

1. 功能特性

- 内核
 - 32 位 ARM® Cortex® -M0 CPU
 - 最高 32MHz 工作频率
- 存储器
 - 最大 128 Kbytes FLASH 存储器
 - 4 Kbytes SRAM
- 时钟模块
 - 内部 32MHz RC 振荡器(HRC), 典型情况下精度±1%
 - 内部 32KHz RC 振荡器(LRC), 典型情况下精度±10%
 - 32.768KHz 低速晶体振荡器(LXT)
- 工作环境
 - VDD 电压: 2.2~3.6V
 - VDDA 电压: 2.2~3.6V
 - 温度范围: -40~105°C
- 电源管理
 - 低功耗模式: 睡眠, 深度睡眠和掉电模式
 - 支持上电/掉电复位 (POR/PDR)
 - 支持低电压检测 (LVD)
- 通用输入输出
 - 35 个 I/O 支持最高 32MHz 工作频率
 - 支持中断向量
- 高精度模数转换器(Sigma-Delta ADC)
 - 24 bit 高精度 Sigma-Delta ADC
 - 最大支持 8 个外部输入通道
 - 支持单端、差分输入
 - 1/2/4/8/16/32/64/128 倍可选增益
 - 积分非线性 (INL) 最大 30ppm
 - ADC 通道温漂 2ppm/°C
 - 输出速率 8Hz~8kHz
 - ENOB≥19.5bit@30sps, PGA=128
 - ENOB≥15.4 bit@8ksps, PGA=128
 - 硬件自动切换 ADC 通道, 自动轮询读 ADC 数据, 中断通知 MCU 或 DMA
 - 自带参考电压, 输出 1.8/2.35/2.45/2.8V 可选
 - 集成温度传感器/电源电压检测通道
- 数字比较器
 - 快速响应的数字比较器
- LCD Driver
 - 集成 4 COM, 20 SEG 配置
 - 集成 charge pump
- LED Driver

- 最大支持 7 x 8 段
- 一路蜂鸣器
- 2 个定时器
 - 4 路 16bit 高级控制定时器 (TIM1), 带死区和互补控制的 6 通道 PWM 输出
 - 4 路 16bit 通用定时器 (TIM2), 带 PWM 输出
- 可编程恒流源
 - 8mA,10mA,12mA,20mA
 - 支持 PWM 控制
- OLED 彩屏 DMA 加速模块
- 串行单线调试 (SWD)
- 芯片唯一 ID
- 封装
 - LQFP48(7mmx7mm)
 - SSOP24(8.2mmx5.3mm)
 - QFN48(6mmx6mm)

2. LH32M0S3 系列产品特性和外设功能

外设		LH32M0S3 -26	LH32M0S3 -28	LH32M0S3 -2B
Flash - Kbytes		32	64	128
SRAM- Kbytes		4	4	4
T i m e r s	General-Purpose	1	1	1
	Advanced-control	1	1	1
	SysTick	1	1	1
	Watchdog	1	1	1
	RTC	1	1	1
通 信 接 口	SPI	1	1	1
	I ² C	1	1	1
	UART	1	1	1
GPIOs		35	35	35
EXTI		10	10	10
DMA		1	1	1
LCD Driver (4X20)		1	1	1
LED Driver (7X8)		1	1	1
Beeper		1	1	1
24 Bit ADC		1 (24Bit, ≤ 8kSPS)	1 (24Bit, ≤ 8kSPS)	1 (24Bit, ≤ 8kSPS)
Reference		1 (1.8/2.35/2.45/2.8V)	1 (1.8/2.35/2.45/2.8V)	1 (1.8/2.35/2.45/2.8V)
ACM (1.2V)		1	1	1
Current Source		1	1	1

3. 说明

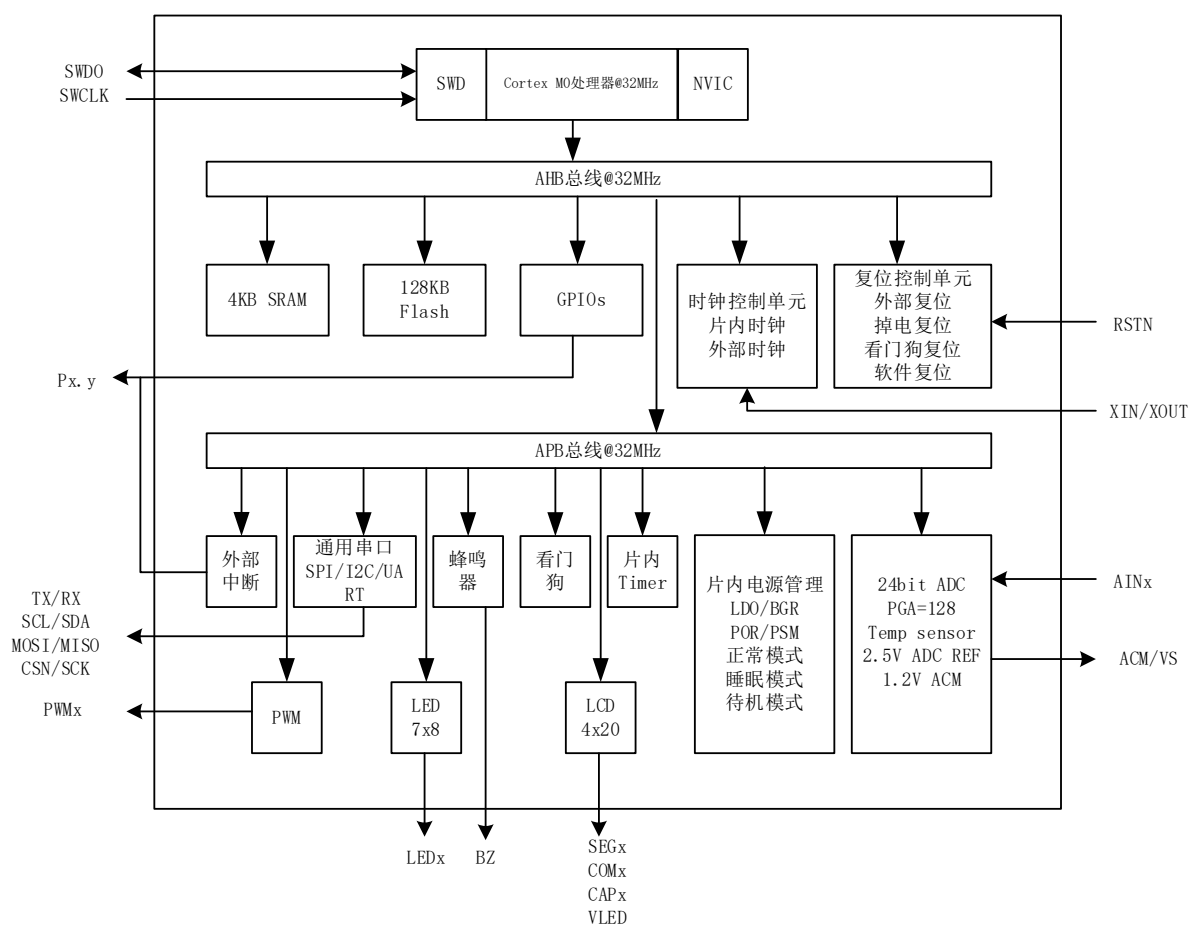
LH32M0S3 系列微控制器采用高性能的 32 位 ARM® Cortex®-M0 内核，嵌入高达 128Kbytes FLASH 和 4Kbytes SRAM，最高工作频率 32MHz。芯片提供标准的通信接口 (I²C、SPI 和 USART)，多路 24bit 高精度 ADC，2 路 16bit 定时器，一路蜂鸣器驱动以及屏幕控制器。

LH32M0S3 系列微控制器支持最大 4x20 LCD 和最大 7x8 LED 控制。

LH32M0S3 系列微控制器的工作温度范围为-40℃~105℃，工作电压范围 2.2V~3.6V。芯片提供一系列电源工作模式，以满足不同的低功耗应用。

LH32M0S3 系列微控制器适用于多种应用场景，例如红外测温、高精度测量、血氧饱和度、血压计、数据采集、工业控制等。

4. 结构框图



目录

1. 功能特性	1	9.3.2. 独立看门狗唤醒	28
2. LH32M0S3 系列产品特性和外设功能	1	9.4. 电源控制寄存器	29
3. 说明	2	10. 中断 (INTERRUPT)	29
4. 结构框图	2	10.1. 嵌套向量中断控制器	29
5. 版本历史	6	10.2. 外部中断/事件控制器 (EXTI)	29
6. 引脚描述	7	10.2.1. 主要特性	29
6.1. LQFP48	7	10.2.2. 框图	30
6.2. QFN48	8	10.2.3. 唤醒事件管理	30
6.3. SSOP24	9	10.2.4. 功能说明	30
6.4. 引脚描述及 IO 复用	10	10.2.5. 外部中断/事件线路映像	31
7. 存储器和系统总线	10	10.3. 中断寄存器	33
7.1. ARM® CORTEX®-M0 内核	10	11. 通用和复用功能 I/O (GPIO AND PIN MUX)	34
7.2. 系统嘀嗒 (SYSTICK)	11	11.1. GPIO 功能描述	34
7.3. 系统功能框图	12	11.1.1. 通用 I/O (GPIO)	34
7.4. 系统地址映射	12	11.1.2. 单独的位设置或位清除	34
7.5. 嵌入式 SRAM	13	11.1.3. 外部中断/唤醒线	34
7.6. 嵌入式 FLASH	13	11.1.4. 复用功能 (AF)	34
7.6.1. Flash 控制器功能点	13	11.1.5. 附加功能	34
7.6.2. Flash 用户空间	13	11.1.6. 输入配置	34
7.6.3. Flash 保护和完整性检查	14	11.1.7. 输出配置	35
7.6.4. Flash 控制器性能和命令周期	15	11.1.8. 复用功能配置	35
7.6.5. Flash 寄存器	15	11.1.9. 模拟输入配置	35
8. 系统时钟和复位 (RCC)	17	11.2. GPIO 寄存器	35
8.1. HSE 时钟	18	12. 直接内存访问 (DMA)	39
8.2. HSI 时钟	19	12.1. DMA 主要特性	39
8.3. LSI/LSE 时钟	19	12.2. DMA 功能描述	39
8.4. 系统时钟选择	19	12.2.1. 基本模式	39
8.5. 看门狗/RTC/BEEP/LCD 时钟	19	12.2.2. 循环模式	40
8.6. 时钟输出	19	12.2.3. 链表模式	40
8.7. 复位架构 (RESET)	20	12.3. DMA 寄存器	42
8.7.1. 系统复位	20	13. 24 位高精度模数转换器 (ADC)	45
8.7.2. 电源复位	20	13.1. 参考电压	45
8.8. RCC 寄存器	21	13.2. 共模电压 VCM	45
9. 工作模式 (POWER MODE)	26	13.3. ADC 上电及复位	45
9.1. 电源管理	26	13.4. MODULATOR 时钟和数据速率	46
9.1.1. 上电复位 (POR) 和掉电复位 (PDR)	26	13.5. 通道选择和数据读取	46
9.1.2. 可编程电压监测器 (PVD)	26	13.6. 计算 ADC 数据	46
9.1.3. Always on (AON) 区域	26	13.7. SEQUENCER 序列器	47
9.2. 低功耗模式	27	13.8. 温度传感器	47
9.2.1. 降低系统时钟	27	13.9. ADC 寄存器	47
9.2.2. 外部时钟的控制	27	14. 通用定时器 (TIM2)	51
9.2.3. 睡眠模式	27	14.1. TIM2 简介	51
9.2.4. 深度睡眠模式 1 和深度睡眠模式 2	28	14.2. TIM2 主要功能	51
9.3. 低功耗模式下的自动唤醒	28	14.3. TIM2 功能描述	52
9.3.1. RTC timer 唤醒	28	14.3.1. 时基单元	52
		14.3.2. 计数器模式	53
		14.3.3. 时钟源选择	61

14.3.4. 捕获/比较通道	63	18.3.2. 发送器	152
14.3.5. 输入捕获模式	64	18.3.3. 接收器	154
14.3.6. PWM 输入模式	65	18.3.4. 分数波特率的产生	157
14.3.7. 强置输出模式	66	18.3.5. USART 接收器容忍时钟的变化	158
14.3.8. 输出比较模式	66	18.3.6. 多处理器通信	159
14.3.9. PWM 模式	67	18.3.7. 校验控制	160
14.3.10. 单脉冲模式	69	18.3.8. LIN (局域互联网) 模式	161
14.3.11. 在外部事件时清除 OCxREF 信号 ...	70	18.3.9. USART 同步模式	163
14.3.12. 编码器接口模式	71	18.3.10. 单线半双工通信	165
14.3.13. 定时器输入异或功能	72	18.3.11. 智能卡	165
14.3.14. 定时器和外部触发的同步	73	18.3.12. IrDA SIR ENDEC 功能模块	167
14.3.15. 定时器同步	75	18.3.13. 利用 DMA 连续通信	168
14.3.16. 调试模式	79	18.3.14. 硬件流控制	170
14.4. TIM2 寄存器描述	80	18.4. USART 中断请求	171
15. 高级控制定时器 (TIM1)	92	18.5. USART 模式配置	172
15.1. TIM1 简介	92	18.6. UART 寄存器描述	172
15.2. TIM1 主要特性	92	19. 内部集成电路通信接口 (I ² C)	177
15.3. TIM1 功能描述	93	19.1. I ² C 简介	177
15.3.1. 时基单元	93	19.2. I ² C 主要特点	177
15.3.2. 预分频器描述	94	19.3. I ² C 功能描述	177
15.3.3. 计数器模式	94	19.3.1. 模式选择	178
15.3.4. 时钟选择	103	19.3.2. I ² C 从模式	179
15.3.5. 输入捕获模式	108	19.3.3. I ² C 主模式	181
15.3.6. PWM 输入模式	108	19.3.4. 错误条件	184
15.3.7. 强置输出模式	109	19.3.5. SDA/SCL 线控制	185
15.3.8. 输出比较模式	109	19.3.6. DMA 请求	185
15.3.9. PWM 模式	110	19.3.7. 包错误校验 (PEC)	186
15.3.10. TIM1 定时器和外部触发的同步 ...	121	19.4. I ² C 中断请求	187
15.3.11. 定时器同步	124	19.5. I ² C 寄存器描述	188
15.4. TIM1 寄存器描述	124	20. 串行外设通信接口 (SPI)	196
16. 实时时钟 (RTC)	139	20.1. SPI 概述	196
16.1. RTC 简介	139	20.2. SPI 功能	196
16.2. 主要特性	140	20.2.1. 主入从出 (MISO) 引脚	196
16.3. 功能描述	140	20.2.2. 主出从入 (MOSI) 引脚	196
16.3.1. 概述	140	20.2.3. 串行时钟 (SCLK) 引脚	196
16.3.2. 复位过程	141	20.2.4. 片选 (CS) 引脚	197
16.3.3. 读 RTC 寄存器	141	20.2.5. 启动 SPI 传输	197
16.3.4. 配置 RTC 寄存器	142	20.2.6. FIFO Underflow 和 Overflow	197
16.3.5. RTC 标志的设置	142	20.2.7. SPI 中断	198
16.4. RTC 寄存器描述	143	20.2.8. 读指令模式	198
17. 独立看门狗 (IWDG)	146	20.3. SPI 寄存器描述	198
17.1. 简介	146	21. QSPI	203
17.2. IWDG 主要性能	146	21.1. QUADSPI 主要特性	203
17.3. IWDG 功能描述	146	21.2. QUADSPI 功能说明	204
17.3.1. 上电默认使能看门狗	146	21.2.1. QUADSPI 框图	204
17.3.2. 寄存器访问保护	146	21.2.2. QUADSPI 命令序列	204
17.3.3. 调试模式	147	21.2.3. QUADSPI 信号接口协议模式	205
17.4. IWDG 寄存器描述	147	21.2.4. QUADSPI 间接模式	206
18. 通用串行异步通信接口 (USART)	149	21.2.5. QUADSPI 状态标志轮询模式	206
18.1. USART 介绍	149	21.2.6. QUADSPI 内存映射模式	207
18.2. USART 主要特性	149	21.2.7. QUADSPI Flash 配置	207
18.3. USART 功能概述	150	21.2.8. QUADSPI 延迟数据采样	207
18.3.1. USART 特性描述	151	21.2.9. QUADSPI 配置	207
		21.2.10. QUADSPI 的用法	208

21.2.11. QUADSPI 中断	210	24.4. 蜂鸣器时钟	227
21.3. QSPI 寄存器描述	211	24.5. 蜂鸣器寄存器描述	227
22. 晶驱动器 (LCD)	216	25. 可编程电压监测器 (LVD)	228
22.1. LCD 主要特性	216	25.1. LVD 主要特征	228
22.2. LCD 功能描述	216	25.2. LVD 功能描述	229
22.2.1. LCD 控制模式	216	25.3. LVD 寄存器描述	229
22.2.2. LCD 时钟及刷新率	216	26. 电气特性	230
22.2.3. LCD 偏置电压选择	217	26.1. 绝对最大额定值	230
22.2.4. LCD 显示内容	218	26.2. 正常工作条件	230
22.2.5. LCD 驱动波形	218	26.3. I/O 端口特性	230
22.3. 电荷泵	220	26.4. FLASH 特性	231
22.4. LCD 寄存器描述	221	26.5. RC 振荡特性	231
23. 发光二极管驱动器 (LED)	223	26.6. 晶振特性	231
23.1. LED 主要特性	223	26.7. ADC 特性	232
23.2. LED 功能描述	223	26.8. 比较器特性	232
23.2.1. LED 控制模式	223	26.9. 温度传感器特性	232
23.2.2. LED 扫描周期	223	26.10. LCD	233
23.2.3. LED 亮灭控制	223	26.11. 恒流源	233
23.2.4. LED 亮度控制	224	26.12. 输出参考电压	233
23.3. LED 寄存器描述	224	26.13. 功耗	233
24. 蜂鸣器 (BEEPER)	225	26.14. ESD 特性	234
24.1. 蜂鸣器主要特性	225	27. 噪声特性	234
24.2. 蜂鸣器功能描述	226	28. 封装信息	236
24.2.1. 蜂鸣器工作模式	226	29. 订购信息	237
24.2.2. 蜂鸣器音调配置	226		
24.3. 蜂鸣器中断配置	227		

5. 版本历史

版本号	日期	更新内容
PreA	2022 年 4 月 17 日	初版
PreB	2022 年 7 月 8 日	增加噪声和功耗测量结果
PreC	2022 年 8 月 22 日	更新电气指标测量结果, DVDD 最低电压 2.2V, DeepSleep2 模式下功耗更新为 4.4uA, 更新 PWM 管脚信息, 增加 SSOP 和 QFN 封装, 更新 AVDD 做 ADC 参考电压的噪声特性
PreD	2022 年 9 月 13 日	变更芯片编码为 LH32M0S3
Rev.A	2023 年 3 月 23 日	量产
Rev.B	2023 年 6 月 20 日	错误修正
	2023 年 9 月 23 日	增加包装信息
Rev.C	2024 年 4 月 3 日	增加小包装规格
Rev.D	2024 年 5 月 16 日	增加 DMA 链表, 增加 QSPI
Rev.E	2024 年 8 月 23 日	增加芯片唯一 ID

6. 引脚描述

6.1. LQFP48

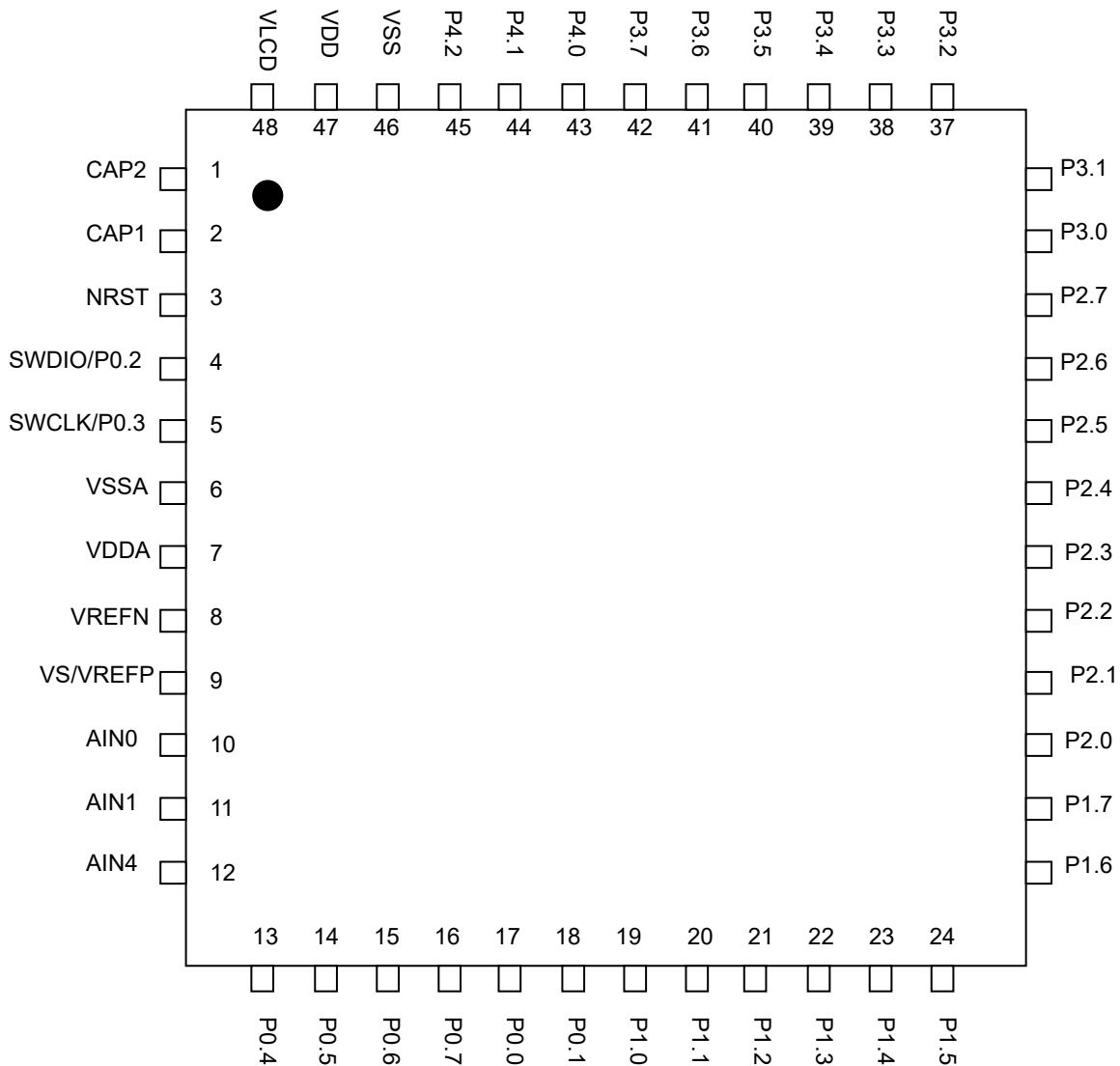


图1. LQFP48 引脚分布

表1. LQFP48 引脚描述

脚位	管脚名称	类型	描述
1	CAP2	AO	LCD 电荷泵产生所需的电荷交换电容
2	CAP1	AO	LCD 电荷泵产生所需的电荷交换电容
3	NRST	I	外部 Reset, 低电平有效
4	P0.2/AIN2/I2C_SDA/UART_TX/SPI_NSS/SWDIO	I/O	GPIO0.2/ADC 输入 AIN2/I2C_SDA/串口输出/SPI_NSS/SWDIO
5	P0.3/AIN3/COMP_N/I2C_SCL/UART_RX/SPI_SCLK/COMP_N/SWCLK	I/O	GPIO0.3/ADC 输入 AIN3/模拟比较器的负输入 COMP_N/I2C_SCL/UART_RX/SPI_SCLK/SWCLK
6	VSSA	G	模拟地
7	VDDA	P	模拟电源
8	VREFN	AO	ADC 参考地, 和模拟地连接
9	VS/REFP	AO	ADC 稳压输出, 最大可提供对地 8mA 电流
10	AIN0	AI	ADC 输入 AIN0
11	AIN1	AI	ADC 输入 AIN1
12	ACM/LVD/AIN4	AO	1.2V 参考源稳压输出/低压检测和模拟比较器正输入/ADC 输入 AIN4

脚位	管脚名称	类型	描述
13	P0.4/AIN5/I2C_SDA	I/O	GPIO0.4/ADC 输入 AIN5/I2C_SDA
14	P0.5/AIN6/PWM2/I2C_SCL	I/O	GPIO0.5/ADC 输入 AIN6/PWM2 输出/I2C_SCL
15	P0.6/AIN7/UART_TX/LVD	I/O	GPIO0.6/ADC 输入 AIN7/UART_TX 低压检测输入
16	P0.7/UART_RX/COMPOUT	I/O	GPIO0.7/UART_RX/模拟比较器输出
17	P0.0/BZ/QSPI_CLK	I/O	GPIO0.0/蜂鸣器输出/QSPI_CLK
18	P0.1/SPI_MOSI/PWM6/QSPI_NSS	I/O	GPIO0.1/SPI_MOSI/PWM6 输出/QSPI_NSS
19	P1.0/SPI_MISO/PWM5/QSPI_IO0	I/O	GPIO1.0/SPI_MISO/PWM5 输出/QSPI_IO0
20	P1.1/SPI_SCLK/PWM4/QSPI_IO1	I/O	GPIO1.1/SPI_CLK/PWM4 输出/QSPI_IO1
21	P1.2/SPI_NSS/PWM3/QSPI_IO2	I/O	GPIO1.2/SPI_NSS/PWM3 输出/QSPI_IO2
22	P1.3/SEG20/PWM1/QSPI_IO3	I/O	GPIO1.3/LCD SEG20/PWM1 输出/QSPI_IO3
23	P1.4/SEG19/SCLK_SPI	I/O	GPIO1.4/LCD SEG19/SCLK_SPI
24	P1.5/SEG18/SPI_NSS/XIN	I/O	GPIO1.5/LCD SEG18/SPI_MISO/32.768kHz 晶振输入
25	P1.6/SEG17/I2C_SCL/GPIO_CLK/BZ/COMPOUT/X OUT	I/O	GPIO1.6/LCD SEG17/I2C_SCL/输出_输入时钟/蜂鸣器输出/模拟比较器输出/32.768kHz 晶振输出
26	P1.7/SEG16/I2C_SDA/SPI_MISO/BZn	I/O	GPIO1.7/LCD SEG16/I2C_SDA/SPI_MISO/蜂鸣器互补输出
27	P2.0/SEG15/UART_TX/SPI_MOSI/PWM7/BZ	I/O	GPIO2.0/LCD SEG15/UART_TX/SPI_MOSI/PWM7/蜂鸣器输出
28	P2.1/SEG14/UART_RX/PWM2/SPI_MISO	I/O	GPIO2.1/LCD SEG14/UART_RX/PWM2 输出/SPI_MISO
29	P2.2/SEG13/LED8/BZ/PWM1/SPI_SCLK	I/O	GPIO2.2/LCD SEG13/LED8/蜂鸣器输出/PWM1 输出/SPI_SCLK
30	P2.3/SEG12/LED7/SPI_MOSI/PWM6	I/O	GPIO2.3/LCD SEG12/LED7/SPI_MOSI/PWM6 输出
31	P2.4/SEG11/LED6/SPI_MISO/PWM5	I/O	GPIO2.4/LCD SEG11/LED6/SPI_MOSI/PWM5 输出
32	P2.5/SEG10/LED5/SPI_SCLK/PWM4	I/O	GPIO2.5/LCD SEG10/LED5/SPI_CLK/PWM4 输出
33	P2.6/SEG9/LED4/SPI_NSS/PWM3	I/O	GPIO2.6/LCD SEG9/LED4/SPI_NSS/PWM3 输出
34	P2.7/SEG8/LED3/GPIO_CLK/TM1_BKIN	I/O	GPIO2.7/LCD SEG8/LED3/输入时钟/Timer1 刹车
35	P3.0/SEG7/LED2/PWM1	I/O	GPIO3.0/LCD SEG7/LED2/PWM1 输出
36	P3.1/SEG6/LED1/PWM2	I/O	GPIO3.1/LCD SEG6/LED8/PWM2 输出
37	P3.2/SEG5	I/O	GPIO3.2/LCD SEG5
38	P3.3/SEG4/UART_CTS	I/O	GPIO3.3/LCD SEG4/UART_CTS
39	P3.4/SEG3/UART_RTS	I/O	GPIO3.4/LCD SEG3/UART_RTS
40	P3.5/SEG2/QSP_NSS/UART_RX/PWM7	I/O	GPIO3.5/LCD SEG2/QSP_NSS/UART_RX/PWM7 输出
41	P3.6/SEG1/QSPI_SCLK/UART_TX/TM1_BKIN	I/O	GPIO3.6/LCD SEG1/UART_TX/Timer1 刹车信号
42	P3.7/COM4/QSPI_IO0/BZ	I/O	GPIO4.0/LCD COM4/QSPI_IO0/蜂鸣器输出
43	P4.0/COM3/QSPI_IO1/BZn	I/O	GPIO4.1/LCD COM3/QSPI_IO1/蜂鸣器互补输出
44	P4.1/COM2/QSPI_IO2	I/O	GPIO4.2/LCD COM2/QSPI_IO2
45	P4.2/COM1/QSPI_IO3/COMPOUT	I/O	GPIO4.3/LCD COM1/QSPI_IO3/模拟比较器输出
46	VSS	G	数字地
47	VDD	P	数字电源
48	VLCD	P	LCD 驱动电路 1.驱动 LCD 时可选内电荷泵, VLCD_LED 接片外电容 2.内部电荷泵关断时候, VLCD 内部和 VDD 短路

6.2. QFN48

QFN 封装引脚描述与 LQFP48 引脚描述相同。

6.3. SSOP24

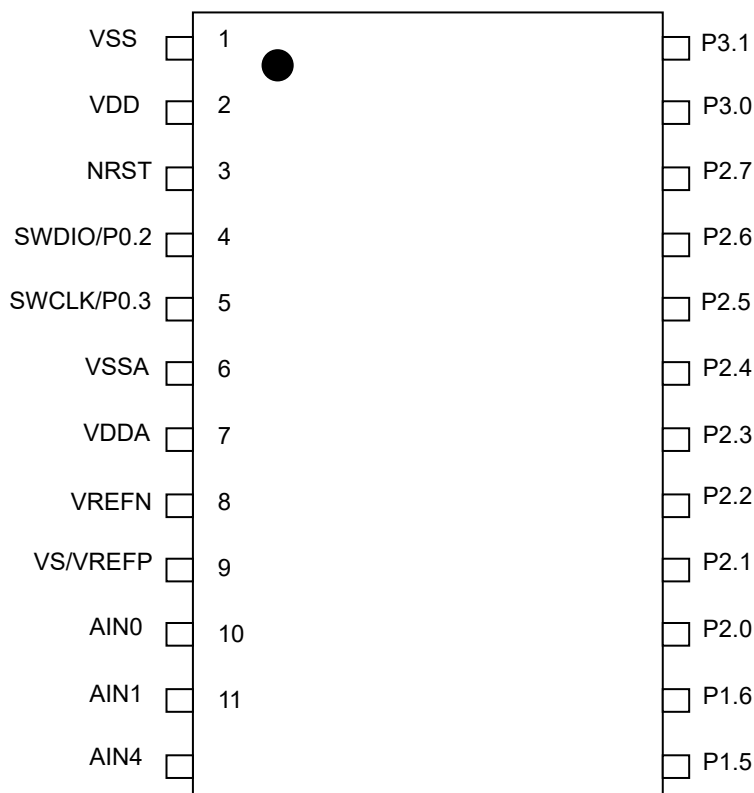


图2. SSOP24 封装引脚分布

表2. SSOP24 引脚列表

脚位	管脚名称	类型	描述
1	VSS	G	数字地
2	VDD	P	数字电源
3	NRST	I	外部 Reset, 低电平有效
4	P0.2/AIN2/I2C_SDA/UART_TX/SPI_NSS/SWDIO	I/O	GPIO0.2/ADC 输入 AIN2/I2C_SDA/串口输出 /SPI_NSS/SWDIO
5	P0.3/AIN3/COMP_N/I2C_SCL/UART_RX/SPI_SCLK/COMP_N/SW CLK	I/O	GPIO0.3/ADC 输入 AIN3/模拟比较器的负输入 COMP_N/I2C_SCL/UART_RX/SPI_SCLK/SWCLK
6	VSSA	G	模拟地
7	VDDA	P	模拟电源
8	VREFN	AO	ADC 参考地, 和模拟地连接
9	VS/REFP	AO	ADC 稳压输出, 最大可提供对地 8mA 电流
10	AIN0	AI	ADC 输入 AIN0
11	AIN1	AI	ADC 输入 AIN1
12	ACM/LVD/AIN4	AO	1.2V 参考源稳压输出/低压检测和模拟比较器正输入 /ADC 输入 AIN4
13	P1.5/SPI_NSS/XIN	I/O	GPIO1.5/SPI_MISO/32.768kHz 晶振输入
14	P1.6/I2C_SCL/GPIO_CLK/BZ/COMP_OUT/XOUT	I/O	GPIO1.6/I2C_SCL/输出_输入时钟/蜂鸣器输出/模拟比较器输出/32.768kHz 晶振输出
15	P2.0/UART_TX/SPI_MOSI/PWM7/BZ	I/O	GPIO2.0/UART_TX/SPI_MOSI/PWM7/蜂鸣器输出
16	P2.1/UART_RX/PWM2/SPI_MISO	I/O	GPIO2.1/UART_RX/PWM2 输出/SPI_MISO
17	P2.2/LED8/BZ/PWM1/SPI_SCLK	I/O	GPIO2.2/LED8/蜂鸣器输出/PWM1 输出/SPI_SCLK
18	P2.3/LED7/SPI_MOSI/PWM6	I/O	GPIO2.3/LED7/SPI_MOSI/PWM6 输出
19	P2.4/LED6/SPI_MISO/PWM5	I/O	GPIO2.4/LED6/SPI_MOSI/PWM5 输出
20	P2.5/LED5/SPI_SCLK/PWM4	I/O	GPIO2.5/LED5/SPI_CLK/PWM4 输出
21	P2.6/LED4/SPI_NSS/PWM3	I/O	GPIO2.6/LED4/SPI_NSS/PWM3 输出
22	P2.7/LED3/GPIO_CLK/TM1_BKIN	I/O	GPIO2.7/LED3/输入时钟/Timer1 刹车

脚位	管脚名称	类型	描述
23	P3.0/LED2/PWM1	I/O	GPIO3.0/LED2/PWM1 输出
24	P3.1/LED1/PWM2	I/O	GPIO3.1/LED8/PWM2 输出

6.4. 引脚描述及 IO 复用

表3. 引脚描述及 IO 复用

Pin Name	Default	AF1	AF2	AF3	AF4	AF5	Additional Analog
P0.0	GPIO	BZ					
P0.1	GPIO	SPI_MOSI	PWM6(TIM1_CH3N)				
P0.2	SWDIO	SDA	UART1_TX	SPI_CS			AIN2
P0.3	SWDCLK	SCL	UART1_RX	T1_ETR	SPI_SCLK		AIN3
P0.4	GPIO	SDA					AIN5
P0.5	GPIO	SCL	PWM2(TIM1_CH1N)				AIN6
P0.6	GPIO	UART1_TX	T1_ETR				AIN7/LVD
P0.7	GPIO	UART1_RX	T1_BKIN		COMPOUT		
P1.0	GPIO	SPI_MISO	PWM5(TIM1_CH3)				
P1.1	GPIO	SPI_SCLK	PWM4(TIM1_CH2N)				
P1.2	GPIO	SPI_CS	PWM3(TIM1_CH2)				
P1.3	GPIO		PWM1(TIM1_CH1)				SEG20
P1.4	GPIO	SPI_SCLK			MCO		SEG19
P1.5	GPIO	SPI_CS	SDA	T4_CH4			SEG18
P1.6	GPIO	SCL	COMPOUT	CLKIN	BZ		SEG17
P1.7	GPIO	SDA		SPI_MISO	BZN		SEG16
P2.0	GPIO	UART1_TX	PWM7(TIM1_CH4)	SPI_MOSI	BZ		SEG15
P2.1	GPIO	UART1_RX	PWM2	SPI_MISO	BZN		SEG14
P2.2	GPIO		PWM1(TIM1_CH1)	SPI_SCLK	BZ		SEG13 LED8
P2.3	GPIO	SPI_MOSI	PWM6(TIM1_CH3N)		BZN		SEG12 LED7
P2.4	GPIO	SPI_MISO	PWM5(TIM1_CH3)				SEG11 LED6
P2.5	GPIO	SPI_SCLK	PWM4(TIM1_CH2N)				SEG10 LED5
P2.6	GPIO	SPI_CS	PWM3(TIM1_CH2)		BZN		SEG9 LED4
P2.7	GPIO		T1BKIN	CLKIN			SEG8 LED3
P3.0	GPIO		PWM1(TIM1_CH1)				SEG7 LED2
P3.1	GPIO		PWM2(TIM1_CH1N)		MCO		SEG6 LED1
P3.2	GPIO						SEG5
P3.3	GPIO	CTS					SEG4
P3.4	GPIO	RTS	T1_ETR				SEG3
P3.5	GPIO	RX	PWM7(TIM1_CH4)				SEG2
P3.6	GPIO	TX					SEG1
P3.7	GPIO		T2_CH1		BZ		
P4.0	GPIO		T2_CH2		BZN		
P4.1	GPIO		T2_CH3				
P4.2	GPIO		T2_CH4		COMPOUT		

7. 存储器和系统总线

7.1. ARM® Cortex® -M0 内核

Cortex-M0 是一个资源占用极小，高效能，专门给微控制器和嵌入式应用需求设计的一款优化的处理器。主要支持如下的 feature:

- Thumb 指令集，高密度指令集并支持 32bit 指令和指令处理性能
- 对于 power 和资源有一定优化
- 针对低功耗设计，支持 sleep mode 和 deep sleep mode
- 硬件乘法器
- 允许慢速系统时钟，高效 code 执行

- 高性能中断处理
- 支持两线 debug

Cortex-M0 核是一款可配置，3 级流水线，32-bit RSIC 架构的处理器核。它本身包含了 AMBA AHB-LITE 接口，NVIC 模块，可选的硬件调试模块，支持 Thumb 指令集并可以和其他 Cortex-M 系列核兼容代码。M0 核的大致功能模块框图如下。

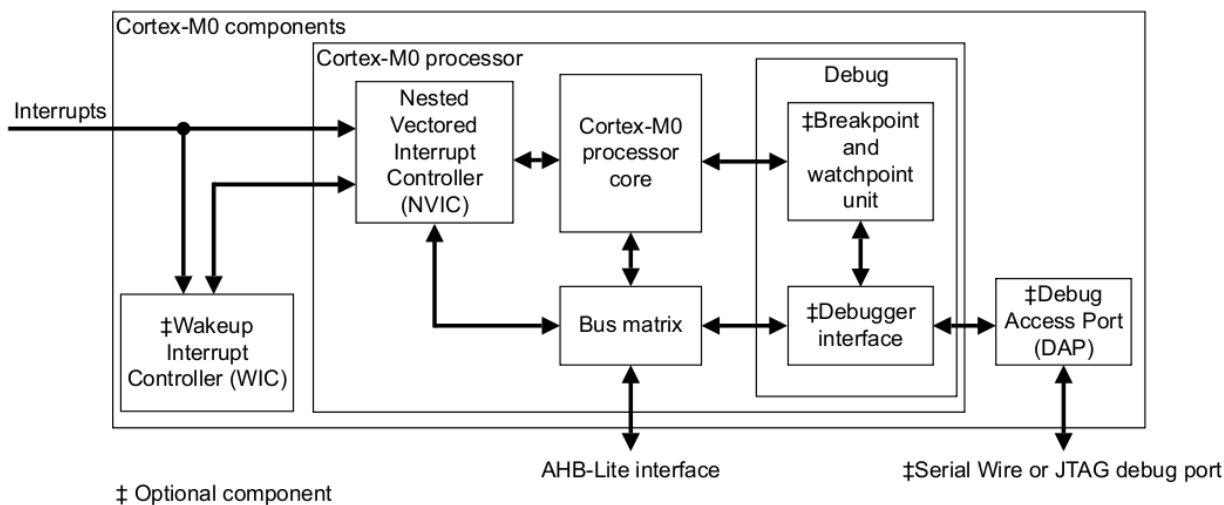


图3. 功能模块框图

M0 核本身提供了一定的配置选择，为了节省面积成本，配置的选择略去了一些不需要的功能，最终 M0 核的集成我们选择的配置如下：

- 中断根据系统的需要，目前支持数量见中断向量列表，支持结合 NVIC 使用
 - NVIC 支持最大 32 个中断连接，每个中断可配置为 4 级优先级
 - NVIC 支持 1 个 None-Maskable Interrupt (NMI) 连接
 - NVIC 支持对中断的电平和脉冲敏感判断
- 硬件乘法器，支持单周期 32-bit 硬件乘法器
- 整个系统为小端模式
- 支持 1 个 systick（滴答定时器）
- 调试模式组件支持
 - 4 个断点（breakpoint），不支持 watchpoints
 - 支持使用 BRPT 指令设置无限制数量的软件断点（breakpoints）
 - 调试者使用 DAP 接口通过 bus matrix 可以对系统外设，系统地址映射做 zero-waitstate 访问
 - 当处理器被 halt 时，可以直接访问 core 本身的寄存器
 - 兼容 CoreSight，通过 DAP 仅可以使用 SWD 访问整个系统地址映射，暂定不支持 JTAG 访问
- 处理器本身的 bus 仅有两种
 - 32-bit AMBA-3 AHB-Lite 系统接口，可以作为 master 访问所有系统外设和系统存储 32-bit 接口，可以作为 slave 接口被 DAP 访问

7.2. 系统嘀嗒（SysTick）

系统嘀嗒校准值固定为 4000，系统嘀嗒时钟固定设为 HCLK。当 SysTick 时钟为 32MHz（HCLK 的最大值），SysTick 产生 10ms 时间基准。更关于 systick 的内容请参考 ARM 内核官方手册。

7.3. 系统功能框图

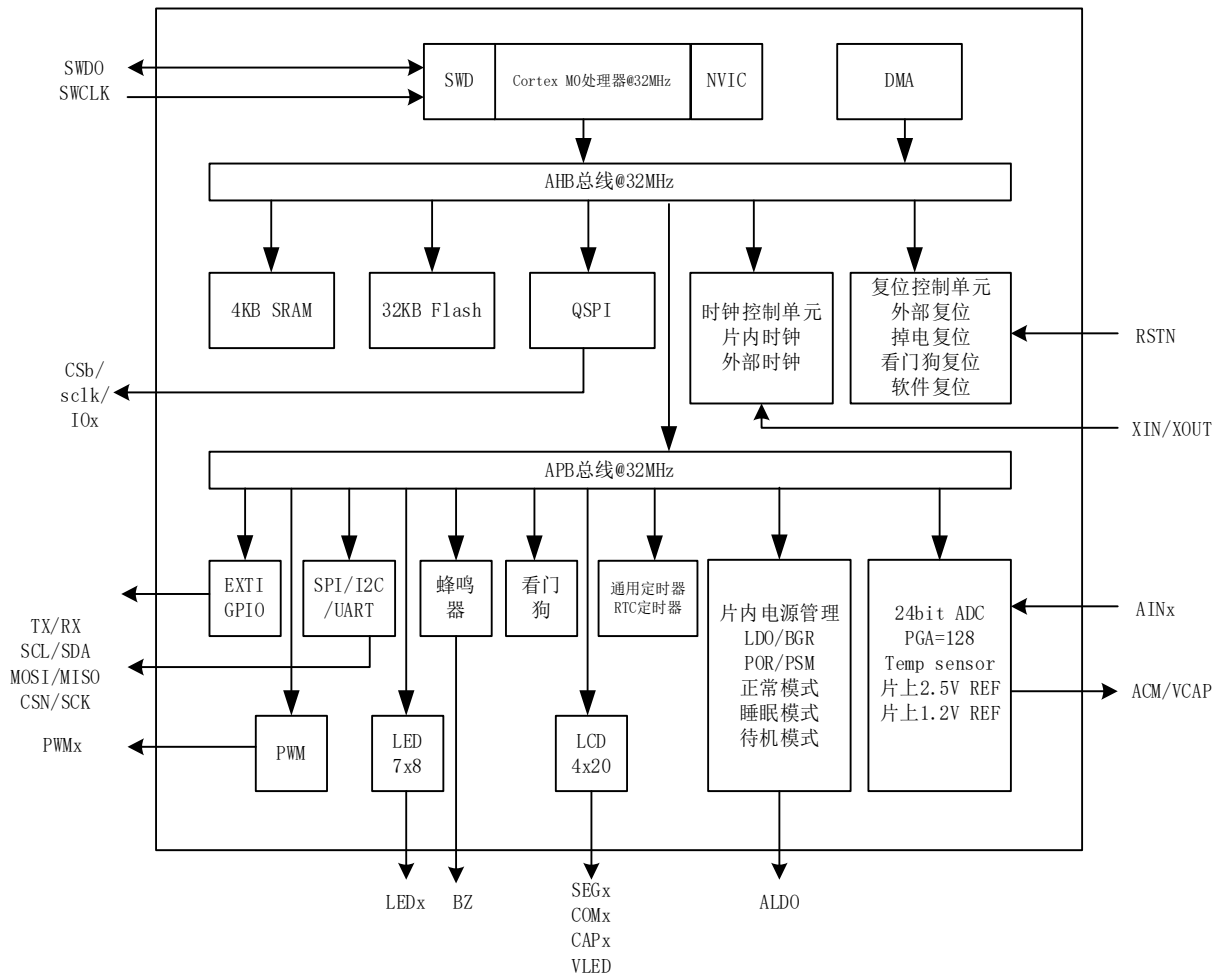


图4. 系统功能框图

系统包括 M0 内核作为中央处理单元，支持 SWD 接口、NVIC 中断处理、Systick 等功能。存储器包括 4KB 的 SRAM、128K 的 Flash 以及 QSPI 接口可控制片外 QSPI flash。时钟复位单元控制器各个模块的时钟产生、分频、关闭，时钟来自内部 32M OSC (HSI)、内部低频 32K OSC (LSI)、外部 32K XTAL (LSE)。模拟外设包括 ADC、LDO、参考电压源、LCD driver、LED driver 等，数字外设包括 I2C、SPI、UART 等串口，计时模块包括通用定时器、唤醒定时器、PWM、看门狗，此外还支持蜂鸣器、LCD 控制、LED 控制器。还支持最多支持 8 路外部中断。DMA 可直接从 QSPI、Flash、SRAM 读取数据并写入 SPI 送给外部显示模块。集成 24 位高精度 Sigma-Delta ADC，支持最大 128 倍程控 PGA 输入放大器。

7.4. 系统地址映射

程序存储器、数据存储器、寄存器和输入输出端口被组织在同一个 4GB 的线性地址空间内。数据字节以小端格式存放在存储器中。一个字里的最低地址字节被认为是该字的最低有效字节，而最高地址字节是最高有效字节。外设寄存器的映像请参考相关章节。可访问的存储器空间被分成 8 个主要块，每个块为 512MB。其他所有没有分配给片上存储器和外设的存储器空间都是保留的地址空间，具体如下表所示。

表4.

预定义的区域	总线	地址范围	外设
外设	AHB	0x4002 3000 - 0xC002 2FFF	QSPI Flash (2GB)
		0x4002 1000 - 0x4002 13FF	RCC
		0x4002 0000 - 0x4002 03FF	DMA1
	APB2	0x4001 3800 - 0x4001 3BFF	USART1
		0x4001 3000 - 0x4001 33FF	SPI1
		0x4001 2C00 - 0x4001 2FFF	TIMER1
		0x4001 2800 - 0x4001 2BFF	ADC
		0x4001 1800 - 0x4001 1BFF	DISPLAY
		0x4001 0800 - 0x4001 0BFF	GPIO
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	FLASH CTRL
	APB1	0x4000 7400 - 0x4000 77FF	AFE

预定义的区域	总线	地址范围	外设
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 5400 - 0x4000 57FF	I2C1
		0x4000 5000 - 0x4000 53FF	BEEPER
		0x4000 3C00 - 0x4000 3FFF	QSPI Reg
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2800 - 0x4000 2BFF	RTC
		0x4000 0000 - 0x4000 03FF	TIMER2
SRAM	AHB	0x2000 0000 - 0x2000 1000	SRAM (4KB)
CODE	AHB	0x0000 0000 - 0x0002 0000	Main Flash 128KB

7.5. 嵌入式 SRAM

内置 2K 字节的静态 SRAM。它可以以字节、半字（16 位）或全字（32 位）访问。SRAM 的起始地址是 0x2000 0000。

7.6. 嵌入式 Flash

7.6.1. Flash 控制器功能点

Flash 控制器使用 1 块 128K byte 的 Flash 组成 128K byte 的主存储区。

Flash 的页大小为 512B，基于 32bit 的位宽进行访问。可以通过锁住 SWD 接口来提供一种 Flash 读写保护的机制。

可以使用 24bit 的签名来对 Flash 的数据进行验证，这是通过为整个 Flash 空间或者部分 flash 空间的数据生成一个 24 位的 CRC 来实现的。当启动签名命令时，硬件自动计算选择区域的 CRC 结果，并和存储在选区域最后一个 page 的最高地址的签名结果进行对比。

7.6.2. Flash 用户空间

Flash 用户空间用来存储用户的数据和程序，其中一小部分用来存储控制读写保护的选项字节和签名。

在 Flash 用户空间的最后一页的最高 24 字节地址空间里存储了签名、用户写保护字以及用户失效分析密码 UserFaaKey。

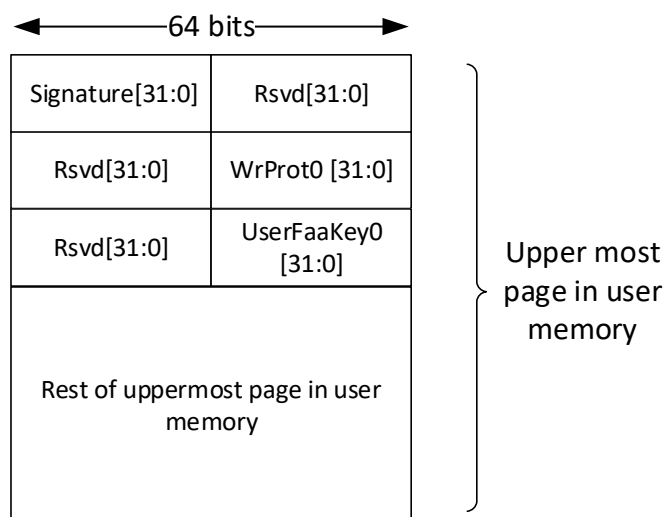


图5. Flash 用户空间最后一页内容

Signature[31:0]: 存储一页或者多页的原始签名数据，作为签名检查的对比数据。

- WrProt:将用户空间平均分成 32 个相同大小的空间，32 位写保护字的每一个 bit 对应其中一个地址空间的写保护权限。

- UserFAAKey:用户失效分析密码，由用户写入该地址。

对于不是最后一页的存储空间来说，每一页只有最后 4 字节的地址可能存储了签名字，该签名字是基于所选择空间（一页或者多页）的内容计算出来的 CRC 结果，签名检查命令会从所选择空间的最后一页的最高 4 字节地址寻找该区域（一页或者多页）的签名，并和硬件自动计算的结果做比较。

7.6.3. Flash 保护和完整性检查

7.6.3.1. Flash 密码

Key (UserKEY): (User key, 0x789A_0123) -这个密码通过 32 位的 KEY 寄存器写入的。当需要运行某些用户命令 (ERASEPAGE, SIGN, MASSERASE, ABORT, and SLEEP) 或者写入每个 flash block 的倒数第二个和三个地址 (WrProt 和 UserFAAKey) 时, 就需要提前写入这个密码。如果写入 WrProt 寄存器 (PROT/PRO1), Key 将会自动清除; 或者给 CON0 寄存器写入了命令, 一旦命令开始执行, Key 也将自动清除。如果是烧写每个 flash block 的倒数第二个和三个地址 (WrProt 和 UserFAAKey), 烧写完成后就需要手动清除 Key, 清除的方法为给 Key 寄存器写入任意一个非 0x789A_0123 的数。

UserFAAKey: 用户失效分析密码。有时候即使 Flash 读保护被使能, 也需要将用户的芯片拿回设计厂商进行用户失效分析, UserFAAKey 就可以用来对已经读保护的 flash 进行失效分析。用户需要提前已经在 Flash 的 UserFAAKey 地址分别写入了 32bit 的 key 值。对于 Flash 来说 UserFAAKey 地址为 0x1FFF4。用户需要将这两个地址的密码告诉领慧公司, 这样领慧公司就可以访问用户代码进行相应的 debug 分析。

7.6.3.2. Flash 读保护

设置 CR 寄存器中的 DebugEn 位为 0 可锁住 SWD 接口, 从而通过 SWD 的写操作被忽视, 读操作返回 0。

7.6.3.3. 用户写保护

Flash 的用户写保护是为了防止对用户代码空间不可预料的误写。如果一小片 Flash 区域被 PROx (PROT or PRO1) 的写保护字 bit 位保护起来了, 那么对这一小片区域的任何地址的写都不起作用。地址 0x1FFF0 的 32bit 数据控制 Flash0 的 32 小片区域的写保护。0x1FFF8 中的每一个 bit 可控制的存储区间大小为 128KB/32=4KB=8 页。

复位后 Flash 控制器自动从 Flash 的相应地址读取写保护字并复制到内部寄存器里 (寄存器 PROT, Bits[31:0])。如果要编程 Flash 的写保护字地址, 需要提前给 KEY 寄存器写入用户 key 0x789A_0123。一旦写保护字被写入, 只有对用户空间执行完 Mass erase 或者擦除 flash 最后一页 (前提是最后一页没有被写保护) 才能再次对写保护字地址进行编程。做完擦除操作后, 要对芯片进行复位, 这样才会重新从 Flash 读取全 1 的写保护字。

下面的步骤具体说明了如何编程写保护字:

1. 确保用户空间的最后一页已经被擦除了
2. 给 KEY 寄存器写入 user key---0x789A_0123
3. 写 32 位的数据到地址 0x1FFF8, 相应位是 0 表示使能该 4KB 区域的写保护。例如地址 0x1FFF8 的 bit2 和 bit3 如果是 0, 那么地址 0x2000 到 0x3FFF 就被写保护。
4. 查询烧写 0x1FFF8 已经完成 (寄存器查询或者中断)
5. 复位芯片, 写保护字自动被 load 并生效

如果写保护字没有被编程, 即 0x1FFF8 中的数据都是 0xFFFFFFFF, 复位后所有用户空间都没有被写保护。可以直接通过 APB 总线配置寄存器 PROT, PROT 寄存器每一个 bit 也对应了 flash 每一小片的写保护配置, 效果和 flash macro 中的 PROT 写保护字相同。如果没有提前输入 user key 就写入 PROT 写保护字, 会产生 CMDFAIL 标志。写保护字的 LSB 对应最低的用户空间, MSB 对应最高的用户空间, 为了防止写保护字被擦除, 应该将 MSB 写为 0 以启动对用户空间最后一页的写保护, 因为写保护字就在最后一页。

7.6.3.4. 签名 (Signature)

签名是为了检查 flash 的数据完整性, 签名检查是基于 32bit 数据宽度, 32bit 的数据不断的被送进 CRC 多项式直到最后一个地址的数据为止, 签名的计算结果是 CRC 中的余数。如果计算的签名和 flash 中存储的签名不一致就认为是检查失败。软件可以随时通过寄存器调用签名检查命令或者在执行一段新的区域代码之前执行签名检查验证该区域的数据完整性, 签名检查的 24bit CRC 多项式为 $x^{24} + x^{23} + x^6 + x^5 + x + 1$ 。

签名检查可以是单页也可以是连续的多页, 将单页或者连续的多页称为一个 block。一个 block 的最后一页的最高地址存储的 32bit 数据为该 block 的签名, 因此签名的产生不包含这 32bit 数据。签名检查之前应该保证签名已经被写入了正确的位置。

下面的步骤说明了如何进行签名检查:

1. 给寄存器 PADDRS 写入 block 的起始地址

2. 给寄存器 PADDRE 写入 block 的结束地址
3. 给命令寄存器写入 sign command
4. 命令执行完成后签名计算结果会存到 signature register, 将 signature register 中的结果和 block 的最后一页的最高地址存储的 32bit 数据做对比, 如果不匹配, 状态寄存器就会返回一个状态。

7.6.4. Flash 控制器性能和命令周期

通常情况下 flash 操作完整的一个动作的时间要慢于 CPU 执行一个指令的时间, 不仅 flash 读操作, 其他操作要更多的执行时间。典型的 flash 操作所需要的时间如下表所示。

软件应该使用 flash 状态寄存器的信息或者中断信息去判断某一个操作是否执行完成了。如果 flash 正在执行某一个操作, 那么其他操作要等待当前操作完成后再执行, 例如 CPU 发出的读指令要一直等待当前 flash 写操作完成。

不能给同一个地址编程超过一次, 如果需要重新写入新值, 需要至少做页擦除。

7.6.5. Flash 寄存器

Register Name: SR, Address: 0x40010000, Default: 0x0

表5. 状态寄存器

Bit Name	Position	Description	Default	Access
BUSY	[0]	当前正在执行 flash 命令时置 1	0x0	RW
COMP	[2]	命令结束, 读清	0x0	RW
WRALCOMP	[3]	写操作即将完成	0x0	RW
CMDFAIL	[4:5]	命令失败状态 0: 命令或写入成功 1: 操作非法或保护地址, 命令被忽略 2: 验证错误 3: 命令被中止	0x0	RW
SLEEP	[31]	休眠状态 0: Flash 处于正常状态 1: Flash 处于休眠状态	0x0	RW

Register Name: IER, Address: 0x40010004, Default: 0x0

表6. 中断使能寄存器

Bit Name	Position	Description	Default	Access
CMD	[0]	命令完成中断使能 0: 关闭中断 1: 使能中断	0x0	RW
WRALCOMP	[1]	写操作即将完成中断使能	0x0	RW
ERR	[2]	命令执行失败中断使能	0x0	RW

Register Name: CMD, Address: 0x40010008, Default: 0x0

表7. 命令寄存器

Bit Name	Position	Description	Default	Access
CMD	[0:4]	写入并开始执行相应命令 0: 空闲 1: 页擦除命令。Page Erase. 写地址到 PADDRS 指定要擦除的页。该命令需要写入 flash key 2: 签名命令。使用该命令给 flash 的一段地址产生签名。该段地址由 PageAddr0 和 PageAddr1 指定。产生的签名可以通过 SIGN 寄存器读取。该命令需要先写入 Flash Key. 注意: 每个 Page 的最后一个 Word 用来存放签名, 产生签名时不计算在内 4: 写命令。执行写命令前需要先写入相应的地址和数据寄存器 5: 整片擦除命令。ChipErase. 该命令需要先写入 Flash Key 8: 中止命令。写入该命令后任何正在执行的命令将被中止。该命令需要先写入 Flash Key	0x0	RW

Register Name: ADR, Address: 0x4001000c, Default: 0x0

表8. 地址寄存器

Bit Name	Position	Description	Default	Access
ADDRESS	[0:18]	地址寄存器, 最低 2 位始终为 0, 写入无效。字对齐	0x0	RW

Register Name: DATA, Address: 0x40010010, Default: 0x0

表9. 数据寄存器

Bit Name	Position	Description	Default	Access
DATA	[0:31]	写入 Flash 的数据	0x0	RW

Register Name: PADDRS, Address: 0x40010018, Default: 0x0

表10. 页开始地址寄存器

Bit Name	Position	Description	Default	Access
ADDRESS	[0:17]	页操作起始地址。低 9 位为 0, 写入无效。1Page=512Byte.页对齐	0x0	RW

Register Name: PADDRE, Address: 0x4001001c, Default: 0x0

表11. 页结束地址寄存器

Bit Name	Position	Description	Default	Access
ADDRESS	[0:17]	页操作结束地址。低 9 位为 0, 写入无效。1Page=512Byte.页对齐	0x0	RW

Register Name: KEY, Address: 0x40010020, Default: 0x0

表12. KEY 寄存器

Bit Name	Position	Description	Default	Access
KEY	[0:31]	UserKey=0x789A0123	0x0	RW

Register Name: WS, Address: 0x40010024, Default: 0x0

表13. Wait State 寄存器

Bit Name	Position	Description	Default	Access
WAIT	[0:2]	WAIT 值需要满足: $HCLK\ period * (WAIT + 1) \geq 40ns$	0x0	RW

Register Name: PROT, Address: 0x40010028, Default: 0x0

表14. 写保护寄存器

Bit Name	Position	Description	Default	Access
PROT	[0:31]	任意 bit 为 0 使能相应 page 的写保护	0x0	RW

Register Name: SIGN, Address: 0x40010034, Default: 0x0

表15. 签名寄存器

Bit Name	Position	Description	Default	Access
SIGN	[0:31]	签名结果	0x0	RW

Register Name: CR, Address: 0x40010038, Default: 0x0

表16. 控制寄存器

Bit Name	Position	Description	Default	Access
DEBUGEN	[0]	使能 SWD 调试 0: 关闭 SWD 调试接口 1: 使能 SWD 调试接口	0x0	RW
DMAEN	[1]	使能 Flash DMA 0: 关闭 DMA 1: 使能 DMA	0x0	RW
AUTOINC	[2]	使能地址自增加模式, DMA 使能时该位自动置 1	0x0	RW

8. 芯片唯一 ID

用户可以通过读取多个寄存器来识别芯片的唯一 ID 标识, 该标识共 85BIT, 可用于区分任意两颗芯片。

Register Name: SIL_ID0, Address: 0x40010104, Default: 0xFFFFFFFF

表17. ID0

Bit Name	Position	Description	Default	Access
LOT0	[15:0]	ID 的 0-15 位	0XXXXX	R

Register Name: SIL_ID1, Address: 0x40010108, Default: 0xFFFFFFFF

表18. ID1

Bit Name	Position	Description	Default	Access
LOT1	[15:0]	ID 的 16-31 位	0XXXXX	R

Register Name: SIL_ID2, Address: 0x4001010C, Default: 0xFFFFFFFF

表19. ID2

Bit Name	Position	Description	Default	Access
LOT2	[15:0]	ID 的低 32-47 位	0XXXXX	R

Register Name: SIL_ID3, Address: 0x40010110, Default: 0xFFFFFFFF

表20. ID3

Bit Name	Position	Description	Default	Access
X	[15:0]	ID 的 ID 的低 48-63 位	0XXXXX	R

Register Name: SIL_ID4, Address: 0x40010114, Default: 0xFFFFFFFF

表21. ID4

Bit Name	Position	Description	Default	Access
Y	[15:0]	ID 的 ID 的低 64-79 位	0XXXXX	R

Register Name: SIL_ID5, Address: 0x40010118, Default: 0xFFFFFFFF

表22. ID5

Bit Name	Position	Description	Default	Access
WID	[4:0]	ID 的 ID 的低 80-84 位	0XXXXX	R

9. 系统时钟和复位 (RCC)

三种不同的时钟源可被用来驱动系统时钟 (SYSCLK):

- HSI (内部 32M OSC) 振荡器时钟
- 32kHz 低速内部 RC (LSI)
- 32kHz 低速外部晶振 (LSE)

32kHz 可以用于驱动独立看门狗和 Wakeup Timer。Wakeup Timer 用于从停机模式下自动唤醒系统。

当不被使用时, 任一个时钟源都可被独立地启动或关闭, 由此优化系统功耗。

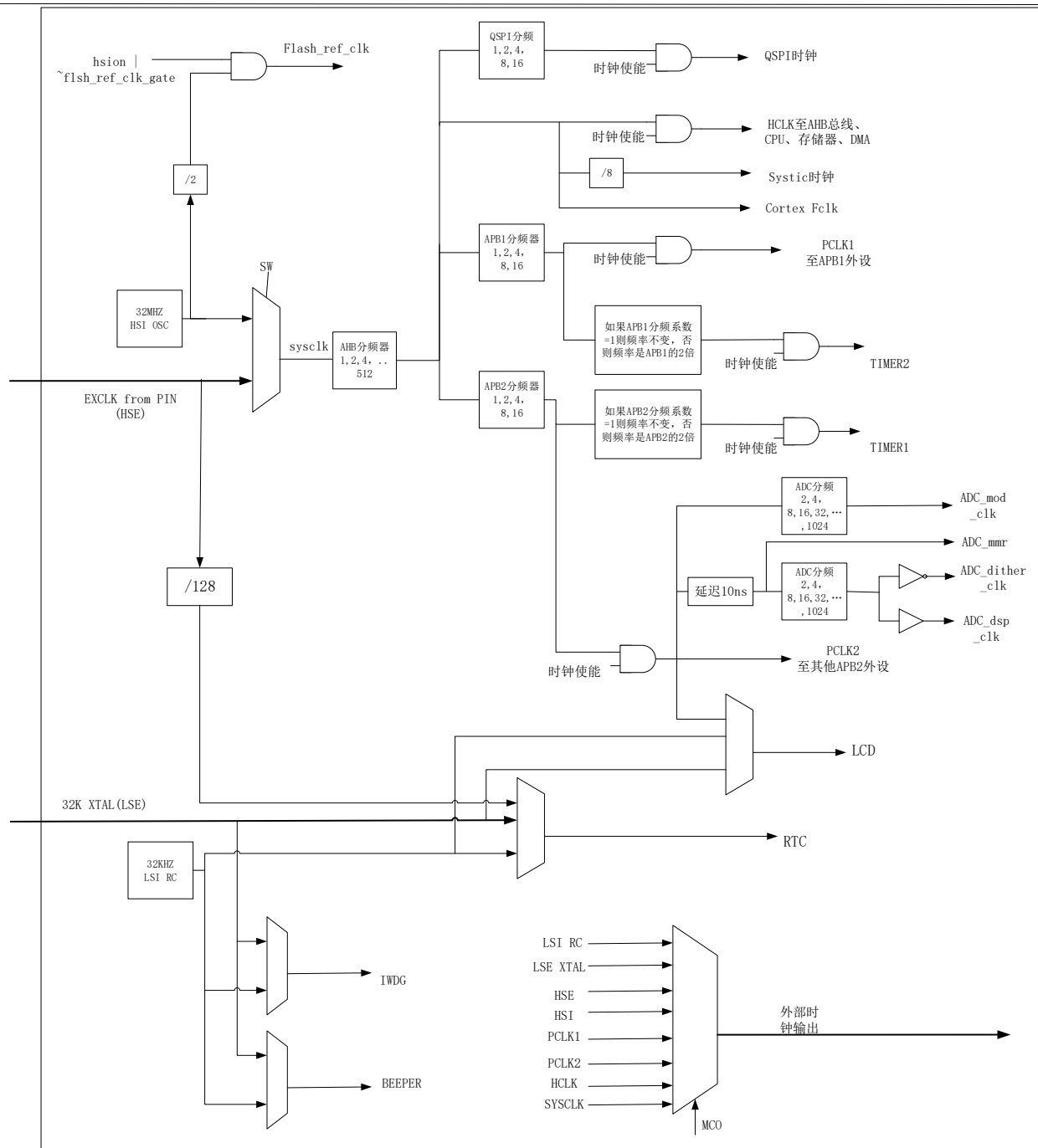


图6. 系统时钟树

用户可通过多个预分频器配置 AHB、APB2 和 APB1 域的频率。最大允许频率是 32MHz。

RCC 通过 AHB 时钟 (HCLK) 8 分频后作为 Cortex 系统定时器 (SysTick) 的外部时钟。通过对 SysTick 控制与状态寄存器的设置, 可选择上述时钟或 Cortex (HCLK) 时钟作为 SysTick 时钟。

定时器 (高级定时器 1 和通用定时器 2) 时钟频率分配由硬件按以下 2 种情况自动设置:

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

FCLK 是 Cortex™-M0 的自由运行时钟

9.1. HSE 时钟

外部管脚可为系统提供更为精确的主时钟, 最高 32MHz。当需要使用 HSE 时, 需要配置 PIN_MUX 使得外部时钟可从 P1.6 或者 P2.7 的 AF3 功能输入。

9.2. HSI 时钟

HSI 时钟信号由内部 32MHz 的 RC 振荡器产生，可直接作为系统时钟。HSI RC 振荡器能够在不需要任何外部器件的条件下提供系统时钟。校准制造工艺决定了不同芯片的 RC 振荡器频率会不同，系统复位时，工厂校准值被装载到时钟控制寄存器的 HSICAL[7:0]位。

HSI RC 可由时钟控制寄存器中的 HSION 位来启动和关闭。

9.3. LSI/LSE 时钟

LSI/LSE 担当低功耗时钟源的角色，它可以在停机模式下保持运行，为独立看门狗、RTC、BEEP、LCD/LED 提供时钟。LSI/LSE 时钟频率大约 32kHz。LSI RC 为常开的。

在控制/状态寄存器 (RCC_CSR) 里的 LSIRDY 位指示低速内部振荡器是否稳定。在启动阶段，直到这个位被硬件设置为 1 后，此时钟才被释放。如果在时钟中断寄存器 (RCC_CIR) 里被允许，将产生 LSI 中断申请。

9.4. 系统时钟选择

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

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

在时钟控制寄存器 (RCC_CR) 里的状态位指示哪个时钟目前被用作系统时钟。

9.5. 看门狗/RTC/BEEP/LCD 时钟

LSI 振荡器将被强制在打开状态，并且不能被关闭。在 LSI 或者 LSE 振荡器稳定后，时钟供应给上述模块。

9.6. 时钟输出

微控制器允许输出时钟信号到外部 MCO 引脚。相应的 GPIO 端口寄存器必须被配置为相应功能。以下七个时钟信号可被选作 MCO 时钟：

- SYSCLK
- HSI
- HSE
- LSI
- LSE
- PCLK1
- PCLK2
- HCLK

时钟的选择由时钟配置寄存器 (RCC_CFGR) 中的 MCO[2:0]位控制。

9.7. 复位架构 (Reset)

支持 2 种复位形式，分别为系统复位、电源复位。

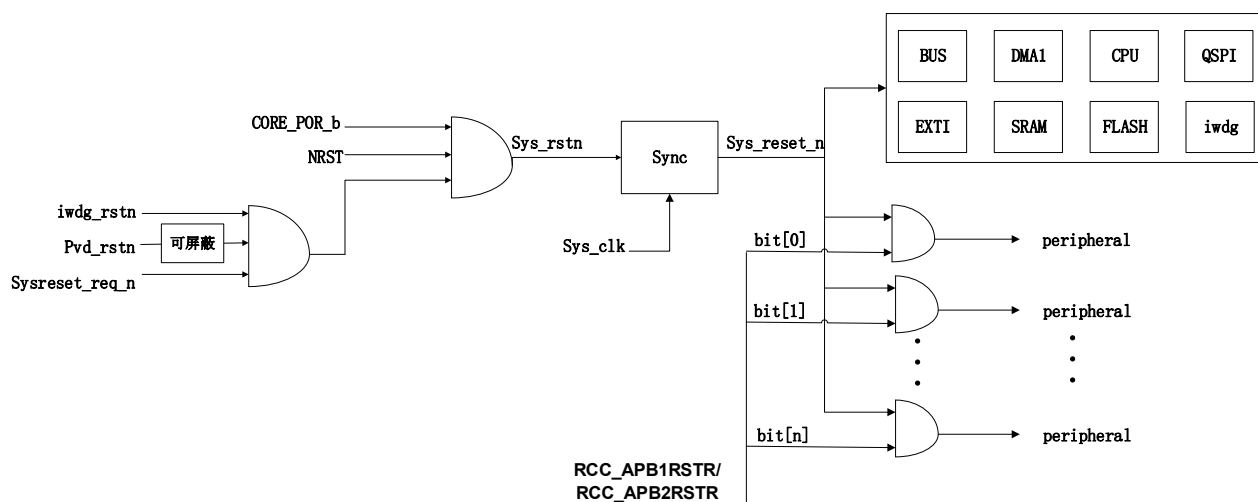


图7. 复位架构

9.7.1. 系统复位

除了时钟控制器的 RCC_CSR 寄存器中的复位标志位，以及开启了复位保持功能的模块外（GPIO, ADC, AFE mmr 的复位保持功能只允许这三个模块的内部寄存器被上电复位和外部管脚复位，并且复位保持使能寄存器只能被上电复位和外部管脚复位），系统复位将复位所有寄存器至它们的复位状态。

当发生以下任一事件时，产生一个系统复位：

- 1) NRST 引脚上的低电平（外部复位）
- 2) 低电压检测复位（PVD 复位）
- 3) 独立看门狗计数终止（IWDG 复位）
- 4) 软件复位（SW 复位）

低电压检测复位（PVD 复位）先经过 80us 左右的滤波之后才进入数字系统产生复位信号。可通过查看 RCC_CSR 控制状态寄存器中的复位状态标志位识别复位事件来源。通过将 Cortex™-M0 中断应用和复位控制寄存器中的 SYSRESETREQ 位置'1'，可实现软件复位。

表23.

类别	作用
上电、下电复位	全局复位
外部复位	全局复位
看门狗/PVD 复位	全局或非全局复位（软件配置）
软件复位	全局或非全局复位（软件配置）

软件可对芯片进行配置，使其在看门狗或 PVD 或软件发生复位时 GPIO 和模拟输出保留输出电压或电流不变。因此，即使发生自身软件、PVD、看门狗复位，产品也能保持功能不变。

表24. 复位保持说明

复位	影响					
	复位 pins	执行 kernel	复位 MMR 除了 RSTSTA	复位所有的外设备	SRAM	复位发生标记
软件复位	Yes/No ¹	yes	Yes/No ¹	Yes/No ¹	No	Yes
看门狗/PVD 复位	Yes/No ¹	yes	Yes/No ¹	Yes/No ¹	No	Yes
外部复位	Yes	yes	yes	yes	No	Yes
上电下电复位	Yes	yes	yes	yes	Yes	Yes

1. GPIO 和模拟外设在看门狗、PVD 和软件复位器件，可由软件配置是否被复位。

9.7.2. 电源复位

当以下事件中发生时，产生电源复位：上电/掉电复位（POR/PDR 复位）。

复位源将最终作用于 RESET 引脚，并在复位过程中保持低电平。复位入口矢量被固定在地址 0x0000_0004。当 NRST 引脚被拉低产生外部复位时，它将产生复位脉冲。

9.8. RCC 寄存器

Register Name: RCC_CR, Address: 0x40021000, Default: 0x1

表25. 时钟控制寄存器

Bit Name	Position	Description	Default	Access
HSION	[0]	内部高速时钟使能 (Internal high-speed clock enable). 由软件置'1'或清零。当从待机和停止模式返回或用作系统时钟的外部 4-16MHz 振荡器发生故障时, 该位由硬件置'1' 来启动内部 8MHz 的 RC 振荡器。当内部 8MHz 振荡器被直接或间接地用作或被选择将要作为系统时钟时, 该位不能被清零。 0: 内部 32MHz 振荡器关闭 1: 内部 32MHz 振荡器开启	0x1	RW
PVDRSTEN	[2]	低电压复位使能 (PVD reset enable) 由软件置'1'或清'0' 0x0: 低电压复位不使能 0x1: 低电压复位使能	0x0	RW

Register Name: RCC_CFGR, Address: 0x40021004, Default: 0x0

表26. 时钟配置寄存器

Bit Name	Position	Description	Default	Access
SW	[0:1]	系统时钟切换 (System clock switch)。由软件置'1'或清'0'来选择系统时钟源。在从停止模式中返回时, 由硬件强制选择 HSI 作为系统时钟 0: HSI 作为系统时钟; 1: HSE 作为系统时钟; 2: 不可用。	0x0	RW
SWS	[2:3]	系统时钟切换状态 (System clock switch status) 。由硬件置'1'或清'0'来指示哪一个时钟源被作为系统时钟。 0: HSI 作为系统时钟; 1: HSE 作为系统时钟; 2: 不可用。	0x0	RW
HPRE	[4:7]	AHB 预分频 (AHB Prescaler)由软件置'1'或清'0'来控制 AHB 时钟的预分频系数。0b0xxx: SYSCLK 不分频; 0b1000: SYSCLK 2 分频; 0b1001: SYSCLK 4 分频 0b1010 SYSCLK 8 分频 0b1011 SYSCLK 16 分频;0b1100:SYSCLK 64 分频;0b1101:SYSCLK 128 分频;0b1110:SYSCLK 256 分频;0b1111:SYSCLK 512 分频 0:	0x0	RW
PPRE1	[8:10]	低速 APB 预分频(APB1) (APB low-speed prescaler (APB1))。由软件置'1'或清'0'来控制低速 APB1 时钟(PCLK1)的预分频系数。警告: 软件必须保证 APB1 时钟频率不超过 36MHz 。0b0xx:HCLK 不分频 ;0b100:HCLK 2 分频;0b101:HCLK 4 分频;0b110:HCLK 8 分频;0b111:HCLK 16 分频	0x0	RW
PPRE2	[11:13]	高速 APB 预分频(APB2) (APB high-speed prescaler (APB2)) 由软件置'1'或清'0'来控制高速 APB2 时钟(PCLK2)的预分频系数。0b0xx:HCLK 不分频;0b100:HCLK 2 分频;0b101:HCLK 4 分频;0b110:HCLK 8 分频;0b111:HCLK 16 分频	0x0	RW
MCO	[24:26]	微控制器时钟输出 (Microcontroller clock output) 由软件置'1'或清零。 0x0: OSC32K (LSI) 时钟输出 0x1: 系统时钟(SYSCLK)输出 0x2: 内部 RC 振荡器时钟(HSI)输出 0x3: 外部管脚时钟(HSE)输出 0x4: XTAL 32K (LSE) 输出 0x5: PCLK1 输出 0x6: PCLK2 输出 0x7: HCLK 输出	0x0	RW
ADCDIV	[28:31]	ADC 预分频 (ADC prescaler) 0: PCLK2 2 分频后作为 ADC dither、adc dsp 时钟 1: PCLK2 4 分频后作为 ADC dither、adc dsp 时钟 2: PCLK2 8 分频后作为 ADC dither、adc dsp 时钟 3: PCLK2 16 分频后作为 ADC dither、adc dsp 时钟 4: PCLK2 32 分频后作为 ADC dither、adc dsp 时钟 5: PCLK2 64 分频后作为 ADC dither、adc dsp 时钟 6: PCLK2 128 分频后作为 ADC dither、adc dsp 时钟 7: PCLK2 256 分频后作为 ADC dither、adc dsp 时钟 8: PCLK2 512 分频后作为 ADC dither、adc dsp 时钟 9: PCLK2 1024 分频后作为 ADC dither、adc dsp 时钟 10: PCLK2 2 分频后作为 ADC dither、adc dsp 时钟	0x0	RW

Register Name: RCC_CIR, Address: 0x40021008, Default: 0x1

表27. 时钟中断寄存器

Bit Name	Position	Description	Default	Access
LSIRDYF	[0]	LSI 就绪中断标志 (LSI ready interrupt flag) 在内部低速时钟就绪且 LSIRDYIE 位被置'1'时, 由硬件置'1'。由软件通过置'1' LSIRDYC 位来清除。 0x0: 无内部 40kHz RC 振荡器产生的时钟就绪中断 0x1: 内部 40kHz RC 振荡器导致时钟就绪中断	0x1	R(ReadOnly)
LSIRDYIE	[8]	LSI 就绪中断使能 (LSI ready interrupt enable) 由软件置'1'或清'0'来使能或关闭内部 40kHz RC 振荡器就绪中断。 0x0: LSI 就绪中断关闭 0x1: LSI 就绪中断使能	0x0	RW
LSIRDYC	[1]	清除 LSI 就绪中断 (LSI ready interrupt clear) 由软件置'1'来清除 LSI 就绪中断标志位 LSIRDYF。 0x0: 无作用 0x1: 清除 LSI 就绪中断标志位 LSIRDYF	0x0	RW

Register Name: RCC_APB2RSTR, Address: 0x4002100c, Default: 0x0

表28. APB2 外设复位寄存器

Bit Name	Position	Description	Default	Access
IOPRST	[2]	IO 端口复位 (IO port reset)。由软件置'1'或清'0' 0x0: 无作用 0x1: 复位 IO 端口	0x0	RW
DISPRST	[6]	LCD/LED 复位 (MDIO reset)。由软件置'1'或清'0' 0x0: 无作用 0x1: 复位 LCD/LED	0x0	RW
ADC1RST	[9]	ADC1 接口复位 (ADC 1 interface reset)。由软件置'1'或清'0' 0x0: 无作用 0x1: 复位 ADC1 接口	0x0	RW
TIM1RST	[11]	TIM1 复位 (timer 1 reset)。由软件置'1'或清'0' 0x0: 无作用 0x1: 复位 TIM1	0x0	RW
SPI1RST	[12]	SPI1 复位 (SPI 1 reset)。由软件置'1'或清'0' 0x0: 无作用 0x1: 复位 SPI1	0x0	RW
USART1RST	[14]	USART1 复位 (USART1 reset)。由软件置'1'或清'0' 0x0: 无作用 0x1: 复位 USART1	0x0	RW

Register Name: RCC_APB1RSTR, Address: 0x40021010, Default: 0x0

表29. APB1 外设复位寄存器

Bit Name	Position	Description	Default	Access
TIM2RST	[0]	定时器 2 复位 (Timer 2 reset)。由软件置'1'或清'0' 0x0: 无作用 0x1: 复位 TIM2 定时器	0x0	RW
QSPIRST	[15]	QSPI 复位 (QSPI reset)。由软件置'1'或清'0' 0x0: 无作用 0x1: 复位 QSPI	0x0	RW
BEEPRST	[20]	BEEP 复位 (BEEP reset)。由软件置'1'或清'0' 0x0: 无作用 0x1: 复位 BEEP	0x0	RW
I2C1RST	[21]	I2C 1 复位 (I2C 1 reset)。由软件置'1'或清'0' 0x0: 无作用 0x1: 复位 I2C 1	0x0	RW
BKPRST	[27]	AON 接口复位 (Backup interface reset)。由软件置'1'或清'0' 0x0: 无作用 0x1: 复位 AON 接口	0x0	RW
PWRRST	[28]	电源接口复位 (Power interface reset)。由软件置'1'或清'0' 0x0: 无作用 0x1: 复位电源接口	0x0	RW
AFERST	[29]	AFE 接口复位 (AFE interface reset)。由软件置'1'或清'0' 0x0: 无作用 0x1: 复位 AFE 接口	0x0	RW

Register Name: RCC_AHBENR, Address: 0x40021014, Default: 0x4

表30. AHB 外设时钟使能寄存器

Bit Name	Position	Description	Default	Access
DMA1EN	[0]	DMA1 时钟使能 (DMA1 clock enable)。由软件置'1'或清'0'。 0x0: DMA1 时钟关闭	0x0	RW

Bit Name	Position	Description	Default	Access
		0x1: DMA1 时钟开启		
SRAMEN	[2]	SRAM 时钟使能 (SRAM interface clock enable) 由软件置'1'或清'0'来开启或关闭睡眠模式时 SRAM 时钟。 0x0: 睡眠模式时 SRAM 时钟关闭 0x1: 睡眠模式时 SRAM 时钟开启	0x1	RW
FILTFEN	[4]	闪存接口电路时钟使能 (FLITF clock enable). 由软件置'1'或清'0'来开启或关闭睡眠模式时闪存接口电路时钟。 0: 睡眠模式时闪存接口电路时钟关闭 1: 睡眠模式时闪存接口电路时钟开启	0x0	RW

Register Name: RCC_APB2ENR, Address: 0x40021018, Default: 0x0

表31. APB2 外设时钟使能寄存器

Bit Name	Position	Description	Default	Access
IOPEN	[2]	IO 端口时钟使能 (I/O port clock enable) 由软件置'1'或清'0' 0x0: IO 端口时钟关闭 0x1: IO 端口时钟开启	0x0	RW
DISPEN	[6]	LCD/LED 时钟使能 (LCD/LED clock enable) 由软件置'1'或清'0' 0x0: LCD/LED 时钟关闭 0x1: LCD/LED 时钟开启	0x0	RW
ADC1EN	[9]	ADC1 接口时钟使能 (ADC 1 interface clock enable) 由软件置'1'或清'0' 0x0: ADC1 时钟关闭 0x1: ADC1 时钟开启	0x0	RW
TIM1EN	[11]	Timer1 时钟使能 (timer 1 clock enable) 由软件置'1'或清'0' 0x0: timer1 时钟关闭 0x1: timer1 时钟开启	0x0	RW
SPI1EN	[12]	SPI1 时钟使能 (SPI 1 clock enable) 由软件置'1'或清'0' 0x0: SPI1 时钟关闭 0x1: SPI1 时钟开启	0x0	RW
USART1EN	[14]	USART1 时钟使能 (SPI 1 clock enable) 由软件置'1'或清'0' 0x0: USART1 时钟关闭 0x1: USART1 时钟开启	0x0	RW

Register Name: RCC_APB1ENR, Address: 0x4002101c, Default: 0x0

表32. APB1 外设时钟使能寄存器

Bit Name	Position	Description	Default	Access
TIM2EN	[0]	定时器 2 时钟使能 (Timer 2 clock enable) 由软件置'1'或清'0' 0x0: 定时器 2 时钟关闭 0x1: 定时器 2 时钟开启	0x0	RW
QSPIEN	[15]	QSPI 时钟使能 (QSPI clock enable) 由软件置'1'或清'0' 0x0: QSPI 时钟关闭 0x1: QSPI 时钟开启	0x0	RW
BEEPEN	[20]	BEEP 时钟使能 (BEEP clock enable) 由软件置'1'或清'0' 0x0: BEEP 时钟关闭 0x1: BEEP 时钟开启	0x0	RW
I2C1EN	[21]	I2C1 时钟使能 (I2C1 clock enable) 由软件置'1'或清'0' 0x0: I2C1 时钟关闭 0x1: I2C1 时钟开启	0x0	RW
BKPEN	[27]	AON 时钟使能 (SPI 1 clock enable) 由软件置'1'或清'0' 0x0: AON 时钟关闭 0x1: AON 时钟开启	0x0	RW
PWREN	[28]	电源接口时钟使能 (Power interface clock enable) 由软件置'1'或清'0' 0x0: 电源接口时钟关闭 0x1: 电源接口时钟开启	0x0	RW
AFEEN	[29]	AFE 接口时钟使能 (AFE interface clock enable) 由软件置'1'或清'0' 0x0: AFE 接口时钟关闭	0x0	RW

Bit Name	Position	Description	Default	Access
		0x1: AFE 接口时钟开启		

Register Name: RCC_BDCR, Address: 0x40021020, Default: 0x0

AON 域控制寄存器中的位在复位后处于写保护状态, 只有在电源控制寄存器(PWR_CR)中的 DBP 位置'1'后才能对这些位进行改动。这些位只能由 AON 域和上电复位清除。

表33. AON 域控制寄存器。

Bit Name	Position	Description	Default	Access
LSEON	[0]	LSEON: 外部低速振荡器使能 (External low-speed oscillator enable) 由软件置'1'或清'0' 0x0: 0: 外部 32kHz 振荡器关闭; 0x1: 1: 外部 32kHz 振荡器开启。	0x0	RW
IWDGCLKSEL	[3]	iwdg 时钟源选择 由软件置'1'或清'0' 0x0: 选择 LSI 0x1: 选择 LSE	0x0	RW
BEEPCLKSEL	[4]	beeper 时钟源选择 由软件置'1'或清'0' 0x0: 选择 LSI 0x1: 选择 LSE 0x0: 选择 pclk2 0x1: 选择 LSI 0x2: 选择 LSE	0x0	RW
DISPCLKSEL	[5:6]	LCD 时钟源选择 0: PCLK2 1: LSI 2: LSE	0x0	RW
RTCSEL	[8:9]	RTC 时钟源选择 (RTC clock source selection) 由软件设置来选择 RTC 时钟源。一旦 RTC 时钟源被选定, 直到下次 AON 域被复位, 它不能再被改变。可通过设置 BDRST 位来清除。 0x0: 无时钟 0x1: LSE XTAL 作为 RTC 时钟 0x2: LSI 振荡器作为 RTC 时钟 0x3: HSE 振荡器在 128 分频后作为 RTC 时钟	0x0	RW
RTCCEN	[15]	RTC 时钟使能 (RTC clock enable) 由软件置'1'或清'0' 0x0: RTC 时钟关闭 0x1: RTC 时钟开启	0x0	RW
BDRST	[16]	AON 域软件复位 (Backup domain software reset) 由软件置'1'或清'0' 0x0: 复位未激活 0x1: 复位整个 AON 域	0x0	RW

Register Name: RCC_CSR, Address: 0x40021024, Default: 0x2

除复位标志外由系统复位清除, 复位标志只能由电源复位清除。

表34. 控制/状态寄存器。

Bit Name	Position	Description	Default	Access
LSIRDY	[1]	内部低速振荡器就绪 (Internal low-speed oscillator ready) 由硬件置'1'或清'0'来指示内部 40kHz RC 振荡器是否就绪。在 LSION 清零后, 3 个内部 40kHz RC 振荡器的周期后 LSIRDY 被清零。 0x0: 内部 40kHz RC 振荡器时钟未就绪 0x1: 内部 40kHz RC 振荡器时钟就绪。	0x1	R(ReadOnly)
GPIO_RETAIN	[2]	当看门狗或者软件复位发生后, 是否保持 GPIO 的状态.且该位只能被上电复位或者外部引脚复位 0x0: 当看门狗或者软件复位发生后, GPIO 的状态保持 0x1: 当看门狗或者软件复位发生后, GPIO 的状态被复位	0x0	RW
ANA_RETAIN	[3]	当看门狗或者软件复位发生后, 是否保持 ADC 和 AFE 的状态.且该位只能被上电复位或者外部引脚复位 0x0: 当看门狗或者软件复位发生后, ADC 和 AFE 的状态保持 0x1: 当看门狗或者软件复位发生后, ADC 和 AFE 的状态被复位	0x0	RW
RMVF	[24]	清除复位标志 (Remove reset flag) 由软件置'1'来清除复位标志。 0x0: 无作用 0x1: 清除复位标志	0x0	RW
PVDRSTF	[25]	PVD 复位标志 (PVD reset flag) 在 PVD 复位发生时由硬件置'1'; 由软件通过写 RMVF 位清除。 0x0: 无 PVD 复位发生 0x1: 发生 PVD 复位	0x0	RW

Bit Name	Position	Description	Default	Access
PINRSTF	[26]	NRST 引脚复位标志 (PIN reset flag) 在 NRST 引脚复位发生时由硬件置'1'; 由软件通过写 RMVF 位清除。 0x0: 无 NRST 引脚复位发生 0x1: 发生 NRST 引脚复位	0x0	R(ReadOnly)
PORRSTF	[27]	上电/掉电复位标志 (POR/PDR reset flag) 在上电/掉电复位发生时由硬件置'1'; 由软件通过写 RMVF 位清除。 0x0: 无上电/掉电复位发生 0x1: 发生上电/掉电复位	0x0	R(ReadOnly)
SFTRSTF	[28]	软件复位标志 (Software reset flag) 在软件复位发生时由硬件置'1'; 由软件通过写 RMVF 位清除。 0x0: 无软件复位发生 0x1: 发生软件复位	0x0	R(ReadOnly)
IWDGRSTF	[29]	独立看门狗复位标志 (Independent watchdog reset flag) 在独立看门狗复位发生时由硬件置'1'; 由软件通过写 RMVF 位清除。 0x0: 无独立看门狗复位发生 0x1: 发生独立看门狗复位	0x0	R(ReadOnly)

10. 工作模式 (Power Mode)

10.1. 电源管理

10.1.1. 上电复位 (POR) 和掉电复位 (PDR)

内部有一个完整的上电复位 (POR) 和掉电复位 (PDR) 电路, 当供电电压达到 2V 时系统既能正常工作。当 V_{DD}/V_{DDA} 低于指定的限位电压 V_{POR}/V_{PDR} 时, 系统保持为复位状态, 而无需外部复位电路。关于上电复位和掉电复位的细节请参考数据手册的电气特性部分。

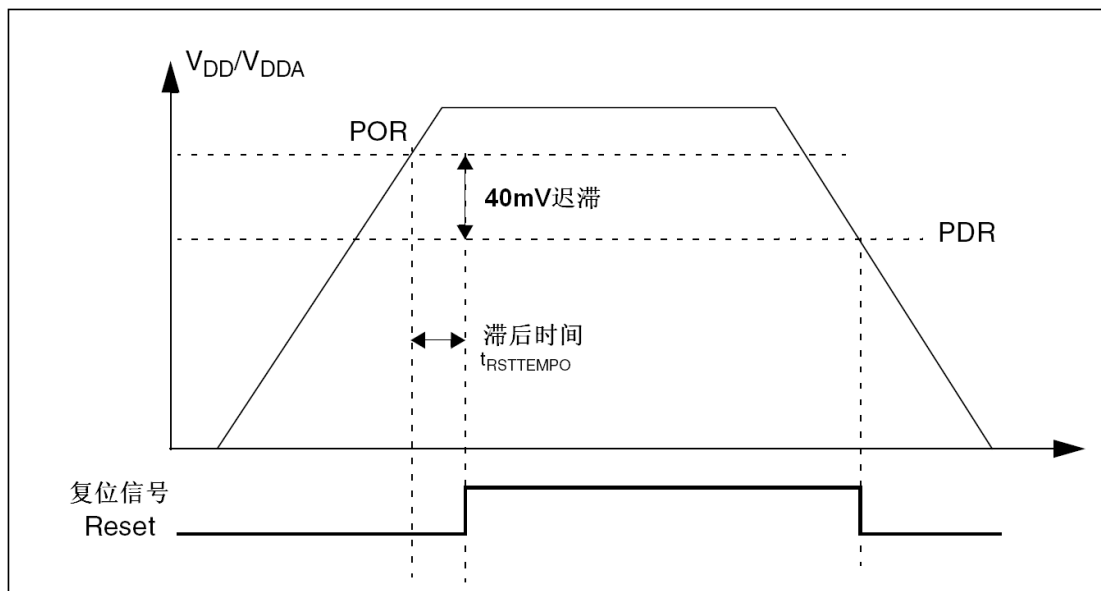


图8. 上电复位和掉电复位的波形

10.1.2. 可编程电压监测器 (PVD)

用户可以利用 PVD 对 V_{DD} 电压与电源控制寄存器 (PWR_CR) 中的 $PLS[2:0]$ 位进行比较来监控电源, 这几位选择监控电压的阈值。通过设置 $PVDE$ 位来使能 PVD。电源控制/状态寄存器 (PWR_CSR) 中的 $PVDO$ 标志用来表明 V_{DD} 是高于还是低于 PVD 的电压阈值。该事件在内部连接到外部中断的第 16 线, 如果该中断在外部中断寄存器中是使能的, 该事件就会产生中断。当 V_{DD} 下降到 PVD 阈值以下和 (或) 当 V_{DD} 上升到 PVD 阈值之上时, 根据外部中断第 16 线的上升/下降边沿触发设置, 就会产生 PVD 中断。例如, 这一特性可用于用于执行紧急关闭任务。

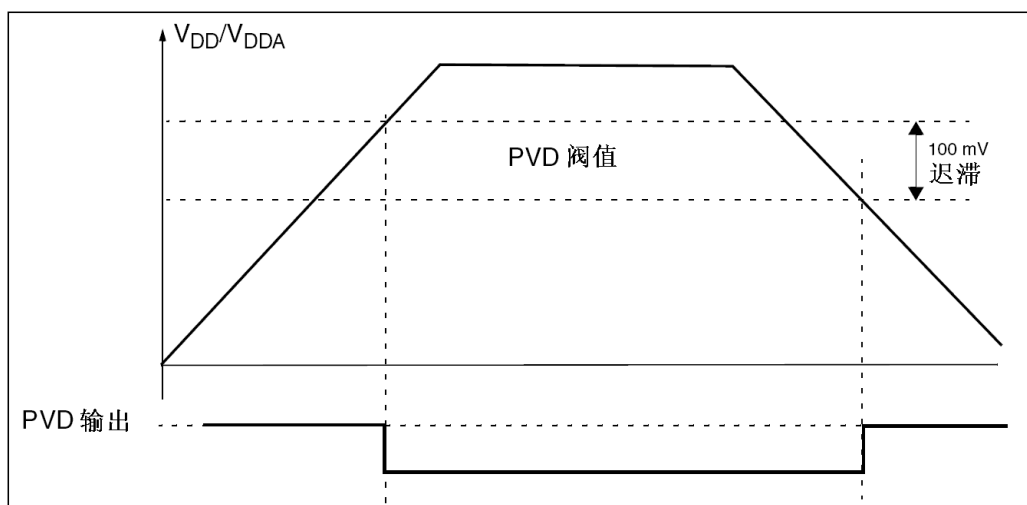


图9. PVD 的门限

10.1.3. Always on (AON) 区域

当进入低功耗模式后, AON 区域可以维持 RTC 的功能。

- P2_3 可以作为通用 I/O 口、RTC 校准时钟、RTC 闹钟或秒输出（参见：AON 寄存器）

10.2. 低功耗模式

系统除了正常工作模式外，目前有 3 种低功耗模式：

- 睡眠模式（Cortex™-M0 内核停止，所有外设包括 Cortex-M0 核心的外设，如 NVIC、系统时钟（SysTick）等仍在运行）
 - 深度睡眠 1（除了 32K LSI 和 LSE 外，所有的时钟都已停止。不启用 low power LDO）
 - 深度睡眠 2（除了 32K LSI 和 LSE 外，所有的时钟都已停止。启用 low power LDO）

表35.

模式	进入方式	唤醒	对数字区域时钟的影响	对模拟区域的影响	保持情况
睡眠 (SLEEP-NO 或 SLEEP-ON- EXIT)	WFI	任一中断	CPU 时钟关，对其他时钟无影响	无	SRAM 和寄存器内容保持
	WFE	唤醒事件			
深度睡眠 1	SLEEPDEEP +WFI 或 WFE +lpds=0	任一外部中断/事件 +RTC（在外部中断寄 存器中设置） +iwdt	除 32K OSC (LSI) 和 32K xtal (LSE) 外，关闭所有数字区域的 时钟	32M OSC (HSI) 关 闭，启用 HPLDO	SRAM 和寄存器内容保持
深度睡眠 2	SLEEPDEEP +WFI 或 WFE +lpds=1	任一外部中断/事件 +RTC（在外部中断寄 存器中设置） +iwdt	除 32K OSC (LSI) 和 32K xtal (LSE) 外，关闭所有数字区域的 时钟	32M OSC (HSI) 关 闭，启用 LPLDO (最后关 HPLDO? ? ?)	SRAM 和寄存器内容保持

在进入深度睡眠模式 1 或者进入深度睡眠模式 2 前，用户软件还可以选择配置额外的寄存器去满足应用需求，如下表所示。

表36. 用户软件配置

CPCON[0]=1	Charge pump 使能深度睡眠模式可能工作，由客户决定是否配置
LCDCON[30]	深度睡眠模式 LCD 可能工作，由客户决定是否配置
LVDCON[0]	LVD 的使能控制深度睡眠可能工作，由客户决定是否配置
XTALCON[0]=1	XTAL 开启控制寄存器深度睡眠模式下可能会使用，由客户决定是否配置
ADCBUFCON[10]=1	ADC buffer 输入下拉使能开启
ANAPLT_CON[17]=1	VCM buffer 下拉使能开启

10.2.1. 降低系统时钟

在运行模式下，通过对预分频寄存器进行编程，可以降低任意一个系统时钟(SYSCLK、HCLK、PCLK1、PCLK2)的速度。进入睡眠模式前，也可以利用预分频器来降低外设的时钟。详见：时钟配置寄存器（RCC_CFGR）。

10.2.2. 外部时钟的控制

在运行模式下，任何时候都可以通过停止为外设和内存提供时钟（HCLK 和 PCLKx）来减少功耗。为了在睡眠模式下更多地减少功耗，可在执行 WFI 或 WFE 指令前关闭所有外设的时钟。通过设置 AHB 外设时钟使能寄存器（RCC_AHBENR）、APB2 外设时钟使能寄存器（RCC_APB2ENR）和 APB1 外设时钟使能寄存器（RCC_APB1ENR）来开关各个外设模块的时钟。

10.2.3. 睡眠模式

进入睡眠模式

通过执行 WFI 或 WFE 指令进入睡眠状态。根据 Cortex™-M0 系统控制寄存器中的 SLEEPONEXIT 位的值，有两种选项可用于选择睡眠模式进入机制：

- SLEEP-NOW：如果 SLEEPONEXIT 位被清除，当 WFI 或 WFE 被执行时，微控制器立即进入睡眠模式。
- SLEEP-ON-EXIT：如果 SLEEPONEXIT 位被置位，系统从最低优先级的中断处理程序中退出时，微控制器就立即进入睡眠模式。在睡眠模式下，所有的 I/O 引脚都保持它们在运行模式时的状态。

退出睡眠模式

如果执行 WFI 指令进入睡眠模式，任意一个被嵌套向量中断控制器响应的外设中断都能将系统从睡眠模式唤醒。如果执行 WFE 指令进入睡眠模式，则一旦发生唤醒事件时，微处理器都将从睡眠模式退出。唤醒事件可以通过下述方式产生：

- 在外设控制寄存器中使能一个中断，而不是在 NVIC（嵌套向量中断控制器）中使能，并且在 Cortex-M0 系统控制寄存器中使能 SEVONPEND 位。当 MCU 从 WFE 中唤醒后，外设的中断挂起位和外设的 NVIC 中断通道挂起位（在 NVIC 中断清除挂起寄存器中）必须被清除。
- 配置一个外部或内部的 EXIT 线为事件模式。当 MCU 从 WFE 中唤醒后，因为与事件线对应的挂起位未被设置，不必清除外设的中断挂起位或外设的 NVIC 中断通道挂起位。该模式唤醒所需的时间最短，因为没有时间损失在中断的进入或退出上。

10.2.4. 深度睡眠模式 1 和深度睡眠模式 2

深度睡眠模式 1 和深度睡眠模式 2 是在 Cortex™-M0 的深睡眠模式基础上结合了外设的时钟控制机制，在深度睡眠模式 1/2 下 LDO 可运行在正常或低功耗模式。此时在 1.8V 供电区域的所有时钟都被停止，HSI 和 HSE 的功能被禁止，SRAM 和寄存器内容被保留下来。在深度睡眠模式 1/2，所有的 I/O 引脚都保持它们在运行模式时的状态。关于如何进入深度睡眠模式 1/2，详见表 11。如果正在进行闪存编程，直到对内存访问完成，系统才进入深度睡眠模式 1/2。如果正在进行对 APB 的访问，直到对 APB 访问完成，系统才进入深度睡眠模式 1/2。可以通过对独立的控制位进行编程，可选择以下功能：

- 独立看门狗 (IWDG)：可通过写入看门狗的键寄存器或硬件选择来启动 IWDG。一旦启动了独立看门狗，除了系统复位，它不能再被停止。
- 实时时钟 (RTC)：通过 AON 域时钟控制寄存器 (RCC_BDCR) 的 RTCEN 位来设置。
- 内部 RC 振荡器 (LSI RC)：通过时钟控制/状态寄存器 (RCC_CSR) 的 LSION 位来设置。
- 外部 32.768kHz 振荡器 (LSE)：通过 AON 域时钟控制寄存器 (RCC_BDCR) 的 LSEON 位设置。

在深度睡眠模式 1/2 下，如果在进入该模式前 ADC 和 DAC 没有被关闭，那么这些外设仍然消耗电流。

10.2.4.1. 退出停止模式

关于如何退出深度睡眠模式 1/2，详见下表。当一个中断或唤醒事件导致退出深度睡眠模式 1/2 时，HSI RC 振荡器被选为系统时钟。

表37. 深度睡眠模式 1/2

深度睡眠模式 1 或 2	说明
进入	在以下条件下执行 WFI（等待中断）或 WFE（等待事件）指令： — 设置 Cortex-M0 系统控制寄存器中的 SLEEPDEEP 位 — 清除电源控制寄存器 (PWR_CR) 中的 PDM 位 — 通过设置 PWR_CR 中 LPDS 位为 0 选择深度睡眠模式 1，LPDS 位为 1 选择深度睡眠模式 2 注：为了进入该模式，所有的外部中断的请求位（挂起寄存器 (EXTI_PR)）和 RTC 的闹钟标志都必须被清除，否则该模式的进入流程将会被跳过，程序继续运行。
退出	如果执行 WFI 进入该模式： 设置任一外部中断线为中断模式（在 NVIC 中必须使能相应的外部中断向量）。 如果执行 WFE 进入停止模式： 设置任一外部中断线为事件模式。参见唤醒事件管理（第 7.4.3 节）。
唤醒延时	HSI RC 唤醒时间+ 低功耗 LDO 唤醒时间（深睡模式 2）

10.3. 低功耗模式下的自动唤醒

10.3.1. RTC timer 唤醒

RTC timer 可以在不需要依赖外部中断的情况下唤醒低功耗模式下的微控制器（自动唤醒模式）。RTC timer 提供一个可编程的时间基数，用于周期性从停止或待机模式下唤醒。选择时钟源中的 LSI、LSE、HSE/128 其一来实现此功能。

低功耗内部 RC 振荡器 (LSI RC)：使用该时钟源，节省了一个 32.768kHz 晶振的成本。但是 RC 振荡器将少许增加电源消耗。

为了用 RTC timer 闹钟事件将系统从停止模式下唤醒，必须进行如下操作：

- 1) 配置 EXTI 外部中断线 9 为上升沿触发。
- 2) 配置 RTC timer 使其可产生闹钟 Alarm 事件。

10.3.2. 独立看门狗唤醒

独立看门狗也会将系统停止或待机模式下唤醒，但也会将系统复位。可通过写入看门狗的键寄存器或硬件选择来启动 IWDG。一旦启动了独立看门狗，除了系统复位，它不能再被停止。

10.4. 电源控制寄存器

Register Name: CR, Address: 0x40007000, Default: 0x0

表38. 电源控制寄存器

Bit Name	Position	Description	Default	Access
LPDS	[0]	当 CPU 的 deepsleep 置起后, 该位决定进入深度睡眠 1 或者深度睡眠 2 0: 系统进入深度睡眠 1 1: 系统进入深度睡眠 2	0x0	RW
RESERVED	[1]	RESERVED	0x0	W(WriteOnly)
DBP	[8]	取消 AON 区域的写保护 在复位后, RTC 和 AON 寄存器处于被保护状态以防意外写入。设置这位允许写入这些寄存器。注: 如果 RTC 的时钟是 HSE/128, 该位必须保持为'1'。 0: 禁止写入 RTC 和 AON 寄存器 1: 允许写入 RTC 和 AON 寄存器	0x0	W(WriteOnly)

Register Name: CSR, Address: 0x40007004, Default: 0x0

表39. 电源控制/状态寄存器

Bit Name	Position	Description	Default	Access
LVDO	[2]	低电压检测 LVD 输出状态 当 LVD 被 AFE_MMR 模块中的 LVDCON_EN 位使能后该位才有效 0: VDD/VDDA 高于由 AFE_MMR 模块中的 LVDCON_REFSSEL 选定的 LVD 阈值 1: VDD/VDDA 低于由 AFE_MMR 模块中的 LVDCON_REFSSEL 选定的 LVD 阈值	0x0	R(ReadOnly)

11. 中断 (Interrupt)

11.1. 嵌套向量中断控制器

特性:

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

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

11.2. 外部中断/事件控制器 (EXTI)

外部中断/事件控制器由 11 个产生事件/中断请求的边沿检测器组成, 每个输入线可以独立地配置输入类型 (脉冲或挂起) 和对应的触发事件 (上升沿或下降沿或者双边沿都触发)。每个输入线都可以独立地被屏蔽。挂起寄存器保持着状态线的中断请求。 EXTI 最高一位对应的检测器的输入固定保持 0, 为保留位, 所以最高一位只可以产生软件中断请求。

11.2.1. 主要特性

EXTI 控制器的主要特性如下:

- 每个中断/事件都有独立的触发和屏蔽
- 每个中断线都有专用的状态位
- 支持多达 11 个软件的中断/事件请求
- 检测脉冲宽度低于 APB2 时钟宽度的外部信号。参见数据手册中电气特性部分的相关参数。

11.2.2. 框图

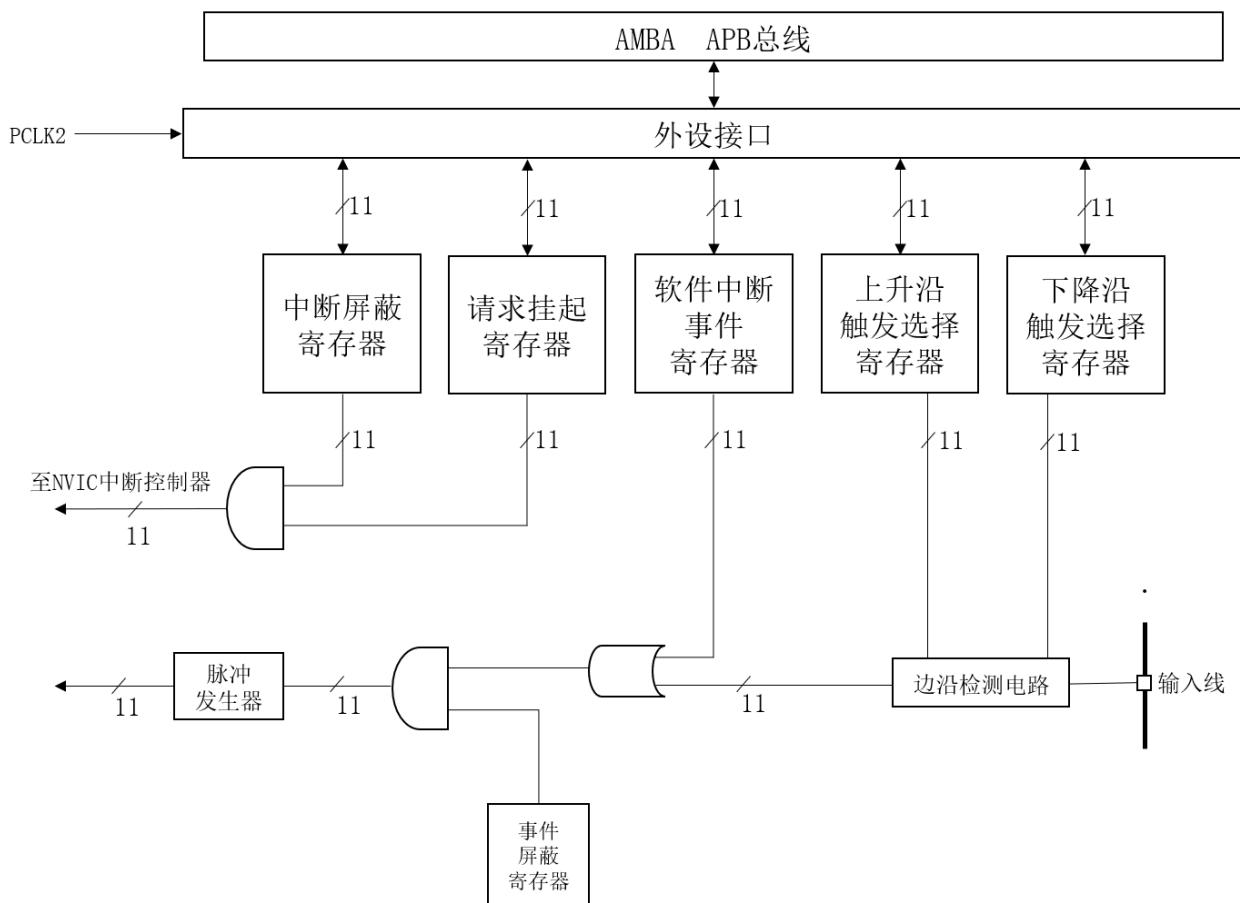


图10. 外部中断/事件控制器框图

11.2.3. 唤醒事件管理

MCU 可以处理外部或内部事件来唤醒内核（WFE）。唤醒事件可以通过下述配置产生：

- 在外设的控制寄存器使能一个中断，但不在 NVIC 中使能，同时在 Cortex-M0 的系统控制寄存器中使能 SEVONPEND 位。当 CPU 从 WFE 恢复后，需要清除相应外设的中断挂起位和外设 NVIC 中断通道挂起位（在 NVIC 中断清除挂起寄存器中）。
- 配置一个外部或内部 EXTI 线为事件模式，当 CPU 从 WFE 恢复后，因为 EXTI 中对应事件线的挂起位没有被置位，不必清除 EXTI 的相应外设的中断挂起位或 NVIC 中断通道挂起位。

使用外部 I/O 端口作为唤醒事件，请参见 7.4.4 节的功能说明

11.2.4. 功能说明

要产生中断，必须先配置好并使能中断线。根据需要的边沿检测设置 2 个触发寄存器，同时在中断屏蔽寄存器的相应位写'1'允许中断请求。当外部中断线上发生了期待的边沿时，将产生一个中断请求，对应的挂起位也随之被置'1'。在挂起寄存器的对应位写'1'，将清除该中断请求。如果需要产生事件，必须先配置好并使能事件线。根据需要的边沿检测通过设置 2 个触发寄存器，同时在事件屏蔽寄存器的相应位写'1'允许事件请求。当事件线上发生了需要的边沿时，将产生一个事件请求脉冲，对应的挂起位不被置'1'。通过在软件中断/事件寄存器写'1'，也可以通过软件产生中断/事件请求。

11.2.4.1. 硬件中断选择

通过下面的过程来配置 10 个线路做为硬件中断源：

- 配置 10 个中断线的屏蔽位（EXTI_IMR）
- 配置所选中断线的触发选择位（EXTI_RTISR 和 EXTI_FTISR）；
- 配置对应到外部中断控制器（EXTI）的 NVIC 中断通道的使能和屏蔽位，使得 10 个中断线中的请求可以被正确地响应。

11.2.4.2. 硬件事件选择

通过下面的过程，可以配置 10 个线路为硬件事件源

- 配置 10 个事件线的屏蔽位 (EXTI_EMR)
- 配置事件线的触发选择位 (EXTI_RTSR 和 EXTI_FTSR)

11.2.4.3. 软件中断/事件的选择

11 个线路可以被配置成软件中断/事件线。下面是产生软件中断的过程:

- 配置 11 个中断/事件线屏蔽位 (EXTI_IMR, EXTI_EMR)
- 设置软件中断寄存器的请求位 (EXTI_SWIER)

11.2.5. 外部中断/事件线路映像

通用 I/O 端口以下图的方式连接到 11 个外部中断/事件线上:

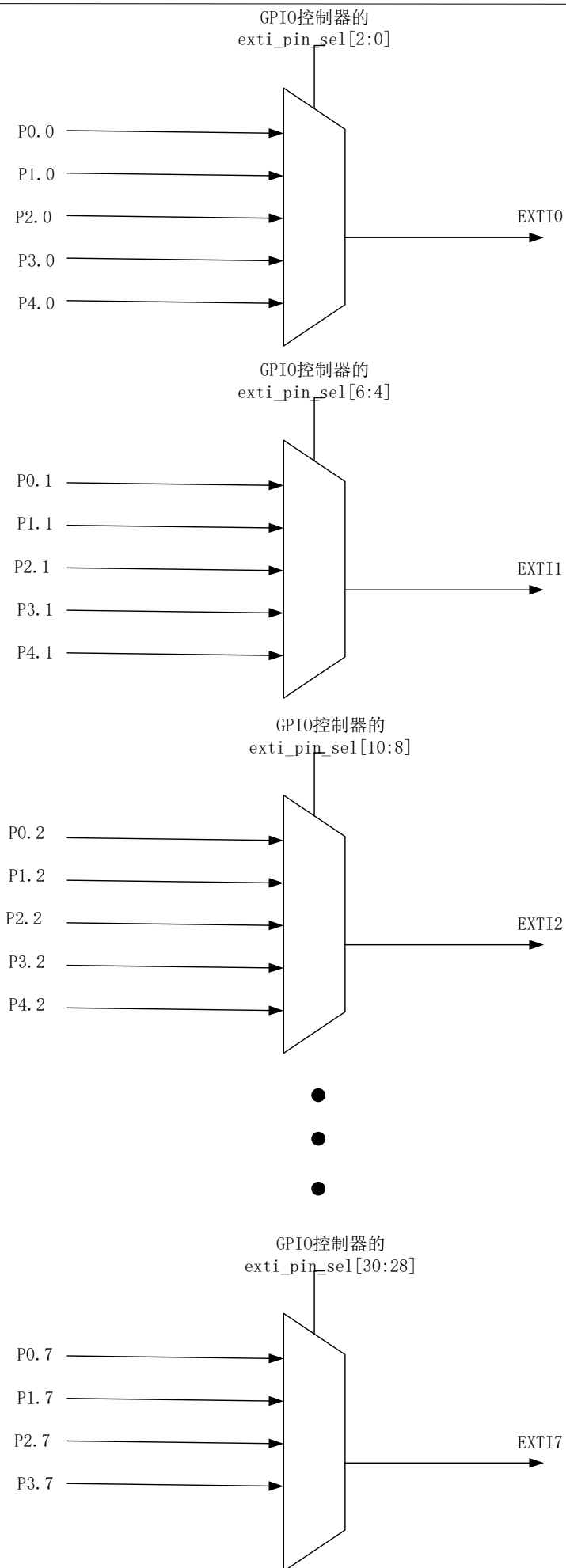


图11. 外部中断通用 I/O 映像

通过 GPIO 寄存器 exti_pin_sel 配置 GPIO 线上的外部中断/事件, 必须先使能 GPIO 时钟。

另外 3 个 EXTI 线的连接方式如下:

- EXTI8 连接到 PVD (电压低于阈值标记) 输出
- EXTI9 连接到 RTC 闹钟事件
- EXTI10 连接到 0, 所以 EXTI10 只能产生软件中断/事件

11.3. 中断寄存器

Register Name: EXTI_IMR, Address: 0x40010400, Default: 0x0

表40. 中断屏蔽寄存器

Bit Name	Position	Description	Default	Access
MRx	[0:10]	MRx: 线 x 上的中断屏蔽 (Interrupt Mask on line x) 0: 屏蔽来自线 x 上的中断请求; 1: 开放来自线 x 上的中断请求。	0x0	RW

Register Name: EXTI_EMR, Address: 0x40010404, Default: 0x0

表41. 事件屏蔽寄存器

Bit Name	Position	Description	Default	Access
MRx	[0:10]	MRx: 线 x 上的事件屏蔽 (Event Mask on line x) 0: 屏蔽来自线 x 上的事件请求; 1: 开放来自线 x 上的事件请求。	0x0	RW

Register Name: EXTI_RTSTR, Address: 0x40010408, Default: 0x0

表42. 上升沿触发选择寄存器

Bit Name	Position	Description	Default	Access
TRx	[0:10]	TRx: 线 x 上的上升沿触发事件配置位 (Rising trigger event configuration bit of line x) 0: 禁止输入线 x 上的上升沿触发(中断和事件) 1: 允许输入线 x 上的上升沿触发(中断和事件)	0x0	RW

Register Name: EXTI_FTSR, Address: 0x4001040c, Default: 0x0

表43. 下降沿触发选择寄存器

Bit Name	Position	Description	Default	Access
TRx	[0:10]	TRx: 线 x 上的下降沿触发事件配置位 (Falling trigger event configuration bit of line x) 0: 禁止输入线 x 上的下降沿触发(中断和事件) 1: 允许输入线 x 上的下降沿触发(中断和事件)	0x0	RW

Register Name: EXTI_SWIER, Address: 0x40010410, Default: 0x0

表44. 软件中断事件寄存器

Bit Name	Position	Description	Default	Access
SWIERx	[0:10]	SWIERx: 线 x 上的软件中断 (Software interrupt on line x) 当该位为'0'时, 写'1'将设置 EXTI_PR 中相应的挂起位。如果在 EXTI_IMR 和 EXTI_EMR 中允许产生该中断, 则此时将产生一个中断。 注: 通过清除 EXTI_PR 的对应位(写入'1'), 可以清除该位为'0'。	0x0	RW

Register Name: EXTI_PR, Address: 0x40010414, Default: 0x0

表45. 挂起寄存器

Bit Name	Position	Description	Default	Access
PRx	[0:10]	可读/写 1 清 PRx: 挂起位 (Pending bit) 0: 没有发生触发请求 1: 发生了选择的触发请求 当在外部中断线上发生了选择的边沿事件, 该位被置'1'。在该位中写入'1'可以清除它, 也可以通过改变边沿检测的极性清除。	0x0	W1C(Write1/auto-clear 0)

12. 通用和复用功能 I/O (GPIO and Pin Mux)

12.1. GPIO 功能描述

每个 GPIO 端口有功能选择寄存器，数据输入输出寄存器，控制寄存器等。根据数据手册中列出的每个 I/O 端口的特定硬件特征，GPIO 端口的每个位可以由软件分别配置成多种模式。

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟输入
- 开漏输出
- 推挽式输出
- 施密特输入使能
- 输入输出使能
- 驱动能力可配
- 功能复用

12.1.1. 通用 I/O (GPIO)

复位期间和刚复位后，复用功能未开启，I/O 端口被配置成浮空输入模式 (IE=1, P2,P1=00)。复位后，SW 引脚被置于输入上拉或下拉模式：

- P0_2: SWDAT 处于上拉模式
- P0_3: SWCLK 处于下拉模式

当作为输出配置时，写到输出数据寄存器上的值 (gpiox_dout) 输出到相应的 I/O 引脚。可以以推挽模式或开漏模式（当输出 0 时，只有 N-MOS 被打开）使用输出驱动器。输入数据寄存器 (gpiox_di) 在每个 APB2 时钟周期捕捉 I/O 引脚上的数据。所有 GPIO 引脚有一个内部弱上拉和弱下拉，当配置为输入时，它们可以被激活也可以被断开。

注意：P0.0-P0.7 以及 P1.0, P1.1, P1.2 输入使能 IE 不能被关闭，因此需要注意休眠时给这些引脚固定的电平或者使能内部上拉或者下拉，这样能够降低休眠功耗。

12.1.2. 单独的位设置或位清除

当对 gpiox_dout 的个别位编程时，软件不需要禁止中断：在单次 APB2 写操作里，可以只更改一个或多个位。这是通过对“置位/复位寄存器” (gpiox_ohset, gpiox_ohclr) 以及翻转寄存器 (gpiox_otgl) 中想要更改的位写‘1’来实现的。没被选择的位将不被更改。

12.1.3. 外部中断/唤醒线

所有端口都有外部中断能力。为了使用外部中断线，端口必须配置成输入模式。更多的关于外部中断的信息，参考：

- 第 7.4 节：外部中断/事件控制器 (EXTI)。
- 第 7.4.3 节：唤醒事件管理。

12.1.4. 复用功能 (AF)

当端口配置为 AFIO（设置 gpiox_con 寄存器中的相应 4bit 位为非 0 值）时，该端口用作外设备用功能。通过配置 GPIO 备用功能选择寄存器 (gpiox_con)，每个端口可以配置 6 个备用功能。端口备用功能分配的详细介绍见 pin_mux 表格。

12.1.5. 附加功能

有些引脚具有附加功能，它们优先于标准 GPIO 寄存器中的配置。当用作 ADC、DAC 和比较器附加功能时，引脚必须配置成模拟模式。当引脚用作 RTC 和振荡器附加功能时，端口类型通过相关的 RTC、PMU 和 RCU 寄存器自动设置。当附加功能禁用时，这些端口可用作普通 GPIO。

12.1.6. 输入配置

当 I/O 端口配置为输入时：

- 输出缓冲器使能应该被禁止

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

12.1.7. 输出配置

当 I/O 端口被配置为输出时，输出缓冲器使能应该被激活：

- 开漏模式 (OD=1)：输出寄存器上的'0'激活 N-MOS，而输出寄存器上的'1'将端口置于高阻状态（P-MOS 从不被激活）。
- 推挽模式 (OD=0)：输出寄存器上的'0'激活 N-MOS，而输出寄存器上的'1'将激活 P-MOS。

12.1.8. 复用功能配置

当 I/O 端口被配置为复用功能时：

- 内置外设的信号驱动输出缓冲器（复用功能输出）
 - 施密特触发输入被激活
 - 在每个 APB2 时钟周期，出现在 I/O 脚上的数据被采样到输入数据寄存器
- 一组复用功能 I/O 寄存器允许用户把一些复用功能重新映射到不同的引脚。

12.1.9. 模拟输入配置

当 I/O 端口被配置为模拟输入配置时：

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

12.2. GPIO 寄存器

GPIO0 的基址为 0x40010800

GPIO1 的基址为 0x40010840

GPIO2 的基址为 0x40010880

GPIO3 的基址为 0x400108C0

GPIO4 的基址为 0x40010900

下表详述 GPIO0 的寄存器内容，其他 GPIO 内容和偏移地址相同

Register Name: CON0, Address: 0x40010800, Default: 0x0

表46. GPIO0 端口复用选择寄存器 0

Bit Name	Position	Description	Default	Access
PORTO_0_SEL	[0:3]	复用功能选择 0x0: Choose alt function 0 0x1: Choose alt function 1 0x2: Choose alt function 2 0x3: Choose alt function 3 0x4: Choose alt function 4 0x5: Choose alt function 5 0x6: Choose alt function 6 0x7: Choose alt function 7	0x0	RW
PORTO_1_SEL	[4:7]	复用功能选择 0x0: Choose alt function 0 0x1: Choose alt function 1 0x2: Choose alt function 2 0x3: Choose alt function 3 0x4: Choose alt function 4 0x5: Choose alt function 5 0x6: Choose alt function 6 0x7: Choose alt function 7	0x0	RW
PORTO_2_SEL	[8:11]	复用功能选择 0x0: Choose alt function 0 0x1: Choose alt function 1 0x2: Choose alt function 2 0x3: Choose alt function 3 0x4: Choose alt function 4 0x5: Choose alt function 5	0x0	RW

Bit Name	Position	Description	Default	Access
		0x6: Choose alt function 6 0x7: Choose alt function 7		
PORT0_3_SEL	[12:15]	复用功能选择 0x0: Choose alt function 0 0x1: Choose alt function 1 0x2: Choose alt function 2 0x3: Choose alt function 3 0x4: Choose alt function 4 0x5: Choose alt function 5 0x6: Choose alt function 6 0x7: Choose alt function 7	0x0	RW
PORT0_4_SEL	[16:19]	复用功能选择 0x0: Choose alt function 0 0x1: Choose alt function 1 0x2: Choose alt function 2 0x3: Choose alt function 3 0x4: Choose alt function 4 0x5: Choose alt function 5 0x6: Choose alt function 6 0x7: Choose alt function 7	0x0	RW
PORT0_5_SEL	[20:23]	复用功能选择 0x0: Choose alt function 0 0x1: Choose alt function 1 0x2: Choose alt function 2 0x3: Choose alt function 3 0x4: Choose alt function 4 0x5: Choose alt function 5 0x6: Choose alt function 6 0x7: Choose alt function 7	0x0	RW
PORT0_6_SEL	[24:27]	复用功能选择 0x0: Choose alt function 0 0x1: Choose alt function 1 0x2: Choose alt function 2 0x3: Choose alt function 3 0x4: Choose alt function 4 0x5: Choose alt function 5 0x6: Choose alt function 6 0x7: Choose alt function 7	0x0	RW
PORT0_7_SEL	[28:31]	复用功能选择 0x0: Choose alt function 0 0x1: Choose alt function 1 0x2: Choose alt function 2 0x3: Choose alt function 3 0x4: Choose alt function 4 0x5: Choose alt function 5 0x6: Choose alt function 6 0x7: Choose alt function 7	0x0	RW

Register Name: OE, Address: 0x40010804, Default: 0x0

表47. GPIO0 输出使能寄存器

Bit Name	Position	Description	Default	Access
PORT0_OE	[0:7]	输出使能	0x0	RW
RESERVED	[8:31]	保留	0x0	RW

Register Name: PUE, Address: 0x40010808, Default: 0x0

表48. GPIO0 上拉使能寄存器

Bit Name	Position	Description	Default	Access
PORT0_PUE	[0:7]	上拉使能	0x0	RW
RESERVED	[8:31]	保留	0x0	RW

Register Name: PDE, Address: 0x4001080c, Default: 0x0

表49. GPIO0 下拉使能寄存器

Bit Name	Position	Description	Default	Access
PORT0_PDE	[0:15]	下拉使能	0x0	RW
RESERVED	[16:31]	保留	0x0	RW

Register Name: OD, Address: 0x40010810, Default: 0x0

表50. GPIO0 开漏使能寄存器

Bit Name	Position	Description	Default	Access
PORT0_OD	[0:7]	漏极开路使能	0x0	RW
RESERVED	[8:31]	保留	0x0	RW

Register Name: IE, Address: 0x40010814, Default: 0x0

表51. GPIO0 输入使能寄存器

Bit Name	Position	Description	Default	Access
PORT0_IE	[0:7]	输入使能	0x0	RW
RESERVED	[8:31]	保留	0x0	RW

Register Name: SMT, Address: 0x40010818, Default: 0x0

表52. GPIO0 施密特输入使能寄存器

Bit Name	Position	Description	Default	Access
PORT0_SMT	[0:7]	施密特输入使能	0x0	RW
RESERVED	[8:31]	保留	0x0	RW

Register Name: DR, Address: 0x40010820, Default: 0x0

表53. GPIO0 驱动能力选择寄存器

Bit Name	Position	Description	Default	Access
PORT0_0_DR	[0:3]	驱动能力选择 0x0: Maxium supooort 2mA current 0x1: Maxium supooort 4mA current 0x2: Maxium supooort 8mA current 0x3: Maxium supooort 16mA current	0x0	RW
PORT0_1_DR	[4:7]	驱动能力选择 0x0: Maxium supooort 2mA current 0x1: Maxium supooort 4mA current 0x2: Maxium supooort 8mA current 0x3: Maxium supooort 16mA current	0x0	RW
PORT0_2_DR	[8:11]	驱动能力选择 0x0: Maxium supooort 2mA current 0x1: Maxium supooort 4mA current 0x2: Maxium supooort 8mA current 0x3: Maxium supooort 16mA current	0x0	RW
PORT0_3_DR	[12:15]	驱动能力选择 0x0: Maxium supooort 2mA current 0x1: Maxium supooort 4mA current 0x2: Maxium supooort 8mA current 0x3: Maxium supooort 16mA current	0x0	RW
PORT0_4_DR	[16:19]	驱动能力选择 0x0: Maxium supooort 2mA current 0x1: Maxium supooort 4mA current 0x2: Maxium supooort 8mA current 0x3: Maxium supooort 16mA current	0x0	RW
PORT0_5_DR	[20:23]	驱动能力选择 0x0: Maxium supooort 2mA current 0x1: Maxium supooort 4mA current 0x2: Maxium supooort 8mA current 0x3: Maxium supooort 16mA current	0x0	RW
PORT0_6_DR	[24:27]	驱动能力选择 0x0: Maxium supooort 2mA current 0x1: Maxium supooort 4mA current 0x2: Maxium supooort 8mA current 0x3: Maxium supooort 16mA current	0x0	RW
PORT0_7_DR	[28:31]	驱动能力选择 0x0: Maxium supooort 2mA current 0x1: Maxium supooort 4mA current 0x2: Maxium supooort 8mA current 0x3: Maxium supooort 16mA current	0x0	RW

Register Name: DATA_IN, Address: 0x40010824, Default: 0x0

表54. GPIO0 输入数据寄存器

Bit Name	Position	Description	Default	Access
PORT0_DIN	[0:7]	输入数据	0x0	RW
RESERVED	[8:31]	保留	0x0	RW

Register Name: DATA_OUT, Address: 0x40010828, Default: 0x0

表55. GPIO0 输出数据寄存器

Bit Name	Position	Description	Default	Access
PORT0_OUT	[0:7]	输出数据	0x0	RW
RESERVED	[8:31]	保留	0x0	RW

Register Name: OSET, Address: 0x4001082c, Default: 0x0

表56. GPIO0 输出数据置位寄存器

Bit Name	Position	Description	Default	Access
PORT0_OSET	[0:7]	输出置 1	0x0	RW
RESERVED	[8:31]	保留	0x0	RW

Register Name: OCLR, Address: 0x40010830, Default: 0x0

表57. GPIO0 输出数据复位寄存器

Bit Name	Position	Description	Default	Access
PORT0_OCLR	[0:7]	输出清零	0x0	RW
RESERVED	[8:31]	保留	0x0	RW

Register Name: OTGL, Address: 0x40010834, Default: 0x0

表58. GPIO0 输出数据翻转寄存器

Bit Name	Position	Description	Default	Access
PORT0_OTGL	[0:7]	输出翻转	0x0	RW
RESERVED	[8:31]	保留	0x0	RW

Register Name: SL, Address: 0x4001081c, Default: 0x0

表59. GPIO0 压摆率

Bit Name	Position	Description	Default	Access
PORT0_SL	[0:7]	0:fast 1:slow	0x0	RW
RESERVED	[8:31]		0x0	RO

Register Name: INT_SEL, Address: 0x4001093c, Default: 0x0

表60. 外中断管脚选择寄存器,注意: 该寄存器只有这一个地址, 并不是每个 Port 都有。

Bit Name	Position	Description	Default	Access
INT_SEL_BIT_3_0	[0:3]	3'b000:P0_0 作为外中断 EXTI0 的源 3'b001:P1_0 作为外中断 EXTI0 的源 3'b010:P2_0 作为外中断 EXTI0 的源 3'b011:P3_0 作为外中断 EXTI0 的源 3'b100:P4_0 作为外中断 EXTI0 的源 其他 : P0_0 作为外中断 EXTI0 的源	0x0	RW
INT_SEL_BIT_7_4	[4:7]	3'b000:P0_1 作为外中断 EXTI1 的源 3'b001:P1_1 作为外中断 EXTI1 的源 3'b010:P2_1 作为外中断 EXTI1 的源 3'b011:P3_1 作为外中断 EXTI1 的源 3'b100:P4_1 作为外中断 EXTI1 的源 其他 : P0_1 作为外中断 EXTI1 的源	0x0	RW
INT_SEL_BIT_11_8	[8:11]	3'b000:P0_2 作为外中断 EXTI2 的源 3'b001:P1_2 作为外中断 EXTI2 的源 3'b010:P2_2 作为外中断 EXTI2 的源 3'b011:P3_2 作为外中断 EXTI2 的源 3'b100:P4_2 作为外中断 EXTI2 的源 其他 : P0_2 作为外中断 EXTI2 的源	0x0	RW
INT_SEL_BIT_15_12	[12:15]	3'b000:P0_3 作为外中断 EXTI3 的源 3'b001:P1_3 作为外中断 EXTI3 的源 3'b010:P2_3 作为外中断 EXTI3 的源 3'b011:P3_3 作为外中断 EXTI3 的源 其他 : P0_3 作为外中断 EXTI3 的源	0x0	RW
INT_SEL_BIT_19_16	[16:19]	3'b000:P0_4 作为外中断 EXTI4 的源 3'b001:P1_4 作为外中断 EXTI4 的源 3'b010:P2_4 作为外中断 EXTI4 的源 3'b011:P3_4 作为外中断 EXTI4 的源 其他 : P0_4 作为外中断 EXTI4 的源	0x0	RW
INT_SEL_BIT_23_20	[20:23]	3'b000:P0_5 作为外中断 EXTI5 的源 3'b001:P1_5 作为外中断 EXTI5 的源 3'b010:P2_5 作为外中断 EXTI5 的源 3'b011:P3_5 作为外中断 EXTI5 的源	0x0	RW

Bit Name	Position	Description	Default	Access
		其他：P0_5 作为外中断 EXTI5 的源		
INT_SEL_BIT_27_24	[24:27]	3'b000:P0_6 作为外中断 EXTI6 的源 3'b001:P1_6 作为外中断 EXTI6 的源 3'b010:P2_6 作为外中断 EXTI6 的源 3'b011:P3_6 作为外中断 EXTI6 的源 其他：P0_6 作为外中断 EXTI6 的源	0x0	RW
INT_SEL_BIT_31_28	[28:31]	3'b000:P0_7 作为外中断 EXTI7 的源 3'b001:P1_7 作为外中断 EXTI7 的源 3'b010:P2_7 作为外中断 EXTI7 的源 3'b011:P3_7 作为外中断 EXTI7 的源 其他：P0_7 作为外中断 EXTI7 的源	0x0	RW

13. 直接内存访问 (DMA)

直接存储器存取 (DMA) 用来提供在外设和存储器之间或者存储器和存储器之间的高速数据传输。无须 CPU 干预，数据可以通过 DMA 快速地移动，这就节省了 CPU 的资源来做其他操作。

在基本模式下 DMA 控制器只能响应一个硬件请求或者软件请求，当想要服务于某个请求源时，配置好 DMA 内部的配置寄存器（源地址、目的地址、数目、地址增量、中断使能等等），并且只写寄存器打开想要服务的请求使能。

13.1. DMA 主要特性

基本模式下 DMA 控制器只能响应一个硬件请求或者软件请求

- 每个请求源都直接连接专用的硬件 DMA 请求，每个请求源都同样支持软件触发。这些功能在基本模式下通过软件来配置，链表模式下从链表读取配置。

- 基本模式直接配置寄存器

- 链表模式下，SRAM 中可以有多个链表，每个链表包含不定数目的子表，DMA 先载入第一个子表的内容到寄存器，执行完第一个子表的 DMA 再载入第二个子表的内容到寄存器并去执行第二个子表的 DMA 传输，依次类推，直到最后一个子表的 DMA 传输执行完又跳回第一个子表继续遍历链表。第一个子表的指针可以被软件改写。

- 独立数据源和目标数据区的传输宽度（字节、半字、全字），模拟打包和拆包的过程。源和目标地址必须按数据传输宽度对齐。

- 基本模式下每个请求支持循环的缓冲器管理

- 有 2 个事件标志（DMA 传输完成和 DMA 传输出错），这 2 个事件标志逻辑或成为一个单独的中断请求。

- 存储器和存储器间的传输

- 外设和存储器、存储器和外设之间的传输

- 闪存、SRAM、外设的 SRAM、APB1、APB2 和 AHB 外设均可作为访问的源和目标。

- 可编程的数据传输数目：最大为 65535

13.2. DMA 功能描述

DMA 控制器和 Cortex™-M0 核心共享系统数据总线，执行直接存储器数据传输。当 CPU 和 DMA 同时访问相同的目标（RAM 或外设）时，DMA 请求会暂停 CPU 访问系统总线达若干个周期，总线仲裁器执行循环调度，以保证 CPU 至少可以得到一半的系统总线（存储器或外设）带宽。

13.2.1. 基本模式

基本模式下，当想要 DMA 服务于某个请求源时，在使能寄存器里配置使能该请求源并关闭其他请求源。并配置其他寄存器，包括源地址寄存器、目的地址寄存器、数目寄存器、config 寄存器等。一个请求源中可能包含一个或者多个请求，这多个请求是或逻辑，用户需要保证一个请求源中同时只有一个请求有效。

配置好上述寄存器后，开启 DMA EN 寄存器。DMA 开始传输数据，如果 config 寄存器的请求类型配置为软件请求模式，DMA 不需要外部请求就连续的按照数目寄存器所规定的数目传输完成；如果 config 寄存器的请求类型配置为硬件请求，需要等待使能的请求源外设的硬件请求每到来一次会搬一个数据（8 位、16 位或 32 位），DMA 收到一次硬件请求 REQ 会反馈一个 ACK 响应，外设收到 ACK 后拉低本次硬件请求 REQ，DMA 看到外设的 REQ 拉低就立刻也拉低 ACK，这样一次握手完成，DMA 也同时传输一个数据从源地址到目的地。等待

下一次 REQ 挂起后, DMA 和外设继续下一次握手以及下一个 DMA 传输。直到数目寄存器所规定的数目被传输完成。

传输完成后, DMA EN 会自动清零。DMA 完成标记位被置起, 如果使能了完成中断还会触发中断。如果传输过程中地址发生在不存在的地址会产生错误标记, 使能错误中断的话还会触发错误中断。

传输完成如果想继续传输, 按照需要配置好寄存器后, 再次使能 DMA EN。

13.2.2. 循环模式

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

在循环模式下, 最后一次传输结束时, 数目寄存器的内容会自动地被重新加载为其初始数值, 内部的当前源/目的地址寄存器也被重新加载为寄存器设定的初始基地址。

表61. DMA 硬件请求源

外设	请求源 1	请求源 2	请求源 3	请求源 4	请求源 5	请求源 6	请求源 7
ADC1	ADC1_CNV					ADC1_SEQ	
SPI		SPI1_RX	SPI1_TX				
USART				USART1_TX	USART1_RX		
I ² C						I ² C1_TX	I ² C1_RX
TIMER2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2 TIM2_CH4
TIMER1		TIM1_CH1	TIM1_CH2	TIM1_TX4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
Flash	Flash						

13.2.3. 链表模式

用户可以将链表存储于 SRAM 中, 可以定义多个链表。软件需要提前将即将链表结构体的基地址写到一个叫做 LIST_BASE_POINTER 寄存器里, 该链表才能被 DMA 硬件使用。链表的基本单元是子表, 一个链表包含多个的子表。

子表,

子表包含 4 个 word:

Offset	Name	Description
0x00	SRC_PTR	源地址的起始
0x04	DST_PTR	目的地址的起始
0x08	CFG	config 字
0x0C	Reserved	保留

CFG 字的格式如下表所示:

BIT	Description
[31:16]	NDT[15:0]: 数据传输数量 数据传输数量为 0 至 65535。这个位域载入内部寄存器后, DMA 传输开始后, 寄存器指示剩余的待传输字节数目。寄存器内容在每次 DMA 传输后递减。数据传输结束后, 寄存器的内容变为 0
[15:14]	Reserved
[13]	Command size 当该域为 1 时(指令或数据), DMA 硬件会等待 SPI 接口传输完本子表的 NDT 所规定的所有字节后才开始执行下一个子表, 目的是为了保证在下次 DMA 传输开始前 SPI 总线上的最后一字节传输完成。 0: 表示该子表所传输的内容不是指令或数据 1: 表示该子表所传输的内容是指令或数据
[12:11]	压缩数据格式. 支持 4 种压缩比例, DMA 传输时将源数据进行位宽扩充 00: 非压缩数据 01: 压缩比例 1:16 10: 压缩比例 1:8

	11: 压缩比例 1:4
[10]	链表尾标志 0: 当前不是链表的最后一个子表 1: 当前为链表的最后一个子表
[9]	REQ_MODE: 软件请求模式 0: 硬件请求模式; 1: 软件请求模式
[8:7]	DST_SIZE[1:0]: 目的数据宽度 00: 8 位 01: 16 位 10: 32 位 11: 保留
[6:5]	SRC_SIZE[1:0]: 源数据宽度 00: 8 位 01: 16 位 10: 32 位 11: 保留
[4]	DST_INC: 目的地址增量模式 0: 不执行目的地址增量操作 1: 执行目的地址增量操作
[3]	SRC_INC: 源地址增量模式 0: 不执行源地址增量操作 1: 执行源地址增量操作
[2]	TEIE: 允许该子表传输错误中断 0: 禁止 TE 中断 1: 允许 TE 中断
[1]	TCIE: 允许该子表传输完成中断 0: 禁止 TC 中断 1: 允许 TC 中断
[0]	子表传输使能 0: 停用该子表的传输 1: 启用该子表的传输

链表模式下, SRAM 中的链表结构如下图所示。该模式下 DMA EN 被使能后, DMA 根据 LIST_BASE_POINTER 的值去 SRAM 读取链表的第一个子表内容, 并将第一个子表规定的 DMA 传输数目执行完成后再去读取第二个子表的内容, 再将第二个子表规定的 DMA 传输数目执行完成, 以此类推, 顺序地将所有子表的 DMA 请求执行完成。直到执行完最后一个子表 (可以根据 CFG 配置字的链表尾标记来判断) 后, DMA 再次根据 LIST_BASE_POINTER 的值去 SRAM 读取链表的第一个子表内容, 开始新一轮循环。

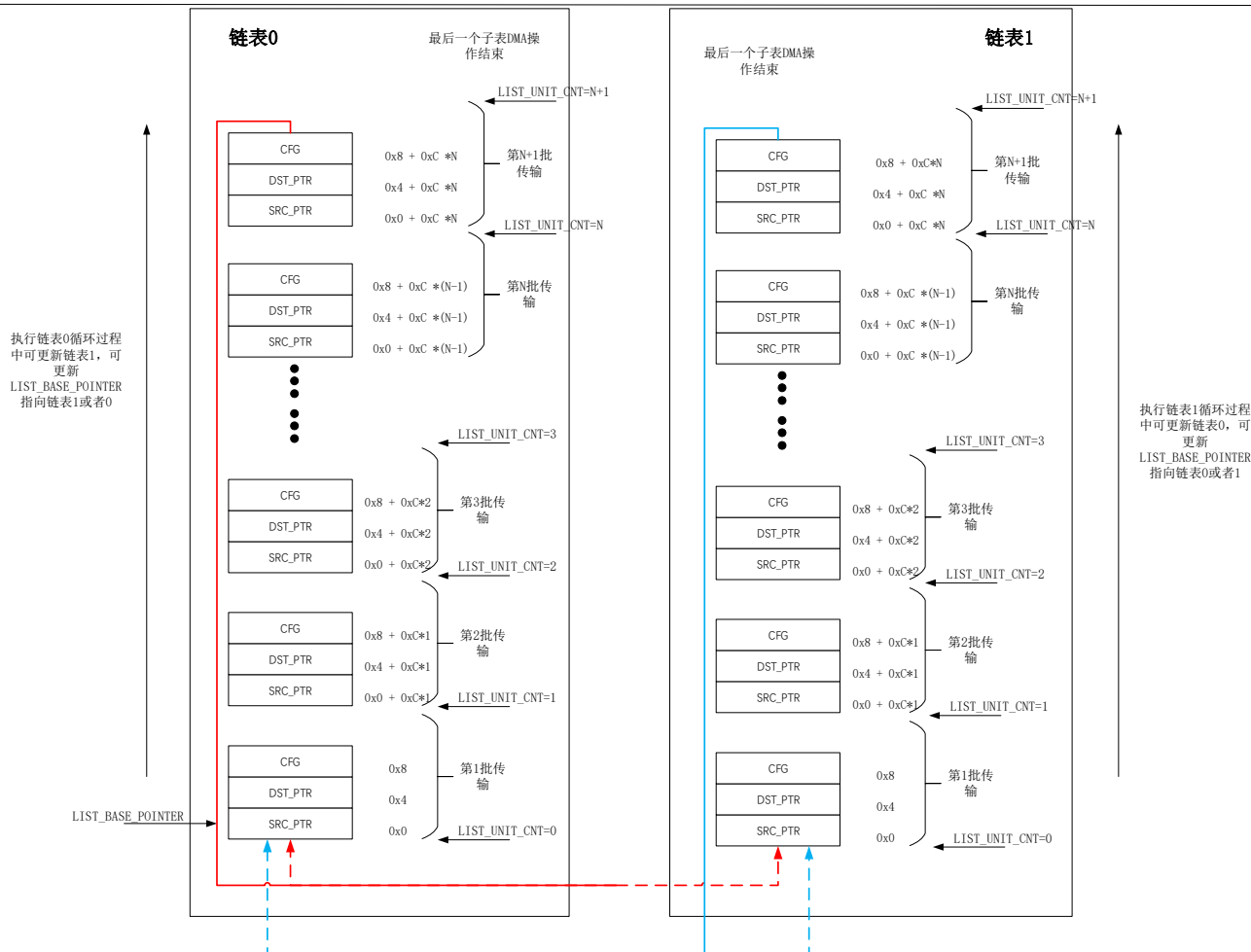
用户可以在 SRAM 中开辟多个链表空间, 下面以两个链表举例, 链表 0 和链表 1。两个链表可包含任意数目的子表(只要 SRAM 空间允许)。

在执行链表 0 过程中, CPU 可以更新链表 1 的内容, 也可以改变 LIST_BASE_POINTER 指向新的链表 1 或者继续指向链表 0。注意执行链表 0 时不能更新链表 0, 且要在已经开始执行链表 0 后再更新 LIST_BASE_POINTER, 否则后果不可预料, 可以通过判断 BUSY 为 1 决定是否已经开始执行链表 0。执行完链表 0 后会有遍历完成中断标志, DMA EN 使能自动关闭。如果要继续 DMA 传输, 再次打开 DMA EN 使能, DMA 再次根据 LIST_BASE_POINTER 的值去 SRAM 读取链表的第一个子表内容, 开始新一轮循环。同理上述规则对链表 1 也适用。

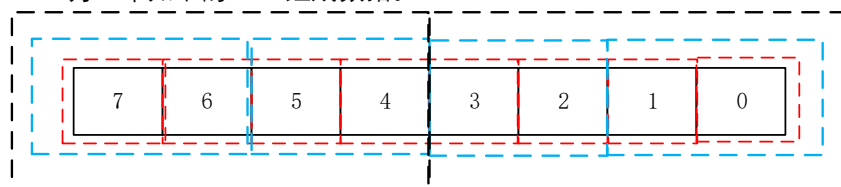
也可以在 DMA EN 使能关闭时更新链表 0 或者 1, 以及更新 LIST_BASE_POINTER。可以写 DMA EN 使能为 0 来终止正在执行的链表。

每个子表的 CFG 字有指示该子表的请求模式的位域, 不同的请求模式功能请参考基本模式下的请求模式描述。每个子表可以配置子表传输完成和子表传输错误的中断, 产生标记和中断的条件和基本模式的描述相同。但所有的子表共用一个完成标记寄存器和一个错误标记寄存器, 所以链表模式有一个子表计数寄存器 (LIST_UNIT_CNT) 指示当前的标记是属于第几个子表, 每完成一个子表, 子表计数寄存器会自动加 1。直到该链表全部完成后, 并再开启 DMA EN 使能后, 子表计数寄存器清 0。

当一个链表完成后, 会产生链表完成标记, 如果使能了链表完成中断, 也会产生中断。用户可以用该标记或中断来判断是否一个链表执行完成。



DMA 数据解压缩只适用于链表模式。每个子表的 CFG 字段 bit12:11 决定了当前子表所要传输的源数据的压缩比例, 针对不同的压缩比, 源数据的格式有所不同。以 8bit 的源数据为例如下图所示, 同理 16bit 为 2 个如下的 8bit 组成, 32bit 为 4 个如下的 8bit 组成数据。



 = 1个bit。对应两种16bit像素, 压缩比为1:16

 = 2个bit。对应四种16bit像素, 压缩比为1:8

 = 4个bit。对应八种16bit像素, 压缩比为1:4

红色框的 1bit 源数据可以对应 2 种 16 位的数据 (0 对应 PIXEL0 寄存器, 1 对应 PIXEL1 寄存器); 蓝色框的 2bit 源数据可以对应 4 种 16 位的数据; 蓝色框的 4bit 源数据可以对应 8 种 16 位的数据 (因为 bit4 是保留位, 只有低 3 位有效)。所以当压缩比为 1:16 时, 需要提前将 2 种数据写入两个 16 位寄存器 PIXEL0, PIXEL1, 根据红色框的内容来 2 选 1 某一个寄存器作为实际要发送给屏幕的数据; 当压缩比为 1:8 时, 需要提前将 4 种数据写入 4 个 16 位寄存器, 根据蓝色框的内容来 4 选 1 某一个寄存器作为实际要发送给屏幕的数据; 当压缩比为 1:4 时, 需要提前将 8 种数据写入 8 个 16 位寄存器, 根据黑色框的低 3 位内容来 8 选 1 某一个寄存器作为实际要发送给屏幕的数据。

13.3. DMA 寄存器

Register Name: ISR, Address: 0x40020000, Default: 0x0

表62. 中断标志寄存器

Bit Name	Position	Description	Default	Access
----------	----------	-------------	---------	--------

TCIF	[0]	传输完成标志 0: 当前传输没有完成或链表模式下当前子表传输没有完成 1: 当前传输完成或链表模式下当前子表传输完成	0x0	RW
TEIF	[1]	传输错误标志 0: 当前传输没有错误或链表模式下当前子表传输没有错误 1: 当前传输错误或链表模式下当前子表传输错误	0x0	RW
LCIF	[2]	链表模式下链表中所有子表传输完成标志 0: 链表没有完成 1: 链表完成	0x0	RW

Register Name: IFCR, Address: 0x40020004, Default: 0x0

表63. 中断标志清除寄存器

Bit Name	Position	Description	Default	Access
CTCIF	[0]	0: 不起作用 1: 清除当前传输或子表传输完成标志	0x0	RW
CTEIF	[1]	0: 不起作用 1: 清除当前传输或子表传输错误标志	0x0	RW
CLCIF	[2]	0: 不起作用 1: 清除链表完成标志	0x0	RW

Register Name: CCR, Address: 0x40020008, Default: 0x0

普通模式下的 CFG 寄存器可读可写，当开启 DMA(DMA_SETUP 的 EN=1)时不能写该寄存器。链表模式下，当 DMA 控制器开始依次执行每个子表的 DMA 传输之前，都要自动从 SRAM 装载子表的内容到 CFG 寄存器，CFG 寄存器实时反映当前要执行的子表的结构体内容。

表64. 配置寄存器

Bit Name	Position	Description	Default	Access
TCIE	[1]	0: 不使能当前传输完成中断 1: 使能当前传输完成中断	0x0	RW
TEIE	[2]	0: 不使能当前传输错误中断 1: 使能当前传输错误中断	0x0	RW
SRCINC	[3]	0: 不执行源地址增量操作 1: 执行源地址增量操作	0x0	RW
DSTINC	[4]	0: 不执行目的地址增量操作 1: 执行目的地址增量操作	0x0	RW
SRCSIZE	[5:6]	0: 8 位 1: 16 位 2: 32 位	0x0	RW
DSTSIZE	[7:8]	0: 8 位 1: 16 位 2: 32 位	0x0	RW
REQMODE	[9]	0: 硬件请求模式 1: 软件请求模式	0x0	RW
CIRC	[15]	0: 不使能循环模式 1: 使能循环模式	0x0	RW

Register Name: CNDTR, Address: 0x4002000c, Default: 0x0

表65. 传输数量寄存器,当开启 DMA(DMA_SETUP 的 EN=1)时不能写该寄存器

Bit Name	Position	Description	Default	Access
DNT	[0:15]	数据传输数量 (Number of data to transfer) 数据传输数量为 0 至 65535。这个寄存器只能在 DMA 不工作(DMA_SETUP 的 EN=0)时写入。DMA 开启后该寄存器变为只读，指示剩余的待传输字节数目。寄存器内容在每次 DMA 传输后递减。 数据传输结束后，寄存器的内容或者变为 0；或者当该 DMA 为循环模式时，寄存器的内容将被自动重新加载为之前配置时的数值。 当寄存器的内容为 0 时，无论 DMA 是否开启，都不会发生任何数据传输。 在链表模式下该寄存器是只读，初始值从子表的 CFG 字的 NDT 域装载得到，一旦 DMA 开启后指示剩余的待传输字节数目。	0x0	RW

Register Name: CDAR, Address: 0x40020010, Default: 0x0

表66. 目的地址寄存器,当开启 DMA(DMA_SETUP 的 EN=1)时不能写该寄存器

Bit Name	Position	Description	Default	Access
DA	[0:31]	目的地址 (Peripheral address) 目的数据寄存器的基地址，作为数据传输的目标。 当 DST_SIZE='01'(16 位)，不使用 DA[0]位。操作自动地与半字地址对齐。 当 DST_SIZE='10'(32 位)，不使用 DA[1:0]位。操作自动地与字地址对齐。	0x0	RW

Register Name: CSAR, Address: 0x40020014, Default: 0x0

表67. 源地址寄存器,当开启 DMA(DMA_SETUP 的 EN=1)时不能写该寄存器

Bit Name	Position	Description	Default	Access
SA	[0:31]	源地址 (Peripheral address) 源数据寄存器的基地址, 作为数据传输的源头。 当 SRC_SIZE='01'(16 位), 不使用 SA[0]位。操作自动地与半字地址对齐。 当 SRC_SIZE='10'(32 位), 不使用 SA[1:0]位。操作自动地与字地址对齐。	0x0	RW

Register Name: SETUP, Address: 0x40020018, Default: 0x0

表68. 设置寄存器

Bit Name	Position	Description	Default	Access
REQ_EN	[10:8]	硬件请求使能, 使能对应 DMA 硬件请求源一览表格的七个请求源中的一个 000: 不使能任何硬件请求源 001: 使能硬件请求源 1 010: 使能硬件请求源 2 011: 使能硬件请求源 3 100: 使能硬件请求源 4 101: 使能硬件请求源 5 110: 使能硬件请求源 6 111: 使能硬件请求源 7	0	RW
RESERVED	[7:4]	保留	0	RO
BUSY	[3]	当使能 EN 位后, DMA 开始传输, 该位立刻置 1; 当普通模式传输完成或者链表模式下所有使能的子表传输完成后, 该位自动清 0	0	RO
LCIE	[2]	链表传输完成中断使能 0: 不使能链表传输完成中断 1: 使能链表传输完成中断	0	RW
MODE	[1]	DMA 模式 0: 普通模式 1: 链表模式	0	RW
EN	[0]	使能 DMA 软件写 1 使能 DMA 后立刻开始 DMA 传输, 当普通模式传输完成或者链表模式下所有使能的子表传输完成后, 该位自动清 0。 或者发生传输错误该位也会自动清 0, 并结束传输。	0	RW

Register Name: LBAR, Address: 0x4002001C, Default: 0x0

表69. DMA 链表基地址寄存器,

Bit Name	Position	Description	Default	Access
LBA	[31:0]	链表基地址 指示 SRAM 中链表的起始地址 只在链表模式下有效	0x0	RW

Register Name: PIXEL0, Address: 0x40020020, Default: 0x0

Register Name: PIXEL1, Address: 0x40020024, Default: 0x0

Register Name: PIXEL2, Address: 0x40020028, Default: 0x0

Register Name: PIXEL3, Address: 0x4002002C, Default: 0x0

Register Name: PIXEL4, Address: 0x40020030, Default: 0x0

Register Name: PIXEL5, Address: 0x40020034, Default: 0x0

Register Name: PIXEL6, Address: 0x40020038, Default: 0x0

Register Name: PIXEL7, Address: 0x4002003C, Default: 0x0

表70. 像素数据寄存器,

Bit Name	Position	Description	Default	Access
RESERVED	[31:16]	保留	0x0	RW
PIXEL	[15:0]	Pixel[15:0]: 用于解码的像素数据	0x0	RW

Register Name: LISTUNIT_NUM, Address: 0x40020040, Default: 0x0

表71. 像素数据寄存器,

Bit Name	Position	Description	Default	Access
RESERVED	[31:11]	保留	0x0	RO
LIST_UNIT_NUM	[10:0]	在链表模式下, 该寄存器记录刚刚传输完成的子表编号, 每完成一个子表, 该寄存器就加 1. 若某个子表被跳过, 寄存器也会加 1.	0x0	RO

14. 24 位高精度模数转换器 (ADC)

LH32M0S3 集成高精度 24 位 Σ - Δ 模数转换器 (ADC) 和前置低噪声可编程放大器 (PGA)。支持的主要功能如下:

- 8 路外部测量通道
- 电源电压, 参考源, 测试信号等内部测量通道
- 数据速率可达 2Hz 至 8KHz SPS
- 支持差分 and 单端输入
- 支持 DMA
- 支持 1, 2, 4, 8, 16, 32, 64, 128 倍 PGA
- 支持多个参考电压
- Sequencer 序列器实现通道增益自动切换

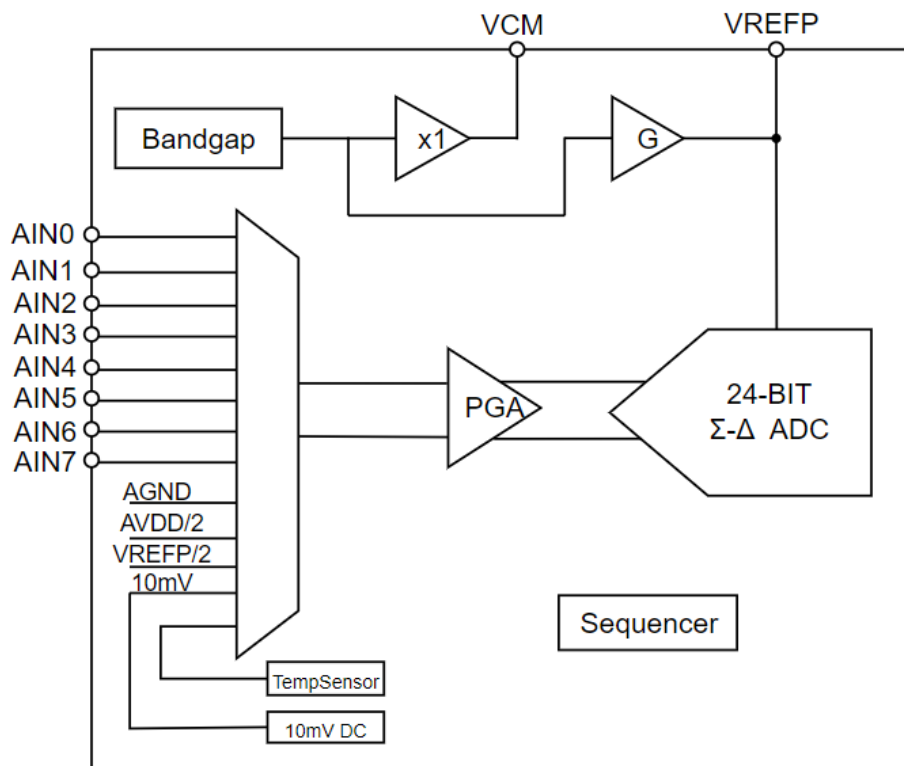


图12. 24 位 Σ - Δ ADC 框图

14.1. 参考电压

ADC 的集成内部参考电压有 2.45V, 1.8V, 2.3V 和 2.8V 四档可选择, 参考寄存器 ADCBUFCON[1]. 参考电压的输出引脚可以使能下拉, 以便在休眠时快速将参考引脚电容上的电荷泄放。

内部参考电压有一定的驱动能力可以用于驱动电桥等传感器, 内部参考的具体参数指标请参照 Datasheet。

另外 LH32M0S3 的 ADC 还支持使用外部参考电压, 将内部参考电压关闭后, 外部电压连接到 VREFP 引脚即可。

14.2. 共模电压 VCM

芯片内部基准经过单位增益缓冲后输出, 该电压可以用于提供 ADC 输入的共模电压, 或其他电流较小的偏置电压。

14.3. ADC 上电及复位

当 ADC 重新上电时, 通常需要等待几个时钟的时间, 该等待过程由硬件完成, 因此, ADC 上电的第一个有效数据会在几个 modulator 时钟后输出, 此后的数据按照过采样率的配置输出数据。

在切换通道、更改 PGA 和输出速率时第一个有效数据的输出同样会稍慢于后续数据的转换输出, 原因是上述操作会复位内部滤波器等。

14.4. Modulator 时钟和数据速率

通过配置 RCC 模块的 CFGR[28], 可将 ADC 的 Modulator 时钟设置为 PCLK2 除以 2, 4, 8, 16, 32, 64, 128, 512 或 1024。ADC 的典型 Modulator 时钟是 500KHz, 例如 PCLK2=32MHz, 则 CFGR[28]分频选择 64, 使用其他 Modulator 时钟可能导致 ADC 数据异常。

ADC 的输出数据速率是在 Modulator 时钟基础上除以过采样率(CONFIG1[0]: DR), 即

$$DataRate = \frac{500K}{2^{(CONFIG1.DR+2)}}$$

Σ-Δ ADC 中集成了 SINC3 滤波器, 该滤波器在阶跃输入时存在建立时间。因此当使能了 ADC 转换后, 第一个有效数据会在略大于 3 倍数据速率的间隔后输出。例如, 如果 DataRate 为 1KHz, 第一个数据点的会在 3ms 之后输出, 后续各点则按照 1ms 的速率输出。同样, 通道切换后的第一个数据点也满足这个规律。

14.5. 通道选择和数据读取

ADCCHCON 寄存器用于选择 ADC 的正向输入端和负向输入端, 选择和取值关系如下表。当正输入选择任意通道时, 负输入也可以选择任意通道。

对于单端测量, 可以将负输入端配置为 AGND。

噪声测量时, 通过将外部输入信号短路, 可以观察包括 ADC、PGA 以及输入电路整个信号链的噪声情况。

ADC 共有 6 个转换结果数据寄存器, 其中前 5 个数据寄存器对应 AIN0-AIN4 正输入通道。如: 当正输入端选择 AIN2 时, 无论负输入端选择哪个通道, 转换结果都将在 ADCDATA2 寄存器中更新。

表72. ADC 输入通道选择

正输入配置	正输入信号	转换结果	负输入配置	负输入信号
0x0	AIN0(Default)	ADCDATA[0]	0x0	AIN0
0x1	AIN1	ADCDATA[1]	0x1	AIN1(Default)
0x2	AIN2	ADCDATA[2]	0x2	AIN2
0x3	AIN3	ADCDATA[3]	0x3	AIN3
0x4	AIN4	ADCDATA[4]	0x4	AIN4
0x5	AIN5	ADCDATA[5]	0x5	AIN5
0x6	AIN6	ADCDATA[5]	0x6	AGND
0x7	AIN7	ADCDATA[5]	0x7	VREFP/2
0x8	AVDD/2	ADCDATA[5]		
0x9	VREFP/2	ADCDATA[5]		
0xa	Temp Sensor	ADCDATA[5]		
0xb	Reserved			
0xc	test signal, 10mV input	ADCDATA[5]		

14.6. 计算 ADC 数据

ADC 寄存器中包含 24bit ADC 数据、1bit 数据有效位和 1bit 数据饱和位。24bitADC 使用 2 进制补码表示 ADC 的差分电压信号。

表73.

Bit 范围	31:8	1	0
内容	ADC data	Data Ready	Overflow

计算 ADC 电压时, 可以先将数据转换成 int 类型再根据参考电压计算 ADC code 代表的电压值, 参考计算程序如下:

```
float ADC_DataToMiniVolt(uint32_t data, ADC_REF_t ref){
    float mv, ref_mv;
    int adc_data;
    adc_data = ((int32_t)data)>>8;
    switch(ref)
    {
        case 0 : ref_mv = 2450;break;
        case 1 : ref_mv = 1800;break;
```



```

        case 2 : ref_mv = 2300;break;
        case 3 : ref_mv = 2800;break;
        case 4: ref_mv = 3300;break;
        default: ref_mv = 2450;break;
    }
    mv = (adc_data)*2.0*ref_mv/((1<<24)-1);
    return mv;
}
    
```

14.7. Sequencer 序列器

Sequencer 序列器最多支持 6 个转换序列，每个序列允许独立配置 ADC 输入通道、PGA 增益和过采样率。尽管 Modulator 时钟等其他配置也是软件可配置的，但是为获得最好的模拟性能，这些配置应该写入推荐值。

下表列出了每个序列配置参数的寄存器，以及该序列转换结果会存储在哪个寄存器。

注意：为使 ADC 获得最好性能，应当保持 BOOST 为默认值 0,调制时钟应该被配置为 500KHz, 如当 PCLK=32M 时，SEQADCCLKCD.CLKCD = 5。

$$500K = \frac{PCLK2}{2^{(SEQADCCLKCD.CLKCD+1)}}$$

DataRate 由 OSR 决定，参考前节的输出速率计算公式。如果使用了多个 ADC 通道，需要注意通道切换导致的输出速率下降。

表74. 序列器配置

序列编号	结果寄存器	通道寄存器	过采样寄存器	增益寄存器	调制时钟	功耗模式
0	ADCDATA0	SEQCFGCHP [0:3] SEQCFGCHN [0:3]	SEQCFGOSR [0:3]	SEQCFGGAIN [0:3]	SEQADCCLKCD [0:3]	SEQCFGBOOST [0:3]
1	ADCDATA1	SEQCFGCHP [4:7] SEQCFGCHN [4:7]	SEQCFGOSR [4:7]	SEQCFGGAIN [4:7]	SEQADCCLKCD [4:7]	SEQCFGBOOST [4:7]
2	ADCDATA2	SEQCFGCHP [8:11] SEQCFGCHN [8:11]	SEQCFGOSR [8:11]	SEQCFGGAIN [8:11]	SEQADCCLKCD [8:11]	SEQCFGBOOST [8:11]
3	ADCDATA3	SEQCFGCHP [12:15] SEQCFGCHN [12:15]	SEQCFGOSR [12:15]	SEQCFGGAIN [12:15]	SEQADCCLKCD [12:15]	SEQCFGBOOST [12:15]
4	ADCDATA4	SEQCFGCHP [16:19] SEQCFGCHN [16:19]	SEQCFGOSR [16:19]	SEQCFGGAIN [16:19]	SEQADCCLKCD [16:19]	SEQCFGBOOST [16:19]
5	ADCDATA5	SEQCFGCHP [20:23] SEQCFGCHN [20:23]	SEQCFGOSR [20:23]	SEQCFGGAIN [20:23]	SEQADCCLKCD [20:23]	SEQCFGBOOST [20:23]

寄存器 ADCDIGCON.CHEN 将决定使能几个序列，当 ADCDIGCON.SEQEN = 1 且 CONFIG1. CONSCNV=1 连续转换开启时，序列中的 ADC 通道开始逐个转换。

14.8. 温度传感器

片上集成了温度传感器，当 ADCCHCON.PSEL=0x0A, ADCCHCON.NSEL=5(AIN5) 时该温度传感器的电压输出可以被 ADC 采集，读取寄存器 ADCDATA5 可以获得温度传感器的 ADC 转换结果，温度的计算公式如下：

$$T(^{\circ}\text{C}) = \frac{V - 1136}{3.942} + 25$$

其中，V 是 ADC code 计算出的电压值，单位为 mV。

14.9. ADC 寄存器

Register Name: CONFIG1, Address: 0x40012800, Default: 0xc

表75. filter related configuration part 1

Bit Name	Position	Description	Default	Access
DR	[0:3]	channel oversampling ratio 0x1: Fmodclk/8,64K SPS @Fmodclk=500K 0x2: Fmodclk/16,32K SPS @Fmodclk=500K 0x3: Fmodclk/32,16K SPS @Fmodclk=500K 0x4: Fmodclk/64,8K SPS @Fmodclk=500K 0x5: Fmodclk/128,4K SPS @Fmodclk=500K	0xc	RW

Bit Name	Position	Description	Default	Access
		0x6: Fmodclk/256,2K SPS @Fmodclk=500K 0x7: Fmodclk/512,1K SPS @Fmodclk=500K 0x8: Fmodclk/1024,500 SPS @Fmodclk=500K 0x9: Fmodclk/2048,250 SPS @Fmodclk=500K 0xA: Fmodclk/4096,125 SPS @Fmodclk=500K 0xB: Fmodclk/8192,62.5 SPS @Fmodclk=500K 0xC: Fmodclk/16384,31.25 SPS @Fmodclk=500K 0xD: Fmodclk/32768,15.625 SPS @Fmodclk=500K 其他: Fmodclk/65536,7.8125 SPS @Fmodclk=500K		
SINGLE_SHOT	[4]	single-shot conversion 0x0: no conversion 0x1: Single-shot conversion mode	0x0	RW
CONSCNV	[8]	连续转换 0: no conversion 1: Continuous conversion mode	0x0	RW
SINCTYPE	[16:17]	SINC 滤波类型,推荐使用默认 SINC3 滤波器 0: SINC3 1: SINC2	0x0	RW

Register Name: ADCCHCON, Address: 0x40012804, Default: 0x0

表76. ADC input channel selection

Bit Name	Position	Description	Default	Access
PSEL	[0:3]	Positive Input Channel Select 0x0: AIN0(Default) 0x1: AIN1 0x2: AIN2 0x3: AIN3 0x4: AIN4 0x5: AIN5 0x6: AIN6 0x7: AIN7 0x8: AVDD/2 0x9: VREFP/2 0xa: Temp Sensor 0xb: VCM,Internal short 0xc: test signal, 10mV input	0x0	RW
NSEL	[4:7]	Negative Input Channel Select 0: AIN0 1: AIN1(Default) 2: AIN2 3: AIN3 4: AIN4 5: AIN5 6: AGND 7: VREFP/2	0x0	RW

Register Name: ADCBUFCON, Address: 0x40012808, Default: 0x00000100

表77. ADC 参考源控制

Bit Name	Position	Description	Default	Access
EN	[0]	VREFP Buffer 使能控制 0x0: Disable 0x1: Enable	0x0	RW
RANGE	[1:2]	Choose output voltage for ADC refbuf 0x0: output voltage is 2.45V 0x1: output voltage is 1.8V 0x2: output voltage is 2.3V 0x3: output voltage is 2.8V	0x0	RW
USER_TRIM	[4:8]	Trim code for ADC refbuf absolute voltage trim 0x0: output voltage is lowest 0x10: middle(Default) 0x1F: output voltage is 2.45V	0x10	RW
VS_TRIM_SEL	[9]	Trim 寄存器选择 0x0: 选择出厂 trim 寄存器 0x1: 选择 user_Trim 寄存器	0x0	RW
PULLDOWN	[10]	ADC VREF 下拉使能控制 0x0: 不使能 0x1: 使能	0x0	RW

Register Name: ADC_CTRL, Address: 0x4001280c, Default: 0x0

表78.

Bit Name	Position	Description	Default	Access
MOD_PDB	[0]	0x0: power down ADC 0x1: power up ADC	0x0	RW
MOD_PDB_AMP	[1]	0x0: power down ADC amps 0x1: enable ADC amps	0x0	RW

Register Name: PGA_CTRL, Address: 0x40012810, Default: 0xd

表79.

Bit Name	Position	Description	Default	Access
PGA_PD	[0]	0x0: pga enable 0x1: pga power down	0x1	RW
PGA_BYP	[1]	0x0: bypass disable 0x1: bypass pga	0x0	RW
PGA_RESET	[2]	0x0: no reset 0x1: reset PGA	0x1	RW
PGA_LP_EN	[3]	0x0: PGA in HP mode 0x1: PGA in LP mode	0x1	RW
BOOST	[8]		0x0	RW

Register Name: PGACTRL1, Address: 0x40012814, Default: 0x3

表80.

Bit Name	Position	Description	Default	Access
PGA_CHOP	[0:1]	0x0: 1/64 modulator clock 0x1: 1/32 modulator clock 0x2: 1/16 modulator clock 0x3: 1/8 modulator clock	0x3	RW

Register Name: PGA_GAIN, Address: 0x40012818, Default: 0x0

表81.

Bit Name	Position	Description	Default	Access
GAIN	[0:3]	0x7: Gain=128 (端口输出 0) 0x6: Gain=64 (端口输出 1) 0x5: Gain=32 (端口输出 3) 0x4: Gain=16 (端口输出 7) 0x3: Gain=8 (端口输出 F) 0x2: Gain=4 (端口输出 1F) 0x1: Gain=2 (端口输出 3F) 0x0: Gain=1 (端口输出 7F)	0x0	RW

Register Name: ADCGAINDIFF, Address: 0x4001281c, Default: 0x7fffff

表82. 差分通道增益误差校准寄存器

Bit Name	Position	Description	Default	Access
GAIN	[0:23]	gain error correction for differential mode	0x7fffff	RW

Register Name: ADCOFFSETDIFF, Address: 0x40012820, Default: 0x0

表83. 差分通道零点误差校准寄存器

Bit Name	Position	Description	Default	Access
OFFSET	[0:23]	Offset error correction for differential mode	0x0	RW

Register Name: ADCGAINSE, Address: 0x40012824, Default: 0x7fffff

表84. 单端通道增益误差校准寄存器

Bit Name	Position	Description	Default	Access
GAIN	[0:23]	gain error correction for single ended mode	0x7fffff	RW

Register Name: ADCOFFSETSE, Address: 0x40012828, Default: 0x0

表85. 单端通道零点误差校准寄存器

Bit Name	Position	Description	Default	Access
OFFSET	[0:23]	Offset error correction for differential mode	0x0	RW

Register Name: ADCDATA[5:0], Address: 0x40012830, Default: 0x0

表86.

Bit Name	Position	Description	Default	Access
OVF	[0]	overflow flag 0x0: ADC data ready	0x0	R(ReadOnly)

		0x1: ADC datanot ready		
RDY	[1]		0x0	R(ReadOnly)
DATA	[8:31]	ADC convert resault for each channel. Channel >5 shares last adcddata channel data register.	0x0	R(ReadOnly)

Register Name: ADCDIGCON, Address: 0x40012864, Default: 0x0

表87.

Bit Name	Position	Description	Default	Access
CHEN	[0:7]	Enable sequencer channel, these bits can be combined 0x1: enable channel 0 0x2: enable channel 1 0x4: enable channel 2 0x8: enable channel 3 0x10: enable channel 4 0x20: enable channel 5	0x0	RW
seqstall	[8]	Pause sequencer 0x0: disable, continuous running sequence 0x1: enable, sequence paused	0x0	RW
seqen	[9]	Enable sequencer 0x0: running 0x1: stall	0x0	RW
seqdmaen	[10]	Enable sequencer DMA 0x0: disable 0x1: enable	0x0	RW
seqinten	[11]	Enable sequencer interrupt 0x0: disable 0x1: enable	0x0	RW
adcdmaen	[12]	Enable ADC DMA 0x0: disable 0x1: enable	0x0	RW
adcinten	[13]	Enable ADC interrupt 0x0: disable 0x1: enable	0x0	RW
adcintsta	[14]	Status of ADC interrupt 0x0: none 0x1: interrupt	0x0	W1C(Write1/auto-clear 0)
seqintsta	[15]	Status of sequencer interrupt 0x0: none 0x1: interrupt	0x0	W1C(Write1/auto-clear 0)
RESTART	[16]		0x0	RW

Register Name: SEQCFGCHN, Address: 0x400128e0, Default: 0x0

表88.

Bit Name	Position	Description	Default	Access
CHN0	[0:3]	Sequence 0 negative input	0x0	RW
CHN1	[4:7]	Sequence 1 negative input	0x0	RW
CHN2	[8:11]	Sequence 2 negative input	0x0	RW
CHN3	[12:15]	Sequence 3 negative input	0x0	RW
CHN4	[16:19]	Sequence 4 negative input	0x0	RW
CHN5	[20:23]	Sequence 5 negative input	0x0	RW

Register Name: SEQCFGCHP, Address: 0x400128e4, Default: 0x0

表89.

Bit Name	Position	Description	Default	Access
CHP0	[0:3]	Sequence 0 postive input	0x0	RW
CHP1	[4:7]	Sequence 1 postive input	0x0	RW
CHP2	[8:11]	Sequence 2 postive input	0x0	RW
CHP3	[12:15]	Sequence 3 postive input	0x0	RW
CHP4	[16:19]	Sequence 4 postive input	0x0	RW
CHP5	[20:23]	Sequence 5 postive input	0x0	RW

Register Name: SEQCFGOSR, Address: 0x400128e8, Default: 0x0

表90.

Bit Name	Position	Description	Default	Access
OSR0	[0:3]	Sequence 0 oversampling rate	0x0	RW
OSR1	[4:7]	Sequence 1 oversampling rate	0x0	RW
OSR2	[8:11]	Sequence 2 oversampling rate	0x0	RW

OSR3	[12:15]	Sequence 3 oversampling rate	0x0	RW
OSR4	[16:19]	Sequence 4 oversampling rate	0x0	RW
OSR5	[20:23]	Sequence 5 oversampling rate	0x0	RW

Register Name: SEQADCCLKCD, Address: 0x400128ec, Default: 0x0

表91.

Bit Name	Position	Description	Default	Access
CLKCD0	[0:3]	Sequence 0 modulator clock divider	0x0	RW
CLKCD1	[4:7]	Sequence 1 modulator clock divider	0x0	RW
CLKCD2	[8:11]	Sequence 2 modulator clock divider	0x0	RW
CLKCD3	[12:15]	Sequence 3 modulator clock divider	0x0	RW
CLKCD4	[16:19]	Sequence 4 modulator clock divider	0x0	RW
CLKCD5	[20:23]	Sequence 5 modulator clock divider	0x0	RW

Register Name: SEQCFGBOOST, Address: 0x400128f0, Default: 0x0

表92.

Bit Name	Position	Description	Default	Access
BOOST0	[0:3]	Sequence 0 boost mode	0x0	RW
BOOST1	[4:7]	Sequence 1 boost mode	0x0	RW
BOOST2	[8:11]	Sequence 2 boost mode	0x0	RW
BOOST3	[12:15]	Sequence 3 boost mode	0x0	RW
BOOST4	[16:19]	Sequence 4 boost mode	0x0	RW
BOOST5	[20:23]	Sequence 5 boost mode	0x0	RW

Register Name: SEQCFGGAIN, Address: 0x400128f4, Default: 0x0

表93.

Bit Name	Position	Description	Default	Access
SEQ0_GAIN	[0:3]	Sequence 0 PGA GAIN	0x0	RW
SEQ1_GAIN	[4:7]	Sequence 1 PGA GAIN	0x0	RW
SEQ2_GAIN	[8:11]	Sequence 2 PGA GAIN	0x0	RW
SEQ3_GAIN	[12:15]	Sequence 3 PGA GAIN	0x0	RW
SEQ4_GAIN	[16:19]	Sequence 4 PGA GAIN	0x0	RW
SEQ5_GAIN	[20:23]	Sequence 5 PGA GAIN	0x0	RW

Register Name: TEMPGAIN, Address: 0x40012868, Default: 0x7ffff

表94. 温度传感器 Gain 系数

Bit Name	Position	Description	Default	Access
Gain	[0]	温度传感器 Gain 系数	0x7ffff	RW

Register Name: TEMPOFFSET, Address: 0x4001286c, Default: 0x0

表95. 温度传感器 Offset 系数

Bit Name	Position	Description	Default	Access
Offset	[0]	温度传感器 Offset 系数	0x0	RW

15. 通用定时器 (TIM2)

15.1. TIM2 简介

通用定时器是一个通过可编程预分频器驱动的 16 位自动装载计数器构成。它适用于多种场合，包括测量输入信号的脉冲长度（输入捕获）或者产生输出波形（输出比较和 PWM）。使用定时器预分频器和 RCC 时钟控制器预分频器，脉冲长度和波形周期可以在几个微秒到几个毫秒间调整。

每个定时器都是完全独立的，没有互相共享任何资源。它们可以一起同步操作。

15.2. TIM2 主要功能

通用 TIM2 定时器功能包括：

- 16 位向上、向下、向上/向下自动装载计数器
- 16 位可编程（可以实时修改）预分频器，计数器时钟频率的分频系数为 1 ~ 65536 之间的任意数值
- 4 个独立通道：
 - 输入捕获
 - 输出比较

- PWM 生成 (边缘或中间对齐模式)
- 单脉冲模式输出
- 使用外部信号控制定时器和定时器互连的同步电路
- 如下事件发生时产生中断/DMA:
 - 更新: 计数器向上溢出/向下溢出, 计数器初始化 (通过软件或者内部/外部触发)
 - 触发事件 (计数器启动、停止、初始化或者由内部/外部触发计数)
 - 输入捕获
 - 输出比较
- 支持针对定位的增量 (正交) 编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理

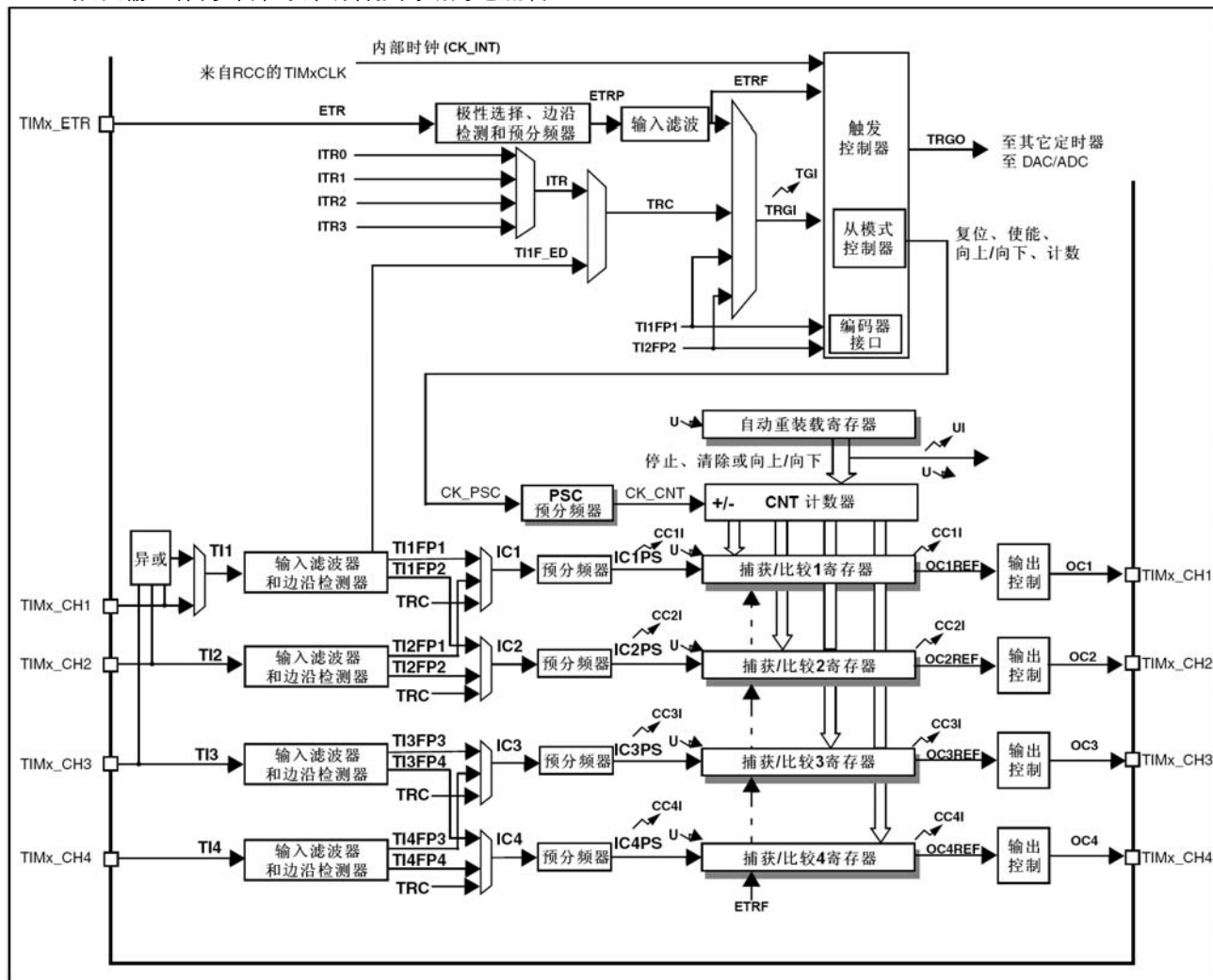


图13. 通用定时器框图

15.3. TIM2 功能描述

15.3.1. 时基单元

可编程通用定时器的主要部分是一个 16 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数、向下计数或者向上向下双向计数。此计数器时钟由预分频器分频得到。

计数器、自动装载寄存器和预分频器寄存器可以由软件读写，在计数器运行时仍可以读写。时基单元包含：

- 计数器寄存器 (TIM2_CNT)
- 预分频器寄存器 (TIM2_PSC)
- 自动装载寄存器 (TIM2_ARR)

自动装载寄存器是预先装载的，写或读自动重载寄存器将访问预装载寄存器。根据在 TIM2_CR1 寄存器中的自动装载预装载使能位 (ARPE) 的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件 (向下计数时的下溢条件) 并当 TIM2_CR1 寄存器中的 UDIS 位等于 '0' 时，产

生更新事件。更新事件也可以由软件产生。随后会详细描述每一种配置下更新事件的产生。

计数器由预分频器的时钟输出 CK_CNT 驱动, 仅当设置了计数器 TIM2_CR1 寄存器中的计数器使能位(CEN)时, CK_CNT 才有效。(有关计数器使能的细节, 请参见控制器的从模式描述)。

注: 真正的计数器使能信号 CNT_EN 是在 CEN 的一个时钟周期后被设置。

15.3.1.1. 预分频器描述

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个 (在 TIM2_PSC 寄存器中的) 16 位寄存器控制的 16 位计数器。这个控制寄存器带有缓冲器, 它能够在工作时被改变。新的预分频器参数在下次更新事件到来时被采用。

下图分别给出了在预分频器运行时, 更改计数器参数的例子。

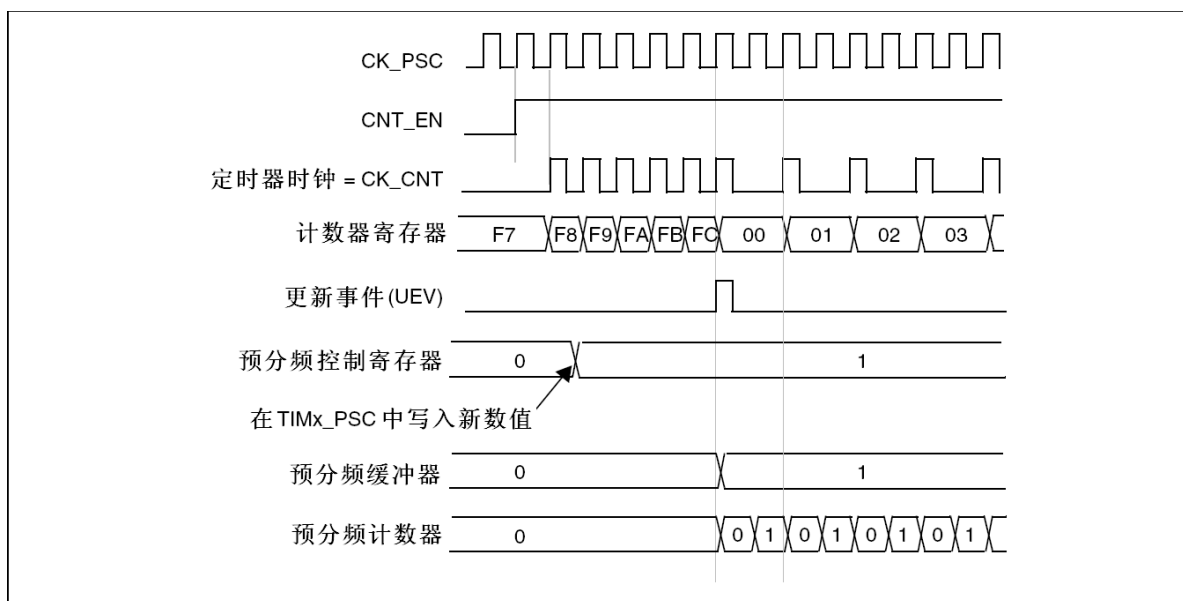


图14. 当预分频器的参数从 1 变到 2 时, 计数器的时序图

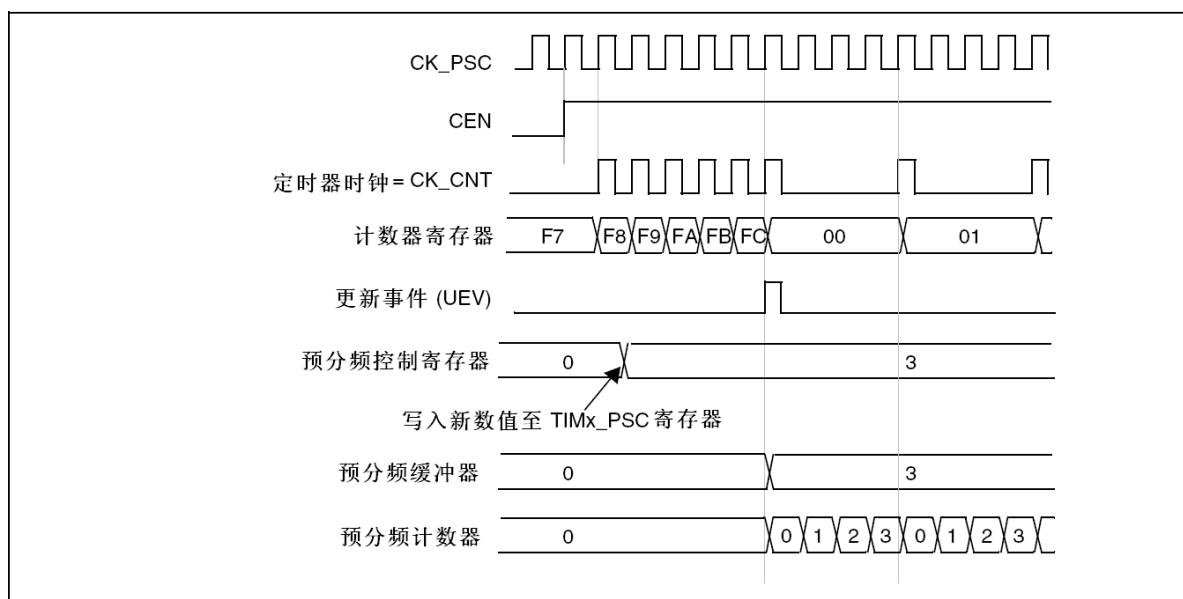


图15. 当预分频器的参数从 1 变到 4 时, 计数器的时序图

15.3.2. 计数器模式

15.3.2.1. 向上计数模式

在向上计数模式中, 计数器从 0 计数到自动加载值 (TIM2_ARR 计数器的内容), 然后重新从 0 开始计数并

且产生一个计数器溢出事件。

每次计数器溢出时可以产生更新事件，在 TIM2_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位也同样可以产生一个更新事件。

设置 TIM2_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清'0'之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清'0'，同时预分频器的计数也被清 0（但预分频系数不变）。此外，如果设置了 TIM2_CR1 寄存器中的 URS 位（选择更新请求），设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志（即不产生中断或 DMA 请求）；这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时（依据 URS 位）设置更新标志位

（TIM2_SR 寄存器中的 UIF 位）。

- 预分频器的缓冲区被置入预装载寄存器的值（TIM2_PSC 寄存器的内容）。
- 自动装载影子寄存器被重新置入预装载寄存器的值（TIM2_ARR）。

下图给出一些例子，当 TIM2_ARR=0x36 时计数器在不同时钟频率下的动作。

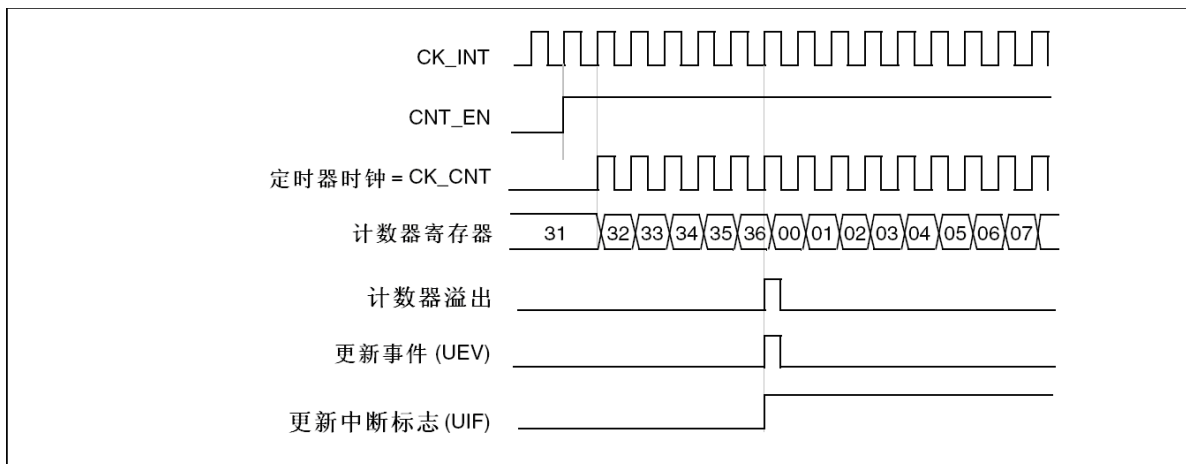


图16. 计数器时序图，内部时钟分频因子为 1

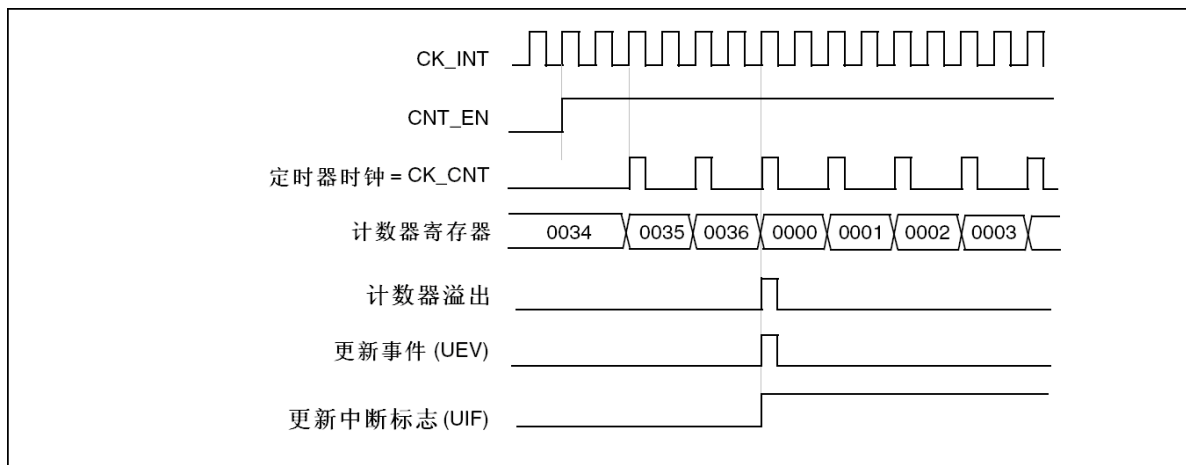


图17. 计数器时序图，内部时钟分频因子为 2

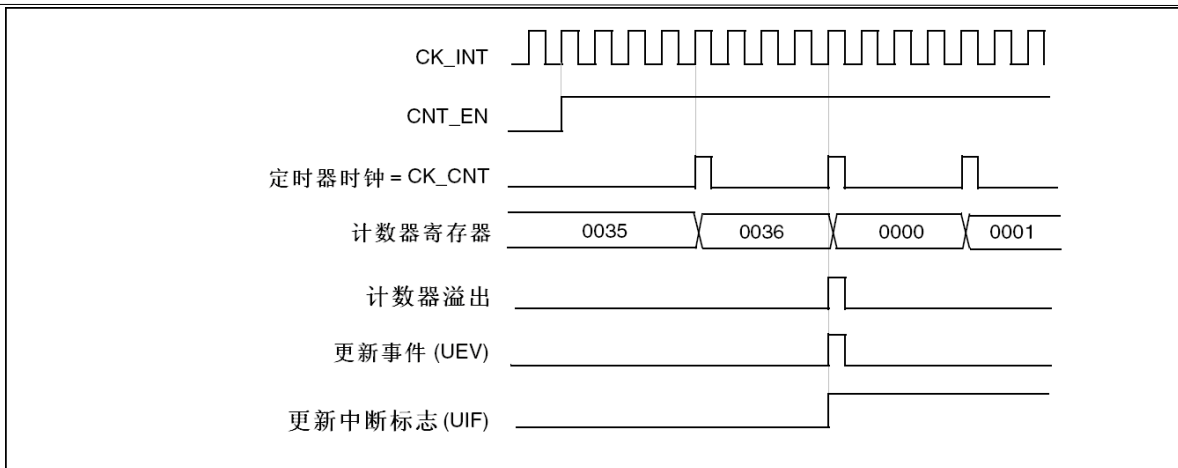


图18. 计数器时序图，内部时钟分频因子为 4

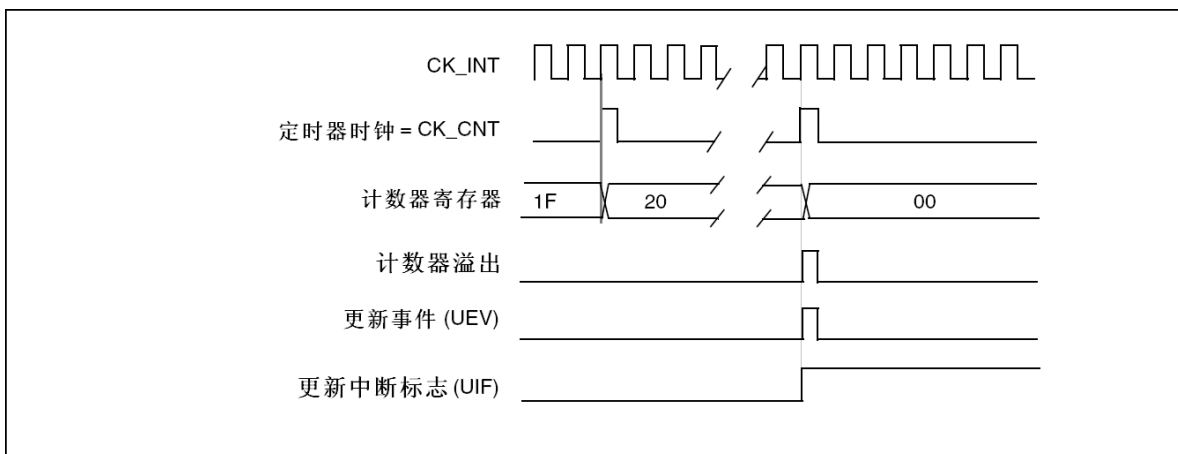


图19. 计数器时序图，内部时钟分频因子为 N

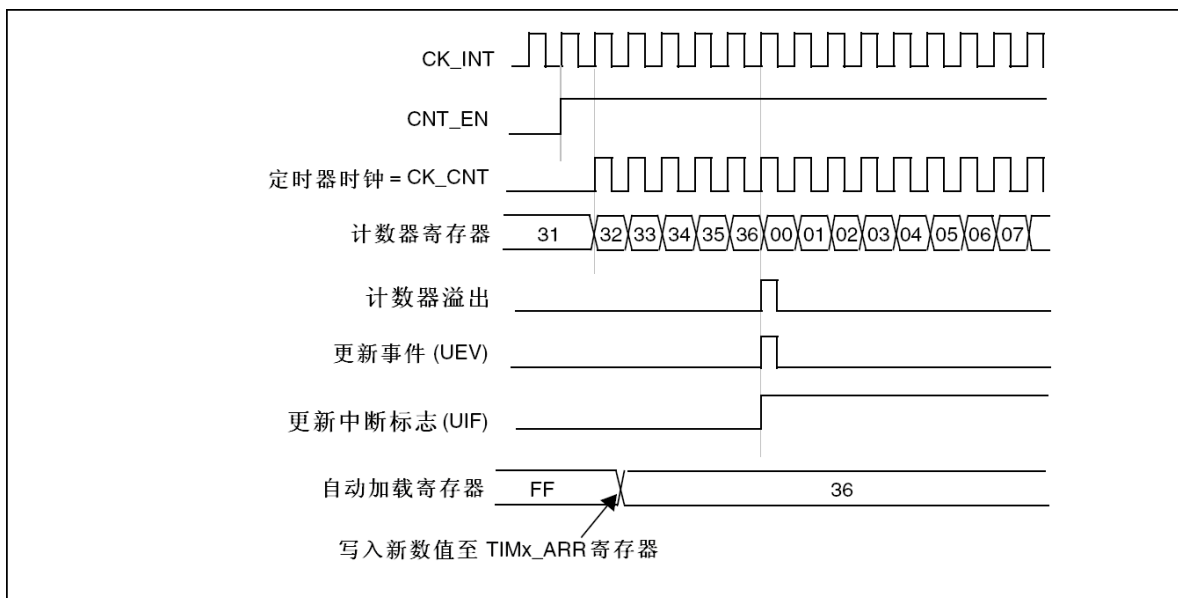


图20. 计数器时序图，当 ARPE=0 时的更新事件（TIM2_ARR 没有预装入）

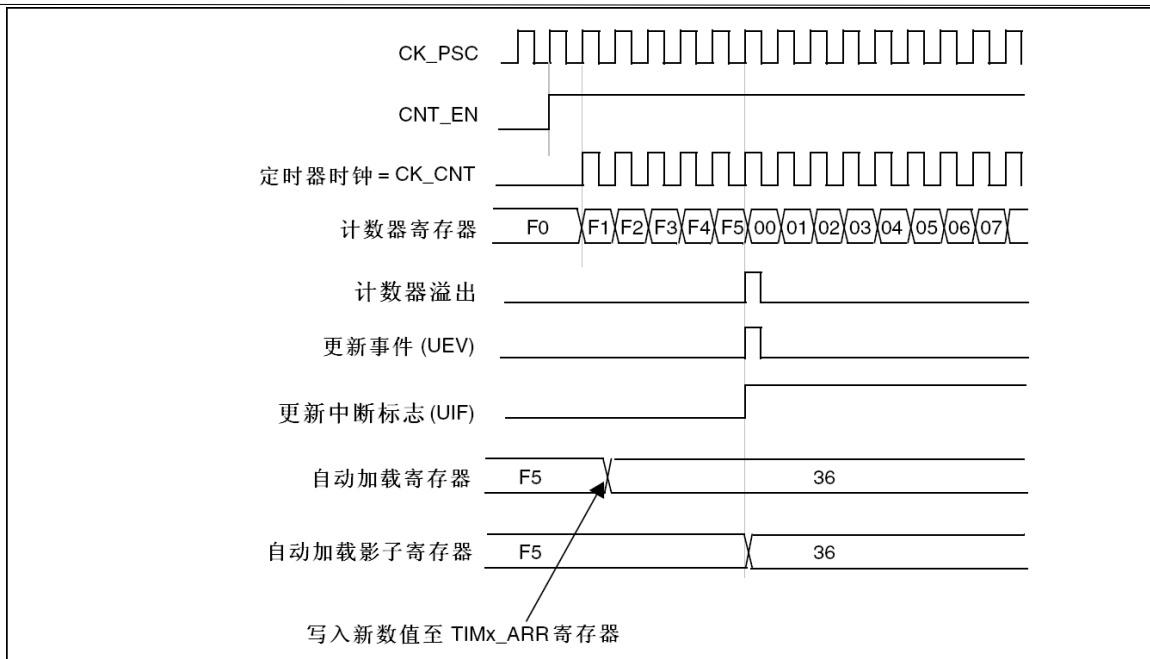


图21. 计数器时序图，当 ARPE=1 时的更新事件（预装入了 TIM2_ARR）

15.3.2.2. 向下计数模式

在向下模式中，计数器从自动装入的值（TIM2_ARR 计数器的值）开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数器向下溢出事件。

每次计数器溢出时可以产生更新事件，在 TIM2_EGR 寄存器中（通过软件方式或者使用从模式控制器）设置 UG 位，也同样可以产生一个更新事件。

设置 TIM2_CR1 寄存器的 UDIS 位可以禁止 UEV 事件。这样可以避免向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为'0'之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，同时预分频器的计数器重新从 0 开始（但预分频系数不变）。

此外，如果设置了 TIM2_CR1 寄存器中的 URS 位（选择更新请求），设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志（因此不产生中断和 DMA 请求），这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且（根据 URS 位的设置）更新标志位（TIM2_SR 寄存器中的 UIF 位）也被设置。

- 预分频器的缓存器被置入预装载寄存器的值（TIM2_PSC 寄存器的值）。
- 当前的自动加载寄存器被更新为预装载值（TIM2_ARR 寄存器中的内容）。

注：自动装载在计数器重载入之前被更新，因此下一个周期将是预期的值。

以下是一些当 TIM2_ARR=0x36 时，计数器在不同时钟频率下的操作例子。

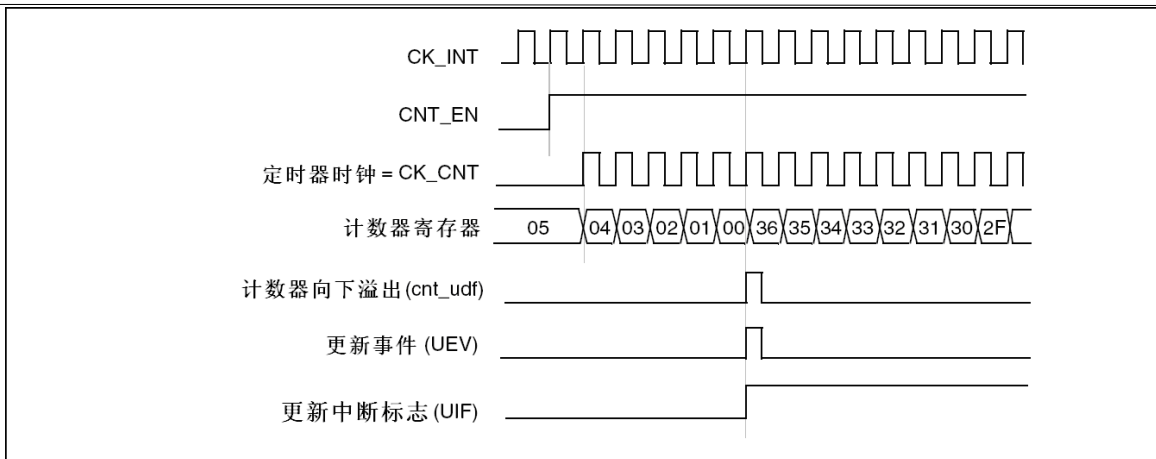


图22. 计数器时序图，内部时钟分频因子为 1

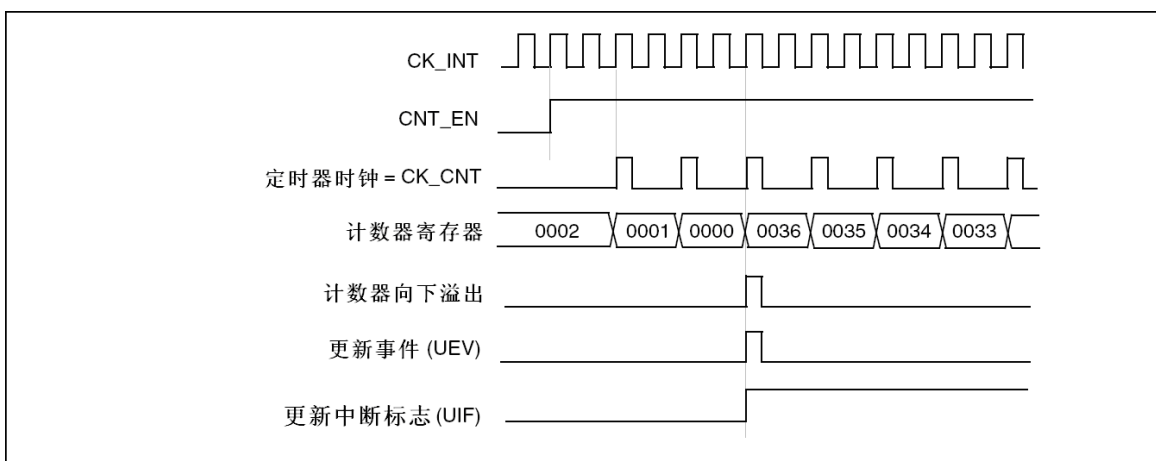


图23. 计数器时序图，内部时钟分频因子为 2

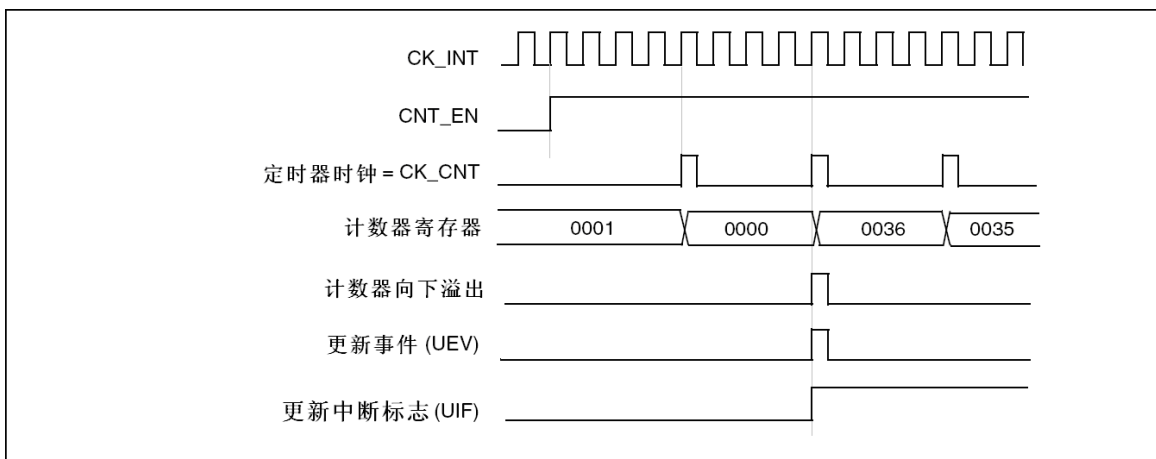


图24. 计数器时序图，内部时钟分频因子为 4

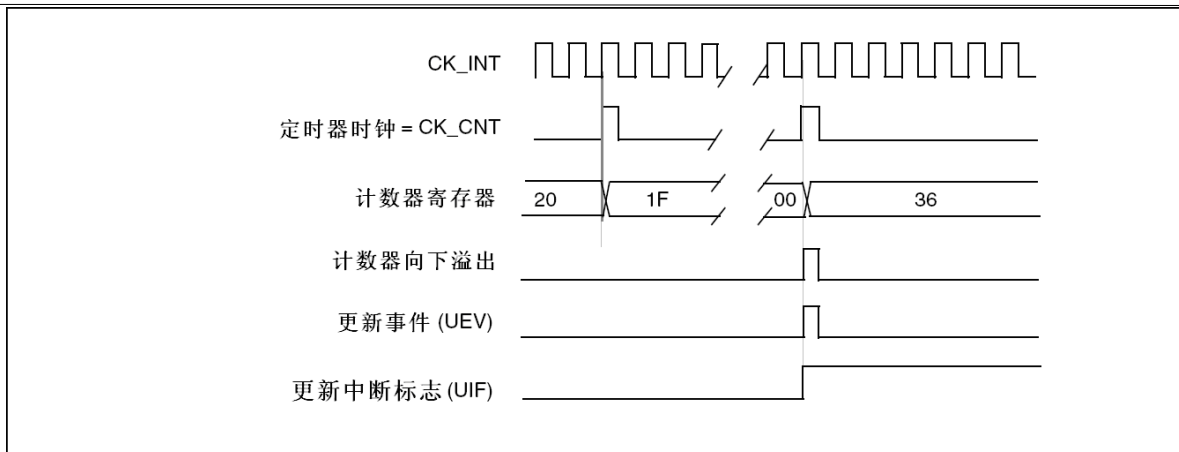


图25. 计数器时序图，内部时钟分频因子为 N

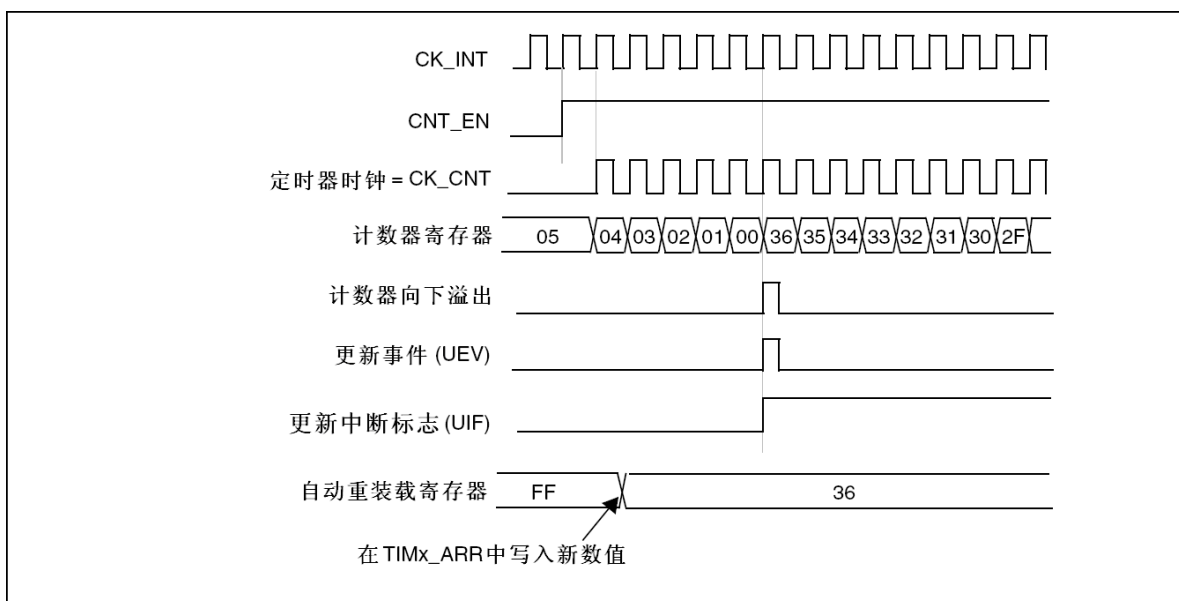


图26. 计数器时序图，当没有使用重复计数器时的更新事件

15.3.2.3. 中央对齐模式（向上/向下计数）

在中央对齐模式，计数器从 0 开始计数到自动加载的值（TIM2_ARR 寄存器）-1，产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。

在这个模式，不能写入 TIM2_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

可以在每次计数上溢和每次计数下溢时产生更新事件；也可以通过（软件或者使用从模式控制器）设置 TIM2_EGR 寄存器中的 UG 位产生更新事件。然后，计数器重新从 0 开始计数，预分频器也重新从 0 开始计数。

设置 TIM2_CR1 寄存器中的 UDIS 位可以禁止 UEV 事件。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为'0'之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

此外，如果设置了 TIM2_CR1 寄存器中的 URS 位（选择更新请求），设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志（因此不产生中断和 DMA 请求），这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且（根据 URS 位的设置）更新标志位（TIM2_SR 寄存器中的 UIF 位）也被设置。

- 预分频器的缓存器被加载为预装载（TIM2_PSC 寄存器）的值。
- 当前的自动加载寄存器被更新为预装载值（TIM2_ARR 寄存器中的内容）。

注：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期将是预

期的值（计数器被装载为新的值）。

以下是一些计数器在不同时钟频率下的操作的例子：

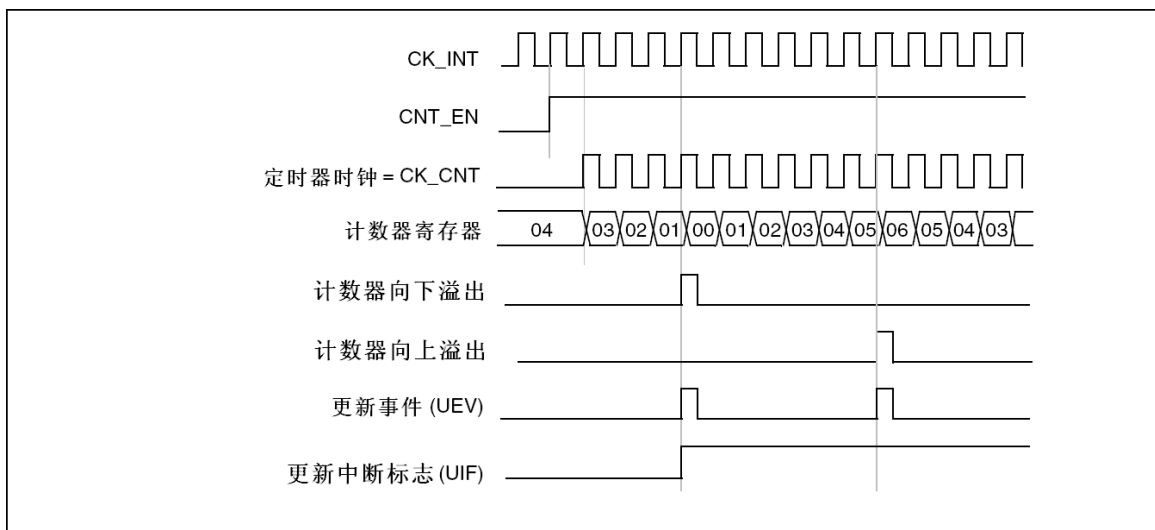


图27. 计数器时序图，内部时钟分频因子为 1，TIM2_ARR=0x6

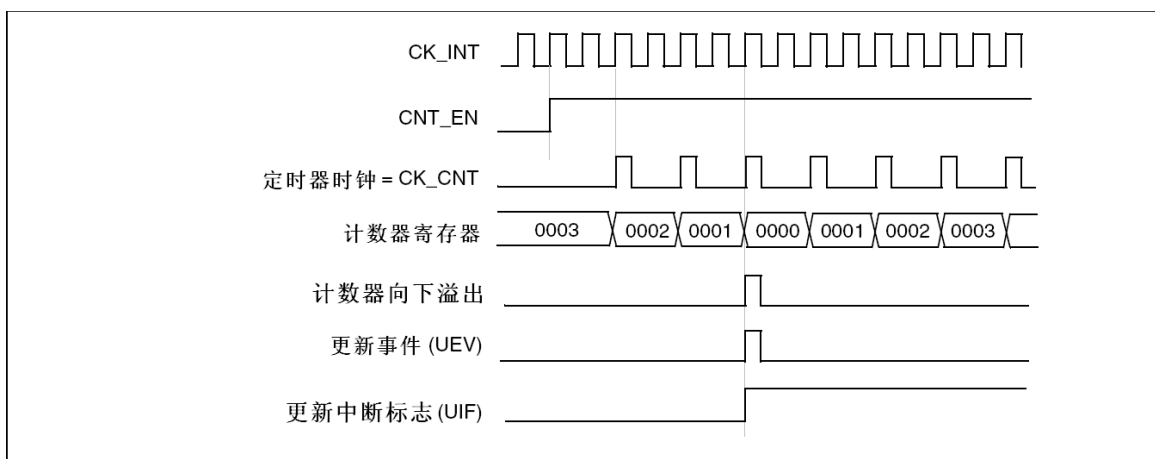


图28. 计数器时序图，内部时钟分频因子为 2

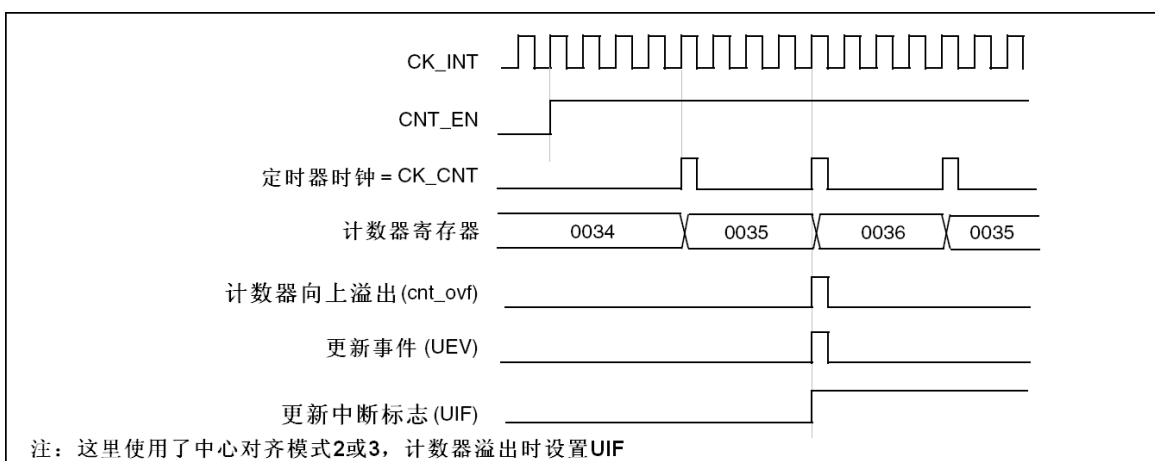


图29. 计数器时序图，内部时钟分频因子为 4，TIM2_ARR=0x36

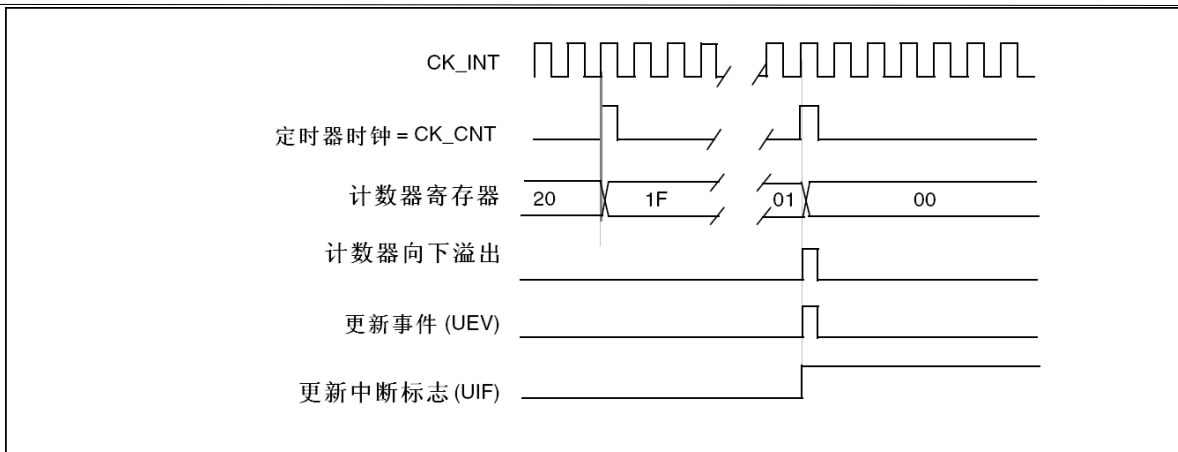


图30. 计数器时序图，内部时钟分频因子为 N

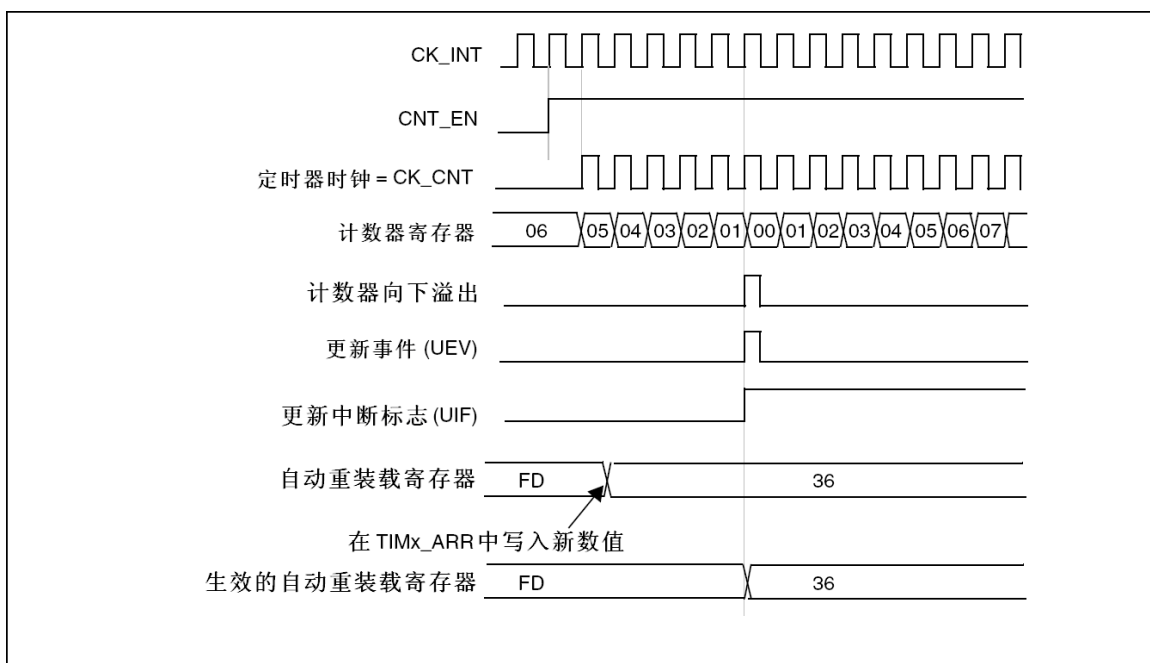


图31. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

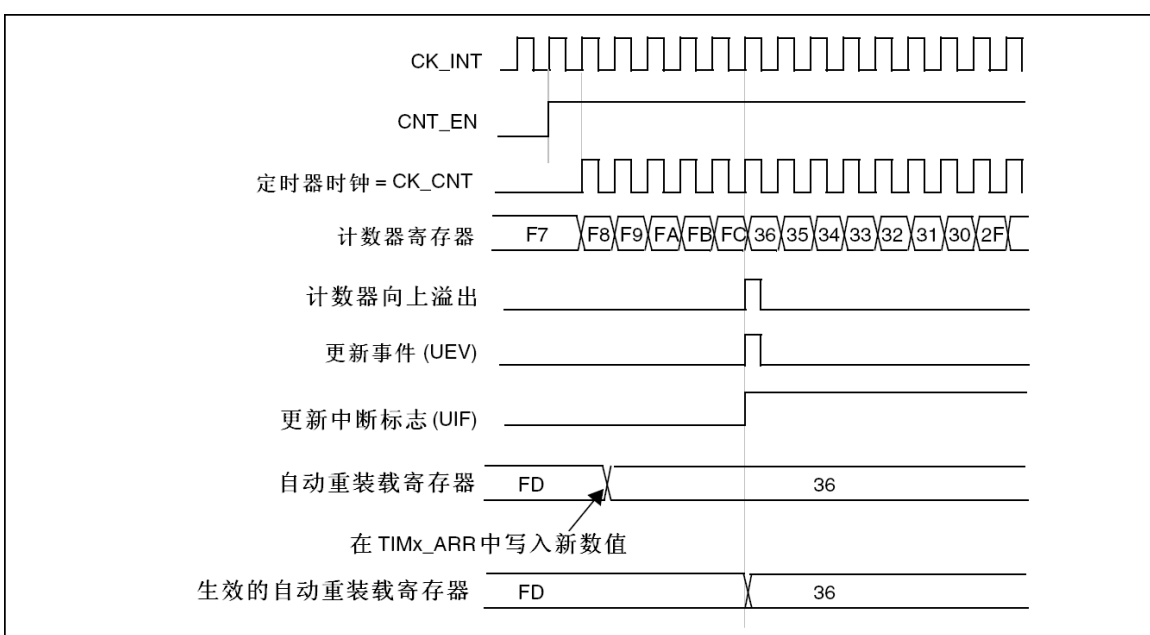


图32. 计数器时序图，ARPE=1 时的更新事件（计数器溢出）

15.3.3. 时钟源选择

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: 外部输入脚 (TIx)
- 外部时钟模式 2: 外部触发输入 (ETRx)
- 内部触发输入 (ITRx): 使用一个定时器作为另一个定时器的预分频器, 如可以配置一个定时器 Timer1 而作为另一个定时器 Timer2 的预分频器。

15.3.3.1. 内部时钟源 (CK_INT)

如果禁止了从模式控制器 (TIM2_SMCR 寄存器的 SMS=000), 则 CEN、DIR (TIM2_CR1 寄存器) 和 UG 位 (TIM2_EGR 寄存器) 是事实上的控制位, 并且只能被软件修改 (UG 位仍被自动清除)。只要 CEN 位被写成'1', 预分频器的时钟就由内部时钟 CK_INT 提供。

下图显示了控制电路和向上计数器在一般模式下, 不带预分频器时的操作。

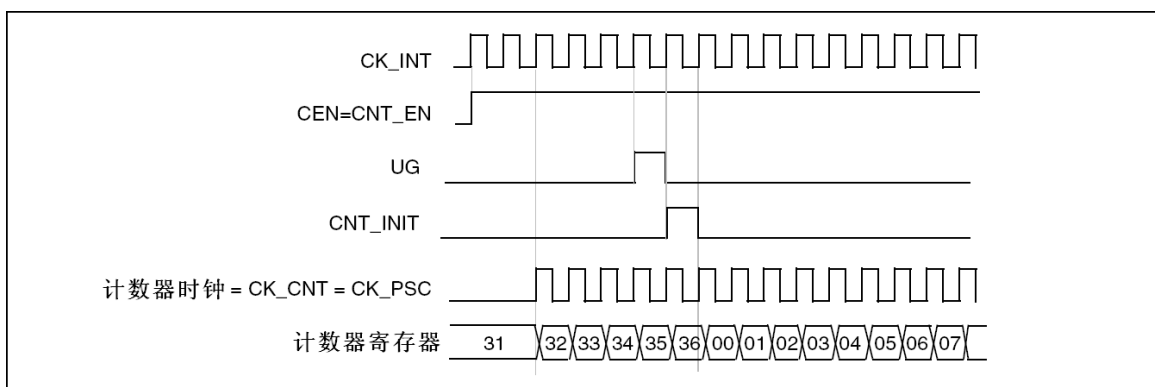


图33. 一般模式下的控制电路, 内部时钟分频因子为 1

15.3.3.2. 外部时钟源模式 1

当 TIM2_SMCR 寄存器的 SMS=111 时, 此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

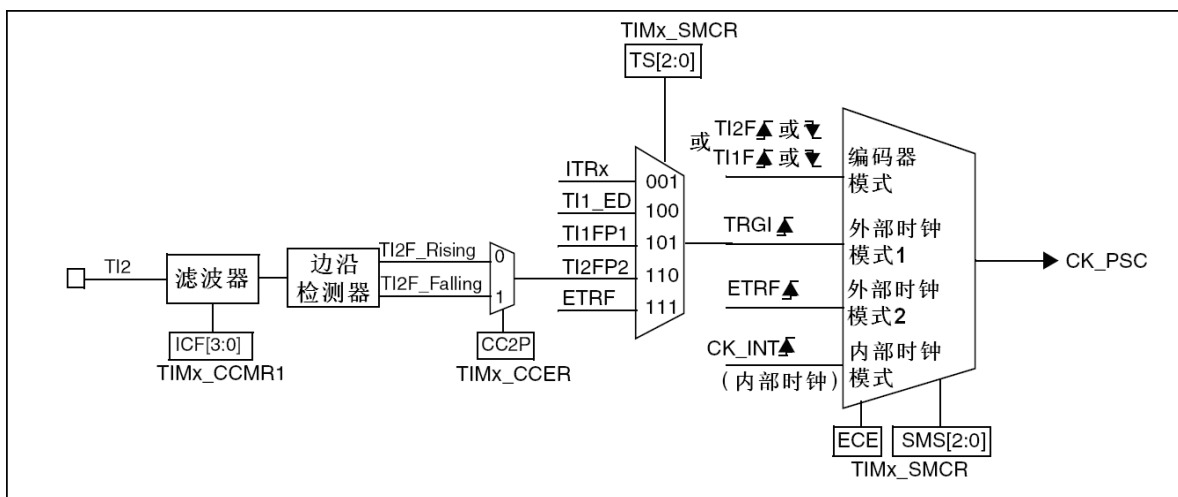


图34. I2 外部时钟连接例子

例如, 要配置向上计数器在 T12 输入端的上升沿计数, 使用下列步骤:

- 1) 配置 TIM2_CCMR1 寄存器 CC2S='01', 配置通道 2 检测 T12 输入的上升沿
- 2) 配置 TIM2_CCMR1 寄存器的 IC2F[3:0], 选择输入滤波器带宽 (如果不需要滤波器, 保持 IC2F=0000)
注: 捕获预分频器不用作触发, 所以不需要对它进行配置。
- 3) 配置 TIM2_CCER 寄存器的 CC2P='0', 选定上升沿极性
- 4) 配置 TIM2_SMCR 寄存器的 SMS='111', 选择定时器外部时钟模式 1
- 5) 配置 TIM2_SMCR 寄存器中的 TS='110', 选定 T12 作为触发输入源

6) 设置 TIM2_CR1 寄存器的 CEN='1', 启动计数器
当上升沿出现在 TI2, 计数器计数一次, 且 TIF 标志被设置。

在 TI2 的上升沿和计数器实际时钟之间的延时, 取决于在 TI2 输入端的重新同步电路。

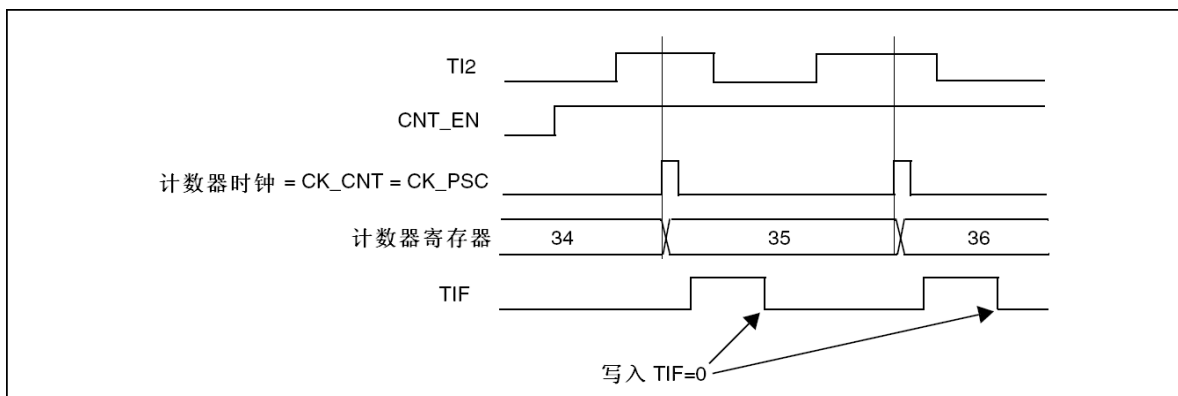


图35. 外部时钟模式 1 下的控制电路

15.3.3.3. 外部时钟源模式 2

选定此模式的方法为: 令 TIM2_SMCR 寄存器中的 ECE=1 计数器能够在外部触发 ETR 的每一个上升沿或下降沿计数。下图是外部触发输入的框图。

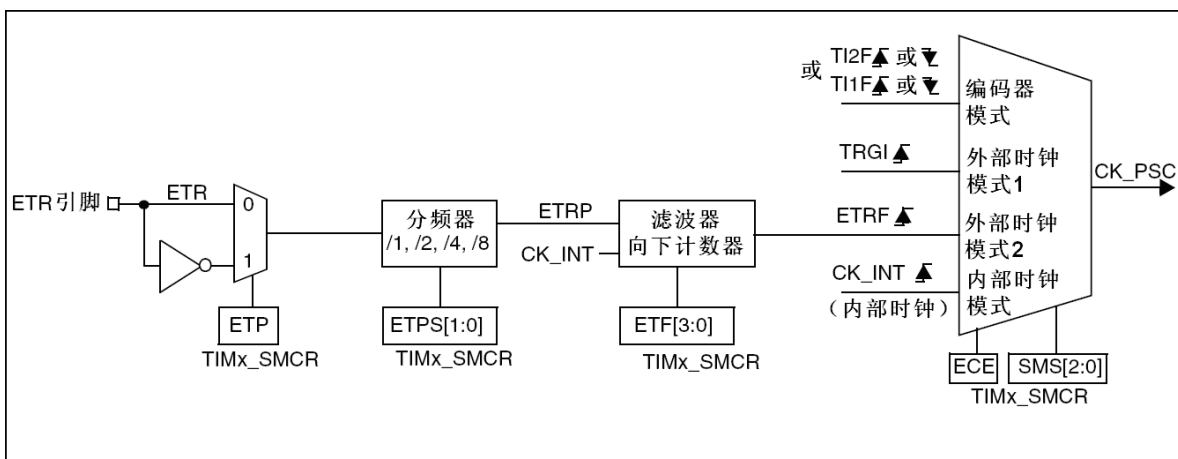


图36. 外部触发输入框图

例如, 要配置在 ETR 下每 2 个上升沿计数一次的向上计数器, 使用下列步骤:

- 1) 本例中不需要滤波器, 置 TIM2_SMCR 寄存器中的 ETF[3:0]=0000
- 2) 设置预分频器, 置 TIM2_SMCR 寄存器中的 ETPS[1:0]=01
- 3) 设置在 ETR 的上升沿检测, 置 TIM2_SMCR 寄存器中的 ETP=0
- 4) 开启外部时钟模式 2, 置 TIM2_SMCR 寄存器中的 ECE=1
- 5) 启动计数器, 置 TIM2_CR1 寄存器中的 CEN=1 计数器在每 2 个 ETR 上升沿计数一次。

在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的重新同步电路。

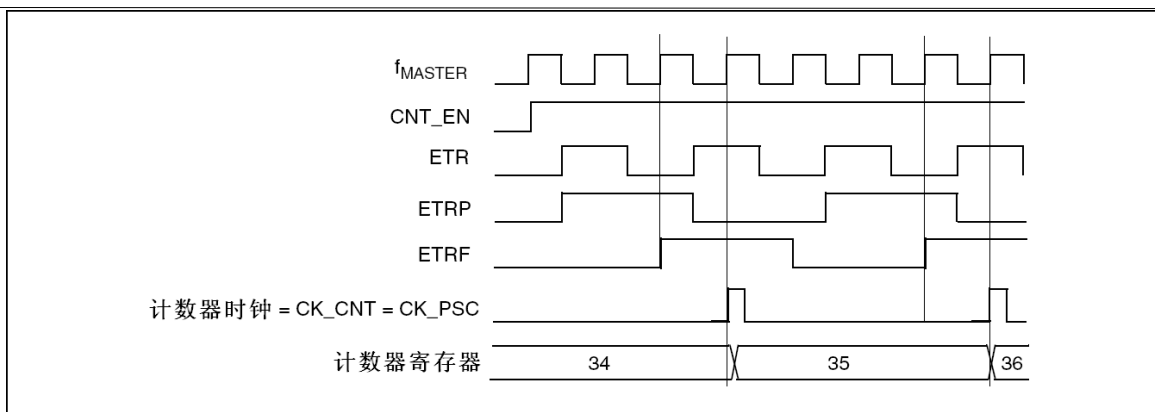


图37. 外部时钟模式 2 下的控制电路

15.3.4. 捕获/比较通道

每一个捕获/比较通道都是围绕着一个捕获/比较寄存器（包含影子寄存器），包括捕获的输入部分（数字滤波、多路复用和预分频器），和输出部分（比较器和输出控制）。下面几张图是一个捕获/比较通道概览。

输入部分对相应的 TIx 输入信号采样，并产生一个滤波后的信号 $TIxF$ 。然后，一个带极性选择的边缘检测器产生一个信号 ($TIxFPx$)，它可以作为从模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器 ($ICxPS$)。

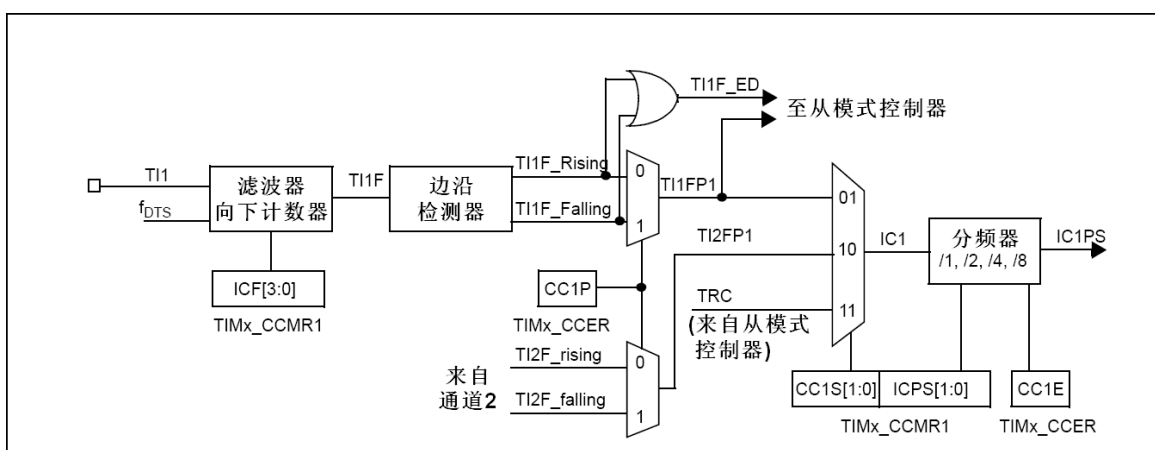


图38. 捕获/比较通道（如：通道 1 输入部分）

输出部分产生一个中间波形 $OCxRef$ （高有效）作为基准，链的末端决定最终输出信号的极性。

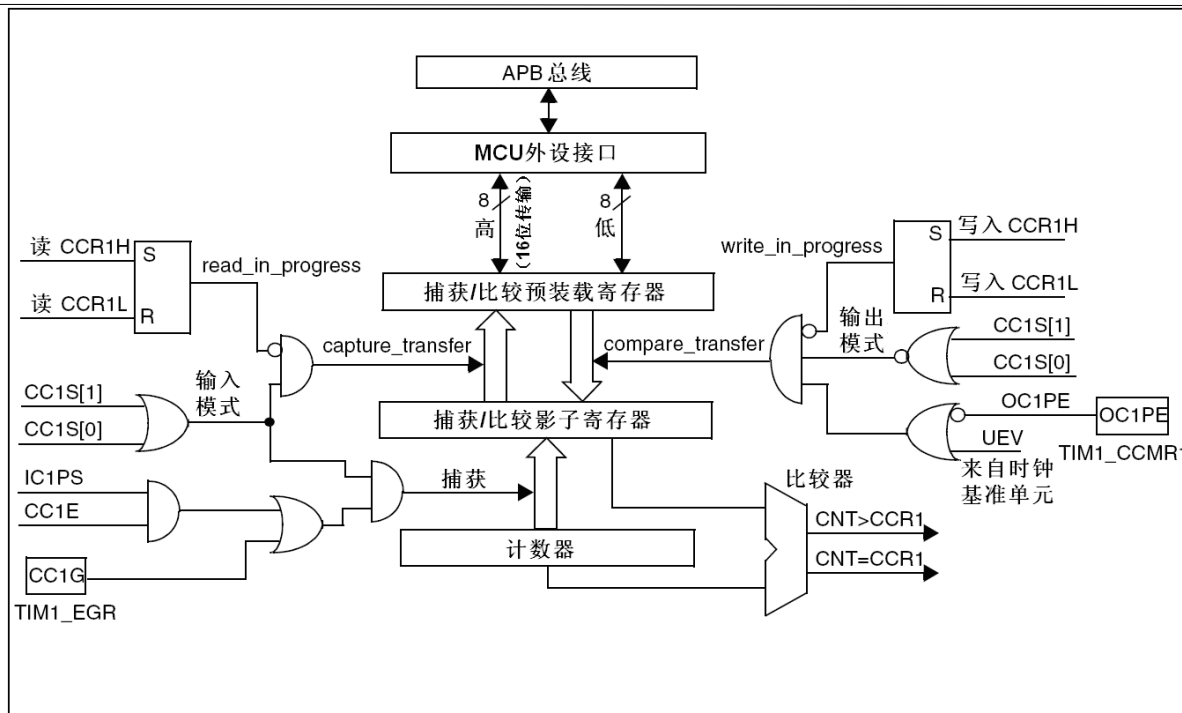


图39. 捕获/比较通道 1 的主电路

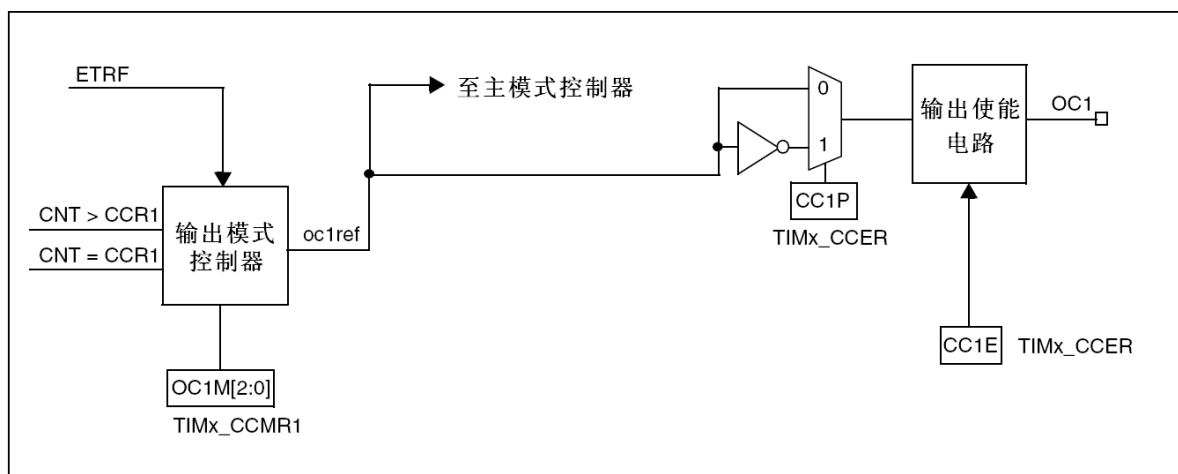


图40. 捕获/比较通道的输出部分（通道 1）

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。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

15.3.5. 输入捕获模式

在输入捕获模式下，当检测到 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（TIM2_CCRx）中。当捕获事件发生时，相应的 CCxIF 标志（TIM2_SR 寄存器）被置'1'，如果使能了中断或者 DMA 操作，则将产生中断或者 DMA 操作。如果捕获事件发生时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF（TIM2_SR 寄存器）被置'1'。写 CCxIF=0 可清除 CCxIF，或读取存储在 TIM2_CCRx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIM2_CCR1 寄存器中，步骤如下：

- 选择有效输入端：TIM2_CCR1 必须连接到 TI1 输入，所以写入 TIM2_CCR1 寄存器中的 CC1S=01，只要 CC1S 不为'00'，通道被配置为输入，并且 TIM1_CCR1 寄存器变为只读。

根据输入信号的特点，配置输入滤波器为所需的带宽（即输入为 TIx 时，输入滤波器控制位是 TIM2_CCMRx

寄存器中的 ICx 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期。因此我们可以 (以 fDTS 频率) 连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIM2_CCMR1 寄存器中写入 IC1F=0011。

- 选择 TI1 通道的有效转换边沿，在 TIM2_CCER 寄存器中写入 CC1P=0 (上升沿)。
- 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止 (写 TIM2_CCMR1 寄存器的 IC1PS=00)。
- 设置 TIM2_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
- 如果需要，通过设置 TIM2_DIER 寄存器中的 CC1IE 位允许相关中断请求，通过设置 TIM2_DIER 寄存器中的 CC1DE 位允许 DMA 请求。当发生一个输入捕获时：
- 产生有效的电平转换时，计数器的值被传送到 TIM2_CCR1 寄存器。
- CC1IF 标志被设置 (中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置'1'。
- 如设置了 CC1IE 位，则会产生一个中断。
- 如设置了 CC1DE 位，则还会产生一个 DMA 请求。

为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注：设置 TIM2_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断和/或 DMA 请求。

15.3.6. PWM 输入模式

该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

- 两个 ICx 信号被映射至同一个 Tlx 输入。
- 这 2 个 ICx 信号为边沿有效，但是极性相反。
- 其中一个 TlxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。

例如，你需要测量输入到 TI1 上的 PWM 信号的长度 (TIM2_CCR1 寄存器) 和占空比 (TIM2_CCR2 寄存器)，具体步骤如下 (取决于 CK_INT 的频率和预分频器的值)

- 选择 TIM2_CCR1 的有效输入：置 TIM2_CCMR1 寄存器的 CC1S=01 (选择 TI1)。
- 选择 TI1FP1 的有效极性 (用来捕获数据到 TIM2_CCR1 中和清除计数器)：置 CC1P=0 (上升沿有效)。
- 选择 TIM2_CCR2 的有效输入：置 TIM2_CCMR1 寄存器的 CC2S=10 (选择 TI1)。
- 选择 TI1FP2 的有效极性 (捕获数据到 TIM2_CCR2)：置 CC2P=1 (下降沿有效)。
- 选择有效的触发输入信号：置 TIM2_SMCR 寄存器中的 TS=101 (选择 TI1FP1)。
- 配置从模式控制器为复位模式：置 TIM2_SMCR 中的 SMS=100。
- 使能捕获：置 TIM2_CCER 寄存器中 CC1E=1 且 CC2E=1。

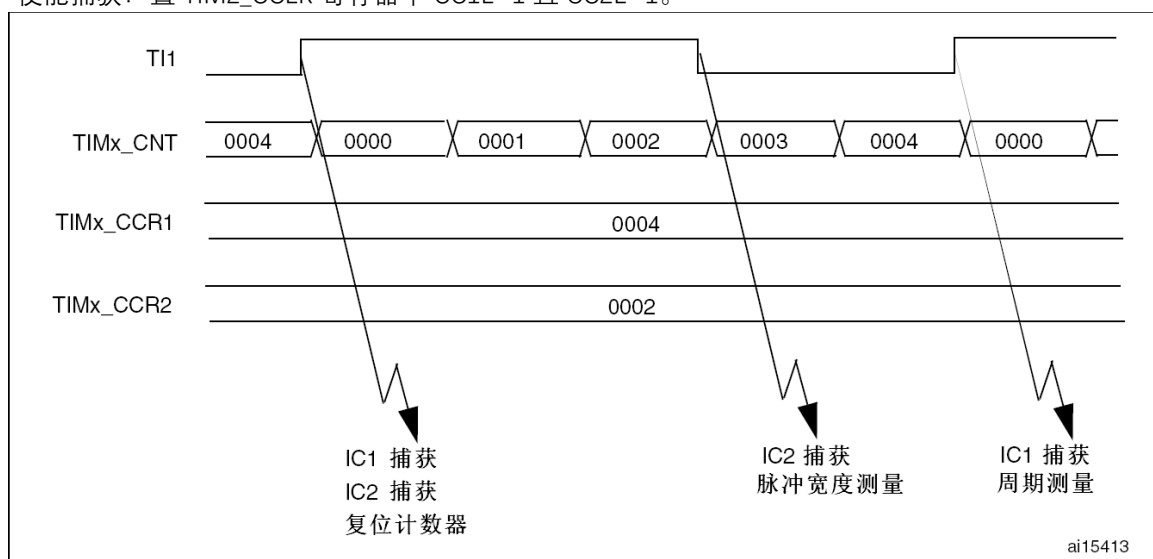


图41. PWM 输入模式时序

由于只有 TI1FP1 和 TI2FP2 连到了从模式控制器，所以 PWM 输入模式只能使用 TIM2_CH1/TIM2_CH2 信

号。

15.3.7. 强置输出模式

在输出模式 (TIM2_CCMRx 寄存器中 CCxS=00) 下, 输出比较信号 (OCxREF 和相应的 OCx) 能够直接由软件强置为有效或无效状态, 而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIM2_CCMRx 寄存器中相应的 OCxM=101, 即可强置输出比较信号 (OCxREF/OCx) 为有效状态。这样 OCxREF 被强置为高电平 (OCxREF 始终为高电平有效), 同时 OCx 得到 CCxP 极性位相反的值。

例如: CCxP=0 (OCx 高电平有效), 则 OCx 被强置为高电平。

置 TIM2_CCMRx 寄存器中的 OCxM=100, 可强置 OCxREF 信号为低。

该模式下, 在 TIM2_CCRx 影子寄存器和计数器之间的比较仍然在进行, 相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。这将会在下面的输出比较模式一节中介绍。

15.3.8. 输出比较模式

此项功能是用来控制一个输出波形, 或者指示一段给定的时间已经到时。当计数器与捕获/比较寄存器的内容相同时, 输出比较功能做如下操作:

- 将输出比较模式 (TIM2_CCMRx 寄存器中的 OCxM 位) 和输出极性 (TIM2_CCER 寄存器中的 CCxP 位) 定义的值输出到对应的引脚上。在比较匹配时, 输出引脚可以保持它的电平 (OCxM=000)、被设置成有效电平 (OCxM=001)、被设置成无效电平 (OCxM=010) 或进行翻转 (OCxM=011)。
- 设置中断状态寄存器中的标志位 (TIM2_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽 (TIM2_DIER 寄存器中的 CCxIE 位), 则产生一个中断。
- 若设置了相应的使能位 (TIM2_DIER 寄存器中的 CCxDE 位, TIM2_CR2 寄存器中的 CCDS 位选择 DMA 请求功能), 则产生一个 DMA 请求。

TIM2_CCMRx 中的 OCxPE 位选择 TIM2_CCRx 寄存器是否需要使用预装载寄存器。在输出比较模式下, 更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。

同步的精度可以达到计数器的一个计数周期。输出比较模式 (在单脉冲模式下) 也能用来输出一个单脉冲。

输出比较模式的配置步骤:

- 1) 选择计数器时钟 (内部, 外部, 预分频器)
- 2) 将相应的数据写入 TIM2_ARR 和 TIM2_CCRx 寄存器中
- 3) 如果要产生一个中断请求和/或一个 DMA 请求, 设置 CCxIE 位和/或 CCxDE 位。
- 4) 选择输出模式, 例如当计数器 CNT 与 CCRx 匹配时翻转 OCx 的输出引脚, CCRx 预装载未用, 开启 OCx 输出且高电平有效, 则必须设置 OCxM='011'、OCxPE='0'、CCxP='0'和 CCxE='1'。
- 5) 设置 TIM2_CR1 寄存器的 CEN 位启动计数器

TIM2_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形, 条件是未使用预装载寄存器 (OCxPE='0', 否则 TIM2_CCRx 影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

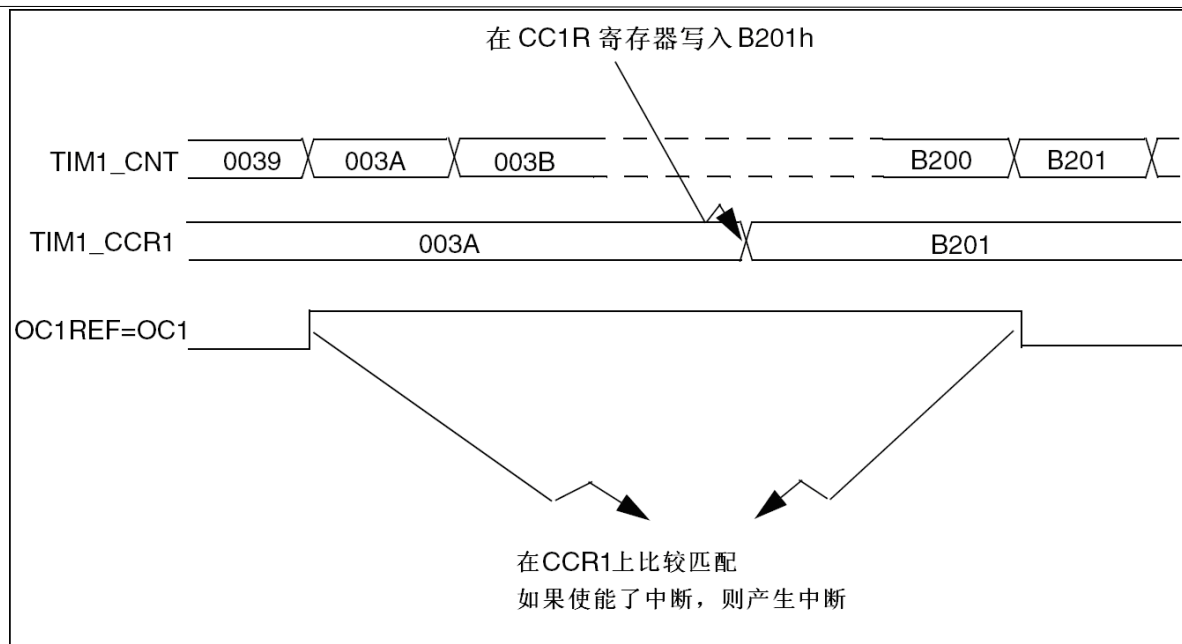


图42. 输出比较模式，翻转 OC1

15.3.9. PWM 模式

脉冲宽度调制模式可以产生一个由 TIM2_ARR 寄存器确定频率、由 TIM2_CCRx 寄存器确定占空比的信号。

在 TIM2_CCMRx 寄存器中的 OCxM 位写入 '110' (PWM 模式 1) 或 '111' (PWM 模式 2)，能够独立地设置每个 OCx 输出通道产生一路 PWM。必须设置 TIM2_CCMRx 寄存器 OCxPE 位以使能相应的预装载寄存器，最后还要设置 TIM2_CR1 寄存器的 ARPE 位，(在向上计数或中心对称模式中)使能自动重载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIM2_EGR 寄存器中的 UG 位来初始化所有的寄存器。

OCx 的极性可以通过软件在 TIM2_CCER 寄存器中的 CCxP 位设置，它可以设置为高电平有效或低电平有效。TIM2_CCER 寄存器中的 CCxE 位控制 OCx 输出使能。详见 TIM2_CCERx 寄存器的描述。

在 PWM 模式 (模式 1 或模式 2) 下，TIM2_CNT 和 TIM2_CCRx 始终在进行比较，(依据计数器的计数方向)以确定是否符合 $TIM2_CCRx \leq TIM2_CNT$ 或者 $TIM2_CNT \leq TIM2_CCRx$ 。然而为了与 OCREF_CLR 的功能 (在下一个 PWM 周期之前，ETR 信号上的一个外部事件能够清除 OCxREF) 一致，OCxREF 信号只能在下述条件下产生：

- 当比较的结果改变，或
- 当输出比较模式 (TIM2_CCMRx 寄存器中的 OCxM 位) 从“冻结” (无比较，OCxM='000') 切换到某个 PWM 模式 (OCxM='110'或'111')。

这样在运行中可以通过软件强置 PWM 输出。

根据 TIM2_CR1 寄存器中 CMS 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

15.3.9.1. PWM 边沿对齐模式

15.3.9.1.1. 向上计数配置

当 TIM2_CR1 寄存器中的 DIR 位为低的时候执行向上计数。

下面是一个 PWM 模式 1 的例子。当 $TIM2_CNT < TIM2_CCRx$ 时 PWM 信号参考 OCxREF 为高，否则为低。如果 TIM2_CCRx 中的比较值大于自动重载值 (TIM2_ARR)，则 OCxREF 保持为 '1'。如果比较值为 0，则 OCxREF 保持为 '0'。下图为 TIM2_ARR=8 时边沿对齐的 PWM 波形实例。

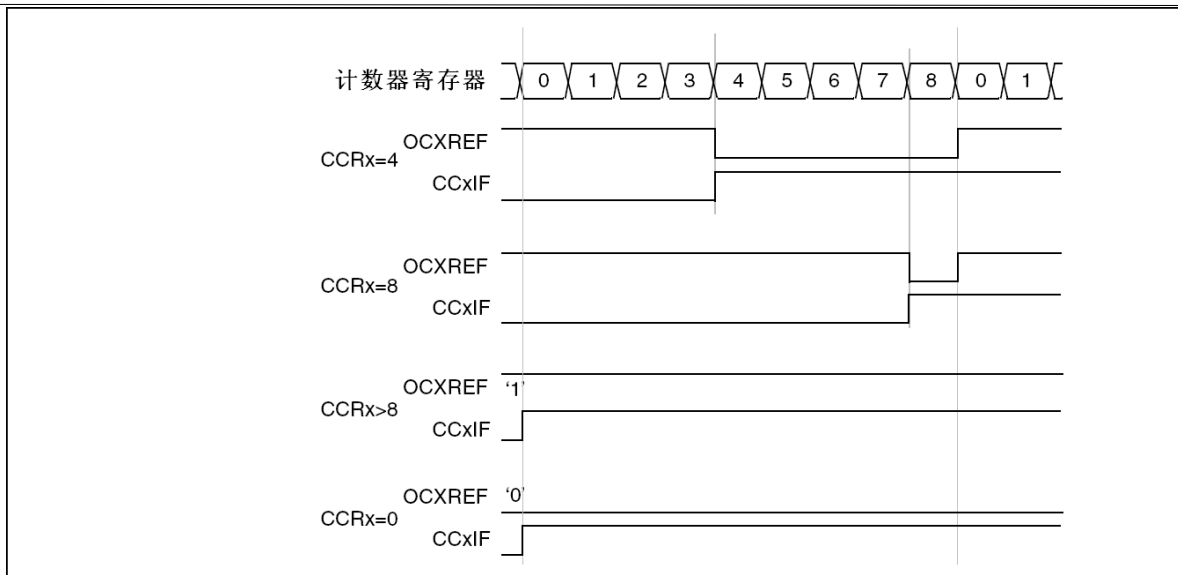


图43. 边沿对齐的 PWM 波形 (ARR=8)

15.3.9.1.2. 向下计数配置

当 TIM2_CR1 寄存器的 DIR 位为高时执行向下计数。

在 PWM 模式 1，当 TIM2_CNT>TIM2_CCRx 时参考信号 OCxREF 为低，否则为高。如果 TIM2_CCRx 中的比较值大于 TIM2_ARR 中的自动重装载值，则 OCxREF 保持为'1'。该模式下不能产生 0% 的 PWM 波形。

15.3.9.2. PWM 中央对齐模式

当 TIM2_CR1 寄存器中的 CMS 位不为'00'时，为中央对齐模式（所有其他的配置对 OCxREF/OCx 信号都有相同的作用）。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置'1'、在计数器向下计数时被置'1'、或在计数器向上和向下计数时被置'1'。TIM2_CR1 寄存器中的计数方向位（DIR）由硬件更新，不要用软件修改它。参看 14.3.2 节的中央对齐模式。

下图给出了一些中央对齐的 PWM 波形的例子

- TIM2_ARR=8
- PWM 模式 1
- TIM2_CR1 寄存器中的 CMS=01，在中央对齐模式 1 时，当计数器向下计数时设置比较标志。

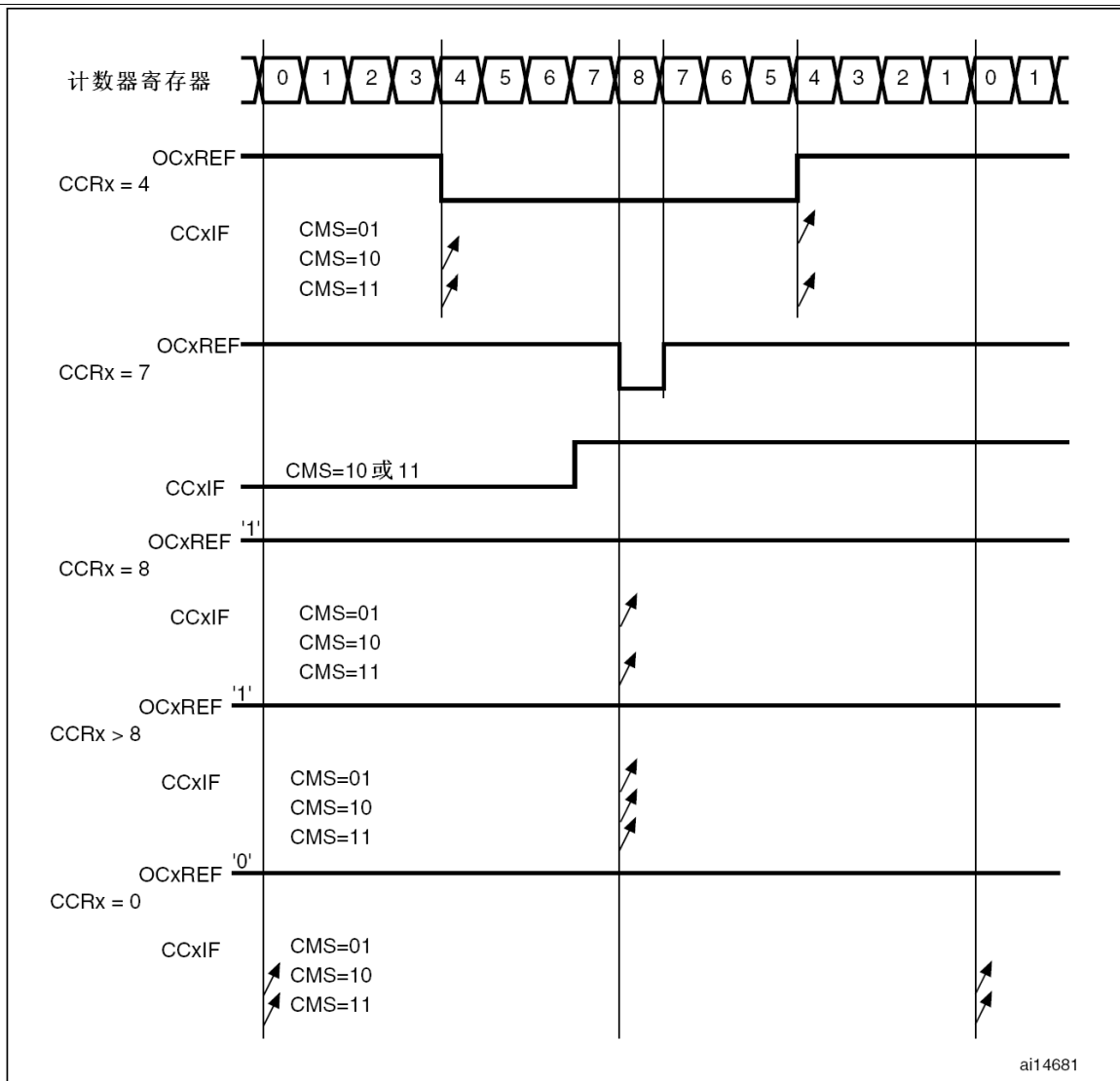


图44. 中央对齐的 PWM 波形 (APR=8)

使用中央对齐模式的提示:

- 进入中央对齐模式时, 使用当前的向上/向下计数配置; 这意味着计数器向上还是向下计数取决于 TIM2_CR1 寄存器中 DIR 位的当前值。此外, 软件不能同时修改 DIR 和 CMS 位。
- 不推荐当运行在中央对齐模式时改写计数器, 因为这会产生不可预知的结果。特别地:
 - 如果写入计数器的值大于自动重加载的值 ($TIM2_CNT > TIM2_ARR$), 则方向不会被更新。例如, 如果计数器正在向上计数, 它就会继续向上计数。
 - 如果将 0 或者 TIM2_ARR 的值写入计数器, 方向被更新, 但不产生更新事件 UEV。
- 使用中央对齐模式最保险的方法, 就是在启动计数器之前产生一个软件更新 (设置 TIM2_EGR 位中的 UG 位), 不要在计数进行过程中修改计数器的值。

15.3.10. 单脉冲模式

单脉冲模式 (OPM) 是前述众多模式的一个特例。这种模式允许计数器响应一个激励, 并在一个程序可控的延时之后, 产生一个脉宽可程序控制的脉冲。

可以通过从模式控制器启动计数器, 在输出比较模式或者 PWM 模式下产生波形。设置 TIM2_CR1 寄存器中的 OPM 位将选择单脉冲模式, 这样可以使计数器自动地在产生下一个更新事件 UEV 时停止。

仅当比较值与计数器的初始值不同时, 才能产生一个脉冲。启动之前 (当定时器正在等待触发), 必须如下配置: 向上计数方式: $CNT < CCRx \leq ARR$ (特别地, $0 < CCRx$), 向下计数方式: $CNT > CCRx$ 。

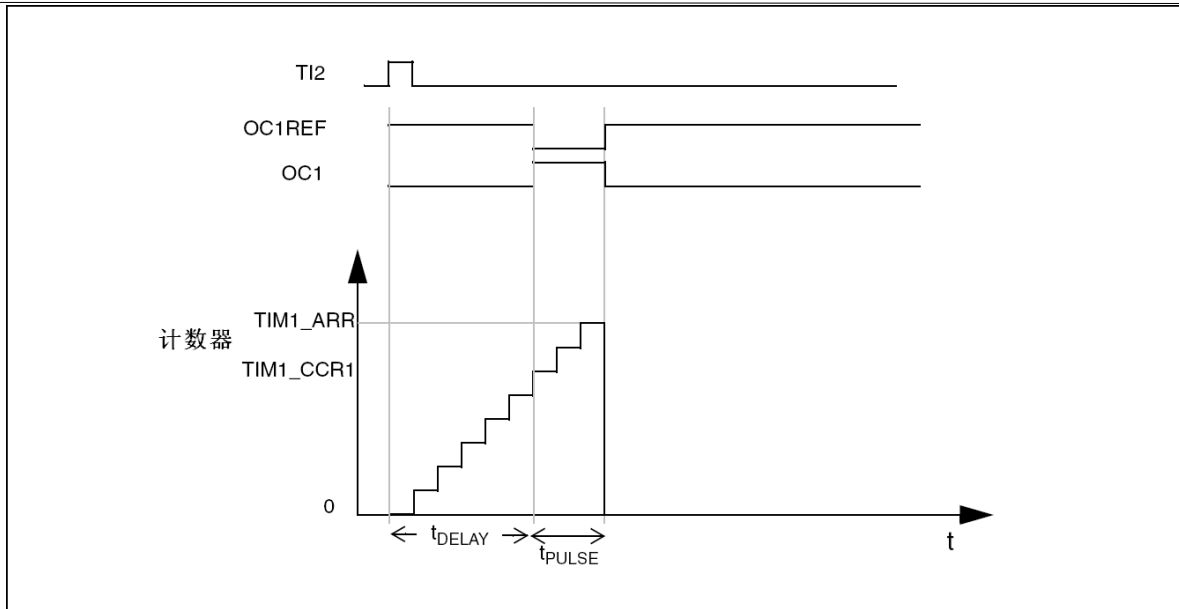


图45. 单脉冲模式

例如，你需要在从 TI2 输入脚上检测到一个上升沿开始，延迟 t_{DELAY} 之后，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

假定 TI2FP2 作为触发 1:

- 置 TIM2_CCMR1 寄存器中的 CC2S='01'，把 TI2FP2 映像到 TI2。
- 置 TIM2_CCER 寄存器中的 CC2P='0'，使 TI2FP2 能够检测上升沿。
- 置 TIM2_SMCR 寄存器中的 TS='110'，TI2FP2 作为从模式控制器的触发 (TRGI)。
- 置 TIM2_SMCR 寄存器中的 SMS='110' (触发模式)，TI2FP2 被用来启动计数器。
- OPM 波形由写入比较寄存器的数值决定 (要考虑时钟频率和计数器预分频器)
- t_{DELAY} 由写入 TIM2_CCR1 寄存器中的值定义。
- t_{PULSE} 由自动装载值和比较值之间的差值定义 ($\text{TIM2_ARR} - \text{TIM2_CCR1}$)。
- 假定当发生比较匹配时要产生从 '0' 到 '1' 的波形，当计数器到达预装载值时要产生一个从 '1' 到 '0' 的波形；

首先要置 TIM2_CCMR1 寄存器的 OC1M='111'，进入 PWM 模式 2；根据需要选择性地使能预装载寄存器：置 TIM2_CCMR1 中的 OC1PE='1' 和 TIM2_CR1 寄存器中的 ARPE；然后在 TIM2_CCR1 寄存器中填写比较值，在 TIM2_ARR 寄存器中填写自动装载值，修改 UG 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P='0'。在这个例子中，TIM2_CR1 寄存器中的 DIR 和 CMS 位应该置低。因为只需一个脉冲，所以必须设置 TIM2_CR1 寄存器中的 OPM='1'，在下一个更新事件（当计数器从自动装载值翻转到 0）时停止计数。

15.3.10.1. 特殊情况：OCx 快速使能：

在单脉冲模式下，在 Tix 输入脚的边沿检测逻辑设置 CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 t_{DELAY} 。

如果要以最小延时输出波形，可以设置 TIM2_CCMRx 寄存器中的 OCxFE 位；此时 OCxREF (和 OCx) 被强制响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCxFE 只在通道配置为 PWM1 和 PWM2 模式时起作用。

15.3.11. 在外部事件时清除 OCxREF 信号

对于一个给定的通道，设置 TIM2_CCMRx 寄存器中对应的 OCxCE 位为 '1'，能够用 ETRF 输入端的高电平把 OCxREF 信号拉低，OCxREF 信号将保持为低直到发生下一次的更新事件 UEV。

该功能只能用于输出比较和 PWM 模式，而不能用于强置模式。例如，OCxREF 信号可以联到一个比较器的输出，用于控制电流。这时，ETR 必须配置如下：

1. 外部触发预分频器必须处于关闭：TIM2_SMCR 寄存器中的 ETPS[1:0]='00'。

2. 必须禁止外部时钟模式 2: TIM2_SMCR 寄存器中的 ECE='0'。
3. 外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可以根据需要配置。

下图显示了当 ETRF 输入变为高时, 对应不同 OCxCE 的值, OCxREF 信号的动作。在这个例子中, 定时器 TIM2 被置于 PWM 模式。

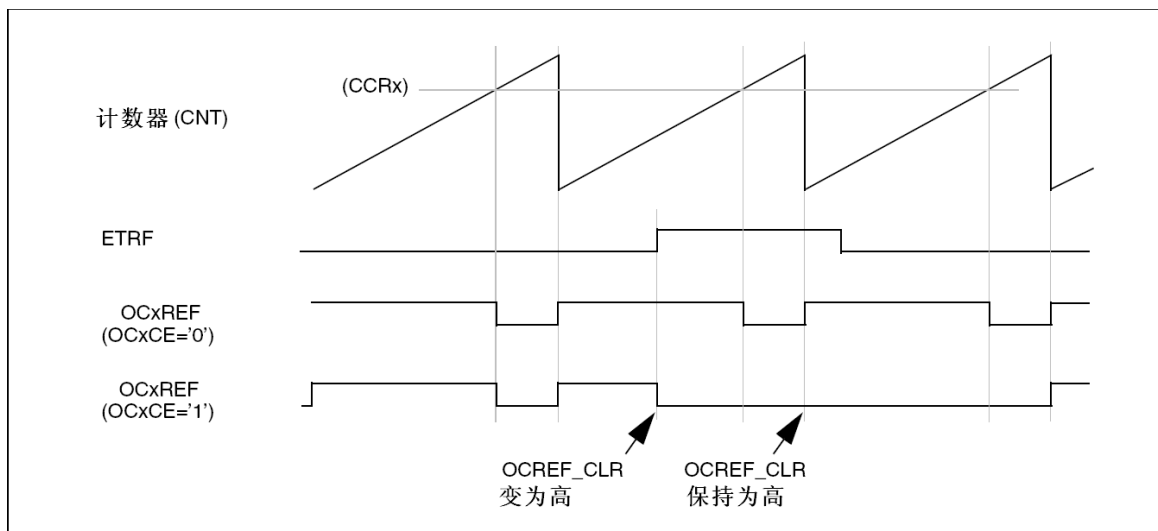


图46. 清除 TIM2 的 OCxREF

15.3.12. 编码器接口模式

选择编码器接口模式的方法是: 如果计数器只在 TI2 的边沿计数, 则置 TIM2_SMCR 寄存器中的 SMS=001; 如果只在 TI1 边沿计数, 则置 SMS=010; 如果计数器同时在 TI1 和 TI2 边沿计数, 则置 SMS=011。

通过设置 TIM2_CCER 寄存器中的 CC1P 和 CC2P 位, 可以选择 TI1 和 TI2 极性; 如果需要, 还可以对输入滤波器编程。

两个输入 TI1 和 TI2 被用来作为增量编码器的接口。参看表 77, 假定计数器已经启动 (TIM2_CR1 寄存器中的 CEN='1'), 计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号; 如果没有滤波和变相, 则 TI1FP1=TI1, TI2FP2=TI2。根据两个输入信号的跳变顺序, 产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序, 计数器向上或向下计数, 同时硬件对 TIM2_CR1 寄存器的 DIR 位进行相应的设置。不管计数器是依靠 TI1 计数、依靠 TI2 计数或者同时依靠 TI1 和 TI2 计数。在任一输入端 (TI1 或者 TI2) 的跳变都会重新计算 DIR 位。

编码器接口模式基本上相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIM2_ARR 寄存器的自动装载值之间连续计数 (根据方向, 或是 0 到 ARR 计数, 或是 ARR 到 0 计数)。所以在开始计数之前必须配置 TIM2_ARR; 同样, 捕获器、比较器、预分频器、触发输出特性等仍工作如常。

在这个模式下, 计数器依照增量编码器的速度和方向被自动的修改, 因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合, 假设 TI1 和 TI2 不同时变换。

表96. 计数方向与编码器信号的关系

有效边沿	相对信号的电平 (TI1FP1对应TI2, TI2FP2对应TI1)	TI1FP1信号		TI2FP2信号	
		上升	下降	上升	下降
仅在TI1计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在TI2计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在TI1和TI2上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

一个外部的增量编码器可以直接与 MCU 连接而不需要外部接口逻辑。但是, 一般会使用比较器将编码器的差动输出转换到数字信号, 这大大增加了抗噪声干扰能力。编码器输出的第三个信号表示机械零点, 可以把它

连接到一个外部中断输入并触发一个计数器复位。

下图是一个计数器操作的实例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。在这个例子中，我们假定配置如下：

- CC1S='01' (TIM2_CCMR1 寄存器, IC1FP1 映射到 TI1)
- CC2S='01' (TIM2_CCMR2 寄存器, IC2FP2 映射到 TI2)
- CC1P='0' (TIM2_CCER 寄存器, IC1FP1 不反相, IC1FP1=TI1)
- CC2P='0' (TIM2_CCER 寄存器, IC2FP2 不反相, IC2FP2=TI2)
- SMS='011' (TIM2_SMCR 寄存器, 所有的输入均在上升沿和下降沿有效)。
- CEN='1' (TIM2_CR1 寄存器, 计数器使能)

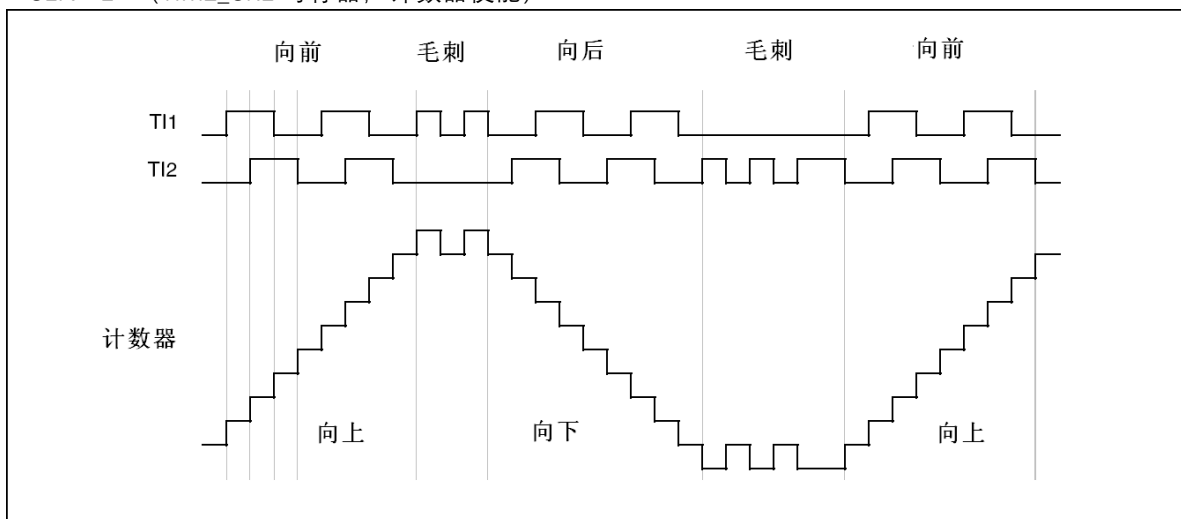


图47. 编码器模式下的计数器操作实例

下图为当 IC1FP1 极性反相时计数器的操作实例 (CC1P='1', 其他配置与上例相同)

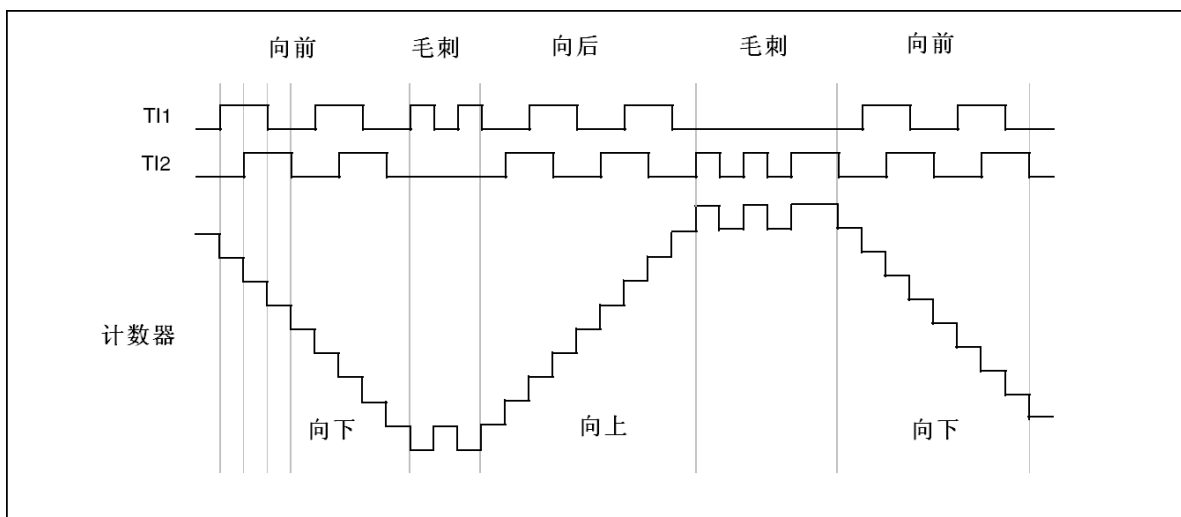


图48. IC1FP1 反相的编码器接口模式实例

当定时器配置成编码器接口模式时，提供传感器当前位置的信息。使用第二个配置在捕获模式的定时器，可以测量两个编码器事件的间隔，获得动态的信息（速度，加速度，减速度）。指示机械零点的编码器输出可被用做此目的。根据两个事件间的间隔，可以按照固定的时间读出计数器。如果可能的话，你可以把计数器的值锁存到第三个输入捕获寄存器（捕获信号必须是周期的并且可以由另一个定时器产生）；也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

15.3.13. 定时器输入异或功能

TIM2_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIM2_CH1、TIM2_CH2 和 TIM2_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。上一章 13.3.18 节给出了此特性用于连接

霍尔传感器的例子。

15.3.14. 定时器和外部触发的同步

TIM2 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

15.3.14.1. 从模式：复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIM2_CR1 寄存器的 URS 位为低，还会产生一个更新事件 UEV；然后所有的预装载寄存器（TIM2_ARR，TIM2_CCRx）都会被更新。

在下面的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽（在本例中，不需要任何滤波器，因此保持 IC1F=0000）。触发操作中不使用捕获预分频器，所以不需要配置它。CC1S 位只选择输入捕获源，即 TIM2_CCMR1 寄存器中 CC1S=01。置 TIM2_CCER 寄存器中 CC1P=0 以确定极性（只检测上升沿）。
- 置 TIM2_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIM2_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIM2_CR1 寄存器中 CEN=1，启动计数器。
- 计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志（TIM2_SR 寄存器中的 TIF 位）被设置，根据 TIM2_DIER 寄存器中 TIE（中断使能）位和 TDE（DMA 使能）位的设置，产生一个中断请求或一个 DMA 请求。

下图显示当自动重装载寄存器 TIM2_ARR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时，取决于 TI1 输入端的重同步电路。

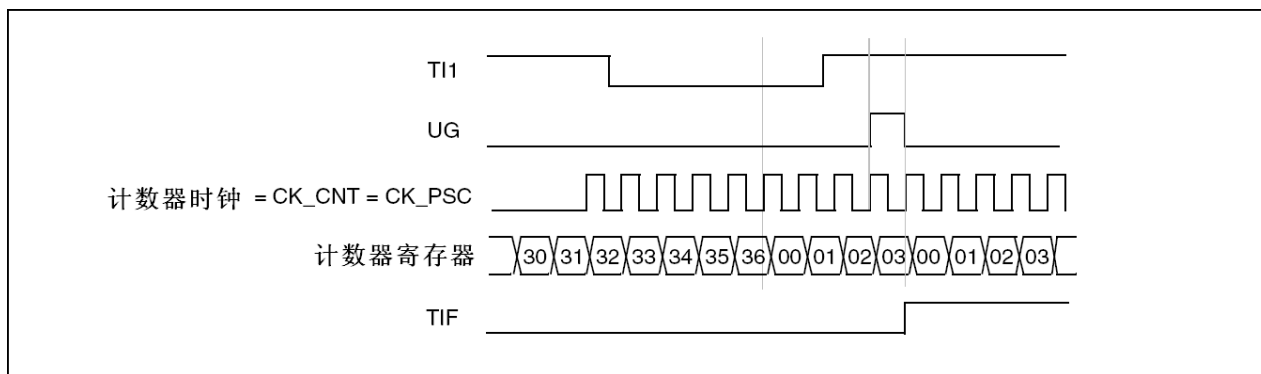


图49. 复位模式下的控制电路

15.3.14.2. 从模式：门控模式

按照选中的输入端电平使能计数器。

在如下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽（本例中，不需要滤波，所以保持 IC1F=0000）。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIM2_CCMR1 寄存器中 CC1S=01。置 TIM2_CCER 寄存器中 CC1P=1 以确定极性（只检测低电平）。
- 置 TIM2_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIM2_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIM2_CR1 寄存器中 CEN=1，启动计数器。在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何。

只要 TI1 为低，计数器开始依据内部时钟计数，在 TI1 变高时停止计数。当计数器开始或停止时都设置 TIM2_SR 中的 TIF 标志。

TI1 上升沿和计数器实际停止之间的延时，取决于 TI1 输入端的重同步电路。

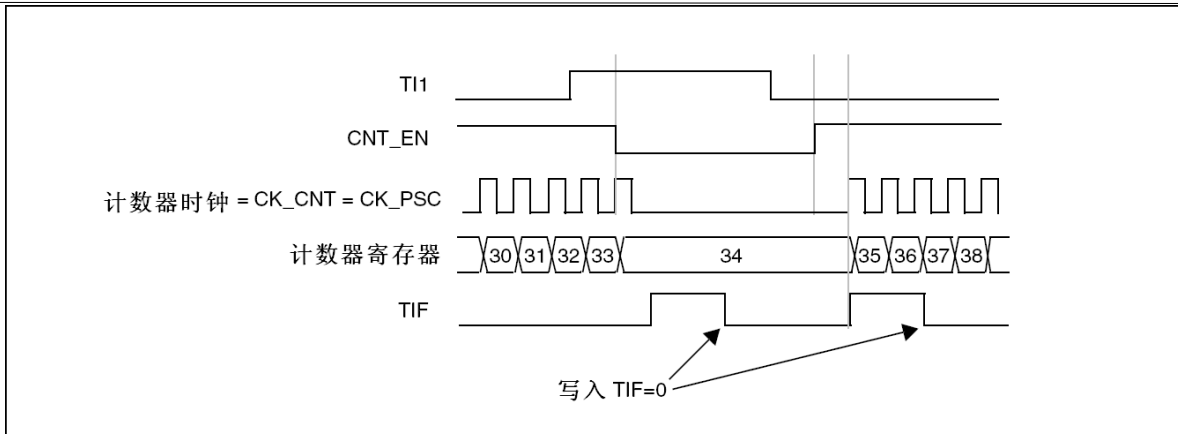


图50. 门控模式下的控制电路

15.3.14.3. 从模式：触发模式

输入端上选中的事件使能计数器。

在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽（本例中，不需要任何滤波器，保持 IC2F=0000）。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIM2_CCMR1 寄存器中 CC2S=01。置 TIM2_CCER 寄存器中 CC2P=1 以确定极性（只检测低电平）。
- 置 TIM2_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIM2_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

当 TI2 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 TIF 标志。

TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

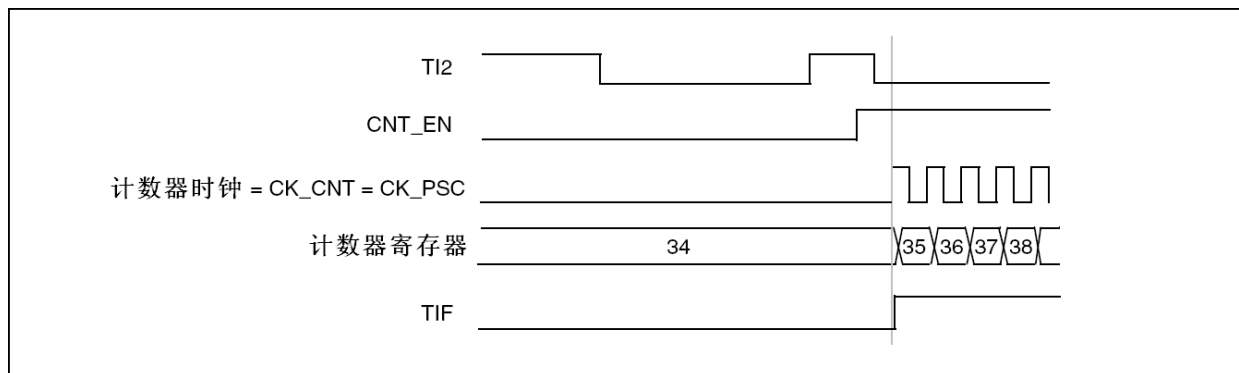


图51. 触发器模式下的控制电路

15.3.14.4. 从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可以与另一种从模式（外部时钟模式 1 和编码器模式除外）一起使用。这时，ETR 信号被用作外部时钟的输入，在复位模式、门控模式或触发模式时可以选择另一个输入作为触发输入。不建议使用 TIM2_SMCR 寄存器的 TS 位选择 ETR 作为 TRGI。

下面的例子中，TI1 上出现一个上升沿之后，计数器即在 ETR 的每一个上升沿向上计数一次：

1) 通过 TIM2_SMCR 寄存器配置外部触发输入电路：

- ETF=0000：没有滤波
- ETPS=00：不用预分频器
- ETP=0：检测 ETR 的上升沿，置 ECE=1 使能外部时钟模式 2

2) 按如下配置通道 1，检测 TI 的上升沿：

- IC1F=0000：没有滤波

- 触发操作中不使用捕获预分频器，不需要配置
- 置 TIM2_CCMR1 寄存器中 CC1S=01，选择输入捕获源
- 置 TIM2_CCER 寄存器中 CC1P=0 以确定极性（只检测上升沿）

3) 置 TIM2_SMCR 寄存器中 SMS=110，配置定时器为触发模式。置 TIM2_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。

当 TI1 上出现一个上升沿时，TIF 标志被设置，计数器开始在 ETR 的上升沿计数。ETR 信号的上升沿和计数器实际复位间的延时，取决于 ETRP 输入端的重同步电路。

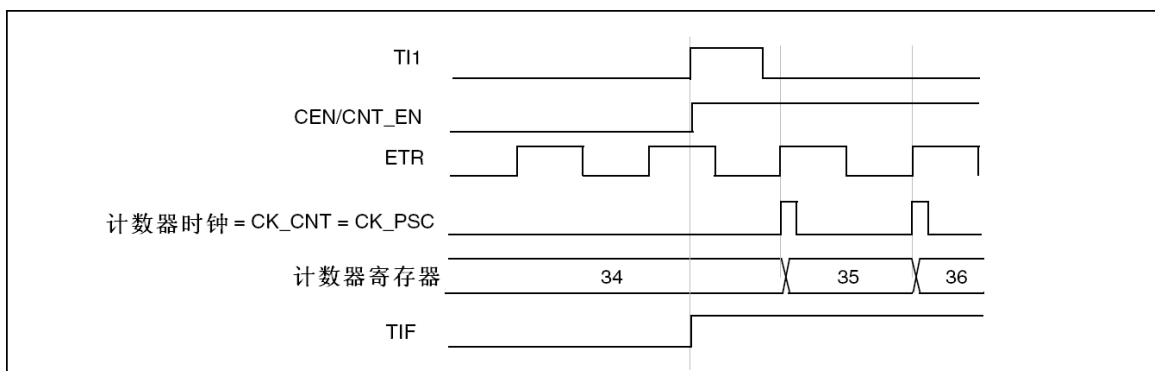


图52. 部时钟模式 2 + 触发模式下的控制电路

15.3.15. 定时器同步

所有 TIM2 定时器在内部相连，用于定时器同步或链接。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。

下图显示了触发选择和主模式选择模块的概况。

Timer1 和 Timer2 的内部连接关系如下表。

表97.

从定时器	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM1	0	TIM2	0	0
TIM2	TIM1	0	0	0

15.3.15.1. 使用一个定时器作为另一个定时器的预分频器

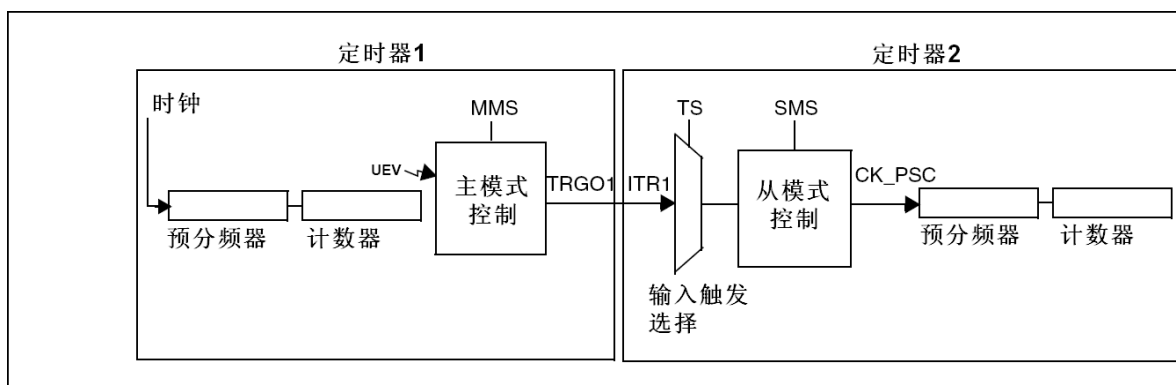


图53. 主/从定时器的例子

如：可以配置定时器 1 作为定时器 2 的预分频器。参考上图，进行下述操作：

- 配置定时器 1 为主模式，它可以在每一个更新事件 UEV 时输出一个周期性的触发信号。在 TIM1_CR2 寄存器的 MMS='010'时，每当产生一个更新事件时在 TRGO1 上输出一个上升沿信号。
- 连接定时器 1 的 TRGO1 输出至定时器 2，设置 TIM2_SMCR 寄存器的 TS='000'，配置定时器 2 为使用 ITR1 作为内部触发的从模式。

- 然后把从模式控制器置于外部时钟模式 1 (TIM2_SMCR 寄存器的 SMS=111); 这样定时器 2 即可由定时器 1 周期性的上升沿 (即定时器 1 的计数器溢出) 信号驱动。
 - 最后, 必须设置相应 (TIM2_CR1 寄存器) 的 CEN 位分别启动两个定时器。
- 注: 如果 OCx 已被选中为定时器 1 的触发输出 (MMS=1xx), 它的上升沿用于驱动定时器 2 的计数器。

15.3.15.2. 使用一个定时器使能另一个定时器

在这个例子中, 定时器 2 的使能由定时器 1 的输出比较控制。参考图 53 的连接。只当定时器 1 的 OC1REF 为高时, 定时器 2 才对分频后的内部时钟计数。两个定时器的时钟频率都是由预分频器对 CK_INT 除以 3 (fCK_CNT=fCK_INT/3) 得到。

- 配置定时器 1 为主模式, 送出它的输出比较参考信号 (OC1REF) 为触发输出 (TIM1_CR2 寄存器的 MMS=100)
 - 配置定时器 1 的 OC1REF 波形 (TIM1_CCMR1 寄存器)
 - 配置定时器 2 从定时器 1 获得输入触发 (TIM2_SMCR 寄存器的 TS=000)
 - 配置定时器 2 为门控模式 (TIM2_SMCR 寄存器的 SMS=101)
 - 置 TIM2_CR1 寄存器的 CEN=1 以使能定时器 2
 - 置 TIM1_CR1 寄存器的 CEN=1 以启动定时器 1
- 注: 定时器 2 的时钟不与定时器 1 的时钟同步, 这个模式只影响定时器 2 计数器的使能信号。

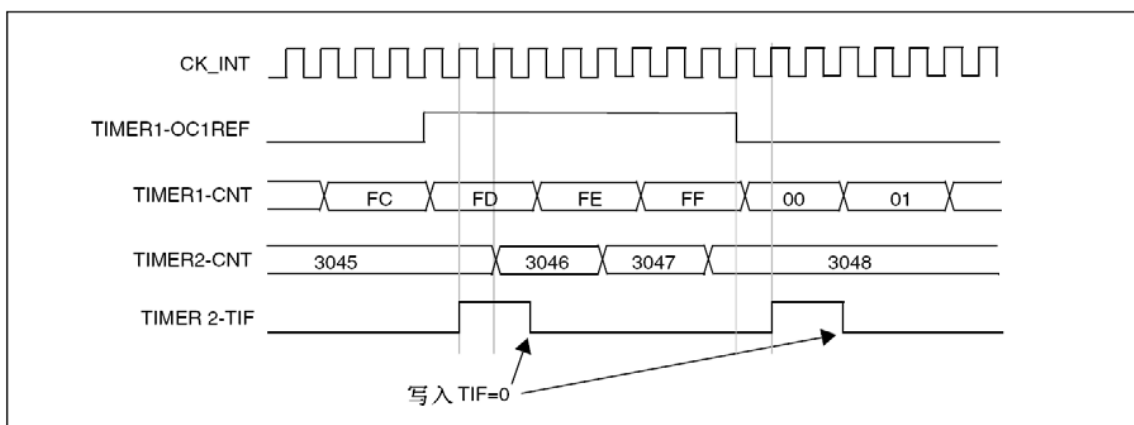


图54. 定时器 1 的 OC1REF 控制定时器 2

在上图的例子中, 在定时器 2 启动之前, 它们的计数器和预分频器未被初始化, 因此它们从当前的数值开始计数。可以在启动定时器 1 之前复位 2 个定时器, 使它们从给定的数值开始, 即在定时器计数器中写入需要的任意数值。写 TIM2_EGR 寄存器的 UG 位即可复位定时器。

在下一个例子中, 需要同步定时器 1 和定时器 2。定时器 1 是主模式并从 0 开始, 定时器 2 是从模式并从 0xE7 开始; 2 个定时器的预分频器系数相同。写'0'到 TIM1_CR1 的 CEN 位将禁止定时器 1, 定时器 2 随即停止。

- 配置定时器 1 为主模式, 送出输出比较 1 参考信号 (OC1REF) 做为触发输出 (TIM1_CR2 寄存器的 MMS=100)。
- 配置定时器 1 的 OC1REF 波形 (TIM1_CCMR1 寄存器)。
- 配置定时器 2 从定时器 1 获得输入触发 (TIM2_SMCR 寄存器的 TS=000)
- 配置定时器 2 为门控模式 (TIM2_SMCR 寄存器的 SMS=101)
- 置 TIM1_EGR 寄存器的 UG='1', 复位定时器 1。
- 置 TIM2_EGR 寄存器的 UG='1', 复位定时器 2。
- 写'0xE7'至定时器 2 的计数器 (TIM2_CNT), 初始化它为 0xE7。
- 置 TIM2_CR1 寄存器的 CEN='1'以使能定时器 2。
- 置 TIM1_CR1 寄存器的 CEN='1'以启动定时器 1。
- 置 TIM1_CR1 寄存器的 CEN='0'以停止定时器 1。

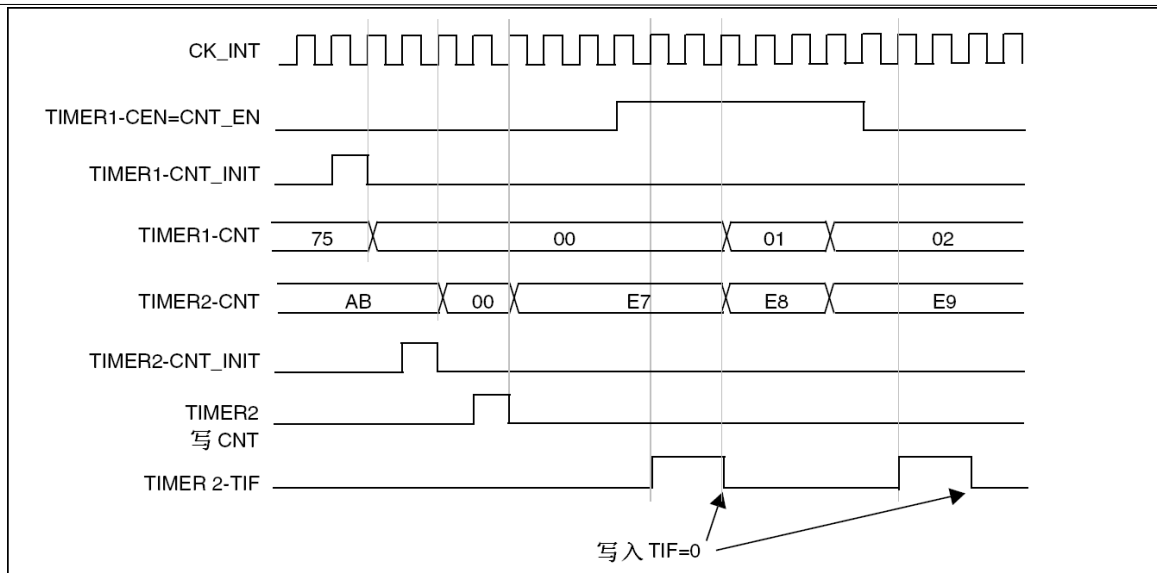


图55. 通过使能定时器 1 可以控制定时器 2

15.3.15.3. 使用一个定时器去启动另一个定时器

在这个例子中，使用定时器 1 的更新事件使能定时器 2。参考图 53 的连接。一旦定时器 1 产生更新事件，定时器 2 即从它当前的数值（可以是非 0）按照分频的内部时钟开始计数。在收到触发信号时，定时器 2 的 CEN 位被自动地置'1'，同时计数器开始计数直到写'0'到 TIM2_CR1 寄存器的 CEN 位。两个定时器的时钟频率都是由预分频器对 CK_INT 除以 3 ($f_{CK_CNT}=f_{CK_INT}/3$)。

- 配置定时器 1 为主模式，送出它的更新事件 (UEV) 做为触发输出 (TIM1_CR2 寄存器的 MMS=010)。
- 配置定时器 1 的周期 (TIM1_ARR 寄存器)。
- 配置定时器 2 从定时器 1 获得输入触发 (TIM2_SMCR 寄存器的 TS=000)
- 配置定时器 2 为触发模式 (TIM2_SMCR 寄存器的 SMS=110)
- 置 TIM1_CR1 寄存器的 CEN=1 以启动定时器 1。

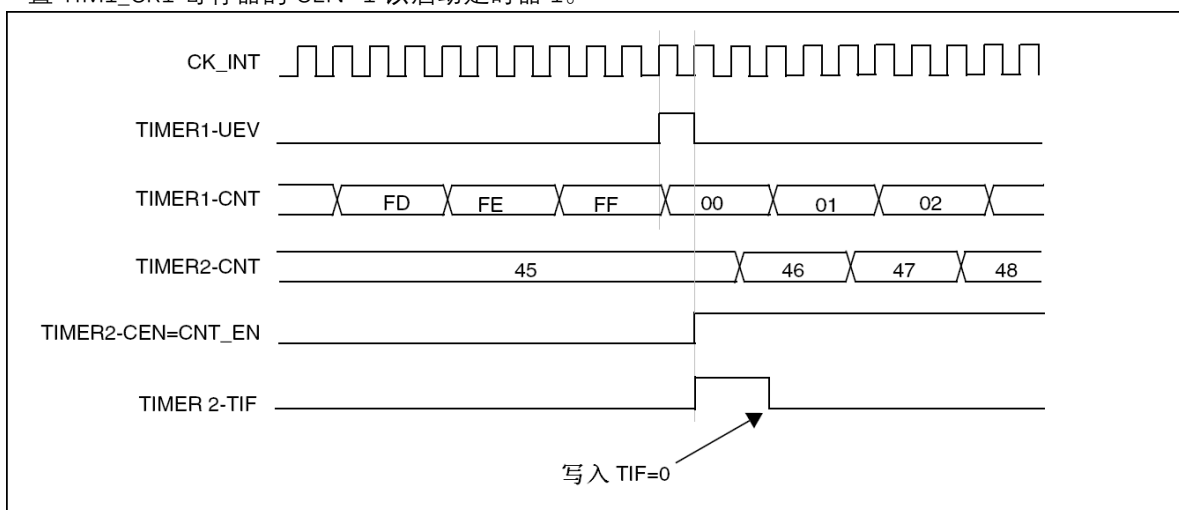


图56. 使用定时器 1 的更新触发定时器 2

在上一个例子中，可以在启动计数之前初始化两个计数器。下图显示在与 0 相同配置情况下，使用触发模式而不是门控模式 (TIM2_SMCR 寄存器的 SMS=110) 的动作。

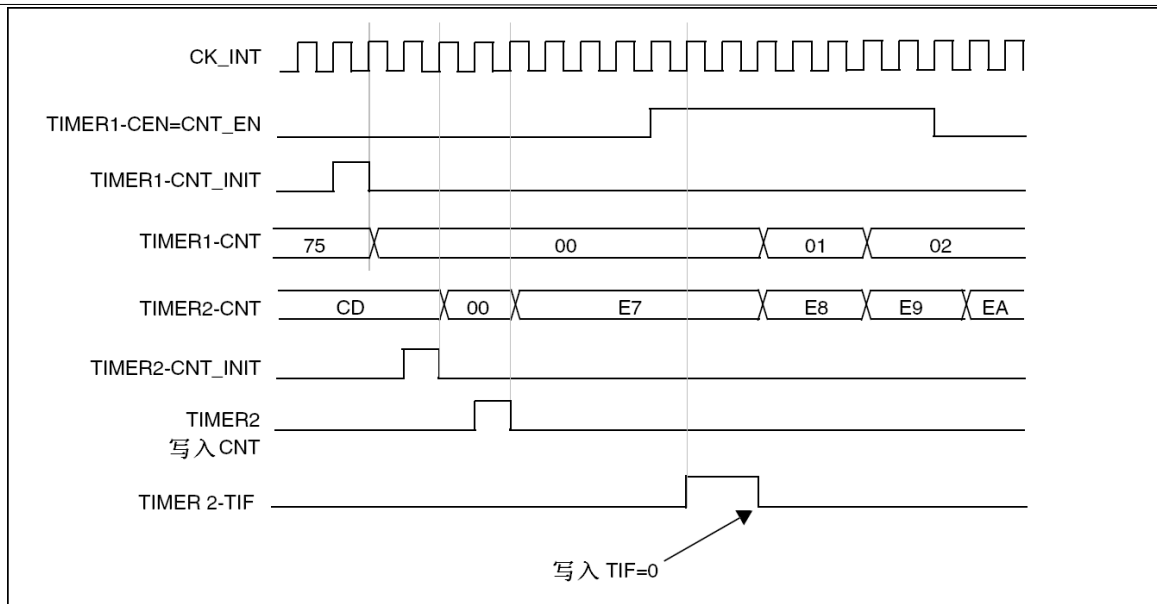


图57. 利用定时器 1 的使能触发定时器 2

15.3.15.4. 使用一个定时器作为另一个的预分频器

这个例子使用定时器 1 作为定时器 2 的预分频器。参考图 53 的连接，配置如下：

- 配置定时器 1 为主模式，送出它的更新事件 UEV 做为触发输出（TIM1_CR2 寄存器的 MMS='010'）。然后每次计数器溢出时输出一个周期信号。
- 配置定时器 1 的周期（TIM1_ARR 寄存器）。
- 配置定时器 2 从定时器 1 获得输入触发（TIM2_SMCR 寄存器的 TS=000）。
- 配置定时器 2 使用外部时钟模式（TIM2_SMCR 寄存器的 SMS=111）。
- 置 TIM1_CR2 寄存器的 CEN=1 以启动定时器 2。
- 置 TIM1_CR1 寄存器的 CEN=1 以启动定时器 1。

15.3.15.5. 使用一个外部触发同步地启动 2 个定时器

这个例子中当定时器 1 的 TI1 输入上升时使能定时器 1，使能定时器 1 的同时使能定时器 2。为保证计数器的对齐，定时器 1 必须配置为主/从模式（对应 TI1 为从，对应定时器 2 为主）：

- 配置定时器 1 为主模式，送出它的使能做为触发输出（TIM1_CR2 寄存器的 MMS='001'）。
- 配置定时器 1 为从模式，从 TI1 获得输入触发（TIM1_SMCR 寄存器的 TS='100'）。
- 配置定时器 1 为触发模式（TIM1_SMCR 寄存器的 SMS='110'）。
- 配置定时器 1 为主/从模式，TIM1_SMCR 寄存器的 MSM='1'。
- 配置定时器 2 从定时器 1 获得输入触发（TIM2_SMCR 寄存器的 TS=000）
- 配置定时器 2 为触发模式（TIM2_SMCR 寄存器的 SMS='110'）。

当定时器 1 的 TI1 上出现一个上升沿时，两个定时器同步地按照内部时钟开始计数，两个 TIF 标志也同时被设置。

注：在这个例子中，在启动之前两个定时器都被初始化（设置相应的 UG 位），两个计数器都从 0 开始，但可以通过写入任意一个计数器寄存器（TIM2_CNT）在定时器间插入一个偏移。下图中能看到主/从模式下在定时器 1 的 CNT_EN 和 CK_PSC 之间有个延迟。

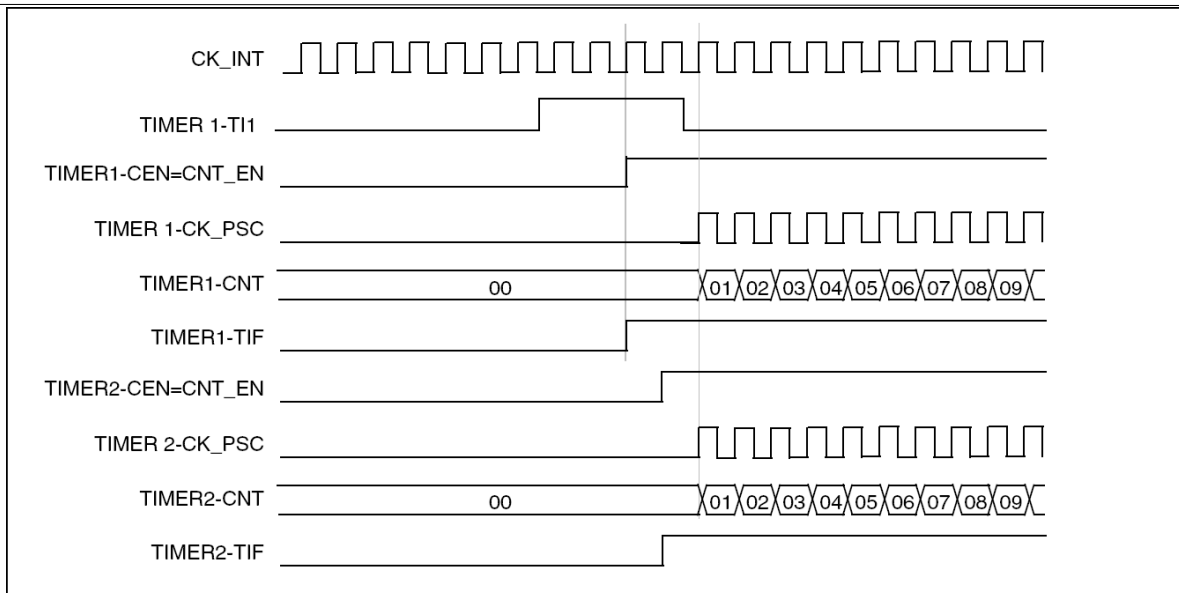


图58. 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 2

15.3.16. 调试模式

当微控制器进入调试模式（Cortex-M0 核心停止），根据 DBG 模块中 DBG_TIM2_STOP 的设置，TIM2 计数器或者继续正常操作，或者停止。

15.4. TIM2 寄存器描述

Timer1 的基地址为 0x40012C00

Timer2 的基地址为 0x40000000

下表详述了 Timer2 的寄存器

Register Name: TIM2_CR1, Address: 0x40000000, Default: 0x0

表98. 控制寄存器 1

Bit Name	Position	Description	Default	Access
CEN	[0]	使能计数器, 注: 在软件设置了 CEN 位后, 外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。 在单脉冲模式下, 当发生更新事件时, CEN 被自动清除。 0: 禁止计数器; 1: 使能计数器	0x0	RW
UDIS	[1]	禁止更新 (Update disable) 软件通过该位允许/禁止 UEV 事件的产生 0: 允许 UEV。更新(UEV)事件由下述任一事件产生: -计数器溢出/下溢 -设置 UG 位 -从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。(更新影子寄存器) 1: 禁止 UEV。不产生更新事件, 影子寄存器(ARR、PSC、CCRx)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位, 则计数器和预分频器被重新初始化	0x0	RW
URS	[2]	更新请求源 (Update request source) 软件通过该位选择 UEV 事件的源 0: 如果使能了更新中断或 DMA 请求, 则下述任一事件产生更新中断或 DMA 请求: -计数器溢出/下溢 -设置 UG 位 -从模式控制器产生的更新 1: 如果使能了更新中断或 DMA 请求, 则只有计数器溢出/下溢才产生更新中断或 DMA 请求。	0x0	RW
OPM	[3]	单脉冲模式 (One pulse mode) 0: 在发生更新事件时, 计数器不停止; 1: 在发生下一次更新事件(清除 CEN 位)时, 计数器停止。	0x0	RW
DIR	[4]	方向 (Direction), 注: 当计数器配置为中央对齐模式或编码器模式时, 该位为只读。 0: 计数器向上计数; 1: 计数器向下计数。	0x0	RW
CMS	[5:6]	选择中央对齐模式, 注: 在计数器开启时(CEN=1), 不允许从边沿对齐模式转换到中央对齐模式。 0: 边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。 1: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道(TIM2_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 只在计数器向下	0x0	RW

Bit Name	Position	Description	Default	Access
		计数时被设置。 2: 中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道 (TIM2_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 只在计数器向上计数时被设置。 3: 中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道 (TIM2_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 在计数器向上和向下计数时均被设置。		
ARPE	[7]	自动重载预装载允许位 (Auto-reload preload enable) 0: TIM2_ARR 寄存器没有缓冲; 1: TIM2_ARR 寄存器被装入缓冲器。	0x0	RW
CKD	[8:9]	时钟分频因子 (Clock division) 定义在定时器时钟(CK_INT)频率与数字滤波器(ETR, Tlx)使用的采样频率之间的分频比例。 0: $tDTS = tCK_INT$ 1: $tDTS = 2 \times tCK_INT$ 2: $tDTS = 4 \times tCK_INT$	0x0	RW

Register Name: TIM2_CR2, Address: 0x40000004, Default: 0x0

表99. 控制寄存器 2

Bit Name	Position	Description	Default	Access
CCDS	[3]	捕获/比较的 DMA 选择 0: 当发生 CCx 事件时, 送出 CCx 的 DMA 请求; 1: 当发生更新事件时, 送出 CCx 的 DMA 请求。	0x0	RW
MMS	[4:6]	主模式选择 (Master mode selection) 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO) 0: 复位 – TIM2_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式), 则 TRGO 上的信号相对实际的复位会有一个延迟。 1: 使能 – 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。 当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟, 除非选择了主/从模式(见 TIM2_SMCR 寄存器中 MSM 位的描述)。 2: 更新 – 更新事件被选为触发输入(TRGO)。例如, 一个主定时器的时钟可以被用作一个从定时器的预分频器 3: 比较脉冲 – 在发生一次捕获或一次比较成功时, 当要设置 CC1IF 标志时(即使它已经为高), 触发输出送出一个正脉冲(TRGO)。 4: 比较 – OC1REF 信号被用于作为触发输出(TRGO)。 5: 比较 – OC2REF 信号被用于作为触发输出(TRGO)。 6: 比较 – OC3REF 信号被用于作为触发输出(TRGO)。 7: 比较 – OC4REF 信号被用于作为触发输出(TRGO)。	0x0	RW
TI1S	[7]	TI1 选 择 0: TIM2_CH1 引脚连到 TI1 输入 1: TIM2_CH1、TIM2_CH2 和 TIM2_CH3 引脚经异或后连到 TI1 输入。	0x0	RW

Register Name: TIM2_SMCR, Address: 0x40000008, Default: 0x0

表100. 从模式控制寄存器

Bit Name	Position	Description	Default	Access
SMS	[0:2]	从模式选择 (Slave mode selection) 当选择了外部信号, 触发信号(TRGI)的有效边沿与选中的外部输入极性相关 (见输入控制寄存器和控制寄存器的说明) 注: 如果 TI1F_EN 被选为触发输入(TS=100)时, 不要使用门控模式。这是因为, TI1F_ED 在每次 TI1F 变化时输出一个脉冲, 然而门控模式是要检查触发输入的电平。 0: 关闭从模式 – 如果 CEN=1, 则预分频器直接由内部时钟驱动 1: 编码器模式 1 – 根据 TI1FP1 的电平, 计数器在 TI2FP2 的边沿向上/下计数 2: 编码器模式 2 – 根据 TI2FP2 的电平, 计数器在 TI1FP1 的边沿向上/下计数。 3: 编码器模式 3 – 根据另一个信号的输入电平, 计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数 4: 复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器, 并且产生一个更新寄存器的信号。 5: 门控模式 – 当触发输入(TRGI)为高时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止(但不复位)。计数器的启动和停止都是受控的。 6: 触发模式 – 计数器在触发输入 TRGI 的上升沿启动(但不复位), 只有计数器的启动是受控的。 7: 外部时钟模式 1 – 选中的触发输入(TRGI)的上升沿驱动计数器。	0x0	RW
TS	[4:6]	触发选择 (Trigger selection) 这 3 位选择用于同步计数器的触发输入。 注: 这些位只能在未用到(如 SMS=000)时被改变, 以避免在改变时产生错误的边沿检测。 0: 内部触发 0(ITR0), TIM1 1: 内部触发 1(ITR1), TIM2 2: 内部触发 2(ITR2), TIM3 3: 内部触发 3(ITR3), TIM4 4: TI1 的边沿检测器(TI1F_ED) 5: 滤波后的定时器输入 1(TI1FP1) 6: 滤波后的定时器输入 2(TI2FP2) 7: 外部触发输入(ETRF)	0x0	RW
MSM	[7]	主/从模式 (Master/slave mode) 0: 无作用 1: 触发输入(TRGI)上的事件被延迟了, 以允许在当前定时器(通过 TRGO)与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。	0x0	RW
ETF	[8:11]	外部触发滤波 (External trigger filter) 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上, 数字滤波器是一个事件计数器, 它记录到 N 个事件后会产生一个输出的跳变 0: 无滤波器, 以 fDTS 采样 1: 采样频率 fSAMPLING=fCK_INT, N=2 2: 采样频率 fSAMPLING=fCK_INT, N=4 3: 采样频率 fSAMPLING=fCK_INT, N=8	0x0	RW

Bit Name	Position	Description	Default	Access
		4: 采样频率 fSAMPLING=fDTS/2, N=6 5: 采样频率 fSAMPLING=fDTS/2, N=8 6: 采样频率 fSAMPLING=fDTS/4, N=6 7: 采样频率 fSAMPLING=fDTS/4, N=8 8: 采样频率 fSAMPLING=fDTS/8, N=6 9: 采样频率 fSAMPLING=fDTS/8, N=8 10: 采样频率 fSAMPLING=fDTS/16, N=5 11: 采样频率 fSAMPLING=fDTS/16, N=6 12: 采样频率 fSAMPLING=fDTS/16, N=8 13: 采样频率 fSAMPLING=fDTS/32, N=5 14: 采样频率 fSAMPLING=fDTS/32, N=6 15: 采样频率 fSAMPLING=fDTS/32, N=8		
ETPS	[12:13]	外部触发预分频 (External trigger prescaler) 外部触发信号 ETRP 的频率必须最多是 CK_INT 频率的 1/4。当输入较快的外部时钟时, 可以使用预分频降低 ETRP 的频率。 0: 关闭预分频 1: ETRP 频率除以 2; 2: ETRP 频率除以 4; 3: ETRP 频率除以 8。	0x0	RW
ECE	[14]	外部时钟使能位 (External clock enable) 该位启用外部时钟模式 2,注 1: 设置 ECE 位与选择外部时钟模式 1 并将 TRGI 连到 ETRF(SMS=111 和 TS=111)具有相同功效。 注 2: 下述从模式可以与外部时钟模式 2 同时使用: 复位模式、门控模式和触发模式; 但是, 这时 TRGI 不能连到 ETRF(TS 位不能是'111')。 注 3: 外部时钟模式 1 和外部时钟模式 2 同时被使能时, 外部时钟的输入是 ETRF。 0: 禁止外部时钟模式 2; 1: 使能外部时钟模式 2。计数器由 ETRF 信号上的任意有效边沿驱动	0x0	RW
ETP	[15]	ETP: 外部触发极性 (External trigger polarity) 该位选择是用 ETR 还是 ETR 的反相来作为触发操作 0: ETR 不反相, 高电平或上升沿有效; 1: ETR 被反相, 低电平或下降沿有效。	0x0	RW

Timer1和Timer2的内部连接关系如下表。

表101.

从定时器	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM1	0	TIM2	0	0
TIM2	TIM1	0	0	0

Register Name: TIM2_DIER, Address: 0x4000000c, Default: 0x0

表102. DMA/中断使能寄存器

Bit Name	Position	Description	Default	Access
UIE	[0]	允许更新中断 (Update interrupt enable)	0x0	RW
CC1IE	[1]	允许捕获/比较 1 中断 (Capture/Compare 1 interrupt enable)	0x0	RW

CC2IE	[2]	允许捕获/比较 2 中断 (Capture/Compare 2 interrupt enable)	0x0	RW
CC3IE	[3]	允许捕获/比较 3 中断 (Capture/Compare 2 interrupt enable)	0x0	RW
CC4IE	[4]	允许捕获/比较 4 中断 (Capture/Compare 2 interrupt enable)	0x0	RW
TIE	[6]	触发中断使能 (Trigger interrupt enable)	0x0	RW
UDE	[8]	允许更新的 DMA 请求 (Update DMA request enable)	0x0	RW
CC1DE	[9]	允许捕获/比较 1 的 DMA 请求 (Capture/Compare 1 DMA request enable)	0x0	RW
CC2DE	[10]	允许捕获/比较 2 的 DMA 请求 (Capture/Compare 2 DMA request enable)	0x0	RW
CC3DE	[11]	允许捕获/比较 3 的 DMA 请求 (Capture/Compare 3 DMA request enable)	0x0	RW
CC4DE	[12]	允许捕获/比较 4 的 DMA 请求 (Capture/Compare 4 DMA request enable)	0x0	RW
TDE	[14]	允许触发 DMA 请求 (Trigger DMA request enable)	0x0	RW

Register Name: TIM2_SR, Address: 0x40000010, Default: 0x0

表103. 状态寄存器

Bit Name	Position	Description	Default	Access
UIF	[0]	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置'1'。它由软件清'0'。 -若 TIM2_CR1 寄存器的 UDIS=0、URS=0, 当 TIM2_EGR 寄存器的 UG=1 时产生更新事件 (软件对计数器 CNT 重新初始化); -若 TIM2_CR1 寄存器的 UDIS=0、URS=0, 当计数器 CNT 被触发事件重初始化时产生更新事件。	0x0	RW
CC1IF	[1]	捕获/比较 1 中断标记	0x0	RW
CC2IF	[2]	捕获/比较 2 中断标记	0x0	RW
CC3IF	[3]	捕获/比较 3 中断标记	0x0	RW
CC4IF	[4]	捕获/比较 4 中断标记	0x0	RW
TIF	[6]	触发器中断标记,仅当相应的通道被配置为输入捕获时, 该标记可由硬件置'1'。写'0'可清除该位。	0x0	RW
CC1OF	[9]	捕获/比较 1 重复捕获标记 (Capture/Compare 1 overcapture flag) 仅当相应的通道被配置为输入捕获时, 该标记可由硬件置'1'。写'0'可清除该位。	0x0	RW
CC2OF	[10]	捕获/比较 2 重复捕获标记,仅当相应的通道被配置为输入捕获时, 该标记可由硬件置'1'。写'0'可清除该位。	0x0	RW
CC3OF	[11]	捕获/比较 2 重复捕获标记,仅当相应的通道被配置为输入捕获时, 该标记可由硬件置'1'。写'0'可清除该位。	0x0	RW
CC4OF	[12]	捕获/比较 2 重复捕获标记,仅当相应的通道被配置为输入捕获时, 该标记可由硬件置'1'。写'0'可清除该位。	0x0	RW

Register Name: TIM2_EGR, Address: 0x40000014, Default: 0x0

表104. 事件产生寄存器

Bit Name	Position	Description	Default	Access
UG	[0]	产生更新事件 (Update generation) 该位由软件置'1', 由硬件自动清'0'。 0: 无动作; 1: 重新初始化计数器, 并产生一个更新事件。注意预分频器的计数器也被清'0'(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清'0', 若 DIR=1(向下计数)则计数器取 TIM2_ARR 的值。	0x0	RW
CC1G	[1]	产生捕获/比较 1 事件 (Capture/compare 1 generation) 该位由软件置'1', 用于产生一个捕获/比较事件, 由硬件自动清'0'。	0x0	RW

Bit Name	Position	Description	Default	Access
		置 CC1IF=1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA。若通道 CC1 配置为输入: 当前的计数器值捕获至 TIM2_CCR1 寄存器; 设置 CC1IF=1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA。若 CC1IF 已经为 1, 则设置 CC1OF=1。 0: 无动作; 1: 在通道 CC1 上产生一个捕获/比较事件: 若通道 CC1 配置为输出:		
CC2G	[2]	产生捕获/比较 2 事件	0x0	RW
CC3G	[3]	产生捕获/比较 3 事件	0x0	RW
CC4G	[4]	产生捕获/比较 4 事件	0x0	RW
TG	[6]	产生触发事件 (Trigger generation) 该位由软件置'1', 用于产生一个触发事件, 由硬件自动清'0'。 0: 无动作; 1: TIM2_SR 寄存器的 TIF=1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA。	0x0	RW

Register Name: TIM2_CCMR1, Address: 0x40000018, Default: 0x0

通道可用于输入(捕获模式)或输出(比较模式), 通道的方向由相应的 CCxS 定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能, ICxx 描述了通道在输出模式下的功能。因此必须注意, 同一个位在输出模式和输入模式下的功能是不同的。

表105. 输出比较模式:

Bit Name	Position	Description	Default	Access
CC1S	[0:1]	捕获/比较 1 选择 (Capture/Compare 1 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: CC1S 仅在通道关闭时(TIM2_CCER 寄存器的 CC1E=0)才是可写的。 0: CC1 通道被配置为输出; 1: CC1 通道被配置为输入, IC1 映射在 TI1 上; 2: CC1 通道被配置为输入, IC1 映射在 TI2 上; 3: CC1 通道被配置为输入, IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。	0x0	RW
OC1FE	[2]	输出比较 1 快速使能 (Output compare 1 fast enable) 该位用于加快 CC 输出对触发器输入事件的响应。 该位只在通道被配置成 PWM1 或 PWM2 模式时起作用。配置为输入捕获时做预分频器 0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入出现一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期。 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。	0x0	RW
OC1PE	[3]	输出比较 1 预装载使能。注 1: 一旦 LOCK 级别设为 3(TIM2_BDTR 寄存器中的 LOCK 位)并且 CC1S='00'(该通道配置成输出)则该位不能被修改。 注 2: 仅在单脉冲模式下(TIM2_CR1 寄存器的 OPM='1'), 可以在未确认预装载寄存器情况下使用 PWM 模式, 否则其动作不确定。	0x0	RW

Bit Name	Position	Description	Default	Access
		0: 禁止 TIM2_CCR1 寄存器的预装载功能, 可随时写入 TIM2_CCR1 寄存器, 并且新写入的数值立即起作用。 1: 开启 TIM2_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM2_CCR1 的预装载值在更新事件到来时被传送至当前寄存器中。		
OC1M	[4:6]	输出比较 1 模式 (Output compare 1 enable) 该 3 位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效, 而 OC1 的有效电平取决于 CC1P 位。 注 1: 一旦 LOCK 级别设为 3(TIM2_BDTR 寄存器中的 LOCK 位)并且 CC1S='00'(该通道配置成输出)则该位不能被修改。 注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。 0: 冻结。输出比较寄存器 TIM2_CCR1 与计数器 TIM2_CNT 间的比较对 OC1REF 不起作用; 1: 匹配时设置通道 1 为有效电平。当计数器 TIM2_CNT 的值与捕获 / 比较寄存器 1(TIM2_CCR1)相同时, 强制 OC1REF 为高。 2: 匹配时设置通道 1 为无效电平。当计数器 TIM2_CNT 的值与捕获 / 比较寄存器 1(TIM2_CCR1)相同时, 强制 OC1REF 为低。 3: 翻转。当 TIM2_CCR1=TIM2_CNT 时, 翻转 OC1REF 的电平。 4: 强制为无效电平。强制 OC1REF 为低。 5: 强制为有效电平。强制 OC1REF 为高。 6: PWM 模式 1 – 在向上计数时, 一旦 TIM2_CNT<TIM2_CCR1 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 TIM2_CNT>TIM2_CCR1 时通道 1 为无效电平(OC1REF=0), 否则为有效电平(OC1REF=1)。 7: PWM 模式 2 – 在向上计数时, 一旦 TIM2_CNT<TIM2_CCR1 时通道 1 为无效电平, 否则为有效电平; 在向下计数时, 一旦 TIM2_CNT>TIM2_CCR1 时通道 1 为有效电平, 否则为无效电平。	0x0	RW
OC1CE	[7]	输出比较 1 清除使能 (Output compare 1 clear enable) 0: OC1REF 不受 ETRF 输入的影响; 1: 一旦检测到 ETRF 输入高电平, 清除 OC1REF=0。	0x0	RW
CC2S	[8:9]	捕获/比较 2 选择 (Capture/Compare 2 selection) 该位定义通道的方向(输入/输出), 及输入脚的选择。注: CC2S 仅在通道关闭时(TIM2_CCER 寄存器的 CC2E='0')才是可写的。 0: CC2 通道被配置为输出; 1: CC2 通道被配置为输入, IC2 映射在 TI2 上; 2: CC2 通道被配置为输入, IC2 映射在 TI1 上; 3: CC2 通道被配置为输入, IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。	0x0	RW
OC2FE	[10]	输出比较 2 快速使能	0x0	RW
OC2PE	[11]	输出比较 2 预装载使能	0x0	RW
OC2M	[12:14]	输出比较 2 模式	0x0	RW
OC2CE	[15]	输出比较 2 清除使能	0x0	RW

表106. 输入捕获模式：

Bit Name	Position	Description	Default	Access
CC1S	[0:1]	捕获/比较 1 选择 (Capture/Compare 1 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1 通道被配置为输出; 01: CC1 通道被配置为输入, IC1 映射在 TI1 上; 10: CC1 通道被配置为输入, IC1 映射在 TI2 上; 11: CC1 通道被配置为输入, IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC1S 仅在通道关闭时(TIM2_CCER 寄存器的 CC1E='0')才是可写的。	0x0	RW
IC1PSC	[2:3]	输入/捕获 1 预分频器 (Input capture 1 prescaler) 这 2 位定义了 CC1 输入(IC1)的预分频系数。 一旦 CC1E='0'(TIM2_CCER 寄存器中), 则预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获; 01: 每 2 个事件触发一次捕获; 10: 每 4 个事件触发一次捕获; 11: 每 8 个事件触发一次捕获。	0x0	RW
IC1F	[4:7]	输入捕获 1 滤波器 (Input capture 1 filter) 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成, 它记录到 N 个事件后会产生一个输出的跳变: 0000: 无滤波器, 以 fDTS 采样 1000 : 采样频率 fSAMPLING=fDTS/8, N=6 0001: 采样频率 fSAMPLING=fCK_INT, N=2 1001 : 采样频率 fSAMPLING=fDTS/8, N=8 0010: 采样频率 fSAMPLING=fCK_INT, N=4 1010 : 采样频率 fSAMPLING=fDTS/16, N=5 0011: 采样频率 fSAMPLING=fCK_INT, N=8 1011 : 采样频率 fSAMPLING=fDTS/16, N=6 0100: 采样频率 fSAMPLING=fDTS/2, N=6 1100 : 采样频率 fSAMPLING=fDTS/16, N=8 0101: 采样频率 fSAMPLING=fDTS/2, N=8 1101 : 采样频率 fSAMPLING=fDTS/32, N=5 0110: 采样频率 fSAMPLING=fDTS/4, N=6 1110 : 采样频率 fSAMPLING=fDTS/32, N=6 0111: 采样频率 fSAMPLING=fDTS/4, N=8 1111 : 采样频率 fSAMPLING=fDTS/32, N=8 注: 在现在的芯片版本中, 当 ICx_F[3:0]=1、2 或 3 时, 公式中的 fDTS 由 CK_INT 替代。	0x0	RW
CC2S	[8:9]	捕获/比较 2 选择 (Capture/compare 2 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC2 通道被配置为输出; 01: CC2 通道被配置为输入, IC2 映射在 TI2 上; 10: CC2 通道被配置为输入, IC2 映射在 TI1 上; 11: CC2 通道被配置为输入, IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIM2_CCER 寄存器的 CC2E='0')才是可写的。	0x0	RW
IC2PSC	[10:11]	输入/捕获 2 预分频器 (input capture 2 prescaler)	0x0	RW

Bit Name	Position	Description	Default	Access
IC2F	[15:12]	输入捕获 2 滤波器 (Input capture 2 filter)	0x0	RW

Register Name: TIM2_CCMR2, Address: 0x4000001c, Default: 0x0

捕获/比较模式寄存器 2(输出比较模式,通道的方向由相应的 CCxS 定义), 参看以上 CCMR1 寄存器的描述。

表107. 输出比较模式:

Bit Name	Position	Description	Default	Access
CC3S	[0:1]	捕获/比较 3 选择 (Capture/Compare 3 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 0: CC3 通道被配置为输出; 1: CC3 通道被配置为输入, IC1 映射在 TI1 上; 2: CC3 通道被配置为输入, IC1 映射在 TI2 上; 3: CC3 通道被配置为输入, IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。	0x0	RW
OC3FE	[2]	输出比较 3 快速使能	0x0	RW
OC3PE	[3]	输出比较 3 预装载使能	0x0	RW
OC3M	[4:6]	输出比较 3 模式	0x0	RW
OC3CE	[7]	输出比较 3 清 0 使能	0x0	RW
CC4S	[8:9]	捕获/比较 4 选择 (Capture/Compare 4 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 0: CC4 通道被配置为输出; 1: CC4 通道被配置为输入, IC4 映射在 TI4 上; 2: CC4 通道被配置为输入, IC4 映射在 TI3 上; 3: CC4 通道被配置为输入, IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。	0x0	RW
OC4FE	[10]	输出比较 4 快速使能	0x0	RW
OC4PE	[11]	输出比较 4 预装载使能	0x0	RW
OC4M	[12:14]	输出比较 4 模式	0x0	RW
OC4CE	[15]	输出比较 4 清 0 使能	0x0	RW

表108. 输入捕获模式:

Bit Name	Position	Description	Default	Access
CC3S	[0:1]	捕获/比较 3 选择 (Capture/Compare 3 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC3 通道被配置为输出; 01: CC3 通道被配置为输入, IC3 映射在 TI3 上; 10: CC3 通道被配置为输入, IC3 映射在 TI4 上; 11: CC3 通道被配置为输入, IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC3S 仅在通道关闭时(TIM2_CCER 寄存器的 CC3E='0')才是可写的。	0x0	RW
IC3PSC	[2:3]	输入/捕获 3 预分频器 (Input capture 3 prescaler)	0x0	RW
IC3F	[4:7]	输入捕获 3 滤波器 (Input capture 3 filter)	0x0	RW
CC4S	[8:9]	捕获/比较 4 选择 (Capture/compare 4 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC4 通道被配置为输出; 01: CC4 通道被配置为输入, IC4 映射在 TI4 上; 10: CC4 通道被配置为输入, IC4 映射在 TI3 上;	0x0	RW

Bit Name	Position	Description	Default	Access
		11: CC4 通道被配置为输入, IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIM2_CCER 寄存器的 CC4E='0')才是可写的。		
IC4PSC	[10:11]	输入/捕获 4 预分频器 (input capture 4 prescaler)	0x0	RW
IC4F	[12:15]	输入捕获 4 滤波器 (Input capture 4 filter)	0x0	RW

Register Name: TIM2_CCER, Address: 0x40000020, Default: 0x0

表109. 捕获/比较使能寄存器

Bit Name	Position	Description	Default	Access
CC1E	[0]	输入/捕获 1 输出使能 (Capture/Compare 1 output enable) CC1 通道配置为输出: 0: 关闭 - OC1 禁止输出。 1: 开启 - OC1 信号输出到对应的输出引脚。 CC1 通道配置为输入: 该位决定了计数器的值是否能捕获入 TIM2_CCR1 寄存器。 0: 捕获禁止; 1: 捕获使能。	0x0	RW
CC1P	[1]	输入/捕获 1 输出极性 (Capture/Compare 1 output polarity) CC1 通道配置为输出: 0: OC1 高电平有效 1: OC1 低电平有效 CC1 通道配置为输入: 该位选择是 IC1 还是 IC1 的反相信号作为触发或捕获信号。 0: 不反相: 捕获发生在 IC1 的上升沿; 当用作外部触发器时, IC1 不反相。 1: 反相: 捕获发生在 IC1 的下降沿; 当用作外部触发器时, IC1 反相。	0x0	RW
CC2E	[4]	输入/捕获 2 输出使能	0x0	RW
CC2P	[5]	输入/捕获 2 输出极性	0x0	RW
CC3E	[8]	输入/捕获 3 输出使能	0x0	RW
CC3P	[9]	输入/捕获 3 输出极性	0x0	RW
CC4E	[12]	输入/捕获 4 输出使能	0x0	RW
CC4P	[13]	输入/捕获 4 输出极性	0x0	RW

表110. 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: 连接到标准OCx通道的外部I/O引脚状态, 取决于OCx通道状态和GPIO以及AFIO寄存器。

Register Name: TIM2_CNT, Address: 0x40000024, Default: 0x0

表111. 计数器

Bit Name	Position	Description	Default	Access
CNT	[0:15]	计数器的值	0x0	RW

Register Name: TIM2_PSC, Address: 0x40000028, Default: 0x0

表112. 预分频器

Bit Name	Position	Description	Default	Access
PSC	[0:15]	预分频器的值 (Prescaler value) 计数器的时钟频率 CK_CNT 等于 fCK_PSC/(PSC[15:0]+1)。 PSC 包含了当更新事件产生时装入当前预分频器寄存器的值。	0x0	RW

Register Name: TIM2_ARR, Address: 0x4000002c, Default: 0x0

表113. 自动重装载寄存器

Bit Name	Position	Description	Default	Access
ARR	[0:15]	自动重装载的值 (Auto reload value) ARR 包含了将要传送到实际的自动重装载寄存器的数值。详细参考 14.3.1 节: 有关 ARR 的更新和动作。 当自动重装载的值为空时, 计数器不工作。	0x0	RW

Register Name: TIM2_CCR1, Address: 0x40000034, Default: 0x0

表114. 捕获/比较寄存器 1

Bit Name	Position	Description	Default	Access
CCR1	[0:15]	捕获/比较 1 的值 (Capture/Compare 1 value) 若 CC1 通道配置为输出: CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。 如果在 TIM2_CCMR1 寄存器(OC1PE 位)中未选择预装载特性, 写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 1 寄存器中。 当前捕获/比较寄存器参与同计数器 TIM2_CNT 的比较, 并在 OC1 端口上产生输出信号。若 CC1 通道配置为输入: CCR1 包含了由上一次输入捕获 1 事件(IC1)传输的计数器值。	0x0	RW

Register Name: TIM2_CCR2, Address: 0x40000038, Default: 0x0

表115. 捕获/比较寄存器 2

Bit Name	Position	Description	Default	Access
CCR2	[0:15]	捕获/比较 2 的值 (Capture/Compare 2 value) 若 CC2 通道配置为输出: CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。 如果在 TIM2_CCMR2 寄存器(OC2PE 位)中未选择预装载特性, 写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 2 寄存器中。 当前捕获/比较寄存器参与同计数器 TIM2_CNT 的比较, 并在 OC2 端口上产生输出信号。若 CC2 通道配置为输入: CCR2 包含了由上一次输入捕获 2 事件(IC2)传输的计数器值。	0x0	RW

Register Name: TIM2_CCR3, Address: 0x4000003c, Default: 0x0

表116. 捕获/比较寄存器 3

Bit Name	Position	Description	Default	Access
CCR3	[0:15]	捕获/比较 3 的值 (Capture/Compare 3 value) 若 CC3 通道配置为输出: CCR3 包含了装入当前捕获/比较 3 寄存器的值(预装载值)。 如果在 TIM2_CCMR3 寄存器(OC3PE 位)中未选择预装载特性, 写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传	0x0	RW

		输至当前捕获/比较 3 寄存器中。 当前捕获/比较寄存器参与同计数器 TIM2_CNT 的比较, 并在 OC3 端口上产生输出信号。若 CC3 通道配置为输入: CCR3 包含了由上一次输入捕获 3 事件(IC3)传输的计数器值。		
--	--	---	--	--

Register Name: TIM2_CCR4, Address: 0x40000040, Default: 0x0

表117. 捕获/比较寄存器 4

Bit Name	Position	Description	Default	Access
CCR4	[0:15]	捕获/比较 4 的值 (Capture/Compare 4 value) 若 CC4 通道配置为输出: CCR4 包含了装入当前捕获/比较 4 寄存器的值(预装载值)。 如果在 TIM2_CCMR4 寄存器(OC4PE 位)中未选择预装载特性, 写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 4 寄存器中。 当前捕获/比较寄存器参与同计数器 TIM2_CNT 的比较, 并在 OC4 端口上产生输出信号。若 CC4 通道配置为输入: CCR4 包含了由上一次输入捕获 4 事件(IC4)传输的计数器值。	0x0	RW

Register Name: TIM2_DCR, Address: 0x40000048, Default: 0x0

表118. DMA 控制寄存器

Bit Name	Position	Description	Default	Access
DBA	[0:4]	DMA 基地址 (DMA base address) 这些位定义了 DMA 在连续模式下的基地址(当对 TIM2_DMAR 寄存器进行读或写时), DBA 定义为从 TIM2_CR1 寄存器所在地址开始的偏移量: 00000: TIM2_CR1, 00001: TIM2_CR2, 00010: TIM2_SMCR,	0x0	RW
DBL	[8:12]	DMA 连续传送长度 (DMA burst length) 这些位定义了 DMA 在连续模式下的传送长度(当对 TIM2_DMAR 寄存器进行读或写时, 定时器则进行一次连续传送), 即: 定义传输的字节数目: 00000: 1 个字节 00001: 2 个字节 00010: 3 个字节	0x0	RW

Register Name: TIM2_DMAR, Address: 0x4000004c, Default: 0x0

表119. 连续模式的 DMA 地址

Bit Name	Position	Description	Default	Access
DMAB	[0:15]	DMA 连续传送寄存器 (DMA register for burst accesses) 对 TIM2_DMAR 寄存器的读或写会导致对以下地址所在寄存器的存取操作: TIM2_CR1 地址 + DBA + DMA 索引, 其中: "TIM2_CR1 地址"是控制寄存器 1(TIM2_CR1)所在的地址; "DBA"是 TIM2_DCR 寄存器中定义的基地址; "DMA 索引"是由 DMA 自动控制的偏移量, 它取决于 TIM2_DCR 寄存器中定义的 DBL。	0x0	RW

16. 高级控制定时器 (TIM1)

16.1. TIM1 简介

高级控制定时器(TIM1)由一个 16 位的自动装载计数器组成, 它由一个可编程的预分频器驱动。它适合多种用途, 包含测量输入信号的脉冲宽度(输入捕获), 或者产生输出波形(输出比较、

PWM、嵌入死区时间的互补 PWM 等)。使用定时器预分频器和 RCC 时钟控制预分频器, 可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

高级控制定时器(TIM1)和通用定时器(TIM2)是完全独立的, 它们不共享任何资源。它们可以同步操作。

16.2. TIM1 主要特性

TIM1 定时器的功能包括:

- 16 位向上、向下、向上/下自动装载计数器
- 16 位可编程(可以实时修改)预分频器, 计数器时钟频率的分频系数为 1 ~ 65535 之间的任意数值
- 多达 4 个独立通道:
 - 输入捕获
 - 输出比较
 - PWM 生成(边缘或中间对齐模式)
 - 单脉冲模式输出
- 死区时间可编程的互补输出
- 使用外部信号控制定时器和定时器互联的同步电路
- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态
- 如下事件发生时产生中断/DMA:
 - 更新: 计数器向上溢出/向下溢出, 计数器初始化(通过软件或者内部/外部触发)
 - 触发事件(计数器启动、停止、初始化或者由内部/外部触发计数)
 - 输入捕获
 - 输出比较
 - 刹车信号输入
- 支持针对定位的增量(正交)编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理

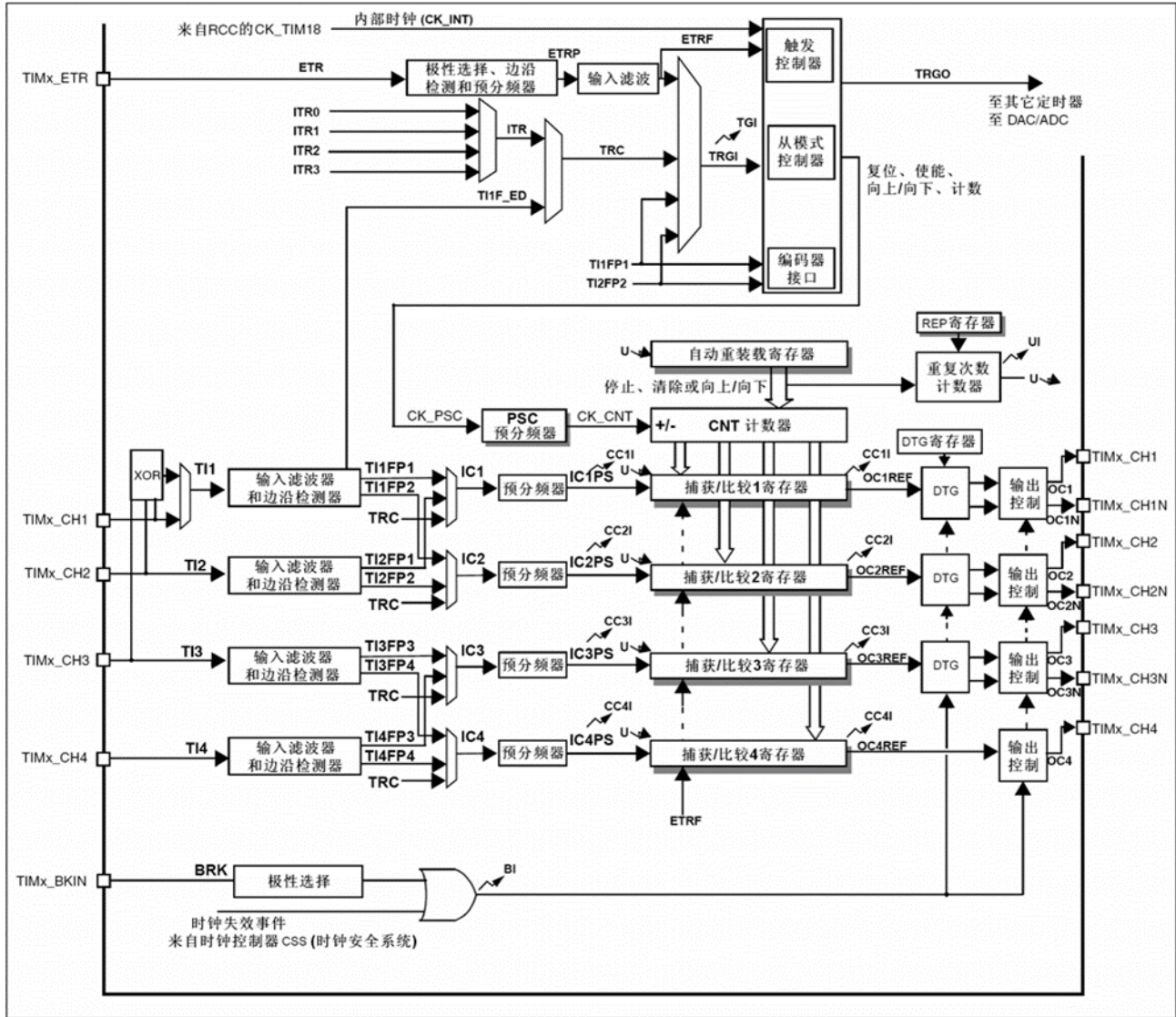


图59. 高级控制定时器框图

16.3. TIM1 功能描述

16.3.1. 时基单元

可编程高级控制定时器的主要部分是一个 16 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数、向下计数或者向上向下双向计数。此计数器时钟由预分频器分频得到。计数器、自动装载寄存器和预分频器寄存器可以由软件读写，即使计数器还在运行读写仍然有效。

时基单元包含：

- 计数器寄存器 (TIM1_CNT)
- 预分频器寄存器 (TIM1_PSC)
- 自动装载寄存器 (TIM1_ARR)
- 重复次数寄存器 (TIM1_RCR)

自动装载寄存器是预先装载的，写或读自动重载寄存器将访问预装载寄存器。根据在 TIM1_CR1 寄存器中的自动装载预装载使能位(ARPE)的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件(向下计数时的下溢条件)并当 TIM1_CR1 寄存器中的 UDIS 位等于 0 时，产生更新事件。更新事件也可以由软件产生。随后会详细描述每一种配置下更新事件的产生。

计数器由预分频器的时钟输出 CK_CNT 驱动，仅当设置了计数器 TIM1_CR1 寄存器中的计数器使能位(CEN)时，CK_CNT 才有效。(更多有关使能计数器的细节，请参见控制器的从模式描述)。

注意，在设置了 TIM1_CR 寄存器的 CEN 位的一个时钟周期后，计数器开始计数。

16.3.2. 预分频器描述

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。它是基于一个(在 TIM1_PSC 寄存器中的)16 位寄存器控制的 16 位计数器。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。下图给出了在预分频器运行时，更改计数器参数的例子。

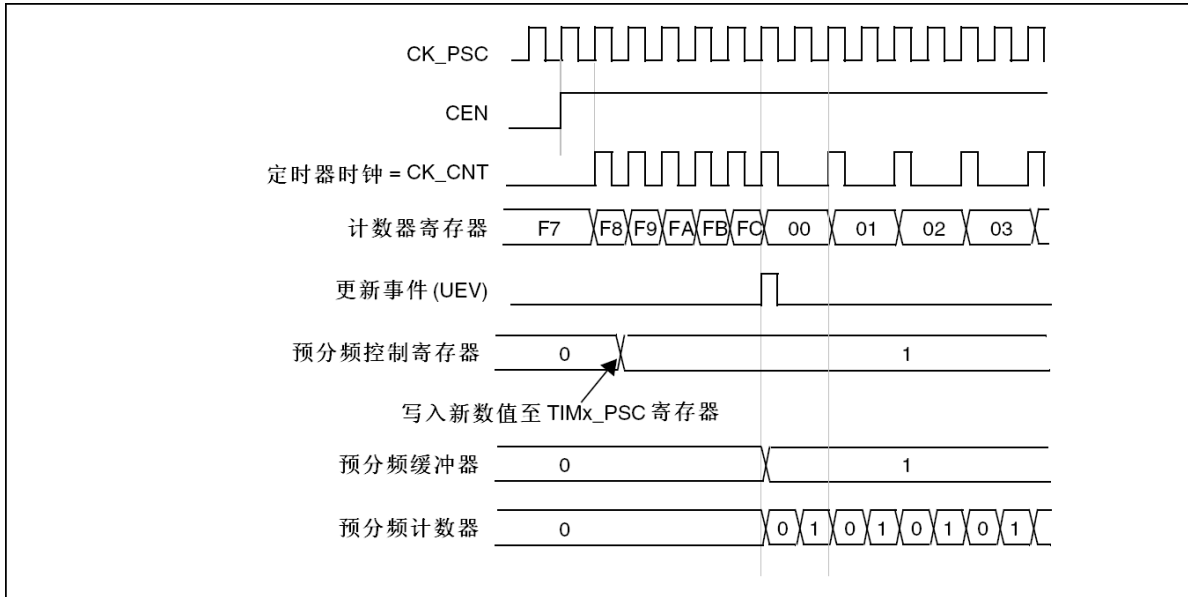


图60. 当预分频器的参数从 1 变到 2 时，计数器的时序图

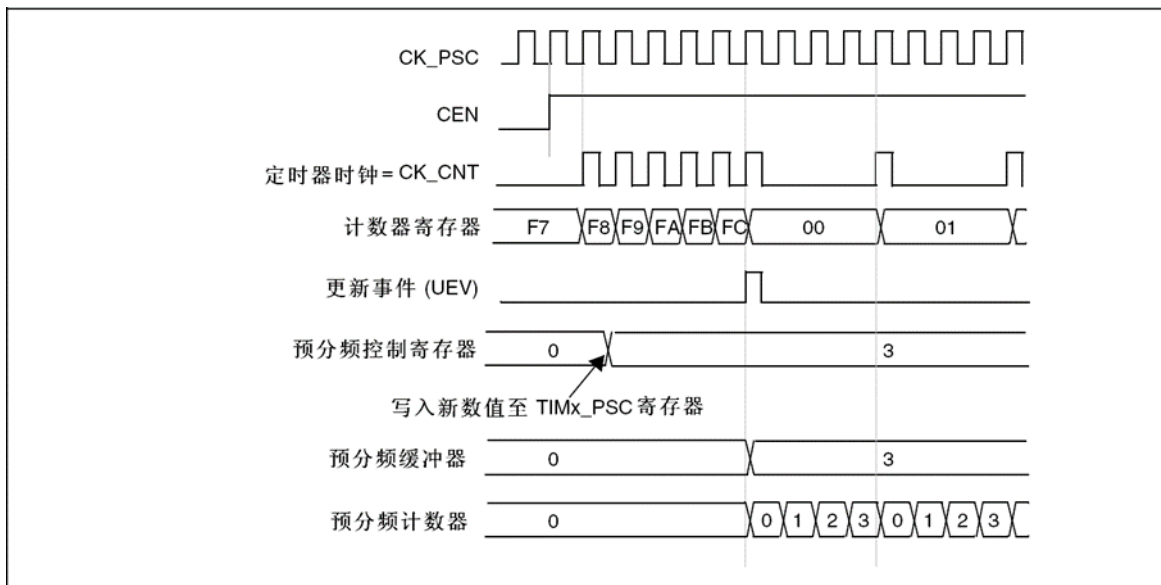


图61. 当预分频器的参数从 1 变到 4 时，计数器的时序图

16.3.3. 计数器模式

向上计数模式

在向上计数模式中，计数器从 0 计数到自动加载值(TIM1_ARR 计数器的内容)，然后重新从 0 开始计数并且

产生一个计数器溢出事件。

如果使用了重复计数器功能，在向上计数达到设置的重复计数次数(TIM1_RCR)时，产生更新事件(UEV)；否则每次计数器溢出时才产生更新事件。

在 TIM1_EGR 寄存器中(通过软件方式或者使用从模式控制器)设置 UG 位也同样可以产生一个更新事件。

设置 TIM1_CR1 寄存器中的 UDIS 位，可以禁止更新事件；这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。在 UDIS 位被清'0'之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清'0'，同时预分频器的计数也被清 0(但预分频器的数值不变)。此外，如果设置了 TIM1_CR1 寄存器中的 URS 位(选择更新请求)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志(即不产生中断或 DMA 请求)。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时(依据 URS 位)设置更新标志位(TIM1_SR 寄存器中的 UIF 位)。

- 重复计数器被重新加载为 TIM1_RCR 寄存器的内容。
- 自动装载影子寄存器被重新置入预装载寄存器的值(TIM1_ARR)。
- 预分频器的缓冲区被置入预装载寄存器的值(TIM1_PSC 寄存器的内容)。下图给出一些例子，当

TIM1_ARR=0x36 时计数器在不同时钟频率下的动作。

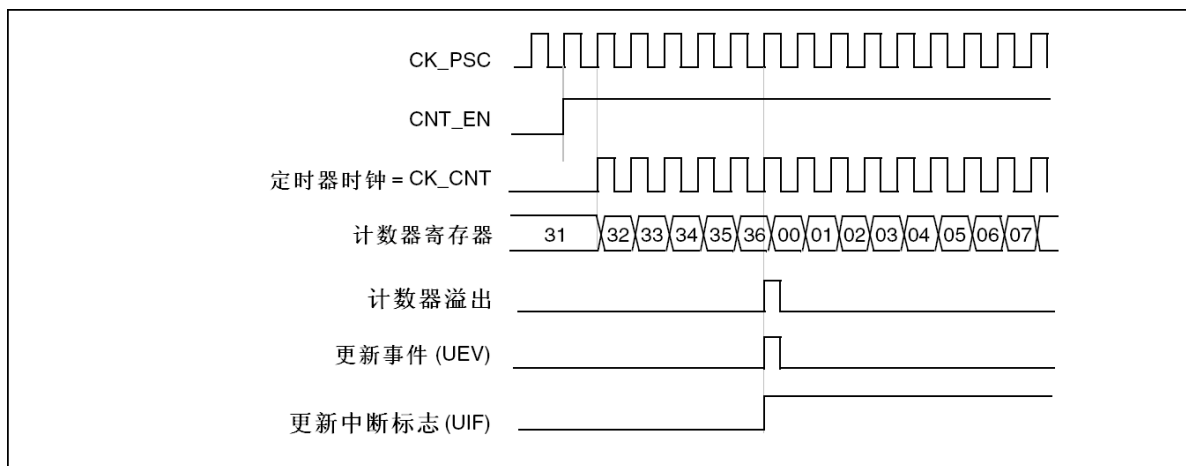


图62. 计数器时序图，内部时钟分频因子为 1

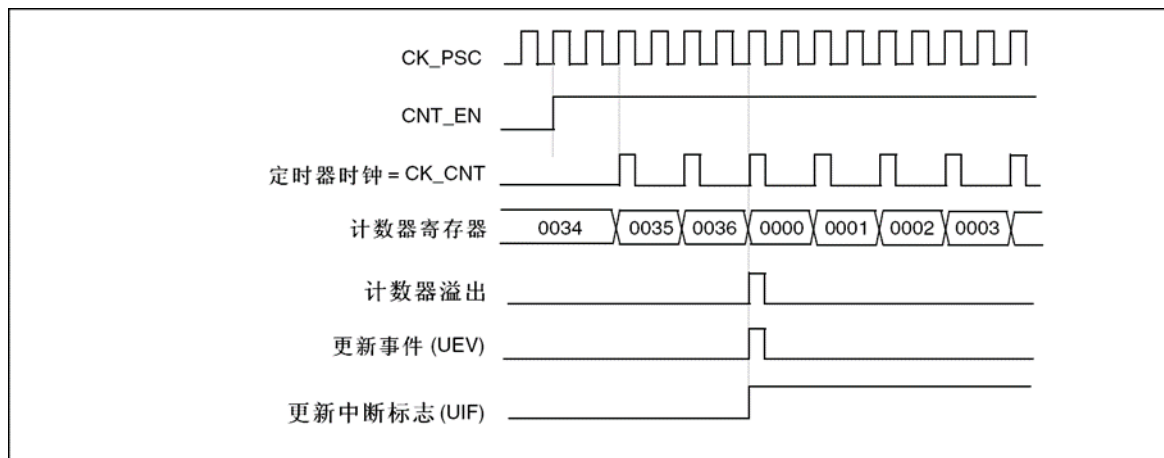


图63. 计数器时序图，内部时钟分频因子为 2

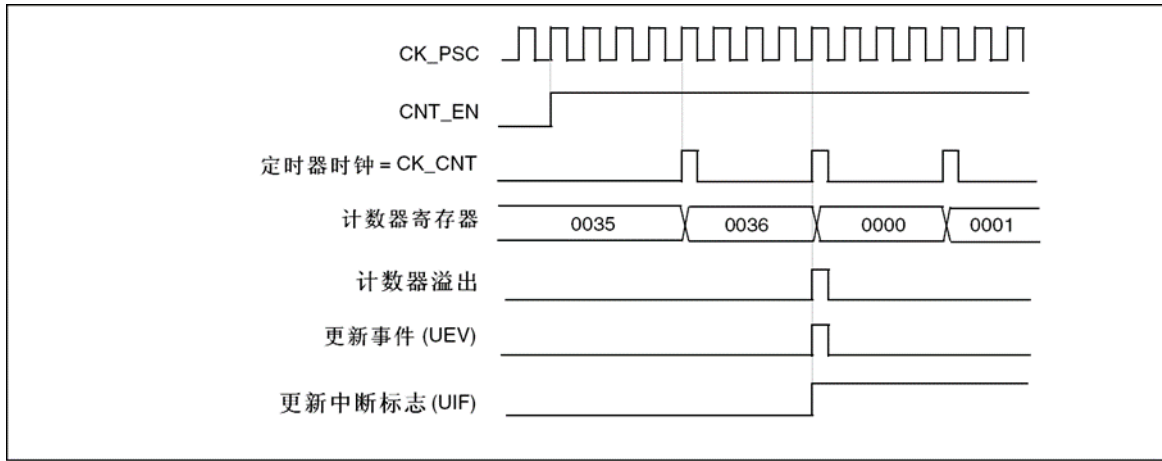


图64. 计数器时序图，内部时钟分频因子为 4

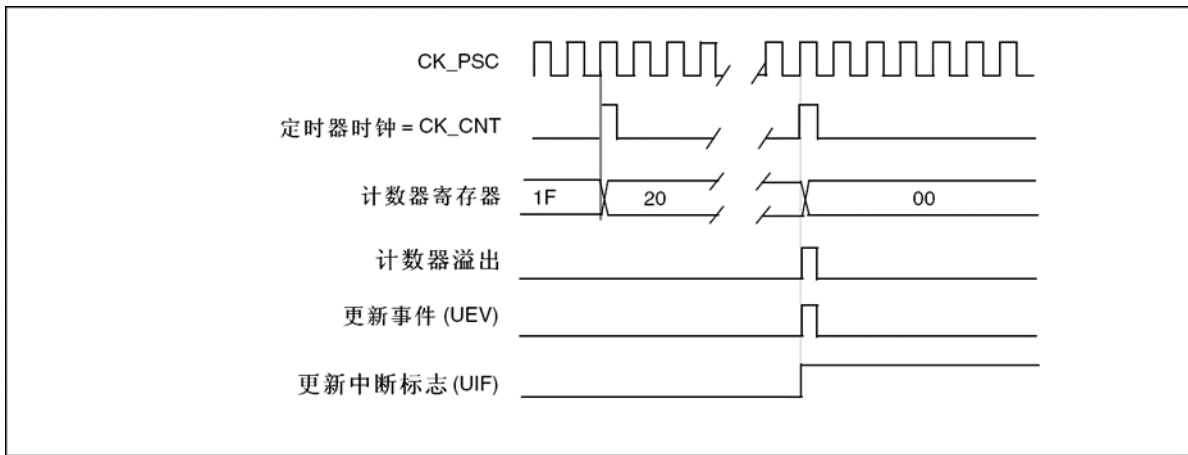


图65. 计数器时序图，内部时钟分频因子为 N

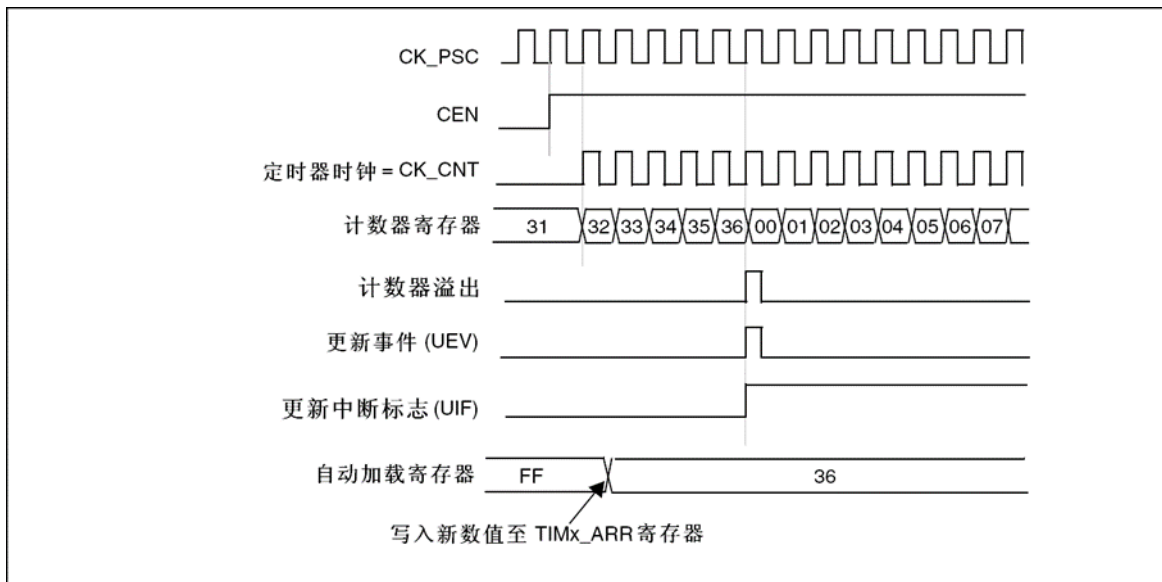


图66. 计数器时序图，当 ARPE=0 时的更新事件(TIM1_ARR 没有预装入)

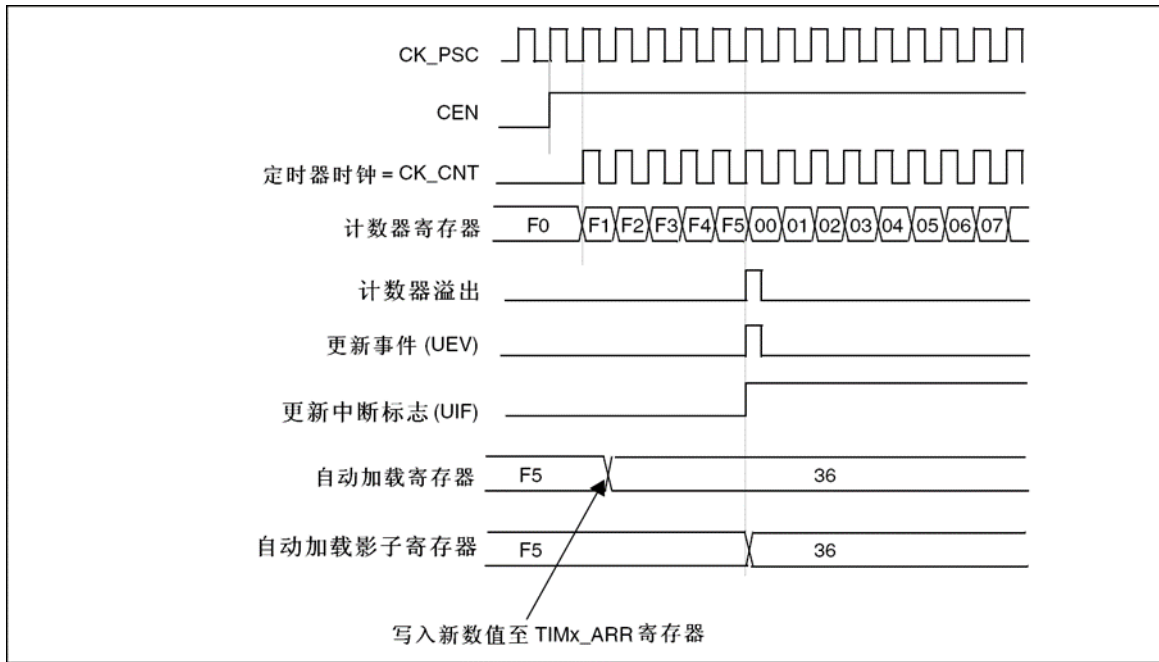


图67. 计数器时序图，当 ARPE=1 时的更新事件(预装入了 TIM1_ARR)

向下计数模式

在向下模式中，计数器从自动装入的值(TIM1_ARR 计数器的值)开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数器向下溢出事件。

如果使用了重复计数器，当向下计数重复了重复计数寄存器(TIM1_RCR)中设定的次数后，将产生更新事件(UEV)，否则每次计数器下溢时才产生更新事件。

在 TIM1_EGR 寄存器中(通过软件方式或者使用从模式控制器)设置 UG 位，也同样可以产生一个更新事件。

设置 TIM1_CR1 寄存器的 UDIS 位可以禁止 UEV 事件。这样可以避免向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，并且预分频器的计数器重新从 0 开始(但预分频系数不变)。此外，如果设置了 TIM1_CR1 寄存器中的 URS 位(选择更新请求)，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志(因此不产生中断和 DMA 请求)，这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且(根据 URS 位的设置)更新标志位(TIM1_SR 寄存器中的 UIF 位)也被设置。

- 重复计数器被重置为 TIM1_RCR 寄存器中的内容
- 预分频器的缓存器被加载为预装载的值(TIM1_PSC 寄存器的值)。
- 当前的自动加载寄存器被更新为预装载值(TIM1_ARR 寄存器中的内容)。

注：自动装载在计数器重载入之前被更新，因此下一个周期将是预期的值。

以下是一些当 TIM1_ARR=0x36 时，计数器在不同时钟频率下的操作例子。

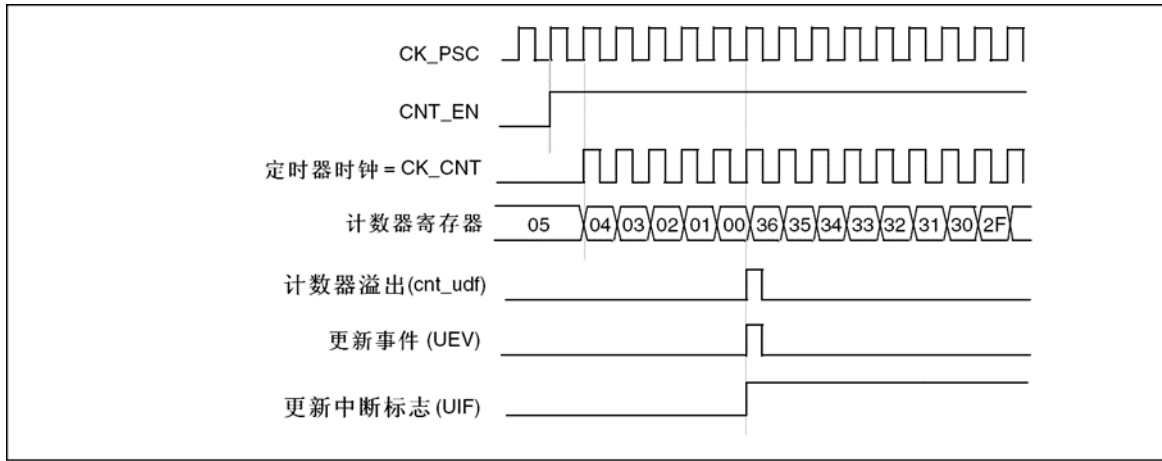


图68. 计数器时序图，内部时钟分频因子为 1

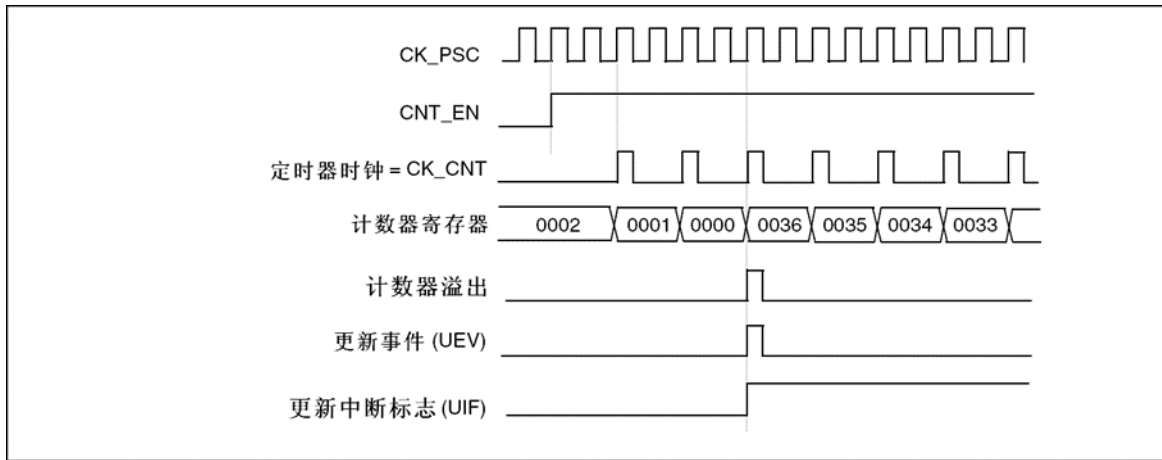


图69. 计数器时序图，内部时钟分频因子为 2

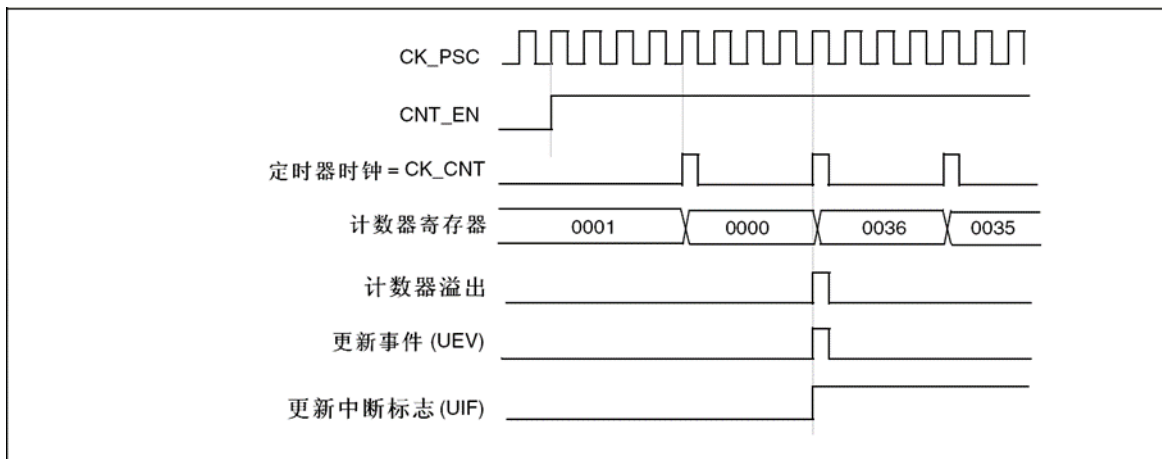


图70. 计数器时序图，内部时钟分频因子为 4

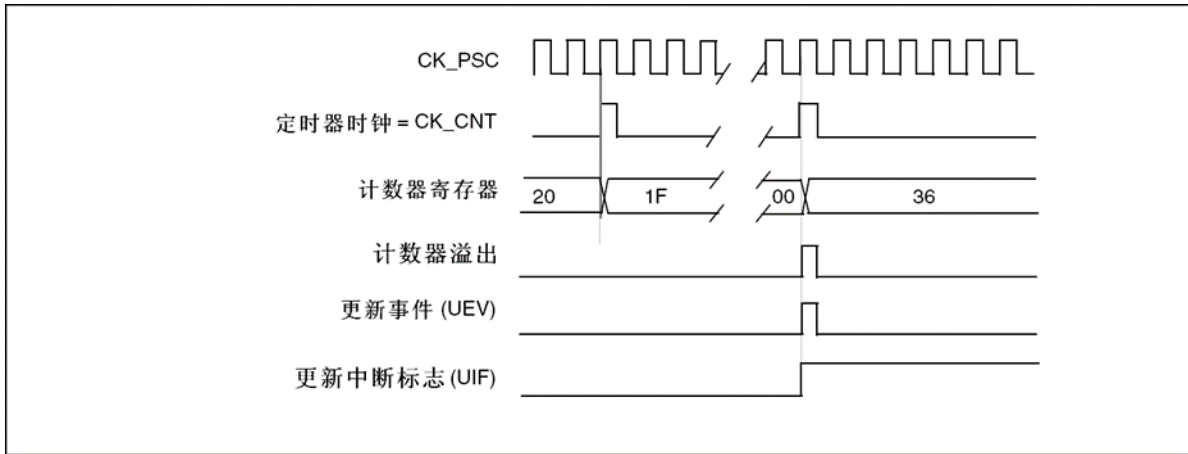


图71. 计数器时序图，内部时钟分频因子为 N

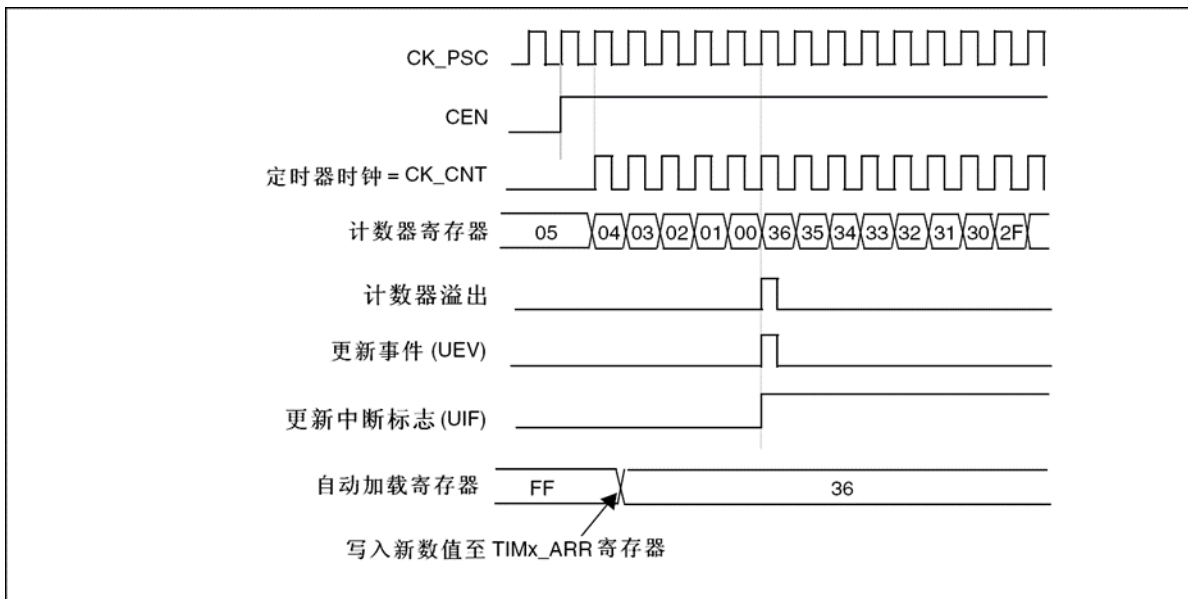


图72. 计数器时序图，当没有使用重复计数器时的更新事件

中央对齐模式(向上/向下计数)

在中央对齐模式，计数器从 0 开始计数到自动加载的值(TIM1_ARR 寄存器)-1，产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。在此模式下，不能写入 TIM1_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。可以在每次计数上溢和每次计数下溢时产生更新事件；也可以通过(软件或者使用从模式控制器) 设置 TIM1_EGR 寄存器中的 UG 位产生更新事件。然后，计数器重新从 0 开始计数，预分频器也重新从 0 开始计数。

设置 TIM1_CR1 寄存器中的 UDIS 位可以禁止 UEV 事件。这样可以避免在向预装载寄存器中写入新值时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

此外，如果设置了 TIM1_CR1 寄存器中的 URS 位(选择更新请求)，设置 UG 位将产生一个更新事件 UEV 但不设置 UIF 标志(因此不产生中断和 DMA 请求)，这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

当发生更新事件时，所有的寄存器都被更新，并且(根据 URS 位的设置)更新标志位(TIM1_SR 寄存器中的 UIF

位)也被设置。

- 重复计数器被重置为 TIM1_RCR 寄存器中的内容
- 预分频器的缓存器被加载为预装载(TIM1_PSC 寄存器)的值。
- 当前的自动加载寄存器被更新为预装载值(TIM1_ARR 寄存器中的内容)。

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

以下是一些计数器在不同时钟频率下的操作的例子

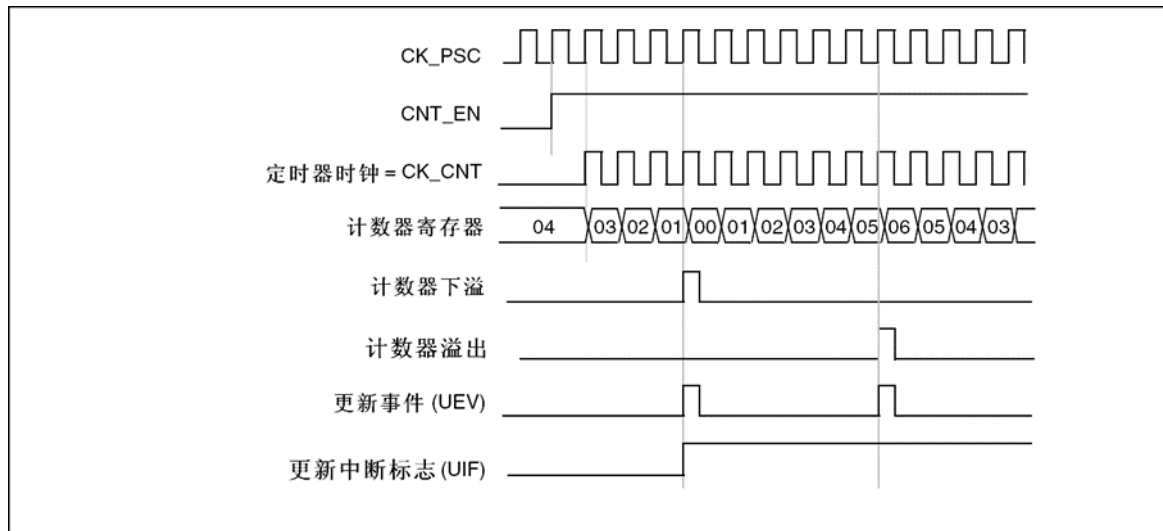


图73. 计数器时序图，内部时钟分频因子为 1，TIM1_ARR=0x6

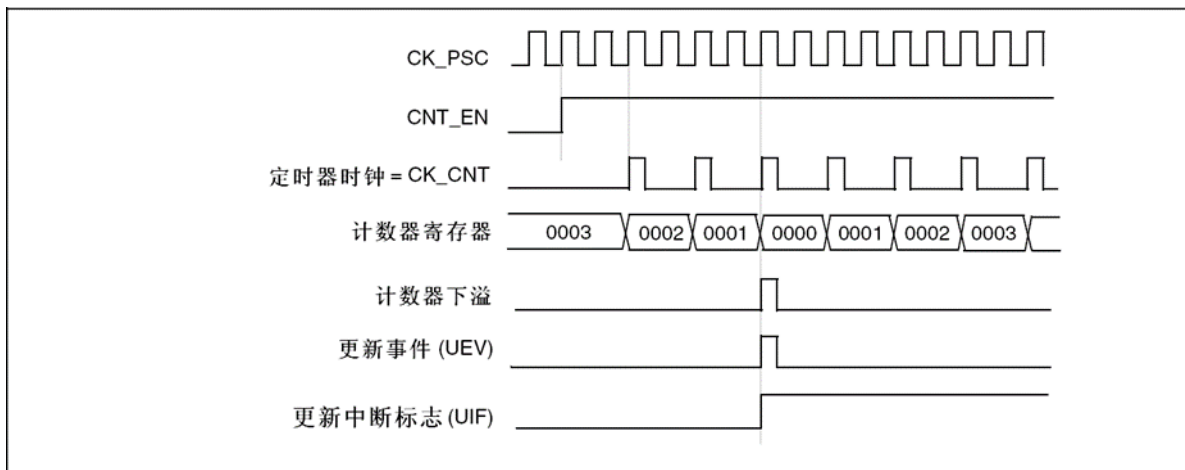


图74. 计数器时序图，内部时钟分频因子为 2

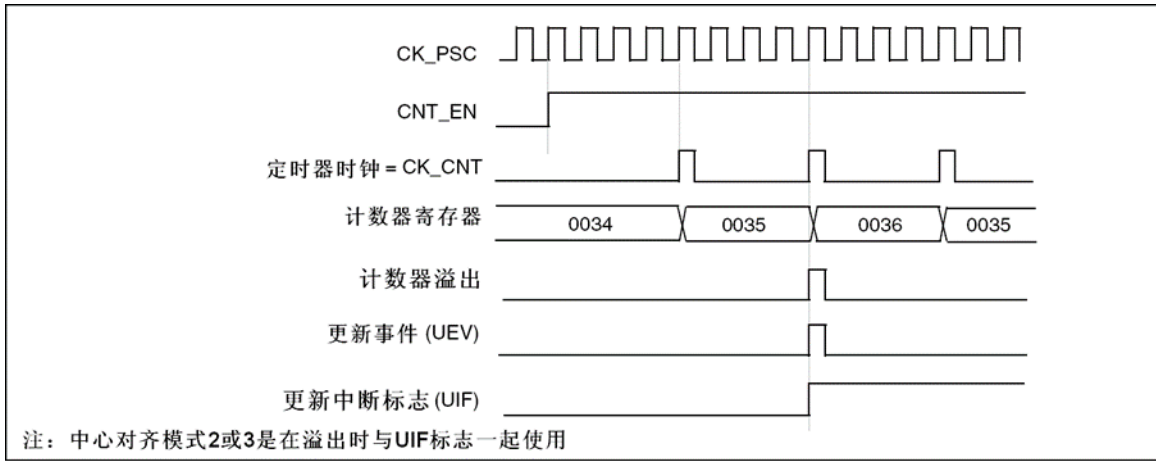


图75. 计数器时序图，内部时钟分频因子为 4，TIM1_ARR=0x36

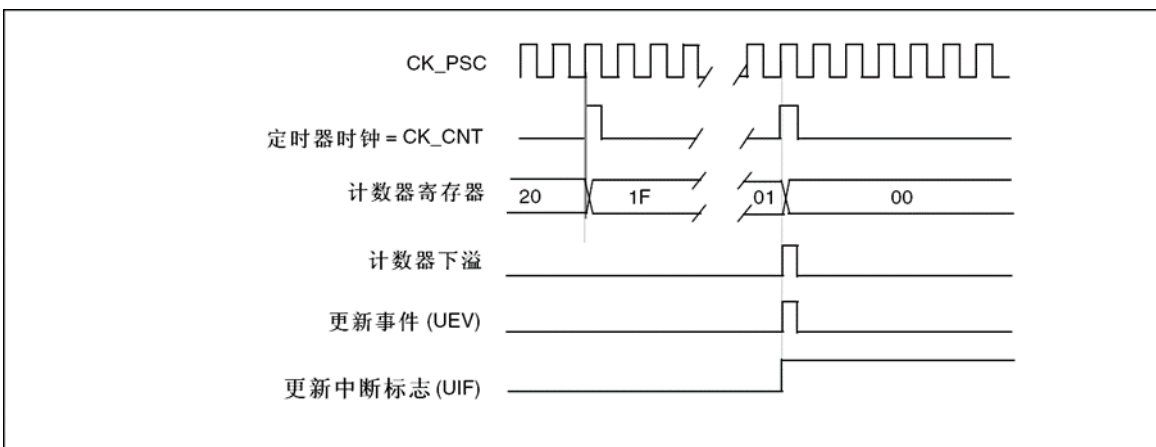


图76. 计数器时序图，内部时钟分频因子为 N

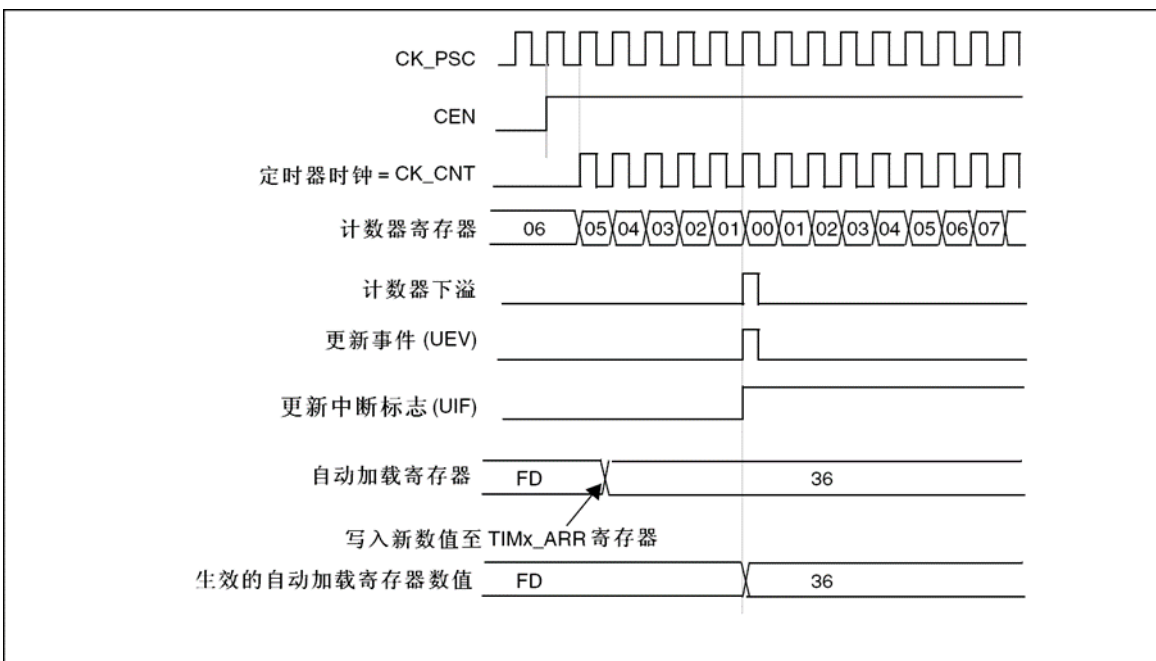


图77. 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

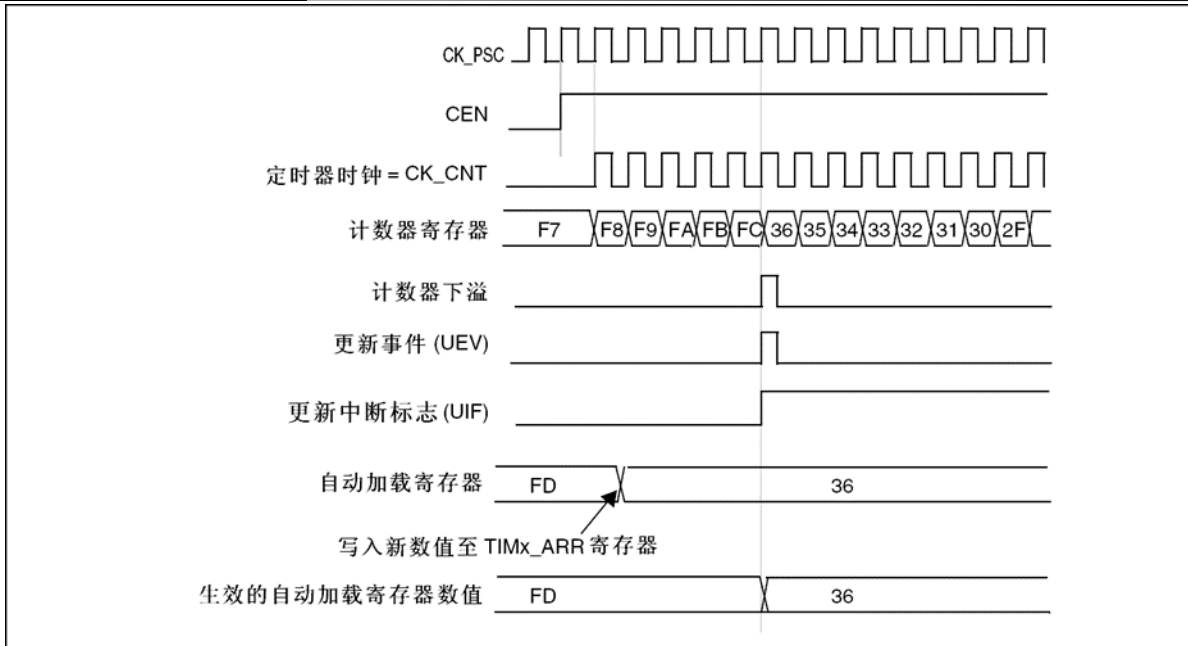


图78. 计数器时序图，ARPE=1 时的更新事件(计数器溢出)

重复计数器

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

这意味着在每 N 次计数上溢或下溢时，数据从预装载寄存器传输到影子寄存器(TIM1_ARR 自动重载入寄存器，TIM1_PSC 预装载寄存器，还有在比较模式下的捕获/比较寄存器 TIM1_CCRx)，N 是 TIM1_RCR 重复计数寄存器中的值。

重复计数器在下述任一条件成立时递减：

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

重复计数器是自动加载的，重复速率是由 TIM1_RCR 寄存器的值定义。当更新事件由软件产生(通过设置 TIM1_EGR 中的 UG 位)或者通过硬件的从模式控制器产生，则无论重复计数器的值是多少，立即发生更新事件，并且 TIM1_RCR 寄存器中的内容被重载入到重复计数器。

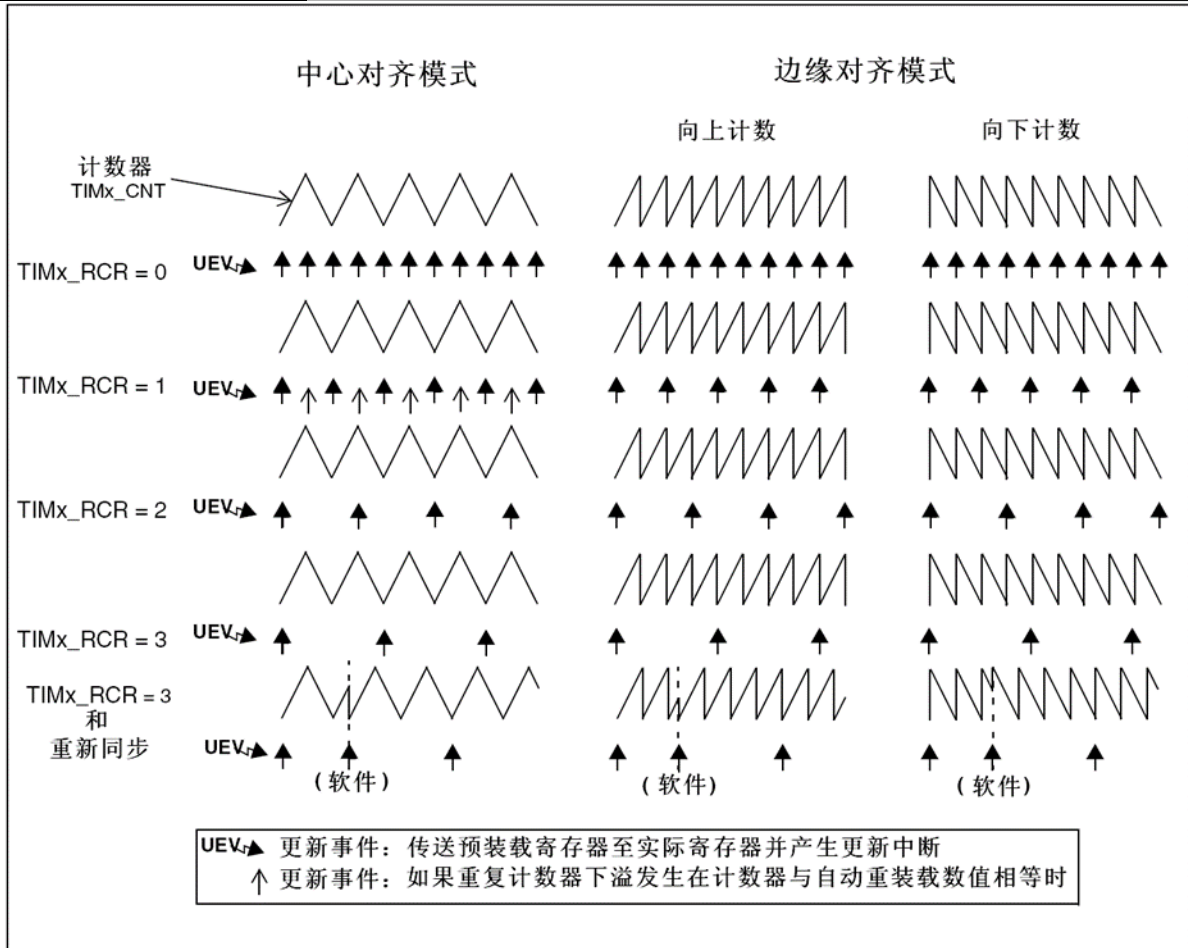


图79. 不同模式下更新速率的例子，及 TIM1_RCR 的寄存器设置

16.3.4. 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
- 外部时钟模式 2: 外部触发输入 ETR
- 内部触发输入(ITRx): 使用一个定时器作为另一个定时器的预分频器。如可以配置一个定时器 Timer1 而作为另一个定时器 Timer2 的预分频器。

内部时钟源

如果禁止了从模式控制器(SMS=000)，则 CEN、DIR(TIM1_CR1 寄存器)和 UG 位(TIM1_EGR 寄存器)是事实上的控制位，并且只能被软件修改(UG 位仍被自动清除)。只要 CEN 位被写成'1'，预分频器的时钟就由内部时钟 CK_INT 提供。

下图显示控制电路和向上计数器在一般模式下，不带预分频器时的操作。

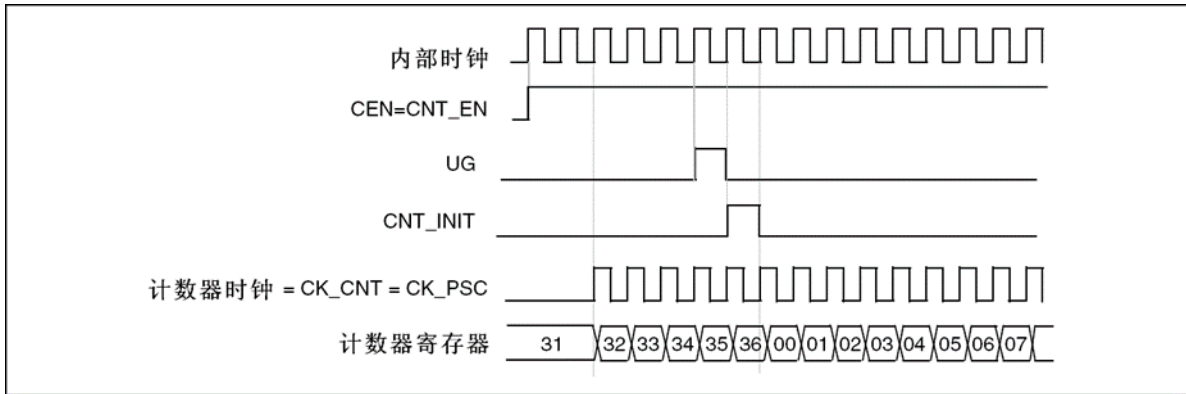


图80. 一般模式下的控制电路，内部时钟分频因子为 1

外部时钟源模式 1

当 TIM1_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

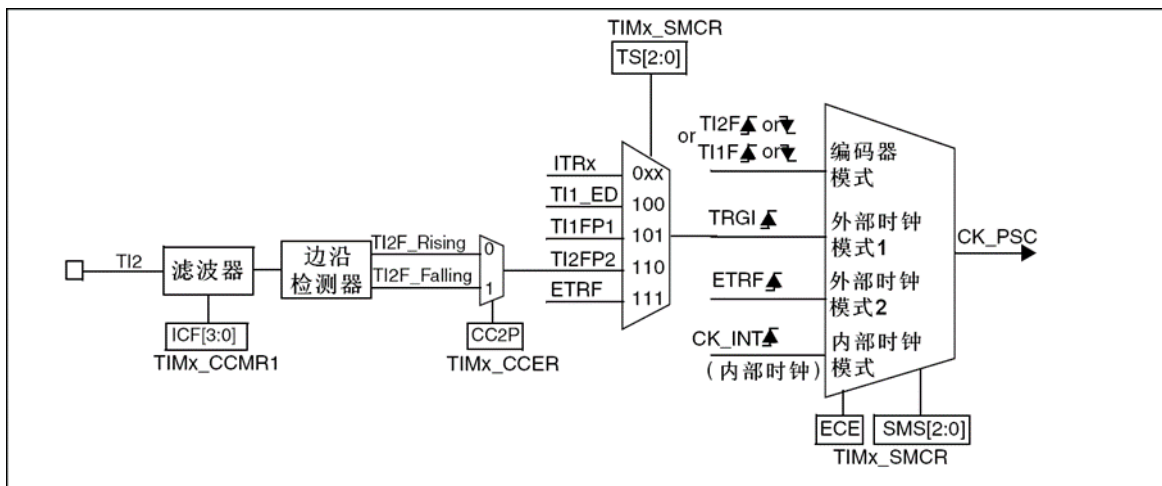


图81. TI2 外部时钟连接例子

例如，要配置向上计数器在 T12 输入端的上升沿计数，使用下列步骤：

- 配置 TIM1_CCMR1 寄存器 CC2S=01，配置通道 2 检测 T12 输入的上升沿
- 配置 TIM1_CCMR1 寄存器的 IC2F[3:0]，选择输入滤波器带宽(如果不需要滤波器，保持 IC2F=0000)
- 配置 TIM1_CCER 寄存器的 CC2P=0，选定上升沿极性
- 配置 TIM1_SMCR 寄存器的 SMS=111，选择定时器外部时钟模式 1
- 配置 TIM1_SMCR 寄存器中的 TS=110，选定 T12 作为触发输入源 设置 TIM1_CR1 寄存器的 CEN=1，启动计数器

注：捕获预分频器不用作触发，所以不需要对它进行配置

当上升沿出现在 T12，计数器计数一次，且 TIF 标志被设置。

在 T12 的上升沿和计数器实际时钟之间的延时，取决于在 T12 输入端的重新同步电路。

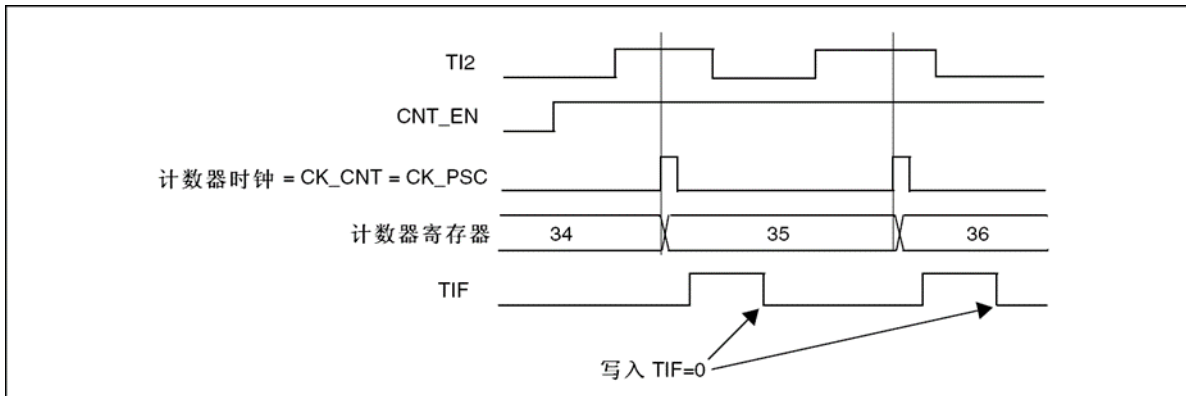


图82. 外部时钟模式 1 下的控制电路

外部时钟模式 2

选定此模式的方法为：令 TIM1_SMCR 寄存器中的 ECE=1 计数器能够在外部触发 ETR 的每一个上升沿或下降沿计数。下图是外部触发输入的框图

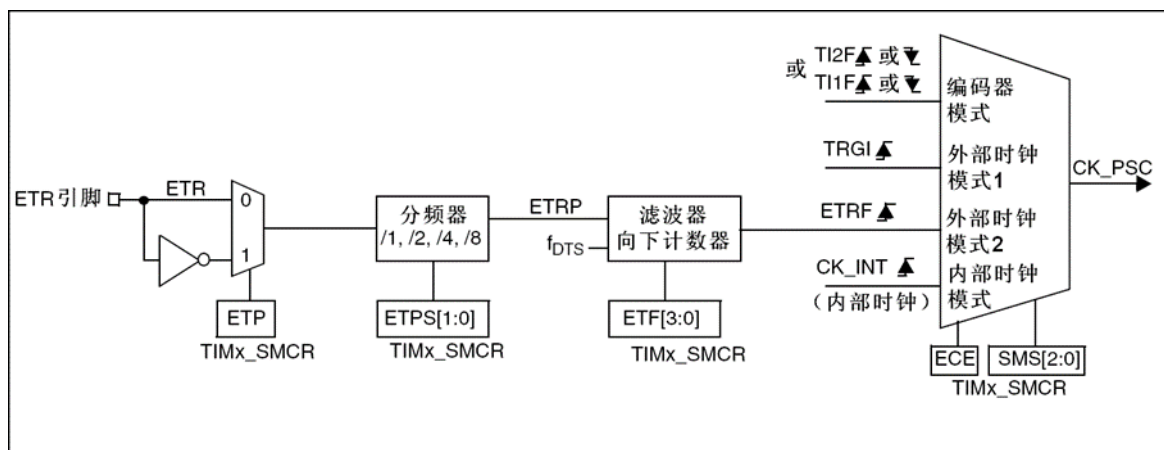


图83. 外部触发输入框图

例如，要配置在 ETR 下每 2 个上升沿计数一次的向上计数器，使用下列步骤：

1. 本例中不需要滤波器，置 TIM1_SMCR 寄存器中的 ETF[3:0]=0000
2. 设置预分频器，置 TIM1_SMCR 寄存器中的 ETPS[1:0]=01
3. 选择 ETR 的上升沿检测，置 TIM1_SMCR 寄存器中的 ETP=0
4. 开启外部时钟模式 2，写 TIM1_SMCR 寄存器中的 ECE=1
5. 启动计数器，写 TIM1_CR1 寄存器中的 CEN=1

计数器在每 2 个 ETR 上升沿计数一次。

在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的重新同步电路。

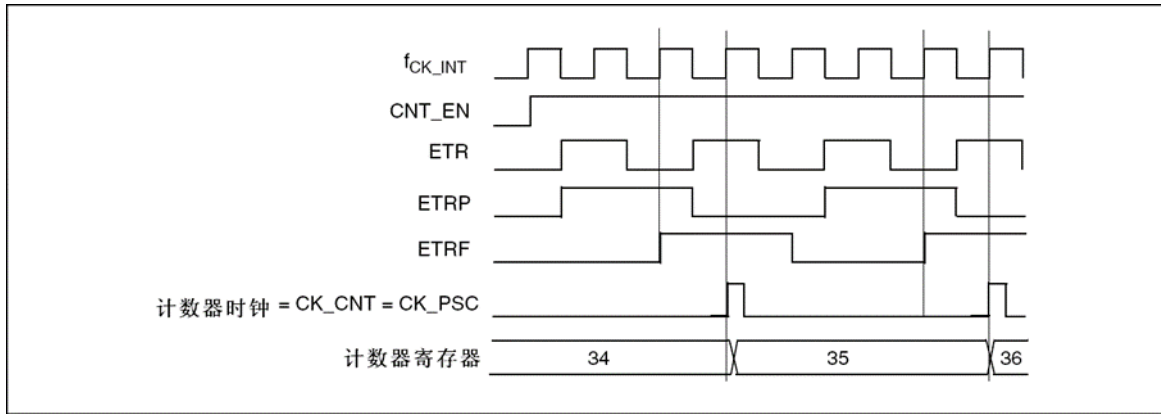


图84. 外部时钟模式 2 下的控制电路

捕获/比较通道

每一个捕获/比较通道都是围绕着一个捕获/比较寄存器(包含影子寄存器), 包括捕获的输入部分(数字滤波、多路复用和预分频器), 和输出部分(比较器和输出控制)。下列各图是一个捕获/比较通道概览。输入部分对相应的 TIx 输入信号采样, 并产生一个滤波后的信号 $TIxF$ 。然后, 一个带极性选择的边缘监测器产生一个信号($TIxFPx$), 它可以作为从模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器($ICxPS$)。

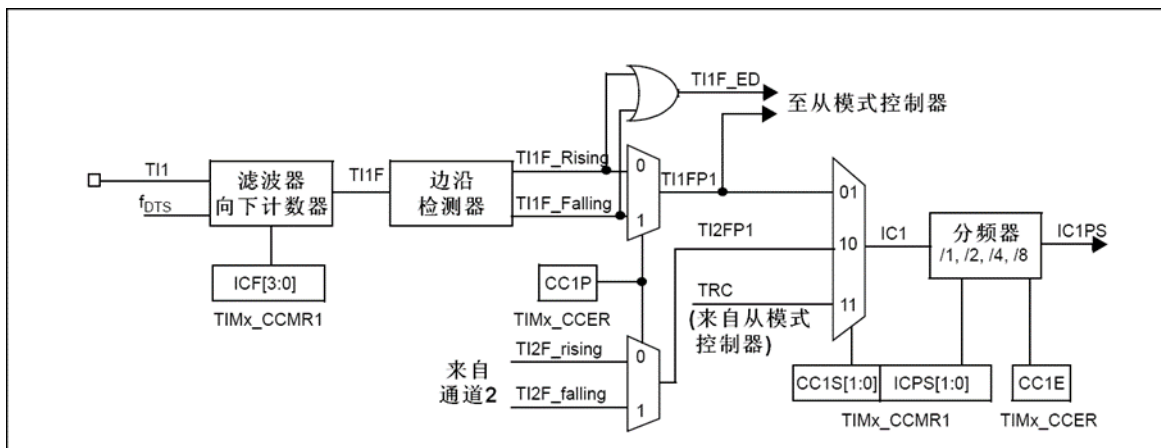


图85. 捕获/比较通道(如: 通道 1 输入部分)

输出部分产生一个中间波形 $OCxRef$ (高有效)作为基准, 链的末端决定最终输出信号的极性。

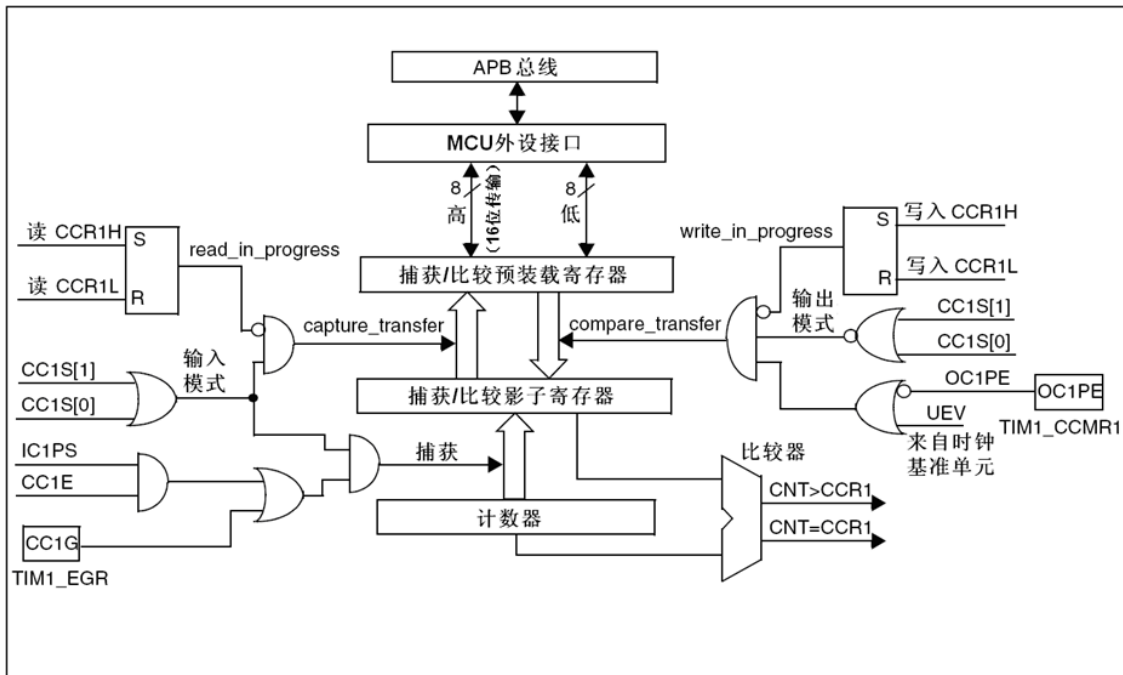


图86. 捕获/比较通道 1 的主电路

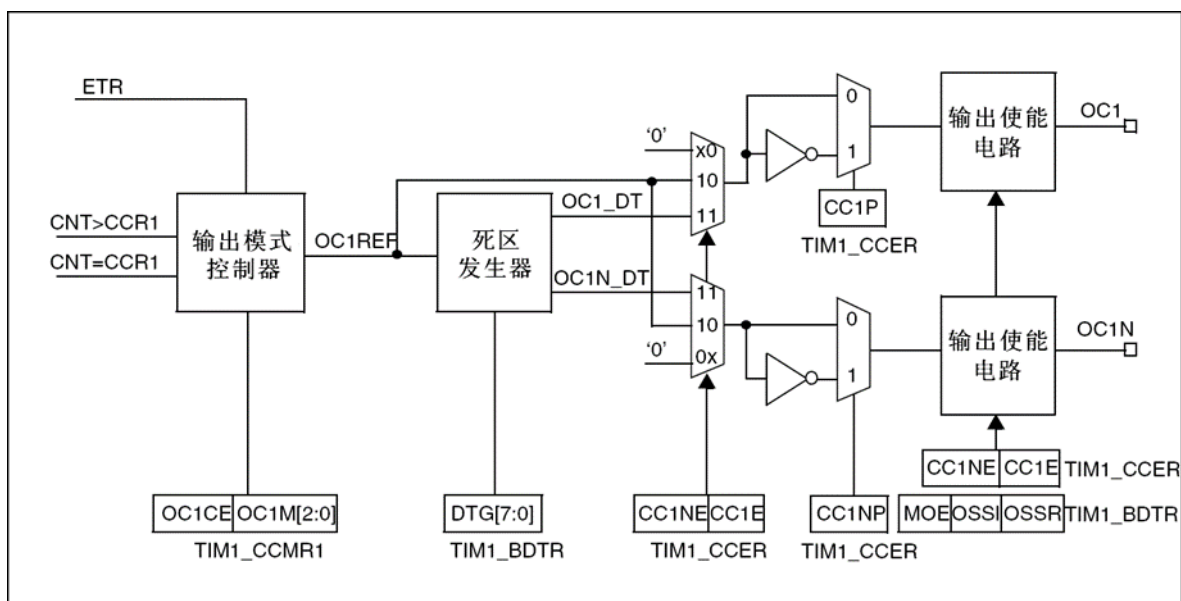


图87. 捕获/比较通道的输出部分(通道 1 至 3)

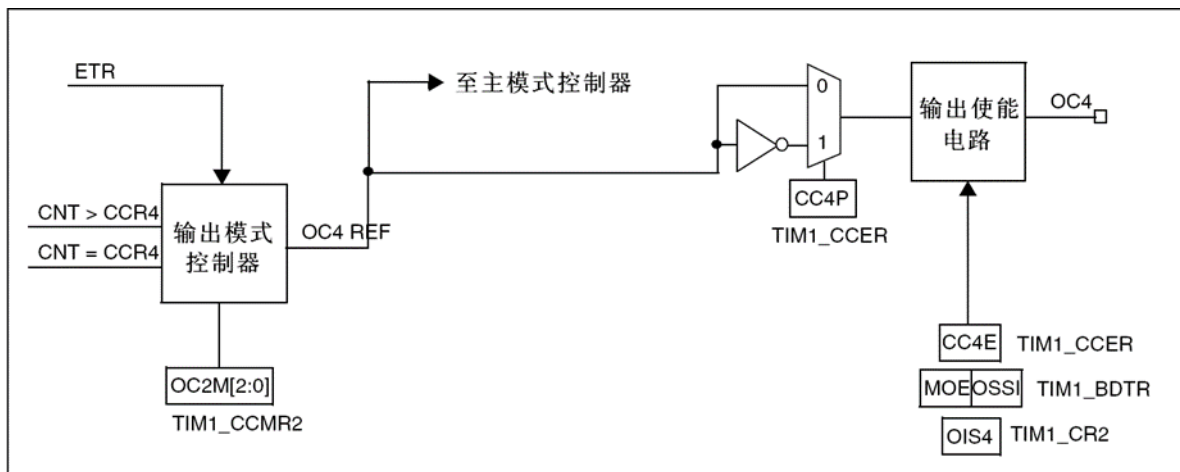


图88. 捕获/比较通道的输出部分(通道 4)

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。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

16.3.5. 输入捕获模式

在输入捕获模式下，当检测到 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器(TIM1_CCRx)中。当发生捕获事件时，相应的 CCxIF 标志(TIM1_SR 寄存器)被置 1，如果开放了中断或者 DMA 操作，则将产生中断或者 DMA 请求。如果发生捕获事件时 CCxIF 标志已经为高，那么重复捕获标志 CCxOF(TIM1_SR 寄存器)被置 1。写 CCxIF=0 可清除 CCxIF，或读取存储在 TIM1_CCRx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOF=0 可清除 CCxOF。

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIM1_CCR1 寄存器中，步骤如下：

- 选择有效输入端：TIM1_CCR1 必须连接到 TI1 输入，所以写入 TIM1_CCR1 寄存器中的 CC1S=01，只要 CC1S 不为'00'，通道被配置为输入，并且 TIM1_CCR1 寄存器变为只读。
- 根据输入信号的特点，配置输入滤波器为所需的带宽(即输入为 Tlx 时，输入滤波器控制位是 TIM1_CCMRx 寄存器中的 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以(以 fDTS 频率)连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIM1_CCMR1 寄存器中写入 IC1F=0011。
- 选择 TI1 通道的有效转换边沿，在 TIM1_CCER 寄存器中写入 CC1P=0(上升沿)。
- 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止(写 TIM1_CCMR1 寄存器的 IC1PS=00)。
- 设置 TIM1_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
- 如果需要，通过设置 TIM1_DIER 寄存器中的 CC1IE 位允许相关中断请求，通过设置
- TIM1_DIER 寄存器中的 CC1DE 位允许 DMA 请求。当发生一个输入捕获时：
- 产生有效的电平转换时，计数器的值被传送到 TIM1_CCR1 寄存器。
- CC1IF 标志被设置(中断标志)。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除，CC1OF 也被置 1。
- 如设置了 CC1IE 位，则会产生一个中断。
- 如设置了 CC1DE 位，则还会产生一个 DMA 请求。

为了处理捕获溢出，建议在读出捕获溢出标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注：设置 TIM1_EGR 寄存器中相应的 CCxG 位，可以通过软件产生输入捕获中断和/或 DMA 请求。

16.3.6. PWM 输入模式

该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

- 两个 ICx 信号被映射至同一个 Tlx 输入。
- 这 2 个 ICx 信号为边沿有效，但是极性相反。
- 其中一个 TlxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。

例如，你需要测量输入到 TI1 上的 PWM 信号的长度(TIM1_CCR1 寄存器)和占空比(TIM1_CCR2 寄存器)，具体步骤如下(取决于 CK_INT 的频率和预分频器的值)

- 选择 TIM1_CCR1 的有效输入：置 TIM1_CCMR1 寄存器的 CC1S=01(选中 TI1)。
- 选择 TI1FP1 的有效极性(用来捕获数据到 TIM1_CCR1 中和清除计数器)：置 CC1P=0(上升沿有效)。

- 选择 TIM1_CCR2 的有效输入: 置 TIM1_CCMR1 寄存器的 CC2S=10(选中 TI1)。
- 选择 TI1FP2 的有效极性(捕获数据到 TIM1_CCR2): 置 CC2P=1(下降沿有效)。
- 选择有效的触发输入信号: 置 TIM1_SMCR 寄存器中的 TS=101(选择 TI1FP1)。
- 配置从模式控制器为复位模式: 置 TIM1_SMCR 中的 SMS=100。
- 使能捕获: 置 TIM1_CCER 寄存器中 CC1E=1 且 CC2E=1。

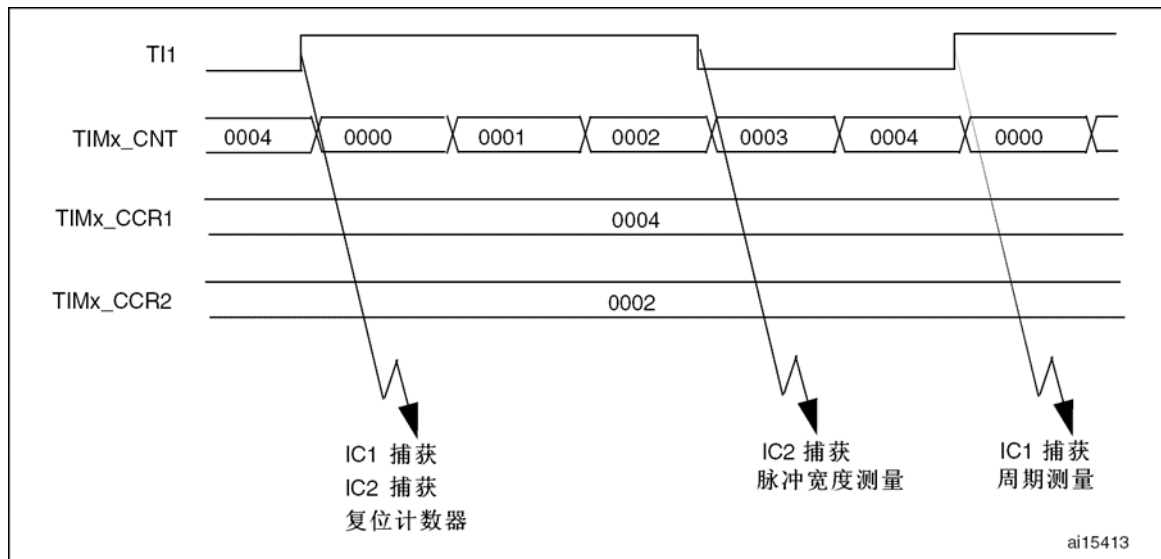


图89. PWM 输入模式时序

因为只有 TI1FP1 和 TI2FP2 连到了从模式控制器, 所以 PWM 输入模式只能使用 TIM1_CH1/TIM1_CH2 信号。

16.3.7. 强置输出模式

在输出模式(TIM1_CCMRx 寄存器中 CCxS=00)下, 输出比较信号(OCxREF 和相应的 OCx/OCxN) 能够直接由软件强置为有效或无效状态, 而不依赖于输出比较寄存器和计数器间的比较结果。置 TIM1_CCMRx 寄存器中相应的 OCxM=101, 即可强置输出比较信号(OCxREF/OCx)为有效状态。这样 OCxREF 被强置为高电平(OCxREF 始终为高电平有效), 同时 OCx 得到 CCxP 极性相反的信号。

例如: CCxP=0(OCx 高电平有效), 则 OCx 被强置为高电平。

置 TIM1_CCMRx 寄存器中的 OCxM=100, 可强置 OCxREF 信号为低。

该模式下, 在 TIM1_CCRx 影子寄存器和计数器之间的比较仍然在进行, 相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。这将会在下面的输出比较模式一节中介绍。

16.3.8. 输出比较模式

此项功能是用来控制一个输出波形, 或者指示一段给定的时间已经到时。当计数器与捕获/比较寄存器的内容相同时, 输出比较功能做如下操作:

- 将输出比较模式(TIM1_CCMRx 寄存器中的 OCxM 位)和输出极性(TIM1_CCER 寄存器中的 CCxP 位)定义的值输出到对应的引脚上。在比较匹配时, 输出引脚可以保持它的电平(OCxM=000)、被设置成有效电平(OCxM=001)、被设置成无效电平(OCxM=010)或进行翻转(OCxM=011)。
- 设置中断状态寄存器中的标志位(TIM1_SR 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽(TIM1_DIER 寄存器中的 CCxIE 位), 则产生一个中断。

• 若设置了相应的使能位(TIM1_DIER 寄存器中的 CCxDE 位, TIM1_CR2 寄存器中的 CCDS 位选择 DMA 请求功能), 则产生一个 DMA 请求。

TIM1_CCMRx 中的 OCxPE 位选择 TIM1_CCRx 寄存器是否需要使用预装载寄存器。在输出比较模式下, 更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。

同步的精度可以达到计数器的一个计数周期。输出比较模式(在单脉冲模式下)也能用来输出一个单脉冲。

输出比较模式的配置步骤:

1. 选择计数器时钟(内部, 外部, 预分频器)。
2. 将相应的数据写入 TIM1_ARR 和 TIM1_CCRx 寄存器中。
3. 如果要产生一个中断请求, 设置 CCxIE 位。
4. 选择输出模式, 例如:
 - 要求计数器与 CCRx 匹配时翻转 OCx 的输出引脚, 设置 OCxM=011
 - 置 OCxPE = 0 禁用预装载寄存器
 - 置 CCxP = 0 选择极性为高电平有效
 - 置 CCxE = 1 使能输出
5. 设置 TIM1_CR1 寄存器的 CEN 位启动计数器

TIM1_CCRx 寄存器能够在任何时候通过软件进行更新以控制输出波形, 条件是未使用预装载寄存器 (OCxPE='0', 否则 TIM1_CCRx 的影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

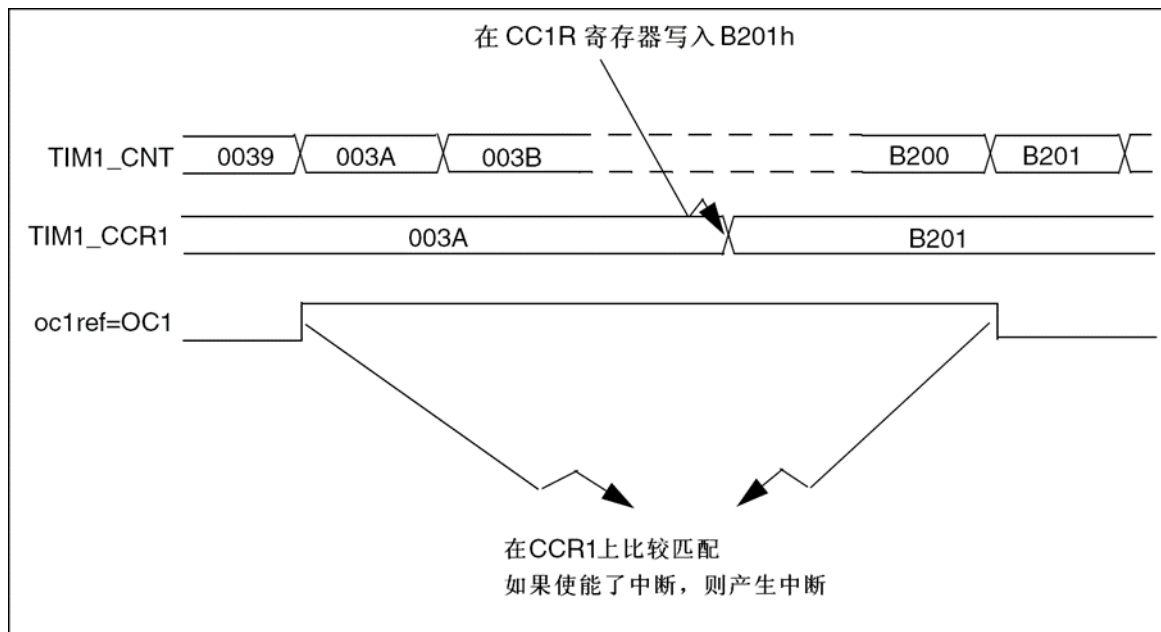


图90. 输出比较模式, 翻转 OC1

16.3.9. PWM 模式

脉冲宽度调制模式可以产生一个由 TIM1_ARR 寄存器确定频率、由 TIM1_CCRx 寄存器确定占空比的信号。在 TIM1_CCMRx 寄存器中的 OCxM 位写入 '110' (PWM 模式 1) 或 '111' (PWM 模式 2), 能够独立地设置每个 OCx 输出通道产生一路 PWM。必须通过设置 TIM1_CCMRx 寄存器的 OCxPE 位使能相应的预装载寄存器, 最后还要设置 TIM1_CR1 寄存器的 ARPE 位, (在向上计数或中心对称模式中) 使能自动重载的预装载寄存器。仅当发生一个更新事件的时候, 预装载寄存器才能被传送到影子寄存器, 因此在计数器开始计数之前, 必须通过设置 TIM1_EGR 寄存器中的 UG 位来初始化所有的寄存器。

OCx 的极性可以通过软件在 TIM1_CCER 寄存器中的 CCxP 位设置, 它可以设置为高电平有效或低电平有效。OCx 的输出使能通过(TIM1_CCER 和 TIM1_BDTR 寄存器中)CCxE、CCxNE、MOE、OSSI 和 OSSR 位的组合控制。详见 TIM1_CCER 寄存器的描述。

在 PWM 模式(模式 1 或模式 2)下, TIM1_CNT 和 TIM1_CCRx 始终在进行比较, (依据计数器的计数方向)以确定是否符合 $TIM1_CCRx \leq TIM1_CNT$ 或者 $TIM1_CNT \leq TIM1_CCRx$ 。

根据 TIM1_CR1 寄存器中 CMS 位的状态, 定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

PWM 边沿对齐模式

• 向上计数配置

当 TIM1_CR1 寄存器中的 DIR 位为低的时候执行向上计数。

下面是一个 PWM 模式 1 的例子。当 $TIM1_CNT < TIM1_CCRx$ 时, PWM 参考信号 OCxREF 为高, 否则为低。如果 TIM1_CCRx 中的比较值大于自动重装载值(TIM1_ARR), 则 OCxREF 保持为'1'。如果比较值为 0, 则 OCxREF 保持为'0'。下图为 TIM1_ARR=8 时边沿对齐的 PWM 波形实例。

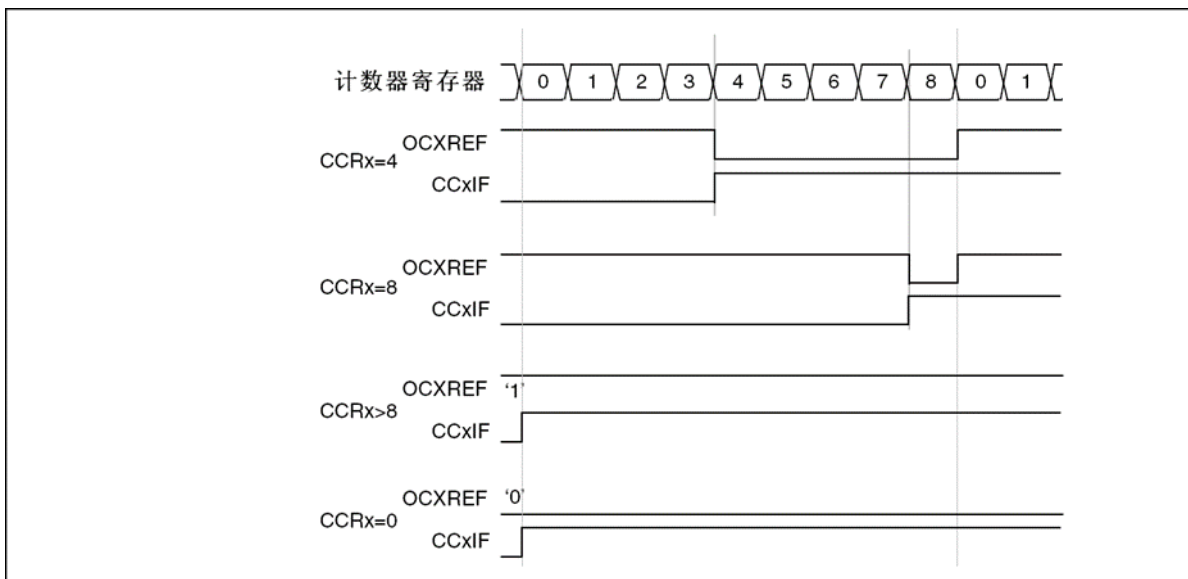


图91. 边沿对齐的 PWM 波形(ARR=8)

• 向下计数的配置

当 TIM1_CR1 寄存器的 DIR 位为高时执行向下计数。

在 PWM 模式 1, 当 $TIM1_CNT > TIM1_CCRx$ 时参考信号 OCxREF 为低, 否则为高。如果 TIM1_CCRx 中的比较值大于 TIM1_ARR 中的自动重装载值, 则 OCxREF 保持为'1'。该模式下不能产生 0% 的 PWM 波形。

PWM 中央对齐模式

当 TIM1_CR1 寄存器中的 CMS 位不为'00'时为中央对齐模式(所有其他的配置对 OCxREF/OCx 信号都有相同的作用)。根据不同的 CMS 位设置, 比较标志可以在计数器向上计数时被置 1、在计数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。TIM1_CR1 寄存器中的计数方向位(DIR)由硬件更新, 不要软件修改它。

下图给出了一些中央对齐的 PWM 波形的例子

- TIM1_ARR=8
- PWM 模式 1
- TIM1_CR1 寄存器的 CMS=01, 在中央对齐模式 1 下, 当计数器向下计数时设置比较标志。

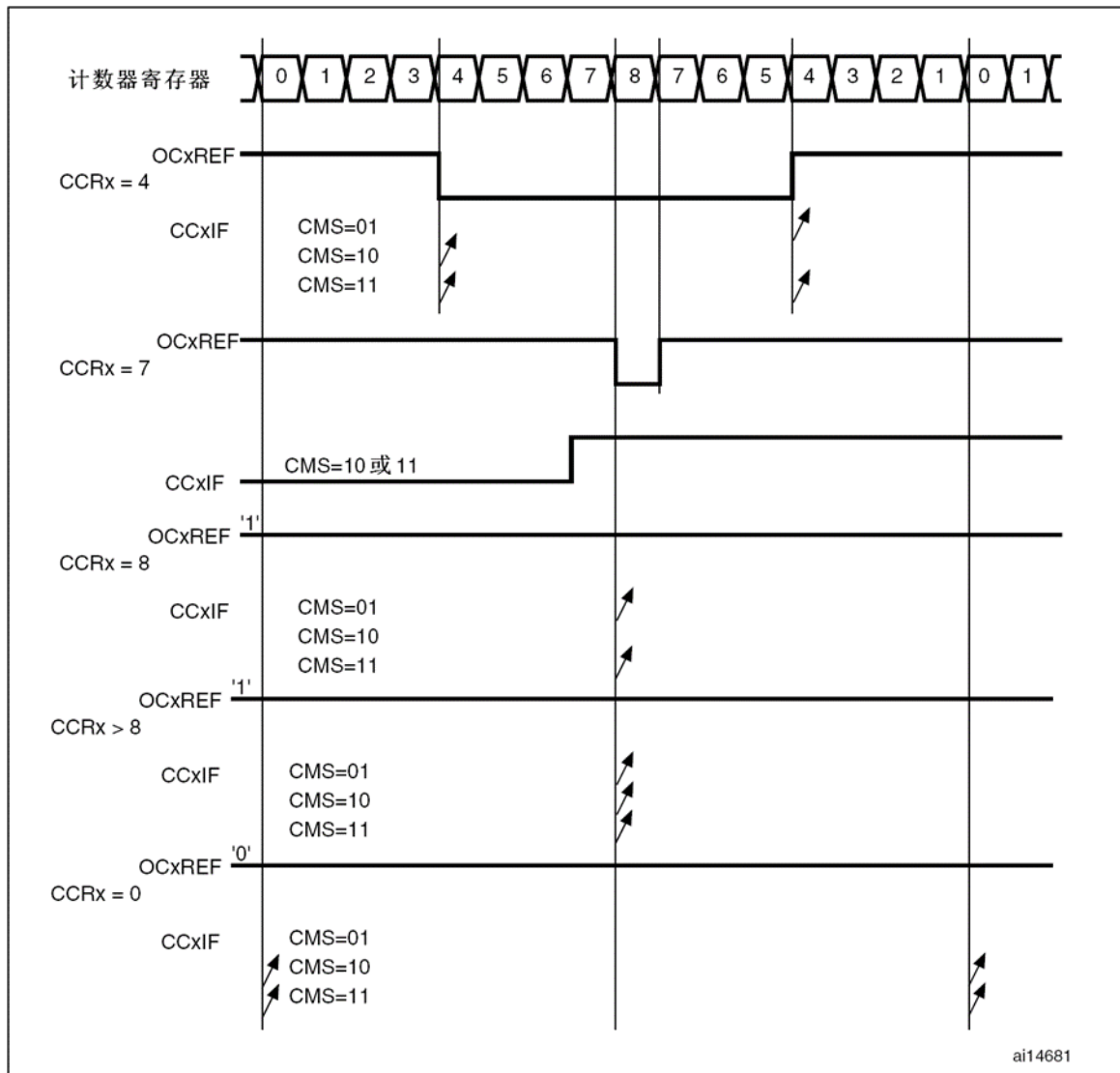


图92. 中央对齐的 PWM 波形(APR=8)

使用中央对齐模式的提示:

- 进入中央对齐模式时, 使用当前的向上/向下计数配置; 这意味着计数器向上还是向下计数取决于 TIM1_CR1 寄存器中 DIR 位的当前值。此外, 软件不能同时修改 DIR 和 CMS 位。
- 不推荐当运行在中央对齐模式时改写计数器, 因为这会产生不可预知的结果。特别地:
 - 如果写入计数器的值大于自动重加载的值(TIM1_CNT>TIM1_ARR), 则方向不会被更新。例如, 如果计数器正在向上计数, 它就会继续向上计数。
 - 如果将 0 或者 TIM1_ARR 的值写入计数器, 方向被更新, 但不产生更新事件 UEV。
- 使用中央对齐模式最保险的方法, 就是在启动计数器之前产生一个软件更新(设置 TIM1_EGR 位中的 UG 位), 并且不要在计数进行过程中修改计数器的值

互补输出和死区插入

高级控制定时器(TIM1 和 TIM8)能够输出两路互补信号, 并且能够管理输出的瞬时关断和接通。

这段时间通常被称为死区，用户应该根据连接的输出器件和它们的特性(电平转换的延时、电源开关的延时等)来调整死区时间。配置 TIM1_CCER 寄存器中的 CCxP 和 CCxNP 位，可以为每一个输出独立地选择极性(主输出 OCx 或互补输出 OCxN)。互补信号 OCx 和 OCxN 通过下列控制位的组合进行控制：TIM1_CCER 寄存器的 CCxE 和 CCxNE 位，TIM1_BDTR 和 TIM1_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位，详见表 75 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。特别的是，在转换到 IDLE 状态时(MOE 下降到 0)死区被激活。

同时设置 CCxE 和 CCxNE 位将插入死区，如果存在刹车电路，则还要设置 MOE 位。每一个通道都有一个 10 位的死区发生器。参考信号 OCxREF 可以产生 2 路输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高有效：

- OCx 输出信号与参考信号相同，只是它的上升沿相对于参考信号的上升沿有一个延迟。
- OCxN 输出信号与参考信号相反，只是它的上升沿相对于参考信号的下降沿有一个延迟。

如果延迟大于当前有效的输出宽度(OCx 或者 OCxN)，则不会产生相应的脉冲。

下列几张图显示了死区发生器的输出信号和当前参考信号 OCxREF 之间的关系。(假设 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1)

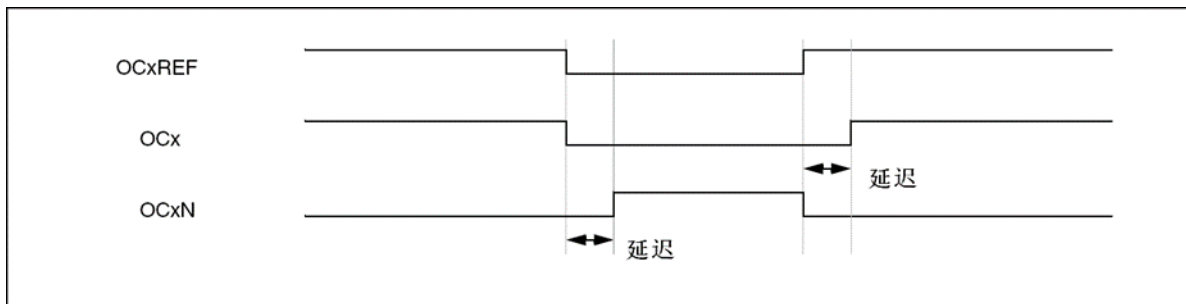


图93. 带死区插入的互补输出

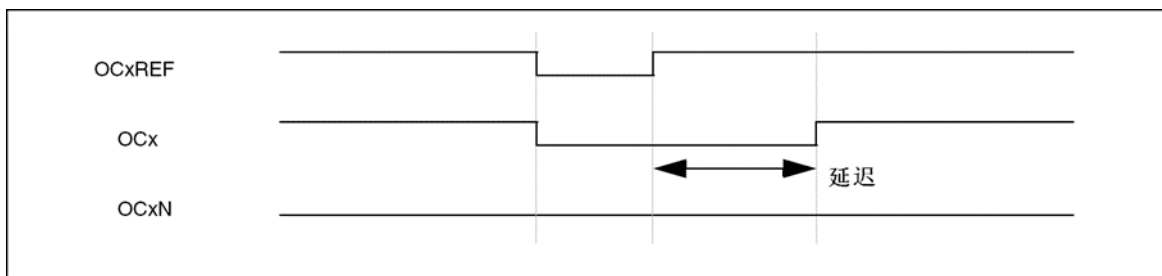


图94. 死区波形延迟大于负脉冲

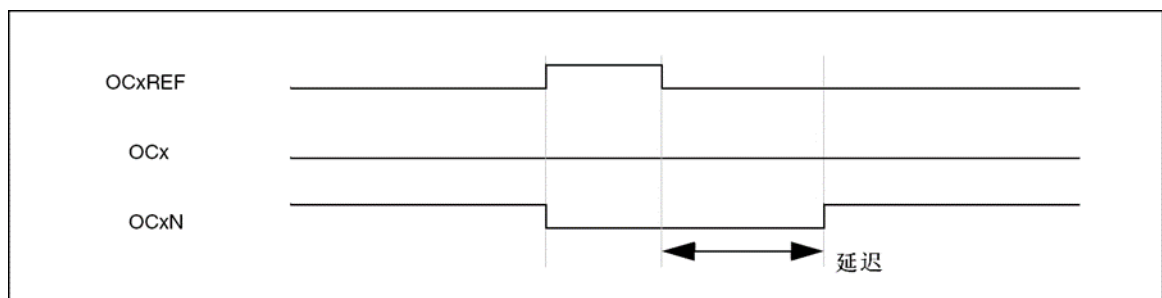


图95. 死区波形延迟大于正脉冲

每一个通道的死区延时都是相同的，是由 TIM1_BDTR 寄存器中的 DTG 位编程配置。

重定向 OCxREF 到 OCx 或 OCxN

在输出模式下(强置、输出比较或 PWM), 通过配置 TIM1_CCER 寄存器的 CCxE 和 CCxNE 位,

OCxREF 可以被重定向到 OCx 或者 OCxN 的输出。这个功能可以在互补输出处于无效电平时, 在某个输出上送出一个特殊的波形(例如 PWM 或者静态有效电平)。另一个作用是, 让两个输出同时处于无效电平, 或处于有效电平和带死区的互补输出。

注: 当只使能 OCxN(CCxE=0, CCxNE=1)时, 它不会反相, 当 OCxREF 有效时立即变高。例如, 如果 CCxNP=0, 则 OCxN=OCxREF。另一方面, 当 OCx 和 OCxN 都被使能时(CCxE=CCxNE=1), 当 OCxREF 为高时 OCx 有效; 而 OCxN 相反, 当 OCxREF 低时 OCxN 变为有效。

使用刹车功能

当使用刹车功能时, 依据相应的控制位(TIM1_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位, TIM1_CR2 寄存器中的 OISx 和 OISxN 位), 输出使能信号和无效电平都会被修改。但无论何时, OCx 和 OCxN 输出不能在同一时间同时处于有效电平上。

刹车源既可以是刹车输入引脚又可以是一个时钟失败事件。时钟失败事件由复位时钟控制器中的时钟安全系统产生。系统复位后, 刹车电路被禁止, MOE 位为低。设置 TIM1_BDTR 寄存器中的 BKE 位可以使能刹车功能, 刹车输入信号的极性可以通过配置同一个寄存器中的 BKP 位选择。BKE 和 BKP 可以同时被修改。当写入 BKE 和 BKP 位时, 在真正写入之前会有 1 个 APB 时钟周期的延迟, 因此需要等待一个 APB 时钟周期之后, 才能正确地读回写入的位。因为 MOE 下降沿可以是异步的, 在实际信号(作用在输出端)和同步控制位(在 TIM1_BDTR 寄存器中)之间设置了一个再同步电路。这个再同步电路会在异步信号和同步信号之间产生延迟。特别的, 如果当它为低时写 MOE=1, 则读出它之前必须先插入一个延时(空指令)才能读到正确的值。这是因为写入的是异步信号而读的是同步信号。

当发生刹车时(在刹车输入端出现选定的电平), 有下述动作:

- MOE 位被异步地清除, 将输出置于无效状态、空闲状态或者复位状态(由 OSSI 位选择)。这个特性在 MCU 的振荡器关闭时依然有效。

- 一旦 MOE=0, 每一个输出通道输出由 TIM1_CR2 寄存器中的 OISx 位设定的电平。如果 OSSI=0, 则定时器释放使能输出, 否则使能输出始终为高。

- 当使用互补输出时:

- 输出首先被置于复位状态即无效的状态(取决于极性)。这是异步操作, 即使定时器没有时钟时, 此功能也有效。
- 如果定时器的时钟依然存在, 死区生成器将会重新生效, 在死区之后根据 OISx 和 OISxN 位指示的电平驱动输出端口。即使在这种情况下, OCx 和 OCxN 也不能被同时驱动到有效的电平。注, 因为重新同步 MOE, 死区时间比通常情况下长一些(大约 2 个 ck_tim 的时钟周期)。
- 如果 OSSI=0, 定时器释放使能输出, 否则保持使能输出; 或一旦 CCxE 与 CCxNE 之一变高时, 使能输出变为高。

- 如果设置了 TIM1_DIER 寄存器中的 BIE 位, 当刹车状态标志(TIM1_SR 寄存器中的 BIF 位)为'1'时, 则产生一个中断。如果设置了 TIM1_DIER 寄存器中的 BDE 位, 则产生一个 DMA 请求。

- 如果设置了 TIM1_BDTR 寄存器中的 AOE 位, 在下一个更新事件 UEV 时 MOE 位被自动置位; 例如, 这可以用来进行整形。否则, MOE 始终保持低直到被再次置'1'; 此时, 这个特性可以被用在安全方面, 你可以把刹车输入连到电源驱动的报警输出、热敏传感器或者其他安全器件上。

注: 刹车输入为电平有效。所以, 当刹车输入有效时, 不能同时(自动地或者通过软件)设置 MOE。同时, 状态标志 BIF 不能被清除。

刹车由 BRK 输入产生，它的有效极性是可编程的，且由 TIM1_BDTR 寄存器中的 BKE 位开启。除了刹车输入和输出管理，刹车电路中还实现了写保护以保证应用程序的安全。它允许用户冻结几个配置参数(死区长度，OCx/OCxN 极性和被禁止的状态，OCxM 配置，刹车使能和极性)。用户可以通过 TIM1_BDTR 寄存器中的 LOCK 位，从三级保护中选择一种，参看 13.4.18 节 TIM1 和 TIM8 刹车和死区寄存器(TIM1_BDTR)。在 MCU 复位后 LOCK 位只能被修改一次。

下图显示响应刹车的输出实例。

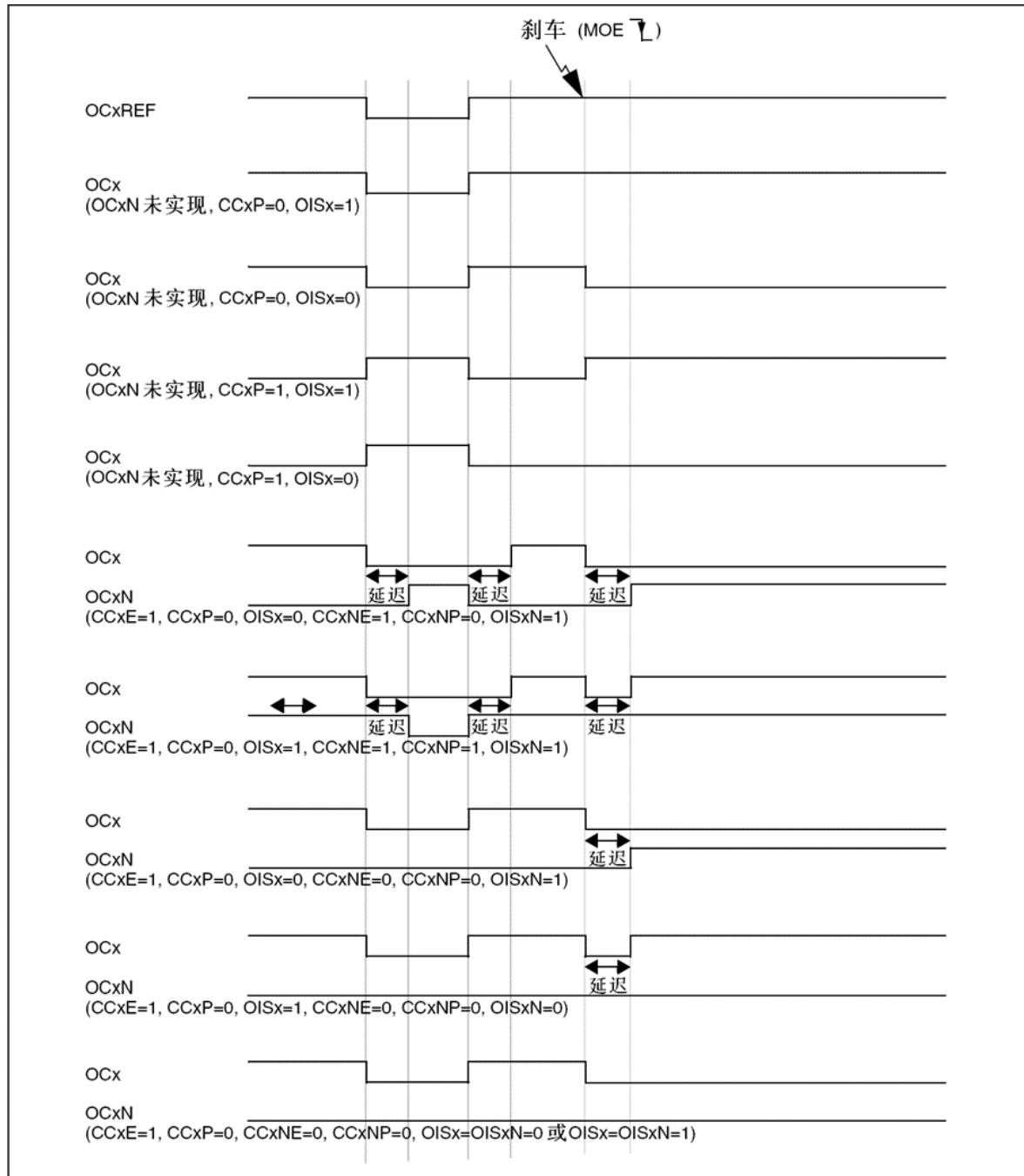


图96. 响应刹车的输出

在外部事件时清除 OCxREF 信号

对于一个给定的通道，设置 TIM1_CCMRx 寄存器中对应的 OCxCE 位为'1'，能够用 ETRF 输入端的高电平把 OCxREF 信号拉低，OCxREF 信号将保持为低直到发生下一次的更新事件 UEV。

该功能只能用于输出比较和 PWM 模式，而不能用于强置模式。

例如，OCxREF 信号可以联到一个比较器的输出，用于控制电流。这时，ETR 必须配置如下：

1. 外部触发预分频器必须处于关闭：TIM1_SMCR 寄存器中的 ETPS[1:0]=00。
2. 必须禁止外部时钟模式 2：TIM1_SMCR 寄存器中的 ECE=0。
3. 外部触发极性(ETP)和外部触发滤波器(ETF)可以根据需要配置。

下图显示了当 ETRF 输入变为高时，对应不同 OCxCE 的值，OCxREF 信号的动作。在这个例子中，定时器 TIM1 被置于 PWM 模式。

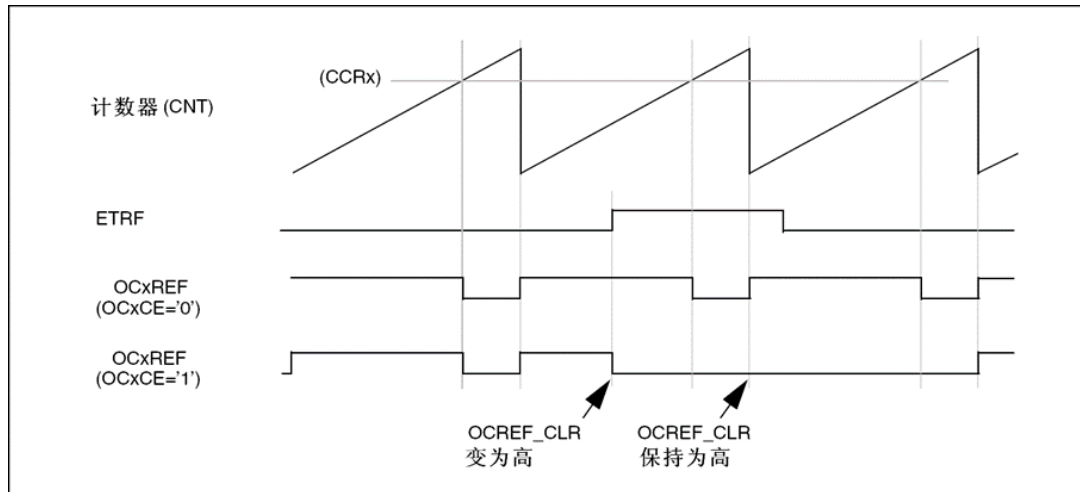


图97. 清除 TIM1 的 OCxREF

产生六步 PWM 输出

当在一个通道上需要互补输出时，预装载位有 OCxM、CCxE 和 CCxNE。在发生 COM 换相事件时，这些预装载位被传送到影子寄存器位。这样你就可以预先设置好下一步配置，并在同一个时刻同时修更改所有通道的配置。COM 可以通过设置 TIM1_EGR 寄存器的 COM 位由软件产生，或在 TRGI 上升沿由硬件产生。当发生 COM 事件时会设置一个标志位(TIM1_SR 寄存器中的 COMIF 位)，这时如果已设置了 TIM1_DIER 寄存器的 COMIE 位，则产生一个中断；如果已设置了 TIM1_DIER 寄存器的 COMDE 位，则产生一个 DMA 请求。

下图显示当发生 COM 事件时，三种不同配置下 OCx 和 OCxN 输出。

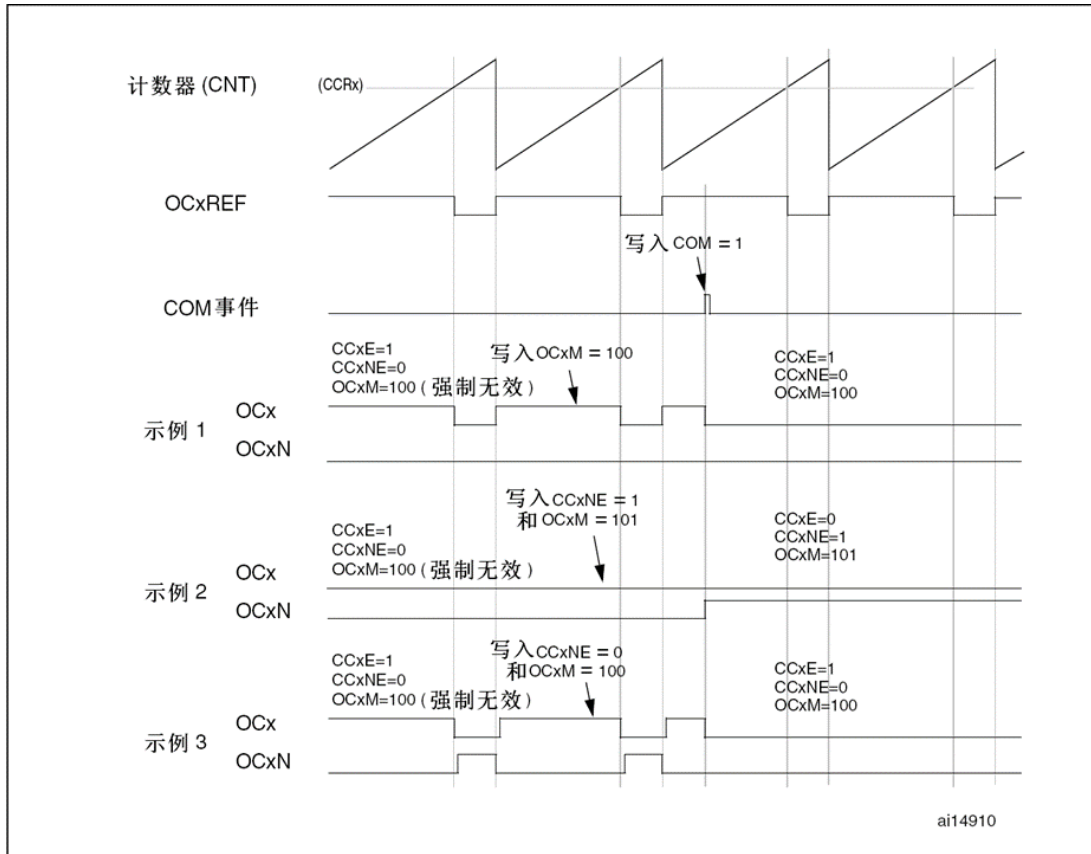


图98. 产生六步 PWM，使用 COM 的例子(OSSR=1)

16.3.9.1. 单脉冲模式

单脉冲模式(OPM)是前述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后产生一个脉宽可程序控制的脉冲。可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIM1_CR1 寄存器中的 OPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前(当定时器正在等待触发)，必须如下配置：

- 向上计数方式：计数器 $CNT < CCR_x \leq ARR$ (特别地, $0 < CCR_x$),
- 向下计数方式：计数器 $CNT > CCR_x$ 。

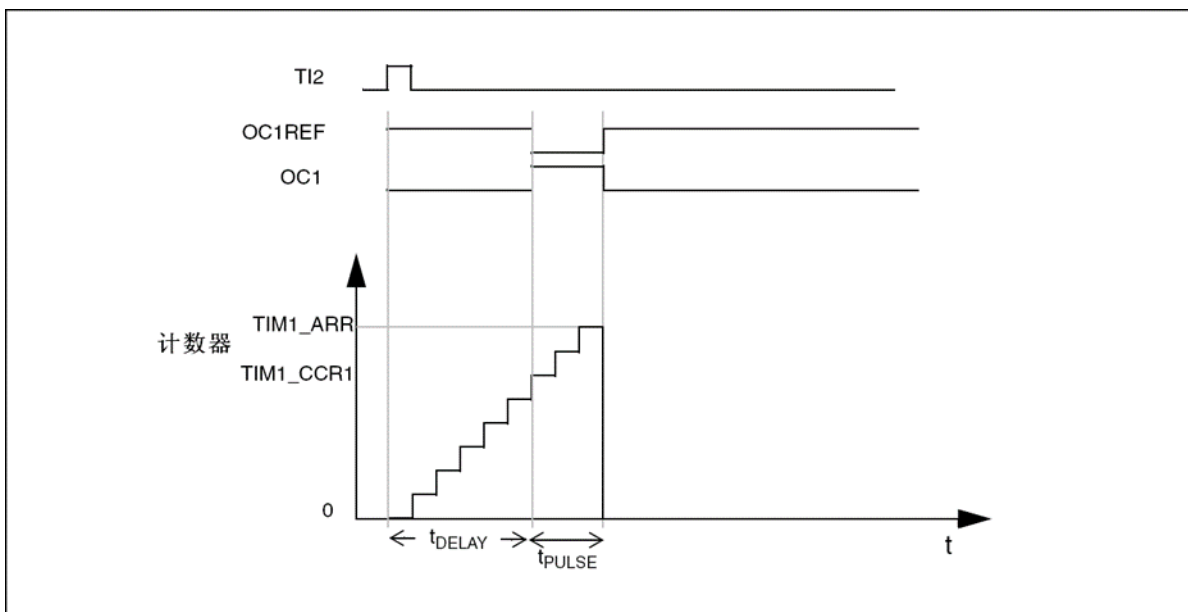


图99. 单脉冲模式的例子

例如, 你需要在从 TI2 输入脚上检测到一个上升沿开始, 延迟 tDELAY 之后, 在 OC1 上产生一个长度为 tPULSE 的正脉冲。

假定 TI2FP2 作为触发 1:

- 置 TIM1_CCMR1 寄存器中的 CC2S=01, 把 TI2FP2 映像到 TI2。
- 置 TIM1_CCER 寄存器中的 CC2P=0, 使 TI2FP2 能够检测上升沿。
- 置 TIM1_SMCR 寄存器中的 TS=110, TI2FP2 作为从模式控制器的触发(TRGI)。
- 置 TIM1_SMCR 寄存器中的 SMS=110(触发模式), TI2FP2 被用来启动计数器。
- OPM 的波形由写入比较寄存器的数值决定(要考虑时钟频率和计数器预分频器)
- tDELAY 由 TIM1_CCR1 寄存器中的值定义。
- tPULSE 由自动装载值和比较值之间的差值定义(TIM1_ARR - TIM1_CCR1)。
- 假定当发生比较匹配时要产生从 0 到 1 的波形, 当计数器达到预装载值时要产生一个从 1 到 0 的波形;

首先要置 TIM1_CCMR1 寄存器的 OC1M=111, 进入 PWM 模式 2; 根据需要选择性地使能预装载寄存器: 置 TIM1_CCMR1 中的 OC1PE=1 和 TIM1_CR1 寄存器中的 ARPE; 然后在 TIM1_CCR1 寄存器中填写比较值, 在 TIM1_ARR 寄存器中填写自动装载值, 设置 UG 位来产生一个更新事件, 然后等待在 TI2 上的一个外部触发事件。本例中, CC1P=0。

在这个例子中, TIM1_CR1 寄存器中的 DIR 和 CMS 位应该置低。因为只需要一个脉冲, 所以必须设置 TIM1_CR1 寄存器中的 OPM=1, 在下一个更新事件(当计数器从自动装载值翻转到 0)时停止计数。

特殊情况: OCx 快速使能

在单脉冲模式下, 在 TIx 输入脚的边沿检测逻辑设置 CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期, 因此它限制了可得到的最小延时 tDELAY。

如果要以最小延时输出波形, 可以设置 TIM1_CCMRx 寄存器中的 OCxFE 位; 此时 OCxREF(和 OCx)直接响应激励而不再依赖比较的结果, 输出的波形与比较匹配时的波形一样。OCxFE 只在通道配置为 PWM1 和 PWM2 模式时起作用。

16.3.9.2. 编码器接口模式

选择编码器接口模式的方法是: 如果计数器只在 TI2 的边沿计数, 则置 TIM1_SMCR 寄存器中的 SMS=001; 如果只在 TI1 边沿计数, 则置 SMS=010; 如果计数器同时在 TI1 和 TI2 边沿计数, 则置 SMS=011。通过设置 TIM1_CCER 寄存器中的 CC1P 和 CC2P 位, 可以选择 TI1 和 TI2 极性; 如果需要, 还可以对输入滤波器编程。

两个输入 TI1 和 TI2 被用来作为增量编码器的接口。参看表 73, 假定计数器已经启动(TIM1_CR1 寄存器中的 CEN=1), 则计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号; 如果没有滤波和变相, 则 TI1FP1=TI1, TI2FP2=TI2。根据两个输入信号的跳变顺序, 产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序, 计数器向上或向下计数, 同时硬件对 TIM1_CR1 寄存器的 DIR 位进行相应的设置。不管计数器是依靠 TI1 计数、依靠 TI2 计数或者同时依靠 TI1 和 TI2 计数, 在任一输入端(TI1 或者 TI2)的跳变都会重新计算 DIR 位。

编码器接口模式基本上相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIM1_ARR 寄存器的自动装载值之间连续计数(根据方向, 或是 0 到 ARR 计数, 或是 ARR 到 0 计数)。所以在开始计数之前必须配置 TIM1_ARR; 同样, 捕获器、比较器、预分频器、重复计数器、触发输出特性等仍工作如常。编码器模式和外部时钟模式 2 不兼容, 因此不能同时操作。

在这个模式下, 计数器依照增量编码器的速度和方向被自动的修改, 因此计数器的内容始终指示着编码器

的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合，假设 TI1 和 TI2 不同时变换。

表120. 计数方向与编码器信号的关系

有效边沿	相对信号的电平 (TI1FP1对应TI2, TI2FP2对应TI1)	TI1FP1信号		TI2FP2信号	
		上升	下降	上升	下降
仅在TI1计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在TI2计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在TI1和TI2上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

一个外部的增量编码器可以直接与 MCU 连接而不需要外部接口逻辑。但是，一般会使用比较器将编码器的差动输出转换到数字信号，这大大增加了抗噪声干扰能力。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发一个计数器复位。

下图是一个计数器操作的实例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。在这个例子中，我们假定配置如下：

- CC1S='01' (TIM1_CCMR1 寄存器, IC1FP1 映射到 TI1)
- CC2S='01' (TIM1_CCMR2 寄存器, IC2FP2 映射到 TI2)
- CC1P='0' (TIM1_CCER 寄存器, IC1FP1 不反相, IC1FP1=TI1)
- CC2P='0' (TIM1_CCER 寄存器, IC2FP2 不反相, IC2FP2=TI2)
- SMS='011' (TIM1_SMCR 寄存器, 所有的输入均在上升沿和下降沿有效).
- CEN='1' (TIM1_CR1 寄存器, 计数器使能)

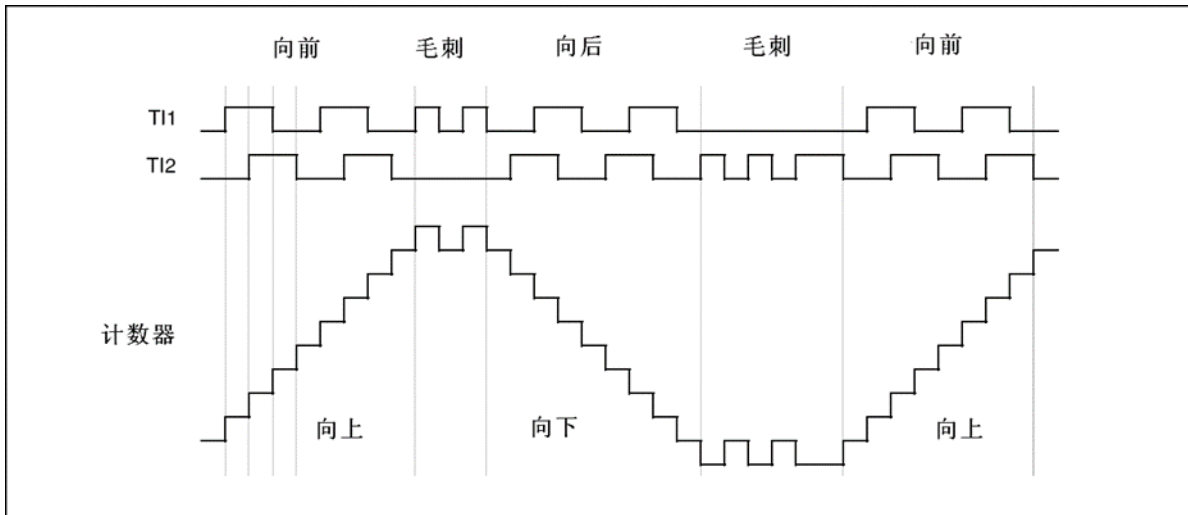


图100. 编码器模式下的计数器操作实例

下图为当 IC1FP1 极性反相时计数器的操作实例(CC1P='1', 其他配置与上例相同)。

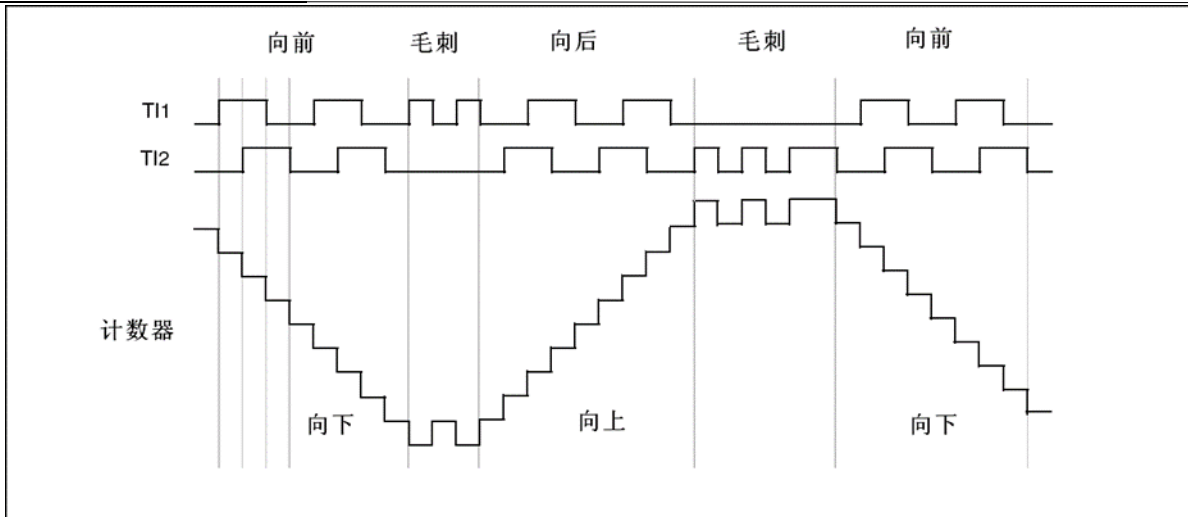


图101. IC1FP1 反相的编码器接口模式实例

当定时器配置成编码器接口模式时，提供传感器当前位置的信息。使用第二个配置在捕获模式的定时器，可以测量两个编码器事件的间隔，获得动态的信息(速度，加速度，减速度)。指示机械零点的编码器输出可被用做此目的。根据两个事件间的间隔，可以按照固定的时间读出计数器。如果可能的话，你可以把计数器的值锁存到第三个输入捕获寄存器(捕获信号必须是周期的并且可以由另一个定时器产生)；也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

16.3.9.3. 定时器输入异或功能

TIM1_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIM1_CH1、TIM1_CH2 和 TIM1_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。下节 13.3.18 给出了此特性用于连接霍尔传感器的例子。

16.3.9.4. 与霍尔传感器的接口

使用高级控制定时器 TIM1 产生 PWM 信号驱动马达时，可以用另一个通用 TIM2 定时器作为“接口定时器”来连接霍尔传感器，见下图，3 个定时器输入脚(CC1、CC2、CC3)通过一个异或门连接到 TI1 输入通道(通过设置 TIM1_CR2 寄存器中的 TI1S 位来选择)，“接口定时器”捕获这个信号。

从模式控制器被配置于复位模式，从输入是 TI1F_ED。每当 3 个输入之一变化时，计数器重新从 0 开始计数。这样产生一个由霍尔输入端的任何变化而触发的时间基准。

“接口定时器”上的捕获/比较通道 1 配置为捕获模式，捕获信号为 TRC。捕获值反映了两个输入变化间的时间延迟，给出了马达速度的信息。

“接口定时器”可以用来在输出模式产生一个脉冲，这个脉冲可以(通过触发一个 COM 事件)用于改变高级定时器 TIM1 各个通道的属性，而高级控制定时器产生 PWM 信号驱动马达。因此“接口定时器”通道必须编程为在一个指定的延时(输出比较或 PWM 模式)之后产生一个正脉冲，这个脉冲通过 TRGO 输出被送到高级控制定时器 TIM1。

举例：霍尔输入连接到 TIM1 定时器，要求每次任一霍尔输入上发生变化之后的一个指定的时刻，改变高级控制定时器 TIM1 的 PWM 配置。

- 置 TIM1_CR2 寄存器的 TI1S 位为'1'，配置三个定时器输入逻辑或到 TI1 输入，
- 时基编程：置 TIM1_ARR 为其最大值(计数器必须通过 TI1 的变化清零)。设置预分频器得到一个最大的计数器周期，它长于传感器上的两次变化的时间间隔。
- 设置通道 1 为捕获模式(选中 TRC)：置 TIM1_CCMR1 寄存器中 CC1S=01，如果需要，还可以设置数字滤波器。
- 设置通道 2 为 PWM2 模式，并具有要求的延时：置 TIM1_CCMR1 寄存器中的 OC2M=111 和 CC2S=00。
- 选择 OC2REF 作为 TRGO 上的触发输出：置 TIM1_CR2 寄存器中的 MMS=101。

在高级控制寄存器 TIM1 中，正确的 ITR 输入必须是触发器输入，定时器被编程为产生 PWM 信号，捕获/比较控制信号为预装载的(TIM1_CR2 寄存器中 CCPC=1)，同时触发输入控制 COM 事件(TIM1_CR2 寄存器中 CCUS=1)。在一次 COM 事件后，写入下一步的 PWM 控制位(CCxE、OCxM)，这可以在处理 OC2REF 上升沿的中断子程序里实现。

下图显示了这个实例：

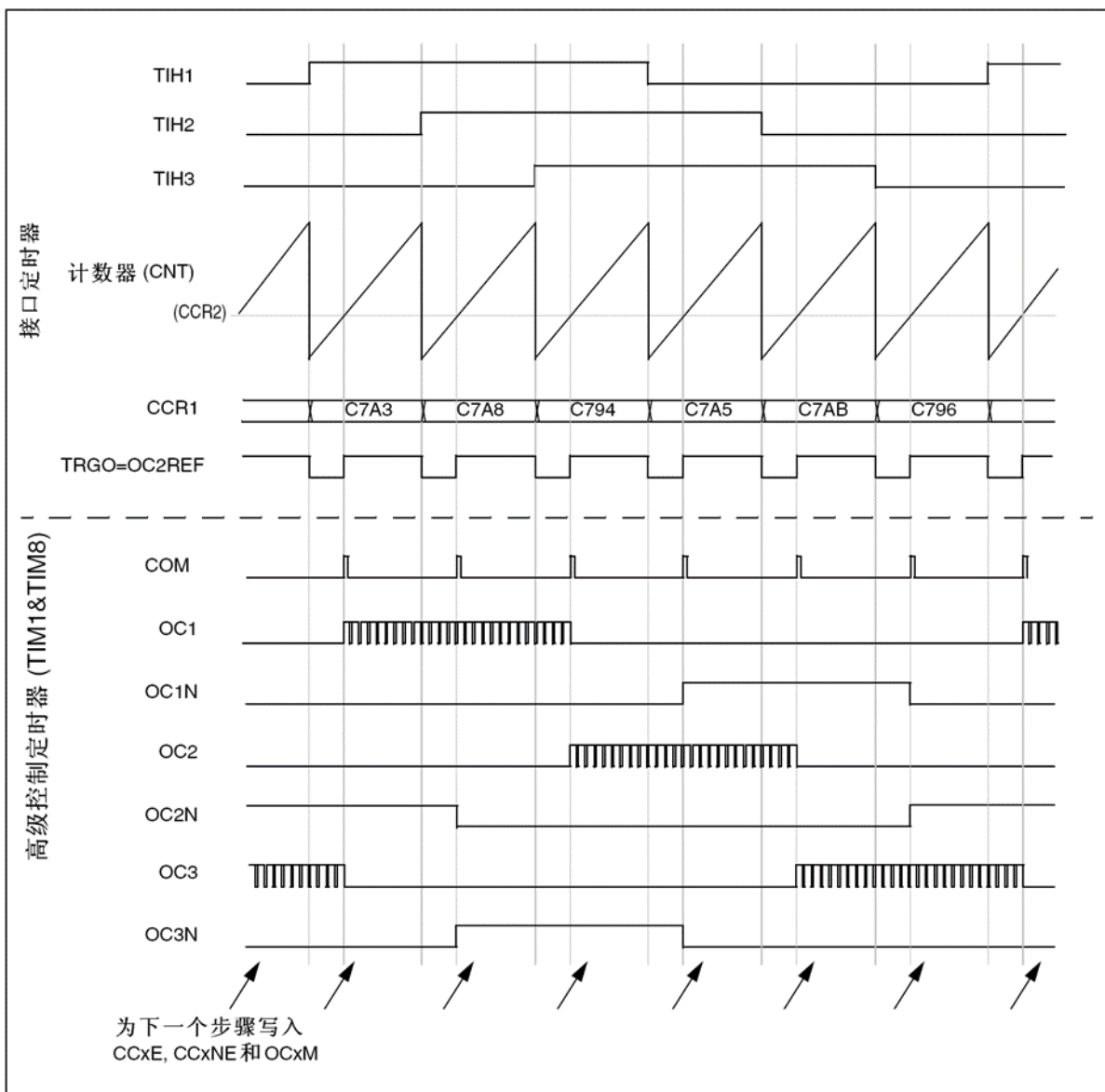


图102. 霍尔传感器接口的实例

16.3.10. TIM1 定时器和外部触发的同步

TIM1 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

从模式：复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIM1_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器(TIM1_ARR, TIM1_CCRx)都被更新了。

在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽(在本例中，不需要任何滤波器，因此保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位只选择输入捕获源，即 TIM1_CCMR1 寄存器中 CC1S=01。置 TIM1_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)。
- 置 TIM1_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIM1_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIM1_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志(TIM1_SR 寄存器中的 TIF 位)被设置，根据 TIM1_DIER 寄存器中 TIE(中断使能)位和 TDE(DMA 使能)位的设置，产生一个中断请求或一个 DMA 请求。

下图显示当自动重装载寄存器 TIM1_ARR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时取决于 TI1 输入端的重同步电路。

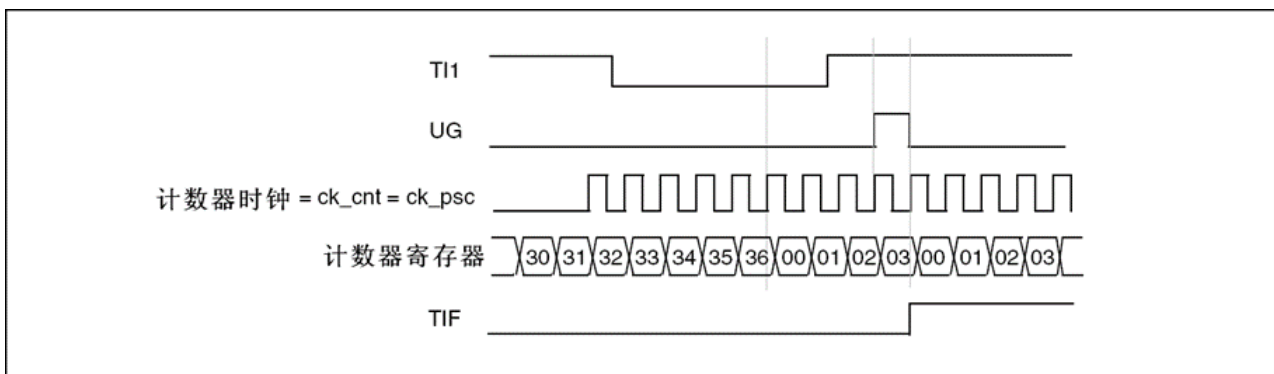


图103. 复位模式下的控制电路

从模式：门控模式

按照选中的输入端电平使能计数器。

在如下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽(本例中，不需要滤波，所以保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。CC1S 位用于选择输入捕获源，置 TIM1_CCMR1 寄存器中 CC1S=01。置 TIM1_CCER 寄存器中 CC1P=1 以确定极性(只检测低电平)。
- 置 TIM1_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIM1_SMCR 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIM1_CR1 寄存器中 CEN=1，启动计数器。在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何。

只要 TI1 为低，计数器开始依据内部时钟计数，一旦 TI1 变高则停止计数。当计数器开始或停止时都设置 TIM1_SR 中的 TIF 标置。

TI1 上升沿和计数器实际停止之间的延时取决于 TI1 输入端的重同步电路。

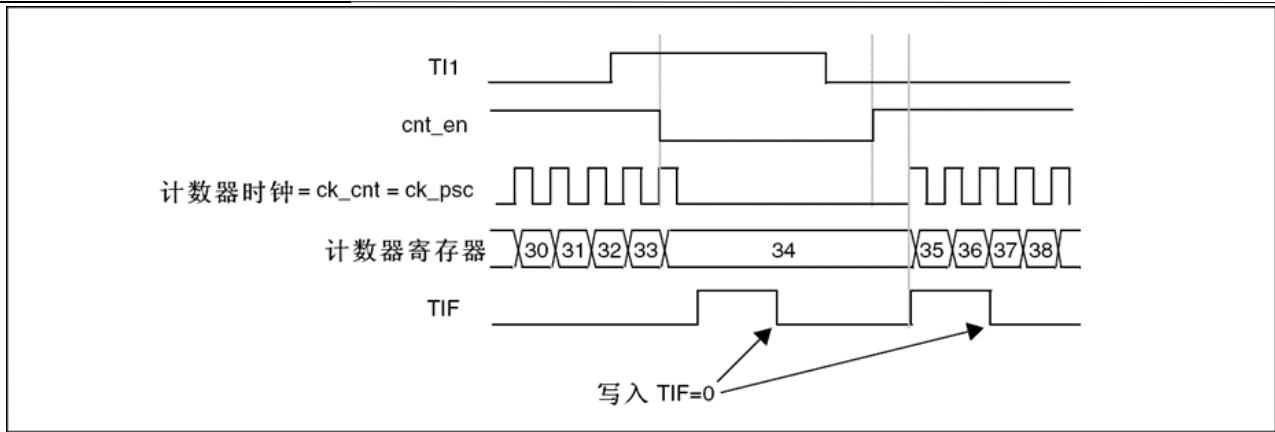


图104. 门控模式下的控制电路

从模式：触发模式

输入端上选中的事件使能计数器。

在下面的例子中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2S 位只用于选择输入捕获源，置 TIM1_CCMR1 寄存器中 CC2S=01。置 TIM1_CCER 寄存器中 CC2P=1 以确定极性(只检测低电平)。
- 置 TIM1_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIM1_SMCR 寄存器中 TS=110，选择 TI2 作为输入源。

当 TI2 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 TIF 标志。

TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

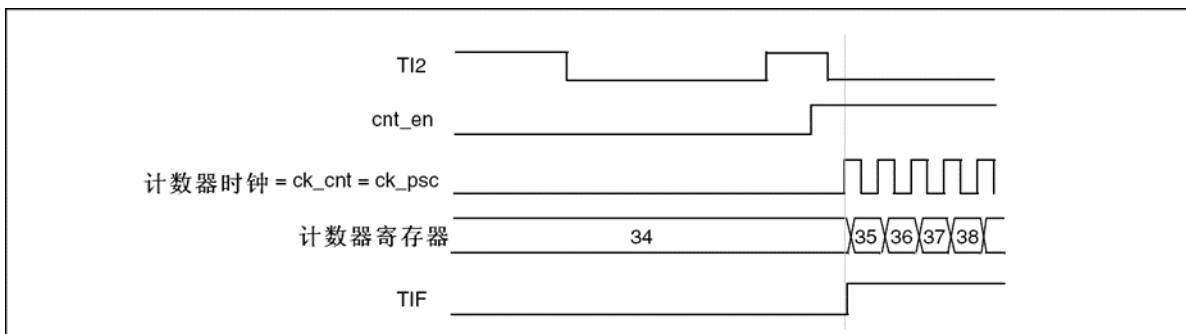


图105. 触发器模式下的控制电路

从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可以与另一种从模式(外部时钟模式 1 和编码器模式除外)一起使用。这时，ETR 信号被用作外部时钟的输入，在复位模式、门控模式或触发模式可以选择另一个输入作为触发输入。不建议使用 TIM1_SMCR 寄存器的 TS 位选择 ETR 作为 TRGI。

在下面的例子中，一旦在 TI1 上出现一个上升沿，计数器即在 ETR 的每一个上升沿向上计数一次：

1. 通过 TIM1_SMCR 寄存器配置外部触发输入电路：
 - ETF=0000：没有滤波
 - ETPS=00：不用预分频器
 - ETP=0：检测 ETR 的上升沿，置 ECE=1 使能外部时钟模式 2。
2. 按如下配置通道 1，检测 TI 的上升沿：

- IC1F=0000: 没有滤波
- 触发操作中不使用捕获预分频器, 不需要配置
- 置 TIM1_CCMR1 寄存器中 CC1S=01, 选择输入捕获源
- 置 TIM1_CCER 寄存器中 CC1P=0 以确定极性(只检测上升沿)
- 3. 置 TIM1_SMCR 寄存器中 SMS=110, 配置定时器为触发模式。置 TIM1_SMCR 寄存器中 TS=101, 选择 TI1 作为输入源。

当 TI1 上出现一个上升沿时, TIF 标志被设置, 计数器开始在 ETR 的上升沿计数。ETR 信号的上升沿和计数器实际复位间的延时, 取决于 ETRP 输入端的重同步电路。

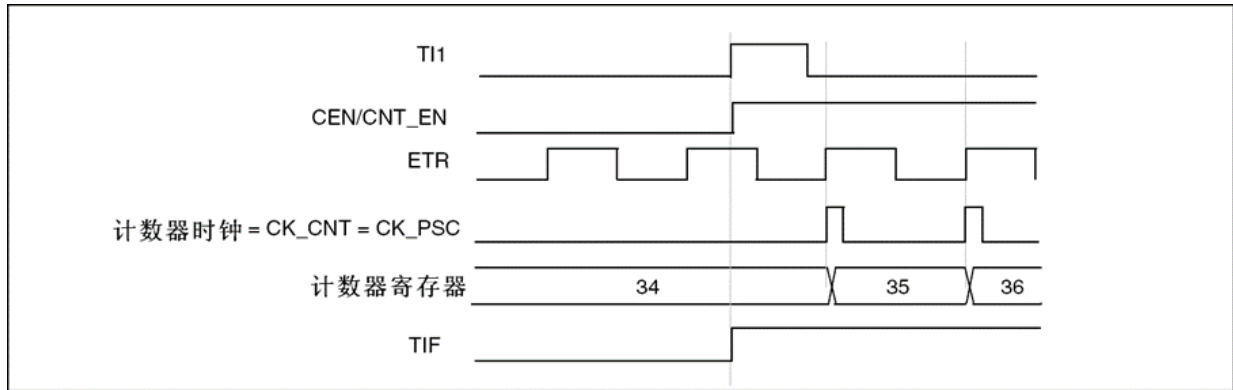


图106. 外部时钟模式 2 + 触发模式下的控制电路

16.3.11. 定时器同步

Timer1 和 Timer2 的内部连接关系如下表

表121.

从定时器	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM1	0	TIM2	0	0
TIM2	TIM1	0	0	0

16.4. TIM1 寄存器描述

Timer1 的基址为 0x40012C00

下表详述了 Timer1 的寄存器

Register Name: TIM1_CR1, Address: 0x40012C00, Default: 0x0

表122. 控制寄存器 1

Bit Name	Position	Description	Default	Access
CEN	[0]	使能计数器 (Counter enable) 0: 禁止计数器; 1: 使能计数器。 注: 在软件设置了 CEN 位后, 外部时钟、门控模式和编码器模式才能工作。 触发模式可以自动地通过硬件设置 CEN 位。	0x0	RW
UDIS	[1]	禁止更新 (Update disable) 软件通过该位允许/禁止 UEV 事件的产生 0: 允许 UEV。更新(UEV)事件由下述任一事件产生: -计数器溢出/下溢 -设置 UG 位 -从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。(更新影子寄存器)	0x0	RW

Bit Name	Position	Description	Default	Access
		1: 禁止 UEV。不产生更新事件，影子寄存器(ARR、PSC、CCRx)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化		
URS	[2]	更新请求源 (Update request source) 软件通过该位选择 UEV 事件的源 0: 如果使能了更新中断或 DMA 请求, 则下述任一事件产生更新中断或 DMA 请求: -计数器溢出/下溢 -设置 UG 位 -从模式控制器产生的更新 1: 如果使能了更新中断或 DMA 请求, 则只有计数器溢出/下溢才产生更新中断或 DMA 请求。	0x0	RW
OPM	[3]	单脉冲模式 (One pulse mode) 0: 在发生更新事件时, 计数器不停止; 1: 在发生下一次更新事件(清除 CEN 位)时, 计数器停止。	0x0	RW
DIR	[4]	方向 (Direction), 注: 当计数器配置为中央对齐模式或编码器模式时, 该位为只读。 0: 计数器向上计数; 1: 计数器向下计数。	0x0	RW
CMS	[5:6]	选择中央对齐模式 (Center-aligned mode selection) 00: 边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。 01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道 (TIM1_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 只在计数器向下计数时被设置。 10: 中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道 (TIM1_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 只在计数器向上计数时被设置。 11: 中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道 (TIM1_CCMRx 寄存器中 CCxS=00)的输出比较中断标志位, 在计数器向上和向下计数时均被设置。 注: 在计数器开启时(CEN=1), 不允许从边沿对齐模式转换到中央对齐模式。	0x0	RW
ARPE	[7]	自动重载预装载允许位 (Auto-reload preload enable) 0: TIM1_ARR 寄存器没有缓冲; 1: TIM1_ARR 寄存器被装入缓冲器。	0x0	RW
CKD	[8:9]	时钟分频因子 (Clock division) 这 2 位定义在定时器时钟(CK_INT)频率、死区时间和由死区发生器与数字滤波器(ETR,Tlx)所用的采样时钟之间的分频比例。 00: $t_{DTS} = t_{CK_INT}$ 01: $t_{DTS} = 2 \times t_{CK_INT}$ 10: $t_{DTS} = 4 \times t_{CK_INT}$ 11: 保留, 不要使用这个配置	0x0	RW

Register Name: TIM1_CR2, Address: 0x40012C04, Default: 0x0

表123. 控制寄存器 2

Bit Name	Position	Description	Default	Access
CCPC	[0]	捕获/比较预装载控制位 (Capture/compare preloaded control) 0: CCxE, CCxNE 和 OCxM 位不是预装载的; 1: CCxE, CCxNE 和 OCxM 位是预装载的; 设置该位后, 它们只在设置了 COM 位后被更新。 注: 该位只对具有互补输出的通道起作用。	0x0	RW
Reserved	[1]	保留, 始终读为 0。	0x0	R
CCUS	[2]	捕获/比较控制更新选择 (Capture/compare control update selection) 0: 如果捕获/比较控制位是预装载的(CCPC=1), 只能通过设置 COM 位更新它们; 1: 如果捕获/比较控制位是预装载的(CCPC=1), 可以通过设置 COM 位或 TRGI 上的一个上升沿更新它们。 注: 该位只对具有互补输出的通道起作用。	0x0	RW
CCDS	[3]	捕获/比较的 DMA 选择 0: 当发生 CCx 事件时, 送出 CCx 的 DMA 请求; 1: 当发生更新事件时, 送出 CCx 的 DMA 请求。	0x0	RW
MMS	[4:6]	主模式选择 (Master mode selection) 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO) 0: 复位 – TIM2_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式), 则 TRGO 上的信号相对实际的复位会有一个延迟。 1: 使能 – 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。 当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟, 除非选择了主/从模式(见 TIM2_SMCR 寄存器中 MSM 位的描述)。 2: 更新 – 更新事件被选为触发输入(TRGO)。例如, 一个主定时器的时钟可以被用作一个从定时器的预分频器 3: 比较脉冲 – 在发生一次捕获或一次比较成功时, 当要设置 CC1IF 标志时(即使它已经为高), 触发输出送出一个正脉冲(TRGO)。 4: 比较 – OC1REF 信号被用于作为触发输出(TRGO)。 5: 比较 – OC2REF 信号被用于作为触发输出(TRGO)。 6: 比较 – OC3REF 信号被用于作为触发输出(TRGO)。 7: 比较 – OC4REF 信号被用于作为触发输出(TRGO)。	0x0	RW
TI1S	[7]	TI1 选择 0: TIM2_CH1 引脚连到 TI1 输入 1: TIM2_CH1、TIM2_CH2 和 TIM2_CH3 引脚经异或后连到 TI1 输入。	0x0	RW
OIS1	[8]	输出空闲状态 1(OC1 输出) (Output Idle state 1) 0: 当 MOE=0 时, 如果实现了 OC1N, 则死区后 OC1=0; 1: 当 MOE=0 时, 如果实现了 OC1N, 则死区后 OC1=1。 注: 已经设置了 LOCK(TIM1_BKR 寄存器)级别 1、2 或 3 后, 该位不能被修改。	0x0	RW
OIS1N	[9]	输出空闲状态 1(OC1N 输出) (Output Idle state 1) 0: 当 MOE=0 时, 死区后 OC1N=0; 1: 当 MOE=0 时, 死区后 OC1N=1。	0x0	RW

Bit Name	Position	Description	Default	Access
		注: 已经设置了 LOCK(TIM1_BKR 寄存器)级别 1、2 或 3 后, 该位不能被修改。		
OIS2	[10]	输出空闲状态 2(OC2 输出)。参见 OIS1 位。	0x0	RW
OIS2N	[11]	输出空闲状态 2(OC2N 输出)。参见 OIS1N 位。	0x0	RW
OIS3	[12]	输出空闲状态 3(OC3 输出)。参见 OIS1 位。	0x0	RW
OIS3N	[13]	输出空闲状态 3(OC3N 输出)。参见 OIS1N 位。	0x0	RW
OIS4	[14]	输出空闲状态 4(OC4 输出)。参见 OIS1 位。	0x0	RW
Reserved	[15]	保留, 始终读为 0。	0x0	R

Register Name: TIM1_SMCR, Address: 0x40012C08, Default: 0x0

表124. 从模式控制寄存器

Bit Name	Position	Description	Default	Access
SMS	[0:2]	从模式选择 (Slave mode selection) 当选择了外部信号, 触发信号(TRGI)的有效边沿与选中的外部输入极性相关 (见输入控制寄存器和控制寄存器的说明) 注: 如果 TI1F_EN 被选为触发输入(TS=100)时, 不要使用门控模式。这是因为, TI1F_ED 在每次 TI1F 变化时输出一个脉冲, 然而门控模式是要检查触发输入的电平。 0: 关闭从模式 – 如果 CEN=1, 则预分频器直接由内部时钟驱动 1: 编码器模式 1 – 根据 TI1FP1 的电平, 计数器在 TI2FP2 的边沿向上/下计数 2: 编码器模式 2 – 根据 TI2FP2 的电平, 计数器在 TI1FP1 的边沿向上/下计数。 3: 编码器模式 3 – 根据另一个信号的输入电平, 计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数 4: 复位模式 – 选中的触发输入(TRGI)的上升沿重新初始化计数器, 并且产生一个更新寄存器的信号。 5: 门控模式 – 当触发输入(TRGI)为高时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止(但不复位)。计数器的启动和停止都是受控的。 6: 触发模式 – 计数器在触发输入 TRGI 的上升沿启动(但不复位), 只有计数器的启动是受控的。 7: 外部时钟模式 1 – 选中的触发输入(TRGI)的上升沿驱动计数器。	0x0	RW
TS	[4:6]	触发选择 (Trigger selection) 这 3 位选择用于同步计数器的触发输入。 注: 这些位只能在未用到(如 SMS=000)时被改变, 以避免在改变时产生错误的边沿检测。 0: 内部触发 0(ITR0), TIM1 1: 内部触发 1(ITR1), TIM2 2: 内部触发 2(ITR2), TIM3 3: 内部触发 3(ITR3), TIM4 4: TI1 的边沿检测器(TI1F_ED) 5: 滤波后的定时器输入 1(TI1FP1) 6: 滤波后的定时器输入 2(TI2FP2) 7: 外部触发输入(ETRF)	0x0	RW
MSM	[7]	主/从模式 (Master/slave mode) 0: 无作用 1: 触发输入(TRGI)上的事件被延迟了, 以允许在当前定时器(通过 TRGO)与它	0x0	RW

Bit Name	Position	Description	Default	Access
		的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。		
ETF	[8:11]	外部触发滤波 (External trigger filter) 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上, 数字滤波器是一个事件计数器, 它记录到 N 个事件后会产生一个输出的跳变 0: 无滤波器, 以 fDTS 采样 1: 采样频率 fSAMPLING=fCK_INT, N=2 2: 采样频率 fSAMPLING=fCK_INT, N=4 3: 采样频率 fSAMPLING=fCK_INT, N=8 4: 采样频率 fSAMPLING=fDTS/2, N=6 5: 采样频率 fSAMPLING=fDTS/2, N=8 6: 采样频率 fSAMPLING=fDTS/4, N=6 7: 采样频率 fSAMPLING=fDTS/4, N=8 8: 采样频率 fSAMPLING=fDTS/8, N=6 9: 采样频率 fSAMPLING=fDTS/8, N=8 10: 采样频率 fSAMPLING=fDTS/16, N=5 11: 采样频率 fSAMPLING=fDTS/16, N=6 12: 采样频率 fSAMPLING=fDTS/16, N=8 13: 采样频率 fSAMPLING=fDTS/32, N=5 14: 采样频率 fSAMPLING=fDTS/32, N=6 15: 采样频率 fSAMPLING=fDTS/32, N=8	0x0	RW
ETPS	[12:13]	外部触发预分频 (External trigger prescaler) 外部触发信号 ETRP 的频率必须最多是 CK_INT 频率的 1/4。当输入较快的外部时钟时, 可以使用预分频降低 ETRP 的频率。 0: 关闭预分频 1: ETRP 频率除以 2; 2: ETRP 频率除以 4; 3: ETRP 频率除以 8。	0x0	RW
ECE	[14]	外部时钟使能位 (External clock enable) 该位启用外部时钟模式 2,注 1: 设置 ECE 位与选择外部时钟模式 1 并将 TRGI 连到 ETRF(SMS=111 和 TS=111)具有相同功效。 注 2: 下述从模式可以与外部时钟模式 2 同时使用: 复位模式、门控模式和触发模式; 但是, 这时 TRGI 不能连到 ETRF(TS 位不能是'111')。 注 3: 外部时钟模式 1 和外部时钟模式 2 同时被使能时, 外部时钟的输入是 ETRF。 0: 禁止外部时钟模式 2; 1: 使能外部时钟模式 2。计数器由 ETRF 信号上的任意有效边沿驱动	0x0	RW
ETP	[15]	ETP: 外部触发极性 (External trigger polarity) 该位选择是用 ETR 还是 ETR 的反相来作为触发操作 0: ETR 不反相, 高电平或上升沿有效; 1: ETR 被反相, 低电平或下降沿有效。	0x0	RW

Register Name: TIM1_DIER, Address: 0x40012C0C, Default: 0x0

表125. DMA/中断使能寄存器

Bit Name	Position	Description	Default	Access
UIE	[0]	允许更新中断 (Update interrupt enable)	0x0	RW
CC1IE	[1]	允许捕获/比较 1 中断 (Capture/Compare 1 interrupt enable)	0x0	RW
CC2IE	[2]	允许捕获/比较 2 中断 (Capture/Compare 2 interrupt enable)	0x0	RW
CC3IE	[3]	允许捕获/比较 3 中断 (Capture/Compare 2 interrupt enable)	0x0	RW
CC4IE	[4]	允许捕获/比较 4 中断 (Capture/Compare 2 interrupt enable)	0x0	RW
TIE	[6]	触发中断使能 (Trigger interrupt enable)	0x0	RW
UDE	[8]	允许更新的 DMA 请求 (Update DMA request enable)	0x0	RW
CC1DE	[9]	允许捕获/比较 1 的 DMA 请求 (Capture/Compare 1 DMA request enable)	0x0	RW
CC2DE	[10]	允许捕获/比较 2 的 DMA 请求 (Capture/Compare 2 DMA request enable)	0x0	RW
CC3DE	[11]	允许捕获/比较 3 的 DMA 请求 (Capture/Compare 3 DMA request enable)	0x0	RW
CC4DE	[12]	允许捕获/比较 4 的 DMA 请求 (Capture/Compare 4 DMA request enable)	0x0	RW
TDE	[14]	允许触发 DMA 请求 (Trigger DMA request enable)	0x0	RW

Register Name: TIM1_SR, Address: 0x40012C10, Default: 0x0

表126. 状态寄存器

Bit Name	Position	Description	Default	Access
UIF	[0]	更新中断标记 (Update interrupt flag) 当产生更新事件时该位由硬件置'1'。它由软件清'0'。 -若 TIM2_CR1 寄存器的 UDIS=0、URS=0, 当 TIM2_EGR 寄存器的 UG=1 时产生更新事件 (软件对计数器 CNT 重新初始化); -若 TIM2_CR1 寄存器的 UDIS=0、URS=0, 当计数器 CNT 被触发事件重初始化时产生更新事件。	0x0	RW
CC1IF	[1]	捕获/比较 1 中断标记	0x0	RW
CC2IF	[2]	捕获/比较 2 中断标记	0x0	RW
CC3IF	[3]	捕获/比较 3 中断标记	0x0	RW
CC4IF	[4]	捕获/比较 4 中断标记	0x0	RW
TIF	[6]	触发器中断标记,仅当相应的通道被配置为输入捕获时, 该标记可由硬件置'1'。写'0'可清除该位。	0x0	RW
CC1OF	[9]	捕获/比较 1 重复捕获标记 (Capture/Compare 1 overcapture flag) 仅当相应的通道被配置为输入捕获时, 该标记可由硬件置'1'。写'0'可清除该位。	0x0	RW
CC2OF	[10]	捕获/比较 2 重复捕获标记,仅当相应的通道被配置为输入捕获时, 该标记可由硬件置'1'。写'0'可清除该位。	0x0	RW
CC3OF	[11]	捕获/比较 2 重复捕获标记,仅当相应的通道被配置为输入捕获时, 该标记可由硬件置'1'。写'0'可清除该位。	0x0	RW
CC4OF	[12]	捕获/比较 2 重复捕获标记,仅当相应的通道被配置为输入捕获时, 该标记可由硬件置'1'。写'0'可清除该位。	0x0	RW

Register Name: TIM1_EGR, Address: 0x40012C14, Default: 0x0

表127. 事件产生寄存器

Bit Name	Position	Description	Default	Access
UG	[0]	产生更新事件 (Update generation) 该位由软件置'1', 由硬件自动清'0'。 0: 无动作;	0x0	RW

Bit Name	Position	Description	Default	Access
		1: 重新初始化计数器, 并产生一个更新事件。注意预分频器的计数器也被清'0'(但是预分频系数不变)。若在中心对称模式下或 DIR=0(向上计数)则计数器被清'0', 若 DIR=1(向下计数)则计数器取 TIM2_ARR 的值。		
CC1G	[1]	产生捕获/比较 1 事件 (Capture/compare 1 generation) 该位由软件置'1', 用于产生一个捕获/比较事件, 由硬件自动清'0'。 置 CC1IF=1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA。 若通道 CC1 配置为输入: 当前的计数器值捕获至 TIM2_CCR1 寄存器; 设置 CC1IF=1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA。若 CC1IF 已经为 1, 则设置 CC1OF=1。 0: 无动作; 1: 在通道 CC1 上产生一个捕获/比较事件: 若通道 CC1 配置为输出:	0x0	RW
CC2G	[2]	产生捕获/比较 2 事件	0x0	RW
CC3G	[3]	产生捕获/比较 3 事件	0x0	RW
CC4G	[4]	产生捕获/比较 4 事件	0x0	RW
TG	[6]	产生触发事件 (Trigger generation) 该位由软件置'1', 用于产生一个触发事件, 由硬件自动清'0'。 0: 无动作; 1: TIM2_SR 寄存器的 TIF=1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA。	0x0	RW
BG	[7]	产生刹车事件 (Break generation) 该位由软件置'1', 用于产生一个刹车事件, 由硬件自动清'0'。 0: 无动作; 1: 产生一个刹车事件。此时 MOE=0、BIF=1, 若开启对应的中断和 DMA, 则产生相应的中断和 DMA。	0x0	RW

Register Name: TIM1_CCMR1, Address: 0x40012C18, Default: 0x0

通道可用于输入(捕获模式)或输出(比较模式), 通道的方向由相应的 CCxS 定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能, ICxx 描述了通道在输出模式下的功能。因此必须注意, 同一个位在输出模式和输入模式下的功能是不同的。

表128. 输出比较模式:

Bit Name	Position	Description	Default	Access
CC1S	[0:1]	捕获/比较 1 选择 (Capture/Compare 1 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: CC1S 仅在通道关闭时(TIM2_CCER 寄存器的 CC1E='0')才是可写的。 0: CC1 通道被配置为输出; 1: CC1 通道被配置为输入, IC1 映射在 TI1 上; 2: CC1 通道被配置为输入, IC1 映射在 TI2 上; 3: CC1 通道被配置为输入, IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。	0x0	RW
OC1FE	[2]	输出比较 1 快速使能 (Output compare 1 fast enable) 该位用于加快 CC 输出对触发器输入事件的响应。 该位只在通道被配置成 PWM1 或 PWM2 模式时起作用。配置为输入捕获时做预分频器	0x0	RW

Bit Name	Position	Description	Default	Access
		0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入出现一个有效沿时, 激活 CC1 输出的最小延时为 5 个时钟周期。 1: 输入到触发器的有效沿的作用就象发生了一次比较匹配。因此, OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和 CC1 输出间的延时被缩短为 3 个时钟周期。		
OC1PE	[3]	输出比较 1 预装载使能。注 1: 一旦 LOCK 级别设为 3(TIM2_BDTR 寄存器中的 LOCK 位)并且 CC1S='00'(该通道配置成输出)则该位不能被修改。 注 2: 仅在单脉冲模式下(TIM2_CR1 寄存器的 OPM='1'), 可以在未确认预装载寄存器情况下使用 PWM 模式, 否则其动作不确定。 0: 禁止 TIM2_CCR1 寄存器的预装载功能, 可随时写入 TIM2_CCR1 寄存器, 并且新写入的数值立即起作用。 1: 开启 TIM2_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM2_CCR1 的预装载值在更新事件到来时被传送至当前寄存器中。	0x0	RW
OC1M	[4:6]	输出比较 1 模式 (Output compare 1 enable) 该 3 位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效, 而 OC1 的有效电平取决于 CC1P 位。 注 1: 一旦 LOCK 级别设为 3(TIM2_BDTR 寄存器中的 LOCK 位)并且 CC1S='00'(该通道配置成输出)则该位不能被修改。 注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式从中冻结模式切换到 PWM 模式时, OC1REF 电平才改变。 0: 冻结。输出比较寄存器 TIM2_CCR1 与计数器 TIM2_CNT 间的比较对 OC1REF 不起作用; 1: 匹配时设置通道 1 为有效电平。当计数器 TIM2_CNT 的值与捕获 / 比较寄存器 1(TIM2_CCR1)相同时, 强制 OC1REF 为高。 2: 匹配时设置通道 1 为无效电平。当计数器 TIM2_CNT 的值与捕获 / 比较寄存器 1(TIM2_CCR1)相同时, 强制 OC1REF 为低。 3: 翻转。当 TIM2_CCR1=TIM2_CNT 时, 翻转 OC1REF 的电平。 4: 强制为无效电平。强制 OC1REF 为低。 5: 强制为有效电平。强制 OC1REF 为高。 6: PWM 模式 1 - 在向上计数时, 一旦 TIM2_CNT<TIM2_CCR1 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 TIM2_CNT>TIM2_CCR1 时通道 1 为无效电平(OC1REF=0), 否则为有效电平(OC1REF=1)。 7: PWM 模式 2 - 在向上计数时, 一旦 TIM2_CNT<TIM2_CCR1 时通道 1 为无效电平, 否则为有效电平; 在向下计数时, 一旦 TIM2_CNT>TIM2_CCR1 时通道 1 为有效电平, 否则为无效电平。	0x0	RW
OC1CE	[7]	输出比较 1 清 0 使能 (Output compare 1 clear enable) 0: OC1REF 不受 ETRF 输入的影响; 1: 一旦检测到 ETRF 输入高电平, 清除 OC1REF=0。	0x0	RW
CC2S	[8:9]	捕获/比较 2 选择 (Capture/Compare 2 selection) 该位定义通道的方向(输入/输出), 及输入脚的选择。注: CC2S 仅在通道关闭时(TIM2_CCER	0x0	RW

Bit Name	Position	Description	Default	Access
		寄存器的 CC2E='0')才是可写的。 0: CC2 通道被配置为输出; 1: CC2 通道被配置为输入, IC2 映射在 TI2 上; 2: CC2 通道被配置为输入, IC2 映射在 TI1 上; 3: CC2 通道被配置为输入, IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。		
OC2FE	[10]	输出比较 2 快速使能	0x0	RW
OC2PE	[11]	输出比较 2 预装载使能	0x0	RW
OC2M	[12:14]	输出比较 2 模式	0x0	RW
OC2CE	[15]	输出比较 2 清 0 使能	0x0	RW

表129. 输入捕获模式:

Bit Name	Position	Description	Default	Access
CC1S	[0:1]	捕获/比较 1 选择 (Capture/Compare 1 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1 通道被配置为输出; 01: CC1 通道被配置为输入, IC1 映射在 TI1 上; 10: CC1 通道被配置为输入, IC1 映射在 TI2 上; 11: CC1 通道被配置为输入, IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC1S 仅在通道关闭时(TIM2_CCER 寄存器的 CC1E='0')才是可写的。	0x0	RW
IC1PSC	[2:3]	输入/捕获 1 预分频器 (Input capture 1 prescaler) 这 2 位定义了 CC1 输入(IC1)的预分频系数。 一旦 CC1E='0'(TIM2_CCER 寄存器中), 则预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获; 01: 每 2 个事件触发一次捕获; 10: 每 4 个事件触发一次捕获; 11: 每 8 个事件触发一次捕获。	0x0	RW
IC1F	[4:7]	输入捕获 1 滤波器 (Input capture 1 filter) 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成, 它记录到 N 个事件后会产生一个输出的跳变: 0000: 无滤波器, 以 fDTS 采样 1000 : 采样频率 fSAMPLING=fDTS/8, N=6 0001: 采样频率 fSAMPLING=fCK_INT, N=2 1001 : 采样频率 fSAMPLING=fDTS/8, N=8 0010: 采样频率 fSAMPLING=fCK_INT, N=4 1010 : 采样频率 fSAMPLING=fDTS/16, N=5 0011: 采样频率 fSAMPLING=fCK_INT, N=8 1011 : 采样频率 fSAMPLING=fDTS/16, N=6 0100: 采样频率 fSAMPLING=fDTS/2, N=6 1100 : 采样频率 fSAMPLING=fDTS/16, N=8 0101: 采样频率 fSAMPLING=fDTS/2, N=8 1101 : 采样频率 fSAMPLING=fDTS/32, N=5 0110: 采样频率 fSAMPLING=fDTS/4, N=6 1110 : 采样频率 fSAMPLING=fDTS/32, N=6	0x0	RW

Bit Name	Position	Description	Default	Access
		0111: 采样频率 fSAMPLING=fDTS/4, N=8 1111: 采样频率 fSAMPLING=fDTS/32, N=8 注: 在现在的芯片版本中, 当 ICx[3:0]=1、2 或 3 时, 公式中的 fDTS 由 CK_INT 替代。		
CC2S	[8:9]	捕获/比较 2 选择 (Capture/compare 2 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC2 通道被配置为输出; 01: CC2 通道被配置为输入, IC2 映射在 TI2 上; 10: CC2 通道被配置为输入, IC2 映射在 TI1 上; 11: CC2 通道被配置为输入, IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC2S 仅在通道关闭时(TIM2_CCER 寄存器的 CC2E='0')才是可写的。	0x0	RW
IC2PSC	[10:11]	输入/捕获 2 预分频器 (input capture 2 prescaler)	0x0	RW
IC2F	[15:12]	输入捕获 2 滤波器 (Input capture 2 filter)	0x0	RW

Register Name: TIM1_CCMR2, Address: 0x40012C1C, Default: 0x0

捕获/比较模式寄存器 2(输出比较模式,通道的方向由相应的 CCxS 定义), 参看以上 CCMR1 寄存器的描述。
表130. 输出比较模式:

Bit Name	Position	Description	Default	Access
CC3S	[0:1]	捕获/比较 3 选择 (Capture/Compare 3 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 0: CC3 通道被配置为输出; 1: CC3 通道被配置为输入, IC1 映射在 TI1 上; 2: CC3 通道被配置为输入, IC1 映射在 TI2 上; 3: CC3 通道被配置为输入, IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。	0x0	RW
OC3FE	[2]	输出比较 3 快速使能	0x0	RW
OC3PE	[3]	输出比较 3 预装载使能	0x0	RW
OC3M	[4:6]	输出比较 3 模式	0x0	RW
OC3CE	[7]	输出比较 3 清 0 使能	0x0	RW
CC4S	[8:9]	捕获/比较 4 选择 (Capture/Compare 4 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 0: CC4 通道被配置为输出; 1: CC4 通道被配置为输入, IC4 映射在 TI4 上; 2: CC4 通道被配置为输入, IC4 映射在 TI3 上; 3: CC4 通道被配置为输入, IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。	0x0	RW
OC4FE	[10]	输出比较 4 快速使能	0x0	RW
OC4PE	[11]	输出比较 4 预装载使能	0x0	RW
OC4M	[12:14]	输出比较 4 模式	0x0	RW
OC4CE	[15]	输出比较 4 清 0 使能	0x0	RW

表131. 输入捕获模式:

Bit Name	Position	Description	Default	Access
CC3S	[0:1]	捕获/比较 3 选择 (Capture/Compare 3 selection) 这 2 位定义通道的	0x0	RW

		方向(输入/输出), 及输入脚的选择: 00: CC3 通道被配置为输出; 01: CC3 通道被配置为输入, IC3 映射在 TI3 上; 10: CC3 通道被配置为输入, IC3 映射在 TI4 上; 11: CC3 通道被配置为输入, IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC3S 仅在通道关闭时(TIM2_CCER 寄存器的 CC3E='0')才是可写的。		
IC3PSC	[2:3]	输入/捕获 3 预分频器 (Input capture 3 prescaler)	0x0	RW
IC3F	[4:7]	输入捕获 3 滤波器 (Input capture 3 filter)	0x0	RW
CC4S	[8:9]	捕获/比较 4 选择 (Capture/compare 4 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC4 通道被配置为输出; 01: CC4 通道被配置为输入, IC4 映射在 TI4 上; 10: CC4 通道被配置为输入, IC4 映射在 TI3 上; 11: CC4 通道被配置为输入, IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM2_SMCR 寄存器的 TS 位选择)。 注: CC4S 仅在通道关闭时(TIM2_CCER 寄存器的 CC4E='0')才是可写的。	0x0	RW
IC4PSC	[10:11]	输入/捕获 4 预分频器 (input capture 4 prescaler)	0x0	RW
IC4F	[12:15]	输入捕获 4 滤波器 (Input capture 4 filter)	0x0	RW

Register Name: TIM1_CCER, Address: 0x40012C20, Default: 0x0

表132. 捕获/比较使能寄存器

Bit Name	Position	Description	Default	Access
CC1E	[0]	输入/捕获 1 输出使能 (Capture/Compare 1 output enable) CC1 通道配置为输出: 0: 关闭 - OC1 禁止输出。 1: 开启 - OC1 信号输出到对应的输出引脚。 CC1 通道配置为输入: 该位决定了计数器的值是否能捕获入 TIM2_CCR1 寄存器。 0: 捕获禁止; 1: 捕获使能。	0x0	RW
CC1P	[1]	输入/捕获 1 输出极性 (Capture/Compare 1 output polarity) CC1 通道配置为输出: 0: OC1 高电平有效 1: OC1 低电平有效 CC1 通道配置为输入: 该位选择是 IC1 还是 IC1 的反相信号作为触发或捕获信号。 0: 不反相: 捕获发生在 IC1 的上升沿; 当用作外部触发器时, IC1 不反相。 1: 反相: 捕获发生在 IC1 的下降沿; 当用作外部触发器时, IC1 反相。	0x0	RW
CC2E	[4]	输入/捕获 2 输出使能	0x0	RW
CC2P	[5]	输入/捕获 2 输出极性	0x0	RW
CC3E	[8]	输入/捕获 3 输出使能	0x0	RW
CC3P	[9]	输入/捕获 3 输出极性	0x0	RW
CC4E	[12]	输入/捕获 4 输出使能	0x0	RW
CC4P	[13]	输入/捕获 4 输出极性	0x0	RW

表133. 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位

控制位					输出状态 ⁽¹⁾	
MOE位	OSSI位	OSSR位	CCxE位	CCxNE位	OCx 输出状态	OCxN 输出状态
1	X	0	0	0	输出禁止(与定时器断开) OCx=0, OCx_EN=0	输出禁止(与定时器断开) OCxN=0, OCxN_EN=0
		0	0	1	输出禁止(与定时器断开) OCx=0, OCx_EN=0	OCxREF + 极性, OCxN= OCxREF xor CCxNP, OCxN_EN=1
		0	1	0	OCxREF + 极性, OCx= OCxREF xor CCxP, OCx_EN=1	输出禁止(与定时器断开) OCxN=0, OCxN_EN=0
		0	1	1	OCxREF + 极性 + 死区, OCx_EN=1	OCxREF反相 + 极性 + 死区, OCxN_EN=1
		1	0	0	输出禁止(与定时器断开) OCx=CCxP, OCx_EN=0	输出禁止(与定时器断开) OCxN=CCxNP, OCxN_EN=0
		1	0	1	关闭状态(输出使能且为无效电平) OCx=CCxP, OCx_EN=1	OCxREF + 极性, OCxN= OCxREF xor CCxNP, OCxN_EN=1
		1	1	0	OCxREF + 极性, OCx= OCxREF xor CCxP, OCx_EN=1	关闭状态(输出使能且为无效电平) OCxN=CCxNP, OCxN_EN=1
		1	1	1	OCxREF + 极性 + 死区, OCx_EN=1	OCxREF反相 + 极性 + 死区, OCxN_EN=1
0	0	X	0	0	输出禁止(与定时器断开) 异步地: OCx=CCxP, OCx_EN=0, OCxN=CCxNP, OCxN_EN=0; 若时钟存在: 经过一个死区时间后 OCx=OISx, OCxN=OISxN, 假设OISx与OISxN并不都对应OCx和OCxN的有效电平。	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0	关闭状态(输出使能且为无效电平) 异步地: OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1; 若时钟存在: 经过一个死区时间后 OCx=OISx, OCxN=OISxN, 假设OISx与OISxN并不都对应OCx和OCxN的有效电平。	
	1		0	1		
	1		1	0		
	1		1	1		

1.如果一个通道的 2 个输出都没有使用(CCxE = CCxNE = 0), 那么 OISx, OISxN, CCxP 和 CCxNP 都必须清零。

注: 引脚连接到互补的 OCx 和 OCxN 通道的外部 I/O 引脚的状态, 取决于 OCx 和 OCxN 通道状态和 GPIO 以及 AFIO 寄存器。

Register Name: TIM1_CNT, Address: 0x40012C24, Default: 0x0

表134. 计数器

Bit Name	Position	Description	Default	Access
CNT	[0:15]	计数器的值	0x0	RW

Register Name: TIM1_PSC, Address: 0x40012C28, Default: 0x0

表135. 预分频器

Bit Name	Position	Description	Default	Access
PSC	[0:15]	预分频器的值 (Prescaler value) 计数器的时钟频率 CK_CNT 等于 fCK_PSC/(PSC[15:0]+1)。 PSC 包含了当更新事件产生时装入当前预分频器寄存器的值。	0x0	RW

Register Name: TIM1_ARR, Address: 0x40012C2C, Default: 0x0

表136. 自动重装载寄存器

Bit Name	Position	Description	Default	Access
ARR	[0:15]	自动重装载的值 (Auto reload value) ARR 包含了将要传至实际的自动重装载寄存器的数值。详细参考 14.3.1 节: 有关 ARR 的更新和动作。 当自动重装载的值为空时, 计数器不工作。	0x0	RW

Register Name: TIM1_RCR, Address: 0x40012C30, Default: 0x0

表137. 自动重装载寄存器

Bit Name	Position	Description	Default	Access
REP	[0:7]	重复计数器的值 (Repetition counter value) 开启了预装载功能后, 这些位允许用户设置比较寄存器的更新速率(即周期性地从预装载寄存器传输到当前寄存器); 如果允许产生更新中断, 则会同时影响产生更新中断的速率。 每次向下计数器 REP_CNT 达到 0, 会产生一个更新事件并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在周期更新事件 U_RC 发生时才重载 REP 值, 因此对 TIM1_RCR 寄存器写入的新值只在下次周期更新事件发生时才起作用。 这意味着在 PWM 模式中, (REP+1)对应着: - 在边沿对齐模式下, PWM 周期的数目; - 在中心对称模式下, PWM 半周期的数目;	0x0	RW

Register Name: TIM1_CCR1, Address: 0x40012C34, Default: 0x0

表138. 捕获/比较寄存器 1

Bit Name	Position	Description	Default	Access
CCR1	[0:15]	捕获/比较 1 的值 (Capture/Compare 1 value) 若 CC1 通道配置为输出: CCR1 包含了装入当前捕获/比较 1 寄存器的值(预装载值)。 如果在 TIM2_CCMR1 寄存器(OC1PE 位)中未选择预装载特性, 写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 1 寄存器中。 当前捕获/比较寄存器参与同计数器 TIM2_CNT 的比较, 并在 OC1 端口上产生输出信号。若 CC1 通道配置为输入: CCR1 包含了由上一次输入捕获 1 事件(IC1)传输的计数器值。	0x0	RW

Register Name: TIM1_CCR2, Address: 0x40012C38, Default: 0x0

表139. 捕获/比较寄存器 2

Bit Name	Position	Description	Default	Access
CCR2	[0:15]	捕获/比较 2 的值 (Capture/Compare 2 value) 若 CC2 通道配置为输出: CCR2 包含了装入当前捕获/比较 2 寄存器的值(预装载值)。 如果在 TIM2_CCMR2 寄存器(OC2PE 位)中未选择预装载特性, 写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 2 寄存器中。 当前捕获/比较寄存器参与同计数器 TIM2_CNT 的比较, 并在 OC2 端口上产生输出信号。若 CC2 通道配置为输入:	0x0	RW

CCR2 包含了由上一次输入捕获 2 事件(IC2)传输的计数器值。

Register Name: TIM1_CCR3, Address: 0x40012C3C, Default: 0x0

表140. 捕获/比较寄存器 3

Bit Name	Position	Description	Default	Access
CCR3	[0:15]	捕获/比较 3 的值 (Capture/Compare 3 value) 若 CC3 通道配置为输出: CCR3 包含了装入当前捕获/比较 3 寄存器的值(预装载值)。 如果在 TIM2_CCMR3 寄存器(OC3PE 位)中未选择预装载特性, 写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 3 寄存器中。 当前捕获/比较寄存器参与同计数器 TIM2_CNT 的比较, 并在 OC3 端口上产生输出信号。若 CC3 通道配置为输入: CCR3 包含了由上一次输入捕获 3 事件(IC3)传输的计数器值。	0x0	RW

Register Name: TIM1_CCR4, Address: 0x40012C40, Default: 0x0

表141. 捕获/比较寄存器 4

Bit Name	Position	Description	Default	Access
CCR4	[0:15]	捕获/比较 4 的值 (Capture/Compare 4 value) 若 CC4 通道配置为输出: CCR4 包含了装入当前捕获/比较 4 寄存器的值(预装载值)。 如果在 TIM2_CCMR4 寄存器(OC4PE 位)中未选择预装载特性, 写入的数值会被立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 4 寄存器中。 当前捕获/比较寄存器参与同计数器 TIM2_CNT 的比较, 并在 OC4 端口上产生输出信号。若 CC4 通道配置为输入: CCR4 包含了由上一次输入捕获 4 事件(IC4)传输的计数器值。	0x0	RW

Register Name: TIM1_BDTR, Address: 0x40012C44, Default: 0x0

注释: 根据锁定设置, AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0]位均可被写保护, 有必要在第一次写入 TIM1_BDTR 寄存器时对它们进行配置。

表142. 刹车和死区寄存器

Bit Name	Position	Description	Default	Access
UTG	[0:7]	死区发生器设置 (Dead-time generator setup) 这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间: $DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times Tdtg, Tdtg = Tdts;$ $DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times Tdtg, Tdtg = 2 \times Tdts;$ $DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times Tdtg, Tdtg = 8 \times Tdts;$ $DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times Tdtg, Tdtg = 16 \times Tdts;$ 例: 若 $Tdts = 125ns(8MHz)$, 可能的死区时间为: 0 到 15875ns, 若步长时间为 125ns; 16us 到 31750ns, 若步长时间为 250ns; 32us 到 63us, 若步长时间为 1us; 64us 到 126us, 若步长时间为 2us; 注: 一旦 LOCK 级别(TIM1_BDTR 寄存器中的 LOCK 位)设为 1、2 或 3, 则不能修改这些位。	0x0	RW

Bit Name	Position	Description	Default	Access
LOOK	[8:9]	锁定设置 (Lock configuration) 该位为防止软件错误而提供写保护。 00: 锁定关闭, 寄存器无写保护; 01: 锁定级别 1, 不能写入 TIM1_BDTR 寄存器的 DTG、BKE、BKP、AOE 位和 TIM1_CR2 寄存器的 OISx/OISxN 位; 10: 锁定级别 2, 不能写入锁定级别 1 中的各位, 也不能写入 CC 极性位(一旦相关通道通过 CCxS 位设为输出, CC 极性位是 TIM1_CCER 寄存器的 CCxP/CCNxP 位)以及 OSSR/OSSI 位; 11: 锁定级别 3, 不能写入锁定级别 2 中的各位, 也不能写入 CC 控制位(一旦相关通道通过 CCxS 位设为输出, CC 控制位是 TIM1_CCMRx 寄存器的 OCxM/OCxPE 位); 注: 在系统复位后, 只能写一次 LOCK 位, 一旦写入 TIM1_BDTR 寄存器, 则其内容冻结直至复位。	0x0	RW
OSSI	[10]	空闲模式下"关闭状态"选择 (Off-state selection for Idle mode) 该位用于当 MOE=0 且通道设为输出时。 参考 OC/OCN 使能的详细说明(13.4.9 节, TIM1 和 TIM8 捕获/比较使能寄存器(TIM1_CCER))。 0: 当定时器不工作时, 禁止 OC/OCN 输出(OC/OCN 使能输出信号=0); 1: 当定时器不工作时, 一旦 CCxE=1 或 CCxNE=1, OC/OCN 首先输出其空闲电平, 然后 OC/OCN 使能输出信号=1。 注: 一旦 LOCK 级别(TIM1_BDTR 寄存器中的 LOCK 位)设为 2, 则该位不能被修改。	0x0	RW
OSSR	[11]	运行模式下"关闭状态"选择 (Off-state selection for Run mode) 该位用于当 MOE=1 且通道为互补输出时。没有互补输出的定时器中不存在 OSSR 位。 参考 OC/OCN 使能的详细说明(13.4.9 节, TIM1 和 TIM8 捕获/比较使能寄存器(TIM1_CCER))。 0: 当定时器不工作时, 禁止 OC/OCN 输出(OC/OCN 使能输出信号=0); 1: 当定时器不工作时, 一旦 CCxE=1 或 CCxNE=1, 首先开启 OC/OCN 并输出无效电平, 然后置 OC/OCN 使能输出信号=1。 注: 一旦 LOCK 级别(TIM1_BDTR 寄存器中的 LOCK 位)设为 2, 则该位不能被修改。	0x0	RW
BKE	[12]	刹车功能使能 (Break enable) 0: 禁止刹车输入(BRK 及 CCS 时钟失效事件); 1: 开启刹车输入(BRK 及 CCS 时钟失效事件)。 注: 当设置了 LOCK 级别 1 时(TIM1_BDTR 寄存器中的 LOCK 位), 该位不能被修改。注: 任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。	0x0	RW
BKP	[13]	刹车输入极性 (Break polarity) 0: 刹车输入低电平有效; 1: 刹车输入高电平有效。 注: 一旦 LOCK 级别(TIM1_BDTR 寄存器中的 LOCK 位)设为'1', 则该位不能被修改。注: 任何对该位的写操作都需要一个 APB 时钟的延迟以后才能起作用。	0x0	RW
AOE	[14]	自动输出使能 (Automatic output enable) 0: MOE 只能被软件置'1'; 1: MOE 能被软件置'1'或在下一个更新事件被自动置'1'(如果刹车输入	0x0	RW

Bit Name	Position	Description	Default	Access
		无效。 注：一旦 LOCK 级别(TIM1_BDTR 寄存器中的 LOCK 位)设为'1', 则该位不能被修改。		
MOE	[15]	主输出使能 (Main output enable) 一旦刹车输入有效, 该位被硬件异步清'0'。根据 AOE 位的设置值, 该位可以由软件清'0'或被自动置 1。它仅对配置为输出的通道有效。 0: 禁止 OC 和 OCN 输出或强制为空闲状态; 1: 如果设置了相应的使能位(TIM1_CCER 寄存器的 CCxE、CCxNE 位), 则开启 OC 和 OCN 输出。 有关 OC/OCN 使能的细节, 参见 13.4.9 节, TIM1 和 TIM8 捕获/比较使能寄存器(TIM1_CCER)。	0x0	RW

Register Name: TIM1_DCR, Address: 0x40012C48, Default: 0x0

表143. DMA 控制寄存器

Bit Name	Position	Description	Default	Access
DBA	[0:4]	DMA 基地址 (DMA base address) 这些位定义了 DMA 在连续模式下的基地址(当对 TIM2_DMAR 寄存器进行读或写时), DBA 定义为从 TIM2_CR1 寄存器所在地址开始的偏移量: 00000: TIM2_CR1, 00001: TIM2_CR2, 00010: TIM2_SMCR,	0x0	RW
DBL	[8:12]	DMA 连续传送长度 (DMA burst length) 这些位定义了 DMA 在连续模式下的传送长度(当对 TIM2_DMAR 寄存器进行读或写时, 定时器则进行一次连续传送), 即: 定义传输的字节数目: 00000: 1 个字节 00001: 2 个字节 00010: 3 个字节	0x0	RW

Register Name: TIM1_DMAR, Address: 0x40012C4C, Default: 0x0

表144. 连续模式的 DMA 地址

Bit Name	Position	Description	Default	Access
DMAB	[0:15]	DMA 连续传送寄存器 (DMA register for burst accesses) 对 TIM2_DMAR 寄存器的读或写会导致对以下地址所在寄存器的存取操作: TIM2_CR1 地址 + DBA + DMA 索引, 其中: "TIM2_CR1 地址"是控制寄存器 1(TIM2_CR1)所在的地址; "DBA"是 TIM2_DCR 寄存器中定义的基地址; "DMA 索引"是由 DMA 自动控制的偏移量, 它取决于 TIM2_DCR 寄存器中定义的 DBL。	0x0	RW

17. 实时时钟 (RTC)

17.1. RTC 简介

实时时钟是一个独立的定时器。RTC 模块拥有一组连续计数的计数器, 在相应软件配置下, 可提供时钟日

历的功能。修改计数器的值可以重新设置系统当前的时间和日期。

RTC 模块和时钟配置系统 (RCC_BDCR 寄存器) 处于 AON 区域。

系统复位后, 对 AON 寄存器和 RTC 的访问被禁止, 这是为了防止对 AON 区域的意外写操作。执行以下操作将使能对 AON 寄存器和 RTC 的访问:

- 设置寄存器 RCC_APB1ENR 的 PWREN 和 BKPEN 位, 使能电源和 AON 接口时钟
- 设置寄存器 PWR_CR 的 DBP 位, 使能对 AON 寄存器和 RTC 的访问。

17.2. 主要特性

- 可编程的预分频系数: 分频系数最高为 220。
- 32 位的可编程计数器, 可用于较长时间段的测量。
- 2 个分离的时钟: 用于 APB1 接口的 PCLK1 和 RTC 时钟 (RTC 时钟的频率必须小于 PCLK1 时钟频率的四分之一以上)。
- 可以选择以下 2 种 RTC 的时钟源:
 - HSE 时钟除以 128;
 - LSI 振荡器时钟。
- 2 个独立的复位类型:
 - APB1 接口由系统复位;
 - RTC 核心 (预分频器、闹钟、计数器和分频器) 只能由 AON 域复位。
- 3 个专门的可屏蔽中断:
 - 闹钟中断, 用来产生一个软件可编程的闹钟中断。
 - 秒中断, 用来产生一个可编程的周期性中断信号 (最长可达 1 秒)。
 - 溢出中断, 指示内部可编程计数器溢出并回转为 0 的状态。

17.3. 功能描述

17.3.1. 概述

RTC 由两个主要部分组成 (参见下图)。第一部分 (APB1 接口) 用来和 APB1 总线相连。此单元还包含一组 16 位寄存器, 可通过 APB1 总线对其进行读写操作 (参见 16.4 节)。APB1 接口由 APB1 总线时钟驱动, 用来与 APB1 总线接口。

另一部分 (RTC 核心) 由一组可编程计数器组成, 分成两个主要模块。第一个模块是 RTC 的预分频模块, 它可编程产生最长为 1 秒的 RTC 时间基准 TR_CLK。RTC 的预分频模块包含了一个 20 位的可编程分频器 (RTC 预分频器)。如果在 RTC_CR 寄存器中设置了相应的允许位, 则在每个 TR_CLK 周期中 RTC 产生一个中断 (秒中断)。第二个模块是一个 32 位的可编程计数器, 可被初始化为当前的系统时间。系统时间按 TR_CLK 周期累加并与存储在 RTC_ALR 寄存器中的可编程时间相比较, 如果 RTC_CR 控制寄存器中设置了相应允许位, 比较匹配时将产生一个闹钟中断。

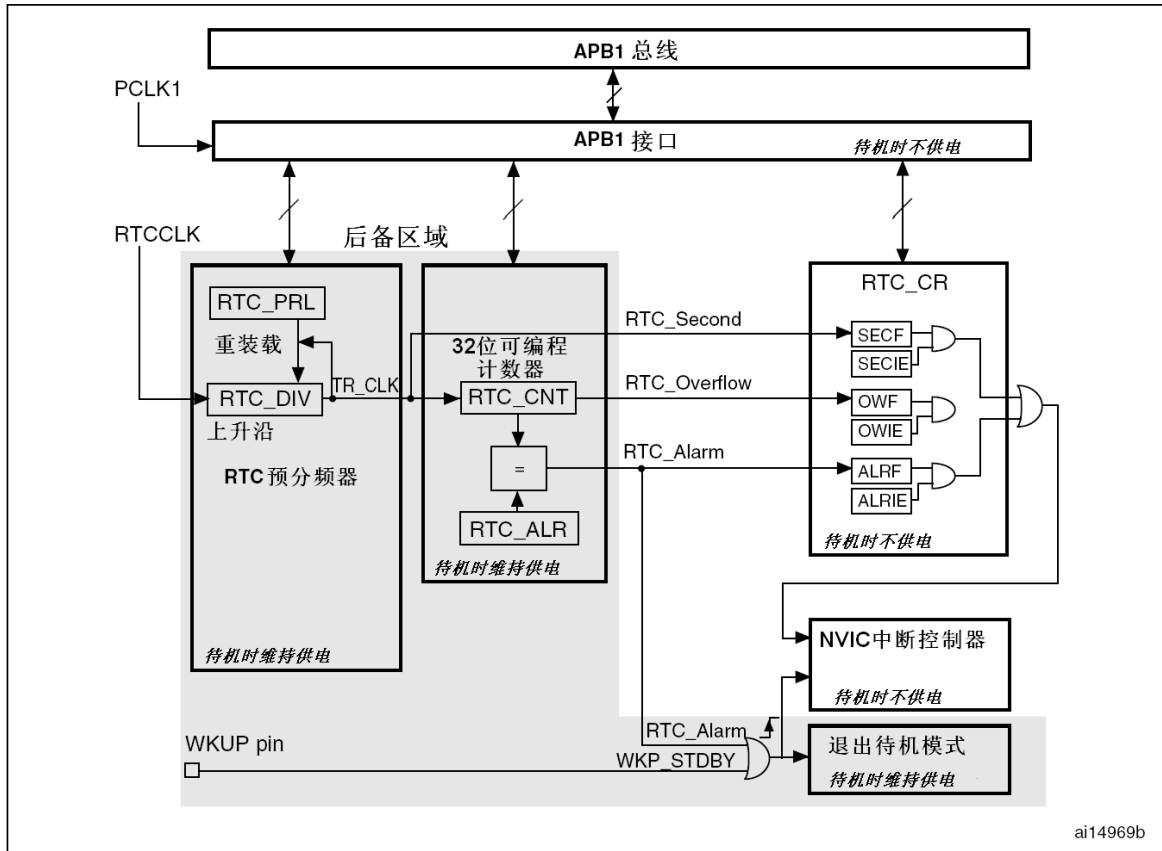


图107. 简化的 RTC 框图

17.3.2. 复位过程

除了 RTC_PRL、RTC_ALR、RTC_CNT 和 RTC_DIV 寄存器外，所有的系统寄存器都由系统复位。

RTC_PRL、RTC_ALR、RTC_CNT 和 RTC_DIV 寄存器仅能通过 AON 域复位信号复位或电源复位进行异步复位。

17.3.3. 读 RTC 寄存器

RTC 核完全独立于 RTC APB1 接口。

软件通过 APB1 接口访问 RTC 的预分频值、计数器值和闹钟值。但是，相关的可读寄存器只在与 RTC APB1 时钟进行重新同步的 RTC 时钟的上升沿被更新。RTC 标志也是如此的。这意味着，如果 APB1 接口曾经被关闭，而读操作又是在刚刚重新开启 APB1 之后，则在第一次的内部寄存器更新之前，从 APB1 上读出的 RTC 寄存器数值可能被破坏了（通常读到 0）。下述几种情况下能够发生这种情形：

- 发生系统复位或电源复位。
- 系统刚从待机模式唤醒。
- 系统刚从停机模式唤醒。

所有以上情况中，APB1 接口被禁止时（复位、无时钟或断电）RTC 核仍保持运行状态。因此，若在读取 RTC 寄存器时，RTC 的 APB1 接口曾经处于禁止状态，则软件首先必须等待 RTC_CR 寄存器中的 RSF 位（寄存器同步标志）被硬件置'1'。

注：RTC 的 APB1 接口不受 WFI 和 WFE 等低功耗模式的影响。

17.3.4. 配置 RTC 寄存器

必须设置 RTC_CRL 寄存器中的 CNF 位, 使 RTC 进入配置模式后, 才能写入 RTC_PRL、RTC_CNT、RTC_ALR 寄存器。

另外, 对 RTC 任何寄存器的写操作, 都必须在前一次写操作结束后进行。可以通过查询 RTC_CR 寄存器中的 RTOFF 状态位, 判断 RTC 寄存器是否处于更新中。仅当 RTOFF 状态位是'1' 时, 才可以写入 RTC 寄存器。

配置过程:

- 1) 查询 RTOFF 位, 直到 RTOFF 的值变为'1'
- 2) 置 CNF 值为 1, 进入配置模式
- 3) 对一个或多个 RTC 寄存器进行写操作
- 4) 清除 CNF 标志位, 退出配置模式
- 5) 查询 RTOFF, 直至 RTOFF 位变为'1'以确认写操作已经完成。

仅当 CNF 标志位被清除时, 写操作才能进行, 这个过程至少需要 3 个 RTCCLK 周期。

17.3.5. RTC 标志的设置

在每一个 RTC 核心的时钟周期中, 更改 RTC 计数器之前设置 RTC 秒标志 (SECF)。在计数器到达 0x0000 之前的最后一个 RTC 时钟周期中, 设置 RTC 溢出标志 (OWF)。

在计数器的值到达闹钟寄存器的值加 1 (RTC_ALR+1) 之前的 RTC 时钟周期中, 设置 RTC_Alarm 和 RTC 闹钟标志 (ALRF)。对 RTC 闹钟的写操作必须使用下述过程之一与 RTC 秒标志同步:

- 使用 RTC 闹钟中断, 并在中断处理程序中修改 RTC 闹钟和/或 RTC 计数器。
- 等待 RTC 控制寄存器中的 SECF 位被设置, 再更改 RTC 闹钟和/或 RTC 计数器。

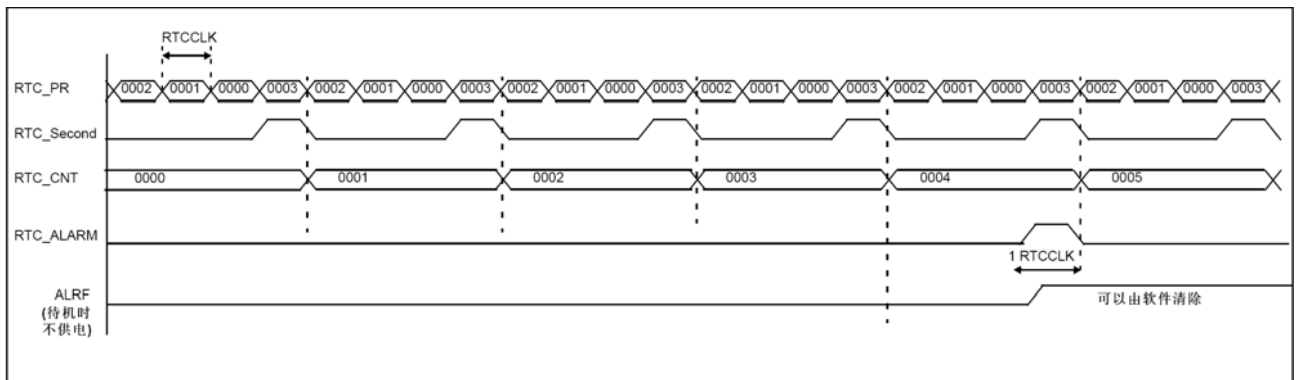


图108. RTC 秒和闹钟波形图示例, PR=0003, ALARM=00004

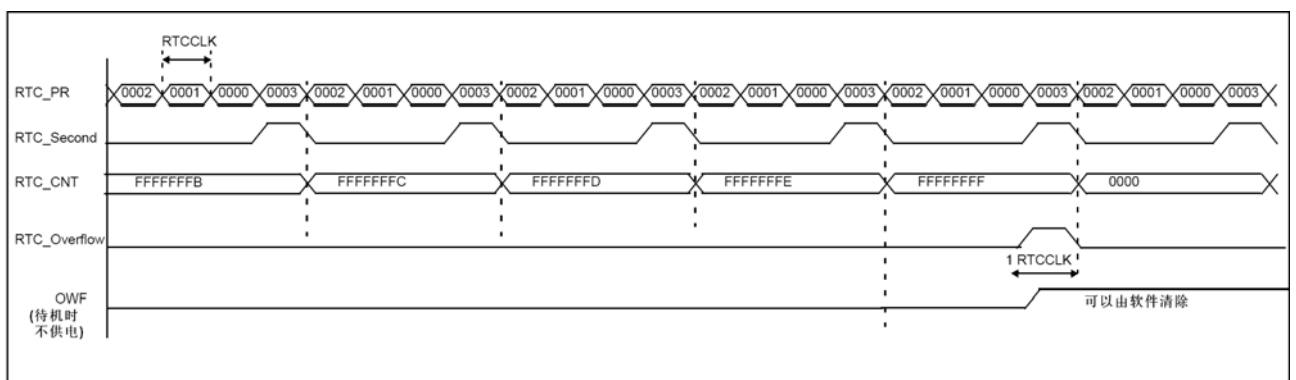


图109. RTC 溢出波形图示例, PR=0003

17.4. RTC 寄存器描述

Register Name: RTC_CRH, Address: 0x40002800, Default: 0x0

表145.

Bit Name	Position	Description	Default	Access
SECIE	[0]	允许秒中断 (Second interrupt enable) 0: 屏蔽(不允许)秒中断1: 允许秒中断	0x0	RW
ALRIE	[1]	允许闹钟中断 (Alarm interrupt enable) 0: 屏蔽(不允许)闹钟中断 1: 允许闹钟中断	0x0	RW
OWIE	[2]	允许溢出中断位 (Overflow interrupt enable) 0: 屏蔽(不允许)溢出中断1: 允许溢出中断	0x0	RW
	[3:15]	保留, 被硬件强制为0	0x0	

这些位用来屏蔽中断请求。注意: 系统复位后所有的中断被屏蔽, 因此可通过写 RTC 寄存器来确保在初始化后没有挂起的中断请求。当外设正在完成前一次写操作时(标志位 RTOFF=0), 不能对 RTC_CRH 寄存器进行写操作。

Register Name: RTC_CRL, Address: 0x40002804, Default: 0x0020

表146.

Bit Name	Position	Description	Default	Access
SECF	[0]	秒标志 (Second flag) 当32位可编程预分频器溢出时, 此位由硬件置'1'同时RTC计数器加1。因此, 此标志为分辨率可编程的RTC计数器提供一个周期性的信号(通常为1秒)。如果RTC_CRH寄存器中SECIE=1, 则产生中断。此位只能由软件清除。对此位写'1'是无效的。 0: 秒标志条件不成立; 1: 秒标志条件成立。	0x0	RW
ALRF	[1]	允许闹钟中断 (Alarm interrupt enable) 当 32 位可编程计数器达到 RTC_ALR 寄存器所设置的预定值, 此位由硬件置'1'。如果 RTC_CRH 寄存器中 ALRIE=1, 则产生中断。此位只能由软件清'0'。对此位写'1'是无效的。 0: 无闹钟; 1: 有闹钟。	0x0	RW
OWF	[2]	溢出标志 (Overflow flag) 当32位可编程计数器溢出时, 此位由硬件置'1'。如果RTC_CRH寄存器中OWIE=1, 则产生中断。此位只能由软件清'0'。对此位写'1'是无效的。 0: 无溢出; 1: 32位可编程计数器溢出。	0x0	RW
RSF	[3]	寄存器同步标志 (Registers synchronized flag) 每当 RTC_CNT 寄存器和 RTC_DIV 寄存器由软件更新或清'0'时, 此位由硬件置'1'。在 APB1 复位后, 或 APB1 时钟停止后, 此位必须由软件清'0'。要进行任何的读操作之前, 用户程序必须等待这位被硬件置'1', 以确保 RTC_CNT、RTC_ALR 或 RTC_PRL 已经被同步。 0: 寄存器尚未被同步;	0x0	RW

Bit Name	Position	Description	Default	Access
		1: 寄存器已经被同步。		
CNF	[4]	配置标志 (Configuration flag) 此位必须由软件置'1'以进入配置模式, 从而允许向 RTC_CNT、RTC_ALR 或 RTC_PRL 寄存器写入数据。只有当此位在被置'1'并重新由软件清'0'后, 才会执行写操作。 0: 退出配置模式(开始更新 RTC 寄存器); 1: 进入配置模式。	0x0	RW
RTOFF	[5]	RTC 操作关闭 (RTC operation OFF) RTC 模块利用这位来指示对其寄存器进行的最后一次操作的状态, 指示操作是否完成。若此位为'0', 则表示无法对任何的 RTC 寄存器进行写操作。此位为只读位。 0: 上一次对 RTC 寄存器的写操作仍在进行; 1: 上一次对 RTC 寄存器的写操作已经完成。	0x1	R
	[6:15]	保留, 被硬件强制为0	0x0	

当前一个写操作还未完成时(RTOFF=0时), 不能写RTC_CR寄存器。

注:

1. 任何标志位都将保持挂起状态, 直到适当的 RTC_CR 请求位被软件复位, 表示所请求的中断已经被接受。
2. 在复位时禁止所有中断, 无挂起的中断请求, 可以对 RTC 寄存器进行写操作。
3. 当 APB1 时钟不运行时, OWF、ALRF、SECF 和 RSF 位不被更新。
4. OWF、ALRF、SECF 和 RSF 位只能由硬件置位, 由软件来清零。
5. 若 ALRF=1 且 ALRIE=1, 则允许产生 RTC 全局中断。如果在 EXTI 控制器中允许产生 EXTI 线 17
6. 中断, 则允许产生 RTC 全局中断和 RTC 闹钟中断。

7. 若 ALRF=1, 如果在 EXTI 控制器中设置了 EXTI 线 17 的中断模式, 则允许产生 RTC 闹钟中断; 如果在 EXTI 控制器中设置了 EXTI 线 17 的事件模式, 则这条线上会产生一个脉冲(不会产生 RTC 闹钟中断)。

Register Name: RTC_PRLH, Address: 0x40002808, Default: 0x0

预分频装载寄存器用来保存RTC预分频器的周期计数值。它们受RTC_CR寄存器的RTOFF位保护, 仅当RTOFF值为'1'时允许进行写操作。

表147.

Bit Name	Position	Description	Default	Access
PRLH	[0:3]	RTC 预分频装载值高位 (RTC prescaler reload value high) 根据以下公式, 这些位用来定义计数器的时钟频率: $f_{TR_CLK} = f_{RTCCLK} / (PRL[19:0] + 1)$ 注: 不推荐使用0值, 否则无法正确的产生RTC中断和标志位。	0x0	W
	[4:15]	保留, 被硬件强制为0	0x0	

Register Name: RTC_PRL, Address: 0x4000280C, Default: 0x8000

表148.

Bit Name	Position	Description	Default	Access
PRL	[0:15]	RTC 预分频装载值低位 根据以下公式, 这些位用来定义计数器的时钟频率: $f_{TR_CLK} = f_{RTCCLK} / (PRL[19:0] + 1)$	0x8000	W

注: 如果输入时钟频率是 32.768kHz(f_{RTCCLK}), 这个寄存器中写入 7FFFh 可获得周期为 1 秒钟的信号。

Register Name: RTC_DIVH, Address: 0x40002810, Default: 0x0

在TR_CLK的每个周期里, RTC预分频器中计数器的值都会被重新设置为RTC_PRL寄存器的值。用户可通过读取RTC_DIV寄存器, 以获得预分频计数器的当前值, 而不停止分频计数器的工作, 从而获得精确的时间测量。此寄存器是只读寄存器, 其值在RTC_PRL或RTC_CNT寄存器中的值发生改变后, 由硬件重新装载。

表149.

Bit Name	Position	Description	Default	Access
RTC_DIVH	[0:3]	RTC时钟分频器余数高位 (RTC clock divider high)	0x0	R
	[4:15]	保留	0x0	

Register Name: RTC_DIVL, Address: 0x40002814, Default: 0x0

表150.

Bit Name	Position	Description	Default	Access
RTC_DIVL	[0:15]	RTC_DIV[15:0] : RTC时钟分频器余数低位 (RTC clock divider low)	0x0	R

Register Name: RTC_CNTH, Address: 0x40002818, Default: 0x0

RTC核有一个32位可编程的计数器, 可通过两个16位的寄存器访问。计数器以预分频器产生的TR_CLK 时间基准为参考进行计数。RTC_CNT 寄存器用来存放计数器的计数值。他们受RTC_CR 的位 RTOFF 写保护, 仅当RTOFF 值为 '1' 时, 允许写操作。在高或低寄存器(RTC_CNTH或RTC_CNTL)上的写操作, 能够直接装载到相应的可编程计数器, 并且重新装载RTC预分频器。当进行读操作时, 直接返回计数器内的计数值(系统时间)。

表151.

Bit Name	Position	Description	Default	Access
RTC_CNTH	[0:15]	RTC 计数器高位 (RTC counter high) 可通过读RTC_CNTH寄存器来获得RTC计数器当前值的高位部分。要对此寄存器进行写操作前, 必须先进入配置模式	0x0	RW

Register Name: RTC_CNTL, Address: 0x4000281C, Default: 0x0

表152.

Bit Name	Position	Description	Default	Access
RTC_CNTL	[0:15]	RTC 计数器低位。 可通过读RTC_CNTL寄存器来获得RTC计数器当前值的低位部分。要对此寄存器进行写操作, 必须先进入配置模式	0x0	RW

Register Name: RTC_ALRH, Address: 0x40002820, Default: 0xFFFF

当可编程计数器的值与RTC_ALR中的32位值相等时, 即触发一个闹钟事件, 并且产生RTC闹钟中断。此寄存器受RTC_CR寄存器里的RTOFF位写保护, 仅当RTOFF值为'1'时, 允许写操作。

表153.

Bit Name	Position	Description	Default	Access
RTC_ALRH	[0:15]	RTC 闹钟值高位 (RTC alarm high) 此寄存器用来保存由软件写入的闹钟时间的高位部分。要对此寄存器进行写操作, 必须先进入配置模式。	0xFFFF	W

Register Name: RTC_ALRL, Address: 0x40002824, Default: 0xFFFF

表154.

Bit Name	Position	Description	Default	Access
RTC_ALRL	[0:15]	RTC 闹钟值低位 (RTC alarm low) 此寄存器用来保存由软件写入的闹钟时间的低位部分。要对此	0xFFFF	W

寄存器进行写操作，必须先进入配置模式。

18. 独立看门狗 (IWDG)

18.1. 简介

独立看门狗 (IWDG) 由专用的低速时钟 (LSI) 驱动，即使主时钟发生故障它也仍然有效。窗口看门狗由从 APB1 时钟分频后得到的时钟驱动，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。

IWDG 最适合应用于那些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场景。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。

18.2. IWDG 主要性能

- 自由运行的递减计数器
- 时钟由独立的 RC 振荡器提供 (可在停止和待机模式下工作)
- 看门狗被激活后，则在计数器计数至 0x000 时产生复位

18.3. IWDG 功能描述

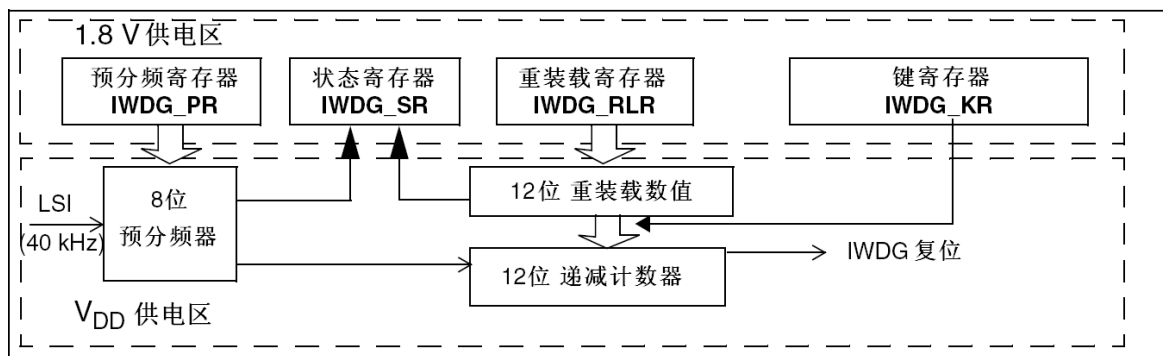


图110. 独立看门狗框图

注：看门狗功能处于 VDD 供电区，即在停机和待机模式时仍能正常工作。

IWDG_EN 寄存器为 1 开始启用独立看门狗；此时计数器开始从其复位值 0xFFFF 递减计数。当计数器计数到末尾 0x000 时，会产生一个复位信号 (IWDG_RESET)。

无论何时，只要在键寄存器 IWDG_KR 中写入 0xAAAA，IWDG_RLR 中的值就会被重新加载到计数器，从而避免产生看门狗复位。

18.3.1. 上电默认使能看门狗

在系统上电复位后，IWDG_EN 寄存器默认为 1，看门狗会自动开始运行；如果在计数器计数结束前，若软件没有向键寄存器写入相应的值，则系统会产生复位。在 IWDG_LOCK 寄存器为 0 的前提下，可以将 IWDG_EN 寄存器改写为 0，这样停止独立看门狗的运行。

18.3.2. 寄存器访问保护

IWDG_PR 和 IWDG_RLR 寄存器具有写保护功能。要修改这两个寄存器的值，在 IWDG_LOCK 寄存器为 0 的前提下，必须先向 IWDG_KR 寄存器中写入 0x5555。以不同的值写入这个寄存器将会打乱操作顺序，寄存器将

重新被保护。重装载操作（即写入 0xAAAA）也会启动写保护功能。如果 IWDG_LOCK 寄存器为 1，WDG_PR、IWDG_RLR、IWDG_EN 三个寄存器无论如何也不会被改变，IWDG_LOCK 寄存器一旦被写为 1，只有任意复位源才能将其复位为 0。

状态寄存器指示预分频值和递减计数器是否正在被更新。

18.3.3. 调试模式

当微控制器进入调试模式时 (Cortex-M3 核心停止)，根据调试模块中的 DBG_IWDG_STOP 配置位的状态，IWDG 的计数器能够继续工作或停止。详见有关调试模块的章节。

表155. 看门狗超时时间（40kHz 的输入时钟（LSI））

预分频系数	PR[2:0]位	最短时间 (ms) RL[11:0] = 0x000	最长时间 (ms) RL[11:0] = 0xFF
/4	0	0.1	409.6
/8	1	0.2	819.2
/16	2	0.4	1638.4
/32	3	0.8	3276.8
/64	4	1.6	6553.6
/128	5	3.2	13107.2
/256	(6或7)	6.4	26214.4

注：这些时间是按照 40kHz 时钟给出。实际上，MCU 内部的 RC 频率会在 30kHz 到 60kHz 之间变化。此外，即使 RC 振荡器的频率是精确的，确切的时序仍然依赖于 APB 接口时钟与 RC 振荡器时钟之间的相位差，因此总会有一个完整的 RC 周期是不确定的。

18.4. IWDG 寄存器描述

Register Name: IWDG_KR, Address: 0x40003000, Default: 0x0

表156. 键寄存器

Bit Name	Position	Description	Default	Access
KEY	[0:15]	键值(只写寄存器，读出值为 0x0000) (Key value) 软件必须以一定的间隔写入 0xAAAA，否则，当计数器为 0 时，看门狗会产生复位。写入 0x5555 表示允许访问 IWDG_PR 和 IWDG_RLR 寄存器。	0x0	W(WriteOnly)

Register Name: IWDG_PR, Address: 0x40003004, Default: 0x7

表157. 预分频寄存器

Bit Name	Position	Description	Default	Access
PR	[0:2]	预分频因子 (Prescaler divider) 这些位具有写保护设置，参见 17.3.2 节。通过设置这些位来选择计数器时钟的预分频因子。要改变预分频因子，IWDG_SR 寄存器的 PVU 位必须为 0。 000: 预分频因子=4 100: 预分频因子=64 001: 预分频因子=8 101: 预分频因子=128 010: 预分频因子=16 110: 预分频因子=256 011: 预分频因子=32 111: 预分频因子=256 注意：对此寄存器进行读操作，将从 VDD 电压域返回预分频值。如果写操作正在进行，则读回的值可能是无效的。因此，只有当 IWDG_SR 寄存器的 PVU 位为 0 时，读出的值才有效。	0x7	RW

Register Name: IWDG_RLR, Address: 0x40003008, Default: 0x9c0

表158. 重装载寄存器

Bit Name	Position	Description	Default	Access
RL	[0:11]	看门狗计数器重装载值 (Watchdog counter reload value) 这些位具有写保护功能。用于定义看门狗计数器的重装载值，每当向 IWDG_KR 寄存器写入 0xAAAA 时，重装载值会被传送到计数器中。随后计数器从这个值开始递减计数。看门狗超时周期可通过此重装载值和时钟预分频值来计算，参照表 83。 只有当 IWDG_SR 寄存器中的 RVU 位为 0 时，才能对此寄存器进行修改。 注：对此寄存器进行读操作，将从 VDD 电压域返回预分频值。如果写操作正在进行，则读回的值可能是无效的。因此，只有当 IWDG_SR 寄存器的 RVU 位为 0 时，读出的值才有效。	0x9c0	RW

Register Name: IWDG_SR, Address: 0x4000300c, Default: 0x0

表159. 状态寄存器

Bit Name	Position	Description	Default	Access
PVU	[0]	看门狗预分频值更新 (Watchdog prescaler value update) 此位由硬件置'1'用来指示预分频值的更新正在进行中。当在 VDD 域中的预分频值更新结束后，此位由硬件清'0'(最多需 5 个 40kHz 的 RC 周期)。预分频值只有在 PVU 位被清'0'后才可更新。 如果在应用程序中使用了多个重装载值或预分频值，则必须在 RVU 位被清除后才能重新改变预装载值，在 PVU 位被清除后才能重新改变预分频值。然而，在预分频和/或重装载值更新后，不必等待 RVU 或 PVU 复位，可继续执行下面的代码。	0x0	R(ReadOnly)
RVU	[1]	看门狗计数器重装载值更新 (Watchdog counter reload value update) 此位由硬件置'1'用来指示重装载值的更新正在进行中。当在 VDD 域中的重装载更新结束后，此位由硬件清'0'(最多需 5 个 40kHz 的 RC 周期)。重装载值只有在 RVU 位被清'0'后才可更新。 如果在应用程序中使用了多个重装载值或预分频值，则必须在 RVU 位被清除后才能重新改变预装载值，在 PVU 位被清除后才能重新改变预分频值。然而，在预分频和/或重装载值更新后，不必等待 RVU 或 PVU 复位，可继续执行下面的代码。	0x0	R(ReadOnly)

Register Name: IWDG_EN, Address: 0x40003010, Default: 0x1

表160. 看门狗使能寄存器

Bit Name	Position	Description	Default	Access
IWDG_AUTO_EN	[0]	上电默认启动看门狗，可在 IWDG_LOCK 为 0 的前提下改写该位为 0 关闭看门狗的运行	0x1	RW

Register Name: IWDG_LOCK, Address: 0x40003014, Default: 0x0

表161. 看门狗配置死锁寄存器

Bit Name	Position	Description	Default	Access
IWDG_LOCK	[0]	当写该位为 1 时，WDG_PR、IWDG_RLR、IWDG_EN 三个寄存器不可被改写。只有任意复位源才能将该位复位到 0。	0x0	RW

19. 通用串行异步通信接口 (USART)

19.1. USART 介绍

通用同步异步收发器 (USART) 提供了一种灵活的方法与使用工业标准 NRZ 异步串行数据格式的外部设备之间进行全双工数据交换。USART 利用分数波特率发生器提供宽范围的波特率选择。

它支持同步单向通信和半双工单线通信, 也支持 LIN (局部互连网), 智能卡协议和 IrDA (红外数据组织) SIR ENDEC 规范, 以及调制解调器 (CTS/RTS) 操作。它还允许多处理器通信。

使用多缓冲器配置的 DMA 方式, 可以实现高速数据通信。

19.2. USART 主要特性

- 全双工的, 异步通信
- NRZ 标准格式
- 分数波特率发生器系统
 - 发送和接收共用的可编程波特率, 最高达 2Mbits/s
- 可编程数据字长度 (8 位或 9 位)
- 可配置的停止位-支持 1 或 2 个停止位
- LIN 主发送同步断开符的能力以及 LIN 从检测断开符的能力
 - 当 USART 硬件配置成 LIN 时, 生成 13 位断开符; 检测 10/11 位断开符
- 发送方为同步传输提供时钟
- IRDA SIR 编码器解码器
 - 在正常模式下支持 3/16 位的持续时间
- 智能卡模拟功能
 - 智能卡接口支持 ISO7816-3 标准里定义的异步智能卡协议
 - 智能卡用到的 0.5 和 1.5 个停止位
- 单线半双工通信
- 可配置的使用 DMA 的多缓冲器通信
 - 在 SRAM 里利用集中式 DMA 缓冲接收/发送字节
- 单独的发送器和接收器使能位
- 检测标志
 - 接收缓冲器满
 - 发送缓冲器空
 - 传输结束标志
- 校验控制
 - 发送校验位
 - 对接收数据进行校验
- 四个错误检测标志
 - 溢出错误
 - 噪音错误
 - 帧错误
 - 校验错误
- 10 个带标志的中断源
 - CTS 改变
 - LIN 断开符检测
 - 发送数据寄存器空

- 发送完成
- 接收数据寄存器满
- 检测到总线为空闲
- 溢出错误
- 帧错误
- 噪音错误
- 校验错误
- 多处理器通信 -- 如果地址不匹配, 则进入静默模式
- 从静默模式中唤醒 (通过空闲总线检测或地址标志检测)
- 两种唤醒接收器的方式: 地址位 (MSB, 第 9 位), 总线空闲

19.3. USART 功能概述

接口通过三个引脚与其他设备连接在一起。任何 USART 双向通信至少需要两个脚: 接收数据输入 (RX) 和发送数据输出 (TX)。

RX: 接收数据串行输入。通过过采样技术来区别数据和噪音, 从而恢复数据。

TX: 发送数据输出。当发送器被禁止时, 输出引脚恢复到它的 I/O 端口配置。当发送器被激活, 并且不发送数据时, TX 引脚处于高电平。在单线和智能卡模式里, 此 I/O 口被同时用于数据的发送和接收。

- 总线在发送或接收前应处于空闲状态
- 一个起始位
- 一个数据字 (8 或 9 位), 最低有效位在前
- 0.5, 1.5, 2 个的停止位, 由此表明数据帧的结束
- 使用分数波特率发生器 —— 12 位整数和 4 位小数的表示方法。
- 一个状态寄存器 (USART_SR)
- 数据寄存器 (USART_DR)
- 一个波特率寄存器 (USART_BRR), 12 位的整数和 4 位小数
- 一个智能卡模式下的保护时间寄存器 (USART_GTPR)

关于以上寄存器中每个位的具体定义, 请参考寄存器描述: USART 寄存器描述。在同步模式中需要下列引脚:

- CK: 发送器时钟输出。此引脚输出用于同步传输的时钟, (在 Start 位和 Stop 位上没有时钟脉冲, 软件可选地, 可以在最后一个数据位送出一个时钟脉冲)。数据可以在 RX 上同步被接收。这可以用来控制带有移位寄存器的外部设备 (例如 LCD 驱动器)。时钟相位和极性都是软件可编程的。在智能卡模式里, CK 可以为智能卡提供时钟。

在 IrDA 模式里需要下列引脚:

- IrDA_RDI: IrDA 模式下的数据输入。
- IrDA_TDO: IrDA 模式下的数据输出。

下列引脚在硬件流控模式中需要:

- nCTS: 清除发送, 若是高电平, 在当前数据传输结束时阻断下一次的数据发送。
- nRTS: 发送请求, 若是低电平, 表明 USART 准备好接收数据

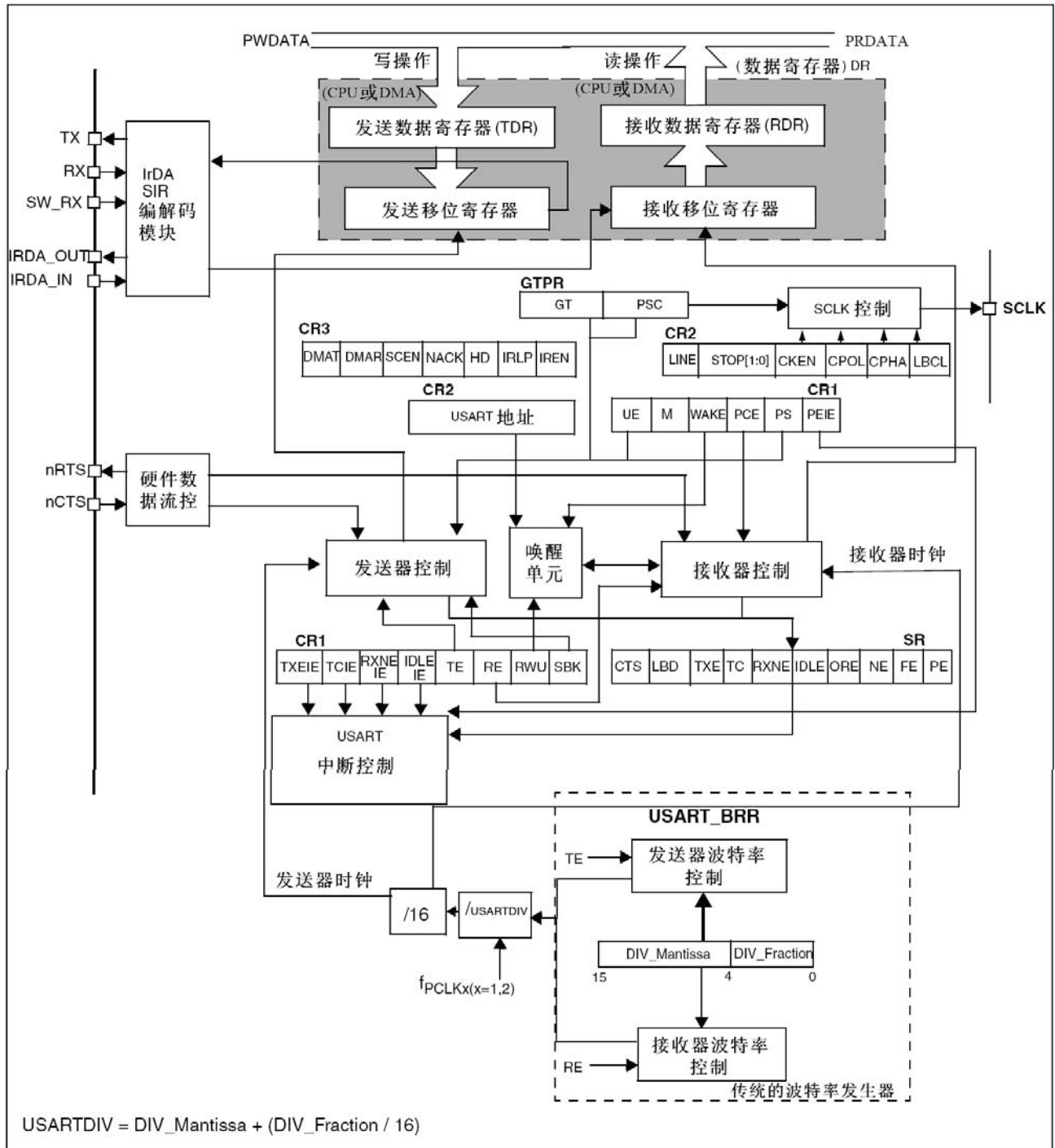


图111. USART 框图

19.3.1. USART 特性描述

字长可以通过编程 USART_CR1 寄存器中的 M 位，选择成 8 或 9 位。在起始位期间，TX 脚处于低电平，在停止位期间处于高电平。

空闲符号被视为完全由'1'组成的一个完整的数据帧，后面跟着包含了数据的下一帧的开始位（'1' 的位数也包括了停止位的位数）。

断开符号 被视为在一个帧周期内全部收到'0'（包括停止位期间，也是'0'）。在断开帧结束时，发送器再插入 1 或 2 个停止位（'1'）来应答起始位。

发送和接收由一共用的波特率发生器驱动，当发送器和接收器的使能位分别置位时，分别为其产生时钟。

随后将有每个功能块的详细说明。

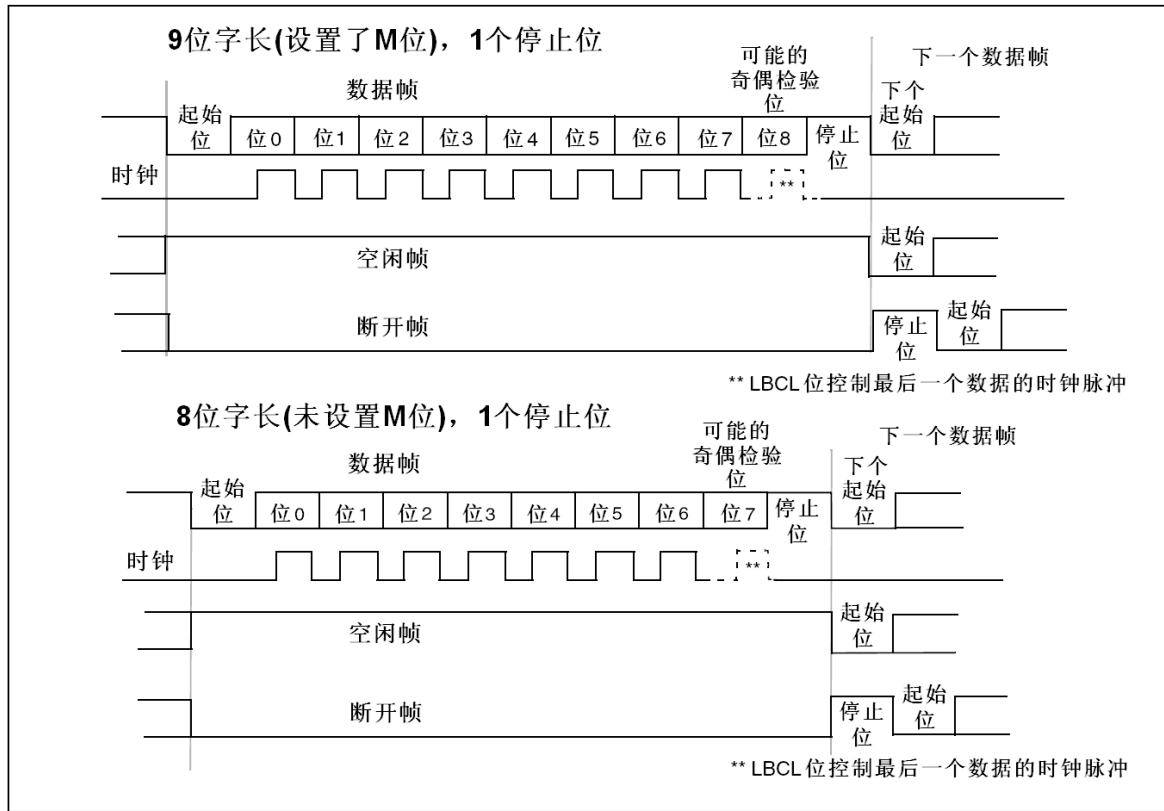


图112. 字长设置

19.3.2. 发送器

发送器根据 M 位的状态发送 8 位或 9 位的数据字。当发送使能位 (TE) 被设置时, 发送移位寄存器中的数据在 TX 脚上输出, 相应的时钟脉冲在 CK 脚上输出。

19.3.2.1. 字符发送

在 USART 发送期间, 在 TX 引脚上首先移出数据的最低有效位。在此模式里, USART_DR 寄存器包含了一个内部总线和发送移位寄存器之间的缓冲器。

每个字符之前都有一个低电平的起始位; 之后跟着的停止位, 其数目可配置。

USART 支持多种停止位的配置: 0.5、1、1.5 和 2 个停止位。

注:

1. 在数据传输期间不能复位 TE 位, 否则将破坏 TX 脚上的数据, 因为波特率计数器停止计数。正在传输的当前数据将丢失。
2. TE 位被激活后将发送一个空闲帧。

19.3.2.2. 可配置的停止位

随每个字符发送的停止位的位数可以通过控制寄存器 2 的位 13、12 进行编程。

1 个停止位: 停止位位数的默认值。

2 个停止位: 可用于常规 USART 模式、单线模式以及调制解调器模式。

0.5 个停止位: 在智能卡模式下接收数据时使用。

1.5 个停止位: 在智能卡模式下发送和接收数据时使用。

空闲帧包括了停止位。

断开帧是 10 位低电平，后跟停止位（当 m=0 时）；或者 11 位低电平，后跟停止位（m=1 时）。不可能传输更长的断开帧（长度大于 10 或者 11 位）。

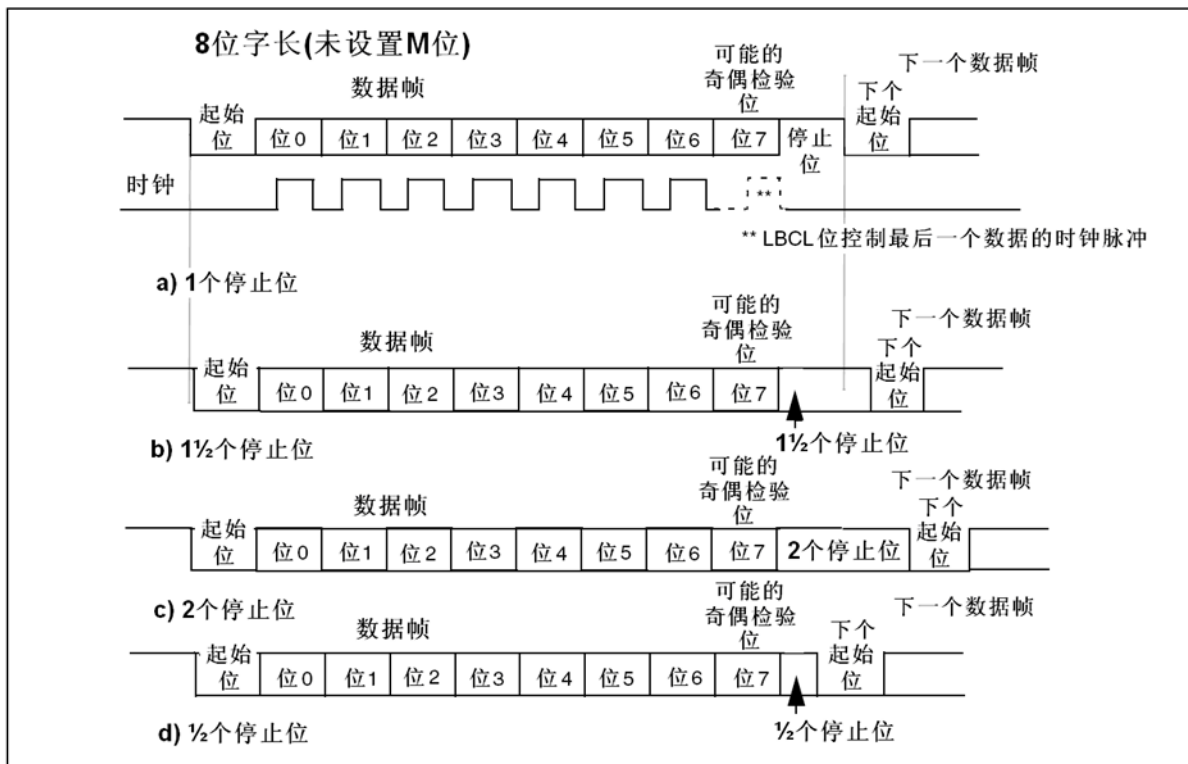


图113. 配置停止位

配置步骤：

- 1) 通过在 USART_CR1 寄存器上置位 UE 位来激活 USART
- 2) 编程 USART_CR1 的 M 位来定义字长。
- 3) 在 USART_CR2 中编程停止位的位数。
- 4) 如果采用多缓冲器通信，配置 USART_CR3 中的 DMA 使能位（DMAT）。按多缓冲器通信中的描述配置 DMA 寄存器。
- 5) 利用 USART_BRR 寄存器选择要求的波特率。
- 6) 设置 USART_CR1 中的 TE 位，发送一个空闲帧作为第一次数据发送。
- 7) 把要发送的数据写进 USART_DR 寄存器（此动作清除 TXE 位）。在只有一个缓冲器的情况下，对每个待发送的数据重复步骤 7。
- 8) 在 USART_DR 寄存器中写入最后一个数据字后，要等待 TC=1，它表示最后一个数据帧的传输结束。当需要关闭 USART 或需要进入停机模式之前，需要确认传输结束，避免破坏最后一次传输。

19.3.2.3. 单字节通信

清零 TXE 位总是通过对数据寄存器的写操作来完成的。TXE 位由硬件来设置，它表明：

- 数据已经从 TDR 移送到移位寄存器，数据发送已经开始
- TDR 寄存器被清空
- 下一个数据可以被写进 USART_DR 寄存器而不会覆盖先前的数据

如果 TXEIE 位被设置，此标志将产生一个中断。

如果此时 USART 正在发送数据，对 USART_DR 寄存器的写操作把数据存进 TDR 寄存器，并在当前传输结束时把该数据复制进移位寄存器。

如果此时 USART 没有在发送数据，处于空闲状态，对 USART_DR 寄存器的写操作直接把数据放进移位寄存器，数据传输开始，TXE 位立即被置起。

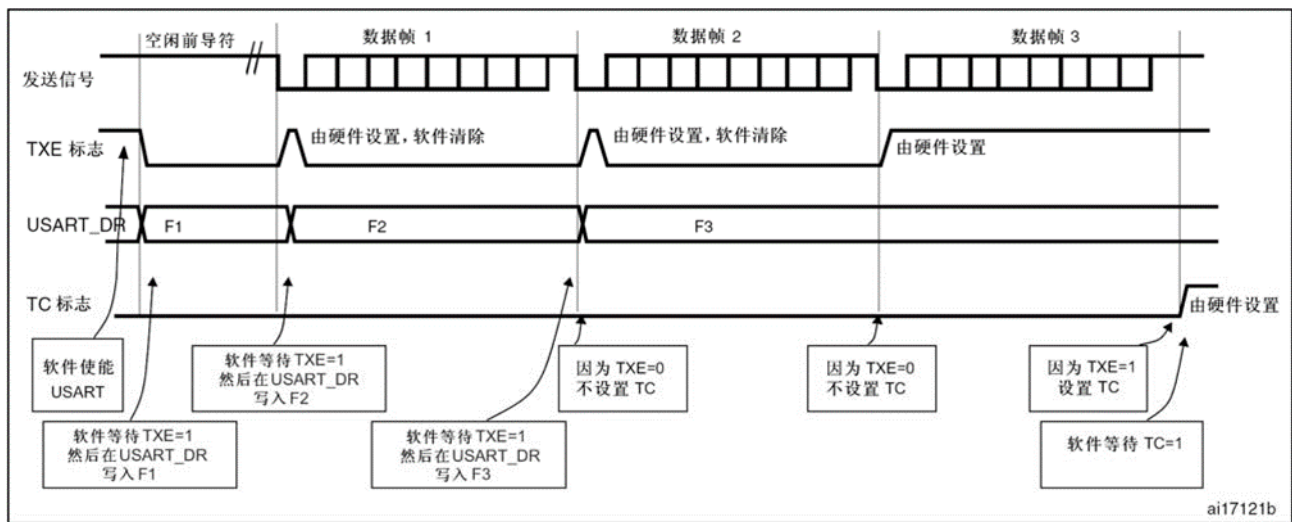
当一帧发送完成时（停止位发送后）并且设置了 TXE 位，TC 位被置起，如果 USART_CR1 寄存器中的 TCIE 位被置起时，则会产生中断。

在 USART_DR 寄存器中写入了最后一个数据字后，在关闭 USART 模块之前或设置微控制器进入低功耗模式（详见下图）之前，必须先等待 TC=1。

使用下列软件过程清除 TC 位：

- 1) 读一次 USART_SR 寄存器；
- 2) 写一次 USART_DR 寄存器。

注：TC 位也可以通过软件对它写'0'来清除。此清零方式只推荐在多缓冲器通信模式下使用。



19.3.2.4. 断开符号

设置 SBK 可发送一个断开符号。断开帧长度取决 M 位。如果设置 SBK=1，在完成当前数据发送后，将在 TX 线上发送一个断开符号。断开字符发送完成时（在断开符号的停止位时）SBK 被硬件复位。USART 在最后一个断开帧的结束处插入一逻辑'1'，以保证能识别下一帧的起始位。

注：如果在开始发送断开帧之前，软件又复位了 SBK 位，断开符号将不被发送。如果要发送两个连续的断开帧，SBK 位应该在前一个断开符号的停止位之后置起。

19.3.2.5. 空闲符号

置位 TE 将使得 USART 在第一个数据帧前发送一空闲帧。

19.3.3. 接收器

USART 可以根据 USART_CR1 的 M 位接收 8 位或 9 位的数据字。

19.3.3.1. 起始位侦测

在 USART 中，如果辨认出一个特殊的采样序列，那么就认为侦测到一个起始位。该序列为：1110X0X0X0000

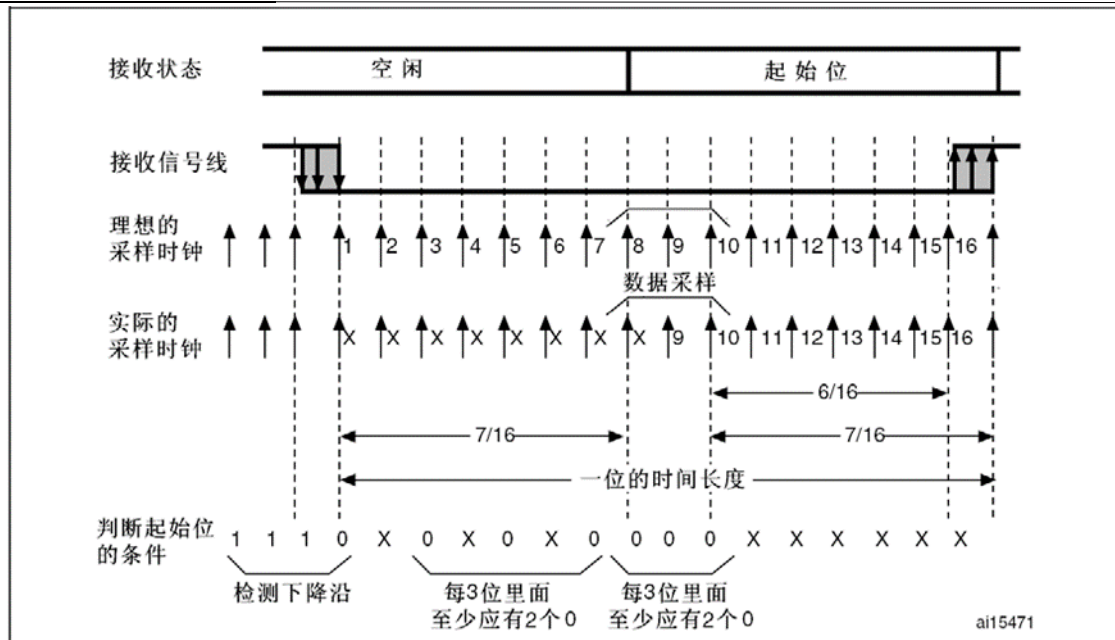


图115. 起始位检测

注：如果该序列不完整，那么接收端将退出起始位检测并回到空闲状态（不设置标志位）等待下降沿。

如果 3 个采样点都为 '0'（在第 3、5、7 位的第一次采样，和在第 8、9、10 的第二次采样都为 '0'），则确认收到起始位，这时设置 RXNE 标志位，如果 RXNEIE=1，则产生中断。

如果两次 3 个采样点上仅有 2 个是 '0'（第 3、5、7 位的采样点和第 8、9、10 位的采样点），那么起始位仍然是有效的，但是会设置 NE 噪声标志位。如果不能满足这个条件，则中止起始位的检测过程，接收器会回到空闲状态（不设置标志位）。

如果有一次 3 个采样点上仅有 2 个是 '0'（第 3、5、7 位的采样点或第 8、9、10 位的采样点），那么起始位仍然是有效的，但是会设置 NE 噪声标志位。

19.3.3.2. 字符接收

在 USART 接收期间，数据的最低有效位首先从 RX 脚移进。在此模式里，USART_DR 寄存器包含的缓冲器位于内部总线和接收移位寄存器之间。

配置步骤：

- 1) 将 USART_CR1 寄存器的 UE 置 1 来激活 USART。
- 2) 编程 USART_CR1 的 M 位定义字长。
- 3) 在 USART_CR2 中编写停止位的个数。

如果需多缓冲器通信，选择 USART_CR3 中的 DMA 使能位 (DMAR)。按多缓冲器通信所要求的配置 DMA 寄存器。

利用波特率寄存器 USART_BRR 选择希望的波特率。

设置 USART_CR1 的 RE 位。激活接收器，使它开始寻找起始位。

当一个字符被接收到时，

• RXNE 位被置位。它表明移位寄存器的内容被转移到 RDR。换句话说，数据已经被接收并且可以被读出（包括与之有关的错误标志）。

- 如果 RXNEIE 位被设置，产生中断。
- 在接收期间如果检测到帧错误，噪音或溢出错误，错误标志将被置起，
- 在多缓冲器通信时，RXNE 在每个字节接收后被置起，并由 DMA 对数据寄存器的读操作而清零。
- 在单缓冲器模式里，由软件读 USART_DR 寄存器完成对 RXNE 位清除。RXNE 标志也可以通过对它写 0 来清除。RXNE 位必须在下一字符接收结束前被清零，以避免溢出错误。

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

19.3.3.3. 断开符号

当接收到一个断开帧时，USART 像处理帧错误一样处理它。

19.3.3.4. 空闲符号

当一空闲帧被检测到时，其处理步骤和接收到普通数据帧一样，但如果 IDLEIE 位被设置将产生一个中断。

19.3.3.5. 溢出错误

如果 RXNE 还没有被复位，又接收到一个字符，则发生溢出错误。数据只有当 RXNE 位被清零后才能从移位寄存器转移到 RDR 寄存器。RXNE 标记是接收到每个字节后被置位的。如果下一个数据已被收到或先前 DMA 请求还没被服务时，RXNE 标志仍是置起的，溢出错误产生。

当溢出错误产生时：

- ORE 位被置位。
- RDR 内容将不会丢失。读 USART_DR 寄存器仍能得到先前的数据。
- 移位寄存器中以前的内容将被覆盖。随后接收到的数据都将丢失。
- 如果 RXNEIE 位被设置或 EIE 和 DMAR 位都被设置，中断产生。
- 顺序执行对 USART_SR 和 USART_DR 寄存器的读操作，可复位 ORE 位

注：当 ORE 位置位时，表明至少有 1 个数据已经丢失。有两种可能性：

1) 如果 RXNE=1，上一个有效数据还在接收寄存器 RDR 上，可以被读出。

2) 如果 RXNE=0，这意味着上一个有效数据已经被读走，RDR 已经没有东西可读。当上一个有效数据在 RDR 中被读取的同时又接收到新的（也就是丢失的）数据时，此种情况可能发生。在读序列期间（在 USART_SR 寄存器读访问和 USART_DR 读访问之间）接收到新的数据，此种情况也可能发生。

19.3.3.6. 噪音错误

使用过采样技术（同步模式除外），通过区别有效输入数据和噪音来进行数据恢复。

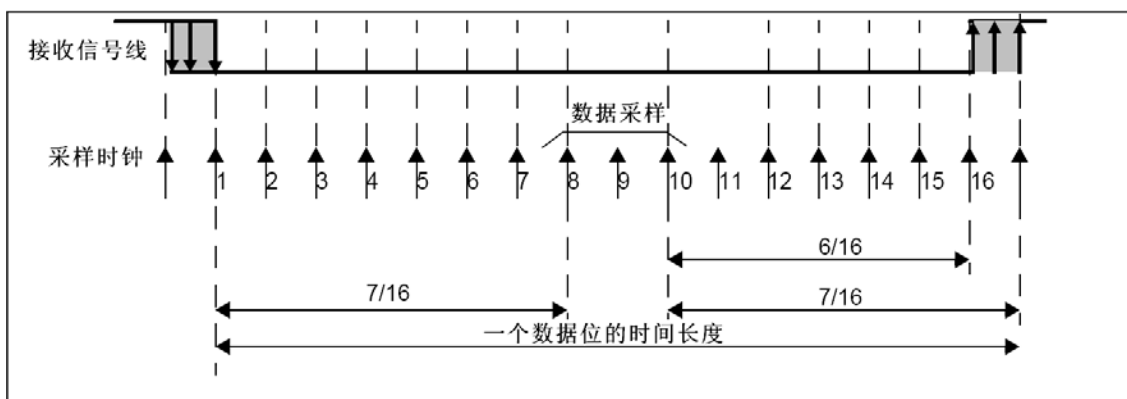


图116. 检测噪声的数据采样

表162. 检测噪声的数据采样

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

当在接收帧中检测到噪声时:

- 在 RXNE 位的上升沿设置 NE 标志。
- 无效数据从移位寄存器传送到 USART_DR 寄存器。

在单个字节通信情况下, 没有中断产生。然而, 因为 NE 标志位和 RXNE 标志位是同时被设置, RXNE 将产生中断。在多缓冲器通信情况下, 如果已经设置了 USART_CR3 寄存器中 EIE 位, 将产生一个中断。

先读出 USART_SR, 再读出 USART_DR 寄存器, 将清除 NE 标志位。

19.3.3.7. 帧错误

当以下情况发生时检测到帧错误:

由于没有同步上或大量噪音的原因, 停止位没有在预期的时间上接和收识别出来。当帧错误被检测到时:

- FE 位被硬件置起
- 无效数据从移位寄存器传送到 USART_DR 寄存器。
- 在单字节通信时, 没有中断产生。然而, 这个位和 RXNE 位同时置起, 后者将产生中断。在多缓冲器通信情况下, 如果 USART_CR3 寄存器中 EIE 位被置位的话, 将产生中断。

顺序执行对 USART_SR 和 USART_DR 寄存器的读操作, 可复位 FE 位。

19.3.3.8. 接收期间的可配置的停止位

被接收的停止位的个数可以通过控制寄存器 2 的控制位来配置, 在正常模式时, 可以是 1 或 2 个, 在智能卡模式里可能是 0.5 或 1.5 个。

• 0.5 个停止位 (智能卡模式中的接收): 不对 0.5 个停止位进行采样。因此, 如果选择 0.5 个停止位则不能检测帧错误和断开帧。

• 1 个停止位: 对 1 个停止位的采样在第 8, 第 9 和第 10 采样点上进行。

• 1.5 个停止位 (智能卡模式): 当以智能卡模式发送时, 器件必须检查数据是否被正确的发送出去。所以接收器功能块必须被激活 (USART_CR1 寄存器中的 RE =1), 并且在停止位的发送期间采样数据线上的信号。如果出现校验错误, 智能卡会在发送方采样 NACK 信号时, 即总线上停止位对应的时间内时, 拉低数据线, 以此表示出现了帧错误。FE 在 1.5 个停止位结束时和 RXNE 一起被置起。对 1.5 个停止位的采样是在第 16, 第 17 和第 18 采样点进行的。1.5 个的停止位可以被分成 2 部分: 一个是 0.5 个时钟周期, 期间不做任何事情。随后是 1 个时钟周期的停止位, 在这段时间的中点处采样。

• 2 个停止位: 对 2 个停止位的采样是在第一停止位的第 8, 第 9 和第 10 个采样点完成的。如果第一个停止位期间检测到一个帧错误, 帧错误标志将被设置。第二个停止位不再检查帧错误。在第一个停止位结束时 RXNE 标志将被设置。

19.3.4. 分数波特率的产生

接收器和发送器的波特率在 USARTDIV 的整数和小数寄存器中的值应设置成相同。

$$\text{Tx / Rx 波特率} = \frac{f_{CK}}{(16 * USARTDIV)}$$

这里的 fCK 是给外设的时钟 (PCLK1 用于 USART2、3、4、5, PCLK2 用于 USART1)

USARTDIV 是一个无符号的定点数。这 12 位的值设置在 USART_BRR 寄存器。

注: 在写入 USART_BRR 之后, 波特率计数器会被波特率寄存器的新值替换。因此, 不要在通信进行中改变波特率寄存器的数值。

如何从 USART_BRR 寄存器值得到 USARTDIV

例 1:

如果 DIV_Mantissa = 27, DIV_Fraction = 12 (USART_BRR=0x1BC),
于是:
Mantissa (USARTDIV) = 27
Fraction (USARTDIV) = 12/16 = 0.75
所以 USARTDIV = 27.75

例 2:

要求 USARTDIV = 25.62,
就有:
DIV_Fraction = 16*0.62 = 9.92
最接近的整数是: 10 = 0x0A
DIV_Mantissa = mantissa (25.620) = 25 = 0x19
于是, USART_BRR = 0x19A

例 3:

要求 USARTDIV = 50.99
就有:
DIV_Fraction = 16*0.99 = 15.84
最接近的整数是: 16 = 0x10 => DIV_frac[3:0]溢出 => 进位必须加到小数部分
DIV_Mantissa = mantissa (50.990 + 进位) = 51 = 0x33
于是: USART_BRR = 0x330, USARTDIV=51

19.3.5. USART 接收器容忍时钟的变化

只有当整体的时钟系统地变化小于 USART 异步接收器能够容忍的范围, USART 异步接收器才能正常地工作。影响这些变化的因素有:

- DTRA: 由于发送器误差而产生的变化 (包括发送器端振荡器的变化)
- DQUANT: 接收器端波特率取整所产生的误差
- DREC: 接收器端振荡器的变化
- DTCL: 由于传输线路产生的变化 (通常是由于收发器在由低变高的转换时序, 与由高变低转换时序之间的不一致性所造成)。

需要满足: DTRA + DQUANT + DREC + DTCL < USART 接收器的容忍度

对于正常接收数据, USART 接收器的容忍度等于最大能容忍的变化, 它依赖于下述选择:

- 由 USART_CR1 寄存器的 M 位定义的 10 或 11 位字符长度
- 是否使用分数波特率产生

表163. 当 DIV_Fraction=0 时, USART 接收器的容忍度

M 位	认为 NF 是错误	不认为 NF 是错误
0	3.75%	4.375%
1	3.41%	3.97%

表164. 当 DIV_Fraction!=0 时, USART 接收器的容忍度

M 位	认为 NF 是错误	不认为 NF 是错误
0	3.33%	3.88%
1	3.03%	3.53%

注: 在特殊的情况下, 即当收到的帧包含一些在 M=0 时, 正好是 10 位 (M=1 时是 11 位) 的空闲帧, 上面 2 个表格中的数据可能会有少许出入。

19.3.6. 多处理器通信

通过 USART 可以实现多处理器通信 (将几个 USART 连在一个网络里)。例如某个 USART 设备可以是主, 它的 TX 输出和其他 USART 从设备的 RX 输入相连接; USART 从设备各自的 TX 输出逻辑地与在一起, 并且和主设备的 RX 输入相连接。

在多处理器配置中, 我们通常希望只有被寻址的接收者才被激活, 来接收随后的数据, 这样就可以减少由未被寻址的接收器的参与带来的多余的 USART 服务开销。

未被寻址的设备可启用其静默功能置于静默模式。在静默模式里:

- 任何接收状态位都不会被设置。
- 所有接收中断被禁止。
- USART_CR1 寄存器中的 RWU 位被置 1。RWU 可以被硬件自动控制或在某个条件下由软件写入。

根据 USART_CR1 寄存器中的 WAKE 位状态, USART 可以用二种方法进入或退出静默模式。

- 如果 WAKE 位被复位: 进行空闲总线检测。
- 如果 WAKE 位被设置: 进行地址标记检测。

19.3.6.1. 空闲总线检测 (WAKE=0)

当 RWU 位被写 1 时, USART 进入静默模式。当检测到一空闲帧时, 它被唤醒。然后 RWU 被硬件清零, 但是 USART_SR 寄存器中的 IDLE 位并不置起。RWU 还可以被软件写 0。下图给出利用空闲总线检测来唤醒和进入静默模式的一个例子。

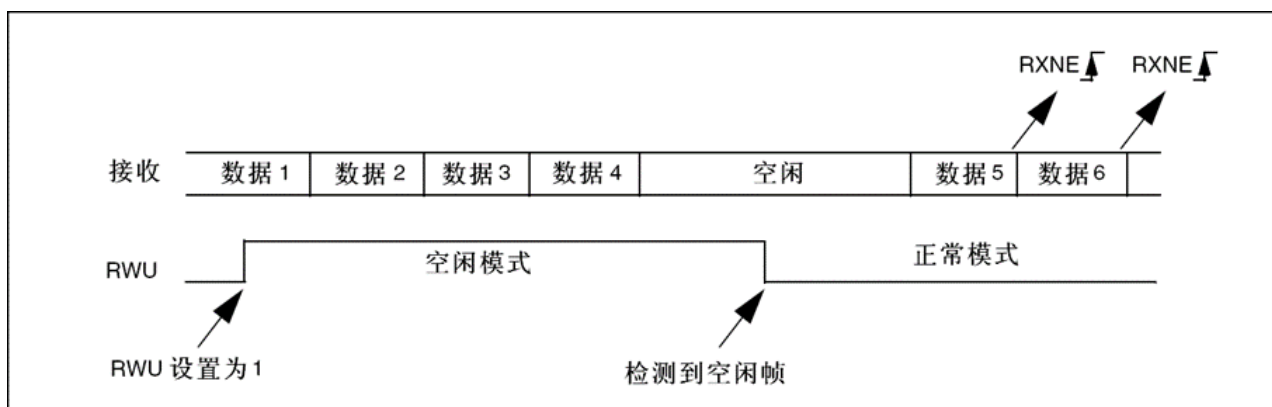


图117. 利用空闲总线检测的静默模式

19.3.6.2. 地址标记 (address mark) 检测 (WAKE=1)

在这个模式里, 如果 MSB 是 1, 该字节被认为是地址, 否则被认为是数据。在一个地址字节中, 目标接收器的地址被放在 4 个 LSB 中。这个 4 位地址被接收器同它自己地址做比较, 接收器的地址被编程在 USART_CR2 寄存器的 ADD。

如果接收到的字节与它的编程地址不匹配时, USART 进入静默模式。此时, 硬件设置 RWU 位。接收该字节既不会设置 RXNE 标志也不会产生中断或发出 DMA 请求, 因为 USART 已经在静默模式。

当接收到的字节与接收器内编程地址匹配时, USART 退出静默模式。然后 RWU 位被清零, 随后的字节被正常接收。收到这个匹配的地址字节时将设置 RXNE 位, 因为 RWU 位已被清零。

当接收缓冲器不包含数据时 (USART_SR 的 RXNE=0), RWU 位可以被写 0 或 1。否则, 该次写操作被忽略。下图给出利用地址标记检测来唤醒和进入静默模式的例子。

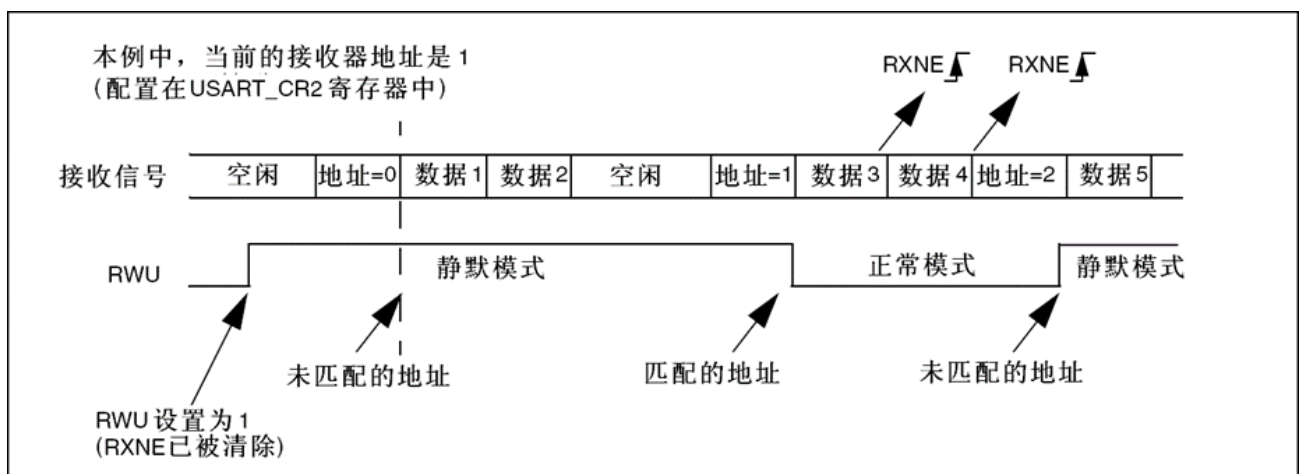


图118. 利用地址标记检测的静默模式

19.3.7. 校验控制

设置 USART_CR1 寄存器上的 PCE 位, 可以使能奇偶控制 (发送时生成一个奇偶位, 接收时进行奇偶校验)。根据 M 位定义的帧长度, 可能的 USART 帧格式列在下表中。

表165. 帧格式

M 位	PCE 位	USART 帧
0	0	起始位 8 位数据 停止位
0	1	起始位 7 位数据 奇偶检验位 停止位
1	0	起始位 9 位数据 停止位
1	1	起始位 8 位数据 奇偶检验位 停止位

注: 在用地址标记唤醒设备时, 地址的匹配只考虑到数据的 MSB 位, 而不用关心校验位。(MSB 是数据位中最后发出的, 后面紧跟校验位或者停止位)

19.3.7.1. 偶校验:

校验位使得一帧中的 7 或 8 个 LSB 数据以及校验位中'1'的个数为偶数。

例如: 数据=00110101, 有 4 个'1', 如果选择偶校验 (在 USART_CR1 中的 PS = 0), 校验位将是'0'。

19.3.7.2. 奇校验:

此校验位使得一帧中的 7 或 8 个 LSB 数据以及校验位中'1'的个数为奇数。

例如: 数据=00110101, 有 4 个'1', 如果选择奇校验 (在 USART_CR1 中的 PS = 1), 校验位将是'1'。

19.3.7.3. 传输模式:

如果 USART_CR1 的 PCE 位被置位, 写进数据寄存器的数据的 MSB 位被校验位替换后发送出去 (如果选择偶校验偶数个'1', 如果选择奇校验奇数个'1')。如果奇偶校验失败, USART_SR 寄存器中的 PE 标志被置'1', 并且如果 USART_CR1 寄存器的 PEIE 在被预先设置的话, 中断产生。

19.3.8. LIN (局域互联网) 模式

LIN 模式是通过设置 USART_CR2 寄存器的 LINEN 位选择。在 LIN 模式下, 下列位必须保持为 0:

- USART_CR2 寄存器的 CLKEN 位
- USART_CR3 寄存器的 STOP[1:0], SCEN, HDSEL 和 IREN

19.3.8.1. LIN 发送

10.5.2 节里所描述的同样步骤适用于 LIN 主发送, 但和正常 USART 发送有以下区别:

- 清零 M 位以配置 8 位字长
- 置位 LINEN 位以进入 LIN 模式。这时, 置位 SBK 将发送 13 位'0'作为断开符号。然后发一位'1', 以允许对下一个开始位的检测。

19.3.8.2. LIN 接收

当 LIN 模式被使能时, 断开符号检测电路被激活。该检测完全独立于 USART 接收器。断开符号只要一出现就能检测到, 不管是在总线空闲时还是在发送某数据帧其间, 数据帧还未完成, 又插入了断开符号的发送。

当接收器被激活时 (USART_CR1 的 RE=1), 电路监测 RX 上的起始信号。监测起始位的方法同检测断开符号或数据是一样的。当起始位被检测到后, 电路对每个接下来的位, 在每个位的第 8, 9, 10 个过采样时钟点上进行采样。如果 10 个 (当 USART_CR2 的 LBDL = 0) 或 11 个 (当 USART_CR2 的 LBDL = 1) 连续位都是'0', 并且又跟着一个定界符, USART_SR 的 LBD 标志被设置。如果 LBDIE 位=1, 中断产生。在确认断开符号前, 要检查定界符, 因为它意味 RX 线已经回到高电平。

如果在第 10 或 11 个采样点之前采样到了'1', 检测电路取消当前检测并重新寻找起始位。如果 LIN 模式被禁止, 接收器继续如正常 USART 那样工作, 不需要考虑检测断开符号。

如果 LIN 模式没有被激活 (LINEN=0), 接收器仍然正常工作于 USART 模式, 不会进行断开检测。

如果 LIN 模式被激活 (LINEN=1), 只要一发生帧错误 (也就是停止位检测到'0', 这种情况出现在断开帧), 接收器就停止, 直到断开符号检测电路接收到一个'1' (这种情况发生于断开符号没有完整的发出来), 或一个定界符 (这种情况发生于已经检测到一个完整的断开符号)。

图 119 说明了断开符号检测器状态机的行为和断开符号标志的关系。图 120 给出了一个断开帧的例子。

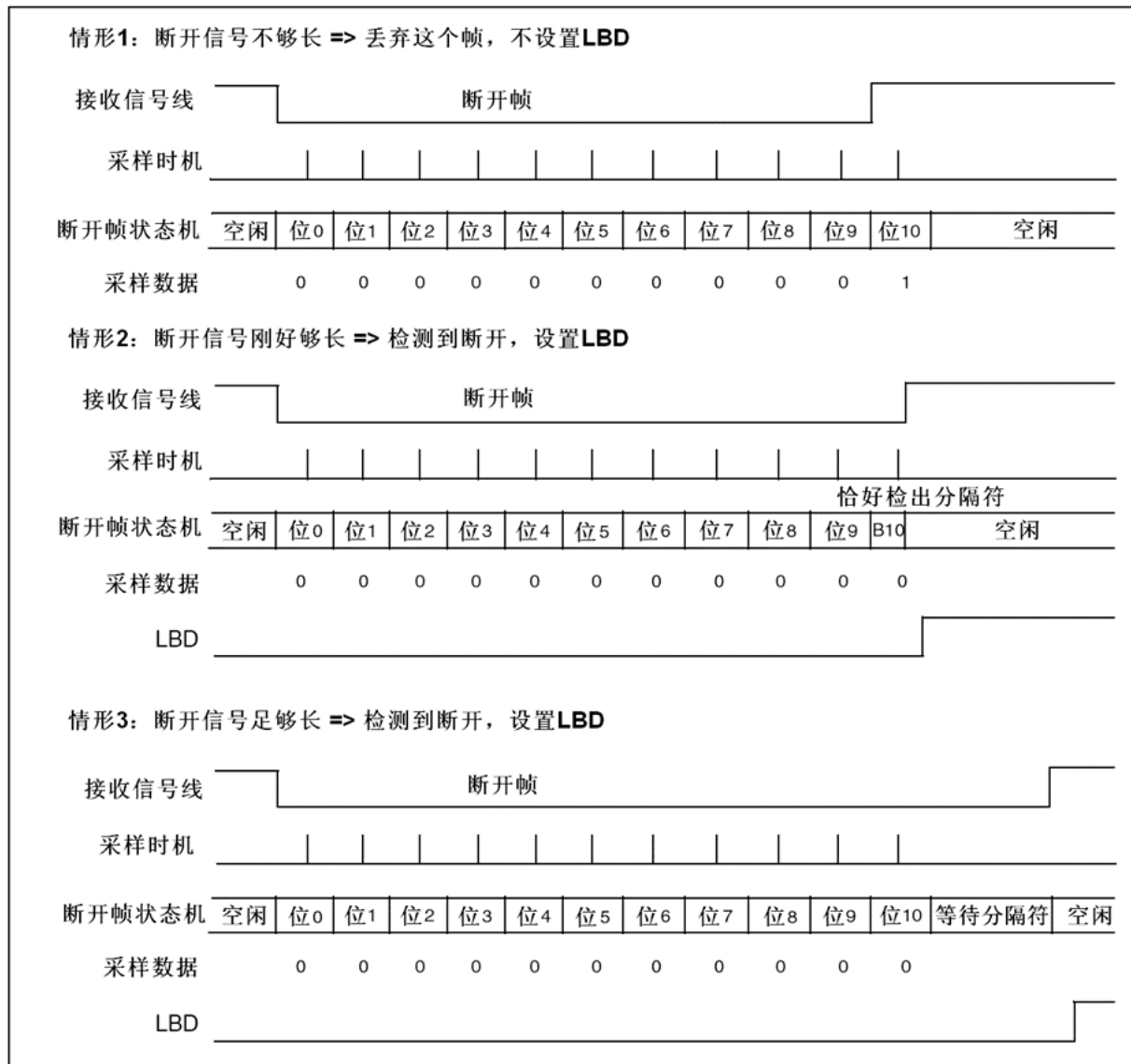


图119. LIN 模式下的断开检测 (11 位断开长度 - 设置了 LBDL 位)

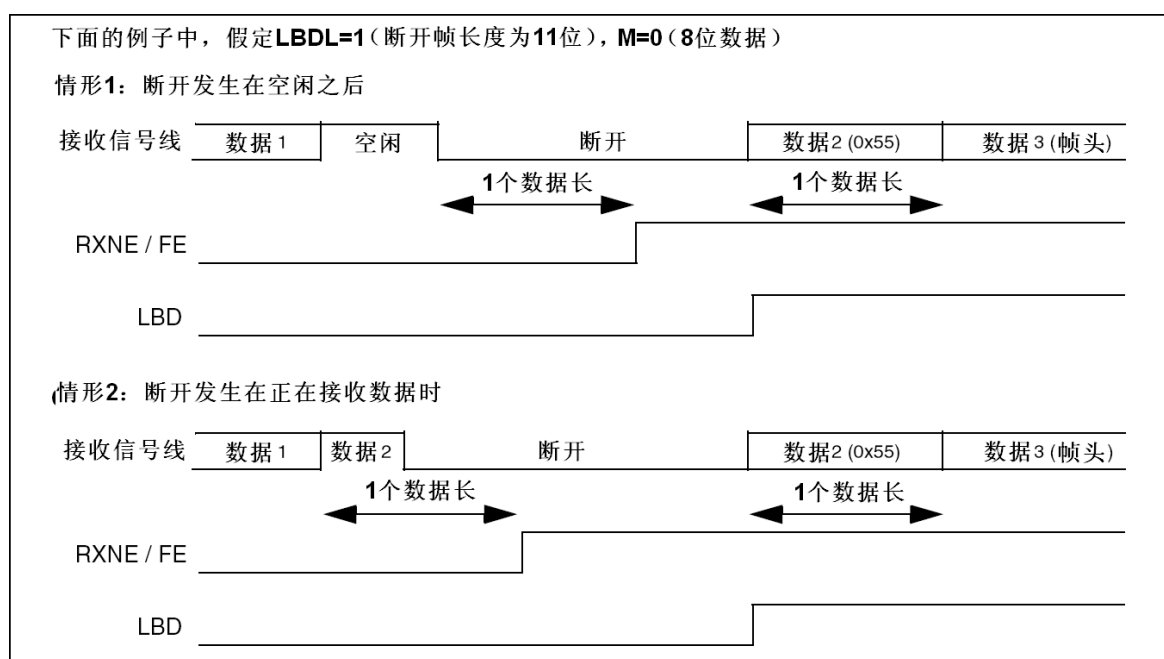


图120. LIN 模式下的断开检测与帧错误的检测

19.3.9. USART 同步模式

通过在 USART_CR2 寄存器上写 CLKEN 位选择同步模式在同步模式里，下列位必须保持清零状态：

- USART_CR2 寄存器中的 LINEN 位
- USART_CR3 寄存器中的 SCEN, HDSEL 和 IREN 位

USART 允许用户以主模式方式控制双向同步串行通信。CK 脚是 USART 发送器时钟的输出。在起始位和停止位期间，CK 脚上没有时钟脉冲。根据 USART_CR2 寄存器中 LBCL 位的状态，决定在最后一个有效数据位期间产生或不产生时钟脉冲。USART_CR2 寄存器的 CPOL 位允许用户选择时钟极性，USART_CR2 寄存器上的 CPHA 位允许用户选择外部时钟的相位。

在总线空闲期间，实际数据到来之前以及发送断开符号的时候，外部 CK 时钟不被激活。

同步模式时，USART 发送器和异步模式里工作一模一样。但是因为 CK 是与 TX 同步的（根据

CPOL 和 CPHA），所以 TX 上的数据是随 CK 同步发出的。

同步模式的 USART 接收器工作方式与异步模式不同。如果 RE=1，数据在 CK 上采样（根据 CPOL 和 CPHA 决定在上升沿还是下降沿），不需要任何的过采样。但必须考虑建立时间和持续时间（取决于波特率，1/16 位时间）。

注：

1. CK 脚同 TX 脚一起联工作。因而，只有在使能了发送器 (TE = 1)，并且发送数据时 (写入数据至 USART_DR 寄存器) 才提供时钟。这意味着在没有发送数据时是不可能接收一个同步数据的。
2. LBCL, CPOL 和 CPHA 位的正确配置，应该在发送器和接收器都被禁止时；当使能了发送器或接收器时，这些位不能被改变
3. 建议在同一条指令中设置 TE 和 RE，以减少接收器的建立时间和保持时间。
4. USART 只支持主模式：它不能来自其他设备的输入时钟接收或发送数据 (CK 永远是输出)。

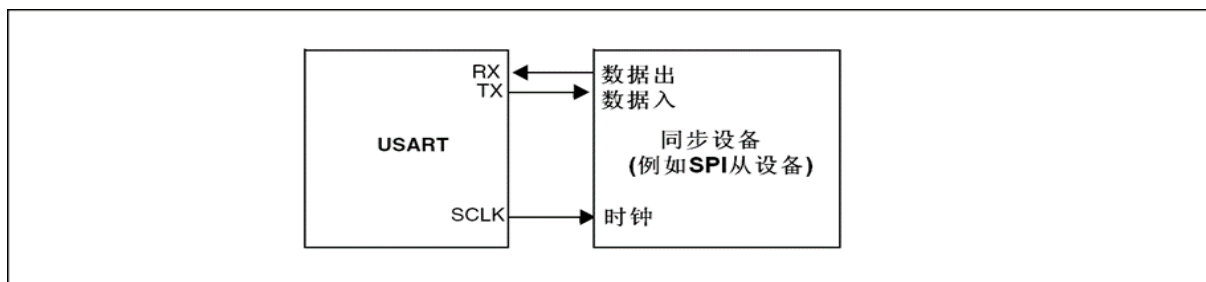


图121. USART 同步传输的例子

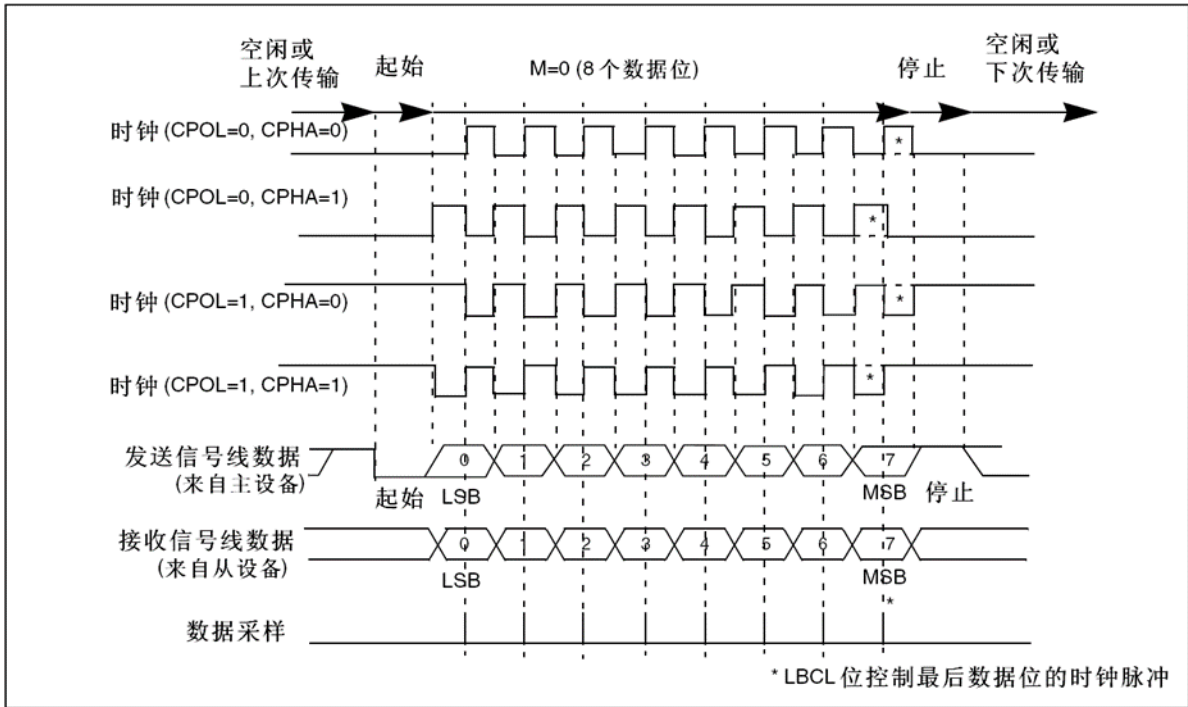


图122. USART 数据时钟时序示例 (M=0)

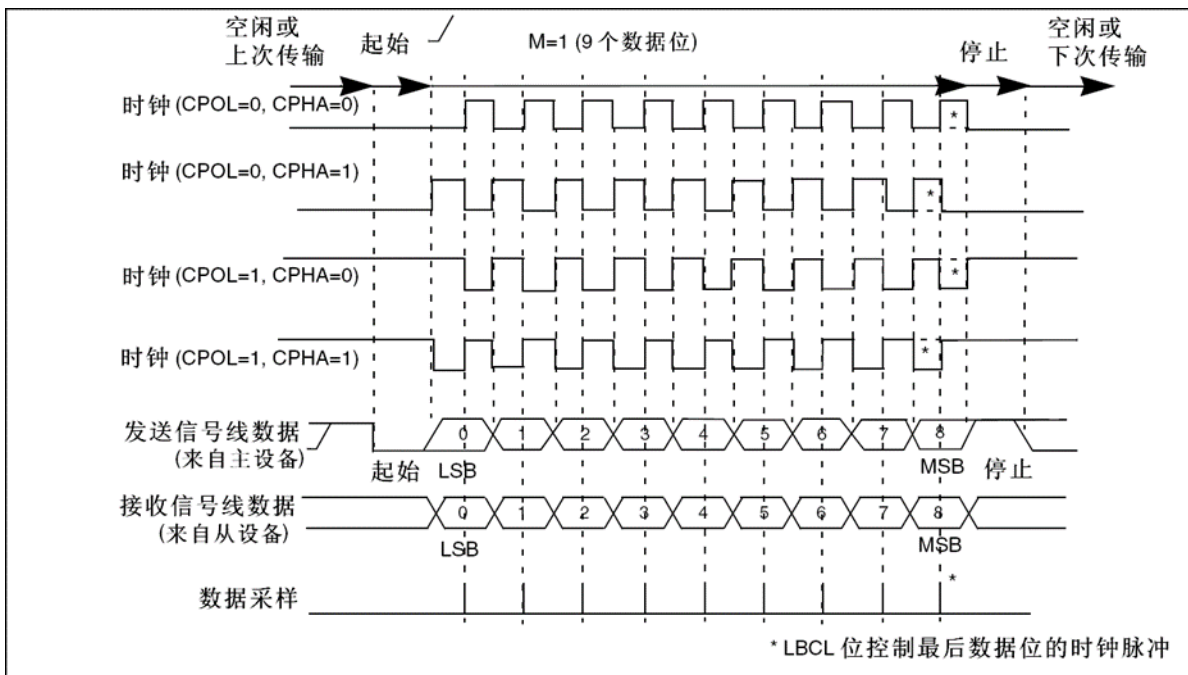


图123. USART 数据时钟时序示例 (M=1)

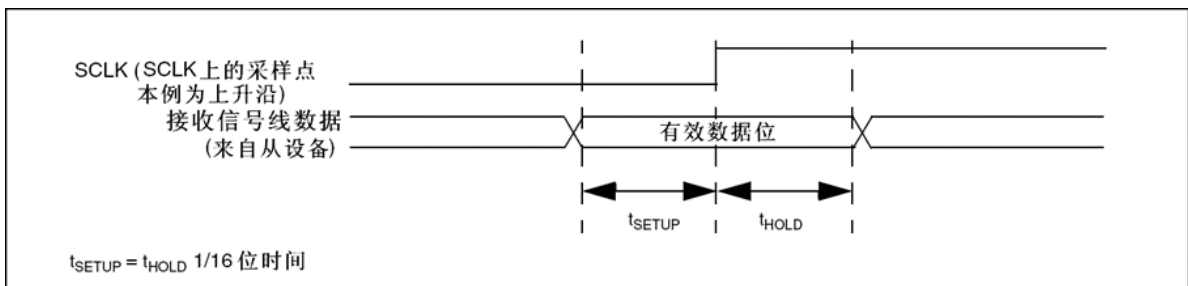


图124. RX 数据采样/保持时间

注：在智能卡模式下 CK 的功能不同，有关细节请参考智能卡模式部分。

19.3.10. 单线半双工通信

单线半双工模式通过设置 USART_CR3 寄存器的 HDSEL 位选择。在这个模式里，下面的位必须保持清零状态：

- USART_CR2 寄存器的 LINEN 和 CLKEN 位
- USART_CR3 寄存器的 SCEN 和 IREN 位

USART 可以配置成遵循单线半双工协议。在单线半双工模式下，TX 和 RX 引脚在芯片内部互连。使用控制位“HALF DUPLEX SEL”（USART_CR3 中的 HDSEL 位）选择半双工和全双工通信。

当 HDSEL 为'1'时

- RX 不再被使用
- 当没有数据传输时，TX 总是被释放。因此，它在空闲状态的或接收状态时表现为一个标准 I/O 口。这就意味该 I/O 在不被 USART 驱动时，必须配置成悬空输入（或开漏的输出高）。

除此以外，通信与正常 USART 模式类似。由软件来管理线上的冲突（例如通过使用一个中央仲裁器）。特别的是，发送从不会被硬件所阻碍。当 TE 位被设置时，只要数据一写到数据寄存器上，发送就继续。

19.3.11. 智能卡

设置 USART_CR3 寄存器的 SCEN 位选择智能卡模式。在智能卡模式下，下列位必须保持清零：

- USART_CR2 寄存器的 LINEN 位
- USART_CR3 寄存器的 HDSEL 位和 IREN 位

此外，CLKEN 位可以被设置，以提供时钟给智能卡。

该接口符合 ISO7816-3 标准，支持智能卡异步协议。USART 应该被设置为：

- 8 位数据位加校验位：此时 USART_CR1 寄存器中 M=1、PCE=1
- 发送和接收时为 1.5 个停止位：即 USART_CR2 寄存器的 STOP=11

注：也可以在接收时选择 0.5 个停止位，但为了避免在 2 种配置间转换，建议在发送和接收时使用 1.5 个停止位。

下图给出的例子说明了数据线上，在有校验错误和没校验错误两种情况下的信号。

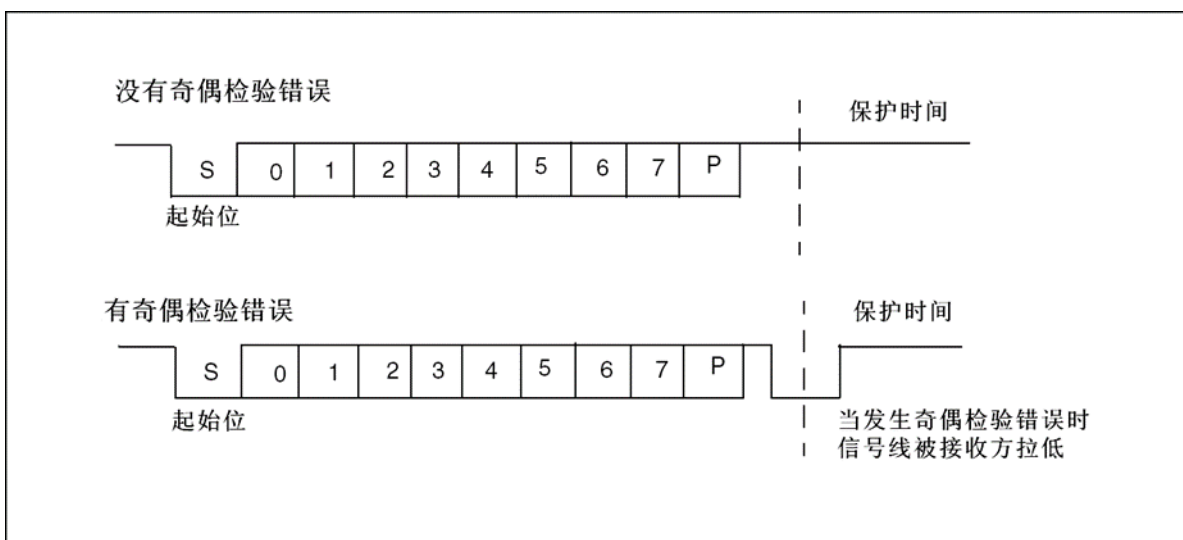


图125. ISO7816-3 异步协议

当与智能卡相连接时，USART 的 TX 驱动一根智能卡也驱动的双向线。为了做到这点，SW_RX 必须和 TX 连

接到相同的 I/O 口。在发送开始位和数据字节期间，发送器的输出使能位 TX_EN 被置起，在发送停止位期间被释放（弱上拉），因此在发现校验错误的情况下接收器可以将数据线拉低。如果 TX_EN 不被使用，在停止位期间 TX 被拉到高电平：这样的话，只要 TX 配置成开漏，接收器也可以驱动这根线。

智能卡是一个单线半双工通信协议

- 从发送移位寄存器把数据发送出去，要被延时最小 1/2 波特时钟。在正常操作时，一个满的发送移位寄存器将在下一个波特时钟沿开始向外移出数据。在智能卡模式里，此发送被延迟 1/2 波特时钟。
- 如果在接收一个设置为 0.5 或 1.5 个停止位的数据帧期间，检测到一奇偶校验错误，在完成接收该帧后（即停止位结束时），发送线被拉低一个波特时钟周期。这是告诉智能卡发送到 USART 的数据没有被正确地接收到。此 NACK 信号（拉低发送线一个波特时钟周期）在发送端将产生一个帧错误（发送端被配置成 1.5 个停止位）。应用程序可以根据协议处理重新发送数据。如果设置了 NACK 控制位，发生校验错误时接收器会给出一个 NACK 信号；否则就不会发送 NACK。
- TC 标志的置起可以通过编程保护时间寄存器得以延时。在正常操作时，当发送移位寄存器变空并且没有新的发送请求出现时，TC 被置起。在智能卡模式里，空的发送移位寄存器将触发保护时间计数器开始向上计数，直到保护时间寄存器中的值。TC 在这段时间被强制拉低。当保护时间计数器达到保护时间寄存器中的值时，TC 被置高。
- TC 标志的撤销不受智能卡模式的影响。
- 如果发送器检测到一个帧错误（收到接收器的 NACK 信号），发送器的接收功能模块不会把 NACK 当作起始位检测。根据 ISO 协议，接收到的 NACK 的持续时间可以是 1 或 2 波特时钟周期。
- 在接收器这边，如果一个校验错误被检测到，并且 NACK 被发送，接收器不会把 NACK 检测成起始位。

注：

1. 断开符号在智能卡模式里没有意义。一个带帧错误的 00h 数据将被当成数据而不是断开符号。
2. 当来回切换 TE 位时，没有 IDLE 帧被发送。ISO 协议没有定义 IDLE 帧。

下图详述了 USART 是如何采样 NACK 信号的。在这个例子里，USART 正在发送数据，并且被配置成 1.5 个停止位。为了检查数据的完整性和 NACK 信号，USART 的接收功能块被激活。

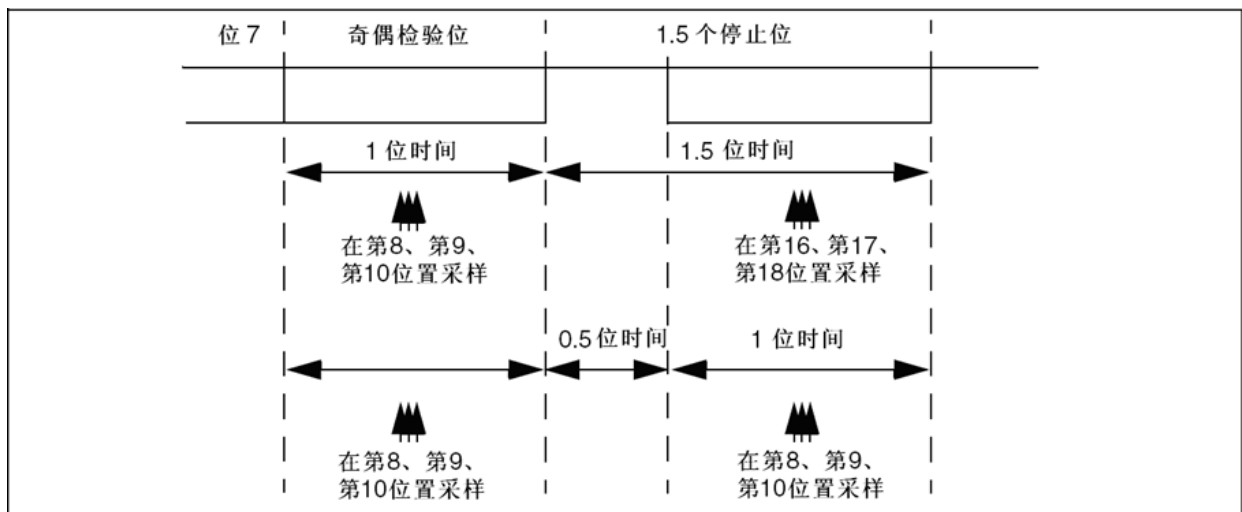


图126. 使用 1.5 停止位检测奇偶检验错

USART 可以通过 CK 输出为智能卡提供时钟。在智能卡模式里，CK 不和通信直接关联，而是先通过一个 5 位预分频器简单地用内部的外设输入时钟来驱动智能卡的时钟。分频率在预分频寄存器 USART_GTPR 中配置。CK 频率可以从 fCK/2 到 fCK/62，这里的 fCK 是外设输入时钟。

19.3.12. IrDA SIR ENDEC 功能模块

通过设置 USART_CR3 寄存器的 IREN 位选择 IrDA 模式。在 IrDA 模式里，下列位必须保持清零：

- USART_CR2 寄存器的 LINEN, STOP 和 CLKEN 位
- USART_CR3 寄存器的 SCEN 和 HDSEL 位。

IrDA SIR 物理层规定使用反相归零调制方案 (RZI)，该方案用一个红外光脉冲代表逻辑'0'。SIR 发送编码器对从 USART 输出的 NRZ (非归零) 比特流进行调制。输出脉冲流被传送到一个外部输出驱动器和红外 LED。USART 为 SIR ENDEC 最高只支持到 115.2Kbps 速率。在正常模式里，脉冲宽度规定为一个位周期的 3/16。

SIR 接收解码器对来自红外接收器的归零位比特流进行解调，并将接收到的 NRZ 串行比特流输出到 USART。在空闲状态里，解码器输入通常是高 (标记状态 marking state)。发送编码器输出的极性和解码器的输入相反。当解码器输入低时，检测到一个起始位。

• IrDA 是一个半双工通信协议。如果发送器忙 (也就是 USART 正在送数据给 IrDA 编码器)，IrDA 接收线上的任何数据将被 IrDA 解码器忽视。如果接收器忙 (也就是 USART 正在接收从 IrDA 解码器来的解码数据)，从 USART 到 IrDA 的 TX 上的数据将不会被 IrDA 编码。当接收数据时，应该避免发送，因为将被发送的数据可能被破坏。

- SIR 发送逻辑把'0'作为高脉冲发送，把'1'作为低电平发送。脉冲的宽度规定为正常模式时位周期的 3/16。
- SIR 接收逻辑把高电平状态解释为'1'，把低脉冲解释为'0'。
- 发送编码器输出与解码器输入有着相反的极性。当空闲时，SIR 输出处于低状态。
- SIR 解码器把 IrDA 兼容的接收信号转变成给 USART 的比特流。

• IrDA 规范要求脉冲要宽于 1.41us。脉冲宽度是可编程的。接收器端的尖峰脉冲检测逻辑滤除宽度小于 2 个 PSC 周期的脉冲 (PSC 是在 IrDA 低功耗波特率寄存器 USART_GTPR 中编程的预分频值)。宽度小于 1 个 PSC 周期的脉冲一定被滤除掉，但是那些宽度大于 1 个而小于 2 个 PSC 周期的脉冲可能被接收或滤除，那些宽度大于 2 个周期的将被视为一个有效的脉冲。当 PSC=0 时，IrDA 编码器/解码器不工作。

- 接收器可以与一低功耗发送器通信。
- 在 IrDA 模式里，USART_CR2 寄存器上的 STOP 位必须配置成 1 个停止位。

19.3.12.1. IrDA 低功耗模式

19.3.12.2. 发送器

在低功耗模式，脉冲宽度不再持续 3/16 个位周期。取而代之，脉冲的宽度是低功耗波特率的 3 倍，它最小可以是 1.42MHz。通常这个值是 1.8432MHz ($1.42 \text{ MHz} < \text{PSC} < 2.12 \text{ MHz}$)。一个低功耗模式可编程分频器把系统时钟进行分频以达到这个值。

19.3.12.3. 接收器

低功耗模式的接收类似于正常模式的接收。为了滤除尖峰干扰脉冲，USART 应该滤除宽度短于 1 个 PSC 的脉冲。只有持续时间大于 2 个周期的 IrDA 低功耗波特率时钟 (USART_GTPR 中的 PSC) 的低电平信号才被接受为有效的信号。

注：

1. 宽度小于 2 个大于 1 个 PSC 周期的脉冲可能会也可能不会被滤除。
2. 接收器的建立时间应该由软件管理。IrDA 物理层技术规范规定了在发送和接收之间最小要有 10ms 的延时 (IrDA 是一个半双工协议)。

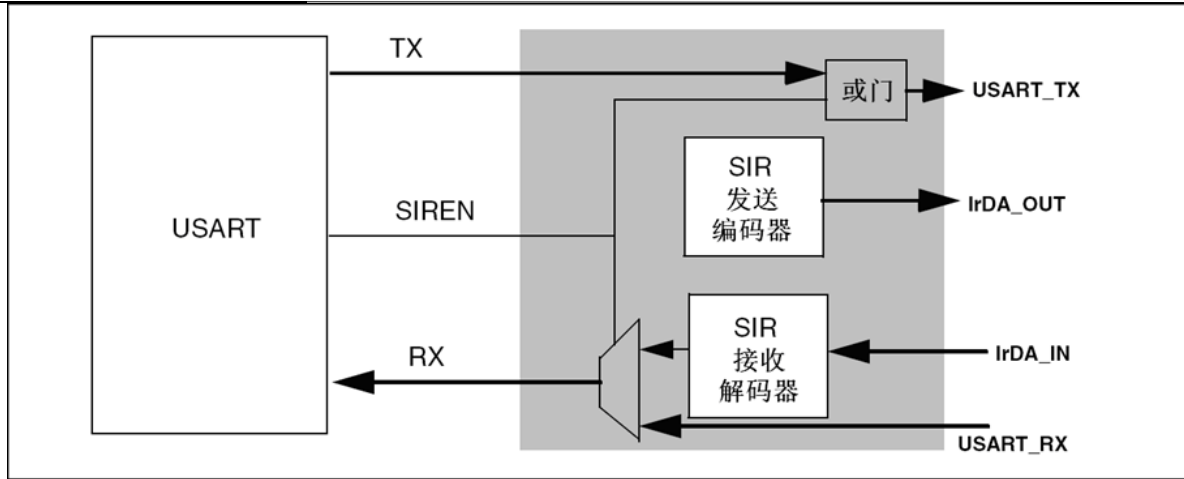


图127. IrDA SIR ENDEC-框图

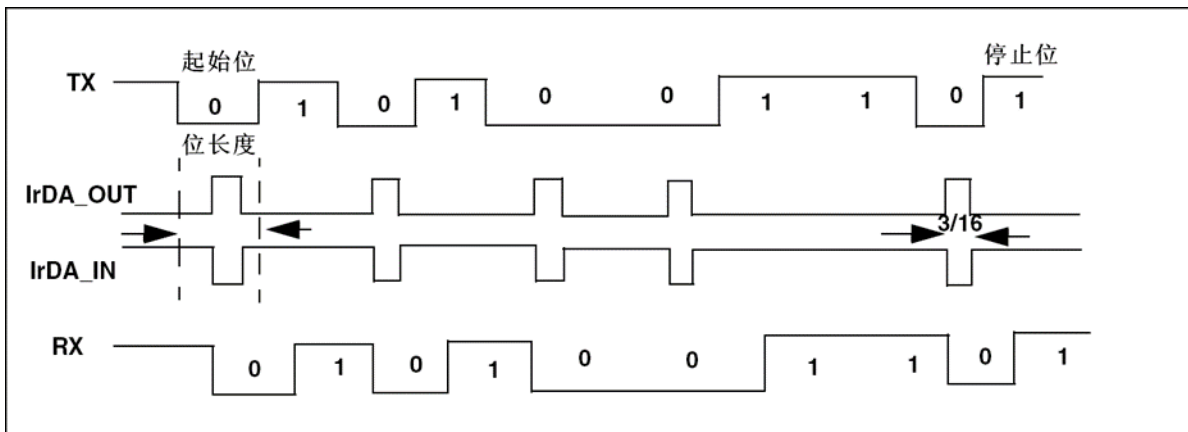


图128. IrDA 数据调制 (3/16) -普通模式

19.3.13. 利用 DMA 连续通信

USART 可以利用 DMA 连续通信。Rx 缓冲器和 Tx 缓冲器的 DMA 请求是分别产生的。

注：参考产品技术说明以确定是否可用 DMA 控制器。如果所用产品无 DMA 功能，应按 25.3.2 节或 25.3.3 节里所描述的方法使用 USART。在 USART2_SR 寄存器里，可以清零 TXE/RXNE 标志来实现连续通信。

19.3.13.1. 利用 DMA 发送

使用 DMA 进行发送，可以通过设置 USART_CR3 寄存器上的 DMAT 位激活。当 TXE 位被置为 '1' 时，DMA 就从指定的 SRAM 区传送数据到 USART_DR 寄存器。为 USART 的发送分配一个 DMA 通道的步骤如下（x 表示通道号）：

- 1) 在 DMA 控制寄存器上将 USART_DR 寄存器地址配置成 DMA 传输的目的地址。在每个 TXE 事件后，数据将被传送到这个地址。
- 2) 在 DMA 控制寄存器上将存储器地址配置成 DMA 传输的源地址。在每个 TXE 事件后，将从此存储器区读出数据并传送到 USART_DR 寄存器。
- 3) 在 DMA 控制寄存器中配置要传输的总的字节数。
- 4) 在 DMA 寄存器上配置通道优先级。
- 5) 根据应用程序的要求，配置在传输完成一半还是全部完成时产生 DMA 中断。
- 6) 在 DMA 寄存器上激活该通道。

当传输完成 DMA 控制器指定的数据量时，DMA 控制器在该 DMA 通道的中断向量上产生中断。

在发送模式下，当 DMA 传输完所有要发送的数据时，DMA 控制器设置 DMA_ISR 寄存器的 TCIF 标志；监

视 USART_SR 寄存器的 TC 标志可以确认 USART 通信是否结束，这样可以在关闭 USART 或进入停机模式之前避免破坏最后一次传输的数据；软件需要先等待 TXE=1，再等待 TC=1。

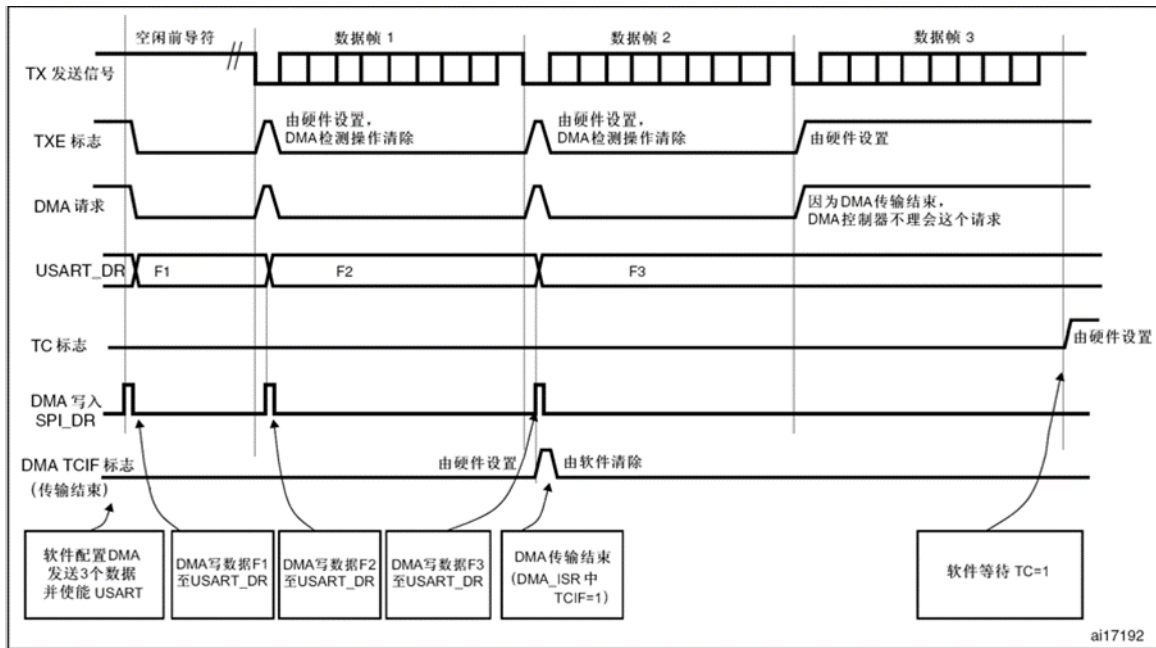


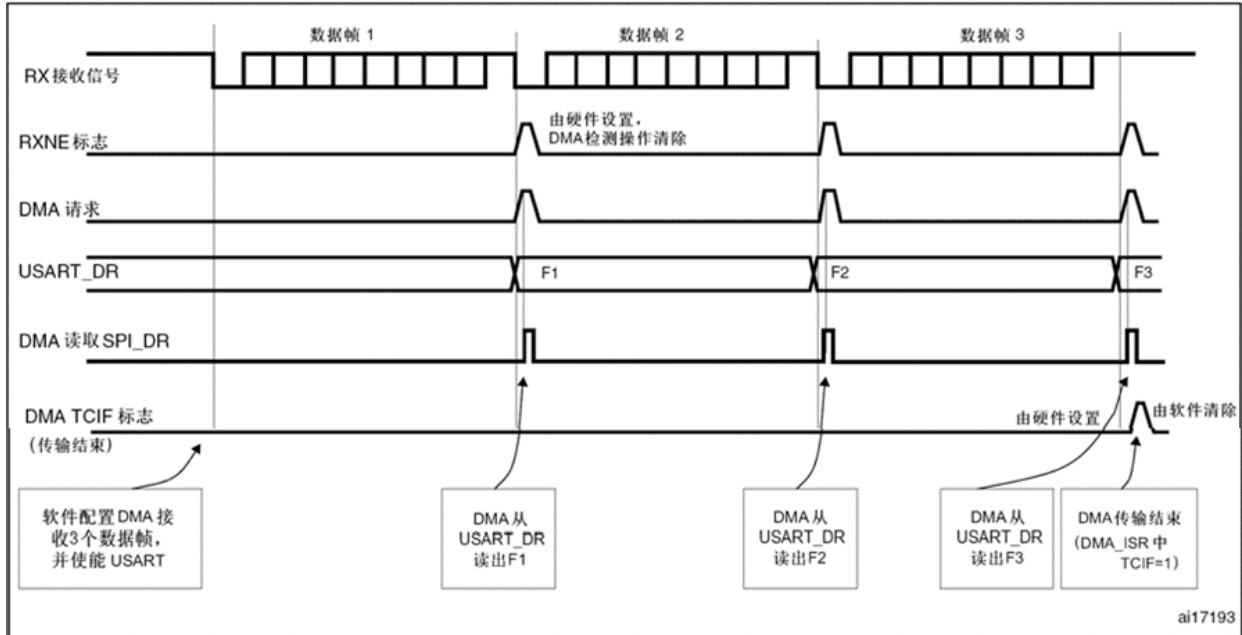
图129. 利用 DMA 传送

19.3.13.2. 利用 DMA 接收

可以通过设置 USART_CR3 寄存器的 DMAR 位激活使用 DMA 进行接收，每次接收到一个字节，DMA 控制器就把数据从 USART_DR 寄存器传送到指定的 SRAM 区（参考 DMA 相关说明）。为 USART 的接收分配一个 DMA 通道的步骤如下（x 表示通道号）：

- 1) 通过 DMA 控制寄存器把 USART_DR 寄存器地址配置成传输的源地址。在每个 RXNE 事件后，将从此地址读出数据并传输到存储器。
- 2) 通过 DMA 控制寄存器把存储器地址配置成传输的目的地址。在每个 RXNE 事件后，数据将从 USART_DR 传输到此存储器区。
- 3) 在 DMA 控制寄存器中配置要传输的总的字节数。
- 4) 在 DMA 寄存器上配置通道优先级。。
- 5) 根据应用程序的要求配置在传输完成一半还是全部完成时产生 DMA 中断。
- 6) 在 DMA 控制寄存器上激活该通道。

当接收完成 DMA 控制器指定的传输量时，DMA 控制器在该 DMA 通道的中断矢量上产生中断。

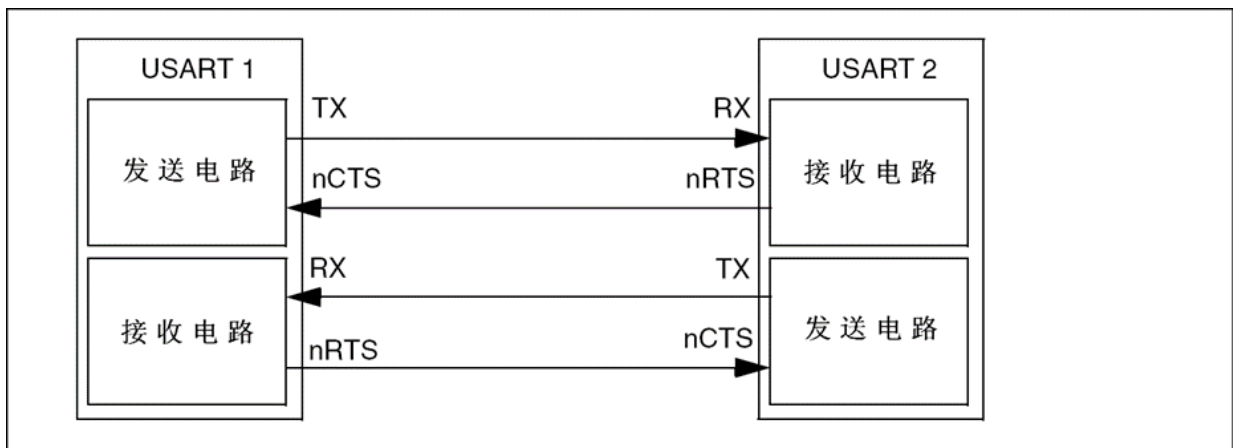


19.3.13.3. 多缓冲器通信中的错误标志和中断产生

在多缓冲器通信的情况下，通信期间如果发生任何错误，在当前字节传输后将置起错误标志。如果中断使能位被设置，将产生中断。在单个字节接收的情况下，和 RXNE 一起被置起的帧错误、溢出错误和噪音标志，有单独的的错误标志中断使能位；如果设置了，会在当前字节传输结束后，产生中断。

19.3.14. 硬件流控制

利用 nCTS 输入和 nRTS 输出可以控制 2 个设备间的串行数据流。下图表明在这个模式里如何连接 2 个设备。



通过将 UASRT_CR3 中的 RTSE 和 CTSE 置位，可以分别独立地使能 RTS 和 CTS 流控制。

19.3.14.1. RTS 流控制

如果 RTS 流控制被使能 (RTSE=1)，只要 USART 接收器准备好接收新的数据，nRTS 就变成有效（接低电平）。当接收寄存器内有数据到达时，nRTS 被释放，由此表明希望当前帧结束时停止数据传输。下图是一个启用 RTS 流控制的通信的例子。

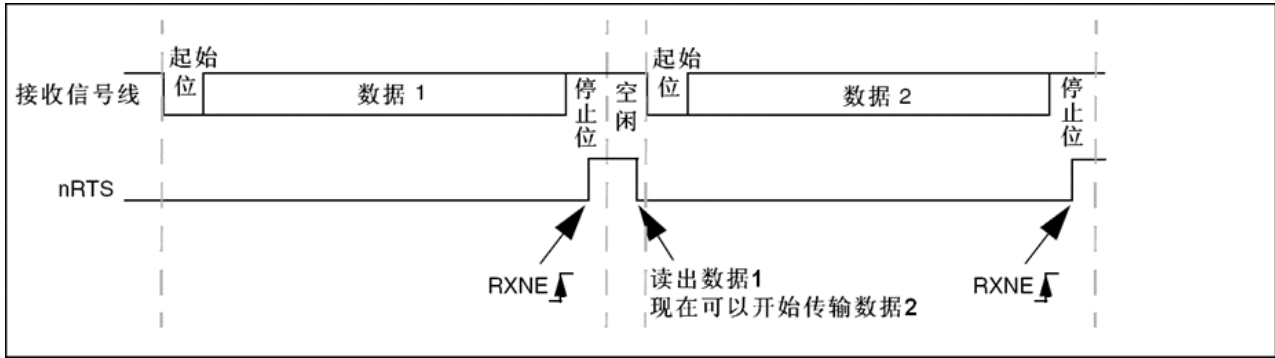


图132. RTS 流控制

19.3.14.2. CTS 流控制

如果 CTS 流控制被使能 (CTSE=1)，发送器在发送下一帧前检查 nCTS 输入。如果 nCTS 有效 (被拉成低电平)，则下一个数据被发送 (假设那个数据是准备发送的，也就是 TXE=0)，否则下一帧数据不被发出去。若 nCTS 在传输期间被变成无效，当前的传输完成后停止发送。

当 CTSE=1 时，只要 nCTS 输入一变换状态，硬件就自动设置 CTSIF 状态位。它表明接收器是否准备好进行通信。如果设置了 USART_CT3 寄存器的 CTSIE 位，则产生中断。下图是一个启用 CTS 流控制通信的例子。

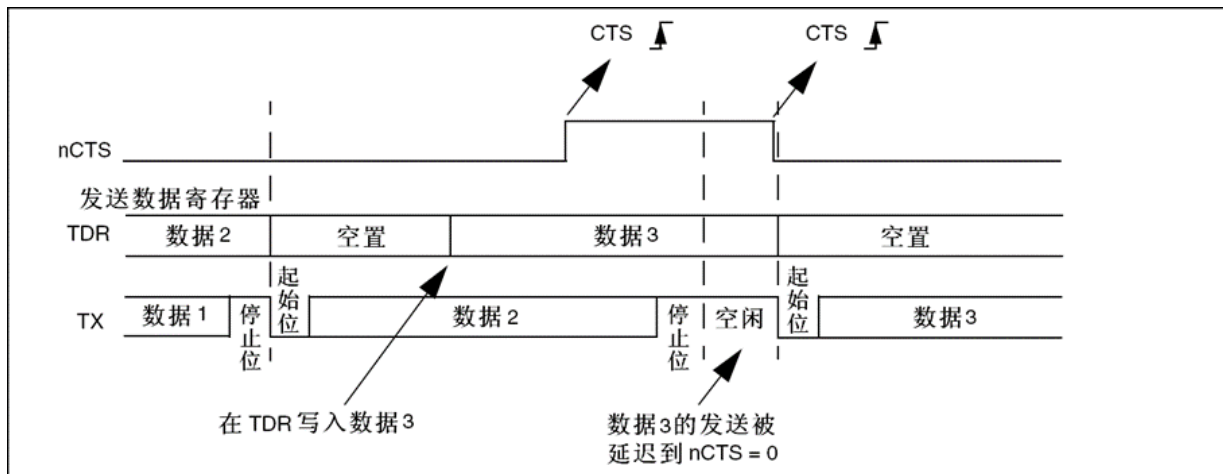


图133. CTS 流控制

19.4. USART 中断请求

表166. USART 中断请求

中断事件	事件标志	使能位
发送数据寄存器空	TXE	TXEIE
CTS 标志	CTS	CTSIE
发送完成	TC	TCIE
接收数据就绪可读	TXNE	TXNEIE
检测到数据溢出	ORE	
检测到空闲线路	IDLE	IDLEIE
奇偶检验错	PE	PEIE
断开标志	LBD	LBDIE
噪声标志，多缓冲通信中的溢出错误和帧错误	NE 或 ORT 或 FE	EIE ⁽¹⁾

1. 仅当使用 DMA 接收数据时，才使用这个标志位。

USART 的各种中断事件被连接到同一个中断向量，有以下各种中断事件：

- 发送期间：发送完成、清除发送、发送数据寄存器空。
 - 接收期间：空闲总线检测、溢出错误、接收数据寄存器非空、校验错误、LIN 断开符号检测、噪音标志（仅在多缓冲器通信）和帧错误（仅在多缓冲器通信）。
- 如果设置了对应的使能控制位，这些事件就可以产生各自的中断。

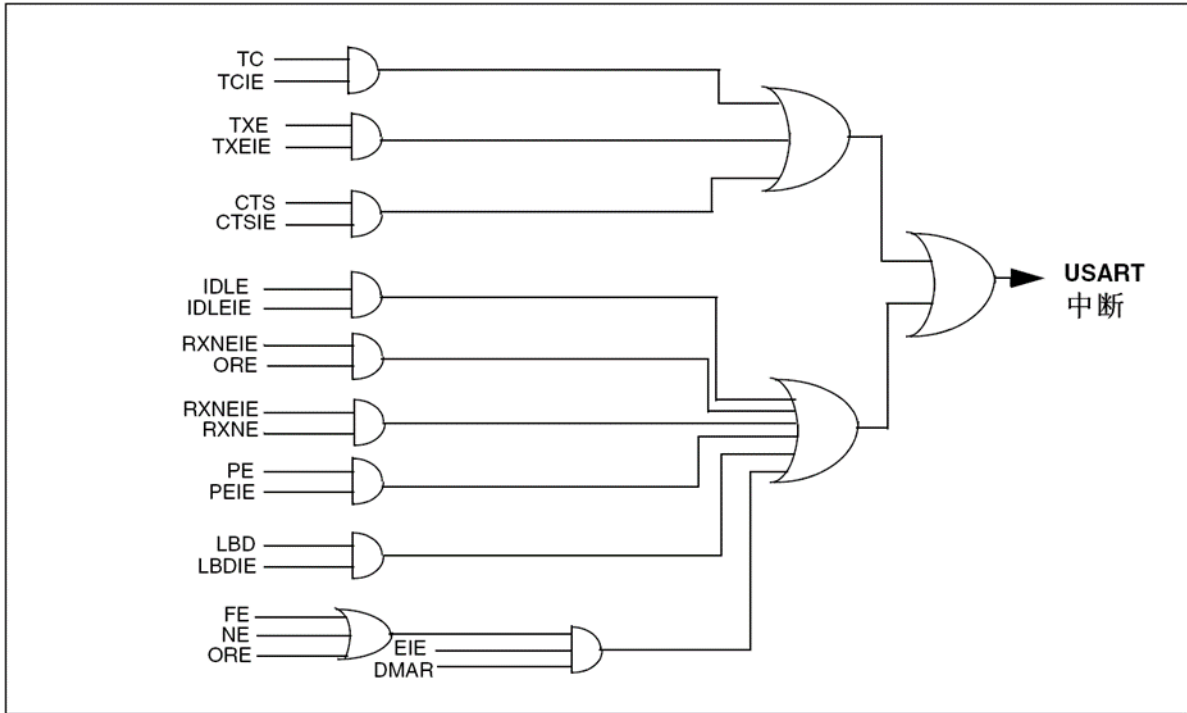


图134. USART 中断映像图

19.5. USART 模式配置

表167. USART 模式设置 ⁽¹⁾

USART 模式	USART1	USART2	USART3	UART4	UART5
异步模式	X	X	X	X	X
硬件流控制	X	X	X	NA	NA
多缓存通讯 (DMA)	X	X	X	X	X
多处理器通讯	X	X	X	X	X
同步	X	X	X	NA	NA
智能卡	X	X	X	NA	NA
半双工 (单线模式)	X	X	X	X	X
IrDA	X	X	X	X	X
LIN	X	X	X	X	X

1. X = 支持, NA = 不支持该应用

19.6. UART 寄存器描述

Register Name: USART_SR, Address: 0x40013800, Default: 0xc0

表168.

Bit Name	Position	Description	Default	Access
PE	[0]	校验错误 0: 没有奇偶校验错误 1: 奇偶校验错误	0x0	RW
FE	[1]	帧错误	0x0	RW

Bit Name	Position	Description	Default	Access
		0: 没有检测到帧错误 1: 检测到帧错误或者 break 符		
NE	[2]	噪声错误标志 0: 没有检测到噪声 1: 检测到噪声	0x0	RW
ORE	[3]	过载错误 0: 没有过载错误 1: 检测到过载错误	0x0	RW
IDLE	[4]	监测到总线空闲 0: 没有检测到空闲总线 1: 检测到空闲总线	0x0	RW
RXNE	[5]	读数据寄存器非空 0: 数据没有收到 1: 收到数据, 可以读出	0x0	RW
TC	[6]	发送完成 0: 发送还未完成 1: 发送完成	0x1	RC(readclear)
TXE	[7]	发送数据寄存器空 0: 数据还没有被转移到移位寄存器 1: 数据已经被转移到移位寄存器	0x1	R(ReadOnly)
LBD	[8]	LIN 断开检测标志 0: 没有检测到 LIN 断开 1: 检测到 LIN 断开	0x0	RW
CTS	[9]	CTS 标志 0: nCTS 状态线上没有变化 1: nCTS 状态线上没有变化	0x0	RW

Register Name: USART_DR, Address: 0x40013804, Default: 0x0

表169.

Bit Name	Position	Description	Default	Access
DR	[0:7]	数据值	0x0	RW

Register Name: USART_BRR, Address: 0x40013808, Default: 0x0

表170.

Bit Name	Position	Description	Default	Access
DIV_Fraction	[0:3]	USARTDIV 的小数部分	0x0	RW
DIV_Mantissa	[4:15]	USARTDIV 的整数部分	0x0	RW

Register Name: USART_CR1, Address: 0x4001380c, Default: 0x0

表171.

Bit Name	Position	Description	Default	Access
SBK	[0]	发送断开帧 0: 没有发送断开字符 1: 将要发送断开字符	0x0	RW
RWU	[1]	接收唤醒 0: 接收器处于正常工作模式	0x0	RW

Bit Name	Position	Description	Default	Access
		1: 接收器处于静默模式		
RE	[2]	接收使能 0: 禁止接收 1: 使能接收, 并开始搜寻 RX 引脚上的起始位	0x0	RW
TE	[3]	发送使能 0: 禁止发送 1: 使能发送	0x0	RW
IDLEIE	[4]	IDLE 中断使能 0: 禁止产生中断 1: 当 USART_SR 中的 IDLE 为'1'时, 产生 USART 中断	0x0	RW
RXNEIE	[5]	接收缓冲区非空中断使能 0: 禁止产生中断 1: 当 USART_SR 中的 ORE 或者 RXNE 为'1'时, 产生 USART 中断	0x0	RW
TCIE	[6]	发送完成中断使能 0: 禁止产生中断 1: 当 USART_SR 中的 TC 为'1'时, 产生 USART 中断	0x0	RW
TXEIE	[7]	发送缓冲区空中断使能 0: 禁止产生中断 1: 当 USART_SR 中的 TXE 为'1'时, 产生 USART 中断	0x0	RW
PEIE	[8]	PE 中断使能 0: 禁止产生中断 1: 当 USART_SR 中的 PE 为'1'时, 产生 USART 中断	0x0	RW
PS	[9]	校验选择 0: 偶校验 1: 奇校验	0x0	RW
PCE	[10]	检验控制使能 0: 禁止校验控制 1: 使能校验控制	0x0	RW
WAKE	[11]	唤醒的方法 0: 被空闲总线唤醒 1: 被地址标记唤醒	0x0	RW
M	[12]	字长 0: 一个起始位, 8 个数据位, n 个停止位 1: 一个起始位, 9 个数据位, n 个停止位	0x0	RW
UE	[13]	USART 使能 0: USART 分频器和输出被禁止 1: USART 模块使能	0x0	RW

Register Name: USART_CR2, Address: 0x40013810, Default: 0x0

表172.

Bit Name	Position	Description	Default	Access
ADD	[0:3]	本设备的 USART 节点地址	0x0	RW
LBDL	[5]	LIN 断开符检测长度 0: 10 位的断开符检测	0x0	RW

Bit Name	Position	Description	Default	Access
		1: 11 位的断开符检测		
LBDIE	[6]	LIN 断开符检测中断使能 0: 禁止中断 1: 只要 USART_SR 寄存器中的 LBD 为'1'就产生中断	0x0	RW
LBCL	[8]	最后一位时钟脉冲 0: 最后一位数据的时钟脉冲不从 CK 输出 1: 最后一位数据的时钟脉冲会从 CK 输出	0x0	RW
CPHA	[9]	时钟相位 0: 在时钟的第一个边沿进行数据捕获 1: 在时钟的第二个边沿进行数据捕获	0x0	RW
CPOL	[10]	时钟极性 0: 总线空闲时 CK 引脚上保持低电平 1: 总线空闲时 CK 引脚上保持高电平	0x0	RW
CLKEN	[11]	时钟使能 0: 禁止 CK 引脚 1: 使能 CK 引脚	0x0	RW
STOP	[12:13]	停止位 0: 1 个停止位 1: 0.5 个停止位 2: 2 个停止位 3: 1.5 个停止位	0x0	RW
LINEN	[14]	LIN 模式使能 0: 禁止 LIN 模式 1: 使能 LIN 模式	0x0	RW

Register Name: USART_CR3, Address: 0x40013814, Default: 0x0

表173.

Bit Name	Position	Description	Default	Access
EIE	[0]	错误中断使能 0: 禁止中断 1: 只要 USART_CR3 中的 DMAR=1, 并且 USART_SR 中的 FE=1, 或者 ORE=1, 或者 NE=1, 则产生中断	0x0	RW
IREN	[1]	红外模式使能 0: 不使能红外模式 1: 使能红外模式	0x0	RW
IRLP	[2]	红外低功耗 0: 通常模式 1: 低功耗模式	0x0	RW
HDSEL	[3]	半双工选择 0: 不选择半双工模式 1: 选择半双工模式	0x0	RW
NACK	[4]	智能卡 NACK 使能 0: 校验错误出现时, 不发送 NACK 1: 校验错误出现时, 发送 NACK	0x0	RW
SCEN	[5]	智能卡模式使能	0x0	RW

Bit Name	Position	Description	Default	Access
		0: 禁止智能卡模式 1: 使能智能卡模式		
DMAR	[6]	DMA 使能接收 0: 禁止接收时的 DMA 模式 1: 使能接收时的 DMA 模式	0x0	RW
DMAT	[7]	DMA 使能发送 0: 禁止发送时的 DMA 模式 1: 使能发送时的 DMA 模式	0x0	RW
RTSE	[8]	RTS 使能 0: 禁止 RTS 硬件流控制 1: RTS 中断使能, 只有接收缓冲区内有空余的空间时才请求下一个数据。当前数据发送完成后, 发送操作就需要暂停下来。如果可以接收数据了, 将 nRTS 输出置为有效(拉至低电平)。	0x0	RW
CTSE	[9]	CTS 使能 0: 禁止 CTS 硬件流控制 1: CTS 模式使能, 只有 nCTS 输入信号有效(拉成低电平)时才能发送数据。如果在数据传输的过程中, nCTS 信号变成无效, 那么发完这个数据后, 传输就停止下来。如果当 nCTS 为无效时, 往数据寄存器里写数据, 则要等到 nCTS 有效时才会发送这个数据。	0x0	RW
CTSIE	[10]	CTS 中断使能 0: 禁止中断 1: USART_SR 寄存器中的 CTS 为'1'时产生中断	0x0	RW

Register Name: USART_GTPR, Address: 0x40013818, Default: 0x0

表174.

Bit Name	Position	Description	Default	Access
PSC	[0:7]	预分频器值	0x0	RW
GT	[8:15]	保护时间值	0x0	RW

20. 内部集成电路通信接口 (I²C)

20.1. I²C 简介

I²C (内部集成电路总线) 模块提供了符合工业标准的两线串行制接口, 可用于 MCU 和外部 I²C 设备的通讯。I²C 总线使用两条串行线: 串行数据线 SDA 和串行时钟线 SCL。

I²C 接口模块实现了 I²C 协议的标速模式, 快速模式以及快速+模式, 具备 CRC 计算和校验功能。此外还支持多主机 I²C 总线架构。I²C 接口模块也支持 DMA 模式, 可有效减轻 CPU 的负担。

20.2. I²C 主要特点

- 并行总线/I²C 总线协议转换器
 - 多主机功能: 该模块既可做主设备也可做从设备
 - I²C 主设备功能
 - 产生时钟
 - 产生起始和停止信号
 - I²C 从设备功能
 - 可编程的 I²C 地址检测
 - 可响应 2 个从地址的双地址能力
 - 停止位检测
 - 产生和检测 7 位/10 位地址和广播呼叫
 - 支持标速 (最高 100 kHz), 快速 (最高 400 kHz) 和快速+模式 (最高 1MHz);
 - 状态标志:
 - 发送器/接收器模式标志
 - 字节发送结束标志
 - I²C 总线忙标志
 - 错误标志
 - 主模式时的仲裁丢失
 - 地址/数据传输后的应答 (ACK) 错误
 - 检测到错位的起始或停止条件
 - 禁止拉长时钟功能时的上溢或下溢
 - 2 个中断向量
 - 1 个中断用于地址/数据通讯成功
 - 1 个中断用于错误
 - 可选的拉长时钟功能
 - 具单字节缓冲器的 DMA
 - 可配置的 PEC (信息包错误检测) 的产生或校验:
 - 发送模式中 PEC 值可以作为最后一个字节传输
 - 用于最后一个接收字节的 PEC 错误校验
- 可配置使能的深度为 2 个 byte 的 I²C fifo 模式, 发送和接收数据都支持。

20.3. I²C 功能描述

I²C 模块接收和发送数据, 并将数据从串行转换成并行, 或并行转换成串行。可以开启或禁止中断。接口通过数据引脚 (SDA) 和时钟引脚 (SCL) 连接到 I²C 总线。允许连接到标准 (高达 100kHz) 或快速 (高达 400kHz) 的 I²C 总线。

20.3.1. 模式选择

接口可以下述 4 种模式中的一种运行：

- 从发送器模式
- 从接收器模式
- 主发送器模式
- 主接收器模式

该模块默认地工作于从模式。接口在生成起始条件后自动地从从模式切换到主模式；当仲裁丢失或产生停止信号时，则从主模式切换到从模式。允许多主机功能。

20.3.1.1. 通信流

主模式时，I²C 接口启动数据传输并产生时钟信号。串行数据传输总是以起始条件开始并以停止条件结束。起始条件和停止条件都是在主模式下由软件控制产生。

从模式时，I²C 接口能识别它自己的地址（7 位或 10 位）和广播呼叫地址。软件能够控制开启或禁止广播呼叫地址的识别。

数据和地址按 8 位/字节进行传输，高位在前。跟在起始条件后的 1 或 2 个字节是地址（7 位模式为 1 个字节，10 位模式为 2 个字节）。地址只在主模式发送。

在一个字节传输的 8 个时钟后的第 9 个时钟期间，接收器必须回送一个应答位（ACK）给发送器。参考下图。

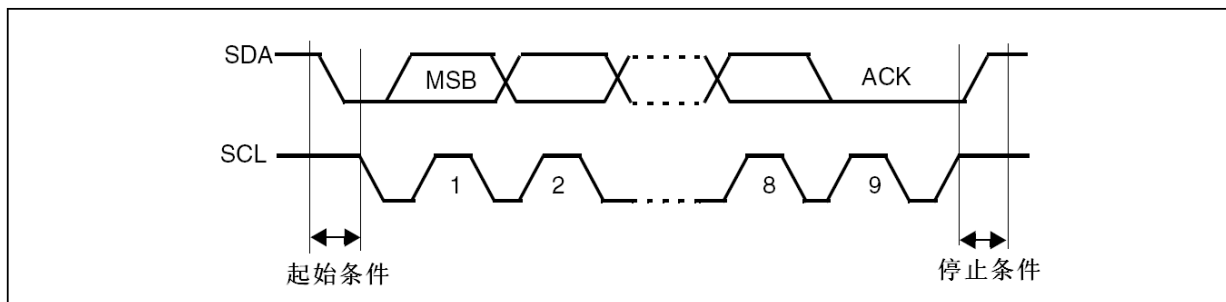


图135. I²C 总线协议

软件可以开启或禁止应答（ACK），并可以设置 I²C 接口的地址（7 位、10 位地址或广播呼叫地址）。

I²C 接口的功能框图示于下图。

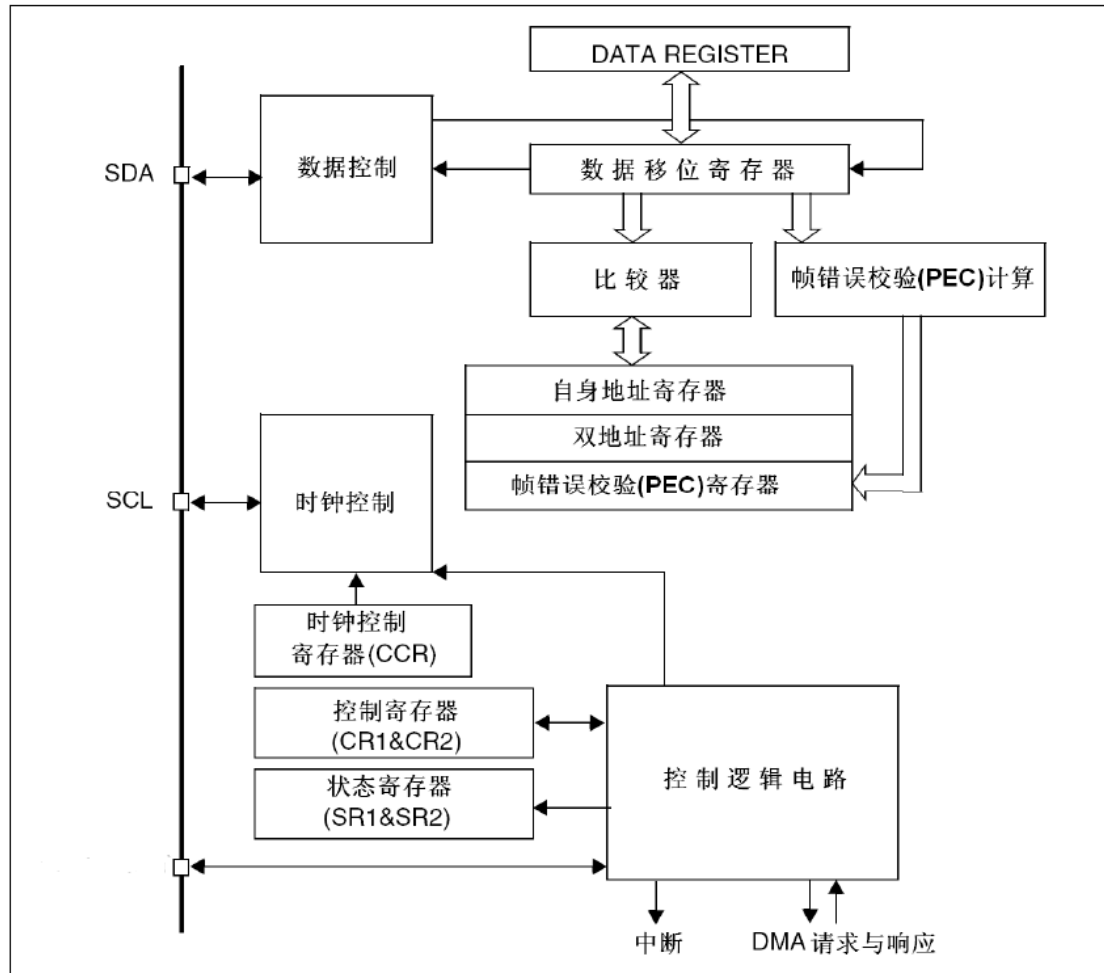


图136. I²C 的功能框图

20.3.2. I²C 从模式

默认情况下，I²C 接口总是工作在从模式。从从模式切换到主模式，需要产生一个起始条件。

为了产生正确的时序，必须在 I2C_CR2 寄存器中设定该模块的输入时钟。输入时钟的频率必须至少是：

- 标准模式下为：2MHz
- 快速模式下为：4MHz

一旦检测到起始条件，在 SDA 线上接收到的地址被送到移位寄存器。然后与芯片自己的地址 OAR1 和 OAR2（当 ENDUAL=1）或者广播呼叫地址（如果 ENGCG=1）相比较。

注：在 10 位地址模式时，比较包括头段序列（11110xx0），其中的 xx 是地址的两个最高有效位。

头段或地址不匹配：I2C 接口将其忽略并等待另一个起始条件。

头段匹配（仅 10 位模式）：如果 ACK 位被置'1'，I²C 接口产生一个应答脉冲并等待 8 位从地址。

地址匹配：I²C 接口产生以下时序：

- 如果 ACK 被置'1'，则产生一个应答脉冲
- 硬件设置 ADDR 位；如果设置了 ITEVFEN 位，则产生一个中断
- 如果 ENDUAL=1，软件必须读 DUALF 位，以确认响应了哪个从地址。

在 10 位模式，接收到地址序列后，从设备总是处于接收器模式。在收到与地址匹配的头序列并且最低位

为'1'（即 11110xx1）后，当接收到重复的起始条件时，将进入发送器模式。

在从模式下 TRA 位指示当前是处于接收器模式还是发送器模式。

20.3.2.1. 从发送器

在接收到地址和清除 ADDR 位后，从发送器将字节从 DR 寄存器经由内部移位寄存器发送到 SDA 线上。

从设备保持 SCL 为低电平，直到 ADDR 位被清除并且待发送数据已写入 DR 寄存器。（见下图中的 EV1 和 EV3）。

当收到应答脉冲时：

- TxE 位被硬件置位，如果设置了 ITEVFEN 和 ITBUFEN 位，则产生一个中断。

如果 TxE 位被置位，但在下一个数据发送结束之前没有新数据写入到 I²C_DR 寄存器，则 BTF 位被置位，在清除 BTF 之前 I²C 接口将保持 SCL 为低电平；读出 I2C_SR1 之后再写入 I²C_DR 寄存器将清除 BTF 位。

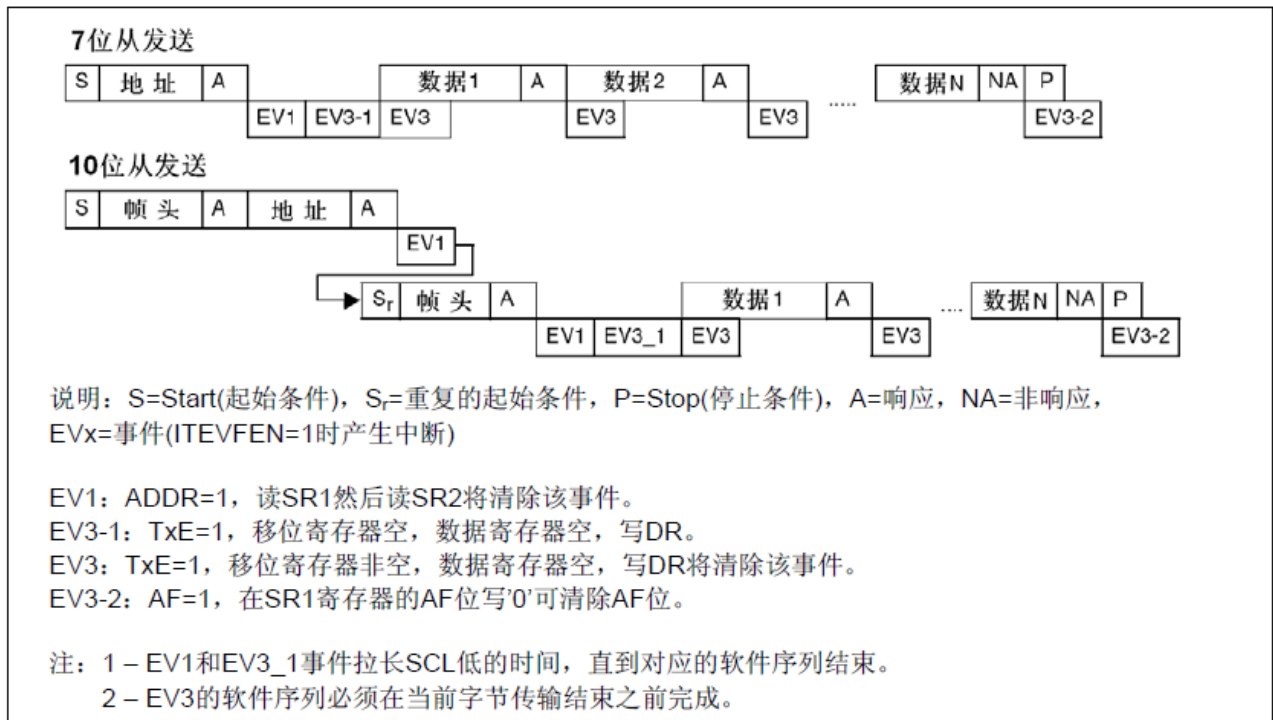


图137. 从发送器的传送序列

20.3.2.2. 从接收器

在接收到地址并清除 ADDR 后，从接收器将通过内部移位寄存器从 SDA 线接收到的字节存进 DR 寄存器。

I²C 接口在接收到每个字节后都执行下列操作：

- 如果设置了 ACK 位，则产生一个应答脉冲
- 硬件设置 RxNE=1。如果设置了 ITEVFEN 和 ITBUFEN 位，则产生一个中断。

如果 RxNE 被置位，并且在接收新的数据结束之前 DR 寄存器未被读出，BTF 位被置位，在清除 BTF 之前 I²C 接口将保持 SCL 为低电平；读出 I²C_SR1 之后再写入 I²C_DR 寄存器将清除 BTF 位。（见下图）。

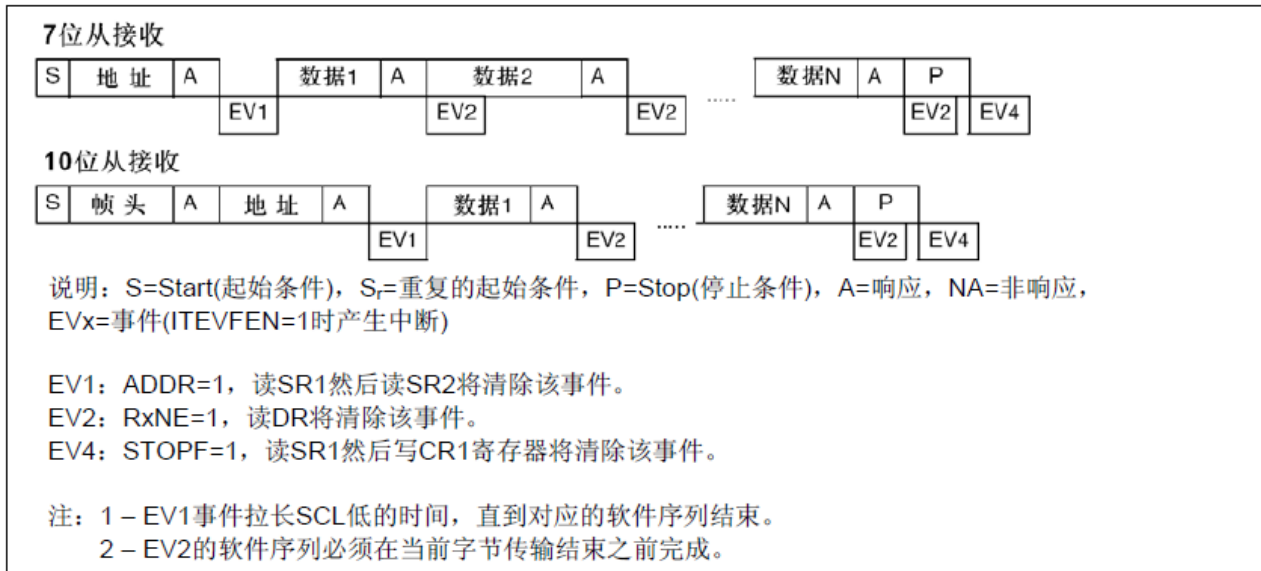


图138. 从接收器的传送序列

20.3.2.3. 关闭从通信

在传输完最后一个数据字节后, 主设备产生一个停止条件, I²C 接口检测到这一条件时:

- 设置 STOPF=1, 如果设置了 ITEVFEN 位, 则产生一个中断。

然后 I²C 接口等待读 SR1 寄存器, 再写 CR1 寄存器。(见上图的 EV4)。

20.3.3. I²C 主模式

在主模式时, I²C 接口启动数据传输并产生时钟信号。串行数据传输总是以起始条件开始并以停止条件结束。当通过 START 位在总线上产生了起始条件, 设备就进入了主模式。

以下是主模式所要求的操作顺序:

- 在 I²C_CR2 寄存器中设定该模块的输入时钟以产生正确的时序
- 配置时钟控制寄存器
- 配置上升时间寄存器
- 编程 I²C_CR1 寄存器启动外设
- 置 I²C_CR1 寄存器中的 START 位为 1, 产生起始条件

I²C 模块的输入时钟频率必须至少是:

- 标准模式下为: 2MHz
- 快速模式下为: 4MHz

20.3.3.1. 起始条件

当 BUSY=0 时, 设置 START=1, I²C 接口将产生一个开始条件并切换至主模式 (M/SL 位置位)。

注: 在主模式下, 设置 START 位将在当前字节传输完后由硬件产生一个重新开始条件。

一旦发出开始条件:

- SB 位被硬件置位, 如果设置了 ITEVFEN 位, 则会产生一个中断。

然后主设备等待读 SR1 寄存器, 紧跟着将从地址写入 DR 寄存器。

20.3.3.2. 从地址的发送

从地址通过内部移位寄存器被送到 SDA 线上。

- 在 10 位地址模式时，发送一个头段序列产生以下事件：
 - ADDR 位被硬件置位，如果设置了 ITEVFEN 位，则产生一个中断。

然后主设备等待读 SR1 寄存器，再将第二个地址字节写入 DR 寄存器。

- ADDR 位被硬件置位，如果设置了 ITEVFEN 位，则产生一个中断。

随后主设备等待一次读 SR1 寄存器，跟着读 SR2 寄存器。

- 在 7 位地址模式时，只需送出一个地址字节。

一旦该地址字节被送出，

- ADDR 位被硬件置位，如果设置了 ITEVFEN 位，则产生一个中断。

随后主设备等待一次读 SR1 寄存器，跟着读 SR2 寄存器。

根据送出从地址的最低位，主设备决定进入发送器模式还是进入接收器模式。

- 在 7 位地址模式时，
 - 要进入发送器模式，主设备发送从地址时置最低位为'0'。
 - 要进入接收器模式，主设备发送从地址时置最低位为'1'。
- 在 10 位地址模式时
 - 要进入发送器模式，主设备先送头字节 (11110xx0)，然后送最低位为'0'的从地址。(这里 xx 代表 10 位地址中的最高 2 位。)
 - 要进入接收器模式，主设备先送头字节 (11110xx0)，然后送最低位为'1'的从地址。然后再重新发送一个开始条件，后面跟着头字节 (11110xx1) (这里 xx 代表 10 位地址中的最高 2 位。)

TRA 位指示主设备是在接收器模式还是发送器模式。

20.3.3.3. 主发送器

在发送了地址和清除了 ADDR 位后，主设备通过内部移位寄存器将字节从 DR 寄存器发送到 SDA 线上。

主设备等待，直到 TxE 被清除。

当收到应答脉冲时：

- TxE 位被硬件置位，如果设置了 INEVFEN 和 ITBUFEN 位，则产生一个中断。

如果 TxE 被置位并且在上一次数据发送结束之前没有写新的数据字节到 DR 寄存器，则 BTF 被硬件置位，在清除 BTF 之前 I²C 接口将保持 SCL 为低电平；读出 I²C_SR1 之后再写入 I²C_DR 寄存器将清除 BTF 位。

20.3.3.4. 关闭通信

在 DR 寄存器中写入最后一个字节后，通过设置 STOP 位产生一个停止条件，然后 I²C 接口将自动回到从模式 (M/S 位清除)。

注：当 TxE 或 BTF 位置位时，停止条件应安排在出现 EV8_2 事件时。

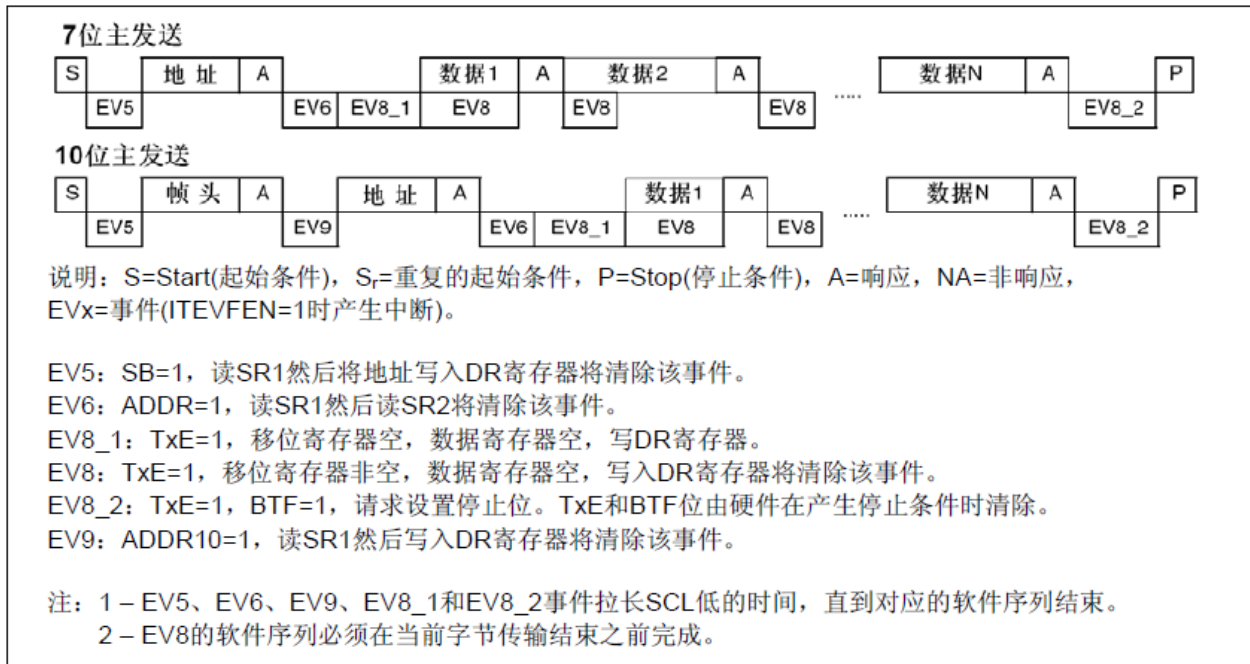


图139. 主发送器传送序列

20.3.3.5. 主接收器

在发送地址和清除 ADDR 之后, I²C 接口进入主接收器模式。在此模式下, I²C 接口从 SDA 线接收数据字节, 并通过内部移位寄存器送至 DR 寄存器。在每个字节后, I²C 接口依次执行以下操作:

- 如果 ACK 位被置位, 发出一个应答脉冲。
- 硬件设置 RxNE=1, 如果设置了 INEVFEN 和 ITBUFEN 位, 则会产生一个中断。

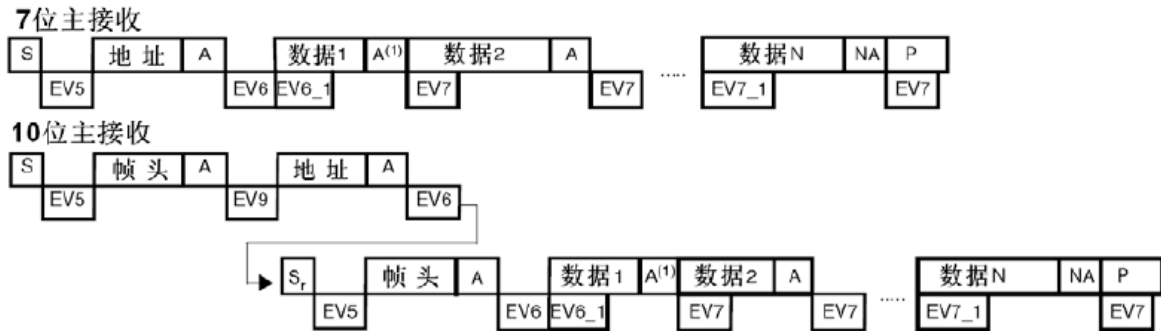
如果 RxNE 位被置位, 并且在接收新数据结束前, DR 寄存器中的数据没有被读走, 硬件将设置 BTF=1, 在清除 BTF 之前 I²C 接口将保持 SCL 为低电平; 读出 I²C_SR1 之后再读出 I²C_DR 寄存器将清除 BTF 位。

20.3.3.6. 关闭通信

主设备在从设备接收到最后一个字节后发送一个 NACK。接收到 NACK 后, 从设备释放对 SCL 和 SDA 线的控制; 主设备就可以发送一个停止/重起始条件。

- 为了在收到最后一个字节后产生一个 NACK 脉冲, 在读倒数第二个数据字节之后 (在倒数第二个 RxNE 事件之后) 必须清除 ACK 位。
- 为了产生一个停止/重起始条件, 软件必须在读倒数第二个数据字节之后 (在倒数第二个 RxNE 事件之后) 设置 STOP/START 位。
- 只接收一个字节时, 刚好在 EV6 之后 (EV6_1 时, 清除 ADDR 之后) 要关闭应答和停止条件的产生位。

在产生了停止条件后, I²C 接口自动回到从模式 (M/SL 位被清除)。



说明：S=Start(起始条件)，S_r=重复的起始条件，P=Stop(停止条件)，A=响应，NA=非响应，EVx=事件(ITEVFEN=1时产生中断)

EV5：SB=1，读SR1然后将地址写入DR寄存器将清除该事件。

EV6：ADDR=1，读SR1然后读SR2将清除该事件。在10位主接收模式下，该事件后应设置CR2的START=1。

EV6_1：没有对应的事件标志，只适于接收1个字节的情况。恰好在EV6之后(即清除了ADDR之后)，要清除响应和停止条件的产生位。

EV7：RxNE=1，读DR寄存器清除该事件。

EV7_1：RxNE=1，读DR寄存器清除该事件。设置ACK=0和STOP请求。

EV9：ADDR10=1，读SR1然后写入DR寄存器将清除该事件。

图140. 主接收器传送序列

- 1) 如果收到一个单独的字节，则是 NA。
- 2) EV5、EV6 和 EV9 事件拉长 SCL 低电平，直到对应的软件序列结束。
- 3) EV7 的软件序列必须在当前字节传输结束前完成。
- 4) EV6_1 或 EV7_1 的软件序列 必须在当前传输字节的 ACK 脉冲之前完成。

20.3.4. 错误条件

以下条件可能造成通讯失败。

20.3.4.1. 总线错误 (BERR)

在一个地址或数据字节传输期间，当 I2C 接口检测到一个外部的停止或起始条件则产生总线错误。此时：

- BERR 位被置位为'1'；如果设置了 ITERREN 位，则产生一个中断；
- 在从模式情况下，数据被丢弃，硬件释放总线：
 - 如果是错误的开始条件，从设备认为是一个重启动，并等待地址或停止条件。
 - 如果是错误的停止条件，从设备按正常的停止条件操作，同时硬件释放总线。
- 在主模式情况下，硬件不释放总线，同时不影响当前的传输状态。此时由软件决定是否要中止当前的传输。

20.3.4.2. 应答错误 (AF)

当接口检测到一个无应答位时，产生应答错误。此时：

- AF 位被置位，如果设置了 ITERREN 位，则产生一个中断；
- 当发送器接收到一个 NACK 时，必须复位通讯：
 - 如果是处于从模式，硬件释放总线。
 - 如果是处于主模式，软件必须生成一个停止条件。

20.3.4.3. 仲裁丢失 (ARLO)

当 I²C 接口检测到仲裁丢失时产生仲裁丢失错误，此时：

- ARLO 位被硬件置位，如果设置了 ITERREN 位，则产生一个中断；
- I²C 接口自动回到从模式（M/SL 位被清除）。当 I²C 接口丢失了仲裁，则它无法在同一个传输中响应它的从地址，但它可以在赢得总线的主设备发送重起始条件之后响应；
- 硬件释放总线。

20.3.4.4. 过载/欠载错误 (OVR)

在从模式下，如果禁止时钟延长，I²C 接口正在接收数据时，当它已经接收到一个字节（RxNE=1），但在 DR 寄存器中前一个字节数据还没有被读出，则发生过载错误。此时：

- 最后接收的数据被丢弃；
- 在过载错误时，软件应清除 RxNE 位，发送器应该重新发送最后一次发送的字节。

在从模式下，如果禁止时钟延长，I²C 接口正在发送数据时，在下一个字节的时钟到达之前，新的数据还未写入 DR 寄存器（TxNE=1），则发生欠载错误。此时：

- 在 DR 寄存器中的前一个字节将被重复发出；
- 用户应该确定在发生欠载错时，接收端应丢弃重复接收到的数据。发送端应按 I2C 总线标准在规定的更新时间更新 DR 寄存器。

在发送第一个字节时，必须在清除 ADDR 之后并且第一个 SCL 上升沿之前写入 DR 寄存器；如果不能做到这点，则接收方应该丢弃第一个数据。

20.3.5. SDA/SCL 线控制

- 如果允许时钟延长：
 - 发送器模式：如果 TxNE=1 且 BTF=1：I2C 接口在传输前保持时钟线为低，以等待软件读取 SR1，然后把数据写进数据寄存器（缓冲器和移位寄存器都是空的）。
 - 接收器模式：如果 RxNE=1 且 BTF=1：I2C 接口在接收到数据字节后保持时钟线为低，以等待软件读 SR1，然后读数据寄存器 DR（缓冲器和移位寄存器都是满的）。
- 如果在从模式中禁止时钟延长：
 - 如果 RxNE=1，在接收到下个字节前 DR 还没有被读出，则发生过载错。接收到的最后一个字节丢失。
 - 如果 TxNE=1，在必须发送下个字节之前却没有新数据写进 DR，则发生欠载错。相同的字节将被重复发出。
- 不控制重复写冲突。

20.3.6. DMA 请求

DMA 请求（当被使能时）仅用于数据传输。发送时数据寄存器变空或接收时数据寄存器变满，则产生 DMA 请求。DMA 请求必须在当前字节传输结束之前被响应。当为相应 DMA 通道设置的数据传输量已经完成时，DMA 控制器发送传输结束信号 EOT 到 I²C 接口，并且在中断允许时产生一个传输完成中断：

- 主发送器：在 EOT 中断服务程序中，需禁止 DMA 请求，然后在等到 BTF 事件后设置停止条件。
- 主接收器：当要接收的数据数目大于或等于 2 时，DMA 控制器发送一个硬件信号 EOT_1，它对应 DMA 传输（字节数 - 1）。如果在 I²C_CR2 寄存器中设置了 LAST 位，硬件在发送完 EOT_1 后的下一个字节，将自动发送 NACK。在中断允许的情况下，用户可以在 DMA 传输完成的中断服务程序中产生一个停止条件。

20.3.6.1. 利用 DMA 发送

通过设置 I²C_CR2 寄存器中的 DMAEN 位可以激活 DMA 模式。只要 TxNE 位被置位，数据将由 DMA 从预置的存储区装载进 I²C_DR 寄存器。为 I²C 分配一个 DMA 通道，须执行以下步骤（x 是通道号）：

- 1) 在 DMA_CPARx 寄存器中设置 I²C_DR 寄存器地址。数据将在每个 TxNE 事件后从存储器传送到这个地址。
- 2) 在 DMA_CMARx 寄存器中设置存储器地址。数据在每个 TxNE 事件后从这个存储区传送到 I²C_DR。

- 3) 在 DMA_CNDTRx 寄存器中设置所需的传输字节数。在每个 TxE 事件后, 此值将被递减。
- 4) 利用 DMA_CCRx 寄存器中的 PL[0:1]位配置通道优先级。
- 5) 设置 DMA_CCRx 寄存器中的 DIR 位, 并根据应用要求可以配置在整个传输完成一半或全部完成时发出中断请求。
- 6) 通过设置 DMA_CCTx 寄存器上的 EN 位激活通道。

当 DMA 控制器中设置的数据传输数目已经完成时, DMA 控制器给 I²C 接口发送一个传输结束的 EOT/ EOT_1 信号。在中断允许的情况下, 将产生一个 DMA 中断。

注: 如果使用 DMA 进行发送时, 不要设置 I²C_CR2 寄存器的 ITBUFEN 位。

20.3.6.2. 利用 DMA 接收

通过设置 I²C_CR2 寄存器中的 DMAEN 位可以激活 DMA 接收模式。每次接收到数据字节时, 将由 DMA 把 I²C_DR 寄存器的数据传送到设置的存储区 (参考 DMA 说明)。设置 DMA 通道进行 I²C 接收, 须执行以下步骤 (x 是通道号):

- 1) 在 DMA_CPARx 寄存器中设置 I²C_DR 寄存器的地址。数据将在每次 RxNE 事件后从此地址传送到存储区。
- 2) 在 DMA_CMARx 寄存器中设置存储区地址。数据将在每次 RxNE 事件后从 I²C_DR 寄存器传送到此存储区。
- 3) 在 DMA_CNDTRx 寄存器中设置所需的传输字节数。在每个 RxNE 事件后, 此值将被递减。
- 4) 用 DMA_CCRx 寄存器中的 PL[0:1]配置通道优先级。
- 5) 清除 DMA_CCRx 寄存器中的 DIR 位, 根据应用要求可以设置在数据传输完成一半或全部完成时发出中断请求。
- 6) 设置 DMA_CCRx 寄存器中的 EN 位激活该通道。

当 DMA 控制器中设置的数据传输数目已经完成时, DMA 控制器给 I²C 接口发送一个传输结束的 EOT/ EOT_1 信号。在中断允许的情况下, 将产生一个 DMA 中断。

注: 如果使用 DMA 进行接收时, 不要设置 I²C_CR2 寄存器的 ITBUFEN 位。

20.3.7. 包错误校验 (PEC)

包错误校验 (PEC) 计算器是用于提高通信的可靠性, 这个计算器使用下述 CRC-8 多项式对每一位串行数据进行计算: $C(x) = x^8 + x^2 + x + 1$

- PEC 计算由 I²C_CR1 寄存器的 ENPEC 位激活。PEC 使用 CRC-8 算法对所有信息字节进行计算, 包括地址和读/写位在内。
 - 在发送时: 在最后一个 TxE 事件时设置 I²C_CR1 寄存器的 PEC 传输位, PEC 将在最后一个字节后被发送。
 - 在接收时: 在最后一个 RxNE 事件之后设置 I²C_CR1 寄存器的 PEC 位, 如果下个接收到的字节不等于内部计算的 PEC, 接收器发送一个 NACK。如果是主接收器, 不管校对的结果如何, PEC 后都将发送 NACK。PEC 位必须在接收当前字节的 ACK 脉冲之前设置。
 - 在 I²C_SR1 寄存器中可获得 PECERR 错误标记/中断。
 - 如果 DMA 和 PEC 计算器都被激活:
 - 在发送时: 当 I2C 接口从 DMA 控制器处接收到 EOT 信号时, 它在最后一个字节后自动发送 PEC。
 - 在接收时: 当 I2C 接口从 DMA 处接收到一个 EOT_1 信号时, 它将自动把下一个字节作为 PEC, 并且将检查它。在接收到 PEC 后产生一个 DMA 请求。

- 为了允许中间 PEC 传输，在 I²C_CR2 寄存器中有一个控制位（LAST 位）用于判别是否真是最后一个 DMA 传输。如果确实是最后一个主接收器的 DMA 请求，在接收到最后一个字节后自动发送 NACK。
- 仲裁丢失时 PEC 计算失效。

20.4. I²C 中断请求

下表列出了所有的 I²C 中断请求。

表175. I²C 中断请求表

中断事件	事件标志	开启控制位
起始位已发送（主）	SB	ITEVFEN
地址已发送（主）或地址匹配（从）	ADDR	
10 位头段已发送（主）	ADD10	
已收到停止（从）	STOPF	
数据字节传输完成	BTF	
FIFO 模式下数据发送移位器空	TxSRE	
接收缓冲区非空（接收 FIFO 非空）	RxNE	ITEVFEN 和 ITBUFEN
发送缓冲区空（发送 FIFO 空）	TxE	
总线错误	BERR	ITERREN
仲裁丢失（主）	ARLO	
响应失败	AF	
过载/欠载	OVR	
PEC 错误	PECERR	

注：

1. SB、ADDR、ADD10、STOPF、BTF、RxNE 和 TxE 通过逻辑或汇到同一个中断通道中。
2. BERR、ARLO、AF、OVR、PECERR 通过逻辑或汇到同一个中断通道中。

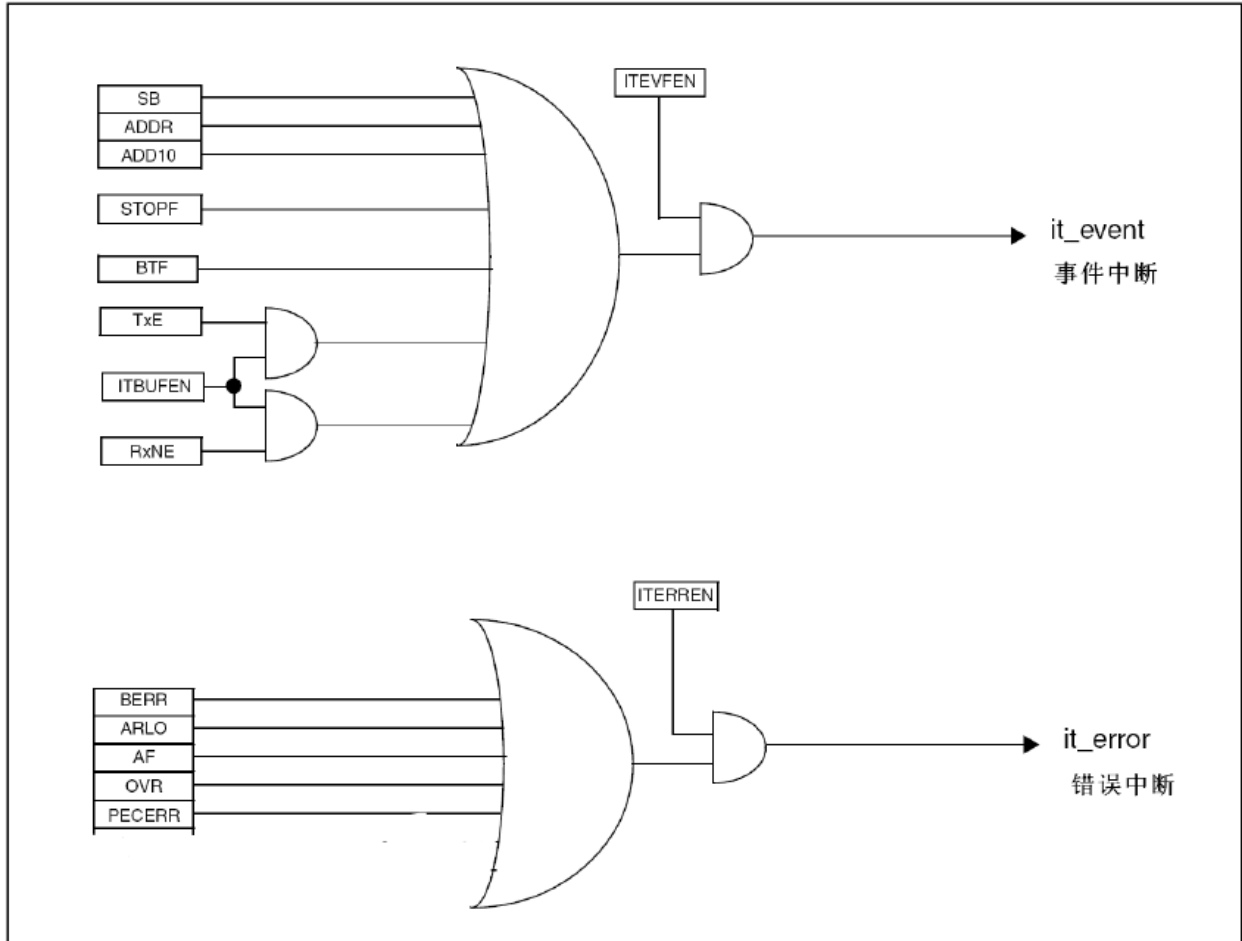


图141. I²C 中断映射图

20.5. I²C 寄存器描述

Register Name: CR1, Address: 0x40005400, Default: 0x0

表176. 控制寄存器 1

Bit Name	Position	Description	Default	Access
PE	[0]	PE: I2C 模块使能 (Peripheral enable) 注: 如果清除该位时通讯正在进行, 在当前通讯结束后, I2C 模块被禁用并返回空闲状态。由于在通讯结束后发生 PE = 0, 所有的位被清除。 在主模式下, 通讯结束之前, 绝不能清除该位。 0: 禁用 I2C 模块 1: 启用 I2C 模块: 根据 SMBus 位的设置, 相应的 I/O 口需配置为复用功能	0x0	RW
ENPEC	[5]	ENPEC: PEC 使能 (PEC enable) 0: 禁止 PEC 计算; 1: 开启 PEC 计算	0x0	RW
ENGCG	[6]	ENGCG: 广播呼叫使能 (General call enable) 0: 禁止广播呼叫。以非应答响应地址 00h 1: 允许广播呼叫。以应答响应地址 00h。	0x0	RW
NOSTRETCH	[7]	NOSTRETCH: 禁止时钟延长(从模式) (Clock stretching disable (Slave mode)) 该位用于当 ADDR 或 BTF 标志被置位, 在从模式下禁止时钟延长, 直到它被软件复位。	0x0	RW

Bit Name	Position	Description	Default	Access
		0: 允许时钟延长 1: 禁止时钟延长。		
START	[8]	START: 起始条件产生 (Start generation) 软件可以设置或清除该位, 或当起始条件发出后或 PE=0 时, 由硬件清除。 在主模式下: 0: 主模式下: 无起始条件产生; 在从模式下: 无起始条件产生; 1: 主模式下: 重复产生起始条件; 在从模式下: 当总线空闲时, 产生起始条件	0x0	RW
STOP	[9]	STOP: 停止条件产生 (Stop generation) 软件可以设置或清除该位; 或当检测到停止条件时, 由硬件清除; 当检测到超时错误时, 硬件将其置位。 注: 当设置了 STOP、START 或 PEC 位, 在硬件清除这个位之前, 软件不要执行任何对 I2C_CR1 的写操作; 否则有可能会第 2 次设置 STOP、START 或 PEC 位。 0: 主模式下: 无停止条件产生; 无停止条件产生; 1: 主模式下: 在当前字节传输或当前起始条件发出后产生停止条件。 在从模式下: 在当前字节传输或释放 SCL 和 SDA 线。	0x0	RW
ACK	[10]	ACK: 应答使能 (Acknowledge enable) 软件可以设置或清除该位, 或当 PE=0 时, 由硬件清除。 0: 无应答返回; 1: 在接收到一个字节后返回一个应答(匹配的地址或数据)。	0x0	RW
POS	[11]	POS: 应答/PEC 位置(用于数据接收) (Acknowledge/PEC Position (for data reception)) 软件可以设置或清除该位, 或当 PE=0 时, 由硬件清除。 注: POS 位只能用在 2 字节的接收配置中, 必须在接收数据之前配置。为了 NACK 第 2 个字节, 必须在清除 ADDR 为之后清除 ACK 位。 为了检测第 2 个字节的 PEC, 必须在配置了 POS 位之后, 拉伸 ADDR 事件时设置 PEC 位。 0: ACK 位控制当前移位寄存器内正在接收的字节的(N)ACK。PEC 位表明当前移位寄存器内的字节是 PEC; 1: ACK 位控制在移位寄存器里接收的下一个字节的(N)ACK。PEC 位表明在移位寄存器里接收的下一个字节是 PEC。	0x0	RW
PEC	[12]	PEC: 数据包出错检测 (Packet error checking) 软件可以设置或清除该位; 当传送 PEC 后, 或起始或停止条件时, 或当 PE=0 时硬件将其清除。 0: 无 PEC 传输; 1: PEC 传输(在发送或接收模式)。注: 仲裁丢失时, PEC 的计算失效。	0x0	RW
SWRST	[15]	SWRST: 软件复位 (Software reset) 当被置位时, I2C 处于复位状态。在复位该位前确信 I2C 的引脚被释放, 总线是空的。 注: 该位可以用于 BUSY 位为'1', 在总线上又没有检测到停止条件时。 0: I2C 模块不处于复位状态;	0x0	RW

Bit Name	Position	Description	Default	Access
		1: I2C 模块处于复位状态。		

Register Name: CR2, Address: 0x40005404, Default: 0x0

表177. 控制寄存器 2

Bit Name	Position	Description	Default	Access
FREQ	[0:6]	FREQ[6:0]: I2C 模块时钟频率 (Peripheral clock frequency) FREQ[6:0]是对应输入 APB 时钟的频率, 必须设置正确的输入时钟频率以产生正确的时序, 允许的范围在 2~72MHz 之间: 0000000: 禁用 0000001: 禁用 0000010: 2MHz ... 0100100: 36MHz 1001000: 72MHz 大于 1001000: 禁用。 注意: 在标准 I2C 模式时, APB 的时钟频率必须大于等于 2MHz。在快速 I2C 模式时, APB 的时钟频率必须大于等于 8MHz。在快速+ 模式下, APB 时钟频率需大于或者等于 24MHz。	0x0	RW
ITERREN	[8]	ITERREN: 出错中断使能 (Error interrupt enable) 0: 禁止出错中断; 1: 允许出错中断。 在下列条件下, 将产生该中断: -BERR = 1; -ARLO = 1; -AF = 1; -OVR = 1; -PECERR = 1; -TIMEOUT = 1; -SMBAlert = 1。	0x0	RW
ITEVTEN	[9]	ITEVTEN: 事件中断使能 (Event interrupt enable) 0: 禁止事件中断; 1: 允许事件中断。 在下列条件下, 将产生该中断: -SB = 1 (主模式); -ADDR = 1 (主/从模式); -ADD10 = 1 (主模式); -STOPF = 1 (从模式); -TxSRE=1 (FIFO 模式下) -BTF = 1, 但是没有 TxE 或 RxNE 事件; -如果 ITBUFEN = 1, TxE 事件为 1; -如果 ITBUFEN = 1, RxNE 事件为 1。	0x0	RW
ITBUFEN	[10]	ITBUFEN: 缓冲器 (发送和接收 FIFO) 中断使能 (Buffer interrupt enable) 0: 当 TxE=1 或 RxNE=1 时, 不产生任何中断; 1: 当 TxE=1 或 RxNE=1 时, 产生事件中断(不管 DMAEN 是何种状态)。	0x0	RW
DMAEN	[11]	DMAEN: DMA 请求使能 (DMA requests enable) 0: 禁止 DMA 请求; 1: 当 TxE=1 或 RxNE =1 时, 允许 DMA 请求。	0x0	RW

Bit Name	Position	Description	Default	Access
LAST	[12]	LAST: DMA 最后一次传输 (DMA last transfer) 0: 下一次 DMA 的 EOT 不是最后的传输; 1: 下一次 DMA 的 EOT 是最后的传输。 注: 该位在主接收模式使用, 使得在最后一次接收数据时可以产生一个 NACK。	0x0	RW

Register Name: OAR1, Address: 0x40005408, Default: 0x0

表178. 自身地址寄存器 1

Bit Name	Position	Description	Default	Access
ADD0	[0]	ADD0: 接口地址 (Interface address) 7 位地址模式时不用关心。 10 位地址模式时为地址第 0 位。	0x0	RW
ADD_7_1	[1:7]	ADD[7:1]: 接口地址 (Interface address) 地址的 7~1 位。	0x0	RW
ADD_9_8	[8:9]	ADD[9:8]: 接口地址 (Interface address) 7 位地址模式时不用关心。 10 位地址模式时为地址的 9~8 位。	0x0	RW
ADDMODE	[15]	ADDMODE: 寻址模式(从模式) (Addressing mode (slave mode)) 0: 7 位从地址(不响应 10 位地址); 1: 10 位从地址(不响应 7 位地址)。	0x0	RW

Register Name: OAR2, Address: 0x4000540c, Default: 0x0

表179. 自身地址寄存器 2

Bit Name	Position	Description	Default	Access
ADD2	[1:7]	ADD2[7:1]: 接口地址 (Interface address) 在双地址模式下地址的 7~1 位。	0x0	RW
ENDUAL	[0]	ENDUAL: 双地址模式使能位 (Dual addressing mode enable) : 在 7 位地址模式下, 只有 OAR1 被识别; 1: 在 7 位地址模式下, OAR1 和 OAR2 都被识别。	0x0	RW

Register Name: DR, Address: 0x40005410, Default: 0x0

表180. 软件中断事件寄存器

Bit Name	Position	Description	Default	Access
DR	[0:7]	DR[7:0]: 8 位数据寄存器 (8-bit data register) 用于存放接收到的数据 (或者从接收 FIFO 读出的数据) 或放置用于发送到总线的数据 (或者写入到发送 FIFO 的数据) 发送器模式: 当写一个字节至 DR 寄存器时, 自动启动数据传输。一旦传输开始(TxE=1), 如果能及时把下一个需传输的数据写入 DR 寄存器, (或者发送 FIFO 非空时), I2C 模块将保持连续的数据流。 接收器模式: 接收到的字节被拷贝到 DR 寄存器(RxNE=1)。在接收到下一个字节(RxNE=1)之前读出数据寄存器, (或者接收 FIFO 非满时), 即可实现连续的数据传送。 注: 在从模式下, 地址不会被拷贝进数据寄存器 DR; 注: 硬件不管理写冲突(如果 TxE=0, 仍能写入数据寄存器); 注: 如果在处理 ACK 脉冲时发生 ARLO 事件, 接收到的字节不会被拷贝到数据寄存器里, 因此不能读到它。	0x0	RW

Register Name: SR1, Address: 0x40005414, Default: 0x0

表181. 状态寄存器 1

Bit Name	Position	Description	Default	Access
SB	[0]	SB: 起始位(主模式) (Start bit (Master mode)) – 当发送出起始条件时该位被置'1'。 – 软件读取 SR1 寄存器后, 写数据寄存器的操作将清除该位, 或当 PE=0 时, 硬件清除该位。 0: 未发送起始条件; 1: 起始条件已发送。	0x0	RW
ADDR	[1]	ADDR: 地址已被发送(主模式)/地址匹配(从模式) (Address sent (master mode)/matched (slave mode)) 在软件读取 SR1 寄存器后, 对 SR2 寄存器的读操作将清除该位, 或当 PE=0 时, 由硬件清除该位。 0: 从模式: 地址匹配, 地址不匹配或没有收到地址; 主模式: 地址发送没有结束; 1: 从模式: 地址匹配, 收到的地址匹配; 主模式: 地址发送结束;	0x0	RW
BTF	[2]	BTF: 字节发送结束 (Byte transfer finished)当 NOSTRETCH=0 时, 在下列情况下硬件将该位置'1': – 在接收时, 当收到一个新字节(包括 ACK 脉冲)且数据寄存器还未被读取 (RxNE=1)。 – 在发送时, 当一个新数据将被发送且数据寄存器还未被写入新的数据 (TxNE=1)。 – 在软件读取 SR1 寄存器后, 对数据寄存器的读或写操作将清除该位; 或在传输中发送一个起始或停止条件后, 或当 PE=0 时, 由硬件清除该位。 注: 在收到一个 NACK 后, BTF 位不会被置位。 如果下一个要传输的字节是 PEC(I2C_SR2 寄存器中 TRA 为'1', 同时 I2C_CR1 寄存器中 PEC 为'1'), BTF 位不会被置位。 0: 字节发送未完成 1: 字节发送结束	0x0	RW
ADD10	[3]	10 位头序列已发送(主模式) (10-bit header sent (Master mode)) – 在 10 位地址模式下, 当主设备已经将第一个字节发送出去时, 硬件将该位置'1'。 – 软件读取 SR1 寄存器后, 对 CR1 寄存器的写操作将清除该位, 或当 PE=0 时, 硬件清除该位。 注: 收到一个 NACK 后, ADD10 位不被置位。 0: 没有 ADD10 事件发生 1: 主设备已经将第一个地址字节发送出去	0x0	RW
STOPF	[4]	STOPF: 停止条件检测位(从模式) (Stop detection (slave mode)) – 在一个应答之后(如果 ACK=1), 当从设备在总线上检测到停止条件时, 硬件将该位置'1'。 – 软件读取 SR1 寄存器后, 对 CR1 寄存器的写操作将清除该位, 或当 PE=0 时, 硬件清除该位。 注: 在收到 NACK 后, STOPF 位不被置位。 0:	0x0	RW
TxSRE	[5]	TxSRE: 当 FIFO 模式使能时的数据发送移位器空标志, 读 SR1 后清除该位	0x0	RW

Bit Name	Position	Description	Default	Access
		0: 数据发送移位器非空 1: 数据发送移位器空		
RxNE	[6]	RxNE: 数据寄存器 (或接收 FIFO) 非空(接收时) (Data register not empty (receivers)) – 在接收时, 当数据寄存器不为空, 该位被置'1'。在接收地址阶段, 该位不被置位。 – 软件对数据寄存器的读写操作清除该位, 或当 PE=0 时由硬件清除。在发生 ARLO 事件时, RxNE 不被置位。 注: 当设置了 BTF 时, 读取数据不能清除 RxNE 位, 因为数据寄存器仍然为满。 0: 数据寄存器 (或接收 FIFO) 为空; 1: 数据寄存器 (或接收 FIFO) 非空	0x0	RW
TxE	[7]	TxE: 数据寄存器 (或发送 FIFO) 为空(发送时) (Data register empty (transmitters)) 如果收到一个 NACK, 或下一个要发送的字节是 PEC(PEC=1), 该位不被置位。 注: 在写入第 1 个要发送的数据后, 或设置了 BTF 时写入数据, 都不能清除 TxE 位, 这是因为数据寄存器仍然为空。 0: 数据寄存器 (或发送 FIFO) 非空; 1: 数据寄存器 (或发送 FIFO) 空。	0x0	RW
BERR	[8]	BERR: 总线出错 (Bus error) – 当接口检测到错误的起始或停止条件, 硬件将该位置'1'。 – 该位由软件写'0'清除, 或在 PE=0 时由硬件清除。 0: 无起始或停止条件出错; 1: 起始或停止条件出错。	0x0	RW
ARLO	[9]	ARLO: 仲裁丢失(主模式) (Arbitration lost (master mode)) 当接口失去对总线的控制给另一个主机时, 硬件将置该位为'1'。 – 该位由软件写'0'清除, 或在 PE=0 时由硬件清除。 在 ARLO 事件之后, I2C 接口自动切换回从模式(M/SL=0)。 注: 在 SMBUS 模式下, 在从模式下对数据的仲裁仅仅发生在数据阶段, 或应答传输区间(不包括地址的应答)。 0: 没有检测到仲裁丢失; 1: 检测到仲裁丢失。	0x0	RW
AF	[10]	AF: 应答失败 (Acknowledge failure) – 当没有返回应答时, 硬件将置该位为'1'。 – 该位由软件写'0'清除, 或在 PE=0 时由硬件清除。 0: 没有应答失败; 1: 应答失败。	0x0	RW
OVR	[11]	OVR: 过载/欠载 (Overrun/Underrun) – 当 NOSTRETCH=1 时, 在从模式下该位被硬件置位, 同时: – 在接收模式中当收到一个新的字节时(包括 ACK 应答脉冲), 数据寄存器里的内容还未被读出, 则新接收的字节将丢失。 – 在发送模式中当要发送一个新的字节时, 却没有新的数据写入数据寄存器, 同样的字节将被发送两次。 – 该位由软件写'0'清除, 或在 PE=0 时由硬件清除。。 注: 如果数据寄存器的写操作发生时间非常接近 SCL 的上升沿, 发送的数据是不确定的, 并发生保持时间错误。	0x0	RW

Bit Name	Position	Description	Default	Access
		0: 无过载/欠载; 1: 出现过载/欠载。		
PECERR	[12]	PECERR: 在接收时发生 PEC 错误 (PEC Error in reception) – 该位由软件写'0'清除, 或在 PE=0 时由硬件清除。 0: 无 PEC 错误: 接收到 PEC 后接收器返回 ACK(如果 ACK=1); 1: 有 PEC 错误: 接收到 PEC 后接收器返回 NACK(不管 ACK 是什么值)。	0x0	RW

Register Name: SR2, Address: 0x40005418, Default: 0x0

表182. 状态寄存器 2

Bit Name	Position	Description	Default	Access
MSL	[0]	MSL: 主从模式 (Master/slave) – 当接口处于主模式(SB=1)时, 硬件将该位置位; – 当总线上检测到一个停止条件、仲裁丢失(ARLO=1 时)、或当 PE=0 时, 硬件清除该位。 0: 从模式 1: 主模式	0x0	R(ReadOnly)
BUSY	[1]	BUSY: 总线忙 (Bus busy) – 在检测到 SDA 或 SCI 为低电平时, 硬件将该位置'1'; – 当检测到一个停止条件时, 硬件将该位清除。 该位指示当前正在进行的总线通讯, 当接口被禁用(PE=0)时该信息仍然被更新 0: 在总线上无数据通讯; 1: 在总线上正在进行数据通讯	0x0	R(ReadOnly)
TRA	[2]	TRA: 发送/接收 (Transmitter/receiver) 在整个地址传输阶段的结尾, 该位根据地址字节的 R/W 位来设定。 在检测到停止条件(STOPF=1)、重复的起始条件或总线仲裁丢失 (ARLO=1)后, 或当 PE=0 时, 硬件将其清除。 0: 接收到数据; 1: 数据已发送	0x0	R(ReadOnly)
GENCALL	[4]	GENCALL: 广播呼叫地址(从模式) (General call address (Slave mode)), – 在产生一个停止条件或一个重复的起始条件时, 或 PE=0 时, 硬件将该位清除。 0: 未收到广播呼叫地址; 1: 当 ENG=1 时, 收到广播呼叫的地址。	0x0	R(ReadOnly)
DUALF	[7]	双标志(从模式) (Dual flag (Slave mode)), 在产生一个停止条件或一个重复的起始条件时, 或 PE=0 时, 硬件将该位清除。 0: 接收到的地址与 OAR1 内的内容相匹配 1: 接收到的地址与 OAR2 内的内容相匹配。	0x0	R(ReadOnly)
PEC	[8:15]	数据包出错检测 (Packet error checking register) 当 ENPEC=1 时, PEC[7:0]存放内部的 PEC 的值。	0x0	R(ReadOnly)

Register Name: CCR, Address: 0x4000541c, Default: 0x0

表183. 时钟控制寄存器

Bit Name	Position	Description	Default	Access
CLKC	[0:11]	CLKC[11:0] : 主机模式下 I2C 时钟控制	0x0	RW

Bit Name	Position	Description	Default	Access
		标准速度模式下: $T_{high}=T_{low}=CLKC*TPCLK1$ 如果 DTCY=0, 快速模式或快速+模式下: $T_{high}=CLKC*TPCLK1$, $T_{low}=2*CLKC*TPCLK1$ 如果 DTCY=1, 快速模式或快速+模式下: $T_{high}=9*CLKC*TPCLK1$, $T_{low}=16*CLKC*TPCLK1$		
DUTY	[14]	DUTY: 快速模式时的占空比 (Fast mode duty cycle) 0: 快速模式下: $T_{low}/T_{high} = 2$; 1: 快速模式下: $T_{low}/T_{high} = 16/9$	0x0	RW
F_S	[15]	F/S: I2C 主模式选项 (I2C master mode selection) 0: 标准模式的 I2C 1: 快速模式的 I2C	0x0	RW

Register Name: TRISE, Address: 0x40005420, Default: 0x0

表184. TRISE 寄存器

Bit Name	Position	Description	Default	Access
TRISE	[0:6]	TRISE[6:0]: 在快速/ 标准模式下的最大上升时间(主模式) (Maximum rise time in Fast/Standard mode (Master mode)) 这些位必须设置为 I2C 总线规范里给出的最大的 SCL 上升时间, 增长步幅为 1。 例如: 标准模式中最大允许 SCL 上升时间为 1000ns。如果在 I2C_CR2 寄存器中 FREQ[6:0]中的值等于 0x08 且 TPCLK1=125ns, 故 TRISE[6:0]中必须写入 09h($1000ns/125ns = 8+1$)。 滤波器的值也可以加到 TRISE[6:0]内。 如果结果不是一个整数, 则将整数部分写入 TRISE[6:0]以确保 tHIGH 参数。 注: 只有当 I2C 被禁用(PE=0)时, 才能设置 TRISE[6:0]	0x0	RW

Register Name: FMPCFG, Address: 0x40005424, Default: 0x0

表185. 快速+模式配置寄存器

Bit Name	Position	Description	Default	Access
FMPEN	[0]	FMPEN 快速+ 模式使能 当该位被置 1 时, I2C 设备支持高达 1MHz 0: 失能快速+ 模式 1: 使能快速+ 模式	0x0	RW
FIFO_EN	[1]	FIFO_EN FIFO 使能 0: 禁用发送和接收数据 FIFO 1: 使能发送和接收数据 FIFO	0x0	RW
RX_FIFO_FULL	[2]	RX_FIFO_FULL 0: 接收数据 FIFO 不满 1: 接收数据 FIFO 满	0x0	RW
TX_FIFO_FULL	[3]	TX_FIFO_FULL 0: 发送数据 FIFO 不满 1: 发送数据 FIFO 满	0x0	RW

21. 串行外设通信接口 (SPI)

21.1. SPI 概述

SPI 特性如下:

- 支持三线读命令模式
- 主从模式
- Tx 和 Rx 都包含 8 字节 FIFO
- 中断可以配置位 1, 2, 3, 4, 5, 6, 7, 8 字节传输后产生中断
- 支持全双工通信
- CS 可由软件控制
- 支持 DMA
- 可配置的时钟相位和极性
- 线或输出模式
- 数据开漏输出模式
- LSB 或 MSB 传输模式
- 回环模式
- 发送和中断模式
- 连续传输模式
- Rx Overflow 和 Tx Underflow

工业级同步串行 SPI 接口可以全双工模式传输 8bit 数据。SPI 主和从模式下最高 SCLK 时钟速率可达 HCKL/4。在系统时钟为 32MHz 时, 最快速率为 8MHz。SPI 模块支持 DMA, 发送和接收可以分别配置不同的 DMA 通道。

21.2. SPI 功能

SPI 主或从模式下四线构成为: MISO, MOSI, SCLK, CS

注: 在使用 SPI 引脚之前需要先将 GPIO 配置为对应的 SPI 接口, 具体配置参考 GPIO 章节。

21.2.1. 主入从出 (MISO) 引脚

MISO 引脚在配置为 SPI 主模式时是输入引脚, 在配置为 SPI 从模式时是输出引脚。主 SPI 的 MISO 应该连接到从 SPI 的 MISO 引脚。默认情况下, 数据以字节 (8bit) 为单位传输, 高 bit 在前。

21.2.2. 主出从入 (MOSI) 引脚

MOSI 引脚在 SPI 主模式时是输出引脚, 在从模式时是输入引脚。主 SPI 的 MOSI 引脚同样应该连接到从 SPI 的 MOSI 引脚。默认情况下, 数据以字节 (8bit) 为单位传输, 高 bit 在前。

21.2.3. 串行时钟 (SCLK) 引脚

SPI 主设备通过 SCLK 同步发送或接收的数据。因此, 一个字节数据通过 8 个 SCLK 时钟周期传输。主 SPI 模式下, SCLK 是输出引脚, 在从 SPI 模式下, SCLK 为输入引脚。

SCLK 时钟由主 SPI 产生, 在主模式下, 通过配置寄存器 CTL[12] 和 CTL[13] 来配置 SCLK 的相位和极性。SCLK 的速率由 DIV 寄存器配置, 速率计算如下:

$$F_{SCLK} = \frac{SPICLK}{2 \times (1 + DIV)}$$

SPICLK 为该 SPI 所处的总线时钟（参考系统时钟或地址映射章节），如：参考 SPI 模块位于 APB2,则 SPICLK 为 APB2 的时钟。

SPICLK 越低，对应 SPI 模块的功耗就越低。最大 SPICLK 为 TBD.

在从模式下，同样需要配置合理的 SCLK 相位和极性。

在主和从模式下，数据都是在 SCLK 的一个沿发送，在另一个沿采样。因此，主从设备的 SCLK 时钟相位和极性应该一致。

21.2.4. 片选 (CS) 引脚

CS 低电平有效，CS 在传输开始时为低电平，在传输完成时为高电平。

CS 在主模式下是输出引脚，在从模式下是输入引脚。

21.2.5. 启动 SPI 传输

主模式下，SPI 根据 CTL[2]决定启动传输的行为，若 CTL[2]为 1，SPI 在写入 Tx FIFO 后启动传输，若 CTL[2]为 0，SPI 则由读取 Rx FIFO 触发传输。在 SPI 的传输过程中读 Rx 并不会触发另一个传输。

如果三线模式 (DMA[9]) 没有使能，SPI 默认工作在全双工模式，即 SPI 在把 Tx FIFO 的数据送往 MOSI 的同时，MISO 的数据也会填入 Rx FIFO. 当 FIFO 接收的数据超过 FIFO 的深度时，会产生 Overflow 状态，通过配置 IEN 寄存器可以使能或者禁用相应的中断。

连续传输在 CS 线上的表现为：多个字节传输中间 CS 一直为低；而非连续传输时，每传输完一个字节，CS 都会拉高。

CTL[11]为 1 时，SPI 为连续传输，直到 Tx FIFO 中没有数据，传输完成，CS 自动置 1。

CTL[11]为 0 时，SPI 为非连续传输，每传送 8bit,CS 都会置 1.

CTL[2]为 0 时读 Rx FIFO 触发传输。如果 CNT 寄存器大于 0，则 SPI 发送 CNT 指定字节数的 SCLK 时钟，同时根据 CTL[11]是否为连续模式操作 CS。比如 CNT=10,CTL[11]=1 连续传输时,读 Rx FIFO 启动传输，SPI 发送 10 个字节的 SCLK 以读取三个字节数据且整个期间 CS 为低，如果 Rx FIFO 已经满了，SPI 停止发送时钟直到 Rx FIFO 不满。而 CTL[11]=0 时，SPI 每读取一个字节 CS 都会置高。

21.2.6. FIFO Underflow 和 Overflow

对于 Tx FIFO，如果 ZEN (CTL[7]) =0，当 Tx FIFO 为空，而传输被触发，SPI 发送之前已经发送的最后一个数据；同样的条件下，如果 ZEN (CTL[7]) =1，SPI 发送数据 0。

对于 Rx FIFO，如果 CTL[5]=1，当 Rx FIFO 满，Rx FIFO 的最后一个数据将会被新接收的数据覆盖；同样的条件下，如果 CTL[5]=0，新数据会被忽略。

21.2.7. SPI 中断

SPI 通过一条中断线输出。STAT 反应对应的中断状态，根据 CTL[2]的值时 Tx 传输还是 Rx 传输，来决定是产生 Tx 中断还是 Rx 中断。

STAT 中的所有中断状态都需要写 1 清零。中断线的清除需要所有中断源清零。

21.2.8. 读指令模式

读指令模式适用于需要写入指令或者地址来读取 SPI 从设备内部数据或寄存器的场景。下图示例了该模式下 (DMA[3]=1) 不同配置是 SPI 总线上的波形。

下图为三线模式下,Tx Byte0 由主发送, Rx Byte0 由从 SPI 发送。

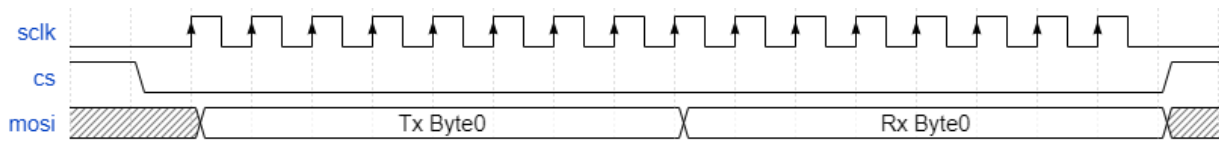


图142. TXBYTES=0, CNT=2, OVERLAP=0, 三线模式

四线模式下

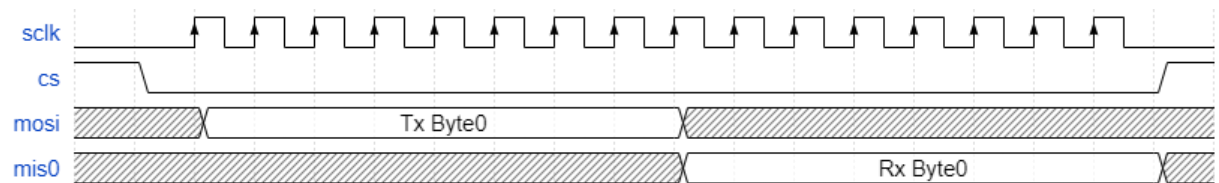


图143. TXBYTES=0, CNT=2, OVERLAP=0, 四线模式

四线模式下, Overlap=1

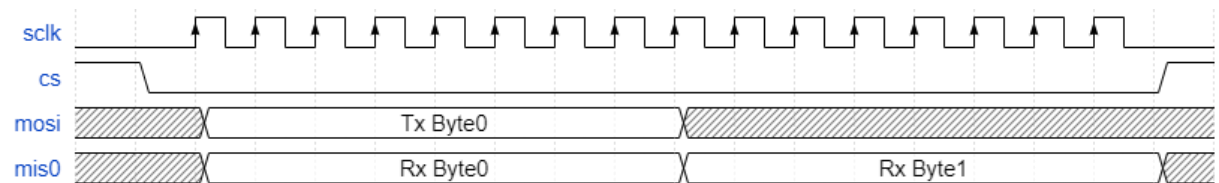


图144. TXBYTES=0, CNT=3, OVERLAP=1, 四线模式

21.3. SPI 寄存器描述

Register Name: CTL, Address: 0x40013000, Default: 0x0

表186. SPI 控制寄存器 1

Bit Name	Position	Description	Default	Access
SPIEN	[0]	SPI 使能 0x0: 关闭 SPI 0x1: 使能 SPI	0x0	RW
MASEN	[1]	主 (master) 模式使能 0x0: 使能从模式	0x0	RW

Bit Name	Position	Description	Default	Access
		0x1: 使能主模式		
TIM	[2]	PI 传输和中断模式 Cleared by user to initiate transfer with a read of the RX register. Interrupt only occurs when Rx-FIFO has IEN[2:0]+1 number of bytes or more. Set by user to initiate transfer with a write to the TX register. Interrupt only occurs when IEN [2:0] +1 number of bytes have been transmitted. 0x0: 关闭 0x1: 打开	0x0	RW
ZEN	[3]	传输 0 使能 0x0: 当发送 FIFO 没有有效数据时, 默认状态传输上一笔传输的数据 0x1: 当发送 FIFO 没有有效数据时, 默认状态传输 0x00	0x0	RW
RXOF	[4]	接收 (RX) 溢出时 (overflow) 数据覆盖使能 0x0: 新接收的数据会被丢弃 0x1: 新接收的数据会被覆盖	0x0	RW
OEN	[5]	接收 (RX) 溢出时 (overflow) 数据覆盖使能 0x0: 新接收的数据会被丢弃 0x1: 新接收的数据会被覆盖	0x0	RW
LSB	[6]	LSB 先进行传输模式 0x0: MSB 先进行传输 0x1: LSB 先进行传输	0x0	RW
RFLUSH	[7]	SPI 清 Rx FIFO 使能 0x0: 关闭 Rx FIFO 的清功能 0x1: 置 1 此位清除 Rx FIFO 的数据, 如果此位被置位, 后面的数据和中断不会产生。 This bit does not clear itself and must be toggled if a single flush is required. If set and TIM = 0, a read of the Rx FIFO initiates a transfer.	0x0	RW
TFLUSH	[8]	SPI 清 Tx FIFO 使能 0x0: 关闭 Tx FIFO 的清功能 0x1: 置 1 此位清除 Rx FIFO 的数据, 如果此位被置位, 后面的数据和中断不会产生。 This bit does not clear itself and must be toggled if a single flush is required. If this bit is left high, either the last transmitted value or 0x00 is transmitted depending on the ZEN bit. Any writes to the Tx FIFO are ignored while this bit is set.	0x0	RW
LOOPBACK	[9]	回环 (loopback) 使能 0x0: 关闭 0x1: 开启回环模式, 内部将 MISO 和 MOSI 连接	0x0	RW
CON	[10]	连续传输模式使能 (Continuous) 0x0: 关闭连续传输模式, 每笔传输包含 8 位串行数据 0x1: 开启连续传输模式, 主模式时, 只要 Tx 寄存器中有有效数据, CS 一直保持有效传输电平, 数据以 8 位串行传输直至 Tx 寄存器空	0x0	RW
CSRST	[11]	片选 CS 复位 0x0: 0: If this bit is clear, the bit counter will continue from where it stopped. SPI can receive the remaining bits when CS is asserted and user code has	0x0	RW

Bit Name	Position	Description	Default	Access
		to ignore the SPI_STAT.CSERR interrupt. 0x1: 1: If this bit is set, the bit counter will be reset after a CS error condition and the user code is expected to clear the CTL[0] (SPIEN). It is strongly recommended to set this bit for a graceful recovery after a CS error.		
CPHA	[12]	时钟相位模式 0x0: 时钟前沿采样 0x1: 时钟后沿采样	0x0	RW
CPOL	[13]	时钟极性选择 0x0: 时钟空闲为低电平 0x1: 时钟空闲为高电平	0x0	RW
WOM	[14]	SPI 线或模式, 打开数据输出使能, 此时管脚需要外部上拉 0x0: 关闭 0x1: 打开	0x0	RW

Register Name: IEN, Address: 0x40013004, Default: 0x0

表187. SPI 控制寄存器 2

Bit Name	Position	Description	Default	Access
CS	[0]	CS 中断使能 从连续模式下, 中断产生于每个 CS 边沿 0x0: 关闭中断 0x1: 当从连续模式时, 如果此位被置位, 每个 CS 边沿都会产生中断, STAT[13], STAT[14]会被置位。 当不是连续模式, 或工作于主模式时, 此位配置不生效	0x0	RW
TXUNDR	[1]	Tx 下溢 (underflow) 中断使能 0x0: 关闭中断 0x1: 开启中断	0x0	RW
TXDONE	[2]	SPI 发送完成中断使能。此中断状态位显示于 STAT[3] 只工作于读指令模式 (read command) 0x0: 关闭中断 0x1: 开启中断	0x0	RW
TXEMPTY	[3]	Tx FIFO 空中断使能 0x0: 关闭中断 0x1: 开启中断	0x0	RW
RXOVR	[4]	Rx 上溢 (overflow) 中断使能 0x0: 关闭中断 0x1: 开启中断	0x0	RW
RDY	[5]	Ready 信号边沿中断使能 当 RDY/MISO 边沿时, 产生此中断, 并于 STAT[15]显示状态位	0x0	RW
XFRDONE	[6]	SPI 传输完成中断使能。此中断状态位显示于 STAT[3] 0x0: 关闭中断 0x1: 开启中断	0x0	RW
IRQMODE	[7:9]	SPI 中断模式选择, 当传输过程中 Tx/Rx 中断发生时, 配置如下选择有效。 当 DMA 模式 Rx 传输时, 此位应为 3'b000 0x0: 中断发生于 1 字节数据发送或接收完成。	0x0	RW

Bit Name	Position	Description	Default	Access
		0x1: 中断发生于 2 字节数据发送或接收完成。 0x2: 中断发生于 3 字节数据发送或接收完成。 0x3: 中断发生于 4 字节数据发送或接收完成。 0x4: 中断发生于 5 字节数据发送或接收完成。 0x5: 中断发生于 6 字节数据发送或接收完成。 0x6: 中断发生于 7 字节数据发送或接收完成。 0x7: 中断发生于 8 字节数据发送或接收完成。		

Register Name: STAT, Address: 0x40013008, Default: 0x100

表188. 状态寄存器

Bit Name	Position	Description	Default	Access
IRQ	[0]	SPI 中断状态 0x0: SPI 中断源都被清除或没发生中断 0x1: SPI 有中断发生	0x0	R(ReadOnly)
TXEMPTY	[1]	SPI TX FIFO 空中断 读 STAT 会清除此位 当 TX FIFO 空同时 IEN[14]位被置位, 此位被置 1	0x0	RW
TXDONE	[2]	SPI 发送完成中断状态。此中断状态位显示于 STAT[3] 只工作于读指令模式 (read command)	0x0	RW
TXUNDR	[3]	Tx 下溢 (underflow) 中断状态	0x0	RW
TXIRQ	[4]	SPI Tx IRQ 状态。当指定字节的数据被发送时置 1	0x0	RW
RXIRQ	[5]	Rx IRQ, 当指定字节的数据被接收时置 1	0x0	RW
RXOVR	[6]	Rx FIFO 溢出	0x0	RW
XFRDONE	[7]	SPI 传输完成。此 bit 表示 master 模式下 SPI 传输完成, 读此位会清除为 0 当 CNT 寄存器设定的数量帧传输完成时, 此位被置 1	0x0	RW
CS	[8]	CS 状态	0x1	RW
CSERR	[9]	CS 错误	0x0	RW
CSRISE	[10]	CS 上升沿	0x0	RW
CSFALL	[11]	CS 下降沿	0x0	RW

Register Name: FIFOSTAT, Address: 0x4001300c, Default: 0x0

表189. FIFO 状态寄存器

Bit Name	Position	Description	Default	Access
TX	[0:3]	SPI Tx FIFO 状态。当 DMA 关闭时, 此位表示为字节数量, 当 DMA 使能时, 此位表示为半字数量 (DMA 为 16 位传输) 0x0: Tx FIFO 空 0x1: 1 有效字节/半字 0x2: 2 有效字节/半字 0x3: 3 有效字节/半字 0x4: 4 有效字节/半字 0x5: 5 有效字节/半字 0x6: 6 有效字节/半字 0x7: 7 有效字节/半字 0x8: 8 有效字节/半字	0x0	RW
RX	[8:11]	SPI Rx FIFO 状态。当 DMA 关闭时, 此位表示为字节数量, 当 DMA 使能时,	0x0	RW

Bit Name	Position	Description	Default	Access
		此位表示为半字数量 (DMA 为 16 位传输) 0x0: Rx FIFO 空 0x1: 1 有效字节/半字 0x2: 2 有效字节/半字 0x3: 3 有效字节/半字 0x4: 4 有效字节/半字 0x5: 5 有效字节/半字 0x6: 6 有效字节/半字 0x7: 7 有效字节/半字 0x8: 8 有效字节/半字		

Register Name: RX, Address: 0x40013010, Default: 0x0

表190. 接收 (RX) 寄存器

Bit Name	Position	Description	Default	Access
BYTE1	[0:7]	8 比特接收缓冲器	0x0	R(ReadOnly)
BYTE2	[8:15]	8 比特接收缓冲器, 仅在 DMA 模式使用, 所有 FIFO 的访问都是半字。 如果 DMA 功能关闭, 此位返回 0	0x0	R(ReadOnly)

Register Name: TX, Address: 0x40013014, Default: 0x0

表191. 发送 (TX) 寄存器

Bit Name	Position	Description	Default	Access
BYTE1	[0:7]	8 比特发送缓冲器	0x0	W(WriteOnly)
BYTE2	[8:15]	8 比特发送缓冲器, 仅在 DMA 模式使用, 所有 FIFO 的访问都是半字。 如果 DMA 功能关闭, 此位返回 0	0x0	W(WriteOnly)

Register Name: DIV, Address: 0x40013018, Default: 0x0

表192. SPI 波特率选择寄存器

Bit Name	Position	Description	Default	Access
DIV	[0:5]	SPI 时钟分频。DIV 是用于根据 PCLK 分频的算式因子	0x0	RW

Register Name: CNT, Address: 0x4001301c, Default: 0x0

表193. 传输字节计数寄存器

Bit Name	Position	Description	Default	Access
VALUES	[0:13]	传输字节计数。 COUNT indicates the number of bytes to be transferred. Count is used in both receive and transmit transfer types. The COUNT value ensures that a master mode transfer terminates at the proper time and that 16-bit DMA transfers are byte padded or discarded as required to match odd transfer counts. Reset by clearing CTL [0] or if CNT register is updated.	0x0	RW
FRAMECOUNT	[14]	延长 SPI 数据传输模式。需要配合 CON[11]和 CNT[13:0]寄存器。 Notes: If CNT[13:0] = 0, then this field has no effect as the SPI master will continue with transfers as long as Tx/Rx FIFO is ready. If CON[11] = 0, then also this field has no effect as all SPI frames are single byte wide irrespective of other control fields. This bit is cleared when the SPI master transfers only one frame of CNT	0x0	RW

Bit Name	Position	Description	Default	Access
		[13:0] bytes. Set to 1 when the SPI master will transfer data in frames of CNT [13:0] bytes each.		

Register Name: DMA, Address: 0x40013020, Default: 0x0

表194. SPI DMA 使能寄存器

Bit Name	Position	Description	Default	Access
EN	[0]	DMA 使能 0x0: 关闭 DMA 0x1: 开启 DMA	0x0	RW
TXEN	[1]	发送 DMA 请求使能。 0x0: 关闭 DMA 0x1: 开启 DMA	0x0	RW
RXEN	[2]	接收 DMA 请求使能。 0x0: 关闭 DMA 0x1: 开启 DMA	0x0	RW
CMDEN	[3]	读命令模式使能, 命令+地址读模式 0x0: 读命令模式关闭 0x1: 读命令模式使能	0x0	RW
OVERLAP	[4]	Tx/Rx 重叠模式。读模式下的 Tx 与 Rx 重叠 0x0: 重叠模式关闭 0x1: 重叠模式使能	0x0	RW
TXBYTES	[5:8]	读命令模式传输字节数。该位应该写入要发送的字节数-1	0x0	RW
THREEPIN	[9]	SPI3 线模式使能, MOSI 即做输出也做输入 0x0: 4 管脚模式 0x1: 3 管脚模式	0x0	RW

22. QSPI

QUADSPI 是一种专用的通信接口, 连接单、双或四 (条数据线) 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 映射到微控制器地址空间, 从而系统将其视作内部存储器

22.1. 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
- 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
- 支持单倍数据速率 (SDR 模式)
- 内存映射模式下集成接收 buffer, 用于缓存接收的 4 个 word 数据
- 允许 8、16 和 32 位数据访问
- 具有适用于内存映射模式操作的 DMA 通道

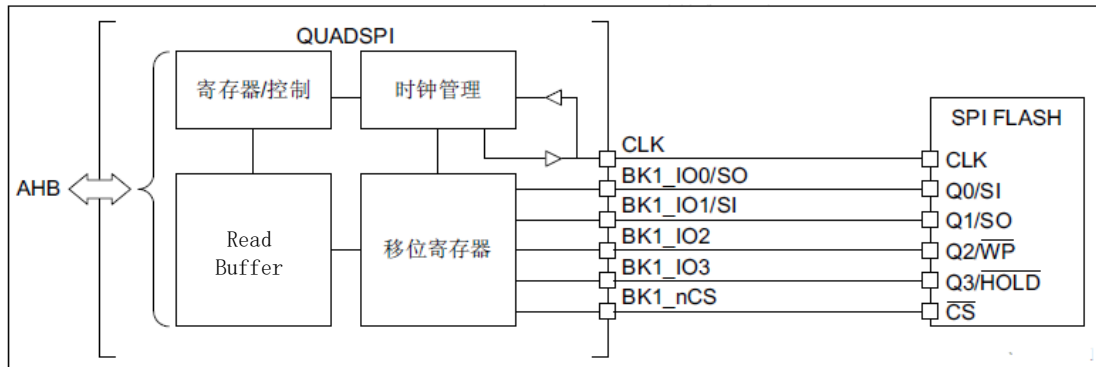
- 操作完成以及发生访问错误时产生中断

不支持如下特性:

- 双闪存模式, 通过并行访问两个 Flash, 可同时发送/接收 8 位数据
- 集成 FIFO, 用于发送和接收
- 间接模式操作不支持 DMA 通道, 只有直接模式支持 DMA 通道
- 无达到 FIFO 阈值、超时中断
- 不支持双倍数据速率 (DDR) 模式, 既在 CLK 的每个上升沿和下降沿发送和接收数据

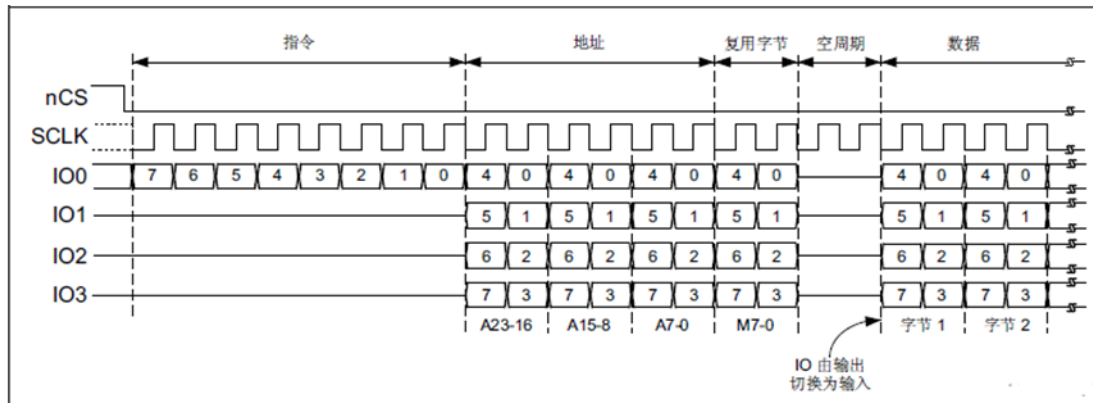
22.2. QUADSPI 功能说明

22.2.1. QUADSPI 框图



22.2.2. 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。



指令阶段

这一阶段, 将在 QUADSPI_CCR[7:0] 寄存器的 INSTRUCTION 字段中配置的一条 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] 字段进行配置。

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] 字段中进行配置。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] 字段进行配置。

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] 字段中进行配置。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] 字段进行配置。

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 5:4, 低 2 位存放在位 1 和 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] 字段中指定。

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“周转”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)、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 ADMODE[1:0] 字段进行配置。

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。

22.2.3. 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 字段设置为 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 字段设置为 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 字段设置为 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 Quad 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

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。

22.2.4. 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。有效值为 0~3, 既数据长度为 1~4 个字节。如果不传输任何字节, DMODE (QUADSPI_CCR[25:24]) 应设置为 00。

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。

触发命令启动

当给寄存器 OP_RUN 写 1 后, 将按照寄存器的配置开始间接模式的操作, 直到操作完成 OP_RUN 自动置 0。

22.2.5. 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 的访问启动与在间接读取模式下相同: 当给寄存器 OP_RUN 写 1 后, 将按照寄存器的配置开始状态标志轮询模式的操作, 直到如果自动轮询模式停止 (APMS) 位置 1 或者检测到匹配时 OP_RUN 自动置 0。

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, 从而不考虑该位。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[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。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。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。

22.2.6. 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 2GB。

如果访问的地址超出 FSIZE 定义的范围但仍在 2GB 范围内, 则生成 AHB 总线错误 (TODO)。此错误的影响具体取决于尝试进行访问的 AHB 主设备:

- 如果为 Cortex® CPU, 则生成硬性故障 (Hard fault) 中断。
- 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 接受下一个微控制器访问并提前加载后面地址中的字节到预取 buffer。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。

BUSY 在第一个存储器映射访问发生时变为高电平。由于进行预取操作, BUSY 在发生中止或外设禁止前不会下降。

22.2.7. 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] 字段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 4GB (使用 32 位进行寻址), 但在内存映射模式下的可寻址空间限制为 2GB。

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) 字段指定 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)。

22.2.8. 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[4]), 可将数据采样移位半个 CLK 周期。

22.2.9. QUADSPI 配置

QUADSPI 配置分两个阶段: QUADSPI IP 配置和 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。

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 CKMODE 位中编程功能模式（模式 0 或模式 3）。

22.2.10. 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。
5. 在 QUADSPI_AR 中指定目标地址
6. 通过 QUADSPI_DR 写入数据（如果是需要写操作）
7. 写 OP_RUN 为 1 启动操作
8. 查询或者中断方式 OP_RUN 自动为 0，表操作完成。（ST 看 BUSY 信号）
9. 读取 QUADSPI_DR 的数据（如果是需要读操作）

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”
- 采样移位设置 (SSHIFT)
- 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 (TODO)

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）

状态标志轮询模式操作步骤

将 FMODE 字段 (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。

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 AHB 主器件访问存储器映射空间时，将发送已编程的指令和帧。

数据长度寄存器 (QUADSPI_DLR) 在内存映射模式中无意义。

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。

在内存映射模式下，当 AHB 主器件执行的访问超出范围或 QUADSPI 被禁止时：将产生 AHB 错误，以响应故障 AHB 请求。

当 AHB 主机访问存储器映射空间，但内存映射模式被禁止时：将产生 AHB 错误，以响应故障 AHB 请求。

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，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。

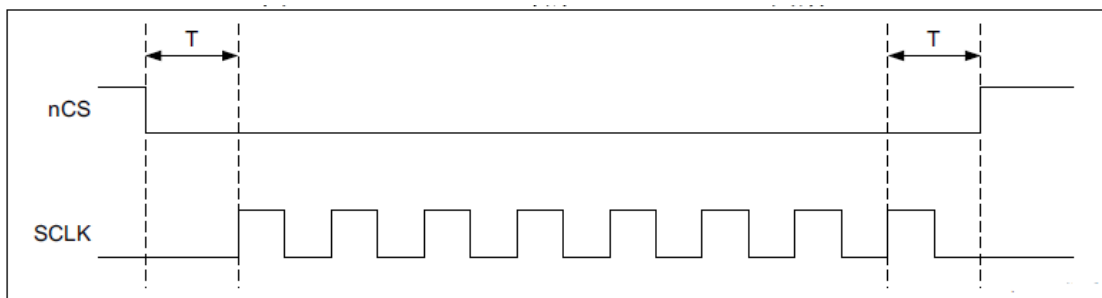
任何操作都可通过将 QUADSPI_CR 中的 ABORT 位置 1 来中止。在完成中止时，BUSY 位和 ABORT 位自动复位。

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。

nCS 行为

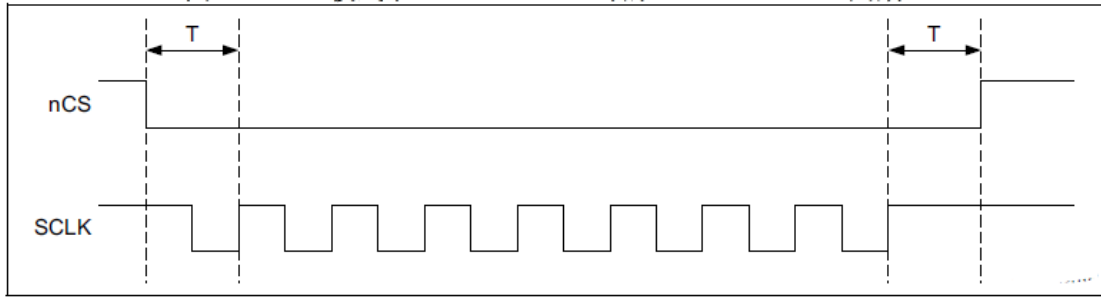
默认情况下，nCS 为高电平，取消选择外部 Flash。nCS 在操作开始前下降，在操作完成时立即上升。

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 周期后升高至高电平，如图所示。



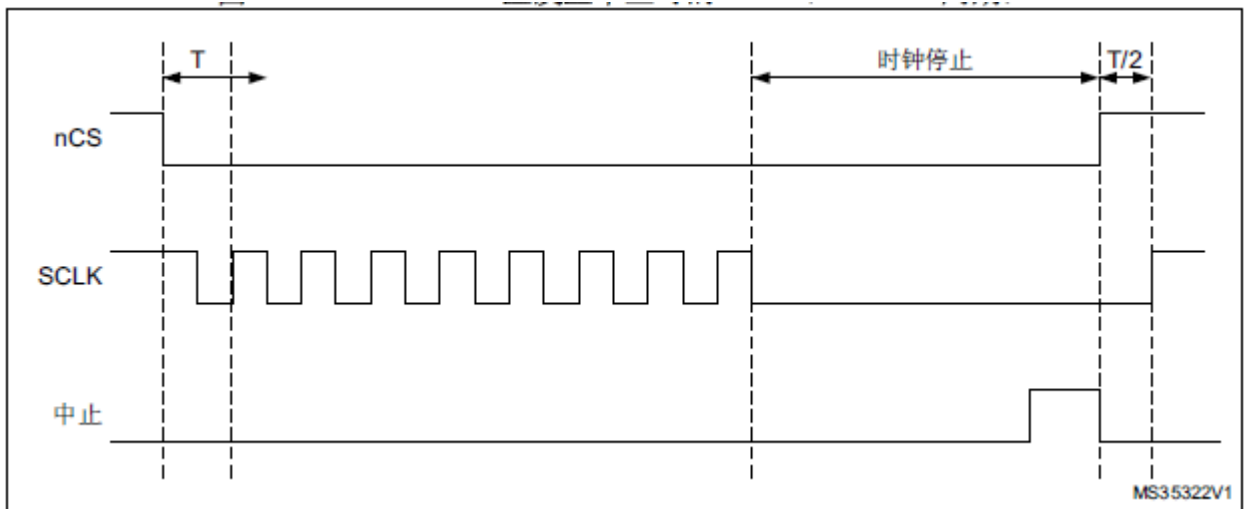
CKMODE = 0 时的 nCS (T = CLK 周期)

当 CKMODE = 1（“模式 3”，在未进行任何操作时 CLK 升高至高电平）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 周期后升高至高电平，如图所示。



CKMODE = 1 时的 nCS (T = CLK 周期)

注意在内存映射模式下，如果总线在每次读或写完成后处于空闲态，那么 SCLK 将保持低电平不跳变，CSn 也处于拉低状态，直到下一次总线发起有效的读或者写 sclk 才开始跳变。如果前后两次都是读操作且读地址连续，或者前后两次都是写操作且写地址连续，CSn 继续保持低电平，QSPI 继续执行数据阶段，不用从指令阶段开始重新执行；否则 CSn 会先拉高再拉低重新从指令阶段开始执行新的 QSPI 周期。内存映射模式下，总线读完后，如果为降低功耗不想让 SCLK 和 CSn 继续保持低电平，可以写 abort 为 1 停止 QSPI 周期；或者总线写完后，flash 厂商规定必须将 CSn 拉高让 Flash 内部开始烧写，就必须写 abort 为 1 停止 QSPI 周期。如下图所示。



需要注意的是，上图发生 abort 后 SCLK 比 nCS 晚半个周期才拉高只适用于 CKMODE=1 时的写数据周期，在 CKMODE=1 时的其他周期，SCLK 和 nCS 将被 abort 同时拉高；CKMODE=0 时的任何周期只有 nCS 被 abort 拉高。

内存映射模式写操作下 nCS 行为的进一步说明

因为绝大多数厂商的 QSPI Flash 产品页大小为 256bytes，一次 nCS 低电平周期最多可以写入 256bytes 的数据且地址不能超过页边界，否则会冲掉本次 nCS 低电平周期已经写入的数据。所以在一次 nCS 低电平周期中，内存映射模式写操作的字节超过 256bytes 或者地址超过页边界，nCS 会自动拉高，QSPI flash 将开始内部烧写。超过 256bytes 或者页边界的那部分数据写入可能无法成功。

在一次 nCS 低电平周期中，内存映射模式写操作的字节在 256bytes 或者页边界内的话，就必须写 abort 强制将 nCS 拉高结束低电平周期，QSPI flash 将开始内部烧写。

22.2.11. QUADSPI 中断

可以使用单独的中断使能位以提高灵活性。发生如下事件时可生成中断：

- 状态匹配
- 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。

QUADSPI 中断请求

中断事件	事件标志	使能控制位
------	------	-------

状态匹配	SMF	SMIE
传输完成	TCF	TCIE
传输错误	TEF	TEIE

22.3. QSPI 寄存器描述

Register Name: CR, Address: 0x40003C00, Default: 0x0

表195. 接收 (RX) 寄存器

Bit Name	Position	Description	Default	Access
PRESCALER	[26:24]	时钟预分频器 (Clock prescaler) 该字段定义基于 AHB 时钟生成 CLK 所用的分频系数。 0xx: HCLK 不分频 100: HCLK 2 分频 101: HCLK 4 分频 110: HCLK 8 分频 111: HCLK 16 分频 仅可在 BUSY = 0 时修改该字段。	0x0	RW
PMM	[23]	轮询匹配模式 (Polling match mode) 该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 时修改该位。	0x0	RW
APMS	[22]	自动轮询模式停止 (Automatic poll mode stop) 该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 时修改该位。	0x0	RW
RESERVED	[21:20]	保留	0x0	RW
SMIE	[19]	状态匹配中断使能 (Status match interrupt enable) 该位使能状态匹配中断。 0: 禁止中断 1: 使能中断	0x0	RW
RESERVED	[18]	保留	0x0	RW
TCIE	[17]	传输完成中断使能 (Transfer complete interrupt enable) 该位使能传输完成中断。 0: 禁止中断 1: 使能中断	0x0	RW
TEIE	[16]	传输错误中断使能 (Transfer error interrupt enable) 该位使能传输错误中断。 0: 禁止中断 1: 使能中断	0x0	RW
RESERVED	[15:5]	保留	0x0	RW
SSSHIFT	[4]	采样移位 (Sample shift)	0x0	RW

		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。使用该位, 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 仅可在 BUSY = 0 时修改该字段。		
RESERVED	[3:2]	保留	0x0	RW
ABORT	[1]	中止请求 (Abort request) 该位中止执行中的命令序列。在中止完成时该位自动复位。该位可停止当前的传输。 注意: 由于外部 flash 往往一次指令周期最多写入的数据量有上限, 所以要求 QSPI 控制器一次指令周期编程的数据量要等于或者小于厂商提供的 Flash 数据手册上注明的最大值。并且在 QSPI 输出最后一个字节完成后, 必须将 CSn 拉高才能成功将本次指令周期 QSPI 接口输出的数据编程进入 flash。因此, 在使用内存映射模式编程外部 Flash 时, AHB 总线在一次指令周期内写入 flash 的数据必须小于等于 Flash 厂商规定的最大值, AHB 总线写完本次指令周期的数据后, CPU 需要将 abort 位拉高, 从而 CSn 拉高使得外部 flash 开始内部编程操作。要想进行下一次编程指令, 需要等待本次编程完成后 (通过轮询模式查询状态), AHB 总线再进行写操作去完成下一次指令周期的编程。 0: 不请求中止 1: 请求中止	0x0	RW
EN	[0]	使能 (Enable) 使能 QUADSPI。 0: 禁止 QUADSPI 1: 使能 QUADSPI	0x0	RW

Register Name: DCR, Address: 0x40003C04, Default: 0x001F0000

表196. 器件配置寄存器

Bit Name	Position	Description	Default	Access
FSIZE	[20:16]	Flash 大小 (Flash memory size)该字段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 4GB (使用 32 位进行寻址), 但在内存映射模式下的可寻址空间限制为 2GB。 仅可在 BUSY = 0 时修改该字段。	0x1F	RW
RESERVED	[15:11]	保留	0x0	RO
CSHT	[10:8]	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 时修改该字段。当 SSHIFT 为 1 时, nCS 高电平时间在上述设置基础上再加一个 CLK 周期。	0x0	RW
RESERVED	[7:1]	保留	0x0	RO
CKMODE	[0]	模式 0/ 模式 3 (Mode 0 / mode 3)	0x0	RW

		该位指示 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 时修改该字段。		
--	--	---	--	--

Register Name: SR, Address: 0x40003C08, Default: 0x00000000

表197. 状态寄存器

Bit Name	Position	Description	Default	Access
BUSY	[5]	忙 (Busy) 操作进行时, 该位置 1。在对 Flash 的操作完成时, 该位自动清零。	0x0	RO
RESERVED	[4]	保留	0x0	RO
SMF	[3]	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, 则该位置 1。向 CSMF 写入 1 可将该位清零。	0x0	RO
RESERVED	[2]	保留	0x0	RO
TCF	[1]	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, 该位置 1。向 CTCF 写入 1 时, 该位清零。	0x0	RO
TEF	[0]	传输错误标志 (Transfer error flag) 在间接模式下访问无效地址时, 该位置 1。向 CTEF 写入 1 可将该位清零。	0x0	RO

Register Name: FCR, Address: 0x40003C0C, Default: 0x00000000

表198. 标志清零寄存器

Bit Name	Position	Description	Default	Access
CSMF	[3]	清除状态匹配标志 (Clear status match flag) 写入 1 可将 QUADSPI_SR 寄存器中的 SMF 标志清零。	0x0	RW
RESERVED	[2]	保留	0x0	RO
CTCF	[1]	清除传输完成标志 (Clear transfer complete flag) 写入 1 可将 QUADSPI_SR 寄存器中的 TCF 标志清零。	0x0	RW
CTEF	[0]	清除传输错误标志 (Clear Transfer error flag) 写入 1 可将 QUADSPI_SR 寄存器中的 TEF 标志清零。	0x0	RW

Register Name: DLR, Address: 0x40003C10, Default: 0x00000000

表199. 数据长度(位宽)寄存器

Bit Name	Position	Description	Default	Access
DL	[1:0]	数据长度 (宽度) (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)。 2'b00: 传输 1 个字节 2'b01: 传输 2 个字节 2'b10: 传输 3 个字节 2'b11: 传输 4 个字节 ... 该字段在内存映射模式 (FMODE = 10) 下表示每次总线写传输写入的数据宽度, 程序应保证总线上写的数据宽度 hsize 和该字段的配置一致。 2'b00: 数据宽度 8 位	0x0	RW

		2'b01: 数据宽度 16 位 2'b10: 数据宽度 32 位 2'b11: 数据宽度 32 位 该字段在内存映射模式 (FMODE = 10) 下总线读操作时无效, 读出的数据宽度取决于总线 hsize 仅可在 BUSY = 0 时写入该字段。		
--	--	---	--	--

Register Name: CCR, Address: 0x40003C14, Default: 0x00000000

表200. 通信配置寄存器

Bit Name	Position	Description	Default	Access
FMODE	[27:26]	功能模式 (Functional mode) 该字段定义 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 仅可在 BUSY = 0 时写入该字段	0x0	RW
DMODE	[25:24]	数据模式 (Data mode) 该字段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 该字段还定义空指令阶段的操作模式。仅可在 BUSY = 0 时写入该字段。 内存映射模式下写外部 QSPI Flash 只能将该域配置为单线或者双线模式, 不支持四线模式	0x0	RW
RESERVED	[23]	保留	0x0	RW
DCYC	[22:18]	空指令周期数 (Number of dummy cycles) 该字段定义空指令阶段的持续时间。它指定 CLK 周期数 (0-31)。仅可在 BUSY = 0 时写入该字段。	0x0	RW
ABSIZE	[17:16]	交替字节长度 (Alternate bytes size) 该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 时写入该字段。	0x0	RW
ABMODE	[15:14]	交替字节模式 (Alternate bytes mode) 该字段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 时写入该字段。	0x0	RW
ADSIZE	[13:12]	地址长度 (Address size) 该位定义地址长度: 00: 8 位地址 01: 16 位地址 10: 24 位地址	0x0	RW

		11: 32 位地址 仅可在 BUSY = 0 时写入该字段。		
ADMODE	[11:10]	地址模式 (Address mode) 该字段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 时写入该字段。	0x0	RW
IMODE	[9:8]	指令模式 (Instruction mode) 该字段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 时写入该字段。	0x0	RW
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 时写入该字段。	0x0	RW

Register Name: AR, Address: 0x40003C18, Default: 0x00000000

表201. 地址寄存器

Bit Name	Position	Description	Default	Access
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, 将忽略写入该字段	0x0	RW

Register Name: ABR, Address: 0x40003C1C, Default: 0x00000000

表202. 交替字节寄存器

Bit Name	Position	Description	Default	Access
ALTERNATE	[31:0]	交替字节 (Alternate Bytes) 指定要在地址后立即发送到外部 SPI 设备的可选数据。 仅可在 BUSY = 0 时写入该字段。	0x0	RW

Register Name: DR, Address: 0x40003C20, Default: 0x00000000

表203. 交替字节寄存器

Bit Name	Position	Description	Default	Access
DATA	[31:0]	数据 (Data)指定要与外部 SPI 设备交换的数据。 在间接写入模式下, 写入该寄存器的数据在数据阶段发送到 Flash。 在间接模式下, 读取该寄存器可获得已从 Flash 接收的数据。 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。 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]。	0x0	RW

Register Name: PSMKR, Address: 0x40003C24, Default: 0x00000000

表204. 轮询状态屏蔽寄存器

Bit Name	Position	Description	Default	Access
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:	0x0	RW

		0: 屏蔽在自动轮询模式下所接收数据的位 n, 在匹配逻辑中不考虑其值 1: 不屏蔽在自动轮询模式下所接收数据的位 n, 在匹配逻辑中考虑其值 仅可在 BUSY = 0		
--	--	--	--	--

Register Name: PSMAR, Address: 0x40003C28, Default: 0x00000000

表205. 轮询状态匹配寄存器

Bit Name	Position	Description	Default	Access
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 时写入该字段	0x0	RW

Register Name: OPRUNR, Address: 0x40003C30, Default: 0x00000000

表206. 轮询状态匹配寄存器

Bit Name	Position	Description	Default	Access
OP_RUN	[0]	当配置好其他寄存器后, 给该位写 1 启动 qspi 传输, 当传输完成后该位自动清 0, 该位在内存映射模式下无效。	0x0	RW

23. 晶驱动器 (LCD)

23.1. LCD 主要特性

LCD 驱动器有 20 个 segment 输出 (SEG0~SEG19) 和 4 个 common 输出 (COM0~COM3)。

- 最大支持 20 SEG x 4 COM
- COM 端数量可选择 1,2,3,4
- 内部集成 ChargePump, 能在全芯片工作电压范围内保持 VLCD 稳定
- ChargePump 电压支持 2.4 - 3.6V 范围类 8 个可选电压值
- 支持 A,B 两种驱动波形
- 支持 VLCD 输出电阻大小选择

23.2. LCD 功能描述

23.2.1. LCD 控制模式

LCD 驱动器支持 4 种控制模式, 设置寄存器 LCDCON[23:20]位 (COMMUM 字段) 选择以下一种模式:

- Static (COM0)
- 1/2 duty (COM0, COM1)
- 1/3 duty (COM0, COM1, COM2)
- 1/4 duty (COM0, COM1, COM2, COM3)

23.2.2. LCD 时钟及刷新率

LCD 驱动器时钟源可以通过配置寄存器 RCC_BDCR[6:5]位 (DISPCLKSEL 字段) 和 LCDCON[16]来选择, 支持以下三种:

- 32MHz 系统时钟 (PCLK2)
- 32KHz 内部低速 RC (LSI)
- 32KHz 外接低速 XTAL (LSE)

由于 32K 时钟在休眠模式 DeepSleep1 和 DeepSleep2 下仍然运行, 当选择 LSI 或 LSE 时, 即使在休眠模式

下 LCD 仍能正常显示

LCD 驱动器帧率通过设置寄存器 LCDCON[12:0]位 (CLKCD 字段), 对时钟源进行分频获得。

如果 4 个 com 有效 (1/4 duty), 那么 8 个分频时钟形成一个帧率;

如果 3 个 com 有效 (1/3 duty), 那么 6 个分频时钟形成一个帧率;

如果 2 个 com 有效 (1/2 duty), 那么 4 个分频时钟形成一个帧率。

注: 当选择 PCLK2 时, 实际分频系数为 CLKCD 值向左移动 10 位 (即乘以 1024); 当选择 LSI 或者 LSE 时, 实际分频系数没有左移 10 位。如果 PCLK=32MHz, LSI=32KHz, 则切换 LCD 时钟时, CLKCD 可以保持不变。

23.2.3. LCD 偏置电压选择

LCD 驱动器支持 3 种偏置电压: V1、V2 及 V3, 支持两种模式选择: 1/3 bias、1/2 bias。可以通过配置寄存器 CPCON[8]位 (LEVEL 字段) 来选择偏置电压。当选择 1/2 bias 时, V2 和 V3 相等。偏置电压的产生电路采用内部电阻分压, 优点是可以节省 V2 与 V1 引脚的外部电容, 缺点是在打开 LCD 时分压电路上存在静态功耗, 静态功耗的大小与分压电阻的阻值有关, 分压电阻越大静态功耗越小, 但驱动能力越弱。可以通过配置寄存器 CPCON[5:4]位 (LCDREF 字段) 选择分压电阻阻值。

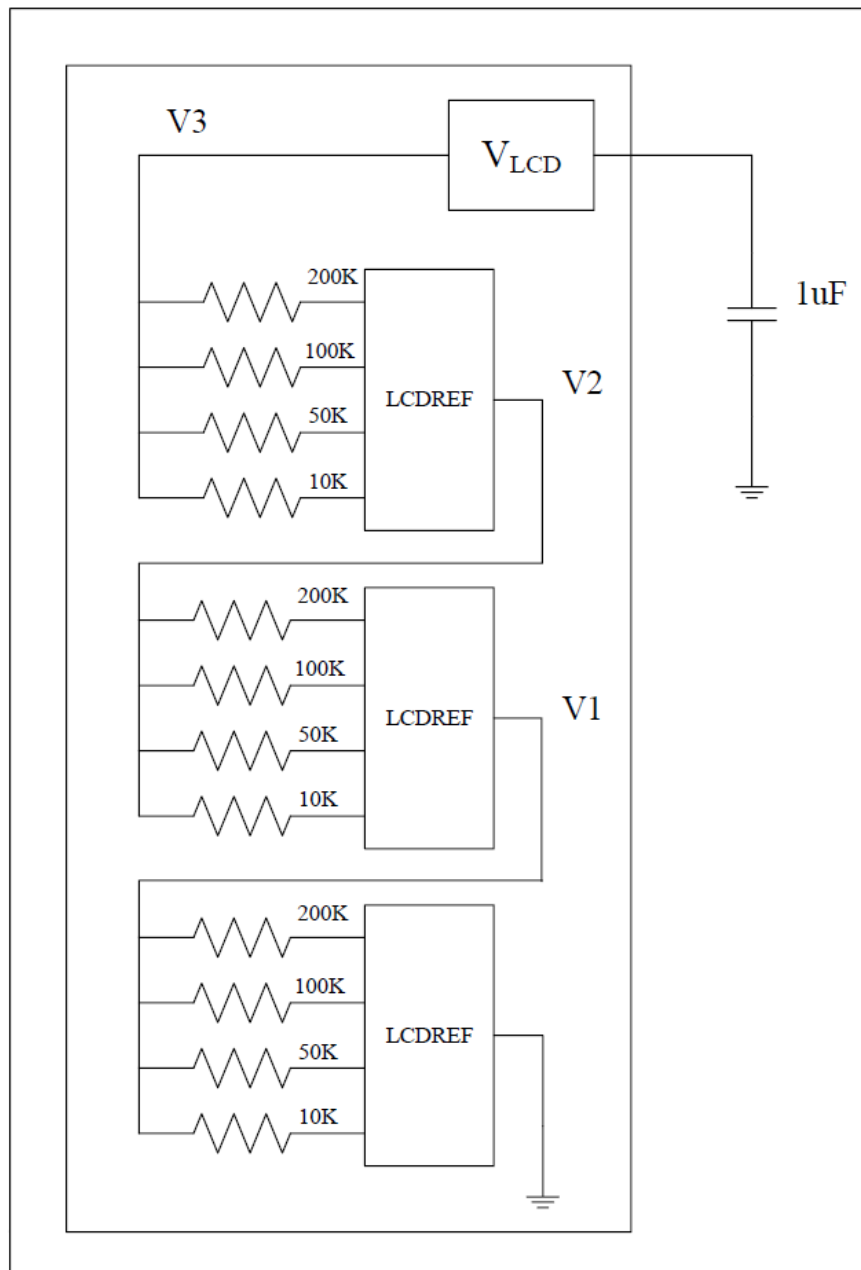


图145. LCD 偏置电压产生电路

LCD 电源可以采用内部的电荷泵来提供，也可以采用内部的电源来提供，此时需要关闭内部的电荷泵，寄存器 CPCON[0]位（PD 字段）打开或关闭电荷泵。当使用内部的电荷泵时，需要先将内部的基准源打开。

23.2.4. LCD 显示内容

SEGOFCOM0, SEGOFCOM1, SEGOFCOM2, SEGOFCOM3 分别用于配置 4 个 COM 上的 SEG 内容，每个寄存器都是 20 个 BIT 对应 20 个 SEG。当扫描到某一 COM 时会使用对应的 SEGOFCOMx 寄存器

23.2.5. LCD 驱动波形

LCD 驱动器输出分为 A 波形和 B 波形，A 波形每帧内正向和反向部分叠加使得 DC offset 为 0，B 波形每帧内正向和反向部分叠加使得 DC offset 为 0 同时相邻帧互为反向，进一步抵消掉 DC offset。通过配置寄存器 LCDCON[28]位（ABWAVE 字段）选择切换 A/B 波形的输出，默认为 A 波形输出。B 波形只适用于非 static 工作模式。下面仅仅以 1/4 duty 和 1/3 bias 波形为例。

图中 $FCLK = LCDCLK/CLKCD$ ，LCDCLK 根据配置可选择为 PCLK2, LSI 和 LSE。

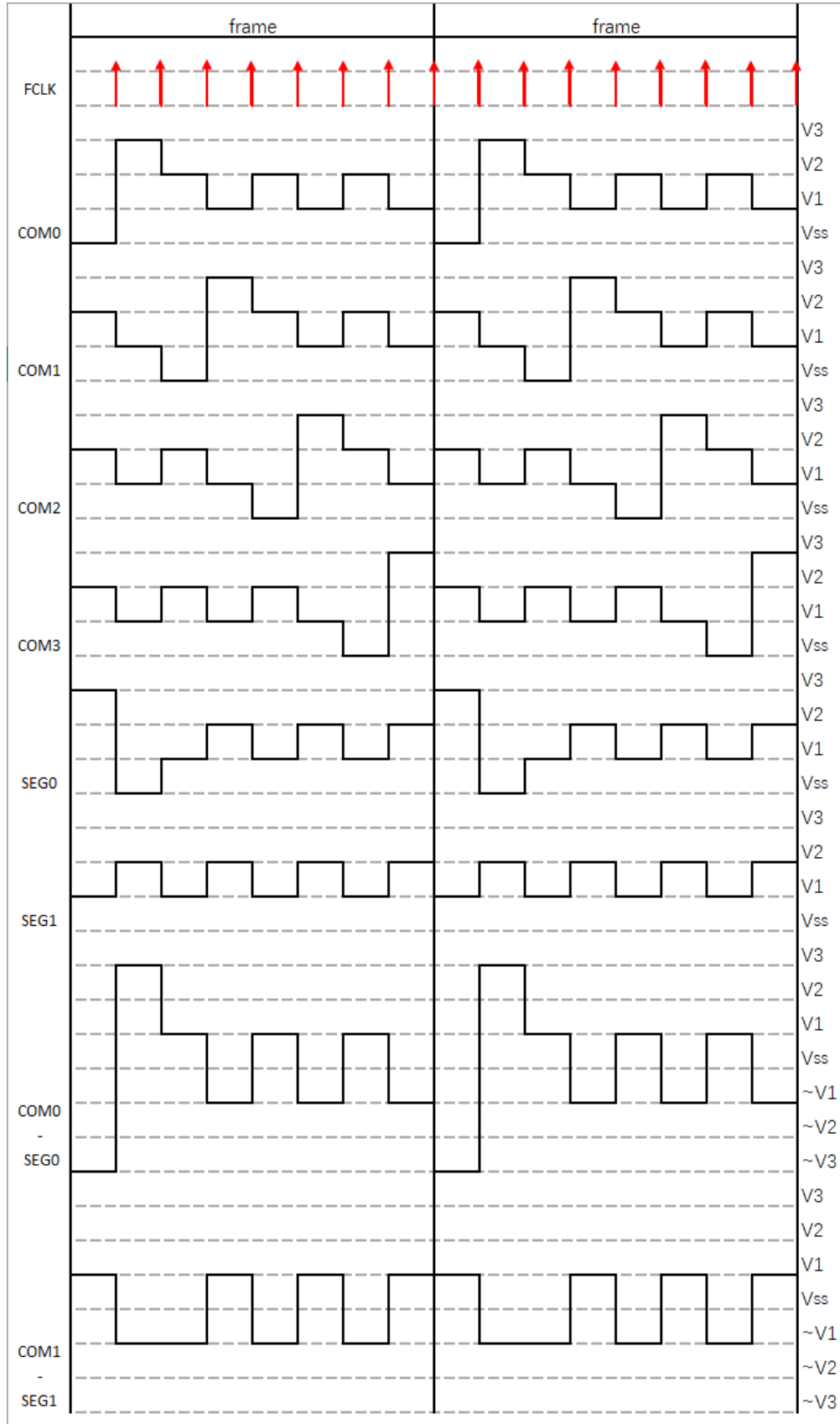


图146. A 波形

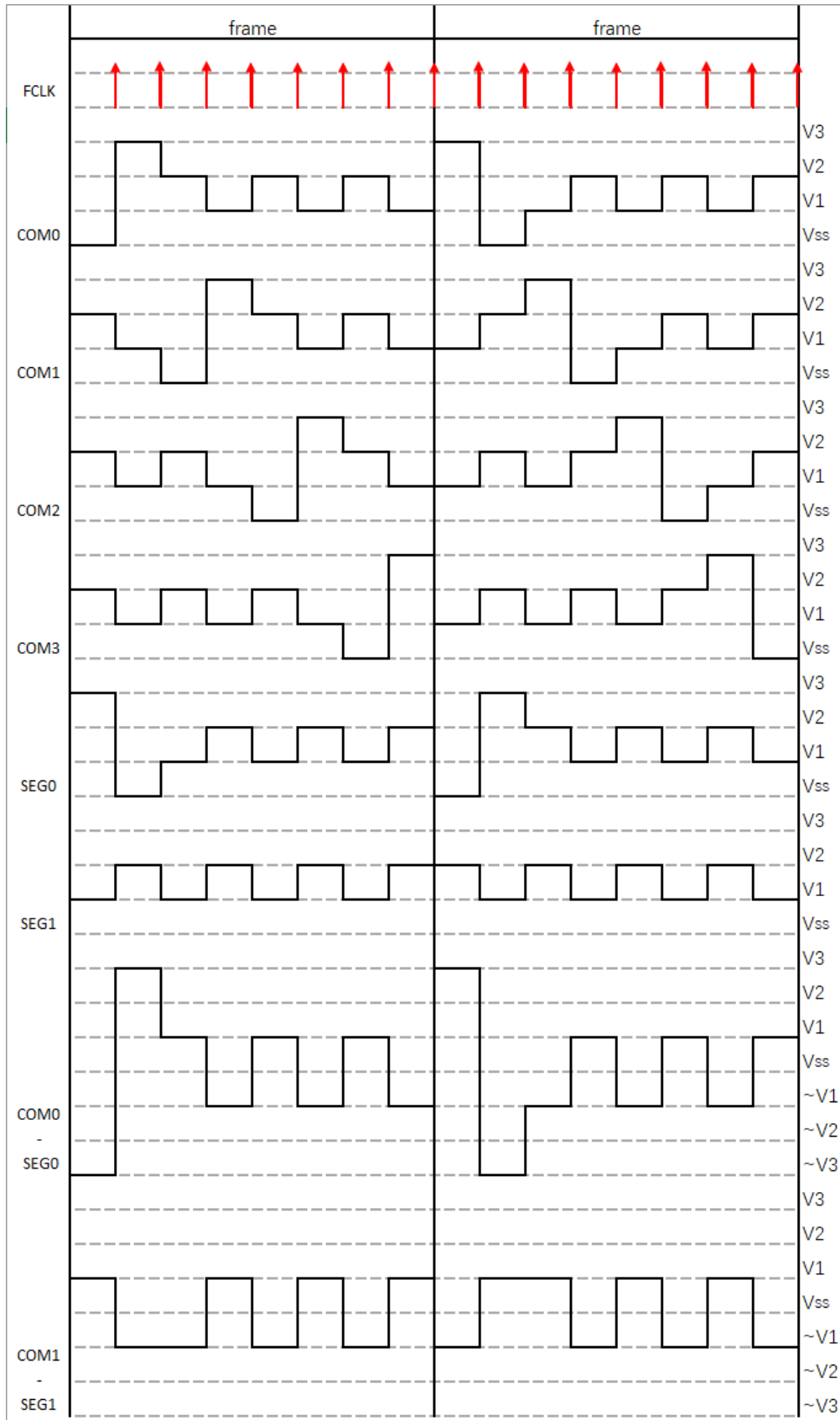


图147. B 波形

23.3. 电荷泵

LCD 电源可以采用内部的电荷泵来提供；也可以采用 VLCD 电压提供，此时需要关闭内部电荷泵。寄存器 CPCON[0] 位用于打开和关闭电荷泵。当使用内部电荷泵时，需要先将内部的基准源打开。

使用内部电荷泵可以在系统电源 VDD 工作于较低电压时，维持 VLCD 的电压较高且稳定。

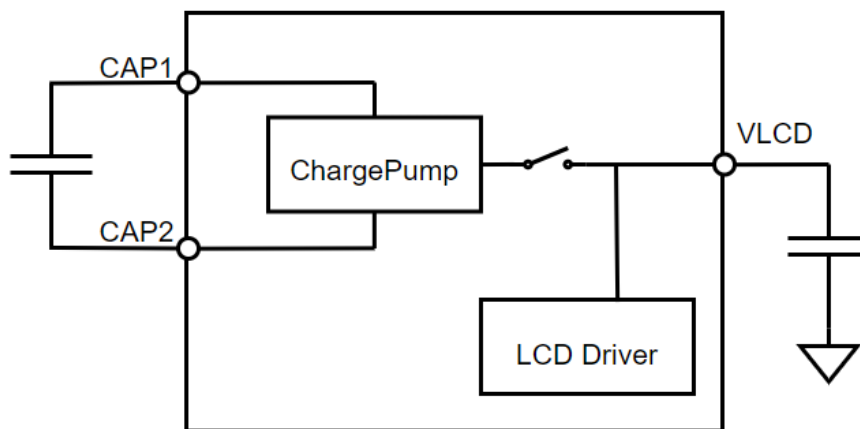


图148. 电荷泵连接

23.4. LCD 寄存器描述

Register Name: LCDCON, Address: 0x40011800, Default: 0x34000ff

表207.

Bit Name	Position	Description	Default	Access
CLKCD	[0:12]	Configure clock of display, if source clock is 32Mhz, CLKCD<<10 is used as clock dividing coefficient. output clock frequency is 32Khz/CLKCD or 32Khz/CLKCD<<10	0xff	RW
COMNUM	[20:22]	the number of COM, that is just duty cycle 1: COM0 2: COM0, COM1 3: COM0, COM1, COM2 4: COM1, COM2, COM3, COM4	0x4	RW
RESERVED	[24:25]		0x3	RO
ABWAVE	[28]	0: A 波形输出 1: B 波形输出	0x0	RW
LCDEN	[30]	0: LCD 驱动器关闭 1: LCD 驱动器开启	0x0	RW
CLKSEL	[16:17]	lcd clock 选择 0: PCLK 1: LSI, 内部 32K 低频时钟 2: LSE, 外部 32KHz 低频时钟	0x0	RW

Register Name: SEGOFCOM0, Address: 0x40011804, Default: 0x0

表208.

Bit Name	Position	Description	Default	Access
value	[0:19]	COM0 的 seg 段显示内容	0x0	RW

Register Name: SEGOFCOM1, Address: 0x40011808, Default: 0x0

表209.

Bit Name	Position	Description	Default	Access
value	[0:19]	COM1 的 seg 段显示内容	0x0	RW

Register Name: SEGOFCOM2, Address: 0x4001180c, Default: 0x0

表210.

Bit Name	Position	Description	Default	Access
value	[0:19]	COM2 的 seg 段显示内容	0x0	RW

Register Name: SEGOFCOM3, Address: 0x40011810, Default: 0x0

表211.

Bit Name	Position	Description	Default	Access
value	[0:19]	COM3 的 seg 段显示内容	0x0	RW

Register Name: CPCON, Address: 0x40011840, Default: 0x3

表212.

Bit Name	Position	Description	Default	Access
PD	[0]	Set "1" to power down LCD charge pump 0: Set "1" to power down LCD charge pump 1: power down charge pump	0x1	RW
BYPASS	[1]	Set "1" to bypass charge pump output VLCD to 3.3V digital power by a switch 0: Don't bypass charge pump 1: Bypass charge pump	0x1	RW
LCDOUT	[2:4]	Select charge pump output voltage 0: VLCD=3.6V 1: VLCD=3.3V 2: VLCD=3.1V 3: VLCD=2.9V 4: VLCD=2.8V 5: VLCD=2.6V 6: VLCD=2.5V 7: VLCD=2.4V	0x0	RW
LCDREF	[5:6]	0x0: 200k ohms 0x1: 100k ohms 0x2: 50k ohms 0x3: 10k ohms	0x0	RW
LEVEL	[8]	0x0: Choose 1/3 bias voltage 0x1: Choose 1/2 bias voltage	0x0	RW
RESERVED	[9]		0x0	RO
RESERVED	[12:13]		0x0	RO
VLCDOK	[16]	indicate if charge pump output voltage is ready for lcd/led to work 0: Charge pump output voltage is not high enough. Charge pump is not in regulation 1: Charge pump is in regulation	0x0	R(ReadOnly)
CPGD	[17]	indicate charge pump starts working or not 0: VLCD is too low, charge pump is still not working 1: charge pump is working.	0x0	R(ReadOnly)

Register Name: COMSEGEN, Address: 0x40011844, Default: 0x0

表213.

Bit Name	Position	Description	Default	Access
LCDSEGEN	[0:19]	开启 SEG 信号从引脚输出	0x0	RW
LCDCOMEN	[20:23]	开启 COM 信号从引脚输出	0x0	RW

Register Name: INTCON, Address: 0x40011848, Default: 0x0

表214.

Bit Name	Position	Description	Default	Access
VLCDOKINTEN	[0]	VLCD 电压 OK 中断使能	0x0	RW
CPGDINTEN	[1]	ChargePump 正常中断使能	0x0	RW
VLCDOKSTA	[2]	VLCD 电压 OK 状态	0x0	R(ReadOnly)
CPGDSTA	[3]	ChargePump 正常状态	0x0	R(ReadOnly)

24. 发光二极管驱动器 (LED)

24.1. LED 主要特性

LED 驱动器包含 8 个输出口 (LED0~LED7)，最多可以驱动 56 个发光二极管。

24.2. LED 功能描述

24.2.1. LED 控制模式

LED 驱动器支持 4 种控制模式，配置寄存器 LEDCON[25:24]位 (LEDNUM 字段) 可以选择，支持不同类型的 LED 矩阵。

- LED0~LED7 (8*7 矩阵)
- LED0~LED6 (7*6 矩阵)
- LED0~LED5 (6*5 矩阵)
- LED0~LED4 (5*4 矩阵)

24.2.2. LED 扫描周期

LED 驱动器时钟源为系统时钟分频得到的，可以通过配置寄存器 LEDCON[20:0]位 (CLKCD 字段) 设置分频系数。一个周期内扫描 8 个 LED 串，因此扫描周期 Tfram 是 8*Tledclk。

24.2.3. LED 亮灭控制

LED 矩阵连接方式如下图所示，如果要点亮 LED1，需要把 PAD_LED0 置低、PAD_LED1 置高，其余 IO 置为高阻输出。同样道理，要点亮其他 LED 时，把连接 LED 正极的 IO 置高，连接 LED 负极的 IO 置低，其余 IO 为高阻输出。在一个扫描周期内，最多只有一个 LED 会被点亮。

LED 驱动器寄存器 LED32[31:0]和 LED24[23:0]的每个位可以独立控制一个 LED 的亮灭，LED32 对应控制 LED1~LED32，LED24 对应控制 LED33~LED56。

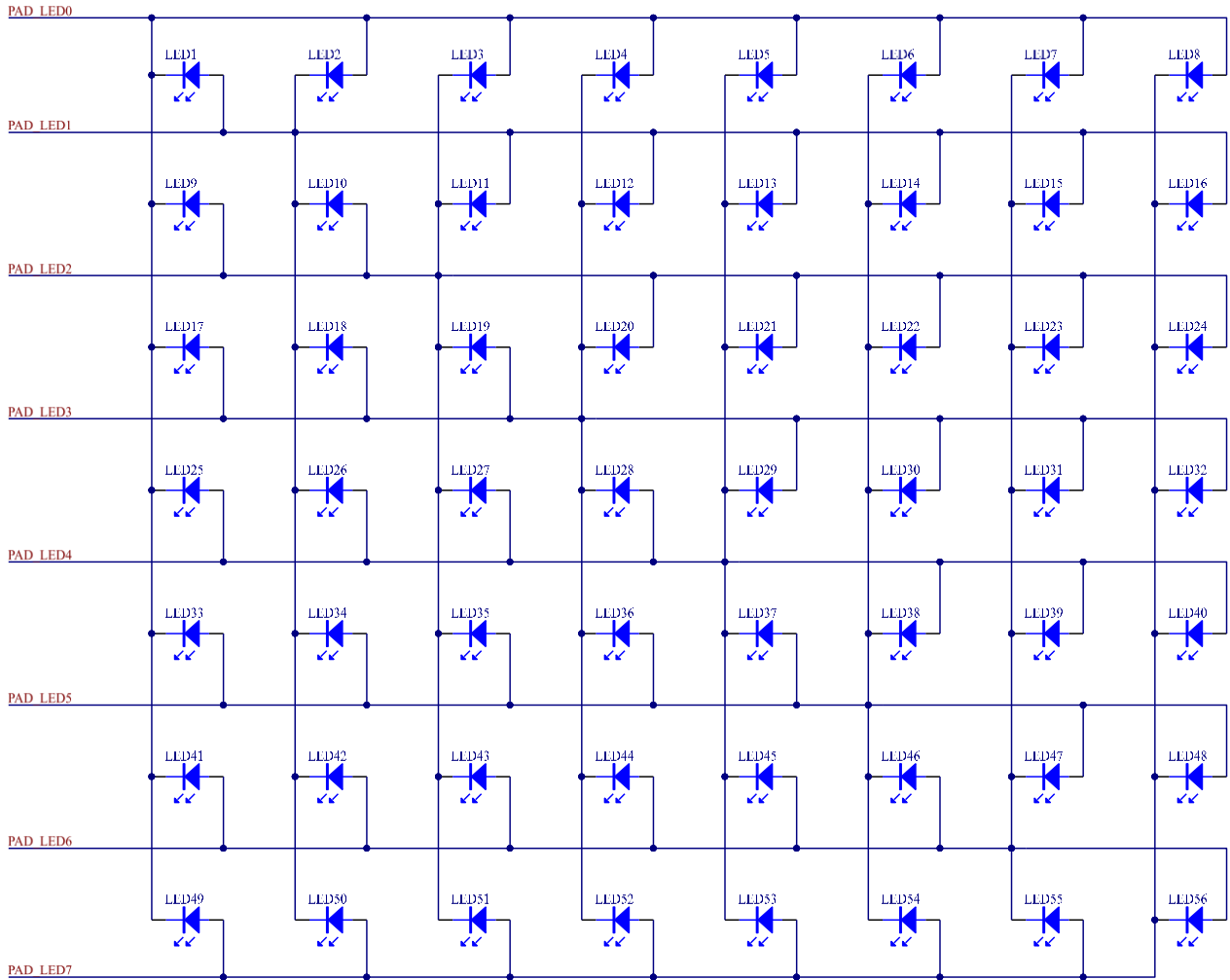


图149. LED 矩阵

24.2.4. LED 亮度控制

LED 的亮度可以通过两种方式控制：IO 的输出电流能力和 LED 点亮的时间占空比。LEDCURRENT 寄存器可以选择 8mA,10mA,12mA 以及 16mA 的 IO 输出能力。

LED 驱动器一个扫描周期内扫描 8 个 LED 灯串，一次可以点亮 1~7 个 LED，最长的亮灯时间是 $T_{frame}/8$ 。实际亮灯的时间由占空比调整，最小的占空比为 $1/256$ ，所以亮灯最小的时间为 $T_{frame}/(8*256)$ 。通过配置寄存器 LEDG0123DUTYCYCLE[31:0] 和 LEDG4567DUTYCYCLE[31:0] 来调整占空比，进而调整 LED 亮度。LEDG0123DUTYCYCLE[7:0] (G0 字段) 配置引脚 PAD_LED0 共阴的 LED 的占空比，其余字段以此类推。

24.3. LED 寄存器描述

Register Name: LEDCON, Address: 0x40011828, Default: 0xff

表215.

Bit Name	Position	Description	Default	Access
CLKCD	[0:20]	LED 时钟分频	0xff	RW
leden	[21]	0: LED 驱动器关闭 1: LED 驱动器开启	0x0	RW
LEDNUM	[24:25]	0x0: LED1~LED8 0x1: LED1~LED7 0x2: LED1~LED6	0x0	RW

Bit Name	Position	Description	Default	Access
		0x3: LED1~LED5		
ledrange	[28:29]	0x0: output 8mA 0x1: output 10mA 0x2: output 12mA 0x3: output 16mA	0x0	RW

Register Name: LED32, Address: 0x4001182c, Default: 0x0

表216.

Bit Name	Position	Description	Default	Access
onoff	[0:31]	LED1~32 开启或关闭	0x0	RW

Register Name: LED24, Address: 0x40011830, Default: 0x0

表217.

Bit Name	Position	Description	Default	Access
onoff	[0:23]	LED33~56 开启或关闭	0x0	RW

Register Name: LEDG0123DUTYCYCLE, Address: 0x40011834, Default: 0x0

表218.

Bit Name	Position	Description	Default	Access
G0	[0:7]	PAD_LED0 共阴 LED 占空比	0x0	RW
G1	[8:15]	PAD_LED1 共阴 LED 占空比	0x0	RW
G2	[16:23]	PAD_LED2 共阴 LED 占空比	0x0	RW
G3	[24:31]	PAD_LED3 共阴 LED 占空比	0x0	RW

Register Name: LEDG4567DUTYCYCLE, Address: 0x40011838, Default: 0x0

表219.

Bit Name	Position	Description	Default	Access
G4	[0:7]	PAD_LED4 共阴 LED 占空比	0x0	RW
G5	[8:15]	PAD_LED5 共阴 LED 占空比	0x0	RW
G6	[16:23]	PAD_LED6 共阴 LED 占空比	0x0	RW
G7	[24:31]	PAD_LED7 共阴 LED 占空比	0x0	RW

Register Name: LEDCURRENT, Address: 0x4001184c, Default: 0x30

表220.

Bit Name	Position	Description	Default	Access
CURRENT	[4:5]	LED IO 的电流控制 0: 输出 8mA 1: 输出 10mA 2: 输出 12mA 3: 输出 16mA	0x3	RW

25. 蜂鸣器 (BEEPER)

25.1. 蜂鸣器主要特性

蜂鸣器驱动器产生可编程频率的差分方波。它用于驱动两端口差分方波输入的外部压电声音元件。该模块由标准 APB 接口连接到系统总线。

蜂鸣器驱动器包含以下特征：

- 可输出 8kHz 至 0.25kHz 的频率
- 在不受系统时钟变化影响的固定独立 32 kHz (32,768 Hz) 时钟源上运行
- 播放持续时间从 4 ms 到 1.02 秒可编程，以 4 ms 为增量
- 提供脉冲（单音）和序列（多音）两种播放模式
- 在序列模式下，蜂鸣器可以编程为播放从 1 到 254 的任意数量的音调对（2 到 508 个音调），或者可以编程为无限期播放（直到被用户停止）
- 任何音调都可以编程为无限持续时间，连续播放直到被用户停止
- 中断可用于指示任何蜂鸣声的开始或结束、序列的结束或序列即将完成

25.2. 蜂鸣器功能描述

25.2.1. 蜂鸣器工作模式

蜂鸣器驱动器支持两种工作模式：脉冲模式序列模式。工作模式在蜂鸣器开启前配置，并且在禁用前无法更改。

在脉冲模式下，蜂鸣器只播放一个音。该模式通过在蜂鸣器配置寄存器 CTL[8:1]位 (SEQ_REPEAT 字段) 写 0x0 来配置。在配置寄存器 CTL[0]位 (BEEP_EN 字段) 写 0x1 开启后，蜂鸣器将按照寄存器 TONE_A 中的配置播放单音。此音播放完成后，蜂鸣器自动禁用，清除蜂鸣器状态寄存器 STAT[0]位 (BEEP_BUSY 字段)。

通过将 0x0 写入寄存器 TONE_A[14:7]位 (DUR 字段) 可以提前终止当前播放的音。或者，可以写 0x0 到 BEEP_EN 位禁用蜂鸣器。

在序列模式下，蜂鸣器可以播放可编程数量的音（对）。该模式通过在蜂鸣器配置寄存器 CTL[8:1]位 (SEQ_REPEAT 字段) 写非 0x0 值来配置。在 BEEP_EN 位写 0x1 开启后，蜂鸣器将按照寄存器 TONE_A 和 TONE_B 中的配置播放一串双音序列。重复次数与 SEQ_REPEAT 位的指示相同。0xFF 是一个特殊值，用于无限期播放序列（或直到被用户代码终止）。在所有迭代完成后，蜂鸣器自动禁用，清除 BEEP_BUSY 位。剩余序列迭代次数可从状态寄存器的 STAT[8:1]位 (SEQ_REMAIN 字段) 中读取。当蜂鸣器运行无限序列时，读取 SEQ_REMAIN 位始终返回 0xFF。当蜂鸣器在序列模式运行时，写入 SEQ_REPEAT 位会将迭代计数器从该值重启，并立即更新 SEQ_REMAIN 位。

通过将 0x0 写入 SEQ_REPEAT 位，序列模式可以提前终止。将 SEQ_REPEAT 设置为 0x0 会导致蜂鸣器在当前双音序列完成后终止播放并自行禁用。或者，可以通过禁用蜂鸣器立即终止所有播放（写 0x0 到 BEEP_EN 位）。

25.2.2. 蜂鸣器音调配置

蜂鸣器驱动器的播放音调频率和持续时间取决于音调寄存器 TONE_A 和 TONE_B 中的配置。每个寄存器配置了一个独立的音调。脉冲模式仅仅播放 TONE_A 的音调。

音调寄存器分为三个字段：频率 FREQ (TONE_x[7:0])，持续时间 DUR (TONE_x[14:8])，禁用 DISABLE (TONE_x[15])。该寄存器在任何时候都可以写入。在当前播放状态下写 0x0 到 DUR 位或置位/清零 DISABLE 位会立即生效，其他修改会在播放下一个音调时生效。因此，蜂鸣器驱动器允许下一个音调的数据在当前播放时写入。

频率 FREQ 用于编程音调相对于源时钟的相对频率 (32.768 kHz)。将值 0、1、2 和 3 写入频域都具有相同的效果：输出在回放期间不振荡（也称为静音音调）。4 到 127 (0x7F) 之间的任何值都可用于将源时钟向下分频为所需的音调频率。因此提供了从 8 kHz 到 ~0.25 kHz 的播放范围。

持续时间 DUR 用于编程选择播放时间。持续时间以 4ms 为单位步长。0 到 255 之间的任何值都可以存储在音调寄存器中。持续时间 0x0 会导致音调播放立即结束。255 (0xFF) 是一个特例值，为无限持续时间的音调编程，启动后音调持续播放，直到用户代码终止音调（将 0x0 写入 DUR）或禁用蜂鸣器（写 0x0 到 BEEP_EN 位）。

禁用 DISABLE 用于在播放过程中进一步控制蜂鸣器的输出引脚。当播放音调时置位禁用位，输出引脚的行为就像蜂鸣器被禁用一样，两个管脚都位于逻辑 0 处，它们之间没有直流电位，也没有振荡。当编程序列中存在长时间的静音时，可使用此功能，用于在这些静默期间防止压电元件损坏。

25.3. 蜂鸣器中断配置

蜂鸣器运行时可能发生六个跟踪事件：音调 A 可能开始或结束，音调 B 可能开始或结束，序列器可能结束或距离结束还有一步。这六个事件始终受到监控，当它们发生时，将在状态寄存器 STAT 中设置一个粘性位，供用户在将来某个时间读取。这些位会保持置位，直到用户代码清除为止。

这六个事件中的任何一个都可以用来生成中断。通过在寄存器 CTL 中设置相应的位来选择产生中断的事件。当事件触发中断时，用户代码必须清除事件位或在中断服务时禁用中断选择位。当服务蜂鸣器中断时，可能会有多个事件触发了中断（如果开启），用户代码必须检查并最终清除蜂鸣器状态寄存器中的所有事件位。将 0xFF 写入寄存器 STAT 将清除所有事件。所有跟踪事件都可独立选择是否产生中断。

25.4. 蜂鸣器时钟

蜂鸣器驱动器的频率和持续时间计时器基于 32.768 kHz 的输入时钟。时钟源在系统范围内配置，通过写入适当的时钟控制模块寄存器 (RCC_APB1ENR[2])，在外部晶体或内部振荡器之间进行选择。所选时钟仅为计时器提供稳定的参考，蜂鸣器的内部逻辑由系统的外围时钟 (PCLK) 计时。因此，当系统处于关闭外围时钟 (PCLK) 的低功率模式时，蜂鸣器无法运行。因此在进入外围时钟选通的低功耗状态之前，用户代码必须禁用蜂鸣器。如果系统在蜂鸣器播放音调时进入低功耗模式，则压电元件可能会损坏，因为压电元件端子之间存在恒定的长期直流电位。

25.5. 蜂鸣器寄存器描述

Register Name: CTL, Address: 0x40005000, Default: 0x0

表221. Beeper 控制寄存器

Bit Name	Position	Description	Default	Access
BEEP_EN	[0]	0: 禁用蜂鸣器 1: 开启蜂鸣器	0x0	RW
SEQ_REPEAT	[1:8]	0: 脉冲模式 1: 序列模式	0x0	RW
IRQ_TONEA_START	[9]	0: 音调 A 开始中断禁用 1: 音调 A 开始中断开启	0x0	RW
IRQ_TONEA_END	[10]	0: 音调 A 结束中断禁用 1: 音调 A 结束中断开启	0x0	RW
IRQ_TONEB_START	[11]	0: 音调 B 开始中断禁用 1: 音调 B 开始中断开启	0x0	RW
IRQ_TONEB_END	[12]	0: 音调 B 结束中断禁用 1: 音调 B 结束中断开启	0x0	RW
IRQ_SEQ_NEAR_END	[13]	0: 序列结束还剩一个音对中断禁用 1: 序列结束还剩一个音对中断开启	0x0	RW
IRQ_SEQ_END	[14]	0: 序列结束中断禁用	0x0	RW

Bit Name	Position	Description	Default	Access
		1: 序列结束中断开启		

Register Name: STAT, Address: 0x40005004, Default: 0x0

表222. Beeper 状态寄存器

Bit Name	Position	Description	Default	Access
BEEP_BUSY	[0]	0: 蜂鸣器空闲 1: 蜂鸣器占用	0x0	R (ReadOnly)
SEQ_REMAIN	[1:8]	序列剩余音对数量	0x0	R (ReadOnly)
IRQ_TONEA_START	[9]	0: 音调 A 开始未产生中断 1: 音调 A 开始产生中断	0x0	RW
IRQ_TONEA_END	[10]	0: 音调 A 结束未产生中断 1: 音调 A 结束产生中断	0x0	RW
IRQ_TONEB_START	[11]	0: 音调 B 开始未产生中断 1: 音调 B 开始产生中断	0x0	RW
IRQ_TONEB_END	[12]	0: 音调 B 结束未产生中断 1: 音调 B 结束产生中断	0x0	RW
IRQ_SEQ_NEAR_END	[13]	0: 序列结束还剩一个音对未产生中断 1: 序列结束还剩一个音对产生中断	0x0	RW
IRQ_SEQ_END	[14]	0: 序列结束未产生中断 1: 序列结束产生中断	0x0	RW

Register Name: TONE_A, Address: 0x40005008, Default: 0x0

表223. Beeper 音调 A 寄存器

Bit Name	Position	Description	Default	Access
FREQ	[0:6]	音调 A 时钟分频	0x0	RW
DUR	[7:14]	音调 A 持续时间	0x0	RW
DISABLE	[15]	音调 A 输出引脚禁用 0: 音调 A 输出引脚未禁用 1: 音调 A 输出引脚禁用	0x0	RW

Register Name: TONE_B, Address: 0x4000500c, Default: 0x0

表224. Beeper 音调 B 寄存器

Bit Name	Position	Description	Default	Access
FREQ	[0:6]	音调 B 时钟分频	0x0	RW
DUR	[7:14]	音调 B 持续时间	0x0	RW
DISABLE	[15]	音调 B 输出引脚禁用 0: 音调 B 输出引脚未禁用 1: 音调 B 输出引脚禁用	0x0	RW

26. 可编程电压监测器 (LVD)

26.1. LVD 主要特征

LVD 包含以下特征:

- 正负输入端可以编程选择
- 负输入端输入电压可以编程
- 输出极性可以编程

26.2. LVD 功能描述

用户配置寄存器 LVDCON，可以使能 LVD 功能，LVD 正负输入通道选择等功能，具体详见寄存器描述

26.3. LVD 寄存器描述

Register Name: LVDCON, Address: 0x40007700, Default: 0x180

表225. 低压检测模块及模拟比较器控制

Bit Name	Position	Description	Default	Access
PD	[0]	LVD/比较器使能控制 0: 使能 1: 不使能	0x0	RW
INPAIR	[1]	比较器输入范围选择 0: 输入范围选择 AVDD~0.7V 1: 输入范围选择 AVDD~0	0x0	RW
REFSEL	[2: 6]	电压比较器检测电压的选择 0~31, 0 对应 1V, 31 对应 1.79V	0x0	RW
SELP	[7:8]	LVD 或者模拟比较器的正输入选择 0x0:空 0x1:AIN4 0x2:AIN7 0x3:内部 AVDD/2	0x3	RW
SELN	[9]	LVD 或者模拟比较器的负输入选择 0x0: 内部参考 0x1: AIN3	0x0	RW
Hysts_sel	[16:17]	比较器功能下, 可以选择不同的迟滞电压,REF+SEL<31,REF-SEL>0 0x0: 没有迟滞 0x1: 20mV 迟滞 (REDSEL+1 or -1) 0x2: 40mV 迟滞 (REDSEL+2 or -2) 0x3: 60mV 迟滞 (REDSEL+3 or -3)	0x0	RW
Comp_polarity	[18]	比较器极性选择 0x0: INP>INN, 输出为高 0x1: INP>INN, 输出为低	0x0	RW
INTEN	[21]		0x0	RW
INTSTA	[22]			

Register Name: CSR, Address: 0x40007004, Default: 0x0

表226. 电源控制/状态寄存器

Bit Name	Position	Description	Default	Access
LVDO	[2]	低电压检测 LVD 输出状态 当 LVD 被 AFE_MMR 模块中的 LVDCON_EN 位使能后该位才有效 0x0: VDD/VDDA 高于 由 AFE_MMR 模块中的 LVDCON_REFSEL 选定的 LVD 阈值 0x1: VDD/VDDA 低于 由 AFE_MMR 模块中的 LVDCON_REFSEL 选定的 LVD 阈值	0x0	R

27. 电气特性

27.1. 绝对最大额定值

表227. 绝对最大电压

参数	描述	最小值	典型值	最大值	单位
AVDD	AVDD 和 AGND 之间的电压	-0.3	-	3.9	V
DVDD	DVDD 和 DGND 之间的电压	-0.3	-	3.9	V
AGND	AGND 和 IOGND 之间的电压	-0.3	-	0.3	V
DGND	DGND 和 IOGND 之间的电压	-0.3	-	0.3	V
VCAP	VREFP 和 VREFN 之间的电压	-0.3	-	3.9	V
VIO	IO 管脚电压	-0.3	-	3.9	V
Tstorage	存储温度	-65	-	150	°C
Tjunction	结温	-40	-	125	°C
Isink	流入任一 IO 管脚的电流	-	-	25	mA
Isource	流出任一 IO 管脚的电流	-25	-	-	mA
Isink_tot	流入所有 IO 管脚的总电流			80	mA
Isource_tot	流出所有 IO 管脚的总电流	-80			mA

27.2. 正常工作条件

表228. 正常工作电压

参数	描述	最小值	典型值	最大值	单位
AVDD	AVDD 的工作电压范围	2.2	-	3.6	V
DVDD	DVDD 的工作电压范围	2.2	-	3.6	V
Ta	芯片工作温度	-40	-	105	°C
Tj	芯片结温	-40	-	125	°C
Fclk	MCU 时钟频率	0	-	32	MHz
VIO	IO 输入电压范围	-0.3	-	3.6	V
TVDD-POR	DVDD 上电复位阈值	2			V
TVDD-fall	DVDD 掉电复位阈值	1.8			V
TDVDD_REG_fall	DVDD_REG 掉电复位阈值	1.62			V
PSM	DVDD 掉电监测阈值	2.2		3.3	V
Treset	外部复位最小脉冲宽度	1.5			μs
Tstart-up	上电启动时间		40		ms

27.3. I/O 端口特性

表229. IO 端口特性

参数	描述	最小值	典型值	最大值	单位
DVDD	IO 电源	2.2	3.3	3.6	V
V _{IH}	输入高电平	2.2		DVDD+0.3	V
V _{IL}	输入低电平	-0.3		0.8V	V
V _{OL}	输出低电平	0.4			V
V _{OH}	输出高电平	2			V
I _L	输入漏电流			±10	μA
I _O	输出电流			30	mA

27.4. Flash 特性

表230. Flash 特性

参数	描述	最小值	典型值	最大值	单位
NE	擦写次数	100K			Cycles
RY	保存年限	10			Years

27.5. RC 振荡特性

表231. 内部振荡器特性

参数	描述	最小值	典型值	最大值	单位
F32M	内部 32MHz 振荡器频率		32		MHz
Faccuray_HRC	32MHz 振荡器精度		1		%
F32k	内部 32kHz 振荡器频率		32		kHz
Faccuray_LRC	32kHz 振荡器频率精度		±3		%

27.6. 晶振特性

下图给出了晶振的晶体和负载及寄生电容等外部条件。

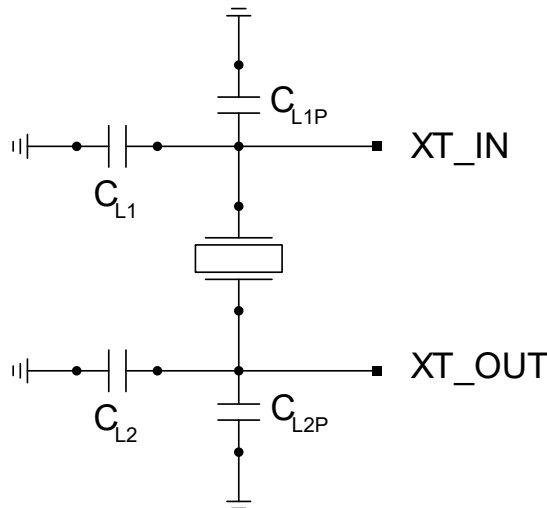


图150. 32kHz 晶振

C_{L1} 和 C_{L2} PCB 版上的电容器件, C_{L1P} 和 C_{L2P} 是 PCB 板和封装的寄生电容。

CL 是晶振的负载电容. 特定的晶体需要接特定范围的负载电容。

$$C'_{L1} = C_{L1} + C_{L1P}$$

$$C'_{L2} = C_{L2} + C_{L2P}$$

$$CL = C'_{L1} * C'_{L2} / (C'_{L1} + C'_{L2})$$

表232. 晶振特性

参数	描述	最小值	典型值	最大值	单位
FXTAL	晶振频率	32			kHz
CL	负载电容		20		pF

27.7. ADC 特性

表233. ADC 特性

参数	描述	最小值	典型值	最大值	单位
VDDA	ADC 正常工作电压范围	2.2	-	3.6	V
FADC	ADC 转换频率		-	8	kSPS
Resolution	ADC 分辨率	24			Bits
ENOB	有效精度@PGA=128, 30Hz		19.5		Bits
	有效精度@PGA=128, 8kHz		15.4		Bits
Input Noise	输入噪声参见 table				
INL	ADC 积分非线性@PGA=128		±10		ppm
FS Error	ADC 满量程误差@PGA=128		±1		%
FS Error Drift	ADC 满量程误差漂移（不包括参考电压温漂）@PGA=128		2		ppm/°C
Offset	ADC 失调误差@PGA=128		2		μV
Offset Drift	ADC 失调电压温漂@PGA=128		1		nV/°C
Vrange	ADC 输入范围(单端)	0	-	VREFP/PGA	V
	ADC 输入范围（差分）	-VREFP/PGA	-	VREFP/PGA	V
Rin	差分输入阻抗		1		GΩ
Pmon	AVDD, DVDD, IOVDD 检测误差		1		%
VREFP(VS)	ADC 参考电压, 外接 100nF		1.8		V
			2.35		V
			2.45		V
			2.8		V
	VREFP 温漂		30		ppm/°C
	最大负载电流		4	8	mA

27.8. 比较器特性

表234. 比较器特性

参数	描述	最小值	典型值	最大值	单位
VDDcomp	比较器正常工作电压范围	2.2	-	3.6	V
Voffset	输入失调电压		±10		mV
Vrange	输入范围	0		AVDD-0.8	V
Vhys	可调迟滞电压范围	10		50	mV
Tresponse	比较器响应时间		10		μS

27.9. 温度传感器特性

表235. 温度传感器特性

参数	描述	最小值	典型值	最大值	单位
V_25	25°C 电压输出		1.16		V
TC	电压系数		4		mV/°C
Accuracy	校正前温度精度		±3		°C

27.10. LCD

表236. LCD 输出特性

参数	描述	最小值	典型值	最大值	单位
VLCD	Charge pump 输出	2.45		3.6	V
R_unit	内部分压电阻串单位电阻值	10		200	kOhm
dVout	COM<0:3>和 SEG<0:19>输出电压变化	-100		+100	mV

27.11. 恒流源

表237. 恒流源输出特性

参数	描述	最小值	典型值	最大值	单位
Iout	恒流源驱动输出电流	8		20	mA
Vcompliance	顺从电压			0.5	V
Accuracy	电流精度	-10%		+10%	

27.12. 输出参考电压

表238. 参考源特性

参数	描述	最小值	典型值	最大值	单位
VREF	外部最大电容 100nF		1.22		V
Accuracy	VCM			±5	mV
Drift	VCM 温漂		30		ppm/°C
Iload	最大负载电流			1	mA

27.13. 功耗

表239. 功耗

AVDD=3.3,DVDD=3.3										
MODE	LCD	ADC	UART	I ² C	PWM	HCLK	PCLK1	PCLK2	DVDD	AVDD
Normal	Disable	Disable	Disable	Disable	Disable	32M	32M	32M	6.05mA	0.9uA
	Disable	Disable	Disable	Disable	Disable	16M	16M	16M	3.94mA	0.9uA
	Disable	Disable	Disable	Disable	Disable	8M	8M	8M	2.70mA	0.9uA
	Disable	Disable	Disable	Disable	Disable	4M	4M	4M	1.91mA	0.9uA
	Disable	Disable	Enable	Disable	Disable	4M	4M	4M	1.99mA	0.9uA
	Disable	Disable	Disable	Enable	Disable	4M	4M	4M	2.00mA	0.9uA
	Disable	Disable	Disable	Disable	f=4K Duty =50%,	4M	4M	4M	2.11mA	0.9uA
	Disable	Convert	Disable	Disable	Disable	32M	32M	32M	6.61mA	750uA
	Disable	Convert	Disable	Disable	Disable	16M	16M	16M	4.28mA	750uA
	Disable	Convert	Disable	Disable	Disable	8M	8M	8M	2.96mA	750uA
	全显	Disable	Disable	Disable	Disable	32M	32M	32M	6.27mA	0.9uA
	全显	Disable	Disable	Disable	Disable	4M	4M	4M	2.03mA	0.9uA
DeepSleep1	-	-	-	-	-	-	-	-	224uA	0uA
DeepSleep2	-	-	-	-	-	-	-	-	4.4uA	0uA

AVDD=3.3,DVDD=3.3										
MODE	LCD	ADC	UART	I ² C	PWM	HCLK	PCLK1	PCLK2	DVDD	AVDD
DeepSleep1	全显	-	-	-	-	-	-	-	232uA	0uA
DeepSleep2	全显	-	-	-	-	-	-	-	30uA	0uA

功耗测量注意事项:

1. 每个模块有对应的时钟使能，关闭不使用的时钟有利于降低功耗。
2. P0.0 – P0.7 以及 P1.0, P1.1, P1.2 默认使能输入，这些引脚悬空会导致功耗增加，可以通过使能内部上拉或下拉，或者接外部上下拉电阻的方式保持确定电平，具体操作可参考 Power 例程。

27.14. ESD 特性

表240. 参考源特性

参数	描述	值	单位
V ESD-HBM	ESD 放电人体模型，基于 AEC-Q100-002	±2000	V
V ESD-CDM	ESD 器件放电模型，基于 AEC-Q100-011	±500	V
ILatch-up	AEC-Q100-004	±100	mA

28. 噪声特性

测试条件：常温下 AVDD=DVDD=3.3V, AIN0 与 AIN1 短路到 AGND,使用 AVDD 做 ADC 参考电压。

表241. 均方根噪声(uV)

Data Rate(SPS)	PGA_ GAIN_1	PGA_ GAIN_2	PGA_ GAIN_4	PGA_ GAIN_8	PGA_ GAIN_16	PGA_ GAIN_32	PGA_ GAIN_64	PGA_ GAIN_128
8	0.221	0.132	0.087	0.044	0.041	0.031	0.034	0.031
16	0.328	0.198	0.108	0.069	0.053	0.048	0.050	0.042
32	0.453	0.257	0.142	0.089	0.078	0.066	0.073	0.068
64	0.633	0.364	0.223	0.135	0.111	0.097	0.097	0.087
125	0.925	0.503	0.293	0.190	0.159	0.144	0.136	0.131
250	1.325	0.702	0.420	0.263	0.223	0.199	0.189	0.181
500	1.822	0.980	0.588	0.373	0.312	0.276	0.277	0.255
1000	2.595	1.405	0.826	0.530	0.430	0.404	0.391	0.368
2000	3.743	1.999	1.172	0.764	0.623	0.574	0.566	0.526
4000	5.773	3.074	1.781	1.135	0.922	0.860	0.824	0.779
8000	14.908	7.611	4.049	2.275	1.588	1.345	1.251	1.178

表242. 峰峰值噪声(uV)

Data Rate(SPS)	PGA_ GAIN_1	PGA_ GAIN_2	PGA_ GAIN_4	PGA_ GAIN_8	PGA_ GAIN_16	PGA_ GAIN_32	PGA_ GAIN_64	PGA_ GAIN_128
8	0.787	0.590	0.492	0.197	0.197	0.160	0.166	0.175
16	1.574	0.787	0.492	0.344	0.270	0.234	0.246	0.243
32	3.147	1.574	0.885	0.541	0.492	0.369	0.461	0.387
64	3.934	2.164	1.574	0.787	0.615	0.639	0.602	0.522
125	5.901	3.344	2.164	1.279	1.082	0.873	1.076	0.882
250	9.048	4.917	2.950	1.967	1.549	1.365	1.260	1.220
500	13.375	7.868	4.131	2.557	2.754	1.992	1.893	1.955

Data Rate(SPS)	PGA_ GAIN_1	PGA_ GAIN_2	PGA_ GAIN_4	PGA_ GAIN_8	PGA_ GAIN_16	PGA_ GAIN_32	PGA_ GAIN_64	PGA_ GAIN_128
1000	19.670	10.425	6.589	3.885	3.221	2.987	3.153	2.665
2000	29.111	15.932	9.540	5.950	5.827	4.610	4.112	4.032
4000	48.780	24.390	14.064	9.491	8.261	7.696	6.405	6.190
8000	282.061	68.057	35.012	20.604	12.785	11.716	10.855	9.617

表243. 无噪声位数 NFB(bit)

Data Rate(SPS)	PGA_ GAIN_1	PGA_ GAIN_2	PGA_ GAIN_4	PGA_ GAIN_8	PGA_ GAIN_16	PGA_ GAIN_32	PGA_ GAIN_64	PGA_ GAIN_128
8	23.000	22.415	21.678	22.000	21.000	20.300	19.245	18.167
16	22.000	22.000	21.678	21.193	20.541	19.752	18.678	17.696
32	21.000	21.000	20.830	20.541	19.678	19.093	17.771	17.023
64	20.678	20.541	20.000	20.000	19.356	18.300	17.385	16.591
125	20.093	19.913	19.541	19.300	18.541	17.850	16.549	15.835
250	19.476	19.356	19.093	18.678	18.023	17.206	16.321	15.367
500	18.913	18.678	18.608	18.300	17.193	16.660	15.733	14.687
1000	18.356	18.272	17.934	17.696	16.967	16.075	14.997	14.240
2000	17.791	17.660	17.400	17.081	16.111	15.449	14.614	13.642
4000	17.046	17.046	16.840	16.408	15.608	14.710	13.975	13.024
8000	14.514	15.565	15.524	15.289	14.978	14.104	13.214	12.389

表244. 有效位数 ENOB(bit)

Data Rate(SPS)	PGA_ GAIN_1	PGA_ GAIN_2	PGA_ GAIN_4	PGA_ GAIN_8	PGA_ GAIN_16	PGA_ GAIN_32	PGA_ GAIN_64	PGA_ GAIN_128
8	24.833	24.578	24.184	24.162	23.278	22.649	21.528	20.649
16	24.264	23.988	23.870	23.511	22.895	22.029	20.984	20.220
32	23.797	23.617	23.475	23.151	22.332	21.583	20.435	19.542
64	23.314	23.111	22.819	22.547	21.821	21.024	20.017	19.171
125	22.767	22.645	22.424	22.048	21.307	20.447	19.536	18.588
250	22.249	22.165	21.905	21.583	20.817	19.987	19.058	18.117
500	21.789	21.683	21.421	21.078	20.335	19.509	18.508	17.625
1000	21.279	21.164	20.930	20.570	19.870	18.962	18.009	17.095
2000	20.750	20.655	20.425	20.042	19.337	18.454	17.476	16.581
4000	20.125	20.034	19.821	19.471	18.770	17.872	16.933	16.015
8000	18.756	18.726	18.637	18.468	17.986	17.226	16.331	15.417

29. 封装信息

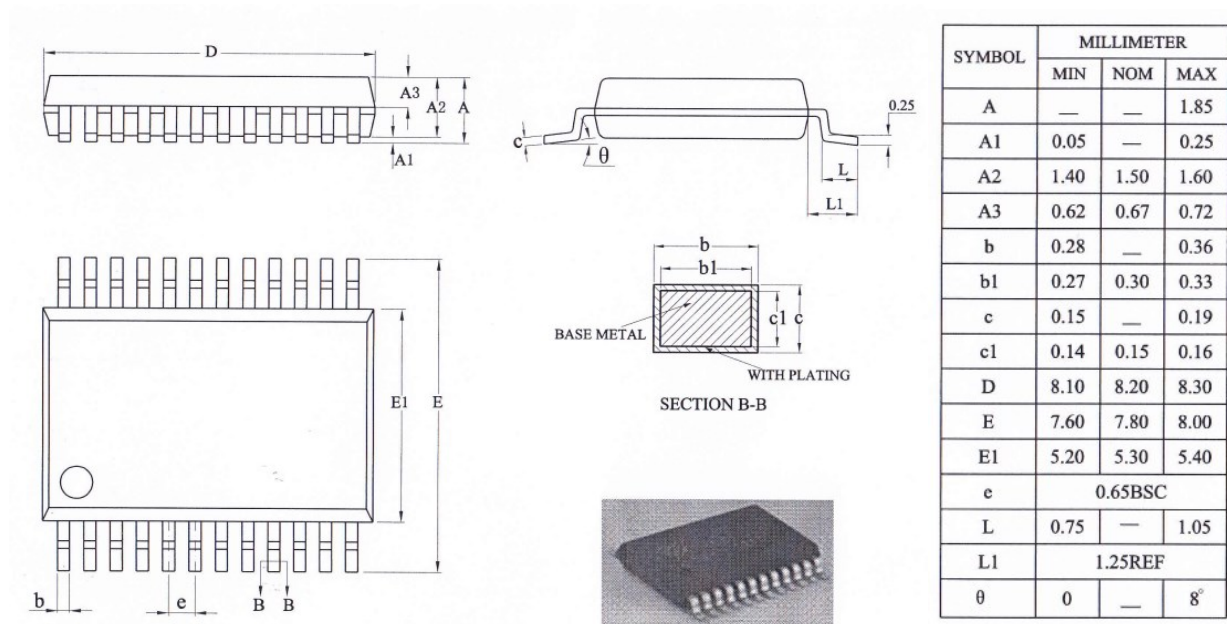


图151. SSOP24 封装尺寸

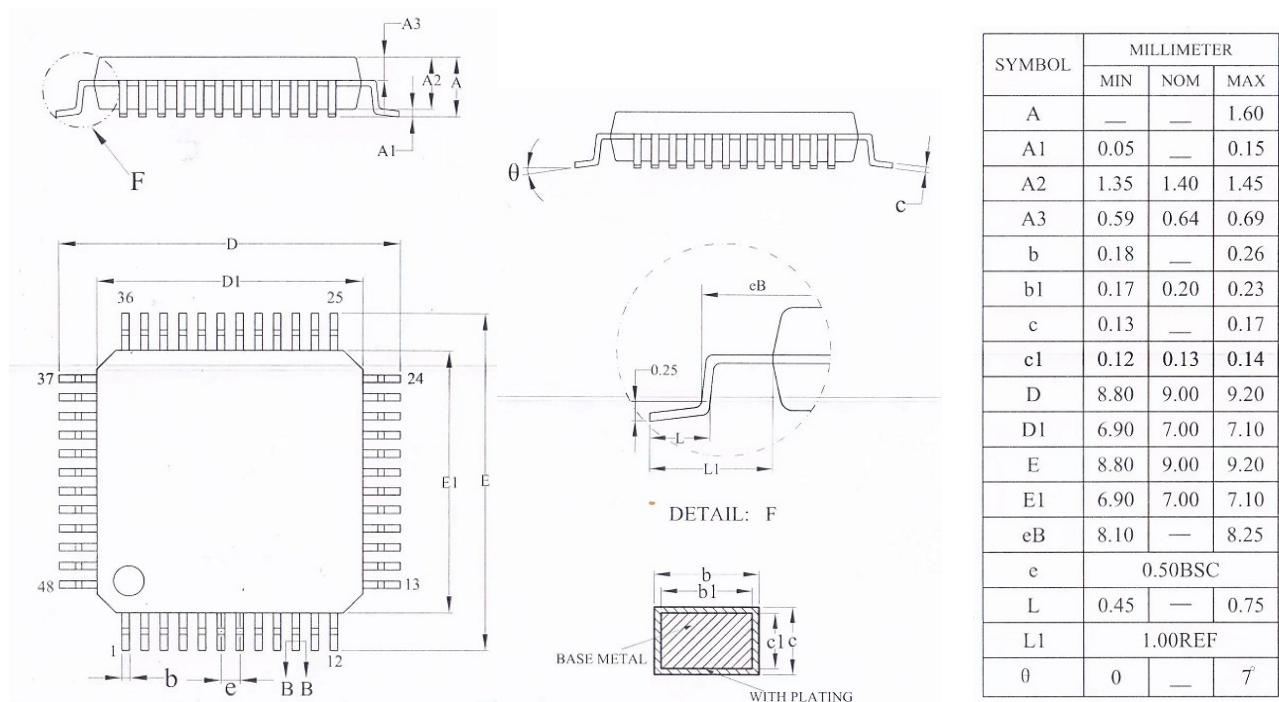


图152. LQFP48 封装尺寸

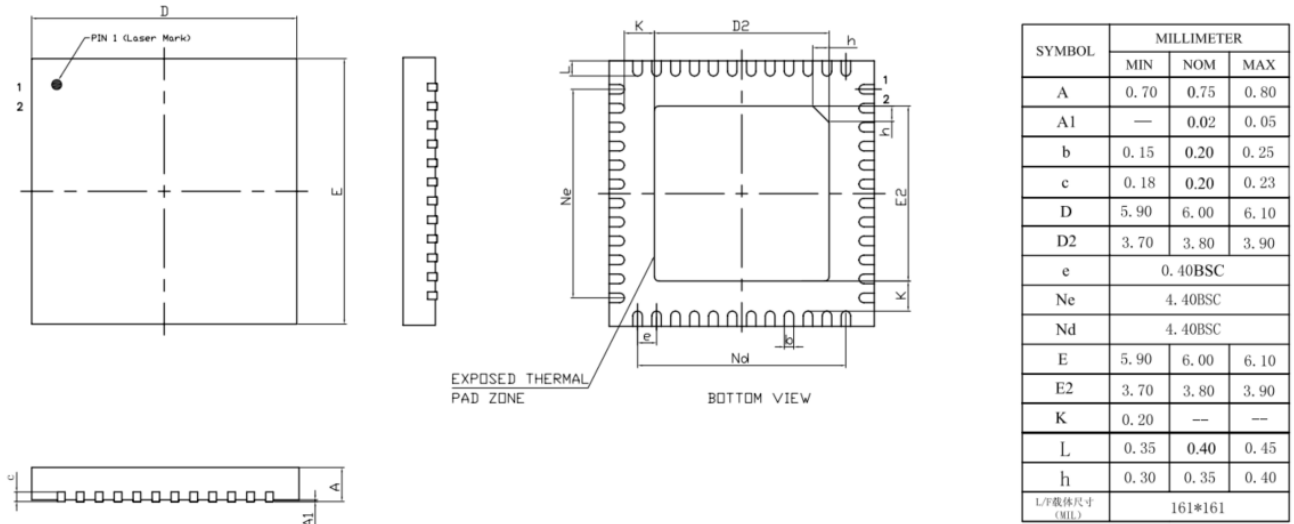


图153. QFN48 封装尺寸

30. 订购信息

表245. 订购信息

芯片编号	封装类型	引脚数	温度范围
LH32M0S32BQME	QFN	48	-40°C ~ 85°C
LH32M0S32BLMG	LQFP	48	-40°C ~ 105°C
LH32M0S32BSGG	SSOP	24	-40°C ~ 105°C

表246. 包装规格

订购型号	封装类型	引脚数	包装类型	大包装量	备注
LH32M0S32BQME	QFN	48	REEL	3000 颗/卷	
LH32M0S32BLMG	LQFP	48	REEL	3000 颗/卷	
LH32M0S32BSGG	SSOP	24	TUBE	600 颗/包	

注释: REEL: 卷带包装;
TRAY: 托盘包装;
TUBE: 管式包装;