续）

日期	版本	变更
2018 年 4 月 12 日	2 （续）	更新了*图 15：时钟树*。 更新了*表 37：最大时钟源频率*、*表 38：SMPS 降压转换器时钟源的选择和分频*和*表 41：RCC 寄存器映射和复位值*。 更新了*第 8.2.5 节：PLL*、*第 8.2.8 节：LSI2 时钟*、*第 8.2.9 节：系统时钟 (SYSCLK) 选择*、*第 8.2.13 节：LSI 源选择*、*第 8.2.20 节：时钟输出功能*、*第 8.2.22 节：外设时钟使能*、*第 8.3 节：低功耗模式*、*第 8.4.1 节：RCC 时钟控制寄存器 (RCC_CR)*、*第 8.4.9 节：RCC SMPS 降压转换器控制寄存器 (RCC_SMPSCR)*、*第 8.4.16 节：RCC APB3 外设复位寄存器 (RCC_APB3RSTR)*、*第 8.4.19 节：RCC AHB3 和 AHB4 外设时钟使能寄存器 (RCC_AHB3ENR)*、*第 8.4.22 节：RCC APB2 外设时钟使能寄存器 (RCC_APB2ENR)*、*第 8.4.29 节：RCC 外设独立时钟配置寄存器 (RCC_CCIPR)*、*第 8.4.31 节：RCC 控制/状态寄存器 (RCC_CSR)*、*第 8.4.34 节：RCC 扩展时钟恢复寄存器 (RCC_EXTCFGR)*、*第 8.4.35 节：RCC CPU2 AHB1 外设时钟使能寄存器 (RCC_C2AHB1ENR)*、*第 8.4.37 节：RCC CPU2 AHB3 和 AHB4 外设时钟使能寄存器 (RCC_C2AHB3ENR)*、*第 8.4.41 节：RCC CPU2 APB3 外设时钟使能寄存器 (RCC_C2APB3ENR)*、*第 8.4.42 节：睡眠模式下的 RCC CPU2 AHB1 外设时钟使能寄存器 (RCC_C2AHB1SMENR)* 和*第 8.4.48 节：睡眠模式下的 RCC CPU2 APB3 外设时钟使能寄存器 (RCC_C2APB3SMENR)*。 更新了*第 9.3.1 节：通用 I/O (GPIO)*、*第 9.3.2 节：I/O 引脚复用功能复用器和映射*、*第 9.4.1 节：GPIO 端口模式寄存器 (GPIOx_MODER)(x =A 到 E 和 H）*以及第 *9.4.4* 节到第 *9.4.11* 节。 更新了*第 10.1 节：SYSCFG 主要特性*、*第 10.2.7 节：SYSCFG SRAM2 控制和状态寄存器 (SYSCFG_SCSR)* 和*第 10.2.16 节：SYSCFG 安全 IP 控制寄存器 (SYSCFG_SIPCR)* 和*表 44：SYSCFG 寄存器映射和复位值*。 更新了*表 57：STM32WB55xx CPU1 向量表*、*表 58：STM32WB55xx CPU2 向量表*和*表 59：STM32WB55xx 唤醒中断表*。 更新了*第 14 节：扩展中断和事件控制器 (EXTI)*、*第 14.3 节：EXTI 功能说明*以及第 *14.5.1* 节到第 *14.5.8* 节的标题，并将旧版中的第 *12.5.9* 节到第 *12.5.12* 节替换为新版中的第 *14.5.9* 节到第 *14.5.16* 节。 更新了*表 65：异步中断/事件控制器寄存器映射和复位值*和*表 59：STM32WB55xx 唤醒中断表*。 删除了旧版中的第 *14.5.13* 节到第 *14.5.18* 节。 更新了*第 15.1 节：简介*、*第 15.2 节：QUADSPI 主要特性*、*第 15.3.2 节：QUADSPI 引脚*、*第 15.3.8 节：QUADSPI Flash 配置*、*第 15.3.10 节：QUADSPI 配置*、*第 15.3.15 节：nCS 行为*、*第 15.5.1 节：QUADSPI 控制寄存器 (QUADSPI_CR)* 和*第 15.5.6 节：QUADSPI 通信配置寄存器 (QUADSPI_CCR)*。 更新了*表 66：QUADSPI 引脚*和*表 68：QUADSPI 寄存器映射和复位值*。 删除了旧版中的*双 Flash 模式*和第 *23.7.5* 节到第 *23.7.7* 节。 更新了*第 16.5.5 节：ADC 配置寄存器 2 (ADC_CFGR2)*。

表 267. 文档版本历史（续）

日期	版本	变更
2018 年 4 月 12 日	2 （续）	更新了*图 41：ADC 框图*和*图 49：停止正在进行的常规转换*。 更新了*RSA 加密/解密原理*和*表 150：PKA 寄存器映射和复位值*。 更新了*第 24.3.29 节：调试模式*。 更新了*使用一个定时器启动另一个定时器*和 *TIM2 从模式控制寄存器 (TIM2_SMCR)* 中的注释。 更新了*第 31.3.5 节：如何设置看门狗超时*、*第 31.4.2 节：配置寄存器 (WWDG_CFR)*和*表 174：WWDG 寄存器映射和复位值*。 更新了*选择时钟源和合适的过采样方法*、*第 34.5.4 节：控制寄存器 3 (LPUART_CR3)*、第 *34.5.7* 节到第 *34.5.12* 节和*表 197：USART 中断请求*。 更新了*帧长度*、*第 36.3.8 节：SAI 时钟发生器*及其子章节、*主模式时钟配置错误 (NODIV = 0)*，并在*第 36.5 节：SAI 寄存器*中为相似的寄存器编排了两个不同的小节。 更新了*表 211：时钟发生器编程示例*和*表 217：SAI 中断源*。 更新了*表 220：IPCC 寄存器映射和复位值*。 删除了旧版中的第 *37.4.9* 节到第 *38.4.13* 节。 更新了*表 223：HSEM 寄存器映射和复位值*。 删除了旧版中的第 *38.4.9* 节到第 *38.4.13* 节。删除了旧版中的第 *38.4.9* 节到第 *38.4.13* 节。 更新了*图 412：CRS 框图*。 更新了*第 41.4.6 节：JTAG 调试端口*、*第 41.5 节：访问端口*、*第 41.8 节：微控制器调试单元 (DBGMCU)*、*第 41.8.1 节：DBGMCU 标识代码寄存器 (DBGMCU_IDCODE)*和*第 41.8.2 节：DBGMCU 配置寄存器 (DBGMCU_CR)*。 更新了*表 253：DBGMCU 寄存器映射和复位值*。
2019 年 1 月 8 日	3	更新了*表 1：STM32WB55xx 存储器映射和外设寄存器边界地址*。 更新了*安全系统 Flash 编程*、*第 3.3.7 节：Flash 主存储器擦除顺序*、*Flash 批量擦除*、*第 3.3.8 节：Flash 主存储器编程顺序*、*用户和读保护选项字节*、*安全 Flash 起始地址选项字节*、*安全 SRAM2 起始地址和 CPU2 复位向量选项字节*、*安全用户选项*、*WRP 区域 A 地址选项字节*、*WRP 区域 B 地址选项字节*、*更改 CPU2 安全模式*、*级别 2：禁止调试*、*更改读保护级别*、*第 3.6.4 节：CPU2 安全 (ESE)*、*CPU2 安全 SRAM2 区域*、*CPU2 调试访问*、*第 3.10.5 节：Flash 控制寄存器 (FLASH_CR)*、*第 3.10.7 节：Flash 选项寄存器 (FLASH_OPTR)*、*第 3.10.15 节：安全 Flash 起始地址寄存器 (FLASH_SFR)*、*第 3.10.16 节：Flash 安全 SRAM2 起始地址和 CPU2 复位向量寄存器 (FLASH_SRRVR)*和*第 3.10.19 节：Flash CPU2 控制寄存器 (FLASH_C2CR)*。 增加了*第 3.9 节：寄存器访问保护*。

表 267. 文档版本历史（续）

日期	版本	变更
2019 年 1 月 8 日	3 （续）	更新了*表 3：Flash—— 单存储区构成*、*表 5：页擦除概述*、*表 6：批量擦除概述*、*表 8：选项字节格式*、*表 9：选项字节构成*、*表 13：RDP 从级别 1 降为级别 0 以及存储器擦除*和*表 18：Flash 接口寄存器映射和复位值*。 更新了第 *5.4.1* 节到第 *5.4.5* 节的标题。 更新了*第 6.1.1 节：独立模拟外设电源*、*第 6.4 节：低功耗模式*、*第 6.6.3 节：PWR 控制寄存器 3 (PWR_CR3)*、第 *6.6.5* 节：*PWR 状态寄存器 1 (PWR_SR1)*、第 *6.6.8* 节到第 *6.6.20* 节 *6.6.22* 和 *6.6.23* 节。 更新了*表 23：子系统低功耗唤醒源*、*表 24：低功耗模式汇总*和*表 25：功能取决于系统工作模式*。 更新了*表 36：STM32WB55xx 外设互连矩阵*。 更新了*第 8.2 节：时钟*、*第 8.2.8 节：LSI2 时钟*、*LSI2 微调参数*、*第 8.2.20 节：时钟输出功能*、*第 8.3 节：低功耗模式*、*第 8.4.1 节：RCC 时钟控制寄存器 (RCC_CR)* 和*第 8.4.31 节：RCC 控制/状态寄存器 (RCC_CSR)*。 更新了*表 39：外设时钟使能*和*表 41：RCC 寄存器映射和复位值*，并增加了*表 40：单核低功耗调试配置*。 更新了*第 10.2.16 节：SYSCFG 安全 IP 控制寄存器 (SYSCFG_SIPCR)*。 更新了*第 12.4.3 节：DMAMUX 通道*、*第 12.4.3 节：DMAMUX 通道*、*同步溢出和中断*、*触发溢出和中断*、*第 12.6.1 节：DMAMUX 请求线复用器通道 x 配置寄存器 (DMAMUX_CxCR)*、*第 12.6.3 节：DMAMUX 请求线复用器中断清除标志寄存器 (DMAMUX_CFR)* 和*第 12.6.6 节：DMAMUX 请求发生器中断清除标志寄存器 (DMAMUX_RGCFR)*。 更新了*图 27：DMAMUX 框图*。 将*第 13 节：嵌套向量中断控制器 (NVIC)* 中的 AIEC 替换为 EXTI，并将*第 38 节：硬件信号量 (HSEM)* 中的 ID 替换为 CoreID。 更新了*表 57：STM32WB55xx CPU1 向量表*和*表 58：STM32WB55xx CPU2 向量表*。 更新了*第 15.2 节：QUADSPI 主要特性*、*FIFO 和数据管理*、*第 15.3.10 节：QUADSPI 配置*、*第 15.3.11 节：QUADSPI 的用法*、*第 15.5.1 节：QUADSPI 控制寄存器 (QUADSPI_CR)*、*第 15.5.3 节：QUADSPI 状态寄存器 (QUADSPI_SR)* 和*第 15.5.6 节：QUADSPI 通信配置寄存器 (QUADSPI_CCR)*。 更新了*表 68：QUADSPI 寄存器映射和复位值*。

表 267. 文档版本历史（续）

日期	版本	变更
2019 年 1 月 8 日	3 （续）	更新了*第 16.3.6 节：ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN)*、*第 16.3.7 节：单端输入通道和差分输入通道*、*第 16.3.8 节：校准（ADCAL、ADCALDIF、ADC_CALFACT）*、*第 16.3.10 节：写入 ADC 控制位时的限制*、*第 16.3.11 节：通道选择（SQRx、JSQRx）*和*第 16.3.12 节：可独立设置各通道采样时间（SMPR1、SMPR2）*。 更新了*表 69：ADC 内部输入/输出信号*和*图 43：ADC1 连接功能*。 更新了*第 21.1 节：简介*、*第 21.2 节：RNG 主要特性*和*运行状态检查*。 更新了*Montgomery 空间与快速模式运算*、*RSA 加密/解密原理*、*执行 PKA 运算*、*使用预先计算的 Montgomery 参数*、*第 23.3.6 节：PKA 错误管理*、*扩展 ECDSA 支持*、*第 23.5.1 节：曲线配置*、*第 23.7.1 节：PKA 控制寄存器 (PKA_CR)*、*第 23.7.2 节：PKA 状态寄存器 (PKA_SR)*、*第 23.7.3 节：PKA 清零标志寄存器 (PKA_CLRFR)* 和*第 23.7.4 节：PKA RAM*。 更新了*表 123：模加*、*表 124：模减*、*表 125：Montgomery 乘法*、*表 129：模约简*、*表 130：算术加法*、*表 131：算术减法*、*表 132：算术乘法*、*表 133：算术比较*、*表 143：ECC 曲线参数*、*表 144：模幂*、*表 145：ECC 标量乘法*、*表 146：ECDSA 签名平均计算时间*、*表 147：ECDSA 验证平均计算时间*和*表 150：PKA 寄存器映射和复位值*。 将*第 24 节：高级控制定时器 (TIM1)* 中的 BKEx 和 BKPx 替换为 BKE、BK2E、BKP 和 BK2P，并将寄存器拆分为 CCMR1 和 CCMR2。 更新了*第 24.3.16 节：使用刹车功能*。 更新了*图 148：高级控制定时器框图*和*图 195：输出重定向（图中未显示 BRK2 请求）*。 更新了*第 26.3.17 节：调试模式*和*第 26.4.6 节：TIMx 捕获/比较模式寄存器 1 [复用] (TIMx_CCMR1)（x = 16 到 17）*。 更新了*表 163：TIM16/TIM17 寄存器映射和复位值*。 更新了*图 278：输出重定向*。 更新了*第 29.2 节：RTC 主要特性*和*第 29.6.20 节：RTC 备份寄存器 (RTC_BKPxR)*。 更新了*表 172：RTC 寄存器映射和复位值*。 增加了*第 30.3.4 节：低功耗冻结*。 更新了*第 30.3.1 节：IWDG 框图*和*第 30.4.2 节：IWDG 预分频器寄存器 (IWDG_PR)*。 删除了旧版中的*第 49.3.6 节：停止和待机模式下的行为*。 更新了*表 173：IWDG 寄存器映射和复位值*。 更新了*第 38.3.8 节：AHB 总线主控 ID 验证*和第 *38.4.1* 节到第 *38.4.7* 节。 增加了*表 222：未经授权的 AHB 总线主控 ID*，并更新了*表 223：HSEM 寄存器映射和复位值*。

表 267. 文档版本历史（续）

日期	版本	变更
2019 年 1 月 8 日	3 （续）	更新了*第 41.1 节：简介*、*第 41.4.13 节：DP 目标标识寄存器 (DP_TARGETIDR)*、*第 41.7 节：交叉触发接口寄存器*、*第 41.8.1 节：DBGMCU 标识代码寄存器 (DBGMCU_IDCODE)*、*第 41.10.14 节：DWT CoreSight 外设标识寄存器 1 (DWT_PIDR1)*和*第 41.14.14 节：DWT CoreSight 外设标识寄存器 1 (DWT_PIDR1)*。 更新了*表 246：调试端口寄存器映射和复位值*和*表 253：DBGMCU 寄存器映射和复位值*。 删除了旧版中的*第 41.11 节：CPU2 指令跟踪宏单元 (ITM)* 及其子章节。 更新了*第 42.1 节：唯一器件 ID 寄存器（96 位）*。
2019 年 3 月 1 日	4	将文档分类从限于 ST 公司变为公开发布。 更新了*第 8.3 节：低功耗模式*、*通道配置流程*、*通道状态和禁止通道*、*第 21.1 节：简介*、*第 21.2 节：RNG 主要特性*、*第 21.3.3 节：随机数生成*及其子章节、*第 21.3.4 节：RNG 初始化*、*第 21.3.5 节：RNG 操作*及其子章节、*第 21.3.6 节：RNG 时钟*、*第 21.3.7 节：错误管理*、*第 21.3.8 节：RNG 低功耗使用*、*第 21.6.1 节：简介*、*第 21.6.3 节：数据采集*、*第 21.7.2 节：RNG 状态寄存器 (RNG_SR)*、*第 21.7.3 节：RNG 数据寄存器 (RNG_DR)*、第 *25.4.7* 节到第 *25.4.24* 节、第 *26.4.6* 节和第 *26.4.7* 节、*第 33.5.4 节：USART FIFO 和阈值*、*第 34.3.4 节：LPUART FIFO 和阈值*和*第 36.3.12 节：SPDIF 输出*。 更新了*表 59：STM32WB55xx 唤醒中断表*、*表 160：TIM2 寄存器映射和复位值*和*表 205：STM32WB55xx SPI 实现*。 更新了*图 114：RNG 框图*、*图 115：熵源模型*和*图 357：使用 DMA 进行接收*。 增加了*图 116：RNG 初始化概述*，并在*图 372：发送和接收FIFO 中的数据封装*中增加了脚注。

索引

D

E

F

G

H

I

L

P

Q

R

S

T

U

V

W

XYZ