

MS8006

ARM[®] Cortex[®]-M0+

32 位微控制器

数据手册

本文所包含的信息是合肥宏晶微电子科技股份有限公司的专有财产，在没有合肥宏晶微电子科技股份有限公司许可的情况下，不允许分发、复制或披露此类信息或部分信息。

1. 基本介绍

MS8006 是一款内嵌 32 位 ARM®Cortex®-M0+内核的超低功耗、Low Pin Count 和宽电压工作范围(2.4V~5.5V)的微控制器,最高可运行在 24MHz,内置 64K 字节的嵌入式 Flash, 4K 字节的 SRAM,集成了 12 位 1Msps 高精度 SAR 型 ADC、RTC、比较器、多路 UART、SPI、I2C 和 PWM 等丰富的外设接口,具有高整合度、高抗干扰、高可靠性的特点。MS8006 配合成熟的 Keil μ Vision 调试开发软件,支持 C 语言及汇编语言在线快速开发与调试。

2. 功能特征

内核

- ◆ ARM®Cortex®-M0+内核,最高频率 24MHz
- ◆ 一个 24 位系统定时器
- ◆ 支持低功耗睡眠模式
- ◆ 单周期 32 位硬件乘法器

存储器

- ◆ 64K 字节嵌入式 Flash,具有擦写保护功能
- ◆ 4K 字节 SRAM

时钟源与电源

- ◆ 4 路可选时钟源
 - 内部高速时钟: 4MHz~24MHz
 - 内部低速 38.4KHz/32.768KHz 时钟
 - 外部高速晶振: 4MHz-24MHz
 - 外部低速晶振: 32.768KHz
- ◆ 电源管理
 - 两种低功耗工作模式: Sleep、Deep Sleep Mode
 - 低电压检测,可配置为中断或复位

中断

- ◆ 嵌套向量中断控制器(NVIC)用于控制 32 个中断源,每个中断源可设置为 4 个优先级

- ◆ 支持串行调试(SWD)带 2 个观察点/4 个断点

通用 I/O 引脚

- ◆ 16 个 I/O 管脚

通信接口

- ◆ UART0-UART1 标准通讯接口
- ◆ 支持低速时钟的超低功耗 UART
- ◆ SPI 标准通讯接口,最高达 8Mbps/s
- ◆ I2C 标准通讯接口,最高达 1Mbps/s
- ◆ One-Wire 通讯接口

蜂鸣器频率发生器

定时器/计数器

- ◆ 1x16 位高级控制定时器: 有 4 通道 PWM 输出/输入捕获,支持 3 路互补输出,以及死区生成和紧急停止功能
- ◆ 1x16 位通用定时器,支持 4 路比较输出/输入捕获, PWM 输出
- ◆ 1x16 位可编程定时器阵列,支持 5 路输入捕获/比较输出, PWM 输出
- ◆ 2x16/32 位基础定时器/计数器
- ◆ 1x16 位低功耗定时器
- ◆ 自动唤醒定时器
- ◆ 系统窗口看门狗和独立看门狗定时器

通讯接口

- ◆ 支持 RTC 计数(秒/分/小时)及万年历功能(日/月/年)
- ◆ 支持闹铃功能寄存器(秒/分/小时/日/月/年)
- ◆ 支持 RTC 从 Deep Sleep 模式唤醒系统

ADC

- ◆ 7 通道 12 位 1Msps 采样速率, 12 位 SAR 型 ADC

工作条件电压比较器(VC)/低电压检查器

(LVD)

硬件 CRC-16 模块

工作条件

- ◆ 宽工作电压 2.4V-5.5V
- ◆ 宽工作频率最高至 24MHz
- ◆ 工作温度-40℃至+85℃

开发工具

- ◆ 全功能的嵌入式调试解决方案

封装

- ◆ TSSOP20

3. 目录

1.	基本介绍	2
2.	功能特征	2
3.	目录	4
4.	功能框图	26
5.	存储器概要	27
5.1	存储器映射	27
5.2	存储空间和模块地址	29
5.3	内嵌 SRAM	30
5.4	Flash 存储器简介	30
6.	工作模式和电源管理	31
6.1	运行模式(Active Mode)	32
6.2	休眠模式(Sleep Mode)	34
6.3	深度休眠模式(Deep Sleep Mode)	35
6.4	Cortex®-M0+内核系统控制寄存器(SCR)	37
7.	系统复位与时钟 (RCC)	39
7.1	复位	39
7.1.1	复位控制器介绍	39
7.1.2	复位源	40
7.2	系统时钟	41
7.2.1	简介	41
7.2.2	特性	42
7.2.3	系统时钟树	42
7.2.4	内部高速 RC 时钟 HIRC	44
7.2.5	内部低速 RC 时钟 LIRC	44
7.2.6	外部高速晶振时钟 HXT	44
7.2.7	外部低速晶振时钟 LXT	45
7.2.8	系统时钟启动过程	45
7.2.9	系统时钟切换	45

7.2.10	系统时钟输出.....	47
7.2.11	系统时钟安全控制.....	48
7.2.12	IWDG 时钟.....	48
7.2.13	RTC 时钟.....	48
7.2.14	AWK 时钟.....	48
7.2.15	低功耗模式.....	48
7.3	寄存器列表.....	49
7.4	寄存器说明.....	49
7.4.1	AHB 时钟分频寄存器(RCC_HCLKDIV).....	49
7.4.2	APB 时钟分频寄存器(RCC_PCLKDIV).....	50
7.4.3	AHB 周边模块时钟使能寄存器(RCC_HCLKEN).....	50
7.4.4	APB 周边模块时钟使能寄存器(RCC_PCLKEN).....	51
7.4.5	时钟输出控制寄存器(RCC_MCOCR).....	53
7.4.6	系统复位控制寄存器(RCC_RSTCR).....	54
7.4.7	系统复位状态寄存器(RCC_RSTSR).....	55
7.4.8	系统时钟源配置寄存器(RCC_SYSCLKCR).....	56
7.4.9	系统时钟源选择寄存器(RCC_SYSCLKSEL).....	57
7.4.10	内部高速 RC 振荡器控制寄存器(RCC_HIRCCR).....	58
7.4.11	外部高速晶体振荡器控制寄存器(RCC_HXTCR).....	59
7.4.12	内部低速 RC 振荡器控制寄存器(RCC_LIRCCR).....	60
7.4.13	外部低速晶体振荡器控制寄存器(RCC_LXTCR).....	61
7.4.14	M0 IRQ 延时控制寄存器(RCC_IRQLATENCY).....	62
7.4.15	System Tick timer 控制寄存器(RCC_STICKCR).....	62
7.4.16	SWDIO 端口控制寄存器(RCC_SWDIOCR).....	63
7.4.17	周边模块复位控制寄存器(RCC_PERIRST).....	63
7.4.18	RTC 复位控制寄存器(RCC_RTCRST).....	66
7.4.19	寄存器写保护控制寄存器(RCC_UNLOCK).....	66
8.	系统控制(SYSCON).....	68
8.1	寄存器列表.....	68
8.2	寄存器说明.....	68

8.2.1	系统配置寄存器 0(SYSCON_CFGR0)	68
8.2.2	端子 Deep Sleep 中断模式控制寄存器(SYSCON_PORTINTCR)	69
8.2.3	端子控制寄存器(SYSCON_PORTCR)	70
8.2.4	PCA 捕获通道控制寄存器(SYSCON_PCACR)	71
8.2.5	TIM1 通道输入源选择(SYSCON_TIM1CR)	72
8.2.6	TIM2 通道输入源选择(SYSCON_TIM2CR)	74
8.2.7	Syscon 寄存器写保护(SYSCON_UNLOCK)	76
9.	中断控制器 (NVIC)	77
9.1	简介	77
9.2	特性	78
9.3	中断优先级	78
9.4	中断向量表	79
9.5	中断唤醒控制 WIC	80
9.5.1	NVIC 从深度休眠模式唤醒设置进入中断 ISR 设置	81
9.5.2	NVIC 从深度休眠模式唤醒设置不执行中断 ISR 设置	81
9.5.3	使用退出休眠特性	82
9.6	软件基本操作	83
9.6.1	外部中断使能	83
9.6.2	NVIC 中断使能和清除使能	83
9.6.3	NVIC 中断挂起和清除挂起	83
9.6.4	NVIC 中断优先级	83
9.6.5	NVIC 中断屏蔽	84
9.7	寄存器列表	84
9.8	寄存器说明	85
9.8.1	中断使能设置寄存器(NVIC_ISER)	85
9.8.2	中断使能清除寄存器(NVIC_ICER)	85
9.8.3	中断挂起设置寄存器(NVIC_ISPR)	86
9.8.4	中断挂起清除寄存器(NVIC_ICPR)	86
9.8.5	中断优先级控制寄存器 0(NVIC_IPR0)	87
9.8.6	中断优先级控制寄存器 1(NVIC_IPR1)	87

9.8.7	中断优先级控制寄存器 2(NVIC_IPR2).....	88
9.8.8	中断优先级控制寄存器 3(NVIC_IPR3).....	88
9.8.9	中断优先级控制寄存器 4(NVIC_IPR4).....	89
9.8.10	中断优先级控制寄存器 5(NVIC_IPR5).....	89
9.8.11	中断优先级控制寄存器 6(NVIC_IPR6).....	90
9.8.12	中断优先级控制寄存器 7(NVIC_IPR7).....	90
10.	通用输入输出口(GPIO)	91
10.1	简介	91
10.2	特性	91
10.3	功能描述	92
10.3.1	通用 I/O(GPIO)	94
10.3.2	I/O 端口控制寄存器	94
10.3.3	I/O 端口数据寄存器	95
10.3.4	I/O 数据位处理	95
10.3.5	输入配置	95
10.3.6	输出配置	96
10.3.7	外部中断/唤醒线	97
10.3.8	I/O 引脚的复用功能和重映射	97
10.3.9	模拟配置	100
10.3.10	HXT 或 LXT 引脚用作 GPIO	101
10.4	GPIO 寄存器列表	102
10.5	GPIO 寄存器说明	103
10.5.1	GPIO 端口方向寄存器(GPIOx_DIRCR)(x=A..D)	103
10.5.2	GPIO 端口输出类型寄存器(GPIOx_OTYPER)(x=A..D)	104
10.5.3	GPIO 端口输出数据寄存器(GPIOx_ODR)(x=A..D)	104
10.5.4	GPIO 端口输入数据寄存器(GPIOx_IDR)(x=A..D)	105
10.5.5	GPIO 端口中断使能寄存器(GPIOx_INTEN)(x=A..D)	106
10.5.6	GPIO 端口中断原始状态寄存器(GPIOx_RAWINTSR)(x=A..D)	107
10.5.7	GPIO 端口中断状态寄存器(GPIOx_MSKINTSR)(x=A..D)	108
10.5.8	GPIO 端口中断清除寄存器(GPIOx_INTCLR)(x=A..D)	109

10.5.9	GPIO 端口中断类型寄存器(GPIOx_INTTYPCR)(x=A..D).....	110
10.5.10	GPIO 端口中断类型寄存器(GPIOx_INTPOLCR)(x=A..D).....	111
10.5.11	GPIO 端口任意边沿触发中断寄存器(GPIOx_INTANY)(x=A..D).....	112
10.5.12	GPIO 端口输出置位寄存器(GPIOx_ODSET)(x=A..D).....	113
10.5.13	GPIO 端口输出清除寄存器(GPIOx_ODCLR)(x=A..D).....	114
10.5.14	GPIO 端口输入去抖动寄存器(GPIOx_INDBEN)(x=A..D).....	115
10.5.15	GPIO 端口输入去抖动时钟配置寄存器(GPIOx_DBCLKCR)(x=A..D).....	116
10.5.16	GPIO 端口上拉/下拉寄存器(GPIOx_PUPDR)(x=A..D).....	117
10.5.17	GPIO 端口电压转换速率配置(GPIOx_SLEWCR)(x=A..D).....	118
10.5.18	GPIO 端口驱动强度配置寄存器(GPIOx_DRVCR)(x=A..D).....	119
10.5.19	GPIO 端口复用功能寄存器(GPIOx_AFR)(x=A..D).....	120
11.	FLASH 控制器 (FMC)	126
11.1	FMC 简介	126
11.2	Flash 结构框图.....	126
11.3	功能描述	127
11.3.1	擦除操作.....	127
11.3.2	写操作.....	129
11.3.3	读操作.....	130
11.3.4	擦写时间.....	130
11.3.5	擦写保护.....	131
11.3.6	系统BOOT 地址映射.....	132
11.4	寄存器列表	133
11.5	寄存器说明	134
11.5.1	Flash_CR(Flash 控制寄存器).....	134
11.5.2	Flash_CR(Flash 控制寄存器).....	134
11.5.3	Flash_ICLR(Flash 中断标记清除寄存器).....	135
11.5.4	Flash_BYPASS(BYPASS 系列寄存器).....	135
11.5.5	Flash_SLOCK0(sector 擦写保护寄存器 0).....	136
11.5.6	Flash_SLOCK1(sector 擦写保护寄存器 1).....	138
12.	CRC	142

12.1 概述	142
12.2 功能描述	142
12.2.1 CRC 编码模式	142
12.2.2 CRC 检验模式	143
12.3 寄存器列表	143
12.4 寄存器说明	143
12.4.1 CRC 结果寄存器(CRC_RESULT)	143
12.4.2 CRC 数据寄存器(CRC_DATA)	144
13. 高级控制定时器(TIM1)	146
13.1 TIM1 简介	146
13.2 TIM1 主要特性	146
13.3 TIM1 功能描述	148
13.3.1 时基单元	148
13.3.2 计数器模式	150
13.3.3 重复计数器	161
13.3.4 时钟选择	162
13.3.5 捕获/比较通道	166
13.3.6 输入捕获模式	168
13.3.7 PWM 输入模式	169
13.3.8 强制输出模式	170
13.3.9 输出比较模式	171
13.3.10 PWM 模式	172
13.3.11 互补输出和死区插入	176
13.3.12 使用刹车功能	178
13.3.13 在外部事件时清除 OCxREF 信号	181
13.3.14 产生六步 PWM 输出	181
13.3.15 单脉冲模式	182
13.3.16 编码器接口模式	184
13.3.17 定时器输入异或功能	187
13.3.18 与霍尔传感器的接口	187

13.3.19 TIM1 定时器和外部触发的同步.....	189
13.3.20 定时器同步.....	193
13.3.21 调试模式.....	193
13.4 TIM1 寄存器列表	194
13.5 TIM1 寄存器说明	194
13.5.1 TIM1 控制寄存器 1(TIM1_CR1)	194
13.5.2 TIM1 控制寄存器 2(TIM1_CR2)	196
13.5.3 TIM1 从模式控制寄存器(TIM1_SMCR).....	198
13.5.4 TIM1 中断使能寄存器(TIM1_DIER)	200
13.5.5 TIM1 状态寄存器(TIM1_SR).....	201
13.5.6 TIM1 事件产生 Th 寄存器(TIM1_EGR)	203
13.5.7 TIM1 捕获/比较模式寄存器 1(TIM1_CCMR1)	204
13.5.8 TIM1 捕获/比较模式寄存器 2(TIM1_CCMR2)	208
13.5.9 TIM1 捕获/比较使能寄存器(TIM1_CCER).....	210
13.5.10 TIM1 计数器(TIM1_CNT).....	213
13.5.11 TIM1 预分频器(TIM1_PSC)	213
13.5.12 TIM1 自动重载寄存器(TIM1_ARR).....	214
13.5.13 TIM1 重复计数寄存器(TIM1_RCR).....	214
13.5.14 TIM1 捕获/比较寄存器 1(TIM1_CCR1).....	215
13.5.15 TIM1 捕获/比较寄存器 2(TIM1_CCR2).....	215
13.5.16 TIM1 捕获/比较寄存器 3(TIM1_CCR3).....	216
13.5.17 TIM1 捕获/比较寄存器 4(TIM1_CCR4).....	217
13.5.18 TIM1 刹车和死区寄存器(TIM1_BDTR)	218
14. 通用定时器 (TIM2)	221
14.1 TIM2 简介	221
14.2 TIM2 主要功能	221
14.3 TIM2 功能描述	223
14.3.1 时基单元.....	223
14.3.2 计数器模式.....	225
14.3.3 时钟选择.....	236

14.3.4 捕获/比较通道.....	240
14.3.5 输入捕获模式.....	242
14.3.6 PWM 输入模式.....	243
14.3.7 强置输出模式.....	244
14.3.8 输出比较模式.....	244
14.3.9 PWM 模式.....	246
14.3.10 单脉冲模式.....	250
14.3.11 在外部事件时清除 OCxREF 信号.....	252
14.3.12 编码器接口模式.....	253
14.3.13 定时器输入异或功能.....	256
14.3.14 定时器和外部触发的同步.....	256
14.3.15 定时器同步.....	260
14.3.16 调试模式.....	266
14.4 TIM2 寄存器列表.....	266
14.5 TIM2 寄存器说明.....	267
14.5.1 TIM2 控制寄存器 1(TIM2_CR1).....	267
14.5.2 TIM2 控制寄存器 2(TIM2_CR2).....	269
14.5.3 TIM2 从模式控制寄存器(TIM2_SMCR).....	271
14.5.4 TIM2 中断使能寄存器(TIM2_DIER).....	273
14.5.5 TIM2 状态寄存器(TIM2_SR).....	274
14.5.6 TIM2 事件产生 Th 寄存器(TIM2_EGR).....	276
14.5.7 TIM2 捕获/比较模式寄存器 1(TIM2_CCMR1).....	277
14.5.8 TIM2 捕获/比较模式寄存器 2(TIM2_CCMR2).....	280
14.5.9 TIM2 捕获/比较使能寄存器(TIM1_CCER).....	282
14.5.10 TIM2 计数器(TIM2_CNT).....	284
14.5.11 TIM2 预分频器(TIM2_PSC).....	284
14.5.12 TIM2 自动重装载寄存器(TIM2_ARR).....	285
14.5.13 TIM2 捕获/比较寄存器 1(TIM2_CCR1).....	285
14.5.14 TIM2 捕获/比较寄存器 2(TIM2_CCR2).....	286
14.5.15 TIM2 捕获/比较寄存器 3(TIM2_CCR3).....	286

14.5.16	TIM2 捕获/比较寄存器 4(TIM2_CCR4).....	287
15.	可编程计数阵列 (PCA)	289
15.1	PCA 简介.....	289
15.2	PCA 功能描述.....	289
15.2.1	PCA 定时/计数器.....	290
15.2.2	捕获功能.....	291
15.2.3	PCA 比较功能.....	293
15.3	PCA 模块与其他模块互连及控制.....	298
15.3.1	ECI 互连.....	298
15.3.2	PCACAP0.....	298
15.3.3	PCACAP[4:1]	298
15.4	PCA 寄存器列表.....	298
15.5	寄存器说明	299
15.5.1	控制寄存器(PCA_CR)	299
15.5.2	模式寄存器(PCA_CMOD).....	300
15.5.3	计数寄存器(PCA_CNT)	301
15.5.4	中断清除寄存器(PCA_INTCLR).....	302
15.5.5	比较捕获模式寄存器(PCA_CCAPM0~4).....	302
15.5.6	比较捕获数据寄存器高 8 位(PCA_CCAP0~4H).....	303
15.5.7	比较捕获数据寄存器低 8 位(PCA_CCAP0~4L)	304
15.5.8	比较捕获 16 位寄存器(PCA_CCAP0~4)	305
15.5.9	比较高速输出标志寄存器(PCA_CCAPO).....	305
15.5.10	端子输出控制寄存器(PCA_POOCR)	306
16.	Base Timer (TIM10,TIM11)	308
16.1	Base Timer 简介.....	308
16.2	Base Timer 特性.....	308
16.3	BaseTimer 功能描述.....	309
16.3.1	计数功能.....	311
16.3.2	定时功能.....	312
16.3.3	Buzzer 功能.....	312

16.4	Base Timer 互联	312
16.4.1	GATE 互联.....	312
16.4.2	Toggle 输出互联.....	313
16.5	Base Timer 寄存器列表	313
16.6	Base Timer 寄存器说明	314
16.6.1	控制寄存器(TIMx_CR)	314
16.6.2	立即重载寄存器(TIMx_LOAD)	315
16.6.3	计数器寄存器(TIMx_CNT)	315
16.6.4	原始中断状态寄存器(TIMx_RAWINTSR)	316
16.6.5	中断标志寄存器(TIMx_MSKINTSR)	316
16.6.6	中断清除寄存器(TIMx_INTCLR)	317
16.6.7	周期重载寄存器(TIMx_BGLOAD)	317
17.	低功耗定时器(LPTIM)	318
17.1	LPTIM 功能描述	318
17.1.1	计数功能.....	320
17.2	LPTIM 互联	321
17.2.1	GATE 互联.....	321
17.2.2	EXT 互联.....	321
17.2.3	TOGGLE 输出互联.....	321
17.3	寄存器列表	321
17.4	寄存器说明	322
17.4.1	LPTIM 计数值只读寄存器(LPTIM_CNTVAL)	322
17.4.2	LPTIM 控制寄存器(LPTIM_CR)	322
17.4.3	LPTIM 立即重载寄存器(LPTIM_LOAD)	323
17.4.4	LPTIM 中断寄存器(LPTIM_INTSR)	324
17.4.5	LPTIM 中断寄存器(LPTIM_INTCLR)	324
17.4.6	LPTIM 周期重载寄存器(LPTIM_BGLOAD)	325
18.	自唤醒定时器(AWK)	326
18.1	寄存器列表	326
18.2	寄存器说明	327

18.2.1	自唤醒定时器控制寄存器(AWK_CR).....	327
18.2.2	自唤醒定时器重装载数据寄存器(AWK_RLOAD).....	328
18.2.3	自唤醒定时器状态寄存器(AWK_SR).....	328
18.2.4	自唤醒中断清除寄存器(AWK_INTCLR).....	328
19.	蜂鸣器 (BEEP)	330
19.1	简介	330
19.2	特性	330
19.3	功能描述	330
19.3.1	蜂鸣器操作.....	330
19.3.2	蜂鸣器校准.....	331
19.4	寄存器列表	331
19.5	寄存器说明	331
19.5.1	蜂鸣器控制/状态寄存器(BEEP_CSR).....	331
20.	独立看门狗 (IWDG)	333
20.1	简介	333
20.2	特性	333
20.3	功能描述	334
20.3.1	超时周期.....	334
20.3.2	IWDG 溢出后产 Th 中断.....	334
20.3.3	IWDG 溢出后产 Th 复位.....	335
20.4	寄存器列表	335
20.5	寄存器说明	335
20.5.1	IWDG 控制命令寄存器(IWDG_CMDCR).....	335
20.5.2	IWDG 配置寄存器(IWDG_CFGR).....	336
20.5.3	IWDG 计数器重装载寄存器(IWDG_RLOAD).....	337
20.5.4	IWDG 计数器值寄存器(IWDG_CNTVAL).....	337
20.5.5	IWDG 中断状态寄存器(IWDG_SR).....	337
20.5.6	IWDG 中断清除寄存器(IWDG_INTCLR).....	338
20.5.7	IWDG 保护寄存器(IWDG_UNLOCK).....	338
20.6	注意	339

21. 系统窗口看门狗(WWDG)	340
21.1 概述	340
21.2 特征	340
21.3 结构框图	340
21.4 基本配置	341
21.5 功能描述	341
21.5.1 窗口看门狗定时器的计数	341
21.5.2 窗口看门狗定时器比较中断	342
21.5.3 窗口看门狗定时器复位系统	342
21.5.4 窗口看门狗定时器的窗口设置限制	343
21.6 与独立看门狗定时器(IWDG)比较	343
21.6.1 复位条件和复位延时	343
21.6.2 唤醒功能	343
21.7 寄存器列表	343
21.8 寄存器说明	344
21.8.1 窗口看门狗定时器重载计数寄存器(WWDG_RLOAD)	344
21.8.2 窗口看门狗定时器控制寄存器(WWDG_CR)	344
21.8.3 窗口看门狗定时器中断使能寄存器(WWDG_INTEN)	345
21.8.4 窗口看门狗定时器状态寄存器(WWDG_SR)	346
21.8.5 窗口看门狗定时器中断清除寄存器(WWDG_INTCLR)	346
21.8.6 窗口看门狗定时器计数器值寄存器(WWDG_CNTVAL)	347
22. 通用异步收发器 (UART)	348
22.1 简介	348
22.2 结构框图	348
22.3 工作模式	348
22.3.1 Mode0 (同步模式, 半双工)	348
22.3.2 Mode1 (异步模式, 全双工)	350
22.3.3 Mode2 (异步模式, 全双工)	351
22.3.4 Mode3 (异步模式, 全双工)	352
22.3.5 波特率编程	352

22.3.6 帧错误检测.....	353
22.3.7 多机通讯.....	353
22.3.8 自动地址识别.....	354
22.3.9 给定地址.....	354
22.3.10 广播地址.....	354
22.3.11 举例.....	354
22.4 收发端缓存	355
22.4.1 接收缓存.....	355
22.4.2 发送缓存.....	355
22.5 IrDA 红外功能	355
22.5.1 IrDA 低功耗模式.....	356
22.6 波特率的分频设置	358
22.7 UART 寄存器列表.....	362
22.8 UART 寄存器说明.....	363
22.8.1 UART 控制寄存器(UARTx_SCON)	363
22.8.2 UART 数据寄存器(UARTx_SBUF).....	364
22.8.3 UART 地址寄存器(UARTx_SADDR).....	364
22.8.4 UART 地址掩码寄存器(UARTx_SADEN).....	365
22.8.5 UART 标志位寄存器(UARTx_INTSR).....	365
22.8.6 UART 标志位清除寄存器(UARTx_INTCLR).....	366
22.8.7 UART 波特率控制寄存器(UARTx_BAUDCR).....	366
23. 低功耗通用异步收发器(LPUART).....	368
23.1 概述	368
23.2 结构框图	369
23.3 工作模式	369
23.3.1 MODE0(同步模式, 半双工).....	369
23.3.2 MODE1(异步模式, 全双工).....	370
23.3.3 MODE2(异步模式, 全双工).....	371
23.3.4 MODE3(异步模式, 全双工).....	372
23.4 波特率编程	373

23.4.1	MODE0	373
23.4.2	MODE1/3	373
23.4.3	MODE2	374
23.5	帧错误检测	374
23.6	多机通讯	374
23.7	自动地址识别	375
23.8	给定地址	375
23.9	广播地址	375
23.10	举例	376
23.11	收发端缓存	376
23.11.1	接收缓存	376
23.11.2	发送缓存 LPUART	376
23.12	寄存器列表	377
23.13	寄存器说明	377
23.13.1	LPUART 控制寄存器(LPUART_SCON)	377
23.13.2	LPUART 数据寄存器(LPUART_SBUF)	379
23.13.3	LPUART 地址寄存器(LPUART_SADDR)	379
23.13.4	LPUART 地址掩码寄存器(LPUART_SADEN)	379
23.13.5	LPUART 标志位寄存器(LPUART_INTSR)	380
23.13.6	LPUART 标志位清除寄存器(LPUART_INTCLR)	381
23.13.7	波特率控制寄存器(LPUART_BAUDCR)	381
24.	I2C 接口	382
24.1	I2C 简介	382
24.2	I2C 主要特性	382
24.3	I2C 协议描述	383
24.3.1	I2C 总线上数据传输	383
24.3.2	起始位或重复起始信号	384
24.3.3	从机地址传输	384
24.3.4	数据传输	384
24.4	I2C 功能描述	385

24.4.1 I2C 工作模式.....	386
24.4.2 仲裁与同步逻辑.....	387
24.4.3 串行时钟发生器.....	388
24.4.4 输入滤波器.....	388
24.4.5 地址比较器.....	388
24.4.6 中断产生器.....	389
24.4.7 I2C 主机发送模式.....	389
24.4.8 I2C 主机接收模式.....	392
24.4.9 I2C 从机接收模式.....	394
24.4.10 I2C 从机发送模式.....	398
24.4.11 其他杂项状态.....	400
24.5 I2C 操作模式	401
24.5.1 初始化程序.....	401
24.5.2 端口配置程序.....	401
24.5.3 启动主机发送功能.....	402
24.5.4 启动主机接收功能.....	402
24.5.5 I2C 中断程序.....	402
24.5.6 无指定模式状态.....	402
24.5.7 主发送状态.....	403
24.5.8 主接收状态.....	405
24.5.9 从接收状态.....	406
24.5.10 从发送状态.....	409
24.6 I2C 寄存器列表	410
24.7 I2C 寄存器说明	411
24.7.1 I2C 配置寄存器(I2C_CR).....	411
24.7.2 I2C 数据寄存器(I2C_DATA).....	411
24.7.3 I2C 地址寄存器(I2C_ADDR).....	412
24.7.4 I2C 状态寄存器(I2C_SR).....	412
24.7.5 I2C 波特率计数器使能寄存器(I2C_TIMRUN)	413
24.7.6 波特率计数器配置寄存器(I2C_BAUDCR).....	413

25. 串行外设接口(SPI)	415
25.1 SPI 简介	415
25.2 SPI 主要特性	415
25.3 功能描述	415
25.3.1 SPI 主机方式	415
25.3.2 操作流程	416
25.3.3 SPI 从机方式	417
25.4 SPI 中断	419
25.5 多主机/多从机模式	420
25.6 SPI 寄存器列表	420
25.7 SPI 寄存器说明	421
25.7.1 SPI 配置寄存器(SPI_CR)	421
25.7.2 SPI 片选配置寄存器(SPI_SSN)	422
25.7.3 SPI 状态寄存器(SPI_SR)	422
25.7.4 SPI 数据寄存器(SPI_DATA)	423
26. One-Wire 接口	424
26.1 单总线协议(One-Wire)	424
26.1.1 特点	424
26.1.2 优点	424
26.2 单总线通信过程	424
26.2.1 初始化	424
26.2.2 写时间间隙	425
26.2.3 读时间间隙	425
26.3 配置说明	426
26.3.1 初始化配置说明	426
26.3.2 读数据配置说明	426
26.3.3 写数据配置说明配置	427
26.4 寄存器列表	427
26.5 寄存器说明	428
26.5.1 1-Wire 模块控制寄存器(OWIRE_CR)	428

26.5.2	1-Wire 输入端子滤波控制寄存器(OWIRE_NFCR).....	429
26.5.3	1-Wire RESET 宽度控制寄存器(OWIRE_RSTCNT)	430
26.5.4	1-Wire Presence Pulse 宽度计数寄存器(OWIRE_PRESCNT).....	430
26.5.5	1-Wire Bit rate 设计计数器(OWIRE_BITRATECNT).....	430
26.5.6	1-Wire 主器件读/写PULL0 驱动时间(OWIRE_DRVCNT).....	431
26.5.7	1-Wire 主器件读采样时间设定(OWIRE_RDSMPCNT).....	431
26.5.8	1-Wire Recover Time 计数区间值(OWIRE_REC CNT).....	432
26.5.9	1-Wire 数据寄存器(OWIRE_DATA).....	432
26.5.10	1-Wire 总线操作命令寄存器(OWIRE_CMD)	433
26.5.11	1-wire 中断使能寄存器(OWIRE_INTEN)	433
26.5.12	1-wire 状态寄存器(OWIRE_SR).....	434
26.5.13	1-wire 状态清除寄存器(OWIRE_INTCLR).....	435
27.	时钟校准/监测模块(CLKTRIM).....	436
27.1	简介	436
27.2	主要特性	436
27.3	CLKTRIM 功能描述	436
27.3.1	CLKTRIM 校准模式.....	436
27.3.2	CLKTRIM 监测模式.....	437
27.4	CLKTRIM 寄存器列表	438
27.5	CLK_TRIM 寄存器说明	439
27.5.1	配置寄存器(CLKTRIM_CR)	439
27.5.2	参考计数器处置配置寄存器(CLKTRIM_REFCON).....	440
27.5.3	参考计数器值寄存器(CLKTRIM_REFCNT)	440
27.5.4	校准计数器值寄存器(CLKTRIM_CALCNT)	440
27.5.5	中断标志位寄存器(CLKTRIM_IFR).....	441
27.5.6	中断标志位清除寄存器(CLKTRIM_ICLR).....	442
27.5.7	校准计数器溢出值配置寄存器(CLKTRIM_CALCON).....	442
28.	实时时钟(RTC).....	443
28.1	简介	443
28.2	主要特性	443

28.3	RTC 功能描述	444
28.3.1	RTC 结构框图	444
28.3.2	RTC 时钟	444
28.3.3	复位过程	445
28.3.4	寄存器的写保护	445
28.3.5	日历初始化及配置	445
28.3.6	读出计数寄存器	446
28.3.7	写入计数寄存器	447
28.3.8	闹钟设定	447
28.3.9	校准 1HZ 输出	447
28.3.10	RTC 时钟校准	448
28.4	RTC 中断	448
28.4.1	RTC 闹钟中断	448
28.4.2	RTC 周期中断	448
28.5	RTC 寄存器列表	449
28.6	RTC 寄存器说明	449
28.6.1	RTC 控制寄存器(RTC_CR)	449
28.6.2	RTC 时钟控制寄存器(RTC_CLKCR)	451
28.6.3	RTC 时间寄存器(RTC_TIME)	451
28.6.4	RTC 日期寄存器(RTC_DATE)	452
28.6.5	RTC 时间闹钟寄存器(RTC_ALM1TIME)	453
28.6.6	RTC 日期闹钟寄存器(RTC_ALM1DATE)	454
28.6.7	RTC 周期闹钟寄存器(RTC_ALM2PRD)	455
28.6.8	RTC 时钟调校寄存器(RTC_CLKTRIM)	456
28.6.9	RTC 初始化和状态寄存器(RTC_ISR)	456
28.6.10	RTC 状态清除寄存器(RTC_INTCLR)	458
29.	模拟/数字转换器(ADC)	459
29.1	模块简介	459
29.2	ADC 框图	460
29.3	转换时序及速度	460

29.4 单次转换模式	461
29.5 连续转换模式	462
29.6 连续转换累加模式	463
29.7 ADC 转换结果比较	465
29.8 ADC 中断	466
29.9 寄存器列表	466
29.10 寄存器说明	467
29.10.1 ADC 配置寄存器 0(ADC_CR0)	467
29.10.2 ADC 配置寄存器 1(ADC_CR1)	468
29.10.3 ADC 配置寄存器 2(ADC_CR2)	471
29.10.4 ADC 通道 0 转换结果(ADC_RESULT 0)	472
29.10.5 ADC 通道 1 转换结果(ADC_RESULT 1)	473
29.10.6 ADC 通道 2 转换结果(ADC_RESULT 2)	473
29.10.7 ADC 通道 3 转换结果(ADC_RESULT 3)	473
29.10.8 ADC 通道 4 转换结果(ADC_RESULT 4)	474
29.10.9 ADC 通道 5 转换结果(ADC_RESULT 5)	474
29.10.10 ADC 通道 6 转换结果(ADC_RESULT 6)	475
29.10.11 ADC 通道 7 转换结果(ADC_RESULT 7)	475
29.10.12 ADC 转换结果(ADC_RESULT)	476
29.10.13 ADC 转换结果累加值(ADC_RESULT_ACC)	476
29.10.14 ADC 比较上阈值(ADC_HT)	477
29.10.15 ADC 比较下阈值(ADC_LT)	477
29.10.16 ADC 中断使能寄存器(ADC_INTEN)	478
29.10.17 ADC 中断清除寄存器(ADC_INTCLR)	479
29.10.18 ADC 掩码前中断状态寄存器(ADC_RAWINTSR)	479
29.10.19 ADC 掩码后中断状态寄存器(ADC_MSKINTSR)	480
30. 低电压检测器 (LVD)	482
30.1 简介	482
30.2 特性	482
30.3 功能描述	483

30.3.1 LVD 框图.....	483
30.3.2 数字滤波.....	483
30.3.3 迟滞功能.....	483
30.4 配置示例	484
30.4.1 LVD 配置为低电压复位	484
30.4.2 LVD 配置为电压变化中断	484
30.5 寄存器列表	485
30.6 寄存器说明	485
30.6.1 LVD 控制寄存器(LVD_CR)	485
30.6.2 LVD 状态寄存器(LVD_SR).....	487
31. 电压比较器(VC).....	488
31.1 简介	488
31.2 VC 框图.....	488
31.3 数字滤波	488
31.4 迟滞功能	489
31.5 配置示例	489
31.6 寄存器列表	490
31.7 寄存器说明	490
31.7.1 VC 电压控制寄存器(VC_CR0).....	490
31.7.2 VC 控制寄存器(VC_CR1).....	491
31.7.3 VC 输出配置寄存器(VC_OUTCFG).....	492
31.7.4 VC 状态寄存器(VC_SR).....	494
32. Debug 支持.....	495
32.1 SWD 保护位工作原理.....	495
32.2 寄存器列表	495
32.3 Debug 模式控制寄存器(DBG_APBZ)	496
33. SysTick 定时器(SYST).....	498
33.1 SysTick 定时器简介	498
33.2 设置 SysTick 定时器	498
33.3 SysTick 定时器寄存器列表	499

33.4 SysTick 定时器寄存器说明	499
33.4.1 SysTick 定时器控制和状态寄存器(SYST_CSR).....	499
33.4.2 SysTick 定时器重载值寄存器(SYST_RVR).....	500
33.4.3 SysTick 定时器当前值寄存器(SYST_CVR)	500
33.4.4 SysTick 定时器校准值寄存器(SYST_CALIB).....	500
34. 引脚图	501
35. 引脚描述	502
36. 电气特性	503
36.1 最大绝对额定值	503
36.2 推荐工作条件	503
36.3 直流特性	504
36.3.1 中断使能设置寄存器(NVIC_ISER).....	504
36.3.2 中断使能设置寄存器(NVIC_ISER).....	506
36.4 交流特性	506
36.4.1 输出特性—端口 PA, PB, PC, PD.....	506
36.4.2 输入特性—端口 PA, PB, PC, PD.....	506
36.4.3 端口漏电特性—端口 PA, PB, PC, PD.....	507
36.4.4 定时器/计数器输入采样要求.....	507
36.4.5 内部 HIRC 振荡器	508
36.4.6 内部 LIRC 振荡器.....	508
36.4.7 外部 32.768KHz 晶振.....	508
36.4.8 外部 HXT 晶振.....	509
36.5 12 位 A/D 转换器.....	510
36.6 模拟电压比较器	511
36.7 低电压检查特性	512
36.8 内存擦/写特性	513
36.9 低功耗模式返回时间	513
36.10 电磁敏感特性	513
36.10.1 ESD	513
36.10.2 静态栓锁(Static Latch-up).....	513



37.	封装信息	514
38.	版本记录	515

5. 存储器概要

5.1 存储器映射

系统的地址空间总共有 4GB，包含程序存储空间，数据存储空间，周边模块寄存器，I/O 端口等。数据使用小端点格式，就是数据的高字节保存在内存的高地址中，而数据的低字节保存在内存的低地址中。整个系统地址空间的划分如下图，图 5-1 所示：

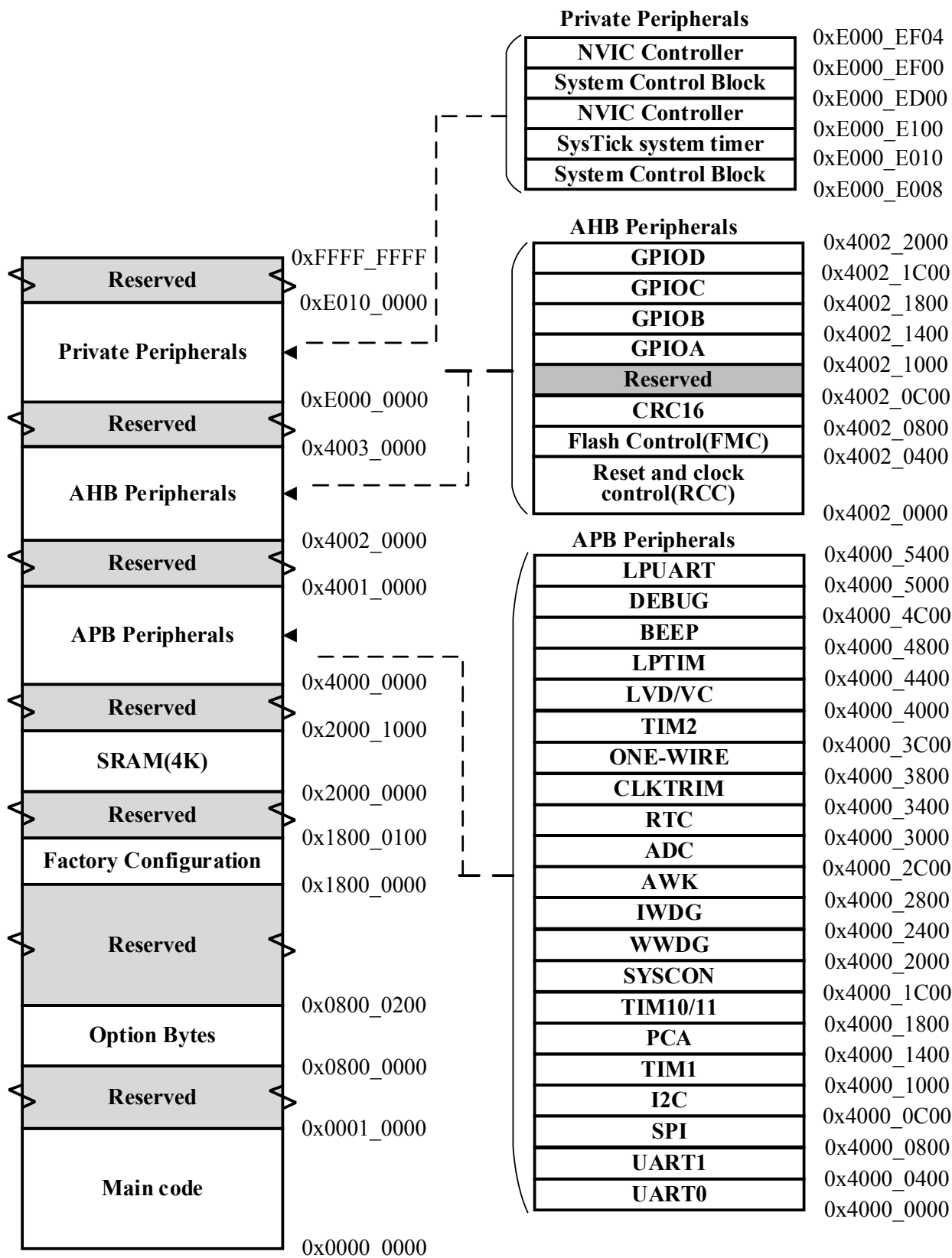


图 5-1. 存储器映射

5.2 存储空间和模块地址

MS8006 器件内部包含的各模块的地址空间和边界信息如下：

表 5.1. MS8006 存储器映射和外设寄存器编址

总线	边界地址	空间大小 (Bytes)	模块
	0xE000_0000-0xE00F_FFFF	1MB	Coretex-M0+peripheral
AHB	0x4003_0000-0xDFFF_FFFF		保留
	0x4002_1000-0x4002_1FFF	1K	GPIOD
	0x4002_1000-0x4002_1BFF	1K	GPIOC
	0x4002_1000-0x4002_17FF	1K	GPIOB
	0x4002_1000-0x4002_13FF	1K	GPIOA
	0x4002_0C00-0x4002_0FFF	1K	保留
	0x4002_0800-0x4002_0BFF	1K	CRC16
	0x4002_0400-0x4002_07FF	1K	FMC
	0x4002_0000-0x4002_03FF	1K	RCC
	0x4000_5400-0x4001_FFFF		保留
APB	0x4000_5000-0x4000_53FF	1K	LPUART
	0x4000_4C00-0x4000_4FFF	1K	DEBUG
	0x4000_4800-0x4000_4BFF	1K	BEEP
	0x4000_4400-0x4000_47FF	1K	LPTIM
	0x4000_4000-0x4000_43FF	1K	LVD/VC
	0x4000_3C00-0x4000_3FFF	1K	TIM2
	0x4000_3800-0x4000_3BFF	1K	ONE-WIER
	0x4000_3400-0x4000_37FF	1K	CLKTRIM
	0x4000_3000-0x4000_33FF	1K	RTC
	0x4000_2C00-0x4000_2FFF	1K	ADC
	0x4000_2800-0x4000_2BFF	1K	AWK
	0x4000_2400-0x4000_27FF	1K	IWDT
	0x4000_2000-0x4000_23FF	1K	WWDT
	0x4000_1C00-0x4000_1FFF	1K	SYSCON
	0x4000_1800-0x4000_1BFF	1K	TIM10/11
	0x4000_1400-0x4000_17FF	1K	PCA
	0x4000_1000-0x4000_13FF	1K	TIM1
	0x4000_0C00-0x4000_0FFF	1K	I2C
	0x4000_0800-0x4000_0BFF	1K	SPI
	0x4000_0400-0x4000_07FF	1K	UART1
	0x4000_0000-0x4000_03FF	1K	UART0
AHB	0x2000_1000-0x3FFF_FFFF		保留

总线	边界地址	空间大小 (Bytes)	模块
	0x2000_0000-0x2000_0FFF	4K	SRAM
	0x1800_0100-0x1FFF_FFFF		保留
	0x1800_0000-0x1800_00FF	256	Factory Information
	0x0800_0200-0x17FF_FFFF		保留
	0x0800_0000-0x0800_01FF	512	Option Bytes
	0x0001_0000-0x07FF_FFFF		保留
	0x0000_0000-0x0000_FFFF	64K	Flash

5.3 内嵌 SRAM

MS8006 内置 4K 字节的静态 SRAM。它可以以字节(8 位)、半字(16 位)或字(32 位)进行访问。

5.4 Flash 存储器简介

Flash 存储器有两个不同存储区域：

- 主存储块，它包括应用程序和用户数据区（若需要时）
- 信息块，其包含两个部分：
 - 选项字节(Option bytes)－内含硬件及存储保护用户配置选项。
 - 系统存储器(System memory)

Flash 存储器接口基于 AHB 协议执行指令和数据存取。

6. 工作模式和电源管理

MS8006 的电源管理模块负责管理本产品各种工作模式之间的切换，以及控制各工作模式下的各功能模块的工作状态。本产品的工作电压(VCC)为 2.4V~5.5V。本产品有如下几个工作模式：

- 1) 运行模式：CPU 运行，周边功能模块运行。
- 2) 休眠模式：CPU 停止运行，周边功能模块运行。
- 3) 深度休眠模式：CPU 停止运行，高速时钟停止运行。

从运行模式，通过执行软件程序，可进入其他低功耗模式。从其他各种低功耗模式，通过中断触发，可回到运行模式。

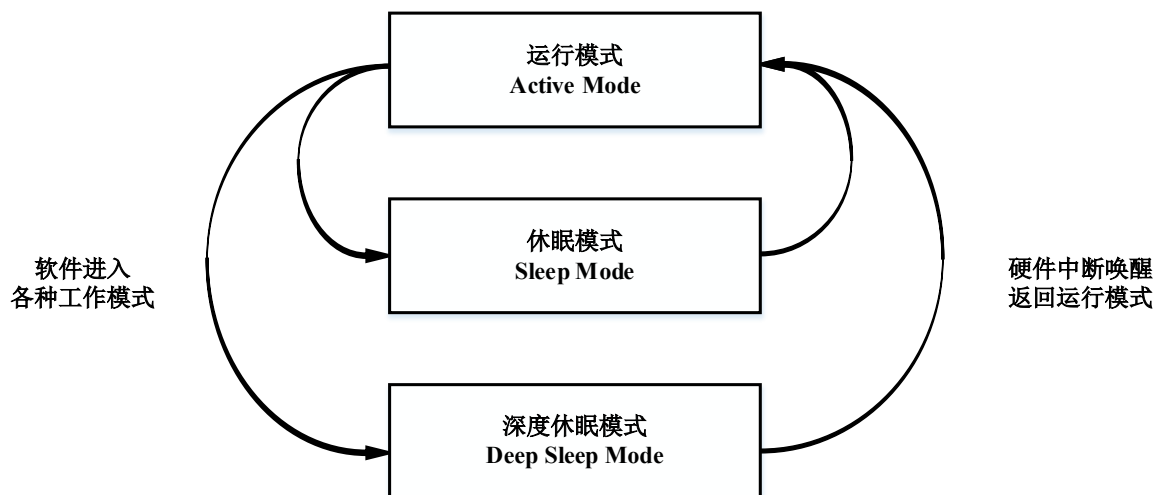


图 6-1. 控制模式框图

在各种模式下，CPU 可响应的中断类型请参考表 9.1 中断源。

各模式下，可以响应的复位源类型见下表。

表 6.1. 可相应的复位源类型

编号	复位源	运行模式	Sleep 模式唤醒	Deep Sleep 模式唤醒
1	上电/掉电复位	Y	Y	Y
2	外部 Reset Pin 复位	Y	Y	Y
3	IWDG 复位	Y	Y	Y
4	WWDG 复位	Y	Y	N
5	系统软件复位	Y	N	N

编号	复位源	运行模式	Sleep 模式唤醒	Deep Sleep 模式唤醒
6	欠电压(LVD)复位	Y	Y	Y
7	LOCKUP 复位	Y	N	N
8	寄存器 CPURST 复位	Y	N	N
9	寄存器 MCURST 复位	Y	N	N

6.1 运行模式(Active Mode)

系统在电源上电复位后，或从各低功耗模式唤醒后，微控制器 MCU 处于运行状态，各模块的运行状态如图 6-2 运行模式下可运行模块一览所示。当 CPU 不需继续运行时，可以利用多种低功耗模式来节能。用户需要根据最低能耗、最快速启动时间、可用的唤醒源等条件，选定一个最佳的低功耗模式。

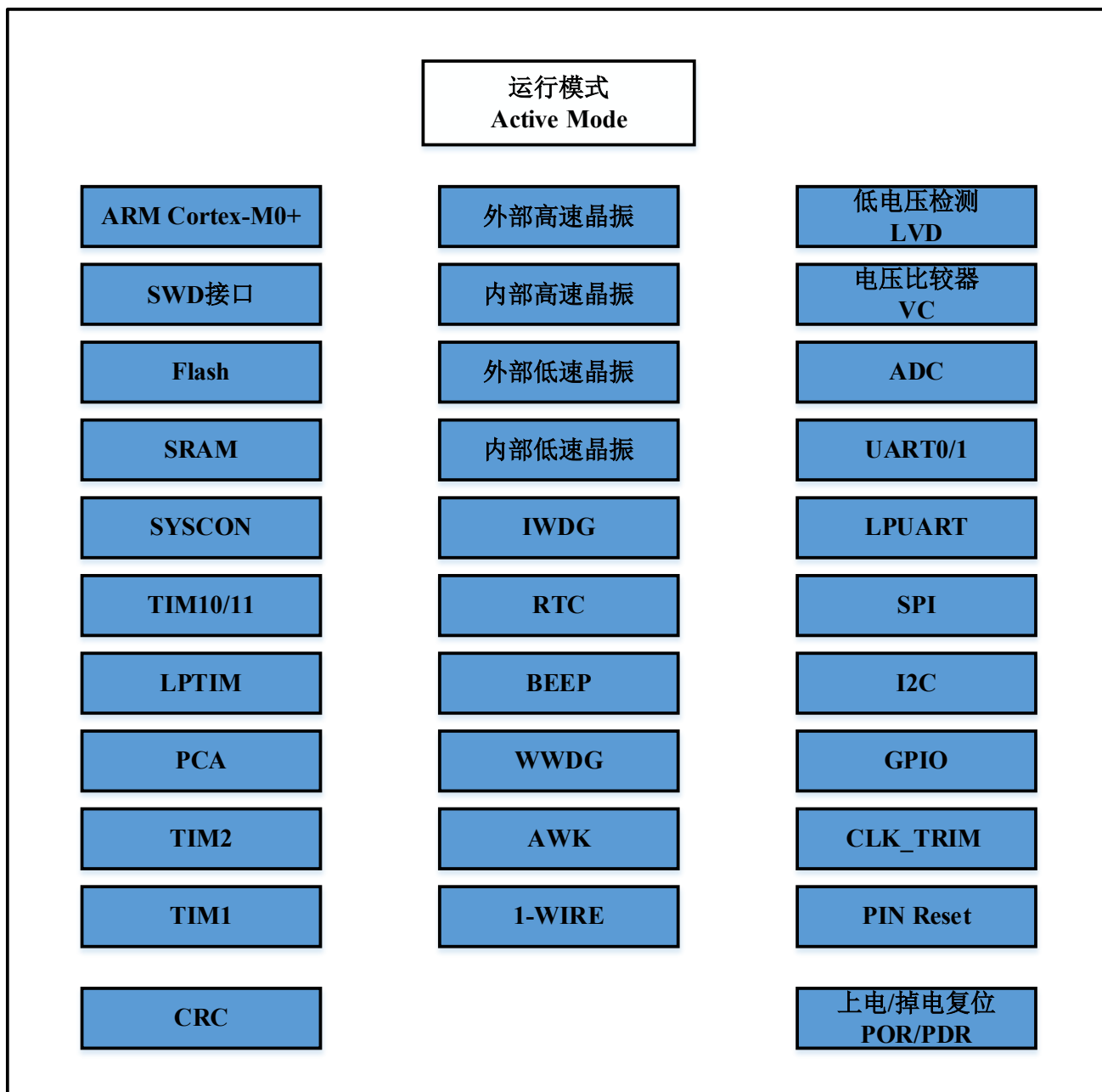


图 6-2. 运行模式下可运行模块一览

几种降低运行模式下芯片功耗的方法：

- 1) 在运行模式下，通过对预分频寄存器(RCC_HCLKDIV,RCC_PCLKDIV)进行编程，可以降低任意一个系统时钟(HCLK，PCLK)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。
- 2) 在运行模式下，关闭不使用外设的时钟(RCC_HCLKEN,RCC_PCLKEN)来减少功耗。
- 3) 使用低功耗模式代替休眠模式，因为本产品的唤醒时间极短(~3uS)，亦可满足系统的实

时响应的需求。

6.2 休眠模式(Sleep Mode)

使用 WFI 指令可以进入休眠模式，休眠模式下，CPU 停止运行，但系统时钟、NVIC 中断处理以及非 HCLK 驱动的周边功能模块仍都可以工作。

系统进入休眠状态，不会改变端口状态，在进入休眠前请根据需要更改 IO 的状态为休眠模式下的状态。

如何进入休眠模式：

通过执行 WFI 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：

- 1) Sleep-NOW: 如果 SleepONEXIT=0，当 WFI 或 WFE 被执行时，微控制器立即进入睡眠模式。
- 2) Sleep-ON-EXIT: 如果 SleepONEXIT=1，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。

如何退出休眠模式：

如果执行 WFI 指令进入睡眠模式，任意一个高优先级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。

使用注意：

- Sleep-ON-EXIT=1，执行完中断自动进入 Sleep，程序不需要写__wfi()；
- Sleep-ON-EXIT=0，main()执行__wfi()后进入 Sleeping，中断触发且执行完中断程序返回 main()后，执行 WFI 指令后进入 Sleeping。等待后续中断触发。
- Sleep-ON-EXIT 位不影响__wfi()指令的执行。Sleep-ON-EXIT=0：main()执行__wfi()后进入 Sleeping，中断触发且执行完中断程序返回 main()后，继续往下执行；
- 若在中断中进入 Sleeping 后，只有优先级高于此中断的中断才能唤醒，先执行高优先

级，再执行低优先级；优先级低于或等于当前中断的中断不能唤醒。

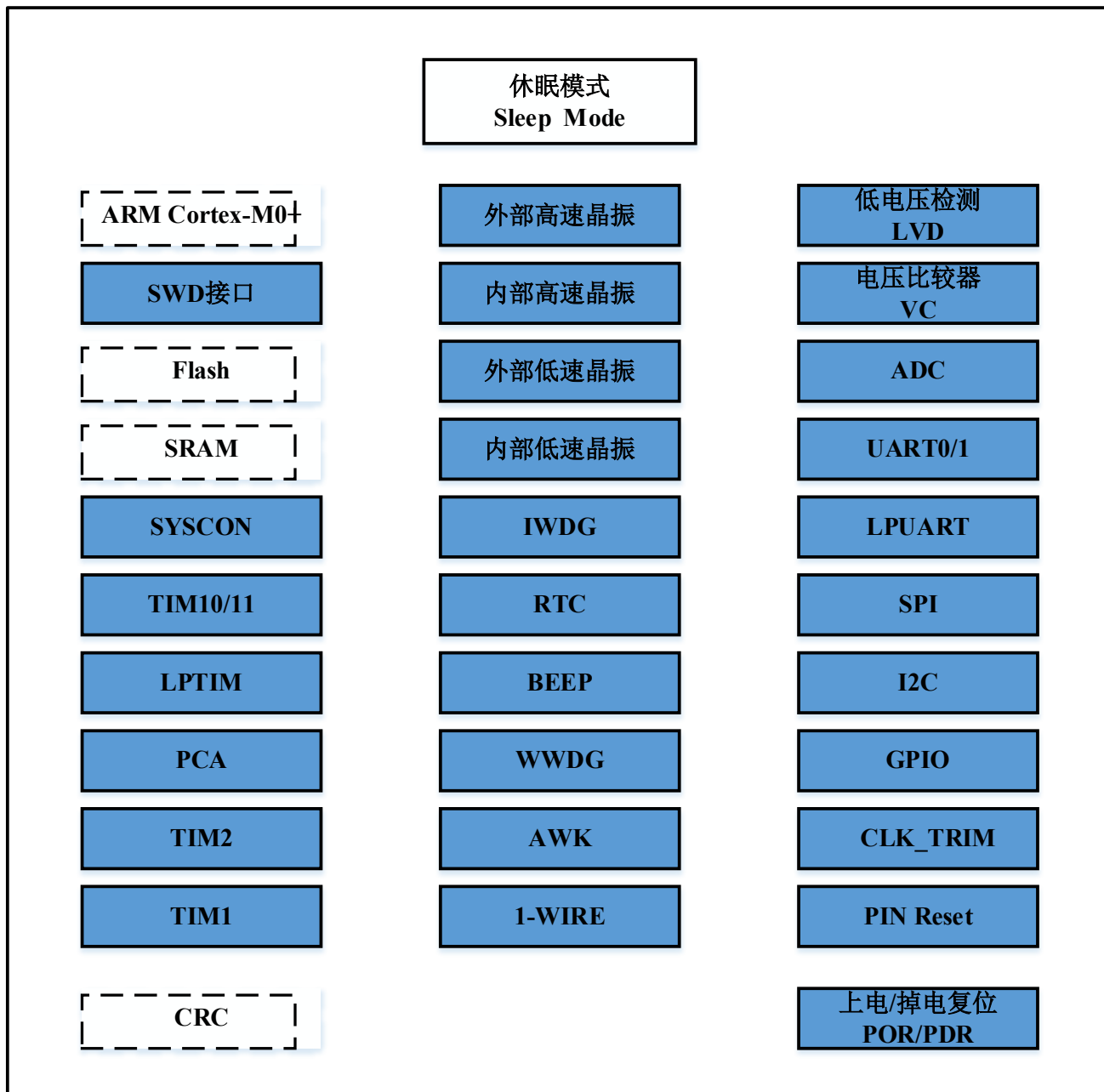


图 6-3. 休眠模式下可运行模块一览

6.3 深度休眠模式(Deep Sleep Mode)

使用 SleepDEEP 配合 WFI 指令可以进入深度休眠模式，在深度休眠模式下，CPU 停止运行，高速时钟关闭，低速时钟可配置是否运行，部分低功耗的周边模块可配置为是否运行，NVIC 中断处理仍可以工作。

- 系统从高速时钟进入深度休眠模式，高速时钟自动关闭，低速时钟保持进入深度睡眠前的状态。
- 系统从低速时钟进入深度休眠模式，低速时钟保持运行，除了低功耗模块可以运行，其他模块自动关闭。
- 系统时钟切换时，所有时钟都不会自动关闭，需要根据功耗及系统需求软件关闭打开相应的时钟。
- 系统进入深度休眠状态，不会改变端口状态，在进入深度休眠前根据需要更改 IO 的状态为深度休眠模式下的状态。

如何进入深度休眠模式：

首先设置 Cortex®-M0+系统控制寄存器中的 SleepDEEP 位，通过执行 WFI 指令进入深度睡眠状态。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深度睡眠模式进入机制：

Sleep-NOW：如果 SleepONEXIT=0，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=1，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。

如何退出深度休眠模式：

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(Deep Sleep 模式下可运行的周边模块中断)都能将系统从睡眠模式唤醒。唤醒设置参考 9.5 中断唤醒控制 WIC。

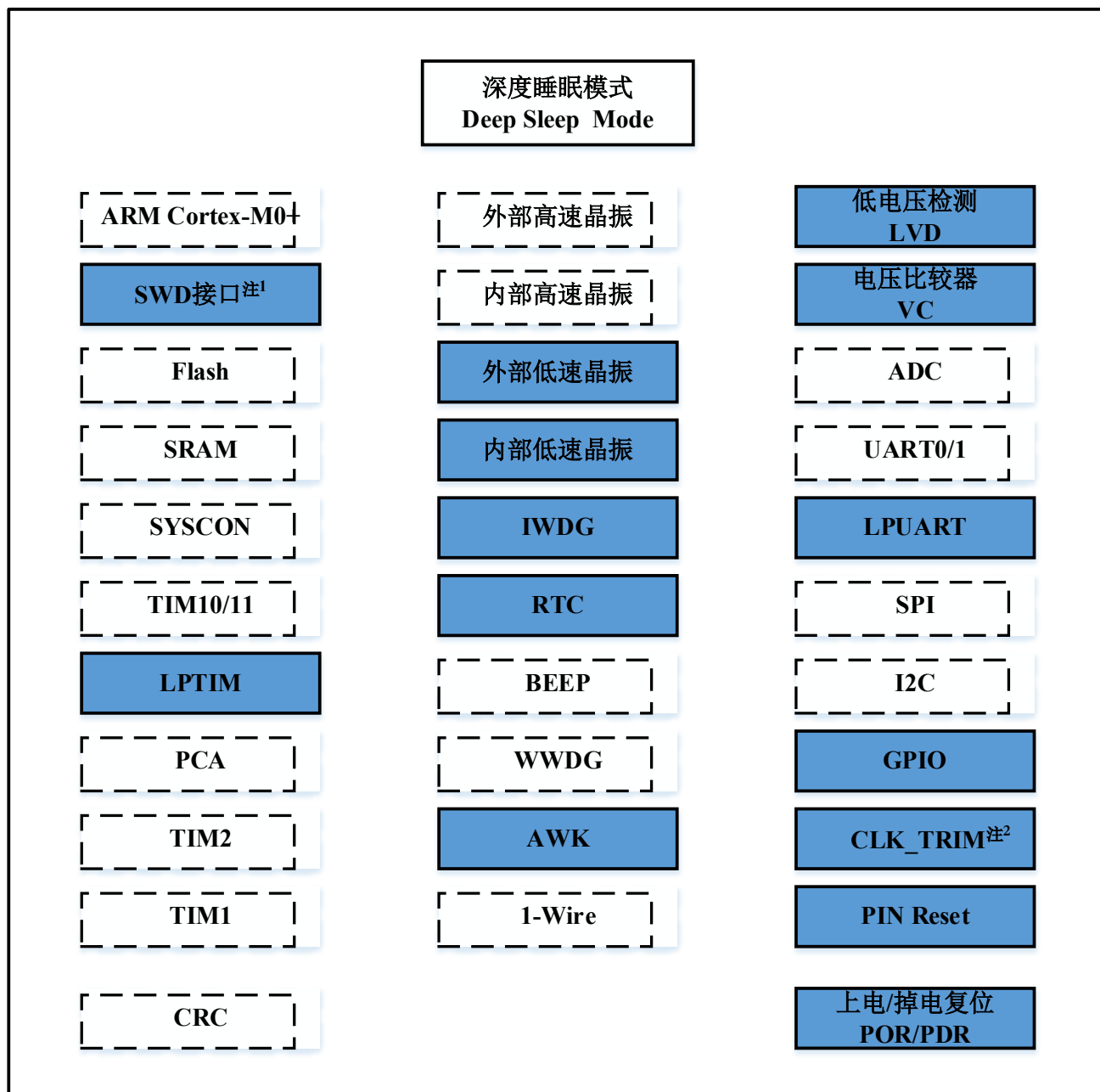


图 6-4. 深度睡眠模式下可运行模块一览

注 1: 在 Deep Sleep 模式，芯片重新复位后，可以通过 SWD 接口唤醒

注 2: 只有在选择内部低速监控外部低速时钟功能时才能唤醒

6.4 Cortex®-M0+内核系统控制寄存器(SCR)

地址: 0xE000ED10

复位值: 0x00000000

位	标记	功能	复位值	读写
31:5	RESERVED	保留	0x0	—
4	SEVONPEND	设置为 1 时，每次新的中断挂起都会产生一个事件，如果使用了 WFE 休眠，它可用于唤醒处理器	0	R/W
3	RESERVED	保留	0	—
2	SleepDEEP	设置为 1 时，执行 WFI 进入深度休眠，本产品进入 Deep Sleep 模式 设置为 0 时，执行 WFI 进入休眠，本产品进入 Sleep/Idle 模式	0	R/W
1	SleepONEXIT	设置为 1 时，当退出异常处理并返回程序线程时，处理器自动进入休眠模式(WFI) 设置为 0 时，该特性就会被自动禁止	0	R/W
0	RESERVED	保留	0	—

进入深度休眠后，唤醒后系统时钟有两种选择，默认使用进入深度休眠的时钟，配置寄存器 RCC_SYSCCLKCR.WKBYHIRC 为 1 后不管进入深度休眠前是什么时钟，唤醒后都使用内部高速时钟 HIRC。如果进入深度睡眠前系统使用外部晶体振荡这样设置可以加速系统唤醒。

7. 系统复位与时钟 (RCC)

7.1 复位

7.1.1 复位控制器介绍

本产品具有 9 个复位信号来源，每个复位信号都可以让 CPU 重新运行，绝大多数寄存器会被复位到复位值，程序计数器 PC 会被复位成复位地址。

- 数字区域上电掉电复位 POR
- 外部复位 PAD，低电平复位
- IWDG 复位
- WWDG 复位
- 低电压复位(LVD)
- 软件复位(Cortex®-M0+SYSRESETREQ)
- Cortex®-M0+LOCKUP 硬件复位
- 寄存器复位(CPURST)
- 寄存器复位(MCURST)

每个复位源都有一个专用的复位标志位来表示，由硬件置“1”，软件清除。除了数字区域的 POR 复位标志位，其他的复位标志位都可以被数字区域的 POR 来清除。

下图描述了每个复位的来源：

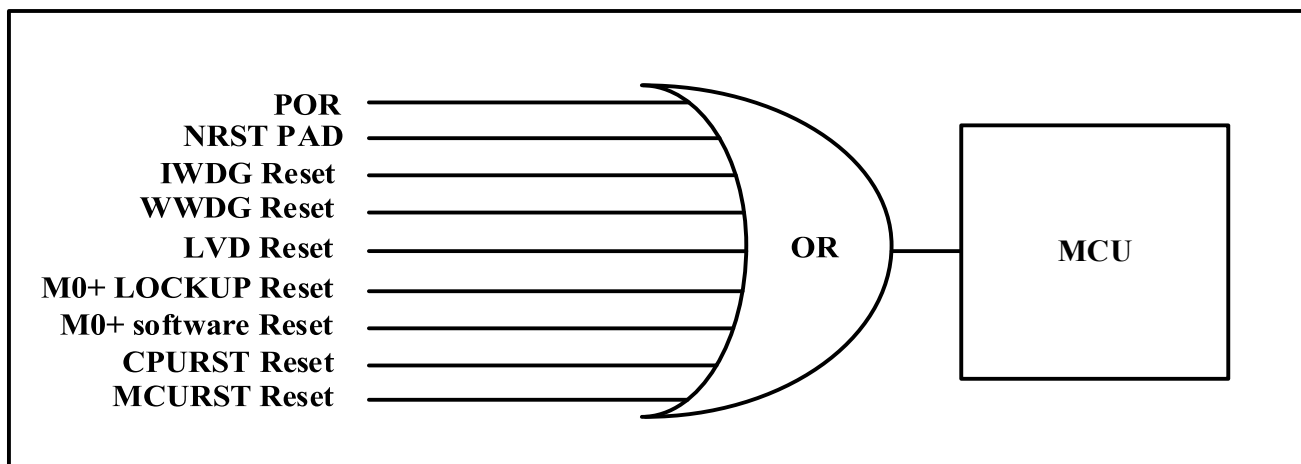


图 7-1. 复位来源示意图

7.1.2 复位源

7.1.2.1 上电/掉电复位 POR

当芯片上电以及掉电时，如果电源低于一个阈值电压(2.2V)，内部会产生一个 POR 信号，当电源高于该阈值电压时，释放 POR 信号。POR 信号会把芯片的 SFR、控制信号复位。本产品有两个电压区域，VDD 区域以及 Vcore 区域，所以有两个 POR：VDD 区域的 POR 以及 Vcore 区域的 POR。

7.1.2.2 外部引脚复位

把外部复位引脚拉到地电位就会产生一个系统复位。这个复位引脚内部带有上拉电阻，另外，在内部集成了一个毛刺滤波电路，系统会自动过滤小于 10uS(典型值)的毛刺信号，因此，用户在使用复位引脚产生复位信号时，必须产生大于 10uS 的低电平以保证芯片可以正确接收到复位信号。

7.1.2.3 IWDG 复位

独立看门狗复位，请参看 IWDG 一章说明。

7.1.2.4 WWDG 复位

窗口看门狗复位，请参看 WWDG 一章说明。

7.1.2.5 LVD 低电压复位

LVD 复位，请参考 LVD 一章说明。

7.1.2.6 Cortex®-M0+软件复位

通过将 Cortex®-M0+系统控制寄存器中的 SYSRESETREQ 位置 1，可实现软件复位。请参考 Cortex®-M0+技术参考手册获得进一步信息。

7.1.2.7 寄存器复位

可以通过写 RCC 里面的 RSTCON 寄存器的 MCURST 和 CPURST 来复位芯片。

7.1.2.8 Cortex®-M0+LOCKUP 硬件复位

当 Cortex®-M0+遇到严重的异常时，它会将自己的 PC 指针停在当前地址处，并锁死自己，并在几个时钟周期延时之后复位整个内核区域。

7.2 系统时钟

7.2.1 简介

时钟控制模块主要控制系统时钟以及外设时钟，可以配置不同的时钟源作为系统时钟、可以配置不同的系统时钟分频、可以启动或禁用外设时钟，另外为了确保高精度，内部时钟都具有校准功能。

本产品支持以下四个不同的时钟源作为系统时钟：

- 内部高速 RC 时钟 HIRC(4MHz)(默认主频)
- 外部低速晶振时钟 LXT(32.768KHz)
- 内部低速 RC 时钟 LIRC(38.4KHz 与 32.768KHz 可配置)
- 外部高速晶振时钟 HXT(4MHz~24MHz)
- LXT、HXT 可以通过端子 PB5、PA1 从外部输入。使用外部振荡输入时，需要使能相应的振荡。选择外部振荡控制选择在 RCC_SYSCCLKCR、RCC_LXTCCR 寄存器中。

每种时钟源都可以单独的打开或关断，当它们不用时，可以关断它们来降低功耗。有多个分频器可用于配置 AHB 和 APB 时钟域，AHB 和 APB 域的最大时钟频率为 24MHz。

Cortex 系统定时器(SysTick)由 AHB 时钟驱动，其可由 AHB/4 或 AHB 时钟频率直接驱动(通过 CortexSystick 配置位来配置)。RCC 通过 AHB 时钟(HCLK)的 4 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 时钟(HCLK)作为 SysTick 时钟。

需要特别注意的是：在切换系统时钟源的过程中需要按照正确的流程，具体流程参见“7.2.9 系统时钟切换”章节。

7.2.2 特性

- ◆ 一个频率为 4M~24MHz 的可配置高精度内部时钟 HIRC。在配置为 16MHz 时，从低功耗模式到工作模式的唤醒时间为 3 μ s，全电压全温度范围内的频率偏差 $\leq \pm 2.5\%$ ，无需外接昂贵的高速晶体。
- ◆ 一个频率为 4M~24MHz 的外部高速晶振 HXT。
- ◆ 一个频率为 32.768kHz 的外部低速晶振 LXT。
- ◆ 一个频率为 32.768kHz/38.4kHz 的内部低速时钟 LIRC。

7.2.3 系统时钟树

下图为本产品的时钟树图。

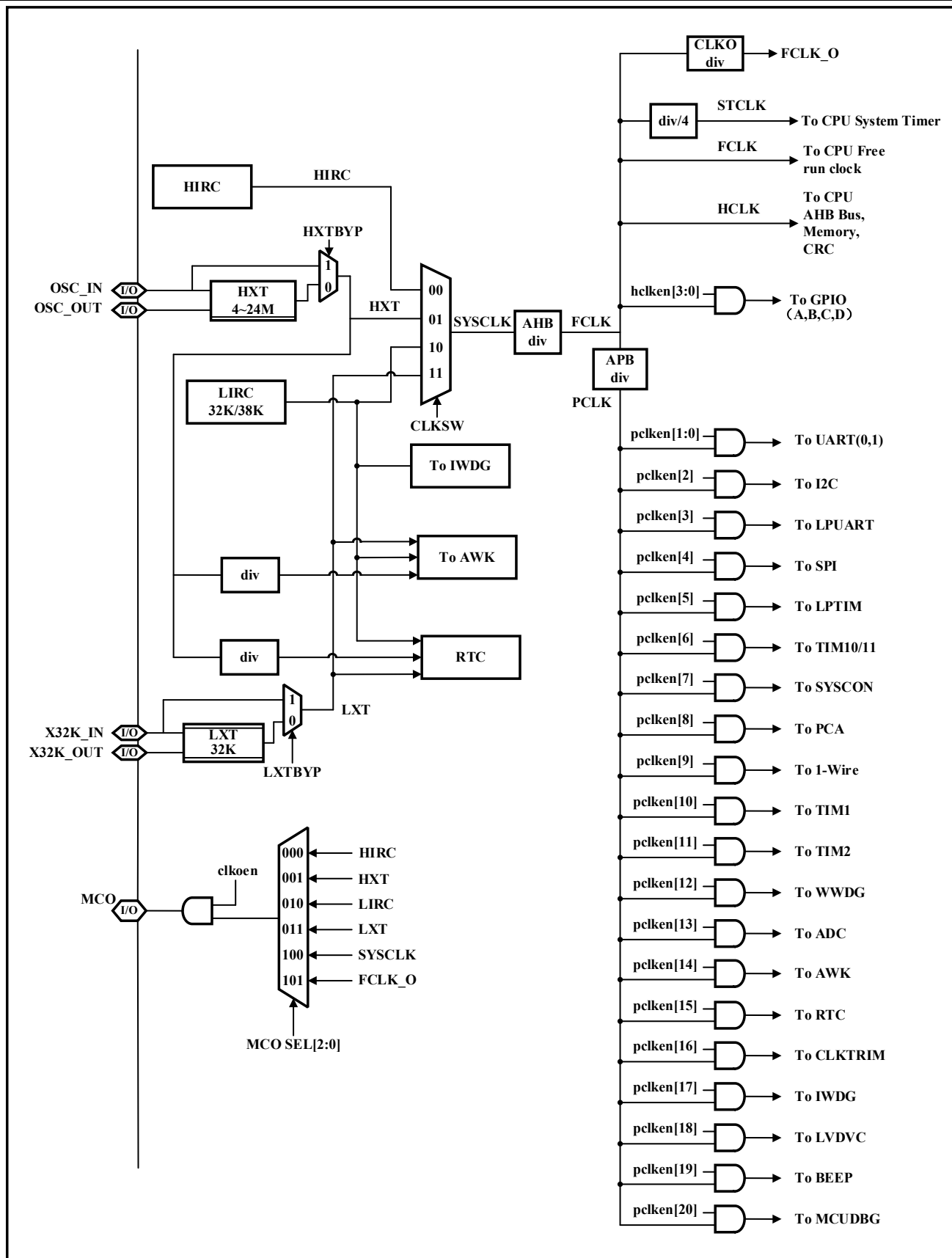


图 7-2. 时钟树结构图

7.2.4 内部高速 RC 时钟 HIRC

默认的系统时钟是内部高速时钟，在芯片上电或复位后即开始工作，通过寄存器 RCC_HIRCCR[10:0]来配置内部高速时钟的频率，给出精确 4M，8M，16M，22.12M，24M 的频率值。因为内部高速时钟启动快，约 $3\mu s$ ，为了让系统更为快速的响应外部中断，系统在从深度休眠模式被唤醒时，可以选择使用该时钟源作为系统时钟。

7.2.5 内部低速 RC 时钟 LIRC

内部低速时钟频率可配置成 38.4K、32.768K，在低速及对精度要求不高的应用场景下，可选择该时钟源作为系统时钟。

7.2.6 外部高速晶振时钟 HXT

外部 4M~24M 晶振时钟需根据用户系统需求外接一个 4M~24M 的高速晶振。

外部晶振时钟可以选择两种输入方式：

- HXT 外部晶体/陶瓷谐振器
- HXT 用户外部时钟

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

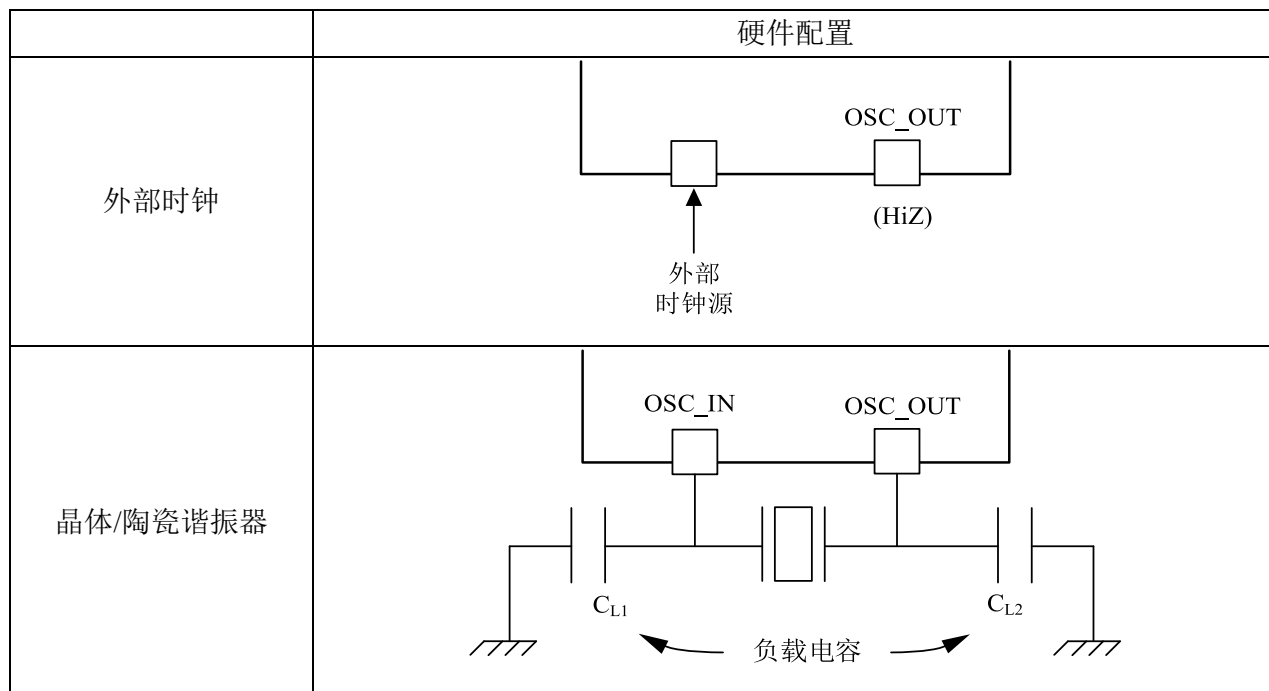


图 7-3. HXT/LXT 时钟源

7.2.7 外部低速晶振时钟 LXT

外部低速晶振时钟需外接一个 32.768K 的低功耗晶振，具有超高精度以及低功耗。超低功耗模式下工作的模块都可以选择此时钟源作为时钟信号。

外部时钟源旁路请参考图 7-3 HXT/LXT 时钟源。

7.2.8 系统时钟启动过程

为了增强系统的可靠性，每个时钟源使能后，硬件会等待一定的预热时间，然后才会把时钟输给系统使用。图 7-4 内部高速时钟启动示意图以内部高速 RC 时钟 HIRC 为例说明时钟的启动稳定过程。

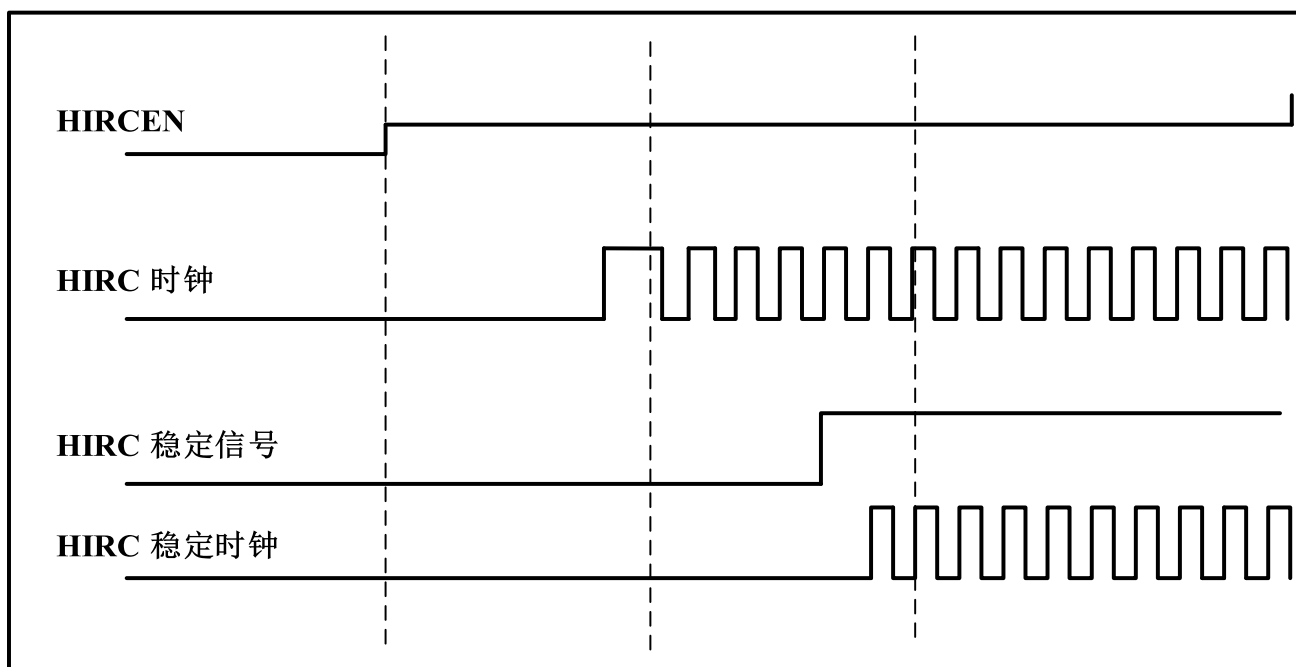


图 7-4. 内部高速时钟启动示意图

芯片上电后，系统使用 4MHz 的内部高速时钟作为启动时钟，启动完成后用户可以根据自己的需要来修改高速时钟的频率以及切换时钟源。

7.2.9 系统时钟切换

时钟源的切换是由寄存器 `RCC_SYSCLKSEL[3:0]` 来控制。在双时钟模式下，当系统时钟从当前时钟切换到目标时钟时，必须按照一定的流程来实现，否则就会出现异常。

7.2.9.1 内部高速切换到外部低速

以从 HIRC(内部高速时钟)切换到 LXT(外部 32.768K 晶振)为例，具体流程如下：

1)通过 RCC_LXTCR.LXTPORT 位来配置要切换的时钟 LXT 使用的引脚为模拟引脚，或者通过对于引脚的 GPIOx_AFR=0xF 来配置为模拟功能

2)写 RCC_LXTCR.LXTEN 使能 LXT 时钟

3)等待要切换的时钟 LXT 时钟稳定

4)写寄存器 RCC_SYSCCLKSEL.CLKSW[3:0]来切换时钟

5)根据需要关闭 HIRC 时钟

7.2.9.2 内部高速切换到外部高速

从 HIRC(内部高速时钟)切换到 HXT(外部 24M 晶振)为例，具体流程如下：

1)通过 RCC_SYSCCLKCR.HXTPORT 位来配置要切换的时钟 HXT 使用的引脚为模拟引脚，或者通过对于引脚的 GPIOx_AFR=0xF 来配置为模拟功能

2)写寄存器 RCC_SYSCCLKCR.HXTEN 位使能 HXT 时钟

3)等待要切换的时钟 HXT 时钟稳定

4)写寄存器 RCC_SYSCCLKSEL.CLKSW[3:0]来切换时钟，

5)根据需要关闭 HIRC 时钟

注意：使用外部高速 24M 晶振时，RCC_HXTCR.HXTSTARTUP 稳定时间控制位设置为 0x3，使用默认配置 0x2 稳定时间可能不够。

下图为 HIRC 到 HXT 切换时序：

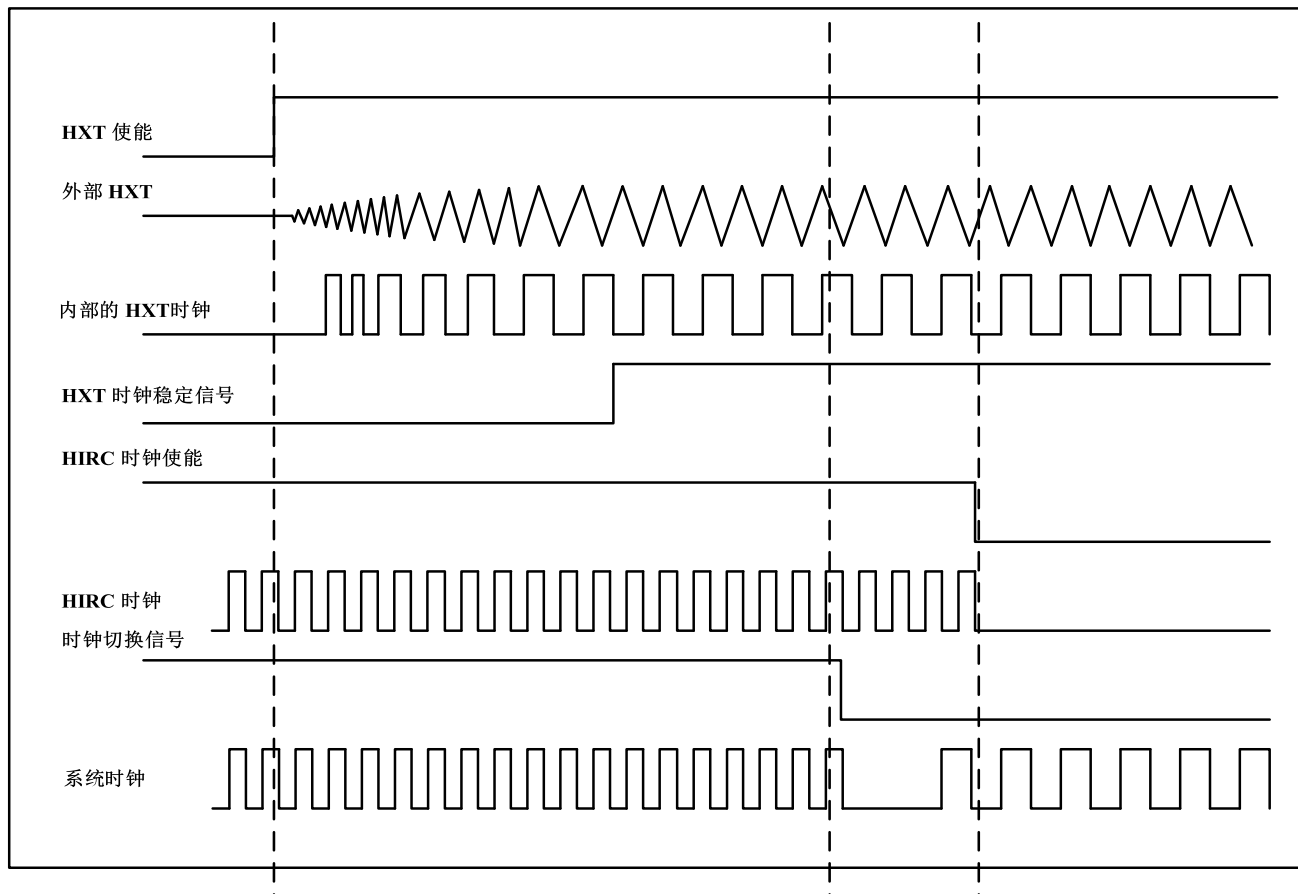


图 7-5. 时钟切换示意图

7.2.9.3 内部低速切换到外部高速

从 LIRC(内部低速时钟)切换到 HXT(外部 24M 晶振)为例，具体流程如下：

- 1)通过 `RCC_SYSCCLKCR.HXTPORT` 位来配置要切换的时钟 HXT 使用的引脚为模拟引脚，或者通过对于引脚的 `GPIOx_AFR=0xF` 来配置为模拟功能
- 2)写寄存器 `RCC_SYSCCLKCR.HXTEN` 位使能 HXT 时钟
- 3)等待要切换的时钟 HXT 时钟稳定
- 4)写寄存器 `RCC_SYSCCLKSEL.CLKSW[3:0]`来切换时钟
- 5)根据需要关闭 LIRC 时钟

7.2.10 系统时钟输出

微控制器允许输出时钟信号到外部 MCO 引脚。

有如下的 6 种信号可选为 MCO 时钟输出：

- HIRC
- HXT
- LIRC
- LXT
- SYSCLK
- FCLK 以及分频输出

MCO 时钟的选择由时钟输出控制寄存器(RCC_MCOCR)的 MCOSEL[2:0]位决定。

7.2.11 系统时钟安全控制

当设定 CLKFAILEN 有效后，并且使能 CLKTRIM 的时钟监视功能后，当 HXT 时钟 LXT 时钟停止后，系统时钟会切换到内部高速时钟。

具体请参考 CLKTRIM 模块的监测功能。

7.2.12 IWDG 时钟

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

7.2.13 RTC 时钟

RTCCLK 时钟源可以由 HSE 分频、LXT 或 LIRC 时钟提供。

7.2.14 AWK 时钟

AWKCLK 时钟源可以由 HSE 分频、LXT 或 LIRC 时钟提供。

7.2.15 低功耗模式

APB 外设时钟以及部分 AHB 外设时钟可以用软件禁止。睡眠模式停止 CPU 时钟，在 CPU 睡眠中存储器接口时钟(Flash 和 RAM 接口)被停止。

当配置了 SYSCON_CFGR0.DBGDLSP_DIS 后，那么 CPU 在相应的深度睡眠模式下也可以具有调试功能。

7.3 寄存器列表

这节详细描述了 RCC 控制模块的寄存器功能。

RCC 基址: 0x4002_0000

表 7.1. RCC 寄存器列表和复位值

偏移地址	名称	描述	复位值
0x00	RCC_HCLKDIV	AHB 时钟分频寄存器	0x00000000
0x04	RCC_PCLKDIV	APB 时钟分频寄存器	0x00000000
0x08	RCC_HCLKEN	AHB 周边模块时钟使能寄存器	0x00000100
0x0C	RCC_PCLKEN	APB 周边模块时钟使能寄存器	0x00000000
0x10	RCC_MCOCR	时钟输出控制寄存器	0x00000000
0x18	RCC_RSTCR	系统 Reset 控制寄存器	0x00000000
0x1C	RCC_RSTSR	Reset 状态寄存器	0x000000A0
0x20	RCC_SYSCLKCR	时钟源设置寄存器	0x00000001
0x24	RCC_SYSCLKSEL	系统时钟源选择寄存器	0x00000001
0x28	RCC_HIRCCR	内部高速 RC 振荡器控制寄存器	0x00001312
0x2C	RCC_HXTCR	外部高速晶体振荡器控制寄存器	0x00000027
0x30	RCC_LIRCCR	内部低速 RC 振荡器控制寄存器	0x0000007F
0x34	RCC_LXTCR	外部低速晶体振荡器控制寄存器	0x0000042F
0x38	RCC_IRQLATENCY	M0 IRQ 延时控制	0x00000000
0x3C	RCC_STICKCR	SysTick Timer 周期校准寄存器	0x01009C3F
0x40	RCC_SWDIOCR	端子特殊功能选择寄存器	0x00000001
0x44	RCC_PERIRST	周边模块复位控制寄存器	0x00000000
0x48	RCC_RTCRST	RTC 控制寄存器	0x00000000
0x60	RCC_UNLOCK	寄存器写保护	0x00000000

7.4 寄存器说明

7.4.1 AHB 时钟分频寄存器(RCC_HCLKDIV)

地址偏移: 0x00

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7:0	AHBCKDIV[7:0]	系统 HCLK 时钟分频 0: HCLK=SYSCLK 1~255: Divide by 2×DIV (HCLK= SYSCLK/ (2×AHBCKDIV)).	0x0	R/W

7.4.2 APB 时钟分频寄存器(RCC_PCLKDIV)

地址偏移: 0x04

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7:0	APBCKDIV[7:0]	系统 PCLK 时钟分频(大 1/16) 0: PCLK=HCLK 1~255: Divide by 2×DIV(PCLK=HCLK/(2×APBCKDIV)).	0x0	R/W

7.4.3 AHB 周边模块时钟使能寄存器(RCC_HCLKEN)

地址偏移: 0x08

复位值: 0x00000100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							Flash CKEN	保留			CRC CKEN	GPIO DCKEN	GPIO CCKEN	GPIO BCKEN	GPIO ACKEN
							R/W				R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:9	—	保留	0x0	—
8	FlashCKEN	Flash 控制器模块时钟使能。关闭后 Flash 配置寄存器不可写，Flash 中的程序仍然可以运行。 0: 关闭 1: 使能	1	R/W
7:5	—	保留	0	R/W
4	CRCKEN	CRC 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
3	GPIOCKEN	GPIO 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
2	GPIOCCKEN	GPIOC 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
1	GPIOBCKEN	GPIOB 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
0	GPIOACKEN	GPIOA 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W

7.4.4 APB 周边模块时钟使能寄存器(RCC_PCLKEN)

地址偏移: 0x0C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											DBG CKE N	BEEP CKEN	LVDV CCKE N	IWDG CKEN	CLK TRIM CKEN
											R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC CKE N	AW KCK EN	ADC CKE N	WW DGC KEN	TIM2 CKEN	TIM 1CK EN	OWI REC KEN	PCA CKE N	SYS CON CKE N	BAS ETI MCK EN	LPTI MCK EN	SPI CKE N	LPUA RTCK EN	I2C CKE N	UART 0CKE N	UART 0CKE N
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:21	—	保留	0x0	—
20	DBGCKEN	Debug PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
19	BEEPCKEN	BEEP PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
18	LVDVCCKEN	LVD/VC PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
17	IWDGCKEN	IWDG PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
16	CLKTRIMCKEN	CLKTRIM PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
15	RTCKEN	RTC PCLK 时钟使能 1: 时钟使能 0: 时钟关闭	0	R/W
14	AWKCKEN	RTC PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
13	ADCCKEN	ADC PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
12	WWDGCKEN	WWDG PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
11	TIM2CKEN	TIM2 PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
10	TIM1CKEN	TIM1 PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W

位	标记	功能描述	复位值	读写
9	OWIRECKEN	1-WIRE PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
8	PCACKEN	PCA PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
7	SYSCONCKEN	SYSCON PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
6	BASETIMCKEN	TIM10/11 PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
5	LPTIMCKEN	Low Power Timer PCLK 时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
4	SPICKEN	SPI PCLK 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
3	LPUARTCKEN	Low Power UART PCLK 寄存器配置时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
2	I2CCKEN	I2C PCLK 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W
1	UART1CKEN	UART1 PCLK 模块时钟使能 1: 时钟使能 0: 时钟关闭	0	R/W
0	UART0CKEN	UART0 PCLK 模块时钟使能 0: 时钟关闭 1: 时钟使能	0	R/W

7.4.5 时钟输出控制寄存器(RCC_MCOCR)

地址偏移: 0x10

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:13	—	保留	0x0	—
12	MCOEN	MCO 输出使能写: 0: MCO 输出禁止 1: MCO 输出使能读: 0: MCO 未开始输出 1: MCO 开始输出 注意, 读该位时是输出使能信号通过输出时钟同步后的信号。	0	R/W
11	—	保留	0	—
10:8	MCOSEL[2:0]	时钟输出源选择 000: HIRC 001: HXT 010: LIRC 011: LXT 100: SYSCLK 101: FCLK 以及分频输出 110,111: Reserved, 设定禁止	0x0	R/W
7:0	MCODIV[7:0]	FCLK 时钟分频系数 0: FCLK 1~255: Divide by 2×DIV(HCLK=FCLK/(2×DIV)).	0x0	R/W

7.4.6 系统复位控制寄存器(RCC_RSTCR)

地址偏移: 0x18

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSTKEY															
WO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSTKEY													CPURST	MCURST	
WO													R/W	R/W	

位	标记	功能描述	复位值	读写
31:2	RSTKEY	必须在写 RSTCON[1:0]时写入 0x156A99A6 才有效。写其他值时都无效。	0x0	WO
1	CPURST	寄存器 CPU 复位, 该复位发生时, 不会重新装载用户配置区中的 ISP 设定 0: Normal	0	R/W

位	标记	功能描述	复位值	读写
		1: Reset CPU		
0	MCURST	寄存器 MCU 复位, 该复位发生时, 会重新装载用户配置区中的 ISP 设定 0: Normal 1: Reset MCU	0	R/W

Note:

MCU reset by write 0x55AA6699 to RSTCON;

CPU reset by write 0x55AA669A to RSTCON.

注意, 只有 **REGLOCK** 寄存器保护解除后, 才能写该寄存器。

7.4.7 系统复位状态寄存器(RCC_RSTSR)

地址偏移: 0x1C

复位值: 0x000000A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							SFT RST	PAD RST	LOCK UPRST	POR RST	LVD RST	IWDG RST	WWDG RST	CPU RST	MCU RST
							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:9	—	保留	0x0	—
8	SFTRST	Cortex-M0+CPU 软件复位标志, 需要软件清除 0: 无 Cortex-M0+CPU 软件复位发生 1: 发生 Cortex-M0+CPU 软件复位	0	R/W
7	PADRST	RESET 端口复位标志, 能够被 POR 复位 0: 无端口复位发生 1: 发生端口复位	1	R/W
6	LOCKUPRST	Cortex-M0+CPU Lockup 复位标志 0: 无 Cortex-M0+CPU Lockup 复位发生 1: 发生 Cortex-M0+CPU Lockup 复位	0	R/W
5	PORRST	Vcore 域 POR 复位标志 0: Vcore 域 POR 无复位发生 1: Vcore 域 POR 发生复位	1	R/W
4	LVD RST	LVD 复位标志 0: LVD 无复位发生	0	R/W

位	标记	功能描述	复位值	读写
		1: LVD 发生复位		
3	IWDGRST	IWDG 复位标志 0: IWDG 无复位发生 1: IWDG 发生复位	0	R/W
2	WWDGRST	WWDG 复位标志 0: WWDG 无复位发 1: WWDG 发生复位	0	R/W
1	CPURST	寄存器 CPU 复位标志, 该复位发生时, 不会重新装载 USER OPTION BYTE 中的 ISP 和 IAP 设定 0: 寄存器 CPURST 无复位发生 1: 寄存器 CPURST 复位发生	0	R/W
0	MCURST	寄存器 MCU 复位标志, 该复位发生时, 会重新装载 USER OPTION BYTE 中的 ISP 和 IAP 设定 0: 寄存器 MCURST 无复位发生 1: 寄存器 MCURST 复位发生	0	R/W

注: 只受 **POR** 控制。

7.4.8 系统时钟源配置寄存器(RCC_SYSCLKCR)

地址偏移: 0x20

复位值: 0x00000001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY															
WO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WK BY HIR C	保留						CLK FAIL EN	保留	HXT POR T	HXT	保留		LI RC EN	CPU RST	MCU RST
R/W							R/W		R/W	R/W			R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效, 写其它值时无效。	0x0	WO
15	WKBYHIRC	0: 从 Deep Sleep 唤醒, system clock 来源怎么进怎么出 1: 从 Deep Sleep 唤醒是使用 HIRC 开始唤醒, 硬件自动 enable HIRC, 并且 system clock 自动切换到 HIRC, 原时钟继续开启	0	R/W
14:9	—	保留	0	—
8	CLKFAIL EN	时钟失效检测使能控制 0: 时钟失效检测禁止 1: 时钟失效检测使能, 当检测到允许时钟失效, 自动切换系统时钟到 HIRC	0	R/W

位	标记	功能描述	复位值	读写
7	—	保留	0	—
6	HXTPORT	OSCIN/OSCOUT 端子配置 0: GPIO 功能模式(数字功能) 1: HXT 端子模式(模拟功能)	0	R/W
5	HXTBYP	外部高速时钟输入选择 0: HXT 内部振荡模块未被旁路, 与 OSC_IN/OSC_OUT 相连 1: HXT 内部振荡模块被旁路, HXT 从端子 OSCIN 直接输入	0	R/W
4:3	—	保留	0x0	—
2	LIRCEN	内部低速时钟 LIRC 使能信号 0: 关闭 1: 使能 当系统时钟选择该时钟时, 不能关闭。	0	R/W
1	HXTEN	外部 4M~24M 晶振 HXTOSC 使能信号 0: 关闭 1: 使能 注意: 1.使用时, 与该晶振连接的两个外部端口需要设置成模拟端口; 当系统进入 Deep Sleep, 此高速时钟会自动关闭。 2.当 HXT 停止检出时, 此位会被硬件清 0 当系统时钟选择该时钟时, 不能关闭。	0	R/W
0	HIRCEN	内部高速时钟 HIRC 使能信号 0: 关闭 1: 使能 注意: 1.当系统进入 deep Sleep, 此高速时钟会自动关闭。 2.当 HXT 停止检出时, 如果系统时钟选择为 HXT, 且 CLKFAIL_EN 使能, HIRC_EN 会由硬件自动置 1。 3.当系统时钟选择该时钟时, 不能关闭。	1	R/W

注意: 只有 REGLOCK 寄存器保护解除后, 才能写该寄存器。

7.4.9 系统时钟源选择寄存器(RCC_SYSCLKSEL)

地址偏移: 0x24

复位值: 0x00000001

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

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:4	—	保留	0x0	—
3:0	CLKSW[3:0]	System Clock Source Select. 0001: HIRC 选择 0010: HXT 选择 0100: LIRC 选择 1000: LXT 选择 注意: 当 HXT 停止检出时, 如果系统时钟选择为 HXT, 且 CLKFAIL_EN 使能, HIRC_EN 会由硬件自动置 1。系统时钟自动选择 HIRC。	0x1	R/W

注意: 该位写受 REGLOCK 保护。

7.4.10 内部高速 RC 振荡器控制寄存器(RCC_HIRCCR)

地址偏移: 0x28

复位值: 0x00001312

该寄存器不受 Cortex-M0+软件复位控制。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY															
WO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			HIRC RDY	HIRCTRIM[11:0]											
			RO	R/W											

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:13	—	保留	0x0	—
12	HIRC RDY	HIRC 时钟稳定标志位。 0: 代表 HIRC 未稳定, 不可以被内部电路使用 1: 代表 HIRC 已经稳定, 可以被内部电路使用	1	RO
11:0	HIRC TRIM [11:0]	主频时钟频率调整(process variation) 注意: 出厂时, 频率调整值保存在 Flash 中, 用户需要将 Flash 值写入 HIRC_CR.TRIM 即可配置精准主频时钟。 24M 校准值地址: 0X1800_00A0 22.12M 校准值地址: 0X1800_00A2 16M 校准值地址: 0X1800_00A4 8M 校准值地址: 0X1800_00A6 4M 校准值地址: 0X1800_00A8	0x312	R/W

位	标记	功能描述	复位值	读写																
		<div>[11:9]: 为档位控制位<table><tr><td>000</td><td>禁止设定</td></tr><tr><td>001</td><td>4Mhz</td></tr><tr><td>010</td><td>8Mhz</td></tr><tr><td>011</td><td>12Mhz</td></tr><tr><td>100</td><td>16Mhz</td></tr><tr><td>101</td><td>20Mhz</td></tr><tr><td>110</td><td>24Mhz</td></tr><tr><td>111</td><td>禁止设定</td></tr></table></div> <div>[8:0]: 为 trim 细调位 0x000: 频率最低 0x1FF: 频率最高</div>	000	禁止设定	001	4Mhz	010	8Mhz	011	12Mhz	100	16Mhz	101	20Mhz	110	24Mhz	111	禁止设定		
000	禁止设定																			
001	4Mhz																			
010	8Mhz																			
011	12Mhz																			
100	16Mhz																			
101	20Mhz																			
110	24Mhz																			
111	禁止设定																			

注意，只有 **REGLOCK** 寄存器保护解除后，才能写该寄存器。

7.4.11 外部高速晶体振荡器控制寄存器(RCC_HXTCR)

地址偏移: 0x2C

复位值: 0x00000027

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY															
WO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									HXTRDY	HXT STARTUP[1:0]		保留	HXTDRV[2:0]		
									RO	R/W			R/W		

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:7	—	保留	0x0	—
6	HXTRDY	外部 4M~24M 晶振稳定标志位 0: 代表外部高速晶振时钟未稳定，不可以被内部电路使用 1: 代表外部高速晶振时钟已经稳定，可以被内部电路使用	0	RO
5:4	HXTSTARTUP[1:0]	外部 4M~32M 晶振稳定时间选择 00: 1024 个周期 01: 2048 个周期 10: 4096 个周期 11: 16384 个周期 使用高速晶振时钟时稳定时间需要设置为 11，否则稳定时间不够，可能导致系统时钟切换时或使用高速晶振时钟深度休眠唤醒时导致系统不稳定。	0x2	R/W

位	标记	功能描述	复位值	读写
3	—	保留	0	—
2:0	HXTDRV[2:0]	外部 4M~32M 晶振驱动选择 000: 最小驱动 111: 最大驱动(推荐值)	0x7	R/W

注意，只有 **REGLOCK** 寄存器保护解除后，才能写该寄存器。

7.4.12 内部低速 RC 振荡器控制寄存器(RCC_LIRCCR)

地址偏移: 0x30

复位值: 0x0000007F

除 LVD, CorTex-M0+软件复位以外的复位控制。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY															
WO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			LIRCRDY	LIRCSTARTUP[1:0]		保留	LIRCTRIM[8:0]								
			RO	R/W			R/W								

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:13	—	保留	0x0	—
12	LIRCRDY	内部低速时钟稳定标志位 0: 代表内部低速未稳定，不可以被内部电路使用 1: 代表内部低速已经稳定，可以被内部电路使用	0	RO
11:10	LIRCSTARTUP[1:0]	内部低速时钟稳定时间选择 00: 4 个周期 01: 16 个周期 10: 64 个周期 11: 256 个周期	0x0	R/W
9	—	保留	0	—
8:0	LIRCTRIM[8:0]	内部低速时钟频率调整。注意：出厂时，频率调整值保存在 Flash 中，用户需要将 Flash 值写入 RCL_CR.TRIM 即可配置经过校准过的 38.4K/32.768K 内部低速时钟。(process variation) 32.768K 校准值地址: 0X1800_00D0 38.4K 校准值地址: 0X1800_00D4	0x07F	R/W

注意，只有 **REGLOCK** 寄存器保护解除后，才能写该寄存器。

7.4.13 外部低速晶体振荡器控制寄存器(RCC_LXTCR)

地址偏移: 0x34

复位值: 0x0000042F

这个寄存器在 RTC 域，只有 POR 才能 reset 掉这个寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY															
WO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				LXT PORT	LXT AON	LXT BYP	LXT EN	保留	LXT RDY	LXTSTART UP[1:0]		LXTDRV[3:0]			
				R/W	R/W	R/W	R/W		RO	R/W		R/W			

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:12	—	保留	0x0	—
11	LXTPORT	X32K_IN/X32K_OUT function selection 0: GPIO 功能(数字功能) 1: X32K 端子模式(模拟功能) 注: 该位写受 REGLOCK 保护	0	R/W
10	LXTAON	LXT 只能使能不能禁止控制 0: LXT_EN 允许禁止控制 1: LXT_EN 只能使能不能禁止控制	1	R/W
9	LXTBYP	由软件设置和清除，该位仅在外外部 32KHz 振荡器关闭的情况下写值。 0: LSE 振荡器未被旁路 1: LSE 振荡器被旁路 注: 使用外部低速振荡时需要使能低速晶体振荡的使能位 LXT_EN。	0	R/W
8	LXTEN	外部 32K 晶振 LXT 使能信号 0: 关闭 1: 使能 当系统时钟选择该时钟时，不能关闭。	0	R/W
7	—	保留	0	—
6	LXTRDY	外部 32K 晶振稳定标志位 0: 代表外部 32K 晶振时钟未稳定，不可以被内部电路使用 1: 代表外部 32K 晶振时钟已经稳定，可以被内部电路使用	0	RO
5:4	LXTSTARTUP[1:0]	外部 32.768K 晶振稳定时间选择 00: 1024 个周期 01: 2048 个周期 10: 4096 个周期	0x2	R/W

位	标记	功能描述	复位值	读写
		11: 16384 个周期 使用高速晶振时钟时稳定时间需要设置为 11，否则稳定时间不够，可能导致系统时钟切换时或使用高速晶振时钟深度休眠唤醒时导致系统不稳定。		
3:0	LXTDRV [3:0]	外部 32K 晶振驱动选择。 0000: 最小驱动 1111: 最大驱动	0xF	R/W

注意，只有 **REGLOCK** 寄存器保护解除后，才能写该寄存器。

7.4.14 M0 IRQ 延时控制寄存器(RCC_IRQLATENCY)

地址偏移: 0x38

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7:0	IRQLATENCY[7:0]	IRQLATENCY[7:0]	0x0	R/W

注意，只有 **REGLOCK** 寄存器保护解除后，才能写该寄存器。

7.4.15 System Tick timer 控制寄存器(RCC_STICKCR)

地址偏移: 0x3C

复位值: 0x01009C3F

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

位	标记	功能描述	复位值	读写
31:26	—	保留	0x0	—
25	NOREF	Systick 定时器是否使用外部参考时钟 0: HCLK/4 1: 使用内核时钟 注意: 1) 与系统寄存器 SysTick->CTRL.CLKSOURCE (0XE000E010)任意一个设置为 1 后使用内核时钟 2) 使用参考时钟 HCLK/4 作为 systick 时钟时, 参考时钟频率不允许高于系统时钟 HCLK	0	R/W
24	SKEW	10 ms STCALIB 值是否准确 0: 准确 1: 不准确	1	R/W
23:0	STCALIB[23:0]	Systick 10ms 校准值, 此值为使用外部参考时钟 HCLK/4(4MHZ)的 10ms 校准值。	0x009C3F	R/W

7.4.16 SWDIO 端口控制寄存器(RCC_SWDIOCR)

地址偏移: 0x40

复位值: 0x00000001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY															
WO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														SWDPORT	
														R/W	

位	标记	功能描述	复位值	读写
31:16	—	只有高位写 0x5A69 时配置该寄存器才有效, 写其它值时无效。	0x0	WO
15:1	—	保留	0x0	
0	SWDPORT	配置 PC7 和 PD1 的端子功能模式 0: 周边模块功能模式 1: SWD 端子功能	1	R/W

注意: 该位写受 REGLOCK 保护。

7.4.17 周边模块复位控制寄存器(RCC_PERIRST)

地址偏移: 0x44

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			CRC RST	GPI ODR ST	GPI OC RST	GPI O B RST	GPI O A RST	保留			DBG RST	BEEP RST	LVD VC RST	保留	CLKT RIM RST
			R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W		R/W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	AW K RST	ADC RST	WW DG RST	TIM 2 RST	TIM 1 RST	OWR IE RST	PCA RST	SYS CON RST	BASE TIM RST	LP TIM RST	SPI RST	LPU ART RST	I2C RST	UAR T1 RST	UART0 RST
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:29	—	保留	0x0	—
28	CRCRST	CRC 模块复位 0: 正常工作 1: 复位	0	R/W
27	GPIODRST	GPIOD 模块复位 0: 正常工作 1: 复位	0	R/W
26	GPIOCRST	GPIOC 模块复位 0: 正常工作 1: 复位	0	R/W
25	GPIOBRST	GPIOB 模块复位 0: 正常工作 1: 复位	0	R/W
24	GPIOARST	GPIOA 模块复位 0: 正常工作 1: 复位	0	R/W
23:21	—	保留	0x0	—
20	DBG RST	MCU DEBUG 模块复位 0: 正常工作 1: 复位	0	R/W
19	BEEP RST	BEEP 模块复位 0: 正常工作 1: 复位	0	R/W
18	LVDVCRST	LVD 模块复位 0: 正常工作 1: 复位	0	R/W
17	—	保留	0	—
16	CLKTRIMRST	Clock TRIM 模块复位 0: 正常工作 1: 复位	0	R/W
15	—	保留	0	—
14	AWKRST	AWK 模块复位	0	R/W

位	标记	功能描述	复位值	读写
		0: 正常工作 1: 复位		
13	ADCRST	ADC 模块复位 0: 正常工作 1: 复位	0	R/W
12	WWDGRST	WWDG 模块复位 0: 正常工作 1: 复位	0	R/W
11	TIM2RST	TIM2 模块复位 0: 正常工作 1: 复位	0	R/W
10	TIM1RST	TIM1 模块复位 0: 正常工作 1: 复位	0	R/W
9	OWIREST	1-Wire 模块复位 0: 正常工作 1: 复位	0	R/W
8	PCARST	PCA 模块复位 0: 正常工作 1: 复位	0	R/W
7	SYSCONRST	SYSCON 模块复位 1: 复位 0: 正常工作	0	R/W
6	BASETIMRST	Base Timer 10/11 复位 0: 正常工作 1: 复位	0	R/W
5	LPTIMRST	Low Power Timer 0/1 复位 1: 复位 0: 正常工作	0	R/W
4	SPIRST	SPI 模块复位 0: 正常工作 1: 复位	0	R/W
3	LPUARTRST	LPUART 模块复位 0: 正常工作 1: 复位	0	R/W
2	I2CRST	I2C 模块复位 0: 正常工作 1: 复位	0	R/W
1	UART1RST	UART1 模块复位 0: 正常工作 1: 复位	0	R/W
0	UART0RST	UART0 模块复位 0: 正常工作 1: 复位	0	R/W

注意，只有 **REGLOCK** 寄存器保护解除后，才能写该寄存器。

7.4.18 RTC 复位控制寄存器(RCC_RTCCRST)

地址偏移：0x48

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY															
WO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														RTC RST	
														R/W	

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效，写其它值时无效。	0x0	WO
15:5	—	保留，必须保持复位值	0x0	—
0	RTCCRST	RTC 模块复位 由软件置 1 或清 0。 0：复位未激活 1：复位整个 RTC	0	R/W

注意，只有 **REGLOCK** 寄存器保护解除后，才能写该寄存器。

7.4.19 寄存器写保护控制寄存器(RCC_UNLOCK)

地址偏移：0x60

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:1	KEY	只有高位写 0x2AD5334C 时配置该寄存器才有效，写其它值时无效。	0x0	WO
0	UNLOCK	0：寄存器写保护启用，不能对受保护的寄存器写操作	0	R/W



位	标记	功能描述	复位值	读写
		1: 寄存器写保护禁止, 可以对受保护的寄存器写操作		

写入 0X55AA6699, 解除保护。

8. 系统控制(SYSCON)

本产品具有一组系统配置寄存器。系统配置控制器的主要用途如下：

- 配置 GPIO 端子的中断产生模式
- 重映射 TIM10/11, PCA, TIM1, TIM2 的输入触发源
- SPI 从模式的 CS 输入重映射设定
- 系统级的 Deep Sleep 调试以及 Lockup 复位控制设定

8.1 寄存器列表

SYSCON 基址：0x4000 1C00

表 8.1. SYSCON 寄存器列表和复位值

偏移地址	名称	描述	复位值
0x00	SYSCON_CFGR0	系统配置寄存器 0	0x00000000
0x04	SYSCON_PORTINTCR	端子的中断模式设定	0x00000000
0x08	SYSCON_PORTCR	端子控制寄存器	0x00000000
0x0C	SYSCON_PCACR	PCA 捕获通道来源选择	0x00000000
0x10	SYSCON_TIM1CR	TIM1 通道输入源选择	0x00000000
0x14	SYSCON_TIM2CR	TIM2 通道输入源选择	0x00000000
0x50	SYSCON_UNLOCK	Syscon 寄存器写保护	0x00000000

8.2 寄存器说明

8.2.1 系统配置寄存器 0(SYSCON_CFGR0)

地址偏移：0x00

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY															
WO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													DBG DLSP DIS	LOCK UP	
													R/W	R/W	

位	标记	功能描述	复位值	读写
31:16	Key	只有高位写 0x5A69 时配置该寄存器才有效, 写其它值时无效。	0x0	WO
15:2	—	保留	0x0	—
1	DBGDLSP_DIS	Debug 模式, 进入 Deep Sleep 模式禁止控制位 0: 允许在 Debug 模式进入 Deep Sleep 1: 不允许在 Debug 模式进入 Deep Sleep	0	R/W
0	LOCKUPEN	Cortex-M0 Look Up 功能使能。 0: 关闭 1: 使能注: 当 Cortex-M0 读取错误的指令时, MCU 会复位, 以增强系统可靠性。	0	R/W

注意, 只有 SYSCON_UNLOCK 寄存器保护解除后, 才能写该寄存器。

8.2.2 端子 Deep Sleep 中断模式控制寄存器(SYSCON_PORTINTCR)

地址偏移: 0x04

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY															
WO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													PADDLSPCON	PADINTSEL	
													R/W	R/W	

位	标记	功能描述	复位值	读写
31:16	Key	只有高位写 0x5A69 时配置该寄存器才有效, 写其它值时无效。	0x0	WO
15:2	—	保留	0x0	—
1	PADDLSPCON	0: 当进入 Deep Sleep 后, PAD 的中断产生模式自动切换到 Deep Sleep 中断产生模式(没有 Debounce 功能) 1: 当进入 Deep Sleep 后, PAD 的中断产生模式不会自动切换, 由 SYSCON_PORTINTSEL.INTSEL 位决定中断产生模式。	0	R/W

位	标记	功能描述	复位值	读写
0	PADINTSEL	端口中断模式选择 0: ACTIVE/Sleep 中断产生模式 1: Deep Sleep 中断产生模式	0	R/W

注意:

- 1.选用 Deep Sleep 中断产生模式时，GPIO 端子的 Debounce 功能需要关闭，抗干扰能力很差，一般推荐用户在芯片需要使用端子中断唤醒 Deep Sleep 模式才选择此模式；
- 2.注意，只有 SYSCON_UNLOCK 寄存器保护解除后，才能写该寄存器。

8.2.3 端子控制寄存器(SYSCON_PORTCR)

地址偏移: 0x08

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						LPTIM_GATE_SEL[1:0]		TIM11_GATE_SEL[1:0]		TIM10_GATE_SEL[1:0]		SPINCS_SEL[3:0]			
						R/W		R/W		R/W		R/W			

位	标记	功能描述	复位值	读写
31:10	—	保留	0x0	—
9:8	LPTIM_GATE_SEL[1:0]	Low Power Timer 门控输入信号来源选择 00: LPTIM_GATE 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W
7:6	TIM11_GATE_SEL[1:0]	Timer11 门控输入信号来源选择 00: TIM11_GATE 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W
5:4	TIM10_GATE_SEL[1:0]	Timer10 门控输入信号来源选择 00: TIM10_GATE 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W
3:0	SPINCS_SEL[3:0]	SPI SLVAE 模式 NCS 信号来源选择	0x0	R/W

位	标记	功能描述		复位值	读写
		0000	固定高电平		
		0001	PA1		
		0010	PA2		
		0011	PA3		
		0100	PB4		
		0101	PB5		
		0110	PC3		
		0111	PC4		
		1000	PC5		
		1001	PC6		
		1010	PC7		
		1011	PD1		
		1100	PD2		
		1101	PD3		
		1110	PD4		
		1111	PD6		

8.2.4 PCA 捕获通道控制寄存器(SYSCON_PCACR)

地址偏移: 0x0C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						PCA_CAP4_SEL[1:0]		PCA_CAP3_SEL[1:0]		PCA_CAP2_SEL[1:0]		PCA_CAP1_SEL[1:0]		PCA_CAP0_SEL[1:0]	
						R/W		R/W		R/W		R/W		R/W	

位	标记	功能描述	复位值	读写
31:10	—	保留	0x0	—
9:8	PCA_CAP4_SEL[1:0]	PCA 捕获通道 4 信号来源选择 00: PCA_CH4 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W
7:6	PCA_CAP3_SEL[1:0]	PCA 捕获通道 3 信号来源选择 00: PCA_CH3	0x0	R/W

位	标记	功能描述	复位值	读写
		01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD		
5:4	PCA_CAP2_SEL[1:0]	PCA 捕获通道 2 信号来源选择 00: PCA_CH2 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W
3:2	PCA_CAP1_SEL[1:0]	PCA 捕获通道 1 信号来源选择 00: PCA_CH1 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W
1:0	PCA_CAP0_SEL[1:0]	PCA 捕获通道 0 信号来源选择 00: PCA_CH0 01: UART0_RXD 10: UART1_RXD 11: LPUART_RXD	0x0	R/W

8.2.5 TIM1 通道输入源选择(SYSCON_TIM1CR)

地址偏移: 0x10

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									CLKFA ILBRK EN	DSL PBRK EN	TIM1BRK OUTCFG	TIM1ETR_SEL [3:0]			
									R/W	R/W	R/W	R/W			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TIM1CH4IN_ SEL[2:0]			保留	TIM1CH3IN_ SEL[2:0]			保留	TIM1CH2IN_SEL[2:0]				保留	TIM1CH1IN_ SEL[2:0]	
	R/W				R/W				R/W					R/W	

位	标记	功能描述	复位值	读写
31:23	—	保留	0x0	—
22	CLKFAILBRKEN	系统时钟停止检出时 TIM1 Break 使能 0: 无效 1: 使能	0	R/W
21	DSLPPBRKEN	Deep Sleep 模式时 TIM1 Break 使能 0: 无效	0	R/W

位	标记	功能描述	复位值	读写																																
		1: 使能																																		
20	TIM1BRKOUTCFG	0: ocxp/ocxnp 在 break 模式符合 STM 协议 1: ocxp/ocxnp 在 break 模式同时输出 0	0	R/W																																
19:16	TIM1ETR_SEL[3:0]	<div>TIM1ETR 信号来源选择<table><tr><td>0000</td><td>固定高电平</td></tr><tr><td>0001</td><td>PA1</td></tr><tr><td>0010</td><td>PA2</td></tr><tr><td>0011</td><td>PA3</td></tr><tr><td>0100</td><td>PB4</td></tr><tr><td>0101</td><td>PB5</td></tr><tr><td>0110</td><td>PC3</td></tr><tr><td>0111</td><td>PC4</td></tr><tr><td>1000</td><td>PC5</td></tr><tr><td>1001</td><td>PC6</td></tr><tr><td>1010</td><td>PC7</td></tr><tr><td>1011</td><td>PD1</td></tr><tr><td>1100</td><td>PD2</td></tr><tr><td>1101</td><td>PD3</td></tr><tr><td>1110</td><td>PD4</td></tr><tr><td>1111</td><td>PD6</td></tr></table></div>	0000	固定高电平	0001	PA1	0010	PA2	0011	PA3	0100	PB4	0101	PB5	0110	PC3	0111	PC4	1000	PC5	1001	PC6	1010	PC7	1011	PD1	1100	PD2	1101	PD3	1110	PD4	1111	PD6	0x0	R/W
0000	固定高电平																																			
0001	PA1																																			
0010	PA2																																			
0011	PA3																																			
0100	PB4																																			
0101	PB5																																			
0110	PC3																																			
0111	PC4																																			
1000	PC5																																			
1001	PC6																																			
1010	PC7																																			
1011	PD1																																			
1100	PD2																																			
1101	PD3																																			
1110	PD4																																			
1111	PD6																																			
15	—	保留	0	—																																
14:12	TIM1CH4IN_SEL[2:0]	TIM1CH4 输入通道信号来源选择 000: TIM1_CH4 001: UART0_RXD 010: UART1_RXD 011: LPUART_RXD 100: LIRC 其他: 保留	0x0	R/W																																
11	—	保留	0	—																																
10:8	TIM1CH3IN_SEL[2:0]	TIM1CH3 输入通道信号来源选择 000: TIM1_CH3 001: UART0_RXD 010: UART1_RXD 011: LPUART_RXD 100: LIRC 其他: 保留																																		
7	—	保留	0	—																																
6:4	TIM1CH2IN_SEL[2:0]	TIM1CH2 输入通道信号来源选择 000: TIM1_CH2 001: UART0_RXD																																		

位	标记	功能描述	复位值	读写
		010: UART1_RXD 011: LPUART_RXD 100: LIRC 其他: 保留		
3	—	保留	0	—
2:0	TIM1CH1IN_SEL[2:0]	TIM1CH1 输入通道信号来源选择 000: TIM_CH1 001: UART0_RXD 010: UART1_RXD 011: LPUART_RXD 100: LIRC 其他: 保留		

8.2.6 TIM2 通道输入源选择(SYSCON_TIM2CR)

地址偏移: 0x14

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													TIM2ETR _SEL[3:0]		
													R/W		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TIM2CH4IN _SEL[2:0]			保留	TIM2CH3IN _SEL[2:0]			保留	TIM2CH2IN _SEL[2:0]			保留	TIM2CH1IN _SEL[2:0]		
	R/W				R/W				R/W				R/W		

位	标记	功能描述	复位值	读写	
31:20	—	保留	0x0	—	
19:16	TIM2ETR_SEL[3:0]	TIM2 ETR 信号来源选择		0x0	R/W
		0000	固定高电平		
		0001	PA1		
		0010	PA2		
		0011	PA3		
		0100	PB4		
		0101	PB5		
		0110	PC3		
		0111	PC4		
		1000	PC5		

位	标记	功能描述		复位值	读写
		1001	PC6		
		1010	PC7		
		1011	PD1		
		1100	PD2		
		1101	PD3		
		1110	PD4		
		1111	PD6		
15	—	保留		0	—
14:12	TIM2CH4IN_SEL[2:0]	TIM2 CH4 输入通道信号来源选择 000: TIM2_CH4 001: UART0_RXD 010: UART1_RXD 011: LPUART_RXD 100: LIRC 其他: 保留		0x0	R/W
11	—	保留		0	—
10:8	TIM2CH3IN_SEL[2:0]	TIM2 CH3 输入通道信号来源选择 000: TIM2_CH3 001: UART0_RXD 010: UART1_RXD 011: LPUART_RXD 100: LIRC 其他: 保留			
7	—	保留		0	—
6:4	TIM2CH2IN_SEL[2:0]	TIM2 CH2 输入通道信号来源选择 000: TIM2_CH2 001: UART0_RXD 010: UART1_RXD 011: LPUART_RXD 100: LIRC 其他: 保留			
3	—	保留		0	—
2:0	TIM2CH1IN_SEL[2:0]	TIM2 CH1 输入通道信号来源选择 000: TIM_CH1 001: UART0_RXD 010: UART1_RXD 011: LPUART_RXD 100: LIRC 其他: 保留			

8.2.7 Syscon 寄存器写保护(SYSCON_UNLOCK)

地址偏移: 0x50

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:1	KEY	只有高位写 0x2AD5334C 时配置该寄存器才有效, 写其它值时无效。	0x0	—
0	UNLOCK	0: 保护有效 1: 解除保护	0	R/W

写 0x55AA6699, 解除保护。

9. 中断控制器 (NVIC)

9.1 简介

Cortex®-M0+提供中断控制器,用于总体管理异常,称之为“嵌套向量中断控制器(NVIC)”。

NVIC 和处理器内核紧密相连,可以实现低延迟的中断处理和高效地处理晚到的中断。支持最多 32 个中断请求(IRQ)输入,以及 1 个不可屏蔽中断(NMI)输入(在本产品系统中并未使用)。另外,处理器还支持多个内部异常。

本章节只对处理器的 32 个外部中断请求(中断 0 到中断 31)做详细介绍,处理器内部异常的具体情况可参考其他相关文档。详情请参考“ARM®Cortex®-M0+ Technical Reference Manual”与“ARM®v6-M Architecture Reference Manual”。

表 9.1. 中断源

中断号	中断源	简介	Sleep 模式唤醒	Deep Sleep 模式唤醒	向量地址
0	GPIO_PA	GPIOA 中断	Y	Y	0x00000040
1	GPIO_PB	GPIOB 中断	Y	Y	0x00000044
2	GPIO_PC	GPIOC 中断	Y	Y	0x00000048
3	GPIO_PD	GPIOD 中断	Y	Y	0x0000004C
4	Flash	Flash 中断	N	N	0x00000050
5	保留	-	-	-	0x00000054
6	UART0	UART0 中断	Y	N	0x00000058
7	UART1	UART1 中断	Y	N	0x0000005C
8	LPUART	LPUART 中断	Y	Y	0x00000060
9	保留	-	-	-	0x00000064
10	SPI	SPI 中断	Y	N	0x00000068
11	保留	-	-	-	0x0000006C
12	I2C	I2C 中断	Y	N	0x00000070
13	保留	-	-	-	0x0000006C
14	TIM10	TIM10 中断	Y	N	0x00000078
15	TIM11	TIM11 中断	Y	N	0x0000007C
16	LPTIM	LPTIM 中断	Y	Y	0x00000080
17	保留	-	-	-	0x0000007C

中断号	中断源	简介	Sleep 模式唤醒	Deep Sleep 模式唤醒	向量地址
18	TIM1	TIM1 中断	Y	N	0x00000088
19	TIM2	TIM2 中断	Y	N	0x0000008C
20	保留	-	-	-	0x00000088
21	PCA	PCA 中断	Y	N	0x00000094
22	WWDG	WWDG 中断	Y	N	0x00000098
23	IWDG	IWDG 中断	Y	Y	0x0000009C
24	ADC	ADC 中断	Y	N	0x000000A0
25	LVD	LVD 中断	Y	Y	0x000000A4
26	VC	VC 中断	Y	Y	0x000000A8
27	保留	-	-	-	0x000000A4
28	AWK	AWK 中断	Y	Y	0x000000B0
29	ONEWIRE	1-WIRE 中断	Y	N	0x000000B4
30	RTC	RTC 中断	Y	Y	0x000000B8
31	CLKTRIM	CLKTRIM 中断	Y	Y ^{注1}	0x000000BC

注 1：只有在选择内部低速监控外部低速时钟功能时才能唤醒。

9.2 特性

- ◆ 32 个可屏蔽中断通道(不包含 16 个 Cortex®-M0+的中断线)
- ◆ 4 个可编程的优先级(使用了 2 位的中断优先级)
- ◆ 低延时的异常和中断处理
- ◆ 电源管理控制
- ◆ 系统控制寄存器的实现
- ◆ 支持嵌套和向量中断
- ◆ 动态改变优先级

9.3 中断优先级

软件可以对外部中断设置 4 级优先级。最高优先级为“0”，最低优先级为“3”，所有用户可配置的优先级的默认值为“0”。

如果处理器已经在运行另外一个中断处理，而新中断的优先级大于正在执行的，这时就会发生抢占。正在运行的中断处理会被暂停，转而执行新的中断，这个过程通常被称为中断嵌套。新的中断执行完毕后，之前的中断处理会继续执行，并且在其结束后返回到程序线程中。

如果处理器正在运行的另外一个中断处理的优先级相同或者更高，新的中断将会等待并且进入挂起状态。挂起的中断将会一直等到当前中断等级改变，例如，当前运行的中断处理完成返回后，当前优先级降低到了比挂起中断还要小。

如果两个中断同时发生，并且它们的优先级相同，中断编号较小的中断将会首先执行。例如，如果中断 0 和中断 1 使能且具有相同的优先级，在它们同时被触发时，中断 0 会首先执行。

9.4 中断向量表

Cortex®-M0+响应中断时，处理器自动从存储器的中断向量表中取出中断服务例程(ISR)的起始地址。向量表包括复位后堆栈(MSP)的初始值以及所有异常处理的入口地址。向量号表示处理异常的先后次序。其中，中断向量的存储顺序同编号一致由于每个都是 1 个字(4 字节)，中断向量的地址为中断编号乘以 4，每个中断向量都是处理的起始地址。

表 9.2. 中断向量表

中断号	优先级	优先级类型	中断源	简介	地址
	-	-	-	MSP 初始值	0x00000000
	-3	固定	Reset	复位向量(RESET)	0x00000004
	-2	固定	NMI	不可屏蔽中断	0x00000008
	-1	固定	HardFault	所有类型的错误(fault)	0x0000000C
	-	-	-	保留	0x0000000C- 0x0000002B
	3	固定	SVCall	通用 SWI 指令调用的系统服务	0x0000002C
	-	-	-	保留	0x00000030- 0x00000037
	5	固定	PendSV	可挂起的系统服务	0x00000038
	6	固定	SysTick	系统嘀嗒定时器	0x0000003C
0		可配置	GPIO_PA	GPIOA 中断	0x00000040
1		可配置	GPIO_PB	GPIOB 中断	0x00000044
2		可配置	GPIO_PC	GPIOC 中断	0x00000048
3		可配置	GPIO_PD	GPIOD 中断	0x0000004C

中断号	优先级	优先级类型	中断源	简介	地址
4		可配置	Flash	Flash 中断	0x00000050
5		可配置	保留	-	0x00000054
6		可配置	UART0	UART0 中断	0x00000058
7		可配置	UART1	UART1 中断	0x0000005C
8		可配置	LPUART	LPUART 中断	0x00000060
9		可配置	保留	-	0x00000064
10		可配置	SPI	SPI 中断	0x00000068
11		可配置	保留	-	0x0000006C
12		可配置	I2C	I2C 中断	0x00000070
13		可配置	保留	-	0x0000006C
14		可配置	TIM10	TIM10 中断	0x00000078
15		可配置	TIM11	TIM11 中断	0x0000007C
16		可配置	LPTIM	LPTIM 中断	0x00000080
17		可配置	保留	-	0x0000007C
18		可配置	TIM1	TIM1 中断	0x00000088
19		可配置	TIM2	TIM2 中断	0x0000008C
20		可配置	保留	-	0x00000088
21		可配置	PCA	PCA 中断	0x00000094
22		可配置	WWDG	WWDG 中断	0x00000098
23		可配置	IWDG	IWDG 中断	0x0000009C
24		可配置	ADC	ADC 中断	0x000000A0
25		可配置	LVD	LVD 中断	0x000000A4
26		可配置	VC	VC 中断	0x000000A8
27		可配置	保留	-	0x000000A4
28		可配置	AWK	AWK 中断	0x000000B0
29		可配置	ONEWIRE	1-WIRE 中断	0x000000B4
30		可配置	RTC	RTC 中断	0x000000B8
31		可配置	CLKTRIM	CLKTRIM 中断	0x000000BC

9.5 中断唤醒控制 WIC

当处理器利用退出休眠(Sleep-on-exit)特性或者执行 WFI 指令进入休眠之后，就会停止指令执行，并且当发生了中断请求(更高优先级)并且需要处理时，处理器就会被唤醒。

WFI 行为	唤醒	ISR 执行
PRIMASK 清除	Y	Y
IRQ 优先级>当前等级	N	N
IRQ 优先级 \leq 当前等级		
PRIMASK 置位(中断禁止)		
IRQ 优先级>当前等级	Y	N
IRQ 优先级 \leq 当前等级	N	N

9.5.1 NVIC 从深度休眠模式唤醒设置进入中断 ISR 设置

- 1) 使能深度休眠需要唤醒的中断 NVIC
- 2) 使能模块中断使能
- 3) 设置 SCR.SleepDEEP 为 1
- 4) 使用 WFI 指令进入深度休眠模式
- 5) 系统进入深度休眠模式等待中断唤醒，唤醒后执行中断服务子程序

例程：

```
SCR|=0x00000004u;
```

```
__asm("nop");
```

```
__asm("nop");
```

```
__asm("nop");
```

```
while(1){
```

```
__asm("WFI");
```

```
}
```

9.5.2 NVIC 从深度休眠模式唤醒设置不执行中断 ISR 设置

- 1) 使能深度休眠需要唤醒的中断 NVIC
- 2) 使用 PRIMASK 寄存器屏蔽中断
- 3) 使能模块中断使能

- 4) 设置 SCR.SleepDEEP 为 1
- 5) 使用 WFI 指令进入深度休眠模式
- 6) 系统进入深度休眠模式等待中断唤醒，唤醒后执行下一条指令
- 7) 清除中断标志，清除中断挂起状态

9.5.3 使用退出休眠特性

退出休眠(Sleep-on-exit)非常适合中断驱动的应用程序。当该特性使能时，只要完成异常处理并且返回到了线程模式，处理器就会进入休眠模式。利用退出休眠特性，处理器可以尽可能的处于休眠模式。

Cortex®-M0+利用退出休眠特性进入休眠，这种情况同执行完异常退出后立即执行 WFI 的效果差不多。不过，为了下次进入异常时，不用再进行压栈操作，处理器不会执行出栈的过程。

- 1) 使能深度休眠需要唤醒的中断 NVIC
- 2) 使能模块中断使能
- 3) 设置 SCR.SleepDEEP 为 1
- 4) 设置 SCR.SleepONEXIT 为 1
- 5) 使用 WFI 指令进入深度休眠模式
- 6) 系统进入深度休眠模式等待中断唤醒，唤醒后执行中断服务子程序
- 7) 退出中断服务时自动进入休眠模式

例程：

```
SCR|=0x00000004u;
```

```
SCR|=0x00000002u;
```

```
__asm("nop");
```

```
__asm("nop");
```

```
__asm("nop");
```

```
while(1){
```

```
__asm("WFI");
```

```
}
```

9.6 软件基本操作

9.6.1 外部中断使能

在每一个外设模块内部都有各自的中断使能寄存器，在需要进行中断操作时，必须首先打开外设自己的中断使能。该使能位的操作没有在本章节讨论，请参考外设模块各自的章节描述。

9.6.2 NVIC 中断使能和清除使能

Cortex-M0+处理器支持最多 32 个中断源，每个中断源都对应有一个中断使能位和清零位。这样就有了 32 位的中断使能置位寄存器 `NVIC_ISER` 和 32 位的中断使能清零寄存器 `NVIC_ICER`。如果想使能某一个中断，则对 `NVIC_ISER` 寄存器的相应位置 1。如果想清零某一个中断使能，则对 `NVIC_ICER` 寄存器的相应位置 1。

注意，这里提到的中断使能仅仅是针对处理器 NVIC 而言的，每个外设的中断生成与否，是由外设的中断控制寄存器决定，与 `NVIC_ISER` 和 `NVIC_ICER` 无关。

9.6.3 NVIC 中断挂起和清除挂起

如果一个中断发生了，却无法立即处理，这个中断请求将会被挂起。挂起状态保存在一个寄存器中，如果处理器的当前优先级还没有降低到可以处理挂起的请求，并且没有手动清除挂起状态，该状态将会一直保持有效。

当处理器开始进入中断处理，硬件会自动清除相应的中断挂起状态。

可以通过操作中断设置挂起 `NVIC_ISPR` 和中断清除挂起 `NVIC_ICPR` 这两个寄存器来访问或修改中断挂起状态。中断挂起状态寄存器允许使用软件来触发中断。

9.6.4 NVIC 中断优先级

设置 `NVIC_IPR0`-`NVIC_IPR7` 寄存器决定 `IRQ0`-`IRQ32` 的优先级。中断优先级寄存器的编程应该在中断使能之前，其通常是在程序开始时完成的。应该避免在中断使能之后改变中断优

先级，这种情况的结果不可预知，并且不被 Cortex-M0+处理器支持。

9.6.5 NVIC 中断屏蔽

有些对时间敏感的应用，需要在一段较短的时间内禁止所有中断，可以利用中断屏蔽寄存器 PRIMASK 实现。PRIMASK 只有 1 位有效，并且在复位后默认为 0。该寄存器为 0 时，所有的中断和异常都处于允许状态；而设为 1 后，只有 NMI(本系统不支持)和硬件错误异常处于使能。实际上，当 PRIMASK 设置为 1 后，处理器的当前优先级就降到了 0。可以通过多种方法编程 PRIMASK 寄存器，使用汇编语言，可以利用 CPSIEi 和 CPSIDi 指令来设置和清除 PRIMASK 寄存器。若使用 C 语言以及 CMSIS 设备驱动库，用户可以使用以下函数来设置和清除 PRIMASK。

```
Void__enable_irq(void);//清除 PRIMASK
```

```
Void__disable_irq(void);//设置 PRIMASK
```

9.7 寄存器列表

SCS 基址：0xE000E000

表 9.3. NVIC 寄存器列表和复位值

偏移地址	名称	描述	复位值
0x100	NVIC_ISER	IRQ0~IRQ31 使能设置控制寄存器	0x00000000
0x180	NVIC_ICER	IRQ0~IRQ31 使能清除控制寄存器	0x00000000
0x200	NVIC_ISPR	IRQ0~IRQ31 挂起设置控制寄存器	0x00000000
0x280	NVIC_ICPR	IRQ0~IRQ31 挂起清除控制寄存器	0x00000000
0x400	NVIC_IPR0	IRQ0~IRQ3 优先级控制寄存器 0	0x00000000
0x404	NVIC_IPR1	IRQ4~IRQ7 优先级控制寄存器 1	0x00000000
0x408	NVIC_IPR2	IRQ8~IRQ11 优先级控制寄存器 2	0x00000000
0x40C	NVIC_IPR3	IRQ12~IRQ15 优先级控制寄存器 3	0x00000000
0x410	NVIC_IPR4	IRQ16~IRQ19 优先级控制寄存器 4	0x00000000
0x414	NVIC_IPR5	IRQ20~IRQ23 优先级控制寄存器 5	0x00000000
0x418	NVIC_IPR6	IRQ24~IRQ27 优先级控制寄存器 6	0x00000000
0x41C	NVIC_IPR7	IRQ28~IRQ31 优先级控制寄存器 7	0x00000000

9.8 寄存器说明

9.8.1 中断使能设置寄存器(NVIC_ISER)

地址偏移: 0x100

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:0	SETENA	设置使能中断 0 到中断 31; 写“1”置位, 写“0”无效 [0]: IRQ0 [1]: IRQ1 [2]: IRQ2 [31]: IRQ31	0x0	R/W

9.8.2 中断使能清除寄存器(NVIC_ICER)

地址偏移: 0x180

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:0	CLRENA	清除使能中断 0 到中断 31; 写“1”清除, 写“0”无效 [0]: IRQ0 [1]: IRQ1 [2]: IRQ2 [31]: IRQ31	0x0	R/W

9.8.3 中断挂起设置寄存器(NVIC_ISPR)

地址偏移: 0x200

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:0	SETPEND	设置中断 0 到中断 31 挂起状态; 写“1”置位, 写“0”无效 [0]: IRQ0 [1]: IRQ1 [2]: IRQ2 [31]: IRQ31	0x0	R/W

9.8.4 中断挂起清除寄存器(NVIC_ICPR)

地址偏移: 0x280

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:0	CLRPEND	清除外部中断 0 到中断 31 的挂起状态; 写“1”清零, 写“0”无效 [0]: IRQ0 [1]: IRQ1 [2]: IRQ2 [31]: IRQ31	0x0	R/W

9.8.5 中断优先级控制寄存器 0(NVIC_IPR0)

地址偏移: 0x400

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR0[31:16]															
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR0[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR0	中断 0 到中断 3 的优先级 [31:30]: 中断 3 的优先级 [23:22]: 中断 2 的优先级 [15:14]: 中断 1 的优先级 [7:6]: 中断 0 的优先级 其中, 00 优先级高, 11 优先级低	0x0	R/W

9.8.6 中断优先级控制寄存器 1(NVIC_IPR1)

地址偏移: 0x404

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR1[31:16]															
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR1[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR1	中断 4 到中断 7 的优先级 [31:30]: 中断 7 的优先级 [23:22]: 中断 6 的优先级 [15:14]: 中断 5 的优先级 [7:6]: 中断 4 的优先级 其中, 00 优先级高, 11 优先级低	0x0	R/W

9.8.7 中断优先级控制寄存器 2(NVIC_IPR2)

地址偏移: 0x408

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR2[31:16]															
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR2[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR2	中断 8 到中断 11 的优先级; [31:30]: 中断 11 的优先级 [23:22]: 中断 10 的优先级 [15:14]: 中断 9 的优先级 [7:6]: 中断 8 的优先级 其中, 00 优先级最高, 11 优先级最低	0x0	R/W

9.8.8 中断优先级控制寄存器 3(NVIC_IPR3)

地址偏移: 0x40C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR3[31:16]															
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR3[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR3	中断 12 到中断 15 的优先级; [31:30]: 中断 15 的优先级 [23:22]: 中断 14 的优先级 [15:14]: 中断 13 的优先级 [7:6]: 中断 12 的优先级 其中, 00 优先级最高, 11 优先级最低	0x0	R/W

9.8.9 中断优先级控制寄存器 4(NVIC_IPR4)

地址偏移: 0x410

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR4[31:16]															
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR3[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR4	中断 16 到中断 19 的优先级; [31:30]: 中断 19 的优先级 [23:22]: 中断 18 的优先级 [15:14]: 中断 17 的优先级 [7:6]: 中断 16 的优先级 其中, 00 优先级最高, 11 优先级最低	0x0	R/W

9.8.10 中断优先级控制寄存器 5(NVIC_IPR5)

地址偏移: 0x414

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR5[31:16]															
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR5[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR5	中断 20 到中断 23 的优先级; [31:30]: 中断 23 的优先级 [23:22]: 中断 22 的优先级 [15:14]: 中断 21 的优先级 [7:6]: 中断 20 的优先级 其中, 00 优先级最高, 11 优先级最低	0x0	R/W

9.8.11 中断优先级控制寄存器 6(NVIC_IPR6)

地址偏移: 0x418

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR6[31:16]															
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR6[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR6	中断 24 到中断 27 的优先级; [31:30]: 中断 27 的优先级 [23:22]: 中断 26 的优先级 [15:14]: 中断 25 的优先级 [7:6]: 中断 24 的优先级 其中, 00 优先级最高, 11 优先级最低	0x0	R/W

9.8.12 中断优先级控制寄存器 7(NVIC_IPR7)

地址偏移: 0x41C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR7[31:16]															
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR7[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:0	IPR7	中断 28 到中断 31 的优先级; [31:30]: 中断 31 的优先级 [23:22]: 中断 30 的优先级 [15:14]: 中断 29 的优先级 [7:6]: 中断 28 的优先级 其中, 00 优先级最高, 11 优先级最低	0x0	R/W

10. 通用输入输出口(GPIO)

10.1 简介

通用输入/输出口用于芯片和外部进行数据传输，共有 4 组 GPIO，GPIOA，GPIOB，GPIOC 和 GPIOD，可以通过配置将 GPIO 映射到对应芯片引脚。4 组 GPIO 的功能基本相同，每个引脚可以被独立配置为数字输入或者数字输出口。另外还可配置为如模拟输入，外部中断，片上外设的输入/输出等复用功能。但是在同一时刻仅有一个复用功能可以映射到引脚上。复用功能的映射是通过端口复用功能寄存器(GPIOx_AFR)控制的。

每个通用 I/O 口都有 5 个配置寄存器(GPIOx_DIRCR，GPIOx_OTYPER，GPIOx_PUPDR，GPIOx_SLEWCR 和 GPIOx_DRVCR)，2 个数据寄存器(GPIOx_IDR 和 GPIOx_ODR)，1 个输出置位寄存器(GPIOx_ODSET)，1 个输出复位寄存器(GPIOx_ODCLR)和 1 个复用功能寄存器(GPIOx_AFR)。

每个端口都可以配置成内部拉高(pull up)/拉低(pull down)的输入/输出，高阻输入(floating input)，推挽输出(push-pull output)，开漏输出(open drain output)，增强驱动能力输出。芯片复位后端口为高阻输入，目的是防止芯片被异常复位时，外部器件产生异常动作。但为了避免高阻输入而产生的漏电，用户要在芯片启动之后对端口进行相应配置(配置成内部拉高输入或者输出)。端口被配成模拟端口后，数字功能被隔离，不能输出数字“1”和“0”，CPU 读取端口的结果为“0”。

所有端口都可以提供外部中断，并且每个中断都可以配置成高电平触发、低电平触发、上升沿触发、下降沿触发或者任意边沿触发，支持边沿模式下的输入消抖。支持工作模式/睡眠模式/深度睡眠模式下产生中断。

10.2 特性

◆ 输出状态：带有上拉或下拉的推挽输出或开漏输出

- ◆ 从数据寄存器(GPIOx_ODR)或外设(复用功能输出)输出数据
- ◆ 可配每个 I/O 口的速度
- ◆ 输入状态：浮空、上拉/下拉、模拟输入
- ◆ 从数据寄存器(GPIOx_IDR)或外设输入数据(复用功能输出)
- ◆ 输出置位/复位寄存器(GPIOx_ODSET, GPIOx_ODCLR)为对 GPIOx_ODR 寄存器提供位改写能力
- ◆ 模拟功能引脚/调试引脚/数字通用引脚/数字功能引脚复用
- ◆ 允许 GPIO 口和外设引脚的高灵活性复用

10.3 功能描述

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：

- 浮空输入
- 上拉输入
- 下拉输入
- 模拟输入
- 具有上拉或下拉能力的开漏输出
- 具有上拉或下拉能力的推挽输出
- 复用功能且具有上拉或下拉能力的推挽输出
- 复用功能且具有上拉或下拉能力的开漏输出

每个 I/O 端口位可以自由编程，I/O 端口寄存器可按 32 位字访问。GPIOx_ODSET，GPIOx_ODCLR 寄存器允许对 GPIOx_ODR 进行位读/改写操作。这种情况下，可以避免在读和更改访问之间产生中断而发生的异常。

下图给出了一个标准 IO 端口的基本结构。

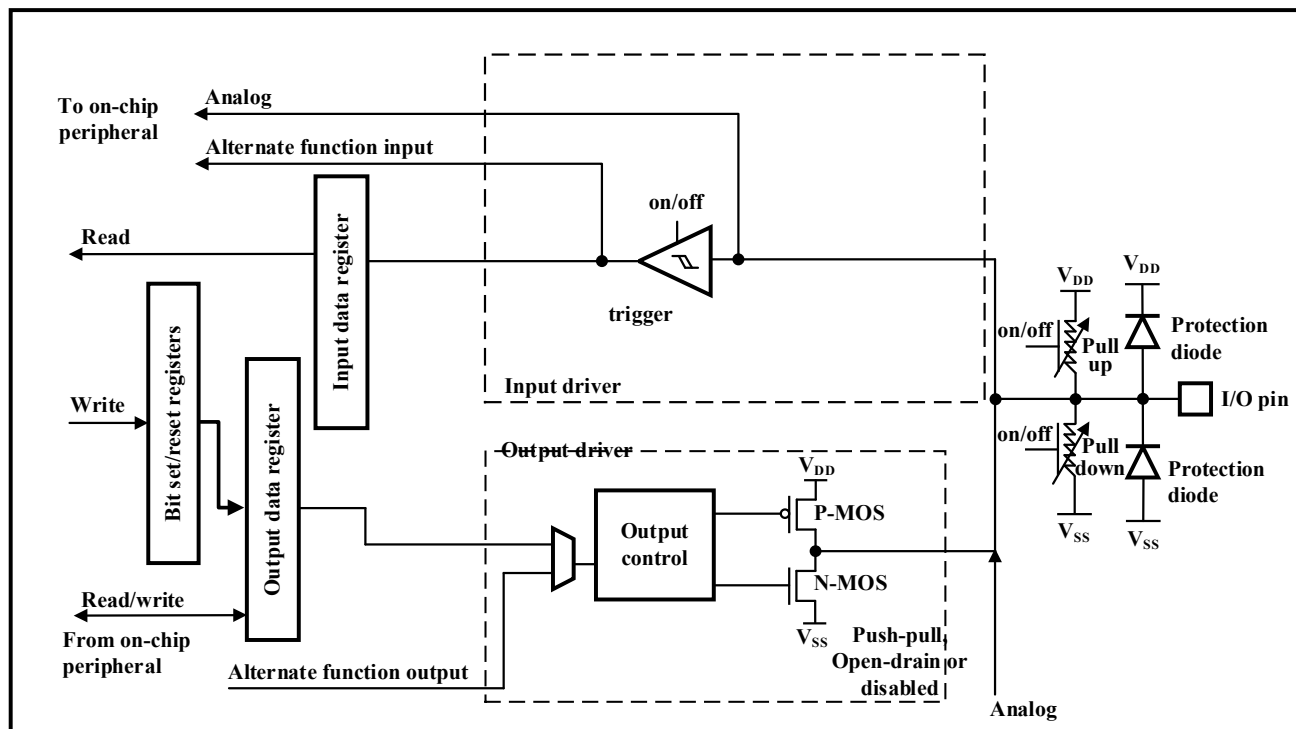


图 10-1. 标准 I/O 端口的位基本结构

表 10.1 给出了端口位的可能配置，表 10.2 给出了 GPIO 口和外设引脚的复用功能映射。

表 10.1. 端口位配置表

AFR[i+3:i]	DIR[i]	OTYPCR[i]	DRVCR[i]	PUPD		IOConfiguration	
0000	1	0	DR[i]	0	0	GPIO output	PP
		0		0	1	GPIO output	PP+PU
		0		1	0	GPIO output	PP+PD
		0		1	1	Reserved	
		1		0	0	GPIO output	OD
		1		0	1	GPIO output	OD+PU
		1		1	0	GPIO output	OD+PD
		1		1	1	Reserved (output OD)	
	0	x	x	0	0	Input	Floating
		x	x	0	1	Input	PU
		x	x	1	0	Input	PD
		x	x	1	1	Reserved (input floating)	
0001 ~ 1110	x	0	DR[i]	0	0	AF	PP
	x	0		0	1	AF	PP+PU
	x	0		1	0	AF	PP+PD
	x	0		1	1	Reserved	

AFR[i+3:i]	DIR[i]	OTYPCR[i]	DRVCR[i]	PUPD		IOConfiguration	
	x	1		0	0	AF	OD
	x	1		0	1	AF	OD+PU
	x	1		1	0	AF	OD+PD
	x	1		1	1	Reserved	
1111	x	x	x	0	0	Input/output	Analog
	x	x	x	0	1	Forbidden	
	x	x	x	1	0		
	x	x	x	1	1		

1.GP=通用，PP=推挽输出，PU=上拉，PD=下拉，OD=开漏，AF=复用功能.

10.3.1 通用 I/O(GPIO)

复位期间和复位后，复用功能未开启，且除了调试引脚以外的所有 I/O 端口都被配置为浮空输入模式。

复位后，调试引脚被置为复用功能的上拉/下拉模式：

PD1：SWDCLK 置于下拉模式

PC7：SWDIO 置于上拉模式

当作为输出配置时，写到输出数据寄存器(GPIOx_ODR)的值输出到相应的引脚上。可以以推挽模式或开漏模式(仅低电平被驱动，高电平表现为高阻)输出。

输入数据寄存器(GPIOx_IDR)在每个 AHB 时钟周期捕捉 I/O 引脚上的数据。所有 GPIO 引脚都有一个内部弱上拉和弱下拉电阻，它们被激活或断开依赖于 GPIOx_PUPD 寄存器的值。

10.3.2 I/O 端口控制寄存器

每个通用 I/O 口都有 5 个配置寄存器(GPIOx_DIRCR, GPIOx_OTYPER, GPIOx_PUPDR, GPIOx_SLEWCR 和 GPIOx_DRVCR)用来配置 I/O 口线。GPIOx_DIRCR 寄存器用来选择方向(输入/输出)。GPIOx_OTYPER、GPIOx_SLEWCR 和 GPIOx_DRVCR 寄存器来选择输出类型(推挽或开漏)、转换速率和驱动能力。GPIOx_PUPDR 寄存器用来选择上拉/下拉方式。

10.3.3 I/O 端口数据寄存器

每个 GPIO 口有两个数据寄存器：输入和输出数据寄存器(GPIOx_IDR 和 GPIOx_ODR)。GPIOx_ODR 用于存储输出数据，其可进行读/写访问。从 I/O 线的输入数据存放在(GPIOx_IDR)寄存器中，该寄存器为只读寄存器。

10.3.4 I/O 数据位处理

输出置位寄存器(GPIOx_ODSET)和输出清零寄存器(GPIOx_ODCLR)寄存器，允许应用对输出数据寄存器(GPIOx_ODR)的每个位进行置位和清零操作。

当对位 GPIOx_ODSET[i]写 1 时则置位相应的 ODR[i]位。当对 GPIOx_ODCLR[i]写 1 时，则清零相应的 ODR[i]位。

对 GPIOx_ODSET, GPIOx_ODCLR 中的任意位写 0 都不会影响 GPIOx_ODR 寄存器的值。不仅可以用 GPIOx_ODSET, GPIOx_ODCLR 寄存器来改变 GPIOx_ODR 的相应位，也可以对 GPIOx_ODR 寄存器直接进行访问。GPIOx_ODSET, GPIOx_ODCLR 寄存器提供对 GPIOx_ODR 寄存器原子位操作处理。

GPIOx_ODR 用 GPIOx_ODSET, GPIOx_ODCLR 置位或清零的访问机制不需要软件去关闭中断来访问 GPIOx_ODR：在一个 AHB 写访问周期改变 1 位或多位数据是可能的。

10.3.5 输入配置

当 I/O 编程配置为输入时：

- 该端口的输出缓冲区禁用
- 由 GPIOx_PUPD 寄存器的值来激活上拉和下拉电阻
- 在每个 AHB 时钟周期 I/O 引脚上的数据被采样进入输入数据寄存器
- 用对输入数据寄存器的读访问来获取同步后的输入数据

下图给出了 I/O 端口的输入配置。

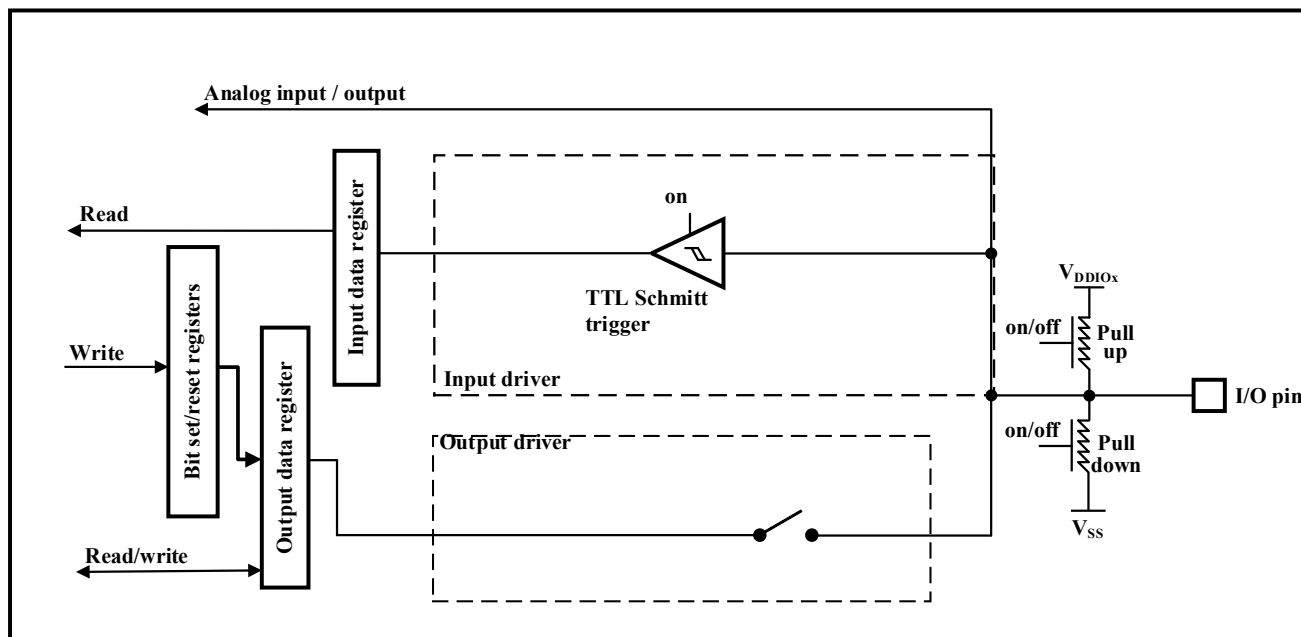


图 10-2. 浮空输入/上拉/下拉配置

10.3.6 输出配置

当 I/O 口配置为输出时：

- 输出缓冲开启：
 - 开漏模式：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：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。
- 施密特触发输入被激活
- 弱上拉和弱下拉电阻是否激活取决于 GPIOx_PUPDR 寄存器的值
- 在每个 AHB 时钟周期 I/O 引脚上的数据被采样进入输入数据寄存器
- 用对输入数据寄存器的读访问来获取同步后的输入数据
- 用对输出寄存器的读访问来获取最后写进该寄存器的值

下图给出了 I/O 端口位的输出配置。

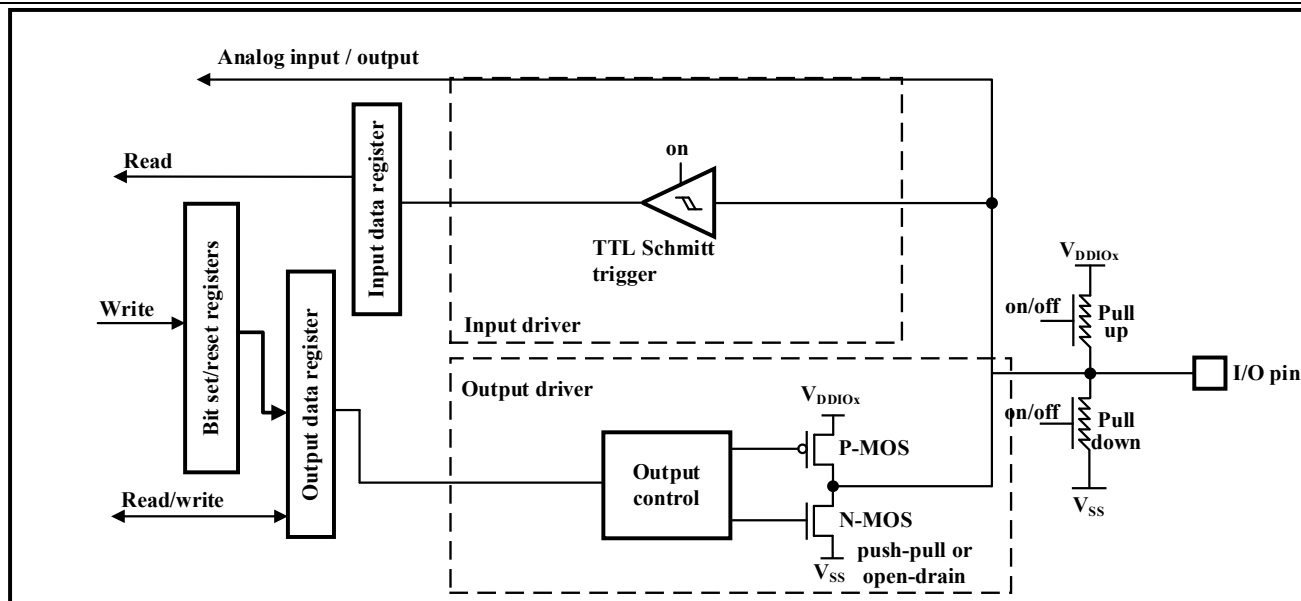


图 10-3. 输出配置

10.3.7 外部中断/唤醒线

所有端口都有中断产生和唤醒能力。进入 Sleep 或者 Deep Sleep 模式后，如果检测到中断产生，WIC 模块会自动唤醒芯片。

每一个数字通用端口都可以由外部信号源产生中断，外部信号源可以是高电平/低电平/上升沿/下降沿 4 种类型的信号，当中断触发时，通过查询中断状态寄存器就可以判是哪一个端口触发了中断，通过置位中断清除寄存器就可以清除对应的中断状态标志位。

10.3.8 I/O 引脚的复用功能和重映射

器件 I/O 通过多路复用器连接到内嵌的外设模块。微观上，同一时刻仅允许外设的复用功能的一个引脚连接到一个 I/O 接口上。因此，同一根线上不能有冲突的外设引脚分配。每个 I/O 引脚都有一个复用功能多路选择器，可通过配置 GPIOx_AFR 寄存器(从引脚 0 到引脚 7)来实现。

复位后，所有的 I/O 口都连接到 GPIO 功能。每个外设还有复用功能映射到不同的 I/O 引脚上，这种方法用于在小封装器件上优化更多的可用外设。

有关每个引脚的具体复用功能请参考：表 10.2.GPIO 口和外设引脚的复用功能映射为了使用一个给定的 I/O 口配置，你必须按如下的原则执行：

- 调试功能：每个器件复位后，这些引脚立即配置为复用功能用来支持调用。
- GPIO：在 GPIOx_DIRCR 寄存器中配置所需的 I/O 为输出，输入。
- 外设的复用功能：
 - 连接 I/O 到所需的 AFRx，AFRx 定义在 GPIOx_AFR 寄存器中；
 - 通过对 GPIOx_PUPDR 和 GPIOx_SLEWCR，GPIOx_DRVCR 寄存器来配置相应引脚的上拉/下拉和输出速度；
 - 通过对 GPIOx_OTYPER 寄存器来配置输出类型：0 表示 push-pull，1 表示 opendrain。
- 附加功能：
 - 对于 ADC/VC，配置 GPIOx_AFR 为 0xF，配置所需的 I/O 口线为模拟方式并在 ADC 或 VC 寄存器中配置所需的功能；
 - 对于附加功能振荡器，在关联的 RCC 寄存器配置相应所需的功能。

表 10.2. GPIO 口和外设引脚的复用功能映射

封装		GPIOx_AFR[i+3:i]									
TSSOP20	Config	0	1	2	3	4	5	6	7	8	F
1		PD4	TIM1_CH1	PCA_CH0	RTC_1HZ	TIM10_TOG	UART0_TXD	TIM10_EXT	BEEP	TIM2_CH1	VCIN2
2		PD5	TIM1_CH1N	PCA_CH4	SPI_MISO	I2C_SCL	UART1_TXD	TIM10_GATE	UART0_TXD	TIM2_CH4	AIN5
3		PD6	TIM1_CH2	PCA_CH3	SPI_MOSI	I2C_SDA	UART1_RXD	LPTIM_EXT	UART0_RXD	TIM2_CH2	AIN6
4	NRST										
5	OSC_IN	PA1	TIM1_CH2N		SPI_CLK	I2C_SDA	UART0_RXD	TIM10_TOG	UART1_RXD		
6	OSC_OUT	PA2	TIM1_CH3		SPI_NSS	I2C_SCL	UART0_TXD	TIM10_TOGN	UART1_TXD	TIM2_CH2	
7	VSS										
8	VCAP										AIN7
9	VDD										
10		PA3	TIM1_CH3N	PCA_CH2	SPI_NSS	RTC_1HZ	LPUART_RXD	PCA_ECI	VC0_OUT	TIM2_CH3	
11	X32K_IN	PB5	TIM1_BKIN	PCA_CH4	SPI_CLK	I2C_SDA	UART0_RXD	TIM11_TOG	LVD_OUT	TIM2_CH1	
12	X32K_OUT	PB4	LPTIM_GATE	PCA_ECI	SPI_NSS	I2C_SCL	UART0_TXD	TIM11_TOGN			
13		PC3	TIM1_CH3	TIM1_CH1N		I2C_SDA	UART1_TXD	PCA_CH1	1-WIRE	TIM2_CH3	AIN1
14		PC4	TIM1_CH4	TIM1_CH2N		I2C_SCL	UART1_RXD	PCA_CH0	CLK_MCO	TIM2_CH4	AIN2
15		PC5	TIM1_BKIN	PCA_CH0	SPI_CLK		LPUART_TXD	TIM11_GATE	LVD_OUT	TIM2_CH1	VCIN1
16		PC6	TIM1_CH1	PCA_CH3	SPI_MOSI		LPUART_RXD	TIM11_EXT	CLK_MCO	TIM2_CH4	AIN0
17	SWDIO	PC7	TIM1_CH2	PCA_CH4	SPI_MISO		UART1_RXD	LIRC_OUT	LXT_OUT		
18	SWDCLK	PD1		PCA_ECI			UART1_TXD	HIRC_OUT	VC0_OUT		
19		PD2	TIM1_CH2	PCA_CH2	SPI_MISO	RTC_1HZ	LPUART_TXD	LPTIM_TOG	1-WIRE	TIM2_CH3	AIN3/VCIN0
20		PD3	TIM1_CH3N	PCA_CH1	SPI_MOSI	HXT_OUT	UART0_RXD	LPTIM_TOGN		TIM2_CH2	AIN4

当 I/O 端口被配置为复用功能时：

- 在开漏或推挽模式下输出缓冲器可被配置
- 外设的信号驱动输出缓冲器
- 施密特触发输入被激活
- 弱上拉和弱下拉电阻是否激活取决于 GPIOx_PUPDR 寄存器的值
- 在每个 AHB 时钟周期 I/O 引脚上的数据被采样进入输入数据寄存器
- 用对输入数据寄存器的读访问来获取同步后的输入数据

下图给出了 I/O 端口位的复用功能配置。

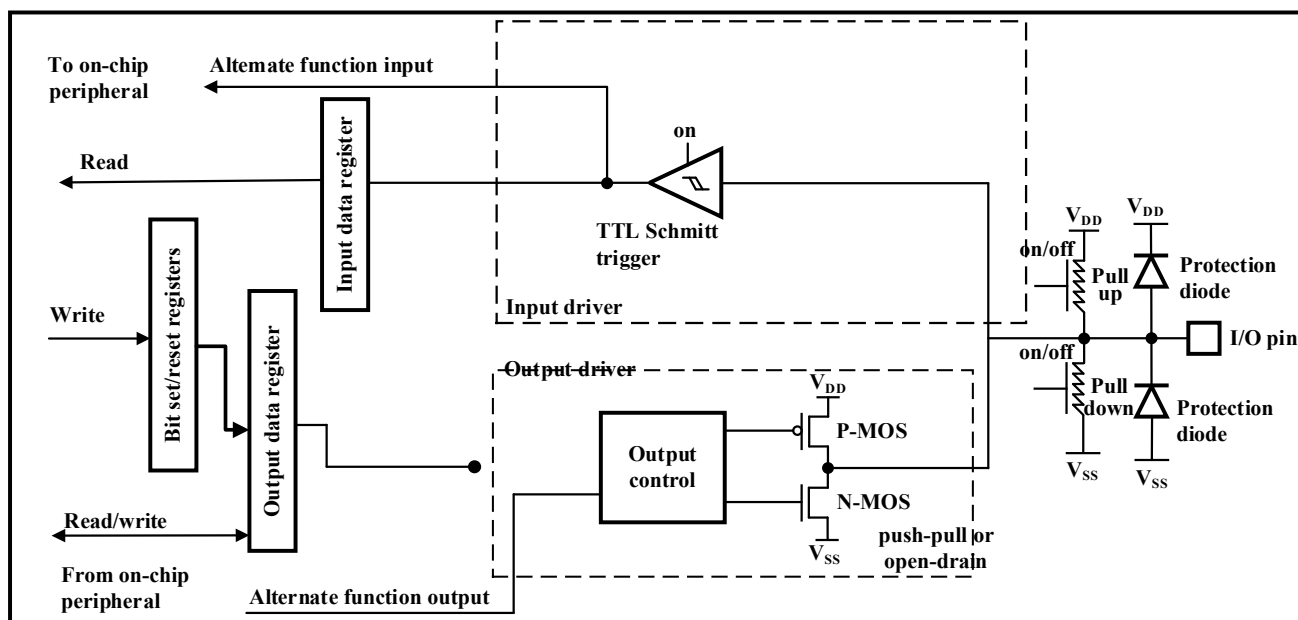


图 10-4. 复用功能配置

10.3.9 模拟配置

当 I/O 端口编程为模拟配置时(GPIOx_AFR.AFR=0xF)：

- 输出缓冲器关闭
- 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制为'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

下图给出了 I/O 端口位的高阻抗模拟输入配置。

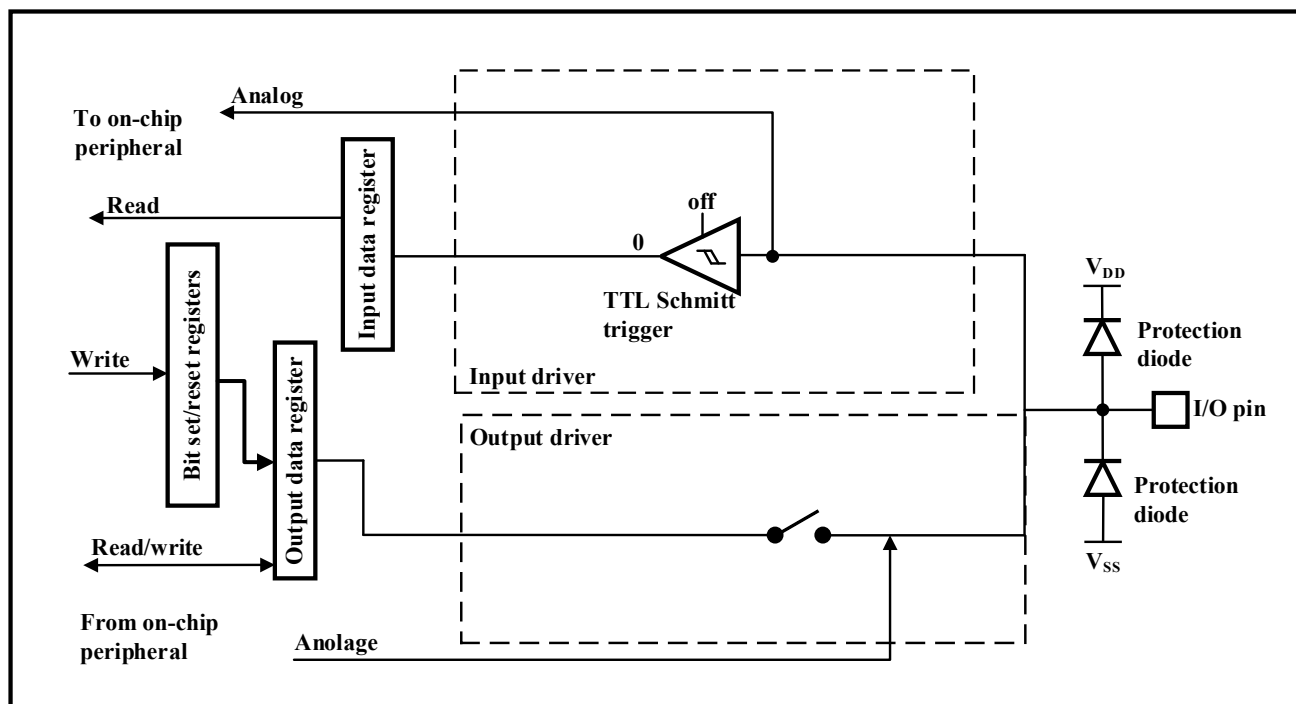


图 10-5. 高阻抗模拟配置

10.3.10 HXT 或 LXT 引脚用作 GPIO

当 HXT 或 LXT 振荡器关断时(复位后的缺省状态), 相关振荡器引脚可以用做普通的 GPIO 口。

当 HXT 或 LXT 振荡器开启(设置 `RCC_SYCLKCR.HXTEN` 或 `RCC_LXTCR.LXTEN` 位来开启)时, 必须配置相应的管脚位模拟功能, 振荡器控制其相关引脚且相关引脚的 GPIO 配置无效。

配置振荡器引脚为模拟功能的方法:

- 通过 `RCC_SYCLKCR.HXTPORT=1` 或 `RCC_LXTCR.LXTPORT=1` 来配置
- 通过设定相应的引脚的 `GPIOx_AFR=0xF` 来配置

当振荡器配置为用户外部时钟输入方式 (`RCC_SYCLKCR.HXTBYP=1` 或 `RCC_LXTCR.LXTBYP=1`), 不需要配置相应的引脚为模拟功能, 仅使用 `OSC_IN` 或 `X32K_IN` 引脚做为时钟输入处理, `OSC_OUT` 或 `X32K_OUT` 引脚仍然可配置为正常的 GPIO 引脚。

10.4 GPIO 寄存器列表

x=A、B、C、D

GPIOx 基地址 0X40021000

GPIOx	偏移地址	描述
GPIOA	0x000	GPIOA 偏移地址
GPIOB	0x400	GPIOB 偏移地址
GPIOC	0x800	GPIOC 偏移地址
GPIOD	0xC00	GPIOD 偏移地址

表 10.3. GPIOx 寄存器列表和复位值

偏移地址	名称	描述	复位值
0x00	GPIOx_DIRCR	输入输出模式寄存器	0x00000000
0x04	GPIOx_OTYPER	输出类型寄存器	0x00000000
0x08	GPIOx_ODR	输出数据寄存器	0x00000000
0x0c	GPIOx_IDR	输入数据寄存器	0x000000xx
0x10	GPIOx_INTEN	中断使能寄存器	0x00000000
0x14	GPIOx_RAWINTSR	中断原始状态寄存器，只读。 不论中断是否使能，都可以读到中断状态。	0x00000000
0x18	GPIOx_MSKINTCR	中断状态寄存器，只读	0x00000000
0x1c	GPIOx_INTCLR	中断清除寄存器	0x00000000
0x20	GPIOx_INTTYPCR	中断类型寄存器	0x00000000
0x24	GPIOx_INTPOLCR	中断类型值寄存器	0x00000000
0x28	GPIOx_INTANY	任意边沿触发中断寄存器	0x00000000
0x2c	GPIOx_ODSET	输出置位寄存器	0x00000000
0x30	GPIOx_ODCLR	输出清除寄存器	0x00000000
0x34	GPIOx_INDBEN	输入去抖动和同步使能寄存器	0x00000000
0x38	GPIOx_DBCLKCR	输入去抖动时钟配置寄存器	0x00000000
0x3c	GPIOx_PUPDR	上拉/下拉寄存器	0x00000000
0x40	GPIOx_SLEWCR	电压转换速率控制	GPIOA: 0x0000000E GPIOB: 0x00000030 GPIOC: 0x000000F8 GPIOD: 0x0000007E
0x44	GPIOx_DRVCR	驱动强度配置	0x00000000
0x48	GPIOx_AFR	复用功能寄存器	0x00000000

0: 表示逻辑值 0; 1: 表示逻辑值 1; X: 表示不确定。

10.5 GPIO 寄存器说明

10.5.1 GPIO 端口方向寄存器(GPIOx_DIRCR)(x=A..D)

偏移地址：0x00

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PxD IR7	PxD IR6	PxD IR5	PxD IR4	PxD IR3	PxD IR2	PxD IR1	PxD IR0
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	PxDIR7	端子 Px7 输入/输出方向选择位 0: 输入模式 1: 输出模式	0	R/W
6	PxDIR6	端子 Px6 输入/输出方向选择位 0: 输入模式 1: 输出模式	0	R/W
5	PxDIR5	端子 Px5 输入/输出方向选择位 0: 输入模式 1: 输出模式	0	R/W
4	PxDIR4	端子 Px4 输入/输出方向选择位 0: 输入模式 1: 输出模式	0	R/W
3	PxDIR3	端子 Px3 输入/输出方向选择位 0: 输入模式 1: 输出模式	0	R/W
2	PxDIR2	端子 Px2 输入/输出方向选择位 0: 输入模式 1: 输出模式	0	R/W
1	PxDIR1	端子 Px1 输入/输出方向选择位 0: 输入模式 1: 输出模式	0	R/W
0	PxDIR0	端子 Px0 输入/输出方向选择位 0: 输入模式 1: 输出模式	0	R/W

10.5.2 GPIO 端口输出类型寄存器(GPIOx_OTYPER)(x=A..D)

偏移地址: 0x04

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PxO TYP7	PxO TYP6	PxO TYP5	PxO TYP4	PxO TYP3	PxO TYP2	PxO TYP1	PxO TYP0
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	PxOTYP7	端子 Px7 输出类型控制位 0: 推挽输出(复位状态) 1: 开漏输出	0	R/W
6	PxOTYP6	端子 Px6 输出类型控制位 0: 推挽输出(复位状态) 1: 开漏输出	0	R/W
5	PxOTYP5	端子 Px5 输出类型控制位 0: 推挽输出(复位状态) 1: 开漏输出	0	R/W
4	PxOTYP4	端子 Px4 输出类型控制位 0: 推挽输出(复位状态) 1: 开漏输出	0	R/W
3	PxOTYP3	端子 Px3 输出类型控制位 0: 推挽输出(复位状态) 1: 开漏输出	0	R/W
2	PxOTYP2	端子 Px2 输出类型控制位 0: 推挽输出(复位状态) 1: 开漏输出	0	R/W
1	PxOTYP1	端子 Px1 输出类型控制位 0: 推挽输出(复位状态) 1: 开漏输出	0	R/W
0	PxOTYP0	端子 Px0 输出类型控制位 0: 推挽输出(复位状态) 1: 开漏输出	0	R/W

10.5.3 GPIO 端口输出数据寄存器(GPIOx_ODR)(x=A..D)

偏移地址: 0x08

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								Px OD7	Px OD6	Px OD5	Px OD4	Px OD3	Px OD2	Px OD1	Px OD0
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

对于原子位的设置/清除, 可单独对 GPIOx_ODSET, GPIOx_ODCLR(x=A..D)寄存器操作来实现。

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	PxOD7	端子 Px7 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
6	PxOD6	端子 Px6 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
5	PxOD5	端子 Px5 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
4	PxOD4	端子 Px4 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
3	PxOD3	端子 Px3 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
2	PxOD2	端子 Px2 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
1	PxOD1	端子 Px1 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W
0	PxOD0	端子 Px0 输出值配置位 0: 输出低电平 1: 输出高电平, 如果为开漏输出, 需要配置或外接上拉电阻	0	R/W

10.5.4 GPIO 端口输入数据寄存器(GPIOx_IDR)(x=A..D)

偏移地址: 0x0C

复位值: 0x000000xx

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								Px ID7	Px ID6	Px ID5	Px ID4	Px ID3	Px ID2	Px ID1	Px ID0
								RO	RO	RO	RO	RO	RO	RO	RO

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	PxID7	端子 Px7 输入值 0: 输入低电平 1: 输入高电平	0	RO
6	PxID6	端子 Px6 输入值 0: 输入低电平 1: 输入高电平	0	RO
5	PxID5	端子 Px5 输入值 0: 输入低电平 1: 输入高电平	0	RO
4	PxID4	端子 Px4 输入值 0: 输入低电平 1: 输入高电平	0	RO
3	PxID3	端子 Px3 输入值 0: 输入低电平 1: 输入高电平	0	RO
2	PxID2	端子 Px2 输入值 0: 输入低电平 1: 输入高电平	0	RO
1	PxID1	端子 Px1 输入值 0: 输入低电平 1: 输入高电平	0	RO
0	PxID0	端子 Px0 输入值 0: 输入低电平 1: 输入高电平	0	RO

注：x 表示不定值

10.5.5 GPIO 端口中断使能寄存器(GPIOx_INTEN)(x=A..D)

偏移地址：0x10

复位值：0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PxI EN7	PxI EN6	PxI EN5	PxI EN4	PxI EN3	PxI EN2	PxI EN1	PxI EN0
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	PxIEN7	端子 Px7 中断屏蔽解除位 0: Px7 端子中断屏蔽 1: Px7 端子中断有效	0	R/W
6	PxIEN6	端子 Px6 中断屏蔽解除位 0: Px6 端子中断屏蔽 1: Px6 端子中断有效	0	R/W
5	PxIEN5	端子 Px5 中断屏蔽解除位 0: Px5 端子中断屏蔽 1: Px5 端子中断有效	0	R/W
4	PxIEN4	端子 Px4 中断屏蔽解除位 0: Px4 端子中断屏蔽 1: Px4 端子中断有效	0	R/W
3	PxIEN3	端子 Px3 中断屏蔽解除位 0: Px3 端子中断屏蔽 1: Px3 端子中断有效	0	R/W
2	PxIEN2	端子 Px2 中断屏蔽解除位 0: Px2 端子中断屏蔽 1: Px2 端子中断有效	0	R/W
1	PxIEN1	端子 Px1 中断屏蔽解除位 0: Px1 端子中断屏蔽 1: Px1 端子中断有效	0	R/W
0	PxIEN0	端子 Px0 中断屏蔽解除位 0: Px0 端子中断屏蔽 1: Px0 端子中断有效	0	R/W

10.5.6 GPIO 端口中断原始状态寄存器(GPIOx_RAWINTSR)(x=A..D)

偏移地址: 0x14

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PxR IS7	PxR IS6	PxR IS5	PxR IS4	PxR IS3	PxR IS2	PxR IS1	PxR IS0
								RO	RO	RO	RO	RO	RO	RO	RO

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	PxRIS7	端子 Px7 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
6	PxRIS6	端子 Px6 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
5	PxRIS5	端子 Px5 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
4	PxRIS4	端子 Px4 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
3	PxRIS3	端子 Px3 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
2	PxRIS2	端子 Px2 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
1	PxRIS1	端子 Px1 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO
0	PxRIS0	端子 Px0 中断原始状态位 0: 无中断发生 1: 中断发生	0	RO

10.5.7 GPIO 端口中断状态寄存器(GPIOx_MSKINTSR)(x=A...D)

偏移地址: 0x18

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								Px MIS7	Px MIS6	Px MIS5	Px MIS4	Px MIS3	Px MIS2	Px MIS1	Px MIS0
								RO	RO	RO	RO	RO	RO	RO	RO

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—

位	标记	功能描述	复位值	读写
7	PxMIS7	端子 Px7 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
6	PxMIS6	端子 Px6 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
5	PxMIS5	端子 Px5 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
4	PxMIS4	端子 Px4 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
3	PxMIS3	端子 Px3 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
2	PxMIS2	端子 Px2 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
1	PxMIS1	端子 Px1 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO
0	PxMIS0	端子 Px0 屏蔽后的中断状态位 0: 无中断发生 1: 中断发生	0	RO

10.5.8 GPIO 端口中断清除寄存器(GPIOx_INTCLR)(x=A..D)

偏移地址: 0x1C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PxICL R7	PxICL R6	PxICL R5	PxICL R4	PxICL R3	PxICL R2	PxICL R1	PxICL R0
								WO	WO	WO	WO	WO	WO	WO	WO

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	PxICLR7	写 1 清除端子 Px7 的中断状态 0: 保留中断标志位	0	WO

位	标记	功能描述	复位值	读写
		1: 清除对应的中断标志位		
6	PxICLR6	写 1 清除端子 Px6 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
5	PxICLR5	写 1 清除端子 Px5 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
4	PxICLR4	写 1 清除端子 Px4 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
3	PxICLR3	写 1 清除端子 Px3 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
2	PxICLR2	写 1 清除端子 Px2 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
1	PxICLR1	写 1 清除端子 Px1 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO
0	PxICLR0	写 1 清除端子 Px0 的中断状态 0: 保留中断标志位 1: 清除对应的中断标志位	0	WO

10.5.9 GPIO 端口中断类型寄存器(GPIOx_INTTYPCR)(x=A..D)

偏移地址: 0x20

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PxITY PE7	PxITY PE6	PxITY PE5	PxITY PE4	PxITY PE3	PxITY PE2	PxITY PE1	PxITY PE0
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	PxITYPE7	端子 Px7 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
6	PxITYPE6	端子 Px6 的中断类型配置位	0	R/W

位	标记	功能描述	复位值	读写
		0: 边沿触发中断类型 1: 电平触发中断类型		
5	PxITYPE5	端子 Px5 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
4	PxITYPE4	端子 Px4 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
3	PxITYPE3	端子 Px3 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
2	PxITYPE2	端子 Px2 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
1	PxITYPE1	端子 Px1 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W
0	PxITYPE0	端子 Px0 的中断类型配置位 0: 边沿触发中断类型 1: 电平触发中断类型	0	R/W

10.5.10 GPIO 端口中断类型寄存器(GPIOx_INTPOLCR)(x=A..D)

偏移地址: 0x24

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PxIVA L7	PxIVA L6	PxIVA L5	PxIVA L4	PxIVA L3	PxIVA L2	PxIVA L1	PxIVA L0
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	PxIVAL7	端子 Px7 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
6	PxIVAL6	端子 Px6 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W

位	标记	功能描述	复位值	读写
5	PxIVAL5	端子 Px5 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
4	PxIVAL4	端子 Px4 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
3	PxIVAL3	端子 Px3 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
2	PxIVAL2	端子 Px2 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
1	PxIVAL1	端子 Px1 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W
0	PxIVAL0	端子 Px0 的中断极性配置位 0: 低电平或下降沿触发中断 1: 高电平或上升沿触发中断	0	R/W

10.5.11 GPIO 端口任意边沿触发中断寄存器(GPIOx_INTANY)(x=A..D)

偏移地址: 0x28

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PxIAN Y7	PxIAN Y6	PxIAN Y5	PxIAN Y4	PxIAN Y3	PxIAN Y2	PxIAN Y1	PxIAN Y0
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	PxIANY7	端子 Px7 任意沿触发中断配置位 0: 中断触发沿由 PxIVAL7 决定 1: 上升/下降沿都触发中断	0	R/W
6	PxIANY6	端子 Px6 任意沿触发中断配置位 0: 中断触发沿由 PxIVAL6 决定 1: 上升/下降沿都触发中断	0	R/W
5	PxIANY5	端子 Px5 任意沿触发中断配置位 0: 中断触发沿由 PxIVAL5 决定	0	R/W

位	标记	功能描述	复位值	读写
		1: 上升/下降沿都触发中断		
4	PxIANY4	端子 Px4 任意沿触发中断配置位 0: 中断触发沿由 PxIVAL4 决定 1: 上升/下降沿都触发中断	0	R/W
3	PxIANY3	端子 Px3 任意沿触发中断配置位 0: 中断触发沿由 PxIVAL3 决定 1: 上升/下降沿都触发中断	0	R/W
2	PxIANY2	端子 Px2 任意沿触发中断配置位 0: 中断触发沿由 PxIVAL2 决定 1: 上升/下降沿都触发中断	0	R/W
1	PxIANY1	端子 Px1 任意沿触发中断配置位 0: 中断触发沿由 PxIVAL1 决定 1: 上升/下降沿都触发中断	0	R/W
0	PxIANY0	端子 Px0 任意沿触发中断配置位 0: 中断触发沿由 PxIVAL0 决定 1: 上升/下降沿都触发中断	0	R/W

10.5.12 GPIO 端口输出置位寄存器(GPIOx_ODSET)(x=A..D)

偏移地址: 0x2C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PxOD SET7	PxOD SET6	PxOD SET5	PxOD SET4	PxOD SET3	PxOD SET2	PxOD SET1	PxOD SET0
								WO	WO	WO	WO	WO	WO	WO	WO

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	PxODSET7	端子 Px7 输出置 1 控制位,读取时 0: 输出保持 1: 输出被置高	0	WO
6	PxODSET6	端子 Px6 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
5	PxODSET5	端子 Px5 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
4	PxODSET4	端子 Px4 输出置 1 控制位	0	WO

位	标记	功能描述	复位值	读写
		0: 输出保持 1: 输出被置高		
3	PxODSET3	端子 Px3 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
2	PxODSET2	端子 Px2 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
1	PxODSET1	端子 Px1 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO
0	PxODSET0	端子 Px0 输出置 1 控制位 0: 输出保持 1: 输出被置高	0	WO

10.5.13 GPIO 端口输出清除寄存器(GPIOx_ODCLR)(x=A..D)

偏移地址: 0x30

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PxOD CLR7	PxOD CLR6	PxOD CLR5	PxOD CLR4	PxOD CLR3	PxOD CLR2	PxOD CLR1	PxOD CLR0
								WO	WO	WO	WO	WO	WO	WO	WO

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	PxODCLR7	端子 Px7 输出清 0 控制位 0: 输出保持 1: 复位相应的 ODRx 位	0	WO
6	PxODCLR6	端子 Px6 输出清 0 控制位 0: 输出保持 1: 复位相应的 ODRx 位	0	WO
5	PxODCLR5	端子 Px5 输出清 0 控制位 0: 输出保持 1: 复位相应的 ODRx 位	0	WO
4	PxODCLR4	端子 Px4 输出清 0 控制位 0: 输出保持 1: 复位相应的 ODRx 位	0	WO

位	标记	功能描述	复位值	读写
3	PxODCLR3	端子 Px3 输出清 0 控制位 0: 输出保持 1: 复位相应的 ODRx 位	0	WO
2	PxODCLR2	端子 Px2 输出清 0 控制位 0: 输出保持 1: 复位相应的 ODRx 位	0	WO
1	PxODCLR1	端子 Px1 输出清 0 控制位 0: 输出保持 1: 复位相应的 ODRx 位	0	WO
0	PxODCLR0	端子 Px0 输出清 0 控制位 0: 输出保持 1: 复位相应的 ODRx 位	0	WO

10.5.14 GPIO 端口输入去抖动寄存器(GPIOx_INDBEN)(x=A..D)

偏移地址: 0x34

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							SYNC_EN		PxDIDBn						
							R/W		R/W						

位	标记	功能描述	复位值	读写
31:9	—	保留	0x0	—
8	SYNC_EN	消抖未使能时通过设定该寄存器来配置输入是否使用 2 级同步来消除亚稳态(只对中断有效)。 0: 不使用两级同步 1: 使用两级同步	0	R/W
7	PxDIDBn N=(0~7)	端子 Pxn(n=0~7)消抖使能配置位, 如果输入信号不能被连续 2 个的去抖采样周期采样, 则输入信号被视为信号抖动, 而不会触发中断。仅用于边沿触发“edge-trigger”中断, 不能用于电平触发(“level trigger”)中断。电平模式时为两级同步输入 0: 禁止端口消抖功能 1: 使能端口消抖功能	0	R/W

10.5.15 GPIO 端口输入去抖动时钟配置寄存器(GPIOx_DBCLKCR)(x=A..D)

偏移地址：0x38

复位值：0x00000000

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

位	标记	功能描述	复位值	读写																																		
31:5	—	保留	0x0	—																																		
4	DBCLKEN	是否使能去抖动时钟 0: 不使能去抖动时钟 1: 使能去抖动时钟	0	R/W																																		
3:0	DBCLK_DIV	去抖动采样周期选择 Debounce 时钟是 hclk 的 (2 ^{^dbclk_div[3:0]})分频。消抖采样周期选择:	0	R/W																																		
		<table><tr><th>DBCLK_DIV</th><th>描述</th></tr><tr><td>0x0</td><td>消抖采样 1 个 HCLK 周期 1 次</td></tr><tr><td>0x1</td><td>消抖采样 2 个 HCLK 周期 1 次</td></tr><tr><td>0x2</td><td>消抖采样 4 个 HCLK 周期 1 次</td></tr><tr><td>0x3</td><td>消抖采样 8 个 HCLK 周期 1 次</td></tr><tr><td>0x4</td><td>消抖采样 16 个 HCLK 周期 1 次</td></tr><tr><td>0x5</td><td>消抖采样 32 个 HCLK 周期 1 次</td></tr><tr><td>0x6</td><td>消抖采样 64个 HCLK 周期 1 次</td></tr><tr><td>0x7</td><td>消抖采样 128个 HCLK 周期 1 次</td></tr><tr><td>0x8</td><td>消抖采样256 个 HCLK 周期 1 次</td></tr><tr><td>0x9</td><td>消抖采样512 个 HCLK 周期 1 次</td></tr><tr><td>0xA</td><td>消抖采样1024 个 HCLK 周期 1 次</td></tr><tr><td>0xB</td><td>消抖采样2*1024 个 HCLK 周期 1 次</td></tr><tr><td>0xC</td><td>消抖采样4*1024 个 HCLK 周期 1 次</td></tr><tr><td>0xD</td><td>消抖采样 8*1024 个 HCLK 周期 1 次</td></tr><tr><td>0xE</td><td>消抖采样 16*1024 个 HCLK 周期 1 次</td></tr><tr><td>0xF</td><td>消抖采样 32*1024 个 HCLK 周期 1 次</td></tr></table>			DBCLK_DIV	描述	0x0	消抖采样 1 个 HCLK 周期 1 次	0x1	消抖采样 2 个 HCLK 周期 1 次	0x2	消抖采样 4 个 HCLK 周期 1 次	0x3	消抖采样 8 个 HCLK 周期 1 次	0x4	消抖采样 16 个 HCLK 周期 1 次	0x5	消抖采样 32 个 HCLK 周期 1 次	0x6	消抖采样 64个 HCLK 周期 1 次	0x7	消抖采样 128个 HCLK 周期 1 次	0x8	消抖采样256 个 HCLK 周期 1 次	0x9	消抖采样512 个 HCLK 周期 1 次	0xA	消抖采样1024 个 HCLK 周期 1 次	0xB	消抖采样2*1024 个 HCLK 周期 1 次	0xC	消抖采样4*1024 个 HCLK 周期 1 次	0xD	消抖采样 8*1024 个 HCLK 周期 1 次	0xE	消抖采样 16*1024 个 HCLK 周期 1 次	0xF	消抖采样 32*1024 个 HCLK 周期 1 次
		DBCLK_DIV			描述																																	
		0x0			消抖采样 1 个 HCLK 周期 1 次																																	
		0x1			消抖采样 2 个 HCLK 周期 1 次																																	
		0x2			消抖采样 4 个 HCLK 周期 1 次																																	
		0x3			消抖采样 8 个 HCLK 周期 1 次																																	
		0x4			消抖采样 16 个 HCLK 周期 1 次																																	
		0x5			消抖采样 32 个 HCLK 周期 1 次																																	
		0x6			消抖采样 64个 HCLK 周期 1 次																																	
		0x7			消抖采样 128个 HCLK 周期 1 次																																	
		0x8			消抖采样256 个 HCLK 周期 1 次																																	
		0x9			消抖采样512 个 HCLK 周期 1 次																																	
		0xA			消抖采样1024 个 HCLK 周期 1 次																																	
		0xB			消抖采样2*1024 个 HCLK 周期 1 次																																	
		0xC			消抖采样4*1024 个 HCLK 周期 1 次																																	
		0xD			消抖采样 8*1024 个 HCLK 周期 1 次																																	
0xE	消抖采样 16*1024 个 HCLK 周期 1 次																																					
0xF	消抖采样 32*1024 个 HCLK 周期 1 次																																					

10.5.16 GPIO 端口上拉/下拉寄存器(GPIOx_PUPDR)(x=A..D)

偏移地址: 0x3C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxPUPD7[1:0]		PxPUPD6[1:0]		PxPUPD5[1:0]		PxPUPD4[1:0]		PxPUPD3[1:0]		PxPUPD2[1:0]		PxPUPD1[1:0]		PxPUPD0[1:0]	
R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W	

位	标记	功能描述	复位值	读写
31:16	—	保留	0x0	—
15:14	PxPUPD7	端子 Px7 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
13:12	PxPUPD6	端子 Px6 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
11:10	PxPUPD5	端子 Px5 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
9:8	PxPUPD4	端子 Px4 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
7:6	PxPUPD3	端子 Px3 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
5:4	PxPUPD2	端子 Px2 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W

位	标记	功能描述	复位值	读写
3:2	PxPUPD1	端子 Px1 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W
1:0	PxPUPD0	端子 Px0 上拉/下拉配置控制位 00: 上拉, 下拉禁止 01: 上拉使能 10: 下拉使能 11: 保留	0x0	R/W

10.5.17 GPIO 端口电压转换速率配置(GPIOx_SLEWCR)(x=A..D)

偏移地址: 0x40

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PxSR7	PxSR6	PxSR5	PxSR4	PxSR3	PxSR2	PxSR1	PxSR0
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	PxSR7	端子 Px7 压摆率配置控制位 0: 高压摆率 1: 低压摆率	1	R/W
6	PxSR6	端子 Px6 压摆率配置控制位 0: 高压摆率 1: 低压摆率	1	R/W
5	PxSR5	端子 Px5 压摆率配置控制位 0: 高压摆率 1: 低压摆率	1	R/W
4	PxSR4	端子 Px4 压摆率配置控制位 0: 高压摆率 1: 低压摆率	1	R/W
3	PxSR3	端子 Px3 压摆率配置控制位 0: 高压摆率 1: 低压摆率	1	R/W
2	PxSR2	端子 Px2 压摆率配置控制位 0: 高压摆率	1	R/W

位	标记	功能描述	复位值	读写
		1: 低压摆率		
1	PxSR1	端子 Px1 压摆率配置控制位 0: 高压摆率 1: 低压摆率	1	R/W
0	PxSR0	端子 Px0 压摆率配置控制位 0: 高压摆率 1: 低压摆率	1	R/W

10.5.18 GPIO 端口驱动强度配置寄存器(GPIOx_DRVCR)(x=A..D)

偏移地址: 0x44

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								PxDRV7	PxDRV6	PxDRV5	PxDRV4	PxDRV3	PxDRV2	PxDRV1	PxDRV0
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	PxDRV7	端子 Px7 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
6	PxDRV6	端子 Px6 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
5	PxDRV5	端子 Px5 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
4	PxDRV4	端子 Px4 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
3	PxDRV3	端子 Px3 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
2	PxDRV2	端子 Px2 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W
1	PxDRV1	端子 Px1 驱动强度选择控制位	0	R/W

位	标记	功能描述	复位值	读写
		0: 高驱动强度 1: 低驱动强度		
0	PxDRV0	端子 Px0 驱动强度选择控制位 0: 高驱动强度 1: 低驱动强度	0	R/W

10.5.19 GPIO 端口复用功能寄存器(GPIOx_AFR)(x=A..D)

偏移地址: 0x48

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PxAFR7[3:0]				PxAFR6[3:0]				PxAFR5[3:0]				PxAFR4[3:0]			
R/W				R/W				R/W				R/W			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PxAFR3[3:0]				PxAFR2[3:0]				PxAFR1[3:0]				PxAFR0[3:0]			
R/W				R/W				R/W				R/W			

10.5.19.1 GPIOA 复用配置

位	标记	功能描述	复位值	读写
31:16	—	保留	0x0	—
15:12	PAAFR3	端口 PA3 功能选择 0000: PA3 0001: TIM1_CH3N 0010: PCA_CH2 0011: SPI_NSS 0100: RTC_1HZ 0101: LPUART_RXD 0110: PCA_ECI 0111: VC0_OUT 1000: TIM2_CH3 1001~1111: 保留	0x0	R/W
11:8	PAAFR2	端口 PA2 功能选择 0000: PA2 0001: TIM1_CH3 0010: 保留	0x0	R/W

位	标记	功能描述	复位值	读写
		0011: SPI_NSS 0100: I2C_SCL 0101: UART0_TXD 0110: TIM10_TOGN 0111: UART1_TXD 1000: TIM2_CH2 1001~1111: 保留		
7:4	PAAFR1	端口 PA1 功能选择 0000: PA1 0001: TIM1_CH2N 0010: 保留 0011: SPI_CLK 0100: I2C_SDA 0101: UART0_RXD 0110: TIM10_TOG 0111: UART1_RXD 1000: 保留 1001~1111: 保留	0x0	R/W
3:0	—	保留	0x0	—

10.5.19.2 GPIOB 复用配置

位	标记	功能描述	复位值	读写
31:24	—	保留	0x0	—
23:30	PBAFR5	端口 PB5 功能选择 0000: PB5 0001: TIM1_BKIN 0010: PCA_CH4 0011: SPI_CLK 0100: I2C_SDA 0101: UART0_RXD 0110: TIM11_TOG 0111: LVD_OUT 1000: TIM2_CH1 1001~1111: 保留	0x0	R/W
19:16	PBAFR4	端口 PB4 功能选择 0000: PB4 0001: LPTIM_GATE 0010: PCA_ECI 0011: SPI_NSS	0x0	R/W

位	标记	功能描述	复位值	读写
		0100: I2C_SCL 0101: UART0_TXD 0110: TIM11_TOGN 0111: 保留 1000: 保留 1001~1111: 保留		
15:0	—	保留	0x0	—

10.5.19.3 GPIOC 复用配置

位	标记	功能描述	复位值	读写
31:28	PCAFR7	端口 PC7 功能选择 0000: PC7 0001: TIM1_CH2 0010: PCA_CH4 0011: SPI_MISO 0100: 保留 0101: UART1_RXD 0110: LIRC_OUT 0111: LXT_OUT 1000: 保留 1001~1111: 保留	0x0	R/W
27:24	PCAFR6	端口 PC6 功能选择 0000: PC6 0001: TIM1_CH1 0010: PCA_CH3 0011: SPI_MOSI 0100: 保留 0101: LPUART_RXD 0110: TIM11_EXT 0111: CLK_MCO 1000: TIM2_CH4 1001~1110: 保留 1111: AIN0	0x0	R/W
23:20	PCAFR5	端口 PC5 功能选择 0000: PC5 0001: TIM1_BKIN 0010: PCA_CH0 0011: SPI_CLK	0x0	R/W

位	标记	功能描述	复位值	读写
		0100: 保留 0101: LPUART_TXD 0110: TIM11_GATE 0111: LVD_OUT 1000: TIM2_CH1 1001~1110: 保留 1111: VCIN1		
19:16	PCA4FR4	端口 PC4 功能选择 0000: PC4 0001: TIM1_CH4 0010: TIM1_CH2N 0011: 保留 0100: I2C_SCL 0101: UART1_RXD 0110: PCA_CH0 0111: CLK_MCO1000: TIM2_CH4 1001~1110: 保留 1111: AIN2	0x0	R/W
15:12	PCA3FR3	端口 PC3 功能选择 0000: PC3 0001: TIM1_CH3 0010: TIM1_CH1N 0011: 保留 0100: I2C_SDA 0101: UART1_TXD 0110: PCA_CH10111: 1-WIRE 1000: TIM2_CH3 1001~1110: 保留 1111: AIN1	0x0	R/W
11:0	—	保留	0x0	—

10.5.19.4 GPIOD 复用配置

位	标记	功能描述	复位值	读写
31:28	—	保留	0x0	—
27:24	PDAFR6	端口 PD6 功能选择 0000: PD6 0001: TIM1_CH2	0x0	R/W

位	标记	功能描述	复位值	读写
		0010: PCA_CH3 0011: SPI_MOSI 0100: I2C_SDA 0101: UART1_RXD 0110: LPTIM_EXT 0111: UART0_RXD 1000: TIM2_CH2 1001~1110: 保留 1111: AIN6		
23:20	PDAFR5	端口 PD5 功能选择 0000: PD5 0001: TIM1_CH1N 0010: PCA_CH4 0011: SPI_MISO 0100: I2C_SCL 0101: UART1_TXD 0110: TIM10_GATE 0111: UART0_TXD 1000: TIM2_CH4 1001~1110: 保留 1111: AIN5	0x0	R/W
19:16	PDAFR4	端口 PD4 功能选择 0000: PD4 0001: TIM1_CH1 0010: PCA_CH0 0011: RTC_1HZ 0100: TIM10_TOG 0101: UART0_TXD 0110: TIM10_EXT 0111: BEEP 1000: TIM2_CH1 1001~1110: 保留 1111: VCIN2	0x0	R/W
15:12	PDAFR3	端口 PD3 功能选择 0000: PD3 0001: TIM1_CH3N 0010: PCA_CH1 0011: SPI_MOSI 0100: HXT_OUT 0101: UART0_RXD 0110: LPTIM_TOGN 0111: 保留	0x0	R/W

位	标记	功能描述	复位值	读写
		1000: TIM2_CH2 1001~1110: 保留 1111: AIN4		
11:8	PDAFR2	端口 PD2 功能选择 0000: PD2 0001: TIM1_CH2 0010: PCA_CH2 0011: SPI_MISO 0100: RTC_1HZ 0101: LPUART_TXD 0110: LPTIM_TOG 0111: 1-WIRE 1000: TIM2_CH3 1001~1110: 保留 1111: VCIN0	0x0	R/W
7:4	PDAFR1	端口 PD1 功能选择 0000: PD1 0001: 保留 0010: PCA_ECI 0011: 保留 0100: 保留 0101: UART1_TXD 0110: HIRC_OUT 0111: VC0_OUT 1000: 保留 1001~1110: 保留 1111: 保留	0x0	R/W
3:0	—	保留	0x0	—

11. FLASH 控制器 (FMC)

11.1 FMC 简介

本芯片包含 1 颗 64KByte 容量的 Flash 存储器，包括一个 64sector 的 main array 区域，一个 2sector 的 NVR 区域。每个 sector 的容量为 512Bytes，Flash 的 main array 区域是给用户使用的，可以存放用户开发的程序和数据。一个 sector 用于存放系统配置值，一个 sector 用于存放用户的配置值。本模块支持对 Flash 存储器的擦除、编程以及读取操作。此外，本模块支持对 Flash 存储器擦写的保护，以及控制寄存器的写保护。

11.2 Flash 结构框图

Main Array 区域用于存放用户的代码。

NVR 区域的 sector0~5 这 6 个 sector 用于存放厂家提供的系统 ISP(在系统可编程)代码，用户可以通过 ISP 代码下载开发的应用程序，用户程序不能对 ISP 代码进行读取或者擦除操作。NVR 的 Sector6 是用户配置参数区域，用于给用户存放重要的信息和配置数据。

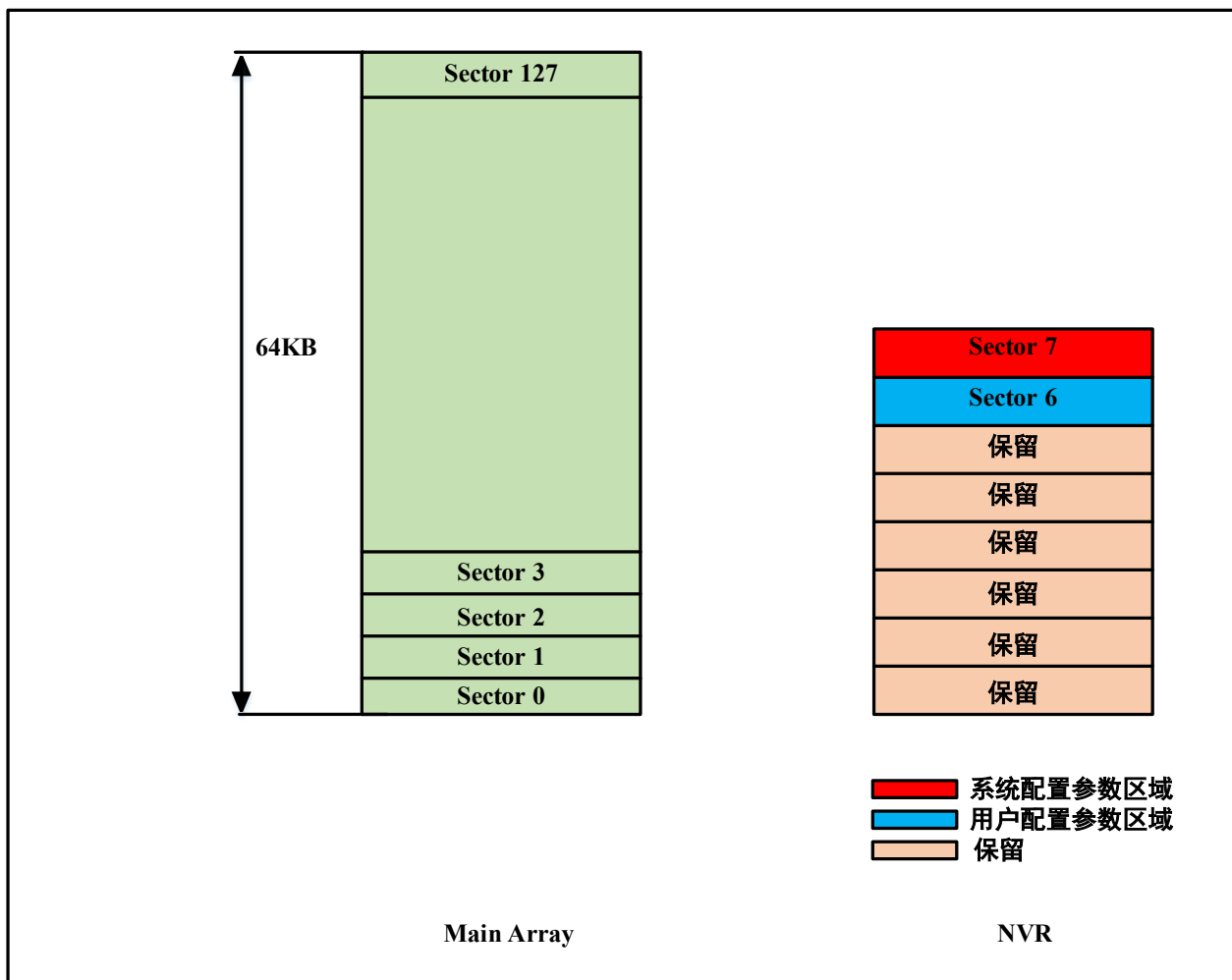


图 11-1. Flash 结构框图

11.3 功能描述

本控制器支持对 Flash 的 byte(8bits)、half-word(16bits)、word(32bits)三种位宽读写操作。注意，byte 操作的地址必须按 byte 对齐，half-word 操作的目标地址必须按 half-word 对齐(地址最低位为 0)，word 操作的目标地址必须按 word 对齐(地址最低两位为 0)。如果读写操作的地址没有按照位宽规定对齐，该操作无效，并且系统会进入 hard fault 出错中断。

11.3.1 擦除操作

11.3.1.1 sector 擦除

sector 擦除操作步骤如下图所示：

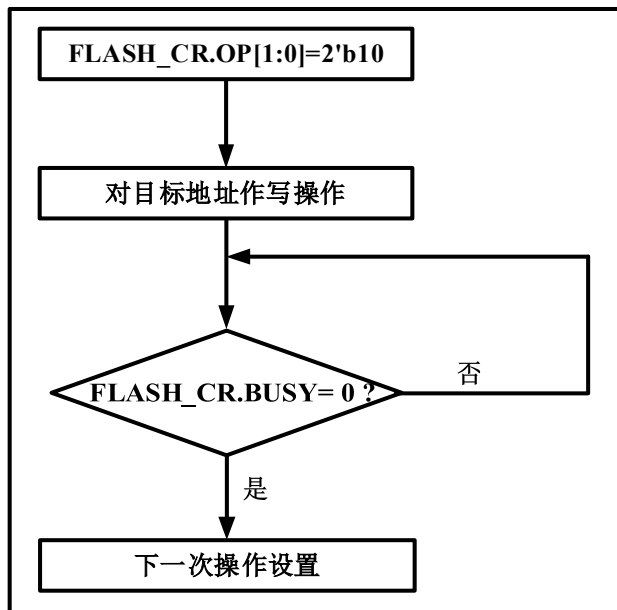


图 11-2. sector 擦除操作步骤

注意：

- 1) 控制器忽略目标地址的低 9 位，只要目标地址落在该页地址范围之内即可。
- 2) 该写操作用于触发页擦除操作，所写数据可以是任意数据。
- 3) 如果当前 sector 擦除指令是在 Flash 内执行，则 CPU 会暂停取指令和执行指令的动作，自动等待 Flash 的 BUSY 状态结束后继续取指令并执行。
- 4) 如果当前 sector 擦除指令是在 RAM 内执行，则 CPU 不会停止取指和执行，在对 Flash 进行任何操作之前，软件必须判断 Flash 的 BUSY 状态是否结束。

11.3.1.2 Chip 擦除

Chip 擦除操作步骤如下图所示：

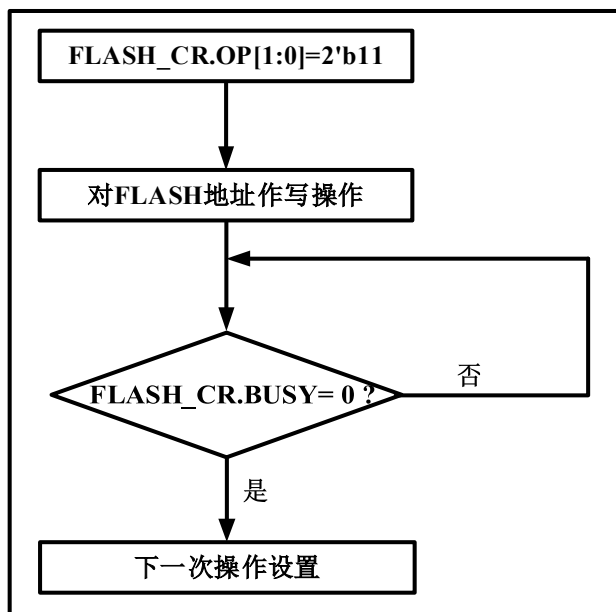


图 11-3. Chip 擦除操作步骤

注意：

- 1) chip 擦除只能针对 Flash 的 main array 操作，NVR 区域只能进行 sector 擦除；
- 2) 控制器忽略目标地址的低 15 位，只要目标地址落在 Flash 地址范围之内即可；
- 3) 该写操作用于触发 chip 擦除操作，所写数据可以是任意数据；
- 4) 如果当前 chip 擦除指令是在 Flash 内执行，则 CPU 会暂停取指令和执行指令的动作，自动等待 Flash 的 BUSY 状态结束后继续取指令并执行；

如果当前 chip 擦除指令是在 RAM 内执行，则 CPU 不会停止取指和执行，在对 Flash 进行任何操作之前，软件必须判断 Flash 的 BUSY 状态是否结束。

11.3.2 写操作

写操作步骤如下图所示：

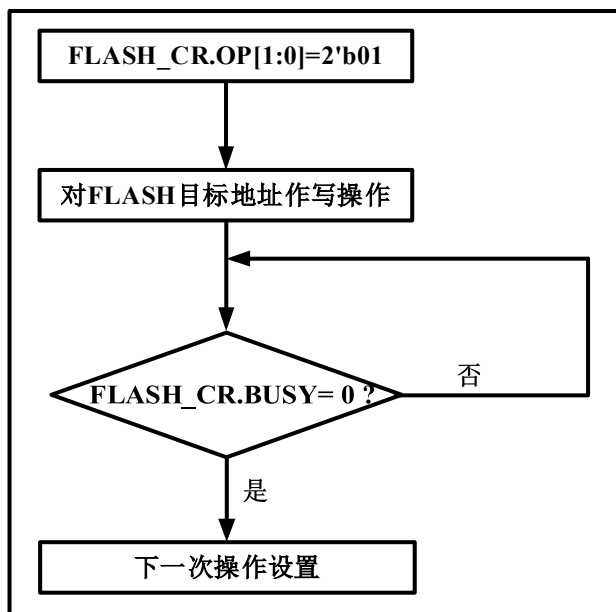


图 11-4. 写操作步骤

注意：

- 1) 如果当前写操作指令是在 Flash 内执行，则 CPU 会暂停取指令和执行指令的动作，自动等待 Flash 的 BUSY 状态结束后继续取指令并执行。
- 2) 如果当前写操作指令是在 RAM 内执行，则 CPU 不会停止取指和执行，在对 Flash 进行任何操作之前，软件必须判断 Flash 的 BUSY 状态是否结束。

11.3.3 读操作

读 Flash 操作和 CPU 读 SRAM 存储器的操作一样，无特殊要求。本芯片片内 Flash 的最大读出速度为 35ns，片内 OSC24M trimming 后精度为 ±1%，折算成 CPU 单周期最快读取时间为 41.3ns，因此本芯片满足在 CPU 使用片内 OSC24M 作为时钟源时对 Flash 的单周期读操作。

11.3.4 擦写时间

Flash 存储器对擦除和编程操作的控制信号都有严格的时间要求，控制信号的有效时间超出设计要求会带来擦除和编程操作失败，本芯片设置了 Flash sector 擦除时间寄存器 (Flash_TSERASE)、Flash chip 擦除时间寄存器 (Flash_TMERASE)、Flash 编程时间寄存器 (Flash_TPROG) 三个寄存器可以对分别 sector、chip 擦除以及编程(写)时间进行调整。

注：芯片在生产阶段会对 Flash 的擦除和编程的时间参数进行测试并写入系统参数配置区域，在芯片上电后电路会自动把这些参数加载到 sector、chip 擦除时间寄存器和编程时间寄存器中，因此一般情况下不建议改动这些已设置好的擦写时间参数。

11.3.5 擦写保护

11.3.5.1 擦写保护位

整个 64kbyteFlash 存储器被划分为 128 个 sector，为了防止应用中的意外擦写操作改变 Flash 中的内容，一共设置了 64 个擦写保护位，每个擦写保护位负责保护 2 个 sector 的区域。保护位寄存器 Flash_SLOCK.SLOCK0/1[31:0]默认值为“0000_0000”，即不允许擦写。只有把相应保护位修改为“1”，该 sector 才能允许被擦写。当 Flash 存储器中的任意 sector 被擦写保护时，对该 Flash 的 chip 擦写也会被控制器自动屏蔽，并且会产生报警标志位和中断信号。如需进行 chip 擦写，必须修改保护位寄存器 Flash_SLOCK.SLOCK0/1[31:0]的值为“FFFF_FFFF”。

11.3.5.2 PC 地址擦写保护

CPU 在 Flash 中运行程序时会遇到一种情况：当前运行的 PC 指针正好落在软件擦写 Flash 的 sector 地址范围之内，那么该擦写操作也会被控制器自动屏蔽，并且会产生报警标志位和中断信号。

11.3.5.3 寄存器写保护

为了防止应用时意外的 Flash 擦写操作改变 Flash 的内容，对 Flash 控制器寄存器的写操作和擦写操作时对 Flash 的写操作必须用写序列方式才能修改。具体操作步骤如下图所示：

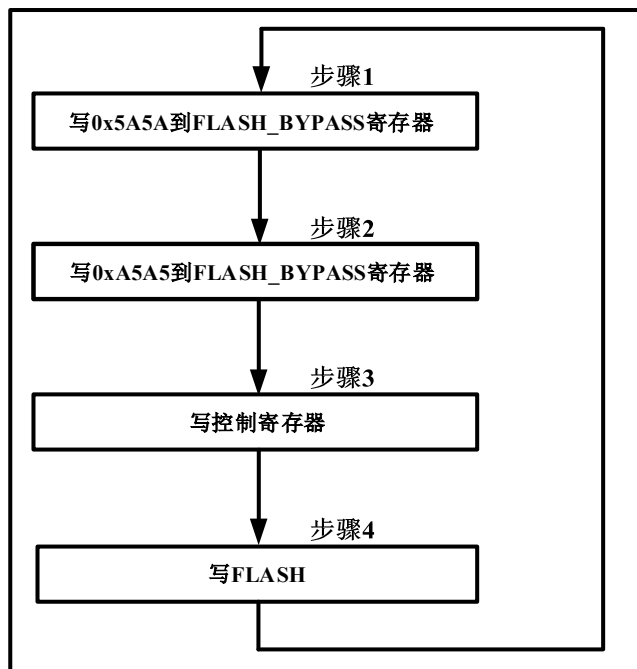


图 11-5. 写操作步骤

注意：步骤 1~4 操作之间不可插入任何其他写操作，否则该 BYPASS 序列操作失效，需要重新写入 0x5A5A-0xA5A5 序列。

11.3.6 系统 BOOT 地址映射

11.3.6.1 APP 程序 boot

下图是 APP 程序区 boot 的地址映射图：

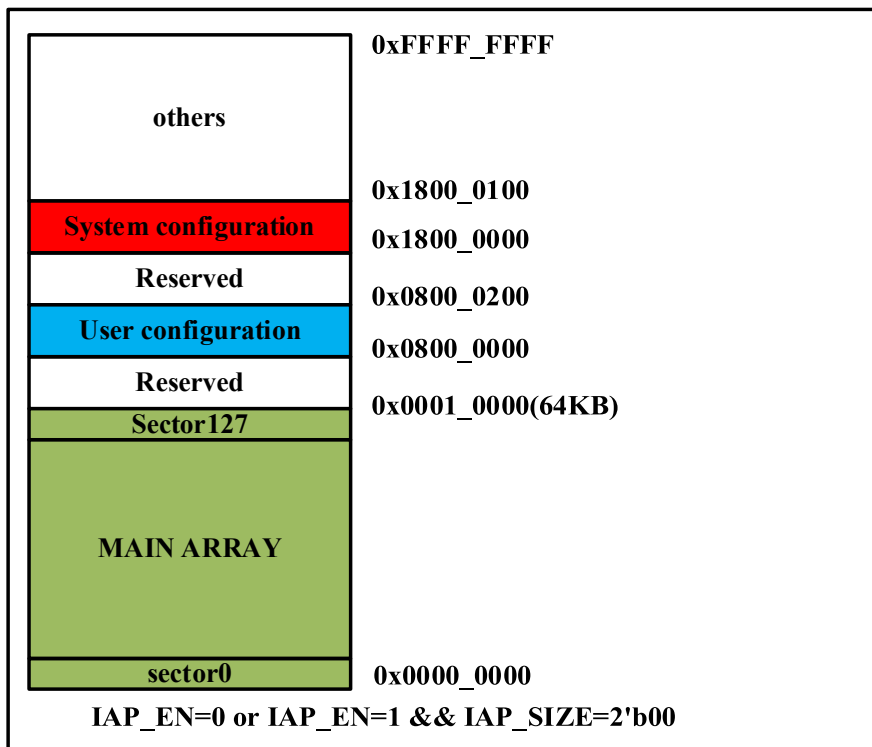


图 11-6. APP 程序区 boot 的地址映射图

APP 程序区 boot 状态也是芯片提供给客户的初始状态。此时 main array 被映射到逻辑地址 0x0000_0000 开始的 64Kbyte 区域，用户配置区域被映射到逻辑地址 0x0800_0000 开始的 512byte 区域，系统配置区域的低 256byte 被映射到 0x1800_0000 开始的 256byte，程序对这 256byte 只能读，不能擦写。

11.4 寄存器列表

基地址：0x40020400

偏移地址	名称	描述	默认值
0x00	Flash_CR	控制寄存器	0x00000000
0x04	Flash_IFR	中断标志寄存器	0x00000000
0x08	Flash_ICLR	中断标志清除寄存器	0x00000000
0x0C	Flash_BYPASS	0x5A5A-0xA5A5 序列寄存器	0x00000000
0x10	Flash_SLOCK0	sector 擦写保护寄存器 0	0x00000000
0x14	Flash_SLOCK1	sector 擦写保护寄存器 1	0x00000000
0x18	Flash_ISPCON	FlashISP 控制寄存器	0x00000001

注：所有 Flash 寄存器只能按 word 方式读写。

11.5 寄存器说明

11.5.1 Flash_CR(Flash 控制寄存器)

地址偏移: 0x00

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											IE[1:0]		BUSY	OP[1:0]	
											R/W		RO	R/W	

位	标记	功能描述	复位值	读写
31:5	—	保留, 始终读为 0。	0	—
4:3	IE[1:0]	IE[1]: Flash 擦写被保护地址中断使能; 0: 不使能; 1: 使能。 IE[0]: Flash 擦写 PC 值中断使能; 0: 不使能; 1: 使能。	2'b00	R/W
2	BUSY	空闲/忙标志位 0: 空闲状态 1: 忙状态	1'b0	RO
1:0	OP[1:0]	Flash 操作模式寄存器 2'b00: read 2'b01: program 2'b10: sector erase 2'b11: chip erase	2'b00	R/W

11.5.2 Flash_CR(Flash 控制寄存器)

地址偏移: 0x04

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													IF1	IF0	
													RO	RO	

位	标记	功能描述	复位值	读写
31:2	–	保留，始终读为 0。	0	–
1	IF1	擦写保护报警中断标志位	1'b0	RO
0	IF0	擦写 PC 地址报警中断标志位	1'b0	RO

11.5.3 Flash_ICLR(Flash 中断标记清除寄存器)

地址偏移：0x08

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														ICLR1	ICLR0
														WO	WO

位	标记	功能描述	复位值	读写
31:2	–	保留，始终读为 0。	0	–
1	ICLR1	清除保护报警中断标志位 写 0 清除；写 1 无效	–	WO
0	ICLR0	清除 PC 地址报警中断标志位 写 0 清除；写 1 无效	–	WO

11.5.4 Flash_BYPASS(BYPASS 系列寄存器)

地址偏移：0x0C

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:16	–	保留，始终读为 0。	0	–

位	标记	功能描述	复位值	读写
15:0	BYPASSSEQ[15:0]	修改本模块寄存器之前必须对 BYPASSSEQ[15:0]寄存器写入 0x5A5A-0xA5A5 序列。一次正确写入序列后只允许写一次寄存器，如需再次修改寄存器必须再次输入正确的 BYPASS 序列	16'h0000	WO

11.5.5 Flash_SLOCK0(sector 擦写保护寄存器 0)

地址偏移：0x10

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOCK0[31:16]															
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK0[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31	SLOCK0[31]	对应：sector62-63 0：不允许擦写 1：允许擦写	1'b0	R/W
30	SLOCK0[30]	对应：sector60-61 0：不允许擦写 1：允许擦写	1'b0	R/W
29	SLOCK0[29]	对应：sector58-59 0：不允许擦写 1：允许擦写	1'b0	R/W
28	SLOCK0[28]	对应：sector56-57 0：不允许擦写 1：允许擦写	1'b0	R/W
27	SLOCK0[27]	对应：sector54-55 0：不允许擦写 1：允许擦写	1'b0	R/W
26	SLOCK0[26]	对应：sector52-53 0：不允许擦写 1：允许擦写	1'b0	R/W
25	SLOCK0[25]	对应：sector50-51 0：不允许擦写 1：允许擦写	1'b0	R/W
24	SLOCK0[24]	对应：sector48-49 0：不允许擦写 1：允许擦写	1'b0	R/W

位	标记	功能描述	复位值	读写
23	SLOCK0[23]	对应: sector46-47 0: 不允许擦写 1: 允许擦写	1'b0	R/W
22	SLOCK0[22]	对应: sector44-45 0: 不允许擦写 1: 允许擦写	1'b0	R/W
21	SLOCK0[21]	对应: sector42-43 0: 不允许擦写 1: 允许擦写	1'b0	R/W
20	SLOCK0[20]	对应: sector40-41 0: 不允许擦写 1: 允许擦写	1'b0	R/W
19	SLOCK0[19]	对应: sector38-39 0: 不允许擦写 1: 允许擦写	1'b0	R/W
18	SLOCK0[18]	对应: sector36-37 0: 不允许擦写 1: 允许擦写	1'b0	R/W
17	SLOCK0[17]	对应: sector34-35 0: 不允许擦写 1: 允许擦写	1'b0	R/W
16	SLOCK0[16]	对应: sector32-33 0: 不允许擦写 1: 允许擦写	1'b0	R/W
15	SLOCK0[15]	对应: sector30-31 0: 不允许擦写 1: 允许擦写	1'b0	R/W
14	SLOCK0[14]	对应: sector28-29 0: 不允许擦写 1: 允许擦写	1'b0	R/W
13	SLOCK0[13]	对应: sector26-27 0: 不允许擦写 1: 允许擦写	1'b0	R/W
12	SLOCK0[12]	对应: sector24-25 0: 不允许擦写 1: 允许擦写	1'b0	R/W
11	SLOCK0[11]	对应: sector22-23 0: 不允许擦写 1: 允许擦写	1'b0	R/W
10	SLOCK0[10]	对应: sector20-21 0: 不允许擦写 1: 允许擦写	1'b0	R/W
9	SLOCK0[9]	对应: sector18-19 0: 不允许擦写 1: 允许擦写	1'b0	R/W

位	标记	功能描述	复位值	读写
8	SLOCK0[8]	对应: sector16-17 0: 不允许擦写 1: 允许擦写	1'b0	R/W
7	SLOCK0[7]	对应: sector14-15 0: 不允许擦写 1: 允许擦写	1'b0	R/W
6	SLOCK0[6]	对应: sector12-13 0: 不允许擦写 1: 允许擦写	1'b0	R/W
5	SLOCK0[5]	对应: sector10-11 0: 不允许擦写 1: 允许擦写	1'b0	R/W
4	SLOCK0[4]	对应: sector8-9 0: 不允许擦写 1: 允许擦写	1'b0	R/W
3	SLOCK0[3]	对应: sector6-7 0: 不允许擦写 1: 允许擦写	1'b0	R/W
2	SLOCK0[2]	对应: sector4-5 0: 不允许擦写 1: 允许擦写	1'b0	R/W
1	SLOCK0[1]	对应: sector2-3 0: 不允许擦写 1: 允许擦写	1'b0	R/W
0	SLOCK0[0]	对应: sector0-1 0: 不允许擦写 1: 允许擦写	1'b0	R/W

11.5.6 Flash_SLOCK1(sector 擦写保护寄存器 1)

地址偏移: 0x14

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOCK1[31:16]															
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLOCK1[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31	SLOCK1[31]	对应: sector126-127	1'b0	R/W

位	标记	功能描述	复位值	读写
		0: 不允许擦写 1: 允许擦写		
30	SLOCK1[30]	对应: sector124-125 0: 不允许擦写 1: 允许擦写	1'b0	R/W
29	SLOCK1[29]	对应: sector122-123 0: 不允许擦写 1: 允许擦写	1'b0	R/W
28	SLOCK1[28]	对应: sector120-121 0: 不允许擦写 1: 允许擦写	1'b0	R/W
27	SLOCK1[27]	对应: sector118-119 0: 不允许擦写 1: 允许擦写	1'b0	R/W
26	SLOCK1[26]	对应: sector116-117 0: 不允许擦写 1: 允许擦写	1'b0	R/W
25	SLOCK1[25]	对应: sector114-115 0: 不允许擦写 1: 允许擦写	1'b0	R/W
24	SLOCK1[24]	对应: sector112-113 0: 不允许擦写 1: 允许擦写	1'b0	R/W
23	SLOCK1[23]	对应: sector110-111 0: 不允许擦写 1: 允许擦写	1'b0	R/W
22	SLOCK1[22]	对应: sector108-109 0: 不允许擦写 1: 允许擦写	1'b0	R/W
21	SLOCK1[21]	对应: sector106-107 0: 不允许擦写 1: 允许擦写	1'b0	R/W
20	SLOCK1[20]	对应: sector104-105 0: 不允许擦写 1: 允许擦写	1'b0	R/W
19	SLOCK1[19]	对应: sector102-103 0: 不允许擦写 1: 允许擦写	1'b0	R/W
18	SLOCK1[18]	对应: sector100-101 0: 不允许擦写 1: 允许擦写	1'b0	R/W
17	SLOCK1[17]	对应: sector98-99 0: 不允许擦写 1: 允许擦写	1'b0	R/W
16	SLOCK1[16]	对应: sector96-97 0: 不允许擦写	1'b0	R/W

位	标记	功能描述	复位值	读写
		1: 允许擦写		
15	SLOCK1[15]	对应: sector94-95 0: 不允许擦写 1: 允许擦写	1'b0	R/W
14	SLOCK1[14]	对应: sector92-93 0: 不允许擦写 1: 允许擦写	1'b0	R/W
13	SLOCK1[13]	对应: sector90-91 0: 不允许擦写 1: 允许擦写	1'b0	R/W
12	SLOCK1[12]	对应: sector88-89 0: 不允许擦写 1: 允许擦写	1'b0	R/W
11	SLOCK1[11]	对应: sector86-87 0: 不允许擦写 1: 允许擦写	1'b0	R/W
10	SLOCK1[10]	对应: sector84-85 0: 不允许擦写 1: 允许擦写	1'b0	R/W
9	SLOCK1[9]	对应: sector82-83 0: 不允许擦写 1: 允许擦写	1'b0	R/W
8	SLOCK1[8]	对应: sector80-81 0: 不允许擦写 1: 允许擦写	1'b0	R/W
7	SLOCK1[7]	对应: sector78-79 0: 不允许擦写 1: 允许擦写	1'b0	R/W
6	SLOCK1[6]	对应: sector76-77 0: 不允许擦写 1: 允许擦写	1'b0	R/W
5	SLOCK1[5]	对应: sector74-75 0: 不允许擦写 1: 允许擦写	1'b0	R/W
4	SLOCK1[4]	对应: sector72-73 0: 不允许擦写 1: 允许擦写	1'b0	R/W
3	SLOCK1[3]	对应: sector70-71 0: 不允许擦写 1: 允许擦写	1'b0	R/W
2	SLOCK1[2]	对应: sector68-69 0: 不允许擦写 1: 允许擦写	1'b0	R/W
1	SLOCK1[1]	对应: sector66-67 0: 不允许擦写	1'b0	R/W

位	标记	功能描述	复位值	读写
		1: 允许擦写		
0	SLOCK1[0]	对应: sector64-65 0: 不允许擦写 1: 允许擦写	1'b0	R/W

注：对于 Flash_FULLSIZE 配置为 0，即 Flash 的容量被配置为 32Kbyte 时，Flash_SLOCK1 寄存器的设置值均无效。

12. CRC

12.1 概述

循环冗余校验(CRC)计算单元是根据固定的生成多项式得到任意字节数据的 CRC 计算结果。在应用中, CRC 技术主要应用于核实数据传输或者数据存储的正确性和完整性。以下示意了 CRC 算法在数据传输中的一个典型的应用:

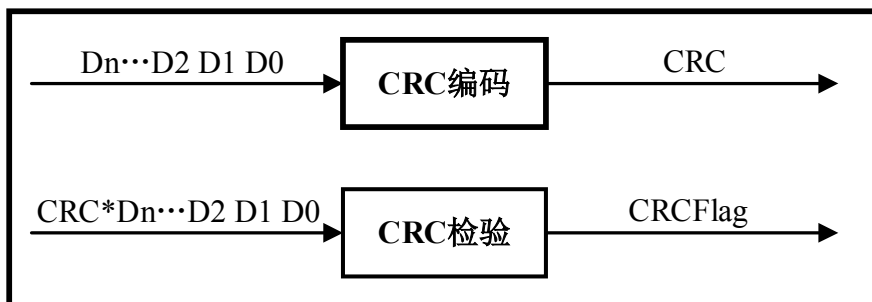


图 12-1. CRC 应用示意图

12.2 功能描述

本模块算法遵从 ISO/IEC13239 的定义, 采用 16 位长度的 CRC, 计算多项式为:

$$x^{16} + x^{12} + x^5 + x$$

计算初始值为 0xFFFF。本模块功能包括:

- CRC 编码和 CRC 校验
- 3 种位宽访问方式 8 位、16 位、32 位
- 8 位位宽下输入数据示例为 0x00,0x11,0x22,0x33,0x44,0x55,0x66,0x77
- 16 位位宽下输入数据示例为 0x1100,0x3322,0x5544,0x7766
- 32 位位宽下输入数据示例为 0x33221100,0x77665544

12.2.1 CRC 编码模式

编码模式可以对原始数据编码以计算其 CRC 值, 操作流程如下所示:

Step1: 向 CRC_RESULT.CRC16 写入 0xFFFF, 初始化 CRC 计算。

Step2: 将待编码的原始数据按 8 位/16 位/32 位的组织方式, 依次写入 CRC_DATA 寄存器。

Step3: 读取 CRC_RESULT.CRC16, 即为 CRC 值。

12.2.2 CRC 检验模式

检验模式可以检验已编码的数据是否被篡改, 操作流程如下所示

Step1: 向 CRC_RESULT[15:0]写入 0xFFFF, 初始化 CRC 计算。

Step2: 将已编码的数据按 8 位/16 位/32 位的组织方式, 依次写入 CRC_DATA 寄存器。注:
按 8 位组织方式写 CRC 值到 CRC_DATA 寄存器时, 应先写入低 8 位, 后写入高 8 位。

Step3: 读取 CRC_RESULT.FLAG, 以判定 CRC 校验是否成功。

12.3 寄存器列表

基地址: 0x40020800

表 12.1. CRC 寄存器列表和复位值

偏移地址	名称	描述	复位值
0x04	CRC_RESULT	CRC 结果寄存器, 计算完成后对该寄存器读取即获得结果。	0x00000000
0x80-0xFF	CRC_DATA	CRC 数据寄存器, 用于输入需要运算的数据。	0x00000000

12.4 寄存器说明

12.4.1 CRC 结果寄存器(CRC_RESULT)

地址偏移: 0x04

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:17	—	保留	0x0	
16	FLAG	校验结果标志： 0：当前校验错误，1：当前校验正确。 寄存器[16]是一个只读位，对其进行写操作不会产生影响。进行 CRC 校验时，应该在所有的数据和 16 位 CRC 编码输入数据寄存器之后读取本位，如果为 1 则表明校验成功。	0x0	R/W
15:0	RESULT	本寄存器用于每次 CRC 计算结果的更新和保存。运算后，读取本寄存器将得到 16 位的 CRC 编码结果。根据标准规定，运算完成后，16 位的 CRC 编码值是运算寄存器取反后的结果，因此本寄存器[15:0]的读取将得到本寄存器当前[15:0]的取反值。	0x0000	R/W

注意：

- 1.根据标准规定，运算完成后，16 位的 CRC 编码值是运算寄存器取反后的结果，因此，对本寄存器位 15~0 的读取将得到本寄存器当前位 15~0 的取反值。
- 2.CRC_RESULT[16]是一个只读的位，对之进行写操作不会产生影响。进行 CRC 校验运算时，应该在所有的数据和 16 位 CRC 编码输入数据寄存器后读取本位，如果为 1 则表明校验成功。

12.4.2 CRC 数据寄存器(CRC_DATA)

地址偏移：0x80-0xFF

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:0	CRC_DATA	本寄存器用于输入需要运算的数据。 本寄存器的地址是一个范围(0x80-0xFF)，对该范围内的任何一个地址进行操作都会认为是对本寄存器进行操作。这样定义的目的就是为了方便软件可以用 STM 指令对本寄存器进行连续的 32 位数据写入操作，以加快运算速度。本寄存器支持 8/16/32 位的输入方式。	0x0	R/W

注意：本寄存器的地址是一个范围(8'h80-8'hFF)，对该范围内的任何一个地址进行操作都会认为是对本寄存器进行操作。这样定义的目的就是方便软件可以用 STM 指令对本寄存器进行连续的 32 位数据写入操作，以加快运算速度。同时本寄存器也支持 8/16 位的输入方式。

13. 高级控制定时器(TIM1)

13.1 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。

13.2 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:
 - 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

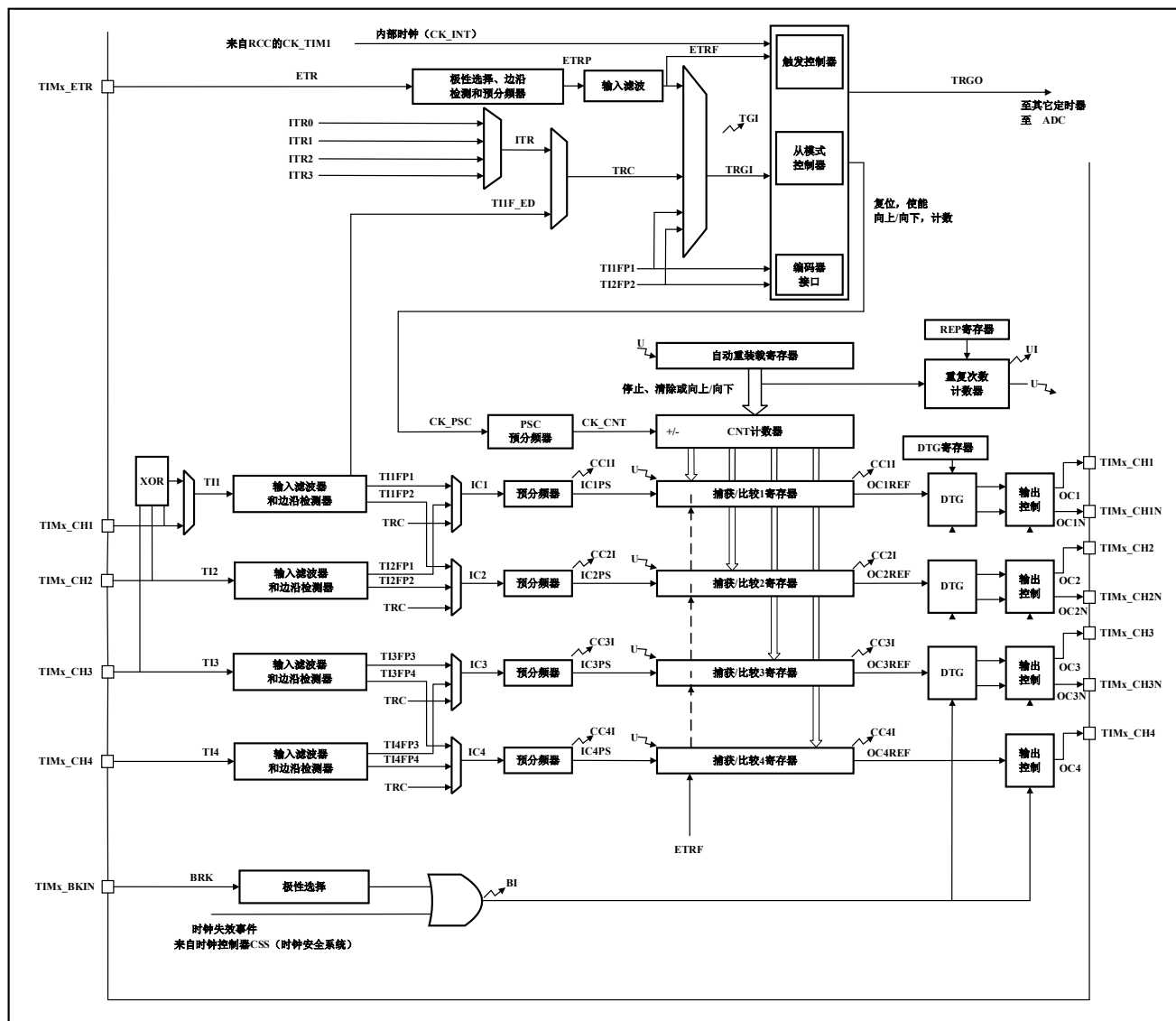


图 13-1. 高级控制定时器框图

注：控制位的设定，在 U(更新)事件时传送预加载寄存器的内容至工作寄存器

事件
 中断

13.3 TIM1 功能描述

13.3.1 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，写或读自动重装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。

13.3.1.1 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 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。

图 13-2 和图 13-3 给出了在预分频器运行时，更改计数器参数的例子。

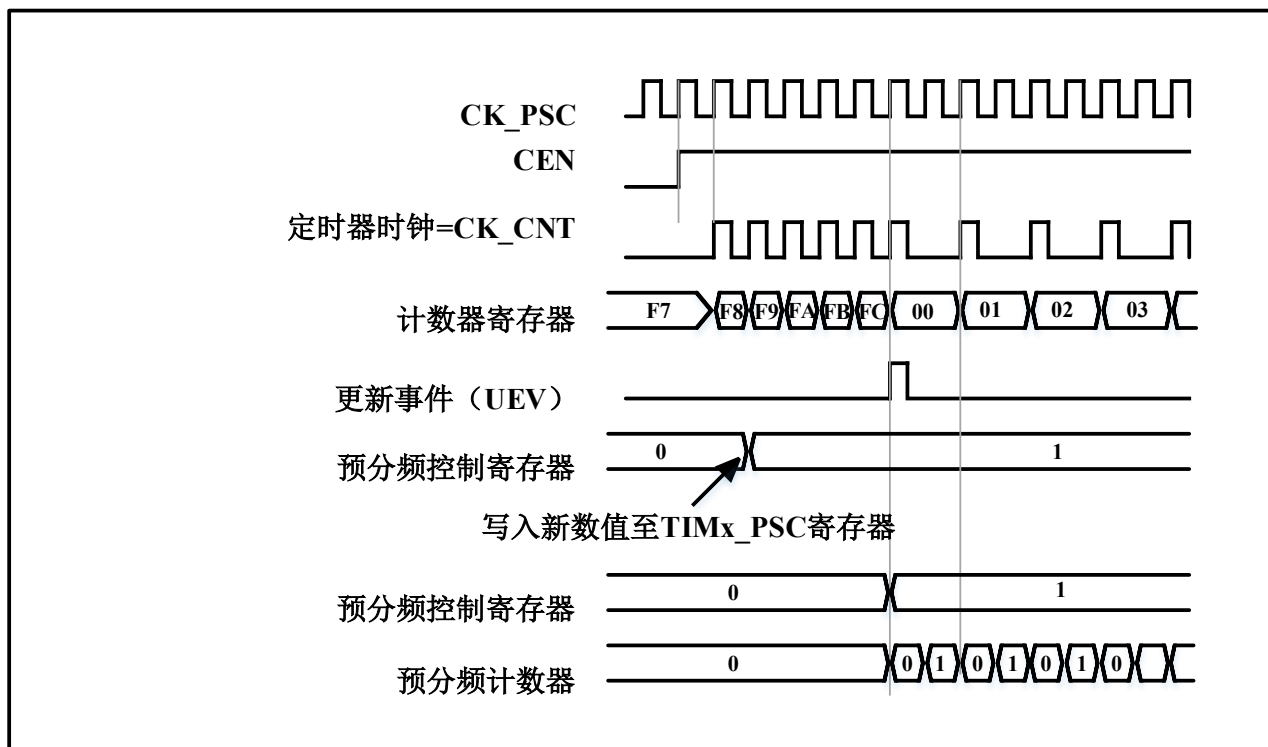


图 13-2. 当预分频器的参数从 1 变到 2 时，计数器的时序图

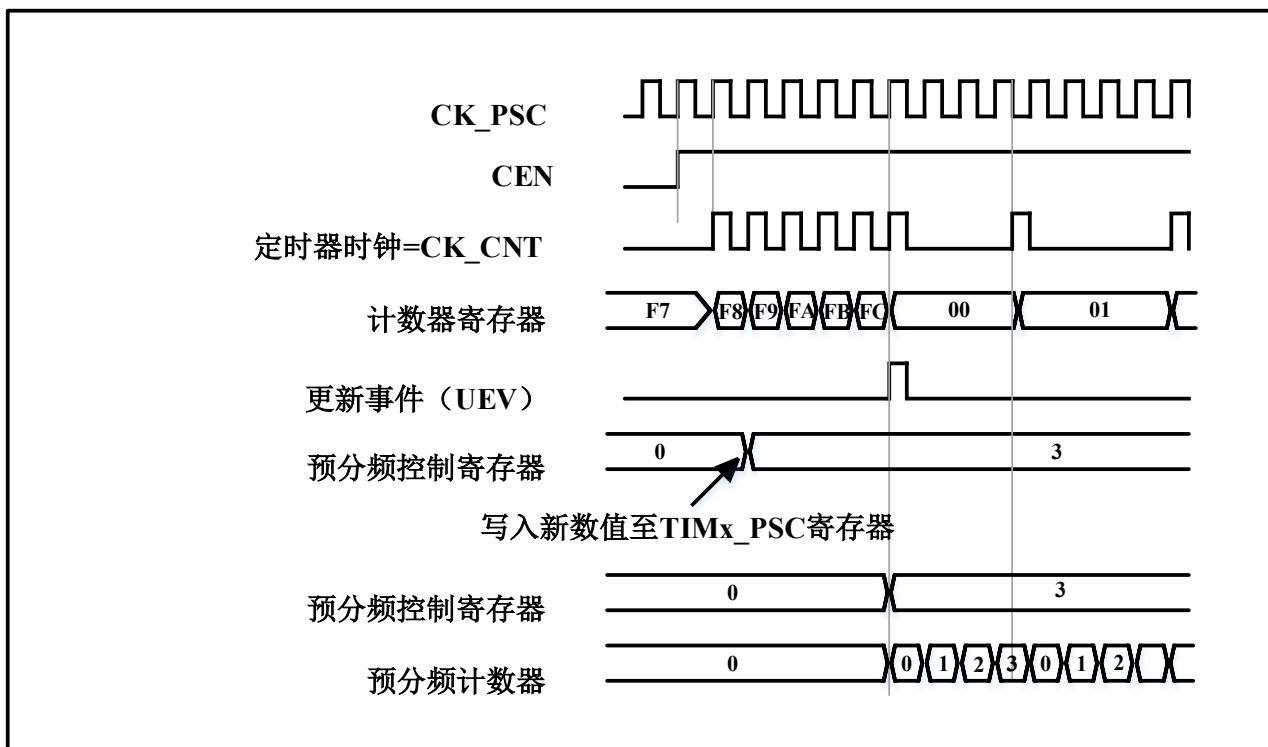


图 13-3. 当预分频器的参数从 1 变到 4 时，计数器的时序图

13.3.2 计数器模式

13.3.2.1 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(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)。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。

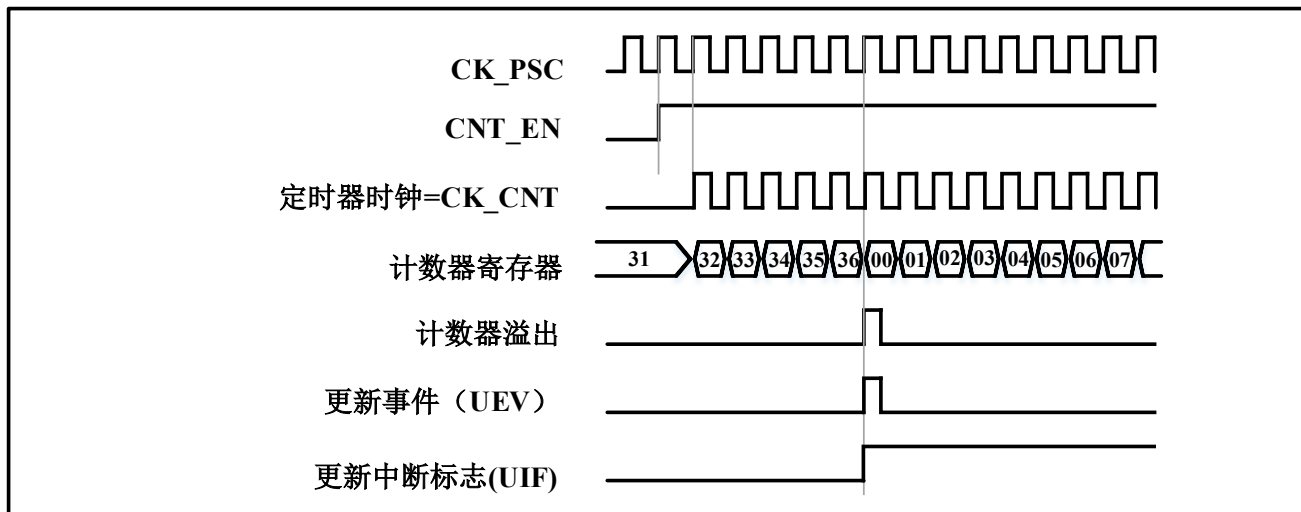


图 13-4. 计数器时序图，内部时钟分频因子为 1

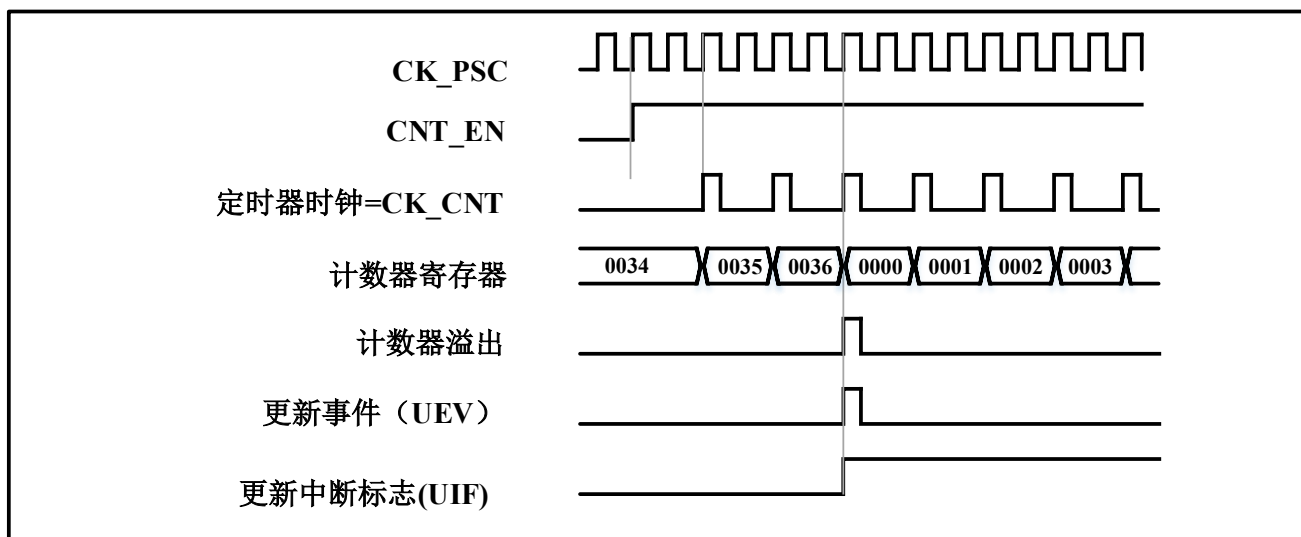


图 13-5. 计数器时序图，内部时钟分频因子为 2

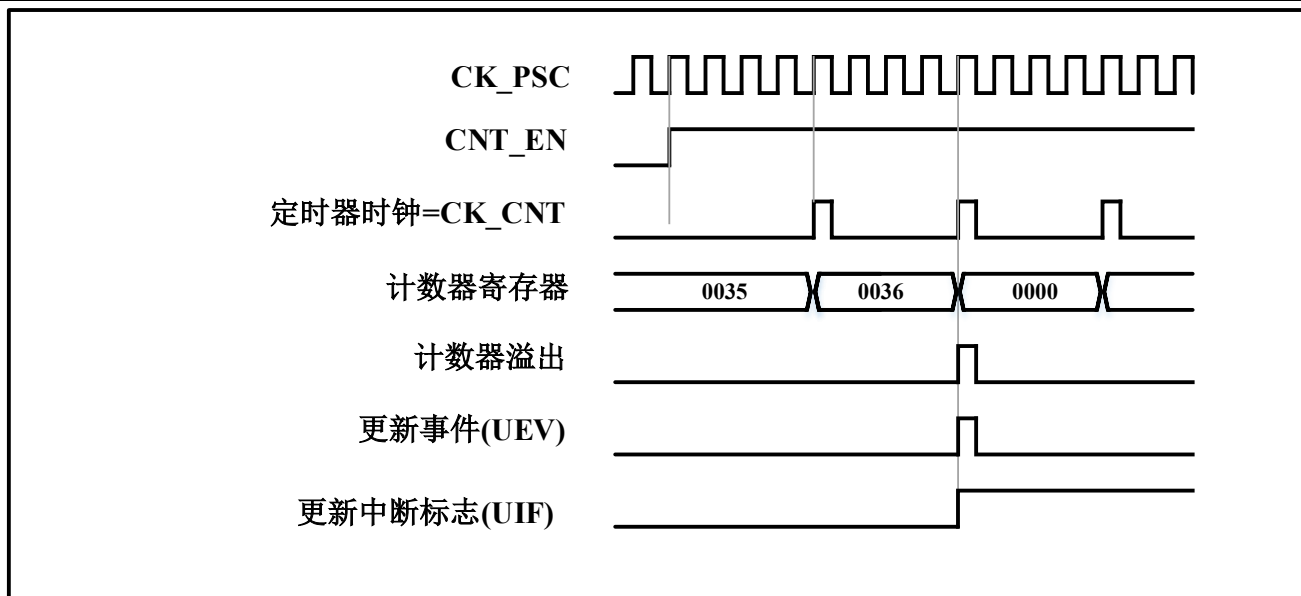


图 13-6. 计数器时序图，内部时钟分频因子为 4

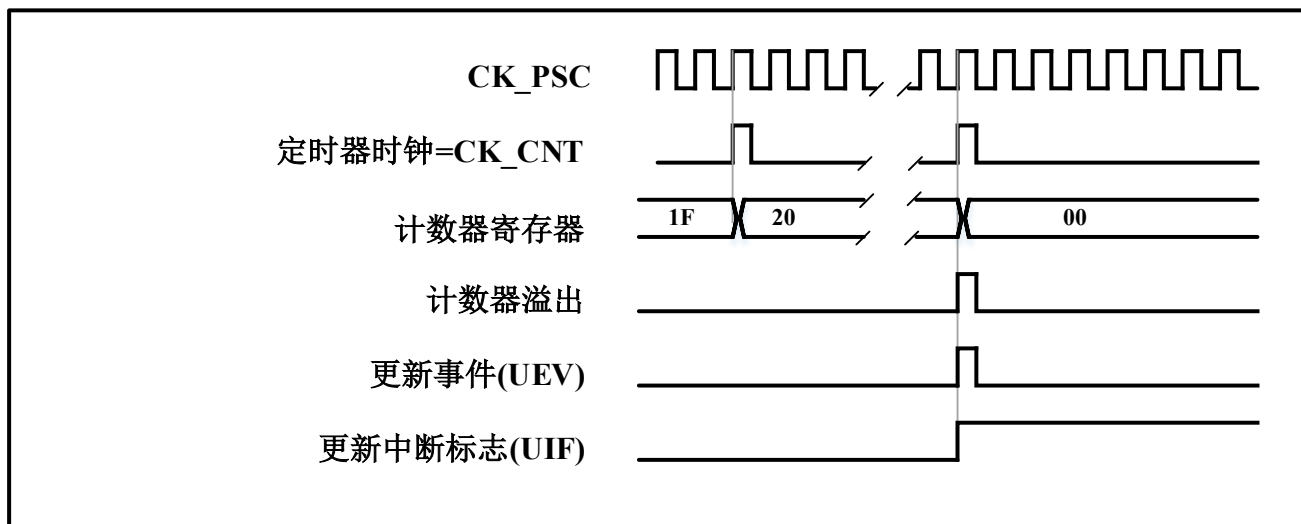


图 13-7. 计数器时序图，内部时钟分频因子为 N

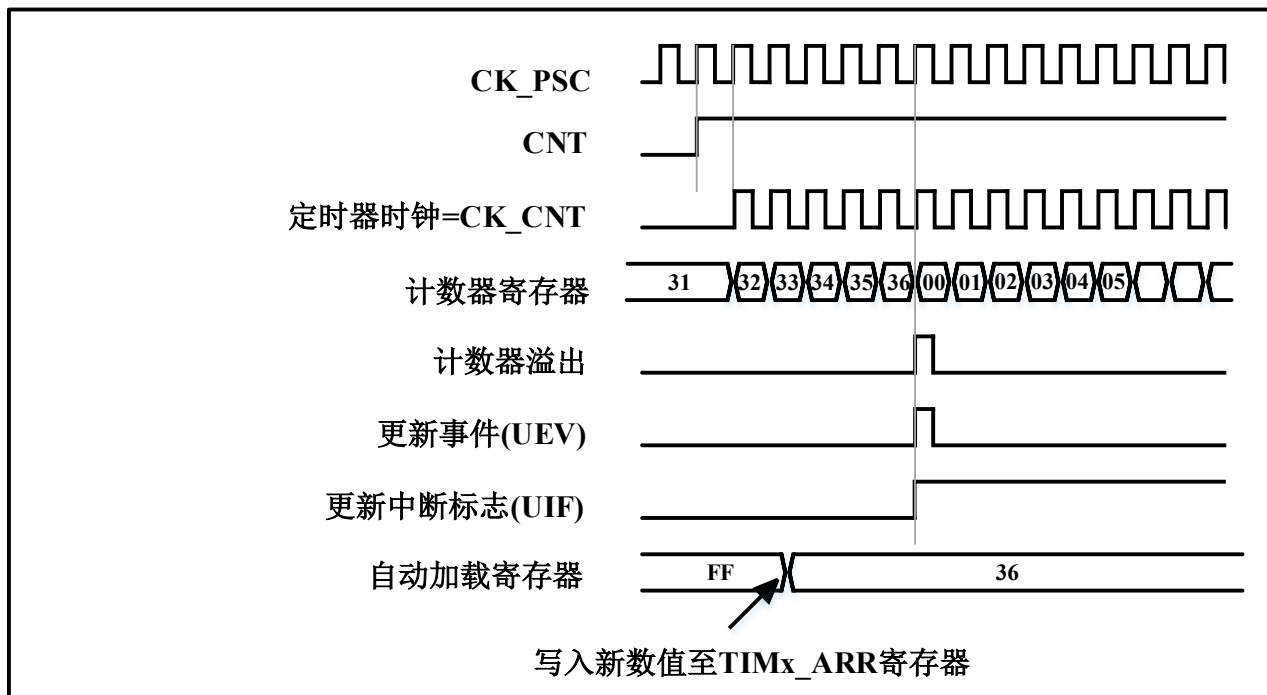


图 13-8. 计数器时序图，当 ARPE=0 时的更新事件(TIM1_ARR 没有预装入)

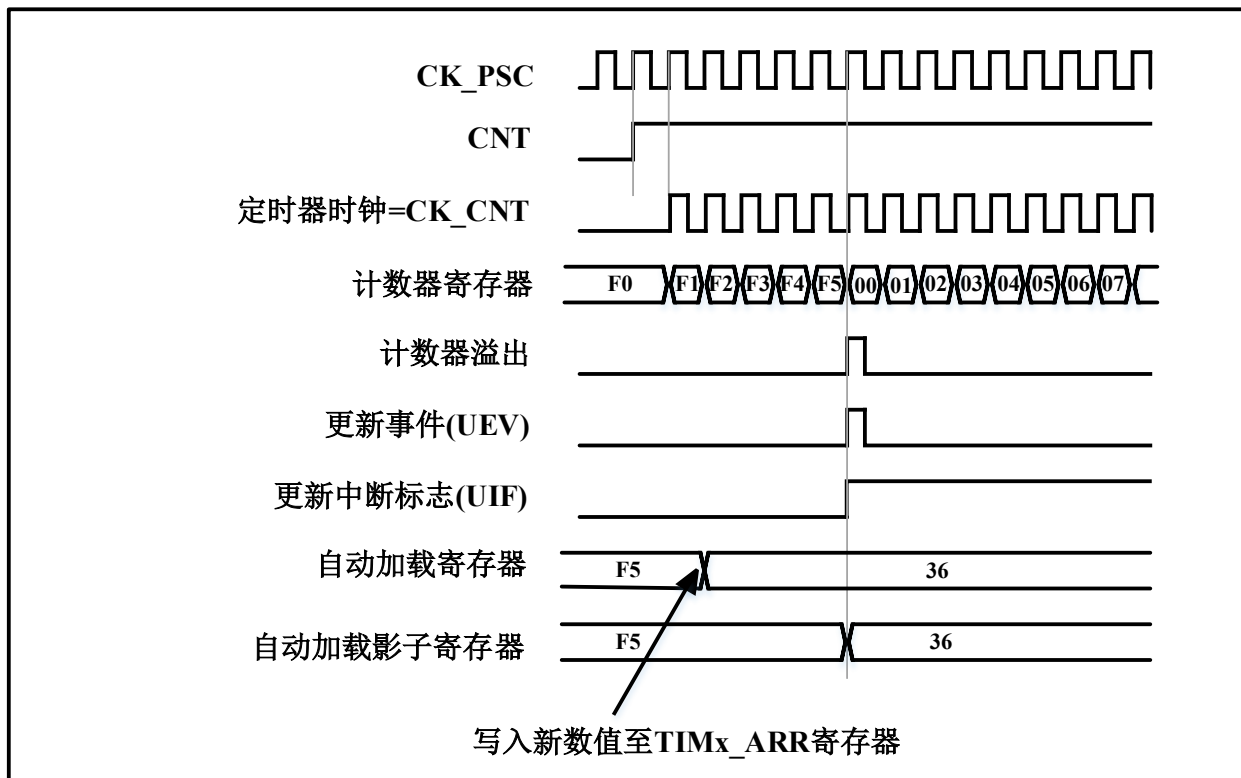


图 13-9. 计数器时序图，当 ARPE=1 时的更新事件(预装入了 TIM1_ARR)

13.3.2.2 向下计数模式

在向下模式中，计数器从自动装入的值(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)，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。

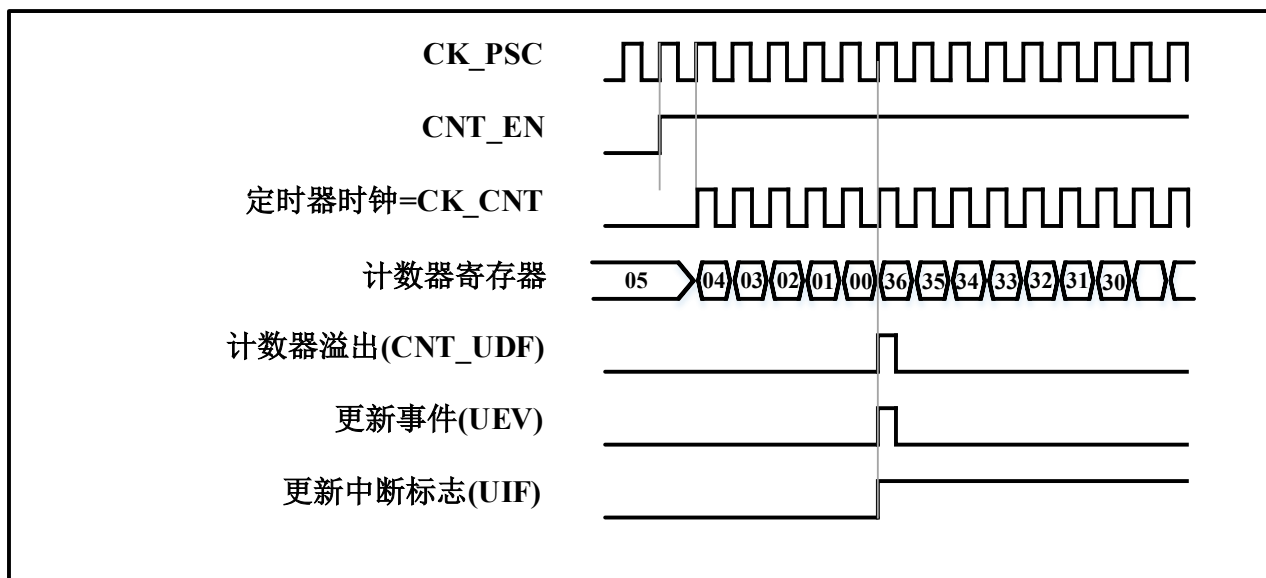


图 13-10. 计数器时序图，内部时钟分频因子为 1

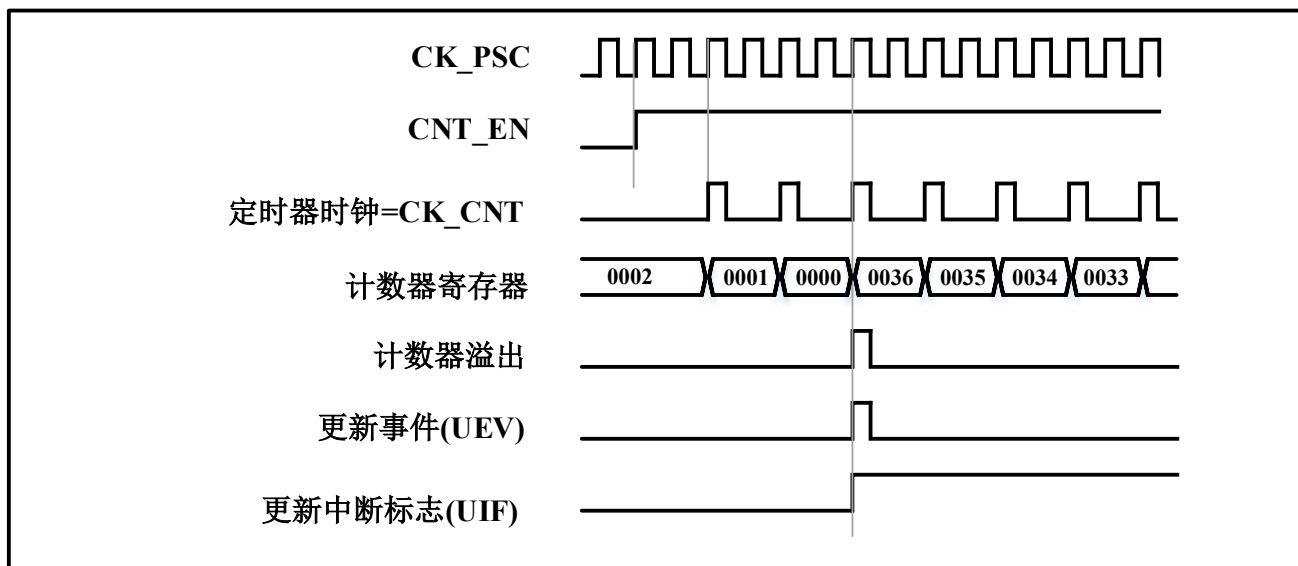


图 13-11. 计数器时序图，内部时钟分频因子为 2

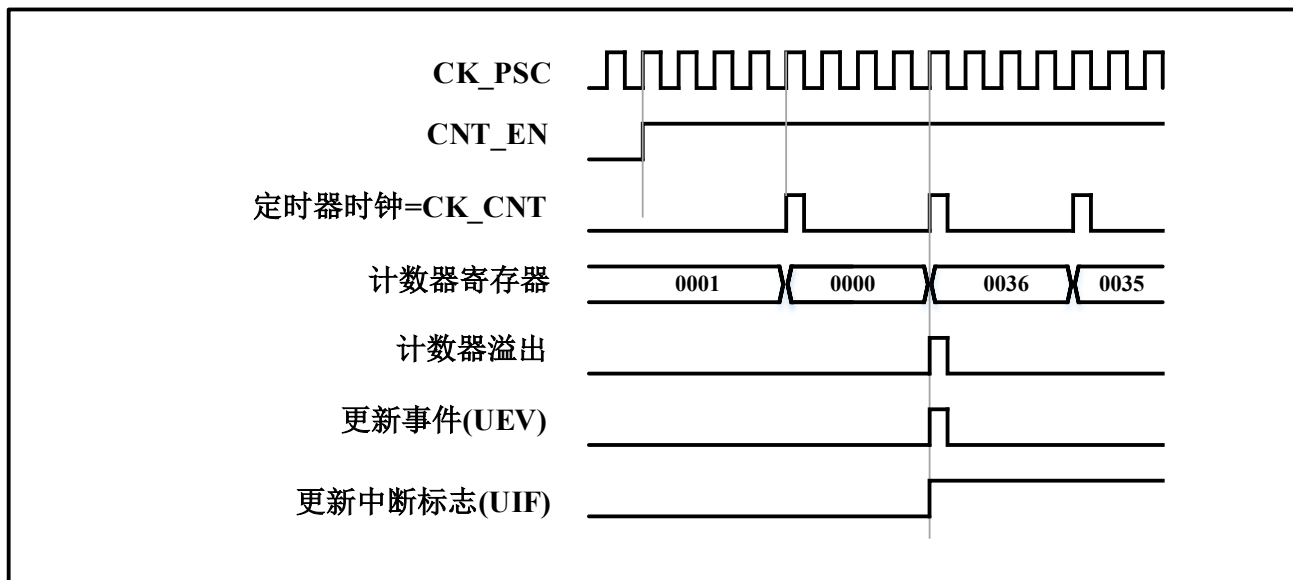


图 13-12. 计数器时序图，内部时钟分频因子为 4

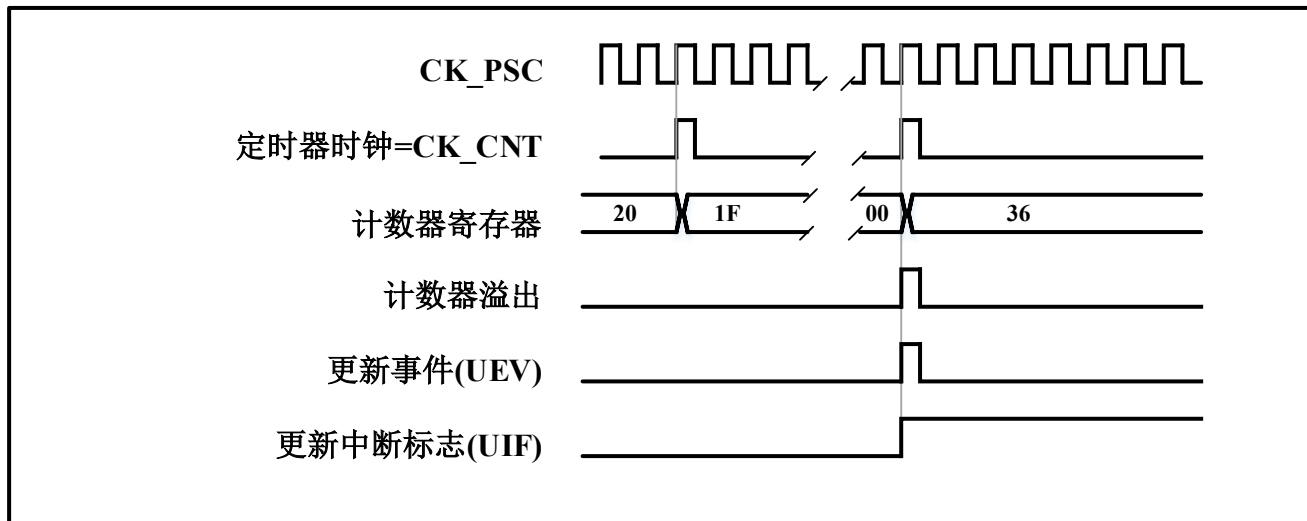


图 13-13. 计数器时序图，内部时钟分频因子为 N

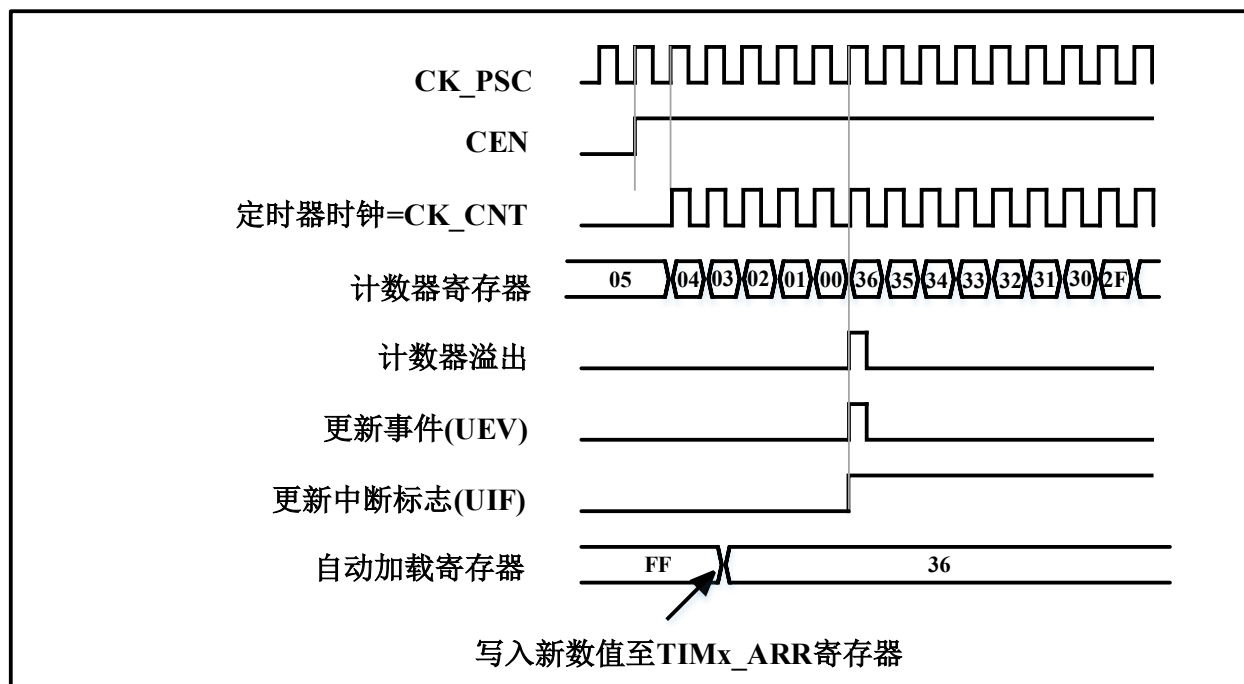


图 13-14. 计数器时序图，当没有使用重复计数器时的更新事件

13.3.2.3 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(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)，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：

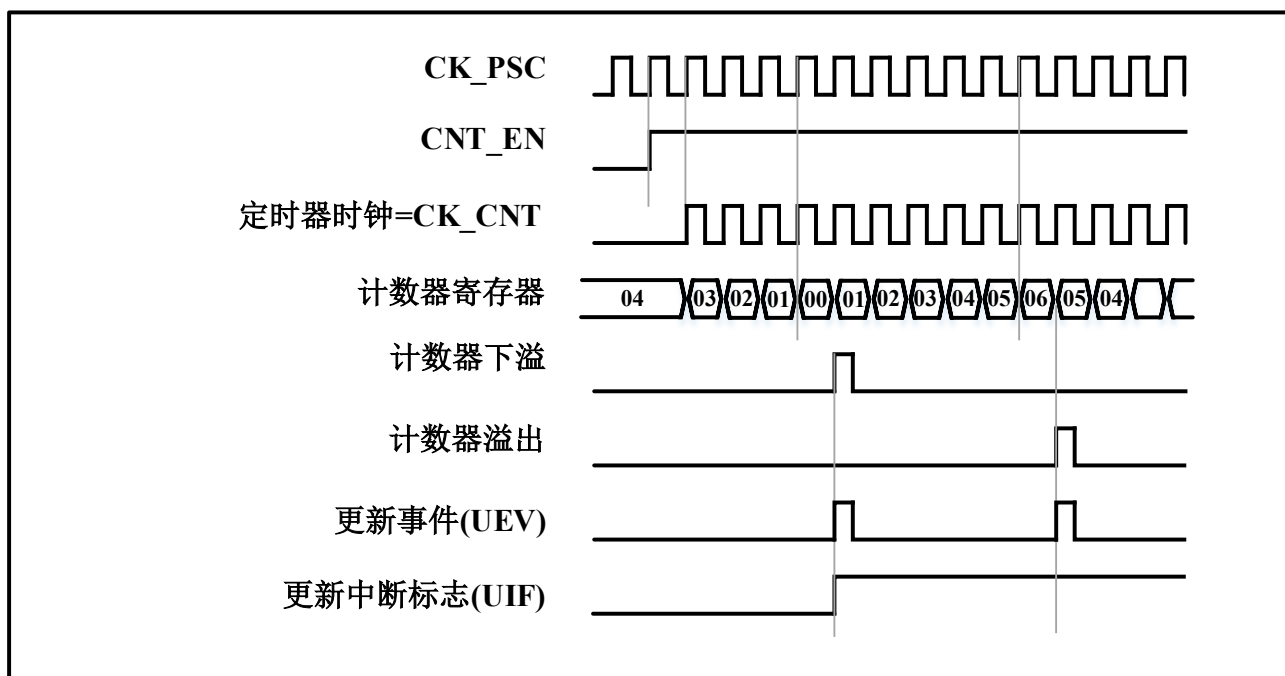


图 13-15. 计数器时序图，内部时钟分频因子为 1，TIM1_ARR=0x6

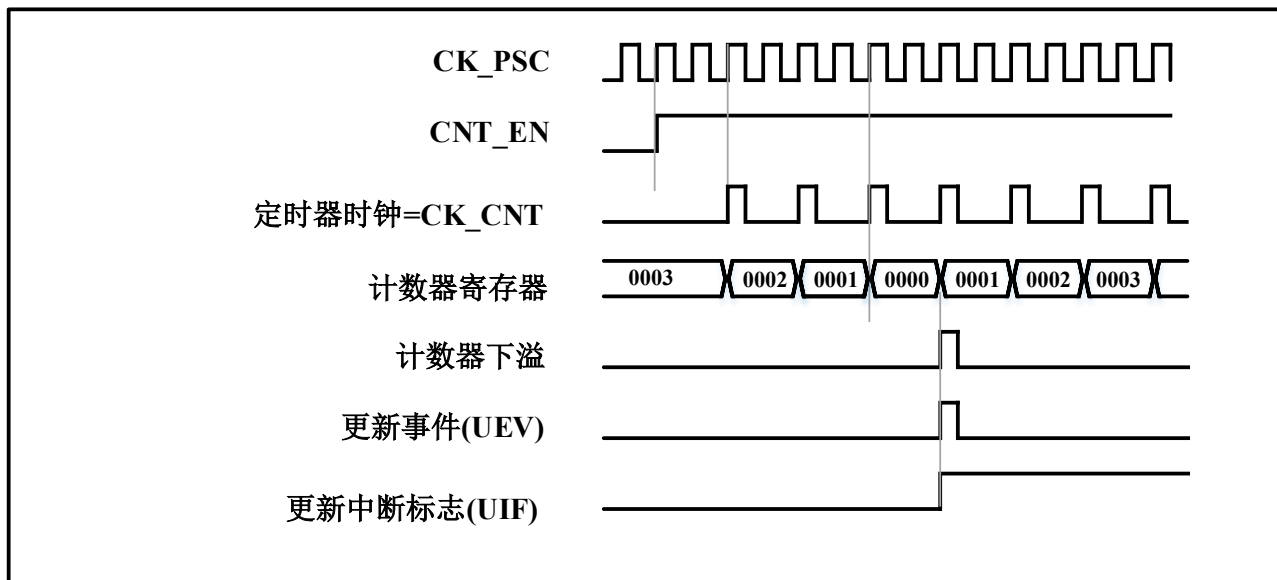


图 13-16. 计数器时序图，内部时钟分频因子为 2

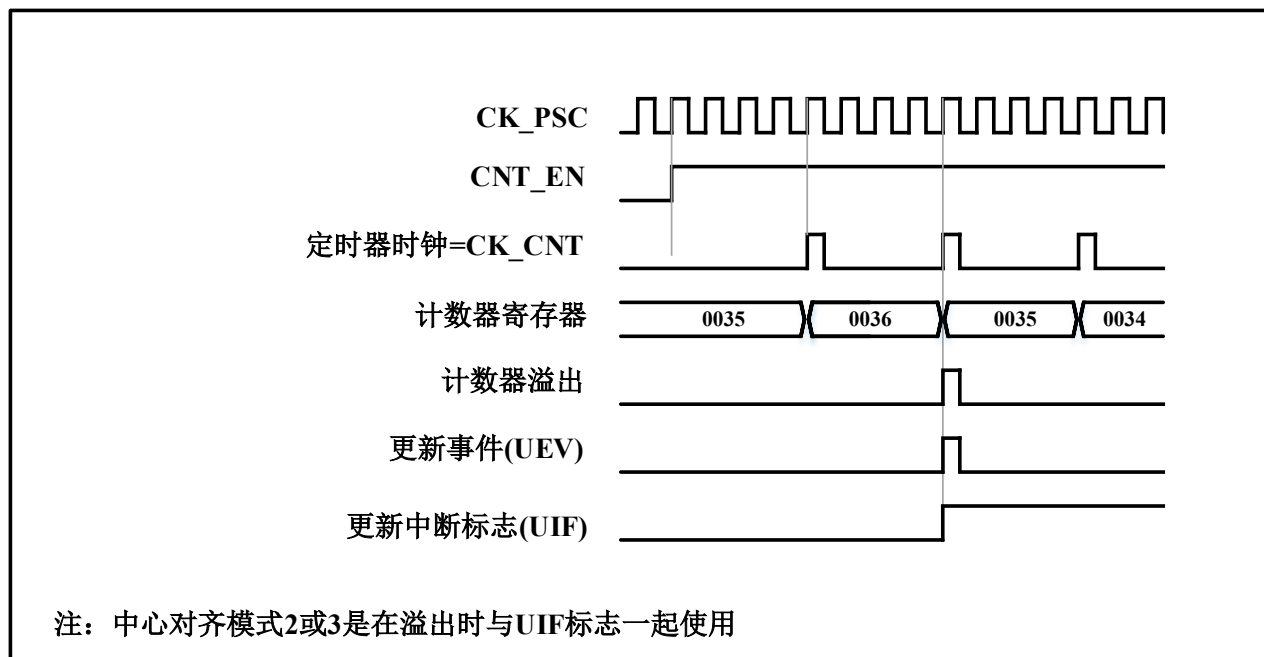


图 13-17. 计数器时序图，内部时钟分频因子为 4，TIM1_ARR=0x36

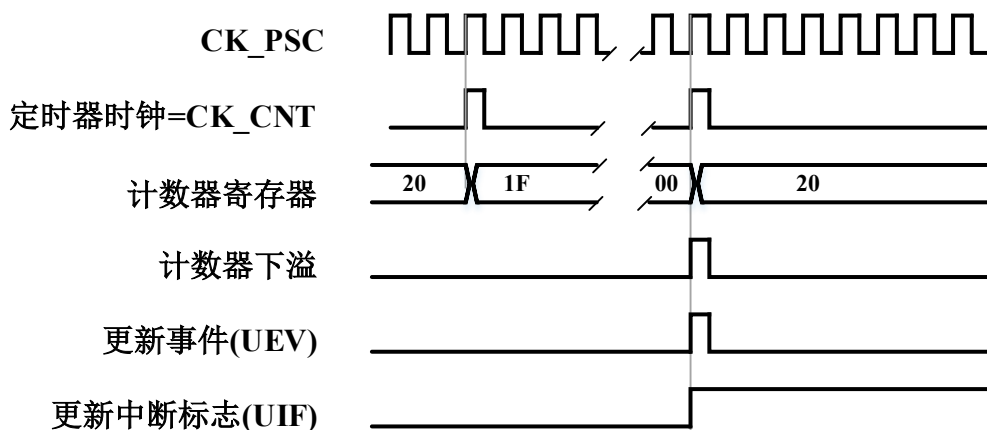


图 13-18. 计数器时序图，内部时钟分频因子为 N

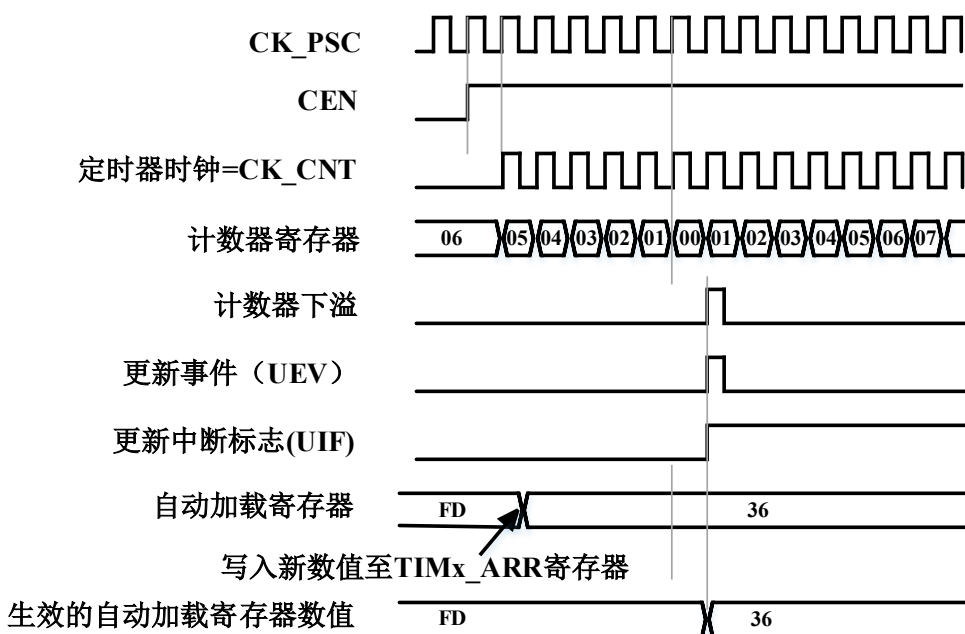


图 13-19. 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

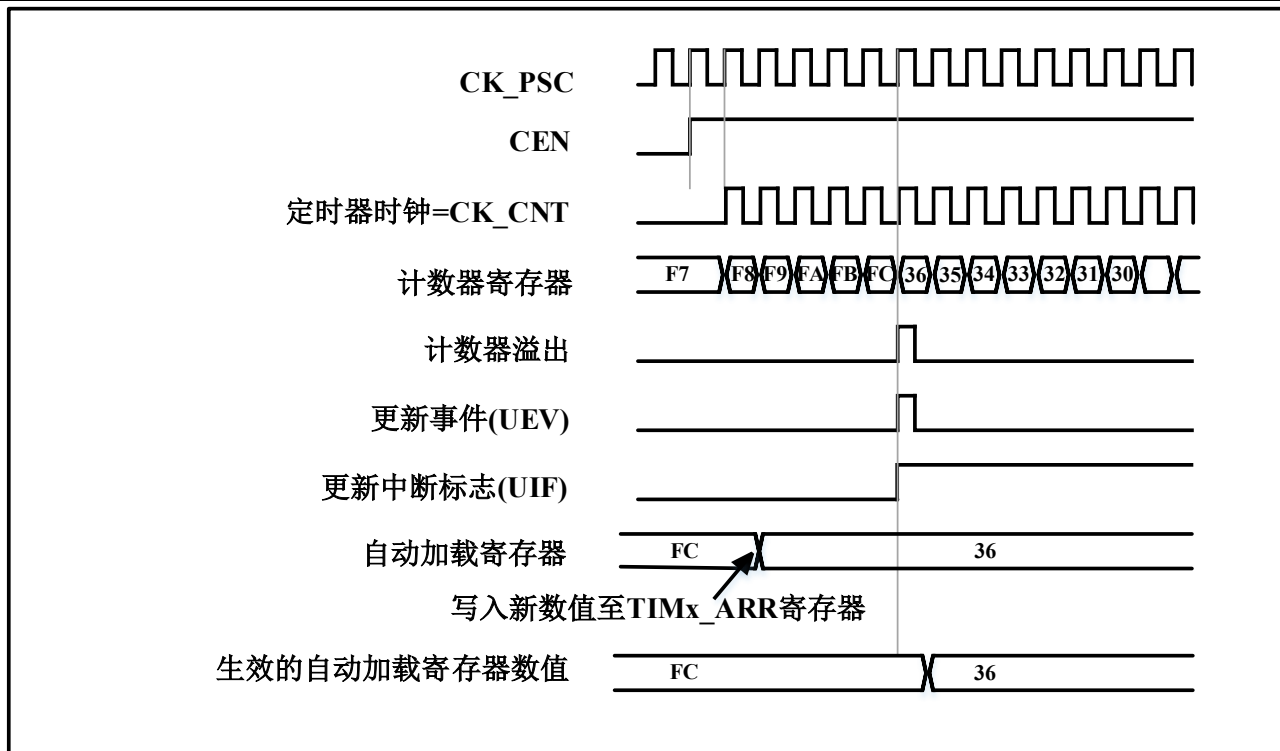


图 13-20. 计数器时序图，ARPE=1 时的更新事件(计数器溢出)

13.3.3 重复计数器

“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(参看图 13-21)。当

个定时器 TIM1 而作为另一个定时器 TIM2 的预分频器。详见下一章。

13.3.4.1 内部时钟源(CK_INT)

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。

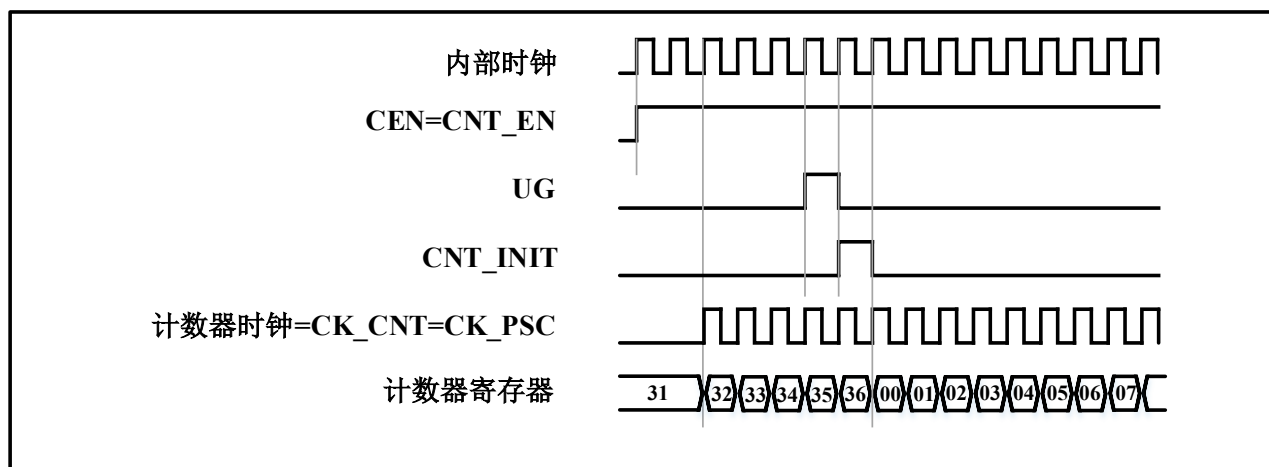


图 13-22. 一般模式下的控制电路，内部时钟分频因子为 1

13.3.4.2 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。

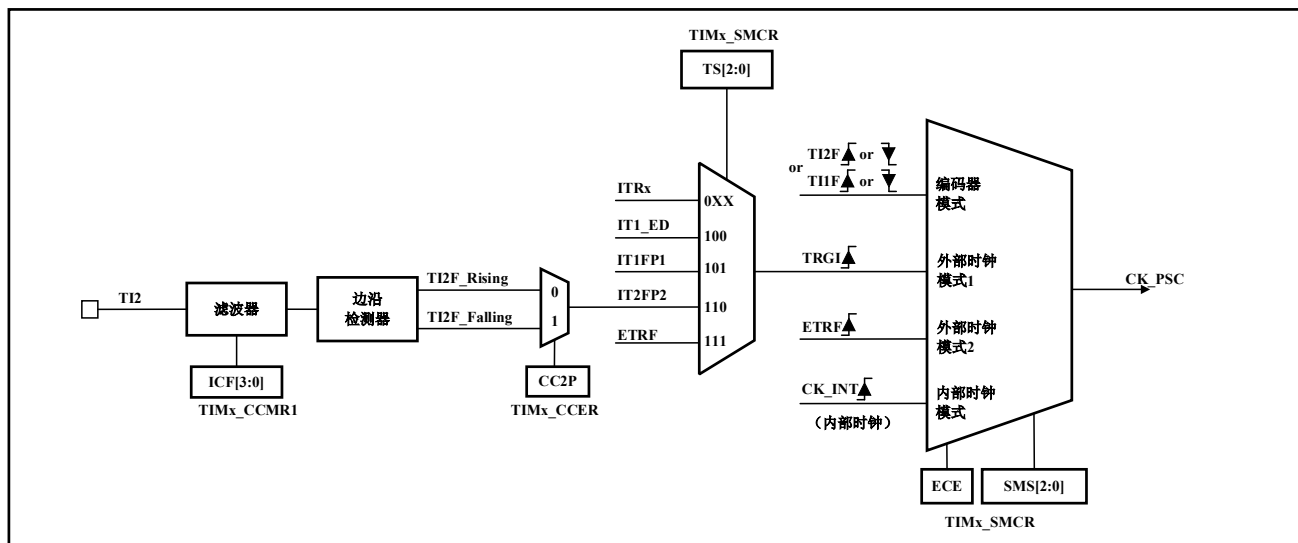


图 13-23. TI2 外部时钟连接例子

例如，要配置向上计数器在 TI2 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 TI2 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 TI2 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 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。

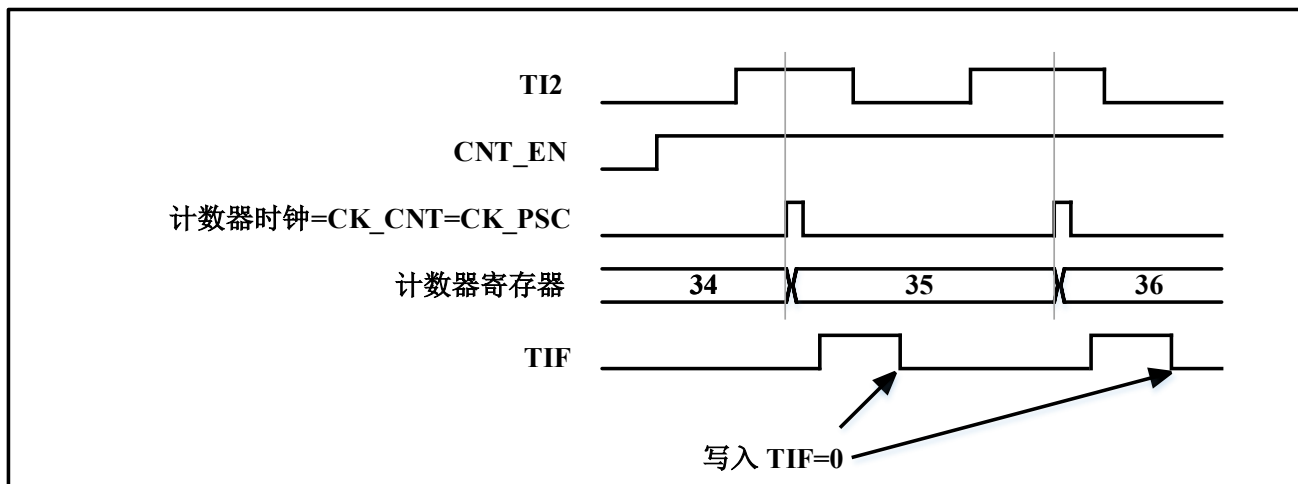


图 13-24. 外部时钟模式 1 下的控制电路

13.3.4.3 外部时钟源模式 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。

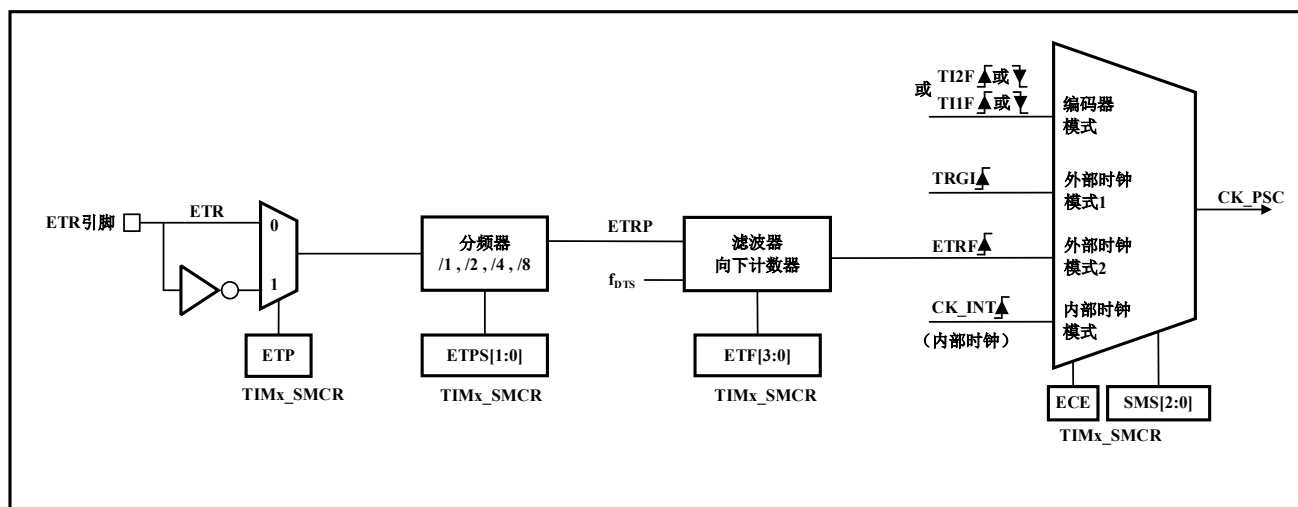


图 13-25. 外部触发输入框图

例如，要配置在 ETR 下每 2 个上升沿计数一次的向上计数器，使用下列步骤：本例中不需要滤波器，置 TIM1_SMCR 寄存器中的 ETF[3:0]=0000；设置预分频器，置 TIM1_SMCR 寄存器中的 ETPS[1:0]=01；选择 ETR 的上升沿检测，置 TIM1_SMCR 寄存器中的 ETP=0；开启外部时钟模式 2，写 TIM1_SMCR 寄存器中的 ECE=1；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。

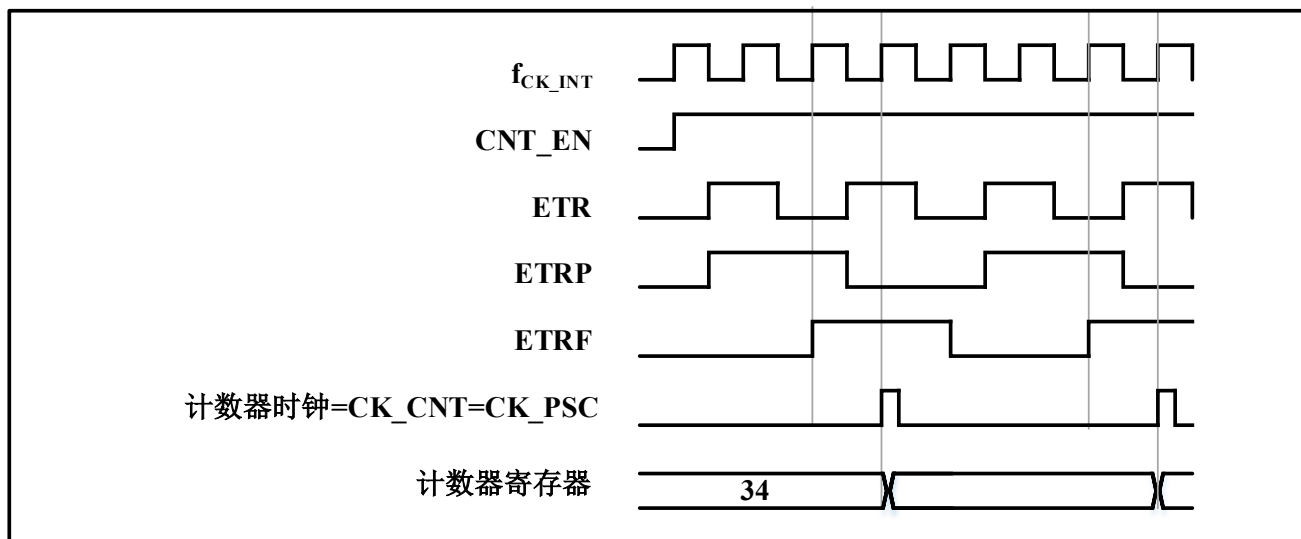


图 13-26. 外部时钟模式 2 下的控制电路

13.3.5 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)。

图 13-27 至图 13-30 是一个捕获/比较通道概览。

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

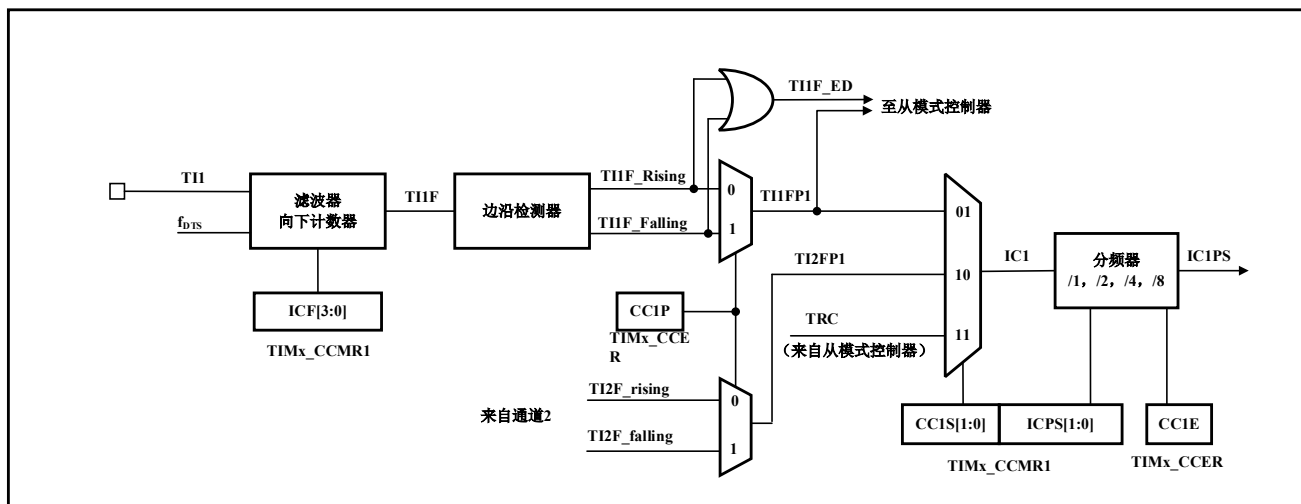


图 13-27. 捕获/比较通道(如: 通道 1 输入部分)

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

性。

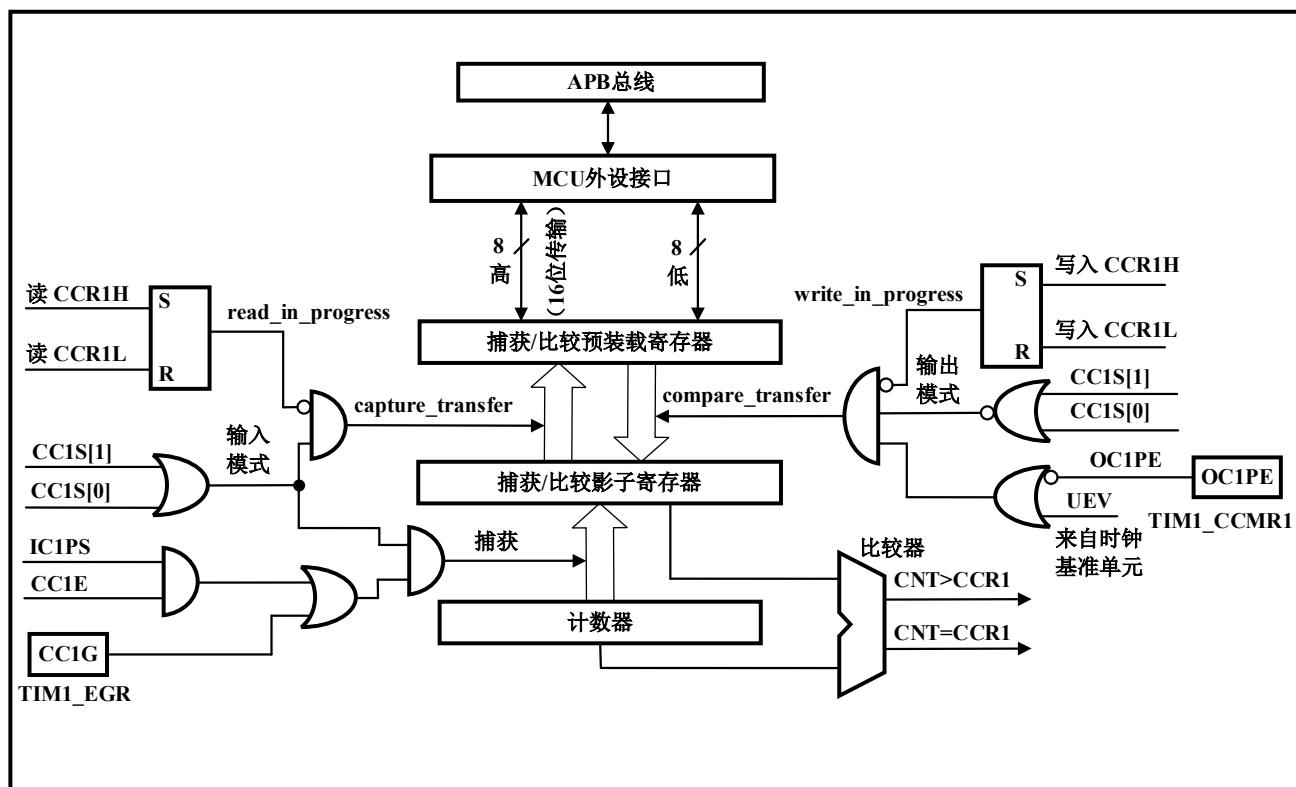


图 13-28. 捕获/比较通道 1 的主电路

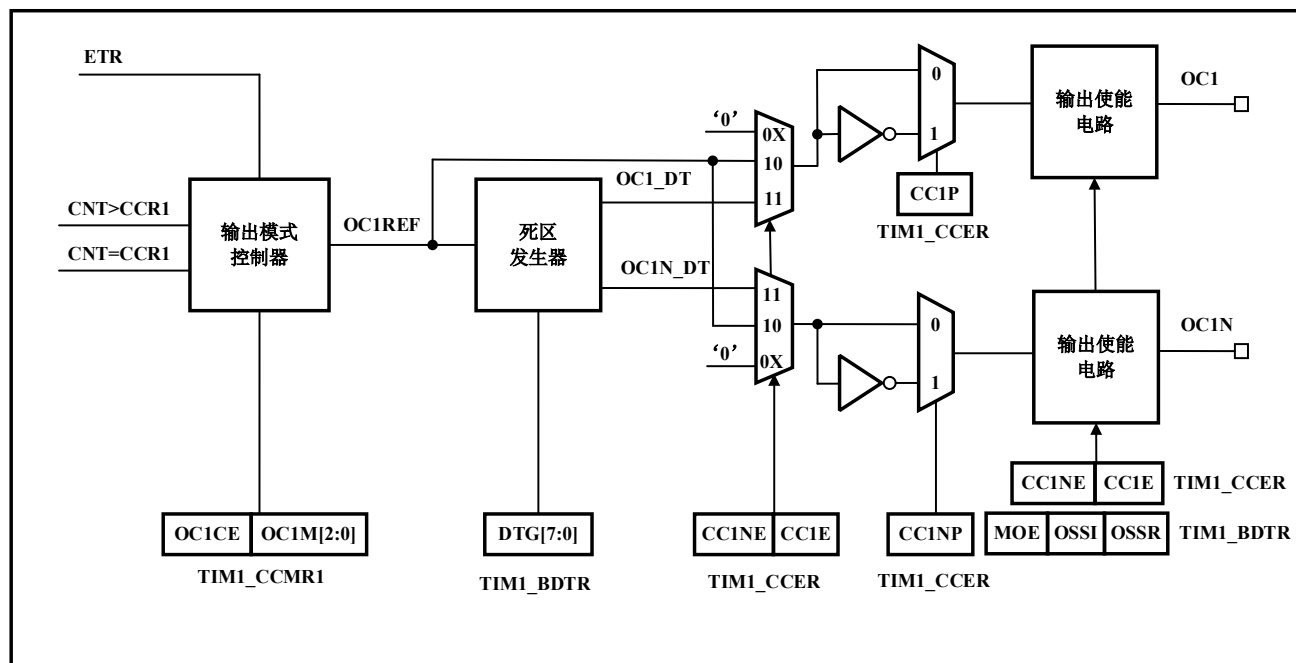


图 13-29. 捕获/比较通道的输出部分(通道 1 至 3)

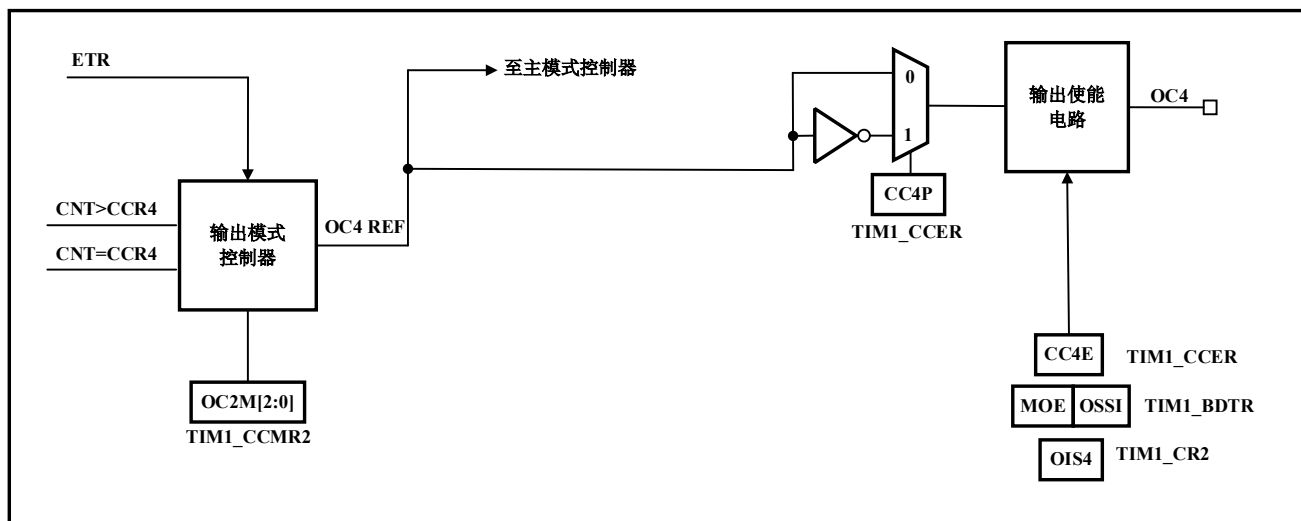


图 13-30. 捕获/比较通道的输出部分(通道 4)

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。

13.3.6 输入捕获模式

在输入捕获模式下，当检测到 IC_x 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_CCR_x)中。当发生捕获事件时，相应的 CC_xIF 标志(TIM1_SR 寄存器)被置 1，如果开放了中断，则将产生中断。如果发生捕获事件时 CC_xIF 标志已经为高，那么重复捕获标志 CC_xOF(TIM1_SR 寄存器)被置 1。写 CC_xIF=0 可清除 CC_xIF，或读取存储在 TIM1_CCR_x 寄存器中的捕获数据也可清除 CC_xIF。写 CC_xOF=0 可清除 CC_xOF。

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 TI_x 时，输入滤波器控制位是 TIM1_CCMR_x 寄存器中的 IC_xF 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。

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。

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。

13.3.7 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 TIx 输入。
- 这 2 个 ICx 信号为边沿有效，但是极性相反。
- 其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。例如，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。

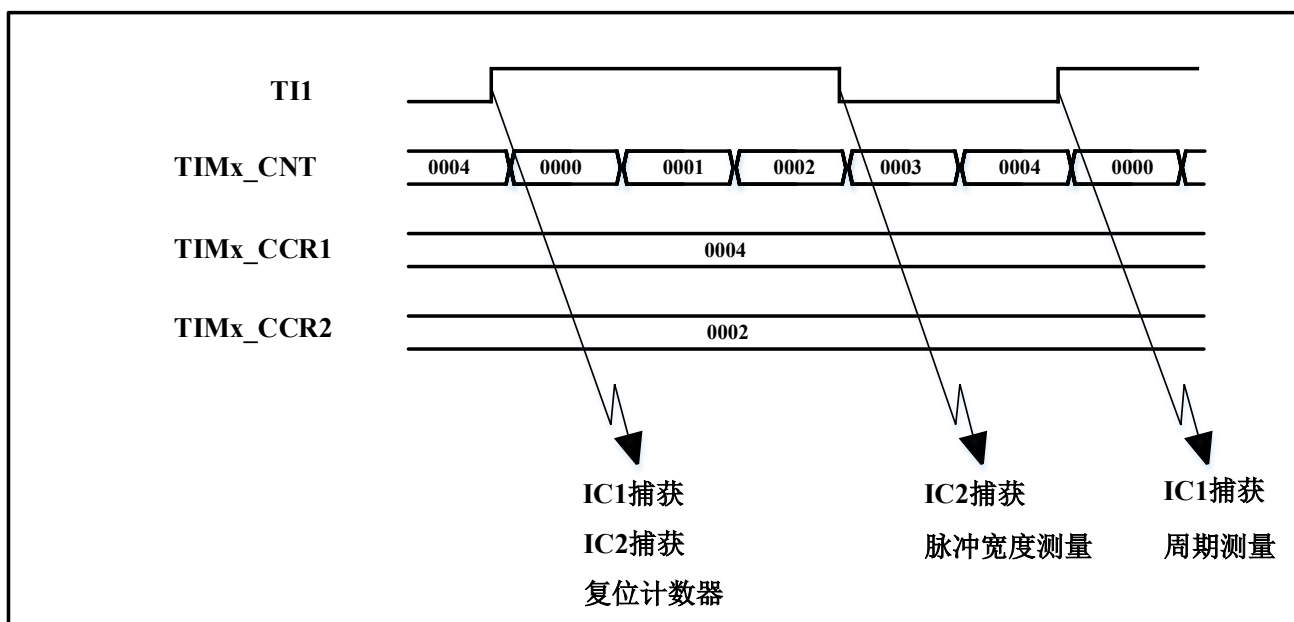


图 13-31. PWM 输入模式时序

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。

13.3.8 强制输出模式

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。这将会在下面的输出比较模式一节中介绍。

13.3.9 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。

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。

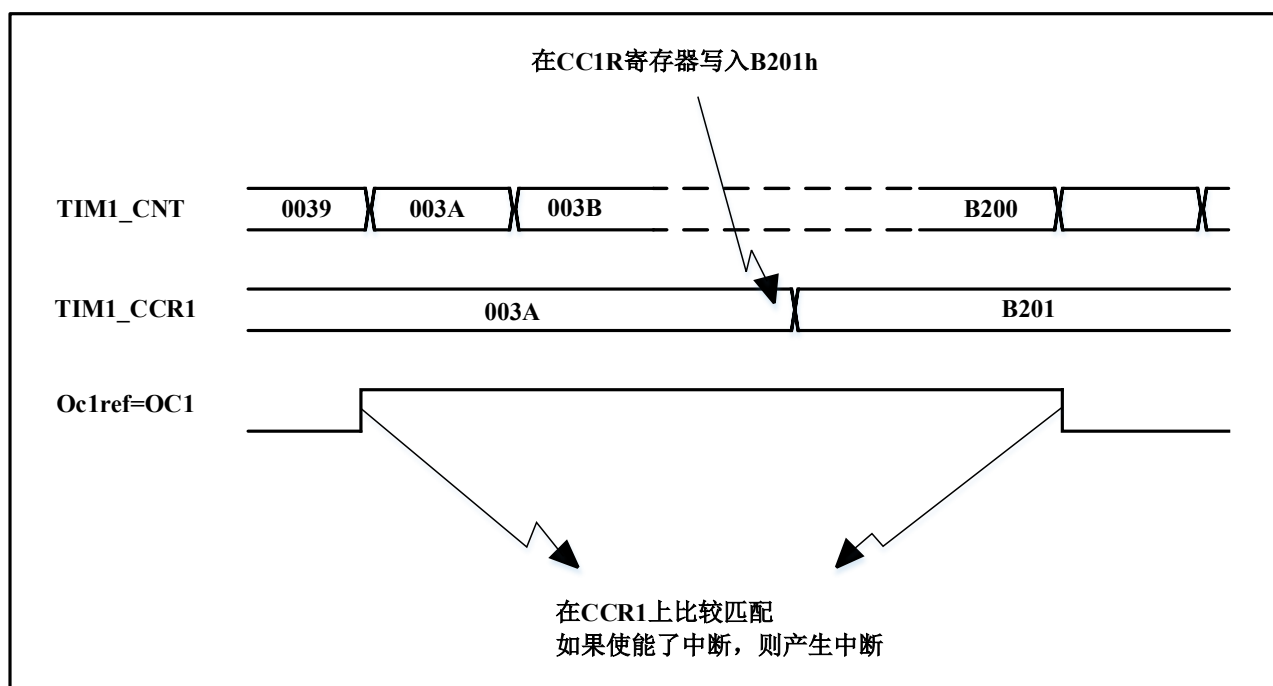


图 13-32. 输出比较模式，翻转 OC1

13.3.10 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。

13.3.10.1 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。

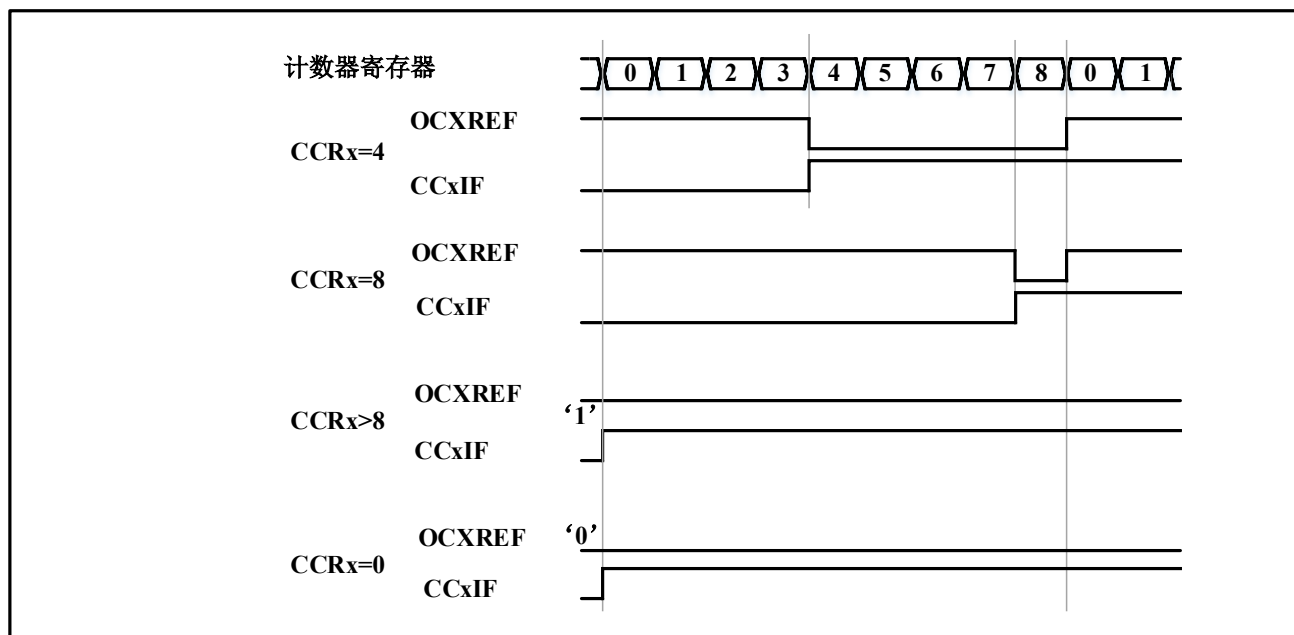


图 13-33. 边沿对齐的 PWM 波形(ARR=8)

向下计数配置

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。

13.3.10.2 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，不要用软件修改它。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。

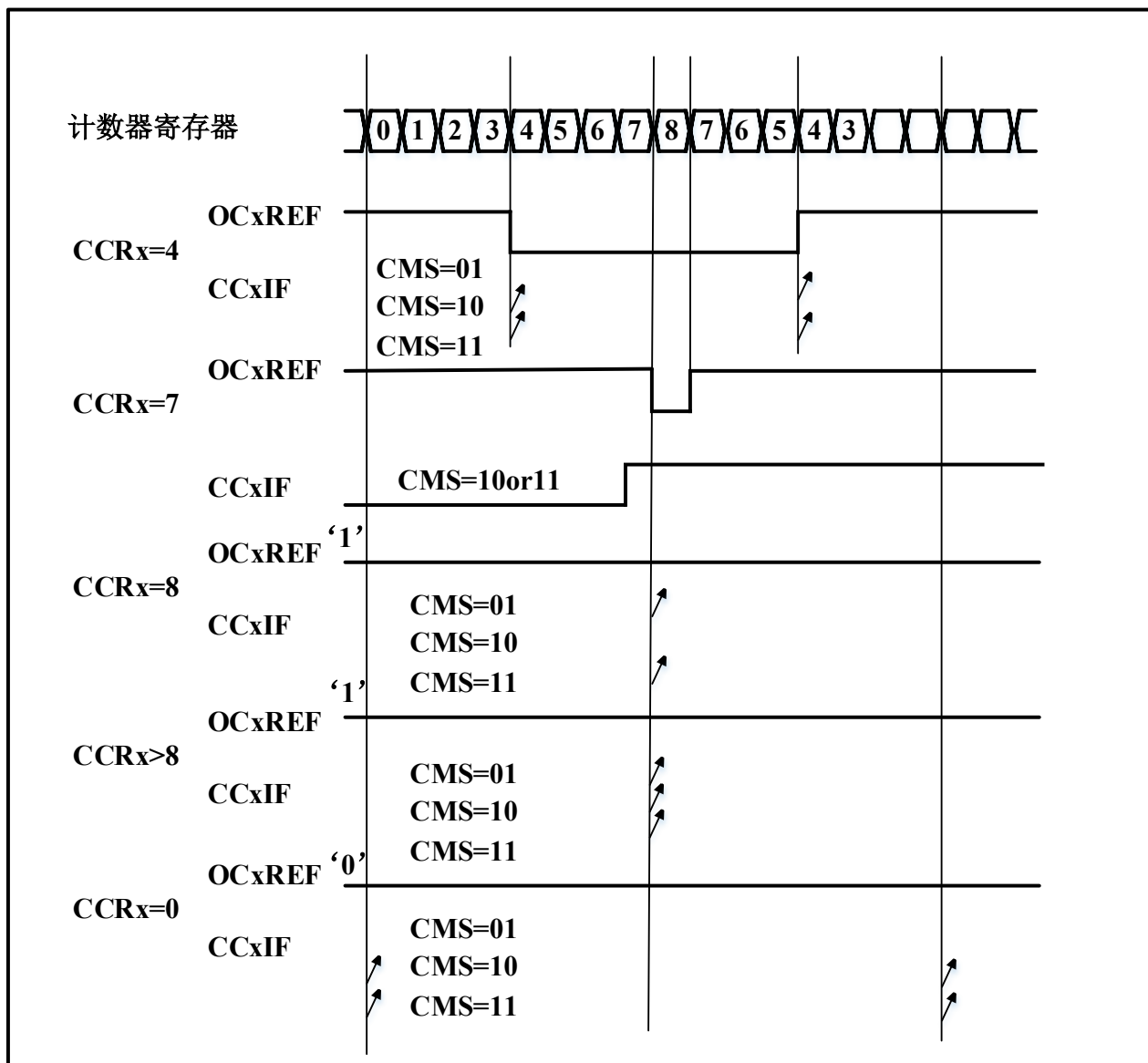


图 13-34. 中央对齐的 PWM 波形(APR=8)

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；这就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。

13.3.11 互补输出和死区插入

高级控制定时器(TIM1)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 13-4 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)

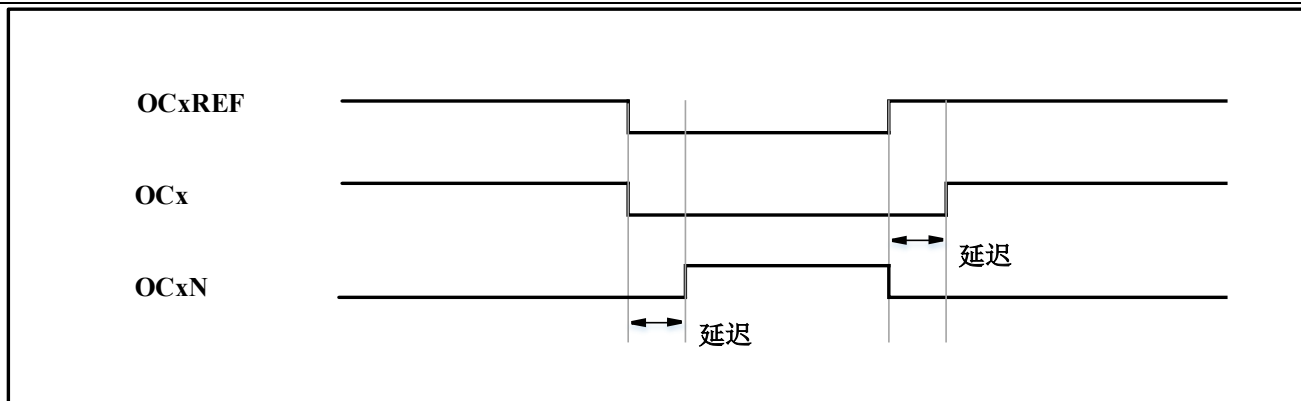


图 13-35. 带死区插入的互补输出

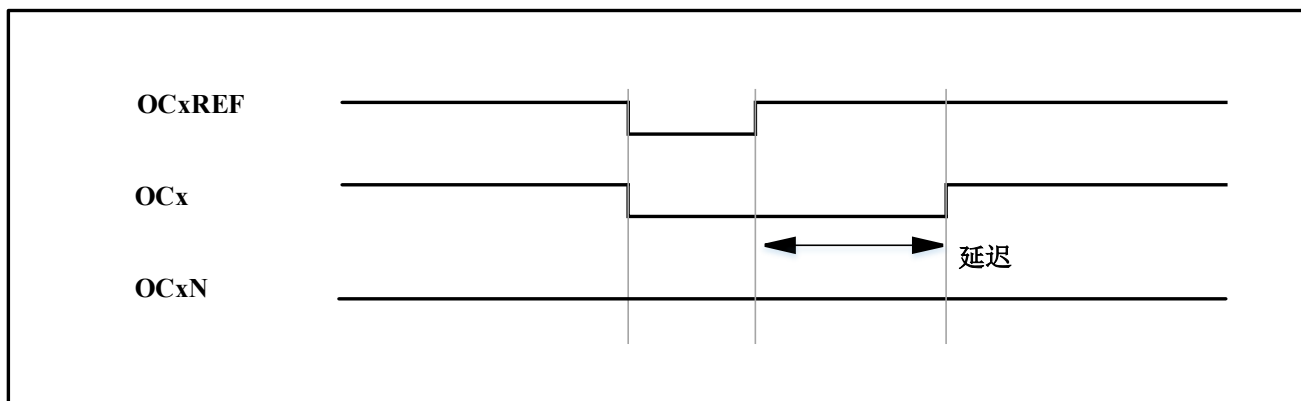


图 13-36. 死区波形延迟大于负脉冲

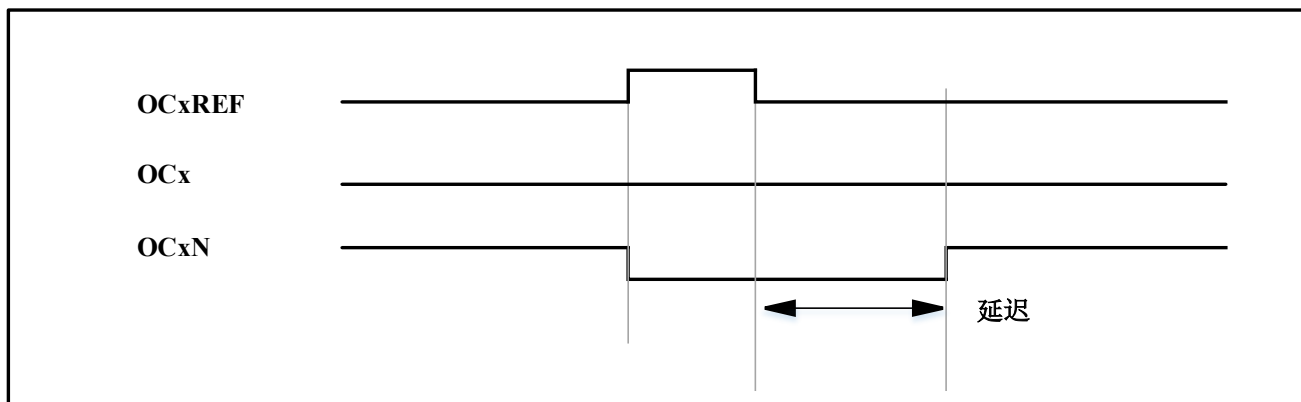


图 13-37. 死区波形延迟大于正脉冲

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。详见 TIM1 刹车和死区寄存器(TIM1_BDTR)中的延时计算。

13.3.11.1 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。

13.3.12 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。详见表 13-4 带刹车功能的互补输出通道 OCx 和 $OCxN$ 的控制位。

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 OSS1=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 OSS1=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_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 13.5.18 节 TIM1 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。

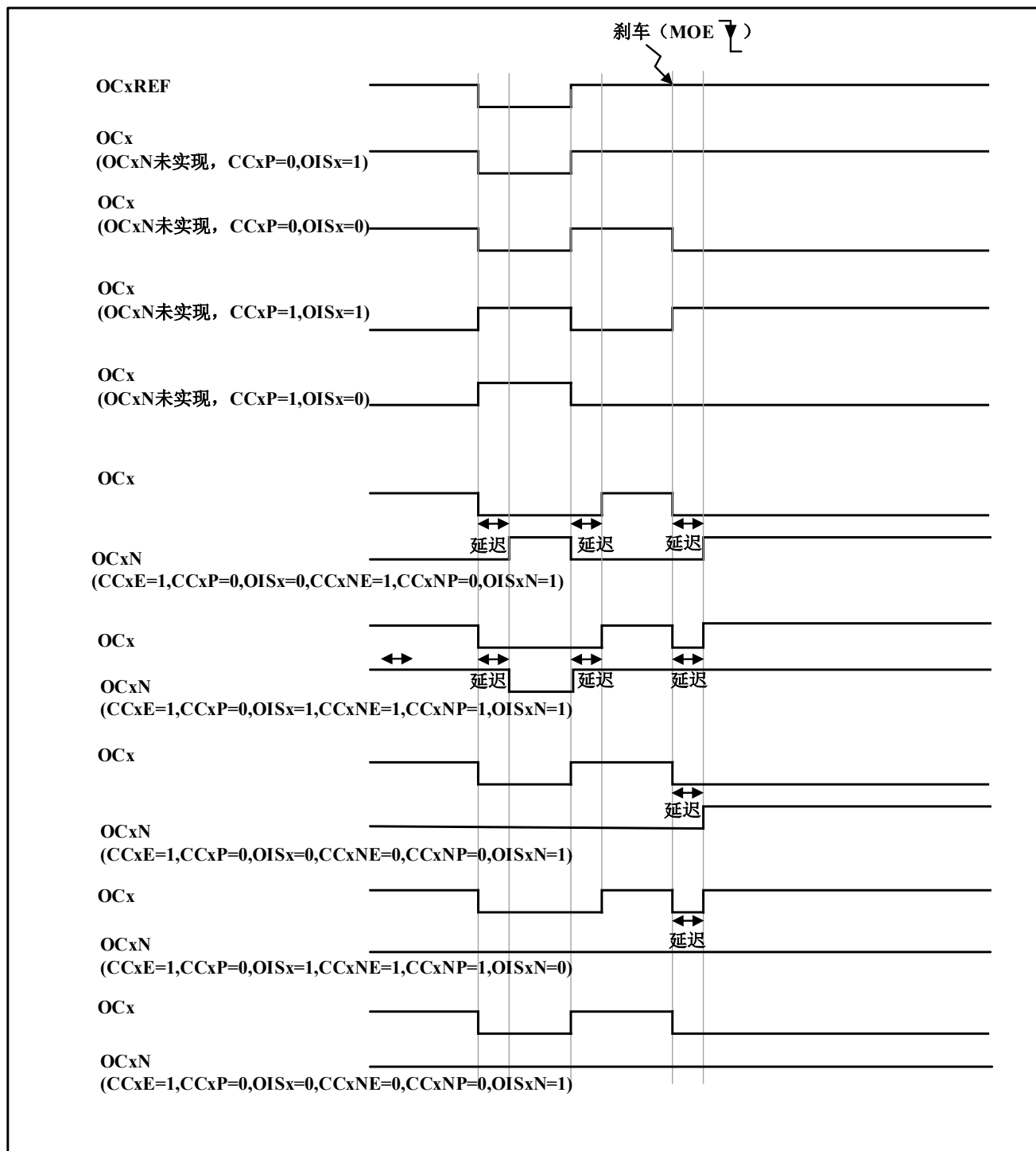


图 13-38. 响应刹车的输出

13.3.13 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。

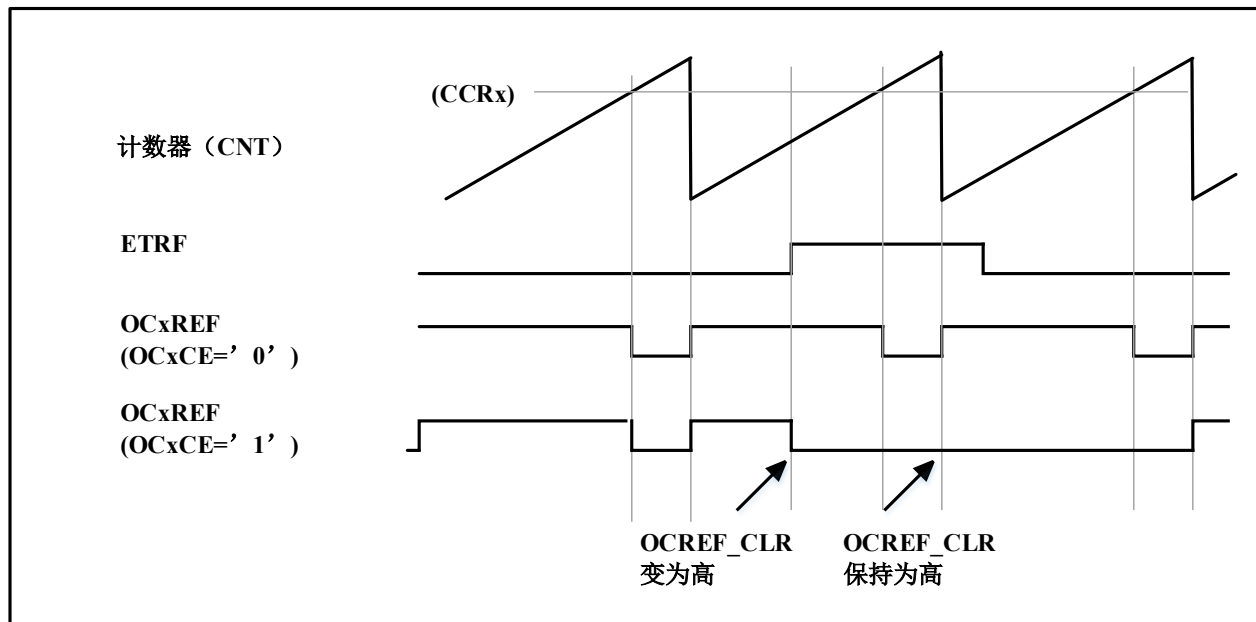


图 13-39. 清除 TIM1 的 OCxREF

13.3.14 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。

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

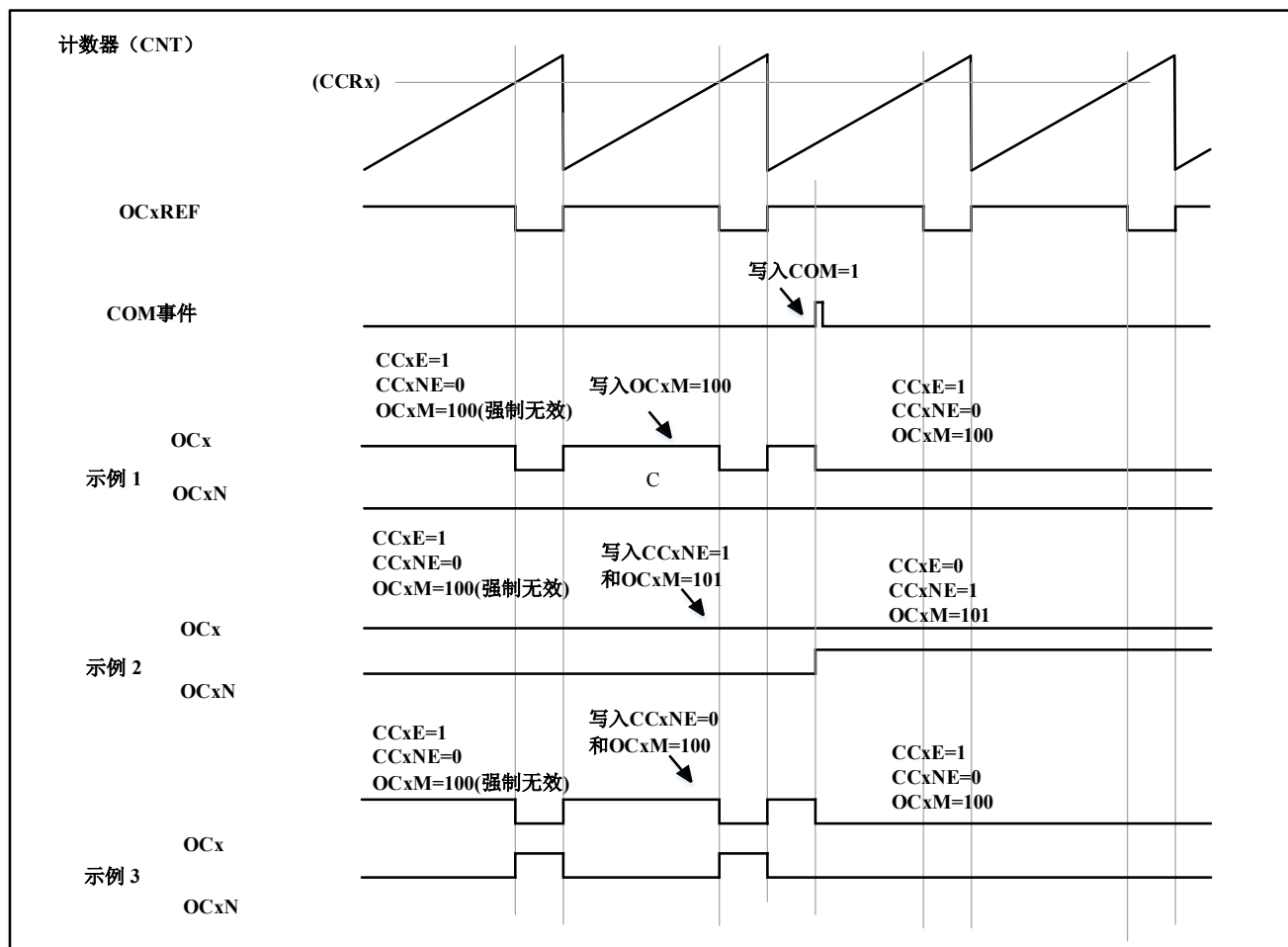


图 13-40. 产生六步 PWM，使用 COM 的例子(OSSR=1)

13.3.15 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 $CNT > CCR_x$ 。

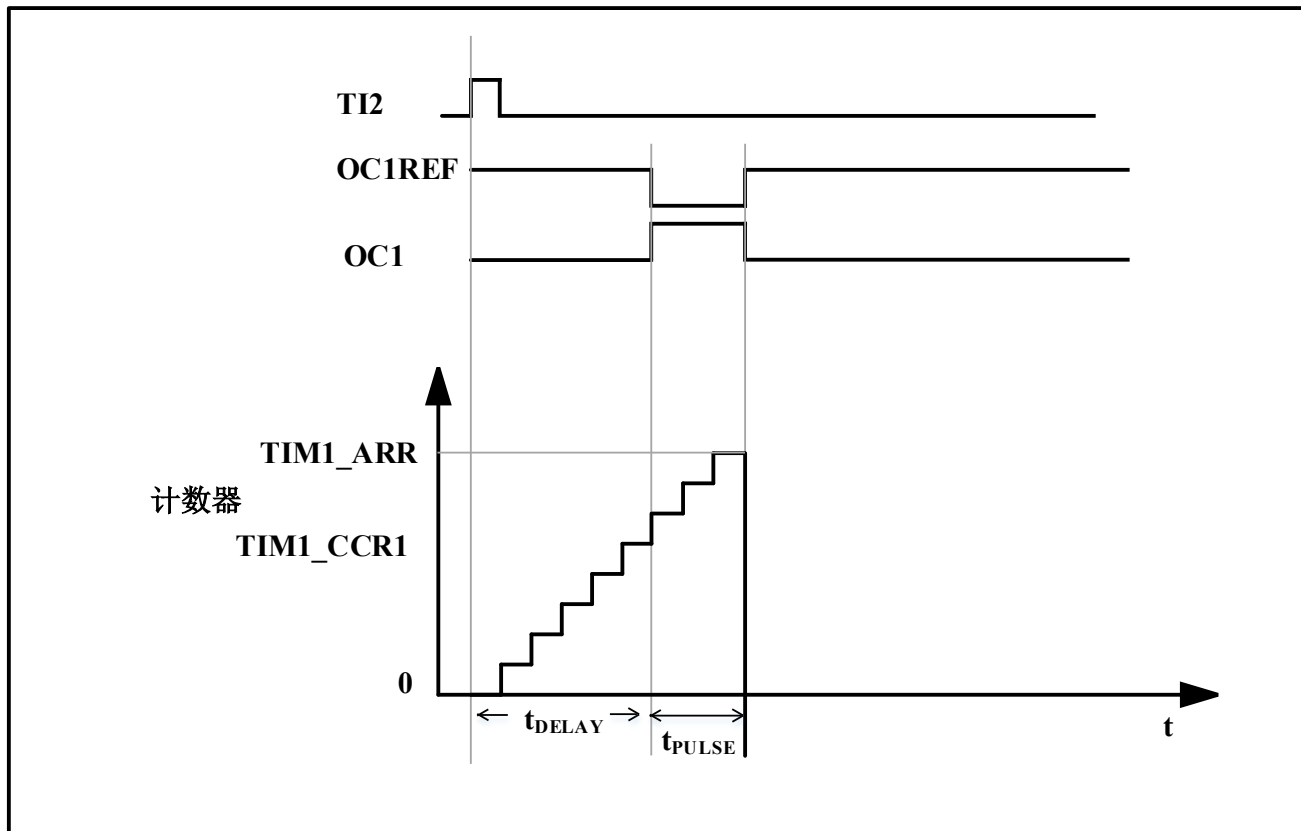


图 13-41. 单脉冲模式的例子

例如，你需要在从 TI2 输入脚上检测到一个上升沿开始，延迟 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 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)

- t_{DELAY} 由 TIM1_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(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；根据需要有选择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 t_{DELAY} 。

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。

13.3.16 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 13-1，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 13.1. 计数方向与编码器信号的关系

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

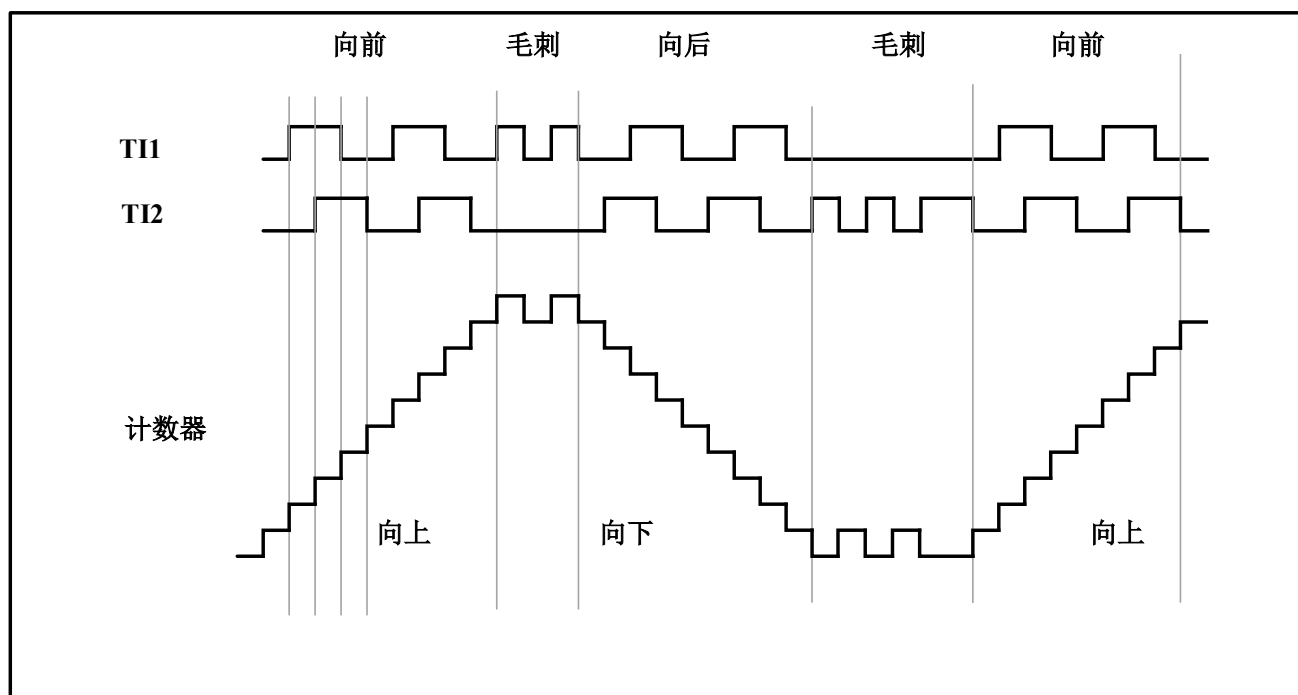


图 13-42. 编码器模式下的计数器操作实例

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

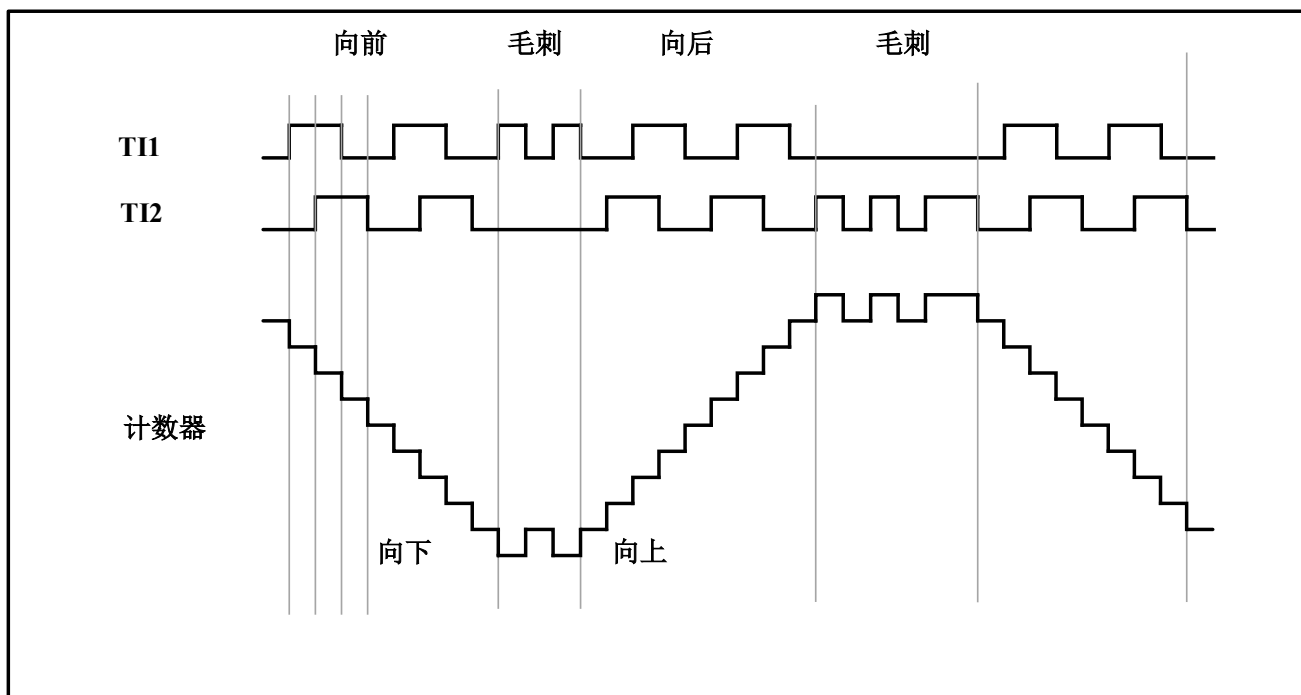


图 13-43. IC1FP1 反相的编码器接口模式实例

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)。

13.3.17 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。

13.3.18 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，见图 13-44，3 个定时器输入脚(CC1、CC2、CC3)通过一个异或门连接到 TI1 输入通道(通过设置 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(见图 13-27)。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:

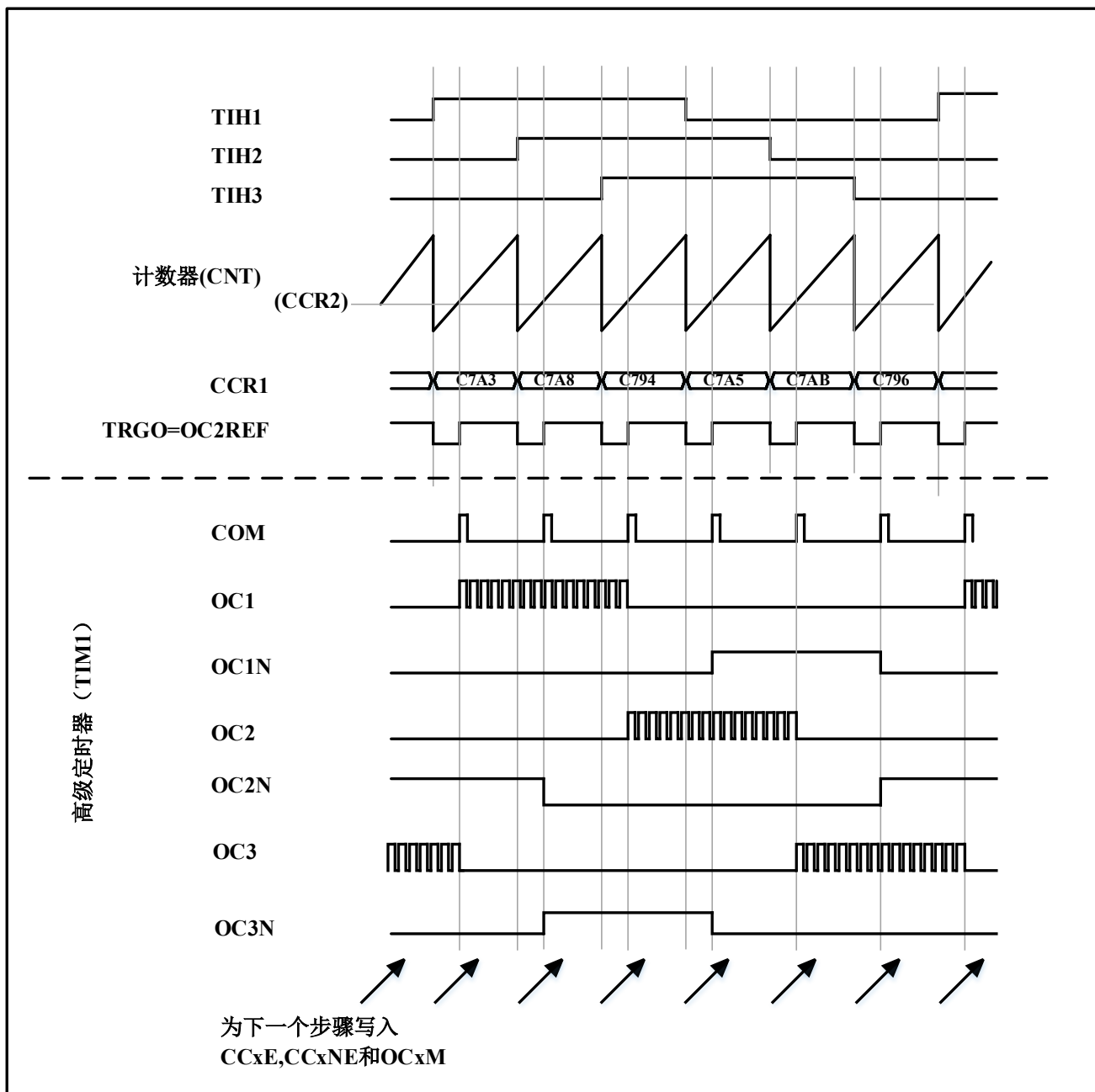


图 13-44. 霍尔传感器接口的实例

13.3.19 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。

13.3.19.1 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 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)位的设置，产生一个中断请求。

下图显示当自动重载寄存器 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。

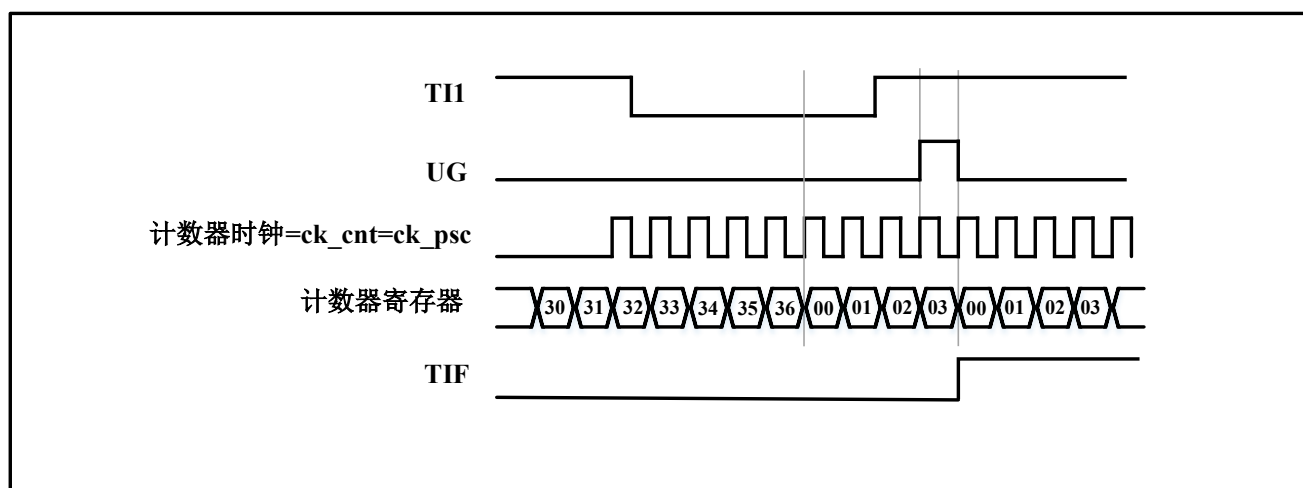


图 13-45. 复位模式下的控制电路

13.3.19.2 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 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。

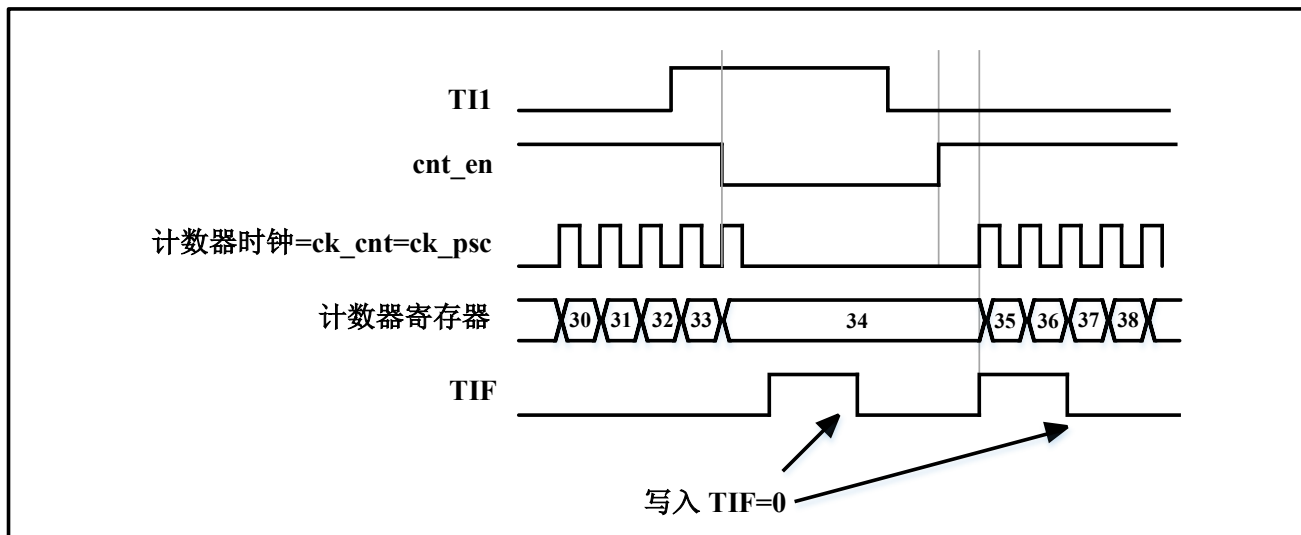


图 13-46. 门控模式下的控制电路

13.3.19.3 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 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。

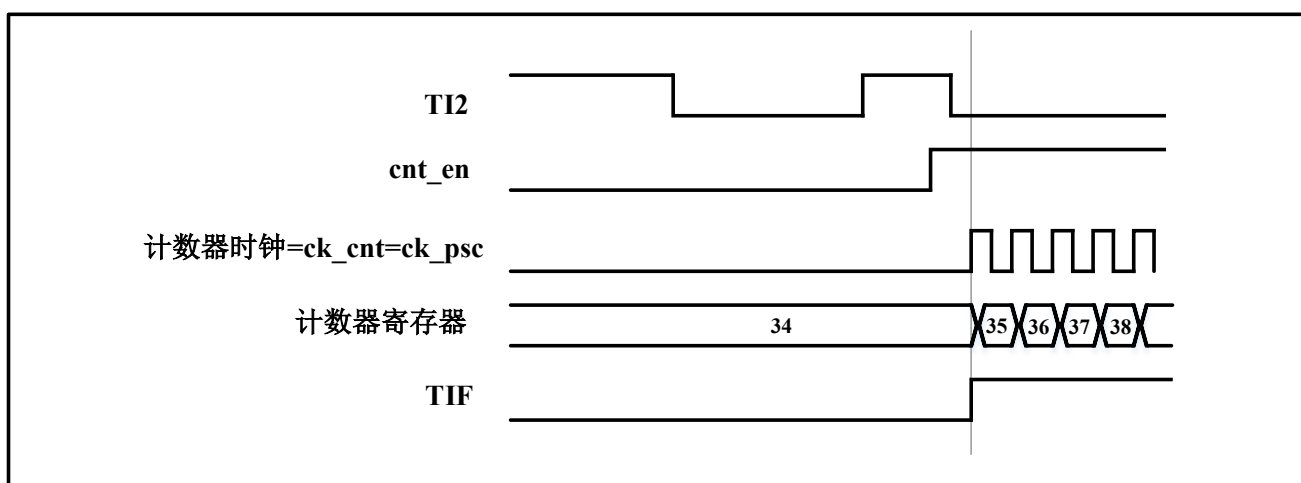


图 13-47. 触发器模式下的控制电路

13.3.19.4 从模式：外部时钟模式 2+触发模式

外部时钟模式 2 可以与另一种从模式(外部时钟模式 1 和编码器模式除外)一起使用。这时，ETR 信号被用作外部时钟的输入，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。

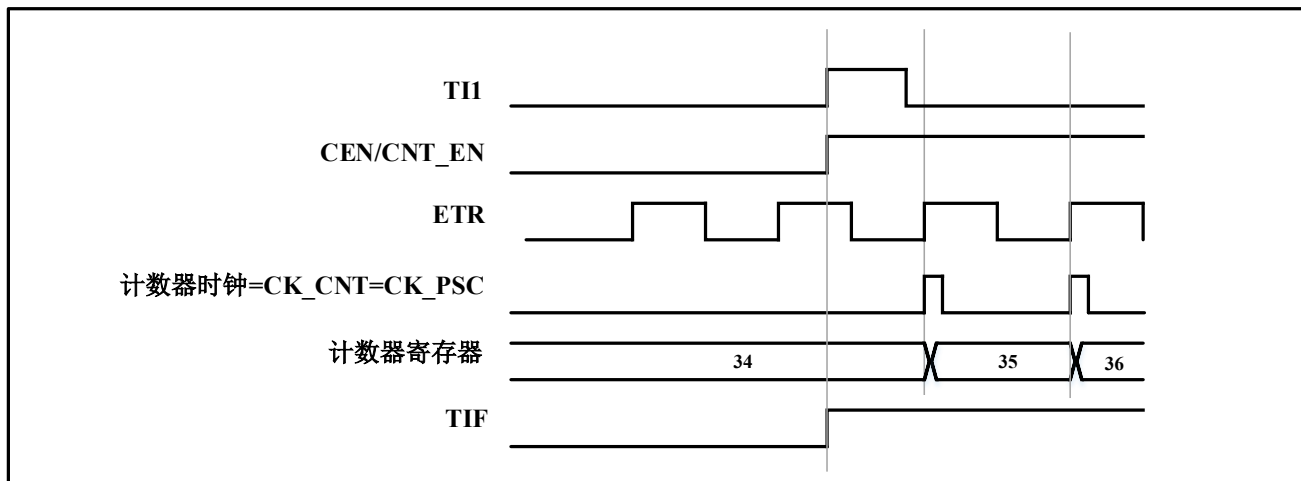


图 13-48. 外部时钟模式 2+触发模式下的控制电路

13.3.20 定时器同步

所有 TIM 定时器在内部相连, 用于定时器同步或链接。详见下一章 14.3.15 定时器同步。

13.3.21 调试模式

当微控制器进入调试模式时(Cortex-M0+核心停止), 根据 DBG 模块中 DBG_TIM1_STOP 的设置, TIM1 计数器可以或者继续正常操作, 或者停止。

13.4 TIM1 寄存器列表

可以用字(32 位)的方式操作这些外设寄存器。

TIM1 基地址：0X40001000

表 13.2. TIM1 寄存器列表和复位值

偏移地址	名称	描述	复位值
0x00	TIM1_CR1	TIM1 控制寄存器 1	0x00000000
0x04	TIM1_CR2	TIM1 控制寄存器 2	0x00000000
0x08	TIM1_SMCR	TIM1 从模式控制寄存器	0x00000000
0x0C	TIM1_DIER	TIM1 中断使能寄存器	0x00000000
0x10	TIM1_SR	TIM1 状态寄存器	0x00000000
0x14	TIM1_EGR	TIM1 事件产生寄存器	0x00000000
0x18	TIM1_CCMR1	TIM1 捕获/比较模式寄存器 1	0x00000000
0x1C	TIM1_CCMR2	TIM1 捕获/比较模式寄存器 2	0x00000000
0x20	TIM1_CCER	TIM1 捕获/比较使能寄存器	0x00000000
0x24	TIM1_CNT	TIM1 计数器	0x00000000
0x28	TIM1_PSC	TIM1 预分频器	0x00000000
0x2C	TIM1_ARR	TIM1 自动重装载寄存器	0x00000000
0x30	TIM1_RCR	TIM1 重复计数寄存器	0x00000000
0x34	TIM1_CCR1	TIM1 捕获/比较寄存器 1	0x00000000
0x38	TIM1_CCR2	TIM1 捕获/比较寄存器 2	0x00000000
0x3C	TIM1_CCR3	TIM1 捕获/比较寄存器 3	0x00000000
0x40	TIM1_CCR4	TIM1 捕获/比较寄存器 4	0x00000000
0x44	TIM1_BDTR	TIM1 刹车和死区寄存器	0x00000000

13.5 TIM1 寄存器说明

13.5.1 TIM1 控制寄存器 1(TIM1_CR1)

地址偏移：0x00

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKD[1:0]	ARPE	CMS[1:0]	DIR	OPM	URS	UDIS	CEN		
						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

位	标记	功能描述	复位值	读写
31:10	—	保留，始终读为 0。	0	—
9:8	CKD[1:0]	时钟分频因子(Clock division) 这 2 位定义在定时器时钟(CK_INT)频率、死区时间和由死区发生器与数字滤波器(ETR,TIx)所用的采样时钟之间的分频比例。 00: $t_{DTS}=t_{CK_INT}$ 01: $t_{DTS}=2\times t_{CK_INT}$ 10: $t_{DTS}=4\times t_{CK_INT}$ 11: 保留，不要使用这个配置	0x0	R/W
7	ARPE	自动重装载预装载允许位(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。	0	R/W
6:5	CMS[1:0]	选择中央对齐模式(Center-aligned mode selection) 00: 边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。 01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道(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	R/W
4	DIR	方向(Direction) 0: 计数器向上计数; 1: 计数器向下计数。 注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。	0	R/W
3	OPM	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。	0	R/W
2	URS	更新请求源(Update request source)软件通过该位选择 UEV 事件的源	0	R/W

位	标记	功能描述	复位值	读写
		0: 如果使能了更新中断, 则下述任一事件产生更新中断: -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, 则只有计数器溢出/下溢才产生更新中断。		
1	UDIS	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。(译注: 更新影子寄存器) 1: 禁止 UEV。不产生更新事件, 影子寄存器(ARR、PSC、CCR _x)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位, 则计数器和预分频器被重新初始化。	0	R/W
0	CEN	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。	0	R/W

13.5.2 TIM1 控制寄存器 2(TIM1_CR2)

地址偏移: 0x04

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[1:0]			保留	CCUS	保留	CCPC
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W				R/W		R/W

位	标记	功能描述	复位值	读写
31:15	—	保留, 始终读为 0。	0x0	R/W
14	OIS4	输出空闲状态 4(OC4 输出)。参见 OIS1 位。	0	R/W
13	OIS3N	输出空闲状态 3(OC3N 输出)。参见 OIS1N 位。	0	R/W
12	OIS3	输出空闲状态 3(OC3 输出)。参见 OIS1 位。	0	R/W

位	标记	功能描述	复位值	读写
11	OIS2N	输出空闲状态 2(OC2N 输出)。参见 OIS1N 位。	0	R/W
10	OIS2	输出空闲状态 2(OC2 输出)。参见 OIS1 位。	0	R/W
9	OIS1N	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。 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。	0	R/W
8	OIS1	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。	0	R/W
7	TI1S	TI1 选择(TI1 selection) 0: TIM1_CH1 引脚连到 TI1 输入; 1: TIM1_CH1、TIM1_CH2 和 TIM1_CH3 引脚经异或后连到 TI1 输入。	0	R/W
6:4	MMS[1:0]	主模式选择(Master mode selection) 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下: 000: 复位-TIM1_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式), 则 TRGO 上的信号相对实际的复位会有一个延迟。 001: 使能-计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。 有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟, 除非选择了主/从模式(见 TIM1_SMCR 寄存器中 MSM 位的描述)。 010: 更新-更新事件被选为触发输入(TRGO)。例如, 一个主定时器的时钟可以被用作一个从定时器的预分频器。 011: 比较脉冲-在发生一次捕获或一次比较成功时, 当要设置 CC1IF 标志时(即使它已经为高), 触发输出送出一个正脉冲(TRGO)。 100: 比较-OC1REF 信号被用于作为触发输出(TRGO)。 101: 比较-OC2REF 信号被用于作为触发输出(TRGO)。 110: 比较-OC3REF 信号被用于作为触发输出(TRGO)。 111: 比较-OC4REF 信号被用于作为触发输出(TRGO)。	0x0	R/W
3	—	保留, 始终读为 0。	0	—
2	CCUS	捕获/比较控制更新选择(Capture/compare control update selection) 0: 如果捕获/比较控制位是预装载的(CCPC=1), 只能通过设置 COM 位更新它们;	0	R/W

位	标记	功能描述	复位值	读写
		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。		
1	—	保留, 始终读为 0。	0	—
0	CCPC	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。	0	R/W

13.5.3 TIM1 从模式控制寄存器(TIM1_SMCR)

地址偏移: 0x08

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]	ETF[3:0]		MSM		TS[2:0]		保留		SMS[2:0]				
R/W	R/W	R/W	R/W		R/W		R/W		R/W		R/W				

位	标记	功能描述	复位值	读写
31:16	—	保留, 始终读为 0。	0	—
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。	0	R/W
14	ECE	外部时钟使能位(External clock enable)该位启用外部时钟模式 2 0: 禁止外部时钟模式 2; 1: 使能外部时钟模式 2。计数器由 ETRF 信号上的任意有效边沿驱动。 注 1: 设置 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	R/W
13:12	ETPS[1:0]	外部触发预分频(External trigger prescaler)	0	R/W

位	标记	功能描述	复位值	读写
		外部触发信号 ETRP 的频率必须多是 TIM1CLK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETRP 的频率。 00: 关闭预分频; 01: ETRP 频率除以 2; 10: ETRP 频率除以 4; 11: ETRP 频率除以 8。		
11:8	ETF[3:0]	外部触发滤波(External trigger filter) 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。 0000: 无滤波器，以 f_{DTS} 采样 0001: 采样频率 $f_{SAMPLING}=f_{CK_INT}$, $N=2$ 0010: 采样频率 $f_{SAMPLING}=f_{CK_INT}$, $N=4$ 0011: 采样频率 $f_{SAMPLING}=f_{CK_INT}$, $N=8$ 0100: 采样频率 $f_{SAMPLING}=f_{DTS}/2$, $N=6$ 0101: 采样频率 $f_{SAMPLING}=f_{DTS}/2$, $N=8$ 0110: 采样频率 $f_{SAMPLING}=f_{DTS}/4$, $N=6$ 0111: 采样频率 $f_{SAMPLING}=f_{DTS}/4$, $N=8$ 1000: 采样频率 $f_{SAMPLING}=f_{DTS}/8$, $N=6$ 1001: 采样频率 $f_{SAMPLING}=f_{DTS}/8$, $N=8$ 1010: 采样频率 $f_{SAMPLING}=f_{DTS}/16$, $N=5$ 1011: 采样频率 $f_{SAMPLING}=f_{DTS}/16$, $N=6$ 1100: 采样频率 $f_{SAMPLING}=f_{DTS}/16$, $N=8$ 1101: 采样频率 $f_{SAMPLING}=f_{DTS}/32$, $N=5$ 1110: 采样频率 $f_{SAMPLING}=f_{DTS}/32$, $N=6$ 1111: 采样频率 $f_{SAMPLING}=f_{DTS}/32$, $N=8$	0	R/W
7	MSM	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。	0	R/W
6:4	TS[2:0]	触发选择(Trigger selection) 这 3 位选择用于同步计数器的触发输入。 000: 内部触发 0(ITR0) 001: 内部触发 1(ITR1) 010: 内部触发 2(ITR2) 011: 内部触发 3(ITR3) 100: TI1 的边沿检测器(TI1F_ED) 101: 滤波后的定时器输入 1(TI1FP1) 110: 滤波后的定时器输入 2(TI2FP2) 111: 外部触发输入(ETRF) 更多有关 ITRx 的细节，参见表 13.3。	0	R/W

位	标记	功能描述	复位值	读写
		注：这些位只能在未用到(如 SMS=000)时被改变，以避免在改变时产生错误的边沿检测。		
3	—	保留，始终读为 0。	0	—
2:0	SMS[2:0]	<p>从模式选择(Slave mode selection)</p> <p>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)</p> <p>000: 关闭从模式—如果 CEN=1，则预分频器直接由内部时钟驱动。</p> <p>001: 编码器模式 1—根据 TI1FP1 的电平，计数器在 TI2FP2 的边沿向上/下计数。</p> <p>010: 编码器模式 2—根据 TI2FP2 的电平，计数器在 TI1FP1 的边沿向上/下计数。</p> <p>011: 编码器模式 3—根据另一个信号的输入电平，计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数。</p> <p>100: 复位模式—选中的触发输入(TRGI)的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。</p> <p>101: 门控模式—当触发输入(TRGI)为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止(但不复位)。计数器的启动和停止都是受控的。</p> <p>110: 触发模式—计数器在触发输入 TRGI 的上升沿启动(但不复位)，只有计数器的启动是受控的。</p> <p>111: 外部时钟模式 1—选中的触发输入(TRGI)的上升沿驱动计数器。</p> <p>注：如果 TI1F_EN 被选为触发输入(TS=100)时，不要使用门控模式。</p> <p>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。</p>	0	R/W

表 13.3. TIM1 内部触发连接

从定时器	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM1	Tim2_trgo	irq_timer10	irq_timer11	irq_pca

13.5.4 TIM1 中断使能寄存器(TIM1_DIER)

地址偏移：0x0C

复位值：0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:8	—	保留，始终读为 0。	0	—
7	BIE	允许刹车中断(Break interrupt enable) 0: 禁止刹车中断; 1: 允许刹车中断。	0	R/W
6	TIE	触发中断使能(Trigger interrupt enable) 0: 禁止触发中断; 1: 使能触发中断。	0	R/W
5	COMIE	允许 COM 中断(COM interrupt enable) 0: 禁止 COM 中断; 1: 允许 COM 中断。	0	R/W
4	CC4IE	允许捕获/比较 4 中断(Capture / Compare 4 interrupt enable) 0: 禁止捕获/比较 4 中断; 1: 允许捕获/比较 4 中断。	0	R/W
3	CC3IE	允许捕获/比较 3 中断(Capture / Compare 3 interrupt enable) 0: 禁止捕获/比较 3 中断; 1: 允许捕获/比较 3 中断。	0	R/W
2	CC2IE	允许捕获/比较 2 中断(Capture / Compare 2 interrupt enable) 0: 禁止捕获/比较 2 中断; 1: 允许捕获/比较 2 中断。	0	R/W
1	CC1IE	允许捕获/比较 1 中断(Capture / Compare 1 interrupt enable) 0: 禁止捕获/比较 1 中断; 1: 允许捕获/比较 1 中断。	0	R/W
0	UIE	允许更新中断(Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。	0	R/W

13.5.5 TIM1 状态寄存器(TIM1_SR)

地址偏移: 0x10

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			CC4 OF	CC3 OF	CC2 OF	CC1 OF	保留	BIF	TIF	COM IF	CC4IF	CC3IF	CC2IF	CC1I F	UIF
			RC W0	RC W0	RC W0	RC W0		RC W0	RC W0	RC W0	RC W0	RC W0	RC W0	RC W0	RC W0

位	标记	功能描述	复位值	读写
31:13	–	保留，始终读为 0。	0	–
12	CC4OF	捕获/比较 4 重复捕获标记(Capture /Compare 4 over capture flag) 参见 CC1OF 描述。	0	RCW 0
11	CC3OF	捕获/比较 3 重复捕获标记(Capture /Compare 3 over capture flag) 参见 CC1OF 描述。	0	RCW 0
10	CC2OF	捕获/比较 2 重复捕获标记(Capture /Compare 2 over capture flag) 参见 CC1OF 描述。	0	RCW 0
9	CC1OF	捕获/比较 1 重复捕获标记(Capture /Compare 1 over capture flag) 仅当相应的通道被配置为输入捕获时，该标记可由硬件置 1。 写 0 可清除该位。 0：无重复捕获产生； 1：计数器的值被捕获到 TIM1_CCR1 寄存器时，CC1IF 的状态已经为‘1’。	0	RCW 0
8	–	保留，始终读为 0。	0	–
7	BIF	刹车中断标记(Break interrupt flag) 一旦刹车输入有效，由硬件对该位置‘1’。如果刹车输入无效，则该位可由软件清‘0’。 0：无刹车事件产生； 1：刹车输入上检测到有效电平。	0	RCW 0
6	TIF	触发器中断标记(Trigger interrupt flag) 当发生触发事件(当从模式控制器处于除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿)时由硬件对该位置‘1’。它由软件清‘0’。 0：无触发器事件产生； 1：触发中断等待响应。	0	RCW 0
5	COMIF	COM 中断标记(COM interrupt flag) 一旦产生 COM 事件(当捕获/比较控制位：CCxE、CCxNE、OCxM 已被更新)该位由硬件置‘1’。它由软件清‘0’。 0：无 COM 事件产生； 1：COM 中断等待响应。	0	RCW 0
4	CC4IF	捕获/比较 4 中断标记(Capture / Compare 4 interrupt flag) 参考 CC1IF 描述。	0	RCW 0
3	CC3IF	捕获/比较 3 中断标记(Capture / Compare 3 interrupt flag) 参考 CC1IF 描述。	0	RCW 0
2	CC2IF	捕获/比较 2 中断标记(Capture / Compare 2 interrupt flag) 参考 CC1IF 描述。	0	RCW 0
1	CC1IF	捕获/比较 1 中断标记(Capture / Compare 1 interrupt flag) 如果通道 CC1 配置为输出模式：当计数器值与比较值匹配时该位由硬件置 1，但在中心对称模式下除外(参考 TIM1_CR1 寄存器的 CMS 位)。它由软件清‘0’。 0：无匹配发生； 1：TIM1_CNT 的值与 TIM1_CCR1 的值匹配。 当 TIM1_CCR1 的内容大于 TIM1_APR 的内容时，在向上或向上/下计数模式时计数器溢出，或向下计数模式时的计数器下	0	RCW 0

位	标记	功能描述	复位值	读写
		溢条件下, CC1IF 位变高 如果通道 CC1 配置为输入模式: 当捕获事件发生时该位由硬件置'1', 它由软件清'0'或通过读 TIM1_CCR1 清'0'。 0: 无输入捕获产生; 1: 计数器值已被捕获(拷贝)至 TIM1_CCR1(在 IC1 上检测到与所选极性相同的边沿)。		
0	UIF	更新中断标记(Update interrupt flag)当产生更新事件时该位由硬件置'1'。它由软件清'0'。 0: 无更新事件产生; 1: 更新中断等待响应。当寄存器被更新时该位由硬件置'1': - 若 TIM1_CR1 寄存器的 UDIS=0, 当重复计数器数值上溢或下溢时(重复计数器=0 时产生更新事件)。 - 若 TIM1_CR1 寄存器的 URS=0、UDIS=0, 当设置 TIM1_EGR 寄存器的 UG=1 时产生更新事件, 通过软件对计数器 CNT 重新初始化时。 - 若 TIM1_CR1 寄存器的 URS=0、UDIS=0, 当计数器 CNT 被触发事件重新初始化时。 (参考 13.5.3TIM1 从模式控制寄存器(TIM1_SMCR))。	0	RCW 0

13.5.6 TIM1 事件产生 Th 寄存器(TIM1_EGR)

地址偏移: 0x14

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
								WO	WO	WO	WO	WO	WO	WO	WO

位	标记	功能描述	复位值	读写
31:8	—	保留, 始终读为 0。	0x0	—
7	BG	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, 则产生相应的中断。	0	WO
6	TG	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	WO

位	标记	功能描述	复位值	读写
		0: 无动作; 1: TIM1_SR 寄存器的 TIF=1, 若开启对应的中断, 则产生相应的中断。		
5	COMG	捕获/比较事件, 产生控制更新(Capture / Compare control update generation)该位由软件置'1', 由硬件自动清'0'。 0: 无动作; 1: 当 CCPC=1, 允许更新 CCxE、CCxNE、OCxM 位。注: 该位只对拥有互补输出的通道有效。	0	WO
4	CC4G	产生捕获/比较 4 事件(Capture / Compare 4 generation) 参考 CC1G 描述。	0	WO
3	CC3G	产生捕获/比较 3 事件(Capture / Compare 3 generation) 参考 CC1G 描述。	0	WO
2	CC2G	产生捕获/比较 2 事件(Capture / Compare 2 generation) 参考 CC1G 描述。	0	WO
1	CC1G	产生捕获/比较 1 事件(Capture / Compare 1 generation) 该位由软件置'1', 用于产生一个捕获/比较事件, 由硬件自动清'0'。 0: 无动作; 1: 在通道 CC1 上产生一个捕获/比较事件: 若通道 CC1 配置为输出: 设置 CC1IF=1, 若开启对应的中断, 则产生相应的中断。 若通道 CC1 配置为输入: 当前的计数器值被捕获至 TIM1_CCR1 寄存器; 设置 CC1IF=1, 若开启对应的中断, 则产生相应的中断。若 CC1IF 已经为 1, 则设置 CC1OF=1。	0	WO
0	UG	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 TIM1_ARR 的值。	0	WO

13.5.7 TIM1 捕获/比较模式寄存器 1(TIM1_CCMR1)

地址偏移: 0x18

复位值: 0x00000000

通道可用于输入(捕获模式)或输出(比较模式), 通道的方向由相应的 CCxS 位定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能, ICxx 描述了通道在输入模式下的功能。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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		OC1 CE	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
IC2F[3:0]			IC2PSC[1:0]		IC1F[3:0]			IC1PSC[1:0]							
R/W															

输出比较模式:

位	标记	功能描述	复位值	读写
31:16	—	保留，始终读为 0。	0x0	—
15	OC2CE	输出比较 2 清 0 使能(Output Compare 2 clear enable)	0	R/W
14:12	OC2M[2:0]	输出比较 2 模式(Output Compare 2 mode)	0x0	R/W
11	OC2PE	输出比较 2 预装载使能(Output Compare 2 preload enable)	0	R/W
10	OC2FE	输出比较 2 快速使能(Output Compare 2 fast enable)	0	R/W
9:8	CC2S[1:0]	捕获/比较 2 选择。(Capture /Compare 2 selection) 该位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC2 通道被配置为输出； 01: CC2 通道被配置为输入，IC2 映射在 TI2 上； 10: CC2 通道被配置为输入，IC2 映射在 TI1 上； 11: CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM1_SMCR 寄存器的 TS 位选择)。 注：CC2S 仅在通道关闭时(TIM1_CCER 寄存器的 CC2E=0)才是可写的。	0x0	R/W
7	OC1CE	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。	0	R/W
6:4	OC1M[2:0]	输出比较 1 模式(Output Compare 1 mode) 该 3 位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效，而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 000: 冻结。输出比较寄存器 TIM1_CCR1 与计数器 TIM1_CNT 间的比较对 OC1REF 不起作用； 001: 匹配时设置通道 1 为有效电平。当计数器 TIM1_CNT 的值与捕获/比较寄存器 1(TIM1_CCR1)相同时，强制 OC1REF 为高。 010: 匹配时设置通道 1 为无效电平。当计数器 TIM1_CNT 的值与捕获/比较寄存器 1(TIM1_CCR1)相同时，强制 OC1REF 为低。 011: 翻转。当 TIM1_CCR1=TIM1_CNT 时，翻转 OC1REF 的电平。 100: 强制为无效电平。强制 OC1REF 为低。 101: 强制为有效电平。强制 OC1REF 为高。 110: PWM 模式 1 — 在向上计数时，一旦	0x0	R/W

位	标记	功能描述	复位值	读写
		<p>TIM1_CNT<TIM1_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 TIM1_CNT>TIM1_CCR1 时通道 1 为无效电平(OC1REF=0), 否则为有效电平(OC1REF=1)。</p> <p>111: PWM 模式 2 — 在向上计数时, 一旦 TIM1_CNT<TIM1_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 TIM1_CNT>TIM1_CCR1 时通道 1 为有效电平, 否则为无效电平。</p> <p>注 1: 一旦 LOCK 级别设为 3(TIM1_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。</p> <p>注 2: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。</p>		
3	OC1PE	<p>输出比较 1 预装载使能(Output Compare 1 preload enable)</p> <p>0: 禁止 TIM1_CCR1 寄存器的预装载功能, 可随时写入 TIM1_CCR1 寄存器, 并且新写入的数值立即起作用。</p> <p>1: 开启 TIM1_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM1_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中。</p> <p>注 1: 一旦 LOCK 级别设为 3(TIM1_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。</p> <p>注 2: 仅在单脉冲模式下(TIM1_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。</p>	0	R/W
2	OC1FE	<p>输出比较 1 快速使能(Output Compare 1 fast enable)该位用于加快 CC 输出对触发输入事件的响应。</p> <p>0: 根据计数器与 CCR1 的值, CC1 正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活 CC1 输出的小延时为 5 个时钟周期。</p> <p>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。</p> <p>OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。</p>	0	R/W
1:0	CC1S[1:0]	<p>捕获/比较 1 选择。(Capture / Compare 1 selection)</p> <p>这 2 位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1 通道被配置为输出;</p> <p>01: CC1 通道被配置为输入, IC1 映射在 TI1 上;</p> <p>10: CC1 通道被配置为输入, IC1 映射在 TI2 上;</p> <p>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 TIM1_SMCR 寄存器的 TS 位选择)。</p> <p>注: CC1S 仅在通道关闭时(TIM1_CCER 寄存器的</p>	0x0	R/W

位	标记	功能描述	复位值	读写
		CC1E=0)才是可写的。		

输入捕获模式：

位	标记	功能描述	复位值	读写
31:16	–	保留，始终读为 0。	0x0	–
15:12	IC2F[3:0]	输入捕获 2 滤波器(Input capture 2 filter)	0x0	R/W
11:10	IC2PSC[1:0]	输入/捕获 2 预分频器(Input capture 2 prescaler)	0x0	R/W
9:8	CC2S[1:0]	捕获/比较 2 选择(Capture /Compare 2 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC2 通道被配置为输出； 01: CC2 通道被配置为输入，IC2 映射在 TI2 上； 10: CC2 通道被配置为输入，IC2 映射在 TI1 上； 11: CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM1_SMCR 寄存器的 TS 位选择)。 注：CC2S 仅在通道关闭时(TIM1_CCER 寄存器的 CC2E=0)才是可写的。	0x0	R/W
7:4	IC1F[3:0]	输入捕获 1 滤波器(Input capture 1 filter) 这几位定义了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变： 0000: 无滤波器，以 f_{DTS} 采样 0001: 采样频率 $f_{SAMPLING}=f_{CK_INT}$ ，N=2 0010: 采样频率 $f_{SAMPLING}=f_{CK_INT}$ ，N=4 0011: 采样频率 $f_{SAMPLING}=f_{CK_INT}$ ，N=8 0100: 采样频率 $f_{SAMPLING}=f_{DTS}/2$ ，N=6 0101: 采样频率 $f_{SAMPLING}=f_{DTS}/2$ ，N=8 0110: 采样频率 $f_{SAMPLING}=f_{DTS}/4$ ，N=6 0111: 采样频率 $f_{SAMPLING}=f_{DTS}/4$ ，N=8 1000: 采样频率 $f_{SAMPLING}=f_{DTS}/8$ ，N=6 1001: 采样频率 $f_{SAMPLING}=f_{DTS}/8$ ，N=8 1010: 采样频率 $f_{SAMPLING}=f_{DTS}/16$ ，N=5 1011: 采样频率 $f_{SAMPLING}=f_{DTS}/16$ ，N=6 1100: 采样频率 $f_{SAMPLING}=f_{DTS}/16$ ，N=8 1101: 采样频率 $f_{SAMPLING}=f_{DTS}/32$ ，N=5 1110: 采样频率 $f_{SAMPLING}=f_{DTS}/32$ ，N=6 1111: 采样频率 $f_{SAMPLING}=f_{DTS}/32$ ，N=8	0x0	R/W
3:2	IC1PSC[1:0]	输入/捕获 1 预分频器(Input capture 1 prescaler) 这 2 位定义了 CC1 输入(IC1)的预分频系数。 一旦 CC1E=0(TIM1_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；	0x0	R/W

位	标记	功能描述	复位值	读写
		10: 每 4 个事件触发一次捕获; 11: 每 8 个事件触发一次捕获。		
1:0	CC1S[1:0]	捕获/比较 1 选择(Capture /Compare 1 Selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1 通道被配置为输出; 01: CC1 通道被配置为输入, IC1 映射在 TI1 上; 10: CC1 通道被配置为输入, IC1 映射在 TI2 上; 11: CC1 通道被配置为输入, IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM1_SMCR 寄存器的 TS 位选择)。 注: CC1S 仅在通道关闭时(TIM1_CCER 寄存器的 CC1E=0)才是可写的。	0x0	R/W

13.5.8 TIM1 捕获/比较模式寄存器 2(TIM1_CCMR2)

地址偏移: 0x1C

复位值: 0x00000000

参看以上 CCMR1 寄存器的描述。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4 CE	OC4M[2:0]			OC4 PE	OC4 FE	CC4S[1:0]		OC3 CE	OC3M[2:0]			OC3 PE	OC3 FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]			IC3PSC[1:0]				
R/W															

输出比较模式:

位	标记	功能描述	复位值	读写
31:16	—	保留, 始终读为 0。	0x0	—
15	OC4CE	输出比较 4 清 0 使能(Output Compare 4 clear enable)	0	R/W
14:12	OC4M[2:0]	输出比较 4 模式(Output Compare 4 mode)	0x0	R/W
11	OC4PE	输出比较 4 预装载使能(Output Compare 4 preload enable)	0	R/W
10	OC4FE	输出比较 4 快速使能(Output Compare 4 fast enable)	0	R/W
9:8	CC4S[1:0]	捕获/比较 4 选择。(Capture /Compare 4 selection) 该 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC4 通道被配置为输出; 01: CC4 通道被配置为输入, IC4 映射在 TI4 上; 10: CC4 通道被配置为输入, IC4 映射在 TI3 上; 11: CC4 通道被配置为输入, IC4 映射在 TRC 上。此模式	0x0	R/W

位	标记	功能描述	复位值	读写
		仅工作在内部触发器输入被选中时(由 TIM1_SMCR 寄存器的 TS 位选择)。 注：CC4S 仅在通道关闭时(TIM1_CCER 寄存器的 CC4E=0)才是可写的。		
7	OC3CE	输出比较 3 清除使能(Output Compare 3 clear enable)	0	R/W
6:4	OC3M[2:0]	输出比较 3 模式(Output Compare 3 mode)	0x0	R/W
3	OC3PE	输出比较 3 预装载使能(Output Compare 3 preload enable)	0	R/W
2	OC3FE	输出比较 3 快速使能(Output Compare 3 fast enable)	0	R/W
1:0	CC3S[1:0]	捕获/比较 3 选择。(Capture /Compare 3 selection) 这 2 位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC3 通道被配置为输出; 01: CC3 通道被配置为输入, IC3 映射在 TI3 上; 10: CC3 通道被配置为输入, IC3 映射在 TI4 上; 11: CC3 通道被配置为输入, IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TIM1_SMCR 寄存器的 TS 位选择)。 注：CC3S 仅在通道关闭时(TIM1_CCER 寄存器的 CC3E=0)才是可写的。	0x0	R/W

输入捕获模式:

位	标记	功能描述	复位值	读写
31:16	—	保留, 始终读为 0。	0x0	—
15:12	IC4F[3:0]	输入捕获 4 滤波器(Input capture 4 filter)	0x0	R/W
11:10	IC4PSC[1:0]	输入/捕获 4 预分频器(Input capture 4 prescaler)	0x0	R/W
9:8	CC4S[1:0]	捕获/比较 4 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: 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 TIM1_SMCR 寄存器的 TS 位选择)。 注：CC4S 仅在通道关闭时(TIM1_CCER 寄存器的 CC4E=0)才是可写的。	0x0	R/W
7:4	IC3F[3:0]	输入捕获 3 滤波器(Input capture 1 filter)	0x0	R/W
3:2	IC3PSC[1:0]	输入/捕获 3 预分频器(Input capture 1 prescaler)	0x0	R/W
1:0	CC3S[1:0]	捕获/比较 3 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: 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 TIM1_SMCR 寄存	0x0	R/W

位	标记	功能描述	复位值	读写
		器的 TS 位选择)。 注：CC3S 仅在通道关闭时(TIM1_CCER 寄存器的 CC3E=0)才是可写的。		

13.5.9 TIM1 捕获/比较使能寄存器(TIM1_CCER)

地址偏移：0x20

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CC 4P	CC 4E	CC 3NP	CC 3NE	CC 3P	CC 3E	CC 2NP	CC 2NE	CC 2P	CC 2E	CC 1NP	CC 1NE	CC 1P	CC 1E	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:14	—	保留，始终读为 0。	0x0	—
13	CC4P	输入/捕获 4 输出极性(Capture /Compare 4 output polarity) 参考 CC1P 的描述。	0	R/W
12	CC4E	输入/捕获 4 输出使能(Capture /Compare 4 output enable) 参考 CC1E 的描述。	0	R/W
11	CC3NP	输入 / 捕获 3 互补输出极性 (Capture /Compare 3 complementary output polarity) 参考 CC1NP 的描述。	0	R/W
10	CC3NE	输入 / 捕获 3 互补输出使能 (Capture /Compare 3 complementary output enable) 参考 CC1NE 的描述。	0	R/W
9	CC3P	输入/捕获 3 输出极性(Capture /Compare 3 output polarity) 参考 CC1P 的描述。	0	R/W
8	CC3E	输入/捕获 3 输出使能(Capture /Compare 3 output enable) 参考 CC1E 的描述。	0	R/W
7	CC2NP	输入 / 捕获 2 互补输出极性 (Capture /Compare 2 complementary output polarity) 参考 CC1NP 的描述。	0	R/W
6	CC2NE	输入 / 捕获 2 互补输出使能 (Capture /Compare 2 complementary output enable) 参考 CC1NE 的描述。	0	R/W
5	CC2P	输入/捕获 2 输出极性(Capture /Compare 2 output polarity) 参考 CC1P 的描述。	0	R/W
4	CC2E	输入/捕获 2 输出使能(Capture /Compare 2 output enable)	0	R/W

位	标记	功能描述	复位值	读写
		参考 CC1E 的描述。		
3	CC1NP	输入 / 捕获 1 互补输出极性 (Capture /Compare 1 complementary output polarity) 0: OC1N 高电平有效; 1: OC1N 低电平有效。 注: 一旦 LOCK 级别(TIM1_BDTR 寄存器中的 LOCK 位) 设为 3 或 2 且 CC1S=00(通道配置为输出)则该位不能被修改。	0	R/W
2	CC1NE	输入 / 捕获 1 互补输出使能 (Capture /Compare 1 complementary output enable) 0: 关闭—OC1N 禁止输出, 因此 OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。 1: 开启—OC1N 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。	0	R/W
1	CC1P	输入/捕获 1 输出极性(Capture /Compare 1 output polarity) CC1 通道配置为输出: 0: OC1 高电平有效; 1: OC1 低电平有效。 CC1 通道配置为输入: 该位选择是 IC1 还是 IC1 的反相信号作为触发或捕获信号。 0: 不反相: 捕获发生在 IC1 的上升沿; 当用作外部触发器时, IC1 不反相。 1: 反相: 捕获发生在 IC1 的下降沿; 当用作外部触发器时, IC1 反相。 注: 一旦 LOCK 级别(TIM1_BDTR 寄存器中的 LOCK 位) 设为 3 或 2, 则该位不能被修改。	0	R/W
0	CC1E	输入/捕获 1 输出使能(Capture /Compare 1 output enable) CC1 通道配置为输出: 0: 关闭—OC1 禁止输出, 因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 1: 开启—OC1 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 CC1 通道配置为输入: 该位决定了计数器的值是否能捕获入 TIM1_CCR1 寄存器。 0: 捕获禁止; 1: 捕获使能。	0	R/W

表 13.4. 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位

控制位					输出状态	
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。

13.5.10 TIM1 计数器(TIM1_CNT)

地址偏移: 0x24

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:16	—	保留, 始终读为 0。	0x0	—
15:0	CNT[15:0]	计数器的值(Counter value)	0x0	R/W

13.5.11 TIM1 预分频器(TIM1_PSC)

地址偏移: 0x28

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:16	—	保留, 始终读为 0。	0x0	—
15:0	PSC[15:0]	预分频器的值(Prescaler value) 计数器的时钟频率(CK_CNT)等于 $f_{CK_PSC}/(PSC[15:0]+1)$ 。 PSC 包含了每次当更新事件产生时, 装入当前预分频器寄存器的值; 更新事件包括计数器被 TIM_EGR 的 UG 位清'0' 或被工作在复位模式的从控制器清'0'。	0x0	R/W

13.5.12 TIM1 自动重装载寄存器(TIM1_ARR)

地址偏移: 0x2C

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:16	—	保留, 始终读为 0。	0x0	—
15:0	ARR[15:0]	自动重装载的值(Prescaler value) ARR 包含了将要装载入实际的自动重装载寄存器的值。详细参考 13.3.1 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	R/W

13.5.13 TIM1 重复计数寄存器(TIM1_RCR)

地址偏移: 0x30

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:8	—	保留, 始终读为 0。	0x0	—
7:0	REP[7:0]	重复计数器的值(Repetition counter value) 开启了预装载功能后, 这些位允许用户设置比较寄存器的更新速率(即周期性地从预装载寄存器传输到当前寄存器); 如果允许产生更新中断, 则会同时影响产生更新中断的速率。 每次向下计数器 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	0x0	R/W

位	标记	功能描述	复位值	读写
		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;		

13.5.14 TIM1 捕获/比较寄存器 1(TIM1_CCR1)

地址偏移: 0x34

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	—	保留, 始终读为 0。	0x0	—
15:0	CCR1[15:0]	捕获/比较通道 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 TIM1_CCMR1 寄存器(OC1PE 位)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 1 寄存器中。 当前捕获/比较寄存器参与同计数器 TIM1_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	R/W

13.5.15 TIM1 捕获/比较寄存器 2(TIM1_CCR2)

地址偏移: 0x38

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	—	保留，始终读为 0。	0x0	—
15:0	CCR2[15:0]	捕获/比较通道 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 TIM1_CCMR2 寄存器(OC2PE 位)中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 2 寄存器中。 当前捕获/比较寄存器参与同计数器 TIM1_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	R/W

13.5.16 TIM1 捕获/比较寄存器 3(TIM1_CCR3)

地址偏移：0x3C

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	—	保留，始终读为 0。	0x0	—
15:0	CCR3[15:0]	捕获/比较通道 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)。	0x0	R/W

位	标记	功能描述	复位值	读写
		如果在 TIM1_CCMR3 寄存器(OC3PE 位)中未选择预装载特性, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 3 寄存器中。 当前捕获/比较寄存器参与同计数器 TIM1_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。		

13.5.17 TIM1 捕获/比较寄存器 4(TIM1_CCR4)

地址偏移: 0x40

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	—	保留, 始终读为 0。	0x0	—
15:0	CCR4[15:0]	捕获/比较通道 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 TIM1_CCMR4 寄存器(OC4PE 位)中未选择预装载特性, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 4 寄存器中。 当前捕获/比较寄存器参与同计数器 TIM1_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	R/W

13.5.18 TIM1 刹车和死区寄存器(TIM1_BDTR)

地址偏移: 0x44

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W							

位	标记	功能描述	复位值	读写
31:16	—	保留, 始终读为 0。	0x0	—
15	MOE	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 13.5.9 TIM1 和 TIM8 捕获/比较使能寄存器(TIM1_CCER)。	0	R/W
14	AOE	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。	0	R/W
13	BKP	刹车输入极性(Break polarity) 0: 刹车输入低电平有效; 1: 刹车输入高电平有效。 注: 一旦 LOCK 级别(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。	0	R/W
12	BKE	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。	0	R/W
11	OSSR	运行模式下“关闭状态”选择(Off-state selection for Run	0	R/W

位	标记	功能描述	复位值	读写
		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(13.5.9 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。		
10	OSSI	空闲模式下“关闭状态”选择(Off-stateselectionforIdlemode) 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(13.5.9 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。	0	R/W
9:8	LOOK[1:0]	锁定设置(Lock configuration)该位为防止软件错误而提供写保护。 00: 锁定关闭, 寄存器无写保护; 01: 锁定级别 1, 不能写入 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	R/W
7:0	UTG[7:0]	死区发生器设置(Dead-time generator setup) 这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间: DTG[7:5]=0xx=>DT=DTG[7:0]×Tdtg,Tdtg=TDTS; DTG[7:5]=10x=>DT=(64+DTG[5:0])×Tdtg,Tdtg=2×TDTS; DTG[7:5]=110=>DT=(32+DTG[4:0])×Tdtg,Tdtg=8×TDTS; DTG[7:5]=111=>DT=(32+DTG[4:0])×Tdtg,Tdtg=16×TDTS; ;	0x0	R/W

位	标记	功能描述	复位值	读写
		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。		

14. 通用定时器 (TIM2)

14.1 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。

14.2 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：
 - 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

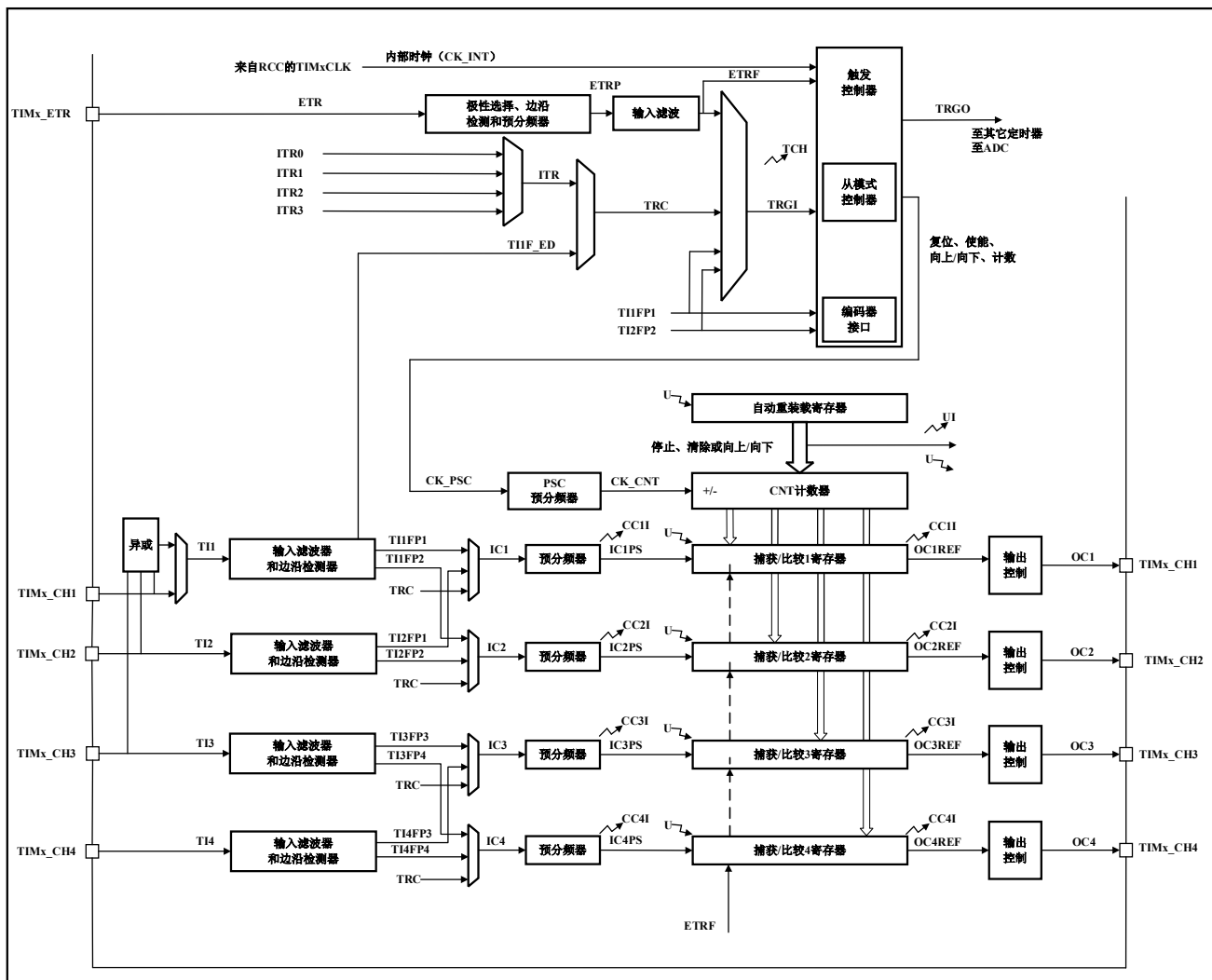


图 14-1. 通用定时器框图

注：根据控制位的设定，在 U(更新)事件时传送预加载寄存器的内容至工作寄存器

事件

中断

14.3 TIM2 功能描述

14.3.1 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，写或读自动重装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。

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一次更新事件到来时被采用。

图 14-2 和图 14-3 给出了在预分频器运行时，更改计数器参数的例子。

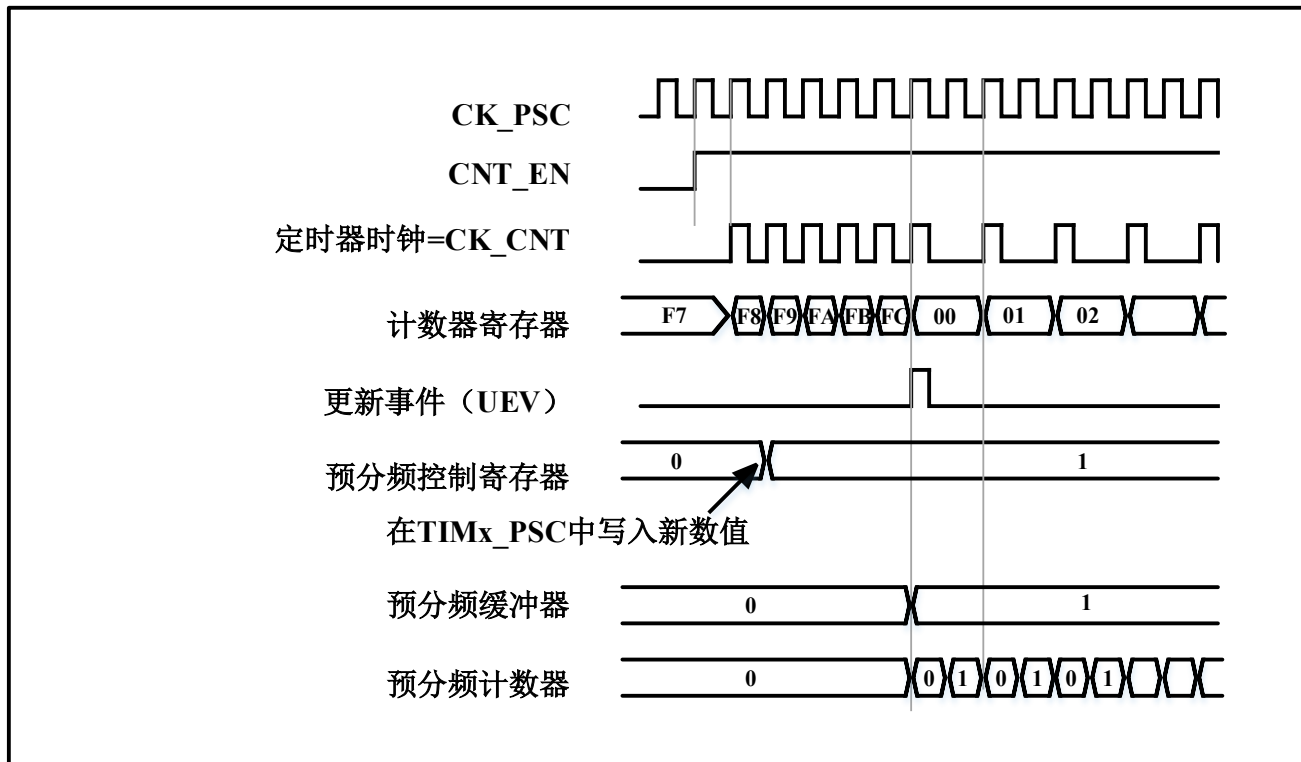


图 14-2. 当预分频器的参数从 1 变到 2 时，计数器的时序图

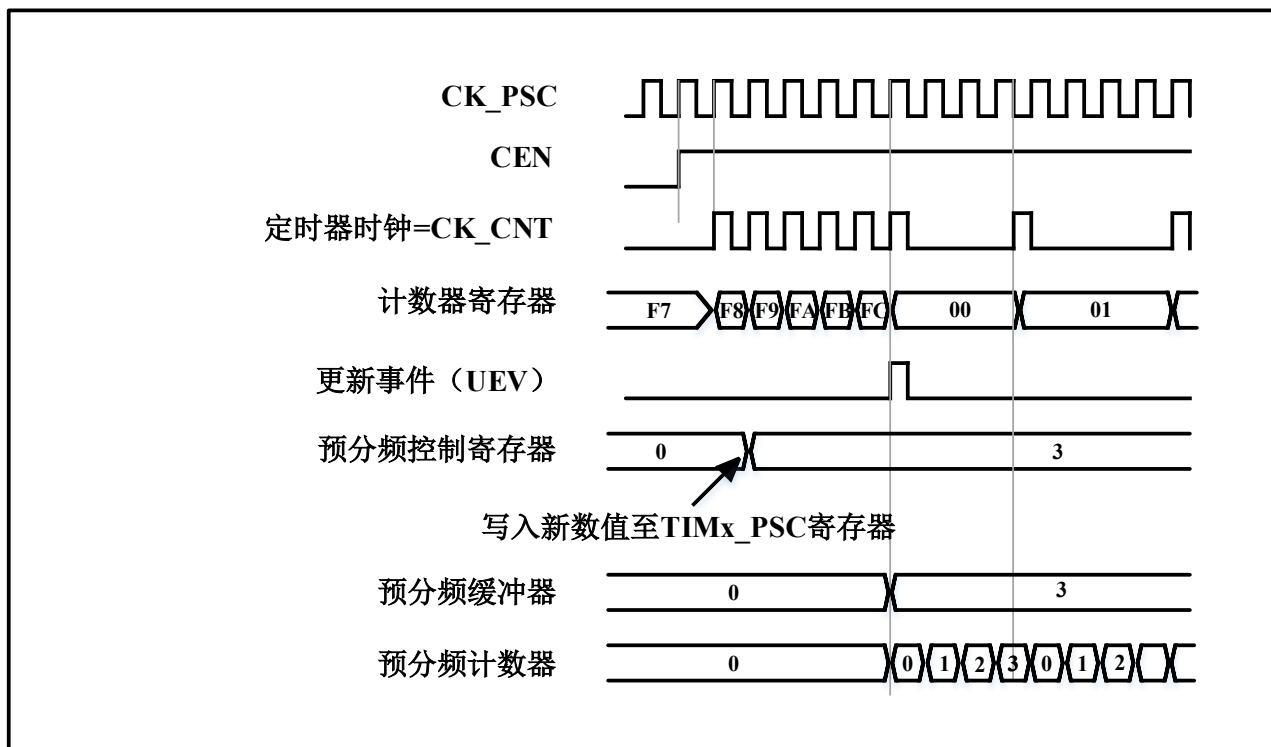


图 14-3. 当预分频器的参数从 1 变到 4 时，计数器的时序图

14.3.2 计数器模式

14.3.2.1 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)；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。

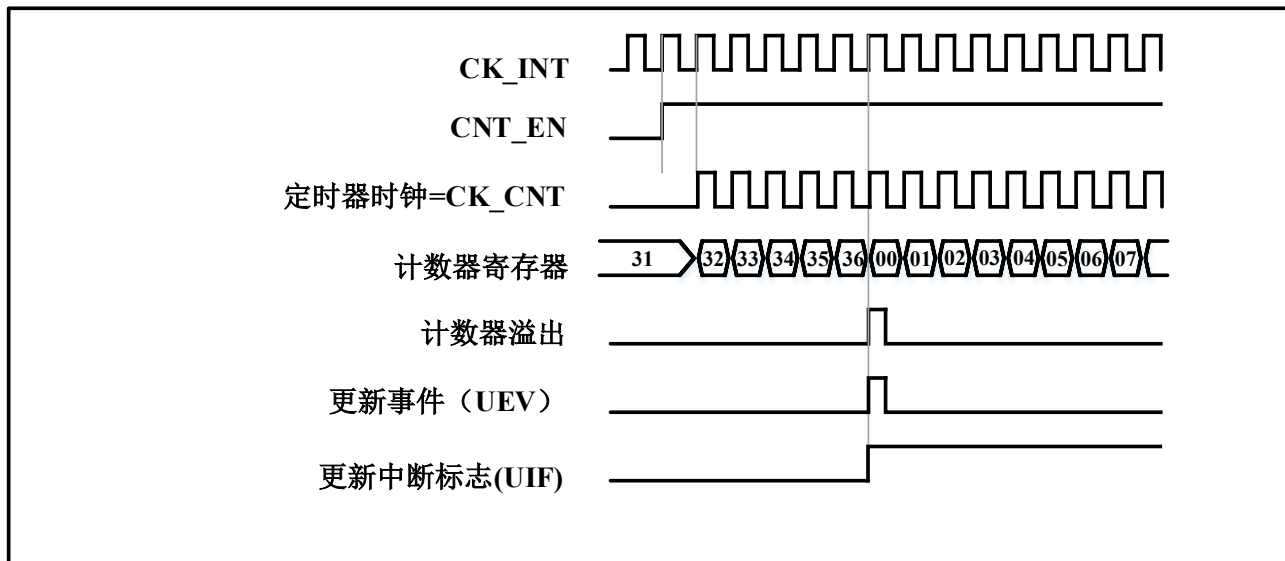


图 14-4. 计数器时序图，内部时钟分频因子为 1

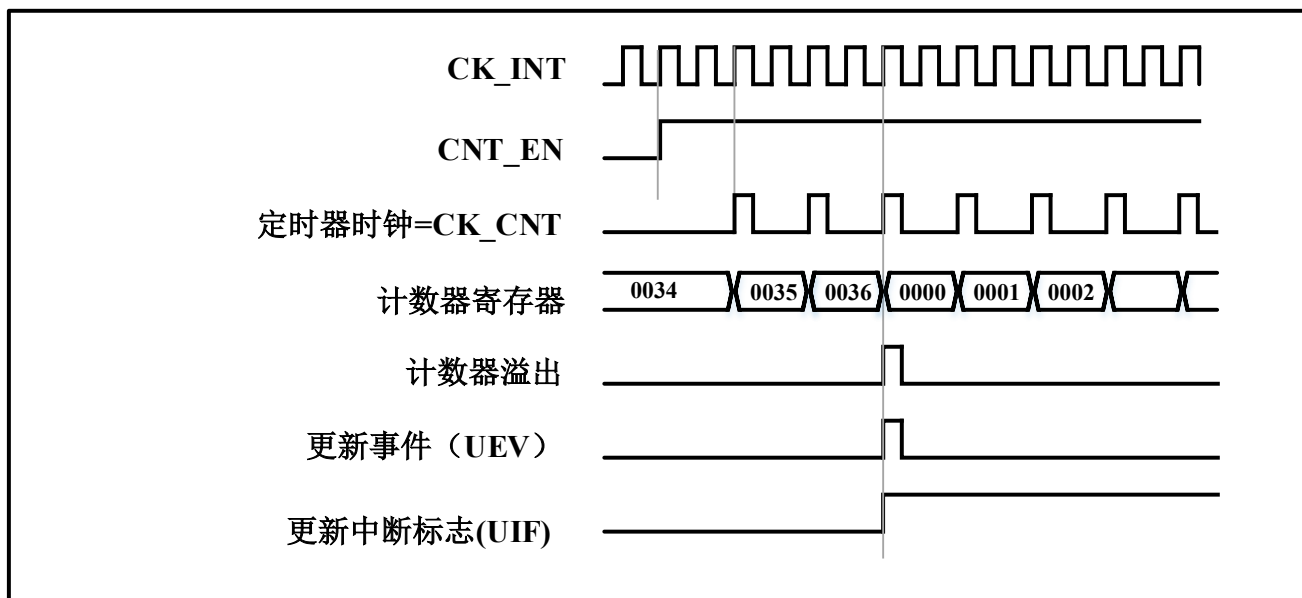


图 14-5. 计数器时序图，内部时钟分频因子为 2

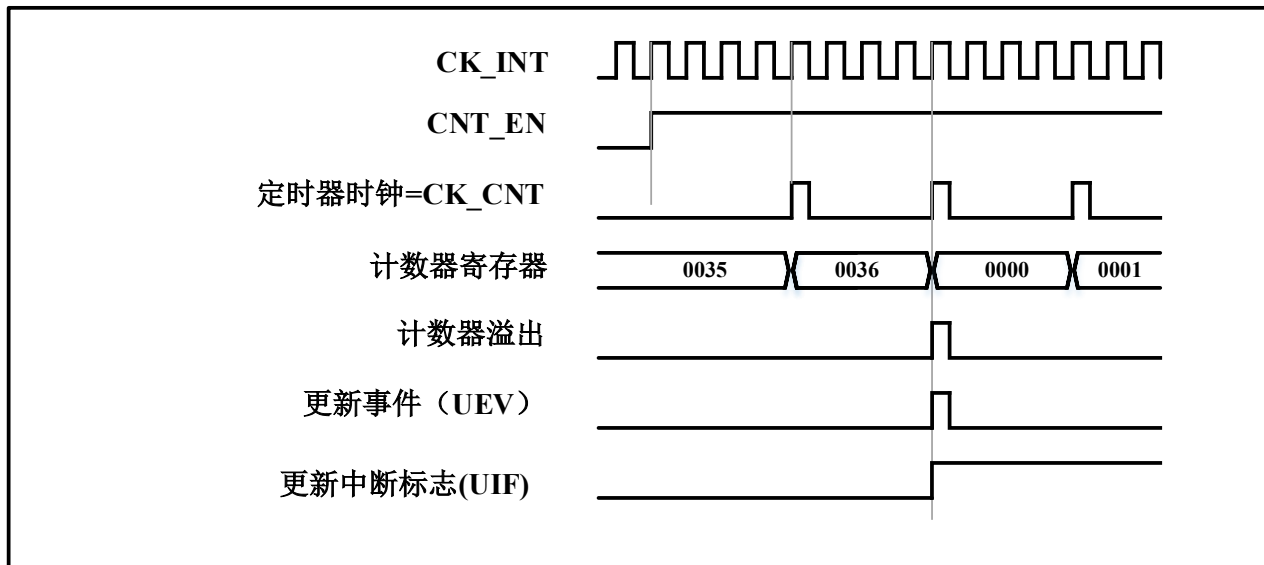


图 14-6. 计数器时序图，内部时钟分频因子为 4

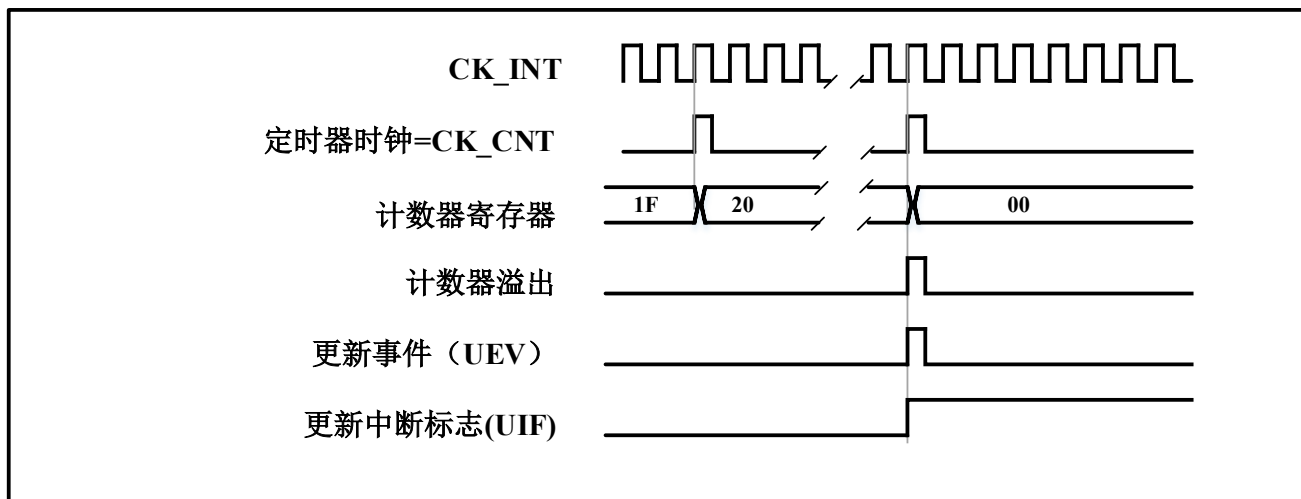


图 14-7. 计数器时序图，内部时钟分频因子为 N

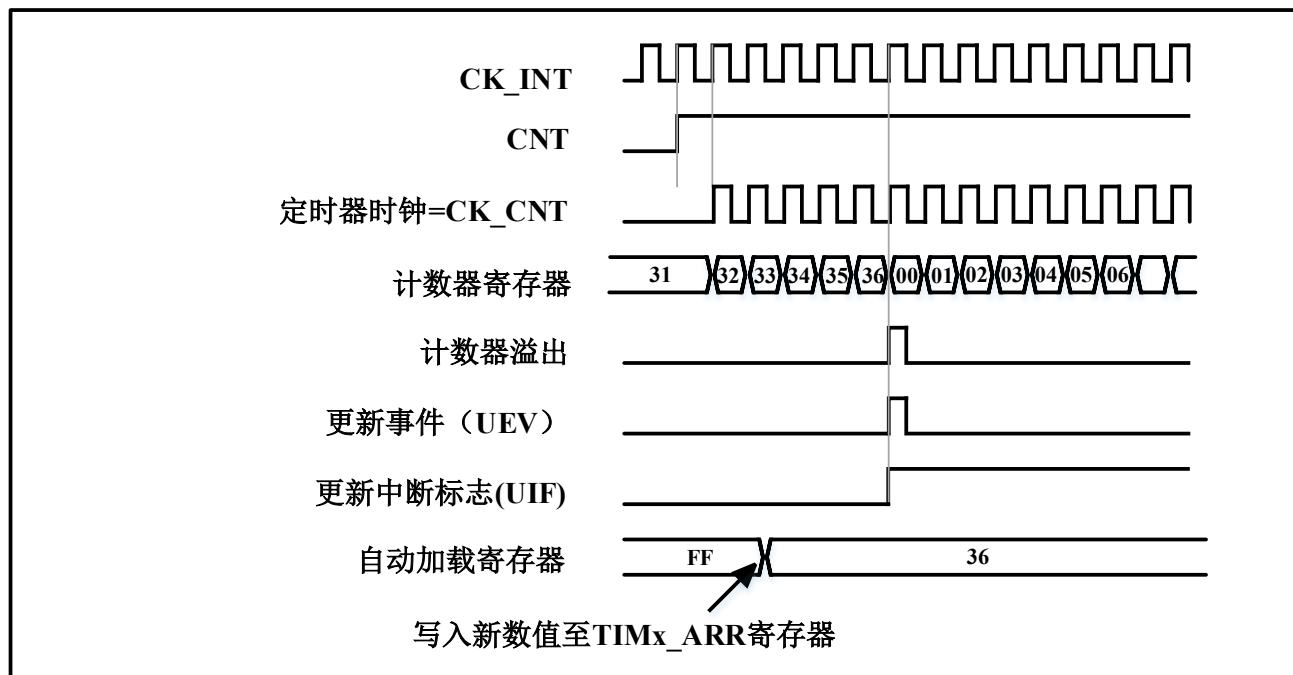


图 14-8. 计数器时序图，当 ARPE=0 时的更新事件(TIM2_ARR 没有预装入)

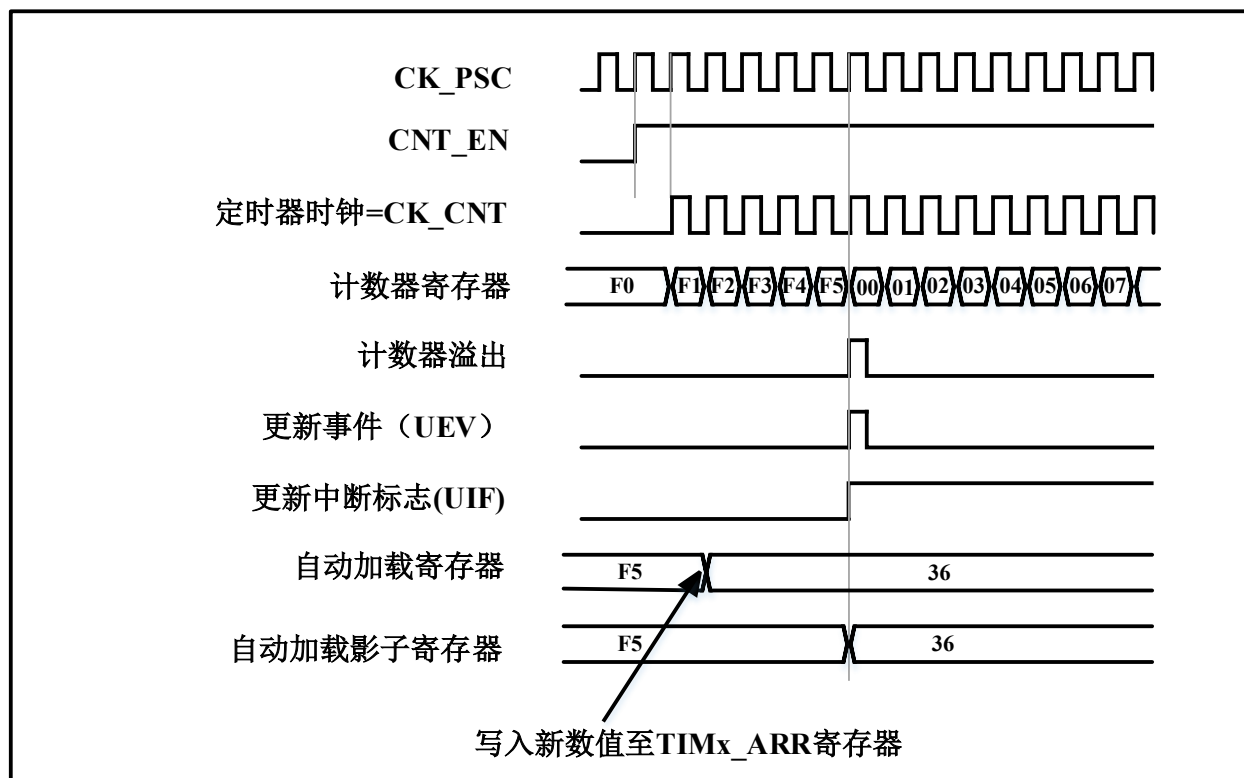


图 14-9. 计数器时序图，当 ARPE=1 时的更新事件(预装入了 TIM2_ARR)

14.3.2.2 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)，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。

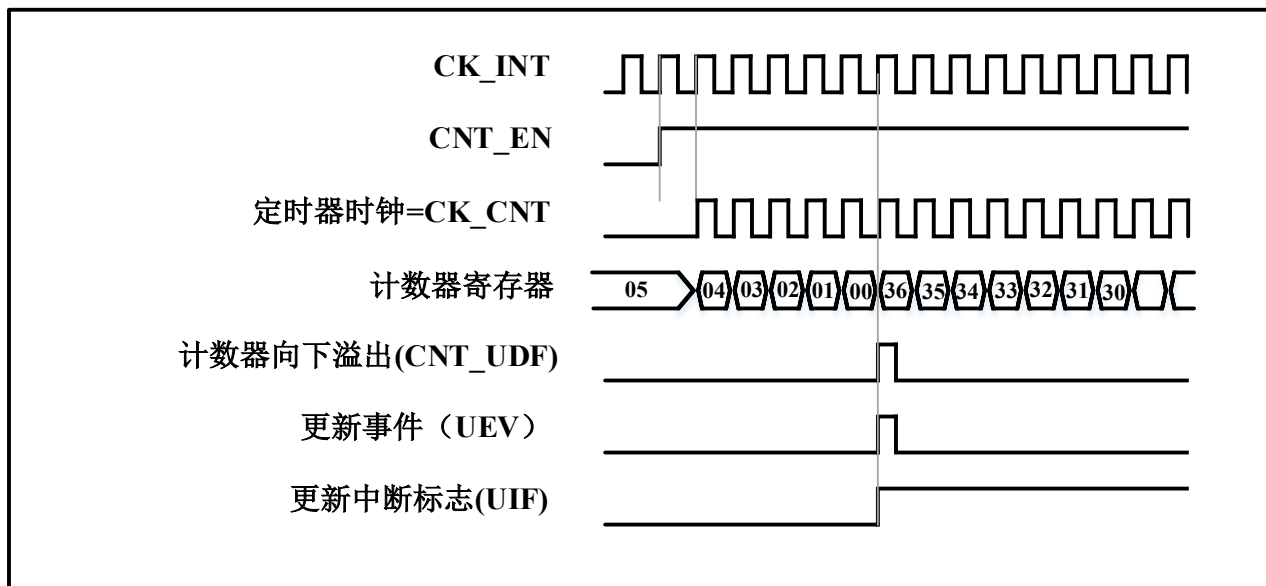


图 14-10. 计数器时序图，内部时钟分频因子为 1

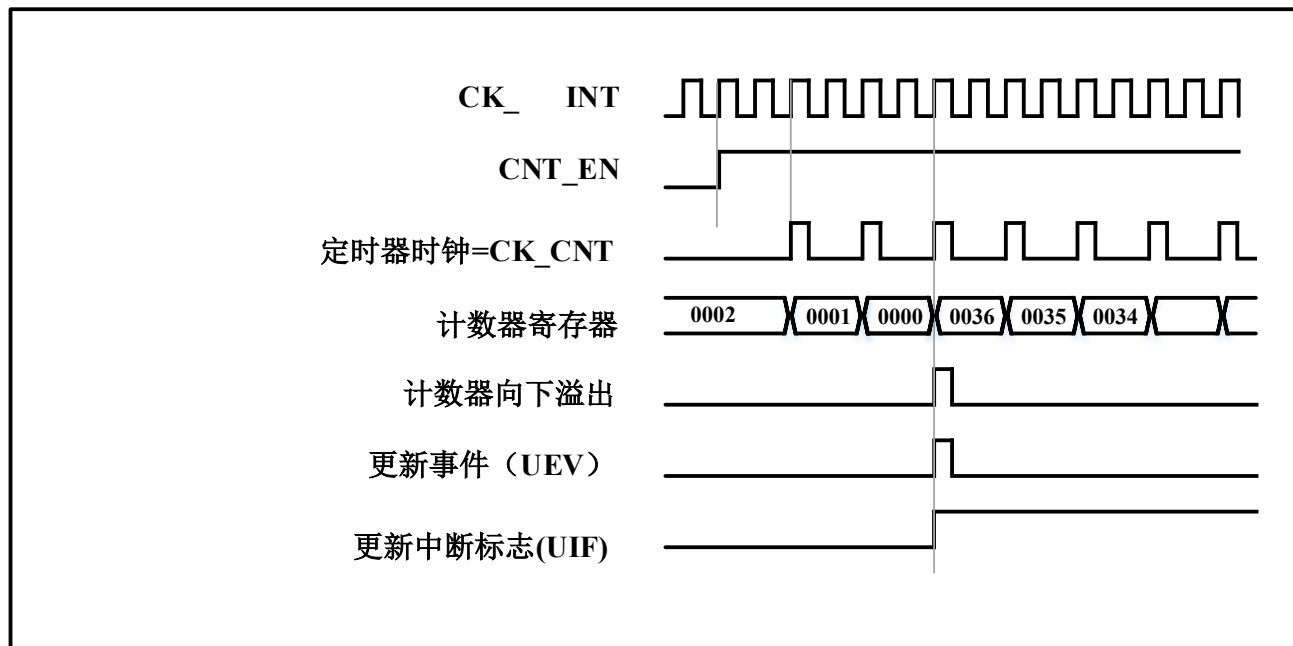


图 14-11. 计数器时序图，内部时钟分频因子为 2

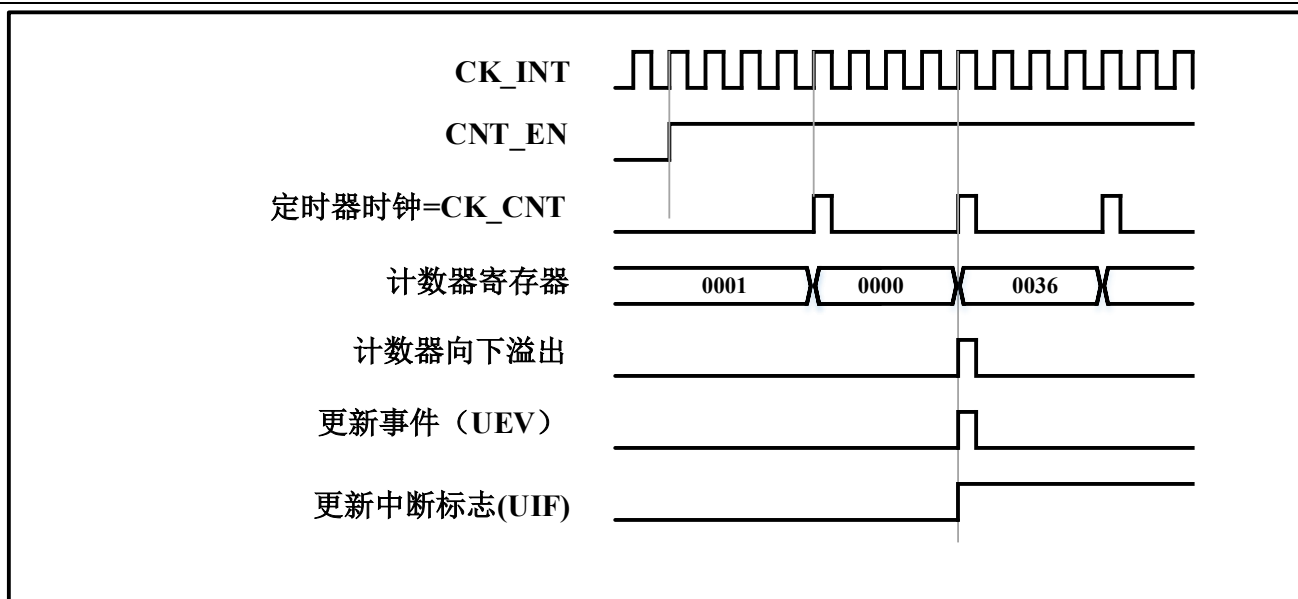


图 14-12. 计数器时序图，内部时钟分频因子为 4

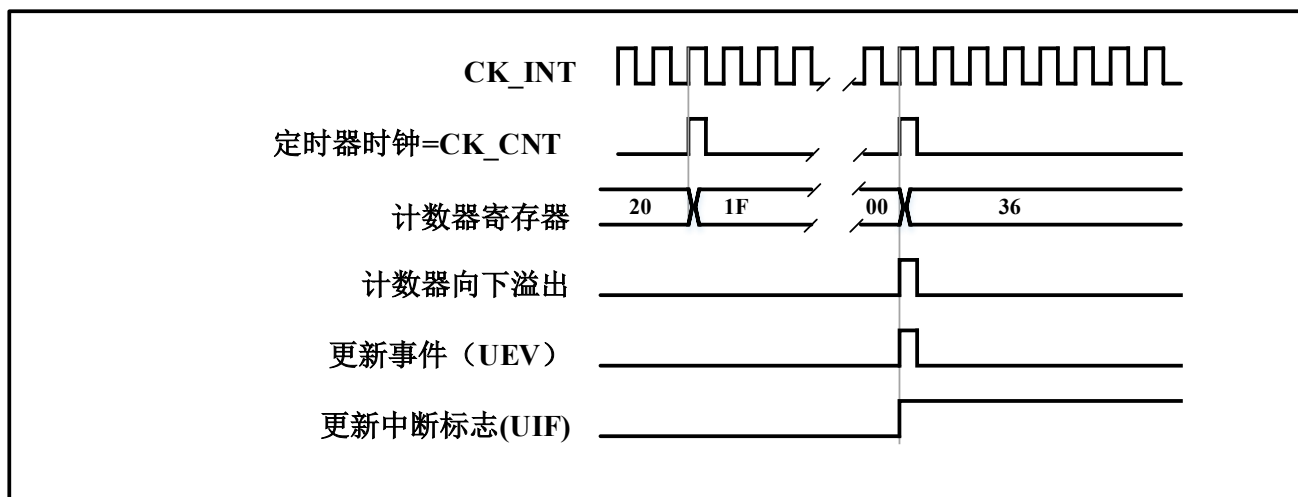


图 14-13. 计数器时序图，内部时钟分频因子为 N

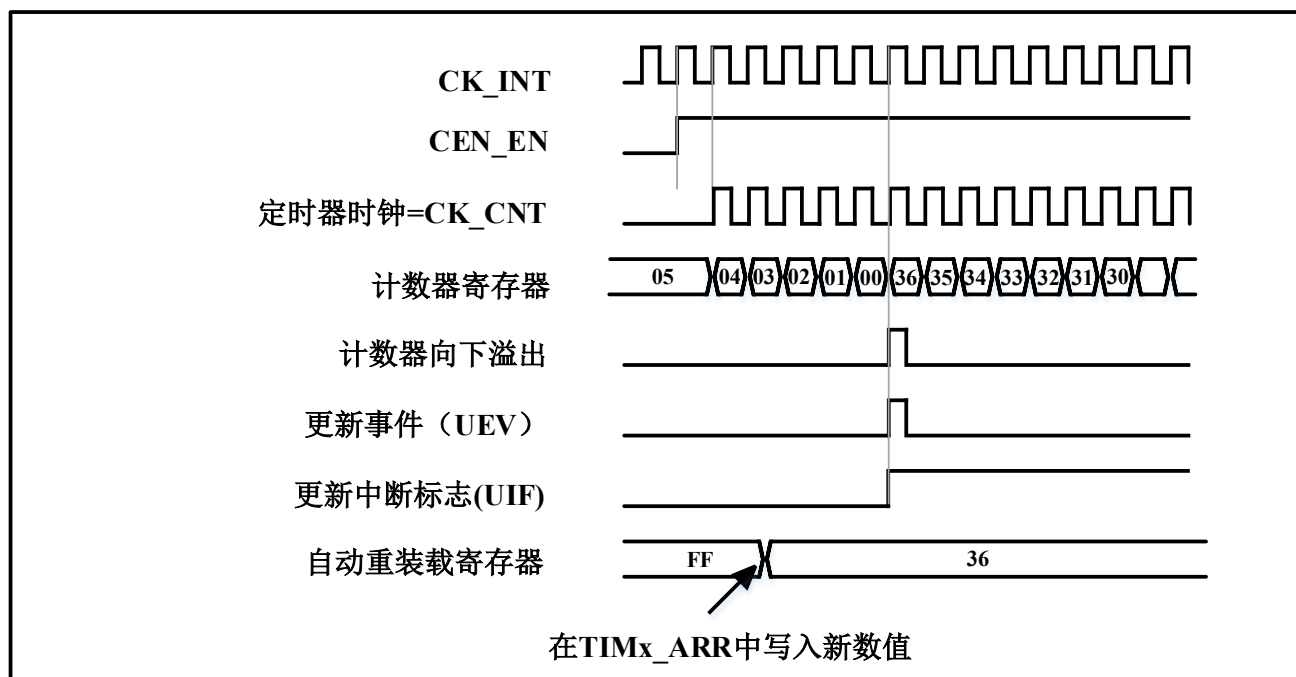


图 14-14. 计数器时序图，当没有使用重复计数器时的更新事件

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(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)，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：

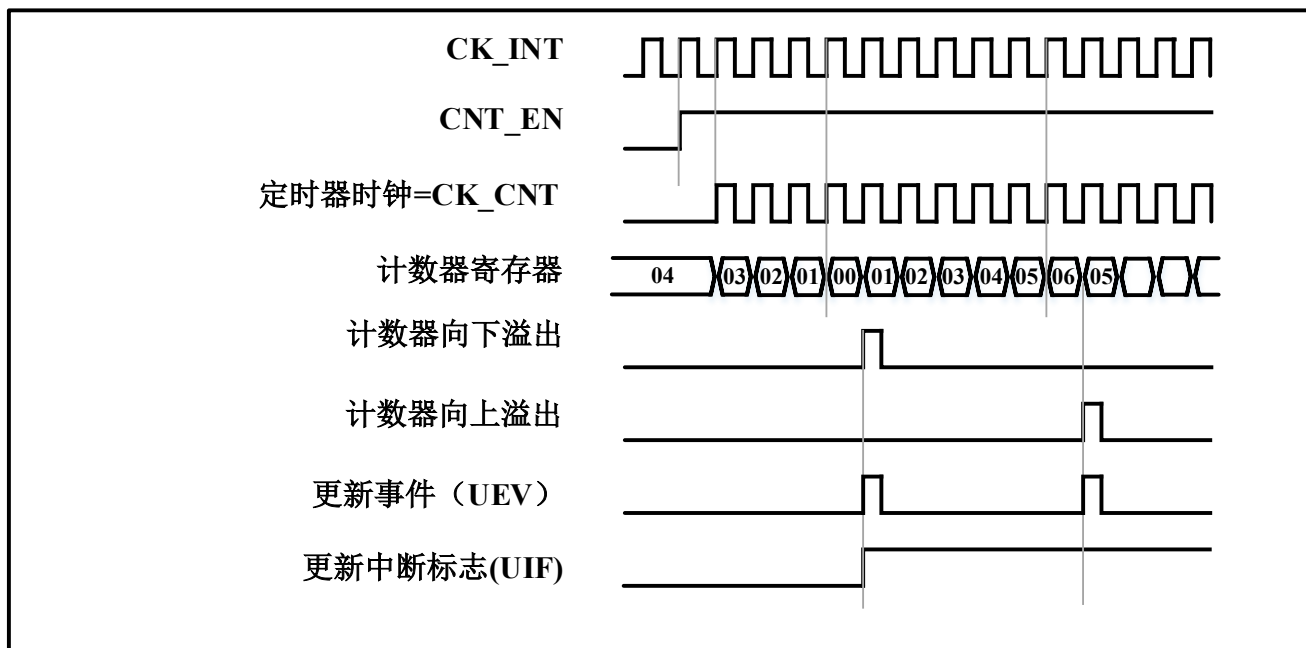


图 14-15. 计数器时序图，内部时钟分频因子为 1，TIM2_ARR=0x6

这里使用了中心对齐模式 1(详见 14.5.1 节)。

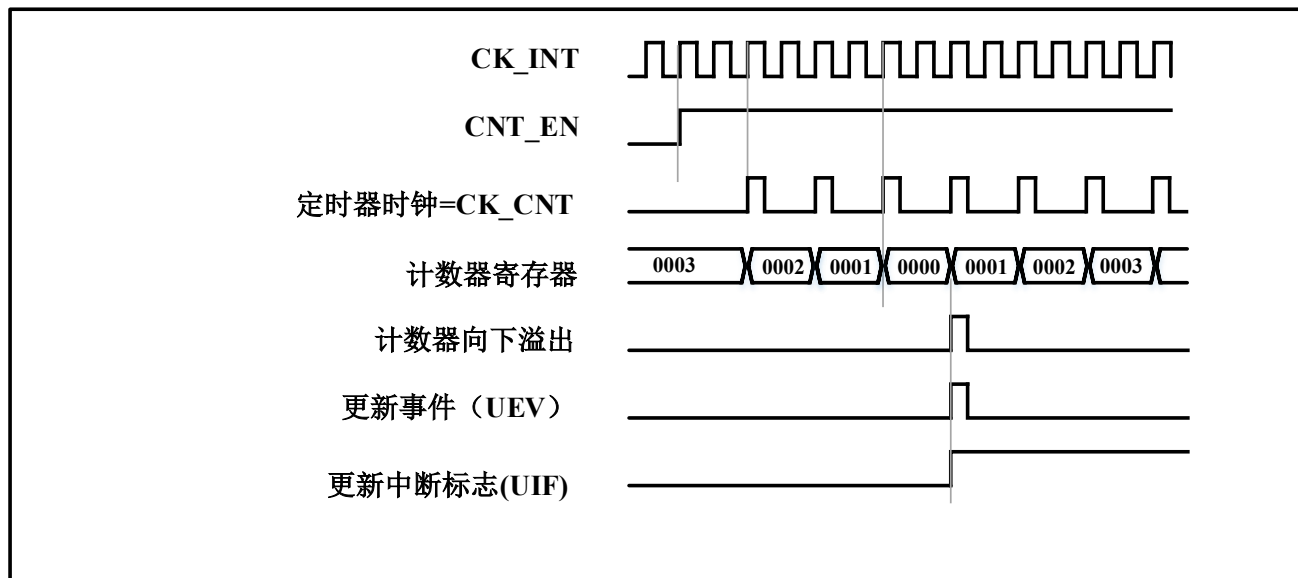


图 14-16. 计数器时序图，内部时钟分频因子为 2

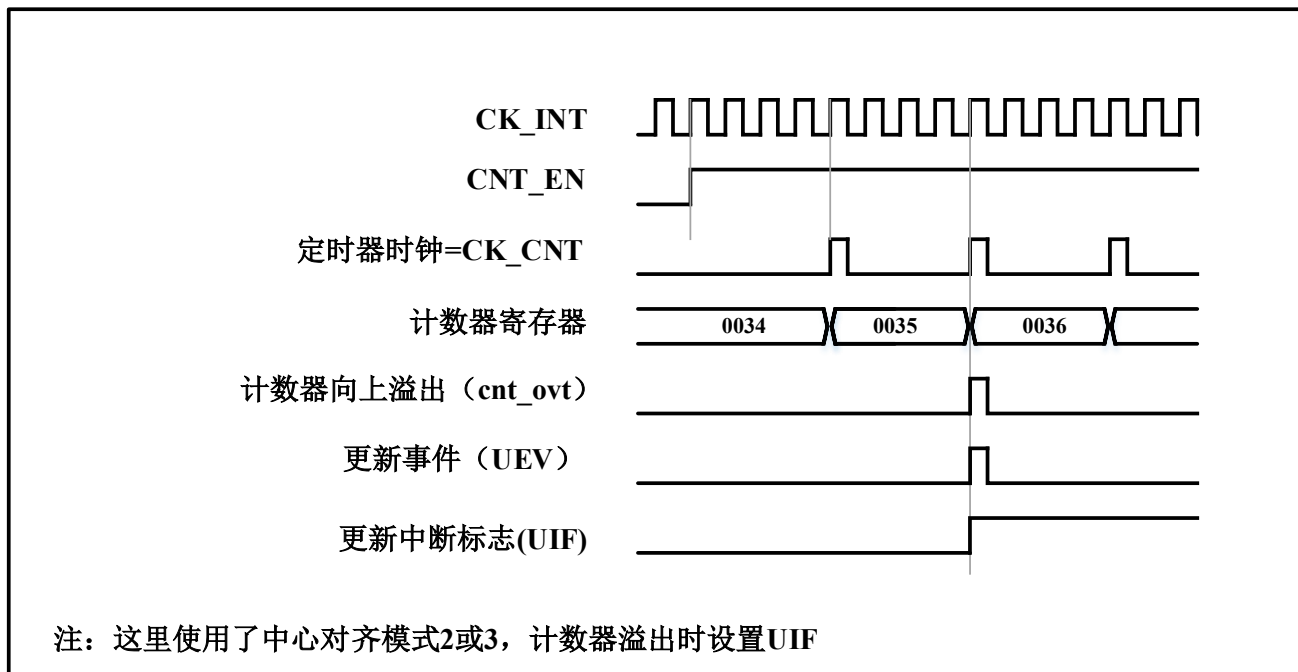


图 14-17. 计数器时序图，内部时钟分频因子为 4，TIM2_ARR=0x36

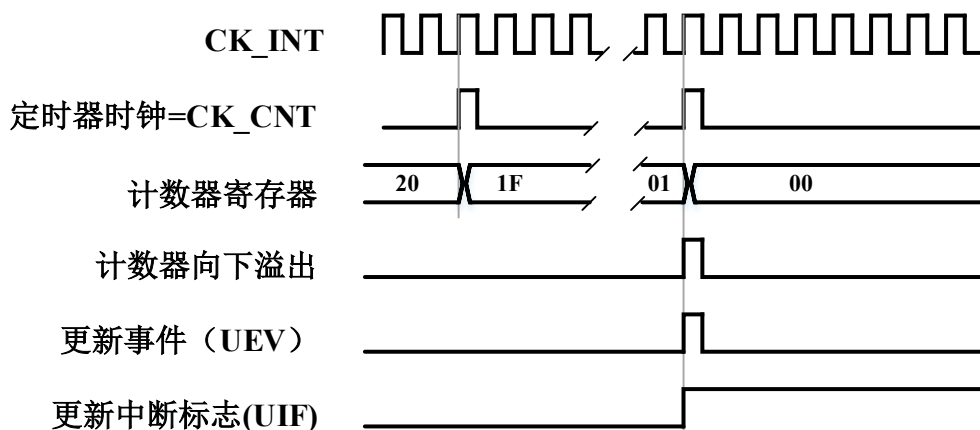


图 14-18. 计数器时序图，内部时钟分频因子为 N

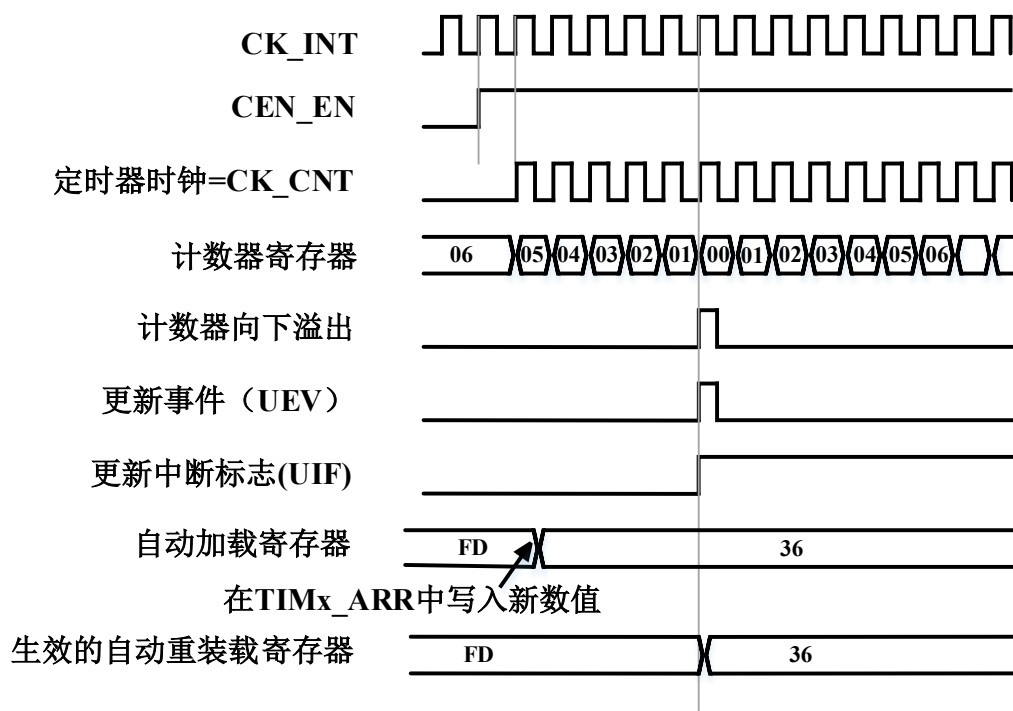


图 14-19. 计数器时序图，ARPE=1 时的更新事件(计数器下溢)

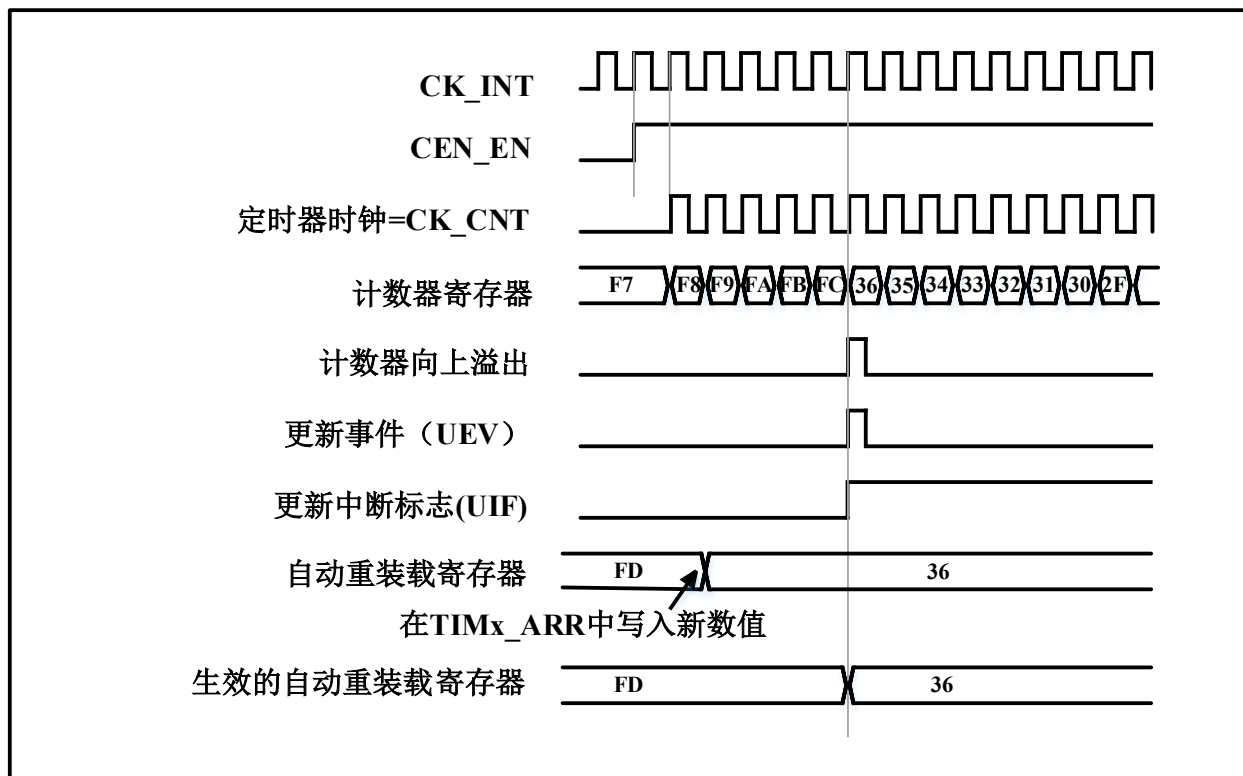


图 14-20. 计数器时序图，ARPE=1 时的更新事件(计数器溢出)

14.3.3 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟(CK_INT)
- 外部时钟模式 1：外部输入脚(TIx)
- 外部时钟模式 2：外部触发输入(ETR)
- 内部触发输入(ITRx)：使用一个定时器作为另一个定时器的预分频器，如可以配置一个定时器 Timer1 而作为另一个定时器 Timer2 的预分频器。

14.3.3.1 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。

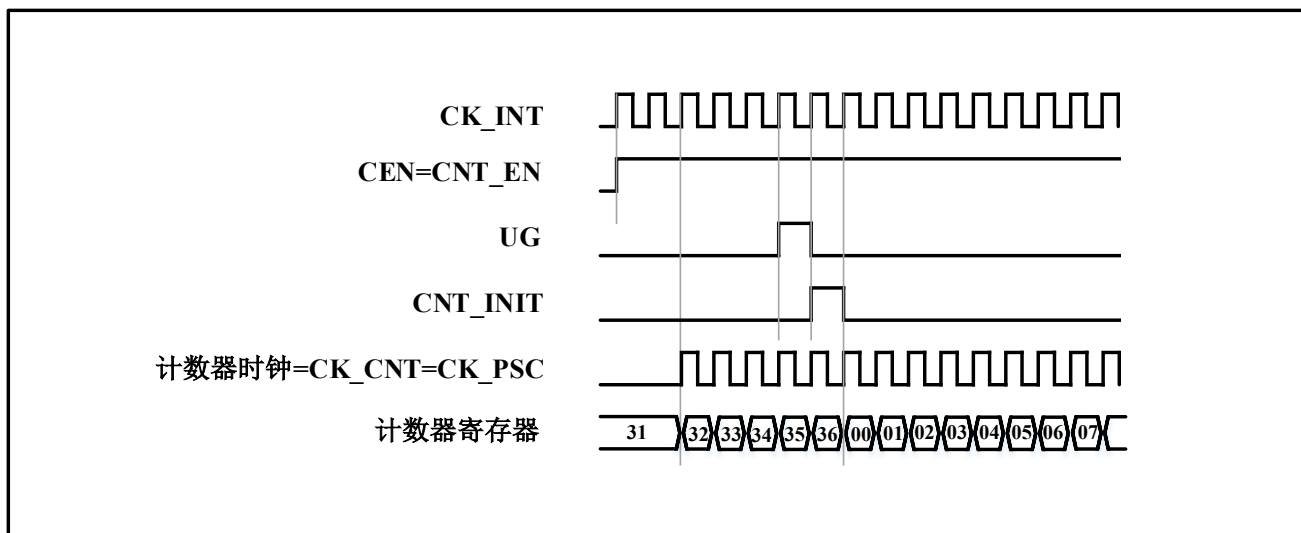


图 14-21. 一般模式下的控制电路，内部时钟分频因子为 1

14.3.3.2 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。

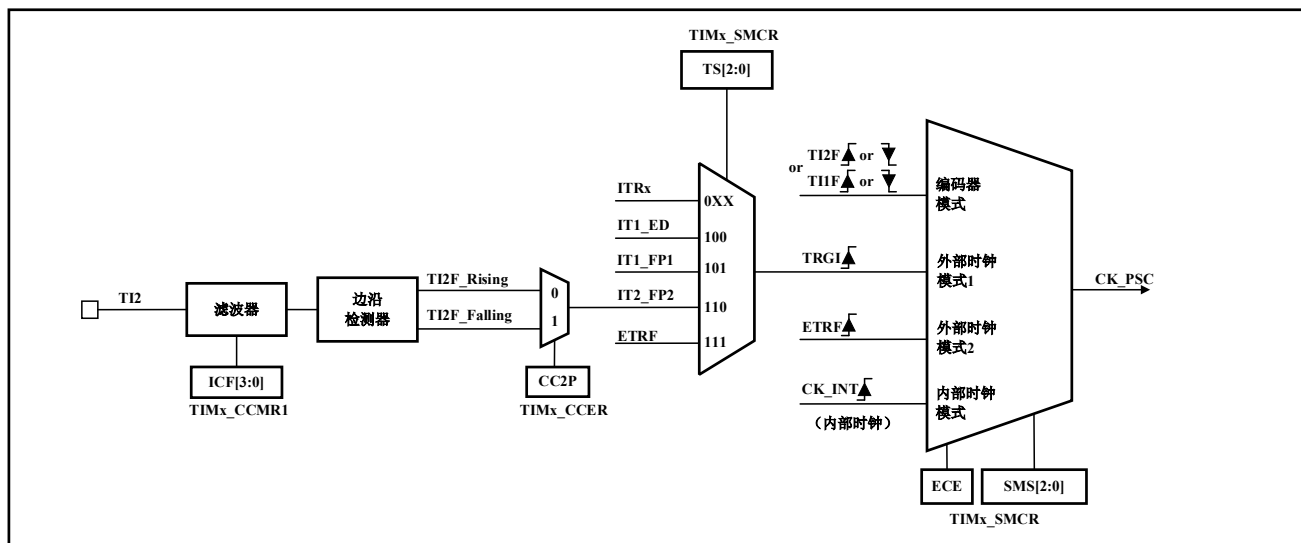


图 14-22. TI2 外部时钟连接例子

例如，要配置向上计数器在 TI2 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 TI2 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 TI2 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。

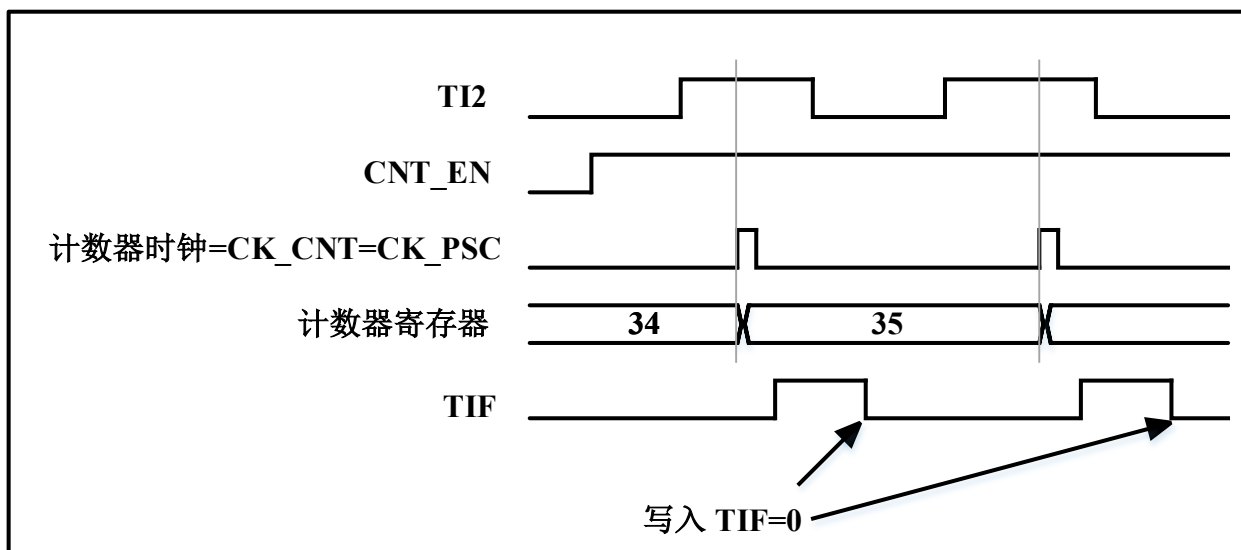


图 14-23. 外部时钟模式 1 下的控制电路

14.3.3.3 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：

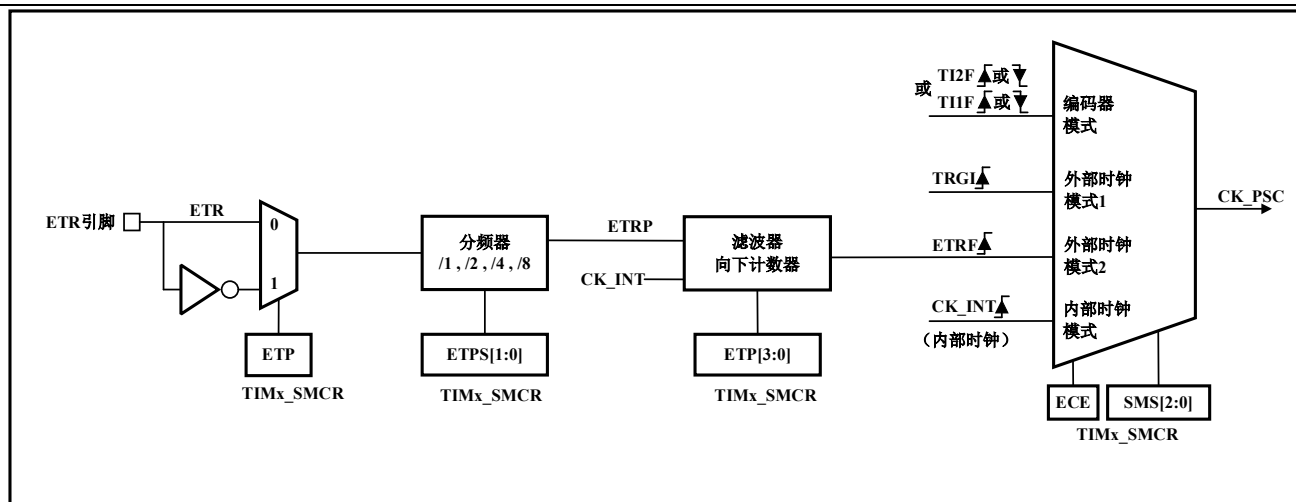


图 14-24. 外部触发输入框图

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

- 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
- 6)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。

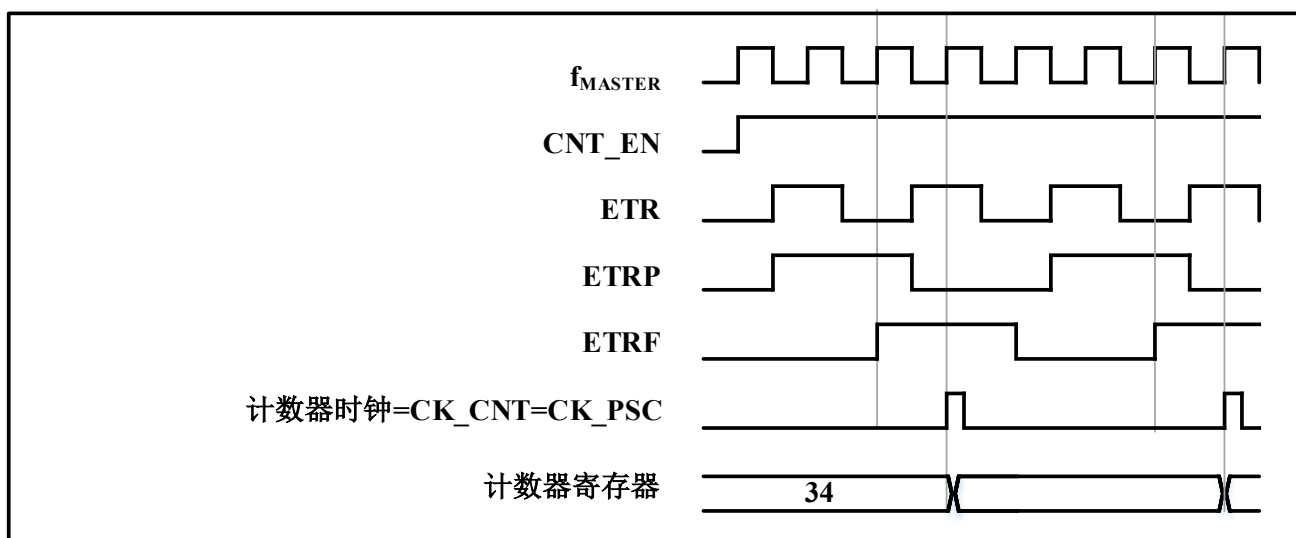


图 14-25. 外部时钟模式 2 下的控制电路

14.3.4 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($ICxPS$)。

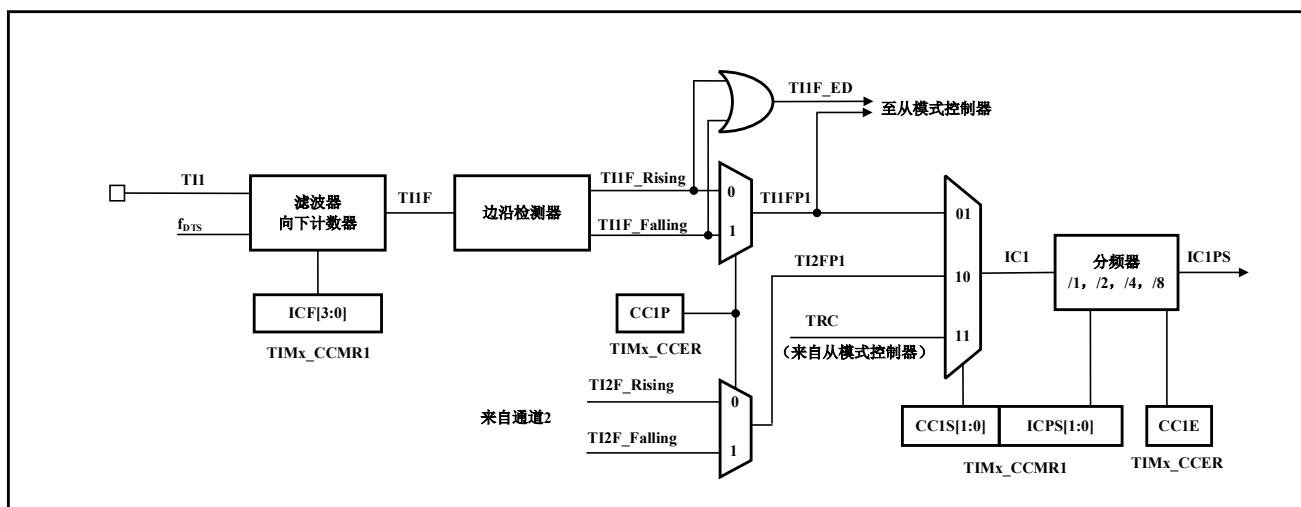


图 14-26. 捕获/比较通道(如: 通道 1 输入部分)

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

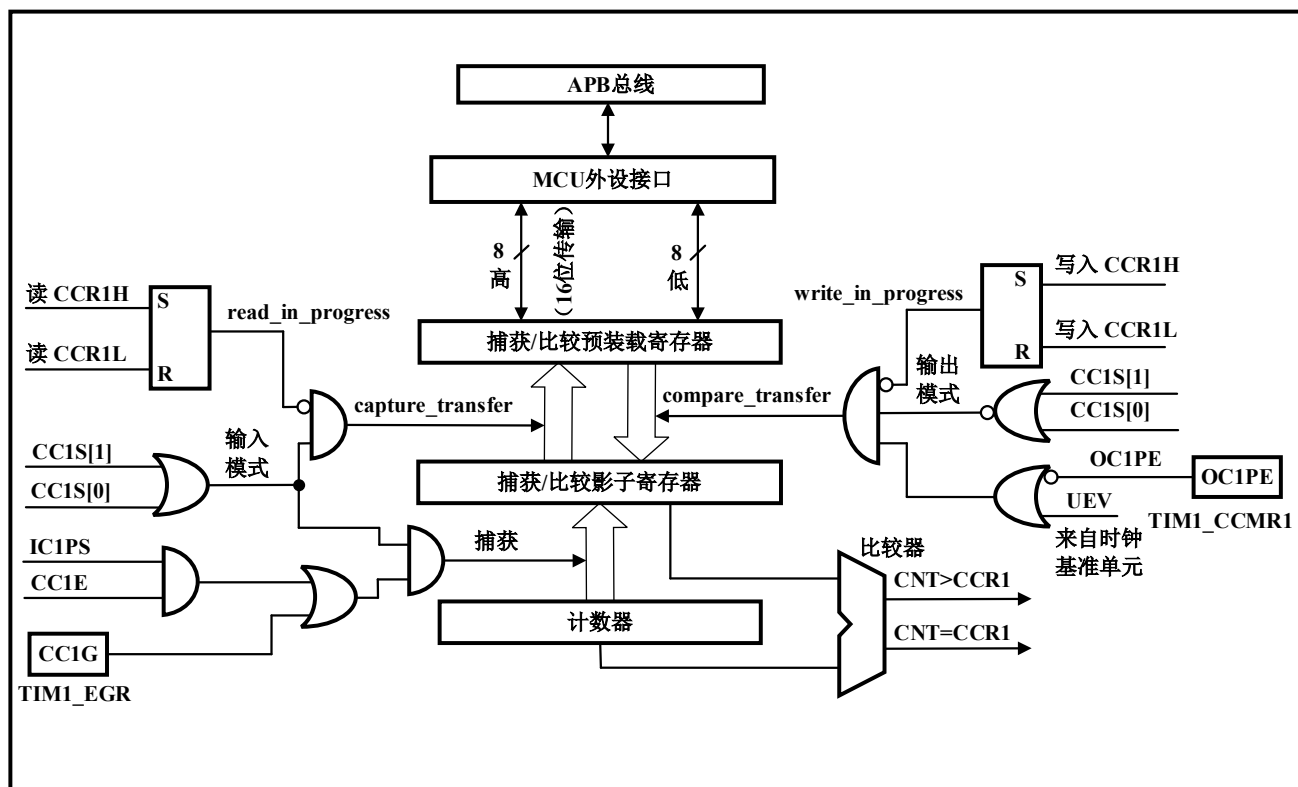


图 14-27. 捕获/比较通道 1 的主电路

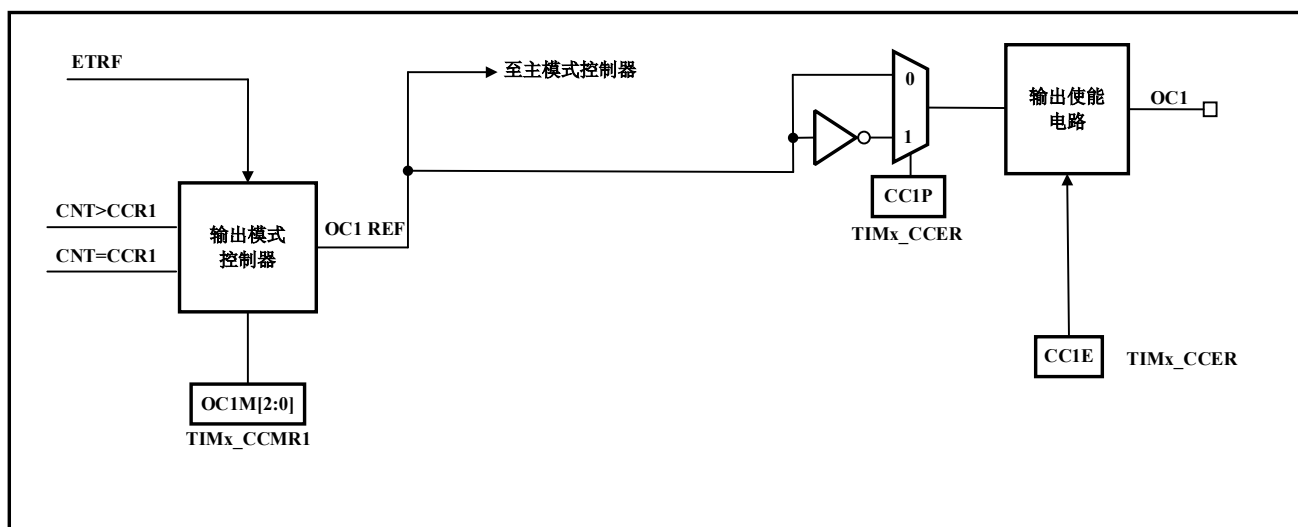


图 14-28. 捕获/比较通道的输出部分（通道 1）

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。

14.3.5 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，则将产生中断。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 TIM2_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 ICxF 位)。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期。因此我们可以(以 f_{DTS} 频率)连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 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。

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，

CC10F 也被置' 1'。

- 如设置了 CC1IE 位，则会产生一个中断。

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。

14.3.6 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 TIx 输入。
- 这 2 个 ICx 信号为边沿有效，但是极性相反。
- 其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。

例如，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。

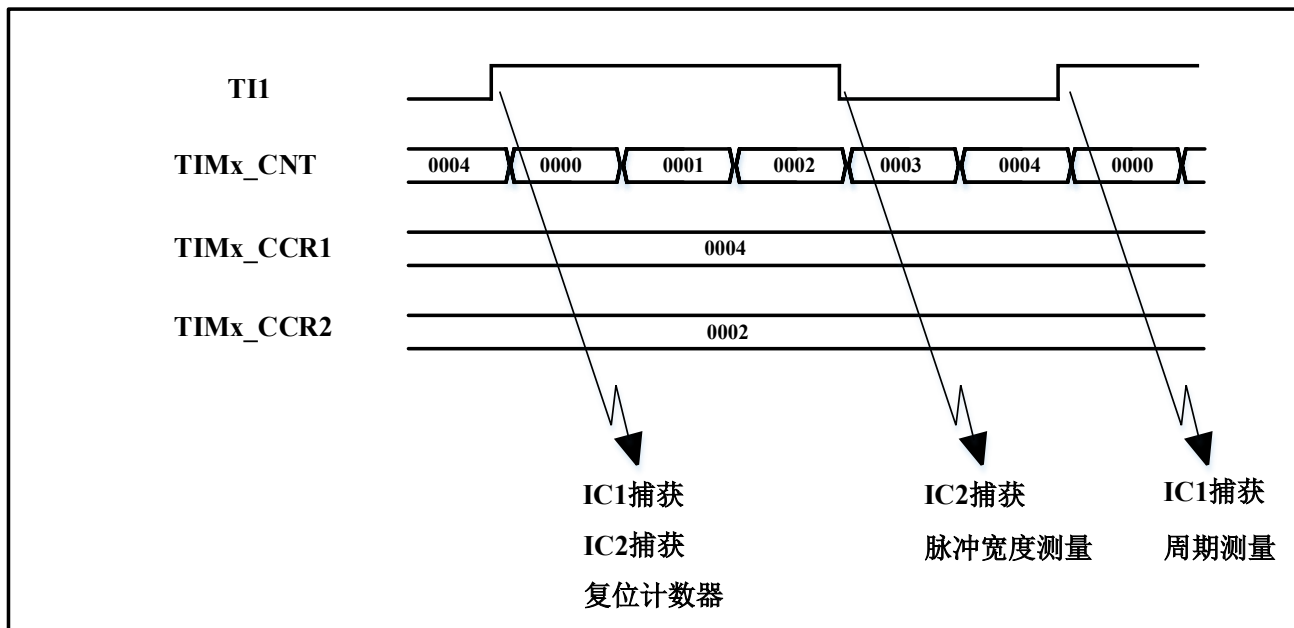


图 14-29. PWM 输入模式时序

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。

14.3.7 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。这将会在下方的输出比较模式一节中介绍。

14.3.8 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。

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，设置 CCxIE 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。

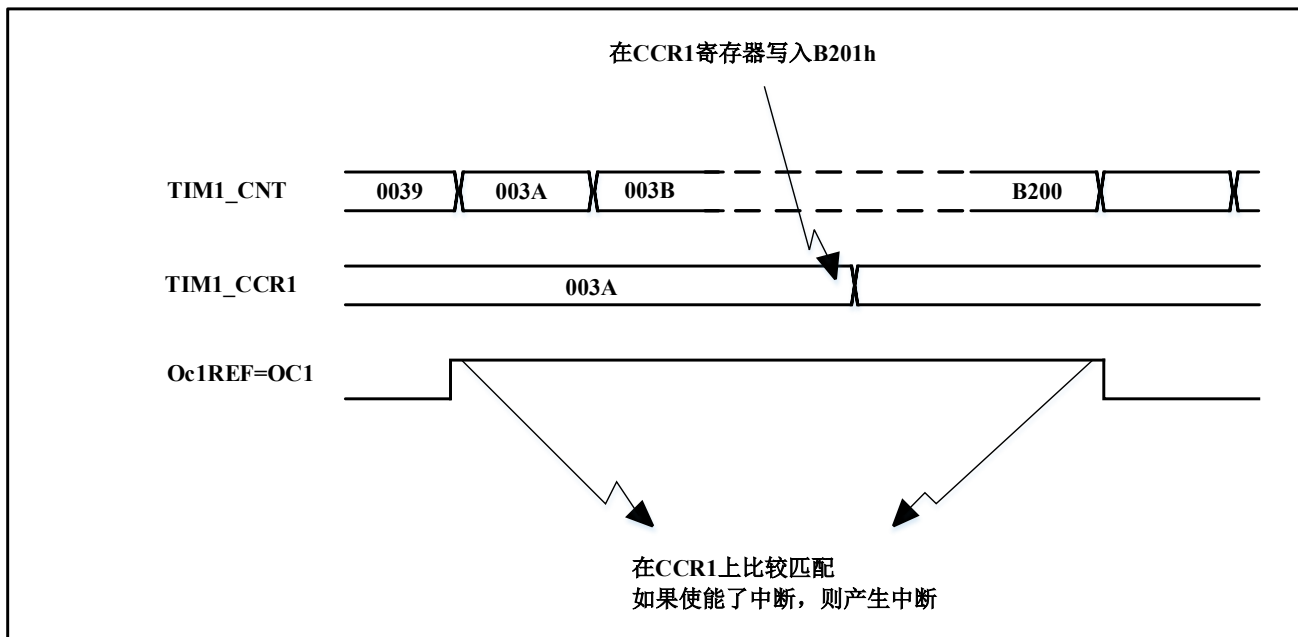


图 14-30. 输出比较模式，翻转 OC1

14.3.9 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。

14.3.9.1 PWM 边沿对齐模式

14.3.9.1.1 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 $TIM2_CNT < TIM2_CCR_x$ 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装载值(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。

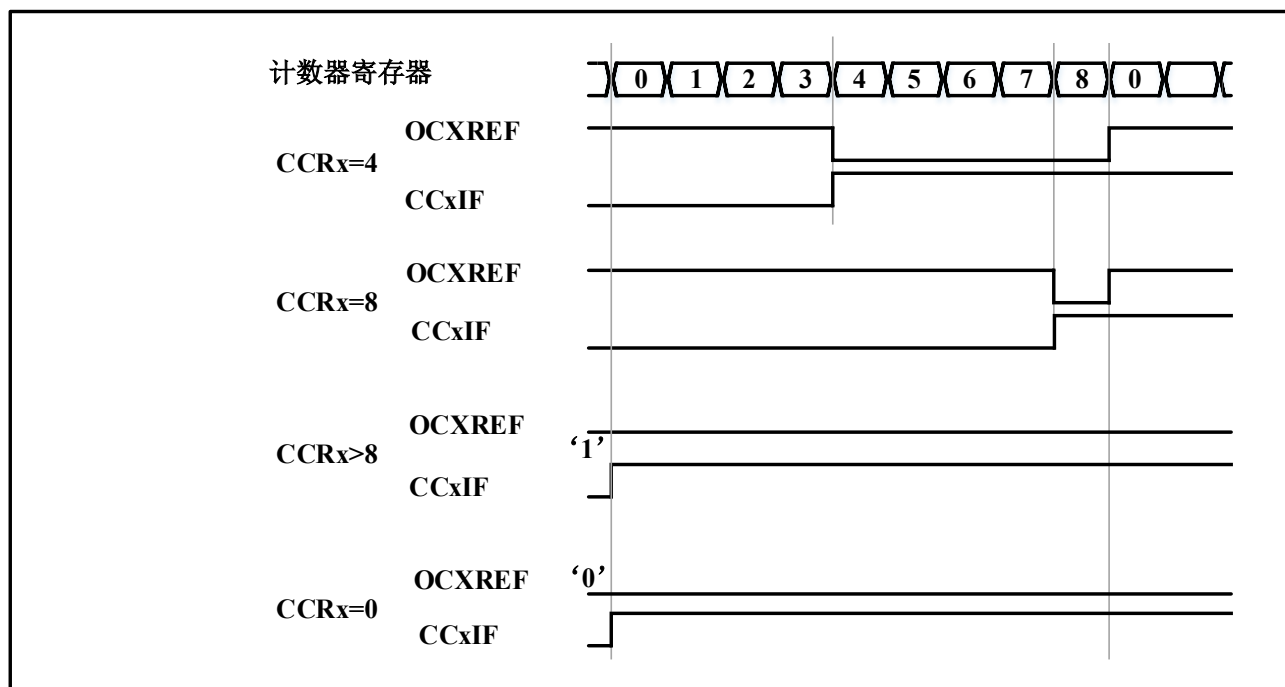


图 14-31. 边沿对齐的 PWM 波形(ARR=8)

14.3.9.1.2 向下计数的配置

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。

14.3.9.2 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。

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。

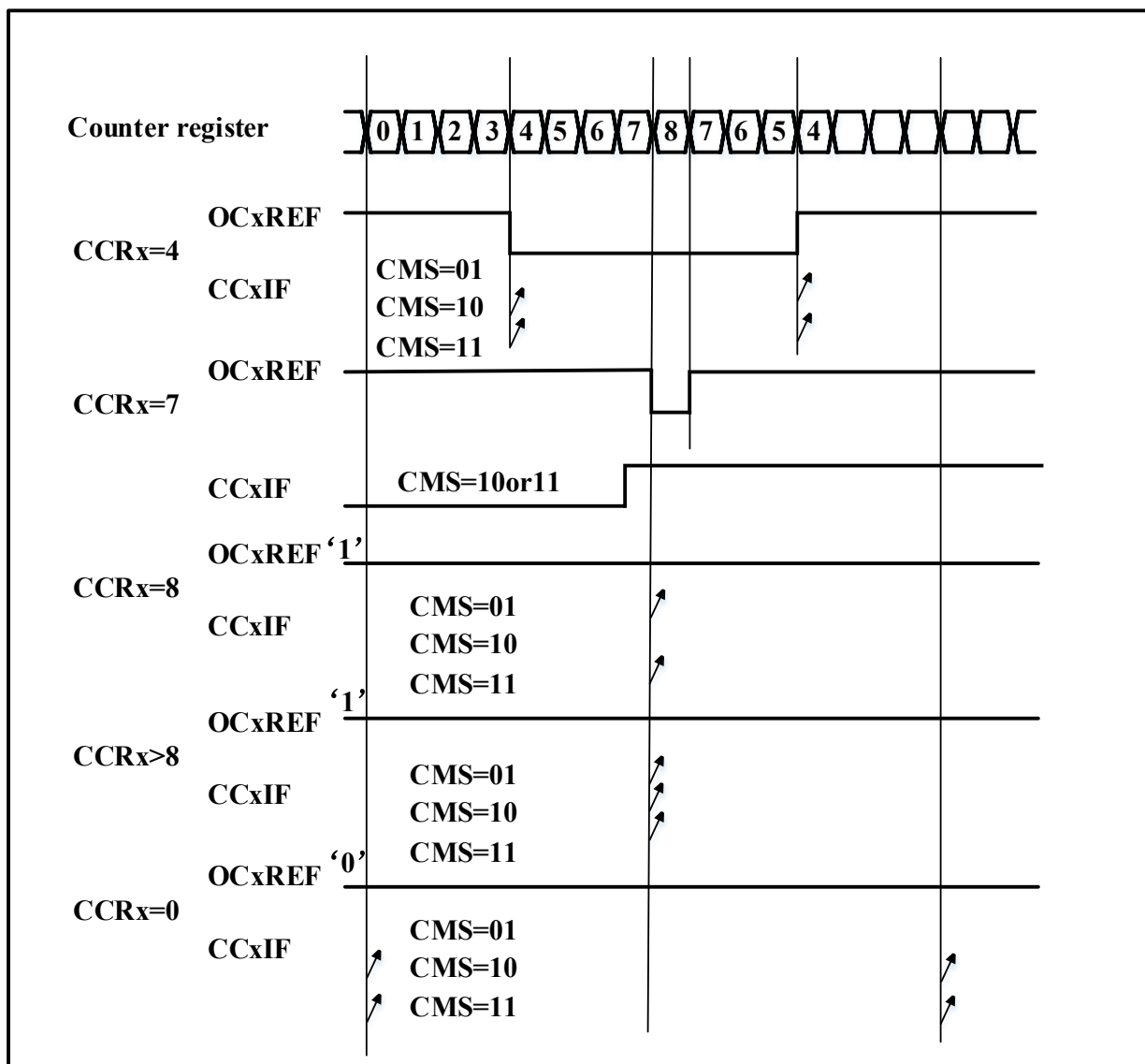


图 14-32. 中央对齐的 PWM 波形(APR=8)

14.3.9.2.1 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；这就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。

14.3.10 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: $CNT < CCRx \leq ARR$ (特别地, $0 < CCRx$), 向下计数方式: $CNT > CCRx$ 。

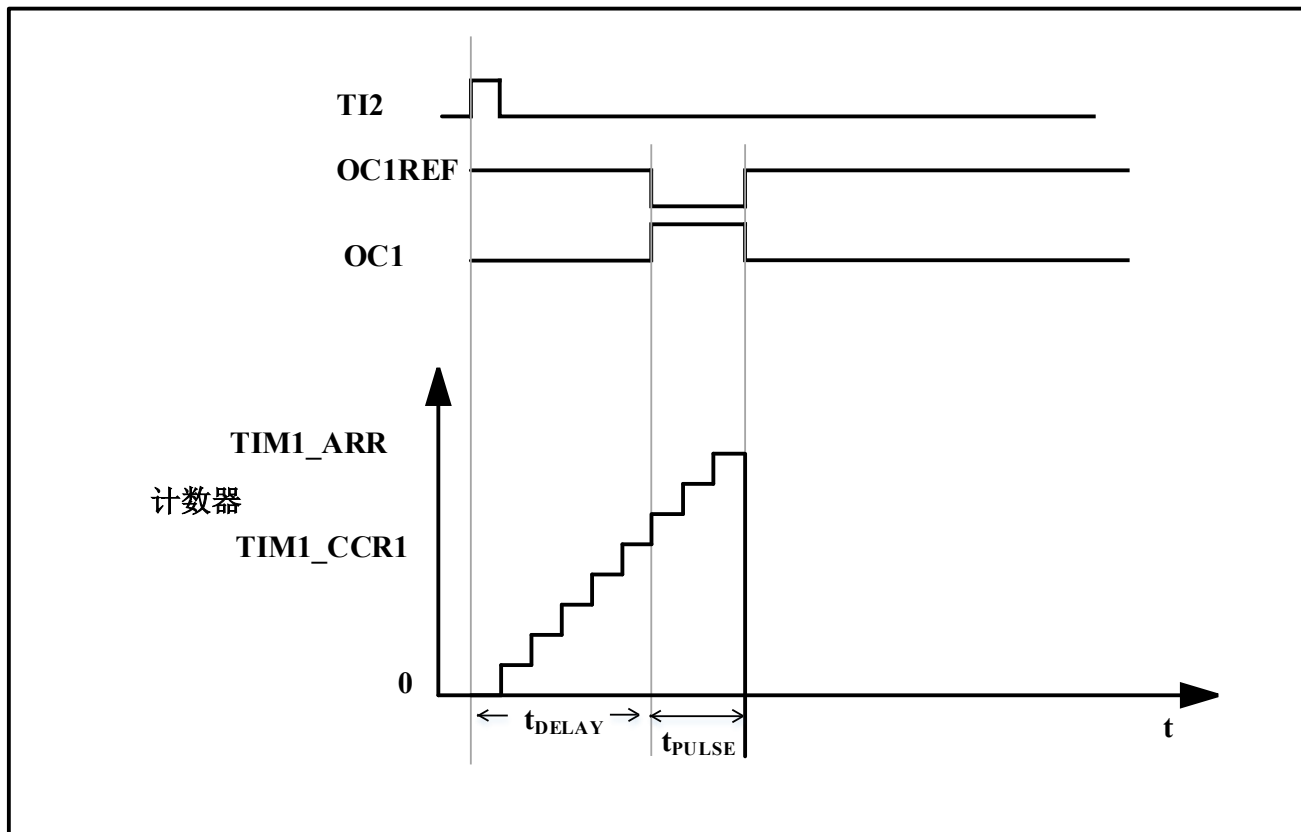


图 14-33. 单脉冲模式的例子

例如，你需要在从 TI2 输入脚上检测到一个上升沿开始，延迟 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(TIM2_ARR-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; 根据需要有选择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。

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。

14.3.11 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。

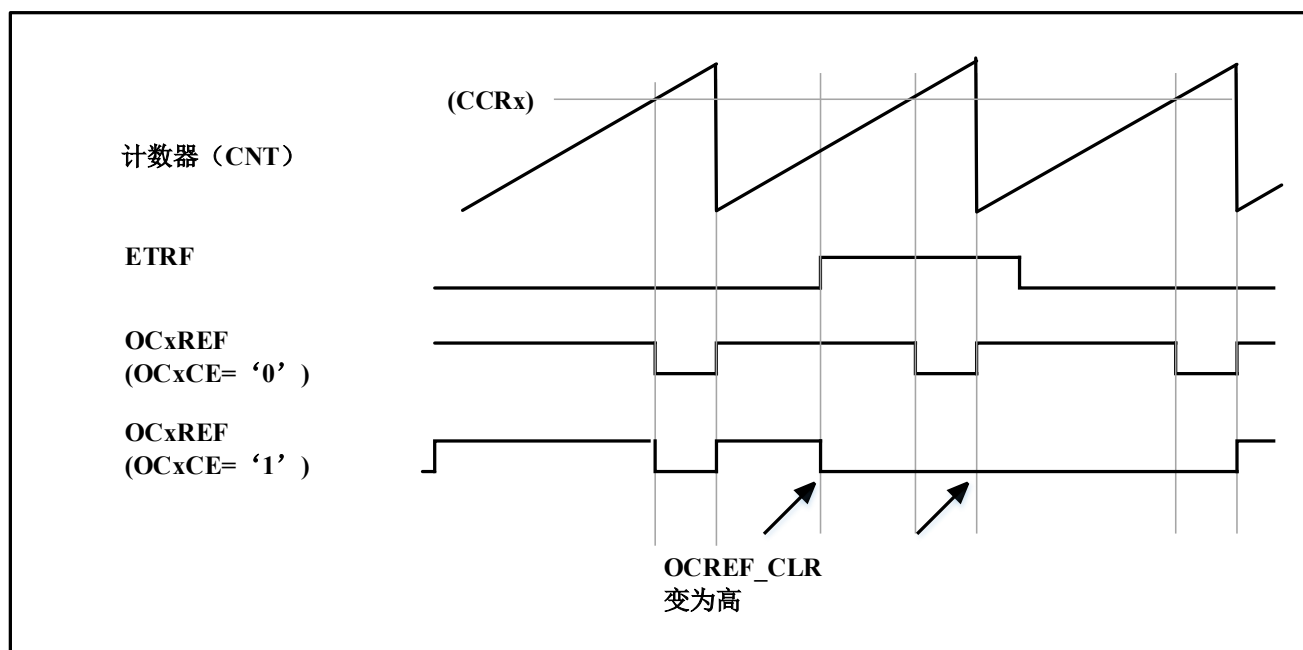


图 14-34. 清除 TIM2 的 OCxREF

14.3.12 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 13-1，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 14.1. 计数方向与编码器信号的关系

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

一个外部的增量编码器可以直接与 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)

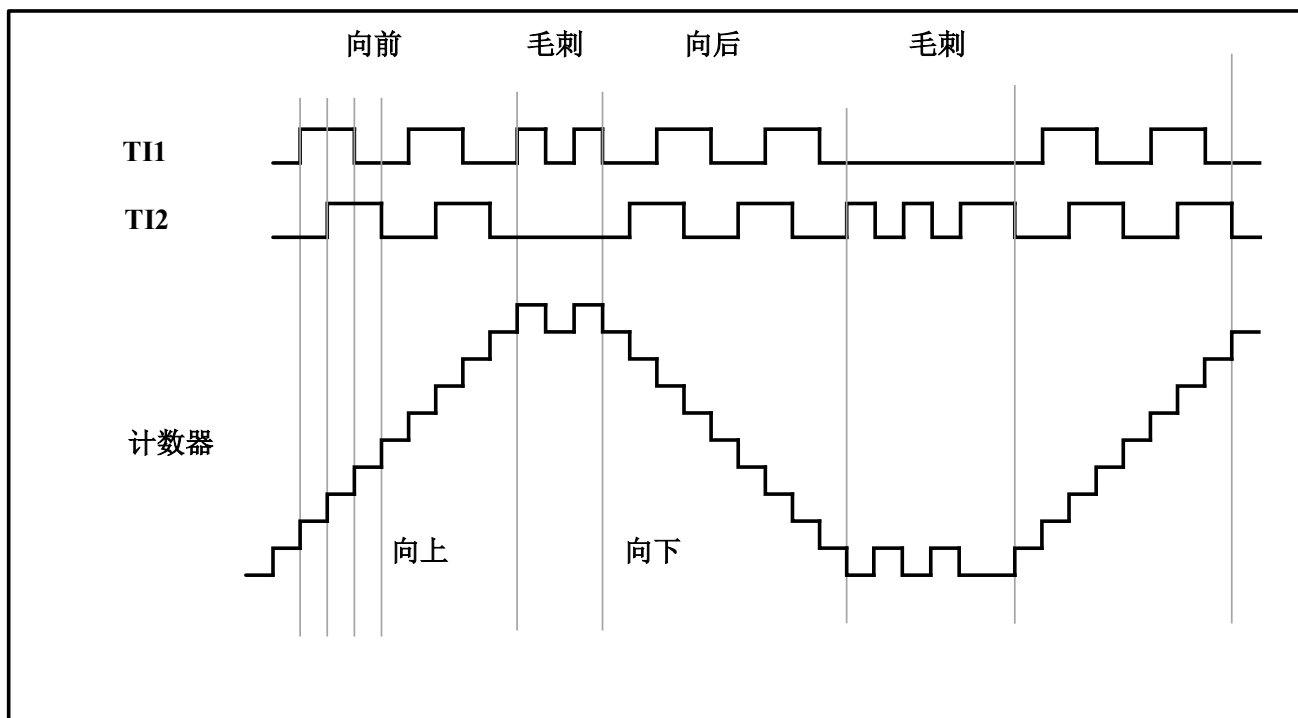


图 14-35. 编码器模式下的计数器操作实例

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

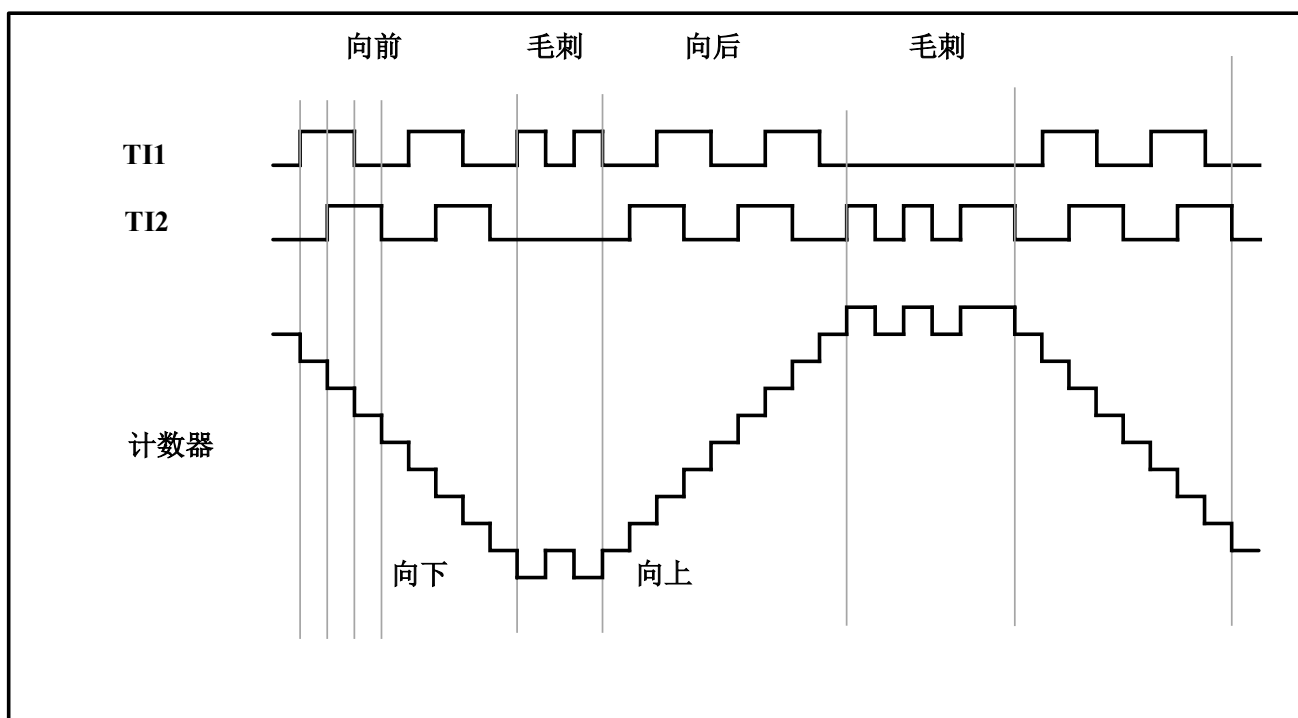


图 14-36. IC1FP1 反相的编码器接口模式实例

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)。

14.3.13 定时器输入异或功能

TIM2_CR2 寄存器中的 TIIS 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。

14.3.14 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。

14.3.14.1 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，产生一个中断请求。

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。

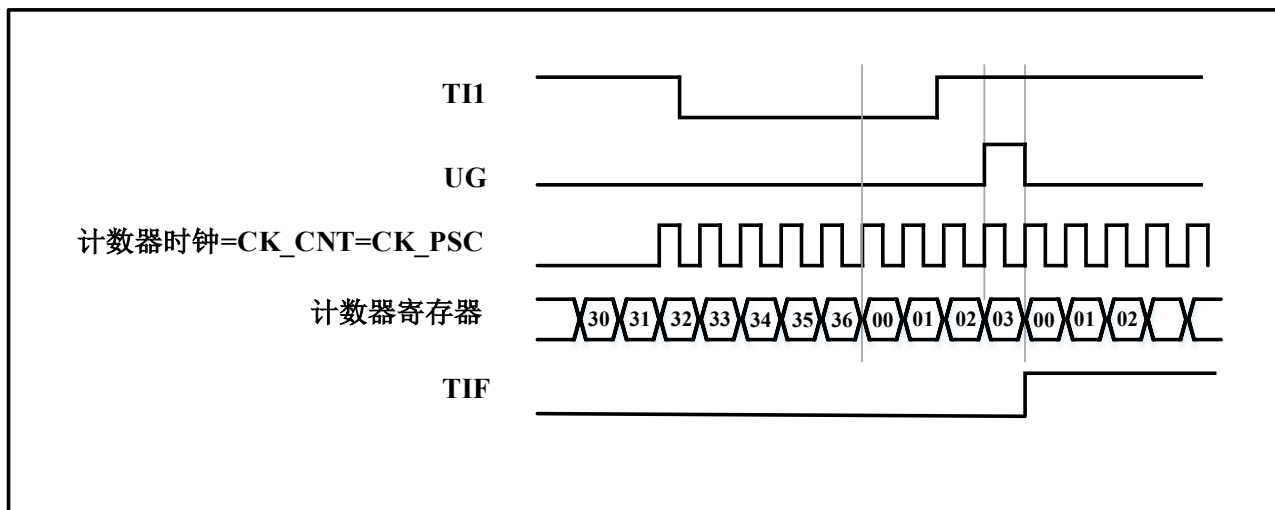


图 14-37. 复位模式下的控制电路

14.3.14.2 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。

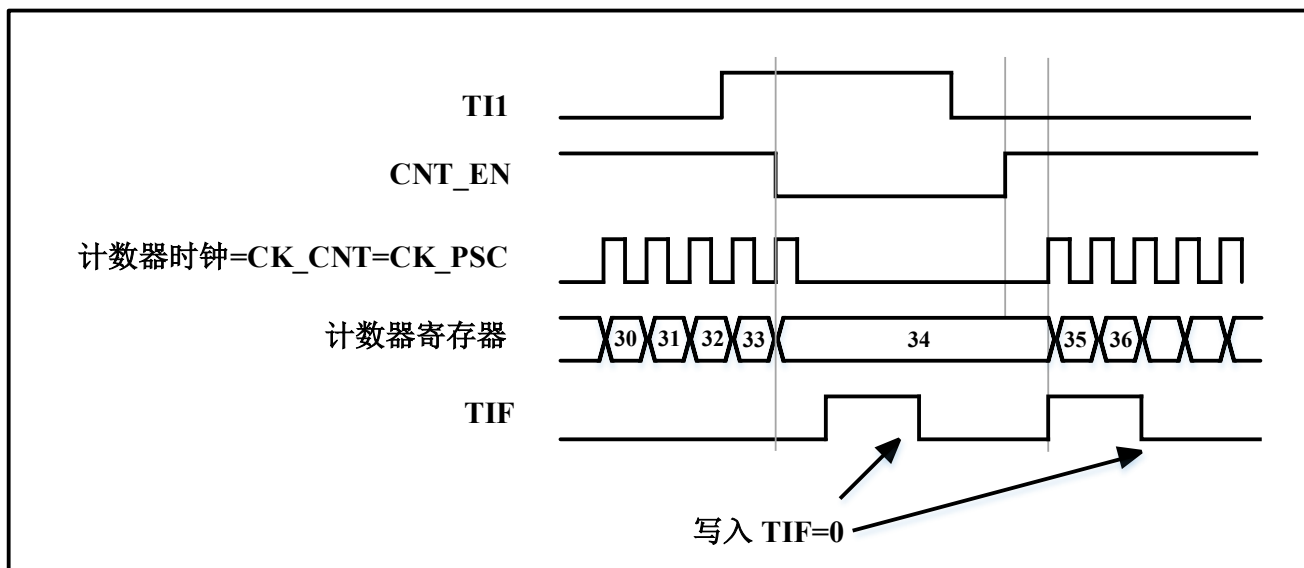


图 14-38. 门控模式下的控制电路

14.3.14.3 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。

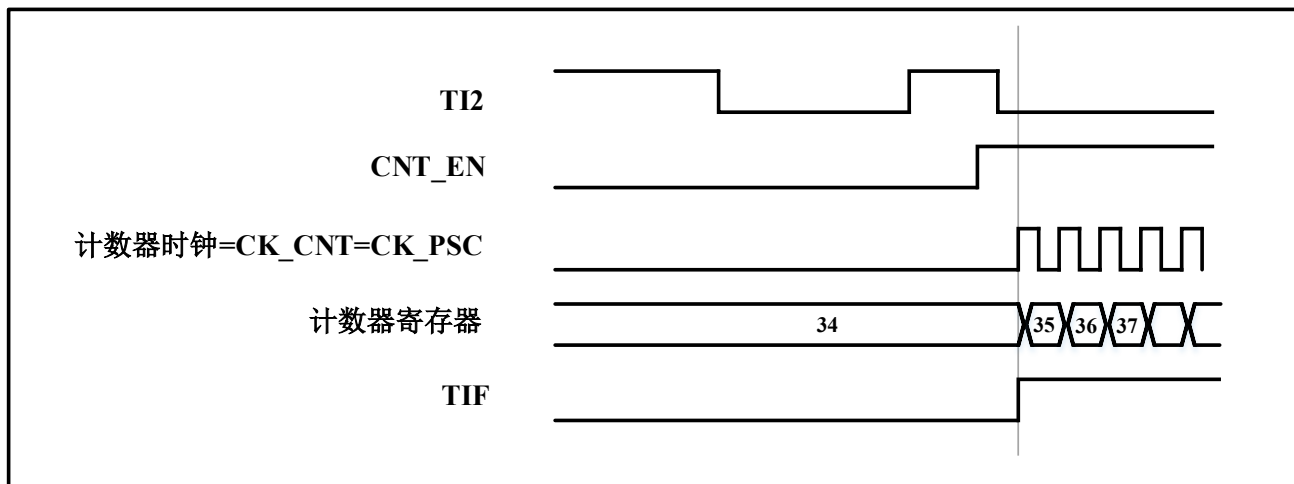


图 14-39. 触发器模式下的控制电路

14.3.14.4 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。

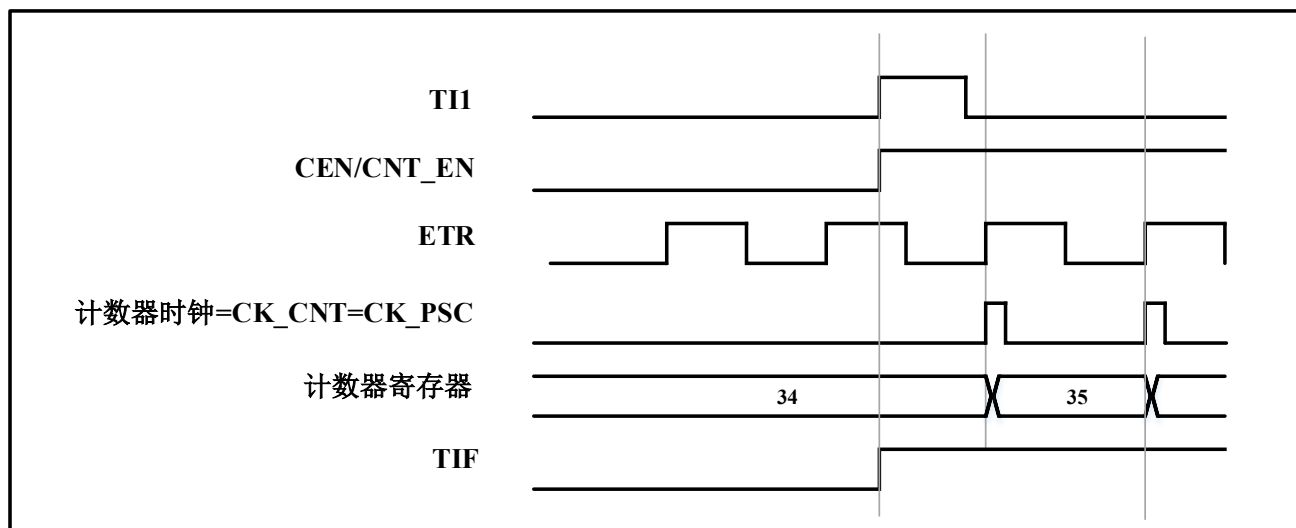


图 14-40. 外部时钟模式 2+触发模式下的控制电路

14.3.15 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。

14.3.15.1 使用一个定时器作为另一个定时器的预分频器

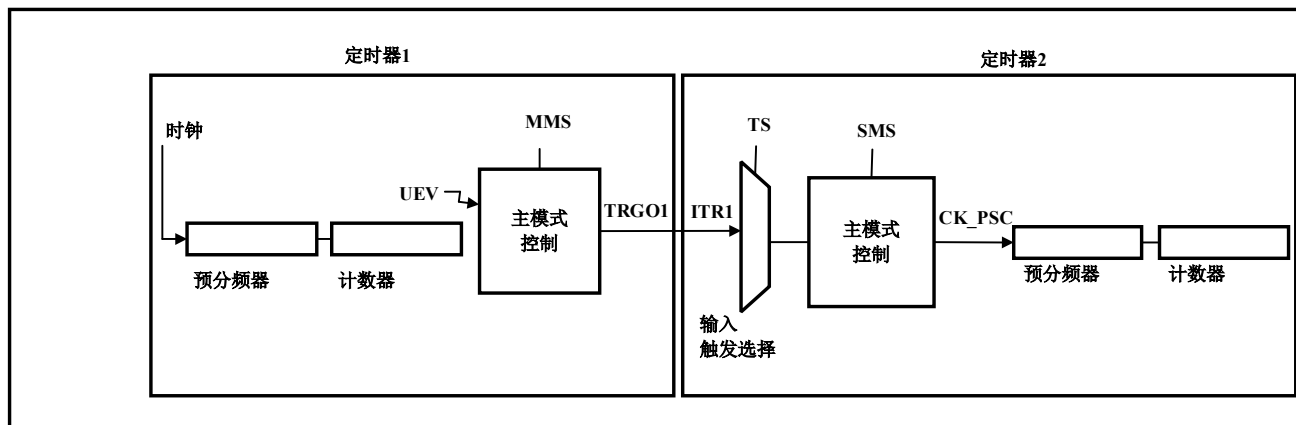


图 14-41. 主/从定时器的例子

如：可以配置定时器 1 作为定时器 2 的预分频器。参考图 14-41，进行下述操作：

- 配置定时器 1 为主模式，它可以在每一个更新事件 UEV 时输出一个周期性的触发信号。在 TIM1_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。

14.3.15.2 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 13-41 的连接。只当定时器 1 的 OC1REF 为高时，定时器 2 才对分频后的内部时钟计数。两个定时器的时钟频率都是由预分频器对 CK_INT 除以 3($f_{CK_CNT}=f_{CK_INT}/3$)得到。

- 配置定时器 1 为主模式，送出它的输出比较参考信号(OC1REF)为触发输出(TIM1_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。

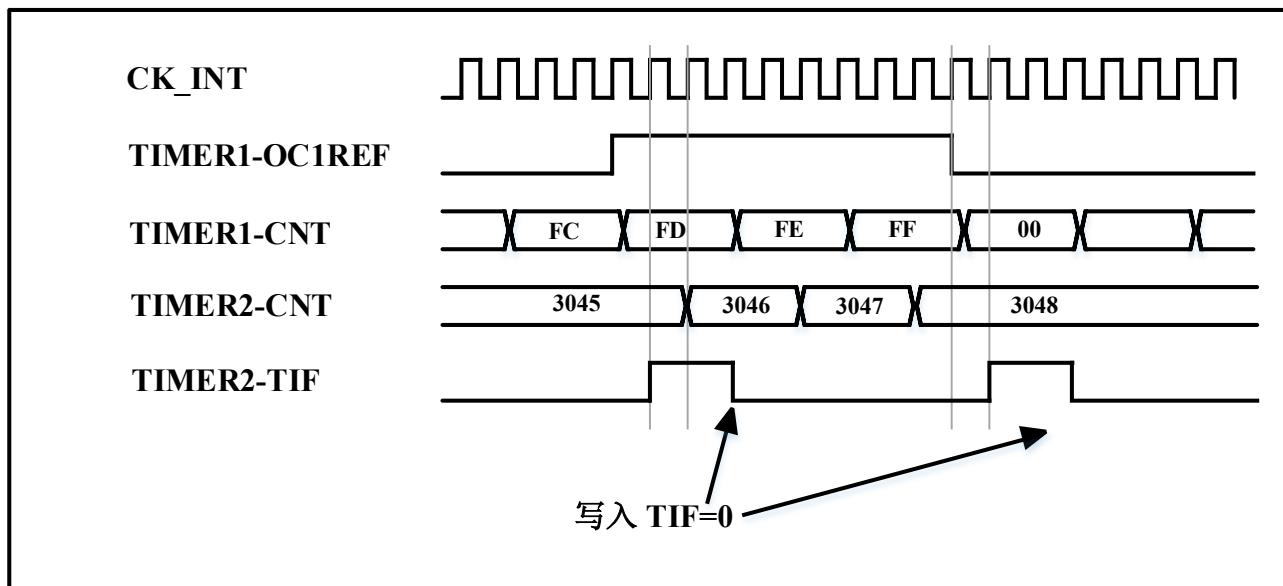


图 14-42. 定时器 1 的 OC1REF 控制定时器 2

在图 14-42 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_CNTL)，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。

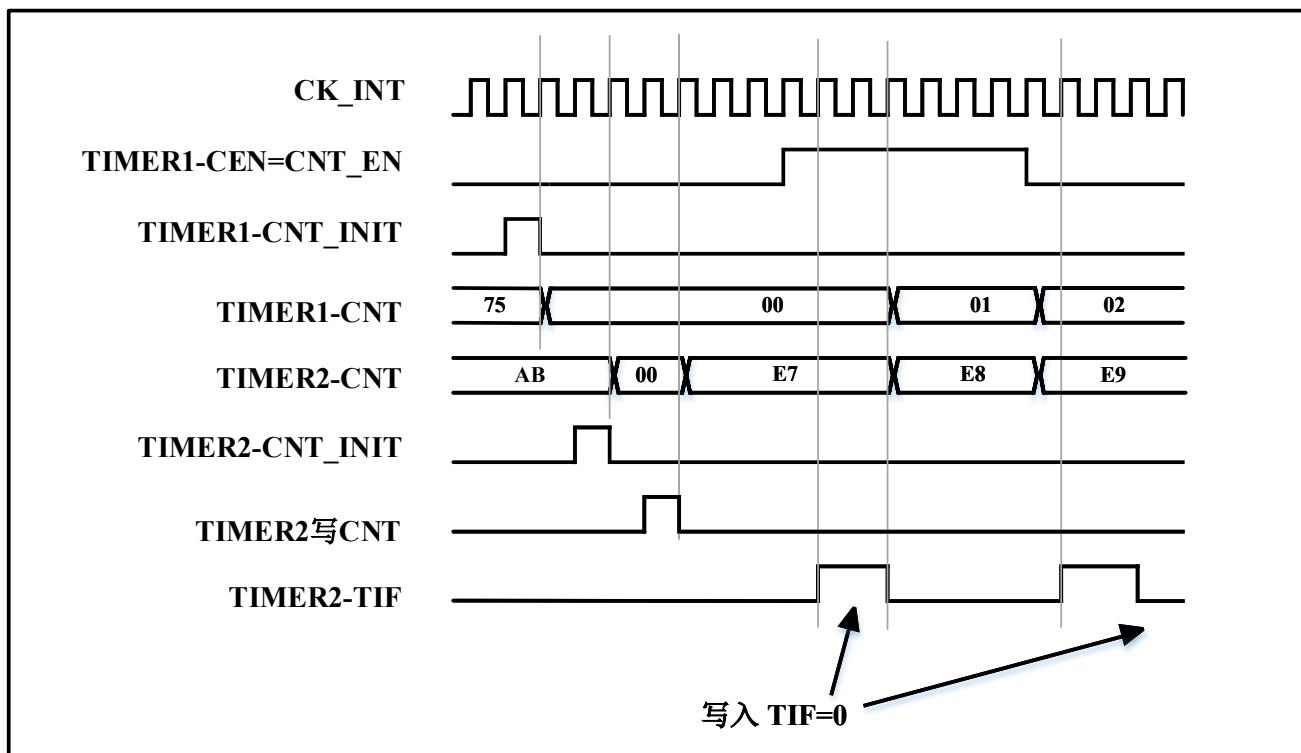


图 14-43. 通过使能定时器 1 可以控制定时器 2

14.3.15.3 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 14-41 的连接。一旦定时器 1 产生更新事件，定时器 2 即从它当前的数值(可以是非 0)按照分频的内部时钟开始计数。

在收到触发信号时，定时器 2 的 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。

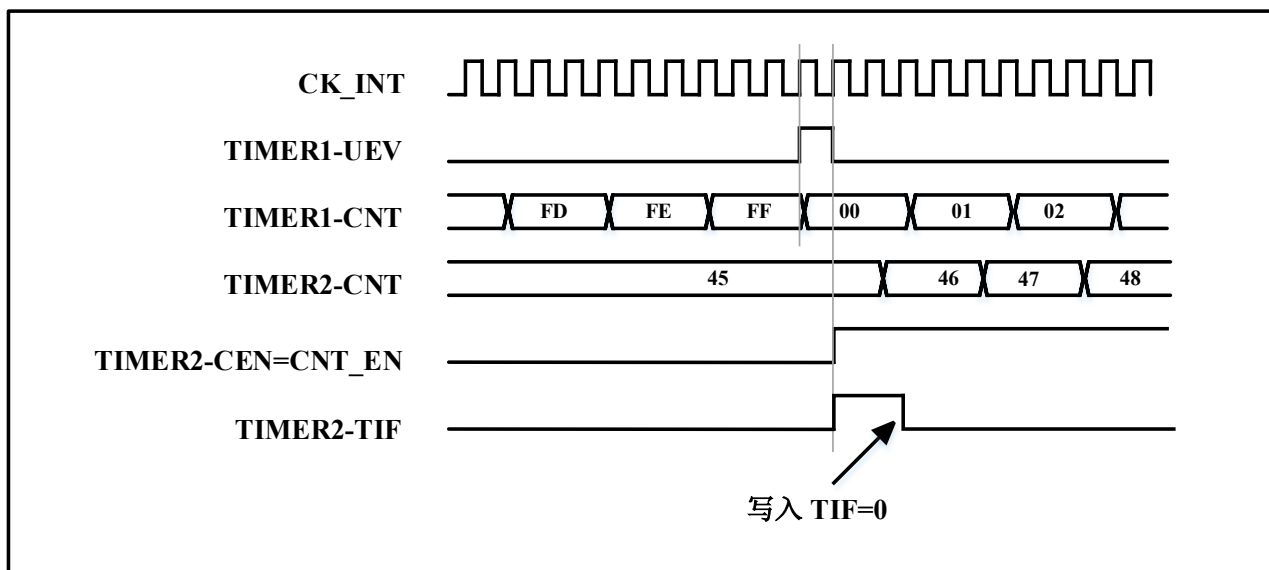


图 14-44. 使用定时器 1 的更新触发定时器 2

在上一个例子中，可以在启动计数之前初始化两个计数器。图 14-45 显示在相同配置情况下，使用触发模式而不是门控模式(TIM2_SMCR 寄存器的 SMS=110)的动作。

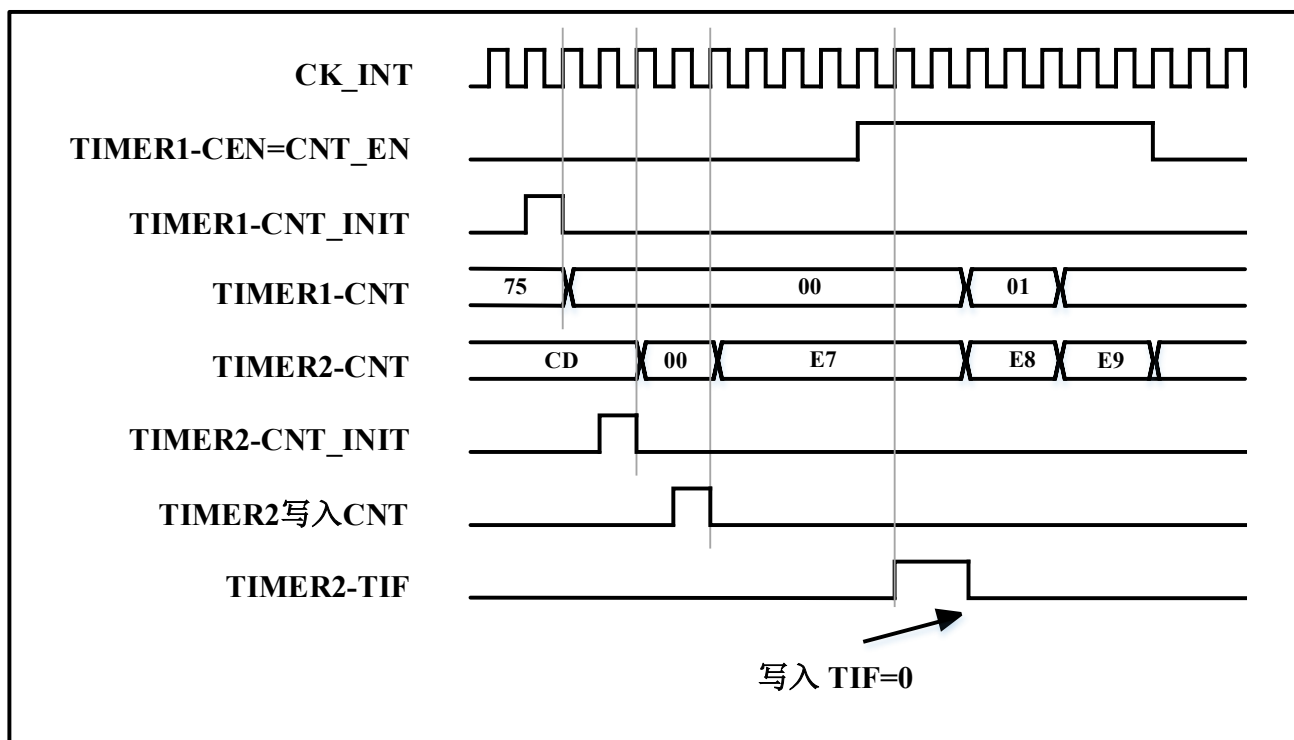


图 14-45. 利用定时器 1 的使能触发定时器 2

14.3.15.4 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 14-41 的连接，配置如下：

- 配置定时器 1 为主模式，送出它的更新事件 UEV 做为触发输出(TIM1_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

14.3.15.5 使用一个外部触发同步地启动 2 个定时器

这个例子中当定时器 1 的 TI1 输入上升时使能定时器 1，使能定时器 1 的同时使能定时器 2，参见图 14-41。为保证计数器的对齐，定时器 1 必须配置为主/从模式(对应 TI1 为从，对应定时器 2 为主)：

- 配置定时器 1 为主模式，送出它的使能做为触发输出(TIM1_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。

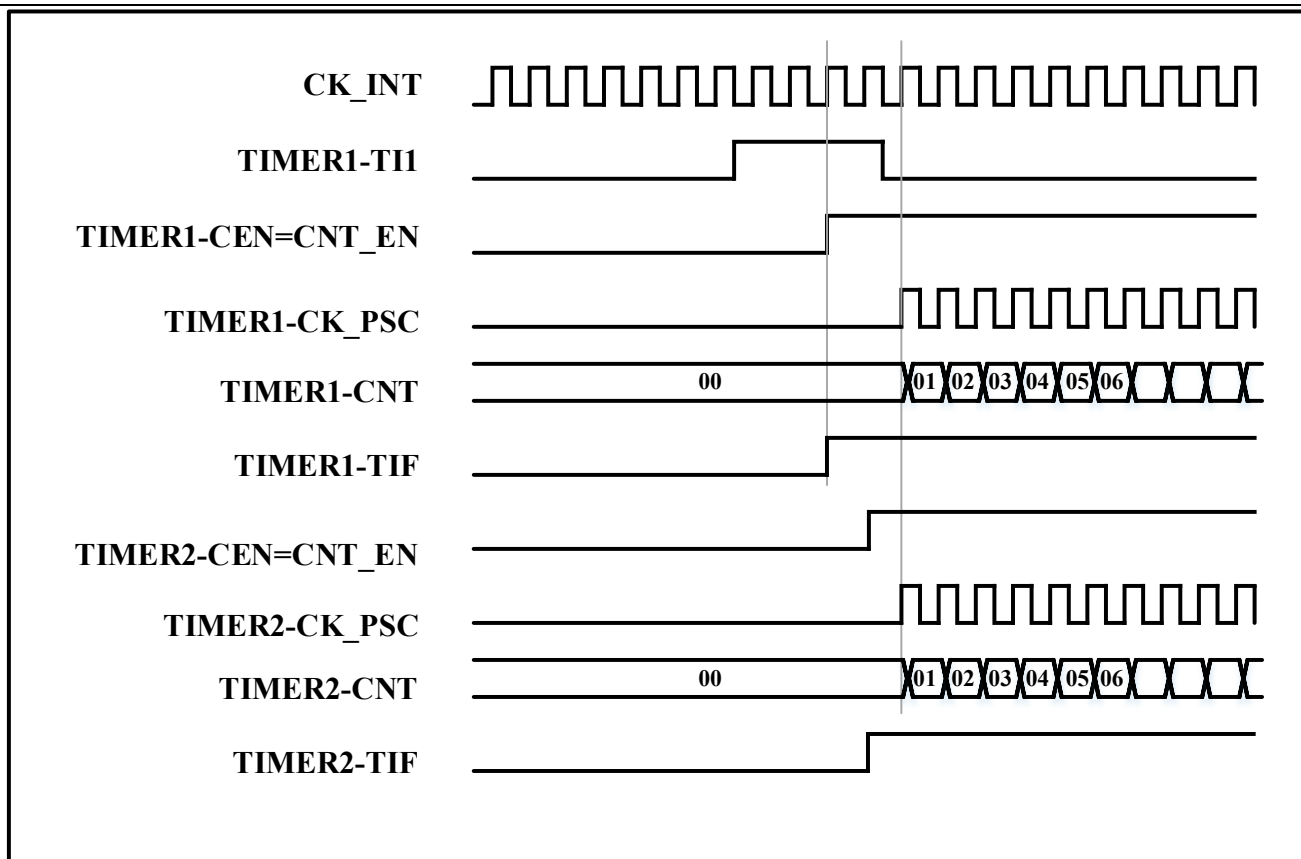


图 14-46. 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 2

14.3.16 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。

14.4 TIM2 寄存器列表

可以用字(32 位)的方式操作这些外设寄存器。

TIM2 基地址 0X40003C00

表 14.2. TIM2 寄存器列表和复位值

偏移地址	名称	描述	复位值
0x00	TIM2_CR1	TIM2 控制寄存器 1	0x00000000
0x04	TIM2_CR2	TIM2 控制寄存器 2	0x00000000
0x08	TIM2_SMCR	TIM2 从模式控制寄存器	0x00000000
0x0C	TIM2_DIER	TIM2 中断使能寄存器	0x00000000
0x10	TIM2_SR	TIM2 状态寄存器	0x00000000

偏移地址	名称	描述	复位值
0x14	TIM2_EGR	TIM2 事件产生寄存器	0x00000000
0x18	TIM2_CCMR1	TIM2 捕获/比较模式寄存器 1	0x00000000
0x1C	TIM2_CCMR2	TIM2 捕获/比较模式寄存器 2	0x00000000
0x20	TIM2_CCER	TIM2 捕获/比较使能寄存器	0x00000000
0x24	TIM2_CNT	TIM2 计数器	0x00000000
0x28	TIM2_PSC	TIM2 预分频器	0x00000000
0x2C	TIM2_ARR	TIM2 自动重装载寄存器	0x00000000
0x30	保留		
0x34	TIM2_CCR1	TIM2 捕获/比较寄存器 1	0x00000000
0x38	TIM2_CCR2	TIM2 捕获/比较寄存器 2	0x00000000
0x3C	TIM2_CCR3	TIM2 捕获/比较寄存器 3	0x00000000
0x40	TIM2_CCR4	TIM2 捕获/比较寄存器 4	0x00000000

14.5 TIM2 寄存器说明

14.5.1 TIM2 控制寄存器 1(TIM2_CR1)

偏移地址：0x00

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
						R/W		R/W	R/W		R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:10	—	保留，始终读为 0。	0x0	—
9:8	CKD[1:0]	时钟分频因子(Clock division) 这 2 位定义在定时器时钟(CK_INT)频率、死区时间和由死区发生器与数字滤波器(ETR,TIx)所用的采样时钟之间的分频比例。 00: $t_{DTS}=t_{CK_INT}$ 01: $t_{DTS}=2 \times t_{CK_INT}$	0	R/W

位	标记	功能描述	复位值	读写
		10: $t_{DTS}=4 \times t_{CK_INT}$ 11: 保留, 不要使用这个配置		
7	ARPE	自动重装载预装载允许位(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。	0	R/W
6:5	CMS[1:0]	选择中央对齐模式(Center-aligned mode selection) 00: 边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。 01: 中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道(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(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。 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(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。 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	R/W
4	DIR	方向(Direction) 0: 计数器向上计数; 1: 计数器向下计数。 注: 当计数器配置为中央对齐模式或编码器模式时, 该位为只读。	0	R/W
3	OPM	单脉冲模式(Onepulse 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。	0	R/W
2	URS	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, 则下述任一事件产生更新中断: -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, 则只有计数器溢出/下溢才产生更新中断。	0	R/W
1	UDIS	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。(译注: 更新影子寄存器) 1: 禁止 UEV。不产生更新事件, 影子寄存器(ARR、PSC、	0	R/W

位	标记	功能描述	复位值	读写
		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。		
0	CEN	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。	0	R/W

14.5.2 TIM2 控制寄存器 2(TIM2_CR2)

偏移地址: 0x04

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[1:0]			保留	CCUS	保留	CCPC
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W				R/W		R/W

位	标记	功能描述	复位值	读写
31:15	—	保留, 始终读为 0。	0x0	—
14	OIS4	输出空闲状态 4(OC4 输出)。参见 OIS1 位。	0	R/W
13	OIS3N	输出空闲状态 3(OC3N 输出)。参见 OIS1N 位。	0	R/W
12	OIS3	输出空闲状态 3(OC3 输出)。参见 OIS1 位。	0	R/W
11	OIS2N	输出空闲状态 2(OC2N 输出)。参见 OIS1N 位。	0	R/W
10	OIS2	输出空闲状态 2(OC2 输出)。参见 OIS1 位。	0	R/W
9	OIS1N	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。 注: 已经设置了 LOCK(TIM2_BKR 寄存器)级别 1、2 或 3 后, 该位不能被修改。	0	R/W
8	OIS1	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(TIM2_BKR 寄存器)级别 1、2 或 3 后, 该位不能被修改。	0	R/W

位	标记	功能描述	复位值	读写
7	TI1S	TI1 选择(TI1 selection) 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。	0	R/W
6:4	MMS[2:0]	主模式选择(Master mode selection) 这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下: 000: 复位-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。 001: 使能-计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟, 除非选择了主/从模式(见 TIM2_SMCR 寄存器中 MSM 位的描述)。 010: 更新-更新事件被选为触发输入(TRGO)。例如, 一个主定时器的时钟可以被用作一个从定时器的预分频器。 011: 比较脉冲-在发生一次捕获或一次比较成功时, 当要设置 CC1IF 标志时(即使它已经为高), 触发输出送出一个正脉冲(TRGO)。 100: 比较-OC1REF 信号被用于作为触发输出(TRGO)。 101: 比较-OC2REF 信号被用于作为触发输出(TRGO)。 110: 比较-OC3REF 信号被用于作为触发输出(TRGO)。 111: 比较-OC4REF 信号被用于作为触发输出(TRGO)。	0	R/W
3	-	保留, 始终读为 0。	0x0	-
2	CCUS	捕获/比较控制更新选择(Capture /compare control update selection) 0: 如果捕获/比较控制位是预装载的(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。	0	R/W
1	-	保留, 始终读为 0。	0	-
0	CCPC	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。	0	R/W

14.5.3 TIM2 从模式控制寄存器(TIM2_SMCR)

偏移地址：0x08

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]	ETF[3:0]				MSM	TS[2:0]				保留	SMS[2:0]		
R/W	R/W	R/W	R/W				R/W	R/W				保留	R/W		

位	标记	功能描述	复位值	读写
31:16	—	保留，始终读为 0。	0x0	—
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。	0	R/W
14	ECE	外部时钟使能位(External clock enable) 该位启用外部时钟模式 2 0: 禁止外部时钟模式 2； 1: 使能外部时钟模式 2。计数器由 ETRF 信号上的任意有效边沿驱动。 注 1: 设置 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	R/W
13:12	ETPS[1:0]	外部触发预分频(External trigger prescaler) 外部触发信号 ETRP 的频率必须多是 TIM2CLK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETRP 的频率。 00: 关闭预分频； 01: ETRP 频率除以 2； 10: ETRP 频率除以 4； 11: ETRP 频率除以 8。	0	R/W
11:8	ETF[3:0]	外部触发滤波(External trigger filter) 这些位定义了对 ETRP 信号采样的频率和对 ETRP 数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。 0000: 无滤波器，以 f_{DTS} 采样 0001: 采样频率 $f_{SAMPLING}=f_{CK_INT}$ ，N=2	0x0	R/W

位	标记	功能描述	复位值	读写
		0010: 采样频率 $f_{\text{SAMPLING}}=f_{\text{CK_INT}}$, $N=4$ 0011: 采样频率 $f_{\text{SAMPLING}}=f_{\text{CK_INT}}$, $N=8$ 0100: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/2$, $N=6$ 0101: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/2$, $N=8$ 0110: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, $N=6$ 0111: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$, $N=8$ 1000: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/8$, $N=6$ 1001: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/8$, $N=8$ 1010: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=5$ 1011: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=6$ 1100: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$, $N=8$ 1101: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=5$ 1110: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=6$ 1111: 采样频率 $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$, $N=8$		
7	MSM	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。	0	R/W
6:4	TS[2:0]	触发选择(Trigger selection) 这 3 位选择用于同步计数器的触发输入。 000: 内部触发 0(ITR0) 001: 内部触发 1(ITR1) 010: 内部触发 2(ITR2) 011: 内部触发 3(ITR3) 100: TI1 的边沿检测器(TI1F_ED) 101: 滤波后的定时器输入 1(TI1FP1) 110: 滤波后的定时器输入 2(TI2FP2) 111: 外部触发输入(ETRF)更多有关 ITRx 的细节, 参见表 14.3。 注: 这些位只能在未用到(如 SMS=000)时被改变, 以避免在改变时产生错误的边沿检测。	0x0	R/W
3	-	保留, 始终读为 0。	0	-
2:0	SMS[2:0]	从模式选择(Slave mode selection) 当选择了外部信号, 触发信号(TRGI)的有效边沿与选中的外部输入极性相关(见输入控制寄存器和控制寄存器的说明) 000: 关闭从模式—如果 CEN=1, 则预分频器直接由内部时钟驱动。 001: 编码器模式 1—根据 TI1FP1 的电平, 计数器在 TI2FP2 的边沿向上/下计数。 010: 编码器模式 2—根据 TI2FP2 的电平, 计数器在 TI1FP1 的边沿向上/下计数。	0	R/W

位	标记	功能描述	复位值	读写
		<p>011: 编码器模式 3—根据另一个信号的输入电平, 计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数。</p> <p>100: 复位模式—选中的触发输入(TRGI)的上升沿重新初始化计数器, 并且产生一个更新寄存器的信号。</p> <p>101: 门控模式—当触发输入(TRGI)为高时, 计数器的时钟开启。</p> <p>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。</p> <p>110: 触发模式—计数器在触发输入 TRGI 的上升沿启动(但不复位), 只有计数器的启动是受控的。</p> <p>111: 外部时钟模式 1—选中的触发输入(TRGI)的上升沿驱动计数器。</p> <p>注: 如果 TI1F_EN 被选为触发输入(TS=100)时, 不要使用门控模式。</p> <p>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。</p>		

表 14.3. TIM2 内部触发连接

从定时器	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM2	tim1_trgo	irq_timer10	irq_timer11	irq_pca

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

14.5.4 TIM2 中断使能寄存器(TIM2_DIER)

偏移地址: 0x0C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									TIE	保留	CC4IE	CC3IE	CC2IE	CC1IE	UIE
									R/W		R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:7	—	保留, 始终读为 0。	0x0	—
6	TIE	<p>触发中断使能(Trigger interrupt enable)</p> <p>0: 禁止触发中断;</p> <p>1: 使能触发中断。</p>	0	R/W

位	标记	功能描述	复位值	读写
5	-	保留，始终读为 0。	0	-
4	CC4IE	允许捕获/比较 4 中断(Capture /Compare 4 interrupt enable) 0: 禁止捕获/比较 4 中断; 1: 允许捕获/比较 4 中断。	0	R/W
3	CC3IE	允许捕获/比较 3 中断(Capture /Compare 3 interrupt enable) 0: 禁止捕获/比较 3 中断; 1: 允许捕获/比较 3 中断。	0	R/W
2	CC2IE	允许捕获/比较 2 中断(Capture /Compare 2 interrupt enable) 0: 禁止捕获/比较 2 中断; 1: 允许捕获/比较 2 中断。	0	R/W
1	CC1IE	允许捕获/比较 1 中断(Capture /Compare 1 interrupt enable) 0: 禁止捕获/比较 1 中断; 1: 允许捕获/比较 1 中断。	0	R/W
0	UIE	允许更新中断(Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。	0	R/W

14.5.5 TIM2 状态寄存器(TIM2_SR)

偏移地址: 0x10

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			CC 4OF	CC 3OF	CC 2OF	CC 1OF	保留		TIF	保留	CC 4IF	CC 3IF	CC 2IF	CC 1IF	UIF
			RC W0	RC W0	RC W0	RC W0			RC W0		RC W0	RC W0	RC W0	RC W0	

位	标记	功能描述	复位值	读写
31:13	-	保留，始终读为 0。	0x0	-
12	CC4OF	捕获/比较 4 重复捕获标记(Capture /Compare 4 over capture flag) 参见 CC1OF 描述。	0	RCW0
11	CC3OF	捕获/比较 3 重复捕获标记(Capture /Compare 3 over capture flag) 参见 CC1OF 描述。	0	RCW0
10	CC2OF	捕获/比较 2 重复捕获标记(Capture /Compare 2 over capture flag) 参见 CC1OF 描述。	0	RCW0

位	标记	功能描述	复位值	读写
9	CC1OF	捕获/比较 1 重复捕获标记(Capture /Compare 1 over capture flag) 仅当相应的通道被配置为输入捕获时, 该标记可由硬件置 1。写 0 可清除该位。 0: 无重复捕获产生; 1: 计数器的值被捕获到 TIM2_CCR1 寄存器时, CC1IF 的状态已经为'1'。	0	RCW0
8: 7	-	保留, 始终读为 0。	0	-
6	TIF	触发器中断标记(Trigger interrupt flag) 当发生触发事件(当从模式控制器处于除门控模式外的其它模式时, 在 TRGI 输入端检测到有效边沿, 或门控模式下的任一边沿)时由硬件对该位置'1'。它由软件清'0'。 0: 无触发器事件产生; 1: 触发中断等待响应。	0	RCW0
5	-	保留, 始终读为 0。	0	-
4	CC4IF	捕获/比较 4 中断标记(Capture /Compare 4 interrupt flag) 参考 CC1IF 描述。	0	RCW0
3	CC3IF	捕获/比较 3 中断标记(Capture /Compare 3 interrupt flag) 参考 CC1IF 描述。	0	RCW0
2	CC2IF	捕获/比较 2 中断标记(Capture /Compare 2 interrupt flag) 参考 CC1IF 描述。	0	RCW0
1	CC1IF	捕获/比较 1 中断标记(Capture /Compare 1 interrupt flag) 如果通道 CC1 配置为输出模式: 当计数器值与比较值匹配时该位由硬件置 1, 但在中心对称模式下除外(参考 TIM2_CR1 寄存器的 CMS 位)。它由软件清'0'。 0: 无匹配发生; 1: TIM2_CNT 的值与 TIM2_CCR1 的值匹配。 当 TIM2_CCR1 的内容大于 TIM2_APR 的内容时, 在向上或向上/下计数模式时计数器溢出, 或向下计数模式时的计数器下溢条件下, CC1IF 位变高 如果通道 CC1 配置为输入模式: 当捕获事件发生时该位由硬件置'1', 它由软件清'0'或通过读 TIM2_CCR1 清'0'。 0: 无输入捕获产生; 1: 计数器值已被捕获(拷贝)至 TIM2_CCR1(在 IC1 上检测到与所选极性相同的边沿)。	0	RCW0
0	UIF	更新中断标记(Update interrupt flag) 当产生更新事件时该位由硬件置'1'。它由软件清'0'。 0: 无更新事件产生; 1: 更新中断等待响应。当寄存器被更新时该位由硬件置'1': - 若 TIM2_CR1 寄存器的 UDIS=0, 当重复计数器数值上溢或下溢时(重复计数器=0 时产生更新事件)。	0	RCW0

位	标记	功能描述	复位值	读写
		<ul style="list-style-type: none"> 若 TIM2_CR1 寄存器的 URS=0、UDIS=0，当设置 TIM2_EGR 寄存器的 UG=1 时产生更新事件，通过软件对计数器 CNT 重新初始化时。 若 TIM2_CR1 寄存器的 URS=0、UDIS=0，当计数器 CNT 被触发事件重新初始化时。(参考 14.5.3: 从模式控制寄存器(TIM2_SMCR))。 		

14.5.6 TIM2 事件产生寄存器(TIM2_EGR)

偏移地址:0x14

复位值:0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									TG	保留	CC4G	CC3G	CC2G	CC1G	UG
									WO		WO	WO	WO	WO	WO

位	标记	功能描述	复位值	读写
31:7	—	保留，始终读为 0。	0x0	—
6	TG	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，则产生相应的中断。	0	WO
5	-	保留，始终读为 0。	0	-
4	CC4G	产生捕获/比较 4 事件(Capture /Compare 4 generation) 参考 CC1G 描述。	0	WO
3	CC3G	产生捕获/比较 3 事件(Capture /Compare 3 generation) 参考 CC1G 描述。	0	WO
2	CC2G	产生捕获/比较 2 事件(Capture /Compare 2 generation) 参考 CC1G 描述。	0	WO
1	CC1G	产生捕获/比较 1 事件(Capture /Compare 1 generation) 该位由软件置'1'，用于产生一个捕获/比较事件，由硬件自动清'0'。 0: 无动作; 1: 在通道 CC1 上产生一个捕获/比较事件: 若通道 CC1 配置为输出: 设置 CC1IF=1，若开启对应的中断，则产生相应的中断。	0	WO

位	标记	功能描述	复位值	读写
		若通道 CC1 配置为输入： 当前的计数器值被捕获至 TIM2_CCR1 寄存器；设置 CC1IF=1，若开启对应的中断，则产生相应的中断。若 CC1IF 已经为 1，则设置 CC1OF=1。		
0	UG	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。	0	WO

14.5.7 TIM2 捕获/比较模式寄存器 1(TIM2_CCMR1)

偏移地址：0x18

复位值：0x00000000

通道可用于输入(捕获模式)或输出(比较模式)，通道的方向由相应的 CCxS 位定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		OC1 CE	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
IC2F[3:0]			IC2PSC[1:0]		IC1F[3:0]			IC1PSC[1:0]							
R/W															

输出比较模式：

位	标记	功能描述	复位值	读写
31:16	—	保留，始终读为 0。	0x0	—
15	OC2CE	输出比较 2 清 0 使能(Output Compare 2 clear enable)	0	R/W
14:12	OC2M[2:0]	输出比较 2 模式(Output Compare 2 mode)	0x0	R/W
11	OC2PE	输出比较 2 预装载使能(Output Compare 2 preload enable)	0	R/W
10	OC2FE	输出比较 2 快速使能(Output Compare 2 fast enable)	0	R/W
9:8	CC2S[1:0]	捕获/比较 2 选择。(Capture /Compare 2 selection) 该位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC2 通道被配置为输出；	0x0	R/W

位	标记	功能描述	复位值	读写
		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。		
7	OC1CE	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。	0	R/W
6:4	OC1M[2:0]	输出比较 1 模式(Output Compare 1 mode) 该 3 位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1、OC1N 的值。OC1REF 是高电平有效, 而 OC1、OC1N 的有效电平取决于 CC1P、CC1NP 位。 000: 冻结。输出比较寄存器 TIM2_CCR1 与计数器 TIM2_CNT 间的比较对 OC1REF 不起作用; 001: 匹配时设置通道 1 为有效电平。当计数器 TIM2_CNT 的值与捕获/比较寄存器 1(TIM2_CCR1)相同时, 强制 OC1REF 为高。 010: 匹配时设置通道 1 为无效电平。当计数器 TIM2_CNT 的值与捕获/比较寄存器 1(TIM2_CCR1)相同时, 强制 OC1REF 为低。 011: 翻转。当 TIM2_CCR1=TIM2_CNT 时, 翻转 OC1REF 的电平。 100: 强制为无效电平。强制 OC1REF 为低。 101: 强制为有效电平。强制 OC1REF 为高。 110: PWM 模式 1 — 在向上计数时, 一旦 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)。 111: 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。 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中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。	0x0	R/W
3	OC1PE	输出比较 1 预装载使能(Output Compare 1 preload enable) 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	0	R/W

位	标记	功能描述	复位值	读写
		到来时被加载至当前寄存器中。 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。		
2	OC1FE	输出比较 1 快速使能(Output Compare 1 fast enable)该位用于加快 CC 输出对触发输入事件的响应。 0：根据计数器与 CCR1 的值，CC1 正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活 CC1 输出的小延时为 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。 OCFE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。	0	R/W
1:0	CC1S[1:0]	捕获/比较 1 选择。(Capture /Compare 1 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC1 通道被配置为输出； 01：CC1 通道被配置为输入，IC1 映射在 TI1 上； 10：CC1 通道被配置为输入，IC1 映射在 TI2 上； 11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 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	R/W

输入比较模式：

位	标记	功能描述	复位值	读写
31:16	—	保留，始终读为 0。	0x0	—
15:12	IC2F[3:0]	输入捕获 2 滤波器(Input capture 2 filter)	0x0	R/W
11:10	IC2PSC[1:0]	输入/捕获 2 预分频器(Input capture 2 prescaler)	0x0	R/W
9:8	CC2S[1:0]	捕获/比较 2 选择(Capture /Compare 2 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC2 通道被配置为输出； 01：CC2 通道被配置为输入，IC2 映射在 TI2 上； 10：CC2 通道被配置为输入，IC2 映射在 TI1 上； 11：CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 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	R/W
7:4	IC1F[3:0]	输入捕获 1 滤波器(Input capture 1 filter)	0x0	R/W

位	标记	功能描述	复位值	读写
		<p>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:</p> <p>0000: 无滤波器, 以 f_{DTS} 采样</p> <p>0001: 采样频率 $f_{SAMPLING}=f_{CK_INT}$, $N=2$</p> <p>0010: 采样频率 $f_{SAMPLING}=f_{CK_INT}$, $N=4$</p> <p>0011: 采样频率 $f_{SAMPLING}=f_{CK_INT}$, $N=8$</p> <p>0100: 采样频率 $f_{SAMPLING}=f_{DTS}/2$, $N=6$</p> <p>0101: 采样频率 $f_{SAMPLING}=f_{DTS}/2$, $N=8$</p> <p>0110: 采样频率 $f_{SAMPLING}=f_{DTS}/4$, $N=6$</p> <p>0111: 采样频率 $f_{SAMPLING}=f_{DTS}/4$, $N=8$</p> <p>1000: 采样频率 $f_{SAMPLING}=f_{DTS}/8$, $N=6$</p> <p>1001: 采样频率 $f_{SAMPLING}=f_{DTS}/8$, $N=8$</p> <p>1010: 采样频率 $f_{SAMPLING}=f_{DTS}/16$, $N=5$</p> <p>1011: 采样频率 $f_{SAMPLING}=f_{DTS}/16$, $N=6$</p> <p>1100: 采样频率 $f_{SAMPLING}=f_{DTS}/16$, $N=8$</p> <p>1101: 采样频率 $f_{SAMPLING}=f_{DTS}/32$, $N=5$</p> <p>1110: 采样频率 $f_{SAMPLING}=f_{DTS}/32$, $N=6$</p> <p>1111: 采样频率 $f_{SAMPLING}=f_{DTS}/32$, $N=8$</p>		
3:2	IC1PSC[1:0]	<p>输入/捕获 1 预分频器(Input capture 1 prescaler)</p> <p>这 2 位定义了 CC1 输入(IC1)的预分频系数。</p> <p>一旦 $CC1E=0$(TIM2_CCER 寄存器中), 则预分频器复位。</p> <p>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;</p> <p>01: 每 2 个事件触发一次捕获;</p> <p>10: 每 4 个事件触发一次捕获;</p> <p>11: 每 8 个事件触发一次捕获。</p>	0x0	R/W
1:0	CC1S[1:0]	<p>捕获/比较 1 选择(Capture /Compare 1 Selection)</p> <p>这 2 位定义通道的方向(输入/输出), 及输入脚的选择:</p> <p>00: CC1 通道被配置为输出;</p> <p>01: CC1 通道被配置为输入, IC1 映射在 TI1 上;</p> <p>10: CC1 通道被配置为输入, IC1 映射在 TI2 上;</p> <p>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)。</p> <p>注: CC1S 仅在通道关闭时(TIM2_CCER 寄存器的 $CC1E=0$)才是可写的。</p>	0x0	R/W

14.5.8 TIM2 捕获/比较模式寄存器 2(TIM2_CCMR2)

地址偏移: 0x1C

复位值：0x00000000

参看以上 CCMR1 寄存器的描述。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4 CE	OC4M[2:0]			OC4 PE	OC4 FE	CC4S[1:0]		OC3 CE	OC3M[2:0]			OC3 PE	OC3 FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
R/W															

输出比较模式：

位	标记	功能描述	复位值	读写
31:16	—	保留，始终读为 0。	0x0	—
15	OC4CE	输出比较 4 清 0 使能(Output Compare 4 clear enable)	0	R/W
14:12	OC4M[2:0]	输出比较 4 模式(Output Compare 4 mode)	0x0	R/W
11	OC4PE	输出比较 4 预装载使能(Output Compare 4 preload enable)	0	R/W
10	OC4FE	输出比较 4 快速使能(Output Compare 4 fast enable)	0	R/W
9:8	CC4S[1:0]	捕获/比较 4 选择。(Capture /Compare 4 selection) 该 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC4 通道被配置为输出； 01：CC4 通道被配置为输入，IC4 映射在 TI4 上； 10：CC4 通道被配置为输入，IC4 映射在 TI3 上； 11：CC4 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 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	R/W
7	OC3CE	输出比较 3 清'0'使能(Output Compare 3 clear enable)	0	R/W
6:4	OC3M[2:0]	输出比较 3 模式(Output Compare 3 mode)	0x0	R/W
3	OC3PE	输出比较 3 预装载使能(Output Compare 3 preload enable)	0	R/W
2	OC3FE	输出比较 3 快速使能(Output Compare 3 fast enable)	0	R/W
1:0	CC3S[1:0]	捕获/比较 3 选择。(Capture /Compare 3 selection) 这 2 位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC3 通道被配置为输出； 01：CC3 通道被配置为输入，IC3 映射在 TI3 上； 10：CC3 通道被配置为输入，IC3 映射在 TI4 上； 11：CC3 通道被配置为输入，IC3 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 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	R/W

输入捕获模式:

位	标记	功能描述	复位值	读写
31:16	—	保留, 始终读为 0。	0x0	—
15:12	IC4F[3:0]	输入捕获 4 滤波器(Input capture 4 filter)	0x0	R/W
11:10	IC4PSC[1:0]	输入/捕获 4 预分频器(Input capture 4 prescaler)	0x0	R/W
9:8	CC4S[1:0]	捕获/比较 4 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: 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	R/W
7:4	IC3F[3:0]	输入捕获 3 滤波器(Input capture 1 filter)	0x0	R/W
3:2	IC3PSC[1:0]	输入/捕获 3 预分频器(Input capture 1 prescaler)	0x0	R/W
1:0	CC3S[1:0]	捕获/比较 3 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: 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	R/W

14.5.9 TIM2 捕获/比较使能寄存器(TIM1_CCER)

地址偏移: 0x20

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CC4P	CC4E	保留	CC3P	CC3E	保留	CC2P	CC2E	保留	CC1P	CC1E				
	R/W	R/W		R/W	R/W		R/W	R/W							

位	标记	功能描述	复位值	读写
31:14	—	保留，始终读为 0。	0x0	—
13	CC4P	输入/捕获 4 输出极性(Capture /Compare 4 output polarity) 参考 CC1P 的描述。	0	R/W
12	CC4E	输入/捕获 4 输出使能(Capture /Compare 4 output enable) 参考 CC1E 的描述。	0	R/W
11:10	—	保留，始终读为 0。	0x0	—
9	CC3P	输入/捕获 3 输出极性(Capture /Compare 3 output polarity) 参考 CC1P 的描述。	0	R/W
8	CC3E	输入/捕获 3 输出使能(Capture /Compare 3 output enable) 参考 CC1E 的描述。	0	R/W
7:6	—	保留，始终读为 0。	0x0	—
5	CC2P	输入/捕获 2 输出极性(Capture /Compare 2 output polarity) 参考 CC1P 的描述。	0	R/W
4	CC2E	输入/捕获 2 输出使能(Capture /Compare 2 output enable) 参考 CC1E 的描述。	0	R/W
3:2	—	保留，始终读为 0。	0x0	—
1	CC1P	输入/捕获 1 输出极性(Capture /Compare 1 output polarity) CC1 通道配置为输出： 0: OC1 高电平有效； 1: OC1 低电平有效。 CC1 通道配置为输入： 该位选择是 IC1 还是 IC1 的反相信号作为触发或捕获信号。 0: 不反相：捕获发生在 IC1 的上升沿；当用作外部触发器时，IC1 不反相。 1: 反相：捕获发生在 IC1 的下降沿；当用作外部触发器时，IC1 反相。 注：一旦 LOCK 级别(TIM2_BDTR 寄存器中的 LOCK 位) 设为 3 或 2，则该位不能被修改。	0	R/W
0	CC1E	输入/捕获 1 输出使能(Capture /Compare 1 output enable) CC1 通道配置为输出： 0: 关闭—OC1 禁止输出，因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 1: 开启—OC1 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 CC1 通道配置为输入： 该位决定了计数器的值是否能捕获入 TIM2_CCR1 寄存器。 0: 捕获禁止； 1: 捕获使能。	0	R/W

表 14.3. 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

14.5.10 TIM2 计数器(TIM2_CNT)

地址偏移: 0x24

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:16	—	保留, 始终读为 0。	0x0	—
15:0	CNT[15:0]	计数器的值(Counter value)	0x0	R/W

14.5.11 TIM2 预分频器(TIM2_PSC)

地址偏移: 0x28

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:16	—	保留, 始终读为 0。	0x0	—
15:0	PSC[15:0]	预分频器的值(Prescaler value) 计数器的时钟频率(CK_CNT)等于 $f_{CK_PSC}/(PSC[15:0]+1)$ 。 PSC 包含了每次当更新事件产生时, 装入当前预分频器寄存器的值; 更新事件包括计数器被 TIM_EGR 的 UG 位清'0' 或被工作在复位模式的从控制器清'0'。	0x0	R/W

14.5.12 TIM2 自动重装载寄存器(TIM2_ARR)

地址偏移: 0x2C

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:16	—	保留, 始终读为 0。	0x0	—
15:0	ARR[15:0]	自动重装载的值(Prescaler value) ARR 包含了将要装载入实际的自动重装载寄存器的值。详细参考 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	R/W

14.5.13 TIM2 捕获/比较寄存器 1(TIM2_CCR1)

地址偏移: 0x34

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	—	保留, 始终读为 0。	0x0	—
15:0	CCR1[15:0]	捕获/比较通道 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)中未选择预装载功能, 写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时, 此预装载值才传输至当前捕获/比较 1	0x0	R/W

位	标记	功能描述	复位值	读写
		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。		

14.5.14 TIM2 捕获/比较寄存器 2(TIM2_CCR2)

地址偏移：0x38

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	—	保留，始终读为 0。	0x0	—
15:0	CCR2[15:0]	捕获/比较通道 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	R/W

14.5.15 TIM2 捕获/比较寄存器 3(TIM2_CCR3)

地址偏移：0x3C

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	—	保留，始终读为 0。	0x0	—
15:0	CCR3[15:0]	<p>捕获/比较通道 3 的值(Capture /Compare 3 value)</p> <p>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。</p> <p>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。</p>	0x0	R/W

14.5.16 TIM2 捕获/比较寄存器 4(TIM2_CCR4)

地址偏移：0x40

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
R/W															

位	标记	功能描述	复位值	读写
31:16	—	保留，始终读为 0。	0x0	—
15:0	CCR4[15:0]	<p>捕获/比较通道 4 的值(Capture /Compare 4 value)</p> <p>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</p>	0x0	R/W



位	标记	功能描述	复位值	读写
		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。		

15. 可编程计数阵列 (PCA)

15.1 PCA 简介

PCA(可编程计数器阵列 Programmable Counter Array)支持最多 5 个 16 位的捕获/比较模块。该定时/计数器可用作为一个通用的时钟计数/事件计数器的捕获/比较功能。PCA 的每个模块都可以进行独立编程, 以提供输入捕捉、输出比较或脉冲宽度调制。另外模块 4 有额外的看门狗定时器模式。

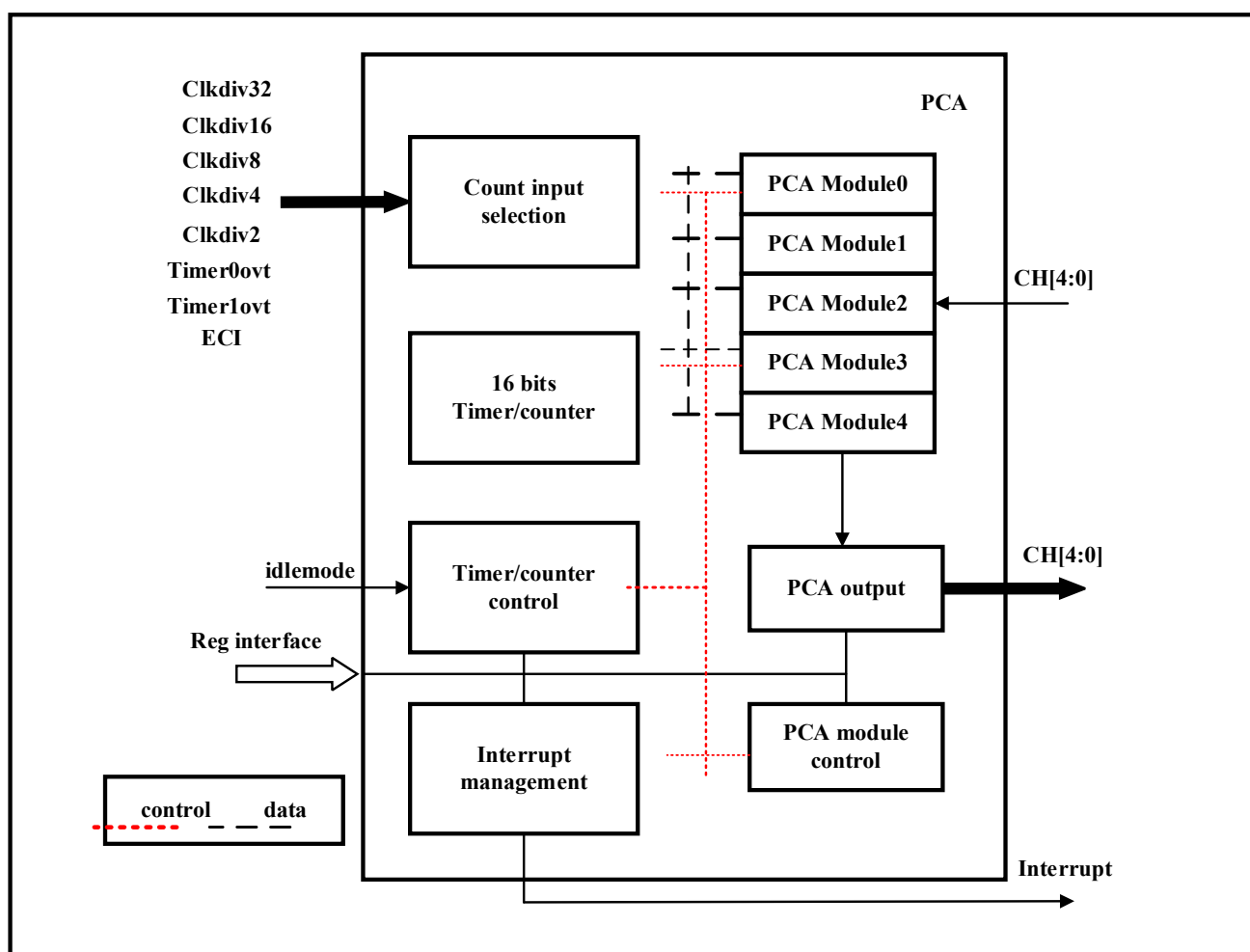


图 15-1. PCA 整体框图

15.2 PCA 功能描述

每个模块都可被配置为独立工作, 有三种工作方式: 边沿触发捕捉、输出比较、8 位脉宽

调制。每个模块在系统控制器中都有属于自己的功能寄存器，这些寄存器用于配置模块的工作方式和与模块交换数据。每组比较/捕获模块是由一个比较/捕获寄存器组(CCAPx)，以及 1 个 16 位比较器，和各种逻辑门控制组成。寄存器组用来存储时间或次数，针对外部触发捕获条件，或内部触发比较条件。在 PWM 模式下，寄存器(CCAPxL)用来控制输出波形的占空比。

每个模块都可以独立编程的操作在任何以下模式：

- 16 位捕获模式的上升沿，下降沿或任意沿触发。
- 比较模式：16 位软件定时器，16 位高速输出或 8 位脉冲宽度调制。
- 未启动。

比较/捕获模块模式寄存器(CCAPMx)确定相应的工作模式。对于比较/捕获模块进行编程时，他们是基于共同的时间计数。定时器/计数器打开和关闭通过 CR.CR 位即可控制 PCA 定时/计数器的运行。如果相应的使能位 (CCAPMx.CCIE) 被设置，当出现匹配或捕获时，比较/捕获标志(CR.CCFx)被设置，并产生 PCA 中断请求。CPU 可以在任何时候读写 CCAPx 寄存器。

15.2.1 PCA 定时/计数器

CNT 的这组特殊功能寄存器可用作为一个 16 位定时器/计数器。这是一个 16 位向上计数的计数器。如果 MOD.CFIE 位被置“1”时，当 CNT 溢出时硬件自动设置 PCA 溢出标志(CR.CF)并产生 PCA 中断请求。MOD.CPS[2: 0]三位选择八个信号输入到定时器/计数器。

- 系统时钟 PCLK 的 32 分频。
- 系统时钟 PCLK 的 16 分频。
- 系统时钟 PCLK 的 8 分频。
- 系统时钟 PCLK 的 4 分频。
- 系统时钟 PCLK 的 2 分频。
- 定时器 0 的溢出(overflow)。每次定时器 0 计数溢出后，CNT 就递增，这样提供了 PCA 的可变编程频率输入。
- 定时器 1 的溢出(overflow)。每次定时器 1 计数溢出后，CNT 就递增，这样提供了 PCA

的可变编程频率输入。

- ECI。CPU 每过 4 个 PCLK 时钟周期就对 PCAECI 进行采样，当每次采样结果从高变低时，CNT_CL(CNT low 8bit)自动加 1，因此最高的 ECI 输入频率不能高于系统时钟 PCLK 的 1/8，以满足采样需求。

设置运行控制器(CR.CR)启动 PCA 定时/计数器。当 MOD.CIDL 置“1”后，PCA 定时器/计数器可以继续运行在空闲模式下。CPU 可以随时读取 CNT 的数值，但当计数启动后(CR.CR=1)时，为了防止计数错误，CNT 是禁止写入的。

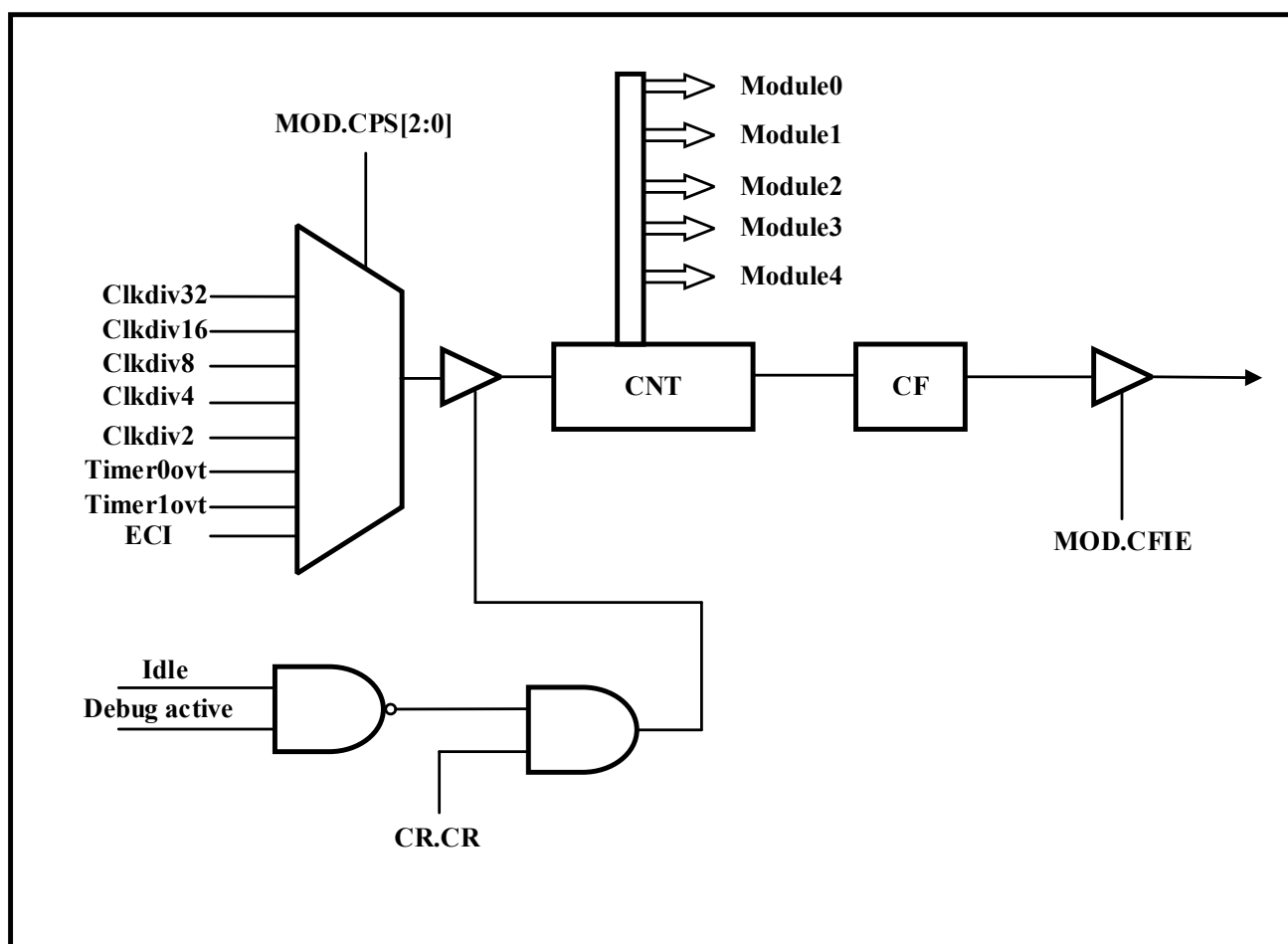


图 15-2. PCA 计数器框图

15.2.2 捕获功能

PCA 捕获模式提供了 5 路 PCA 测量脉冲周期、脉冲宽、占空比和相位差的功能。引脚上出现的电平跳变导致 PCA 捕捉 PCA 计数器/定时器的值并将其装入到对应模块的 16 位捕捉/比

较寄存器(CCAPx)。CCAPMx.CAPP 以及 CCAPMx.CAPN 位用于选择触发捕捉的电平变化类型：低电平到高电平(正沿)、高电平到低电平(负沿)或任何变化(正沿或负沿)。当捕捉发生时，CR 中的捕捉/比较标志(CCFx)被置为逻辑‘1’并产生一个中断请求(如果 CCF 中断被允许)。当 CPU 转向中断服务程序时，CR.CCFx 位不能被硬件自动清除，用户软件写 INTCLR.CCFx 寄存器清除此标志位。

如果 CCPMx.CAPP 以及 CCAPMx.CAPN 位都被设置为逻辑‘1’，可以通过直接读对应端口引脚的状态来确定本次捕捉是由上升沿触发还是由下降沿触发。分辨率等于定时器/计数器的时钟。输入信号必须在高电平或低电平期间至少保持 2 个时钟周期，以保证输入信号能够被硬件识别。CPU 可以在任何时候读取或写入 CCAPx 的寄存器。捕获设置：

- 当需要在外部上升沿进行捕获，CCPMx.CAPP=“1”以及 CCAPMx.CAPN=“0”
- 当需要在外部下降沿进行捕获，CCPMx.CAPP=“0”以及 CCAPMx.CAPN=“1”
- 当需要在外部上升、下降沿进行捕获，CCPMx.CAPP=“1”以及 CCAPMx.CAPN=“1”

注意：

随后由同一模块的捕获值会覆盖现有捕获的值。为了保持捕获的值，在中断服务程序中将它保存在 RAM 里面，这个操作必须在下一次事件出现之前完成，否则就会丢失前面一次捕获采样值。

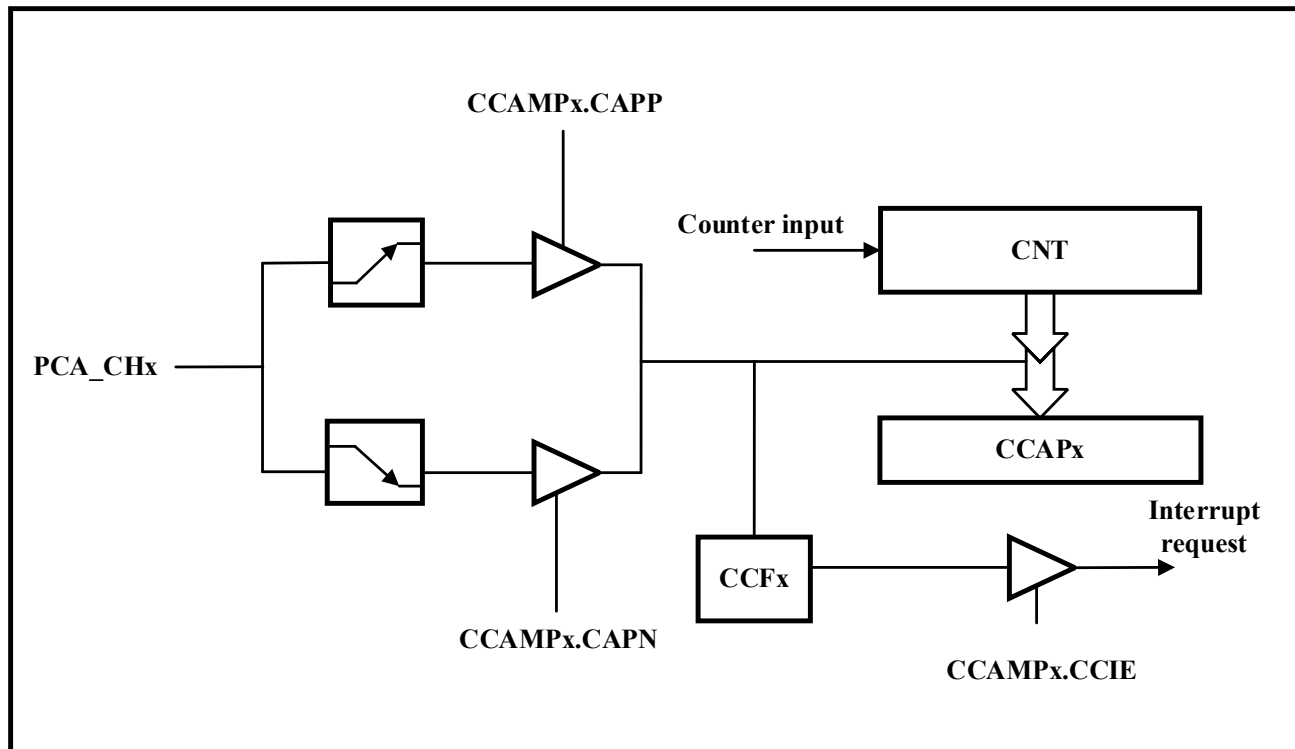


图 15-3. PCA 捕获功能框图

15.2.3 PCA 比较功能

比较功能提供如下功能，定时器，事件计数器，脉冲宽度调制。三种模式采用比较功能：16 位软件定时器模式，高速输出模式，PWM 模式。在前两个功能中，比较/捕获模块比较 16 位 PCA 定时器/计数器的值与预先加载到该模块的 CCAPx 寄存器中的 16 位值。在 PWM 模式下，PCA 模块不断地将 PCA 定时器/计数器低字节寄存器(CNT)与一个在 CCAPxL 模块寄存器 8 位的值进行比较。每 4 个时钟周期比较一次，即与最快的 PCA 定时器/计数器的时钟速率相匹配。设置 CCAPMx.ECOM 位选择该模块的比较功能。若要正确使用在比较模式下的模块，请遵守以下的一般程序：

- 选择 PCA 模块的操作模式
- 选择 PCA 定时器/计数器的输入信号。
- 比较值加载到模块的比较/捕获寄存器对。
- 设置 PCA 定时器/计数器运行控制位。
- 匹配后产生中断，清除模块的比较/捕获标志。

15.2.3.1 16 位软件计数器模式

要设定一个比较/捕获模块工作在 16 位软件定时器模式下，需要设置 CCAPMx.ECOM 和 CCAPMx.MAT 位。一旦在 PCA 定时器/计数器和比较/捕获的寄存器(CCAPx)之间发生了匹配，这将设置模块的比较/捕获标志(CR.CCFx)。这将产生一个中断请求，如果相应的中断使能位(CCAPMx.CCIE)设置。由于硬件并不清除比较/捕获标志(CR.CCFx)，用户必须清除软件标志。在中断服务程序中，一个新的 16 位比较值可以被写入比较/捕获的寄存器(CCAPx)。

注意：在更新这些寄存器时，为了防止无效的匹配发生，用户软件应该先写 CCAPxL，后写 CCAPxH。一旦写入 CCAPxL 就会清除禁用比较功能 ECOMx 位，而写入 CCAPxH 会同时设置的 ECOMx 位，重新启用比较功能。即当向 PCA0 的捕捉/比较寄存器写入一个 16 位数值时，应先写低字节。

15.2.3.2 高速输出模式

在高速输出模式，每当 PCA 计数器内的值与模块的 16 位捕捉/比较寄存器(CCAPx)发生匹配时，模块 PCA 的 PCA_CCAPOx 寄存器（对应于 PCA_CHx 引脚）的值将发生翻转。这可以提供比切换 IO 输出有更高精度，因为这个高速输出不会响应中断而影响输出频率，靠 CPU 来切换 IO 输出的话，功耗、精度都有所欠缺。

要设定一个比较/捕获模块的高速输出模式，设置 CCAPMx.ECOM，CCAPMx.MAT 和 CCAPMx.TOG 位。PCA 定时器/计数器和比较/捕获的寄存器(CCAPx)之间的匹配翻转 PCA 的 PCA_CCAPOx 寄存器的值，并设置模块的比较/捕获标志(CR.CCFx)。

用户也可以选择产生一个中断请求，通过设置相应的中断使能位(CCAPMx.CCIE)当匹配发生时，即可产生中断请求。由于硬件无法清除比较/捕获标志，用户必须在软件中清除这个标志位。如果用户在中断程序中不去改变比较/捕获寄存器，PCA 将重新计数，如相匹配则发生下一次翻转。在中断服务程序中，一个新的 16 位比较值可以被写入比较/捕获的寄存器(CCAPx)。

注意：为了防止无效的匹配，而更新这些寄存器，用户软件应该写 CCAPxL 的首先然后 CCAPxH。写到 CCAPxL 清除禁用比较功能 ECOM 位，而写到 CCAPxH 设置的 ECOM 位，重

新启用比较功能。

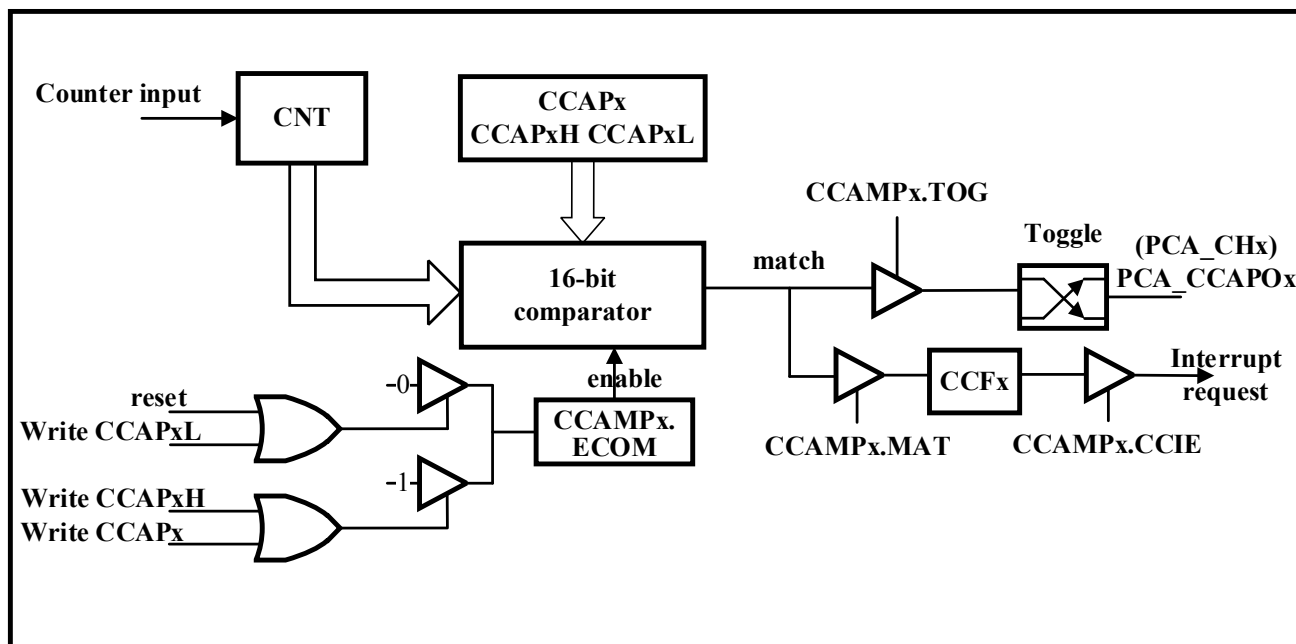


图 15-4. PCA 比较功能框图

15.2.3.3 PCA 8 位脉宽调制功能

脉宽调制是一种使用程序来控制波形占空比、周期、相位的技术。5 个 PCA 模块都可以被独立地用于在对应 PCA 的 PCA_CHx 引脚产生脉宽调制(PWM)输出，脉冲宽度为 8 位分辨率。PWM 输出的频率取决于 PCA 计数器/定时器的时基。使用模块的捕捉/比较寄存器 CCAPxL 来改变 PWM 输出信号的占空比。当 PCA 计数器/定时器的低字节(CNT_L)与 CCAPxL 中的值相等时,PCA 的 PCA_CHx 引脚上的输出被置“1”;当 CNT_L 中的计数值溢出时,PCA 的 PCA_CHx 输出被复位“0”。当计数器/定时器的低字节 CNT_L 溢出时(从 0xFF 到 0x00),保存在 CCAPxH 中的值被自动装入到 CCAPxL, 不需软件干预。

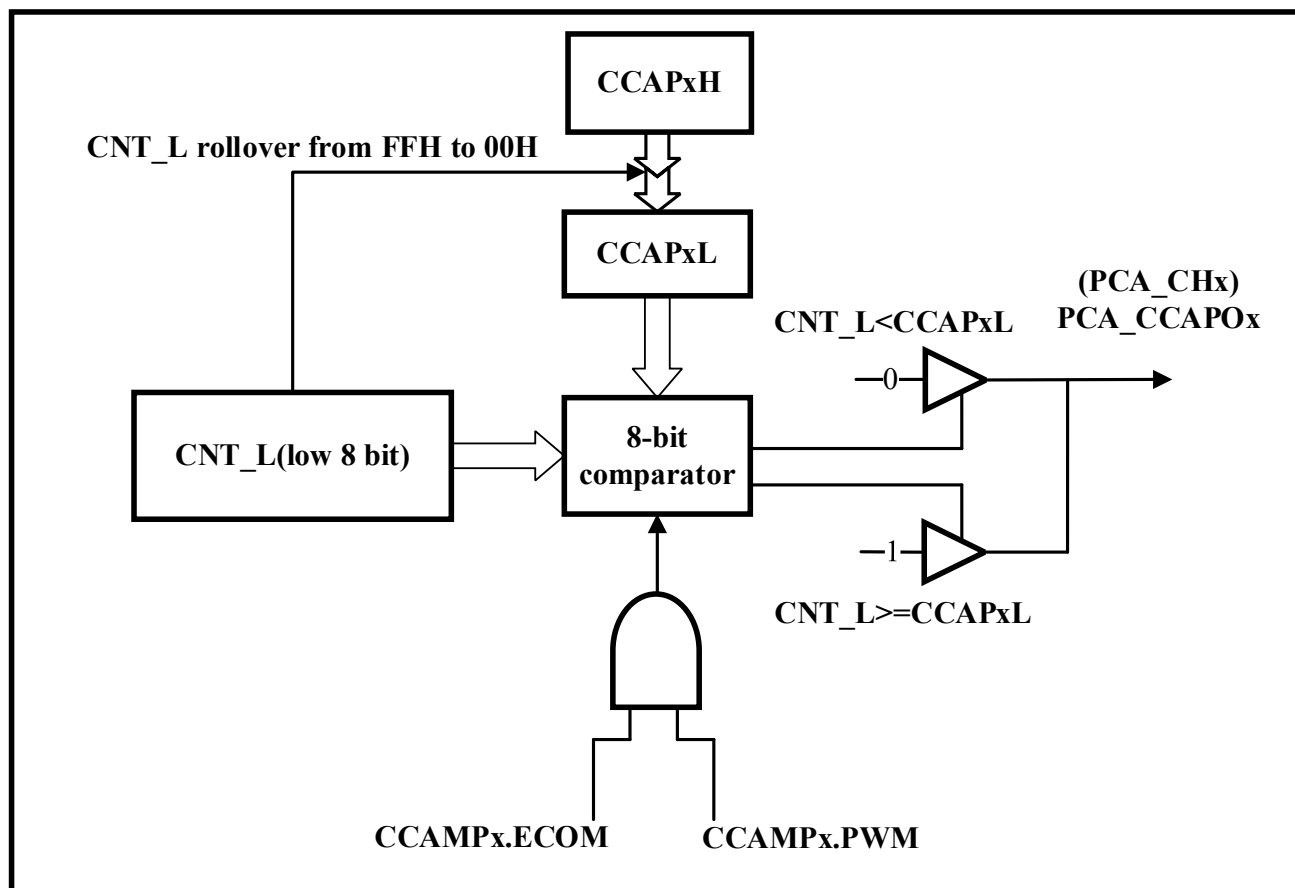


图 15-5. PCA PWM 功能框图

在这种模式下,PCA 定时器/计数器 CNT_L 的值是不断在低字节比较/捕获寄存器(CCAPxL)的值相比。当 $CNT_L < CCAPxL$, 输出波形为低。当两者匹配时($CNT_L = CCAPxL$), 输出波形去到高, 直到 CNT_L 溢出从 FFH 到 00H, 结束期间仍然很高。在溢出时, 在 CCAPxH 的值自动装载到 CCAPxL 内, 一个新的周期的开始。

在 CCAPxL 的值决定当前波形的占空比。在 CCAPxH 的值确定下一个波形的占空比。改变 CCAPxL 中的值即可更改的脉冲宽度调制。正如图所示, 8 位值在 CCAPxL 可以从 0(100% 占空比), 到 255(0.4% 占空比)。要改变 CCAPxL 值而不产生毛刺, 需要在高字节寄存器(CCAPxH)写入一个新值。当 CL 超过 0xFF 滚动到 0x00, 这个值是由硬件自动加载到 CCAPxL。

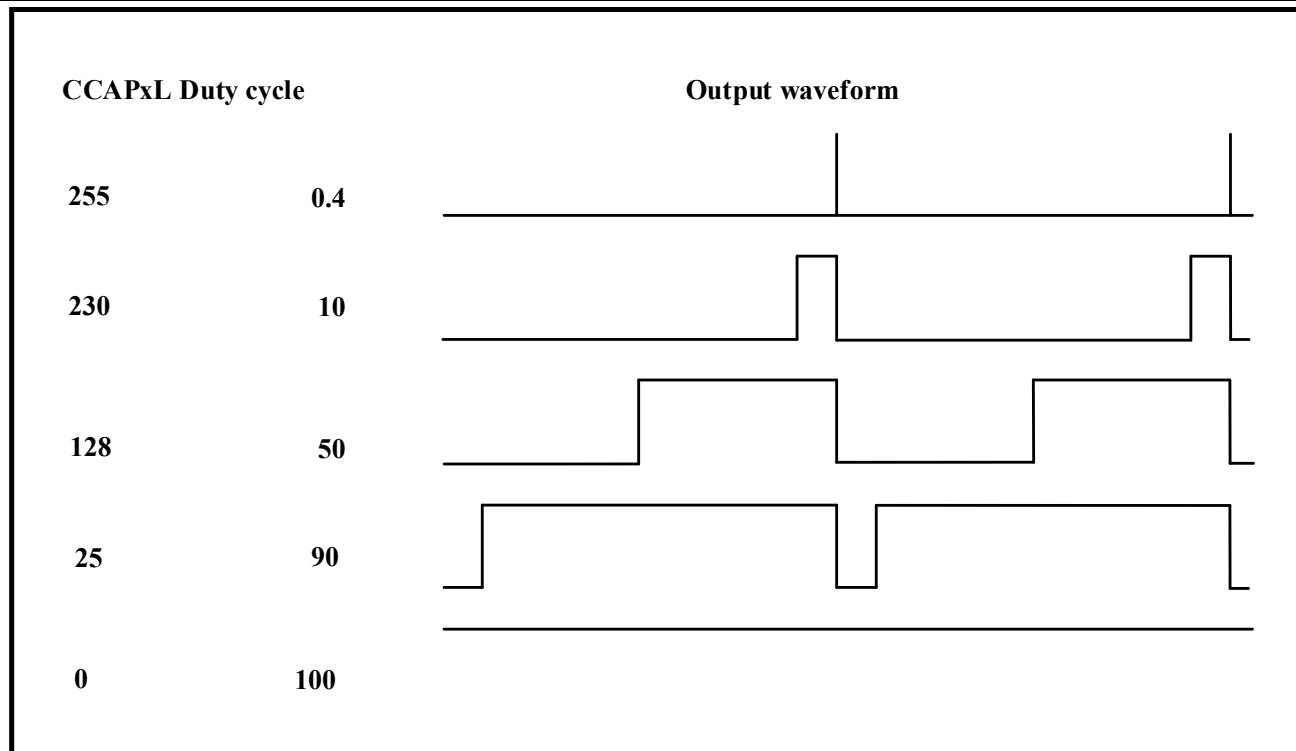


图 15-6. PCA PWM 输出波形

要设定一个比较/捕获模块在 PWM 模式下，需要设置 CCAPMx.ECOM 和 CCAPMx.PWM 位。另外 PCA 定时器/计数器由编程 MOD.CSP[2:0]可以选择输入计数信号频率。在 CCAPxL 输入一个 8 位的值指定第一个 PWM 波形的占空比。在 CCAPxH 输入一个 8 位的值会指定第二个 PWM 波形的占空比。设置定时器/计数器运行控制位(CR.CR)启动 PCA 定时器/计数器。

表 15.1. PCA 比较/捕获功能模块设置

ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	工作方式
X	1	0	0	0	0	X	用 CCPn 的正沿触发捕捉
X	0	1	0	0	0	X	用 CCPn 的负沿触发捕捉
X	1	1	0	0	0	X	用 CCPn 的跳变触发捕捉
1	0	0	1	0	0	X	软件定时器
1	0	0	1	1	0	X	高速输出
1	0	0	X	0	1	0	8 位脉冲宽度调制器

15.3 PCA 模块与其他模块互连及控制

15.3.1 ECI 互连

ECI 输入可以是外部通过 IOMUX 选择不同的输入端口，也可以是内部 VC 的滤波输出。VC 输出控制寄存器在 VC 控制模块。

15.3.2 PCACAP0

通道 0 的捕获输入可以是：

- 外部的 IOMUX 的输入端口,外部 UART 的 RX 的 MUX 输入
- 内部的 VC 的滤波后的输出 UART 选择控制在 PCA 捕获通道控制寄存器 SYSCON_PCACR 中，VC 输出控制寄存器在 VC 控制模块。

15.3.3 PCACAP[4:1]

通道 1，2，3，4 的捕获输入可以是：

- 外部的 IOMUX 的输入端口，外部 UART 的 RX 的 MUX 输入

UART 选择控制 UART 选择控制在 PCA 捕获通道控制寄存器 SYSCON_PCACR 中。

15.4 PCA 寄存器列表

基地址 0x40001400

表 15.2. PCA 寄存器列表和复位值

偏移地址	名称	描述	复位值
0X000	PCA_CR	PCA 控制寄存器	0x00000000
0X004	PCA_MOD	PCA 模式寄存器	0x00000000
0X008	PCA_CNT	PCA 计数寄存器	0x00000000
0X00C	PCA_INTCLR	PCA 中断清除寄存器	0x0000009F
0x010	PCA_CCAPM0	PCA 比较/捕获模块 0 模式寄存器	0x00000000
0x014	PCA_CCAPM1	PCA 比较/捕获模块 1 模式寄存器	0x00000000
0x018	PCA_CCAPM2	PCA 比较/捕获模块 2 模式寄存器	0x00000000
0x01C	PCA_CCAPM3	PCA 比较/捕获模块 3 模式寄存器	0x00000000
0x020	PCA_CCAPM4	PCA 比较/捕获模块 4 模式寄存器	0x00000000
0X030	PCA_CCAP0L	PCA 比较/捕获模块 0 低 8 位寄存器	0x00000000

偏移地址	名称	描述	复位值
0X034	PCA_CCAP0H	PCA 比较/捕获模块 0 高 8 位寄存器	0x00000000
0X038	PCA_CCAP1L	PCA 比较/捕获模块 1 低 8 位寄存器	0x00000000
0X03c	PCA_CCAP1H	PCA 比较/捕获模块 1 高 8 位寄存器	0x00000000
0X040	PCA_CCAP2L	PCA 比较/捕获模块 2 低 8 位寄存器	0x00000000
0X044	PCA_CCAP2H	PCA 比较/捕获模块 2 高 8 位寄存器	0x00000000
0X048	PCA_CCAP3L	PCA 比较/捕获模块 3 低 8 位寄存器	0x00000000
0X04c	PCA_CCAP3H	PCA 比较/捕获模块 3 高 8 位寄存器	0x00000000
0X050	PCA_CCAP4L	PCA 比较/捕获模块 4 低 8 位寄存器	0x00000000
0X054	PCA_CCAP4H	PCA 比较/捕获模块 4 高 8 位寄存器	0x00000000
0X058	PCA_CCAPO	PCAPWM 与高速输出标志寄存器	0x00000000
0X05C	PCA_POCR	PCA 端子输出控制寄存器	0x00000000
0X060	PCA_CCAP0	PCA 比较/捕获模块 0 的 16 位寄存器	0x00000000
0X064	PCA_CCAP1	PCA 比较/捕获模块 1 的 16 位寄存器	0x00000000
0X068	PCA_CCAP2	PCA 比较/捕获模块 2 的 16 位寄存器	0x00000000
0X06C	PCA_CCAP3	PCA 比较/捕获模块 3 的 16 位寄存器	0x00000000
0X070	PCA_CCAP4	PCA 比较/捕获模块 4 的 16 位寄存器	0x00000000

15.5 寄存器说明

15.5.1 控制寄存器(PCA_CR)

偏移地址：0x000

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								CF	CR	保留	CCF4	CCF3	CCF2	CCF1	CCF0
								RO	R/W		RO	RO	RO	RO	RO

位	标记	功能描述	复位值	读写
31:8	—	保留位。	0x0	—
7	CF	PCA 计数器溢出标志： 0：无溢出；	0	RO

位	标记	功能描述	复位值	读写
		1: 发生计数器溢出。 当 PCA 计数溢出时, CF 由硬件置位; 当 MOD 寄存器的 CFIE 位为 1, CF 标志可以产生中断。		
6	CR	PCA 计数器运行控制位 0: 关闭 PCA 计数器计数 1: 启动 PCA 计数器计数	0	R/W
5	-	保留位	0	-
4	CCF4	PCA 计数器模块 4 比较/捕获标志位: 当出现匹配或捕获时, 该位由硬件置位; 当 CCAPM4.CCIE 置位时, 这个标志位会产生一个 PCA 中断。 0: 无匹配或捕获; 1: 匹配或捕获发生。	0	RO
3	CCF3	PCA 计数器模块 3 比较/捕获标志位: 当出现匹配或捕获时, 该位由硬件置位; 当 CCAPM3.CCIE 置位时, 这个标志位会产生一个 PCA 中断。 0: 无匹配或捕获; 1: 匹配或捕获发生。	0	RO
2	CCF2	PCA 计数器模块 2 比较/捕获标志位: 当出现匹配或捕获时, 该位由硬件置位; 当 CCAPM2.CCIE 置位时, 这个标志位会产生一个 PCA 中断。 0: 无匹配或捕获; 1: 匹配或捕获发生。	0	RO
1	CCF1	PCA 计数器模块 1 比较/捕获标志位: 当出现匹配或捕获时, 该位由硬件置位; 当 CCAPM1.CCIE 置位时, 这个标志位会产生一个 PCA 中断。 0: 无匹配或捕获; 1: 匹配或捕获发生。	0	RO
0	CCF0	PCA 计数器模块 0 比较/捕获标志位: 当出现匹配或捕获时, 该位由硬件置位; 当 CCAPM0.CCIE 置位时, 这个标志位会产生一个 PCA 中断。 0: 无匹配或捕获; 1: 匹配或捕获发生。	0	RO

15.5.2 模式寄存器(PCA_CM0D)

偏移地址: 0x004

复位值: 0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								CIDL	保留			CPS		CFIE	
								R/W				R/W		R/W	

位	标记	功能描述	复位值	读写
31:8	—	保留位。	0x0	—
7	CIDL	空闲模式 IDLE 下, PCA 是否停止工作 0: 休眠模式(Sleep)下, PCA 继续工作 1: 休眠模式(Sleep)下, PCA 停止工作	0	R/W
6:4	-	保留位。	0x0	-
3:1	CPS	时钟分频选择及时钟源选择 000: PCLK/32 001: PCLK/16 010: PCLK/8 011: PCLK/4 100: PCLK/2 101: Timer0 overflow 110: Timer1 overflow 111: ECI 外部时钟, 时钟 PCLK 四分频采样	0	R/W
0	CFIE	PCA 计数器中断使能控制信号 0: 关闭中断 1: 使能中断	0	R/W

15.5.3 计数寄存器(PCA_CNT)

偏移地址: 0x008

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
R/W															

位	标记	功能描述	复位值	读写
31:16	Reserved	保留位。	0x0	—
15:0	CNT	定时器计数器的值: 只有在 PCA 停止状态, CNT 才可以写入, 否则写入无效。	0x0	R/W

15.5.4 中断清除寄存器(PCA_INTCLR)

偏移地址：0x00C

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								CF	保留		CCF4	CCF3	CCF2	CCF1	CCF0
								WO			WO	WO	WO	WO	WO

位	标记	功能描述	复位值	读写
31:8	—	保留位。	0x0	—
7	CF	PCA 计数器溢出标志清除： 0：写 0 无效； 1：软件写 1 清零对应的标志。	0	WO
6:5	-	保留位。	0x0	-
4	CCF4	PCA 计数器模块 4 比较/捕获标志位清除： 0：写 0 无效； 1：软件写 1 清零对应的标志。	0	WO
3	CCF3	PCA 计数器模块 3 比较/捕获标志位清除： 0：写 0 无效； 1：软件写 1 清零对应的标志。	0	WO
2	CCF2	PCA 计数器模块 2 比较/捕获标志位清除： 0：写 0 无效； 1：软件写 1 清零对应的标志。	0	WO
1	CCF1	PCA 计数器模块 1 比较/捕获标志位清除： 0：写 0 无效； 1：软件写 1 清零对应的标志。	0	WO
0	CCF0	PCA 计数器模块 0 比较/捕获标志位清除： 0：写 0 无效； 1：软件写 1 清零对应的标志。	0	WO

15.5.5 比较捕获模式寄存器(PCA_CCAPM0~4)

偏移地址：

CCAPM0: 0x010; CCAPM1: 0x014; CCAPM2: 0x018; CCAPM3: 0x01C; CCAPM4: 0x020

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									ECOM	CAPP	CAPN	MAT	TOG	PWM	CCIE
									R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:7	—	保留位。	0x0	—
6	ECOM	允许比较器功能控制位： 0：禁止比较强功能； 1：允许比较器功能。 当 PCA 用于软件计数器，高速输出，PWM 模式，要置位 ECOM； 当写 CCAMPHx 或 CCAMPx 寄存器会自动置位 ECOM； 当写 CCAMPLx 寄存器会自动清除 ECOM 位。	0	R/W
5	CAPP	正沿捕获控制位： 0：禁止上升沿捕获； 1：允许上升沿捕获。	0	R/W
4	CAPN	负沿捕获控制位 0：禁止下降沿捕获； 1：允许下降沿捕获。	0	R/W
3	MAT	允许匹配控制位： 0：禁止匹配功能； 1：PCA 计数值与模块的比较/捕获寄存器的值一旦匹配，将置位 CR 寄存的中断标志 CCFx(x=0-4)。	0	R/W
2	TOG	翻转控制位： 0：禁止翻转功能； 1：工作在 PCA 高速输出模式，PCA 计数器的值与模块的比较/捕获寄存器的值一旦匹配，CCPx 引脚翻转。	0	R/W
1	PWM	脉宽调制控制位： 0：禁止 PWM 脉宽调制功能； 1：允许 CCPx 引脚作为 PWM 输出只有 CCAPMx[6:0]=100_0010 时，PWM 功能才有效。	0	R/W
0	CCIE	PCA 使能中断： 0：PCA 比较/捕获功能中断禁止； 1：使能比较/捕获中断。	0	R/W

15.5.6 比较捕获数据寄存器高 8 位(PCA_CCAP0~4H)

偏移地址：

CCAP0H: 0x034; CCAP1H: 0x03C; CCAP2H: 0x044; CCAP3H: 0x04C; CCAP4H: 0x054

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:8	—	保留位。	0x0	—
7:0	CCAPx[15:8]	比较/捕获模式高 8 位寄存器： 当 PCA 模式用于比较/捕获模式时，用于保存 16 位捕获计数值的高 8 位，写 CCAPxH 寄存器会自动置位寄存器 CCAPMx 的 ECOM 位。 当 PCA 模式用于 PWM 模式时，用于控制输出占空比装载寄存器，在计数器低 8 位溢出时，装载寄存器会自动更新到 PWM 比较寄存器	0x0	R/W

15.5.7 比较捕获数据寄存器低 8 位(PCA_CCAP0~4L)

偏移地址：

CCAP0L: 0x030; CCAP1L: 0x038; CCAP2L: 0x040; CCAP3L: 0x048; CCAP4L: 0x050

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:8	—	保留位。	0x0	—
7:0	CCAPx[7:0]	比较/捕获模式低 8 位寄存器： 当 PCA 模式用于比较/捕获模式时，用于保存 16 位捕获计数值的低 8 位；写 CCAPxH 寄存器会自动清除寄存器 CCAPMx 的 ECOM 位。 当 PCA 模式用于 PWM 模式时，用于控制输出占空比比较寄存器，在 PWM 模式，计数器的低 8 位的值小于	0x0	R/W

位	标记	功能描述	复位值	读写
		CCAPx[7:0]的值 PWM 输出低电平，否则 PWM 输出高电平。		

15.5.8 比较捕获 16 位寄存器(PCA_CCAP0~4)

偏移地址：

CCAP0: 0x060; CCAP1: 0x064; CCAP2: 0x068; CCAP3: 0x06C; CCAP4: 0x070

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCAPx															
R/W															

位	标记	功能描述	复位值	读写
31:16	—	保留位。	0x0	—
15:0	CCAPx	比较/捕获模式 16 位寄存器： 当 PCA 用于比较/捕获模式时，用于保存 16 位捕获计数值； 写 CCAPx 寄存器会置位寄存器 CCAPMx 的 ECOM 位。写 CCAPX 寄存器相当于写 CCAPxL 及 CCAPxH 这两个 8 位寄存器。在比较/捕获模式下可以直接读写这个寄存器，在 PWM 模式下，使用 CCAPxL 及 CCAPxH 寄存器	0x0	R/W

15.5.9 比较高速输出标志寄存器(PCA_CCAPO)

偏移地址: 0x058

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											CCA PO4	CCA PO3	CCA PO2	CCA PO1	CCA PO0
											R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:5	—	保留位。	0x0	—
4	CCAPO4	高速模式比较模块 4 的输出值 0: 输出 0 1: 输出 1	0	R/W
3	CCAPO3	高速模式比较模块 3 的输出值 0: 输出 0 1: 输出 1	0	R/W
2	CCAPO2	高速模式比较模块 2 的输出值 0: 输出 0 1: 输出 1	0	R/W
1	CCAPO1	高速模式比较模块 1 的输出值 0: 输出 0 1: 输出 1	0	R/W
0	CCAPO0	高速模式比较模块 0 的输出值 0: 输出 0 1: 输出 1	0	R/W

15.5.10 端子输出控制寄存器(PCA_POCR)

偏移地址: 0x05C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			POI NV4	POI NV3	POI NV2	POI NV1	POI NV0	保留			POE4	POE3	POE2	POE1	POE0
			R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W	R/W	

位	标记	功能描述	复位值	读写
31:13	—	保留位。	0x0	—
12	POINV4	比较通道 4 的输出极性反转 0 =禁止 PWM0 输出极性反转 1 =使能 PWM0 输出极性反转	0	R/W
11	POINV3	比较通道 3 的输出极性反转 0 =禁止 PWM0 输出极性反转 1 =使能 PWM0 输出极性反转	0	R/W
10	POINV2	比较通道 2 的输出极性反转 0 =禁止 PWM0 输出极性反转 1 =使能 PWM0 输出极性反转	0	R/W

位	标记	功能描述	复位值	读写
9	POINV1	比较通道 1 的输出极性反转 =禁止 PWM0 输出极性反转 =使能 PWM0 输出极性反转	0	R/W
8	POINV0	比较通道 0 的输出极性反转 =禁止 PWM0 输出极性反转 =使能 PWM0 输出极性反转	0	R/W
7:5	-	保留位。	0x0	-
4	POE4	比较通道 4 的输出使能 0: 输出禁止 1: 输出使能	0	R/W
3	POE3	比较通道 3 的输出使能 0: 输出禁止 1: 输出使能	0	R/W
2	POE2	比较通道 2 的输出使能 0: 输出禁止 1: 输出使能	0	R/W
1	POE1	比较通道 1 的输出使能 0: 输出禁止 1: 输出使能	0	R/W
0	POE0	比较通道 0 的输出使能 0: 输出禁止 1: 输出使能	0	R/W

16. Base Timer (TIM10,TIM11)

16.1 Base Timer 简介

Base Timer 包含两个定时器 TIM10/11。TIM10/11 功能完全相同。TIM10/11 是同步定时/计数器，可以作为 16/32 位自动重载功能的定时/计数器，也可以作为 16/32 位无重载功能的定时/计数器。TIM10/11 可以对外部脉冲进行计数或者实现系统定时。

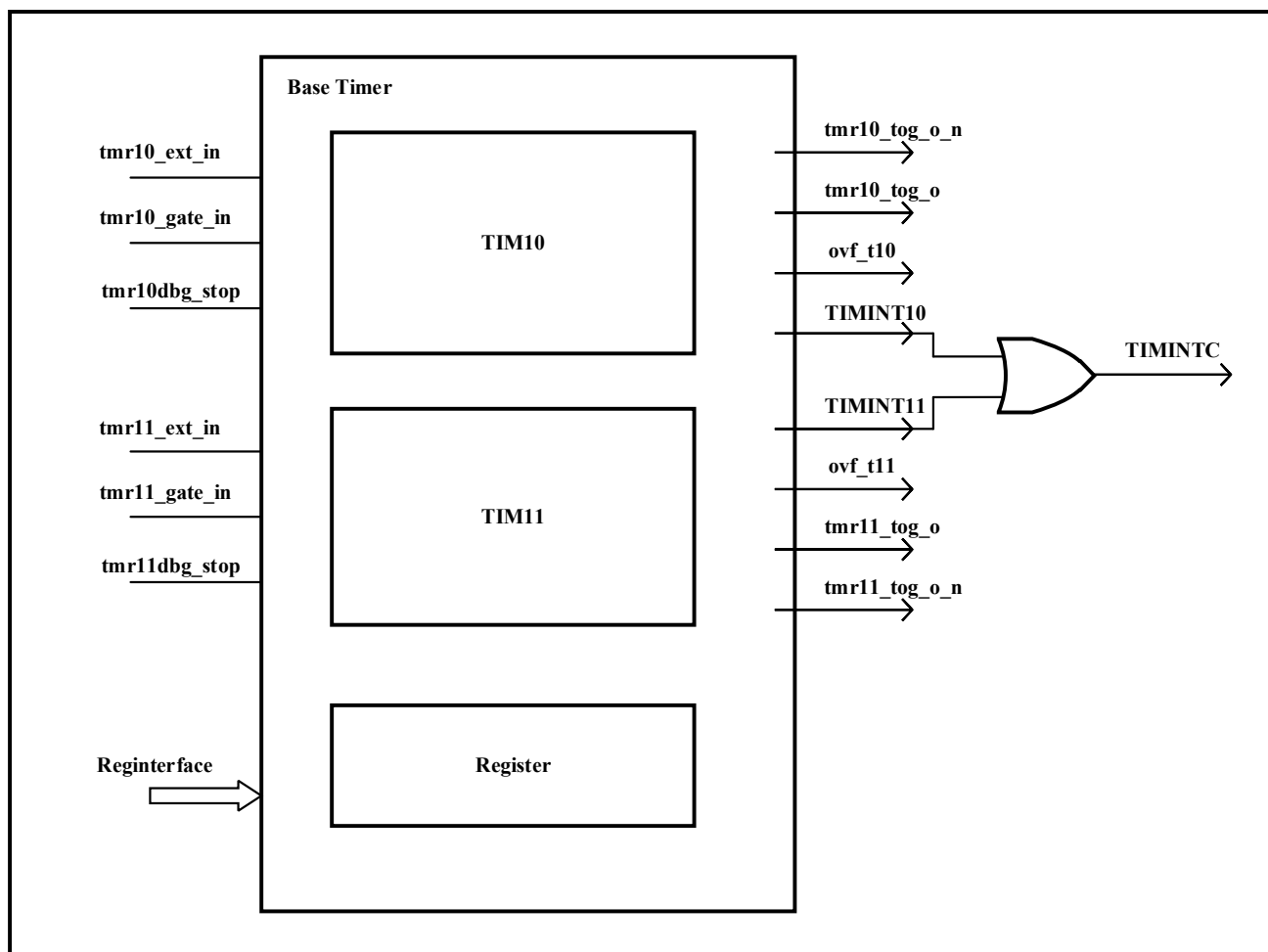


图 16-1. Base Timer 框图

16.2 Base Timer 特性

- ◆ 16/32 位同步定时/计数器
- ◆ 16/32 位自动重载功能

◆ 可以对外部脉冲进行计数或者实现系统定时

16.3 BaseTimer 功能描述

TIM10/11 每个定时/计数器都有独立的控制启动信号，以及外部输入时钟，门控信号。TIM10/11 使用 EXT, GATE 来进行计数功能，EXT 用于计数器的外部输入时钟信号，GATE 用于有效电平计数使能信号。当门控功能使能后，当且仅当外部输入 GATE 电平有效时，计数器才会计数，否则计数器处于保持状态。门控使能使用 TIMx_CR.GATE_EN 控制。默认门控功能关闭。门控电平选择使用 TIMx_CR.GATE_P 控制。默认高电平为门控有效电平；设置 TIMx_CR.GATE_P 为 1 后，门控低电平为有效电平。

TIM10/11 使用 PCLK, GATE 来进行定时功能，PCLK 用于定时器的内部输入时钟信号，GATE 可用于有效电平定时使能信号。当门控功能使能后，当且仅当外部输入 GATE 电平有效时，定时器才会计数，否则定时器处于定时计数器停止状态。门控使能使用 TIMx_CR.GATE_EN 控制。默认门控功能关闭。门控电平选择使用 TIMx_CR.GATE_P 控制。默认高电平为门控有效电平；设置为 1 后门控有效电平是低电平。定时功能可以配置预除频。TIMx_CR.TMR_PRE 控制分频比。

TMR_PRE[2:0]	000	001	010	011	100	101	110	111
分频比	1	2	4	8	16	32	64	128

TIM10/11 支持定时/计数器两种功能，可通过设置定时器控制寄存器(TIMx_CR)中 CT_SEL 进行配置。每种功能支持 2 种模式，模式 1 为 16/32 位自由计数模式，模式 2 是 16/32 位重载模式。

A) 模式 1 自由计数模式

计数到最大值(16 位 Max=0xFFFF, 32 位最大值为 0xFFFFFFFF)溢出后产生中断，定时/计数器清零，然后继续计数；

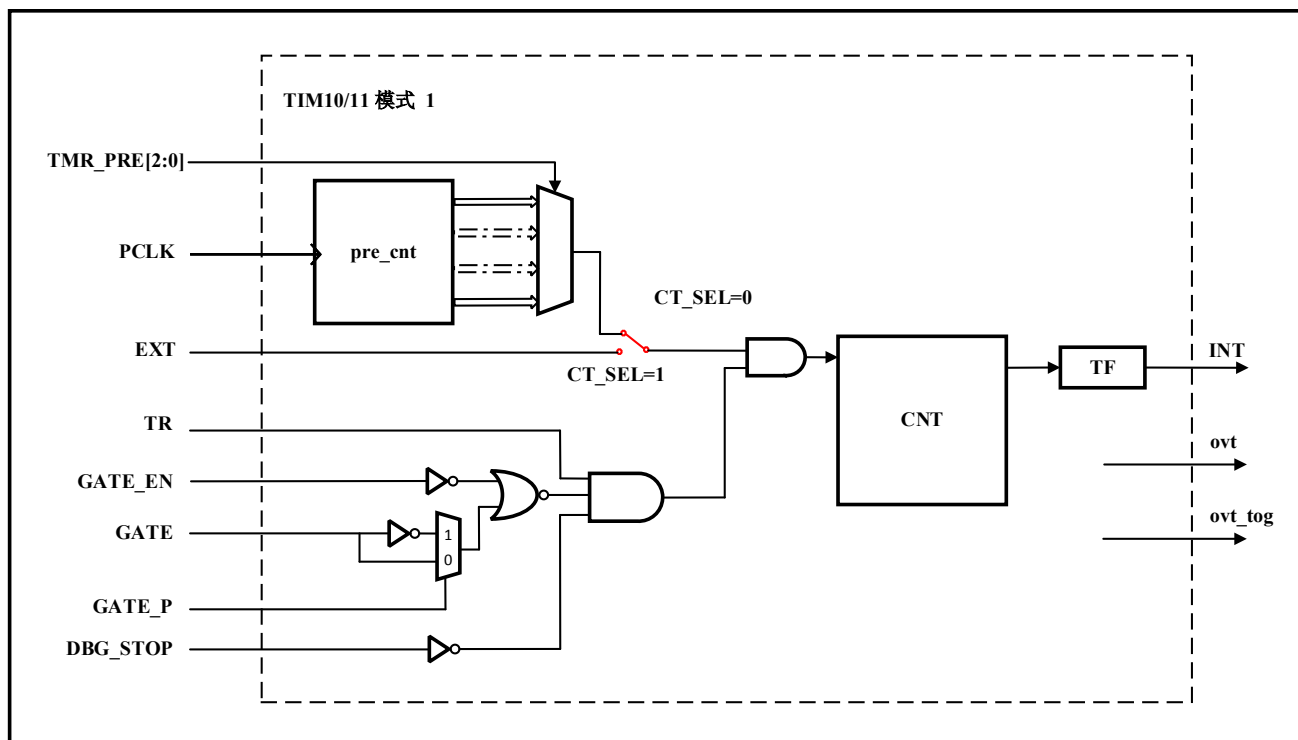


图 16-2. Timer 模式 1 框图

B) 模式 2 重载模式

重载模式，计数到最大值后溢出，产生中断，定时/计数器的值被装载为 BGLOAD 的值，然后继续向上计数。在重载模式下，定时时间设的小的情况下需要考虑软件处理速度，否则中断会来不及处理而造成中断丢失。

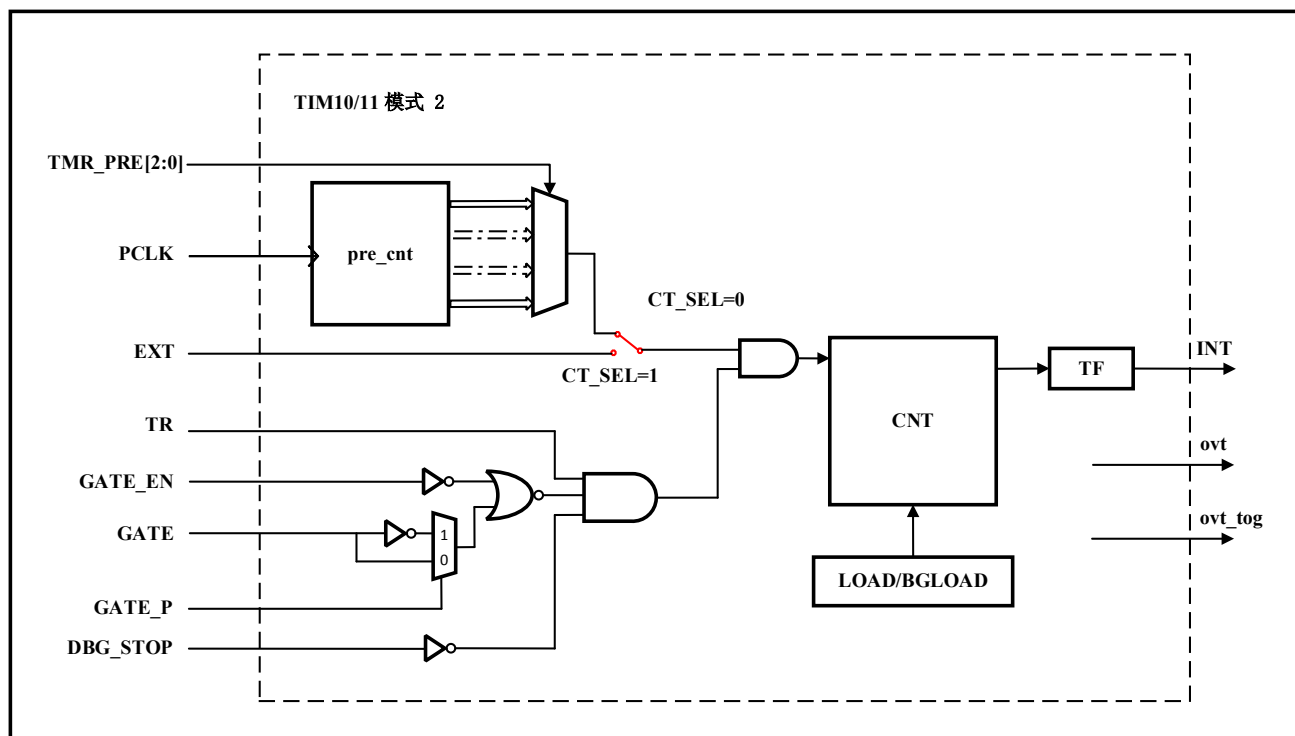


图 16-3. Timer 模式 2 框图

当设置对应定时器 TIMx_CR.TR 为 1 后定时器开始运行。模式 1，启动后从寄存器设定的初值开始计数，计数到最大后产生溢出中断，然后从 0 继续计数。模式 2，启动后从寄存器初值 CNT 开始向上计数，计数到最大值后产生中断后重载寄存器 BGLOAD 的值到计数器 CNT 中，继续向上计数。不论是自由计数模式，还是重载模式，只要是写 LOAD 值，都会立即更新定时/计数器的值，然后继续向上计数。

16.3.1 计数功能

计数功能用于测定某个事件发生的次数。在计数功能中，计数器在每个相应的输入时钟的下降沿累加一次。输入信号被内部的 PCLK 采样，因此外部输入时钟频率不能超过系统的 PCLK 时钟。计数到最大值会溢出并且产生中断。中断标志需要软件清除。

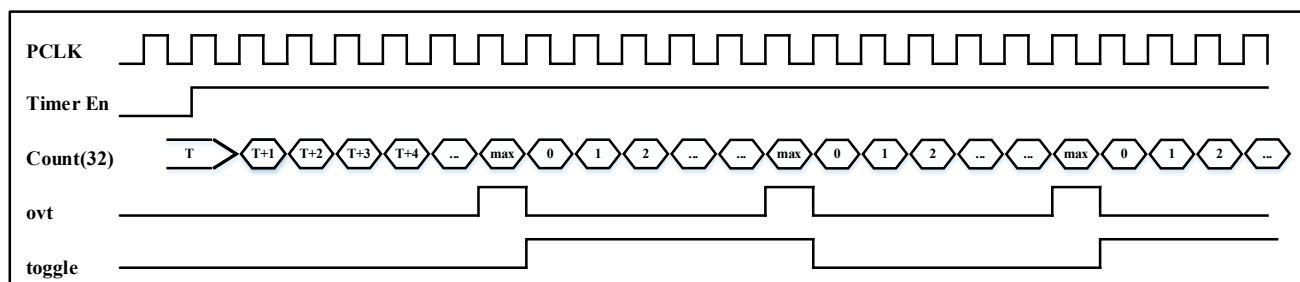


图 16-4.32 位模式 1 时序图(max=0xFFFFFFFF)

16.3.2 定时功能

定时功能用于产生间隔定时。在定时功能中，定时器有预分频，定时器在每个预分频的一个时钟累加一次，计数到最大值会溢出并且产生中断。中断标志需要软件清除。

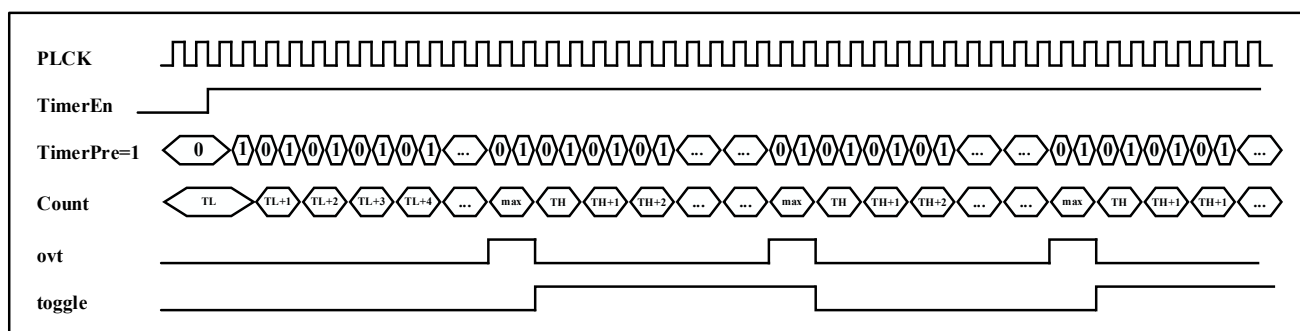


图 16-5.32 位模式 2 时序图(PCLK 二分频,max=0xFFFFFFFF)

16.3.3 Buzzer 功能

通过定时器的翻转输出功能可以实现驱动 Buzzer 的功能。TIMx_CR.TOG_EN 为 1 时，TOG，TOGN 输出反向。设置 TIMx_CR.TOG_EN 为 0 可以同时设置端口 TOG，TOGN 输出为 0。在计数时钟为 4M 情况下 Buzzer 输出不同频率的 Timer 重载模式配置如下：

Buzzer 频率	计数周期	计数值	重载值	CNT 初始值	LOAD 重载值
1KHz	0.5ms	2000	63536	0xF830	0xF830
2KHz	0.25ms	1000	64536	0xFC18	0xFC18
4KHz	0.125ms	500	65036	0xFE0C	0xFE0C

16.4 Base Timer 互联

16.4.1 GATE 互联

GATE 输入可以从端口直接输入，也可以输入 UART/LPUART 的 RX 信号；还可以配置为 VC 的输出作为 GATE 信号。TIM10/11 的 GATE 都可以配置。通过内部互联配置，可以实现

UART 波特率的自动识别，可以测量 VC 比较输出的脉冲宽度，可以实现外部控制计数。配置选择 RX 输入在 SYSCON_PORTCR 寄存器控制，VC 控制在 VC 控制寄存器控制。端口选择，UART/LPUART 输入选择与 VC 输出选择作为门控输入只能选择一个有效。VC 的输出选择优先级最高。

16.4.2 Toggle 输出互联

TIM10 的翻转输出 tmr10_tog_o 到内部模块 UART0，控制 UART0 的波特率；TIM11 的翻转输出 tmr11_tog_o 到内部模块 UART1，控制 UART1 的波特率；TIM10/11 的翻转输出还输出到端口上，可以驱动 Buzzer 实现蜂鸣器的控制。

16.5 Base Timer 寄存器列表

x=10,11

BaseTimer 基地址：0X40001800

	偏移地址	描述
TIM10	0x00	TIM10 偏移地址
TIM11	0x100	TIM11 偏移地址

表 16.1. PCA 寄存器列表和复位值

偏移地址	名称	描述	复位值
0x00	TIMx_CR	控制寄存器	0x00000000
0x04	TIMx_LOAD	32 位立即重载寄存器	0x00000000
0x08	TIMx_CNT	读计数器寄存器，只读	0x00000000
0x0C	TIMx_RAWINTSR	读原始中断寄存器，	0x00000000
0x10	TIMx_MSKINTSR	读中断寄存器	0x00000000
0x14	TIMx_INTCLR	中断清除寄存器	0x00000000
0x18	TIMx_BGLOAD	32 位周期重载寄存器	0x00000000

16.6 Base Timer 寄存器说明

16.6.1 控制寄存器(TIMx_CR)

偏移地址：0x00

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				GAT E_P	GATE _EN	TOG _EN	CT_ SEL	TR	MO DE	INT EN	TMR _SIZE	ONE SHOT	TMR_PRSC [2:0]		
				R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

位	符号	描述	复位值	读写
31:12	保留	保留位，读为 0	0x0	—
11	GATE_P	端口 GATE 极性控制，默认高电平 gate 有效，设置为 1 后低电平有效 0：高电平有效 1：低电平有效	0	R/W
10	GATE_EN	定时器门控 0：无门控，TR=1 时定时器工作 1：只有端口 GATE 有效并且 TR=1 时才工作	0	R/W
9	TOG_EN	TOG 输出使能 0：TOG,TOGN 同时输出 0 1：TOG,TOGN 输出相位相反的信号。可供 buzzer 使用。	0	R/W
8	CT_SEL	计数器/定时器功能选择 0：定时器功能，定时器由 PCLK 来进行计数。 1：计数器功能，计数器由外部输入的下降沿进行计数。外部输入由 PCLK 采样，外部输入时钟频率要低于 1/2 采样时钟。	0	R/W
7	TR	定时器运行控制 0：定时器停止 1：定时器运行	0	R/W
6	MODE	定时器工作模式 0：模式 1 计数器/定时器 1：模式 2 自动重装载计数器/定时器	0	R/W
5	INTEN	中断使能控制，写 1 后使能中断	0	R/W
4	TMR_SIZE	Timer Size=0, max count value=0xFFFF Timer Size=1, max count value=0xFFFFFFFF	0	R/W
3	ONESHOT	计数器运行一次使能 0：重复模式 1：oneshot 模式	0	R/W

位	符号	描述	复位值	读写
2:0	TMR_PRSC [2:0]	TIM 预除频选择 000: 分频数 1 001: 分频数 2 010: 分频数 4 011: 分频数 8 100: 分频数 16 101: 分频数 32 110: 分频数 64 111: 分频数 128	0x0	R/W

16.6.2 立即重载寄存器(TIMx_LOAD)

偏移地址: 0x04

复位值: 0x00000000

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

位	符号	描述	复位值	读写
31:0	LOAD[31:0]	立即重载寄存器 写此寄存器, 会立即更新 Count 计数器的值 读 0x4, 0x18, 读到最近更新的 Load 或者 BGLOAD 寄存器	0x0	R/W

16.6.3 计数器寄存器(TIMx_CNT)

偏移地址: 0x08

复位值: 0x00000000

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

位	符号	描述	复位值	读写
31:0	CNT[31:0]	计数器寄存器	0x0	RO

16.6.4 原始中断状态寄存器(TIMx_RAWINTSR)

偏移地址: 0x0c

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:1	保留	保留位, 读为 0	0x0	—
0	RIS	中断标志, 硬件置位; 论 Int Enable=0 or 1, 都可以读中断	0	RO

16.6.5 中断标志寄存器(TIMx_MSKINTSR)

偏移地址: 0x10

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:1	保留	保留位, 读为 0	0x0	—
0	TF	INTEN=1, 才可以读中断寄存器	0	RO

16.6.6 中断清除寄存器(TIMx_INTCLR)

偏移地址：0x14

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:1	保留	保留位，读为 0	0x0	—
0	INTCLR	中断标志清除，写 1 清除，写 0 无效	0	WO

16.6.7 周期重载寄存器(TIMx_BGLOAD)

偏移地址：0x018

复位值：0x00000000

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

位	符号	描述	复位值	读写
31:0	BGLOAD[31:0]	BACKGROUND 周期重载寄存器，写此寄存器不会立即更新 Count 计数器的值。只有当 Count 值溢出时，才会重新装载 BGLOAD 的值到 Count 计数器里面。 读 0x4，0x18，读到为最近更新的 Load 或者 BGLOAD 寄存器的值。	0x0	R/W

17. 低功耗定时器(LPTIM)

LPTIM 是异步 16 位定时/计数器，在系统时钟关闭后仍然可以通过内部低速 RC 或者外部低速晶体振荡计时/计数。通过中断在低功耗模式下唤醒系统。

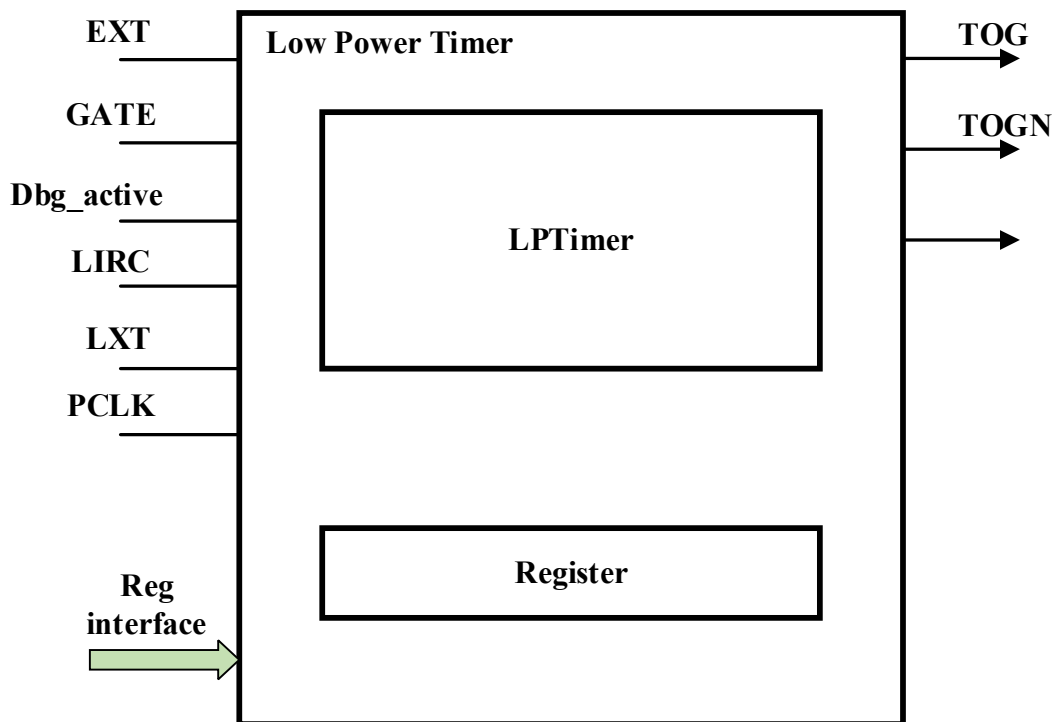


图 17-1. LPTIMER 结构框图

17.1 LPTIM 功能描述

LPTIM 具有独立的控制启动信号，以及外部输入时钟，门控信号。

LPTIM 使用 EXT，GATE 来进行计数功能，EXT 用于计数器的外部输入时钟信号，GATE 为有效电平计数使能信号。

LPTIM 的定时器支持两种工作模式，通过设置定时器控制寄存器(LPTIM_CR)中 MODE 选择工作模式。

模式 1 为 16 位自由计数模式，计数器从 LPTIM_LOAD 设定的值开始计数，溢出后计数值从 0x0000 开始重新计数。

模式 2 为 16 位重载模式，LPTIM 启动时会自动装载重载寄存器 LPTIM_LOAD 的值到计

数器中，当溢出后会自动装载 Period LOAD 的值到计数器中。用户写 LPTIM_LOAD 或者 LPTIM_BGLOAD 都会更新 Period LOAD 的值，PeriodLOAD 的值为用户近一次跟新 LPTIM_LOAD 或 LPTIM_BGLOAD 的值。

LPTIM 可选三种时钟作为定时器时钟，通过控制寄存器 LPTIM_CR.TCK_SEL 来选择。默认选择 PCLK。时钟选择如表：

TCK_SEL	00	01	10	11
定时器时钟	PCLK	PCLK	LXT	LIRC
读定时器计数值	读经过同步	无同步	读经过同步	读经过同步

当选择相应的时钟源，然后设定 TCK_EN 为 1 后可以打开计数器的计数时钟源。

当时钟源选择并打开后，设置对应定时器的 TIM_RUN 为 1 后定时器开始运行。

对于模式 1 和模式 2，如果设置 LOAD 值，任何时刻，计数值会被立即更新为 LOAD 值，计数器从 LOAD 值开始重新计数。设置 LOAD 值的优先级高于其他计数器被更新的优先级。

在模式 2，设定 BGLOAD 的值，只有在计数器发生溢出后，设定值才会被更新到计数器。

模式 1：如果没有设置 LOAD 值，计数器从 0 开始计数，计数到最大 0xFFFF 后产生溢出中断。计数器计数到最大 0xFFFF 后，计数器再次从 0 开始计数。

模式 2：如果没有设置 LOAD 值，计数器从 0 开始计数，计数到最大 0xFFFF 后产生溢出中断。计数器计数到最大 0xFFFF 后，计数值会被更新为 PeriodLOAD 的值，然后向上计数。

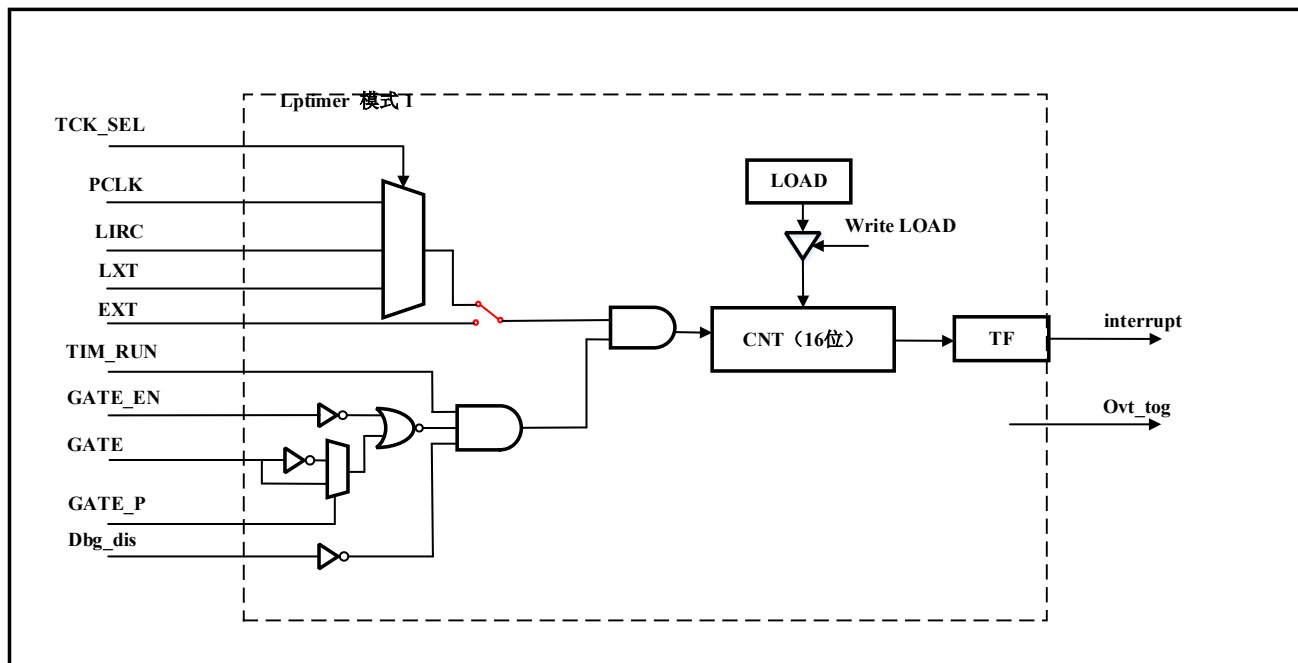


图 17-2. LPTIMER 模式 1

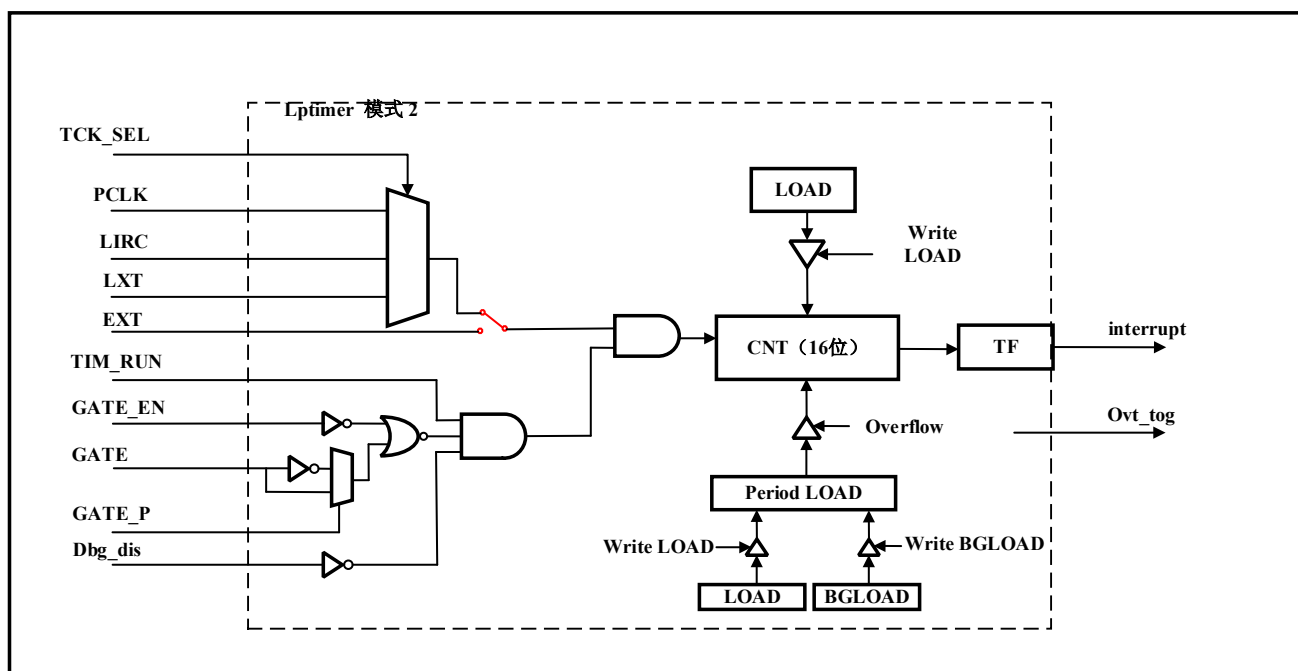


图 17-3. LPTIMER 模式 2

17.1.1 计数功能

计数功能用于测定某个事件发生的次数。在计数功能中，计数器在每个相应的输入时钟的上升沿累加一次。输入信号被内部的计数时钟采样，因此外部输入时钟频率不能超过系统的计数时钟。计数到大值会溢出并且产生中断。

定时功能

定时功能用于产生间隔定时。在定时功能中，定时器一个时钟累加一次，计数到大值会溢出并且产生中断。

17.2 LPTIM 互联

17.2.1 GATE 互联

GATE 输入可以从端口直接输入，也可以输入 UART 的 RX 信号；

通过内部互联配置，可以实现 UART 波特率的自动识别，可以测量 VC 比较输出的脉冲宽度，可以实现外部控制计数。

配置选择 RX 输入在 SYSCON_PORTCR 寄存器控制，VC 控制在 VC 控制寄存器控制。

17.2.2 EXT 互联

EXT 输入可以从端口直接输入，也配置为 VC 的输入作为 EXT 信号。

通过内部互联配置，可以测量 VC 脉冲计数。VC 输出控制寄存器在 VC 控制模块。

17.2.3 TOGGLE 输出互联

LPTIM 的翻转输出到端口上，可以驱动 BUZZER 实现蜂鸣器的控制。

17.3 寄存器列表

基地址：0X40004400

表 17.1. LPTIM 寄存器列表和复位值

偏移地址	名称	描述	复位值
0x00	LPTIM_CNTVAL	LPTIM 计数值只读寄存器	0x00000000
0x04	LPTIM_CR	LPTIM 控制寄存器	0x00000000
0x08	LPTIM_LOAD	LPTIM 立即重载寄存器	0x00000000
0x0C	LPTIM_INTSR	LPTIM 中断寄存器	0x00000000
0x10	LPTIM_INTCLR	LPTIM 中断清除寄存器	0x00000000
0x14	LPTIM_BGLOAD	LPTIM 周期重载寄存器	0x00000000

17.4 寄存器说明

17.4.1 LPTIM 计数值只读寄存器(LPTIM_CNTVAL)

地址偏移: 0x00

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:16	—	保留	0x0	—
15:0	LPT_CNT[15:0]	计数值只读寄存器	0x0	RO

17.4.2 LPTIM 控制寄存器(LPTIM_CR)

地址偏移: 0x04

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															WT_FLAG
															RO
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						TCK_EN	INT_EN	GATE_POLE	GATE	TCK_SEL [1:0]	TOG_EN	CT_SEL	MODE	TIMER_RUN	
						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

位	标记	功能描述	复位值	读写
31:17	—	保留	0x0	—
16	WT_FLAG	WT 写同步标志 0: 同步完成, 此时可更改 LOAD/BGLOAD 1: 正在同步, 此时写 LOAD/BGLOAD 无效	0x0	RO
15:10	—	保留	0x0	—
9	TCK_EN	LPTIM 计数时钟使能 0: LPTIM 计数时钟关闭	0	R/W

位	标记	功能描述	复位值	读写
		1: LPTIM 计数时钟使能只有计数时钟使能后才能进行 LOAD/BGLOAD 的配置		
8	INT_EN	中断使能控制, 写 1 后使能中断	0	R/W
7	GATE_P	输入 GATE 的有效极性默认高电平 GATE 有效, 设置为 1 后低电平有效	0	R/W
6	GATE_EN	定时器门控 0: 无门控 1: 有门控	0	R/W
5:4	TCK_SEL[1:0]	LPTIM 时钟选择 00: PCLK 10: LXT 11: LIRC	0x0	R/W
3	TOG_EN	TOG 输出使能 0: TOG, TOGN 同时输出 0 1: TOG, TOGN 输出相位相反的信号。可供 BUZZER 使用。	0	R/W
2	CT_SEL	计数器/定时器功能选择 0: 定时器功能, 定时器使用 TCK_SEL 选择的时钟进行计数。 1: 计数器功能, 计数器使用外部输入的下降沿进行计数。采样时钟使用 TCK_SEL 选择的时钟, 外部输入时钟要低于 1/2 采样时钟。	0	R/W
1	MODE	定时器工作模式 0: 模式 1 无重载模式 16 位计数器/定时器 1: 模式 2 自动重载 16 位计数器/定时器	0	R/W
0	TIM_RUN	定时器运行控制位 0: 定时器停止 1: 定时器运行	0	R/W

17.4.3 LPTIM 立即重载寄存器(LPTIM_LOAD)

地址偏移: 0x08

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOAD															
R/W															

位	标记	功能描述	复位值	读写
31:16	—	保留	0x0	—
15:0	LOAD	即刻重载寄存器 与TIMER是否运行无关，与MODE无关。 写LOAD前需要读取LPTIM_CR.WT_FLAG，当且仅当WT_FLAG为0时，才能写入数据。写LOAD寄存器完成后WT_FLAG会变低。 写该寄存器时会立即更新计数器的值。 读该寄存器时返回最新更新到LOAD或者BGLOAD的值。	0x0	R/W

17.4.4 LPTIM 中断寄存器(LPTIM_INTSR)

地址偏移：0x0C

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:1	—	保留	0x0	—
0	INTF	中断标志： 0：未发生中断 1：发生溢出中断	0x0	RO

17.4.5 LPTIM 中断寄存器(LPTIM_INTCLR)

地址偏移：0x10

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:1	–	保留	0x0	–
0	ICLR	写 1 清除中断标志，写 0 无效	0x0	WO

17.4.6 LPTIM 周期重载寄存器(LPTIM_BGLOAD)

地址偏移：0x14

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:16	–	保留	0x0	–
15:0	BGLOAD[15:0]	BACKGROUND周期重载寄存器 写BGLOAD前需要读取LPTIM_CR.WT_FLAG，当且仅当WT_FLAG为0时，才能写入数据。写BGLOAD寄存器完成后WT_FLAG会变低。 写该寄存器时，当计数器溢出后，设定值会更新到计数器。 读该寄存器时返回最新更新到 LOAD 或者 BGLOAD 的值。	0x0	R/W

18. 自唤醒定时器(AWK)

MS8006 有一个专用的自唤醒定时器(AWK), 为芯片在低功耗模式下提供一个唤醒事件基准。AWK 在 Sleep 或 Deep Sleep 模式保持计数。当 AWK 用作唤醒定时器时, AWK 要在进入省电模式之前开启。AWK 可以配置片内 32KHZ 时钟源 LIRC、外部 32.768K 时钟源 LXT、HXT 分频后的时钟。注意系统时钟频率必须大于 AWK 时钟两倍以上。如果 AWK 开始计数, 在设备进入 Sleep 或 Deep Sleep 模式时, 选择的时钟源会也要保持工作。注意选择的 AWK 时钟源不会连同 AWK 的配置自动使能, 用户应该手动使能选择的时钟源并等待它稳定来确保操作的成功。

AWK 配备了一个简单的 8 位自动重载向上计数定时器。它的预分频可选择从 1/2 到 1/65536, 通过 DIVSEL[3:0](AWK_CR[3:0])来设置。用户填写重载值到 AWK_RLOAD 寄存器来决定它的溢出速率。AWKEN(AWK_CR.4)置位后, 当 CPU 进入 Sleep/Deep Sleep 模式时装载 AWK_RLOAD 寄存器的值到内部 8 位计数器并开始计数。当计数器溢出, AWUF(AWK_SR.0) 置为 1, 唤醒 CPU。

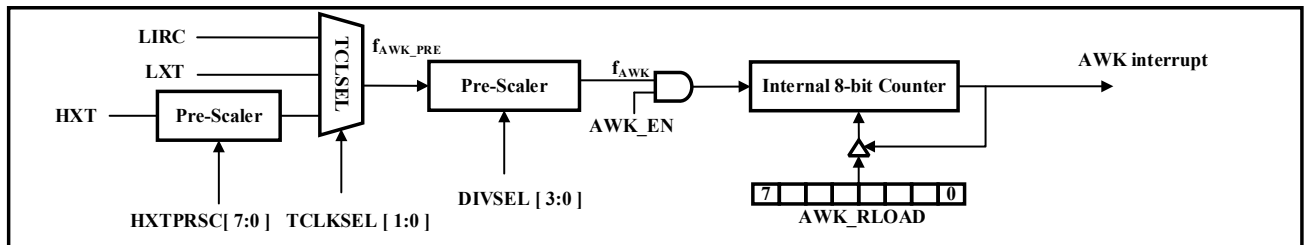


图 18-1. 自唤醒定时器结构图

18.1 寄存器列表

基地址: 0x40001C00

表 18.1. AWK 寄存器列表和复位值

偏移地址	名称	描述	复位值
0x000	AWK_CR	自唤醒定时器控制寄存器	0x0000BC00
0x004	AWK_RLOAD	自唤醒定时器重载数据寄存器	0x00000000
0x008	AWK_SR	自唤醒定时器状态寄存器	0x00000000

偏移地址	名称	描述	复位值
0x00C	AWK_INTCLR	自唤醒中断清除寄存器	0x00000000

18.2 寄存器说明

18.2.1 自唤醒定时器控制寄存器(AWK_CR)

地址偏移：0x00

复位值：0x0000BC00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XTLPRSC[7:0]								保留	TCLKSEL [1:0]		AWK EN	DIVSEL[3:0]			
R/W									R/W		R/W	R/W			

位	标记	功能描述	复位值	读写
31:16	—	保留	0x0	—
15:8	HXTPRSC[7:0]	HXT 时钟分频系数： $F_{HXT}/(N+1)$	0xBC	R/W
7	—	保留	0x0	—
6:5	TCLKSEL[1:0]	AWK 计数时钟源选择 00: 停止 01: LIRC 时钟 10: HXT 分频后的时钟 11: LXT 时钟	0x0	R/W
4	AWKEN	AWK 使能 0: 不使能 1: 使能	0	R/W
3:0	DIVSEL[3:0]	计数器用时钟源选择位 0000: $F_{AWK_PRE}/2^1$ 0001: $F_{AWK_PRE}/2^2$ 0010: $F_{AWK+PRE}/2^3$ 1111: $F_{AWK_PRE}/2^{16}$	0x0	R/W

18.2.2 自唤醒定时器重载数据寄存器(AWK_RLOAD)

地址偏移: 0x04

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7:0	RLDVAL[7:0]	计数器	0x0	R/W

18.2.3 自唤醒定时器状态寄存器(AWK_SR)

地址偏移: 0x08

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:1	—	保留	0x0	—
0	AWUF	自动唤醒发生, 硬件置 1, 软件清零 0: 未发生自动唤醒 1: 计数器溢出, 自动唤醒发生	0x0	RO

18.2.4 自唤醒中断清除寄存器(AWK_INTCLR)

地址偏移: 0x08

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:1	—	保留	0x0	—
0	INTCLR	自动唤醒中断清除 0: 无作用 1: 清除自动唤醒中断	0x0	WO

19. 蜂鸣器 (BEEP)

19.1 简介

选择 LIRC 时钟, HXT 时钟(4~24MHz)或者 PCLK, 可通过分频设定来产生各种频率的蜂鸣信号。

BEEP_CSR.CLKSEL[1:0]位来选择得到 $f_{\text{BEEP_PRE}}$ 时钟, 通过设置 BEEPDIV[11:0]将蜂鸣器时钟 $f_{\text{BEEP_PRE}}$ 分频得到 f_{BEEP} 。通过设置 BEEPSEL[1:0], 得到 $f_{\text{BEEP_O}}$ 蜂鸣信号。

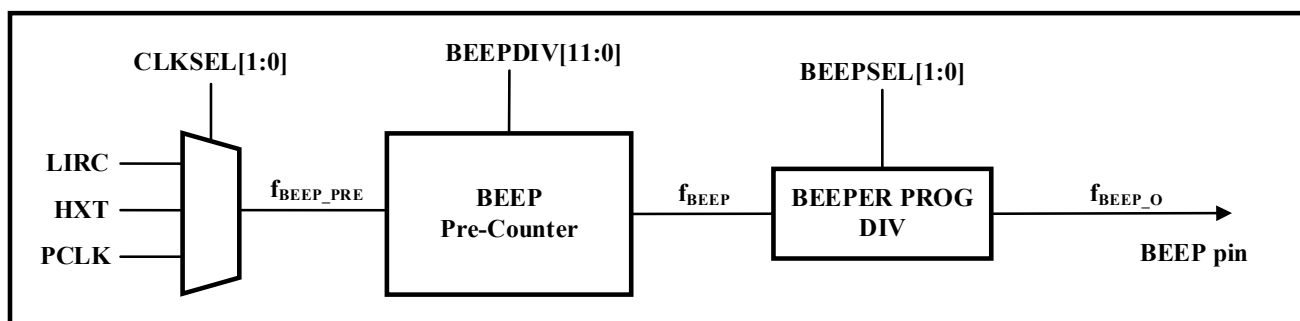


图 19-1. 蜂鸣器功能图

19.2 特性

- ◆ MS8006 内建蜂鸣器信号发生器, 可硬件自动输出 1、2、4kHz 方波。
- ◆ 2 个基础定时器 TIM10/TIM11 可以功能复用输出, 为 Buzzer 提供可编程驱动频率。
- ◆ 可以支持互补输出, 不需要额外的三极管。

19.3 功能描述

19.3.1 蜂鸣器操作

为了使用蜂鸣功能, 按顺序执行如下的步骤:

- 1) 根据需要的蜂鸣器频率输出值确定 BEEPDIV[11:0]的值;
- 2) 通过写 BEEP_CSR 的 BEEPSEL[1:0]位来选择 $f_{\text{BEEP_O}}$ 的输出频率;
- 3) 置位 BEEP_CSR 的 BEEPEN 位来使能时钟源。

注意:

- 预分频计算器仅仅在当 $BEEPDIV[11:0]$ 的值不同于复位值 $0xFFF$ 时才开始运行。
- 在蜂鸣器运行过程中应该保持 $BEEPDIV[11:0]$ 的值不变。

19.3.2 蜂鸣器校准

该步骤可以用来校准 LIRC 时钟以便达到更标准的 f_{BEEP_O} (1kHz, 2kHz 或 4kHz) 频率输出。

采用如下的步骤:

- 1) TIM1, TIM2 或者 CLKTRIM 模块来测量内部的低速 RC 振荡器时钟(LIRC)的时钟频率;
- 2) 采用如下方法计算 BEEPDIV 的值, 这里 A 和 x 是 f_{BEEP_PRE}/f_{BEEP} 的整数和小数部分值:
当 x 小于或者等于 $A/(1+2*A)$ 时, $BEEPDIV=A-1$; 否则 $BEEPDIV=A$ 。
- 3) 将 BEEPDIV 值写入到 BEEP_CSR 的 BEEPDIV[11:0]位。

19.4 寄存器列表

BEEP 基地址: $0x40004800$

表 19.1. BEEP 寄存器列表和复位值

偏移地址	名称	描述	复位值
$0x000$	BEEP_CSR	蜂鸣器控制寄存器	$0x00000FFF$

19.5 寄存器说明

19.5.1 蜂鸣器控制/状态寄存器(BEEP_CSR)

地址偏移: $0x00$

复位值: $0x00000FFF$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										CLKSEL	保留	BEE PEN	BEEPSEL		
										R/W		R/W	R/W		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				BEEPDIV											
				R/W											

位	标记	功能描述	复位值	读写
31:22	—	保留	0x0	—
21:20	CLKSEL	LS 时钟选择. 00: 停止 01: LIRC(38kHz) 10: HXT(4~24mHz) 11: PCLK	0x0	R/W
19	—	保留		
18	BEEPEN	使能蜂鸣	0	R/W
17:16	BEEPSEL	蜂鸣器输出 BEEP_O 频率选择位 00: $f_{BEEP_PRE}/8$ 01: $f_{BEEP_PRE}/4$ 1x: $f_{BEEP_PRE}/2$	0x0	R/W
15:12	—	保留	0x0	—
11:0	BEEPDIV	蜂鸣器预分频器将 LS 时钟分频得到 8kHz 周期的信号分频因子为 BEEPDIV+1 $f_{BEEP}=f_{BEEP_PRE}/(BEEPDIV+1)$	0xffff	R/W

20. 独立看门狗 (IWDG)

20.1 简介

IWDG 的作用是为了防止软件系统在异常情况下，程序执行错误，导致系统异常工作或崩溃；而 IWDG 复位可以帮助系统自动恢复。工作原理是软件系统出错时，在固定的时间(这个时间可以配置)产生一个复位或者中断，让程序重新执行或者按照中断服务程序执行，而不至于系统崩溃。从而增加了软件系统的安全性能。

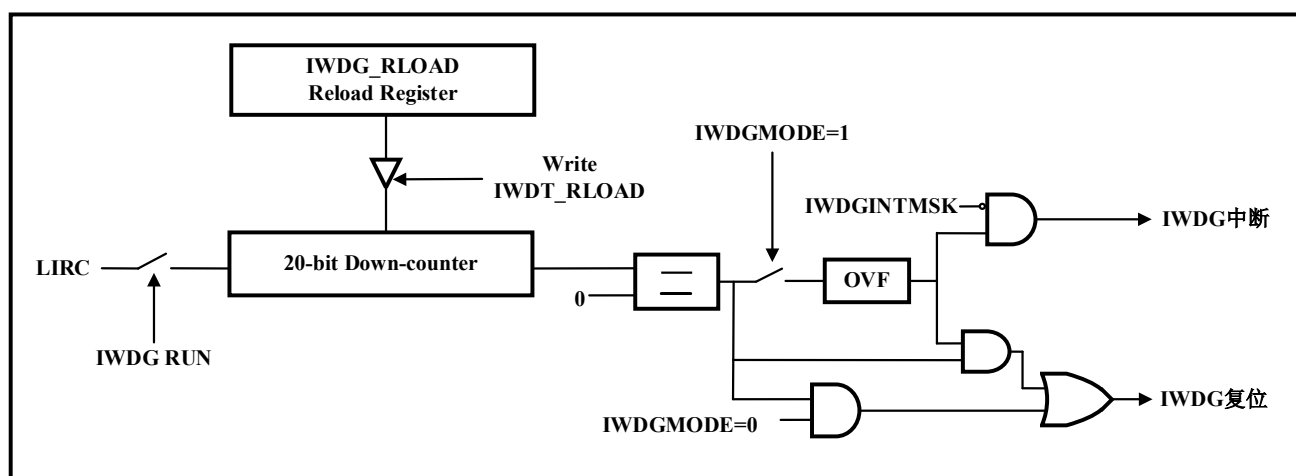


图 20-1. IWDG 整体框图

20.2 特性

- ◆ IWDG 模块是一个 20 位的计数器，每一个计数时钟 LIRC 计数累加一次，计数时间可配置为 26us-27s；
- ◆ 内部低速时钟计数；
- ◆ 当计数溢出时，支持中断和复位 2 种方式；
- ◆ 可配置的计数的溢出时间；
- ◆ 启动并清零 IWDG 具有操作序列要求，增加安全性能；
- ◆ 部分寄存器写保护功能，防止程意外操作。

20.3 功能描述

20.3.1 超时周期

看门狗超时周期由计数器数值决定，下表列出了它们的数值

表 20.1. 看门狗超时周期(假定计数器时钟为 38.4KHz)

IWDG_RLOAD[19:0]寄存器配置值	超时周期
0x00000	26us
...	...
0x003FF	26.6ms
...	...
0xFFFFF	27s

20.3.2 IWDG 溢出后产 Th 中断

在本模式下，IWDG 将按所设定的时间周期性的产生中断。在中断服务程序中需要清除 IWDG 溢出标志。

配置方法如下所示：

- 1) 写 0x55aa6699 到 IWDG_UNLOCK 解除 IWDG 的寄存器写保护，如果 IWDG_UNLOCK.IWDGREN 为 1 可省略本步骤；
- 2) 配置 IWDG_CFGR.IWDGMODE 为 1，选择中断方式；
- 3) 选择配置 IWDG_CFGR.IWDGINTMSK 为 0，则 CPU 响应 IWDG 的中断信号；配置为 1，则中断被屏蔽，CPU 只能通过读取 IWDG_SR.IWDGOVF 来判断是否计数溢出；
- 4) 配置 IWDG_RLOAD 寄存器。选择 IWDG 计数溢出时间；
- 5) 写非 0x55aa6699 到 IWDG_UNLOCK 开启 IWDG 的寄存器写保护；
- 6) 写 0x55 到 IWDG_CMDCR，启动 IWDG；
- 7) 如果中断产生在中断服务程序中先接触 IWDG 的寄存器保护，然后对 IWDG_INTCLR 写 1 清除中断标记。

20.3.3 IWDG 溢出后产 Th 复位

在本模式下，IWDG 计数器溢出后会产生 Reset 信号，该信号会复位 MCU。用户需要在 IWDG 溢出前重装载 IWDG 计数器，从而避免产生 IWDG 复位。

配置方法如下：

- 1) 写 0x55aa6699 到 IWDG_UNLOCK 解除 IWDG 的寄存器写保护，如果 IWDG_UNLOCK.IWDGREN 为 1 可省略本步骤；
- 2) 配置 IWDG_CFGR.IWDGMODE 为 0，选择复位方式；
- 3) 配置 IWDG_RLOAD 寄存器。选择 IWDG 计数溢出时间；
- 4) 写非 0x55aa6699 到 IWDG_UNLOCK 开启 IWDG 的寄存器写保护；
- 5) 写 0x55 到 IWDG_CMDCR，启动 IWDG；
- 6) 在计数溢出前写 0xAA 到 IWDG_CMDCR，刷新 IWDG 计数器。

20.4 寄存器列表

基地址：0x40002400

表 20.2. AWK 寄存器列表和复位值

偏移地址	名称	描述	默认值
0x00	IWDG_CMDCR	IWDG 控制命令寄存器	0x00000000
0x04	IWDG_CFGR	IWDG 配置寄存器	0x00000000
0x08	IWDG_RLOAD	IWDG 计数器重装载寄存器	0x000FFFFF
0x0c	IWDG_CNTVAL	IWDG 计数器值	0x000FFFFF
0x10	IWDG_SR	IWDG 中断状态寄存器	0x00000000
0x14	IWDG_INTCLR	IWDG 中断清除寄存器	0x00000000
0x18	IWDG_UNLOCK	IWDG 寄存器访问保护	0x00000000

20.5 寄存器说明

20.5.1 IWDG 控制命令寄存器(IWDG_CMDCR)

地址偏移：0x00

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7:0	CMD[7:0]	0x55: IWDG 启动命令 0xaa: IWDG 重装载刷新命令	0x0	WO

注意：在 IWDG 运行时才能写重装载命令

20.5.2 IWDG 配置寄存器(IWDG_CFGR)

地址偏移：0x04

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													IWDG RUNF	IWDG INTMS	IWDG MODE
													RO	R/W	R/W

位	标记	功能描述	复位值	读写
31:3	—	保留	0x0	—
2	IWDGRUNF	IWDG 运行标志 0: IWDG 停止 1: IWDG 运行(即在计数)	0	RO
1	IWDGINTMSK	IWDG 中断屏蔽 0: 中断不屏蔽(通知给 CPU) 1: 中断被屏蔽	0	R/W
0	IWDGMODE	IWDG 计数溢出模式选择位 0: 复位方式 1: 中断方式, 生成一个中断信号, 然后重启计数器, 如果中断没有在第二次超时发生之前被清除的话, 则生成一个系统复位信号。 注意: IWDG 产生复位后会复位整个系统	0	R/W

20.5.3 IWDG 计数器重装载寄存器(IWDG_RLOAD)

地址偏移: 0x08

复位值: 0x000FFFFF

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

位	标记	功能描述	复位值	读写
31:20	—	保留位		
19:0	IWDGRLOAD[19:0]	IWDG 重装载寄存器	0xFFFFF	R/W

20.5.4 IWDG 计数器值寄存器(IWDG_CNTVAL)

地址偏移: 0x0C

复位值: 0x000FFFFF

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

位	标记	功能描述	复位值	读写
31:20	—	保留位	0x0	—
19:0	IWDGCNT[19:0]	IWDG 计数值寄存器	0xFFFFF	RO

20.5.5 IWDG 中断状态寄存器(IWDG_SR)

地址偏移: 0x10

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:1	—	保留	0x0	—
0	IWDGOVF	IWDG 溢出中断标志位 0: IWDG 无溢出中断发生 1: IWDG 有溢出中断发生 注意: 1. 当 IWDG 配置为复位方式, 不管 IWDG 计数器是否溢出, 这位都不会置高。 2. 当 IWDG 配置为中断方式, 不管 IWDGINTMSK 位是否置高, 只要 IWDG 计数器溢出, 这位就置高。	0	RO

20.5.6 IWDG 中断清除寄存器(IWDG_INTCLR)

地址偏移: 0x14

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:1	—	保留	0x0	—
0	IWDGINTCLR	IWDG 中断清除 0: 写 0, 无任何动作 1: 写 1, 清除 IWDG 中断标志	0x0	WO

20.5.7 IWDG 保护寄存器(IWDG_UNLOCK)

地址偏移: 0x18

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														IWDGREN	
														R/W	

位	标记	功能描述	复位值	读写
31:1	—	保留	0x0	—
0	IWDGREN	IWDG 中断清除 0: 不能更改 IWDG 的相关寄存器 1: 可以更改 IWDG 的相关寄存器	0	R/W

注：写 0x55aa6699 到 IWDG_UNLOCK 可以解除 IWDG 的寄存器写保护，写任意非 0x55aa6699 值到 IWDG_UNLOCK 可以开启 IWDG 的寄存器写保护(IWDG_CFG，IWDG_RLOAD，IWDG_INTCLR)

20.6 注意

- 喂狗指令到看门狗定时器被更新需要两个看门狗计数时钟源时钟的延迟。
- 系统操作看门狗时，两次喂狗间隔需要允许有至少三个看门狗时钟。

21. 系统窗口看门狗(WWDG)

21.1 概述

窗口看门狗定时器(WWDG)的目的是在一个指定的窗口周期中执行系统复位，防止软件在任何不可预知的条件下进入不可控制的状态。

21.2 特征

- 一个 8 位向下计数器(WWDG_CNT)和一个 8 位比较值(WINCMP)使 WWDG 超时窗口周期可调；
- 支持 20 位值(PRSC)选择看门狗预分频值；
- 支持窗口计数值比较中断和计数溢出、加载计数值出错复位。

21.3 结构框图

窗口看门狗定时器框图如下：

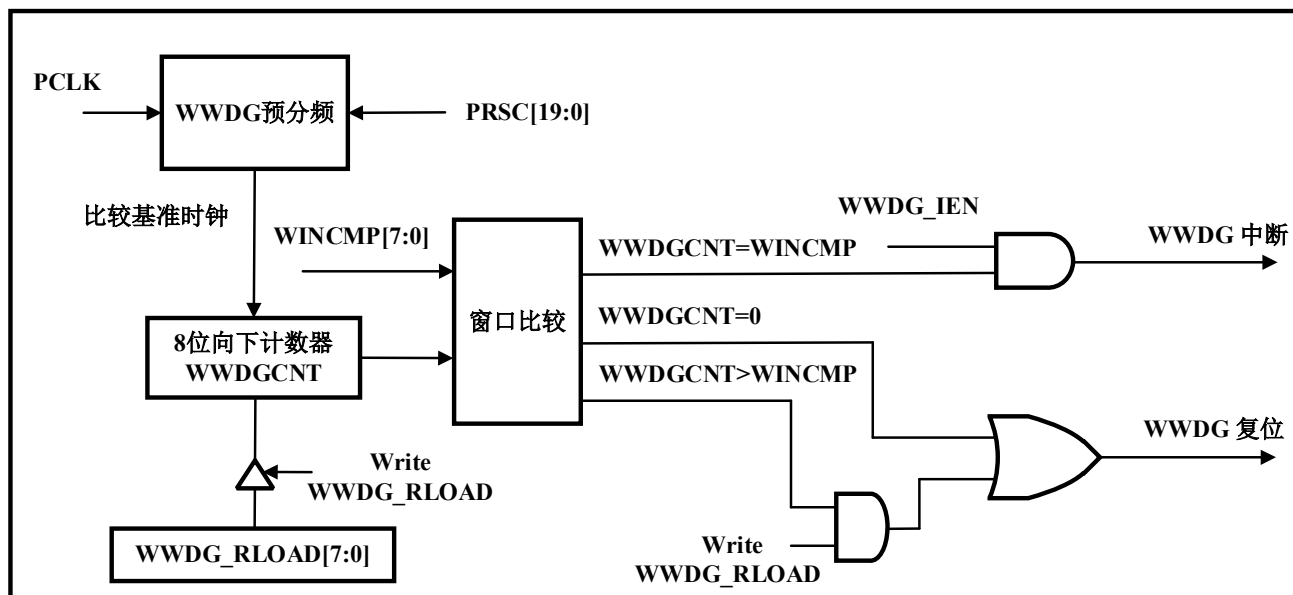


图 21-1. WWDG 结构框图

21.4 基本配置

WWDG 外设时钟源通过 WWDGCKEN(APBCLKEN[12])来使能。

- 1.通过 WWDG_RLOAD[19:0]配置窗看门狗计数初始值；
- 2.通过 WWDG_CR.PRSC[19:0]配置计数时钟预分频；
- 3.通过 WWDG_CR.WINCMP[7:0]配置窗比较值；
- 4.根据是否需要使能中断，来配置 WWDG_INTEN.WWDGIEN；
- 5.通过 WWDG_CR.WINCMP[7:0]来配置窗比较值；
- 6.通过 WWDG_CR.WWDGEN 写 1 来开启窗看门狗。

21.5 功能描述

窗口看门狗定时器(WWDG)是一个 8 位向下计数器，该计数器带一个可选择预分频值，不同的预分频值对应不同的看门狗定时溢出时间。8 位窗口看门狗定时器的时钟源是 PCLK 时钟经分频后的时钟，看门狗的时钟源带一个可选择的 20 位预分频值，该值可通过 WWDG_CR.PRSC[19:0]位来设置选择，对应预分频值如下表。

表 21.1. 窗口看门狗定时器预分频值选择

PRSC[19:0]	预分频值	定时溢出周期	定时溢出间隔 PCLK=24MHz
0x00000	1	Tplck*1	41.7ns
0x00001	2	Tplck*2	83.4ns
0x00002	3	Tplck*3	125.1ns
0x00003	4	Tplck*4	166.8ns
0x00004	5	Tplck*5	208.5ns
...
0x80000	524289	Tplck*524289	21.9ms
...
0xFFFFF	1048576	Tplck*1048576	43.8ms

21.5.1 窗口看门狗定时器的计数

当 WWDGEN(WWDG_CR[28])位被使能,窗口看门狗向下计数器将会从 WWDG_CNT[7:0]

向下递减计数到 0，并且不能够被软件关闭。为了防止程序在非用户指定位置关闭窗口看门狗定时器，窗口看门狗定时器控制寄存器的 WWDGEN 在芯片上电或复位后仅可写一次。当 WWDGEN(WWDG_CR[28]) 位被软件使能以后，用户不能禁止窗口看门狗定时器 (WWDG_CR[28])，修改计数器预分频周期 (WWDG_CR[27:8])，或修改窗口比较值 (WWDG_CR[7:0])，除非芯片复位。窗口看门狗定时器在 CPU 进入 Sleep 模式或是 Deep Sleep 模式时会停止计数，CPU 被唤醒后恢复正常工作。

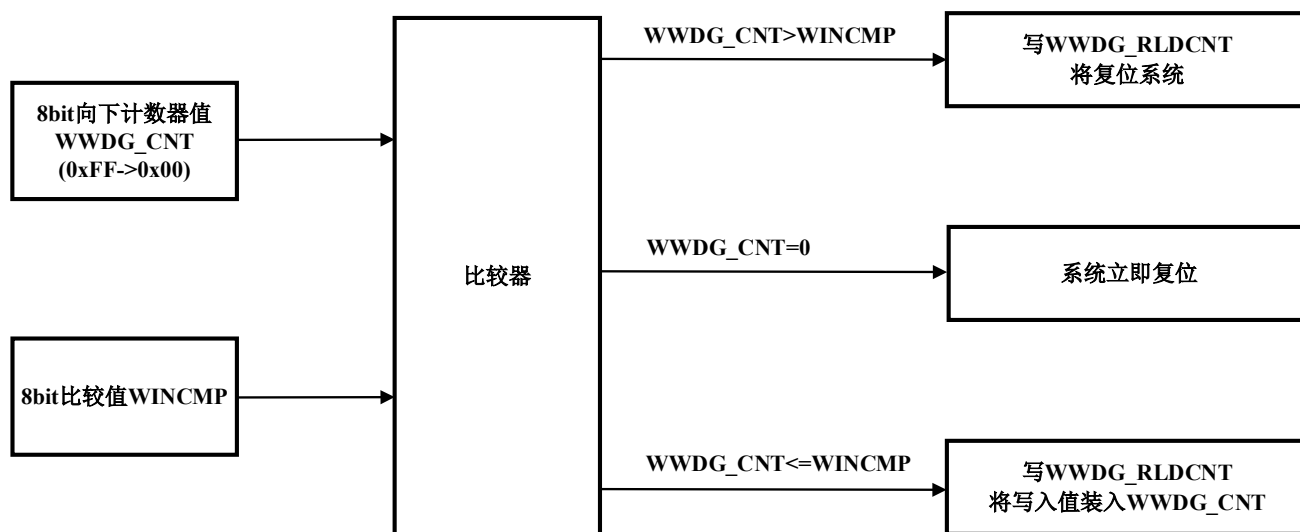


图 21-2. WWDG 复位和重载过程

21.5.2 窗口看门狗定时器比较中断

窗口看门狗定时器向下计数过程中，当窗口看门狗定时器计数值 WWDG_CNT[7:0] 等于窗口比较值 WINCMP(WWDG_CR[7:0])，WWDGIF(WWDG_SR[0]) 会被置 1 并且 WWDGIF 可以被软件清零。如果 WWDGIEN(WWDG_INTEN[0]) 位被使能，当 WWDGIF 位被硬件置 1，就会产生窗口看门狗比较匹配中断。

21.5.3 窗口看门狗定时器复位系统

当 WWDG 计数器的值计到 0 时 WWDGRST(RCC_RSTSR[2]) 会被置 1。在 WWDG 计数器下数到 0 前，用户必须通过对 WWDG_RLOAD 写值来进行重载，从而阻止 WWDG 复位的发生。重载的动作只能在计数器的值小于或等于 WINCMP 值时进行。如果 WWDG 计数器当前值大于 WINCMP 的值，用户对 WWDG_RLOAD 寄存器写，窗口看门狗定时器复位系统信号

将立刻产生，并导致芯片复位。

21.5.4 窗口看门狗定时器的窗口设置限制

当用户对 WWDG_RLOAD 寄存器写重载 WWDG 的值的时候：

$$Tpclk = Thclk * (2 * ABPCKDIV[7:0])$$

设定时间间隔：

$$T = Tpclk * (WWDG_PRSC[19:0] + 1) * (WWDG_RLOAD[7:0] + 1)$$

用户可以根据需要来配置分频寄存器 WWDG_PRSC[19:0]和 WWDG_RLOAD[7:0]的值来达到想要的时间间隔。为了保证正常工作，WWDG_RLOAD[7:0]的值要大于等于 1。

21.6 与独立看门狗定时器(IWDG)比较

21.6.1 复位条件和复位延时

IWDG 和 WWDG 通常是应用在系统跑到不可控制的状态后复位系统。IWDG 只有一个条件可以触发复位信号，WWDG 有两种条件可以触发 WWDG 产生复位信号：

WWDGCNT=0;

WWDGCNT 大于 WINCMP 时往 WWDG_RLOAD 写入。

一旦 WWDGRST 被置 1，WWDG 将立即复位系统。

21.6.2 唤醒功能

IWDG 支持该功能并且在 Deep Sleep 模式下继续工作。相比之下，WWDG 不支持该功能并且 WWDG 的计数器在 Deep Sleep 模式下会停止计数。

21.7 寄存器列表

WWDG: 基址: 0x40002000

偏移地址	名称	描述	默认值
0x00	WWDG_RLOAD	窗口看门狗定时器重载计数寄存器	0x000000FF
0x04	WWDG_CR	窗口看门狗定时器控制寄存器	0x080000FF
0x08	WWDG_INTEN	窗口看门狗定时器中断使能寄存器	0x00000000
0x0C	WWDG_SR	窗口看门狗定时器状态寄存器	0x00000000
0x10	WWDG_INTCLR	窗口看门狗定时器中断清除寄存器	0x00000000
0x14	WWDG_CNTVAL	窗口看门狗定时器计数器值寄存器	0x000000FF

21.8 寄存器说明

21.8.1 窗口看门狗定时器重载计数寄存器(WWDG_RLOAD)

地址偏移: 0x00

复位值: 0x000000FF

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

位	标记	功能描述	复位值	读写
31:16	—	保留	0x0	—
7:0	WWDG_RLOAD [7:0]	窗口看门狗定时器重载计数寄存器。写入值大于 0	0xFF	WO

21.8.2 窗口看门狗定时器控制寄存器(WWDG_CR)

地址偏移: 0x04

复位值: 0x080000FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			WWDG EN	PRSC[19:8]											
			R/W	R/W											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRSC[7:0]								WINCMP[7:0]							
R/W								R/W							

位	标记	功能描述	复位值	读写
31:29	—	保留	0x0	—
28	WWDGEN	窗口看门狗使能位设置该位使能窗口看门狗定时器 0=禁止窗口看门狗定时器功能 1=使能窗口看门狗定时器功能	0	R/W
27:8	PRSC[19:0]	WWDG 预分频 Fpclk/(PRSC+1)	0x80000	R/W
7:0	WINCMP[7:0]	WWDG 窗口比较寄存器设置该寄存器调整有效的重载窗口。 注：软件仅能写 WWDG_RLOAD 当 WWDG 计数器值在 0 和 WINCMP 之间。如果软件写 WWDG_RLOAD 当 WWDG 计数器值大于 WINCMP，WWDG 会产生复位信号	0xFF	R/W

注意：当 WWDGEN 设定为 1 后，该寄存器的软件配置将被禁止。

21.8.3 窗口看门狗定时器中断使能寄存器(WWDG_INTEN)

地址偏移：0x08

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														WWDGIEN	
														R/W	

位	标记	功能描述	复位值	读写
31:1	—	保留	0x0	—
0	WWDGIEN	WWDG 中断使能位 设置该位使能窗口看门狗定时器中断功能。 0=禁止窗口看门狗定时器中断功能 1=使能窗口看门狗定时器中断功能	0x0	R/W

21.8.4 窗口看门狗定时器状态寄存器(WWDG_SR)

地址偏移: 0x0C

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:1	—	保留	0x0	—
0	WWDGIF	WWDG比较匹配中断标志 0: 无窗口看门狗定时器中断 1: 有窗口看门狗定时器中断 当WINCMP和WWDG计数器匹配, 该位置1, 软件对WWDG_INTCLR的INTCLR写1清0该位。	0	RO

21.8.5 窗口看门狗定时器中断清除寄存器(WWDG_INTCLR)

地址偏移: 0x10

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:1	—	保留	0x0	—
0	INTCLR	WWDG比较匹配中断标志清除 软件写1清0该位。	0	WO

21.8.6 窗口看门狗定时器计数器值寄存器(WWDG_CNTVAL)

地址偏移: 0x14

复位值: 0x000000FF

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

位	标记	功能描述	复位值	读写
31:16	—	保留	0x0	—
7:0	WWDGCNT[7:0]	WWDG计数器值 该寄存器表示窗口看门狗计数器当前的值，该寄存器只读。	0xFF	RO

22. 通用异步收发器 (UART)

22.1 简介

本产品带有 2 个通用 UART 模块(UART0/1)，支持半双工和全双工传输；支持 8bit、9bit 数据格式；支持 Mode0/1/2/3 四种不同传输模式；UART0 的波特率可以由 TIM10 产生或者自动波特率发生器产生，UART1 的波特率可以由 TIM11 产生或者自动波特率发生器产生；支持多机通讯模式；支持自动地址识别；支持给定地址和广播地址。

通用 UART(UART0/1)只有一个时钟输入 PCLK，寄存器配置逻辑和数据收发逻辑都工作在该时钟域。

22.2 结构框图

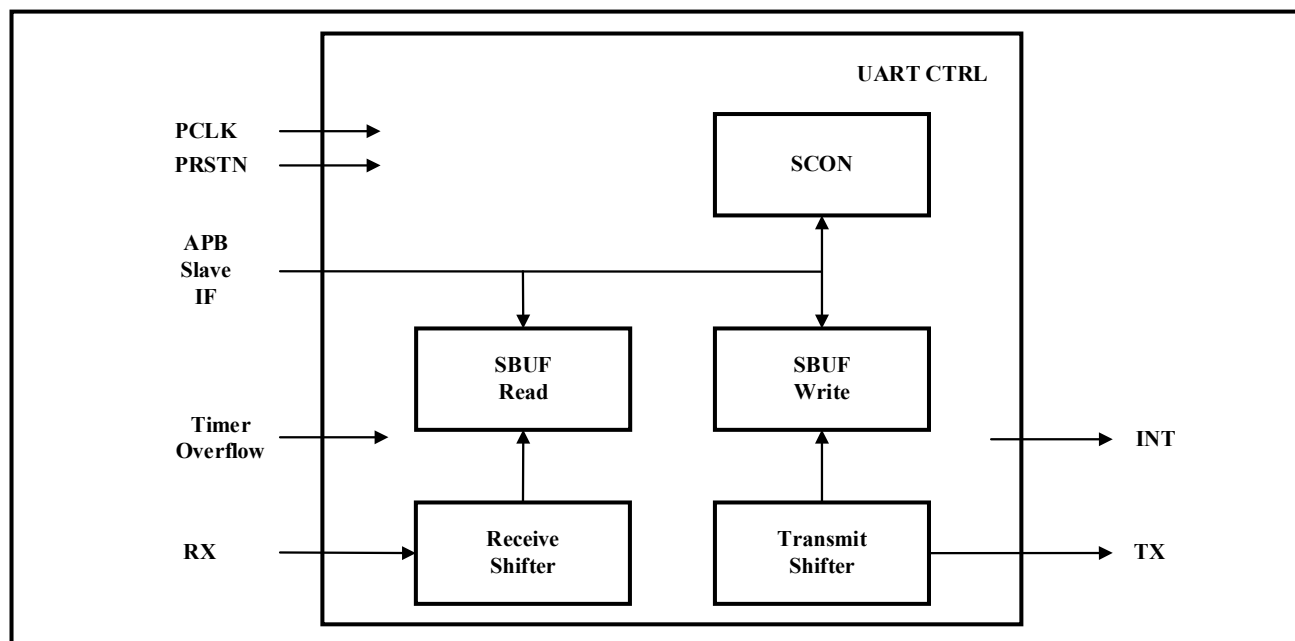


图 22-1. UART 结构图

22.3 工作模式

22.3.1 Mode0 (同步模式, 半双工)

当工作在 Mode0 时, UART 工作在同步模式, 其波特率为固定的 PCLK 时钟的 1/12。UART

接收数据由 RXD 输入、UART 发送数据由 TXD 输出，RXD 此时为输入输出端口。UART 同步移位时钟由 TXD 输出，TXD 此时为输出端口。注意，本模式只能作为主机发送同步移位时钟，不可以作为从机从外部接收移位时钟。该模式下，传输的数据位宽只能是 8 位的，没有起始位和结束位。

将 UARTx_SCON.SM0 和 UARTx_SCON.SM1 清零，可进入 Mode0 工作模式。

22.3.1.1 发送数据

发送数据时，清除 UARTx_SCON.REN 位，并将数据写入 UARTx_SBUF 寄存器。此时，发送数据将从 TXD 输出(低位在先，高位在后)，同步移位时钟从 TXD 输出。

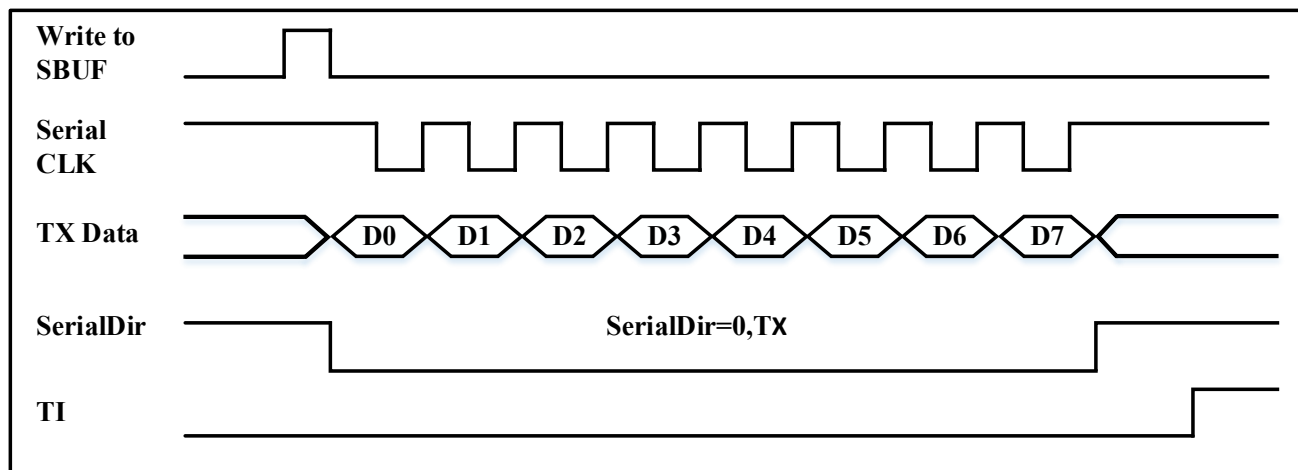


图 22-2. Mode0 发送数据

22.3.1.2 接收数据

接收数据时，将 UARTx_SCON.REN 位置 1，并将 UARTx_ISR.RI 位清零。当接收结束，数据可从 UARTx_SBUF 寄存器读出。此时，接收数据从 RXD 输入(低位在先，高位在后)，同步移位时钟从 TXD 输出。

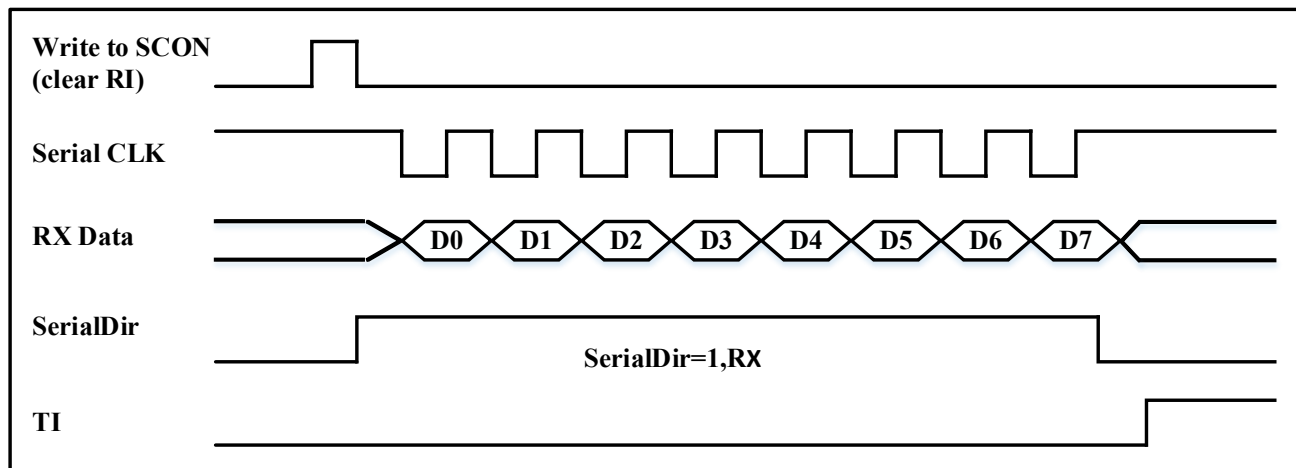


图 22-3. Mode0 接收数据

22.3.2 Mode1（异步模式，全双工）

当工作在 Mode1 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由 10 位组成：起始位“0”开始，紧接着 8 位数据位(低位在先，高位在后)，后是结束位“1”。该模式下，波特率可以由可编程定时器模块产生，也可以由模块内部的自动波特率发生器产生。当 UARTx_BAUDCR.SELF_BRG 为 0 时，选择由定时器产生波特率时，UART0 的波特率由 TIM10 产生，UART1 的波特率由 TIM11 产生；当 UARTx_BAUDCR.SELF_BRG 置 1 时，UART0、UART1 的波特率都由各自内部的自动波特率发生器产生。波特率产生公式请参考 22.3.5.2 Mode1/3。

将 UARTx_SCON.SM0 清 0，UARTx_SCON.SM1 置 1，可进入 Mode1 工作模式。

22.3.2.1 发送数据

发送数据时，与 UARTx_SCON.REN 的值无关，将所发送数据写入 UARTx_SBUF 寄存器中，数据就会从 TXD 移出(低位在先，高位在后)。

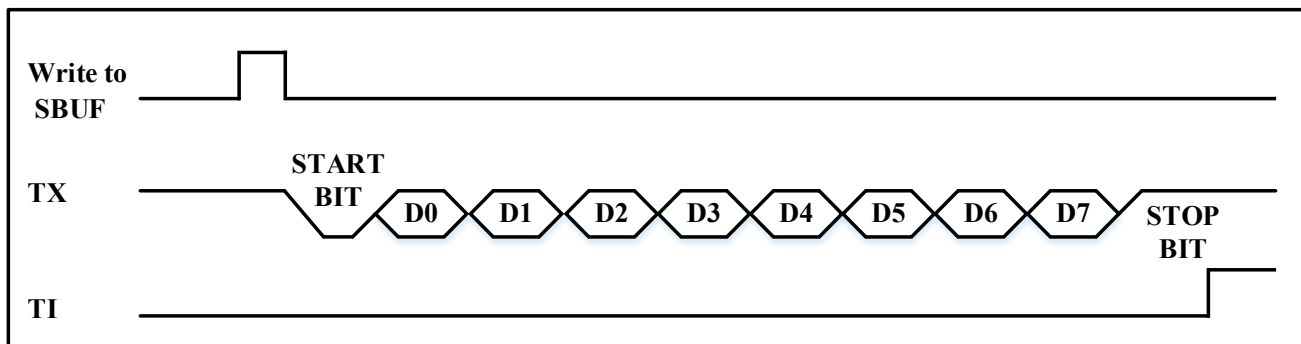


图 22-4. Mode1 发送数据

22.3.2.2 接收数据

接收数据时，需将 UARTx_SCON.REN 位置 1，并将 UARTx_ISR.RI 位清 0。开始接收 RXD 上数据(低位在先，高位在后)，当接收完毕，可以从 UARTx_SBUF 寄存器读出。

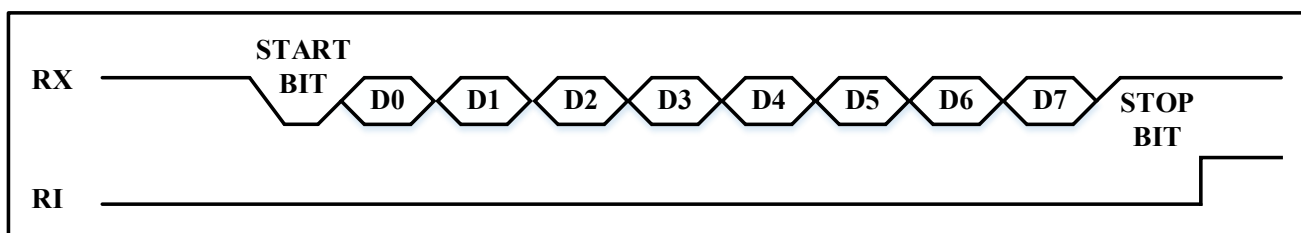


图 22-5. Mode1 接收数据

22.3.3 Mode2（异步模式，全双工）

当工作在 Mode2 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由 11 位组成：起始位“0”开始，接着是 8 个数据位，1 个 TB8 位和结束位。额外的 TB8 位是用来在多机通讯环境下使用，当 TB8=1，表明所接收的是地址帧；当 TB8=0，表明所接收的是数据帧。当不需要多机通讯时，此位也可以作为奇偶校验位来使用。该模式下，波特率可以独立产生，不需要外部定时器模块产生。

将 UARTx_SCON.SM0 置 1，UARTx_SCON.SM1 清 0，可进入 Mode2 工作模式。

22.3.3.1 发送数据

发送数据时，与 UARTx_SCON.REN 的值无关，并将所发送数据写入 UARTx_SBUF 寄存器中，数据就会从 TXD 移出(低位在先，高位在后)。

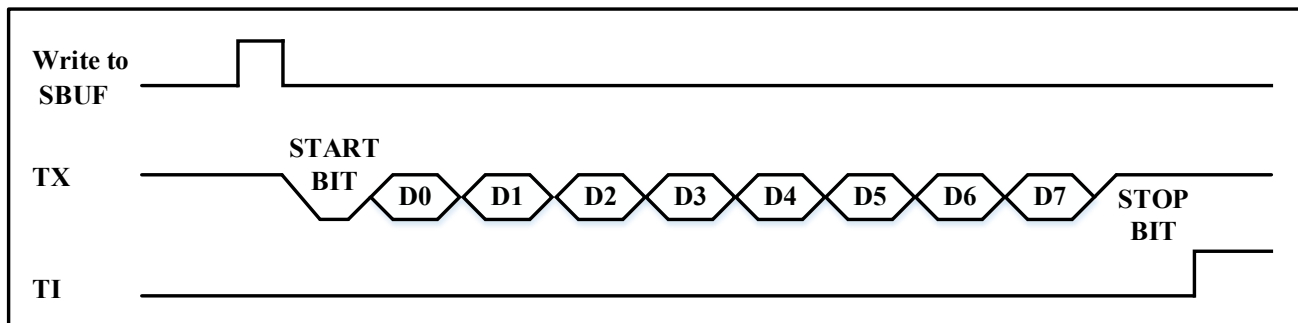


图 22-6. Mode2 发送数据

22.3.3.2 接收数据

接收数据时，需将 UARTx_SCON.REN 位置 1，并将 UARTx_ISR.RI 位清 0。开始接收 RXD 上数据(低位在先，高位在后)，当接收完毕，可以从 UARTx_SBUF 寄存器读出。

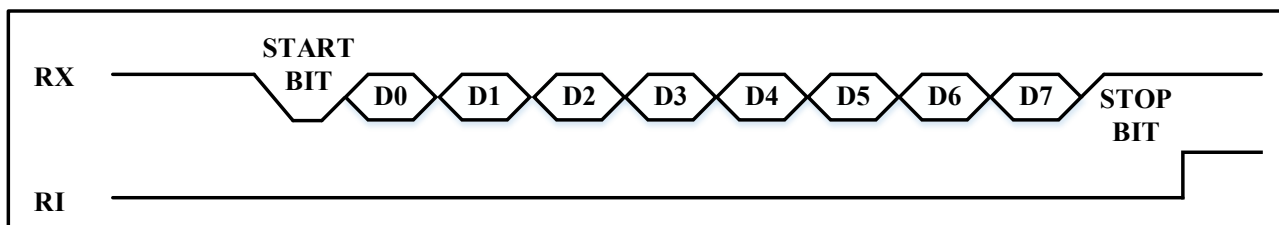


图 22-7. Mode2 接收数据

22.3.4 Mode3（异步模式，全双工）

Mode3 的数据格式，传输时序以及操作方式都与 Mode2 相同，唯一的区别是 Mode3 的波特率选择由可编程定时器产生或者内部自动波特率发生器产生，而不是像 Mode2 只由设备自己独立产生。Mode3 的波特率是可编程的，波特率生成方式与 Mode1 相同。

将 UARTx_SCON.SM0 置 1，UARTx_SCON.SM1 置 1，可进入 Mode3 工作模式。

22.3.5 波特率编程

22.3.5.1 Mode0

当工作在 Mode0 时，波特率被固定在 PCLK 的 1/12，不需要可编程定时器(TIMER)的支持。

22.3.5.2 Mode1/3

当工作在 Mode1 或者 Mode3 时，波特率产生公式如下所示：

UARTx_BAUDCR.SELF_BRG=0,使用可编程定时器(TIMER)波特率模式:

$$\text{BaudRate} = \frac{(\text{UARTx_SCON.BAUD} + 1) * F_{pclk}}{32 * (2^{16} - \text{TIMx_BGLOAD}[15:0])}$$

UARTx_BAUDCR.SELF_BRG=1,使用自身波特率生成模式:

$$\text{BaudRate} = \frac{(\text{UARTx_SCON.BAUD} + 1) * F_{pclk}}{32 * (\text{UARTx_BAUDCR.BRG} + 1)}$$

其中, UARTx_SCON.DBAUD 表示双倍波特率, F_{pclk} 为 PCLK 时钟频率, TIMx_BGLOAD 为 TIMER 的周期装载计数值。注意, TIMER 必须配置为 16 位自动重载模式, 立即重载寄存器(TIMx_LOAD)和周期重载寄存器(TIMx_BGLOAD)要写入相同的初始值。

22.3.5.3 Mode2

当工作在 Mode2 时, 波特率被固定在如下公式所得值:

$$\text{BaudRate} = \frac{(\text{UARTx_SCON.BAUD} + 1) * F_{pclk}}{64}$$

其中, UARTx_SCON.DBAUD 表示双倍波特率, F_{pclk} 为 PCLK 时钟频率。

22.3.6 帧错误检测

Mode1/2/3 具有帧错误检测功能, 硬件会自动检测接收到的帧数据是否带有效的 Stop 位。如果没有收到有效 Stop 位, 则 UARTx_ISR.FE 置 1。UARTx_ISR.FE 位由硬件置 1, 软件清 0, 如果软件未及时清 0, 则后续收到数据即使带有效 Stop 位, 也不会把 UARTx_ISR.FE 标志清 0。

22.3.7 多机通讯

Mode2/3 具有多机通讯功能, 为此在其帧格式中增加了 1 位 TB8/RB8。将 UARTx_SCON.SM2 置“1”, 可开启多机通讯位。当开启多机通讯位后, 发送数据时, 主机可以通过 UARTx_SCON.TB8 来区分当前帧是地址帧(UARTx_SCON.TB8=1)还是数据帧(UARTx_SCON.TB8=0)。接收数据时, 从机会忽略 RB8 位(第 9 位)为“0”的当前接收帧。当收到帧的 RB8 位(第 9 位)为“1”表明其是地址帧, 从机会继续判断接收到的地址与其自身地

址是否相等。如果匹配，则从机会对 UARTx_SCON.RB8 置“1”，并对 UARTx_ISR.RI 置“1”，以表明该帧为地址帧并且地址已经匹配。从机软件看到 UARTx_SCON.RB8=1 并且 UARTx_ISR.RI=1 后，先把 UARTx_SCON.SM2 位清“0”，然后准备接受给它的数据帧。如果地址不等，表明主机并不是寻址该从机，从机硬件保持 UARTx_SCON.RB8 和 UARTx_ISR.RI 为“0”，软件保持 UARTx_SCON.SM2 位为“1”，从机继续处于地址监听状态。

22.3.8 自动地址识别

当开启多机通讯位后(UARTx_SCON.SM2 置“1”)，自动地址识别功能也将开启。该功能由硬件实现，使得从机可以检测接收到每个地址帧，如果该地址与从机地址匹配，接收端会给出 UARTx_ISR.RI 接收标志。如果地址不匹配，则接收端不会给出任何接收标志。如果有需要，也可以在 Mode1 下开启多机通讯位，此时 TB8 位由 Stop 位代替。当从机接收到匹配的地址帧和有效的 Stop 位时，UARTx_ISR.RI 会被置“1”。为了支持自动地址识别，定义了广播地址和给定地址的概念。

22.3.9 给定地址

UART 设备的 UARTx_SADDR 寄存器用来表示自己的设备给定地址，UARTx_SADEN 寄存器是地址掩码，可以用来定义地址中的无关位。当 UARTx_SADEN 的某一位为“0”，表示该位地址为无关位，也就是说在地址匹配过程中，该位地址不参与地址匹配。这些无关位增加了寻址的灵活性，使得主机可以同时寻址一个或者多个从机设备。注意，如果需要给出唯一匹配地址，UARTx_SADEN 寄存器必须设为 8'hFF。

GivenAddr=SADDR&SADEN

22.3.10 广播地址

广播地址是用来同时寻址所有从机设备的，一般广播地址为 8'hFF。

BoardCastAddr=UARTx_SADDR&UARTx_SADEN

22.3.11 举例

假设某从机的 UARTx_SADDR 和 UARTx_SADEN 配置如下：

SADDR: 0b01101001

SADEN: 0b11111011

那么其给定地址和广播地址如下:

Given: 0b01101x01

Broadcast: 0b11111x11

可见, 主机可以用四个地址寻址到本从机, 分别是:

0b01101001 和 0b01101101(given address)

0b11111011 和 0b11111111(broadcast address)。

22.4 收发端缓存

22.4.1 接收缓存

通用 UART(UART0/1)接收端有一个帧长度(8/9bits)的接收缓存, 也就是说当一帧数据接收完毕后, 接收缓存中的数据会被一直保持, 直到下一帧数据的 Stop 位接收完毕后, 接收缓存才会更新为新一帧数据。

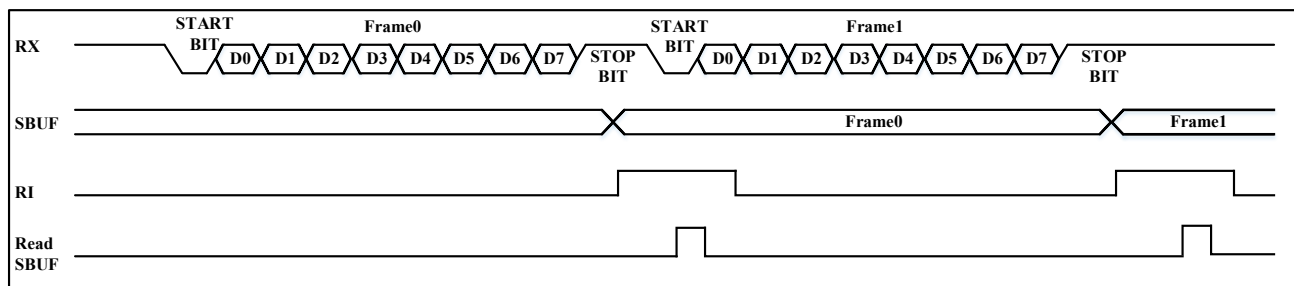


图 22-8. 接收缓存

22.4.2 发送缓存

通用 UART(UART0/1)发送端不支持发送缓存。如果在发送数据过程中, 填写 UARTx_SBUF 寄存器, 将会破坏当前正在发送数据。软件应该避免这种操作。

22.5 IrDA 红外功能

IrDASIR 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'

(见图 22-9. IrDA 结构框图)。SIR 发送编码器对从 UART 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。对于 SIRENDEC 应用, UART 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 UART。在空闲状态里, 解码器输入通常是高(标记状态)。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 UART 正在送数据给 IrDA 编码器), IrDA 接收线上的任何数据都将被 IrDA 解码器所忽略。如果接收器忙(也就是 UART 正在接收从 IrDA 解码器来的解码数据), 从 UART 的 TX 上到 IrDA 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; 或者取反发送。脉冲的宽度规定为正常模式时位周期的 3/16(见图 22-10. IrDA 收发脉冲)。
- SIR 解码器把接收到的 IrDA 信号转变成比特流后发送给 UART。
- SIR 接收逻辑把高电平状态解释为 '0', 把低脉冲解释为 '1'; 或者取反接收。
- 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。
- 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电路会过对宽度小于 2 个 PSC 周期的脉冲进行过滤操作(PSC 是在 UARTx_IRDACR 中编程的预分频值)。宽度小于 1 个 PSC 周期的脉冲一定会被过滤掉, 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。
- 接收器可以与一低功耗发送器通信。

22.5.1 IrDA 低功耗模式

IrDA 可以工作在正常模式, 也可以工作在低功耗模式。选择低功耗模式需要把 UART_IRDACR.IRLPMODE 寄存器置 1。

● 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时钟周期的 3 倍，该波特率的频率最小可以是 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。

● 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，UART 应该滤除宽度短于 1 个周期的脉冲。只有持续时间大于 2 个周期的 IrDA 低功耗波特率时钟(UART_IRDACR 中的 PSC)的低电平信号才被接受为有效的信号。

注意：

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)。

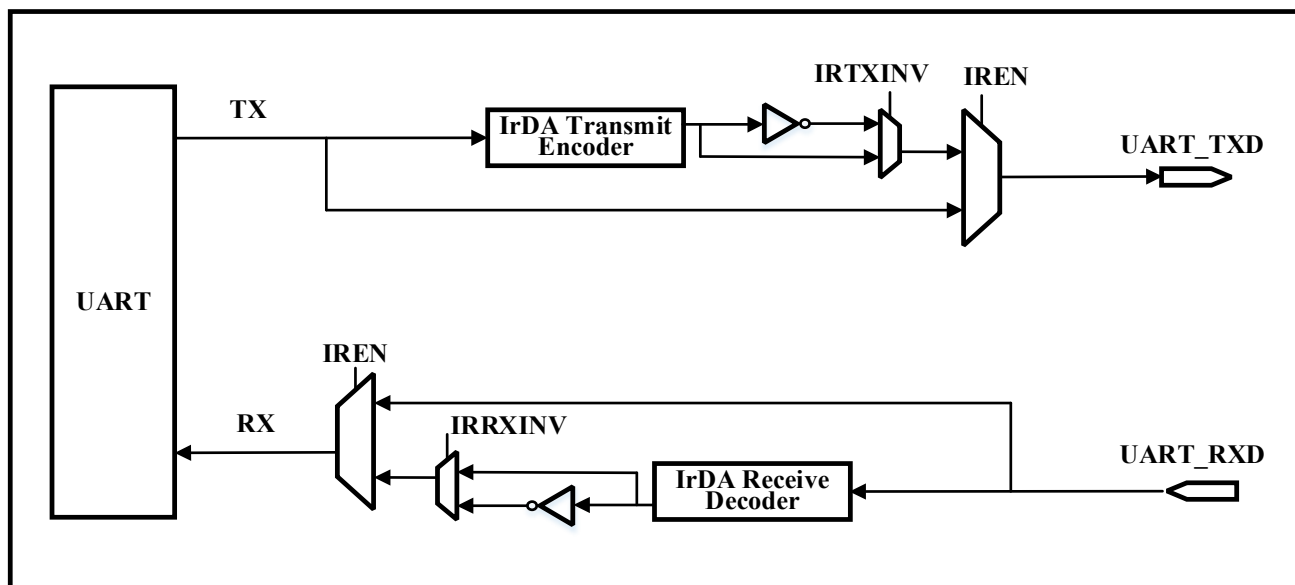


图 22-9. IrDA 结构框图

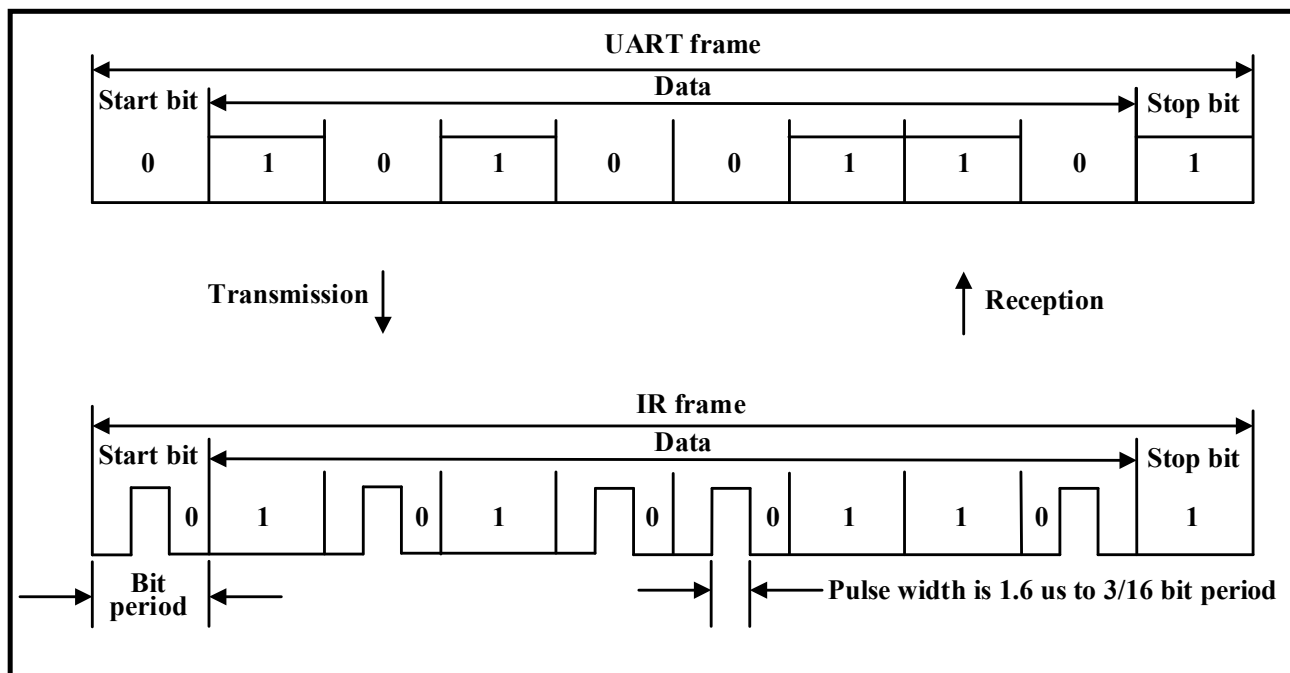


图 22-10. IrDA 收发脉冲

22.6 波特率的分频设置

波特率	PCLK=1MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	26	2403.85	0.16%	13	2403.85	0.16%
4800	13	4807.69	0.16%	7	4464.29	-6.99%
9600	7	8928.57	-6.99%	3	10416.67	8.51%
19200	3	20833.33	8.51%	2	15625.00	-18.62%
38400	2	31250.00	-18.62%	1	31250.00	-18.62%
57600	1	62500.00	8.51%	1	31250.00	-45.75%
76800	1	62500.00	-18.62%	0	-	-
115200	1	62500.00	-45.75%	0	-	-

波特率	PCLK=4MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	104	2403.85	0.16%	52	2403.85	0.16%
4800	52	4807.69	0.16%	26	4807.69	0.16%
9600	26	9615.38	0.16%	13	9615.38	0.16%
19200	13	19230.77	0.16%	7	17857.14	-6.99%
38400	7	35714.29	-6.99%	3	41666.67	8.51%
57600	4	62500.00	8.51%	2	62500.00	8.51%
76800	3	83333.33	8.51%	2	62500.00	-18.62%
115200	2	125000.00	8.51%	1	125000.00	8.51%

波特率	PCLK=10MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	260	2403.85	0.16%	130	2403.85	0.16%
4800	130	4807.69	0.16%	65	4807.69	0.16%
9600	65	9615.38	0.16%	33	9469.70	-1.36%
19200	33	18939.39	-1.36%	16	19531.25	1.73%
38400	16	39062.50	1.73%	8	39062.50	1.73%
57600	11	56818.18	-1.36%	5	62500.00	8.51%
76800	8	78125.00	1.73%	4	78125.00	1.73%
115200	5	125000.00	8.51%	3	104166.67	-9.58%

波特率	PCLK=14MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	365	2397.26	-0.11%	182	2403.85	0.16%
4800	182	4807.69	0.16%	91	4807.69	0.16%
9600	91	9615.38	0.16%	46	9510.87	-0.93%
19200	46	19021.74	-0.93%	23	19021.74	-0.93%
38400	23	38043.48	-0.93%	11	39772.73	3.57%
57600	15	58333.33	1.27%	8	54687.50	-5.06%
76800	11	79545.45	3.57%	6	72916.67	-5.06%
115200	8	109375.00	-5.06%	4	109375.00	-5.06%

波特率	PCLK=20MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	521	2399.23	-0.03%	260	2403.85	0.16%
4800	260	4807.69	0.16%	130	4807.69	0.16%
9600	130	9615.38	0.16%	65	9615.38	0.16%
19200	65	19230.77	0.16%	33	18939.39	-1.36%
38400	33	37878.79	-1.36%	16	39062.50	1.73%
57600	22	56818.18	-1.36%	11	56818.18	-1.36%
76800	16	78125.00	1.73%	8	78125.00	1.73%
115200	11	113636.36	-1.36%	5	125000.00	8.51%

波特率	PCLK=24MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	625	2400.00	0.00%	313	2396.17	-0.16%
4800	313	4792.33	-0.16%	156	4807.69	0.16%
9600	156	9615.38	0.16%	78	9615.38	0.16%
19200	78	19230.77	0.16%	39	19230.77	0.16%
38400	39	38461.54	0.16%	20	37500.00	-2.34%
57600	26	57692.31	0.16%	13	57692.31	0.16%
76800	20	75000.00	-2.34%	10	75000.00	-2.34%
115200	13	115384.62	0.16%	7	107142.86	-6.99%

波特率	PCLK=2MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	52	2403.85	0.16%	26	2403.85	0.16%
4800	26	4807.69	0.16%	13	4807.69	0.16%
9600	13	9615.38	0.16%	7	8928.57	-6.99%
19200	7	17857.14	-6.99%	3	20833.33	8.51%
38400	3	41666.67	8.51%	2	31250.00	-18.62%
57600	2	62500.00	8.51%	1	62500.00	8.51%
76800	2	62500.00	-18.62%	1	62500.00	-18.62%
115200	1	125000.00	8.51%	1	62500.00	-45.75%

波特率	PCLK=8MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	208	2403.85	0.16%	104	2403.85	0.16%
4800	104	4807.69	0.16%	52	4807.69	0.16%
9600	52	9615.38	0.16%	26	9615.38	0.16%
19200	26	19230.77	0.16%	13	19230.77	0.16%
38400	13	38461.54	0.16%	7	35714.29	-6.99%
57600	9	55555.56	-3.55%	4	62500.00	8.51%
76800	7	71428.57	-6.99%	3	83333.33	8.51%
115200	4	125000.00	8.51%	2	125000.00	8.51%

波特率	PCLK=11.0592MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	288	2400.00	0.00%	144	2400.00	0.00%
4800	144	4800.00	0.00%	72	4800.00	0.00%
9600	72	9600.00	0.00%	36	9600.00	0.00%
19200	36	19200.00	0.00%	18	19200.00	0.00%
38400	18	38400.00	0.00%	9	38400.00	0.00%
57600	12	57600.00	0.00%	6	57600.00	0.00%
76800	9	76800.00	0.00%	5	69120.00	-10.00%
115200	6	115200.00	0.00%	3	115200.00	0.00%

波特率	PCLK=16MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	417	2398.08	-0.08%	208	2403.85	0.16%
4800	208	4807.69	0.16%	104	4807.69	0.16%
9600	104	9615.38	0.16%	52	9615.38	0.16%
19200	52	19230.77	0.16%	26	19230.77	0.16%
38400	26	38461.54	0.16%	13	38461.54	0.16%
57600	17	58823.53	2.12%	9	55555.56	-3.55%
76800	13	76923.08	0.16%	7	71428.57	-6.99%
115200	9	111111.11	-3.55%	4	125000.00	8.51%

波特率	PCLK=22.12MHz					
	双波特率			单波特率		
	CNT	实际波特率	误差%	CNT	实际波特率	误差%
2400	576	2400.17	0.01%	288	2400.17	0.01%
4800	288	4800.35	0.01%	144	4800.35	0.01%
9600	144	9600.69	0.01%	72	9600.69	0.01%
19200	72	19201.39	0.01%	36	19201.39	0.01%
38400	36	38402.78	0.01%	18	38402.78	0.01%
57600	24	57604.17	0.01%	12	57604.17	0.01%
76800	18	76805.56	0.01%	9	76805.56	0.01%
115200	12	115208.33	0.01%	6	115208.33	0.01%

22.7 UART 寄存器列表

UART0 基地址: 0x40000000

UART1 基地址: 0x40000400

偏移地址	名称	描述	复位值
0x00	UARTx_SCON	控制寄存器	0x00000000
0x04	UARTx_SBUF	数据寄存器	0x00000000
0x08	UARTx_SADDR	地址寄存器	0x00000000
0x0C	UARTx_SADEN	地址掩码寄存器	0x00000000
0x10	UARTx_INTSR	中断标志位寄存器	0x00000000
0x14	UARTx_INTCLR	中断标志位清除寄存器	0x00000000
0x18	UARTx_BAUDCR	波特率控制寄存器	0x00000000
0x1C	UARTx_IRDACR	IrDA 控制寄存器	0x00000000

SM0	SM1	MODE	描述	波特率
0	0	0	位移寄存器	CLK/12
0	1	1	8 位串口传输	可变波特率
1	0	2	9 位串口传输	CLK/32,CLK/64
1	1	3	9 位串口传输	可变波特率

22.8 UART 寄存器说明

22.8.1 UART 控制寄存器(UARTx_SCON)

地址偏移：0x00

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						DBAUD	保留	SM01[1:0]	SM2	REN	TB8	RB8	TIEN	RIEN	
						R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	

位	标记	功能描述	复位值	读写
31:10	—	保留	0x0	—
9	DBAUD	双倍波特率 0: 单倍波特率 1: 双倍波特率	0	R/W
8	—	保留	0	—
7:6	SM0:SM1	工作模式: 00: mode0 01: mode1 10: mode2 11: mode3	0x0	R/W
5	SM2	多主机通讯 0: disable 1: enable SM2: 软件配置多机通讯以及自动地址匹配模式 1: 启动多从机通讯以及地址自动匹配 0: 关闭多从机通讯以及地址自动匹配 在模式 2 和模式 3 中: 如果 SM2=1, 并且 REN=1, 则接收机处于地址帧监测模式, 可以使用接收到的第 9 位 RB8 来进行地址筛选。RB8=1 为 地址帧, 通讯数据可以进入 SBUF, 置位 RI, 进入中断服务 程序中进行地址比较; RB8=0 为数据帧, 接收机忽略这些数 据帧并保持 RI=0 如果 SM2=0, 并且 REN=1, 则接收机不使用地址监测模式, 无论收到的 RB8 为 0 或 1, 都直接接收并且进入 SUBF, 置 位 RI, RB8 在这种模式下为校验位。	0	R/W
4	REN	接收使能 mode0: 0: 发送, 1: 接收	0	R/W

位	标记	功能描述	复位值	读写
		其他: 0: 发送, 1: 接收/发送		
3	TB8	发送 TB8 位	0	R/W
2	RB8	接收 RB8 位	0	R/W
1	TIEN	发送完成中断使能 0: disable 1: enable	0	R/W
0	RIEN	接收完成中断使能 0: disable 1: enable	0	R/W

22.8.2 UART 数据寄存器(UARTx_SBUF)

地址偏移: 0x04

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7:0	SBUF[7:0]	发送数据时, 当发送数据写入该寄存器; 接收数据时, 数据接收完毕后, 从该寄存器中读出。 注意, 对该寄存器读的值实际是 RX Buffer 中的值, 对该寄存器写的值实际是写到了 TX Shifter 中。	0x0	R/W

22.8.3 UART 地址寄存器(UARTx_SADDR)

地址偏移: 0x08

复位值: 0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								SADDR[7:0]							
								R/W							

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7:0	SADDR[7:0]	从机设备地址寄存器	0x0	R/W

22.8.4 UART 地址掩码寄存器(UARTx_SADEN)

地址偏移：0x0C

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7:0	SADEN[7:0]	从机设备地址掩码寄存器	0x0	R/W

22.8.5 UART 标志位寄存器(UARTx_INTSR)

地址偏移：0x10

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												TE	FE	TI	RI
												RO	RO	RO	RO

位	标记	功能描述	复位值	读写
31:4	—	保留	0x0	—
3	TE	发送缓存空中断标志位，硬件置位，硬件清零 0: TE 中断无效 1: TE 中断有效	0	RO
2	FE	接收帧错误标志位，硬件置位，软件清零 0: FE 中断无效 1: FE 中断有效	0	RO
1	TI	发送完成中断标志位，硬件置位，软件清零 0: TI 中断无效 1: TI 中断有效	0	RO
0	RI	接收完成中断标志位，硬件置位，软件清零 0: RI 中断无效 1: RI 中断有效	0	RO

22.8.6 UART 标志位清除寄存器(UARTx_INTCLR)

地址偏移: 0x14

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													FECLR	TICLR	RICLR
													WO	WO	WO

位	标记	功能描述	复位值	读写
31:3	—	保留	0x0	—
2	FECLR	清除接收帧错误标志位；写 1 清零，写 0 无效	0x0	WO
1	TICLR	清除发送完成中断标志位；写 1 清零，写 0 无效	0x0	WO
0	RICLR	清除接收完成中断标志位；写 1 清零，写 0 无效	0x0	WO

22.8.7 UART 波特率控制寄存器(UARTx_BAUDCR)

地址偏移: 0x18

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:17	—	保留	0x0	—
16	SELF_BRG	UART 波特率选择位 0: UART 的波特率由 timer 产生 1: UART 的波特率由 $(BAUD+1)*F_{pclk}/(32*(BRG[15:0]+1))$ 生成	0	R/W
7:0	BRG[15:0]	UART 自动波特率生成配置位 波特率 $= (BAUD+1)*F_{pclk}/(32*(BRG[15:0]+1))$	0x0	R/W

23. 低功耗通用异步收发器(LPUART)

23.1 概述

本产品带有 1 个 LPUART 模块，支持半双工和全双工传输；支持 8BIT、9BIT 数据格式；支持 MODE0/1/2/3 四种不同传输模式；LPUART 的波特率由 LPTIM 产生，也可以由内部自动波特率发生器产生；支持多机通讯模式；支持自动地址识别；支持给定地址和广播地址，支持低功耗模式。

LPUART 为支持低功耗应用，除了原本的 PCLK 时钟外，增加了一路 SCLK 时钟，并可以控制 LPUART 工作状态。LPUART 模块内部寄存器配置逻辑工作于 PCLK 时钟域，数据收发逻辑工作于 SCLK 时钟域。当系统进入低功耗模式且后且处在 LPUART 工作状态，关闭高频 PCLK 时钟，打开低频 SCLK 时钟，LPUART 仍旧可以进行正常的数据收发。关闭工作状态则停止波特率的生成。

SCLK 时钟来源可以选择：PCLK、外部低速时钟(XTL)，内部低速时钟(RCL)。在 LPMODE=1 时，SCLK 时钟还支持 1/2/4/8/16/32/64/128 倍的预分频。

注意，在 LPMODE=0 时，LPUART 接收的是 LPTIM 时钟的 TOGGLE 输出信号而非 OVERFLOW 信号，因此必须使能 LPTIM 的 TOGGLE 输出。

23.2 结构框图

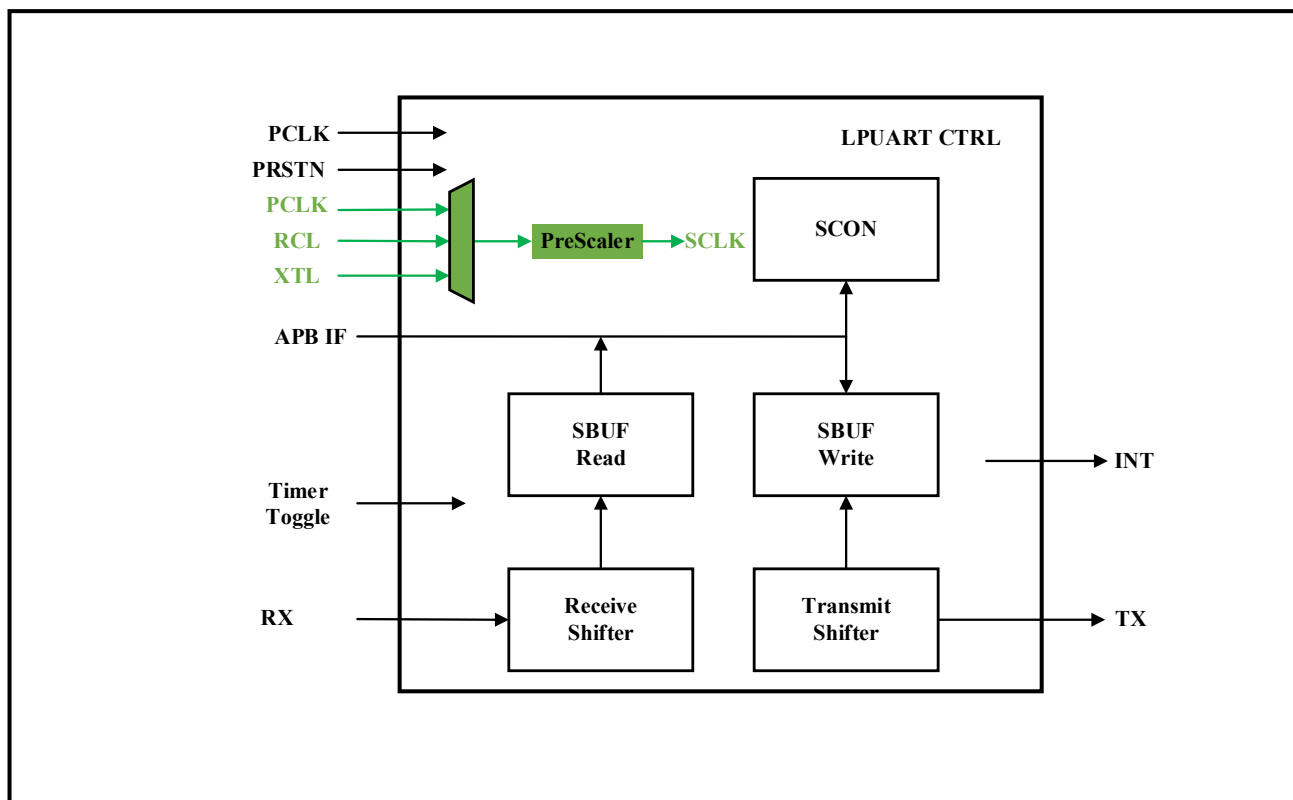


图 23-1. LPUART 结构框图

23.3 工作模式

与通用 UART(UART0/1)相比, LPUART 增加了一个 LPMODE 控制位。当该位置“1”时, 只支持 MODE1/3 工作模式, 并且波特率生成方式也会发生改变。具体描述请参考以下章节。

23.3.1 MODE0(同步模式, 半双工)

当工作在 MODE0 时, UART 工作在同步模式, 其波特率为固定的 SCLK 时钟的 1/12。UART 接收数据由 RXD 输入、UART 发送数据有 RXD 输出, RXD 此时为输入输出端口。UART 同步移位时钟由 TXD 输出, TXD 此时为输出端口。注意, 本模式只能作为主机发送同步移位时钟, 不可以作为从机从外部接收该时钟。该模式下, 传输的数据位宽只能是 8 位的, 没有起始位和结束位。

将 LPUARAT_SCON.SM0 和 LPUART_SCON.SM1 清零, 可进入 MODE0 工作模式。

当 LPMODE=1 时, 不支持 MODE0 工作模式。

23.3.1.1 发送数据

发送数据时，清除 LPUART_SCON.REN 位，并将数据写入 SBUF 寄存器。此时，发送数据将从 RXD 输出(低位在先，高位在后)，同步移位时钟从 TXD 输出。

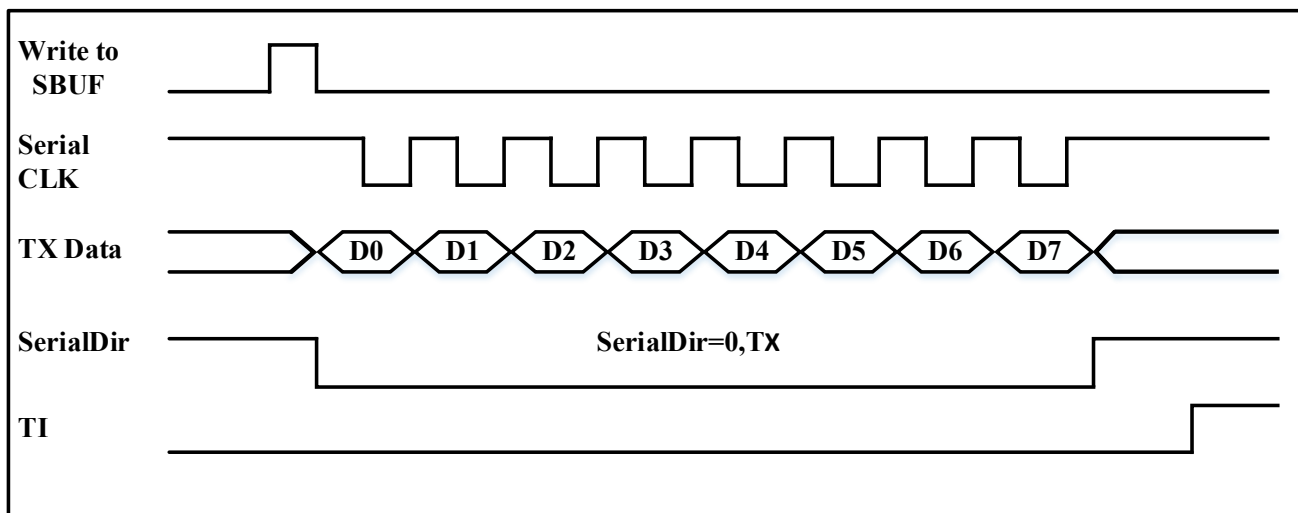


图 23-2. MODE0 发送数据

23.3.1.2 接收数据

接收数据时，将 LPUART_SCON.REN 位置 1，并将 LPUART_ISR.RI 位清零。当接收结束，数据可从 LPUART_SBUF 寄存器读出。此时，接收数据从 RXD 输入(低位在先，高位在后)，同步移位时钟从 TXD 输出。

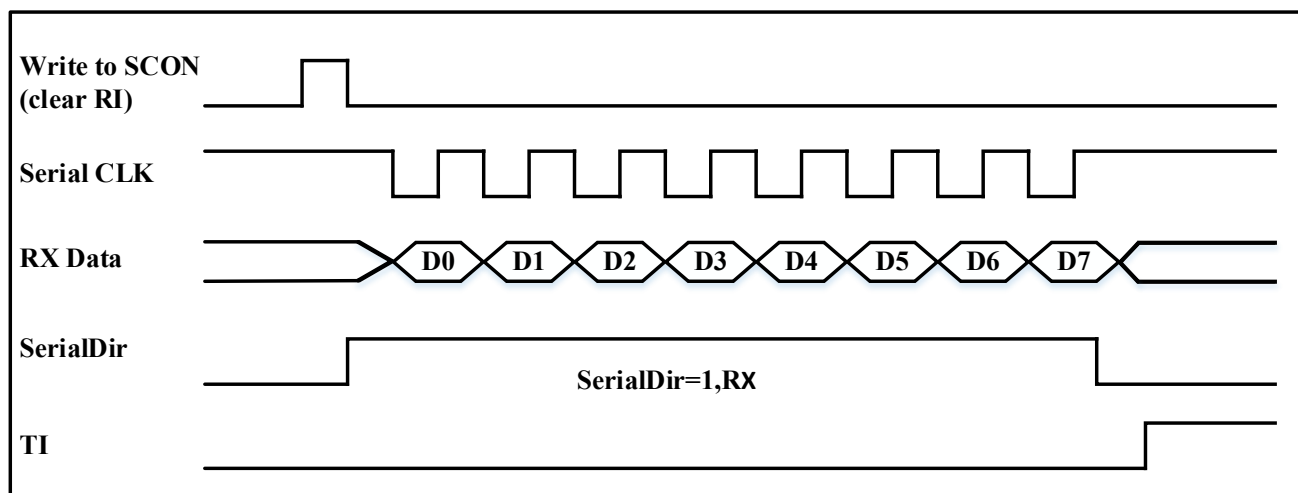


图 23-3. MODE0 接收数据

23.3.2 MODE1(异步模式，全双工)

当工作在 MODE1 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由 10

位组成：起始位“0”开始，紧接着 8 位数据位(低位在先，高位在后)，最后是结束位“1”。将 LPUART_SCON.SM0 清 0，LPUART_SCON.SM1 置 1，可进入 MODE1 工作模式。该模式下，当 LPMODE=0 时，LPUART 的波特率可以选择由自动波特率发生器或者定时器 LPTIM 模块产生，并且是可编程的。

当 LPMODE=1 时，波特率计算方式发生改变，具体参考波特率编程章节。

23.3.2.1 发送数据

发送数据时，与 LPUART_SCON.REN 的值无关，将所发送数据写入 LPUART_SBUF 寄存器中，数据就会从 TXD 移出(低位在先，高位在后)。

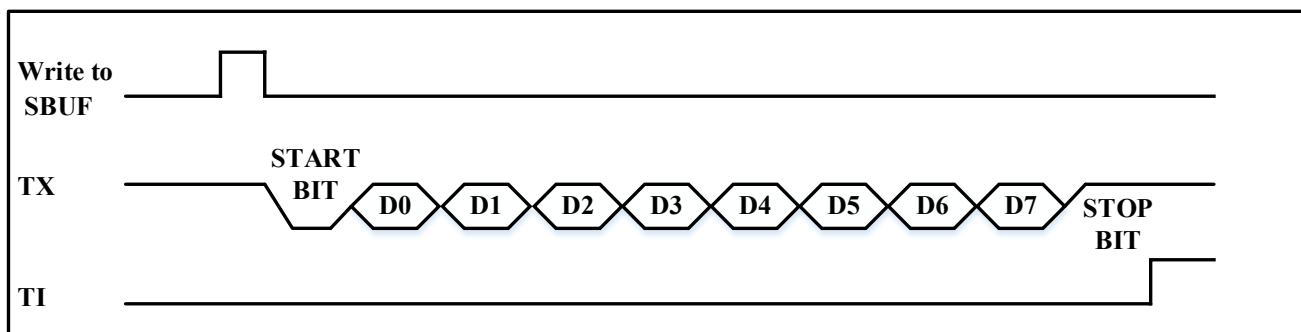


图 23-4. MODE1 发送数据

23.3.2.2 接收数据

接收数据时，需将 LPUART_SCON.REN 位置 1，并将 LPUART_ISR.RI 位清 0。开始接收 RXD 上数据(低位在先，高位在后)，当接收完毕，可以从 LPUART_SBUF 寄存器读出。

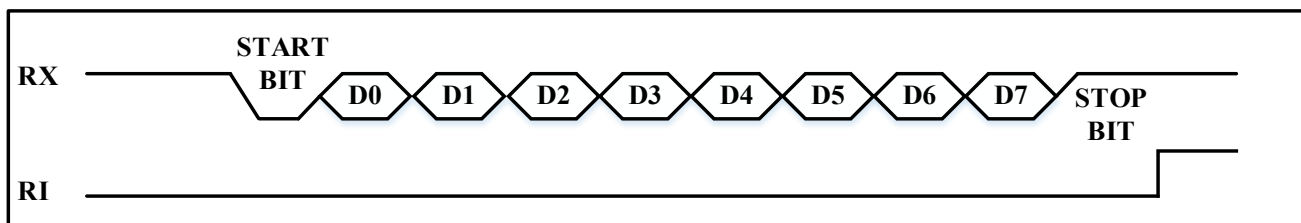


图 23-5. MODE1 接收数据

23.3.3 MODE2(异步模式，全双工)

当工作在 MODE2 时，发送数据通过 TXD 发送，接收数据通过 RXD 接收。该数据由 11 位组成：起始位“0”开始，接着是 8 个数据位，1 个 TB8 位和结束位。额外的 TB8 位是用来在多机通讯环境下使用，当 TB8=1，表明所接收的是地址帧；当 TB8=0，表明所接收的是数据

帧。当不需要多机通讯时，此位也可以作为奇偶校验位来使用。

将 LPUART_SCON.SM0 置 1，LPUART_SCON.SM1 清 0，可进入 MODE2 工作模式。该模式下，波特率可以独立产生，不需要外部 TIMER 产生。

当 LPMODE=1 时，不支持 MODE2 工作模式。

23.3.3.1 发送数据

发送数据时，与 LPUART_SCON.REN 的值无关，并将所发送数据写入 LPUART_SBUF 寄存器中，数据就会从 TXD 移出(低位在先，高位在后)。

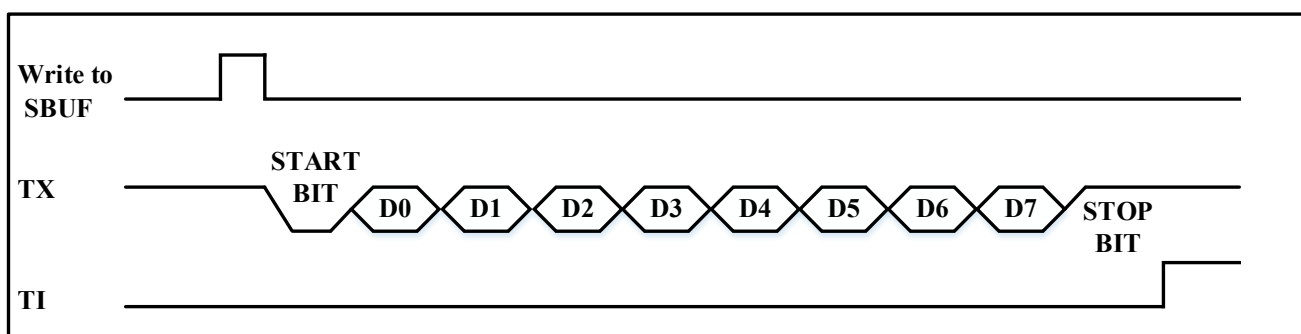


图 23-6. MODE2 发送数据

23.3.3.2 接收数据

接收数据时，需将 LPUART_SCON.REN 位置 1，并将 LPUART_ISR.RI 位清 0。开始接收 RXD 上数据(低位在先，高位在后)，当接收完毕，可以从 LPUART_SBUF 寄存器读出。

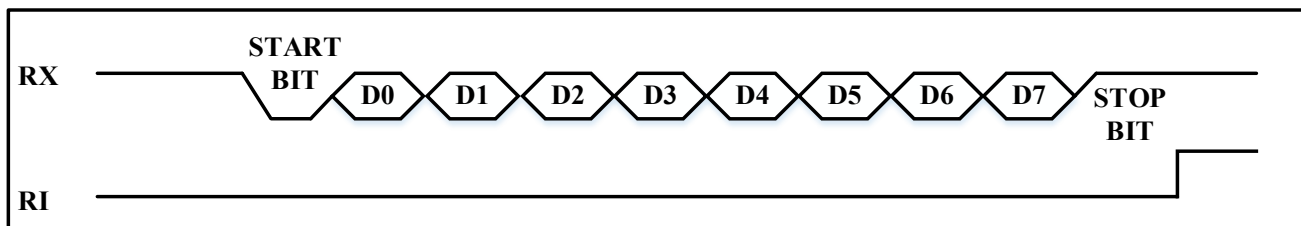


图 23-7. MODE2 接收数据

23.3.4 MODE3(异步模式，全双工)

MODE3 的数据格式，传输时序以及操作方式都与 MODE2 相同，唯一的区别是 MODE3 的波特率由 LPTIM 产生或者内部自动波特率发生器产生，而不是像 MODE2 由设备自己独立产生。MODE3 的波特率是可编程的，波特率生成方式与 MODE1 相同。

将 LPUART_SCON.SM0 置 1，LPUART_SCON.SM1 置 1，可进入 MODE3 工作模式。当

LPMODE=1 时，支持 MODE3 工作模式。但是波特率计算方式发生改变，具体参考波特率编程章节。

23.4 波特率编程

23.4.1 MODE0

LPMOD=0

当工作在 MODE0 时，波特率被固定在 PCLK 的 1/12，不需要 LPTIM 的支持。

LPMODE=1

当 LPMODE=1 时，不支持该模式。

23.4.2 MODE1/3

LPMODE=0

当工作在 MODE1 或者 MODE3 时，波特率可以由 LPTIM 的溢出时间决定。具体公式如下：

$$\text{BaudRate} = \frac{(LPUARTx_SCON.BAUD + 1) * F_{sclk}}{32 * (2^{16} - LPTIM_BGLOAD[15:0])}$$

其中，LPUART_SCON.DBAUD 表示双倍波特率，FSCLK 为 SCLK 时钟频率，LPTIM_BGLOAD 为 LPTIM 的周期装载计数值。

注意，LPTIM 必须配置为 16 位自动重载模式，立即重载寄存器(LPTIM_LOAD)和周期重载寄存器(LPTIM_BGLOAD)要写入相同的初始值。

也可以使用自身波特率生成模式

$$\text{BAUDRATE} = \frac{(LPUART_SCON.DBAUD + 1) * F_{sclk}}{32 * (LPUART_BAUDCR.BRG + 1)}$$

其中，UARTX_SCON.DBAUD 表示双倍波特率，F_{sclk} 为 SCLK 时钟频率。

LPMODE=1

当 LPMODE 设为“1”时，波特计算公式与上述公式不同，简化为：

$$\text{BAUDRATE} = \frac{F_{sclk}}{4 * \text{LPUART_SCON.PRSC}}$$

其中，F_{sclk} 为 SCLK 时钟频率，LPUART_SCON.PRSC 为预分频系数。

23.4.3 MODE2

LPMODE=0

当工作在 MODE2 时，传输时钟只能选择 PCLK，波特率被固定在如下公式所得值：

$$\text{BAUDRATE} = \frac{(\text{LPUART_SCON.DBAUD} + 1) * F_{sclk}}{64}$$

其中，LPUART_SCON.DBAUD 表示双倍波特率，F_{pclk} 为 PCLK 时钟频率。

LPMODE=1

当 LPMODE=1 时，不支持该模式。

23.5 帧错误检测

MODE1/2/3 具有帧错误检测功能，硬件会自动检测接收到的帧数据是否带有效的 STOP 位。

如果没有收到有效 STOP 位，则 LPUART_ISR.FE 置 1。LPUART_ISR.FE 位由硬件置 1，软件清 0，如果软件未及时清 0，则后续收到数据即使带有效 STOP 位，也不会把 LPUART_ISR.FE 标志清 0。

23.6 多机通讯

MODE2/3 具有多机通讯功能，为此在其帧格式中增加了 1 位 TB8/RB8。将 LPUART_SCON.SM2 置“1”，可开启多机通讯位。当开启多机通讯位后，发送数据时，主机可以通 LPUART_SCON.TB8 来区分当前帧是地址帧(LPUART_SCON.TB8=1)还是数据帧(LPUART_SCON.TB8=0)。接收数据时，从机会忽略 RB8 位(第 9 位)为“0”的当前接收帧。当收到帧的 RB8 位(第 9 位)为“1”表明其是地址帧，从机会继续判断接收到的地址与其自身地址是否相等。如果匹配，则从机会对 LPUART_SCON.RB8 置“1”，并对 LPUART_ISR.RI 置“1”，以表明该帧为地址帧并且地址已经匹配。从机软件看到 LPUART_SCON.RB8=1 并且

LPUART_ISR.RI=1 后, 先把 LPUART_SCON.SM2 位清“0”, 然后准备接受给它的数据帧。如果地址不等, 表明主机并不是寻址该从机, 从机硬件保持 LPUART_SCON.RB8 和 LPUART_ISR.RI 为“0”, 软件保持 LPUART_SCON.SM2 位为“1”, 从机继续处于地址监听状态。

23.7 自动地址识别

当开启多机通讯位后(LPUART_SCON.SM2 置“1”), 自动地址识别功能也将开启。该功能由硬件实现, 使得从机可以检测接收到每个地址帧, 如果该地址与从机地址匹配, 接收端会给出 LPUART_ISR.RI 接收标志。如果地址不匹配, 则接收端不会给出任何接收标志。如果有需要, 也可以在 MODE1 下开启多机通讯位, 此时 TB8 位由 STOP 位代替。当从机接收到匹配的地址帧和有效的 STOP 位时, LPUART_ISR.RI 会被置“1”。为了支持自动地址识别, 定义了广播地址和给定地址的概念。

23.8 给定地址

LPUART 设备的 LPUART_SADDR 寄存器用来表示自己的设备给定地址, LPUART_SADEN 寄存器是地址掩码, 可以用来定义地址中的无关位。当 LPUART_SADEN 的某一位为“0”, 表示该位地址为无关位, 也就是说在地址匹配过程中, 该位地址不参与地址匹配。这些无关位增加了寻址的灵活性, 使得主机可以同时寻址一个或者多个从机设备。

注意, 如果需要给出唯一匹配地址, LPUART_SADEN 寄存器必须设为 8' HFF。

GIVENADDR=SADDR&SADEN

23.9 广播地址

广播地址是用来同时寻址所有从机设备的, 一般广播地址为 8' HFF。

BoardCastAddr=SADDR|SADEN

23.10 举例

假设某从机的 LPUART_SADDR 和 LPUART_SADEN 配置如下：

SADDR: 0b01101001

SADEN: 0b11111011

那么其给定地址和广播地址如下：

Given: 0b01101X01

BroadCast: 0b11111X11

可见，主机可以用四个地址寻址到本从机，分别是：

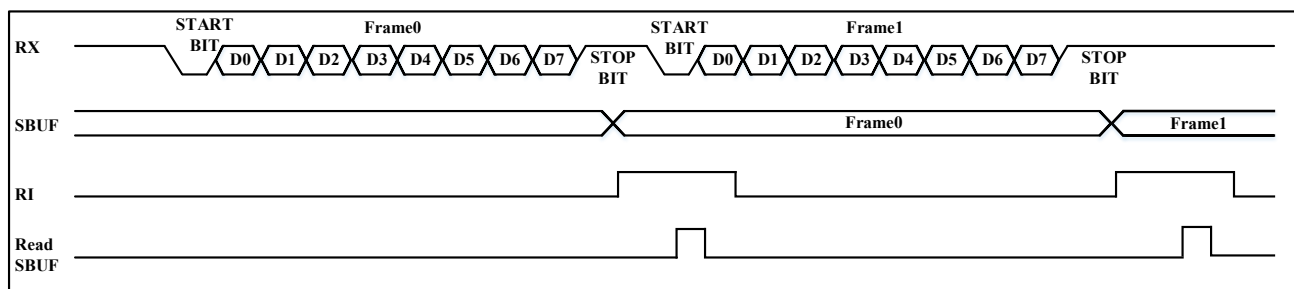
0b01101001 和 0b01101101(given address)

0b11111011 和 0b11111111(broadcast address)。

23.11 收发端缓存

23.11.1 接收缓存

LPUART 接收端有一个帧长度(8/9BITS)的接收缓存，也就是说当一帧数据接收完毕后，接收缓存中的数据会被一直保持，直到下一帧数据的 STOP 位接收完毕后，接收缓存才会更新为新一帧数据。



23.11.2 发送缓存 LPUART

发送端不支持发送缓存。如果在发送数据过程中，填写 LPUART_SBUF 寄存器，将会屏蔽该写操作。软件应该避免这种操作。

23.12 寄存器列表

LPUART 基地址: 0x4000_5000

偏移地址	名称	描述	复位值
0x00	LPUART_SCON	控制寄存器	0x0000E000
0x04	LPUART_SBUF	数据寄存器	0x00000000
0x08	LPUART_SADDR	地址寄存器	0x00000000
0x0C	LPUART_SADEN	地址掩码寄存器	0x00000000
0x10	LPUART_INTSR	中断标志位寄存器	0x00000000
0x14	LPUART_INTCLR	中断标志位清除寄存器	0x00000000
0x18	LPUART_BAUDCR	波特率控制寄存器	0x00000000

23.13 寄存器说明

23.13.1 LPUART 控制寄存器(LPUART_SCON)

地址偏移: 0x00

复位值: 0x0000E000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															EN
															R/W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRSC[2:0]			SCLK SEL		LPM ODE	DBA UD	TE EN	SM0:SM1		SM 2	RE N	TB 8	RB 8	TI EN	RI EN
R/W			R/W		R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:17	—	保留	0x0	—
16	EN	Low-Power UART 工作使能 0: Low-Power UART 关闭, 不接收/发送数据 1: Low-Power UART 使能, 进行数据传输之前必须将该位置 1	0	R/W
15:13	PRSC[2:0]	传输时钟 SCLK 预分频选择: 000: DIV128; 001: DIV64; 010: DIV32; 011: DIV 16; 100: DIV 8; 101: DIV 4; 110: DIV2; 111: DIV1 PRS[2:0]只有当 LPMODE=1 时有效; 当 LPMODE=0 时, PRS[2:0]不会对 SCLK 预分频。	0x0	R/W
12:11	SCLKSEL[1:0]	传输时钟 SCLK 选择	0x0	R/W

位	标记	功能描述	复位值	读写																									
		00,01: PCLK; 10: XTL; 11: RCL。																											
10	LPMODE	低功耗模式 0: 正常工作模式; 1: 低功耗工作模式。	0	R/W																									
9	DBAUD	双倍波特率 0: 单倍波特率; 1: 双倍波特率。	0	R/W																									
8	TEEN	发送缓存空中断使能 0: DISABLE; 1: ENABLE。	0	R/W																									
7:6	SM0:SM1	工作模式 00: MODE0; 01: MODE1; 10: MODE2; 11: MODE3 <table border="1"> <thead> <tr> <th>SM0</th><th>SM1</th><th>MODE</th><th>描述</th><th>波特率</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>位移寄存器</td><td>PCLK/12</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>8 位串口传输</td><td>可变波特率</td></tr> <tr> <td>1</td><td>0</td><td>2</td><td>9 位串口传输</td><td>PCLK/32,PCLK/64</td></tr> <tr> <td>1</td><td>1</td><td>3</td><td>9 位串口传输</td><td>可变波特率</td></tr> </tbody> </table>	SM0	SM1	MODE	描述	波特率	0	0	0	位移寄存器	PCLK/12	0	1	1	8 位串口传输	可变波特率	1	0	2	9 位串口传输	PCLK/32,PCLK/64	1	1	3	9 位串口传输	可变波特率	0x0	R/W
SM0	SM1	MODE	描述	波特率																									
0	0	0	位移寄存器	PCLK/12																									
0	1	1	8 位串口传输	可变波特率																									
1	0	2	9 位串口传输	PCLK/32,PCLK/64																									
1	1	3	9 位串口传输	可变波特率																									
5	SM2	多主机通讯: 0: DISABLE, 1: ENABLE SM2: 软件配置多机通讯以及自动地址匹配模式 1: 启动多从机通讯以及地址自动匹配 0: 关闭多从机通讯以及地址自动匹配在模式 2 和模式 3 中: 如果 SM2=1, 并且 REN=1, 则接收机处于地址帧监测模式, 可以使用接收到的第 9 位 RB8 来进行地址筛选。 RB8=1 为地址帧, 通讯数据可以进入 SBUF, 置位 RI, 进入中 断服务程序中进行地址比较; RB8=0 为数据帧, 接收机忽略这 些数据帧并保持 RI=0 如果 SM2=0, 并且 REN=1, 则接收机不使用地址监测模式, 无论收到的 RB8 为 0 或 1, 都直接接收并且进入 SUBF, 置位 RI, RB8 在这种模式下为校验位。	0	R/W																									
4	REN	接收使能 MODE0: 0: 发送, 1: 接收 其他: 0: 发送, 1: 接收/发送	0	R/W																									
3	TB8	发送 TB8 位	0	R/W																									
2	RB8	接收 RB8 位	0	R/W																									
1	TIEN	接收完成中断使能 0: DISABLE; 1: ENABLE	0	R/W																									
0	RIEN	接收完成中断使能 0: DISABLE; 1: ENABLE	0	R/W																									

23.13.2 LPUART 数据寄存器(LPUART_SBUF)

地址偏移: 0x04

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	SBUF[7:0]	发送数据时, 当发送数据写入该寄存器; 接收数据时, 数据接收完毕后, 从该寄存器中读出。	0x0	R/W

23.13.3 LPUART 地址寄存器(LPUART_SADDR)

地址偏移: 0x08

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	SADDR[7:0]	从机设备地址寄存器	0x0	R/W

23.13.4 LPUART 地址掩码寄存器(LPUART_SADEN)

地址偏移: 0x0C

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	SADEN[7:0]	从机设备地址掩码寄存器	0x0	R/W

23.13.5 LPUART 标志位寄存器(LPUART_INTSR)

地址偏移：0x10

复位值：0X00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												TE	FE	TI	RI
												RO	RO	RO	RO

位	标记	功能描述	复位值	读写
31:4	-	保留	0x0	-
3	TE	发送缓存空中断标志位，硬件置位，硬件清零。 注意：当该位值为“0”时，硬件自动屏蔽软件写 SBUF 操作。 0：TE 中断无效 1：TE 中断有效	0	RO
2	FE	接收帧错误标志位，硬件置位，软件清零 0：FE 中断无效 1：FE 中断有效	0	RO
1	TI	发送完成中断标志位，硬件置位，软件清零 0：TI 中断无效 1：TI 中断有效	0	RO
0	RI	接收完成中断标志位，硬件置位，软件清零 0：RI 中断无效 1：RI 中断有效	0	RO

23.13.6 LPUART 标志位清除寄存器(LPUART_INTCLR)

地址偏移: 0x14

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													FEC LR	TIC LR	RIC LR
													WO	WO	WO

位	标记	功能描述	复位值	读写
31:3	-	保留	0x0	-
2	FECLR	清除接收帧错误标志位; 写 1 清零, 写 0 无效	0x0	WO
1	TICLR	清除发送完成中断标志位; 写 1 清零, 写 0 无效	0x0	WO
0	RICLR	清除接收完成中断标志位; 写 1 清零, 写 0 无效	0x0	WO

23.13.7 波特率控制寄存器(LPUART_BAUDCR)

地址偏移: 0x18

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:17	-	保留	0x0	-
16	SELF_BRG	LPUART 波特率选择位: 0: LPUART 的波特率由 LPTIMER 产生 1: LPUART 的波特率由(BAUD+1)*Fclk/(32*(BRG[15:0]+1))生成	0	R/W
15:0	BRG[15:0]	LPUART 自动波特率生成配置位: 波特率=(BAUD+1)*Fclk/(32*(BRG[15:0]+1))	0x0	R/W

24. I2C 接口

24.1 I2C 简介

I2C 是双线双向的串行总线，它为设备之间数据交换提供了一种简单高效的方法。I2C 标准是一个具有冲突检测机制和仲裁机制的真正意义上的多主机总线。它能防止两个或者多个主机在同时请求控制总线时发生数据冲突。I2C 总线控制器，能满足 I2C 总线的各种规格并支持所有与 I2C 总线通信的传输模式。I2C 总线使用连接设备的” SCL” (串行时钟总线)和” SDA” (串行数据总线)来传送信息。数据在主机与从机之间通过 SCL 时钟线控制在 SDA 数据线上实现一个字节一个字节的同步传输，每个字节为 8 位长度，一个 SCL 时钟脉冲传输一个数据位，数据由最高位 MSB 开始传输，每个传输字节后跟随一个应答位，每个位在 SCL 为高时采样；因此，SDA 线只有在 SCL 为低时才可以改变，在 SCL 为高时 SDA 保持稳定。当 SCL 为高时，SDA 线上的跳变视为命令中断(START 或 STOP)，I2C 逻辑能自主地处理字节的传输。它能保持跟踪串行传送，而且还有一个状态寄存器(I2C_STAT)能反映 I2C 总线控制器和 I2C 总线的状态。

24.2 I2C 主要特性

I2C 控制器支持以下特性：

- 支持主机发送/接收，从机发送/接收四种工作模式
- 支持标准(100Kbps)/快速(400Kbps)/高速(1Mbps)三种工作速率
- 支持 7 位寻址功能
- 支持噪声过滤功能
- 支持广播地址
- 支持中断状态查询功能

24.3 I2C 协议描述

通常标准 I2C 传输协议包含四个部分：

- 1)起始信号或重复起始信号
- 2)从机地址传输和 R/W 位传输
- 3)数据传输
- 4)停止信号

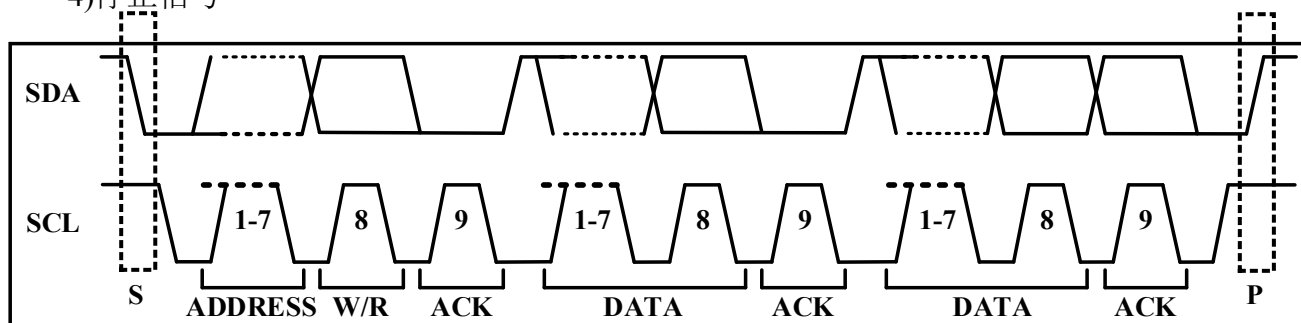


图 24-1. I2C 传输协议

24.3.1 I2C 总线上数据传输

主机发出从机接收 7 位地址(一个字节)，传输方向未改变。

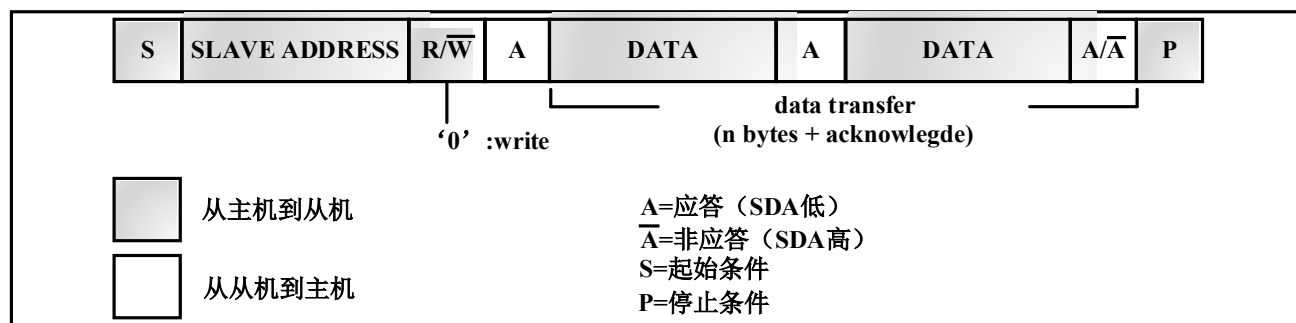


图 24-2. 主机向从机传输数据

第一个字节后主机紧接着由从机读取数据(内容为从机地址)，传输方向改变。

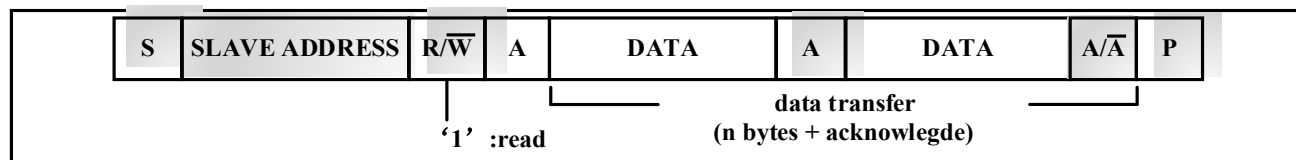


图 24-3. 主机由从机读取地址

24.3.2 起始位或重复起始信号

当总线处于空闲状态下,说明没有主机对总线发起传输请求(SCL 和 SDA 线同时为高),主机可以通过发送一个 START 信号来发起传输请求。

起始信号:通常表示为 S-bit,当 SCL 线为高时,SDA 线上信号由高至低,标示总线上产生起始信号,新的传输开始。

重复起始信号(Sr):即在两个 START 信号之间没有 STOP 信号。主机采用这种方法与另一个从机或相同的从机以不同传输方向进行通信(例如:从写入设备到从设备读出)。而不释放总线。

STOP 信号:主机向总线发出停止信号结束数据传送。停止信号,通常用 P-bit 表示,当 SCL 线为高时,SDA 线上出现由低到高的信号,被定义为停止信号。

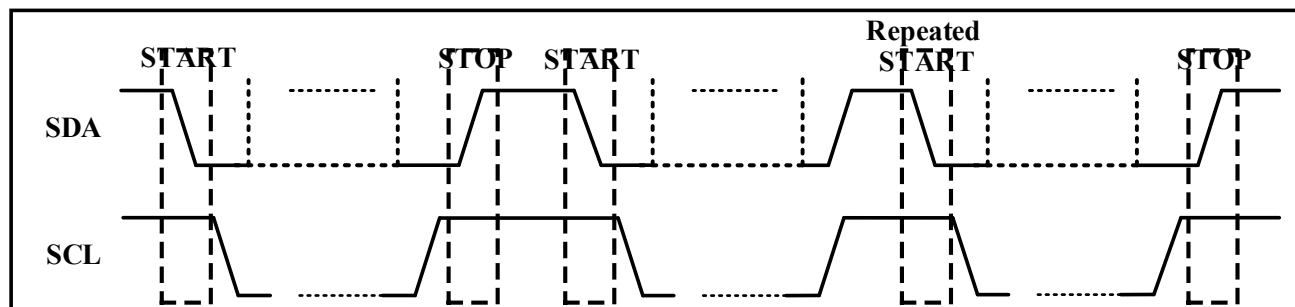


图 24-4. START 和 STOP 条件

24.3.3 从机地址传输

START 信号是从机地址时,主机立即传输数据的第一位。这是一个跟随有一个 RW 位的 7 位调用地址, RW 位控制从机的信号传输方向。系统中没有两个从机有相同的地址,只有被主机寻址的从机会通过在第 9 个 SCL 时钟周期将 SDL 置为低电平作为应答。

24.3.4 数据传输

当从机地址被成功识别,就可以根据 RW 所决定的方向,开始一字节一字节的数据传输,每个传输字节最后带一个第 9 时钟周期上的响应信号,如果从机上产生无响应信号(NACK),主机可以产生停止信号来退出数据传输,或者产生重复起始信号开始新一轮的数据传输。当主机作为接收器件时,发生无响应信号(NACK),从机释放 SDA 线,使主机产生停止信号或重复

起始信号。

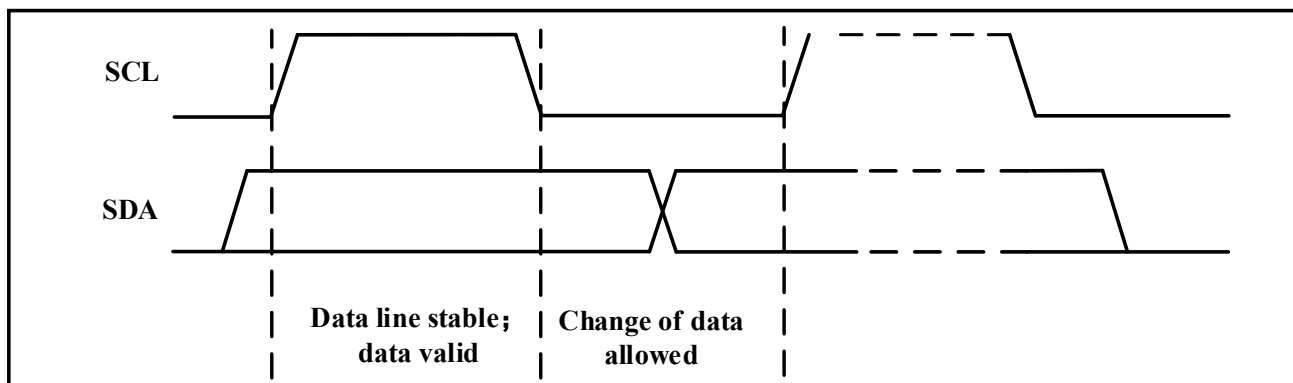


图 24-5. I2C 总线上位传输

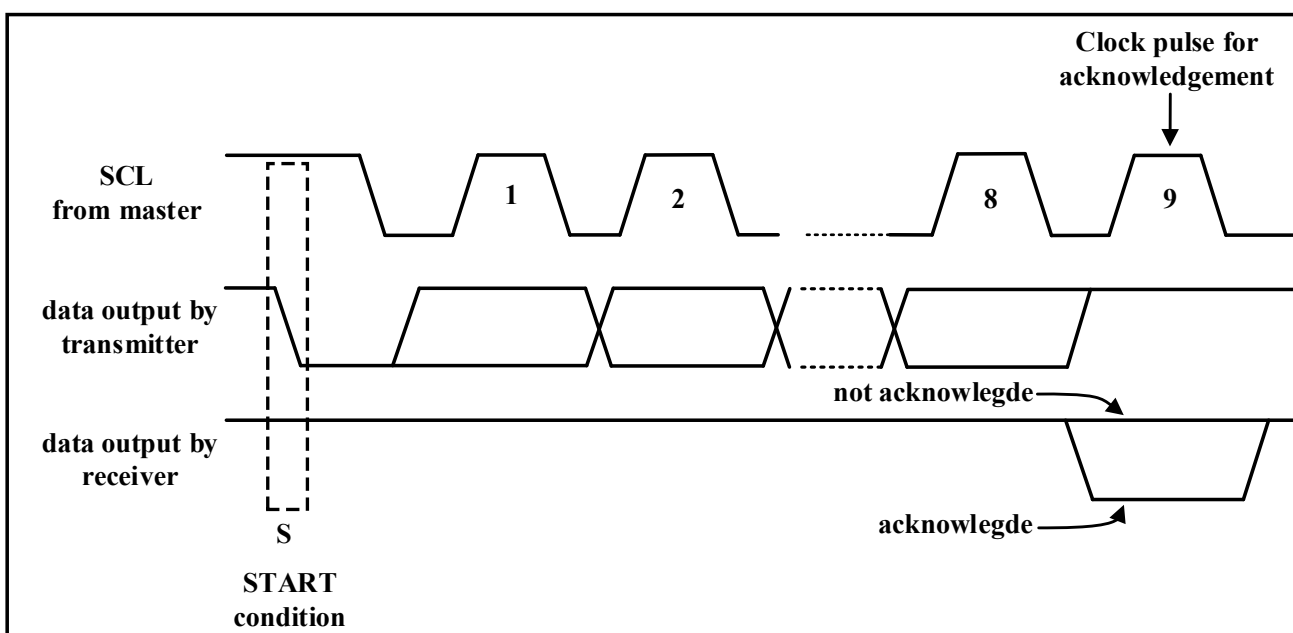


图 24-6. I2C 总线上应答信号

24.4 I2C 功能描述

I2C 总线使用双线在连接到总线“SCL”(串行时钟线)和“SDA”(串行数据线)的设备间传送信息。由于只有无方向端口，I2C 组件需要使用到引脚的漏端开路缓冲器。每个连接到总线的设备都能使用软件通过特定地址寻址。I2C 标准是一个具有冲突检测机制和仲裁机制的真正意义上的多主机总线。它能防止两个或者多个主机在同时开始传输数据时发生数据冲突。滤波逻辑可以过滤数据总线上的毛刺来保护数据的完整性。

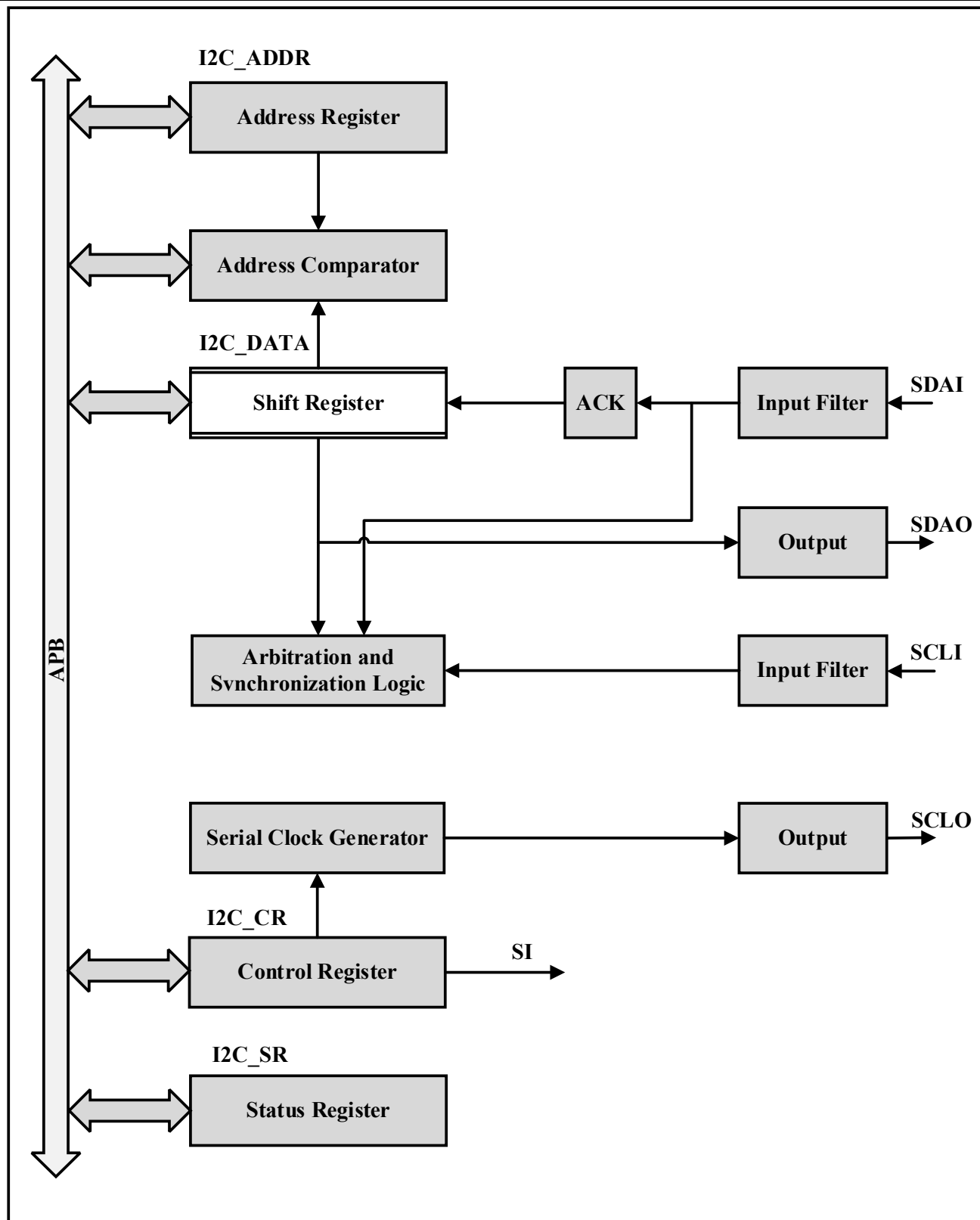


图 24-7. I2C 功能模块图

24.4.1 I2C 工作模式

I2C 组件可实现 8 位的双向数据传输，传输速率在标准模式下可达到 100Kbits/s 而在高速

模式下可达 400Kbits/s，在超高速模式下可达 1Mbits/s，并且可以在以下四种模式下工作：

- 1) 主机发送模式：当“SCL”输出串行时钟信号时“SDA”输出串行数据。
- 2) 主机接收模式：当“SCL”输出串行时钟信号时串行数据通过“SDA”接收。
- 3) 从机接收模式：串行数据和串行时钟分别通过“SDA”和“SCL”接收。
- 4) 从机发送模式：当串行时钟从“SCL”口输入时串行数据通过“SDA”口发送。

24.4.2 仲裁与同步逻辑

在主发送模式中，仲裁逻辑检查每个发送的逻辑 1 是否真正出现在总线上。如果总线的另一个器件撤消了一个逻辑 1 并将 SDA 线拉低，仲裁丢失，I2C 模块立刻由主发送器变为从接收器。I2C 模块将继续输出时钟脉冲(在 SCL 上)，直至发送完当前的串行字节。仲裁也可能在主接收模式中丢失。这种情况只在 I2C 模块正在向总线返回一个“非应答(逻辑 1)”时出现。当总线的另一个器件将信号拉低时仲裁丢失。由于它只在串行字节结束时出现，因此 I2C 模块不会再产生时钟脉冲。

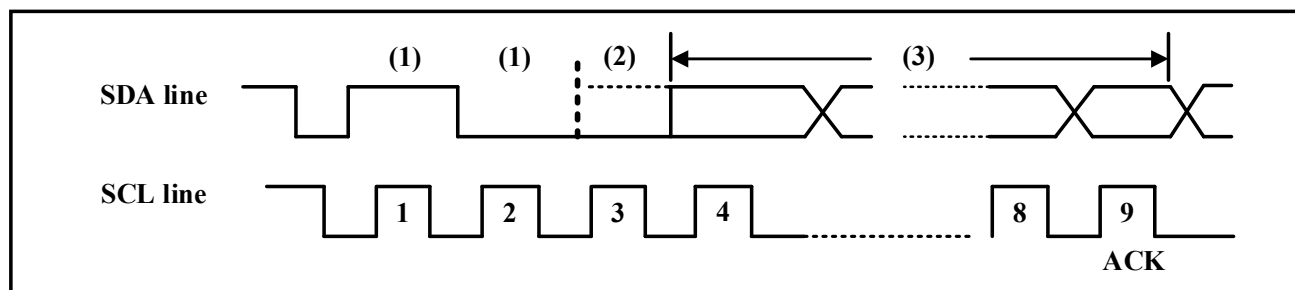


图 24-8. I2C 总线上的仲裁

- 1) 另一器件发送串行数据；
- 2) 另一器件通过拉低 SDA 先撤消了该 I2C 主机发送的一个逻辑 1(虚线)。仲裁丢失，I2C 进入从接收模式；
- 3) 此时 I2C 处于从接收模式，但仍产生时钟脉冲，直至发送完当前字节。I2C 将不为下个字节的传输产生时钟脉冲。一旦赢得仲裁，SDA 上的数据传输由新的主机来启动。同步逻辑使得串行时钟发生器与另一个器件 SCL 线上的时钟脉冲同步。如果 2 个或更多主器件产生时钟脉冲，则高电平周期取决于产生最短高电平时间的器件；低电平周期

取决于产生最长低电平时间的器件。

24.4.3 串行时钟发 Th 器

串行时钟发生器采用一个 8 位的计数器作为波特率发生器，SCL 信号和 pclk 信号的频率关系为 $F_{scl}=F_{pclk}/8*(N+1)$

下面的表格表示 pclk 为各种频率时，分频系数为 1-7 时，SCL 信号的频率值。

表 24.1. I2C 时钟信号波特率

频率(Khz)	1	2	3	4	5	6	7
1000	62	41	31	25	20	17	15
2000	125	83	62	50	41	35	31
4000	250	166	125	100	83	71	62
6000	375	250	187	150	125	107	93
8000	500	333	250	200	166	142	125
10000	625	416	312	250	208	178	156
12000	750	500	375	300	250	214	187
14000	875	583	437	350	291	250	218
16000	1000	666	500	400	333	285	250

24.4.4 输入滤波器

输入信号与时钟信号(clk)同步，低于 3 个时钟周期的尖峰脉冲信号会被滤除。每个滤波器由 3 个触发器组成。第一个触发器用来直接锁存输入信号，并将数据载入由另外两个构成的移位寄存器中。当第二和第三个触发器的状态是” 11” 或” 00” 时，内部的滤除信号会各自被置 1 或置 0。

24.4.5 地址比较器

I2C 比较器将自己的从机地址与接收到的 7 位从机地址做比较。它可使用” I2C_ADDR” 寄存器对自己的从机地址进行编程。并且会根据” I2C_ADDR” 寄存器的” GC” 位与首次接收到的 8 位字节与通用调用地址(0x00)相比较。如果任何一者相同，” I2C_CR” 寄存器的” si” 位会被置 1 并产生一个中断请求。

24.4.6 中断产 Th 器

I2C 模块的所有四种模式都被使用时，则有 26 种可能的总线状态。当 I2C 进入 26 种状态的 25 种状态时，” I2C_CR” 寄存器的” si” 标志位会被硬件置 1。” si” 位唯一不会被置 1 的状态是:0xF8，这表明没有有效的相关状态信息。” si” 标志位必须通过软件清零。为了清除” si” 位，必须把 0 写入此位。若在” si” 里写 1 不会改变” si” 的值。为了确定中断的实际中断源，中断服务程序在清除” si” 标志位之前，会对 I2C 状态寄存器进行查询。

24.4.7 I2C 主机发送模式

必须将 ens 置 “1” 来使能 I2C 模块。如果 aa 位复位，当另一个器件正变成总线主机时，I2C 模块将不会应答其自身的从机地址或通用调用地址。换句话说，如果 aa 位复位，I2C 接口就不能进入从机模式。Sta, sto 和 si 必须复位。此时，可通过置位 sta 位进入主发送模式。一旦总线空闲，I2C 逻辑会马上测试 I2C 总线并产生一个起始条件。当发送起始条件时，串行中断标志(si)置位，状态寄存器(I2C_STAT)中的状态代码为 0x08。中断服务程序利用该状态代码进入相应的状态服务程序，将从机地址和数据方向位(SLA+W)装入 I2C_DATA。I2C_CR 的 si 位必须在串行传输继续之前复位。当发送完从机地址和方向位且接收到一个应答位时，串行中断标志(si)再次置位，I2C_STAT 中可能是一系列不同的状态代码。主机模式下为 0x18, 0x20 或 0x38，从机模式(aa=1)下为 0x68, 0x78 或 0xB0。每个状态代码对应的操作在下表中详细介绍。在发送完重复起始条件(状态 0x10)后，I2C 模块通过将 SLA+R 装入 I2C_DATA 切换到主接收模式。

表 24.2. I2C 主机发送模式状态表

状 态 代码	I2C 总线 和硬件状态	应用软件响应					I2C 硬件执行的下一个动作
		读/写 I2C_DATA	写 I2C_CR				
			sta	sto	si	aa	
08H	已发送起始条件	装入 SLA+W	X	0	0	X	将发送 SLA+W，接收 ACK
10H	已发送重复起始条件	装入 SLA+W	X	0	0	X	同上
		装入 SLA+R	X	0	0	X	将发送 SLA+R, I2C 自动切换到主接收模式
18H	已 发 送	装入数据字节	0	0	0	X	将发送数据字节，将接收 ACK

状态 代码	I2C 总线 和硬件状态	应用软件响应					I2C 硬件执行的下一个动作
		读/写 I2C_DATA	写 I2C_CR				
			sta	sto	si	aa	
	SLA+W 已 接 收 ACK	无 I2C_DATA 动作	1	0	0	X	将发送重复起始条件
		无 I2C_DATA 动作	0	1	0	X	将发送停止条件，sto 标志位复位
		无 I2C_DATA 动作	1	1	0	X	将先发送停止条件，随后发送起始条件 sto 标志位复位
20H	已 发 送 SLA+W 已接收非 ACK	装入数据字节	0	0	0	X	将发送数据字节，将接收 ACK
		无 I2C_DATA 动作	1	0	0	X	将发送重复起始条件
		无 I2C_DATA 动作	0	1	0	X	将发送停止条件，sto 标志位复位
		无 I2C_DATA 动作	1	1	0	X	将先发送停止条件，随后发送起始条件 sto 标志位复位
28H	已 发 送 I2C_DATA 中的数 据，已接 收 ACK	装入数据字节	0	0	0	X	将发送数据字节，将接收 ACK
		无 I2C_DATA 动作	1	0	0	X	将发送重复起始条件
		无 I2C_DATA 动作	0	1	0	X	将发送停止条件，sto 标志位复位
		无 I2C_DATA 动作	1	1	0	X	将先发送停止条件，随后发送起始条件 sto 标志位复位
		装入数据字节	0	0	0	X	将发送数据字节，将接收 ACK
30H	已 发 送 I2C_DATA 中的数 据	装入数据字节	0	0	0	X	将发送数据字节，将接收 ACK
		无 I2C_DATA 动作	1	0	0	X	将发送重复起始条件
		无 I2C_DATA 动作	0	1	0	X	将发送停止条件，sto 标志位复位
		无 I2C_DATA 动作	1	1	0	X	将先发送停止条件，随后发送起始条件 sto 标志位复位
38H	在 SLA+ R/W 或写 数据字节 时	无 I2C_DATA 动作	0	0	0	X	I2C 总线被释放，进入不可寻址从模式
		无 I2C_DATA 动作	1	0	0	X	当 I2C 总线空闲时发送起始条件

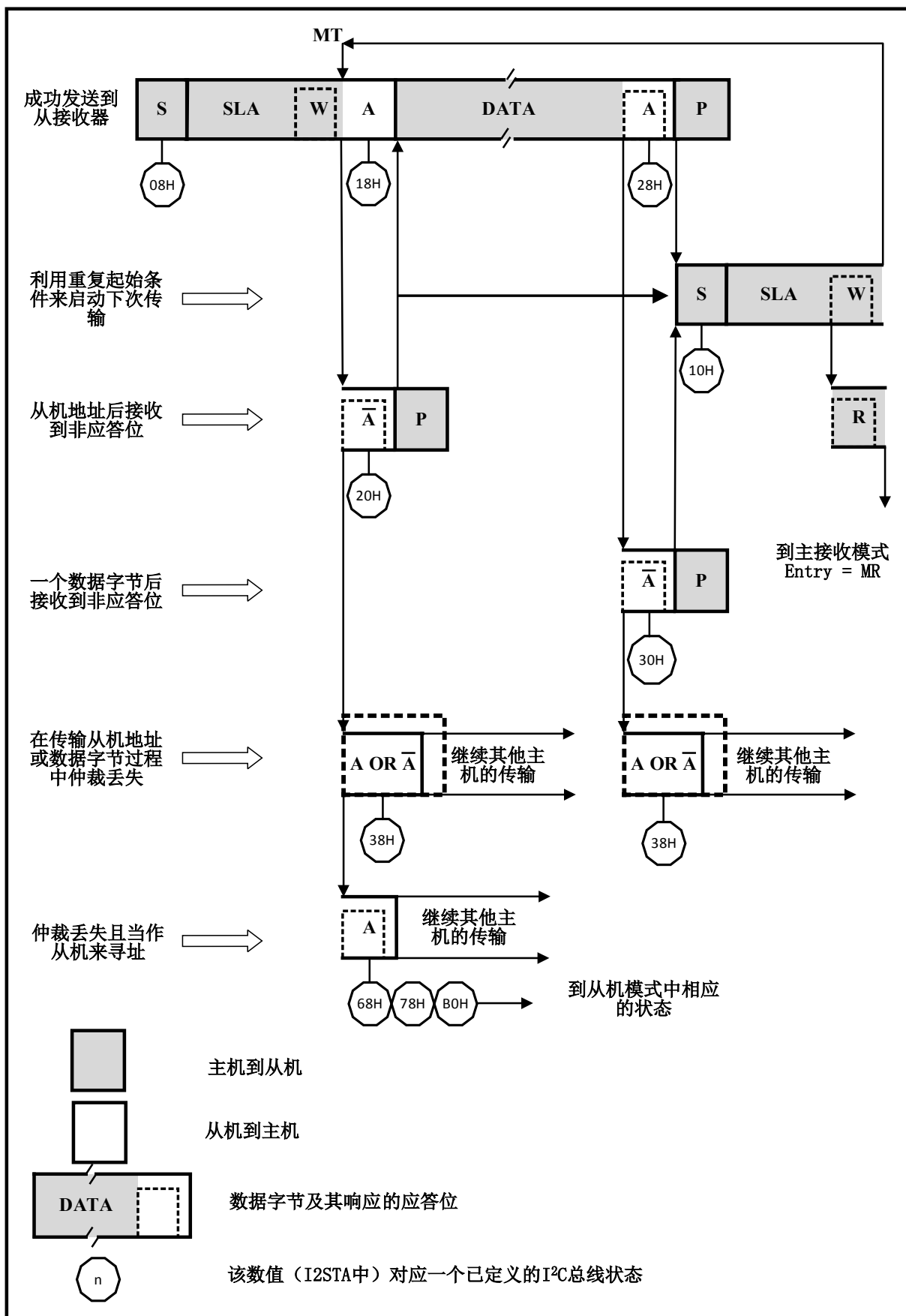


图 24-9. I2C 主机发送状态图

24.4.8 I2C 主机接收模式

在主接收模式中，主机所接收的数据字节来自从发送器。按主发送模式中的方法初始化传输。当发送完起始条件后，中断服务程序必须把 7 位从机地址和数据方向位(SLA+R)装入 I2C_DATA。必须先清除 I2C_CR 中的 si 位，再继续执行串行传输。当发送完从机地址和数据方向位且接收到一个应答位时，串行中断标志 si 再次置位，这时，I2C_STAT 中可能是一系列不同的状态代码。主机模式下为 0x40，0x48 或 0x38，从机模式(aa=1)为 0x68，0x78 或 0xB0。每个状态代码对应的操作详见下表。在发送完重复起始条件(状态 0x10)后，I2C 模块通过将 SLA+W 装入 I2C_DATA 切换到主发送模式。

表 24.3. I2C 主机接收模式状态表

状态 代码	I2C 总线 和硬件状态	应用软件响应					I2C 硬件执行的下一个动作
		读/写 I2C_DATA	写 I2C_CR				
			sta	sto	si	aa	
08H	已发送起始条件	装入 SLA+R	X	0	0	X	将发送 SLA+R，接收 ACK
10H	已发送重复起始条件	装入 SLA+R	X	0	0	X	同上
		装入 SLA+W	X	0	0	X	将发送 SLA+W，I2C 自动切换到主发送模式
38H	在非 ACK 中丢失仲裁	无 I2C_DATA 动作	0	0	0	X	I2C 总线将被释放；进入从模式
		无 I2C_DATA 动作	1	0	0	X	当总线空闲时发起起始条件
40H	已发送 SLA+R 已接收 ACK	无 I2C_DATA 动作	0	0	0	0	将接收数据字节，将返回非 ACK
		无 I2C_DATA 动作	0	0	0	1	将接收数据字节，将返回 ACK
48H	已发送 SLA+R 已接收非 ACK	无 I2C_DATA 动作	1	0	0	X	将发送重复起始条件
		无 I2C_DATA 动作	0	1	0	X	将发送停止条件，sto 标志位复位
		无 I2C_DATA 动作	1	1	0	X	将先发送停止条件，随后发送起始条件，sto 标志位复位
50H	已接收数据字节， ACK 已返回	读取数据字节	0	0	0	0	将接收数据字节，将返回非 ACK
		读取数据字节	0	0	0	1	将接收数据字节，将返回 ACK
58H	已接收数据字节， 非 ACK 已返回	读取数据字节	1	0	0	X	将发送重复起始条件
		读取数据字节	0	1	0	X	将发送停止条件，sto 标志位复位

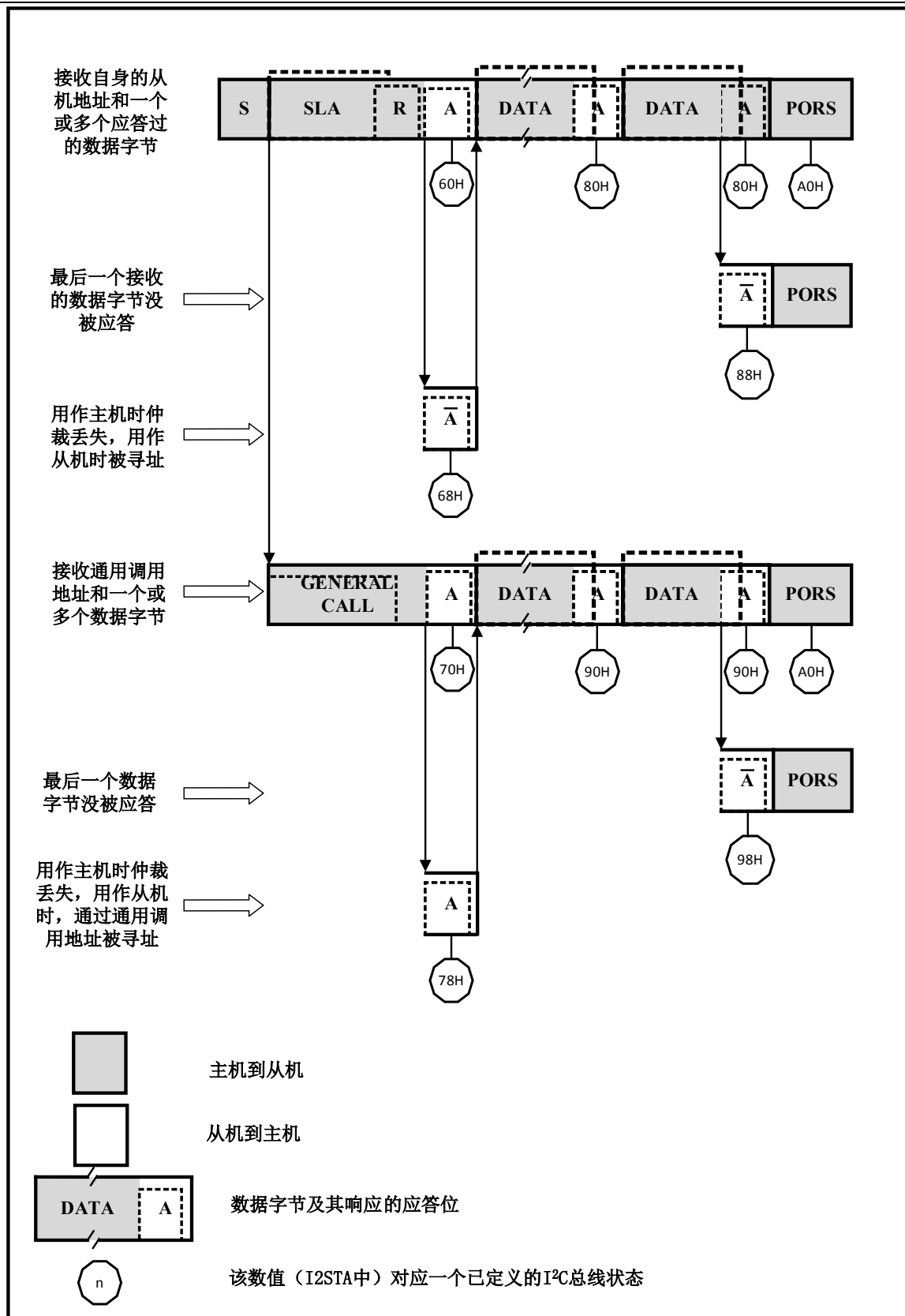


图 24-10. I2C 主机接收状态图

24.4.9 I2C 从机接收模式

在从接收模式中，从机接收的数据字节来自主发送器。高 7 位是主机寻址时 I2C 模块响应的地址。如果 LSB(GC)被置位，I2C 模块将响应通用调用地址(0x00)；否则忽略通用调用地址。I2C 总线速率的设置不影响从机模式中的 I2C 模块。必须置位 I2CEN 来使能 I2C 模块。Aa 位必须置位以使能 I2C 模块来应答其自身从机地址或通用调用地址。sta，sto 和 si 必须复位。当 I2C_ADDR 和 I2C_CR 完成初始化后，I2C 模块一直等待，直至被从机地址寻址，之后是数据方向位寻址，为了工作在从接收模式中，数据方向位必须为“0”(W。)接收完其自身的从机地址和 W 位后，串行中断标志(si)置位，可从 I2C_STAT 中读出一个有效的状态代码。该状态代码用作状态服务程序的向量。每个状态代码的对应操作见下表。如果 I2C 模块在主机模式中仲裁丢失，也可进入从接收模式(请参考状态 0x68 和 0x78 的描述)。

如果 aa 位在传输过程中复位，则在接收完下一个数据字节后 I2C 模块将向 SDA 返回一个非应答(逻辑 1)。当 aa 复位时，I2C 模块不响应其自身的从机地址或通用调用地址。但是，I2C 总线仍被监控，而且，地址识别可随时通过置位 aa 来恢复。这就意味着 aa 位可临时将 I2C 模块从 I2C 总线上分离出来。

表 24.4. I2C 从机接收模式状态表

状态代码	I2C 总线和硬件状态	应用软件响应					I2C 硬件执行的下一个动作
		读/写 I2C_DATA	写 I2C_CR				
			sta	sto	si	aa	
60H	已接收自身的SLA+W；已接收ACK	无 I2C_DATA 动作	X	0	0	0	将接收数据字节，将返回非 ACK
		无 I2C_DATA 动作	X	0	0	1	将接收数据字节，将返回ACK
68H	主控时在SLA+R/W 丢失仲裁；已接收自身的SLA+W；已返回ACK	无 I2C_DATA 动作	X	0	0	0	将接收数据字节，将返回非 ACK
		无 I2C_DATA 动作	X	0	0	1	将接收数据字节，将返回ACK
70H	已接收通用调用地址(0x00)；已返回ACK	无 I2C_DATA 动作	X	0	0	0	将接收数据字节，将返回非 ACK
		无 I2C_DATA 动作	X	0	0	1	将接收数据字节，将返回ACK
78H	主控时在	无 I2C_DATA 动作	X	0	0	0	将接收数据字节，将返回

状态 代码	I2C 总线和硬件状态	应用软件响应					I2C 硬件执行的下一个动作
		读/写 I2C_DATA	写 I2C_CR				
			sta	sto	si	aa	
	SLA+R/W 中丢失仲裁；已接收通用调用地址；已返回 ACK						非 ACK
		无 I2C_DATA 动作	X	0	0	1	将接收数据字节，将返回 ACK
80H	前一次寻址使用自身从地址；已接收数据字节；已返回 ACK	无 I2C_DATA 动作	X	0	0	0	将接收数据字节，将返回非 ACK
		无 I2C_DATA 动作	X	1	0	1	将接收数据字节，将返回 ACK
88H	前一次寻址使用自身从地址；已接收数据字节；已返回非 ACK	读取数据字节	0	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址
		读取数据字节	0	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址
		读取数据字节	1	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址
		读取数据字节	1	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件
90H	前一次寻址使用通用调用地址；已接收数据；已返回 ACK	读取数据字节	1	0	0	X	将接收数据字节，将返回非 ACK
		读取数据字节	0	1	0	X	将接收数据字节，将返回 ACK
98H	前一次寻址使用通用调用地址；已接收数据；已返回非 ACK	读取数据字节	0	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址
		读取数据字节	0	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址
		读取数据字节	1	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件
		读取数据字节	1	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件
A0H	当使用从接收/从发送模式中静态寻址时接收到停止条件或重复起始条件	无 I2C_DATA 动作	0	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址
		无 I2C_DATA 动作	0	0	0	1	切换到不可寻址从模式；

状态 代码	I2C 总线和硬件状态	应用软件响应				I2C 硬件执行的下一个动作	
		读/写 I2C_DATA	写 I2C_CR				
			sta	sto	si		aa
						不识别自身从地址或通用地址	
		无 I2C_DATA 动作	1	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件
		无 I2C_DATA 动作	1	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件

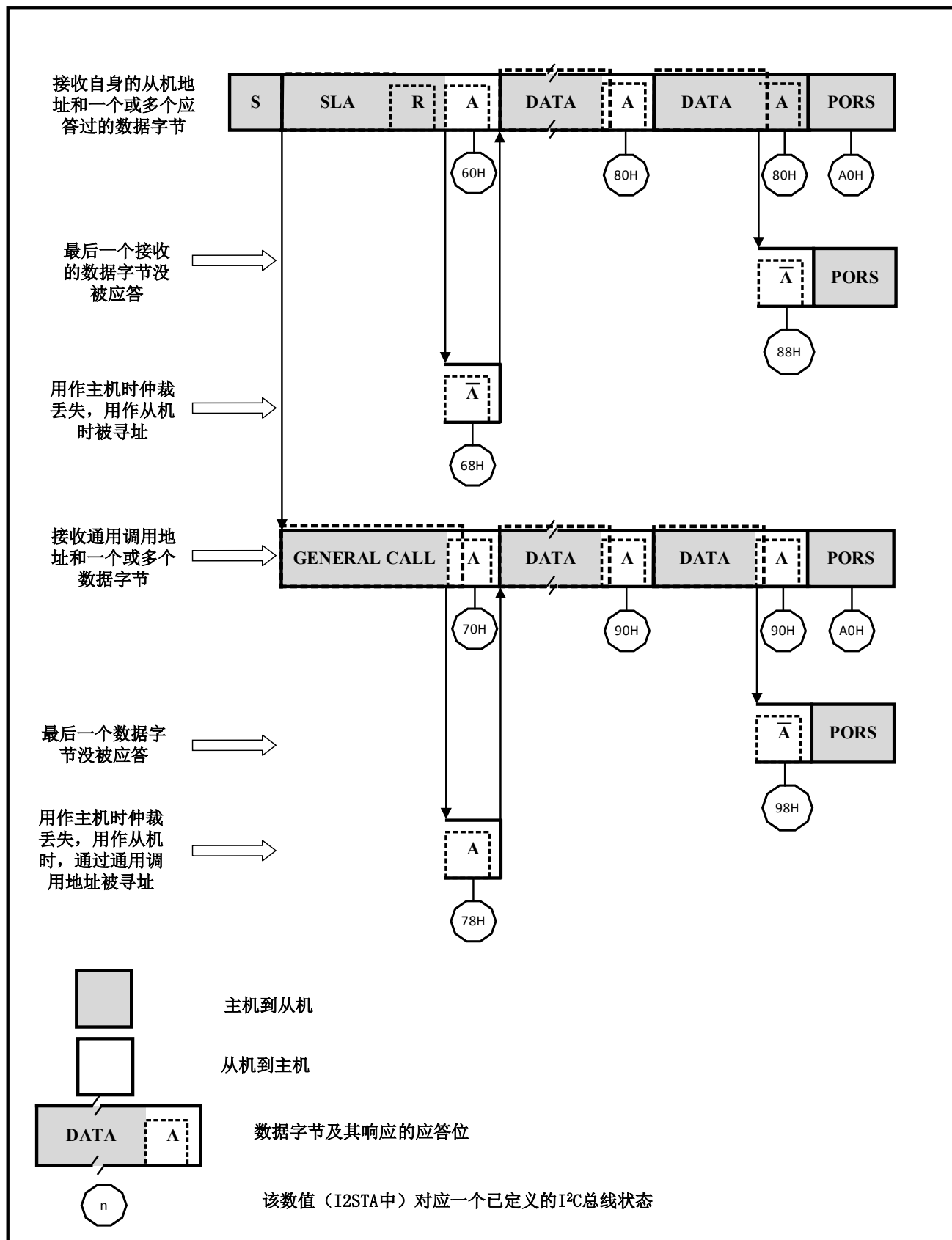


图 24-11. I2C 从机接收状态图

24.4.10 I2C 从机发送模式

在从发送模式中，向主接收器发送数据字节。数据传输按照从接收模式中的情况初始化。当初始化 I2C_ADDR 和 I2C_CR 后，I2C 模块一直等待，直至被自身的从机地址寻址，之后是数据方向位，该数据方向位必须为“1” I，以便 I2C 模块工作在从发送模式下。接收完其自身的从机地址和 R 位后，串行中断标志(si)置位，并且可从 I2C_STAT 中读取一个有效的状态代码。该状态代码用作状态服务程序的向量，每个状态代码的对应操作见下表所示。如果 I2C 模块在主机模式下时仲裁丢失，则可进入从发送模式(见状态 0xB0)。

如果 aa 位在传输过程中复位，则 I2C 模块将发送后一个字节并进入状态 0xC0 或 0xC8。I2C 模块切换到非寻址的从机模式，如果继续传输，它将忽略主接收器。因此主接收器接收所有 1 作为串行数据。当 aa 复位时，I2C 模块不响应其自身的从机地址或通用调用地址。但是，I2C 总线仍被监控，而且，地址识别可随时通过置位 aa 来恢复。这就意味着 aa 位可用来暂时将 I2C 模块从 I2C 总线上分离出来。

表 24.5. 从机发送模式状态表

状态 代码	I2C 总线和硬 件状态	应用软件响应					I2C 硬件执行的下一个动作
		读/写 I2C_DATA	写 I2C_CR				
			sta	sto	si	aa	
A8H	已接收自身的 SLA+R; 已返 回 ACK	装入数据字节	X	0	0	0	将发送后一个数据字节; 将接收 ACK
		装入数据字节	X	0	0	1	将发送一个数据字节; 将接收 ACK
B0H	当主控时在 SLA+R/W 中 丢失仲裁; 已 接 收 自 身 SLA+R; 已返 回 ACK	装入数据字节	X	0	0	0	将发送后一个数据字节; 将接收 ACK
		装入数据字节	X	0	0	1	将发送一个数据字节; 将接收 ACK
B8H	已发送数据; 已接收 ACK	装入数据字节	X	0	0	0	将发送后一个数据字节; 将接收 ACK;
		装入数据字节	X	0	0	1	将发送一个数据字节; 将接收 ACK;
C0H	已发送数据字 节; 已接收非 ACK	无 I2C_DATA 动作	0	0	0	0	切换到不可寻址从模式; 不识别自身 从地址或通用地址;
		无 I2C_DATA 动作	0	0	0	1	切换到不可寻址从模式; 不识别自身 从地址或通用地址;
		无 I2C_DATA	1	0	0	0	切换到不可寻址从模式; 不识别自身

状态 代码	I2C 总线和硬 件状态	应用软件响应				I2C 硬件执行的下一个动作	
		读/写 I2C_DATA	写 I2C_CR				
			sta	sto	si		aa
		动作					从地址或通用地址；当总线空闲后发送起始条件
		无 I2C_DATA 动作	1	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件
C8H	装入的数据字节已被发送；已接收 ACK	无 I2C_DATA 动作	0	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址
		无 I2C_DATA 动作	0	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址
		无 I2C_DATA 动作	1	0	0	0	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件
		无 I2C_DATA 动作	1	0	0	1	切换到不可寻址从模式；不识别自身从地址或通用地址；当总线空闲后发送起始条件

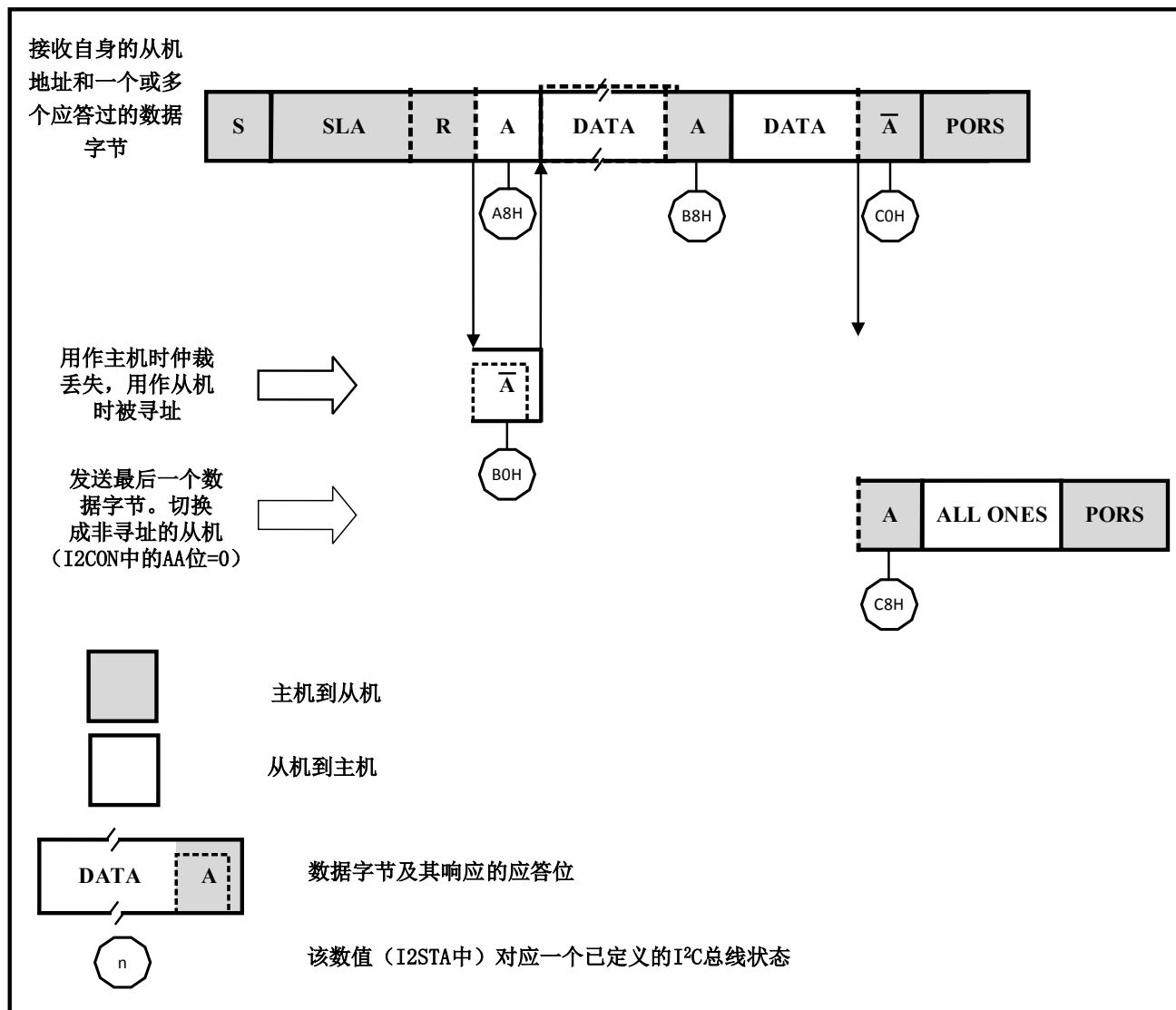


图 24-12. I2C 从机发送状态图

24.4.11 其他杂项状态

I2C_STAT=0xF8:

这个状态码表示没有任何可用的相关信息, 因为串行中断标志 si 还没有置位。这种情况在其它状态和 I2C 模块还未开始执行串行传输之间出现。

I2C_STAT=0x00:

该状态代码表示在 I2C 串行传输过程中出现了总线错误。当格式帧的非法位置上出现了起始或停止条件时总线错误产生。这些非法位置是指在串行传输过程中的地址字节、数据字节或应答位。当外部干扰影响到内部 I2C 模块信号时也会产生总线错误。总线错误出现时 si 置位。

要从总线错误中恢复，sto 标志必须置位，si 必须被清除。这使得 I2C 模块进入“非寻址的”从机模式(已定义的状态)并清除 sto 标志(I2C_CR 中的其它位不受影响)。SDA 和 SCL 线被释放(不发送停止条件)。

表 24.6. 其他杂项状态表

状态 代码	I2C 总线和硬件状态	应用软件响应				I2C 硬件执行的下一个动作	
		读/写 I2C_DATA	写 I2C_CR				
			sta	sto	si		aa
F8H	无可用的相关状态信息； si=0	无 I2C_DATA 动作	无 I2C_DATA 动作				等待或执行当前传输
00H	由于非法的起始或停止条件的出现，在主机或被选中的从机将出现总线错误；当外部干扰使 I2C 进入未定义的状态时也会出 0x00	无 I2C_DATA 动作	0	1	0	X	只有在主机或被寻址的从机模式中，内部硬件受影响。一般情况下，总线被释放，I2C 模块切换到非寻址的从机模式。Sto 复位。

24.5 I2C 操作模式

24.5.1 初始化程序

将 I2C 接口初始化用作从机和/或主机的例子。

- 将自身的从机地址装入 I2C_ADDR，使能通用调用识别(如果需要的话)
- 使能 I2C 中断；
- 向寄存器 I2C_CR 写入 0x44 来置位 ens 和 aa 位，并使能从机功能。对于主机功能，可向寄存器 I2C_CR 写入 0x40。

24.5.2 端口配置程序

I2C 接口信号 SCL，SDA 映射到芯片引脚 PB4，PB5 的例子。

- 配置 PB4，PB5 为开漏输出模式：P2OD[6]，P2OD[5]配置为 0x1
- 配置 PB4，PB5 的功能配置寄存器：PBAFR4，PBAFR5 配置为 0x4
- 配置 PB4，PB5 的上拉使能配置寄存器：PBPUPD4，PBPUPD5 配置为 0x1

24.5.3 启动主机发送功能

通过建立缓冲区、指针和数据计数然后发启起始条件便可执行主发送操作。

- a) 初始化主机数据计数器;
- b) 建立数据将被发送到的从机地址, 并且添加写位;
- c) 向 I2C_CR 写入 0x20 来置位 sta 位;
- d) 在主发送缓冲区内建立要发送的数据;
- e) 初始化主机数据计数器来匹配正在发送的信息长度;
- f) 退出。

24.5.4 启动主机接收功能

通过建立缓冲区、指针和数据计数然后发启起始条件便可执行主接收操作。

- a) 初始化主机数据计数器;
- b) 建立数据将被发送到的从机地址, 并且添加读位;
- c) 向 I2C_CR 写入 0x20 来置位 sta 位;
- d) 在主接收缓冲区内建立要发送的数据;
- e) 初始化主机数据计数器来匹配正在发送的信息长度;
- f) 退出。

24.5.5 I2C 中断程序

确定 I2C 的状态和处理该状态的状态程序。

- a) 从 I2C_STAT 中读出 I2C 的状态;
- b) 使用状态值跳转到 26 个可能状态程序中的一个。

24.5.6 无指定模式状态

- 1) 状态: 0x00 总线错误。进入非寻址的从机模式并释放总线。
 - a) 向 I2C_CR 写入 0x14 来置位 sto 和 aa 位;
 - b) 向 I2C_CR 写入 0xF7 来清除 si 标志;

c) 退出。

2) 主机状态

状态 08 和 10 适用于主发送模式和主接收模式。R/W 位决定了下一个状态是在主发送模式中还是在主接收模式中

3) 状态：0x08

已发送起始条件。即将发送从机地址+R/W 位和接收 ACK 位。

- a) 向 I2C_DATA 写入从机地址和 R/W 位；
- b) 向 I2C_CR 写入 0x04 来置位 aa 位；
- c) 向 I2C_CR 写入 0xF7 来清除 si 标志；
- d) 建立主发送模式数据缓冲区；
- e) 建立主接收模式数据缓冲区；
- f) 初始化主机数据计数器；
- g) 退出。

4) 状态：0x10

已发送重复起始条件。即将发送从机地址+R/W 位和接收 ACK 位。

- a) 向 I2C_DATA 写入从机地址和 R/W 位；
- b) 向 I2C_CR 写入 0x04 来置位 aa 位；
- c) 向 I2C_CR 写入 0xF7 来清除 si 标志；
- d) 建立主发送模式数据缓冲区；
- e) 建立主接收模式数据缓冲区；
- f) 初始化主机数据计数器；
- g) 退出。

24.5.7 主发送状态

1) 状态：0x18

之前状态为 8 或 10 表示已发送从机地址和写操作位，并接收了应答。即将发送第一个数据字节和接收 ACK 位。

- a) 将主发送缓冲区的第一个数据字节装入 I2C_DATA;
 - b) 向 I2C_CR 写入 0x04 来置位 aa 位;
 - c) 向 I2C_CR 写入 0xF7 来清除 si 标志;
 - d) 主发送缓冲区指针加 1;
 - e) 退出。
- 2) 状态：0x20 已发送从机地址和写操作位并接收了非应答。即将发送停止条件。
- a) 向 I2C_CR 写入 0x14 来置位 sto 和 aa 位;
 - b) 向 I2C_CR 写入 0xF7 来清除 si 标志;
 - c) 退出。

3) 状态：0x28

已发送数据并接收了 ACK。如果发送的数据是后一个数据字节则发送一个停止条件，否则发送下一个数据字节。

- a) 主机数据计数器减 1，如果发送的不是后一个数据字节就跳至第 e)步;
 - b) 向 I2C_CR 写入 0x14 来置位 sto 和 aa 位;
 - c) 向 I2C_CR 写入 0xF7 来清除 si 标志;
 - d) 退出;
 - e) 将主发送缓冲区的下一个数据字节装入 I2C_DATA;
 - f) 向 I2C_CR 写入 0x04 来置位 aa 位;
 - g) 向 I2C_CR 写入 0xF7 来清除 si 标志;
 - h) 主机发送缓冲区指针加 1;
 - i) 退出。
- 4) 状态：0x30 已发送数据并接收到非应答。即将发送停止条件;

- a) 向 I2C_CR 写入 0x14 来置位 sto 和 aa 位;
- b) 向 I2C_CR 写入 0xF7 来清除 si 标志;
- c) 退出。

5) 状态: 0x38 仲裁已在发送从机地址和写操作位或数据的过程中丢失。总线已被释放且进入非寻址的从机模式。当总线再次空闲时将发送一个新的起始条件。

- a) 向 I2C_CR 写入 0x24 来置位 sta 和 aa 位;
- b) 向 I2C_CR 写入 0xF7 来清除 si 标志;
- c) 退出。

24.5.8 主接收状态

- 1) 状态: 0x40

前面的状态是 08 或 10 表示已发送从机地址和读操作位, 并接收到 ACK。将接收数据和返回 ACK。

- a) 向 I2C_CR 写入 0x04 来置位 aa 位;
- b) 向 I2C_CR 写入 0xF7 来清除 si 标志;
- c) 退出。

- 2) 状态: 0x48

已发送从机地址和读操作位, 并接收到非应答。将发送停止条件。

- a) 向 I2C_CR 写入 0x14 来置位 sto 和 aa 位;
- b) 向 I2C_CR 写入 0xF7 来清除 si 标志;
- c) 退出。

- 3) 状态: 0x50

已接收到数据, 并返回 ACK。将从 I2C_DATA 读取数据。将接收其它的数据。如果这是一个数据字节, 则返回非应答, 否则返回 ACK。

- a) 读取 I2C_DATA 中的数据字节, 存放到主机接收缓冲区;

- b) 主机数据计数器减 1，如果不是后一个数据字节就跳到第 e)步；
 - c) 向 I2C_CR 写入 0xF3 来清除 si 标志和 aa 位；
 - d) 退出；
 - e) 向 I2C_CR 写入 0x04 来置位 aa 位；
 - f) 向 I2C_CR 写入 0xF7 来清除 si 标志；
 - g) 主机接收缓冲区指针加 1；
 - h) 退出。
- 4) 状态：0x58

已接收到数据，已返回非应答。将从 I2C_DATA 中读取数据和发送停止条件。

- a) 读取 I2C_DATA 中的数据字节，存放到主机接收缓冲区；
- b) I2C_CR 写入 0x14 来置位 sto 和 aa 位；
- c) 向 I2C_CR 写入 0xF7 来清除 si 标志；
- d) 退出。

24.5.9 从接收状态

- 1) 状态：0x60

已接收到自身从机地址和写操作位，已返回 ACK。将接收数据和返回 ACK。

- a) 向 I2C_CR 写入 0x04 来置位 aa 位；
- b) 向 I2C_CR 写入 0xF7 来清除 si 标志；
- c) 建立从接收模式数据缓冲区；
- d) 初始化从机数据计数器；
- e) 退出。

- 2) 状态：0x68

用作总线主机时仲裁已在传输从机地址和 R/W 位时丢失。已接收到自身从机地址和写操作位，并已返回 ACK。将接收数据和返回 ACK。当总线再次空闲后置位 sta 来重启主机模式。

- a) 向 I2C_CR 写入 0x24 来置位 sta 和 aa 位;
- b) 向 I2C_CR 写入 0xF7 来清除 si 标志;
- c) 建立从接收模式数据缓冲区;
- d) 初始化从机数据计数器;
- e) 退出。

3) 状态: 0x70

已接收到通用调用和返回 ACK。将接收数据和返回 ACK。

- a) 向 I2C_CR 写入 0x04 来置位 aa 位;
- b) 向 I2C_CR 写入 0xF7 来清除 si 标志;
- c) 建立从接收模式数据缓冲区;
- d) 初始化从机数据计数器;
- e) 退出。

4) 状态: 0x78

用作总线主机时仲裁已在传输从机地址和 R/W 位时丢失。已接收到通用调用和返回 ACK。

将接收数据和返回 ACK。当总线再次空闲后置位 sta 来重启主机模式。

- a) 向 I2C_CR 写入 0x24 来置位 sta 和 aa 位;
- b) 向 I2C_CR 写入 0xF7 来清除 si 标志;
- c) 建立从接收模式数据缓冲区;
- d) 初始化从机数据计数器;
- e) 退出。

5) 状态: 0x80

之前寻址自身从机地址。已接收到数据并返回 ACK。将读取其它数据。

- a) 读取 I2C_DATA 的数据字节, 存放到从机接收缓冲区。
- b) 从机数据计数器减 1, 如果不是后一个数据字节就跳到第 e)步;

c) 向 I2C_CR 写入 0xF3 来清除 si 标志和 aa 位;

d) 退出;

e) 向 I2C_CR 写入 0x04 来置位 aa 位;

f) 向 I2C_CR 写入 0xF7 来清除 si 标志;

g) 从机接收缓冲区指针加 1;

h) 退出。

6) 状态: 0x88 之前寻址自身从机地址。已接收到数据并返回非应答。不会保存接收到的数据。进入非寻址的从机模式。

a) 向 I2C_CR 写入 0x04 来置位 aa 位;

b) 向 I2C_CR 写入 0xF7 来清除 si 标志;

c) 退出。

7) 状态: 0x90

之前寻址通用调用地址。已接收到数据并返回 ACK。将保存接收到的数据。只接收第一个数据字节并返回 ACK。接收其它数据字节后返回非应答。

a) 读取 I2C_DATA 的数据字节, 并放入从机接收缓冲区;

b) 向 I2C_CR 写入 0xF3 来清除 si 标志和 aa 位;

c) 退出。

8) 状态: 0x98 之前寻址通用调用地址。已接收到数据并返回非应答。不会保存接收到的数据。进入非寻址的从机模式。

a) 向 I2C_CR 写入 0x04 来置位 aa 位;

b) 向 I2C_CR 写入 0xF7 来清除 si 标志;

c) 退出。

9) 状态: 0xA0 已接收停止条件或重复起始条件, 但仍作为从机寻址。不保存接收到的数据。进入非寻址的从机模式。

- a) 向 I2C_CR 写入 0x04 来置位 aa 位;
- b) 向 I2C_CR 写入 0xF7 来清除 si 标志;
- c) 退出。

24.5.10 从发送状态

1) 状态: 0xA8

已接收自身从机地址和读操作位并返回 ACK。将发送数据和接收 ACK 位。

- a) 将从机发送缓冲区的第一个数据字节装入 I2C_DATA;
- b) 向 I2C_CR 写入 0x04 来置位 aa 位;
- c) 向 I2C_CR 写入 0xF7 来清除 si 标志;
- d) 建立从发送模式数据缓冲区;
- e) 从机发送缓冲区指针加 1;
- f) 退出。

2) 状态: 0xB0

用作总线主机时, 在传输从机地址和 R/W 位时丢失仲裁。已接收自身从机地址和读操作位并返回 ACK。将发送数据和接收 ACK 位。当总线再次空闲后置位 sta 来重启主机模式。

- a) 将从机发送缓冲区的第一个数据字节装入 I2C_DATA;
- b) 向 I2C_CR 写入 0x24 来置位 sta 和 aa 位;
- c) 向 I2C_CR 写入 0xF7 来清除 si 标志;
- d) 建立从发送模式数据缓冲区;
- e) 从机发送缓冲区指针加 1;
- f) 退出。

3) 状态: 0xB8

已发送数据并接收到 ACK。将发送数据和接收 ACK 位。

- a) 将从机发送缓冲区的数据字节装入 I2C_DATA;

- b) 向 I2C_CR 写入 0x04 来置位 aa 位;
 - c) 向 I2C_CR 写入 0xF7 来清除 si 标志;
 - d) 从机发送缓冲区指针加 1;
 - e) 退出。
- 4) 状态: 0xC0

已发送数据并接收到非应答。进入非寻址的从机模式。

- a) 向 I2C_CR 写入 0x04 来置位 aa 位;
 - b) 向 I2C_CR 写入 0xF7 来清除 si 标志;
 - c) 退出。
- 5) 状态: 0xC8

已发送后一个数据字节并接收到 ACK。进入非寻址的从机模式。

- a) 向 I2C_CR 写入 0x04 来置位 aa 位;
- b) 向 I2C_CR 写入 0xF7 来清除 si 标志;
- c) 退出。

24.6 I2C 寄存器列表

I2C 基地址: 0x40000C00

偏移地址	名称	寄存器描述	复位值
0x00	I2C_CR	I2C 配置寄存器。	0x00000000
0x04	I2C_DATA	I2C 数据寄存器。	0x00000000
0x08	I2C_ADDR	I2C 地址寄存器。	0x00000000
0x0c	I2C_SR	I2C 状态寄存器。	0x000000F8
0x10	I2C_TIMRUN	I2C 波特率计数器使能寄存器。	0x00000000
0x14	I2C_BAUDCR	I2C 波特率计数器配置寄存器。	0x00000000

24.7 I2C 寄存器说明

24.7.1 I2C 配置寄存器(I2C_CR)

偏移地址：0x00

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									ENS	STA	STO	SI	AA	保留	H1M
									R/W	R/W	R/W	R/W	R/W		R/W

位	标记	功能描述	复位值	读写
31:7	-	保留	0x0	-
6	ENS	I2C 模块使能 0: 禁止 1: 使能	0	R/W
5	STA	开始标志使能 0: 禁止 1: 使能	0	R/W
4	STO	停止标志使能 0: 禁止 1: 使能	0	R/W
3	SI	I2C 中断标志位	0	R/W
2	AA	应答标志使能 0: 禁止 1: 使能	0	R/W
1	-	保留	0	-
0	H1M	I2C 高速 1Mbps 模式使能 0: 禁止 1: 使能	0	R/W

24.7.2 I2C 数据寄存器(I2C_DATA)

偏移地址：0x04

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								I2CDAT[7:0]							
								R/W							

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	I2CDAT[7:0]	I2C 数据寄存器。 在 I2C 发送模式下，写发送数据到这个寄存器。在 I2C 接收模式下读接收数据从这个寄存器。	0x00	R/W

24.7.3 I2C 地址寄存器(I2C_ADDR)

偏移地址：0x08

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								I2CADDR[6:0]						GC	
								R/W						R/W	

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:1	I2CADDR[6:0]	I2C 从机模式地址	0x0	R/W
0	GC	广播地址应答使能 0: 禁止 1: 使能	0	R/W

24.7.4 I2C 状态寄存器(I2C_SR)

偏移地址：0x0C

复位值：0x000000F8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								I2CSTA[7:0]							
								RO							

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	I2CSTA[7:0]	I2C 状态寄存器	0xF8	R/W

24.7.5 I2C 波特率计数器使能寄存器(I2C_TIMRUN)

偏移地址：0x10

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														TME	
														R/W	

位	标记	功能描述	复位值	读写
31:1	-	保留	0x0	-
0	TME	波特率计数器使能寄存器 0：禁止 1：使能	0	R/W

24.7.6 波特率计数器配置寄存器(I2C_BAUDCR)

偏移地址：0x14

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	TM	TM: 波特率计数器配置值 $F_o = F_i / 8 * (N + 1)$ $N = tm, N > 0$	0x0	R/W

25. 串行外设接口(SPI)

25.1 SPI 简介

SPI(Serial Peripheral Interface)总线是一种同步串行外设接口，它可以使 MCU 与各种外围设备以串行的方向进行信息交换。SPI 接口使用 4 条线：串行时钟线(SCLK)主机输出/从机输入线(MOSI)主机输入/从机输出线(MISO)低电平有效从机选择线(SSN)。

25.2 SPI 主要特性

SPI 控制器支持以下特性：

- 通过编程可以配置为主机或者从机
- 全双工通信能力
- 7 种波特率可配置。
- 4 线传输方式
- 主方式大波特率为 1/2 系统时钟
- 从方式大波特率为 1/4 系统时钟
- 可配置的串行时钟极性和相位
- 支持中断方式
- 8 位的数据传输先传输高位后低位

25.3 功能描述

25.3.1 SPI 主机方式

SPI 总线上的所有数据传输都由 SPI 主器件启动，通过将主机/从机控制位 SPI_CR.4 置“1”将 SPI 置于主机方式。当 SPI 处于主机方式时，使能 SPI(将 SPI_CR.6 置“1”)同时向 SPI 数据寄存器 SPI_DATA 写入一个字节时，数据传输开始。SPI 主器件立即在 MOSI 线上串行移出数据，同时在 SCK 上提供串行时钟，在传输结束后 spif 中断标志被置为“1”。如果中断被允许，

将产生一个中断请求。在全双工应用中，当 SPI 主器件向从器件发送数据时，被寻址的 SPI 从器件可在 MISO 线上向主器件发送数据。因此 spif 标志既作为发送完成标志又作为接收数据准备好标志。处理器通过读 SPI_DATA 寄存器得到接收到的数据。

25.3.2 操作流程

1. 端口配置：配置端口控制器，把 SCK, MISO, MOSI 信号映射到正确的引脚，并设定为正确的输入/输出状态；
2. 主机模式下由寄存器 SPI_SSN.0 的值决定片选信号 SPI_CS 的电平；
3. SPI 波特率配置：设置 SPI_CR.SPR2、SPI_CR.SPR1、SPI_CR.SPR0；
4. 串行时钟配置：设置 SPI 时钟极性 SPI_CR.CPOL，时钟相位 SPI_CR.CPHA。详细见 SPI_CR 寄存器；
5. 主机方式配置：SPI_CR.MSTR=1；
6. SPI 使能打开：SPI_CR.SPEN=1；
7. 从机选择：配置 SPI_SSN.SSN=0；
8. 启动发送数据：要发送给从机的数据写到 SPI 数据寄存器 SPI_DATA 中；
9. 等待发送/接收数据完成，准备发送/接收下一个数据；

以下是中断服务程序：

10. 读 SPI_DATA 寄存器，也就是接收从设备发来的数据。

注意：

- 第 10 步完成后 SPIF 中断被清“0”；如果要连续不断的发送/接收数据重复第 8、9 步骤即可。
- 在多机通信中，SSN 引脚可以用 GPIO 代替。

25.3.2.1 时序

使用 SPI 控制寄存器 SPI_CR 中的时钟极性 CPOL 和相位 CPHA，串行时钟可以 4 种组合中选择其一。CPOL(SPI_CR.3)是在 SPI 空闲时，SCK 是处于高电平还是低电平。

CPHA(SPI_CR.2)是选择两种时钟相位(采样数据所用的边沿)一种。主机和从机必须配置为使用相同的时钟相位和极性。波特率的设置只对主机有效，对从机的波特率设置会被忽略掉。主方式数据/时钟时序如下图：

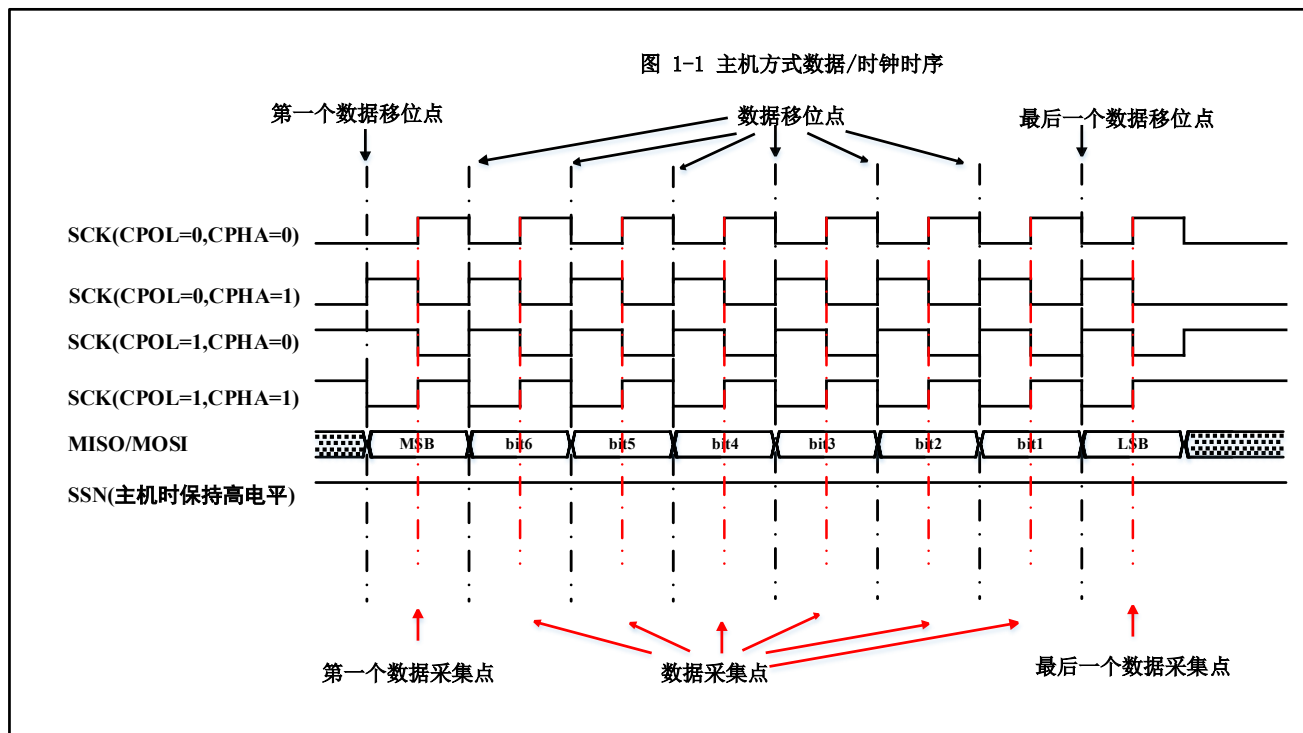


图 25-1. 从机方式数据/时钟时序图

25.3.3 SPI 从机方式

当 SPI 被使能时未被配置为主器件时，它将作为 SPI 从机方式。通过将主机/从机控制位 SPI_CR.4 置“0”将 SPI 置于从机方式。当 SPI 处于从机方式时，由主器件控制串行时钟(SCK)，从 MOSI 移入数据。SPI 逻辑中的计数器对 SCK 边沿计数，当 8 位数据移位完成后，SPIF 标志被置“1”。通过读 SPI_DATA 得到接收到的数据。从器件不能启动数据发送数据功能，通过写 SPI_DATA 来预装要发送给主器件的数据，在主器件 SCK 的作用下，一位一位的移到 MISO 线上发动给主器件。

25.3.3.1 操作流程

1. 端口配置：配置端口控制器，把 SCK，MISO，MOSI 信号映射到正确的引脚，并设定为正确的输入/输出状态；

2. 从机模式下由端口控制寄存器选择一个 GPIO 作为片选信号的来源，参见端口控制寄存器；
3. SPI 波特率配置：设置 SPI_CR.7，SPI_CR.1，SPI_CR.0；
4. 串行时钟配置：设置 SPI 时钟极性 SPI_CR.CPOL，时钟相位 SPI_CR.CPHA。详细见 SPI_CR 寄存器；
5. 串行时钟配置：设置 SPI 时钟极性 CPOL(SPI_CR.3)，时钟相位 CPHA(SPI_CR.2)。详细见 SPI_CR 寄存器；
6. 从机方式配置：MSTR(SPI_CR.4)=0；
7. SPI 使能打开；SPEN(SPI_CR.6)=1；
8. 要发送给主机的数据写到 SPI 数据寄存器 SPI_DATA 中；
9. 等待发送/接收数据完成，准备发送/接收下一个数据以下是中断服务程序；
10. 读 SPI_DATA 寄存器，也就是接收主设备发来的数据。

注意：

- 第 9 步完成后 spif 中断被清 “0”
- 注意当从机时钟相位 CPHA(SPI_CR.2)配置为 0 时，主机每次拉低 SPI_CS 信号，只能向从机传输一个字节的数据。如果主机每次拉低 SPI_CS 信号，要向从机连续传输数据时，传输完成中断发生后需要至少等待 1/2 个 SCK 周期才能进行下一次 SPI_DATA 写操作。
- 注意：如果要连续不断的发送/接收数据重复第 7、8 步骤即可。

从方式数据/时钟时序如下图：

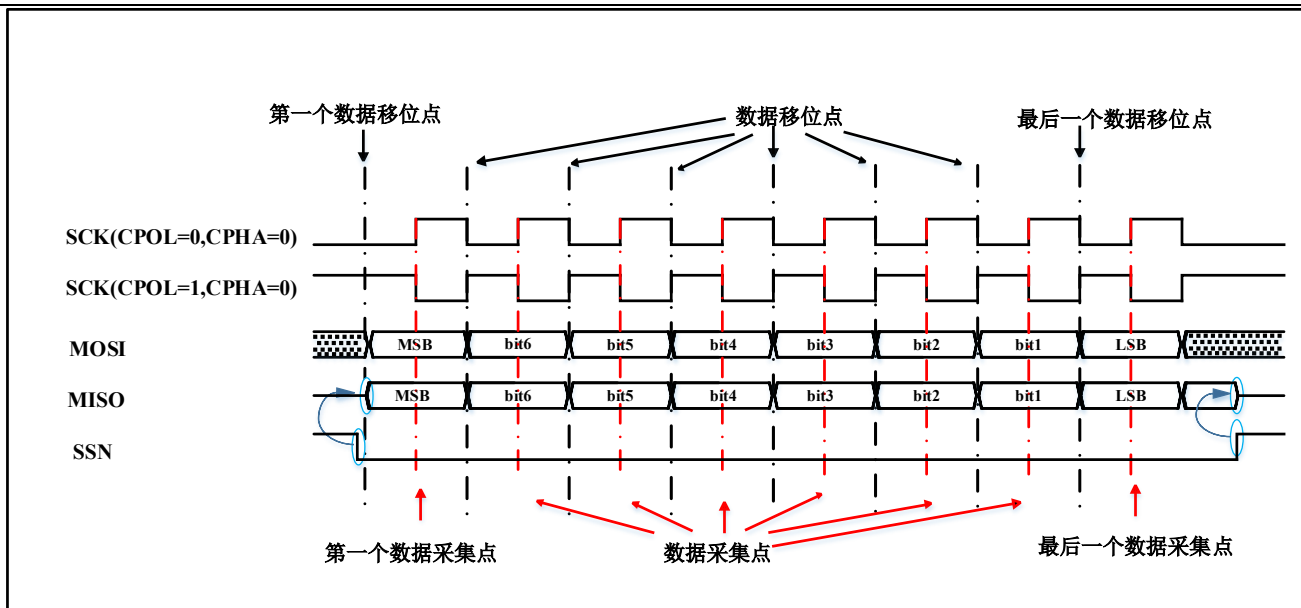


图 25-2. 从机方式数据/时钟时序图(CPHA=0)

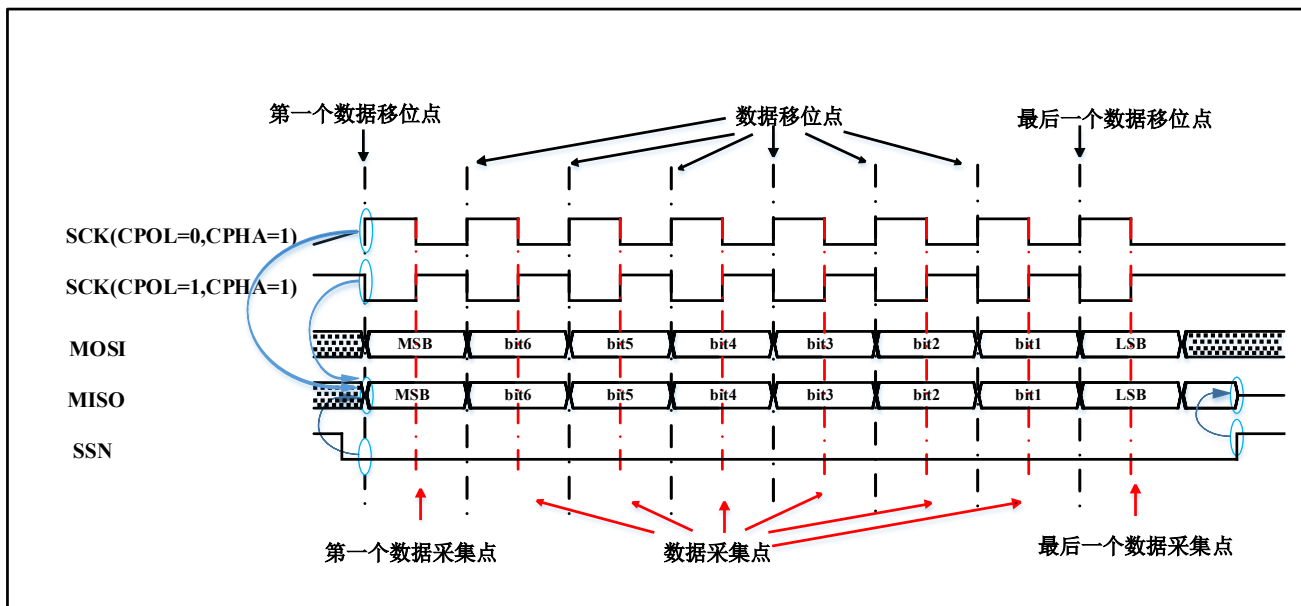


图 25-3. 从机方式数据/时钟时序图(CPHA=1)

25.4 SPI 中断

如果 SPI 中断被允许, SPI 传输完成中断标志位 SPIF(SPI_STAT.7)被置“1”时将产生中断; 或者在 SPI 的主机模式错误中断标志位 MDF(SPI_STAT.4)被置“1”也将产生中断。

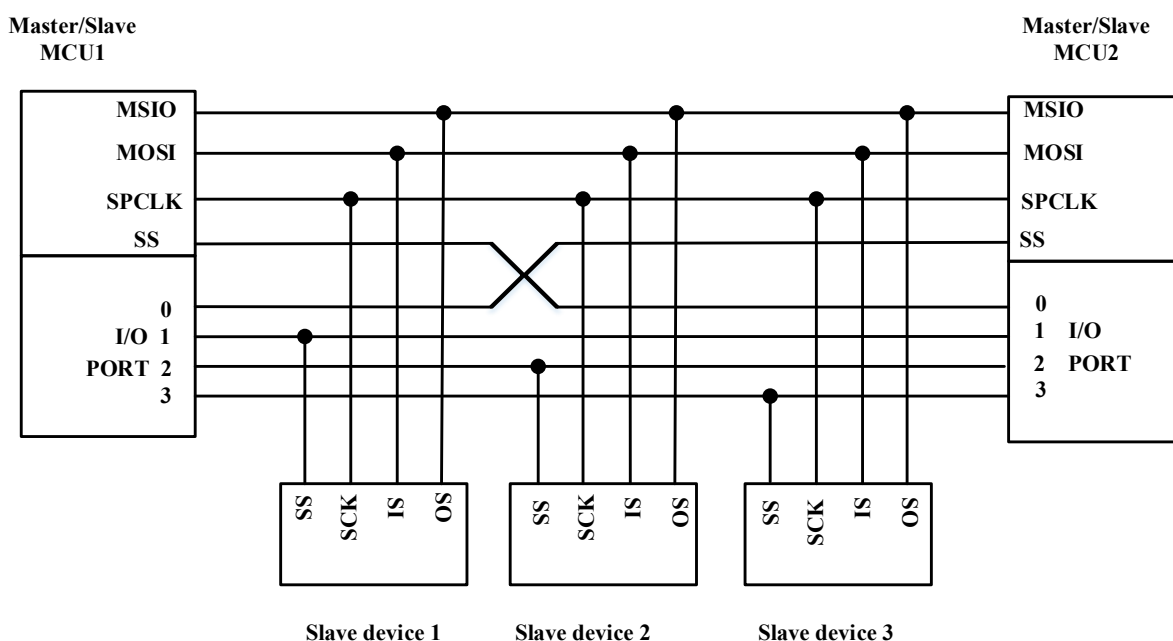
在每次字节传送结束, SPI 传输完成中断标志位 SPIF(SPI_STAT.7)都会被硬件自动置“1”。

当 SPI 被配置为主机方式时, 外部 SSN 输入为低电平, 这时候 SSN 输入电平和 SPI 工作方式相冲突, SPI 的主机模式错误中断标志位 MDF(SPI_STAT.4)被硬件自动置“1”。

25.5 多主机/多从机模式

本产品在 SPI 单主机单从机系统里作为主机时可以配置 SPI 片选配置寄存器，输出高低电平信号到 SPI_CS 引脚。作为从机时，可以配置 8.2.3 端子控制寄存器(SYSCON_PORTCR)寄存器选择一个 GPIO 引脚作为 SPI_SSN 的来源。但是在多主机多从机系统里，如下图所示，必须按照相应流程配置。

当系统为单主机多从机时，可以使用 SPI_CS 引脚作为从机 1 的片选信号，其他从机的片选信号通过 GPIO 引脚连接。当系统为多主机多从机时，所有的从机片选信号都通过 GPIO 引脚连接，主机还必须通过 GPIO 引脚和其他主机的 SPI_CS 信号相连，来监测总线是否被占用。



25.6 SPI 寄存器列表

基地址：0x40000800

表 25.1. 寄存器列表

偏移地址	名称	描述	复位值
0x00	SPI_CR	SPI 配置寄存器.	0x00000014
0x04	SPI_SSN	SPI 片选配置寄存器.	0x00000001
0x08	SPI_SR	SPI 状态寄存器.	0x00000000
0x0c	SPI_DATA	SPI 数据寄存器.	0x00000000

25.7 SPI 寄存器说明

25.7.1 SPI 配置寄存器(SPI_CR)

偏移地址：0x00

复位值：0x00000014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								SPR2	SPEN	保留	MSTR	CPOL	CPHA	SPR1	SPR0
								R/W	R/W		R/W	R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读
31:8	—	保留	0x0	—
7	SPR2	波特率选择位 2, 参考表 25.2.波特率配置表	0	R/W
6	SPEN	SPI 模块使能寄存器 0 —禁止 1 —使能	0	R/W
5	—	保留	0	—
4	MSTR	主机/从机模式选择 0 —从机 1 —主机	1	R/W
3	CPOL	时钟极性选择寄存器 0 —低 1 —高	0	R/W
2	CPHA	时钟相位选择寄存器 0 —第一边沿 1 —第二边沿	1	R/W
1	SPR1	波特率选择位 1, 参考表 25.2.波特率配置表	0	R/W
0	SPR0	波特率选择位 0, 参考表 25.2.波特率配置表	0	R/W

表 25.2. 波特率配置表

SPR2	SPR1	SPR0	SPI_CLKRate
0	0	0	Fsys/2
0	0	1	Fsys/4
0	1	0	Fsys/8

SPR2	SPR1	SPR0	SPI_CLKRate
0	1	1	Fsys/16
1	0	0	Fsys/32
1	0	1	Fsys/64
1	1	0	Fsys/128

25.7.2 SPI 片选配置寄存器(SPI_SSN)

偏移地址：0x04

复位值：0x00000001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														SSN	
														R/W	

位	标记	功能描述	复位值	读
31:1	—	保留	0x0	—
0	SSN	SSN 输出值,在主机模式下,软件配置 SSN 值控制 SPI_CS 端口电平高低	1	R/W

25.7.3 SPI 状态寄存器(SPI_SR)

偏移地址：0x08

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								SPI F	WC OL	SSE RR	MD F	保留			
								RO	RO	RO	RO				

位	标记	功能描述	复位值	读
31:8	—	保留	0x0	—
7	SPIF	传输结束中断标志	0	RO
6	WCOL	写冲突标志	0	RO
5	SSERR	从机模式 SSN 错误标志	0	RO
4	MDF	主机模式错误标志	0	RO
3:0	—	保留	0x0	—

25.7.4 SPI 数据寄存器(SPI_DATA)

偏移地址：0x0c

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								SPI F	WC OL	SSE RR	MD F	保留			
								RO	RO	RO	RO				

位	标记	功能描述	复位值	读
31:8	—	保留位	0x0	—
7:0	SPDATA[7:0]	传输结束中断标志	0x0	R/W

26. One-Wire 接口

26.1 单总线协议(One-Wire)

主机和从机通过 1 根线进行通信，在一条总线上可挂接的从器件数量几乎不受限制。

26.1.1 特点

它采用单根信号线，既可传输时钟，又能传输数据，而且数据传输是双向的。

26.1.2 优点

单总线技术具有线路简单，硬件开销少，成本低廉，便于总线扩展和维护等。

26.2 单总线通信过程

26.2.1 初始化

初始化过程=复位脉冲+从机应答脉冲。

主机通过拉低单总线 480~960us 产生复位脉冲，然后释放总线，进入接收模式。主机释放总线时，会产生低电平跳变为高电平的上升沿，单总线器件检测到上升沿之后，延时 15~60us，单总线器件拉低总线 60~240us 来产生应答脉冲。主机接收到从机的应答脉冲说明单总线器件就绪，初始化过程完成。

初始化时序图如图 26-1 初始化过程中的复位与应答脉冲所示：

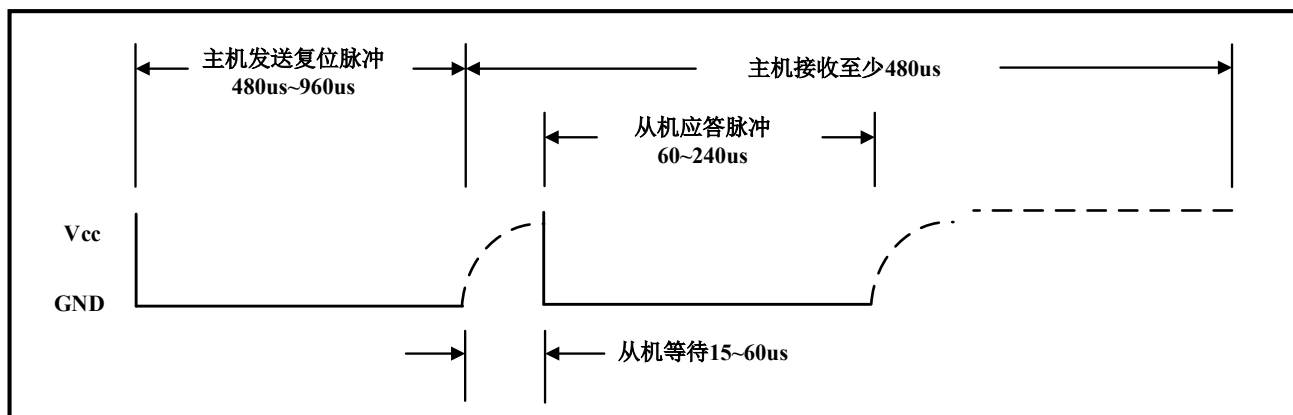


图 26-1. 初始化过程中的复位与应答脉冲

26.2.2 写时间间隙

写时间间隙有两种，包括写 0 的时间间隙和写 1 的时间间隙。

当数据线拉低后，在 15~60us 的时间窗口内对数据线进行采样。如果数据线为低电平，就是写 0，如果数据线为高电平，就是写 1。主机要产生一个写 1 时间间隙，就必须把数据线拉低，在写时间间隙开始后的 15us 内允许数据线拉高。主机要产生一个写 0 时间间隙，就必须把数据线拉低并保持 60us。

写时间间隙时序图如图 26-2 单总线通信协议中写时间间隙时序图所示：

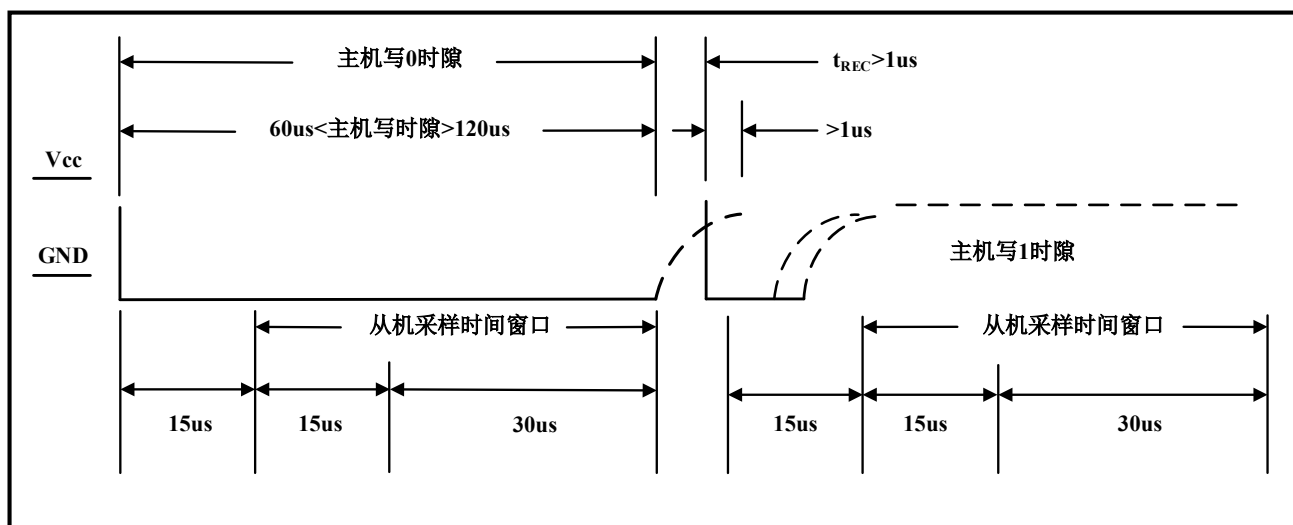


图 26-2.单总线通信协议中写时间间隙时序图

26.2.3 读时间间隙

当主机把总线拉低是，并保持至少 1us 后释放总线，必须在 15us 内读取数据。

读时间间隙时序图如图 26-3 单总线通信协议中读时间间隙时序图所示：

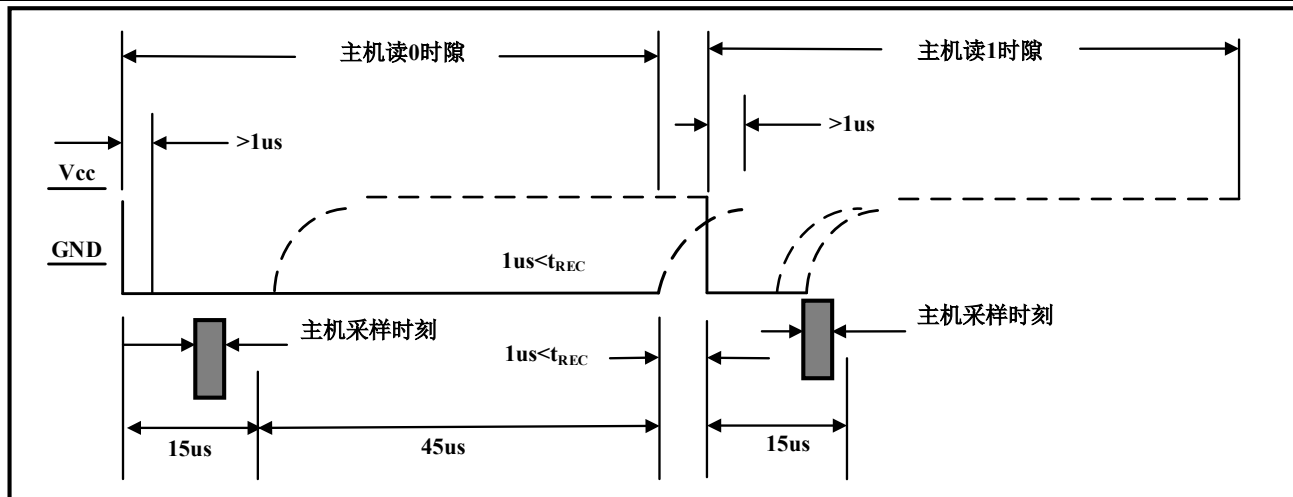


图 26-3. 单总线通信协议中读时间隙时序图

26.3 配置说明

26.3.1 初始化配置说明

- 1) 配置 GPIO 相应的管脚复用 One-Wire 管脚;
- 2) 配置 OWIRE_CR.CLKDIV 寄存器, 设置 One-Wire 模式时钟选择;
- 3) 配置 One-Wire Reset 宽度控制寄存器 OWIRE_RSTCNT, 设置 One-Wire 主发送复位时间(480us~960us);
- 4) 配置 One-Wire 应答宽度计数寄存器 OWIRE_PRESCNT, 设置 One-Wire 从应答设定计数值(60us~240us);
- 5) 配置 One-Wire 中断使能寄存器 OWIRE_INTEN.INITEN 使能初始化完成中断;
- 6) 配置 OWIRE_CR.EN 寄存器, 使能 One-Wire 模块;
- 7) 配置 One-Wire 总线操作命令寄存器 OWIRE_CMD, 设置 Initial 指令;
- 8) 系统进入中断子程序, 在中断子程序中配置状态清除寄存器 OWIRE_INTCLR, 清除相应的中断标志。

26.3.2 读数据配置说明

- 1) 配置 One-Wire Bit rate 计数器 OWIRE_BITRATECNT, 设置 1BIT 数据宽度 (15us~60us);

- 2) 配置 One-Wire 主器件读/写 PULL0 驱动时间 OWIRE_DRVCNT, 设置驱动时间宽度 (0us~15us);
- 3) 配置 One-Wire 主器件读采样时间设定 OWIRE_RDSMPCNT, 设置读采样时间 (1us~15us);
- 4) 配置 1-Wire Recover Time 计数区间值 OWIRE_RECCNT, 设置 RECOVER 时间为 (Trec>1us);
- 5) 配置 One-Wire 中断使能寄存器 OWIRE_INTEN.RXDONEEN 使能接收完成中断;
- 6) 配置 One-Wire 总线操作命令寄存器 OWIRE_CMD, 设置 RX 指令;
- 7) 系统进入中断子程序, 中断子程序中配置状态清除寄存器 OWIRE_INTCLR, 清除接收完成中断 FLAG, 并读 One-Wire 数据寄存器 OWIRE_DATA。

26.3.3 写数据配置说明配置

- 1) 配置 One-Wire Bit rate 计数器 OWIRE_BITRATECNT, 设置 1Bit 数据宽度(15us~60us);
- 2) 配置 One-Wire 主器件读/写 PULL0 驱动时间 OWIRE_DRVCNT, 设置驱动时间宽度 (0us~15us);
- 3) 配置 1-Wire Recover Time 计数区间值 OWIRE_RECCNT, 设置 RECOVER 时间为 (TREC>1us);
- 4) 配置 One-Wire 中断使能寄存器 OWIRE_INTEN.TXDONEEN 使能发送完成中断;
- 5) 写数据到 One-Wire 数据寄存器 OWIRE_DATA;
- 6) 配置 One-Wire 总线操作命令寄存器 OWIRE_CMD, 设置 TX 指令;
- 7) 数据发送完后系统进入中断子程序, 中断子程序中配置状态清除寄存器 OWIRE_INTCLR, 清除 TX 完成中断 FLAG。

26.4 寄存器列表

基地址: 0x40003800

偏移地址	名称	描述	默认值
0x00	OWIRE_CR	1-wire 模块控制寄存器	0x00000000
0x04	OWIRE_NFCR	1-Wire 输入端子滤波控制寄存器	0x00000000
0x08	OWIRE_RSTCNT	1-Wire Master Reset pulse 宽度计数寄存器	0x00000000
0x0c	OWIRE_PRESCNT	1-Wire Device Presence Pulse 宽度计数寄存器	0x00000000
0x10	OWIRE_BITRATECNT	1-Wire Bit rate 设计计数器	0x00000000
0x14	OWIRE_DRVCNT	1-Wire 主器件读/写 PULL0 驱动时间	0x00000000
0x18	OWIRE_RDSMPCNT	1-Wire 主器件读的采样时间设定	0x00000000
0x1C	OWIRE_RECCNT	1-Wire Recover Time 计数区间值	0x00000000
0x20	OWIRE_DATA	1-Wire 数据寄存器	0x00000000
0x24	OWIRE_CMD	1-Wire 总线操作命令寄存器	0x00000000
0x28	OWIRE_INTEN	1-wire 中断使能寄存器	0x00000000
0x2C	OWIRE_SR	1-wire 状态寄存器	0x00000000
0x30	OWIRE_INTCLR	1-wire 中断状态清除寄存器	0x00000000

26.5 寄存器说明

26.5.1 1-Wire 模块控制寄存器(OWIRE_CR)

地址偏移: 0x00

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								RDM ODE	MSB FI RST	EN	SIZE	保留		CLKDIV [1:0]	
								R/W	R/W	R/W	R/W			R/W	

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7	RDMODE	0: 普通模式 1: 写 0/读 0 间隙相等	0	R/W
6	MSBFIRST	字节发送的位模式设定 0: LSB(bit0) send /receive first 1: MSB(bit7) send /receive first 当 OWIRE_CR.SIZE=0 时, 该位要设定为 0;	0	R/W

位	标记	功能描述	复位值	读写
5	EN	wire 模块使能控制位 0: 1-wire 模块停止 1: 1-wire 模块使能	0	R/W
4	SIZE	数据处理位数控制位 0: 单次处理 1bit(Bit mode) 1: 单次处理 8bit(Byte mode)	0	R/W
3:2	-	保留	0x0	—
1:0	CLKDIV[1:0]	计数器用时钟源选择位 00: Fpclk 01: Fpclk/2 10: Fpclk/4 11: Fpclk/16	0	R/W

26.5.2 1-Wire 输入端子滤波控制寄存器(OWIRE_NFCR)

地址偏移: 0x04

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:5	—	保留	0x0	—
4	NFEN	输入端子滤波使能控制位 0: 滤波功能无效 1: 滤波功能有效	0	R/W
3:2	-	保留	0x0	—
1:0	NFDIV[1:0]	输入端子滤波时钟源选择位 输入端子滤波时钟源选择位 00: Fpclk 01: Fpclk/2 10: Fpclk /4 11: Fpclk /8	0x0	R/W

26.5.3 1-Wire RESET 宽度控制寄存器(OWIRE_RSTCNT)

地址偏移：0x08

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:16	—	保留	0x0	—
15:0	RSTCNT[15:0]	主发送复位时间设定计数值	0	R/W

26.5.4 1-Wire Presence Pulse 宽度计数寄存器(OWIRE_PRESCNT)

地址偏移：0x00C

复位值：0x00000000

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

位	标记	功能描述	复位值	读写
31:13	—	保留	0x0	—
12:0	PRESCNT[12:0]	从应答时间设定计数值	0	R/W

26.5.5 1-Wire Bit rate 设计计数器(OWIRE_BITRATECNT)

地址偏移：0x010

复位值：0x0000 0000

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

位	标记	功能描述	复位值	读写
31:12	—	保留	0x0	—
11:0	BITRATECNT[11:0]	Bit Rate 时间设定计数值	0	R/W

26.5.6 1-Wire 主器件读/写 PULL0 驱动时间(OWIRE_DRVCNT)

地址偏移： 0x014

复位值： 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:9	—	保留	0x0	—
8:0	DRVCNT[8:0]	主器件读/写 PULL0 驱动时间设定计数值	0x0	R/W

26.5.7 1-Wire 主器件读采样时间设定(OWIRE_RDSMPCNT)

地址偏移： 0x018

复位值： 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:9	—	保留	0x0	—
8:0	RDSMPCNT[8:0]	主器件读采样时间设定计数值	0x0	R/W

26.5.8 1-Wire Recover Time 计数区间值(OWIRE_REC CNT)

地址偏移： 0x01C

复位值： 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:11	—	保留	0x0	—
10:0	RECCNT[10:0]	Recover Time 计数区间值	0x0	R/W

26.5.9 1-Wire 数据寄存器(OWIRE_DATA)

地址偏移： 0x020

复位值： 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:8	—	保留	0x0	—
7:0	DRVCNT[7:0]	1bit 模式(Bit mode): 只能发送和接收 bit0 8bit 模式(Byte mode): 能发送和接收全部 8 个 bit	0x0	R/W

26.5.10 1-Wire 总线操作命令寄存器(OWIRE_CMD)

地址偏移： 0x024

复位值： 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:2	—	保留	0x0	—
1:0	CMD [1:0]	00: 保留 01: Initial 10: TX 11: RX	0x0	R/W

26.5.11 1-wire 中断使能寄存器(OWIRE_INTEN)

地址偏移： 0x28

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												RXD ONE EN	TXD ONE EN	INIT DON E	ACK ERR EN
												R/W	R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:4	—	保留	0x0	—
3	RXDONEEN	接收完成中断使能 0: 禁止 1: 使能	0	R/W
2	TXDONEEN	发送完成中断使能 0: 禁止 1: 使能	0	R/W
1	INITEN	初始化完成中断使能 0: 禁止 1: 使能	0	R/W
0	ACKERREN	从机应答错误中断使能 0: 禁止 1: 使能	0	R/W

26.5.12 1-wire 状态寄存器(OWIRE_SR)

地址偏移: 0x2c

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												RXDO NE	TXD ONE	INITD ONE	ACK ERR
												RO	RO	RO	RO

位	标记	功能描述	复位值	读写
31:4	—	保留	0x0	—
3	RXDONE	接收完成中断状态 0: 接收完成未发生 1: 接收完成	0	RO

位	标记	功能描述	复位值	读写
2	TXDONE	发送完成中断标志 0: 发送完成未发生 1: 发送完成	0	RO
1	INITDONE	初始化完成标志 0: 初始化完成未发生 1: 初始化完成	0	RO
0	ACKERR	从机应答错误中断标志 0: 从机应答错误未发生 1: 从机应答错误发生	0	RO

26.5.13 1-wire 状态清除寄存器(OWIRE_INTCLR)

地址偏移: 0x2c

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												RXDO NECL R	TXD ONE CLR	INITD ONEC LR	ACK ERR CLR
												WO	WO	WO	WO

位	标记	功能描述	复位值	读写
31:4	—	保留	0x0	—
3	RXDONECLR	接收完成中断清除 写 0: 无作用 写 1: 清除发送完成中断	0	WO
2	TXDONECLR	发送完成中断标志清除 写 0: 无作用 写 1: 清除接收完成中断	0	WO
1	INITDONECLR	初始化完成中断标志清除 写 0: 无作用 写 1: 清除初始化完成中断	0	WO
0	ACKERRCLR	从机应答错误中断标志清除 写 0: 无作用 写 1: 清除从机应答错误中断	0	WO

27. 时钟校准/监测模块(CLKTRIM)

27.1 简介

CLKTRIM (Clock Trimming)模块是一个专门用来校准/监测时钟的电路。在校准模式下选择精准的时钟源来校准不精准的时钟源，反复校准，调节不精准时钟源的参数，直到被校准时钟源的频率达到精度要求。校准模式下计数值会有一定的误差，但是在允许的精度误差范围内。在监测模式下选择稳定的时钟源来监测系统工作时钟，在设定的监测周期下，监测系统工作时钟是否有失效的情况发生并产生中断。在校准模式和监控模式下，所需的时钟源都必须初始化和使能，具体的配置过程请参考第 7 章系统复位与时钟(RCC)。

27.2 主要特性

CLKTRIM 支持以下特性：

- 校准模式
- 监测模式
- 32 位参考时钟计数器可加载初值
- 32 位待校准时钟计数器可配置溢出值
- 6 种参考时钟源
- 4 种待校准时钟源
- 支持中断方式

27.3 CLKTRIM 功能描述

27.3.1 CLKTRIM 校准模式

校准模式主要用于选择一个精准的时钟源作为参考时钟来校准一个不精准的待校准时钟源。由软件按照下面的操作流程反复校准，调节待校准时钟源的参数，直到待校准时钟源满足频率精度要求。

27.3.1.1 操作流程

1. 设置 CLKTRIM_CR.REFCLK_SEL 寄存器选择参考时钟。
2. 设置 CLKTRIM_CR.CALCLK_SEL 寄存器选择被校准时钟。
3. 设置 CLKTRIM_CR.CLKEN 使能校准和参考时钟。
4. 设置 CLKTRIM_REFCON.RCNTVAL 寄存器为校准时间。
5. 设置 CLKTRIM_CR.IE 寄存器使能中断。
6. 设置 CLKTRIM_CR.TRIM_START 寄存器开始校准。
7. 参考时钟计数器和待校准时钟计数器开始计数。
8. 当参考时钟计数器从初始值减计数到 0 时，CLKTRIM_IFR.STOP 置 1，触发中断。
9. 中断服务子程序判断 CLKTRIM_IFR.STOP 为 1，读取寄存器 CLKTRIM_REFCNT 和 CLKTRIM_CALCNT 的值，清零 CLKTRIM_CR.TRIM_START 寄存器结束校准。

注意，校准模式在校准过程中有可能因为校准时间设置过长，发生待校准时钟计数器在 CLKTRIM_IFR.STOP 置 1 之前溢出的情况，CLKTRIM_IFR.CALCNT_OVF 置 1，触发中断。中断服务子程序发现 CLKTRIM_IFR.CALCNT_OVF 置 1 时，清零 CLKTRIM_CR.TRIM_START 寄存器结束校准。这种情况下校准是无法正确进行的，必须调整校准时间，重新校准。具体步骤是：

- 设置 CLKTRIM_REFCON.RCNTVAL 寄存器调整校准时间。
- 设置 CLKTRIM_CR.TRIM_START 寄存器重新开始校准。

27.3.2 CLKTRIM 监测模式

监测模式主要用于选择一个稳定的时钟源作为参考时钟，在设定的时间周期下监测系统工作时钟的异常状态。在监测模式下只能选择外部 HXT 时钟或者外部 LXT 时钟作为被监测时钟。

27.3.2.1 操作流程

1. 设置 CLKTRIM_CR.REFCLK_SEL 寄存器选择参考时钟。

2. 设置 CLKTRIM_CR.CALCLK_SEL 寄存器选择被监控时钟。
3. 设置 CLKTRIM_CR.CLKEN 使能被监控和参考时钟。
4. 设置 CLKTRIM_REFCON.RCNTVAL 寄存器为监控间隔时间。
5. 设置 CLKTRIM_CALCON.CALOV CNT 寄存器为被监控时钟计数器溢出时间。
6. 设置 CLKTRIM_CR.MON_EN 寄存器使能监控功能。
7. 设置 CLKTRIM_CR.IE 寄存器使能中断。
8. 设置 CLKTRIM_CR.TRIM_START 寄存器开始监控。
9. 参考时钟计数器和被监控时钟计数器开始计数。
10. 当参考时钟计数器计数到达监控间隔时间时，判断被监控时钟计数器是否溢出。如果溢出表示被监控时钟工作正常。如果没有溢出表示被监控时钟失效，CLKTRIM_IFR.HXT_FAULT 或 CLKTRIM_IFR.LXT_FAULT 置 1，触发中断。
11. 如果配置 RCC_SYSCCLKCR.CLKFAILEN 为 1(参 7.4.8 系统时钟源配置寄存器 (RCC_SYSCCLKCR))，在中断发生后，会自动切换系统时钟源到内部高速 RC 时钟 (HIRC)，处理中断服务程序，清除中断标志位 CLKTRIM_IFR.HXT_FAULT 或 CLKTRIM_IFR.LXT_FAULT，清零 CLKTRIM_CR.TRIM_START 寄存器结束监测。

27.4 CLKTRIM 寄存器列表

基地址： 0x4000 3400

表 27.1. CLKTRIM 寄存器列表

偏移地址	名称	寄存器描述	复位值
0x00	CLKTRIM_CR	配置寄存器.	0x00000000
0x04	CLKTRIM_REFCON	参考计数器初值配置寄存器.	0x00000000
0x08	CLKTRIM_REFCNT	参考计数器值寄存器.	0x00000000
0x0c	CLKTRIM_CALCNT	校准计数器值寄存器.	0x00000000
0x10	CLKTRIM_IFR	中断标志位寄存器.	0x00000000
0x14	CLKTRIM_ICLR	中断标志位清除寄存器	0x00000000
0x18	CLKTRIM_CALCON	校准计数器溢出值配置寄存器	0xFFFFFFFF

27.5 CLK_TRIM 寄存器说明

27.5.1 配置寄存器(CLKTRIM_CR)

偏移地址：0x00

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								CLK EN	IE	MON _EN	CALCL K_SEL [1:0]	REFCLK_SEL [2:0]		TRIM _STA RT	
								R/W	R/W	R/W	R/W	R/W		R/W	

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
8	CLKEN	参考时钟和校准时钟使能 0: 禁止 1: 时钟使能	0x0	R/W
7	IE	中断使能寄存器 0: 禁止 1: 使能	0x0	R/W
6	MON_EN	监视模式使能寄存器 0: 禁止 1: 使能	0x0	R/W
5:4	CALCLK_SEL[1:0]	待校准/监测时钟选择寄存器 00: HIRC 01: HXT 10: LIRC 11: LXT	0x0	R/W
3:1	REFCLK_SEL[2:0]	参考时钟选择寄存器 000: HIRC 001: HXT 010: LIRC 011: LXT 100: HXT 旁路时钟	0x0	R/W
0	TRIM_START	校准/监测开始寄存器 0: 停止 1: 开始	0x0	R/W

27.5.2 参考计数器处置配置寄存器(CLKTRIM_REFCON)

偏移地址： 0x04

复位值： 0xFFFF FFFF

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

位	标记	功能描述	复位值	读写
31:0	RCNTVAL	参考计数器初始值	0x0	R/W

27.5.3 参考计数器值寄存器(CLKTRIM_REFCNT)

偏移地址： 0x08

复位值： 0xFFFFFFFF

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

位	标记	功能描述	复位值	读写
31:0	REFCNT	参考计数器值 读该寄存器需要先打开时钟使能,当 TRIM_START 有效后, 写入的初始值就会更新到该寄存器	0x0	RO

27.5.4 校准计数器值寄存器(CLKTRIM_CALCNT)

偏移地址： 0x0C

复位值： 0x00000000

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

位	标记	功能描述	复位值	读写
31:0	CALCNT	校准计数器值	0x0	RO

27.5.5 中断标志位寄存器(CLKTRIM_IFR)

偏移地址： 0x10

复位值： 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												HXT _FA _ULT	LXT _FA _ULT	CALC NT_O VF	ST OP
												RO	RO	RO	RO

位	标记	功能描述	复位值	读写
31:4	-	校准计数器值	0x0	-
3	HXT_FAULT	HXT 失效标志 CLKTRIM_ICLR.HXT_FAULT_CLR 写零清除此标志位	0	RO
2	LXT_FAULT	LXT 失效标志 CLKTRIM_ICLR.LXT_FAULT_CLR 写零清除此标志位	0	RO
1	CALCNT_OVF	校准计数器溢出标志. CLKTRIM_CR.START 写零清除此标志位	0	RO
0	STOP	参考计数器停止标志. CLKTRIM_CR.START 写零清除此标志位	0	RO

27.5.6 中断标志位清除寄存器(CLK_RIM_ICLR)

偏移地址： 0x14

复位值： 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												HXT_F AULT_ CLR	LXT_F AULT_ CLR	保留	
												WO	WO		

位	标记	功能描述	复位值	读写
31:4	-	保留	0x0	-
3	HXT_FAULT_CLR	清除 HXT 失效标志，写 1 清除。	0x0	WO
2	LXT_FAULT_CLR	清除 LXT 失效标志，写 1 清除。	0x0	WO
1	-	保留	0x0	-

27.5.7 校准计数器溢出值配置寄存器(CLKTRIM_CALCON)

偏移地址： 0x18

复位值： 0x0000 FFFF

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

位	标记	功能描述	复位值	读写
31:16	-	保留	0x0	-
15:0	CALOVCNT	校准计数器溢出值比较值，在监控模式，在监控周期内如果被监控时钟计数到 CALOVCNT 值时，表示被监控时钟工作正常；如果小于该设定值表示监控时钟停止，会置起被监控时钟的失效位。	0x0	R/W

28. 实时时钟(RTC)

28.1 简介

实时时钟(RTC)是一个独立的 BCD 定时器/计数器，提供秒、分、时(12/24 小时制)、周、日、月和年的信息。

RTC 模块拥有自动唤醒功能，用于管理所有的低功耗模式。

两个 32 位寄存器以 BCD 格式存储秒、分、时(12/24 小时制)、周、日、月和年。

RTC 具有自动月份天数补偿功能，每月的天数和闰年的天数可自动调整。

使用两个 32 位寄存器存储可编程报警信息，包括秒、分、时、周、日、月和年。

对由晶体本身的频偏、温度漂移及其他原因引起的任何误差，可以利用 RTC 本身的数字校准功能进行修正。

上电复位后，所有 RTC 寄存器将被禁止访问，以防止意外的写操作。

当设备处于运行模式、低功耗模式，或复位状态(POR 复位除外)，只要电压在工作范围内，RTC 将保持正常运行。

28.2 主要特性

RTC 模块的主要特性如下：

- 日历功能，可显示秒、分、时(12/24 小时制)、周、日、月和年。
- 可进行自动闰年调整。
- 具有闹钟中断和周期中断功能。
- 数字校准电路(计数器定期修正)：0.95ppm 精准度，来自一个几秒钟的校准窗口。
- 时钟源可选：片外低速晶振(LXT)、片内低速振荡器(LIRC)、片外高速晶振(HXT)。
- 1Hz 方波输出。

28.3 RTC 功能描述

28.3.1 RTC 结构框图

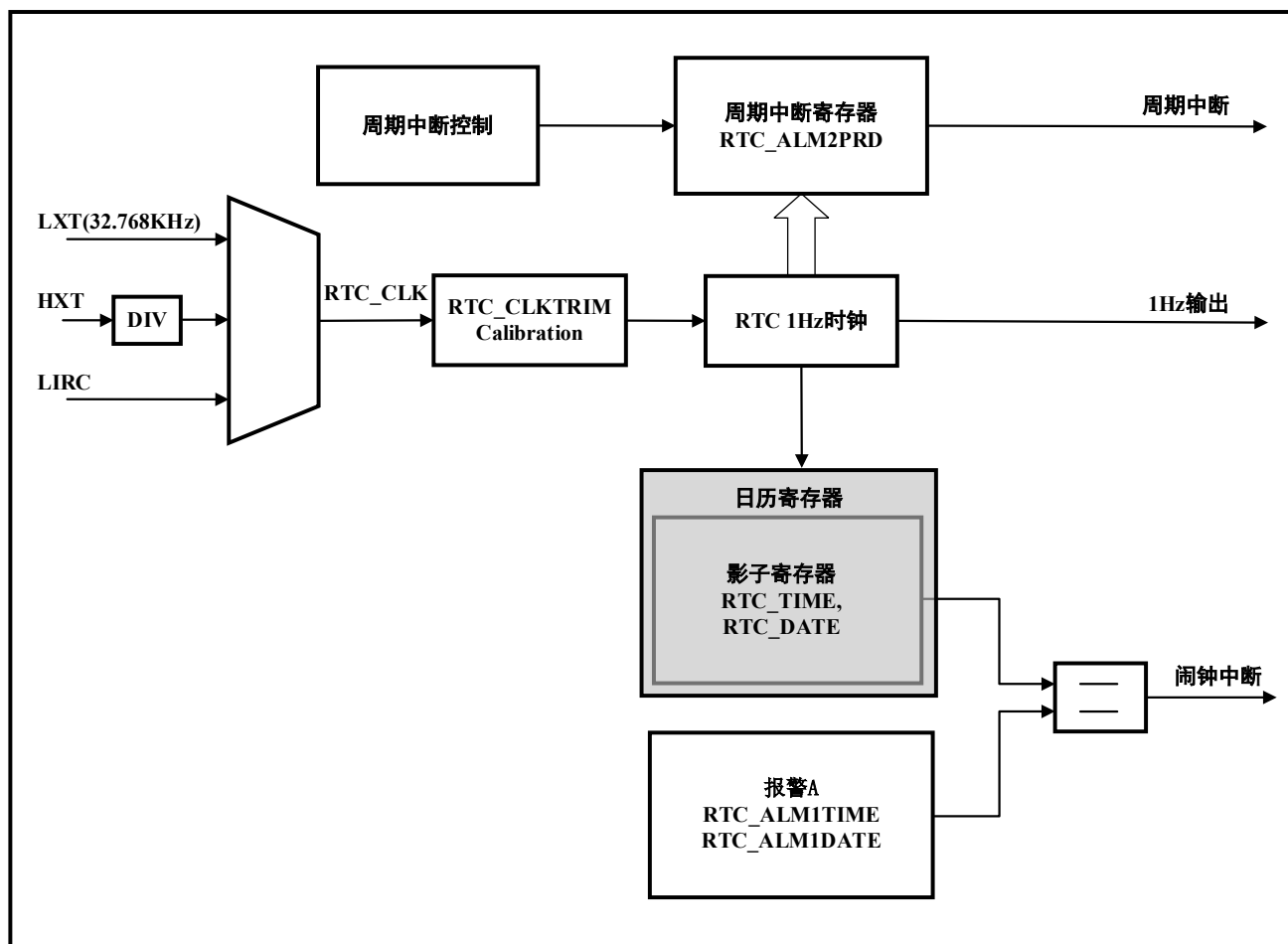


图 28-1. RTC 框图

RTC 模块包括:

一个闹钟报警中断

一个周期中断

校准后的 1Hz 时钟输出,

万年历寄存器

28.3.2 RTC 时钟

由时钟控制器从以下 3 种时钟中选择 RTC 时钟源(RTC_CLK):

LXT 振荡器作为 RTC 时钟;

LIRC 振荡器作为 RTC 时钟；

HXT 振荡器作为 RTC 时钟。

更多有关 RTC 时钟源的配置信息，请参考第 7 章 一系统复位与时钟(RCC)。

28.3.3 复位过程

任何可用的系统复位源都将导致日历影子寄存器和 RTC 状态寄存器(RTC_ISR)复位至默认值。

然而，下列寄存器的复位于系统复位无任何关联，只与上电复位有关：RTC 配置寄存器(RTC_CR)、时钟控制寄存器(RTC_CLKCR)、RTC 校准寄存器(RTC_CLKTR)、ALM 寄存器(RTC_ALM1DATE/RTC_ALM1TIME/RTC_ALM2PRD)、RTC 当前日历寄存器(RTC_TIME/RTC_DATA)。

除上电复位外，发生任何系统复位时 RTC 将维持运行状态。发生上电复位后，RTC 停止运行，所有 RTC 寄存器复位至默认值。

28.3.4 寄存器的写保护

上电复位后，所有 RTC 寄存器将处于写保护状态。通过向写保护寄存器 RTC_WPR 写入指定关键字来启动 RTC 寄存器的写权限。通过以下操作解除所有 RTC 寄存器(RTC_ISR[13:8],RTC_TAFCR 和 RTC_BKPxR 除外)的写保护。

1. 向 RTC_WPR 寄存器写入 ‘0xCA’；
2. 向 RTC_WPR 寄存器写入 ‘0x53’。

注意：保护解除后，任何对该寄存器的再一次写将重新激活写保护。

28.3.5 日历初始化及配置

按照以下顺序完成时间和日期值的初始化，包括时间格式和预分频器的配置：

1. 通过 RTC_CLKCR.CKSEL 选择 RTC 计时时钟源，如果选择 HXT 时钟还要先设定预分频器。
2. 设定 RTC_CLKCR.RTCCKEN 使能 RTC 计时时钟。

3. RTC_ISR 寄存器的 WAIT 位置“1”，进入初始化模式。
4. 等待 RTC_ISR 寄存器的 WAITF 位置“1”，确保已经正式进入初始化模式。由于时钟同步的延迟，该过程大约需要 2 个 RTCCLK 时钟周期。在该模式下日历计数器暂停运行，此时可更新时间和日期计数器的值。
5. 通过 RTC_CR 寄存器的 FMT 位设置时间格式(12 小时制/24 小时制)。
6. 将初始时间和日期值加载到时间寄存器(RTC_TIME 与 RTC_DATE)。
7. 需要进行时钟误差补偿时，设定时钟补偿寄存器 RTC_CLKTRM。
8. 清除 RTC_ISR.WAIT 位的值退出初始化模式。日历计数器的实际值将会自动加载，并在 4 个 RTCCLK 时钟周期后重新启动。

在完成上述一系列初始化操作后，日历将开始计时。

28.3.6 读出计数寄存器

当 RTC_CTRL 寄存器的 BYPSHAD 控制位被清除时：

为确保在安全同步机制下正常读 RTC 日历寄存器(RTC_TIME 和 RTC_DATE)，APB1 时钟频率(f_{PCLK})应至少为 RTC 时钟频率(f_{RTCCLK})的 7 倍以上。

当 APB 时钟频率低于 7 倍 RTC 时钟频率时，软件必须两次读取日历时间和日期寄存器。如果第二次读取的值与第一次读取的值相同，说明返回值是正确的。否则需再次读取。任何情况下，APB 时钟频率都必须大于 RTC 时钟频率。日历寄存器的内容被复制到 RTC_TIME 和 RTC_DATE 影子寄存器中时，RTC_ISR 寄存器的 RSF 位被置位。复制操作每两个 RTCCLK 周期执行一次。为确保三者的值保持一致，读 RTC_TIME，锁定高阶日历影子寄存器中的值，直至 RTC_DATE 中的值被读取。为避免软件在时间间隔少于 2 个 RTCCLK 周期的情况下多次访问日历，每次读日历 RSF 位应由软件清零，软件必须等待 RSF 位被置位后才能读 RTC_TIME 和 RTC_DATE 寄存器。从低功耗模式(待机)唤醒后，RSF 位应由软件清零，软件必须等待 RSF 位被再次置位后才能读 RTC_TIME 和 RTC_DATE 寄存器。

RSF 位应在唤醒后被清除，而不是进入低功耗模式前。

系统复位后，软件必须等待 RSF 位被置位后才能读 RTC_TIME 和 RTC_DATE 寄存器。

事实上系统复位将导致影子寄存器复位至其默认值。

当 RTC_CR 寄存器的 BYPSHAD 控制位被置位时(无需考虑影子寄存器):

读日历寄存器，直接从日历计数器获取值，无需等待 RSF 位被置位。此功能在刚退出低功耗模式(停止或待机模式)时非常有用，因为影子寄存器在低功耗模式下不会自动更新。

在 BYPSHAD 为“1”时，如果两次读寄存器之间出现 RTCCLK 边沿，不同寄存器中的结果可能会互不相关。此外，如果在读操作过程中遇到 RTCCLK 边沿，则某个寄存器的值可能不正确。软件必须读取所有的寄存器两次，并比较两次读取的结果，或者通过比较两组最低有效日历寄存器的结果，以检验数据是否正确且有一定关联。

28.3.7 写入计数寄存器

1. 设定 RTC_ISR.WAIT=1，停止日历寄存器计数，进入写模式；
2. 查询直到 RTC_ISR.WAITF=1；
3. 写入秒，分，时，周，日，月年计数寄存器值；
4. 设定 RTC_ISR.WAIT=0，计数器重新开始。注意，须在 1 秒内完成所有写操作；
5. 查询直到 RTC_ISR.WAITF=0。

28.3.8 闹钟设定

设置 RTC_CR 寄存器的 ALM1EN 位，启动闹钟功能。当日历中秒,分,时,日期,星期,月,年与报警寄存器 RTC_ALM1TIME 和 RTC_ALM1DATE 中设定的值匹配，ALM1_F 由硬件置 1。所有日历字段都可以通过 RTC_ALM1DATE 寄存器中的 ALMxEN 位选择为报警源。设置 RTC_CR 寄存器的 ALM1_INTEN 位，会产生报警中断。

28.3.9 校准 1HZ 输出

RTC 可选择输出校准后的 1Hz 时钟。通过 RTC1HZOE 设定输出使能，通过 RTC_CLKTRM 来设定校准值。

28.3.10 RTC 时钟校准

本产品通过每隔一个固定的时间周期屏蔽指定的 RTC 时钟周期数来补偿 RTC 时钟的频率，通过 RTC_CLKTRM.MODE[1:0]来选择调整的时间间隔：

0b00：每 60 秒(SEC=00)校准一次

0b01：每 30 秒(SEC=00, 30)校准一次

0b10：每 15 秒(SEC=00,15,30,45)校准一次

0b11：每 6 秒(SEC=00,06,12,18,24,30,36,42,48,54)校准一次

通过 RTC_CLKTRM.TRIM[7:0]来指定屏蔽的 RTC 时钟周期数。

注意 RTC_CLKTRM.TRIM[7:0]的值时有符号整数，范围为-128~+127。

28.4 RTC 中断

RTC 支持两种中断类型。闹钟中断、定周期中断。闹钟中断与定周期中断共用一个中断信号，通过标志寄存器位来区分中断源。

28.4.1 RTC 闹钟中断

1. 设定 RTC_CR.ALM1EN=0，禁止闹钟功能；
2. 设定时间闹钟寄存器 RTC_ALM1TIME、日期闹钟寄存器 RTC_ALM1DATE；
3. 设定 RTC_CR.ALM1EN=1，闹钟许可；
4. 清除中断标志位 ALM1_F；
5. 设定 RTC_CR1.INTEN=1，闹钟中断许可，若当前日历时间与闹钟寄存器相等时，触发闹钟中断；
6. 等待发生中断。

28.4.2 RTC 周期中断

控制寄存器 RTC_CTRL 的 ALPR_IRQ_ENA=1 时，选择的周期发生后，触发定周期唤醒中断，由于闹钟和定周期共用中断，通过标志寄存器位来区分。

1. 设定 RTC_CR.ALM2_INTEN=0，禁止周期中断功能；
2. 设定周期闹钟寄存器 RTC_ALM2PRD；
3. 清除中断标志位 ALM2_F；
4. 设定 RTC_CR1.INTEN=1，周期中断许可，选择的周期发生后，触发定周期唤醒中断；
5. 等待发生中断。

28.5 RTC 寄存器列表

基地址： 0x4000 3000

偏移地址	名称	描述	默认值
0x00	RTC_CR	RTC 控制寄存器	0x00000000
0x04	RTC_CLKCR	RTC 时钟控制寄存器	0x00000000
0x08	RTC_TIME	RTC 时间寄存器	0x00000000
0x0C	RTC_DATE	RTC 日期寄存器	0x00000000
0x10	RTC_ALM1TIME	RTC 时间闹钟寄存器	0x00000000
0x14	RTC_ALM1DATE	RTC 日期闹钟寄存器	0x00000000
0x18	RTC_ALM2PRD	RTC 周期闹钟寄存器	0x00000000
0x1C	RTC_CLKTRM	RTC 时钟调校寄存器	0x00000000
0x20	RTC_ISR	初始化和状态寄存器	0x00000000
0x24	RTC_INTCLR	RTC 状态清除寄存器	0x00000000
0x28	RTC_WPR	RTC 写保护寄存器	0x00000000

28.6 RTC 寄存器说明

28.6.1 RTC 控制寄存器(RTC_CR)

偏移地址： 0x00

复位值： 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							STA RT	保留	ALM 1EN	ALM2 _INTE N	ALM1_I NTEN	保留	FMT	RTC1 HZOE	BYPS HAD
							R/W		R/W	R/W	R/W		R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:10	-	保留	0x0	-
8	START	0: 停止 RTC 计数器 1: 使能 RTC 计数器	0	R/W
7	-	保留	0	-
6	ALM1EN	ALM1 闹钟功能使能 0: 禁止 ALM1 闹钟功能 1: 使能 ALM1 闹钟功能 注意：在 RTCCKEN=1 日历计数过程中并且 ALM1_INTEN=1 中断许可的情况下使能 ALM1_INTEN 时，为防止误动作请将系统中断关闭。使能后请将 ALM1_F	0	R/W
5	ALM2_INTEN	ALM2 周期中断使能 0: 禁止 ALM2 周期中断 1: 使能 ALM2 周期中断	0	R/W
4	ALM1_INTEN	ALM1 闹钟中断使能 0: 禁止 ALM1 闹钟中断 1: 使能 ALM1 闹钟中断	0	R/W
3	-	保留	0	-
2	FMT	时间格式 0: AM/PM 时间格式(12 小时制) 1: 24 小时/天格式	0	R/W
1	RTC1HZOE	RTC 1Hz 输出时能 0-禁止 1-使能	0	R/W
0	BYPSHAD	绕过影子寄存器 0: 从影子寄存器读取日历值，每两个 RTCCLK 周期更新一次； 1: 直接从日历计数器读取日历值。 注:如果 APB1 时钟频率低于 RTCCLK 频率的 7 倍，BYPSHAD 必须置“1”。	0	R/W

28.6.2 RTC 时钟控制寄存器(RTC_CLKCR)

偏移地址： 0x04

复位值： 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											RTCC KEN	保留		RTCCKSE L[1:0]	
											R/W			R/W	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						HXTDIV[9:0]									
						R/W									

位	标记	功能描述	复位值	读写
31:21	-	保留	0x0	-
20	RTCCKEN	RTC 计数时钟使能： 由软件置 1 或清 0 写： 0: RTC 时钟关闭 1: RTC 时钟开启 读： 0: RTC 时钟关闭 1: RTC 时钟开启	0	R/W
19:18	-	保留	0x0	-
17:16	RTCCKSEL	RTCCKSEL[1:0]: RTC 时钟源选择 由软件设置来选择 RTC 时钟源。一旦 RTC 时钟源被选定，这些位值不能被改变，除非 RTC 被复位。可通过设置 RTCRST 位来复位 RTC 域。 00: LXT 振荡器作为 RTC 时钟 01: LIRC 振荡器作为 RTC 时钟 10: Fhxt/(HXTDIV+1) 11: 保留	0x0	R/W
15:10	-	保留	0x0	-
9:0	HXTDIV[9:0]	外部高速晶振时钟分频 0: 停止 $F = F_{hxt} / (HXTDIV[9:0] + 1)$	0x0	R/W

28.6.3 RTC 时间寄存器(RTC_TIME)

偏移地址： 0x08

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					WEEK[2:0]			保留		H20 PA	HOUR19[4:0]				
					R/W					R/W	R/W				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	MIN[6:0]							保留	SEC[6:0]						
	R/W								R/W						

位	标记	功能描述	复位值	读写
31:27	-	保留	0x0	-
26:24	WEEK[2:0]	星期计数器。星期计数器值为二进制计数,计数区间从 0 到 6。(7 不被使用,除非不使用星期计数)星期和星期计数器值得对应关系由用户定义。(比如星期日=0, 星期一=1.....星期六=6)	0x0	R/W
23:22	-	保留	0x0	-
21	H20_PA	这两位表示小时计数器。HOUR19 的值为 BCD 编码。时间格式是由时钟系统决定的。	0	R/W
20:16	HOUR19[4:0]	12 小时制模式, H20_PA 指上午或者下午。24 小时制模式, H20_PA 决定了计数器的十位是否为 2。 12 小时制模式,当[H20_PA,HOUR19]从[1,11](11PM)数到[0,12](12AM)的时候,日期计数器增加一天。24 小时制模式,当[H20_PA, HOUR19]从[1,3](23H)数到[0,0](0H)的时候,日期计数器增加一天。	0x0	R/W
15	-	保留	0x0	-
14:8	MIN[6:0]	分钟计数器。分钟计数器的值为 BCD 编码,计数区间从 0 到 59。当分钟计数器从 59 数到 0 的时候,小时计数器增长 1。当这个计数器被写入时,小于一秒的时间将被忽略掉。	0x0	R/W
7	-	保留	0	-
6:0	SEC[6:0]	秒计数器。秒计数器的值为 BCD 编码,计数区间从 0 到 59。当秒计数器从 59 数到 0 的时候,分钟计数器增长 1。当这个计数器被写入时,小于一秒的时间将被忽略掉。	0x0	R/W

28.6.4 RTC 日期寄存器(RTC_DATE)

偏移地址: 0x0C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								YEAR[7:0]							
								R/W							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CEN	保留		MONTH[4:0]					保留		DAY[5:0]					
R/W			R/W							R/W					

位	标记	功能描述	复位值	读写
31:24	-	保留	0x0	-
23:16	YEAR[7:0]	年计数器。 年计数器代表了十进制年的十位和个位。年计数器为 BCD 编码，计数区间从 00 到 99。当年计数器从 99 数到 00 时，世纪计数器增长 1。 当世纪计数器为 0 时，04, 08, ... 92, 96 为闰年。 当世纪计数器为 1 时，00, 04, 08, ... 92, 96 为闰年。	0x0	R/W
15	CEN	世纪计数器。0 代表 20 世纪，1 代表 21 世纪	0	R/W
14:13	-	保留	0x0	-
12:8	MONTH[4:0]	月计数器。月计数器为 BCD 编码，计数区间从 01 到 12。当年计数器从 12 数到 01 时，年计数器增长 1。	0x0	R/W
7:6	-	保留	0x0	-
5:0	DAY[5:0]	天计数器。 天计数器为 BCD 编码，计数区间如下： 1 到 31：一月，三月，五月，七月，八月，十月，十二月； 1 到 30：四月，六月，九月，十一月； 1 到 29：闰年的二月； 1 到 28：非闰年的二月。 当年计数器从 99 数到 00 时，世纪计数器增长 1。	0x0	R/W

28.6.5 RTC 时间闹钟寄存器(RTC_ALM1TIME)

偏移地址：0x10

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					ALWEEK[2:0]			保留	ALH 20_P A	ALHOUR19[4:0]					
					R/W				R/W	R/W					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	ALMIN[6:0]							保留	ALSEC[6:0]						
	R/W								R/W						

位	标记	功能描述	复位值	读写
31:27	-	保留	0x0	-
26:24	ALWEEK[2:0]	闹钟星期设定	0x0	R/W
23:22	-	保留	0x0	-
21	ALH20_PA	闹钟小时设定。参考小时计数寄存器。	0x0	R/W
20:16	ALHOUR19[4:0]			
15	-	保留	0	-
14:8	ALMIN[6:0]	闹钟分钟设定	0x0	R/W
7	-	保留	0	-
6:0	ALSEC[6:0]	闹钟秒设定	0x0	R/W

28.6.6 RTC 日期闹钟寄存器(RTC_ALM1DATE)

偏移地址：0x14

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	ALM YEA REN	ALM MON EN	ALM DAY EN	ALM WEE KEN	ALM HOU REN	ALM MINE N	ALM SEC EN	ALYEAR[7:0]							
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALC EN	保留		ALMONTH[4:0]				保留		ALDAY[5:0]						
R/W			R/W						R/W						

位	标记	功能描述	复位值	读写
31	-	保留	0	-
30	ALMYEAREN	闹钟年设定使能	0	R/W
29	ALMMONEN	闹钟月设定使能	0	R/W

位	标记	功能描述	复位值	读写
28	ALMDAYEN	闹钟日设定使能	0	R/W
27	ALMWEEKEN	闹钟星期设定使能	0	R/W
26	ALMHOUREN	闹钟小时设定使能	0	R/W
25	ALMMINEN	闹钟分钟设定使能	0	R/W
24	ALMSECEN	闹钟秒设定使能	0	R/W
23:16	ALYEAR[7:0]	闹钟年设定	0x0	R/W
15	ALCEN	闹钟世纪设定	0	R/W
14:13	-	保留	0x0	-
12:8	ALMONTH[4:0]	闹钟月设定	0x0	R/W
7:6	-	保留	0x0	-
5:0	ALDAY[5:0]	闹钟日设定	0x0	R/W

28.6.7 RTC 周期闹钟寄存器(RTC_ALM2PRD)

偏移地址： 0x18

复位值： 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													ALM2PR_CNT [3:0]		
													R/W		

位	标记	功能描述	复位值	读写
31:4	-	保留	0x0	-
3:0	ALM2PR_CNT	周期闹钟 2 计数周期设定。 0x0:关闭周期闹钟 2 0x1:1 秒 0x2:1/2 秒 0x3:1/4 秒 0x4:1/8 秒 0x5:1/16 秒 0x6:1/32 秒 0x7:1/64 秒 0x8:1/128 秒	0x0	R/W

位	标记	功能描述	复位值	读写
		0x9:10 秒 0xA:30 秒 0xB:1 分钟 0xC:30 分钟 0xD:60 分钟 0xE:12 小时 0xF:24 小时		

28.6.8 RTC 时钟调校寄存器(RTC_CLKTRIM)

偏移地址: 0x1C

复位值: 0x00000000

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

位	标记	功能描述	复位值	读写
31:10	-	保留	0x0	-
9:8	MODE[1:0]	时钟调节寄存器。决定了时钟调节的频率。 0x0: 每 60 秒(SEC=00) 0x1: 每 30 秒(SEC=00, 30) 0x2: 每 15 秒(SEC=00,15,30,45) 0x3: 每 6 秒(SEC=00,06,12,18,24,30,36,42,48,54)	0x0	R/W
7:0	TRIM[7:0]	时钟补偿时间寄存器。此寄存器为有符号整数。 (-128~+127)	0x0	R/W

28.6.9 RTC 初始化和状态寄存器(RTC_ISR)

偏移地址: 0x20

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										ALM2_F	ALM1_F	保留	RSF	WAIT_F	WAIT
										RO	RO		R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:6	-	保留	0x0	-
5	ALM2_F	周期闹钟 2 中断原始状态寄存器。 当此寄存器被读出时，状态值被返回： 0：周期闹钟 2 中断没有被激活； 1：周期闹钟 2 中断被激活。	0	RO
4	ALM1_F	闹钟中断原始状态寄存器。 当此寄存器被读出时，状态值被返回： 0：闹钟中断没有被激活； 1：闹钟中断被激活。	0	RO
3	-	保留	0	-
2	RSF	寄存器同步标志 每当日历寄存器中的内容复制到影子寄存器 (RTC_SSRx, RTC_TRx and RTC_DRx) 中时，该位由硬件置位。当处于忽略影子寄存器模式 (BYP SHAD=1) 下时，该位由硬件在初始化模式下清除。该位也可由软件清除。 在初始化模式下，该位可由硬件/软件清除。 0：日历影子寄存器尚未同步； 1：日历影子寄存器已经同步。 注意不能软件写 1	0	R/W
1	WAIT_F	0：非写入/读出状态； 1：写入/读出状态。 注意：WAIT 位设定是否有效标志。在写入/读出前请确认该位是否为“1”。计数过程中，在 WAIT 位清“0”后等待写入完成后该位才清“0”。	0	R/W
0:	WAIT	0：正常计数模式 1：写入/读出模式 注意：在写入/读出时请将该位置“1”，由于计数器在连续计数，请在 1 秒的时间内完成写入/读出操作并将该位清“0”。	0	R/W

28.6.10 RTC 状态清除寄存器(RTC_INTCLR)

偏移地址： 0x24

复位值： 0x00000000

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

位	标记	功能描述	复位值	读写
31:8	-	保留	0x0	-
7:0	WPR	写入指定关键字来启动 RTC 寄存器的写权限。 1. 向 RTC_WPR 寄存器写入‘0xCA’; 2. 向 RTC_WPR 寄存器写入‘0x53’。 注意: 保护解除后, 任何对该寄存器的再一次写将重新激活写保护。	0x0	WO

29. 模拟/数字转换器(ADC)

29.1 模块简介

本芯片内部集成了一个 12 位高精度、高转换速率的逐次逼近型模数转换器(SARADC)模块。具有以下特性：

- 12 位转换精度；
- 1M SPS 转换速度；
- 8 路转换通道：7 个引脚通道、1 个电源电压；
- 参考电压固定为电源电压；
- ADC 的电压输入范围：0~Vref；
- 3 种转换模式：单次转换、连续转换、累加转换；
- 软件可配置 ADC 的转换速率；
- 支持片内外设自动触发 ADC 转换。有效降低芯片工号并提高转换的实时性。

29.2 ADC 框图

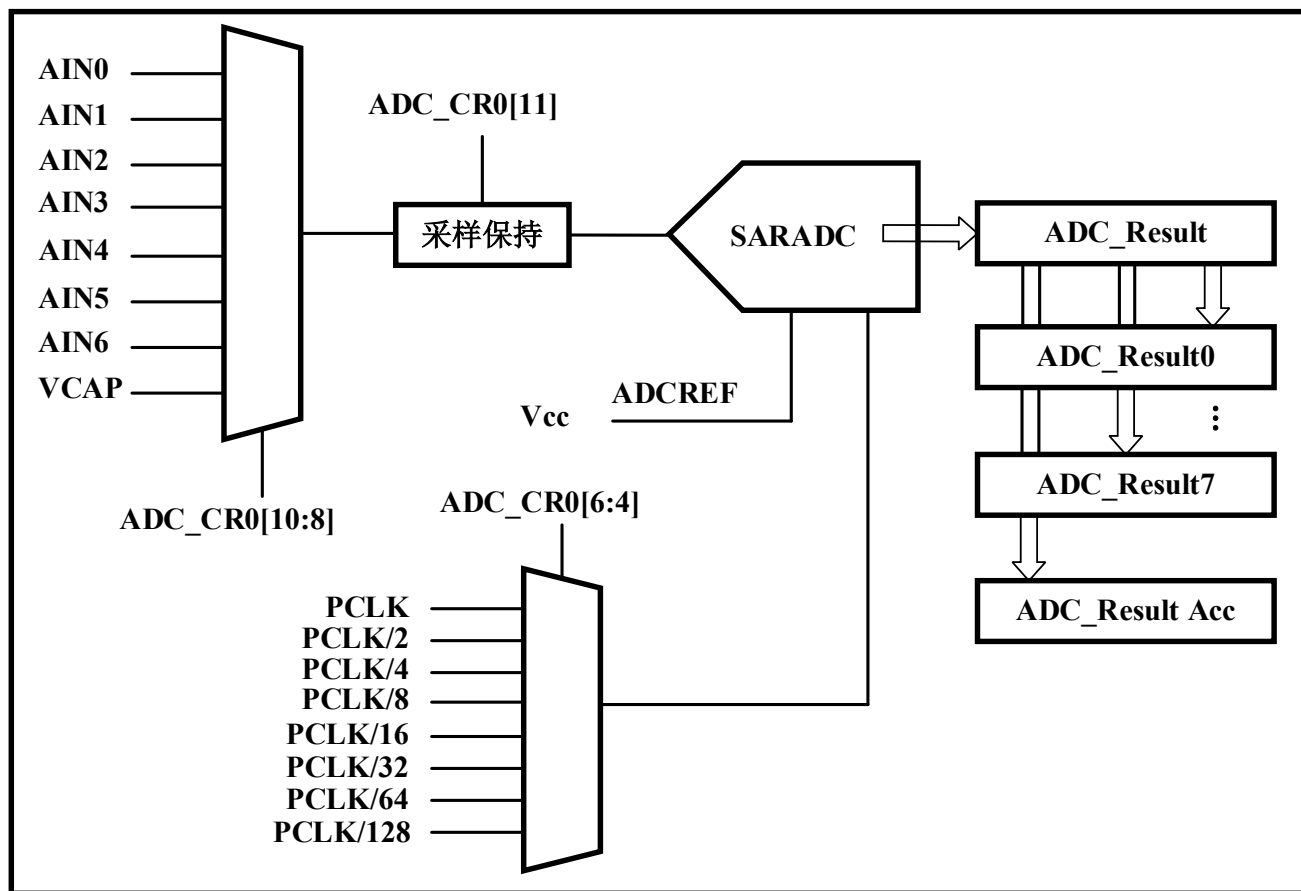


图 29-1. ADC 示意框图

29.3 转换时序及速度

ADC 的转换时序如下图所示：一次完整的 ADC 转换由采样过程及逐次比较过程组成。其中采样过程需要 4~8 个 ADC 时钟，由 ADC_CR0.SAM 配置；逐次比较过程需要 12 个 ADC 时钟。所以，一次 ADC 转换共需要 16~20 个 ADC 时钟。

ADC 转换速度的单位为 SPS，即每秒进行多少次 ADC 转换。ADC 转换速度的计算方法为：ADC 时钟的频率/一次 ADC 转换所需要的 ADC 时钟的个数。

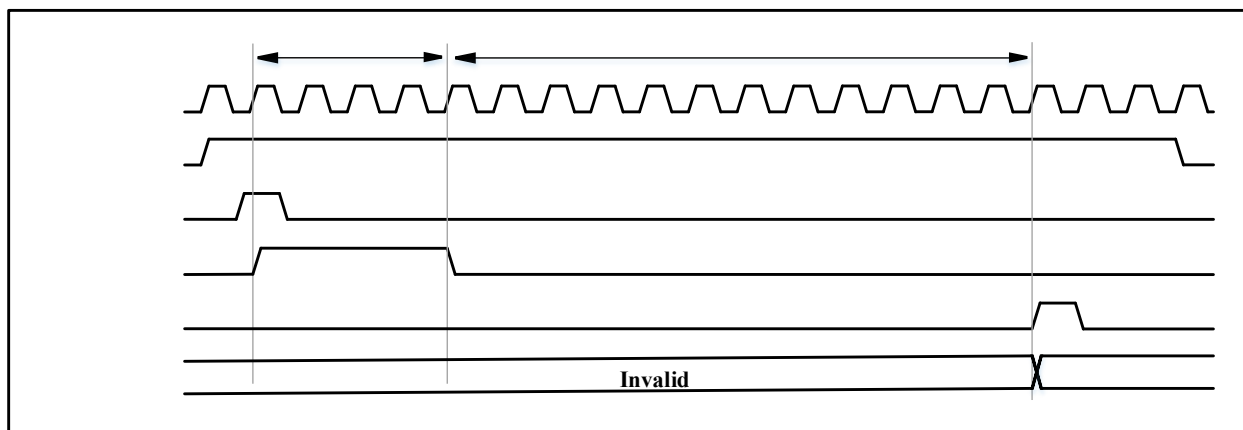


图 29-2. ADC 转换时序图

29.4 单次转换模式

在单次转换模式下，ADC 启动后只执行一次转换，可对所有的 8 路 ADC 通道进行转换。该模式既可通过设置 ADC_CR0.START 位启动也可通过设置 ADC_CR1[9:0]的外部触发启动。一旦选定通道的 ADC 转换完成，ADC_CR0.START 位自动清零，转换结果保存在 ADC_result 寄存器中。

通过 START 位启动 ADC 单次转换操作流程：

1. 根据 pin 配置对应的 GPIO，将待转换的 ADC 通道配置为模拟端口。
2. 配置 ADC_CR2.circle_mode 为 0，选择非循环模式。
3. 配置 ADC_CR1.ct 为 0，选择单次转换模式。
4. 配置 ADC_CR0.sam 及 ADC_CR0.clksel，设置 ADC 的转换速度。
5. 配置 ADC_CR0.sel，选择待转换的通道(注意需要和步骤 1 保持一致)。
6. 配置 ADC_CR0.adcen 为 1，使能 ADC 模块。
7. 配置 ADC_CR0.start 为 1，启动 ADC 单次转换。
8. 等待 ADC_CR0.start 为 0，读取 ADC_result 寄存器以获取 ADC 转换结果。
9. 如需对其他通道进行转换，重复执行步骤 4~7。
10. 配置 ADC_CR0.adcen，关闭 ADC 模块。

注：内部信号触发 ADC 转换的流程配置类似，额外需增加 trigger 的选择。

29.5 连续转换模式

在连续转换模式下，启动一次 ADC 可对多个通道依次进行多次转换；可转换的 ADC 通道为 AIN0~AIN7。ADC 转换的总次数由 ADCCR2.ADCCNT 进配置；待进行转换的通道由 ADC_CR2[7:0] 进行配置。该模式既可通过设置 ADC_CR0.START 位启动也可通过设置 ADC_CR2[9:0] 的外部触发启动。启动连续转换后，ADC 模块依次转换 AIN0~AIN7 中待转换的通道直到总转换次数完成。ADC 模块完成总转换次数后，ADC_RAWINTSR.CONT_INTF 位会自动置 1，转换结果保存在转换通道所对应的 ADC_result0~ADC_result7 寄存器中。如果总转换次数大于待转换的 ADC 通道的数量，则 ADC_result0~ADC_result7 寄存器中只保存后一次的转换结果。

下图演示了对 AIN0、AIN1、AIN5 进行 10 次连续转换的过程。通过寄存器将 START 置 1 后，ADC 内部的状态机会依次对 AIN0、AIN1、AIN5 进行转换，直到 ADCCNT 的计数值变为 0。

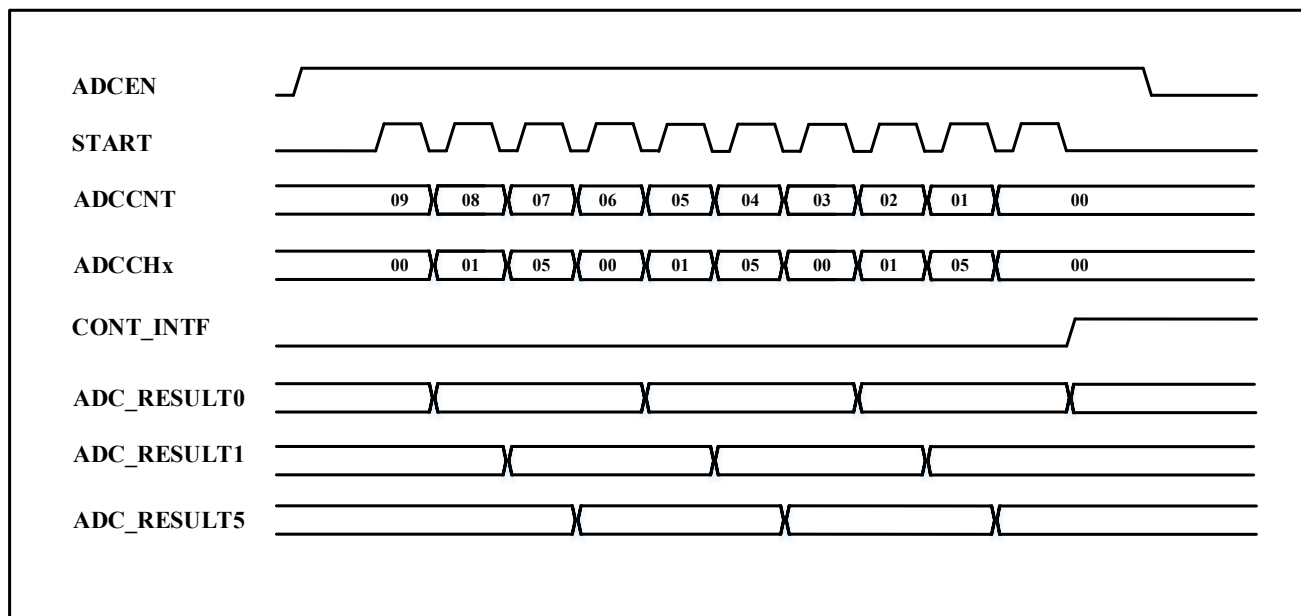


图 29-3. ADC 连续转换过程示例

图 29-3.ADC 连续转换过程示例配置步骤：

1. 根据 pin 配置对应的 GPIO，将待转换的 ADC 通道配置为模拟端口。

2. 配置 ADC_CR2.circle_mode 为 0，选择非循环模式。
3. 配置 ADC_CR1.ct 为 1，选择连续转换模式。
4. 配置 ADC_CR1.adcnt，选择连续转换的总转换次数。
5. 配置 ADC_CR0.sam 及 ADC_CR0.clksel，设置 ADC 的转换速度。
6. 配置 ADC_CR2.chen[7:0]，选择待转换的通道。
7. 配置 ADC_INTCLR.cont_intc 为 0，清除 ADC_RAWINTSR.cont_intf 标志。
8. 配置 ADC_MSKINTSR.cont_imsr 为 1，打开掩码中断 9.配置 ADC_CR0.staterst 为 1，复位连续转换状态。
10. 配置 ADC_CR0.adcen 为 1，使能 ADC 模块。
11. 配置 ADC_CR0.start 为 1，启动 ADC 连续转换。
12. 等待 ADC_RAWINTSR.cont_intf 变为 1，读取 ADC_result0~7 寄存器以获取相应通道的转换结果。
13. 如需对其他通道进行转换，重复执行步骤 6~11。
14. 配置 ADC_CR0.adcen，关闭 ADC 模块。

29.6 连续转换累加模式

在连续转换累加模式下，启动一次 ADC 可对多个通道进行多次转换并对每次转换的结果进行累加；可转换的 ADC 通道为 AIN0~AIN7。ADC 转换的总次数 ADCCR2.ADCCNT 进配置；待进行转换的通道由 ADC_CR2[7:0]进行配置。该模式既可通过设置 ADC_CR0.START 位启动也可通过设置 ADC_CR2[9:0]的外部触发启动。启动连续转换后，ADC 模块依次转换 AIN0~AIN7 中待转换的通道直到总转换次数完成。ADC 模块完成总转换次数后，ADC_RAWINTSR.CONT_INTF 位会自动置 1，转换结果的累加值保存在 ADC_result_acc 寄存器中。

下图演示了对 AIN0、AIN1、AIN5 进行 10 次连续转换累加的过程。通过寄存器将 START

置 1 后，ADC 内部的状态机会依次对 AIN0、AIN1、AIN5 进行转换，直到 ADCCNT 的计数值变为 0。每次转换完成时，ADC_result_acc 寄存器都会自动进行累加。图中给定的 AIN0、AIN1、AIN5 的转换结果依次为 0x010、0x020、0x040。

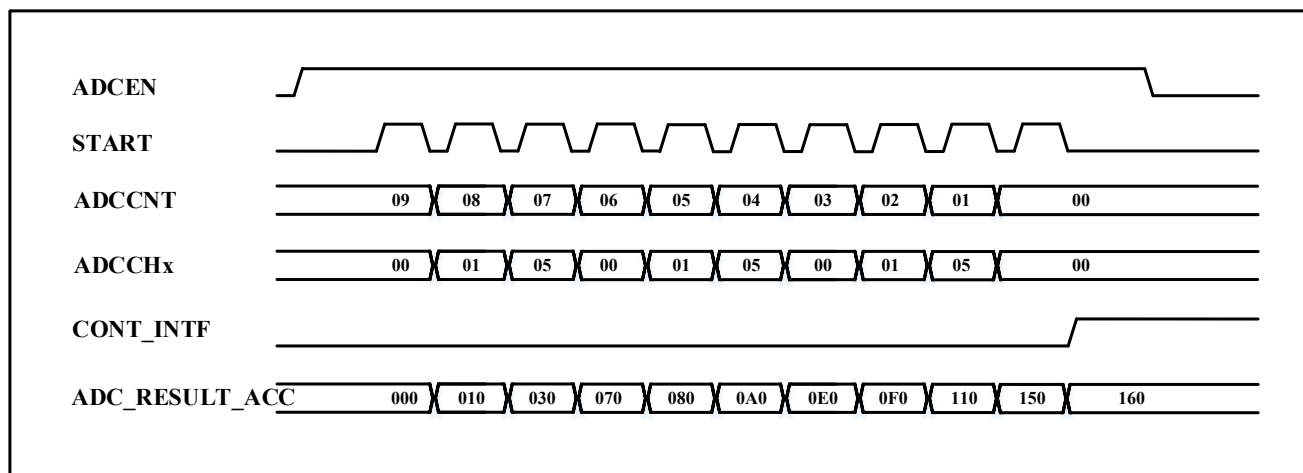


图 29-4. ADC 连续转换累加过程示例

1. 根据 pin 配置对应的 GPIO，将待转换的 ADC 通道配置为模拟端口。
2. 配置 ADC_CR2.circle_mode 为 0，选择非循环模式。
3. 配置 ADC_CR1.ct 为 1，选择连续转换模式。
4. 配置 ADC_CR1.racc_en 为 1，选择 ADC 转换自动累加功能。
5. 配置 ADC_CR1.adccnt，选择连续转换的总转换次数。
6. 配置 ADC_CR0.sam 及 ADC_CR0.clksel，设置 ADC 的转换速度。
7. 配置 ADC_CR2.chen[7:0]，选择待转换的通道。
8. 配置 ADC_INTCLR.cont_intc 为 0，清除 ADC_RAWINTSR.cont_intf 标志。
9. 配置 ADC_MSKINTSR.cont_imsc 为 1，打开掩码中断。
10. 设置 ADC_CR1.racc_clr 为 0，清除 ADC_result_acc 寄存器。
11. 配置 ADC_CR0.staterst 为 1，复位连续转换状态。
12. 配置 ADC_CR0.adcen 为 1，使能 ADC 模块。
13. 配置 ADC_CR0.start 为 1，启动 ADC 连续转换。
14. 等待 ADC_RAWINTSR.cont_intf 变为 1，读取 ADC_result0~7 寄存器以获取相应通

道的转换结果。

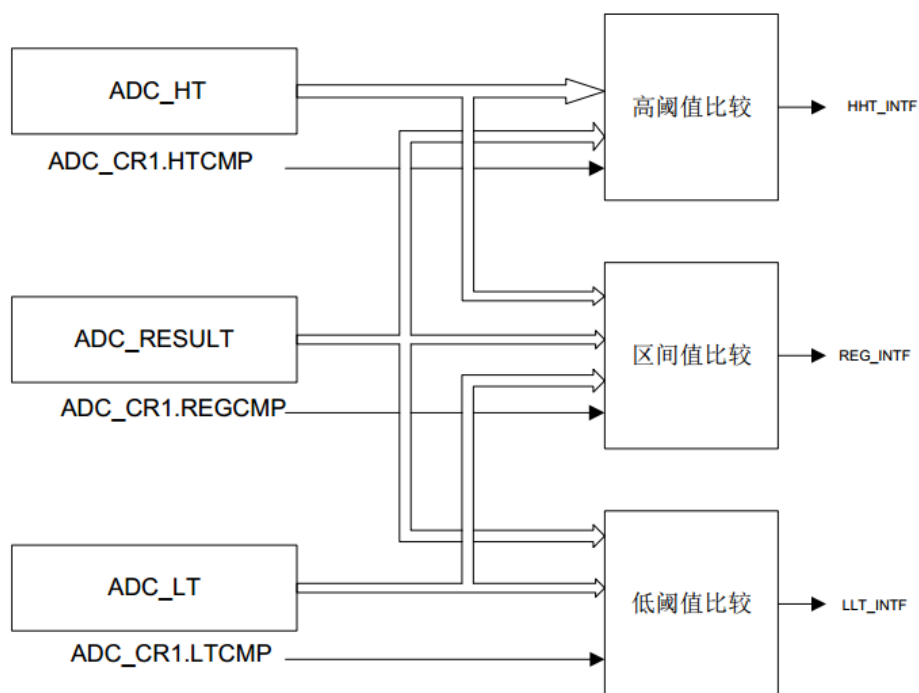
15. 如需对其他通道进行转换，重复执行步骤 6~11。
16. 配置 ADC_CR0.adcen，关闭 ADC 模块。

29.7 ADC 转换结果比较

ADC 转换完成时，ADC 转换结果可以与用户设定的阈值进行比较，支持上阈值比较、下阈值比较、区间值比较。该功能需要将相应的控制位 HtCmp、LtCmp、RegCmp 置 1。该功能可实现对模拟量的自动监测，直到 ADC 转换结果符合用户预期时才产生中断申请用户程序界入。

上阈值比较：当 ADC 转换结果位于 [ADC_HT, 4095] 区间内则 ADC_RAWINTSR.HHT_INTF 置 1；向 ADC_INTCLR.HHT_INTC 写入 0 则清零 ADC_RAWINTSR.HHT_INTF。

下阈值比较：当 ADC 转换结果位于 [0, ADC_LT] 区间内则 ADC_RAWINTSR.LLT_INTF 置 1；向 ADC_INTCLR.LLT_INTC 写入 0 则清零 ADC_RAWINTSR.LLT_INTF。区间值比较：当 ADC 转换结果位于 [ADC_LT, ADC_HT] 区间内则 ADC_RAWINTSR.REG_INTF 置 1；向 ADC_INTCLR.REG_INTC 写入 0 则清零 ADC_RAWINTSR.REG_INTF。



29.8 ADC 中断

ADC 中断请求如下表所示：

中断源	中断标志	中断使能
ADC 连续转换完成	ADC_RAWINTSR.CONT_INTF	ADC_CR0.IE
ADC 转换结果位于区间值区域	ADC_RAWINTSR.REG_INTF	
ADC 转换结果位于上阈值区域	ADC_RAWINTSR.HHT_INTF	
ADC 转换结果比较下阈值区域	ADC_RAWINTSR.LLT_INTF	

29.9 寄存器列表

基地址： 0x4000 2C00

偏移地址	名称	描述	复位值
0x00	ADC_CR0	ADC 配置寄存器 0	0x0000 0000
0x04	ADC_CR1	ADC 配置寄存器 1	0x0000 7000
0x08	ADC_CR2	ADC 配置寄存器 2	0x0000 0000
0x0C	ADC_RESULT0	ADC 通道 0 转换结果	0x0000 0000
0x10	ADC_RESULT1	ADC 通道 1 转换结果	0x0000 0000
0x14	ADC_RESULT2	ADC 通道 2 转换结果	0x0000 0000

偏移地址	名称	描述	复位值
0x18	ADC_RESULT3	ADC 通道 3 转换结果	0x0000 0000
0x1C	ADC_RESULT4	ADC 通道 4 转换结果	0x0000 0000
0x20	ADC_RESULT5	ADC 通道 5 转换结果	0x0000 0000
0x24	ADC_RESULT6	ADC 通道 6 转换结果	0x0000 0000
0x28	ADC_RESULT7	ADC 通道 7 转换结果	0x0000 0000
0x2C	ADC_RESULT	ADC 转换结果	0x0000 0000
0x30	ADC_RESULT_ACC	ADC 转换结果累加值	0x0000 0000
0x34	ADC_HT	ADC 比较上阈值	0x0000 0FFF
0x38	ADC_LT	ADC 比较下阈值	0x0000 0000
0x44	ADC_INTEN	ADC 中断使能寄存器	0x0000 0000
0x48	ADC_INTCLR	ADC 中断清除寄存器	0x0000 0000
0x4C	ADC_RAWINTSR	ADC 掩码前中断状态寄存器	0x0000 0000
0x50	ADC_MSKINTSR	ADC 掩码后中断状态寄存器	0x0000 0000

29.10 寄存器说明

29.10.1 ADC 配置寄存器 0(ADC_CR0)

地址偏移： 0x00

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STA TER ST	保留			S A M	SEL[2:0]			保留	CLKSEL[2:0]			保留	STA RT	ADC EN	
R/W				R/W	R/W				R/W				R/W	R/W	

位	标记	功能描述	复位值	读写
31:16	-	保留，始终读为 0。	0	-
15	STATERST	ADC 连续转换状态控制 0: 无效 1: 复位 ADC 连续转换状态	0x0	R/W
14:12	-	保留，始终读为 0。	0	-

位	标记	功能描述	复位值	读写
11	SAM	ADC 采样周期选择 0: 4 个采样周期 1: 8 个采样周期	0x0	R/W
10:8	SEL[2:0]	ADC 转换通道选择(单次转换模式) 000: 选择通道 0 001: 选择通道 1 010: 选择通道 2 011: 选择通道 3 100: 选择通道 4 101: 选择通道 5 110: 选择通道 6 111: 选择通道 7(VCAP)	0x0	R/W
7	-	保留, 始终读为 0。	0	-
6:4	CLKSEL[2:0]	ADC 时钟选择 000: PCLK 时钟 001: PCLK 时钟 2 分频 010: PCLK 时钟 4 分频 011: PCLK 时钟 8 分频 100: PCLK 时钟 16 分频 101: PCLK 时钟 32 分频 110: PCLK 时钟 64 分频 111: PCLK 时钟 128 分频	0x0	R/W
3:2	-	保留, 始终读为 0。	0	-
1	START	ADC 转换控制 0: 停止 ADC 转换 1: 启动 ADC 转换 注: 该位域软件写 1, 逻辑清零。	0x0	R/W
0	ADCEN	ADC 使能控制 0: 禁止 ADC 1: 使能 ADC	0x0	R/W

29.10.2 ADC 配置寄存器 1(ADC_CR1)

地址偏移: 0x04

复位值: 0x0000 7000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RACC CLR	REGCMP	HTCMP	LTCMP	RACC_EN	CT	TRIGS1[4:0]				TRIGS0[4:0]					
R/W	R/W	R/W	R/W	R/W	R/W	R/W				R/W					

位	标记	功能描述	复位值	读写
31:16	-	保留，始终读为 0。	0	-
15	RACC_CLR	ADC 转换结果累加寄存器清零 0: 无作用; 1: ADC 转换结果累加寄存器(ADC_result_acc)清零。 注: 该 bit 读出为 0, 故操作本寄存器时需要特别注意该 bit 的值以防误动作。	0x0	R/W
14	REGCMP	ADC 区间比较控制 0: 禁止区间比较 1: 使能区间比较	0x1	R/W
13	HTCMP	ADC 高阈值比较控制 0: 禁止高阈值比较 1: 使能高阈值比较	0x1	R/W
12	LTCMP	ADC 低阈值比较控制 0: 禁止低阈值比较 1: 使能低阈值比较	0x1	R/W
11	RACC_EN	ADC 转换结果自动累加控制 0: 禁止 ADC 转换结果自动累加功能 1: 使能 ADC 转换结果自动累加功能	0x0	R/W
10	CT	ADC 转换模式选择 0: 单次转换模式 1: 连续转换模式	0x0	R/W
9:5	TRIGS1[4:0]	ADC 转换自动触发选择 1 00000: 禁用自动触发 ADC 转换 00001: Timer10 中断, 自动触发 ADC 转换 00010: Timer11 中断, 自动触发 ADC 转换 00011: TIM1 中断, 自动触发 ADC 转换 00100: LPTimer 中断, 自动触发 ADC 转换 00101: TIM1TRGO, 自动触发 ADC 转换 00110: TIM2TRGO, 自动触发 ADC 转换	0x0	R/W

位	标记	功能描述	复位值	读写
		00111: TIM2 中断, 自动触发 ADC 转换 01000: UART0 中断, 自动触发 ADC 转换 01001: UART1 中断, 自动触发 ADC 转换 01010: LPUART 中断, 自动触发 ADC 转换 01011: VC0 中断, 自动触发 ADC 转换 01100: NC 01101: RTC 中断, 自动触发 ADC 转换 01110: PCA 中断, 自动触发 ADC 转换 01111: SPI 中断, 自动触发 ADC 转换 10000: PA1 中断, 自动触发 ADC 转换 10001: PA2 中断, 自动触发 ADC 转换 10010: PA3 中断, 自动触发 ADC 转换 10011: PB4 中断, 自动触发 ADC 转换 10100: PB5 中断, 自动触发 ADC 转换 10101: PC3 中断, 自动触发 ADC 转换 10110: PC4 中断, 自动触发 ADC 转换 10111: PC5 中断, 自动触发 ADC 转换 11000: PC6 中断, 自动触发 ADC 转换 11001: PC7 中断, 自动触发 ADC 转换 11010: PD1 中断, 自动触发 ADC 转换 11011: PD2 中断, 自动触发 ADC 转换 11100: PD3 中断, 自动触发 ADC 转换 11101: PD4 中断, 自动触发 ADC 转换 11110: PD5 中断, 自动触发 ADC 转换 11111: PD6 中断, 自动触发 ADC 转换 Note: 1)TIM4/5/6 中断触发 ADC 自动转换, 除了需要使能 TIM4/5/6 的相应中断外, 还需要配置 AdvancedTimer 的展频及中断触发选择寄存器 TIMX_CR 选择可以触发 ADC 的中断源。2)触发 ADC 使用的是各中断标志位的上升沿。如果需要重复触发, 需要清除中断标志。如果不需要进入中断服务程序, 请不要使能 NVIC 的中断使能。		
4:0	TRIGS0[4:0]	ADC 转换自动触发选择 1 00000: 禁用自动触发 ADC 转换 00001: Timer10 中断, 自动触发 ADC 转换 00010: Timer11 中断, 自动触发 ADC 转换 00011: TIM1 中断, 自动触发 ADC 转换 00100: LPTimer 中断, 自动触发 ADC 转换 00101: TIM1TRGO, 自动触发 ADC 转换	0x0	R/W

位	标记	功能描述	复位值	读写
		00110: TIM2TRGO, 自动触发 ADC 转换 00111: TIM2 中断, 自动触发 ADC 转换 01000: UART0 中断, 自动触发 ADC 转换 01001: UART1 中断, 自动触发 ADC 转换 01010: LPUART 中断, 自动触发 ADC 转换 01011: VC0 中断, 自动触发 ADC 转换 01100: NC 01101: RTC 中断, 自动触发 ADC 转换 01110: PCA 中断, 自动触发 ADC 转换 01111: SPI 中断, 自动触发 ADC 转换 10000: PA1 中断, 自动触发 ADC 转换 10001: PA2 中断, 自动触发 ADC 转换 10010: PA3 中断, 自动触发 ADC 转换 10011: PB4 中断, 自动触发 ADC 转换 10100: PB5 中断, 自动触发 ADC 转换 10101: PC3 中断, 自动触发 ADC 转换 10110: PC4 中断, 自动触发 ADC 转换 10111: PC5 中断, 自动触发 ADC 转换 11000: PC6 中断, 自动触发 ADC 转换 11001: PC7 中断, 自动触发 ADC 转换 11010: PD1 中断, 自动触发 ADC 转换 11011: PD2 中断, 自动触发 ADC 转换 11100: PD3 中断, 自动触发 ADC 转换 11101: PD4 中断, 自动触发 ADC 转换 11110: PD5 中断, 自动触发 ADC 转换 11111: PD6 中断, 自动触发 ADC 转换 Note: 1)TIM4/5/6 中断触发 ADC 自动转换, 除了需要使能 TIM4/5/6 的相应中断外, 还需要配置 Advanced Timer 的展频及中断触发选择寄存器 TIMX_CR 选择可以触发 ADC 的中断源。2)触发 ADC 使用的是各中断标志位的上升沿。如果需要重复触发, 需要清除中断标志。如果不需要进入中断服务程序, 请不要使能 NVIC 的中断使能。		

29.10.3 ADC 配置寄存器 2(ADC_CR2)

地址偏移: 0x08

复位值： 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:17	-	保留，始终读为 0。	0	-
16	CIRCLE_MODE	ADC 转换循环模式选择 0: 非循环模式 1: 循环模式	0x0	R/W
15:8	ADCCNT[7:0]	ADC 连续转换次数配置 0: 连续转换 1 次 1: 连续转换 2 次 255: 连续转换 256 次	0x0	R/W
7:0	CHEN[7:0]	ADC 连续转换通道 7~0 使能 0: 禁止 1: 使能	0x0	R/W

29.10.4 ADC 通道 0 转换结果(ADC_RESULT 0)

地址偏移： 0x0C

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT0[11:0]											
				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留，始终读为 0。	0	-
11:0	RESULT0[11:0]	ADC 通道 0 转换结果	0x0	RO

29.10.5 ADC 通道 1 转换结果(ADC_RESULT 1)

地址偏移: 0x10

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT1[11:0]											
				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留, 始终读为 0。	0	-
11:0	RESULT1[11:0]	ADC 通道 1 转换结果	0x0	RO

29.10.6 ADC 通道 2 转换结果(ADC_RESULT 2)

地址偏移: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT2[11:0]											
				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留, 始终读为 0。	0	-
11:0	RESULT2[11:0]	ADC 通道 2 转换结果	0x0	RO

29.10.7 ADC 通道 3 转换结果(ADC_RESULT 3)

地址偏移: 0x18

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT3[11:0]											
				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留，始终读为 0。	0	-
11:0	RESULT3[11:0]	ADC 通道 3 转换结果	0x0	RO

29.10.8 ADC 通道 4 转换结果(ADC_RESULT 4)

地址偏移： 0x1C

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT4[11:0]											
				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留，始终读为 0。	0	-
11:0	RESULT4[11:0]	ADC 通道 4 转换结果	0x0	RO

29.10.9 ADC 通道 5 转换结果(ADC_RESULT 5)

地址偏移： 0x20

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT5[11:0]											
				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留，始终读为 0。	0	-
11:0	RESULT5[11:0]	ADC 通道 5 转换结果	0x0	RO

29.10.10 ADC 通道 6 转换结果(ADC_RESULT 6)

地址偏移： 0x24

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT6[11:0]											
				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留，始终读为 0。	0	-
11:0	RESULT6[11:0]	ADC 通道 6 转换结果	0x0	RO

29.10.11 ADC 通道 7 转换结果(ADC_RESULT 7)

地址偏移： 0x28

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RESULT7[11:0]											
				RO											

位	标记	功能描述	复位值	读写
31:12	-	保留，始终读为 0。	0	-
11:0	RESULT7[11:0]	ADC 通道 7 转换结果	0x0	RO

29.10.12 ADC 转换结果(ADC_RESULT)

地址偏移： 0x2C

复位值： 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:12	-	保留，始终读为 0。	0	-
11:0	RESULT[11:0]	ADC 转换结果	0x0	RO

29.10.13 ADC 转换结果累加值(ADC_RESULT_ACC)

地址偏移： 0x30

复位值： 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:20	-	保留，始终读为 0。	0	-
19:0	RESULT_ACC[19:0]	ADC 转换结果累加值	0x0	RO

29.10.14 ADC 比较上阈值(ADC_HT)

地址偏移： 0x34

复位值： 0x0000 0FFF

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

位	标记	功能描述	复位值	读写
31:12	-	保留，始终读为 0。	0	-
11:0	HT[11:0]	ADC 转换结果比较上阈值	0xFFFF	R/W

29.10.15 ADC 比较下阈值(ADC_LT)

地址偏移： 0x38

复位值： 0x0000 0000

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

位	标记	功能描述	复位值	读写
31:12	-	保留，始终读为 0。	0	-
11:0	LT[11:0]	ADC 转换结果比较下阈值	0x0	R/W

29.10.16 ADC 中断使能寄存器(ADC_INTEN)

地址偏移： 0x44

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CO NT _IEN	REG _IEN	HHT _IEN	LLT _IEN	ADCXIEN[7:0]							
				R/W	R/W	R/W	R/W	R/W							

位	标记	功能描述	复位值	读写
31:12	-	保留，始终读为 0。	0	-
11	CONT_IEN	连续转换完成中断掩码配置 0: 禁止中断 1: 使能中断	0x0	R/W
10	REG_IEN	ADC 转换结果比较区间中断掩码配置 0: 禁止 1: 使能	0x0	R/W
9	HHT_IEN	ADC 转换结果比较上阈值中断掩码配置 0: 禁止 1: 使能	0x0	R/W
8	LLT_IEN	ADC 转换结果比较下阈值中断掩码配置 0: 禁止 1: 使能	0x0	R/W
7:0	ADCXIEN[7:0]	ADC 通道 7~0 中断掩码配置	0x0	R/W

位	标记	功能描述	复位值	读写
		0: 禁止 1: 使能		

29.10.17 ADC 中断清除寄存器(ADC_INTCLR)

地址偏移: 0x48

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CON T_IN TC	REG _INT _C	HHT _INT _C	LLT _INT _C	ADCICLR[7:0]							
				WO	WO	WO	WO	WO							

位	标记	功能描述	复位值	读写
31:12	-	保留, 始终读为 0。	0	-
11	CONT_INTC	写 1 清除连续转换完成标志写 0 无作用	0x0	WO
10	REG_INTC	写 1 清除 ADC 转换结果比较区间标志写 0 无作用	0x0	WO
9	HHT_INTC	写 1 清除 ADC 转换结果比较上阈值写 0 无作用	0x0	WO
8	LLT_INTC	写 1 清除 ADC 转换结果比较下阈值标志写 0 无作用	0x0	WO
7:0	ADCICLR[7:0]	写 1 清除 ADC 通道 7~0 中断状态写 0 无作用	0x0	WO

29.10.18 ADC 掩码前中断状态寄存器(ADC_RAWINTSR)

地址偏移: 0x4C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CON T_IN TF	REG _INT _F	HHT _INT _F	LLT _INT _F	ADCRIS[7:0]							
				RO	RO	RO	RO	RO							

位	标记	功能描述	复位值	读写
31:12	-	保留，始终读为 0。	0	-
11	CONT_INTF	连续转换完成标志 0: ADC 连续转换未完成 1: ADC 连续转换完成	0x0	RO
10	REG_INTF	ADC 转换结果比较区间标志 0: ADC 转换结果位于[ADC_LT, ADC_HT]区间外 1: ADC 转换结果位于[ADC_LT, ADC_HT]区间内	0x0	RO
9	HHT_INTF	ADC 转换结果比较上阈值标志 0: ADC 转换结果位于[ADC_HT, 4095]区间外 1: ADC 转换结果位于[ADC_HT, 4095]区间内	0x0	RO
8	LLT_INTF	ADC 转换结果比较下阈值标志 0: ADC 转换结果位于[0, ADC_LT]区间外 1: ADC 转换结果位于[0, ADC_LT]区间内	0x0	RO
7:0	ADCRIS[7:0]	ADC 通道 7~0 掩码前中断状态	0x0	RO

29.10.19 ADC 掩码后中断状态寄存器(ADC_MSKINTSR)

地址偏移: 0x50

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CON T_M IF	REG _MI _F	HHT _MI _F	LLT _MI _F	ADCMIS[7:0]							
				RO	RO	RO	RO	RO							

位	标记	功能描述	复位值	读写
31:12	-	保留，始终读为 0。	0	-

位	标记	功能描述	复位值	读写
11	CONT_MIF	连续转换完成掩码后中断 0: ADC 连续转换未完成 1: ADC 连续转换完成	0x0	RO
10	REG_MIF	ADC 转换结果比较区间掩码后中断 0: ADC 转换结果位于[ADC_LT, ADC_HT]区间外 1: ADC 转换结果位于[ADC_LT, ADC_HT]区间内	0x0	RO
9	HHT_MIF	ADC 转换结果比较上阈值掩码后中 0: ADC 转换结果位于[ADC_HT, 4095]区间外 1: ADC 转换结果位于[ADC_HT, 4095]区间内	0x0	RO
8	LLT_MIF	ADC 转换结果比较下阈值掩码后中断 0: ADC 转换结果位于[0, ADC_LT]区间外 1: ADC 转换结果位于[0, ADC_LT]区间内	0x0	RO
7:0	ADCMIS[7:0]	ADC 通道 7~0 掩码后中断状态	0x0	RO

30. 低电压检测器 (LVD)

30.1 简介

LVD 可用于监测工作电压，当被监测电压与 LVD 阈值的比较结果满足触发条件时，LVD 会产生中断或复位信号。中断或复位信号只能被中断或复位清零信号清除。只有当中断或复位信号被清零后，才会产生在触发条件下，再次产生中断或复位信号。

支持检测一路的输入电压与阈值电压 1.25V 比较。

采样滤波时钟可配置，可配置为 apb 时钟或者 LIRC。滤波计数值可配置。采样达到滤波计数值次的结果一致输出。

3 种触发条件：高电平，上升沿，下降沿组合。

2 种触发结果：中断，复位信号(禁止在滤波时钟选择 PCLK 时选择产生复位信号)。中断与复位信号不能同时产生。

30.2 特性

- ◆ 可对芯片电源电压或芯片引脚电压进行检测
- ◆ 8 档电压监测值(2.5-4.4V)
- ◆ 可根据上升/下降边沿产生异步中断或复位
- ◆ 具有硬件迟滞电路和可配置的软件防抖功能

30.3 功能描述

30.3.1 LVD 框图

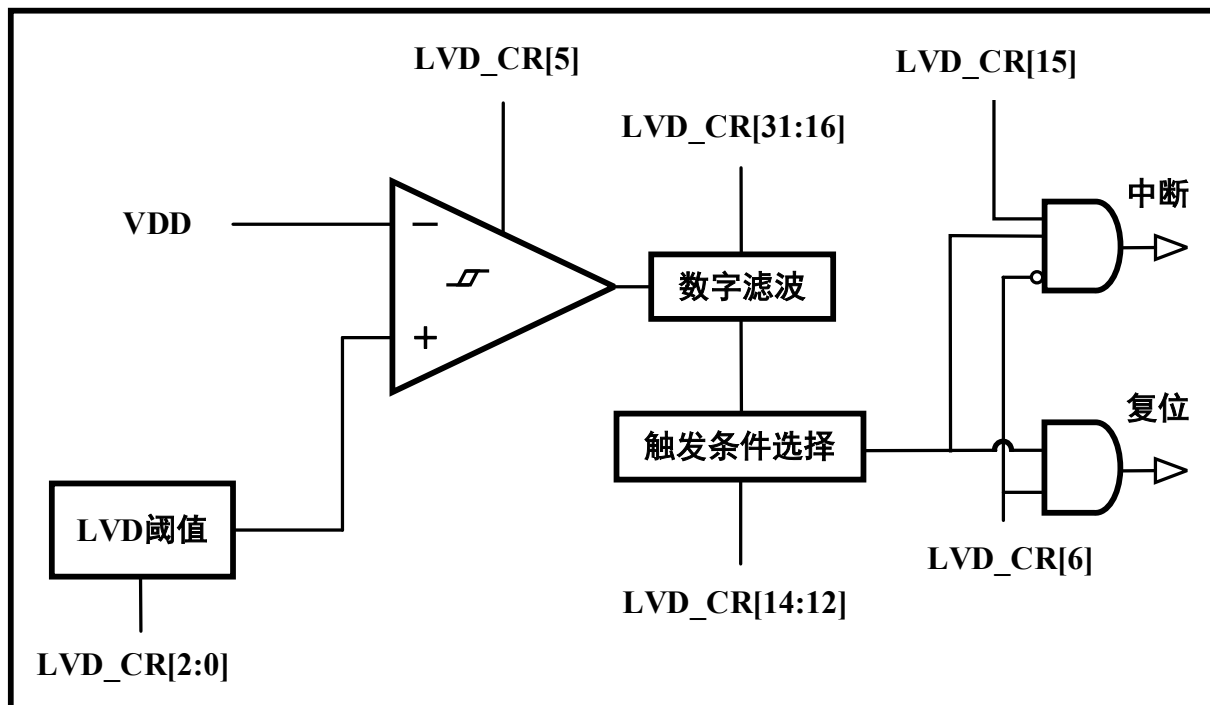


图 30-1. LVD 结构框图

30.3.2 数字滤波

如果芯片的工作环境恶劣，迟滞比较器的输出会出现噪声信号。使能数字滤波模块，则迟滞比较器的输出波形中脉宽小于 LVD_CR.Debounce_time 的噪声信号都可以被滤除。禁止数字滤波模块，则数字滤波模块的输入输出信号相同。使能数字滤波模块，滤波示意如下所示：

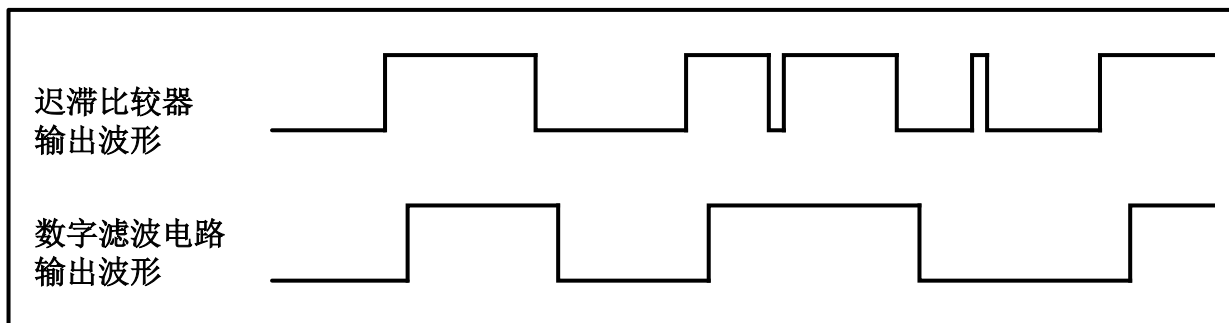


图 30-2. LVD 滤波输出

30.3.3 迟滞功能

LVD 内置的电压比较器具有迟滞功能，其输出信号会等到输入信号高于或低于阈值电压

20mV 后才发生翻转。迟滞功能可以增强芯片的抗干扰能力，如下图所示。

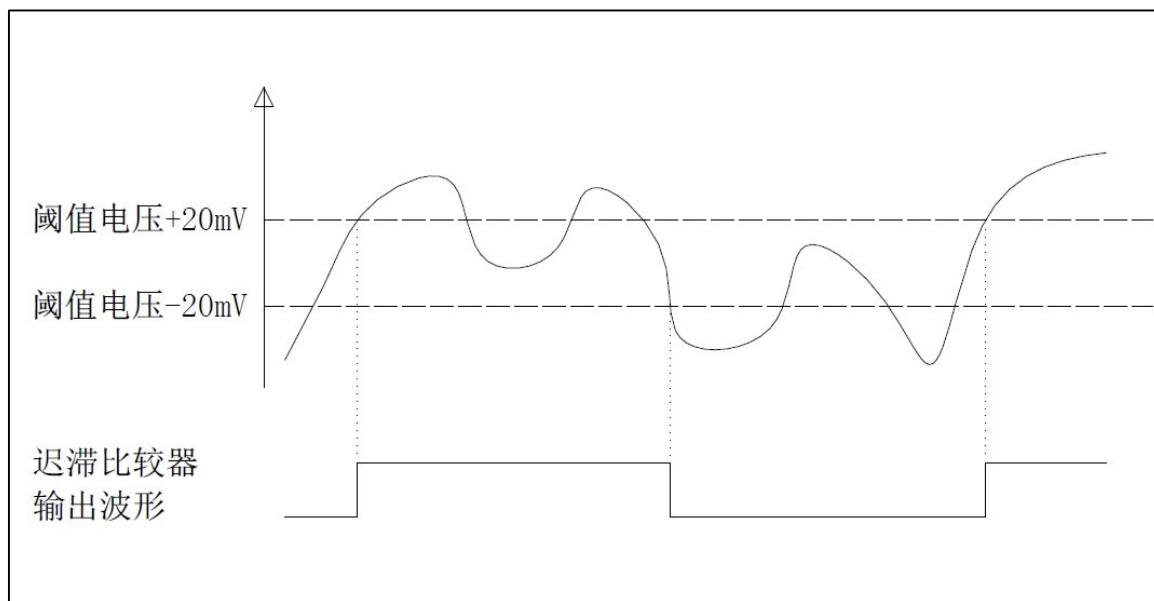


图 30-3. LVD 迟滞响应

30.4 配置示例

30.4.1 LVD 配置为低电压复位

在本模式下，监测电压低于阈值电压时复位 MCU。配置方法如下所示：

- 1) 配置 LVD_CR.DIV_SEL，选择待监测的电压分压。
- 2) 配置 LVD_CR.FLT_NUM，选择 LVD 滤波时间。
- 3) 配置 LVD_CR.FLTCLK_SEL,选择滤波时钟。
- 4) 配置 LVD_CR.FLTEN，使能 LVD 滤波。
- 5) 设置 LVD_CR.HTEN 为 1，选择高电平触发 LVD 动作。
- 6) 设置 LVD_CR.ACT 为 1，选择 LVD 动作为复位。
- 7) 设置 LVD_CR.LVDEN 为 1，使能 LVD。

30.4.2 LVD 配置为电压变化中断

在本模式下，监测电压高于或低于阈值电压时产生中断。配置方法如下所示：

- 1) 配置 LVD_CR.DIV_SEL，选择待监测的电压来源。
- 2) 配置 LVD_CR.FLT_NUM，选择 LVD 滤波时间。

- 3) 配置 LVD_CR.FLTCLK_SEL，选择滤波时钟。
- 4) 配置 LVD_CR.FLTEN，使能 LVD 滤波。
- 5) 设置 LVD_CR.HighintEN 和 LVD_CR.FTEN 为 1，选择高电平触发 LVD 动作。
- 6) 设置 LVD_CR.ACT 为 0，选择 LVD 动作为中断。
- 7) 设置 LVD_CR.INT_EN 为 1，使能 LVD 中断。
- 8) 设置 LVD_CR.LVDEN 为 1，使能 LVD。
- 9) 在中断服务程序中向 LVD_IFR 写入 0x00 以清除中断标志。

30.5 寄存器列表

基地址：0x40004000

偏移地址	名称	描述	默认值
0x00	LVD_CR	LVD 控制寄存器	0x0000007
0x04	LVD_SR	LVD 状态寄存器	0x0000000

30.6 寄存器说明

30.6.1 LVD 控制寄存器(LVD_CR)

地址偏移：0x00

复位值：0x00000007

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT_NUM[15:0]															
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_EN	HIGH_INT_EN	RISE_INT_EN	FALL_INT_EN	保留		FLTCLK_SEL [1:0]		FLT_EN	ACT	LVD_EN	保留		DIV_SEL [2:0]		
R/W	R/W	R/W	R/W			R/W		R/W	R/W	R/W			R/W		

位	标记	功能描述	复位值	读写
31:16	FLT_NUM[15:0]	LVD 采样滤波次数 采样时钟为 apb 时钟或者 clk_40khz。滤波计数值可	0x0	R/W

位	标记	功能描述	复位值	读写
		配置。 采样次数达到滤波计数值时,结果一致输出。 采样计数周期=FLT_NUM[15:0]		
15	INT_EN	LVD 中断使能 0: 禁止 1: 使能	0	R/W
14	HIGHINTEN	高电平触发使能(被监测电压低于阈值电压) 0: 禁止 1: 使能	0	R/W
13	RISEINTEN	上升沿触发使能(被监测电压从高于阈值电压变为低于阈值电压) 0: 禁止 1: 使能	0	R/W
12	FALLINTEN	下降沿触发使能(被监测电压从低于阈值电压变为高于阈值电压) 0: 禁止 1: 使能	0	R/W
11:10	—	保留	0x0	—
9:8	FLTCLK_SEL[1:0]	滤波时钟选择。 00: 滤波时钟无效 01: 滤波时钟选择为 pclk; (只能配置成中断模式) 10: 滤波时钟选择为 clk_38.4khz 11: 保留	0x0	R/W
7	FLTEN	数字滤波功能配置 0: 禁止数字滤波 1: 使能数字滤波	0	R/W
6	ACT	LVD 中断复位选择位 0: 产生中断 1: 产生复位	0	R/W
5	LVDEN	LVD 使能, 用于控制 analog 模块。 LVD 控制 0: 禁止 LVD 1: 使能 LVD	0	R/W
4:3	-	保留	0x0	—
2:0	DIV_SEL[2:0]	LVD 分压配置, 用于控制 analog 模块。 000: 4.4V 001: 4.0V 010: 3.6V 011: 3.3V 100: 3.1V 101: 2.9V 110: 2.7V 111: 2.5V	0x7	R/W

30.6.2 LVD 状态寄存器(LVD_SR)

地址偏移: 0x04

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														INTF	
														W0C	

位	标记	功能描述	复位值	读写
31:1	—	保留	0x0	—
0	INTF	LVD 中断标志 0: 未发生中断 1: 发生 LVD 中断 写 0 清除中断标志, 写 1 无效	0	W0C

31. 电压比较器(VC)

31.1 简介

模拟电压比较器 VC 用于比较两个输入模拟电压的大小，并根据比较结果输出高/低电平。

当“+”输入端电压高于“-”输入端电压时，电压比较器输出高电平；当“+”输入端电压低于“-”输入端电压时，电压比较器输出低电平。

“+”输入端与“-”输入端均支持 4 路电源输入选择

3 种触发条件：高电平，上升沿，下降沿组合。

2 种触发结果：中断，复位信号。中断与复位信号不能同时产生。

31.2 VC 框图

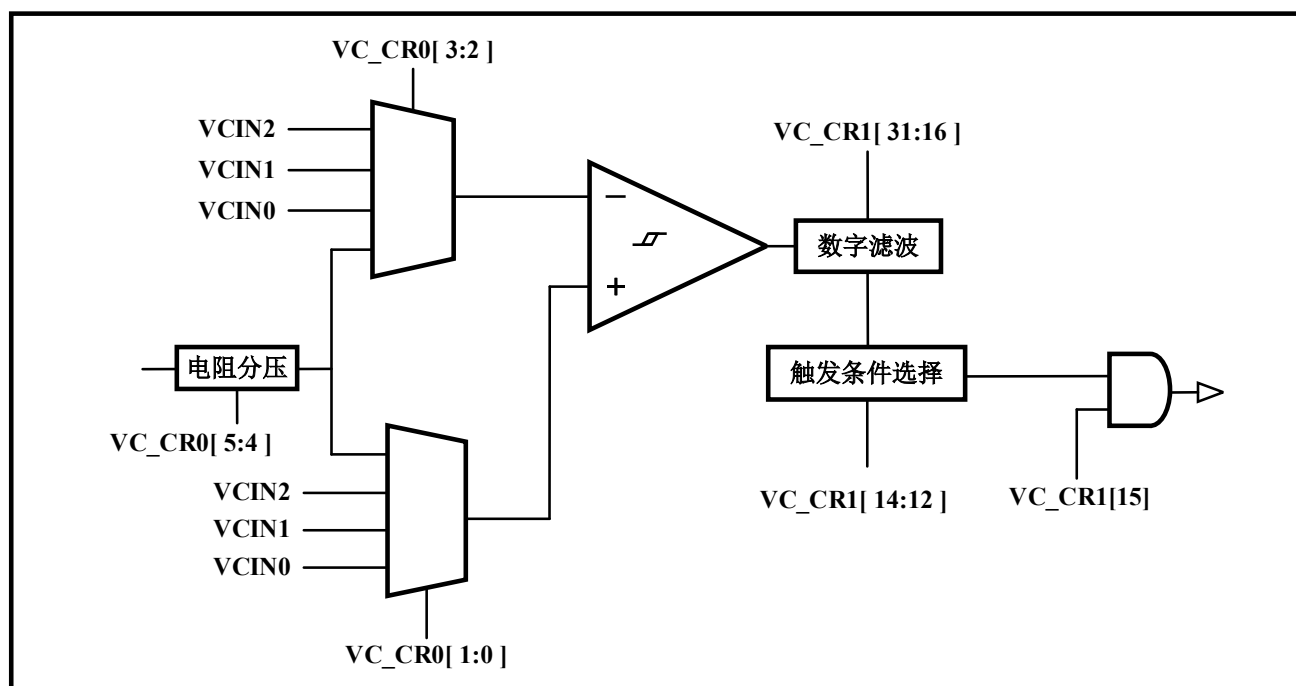


图 31-1. VC 结构框图

31.3 数字滤波

如果芯片的工作环境恶劣，迟滞比较器的输出会出现噪声信号。使能数字滤波模块，则迟滞比较器的输出波形中脉宽小于 LVD_CR.Debounce_time 的噪声信号都可以被滤除。禁止数字

滤波模块，则数字滤波模块的输入输出信号相同。使能数字滤波模块，滤波示意如下所示：

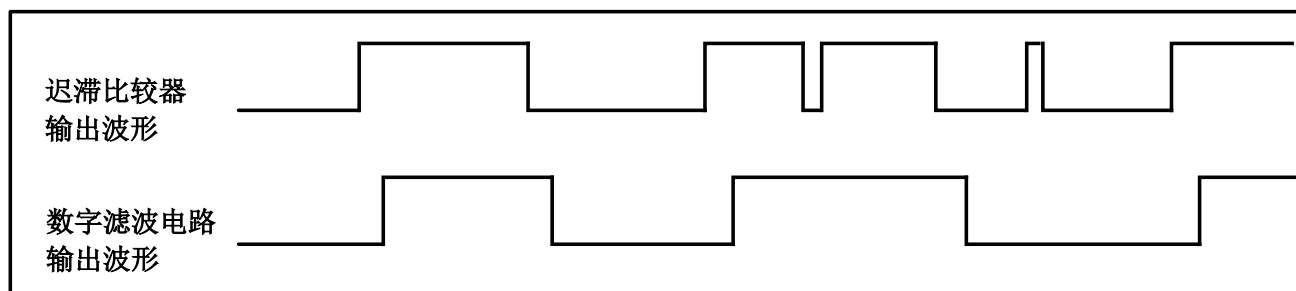


图 31-2. LVD 滤波输出

31.4 迟滞功能

LVD 内置的电压比较器具有迟滞功能，其输出信号会等到输入信号高于或低于阈值电压 20mV 后才发生翻转。迟滞功能可以增强芯片的抗干扰能力，如下图所示。

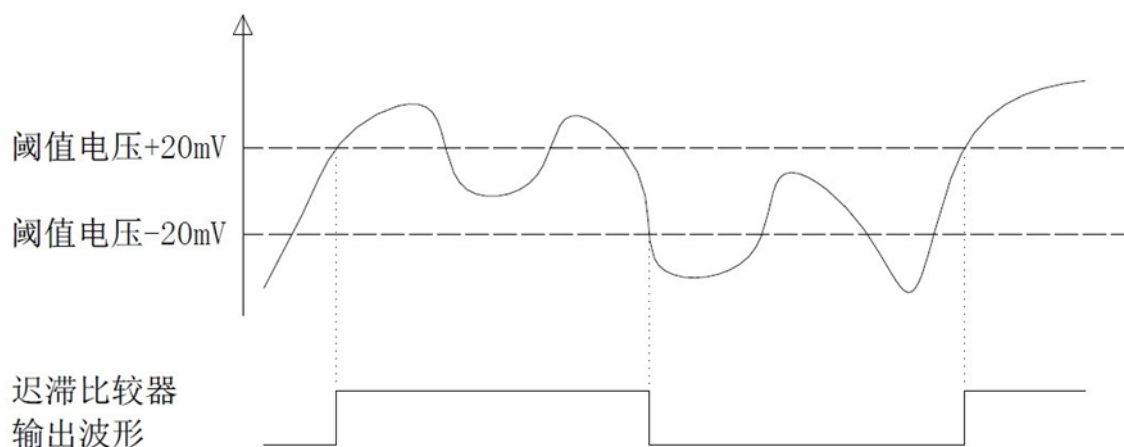


图 31-3. LVD 迟滞响应

31.5 配置示例

在本模式下，监测电压高于或低于阈值电压时产生中断。配置方法如下所示：

Step1: 配置 VC_PCR.V25DIV_EN 使能分压。

Step2: 配置 VC_PCR.V25DIV，设置分压系数。

Step3: 配置 VC_PCR.NinSEL，选择“-”端待监测的电压来源。

Step4: 配置 VC_PCR.PinSEL, 选择 “+” 端待监测的电压来源。

Step5: 配置 VC_CR.FLT_NUM, 选择 VC 滤波时间。

Step6: 配置 VC_CR.FLTCLK_SEL, 选择滤波时钟。

Step7: 配置 VC_CR.FLTEN, 使能 VC 滤波。

Step8: 设置 VC_CR.HighintEN、RiseintEN、FallintEN, 选择触发模式。

Step9: 设置 VC_CR.INT_EN 为 1, 使能 VC 中断。

Step10: 设置 VC_CR.VCEN 为 1, 使能 VC。

Step11: 在中断服务程序中向 VC_ISR 写入 0x00 以清除中断标志。

31.6 寄存器列表

基地址: 0x40004000

偏移地址	名称	描述	默认值
0x080	VC_CR0	VC 控制寄存器 0	0x0000000
0x084	VC_CR1	VC 控制寄存器 1	0x0000000
0x088	VC_OUTCFG	VC 输出配置寄存器	0x0000000
0x08c	VC_SR	VC 状态寄存器	0x0000000

31.7 寄存器说明

31.7.1 VC 电压控制寄存器(VC_CR0)

地址偏移: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									V25 DIV _{EN}	V25DIV [1:0]	NINSEL [1:0]	PINSEL [1:0]			
									R/W	R/W	R/W	R/W			

位	标记	功能描述	复位值	读写
31:7	—	保留	0x0	—
6	V25DIV_EN	VC 使能，用于控制 analog 模块。 V25 分频控制 0: 禁止 1: 使能	0x0	R/W
5:4	V25DIV[1:0]	VC 电压分压控制，用于控制 analog 模块。 00: 1/4 V25 01: 2/4 V25 10: 3/4 V25 11: V25	0x0	R/W
3:2	NINSEL[1:0]	”-”输入端电压电压选择。 00: VCIN[0] 01: VCIN[1] 10: VCIN[2] 11: V25DIV	0x0	R/W
1:0	PINSEL[1:0]	”+”输入端电压电压选择 00: VCIN[0] 01: VCIN[1] 10: VCIN[2] 11: V25DIV	0x0	R/W

31.7.2 VC 控制寄存器(VC_CR1)

地址偏移： 0x04

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT_NUM[15:0]															
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_EN	HIGH_INT_EN	RIS_EI	INT_EN	保留			FLT_EN	保留				VC_FLTC_LK_SEL [1:0]		保留	VCE_N
R/W	R/W	R/W	R/W				R/W					R/W			R/W

位	标记	功能描述	复位值	读写
31:16	FLT_NUM[15:0]	VC 采样滤波计数值 采样时钟为 APB 时钟或者 LIRC。滤波计数值可配置。采样次数达到滤波计数值时，结果一致输出。 采样计数周期=FLT_NUM[15:0]	0x0	—

位	标记	功能描述	复位值	读写
15	INT_EN	VC 中断使能 0: 禁止 1: 使能	0	R/W
14	HIGHINTEN	VC 输出信号高电平触发使能 0: 禁止 1: 使能	0	R/W
13	RISEINTEN	VC 输出信号上升沿触发使能 0: 禁止 1: 使能	0	R/W
12	FALLINTEN	VC 输出信号下降沿触发使能 0: 禁止 1: 使能	0	R/W
11:9	-	保留	0x0	R/W
8	FLTEN	数字滤波功能配置 0: 禁止数字滤波 1: 使能数字滤波	0	R/W
7:4	-	保留	0x0	R/W
3:2	VC_FLTCLK_SEL[1:0]	VC 滤波时钟选择 00: 滤波时钟无效 01: 滤波时钟选择为 PCLK 10: 滤波时钟选择为 LIRC 11: 保留	0x0	R/W
1	-	保留	0	R/W
0	VCEN	VC Enable 0: 电压比较功能禁止 1: 电压比较功能使能	0	R/W

31.7.3 VC 输出配置寄存器(VC_OUTCFG)

地址偏移: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													INV_PAD	TM1_BKE	TM1_CH3_EN
													R/W	R/W	R/W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INV_TM1_CH4	TM1_CH3_EN	INV_TM1_CH3	TM1_CH2_EN	INV_TM1_CH2	TM1_CH1_EN	INV_TM1_CH1	PCAE_CI_EN	PCAE_CAP0E	INV_PCA	LPTIM_EXT_EN	LPTIM_EN	保留	TIM1_EN	TIM0_EN	INV_TIM_X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:19	-	保留	0x0	-
18	INV_PAD	VC filter 结果输出到 PAD 反向使能 0: 禁止; 1: 使能	0	R/W
17	TM1BKE	VC 中断作为 TIM1 刹车控制 0: 禁止; 1: 使能	0	R/W
16	TM1CH4_EN	VC filter 结果输出到 TIM1 CH4 门控使能 0: 禁止; 1: 使能	0	R/W
15	INV_TM1CH4	VC filter 结果输出到 TIM1 CH4 门控反向使能 0: 禁止; 1: 使能	0	R/W
14	TM1CH3_EN	VC filter 结果输出到 TIM1 CH3 门控使能 0: 禁止; 1: 使能	0	R/W
13	INV_TM1CH3	VC filter 结果输出到 TIM1 CH3 门控反向使能 0: 禁止; 1: 使能	0	R/W
12	TM1CH2_EN	VC filter 结果输出到 TIM1 CH2 门控使能 0: 禁止; 1: 使能	0	R/W
11	INV_TM1CH2	VC filter 结果输出到 TIM1 CH2 门控反向使能 0: 禁止; 1: 使能	0	R/W
10	TM1CH1_EN	VC filter 结果输出到 TIM1 CH1 门控使能 0: 禁止; 1: 使能	0	R/W
9	INV_TM1CH1	VC filter 结果输出到 TIM1 CH1 门控反向使能 0: 禁止; 1: 使能	0	R/W
8	PCAECI_EN	VC filter 结果输出到 PCA 外部时钟使能 0: 禁止; 1: 使能	0	R/W
7	PCACAP0_EN	VC filter 结果输出到 PCA 捕获 0 使能 0: 禁止; 1: 使能	0	R/W
6	INV_PCA	VC filter 结果输出到 PCA 反向使能 0: 禁止; 1: 使能	0	R/W
5	LPTIMEXT_EN	VC filter 结果输出到 LPTIM 外部时钟使能控制 0: 禁止; 1: 使能	0	R/W
4	LPTIM_EN	VC filter 结果输出到 LPTIM 门控使能 0: 禁止; 1: 使能	0	R/W
3	-	保留	0	-
2	TIM1_EN	VC filter 结果输出到 TIM11 门控使能 0: 禁止; 1: 使能	0	R/W
1	TIM0_EN	VC filter 结果输出到 TIM10 门控使能 0: 禁止; 1: 使能	0	R/W
0	INV_TIMX	VC filter 结果输出到 TIM10, TIM11, LPTIM 门控反向使能 0: 禁止; 1: 使能	0	R/W

31.7.4 VC 状态寄存器(VC_SR)

地址偏移： 0x0C

复位值： 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													VC_FLOUT	INTF	
													RO	W0C	

位	标记	功能描述	复位值	读写
31:2	-	保留	0x0	-
1	VC_FLOUT	VC 的 Filter 后状态 0: VC 滤波结果为 0 1: VC 滤波结果为 1	0	RO
0	INTF	VC 中断标志 0: 未发生 VC 中断 1: 发生 VC 中断 写 0 清除中断标志, 写 1 无效	0	W0C

32. Debug 支持

通用 MCU 的 JTAG/SWD 是非加密的传统型 JTAG/SWD 调试接口，这在客户程序的保密性，以及系统的安全性方面具有非常大的隐患。本产品为了保护用户程序及系统安全性，在 SWD 端口上集成设计了安全授权电路。本产品出厂时，端口默认配置 SWD 调试接口，配合本产品的上位机软件或客户程序，在上位机软件 security bit 位置处写入 0 值时，SWD 调试端口在复位或下次上电后自动断开。调试接口 SWD 不可以通过 security byte 写 0XFF 打开，如果需要打开 SWD 接口只有对芯片进行全片擦除。

注意：

- ◆ 当用户没有设置 security bit 时，PC7 和 PD1 自动被配置成 SWD 调试端口(PC7 上拉，PD1 下拉)；用户也可以通过配置 RCC_SWDIOCR.SWDPORT 寄存器将调试接口配置成 GPIO。
- ◆ 当用户设置 security bit 的值为 0 时，PC7 和 P31 端口与 SWD 调试端口自动断开，即不能使用 SWD 调试功能。RCC_SWDIOCR.SWDPORT 不能被写 1。

32.1 SWD 保护位工作原理

- 1) 客户拿到的芯片是空白片，security bit 的值是 1，所以 SWD 是默认打开的。
- 2) 客户使用 Keil 进行软件开发，开发完成后，可以通过 Keil 直接下载，也可以通过烧录器下载。
- 3) 客户下载完程序之后，通过上位机进行密钥下载,开启保护位。
- 4) MCU 重新复位之后，密钥即刻生效。
- 5) 客户再需要进行 SWD 调试的话，需要将芯片全片擦除，才能重新打通 SWD 通道。

32.2 寄存器列表

Debug 基址：0x40004C00

表 32.1. Debug 寄存器映象和复位值

偏移地址	名称	描述	默认值
0x0	DBG_APBFZ	Debug 模式控制寄存器	0x00000000

32.3 Debug 模式控制寄存器(DBG_APBFZ)

地址偏移: 0x00

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY															
WO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				TIM2 DBG STOP	WWDG DBG STOP	IWDG DBG STOP	BEEP DBG STOP	保留	RTC DBG STOP	TIM1 DBG STOP	PCA DBG STOP	保留	LPTIM DBG STOP	TIM11 DBG STOP	TIM10 DBG STOP
				R/W	R/W	R/W	R/W		R/W	R/W	R/W		R/W	R/W	R/W

位	标记	功能描述	复位值	读写
31:16	KEY	只有高位写 0x5A69 时配置该寄存器才有效, 写其它值时无效。	0x0	WO
15:12	—	保留	14:12	—
11	TIM2DBGSTOP	TIM2 调试模式停止工作 0: 调试模式计数器仍然工作 1: 调试模式计数器停止工作	0	R/W
10	WWDGDBGSTOP	WWDG 调试模式停止工作 0: 调试模式计数器仍然工作 1: 调试模式计数器停止工作	0	R/W
9	IWDGDBGSTOP	IWDG 调试模式停止工作 0: 调试模式计数器仍然工作 1: 调试模式计数器停止工作	0	R/W
8	BEEPDBGSTOP	BEEP 调试模式停止工作 0: 调试模式计数器仍然工作 1: 调试模式计数器停止工作	0	R/W
7	—	保留	0	—
6	RTCDBGSTOP	RTC 调试模式停止工作 0: 调试模式计数器仍然工作 1: 调试模式计数器停止工作	0	R/W
5	TIM1DBGSTOP	TIM1 调试模式停止工作 0: 调试模式计数器仍然工作 1: 调试模式计数器停止工作	0	R/W
4	PCADBGSTOP	PCA 调试模式停止工作 0: 调试模式计数器仍然工作	0	R/W

位	标记	功能描述	复位值	读写
		1: 调试模式计数器停止工作		
3	—	保留	0	—
2	LPTIMDBGSTOP	Low Power Timer 调试模式停止工作 0: 调试模式计数器仍然工作 1: 调试模式计数器停止工作	0	R/W
1	TIM11DBGSTOP	TIM11 调试模式停止工作 0: 调试模式计数器仍然工作 1: 调试模式计数器停止工作	0	R/W
0	TIM10DBGSTOP	TIM10 调试模式停止工作 0: 调试模式计数器仍然工作 1: 调试模式计数器停止工作	0	R/W

33. SysTick 定时器(SYST)

33.1 SysTick 定时器简介

OS 要想支持多任务，就需要周期执行上下文切换，这样就需要有定时器之类的硬件资源打断程序执行。当定时器中断产生时，处理器就会在异常处理中进行 OS 任务调度，同时还会进行 OS 维护的工作。Cortex-M0+处理器中有一个称为 SysTick 的简单定时器，用于产生周期性的中断请求。

SysTick 为 24 位的定时器，并且向下计数。定时器的计数减到 0 后，就会重新装载一个可编程的数值，并且同时产生 SysTick 中断（中断号为 15），该异常事件会引起 SysTick 中断处理的执行，这个过程是 OS 的一部分。

对于不需要 OS 的系统，SysTick 定时器也可以用作其他用途，比如定时、计时或者为需要周期执行的任务提供中断源。SysTick 中断的产生是可控的，如果中断被禁止，仍然可以用轮询的方法使用 SysTick 定时器，比如检查当前的计数值或者轮询溢出标志。

33.2 设置 SysTick 定时器

由于 SysTick 定时器的重载值和当前值在复位时都是未定义的，为了防止产生异常结果，对 SysTick 的配置需要遵循一定的流程：

- 1.配置 SYST_CSR.ENABLE 为 0，禁止 SysTick。
- 2.配置 SYST_CSR.CLKSOURCE，选择 SysTick 的时钟源。
- 3.配置 SYST_RVR，选择 SysTick 的溢出周期。
- 4.向 SYST_CVR 写入任意值，清零 SYST_CVR 及 SYST_CSR.COUNTFLAG。
- 5.配置 SYST_CSR.TICKINT 为 1，使能 SysTick 中断。
- 6.配置 SYST_CSR.ENABLE 为 1，使能 SysTick，计数开始。
- 7.在中断服务程序中读取 SYST_CSR 以清除溢出标志。

注：SysTick 定时器溢出时间为 SYST_RVR+1 个 SysTick 时钟周期，配置举例：

SysTick 时钟源	SysTick 时钟周期	SYST_RVR	SysTick 定时器溢出时间
内核时钟 HCLK (被设为 4MHz)	0.25us	3999	1ms=0.25us*(3999+1)

33.3 SysTick 定时器寄存器列表

地址	名称	描述	复位值	CMSIS 符号
0xE000E010	SYST_CSR	SysTick 定时器控制和状态寄存器	0x00000000	SysTick->CTRL
0xE000E014	SYST_RVR	SysTick 定时器重载值寄存器	未定义	SysTick->LOAD
0xE000E018	SYST_CVR	SysTick 定时器当前值寄存器	未定义	SysTick->VAL
0xE000E01C	SYST_CALIB	SysTick 定时器校准值寄存器	0x40009C3F	SysTick->CALIB

注：SYST_CALIB 寄存器与 RCC_STICKCR 有对应关系，请参考 7.4.15 节 SysTick Timer 控制寄存器(RCC_STICKCR)。

33.4 SysTick 定时器寄存器说明

33.4.1 SysTick 定时器控制和状态寄存器(SYST_CSR)

位	符号	功能描述	类型	复位值
31:17	保留	—	—	—
16	COUNTFLAG	SysTick 定时器溢出标志 1: SysTick 定时器发生下溢出； 0: SysTick 定时器未发生溢出。 读该寄存器，可清除 COUNTFLAG 标志	RO	0
15:3	保留	—	—	—
2	CLKSOURCE	SysTick 时钟源选择 1: 使用内核时钟 (HCLK) 0: HCLK/4	RW	0
1	TICKINT	SysTick 中断使能 1: 使能中断 0: 禁止中断	RW	0
0	ENABLE	SysTick 定时器使能 1: 使能 SysTick 0: 禁止 SysTick	RW	0

33.4.2 SysTick 定时器重载值寄存器(SYST_RVR)

位	符号	功能描述	类型	复位值
31:24	保留	—	—	—
23:0	RELOAD	SysTick 定时器重载值	RW	未定义

33.4.3 SysTick 定时器当前值寄存器(SYST_CVR)

位	符号	功能描述	类型	复位值
31:24	保留	—	—	—
23:0	CURRENT	读该寄存器，获取 SysTick 定时器的当前计数值； 写任意值到该寄存器，清零该寄存器及 COUNTFLAG。	RW	未定义

33.4.4 SysTick 定时器校准值寄存器(SYST_CALIB)

位	符号	功能描述	类型	复位值
31	NOREF	SysTick 定时器是否使用外部参考时钟 0: HCLK/4 1: 使用内核时钟(HCLK)	RO	0
30	SKEW	10msTENMS 值是否准确 0: 准确 1: 不准确	RO	1
29:24	保留	—	—	—
23:0	TENMS[23:0]	SysTick 10ms 校准值，此值为使用外部参考时钟 HCLK/4(4MHZ)的 10ms 校准值。	RO	0x009C 3F

34. 引脚图

● TSSOP20 引脚配置

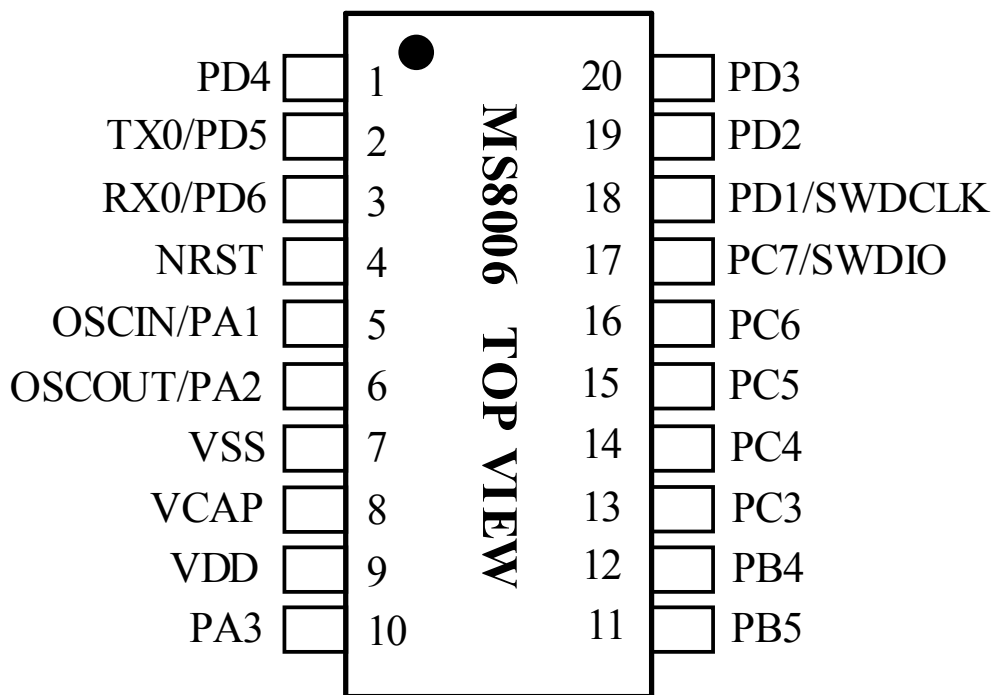


图 34-1. TSSOP20 引脚配置图

35. 引脚描述

表 35.1. 引脚功能说明

引脚名称	引脚#	类型	描述
PD4	1	I/O	PORTPD4
PD5/TX0	2	I/O	功能 1: PORTPD5 功能 2: Uart0TX
PD6/RX0	3	I/O	功能 1: PORTPD6 功能 2: Uart0RX
NRST	4	P	ResetPin, 低电平复位, 内建上拉电阻
PA1/OSCIN	5	I/O	功能 1: PORTPA1 功能 2: 晶振输入 PIN
PA2/OSCOUT	6	I/O	功能 1: PORTPA2 功能 2: 晶振输出 POUT
VSS	7	G	Ground
VCAP	8	P	ICP 模式: 外灌 6.5V; IAP 模式: 6.5V 输出。
VDD	9	P	Power(2.5~5.5V)
PA3	10	I/O	PORTPA3
PB5	11	I/O	PORTPB5
PB4	12	I/O	PORTPB4
PC3	13	I/O	PORTPC3
PC4	14	I/O	PORTPC4
PC5	15	I/O	PORTPC5
PC6	16	I/O	PORTPC6
PC7/SWDIO	17	I/O	功能 1: PORTPC7 功能 2: SWDIO
PD1/SWDCLK	18	I/O	功能 1: PORTPD1 功能 2: SWDCLK
PD2	19	I/O	PORTPD2
PD3	20	I/O	PORTPD3

Type 栏: P 表示电源; G 表示地; I 表示输入; O 表示输出; I/O 表示可输入输出。

其他复用功能具体参照表 10.2. GPIO 口和外设引脚的复用功能映射。

36. 电气特性

36.1 最大绝对额定值

如无特殊说明，所有典型值均基于室温和电源电压 3.3V 测试。

最小值和最大值如下表中定义的工作温度、工作电压、工作频率范围所示。如无特殊说明，所有数据均在此范围内测试。

表 36.1. 工作和贮藏条件

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V _{power}	电源电压		0		5.5	V
V _{iopin}	IO 的电压		-0.3		V _{power} +0.3	V
V _{op}	工作电压		2.5	3.3	5.5	V
T _{stg}	存储温度		-40	25	150	°C
T _{op}	工作温度		-40	25	85	°C
F _{cpu}	CPU 工作频率		32.768K	4M	24M	Hz
V _{ESD,HBM}	参见 36.10					
V _{ESD,CDM}	参见 36.10					
V _{ESD,MM}	参见 36.10					

注意：

1. 温度测试方法：CP 测试高温 85°C，低温 -40°C 和 85°C 的 chip level 测试仅在实验室和 Production Quality Qualification 时测试。

2. 频率测试方法：出厂阶段测试 24MHz 频率，Final Test 只 focus 在封装工艺的缺陷。

36.2 推荐工作条件

表 36.2. 推荐工作条件

参数	符号	条件	额定值		单位	参考
			最小值	最大值		
电源电压	VDD	-	2.5	5.5	V	
VCAP 电容	C _s	-	0.47	2.2	μF	推荐 1.0μF
工作温度	Ta	-	-40	85	°C	

注意：

1.推荐工作条件是确保半导体芯片正常工作的条件。在推荐工作条件的范围内，电气特性的所有规格值均可得到保证。务必在推荐工作条件下使用半导体芯片。超出该条件的使用可能会影响半导体的可靠性。

2.对于本数据手册中未记载的项目、使用条件或逻辑组合的使用，本公司不做任何保障。如果用户考虑在所列条件之外使用本芯片，请事前联系销售代表。

36.3 直流特性

36.3.1 中断使能设置寄存器(NVIC_ISER)

表 36.3. 工作电流特性

Symbol	Parameter	Condition			TYP	MAX	Unit
I _{DD} (RuninRAM)	All Peripherals clock OFF, Run while(1) in RAM	V _{core} =1.2V VDD=3.3V	HIRC Clock source	4M	172	189	μA
				8M	270	297	
				16M	475	523	
				24M	741	815	
I _{DD} (RunMode)	All Peripherals clock ON, Run while(1) in Flash	V _{core} =1.2V VDD=2.5V- 5.5V	HIRC Clock source	4M	394.8	453.4	μA
				8M	720.9	823.4	
				16M	1402.8	1602.7	
				24M	2168.0	2505.0	
	All Peripherals clock OFF, Run while(1) in Flash	V _{core} =1.2V VDD=2.5V- 5.5V	HIRC Clock source	4M	223.0	384.5	μA
				8M	390.2	681.9	
				16M	1106.8	1317.8	
				24M	1694.5	2056.7	
	All Peripherals clock ON, Run while(1) in Flash	V _{core} =1.2V VDD=2.5V- 5.5V	LXT 32.768KHz Clock source (Driver=1)	Ta=- 40to25°C	28	40	μA
				Ta=50°C	30	42	
				Ta=85°C	34	46	
	All Peripherals clock OFF, Run while(1) in Flash	V _{core} =1.2V VDD=2.5V- 5.5V	LXT 32.768KHz Clock source (Driver=1)	Ta=- 40to25°C	27	40	μA
				Ta=50°C	29	42	
				Ta=85°C	33	45	
I _{DD} (SleepMode)	All Peripherals clock ON	V _{core} =1.2V VDD=2.5V- 5.5V	HIRC Clock source	4M	157.65	184.53	μA
				8M	250	286	
				16M	441	499	

Symbol	Parameter	Condition			TYP	MAX	Unit
	All Peripherals clock OFF	$V_{core}=1.2V$ $VDD=2.5V-5.5V$	HIRC Clock source	24M	659	753	μA
				4M	95	116	
				8M	126	150	
				16M	191	222	
	All Peripherals clock ON	$V_{core}=1.2V$ $VDD=2.5V-5.5V$	LXT 32.768KHz Clock source (Driver=1)	Ta=-40to25°C	26	40	μA
				Ta=50°C	29	42	
				Ta=85°C	33	45	
	All Peripherals clock OFF	$V_{core}=1.2V$ $VDD=2.5V-5.5V$	LXT 32.768KHz Clock source (Driver=1)	Ta=-40to25°C	25	40	μA
				Ta=50°C	28	42	
				Ta=85°C	32	45	
I _{DD} (Deep Sleep Mode)	All Peripherals clock OFF,Except RTC,IWDG ,LPTIM,AWK	$V_{core}=1.2V$ $VDD=2.5V-5.5V$		Ta=-40to25°C	1.52	1.67	μA
				Ta=50°C	1.88	2.07	
				Ta=85°C	3.98	4.38	
	All Peripherals clock OFF,except RTC	$V_{core}=1.2V$ $VDD=2.5V-5.5V$	LXT 32.768KHz Clock source (Driver=1)	Ta=-40to25°C	0.98	1.07	μA
				Ta=50°C	1.30	1.43	
				Ta=85°C	3.98	4.38	
	All Peripherals clock OFF,except RTC	$V_{core}=1.2V$ $VDD=2.5V-5.5V$	LIRC 32.768KHz Clock source	Ta=-40to25°C	1.1	1.3	μA
				Ta=50°C	1.5	2.0	
				Ta=85°C	4.1	5.0	
	All Peripherals clock OFF,except IWDG	$V_{core}=1.2V$ $VDD=2.5V-5.5V$		Ta=-40to25°C	0.95	1.04	μA
				Ta=50°C	1.29	1.42	
				Ta=85°C	3.39	3.73	
	All Peripherals clock OFF,except LPTIM	$V_{core}=1.2V$ $VDD=2.5V-5.5V$		Ta=-40to25°C	0.98	1.04	μA
				Ta=50°C	1.33	1.46	
				Ta=85°C	3.43	3.77	
	All Peripherals clock OFF,except AWK	$V_{core}=1.2V$ $VDD=2.5V-5.5V$		Ta=-40to25°C	0.93	1.02	μA
				Ta=50°C	1.29	1.42	
				Ta=85°C	3.39	3.72	
	All Peripherals clock OFF	$V_{core}=1.2V$ $VDD=2.5V-5.5V$		Ta=-40to25°C	0.95	1.04	μA
				Ta=50°C	1.28	1.41	
				Ta=85°C	3.40	3.74	

注意:

1. Guaranteed by characterization results at 85 °C, unless otherwise specified.
- 2.使用 LXT 32.768KHz 时，需要在外部晶振并联一个 10MΩ 电阻。

36.3.2 中断使能设置寄存器(NVIC_ISER)

表 36.4. POR/Brown Out

Symbol	Parameter	Condition	MIN	TYP	MAX	Unit
V _{por}	POR 释放电压(上电过程)BOR 检测电压(掉电过程)		2.2	2.25	2.3	V

36.4 交流特性

36.4.1 输出特性—端口 PA, PB, PC, PD

Symbol	Parameter	Condition	MIN	MAX	Unit
V _{OH}	High level output voltage Source Current	Sourcing 4mA, VDD=3.3V(see Note1)	VDD-0.2		V
		Sourcing 6mA, VDD=3.3V(see Note2)	VDD-0.3		
V _{OL}	Low level output voltage Sink Current	Sinking 4mA, VDD=3.3V(see Note1)		VSS+0.2	V
		Sinking 6mA, VDD=3.3V(see Note2)		VSS+0.3	
V _{OHD}	High level output voltage Double Source Current	Sourcing 8mA, VDD=3.3V(see Note1)	VDD-0.2		V
		Sourcing 12mA, VDD=3.3V(see Note2)	VDD-0.3		
V _{OLD}	Low level output voltage Double Sink Current	Sinking 8mA, VDD=3.3V(see Note1)		VSS+0.2	V
		Sinking 12mA, VDD=3.3V(see Note2)		VSS+0.3	

注意:

1. The maximum total current, I_{OH}(max) and I_{OL}(max), for all outputs combined, should not exceed 40 mA to satisfy the maximum specified voltage drop.
- 2.The maximum total current, I_{OH}(max) and I_{OL}(max), for all output scombined, should not exceed 100mA to satisfy the maximum specified voltage drop.

36.4.2 输入特性—端口 PA, PB, PC, PD

Symbol	Parameter	Conditions	MIN	TYP	MAX	Unit
V _{IT+}	Positive-going input	VDD=2.5	1.4			V

Symbol	Parameter	Conditions	MIN	TYP	MAX	Unit
	threshold voltage	VDD=3.3	1.8			V
		VDD=5.5	3			V
V _{IT-}	Negative-going input threshold voltage	VDD=2.5			0.9	V
		VDD=3.3			1.3	V
		VDD=5.5			2.4	V
V _{hys}	Input voltage hysteresis (V _{IT+} -V _{IT-})	VDD=2.5		0.5		V
		VDD=3.3		0.5		V
		VDD=5.5		0.6		V
R _{pullhigh}	Pullup Resistor	Pullup enable	40	50	60	Kohm
C _{input}	Input Capacitance			5		pf

36.4.3 端口漏电特性—端口 PA, PB, PC, PD

Symbol	Parameter	Conditions	VDD	MAX	Unit
I _{lkg}	Leakage current	V(see Note1,2)	2.5V/3.6V	±50	nA

注意:

1. The leakage current is measured with VSS or VDD applied to the corresponding pin(s), unless otherwise noted.
2. The port pin must be selected as input.

36.4.4 定时器/计数器输入采样要求

Symbol	Parameter	Conditions	MIN	MAX	Unit
T(int)	External interrupt timing	External trigger signal for the interrupt flag(see Note 1)	30		ns
T(cap)	Timer Captuter timing	TIM1/TIM2 capture pulse width Fsystem=4MHz	0.5		μs
f _{EXT}	Timer clock frequency applied to pin	TIM1,TIM2,TIM10,TIM11 external clock input Fsystem=4MHz	0	f _{TIMxCLK} /2	MHz
T(PCA)	PCA clock frequency applied to pin	PCA external clock input Fsystem=4MHz	0	f _{PCACLK} /2	MHz

注意:

1. The external signal sets the interrupt flag every time the minimum t(int) parameters are met. It may be set even with trigger signals shorter than t(int).

36.4.5 内部 HIRC 振荡器

Symbol	Parameter	Conditions	MIN	TYP	MAX	Unit
F _{MCLK}	Internal RC Oscillation frequency		4.0	4.0 8.0 16.0 24.0	24	MHz
T _{Mstart}	Start-up time Not including software calibration	F _{MCLK} =4MHz	2	2.5	5	μs
		F _{MCLK} =8MHz	1.2	1.5	3	μs
		F _{MCLK} =16MHz	1.1	1.4	2.8	μs
		F _{MCLK} =24MHz	1.1	1.4	2.8	μs
I _{MCLK}	Current consumption	F _{MCLK} =4MHz	30	60	120	μA
		F _{MCLK} =8MHz	40	80	160	μA
		F _{MCLK} =16MHz	75	150	300	μA
		F _{MCLK} =24MHz	100	200	400	μA
DC _{MCLK}	duty cycle		45	50	55	%
D _{evM}	Frequency Deviation	VDD=2.5V~5.5V TAMB=-40°C~85°C	-2.5	±1	+2.5	%
		VDD=2.5V~5.5V TAMB=-40°C~50°C	-2.0	±1	+2.0	%

36.4.6 内部 LIRC 振荡器

Symbol	Parameter	Conditions	MIN	TYP	MAX	Unit
F _{ACLK}	Internal RC Oscillation frequency		37.83 32.28	38.4 32.768	38.97 33.26	KHz
T _{Astart}	Start-up time		50	75	150	μs
I _{ACLK}	Current consumption		0.2	0.25	0.35	μA
DC _{ACLK}	duty cycle		45	50	55	%
D _{evA}	Frequency Deviation	VDD=2.5V~5.5V TAMB=-40°C~85°C	-2.0	±1	+2.0	%
		VDD=2.5V~5.5V TAMB=-40°C~50°C	-1.5	±1	+1.5	%

36.4.7 外部 32.768KHz 晶振

Symbol	Parameter	Conditions	MIN	TYP	MAX	Unit
F _{SCLK}	Crystal frequency		32.75	32.768	32.78	KHz
ESR _{SCLK}	Supported crystal equivalent series resistance		40	65	85	KOhm
C _{SCLK}	Supported crystal external	There are 2 CSCLK on		12 ⁽²⁾		pF

Symbol	Parameter	Conditions	MIN	TYP	MAX	Unit
	external load range	2 crystal pins individually				
$I_{dd}^{(1)}$	Current consumption when stable	ESR=65kOhm $C_{SCLK}=12pF$	200	250	350	nA
DC_{SCLK}	duty cycle		40	50	60	%
T_{start}	Start-up time	ESR=65kOhm $C_{SCLK}=12pF$ 40%-60% Duty cycle has been reached		2		s

注意:

1.RCC_LXTCR.LXTDRV=0011, ESR=65K

36.4.8 外部 HXT 晶振

Symbol	Parameter	Conditions	MIN	TYP	MAX	Unit
F_{FCLK}	Crystal frequency		4	16	24	MHz
ESR_{FCLK}	Supported crystal equivalent series resistance		30	60	1500	Ohm
C_{FCLK}	Supported crystal external external load range	There are 2 C_{FCLK} on 2 crystal pins individually		12 ⁽³⁾		pF
$I_{dd}^{(4)}$	Current consumption	24MHz Xtal ESR=30Ohm $C_{FCLK}=12pF$		300		MA
DC_{FCLK}	duty cycle		40	50	60	%
T_{start}	Start-up time	4M~24MHz		250		μs

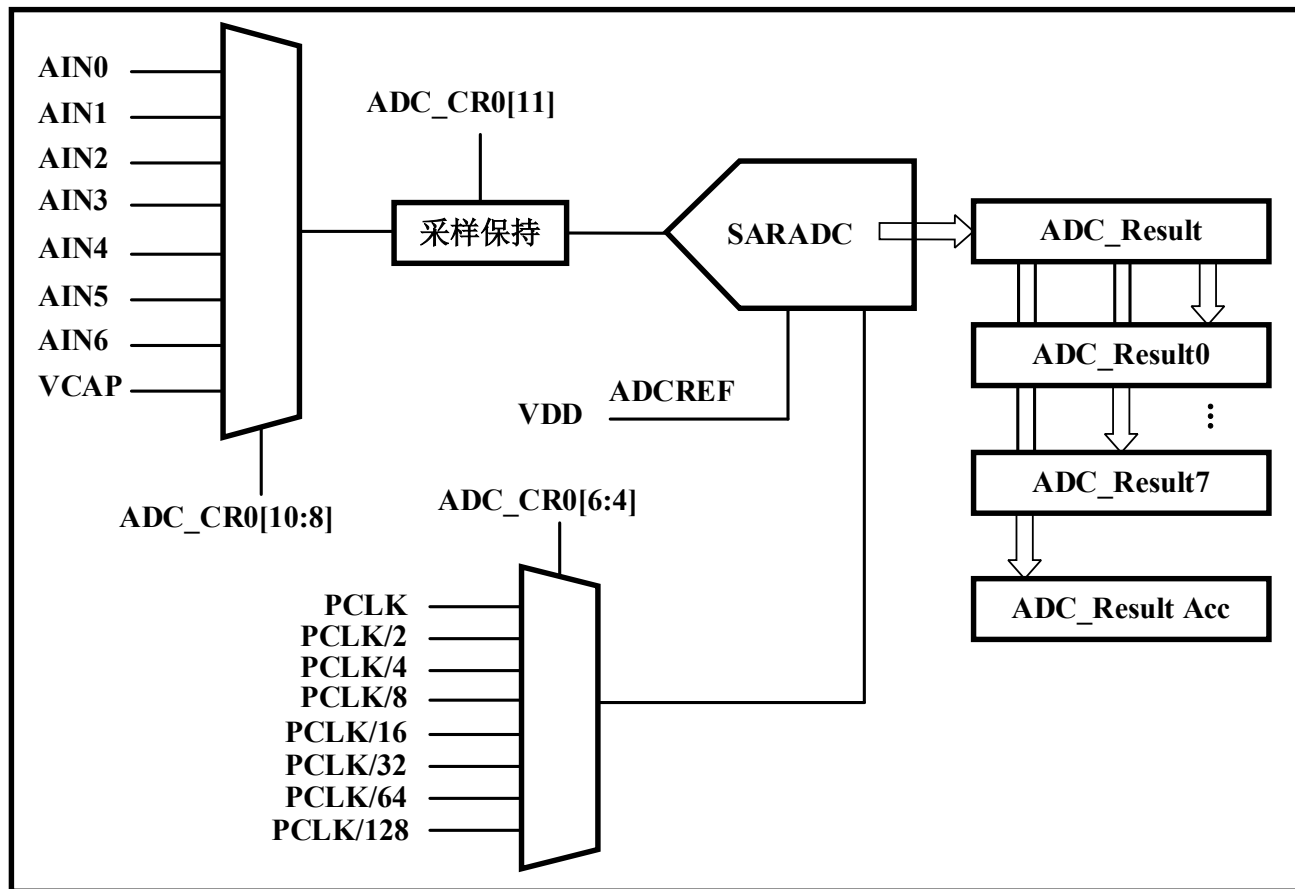
注意:

1.建议使用晶体给出参考值

2.建议使用晶体给出参考值

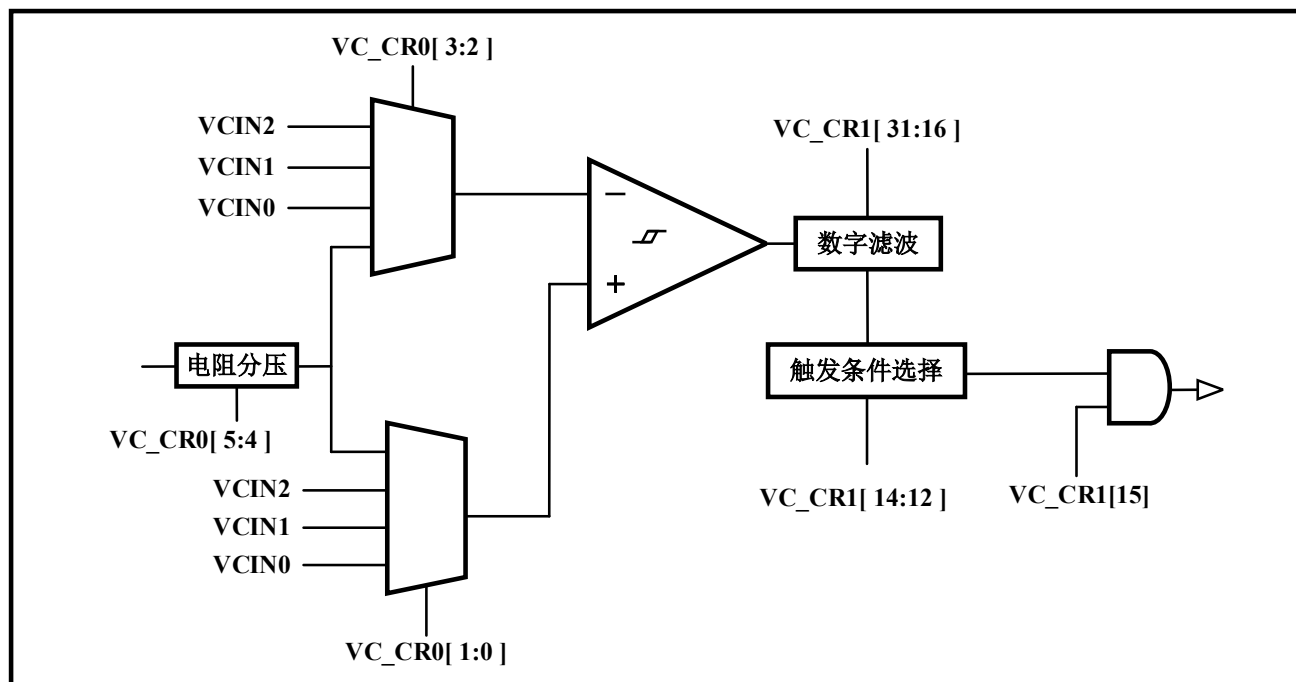
3.Current consumption could vary with oscillating frequency, RCC_HXTCR.HXTDRV=110.

36.5 12 位 A/D 转换器



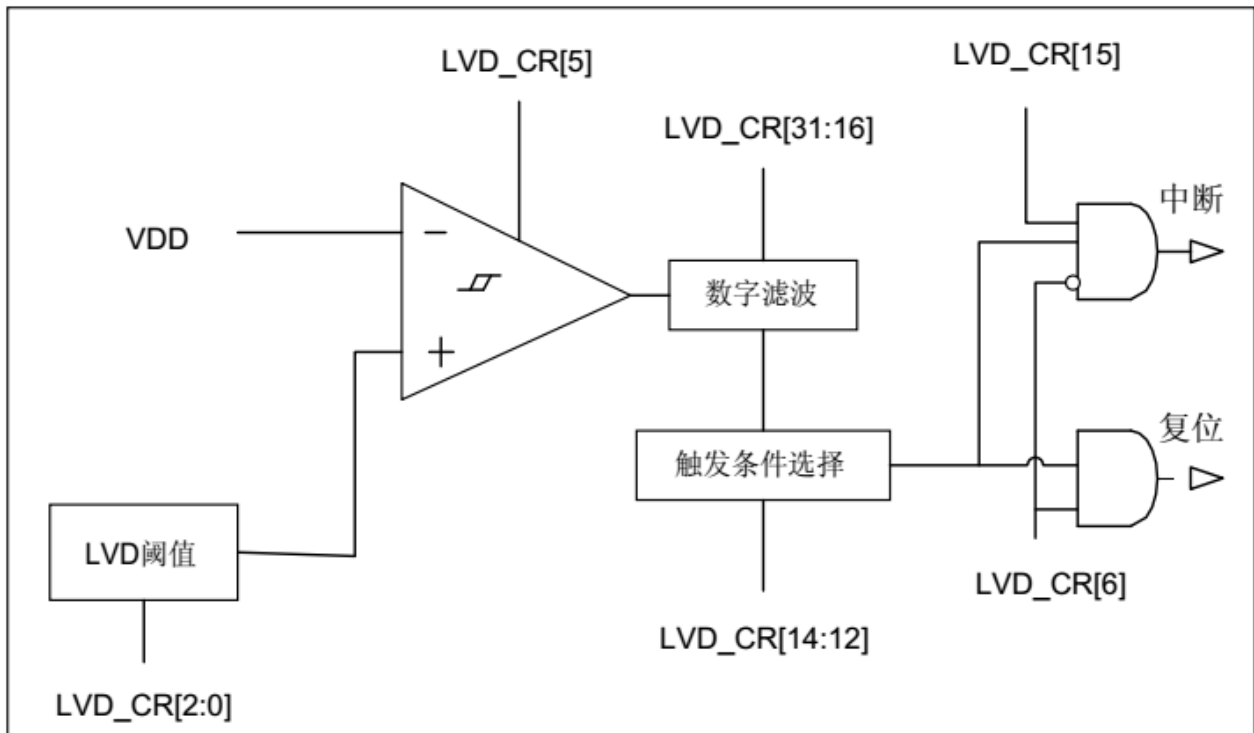
Symbol	Parameter	Conditions	MIN	TYP	MAX	Unit
V_{ADCIN}	Input voltage range	Single ended	0		VDD	V
V_{REF}	ADC reference Voltage			VDD		V
I_{ADC}			0.7	0.9	1.2	mA
C_{ADCIN}	ADC input capacitance		3.5	4	4.5	pF
F_{ADCCLK}	ADC clock Frequency		0.5	4	24	MHz
$T_{ADCSTART}$	Startup time of ADC bias current		2	3	4	μs
T_{ADCONV}	Conversion time		16	16	20	cycles
ENOB			9.5	10	10.4	Bit
DNL	Differential non-linearity		-2	± 1	2	LSB
INL	Integral non-linearity		-3	± 1	3	LSB
E_o	Offset error		-2	± 1	2	LSB
E_g	Gain error		-2	± 1	2	LSB

36.6 模拟电压比较器



Symbol	Parameter	Conditions	MIN	TYP	MAX	Unit
V_{in}	Input voltage range		0		5.5	V
V_{incom}	Input common mode range		0		5.5	V
V_{offset}	Input offset		-10	± 5	+10	mV
I_{comp}	Comparator's current			12		μA
$T_{respoNSe}$	Comparator's response			5		μs

36.7 低电压检查特性



Symbol	Parameter	Conditions	MIN	TYP	MAX	Unit
V_{level}	VDD Detectable threshold	LVD_CR[2:0]=000	Typ-0.1	4.44	Typ+0.1	V
		LVD_CR[2:0]=001		4.0		
		LVD_CR[2:0]=010		3.64		
		LVD_CR[2:0]=011		3.33		
		LVD_CR[2:0]=100		3.08		
		LVD_CR[2:0]=101		2.86		
		LVD_CR[2:0]=110		2.67		
		LVD_CR[2:0]=111		2.5		
I_{comp}	Detector's current		1	1.5	2	μA
$T_{respoNSe}$	Detector's response time when VDD fall below or rise above the threshold		30	50	80	μs
T_{setup}	Detector's setup time when ENABLE.VDD unchanged.		3	5	10	μs

36.8 内存擦/写特性

Symbol	Parameter	Conditions	MIN	TYP	MAX	Unit
ECflash	Sector Endurance		20k			cycles
RETflash	Data Retention		20			Years
Tprog	Byte Program Time		30	45	60	μs
Terase	Sector Erase Time		3.5	3.7	4.5	ms
	Chip Erase Time		20	30	40	ms

36.9 低功耗模式返回时间

Symbol	Parameter	Conditions	MIN	TYP	MAX	Unit
Twakeup	Deep sleep mode to Active mode	4M	2.0	2.5	3.5	μs
		8M	1.4	1.6	2.5	
		16M	1.3	1.4	2.0	
		24M	1.2	1.3	1.8	

36.10 电磁敏感特性

36.10.1 ESD

Symbol	Parameter	Conditions	MIN	TYP	MAX	Unit
VESDHBM	ESD@Human Body Mode		8			KV
VESDCDM	ESD@Charge Device Mode		1.5			KV
VESDMM	ESD@machine Mode		400			V
Ilatchup	Latch up current		100			mA

36.10.2 静态栓锁(Static Latch-up)

为了评估栓锁性能，需要在 6 个样品上进行 2 个互补的静态栓锁测试：

- 为每个电源引脚，提供超过极限的供电电压。
- 在每个输入、输出和可配置的 I/O 引脚上注入电流。

这个测试符合 EIA/JESD78A 集成电路栓锁标准。

符号	参数	条件	类型
LU	Static latch-up class	TA=+85°C conforming to JESD78A	II Level A

37. 封装信息

■ TSSOP20 封装外形尺寸

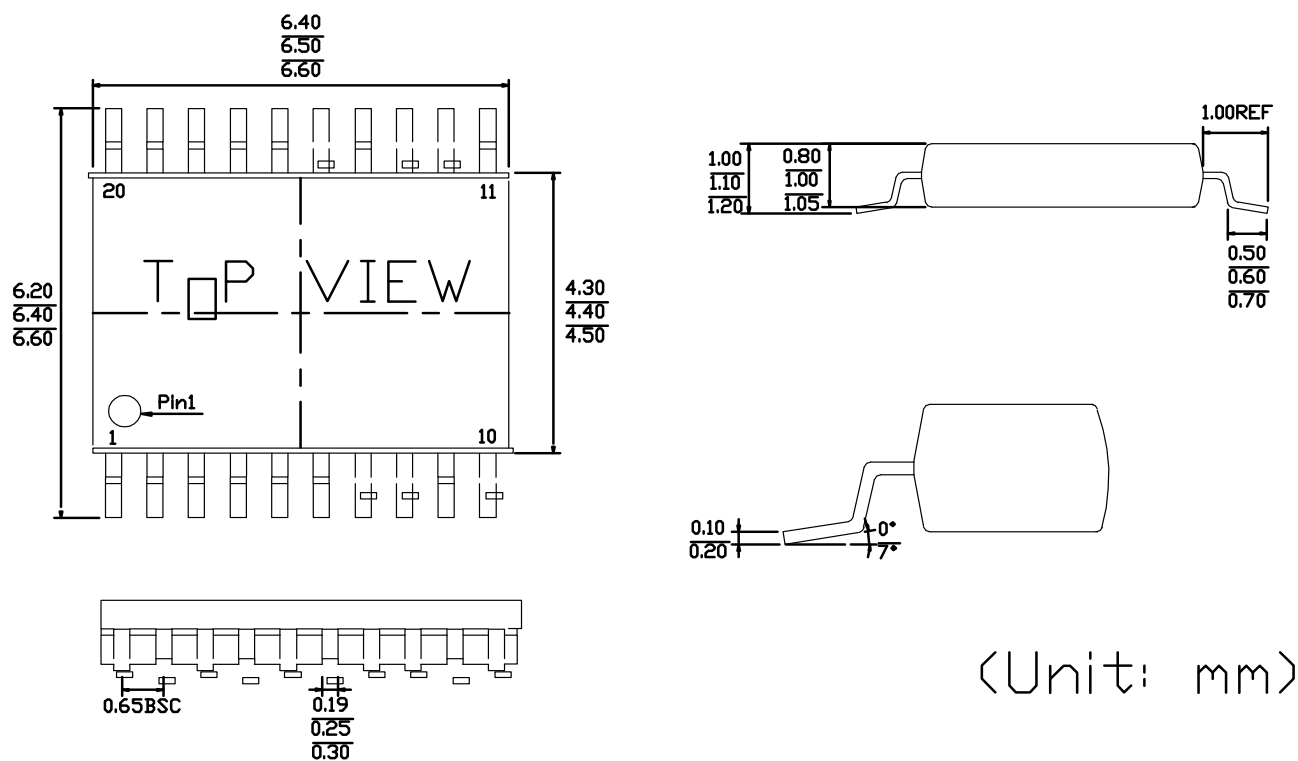


图 37-1. TSSOP20 封装框图

38. 版本记录

日期	版本	作者	备注
2020-12	V1.0	Yangyang, Wang	初始版本