



# HC32F002 系列

32 位 ARM<sup>®</sup> Cortex<sup>®</sup>-M0+ 微控制器

## 用户手册

Beta 版本，仅供参考

## 声 明

- 华大半导体有限公司（以下简称：“HDSC”）保留随时更改、更正、增强、修改华大半导体产品和/或本文档的权利，恕不另行通知。用户可在下单前获取最新相关信息。HDSC 产品依据购销基本合同中载明的销售条款和条件进行销售。
- 用户对 HDSC 产品的选择和使用承担全部责任，用户将 HDSC 产品用于其自己或指定第三方产品上的，HDSC 不提供服务支持且不对此类产品承担任何责任。
- HDSC 在此确认未以明示或暗示方式授予任何知识产权许可。
- HDSC 产品的转售，若其条款与此处规定不同，HDSC 对此类产品的任何保修承诺无效。
- 任何带有“®”或“™”标识的图形或字样是 HDSC 的商标。所有其他在 HDSC 产品上显示的产品或服务名称均为其各自所有者的财产。
- 本通知中的信息取代并替换先前版本中的信息。

©2019 华大半导体有限公司 - 保留所有权利

## 目 录

声 明 .....	2
目 录 .....	3
产品特性 低功耗 MCU .....	22
1 功能模块 .....	23
1.1 功能模块框图 .....	23
1.2 32 位 Cortex M0+ 内核 .....	24
1.3 存储器 .....	24
1.3.1 18 Kbytes FLASH .....	24
1.3.2 2 Kbytes RAM .....	24
1.4 时钟系统 .....	24
1.5 工作模式 .....	25
1.6 端口控制器 GPIO .....	25
1.7 中断控制器 NVIC .....	25
1.8 复位控制器 RESET .....	25
1.9 定时器 TIM .....	25
1.10 看门狗 WDT .....	26
1.11 低功耗同步异步收发器 LPUART .....	26
1.12 串行外设接口 SPI .....	27
1.13 I2C 总线 .....	27
1.14 时钟校准器 CTRIM .....	27
1.15 蜂鸣器 Buzzer .....	28
1.16 模数转换器 ADC .....	28
1.17 低电压检测器 LVD .....	28
1.18 嵌入式调试系统 .....	29
1.19 编程模式 .....	29
1.20 器件电子签名 .....	29
1.21 高安全性 .....	29
2 引脚配置及功能 .....	30
2.1 引脚配置图 .....	30
2.2 引脚功能说明 .....	33
2.3 模块信号说明 .....	35
3 系统结构 .....	36
3.1 概述 .....	36
3.2 存储器和模块地址分配 .....	37
4 工作模式 .....	38
4.1 工作模式概述 .....	38
4.2 工作模式切换 .....	38
4.3 运行模式 (Active Mode) .....	41
4.4 休眠模式 (Sleep Mode) .....	41
4.5 深度休眠模式 (Deep Sleep Mode) .....	41

4.6	功能模块工作状态.....	42
4.7	寄存器.....	43
4.7.1	系统控制寄存器 (SCB_SCR) .....	43
5	系统控制器 (SYSCTRL) .....	44
5.1	系统时钟介绍.....	44
5.1.1	时钟架构图.....	45
5.1.2	内部高速 RC 时钟 RCH.....	45
5.1.3	内部低速 RC 时钟 RCL.....	46
5.1.4	外部输入时钟 EXTCLK .....	46
5.2	系统时钟切换.....	47
5.2.1	标准的时钟切换流程 .....	47
5.2.2	RCH 不同输出频率间切换流程 .....	48
5.2.3	从其它时钟切换到 RCL 示例 .....	48
5.2.4	从其它时钟切换到 RCHCLKD 示例 .....	49
5.3	片内外设时钟控制.....	50
5.4	中断唤醒控制 .....	51
5.4.1	从深度休眠模式唤醒后执行中断服务程序的方法 .....	51
5.4.2	从深度休眠模式唤醒后不执行中断服务程序的方法 .....	51
5.4.3	退出休眠 .....	52
5.5	寄存器 .....	54
5.5.1	系统控制寄存器 0 (CR0) .....	55
5.5.2	系统控制寄存器 1 (CR1) .....	56
5.5.3	系统控制寄存器 2 (CR2) .....	57
5.5.4	系统控制寄存器 3 (CR3) .....	58
5.5.5	RCH 控制寄存器 (RCH) .....	59
5.5.6	RCL 控制寄存器 (RCL) .....	60
5.5.7	片内外设时钟控制寄存器 0 (PeriClkEn0) .....	61
5.5.8	片内外设时钟控制寄存器 1 (PeriClkEn1) .....	63
5.5.9	调试模式模块工作状态控制寄存器 (DebugActive) .....	64
6	复位控制器 (RESET) .....	65
6.1	复位控制器介绍.....	65
6.1.1	上电下电复位 POR .....	65
6.1.2	外部复位管脚复位 .....	66
6.1.3	WDT 复位 .....	66
6.1.4	LVD 低电压复位 .....	66
6.1.5	Cortex-M0+ SYSRESETREQ 复位 .....	66
6.1.6	Cortex-M0+ LOCKUP 复位 .....	66
6.2	寄存器.....	67
6.2.1	复位标识寄存器 (Reset_Flag) .....	67
6.2.2	片内外设复位控制寄存器 0 (PeriReset0) .....	69
6.2.3	片内外设复位控制寄存器 1 (PeriReset1) .....	71
7	中断控制器 (NVIC) .....	72

7.1	概述	72
7.2	中断优先级	73
7.3	中断向量表	74
7.4	中断输入和挂起行为	75
7.5	中断等待	78
7.6	中断源	79
7.7	中断结构图	80
7.8	软件基本操作	82
7.8.1	外部中断使能	82
7.8.2	NVIC 中断使能和清除使能	82
7.8.3	NVIC 中断挂起和清除挂起	82
7.8.4	NVIC 中断优先级	82
7.8.5	NVIC 中断屏蔽	83
7.9	寄存器	84
7.9.1	中断使能设置寄存器 (NVIC_ISER)	84
7.9.2	中断使能清除寄存器 (NVIC_ICER)	85
7.9.3	中断挂起状态设置寄存器 (NVIC_ISPR)	85
7.9.4	中断挂起状态清除寄存器 (NVIC_ICPR)	86
7.9.5	中断优先级寄存器 (NVIC_IPR0)	87
7.9.6	中断优先级寄存器 (NVIC_IPR1)	88
7.9.7	中断优先级寄存器 (NVIC_IPR2)	89
7.9.8	中断优先级寄存器 (NVIC_IPR3)	90
7.9.9	中断优先级寄存器 (NVIC_IPR4)	91
7.9.10	中断优先级寄存器 (NVIC_IPR5)	92
7.9.11	中断优先级寄存器 (NVIC_IPR6)	93
7.9.12	中断优先级寄存器 (NVIC_IPR7)	94
7.9.13	中断屏蔽特殊寄存器 (PRIMASK)	95
8	通用输入输出端口控制器 (GPIO)	96
8.1	简介	96
8.2	主要特性	96
8.3	功能描述	97
8.3.1	端口电路框图	97
8.3.2	端口模式配置	98
8.3.3	端口复位状态	99
8.3.4	端口模拟功能	99
8.3.5	端口通用输入输出功能	100
8.3.5.1	输入模式	101
8.3.5.2	输出模式	102
8.3.6	端口复用功能	103
8.3.7	端口时钟输出	105
8.3.7.1	输出 HCLK	105
8.3.7.2	输出 TCLK	105

8.3.8 端口外部中断 .....	106
8.4 寄存器描述 .....	107
8.4.1 寄存器列表 .....	107
8.4.2 端口数模配置寄存器 (GPIOx_ADS) .....	109
8.4.3 端口方向配置寄存器(GPIOx_DIR) .....	110
8.4.4 端口输出类型寄存器 (GPIOx_OpenDrain) .....	111
8.4.5 端口上拉寄存器 (GPIOx_PU) .....	111
8.4.6 端口数据输入寄存器 (GPIOx_IN) .....	112
8.4.7 端口数据输出寄存器 (GPIOx_OUT) .....	112
8.4.8 端口复位寄存器 (GPIOx_BRR) .....	113
8.4.9 端口置位复位寄存器 (GPIOx_BSRR) .....	114
8.4.10 端口复用功能低位寄存器 (GPIOx_AFRL) .....	115
8.4.11 端口高电平中断使能配置寄存器(GPIOx_HIGHIE) .....	116
8.4.12 端口低电平中断使能配置寄存器(GPIOx_LOWIE) .....	116
8.4.13 端口上升沿中断使能配置寄存器(GPIOx_RISEIE) .....	117
8.4.14 端口下降沿中断使能配置寄存器(GPIOx_FALLIE) .....	117
8.4.15 端口中断状态寄存器(GPIOx_IFR) .....	118
8.4.16 端口中断清除寄存器(GPIOx_ICR) .....	119
8.4.17 端口辅助功能配置寄存器 1(GPIOx_CR1) .....	120
8.4.18 端口辅助功能配置寄存器 4 (GPIO_CR4) .....	121
9 FLASH 控制器 (FLASH) .....	122
9.1 概述 .....	122
9.2 容量划分 .....	122
9.3 读等待周期 .....	122
9.4 FLASH 操作 (读、写、擦) .....	123
9.4.1 页擦除 (Sector Erase) .....	123
9.4.2 全片擦除 (Chip Erase) .....	123
9.4.3 写操作 (Program) .....	124
9.4.4 读操作 (Read) .....	126
9.5 FLASH 安全保护 .....	127
9.5.1 页面擦写保护 .....	127
9.5.2 PC 地址擦写保护 .....	127
9.5.3 寄存器写保护 .....	127
9.5.4 数据读出保护 .....	128
9.6 寄存器描述 .....	129
9.6.1 控制寄存器列表 .....	129
9.6.2 控制寄存器 (FLASH_CR) .....	130
9.6.3 中断标志寄存器 (FLASH_IFR) .....	131
9.6.4 中断标志清除寄存器 (FLASH_ICLR) .....	131
9.6.5 序列寄存器 (FLASH_BYPASS) .....	132
9.6.6 擦写保护寄存器 (FLASH_SLOCK) .....	132
9.6.7 读等待周期寄存器 (FLASH_WAIT) .....	133

9.6.8 读保护状态寄存器 (FLASH_LockState) .....	133
10 RAM 控制器 (RAM) .....	134
10.1 概述.....	134
10.2 功能描述.....	134
10.2.1 RAM 地址范围.....	134
10.2.2 读写位宽 .....	134
11 基本定时器 (BTIM) .....	135
11.1 概述.....	135
11.2 主要特性.....	135
11.3 功能描述.....	136
11.3.1 功能框图.....	136
11.3.2 滤波单元.....	136
11.3.3 计数单元.....	136
11.3.4 定时器模式.....	137
11.3.5 计数器模式.....	138
11.3.6 触发启动模式.....	138
11.3.7 门控模式.....	139
11.3.8 定时器级联.....	140
11.4 寄存器描述.....	141
11.4.1 重载寄存器 (BTIMx_ARR) (x=3,4,5).....	142
11.4.2 计数寄存器 (BTIMx_CNT) (x=3,4,5).....	142
11.4.3 控制寄存器 (BTIMx_CR) (x=3,4,5).....	143
11.4.4 中断使能 (BTIMx_IER) (x=3,4,5) .....	144
11.4.5 中断标志寄存器 (BTIMx_IFR) (x=3,4,5) .....	144
11.4.6 中断标志清除寄存器 (BTIMx_ICR) (x=3,4,5) .....	145
11.4.7 复合中断标志寄存器 (BTIM345_AIFR) .....	146
11.4.8 复合中断标志清除寄存器 (BTIM345_AICR) .....	147
12 通用定时器 (GTIM) .....	148
12.1 概述.....	148
12.2 主要特性.....	148
12.3 功能描述.....	149
12.3.1 功能框图 .....	149
12.3.2 滤波单元 .....	149
12.3.3 计数单元 .....	150
12.3.4 定时器模式 .....	151
12.3.5 计数器模式 .....	151
12.3.6 触发启动模式.....	151
12.3.7 门控模式 .....	152
12.3.8 比较捕获功能.....	153
12.3.8.1 捕获功能.....	153
12.3.8.2 比较功能.....	154
12.3.9 定时器级联 .....	155

12.3.10 片内外设互联.....	155
12.4 寄存器描述.....	156
12.4.1 重载寄存器 (GTIM_ARR) .....	157
12.4.2 计数寄存器 (GTIM_CNT) .....	157
12.4.3 控制寄存器 1 (GTIM_CR1) .....	158
12.4.4 控制寄存器 0 (GTIM_CR0) .....	159
12.4.5 中断使能控制寄存器 (GTIM_IER) .....	160
12.4.6 中断标志寄存器 (GTIM_IFR) .....	161
12.4.7 中断标志清除寄存器 (GTIM_ICR) .....	162
12.4.8 比较捕获控制寄存器 (GTIM_CMMR) .....	163
12.4.9 比较捕获寄存器 (GTIM_CCRy) (y=0,1,2,3).....	164
13 高级定时器 (ATIM) .....	165
13.1 概述.....	165
13.2 主要特性.....	165
13.3 功能描述.....	167
13.3.1 定时器时钟 .....	167
13.3.2 定时计数器 .....	167
13.3.3 定时器预分频.....	167
13.3.4 模式 0 计数定时器功能.....	168
13.3.4.1 功能框图.....	168
13.3.4.2 计数波形.....	169
13.3.4.3 计数功能.....	170
13.3.4.4 定时功能.....	170
13.3.4.5 时序图.....	170
13.3.4.6 Buzzer 功能.....	171
13.3.4.7 设置示例.....	171
13.3.5 模式 1 脉宽测量 PWC.....	172
13.3.5.1 PWC 功能框图.....	172
13.3.5.2 PWC 波形测量时序图.....	173
13.3.5.3 PWC 单次触发模式.....	175
13.3.5.1 设置示例.....	175
13.3.6 模式 2/3 比较捕获模式 .....	177
13.3.6.1 计数器.....	177
13.3.6.2 计数器波形.....	178
13.3.6.3 重复计数.....	181
13.3.6.4 数据缓存.....	183
13.3.6.5 比较输出 OCREF.....	186
13.3.6.6 独立 PWM 输出 .....	189
13.3.6.7 互补 PWM 输出 .....	190
13.3.6.8 有死区的 PWM 输出 .....	191
13.3.6.9 单脉冲输出.....	192
13.3.6.10 比较中断.....	193

13.3.6.11 捕获输入.....	194
13.3.6.12 设置示例.....	197
13.3.7 模式 2/3 从模式 .....	200
13.3.7.1 门控计数.....	200
13.3.7.2 触发功能.....	201
13.3.7.3 复位计数.....	201
13.3.8 正交编码计数功能.....	202
13.3.9 Timer 触发 ADC .....	204
13.3.10 刹车控制 .....	205
13.3.11 定时器互联.....	205
13.3.12 CH0B 捕获输入互联 .....	205
13.4 寄存器描述.....	206
13.4.1 模式 0 寄存器描述.....	207
13.4.1.1 16 位模式重载寄存器 (ATIMx_ARR) .....	207
13.4.1.2 16 位模式计数寄存器 (ATIMx_CNT) .....	207
13.4.1.3 32 位模式计数寄存器 (ATIMx_CNT32) .....	208
13.4.1.4 控制寄存器 (ATIMx_M0CR) .....	209
13.4.1.5 中断标志寄存器 (ATIMx_IFR) .....	211
13.4.1.6 中断标志清除寄存器 (ATIMx_ICLR) .....	211
13.4.1.7 死区时间寄存器 (ATIMx_DTR) .....	212
13.4.2 模式 1 寄存器描述.....	213
13.4.2.1 16 位模式计数寄存器 (ATIMx_CNT) .....	213
13.4.2.2 控制寄存器 (ATIMx_M1CR) .....	214
13.4.2.3 中断标志寄存器 (ATIMx_IFR) .....	216
13.4.2.4 中断标志清除寄存器 (ATIMx_ICLR) .....	216
13.4.2.5 主从模式控制寄存器 (ATIMx_MSCR) .....	217
13.4.2.6 输出控制滤波 (ATIMx_FLTR) .....	218
13.4.2.7 控制寄存器 (ATIMx_CR0) .....	219
13.4.2.8 比较捕获寄存器 (ATIMx_CCR0A) .....	219
13.4.3 模式 2,3 寄存器描述.....	220
13.4.3.1 16 位模式重载寄存器 (ATIMx_ARR) .....	220
13.4.3.2 16 位模式计数寄存器 (ATIMx_CNT) .....	220
13.4.3.3 控制寄存器 (ATIMx_M23CR) .....	221
13.4.3.4 中断标志寄存器 (ATIMx_IFR) .....	224
13.4.3.5 中断标志清除寄存器 (ATIMx_ICLR) .....	226
13.4.3.6 主从模式控制寄存器 (ATIMx_MSCR) .....	227
13.4.3.7 输出控制/输入滤波 (ATIMx_FLTR) .....	229
13.4.3.8 ADC 触发控制寄存器 (ATIMx_ADTR) .....	232
13.4.3.9 通道 0 控制寄存器 (ATIMx_CRCH0) .....	233
13.4.3.10 通道 1/2 控制寄存器 (TIM3_CRCH1/2) .....	235
13.4.3.11 死区时间寄存器 (ATIMx_DTR) .....	237
13.4.3.12 重复周期设置值寄存器 (ATIMx_RCR) .....	238

13.4.3.13 通道 0 比较捕获寄存器 (ATIMx_CCR0A/B).....	238
13.4.3.14 通道 1/2 比较捕获寄存器 (TIM3_CCR1/2 A/B) .....	239
13.4.3.15 通道 3 比较捕获寄存器 (TIM3_CCR3).....	239
13.4.3.16 通道 3 控制寄存器 (ATIM3_CRCH3) .....	240
14 时钟校准模块 (CTRIM).....	241
14.1    概述.....	241
14.2    主要特性.....	241
14.3    功能描述.....	241
14.3.1 RCH 时钟校准模式.....	241
14.3.2 RCL 时钟校准模式 .....	242
14.3.3 低功耗定时器模式.....	243
14.4    软件操作流程.....	244
14.4.1 RCH 时钟自动校准示例.....	244
14.4.2 RCL 时钟自动校准示例 .....	244
14.5    寄存器描述.....	246
14.5.1 自动装载寄存器 (CTRIM_ARR) .....	246
14.5.2 误差计数器寄存器 (CTRIM_CNT) .....	247
14.5.3 控制寄存器 0 (CTRIM_CR0) .....	248
14.5.4 控制寄存器 1 (CTRIM_CR1) .....	249
14.5.5 中断使能寄存器 (CTRIM_IER) .....	250
14.5.6 中断和状态寄存器 (CTRIM_ISR) .....	251
14.5.7 中断状态清除寄存器 (CTRIM_ICR) .....	252
14.5.8 误差结果寄存器 (CTRIM_FCAP) .....	252
14.5.9 TrimCode 寄存器 (CTRIM_TVAL) .....	253
14.5.10 误差上限寄存器 (CTRIM_FLIM) .....	253
15 独立看门狗 (IWDT) .....	254
15.1    概述.....	254
15.2    主要特性.....	254
15.3    功能描述.....	255
15.3.1 功能框图 .....	255
15.3.2 窗口看门狗模式 .....	255
15.3.3 独立看门狗模式 .....	255
15.3.4 看门狗溢出处理 .....	255
15.3.5 看门狗暂停 .....	256
15.3.6 看门狗停止及再次运行.....	256
15.3.7 寄存器写保护.....	256
15.4    编程示例.....	257
15.4.1 配置 IWDT 为窗口看门狗 .....	257
15.4.2 配置 IWDT 为独立看门狗 .....	257
15.5    寄存器描述.....	258
15.5.1 键值寄存器 (IWDT_KR) .....	258
15.5.2 控制寄存器 (IWDT_CR) .....	259

15.5.3 重载寄存器 (IWDT_ARR) .....	260
15.5.4 窗口寄存器 (IWDT_WINR) .....	261
15.5.5 状态寄存器 (IWDT_SR) .....	262
16 窗口看门狗 (WWDT) .....	263
16.1 概述.....	263
16.2 主要特性.....	263
16.3 功能描述.....	264
16.3.1 功能框图 .....	264
16.3.2 使能看门狗 .....	264
16.3.3 配置看门狗超时.....	264
16.3.4 更新递减计数器.....	264
16.3.5 预溢出中断 .....	265
16.4 编程示例.....	265
16.5 寄存器描述.....	266
16.5.1 控制寄存器 0 (WWDT_CR0) .....	266
16.5.2 控制寄存器 1 (WWDT_CR1) .....	267
16.5.3 配置寄存器 (WWDT_SR) .....	268
17 低功耗同步异步收发器 (LPUART) .....	269
17.1 概述.....	269
17.2 主要特性.....	269
17.3 功能描述.....	270
17.3.1 功能框图 .....	270
17.3.2 时钟说明 .....	270
17.3.3 工作模式 .....	271
17.3.3.1 四种工作模式对比.....	271
17.3.3.2 Mode0 数据收发说明 .....	273
17.3.3.3 Mode1 数据收发说明 .....	273
17.3.3.4 Mode2/Mode3 数据收发说明 .....	274
17.3.3.5 单总线数据收发说明.....	275
17.3.4 波特率生成 .....	276
17.3.4.1 传输时钟为 4MHz 波特率设置示例 (Mode1 / Mode3) .....	276
17.3.4.2 传输时钟为 8MHz 波特率设置示例 (Mode1 / Mode3) .....	277
17.3.4.3 传输时钟为 16MHz 波特率设置示例 (Mode1 / Mode3) .....	277
17.3.4.4 传输时钟为 24MHz 波特率设置示例 (Mode1 / Mode3) .....	278
17.3.4.5 传输时钟为 32MHz 波特率设置示例 (Mode1 / Mode3) .....	278
17.3.4.6 传输时钟为 48MHz 波特率设置示例 (Mode1 / Mode3) .....	279
17.3.5 帧错误检测 .....	280
17.3.6 多机通讯 .....	280
17.3.6.1 主机发送地址帧及数据帧.....	280
17.3.6.2 从机接收地址帧及数据帧.....	280
17.3.6.3 从机地址配置.....	280
17.3.7 硬件流控 .....	281

17.3.7.1 RTS 流控 .....	281
17.3.7.2 CTS 流控 .....	281
17.4 编程示例.....	283
17.4.1 发送数据示例.....	283
17.4.2 接收数据示例.....	283
17.5 寄存器描述.....	285
17.5.1 数据寄存器 (LPUARTx_DR) .....	285
17.5.2 控制寄存器 (LPUARTx_CR) .....	286
17.5.3 地址寄存器 (LPUARTx_ADDR) .....	288
17.5.4 地址掩码寄存器 (LPUARTx_AddrMask) .....	288
17.5.5 标志位寄存器 (LPUARTx_ISR) .....	289
17.5.6 标志位清除寄存器 (LPUARTx_ICR) .....	290
17.5.7 波特率寄存器 (LPUARTx_BRR) .....	291
18 串行外设接口 (SPI) .....	292
18.1 概述.....	292
18.2 主要特性.....	292
18.3 功能描述.....	293
18.3.1 功能框图 .....	293
18.3.2 通信格式和时序 .....	293
18.3.3 从机选择管脚配置.....	295
18.3.4 全双工通信 .....	296
18.3.5 单线半双工通信 .....	297
18.3.6 单工通信 .....	298
18.3.7 多机通信 .....	299
18.3.8 状态标志和错误标志.....	300
18.3.9 中断的产生和清除.....	303
18.4 编程示例.....	304
18.4.1 主机全双工收发数据示例 .....	304
18.4.2 从机全双工收发数据示例 .....	305
18.4.3 单线半双工发送接收数据示例 .....	306
18.6 寄存器描述.....	307
18.6.1 SPI 控制寄存器 0(SPI_CR0).....	308
18.6.2 SPI 控制寄存器 1(SPI_CR1).....	310
18.6.3 SPI 单线半双工输出使能寄存器(SPI_HDOE).....	312
18.6.4 SPI 内部从机选择寄存器(SPI_SSI) .....	313
18.6.5 SPI 状态寄存器(SPI_SR) .....	314
18.6.6 SPI 中断清除寄存器(SPI_ICR) .....	316
18.6.7 SPI 数据寄存器(SPI_DR) .....	317
19 I2C 总线 (I2C) .....	318
19.1 概述.....	318
19.2 主要特性.....	318
19.3 协议描述.....	318

19.3.1 I2C 总线上数据传输 .....	319
19.3.2 I2C 总线上的应答 .....	320
19.3.3 I2C 总线上的仲裁 .....	321
19.4 功能描述 .....	322
19.4.1 功能框图 .....	322
19.4.2 串行时钟发生器 .....	323
19.4.3 输入滤波器 .....	323
19.4.4 地址比较器 .....	323
19.4.5 应答标志位 .....	324
19.4.6 中断产生器 .....	324
19.4.7 工作模式 .....	324
19.4.8 状态码表述 .....	330
19.5 编程示例 .....	333
19.5.1 主机发送示例 .....	333
19.5.2 主机接收示例 .....	334
19.5.3 从机接收示例 .....	335
19.5.4 从机发送示例 .....	336
19.6 寄存器描述 .....	337
19.6.1 寄存器列表 .....	337
19.6.2 波特率发生器使能寄存器(I2C_BRREN) .....	337
19.6.3 波特率计数器配置寄存器(I2C_BRR) .....	338
19.6.4 配置寄存器(I2C_CR) .....	339
19.6.5 数据寄存器(I2C_DR) .....	340
19.6.6 状态寄存器(I2C_STAT) .....	340
19.6.7 从机地址 0 寄存器(I2C_ADDR0) .....	341
19.6.8 从机地址 1 寄存器(I2C_ADDR1) .....	342
19.6.9 从机地址 2 寄存器(I2C_ADDR2) .....	342
19.6.10 从机地址匹配寄存器(I2C_MATCH) .....	343
20 模数转换器 (ADC) .....	344
20.1 模块简介 .....	344
20.2 ADC 框图 .....	345
20.3 转换时序及转换速度 .....	346
20.4 单次转换模式 .....	347
20.5 外部触发转换模式 .....	348
20.6 持续转换模式 .....	349
20.7 测量 AVCC 电压 .....	350
20.8 ADC 中断 .....	351
20.9 ADC 模块寄存器 .....	352
20.9.1 ADC 配置寄存器 0 (ADC_CR0) .....	353
20.9.2 ADC 配置寄存器 1 (ADC_CR1) .....	354
20.9.3 ADC 转换结果寄存器 (ADC_Result) .....	355
20.9.4 ADC 中断标志寄存器 (ADC_IFR) .....	355

20.9.5 ADC 中断使能寄存器 (ADC_IER) .....	356
20.9.6 ADC 标志清除寄存器 (ADC_ICR) .....	356
20.9.7 ADC 外部触发配置寄存器 (ADC_ExtTrigger) .....	357
20.9.8 ADC 转换启动控制寄存器 (ADC_Start) .....	358
20.9.9 ADC 持续转换控制寄存器 (ADC_AllStart) .....	358
21 低电压检测器 (LVD) .....	359
21.1 LVD 简介 .....	359
21.2 LVD 框图 .....	359
21.3 迟滞功能 .....	360
21.4 数字滤波 .....	360
21.5 配置示例 .....	361
21.5.1 LVD 配置为低电压复位 .....	361
21.5.2 LVD 配置为电压变化中断 .....	362
21.6 LVD 寄存器 .....	363
21.6.1 LVD 配置寄存器 (LVD_CR) .....	364
21.6.2 LVD 中断寄存器 (LVD_IFR) .....	366
22 器件电子签名 .....	367
22.1 产品唯一身份标识码 (UID) 寄存器 .....	367
22.2 产品型号寄存器 .....	368
22.3 FLASH 容量寄存器 .....	368
22.4 RAM 容量寄存器 .....	369
22.5 管脚数量寄存器 .....	369
22.6 一次性编程区域 (OTP 区域) .....	370
23 SWD 调试接口 .....	371
23.1 SWD 调试附加功能 .....	371
23.2 ARM® 参考文档 .....	372
23.3 调试端口管脚 .....	373
23.3.1 SWD 端口管脚 .....	373
23.3.2 SW-DP 管脚分配 .....	373
23.3.3 SWD 管脚上的内部上拉 .....	373
23.4 SWD 端口 .....	374
23.4.1 SWD 协议简介 .....	374
23.4.2 SWD 协议序列 .....	374
23.4.3 SW-DP 状态机 (复位、空闲状态、ID 代码) .....	375
23.4.4 DP 和 AP 读/写访问 .....	375
23.4.5 SW-DP 寄存器 .....	376
23.4.6 SW-AP 寄存器 .....	377
23.5 内核调试 .....	378
23.6 BPU (断点单元) .....	378
23.6.1 BPU 功能 .....	378
23.7 DWT (数据观察点) .....	379
23.7.1 DWT 功能 .....	379

23.7.2 DWT 程序计数器采样寄存器 .....	379
23.8 调试组件 (DBG) .....	380
23.8.1 对低功耗模式的调试支持 .....	380
23.8.2 对定时器、看门狗的调试支持 .....	380
24 典型应用电路图 .....	381
25 电气特性 .....	382
25.1 测试条件 .....	382
25.1.1 最小和最大数值 .....	382
25.1.2 典型数值 .....	382
25.2 绝对最大额定值 .....	383
25.3 工作条件 .....	385
25.3.1 通用工作条件 .....	385
25.3.2 上电和掉电时的工作条件 .....	385
25.3.3 内嵌复位和 LVD 模块特性 .....	386
25.3.4 内置的参考电压 .....	388
25.3.5 供电电流特性 .....	388
25.3.6 从低功耗模式唤醒的时间 .....	391
25.3.7 外部时钟源特性 .....	392
25.3.7.1 外部输入高速时钟 .....	392
25.3.8 内部时钟源特性 .....	393
25.3.8.1 内部 RCH 振荡器 .....	393
25.3.8.2 内部 RCL 振荡器 .....	393
25.3.9 存储器特性 .....	394
25.3.10 EFT 特性 .....	394
25.3.11 ESD 特性 .....	395
25.3.12 I/O 端口特性 .....	395
25.3.12.1 输出特性——端口 .....	395
25.3.12.2 输入特性——端口 PA,PB,PC,PD .....	397
25.3.12.3 端口外部输入采样要求——Timer Clock .....	397
25.3.12.4 端口漏电特性——PA,PB,PC,PD .....	398
25.3.13 RESETB 引脚特性 .....	398
25.3.14 ADC 特性 .....	398
25.3.15 TIM 定时器特性 .....	401
25.3.16 通信接口 .....	403
25.3.16.1 I2C 特性 .....	403
25.3.16.2 SPI 特性 .....	404
26 封装尺寸 .....	407
27 附录 A SysTick 定时器 .....	411
27.1 SysTick 定时器简介 .....	411
27.2 设置 SysTick .....	411
27.3 SysTick 寄存器 .....	412
27.3.1 SysTick 控制和状态寄存器 (CTRL) .....	412

---

27.3.2 SysTick 重载寄存器（LOAD） .....	412
27.3.3 SysTick 当前值寄存器（VAL） .....	412
27.3.4 SysTick 校准值寄存器（CALIB） .....	413
28 附录 B 文档约定 .....	414
28.1    寄存器相关缩写词列表.....	414
28.2    词汇表.....	414
版本记录 & 联系方式 .....	415

Preliminary  
to 立创商城

## 表目录

表 2-1 模块信号说明 .....	35
表 3-1 地址划分表 .....	37
表 7-1 Cortex-M0+ 处理器中断一览 .....	72
表 7-2 外部中断与 NVIC 中断输入对应关系 .....	79
表 8-1 端口功能配置表 .....	98
表 8-2 GPIO 寄存器一览表 .....	108
表 9-1 FLASH 容量划分 .....	122
表 10-1 RAM 地址映射 .....	134
表 11-1 BTIM 寄存器列表 .....	141
表 12-1 Timer 寄存器列表 .....	156
表 13-1 Timer 寄存器列表 .....	206
表 14-1 CTRIM 寄存器 .....	246
表 15-1 IWDT 寄存器列表 .....	258
表 16-1 WWDT 寄存器列表 .....	266
表 17-1 Mode0/1/2/3 数据结构 .....	271
表 17-2 B8 数据含义 .....	272
表 17-3 SCL 为 4MHz 波特率计算表 .....	276
表 17-4 SCLK 为 8MHz 波特率计算表 .....	277
表 17-5 SCLK 为 16MHz 波特率计算表 .....	277
表 17-6 SCLK 为 24MHz 波特率计算表 .....	278
表 17-7 SCLK 为 32MHz 波特率计算表 .....	278
表 17-8 SCLK 为 48MHz 波特率计算表 .....	279
表 18-1 SPI 中断 .....	303
表 18-2 SPI 寄存器列表 .....	307
表 19-1 I2C 时钟信号波特率 .....	323
表 20-1 ADC 寄存器 .....	352
表 21-1 LVD 寄存器 .....	363
表 25-1 工作电流特性 .....	390
表 25-2 高级定时器 (ADVTIM) 特性 .....	401
表 25-3 通用定时器特性 .....	401
表 25-4 低功耗定时器特性 .....	402
表 25-5 IWDT 特性 .....	402
表 25-6 WWDT 特性 .....	402
表 25-7 I2C 接口特性 .....	403

## 图目录

图 2-1 管脚配置图 .....	32
图 3-1 系统架构示意图 .....	36
图 4-1 工作模式状态转换图 .....	38
图 5-1 时钟控制模块框图 .....	45
图 5-2 时钟切换示意图 .....	48
图 6-1 复位来源示意图 .....	65
图 7-1 只使用了高两位的优先级寄存器 .....	73
图 7-2 中断向量表 .....	74
图 7-3 中断激活和挂起状态 .....	75
图 7-4 中断挂起状态被清除然后被重新确认 .....	76
图 7-5 中断退出时若中断请求保持高电平就会引起中断处理的再次执行 .....	76
图 7-6 中断处理中产生的中断挂起也可以被确认 .....	77
图 7-7 中断结构图 .....	80
图 8-1 通用端口电路示意图 .....	97
图 8-2 高阻-模拟配置示意图 .....	99
图 8-3 输入悬空/上拉来配置示意图 .....	101
图 8-4 读取端口管脚数据同步图 .....	101
图 8-5 输出配置示意图 .....	102
图 8-6 复用功能配置示意图 .....	103
图 11-1 BTIM 框图 .....	136
图 11-2 计数器波形 .....	137
图 11-3 定时模式框图 .....	138
图 11-4 计数器模式框图 .....	138
图 11-5 触发启动器模式框图 .....	139
图 11-6 触发启动模式计数示意图 .....	139
图 11-7 门控模式框图 .....	140
图 11-8 门控模式计数示意图 .....	140
图 12-1 GTIM 框图 .....	149
图 12-2 计数器波形 .....	150
图 12-3 定时模式框图 .....	151
图 12-4 计数器模式框图 .....	151
图 12-5 触发启动器模式框图 .....	152
图 12-6 触发启动模式计数示意图 .....	152
图 12-7 门控模式框图 .....	153
图 12-8 门控模式计数示意图 .....	153
图 12-9 捕获功能示意图 .....	154
图 12-10 比较功能示意图 .....	155
图 13-1 ATIM3 框图 .....	166
图 13-2 自由计数框图 .....	168
图 13-3 重载计数波形 .....	168

图 13-4 16 位重载计数波形 .....	169
图 13-5 32 位自由计数波形 .....	169
图 13-6 自由计数时序图 .....	170
图 13-7 重载计数时序图 (预分频设置为 2) .....	170
图 13-8 PWC 测量框图 .....	172
图 13-9 高电平脉冲宽度测量 .....	173
图 13-10 下降沿到下降沿周期测量 .....	173
图 13-11 上升沿到上升沿周期测量 .....	173
图 13-12 上升沿到上升沿周期测量单次模式 .....	175
图 13-13 计数器框图 .....	177
图 13-14 无预分频的向上计数 .....	178
图 13-15 带预分频的上计数 .....	179
图 13-16 不带预分频的下计数 .....	179
图 13-17 带预分频的下计数 .....	180
图 13-18 带预分频的上下计数 .....	180
图 13-19 边沿对齐计时器波形 (DIR =1) .....	180
图 13-20 边沿对齐计时器波形 (DIR =0) .....	181
图 13-21 中心对齐计数器波形 .....	181
图 13-22 重复计数器产生更新时序 (RCR.UD=0 RCR.OV=0) .....	182
图 13-23 三角波模式下缓存使能 .....	183
图 13-24 三角波模式下缓存无效 .....	183
图 13-25 锯齿波模式下上计数缓存使能 .....	184
图 13-26 锯齿波模式下上计数缓存无效 .....	184
图 13-27 锯齿波模式下计数缓存使能 .....	184
图 13-28 锯齿波模式下计数缓存无效 .....	185
图 13-29 锯齿波模式下计数比较缓存使能 .....	185
图 13-30 OCREF 输出框图 .....	186
图 13-31 锯齿波计数单点比较 OCREF 输出波形 (OCMx=111) .....	187
图 13-32 三角波计数单点比较 OCREF 输出波形 (OCMx=111) .....	187
图 13-33 锯齿波计数双点比较 OCREF 输出 (OCMx=111) .....	188
图 13-34 三角波计数双点比较 OCREF 输出 (OCMx=111) .....	188
图 13-35 独立 PWM 输出框图 .....	189
图 13-36 CCPx=0 时 PWM 输出波形 .....	189
图 13-37 CCPx=1 时 PWM 输出波形 .....	189
图 13-38 互补 PWM 输出框图 .....	190
图 13-39 互补 PWM 输出波形图 .....	190
图 13-40 互补 PWM 输出波形图 .....	191
图 13-41 死区时间 .....	191
图 13-42 三角波模式单脉冲计数 .....	192
图 13-43 锯齿波上计数单脉冲模式 .....	193
图 13-44 锯齿波下计数单脉冲模式 .....	193
图 13-45 中断示意图 .....	194

---

图 13-46 捕获功能框图 .....	195
图 13-47 捕获时序图 .....	195
图 13-48 CHA 端口选择 .....	195
图 13-49 CHB 端口选择 .....	196
图 13-50 从模式示意图 .....	200
图 13-51 门控功能 .....	201
图 13-52 触发和复位功能 .....	201
图 13-53 编码计数 .....	203
图 13-54 ADC 触发 .....	204
图 17-1 结构框图 .....	270
图 17-2 Mode0 发送数据 .....	273
图 17-3 Mode0 接收数据 .....	273
图 17-4 Mode1 发送数据 .....	274
图 17-5 Mode1 接收数据 .....	274
图 17-6 Mode2 发送数据 .....	274
图 17-7 Mode2 接收数据 .....	275
图 17-8 RTS 硬件流控信号 .....	281
图 17-9 CTS 硬件流控信号 .....	282
图 18-1 SPI 框图 .....	293
图 18-2 SPI 通信格式时序图 .....	294
图 18-5 从机选择 NSS 管脚配置 .....	295
图 22-6 SPI 全双工通信 .....	296
图 22-7 SPI 单线半双工通信 .....	297
图 22-8 SPI 单工发送通信示意 .....	298
图 18-9 SPI 多机系统 .....	299
图 18-3 SPI 主机模式一般标志位说明图 .....	302
图 18-4 SPI 从机模式一般标志位说明图 .....	302
图 19-1 I2C 传输协议 .....	319
图 20-1 ADC 示意框图 .....	345
图 20-2 ADC 转换时序图 .....	346
图 20-3 ADC 转换外部触发源示意图 .....	348
图 21-1 LVD 框图 .....	359
图 21-2 LVD 迟滞响应 .....	360
图 21-3 LVD 滤波输出 .....	360
图 23-1 调试支持框图 .....	371
图 25-1 POR/Brown Out 示意图 .....	386
图 25-2 输出端口 VOH/VOL 实测曲线 .....	396
图 25-3 I2C 接口时序 .....	403

## 简介

HC32F002 系列是一款旨在延长便携式测量系统的电池使用寿命的低功耗、宽电压工作范围的 MCU。集成 10 位 1Msps 高精度 SARADC，内置高性能 PWM 定时器，UART、SPI、I2C 等丰富的通讯外设，具有高整合度、高抗干扰、高可靠性和低功耗的特点。本产品内核采用 Cortex-M0+ 内核，配合成熟的 Keil & IAR 调试开发软件，支持 C 语言及汇编语言。

## 低功耗 MCU 典型应用

- 传感器应用、物联网应用
- 智能仪表、无线模块、温控器、货架标签
- 智能交通、报警系统
- 智能家居、医疗设备

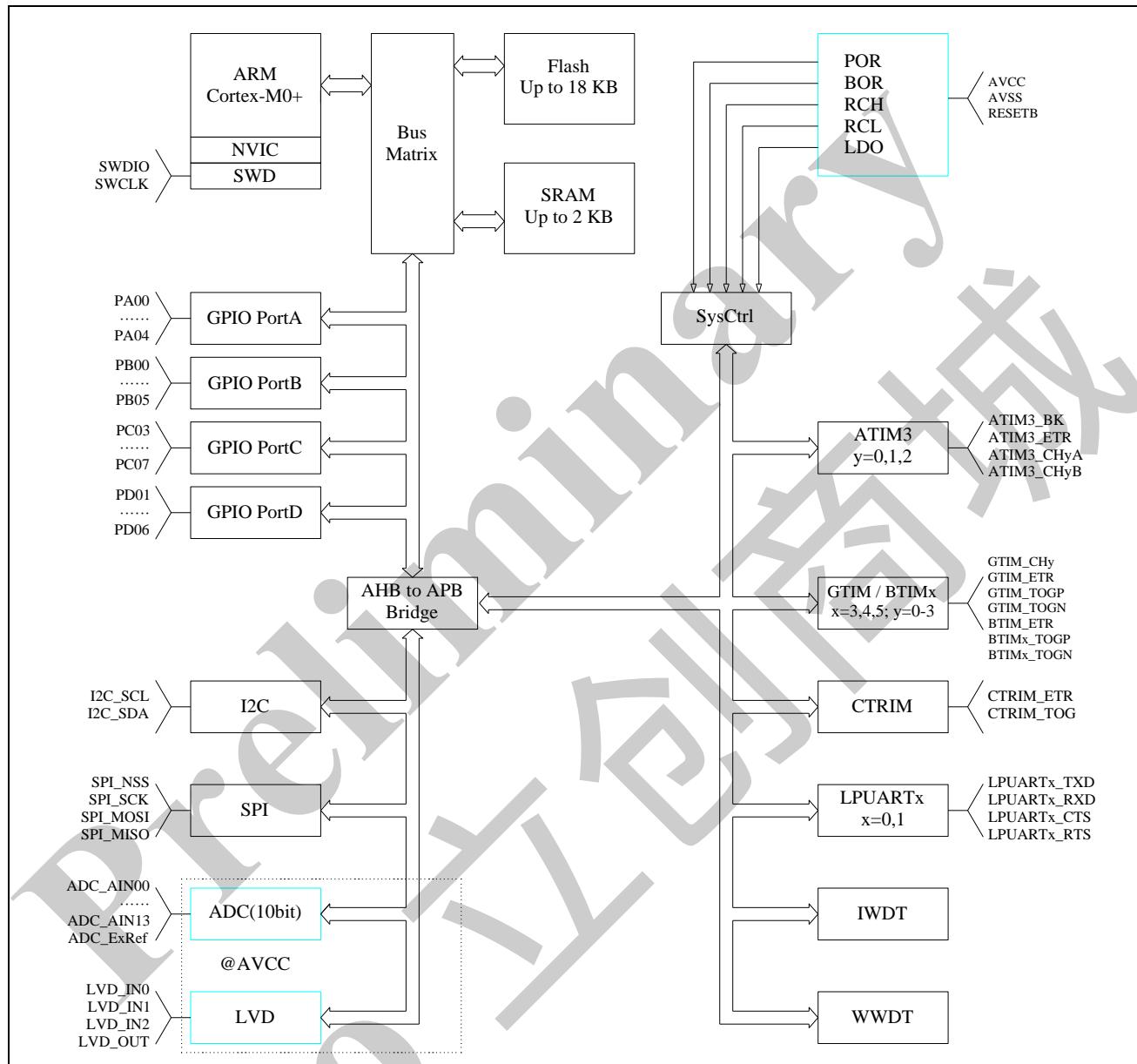
Preliminary  
to be confirmed

## 产品特性 低功耗 MCU

- 48MHz Cortex-M0+ 32 位 CPU 平台
- HC32F002 系列具有灵活的功耗管理系统
  - 5 $\mu$ A @ 3V 深度休眠模式: 所有时钟关闭, 上电复位有效, IO 状态保持, IO 中断有效, 所有寄存器、RAM 和 CPU 数据保存状态时的功耗
  - 15 $\mu$ A/MHz@3V@48MHz 休眠模式: CPU 停止, 外设处于可运行状态, 主时钟运行
  - 100 $\mu$ A/MHz@3V@48MHz 工作模式: CPU 和外设可运行, 从 FLASH 运行程序
- 18K 字节 FLASH 存储器, 具有擦写保护功能, 支持 ISP、ICP、IAP, 4 级安全保护
- 2K 字节 RAM 存储器
- 通用 I/O 管脚 (22IO/24PIN, 18IO/20PIN)
- 时钟
  - 内部高速时钟 44.24/48MHz
  - 内部低速时钟 32.8/38.4KHz
  - 外部输入 1-16MHz
- 定时器/计数器
  - 1 个高级 16 位定制器, 支持 3 相互补 PWM 输出
  - 1 个复合定时器, 可配置为通用 16 位定时器/计数器或 3 个 16 位基本定时器; 作为通用定时器时支持 4 通道捕获比较, 4 通道 PWM 输出; 作为基本定时器时, 每个定时器支持两路翻转输出
  - 1 个独立看门狗电路, 内部低速时钟振荡器提供 WDT 计数
  - 1 个窗口看门狗电路, 使用系统时钟
  - 1 个低功耗 16 位定时器, 支持自动唤醒
  - 1 个 CM0+ 内置 24 位 SysTick 定时器
- 通讯接口
  - 2 路 LPUART 通讯接口
  - 1 路 SPI 标准通讯接口
  - 1 路 I2C 标准通讯接口
- 定时器作为蜂鸣器频率发生器
- 全球唯一 10 字节 ID 号
- 128 字节 OTP 存储器, 只能通过编程器写入
- 10 位高达 1Msps 采样速率的 SARADC
- 集成低电压侦测器, 可配置 16 阶比较电压, 可监控端口电压以及电源电压
- SWD 调试解决方案, 提供全功能调试器
- 工作条件: -40 ~ 105°C, 1.7V ~ 5.5V
- 封装形式: QFN24/20, TSSOP24/20

# 1 功能模块

## 1.1 功能模块框图



## 1.2 32 位 Cortex M0+ 内核

ARM® Cortex®-M0+ 处理器源于 Cortex-M0，包含了一颗 32 位 RISC 处理器，运算能力达到 0.95 Dhystone MIPS/MHz。同时加入了多项全新设计，改进调试和追踪能力、减少每条指令循环 (IPC) 数量和改进 Flash 访问的两级流水线等，更纳入了节能降耗技术。Cortex-M0+ 处理器全面支持已整合 Keil & IAR 调试器。

Cortex-M0+ 包含了一个硬件调试电路，支持 2-pin 的 SWD 调试界面。

ARM Cortex-M0+ 特性：

指令集	Thumb / Thumb-2
流水线	2级流水线
性能效率	2.46 CoreMark / MHz
性能效率	0.95 DMIPS / MHz in Dhystone
中断	15个快速中断
中断优先级	可配置4级中断优先级
增强指令	多周期32位乘法器
调试	Serial-wire 调试端口，支持4个硬中断 (break point) 以及1个观察点 (watch point)

## 1.3 存储器

### 1.3.1 18 Kbytes FLASH

内建全集成 FLASH 控制器，无需外部高压输入，由全内置电路产生高压来编程。支持 ISP、IAP、ICP 功能。具有 4 级安全保护等级。

### 1.3.2 2 Kbytes RAM

任意功耗模式，RAM 数据均不会丢失。

## 1.4 时钟系统

一个频率为 44~48MHz 的高精度内部时钟 RCH。出厂已预置 44.24MHz、48MHz 校准值。

一个频率为 32.8KHz / 38.4KHz 的内部时钟 RCL。

## 1.5 工作模式

- 运行模式(Active Mode): CPU 运行, 片内外设运行。
- 休眠模式(Sleep Mode): CPU 停止运行, 片内外设运行。
- 深度休眠模式(Deep sleep Mode): CPU 停止运行, 低功耗片内外设运行。

## 1.6 端口控制器 GPIO

最多可提供 22 个 GPIO 端口, 其中部分 GPIO 与模拟端口复用。每个端口由独立的控制寄存器位来控制。所有管脚支持边沿触发中断和电平触发中断, 可从各种休眠模式下将 MCU 唤醒到工作模式。支持位置位、位清零、位置位清零操作。支持 Push-Pull CMOS 推挽输出、Open-Drain 开漏输出。内置上拉电阻, 带有施密特触发器功能。每个 IO 最大支持 18mA 的电流驱动能力, 20 管脚及以下封装其中 4 个 IO 支持大驱动输出。

## 1.7 中断控制器 NVIC

Cortex-M0+处理器内置了嵌套向量中断控制器(NVIC), 支持最多 15 个中断请求(IRQ)输入; 有四个中断优先级, 可处理复杂逻辑, 能够进行实时控制和中断处理。

## 1.8 复位控制器 RESET

本产品具有 7 个复位信号来源, 每个复位信号可以让 CPU 重新运行, 绝大多数寄存器会被重新复位, 程序计数器 PC 会指向起始地址。

## 1.9 定时器 TIM

类型	名称	位宽	预除频	计数方向	PWM	捕获	互补输出
高级定时器	ATIM3	16/32	1/2/4/8/16/ 32/64/256	上计数/ 下计数/ 上下计数	6	6	3
通用定时器	GTIM	16	1~32768	上计数	4	4	无
	BTIM3-5	16	1~32768	上计数	无	无	无
低功耗定时器	CTRIM	16	1~32768	上计数	无	无	无

通用定时器是支持 4 路比较捕获功能的定时器, 可以配置成 3 个基本定时器。

低功耗定时器是异步 16 位定时/计数器, 在系统时钟关闭后仍然可以通 RCL 进行计时/

计数。通过中断在低功耗模式下唤醒系统。

高级定时器包含定时器 ATIM3，其特性如下：

- PWM 独立输出，互补输出
- 捕获输入
- 死区控制
- 刹车控制
- 边沿对齐、对称中心对齐与非对称中心对齐 PWM 输出
- 正交编码计数功能
- 单脉冲模式
- 外部计数功能

ATIM3 是多通道的通用定时器，可以产生 3 组 PWM 互补输出或 6 路 PWM 独立输出，最多 6 路输入捕获。具有死区控制功能。

## 1.10 看门狗 WDT

IWDT 是一个可配置的 12 位定时器，在 MCU 异常的情况下提供复位；RCL 时钟输入作为计数器时钟。只有写入特定序列才能重启 WDT。

WWDT 是一个 7 位定时器，通常被用来监测，由外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行序列而产生的软件故障。

## 1.11 低功耗同步异步收发器 LPUART

2 路低功耗模式下可以工作的同步异步收发器（Low Power Universal Asynchronous Receiver/Transmitter），LPUART0/LPUART1，其基本功能如下所示：

- 传输时钟 SCLK（SCLK 可选择 RCL 或 PCLK）
- 系统低功耗模式下收发数据
- 半双工和全双工传输
- 8/9-Bit 传输数据长度
- 硬件奇偶校验
- 1/1.5/2-Bit 停止位
- 四种不同传输模式

- 16-Bit 波特率计数器
- 多机通讯
- 硬件地址识别
- 硬件流控
- 支持单线模式

## 1.12 串行外设接口 SPI

- 可配置为主机或者从机，支持多机模式
- 主机模式最大分频系数为 PCLK/2，最高通信速率为 12M bps
- 从机模式最大分频系数为 PCLK/4，最高通信速率为 8M bps
- 多种通信模式：全双工、单线半双工、单工
- 两种传输顺序：先收发 MSB 或先收发 LSB
- 多种数据帧长度：4bit ~ 16bit
- 两种 NSS 方式：硬件控制、软件控制
- 可配置的串行时钟极性和相位

## 1.13 I2C 总线

1 路 I2C，采用串行同步时钟，可实现设备之间以不同的速率传输数据。

I2C 基本特性：

- 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

## 1.14 时钟校准器 CTRIM

时钟校准定时器可以调整校准 RC 时钟频率，也可以调整校准其他 RC 振荡的时钟频率，还可以作为一个通用定时器或者自动唤醒定时器来使用。

## 1.15 蜂鸣器 Buzzer

定时器功能复用输出为 Buzzer 提供可编程驱动频率。该蜂鸣器端口可提供 18mA 的 sink 电流，互补输出，不需要额外的三极管。

## 1.16 模数转换器 ADC

外部的模拟信号需要转变成数字信号才能由 MCU 进一步处理。内部集成了一个 10 位高精度、高转换速率的逐次逼近型模数转换器(SAR ADC)模块。具有以下特性：

- 10 位转换精度；
- 1M SPS 转换速度；
- 支持单次转换及持续转换；
- 2 种参考源：AVCC 电压、ExRef 管脚；
- 15 路输入通道，包括 14 路外部管脚输入、1 路内建 BGR 0.9V 电压；  
注：选择 AVCC 作为参考源，测量 BGR 0.9V；可计算出 AVCC 电压。
- ADC 的电压输入范围：0~Vref；
- 支持片内外设自动触发 ADC 转换，有效降低芯片功耗并提高转换的实时性。

## 1.17 低电压检测器 LVD

对芯片电源电压或芯片管脚电压进行检测。16 档电压监测值（1.8~3.3V）。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。

LVD 基本特性：

- 4 路监测源，AVCC、PA03、PC03、PD04；
- 16 阶阈值电压，1.8~3.3V 可选；
- 8 种触发条件，高电平、上升沿、下降沿组合；
- 2 种触发结果，复位、中断；
- 8 阶滤波配置，防止误触发；
- 具备迟滞功能，强力抗干扰。

## 1.18 嵌入式调试系统

嵌入式调试解决方案，提供全功能的实时调试器，配合标准成熟的 Keil/IAR 等调试开发软件。支持 4 个硬断点以及多个软断点。

## 1.19 编程模式

支持两种编程模式：在线编程、离线编程。

支持两种编程协议：ISP 协议、SWD 协议。

支持统一编程接口：ISP 协议与 SWD 协议共用 SWD 端口。

## 1.20 器件电子签名

每颗芯片出厂前具备唯一的 10 字节设备标识号，包括 wafer lot 信息，以及芯片坐标信息等。内建 128 字节 OTP，仅可通过编程器写入一次。

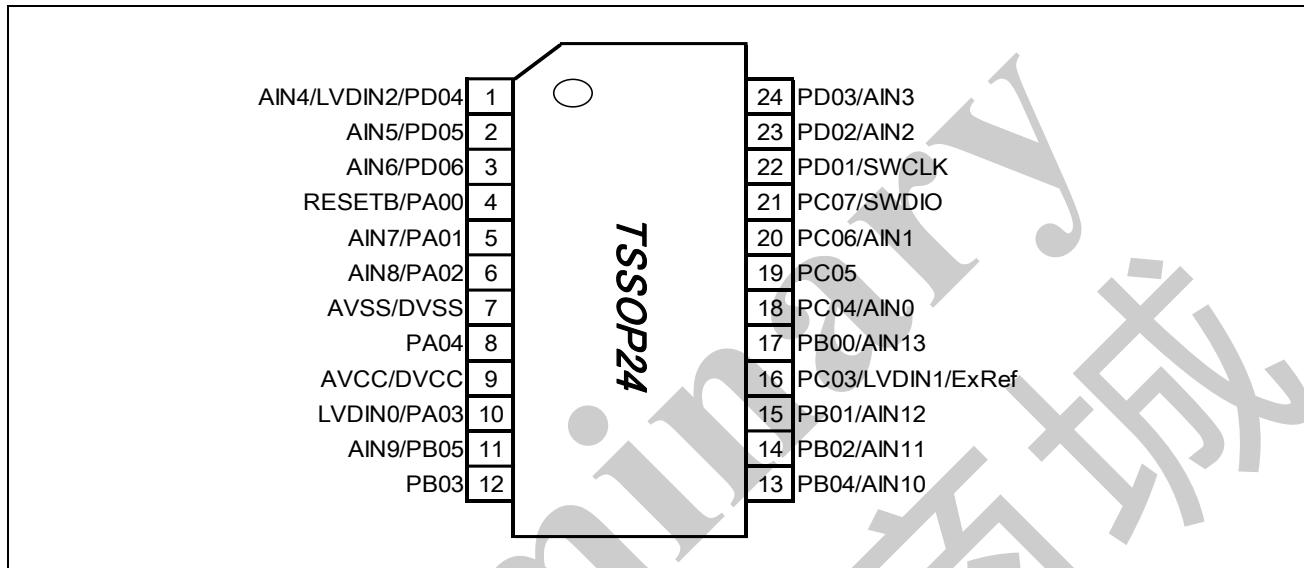
## 1.21 高安全性

加密型嵌入式调试解决方案，提供全功能的实时调试器。

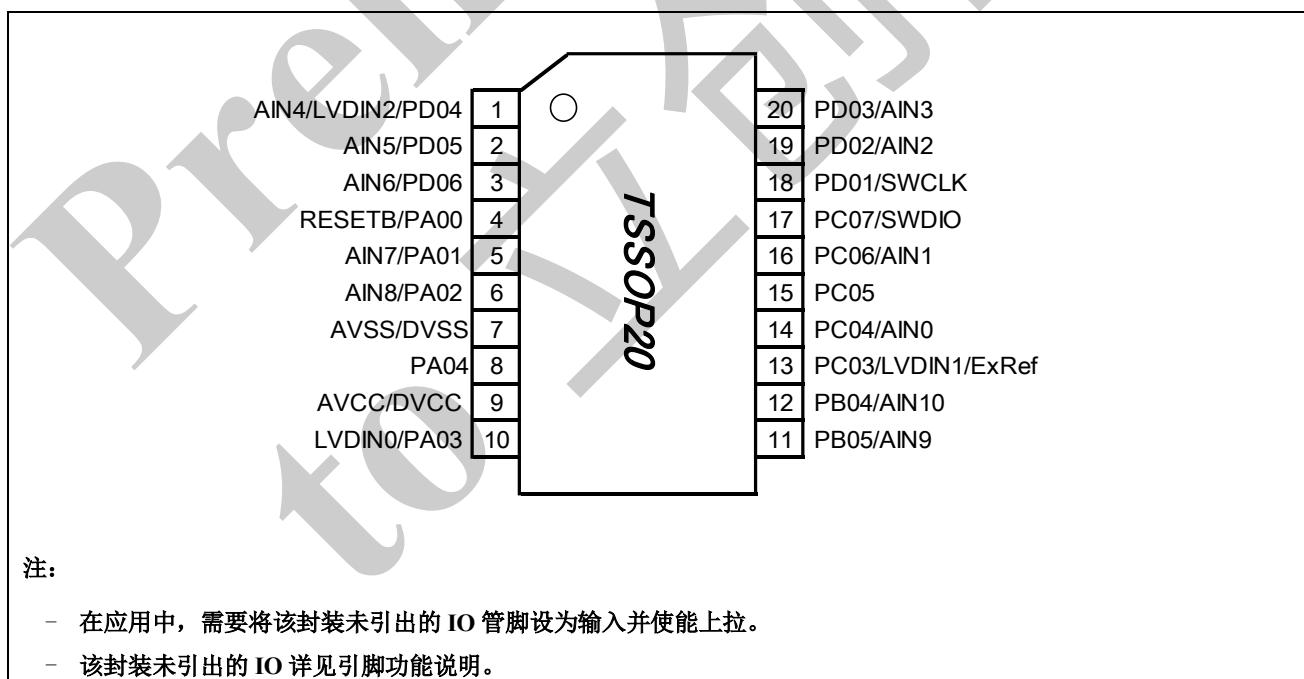
## 2 引脚配置及功能

### 2.1 引脚配置图

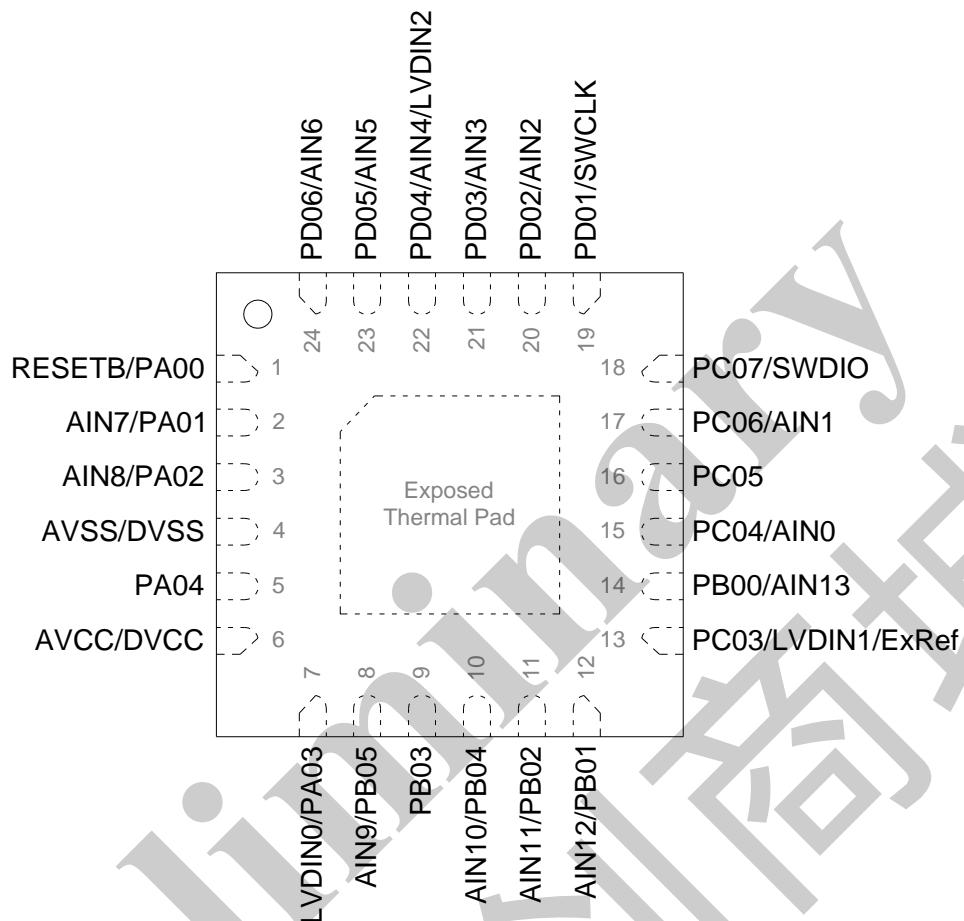
HC32F002D4PB-TSSOP24



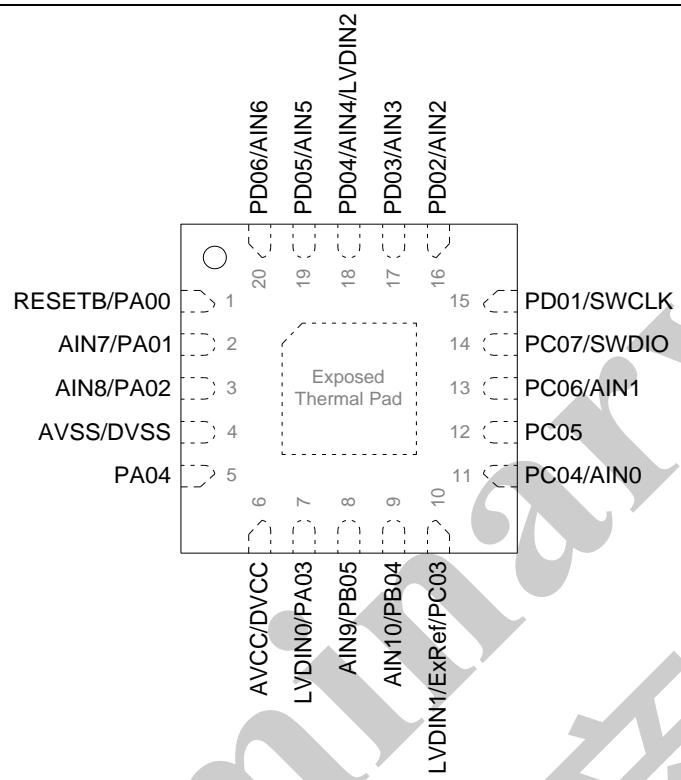
HC32F002C4PB-TSSOP20



## HC32F002D4UB-QFN24TR



## HC32F002C4UB-ZFN20TR



注:

- 在应用中, 需要将该封装未引出的 IO 管脚设为输入并使能上拉。
- 该封装未引出的 IO 详见引脚功能说明。

图 2-1 引脚配置图

## 2.2 引脚功能说明

QFN20	QFN24	TSSOP20	TSSOP24	NAME	DIGITAL	ANALOG
1	1	4	4	PA00	-	RESETB
2	2	5	5	PA01	LPUART0_RXD I2C_SDA GTIM_TOGP HCLK_OUT	ADC_AIN7
3	3	6	6	PA02	LPUART0_TXD I2C_SCL GTIM_TOGN LVD_OUT	ADC_AIN8
4	4	7	7	AVSS/DVSS		
5	5	8	8	PA04	GTIM_TOGP LVD_OUT SPI_NSS	-
6	6	9	9	AVCC/DVCC		
7	7	10	10	PA03	GTIM_CH2 SPI_NSS CTRIM_ETR CTIRM_TOG	LVD_IN0
8	8	11	11	PB05	I2C_SDA ATIM3_BK CTRIM_ETR CTIRM_TOG	ADC_AIN9
	9		12	PB03	ATIM3_ETR GTIM_CH0 LPUART1_CTS	-
9	10	12	13	PB04	I2C_SCL ATIM3_CH2B LPUART0_CTS	ADC_AIN10
	11		14	PB02	ATIM3_CH2B GTIM_CH1 LPUART1_RTS	ADC_AIN11
	12		15	PB01	ATIM3_CH1B GTIM_CH2 LPUART0_CTS	ADC_AIN12

QFN20	QFN24	TSSOP20	TSSOP24	NAME	DIGITAL	ANALOG
10	13	13	16	PC03	ATIM3_CH2A ATIM3_CH0B LPUART0_RTS	LVD_IN1 ADC_ExRef
	14		17	PB00	ATIM3_CH0B GTIM_CH3 LPUART0_RTS	ADC_AIN13
11	15	14	18	PC04	ATIM3_CH0B ATIM3_CH1B TCLK_OUT	ADC_AIN0
12	16	15	19	PC05	SPI_SCK GTIM_CH3 ATIM3_ETR	-
13	17	16	20	PC06	SPI_MOSI ATIM3_CH0A LPUART1_RTS	ADC_AIN1
14	18	17	21	PC07 SWDIO	SPI_MISO ATIM3_CH1A LPUART0_RXD	-
15	19	18	22	PD01 SWCLK	GTIM_ETR ATIM3_BK LPUART0_TXD	-
16	20	19	23	PD02	GTIM_CH2 ATIM3_CH2A LPUART1_CTS	ADC_AIN2
17	21	20	24	PD03	GTIM_CH1 ATIM3_CH1A LPUART1_RXD	ADC_AIN3
18	22	1	1	PD04	GTIM_CH0 LPUART1_TXD SPI_SCK	ADC_AIN4 LVD_IN2
19	23	2	2	PD05	LPUART1_TXD ATIM3_CH0A SPI_MISO	ADC_AIN5
20	24	3	3	PD06	LPUART1_RXD GTIM_ETR SPI_MOSI	ADC_AIN6

每个管脚的数字功能由 AFRx 寄存器进行控制，详见 GPIO 章节。

## 2.3 模块信号说明

模块	管脚名称	描述
电源	DVCC	数字电源
	AVCC	模拟电源
	DVSS	数字地
	AVSS	模拟地
ADC	ADC_AIN0~13	ADC输入通道0~13
	ADC_ExRef	ADC外部参考电压
LVD	LVD_IN0	电压侦测输入0
	LVD_IN1	电压侦测输入1
	LVD_IN2	电压侦测输入2
	LVD_OUT	电压侦测输出
LPUART x=0,1	LPUARTx_TXD	LPUART数据发送端
	LPUARTx_RXD	LPUART数据接收端
	LPUARTx_CTS	LPUART CTS
	LPUARTx_RTS	LPUART RTS
CTRIM	CTRIM_ETR	CTRIM外部同步信号
	CTRIM_TOG	CTRIM的翻转输出信号
SPI	SPI_MISO	SPI模块主机输入从机输出数据信号
	SPI_MOSI	SPI模块主机输出从机输入数据信号
	SPI_SCK	SPI模块时钟信号
	SPI_NSS	SPI片选
I2C	I2C_SDA	I2C模块数据信号
	I2C_SCL	I2C模块时钟信号
基本定时器 BTIMx x = 3,4,5	BTIMx_TOGP	BTIMx的翻转输出信号
	BTIMx_TOGN	
通用定时器 GTIM y=0,1,2,3	GTIM_CHy	GTIM的捕获输入比较输出
	GTIM_ETR	GTIM的外部计数输入信号
	GTIM_TOGP	GTIM的翻转输出信号
	GTIM_TOGN	
高级定时器 ATIM3 y=0,1,2	ATIM3_CHyA	ATIM3的捕获输入比较输出A
	ATIM3_CHyB	ATIM3的捕获输入比较输出B
	ATIM3_ETR	ATIM3的外部计数输入信号
	ATIM3_BK	ATIM3的刹车信号

表 2-1 模块信号说明

注意：

- IO 端口复位为输入高阻状态，休眠模式和深度休眠模式保持之前的端口状态。

### 3 系统结构

#### 3.1 概述

本产品系统由以下部分组成:

- 1 个 AHB 总线 Master:
  - Cortex-M0+
- 4 个 AHB 总线 Slaves:
  - FLASH 存储器
  - SRAM 存储器
  - AHB0, AHB to APB Bridge 包含所有 APB 接口外设
  - AHB1, 包含所有 AHB 接口外设

整个系统总线结构采用多层次 AHB-lite 总线互连实现。如下图所示:

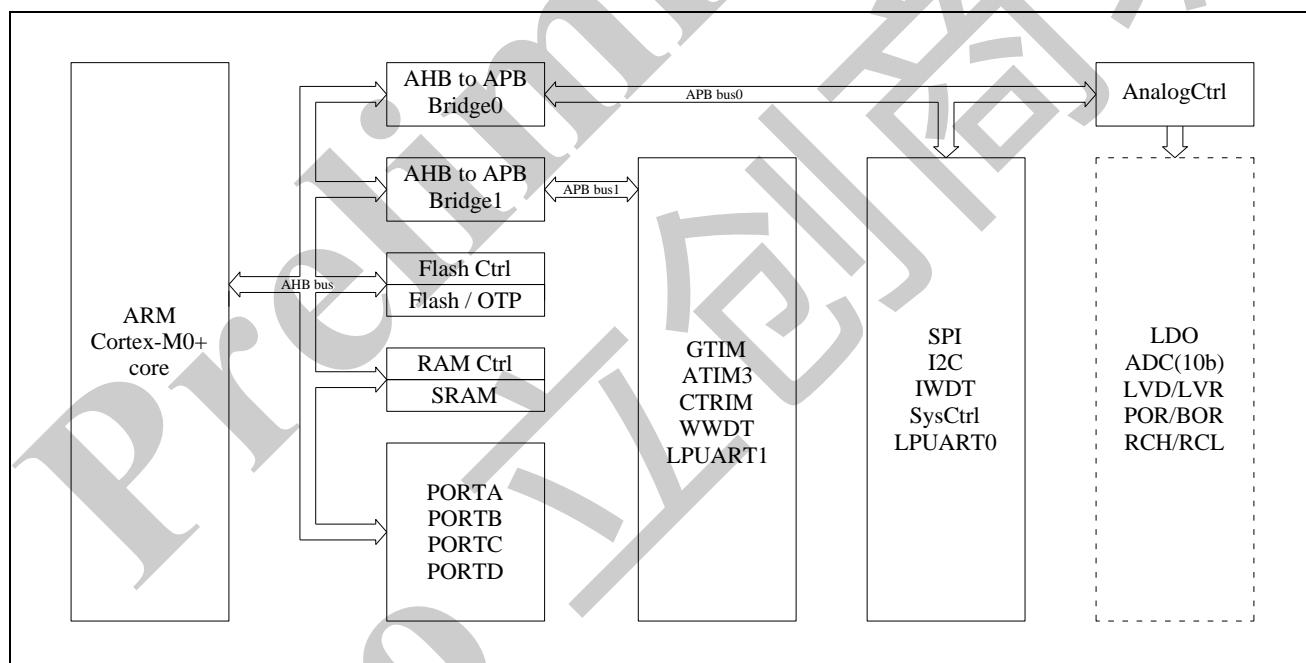


图 3-1 系统架构示意图

### 3.2 存储器和模块地址分配

类别	Boundary Address	Size	Memory Area
存储器	0x0000_0000 – 0x0000_47FF	18K Byte	FLASH
	0x0010_0F00 – 0x0010_0F7F	128 Byte	OTP
	0x2000_0000 – 0x2000_07FF	2K Byte	SRAM
APB0 外设	0x4000_0000 – 0x4000_03FF	1K Byte	LPUART0
	0x4000_0400 – 0x4000_07FF	1KByte	I2C
	0x4000_0800 – 0x4000_0BFF	1KByte	SPI
	0x4000_0F80 – 0x4000_0FFF	128Byte	IWDT
	0x4000_2000 – 0x4000_23FF	1KByte	SYSCTRL
	0x4000_2400 – 0x4000_27FF	1KByte	ANALOGCTRL
APB1 外设	0x4000_4000 – 0x4000_43FF	1KByte	LPUART1
	0x4000_5000 – 0x4000_53FF	1KByte	CTRIM
	0x4000_5800 – 0x4000_5BFF	1KByte	ATIM3
	0x4000_6800 – 0x4000_6BFF	1KByte	WWDT
	0x4000_7400 – 0x4000_77FF	1KByte	GTIM
AHB 外设	0x4002_0000 – 0x4002_03FF	1KByte	FLASH CTRL
	0x4002_0C00 – 0x4002_0FFF	1KByte	PORT CTRL

表 3-1 地址划分表

## 4 工作模式

### 4.1 工作模式概述

本产品在电源管理模块的控制下可实现三种工作模式。各工作模式下可工作的功能模块及功耗各不相同。

本产品的工作模式如下所示：

- 运行模式 (Active Mode): CPU 运行, 片内外设运行。
- 休眠模式 (Sleep Mode): CPU 停止, 片内外设运行。
- 深度休眠模式 (Deep Sleep Mode): CPU 停止, 低功耗片内外设运行。

### 4.2 工作模式切换

本产品工作时可在三种工作模式之间自由切换。在运行模式下执行 WFI 指令, 可使本产品进入休眠模式或深度休眠模式。从休眠模式或深度休眠模式通过中断唤醒, 回到运行模式。

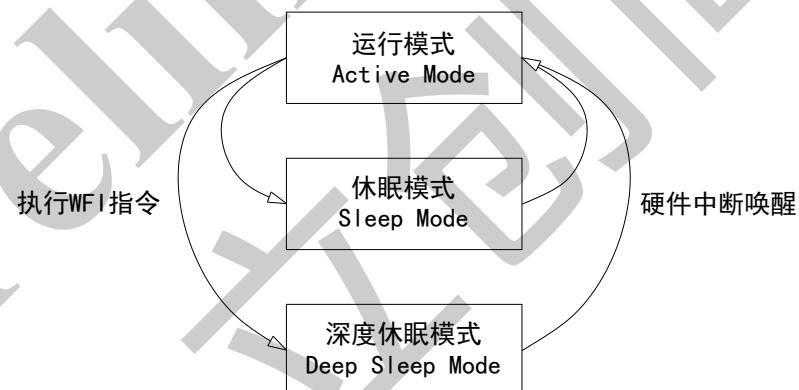


图 4-1 工作模式状态转换图

在各种工作模式下，CPU 可响应的中断见下方表格。

中断向量号	中断来源	运行模式	休眠模式	深度休眠模式
[0]	GPIO_PA	✓	✓	✓
[1]	GPIO_PB	✓	✓	✓
[2]	GPIO_PC	✓	✓	✓
[3]	GPIO_PD	✓	✓	✓
[4]	-			
[5]	ATIM3	✓	✓	
[7]	-			
[8]	LPUART0	✓	✓	✓
[9]	LPUART1	✓	✓	✓
[10]	SPI	✓	✓	
[11]	-			
[12]	I2C	✓	✓	
[13]	-			
[14]	-			
[15]	GTIM / BTIM3-5	✓	✓	
[17]	-			
[20]	-			
[21]	-			
[22]	IWDT	✓	✓	✓
	WWDT	✓	✓	
[23]	-			
[24]	ADC	✓	✓	
[27]	-			
[28]	LVD	✓	✓	✓
[30]	FLASH / RAM	✓	✓	
[31]	CTRIM	✓	✓	✓

在各种工作模式下，可响应的复位信号见下方下表。

	复位源	运行模式	休眠模式	深度休眠模式
[0]	上电掉电复位 POR / BOR	✓	✓	✓
[1]	硬件复位 RESETB管脚	✓	✓	✓
[2]	硬件复位 LVD	✓	✓	✓
[3]	硬件复位 IWDT	✓	✓	✓
[4]	硬件复位 Cortex-M0+ LOCKUP	✓		
[5]	软件复位 Cortex-M0+ SYSRESETREQ	✓		
[6]	硬件复位 WWDT	✓	✓	

### 4.3 运行模式 (Active Mode)

在该工作模式 MCU 所有功能模块均正常工作，功耗最大。

系统复位后或从中断唤醒后，MCU 均工作于运行模式。

在运行模式下可通过一定的措施降低运行功耗：

- 为系统时钟 (SystemClk) 选择更低的频率，修改 SysCtrl.CR0.ClkSrc
- 降低系统的内核时钟 (HCLK) 的频率，修改 SysCtrl.CR0.HclkPrs
- 降低系统的片内外设时钟 (PCLK) 的频率，修改 SysCtrl.CR0.PclkPrs
- 关闭未使用的片内外设的时钟，修改 SysCtrl.PeriClkEnx

### 4.4 休眠模式 (Sleep Mode)

在该工作模式下 CPU 停止运行，其它片内外设均正常工作，功耗低于运行模式。

在运行模式下，设置 SCB->SCR.SLEEPDEEP 为 0，执行 WFI 指令即可使 MCU 工作于休眠模式。当 MCU 响应硬件中断时，即可退出休眠模式回到运行模式。

在休眠模式下可通过一定的措施降低运行功耗：

- 为系统时钟 (SystemClk) 选择更低的频率，修改 SysCtrl.CR0.ClkSrc
- 降低系统的内核时钟 (HCLK) 的频率，修改 SysCtrl.CR0.HclkPrs
- 降低系统的片内外设时钟 (PCLK) 的频率，修改 SysCtrl.CR0.PclkPrs
- 关闭未使用的片内外设的时钟，修改 SysCtrl.PeriClkEnx
- 设置 SCB\_SCR.SLEEPONEXIT 为 1，使 MCU 执行完成中断服务程序后自动进入休眠模式。

### 4.5 深度休眠模式 (Deep Sleep Mode)

在该工作模式下 CPU 停止运行，片内高速时钟 RCH 自动关闭。如果 HCLK 来源为片内高速时钟，则片内高速外设停止运行，仅片内低功耗外设正常工作；如果 HCLK 来源为片内低速时钟 RCL，则片内外设均正常工作。该模式下功耗远低于休眠模式下的功耗。

若设置 SysCtrl.CR1.AutoSwitch 为 1，则 MCU 退出深度休眠模式时的系统时钟的来源为 RCH，以实现快速唤醒。

在运行模式下，设置 SCB\_SCR.SLEEPDEEP 为 1，执行 WFI 指令即可使 MCU 工作于深度休眠模式。当 MCU 响应硬件中断时，即可退出深度休眠模式。

在休眠模式下可通过一定的措施降低运行功耗：

- 选择高速时钟作为系统时钟（SystemClk），修改 SysCtrl.CR0.ClkSrc
- 关闭未使用的片内低功耗外设的时钟，修改 SysCtrl.PeriClkEnx
- 设置 SCB\_SCR.SLEEPONEXIT 为 1，使 MCU 执行完成中断服务程序后自动进入休眠模式。

## 4.6 功能模块工作状态

在三种工作模式下功能模块的工作状态对比如下表所示：

功能模块	运行模式	休眠模式	深度休眠模式
CPU	✓		
WWDT	✓	✓	
GTIM / BTIM3-5	✓	✓	
ATIM3	✓	✓	
SPI	✓	✓	
I2C	✓	✓	
ADC	✓	✓	
GPIO_PA	✓	✓	✓
GPIO_PB	✓	✓	✓
GPIO_PC	✓	✓	✓
GPIO_PD	✓	✓	✓
IWDT	✓	✓	✓
LPUART0 / LPUART1	✓	✓	✓
CTRIM	✓	✓	✓
LVD	✓	✓	✓

## 4.7 寄存器

### 4.7.1 系统控制寄存器 (SCB\_SCR)

地址: 0xE000ED10

复位值: 0x0000 0000

位	标记	功能描述	读写
31:5	RESERVED	保留	
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	RW
3	RESERVED	保留	
2	SLEEPDEEP	设为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	RW
1	SLEEPONEXIT	设为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	RW
0	RESERVED	保留	

## 5 系统控制器 (SYSCTRL)

### 5.1 系统时钟介绍

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。为了确保振荡器精度，内部时钟均具有校准功能。

本产品支持以下三个不同的时钟源作为系统时钟：

- 内部高速 RC 时钟 RCH (输出频率为 44~48MHz)
- 内部低速 RC 时钟 RCL (38.4K 与 32.8K 可配置)
- 外部管脚 PA01 输入的时钟信号 (频率为 1~16MHz)

*注 1：切换系统时钟的时钟源时，请严格按照操作步骤进行切换，详见 5.2 章节。*

通过 HCLK\_OUT 管脚，可以输出 HCLK 时钟信号；通过 TCLK\_OUT 管脚可以输出 RCH/RCL/HCLK/PCLK 时钟信号。

### 5.1.1 时钟架构图

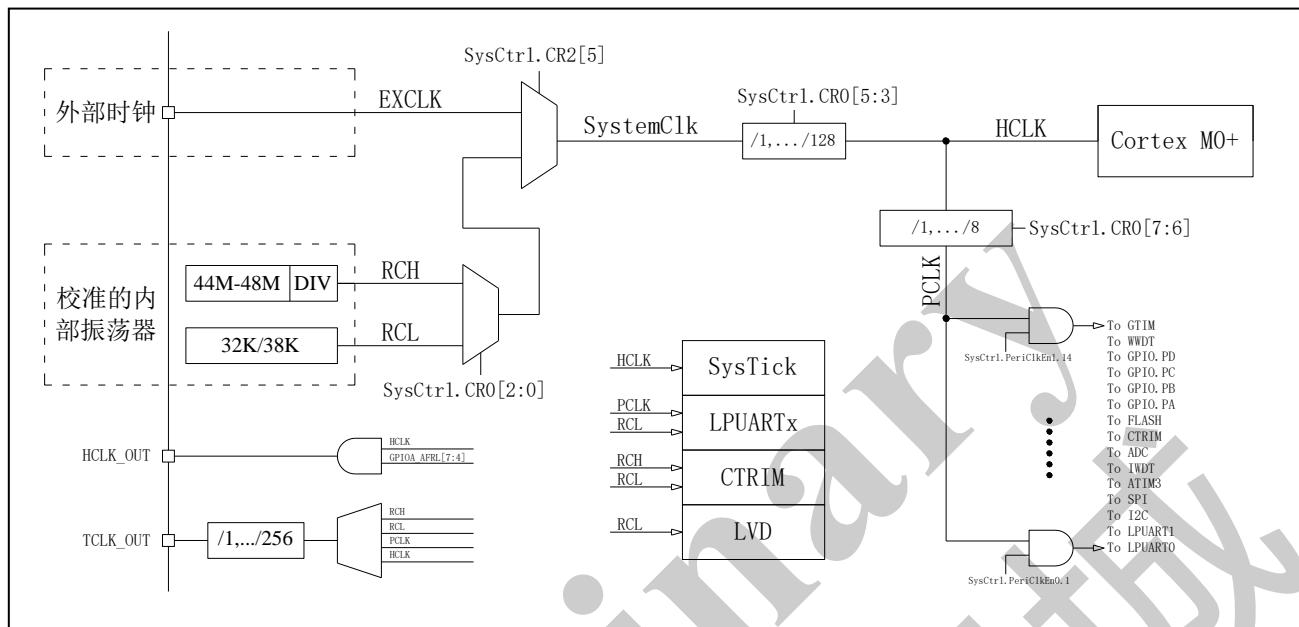


图 5-1 时钟控制模块框图

### 5.1.2 内部高速 RC 时钟 RCH

芯片上电或复位后的默认时钟源为频率为 4MHz 的 RCHCLKD；当系统进入 DeepSleep，此高速时钟会自动关闭。内部高速时钟从唤醒到稳定仅需 5us，在进入深度休眠模式前将系统时钟切换为 RCHCLKD 则系统可快速响应中断。

更改寄存器 SysCtrl.RCH.TRIM 的数值即可微调 RCH 的振荡频率。寄存器数值每增加 1 则 RCH 的振荡频率增加约 0.2%，RCH 振荡频率的安全工作范围为 44~48MHz，若将 RCH 的振荡频率调到大于 48MHz，则芯片可能出现异常。芯片在出厂时已预调好的 2 个频率点 44.24MHz 和 48MHz；如需其它频率请手动调整该寄存器的数值。若在 CTRIM 模块中使能 RCH 实时校准功能，则 RCH 在全工作范围内的精度均可达 0.5%，详见 CTRIM 章节。

更改寄存器 SysCtrl.RCH.DIV 的数值即可调整 RCH 振荡频率与 RCHCLKD 的分频比，调整范围为 1 – 16。

### 5.1.3 内部低速 RC 时钟 RCL

内部低速时钟可以通过寄存器 SysCtrl.RCL.TRIM 来选择其输出频率，可供选择的频率为 38.4KHz、32.768KHz。当系统进入 DeepSleep，此低速时钟不会自动关闭，超低功耗外设模块可以选择 RCL 作为其时钟。

### 5.1.4 外部输入时钟 EXTCLK

可以从 PA1 管脚输入频率为 1~16MHz 时钟信号作为系统时钟。从 PA1 管脚输入时钟信号的方法为：配置管脚为 GPIO 输入；设置 SysCtrl.CR2.EXTEN 为 1。

Preliminary  
to be confirmed

## 5.2 系统时钟切换

系统时钟可选 3 种时钟源：RCHCLKD、RCL、外部管脚输入时钟，由寄存器 SysCtrl.CR0.ClkSrc 及 SysCtrl.CR2.EXTEN 进行控制。RCHCLKD、RCL、外部时钟这 3 个时钟源中的任意两者可以相互切换，切换过程中需确保 HCLK 不大于 48MHz。建议按照下文所描述的时钟切换流程进行时钟切换，以减少时钟切换出错的机率。

时钟切换时需要同步配置 FLASH\_WAIT 寄存器。时钟频率不大于 24MHz 则应设置 FLASH\_WAIT 为 0；时钟频率大于 24MHz 则应设置 FLASH\_WAIT 为 1。

注：设置 FLASH\_WAIT 的值，需要先向 FLASH\_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，然后再对 FLASH\_WAIT 进行赋值，详见 FLASH 控制器章节。

### 5.2.1 标准的时钟切换流程

操作流程如下：

Step1：如新时钟源需要外部管脚，则将该管脚设置为适当的模式。

注：接外部时钟输入时需要 GPIO 输入并使能外部时钟输入。

Step2：配置新时钟源的工作参数。

Step3：使能新时钟源的振荡器。

Step4：根据当前时钟源和新时钟源两者中较高的频率，按 Flash 控制器章节流程配置 FLASH\_WAIT 寄存器。

Step5：等待新时钟源输出稳定的频率。

Step6：配置 SysCtrl.CR0.ClkSrc 及 SysCtrl.CR2.EXTEN，选择新时钟源作为系统时钟。

Step7：根据新时钟源的频率，按 Flash 控制器章节流程配置 FLASH\_WAIT 寄存器。

Step8：关闭不再使用的时钟源。

下图为时钟切换时序图：

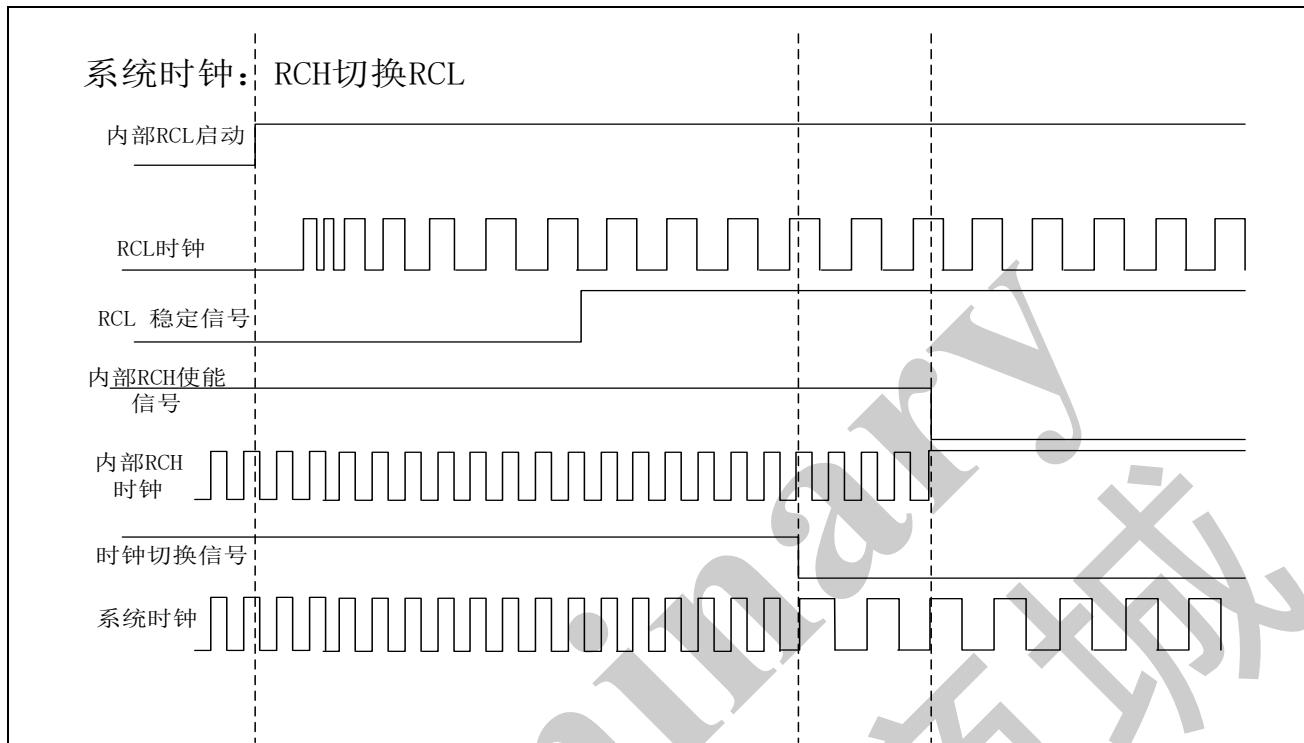


图 5-2 时钟切换示意图

### 5.2.2 RCH 不同输出频率间切换流程

配置 SysCtrl.RCH.DIV 即可使 RCH 输出不同的时钟频率。

操作流程如下方所示：

Step1：根据当前 RCHCLKD 与目标 RCHCLKD 两者中较高的频率，按 Flash 章节流程配置 FLASH\_WAIT 寄存器。

Step2：配置 SysCtrl.RCH.DIV，选择 RCH 振荡频率与 RCHCLKD 频率的分频比。

Step3：根据当前 RCHCLKD 的频率，按 Flash 章节流程配置 FLASH\_WAIT 寄存器。

### 5.2.3 从其它时钟切换到 RCL 示例

操作流程如下：

Step1：配置 SysCtrl.RCL.TRIM 及 SysCtrl.RCL.WaitCycle。

Step2：向 SysCtrl.CR3 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。

Step3：设置 SysCtrl.CR2.RCLEN 为 1，使能 RCL 振荡电路。

Step4：查询等待 SysCtrl.RCL.Stable 标志变为 1，RCL 输出稳定时钟。

Step5: 向 SysCtrl.CR3 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step6: 设置 SysCtrl.CR0.ClkSrc 为 2, 将系统时钟切换为 RCL。

Step7: 按 Flash 控制器章节流程设置 FLASH\_WAIT 为 0。

Step8: 向 SysCtrl.CR3 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step9: 设置 SysCtrl.CR2. xxxEN 为 0, 关闭不再使用的时钟源。

#### 5.2.4 从其它时钟切换到 RCHCLKD 示例

操作流程如下:

Step1: 配置 SysCtrl.RCH.TRIM 及 SysCtrl.RCH.DIV。

Step2: 向 SysCtrl.CR3 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step3: 设置 SysCtrl.CR2.RCHEN 为 1, 使能 RCH 振荡电路。

Step4: 查询等待 SysCtrl.RCH.Sable 标志变为 1, RCH 输出稳定时钟。

Step5: 根据当前时钟与目标 RCHCLKD 两者中较高的频率, 按 Flash 章节流程配置 FLASH\_WAIT 寄存器。

Step6: 向 SysCtrl.CR3 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step7: 设置 SysCtrl.CR0.ClkSrc 为 0, 将系统时钟切换为 RCHCLKD。

Step8: 根据 RCHCLKD 的频率, 按 Flash 章节流程配置 FLASH\_WAIT。

Step9: 向 SysCtrl.CR3 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

Step10: 设置 SysCtrl.CR2. xxxEN 为 0, 关闭不再使用的时钟源。

### 5.3 片内外设时钟控制

大多数片内外设都需要有时钟才能正常工作。当使能片内外设的配置时钟 (PCLK) 时，用户程序可以正常读写片内外设的寄存器；当使能片内外设的工作时钟时，片内外设可以正常工作。部分片内外设的工作时钟固定为 PCLK，其它片内外设的工作时钟可以由用户程序自行配置。

通过设置 `SysCtrl_Periph.ClkEnx` 的控制位为 1，即可使能相应片内外设的配置时钟。

Preliminary  
to be confirmed

## 5.4 中断唤醒控制

当处理器执行 WFI 指令进入休眠状态时，会停止执行指令。在休眠状态下发生了中断请求（更高优先级）且需要处理时，处理器就会被唤醒。

休眠状态下的处理器收到中断请求时的行为如下表所示：

PRIMASK 状态	WFI 行为	唤醒	ISR 执行
0	IRQ 优先级>当前等级	Y	Y
0	IRQ 优先级≤当前等级	N	N
1	IRQ 优先级>当前等级	Y	N
1	IRQ 优先级≤当前等级	N	N

### 5.4.1 从深度休眠模式唤醒后执行中断服务程序的方法

- Step1. 使能需要唤醒处理器的模块所对应的 NVIC
- Step2. 使能需要唤醒处理器的模块所对应的中断
- Step3. 设置 SCB->SCR.SLEEPDEEP 为 1
- Step4. 执行 WFI 指令以进入深度休眠模式
- Step5. 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程序

例程：

```
SCB->SCR |= 0x04;
while(1)
{
    __asm("WFI");
}
```

### 5.4.2 从深度休眠模式唤醒后不执行中断服务程序的方法

- Step1. 使能需要唤醒处理器的模块所对应的 NVIC
- Step2. 使能需要唤醒处理器的模块所对应的中断
- Step3. 设置 PRIMASK 为 1
- Step4. 设置 SCB->SCR.SLEEPDEEP 为 1

Step5. 执行 WFI 指令以进入深度休眠模式

Step6. 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

Step7. 清除中断标志，清除中断挂起状态

Step8. 执行用户定义的操作

例程：

```
__asm("CPSID I");      //Set PRIMASK
SCB_SCR |= 0x04;
while(1)
{
    __asm("WFI");
    M0P_TIMx->ICLR = 0x00; //Clear Int Flag
    NVIC_ClearPendingIRQ(TIMx_IRQn); //Clear Pending Flag
    ... //执行用户定义的操作
}
```

### 5.4.3 退出休眠

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，处理器可以尽可能多的处于休眠模式。

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. 使能需要唤醒处理器的模块所对应的 NVIC
2. 使能需要唤醒处理器的模块所对应的中断
3. 设置 SCB->SCR.SLEEPDEEP 为 1
4. 设置 SCB->SCR.SLEEPONEXIT 为 1
5. 执行 WFI 指令以进入深度休眠模式
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:

```
SCB_SCR |= 0x04;  
SCB_SCR |= 0x02;  
while(1)  
{  
    __asm("WFI");  
}
```

Preliminary  
to be confirmed

## 5.5 寄存器

基地址 0x40002000

寄存器	偏移地址	描述
CR0	0x000	系统控制寄存器0
CR1	0x004	系统控制寄存器1
CR2	0x008	系统控制寄存器2
CR3	0x00C	系统控制寄存器3
RCH	0x010	RCH 控制寄存器
RCL	0x018	RCL 控制寄存器
PeriClkEN0	0x030	片内外设时钟控制寄存器0
PeriClkEN1	0x034	片内外设时钟控制寄存器1
DebugActive	0x044	调试模式模块工作状态控制寄存器

表 5-1 系统控制寄存器表格

### 5.5.1 系统控制寄存器 0 (CR0)

偏移地址: 0x000

复位值: 0x0000 0000

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

位	标记	功能描述
31:8	Reserved	保留
7:6	PclkPrs	PCLK时钟来源选择 00: HCLK 01: HCLK / 2 10: HCLK / 4 11: HCLK / 8
5:3	HclkPrs	HCLK时钟来源选择 000: SystemClk 001: SystemClk / 2 010: SystemClk / 4 011: SystemClk / 8 100: SystemClk / 16 101: SystemClk / 32 110: SystemClk / 64 111: SystemClk / 128
2:0	ClkSrc	当CR2.EXTN为0时, SystemClk时钟来源选择 x0x: 内部高速时钟 RCHCLKD x1x: 内部低速时钟 RCL

注意:

- 每次改写 CR0 / CR1 / CR2 的值, 均需要先对 CR3 依次写入 0x5A5A、 0xA5A5。  
这样的步骤可有效防止对 CR0 / CR1 / CR2 寄存器的误操作。

## 5.5.2 系统控制寄存器 1 (CR1)

偏移地址: 0x004

复位值: 0x0000 0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								GTIM CFG	Reserved			RESIO	WakeU pClk	Lock UpEn	SWD IO
								RW				RW	RW	RW	RW

位	标记	功能描述
31:9	Reserved	内部保留位, 请勿更改其值。
8	GTIMCFG	GTIM / BTIM3-5定时器功能配置 0: GTIM有效, BTIM3/4/5无效 1: BTIM3/4/5有效, GTIM无效
7:4	Reserved	保留位
3	RESIO	RESET管脚功能配置 1: RESET管脚配置为IO输入管脚, 不再具有复位功能 0: RESET管脚配置为复位功能, 无IO输入功能
2	WakeUpClk	DeepSleep唤醒时, 系统时钟的来源配置 1: 切换系统时钟来源为RCHCLKD, 原系统时钟来源保持使能 0: 保持原系统时钟来源
1	LockUpEn	Cortex-M0+ LockUp 功能配置 0: 关闭 1: 使能 注: 使能该功能, 则CPU读到无效指令时会复位MCU。
0	SWDIO	SWD端口功能配置 0: PC07、PD01作为SWD端口, GPIO功能不可用。 1: PC07、PD01作为GPIO端口, SWD功能不可用。

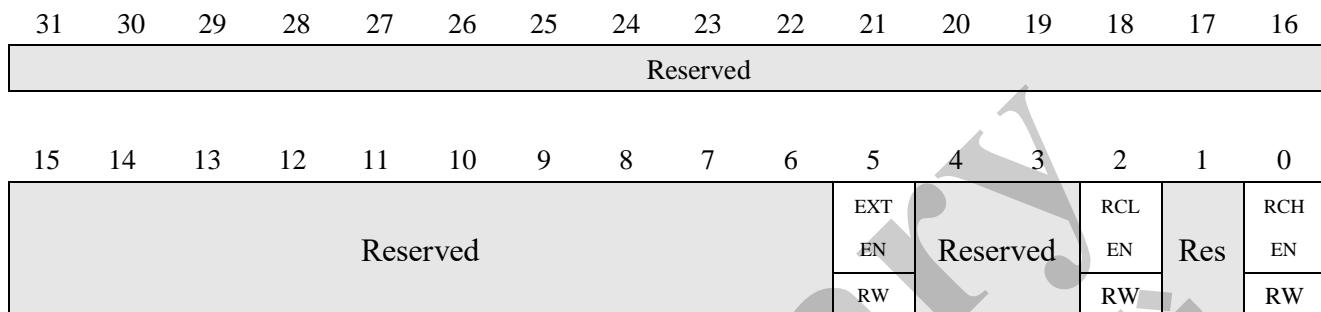
注意:

- 每次改写 CR0 / CR1 / CR2 的值, 均需要先对 CR3 依次写入 0x5A5A、0xA5A5。  
这样的步骤可有效防止对 CR0 / CR1 / CR2 寄存器的误操作。

### 5.5.3 系统控制寄存器 2 (CR2)

偏移地址: 0x008

复位值: 0x0000 0081



位	标记	功能描述
31:6	Reserved	保留
5	EXTEN	SystemClk时钟来源选择 0: SystemClk来源由CR0.ClkSrc决定 1: SystemClk来自PA1管脚输入的时钟 注意: 使能该控制位前, 应确保PA1已设为数字输入并已输入时钟信号
4:3	Reserved	保留
2	RCLEN	内部低速时钟 RCL 使能控制 0: 关闭 1: 使能 注: 当系统进入DeepSleep, 此低速时钟保持之前状态。
1	Reserved	保留
0	RCHEN	内部高速时钟 RCH 使能控制 0: 关闭 1: 使能 注: 当系统进入DeepSleep, 此高速时钟会自动关闭。

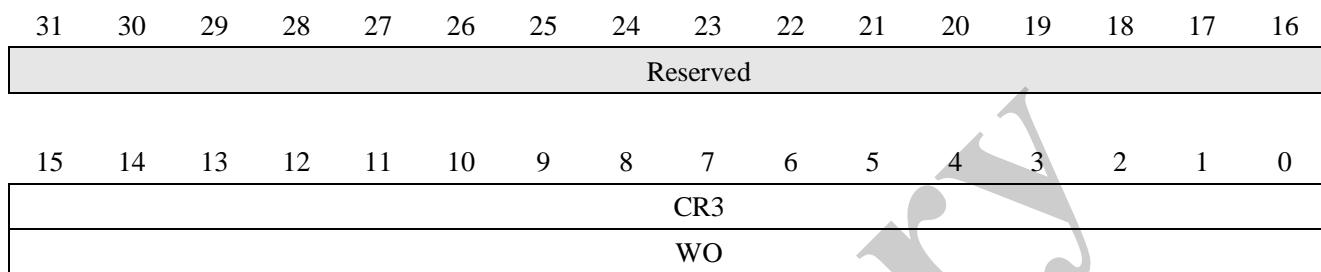
注意:

- 每次改写 CR0 / CR1 / CR2 的值, 均需要先对 CR3 依次写入 0x5A5A、0xA5A5。  
 这样的步骤可有效防止对 CR0 / CR1 / CR2 寄存器的误操作。

### 5.5.4 系统控制寄存器 3 (CR3)

偏移地址: 0x00C

复位值: 0x0000 0000



位	标记	功能描述
31:16	Reserved	保留位
15:0	CR3	寄存器 CR0 / CR1 / CR2 写保护寄存器, 对 CR3 依次写入 0x5A5A、0xA5A5, 则可解除 CR0 / CR1 / CR2 的写保护; 当对 CR0 / CR1 / CR2 进行了写操作, 则写保护自动复位。每次写数据到 CR0 / CR1 / CR2 之前, 均需要对 CR3 依次写入 0x5A5A、0xA5A5 序列。

### 5.5.5 RCH 控制寄存器 (RCH)

偏移地址: 0x010

复位值: 0x0000 0B47

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

位	标记	功能描述
31:16	Reserved	保留位
15	Stable	RCH 时钟稳定标志位 1: 代表 RCH 已经稳定, 可以被内部电路使用 0: 代表 RCH 未稳定, 不可以被内部电路使用
14:13	Reserved	保留位
12:9	DIV	RCHCLKD 输出频率与RCH振荡频率分频比配置 0000: RCHCLKD = RCH / 2 0001: RCHCLKD = RCH / 4 0010: RCHCLKD = RCH / 6 0011: RCHCLKD = RCH / 8 0100: RCHCLKD = RCH / 10 0101: RCHCLKD = RCH / 12 0110: RCHCLKD = RCH / 14 0111: RCHCLKD = RCH / 16 1000: RCHCLKD = RCH / 1
8:0	TRIM	时钟频率调整, 更改该寄存器的数值即可调整RCH的振荡频率。寄存器数值每增加1则RCH的振荡频率增加约0.2%, 总调整范围为 44 - 48MHz。 Flash中已保存了44.24MHz和48MHz的校准值, 将Flash内的校准值读出并写入SysCtrl.RCH.TRIM即可获得精准的频率。 44.24M校准值地址: 0x00100DF0 - 0x00100DF1 48.00M校准值地址: 0x00100DF2 - 0x00100DF3

### 5.5.6 RCL 控制寄存器 (RCL)

偏移地址: 0x018

复位值: 0x0000 033F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Stable	Res.	WaitCycle		TRIM										RW	
RO		RW												RW	

位	标记	功能描述
31:16	Reserved	保留位
15	Stable	RCL 时钟稳定标志位 1: 代表 RCL 已经稳定, 可以被内部电路使用 0: 代表 RCL 尚未稳定, 不可以被内部电路使用
14:12	Reserved	保留位
11:10	WaitCycle	内部低速时钟 RCL 稳定时间选择 11: 256个周期 10: 64个周期 01: 16个周期 00: 4个周期
9:0	TRIM	内部低速时钟频率调整, Flash中保存了2组频率的校准值。 将Flash内的校准值读出并写入RCL.TRIM即可获得精准的频率。 32.8K校准值地址: 0x00100DF4 - 0x00100DF5 38.4K校准值地址: 0x00100DF6 - 0x00100DF7

### 5.5.7 片内外设时钟控制寄存器 0 (PeriClkEn0)

复位值: 0x8080\_0000

偏移地址: 0x030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLASH	Reserved										CTRIM	Reserved			ADC
RW											RW				RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDT	Reserved		ATIM3	Res.	Res.	Reserved		SPI	Res.	I2C	LPUA	LPUA	Reserved		
RW			RW		RW			RW	Res.	RW	RW	RW			

位	标记	功能描述
31	FLASH	FLASH 模块配置时钟使能控制 1: FLASH寄存器可以读写 0: FLASH寄存器不能读写
30:22	Reserved	保留
21	CTRIM	CTRIM 模块配置时钟使能控制 1: 使能 0: 关闭
20:17	Reserved	保留
16	ADC	ADC 模块配置时钟及工作时钟使能控制 1: 使能 0: 关闭
15	IWDT	IWDT 模块配置时钟使能控制 1: 使能 0: 关闭
14:12	Reserved	保留位
11	ATIM3	ATIM3 模块配置时钟及工作时钟使能控制 1: 使能 0: 关闭
10:7	Reserved	保留位
6	SPI	SPI 模块配置时钟及工作时钟使能。 1: 使能 0: 关闭
5	Reserved	保留位
4	I2C	I2C 模块配置时钟及工作时钟使能控制 1: 使能 0: 关闭

3	LPUART1	LPUART1 模块配置时钟使能控制 1: 使能 0: 关闭
2	LPUART0	LPUART0 模块配置时钟使能控制 1: 使能 0: 关闭
1:0	Reserved	保留位

Preliminary  
to 立创商城

### 5.5.8 片内外设时钟控制寄存器 1 (PeriClkEn1)

复位值: 0x0000\_0000

偏移地址: 0x034

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	GTIM	Reserved		WWDT	Reserved		PD	PC	PB	PA					
	R/W			R/W			R/W	R/W	R/W	R/W					

位	标记	功能描述
31:15	Reserved	保留位
14	GTIM	GTIM / BTIM3 - 5模块配置时钟及工作时钟使能控制 1: 使能 0: 关闭
13:11	Reserved	保留位
10	WWDT	WWDT模块配置时钟及工作时钟使能控制 1: 使能 0: 关闭
9:4	Reserved	保留位
3	PD	PD 端口配置时钟时钟使能控制 1: 使能 0: 关闭
2	PC	PC端口配置时钟时钟使能控制 1: 使能 0: 关闭
1	PB	PB端口配置时钟时钟使能控制 1: 使能 0: 关闭
0	PA	PA端口配置时钟时钟使能控制 1: 使能 0: 关闭

### 5.5.9 调试模式模块工作状态控制寄存器 (DebugActive)

复位值 0x0001\_4980 (仅在 SWD 调试模式下, 此寄存器设置才起作用)

偏移地址: 0x4000\_2044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															WWDT
RW															RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	GTIM	Reserved	ATIM3	Reserved	Reserved	IWDT	Ctim	Reserved							
	RW		RW			RW	rw								

位	标记	功能描述
31:17	Reserved	保留
16	WWDT	调试时, WWDT 计数功能配置 1: 在 SWD 调试界面下, 暂停 WWDT 计数功能 0: 在 SWD 调试界面下, WWDT 正常计数功能
15	Reserved	保留
14	GTIM	调试时, GTIM / BTIM3 - 5 计数功能配置 1: 在 SWD 调试界面下, 暂停 GTIM / BTIM3 - BTIM5 计数功能 0: 在 SWD 调试界面下, GTIM / BTIM3 - BTIM5 正常计数功能
13:12	Reserved	保留
11	ATIM3	调试时, ATIM3 计数功能配置 1: 在 SWD 调试界面下, 暂停 ATIM3 计数功能 0: 在 SWD 调试界面下, ATIM3 正常计数功能
10:9	Reserved	保留
8	IWDT	调试时, IWDT 计数功能配置 1: 在 SWD 调试界面下, 暂停 IWDT 计数功能 0: 在 SWD 调试界面下, IWDT 正常计数功能
7	CTRIM	调试时, CTRIM 计数功能配置 1: 在 SWD 调试界面下, 暂停 CTRIM 计数功能 0: 在 SWD 调试界面下, CTRIM 正常计数功能
6:0	Reserved	保留

## 6 复位控制器 (RESET)

### 6.1 复位控制器介绍

本产品具有 7 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，程序会从复位向量处开始执行。

- 数字区域上电掉电复位 POR
- 外部 Reset PAD，低电平为复位信号
- IWDT 复位
- WWDT 复位
- LVD 低电压复位
- Cortex-M0+ SYSRESETREQ 软件复位
- Cortex-M0+ LOCKUP 硬件复位

每个复位源由相应的复位标志进行指示；复位完成时查询 ResetFlag 寄存器即获取上一次的复位来源。每当完成对 ResetFlag 寄存器的查询后，应将该寄存器清零，以便下一次复位时该寄存器能正常指示其复位源。

下图描述各区域的复位来源。

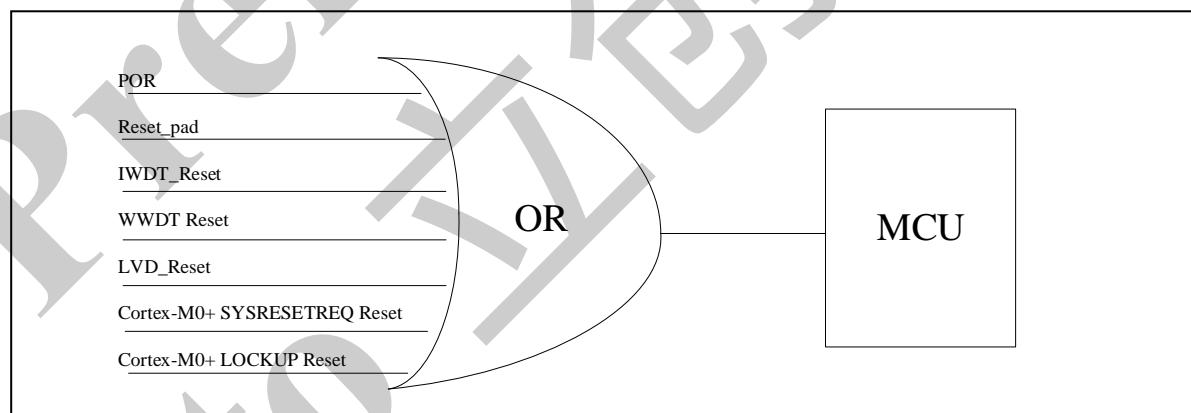


图 6-1 复位来源示意图

#### 6.1.1 上电下电复位 POR

本产品有两个供电区域：VCC 区域、Vcore 区域。所有的模拟模块及 IO 工作于 VCC 区域；其它模块工作于 Vcore 区域。

VCC 区域上电时，当 VCC 电压低于 POR 阈值电压时(典型值为 1.65V)，会产生 POR5V

信号；VCC 区域下电时，当 VCC 电压低于 BOR 阈值电压时（典型值为 1.5V），会产生 POR5V 信号。

Vcore 区域上电时，当 Vcore 电压低于 POR 阈值电压时，会产生 POR15V 信号；Vcore 区域下电时，当 Vcore 电压低于 BOR 阈值电压时，会产生 POR15V 信号。

POR5V 信号和 POR15V 信号均会将芯片的寄存器复位到初始化状态。

### 6.1.2 外部复位管脚复位

当外部复位管脚检测到低电平时会产生一个系统复位。该复位管脚已内置上拉电阻，并集成了一个毛刺过滤电路。毛刺过滤电路会过滤小于 20uS（典型值）的毛刺信号，因此，加到复位管脚上的低电平信号必须大于 20uS，才能确保芯片可靠复位。

### 6.1.3 WDT 复位

看门狗复位，请参看 IWDT / WWDT 章节说明。

### 6.1.4 LVD 低电压复位

LVD 复位，请参考 LVD 章节说明。

### 6.1.5 Cortex-M0+ SYSRESETREQ 复位

Cortex-M0+ 软件复位

### 6.1.6 Cortex-M0+ LOCKUP 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。

## 6.2 寄存器

### 6.2.1 复位标识寄存器 (Reset\_Flag)

复位值: 0x0000 0003

地址: 0x40002040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RSTB	SysReq	Lockup	WWDT	IWDT	LVD	Veore	Vcc
								RW0	RW0	RW0	RW0	RW0	RW0	RW0	RW0

位	标记	功能描述
31:8	Reserved	保留位
7	RSTB	RESETB 端口复位标志 1: 已发生端口复位 0: 未发生端口复位 写0清除, 写1无效
6	SysReq	Cortex-M0+ CPU 软件复位标志 1: 已发生 Cortex-M0+ CPU 软件复位 0: 未发生 Cortex-M0+ CPU 软件复位 写0清除, 写1无效
5	Lockup	Cortex-M0+ CPU Lockup 复位标志 1: 已发生 Cortex-M0+ CPU Lockup 复位 0: 未发生 Cortex-M0+ CPU Lockup 复位 写0清除, 写1无效
4	WWDT	WWDT 复位标志, 需要软件初始化及清除 1: 已发生 WWDT 复位 0: 未发生 WWDT 复位 写0清除, 写1无效
3	IWDT	IWDT 复位标志 1: 已发生 IWDT 复位 0: 未发生 IWDT 复位 写0清除, 写1无效
2	LVD	LVD 复位标志 1: 已发生 LVD 复位 0: 未发生 LVD 复位 写0清除, 写1无效

1	Vcore	Vcore域复位标志 1: Vcore域已发生复位 0: Vcore域发生未复位 写0清除, 写1无效
0	Vcc	Vcc电源域复位标志 1: VCC电源域已发生复位 0: VCC电源域未发生复位 写0清除, 写1无效

注意: 上电复位时复位值为 0x03。程序判断本次复位源后应该清除该寄存器, 以便下次发生复位后判断复位源。

Preliminary  
to be confirmed

## 6.2.2 片内外设复位控制寄存器 0 (PeriReset0)

复位值: 0x0021\_0ADC

地址: 0x40002038

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										CTRIM	Reserved			ADC	
										RW				RW	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			ATIM3	Res.			SPI	Res.	I2C	LPUA	LPUA	Res.			
									RW	RT1	RT0				

位	标记	功能描述
31:22	Res.	保留位
21	CTrim	CTRIM 模块复位控制 1: 模块正常工作 0: 模块处于复位状态
20:17	Res.	保留位
16	ADC	ADC 模块复位控制 1: 模块正常工作 0: 模块处于复位状态
15:12	Res.	保留位
11	ATIM3	ATIM3 模块复位控制 1: 模块正常工作 0: 模块处于复位状态
10:7	Res.	保留位
6	SPI	SPI 模块复位控制 1: 模块正常工作 0: 模块处于复位状态
5	Res.	保留位
4	I2C	I2C 模块复位控制 1: 模块正常工作 0: 模块处于复位状态
3	LPUART1	LPUART1 模块复位控制 1: 模块正常工作 0: 模块处于复位状态
2	LPUART0	LPUART0 模块复位控制 1: 模块正常工作

		0: 模块处于复位状态
1:0	Res.	保留位

Preliminary  
to 立创商城

### 6.2.3 片内外设复位控制寄存器 1 (PeriReset1)

复位值: 0x0000 440F

地址: 0x4000203C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	GTIM	Reserved		WWDT	Reserved				PD	PC	PB	PA	R/W	R/W	R/W	R/W

位	标记	功能描述
31:15	Reseved	保留位
14	GTIM	GTIM / BTIM3 - 5模块复位控制 1: 模块正常工作 0: 模块处于复位状态
13:11	Reseved	保留位
10:	WWDT	WWDT模块复位控制 1: 模块正常工作 0: 模块处于复位状态
9:4	Reseved	保留位
3	PD	PD端口复位控制 1: 模块正常工作 0: 模块处于复位状态
2	PC	PC端口复位控制 1: 模块正常工作 0: 模块处于复位状态
1	PB	PB端口复位控制 1: 模块正常工作 0: 模块处于复位状态
0	PA	PA端口复位控制 1: 模块正常工作 0: 模块处于复位状态

## 7 中断控制器 (NVIC)

### 7.1 概述

Cortex-M0+ 处理器内置了嵌套向量中断控制器(NVIC)，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。

每个异常源都有一个单独的异常编号，每种异常类型都有对应的优先级，有些异常的优先级是固定的，而有些则是可编程的。具体如下表所示：

异常编号	异常类型	优先级	描述
1	复位	-3 (最高)	复位
2	NMI	-2	不可屏蔽中断 (在本系统中没有使用)
3	硬件错误	-1	错误处理异常
4-10	保留	NA	...
11	SVC	可编程	通过SVC指令调用管理程序
12-13	保留	NA	...
14	PendSV	可编程	系统服务的可挂起请求
15	SysTick	可编程	SysTick定时器
16	中断#0	可编程	外部中断#0
17	中断#1	可编程	外部中断#1
...	...	...	...
47	中断#31	可编程	外部中断#31

表 7-1 Cortex-M0+ 处理器中断一览

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。同时，本章节只讨论处理器内核中 NVIC 的中断处理机制，外设模块自身的中断产生机制不在这里展开讨论。

## 7.2 中断优先级

每一个外部中断都对应一个优先级寄存器，每个优先级都是 2 位宽，并且使用中断优先级寄存器的最高两位，每个寄存器占 1 个字节（8 位）。在这个设定下，可以使用的优先级为 0x00（最高）、0x40、0x80 和 0xc0（最低）。

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
已使用	未使用，读出为 0						

图 7-1 只使用了高两位的优先级寄存器

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。

## 7.3 中断向量表

当 Cortex-M0+处理器要处理中断服务请求时, 它需要首先确定异常处理的起始地址, 所需的信息叫做向量表, 如图 7-2 所示。向量表存储在存储器空间的开始位置, 包含了系统中可用异常 (中断) 的异常 (中断) 向量, 以及主栈指针 (MSP) 的初始值。



图 7-2 中断向量表

其中, 中断向量的存储顺序同中断编号一致, 由于每个向量都是 1 个字 (4 字节), 中断向量的地址为中断编号乘 4, 每个中断向量都是中断处理的起始地址。

## 7.4 中断输入和挂起行为

Cortex-M0+ 处理器的 NVIC 模块中，每一个中断输入都对应着一个挂起状态寄存器，且每个寄存器只有 1 位，用于保存中断请求，而不管这个请求有没有得到确认。当处理器开始处理这个中断时，硬件将会自动清除挂起状态位。

本系统的外设使用电平触发中断输出，当中断事件发生时，由于外设连接到了 NVIC 上，中断信号会得到确认。在处理器执行中断服务并且清除外设的中断信号以前，该信号会保持高电平。在 NVIC 内部，当检测到有中断发生时，该中断的挂起状态会被置位，当处理器接收该中断并且开始执行中断服务程序后，挂起状态就会被清除。该过程如图 7-3 所示：

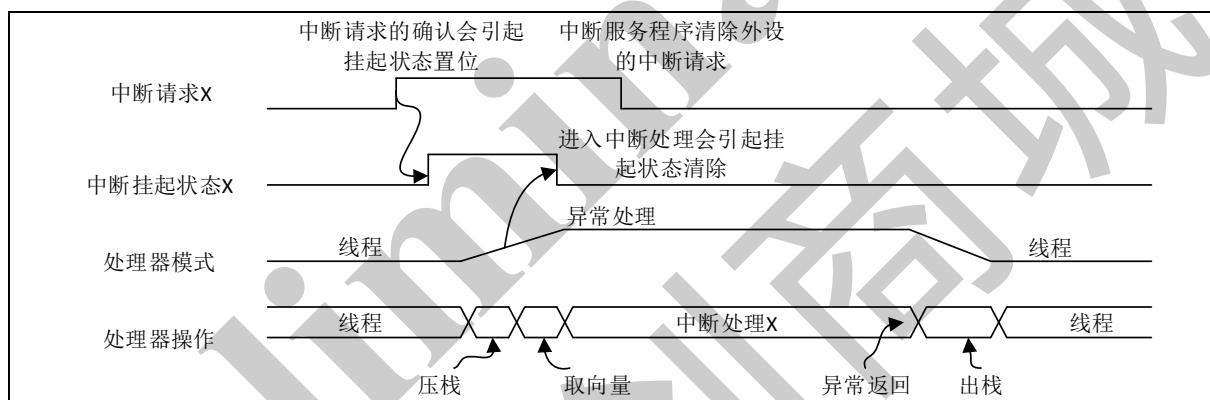


图 7-3 中断激活和挂起状态

如果中断请求没有立即执行，并且在确认之前被软件清除了，这样处理器会忽略掉本次请求，并且不会执行中断处理。可以通过写 NVIC\_ICPR 寄存器来清除中断挂起状态，这种处理在设置外设时非常有用，因为在设置以前，该外设可能已经产生了一个中断请求。

如果在软件清除挂起状态时，外设仍然保持着中断请求，挂起状态还会立即生成。该过程如图 7-4 所示：

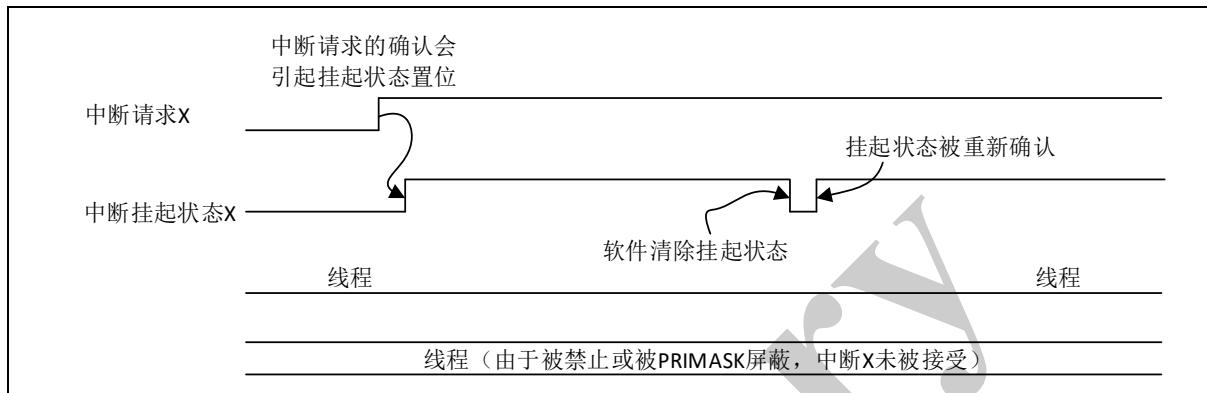


图 7-4 中断挂起状态被清除然后被重新确认

如果外设产生的中断请求在异常处理时没有被清除，异常返回后挂起状态就会被又一次激活，这样中断服务程序会再次执行。该过程如图 7-5 所示：

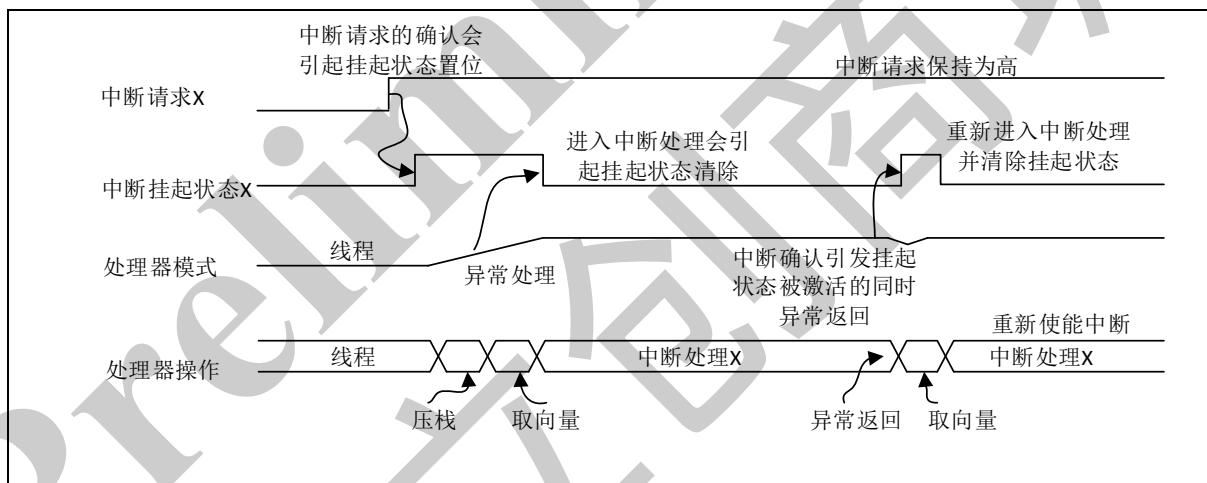


图 7-5 中断退出时若中断请求保持高电平就会引起中断处理的再次执行

如果在中断服务程序执行的过程中产生外设中断请求，该请求会被当作新的中断请求，并且在本次中断退出后，还会引起中断服务程序的再次执行。该过程如图 7-6 所示：

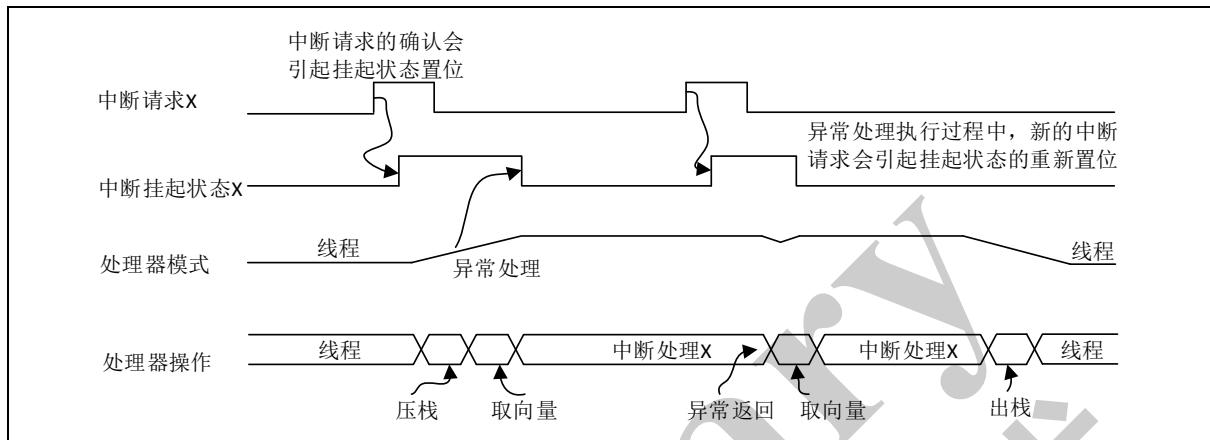


图 7-6 中断处理中产生的中断挂起也可以被确认

Preliminary  
to be confirmed

## 7.5 中断等待

通常情况下，NVIC 的中断等待时间为 16 个周期。这个等待时间从中断确认的处理器时钟周期开始，一直到中断处理开始执行结束。计算中断等待需要具备以下前提：

- 该中断使能并且没有被 PRIMASK 或是其他正在执行的异常处理所屏蔽。
- 存储器系统没有任何等待状态，在中断处理、压栈、取向量或者中断处理开始时的取指都会用到总线传输，如果存储器系统需要等待，那么发生总线传输时产生的等待状态则可能使得中断延迟。

下面的几种情况可能会导致不同的中断等待：

- 中断的末尾连锁，如果中断返回时产生了另外一个中断请求，处理器就会跳过出栈和压栈过程，这样就减少了中断等待时间。
- 延迟到达，如果中断发生时，另外一个低优先级的中断正在进行压栈处理，由于延迟到达机制的存在，高优先级的中断会首先执行，这样也会导致高优先级中断的等待时间减小。

## 7.6 中断源

因为 Cortex-M0+ 处理器的 NVIC 支持最多 32 个外部中断，而在本系统中，外部中断源为 15 个，所以部分外部中断复用在同一个 NVIC 中断输入上，其中 NMI (不可屏蔽中断) 并没有使用。本系统所有外部中断源和 NVIC 中断输入的对应关系如下表所示：

NVIC中断输入	外部中断源	Active模式	Sleep模式	DeepSleep模式
中断#0	PORTA	✓	✓	✓
中断#1	PORTB	✓	✓	✓
中断#2	PORTC	✓	✓	✓
中断#3	PORTD	✓	✓	✓
中断#4	-	-	-	-
中断#5	ATIM3	✓	✓	-
中断#6 - #7	-	-	-	-
中断#8	LPUART0	✓	✓	✓
中断#9	LPUART1	✓	✓	✓
中断#10	SPI	✓	✓	-
中断#11	-	-	-	-
中断#12	I2C	✓	✓	-
中断#13 - #14	-	-	-	-
中断#15	GTIM / BTIM3 - 5	✓	✓	-
中断#16 - #21	-	-	-	-
中断#22	IWDT	✓	✓	✓
	WWDT	✓	✓	-
中断#23	-	-	-	-
中断#24	ADC	✓	✓	-
中断#25 - #27	-	-	-	-
中断#28	LVD	✓	✓	-
中断#29	-	-	-	-
中断#30	FLASH	✓	✓	-
中断#31	CTRIM	✓	✓	✓

表 7-2 外部中断与 NVIC 中断输入对应关系

注意：

- 由于某些模块中断被复用于同一个 IRQ 中断源，当 CPU 进入该中断操作时，必须先判断是哪个模块产生的中断，再进行相对应的中断操作。

## 7.7 中断结构图

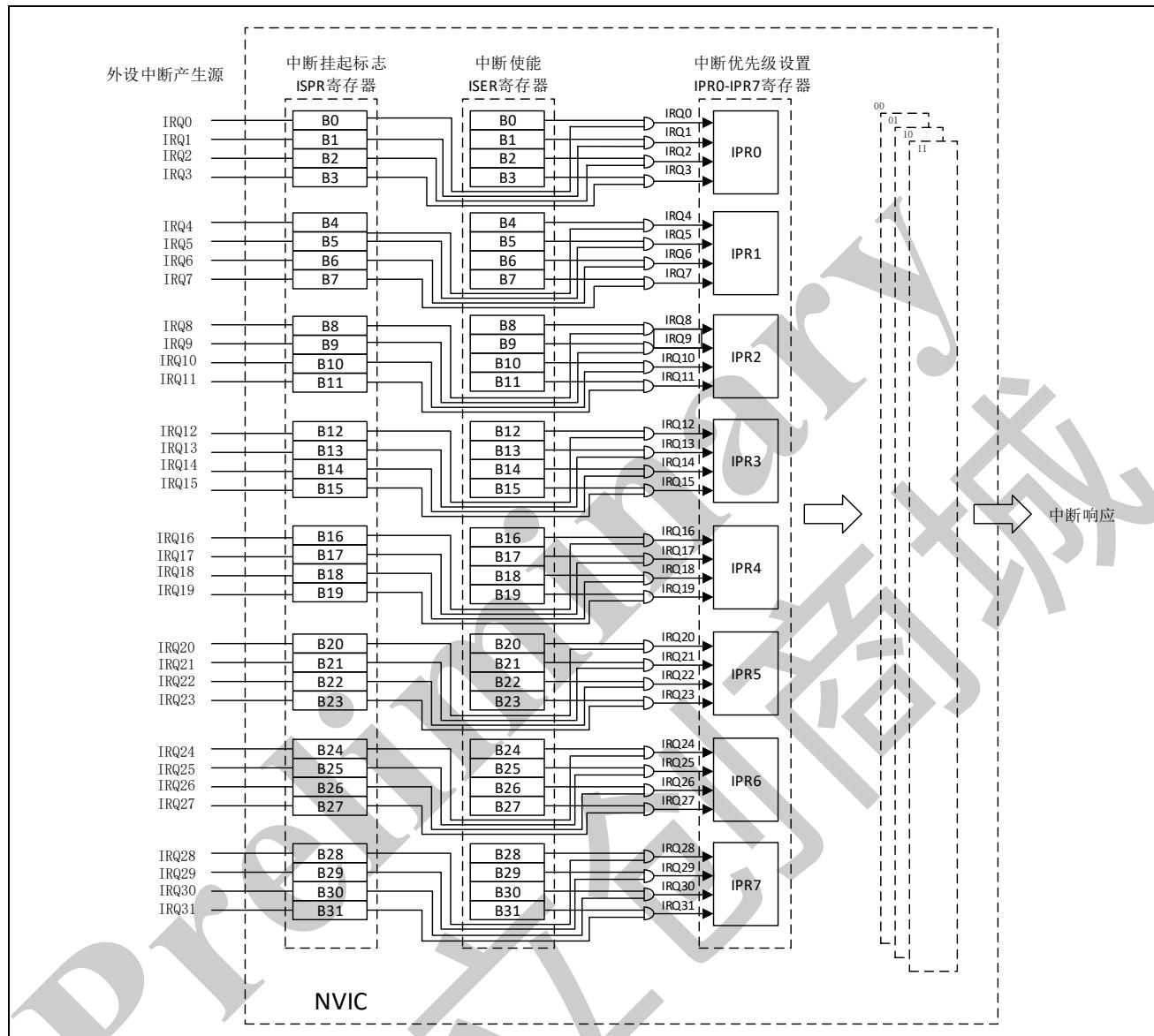


图 7-7 中断结构图

本系统的中断结构框图如图 7-7 所示。需要注意几点：

- 外设中断源各自的中断使能没有在图中标出，这里只包括外设中断产生之后的中断信号逻辑框图。
- IRQ 有多于 1 个外设中断输入复用，必须分别读取这些外设的中断标志位来判定是哪个外设的中断。
- 如果外设中断源有高电平产生，无论 NVIC 中断使能寄存器 SCS\_SETENA 置位与否，中断挂起寄存器 SCS\_SEPEND 都会被置位，表示相应的外设中断源有中断产生。
- 只有中断使能寄存器 NVIC\_ISER 置位，相应的中断 IRQ 才会给处理器相应，执行

相应的中断程序。

- 在中断程序中必须清除外设中断源高电平中断信号，中断挂起寄存器 NVIC\_ISPR 由硬件自动清除。
- 中断优先级寄存器 NVIC\_IPR0- NVIC\_IPR7 设置了 32 个中断源的优先级，00 优先级最高，11 优先级最低。当优先级相同时，优先级由中断编号决定，编号越小优先级越高。

Preliminary  
to 立创商城

## 7.8 软件基本操作

### 7.8.1 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。

### 7.8.2 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使能位。这样就有了 32 位的中断使能寄存器 NVIC\_ISER 和 32 位的中断请求清除寄存器 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，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。

### 7.8.3 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合法。

当处理器开始进入中断处理，硬件会自动清除挂起状态。

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。

### 7.8.4 NVIC 中断优先级

设置 NVIC\_IPR0- NVIC\_IPR7 寄存器决定 SCS\_IRQ0- SCS\_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。

### 7.8.5 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 SCS\_PRIMASK 实现。特殊寄存器 SCS\_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 SCS\_PRIMASK 设置为 1 后，处理器的当前优先级就降到了 0（可配值的最高优先级）。

可以通过多种方法编程 SCS\_PRIMASK 寄存器，使用汇编语言，可以利用 MSR 指令在设置和清除 SCS\_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
```

## 7.9 寄存器

基地址: 0xE000 E000

寄存器	偏移地址	描述
NVIC_ISET	0x100	中断使能设置寄存器
NVIC_ICER	0x180	中断使能清除寄存器
NVIC_ISPR	0x200	中断挂起状态设置寄存器
NVIC_ICPR	0x280	中断挂起状态清除寄存器
NVIC_IPR0	0x400	中断优先级寄存器0
NVIC_IPR1	0x404	中断优先级寄存器1
NVIC_IPR2	0x408	中断优先级寄存器2
NVIC_IPR3	0x40C	中断优先级寄存器3
NVIC_IPR4	0x410	中断优先级寄存器4
NVIC_IPR5	0x414	中断优先级寄存器5
NVIC_IPR6	0x418	中断优先级寄存器6
NVIC_IPR7	0x41C	中断优先级寄存器7
PRIMASK	-	中断屏蔽特殊寄存器

### 7.9.1 中断使能设置寄存器 (NVIC\_ISET)

偏移地址: 0x100

复位值: 0x0000 0000

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

位	标记	功能描述
31:0	SETENA [31:0]	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

## 7.9.2 中断使能清除寄存器 (NVIC\_ICER)

偏移地址: 0x180

复位值: 0x0000 0000

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

位	标记	功能描述
31:0	CLRENA	清除使能中断#0到中断#31; 写“1”清零，写“0”无效

## 7.9.3 中断挂起状态设置寄存器 (NVIC\_ISPR)

偏移地址: 0x200

复位值: 0x0000 0000

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

位	标记	功能描述
31:0	SETPEND	设置中断#0到中断#31的挂起状态; 写“1”置位, 写“0”无效 [0]: IRQ0 [1]: IRQ1 [2]: IRQ2 ..... [31]: IRQ31

### 7.9.4 中断挂起状态清除寄存器 (NVIC\_ICPDR)

偏移地址: 0x280

复位值: 0x0000 0000

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

位	标记	描述
31:0	CLRPEND	清除中断#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

### 7.9.5 中断优先级寄存器 (NVIC\_IPR0)

偏移地址: 0x400

复位值: 0x0000 0000

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

位	标记	功能描述
31:0	IPR0[31:0]	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 优先级最高, 11 优先级最低

### 7.9.6 中断优先级寄存器 (NVIC\_IPR1)

偏移地址: 0x404

复位值: 0x0000 0000

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

位	标记	功能描述
31:0	IPR1[31:0]	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 优先级最高, 11 优先级最低

### 7.9.7 中断优先级寄存器 (NVIC\_IPR2)

偏移地址: 0x408

复位值: 0x0000 0000

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

位	标记	功能描述
31:0	IPR2[31:0]	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

### 7.9.8 中断优先级寄存器 (NVIC\_IPR3)

偏移地址: 0x40C

复位值: 0x0000 0000

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

位	标记	功能描述
31:0	IPR3[31:0]	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

### 7.9.9 中断优先级寄存器 (NVIC\_IPR4)

偏移地址: 0x410

复位值: 0x0000 0000

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

位	标记	功能描述
31:0	IPR4[31:0]	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

### 7.9.10 中断优先级寄存器 (NVIC\_IPR5)

偏移地址: 0x414

复位值: 0x0000 0000

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

位	标记	功能描述
31:0	IPR5[31:0]	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

### 7.9.11 中断优先级寄存器 (NVIC\_IPR6)

偏移地址: 0x418

复位值: 0x0000 0000

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

位	标记	功能描述
31:0	IPR6[31:0]	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

### 7.9.12 中断优先级寄存器 (NVIC\_IPR7)

偏移地址: 0x41C

复位值: 0x0000 0000

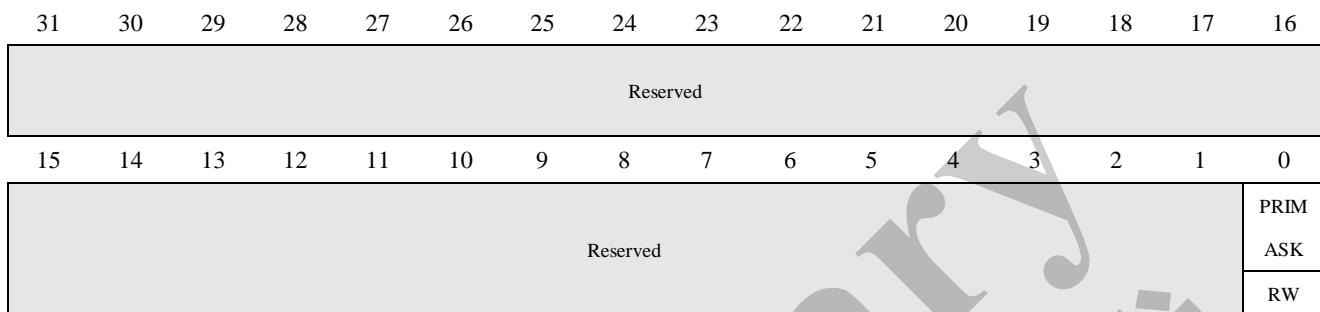
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPR7[31:16]															
RW															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPR7[15:0]															
RW															

位	标记	功能描述
31:0	IPR7[31:0]	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

### 7.9.13 中断屏蔽特殊寄存器 (PRIMASK)

偏移地址: ---

复位值: 0x0000 0000



位	标记	功能描述
31:1	Reserved	
0	PRIMASK	<p>置位后，除了 NMI 和硬件错误异常外的其他中断都会被屏蔽掉</p> <p>清零后，所有异常和中断都不会被屏蔽掉</p> <p>该特殊寄存器需要通过 MSR 和 MRS 特殊寄存器操作指令访问，也可以用改变处理器状态指令 CPS 访问。在处理对时间敏感的应用时，需要操作 PRIMASK 寄存器。</p>

## 8 通用输入输出端口控制器 (GPIO)

### 8.1 简介

GPIO 是 MCU 最基本的输入输出通道。本产品包含四组输入输出端口，其中 PA 为 5 位，PB 为 6 位，PC 为 5 位，PD 为 6 位。实际可用的 GPIO 端口数量请参考相应的规格书以获得更多信息。

每个管脚都有相关的控制和配置寄存器，以满足特定应用的需求。通过配置相应的寄存器，管脚可以被配置为以下三类工作模式：模拟外设的输入输出、数字外设的输入输出、GPIO 的输入输出。

### 8.2 主要特性

- 功能复用：每只管脚可配置为多种数字或模拟功能，PA00 除外
- 输入模式：模拟输入、高阻输入、上拉输入
- 输出模式：模拟输出、推挽输出、开漏输出；可使能上拉
- 信号输入：输入信号到达数据输入寄存器 (GPIOx\_IN) 及片内外设
- 信号输出：输出信号来源为数据输出寄存器 (GPIOx\_OUT) 或片内外设
- 原子操作：位操作寄存器 (GPIOx\_BSRR、GPIOx\_BRR) 实现 IO 操作原子化
- 中断类型：上升沿中断、下降沿中断、高电平中断、低电平中断
- 辅助功能：输出各种内部内钟、片内外设输入信号重定向
- 双倍驱动：TSSOP20 和 QFN20 的 PB04-05、PC03-04 具有双倍驱动能力

## 8.3 功能描述

### 8.3.1 端口电路框图

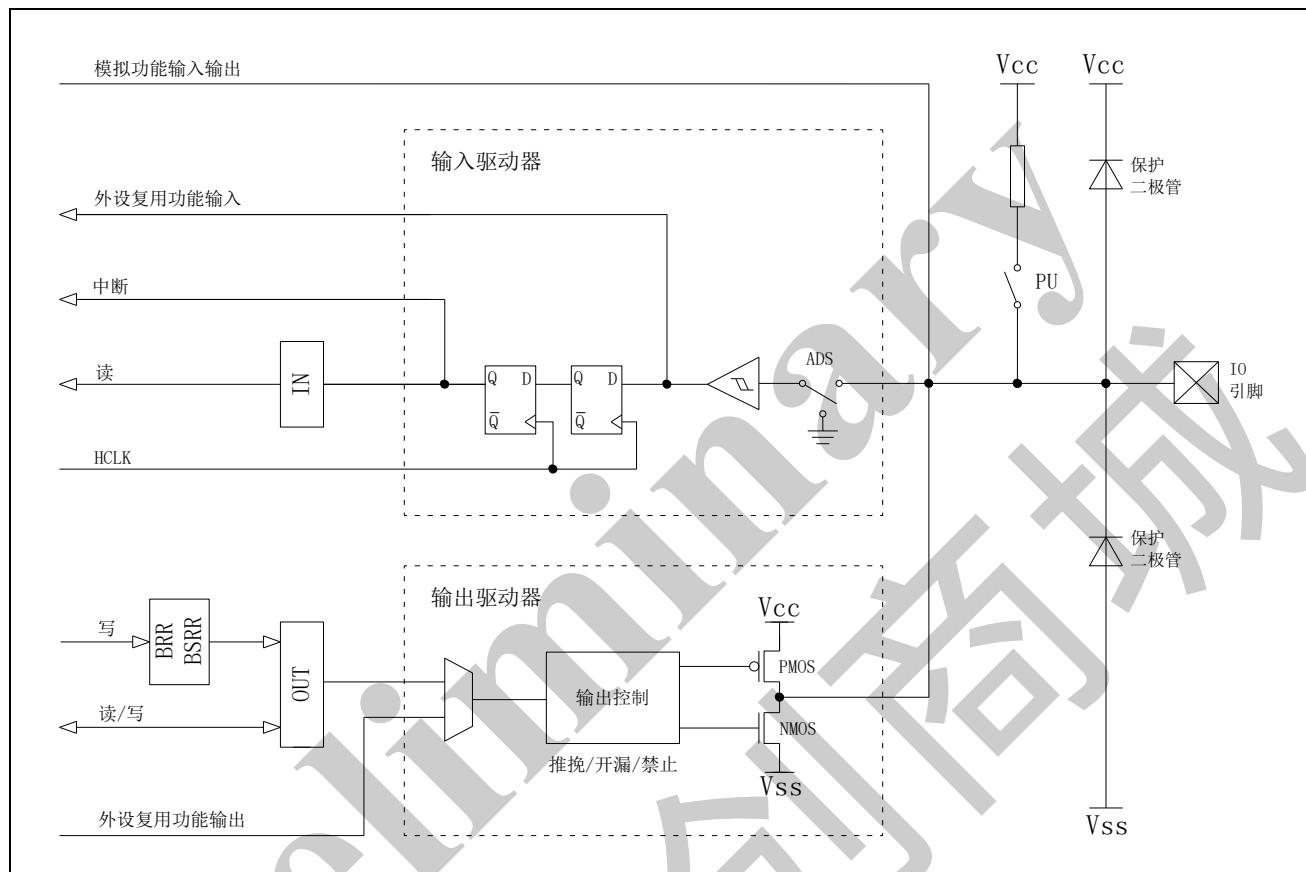


图 8-1 通用端口电路示意图

### 8.3.2 端口模式配置

通过控制寄存器，可以将端口配置为多种工作模式，其真值表如下方所示：

管脚模式	IN	ADS	AFRL	DIR	Open Drain	OUT	BRR.BR	BSRR.BR	BSRR.BS	PU
模拟	0	1	-	-	-	-	-	-	-	-
高阻输入	X	0	0	1	-	-	-	-	-	0
上拉输入	1	0	0	1	-	-	-	-	-	1
高阻输出	X	0	0	0	1	1	0	0	0	0
高阻输出	X	0	0	0	1	-	0	0	1	0
高阻输出	X	0	0	0	1	-	0	1	1	0
上拉输出	1	0	0	0	1	1	0	0	0	1
上拉输出	1	0	0	0	1	-	0	0	1	1
上拉输出	1	0	0	0	1	-	0	1	1	1
输出0	0	0	0	0	1	0	0	0	0	-
输出0	0	0	0	0	1	-	1	0	0	-
输出0	0	0	0	0	1	-	0	1	0	-
输出0	0	0	0	0	0	0	0	0	0	-
输出0	0	0	0	0	0	-	1	0	0	-
输出0	0	0	0	0	0	-	0	1	0	-
输出1	0	0	0	0	0	1	0	0	0	-
输出1	0	0	0	0	0	-	0	0	1	-
输出1	0	0	0	0	0	-	0	1	1	-
外设输出	X	0	>0	0	0	-	-	-	-	-
外设上拉输入	X	0	>0	1	0	-	-	-	-	1

注：-代表任意值；X 代表不确定

表 8-1 端口功能配置表

### 8.3.3 端口复位状态

在复位期间及复位完成未配置 GPIO 寄存器之前，大多数 I/O 端口被配置为模拟输入无上下拉状态；其它管脚的状态如下：

- SWCLK( PD01 )： 数字输入+上拉
- SWDIO( PC07)： 数字输入+上拉
- PC06： 数字输入无上拉

### 8.3.4 端口模拟功能

当需要使用模拟外设（ADC、LVD……）的相应功能时，需要配置端口为模拟功能。

实现方法为设置 GPIOx\_ADS 寄存器相应的位为 1，使片内模拟外设连接到该端口。

在该配置下，端口具有如下特性：

- 模拟外设连接到端口
- 端口数字功能关闭：不能输出电平；读取管脚电平恒为 0；无内置上拉及下拉

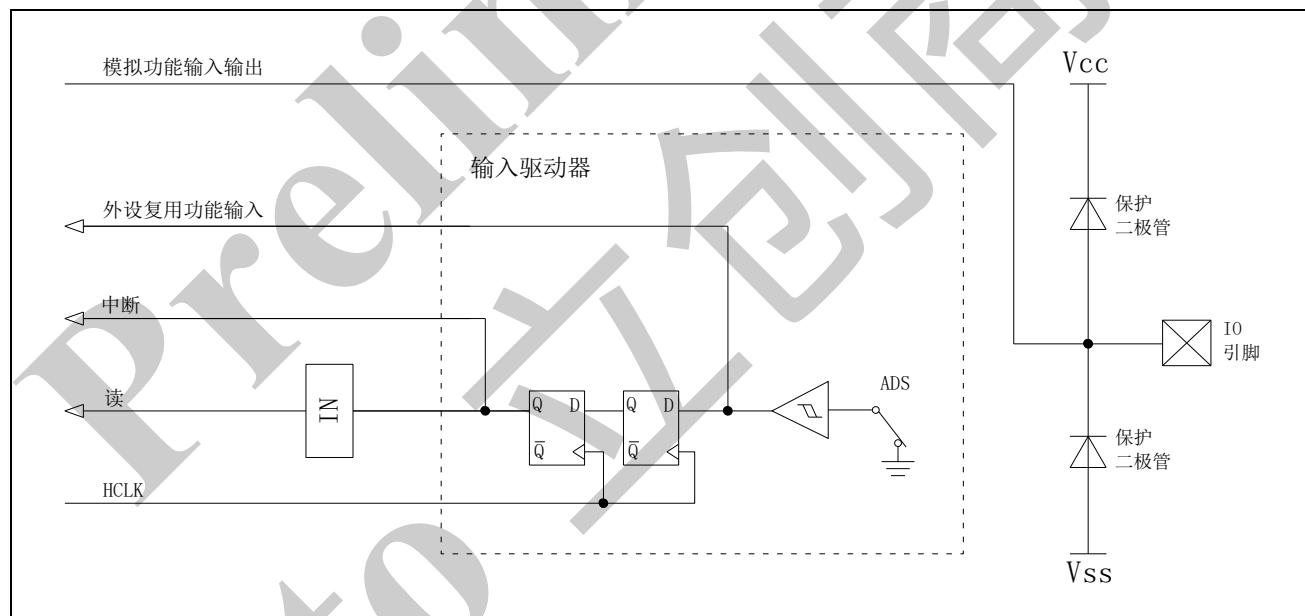


图 8-2 高阻-模拟配置示意图

### 8.3.5 端口通用输入输出功能

当需要使用端口的通用输入输出功能时，需要配置端口为数字功能并设置 AFRL 的值为 0。实现方法为设置 GPIOx\_ADS 寄存器相应的位为 0，设置 GPIOx\_AFRL 寄存器相应的位为 0。在该配置下，端口具有如下特性：

- 输入模式：可使能上拉
- 推挽输出：可使能上拉；支持位操作
- 开漏输出：可使能上拉；支持位操作

每个 GPIO 具有 4 个控制寄存器，其功能描述如下：

- GPIOx\_ADS      寄存器可配置端口为数字端口
- GPIOx\_DIR      寄存器可配置端口为输入或输出
- GPIOx\_PU      寄存器可使能端口的上拉
- GPIOx\_OpenDrain      寄存器可配置端口的输出类型为推挽或开漏

每个 GPIO 具有 2 个数据寄存器，其功能描述如下：

- GPIOx\_IN      寄存器存储通过 I/O 输入的数据，只可进行读操作
- GPIOx\_OUT      寄存器存储待输出数据，可对其进行读/写访问

**注意：对 *GPIOx\_OUT* 寄存器进行【读-修改-写】的过程不是原子操作，不建议采用该方式修改 *GPIOx\_OUT* 寄存器的值。**

每个 GPIO 有 2 个位操作寄存器，其功能描述如下：

- GPIOx\_BSRR      寄存器的 BS 位可将 GPIOx\_OUT 相应的比特置 1  
    寄存器的 BR 位可将 GPIOx\_OUT 相应的比特清 0
- GPIOx\_BRR      寄存器位可将 GPIOx\_OUT 相应的比特清 0

**注意：通过位操作寄存器对 *GPIOx\_OUT* 的值进行修改的过程为原子操作，建议采用位操作的方式修改 *GPIOx\_OUT* 寄存器的值。**

### 8.3.5.1 输入模式

当 GPIO 被配置为输入模式下，其简化框图如下所示：

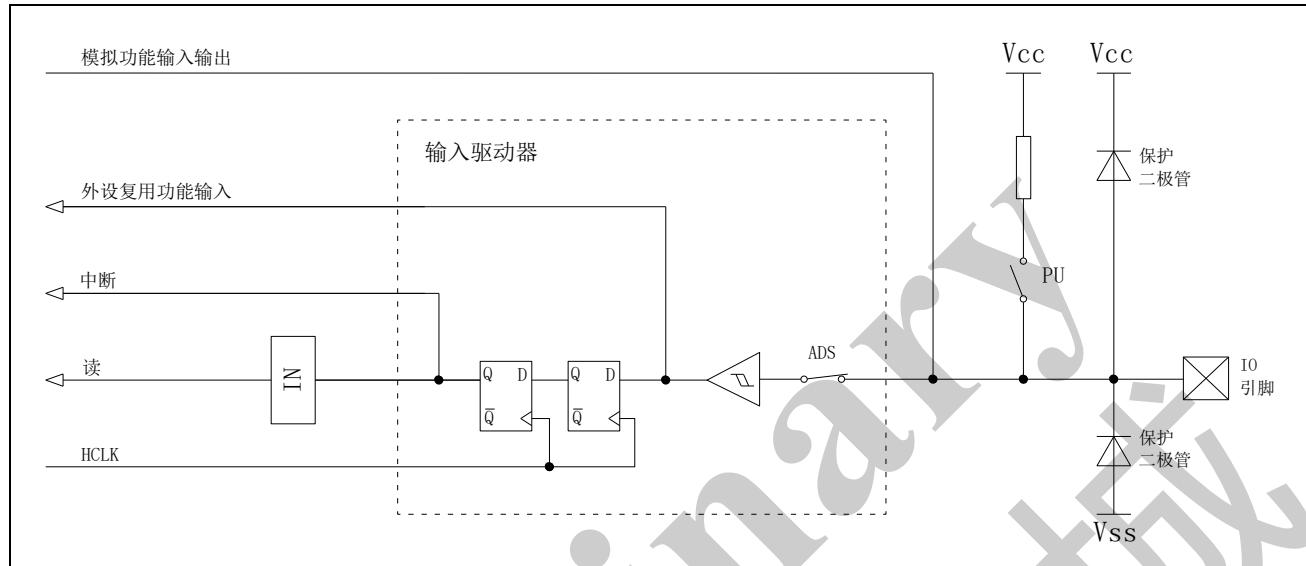


图 8-3 输入悬空/上拉来配置示意图

每一个端口可以通过读取 `GPIOx_IN` 寄存器来获得端口管脚电平。如图 8-4 所示，`GPIOx_IN` 寄存器的各个位与其前面的锁存器组成了一个同步器从而避免了系统时钟状态发生改变时在短时间内管脚电平变化而造成的信号不稳定，但是同时也引入了延迟。读取端口管脚数据的同步图如下：

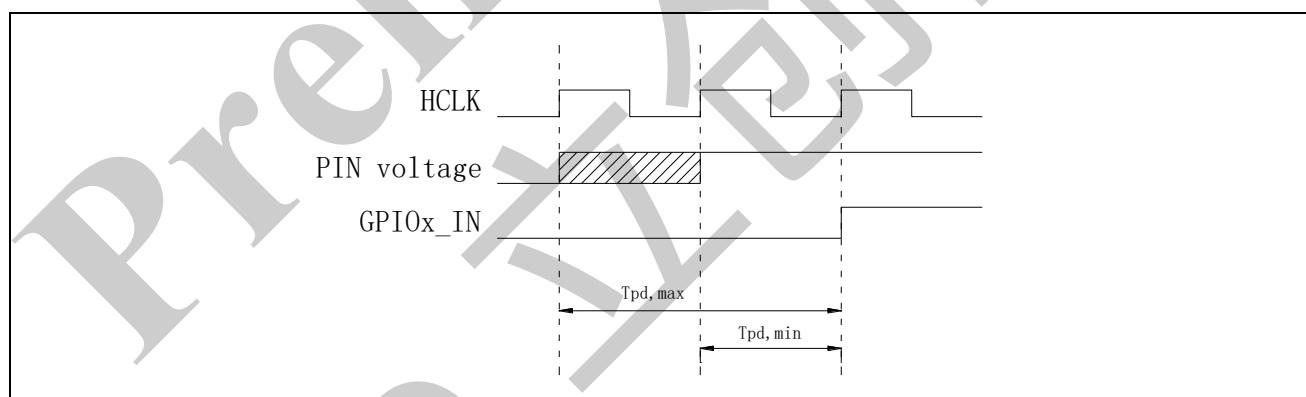


图 8-4 读取端口管脚数据同步图

在系统时钟上升沿之后的时钟时期，管脚电平信号会锁存在内部寄存器，如阴影部分所示，在下一次系统时钟上升沿之后，稳定的管脚电平信号能够被读取。之后在系统时钟上升沿时，数据锁存到 `GPIOx_IN` 寄存器中。信号转换延迟  $T_{pd}$  为 1-2 个系统时钟。

### 8.3.5.2 输出模式

当 GPIO 被配置为输出模式下，其简化框图如下所示：

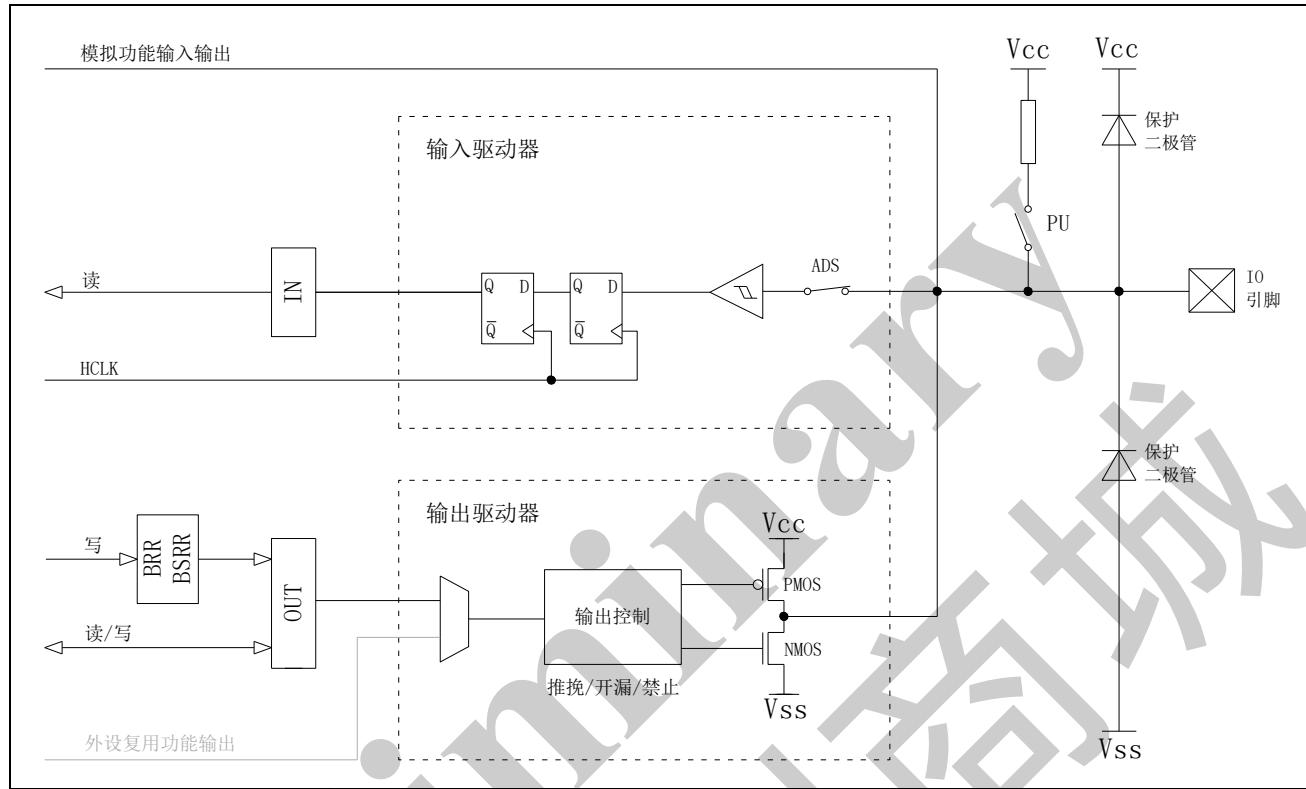


图 8-5 输出配置示意图

每一个端口可以通过向 `GPIOx_BRR`、`GPIOx_BSRR` 或 `GPIOx_OUT` 写入数据以控制 IO 口的输出电平。

`GPIOx_BRR`、`GPIOx_BSRR` 为位操作寄存器，每次可修改同一组端口中 IO 的数量为 1~6 个，且同一组端口中未被修改的 IO 保持其状态不变，使用该两个寄存器对 IO 进行修改的过程为原子操作。对寄存器 `GPIOx_OUT` 的写入值即为每个位所对应的输出 PIN 脚的输出值。

### 8.3.6 端口复用功能

当需要使用数字外设 (GPIO、LPUART、SPI……) 的相应功能时，需要配置端口为数字功能。操作方法为设置 `GPIOx_ADS` 相应的位为 0 并配置 `GPIOx_AFRL`，以实现将特定的片内外设连接到端口。

在该配置下，端口具有如下特性：

- **输入模式：** 可使能上拉；外部信号输入到片内外设；`GPIOx_IN` 可读取外部信号  
可以产生端口中断
- **输出模式：** 可选择开漏或推挽；可使能上拉  
片内外设输出信号到端口；不可以产生端口中断

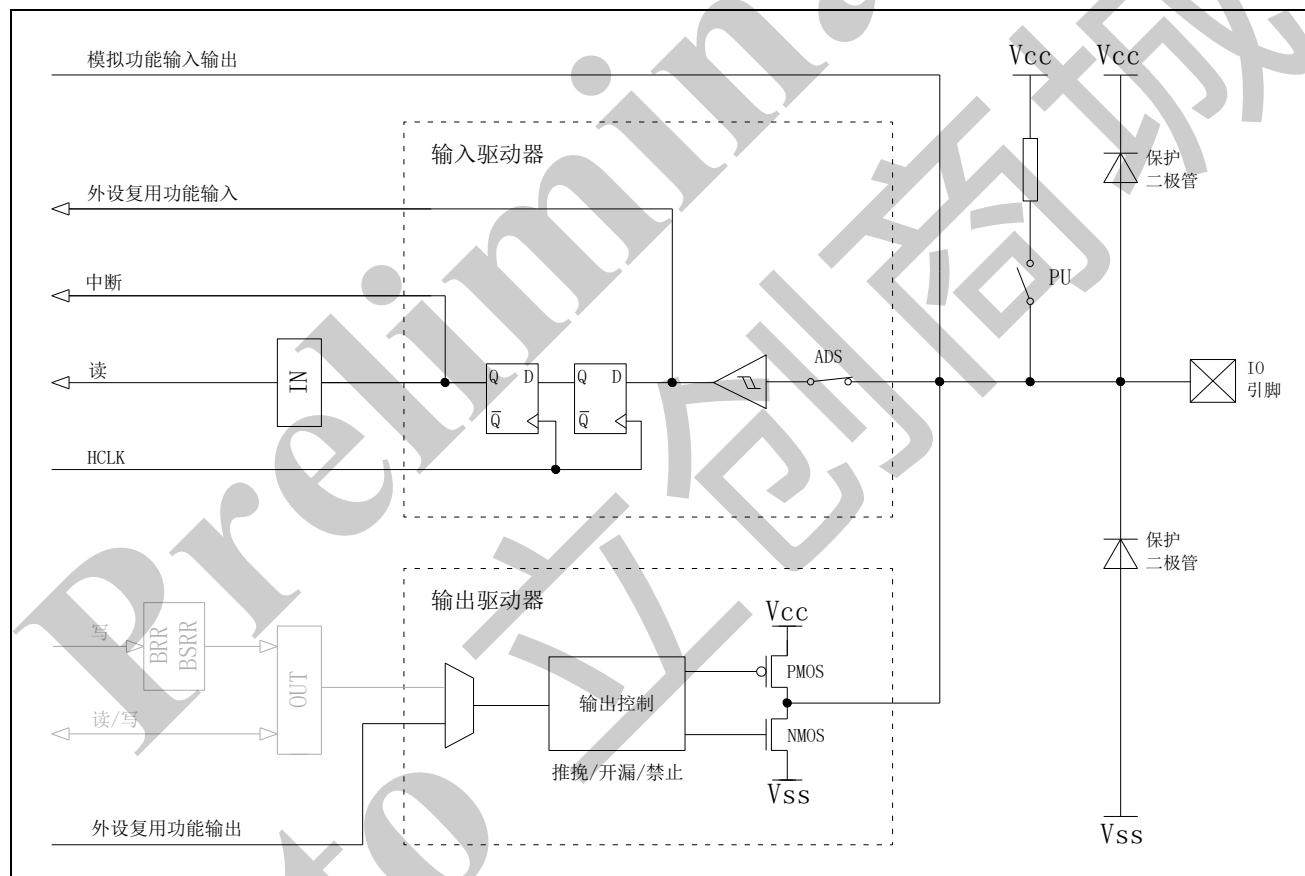


图 8-6 复用功能配置示意图

本芯片每一个端口的数字复用功能如下表所示。

管脚名称	数字功能				
	AF0	AF1	AF2	AF3	AF4
PA01	GPIO	LPUART0_RXD	I2C_SDA	GTIM_TOGP	HCLK_OUT
PA02	GPIO	LPUART0_TXD	I2C_SCL	GTIM_TOGN	LVD_OUT
PA03	GPIO	GTIM_CH2	SPI_NSS	CTRIM_ETR CTRIM_TOG	
PA04	GPIO	GTIM_TOGP	LVD_OUT	SPI_NSS	
PB00	GPIO	ATIM3_CH0B	GTIM_CH3	LPUART0_RTS	
PB01	GPIO	ATIM3_CH1B	GTIM_CH2	LPUART0_CTS	
PB02	GPIO	ATIM3_CH2B	GTIM_CH1	LPUART1_RTS	
PB03	GPIO	ATIM3_ETR	GTIM_CH0	LPUART1_CTS	
PB04	GPIO	I2C_SCL	ATIM3_CH2B	LPUART0_CTS	
PB05	GPIO	I2C_SDA	ATIM3_BK	CTRIM_ETR CTRIM_TOG	
PC03	GPIO	ATIM3_CH2A	ATIM3_CH0B	LPUART0_RTS	
PC04	GPIO	ATIM3_CH0B	ATIM3_CH1B	TCLK_OUT	
PC05	GPIO	SPI_SCK	GTIM_CH3	ATIM3_ETR	
PC06	GPIO	SPI_MOSI	ATIM3_CH0A	LPUART1_RTS	
PC07	GPIO	SPI_MISO	ATIM3_CH1A	LPUART0_RXD	
PD01	GPIO	GTIM_ETR	ATIM3_BK	LPUART0_TXD	
PD02	GPIO	GTIM_CH2	ATIM3_CH2A	LPUART1_CTS	
PD03	GPIO	GTIM_CH1	ATIM3_CH1A	LPUART1_RXD	
PD04	GPIO	GTIM_CH0	LPUART1_TXD	SPI_SCK	
PD05	GPIO	LPUART1_TXD	ATIM3_CH0A	SPI_MISO	
PD06	GPIO	LPUART1_RXD	GTIM_ETR	SPI_MOSI	

### 8.3.7 端口时钟输出

#### 8.3.7.1 输出 HCLK

配置 GPIO 的寄存器，可以输出 HCLK，下方示例从 PA01 输出 HCLK：

- Step1. 设置 GPIOA\_ADS.PIN1 为 0，配置 PA01 为数字端口。
- Step2. 设置 GPIOA\_DIR.PIN1 为 0，配置 PA01 为输出模式。
- Step3. 设置 GPIOA\_AFRL.AFSEL1 为 4，配置 PA01 的复用功能为 HCLK。

#### 8.3.7.2 输出 TCLK

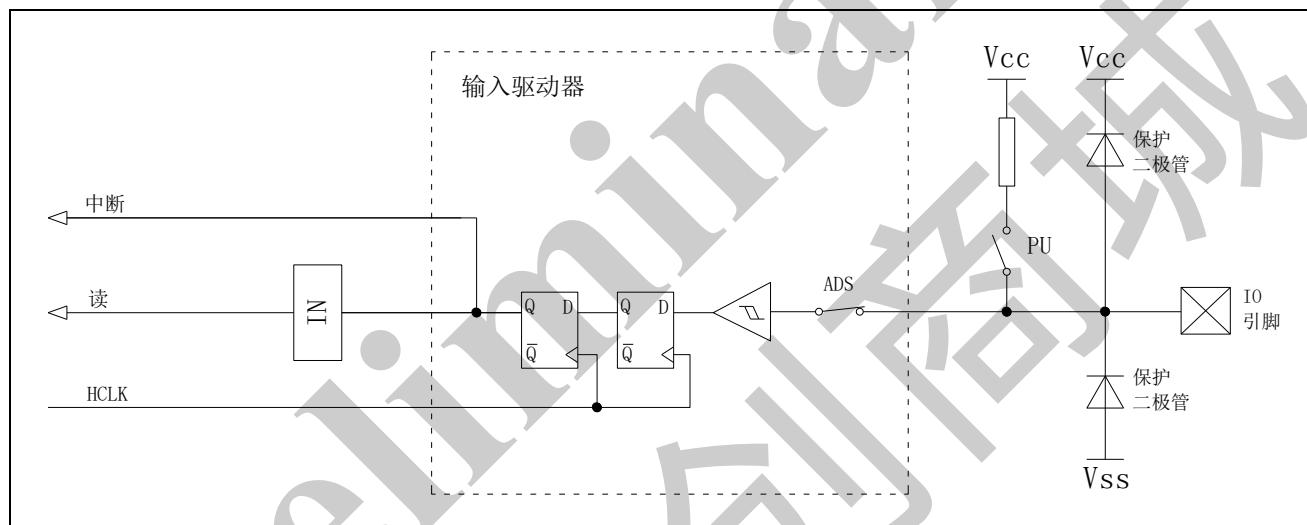
配置 GPIO 的寄存器，可以输出 RCH/RCL/PCLK/HCLK，下方示例从 PC04 输出 RCH/8：

- Step1. 设置 GPIOC\_ADS.PIN4 为 0，配置 PC04 为数字端口。
- Step2. 设置 GPIOC\_DIR.PIN4 为 0，配置 PC04 为输出模式。
- Step3. 设置 GPIOC\_AFRL.AFSEL4 为 3，配置 PC04 的复用功能为 TCLK。
- Step4. 设置 GPIOC\_CR1.TCLKSRC 为 0，选择 TCLK 时钟源为 RCH。
- Step5. 设置 GPIOC\_CR1.TCLKDIV 为 1，输出 TCLK 的 8 分频。

### 8.3.8 端口外部中断

当端口被配置为数字功能（GPIOx\_ADS 为 0）且工作于输入模式（GPIOx\_DIR 为 1）时，每一个端口均具有中断功能。每个端口支持 4 种类型的中断：上升沿中断、下降沿中断、高电平中断、低电平中断；对应的中断使能寄存器分别为：GPIOx\_RISEIE、GPIOx\_FALLIE、GPIOx\_HIGHIE、GPIOx\_LOWIE。当中断触发时，通过查询中断状态寄存器（GPIOx\_IFR）即可判断是哪一个端口触发了中断，通过中断清除寄存器（GPIOx\_ICR）就可以清除相应的中断状态标志位。

GPIO 外部中断相关的简化框图如下所示：



## 8.4 寄存器描述

### 8.4.1 寄存器列表

基地址:

GPIOA: 0x4002 0C00

GPIOB: 0x4002 0D00

GPIOC: 0x4002 0E00

GPIOD: 0x4002 0F00

注:

- GPIOA 有 5 比特有效位 (PA00-40), GPIOB 有 6 比特有效位 (PB00-05), GPIOC 有 5 比特有效位 (PC03-07), GPIOD 有 6 比特有效位 (PD01-06)。对不存在的位, 写操作无效, 读出值始终为 0。

其中 2 个辅助功能寄存器 (CR1, CR4) 为共享寄存器, GPIOA/GPIOB/GPIOC/GPIOD 均可使用并且对 4 个端口均有效, 即 0x4002 0CA4、0x4002 0DA4、0x4002 0EA4 和 0x4002 0FA4 是同一个寄存器。

下面表中 x = A、B、C、D

寄存器名	偏移地址	复位值
端口数模配置寄存器 (GPIOx_AMR)	0x00	(PA)0x0000 001E (PB)0x0000 003F (PC)0x0000 0038 (PD)0x0000 007C
端口输入输出配置寄存器 (GPIOx_DIR)	0x04	(PA)0x0000 001F (PB)0x0000 003F (PC)0x0000 00F8 (PD)0x0000 007E
端口输出类型寄存器 (GPIOx_OpenDrain)	0x08	0x0000 0000
端口上拉寄存器 (GPIOx_PU)	0x18	(PA)0x0000 0001 (PB/C/D)0x0000 0000
端口数据输入寄存器 (GPIOx_IN)	0x1C	0x0000 xxxx
端口数据输出寄存器 (GPIOx_OUT)	0x20	0x0000 xxxx
端口复位寄存器 (GPIOx_BRR)	0x24	0x0000 0000
端口置位/复位寄存器 (GPIOx_BSRR)	0x28	0x0000 0000
端口复用功能低位寄存器 (GPIOx_AFRL)	0x30	0x0000 0000
端口高电平中断使能配置寄存器 (GPIOx_HIGHIE)	0x38	0x0000 0000
端口低电平中断使能配置寄存器 (GPIOx_LOWIE)	0x3C	0x0000 0000
端口上升沿中断使能配置寄存器 (GPIOx_RISEIE)	0x40	0x0000 0000
端口下降沿中断使能配置寄存器 (GPIOx_FALLIE)	0x44	0x0000 0000
端口中断状态寄存器 (GPIOx_IFR)	0x48	0x0000 0000
端口中断清除寄存器 (GPIOx_ICR)	0x4C	(PA)0x0000 001F (PB)0x0000 003F (PC)0x0000 00F8 (PD)0x0000 007E
端口辅助功能配置寄存器1 (GPIOx_CR1)	0xA4	0x0000 0000
端口辅助功能配置寄存器4 (GPIOx_CR4)	0xB0	0x0000 0000

表 8-2 GPIO 寄存器一览表

### 8.4.2 端口数模配置寄存器 (GPIOx\_ADS)

地址偏移量: 0x00

复位值: (PA) 0x0000 001E

(PB) 0x0000 003F

(PC) 0x0000 0038

(PD) 0x0000 007C

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

位	符号	功能描述
31:8	Reserved	保留
7:0	PIN	端口数字功能或模拟功能控制 1: 将端口配置为模拟功能 0: 将端口配置为数字功能 注: PA 端口有效位[4:1], PA[0]仅作为数字端口用, PB 端口有效位[5:0], PC 端口有效位[7:3], PD 端口有效位[6:1]

### 8.4.3 端口方向配置寄存器(GPIOx\_DIR)

地址偏移量: 0x04

复位值: (PA)0x0000 001F

(PB)0x0000 003F

(PC)0x0000 00F8

(PD)0x0000 007E

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

位	符号	功能描述
31:8	Reserved	保留
7:0	PIN	端口输入输出方向控制 1: 将端口配置成输入 0: 将端口配置成输出 注: PA 端口有效位[4:1], PA[0]仅输入, PB 端口有效位[5:0], PC 端口有效位[7:3], PD 端口有效位[6:1]

#### 8.4.4 端口输出类型寄存器 (GPIOx\_OpenDrain)

地址偏移量: 0x08

复位值: 0x0000 0000

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

位	符号	功能描述
31:8	Reserved	保留
7:0	PIN	端口输出方式控制位 0: 推挽输出 (复位值) 1: 开漏输出 注: PA 端口有效位[4:1], PB 端口有效位[5:0], PC 端口有效位[7:3], PD 端口有效位[6:1]

#### 8.4.5 端口上拉寄存器 (GPIOx\_PU)

地址偏移量: 0x18

复位值: (PA)0x0000 0001

(PB/C/D)0x0000 0000

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

位	符号	功能描述
31:8	Reserved	保留
7:0	PUx	端口上位电阻使能控制 0: 禁止上拉电阻 1: 使能上拉电阻 注: PA 端口有效位[4:0], PA[0]作为 RESET PAD 时自动上拉, PB 端口有效位[5:0], PC 端口有效位[7:3], PD 端口有效位[6:1]

#### 8.4.6 端口数据输入寄存器 (GPIOx\_IN)

地址偏移量: 0x1C

复位值: 0x0000 XXXX

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

位	符号	功能描述
31:8	Reserved	保留
7:0	PIN	端口输入数据 PA 端口有效位[4:0], PB 端口有效位[5:0], PC 端口有效位[7:3], PD 端口有效位[6:1]

#### 8.4.7 端口数据输出寄存器 (GPIOx\_OUT)

地址偏移量: 0x20

复位值: 0x0000 XXXX

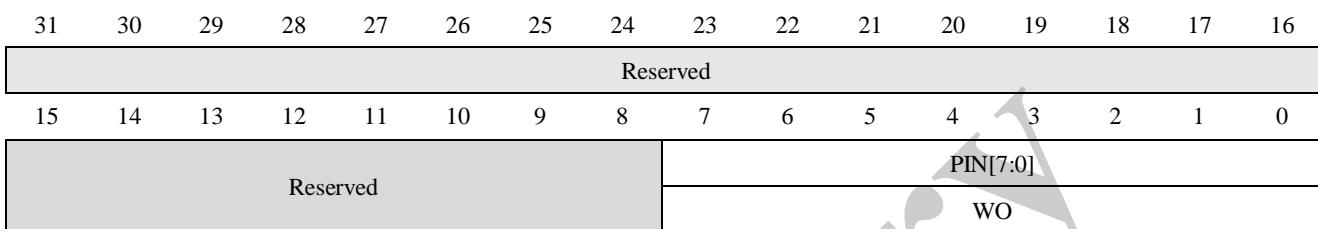
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									PIN[7:0]						
									RW						

位	符号	功能描述
31:8	Reserved	保留
7:0	PIN	端口输出数据 注: PA 端口有效位[4:1], PB 端口有效位[5:0], PC 端口有效位[7:3], PD 端口有效位[6:1]

### 8.4.8 端口复位寄存器 (GPIOx\_BRR)

地址偏移量: 0x24

复位值: 0x0000 0000



位	符号	功能描述
31:8	Reserved	保留
7:0	PIN	端口位清零控制 0: 不影响 GPIOx_OUT 寄存器相应的比特 1: 设置 GPIOx_OUT 寄存器相应的比特为 0 注: PA 端口有效位[4:1], PB 端口有效位[5:0], PC 端口有效位[7:3], PD 端口有效位[6:1]

### 8.4.9 端口置位复位寄存器 (GPIOx\_BSRR)

地址偏移量: 0x28

复位值: 0x00000000

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

位	符号	功能描述
31:24	Reserved	保留
23:16	BR	端口位清零控制 0: 不影响 GPIOx_OUT 寄存器相应的比特 1: 设置 GPIOx_OUT 寄存器相应的比特为 0 注: PA 端口有效位[20:17], PB 端口有效位[21:16], PC 端口有效位[23:19], PD 端口有效位[22:17]
15:8	Reserved	保留
7:0	BS	端口位置位控制 0: 不影响 GPIOx_OUT 寄存器相应的比特 1: 设置 GPIOx_OUT 寄存器相应的比特为 1 注: PA 端口有效位[4:1], PB 端口有效位[5:0], PC 端口有效位[7:3], PD 端口有效位[6:1]

### 8.4.10 端口复用功能低位寄存器 (GPIOx\_AFRL)

地址偏移量: 0x30

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL7				AFSEL6				AFSEL5				AFSEL4			
RW	RW	RW	RW												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL3				AFSEL2				AFSEL1				AFSEL0			
RW	RW	RW	RW												

位	符号	功能描述
31:0	AFSEL <sub>y</sub> y = 0-7	端口复用功能控制 0000: GPIO 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101-1111: 保留位 注 1: AF1 – AF4 的具体功能详见【端口复用功能】章节 注 2: PA 端口有效位[19:4], PB 端口有效位[23:0], PC 端口有效位[31:12], PD 端口有效位[27:4]

### 8.4.11 端口高电平中断使能配置寄存器(GPIOx\_HIGHIE)

地址偏移量: 0x38

复位值: 0x0000 0000

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

位	标记	功能描述
31:8	Reserved	保留
7:0	PIN	端口高电平中断使能控制 1: 使能相应端口的高电平中断 0: 禁止相应端口的高电平中断 注: PA 端口有效位[4:0], PB 端口有效位[5:0], PC 端口有效位[7:3], PD 端口有效位[6:1]

### 8.4.12 端口低电平中断使能配置寄存器(GPIOx\_LOWIE)

地址偏移量: 0x3C

复位值: 0x0000 0000

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

位	标记	功能描述
31:8	Reserved	保留
7:0	PIN	端口低电平中断使能控制 1: 使能相应端口的低电平中断 0: 禁止相应端口的低电平中断 注: PA 端口有效位[4:0], PB 端口有效位[5:0], PC 端口有效位[7:3], PD 端口有效位[6:1]

### 8.4.13 端口上升沿中断使能配置寄存器(GPIOx\_RISEIE)

地址偏移量: 0x40

复位值: 0x0000 0000

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

位	标记	功能描述
31:8	Reserved	保留
7:0	PIN	端口上升沿中断使能控制 1: 使能相应端口的上升沿中断 0: 禁止相应端口的上升沿中断 注: PA端口有效位[4:0], PB端口有效位[5:0], PC端口有效位[7:3], PD端口有效位[6:1]

### 8.4.14 端口下降沿中断使能配置寄存器(GPIOx\_FALLIE)

地址偏移量: 0x44

复位值: 0x0000 0000

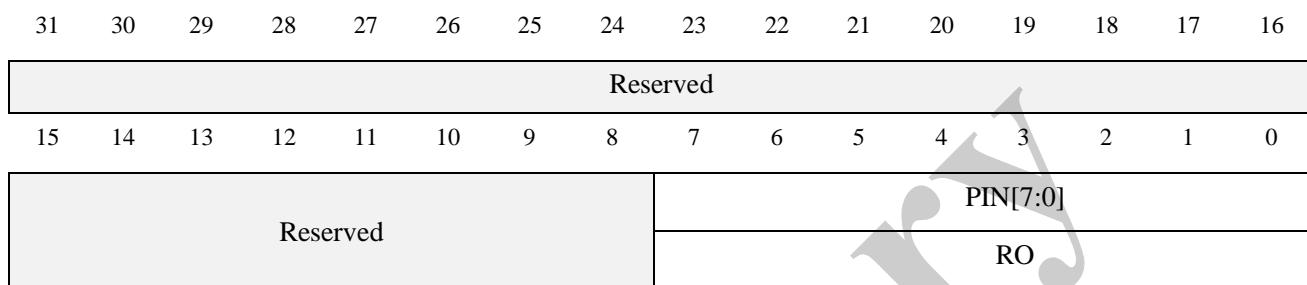
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								PIN[7:0]							
								RW							

位	标记	功能描述
31:8	Reserved	保留
7:0	PIN	端口下降沿中断使能控制 1: 使能相应端口的下降沿中断 0: 禁止相应端口的下降沿中断 注: PA 端口有效位[4:0], PB 端口有效位[5:0], PC 端口有效位[7:3], PD 端口有效位[6:1]

### 8.4.15 端口中断状态寄存器(GPIOx\_IFR)

地址偏移量: 0x48

复位值: 0x0000 0000



位	标记	功能描述
31:8	Reserved	保留
7:0	PIN	端口中断状态标志 1: 已发生符合条件的中断 0: 未发生符合条件的中断 注: PA端口有效位[4:0], PB端口有效位[5:0], PC端口有效位[7:3], PD端口有效位[6:1]

### 8.4.16 端口中断清除寄存器(GPIOx\_ICR)

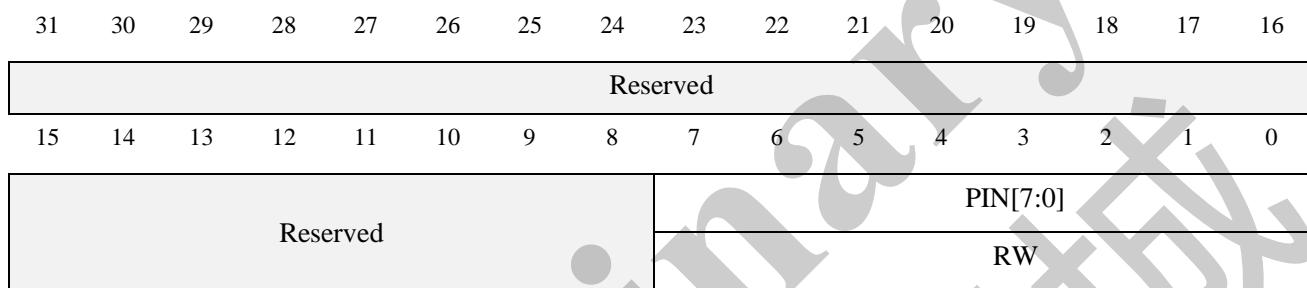
地址偏移量: 0x4C

复位值: (PA)0x0000 001F

(PB)0x0000 003F

(PC)0x0000 00F8

(PD)0x0000 007E



位	标记	功能描述
31:8	Reserved	保留
7:0	PIN	端口中断标志清除 1: 无功能 0: 清除相应的中断标志位 注: PA端口有效位[4:0], PB端口有效位[5:0], PC端口有效位[7:3], PD端口有效位[6:1]

### 8.4.17 端口辅助功能配置寄存器 1(GPIOx\_CR1)

地址偏移量: 0xA4

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										TCLKDIV	Res.	TCLKSRC			
										RW				RW	

位	标记	功能描述
31:5	Reserved	保留
4:3	TCLKDIV	TCLK_OUT输出分频控制 00: TCLK 01: TCLK/8 10: TCLK/64 11: TCLK/256
2	Reserved	保留
1:0	TCLKSRC	TCLK_OUT输出信号选择 00: RCH 01: RCL 10: PCLK 11: HCLK

### 8.4.18 端口辅助功能配置寄存器 4 (GPIO\_CR4)

地址偏移量: 0xB0

复位值: 0x0000 0000

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

Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				GTIMCH0		Reserved		ATIM3CH0B		Reserved					
				RW		Reserved		RW							

位	标记	功能描述
31:10	Reserved	保留
9:8	GTIMCH1	GTimer 定时器CH0输入来源配置 详见 GPIO_CR4[5:4]
7:6	Reserved	保留
5:4	ATIM3CH0B	ATimer3 定时器CH0B输入来源配置 00: 由GPIOx_AFRL决定 01: LPUART0_RXD 10: LPUART1_RXD 11: LVD_OUT
3:0	Reserved	保留

## 9 FLASH 控制器 (FLASH)

### 9.1 概述

本系统包含一块 18k 字节 (Byte) 容量的 FLASH 存储器, 共划分为 36 个页 (Sector), 每个页 (Sector) 的容量为 512 字节 (Byte)。

本控制器支持对 FLASH 存储器的擦除、编程以及读操作。本控制器支持的读写位宽为 Byte (8 位)、half-word (16 位) 和 word (32 位)。注意, Byte 操作的目标地址必须按字节对齐, half-word 操作的目标地址必须按半字对齐 (地址最低比特为 0), word 操作的地址必须按字对齐 (地址最低两比特均为 0)。如果目标地址没有按照位宽对齐, 则该操作无效, 并且 CPU 会进入 Hard Fault 出错中断。

### 9.2 容量划分

本控制器支持的 FLASH 容量为 18KB, 每个页 (Sector) 的容量为 512 字节 (Byte), 其组成方式如下方所示。

地址	序号	.....	地址	序号
0x0E00 – 0x0FFF	Sector7	.....	0x4600 – 0x47FF	Sector35
0x0C00 – 0x0DFF	Sector6	.....	0x4200 – 0x43FF	Sector34
0x0A00 – 0x0BFF	Sector5	.....	0x4400 – 0x45FF	Sector33
0x0800 – 0x09FF	Sector4	.....	0x4000 – 0x41FF	Sector32
0x0600 – 0x07FF	Sector3	.....	0x3E00 – 0x3FFF	Sector31
0x0400 – 0x05FF	Sector2	.....	0x3C00 – 0x3DFF	Sector30
0x0200 – 0x03FF	Sector1	.....	0x3A00 – 0x3BFF	Sector29
0x0000 – 0x01FF	Sector0	.....	0x3800 – 0x39FF	Sector28

表 9-1 FLASH 容量划分

### 9.3 读等待周期

本设备内置的 FLASH 支持的最快取指频率为 24MHz, 故当 CPU 频率 (HCLK) 大于 24MHz 时, CPU 从 FLASH 取指必需要插入等待周期 (设置 FLASH\_WAIT 大于 0)。当 FLASH\_WAIT 等于 1 时, 每两个 HCLK 可以完成一次 FLASH 取指操作。

## 9.4 FLASH 操作（读、写、擦）

### 9.4.1 页擦除（Sector Erase）

页擦除每次可以擦除用户指定的一个页（Sector）。擦除操作完成后，页（Sector）内的数据均为 0xFF。如果该擦除操作是从 FLASH 内执行，则 CPU 会停止取指，硬件自动等待该操作完成（FLASH\_CR.BUSY 变为 0）；如果该擦除操作是从 RAM 内执行，则 CPU 不会停止取指，用户软件应等待该操作完成（FLASH\_CR.BUSY 变为 0）。

Sector 擦除操作步骤如下：

- Step1：**向 FLASH\_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step2：**配置 FLASH\_CR.OP 为 2，设置 Flash 操作模式为 Sector 擦除。
- Step3：**检查 FLASH\_CR.OP 是否为 2，如不为 2 则跳转到 Step1。
- Step4：**向 FLASH\_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step5：**设置 FLASH\_SLOCK 相应的比特为 1，去除该 Sector 的擦写保护。
- Step6：**检查 FLASH\_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step4。
- Step7：**对待擦除的 Sector 内的任意地址进行写入任意数据，触发 Sector 擦除。  
例：`* ((unsigned char *) 0x00000200) = 0x00。`
- Step8：**等待 FLASH\_CR.BUSY 变为 0，Sector 擦除操作完成。
- Step9：**如需擦除其它 Sector，重复 Step4 – Step8。

### 9.4.2 全片擦除（Chip Erase）

全片擦除可以一次性擦除全部的页（Sector）。擦除操作完成后，所有页（Sector）内的数据均为 0xFF。如果该擦除操作是从 FLASH 内执行，则该操作会被禁止，并置起相应的出错标志位；如果该擦除操作是从 RAM 内执行，则 CPU 不会停止取指，用户软件应等待该操作完成（FLASH\_CR.BUSY 变为 0）。

Chip 擦除操作步骤如下：

- Step1：**向 FLASH\_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step2：**配置 FLASH\_CR.OP 为 3，设置 Flash 操作模式为 Chip 擦除。

**Step3:** 检查 FLASH\_CR.OP 是否为 3, 如不为 3 则跳转到 Step1。

**Step4:** 向 FLASH\_BYPASS 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

**Step5:** 设置 FLASH\_SLOCK 为 0xFFFFFFFF, 去除所有 Sector 的擦写保护。

**Step6:** 检查 FLASH\_SLOCK 是否为 0xFFFFFFFF, 如不为 0xFFFFFFFF 则跳转到 Step4。

**Step7:** 对待擦除的 Chip 内的任意地址进行写操作, 触发 Chip 擦除。

例: `*((unsigned char *) 0x00000000) = 0x00.`

**Step11:** 等待 FLASH\_CR.BUSY 变为 0, Chip 擦除操作完成。

#### 9.4.3 写操作 (Program)

写操作只能将 FLASH 内的比特数据由 1 写成 0, 故写入数据前应确保待写入的地址内的数据为 0xFF。支持写入 3 种数据长度: Byte(8bits)、Half-word(16bits)、Word(32bits), 写入的数据以小端模式存放在 FLASH 中, 即低地址存放数据的低字节。如果写操作是从 FLASH 内执行, 则 CPU 会停止取指, 硬件自动等待该操作完成 (FLASH\_CR.BUSY 变为 0); 如果该写操作是从 RAM 内执行, 则 CPU 不会停止取指, 用户软件应等待该操作完成 (FLASH\_CR.BUSY 变为 0)。

Byte 写操作步骤如下:

**Step1:** 向 FLASH\_BYPASS 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

**Step2:** 配置 FLASH\_CR.OP 为 1, 设置 Flash 操作模式为写入。

**Step3:** 检查 FLASH\_CR.OP 是否为 1, 如不为 1 则跳转到 Step1。

**Step4:** 向 FLASH\_BYPASS 寄存器依次写入 0x5A5A、0xA5A5, 使能寄存器改写。

**Step5:** 设置 FLASH\_SLOCK 相应的比特为 1, 去除擦写保护。

**Step6:** 检查 FLASH\_SLOCK 相应的比特是否为 1, 如不为 1 则跳转到 Step4。

**Step7:** 对待写入的目标地址进行 Byte 写操作, 触发写入操作。

例: `*((unsigned char *) 0x00001231) = 0x5A.`

**Step8:** 等待 FLASH\_CR.BUSY 变为 0, 写入操作完成。

**Step9:** 如需写 Byte 到其它地址, 重复 Step7 – Step8。

Half-word 写操作步骤如下：

- Step1:** 向 FLASH\_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step2:** 配置 FLASH\_CR.OP 为 1，设置 Flash 操作模式为写入。
- Step3:** 检查 FLASH\_CR.OP 是否为 1，如不为 1 则跳转到 Step1。
- Step4:** 向 FLASH\_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step5:** 设置 FLASH\_SLOCK 相应的比特为 1，去除擦写保护。
- Step6:** 检查 FLASH\_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step4。
- Step7:** 对待写入的目标地址进行 Half-word 写操作，触发写入操作。

例：\* ((unsigned short int \*) 0x00001232) = 0xABCD。

- Step8:** 等待 FLASH\_CR.BUSY 变为 0，写入操作完成。

- Step9:** 如需写 Half-word 到其它地址，重复 Step7 – Step8。

Word 写操作步骤如下：

- Step1:** 向 FLASH\_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step2:** 配置 FLASH\_CR.OP 为 1，设置 Flash 操作模式为写入。
- Step3:** 检查 FLASH\_CR.OP 是否为 1，如不为 1 则跳转到 Step1。
- Step4:** 向 FLASH\_BYPASS 寄存器依次写入 0x5A5A、0xA5A5，使能寄存器改写。
- Step5:** 设置 FLASH\_SLOCK 相应的比特为 1，去除擦写保护。
- Step6:** 检查 FLASH\_SLOCK 相应的比特是否为 1，如不为 1 则跳转到 Step4。
- Step7:** 对待写入的目标地址进行 Word 写操作，触发写入操作。

例：\* ((unsigned int \*) 0x00001234) = 0x55667788。

- Step8:** 等待 FLASH\_CR.BUSY 变为 0，写入操作完成。

- Step9:** 如需写 Word 到其它地址，重复 Step7 – Step8。

#### 9.4.4 读操作 (Read)

支持读出 3 种数据长度: Byte (8bits)、Half-word (16bits)、Word (32bits), 读出的数据为小端模式, 即低地址存放数据的低字节。读操作无需操作步骤, 任何时刻都可以读出 FLASH 内的数据。

例:

Byte 读操作: `temp = * ((unsigned char *) 0x00001231)`

Half-word 读操作 `temp = * ((unsigned short int *) 0x00001232)`

Word 读操作 `temp = * ((unsigned int *) 0x00001234)`

Preliminary  
to be confirmed

## 9.5 FLASH 安全保护

### 9.5.1 页面擦写保护

FLASH 存储器共 18K 字节，被划分为 36 个页，每 4 个页共用一个擦写保护位。当页被保护时，对该页进行的擦写操作均无效并产生报警标志位和中断信号。当 FLASH 存储器中的任意页被保护时，对该 FLASH 的全片擦写无效，并产生报警标志位和中断信号。

### 9.5.2 PC 地址擦写保护

若 CPU 在 FLASH 中执行程序且当前 PC 指针正好落在待擦写的页地址范围之内，那么该擦写操作无效并产生报警标志位和中断信号。基于该特性，只有从 RAM 中执行代码时，才可以对 FLASH 进行片擦操作。

### 9.5.3 寄存器写保护

本模块的重要控制器屏蔽普通的写操作，必须用写序列方式才能修改。

需要通过写序列方式才能更改的寄存器如下所示：

FLASH\_CR、FLASH\_SLOCK。

不需要通过写序列方式即可更改的寄存器如下所示：

FLASH\_ICLR、FLASH\_BYPASS。

通过写序列方式修改寄存器值的具体操作步骤如下所示：

**Step1：**向 FLASH\_BYPASS 寄存器写入 0x5A5A。

**Step2：**向 FLASH\_BYPASS 寄存器写入 0xA5A5。

**Step3：**对待修改的寄存器写入目标值。

**Step4：**验证待修改的寄存器当前值是否与目标相同，如不则跳转到 Step1。

**Step5：**执行其他操作。

注意：

- 写 0x5A5A 和写 0xA5A5 这两步操作之间不可插入任何写操作，否则无法改写目标寄存器的数值。如改写失败，需要重新进行这三步操作。

### 9.5.4 数据读出保护

本芯片内置 FLASH 支持 4 级读保护，当前保护等级可通过 FLASH\_LockState 寄存器读出。使用编程器对芯片进行编程时，可配置 FLASH 的读保护等级。

芯片在各保护等级下的功能如下表所示。

保护等级	芯片功能
Level0	可以通过 ISP 及 SWD 对 FLASH 进行读写
Level1	数据不可读出 可以通过 ISP 及 SWD 降级到 Level0，降级后 FLASH 内容全为 FF
Level2	数据不可读出 可以通过 ISP 降级到 Level0，降级后 FLASH 内容全为 FF SWD 接口被禁止，数据不可读出
Level3	数据不可读出 ISP 及 SWD 接口均被禁止，即芯片只能进行一次编程

## 9.6 寄存器描述

### 9.6.1 控制寄存器列表

基地址: 0x4002 0000

寄存器	偏移地址	描述
FLASH_CR	0x20	控制寄存器
FLASH_IFR	0x24	中断标志寄存器
FLASH_ICLR	0x28	中断标志清除寄存器
FLASH_BYPASS	0x2C	序列寄存器
FLASH_SLOCK	0x30	擦写保护寄存器
FLASH_WAIT	0x50	读等待周期寄存器
FLASH_LockState	0x54	读保护状态寄存器

## 9.6.2 控制寄存器 (FLASH\_CR)

偏移地址: 0x20

复位值: 0x0000 0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						Lp Mode	Reserved	FLASH IE	PCIE	BUSY	Reserved	OP			
						RW		RW	RW	RO		RW			

位	标记	功能描述
31:10	Reserved	保留位
9	LpMode	FLASH 低功耗控制 0: 当系统进入 DeepSleep 模式, FLASH 不进入低功耗模式; 1: 当系统进入 DeepSleep 模式, FLASH 进入低功耗模式; 注: 保持该寄存器值为 1
8:7	Reserved	保留位
6	FLASHIE	FLASH 被保护地址擦写中断使能控制 0: 禁止 1: 使能
5	PCIE	擦写 PC 所在FLASH页面中断使能控制 0: 禁止 1: 使能
4	BUSY	FLASH擦写状态标志 0: 擦写操作已完成 1: 擦写操作未完成
3:2	Reserved	保留位
1:0	OP	FLASH 操作模式配置 00: 读 (Read) 01: 写 (Program) 10: 页擦除 (Sector Erase) 11: 全片擦除 (Chip Erase)

### 9.6.3 中断标志寄存器 (FLASH\_IFR)

偏移地址: 0x24

复位值: 0x0000 0000

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

位	标记	描述
31:2	Reserved	保留位
1	FLASH	FLASH擦写保护报警中断标志位
0	PC	FLASH擦写 PC 地址报警中断标志位

### 9.6.4 中断标志清除寄存器 (FLASH\_ICLR)

偏移地址: 0x28

复位值: 0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved																
															FLASH	PC
															R1W0	R1W0

位	标记	功能描述
31:2	Reserved	保留位
1	FLASH	清除保护报警中断标志位 写0清除 写1无效
0	PC	清除 PC 地址报警中断标志位 写0清除 写1无效

### 9.6.5 序列寄存器 (FLASH\_BYPASS)

偏移地址: 0x2C

复位值: 0x0000 0000

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

位	标记	描述
31:16	Reserved	保留位
15:0	BYPASS	在修改本模块特定寄存器之前，必须对本寄存器写入 0x5A5A - 0xA5A5 序列。每次写入该 Bypass 序列后，只可以修改一次寄存器。如需再次修改寄存器，必须再次输入 0x5A5A - 0xA5A5 序列。

### 9.6.6 擦写保护寄存器 (FLASH\_SLOCK)

偏移地址: 0x30

复位值: 0x0000 0000

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

位	标记	描述
31:9	Reserved	保留位
8:0	SLOCK	FLASH页面擦写保护配置 0: 不允许擦写 1: 允许擦写 Bit0 对应: Sector0 - Sector3 Bit1 对应: Sector4 - Sector7 Bit2 对应: Sector8 - Sector11 ..... Bit8 对应: Sector32 - Sector35

### 9.6.7 读等待周期寄存器 (FLASH\_WAIT)

偏移地址: 0x50

复位值: 0x0000 0000

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

位	标记	功能描述
31:2	Reserved	保留位
1:0	WAIT	FLASH取指等待周期配置 00: 1个周期, 适用于HCLK <= 24MHz 01: 2个周期, 适用于24MHz < HCLK <= 48MHz 10: 保留 11: 保留

### 9.6.8 读保护状态寄存器 (FLASH\_LockState)

偏移地址: 0x54

复位值: 0x0000 0000

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

位	标记	描述
31:2	Reserved	保留位
1:0	STATE	FLASH 保护状态 00: Level0, ISP可读写, SWD可读写 01: Level1, ISP可降级, SWD可降级; 数据不可读出 10: Level2, ISP可降级, SWD无功能; 数据不可读出 11: Level3, ISP无功能, SWD无功能; 数据不可读出

## 10 RAM 控制器 (RAM)

### 10.1 概述

本系统中包含一块容量为 2k 字节 (Byte) 的 SRAM，支持字节 (8 位)、半字 (16 位 bits)、字 (32 位) 三种读写操作。可在系统时钟频率下进行读写操作，无须等待周期。

### 10.2 功能描述

#### 10.2.1 RAM 地址范围

RAM 在系统映射中的地址范围如下表所示：

地址范围	大小	Memory类型
0x20000000 – 0x200007FF	2K Byte	SRAM

表 10-1 RAM 地址映射

#### 10.2.2 读写位宽

本控制器支持 字节 (8 位)、半字 (16 位)、字 (32 位) 三种位宽的读写操作。字节操作的地址必须按字节对齐，半字操作的目标地址必须按半字对齐 (地址最低比特为 0)，字操作的地址必须按字对齐 (地址最低两比特均为 0)。如果读写操作的目标地址没有按照位宽规定对齐，该操作无效，并且系统会产生 Hard Fault 出错中断。

## 11 基本定时器 (BTIM)

### 11.1 概述

BTIM 是一组具有基本定时功能的定时计数器，每组包含 3 个 16 位的具有重载功能的定时器。本产品含有基本定时器：BTIM3/4/5。BTIM3/4/5 与 GTIM 功能不可同时使用，由 SysCtrl\_CR1.GTIMCFG 控制，该位为 1 时 BTIM3/4/5 有效。

### 11.2 主要特性

- 3 个 16bit 定时器
- 每个定时器 2 个翻转输出
- 单脉冲模式
- 外部输入计数功能
- 门控功能
- 外部触发功能
- 级联功能

Preliminary  
to be confirmed

## 11.3 功能描述

### 11.3.1 功能框图

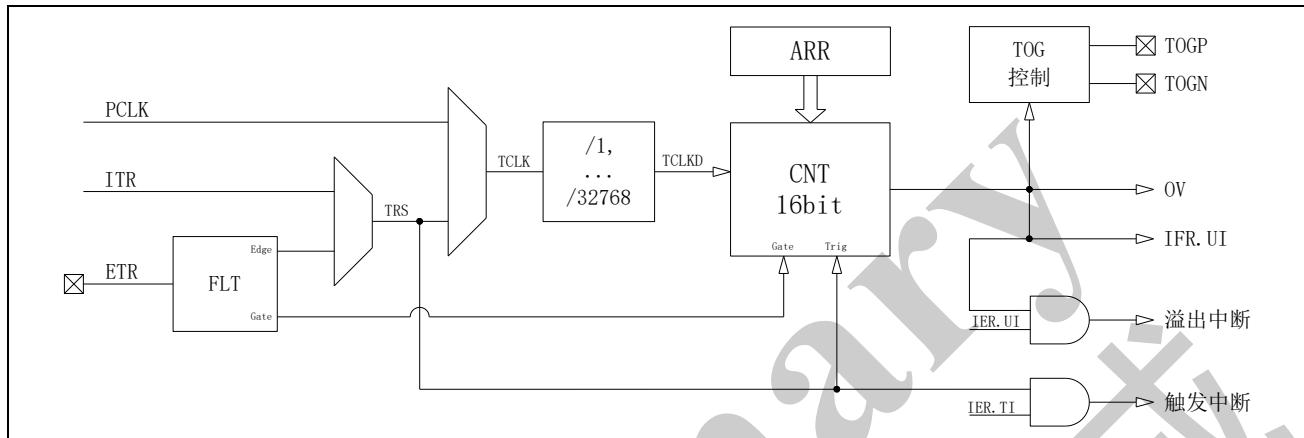


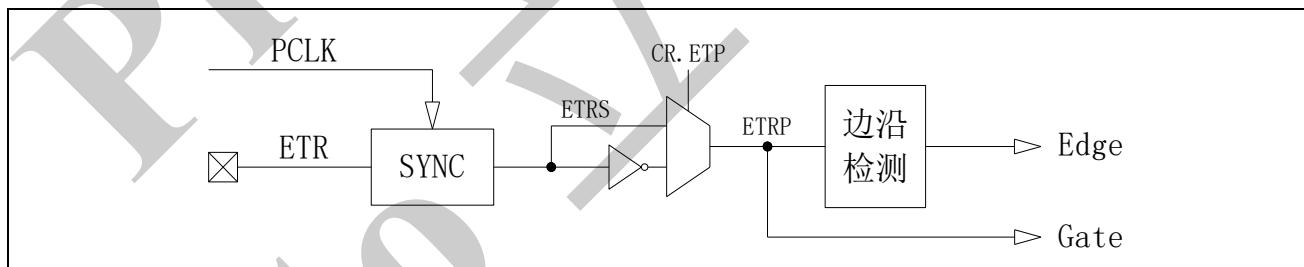
图 11-1 BTIM 框图

注：

- BIM3/4/5 共用一个 ETR
- ITR 信号来源详见级联功能章节

### 11.3.2 滤波单元

外部管脚输入的 ETR 信号经滤波、极性选择、边沿检测后，输出门控信号及边沿信号供计数单元及触发单元使用。由于 ETR 信号采用 PCLK 及其分频进行同步滤波，故输入的 ETR 信号的频率应低于 PCLK 的频率的二分之一。



### 11.3.3 计数单元

在每个 TCLKD 信号的上升沿，内部计数器 CNT 自动加 1。计数到 ARR 后产生溢出信号 OV（宽度为一个 PCLK），并复位内部计数值为 0 并再次开始加计数。计数器的计数范围为 0 至 ARR，计数单元的计数周期为 ARR+1。

定时器支持 2 种计数模式，One-Shot 模式与连续计数模式。当设置 BTIMx\_CR.OST 为

1 时，定时器工作于 One-Shot 模式；发生溢出后定时器立即停止计数，硬件自动清除 CEN。当设置 BTIMx\_CR.OST 为 0 时，定时器工作于连续计数模式；只要 BTIMx\_CR.CEN 为 1，定时器会一直进行计数。

每次溢出时，BTIMx\_IFR.UI 会被硬件置 1，若使能相应中断则可以产生溢出中断，用户应在中断服务程序中清除该溢出标志。

若 BTIMx\_CR.TOGN 为 1，则每溢出一次 TOGP/TOGN 输出电平翻转一次；用户可通过 GPIO 的辅助功能将 TOGP/TOGN 信号输出到管脚用于驱动外部电路。

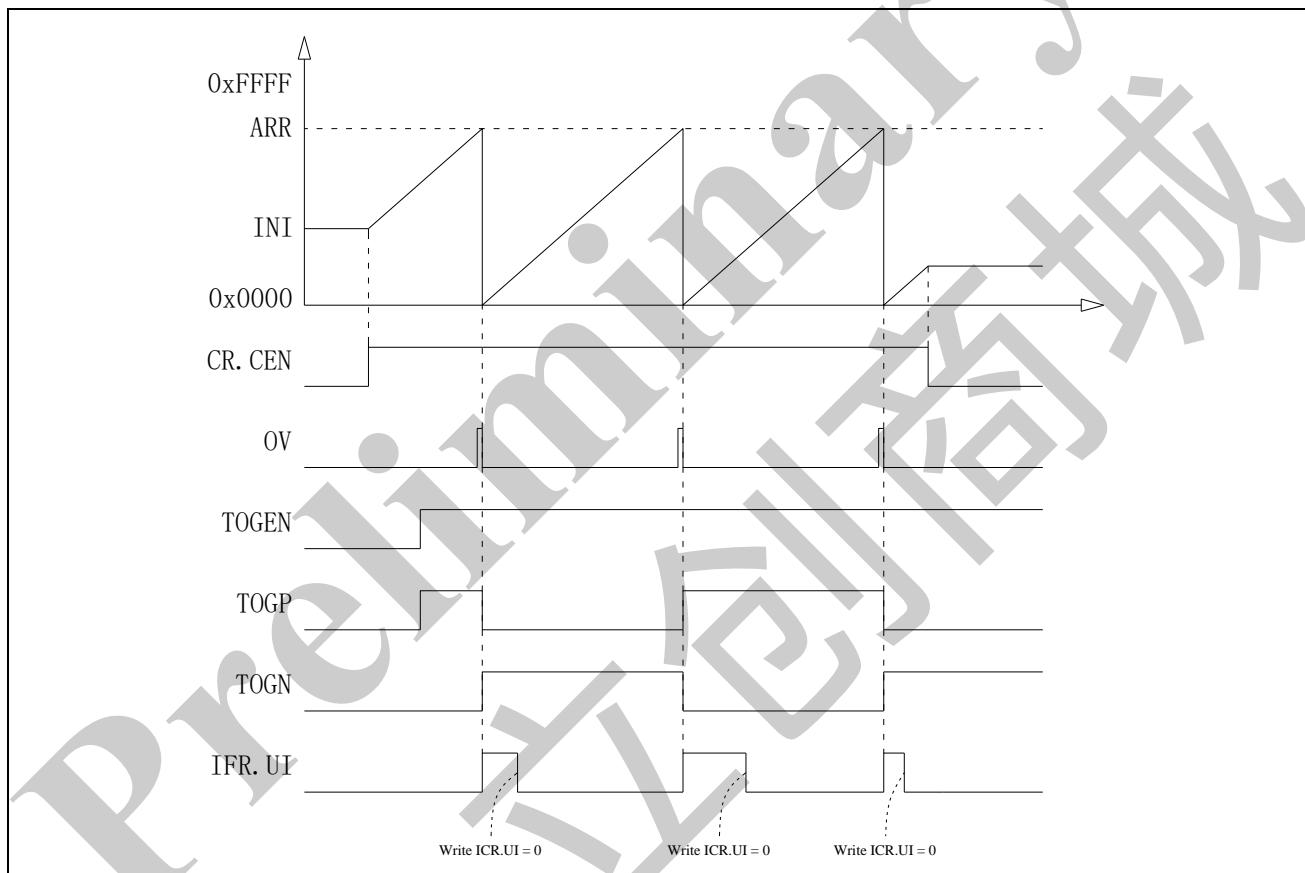


图 11-2 计数器波形

#### 11.3.4 定时器模式

当设置 BTIMx\_CR.MD 为 0x00 时，BTIMx 工作于定时器模式。在该模式下，计数时钟为 PCLK 分频后的时钟 TCLKD，当设置 BTIMx\_CR.CEN 为 1 后，定时器即从初值开始加计数，计数到 ARR 后产生溢出并重新从 0 开始计数。

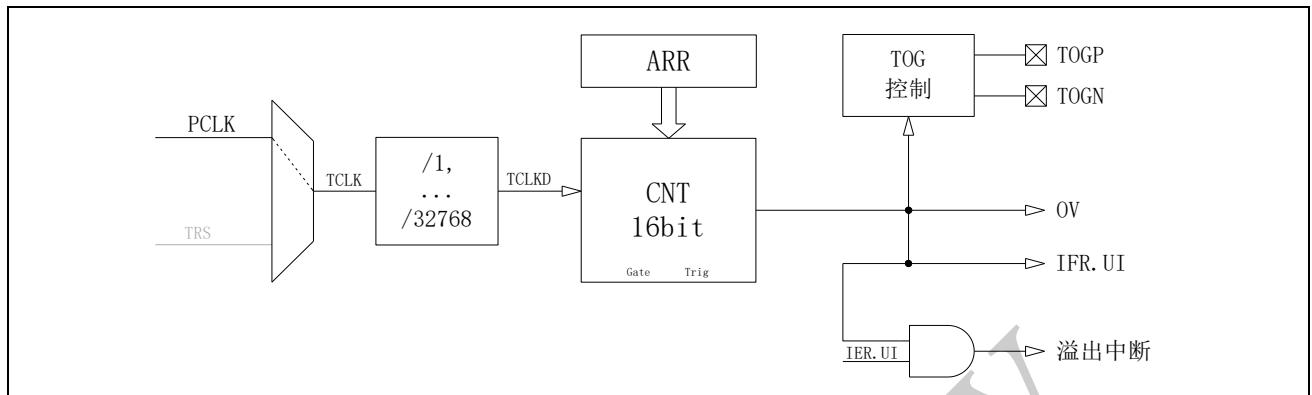


图 11-3 定时模式框图

### 11.3.5 计数器模式

当设置 BTIMx\_CR.MD 为 0x01 时，BTIMx 工作于计数器模式。在该模式下，计数时钟为分频后的 ITR 或 ETR 管脚输入的信号。当设置 BTIMx\_CR.CEN 为 1 后，计数器即从初值开始加计数，计数到 ARR 后产生溢出并重新从 0 开始计数。

通过 BTIMx\_CRTRS 可以选择计数时钟源为 ITR 或 ETR；通过 BTIMx\_CRET 可以选择对 ETR 管脚输入信号的上升沿或下降沿进行计数。

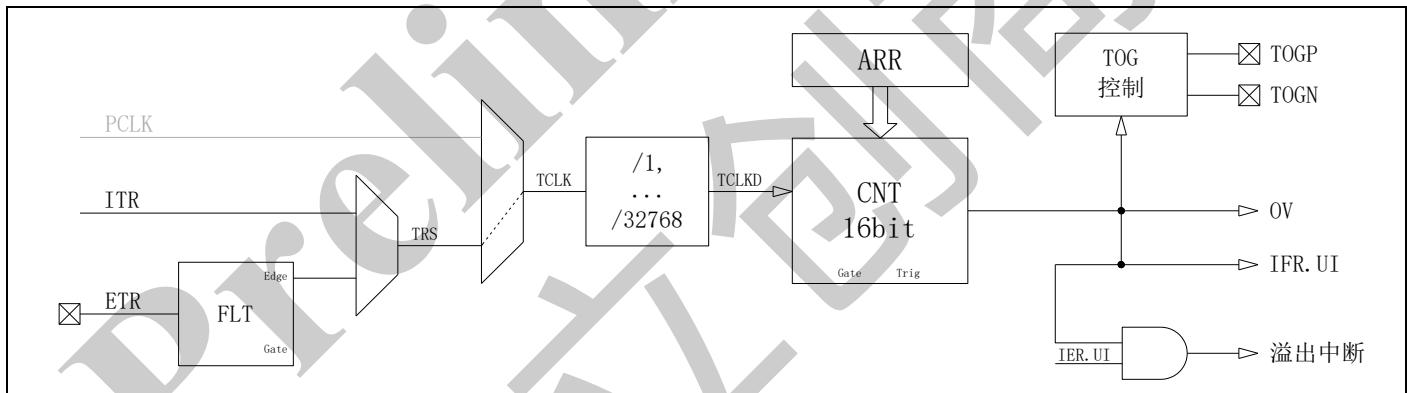


图 11-4 计数器模式框图

### 11.3.6 触发启动模式

当设置 BTIMx\_CR.MD 为 0x02 时，BTIMx 工作于触发启动模式，计数时钟为 PCLK 分频后的时钟 TCLKD。该模式下定时器具有两种启动方法：设置 BTIMx\_CR.CEN 为 1 或 TRS 信号上出现符合预期的电平变化。

启动后，计数器即从初值开始加计数，计数到 ARR 后产生溢出并重新从 0 开始计数；在任意时刻设置 BTIMx\_CR.CEN 为 0，则可以停止定时器。

通过 BTIMx\_CRTRS 选择 TRS 信号来源为 ETR 或 ITR。

通过 BTIMx\_CR.ETP 可以选择 ETR 管脚输入信号的上升沿或下降沿启动定时器。

每次检测到符合预期的触发信号时, BTIMx\_IFR.TI 会被硬件置 1, 若使能相应中断则可以产生触发中断, 用户应在中断服务程序中清除该触发标志。

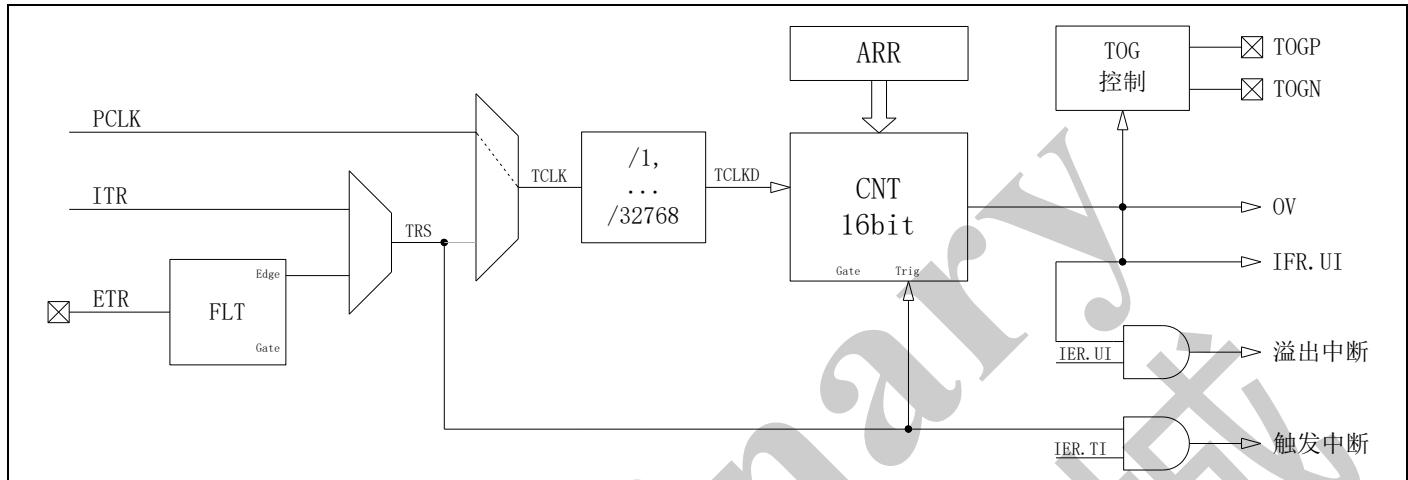


图 11-5 触发启动器模式框图

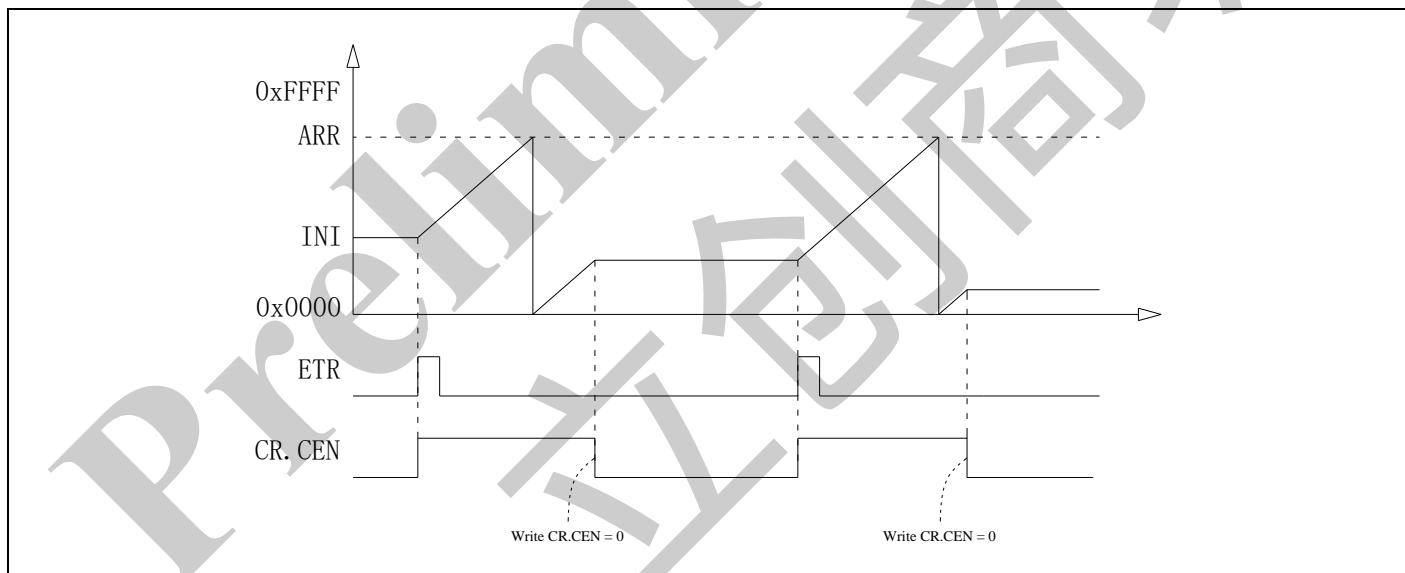


图 11-6 触发启动模式计数示意图

### 11.3.7 门控模式

当设置 BTIMx\_CR.MD 为 0x03 时, BTIMx 工作于门控模式, 计数时钟为 PCLK 分频后的时钟 TCLKD。该模式下定时器需要满足两个条件才会进行计数: BTIMx\_CR.CEN 为 1、ETR 管脚上的电平符合预期。定时器运行时, 计数器从初值开始加计数, 计数到 ARR 后产生溢出并重新从 0 开始计数。

通过 BTIMx\_CR.ETP 可以选择 ETR 管脚输入信号的高电平或低电平为计数电平。

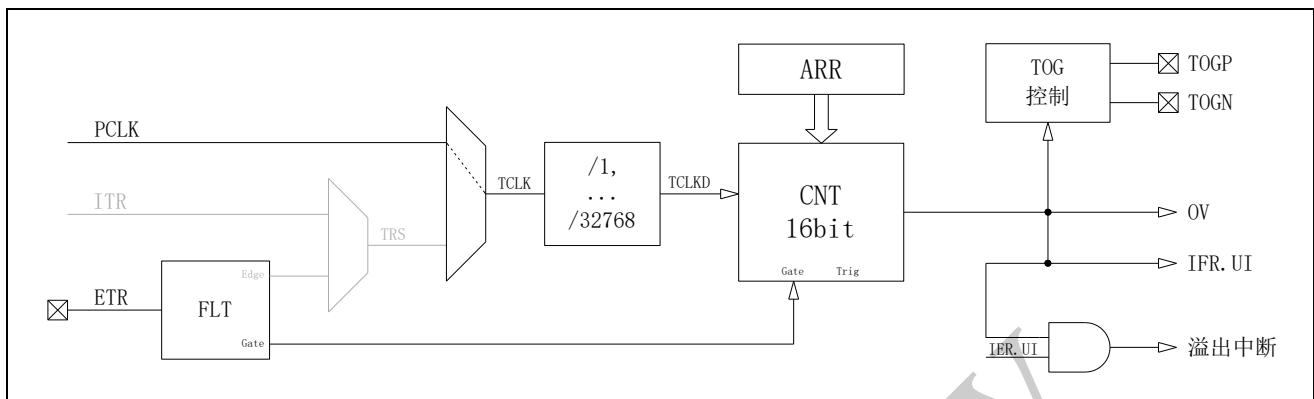


图 11-7 门控模式框图

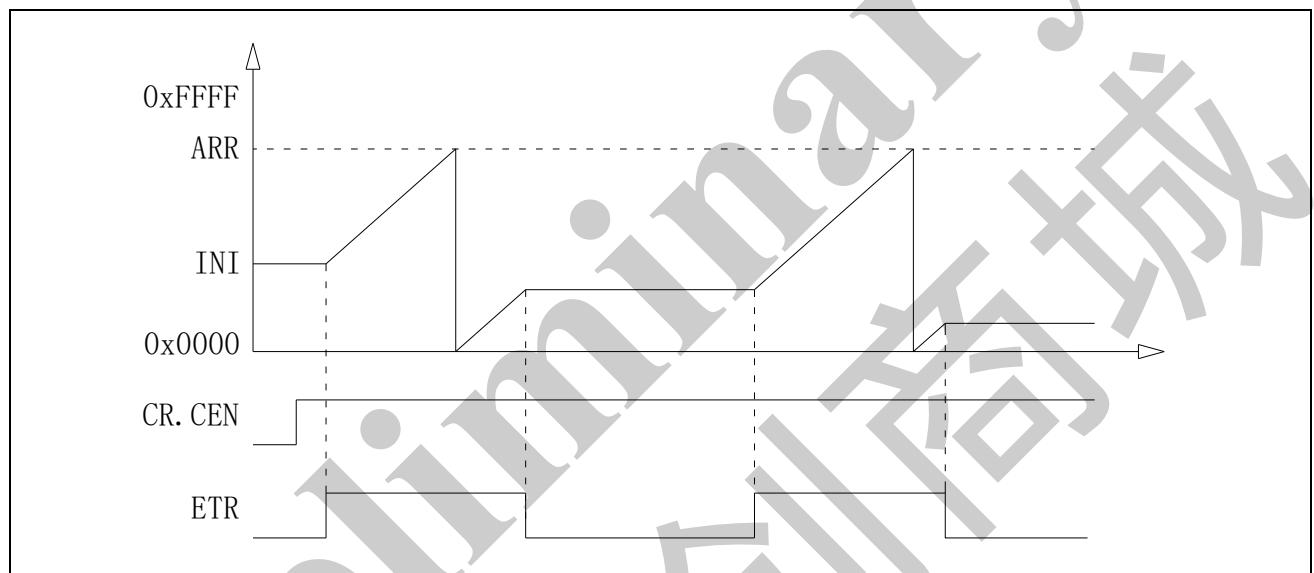
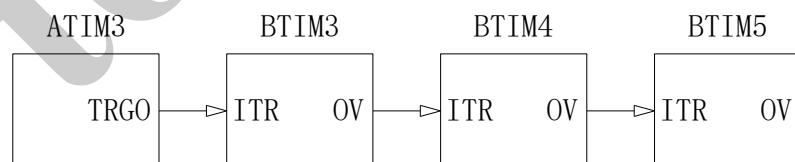


图 11-8 门控模式计数示意图

### 11.3.8 定时器级联

通过适当的配置 BTIM 可以级联使用，如将两个 16 位的定时器组合成一个 32 位的定时器。级联使用时，第一级定时器的计数时钟可以是 PCLK 或来自 ETR 管脚的输入信号，后级定时器的计数时钟为前级定时器的溢出信号。信号级联已在芯片内固化，其级联方式如下所示。



## 11.4 寄存器描述

名称	基地址	描述
BTIM3	0x40007400	BTIM3 基地址
BTIM4	0x40007500	BTIM4 基地址
BTIM5	0x40007600	BTIM5 基地址

注意：BTIM3、BTIM4、BTIM5 在 SysCtrl\_CR1.GTIMCFG 为 1 时才有效。

寄存器	偏移地址	描述
BTIMx_ARR	0x00	BTIMx 重载寄存器
BTIMx_CNT	0x04	BTIMx 计数器寄存器
BTIMx_CR	0x10	BTIMx 控制寄存器
BTIMx_IER	0x14	BTIMx 中断使能寄存器
BTIMx_IFR	0x18	BTIMx 中断标志寄存器
BTIMx_ICR	0x1C	BTIMx 中断清除寄存器
BTIM345_AIFR	0x48	BTIM345 复合中断标志寄存器
BTIM345_AICR	0x4C	BTIM345 复合中断清除寄存器

表 11-1 BTIM 寄存器列表

### 11.4.1 重载寄存器 (BTIMx\_ARR) (x=3,4,5)

偏移地址: 0x000

复位值: 0x0000 FFFF

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

位	符号	描述
31:16	Reserved	保留位
15:0	ARR	定时器重载值

### 11.4.2 计数寄存器 (BTIMx\_CNT) (x=3,4,5)

偏移地址: 0x004

复位值: 0x0000 0000

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

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	定时器计数值

### 11.4.3 控制寄存器 (BTIMx\_CR) (x=3,4,5)

偏移地址: 0x010

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ETP	TRS	OST	PRS			TOGEN	MD	CEN			
				RW	RW	RW	RW			RW	RW	RW			

位	符号	描述
31:12	Reserved	保留位
11	ETP	外部管脚输入的ETR信号极性选择 0: ETR正向(触发模式上升沿有效, 门控模式高电平有效) 1: ETR反向(触发模式下降沿有效, 门控模式低电平有效)
10:9	TRS	触发源选择 00: ETR管脚输入的信号 01: ITR, 详见级联功能章节 注意: BTIM3-5共用另一个ETR管脚
8	OST	单次/连续计数模式控制 0: 连续计数模式 1: 单次计数模式
7:4	PRS	预分频器分频系数配置 0: DIV1 4: DIV16 8: DIV256 12: DIV4096 1: DIV2 5: DIV32 9: DIV512 13: DIV8192 2: DIV4 6: DIV64 10: DIV1024 14: DIV16384 3: DIV8 7: DIV128 11: DIV2048 15: DIV32768
3	TOGEN	TOG管脚输出使能控制 1: TOGP、TOGN输出电平相反的信号 0: TOGP、TOGN输出电平均为0
2:1	MD	定时器工作模式配置 00: 定时器模式, 计数时钟为PCLK 01: 计数器模式, 计数时钟为TRS信号 10: 触发启动模式, 计数时钟为PCLK, TRS信号触发计数器启动 11: 门控模式, 计数时钟为PCLK, ETR管脚输入信号作为门控
0	CEN	定时器使能控制 0: 定时器停止 1: 定时器使能 注意: 触发启动模式触发后, CEN由硬件置1

#### 11.4.4 中断使能 (BTIMx\_IER) (x=3,4,5)

偏移地址: 0x014

复位值: 0x0000 0000

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

位	符号	描述
31:2	Reserved	保留位
1	TI	触发中断使能控制 1: 使能触发中断 0: 禁止触发中断
0	UI	溢出中断使能控制 1: 使能溢出中断 0: 禁止溢出中断

#### 11.4.5 中断标志寄存器 (BTIMx\_IFR) (x=3,4,5)

偏移地址: 0x018

复位值: 0x0000 0000

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

位	符号	描述
31:2	Reserved	保留位
1	TI	触发标志 1: 已发生触发事件 0: 未发生触发事件
0	UI	计数器溢出标志 1: 计数器已溢出 0: 计数器未溢出

### 11.4.6 中断标志清除寄存器 (BTIMx\_ICR) (x=3,4,5)

偏移地址: 0x01C

复位值: 0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved																
															TI	UI
															RIW0	R1W0

位	符号	描述
31:2	Reserved	保留位
1	TI	触发标志清除 1: 无功能 0: 清除触发标志
0	UI	计数器溢出标志清除 1: 无功能 0: 清除计数器溢出标志

### 11.4.7 复合中断标志寄存器 (BTIM345\_AIFR)

偏移地址: 0x048

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										TI5	UI5	TI4	UI4	TI3	UI3
										RO	RO	RO	RO	RO	RO

位	符号	描述
31:6	Reserved	保留位
5	TI5	BTIM5触发标志 1: 已发生触发事件 0: 未发生触发事件
4	UI5	BTIM5溢出标志 1: 计数器已溢出 0: 计数器未溢出
3	TI4	BTIM4触发标志 1: 已发生触发事件 0: 未发生触发事件
2	UI4	BTIM4溢出标志 1: 计数器已溢出 0: 计数器未溢出
1	TI3	BTIM3触发标志 1: 已发生触发事件 0: 未发生触发事件
0	UI3	BTIM3溢出标志 1: 计数器已溢出 0: 计数器未溢出

### 11.4.8 复合中断标志清除寄存器 (BTIM345\_AICR)

偏移地址: 0x04C

复位值: 0x0000 003F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										TI5	UI5	TI4	UI4	TI3	UI3
										R1W0	R1W0	R1W0	R1W0	R1W0	R1W0

位	符号	描述
31:6	Reserved	保留位
5	TI5	BTIM5触发标志清除 1: 无功能 0: 清除触发标志
4	UI5	BTIM5溢出标志清除 1: 无功能 0: 清除溢出标志
3	TI4	BTIM4触发标志清除 1: 无功能 0: 清除触发标志
2	UI4	BTIM4溢出标志清除 1: 无功能 0: 清除溢出标志
1	TI3	BTIM3触发标志清除 1: 无功能 0: 清除触发标志
0	UI3	BTIM3溢出标志清除 1: 无功能 0: 清除溢出标志

## 12 通用定时器 (GTIM)

### 12.1 概述

GTIM 是一个可具有 4 路比较捕获功能的 16 位定时器。BTIM3/4/5 与 GTIM 功能不可同时使用，由 SysCtrl\_CR1.GTIMCFG 控制，该位为 0 时 GTIM 有效。

### 12.2 主要特性

- 16bit 定时器
- 2 个翻转输出
- 单脉冲模式
- 外部输入计数功能
- 门控功能
- 外部触发功能
- 4 路独立 PWM 输出，调整范围 0% ~ 100%
- 4 路独立捕获
- 级联功能

## 12.3 功能描述

### 12.3.1 功能框图

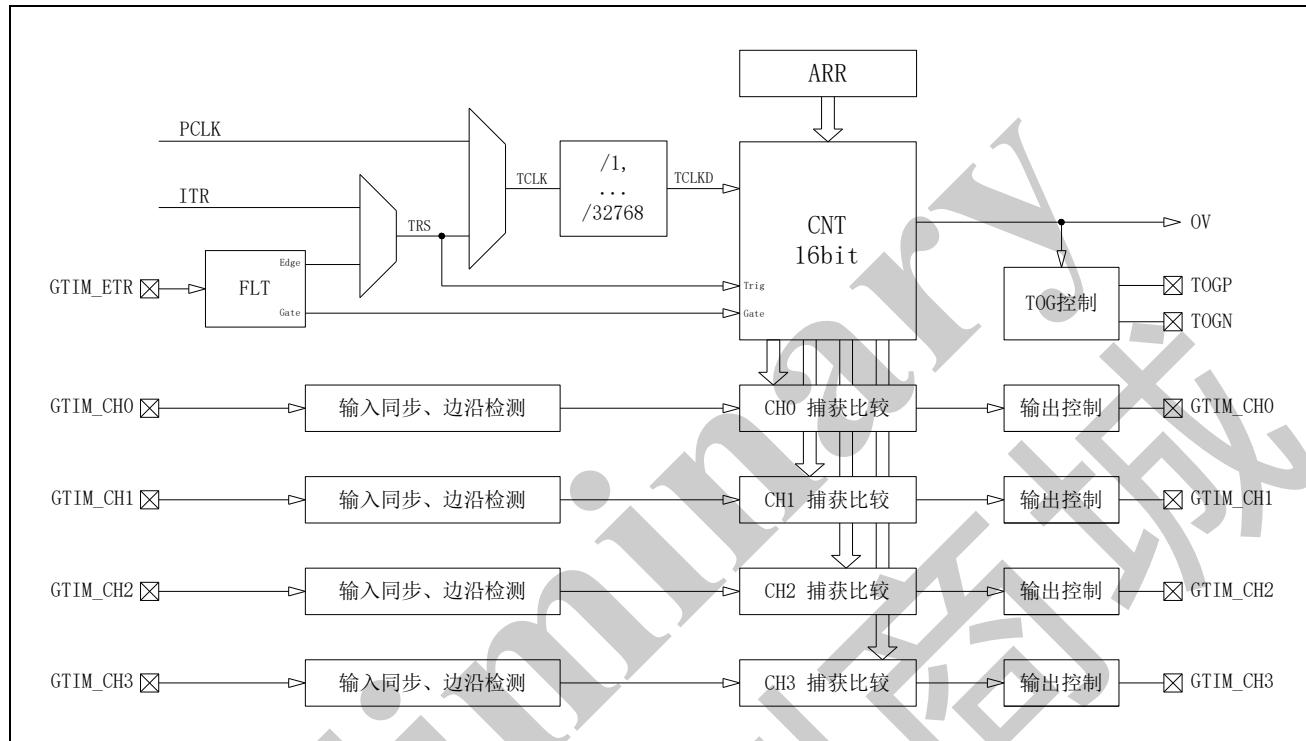
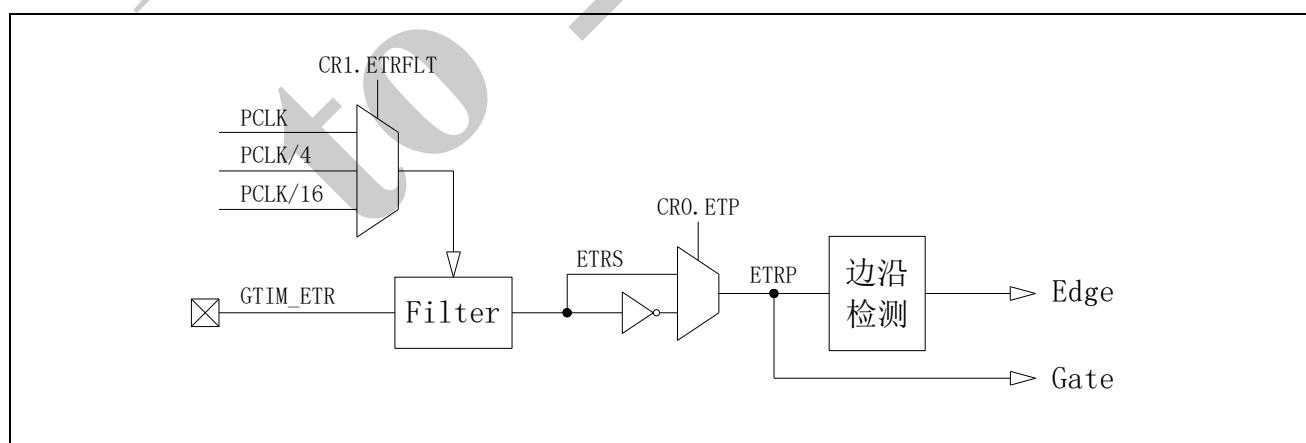


图 12-1 GTIM 框图

注：ITR 信号来源详见级联功能章节

### 12.3.2 滤波单元

外部管脚输入的 ETR 信号经同步滤波、极性选择、边沿检测后，输出门控信号及边沿信号供计数单元及触发单元使用。由于 ETR 信号采用滤波时钟进行同步滤波，故输入的 ETR 信号的频率应低于滤波时钟的频率的二分之一。



### 12.3.3 计数单元

在每个 TCLKD 信号的上升沿，内部计数器 CNT 自动加 1。计数到 ARR 后产生溢出信号 OV (宽度为一个 PCLK)，并复位内部计数值为 0 并再次开始加计数。计数器的计数范围为 0 至 ARR，计数单元的计数周期为 ARR+1。

定时器支持 2 种计数模式，One-Shot 模式与连续计数模式。当设置 GTIM\_CR0.OST 为 1 时，定时器工作于 One-Shot 模式；发生溢出后定时器立即停止计数，硬件自动清除 CEN。当设置 GTIM\_CR0.OST 为 0 时，定时器工作于连续计数模式；只要 GTIM\_CR0.CEN 为 1，定时器会一直进行计数。

每次溢出时，GTIM\_IFR.UI 会被硬件置 1，若使能相应中断则可以产生溢出中断，用户应在中断服务程序中清除该溢出标志。

若 GTIM\_CR0.TOGEN 为 1，则每溢出一次 TOGP/TOGN 输出电平翻转一次；用户可通过 GPIO 的辅助功能将 TOGP/TOGN 信号输出到管脚用于驱动外部电路。

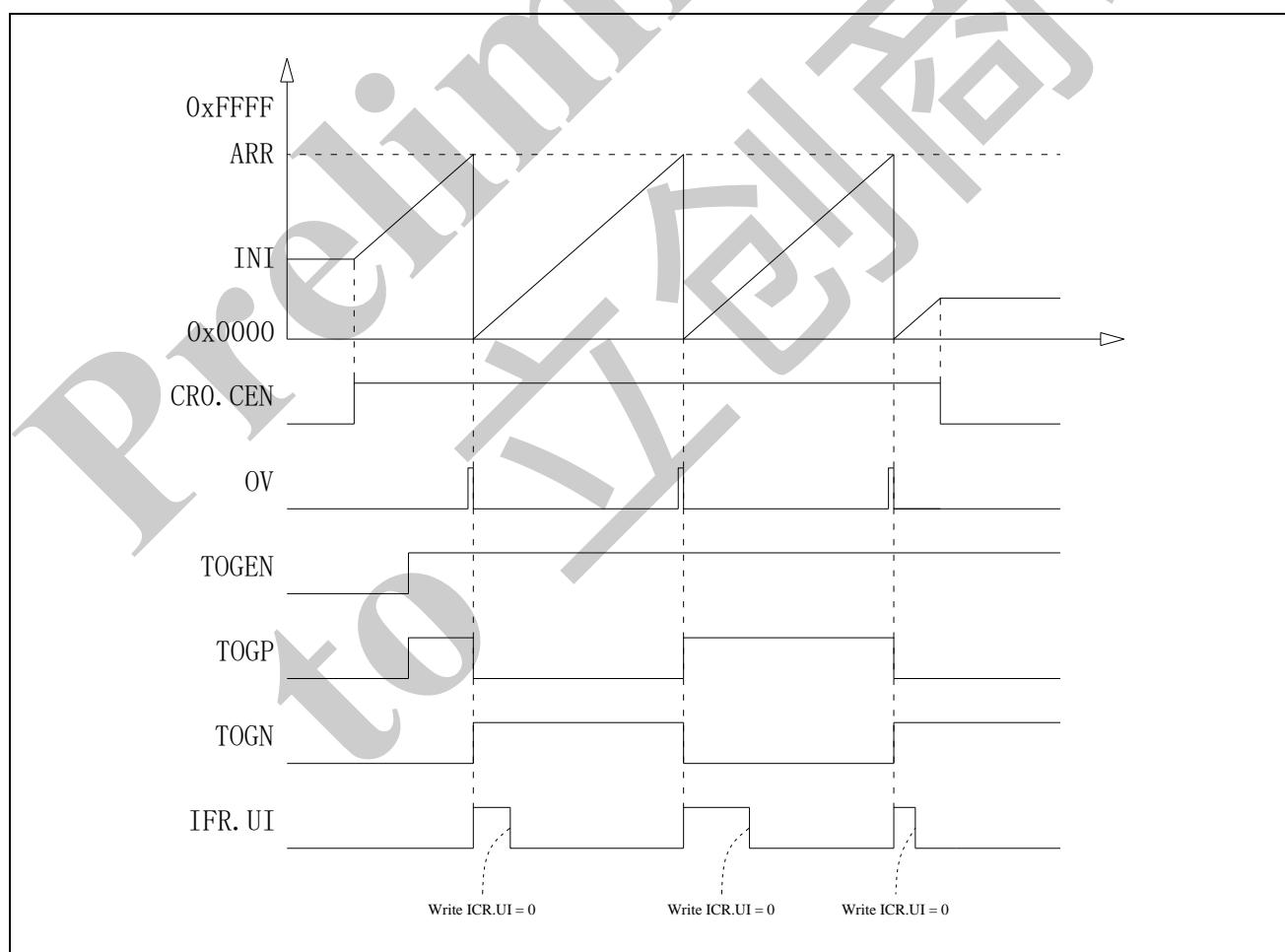


图 12-2 计数器波形

### 12.3.4 定时器模式

当设置 GTIM\_CR0.MD 为 0x00 时, GTIM 工作于定时器模式。在该模式下, 计数时钟为 PCLK 分频后的 TCLKD, 当设置 GTIM\_CR0.CEN 为 1 后, 定时器即从初值开始加计数, 计数到 ARR 后产生溢出并重新从 0 开始计数。

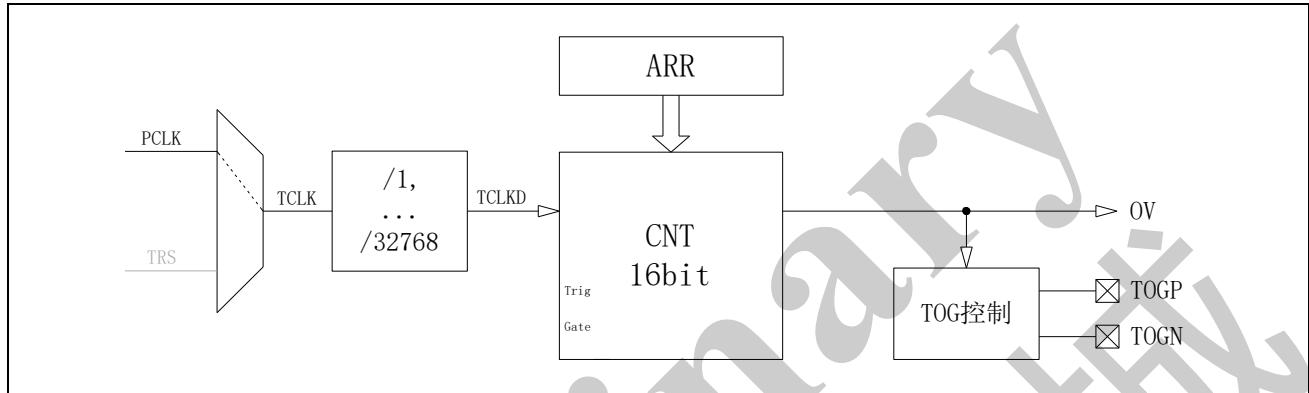


图 12-3 定时模式框图

### 12.3.5 计数器模式

当设置 GTIM\_CR0.MD 为 0x01 时, GTIM 工作于计数器模式。在该模式下, 计数时钟为分频后的 ITR 或 ETR 管脚输入的信号。当设置 GTIM\_CR0.CEN 为 1 后, 计数器即从初值开始加计数, 计数到 ARR 后产生溢出并重新从 0 开始计数。

通过 GTIM\_CR0.TRS 可以选择计数时钟源为 ITR 或 ETR; 通过 GTIM\_CR0.ETP 可以选择对 ETR 管脚输入信号的上升沿或下降沿进行计数。

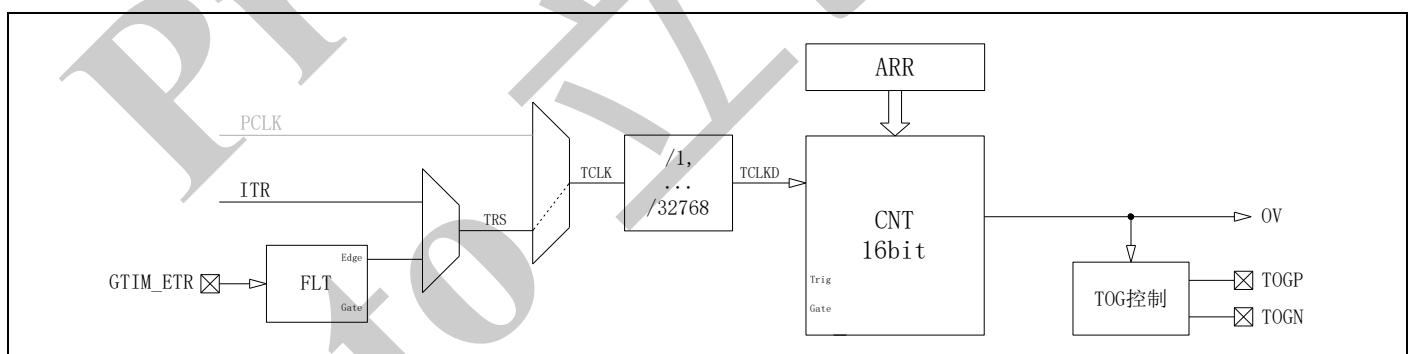


图 12-4 计数器模式框图

### 12.3.6 触发启动模式

当设置 GTIM\_CR0.MD 为 0x02 时, GTIM 工作于触发启动模式, 计数时钟为 PCLK 分频后的 TCLKD。该模式下定时器具有两种启动方法: 设置 GTIM\_CR0.CEN 为 1 或

TRS 信号上符合预期的电平变化。

启动后,计数器即从初值开始加计数,计数到 ARR 后产生溢出并重新从 0 开始计数;在任意时刻设置 GTIM\_CR0.CEN 为 0, 则可以停止定时器。

通过 GTIM\_CR0.TRS 选择 TRS 信号来源为 ETR 或 ITR。

通过 GTIM\_CR0.ETP 可以选择 ETR 管脚输入信号的上升沿或下降沿启动定时器。

每次检测到符合预期的触发信号时, GTIM\_IFR.TI 会被硬件置 1, 若使能相应中断则可以产生触发中断, 用户应在中断服务程序中清除该触发标志。

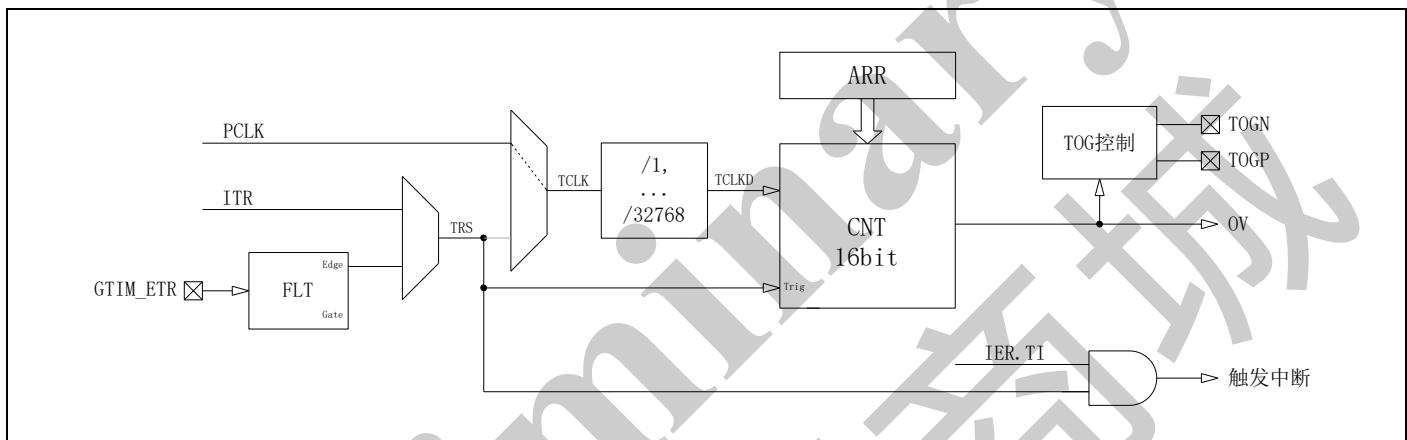


图 12-5 触发启动器模式框图

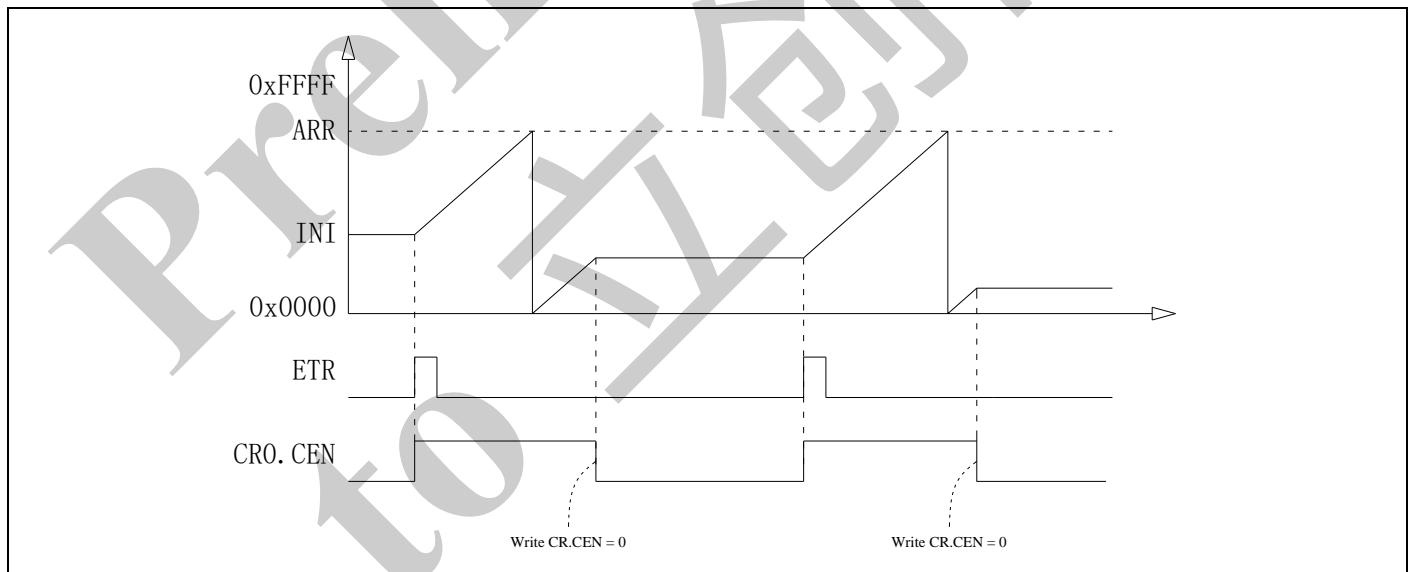


图 12-6 触发启动模式计数示意图

### 12.3.7 门控模式

当设置 GTIM\_CR0.MD 为 0x03 时, GTIM 工作于门控模式, 计数时钟为 PCLK 分频后的 TCLKD。该模式下定时器需要满足两个条件才会进行计数: GTIM\_CR0.CEN 为

1、ETR 管脚上的电平符合预期。定时器运行时，计数器从初值开始加计数，计数到 ARR 后产生溢出并重新从 0 开始计数。

通过 GTIM\_CR0.ETP 可以选择 ETR 管脚输入信号的高电平或低电平为计数电平。

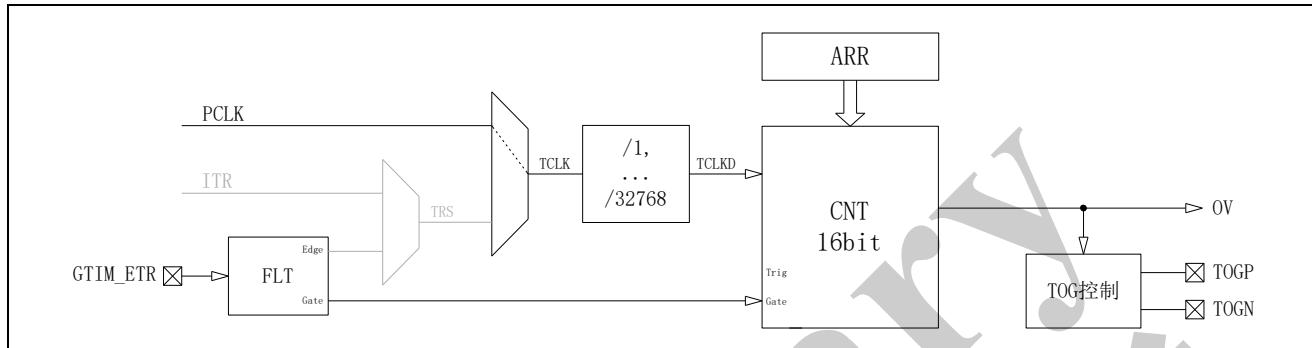


图 12-7 门控模式框图

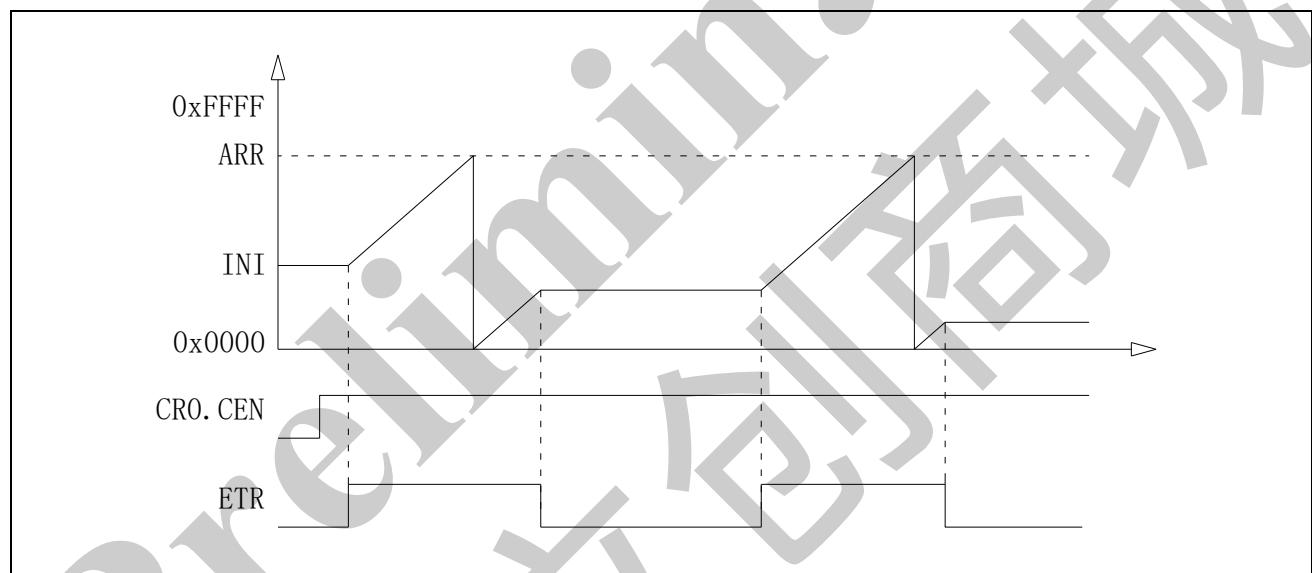


图 12-8 门控模式计数示意图

### 12.3.8 比较捕获功能

GTIM 具有 4 路比较捕获通道，CH0 ~ CH3。

#### 12.3.8.1 捕获功能

当配置 CMMR.CCyM 为 1 ~ 3 时，CHy 工作于捕获模式，支持三种捕获方法：

- 001：上升沿捕获
- 010：下降沿捕获
- 011：上下沿同时捕获

当 CHy 通道上出现所期望的脉冲边沿时，硬件自动将当前的记数值写入 CCRy 寄存器

并置位 IFR.CCyF；若使能相应通道的中断，则产生比较捕获中断，用户应在中断服务程序中清除该中断标志。

下图给出了两个通道进行输入捕获时的示意。其中 CH0 在上升沿进行捕获；CH1 上升沿和下降沿同时进行捕获。

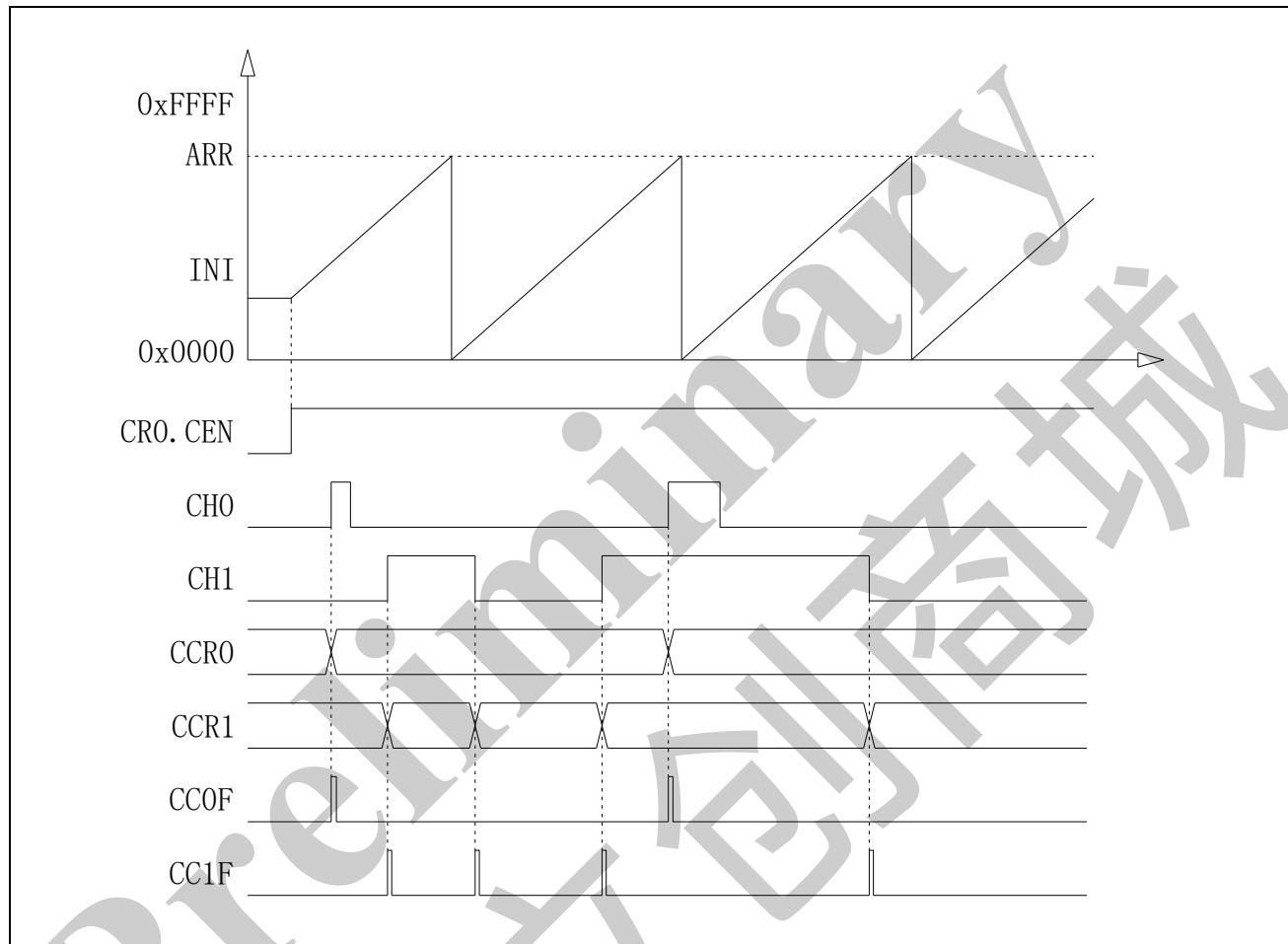


图 12-9 捕获功能示意图

### 12.3.8.2 比较功能

当配置 CMMR.CCyM 为 4~7 时，CHy 工作于比较模式，可以较出四种波形：

- 100：强制输出低电平
- 101：强制输出高电平
- 110：PWM 正向输出（ $CNT \geq CCRy$  输出高电平）
- 111：PWM 反向输出（ $CNT < CCRy$  输出高电平）

注：当  $ARR$  值不为  $0xFFFF$  时，可以实现 PWM 调整范围为  $0\% \sim 100\%$ 。

例：在正向输出模式  $ARR$  为 99 时， $CCR$  为 0~100 对应的占空比为  $100\% \sim 0\%$ 。

下图给出了两个通道进行比较输出时的示意。其中 CH0 采用 PWM 正向输出；CH1 采用 PWM 反向输出。

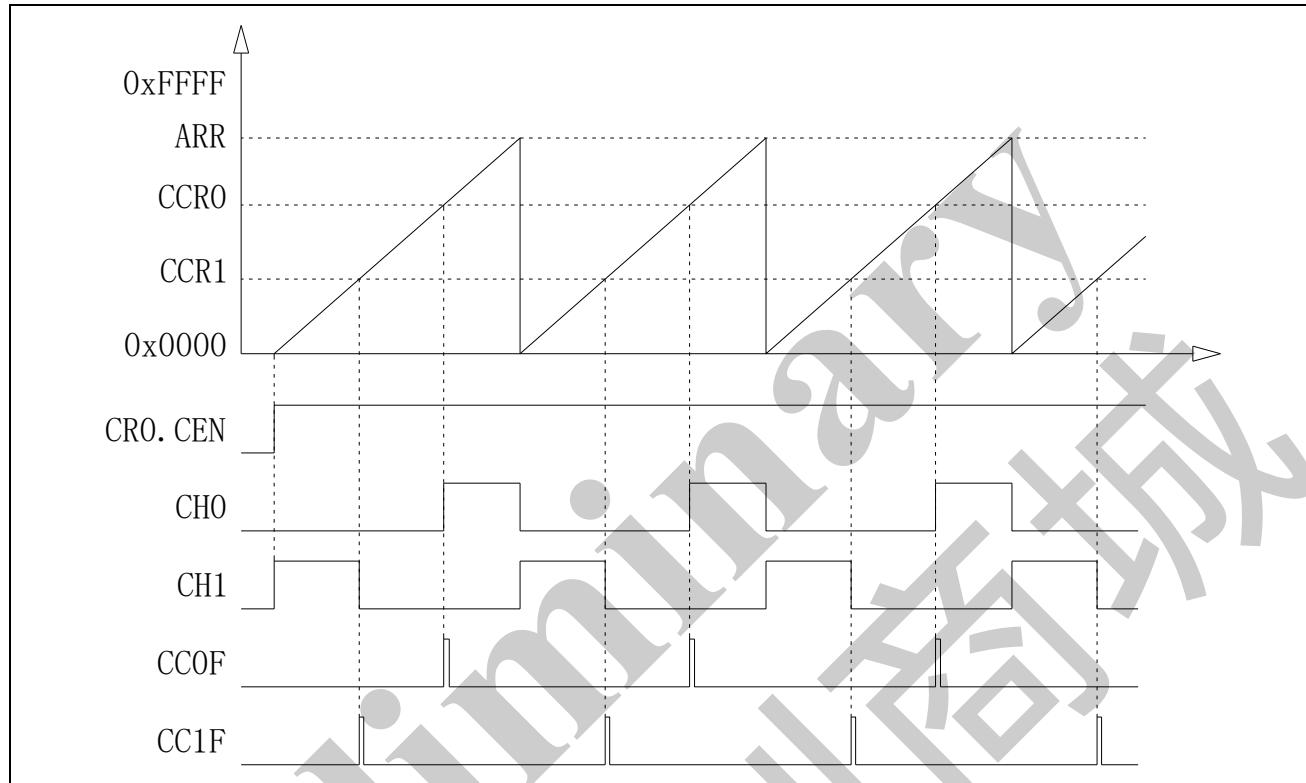


图 12-10 比较功能示意图

### 12.3.9 定时器级联

通过寄存器配置可以将 ATIM3 与 GTIM 配合为级联的 32 位定时器。ATIM3 定时器使用系统时钟 PCLK 作为定时器时钟，GTIM 选用前面的溢出作为外部脉冲计数并且选择外部计数功能。

### 12.3.10 片内外设互联

当 GPIOx\_CR4.GTIMCH0 为 0 时，GTIM\_CHx 信号来源由 GPIO\_AFRH/GPIO\_AFRL 决定；当 GPIOx\_CR4.GTIMCH0 为 1~6 时，GTIM\_CHx 信号的来源由下表决定。

GPIOx_CR4.GTIMCH0	GTIM_CH0来源
001	LPUART0_RX
010	LPUART1_RX
011	LVD_OUT

## 12.4 寄存器描述

Timer	基地址	描述
GTIM	0x40007400	Timer基地址

注：GTIM 在 SysCtrl\_CR1.GTIMCFG 为 0 时有效，为 1 时无效。

寄存器	偏移地址	描述
GTIM_ARR	0x300	重载寄存器
GTIM_CNT	0x304	计数寄存器
GTIM_CMMR	0x308	比较捕获控制寄存器
GTIM_CR1	0x30C	控制寄存器1
GTIM_CR0	0x310	控制寄存器0
GTIM_IER	0x314	中断使能控制寄存器
GTIM_IFR	0x318	中断标志存器
GTIM_ICR	0x31C	中断标志清除寄存器
GTIM_CCR0	0x320	CH0 比较/捕获寄存器
GTIM_CCR1	0x324	CH1 比较/捕获寄存器
GTIM_CCR2	0x328	CH2 比较/捕获寄存器
GTIM_CCR3	0x32C	CH3 比较/捕获寄存器

表 12-1 Timer 寄存器列表

### 12.4.1 重载寄存器 (GTIM\_ARR)

偏移地址: 0x300

复位值: 0x0000 FFFF

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

位	符号	描述
31:16	Reserved	保留位
15:0	ARR	定时器重载值

### 12.4.2 计数寄存器 (GTIM\_CNT)

偏移地址: 0x304

复位值: 0x0000 0000

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

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	定时器计数值

### 12.4.3 控制寄存器 1 (GTIM\_CR1)

偏移地址: 0x30C

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ETRFLT		Reserved				RW	

位	符号	描述
31:9	Reserved	保留位
8:7	Reserved	保持为0, 禁止修改
6:4	ETRFLT	ETR 管脚输入信号滤波配置; 采样时钟为PCLK或PCLK分频, 采样点个数为N 000: 无滤波 001: $F_{sample} = PCLK$ , N=2 010: $F_{sample} = PCLK$ , N=4 011: $F_{sample} = PCLK$ , N=6 100: $F_{sample} = PCLK/4$ , N=4 101: $F_{sample} = PCLK/4$ , N=6 110: $F_{sample} = PCLK/16$ , N=4 111: $F_{sample} = PCLK/16$ , N=6
3:0	Reserved	保留位

#### 12.4.4 控制寄存器 0 (GTIM\_CR0)

偏移地址: 0x310

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ETP	TRS	OST	PRS			TOGEN	MD	CEN			
				RW	RW	RW	RW			RW	RW	RW			

位	符号	描述
31:12	Reserved	保留位
11	ETP	外部输入ETR极性选择 0: 外部输入ETR正向(触发模式上升沿有效, 门控模式高电平有效) 1: 外部输入ETR反向(触发模式下降沿有效, 门控模式低电平有效)
10:9	TRS	触发源选择 00: ETR管脚输入的信号 01: ITR管脚输入的信号(ATIM3_TRGO) 1x: 保留
8	OST	单次/连续计数模式控制 0: 连续计数模式 1: 单次计数模式
7:4	PRS	预分频器分频系数配置 0: DIV1 4: DIV16 8: DIV256 12: DIV4096 1: DIV2 5: DIV32 9: DIV512 13: DIV8192 2: DIV4 6: DIV64 10: DIV1024 14: DIV16384 3: DIV8 7: DIV128 11: DIV2048 15: DIV32768
3	TOGEN	TOG管脚输出使能控制 1: TOGP、TOGN输出电平相反的信号 0: TOGP、TOGN输出电平均为0
2:1	MD	定时器工作模式配置 00: 定时器模式, 计数时钟为PCLK 01: 计数器模式, 计数时钟来自TRS选择信号 10: 触发启动模式, 计数时钟为PCLK, TRS信号触发计数器启动 11: 门控模式, 计数时钟为PCLK, TRS信号作为门控
0	CEN	定时器使能控制 0: 定时器停止 1: 定时器使能 注意: 触发启动模式触发后, CEN由硬件置1

### 12.4.5 中断使能控制寄存器 (GTIM\_IER)

偏移地址: 0x314

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										CC3	CC2	CC1	CC0	TI	UI
										RW	RW	RW	RW	RW	RW

位	符号	描述
31:6	Reserved	保留位
5	CC3	CH3捕获比较中断使能控制 1: 使能 0: 禁止
4	CC2	CH2捕获比较中断使能控制 1: 使能 0: 禁止
3	CC1	CH1捕获比较中断使能控制 1: 使能 0: 禁止
2	CC0	CH0捕获比较中断使能控制 1: 使能 0: 禁止
1	TI	触发中断使能控制 1: 使能 0: 禁止
0	UI	溢出中断使能控制 1: 使能 0: 禁止

### 12.4.6 中断标志寄存器 (GTIM\_IFR)

偏移地址: 0x318

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										CC3	CC2	CC1	CC0	TI	UI
										RW	RW	RW	RW	RW	RW

位	符号	描述
31:6	Reserved	保留位
5	CC3	CH3比较捕获中断标志 1: 已发生比较捕获事件 0: 未发生比较捕获事件
4	CC2	CH2比较捕获中断标志 1: 已发生比较捕获事件 0: 未发生比较捕获事件
3	CC1	CH1比较捕获中断标志 1: 已发生比较捕获事件 0: 未发生比较捕获事件
2	CC0	CH0比较捕获中断标志 1: 已发生比较捕获事件 0: 未发生比较捕获事件
1	TI	触发中断标志 1: 已发生触发事件 0: 未发生触发事件
0	UI	溢出中断标志 1: 计数器已溢出 0: 计数器未溢出

### 12.4.7 中断标志清除寄存器 (GTIM\_ICR)

偏移地址: 0x31C

复位值: 0x0000 003F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										CC3	CC2	CC1	CC0	TI	UI
										RW	RW	RW	RW	RW	RW

位	符号	描述
31:6	Reserved	保留位
5	CC3	CH3比较捕获中断标志清除 1: 无功能 0: 清除比较捕获中断标志
4	CC2	CH2比较捕获中断标志清除 1: 无功能 0: 清除比较捕获中断标志
3	CC1	CH1比较捕获中断标志清除 1: 无功能 0: 清除比较捕获中断标志
2	CC0	CH0比较捕获中断标志清除 1: 无功能 0: 清除比较捕获中断标志
1	TI	触发中断标志清除 1: 无功能 0: 清除触发中断标志
0	UI	溢出中断标志清除 1: 无功能 0: 清除计数器溢出标志

## 12.4.8 比较捕获控制寄存器 (GTIM\_CMMR)

偏移地址: 0x308

复位值: 0x0000\_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	CC3M		Res	CC2M		Res.	CC1M		Res.	CC0M		Res	CC0M		RW
	RW			RW			RW			RW			RW		

位	符号	描述
31:15	Reserved	保留位
14:12	CC3M	CH3捕获比较模式配置 功能描述详见CC0M
11	Reserved	保留位
10:8	CC2M	CH2捕获比较模式配置 功能描述详见CC0M
7	Reserved	保留位
6:4	CC1M	CH1捕获比较模式配置 功能描述详见CC0M
3	Reserved	保留位
2:0	CC0M	CH0捕获比较模式配置 000: 无功能 001: 上升沿捕获 010: 下降沿捕获 011: 上下沿同时捕获 100: 强制输出低电平 101: 强制输出高电平 110: PWM正向输出 (CNT >= CCR0输出高电平) 111: PWM反向输出 (CNT < CCR输出高电平)

### 12.4.9 比较捕获寄存器 (GTIM\_CCRy) (y=0,1,2,3)

偏移地址: 0x320 / 0x324 / 0x328 / 0x32C

复位值: 0xFFFF / 0x0000 / FFFF / 0x0000

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

位	符号	描述
31:16	Reserved	保留位
15:0	CCR	16位比较捕获寄存器

## 13 高级定时器 (ATIM)

### 13.1 概述

ATIM3 是由 1 个计数单元和 7 个比较单元组成的定时器，支持 6 个独立 PWM 输出或 3 对互补 PWM 输出。支持 6 个捕获输入。

使用定时器预分频、系统预分频器和系统时钟选择，可灵活的调整脉冲宽度和波形周期；可以方便的测量脉冲宽度。

### 13.2 主要特性

- 6 路独立 PWM 输出 CH0A,CH0B, CH1A,CH1B, CH2A,CH2B
- 3 路互补 PWM 输出 (CHxA,CHxB)
- 3 路互补 PWM 输出 (CHxA,CHxB)
- 最多 6 路捕获输入
- 脉冲宽度测量
- 死区控制
- 刹车控制
- 边沿对齐、对称中心对齐与非对称中心对齐 PWM 输出
- 正交编码计数功能
- 单脉冲模式
- 外部输入计数功能

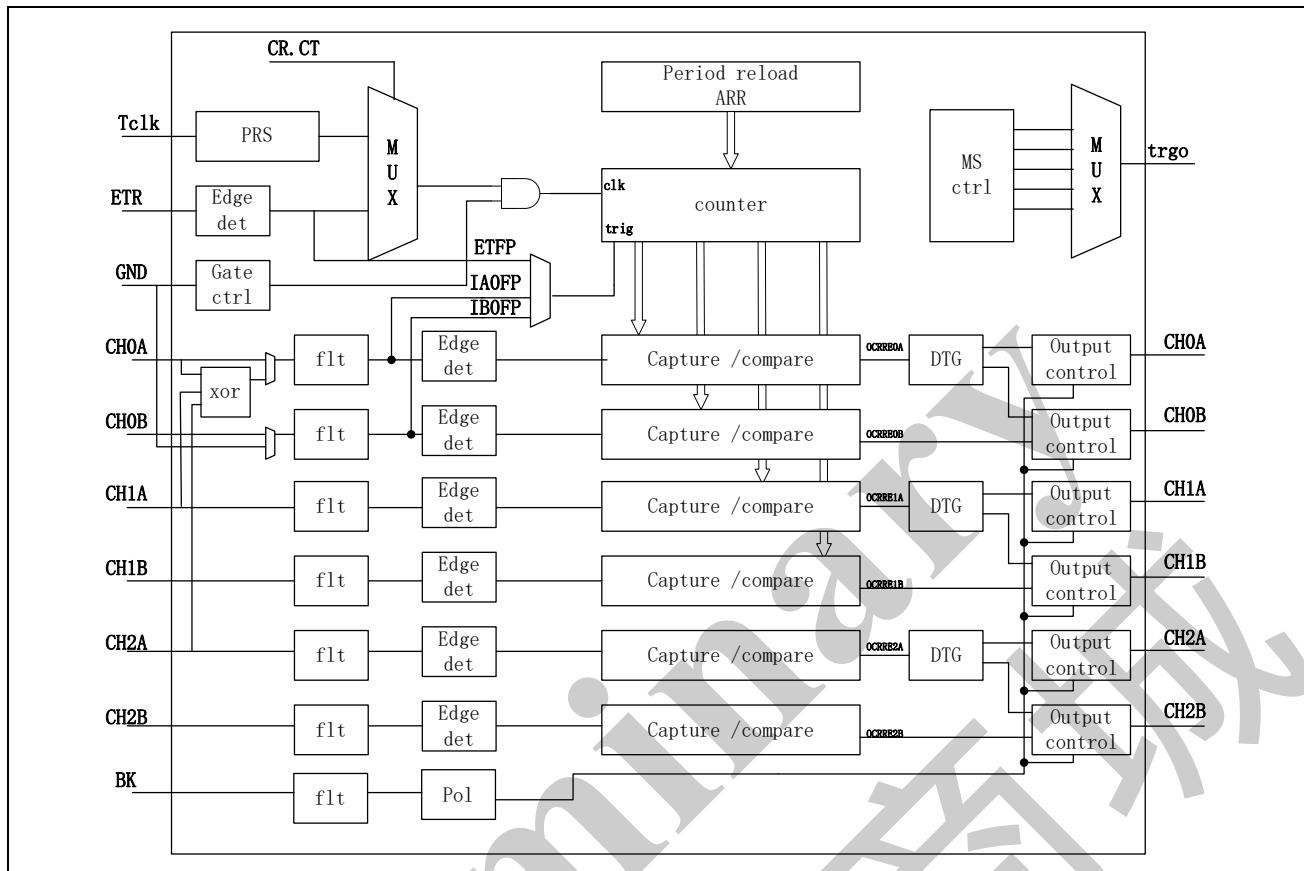


图 13-1 ATIM3 框图

Preliminary Release

## 13.3 功能描述

### 13.3.1 定时器时钟

定时器使用 PCLK 作为定时器时钟，下文定时器时钟使用 TCLK 标示。

### 13.3.2 定时计数器

高级定时器的主要模块是一个 16 位计数器及其相关的自动重载寄存器。计数器可递增计数、递减计数或交替进行递增和递减计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

### 13.3.3 定时器预分频

使用 TCLK 做为 Timer 定时器时钟时，可以使用预分频。预分频设置如下：

PRS	000	001	010	011	100	101	110	111
分频比	1	2	4	8	16	32	64	256

预分频器没有预装载缓存，因此对预分频进行更改，预分频将会立刻生效。

### 13.3.4 模式 0 计数定时器功能

这种模式下，计数器向上计数。计数器支持两种计数模式，重载模式与自由计数模式，可以选择外部时钟 ETR 计数或系统时钟计数。计数到最大值溢出产生中断。翻转输出端口 CHA,CHB，这种模式下 CHA,CHB 反相。

重载模式计数范围从 ARR 向上计数到 0xFFFF 溢出，然后从 ARR 再开始计数，计数周期为 0xFFFF-ARR+1；自由计数模式从设置计数值计数到 0xFFFFFFFF 后溢出，溢出后计数值从 0x0 重新开始计数。

#### 13.3.4.1 功能框图

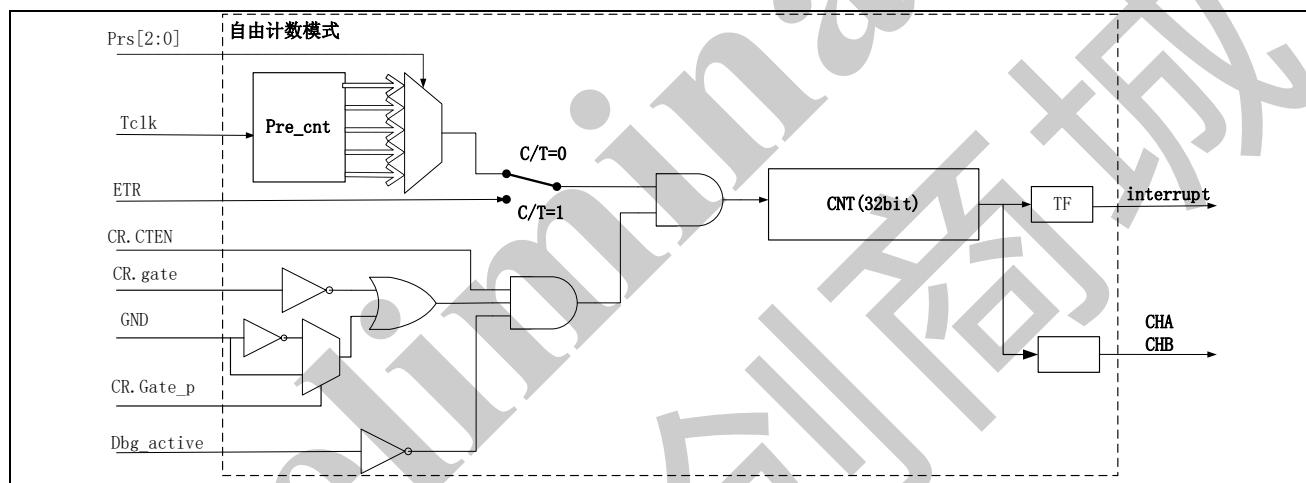


图 13-2 自由计数框图

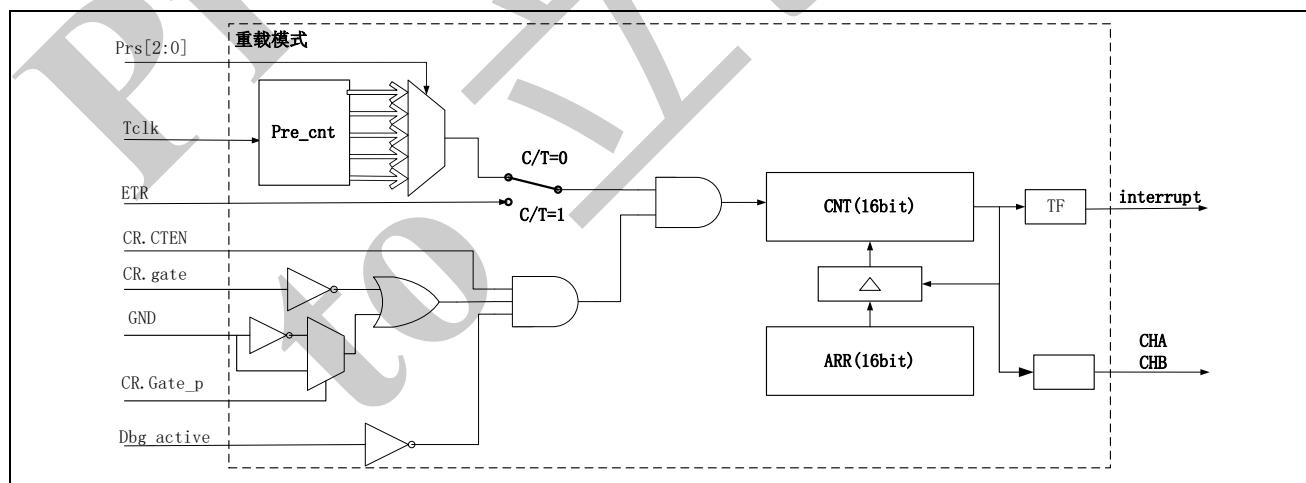


图 13-3 重载计数波形

### 13.3.4.2 计数波形

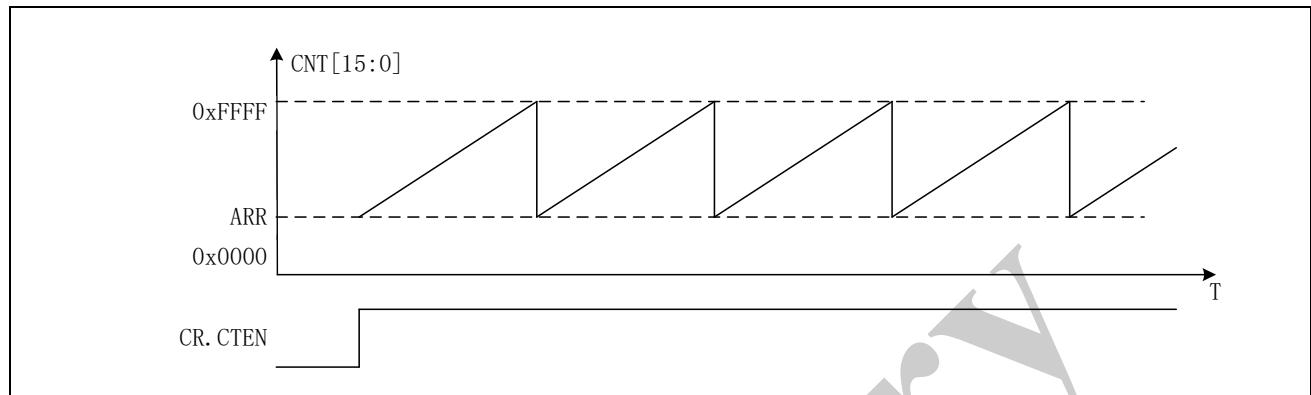


图 13-4 16位重载计数波形

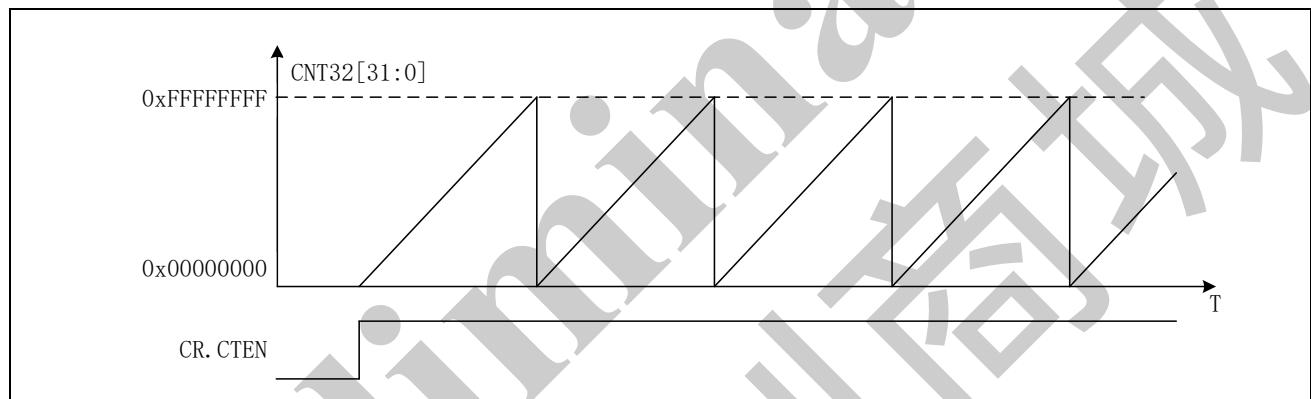


图 13-5 32位自由计数波形

### 13.3.4.3 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 TCLK 采样，因此外部输入时钟频率不能超过系统的 TCLK 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。

### 13.3.4.4 定时功能

定时功能用于产生间隔定时。在定时功能中，定时器有预除频，定时器在每个预除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。

### 13.3.4.5 时序图

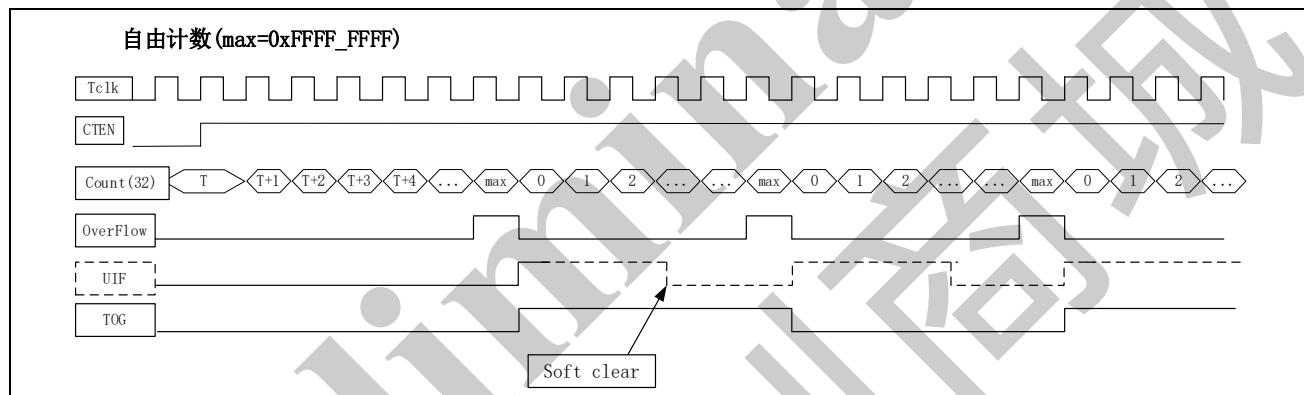


图 13-6 自由计数时序图

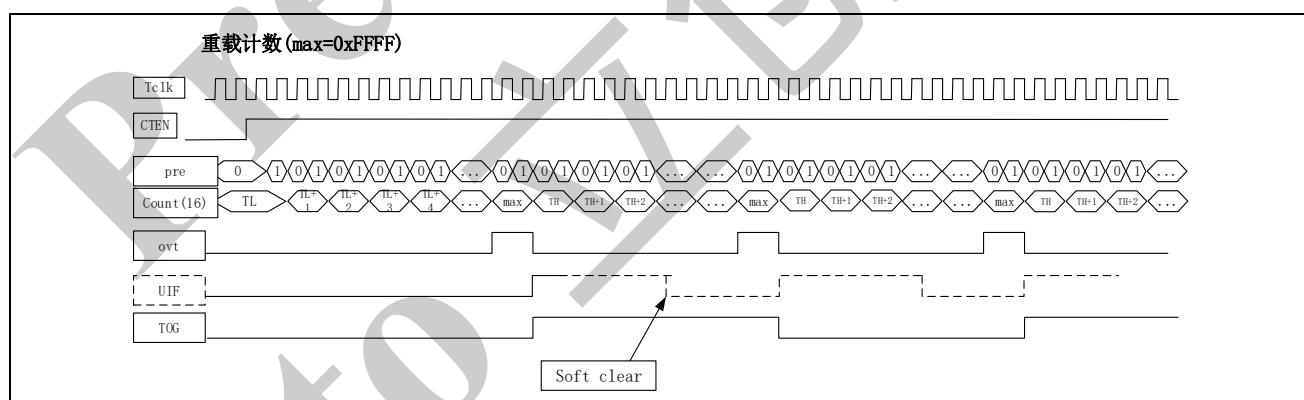


图 13-7 重载计数时序图 (预分频设置为2)

### 13.3.4.6 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。使用 toggle 输出需要使能 DTR.MOE 控制位。设置 CR.TOGEN 为 0 可以同时设置端口 CHA,CHB 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	计数器周期	计数器计数值	计数器重载值	CNTL 初始值	CNTH 重载值
1000Hz	0.5ms	2000	63536	0xF830	0xF830
2000Hz	0.25ms	1000	64536	0xFC18	0xFC18
4000Hz	0.125ms	500	65036	0xFE0C	0xFE0C

### 13.3.4.7 设置示例

#### 重载定时器设置

1. 设置定时器模式 M0CR.MODE=0
2. 设置装载值 ARR
3. 设置计数器初值 CNT
4. 清除中断标志
5. 使能中断 M0CR.UIE
6. 使能重载模式 M0CR.MD
7. 开启定时器 M0CR.CTEN

#### BUZZER 输出控制

1. 根据输出频率设置合适的 ARR 值
2. 设置定时器为重载模式，参考重载定时器设置
3. 使能输出使能 DTR.MOE
4. 启动另外一个定时器控制 M0CR.TOGEN 实现频率的间隔输出。

### 13.3.5 模式 1 脉宽测量 PWC

这种模式下可以自动测量输入脉冲的高电平低电平或者周期宽度。

第一个有效边沿计数器初始化为 0x0001，第二个有效边沿将停止计数，并将当前计数值存入 CMAR，并且产生捕获中断 CAF，如果计数器发生溢出，会产生溢出标志。设置溢出中断使能会产生溢出中断。

M1CR.edg1st	0	0	1	1
M1CR.edg2nd	0	1	0	1
脉宽测量	上沿~上沿 周期宽度	上沿~下沿 高电平宽度	下沿~上沿 低电平宽度	下沿~下沿 周期宽度

周期测量时，会间隔一个周期测量一个周期。

#### 13.3.5.1 PWC 功能框图

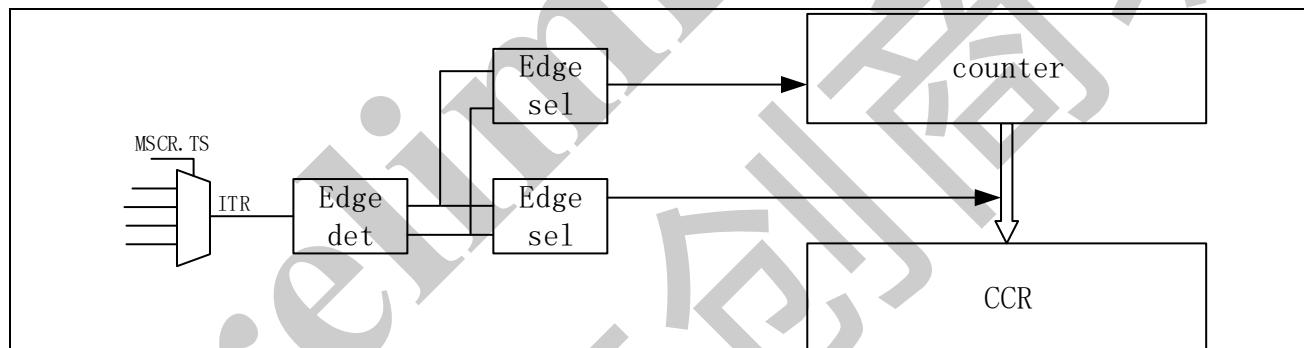


图 13-8 PWC测量框图

MSCR.TS	<p>触发选择</p> <p>000: 端口ETR的滤波相位选择后的信号ETFP；</p> <p>001: 内部互联信号 ITR0</p> <p>010: 内部互联信号 ITR1;</p> <p>011: 内部互联信号 ITR2;</p> <p>100: 内部互联信号 ITR3;</p> <p>101: 无效</p> <p>110: 端口CH0A的滤波后的信号IAFP（极性选择在脉宽测量模式下无效）</p> <p>111: 端口CH0B的滤波后的信号IBFP（极性选择在脉宽测量模式下无效）</p>
---------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 13.3.5.2 PWC 波形测量时序图

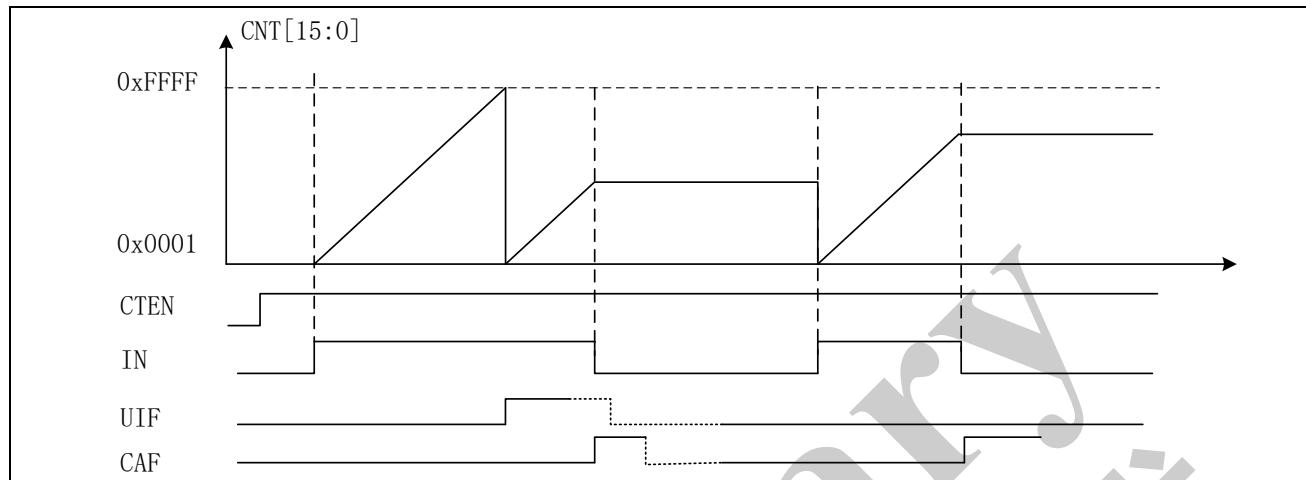


图 13-9 高电平脉冲宽度测量

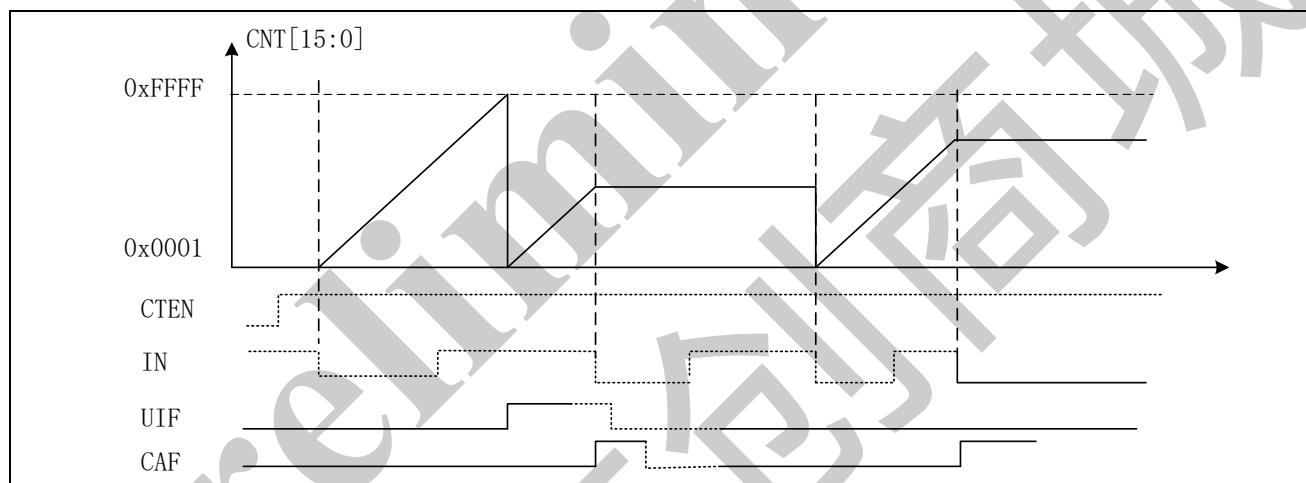


图 13-10 下降沿到下降沿周期测量

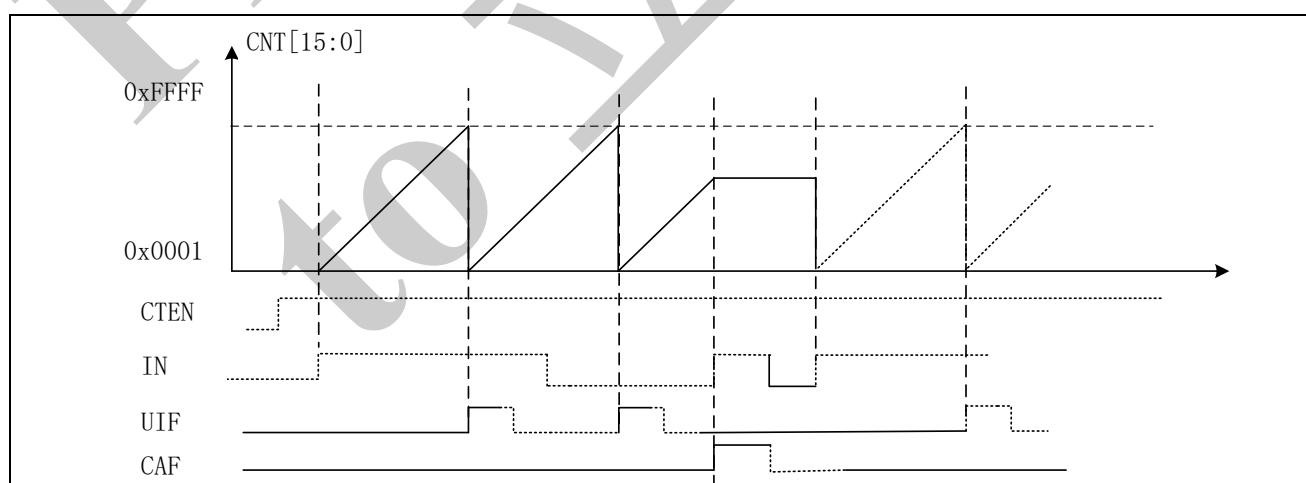


图 13-11 上升沿到上升沿周期测量

通过寄存器 MSCR.TS 选择测量信号源。

- 000 ETFP: ETR 外部输入滤波后的相位选择信号, 可选择外部滤波与输入反向  
001 ITR0: Timer 内部互联信号 0, 其他 timer 的 TRGO 输出  
010 ITR1: Timer 内部互联信号 1, 其他 timer 的 TRGO 输出  
011 ITR2: Timer 内部互联信号 2, 其他 timer 的 TRGO 输出  
100 ITR3: Timer 内部互联信号 3, 其他 timer 的 TRGO 输出  
101 IA0ED: 无效  
110 IAfp: CH0A 外部输入滤波后的相位选择信号, 可选择外部滤波与输入反向  
111 IBfp: CH0B 外部输入滤波后的相位选择信号, 可选择外部滤波与输入反向

	ITR0	ITR1	ITR2	ITR3
ATIM3	-	-	GTIM_OV	-

注: 关于 TRGO 输出参考寄存器 MSCR.MMS 描述。

Preliminary  
to be confirmed

### 13.3.5.3 PWC 单次触发模式

设置 M1CR.ONESHOT=1 可以设置 PWC 单次测量，测量完成后 CTEN 将被清除。

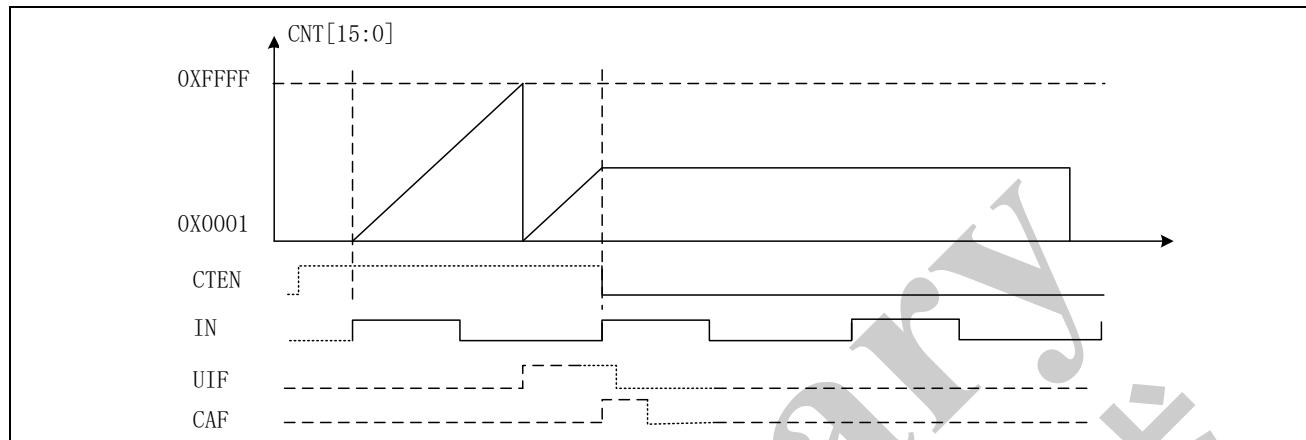


图 13-12 上升沿到上升沿周期测量单次模式

#### 13.3.5.1 设置示例

##### 脉冲低电平测量设置

1. 设置为脉冲测量模式 M1CR.MODE=1
2. 设置 MSCR.TS 选择测量的信号
3. 设置 M1CR.edg2dn=0,M1CR.edg1st=1 选择测量低电平
4. 清除中断标志
5. 使能溢出中断 M1CR.UIE
6. 使能测量结束中断 CR0.CIEA
7. 使能定时器 M1CR.CTEN
8. 中断服务程序中读取 CCR0A 及溢出次数并清除中断标志
9. 等待下次测量

##### 脉冲高电平单次测量设置

1. 设置为脉冲测量模式 M1CR.MODE=1
2. 设置 MSCR.TS 选择测量的信号
3. 设置脉冲单次测量模式 M1CR.ONESHOT=1
4. 设置 M1CR.edg2dn=1, M1CR.edg1st=0 选择测量低电平
5. 清除中断标志
6. 使能溢出中断 M1CR.UIE

7. 使能测量结束中断 CR0.CIEA
8. 使能定时器 M1CR.CTEN
9. 中断服务程序中读取 CCR0A 及溢出次数并清除中断标志
10. 测量结束

Preliminary  
to 立创商城

### 13.3.6 模式 2/3 比较捕获模式

#### 13.3.6.1 计数器

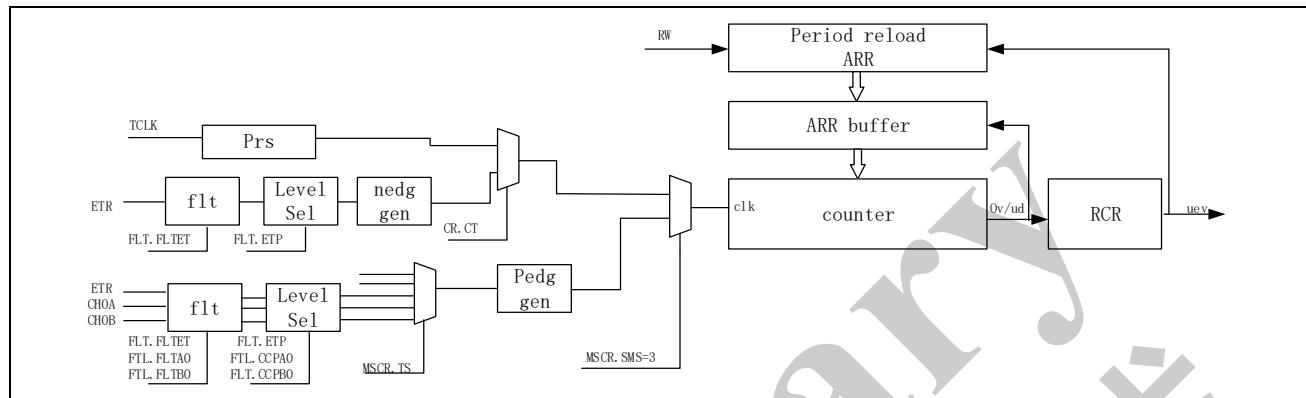


图 13-13 计数器框图

计数器主要部分是一个 16 位计数器与相关的自动装载寄存器。这个计数器可以向上计数（模式 2），向下计数（模式 2）或向上向下双向计数（模式 3）。计数器的时钟可以由预分频器 PRS 分频得到，也可以选择 ETR 输入外部时钟或者通过 MSCR.TS 选择的外部输入信号和内部互联信号。

- 计数器基本单元包括：
- 计数器寄存器 CNT
- 预分频寄存器 CR.PRS
- 自动装载寄存器 ARR
- 重复次数寄存器 RCR
- 时钟选择控制寄存器 FLT,CR0,MSCR,CR

自动装载寄存器具有缓存功能，计数器产生事件更新后重载值从缓存寄存器更新到计数器。当计数器停止状态或者缓存功能关闭状态，自动装载寄存器立刻更新到缓存寄存器。当定时器处于运行状态并且缓存功能有效时，写入到自动装载寄存器的值不会立刻更新到缓存寄存器，当事件更新后才从自动装载寄存器更新到缓存寄存器。

时钟选择及门控功能，触发功能，复位功能参考模式 2/3 从模式章节。

### 13.3.6.2 计数器波形

模式 2 为锯齿波计数波形，通过设置 CR.DIR 可以更改计数方向。

设置 CR.DIR 为 0 时，计数器为递增计数模式，这种模式下，计数器从 0 计数到自动重载值 (ATIMx\_ARR)，然后重新从 0 开始计数并生成计数器上溢事件。如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中编程的次数加一次 (ATIMx\_RCR+1) 后，将生成更新事件 (UEV)。否则，将在每次计数器上溢时产生更新事件。

将 ATIMx\_CR 寄存器的 UG 位置 1 (通过软件或使用从模式控制器) 时，也将产生更新事件。

发生更新事件时，将更新所有寄存器且将更新标志 (ATIMx\_IFR 寄存器中的 UIF 位) 置 1 (取决于 URS 位)：

- 自动重载缓存值将以 ARR 寄存器值进行更新
- 比较缓存值将以比较寄存器 CCRxy 进行更新

以下图示显示 ARR=0x2C 时不同计数方向的计数器波形：

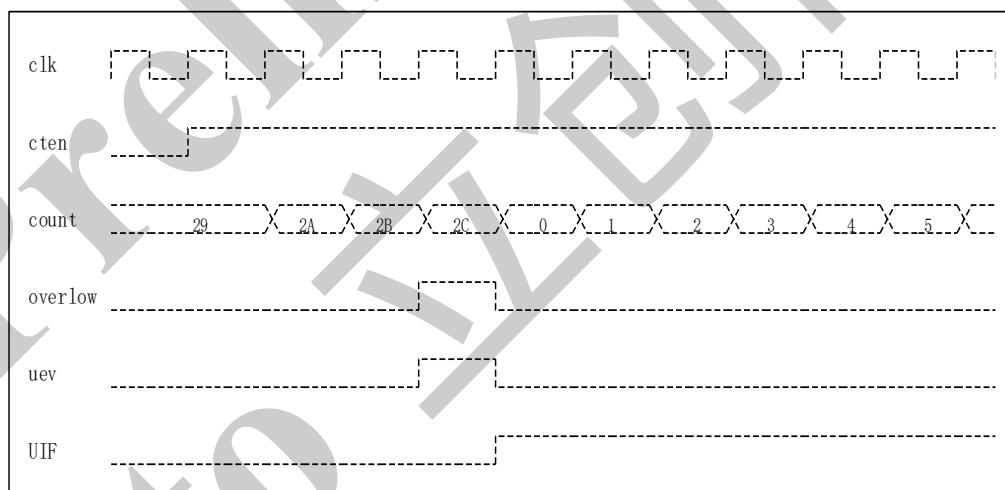


图 13-14 无预分频的向上计数

计数溢出周期时钟个数为 ARR+1，

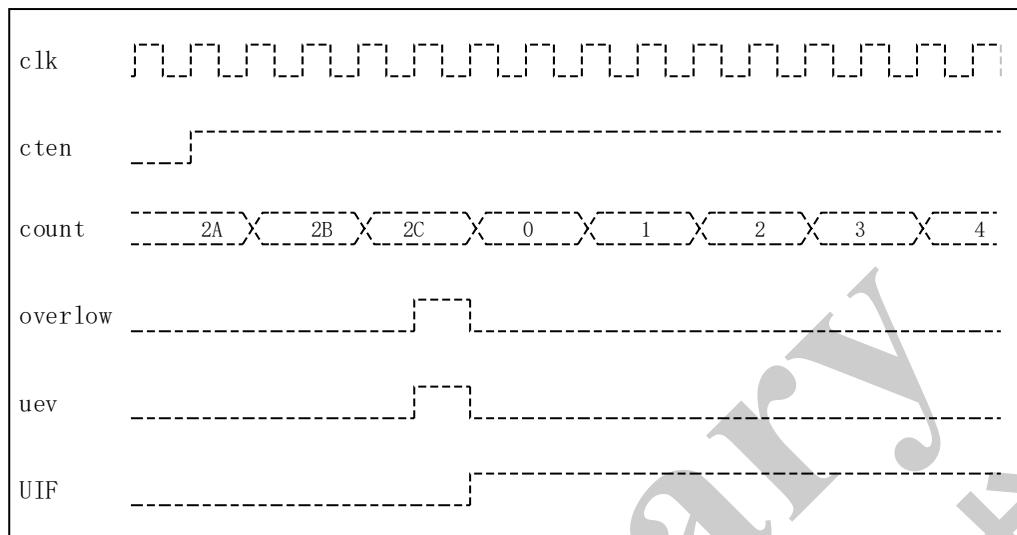


图 13-15 带预分频的上计数

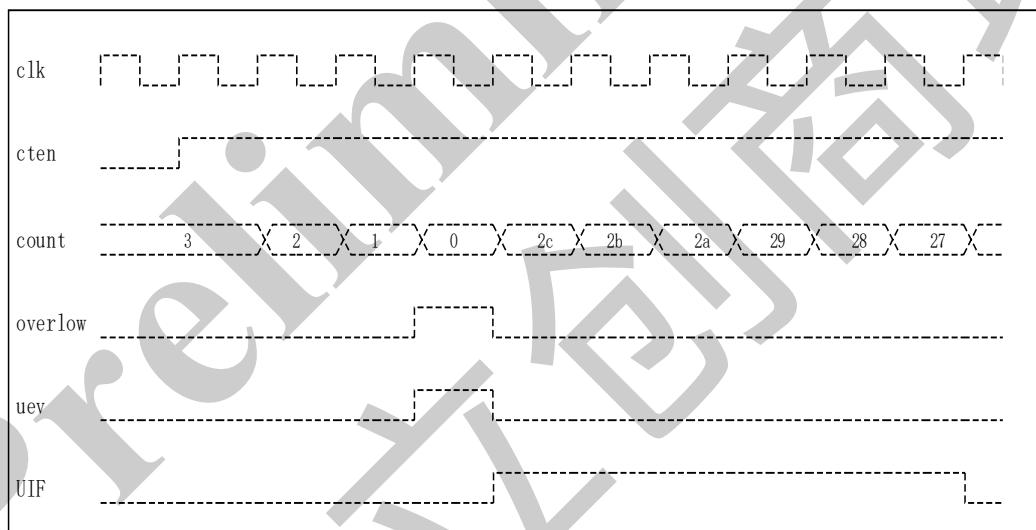


图 13-16 不带预分频的下计数

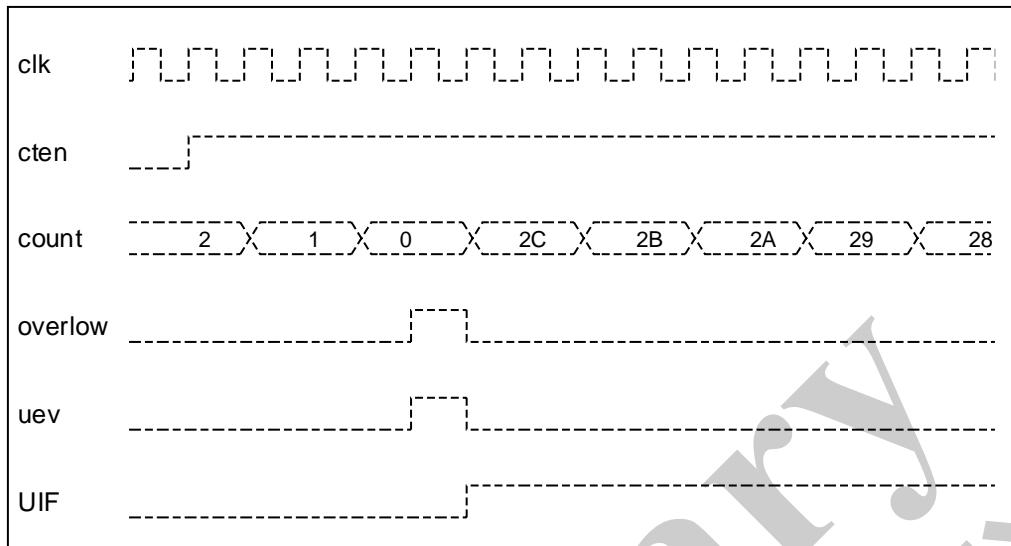


图 13-17 带预分频的下计数

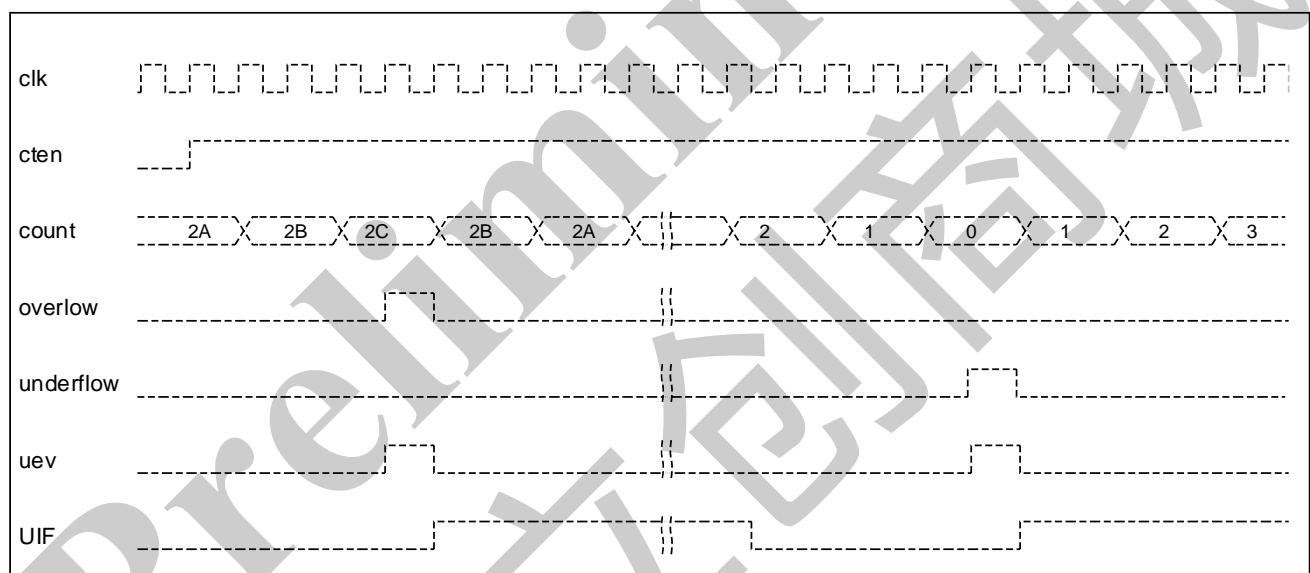


图 13-18 带预分频的上下计数

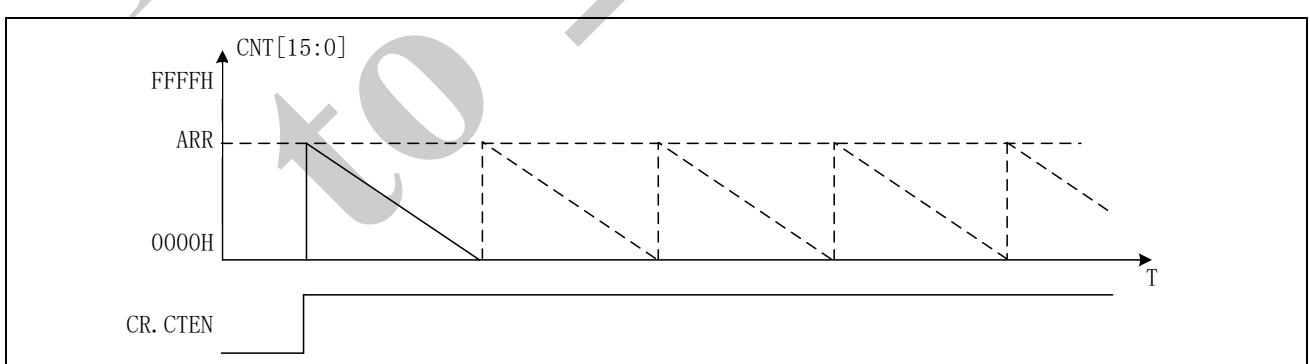


图 13-19 边沿对齐计时器波形( $DIR = 1$ )

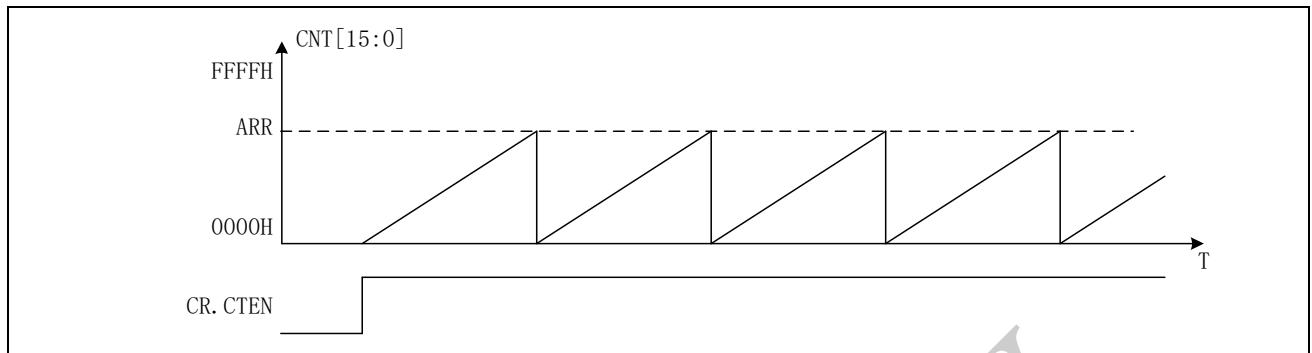


图 13-20 边沿对齐计时器波形( DIR =0)

模式 3 为三角波计数波形，计数方向控制位只读，不可以更改计数方向。

中心对齐（三角波）模式下 CR.DIR 方向位是只读的。写值无效。从其他模式切换到中心对齐模式 DIR 自动清 0。软件事件更新和从模式外部触发复位模式 DIR 自动清零。

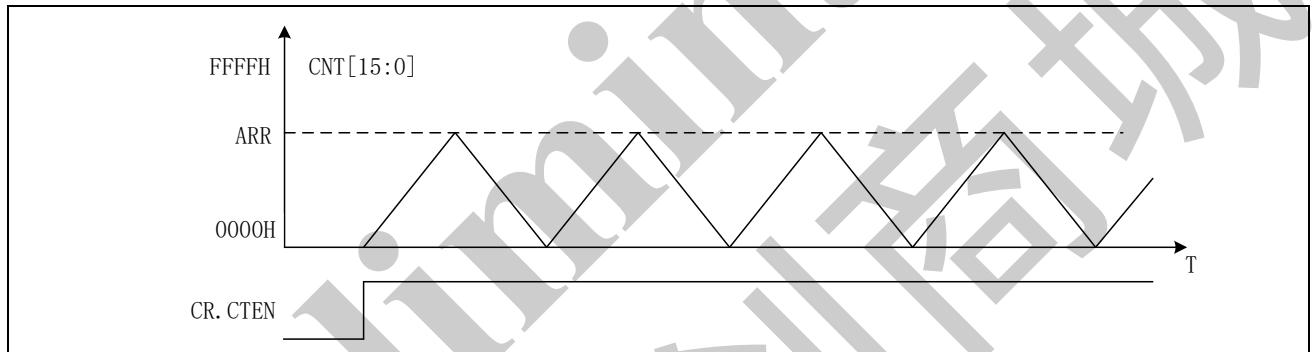


图 13-21 中心对齐计数器波形

### 13.3.6.3 重复计数

重复计数器使用计数器的溢出进行向下计数。计数到 0 时，即计数器发生重复寄存器设置的值加一次溢出时。当缓存寄存器使能时，周期重载寄存器更新到周期缓存寄存器。比较模式下比较寄存的值更新到比较缓存寄存中。

重复计数器在下面条件成立时递减

- 上计数模式下每次计数器溢出时
- 下计数模式下每次计数器下溢时
- 三角波模式下每次上溢出和每次下溢出时

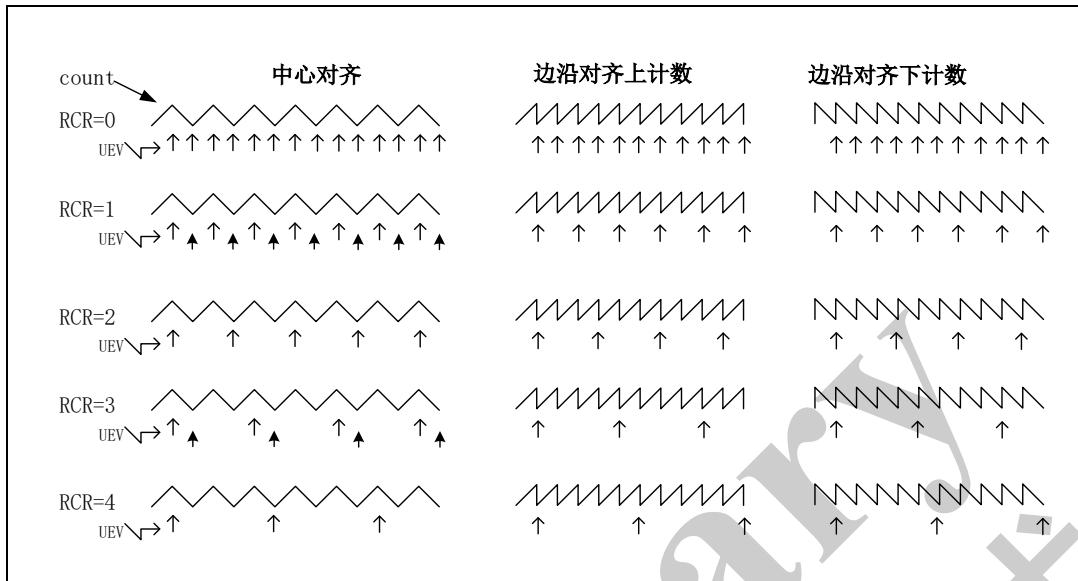


图 13-22 重复计数器产生更新时序 ( $RCR.UD=0$   $RCR.OV=0$ )

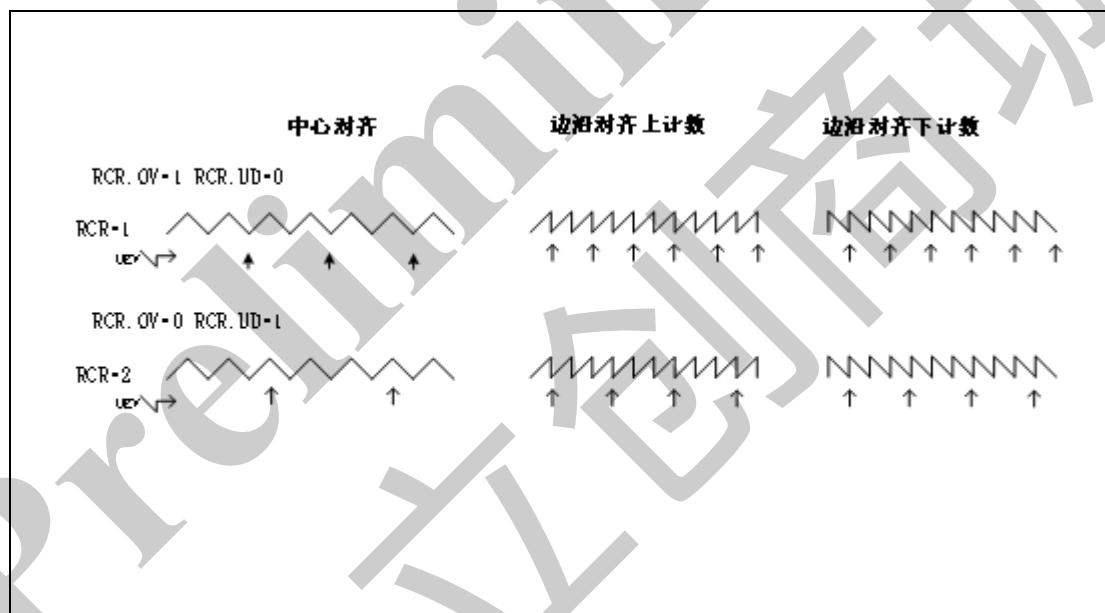


图 13-23 重复计数器产生更新时序 ( $RCR.UD \neq 0$   $RCR.OV \neq 0$ )

除了上下溢出通过重复计数器可以产生事件更新 UEV 外,还可以通过写寄存器 CR.UG 产生软件及从模式复位事件更新 UEV; 这时需要配置 CR.URS。

### 13.3.6.4 数据缓存

自动重载数据 ARR 与比较寄存器都可以配置缓存功能，当缓存功能有效时，当发生 UEV 事件更新时，写入的周期值 ARR 与比较值 CCR 才会生效。

自动重载值在不同计数模式下的更新时序图如下

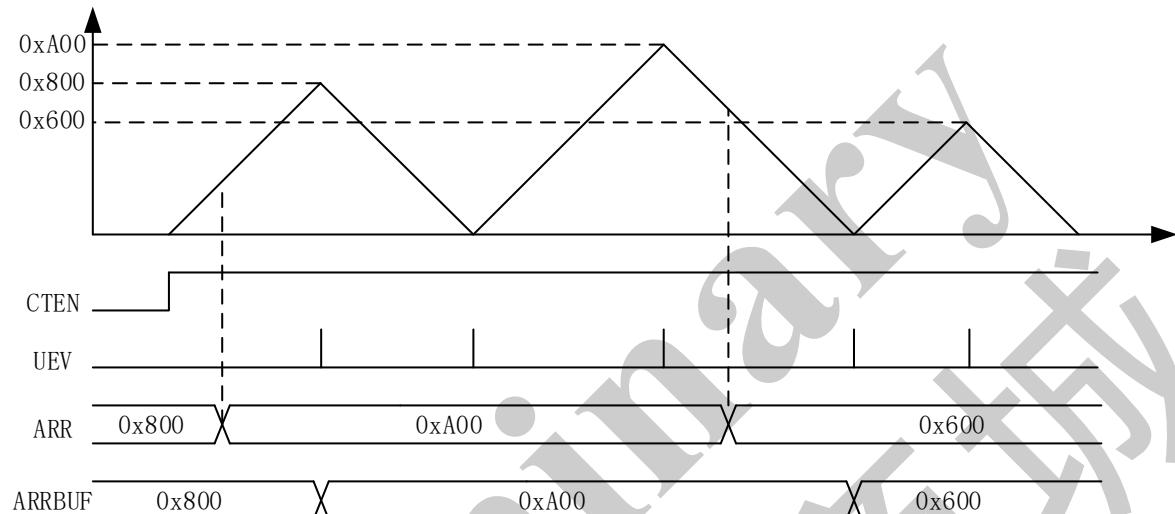


图 13-24 三角波模式下缓存使能

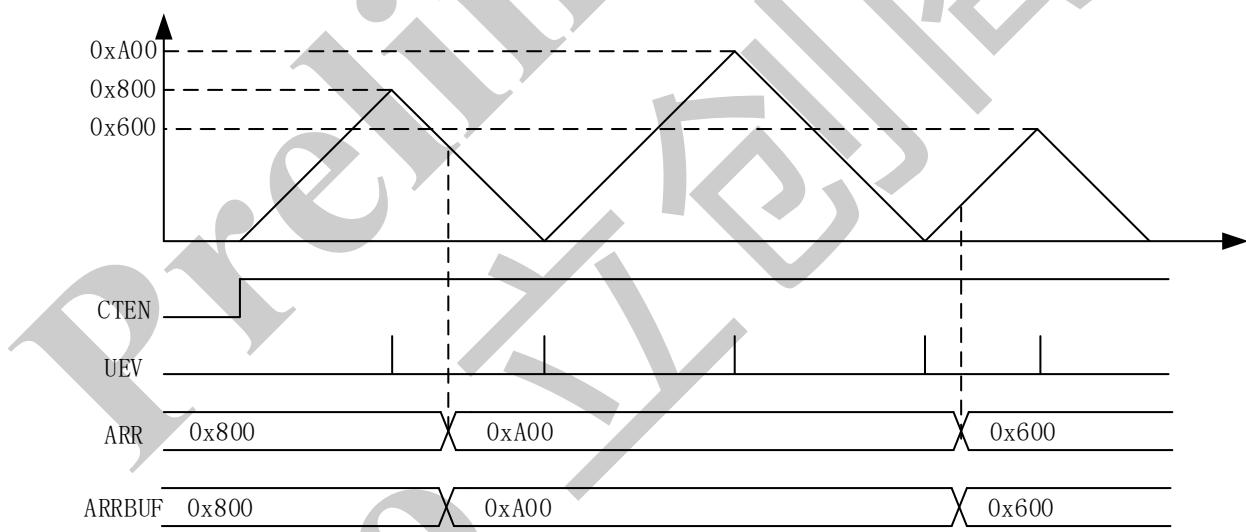


图 13-25 三角波模式下缓存无效

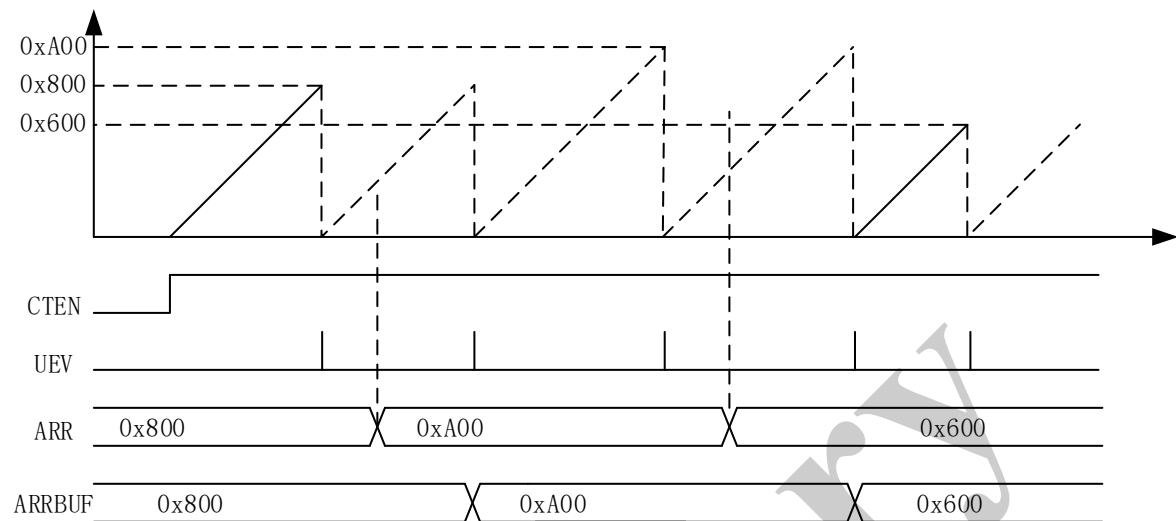


图 13-26 锯齿波模式下上计数缓存使能

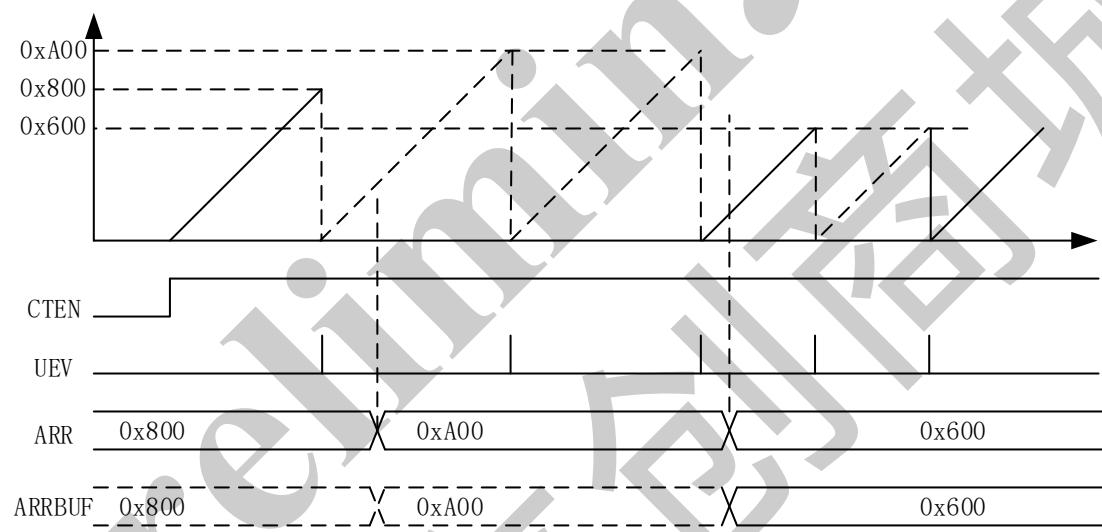


图 13-27 锯齿波模式下上计数缓存无效

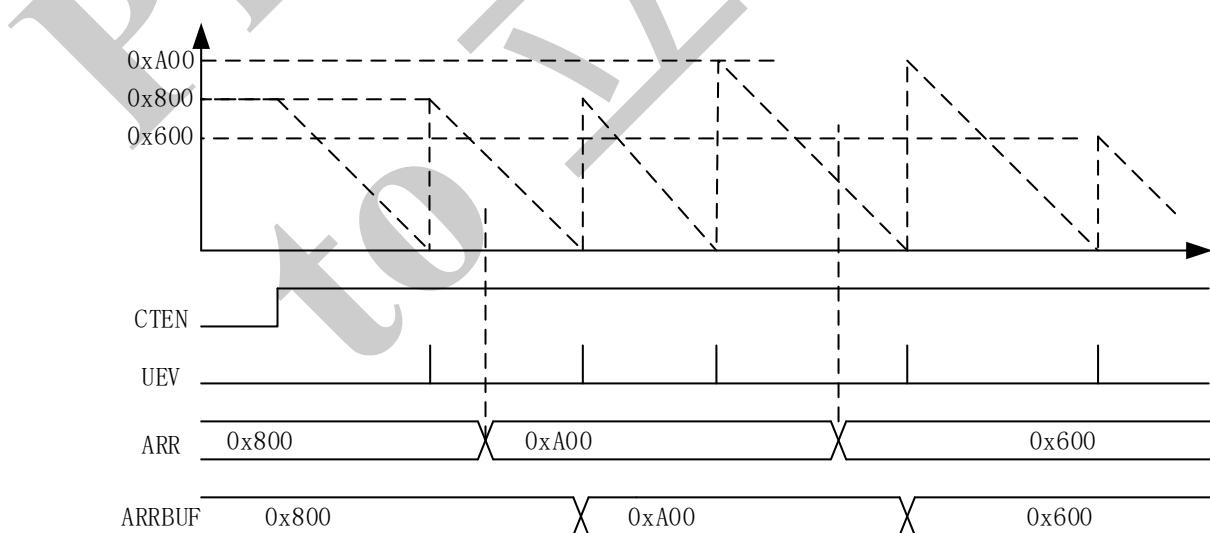


图 13-28 锯齿波模式下计数缓存使能

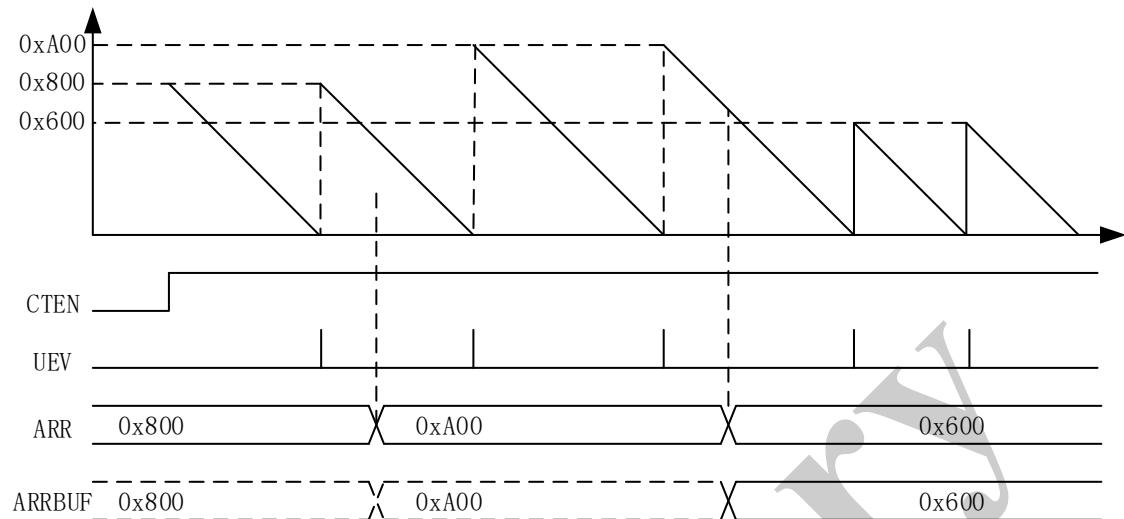


图 13-29 锯齿波模式下计数缓存无效

在三角波模式与锯齿波上计数模式时, 如果缓存不使能, 更改的 ARR 时, 当前计数器的值要小于要更改的 ARR 周期值, 否则当前周期会计数到 0xFFFF。

比较缓存与周期缓存更新状态一致, 这里不一一列出时序图。

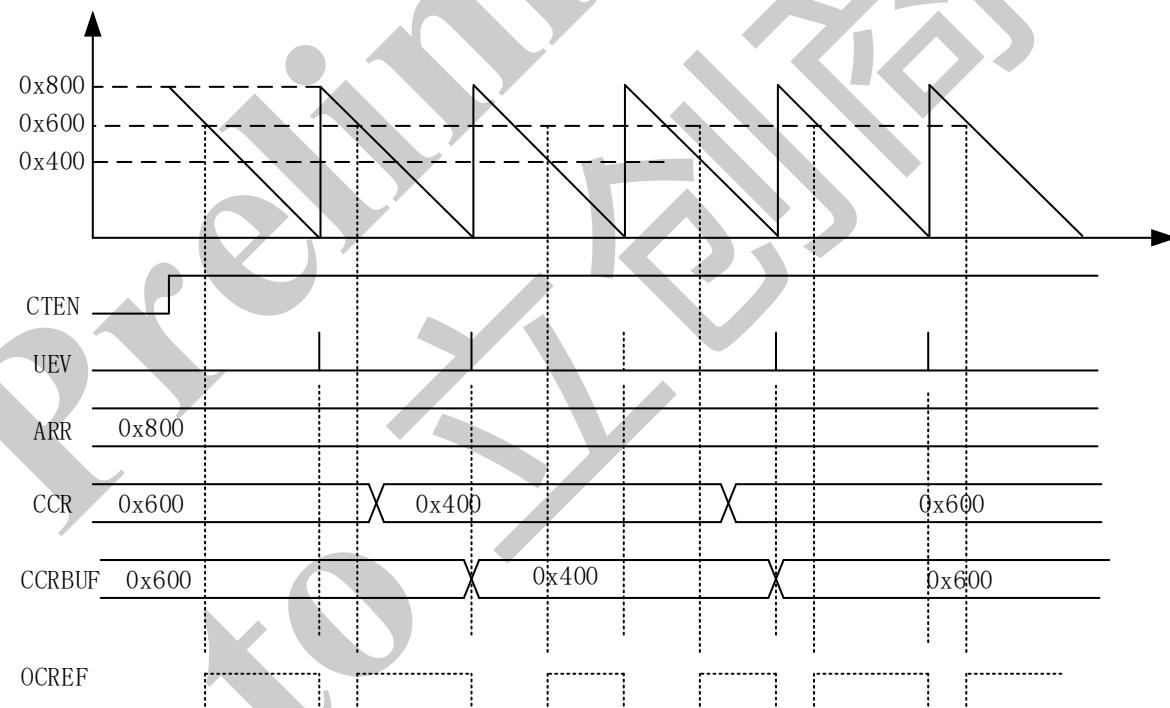


图 13-30 锯齿波模式下计数比较缓存使能

### 13.3.6.5 比较输出 OCREF

比较输出 OCREFA 可以配置为单点比较，使用比较寄存器 CCRA 控制 OCREFA 的输出；OCREFA 的比较输出也可以配置为双点比较，使用比较寄存器 CCRA,CCRB 一起控制 OCREFA 的比较输出。

OCREFB 的比较输出只能使用单点比较，使用比较寄存器 CCRB 控制 OCREFB 的比较输出。

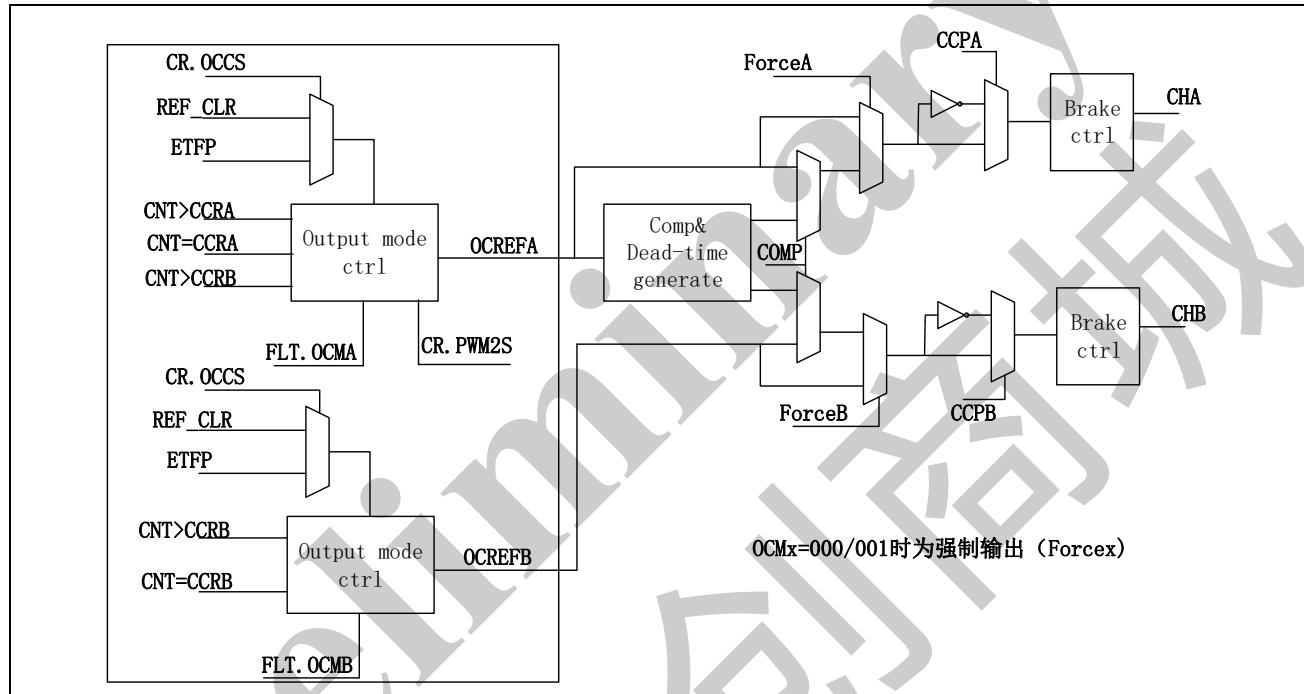


图 13-31 OCREF 输出框图

#### OCREF 输出使用 OCMx 选择

000: 强制为 0

001: 强制为 1

010: 比较匹配时强制为 0

011: 比较匹配时强制为 1

100: 比较匹配时翻转

101: 比较匹配时输出一个计数周期的高电平

110: PWM 模式 1

#### 单点比较：

上计数时 CNT < CCRxy 输出高，下计数时 CNT > CCRxy 输出为低电平

双点比较:

- 1) 锯齿波上计数  $CCRx_A < CNT \leq CCRx_B$  输出为低电平
- 2) 锯齿波下计数  $CCRx_A < CNT \leq CCRx_B$  输出为高电平
- 3) 三角波上计数  $CNT < CCRx_A$  输出高, 下计数  $CNT > CCRx_B$  为低电平

111: PWM 模式 2

单点比较:

上计数时  $CNT < CCRxy$  输出低, 下计数时  $CNT > CCRxy$  输出为高 电平

双点比较:

- 1) 锯齿波上计数  $CCRx_A \leq CNT < CCRx_B$  输出为高电平
- 2) 锯齿波下计数  $CCRx_A \leq CNT < CCRx_B$  输出为低电平
- 3) 三角波上计数  $CNT < CCRx_A$  输出低, 下计数  $CNT > CCRx_B$  为高电平

注: 强制输出有高优先级, 当强制输出有效时, 互补输出控制无效。

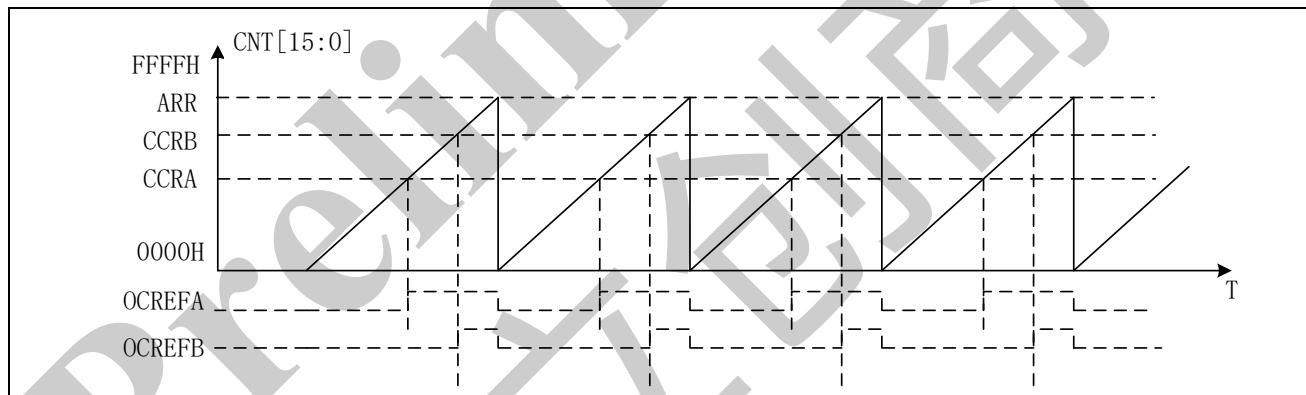


图 13-32 锯齿波计数单点比较OCREF 输出波形 (OCMx=111)

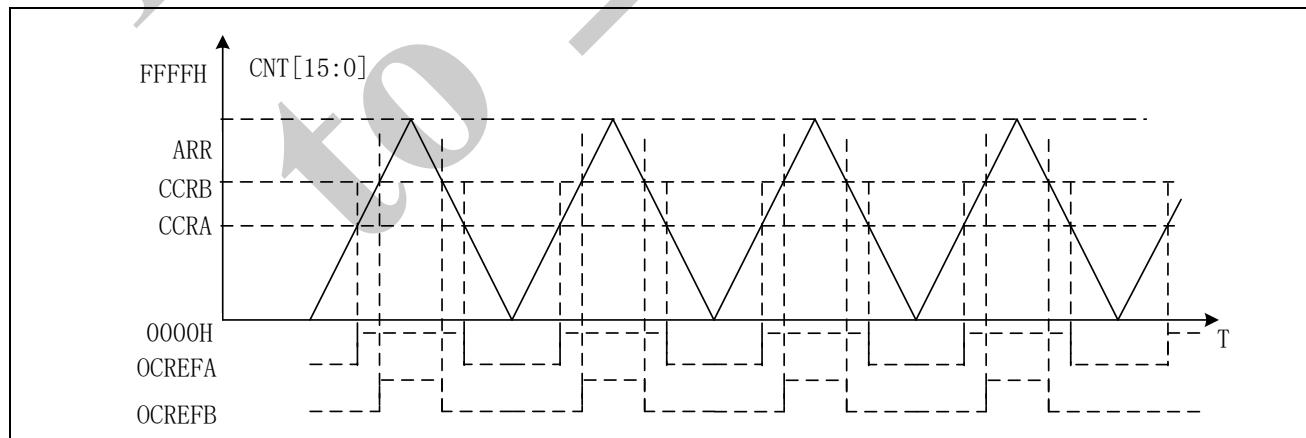


图 13-33 三角波计数单点比较 OCREF 输出波形 (OCMx=111)

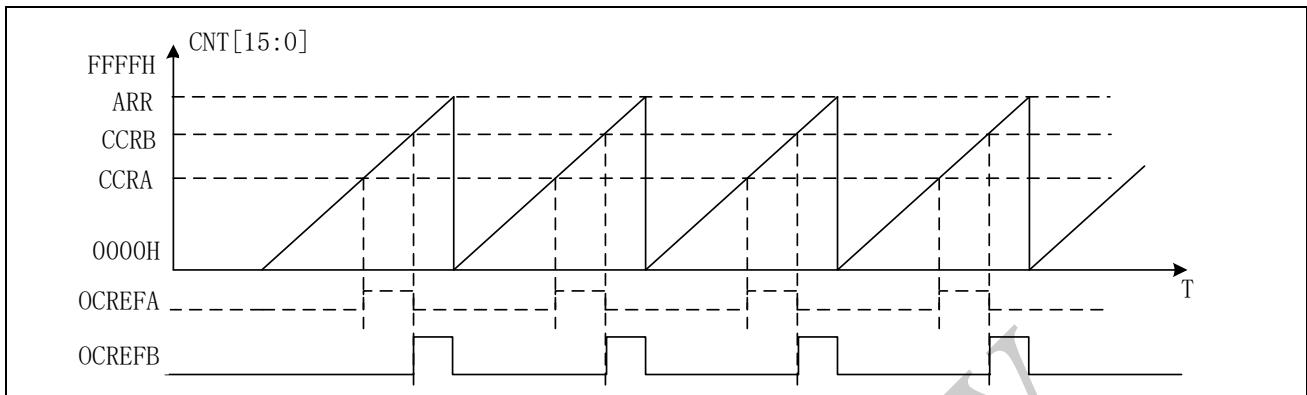


图 13-34 锯齿波计数双点比较 OCREF 输出 (OCMx=111)

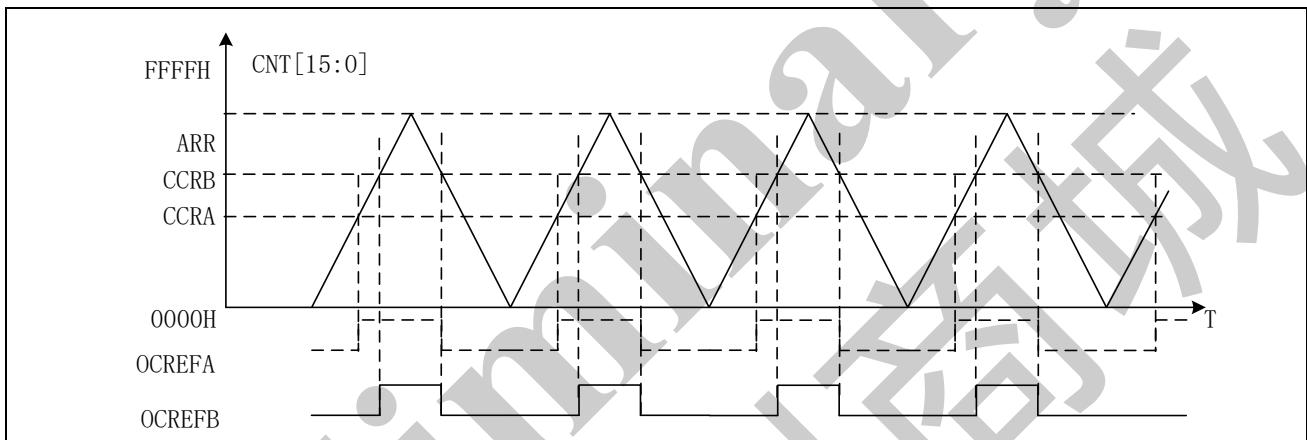


图 13-35 三角波计数双点比较 OCREF 输出 (OCMx=111)

Prepared exclusively for  
to my love

### 13.3.6.6 独立 PWM 输出

由 OCREFA 控制 CHA 的输出， OCREFB 控制 CHB 的输出。通过 FLTR.CCPA,FLTR.CCPB 可以控制 CHA, CHB 输出的反向。

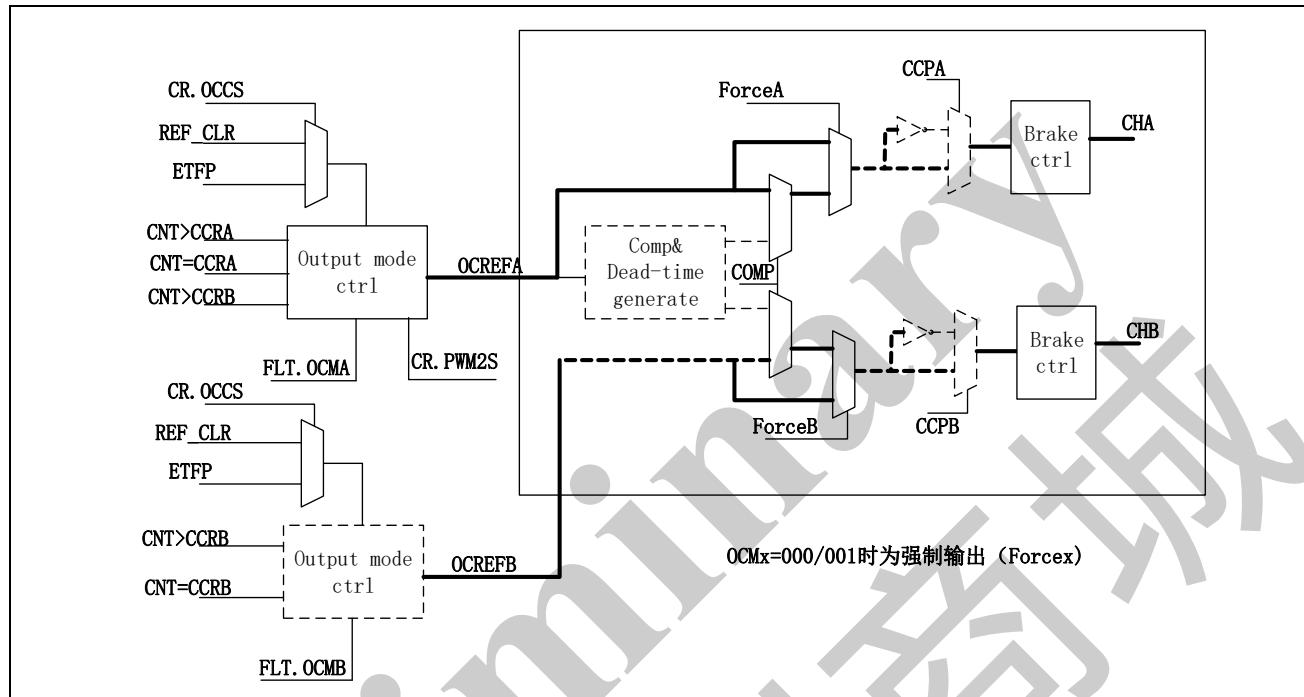


图 13-36 独立 PWM 输出框图

#### PWM 输出与 OCREF 关系

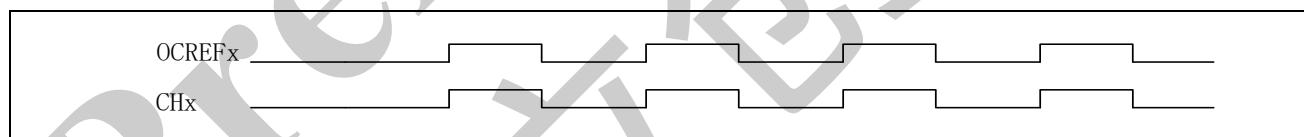


图 13-37 CCPx=0 时 PWM 输出波形

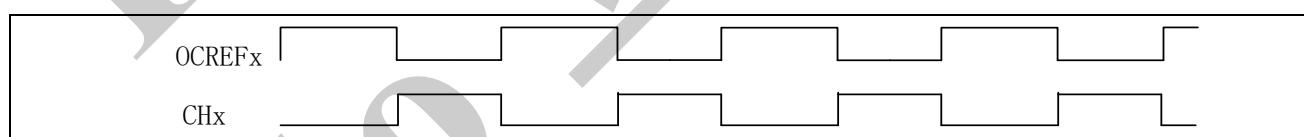


图 13-38 CCPx=1 时 PWM 输出波形

### 13.3.6.7 互补 PWM 输出

由 OCREF<sub>A</sub> 控制 CHA 的输出, OCREF<sub>A</sub> 同时控制 CHB 的输出。比较寄存器 CCR<sub>x</sub>B 可以作为专用比较控制 ADC 触发。

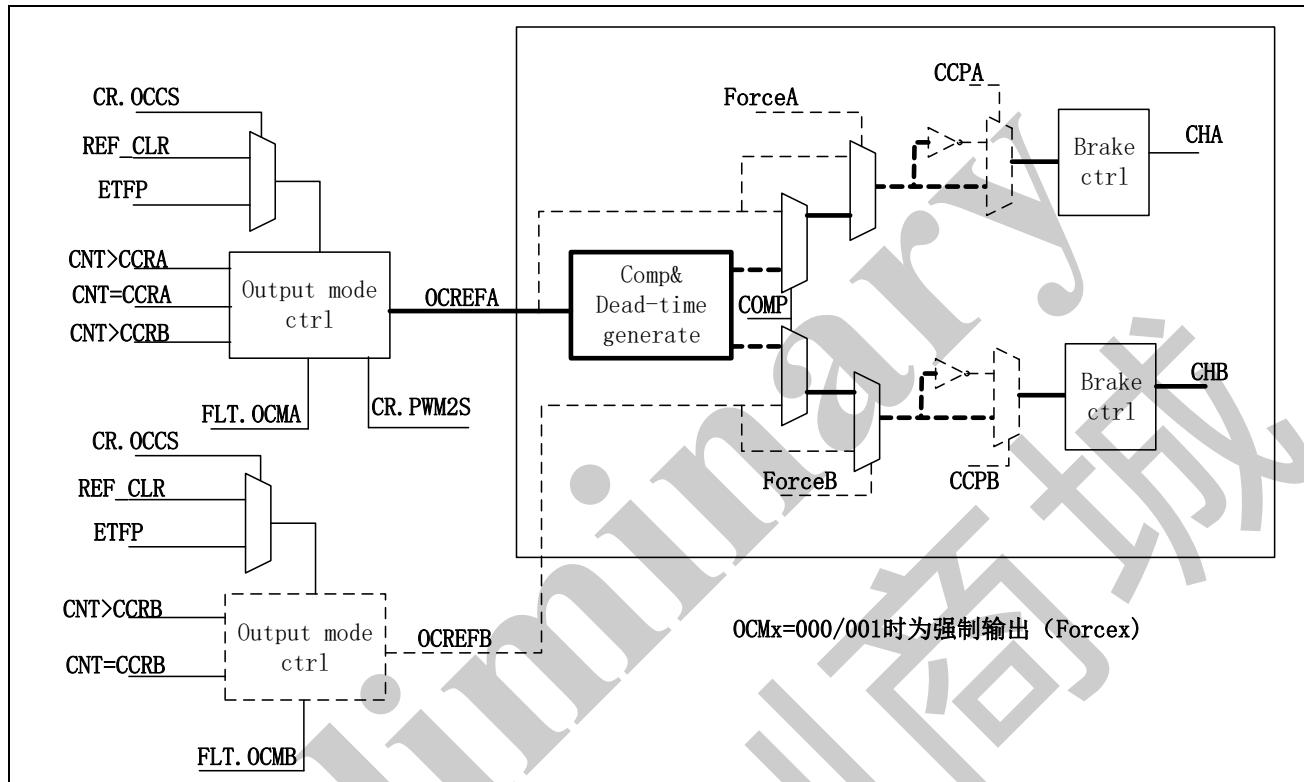


图 13-39 互补 PWM 输出框图

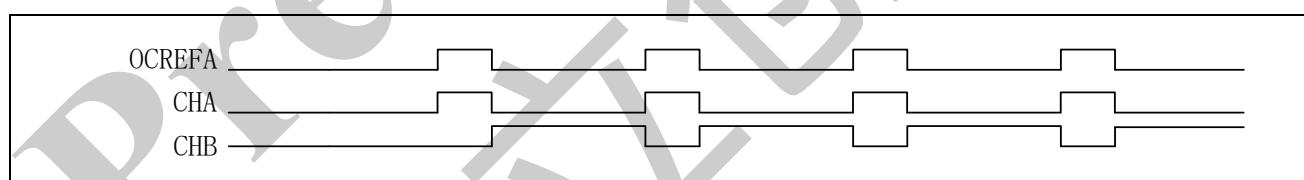


图 13-40 互补 PWM 输出波形图

### 13.3.6.8 有死区的 PWM 输出

在互补 PWM 输出模式下可以设置死区功能。

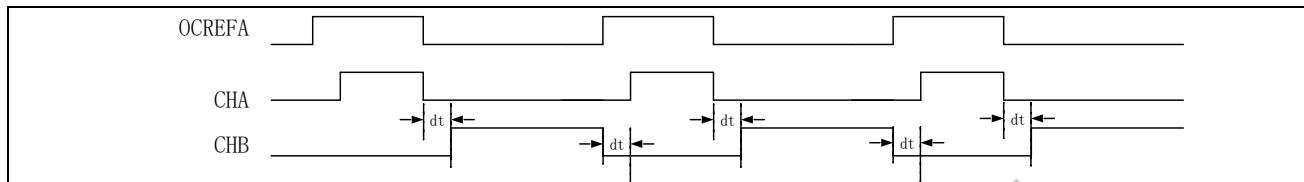
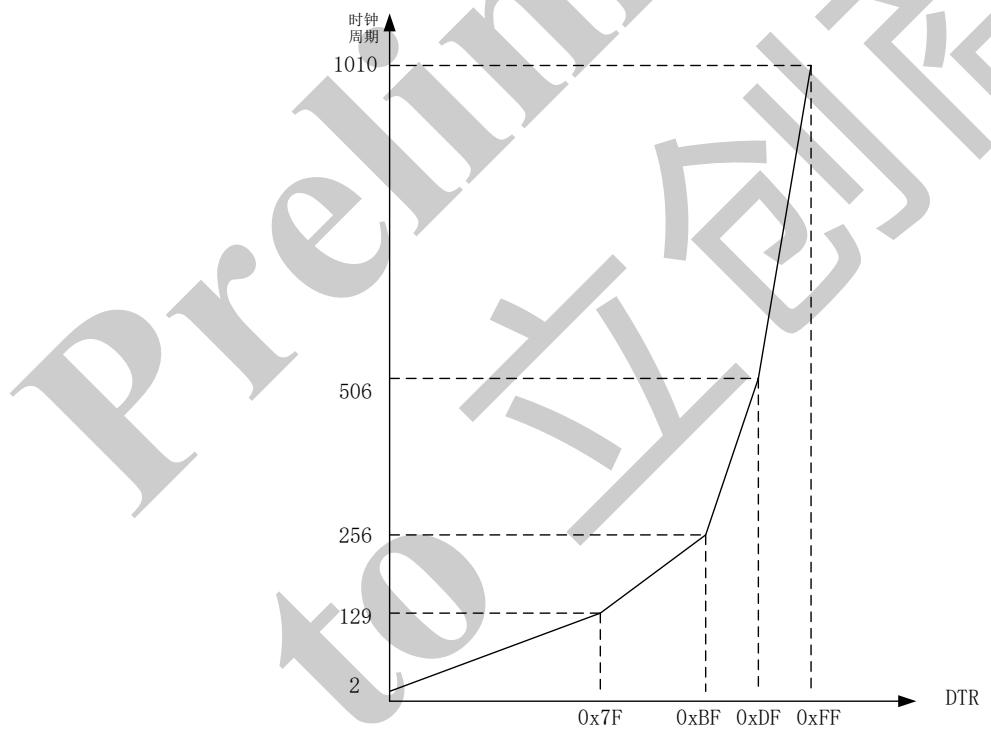


图 13-41 互补PWM输出波形图

死区时间使用 8 位 DTR 控制，死区时间  $dt$  与 DTR 的关系如下

$DTR[7] = 0$	$T = DTR[6:0] + 2$	2-129	step=1
$DTR[7:6] = 10$	$T = \{DTR[5:0] + 64\} * 2 + 2$	130-256	step=2
$DTR[7:5] = 110$	$T = \{DTR[4:0] + 32\} * 8 + 2$	258-506	step=8
$DTR[7:5] = 111$	$T = \{DTR[4:0] + 32\} * 16 + 2$	514-1010	step=16



100M clock 死区时间由 2.56us 调整为 10.1us

图 13-42 死区时间

### 13.3.6.9 单脉冲输出

单脉冲模式 (ONE SHOT) 是上述 PWM 模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将

ATIMx\_M23CR 寄存器中的 ONESHOT 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

单脉冲模式在锯齿波下计数模式计数器初值不要设置为 0, 上计数模式计数值不要设置到大于等于 ARR。

只有当比较值与计数器初始值不同时，才能正确产生输出脉冲。启动前（定时器等待触发时），必须进行如下配置：

- 递增计数时： $CNT < CCRxy \leq ARR$ （特别注意， $0 < CCRxy$ ），
- 递减计数时： $CNT > CCRxy$ 。

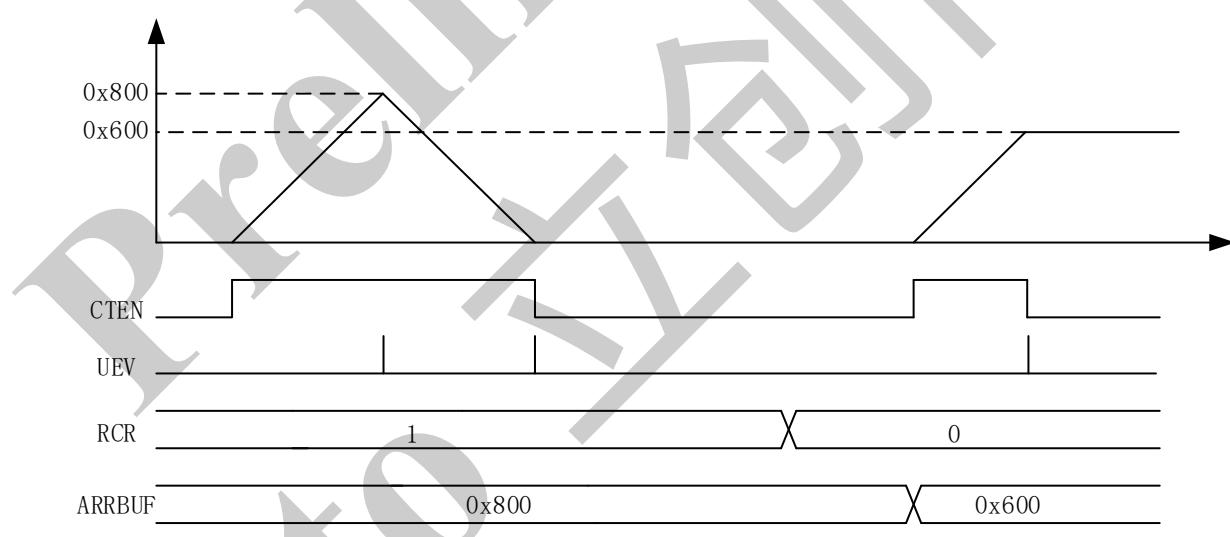


图 13-43 三角波模式单脉冲计数

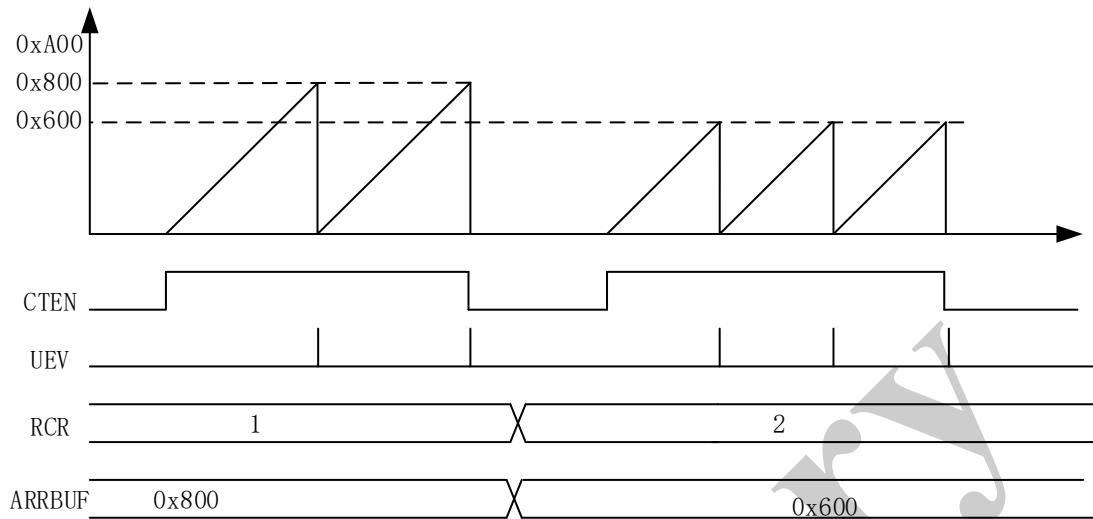


图 13-44 锯齿波上计数单脉冲模式

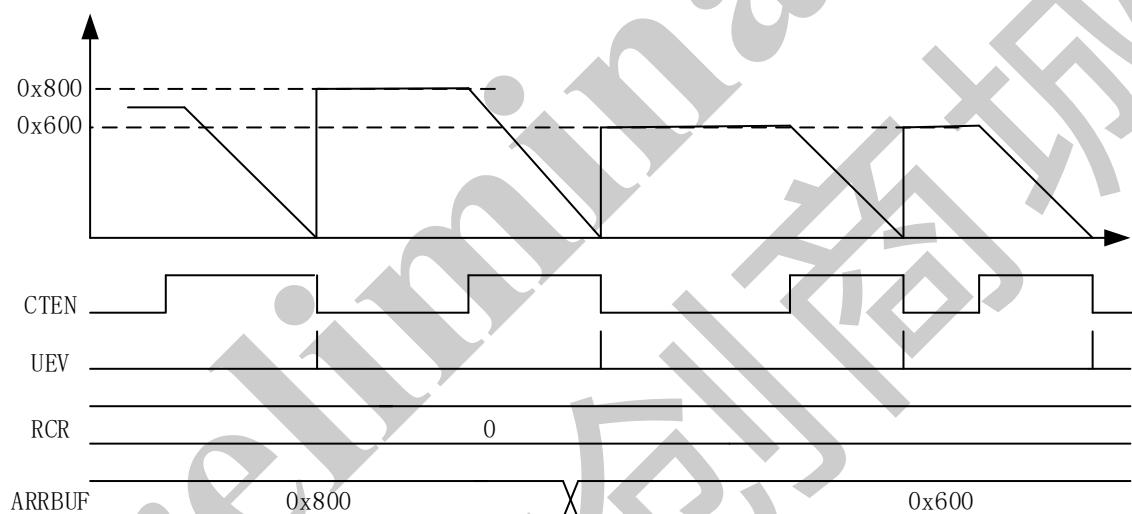


图 13-45 锯齿波下计数单脉冲模式

### 13.3.6.10 比较中断

锯齿波比较匹配会将 IFR 寄存器相应的标志置 1，如果中断使能 CRCHx.CIEy (x=0,1,2;y=A,B)将会触发中断。

三角波比较匹配可单独选择上升计数比较匹配，下降计数比较匹配或者两者比较都匹配。

比较 A 比较匹配当计数值与 CCRxA 相等时，统一使用 M23CR.CIS 控制。

M23CR.CIS=2'b00 时 比较匹配无输出

M23CR.CIS=2'b01 时 上计数时比较匹配

M23CR.CIS=2'b10 时 下计数时比较匹配

M23CR.CIS=2'b11 时 上下计数时比较都匹配

比较 B 比较匹配当计数值与 CCRxB 相等时，不同通道可以使用 CRCHx.CISB 单独控制。

CRCHx.CISB=2'b00 时 通道 x 比较匹配无输出

CRCHx.CISB=2'b01 时 通道 x 上计数时比较匹配

CRCHx.CISB=2'b10 时 通道 x 下计数时比较匹配

CRCHx.CISB=2'b11 时 通道 x 上下计数时比较都匹配

B 通道比较匹配单独控制为了更灵活的触发 ADC。详见 Timer 触发 ADC 章节。

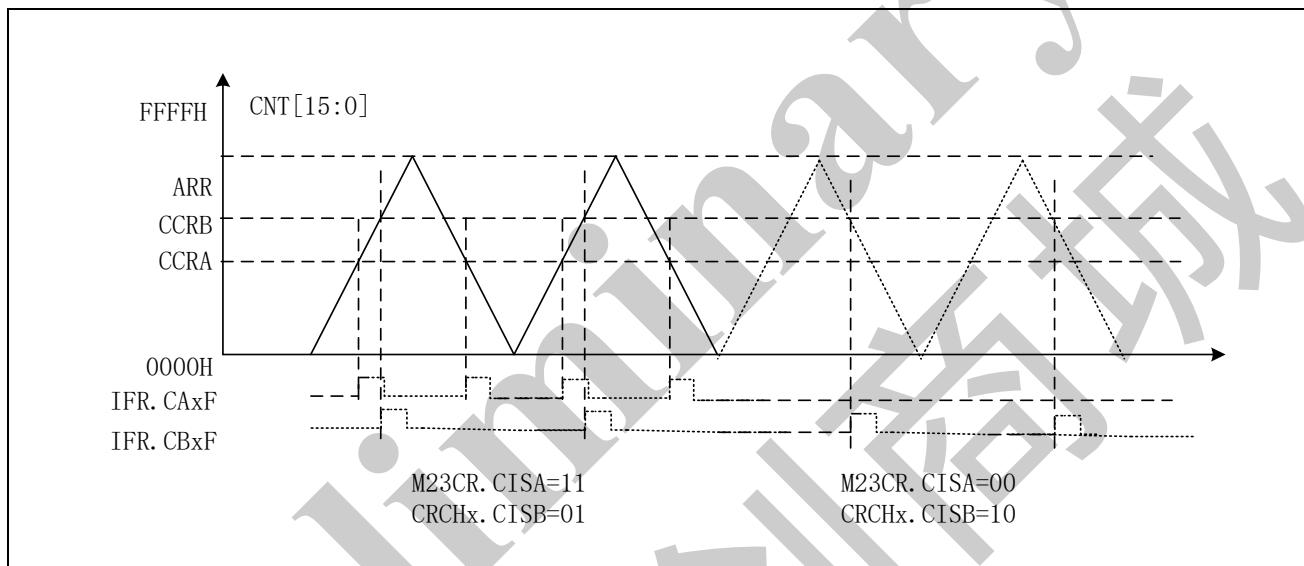


图 13-46 中断示意图

### 13.3.6.11 捕获输入

M23CR.MODE=2/3

在三角波计数或者锯齿波计数模式下都可以设置捕获功能，可以设置捕获的电平边沿，当发生捕获时捕获的值存入比较捕获寄存器并产出捕获中断。

每个通道的比较捕获功能可以单独设置，通过寄存器 CRCHx.CSA/CSB 选择。

每个通道的捕获边沿可以单独设置，通过寄存器 CRCHx.CFy/CRy(x=0/1/2;y=A/B)选择捕获触发的边沿。

当捕获发生后捕获标志未清除前再次发生捕获动作，会产生捕获数据覆盖标志。

定时器未启动时如果有效的捕获边沿也会产生捕获标志及捕获动作。

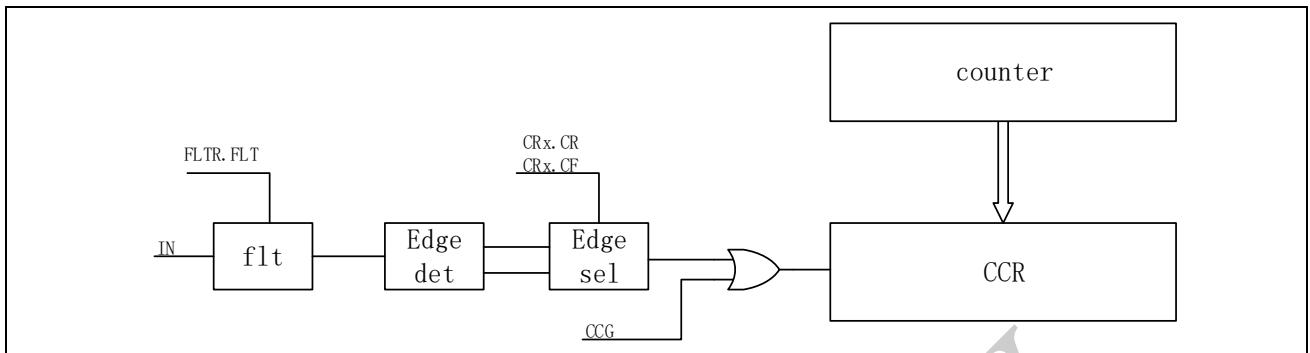


图 13-47 捕获功能框图

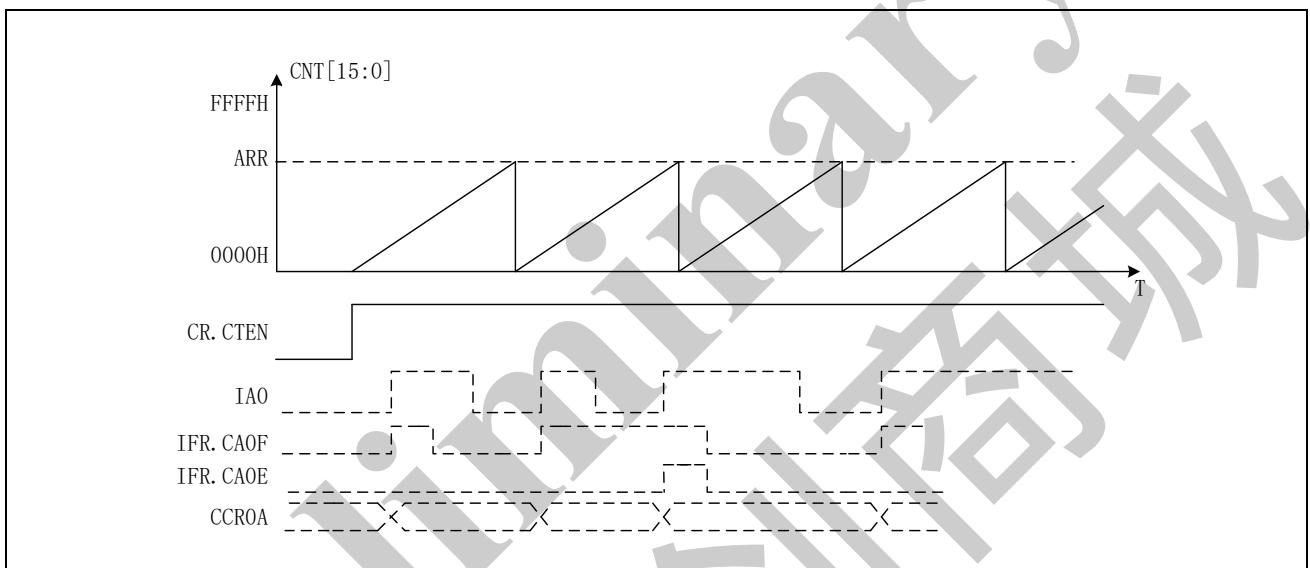


图 13-48 捕获时序图

CH0A 的捕获输入通过 MSCR.IA0S 选择 CH0A 输入还是 CH0A, CH1A, CH2A 的异或输入。

IA0S	0	1
ATIM3	CHA0	CHA0 CHA1 CHA2 异或输入

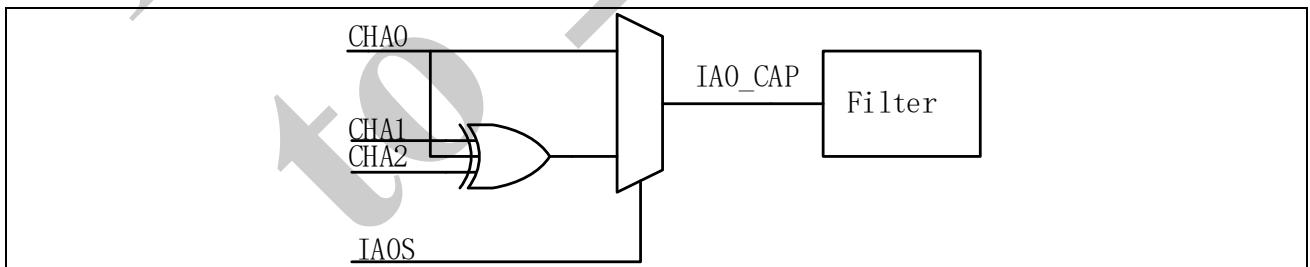


图 13-49 CHA 端口选择

CH0B 的捕获输入通过 MSCR. IB0S 选择 CH0B 输入还是内部触发 MSCR.TS 选择的信号。

IB0S	0	1																		
ATIM3	CHB0	<p>内部触发 MSCR.TS 选择信号</p> <table border="1"> <thead> <tr> <th>MCSCR.TS</th> <th>通道B捕获输入</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>ETR 滤波相位选择输出信号, 滤波相位选择可配</td> </tr> <tr> <td>001</td> <td>内部互联ITR0</td> </tr> <tr> <td>010</td> <td>内部互联ITR1</td> </tr> <tr> <td>011</td> <td>内部互联ITR2</td> </tr> <tr> <td>100</td> <td>内部互联ITR3</td> </tr> <tr> <td>101</td> <td>CH0A 边沿</td> </tr> <tr> <td>110</td> <td>CH0A 滤波输出信号, 滤波功能可配</td> </tr> <tr> <td>111</td> <td>CH0B 滤波输出信号, 滤波功能可配</td> </tr> </tbody> </table>	MCSCR.TS	通道B捕获输入	000	ETR 滤波相位选择输出信号, 滤波相位选择可配	001	内部互联ITR0	010	内部互联ITR1	011	内部互联ITR2	100	内部互联ITR3	101	CH0A 边沿	110	CH0A 滤波输出信号, 滤波功能可配	111	CH0B 滤波输出信号, 滤波功能可配
MCSCR.TS	通道B捕获输入																			
000	ETR 滤波相位选择输出信号, 滤波相位选择可配																			
001	内部互联ITR0																			
010	内部互联ITR1																			
011	内部互联ITR2																			
100	内部互联ITR3																			
101	CH0A 边沿																			
110	CH0A 滤波输出信号, 滤波功能可配																			
111	CH0B 滤波输出信号, 滤波功能可配																			

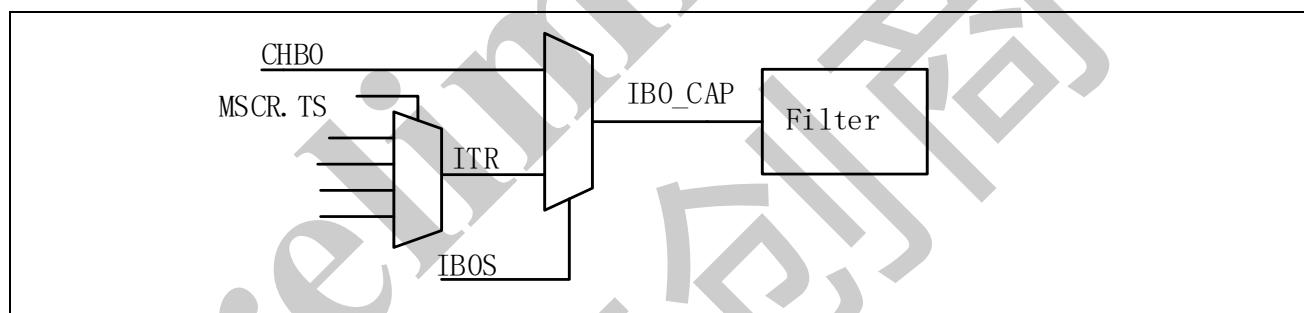


图 13-50 CHB 端口选择

### 13.3.6.12 设置示例

#### 边沿对齐独立 PWM 输出设置

1. 设置模式 M23CR.MODE=2
2. 设置锯齿波计数方向 M23CR.DIR
3. 设置 PWM 周期值 ARR
4. 设置计数器初值（初值必须小于周期值）
5. 设置 PWM 比较值 CCRxA,CCRx B
6. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
7. 清除相关中断标志
8. 使能相应的中断
9. 设置输出极性 FLT.CCPAx FLT.CCPBx
10. 使能输出 DTR.MOE
11. 使能定时器 M23CR.CTEN

#### 中心对齐互补 PWM 输出设置

1. 设置模式 M23CR.MODE=3
2. 设置互补输出 M23CR.COMP
3. 设置 PWM 周期值 ARR
4. 设置计数器初值（初值必须小于周期值）
5. 设置 PWM 比较值 CCRxA(CCRxB 不需要设置， PWM 输出与这个寄存器无关)
6. 设置 PWM 输出比较模式 FLT.OCMA FLT.OCMB 为 6 或 7
7. 清除相关中断标志
8. 使能相应的中断
9. 设置输出极性 FLT.CCPAx FLT.CCPBx
10. 使能输出 DTR.MOE
11. 使能定时器 M23CR.CTEN

### 三角波非中心对齐带死区互补 PWM 输出设置

1. 设置模式 M23CR.MODE=3
2. 设置互补输出 M23CR.COMP
3. 设置两个比较使能 M23CR.PWM2S
4. 设置 PWM 周期值 ARR
5. 设置计数器初值（初值必须小于周期值）
6. 设置 PWM 比较值 CCRxA,CCRx B,上计数比较点为 CCRxA,下计数比较点为 CCRxB
7. 设置 PWM 输出比较模式 FLT.OCMA,FLT.OCMB 为 6 或 7
8. 清除相关中断标志
9. 使能相应的中断
10. 设置输出极性 FLT.CCPAx,FLT.CCPBx
11. 设置死区使能 DTR.DTEN
12. 设置死区时间 DTR.DT
13. 使能输出 DTR.MOE
14. 使能定时器 M23CR.CTEN

### 捕获功能设置

设置 CH0B 为上升沿捕获功能

1. 选择计数器计数方式，设置 mode=2
2. 选择 CH0B 为捕获模式 CRCHx.CSB=1
3. 根据需要选择输入滤波 FLT.FLTB0
4. 选择捕获的边沿 CRCHx.CRB=1
5. 设置 ARR 改变周期值
6. 启动定时器 M23CR.CTEN
7. 清除相关中断标志
8. 清除捕获标志，使能相应中断
9. 查询到捕获标志后，读取 CCR0B 获得捕获值

注：定时器未启动时如果有效的捕获边沿也会产生捕获标志及捕获动作。

Preliminary  
to 立创商城

### 13.3.7 模式 2/3 从模式

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。

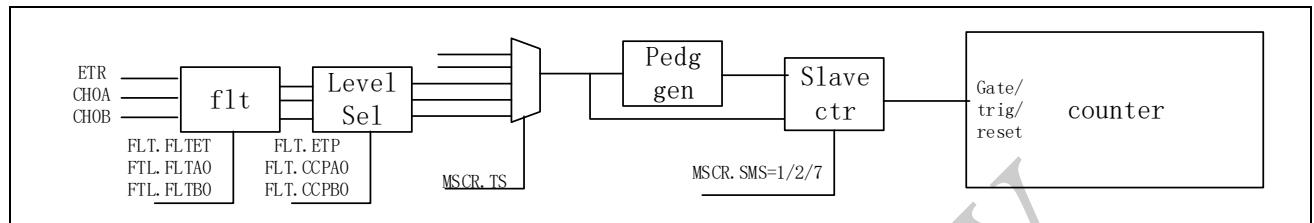


图 13-51 从模式示意图

从模式功能选择通过 MSCR.SMS 选择；

MSCR.SMS	从模式功能选择 001: 复位功能; 010: 触发模式; 111: 门控功能
MSCR.TS	触发选择 000: 端口ETR的滤波相位选择后的信号ETFP, 滤波功能可配置 001: 内部互联信号 ITR0 010: 内部互联信号 ITR1; 011: 内部互联信号 ITR2; 100: 内部互联信号 ITR3; 101: 端口CH0A的边沿信号; 110: 端口CH0A的滤波相位选择后的信号IAFP, 滤波功能可配置 111: 端口CHOB的滤波相位选择后的信号IBFP, 滤波功能可配置

#### 13.3.7.1 门控计数

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 CH0A 为低电平时向上计数：

- 配置通道 CH0A 以检测 CH0A 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 FLT.FLTA0=000)。置 FLTR 寄存器中 CCPA0=1 以确定极性(只检测低电平)。
- 置 MSCR 寄存器中 SMS=111，配置定时器为门控模式；置 MSCR 寄存器中 TS=110，选择 CH0A 作为输入源。
- 置 CR 寄存器中 CTEN=1，启动计数器。在门控模式下，如果 CTEN=0，则计数器不能启动，不论触发输入电平如何。只要 CH0A 为低，计数器开始依据内部时钟计数，一旦 CH0A 变高则停止计数。

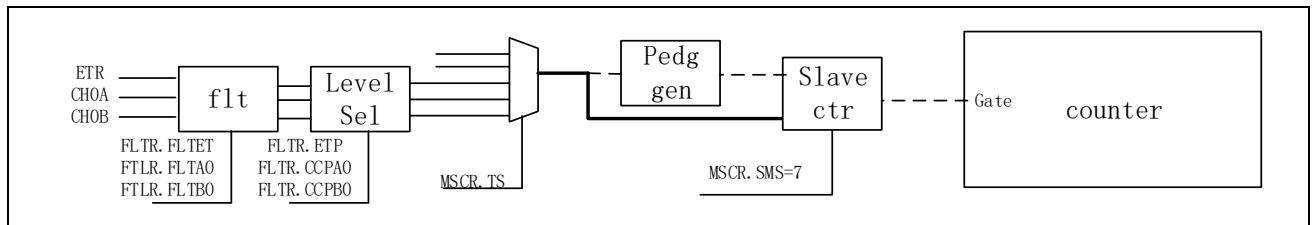


图 13-52 门控功能

### 13.3.7.2 触发功能

使用外部触发 (CH0A、CH0B、ETR) 可以使定时器同步启动，使用定时器内部互联信号结合 MSCR.MMS 也可以配置定时器同步启动。触发信号为输入信号的上升沿。也可以使用软件写 CR.TG 启动软件触发功能。

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 CH0B 输入的上升沿开始向上计数：

- 配置通道 CH0B 检测 CH0B 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 FLTR.FLTB0=000)。设 FLTR 寄存器中 CCPB0=0 以确定极性(不反向)。
- 置 MSCR 寄存器中 SMS=010，配置定时器为触发模式；置 MSCR 寄存器中 TS=111，选择 CH0B 作为输入源。当 CH0B 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。CH0B 上升沿和计数器启动计数之间的延时，取决于 CH0B 输入端的同步电路。

注意：如果使用下降沿触发，先选择触发极性，然后再选择模式，否则会产生误触发。

### 13.3.7.3 复位计数

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 CR 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(ARR，CCRx)都被更新了。

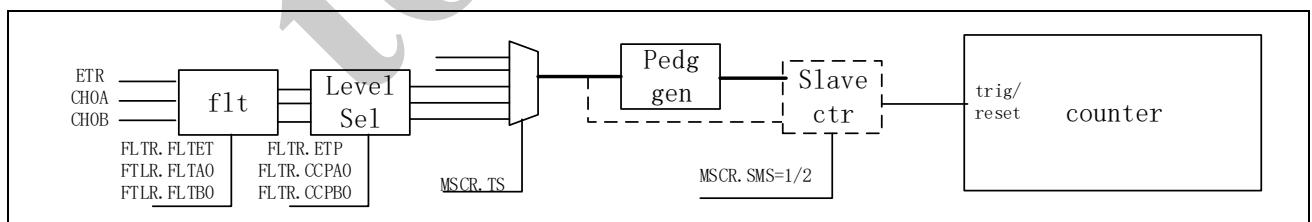


图 13-53 触发和复位功能

### 13.3.8 正交编码计数功能

MSCR.SMS=4/5/6, 对应正交编码模式的模式 1/2/3。这时计数器根据 IAFP, IBFP 的相位关系进行编码计数。IAFP,IBFP 为端口输入 CH0A, CH0B 的滤波相位选择的信号。模式 1 使用 CH0A 的边沿计数。模式 2 使用 CH0B 的边沿计数。模式 3 使用 CH0A,CH0B 的边沿都计数。

为了保证计数相位的正确性, 需要保证 A/B 输入的相位大于一个脉冲宽度的相位差, A/B 输入脉冲宽度需要大于两个脉冲宽度。

			IAFP		IBFP	
	IBFP	IAFP	Rising	falling	Rising	falling
MOD1	High		Down	Up	-	-
	Low		up	Down	-	-
MOD2		High	-	-	up	Down
		Low	-	-	Down	Up
MOD3	High	High	Down	Up	up	Down
	Low	Low	up	Down	Down	Up

CHAF/CHBF 为端口 CH0A/CH0B 滤波的信号, IAFP/IBFP 为端口滤波相位选择后的信号。

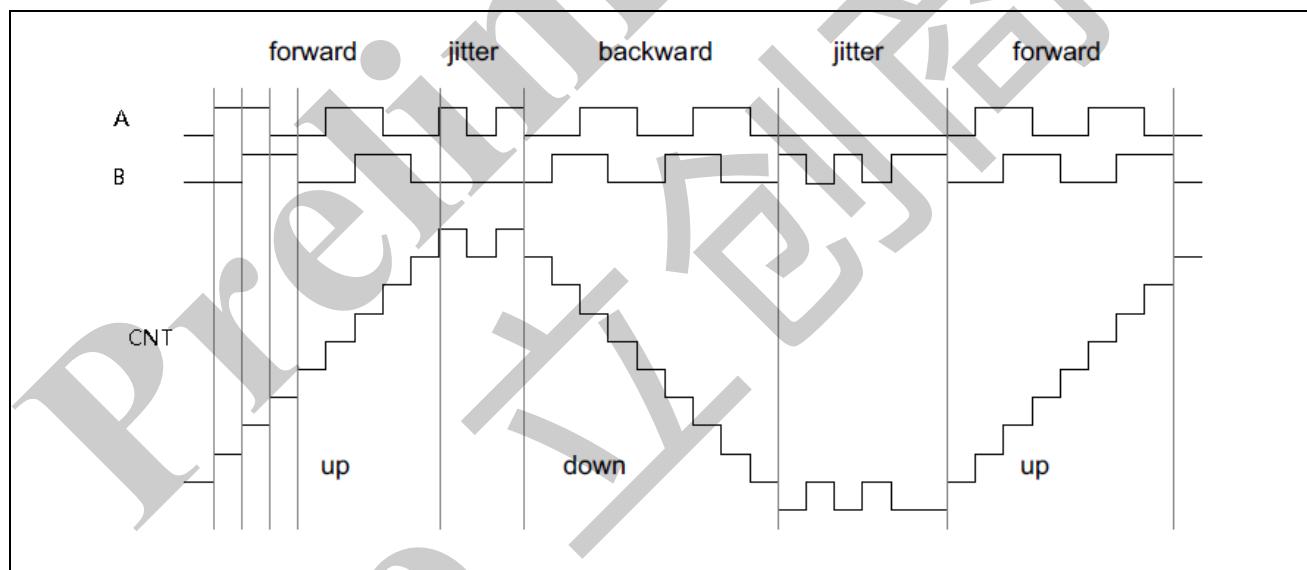
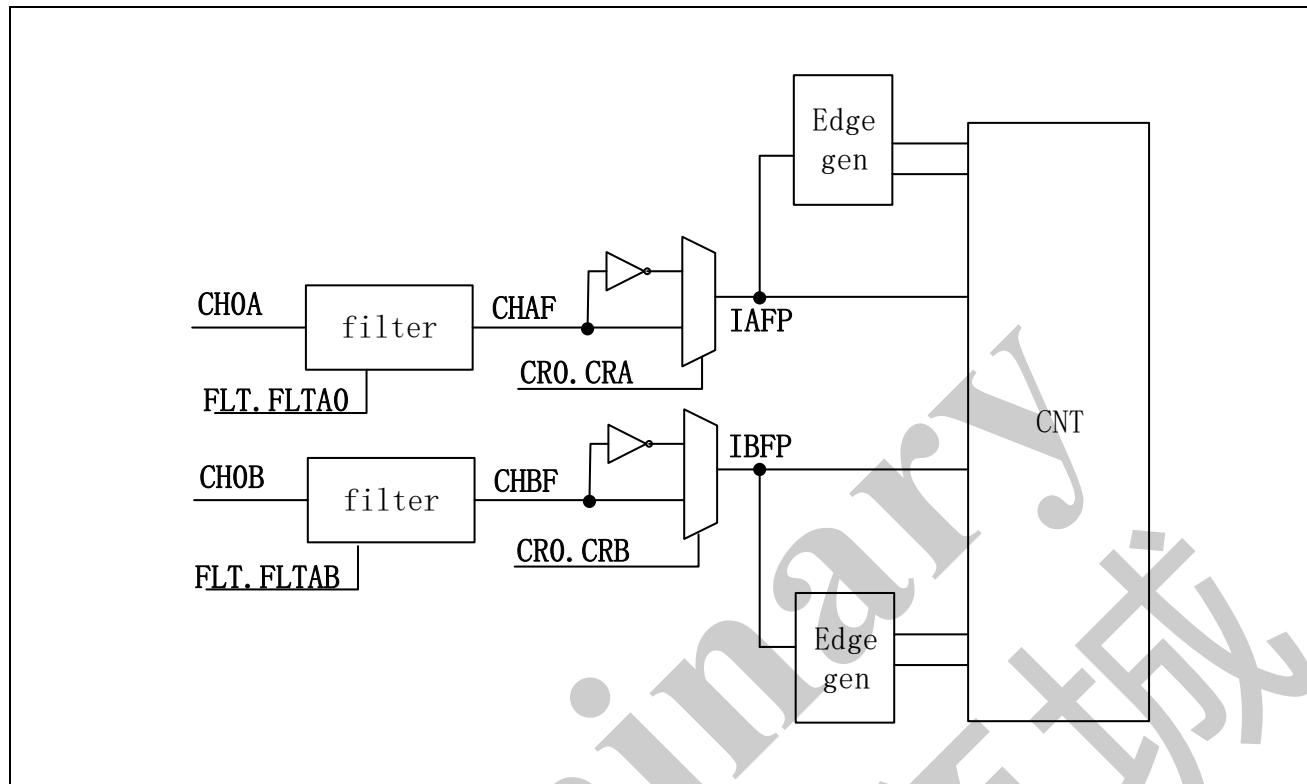


图 13-54 编码计数

### 13.3.9 Timer 触发 ADC

CCR0A,CCR1A,CCR2A 比较匹配可以配置触发 ADC，中心对齐 PWM 时只能通过控制寄存器 M23CR.CIS 可以选择在上升匹配触发或者下降匹配触发。

CCR0B,CCR1B,CCR2B,CCR3 比较匹配可以配置触发 ADC，中心对齐 PWM 时，可以通过寄存器 CRCHx.CISB 分别控制三个匹配触发点（上升，下降，上升下降）。

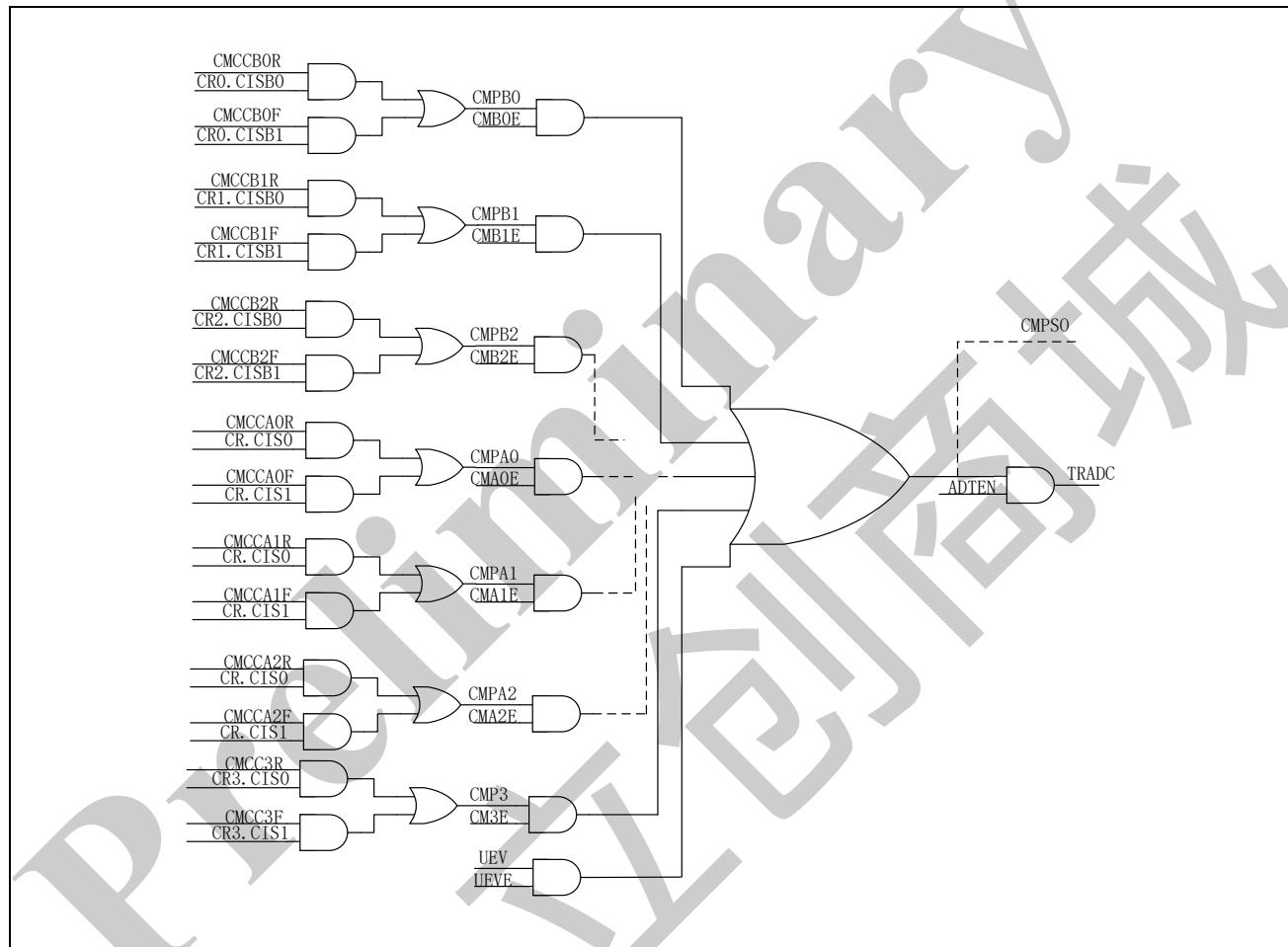


图 13-55 ADC 触发

#### 互补 PWM 输出时 B 比较作为 ADC 触发功能

1. 设置为 PWM 互补输出模式，参考“中心对齐互补 PWM 输出设置”
2. 设置 CCRxB 设置 ADC 触发比较点
3. 设置 CRCHx.CISB 选择上计数下计数比较匹配（锯齿波不需要选择）
4. 选择 ADC 触发比较的源 ADTR
5. 使能 ADC 触发 ADTR.ADTE

### 13.3.10 刹车控制

外部 BK 端口输入可控制刹车，系统 fail 可以控制刹车功能。通过 CR.BG 可以实现软件刹车功能，控制输出端口到设定的状态。

### 13.3.11 定时器互联

TRGO 输出信号可以连接到其他定时器的 ITR 信号。连接关系如下：

	ITR0	ITR1	ITR2	ITR3
ATIM3	NC	NC	GTIM_OV	NC

### 13.3.12 CH0B 捕获输入互联

ATIM3 的 CH0B 输入可以从端口直接输入，也可通过端口功能寄存器 GPIO\_CR4 选择可以连通到其他模块或端口。

当 ATIM3\_CHy=0x0 时，捕获输入是 PX\_SEL 选择的端口输入，当 ATIM3\_CHy=0x1~0x3 时，连接其他模块的输入或输出。

ATIM3	CH0B
000	PX_SEL
001	LPUART0_RXD
010	LPUART1_RXD
011	LVD_OUT

## 13.4 寄存器描述

Timer	基地址	描述
ATIM3	0x40005800	ATIM3基地址
ATIM0	0x40000C00	ATIM0基地址

(x=0,3)

寄存器	偏移地址	描述
ATIMx_ARR	0x000	Timer重载寄存器/周期
ATIMx_CNT	0x004	Timer 16位模式计数寄存器
ATIMx_M1CR	0x00C	Timer模式1控制寄存器 (按不同模式进行描述)
ATIMx_M23CR	0x00C	Timer模式23控制寄存器 (按不同模式进行描述)
ATIMx_IFR	0x010	Timer中断标志
ATIMx_ICLR	0x014	Timer中断清除寄存器
ATIMx_MSCR	0x018	主从模式控制
ATIMx_FLTR	0x01C	滤波控制
ATIMx_ADTR	0x020	ADC触发控制
ATIMx_CRCH0	0x024	比较单元0控制寄存器
ATIM3_CRCH1	0x028	比较单元1控制寄存器
ATIM3_CRCH2	0x02C	比较单元2控制寄存器
ATIMx_DTR	0x030	死区寄存器
ATIMx_RCR	0x034	重复计数寄存器
ATIMx_ARRDM	0x038	Timer重载寄存器/周期映射地址
ATIMx_CCR0A	0x03C	比较0A寄存器
ATIMx_CCR0B	0x040	比较0B寄存器
ATIM3_CCR1A	0x044	比较1A寄存器
ATIM3_CCR1B	0x048	比较1B寄存器
ATIM3_CCR2A	0x04C	比较2A寄存器
ATIM3_CCR2B	0x050	比较2B寄存器
ATIM3_CCR3	0x054	比较3寄存器
ATIM3_CRCH3	0x058	比较单元3控制寄存器

表 13-1 Timer 寄存器列表

### 13.4.1 模式 0 寄存器描述

#### 13.4.1.1 16 位模式重载寄存器 (ATIMx\_ARR)

偏移地址: 0x000

复位值: 0x0000 0000

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

位	符号	描述
31:16	Reserved	保留位
15:0	ARR	16位重载定时器重载值寄存器

#### 13.4.1.2 16 位模式计数寄存器 (ATIMx\_CNT)

偏移地址: 0x004

复位值: 0x0000 0000

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

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	16位重载定时器 计数值寄存器

### 13.4.1.3 32 位模式计数寄存器 (ATIMx\_CNT32)

偏移地址: 0x008

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT32[31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT32[15:0]															
RW															

位	符号	描述
31:0	CNT32	32位定时器 计数值寄存器 注: 仅在模式0 32位定时器自由计数模式下有效, 其他模式禁止写该寄存器

### 13.4.1.4 控制寄存器 (ATIMx\_M0CR)

偏移地址: 0x00C

复位值: 0x0060 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8								
Resvered	Mode			Resvered	UI	GATEP	GATE								
	RW				RW	RW	RW								
7	6	5	4	3	2	1	0								
Resvered	PRS			TOGEN	CT	MD	CEN								
	RW			RW	RW	RW	RW								

位	符号	描述
31:14	Reserved	保留位
13:12	MODE	工作模式 <b>00</b> 定时器模式0; 01 PWC模式 10 锯齿波模式; 11三角波模式
11	Reserved	保留位
10	UI	中断使能控制, 写1后使能中断
9	GATEP	保留位
8	GATE	保留位, 保持该位为 <b>0</b>
7	Reserved	保留位
6:4	PRS	内部时钟分频选择 000: 1; 001: 2; 010: 4; 011: 8; 100: 16; 101: 32; 110: 64; 111: 256;
3	TOGEN	模式0下翻转输出使能 1: 翻转输出使能 0: 翻转输出关闭CHA,CHB输出为低电平
2	CT	计数时钟选择 0: 内部计数时钟PCLK 1: 外部计数时钟ETR;
1	MD	模式选择32定时/16定时模式选择 0: 32位自由计数 1: 16位重载计数
0	CEN	定时器使能 0: 定时器停止; 1: 定时器使能

Preliminary  
to 立创商城

### 13.4.1.5 中断标志寄存器 (ATIMx\_IFR)

偏移地址: 0x010

复位值: 0x0000 0000

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

位	符号	描述
31:1	Reserved	保留位
0	UI	溢出中断

### 13.4.1.6 中断标志清除寄存器 (ATIMx\_ICLR)

偏移地址: 0x014

复位值: 0x0000 FFFF

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

位	符号	描述
31:1	Reserved	保留位
0	UI	溢出中断清除, 写0清除

### 13.4.1.7 死区时间寄存器 (ATIMx\_DTR)

偏移地址: 0x030

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MOE		Resvered												
	RW														

位	符号	描述
31:13	Reserved	保留位
12	MOE	翻转输出使能 0: 翻转输出为输入状态 1: 翻转端口为输出状态
11:0	Reserved	保留位

### 13.4.2 模式 1 寄存器描述

#### 13.4.2.1 16 位模式计数寄存器 (ATIMx\_CNT)

偏移地址: 0x004

复位值: 0x0000 0000

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

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	计数值寄存器

### 13.4.2.2 控制寄存器 (ATIMx\_M1CR)

偏移地址: 0x00C

复位值: 0x0060 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Reserved
15	14	13	12	11	10	9	8									
Resvered	Oneshot	Mode		Resvered	UI	Edg2nd	Edg1st									
	RW	RW			RW	RW	RW									
7	6	5	4	3	2	1	0									
Resvered	PRS		Resvered	CT	Resvered	CEN										
	RW			RW		RW										

位	符号	描述												
31:15	Reserved	保留位												
14	Oneshot	单次触发模式选择 1: 完成一次脉冲测量自动结束, 再次测量需要重新使能CTEN 0: 循环测量												
13:12	MODE	工作模式 00 定时器模式0; <b>01 PWC模式</b> 10 锯齿波模式; 11三角波模式												
11	Reserved	保留位												
10	UI	中断使能控制, 写1后使能中断 计数到0xFFFF才会溢出并产生溢出标志												
9	Edg2nd	脉冲宽度测量结束边沿选择												
8	Edg1st	脉冲宽度测量开始边沿选择 <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">edg2nd</td><td style="width: 25%;">Edg1st</td><td style="width: 25%;">00</td><td style="width: 25%;">01</td> </tr> <tr> <td>测量</td><td>上沿-上沿周期</td><td>低电平宽度</td><td>高电平宽度</td> </tr> <tr> <td></td><td></td><td></td><td>下沿-下沿周期</td> </tr> </table>	edg2nd	Edg1st	00	01	测量	上沿-上沿周期	低电平宽度	高电平宽度				下沿-下沿周期
edg2nd	Edg1st	00	01											
测量	上沿-上沿周期	低电平宽度	高电平宽度											
			下沿-下沿周期											
7	Reserved	保留位												
6:4	PRS	内部时钟分频选择 000: 1; 001: 2; 010: 4; 011: 8; 100: 16; 101: 32; 110: 64; 111: 256;												
3	Reserved	保留位												
2	CT	计数时钟选择 0: 内部计数时钟TCLK 1: 外部计数时钟ETR;												
1	Reserved	保留位												
0	CEN	脉冲宽度测量使能												

		0: 禁止; 1: 使能
--	--	-----------------

Preliminary  
to 立创商城

### 13.4.2.3 中断标志寄存器 (ATIMx\_IFR)

偏移地址: 0x010

复位值: 0x0000 0000

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

位	符号	描述
31:3	Reserved	保留位
2	PWC	脉冲宽度测量中断标志
1	Reserved	保留
0	UI	溢出中断标志

### 13.4.2.4 中断标志清除寄存器 (ATIMx\_ICLR)

偏移地址: 0x014

复位值: 0x0000 FFFF

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

位	符号	描述
31:3	Reserved	保留位
2	CA0F	脉冲宽度测量中断标志清除, 写0清除
1	Reserved	保留
0	UI	溢出中断清除, 写0清除

### 13.4.2.5 主从模式控制寄存器 (ATIMx\_MSCR)

偏移地址: 0x018

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Reserved
Resvered		IB0S	IA0S	Resvered			TS		RW						Resvered	

位	符号	描述
31:13	Resvered	保留位
12	IB0S	CH0B 输入选择 0: CH0B; 1: 内部触发TS选择信号;
11	IA0S	IA0 输入选择 0: CH0A; 1: CH0A CH1A CH2A XOR(TIM3) 注: 设置为1后端口的任意一个端口变化将导致输入变化
10:8	Resvered	保留位
7:5	TS	触发选择 000: 端口ETR的滤波相位选择后的信号ETFP ; 001: 内部互联信号 ITR0 010: 内部互联信号 ITR1; 011: 内部互联信号 ITR2; 100: 内部互联信号 ITR3; 101: 端口CH0A的边沿信号; 110: 端口CH0A的滤波相位选择后的信号IAFP 111: 端口CH0B的滤波相位选择后的信号IBFP;
4:0	Resvered	保留位

### 13.4.2.6 输出控制滤波 (ATIMx\_FLTR)

偏移地址: 0x01C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETP	FLTET		Resvered												
RW	RW														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resvered								FLTB0	Resvered		FLTA0	RW			
								RW							

位	符号	描述
31	ETP	ETR 输入相位选择 0: 同相位; 1: 反向输入;
30:28	FLTET	ETR滤波控制 滤波设置 0xx: 滤波无效 100: pclk 3个连续有效; 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效; 111: pclk/64 3个连续有效
27:7	Resvered	保留位
6:4	FLTB0	CHB 输入滤波控制; 滤波设置 0xx: 滤波无效 100: pclk 3个连续有效; 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效; 111: pclk/64 3个连续有效
3	Resvered	保留位
2:0	FLTA0	CHA 输入滤波控制 滤波设置 0xx: 滤波无效 100: pclk 3个连续有效; 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效; 111: pclk/64 3个连续有效

### 13.4.2.7 控制寄存器 (ATIMx\_CR0)

偏移地址: 0x024;

复位值: 0x0000 3000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved								CIEA	Reserved							
								RW								

位	符号	描述
31:9	Resvered	保留位
8	CIEA	CIEA 脉冲宽度测量完成中断使能 0: 禁止 1: 使能
7:0	Resvered	保留位

### 13.4.2.8 比较捕获寄存器 (ATIMx\_CCR0A)

偏移地址: 0x03C

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR0A								R							

位	符号	描述
31:16	Reserved	保留位
15:0	CCR0A	脉冲宽度测量结果

### 13.4.3 模式 2,3 寄存器描述

#### 13.4.3.1 16 位模式重载寄存器 (ATIMx\_ARR)

偏移地址: 0x000(0x038 dummy address)

复位值: 0x0000 0000

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

位	符号	描述
31:16	Reserved	保留位
15:0	ARR	重载寄存器/周期寄存器, 具有缓存功能 在计数器未使能或者缓存没有使能时, 缓存寄存器可以立刻更新

#### 13.4.3.2 16 位模式计数寄存器 (ATIMx\_CNT)

偏移地址: 0x004

复位值: 0x0000 0000

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

位	符号	描述
31:16	Reserved	保留位
15:0	CNT	重载定时器 计数值寄存器

注: 使用 PWM 比较输出时, 初始化 CNT 值需要小于 ARR 的值。

### 13.4.3.3 控制寄存器 (ATIMx\_M23CR)

偏移地址: 0x00C

复位值: 0x0060 0008

31	30	29	28	27	26	25	24
Resvered		UndE	OVE	DIR	BG	UG	TG
		RW	RW	RW/RO	W1	W1	W1

23	22	21	20	19	18	17	16
OCCE		CIS	BIE	TIE	TDE	URS	OCCS
RW		RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8
CSG	Oneshot		Mode	UDE	UIE	CFG	CRG
RW	RW		RW	RW	RW	RW	RW

7	6	5	4	3	2	1	0
BUFPEN		PRS		Pwm2s	CT	Comp	CTEN
RW		RW		RW	RW	RW	RW

位	符号	描述
31:30	Reserved	保留位
29	UndE	下溢出中断使能
28	OVE	上溢出中断使能
27	DIR	计数方向, 只有在锯齿波模式下可以写。其他模式下只读, 写无效 0: 上计数 1: 下计数 注: 从其他模式切换到中心对齐模式DIR自动清0。软件事件更新和从模式外部触发复位模式DIR自动清零。
26	BG	软件刹车, 自动清零 写1产生软件刹车; 写0无效
25	UG	软件更新, 自动清零 写1产生软件更新; 写0无效 初始化计数器并更新缓存寄存器到相应寄存器(缓存使能), 预除频计数器也会被清零。
24	TG	软件触发, 自动清零, 需要在触发模式SMS=2且mode=2/3下都可以触发。 写1产生软件触发; 写0无效

23	OCCE	OCREF 清除使能 1: OCREF_CLR信号可以清除OCREF输出 0: OCREF输出不受OCREF_CLR影响
22:21	CIS	中心对齐A比较中断模式(B比较中断单独控制在CRCHx寄存器中CISB) 00: 无中断, 01: 上沿中断, 10: 下沿中断, 11: 上下沿都中断
20	BIE	刹车中断使能 1: 中断使能 0: 中断禁止
19	TIE	触发中断使能 1: 中断使能 0: 中断禁止
18	TDE	保留位
17	URS	更新源 0: 上溢出/下溢出/软件更新UG/从模式复位; 1: 上溢出/下溢出
16	OCCS	OCREF 清除源选择 0: 电压比较器VC输出, VC选择在VCx_OUTCFG寄存器设置 1: ETR端口滤波相位选择后的信号 当OCCE有效时, OC_clr可以清除OCREF的比较输出信号为零, (当OCMx>1时有效), 下一个uev事件后继续比较输出
15	CSG	保留
14	Oneshot	单次触发模式选择 1: 发生事件更新后定时器停止。 0: 循环计数
13:12	MODE	工作模式 00 定时器模式0; 01 PWC模式 <b>10 锯齿波模式;</b> 11 三角波模式
11	UDE	保留位
10	UIE	UIE 更新中断使能 1: 使能更新中断 0: 禁止更新中断
9	CFG	保留位
8	CRG	保留位
7	BUFPEN	重载缓存使能 1: 周期缓存使能, 写入后下个事件更新后才影响到周期值。 0: 周期缓存无效, 写入后立刻影响周期值
6:4	PRS	内部时钟分频选择 000: 1; 001: 2; 010: 4; 011: 8; 100: 16; 101: 32; 110: 64; 111: 256;

3	PWM2S	OCREFA双点比较选择(缺省值为1) 0: 双点比较使能, 使用CCRA,CCRB比较控制OCREFA输出 1: 单点比较使能, 只使用CCRA比较控制OCREFA输出 注: OCREFB不受影响, 仍然使用CCRB控制OCREFB输出
2	CT	计数时钟选择 0: 内部计数时钟TCLK 1: 外部计数时钟ETR;
1	Comp	PWM互补输出模式选择 0: 独立PWM输出 1: 互补PWM输出
0	CTEN	定时器使能 0: 禁止; 1: 使能 可以外部触发使能, 在oneshot模式下结束时该位自动清零

Preliminary  
to be confirmed

### 13.4.3.4 中断标志寄存器 (ATIMx\_IFR)

偏移地址: 0x010

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved														CA3F	UndF	OVF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TIF	BIF	CB2E	CB1E	CB0E	CA2E	CA1E	CA0E	CB2F	CB1F	CB0F	CA2F	CA1F	CA0F	Res.	UIF	
RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	

位	符号	描述
31:19	Reserved	保留位
18	CA3F	通道CH3比较匹配标志
17	UndF	下溢出中断标志
16	OVF	上溢出中断标志
15	TIF	触发中断标志
14	BIF	刹车中断标志
13	CB2E	通道CH2B捕获数据丢失标志 0: 无数据丢失; 1: 数据丢失
12	CB1E	通道CH1B捕获数据丢失标志 0: 无数据丢失; 1: 数据丢失
11	CB0E	通道CH0B捕获数据丢失标志 0: 无数据丢失; 1: 数据丢失
10	CA2E	通道CH2A捕获数据丢失标志 0: 无数据丢失; 1: 数据丢失
9	CA1E	通道CH1A捕获数据丢失标志 0: 无数据丢失; 1: 数据丢失
8	CA0E	通道CH0A捕获数据丢失标志 0: 无数据丢失; 1: 数据丢失
7	CB2F	通道CH2B发生捕获/比较匹配标志 0: 没有发生 1: 发生
6	CB1F	通道CH1B发生捕获/比较匹配标志 0: 没有发生 1: 发生
5	CB0F	通道CH0B发生捕获/比较匹配标志 0: 没有发生 1: 发生
4	CA2F	通道CH2A发生捕获/比较匹配标志 0: 没有发生 1: 发生
3	CA1F	通道CH1A发生捕获/比较匹配标志

		0: 没有发生 1: 发生
2	CA0F	通道CH0A发生捕获/比较匹配标志 0: 没有发生 1: 发生
1	Reserved	保留位
0	UIF	事件更新中断标志 0: 没有发生 1: 发生

Preliminary  
to 业创商城

### 13.4.3.5 中断标志清除寄存器 (ATIMx\_ICLR)

偏移地址: 0x014

复位值: 0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved														CA3F	UndF	OVF
R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TIF	BIF	CB2E	CB1E	CB0E	CA2E	CA1E	CA0E	CB2F	CB1F	CB0F	CA2F	CA1F	CA0F	Res.	UIF	R1W0

位	符号	描述
31:19	Reserved	保留位
18	CA3F	通道CH3比较匹配标志清除, 写0清除
17	UndF	下溢出中断标志清除
16	OVF	上溢出中断标志清除
15	TIF	触发中断标志清除, 写0清除
14	BIF	刹车中断标志清除, 写0清除
13	CB2E	通道CH2B捕获数据丢失标志清除, 写0清除
12	CB1E	通道CH1B捕获数据丢失标志清除, 写0清除
11	CB0E	通道CH0B捕获数据丢失标志清除, 写0清除
10	CA2E	通道CH2A捕获数据丢失标志清除, 写0清除
9	CA1E	通道CH1A捕获数据丢失标志清除, 写0清除
8	CA0E	通道CH0A捕获数据丢失标志清除, 写0清除
7	CB2F	通道CH2B捕获/比较匹配标志清除, 写0清除
6	CB1F	通道CH1B捕获/比较匹配标志清除, 写0清除
5	CB0F	通道CH0B捕获/比较匹配标志清除, 写0清除
4	CA2F	通道CH2A捕获/比较匹配标志清除, 写0清除
3	CA1F	通道CH1A捕获/比较匹配标志清除, 写0清除
2	CA0F	通道CH0A捕获/比较匹配标志清除, 写0清除
1	Reserved	保留
0	UIF	事件更新中断清除, 写0清除

### 13.4.3.6 主从模式控制寄存器 (ATIMx\_MSCR)

偏移地址: 0x018

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	IB0S	IA0S	SMS		TS		MSM	CCDS	MMS						
	RW	RW	RW		RW		RW	RW	RW						

位	符号	描述
31:13	Reserved	保留位
12	IB0S	CH0B 输入选择 0: CH0B; 1: 内部触发TS选择信号;
11	IA0S	IA0 输入选择 0: CH0A 1: CH0A CH1A CH2A XOR 注: 设置为1后端口的任意一个端口变化将导致输入变化
10:8	SMS	从模式功能选择 000: 使用内部时钟; 001: 复位功能; 010: 触发模式; 011: 外部时钟模式 100: 正交编码计数模式1; 101: 正交编码计数模式2; 110: 正交编码计数模式3; 111: 门控功能
7:5	TS	触发选择 000: 端口ETR的滤波相位选择后的信号ETFP ; 001: 内部互联信号 ITR0 010: 内部互联信号 ITR1; 011: 内部互联信号 ITR2; 100: 内部互联信号 ITR3; 101: 端口CH0A的边沿信号; 110: 端口CH0A的滤波相位选择后的信号IAFP 111: 端口CH0B的滤波相位选择后的信号IBFP;
4	MSM	主从选择

		<p>0: 无延时 1: 延时使能, 使主送计数器同时启动. 注: 使用触发模式时, 从模式设置为0, 主模式设置为1, 可以使主从计数同时启动</p>
3	CCDS	<p>比较模式下 DMA 比较触发选择; 0 : 比较匹配触发DMA; 1: 比较匹配不触发DMA, 事件更新代替比较匹配触发DMA</p>
2:0	MMS	<p>主模式输出选择, 用于内部互联, 连接到其他定时器的ITRx 000: 软件更新UG, 写CR.UG 001: 定时器使能CTEN 010: 定时器事件更新UEV; 011: 比较匹配选择输出CMPSO; 100: 定时器比较参数输出OCREF0A 101: 定时器比较参数输出OCREF1A 110: 定时器比较参数输出OCREF2A 111: 定时器比较参数输出OCREF0B</p>

Preliminary  
to be confirmed

### 13.4.3.7 输出控制/输入滤波 (ATIMx\_FLTR)

偏移地址: 0x01C

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETP	FLTET	BKP		FLTBK	CCPB2		OCMB2	CCPA2		OCMA2					
-	-	-		-	-		FLTB2	-		FLTA2					
RW	RW	RW		RW	RW		RW	RW		RW					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCPB1	OCMB1	CCPA1		OCMA1	CCPB0		OCMB0	CCPA0		OCMA0					
-	FLTB1	-		FLTA1	CCPB0		FLTB0	CCPA0		FLTA0					
RW	RW	RW		RW	RW		RW	RW		RW					

位	符号	描述
31	ETP	ETR 输入相位选择 0: 同相位; 1: 反向输入;
30:28	FLTET	ETR滤波控制 滤波设置 0xx: 滤波无效 100: pclk 3个连续有效; 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效; 111: pclk/64 3个连续有效
27	BKP	刹车BK输入相位选择 0: 同相位; 1: 反向输入;
26:24	FLTBK	刹车输入滤波控制 滤波设置 0xx: 滤波无效 100: pclk 3个连续有效; 101: pclk/4 3个连续有效 110: pclk/16 3个连续有效; 111: pclk/64 3个连续有效
23	CCPB2	比较功能: CH2B通道比较输出相位控制 0: 正常输出; 1: 反向输出
22:20	OCMB2	比较功能: CH2B通道比较控制;参考OCMB0

	FLTB2	捕获功能: CH2B输入通道滤波设置, 参考FLTBK
19	CCPA2	比较功能: CH2A通道比较输出相位控制 0: 正常输出; 1: 反向输出
18:16	OCMA2 FLTA2	比较功能: CH2A通道比较控制;参考OCMB0 捕获功能: CH2A输入通道滤波设置, 参考FLTBK
15	CCPB1	比较功能: CH1B通道比较输出相位控制 0: 正常输出; 1: 反向输出
14: 12	OCMB1 FLTB1	比较功能: CH1B通道比较控制;参考OCMB0 捕获功能: CH1B输入通道滤波设置, 参考FLTBK
11	CCPA1	比较功能: CH1A通道比较输出相位控制 0: 正常输出; 1: 反向输出
10:8	OCMA1 FLTA1	比较功能: CH1A通道比较控制;参考OCMB0 捕获功能: CH1A输入通道滤波设置, 参考FLTBK
7	CCPB0	比较功能: 输出比较模式 CCPBx比较输出CHBx端口极性控制 0: 正常输出; 1: 反向输出 编码计数与从模式门控功能: 输入相位控制 CCPB0从模式门控, 复位, 外部触发, 外部时钟使用CH0B端口输入极性控制 0: 正常输入; 1: 反向输入
6:4	OCMB0 FLTB0	比较功能: CH0B通道比较控制 000: 强制为0 001: 强制为1 010: 比较匹配时强制为0 011: 比较匹配时强制为1 100: 比较匹配时翻转 101: 比较匹配时输出一个计数周期的高电平 110: PWM 模式1 单点比较: 上计数时CNT<CCRxy输出高, 下计数时CNT>CCRxy输出为低电平 双点比较: 1) 锯齿波上计数 CCRxA<CNT≤CCRxB输出为低电平 2) 锯齿波下计数 CCRxA<CNT≤CCRxB输出为高电平 3) 三角波上计数CNT<CCRxA输出高, 下计数CNT>CCRxB为低电平 111 PWM 模式2 单点比较: 上计数时CNT<CCRxy输出低, 下计数时CNT>CCRxy输出为高电平 双点比较:

		<p>1) 锯齿波上计数 <math>CCRx A \leq CNT &lt; CCRx B</math> 输出为高电平      2) 锯齿波下计数 <math>CCRx A \leq CNT &lt; CCRx B</math> 输出为低电平      3) 三角波上计数 <math>CNT &lt; CCRx A</math> 输出低, 下计数 <math>CNT &gt; CCRx B</math> 为高电平      捕获功能: CH0B输入通道滤波设置, 参考FLTBK</p>
3	CCPA0	<p>比较功能: CCPAx 比较输出 CHAx 端口极性控制      0: 正常输出;      1: 反向输出</p> <p>编码计数与从模式门控功能: 输入相位控制      CCPA0 从模式门控, 复位, 外部触发, 外部时钟使用 CH0A 端口输入极性控制      0: 正常输入;      1: 反向输入</p>
2:0	OCMA0 FLTA0	<p>比较功能: A 通道比较控制; 参考 OCMB0      捕获功能: CH0A 输入通道滤波设置, 参考 FLTBK</p>

Preliminary  
to be confirmed

### 13.4.3.8 ADC 触发控制寄存器 (ATIMx\_ADTR)

偏移地址: 0x020

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ADTE	CMB2E	CMB1E	CMB0E	CMA2E	CMA1E	CMA0E	UEVE
								RW	RW	RW	RW	RW	RW	RW	RW

位	符号	描述
31:8	Resvered	保留位
7	ADTE	使能ADC触发全局控制 1: 使能 0: 禁止
6	CMB2E	通道2B比较匹配触发ADC使能 1: 使能 0: 禁止
5	CMB1E	通道1B比较匹配触发ADC使能 1: 使能 0: 禁止
4	CMB0E	通道0B比较匹配触发ADC使能 1: 使能 0: 禁止
3	CMA2E	通道2A比较匹配触发ADC使能 1: 使能 0: 禁止
2	CMA1E	通道1A比较匹配触发ADC使能 1: 使能 0: 禁止
1	CMA0E	通道0A比较匹配触发ADC使能 1: 使能 0: 禁止
0	UEVE	事件更新触发ADC使能 1: 使能 0: 禁止

### 13.4.3.9 通道 0 控制寄存器 (ATIMx\_CRCH0)

偏移地址: 0x024

复位值: 0x0000 3000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCGB	CCGA	CISB	CDEB	CDEA	CIEB	CIEA	-	-	CSB	CSA	CFB	CRB	CFA	CRA	
CCGB	CCGA	CISB	CDEB	CDEA	CIEB	CIEA	BUFEB	BUFEA	CSB	CSA	bksb		bksa		
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW		

位	符号	描述
31:16	Resvered	保留位
15	CCGB	捕获比较B软件触发, 硬件自动清零。 在比较模式下只产生中断; 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
14	CCGA	捕获比较A软件触发, 硬件自动清零。 在比较模式下只产生中断; 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
13:12	CISB	B通道比较匹配设置 00 无匹配; 01上升匹配; 10 下降匹配; 11双匹配
11	CDEB	B捕获比较触发DMA使能 0: 禁止 1: 使能
10	CDEA	A捕获比较触发DMA使能 0: 禁止 1: 使能
9	CIEB	B捕获比较触发中断使能 0: 禁止 1: 使能
8	CIEA	A捕获比较触发中断使能 0: 禁止 1: 使能
7	BUFEB	比较功能: B比较缓存使能控制 0: 禁止 1: 使能
6	BUFEA	比较功能: A比较缓存使能控制 0: 禁止 1: 使能
5	CSB	B通道捕获/比较功能选择

		0: 比较模式 1: 捕获模式
4	CSA	A通道捕获/比较功能选择 0: 比较模式 1: 捕获模式
3	CFB	B通道下降沿捕获使能 0: 禁止 1: 使能
2	CRB	B通道上升沿捕获使能 0: 禁止 1: 使能
3:2	BKSB	B通道比较功能输出刹车电平控制 00: 高阻输出 01: 对输出无影响 10: 强制输出低电平 11: 强制输出高电平;
1	CFA	A通道下降沿捕获使能 0: 禁止 1: 使能
0	CRA	A通道上升沿捕获使能 0: 禁止 1: 使能
1:0	BKSA	A通道比较功能输出刹车电平控制 00: 高阻输出 01: 对输出无影响 10: 强制输出低电平 11: 强制输出高电平;

### 13.4.3.10 通道 1/2 控制寄存器 (TIM3\_CRCH1/2)

偏移地址:

TIM3\_CRCH1: 0x028;

TIM3\_CRCH2: 0x02C

复位值: 0x0000 3000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCGB	CCGA	CISB	CDEB	CDEA	CIEB	CIEA	-	-	CSB	CSA	CFB	CRB	CFA	CRA	
CCGB	CCGA	CISB	CDEB	CDEA	CIEB	CIEA	BUFEB	BUFEA	CSB	CSA	bksb		bksa		
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW		

位	符号	描述
31:16	Resvered	保留位
15	CCGB	捕获比较B软件触发, 硬件自动清零。 在比较模式下只产生中断; 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
14	CCGA	捕获比较A软件触发, 硬件自动清零。 在比较模式下只产生中断; 在捕获模式下产生中断并且捕获计数器的值到捕获寄存器中。
13:12	CISB	B通道比较匹配设置 00 无匹配; 01上升匹配; 10 下降匹配; 11双匹配
11	CDEB	保留位
10	CDEA	保留位
9	CIEB	B捕获比较触发中断使能 0: 禁止 1: 使能
8	CIEA	A捕获比较触发中断使能 0: 禁止 1: 使能
7	BUFEB	比较功能: B比较缓存使能控制 0: 禁止 1: 使能
6	BUFEA	比较功能: A比较缓存使能控制 0: 禁止 1: 使能
5	CSB	B通道捕获/比较功能选择

		0: 比较模式 1: 捕获模式
4	CSA	A通道捕获/比较功能选择 0: 比较模式 1: 捕获模式
3	CFB	B通道下降沿捕获使能 0: 禁止 1: 使能
2	CRB	B通道上升沿捕获使能 0: 禁止 1: 使能
3:2	BKSB	B通道比较功能输出刹车电平控制 00: 高阻输出 01: 保持之前输出1 10: 强制输出低电平 11: 强制输出高电平;
1	CFA	A通道下降沿捕获使能 0: 禁止 1: 使能
0	CRA	A通道上升沿捕获使能 0: 禁止 1: 使能
1:0	BKSA	A通道比较功能输出刹车电平控制 00: 高阻输出 01: 保持之前输出1 10: 强制输出低电平 11: 强制输出高电平;

### 13.4.3.11 死区时间寄存器 (ATIMx\_DTR)

偏移地址: 0x030

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	VCE	Safeen	MOE	AOE	BKE	DTEN	Bksel								DTR
RW	RW	RW	RW	RW	RW	RW	RW								RW

位	符号	描述
31:15	Reserved	保留位
14	VCE	保留位
13	Safeen	Safety 刹车使能 (osc fail,brown down,lockup) 0: 禁止 1: 使能
12	MOE	PWM输出使能 0: 禁止 1: 使能
11	AOE	PWM输出自动使能 0: 禁止 1: 使能
10	BKE	刹车使能 0: 禁止 1: 使能
9	DTEN	死区控制使能 0: 禁止 1: 使能
8	bksel	保留位
7:0	DTR	死区时间寄存器 DTR[7]=0 T=DTR[6:0]+2 2-129 step=1 DTR[7:6]=10 T={DTR[5:0]+64}*2 +2 130-256 step=2 DTR[7:5]=110 T={DTR[4:0]+32}*8 +2 258-506 step=8 DTR[7:5]=111 T={DTR[4:0]+32}*16 +2 514-1010 step=16

### 13.4.3.12 重复周期设置值寄存器 (ATIMx\_RCR)

偏移地址: 0x034

复位值: 0x0000 0000

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

位	符号	描述
31:10	Reserved	保留位
9	UD	重复计数控制下溢出屏蔽 0: 下计数溢出使能 1:下计数溢出屏蔽
8	OV	重复计数控制上溢出屏蔽 0: 下计数溢出使能 1:下计数溢出屏蔽
7:0	RCR	重复周期计数值 设置RCR+1个周期个由[9:8]控制的溢出产生事件更新, 计数器上溢出或下溢出时内部 RCR_CNT减1, 当计数到零后RCR_CNT重载RCR的值, 并且产生事件更新UEV信号

### 13.4.3.13 通道 0 比较捕获寄存器 (ATIMx\_CCR0A/B)

偏移地址:

ATIMx\_CCR0A : 0x03C;

ATIMx\_CCR0B : 0x040

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR0y															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CCR0y	比较捕获寄存器, 比较具有缓存功能(y=A,B)

### 13.4.3.14 通道 1/2 比较捕获寄存器 (TIM3\_CCR1/2 A/B)

偏移地址:

TIM3\_CCR1A: 0x044

TIM3\_CCR1B: 0x048

TIM3\_CCR2A: 0x04C

TIM3\_CCR2B: 0x050

复位值: 0x0000 0000

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

位	符号	描述
31:16	Reserved	保留位
15:0	CCRxy	比较捕获寄存器, 比较具有缓存功能(x=1,2; y=A,B)

### 13.4.3.15 通道 3 比较捕获寄存器 (TIM3\_CCR3)

偏移地址:

TIM3\_CCR3: 0x054

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3															
RW															

位	符号	描述
31:16	Reserved	保留位
15:0	CCR3	比较捕获寄存器, 比较具有缓存功能

### 13.4.3.16 通道3控制寄存器 (ATIM3\_CRCH3)

偏移地址: 0x058

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										C3EN	CIS	CDE	CIE	BUFE	
										RW	RW	RW	RW	RW	

位	符号	描述
31:6	Resvered	保留位
5	C3EN	通道比较使能
4:3	CIS	通道比较匹配设置 00 无匹配; 01上升匹配; 10 下降匹配; 11双匹配
2	CDE	比较触发DMA使能 0: 禁止 1: 使能
1	CIE	比较触发中断使能 0: 禁止 1: 使能
0	BUFE	比较功能: 比较缓存使能控制 0: 禁止 1: 使能

## 14 时钟校准模块 (CTRIM)

### 14.1 概述

该模块工作于校准模式时可自动实时校准 RCH / RCL 的输出频率，使 RCH/RCL 输出频率的精度不再受环境变化影响。该模块工作于定时器模式时，具备通用定时功能且在 DeepSleep 下仍可正常工作。

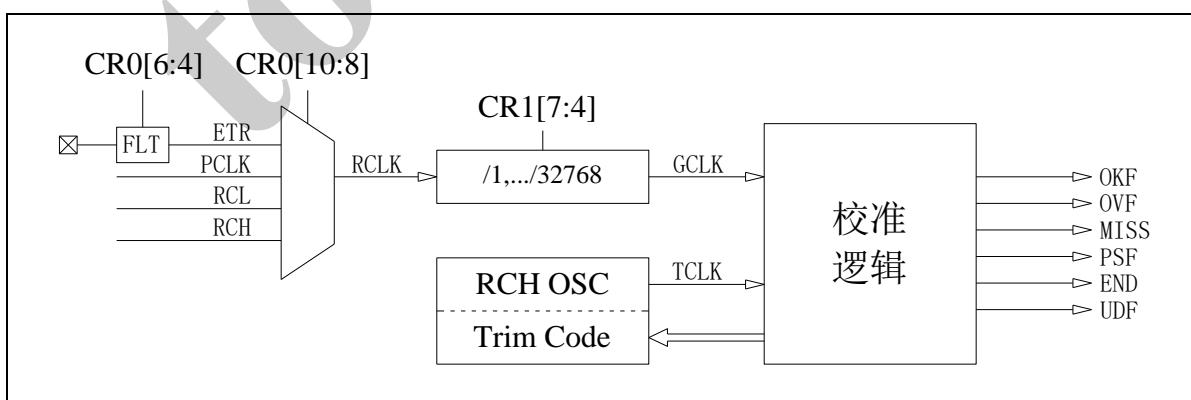
### 14.2 主要特性

- 校准模式：支持 RCH / RCL 自动实时校准 / 单次校准
- 校准模式：多种校准步长，快速逼近目标
- 定时模式：DeepSleep 模式下可工作
- 定时模式：支持自动唤醒，周期 30us ~ 65536s
- 定时模式：4 种时钟源可供选择
- 定时模式：翻转输出

### 14.3 功能描述

#### 14.3.1 RCH 时钟校准模式

当配置 CTRIM.CR1.MD 为 0x00 时，该模块工作于 RCH 校准模式。该模式可自动实时校准 RCH 的输出频率，使 RCH 输出频率的精度不再受环境变化的影响。校准 RCH 时，需要向该模块提供一个精准的低频时钟信号；该精准的低频时钟信号可以是片内的 PCLK，也可以通过外部 ETR 管脚输入。注意，RCLK 频率应小于 RCH 的频率的 1/2。RCH 时钟校准时的工作框图如下方所示。

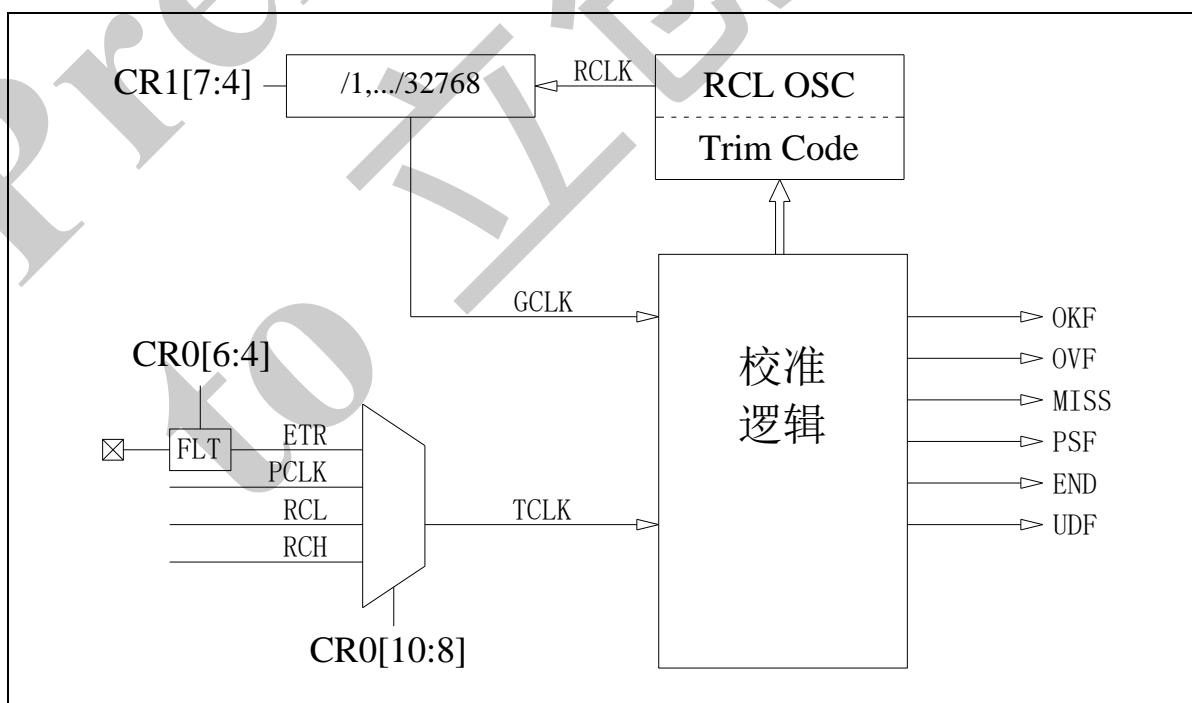


当自动校准开始工作时，误差计数器在每个 GCLK 的周期内对 TCLK 信号(RCH 振荡频率)从 CTRIM.ARR 开始进行递减计数，计到 0 后开始递增计数。若 GCLK 周期内的误差计数器的计数值小于 CTRIM\_FLIM 寄存器的值则置位 CTRIM\_ISR.OKF 以标识 RCH 输出频率已达到预期；若 GCLK 周期内的误差计数器的计数值大于 CTRIM\_FLIM 寄存器的值则自动调整 RCH 的 Trim Code 并进行下一轮计数。自动调整规则为：若计数方向为减计数则对 RCH 的 Trim Code 增加 CTRIM\_CR0.STEP；若计数方式为加计数则对 RCH 的 Trim Code 减少 CTRIM\_CR0.STEP。校准逻辑会自动减小 CTRIM\_CR0.STEP 的数值以实现 RCH 输出的频率逐次逼近目标值。

当误差计数器计数值递减为零时，生成 UDF 事件；当误差计数器计数值递加到 CTRIM\_ARR 时，停止计数并生成 MISS 事件。每当 GCLK 的上升沿到达时，误差计数器重新开始计数。

### 14.3.2 RCL 时钟校准模式

当配置 CTRIM.CR1.MD 为 0x01 时，该模块工作于 RCL 校准模式。该模式可自动实时校准 RCL 的输出频率，使 RCL 输出频率的精度不再受环境变化的影响。校准 RCL 时，需要向该模块提供一个精准的高频时钟信号；该精准的高频时钟信号可以是片内的 PCLK，也可以通过外部 ETR 管脚输入。RCL 时钟校准时的工作框图如下方所示。

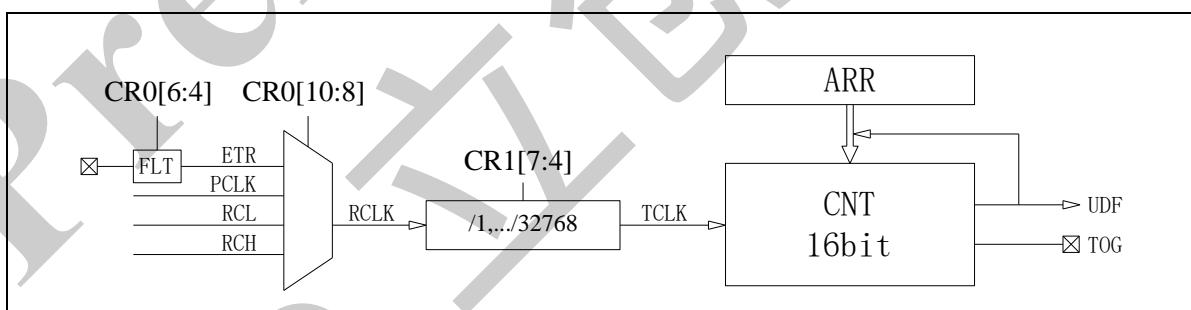


当自动校准开始工作时，误差计数器在每个 GCLK (分频后的 RCL 振荡频率) 的周期内对 TCLK 时钟信号从 CTRIM.ARR 开始进行递减计数，计到 0 后开始递增计数。若 GCLK 周期内的误差计数器的计数值小于 CTRIM\_FLIM 寄存器的值则置位 CTRIM\_ISR.OKF 以标识 RCL 输出频率已达到预期；若 GCLK 周期内误差计数器的计数值大于 CTRIM\_FLIM 寄存器的值则自动调整 RCL 的 Trim Code 并进行下一轮计数。自动调整规则为：若计数方向为减计数则对 RCL 的 Trim Code 减少 CTRIM\_CR0.STEP；若计数方式为加计数则对 RCL 的 Trim Code 增加 CTRIM\_CR0.STEP。校准逻辑会自动减小 CTRIM\_CR0.STEP 的数值以实现 RCL 输出的频率逐次逼近目标值。

当误差计数器计数值递减为零时，生成 UDF 事件；当误差计数器计数值递加到 CTRIM\_ARR 时，停止计数并生成 MISS 事件。每当 GCLK 的上升沿到达时，误差计数器重新开始计数。

### 14.3.3 低功耗定时器模式

当配置 CTRIM.CR1.MD 为 0x02 时，该模块工作于通用低功耗定时器模式。该模式下 CTIRM 具有通用定时器的定时及中断功能，若定时器的时钟为 RCL 则其在 DeepSleep 下可正常工作，支持自动唤醒，周期 30us ~ 65536s。通用低功耗定时器时的工作框图如下方所示。



定时器计数值从 ARR 减计数，当计数到 0 后产生下溢出，如果中断使能会产生下溢出中断，计数周期为  $ARR+1$ 。

定时器每溢出一次 TOG 输出电平翻转一次；用户可通过 GPIO 的辅助功能将 TOG 信号输出到管脚用于驱动外部电路。

## 14.4 软件操作流程

### 14.4.1 RCH 时钟自动校准示例

- Step1. 使用外设复位对 CTRIM 模块进行复位初始化。
- Step2. 配置 PA03 功能为 CTRIM.ETR，并向其提供 2KHz 的方波。
- Step3. 设置 CTRIM\_CR1.MD 为 0，选择 RCH 时钟校准模式。
- Step4. 设置 CTRIM\_CR0.SRC 为 0，外部 ETR 管脚输入信号作 RCLK。
- Step5. 设置 CTRIM\_CR1.PRS 为 1，对 RCLK 进行 2 分频，GCLK 频率为 1KHz。
- Step6. 配置 CTRIM\_ARR 为 47999，一个 GCLK 的计数周期内目标计数值为 48000。

注：ETR 管脚输入信号为 2KHz 方波，GCLK 的周期为 1ms。1ms 内误差计数器对 RCH 输出频率的理论计数值应为 48000。

- Step7. 设置 CTRIM\_CR0.FLIM 为 240，选择 RCH 校正精度为 0.5%。

注： $48000 * 0.005 = 240$ 。

- Step8. 设置 CTRIM\_CR0.STEP 为 4，选择初始校准步长为 16。
- Step9. 设置 CTRIM\_CR1.AUTO 为 1，使能自动校准。
- Step10. 设置 CTRIM\_CR1.EN 为 1，启动时钟校准。
- Step11. 等待 CTRIM\_ISR.END 为 1，自动校准完成。

### 14.4.2 RCL 时钟自动校准示例

- Step1. 使用外设复位对 CTRIM 模块进行复位初始化。
  - Step2. 配置 PA03 功能为 CTRIM.ETR，并向其提供 1MHz 的方波。
  - Step3. 设置 CTRIM\_CR1.MD 为 1，选择 RCL 时钟校准模式。
  - Step4. 设置 CTRIM\_CR0.SRC 为 0，外部 ETR 管脚输入信号作 TCLK。
  - Step5. 设置 CTRIM\_CR1.PRS 为 0x0A，对 RCLK(RCL 输出频率)进行 1024 分频，GCLK 目标频率为 32Hz。
  - Step6. 配置 CTRIM\_ARR 为 31249，一个 GCLK 的计数周期内目标计数值为 31250。
- 注：GCLK 的目标周期为 31.25ms，该时间内误差计数器对 TCLK(1MHz)的理论计数值应为 31250。

Step7. 设置 CTRIM\_CR0.FLIM 为 156，选择 RCL 校正精度为 0.5%。

注:  $31250 * 0.005 = 156$ 。

Step8. 设置 CTRIM\_CR0.STEP 为 4，选择初始校准步长为 16。

Step9. 设置 CTRIM\_CR1.AUTO 为 1，使能自动校准。

Step10. 设置 CTRIM\_CR1.EN 为 1，启动时钟校准。

Step11. 等待 CTRIM\_ISR.END 为 1，自动校准完成。

Preliminary  
to be confirmed

## 14.5 寄存器描述

基地址 0x40005000

寄存器	偏移地址	描述
CTRIM_ARR	0x00	自动重装值
CTRIM_CNT	0x04	计数器值
CTRIM_CR0	0x0C	控制寄存器0
CTRIM_CR1	0x10	控制寄存器1
CTRIM_IER	0x14	中断使能控制寄存器
CTRIM_ISR	0x18	中断及状态寄存器
CTRIM_ICR	0x1C	中断标志清除寄存器
CTRIM_FCAP	0x20	GCLK周期结束时计数器的值
CTRIM_TVAL	0x24	TRIM值寄存器, TRIM过程软件读需要读两次相同有效
CTRIM_FLIM	0x28	误差允许值

表 14-1 CTRIM 寄存器

### 14.5.1 自动装载寄存器 (CTRIM\_ARR)

偏移地址 0x00

复位值 0x0000FFFF

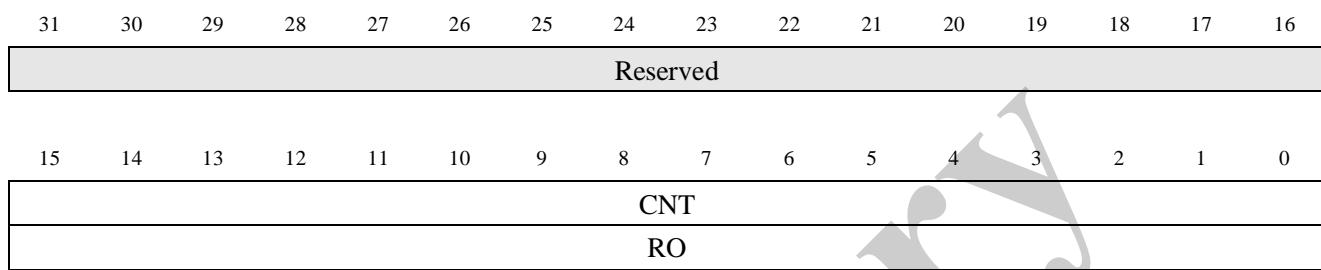
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR															
RW															

位	标记	功能描述
31:16	Reserved	保留
15:0	ARR	误差计数器重载值

### 14.5.2 误差计数器寄存器 (CTRIM\_CNT)

偏移地址 0x04

复位值 0x00000000



位	标记	功能描述
31:16	Reserved	保留
15:0	CNT	误差计数器计数值

### 14.5.3 控制寄存器 0 (CTRIM\_CR0)

偏移地址 0x0c

复位值 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SRC		Res.		EtrFlt		Res.		STEP		RW	

位	标记	功能描述
31:11	Reserved	保留
10:8	SRC	时钟来源选择 000: ETR 001: 保留 010: 保留 011: PCLK 100: RCL 101: RCH 110: 保留 111: 保留
7	Reserved	内部保留
6:4	EtrFlt	外部输入管脚(ETR)滤波配置 000: 无滤波 001: $F_{sample} = PCLK$ , N=2 010: $F_{sample} = PCLK$ , N=4 011: $F_{sample} = PCLK$ , N=6 100: $F_{sample} = PCLK/4$ , N=4 101: $F_{sample} = PCLK/4$ , N=6 110: $F_{sample} = PCLK/8$ , N=4 111: $F_{sample} = PCLK/8$ , N=6
3	Reserved	保留
2:0	STEP	TRIM初始步进量配置 000: 0x10 100: 0x10 001: 0x02 101: 0x20 010: 0x04 110: 0x40 011: 0x08 111: 0x80

注意：该寄存器只有在 EN 为 0 时才可以更改。

#### 14.5.4 控制寄存器1 (CTRIM\_CR1)

偏移地址 0x010

复位值 0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				OST	PRS			AUTO	MD		EN				
				RW	RW			RW	RW		RW				

位	标记	功能描述
31:9	Reserved	保留
8	OST	校准模式配置 0: 实时校准模式, TRIM过程持续进行 1: 单次校准模式, STEP变为1时结束
7:4	PRS	参考时钟预分频配置 0000: 保留, 不可配置 0001: 2分频 1001: 512分频 0010: 4分频 1010: 1024分频 0011: 8分频 1011: 2048分频 0100: 16分频 1100: 4096分频 0101: 32分频 1101: 8192分频 0110: 64分频 1110: 16384分频 0111: 128分频 1111: 32768分频
3	AUTO	RCH / RCL自动TRIM使能配置 1: 使能 0: 禁止
2:1	MD	工作模式配置 00: RCH校准模式, CR0.SRC需选择低速精准时钟 01: RCL校准模式, CR0.SRC需选择高速精准时钟 1x: 定时器模式
0	EN	定时器使能控制 1: 使能 0: 禁止

### 14.5.5 中断使能寄存器 (CTRIM\_IER)

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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved												OV	MISS	PS	END	UD
												RW	RW	RW	RW	RW

位	标记	功能描述
31:5	Reserved	保留
4	OV	TrimCode溢出中断使能配置 1: 使能 0: 禁止
3	MISS	计数失败中断使能配置 1: 使能 0: 禁止
2	PS	误差计数器完成中断使能配置 1: 使能 0: 禁止
1	END	自动Trim结束中断使能配置 1: 使能 0: 禁止
0	UD	误差计数器下溢中断使能配置 1: 使能 0: 禁止

### 14.5.6 中断和状态寄存器 (CTRIM\_ISR)

偏移地址 0x18

复位值 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								OK	DIR	OV	MISS	PS	END	UD	
								RO	RO	RO	RO	RO	RO	RO	RO

位	标记	功能描述
31:7	Reserved	保留
6	OK	Trim精度标志 1: Trim完成时精度满足FLIM要求 0: Trim完成时精度不满足FLIM要求
5	DIR	误差计数器计数方向 0: 递增计数, RCH输出频率高于目标频率, RCL输出频率低于目标频率 1: 递减计数, RCH输出频率低于目标频率, RCL输出频率高于目标频率
4	OV	TrimCode溢出标志 1: RCH/RCL的TrimCode已调整为0x000或0x1FF 0: RCH/RCL的TrimCode不为0x000或0x1FF
3	MISS	计数失败标志 1: 在GCLK的一个周期内, 误差计数器已记数到ARR 0: 在GCLK的一个周期内, 误差计数器未记数到ARR
2	PS	误差计数器完成标志 1: 误差计数器已完成一次计数并停止 0: 误差计数器正在计数中
1	END	自动Trim结束标志 1: 自动Trim已完成 0: 自动Trim进行中
0	UD	误差计数器下溢标志 1: 误差计数器已计数到0 0: 误差计数器未计数到0

### 14.5.7 中断状态清除寄存器 (CTRIM\_ICR)

偏移地址 0x1C

复位值 0x00000001F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								OK	OV	MISS	PS	END	UD		
								R1W0	R1W0	R1W0	R1W0	R1W0	R1W0		

位	标记	功能描述
31:6	Reserved	保留
5	OK	写0清除校准成功OK标志
4	OV	写0清除TrimCode溢出标志
3	MISS	写0清除计数失败标志
2	PS	写0清除误差计数器完成标志
1	END	写0清除自动Trim结束标志
0	UD	写0清除误差计数器下溢标志

### 14.5.8 误差结果寄存器 (CTRIM\_FCAP)

偏移地址 0x20

复位值 0x0000 FFFF

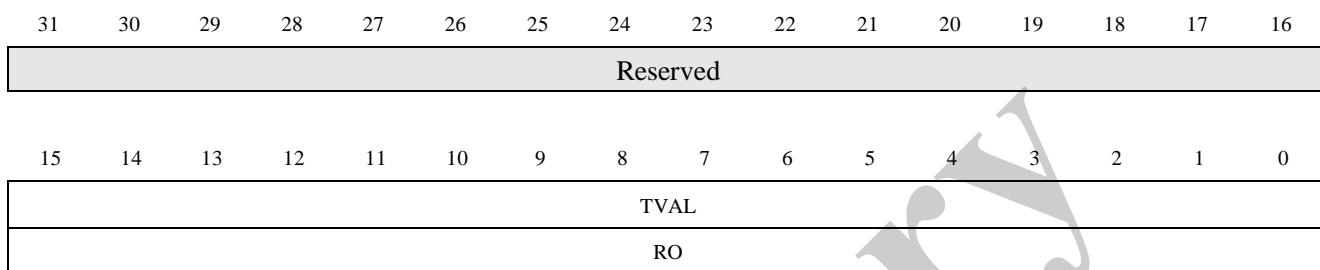
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCAP								RO							

位	标记	功能描述
31:16	Reserved	保留
15:0	FCAP	频率误差捕获值

### 14.5.9 TrimCode 寄存器 (CTRIM\_TVAL)

偏移地址 0x24

复位值 0x0000 0100

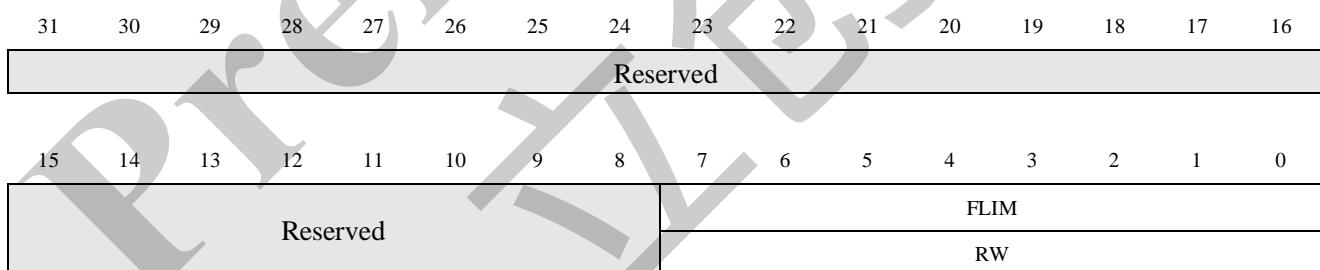


位	标记	功能描述
31:16	Reserved	保留
15:0	TVAL	自动TRIM结果，自动TRIM结束需要将TVAL的值写入相应的TRIM寄存器中

### 14.5.10 误差上限寄存器 (CTRIM\_FLIM)

偏移地址 0x28

复位值 0x0000 0010



位	标记	功能描述
31:8	Reserved	保留
7:0	FLIM	误差允许值

## 15 独立看门狗 (IWDT)

### 15.1 概述

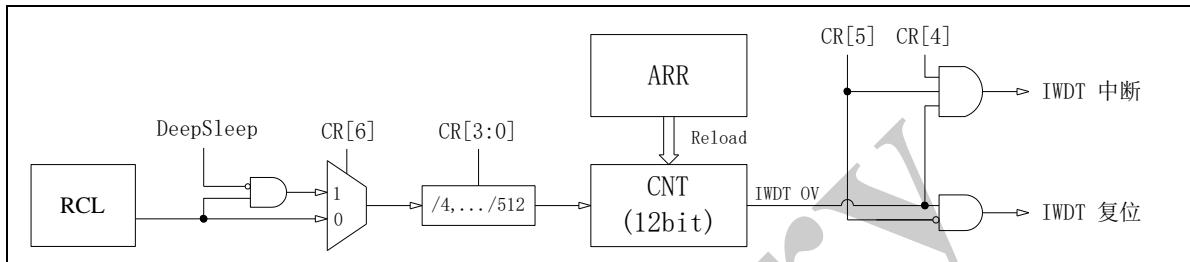
此器件具有一个嵌入式独立看门狗外设，具有安全性高和使用灵活的优点。此独立看门狗可检测并解决由软件错误导致的故障，在其计数值达到设定的超时值时触发中断或复位。该看门狗由内部低速时钟 RCL 驱动，即使主时钟发生故障时仍可正常工作。

### 15.2 主要特性

- 溢出周期 132us - 64s @ RCL32.8KHz
- 溢出时触发中断或复位
- 通过特定写序列，可以停止正在运行的看门狗
- DeepSleep 模式时可自动暂停计数

## 15.3 功能描述

### 15.3.1 功能框图



启动看门狗后，计数器从 0xFFFF 开始递减计数。每当向 IWDT\_KR 寄存器写入 0xAAAA 时，IWDT\_ARR 的值就会被重载到计数器。当计数值递减到 0x000 时，看门狗会产生溢出信号；当对计数器进行重载时计数值大于 IWDT\_WINR 寄存器的值也会产生溢出信号。配置 IWDT\_CR 相应控制位可使看门狗溢出后产生复位或中断。通过配置 IWDT\_CR 相应控制位可使看门狗在 DeepSleep 时自动暂停计数。

### 15.3.2 窗口看门狗模式

配置 IWDT\_WINR 寄存器的值小于 IWDT\_ARR 时，IWDG 工作于窗口看门狗模式。在该模式下，有两种条件触发看门狗溢出：计数值递减到 0x000、在计数值大于窗口值时对看门狗进行重载。更改 IWDT\_WINR 寄存器值时，硬件会自动对看门狗进行重载操作，将递减计数器值复位为 IWDT\_ARR。用户程序应当在计数值小于等于 IWDT\_WINR 寄存器数值且计数值递减到 0x000 前对看门狗进行重载操作。

### 15.3.3 独立看门狗模式

配置 IWDT\_WINR 寄存器的值大于等于 IWDT\_ARR 时，IWDG 工作于独立看门狗模式。在该模式下，仅有一种条件会触发看门狗溢出：计数值递减到 0x000。用户程序应当在计数值递减到 0x000 前对看门狗进行重载操作。

### 15.3.4 看门狗溢出处理

看门狗的溢出信号，可以触发中断也可以触发复位。当设置 IWDT\_CR.Action 为 0 时，看门狗的溢出信号会复位 MCU；当设置 IWDT\_CR.Action 为 1 且 IWDT\_CR.IE 为 1 时，看门狗的溢出信号会触发中断。

### 15.3.5 看门狗暂停

若设置 IWDT\_CR.Pause 为 1，则 MCU 进入 DeepSleep 时看门狗自动暂停运行；MCU 退出 DeepSleep 时，看门狗继续运行。使用该特性，则 MCU 在低功耗状态时不用频繁唤醒以喂狗，可实现更低的待机功耗。

### 15.3.6 看门狗停止及再次运行

向 IWDT\_KR 寄存器依次写入 0x1234, 0x5678 可以停止正在运行的看门狗。

向 IWDT\_KR 寄存器写入 0xCCCC 即可使看门狗重新运行，其配置保持不变。

### 15.3.7 寄存器写保护

IWDT\_CR、IWDT\_ARR 和 IWDT\_WINR 寄存器具有写保护。若需修改这三个寄存器，应向 IWDT\_KR 寄存器写入 0x5555 以解除这些寄存器的写保护；向 IWDT\_KR 寄存器写入其它值则会使能这些的寄存器写保护。

## 15.4 编程示例

### 15.4.1 配置 IWDT 为窗口看门狗

- Step1. 向 IWDT\_KR 寄存器写入 0xCCCC，使能 IWDG。
- Step2. 向 IWDT\_KR 寄存器写入 0x5555，解除寄存器写保护。
- Step3. 配置 IWDT\_CR，选择看门狗的计数时钟频率、溢出后动作、DeepSleep 特性。
- Step4. 配置 IWDT\_ARR，选择门狗的溢出周期。
- Step5. 等待 IWDT\_SR.ARNF 为 0。
- Step5. 向 IWDT\_WINR 写入需要的数值，配置窗口大小。
- Step6. 等待 IWDT\_SR.WINRF 变为 0，自动刷新计数器值为 IWDT\_ARR。
- Step7. 在计数值小于等于 IWDT\_WINR 寄存器数值且计数值递减到 0 前，向 IWDT\_KR 寄存器写入 0xAAAA，对 IWDT 进行重载。

### 15.4.2 配置 IWDT 为独立看门狗

- Step1. 向 IWDT\_KR 寄存器写入 0xCCCC，使能 IWDG。
- Step2. 向 IWDT\_KR 寄存器写入 0x5555，解除寄存器写保护。
- Step3. 配置 IWDT\_CR，选择看门狗的计数时钟频率、溢出后动作、DeepSleep 特性。
- Step4. 配置 IWDT\_ARR，选择门狗的溢出周期。
- Step5. 等待 IWDT\_SR 变为 0x10，分频系数、重载值均写入完成。
- Step6. 刷新计数器值为 IWDG\_ARR 的值 (IWDG\_KR = 0x0000 AAAA)
- Step7. 在计数值递减到 0 前，向 IWDT\_KR 寄存器写入 0xAAAA，对 IWDT 进行重载。

## 15.5 寄存器描述

IWDT 基地址: 0x40000F80

寄存器	偏移地址	描述
IWDT_KR	0x00	IWDT键值寄存器
IWDT_CR	0x04	IWDT控制寄存器
IWDT_ARR	0x08	IWDT重载寄存器
IWDT_SR	0x0C	IWDT状态寄存器
IWDT_WINR	0x10	IWDT窗口寄存器

表 15-1 IWDT 寄存器列表

### 15.5.1 键值寄存器 (IWDT\_KR)

偏移地址: 0x000

复位值: 0x0000 007F

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

位	符号	描述
31:16	Reserved	保留位
15:0	KR	<p>写入0xCCCC: 启动IWDT计数器</p> <p>写入0xAAAA: 重载IWDT计数值</p> <p>依次写入0x1234 - 0x5678: 停止看门狗</p> <p>写入0x5555: 解除IWDT_PR、IWDT_ARR、IWDT_WINR、IWDT_CR的写保护</p> <p>写入非0x5555: 使能IWDT_PR、IWDT_ARR、IWDT_WINR、IWDT_CR的写保护</p>

### 15.5.2 控制寄存器 (IWDT\_CR)

偏移地址: 0x004

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								Pause	Action	IE	PRS				
Reserved								RW	RW	RW	RW				

位	符号	描述
31:7	Reserved	保留位
6	Pause	Deepsleep模式下WDT暂停控制 0: DeepSleep模式下IWDT继续运行 1: DeepSleep模式下IWDT自动暂停
5	Action	IWDT溢出后动作配置 0: 看门狗溢出后产生复位 1: 看门狗溢出后产生中断
4	IE	IWDT中断使能控制 0: 禁止看门狗中断 1: 使能看门狗中断
3:0	PRS	配置看门狗计数时钟与RCL振荡器的分频比 111: 512分频 110: 256分频 101: 128分频 100: 64分频 011: 32分频 010: 16分频 001: 8分频 000: 4分频

注意:

- 需要向 IWDT\_KR 寄存器写入 0x5555 后, 才可修改本寄存器。
- IWDT\_SR.PRSF 为 0 时, 才可修改 IWDT\_CR.PRS。

### 15.5.3 重载寄存器 (IWDT\_ARR)

偏移地址: 0x008

复位值: 0x0000 0FFF

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

位	符号	描述
31:12	Reserved	保留位
11:0	ARR	IWDT重载值

注意:

- 向 IWDT\_KR 寄存器写入 0x5555 后, 才可修改本寄存器。
- IWDT\_SR.ARRF 为 0, 才可修改本寄存器。

### 15.5.4 窗口寄存器 (IWDT\_WINR)

偏移地址: 0x010

复位值: 0x0000 0FFF

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

位	符号	描述
31:12	Reserved	保留位
11:0	WINR	窗口比较器比较值

注意:

- 向 IWDT\_KR 寄存器写入 0x5555 后, 才可修改本寄存器。
- IWDT\_SR.WINRF 为 0, 才可修改本寄存器。

### 15.5.5 状态寄存器 (IWDT\_SR)

偏移地址: 0x00C

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved												RUN	OV	WINRF	ARRF	PRSF
												RO	R/W0	RO	RO	RO

位	符号	描述
31:5	Res.	保留位
4	RUN	WDT运行标志 1: WDT正在运行 0: WDT没有运行
3	OV	WDT溢出标志 1: WDT溢出, 写0清零 0: WDT没有溢出
2	WINRF	WINR寄存器更新标志 1: WINR寄存器正在更新 0: WINR寄存器更新完成 注意: 只有 WINRF 位为 0 时才可以更新WINR寄存器。
1	ARRF	ARR寄存器更新标志 1: ARR寄存器正在更新 0: ARR寄存器更新完成 注意: 只有 ARRF 位为 0 时才可以更新ARR寄存器。
0	PRSF	CR.PRS 寄存器更新标志 1: CR.PRS寄存器正在更新 0: CR.PRS寄存器更新完成 注意: 只有 PRSF 位为 0 时才可以更新CR.PRS寄存器。

## 16 窗口看门狗 (WWDT)

### 16.1 概述

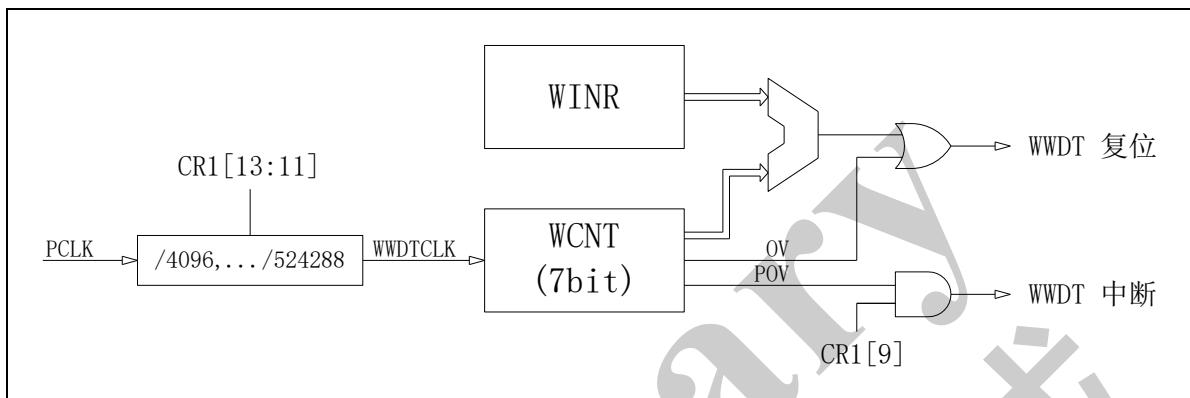
窗口看门狗 (WWDT) 通常用于监测由非预期错误所触发的软件故障。当计数器数值递减到 0x40 时产生中断，当计数器数值递减到 0x3F 时产生复位；若用户程序在计数器数值大于窗口值时更新计数器值则产生复位。故用户程序只能在限定的窗口内（计数器值小于窗口值且大于 0x3F）更新 WWDT 计数器的数值。

### 16.2 主要特性

- 6 比特递减计数器
- PCLK 分频时钟作为计数时钟
- 当 PCLK 为 48MHz 时，其超时时间为 85us ~ 699ms
- 计数值递减到 0x40 时产生中断
- 计数值递减到 0x3F 时产生复位
- 计数值大于窗口值时更新计数器产生复位

## 16.3 功能描述

### 16.3.1 功能框图



启动看门狗后，计数器 WCNT 的数值在每个 WWDTCLK 的上升沿递减 1。当计数器递减到 0x40 时看门狗会产生预溢出信号 POV，使能 WWDT\_CR1.IE 可使看门狗产生预溢出中断；当计数器递减到 0x3F 时看门狗会产生复位信号。用户程序可通过写 WWDT\_CR0.WCNT 以更新 WCNT 从而阻止看门狗产生复位信号。若户更新 WCNT 时，WCNT 的数值大于 WINR 的数值，则看门狗也会产生复位信号。

### 16.3.2 使能看门狗

系统复位后，看门狗处于关闭状态，设置 WWDT\_CR0.EN 为 1 即可使能看门狗。用户程序不能通过向 WWDT\_CR0.EN 写入 0 来关闭已经使能的看门狗。建议采用向 WWDT\_CR0 写入 0xFF 的方式使能看门狗，以确保使能看门狗时不会触发复位。

### 16.3.3 配置看门狗超时

通过合理配置 WWDT\_CR0.WCNT 和 WWDT\_CR1.PRS 即可获得所需要的看门狗超时周期。当 PCLK 为 48MHz 时，其超时周期的范围为 85us ~ 699ms。

超时时间的计算公式如下方所示：

$$T_{WWDT} = T_{PCLK} \times 4096 \times 2^{CR1.PRS} \times (CR0.WCNT - 0x3F)$$

### 16.3.4 更新递减计数器

向 WWDT\_CR0.WCNT 写入数据，即可更新 WWDT 递减计数器。若写入的数值小于等于 0x3F 则直接发生复位；若更新计数器时当前计数值大于窗口值则直接发生复

位。故用户程序只能在递减计数器的数值小于等于窗口值且大于 0x3F 时，对递减计数器进行更新。

### 16.3.5 预溢出中断

若递减计数器递减到 0x40 且 WWDT\_CR1.IE 为 1 则 WWDT 会产生预溢出中断，用户程序可以在中断服务程序中更新递减计数器以阻止 WWDT 产生复位；退出中断服务前用户程序应向 WWDT\_SR 寄存器写入 0x00 以清除 WWDT 预溢出标志。

## 16.4 编程示例

以下示例配置看门狗的溢出周期为 786432 个 PCLK，窗口为溢出周期的 1/3。

- Step1. 配置 WWDT\_CR1.PRS 为 2，WWDT 的计数时钟为 PCLK / 16384。
- Step2. 配置 WWDT\_CR1.WINR 为 0x4F，WWDT 的更新窗口为 0x4F~0x40。
- Step3. 配置 WWDT\_CR0.WCNT 为 0x6F，WWDT 的溢出周期为 786432 个 PCLK。
- Step4. 设置 WWDT\_CR0.EN 为 1，启动看门狗。
- Step5. 当递减计数器数值小于等于窗口值时，向 WWDT\_CR0.WCNT 写入 0x6F，以重载看门狗计数器。

## 16.5 寄存器描述

WWDT 基地址: 0x40006800

寄存器	偏移地址	描述
WWDT_CR0	0x00	WWDT控制寄存器0
WWDT_CR1	0x04	WWDT控制寄存器1
WWDT_SR	0x08	WWDT状态寄存器

表 16-1 WWDT 寄存器列表

### 16.5.1 控制寄存器 0 (WWDT\_CR0)

偏移地址: 0x000

复位值: 0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								EN	WCNT						
								RW1	RW						

位	符号	描述
31:8	Reserved	保留位
7	EN	WWDT 使能控制, 使能后不能禁止 1: 使能WWDT 0: 禁止WWDT
6:0	WCNT	读: WWDT计数器当前计数值 写: 更新WWDT的计数器当前计数值

### 16.5.2 控制寄存器1 (WWDT\_CR1)

偏移地址: 0x004

复位值: 0x0000 007F

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		PRS		Res.	IE		Res.					WINR			
		RW			RW							RW			

位	符号	描述
31:14	Reserved	保留位
13:11	PRS	WWDT计数时钟配置 000: PCLK / 4096 001: PCLK / 8192 010: PCLK / 16384 011: PCLK / 32768 100: PCLK / 65536 101: PCLK / 131072 110: PCLK / 262144 111: PCLK / 524288
10	Res.	保留位, 读出为0
9	IE	WWDT预溢出中断使能控制 1: 使能预溢出中断 0: 禁止预溢出中断
8:7	Res.	保留位
6:0	WINR	看门狗窗口值

### 16.5.3 配置寄存器 (WWDT\_SR)

偏移地址: 0x008

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														POV	
Reserved														RW0	

位	符号	描述
31:1	Res.	保留位
0	POV	WWDT预溢出标志 1: WWDT已发生预溢出, 写0以清除该标志 0: WWDT未发生预溢出

## 17 低功耗同步异步收发器 (LPUART)

### 17.1 概述

低功耗同步异步接收器 (LPUART) 是一种 UART，允许在低功耗模式下进行全双工 UART 通信。即使微控制器处于 DeepSleep 模式时，LPUART 也可以正常接收 UART 数据帧，接收完成中断可以唤醒微控制器。

### 17.2 主要特性

- 配置时钟 PCLK
- 传输时钟 SCLK (SCLK 可选择 RCL 以及 PCLK)
- 两种字符长度：8 比特、9 比特
- 三种校验方式：无检验、奇校验、偶校验
- 三种停止长度：1 比特、2 比特、1.5 比特
- 低功耗模式下收发数据
- 双线全双工通讯
- 单线半双工通讯
- 硬件流控 (RTS、CTS)
- 多机通讯、自动地址识别

## 17.3 功能描述

### 17.3.1 功能框图

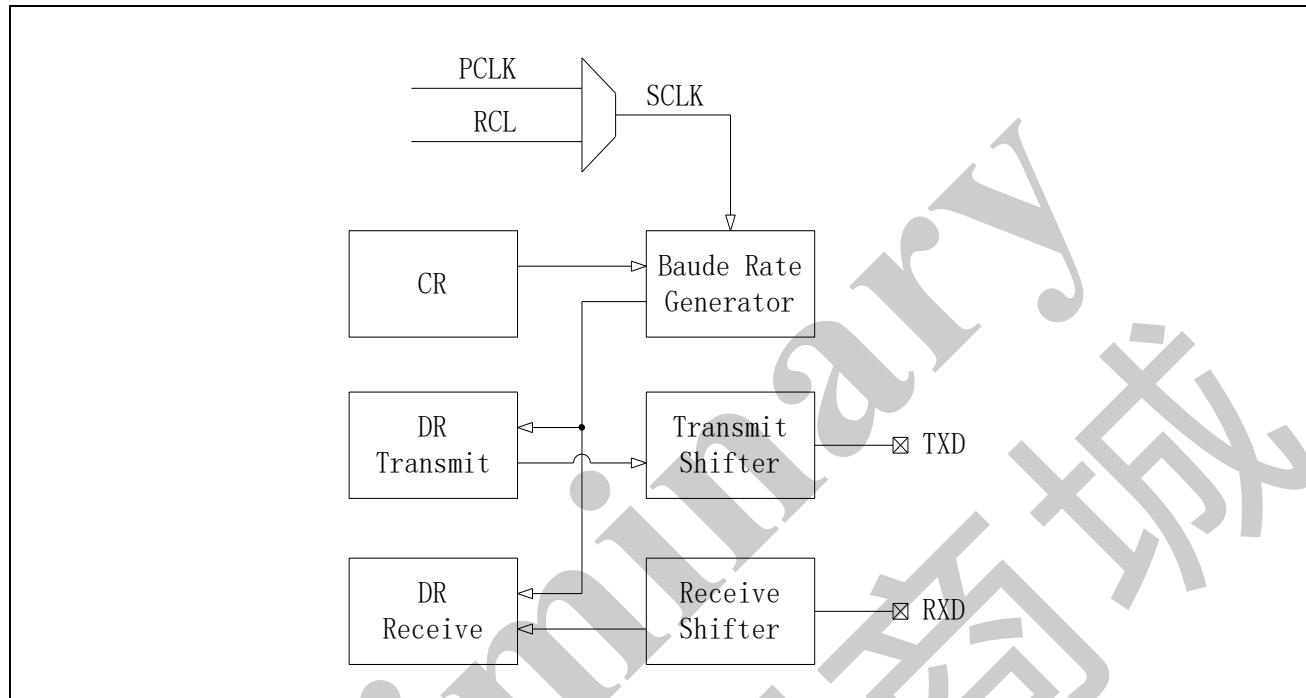


图 17-1 结构框图

### 17.3.2 时钟说明

LPUART 模块有两个时钟：配置时钟 PCLK 和传输时钟 SCLK。

- **配置时钟**

配置时钟 (PCLK) 用于系统 APB 总线对 LPUART 模块进行寄存器配置，配置时钟固定为 APB 总线时钟 PCLK。当系统进入深度休眠 (DeepSleep) 模式，PCLK 时钟将会停止。

- **传输时钟**

传输时钟 (SCLK) 用于 LPUART 数据收发逻辑工作，可选择内部低速时钟 (RCL) 以及 PCLK 时钟。

当系统进入深度休眠 (DeepSleep) 模式，如果 SCLK 选择内部低速时钟 (RCL)，LPUART 仍然可以进行正常的数据收发，而不受系统深度休眠 (DeepSleep) 模式的影响。

### 17.3.3 工作模式

LPUART 支持多种工作模式：同步半双工模式、异步全双工模式。通过 LPUARTx\_CR.MODE 和 LPUARTx\_CR.HDSEL 搭配，即可配置出所需要的各种工作模式。

#### 17.3.3.1 四种工作模式对比

配置 LPUARTx\_CR.MODE 可选择不同的传输模式：Mode0~Mode3。这四种工作模式的主要功能对比如下表所示：

工作模式		传输位宽	数据组成	波特率
Mode0	同步模式 半双工	8-bit	Data(8bit)	$BaudRate = \frac{f_{sclk}}{12}$
Mode1	异步模式 全双工	10-bit	Start(1bit) + Data(8bit) + Stop (1~2bit)	$BaudRate = \frac{f_{sclk}}{OVER * SCNT}$
Mode2	异步模式 全双工	11-bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop (1~2bit)	$BaudRate = \frac{f_{sclk}}{OVER}$
Mode3	异步模式 全双工	11-bit	Start (1bit) + Data(8bit) + B8(1bit) + Stop (1~2bit)	$BaudRate = \frac{f_{sclk}}{OVER * SCNT}$

表 17-1 Mode0/1/2/3 数据结构

注：

- Mode0 只能作为主机发送 LPUART 同步移位时钟，不可以作为从机接收外部输入的 LPUART 同步移位时钟；
- $f_{sclk}$  为 SCLK 的时钟频率；
- OVER 的定义详见 LPUARTx\_CR；
- SCNT 的定义详见 LPUARTx\_BRR；
- B8 数据位比较特殊，在不同应用下具有不同的含义，请参考以下表格：

应用场景	LPUARTx_CR. ADRDET	LPUARTx_CR. B8CFG[1:0]	DR8 数据含义
奇偶校验	--	01/10	接收时, DR8 是所收到的 8-Bit 数据的奇偶校验位; 发送时, DR8 是所发送的 8-Bit 数据的奇偶校验位;
多机通讯	1	--	DR8=1, 代表当前是地址帧; DR8=0, 代表当前是数据帧;
其他	0	00/11	接收/发送的 DATA[8]

表 17-2 B8 数据含义

注意:

- 当开启多机通讯模式, 接收数据奇偶校验自动关闭; 发送数据奇偶校验仍受 B8CFG 控制。

Preliminary  
to be confirmed

### 17.3.3.2 Mode0 数据收发说明

当设置 LPUARTx\_CR.RXEN 为 0 时 LPUART 工作于发送状态。将待发送的数据写入 LPUARTx\_DR 寄存器中，则数据将从 RXD 输出，同步移位时钟从 TXD 输出。

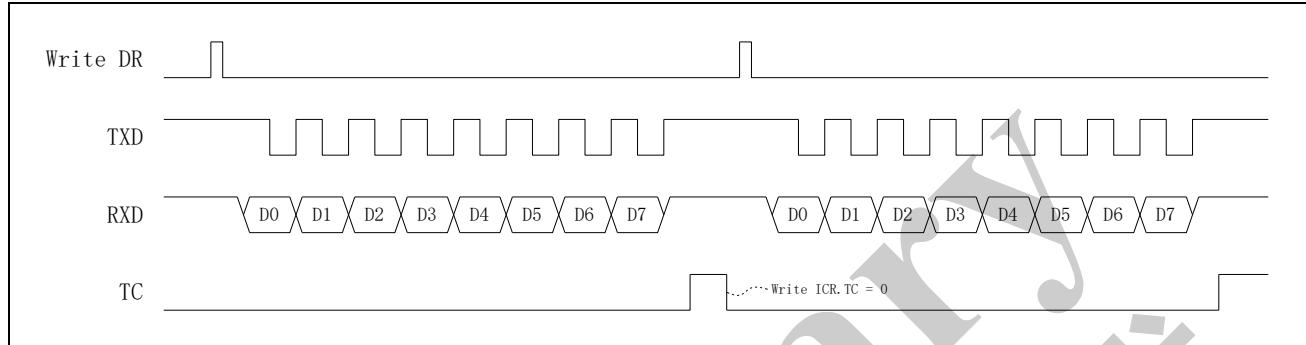


图 17-2 Mode0 发送数据

当设置 LPUARTx\_CR.RXEN 为 1 时 LPUART 工作于接收状态。每当 RC 标志为 0 时，TXD 管脚会输出 8 个同步移位时钟，数据将从 RXD 输入。每次将 RC 标志清零即可接收下一个字节。

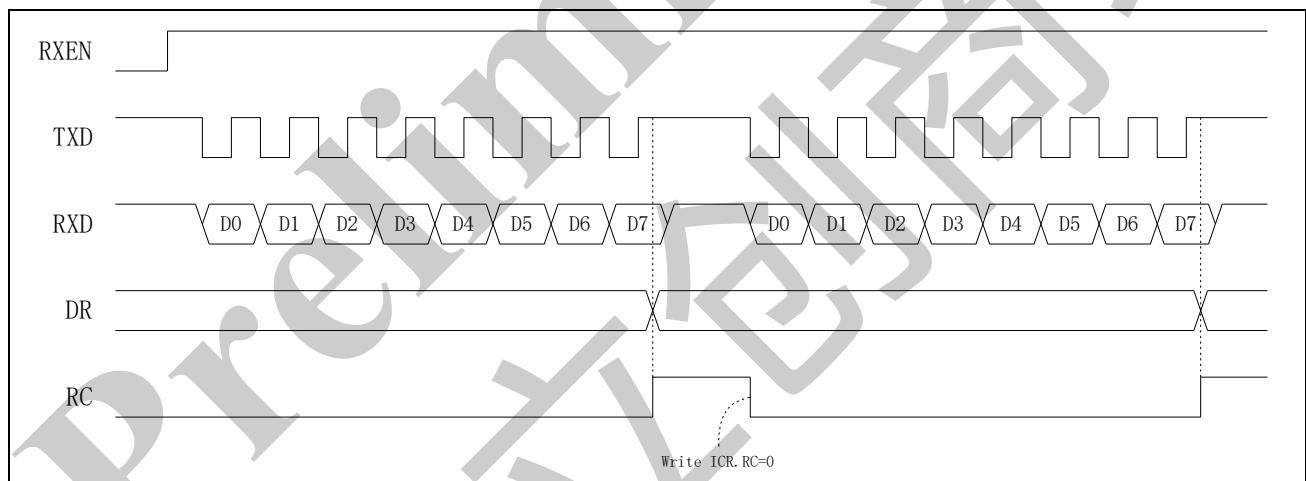


图 17-3 Mode0 接收数据

### 17.3.3.3 Mode1 数据收发说明

无论 LPUARTx\_CR.RXEN 为 0 或 1，都可以正常发送数据。当 LPUARTx\_DR 寄存器内有数据，则硬件自动将数据加载到发送移位寄存器并从 TXD 管脚移位输出。每当 LPUARTx\_DR 寄存器内的数据加载到移位寄存器时，TXE 标志硬件置 1。用户查询到 TXE 标志为 1 时，即可向 LPUARTx\_DR 寄存器写入下一个待发送的数据。每当 TXD 管脚发送完成一帧数据时，TC 标志会被硬件置 1 以指示当前的发送进展。

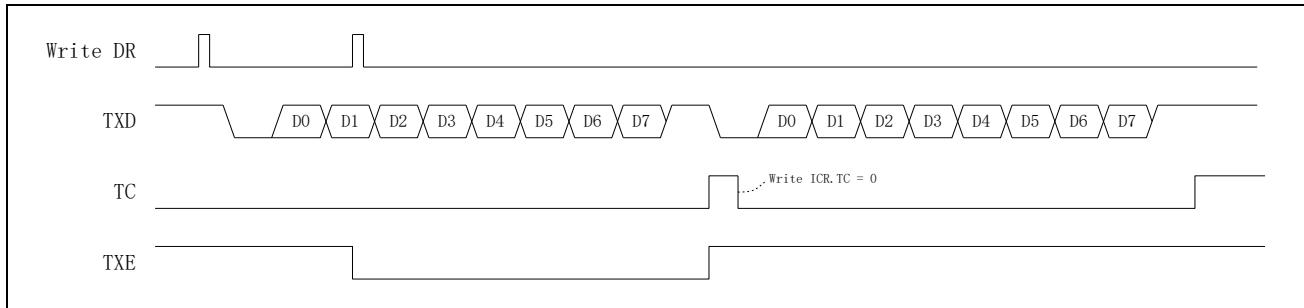


图 17-4 Mode1 发送数据

接收数据时，需要将 LPUARTx\_CR.RXEN 设置为 1。每当一帧数据接收完成，ISR.RC 标志会被硬件置 1，接收到的数据可以从 LPUARTx\_DR 寄存器中读出。

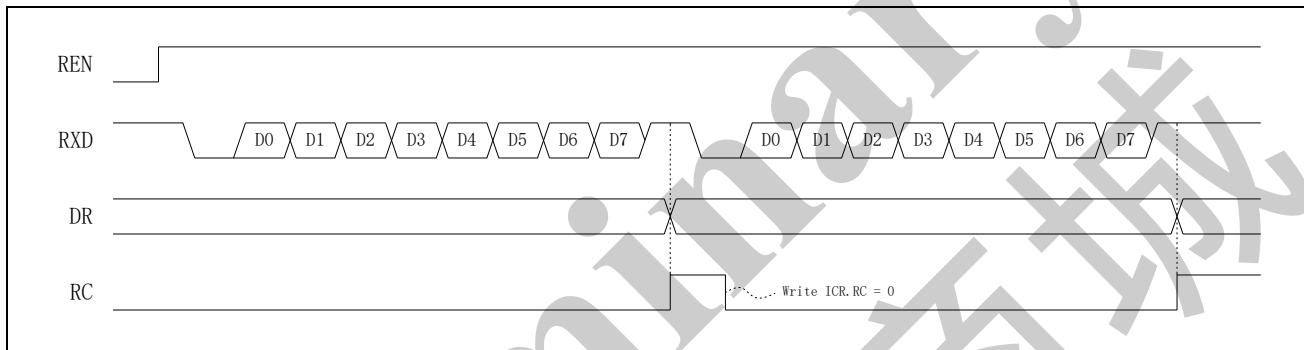


图 17-5 Mode1 接收数据

#### 17.3.3.4 Mode2/Mode3 数据收发说明

无论 LPUARTx\_CR.RXEN 为 0 或 1，都可以正常发送数据。当 LPUARTx\_DR 寄存器内有数据，则硬件自动将数据加载到发送移位寄存器并从 TXD 管脚移位输出。每当 LPUARTx\_DR 寄存器内的数据加载到移位寄存器时，TXE 标志硬件置 1。用户查询到 TXE 标志为 1 时，即可向 LPUARTx\_DR 寄存器写入下一个待发送的数据。每当 TXD 管脚发送完成一帧数据时，TC 标志会被硬件置 1 以指示当前的发送进展。

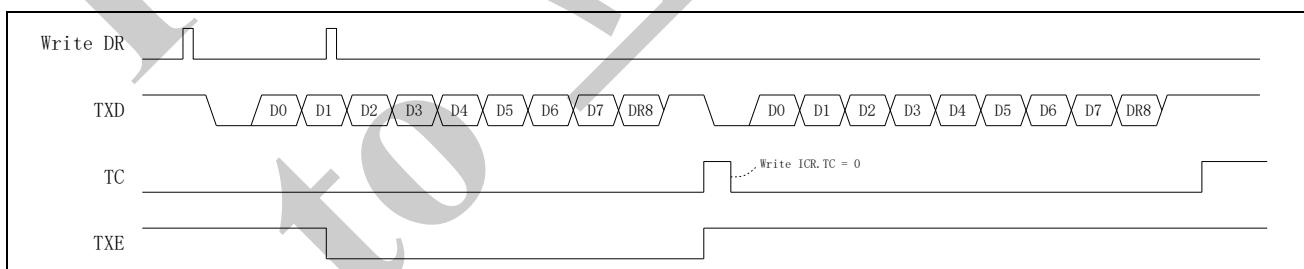


图 17-6 Mode2 发送数据

接收数据时，需要将 LPUARTx\_CR.RXEN 设置为 1。每当一帧数据接收完成，ISR.RC 标志会被硬件置 1，接收到的数据可以从 LPUARTx\_DR 寄存器中读出。

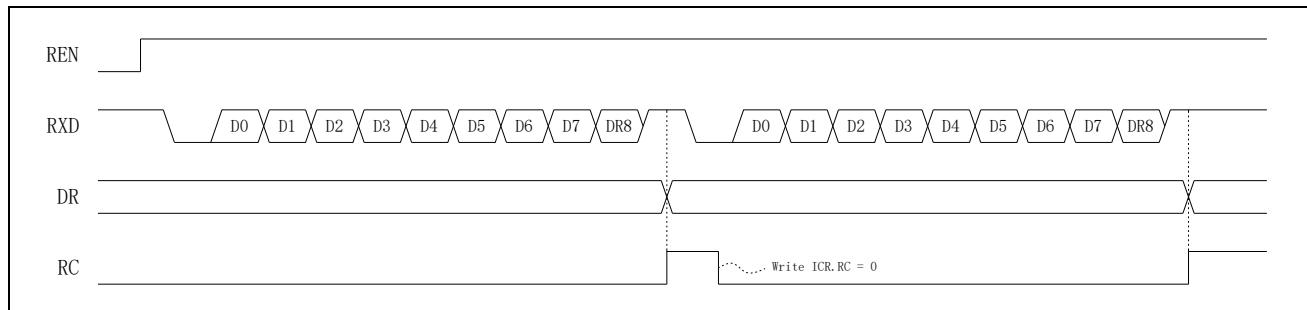


图 17-7 Mode2 接收数据

### 17.3.3.5 单总线数据收发说明

当设置 `LPUARTx_CR.HDSEL` 为 1 时, LPUART 仅使用一个 TXD 管脚进行数据的发送与接收; 用户需要将 TXD 管脚配置为开漏输出并外接上拉电阻。

当没有发送数据时, TXD 管脚处于输入态, 等待接收总线上其它主机发出数据。每当用户向 `LPUART_DR` 寄存器写入数据后, TXD 管脚变为输出态以便向总线上发送用户写到 `LPUART_DR` 内寄存器的数据。

为防止单总线出现冲突, 用户应采取适当的应用层保护措施, 以确保不会出现多个主机同时向总线发送数据。

### 17.3.4 波特率生成

Mode0~Mode3 生成波特率的公式不尽相同，详见下方所示。

$$\text{Mode0 波特率生成公式: } BaudRate = \frac{f_{SCLK}}{12}$$

$$\text{Mode1 波特率生成公式: } BaudRate = \frac{f_{SCLK}}{OVER * SCNT}$$

$$\text{Mode2 波特率生成公式: } BaudRate = \frac{f_{SCLK}}{OVER}$$

$$\text{Mode3 波特率生成公式: } BaudRate = \frac{f_{SCLK}}{OVER * SCNT}$$

注:

- $f_{SCLK}$  代表当前 SCLK 的频率;
- OVER 的定义详见 LPUARTx\_CR;
- SCNT 的定义详见 LPUARTx\_BRR。

#### 17.3.4.1 传输时钟为 4MHz 波特率设置示例 (Mode1 / Mode3)

波特率	SCLK = 4 MHz								
	OVER4			OVER8			OVER16		
	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%
2400	417	2398.08	-0.08%	208	2403.85	0.16%	104	2403.85	0.16%
4800	208	4807.69	0.16%	104	4807.69	0.16%	52	4807.69	0.16%
9600	104	9615.38	0.16%	52	9615.38	0.16%	26	9615.38	0.16%
19200	52	19230.77	0.16%	26	19230.77	0.16%	13	19230.77	0.16%
38400	26	38461.54	0.16%	13	38461.54	0.16%	7	35714.29	-6.99%
57600	17	58823.53	2.12%	9	55555.56	-3.55%	4	62500.00	8.51%
76800	13	76923.08	0.16%	7	71428.57	-6.99%	3	83333.33	8.51%
115200	9	111111.11	-3.55%	4	125000.00	8.51%	2	125000.00	8.51%
128000	8	125000.00	-2.34%	4	125000.00	-2.34%	2	125000.00	-2.34%
256000	4	250000.00	-2.34%	2	250000.00	-2.34%	1	250000.00	-2.34%
1000000	1	1000000.00	0.00%	1	500000.00	-50.00%	0	/	/

表 17-3 SCL 为 4MHz 波特率计算表

### 17.3.4.2 传输时钟为 8MHz 波特率设置示例 (Mode1 / Mode3)

波特率	SCLK = 8 MHz								
	OVER4			OVER8			OVER16		
	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%
2400	833	2400.96	0.04%	417	2398.08	-0.08%	208	2403.85	0.16%
4800	417	4796.16	-0.08%	208	4807.69	0.16%	104	4807.69	0.16%
9600	208	9615.38	0.16%	104	9615.38	0.16%	52	9615.38	0.16%
19200	104	19230.77	0.16%	52	19230.77	0.16%	26	19230.77	0.16%
38400	52	38461.54	0.16%	26	38461.54	0.16%	13	38461.54	0.16%
57600	35	57142.86	-0.79%	17	58823.53	2.12%	9	55555.56	-3.55%
76800	26	76923.08	0.16%	13	76923.08	0.16%	7	71428.57	-6.99%
115200	17	117647.06	2.12%	9	111111.11	-3.55%	4	125000.00	8.51%
128000	16	125000.00	-2.34%	8	125000.00	-2.34%	4	125000.00	-2.34%
256000	8	250000.00	-2.34%	4	250000.00	-2.34%	2	250000.00	-2.34%
1000000	2	1000000.00	0.00%	1	1000000.00	0.00%	1	500000.00	-50.00%
2000000	1	2000000.00	0.00%	1	1000000.00	-50.00%	0	/	/

表 17-4 SCLK 为 8MHz 波特率计算表

### 17.3.4.3 传输时钟为 16MHz 波特率设置示例 (Mode1 / Mode3)

波特率	SCLK = 16 MHz								
	OVER4			OVER8			OVER16		
	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%
2400	1667	2399.52	-0.02%	833	2400.96	0.04%	417	2398.08	-0.08%
4800	833	4801.92	0.04%	417	4796.16	-0.08%	208	4807.69	0.16%
9600	417	9592.33	-0.08%	208	9615.38	0.16%	104	9615.38	0.16%
19200	208	19230.77	0.16%	104	19230.77	0.16%	52	19230.77	0.16%
38400	104	38461.54	0.16%	52	38461.54	0.16%	26	38461.54	0.16%
57600	69	57971.01	0.64%	35	57142.86	-0.79%	17	58823.53	2.12%
76800	52	76923.08	0.16%	26	76923.08	0.16%	13	76923.08	0.16%
115200	35	114285.71	-0.79%	17	117647.06	2.12%	9	111111.11	-3.55%
128000	31	129032.26	0.81%	16	125000.00	-2.34%	8	125000.00	-2.34%
256000	16	250000.00	-2.34%	8	250000.00	-2.34%	4	250000.00	-2.34%
1000000	4	1000000.00	0.00%	2	1000000.00	0.00%	1	1000000.00	0.00%
2000000	2	2000000.00	0.00%	1	2000000.00	0.00%	1	1000000.00	-50.00%
4000000	1	4000000.00	0.00%	1	2000000.00	-50.00%	0	/	/

表 17-5 SCLK 为 16MHz 波特率计算表

#### 17.3.4.4 传输时钟为 24MHz 波特率设置示例 (Mode1 / Mode3)

波特率	SCLK = 24 MHz								
	OVER4			OVER8			OVER16		
	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%
2400	2500	2400.00	0.00%	1250	2400.00	0.00%	625	2400.00	0.00%
4800	1250	4800.00	0.00%	625	4800.00	0.00%	313	4792.33	-0.16%
9600	625	9600.00	0.00%	313	9584.66	-0.16%	156	9615.38	0.16%
19200	313	19169.33	-0.16%	156	19230.77	0.16%	78	19230.77	0.16%
38400	156	38461.54	0.16%	78	38461.54	0.16%	39	38461.54	0.16%
57600	104	57692.31	0.16%	52	57692.31	0.16%	26	57692.31	0.16%
76800	78	76923.08	0.16%	39	76923.08	0.16%	20	75000.00	-2.34%
115200	52	115384.62	0.16%	26	115384.62	0.16%	13	115384.62	0.16%
128000	47	127659.57	-0.27%	23	130434.78	1.90%	12	125000.00	-2.34%
256000	23	260869.57	1.90%	12	250000.00	-2.34%	6	250000.00	-2.34%
1000000	6	1000000.00	0.00%	3	1000000.00	0.00%	2	750000.00	-25.00%
2000000	3	2000000.00	0.00%	2	1500000.00	-25.00%	1	1500000.00	-25.00%
6000000	1	6000000.00	0.00%	1	3000000.00	-50.00%	0	/	/

表 17-6 SCLK 为 24MHz 波特率计算表

#### 17.3.4.5 传输时钟为 32MHz 波特率设置示例 (Mode1 / Mode3)

波特率	SCLK = 32 MHz								
	OVER4			OVER8			OVER16		
	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%
2400	3333	2400.24	0.01%	1667	2399.52	-0.02%	833	2400.96	0.04%
4800	1667	4799.04	-0.02%	833	4801.92	0.04%	417	4796.16	-0.08%
9600	833	9603.84	0.04%	417	9592.33	-0.08%	208	9615.38	0.16%
19200	417	19184.65	-0.08%	208	19230.77	0.16%	104	19230.77	0.16%
38400	208	38461.54	0.16%	104	38461.54	0.16%	52	38461.54	0.16%
57600	139	57553.96	-0.08%	69	57971.01	0.64%	35	57142.86	-0.79%
76800	104	76923.08	0.16%	52	76923.08	0.16%	26	76923.08	0.16%
115200	69	115942.03	0.64%	35	114285.71	-0.79%	17	117647.06	2.12%
128000	63	126984.13	-0.79%	31	129032.26	0.81%	16	125000.00	-2.34%
256000	31	258064.52	0.81%	16	250000.00	-2.34%	8	250000.00	-2.34%
1000000	8	1000000.00	0.00%	4	1000000.00	0.00%	2	1000000.00	0.00%
2000000	4	2000000.00	0.00%	2	2000000.00	0.00%	1	2000000.00	0.00%
4000000	2	4000000.00	0.00%	1	4000000.00	0.00%	1	2000000.00	-50.00%
8000000	1	8000000.00	0.00%	1	4000000.00	-50.00%	0	/	/

表 17-7 SCLK 为 32MHz 波特率计算表

### 17.3.4.6 传输时钟为 48MHz 波特率设置示例 (Mode1 / Mode3)

波特率	SCLK = 48 MHz								
	OVER4			OVER8			OVER16		
	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%	SCNT	实际波特率	误差%
2400	5000	2400.00	0.00%	2500	2400.00	0.00%	1250	2400.00	0.00%
4800	2500	4800.00	0.00%	1250	4800.00	0.00%	625	4800.00	0.00%
9600	1250	9600.00	0.00%	625	9600.00	0.00%	313	9584.66	-0.16%
19200	625	19200.00	0.00%	313	19169.33	-0.16%	156	19230.77	0.16%
38400	313	38338.66	-0.16%	156	38461.54	0.16%	78	38461.54	0.16%
57600	208	57692.31	0.16%	104	57692.31	0.16%	52	57692.31	0.16%
76800	156	76923.08	0.16%	78	76923.08	0.16%	39	76923.08	0.16%
115200	104	115384.62	0.16%	52	115384.62	0.16%	26	115384.62	0.16%
128000	94	127659.57	-0.27%	47	127659.57	-0.27%	23	130434.78	1.90%
256000	47	255319.15	-0.27%	23	260869.57	1.90%	12	250000.00	-2.34%
1000000	12	1000000.00	0.00%	6	1000000.00	0.00%	3	1000000.00	0.00%
2000000	6	2000000.00	0.00%	3	2000000.00	0.00%	2	1500000.00	-25.00%
4000000	4	3000000.00	0.00%	2	3000000.00	0.00%	1	3000000.00	0.00%
6000000	3	4000000.00	0.00%	2	3000000.00	/	1	3000000.00	/
12000000	1	12000000.00	0.00%	1	6000000.00	/	0	/	/

表 17-8 SCLK 为 48MHz 波特率计算表

### 17.3.5 帧错误检测

当 LPUART 工作在 Mode1/2/3 时，具有帧错误检测功能。在接收数据时，若未在预期时间内识别出停止位，即为帧错误。检测到帧错误时，LPUARTx\_ISR.FE 会被硬件置 1，用户程序应及时清除该标志。

### 17.3.6 多机通讯

当工作在 Mode2/3 时，将 LPUARTx\_CR.ADRDET 设置为 1，即可使能多机通讯功能。主机可以发送地址帧及数据帧，从机可以在收到匹配的地址帧后才接收数据帧。

#### 17.3.6.1 主机发送地址帧及数据帧

当 LPUARTx\_DR[8] 为 1 则发送地址帧，LPUARTx\_DR[7:0] 为目从机的地址。

当 LPUARTx\_DR[8] 为 0 则发送数据帧，LPUARTx\_DR[7:0] 为待发送的数据。

#### 17.3.6.2 从机接收地址帧及数据帧

设置 LPUARTx\_CR.ADRDET 为 1 时，从机只接收与自己地址相匹配的地址帧，忽略数据帧及不匹配的地址帧。当从机从 RXD 接收到地址帧时，硬件自动对比地址帧内的地址与本机地址；只有两者匹配时才会将收到的地址存入 LPUARTx\_DR 寄存器并置位 LPUARTx\_ISR.RC。

设置 LPUARTx\_CR.ADRDET 为 0 时，从机只接收数据帧并忽略地址帧。用户程序应当在地址匹配后才设置 LPUARTx\_CR.ADRDET 为 0 以接收主机后继发送的数据帧。

#### 17.3.6.3 从机地址配置

从机地址由广播地址和用户配置地址组成：

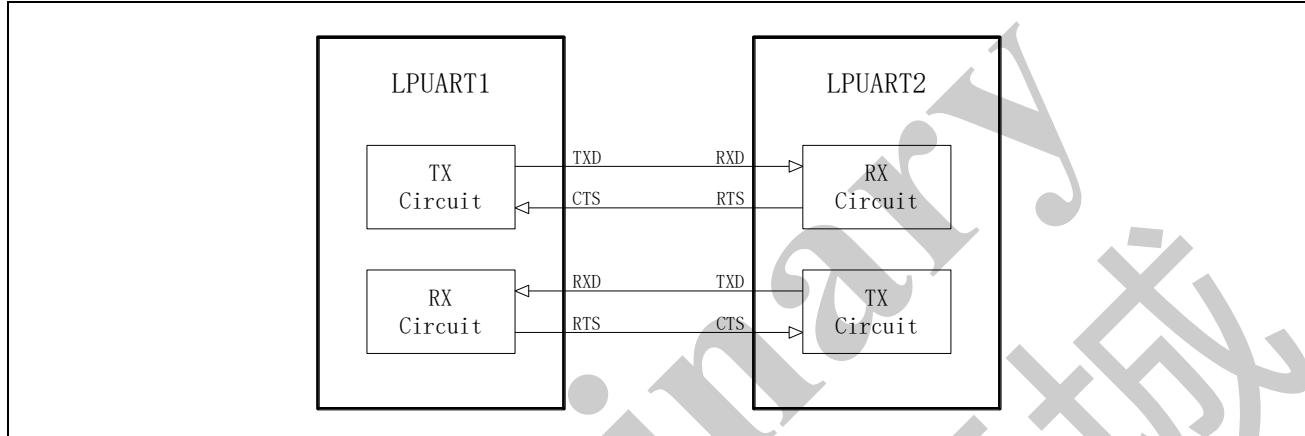
- 广播地址：0xFF
- 用户配置地址：LPUARTx\_ADDR & LPUARTx\_AddrMask

收到地址帧时，检测到从总线收到的地址与 LPUARTx\_AddrMask 的值相与后的值与用户配置地址相同，则认为收到了与从机地址相匹配的地址帧。

例：LPUARTx\_ADDR = 0x55，LPUARTx\_AddrMask = 0xF0，则从机可以匹配的地址为 0xFF(广播地址)、0x50 ~ 0x5F(用户配置地址)。

### 17.3.7 硬件流控

通过增加 CTS 和 RTS 信号，可以实现 LPUART 硬件流控的功能，即 LPUART 硬件模块根据 CTS 和 RTS 的高低电平自动控制数据的收发，而无需通过软件来判断。两个 LPUART 模块之间的硬件流控示意图如下所示：



#### 17.3.7.1 RTS 流控

当设置 `LPUARTx_CR.RTSEN` 为 1 时，RTS 管脚指示本模块的接收状态：

- 接收缓存空时，RTS 变为低电平，通知发送端发送下一帧数据
- 接收缓存满时，RTS 变为高电平，通知发送端暂停发送下一帧数据

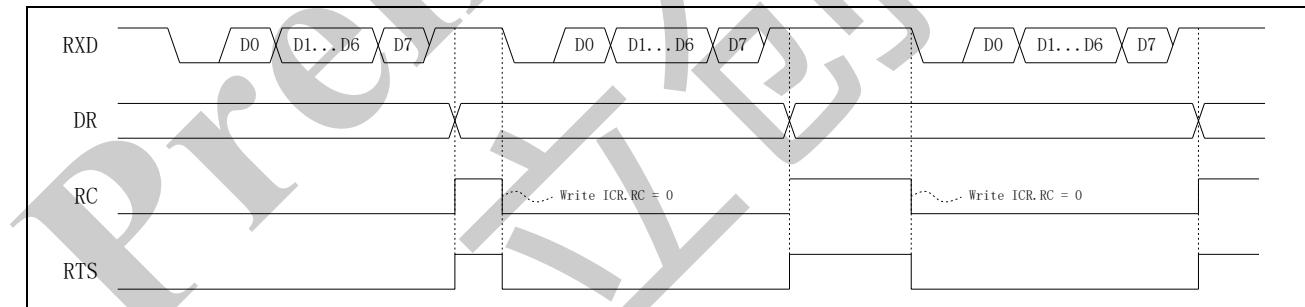


图 17-8 RTS 硬件流控信号

#### 17.3.7.2 CTS 流控

当设置 `LPUARTx_CR.CTSEN` 为 1 时，CTS 管脚的电平控制本模块的发送节奏：

- 如果 CTS 为低电平，则 LPUART 发送下一帧数据。
- 如果 CTS 为高电平，则 LPUART 发送完当前帧后，等待 CTS 变成低电平。
- 如果 CTS 电平发生变化，`LPUARTx_ISR.CTSIF` 自动置 1。
- 从 `LPUARTx_ISR.CTS` 可以获取当前 CTS 管脚的电平值。

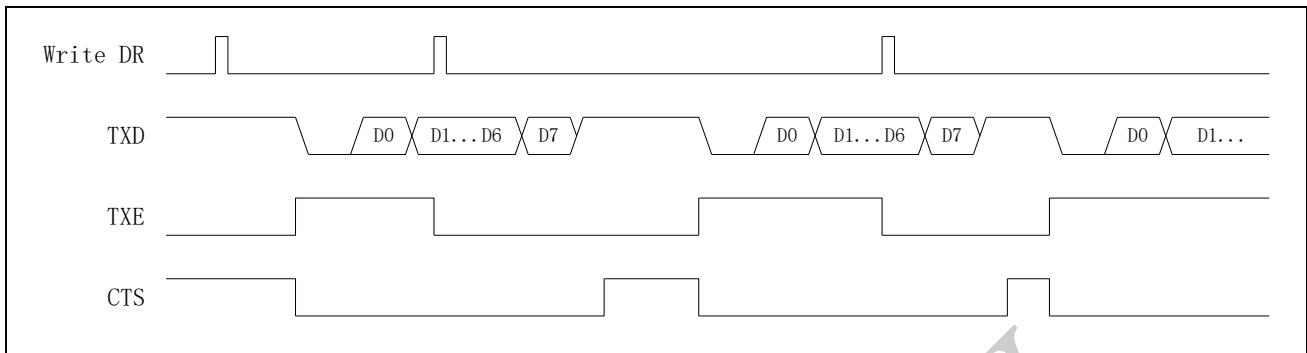


图 17-9 CTS 硬件流控信号

Preliminary  
to be confirmed

## 17.4 编程示例

### 17.4.1 发送数据示例

- Step1. 按 GPIO 章节管脚数字复用功能的相关描述，将 TXD 映射到需要的管脚；并配置 TXD 为 CMOS 输出。
- Step2. 设置 SysCtrl\_PeriphClkEn0.LpUartx 为 1，使能 LPUARTx 配置时钟及工作时钟。
- Step3. 配置 CR.CLKSRC，选择 LPUART 的传输时钟来源。
- Step4. 配置 CR.MODE，择择 LPUART 的工作模式。
- Step5. 配置 CR.OVER 及 BRR 寄存器，配置通信波特率。
- Step6. 配置 CR.STOP、CR.DR8CFG，配置 LPUART 的 STOP 长度，奇偶校验方式。
- Step7. 向 ICR.TC 写入 0x00，清除发送完成标志。
- Step8. 向 DR 寄存器写入一帧待发送的数据。
- Step9. 查询等待 ISR.TC 变为 1，本次写入的数据帧发送完成。
- Step10. 如待发送的数据未完成，则跳转到 Step7 继续执行。

### 17.4.2 接收数据示例

- Step1. 按 GPIO 章节管脚数字复用功能的相关描述，将 RXD 映射到需要的管脚；并配置 RXD 为输入并使能上拉电阻。
- Step2. 设置 Sysctrl\_PeriphClkEN0.LpUartx 为 1，使能 LPUART 配置时钟及工作时钟。
- Step3. 配置 CR.CLKSRC，选择 LPUART 的传输时钟来源。
- Step4. 配置 CR.MODE，择择 LPUART 的工作模式。
- Step5. 配置 CR.OVER 及 BRR 寄存器，配置通信波特率。
- Step6. 配置 CR.STOP、CR.DR8CFG，配置 LPUART 的 STOP 长度，奇偶校验方式。
- Step7. 向 ICR 寄存器写入 0x00，清除所有状态标志。
- Step8. 设置 CR.RXEN 为 1，使能 RXD 数据接收。
- Step9. 查询等待 ISR.RC 变为 1，接收完成一帧数据。
- Step10. 查询 ISR.PE 和 FE，以确认接收到的数据帧是否有效。若无效则进行出错处

理；若有效则从 DR 寄存器读出数据并保存。

Step11. 向 ICR.RC 写入 0x00，清除接收完成标志。

Step12. 如待接收的数据未完成，则跳转到 Step9 继续执行。

Preliminary  
to 立创商城

## 17.5 寄存器描述

LPUART0 基地址: 0x4000 0000

LPUART1 基地址: 0x4000 4000

寄存器	偏移地址	描述
LPUARTx_DR	0x00	数据寄存器
LPUARTx_CR	0x04	控制寄存器
LPUARTx_ADDR	0x08	地址寄存器
LPUARTx_AddrMask	0x0C	地址掩码寄存器
LPUARTx_ISR	0x10	中断标志位寄存器
LPUARTx_ICR	0x14	中断标志位清除寄存器
LPUARTx_BRR	0x18	波特率寄存器

### 17.5.1 数据寄存器 (LPUARTx\_DR)

偏移地址: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved								DR8	DR							
RW								RW	RW							

位	标记	功能描述
31:9	Reserved	保留
8	DR8	<p>在Mode0/1下, 读取该位为0, 写入该位无效;</p> <p>在Mode2/3下, 该位表示Bit8数据位, 分以下两种情况:</p> <ul style="list-style-type: none"> <li>(1) 当硬件奇偶校验位开启时, 接收时该位为接收数据奇偶校验位, 校验由硬件进行, 如校验出错, 校验错误标志位PE置1; 发送时该位无效, 发送数据奇偶校验位由硬件计算并发送;</li> <li>(2) 当硬件奇偶校验位关闭时, 接收时该位为接收数据Bit8; 发送时该位为发送数据Bit8;</li> </ul> <p>注意: 当开启多机通讯模式, 接收数据奇偶校验自动关闭; 发送数据奇偶校验仍受B8CFG控制;</p>
7:0	DR	发送数据时, 将待发送数据写入该寄存器 接收数据时, 从该寄存器读出已收到的数据

## 17.5.2 控制寄存器 (LPUARTx\_CR)

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								HDSEL	FEIE	CTSIE	CTSEN	RTSEN	Reserved		
								RW	RW	RW	RW	RW			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP	PEIE	CLKSRC	OVER	TXEIE	MODE	ADRDET	RXEN	DR8CFG	TCIE	RCIE					
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW					

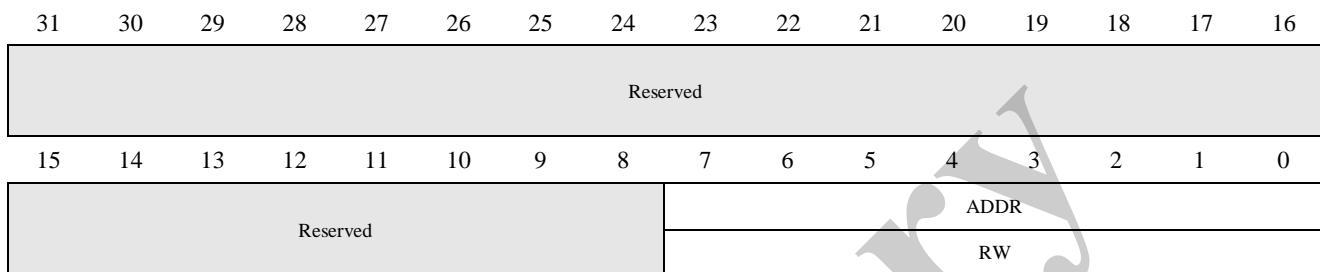
位	标记	功能描述
31:23	Reserved	保留位
22	HDSEL	单总线功能使能控制 0: 禁止单总线功能 1: 使能单总线功能
21	FEIE	帧错误中断使能控制 0: 禁止帧错误中断 1: 使能帧错误中断
20	CTSIE	CTS信号翻转中断使能控制 0: 禁止CTS信号翻转中断 1: 使能CTS信号翻转中断
19	CTSEN	CTS信号使能控制 0: 禁止CTS信号 1: 使能CTS信号
18	RTSEN	RTS信号使能控制 0: 禁止RTS信号 1: 使能RTS信号
17:16	Reserved	保留位
15:14	STOP	STOP位持续时间配置 00: 1比特 01: 1.5比特 10: 2比特 11: 保留 注意: Mode0时需设置STOP为00
13	PEIE	奇偶校验错误中断使能控制 1: 使能奇偶校验错误中断 0: 禁止奇偶校验错误中断
12:11	CLKSRC	传输时钟SCLK来源选择

		11: RCL 10: RCL 01: PCLK 00: PCLK																									
10:9	OVER	各模式下采样分频配置 <table border="1" data-bbox="414 406 1441 743"> <thead> <tr> <th>OVER</th><th>Mode0</th><th>Mode1</th><th>Mode2</th><th>Mode3</th></tr> </thead> <tbody> <tr> <td>11</td><td>Res.</td><td>Res.</td><td>Res.</td><td>Res.</td></tr> <tr> <td>10</td><td>Res.</td><td>4 倍采样</td><td>8 倍采样</td><td>4 倍采样</td></tr> <tr> <td>01</td><td>Res.</td><td>8 倍采样</td><td>16 倍采样</td><td>8 倍采样</td></tr> <tr> <td>00</td><td>Res.</td><td>16 倍采样</td><td>32 倍采样</td><td>16 倍采样</td></tr> </tbody> </table>	OVER	Mode0	Mode1	Mode2	Mode3	11	Res.	Res.	Res.	Res.	10	Res.	4 倍采样	8 倍采样	4 倍采样	01	Res.	8 倍采样	16 倍采样	8 倍采样	00	Res.	16 倍采样	32 倍采样	16 倍采样
OVER	Mode0	Mode1	Mode2	Mode3																							
11	Res.	Res.	Res.	Res.																							
10	Res.	4 倍采样	8 倍采样	4 倍采样																							
01	Res.	8 倍采样	16 倍采样	8 倍采样																							
00	Res.	16 倍采样	32 倍采样	16 倍采样																							
8	TXEIE	TX Buf 空中断使能控制 1: 使能TX Buf 空中断 0: 禁止TX Buf 空中断																									
7:6	MODE	工作模式配置 00: Mode0, 同步半双工, Data(8bit) 01: Mode1, 异步全双工, Start(1bit) + Data(8bit) + Stop (1~2bit) 10: Mode2, 异步全双工, Start (1bit) + Data(8bit) + B8(1bit) + Stop (1~2bit) 11: Mode3, 异步全双工, Start (1bit) + Data(8bit) + B8(1bit) + Stop (1~2bit)																									
5	ADRDET	多机通讯使能控制 0: 禁止, 发送与接收时B8功能由DR8CFG决定 1: 使能, 发送时B8固定为1, 接收时会进行地址匹配检测																									
4	RXEN	Mode0: 0: 发送; 1: 接收; Mode1/2/3: 0: 发送; 1: 接收/发送;																									
3:2	DR8CFG	DR8数据位功能配置 00: 由软件读写DR[8]来决定 01: 硬件偶校验 10: 硬件奇校验 11: 保留																									
1	TCIE	发送完成中断使能控制 1: 使能发送完成中断 0: 禁止发送完成中断																									
0	RCIE	接收完成中断使能控制 1: 使能接收完成中断 0: 禁止接收完成中断																									

### 17.5.3 地址寄存器 (LPUARTx\_ADDR)

偏移地址: 0x08

复位值: 0x0000 0000

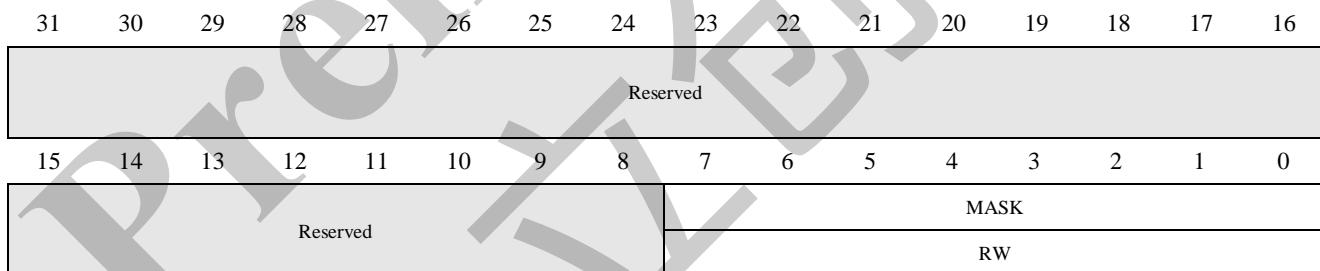


位	标记	功能描述
31:8	Reserved	保留
7:0	ADDR	从机设备地址

### 17.5.4 地址掩码寄存器 (LPUARTx\_AdrMask)

偏移地址: 0x0C

复位值: 0x0000 0000



位	标记	功能描述
31:8	Reserved	保留
7:0	MASK	从机设备地址掩码寄存器

### 17.5.5 标志位寄存器 (LPUARTx\_ISR)

偏移地址: 0x10

复位值: 0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CTS	CTSIF	PE	TXE	FE	TC	RC	
								RO	RO	RO	RO	RO	RO	RO	RO

位	符号	描述
31:7	Reserved	保留
6	CTS	CTS管脚电平 0: CTS管脚为低电平 1: CTS管脚为高电平
5	CTSIF	CTS管脚电平变化标志 0: CTS管脚电平未发生变化 1: CTS管脚电平已发生变化
4	PE	奇偶校验错误标志 0: 未发生奇偶校验错误 1: 已发生奇偶校验错误
3	TXE	Tx Buffer 空闲标志 0: Tx Buffer非空 1: Tx Buffer空闲
2	FE	帧错误标志 0: 未发生帧错误 1: 已发生帧错误
1	TC	TXD发送完成标志位 0: TXD正在发送数据 1: TXD发送完成
0	RC	接收完成标志 0: 未收到数据 1: 已收到一帧数据

### 17.5.6 标志位清除寄存器 (LPUARTx\_ICR)

偏移地址: 0x14

复位值: 0x0000 0037

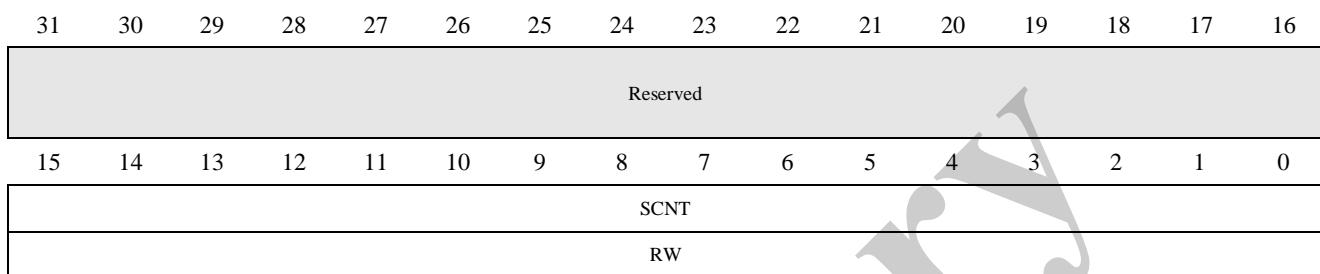
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										CTSIF	PE	Res.	FE	TC	RC
										R1W0	R1W0		R1W0	R1W0	R1W0

位	标记	功能描述
31:6	Reserved	保留
5	CTSIF	ISR.CTSIF标志清除控制 1: 无功能 0: 设置CTSIF标志为0
4	PE	ISR.PE标志清除控制 1: 无功能 0: 设置PE标志为0
3	Res.	保留
2	FE	ISR.FE标志清除控制 1: 无功能 0: 设置FE标志为0
1	TC	ISR.TC标志清除控制 1: 无功能 0: 设置TC标志为0
0	RC	ISR.RC标志清除控制 1: 无功能 0: 设置RC标志为0

### 17.5.7 波特率寄存器 (LPUARTx\_BRR)

偏移地址: 0x18

复位值: 0x0000 0000



位	标记	功能描述
31:16	Reserved	保留
15:0	SCNT	波特率计数器

## 18 串行外设接口 (SPI)

### 18.1 概述

串行外设接口 (SPI) 在主从模式下均提供了符合 SPI 通信协议的数据传输功能，可用于 MCU 和外部器件之间进行同步串行通信，本模块支持全双工、单线半双工及单工通信。

### 18.2 主要特性

- 可配置为主机或者从机，支持多机模式
- 主机模式最大分频系数为  $\text{PCLK}/2$ ，最高通信速率为 12M bps
- 从机模式最大分频系数为  $\text{PCLK}/4$ ，最高通信速率为 8M bps
- 多种通信模式：全双工、单线半双工、单工
- 两种传输顺序：先收发 MSB 或先收发 LSB
- 多种数据帧长度：4bit ~ 16bit
- 两种 NSS 方式：硬件控制、软件控制
- 可配置的串行时钟极性和相位

## 18.3 功能描述

### 18.3.1 功能框图

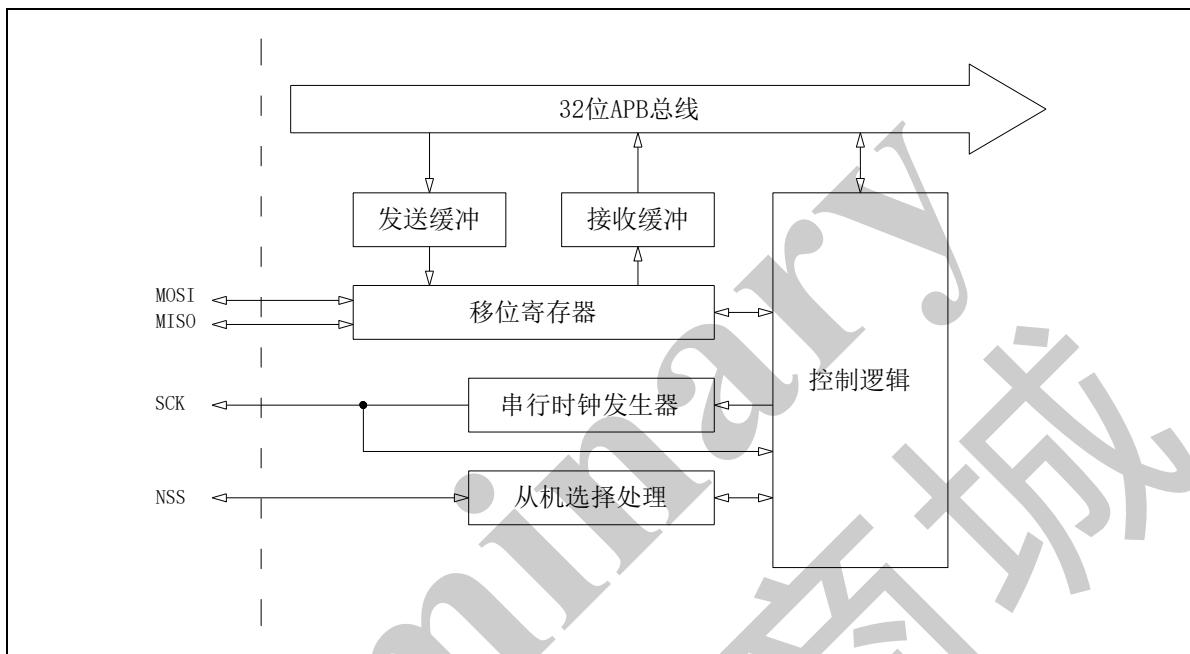


图 18-1 SPI 框图

### 18.3.2 通信格式和时序

SPI 串行帧格式取决于串行时钟极性 CPOL、串行时钟相位 CPHA 和数据帧格式。SPI 接口正常通信要求所有的主机和从机必须配置为相同的通信格式。

时钟极性 CPOL 是指 SPI 不传输数据的空闲状态下串行时钟的电平状态，当 SPI\_CR0.CPOL 为 0 时，串行时钟 SCK 在空闲时为低电平；当 SPI\_CR0.CPOL 为 1 时，串行时钟 SCK 在空闲时为高电平。

时钟相位 CPHA 是指 SPI 的主从机根据串行时钟 SCK 进行采样和移位的先后顺序，当 SPI\_CR0.CPHA 为 0 时，主从机在前边沿采样数据，后边沿移位数据；当 SPI\_CR0.CPHA 为 1 时，主从机在前边沿移位数据，后边沿采样数据。

通过设置 SPI\_CR0.WIDTH 可配置每帧数据的宽度为 4bit 到 16bit。

通过设置 SPI\_CR0.LSBF 可配置每帧数据的传输顺序为先传输 LSB 或先传 MSB。

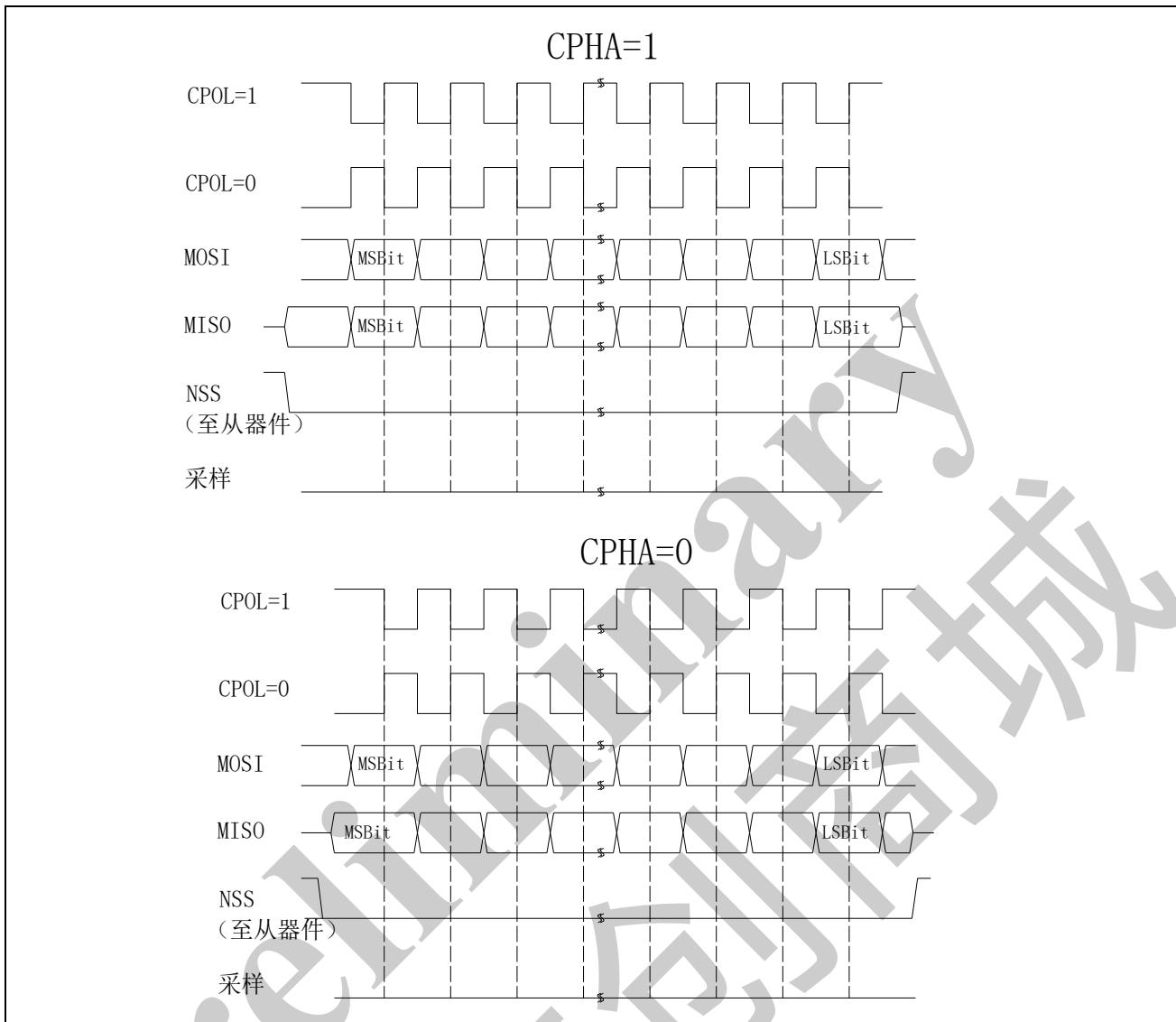


图 18-2 SPI 通信格式时序图

### 18.3.3 从机选择管脚配置

本模块提供了多种从机选择方式以适配不同的应用场景。

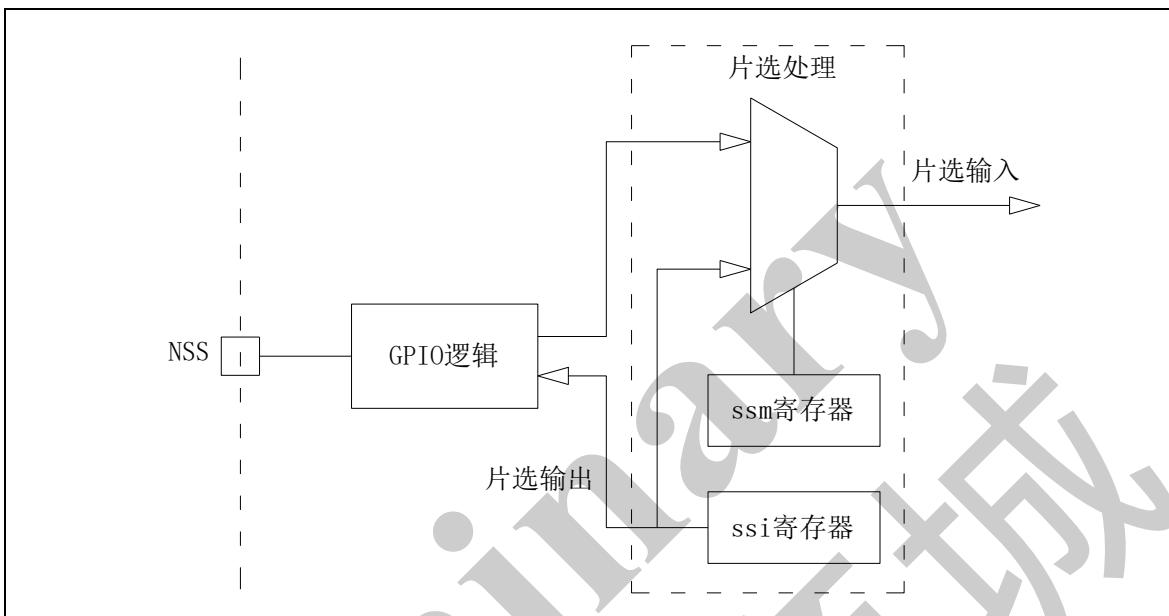


图 18-3 从机选择 NSS 管脚配置

当 SPI 工作于主机模式且 SPI\_CR0.SSM 为 0 时，从机选择 NSS 是输入，其状态由 NSS 管脚的电平决定。

当 SPI 工作于主机模式且 SPI\_CR0.SSM 为 1 时，从机选择 NSS 是输出，NSS 管脚的输出电平由内部寄存器 SPI\_SSI.SSI 决定。当 SPI\_SSI.SSI 为 0 时，NSS 管脚输出低电平；当 SPI\_SSI.SSI 为 1 时，NSS 管脚输出高电平。

当 SPI 工作于从机模式且 SPI\_CR0.SSM 为 0 时，NSS 管脚的低电平可选中本模块。

当 SPI 工作于从机模式且 SPI\_CR0.SSM 为 1 时，NSS 管脚不再具有片选功能，寄存器 SPI\_SSI.SSI 为 0 时可选中本模块。

#### 18.3.4 全双工通信

全双工通信的方式，允许主从机依托两根单向的数据线进行同时收发的双向的通信。该模式下，主机通过 SCK 脚提供时钟信号来同步数据的输入与输出。主从机 MOSI 脚相互连接，MISO 脚相互连接。主机通过 MOSI 脚把数据发送给从机，从机通过 MISO 管脚回传数据。每当用户向 SPI\_DR 寄存器写入数据时，SPI 主机会通过 SCK、MOSI 将数据发送到 SPI 从机；并通过 SCK、MISO 接收来自从机的数据。

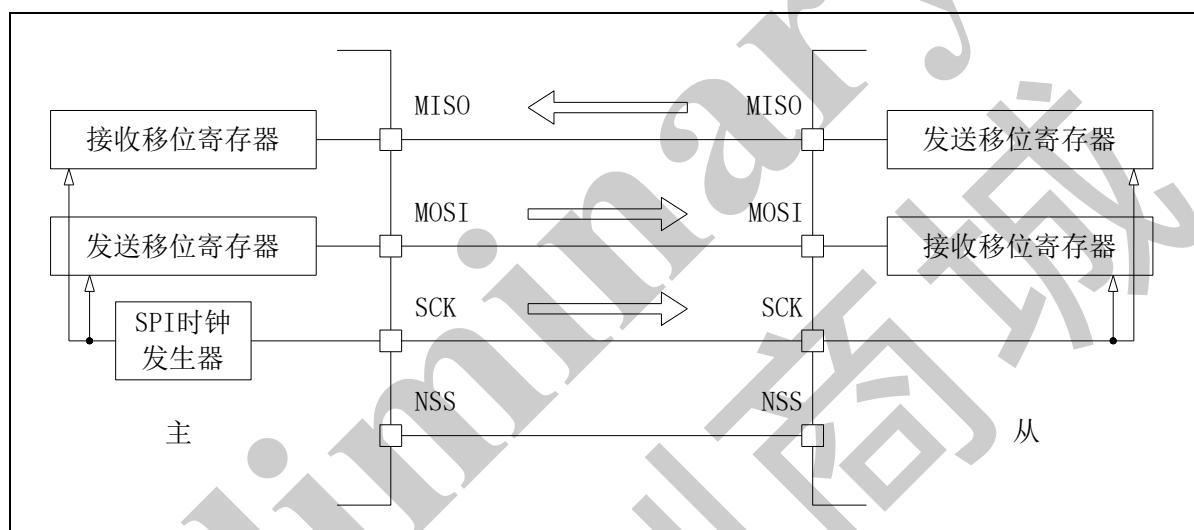


图 18-4 SPI 全双工通信

to  
by  
Preliminary Draft

### 18.3.5 单线半双工通信

单线半双工通信的方式，允许主从机依托一根双向的数据线进行分时收发的双向的通信。作为主机时使用的管脚是 MOSI，作为从机时使用的管脚是 MISO。数据传输方向由 HDOE.HDOE 决定，当 HDOE.HDOE 为 1 时，数据线进行发送。未用到的管脚不参与 SPI 通信，可供其他功能使用。当主机工作于接收状态时，需要向 DR 寄存器写入任意数据以触发 SPI 主机发送 SCK 进行数据传输。

当主从机之间进行单线半双工通信时，由于收发的切换可能不同步，它们对于双向的共用数据线可能出现驱动冲突。为避免此类情况对器件可能产生的损害，建议在数据线上串入限流电阻以保护芯片管脚。

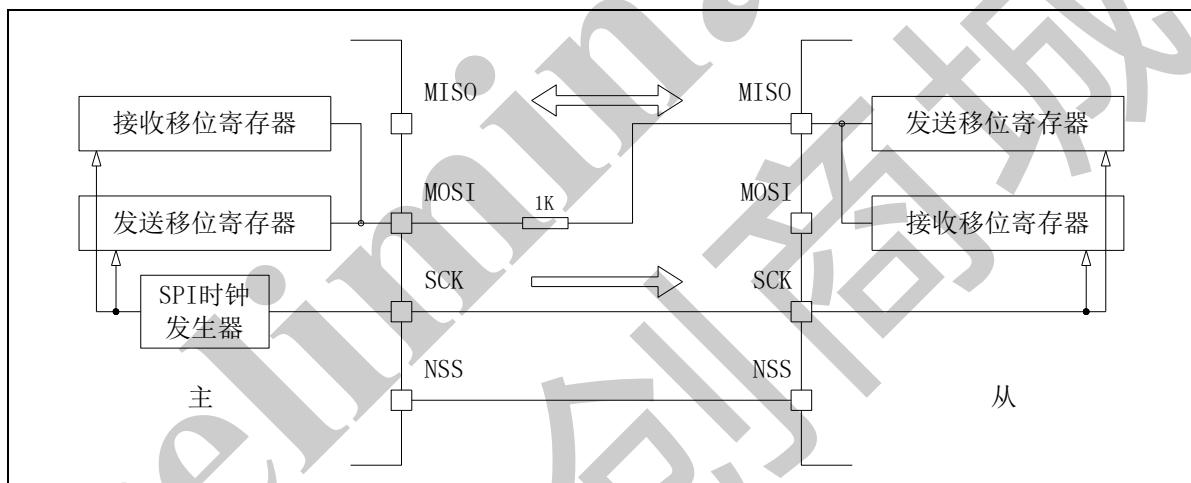


图 18-5 SPI 单线半双工通信

### 18.3.6 单工通信

单工通信的方式，允许主从机依托一根单向的数据线进行固定接收或固定发送的单向的通信，通过设置 SPI\_CR0.CM 选择单工只发或单工只收模式。未用到的管脚不参与 SPI 通信，可供其他功能使用。当主机工作于接收状态时，需要向 DR 寄存器写入任意数据以触发 SPI 主机发送 SCK 进行数据传输。

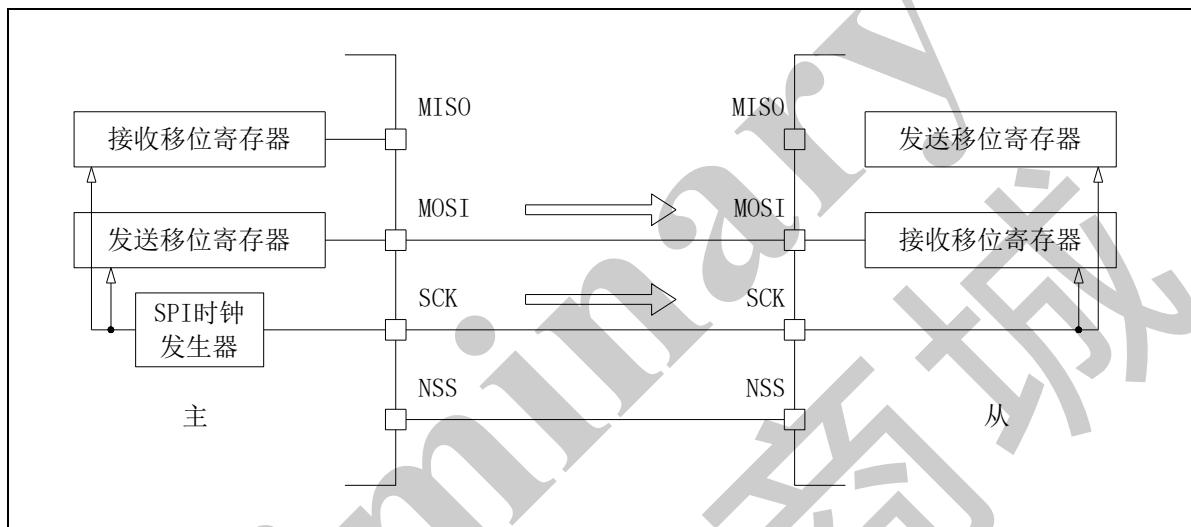


图 18-6 SPI 单工发送通信示意

注意：

- NSS 管脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些管脚。之后，必须在内部为主器件和从器件处理硬件控制流。
- 该配置下，两个 MISO 管脚均可用作 GPIO。

### 18.3.7 多机通信

多机通信的系统中, 主机的从机选择 NSS 管脚配置为硬件输入 (SPI\_CR0.SSM 为 0), 由 GPIO 负责输出从机选择信号, 且主机的 NSS 输入都由另一台主机的 GPIO 驱动。主机在尝试获取 SPI 总线控制权时, 能结合模式错误 MODF 来检测总线是否发生冲突。多个从机的系统中, 每台从机共享串行时钟和数据线, 但是需要各自独立的从机选择信号, 这就要求主机通过不同的 GPIO 向每个从机输出对应的从机选择信号。

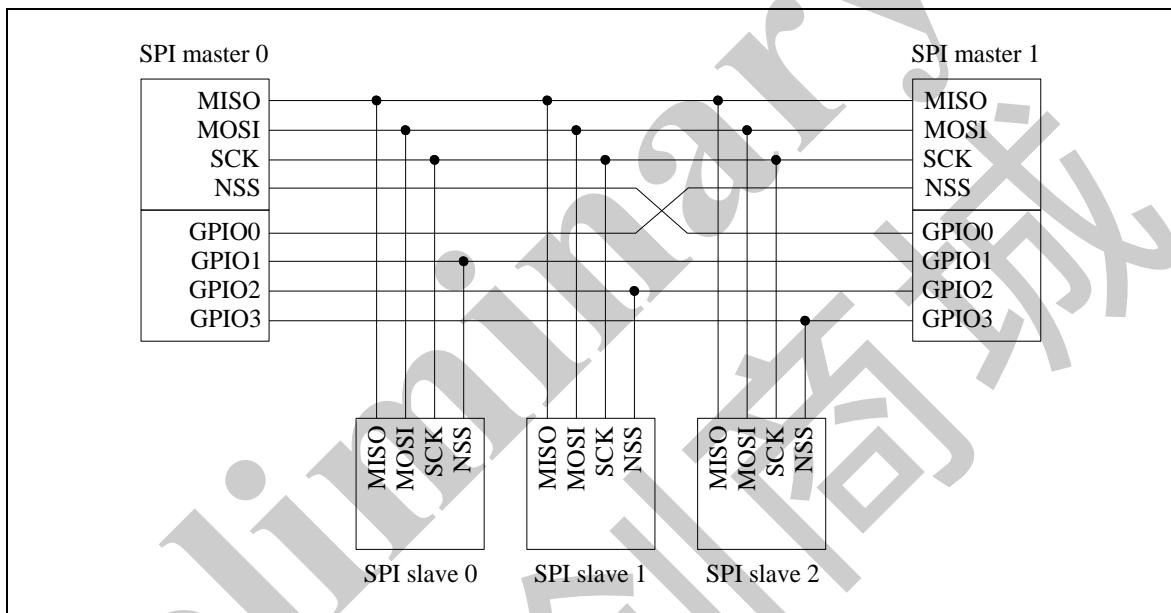


图 18-7 SPI 多机系统

### 18.3.8 状态标志和错误标志

SPI 提供多种状态标志位来指示当前的工作状态，包括一般状态标志和错误标志。

#### 从机选择输入 (NSS) 状态

从机选择输入状态指示当前输入的从机选择信号的电平，对应寄存器 SPI\_SR.SSLVL。

当 SPI\_SR.SSLVL 为 0 时，从机选择输入是低电平；当 SPI\_SR.SSLVL 为 1 时，从机选择输入是高电平。

#### 总线忙标志位

总线忙标志位指示当前 SPI 接口处于忙碌还是空闲状态，对应寄存器 SPI\_SR.BUSY。

当 SPI\_SR.BUSY 为 0 时，SPI 接口处于空闲状态，当前没有数据传输；当 SPI\_SR.BUSY 为 1 时，SPI 接口处于忙碌状态，当前正在准备进行或正在进行数据传输。

#### 从机选择输入上升沿标志位

从机选择输入上升沿标志位指示当前输入的从机选择信号是否出现了上升沿，对应寄存器 SPI\_SR.SSR。当 SPI\_SR.SSR 为 0 时，从机选择输入未出现上升沿；当 SPI\_SR.SSR 为 1 时，从机选择输入出现了上升沿。

#### 从机选择输入下降沿标志位

从机选择输入下降沿标志位指示当前输入的从机选择信号是否出现了下降沿，对应寄存器 SPI\_SR.SSF。当 SPI\_SR.SSF 为 0 时，从机选择输入未出现下降沿；当 SPI\_SR.SSF 为 1 时，从机选择输入出现了下降沿。

#### 发送缓冲空标志位

发送缓冲空标志位指示当前 SPI 的发送缓冲是否已装载待发送数据，对应寄存器 SPI\_SR.TXE。当 SPI\_SR.TXE 为 0 时，SPI 的发送缓冲已装载待发送数据；当 SPI\_SR.TXE 为 1 时，SPI 的发送缓冲处于空状态，可以将下一个待发送的数据写入。

#### 接收缓冲非空标志位

接收缓冲非空标志位指示当前 SPI 的接收缓冲是否存在已接收的数据，对应寄存器 SPI\_SR.RXNE。当 SPI\_SR.RXNE 为 0 时，SPI 的接收缓冲尚未装载有效的接收数据；当 SPI\_SR.RXNE 为 1 时，SPI 的接收缓冲装载了已接收的数据，可供读出。

#### 从机模式下的从机选择错误标志位

当 SPI 工作于从机模式且正在进行有效的数据传输时，输入的从机选择信号要保持选

中状态（低电平）来保证正常的通信，如果从机选择在此期间被拉高，就会发生从机选择错误，对应的寄存器 SPIx\_SR.SSERR 会置位。

发生从机选择错误，会造成对应的数据传输出错，SPI 接口也会因从机选择的拉高而进入未选中状态。从机选择错误发生后，对应的错误标志位置位并不阻止随后进行正常的数据传输，但是错误标志位会保持置位状态直到用户程序将其清除或 SPI 被系统控制模块复位。

需要注意的是，从机选择错误 SSERR 是只出现于从机模式下的错误，且仅当有效数据尚在传输时从机选择被拉高才会触发，帧与帧之间没有有效数据传输时正常的从机选择拉高并不触发该错误。

### 接收缓冲上溢错误标志位

SPI 的接收缓冲器的大小为一帧，当接收缓冲中的数据没有被读出且新一帧的数据已经接收完成时，发生的就是接收缓冲上溢错误，对应的寄存器 SPIx\_SR.OVF 会置位。发生接收缓冲上溢错误，缓冲器中原有数据会被新接收的数据覆盖，造成数据的丢失。接收缓冲上溢错误发生后，对应的错误标志位置位并不阻止随后进行正常的数据传输，但是错误标志位会保持置位状态直到用户程序将其清除或 SPI 被系统控制模块复位。

### 从机模式下的发送缓冲下溢错误标志位

SPI 的发送缓冲器的大小为一帧，当从机模式下发送缓冲已空且新一帧的传输开始时，发生的就是发送缓冲下溢错误，对应的寄存器 SPIx\_SR.UDF 会置位。

发生从机模式下发送缓冲下溢错误，意味着发送缓冲器中没有及时填入数据，错过了给主机发送数据。从机模式下缓冲下溢错误发生后，对应的错误标志位置位并不阻止随后进行正常的数据传输，但是错误标志位会保持置位状态直到用户程序将其清除或 SPI 被系统控制模块复位。

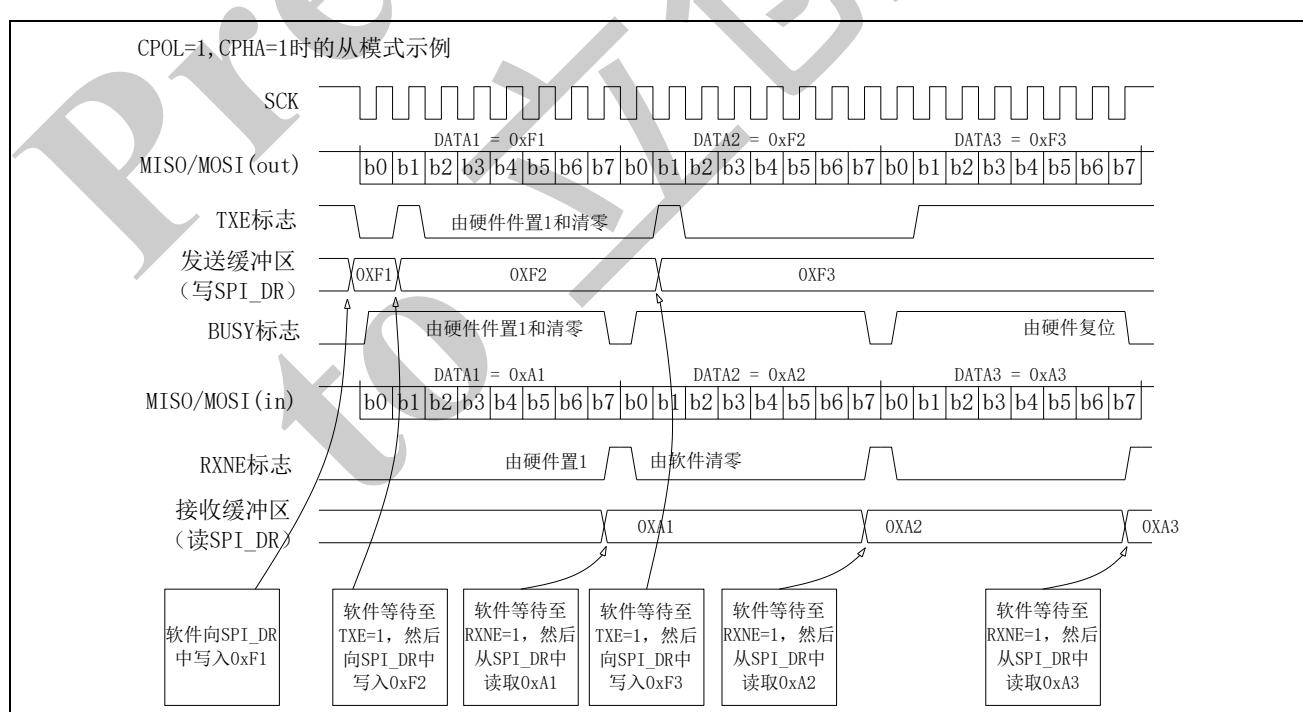
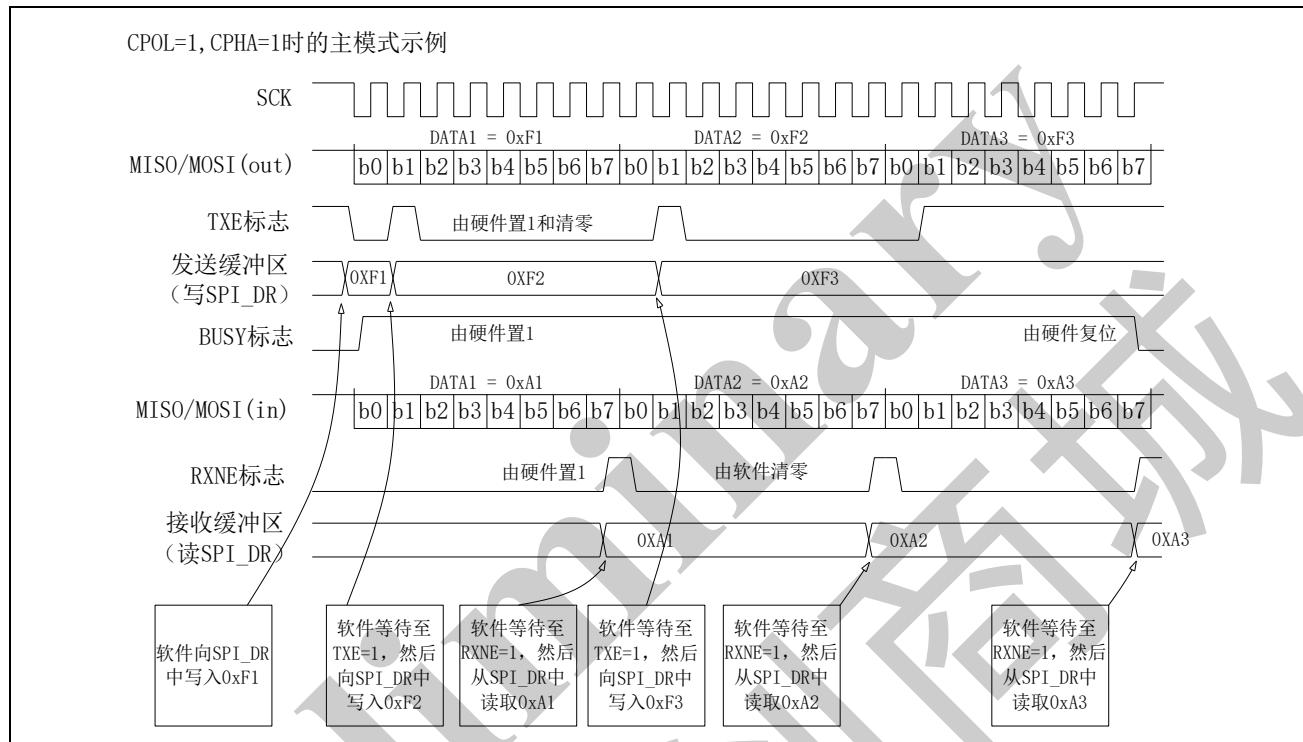
### 主机模式下的模式错误标志位

当 SPI 工作于主机模式且从机选择管脚被配置为输入时，输入的从机选择信号要一直保持非选中状态（高电平）来保证正常的通信，如果从机选择被拉低，就会发生模式错误，对应的寄存器 SPIx\_SR.MODF 会置位。

发生模式错误，会造成 SPI 强制返回未使能状态（即 SPIx\_CR0.EN 被清零），主从模式选择寄存器 (SPIx\_CR0.MSTR) 也会被清零（即切换到从机模式）。模式错误发生后，

对应的错误标志位置位并不阻止用户程序重新使能 SPI 并进行正常的数据传输，但是错误标志位会保持置位状态直到用户程序将其清除或 SPI 被系统控制模块复位。

需要注意的是，模式错误 MODF 是只出现于主机模式特定配置下的错误，仅当从机选择被配置为输入且被拉低才会触发，其他情况不会触发该错误。



### 18.3.9 中断的产生和清除

SPI 支持从不同的事件产生中断请求，具体信息如下：

中断事件	对应标志位 ( <b>SPI_SR</b> )	对应中断使能 ( <b>SPI_CR1</b> )	清除中断标志的方法
从机选择输入上升沿	SSR	SSRIE	向 SPI_ICR.SSRC 写 0
从机选择输入下降沿	SSF	SSFIE	向 SPI_ICR.SSFC 写 0
发送缓冲空	TXE	TXEIE	向数据寄存器 DR 中写数据
接收缓冲非空	RXNE	RXNEIE	从数据寄存器 DR 中读数据 或 向 SPI_ICR.RXNEC 写 0
从机模式下从机选择错误	SSERR	SSERRIE	向 SPI_ICR.SSERRC 写 0
接收缓冲上溢错误	OVF	OVFIE	向 SPI_ICR.OVFC 写 0
从机模式发送缓冲下溢错误	UDF	UDFIE	向 SPI_ICR.UDFC 写 0
主机模式下模式错误	MODF	MODFIE	向 SPI_ICR.MODFC 写 0

表 18-1 SPI 中断

如上表，每种中断事件都有独立的中断使能控制，其中由错误事件产生的中断也都有独立的中断清除寄存器。对于上述的中断，无论中断是否使能，都可以使用表中所述的方法来清除中断标志。

## 18.4 编程示例

### 18.4.1 主机全双工收发数据示例

- Step1. 按 GPIO 章节管脚数字复用功能的相关描述，将 NSS、SCK、MOSI、MISO 映射到需要的管脚。
- Step2. 设置 Sysctrl\_PerClkEN0.SPI 为 1，使能 SPI 配置时钟及工作时钟。
- Step3. 设置 CR0.MSTR 为 1，选择主机模式。
- Step4. 设置 CR0.BRR，配置通信波特率。
- Step5. 设置 CR0.CM 为 0，选择全双工通信方式。
- Step6. 设置 CR0.SSM，配置从机选择模式。
- Step7. 设置 CR0.CPOL、CR0.CPHA，选择串行时钟极性与相位。
- Step8. 设置 CR0.WIDTH，配置每帧的数据宽度。
- Step9. 设置 CR0.LSBF，配置收发数据时的数据序列顺序。
- Step10. 设置 CR0.EN 为 1，使能 SPI 模块。
- Step11. 设置 SSI.SSI 为 0，选中对应的从机。
- Step12. 将待发送数据写入 DR 寄存器，SPI 自动开始数据传输。
- Step13. 查询等待 SR.RXNE 变为 1，完成一帧数据的发送与接收。
- Step14. 从 DR 寄存器中读出接收到的数据并保存。
- Step15. 若待发送与接收的数据未完成，则跳转到 Step12 继续执行。

### 18.4.2 从机全双工收发数据示例

- Step1. 按 GPIO 章节管脚数字复用功能的相关描述, 将 MOSI 及 MISO 映射到需要的管脚。
- Step2. 设置 Sysctrl\_PerClkEN0.SPI 为 1, 使能 SPI 配置时钟及工作时钟。
- Step3. 设置 CR0.MSTR 为 0, 选择从机模式。
- Step4. 设置 CR0.CM 为 0, 选择全双工通信方式。
- Step5. 设置 CR0.SSM, 配置从机选择模式。
- Step6. 设置 CR0.CPOL、CR0.CPHA, 选择串行时钟极性与相位。
- Step7. 设置 CR0.WIDTH, 配置每帧的数据宽度。
- Step8. 设置 CR0.LSBF, 配置收发数据时的数据序列顺序。
- Step9. 设置 CR0.EN 为 1, 使能 SPI 模块。
- Step10. 设置 ICR.TXES 为 0, 清空发送缓冲区。
- Step11. 查询等待 SR.SSLVL 变为 0, SPI 主机选中本模块。
- Step12. 将待发送的数据写入 DR 寄存器, 准备好待发送的数据。
- Step13. 查询等待 SR.RXNE 变为 1, 完成一帧数据的发送与接收。
- Step14. 从 DR 寄存器中读出接收到的数据并保存。
- Step15. 若待发送与接收的数据未完成, 则跳转到 Step12 继续执行。
- Step16. 等待 SR.SSLVL 变为 1, 完成本次通信。

### 18.4.3 单线半双工发送接收数据示例

- Step1. 按 GPIO 章节管脚数字复用功能的相关描述, 将主机的 NSS、SCK、MOSI 和从机的 NSS、SCK、MISO 映射到需要的管脚。
- Step2. 设置 Sysctrl\_PerClkEN0.SPI 为 1, 使能 SPI 配置时钟及工作时钟。
- Step3. 主机设置 CR0.MSTR 为 1, 从机设置 CR0.MSTR 为 0。
- Step4. 主机设置 CR0.BRR, 配置通信波特率。
- Step5. 设置 CR0.CM 为 11, 选择单线半双工通信方式。
- Step6. 配置 CR0.SSM, 配置从机选择模式。
- Step7. 配置 CR0.CPOL、CR0.CPHA, 选择串行时钟极性与相位。
- Step8. 配置 CR0.WIDTH, 配置每帧的数据宽度。
- Step9. 配置 CR0.LSBF, 配置发送数据时的数据序列顺序。
- Step10. 设置 CR0.EN 为 1, 使能 SPI 模块。
- Step11. 主机设置 SSI.SSI 为 0, 选中对应的从机;  
从机等待 SR.SSLVL 变为 0 选中本模块。
- Step12. 主机设置 HDOE.HDOE 为 1, 切换至发送模式;  
从机设置 HDOE.HDOE 为 0, 切换至接收模式。
- Step13. 主机将待发送数据写入 DR 寄存器, SPI 自动开始数据传输。
- Step14. 主机查询等待 SR.BUSY 变为 0, 完成一帧数据的发送;  
从机查询等待 SR.RXNE 变为 1, 完成一帧数据的接收。
- Step15. 从机从 DR 寄存器中读出接收到的数据并保存。
- Step16. 若主机待发送的数据未完成, 则跳转到 Step13 继续执行。
- Step17. 主机设置 HDOE.HDOE 为 0, 切换至接收模式;  
从机设置 HDOE.HDOE 为 1, 切换至发送模式。
- Step18. 从机查询等待 SR.TXE 变为 1, 发送缓存空闲。
- Step19. 若从机待发送的数据未完成, 则向 DR 寄存器写入一帧数据。
- Step20. 主机向 DR 寄存器写入任意数据, SPI 主机发送 SCK 以进行数据传输。
- Step21. 主机查询等待 SR.RXNE 变为 1, 完成一帧数据的接收。
- Step22. 主机从 DR 寄存器中读出接收到的数据并保存。
- Step23. 若从机待发送的数据未完成, 则跳转到 Step18 继续执行。
- Step24. 从机等待 SR.SSLVL 变为 1, 设置 HDOE.HDOE 为 0, 完成本次通信。

## 18.6 寄存器描述

### 寄存器列表

SPI 基地址: 0x40000800

偏移量	寄存器名称	访问	寄存器描述
0x00	SPI_CR0	RW	SPI 控制寄存器 0
0x04	SPI_CR1	RW	SPI 控制寄存器 1
0x08	SPI_HDOE	RW	SPI 单线半双工输出使能寄存器
0x0C	SPI_SSI	RW	SPI 内部从机选择寄存器
0x10	SPI_SR	RO	SPI 状态寄存器
0x14	SPI_ICR	WO	SPI 中断清除寄存器
0x18	SPI_DR	RW	SPI 数据寄存器

表 18-2 SPI 寄存器列表

### 18.6.1 SPI 控制寄存器 0(SPI\_CR0)

地址偏移量: 0x00

复位值: 0x000000700

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CM	SSM	WIDTH	LSBF	MSTR	CPOL	CPHA	BRR					EN		
	RW	RW	RW	RW	RW	RW	RW	RW					RW		RW

位	标记	功能描述
31:15	Reserved	
14:13	CM	通信方式配置 00: 全双工双向 01: 单工只发 10: 单工只收 11: 单线半双工
12	SSM	从机选择配置 (主机模式) 1: SSI 寄存器决定从机选择输出值 0: NSS 管脚决定从机选择输入值 从机选择配置 (从机模式) 1: SSI 寄存器决定从机选择输入值 0: NSS 管脚决定从机选择输入值
11:8	WIDTH	帧数据宽度配置 0011: 4bit 0100: 5bit ... 1110: 15bit 1111: 16bit

7	LSBF	数据帧高低位顺序选择 1: 最低有效位 LSB 收发在前 0: 最高有效位 MSB 收发在前
6	MSTR	工作模式配置 1: 主机模式 0: 从机模式
5	CPOL	串行时钟极性配置 1: 待机时高电平 0: 待机时低电平
4	CPHA	串行时钟相位配置 1: 前边沿移位/后边沿采样 0: 前边沿采样/后边沿移位
3:1	BRR	主机模式波特率配置 000: PCLK/2 001: PCLK/4 010: PCLK/8 011: PCLK/16 ... 110: PCLK/128 111: 保留
0	EN	使能控制 1: 使能 SPI 模块 0: 禁止 SPI 模块

## 18.6.2 SPI 控制寄存器 1(SPI\_CR1)

地址偏移量: 0x04

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								MODFIE	SSERRIE	OVFIE	UDFIE	SSRIE	SSFIE	RXNEIE	TXEIE
								RW	RW	RW	RW	RW	RW	RW	RW

位	标记	功能描述
31:8	Reserved	
7	MODFIE	模式错误中断使能控制 1: 使能 0: 禁用
6	SSERRIE	从机模式下的从机选择错误中断使能控制 1: 使能 0: 禁用
5	OVFIE	接收缓冲上溢错误中断使能控制 1: 使能 0: 禁用
4	UDFIE	从机模式下发送缓冲下溢错误中断使能控制 1: 使能 0: 禁用
3	SSRIE	从机选择输入上升沿中断使能控制 1: 使能 0: 禁用
2	SSFIE	从机选择输入下降沿中断使能控制 1: 使能

		0: 禁用
1	RXNEIE	接收缓冲非空中断使能控制 1: 使能 0: 禁用
0	TXEIE	发送缓冲空中断使能控制 1: 使能 0: 禁用

Preliminary  
to 立创商城

### 18.6.3 SPI 单线半双工输出使能寄存器(SPI\_HDOE)

地址偏移量: 0x08

复位值: 0x00000000

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

位	标记	功能描述
31:1	Reserved	保留位
0	HDOE	MOSI/MISO 收发切换控制 0: 只收 1: 只发 注: 仅在单线半双工通信时有效

#### 18.6.4 SPI 内部从机选择寄存器(SPI\_SSI)

地址偏移量: 0x0C

复位值: 0x00000001

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

位	标记	功能描述
31:1	Reserved	保留位
0	SSI	内部从机选择, 当 SPI_CR0.SSM 寄存器为 1 时有效 (主机模式下寄存器决定从机选择输出值/从机模式下寄存器决定从机选择输入值)

### 18.6.5 SPI 状态寄存器(SPI\_SR)

地址偏移量: 0x10

复位值: 0x00000001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					SSLVL	BUSY	MODF	SSERR	OVF	UDF	SSR	SSF	RXNE	TXE	
					RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	

位	标记	功能描述
31:10	Reserved	
9	SSLVL	从机选择输入的状态 1: 从机选择输入是高电平 0: 从机选择输入是低电平 注: 当主机模式且 SPI_CR0.SSM=1 时, 本 bit 无意义。
8	BUSY	总线忙标志位 1: 总线忙 0: 总线空闲
7	MODF	模式错误标志位 (主机模式下从机选择输入为低则触发 MODF) 1: 出错 0: 正常
6	SSERR	从机模式下的从机选择错误标志位 1: 出错 0: 正常
5	OVF	接收缓冲上溢错误标志位 1: 出错 0: 正常
4	UDF	从机模式下的发送缓冲下溢错误标志位

		1: 出错 0: 正常
3	SSR	从机选择输入上升沿标志位 1: 出现了上升沿 0: 未出现上升沿 注: 当主机模式且 SPI_CR0.SSM=1 时, 本 bit 无意义。
2	SSF	从机选择输入下降沿标志位 1: 出现了下降沿 0: 未出现下降沿 注: 当主机模式且 SPI_CR0.SSM=1 时, 本 bit 无意义。
1	RXNE	接收缓冲非空标志位 1: 接收缓冲非空 0: 接收缓冲空
0	TXE	发送缓冲空标志位 1: 发送缓冲空 0: 发送缓冲非空

### 18.6.6 SPI 中断清除寄存器(SPI\_ICR)

地址偏移量: 0x14

复位值: 0x000000FF

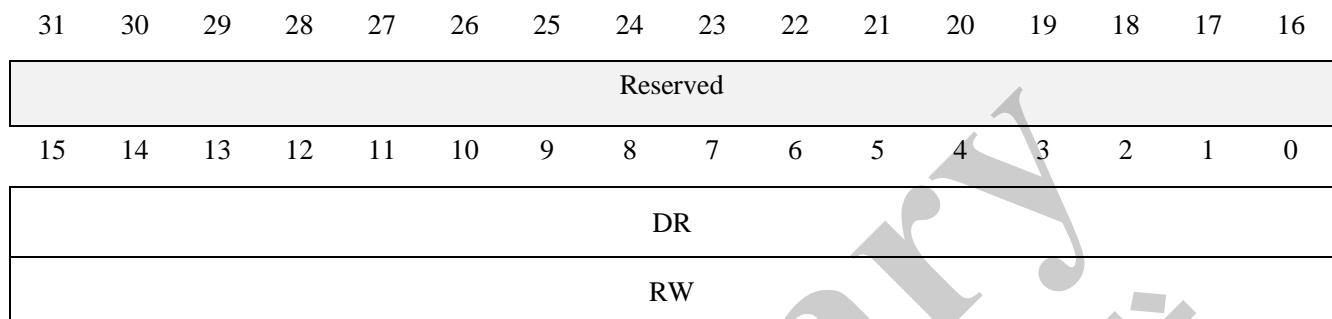
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Reserved																	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reserved										MODF	SSERR	OVF	UDF	SSR	SSF	RXNE	TXE
										R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	R1W0	

位	标记	功能描述
31:8	Reserved	保留位
7	MODF	写 0 清除 ISR.MODF 标志, 读出为 1。
6	SSERR	写 0 清除 ISR.SSERR 标志, 读出为 1。
5	OVF	写 0 清除 ISR.OVF 标志, 读出为 1。
4	UDF	写 0 清除 ISR.UDF 标志, 读出为 1。
3	SSR	写 0 清除 ISR.SSR 标志, 读出为 1。
2	SSF	写 0 清除 ISR.SSF 标志, 读出为 1。
1	RXNE	写 0 清除 ISR.RXNE 标志, 读出为 1。
0	TXE	写 0 设置 ISR.TXE 标志为 1, 读出为 1。

### 18.6.7 SPI 数据寄存器(SPI\_DR)

地址偏移量: 0x18

复位值: 0x0000 0000



位	标记	功能描述
31:16	Reserved	
15:0	DR	发送数据时, 将待发送数据写入该寄存器 接收数据时, 从该寄存器读出已收到的数据

## 19 I2C 总线 (I2C)

### 19.1 概述

I2C 是双线双向的同步串行总线，它利用一根时钟线和一根数据线在连接总线的两个器件之间进行信息的传递，为设备之间数据交换提供了一种简单高效的方法。每个连接到总线上的器件都有唯一的地址，任何器件既可以作为主机也可以作为从机，但同一时刻只允许有一个主机。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，它能在多个主机同时请求控制总线时利用仲裁机制避免数据冲突并保护数据。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 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。

### 19.2 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
- 从机支持 7 位寻址功能
- 主机支持 10 位寻址功能
- 支持噪声过滤功能
- 支持三个从机地址
- 支持广播地址
- 支持中断状态查询功能

### 19.3 协议描述

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。主机在 SCL 线上输出串行时钟信号，数据在 SDA 线上进行传输，每传输一个字节（最高位 MSB 开始传输），后面跟随一个应答位。一个 SCL 时钟脉冲传输一个数据位。

### 19.3.1 I2C 总线上数据传输

通常标准 I2C 传输协议包含四个部分：起始(S)或重复起始信号(Sr)，从机地址及读写位，传输数据，停止信号(P)。

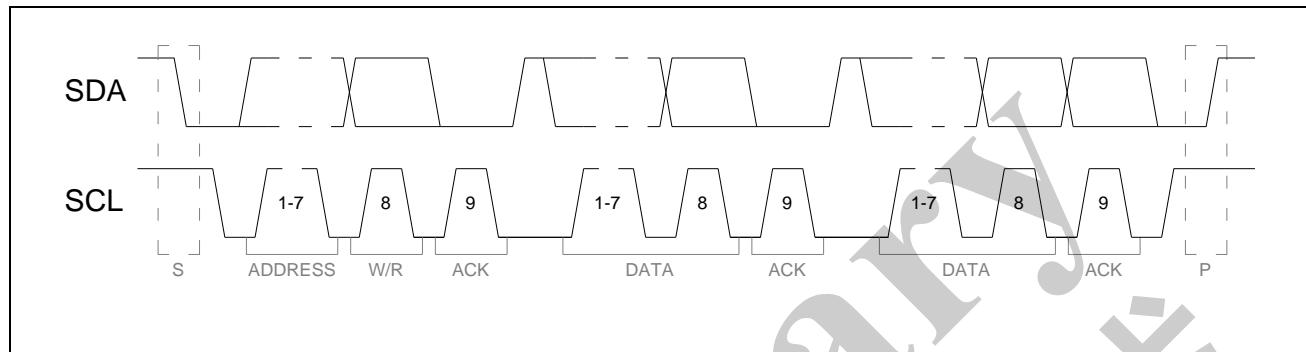


图 19-1 I2C 传输协议

- 起始信号、重复起始信号、停止信号

当总线处于空闲状态下(SCL 和 SDA 线同时为高)，SDA 线上出现由高到低的信号，表明总线上产生了起始信号。

当两个起始信号之间没有停止信号时，即产生了重复起始信号。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。

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。主机向总线发出停止信号结束数据传送。

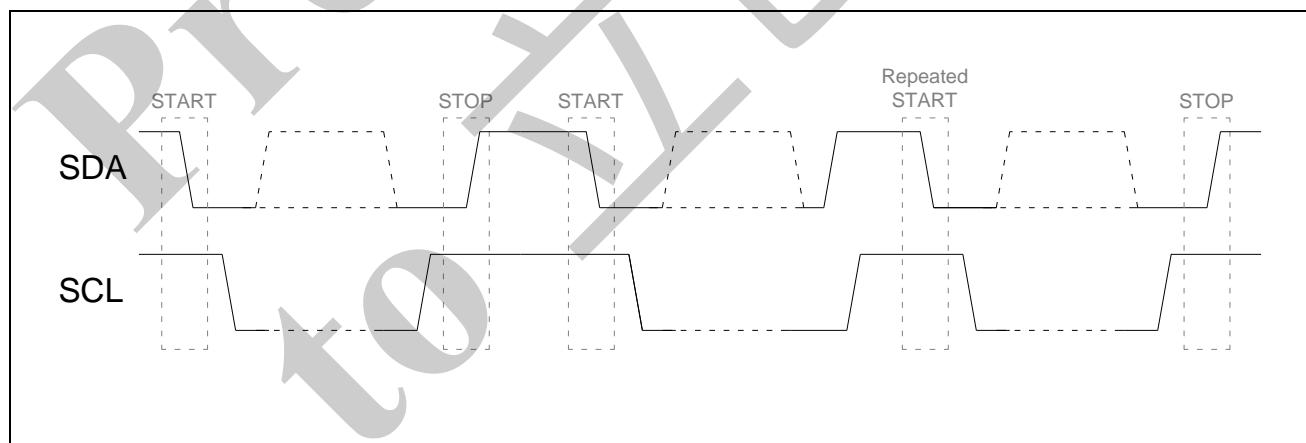


图 19-2 START 和 STOP 条件

- 从机地址及读写位

当起始信号产生后，主机立即传输数据的第一字节：7 位从机地址 + 读写位，读写位控制从机的数据传输方向 (0: 写; 1: 读)。被主机寻址的从机会通过在第 9 个 SCL

时钟周期将 SDA 置为低电平作为应答。

- 传输数据

数据传输过程中，一个 SCL 时钟脉冲传输一个数据位，且 SDA 线只有在 SCL 为低时才可以改变。

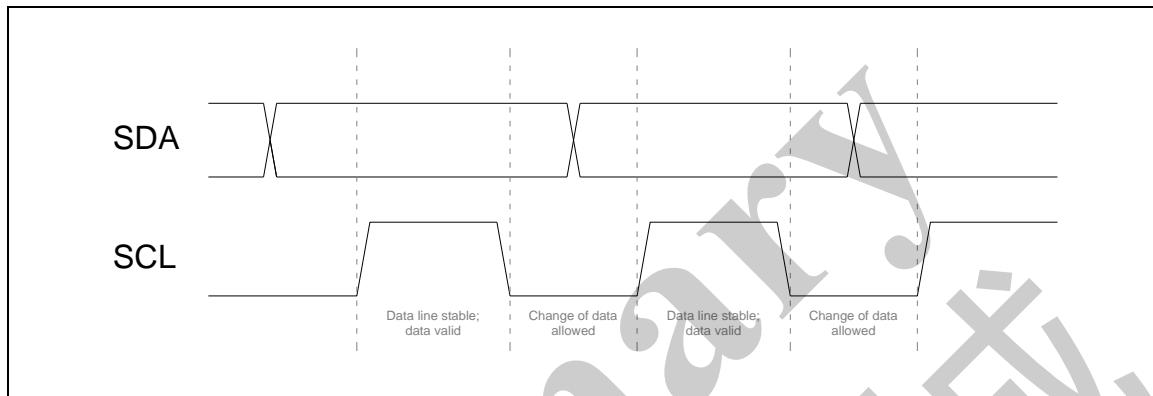


图 19-3 I2C 总线上位传输

### 19.3.2 I2C 总线上的应答

每传输一个字节，后面跟随一个应答位。通过将 SDA 线拉低，来允许接收端回应发送端。ACK 为一个低电平信号，当时钟信号为高时，SDA 保持低电平则表明接收端已成功接收到发送端的数据。

当主机作为发送器件时，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。

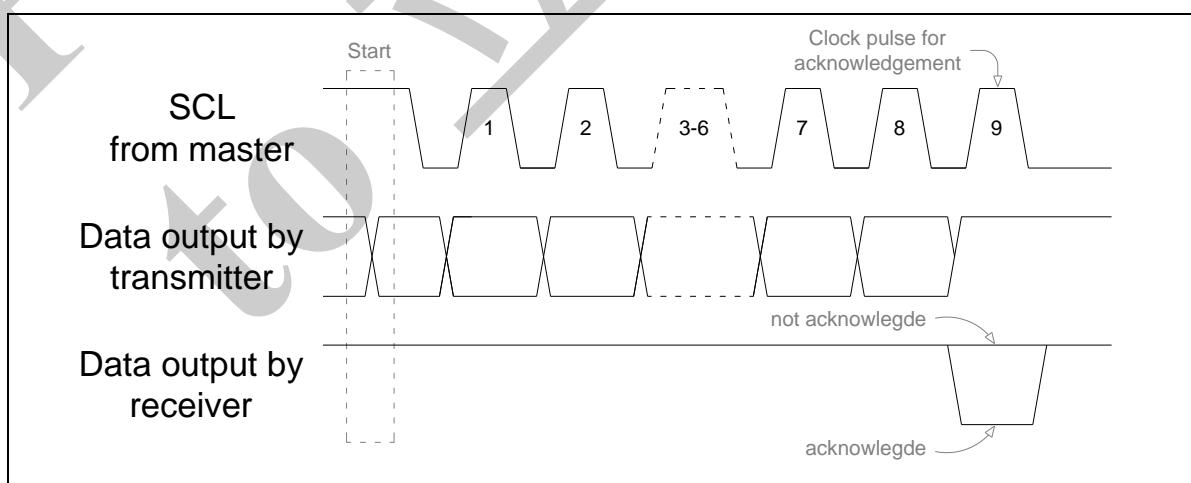


图 19-4 I2C 总线上应答信号

### 19.3.3 I2C 总线上的仲裁

I2C 总线上的仲裁分为两个部分：SCL 线上的同步和 SDA 线上的仲裁。

- SCL 线上的同步（时钟同步）

由于 I2C 总线具有线与的逻辑功能，SCL 线上只要有一个节点发送低电平，总线上就表现低电平。当所有的节点都发送高电平时，总线才能表现为高电平。所以，时钟低电平的时间由时钟电平期最长的器件决定，而时钟的高电平时间由时钟高电平期最短的器件决定。由于 I2C 这种特性，当多个主机同时发送时钟信号时，在总线上表示的是统一的时钟信号。如果从机希望主机降低传送速度可以通过将 SCL 主动拉低延长其低电平时间来通知主机，当主机在准备下一次传送时发现 SCL 的电平被拉低时进行等待，直到从机完成操作并释放 SCL 线的控制权。

- SDA 线上的仲裁

SDA 线上的仲裁也是由于 I2C 总线具有线与的逻辑功能。主机在发送数据后，通过比较总线上的数据来决定是否退出竞争。丢失仲裁的主机立即切换到未被寻址的从机状态，以确保自身能被仲裁胜利的主机寻址到。仲裁失败的主机继续输出时钟脉冲（在 SCL 上），直到发送完当前的串行字节。通过这种原理可以保证 I2C 总线在多个主机企图控制总线时保证数据的不丢失。

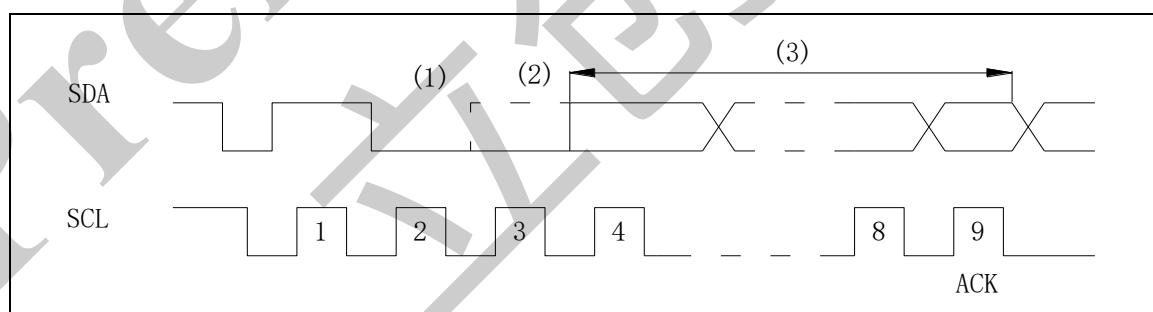


图 19-5 I2C 总线上的仲裁

- (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。

## 19.4 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。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。由于只有无方向端口，I2C 组件需要使用到管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。I2C 总线状态能在状态寄存器中进行查询。

### 19.4.1 功能框图

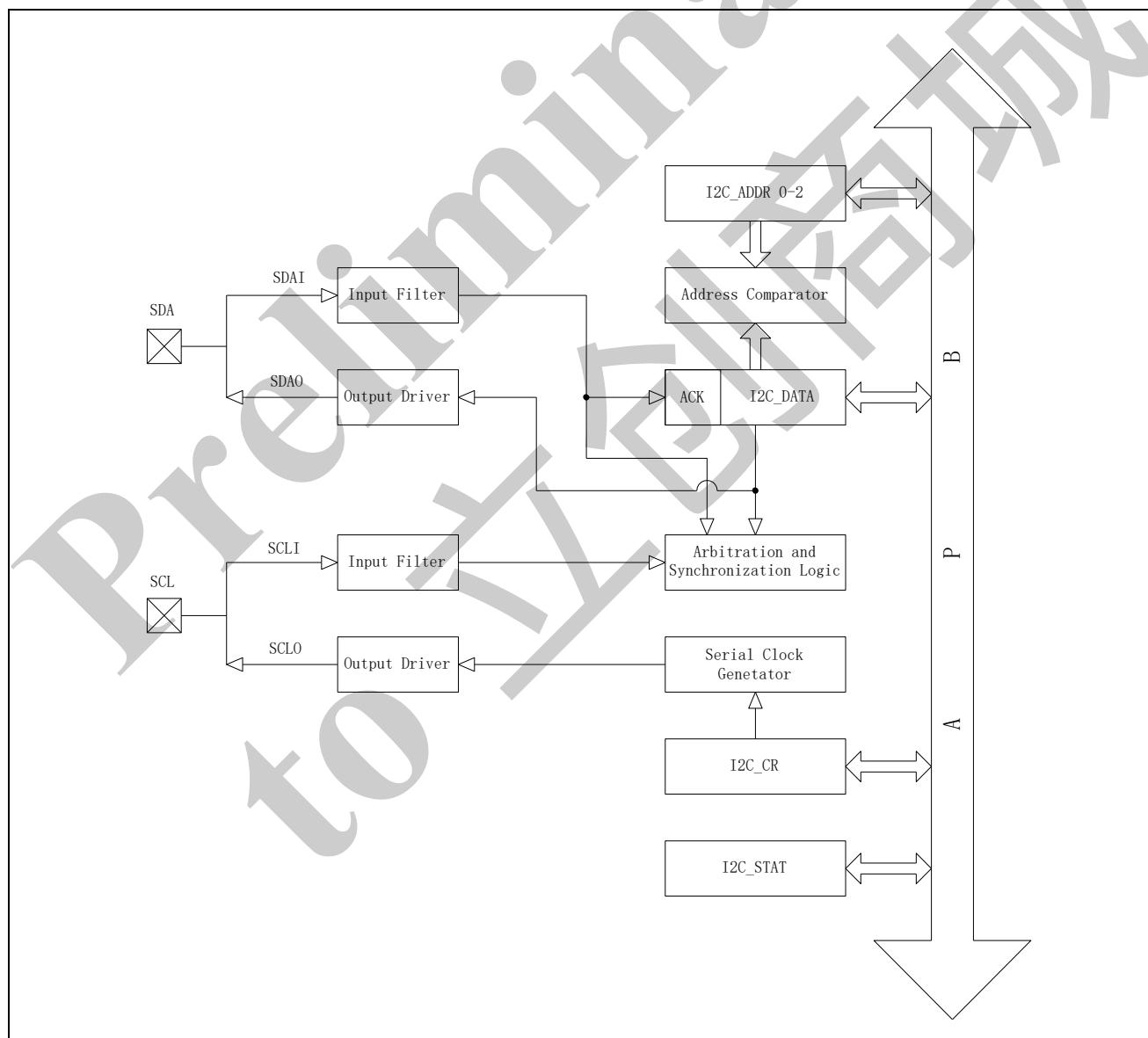


图 19-6 I2C 功能模块图

### 19.4.2 串行时钟发生器

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  $f_{SCL} = f_{PCLK} / 8 / (I2C\_BRR+1)$ , 其中 I2C\_BRR 应大于 0。

下表列出了 PCLK 频率与 I2C\_BRR 组合时, SCL 信号的输出频率值。

PCLK (KHz)	I2C_BRR						
	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

表 19-1 I2C 时钟信号波特率

### 19.4.3 输入滤波器

输入信号与 PCLK 同步, 低于 PCLK 周期的尖峰脉冲信号会被滤除。通过调整 I2C\_CR.FLTR 的值可变滤波的强度。

当本模块作为主机时, 如果 I2C\_BRR 的值小于等于 9, 应设置 I2C\_CR.FLT 为 1; 如果 I2C\_BRR 的值大于 9, 应设置 I2C\_CR.FLT 为 0。

当本模块作为从机时, 如果 PCLK 与 SCL 频率比值小于等于 30 时, 应设置 I2C\_CR.FLT 为 1; 如果 PCLK 与 SCL 频率比值大于 30 时, 应设置 I2C\_CR.FLT 为 0。

### 19.4.4 地址比较器

I2C 地址比较器将接收到的 7 位地址与三个从机地址与进行比较。并根据 I2C\_ADDRx 寄存器的值与首次接收到的 8 位字节或广播地址 (0x00) 相比较。如果任何一者相同,

I2C\_CR.SI 位会被置 1 并产生一个中断请求。查询 I2C\_MATCH 寄存器，以获取匹配成功的地址为序号。

#### 19.4.5 应答标志位

I2C\_CR.AA 标志位为应答标志位。当 AA 位为 1 时，I2C 模块收到数据后回应答位，当 AA 位为 0 时，I2C 模块收到数据后回非应答位。

#### 19.4.6 中断产生器

I2C\_CR.SI 为中断标志，每当状态寄存器 (I2C\_STAT) 的值产生变化时（除变为 0xF8 外），SI 标志位都会被置 1。当产生中断时，通过查询状态寄存器 (I2C\_STAT) 可获知 I2C 总线的状态，以确定中断的实际来源。为了进行下一步的操作，SI 标志位必须通过软件清零。

#### 19.4.7 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 100Kbps 而在高速模式下可达 400Kbps，在超高速模式下可达 1Mbps，并且可以在四种模式下工作：主机发送模式、主机接收模式、从机接收模式、从机发送模式。还有一种特殊模式广播呼叫模式，其操作方式与从机接收模式类似。

- 主机发送模式

主机发送多个字节到从机，主机产生时钟，所以需要在 I2C\_BRR 中填入设定值。主机发送模式中需要将 I2C\_CR.STA 置 1。当总线空闲时，主机发起一个起始位 START。若成功 I2C\_CR.SI 被置 1。接下来把从机地址和写位 (SLA+W) 写入 I2C\_DATA 中，清零 SI 位后，总线上发出 SLA+W。

主机发出 SLA+W 收到从机应答位 ACK 后，SI 被置 1。接下来根据用户定义格式发送数据。所有数据发送完后，将 I2C\_CR.STO 置 1，清零 SI 位后发出 STOP 信号，也可以发送重复起始信号进行新一轮数据传输。

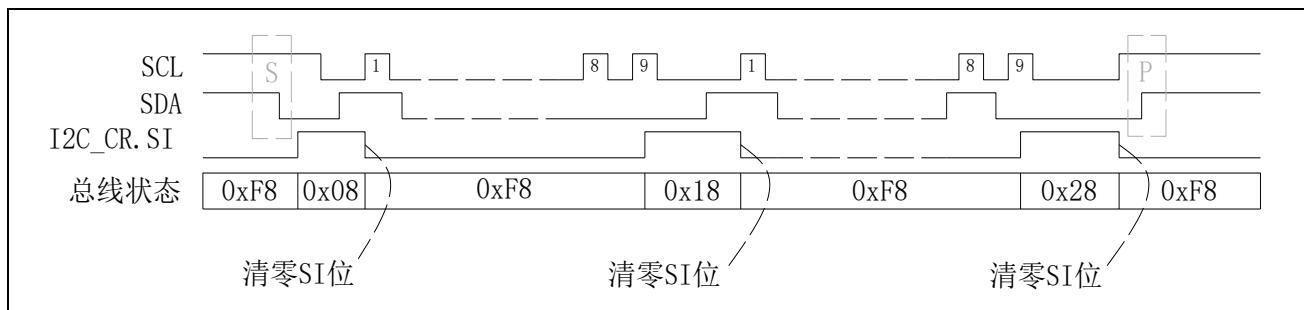


图 19-7 主发送模式数据同步图

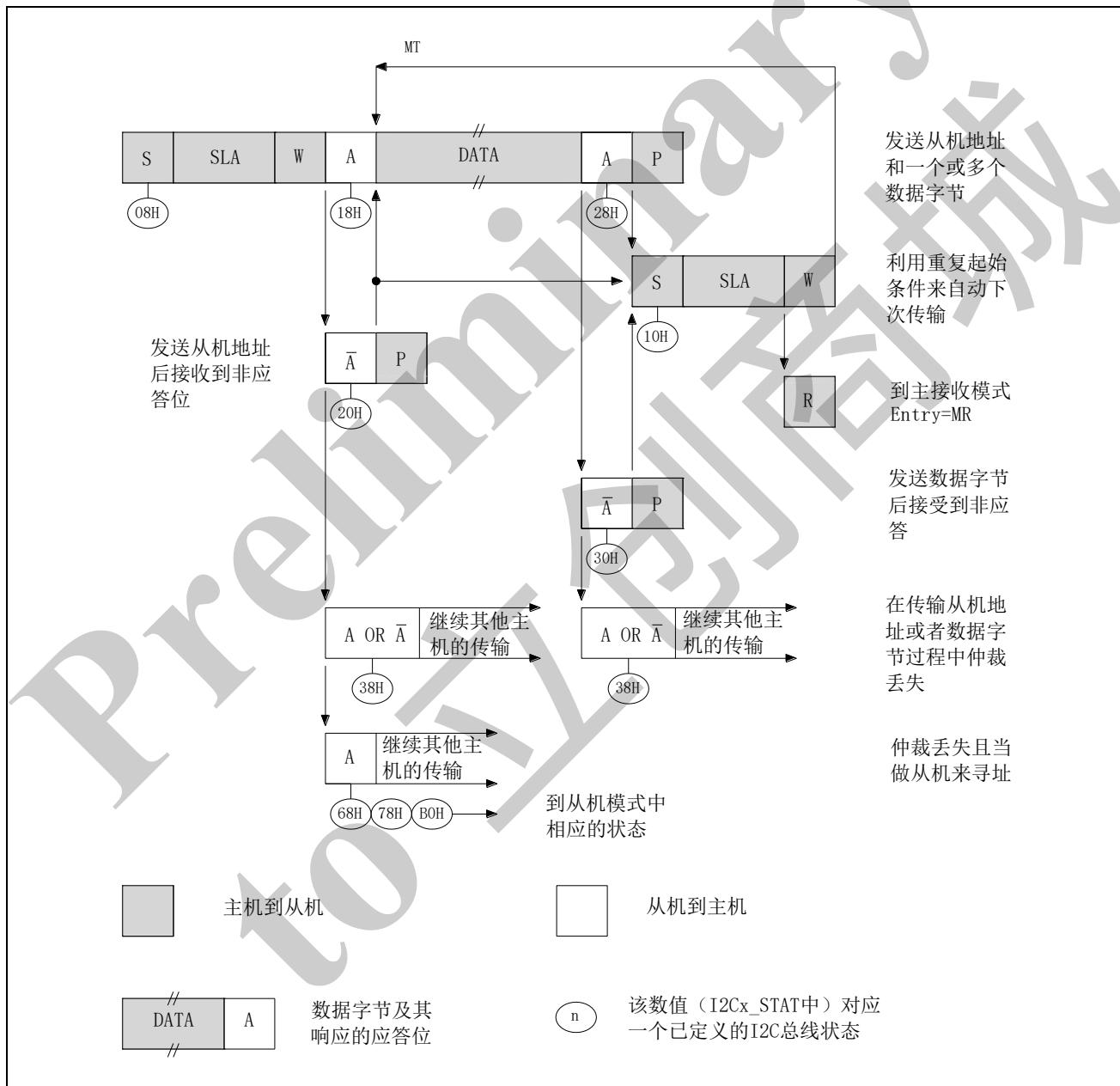


图 19-8 I2C 主机发送状态图

- 主机接收模式

主机接收模式，由从机传输数据。初始化设置与主机发送模式相同，主机发送起始位以后， I2C\_DATA 应写入从机地址和“读位” (SLA+R)。收到从机应答位 ACK 后 I2C\_CR.SI 被置 1。SI 清 0 后开始接收从机数据，若 I2C\_CR.AA 为 1，主机收到数据后回应答位；若为 0 主机收到数据后不回应答 NACK。然后主机可以发停止信号或重复起始信号开始下一轮的数据传输。

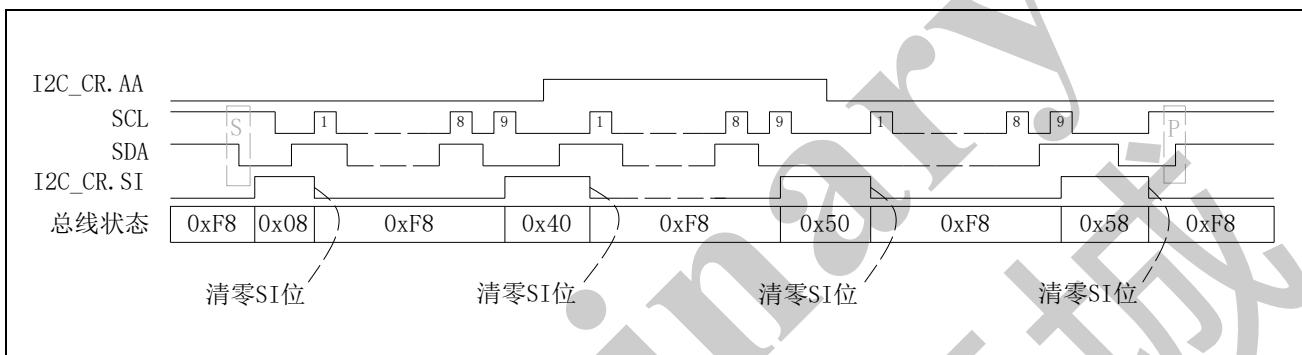


图 19-9 主接收模式数据同步图

Preliminary Draft

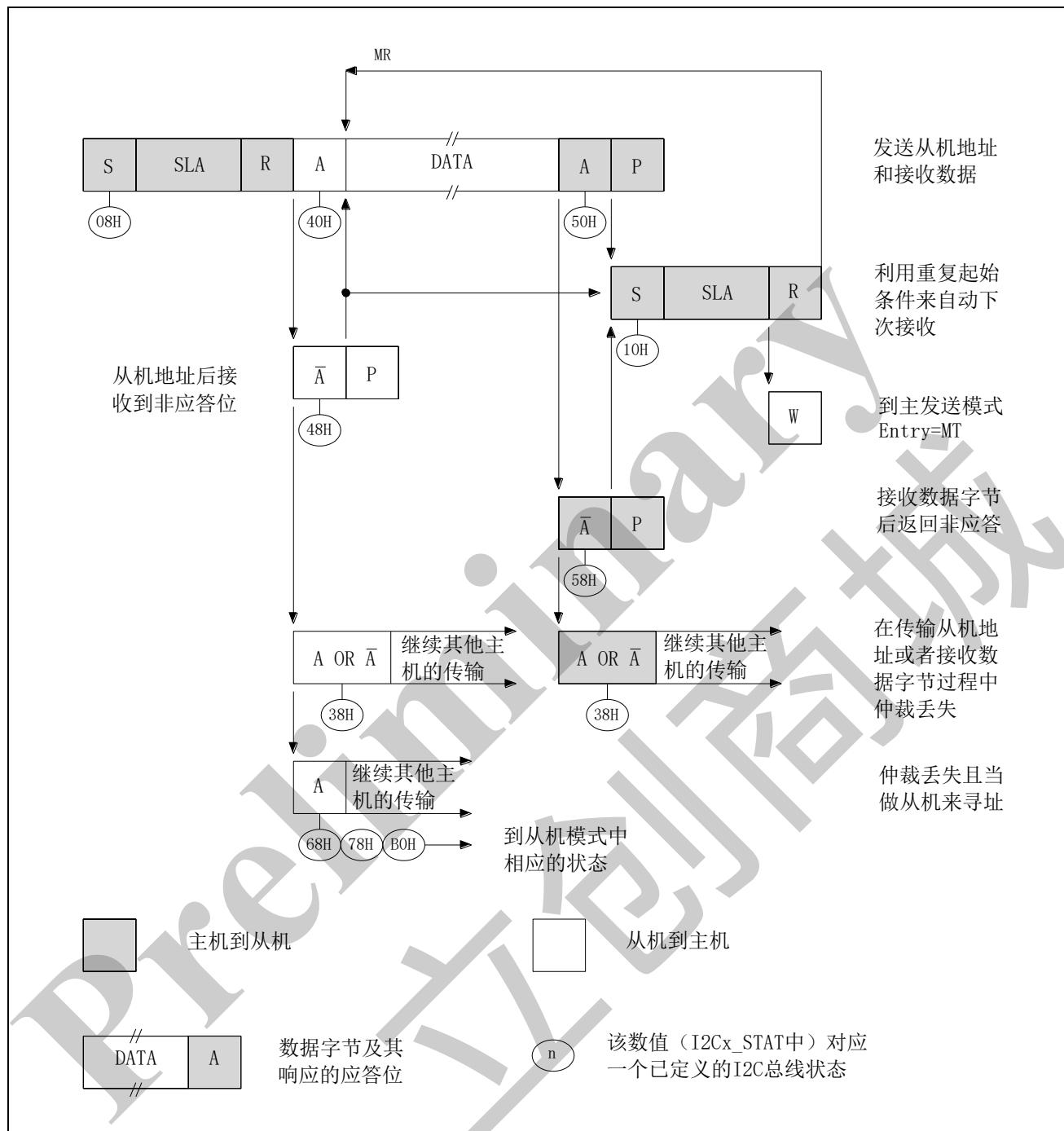


图 19-10 I2C 主机接收状态图

- 从机接收模式

在从机接收模式下，从机接收主机发来的数据。在传输开始前，I2C\_ADDRx 应写入从机地址，I2C\_CR.AA 置 1 用以响应主机的寻址。上述初始化后，从机进入空闲模式，等待“写”信号 (SLA+W)。若主机仲裁失败，也会直接进入从机接收模式。

当从机被“写”信号 SLA+W 寻址到后，需要清零 SI 位，以便从主机接收数据。如果在传输过程中 I2C\_CR.AA=0，从机将在下一字节返回无应答位 NACK，从机也将转为未寻址从机，与主机联系终止，不再接收数据，且 I2C\_DATA 保持之前接收到的数据。从机地址识别可通过置位 AA 来恢复，这意味着 AA 位可临时将 I2C 模块从 I2C 总线上分离出来。

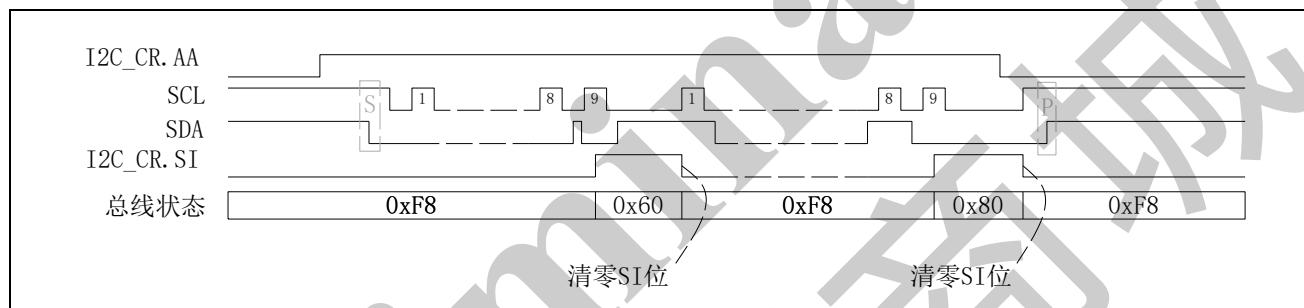


图 19-11 从接收模式数据同步图

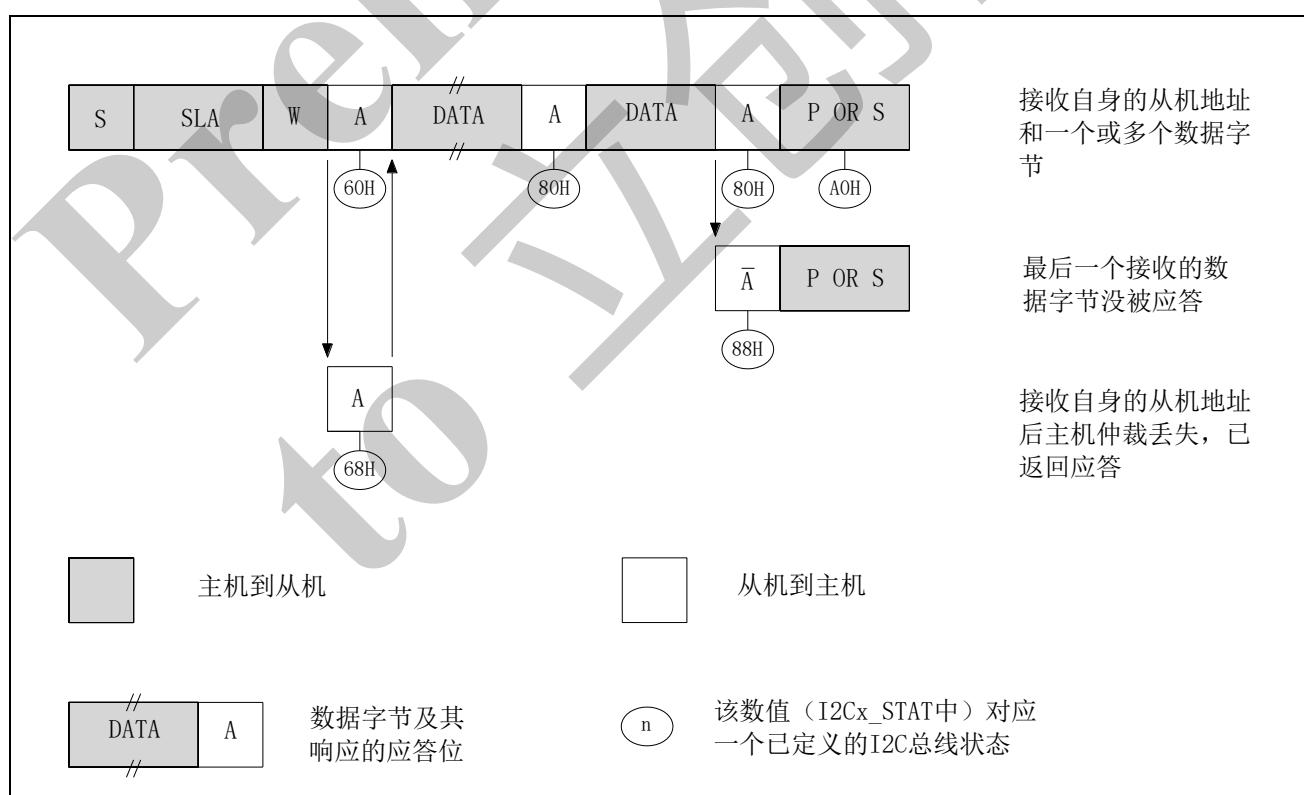


图 19-12 从机接收状态图

- 从机发送模式

从机发送模式，数据由从机发送给主机。当初始化 I2C\_ADDRx 及 I2C\_CR.AA 值后，器件等待直到自身地址被“读”信号(SLA+R)寻址。若主机仲裁失败，也可进入从机发送模式。

当从机被“读”信号 SLA+R 寻址，需要将 SI 清零用以向主机发送数据。通常主机接收每字节数据后会返回应答位。

如果传输过程中 I2C\_CR.AA 清零，从机将发送最后一个字节数据，并在接下去的传输中发送全 1 数据，并将自身变为未寻址从机。

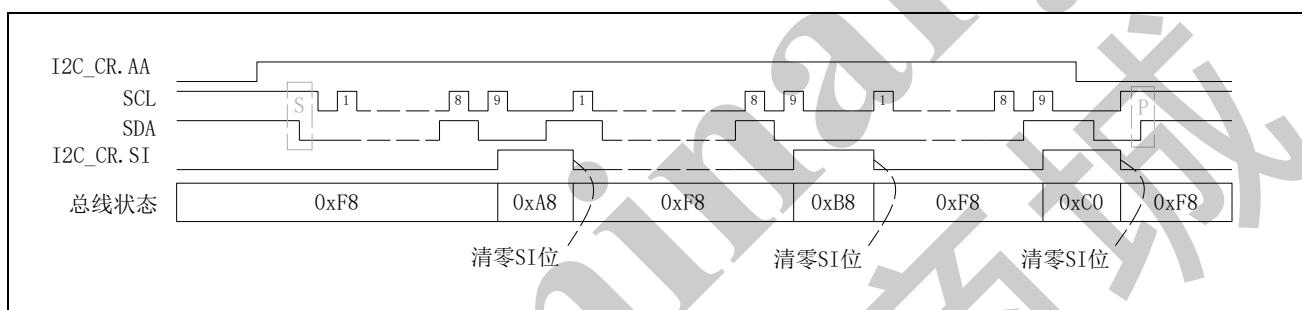


图 19-13 从发送模式数据同步图

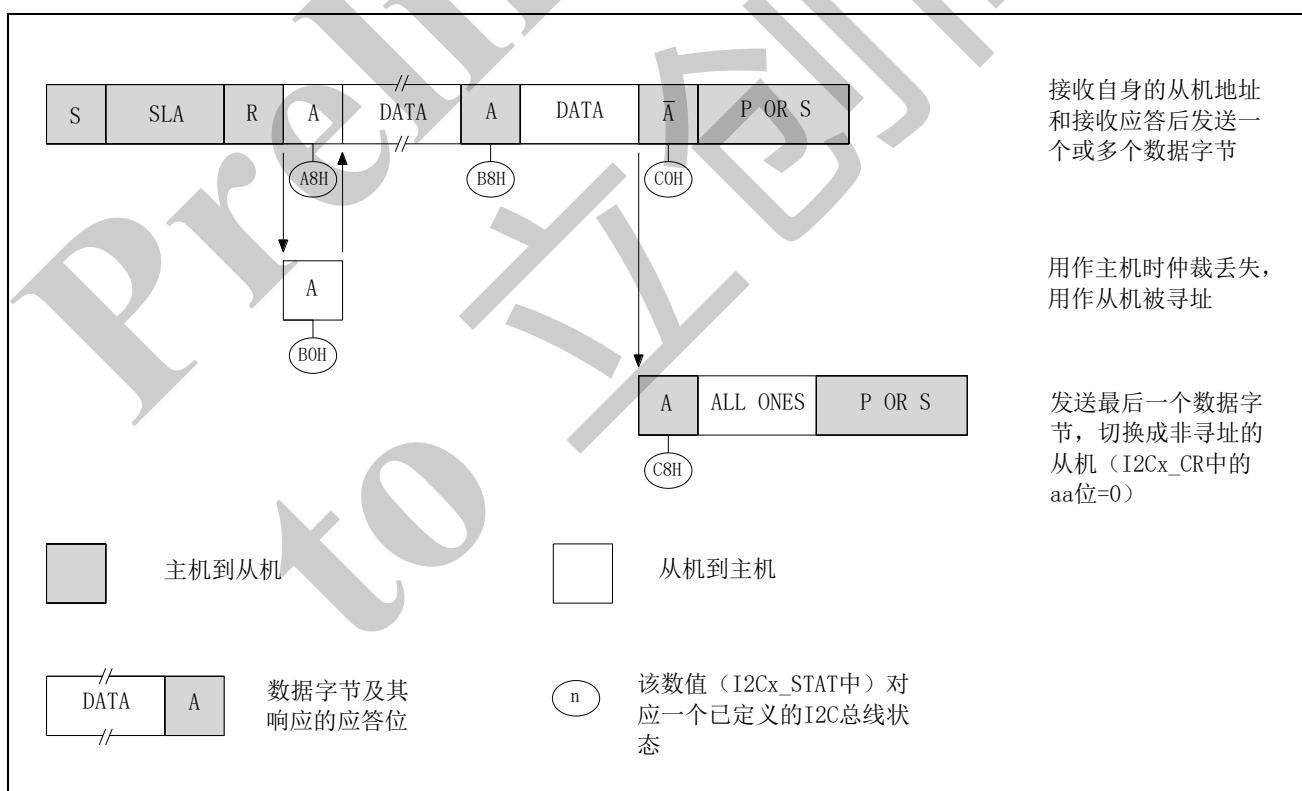


图 19-14 I2C 从机发送状态图

- 广播呼叫模式

广播呼叫模式是一种特殊的从机接收模式，寻址方式为 0x00，从机地址和读写都为 0。当 I2C\_ADDR0.GC 及 I2C\_CR.AA 都为置 1，使能接收广播呼叫模式。在该模式下 I2C\_STAT 值与普通从机接收模式 I2C\_STAT 值不同。仲裁失败也可能进入广播呼叫模式。

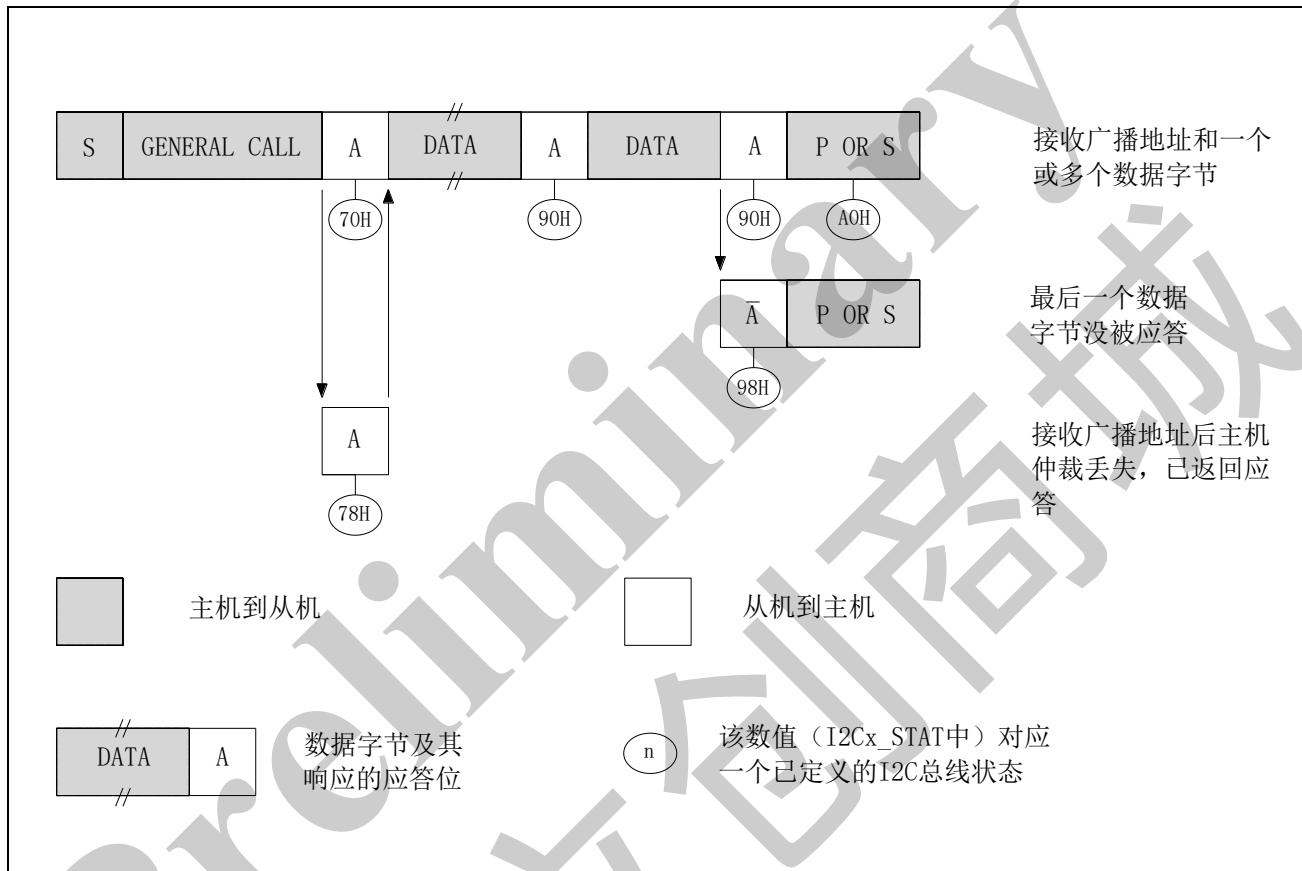


图 19-15 I2C 广播呼叫状态图

#### 19.4.8 状态码表述

I2C 状态寄存器中有两种特殊的状态：F8H 和 00H。

**F8H：**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。

**00H：**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。

状态代码	描述
主发送模式	
08H	已发送起始条件
10H	已发送重复起始条件
18H	已发送 SLA+W, 已接收 ACK
20H	已发送 SLA+W, 已接收非 ACK
28H	已发送 I2C_DATA 中的数据, 已接收 ACK
30H	已发送 I2C_DATA 中的数据, 已接收非 ACK
38H	在 SLA+读写或写数据字节时丢失仲裁
主接收模式	
08H	已发送起始条件
10H	已发送重复起始条件
38H	在非 ACK 中丢失仲裁
40H	已发送 SLA+R, 已接收 ACK
48H	已发送 SLA+R, 已接收非 ACK
50H	已接收数据字节, ACK 已返回
58H	已接收数据字节, 非 ACK 已返回
从接收模式	
60H	已接收自身的 SLA+W, 已返回 ACK
68H	主控时在 SLA+读写丢失仲裁, 已接收自身的 SLA+W, 已返回 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
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
A0H	静态寻址时, 接收到停止条件或重复起始条件
从发送模式	
A8H	已接收自身的 SLA+R, 已返回 ACK
B0H	当主机时丢失仲裁, 已接收自身 SLA+R, 已返回 ACK
B8H	已发送数据, 已接收 ACK

C0H	已发送数据字节, 已接收非 ACK
C8H	装入的数据字节已被发送, 已接收 ACK
广播呼叫模式	
70H	已接收广播地址 (0x00), 已返回 ACK
78H	主控时在 SLA+读写丢失仲裁, 已接收广播地址, 已返回 ACK
90H	前一次寻址使用广播地址, 已接收数据字节, 已返回 ACK
98H	前一次寻址使用广播地址, 已接收数据字节, 已返回非 ACK
A0H	静态寻址时, 接收到停止条件或重复起始条件
其余杂项状态	
F8H	无可用的相关状态信息, SI=0
00H	传输过程出现总线错误, 或外部干扰使 I2C 进入未定义的状态

表 19-2 I2C 状态码表述

Preliminary Draft  
to be confirmed

## 19.5 编程示例

### 19.5.1 主机发送示例

- Step1: 按 GPIO 章节管脚数字复用功能的相关描述, 将 SCL、SDA 映射到需要的管脚; 并配置 SCL、SDA 管脚为开漏输出模式。注意: SCL、SDA 需按照 I2C 协议的要求外接上拉电阻。
- Step2: 设置 SysCtrl\_PerClkEn0.I2C 为 1, 使能 I2C 配置时间钟和工作时钟。
- Step3: 向 SysCtrl\_PerReset0.I2C 依次写入 0、1, 复位 I2C 模块。
- Step4: 配置 I2C\_BRR, 使 SCL 的时钟速率符合应用需求。
- Step5: 设置 I2C\_BRREN 为 1, 使能 SCL 时钟发生器。
- Step6: 设置 I2C\_CR.EN 为 1, 使能 I2C 模块。
- Step7: 设置 I2C\_CR.STA 为 1, 总线尝试发送 Start 信号。
- Step8: 等待 I2C\_CR.SI 变为 1, Start 信号已发送到总线上。
- Step9: 查询 I2C\_STAT, 如果该寄存器值为 0x08 或 0x10, 继续执行下一步骤, 否则进行出错处理。
- Step10: 向 I2C\_DATA 中写入 SLA+W, 设置 I2C\_CR.SI 为 0, 发送 SLA+W。
- Step11: 等待 I2C\_CR.SI 变为 1, SLA+W 已发送到总线上。
- Step12: 查询 I2C\_STAT, 如果该寄存器值为 0x18, 继续执行下一步骤。否则进行出错处理。
- Step13: 向 I2C\_DATA 写入待发送的数据, 设置 I2C\_CR.SI 为 0, 发送数据。
- Step14: 等待 I2C\_CR.SI 变为 1, 数据已发送到总线上。
- Step15: 查询 I2C\_STAT, 如果该寄存器值为 0x28, 继续执行下一步骤。否则进行出错处理。
- Step16: 如待发送的数据未完成, 则跳转到 Step13 继续执行。
- Step17: 设置 I2C\_CR.STO 为 1, 设置 I2C\_CR.SI 为 0, 总线尝试发送 Stop 信号。
- Step18: 等待 I2C\_CR.SI 变为 1, Stop 信号已发送到总线上。

## 19.5.2 主机接收示例

- Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 SCL、SDA 映射到需要的管脚；并配置 SCL、SDA 管脚为开漏输出模式。注意：SCL、SDA 需按照 I2C 协议的要求外接上拉电阻。
- Step2：设置 SysCtrl\_PerIclkEn0.I2C 为 1，使能 I2C 配置时间钟和工作时钟。
- Step3：向 SysCtrl\_PerIReset0.I2C 依次写入 0、1，复位 I2C 模块。
- Step4：配置 I2C\_BRR，使 SCL 的时钟速率符合应用需求。
- Step5：设置 I2C\_BRREN 为 1，使能 SCL 时钟发生器。
- Step6：设置 I2C\_CR.EN 为 1，使能 I2C 模块。
- Step7：设置 I2C\_CR.STA 为 1，总线尝试发送 Start 信号。
- Step8：等待 I2C\_CR.SI 变为 1，Start 信号已发送到总线上。
- Step9：查询 I2C\_STAT，如果寄存器值为 0x08 或 0x10，继续执行下一步骤，否则进行出错处理。
- Step10：向 I2C\_DATA 写入 SLA+R，设置 I2C\_CR.SI 为 0，发送 SLA+R。
- Step11：等待 I2C\_CR.SI 变为 1，SLA+R 已发送到总线上。
- Step12：查询 I2C\_STAT，如果寄存器值为 0x40，继续执行下一步骤，否则进行出错处理。
- Step13：设置 I2C\_CR.AA 为 1，使能应答标志。
- Step14：设置 I2C\_CR.SI 为 0，从机发送数据，主机根据 I2C\_CR.AA 发送 ACK 或 NACK。
- Step15：等待 I2C\_CR.SI 变为 1，从 I2C\_DATA 读取已接收到的数据。
- Step16：查询到 I2C\_STAT，如果该寄存器值为 0x50 或 0x58，继续执行下一步骤，否则进行出错处理。
- Step17：如果待接收的数据只差最后一个字节，设置 I2C\_CR.AA 为 0，使能非应答标志。
- Step18：如待接收的数据未完成，则跳转到 Step14 继续执行。
- Step19：设置 I2C\_CR.STO 为 1，设置 I2C\_CR.SI 为 0，总线尝试发送 Stop 信号。
- Step20：等待 I2C\_CR.SI 变为 1，Stop 信号已发送到总线上。

### 19.5.3 从机接收示例

- Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 SCL、SDA 映射到需要的管脚；并配置 SCL、SDA 管脚为开漏输出模式。
- Step2：设置 SysCtrl\_PerIclkEn0.I2C 为 1，使能 I2C 配置时间钟和工作时钟。
- Step3：向 SysCtrl\_PerIReset0.I2C 依次写入 0、1，复位 I2C 模块。
- Step4：设置 I2C\_CR.EN 为 1，使能 I2C 模块。
- Step5：配置 I2C\_ADDRx 为从机地址。
- Step6：设置 I2C\_CR.AA 为 1，使能应答标志。
- Step7：等待 I2C\_CR.SI 变为 1，被 SLA+W 寻址。
- Step8：查询 I2C\_STAT，如果该寄存器值为 0x60，继续执行下一步骤，否则进行出错处理。
- Step9：设置 I2C\_CR.SI 为 0，主机发送数据，从机根据 I2C\_CR.AA 返回 ACK 或 NACK。
- Step10：等待 I2C\_CR.SI 变为 1，从 I2C\_DATA 中读取已接收到的数据。
- Step11：查询 I2C\_STAT，如果该寄存器值为 0x80，继续执行下一步骤，否则进行出错处理。
- Step12：如待接收的数据未完成，则跳转到 Step9 继续执行。
- Step13：设置 I2C\_CR.AA 为 0，设置 I2C\_CR.SI 为 0。

#### 19.5.4 从机发送示例

- Step1：按 GPIO 章节管脚数字复用功能的相关描述，将 SCL、SDA 映射到需要的管脚；并配置 SCL、SDA 管脚为开漏输出模式。
- Step2：设置 SysCtrl\_PerIclkEn0.I2C 为 1，使能 I2C 配置时间钟和工作时钟。
- Step3：向 SysCtrl\_PerIReset0.I2C 依次写入 0、1，复位 I2C 模块。
- Step4：设置 I2C\_CR.EN 为 1，使能 I2C 模块。
- Step5：配置 I2C\_ADDRx 为从机地址。
- Step6：设置 I2C\_CR.AA 为 1，使能应答标志。
- Step7：等待 I2C\_CR.SI 变为 1，被 SLA+R 寻址。
- Step8：查询 I2C\_STAT，如果该寄存器的值为 0xA8，继续执行下一步骤，否则进行出错处理。
- Step9：向 I2C\_DATA 写入待发送的数据，设置 I2C\_CR.SI 为 0，发送数据。
- Step10：等待 I2C\_CR.SI 变为 1，数据已发送到总线上。
- Step11：查询 I2C\_STAT，如果该寄存器的值为 0xB8 或 0xC0 时，继续执行下一步骤，否则进行出错处理。
- Step12：如待发送的数据未完成，则跳转到 Step9 继续执行。
- Step13：设置 I2C\_CR.AA 为 0，设置 I2C\_CR.SI 为 0。

## 19.6 寄存器描述

### 19.6.1 寄存器列表

基地址: 0x40000400

偏移量	寄存器名称	访问	寄存器描述
0x00	I2C_BRREN	RW	I2C 波特率发生器使能寄存器
0x04	I2C_BRR	RW	I2C 波特率计数器配置寄存器
0x08	I2C_CR	RW	I2C 配置寄存器
0x0c	I2C_DR	RW	I2C 数据寄存器
0x14	I2C_STAT	RO	I2C 状态寄存器
0x10	I2C_ADDR0	RW	I2C 从机地址0寄存器
0x20	I2C_ADDR1	RW	I2C 从机地址1寄存器
0x24	I2C_ADDR2	RW	I2C 从机地址2寄存器
0x28	I2C_MATCH	RW	I2C 从从机地址匹配寄存器

表 19-3 寄存器列表

### 19.6.2 波特率发生器使能寄存器(I2C\_BRREN)

地址偏移量: 0x00

复位值: 0x0000 0000

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

位	标记	功能描述
31: 1	Reserved	保留位
0	EN	波特率发生器使能控制 1: 使能 0: 禁止

### 19.6.3 波特率计数器配置寄存器(I2C\_BRR)

地址偏移量: 0x04

复位值: 0x0000 0000

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

位	标记	功能描述
31:8	Reserved	保留位
7:0	BRR	I2C 总线 SCL 波特率配置 $f_{SCL} = f_{PCLK} / 8 / (BRR+1)$ , 其中 BRR >0

#### 19.6.4 配置寄存器(I2C\_CR)

地址偏移量: 0x08

复位值: 0x0000 0000

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

Reserved	EN	STA	STO	SI	AA	Res	FLT
	RW	RW	RW	RW	RW		RW

位	标记	功能描述
31:7	Reserved	保留位
6	EN	模块使能控制 1: 使能 0: 禁止
5	STA	总线状态控制 1: 向总线发送 START 0: 无功能 注意: START 发送完成后, 用户程序应设置该比特为 0。
4	STO	总线状态控制 1: 向总线发送 STOP 0: 无功能 注意: STOP 发送完成后硬件自动设置该比特为 0。
3	SI	I2C 中断标志 读出 1: 已发生 I2C 中断 读出 0: 未发生 I2C 中断 写入 1: 无功能 写入 0: 清除 I2C 中断标志并使状态机执行下一个动作
2	AA	应答控制 1: 在应答阶段发送 ACK 0: 在应答阶段发送 NAK
1	Reserved	
0	FLT	I2C 滤波参数配置 1: 简单滤波, 更快的通信速率 0: 高级滤波, 更高的抗干扰性能 注: 详见输入滤波器章节。

### 19.6.5 数据寄存器(I2C\_DR)

地址偏移量: 0x0c

复位值: 0x00000000

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

位	标记	功能描述
31:8	Reserved	保留位
7:0	DR	数据寄存器 在发送模式下, 写入待发送的数据 在接收模式下, 读出接收到的数据

### 19.6.6 状态寄存器(I2C\_STAT)

地址偏移量: 0x14

复位值: 0x00000000

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

位	标记	功能描述
31:8	Reserved	保留位
7:0	STAT	I2C 状态寄存器, 状态值的具体定义详见【状态码表述】章节

### 19.6.7 从机地址 0 寄存器(I2C\_ADDR0)

地址偏移量: 0x10

复位值: 0x00000000

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

位	标记	功能描述
31:8	Reserved	保留位
7:1	Addr	I2C 从机模式地址 0
0	GC	广播地址应答使能 1: 使能 0: 禁止

### 19.6.8 从机地址 1 寄存器(I2C\_ADDR1)

地址偏移量: 0x20

复位值: 0x00000000

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

位	标记	功能描述
31:8	Reserved	保留位
7:1	Addr	I2C 从机模式地址 1.
0	Reserved	保留位

### 19.6.9 从机地址 2 寄存器(I2C\_ADDR2)

地址偏移量: 0x24

复位值: 0x00000000

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

位	标记	功能描述
31:8	Reserved	保留位
7:1	Addr	I2C 从机模式地址 2.
0	Reserved	保留位

### 19.6.10 从机地址匹配寄存器(I2C\_MATCH)

地址偏移量: 0x28

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															
												AD2F	AD1F	AD0F	
												RO	RO	RO	

位	标记	功能描述
31:3	Reserved	保留位
2	AD2F	I2C 从机模式地址 2 匹配标志位 1: 从总线收到的设备地址与 ADDR2 相同 0: 从总线收到的设备地址与 ADDR2 不相同
1	AD1F	I2C 从机模式地址 1 匹配标志位 1: 从总线收到的设备地址与 ADDR1 相同 0: 从总线收到的设备地址与 ADDR1 不相同
0	AD0F	I2C 从机模式地址 0 匹配标志位 1: 从总线收到的设备地址与 ADDR0 相同 0: 从总线收到的设备地址与 ADDR0 不相同

注: 地址匹配标志位在以下三种情况下会清零:

- 模块复位时
- START/STOP 发送时
- 从机内部 STOP 时

## 20 模数转换器 (ADC)

### 20.1 模块简介

外部的模拟信号需要转变成数字信号才能由 MCU 进一步处理。内部集成了一个 10 位高精度、高转换速率的逐次逼近型模数转换器(SAR ADC)模块。具有以下特性：

- 10 位转换精度；
- 1M SPS 转换速度；
- 支持单次转换及持续转换；
- 2 种参考源：AVCC 电压、ExRef 管脚；
- 15 路输入通道，包括 14 路外部管脚输入、1 路内建 BGR 0.9V 电压；  
注：选择 AVCC 作为参考源，测量 BGR 0.9V；可计算出 AVCC 电压。
- ADC 的电压输入范围：0~Vref；
- 支持片内外设自动触发 ADC 转换，有效降低芯片功耗并提高转换的实时性。

## 20.2 ADC 框图

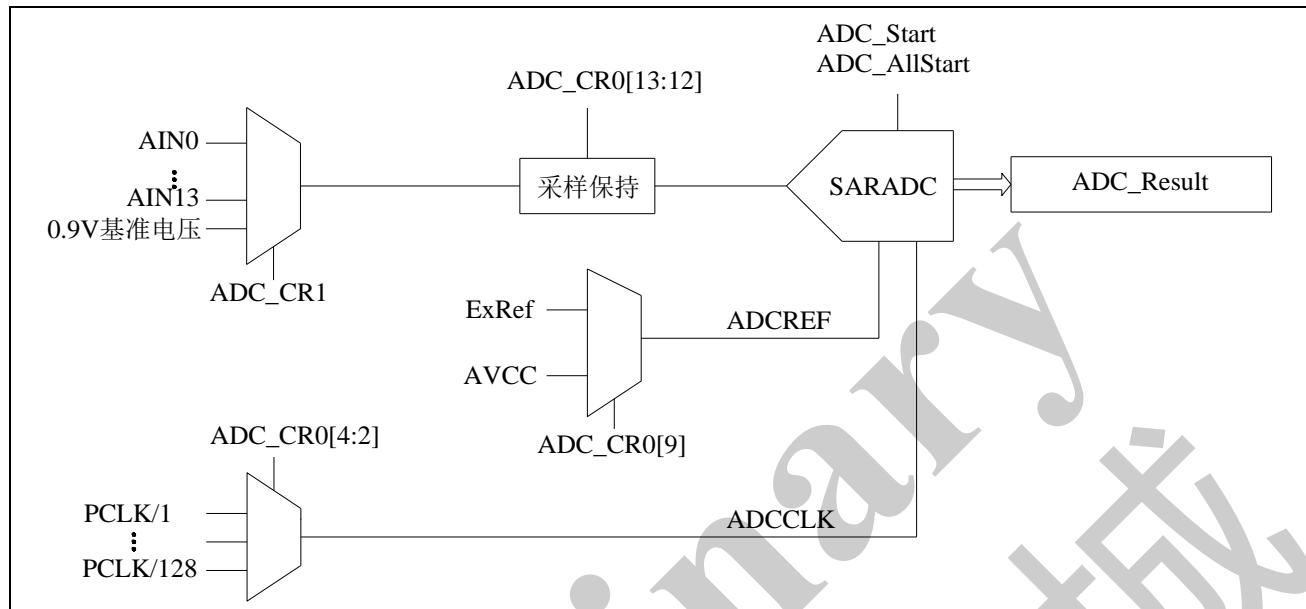


图 20-1 ADC 示意框图

## 20.3 转换时序及转换速度

ADC 的转换时序如下图所示:一次完整的 ADC 转换由转换过程及逐次比较过程组成。其中转换过程需要 6~12 个 ADCCLK, 由 ADC\_CR0.SAM 配置; 逐次比较过程需要 13 个 ADCCLK。所以, 一次 ADC 转换共需要 19~25 个 ADCCLK。

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: ADCCLK 的频率 / 一次 ADC 转换所需要的 ADCCLK 的个数。

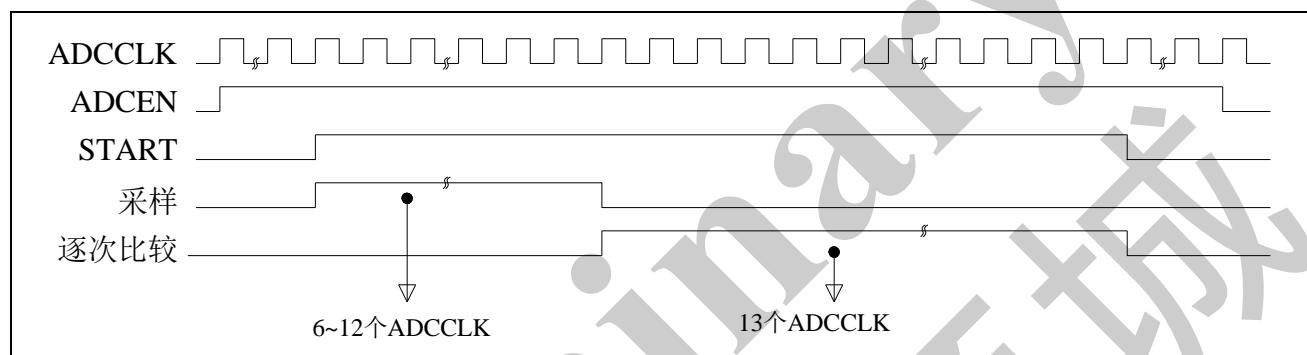


图 20-2 ADC 转换时序图

ADC 转换速度与 ADC 参考电压及 AVCC 电压相关, 最高转换速度如下表所示:

ADC 参考电压	AVCC 电压	最高转换速度	最大 ADCCLK 频率
AVCC / ExRef	1.8V~2.4V	200K SPS	4MHz
AVCC / ExRef	2.4V~2.7V	500K SPS	16MHz
AVCC / ExRef	2.7V~5.5V	1M SPS	24MHz

## 20.4 单次转换模式

在单次转换模式下，ADC 启动后只执行一次设定通道的转换。该模式既可通过设置 ADC\_start.Start 位启动也可通过设置 ADC\_ExtTrigger 的外部触发启动。每个通道 ADC 转换完成后 ADC\_IFR.EOC 位会自动置 1，结果保存在 ADC\_Result 寄存器中。

### ADC 单次转换操作流程：

- Step1：配置 GPIOx\_ADS 相应的位，将待转换的 ADC 通道配置为模拟端口。
- Step2：设置 GPIOC\_ADS.bit3 为 1，将 ADC 外部参考电压管脚配置为模拟端口。  
注：如果 ADC 参考电压不选择外部参考电压管脚，则可以略过本步骤。
- Step3：设置 ADC\_CR0.En 为 1，使能 ADC 模块。
- Step4：配置 ADC\_CR0.Ref，选择 ADC 的参考电压。
- Step5：配置 ADC\_CR0.SAM 及 ADC\_CR0.CkDiv，设置 ADC 的转换速度。
- Step6：延时 5us，等待 ADC 模块启动完成。
- Step7：配置 ADC\_CR1.CHSEL，选择待转换的通道。
- Step8：设置 ADC\_ICR 为 0，清除 ADC\_IFR 标志。
- Step9：设置 ADC\_Start.Start 为 1，启动 ADC 单次转换。
- Step10：等待 ADC\_IFR.EOC 变为 1，读取 ADC\_Result 寄存器以获取 ADC 转换结果。
- Step11：如需对其他通道进行转换，重复执行 Step7~Step10。
- Step12：设置 ADC\_CR0.En 关闭 ADC 模块。

## 20.5 外部触发转换模式

ADC 转换既可通过软件配置启动，也可通过外部触发启动。

配置 ADC\_ExtTrigger 寄存器可设置 ADC 转换的外部触发源。

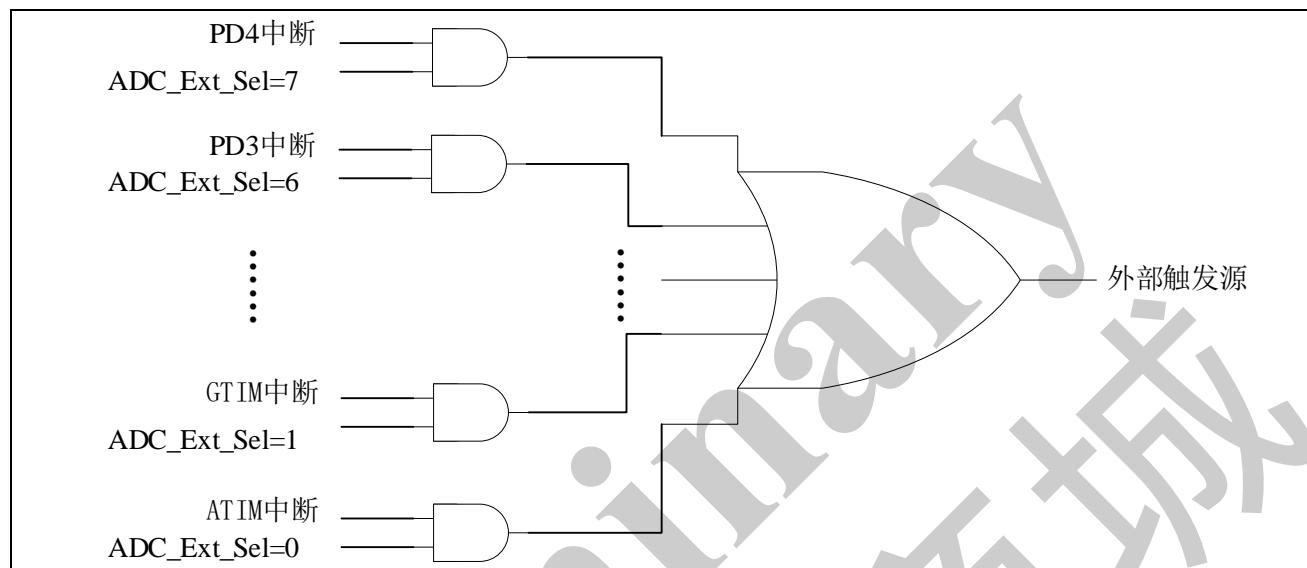


图 20-3 ADC 转换外部触发源示意图

## 20.6 持续转换模式

当设置 ADC\_AllStart.Start 为 1 时，ADC 模块工作在持续转换模式下；当设置 ADC\_AllStart.Start 为 0 时，ADC 模块退出续转换模式。

在持续转换模式下，ADC 模块将持续执行 ADC 转换，每次 ADC 转换完成后 ADC\_IFR.EOC 位会自动置 1，结果保存在 ADC\_Result 寄存器中。当第一次转换完成后，在任意时刻用户程序均可读取 ADC\_Result 寄存器以获取当前时刻的 ADC 转换的结果；每次读取 ADC\_Result 均会清除 ADC\_IFR.EOC 标志。

### ADC 持续转换操作流程：

Step1：配置 GPIOx\_ADS 相应的位，将待转换的 ADC 通道配置为模拟端口。

Step2：设置 GPIOC\_ADS.bit3 为 1，将 ADC 外部参考电压管脚配置为模拟端口。

注：如果 ADC 参考电压不选择外部参考电压管脚，则可以略过本步骤。

Step3：设置 ADC\_CR0.En 为 1，使能 ADC 模块。

Step4：配置 ADC\_CR0.Ref，选择 ADC 的参考电压。

Step5：配置 ADC\_CR0.SAM 及 ADC\_CR0.CkDiv，设置 ADC 的转换速度。

Step6：延时 5us，等待 ADC 模块启动完成。

Step7：配置 ADC\_CR1.CHSEL，选择待转换的通道。

Step8：设置 ADC\_AllStart.Start 为 1，启动 ADC 持续转换。

Step9：首次 ADC 转换完成后，在任意时刻均可读取 ADC\_Result 寄存器以获取 ADC 实时转换的结果。

Step10：设置 ADC\_AllStart.Start 为 0，退出持续转换模式。

Step11：设置 ADC\_CR0.En 关闭 ADC 模块。

## 20.7 测量 AVCC 电压

设置 ADC 的参考源为 AVCC，设置 ADC 的输入通道为内部 BGR 0.9V，通过 ADC 转换结果可以计算出 AVCC 的电压。注意，该条件下 ADC 的转换速度不应超过 200K SPS。该条件下，AVCC 电压的计算公式为： $V_{AVCC} = \frac{V_{BGR} \times 1023}{ADC\_Result}$ 。

其中  $V_{BGR}$  为内部参考电压，其典型值为 0.9V。

计算示例：

当 ADC\_Result 为 279 时，AVCC 的电压为  $\frac{0.9 \times 1023}{279} = 3.3V$ 。

当 ADC\_Result 为 460 时，AVCC 的电压为  $\frac{0.9 \times 1023}{460} = 2.0V$ 。

## 20.8 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 单次转换完成	ADC_IFR.EOC	ADC_IER.EOC

Preliminary  
to be confirmed

## 20.9 ADC 模块寄存器

基地址 0x40002400

寄存器	偏移地址	描述
ADC_CR0	0x004	ADC 配置寄存器0
ADC_CR1	0x008	ADC 配置寄存器1
ADC_Result	0x0A0	ADC 转换结果寄存器
ADC_IFR	0x0B0	ADC 中断标志寄存器
ADC_IER	0x0B4	ADC 中断使能寄存器
ADC_ICR	0x0B8	ADC 中断清除寄存器
ADC_ExtTrigger	0x0BC	ADC 转换外部中断触发源配置寄存器
ADC_Start	0x0C0	ADC 转换启动控制寄存器
ADC_AllStart	0x0CC	ADC 持续转换启动控制寄存器

表 20-1 ADC 寄存器

### 20.9.1 ADC 配置寄存器 0 (ADC\_CR0)

偏移地址 0x004

复位值 0x00002600

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SAM	Res.	Ref		Res.				CkDiv	Res.	En			RW
		RW		RW						RW					RW

位	标记	功能描述
31:14	Reserved	保留
13:12	SAM	ADC采样周期选择 (转换另外需要13个ADC时钟周期) 00: 6个ADC时钟周期 01: 8个ADC时钟周期 10: 11个ADC时钟周期 11: 12个ADC时钟周期
11	Reserved	保留
10:9	Ref	ADC参考电压选择 x0: 外部参考电压ExRef (PC03) x1: AVCC电压
8:5	Reserved	保留
4:2	CkDiv	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
1	Reserved	保留
0	En	ADC使能控制 1: 使能ADC 0: 禁止ADC

### 20.9.2 ADC 配置寄存器1 (ADC\_CR1)

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CHSEL	
														RW	

位	标记	功能描述
31:4	Reserved	保留
3:0	CHSEL	通道选择 0000: 通道0 管脚PC04 0001: 通道1 管脚PC06 0010: 通道2 管脚PD02 0011: 通道3 管脚PD03 0100: 通道4 管脚PD04 0101: 通道5 管脚PD05 0110: 通道6 管脚PD06 0111: 通道7 管脚PA01 1000: 通道8 管脚PA02 1001: 通道9 管脚PB05 1010: 通道10 管脚PB04 1011: 通道11 管脚PB02 1100: 通道12 管脚PB01 1101: 通道13 管脚PB00 1110: 通道14 内部0.9V基准电压

### 20.9.3 ADC 转换结果寄存器 (ADC\_Result)

偏移地址 0x0A0

复位值 0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Result															
RO															

位	标记	功能描述
31:16	Reserved	保留
15:0	Result	ADC转换结果 注: 读ADC_Result可将ADC_IFR.EOC清零

### 20.9.4 ADC 中断标志寄存器 (ADC\_IFR)

偏移地址 0x0B0

复位值 0x00000000

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

位	标记	功能描述
31:1	Reserved	保留
0	EOC	ADC一次转换完成标志 1: ADC一次转换完成 0: ADC一次转换未完成

### 20.9.5 ADC 中断使能寄存器 (ADC\_IER)

偏移地址 0x0B4

复位值 0x00000000

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

位	标记	功能描述
31:1	Reserved	保留
0	EOC	ADC一次转换完成中断使能 1: 使能 0: 禁止

### 20.9.6 ADC 标志清除寄存器 (ADC\_ICR)

偏移地址 0x0B8

复位值 0x00000001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															EOC
															R1W0

位	标记	功能描述
31:1	Reserved	保留
0	EOC	写0清除ADC单次转换完成标志 写1无作用

### 20.9.7 ADC 外部触发配置寄存器 (ADC\_ExtTrigger)

偏移地址 0x0BC

复位值 0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												Source			
												RW			

位	标记	功能描述
31:4	Reserved	保留
3:0	Source	<p>ADC触发源选择</p> <p>0000: ATIM3输出的触发信号, 详见ATIM3_ADTR</p> <p>0001: GTIM溢出信号</p> <p>0010: PB03中断标志由0变为1的信号</p> <p>0011: PB04中断标志由0变为1时信号</p> <p>0100: PC03中断标志由0变为1时信号</p> <p>0101: PC04中断标志由0变为1时信号</p> <p>0110: PD03中断标志由0变为1时信号</p> <p>0111: PD04中断标志由0变为1时信号</p> <p>注意: 当选定的触发源的GPIO时, 用户程序应及时清除触发源的中断标志。</p>

### 20.9.8 ADC 转换启动控制寄存器 (ADC\_Start)

偏移地址 0x0C0

复位值 0x00000000

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

位	标记	功能描述
31:1	Reserved	保留
0	Start	ADC转换控制 1: 启动ADC转换，一次转换完成后自动清零。 0: 无动作

### 20.9.9 ADC 持续转换控制寄存器 (ADC\_AllStart)

偏移地址 0x0CC

复位值 0x00000000

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

位	标记	功能描述
31:1	Reserved	保留
0	Start	ADC持续转换控制 1: 启动ADC持续转换 0: 停止ADC持续转换，需要等待当次转换结束

## 21 低电压检测器 (LVD)

### 21.1 LVD 简介

LVD 可用于监测 VCC 及芯片管脚的电压。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，用户可根据该信号执行一些紧急任务。该模块在 Active、Sleep、DeepSleep 模式时均可正常工作。

LVD 具有以下特性：

- 4 路监测源，AVCC、PA03、PC03、PD04；
- 16 阶阈值电压，1.8V~3.3V 可选；
- 8 种触发条件，高电平、上升沿、下降沿组合；
- 2 种触发结果，复位、中断；
- 8 阶滤波配置，防止误触发；
- 具备迟滞功能，强力抗干扰；

### 21.2 LVD 框图

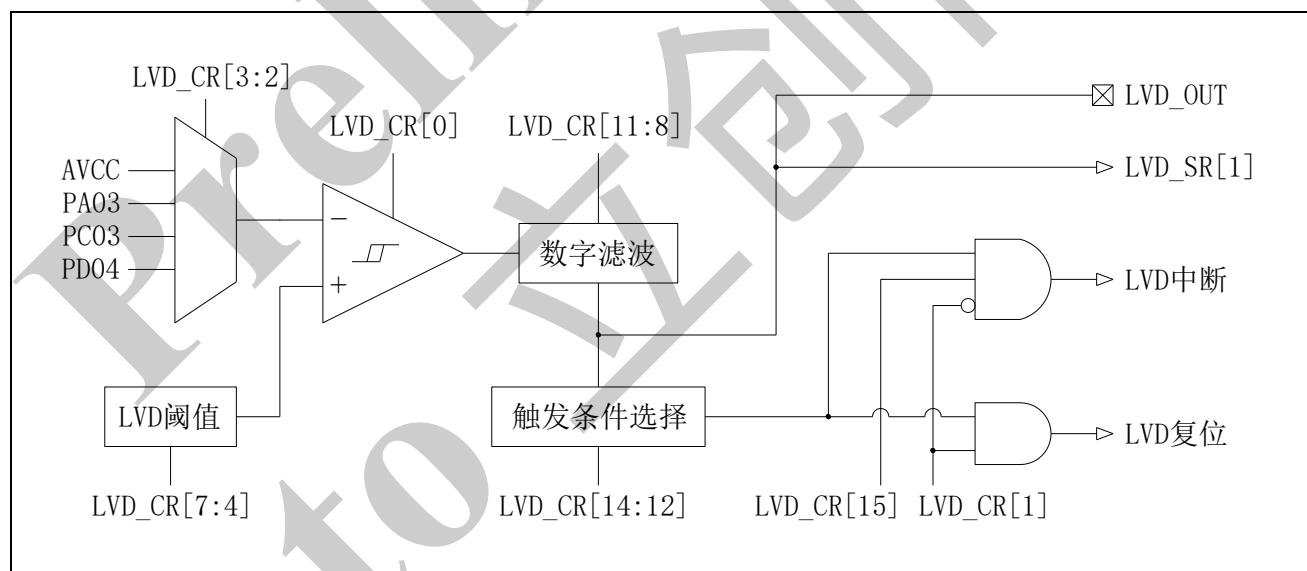


图 21-1 LVD 框图

## 21.3 迟滞功能

LVD 内置的电压比较器具有迟滞功能，可以增强芯片的抗干扰能力。当禁止 LVD 时，LVD 的输出信号为低电平；当使能 LVD 时，LVD 的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，其输入输出信号示意如下：

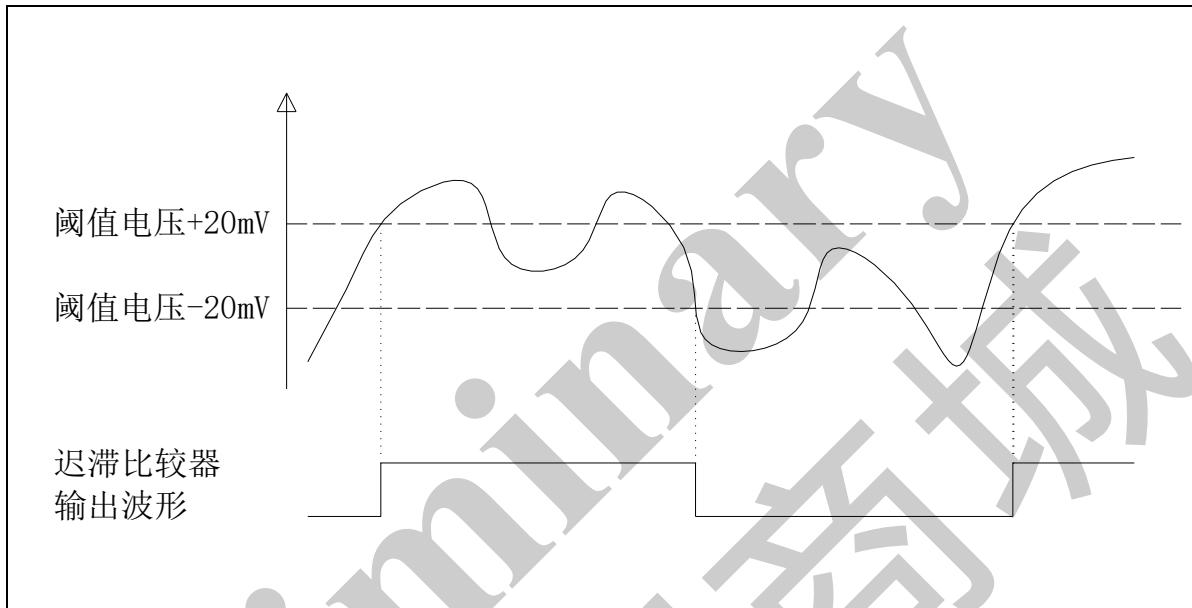


图 21-2 LVD 迟滞响应

## 21.4 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.FltTime 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：

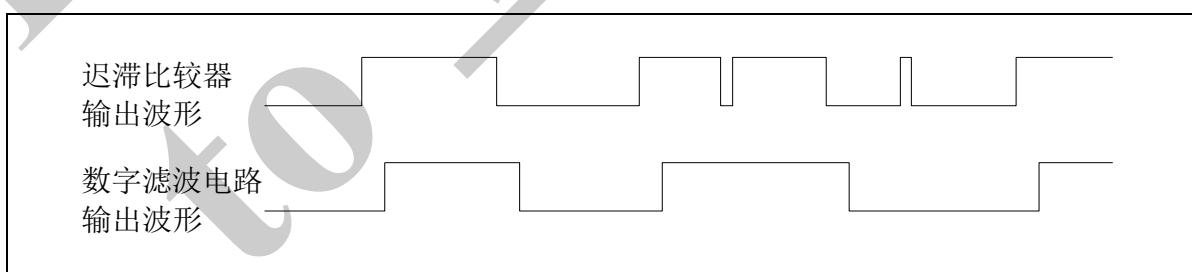


图 21-3 LVD 滤波输出

## 21.5 配置示例

### 21.5.1 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：

Step1：按 GPIO 章节描述，配置待监测的电压来源的管脚为模拟功能。

注：如待监测电压来源为 AVCC 则忽略本步骤。

Step2：配置 LVD\_CR.Source，选择待监测的电压来源。

Step3：配置 LVD\_CR.VTDS，选择 LVD 的阈值电压。

Step4：配置 LVD\_CR.FltTime，选择 LVD 滤波时间。

Step5：配置 LVD\_CR.FltEN，使能 LVD 滤波。

Step6：设置 LVD\_CR.HTEN 为 1，选择高电平触发 LVD 动作。

Step7：设置 LVD\_CR.ACT 为 1，选择 LVD 动作为复位。

Step8：设置 LVD\_CR.EN 为 1，使能 LVD。

注意：用户程序初始化时，读取复位标识寄存器可获知本次复位源是否为 LVD。

### 21.5.2 LVD 配置为电压变化中断

在本模式下，监测到电压相对于阈值点发生变化时产生中断。即当电压从低于阈值点上升到高于阈值点或从高于阈值点下降到低于阈值点时产生中断。

配置方法如下所示：

Step1：按 GPIO 章节描述，配置待监测的电压来源的管脚为模拟功能。

注：如待监测电压来源为 AVCC 则忽略本步骤。

Step2：配置 LVD\_CR.Source，选择待监测的电压来源。

Step3：配置 LVD\_CR.VTDS，选择 LVD 的阈值电压。

Step4：配置 LVD\_CR.FiltTime，选择 LVD 滤波时间。

Step5：配置 LVD\_CR.FltEn，使能 LVD 滤波。

Step6：设置 LVD\_CR.RTEN 和 LVD\_CR.FTEN 为 1，选择电平变化触发 LVD 动作。

Step7：设置 LVD\_CR.ACT 为 0，选择 LVD 动作为中断。

Step8：设置 LVD\_CR.IE 为 1，使能 LVD 中断。

Step9：使能 NVIC 中断向量表中的 LVD 中断。

Step10：设置 LVD\_CR.EN 为 1，使能 LVD。

Step11：在 LVD 的中断服务程序中执行用户需要进行的操作；退出中断服务程序前向 LVD\_IFR 写入 0x00 以清除中断标志。

## 21.6 LVD 寄存器

基地址 0x40002680

寄存器	偏移地址	描述
LVD_CR	0x000	LVD 配置寄存器
LVD_IFR	0x004	LVD 中断标志寄存器

表 21-1 LVD 寄存器

Preliminary  
to be confirmed

### 21.6.1 LVD 配置寄存器 (LVD\_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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	HTEN	RTEN	FTEN	FltTime		FltEn	VTDS		Source	ACT		EN			
RW	RW	RW	RW	RW		RW	RW		RW	RW		RW	RW		

位	标记	功能描述
31:16	Reserved	保留
15	IE	LVD 中断使能 1: 使能; 0: 禁止。
14	HTEN	高电平触发使能 (被监测电压低于阈值电压) 1: 使能; 0: 禁止。
13	RTEN	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) 1: 使能; 0: 禁止。
12	FTEN	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) 1: 使能; 0: 禁止。
11:9	FltTime	数字滤波时间配置(滤波时钟为RCL时钟) 111: 滤波宽度小于4095个时钟周期 110: 滤波宽度小于1023个时钟周期 101: 滤波宽度小于255个时钟周期 100: 滤波宽度小于63个时钟周期 011: 滤波宽度小于15个时钟周期 010: 滤波宽度小于7个时钟周期 001: 滤波宽度小于3个时钟周期 000: 滤波宽度小于1个时钟周期 注: 滤波时间仅在FTEN为1时有效。
8	FltEn	数字滤波使能配置 1: 使能数字滤波 0: 禁止数字滤波 注: 使能数字滤波时, 系统将自动使能RCL时钟。

7:4	VTDS	LVD 阈值电压选择 1111: 3.3v 1110: 3.2v 1101: 3.1v 1100: 3.0v 1011: 2.9v 1010: 2.8v 1001: 2.7v 1000: 2.6v 0111: 2.5v 0110: 2.4v 0101: 2.3v 0100: 2.2v 0011: 2.1v 0010: 2.0v 0001: 1.9v 0000: 1.8v
3:2	Source	LVD 监测来源选择 11: PD04端口输入电压 10: PC03端口输入电压 01: PA03端口输入电压 00: AVCC 电源电压
1	ACT	LVD 触发动作选择 1: 系统复位 0: NVIC中断
0	EN	LVD 使能控制 1: 使能LVD 0: 禁止LVD

## 21.6.2 LVD 中断寄存器 (LVD\_IFR)

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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														FLTV	INTF
Reserved														RO	RW0

位	标记	功能描述
31:2	Reserved	保留
1	FLTV	数字滤波器输出的电平值 1: 数字滤波器输出高电平 0: 数字滤波器输出低电平
0	INTF	LVD 中断标志 1: 已产生LVD中断标志, 写0清零 0: 未产生LVD中断标志

## 22 器件电子签名

电子签名存放在闪存（Flash）的系统存储区域，包含的芯片识别信息。用户固件或外部设备可以读取电子签名，用以匹配不同配置的 HC32Fxxx / HC32Lxxx 微控制器。

### 22.1 产品唯一身份标识码（UID）寄存器

唯一身份标识码典型应用场景：

- 用作序列号
- 对内部闪存编程时，将其用作密钥或加密原语以提高代码的安全性
- 激活安全自举过程等

80 比特的唯一设备标识符提供了一个对于任何设备和任何上下文都唯一的参考号码。用户永远不能改变这些位。80 位的唯一设备标识符也可以以单字节/半字/字等不同方式读取，然后使用自定义算法连接起来。

基址：0x0010 0980

偏移地址	功能描述	位域	备注
0	Y 坐标	UID[07:00]	0 – 127
1	X 坐标	UID[15:08]	0 – 127
2	Wafer 序号	UID[23:16]	1 - 25
3	LOT 序号 0	UID[31:24]	
4	LOT 序号 1	UID[39:32]	
5	LOT 序号 2	UID[47:40]	
6	RFU	UID[55:48]	
7	RFU	UID[63:56]	
8	RFU	UID[71:64]	
9	RFU	UID[79:72]	

## 22.2 产品型号寄存器

0x0010 0D90 ~ 0x0010 0DAF 存储了产品型号的 ASCII 码。如产品型号不足 32 字节，则以 0x00 进行填充。

例：484333246303032433450422D5453534F5032300000000000000000000000000000000 所代表的产品型号为 HC32F002C4PB-TSSOP20。

## 22.3 FLASH 容量寄存器

基地址：0x0010 0DB0

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

位	标记	功能描述
31:0	FlashSize	产品内置 Flash 的容量，以字节为单位 例：0x00004800 代表 Flash 容量为 18K Byte

## 22.4 RAM 容量寄存器

基地址: 0x0010 0DB4

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

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

位	标记	功能描述
31:0	RamSize	产品内置 RAM 的容量, 以字节为单位 例: 0x00000800 代表 RAM 容量为 2K Byte

## 22.5 管脚数量寄存器

基地址: 0x0010 08AA

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

PinCount[15:0]															
R															

位	标记	功能描述
15:0	PinCount	产品管脚的数量, 以只为单位 例: 0x0014 代表产品管脚数量为 20

## 22.6 一次性编程区域（OTP 区域）

芯片内具有 128 字节的 OTP 存储区域，其地址范围为 0x00100F00 – 0x00100F7F。该数据区域只可通过 ISP 指令写入数据，且仅能写入一次。片擦、页擦操作均不能擦除该区域内的数据。用户程序通过读操作即可读出该区域内的数据。

基于 OTP 区域的特性，用户程序可以在该 OTP 区域存储不可更改的信息。

Preliminary  
to be confirmed

## 23 SWD 调试接口

HC32F002 系列使用 ARM Cortex-M0+内核，该内核具有硬件调试模块 SWD，支持复杂的调试操作。硬件调试模块允许内核在取指(指令断点)或访问数据(数据断点)时停止。内核停止时，内核的内部状态和系统的外部状态都可以在 IDE 中进行查询。完成查询后，内核和外设可以被复原，程序将继续执行。当 HC32F002 微控制器连接到调试器并开始调试时，调试器将使用内核的硬件调试模块进行调试操作。

注意：

- SWD 在 DeepSleep 模式下不能工作，请在 Active 和 Sleep 模式下进行调试操作。

### 23.1 SWD 调试附加功能

本产品使用了 ARM Cortex-M0+ CPU，该内核包含用于高级调试功能的硬件扩展，因此本产品所拥有的调试功能与 Cortex-M0+一致。调试扩展允许内核可以在取指(指令断点)或取访问数据(数据断点)时停止内核。内核停止时，可以查询内核的内部状态和系统的外部状态。查询完成后，将恢复内核和系统并恢复程序执行。

当调试主机与 MCU 相连并进行调试时，将使用调试功能。

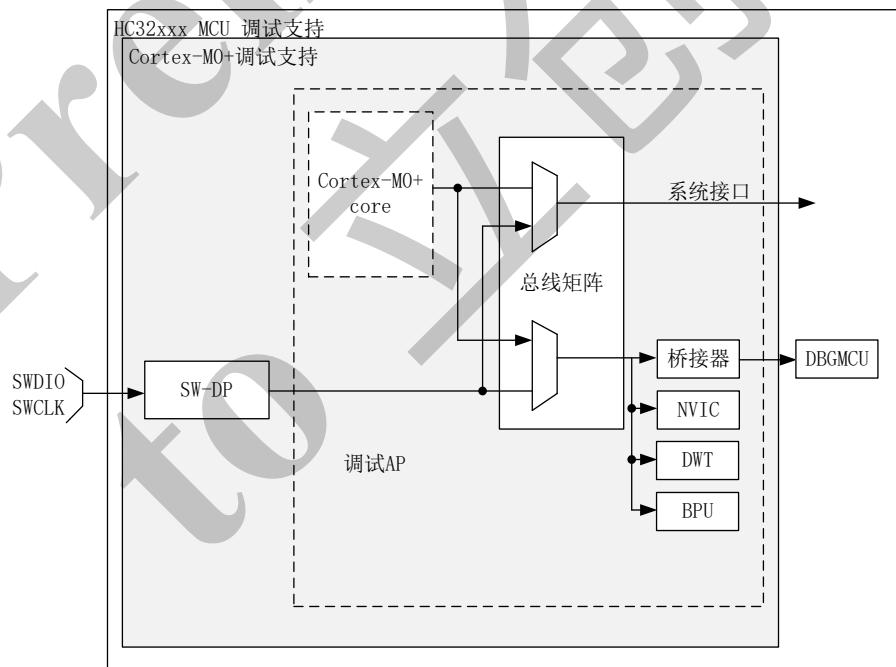


图 23-1 调试支持框图

Cortex®-M0+ 内核中内置的调试功能是 ARM® CoreSight 设计套件的一部分。

ARM® Cortex®-M0+内核提供集成片上调试支持。它包括：

- SW-DP：串行线
- BPU：断点单元
- DWT：数据观察点触发

注：

- 有关 ARM® Cortex®-M0+ 内核支持的调试功能的详细信息，请参见 Cortex®-M0+ 技术参考手册。

## 23.2 ARM® 参考文档

- Cortex®-M0+技术参考手册 (TRM)  
可从 [developer.arm.com](http://developer.arm.com) 获取。
- ARM® 调试接口 V5
- ARM® CoreSight 设计套件版本 r1p1 技术参考手册

Preliminary  
to be confirmed

## 23.3 调试端口管脚

### 23.3.1 SWD 端口管脚

HC32F002 系列的 SWD 端口，需要用到 2 个管脚，如下表所示。

SWD 端口名称	调试功能	管脚分配
SWCLK	串行时钟	PD01
SWDIO	串行数据输入/输出	PC07

### 23.3.2 SW-DP 管脚分配

如果烧录代码时使能了【加密芯片】选项，则上电后 SWD 调试功能被禁止。如果烧录程序时没有使能【加密芯片】选项，则上电后 PC07/PD01 管脚均被初始化为可被调试器使用的专用管脚。用户可设置 SysCtrl\_CR1.SWDIO 寄存器来禁止 SWD 管脚的调试功能，SWD 管脚将被释放以用作普通 GPIO。SWD 管脚的配置与功能小结如下表所示：

【加密芯片】选项	SWDIO 配置	PC07/PD01 功能
不加密	0	SWD
不加密	1	GPIO
1 级加密	0	SWD (仅可实现解密)
1 级加密	1	GPIO
2-3 级加密	0	NA
2-3 级加密	1	GPIO

### 23.3.3 SWD 管脚上的内部上拉

用户软件释放 SWDIO 后，GPIO 控制器便会控制这些管脚。GPIO 控制寄存器的复位状态会将 I/O 置于等效的状态：

- SWDIO：输入上拉
- SWCLK：输入上拉

由于内置上拉和下拉电阻，因此无需添加外部电阻。

## 23.4 SWD 端口

### 23.4.1 SWD 协议简介

此同步串行协议使用两个管脚：

- SWCLK：从主机到目标的时钟
- SWDIO：双向

利用该协议，可以同时读取和写入两组寄存器组（DPACC 寄存器组和 APACC 寄存器组）。传输数据时，LSB 在前。

对于 SWDIO 双向管理，必须在电路板上对线路进行上拉（ARM® 建议采用 100 K）。这些上拉电阻可在内部配置。无需外部上拉电阻。

每次在协议中更改 SWDIO 的方向时，都会插入转换时间，此时线路即不受主机驱动也不受目标驱动。默认情况下，此转换时间为一位时间，但可以通过配置 SWCLK 频率来调整。

### 23.4.2 SWD 协议序列

每个序列包括三个阶段：

1. 主机发送的数据包请求（8 位）
2. 目标发送的确认响应（3 位）
3. 主机或目标发送的数据传输阶段（33 位）

位	名称	说明
0	启动	必须为1
1	APnDP	0: DP 访问； 1: AP 访问
2	RnW	0: 写请求； 1: 读请求
4:3	A[3:2]	DP 或 AP 寄存器的地址字段
5	奇偶校验	前面几位的单位奇偶校验
6	停止	0
7	驻留	不受主机驱动。由于存在上拉，因此必须由目标读为 1

有关 DPACC 和 APACC 寄存器的详细说明, 请参见 Cortex®-M0+ TRM。

数据包请求后面始终为转换时间 (默认 1 位), 此时主机和目标都不会驱动。

位	名称	说明
0	ACK	001: FAULT 010: WAIT 100: OK

仅当发生 READ 事务或者接收到 WAIT 或 FAULT 确认时, ACK 响应后才必须是转换时间。

位	名称	说明
0:31	WDATA 或 RDATA	写入或读取数据
32	奇偶校验	32 个数据位的单奇偶校验

仅当发生 READ 事务时, DATA 传输后才必须是转换时间。

### 23.4.3 SW-DP 状态机 (复位、空闲状态、ID 代码)

SW-DP 的状态机有一个用于标识 SW-DP 的内部 ID 代码。该代码符合 JEP-106 标准。此 ID 代码是默认的 ARM® 代码, 设置为 **0x0BB11477** (相当于 Cortex®-M0+)。

注意:

- 在目标读取此 ID 代码前, SW-DP 状态机是不工作的。
- 在上电复位后或者线路处于高电平超过 50 个周期后, SW-DP 状态机处于复位状态。
- 如果在复位状态后线路处于低电平至少两个周期, SW-DP 状态机处于空闲状态。
- 复位状态后, 该状态机必须首先进入空闲状态, 然后对 DP-SW ID CODE 寄存器执行读访问。否则, 目标将在另一个事务上发出 FAULT 确认响应。

有关 SW-DP 状态机的更多详细信息, 请参见 Cortex®-M0+ TRM 和 CoreSight 设计套件 *r1p0TRM*。

### 23.4.4 DP 和 AP 读/写访问

- 不延迟对 DP 的读访问: 可以立即发送目标响应 (如果 ACK=OK), 也可以延迟发送目标响应 (如果 ACK=WAIT)。
- 延迟对 AP 的读访问。这意味着会在下次传输时返回访问结果。如果要执行的下

次访问不是 AP 访问，则必须读取 DP-RDBUFF 寄存器来获取结果。

- 每次进行 AP 读访问或 RDBUFF 读请求时都会更新 DP-CTRL/STAT 寄存器的 READOK 标志，以便了解 AP 读访问是否成功。
- SW-DP 有写缓冲区（用于 DP 或 AP 写入），这样即使在其它操作仍未完成时，也可以接受写入操作。如果写缓冲区已满，则目标确认响应为 WAIT。但 IDCODE 读取、CTRL/STAT 读取或 ABORT 写入除外，这几项操作在写缓冲区已满时也会被接受。
- 由于存在异步时钟域 SWCLK 和 HCLK，因此写操作后（奇偶校验位后）还需要两个额外的 SWCLK 周期，以使写入操作在内部生效。应在将线路驱动为低电平时（空闲状态）应用这些周期。

在写 CTRL/STAT 寄存器以提出一个上电请求时，这一点特别重要。否则下一个操作（在内核上电后才有效的操作）会立即执行，这将导致失败。

#### 23.4.5 SW-DP 寄存器

当 APnDP=0 时能够访问这些寄存器：

A[3:2]	RW	SELECT 寄存器的 CTRLSEL 位	寄存器	注释
00	读取		IDCODE	制造商代码设置为 Cortex®-M0+ 的默认 ARM® 代码。 <b>0x0BB11477</b> (标识 SW-DP)
00	写		ABORT	
01	读/写	0	DP-CTRL/STAT	目的： – 请求系统或调试上电 – 配置 AP 访问的传输操作 – 控制比较和验证操作 – 读取一些状态标志（上溢和上电确认）
01	读/写	1	WIRE CONTROL	用于配置物理串行端口协议（如转换时间的持续时间）
10	读取		READ RESEND	允许从已损坏的调试软件传输中恢复读取数据，无需重复执行原始 AP 传输。
10	写		SELECT	用于选择当前访问端口和活动的 4 字寄存器窗口
11	读/写		READ BUFFER	由于已发出 AP 访问，因此该读缓冲区非常有用（在执行下个 AP 事务时提供读取 AP 请求的结果）。此读取缓冲区捕获 AP 中的数据，显示为

前一次读取的结果，无需启动新操作。

### 23.4.6 SW-AP 寄存器

当 APnDP=1 时能够访问这些寄存器：

有多个 AP 寄存器，这些寄存器按以下组合进行寻址：

- 移位值 A[3:2]
- DP SELECT 寄存器的当前值

地址	A[3:2] 值	说明
0x0	00	保留，必须保持复位值。
0x4	01	DP CTRL/STAT 寄存器。用于： – 请求系统或调试上电 – 配置 AP 访问的传输操作 – 控制比较和验证操作 – 读取一些状态标志（上溢和上电确认）
0x8	10	DP SELECT 寄存器：用于选择当前访问端口和活动的 4 字寄存器窗口。 – 位 31:24: APSEL：选择当前 AP (select the current AP) – 位 23:8：保留 – 位 7:4: APBANKSEL：在当前 AP 上选择活动的 4 字寄存器窗口 – 位 3:0：保留
0xC	11	DP RDBUFF 寄存器：用于通过调试器在执行一系列操作后获取最后结果

## 23.5 内核调试

通过内核调试寄存器调试内核。通过调试访问端口调试访问这些寄存器。它由四个寄存器组成：

寄存器	说明
DHCSR	32 位调试停止控制和状态寄存器 此寄存器提供有关处理器状态的信息，能够使内核进入调试停止状态并提供处理器步进功能。
DCRSR	17 位调试内核寄存器选择器寄存器： 此寄存器选择需要进行读写操作的处理器寄存器。
DCRDR	32 位调试内核寄存器数据寄存器： 此寄存器保存在寄存器与 DCRSR（选择器）寄存器选择的处理器之间读取和写入的数据。
DEMCR	32 位调试异常和监视控制寄存器： 此寄存器提供向量捕获和调试监视控制。

这些寄存器在系统复位时不复位。它们只能通过上电复位来复位。有关更多详细信息，请参见 Cortex®-M0+ TRM。

为了在复位后立即使内核进入调试停止状态，必须：

- 使能调试和异常监视控制寄存器的位 0 (VC\_CORRESET)
- 使能调试停止控制和状态寄存器的位 0 (C\_DEBUGEN)

## 23.6 BPU (断点单元)

Cortex®-M0+ BPU 实现提供四个断点寄存器。

### 23.6.1 BPU 功能

处理器断点实现了基于 PC 的断点功能。

有关 BPU CoreSight 标识寄存器及其地址和访问类型的更多信息，请参见 ARMv6-M ARM® 和 ARM® CoreSight 组件技术参考手册。

## 23.7 DWT (数据观察点)

Cortex®-M0 + DWT 实现提供了一个观察点寄存器组。

### 23.7.1 DWT 功能

处理器观察点实现了数据地址和基于 PC 的观察点功能（即 PC 采样寄存器），并支持比较器地址掩码，如 ARMv6-M ARM® 中所述。

### 23.7.2 DWT 程序计数器采样寄存器

实现数据观察点单元的处理器还实现了 ARMv6-M 可选 DWT 程序计数器采样寄存器(DWT\_PCSR)。此寄存器允许调试程序定期采样 PC，无需停止处理器。这可提供粗略分析。有关更多信息，请参见 ARMv6-M ARM®。

Cortex®-M0+ DWT\_PCSR 记录通过条件代码和指令以及未通过条件代码的指令。

## 23.8 调试组件 (DBG)

MCU 调试组件帮助调试器为以下各项提供支持:

- 低功耗模式
- 断点期间的定时器、看门狗的时钟控制

### 23.8.1 对低功耗模式的调试支持

要进入低功耗模式 (Sleep、DeepSleep)，必须执行指令 WFI 或 WFE。

MCU 支持多个低功耗模式，这些模式可以禁止 CPU 时钟或降低 CPU 功耗。

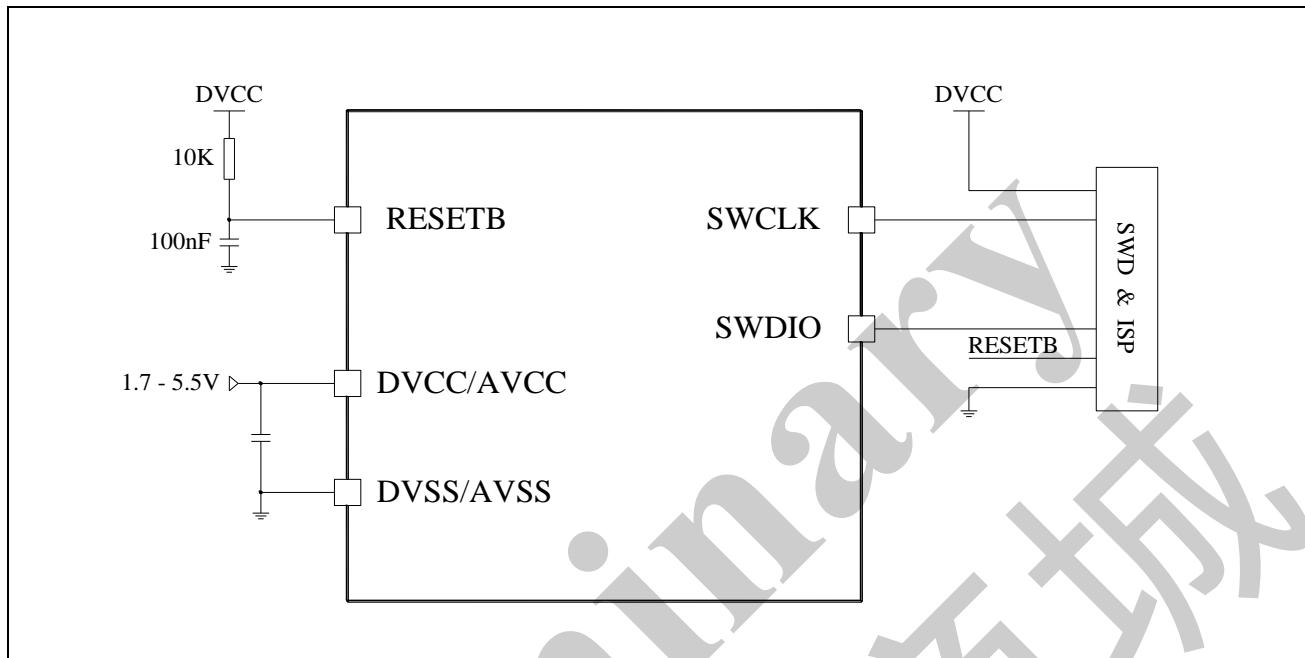
内核不允许在调试会话期间关闭 FCLK 或 HCLK。由于调试期间需要使用它们进行调试连接，因此其必须保持激活状态。MCU 集成了特殊方法，允许用户在低功耗模式下调试软件。

### 23.8.2 对定时器、看门狗的调试支持

断点期间，必须选择定时器和看门狗的计数器的行为方式:

- 在产生断点时，计数器继续计数。例如，当 PWM 控制电机时，通常需要这种方式。
- 在产生断点时，计数器停止计数。用于看门狗时需要这种方式。

## 24 典型应用电路图



注意：

- AVCC 与 DVCC 电压必须相同。
- 每组电源都需要添加去耦电容，去耦电容尽量靠近相应电源管脚。

## 25 电气特性

### 25.1 测试条件

除非特别说明，所有电压的都以 VSS 为基准。

#### 25.1.1 最小和最大数值

除非特别说明，在生产线上通过对 100% 的产品在环境温度  $TA=25^{\circ}\text{C}$  和  $TA=TA_{\max}$  下执行的测试( $TA_{\max}$  与选定的温度范围匹配)，所有最小和最大值将在最坏的环境温度、供电电压和时钟频率条件下得到保证。

在每个表格下方的注解中说明为通过综合评估、设计模拟和/或工艺特性得到的数据，不会在生产线上进行测试；在综合评估的基础上，最小和最大数值是通过样本测试后，取其平均值再加减三倍的标准分布(平均 $\pm 3\Sigma$ )得到。

#### 25.1.2 典型数值

除非特别说明，典型数据是基于  $TA=25^{\circ}\text{C}$  和  $VCC=3.3\text{V}$ 。这些数据仅用于设计指导而未经测试。典型的 ADC 精度数值是通过对一个标准的批次采样，在所有温度范围下测试得到，95% 产品的误差小于等于给出的数值(平均 $\pm 2\Sigma$ )。

## 25.2 绝对最大额定值

加在器件上的载荷如果超过“绝对最大额定值”列表中给出的值，可能会导致器件永久性地损坏。这里只是给出能承受的最大载荷，并不意味在此条件下器件的功能性操作无误。器件长期工作在最大值条件下会影响器件的可靠性。

符号	描述	最小值	最大值	单位
VCC - VSS	外部主供电电压(包含AVCC和DVCC) <sup>(1)</sup>	-0.3	5.5	V
V <sub>IN</sub>	在其它引脚上的输入电压 <sup>(2)</sup>	VSS - 0.3	VCC + 0.3	V
ΔVCCx	不同供电引脚之间的电压差		50	mV
V <sub>SSX</sub> - V <sub>SS</sub>	不同接地引脚之间的电压差		50	mV
V <sub>ESD(HBM)</sub>	ESD静电放电电压(人体模型)	参考绝对最大值电气参数		V

表 25-1 电压特性

1. 所有的电源(DVCC,AVCC)和地(DVSS, AVSS)引脚必须始终连接到外部允许范围内的供电系统上。
2. I<sub>INJ(PIN)</sub>绝对不可以超过它的极限，即保证 V<sub>IN</sub> 不超过其最大值。如果不能保证 V<sub>IN</sub> 不超过其最大值，也要保证在外部限制 I<sub>INJ(PIN)</sub>不超过其最大值。当 V<sub>IN</sub>>VCC 时，有一个正向注入电流；当 V<sub>IN</sub><VSS 时，有一个反向注入电流。

符号	描述	最大值 <sup>(1)</sup>	单位
I <sub>VCC</sub>	经过DVCC/AVCC电源线的总电流(供应电流) <sup>(1)</sup>	300	mA
I <sub>VSS</sub>	经过VSS地线的总电流(流出电流) <sup>(1)</sup>	300	mA
I <sub>IO</sub>	任意I/O和控制引脚上的输出灌电流	25	mA
	任意I/O和控制引脚上的输出电流	-25	mA
I <sub>INJ(PIN)</sub> <sup>(2)(3)</sup>	RESETB引脚的注入电流	+/-5	mA
	XTH的XTHI引脚和XTL的XTLI引脚的注入电流	+/-5	mA
	其他引脚的注入电流 <sup>(4)</sup>	+/-5	mA
$\sum I_{INJ(PIN)}$ <sup>(2)</sup>	所有I/O和控制引脚上的总注入电流 <sup>(4)</sup>	+/-25	mA

表 25-2 电流特性

1. 所有的电源(DVCC,AVCC)和地(DVSS, AVSS)引脚必须始终连接到外部允许范围内的供电系统上。
2. I<sub>INJ(PIN)</sub>绝对不可以超过它的极限，即保证 V<sub>IN</sub> 不超过其最大值。如果不能保证 V<sub>IN</sub> 不超过其最大值，也要保证在外部限制 I<sub>INJ(PIN)</sub>不超过其最大值。当 V<sub>IN</sub>>VCC 时，有一个正向注入电流；当 V<sub>IN</sub><VSS 时，有一个反向注入电流。
3. 反向注入电流会干扰器件的模拟性能。
4. 当几个 I/O 口同时有注入电流时， $\sum I_{INJ(PIN)}$ 的最大值为正向注入电流与反向注入电流的即时绝对值之和。该结果基于在器件 4 个 I/O 端口上 $\sum I_{INJ(PIN)}$ 最大值的特性。

符号	描述	数值	单位
TSTG	储存温度范围	-60 ~ + 150	°C
TJ	最大结温度	125	°C

表 25-3 温度特性

Preliminary  
to IT尙城

## 25.3 工作条件

### 25.3.1 通用工作条件

符号	参数	条件	最小值	最大值	单位
$f_{HCLK}$	内部AHB时钟频率		0	48	MHz
$f_{PCLK0}$	内部APB0时钟频率		0	48	MHz
$f_{PCLK1}$	内部APB1时钟频率		0	48	MHz
DVCC	数字部分工作电压		1.7	5.5	V
AVCC <sup>(1)</sup>	模拟部分工作电压	必须与DVCC <sup>(2)</sup> 相同	1.7	5.5	V
PD	功率耗散 TA=105°C	TSSOP20		283	mW
	功率耗散 TA=105°C	TSSOP24		291	mW
TA	环境温度	最大功率消耗	-40	105	°C
		低功率消耗 <sup>(3)</sup>	-40	125	°C
TJ	结温度范围		-40	125	°C

表 25-4 通用工作条件

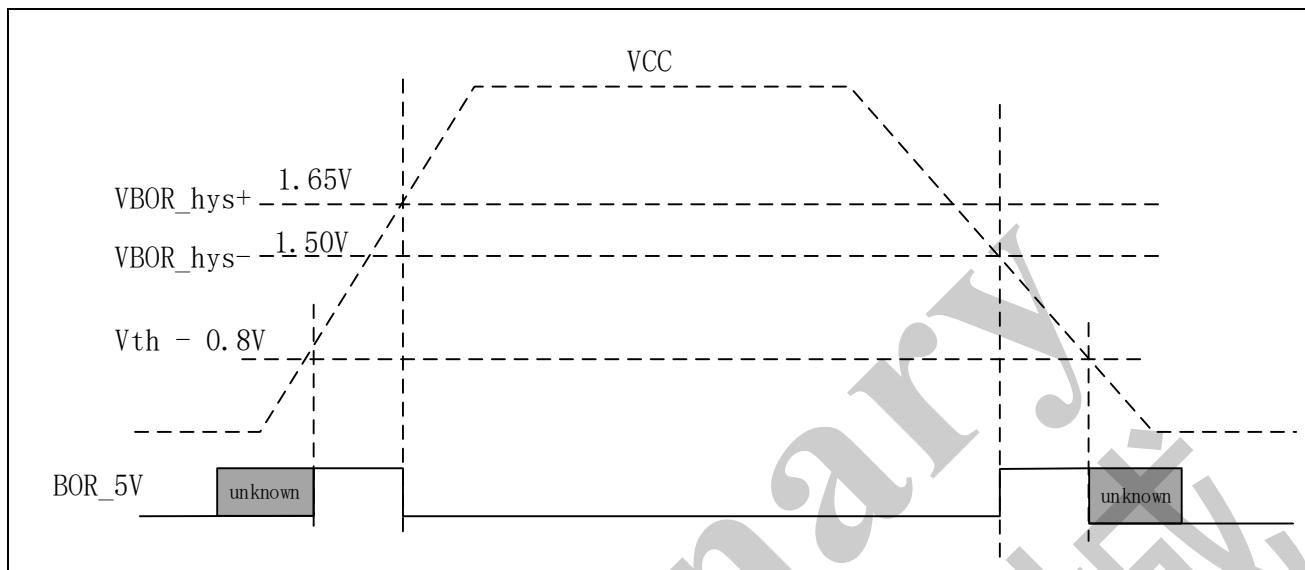
1. 当使用 ADC 时, 参见 ADC 电气参数。
2. 建议使用相同的电源为 DVCC 和 AVCC 供电, 在上电和正常操作期间, DVCC 和 AVCC 之间最多允许有 300mV 的差别。
3. 在较低的功率耗散的状态下, 只要  $T_J$  不超过  $T_{Jmax}$ ,  $T_A$  可以扩展到这个范围。

### 25.3.2 上电和掉电时的工作条件

符号	参数	条件	最小值	最大值	单位
$t_{V_{CC}}$	VCC上升速率		0	5	V/μs
$t_{V_{CC}}$	VCC下降速率		0	5	V/μs

表 25-5 上电和掉电的工作条件

### 25.3.3 内嵌复位和 LVD 模块特性



1. 设计保证, 不在生产中测试。

图 25-1 POR/Brown Out 示意图

符号	参数	条件	最小值	典型值	最大值	单位
Vpor	POR 释放电压 (上电过程) BOR 检测电压 (掉电过程)		1.45	1.50	1.65	V

表 25-6 POR/Brown Out

符号	参数	条件	最小值	典型值	最大值	单位
Vex	外部输入电压范围		0		VCC	V
Vlevel	检测阈值	LVD_CR.VTDS=0000 LVD_CR.VTDS=0001 LVD_CR.VTDS=0010 LVD_CR.VTDS=0011 LVD_CR.VTDS=0100 LVD_CR.VTDS=0101 LVD_CR.VTDS=0110 LVD_CR.VTDS=0111 LVD_CR.VTDS=1000 LVD_CR.VTDS=1001 LVD_CR.VTDS=1010 LVD_CR.VTDS=1011 LVD_CR.VTDS=1100 LVD_CR.VTDS=1101 LVD_CR.VTDS=1110 LVD_CR.VTDS=1111		1.8±3.5% 1.9±3.5% 2.0±3.5% 2.1±3.5% 2.2±3.5% 2.3±3.5% 2.4±3.5% 2.5±3.5% 2.6±3.5% 2.7±3.5% 2.8±3.5% 2.9±3.5% 3.0±3.5% 3.1±3.5% 3.2±3.5% 3.3±3.5%		V
Icomp	功耗			0.36		uA
Tresponse	响应时间			100		μs
Tsetup	建立时间			300		μs
Vhyste	迟滞电压			40		mV
Tfilter	滤波时间	LVD_CR.FltTime = 000 LVD_CR.FltTime = 001 LVD_CR.FltTime = 010 LVD_CR.FltTime = 011 LVD_CR.FltTime = 100 LVD_CR.FltTime = 101 LVD_CR.FltTime = 110 LVD_CR.FltTime = 111		40 100 220 460 2000 7800 31200 125000		μs

表 25-7 LVD 模块特性

### 25.3.4 内置的参考电压

符号	参数	条件	最小值	典型值	最大值	单位
$V_{REF09}$	Internal 0.9V Reference Voltage	常温25°C; 3.3V		0.9		V
$V_{REF09}$	Internal 0.9V Reference Voltage	-40~105°C; 1.7~5.5V	0.885		0.915	V
$T_{Coeff}$	Internal 0.9V temperature coefficient	-40 ~ 105°C		50		ppm/ °C

1. 数据基于考核结果, 不在生产中测试

### 25.3.5 供电电流特性

电流消耗是多种参数和因素的综合指标, 这些参数和因素包括工作电压、环境温度、I/O 引脚的负载、产品的软件配置、工作频率、I/O 脚的翻转速率、程序在存储器中的位置以及执行的代码等。

微控制器处于下列条件:

- 所有的 I/O 引脚都处于输入模式, 并连接到 VCC 或 VSS(无负载)。
- 所有的外设都处于关闭状态, 除非特别说明。
- 闪存存储器的访问时间调整到  $f_{HCLK}$  的频率(0~24MHz 时为 0 个等待周期, 24~48MHz 时为 1 个等待周期)。
- 当开启外设时:  $f_{PCLK0} = f_{HCLK}$ ,  $f_{PCLK1} = f_{HCLK}$ 。

Symbol	Parameter	Conditions		Typ <sup>(1)</sup>	Max <sup>(2)</sup>	Unit
IDD (Run in RAM)	All peripherals clock ON, Run while(1) in RAM	VCC=3.3V TA=25°C	RCH clock source	4M	505	uA
	8M			730		
	12M			955		
	22.12M			1515		
	24M			1630		
	44.24M			2755		
	48M			2980		
	All peripherals clock OFF, Run while(1) in RAM			4M	410	
	RCH clock source		8M	545	uA	
			12M	680		
			22.12M	1005		
			24M	1080		
			44.24M	1735		
			48M	1870		
				4M	725	870
						uA

IDD (Run mode)	All peripherals clock ON, Run while(1) in Flash	VCC=1.7~5.5V TA=-40~105°C	RCH clock source	8M	1160	1350	
				12M	1575	1820	
				22.12M	2545	2960	
				24M	2735	3170	
	All peripherals clock OFF, Run while(1) in Flash	VCC=1.7~5.5V TA=-40~105°C	RCH clock source	44.24M FlashWait=1	3515	4070	uA
				48M FlashWait=1	3790	4380	
				4M	635	770	
				8M	970	1140	
IDD (Sleep mode)	All peripherals clock ON	VCC=1.7~5.5V TA=-40~105°C	RCH clock source	12M	1295	1510	uA
				22.12M	2035	2390	
				24M	2185	2550	
				44.24M FlashWait=1	2500	2930	
	All peripherals clock OFF	VCC=1.7~5.5V TA=-40~105°C	RCH clock source	48M FlashWait=1	2685	2950	uA
				4M	400	470	
				8M	520	610	
				12M	635	740	
IDD (LP Run)	All peripherals clock ON, Run while(1) in Flash	VCC=1.7~5.5V	RCL32K clock source	22.12M	925	1070	uA
				24M	990	1140	
				44.24M FlashWait=1	1570	1800	
				48M FlashWait=1	1695	1930	
	All peripherals clock OFF,	VCC=1.7~5.5V		4M	305	370	uA
				8M	335	400	
				12M	360	430	
				22.12M	415	490	
				24M	435	520	
				44.24M FlashWait=1	550	640	
				48M FlashWait=1	585	680	

	Run while(1) in Flash		RCL32K clock source	TA=85°C	8.5	14.9	
				TA=105°C	10.6	17.7	
IDD (LP Sleep)	All peripherals clock ON	VCC=1.7~5.5V	RCL32K clock source	TA=-40~25°C	4.4	10.3	uA
				TA=50°C	4.8	8.6	
				TA=85°C	6.1	10.4	
				TA=105°C	8.1	13.9	
	All peripherals clock OFF	VCC=1.7~5.5V	RCL32K clock source	TA=-40~25°C	3.7	7.3	uA
				TA=50°C	4.0	7.8	
				TA=85°C	5.3	9.9	
				TA=105°C	7.4	13.2	
IDD (DeepSleep mode)	WDT+LVD+ RCL32K +DeepSleep	VCC=1.7~5.5V		TA=-40~25°C	3.2	6.9	uA
				TA=50°C	3.7	7.5	
				TA=85°C	4.9	9.2	
				TA=105°C	6.7	12.3	
	LVD+RCL32K +DeepSleep	VCC=1.7~5.5V		TA=-40~25°C	2.6	6.2	uA
				TA=50°C	3.0	6.7	
				TA=85°C	4.0	8.7	
				TA=105°C	5.5	12.3	
	WDT+RCL32K +DeepSleep	VCC=1.7~5.5V		TA=-40~25°C	3.0	6.5	uA
				TA=50°C	3.4	7.0	
				TA=85°C	4.6	8.8	
				TA=105°C	6.3	11.7	
	IRC38K + DeepSleep	VCC=1.7~5.5V		TA=-40~25°C	3.0	6.5	uA
				TA=50°C	3.4	7.0	
				TA=85°C	4.6	8.8	
				TA=105°C	6.3	11.8	
	RCL32K + DeepSleep	VCC=1.7~5.5V		TA=-40~25°C	3.0	6.5	uA
				TA=50°C	3.4	7.0	
				TA=85°C	4.6	8.8	
				TA=105°C	6.3	11.9	
	DeepSleep	VCC=1.7~5.5V		TA=-40~25°C	2.4	5.8	uA
				TA=50°C	2.7	6.3	
				TA=85°C	3.8	8.2	
				TA=105°C	5.2	11.8	

- 若没有其他指定条件, 该 Typ 的值是在 25°C & V<sub>CC</sub> = 3.3V 测得。
- 若没有其他指定条件, 该 Max 的值是 V<sub>CC</sub> = 1.7~5.5 & Temperature = -40 ~ 105°C 范围内的最大值。
- 数据基于考核结果, 不在生产中测试

表 25-1 工作电流特性

### 25.3.6 从低功耗模式唤醒的时间

唤醒时间是在 RCH 振荡器的唤醒阶段测量得到。唤醒时使用的时钟源依当前的操作模式而定：

- 休眠模式：时钟源是 RCH 振荡器
- 深度休眠模式：时钟源是进入深度休眠时所使用的时钟是 RCH 振荡器

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$T_{wu}$	休眠模式唤醒时间			2.0		$\mu s$
	深度休眠唤醒时间	$F_{HCLK} = 4 \sim 48MHz$		5.0		$\mu s$

1. 唤醒时间的测量是从唤醒事件开始至用户程序读取第一条指令。

Preliminary  
to be confirmed

### 25.3.7 外部时钟源特性

#### 25.3.7.1 外部输入高速时钟

符号	参数	条件	最小值	典型值	最大值	单位
$f_{EXT}$	用户外部时钟频率 <sup>(1)</sup>		0	8	16	MHz
$V_{EXTH}$	输入引脚高电平电压		0.7VCC		VCC	V
$V_{EXTL}$	输入引脚低电平电压		VSS		0.3VCC	V
$T_{rEXT}$	上升的时间 <sup>(1)</sup>				20	ns
$T_{fEXT}$	下降的时间 <sup>(1)</sup>				20	ns
$T_{WEXT}$	输入高或低的时间 <sup>(1)</sup>		16			ns
$C_{inEXT}$	输入容抗 <sup>(1)</sup>			5		pF
Duty	占空比		40		60	%
$I_L$	输入漏电流				$\pm 1$	$\mu A$

1. 由设计保证, 不在生产中测试。

Preliminary and Subject to Change

## 25.3.8 内部时钟源特性

### 25.3.8.1 内部 RCH 振荡器

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Dev	RCH振荡器精度	User trimming step for given VCC and TA conditions		0.25		%
		VCC = 1.7 ~ 5.5V T <sub>AMB</sub> = -40 ~ 105°C	-2.5		+2.5	%
		VCC = 1.7 ~ 5.5V T <sub>AMB</sub> = -20 ~ 50°C	-1.5		+1.5	%
F <sub>CLK</sub>	振荡频率		40.0		48.0	MHz
I <sub>CLK</sub>	功耗	F <sub>MCLK</sub> = 48MHz		170		μA
DC <sub>CLK</sub>	占空比 <sup>(1)</sup>		45	50	55	%

1. 由综合评估得出, 不在生产中测试。

### 25.3.8.2 内部 RCL 振荡器

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Dev	RCL振荡器精度	User trimming step for given VCC and TA conditions		0.5		%
		VCC = 1.7 ~ 5.5V T <sub>AMB</sub> = -40 ~ 105°C	-5		+5	%
		VCC = 1.7 ~ 5.5V T <sub>AMB</sub> = -20 ~ 50°C	-3		+3	%
F <sub>CLK</sub>	振荡频率			38.4 32.8		KHz
T <sub>CLK</sub>	启动时间			150		μs
DC <sub>CLK</sub>	占空比 <sup>(1)</sup>		25	50	75	%
I <sub>CLK</sub>	功耗			1.5		μA

1. 由综合评估得出, 不在生产中测试。

### 25.3.9 存储器特性

符号	参数	条件	最小值	典型值	最大值	单位
EC <sub>FLASH</sub>	擦写次数	T <sub>AMB</sub> = 25°C	20K			Cycles
RET <sub>FLASH</sub>	数据保存期限	T <sub>AMB</sub> = 105°C	20			Years
		常温	100			Years
T <sub>b_prog</sub>	编程时间 (字节)		22		30	μs
T <sub>w_prog</sub>	编程时间 (字)		40		52	μs
T <sub>p_erase</sub>	页擦除时间		4		5	ms
T <sub>m_erase</sub>	整片擦除时间		30		40	ms

### 25.3.10 EFT 特性

芯片复位可以使系统恢复正常操作。

符号	级别/类型
EFT to IO (IEC61000-4-4)	TBD
EFT to Power (IEC61000-4-4)	TBD

#### 软件建议

软件的流程中必须包含应对程序跑飞的控制，如：

- 被破坏的程序计数器
- 意外的复位
- 关键数据被破坏（控制寄存器等）

在进行 EFT 测试时，可以把超出应用要求的干扰直接施加在芯片电源或 IO 上，当检测到意外动作的地方，软件部分进行加强以防止发生不可恢复的错误。

### 25.3.11 ESD 特性

使用特定的测量方法，对芯片进行强度测试以决定它的电气敏感性方面的性能。

符号	参数	条件	最小值	典型值	最大值	单位
$V_{ESD_{HBM}}$	ESD @ Human Body Mode			4		KV
$V_{ESD_{CDM}}$	ESD @ Charge Device Mode			1		KV
$V_{ESD_{MM}}$	ESD @ machine Mode			200		V
$I_{latchup}$	Latch up current			200		mA

### 25.3.12 I/O 端口特性

#### 25.3.12.1 输出特性——端口

符号	参数	条件	最小值	最大值	单位
$V_{OH}$	High level output voltage	Sourcing 4 mA, $V_{CC} = 3.3$ V (see Note 1)	$V_{CC}-0.25$		V
	Source Current	Sourcing 8 mA, $V_{CC} = 3.3$ V (see Note 2)	$V_{CC}-0.6$		V
$V_{OL}$	Low level output voltage	Sinking 5 mA, $V_{CC} = 3.3$ V (see Note 1)		$V_{SS}+0.25$	V
	Sink Current	Sinking 14 mA, $V_{CC} = 3.3$ V (see Note 2)		$V_{SS}+0.6$	V
$V_{OHD}$	High level output voltage	Sourcing 8 mA, $V_{CC} = 3.3$ V (see Note 1)	$V_{CC}-0.25$		V
	Double source Current	Sourcing 18 mA, $V_{CC} = 3.3$ V (see Note 2)	$V_{CC}-0.6$		V
$V_{OLD}$	Low level output voltage	Sinking 8 mA, $V_{CC} = 3.3$ V (see Note 1)		$V_{SS}+0.25$	V
	Double Sink Current	Sinking 18 mA, $V_{CC} = 3.3$ V (see Note 2)		$V_{SS}+0.6$	V

表 25-4 端口输出特性

- NOTES: 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 outputs combined, should not exceed 100 mA to satisfy the maximum specified voltage drop.

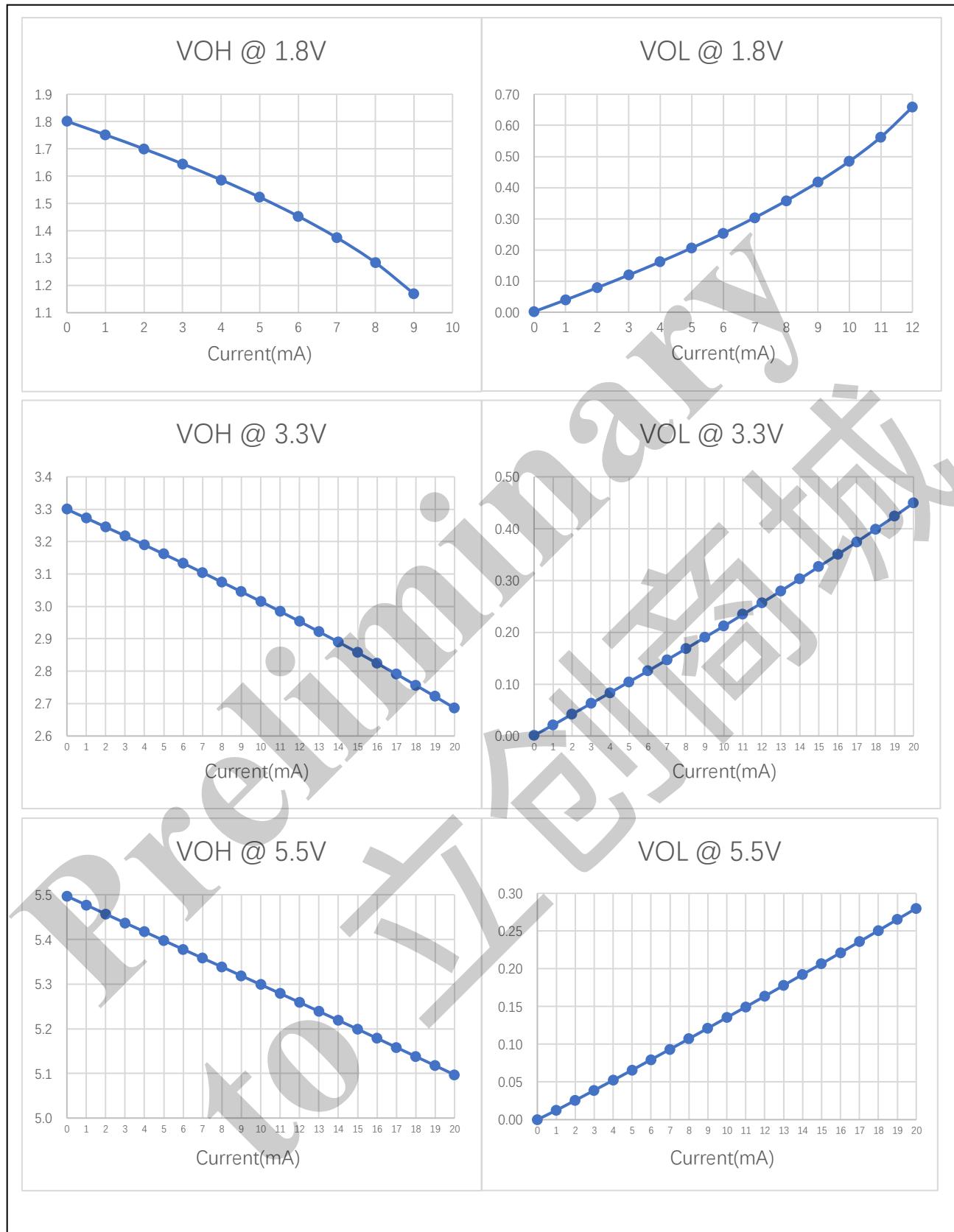


图 25-2 输出端口 VOH/VOL 实测曲线

### 25.3.12.2 输入特性——端口 PA,PB,PC,PD

符号	参数	条件	最小值	典型值	最大值	单位
$V_{IH}$	Positive-going input threshold voltage	VCC=1.8V	0.7VCC			V
		VCC=3.3V	0.7VCC			V
		VCC=5.5V	0.7VCC			V
$V_{IL}$	Negative-going input threshold voltage	VCC=1.8V			0.3VCC	V
		VCC=3.3V			0.3VCC	V
		VCC=5.5V			0.3VCC	V
$V_{hys(1)}$	Input voltage hysteresis ( $V_{IH} - V_{IL}$ )	VCC=1.8V		0.3		V
		VCC=3.3V		0.4		V
		VCC=5.5V		0.6		V
$R_{pullhigh}$	Pullup resistor	Pullup enabled VCC=3.3V		80		Kohm
$R_{pulllow}$	Pulldown resistor	Pulldown enabled VCC=3.3V		40		Kohm
$C_{input}$	Input capacitance			5		pF

1. 由综合评估得出, 不在生产中测试。

### 25.3.12.3 端口外部输入采样要求——Timer Clock

符号	参数	条件	最小值	典型值	最大值	单位
$t(int)$	External interrupt timing	External trigger signal for the interrupt flag (see Note 1)	1.8V	30		ns
			3.3V	30		ns
			5.5V	30		ns
$t(cap)$	Timer capture timing	TIMx capture pulse width Fsystem = 4MHz	1.8V	0.5		μs
			3.3V	0.5		μs
			5.5V	0.5		μs
$t(clk)$	Timer clock frequency applied to pin	TIMx external clock input Fsystem = 4MHz	1.8V		PCLK/2	MHz
			3.3V		PCLK/2	MHz
			5.5V		PCLK/2	MHz

NOTES: 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  $t_{(int)}$ .

2. 由综合评估得出, 不在生产中测试。

### 25.3.12.4 端口漏电特性——PA,PB,PC,PD

符号	参数	条件	最小值	典型值	最大值	单位
$I_{lkg(Px,y)}$	Leakage current	$V_{(Px,y)}$ (see Note 1,2)		$\pm 50$		nA

NOTES: 1. The leakage current is measured with VSS or VCC applied to the corresponding pin(s), unless otherwise noted.

2. The port pin must be selected as input.

### 25.3.13 RESETB 引脚特性

RESETB 引脚输入驱动使用 CMOS 工艺, 它连接了一个不能断开的上拉电阻。

符号	参数	条件	最小值	典型值	最大值	单位
$V_{IL(RESETB)}^{(1)}$	输入低电平电压		-0.3		0.3VCC	V
$V_{IH(RESETB)}$	输入高电平电压		0.7VCC		VCC+0.3	V
$V_{hys(RESETB)}$	施密特触发器电压迟滞			200		mV
$R_{PU}$	弱上拉等效电阻	$V_{IN} = V_{SS}$		80		K $\Omega$
$V_{F(RESETB)}^{(1)}$	输入滤波脉冲				100	ns
$V_{NF(RESETB)}^{(1)}$	输入非滤波脉冲		300			ns

1. 由设计保证, 不在生产中测试。

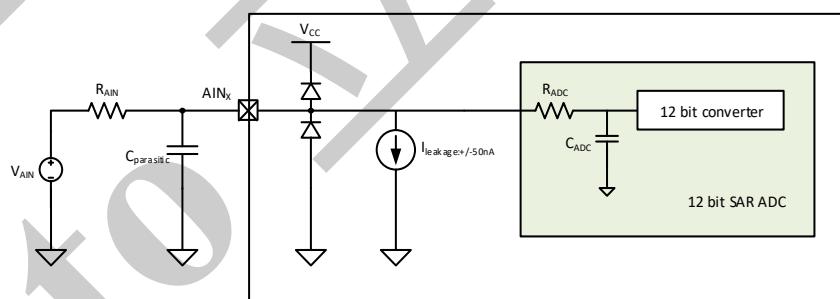
### 25.3.14 ADC 特性

符号	参数	条件	最小值	典型值	最大值	单位
$V_{ADCIN}$	Input voltage range	Single ended	0		$V_{ADCREFIN}$	V
$V_{ADCREFIN}$	Input range of external reference voltage	Single ended	0		AVCC	V
$I_{ADC1}$	Active current including reference generator and buffer	200Ksps		0.16		mA
$I_{ADC2}$	Active current without reference generator and buffer	1Msps		0.22		mA
$C_{ADCIN}$	ADC input capacitance			1.6	2	pF
$R_{ADC}^{(1)}$	ADC sampling switch impedance			5		KOhm
$R_{AIN}^{(1)}$	ADC external input resistor <sup>(2)</sup>				100	KOhm
$F_{ADCCLK}$	ADC clock Frequency				24	MHz
$T_{ADCSTART}$	Startup time of reference			5		$\mu$ s

	generator and ADC core					
T <sub>ADC</sub> CONV	Conversion time		19	24	25	cycles
ENOB	Effective Bits	1Msps@VCC>=2.7V 500Ksps@VCC>=2.4V 200Ksps@VCC>=1.7V REF=EXREF		9.5		Bit
		1Msps@VCC>=2.7V 500Ksps@VCC>=2.4V 200Ksps@VCC>=1.7V REF=VCC		9.4		Bit
SNR	Signal to Noise Ratio	1Msps@VCC>=2.7V 500Ksps@VCC>=2.4V 200Ksps@VCC>=1.7V REF=EXREF		63		dB
		1Msps@VCC>=2.7V 500Ksps@VCC>=2.4V 200Ksps@VCC>=1.7V REF=VCC		63		dB
DNL <sup>(1)</sup>	Differential non-linearity	500Ksps; VREF=EXREF/AVCC	-1		1	LSB
INL <sup>(1)</sup>	Integral non-linearity	500Ksps; VREF=EXREF/AVCC	-2		2	LSB
E <sub>o</sub> <sup>(1)</sup>	Offset error		-1		1	LSB
E <sub>g</sub> <sup>(1)</sup>	Gain error		-1		1	LSB

1. 由设计保证, 不在生产中测试。

2. ADC 的典型应用如下图所示:



对于 0.5LSB 采样误差精度要求的条件下, 外部输入阻抗的计算公式如下:

$$R_{AIN} = \frac{M}{F_{ADC} * C_{ADC} * (N + 1) * \ln(2)} - R_{ADC}$$

其中  $F_{ADC}$  为 ADC 时钟频率, 寄存器 ADC\_CR0[4:2] 可设定其与 PCLK 的关系, 如下

表。

下表为 ADC 时钟频率  $F_{ADC}$  和 PCLK 分频比关系:

ADC_CR0[4:2]	N
000	1
001	2
010	4
011	8
.....	
111	128

M 为采样周期个数, 由寄存器 ADC\_CR0[13:12]设定。

下表为采样时间  $t_{sa}$  和 ADC 时钟频率  $F_{ADC}$  的关系:

ADC_CR0[13:12]	M
00	6
01	8
10	11
11	12

下表为 ADC 时钟频率  $F_{ADC}$  和外部电阻  $R_{AIN}$  的关系 (M=12, 采样误差 0.5LSB 的条件下):

$R_{AIN}$ (KOhm)	$F_{ADC}$ (KHz)
10	24000
30	22000
50	14000
80	9000
100	7000
120	6000
150	5000

对于上述典型应用, 应注意:

- 尽量减小 ADC 输入端口  $AIN_X$  的寄生电容  $C_{PARACITIC}$ ;
- 除了考虑  $R_{AIN}$  值外, 如果信号源  $V_{AIN}$  的内阻较大时, 也需要加入考虑。

### 25.3.15 TIM 定时器特性

有关输入输出复用功能引脚（输出比较、输入捕获、外部时钟、PWM 输出）的特性详情，参见下表。

符号	参数	条件	最小值	最大值	单位
$t_{res}$	定时器分辨时间		1		$t_{TIMCLK}$
		$f_{TIMCLK}=48MHz$	20.8		ns
$f_{ext}$	外部时钟频率		0	$f_{TIMCLK}/2$	MHz
		$f_{TIMCLK}=48MHz$	0	24	MHz
$Res_{Tim}$	定时器分辨率			16	位
		自由计数		32	位
$T_{counter}$	选择内部时钟时，16 位计数器时钟周期		1	65536	$t_{TIMCLK}$
		$f_{TIMCLK}=48MHz$	0.0208	1363	us
$T_{MAX\_COUNT}$	最大可能计数			16777216	$t_{TIMCLK}$
		$f_{TIMCLK}=48MHz$		349.5	ms

表 25-2 高级定时器 (ATIM) 特性

1. 由设计保证，不在生产中测试。

符号	参数	条件	最小值	最大值	单位
$t_{res}$	定时器分辨时间		1		$t_{TIMCLK}$
		$f_{TIMCLK}=48MHz$	20.8		ns
$f_{ext}$	外部时钟频率		0	$f_{TIMCLK}/2$	MHz
		$f_{TIMCLK}=48MHz$	0	24	MHz
$Res_{Tim}$	定时器分辨率			16	位
$T_{counter}$	选择内部时钟时，16 位计数器时钟周期		1	65536	$t_{TIMCLK}$
		$f_{TIMCLK}=48MHz$	0.0208	1363	us
$T_{MAX\_COUNT}$	最大可能计数			2147483648	$t_{TIMCLK}$
		$f_{TIMCLK}=48MHz$		44.7	s

表 25-3 通用定时器特性

1. 由设计保证，不在生产中测试。

符号	参数	条件	最小值	最大值	单位
$t_{res}$	定时器分辨时间		1		$t_{TIMCLK}$
		$f_{TIMCLK}=48MHz$	20.8		ns
$f_{ext}$	外部时钟频率		0	$f_{TIMCLK/2}$	MHz
		$f_{TIMCLK}=48MHz$	0	24	MHz
$Res_{Tim}$	定时器分辨率			16	位
$T_{counter}$	选择内部时钟时, 16 位计数器时钟周期		1	65536	$t_{TIMCLK}$
		$f_{TIMCLK}=48MHz$	0.0208	1363	us
$T_{MAX\_COUNT}$	最大可能计数			2147483648	$t_{TIMCLK}$
		$f_{TIMCLK}=48MHz$		44.7	s

表 25-4 低功耗定时器特性

1. 由设计保证, 不在生产中测试。

符号	参数	条件	最小值	最大值	单位
$t_{res}$	WDT 溢出时间	$f_{WDTCLK}=32.8KHz$	0.13	64000	ms

表 25-5 IWDT 特性

1. 由设计保证, 不在生产中测试。

符号	参数	条件	最小值	最大值	单位
$t_{res}$	WDT 溢出时间	$f_{WDTCLK}=48MHz$	0.085	699	ms

表 25-6 WWDT 特性

1. 由设计保证, 不在生产中测试。

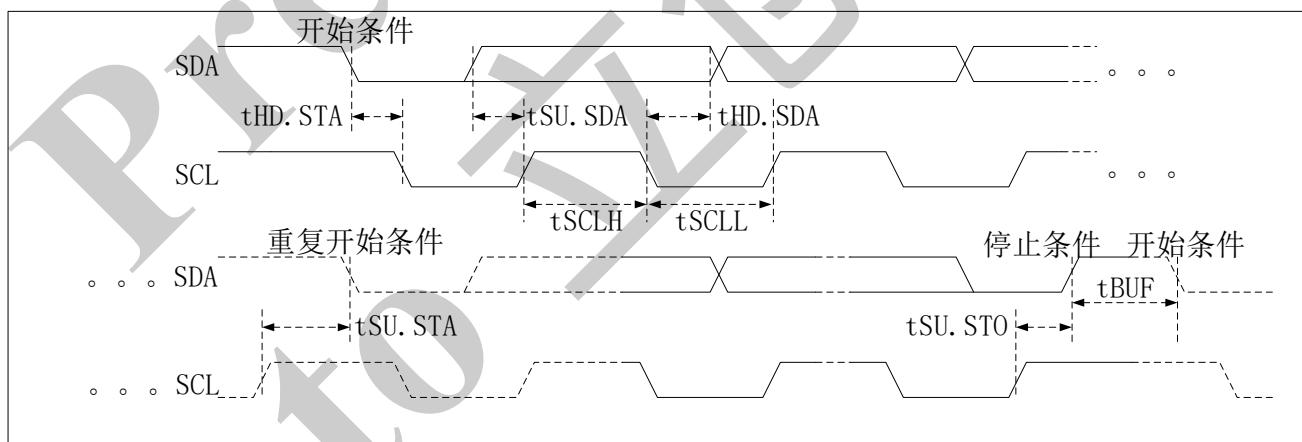
### 25.3.16 通信接口

#### 25.3.16.1 I2C 特性

I2C 接口特性如下表：

符号	参数	标准模式 (100K)		快速模式(400K)		高速模式 (1M)		单 位
		最小值	最大值	最小值	最大值	最小值	最大值	
$t_{SCLL}$	SCL 时钟低时间	4.7		1.25		0.5		us
$t_{SCLH}$	SCL 时钟高时间	4.0		0.6		0.26		us
$t_{SU.SDA}$	SDA 建立时间	250		100		50		ns
$t_{HD.SDA}$	SDA 保持时间	0		0		0		us
$t_{HD.STA}$	开始条件保持时间	4.0		0.6		0.26		us
$t_{SU.STA}$	重复的开始条件建立时间	2.5		0.6		0.25		us
$t_{SU.STO}$	停止条件建立时间	4.0		0.6		0.26		us
$t_{BUF}$	总线空闲(停止条件至开 始条件)	4.7		1.3		0.5		us

表 25-7 I2C 接口特性



1. 由设计保证, 不在生产中测试。

图 25-3 I2C 接口时序

### 25.3.16.2 SPI 特性

符号	参数	条件	最小值	最大值	单位
$t_c(SCK)$	串行时钟的周期	主机模式	62.5		ns
		从机模式 $f_{PCLK} = 48MHz$	125		ns
		从机模式 $f_{PCLK} = 16MHz$	250		ns
$t_w(SCKH)$	串行时钟的高电平时间	主机模式	$0.5 \times t_c(SCK)$		ns
		从机模式	$0.5 \times t_c(SCK)$		ns
$t_w(SCKL)$	串行时钟的低电平时间	主机模式	$0.5 \times t_c(SCK)$		ns
		从机模式	$0.5 \times t_c(SCK)$		ns
$t_{su(NSS)}$	从机选择的建立时间	从机模式	$0.5 \times t_c(SCK)$		ns
$t_{h(NSS)}$	从机选择的保持时间	从机模式	$0.5 \times t_c(SCK)$		ns
$t_{v(MO)}$	主机数据输出的生效时间	$f_{PCLK} = 48MHz$		3	ns
$t_{h(MO)}$	主机数据输出的保持时间	$f_{PCLK} = 48MHz$	2		ns
$t_{v(SO)}$	从机数据输出的生效时间	$f_{PCLK} = 48MHz$		50	ns
$t_{h(SO)}$	从机数据输出的保持时间	$f_{PCLK} = 48MHz$	30		ns
$t_{su(MI)}$	主机数据输入的建立时间		10		ns
$t_{h(MI)}$	主机数据输入的保持时间		2		ns
$t_{su(SI)}$	从机数据输入的建立时间		10		ns
$t_{h(SI)}$	从机数据输入的保持时间		2		ns

1. 由设计保证, 不在生产中测试。

表 25-8 SPI 接口特性

SPI 接口信号的波形和时序参数如下：

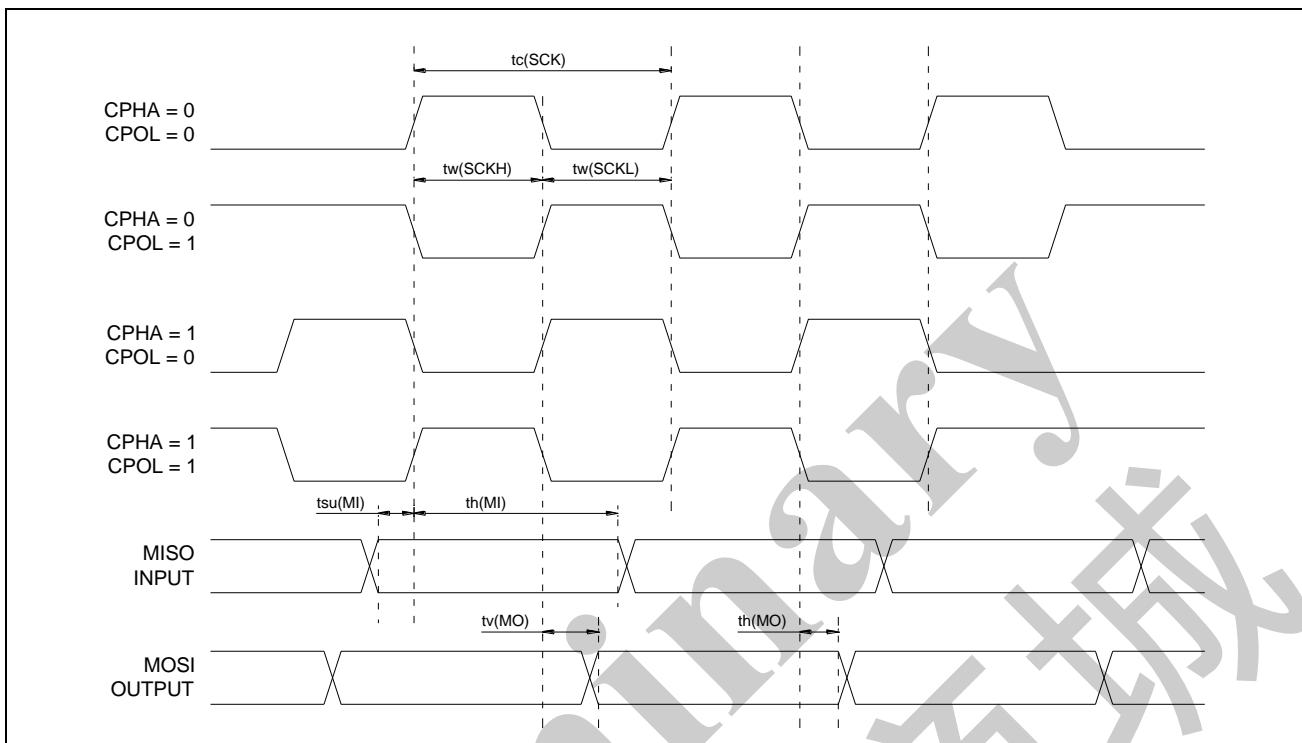


图 25-4 SPI 时序图 (主机模式)

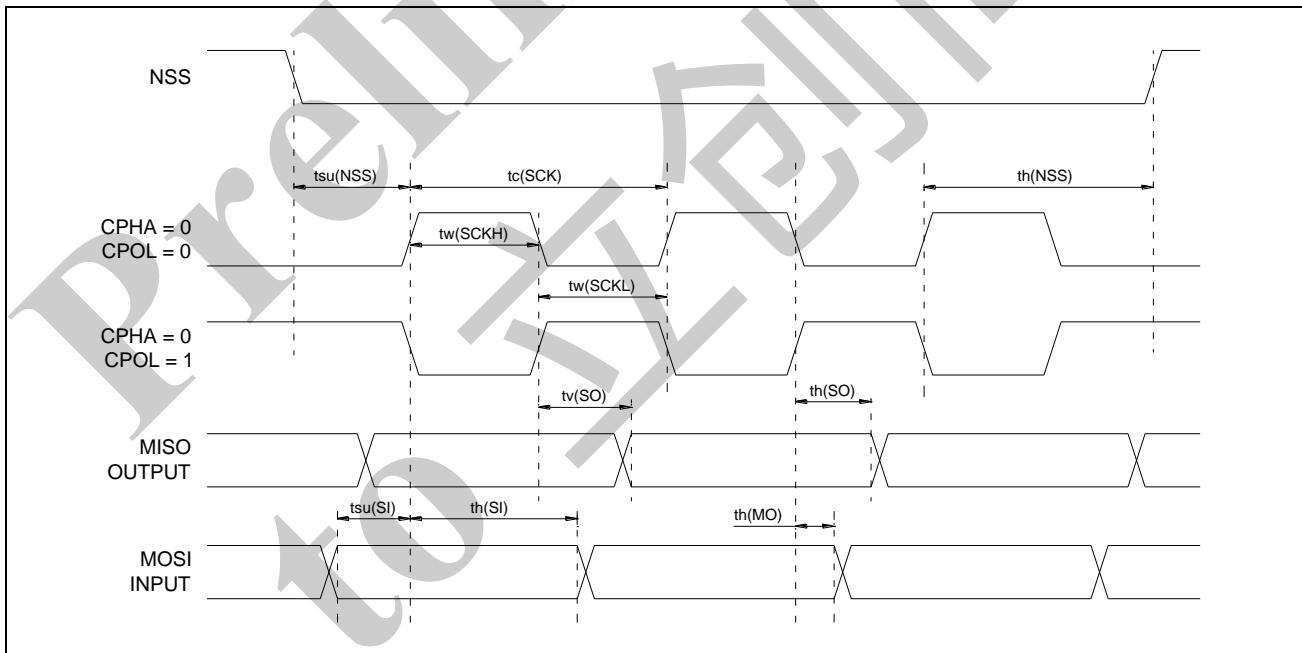


图 25-5 SPI 时序图 (从机模式 CPHA=0)

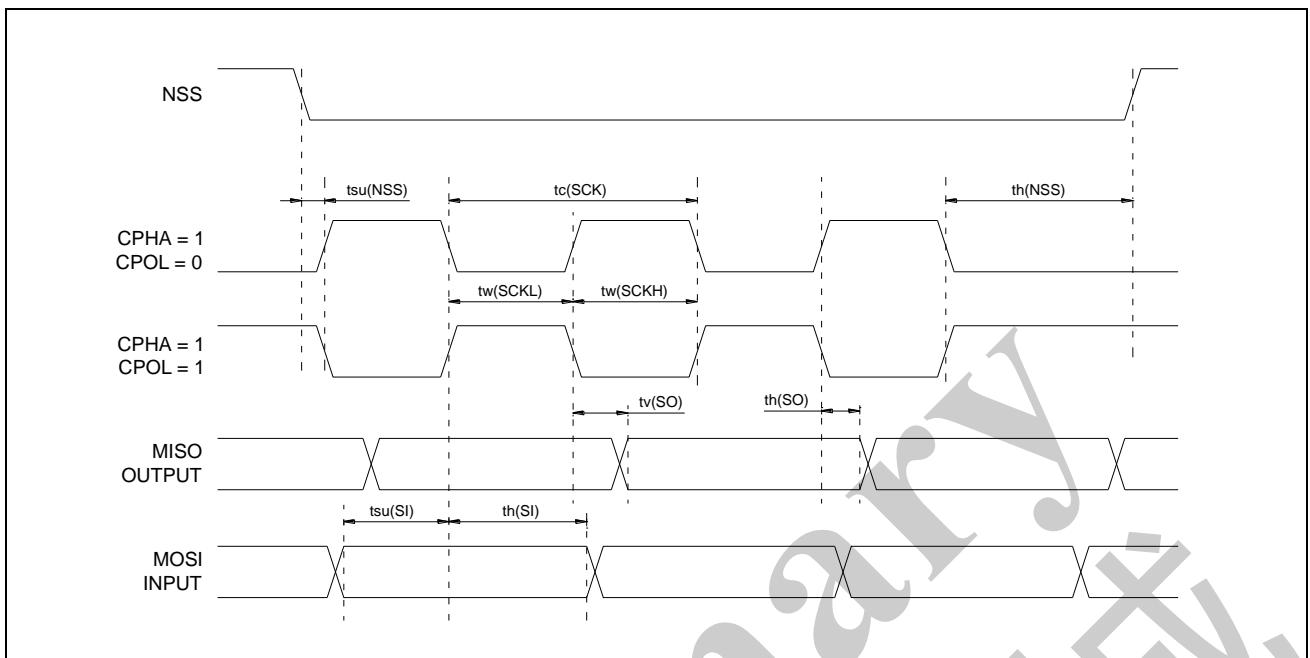
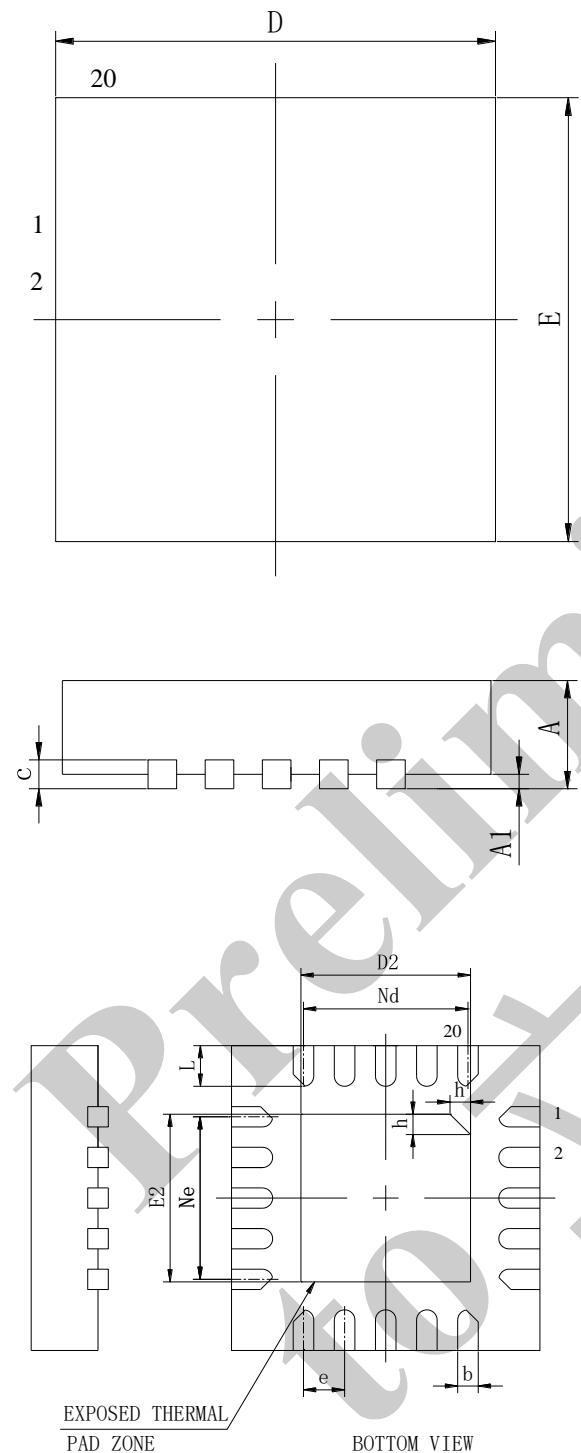


图 25-6 SPI 时序图 (从机模式 CPHA=1)

Preliminary  
to be confirmed

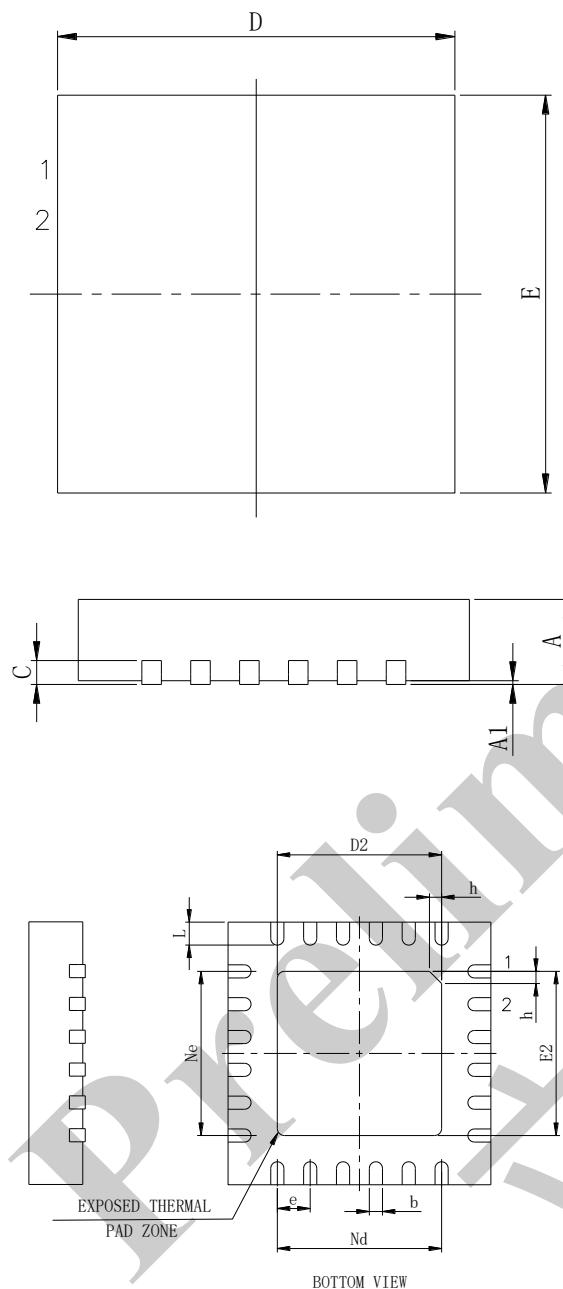
## 26 封装尺寸

### QFN20 封装



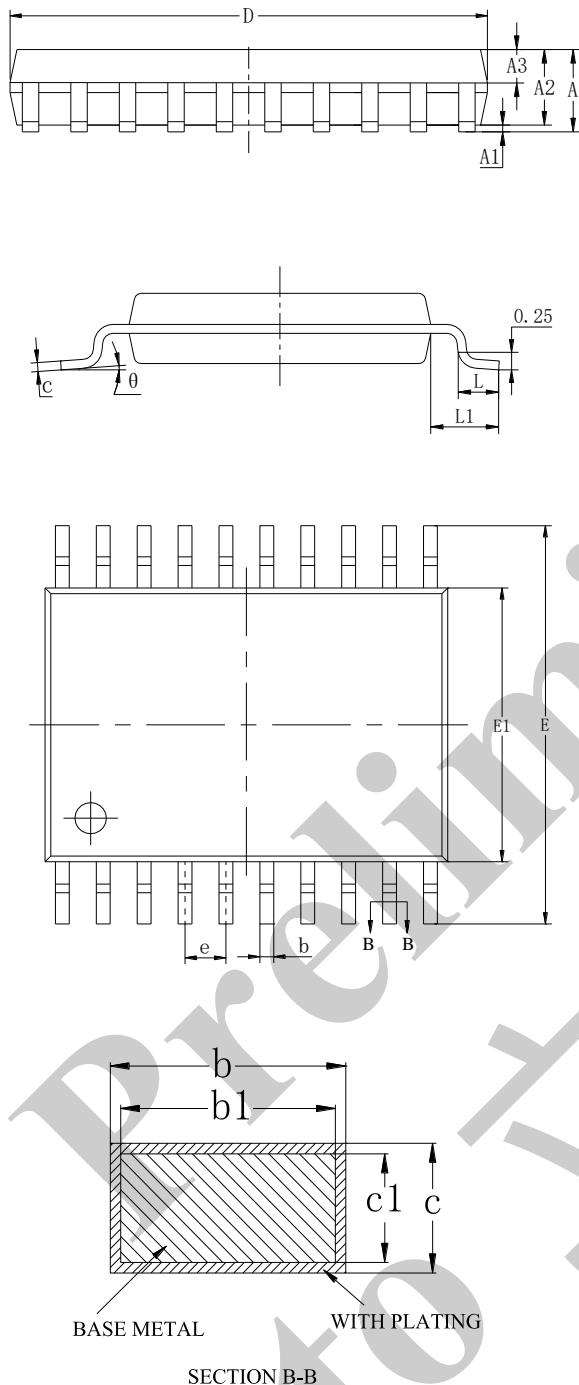
Symbol	QFN20 (3x3) millimeter		
	Min	Nom	Max
A	0.70	0.75	0.80
A1	--	0.02	0.05
b	0.15	0.20	0.25
c	0.18	0.20	0.25
D	2.90	3.00	3.10
D2	1.55	1.65	1.75
e	0.40BSC		
Ne	1.60BSC		
Nd	1.60BSC		
E	2.90	3.00	3.10
E2	1.55	1.65	1.75
L	0.35	0.40	0.45
h	0.20	0.25	0.30
L/F 载体尺寸 (Mil)	75 x 75		

QFN24 封装



Symbol	QFN24 (4x4) millimeter		
	Min	Nom	Max
A	0.70	0.75	0.80
A1	0	0.02	0.05
b	0.20	0.25	0.30
c	0.18	0.20	0.25
D	3.90	4.00	4.10
D2	2.40	2.50	2.60
e	0.50BSC		
Ne	2.50BSC		
Nd	2.50BSC		
E	3.90	4.00	4.10
E2	2.40	2.50	2.60
L	0.35	0.40	0.45
h	0.30	0.35	0.40
L/F 载体尺寸 (Mil)	110 x 110		

## TSSOP20 封装

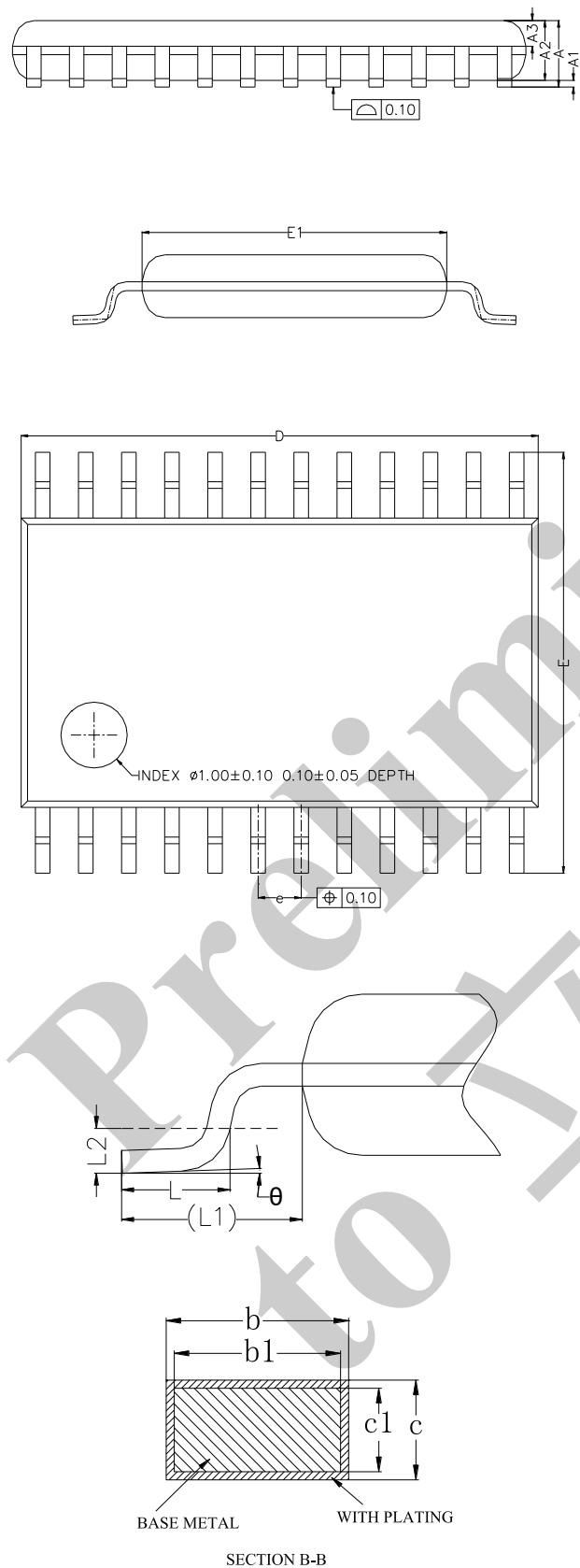


Symbol	TSSOP20 millimeter		
	Min	Nom	Max
A	--	--	1.20
A1	0.05	--	0.15
A2	0.80	1.00	1.05
A3	0.39	0.44	0.49
b	0.20	--	0.28
b1	0.19	0.22	0.25
c	0.13	--	0.18
c1	0.12	0.13	0.14
D	6.40	6.50	6.60
E	6.20	6.40	6.60
E1	4.30	4.40	4.50
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00REF		
$\theta$	0	--	8°

### NOTE:

- Dimensions “D” and “E1” do not include mold flash.

## TSSOP24 封装



Symbol	TSSOP24 millimeter		
	Min	Nom	Max
A	--	--	1.20
A1	0.05	--	0.15
A2	0.80	0.90	1.00
A3	0.34	0.39	0.44
b	0.20	--	0.29
b1	0.19	0.22	0.25
c	0.10	--	0.19
c1	0.10	0.13	0.15
D	7.70	7.80	7.90
E	6.20	6.40	6.60
E1	4.30	4.40	4.50
e	0.55	0.65	0.75
L	0.45	0.60	0.75
L1	1.00REF		
L2	0.25BSC		
θ	0	--	8°

### NOTE:

- Dimensions "D" and "E1" do not include mold flash.

## 27 附录 A SysTick 定时器

### 27.1 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 异常（异常编号为 15），该异常事件会引起 SysTick 异常处理的执行，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 异常的产生是可控的，如果异常被禁止，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计数标志。

### 27.2 设置 SysTick

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：

- Step1：配置 SysTick->CTRL.ENABLE 为 0，禁止 SysTick。
- Step2：配置 SysTick->CTRL.CLKSOURCE，选择 SysTick 的时钟源。
- Step3：配置 SysTick->LOAD，选择 SysTick 的溢出周期。
- Step4：向 SysTick->VAL 写入任意值，清零 SysTick->VAL 及 SysTick->CTRL.COUNTFLAG。
- Step5：配置 SysTick->CTRL.TICKINT 为 1，使能 SysTick 中断。
- Step6：配置 SysTick->CTRL.ENABLE 为 1，使能 SysTick。
- Step7：在中断服务程序中读取 SysTick->CTRL 以清除溢出标志。

注意：SysTick 溢出周期为 SysTick->LOAD+1，配置示例如下：

时钟源	SysTick->LOAD	溢出周期
HCLK 4MHz	3999	1ms

## 27.3 SysTick 寄存器

地址	名称	CMSIS 符号	全名
0xE000 E010	SYS_CSR	SysTick->CTRL	SysTick控制和状态寄存器
0xE000 E014	SYS_RVR	SysTick->LOAD	SysTick重装载寄存器
0xE000 E018	SYS_CVR	SysTick->VAL	SysTick当前值寄存器
0xE000 E01C	SYS_CALIR	SysTick->CALIB	SysTick校准值寄存器

### 27.3.1 SysTick 控制和状态寄存器 (CTRL)

位	符号	功能描述	类型	复位值
31:17	Reserved	-	-	-
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	Reserved	-	-	-
2	CLKSOURCE	SysTick时钟源选择 无参考时钟, 始终以系统时钟作为SysTick时钟	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

### 27.3.2 SysTick 重载寄存器 (LOAD)

位	符号	功能描述	类型	复位值
31:24	Reserved	-	-	-
23:0	RELOAD	SysTick 定时器重载值	RW	未定义

### 27.3.3 SysTick 当前值寄存器 (VAL)

位	符号	功能描述	类型	复位值
31:24	Reserved	-	-	-
23:0	CURRENT	读取该寄存器, 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

### 27.3.4 SysTick 校准值寄存器 (CALIB)

位	符号	功能描述	类型	复位值
31	NOREF	SysTick当前计数时钟标志 1: 当前计数时钟为内核时钟 0: 无功能	RO	-
30	SKEW	TENMS精度指示 1: TENMS值代表粗略的10ms 0: TENMS值代表精确的10ms	RO	-
29:24	Reserved	-	-	-
23:0	TENMS	10ms校准值	RO	-

## 28 附录 B 文档约定

### 28.1 寄存器相关缩写词列表

寄存器说明中使用以下缩写词：

RW	读写，软件可以读写这些位。
RO	只读，软件只能读取这些位。
WO	只写，软件只能写入该位。读取该位时将返回无效数据。
W1	只写 1，硬件自动清 0，写 0 无效
R0W1	软件读取该位为 0，写入 1 将该位清零。写入 0 对该位的值无影响。
RW0	软件可以读写该位，写 1 无效，写 0 清除
R1W0	软件读取该位为 1，写入 0 将该位清零。写入 1 对该位的值无影响。
RC	软件可以读取该位。读取该位时，将自动清零。 写入“0”对该位的值无影响。
Res, Reserverd	保留位，必须保持复位值。

### 28.2 词汇表

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

**Word:** 32 位数据。

**Half Word:** 16 位数据。

**Byte:** 8 位数据。

**IAP** (在应用中编程): IAP 是指可以在用户程序运行期间对微控制器的 Flash 进行重新编程。

**ICP** (在电路编程): ICP 是指可以在器件安装于用户应用电路板上时使用 JTAG 协议、SWD 协议或自举程序对微控制器的 Flash 进行编程。

**AHB:** 高级高性能总线。

**APB:** 高级外设总线。

**DMA:** 直接存储器访问。

**TIM:** 定时器

## 版本记录 & 联系方式

版本	修订日期	修订内容摘要
Beta1.0	2020/8/14	初稿发布。

**Beta 版本，仅供参考**



如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。

Email: [mcu@hdsc.com.cn](mailto:mcu@hdsc.com.cn)

网址: <http://www.hdsc.com.cn/mcu.htm>

通信地址: 上海市浦东新区中科路 1867 号 A 座 10 层

邮编: 201203

