

JMT18F003PLUS MCU

用户手册

IRONCHIP

江苏宏云技术有限公司

www.macrocloudtec.com

0512-58191012

1	产品概述.....	9
1.1	功能描述	9
1.2	主要特性	9
1.3	框图	11
1.4	应用范围	11
1.5	引脚配置	11
1.5.1	TSSOP20L 引脚图	11
1.5.2	引脚说明	11
2	电源管理（PMU）	13
2.1	概述	13
2.2	低功耗控制	13
2.2.1	降低系统时钟频率	14
2.2.2	外设时钟门控	14
2.2.3	关闭模拟模块	14
2.2.4	四种低功耗模式	14
2.2.5	OSC 开关说明	16
2.3	寄存器	16
2.3.1	低功耗控制寄存器(PMUCTR) [0xA1]	16
3	时钟复位管理（CRM）	18
3.1	时钟管理	18
3.1.1	概述	18
3.1.2	框图	18
3.1.3	功能描述	18
3.1.4	寄存器描述	22
3.2	复位管理	28
3.2.1	概述	28
3.2.2	框图	29
3.2.3	功能描述	29
3.2.4	寄存器	31
4	JMT51 MCU 核	34
4.1	概述	34

4.2	指令集	34
5	中断	38
5.1	概述	38
5.2	外部管脚中断 (EXINT)	39
5.3	中断优先级	39
5.4	中断处理过程	39
5.5	中断响应时间	40
5.6	中断寄存器	40
5.6.1	中断使能寄存器 0(IEN0) [0xA8]	40
5.6.2	中断使能寄存器 1(IEN1) [0xB8]	40
5.6.3	中断使能寄存器 2 (IEN2) [0xC0]	41
5.6.4	中断优先级寄存器 0(IP0) [0xB9]	41
5.6.5	中断优先级寄存器 1(IP1) [0xBA]	42
5.6.6	中断优先级寄存器 2(IP2) [0xBB]	42
6	存储器	43
6.1	JMT51 存储器	43
6.1.1	程序存储器	43
6.1.2	外部数据存储器	43
6.1.3	内部数据存储器	43
6.1.4	特殊功能寄存器	45
6.2	JMT51 核寄存器	49
6.2.1	ACC 寄存器(A) [0xE0]	49
6.2.2	B 寄存器(B) [0xF0]	49
6.2.3	堆栈指针(SP) [0x81]	49
6.2.4	程序状态字寄存器(PSW) [0xD0]	50
6.2.5	数据指针低字节(DPL) [0x82]	50
6.2.6	数据指针高字节(DPH) [0x83]	50
6.2.7	MOVX 指令设置寄存器(MOVXCON) [0x86]	51
6.3	PAGE 堆栈寄存器	51
6.3.1	SFR PAGE 寄存器(SFRPAGE) [0x84]	51
6.3.2	PAGE 堆栈使能寄存器(SFRPGEN) [0x85]	52
6.3.3	PAGE 堆栈寄存器 0(PGSTACK0) [0xF9]	52
6.3.4	PAGE 堆栈寄存器 1(PGSTACK1) [0xFA]	52

6.3.5	PAGE 堆栈寄存器 2(PGSTACK2) [0xFB].....	52
6.3.6	PAGE 堆栈寄存器 3(PGSTACK3) [0xFC].....	52
7	Flash 存储器	53
7.1	概述	53
7.2	FLASH 操作	53
7.2.1	FLASH 数据读取	53
7.2.2	FLASH 扇区擦除	54
7.2.3	FLASH 编程	54
7.3	FLASH 寄存器	54
7.3.1	FLASH 控制寄存器(FLSC) [0xF9]	54
7.3.2	FLASH 关键字寄存器(FLSK) [0xFA]	54
8	看门狗定时器 (WDT)	56
8.1	概述	56
8.2	功能描述	56
8.3	寄存器	56
8.3.1	看门狗分频系数寄存器(WDTCKDIV) [0xD9]	56
8.3.2	看门狗比较值寄存器(WDTINT) [0xDA]	56
8.3.3	看门狗中断标志寄存器(WDTINTF) [0xDB]	57
8.3.4	看门狗喂狗寄存器(WDTLD) [0xDC].....	57
8.3.5	看门狗启停寄存器(WDTST) [0xDD]	57
9	定时器/计数器 (TIMER)	58
9.1	概述	58
9.2	TIMER 管脚配置	58
9.3	Timer0 工作模式	58
9.3.1	模式 0(13 位定时器/计数器)	58
9.3.2	模式 1(16 位定时器/计数器)	59
9.3.3	模式 2(8 位自动重装定时器/计数器)	59
9.3.4	模式 3(两个 8 比特定定时器)	59
9.4	Timer1 工作模式	60
9.4.1	模式 0(13 位定时器/计数器)	60
9.4.2	模式 1(16 位定时器/计数器)	60
9.4.3	模式 2(8 位自动重装定时器/计数器)	61

9.5	Timer2 工作模式	61
9.5.1	模式 0(13 位定时器/计数器)	61
9.5.2	模式 1(16 位定时器/计数器)	62
9.5.3	模式 2(16 位自动重装定时器/计数器)	62
9.5.4	模式 3(1 个 8 比特定定时器/计数器)	63
9.6	TIMER 寄存器	63
9.6.1	Timer0/1 控制寄存器(TCON)[0x88]	64
9.6.2	Timer0/1 模式寄存器(TMOD)[0x89]	64
9.6.3	Timer0 低 8 位(TL0)[0x8A]	65
9.6.4	Timer1 低 8 位(TL1)[0x8B]	65
9.6.5	Timer0 高 8 位(TH0)[0x8C]	65
9.6.6	Timer1 高 8 位(TH1)[0x8D]	65
9.6.7	Timer0/1 预分频控制器(TPSC)[0x8E]	65
9.6.8	Timer2 控制寄存器(T2CON)[0xD1]	66
9.6.9	Timer2 重载寄存器低 8 位(RL2)[0xD2]	66
9.6.10	Timer2 重载寄存器高 8 位(RH2)[0xD3]	66
9.6.11	Timer2 低 8 位(TL2)[0xD4]	66
9.6.12	Timer2 高 8 位(TH2)[0xD5]	66
9.6.13	Timer2 预分频控制器(T2PSC)[0xD6]	66
10	高级定时器 0 (PWM0)	68
10.1	概述	68
10.2	特性	68
10.3	框图	68
10.4	输入输出引脚配置	69
10.5	功能描述	69
10.5.1	计数器	69
10.5.2	输入捕获功能	79
10.5.3	从模式控制	81
10.5.4	输出比较功能	84
10.5.5	主模式控制	101
10.5.6	中断	102
10.6	寄存器描述	103
10.6.1	TIM0 控制寄存器 0 (TIM0_CONR0) [0xD1]	104
10.6.2	TIM0 控制寄存器 1 (TIM0_CONR1) [0xD2]	106

10.6.3	TIM0 控制寄存器 2 (TIM0_CONR2) [0xD3]	106
10.6.4	TIM0 控制寄存器 3 (TIM0_CONR3) [0xD4]	107
10.6.5	从模式控制寄存器 0 (TIM0_TGICR0) [0xD5]	108
10.6.6	从模式控制寄存器 1 (TIM0_TGICR1) [0xD6]	109
10.6.7	TIM0 中断使能寄存器 0 (TIM0_IER0) [0xD7]	109
10.6.8	TIM0 中断使能寄存器 1 (TIM0_IER1) [0xFB]	110
10.6.9	TIM0 通道 CC0 输入捕获/输出比较模式控制寄存器 (TIM0_CC0MR) [0xD9] 110	
10.6.10	TIM0 通道 CC1 输入捕获/输出比较模式控制寄存器 (TIM0_CC1MR) [0xDA] 113	
10.6.11	TIM0 通道 CC2 输入捕获/输出比较模式控制寄存器 (TIM0_CC2MR) [0xDB] 115	
10.6.12	TIM0 通道 CC3 输入捕获/输出比较模式控制寄存器 (TIM0_CC3MR) [0xDC] 118	
10.6.13	通道使能控制寄存器 (TIM0_CCENR) [0xDD]	120
10.6.14	通道极性控制寄存器 (TIM0_CCPS) [0xDE]	121
10.6.15	TIM0 死区时间寄存器 (TIM0_DTG) [0xDF]	122
10.6.16	TIM0 周期值自动装载寄存器的低 8 位 (TIM0_ARRL) [0xE1]	122
10.6.17	TIM0 周期值自动装载寄存器的高 8 位 (TIM0_ARRH) [0xE2]	123
10.6.18	TIM0 预分频寄存器的低 8 位 (TIM0_PSCL) [0xE3]	123
10.6.19	TIM0 预分频寄存器的高 8 位 (TIM0_PSCH) [0xE4]	123
10.6.20	TIM0 重复计数寄存器 (TIM0_RCR) [0xE5]	123
10.6.21	通道 CC0 捕获/比较寄存器的低 8 位 (TIM0_CC0RL) [0xE6]	124
10.6.22	通道 CC0 捕获/比较寄存器的高 8 位 (TIM0_CC0RH) [0xE7]	124
10.6.23	通道 CC1 捕获/比较寄存器的低 8 位 (TIM0_CC1RL) [0xE9]	124
10.6.24	通道 CC1 捕获/比较寄存器的高 8 位 (TIM0_CC1RH) [0xEA]	125
10.6.25	通道 CC2 捕获/比较寄存器的低 8 位 (TIM0_CC2RL) [0xEB]	125
10.6.26	通道 CC2 捕获/比较寄存器的高 8 位 (TIM0_CC2RH) [0xEC]	125
10.6.27	通道 CC3 捕获/比较寄存器的低 8 位 (TIM0_CC3RL) [0xED]	126
10.6.28	通道 CC3 捕获/比较寄存器的高 8 位 (TIM0_CC3RH) [0xEE]	126
10.6.29	TIM0 刹车控制寄存器 (TIM0_BRKC) [0xEF]	126
10.6.30	TIM0 事件产生寄存器 (TIM0_EGR) [0xF1]	128
10.6.31	相位偏移控制寄存器 (TIM0_PHACON) [0XF2]	129
10.6.32	TIM0 中断标志寄存器 1 (TIM0_ISR0) [0xF3]	129
10.6.33	TIM0 中断标志寄存器 2 (TIM0_ISR1) [0xF4]	130

10.6.34	TIM0 计数器的低 8 位 (TIM0_CNTL) [0xF5]	130
10.6.35	TIM0 计数器的高 8 位 (TIM0_CNTH) [0xF6]	131
10.6.36	相位偏移长度寄存器的低 8 位(TIM0_PHASEL) [0XF7]	131
10.6.37	相位偏移长度寄存器的高 8 位(TIM0_PHASEL) [0XF9]	131
10.6.38	TIM 计数器使能寄存器(TIM0_CNTEN) [0xFA]	131
11	高级定时器 1 (PWM1)	132
11.1	概述	132
11.2	特性	132
11.3	框图	132
11.4	输入输出引脚配置	133
11.5	功能描述	133
11.5.1	计数器.....	133
11.5.2	输入捕获功能.....	142
11.5.3	从模式控制.....	143
11.5.4	输出比较功能.....	147
11.5.5	主模式控制.....	155
11.5.6	中断.....	156
11.6	寄存器描述	158
11.6.1	TIM1 控制寄存器 0 (TIM1_CONR0) [0xA1]	158
11.6.2	TIM1 控制寄存器 1 (TIM1_CONR1) [0xA3]	160
11.6.3	TIM1 控制寄存器 2 (TIM1_CONR2) [0xA4]	160
11.6.4	从模式控制寄存器 0 (TIM1_TGICR0) [0xA5].....	161
11.6.5	TIM1 中断使能寄存器 0 (TIM1_IER0) [0xA7].....	162
11.6.6	TIM1 中断使能寄存器 1 (TIM1_IER1) [0xCB]	162
11.6.7	TIM1 通道 CC0 输入捕获/输出比较模式控制寄存器 (TIM1_CC0MR) [0xA9]	163
11.6.8	TIM1 通道 CC1 输入捕获控制寄存器 (TIM1_CC1MR) [0xAA].....	165
11.6.9	TIM1 通道 CC2 输入捕获控制寄存器 (TIM1_CC2MR) [0xAB]	166
11.6.10	通道使能控制寄存器 (TIM1_CCENR) [0xAD]	166
11.6.11	通道极性控制寄存器 (TIM1_CCPS) [0xAE]	167
11.6.12	TIM1 死区时间寄存器 (TIM1_DTG) [0xAF].....	167
11.6.13	TIM1 周期值自动装载寄存器的低 8 位 (TIM1_ARRL) [0xB1].....	167
11.6.14	TIM1 周期值自动装载寄存器的高 8 位 (TIM1_ARRH) [0xB2]	168

11.6.15	TIM1 预分频寄存器的低 8 位 (TIM1_PSCL) [0xB3]	168
11.6.16	TIM1 预分频寄存器的高 8 位 (TIM1_PSCH) [0xB4]	168
11.6.17	TIM1 重复计数寄存器 (TIM1_RCR) [0xB5]	168
11.6.18	通道 CC0 捕获/比较寄存器的低 8 位 (TIM1_CC0RL) [0xB6]	169
11.6.19	通道 CC0 捕获/比较寄存器的高 8 位 (TIM1_CC0RH) [0xB7]	169
11.6.20	通道 CC1 捕获寄存器的低 8 位 (TIM1_CC1RL) [0xB9]	169
11.6.21	通道 CC1 捕获寄存器的高 8 位 (TIM1_CC1RH) [0xBA]	170
11.6.22	通道 CC2 捕获/比较寄存器的低 8 位 (TIM1_CC2RL) [0xBB]	170
11.6.23	通道 CC2 捕获/比较寄存器的高 8 位 (TIM1_CC2RH) [0xBC]	170
11.6.24	TIM1 刹车控制寄存器 (TIM1_BRKC) [0xBF]	170
11.6.25	TIM1 事件产生寄存器 (TIM1_EGR) [0xC1]	171
11.6.26	TIM1 中断标志寄存器 1 (TIM1_ISR0) [0xC3]	172
11.6.27	TIM1 中断标志寄存器 2 (TIM1_ISR1) [0xC4]	173
11.6.28	TIM1 计数器使能寄存器 (TIM1_CNTEN) [0xCA]	173
12	通用异步收发传输器 (UART)	176
12.1	概述	176
12.2	框图	176
12.3	UART 管脚配置	176
12.4	UART 工作模式	176
12.4.1	标准 UART 工作模式 0: 同步移位寄存器	176
12.4.2	标准 UART 工作模式 1: 8 位 UART, 波特率可变	177
12.4.3	标准 UART 工作模式 2: 9 位 UART, 波特率固定	178
12.4.4	标准 UART 工作模式 3: 9 位 UART, 波特率可变	179
12.4.5	38KHz 红外调制发送	179
12.4.6	IrDA(SIR)红外通信	180
12.4.7	LIN 模式	181
12.5	UART 多机通信	181
12.6	UART 奇偶校验	182
12.7	UART 波特率计算	182
12.8	UART 寄存器	188
12.8.1	UART 控制寄存器(SCON) [0x98]	188
12.8.2	UART 数据缓冲寄存器(SBUF) [0x99]	189
12.8.3	UART 波特率重载寄存器低 8 位(SRELL) [0x9A]	189

12.8.4	UART 波特率重载寄存器高 2 位(SRELH) [0x9B].....	189
12.8.5	UART 红外控制寄存器(SIRCON) [0x9C].....	189
12.8.6	LIN 控制寄存器(LINCON) [0x9D].....	189
13	I2C 接口	191
13.1	概述	191
13.2	框图	191
13.3	I2C 管脚配置	191
13.4	I2C 模块连接	191
13.5	I2C 数据格式	191
13.5.1	数据有效.....	192
13.5.2	读写控制比特.....	192
13.5.3	反馈 (ACK)	192
13.5.4	起始条件和结束条件.....	192
13.5.5	数据格式.....	192
13.5.6	时钟同步.....	193
13.5.7	仲裁.....	193
13.6	I2C 时钟产生	194
13.7	I2C 工作模式	195
13.7.1	从机模式.....	195
13.7.2	主机模式.....	197
13.8	I2C 中断	199
13.9	I2C 寄存器	200
13.9.1	I2C 控制寄存器(I2CCON) [0xE1].....	200
13.9.2	I2C 从机地址寄存器低 8 位(I2CSADDRL) [0xE2]	201
13.9.3	I2C 从机地址寄存器高 2 位(I2CSADDRH) [0xE3]	201
13.9.4	I2C 本机地址寄存器低 8 位(I2COADDRL) [0xE4].....	201
13.9.5	I2C 本机地址寄存器高 2 位(I2COADDRH) [0xE5]	201
13.9.6	I2C 工作时钟分频控制寄存器(I2CDIV) [0xE6]	202
13.9.7	I2C 时钟 SCL 低电平时间配置寄存器(I2CDUTYL) [0xE7]	202
13.9.8	I2C 时钟 SCL 高电平时间配置寄存器(I2CDUTYH) [0xE9]	202
13.9.9	I2C 数据 SDA 保持时间配置寄存器(I2CHOLD) [0xEA]	202
13.9.10	I2C 数据写缓冲寄存器(I2CWBUF) [0xEB]	203
13.9.11	I2C 数据读缓冲寄存器(I2CRBUF) [0xEC]	203

13.9.12	I2C 状态寄存器(I2CSTS) [0xED]	203
13.9.13	I2C 中断标志寄存器(I2CISC) [0xEE].....	203
13.9.14	I2C 中断使能寄存器(I2CIEN) [0xEF]	204
14	串行外设接口 (SPI)	205
14.1	概述	205
14.2	框图	205
14.3	SPI 管脚配置	205
14.4	SPI 主/从机互连	205
14.4.1	4 线互连模式	205
14.4.2	3 线互连模式	206
14.5	SPI 时序设置	207
14.6	数据发送和接收过程	208
14.6.1	主机模式	208
14.6.2	从机模式	209
14.7	SPI 中断	210
14.8	SPI 寄存器	211
14.8.1	SPI 控制寄存器(SPICON)[0xC9]	211
14.8.2	SPI 中断使能寄存器(SPIIE)[0xCA]	212
14.8.3	SPI 中断标志寄存器(SPIIF)[0xCB]	212
14.8.4	SPI 波特率控制寄存器(SPIBR)[0xCC]	213
14.8.5	SPI 数据缓冲寄存器(SPIBUF) [0xCD]	213
14.8.6	SPI 主机采样控制寄存器(SPISPC)[0xCE]	213
14.8.7	SPI 状态寄存器(SPISTS)[0xCF]	214
15	模拟/数字转换 (ADC) 和可编程增益放大器 (PGA)	215
15.1	概述	215
15.2	ADC 输入端口配置	215
15.3	可编程增益放大器 (PGA) 说明	216
15.4	ADC 参考电压配置	218
15.5	ADC 转换功能	218
15.6	ADC 转换结果数据格式	218
15.7	ADC 转换时序	218
15.8	触发事件	219

15.8.1	软件触发事件	219
15.8.2	外部触发事件	219
15.9	ADC 操作流程	220
15.10	ADC 寄存器	220
15.10.1	ADC 控制寄存器 0(ADCCON0)[0x91]	220
15.10.2	ADC 控制寄存器 1(ADCCON1)[0x92]	221
15.10.3	放大器控制寄存器(ADCPGAC)[0x93].....	221
15.10.4	ADC 外部触发事件控制寄存器(ADCETC)[0x94]	222
15.10.5	ADC 外部触发延时控制寄存器(ADCDLTY)[0x95]	222
15.10.6	ADC 结果寄存器低 8 位(ADCRL)[0x96].....	223
15.10.7	ADC 结果寄存器高 8 位(ADCRH)[0x97]	223
16	模拟比较器 (CMP)	224
16.1	模拟比较器概述	224
16.2	低压检测	225
16.3	模拟比较器寄存器	225
16.3.1	模拟比较器使能寄存器(CMPEN) [0xB9]	226
16.3.2	模拟比较器滤波选择寄存器(CMPFT) [0xBA]	226
16.3.3	模拟比较器 VDD 分压寄存器(CMPVDD) [0xBB].....	226
16.3.4	模拟比较器参考电压选择寄存器(CMPREF) [0xBC].....	226
16.3.5	模拟比较器中断寄存器(CMPINT) [0xBD]	227
16.3.6	模拟比较器比较结果寄存器(CMPFLAG) [0xBE].....	227
16.3.7	OC 控制寄存器(OCPRSEL) [0xC1].....	227
16.3.8	OD 控制寄存器(ODPRSEL) [0xC2]	227
16.3.9	PWM 刹车控制寄存器(BKPRSEL) [0xC3]	227
17	通用输入/输出 (GPIO)	229
17.1	管脚设置	229
17.2	管脚复用	232
17.3	GPIO 功能复用	234
17.4	管脚数据寄存器读写	235
17.5	外设管脚复用配置	235
17.6	管脚滤波	239
17.7	EXINT 中断.....	239

17.8	PC0 唤醒	241
17.9	GPIO 寄存器	242
17.9.1	PA0 控制寄存器(PA0CTRL) [0xB9]	242
17.9.2	PA1 控制寄存器(PA1CTRL) [0xBA]	243
17.9.3	PA2 控制寄存器(PA2CTRL) [0xBB]	243
17.9.4	PA3 控制寄存器(PA3CTRL) [0xBC]	244
17.9.5	PA4 控制寄存器(PA4CTRL) [0xBD]	244
17.9.6	PA5 控制寄存器(PA5CTRL) [0xBE]	245
17.9.7	PA6 控制寄存器(PA6CTRL) [0xBF]	246
17.9.8	PA7 控制寄存器(PA7CTRL) [0xC1]	246
17.9.9	PB0 控制寄存器(PB0CTRL) [0x89]	247
17.9.10	PB1 控制寄存器(PB1CTRL) [0x8A]	247
17.9.11	PB2 控制寄存器(PB2CTRL) [0x8B]	248
17.9.12	PB3 控制寄存器(PB3CTRL) [0x8C]	248
17.9.13	PB4 控制寄存器(PB4CTRL) [0x8D]	249
17.9.14	PB5 控制寄存器(PB5CTRL) [0x8E]	249
17.9.15	PB6 控制寄存器(PB6CTRL) [0x8F]	250
17.9.16	PB7 控制寄存器(PB7CTRL) [0xA9]	250
17.9.17	PC0 控制寄存器(PC0CTRL) [0x91]	251
17.9.18	PAD 转换速率和驱动能力寄存器(PADSRDR) [0xAD]	251
17.9.19	模拟管脚使能寄存器 0(AIOEN0) [0xB6]	252
17.9.20	模拟管脚使能寄存器 1(AIOEN1) [0xC3]	253
17.9.21	滤波时钟预分频寄存器(FPSC) [0xB7]	253
17.9.22	EXINT 上升沿中断使能寄存器(EXINTPE) [0xC4]	253
17.9.23	EXINT 下降沿中断使能寄存器(EXINTNE) [0xC5]	254
17.9.24	EXINT 中断使能寄存器(EXINTEN) [0xC6]	254
17.9.25	EXINT 中断控制寄存器(EXINTCON) [0xB5]	255
17.9.26	EXINT 中断管脚选择寄存器 0(EXINTSEL0) [0xB1]	256
17.9.27	EXINT 中断管脚选择寄存器 1(EXINTSEL1) [0xB2]	256
17.9.28	PA 数据寄存器(P0) [0x80]	257
17.9.29	PB 数据寄存器(P1) [0x90]	257
17.9.30	PC 数据寄存器(P2) [0xA0]	257
18	电气特性	258
18.1	极限参数	258

18.2	工作条件	258
18.2.1	芯片供电	258
18.2.2	功耗特性	258
18.2.3	I/O 特性	259
18.2.4	REGC 引脚特性	259
18.2.5	NRST 引脚特性	259
18.2.6	上电/掉电条件	259
18.2.7	时钟特性	260
18.2.8	通信接口	261
18.2.9	FLASH 特性	264
18.2.10	ADC 特性	264
18.2.11	PGA 特性	265
18.2.12	CMP 特性	266
18.2.13	电气敏感性	266
19	封装特性	267
19.1	TSSOP20L	267

1 产品概述

1.1 功能描述

JMT18F003PLUS 是一款 MCU 芯片，其 MCU 核为兼容 8051 的 JMT51 核，最高运行频率为 24.576MHz。JMT18F003PLUS 集成了：8K 字节 Flash、1K 字节外部 RAM、128 字节内部 RAM、2 个高性能 16 位 PWM 模块，其中 PWM0 支持 3 路互补 PWM，PWM1 支持 1 路互补 PWM、1 个 12 通道的 10 位 ADC、1 个可编程运算放大器（4 组差分输入）、1 个模拟比较器（4 组可切换）、3 个 Timer、1 个 WDT、1 个 SPI、1 个 I2C、1 个 UART 通信接口、1 个蜂鸣器输出。

1.2 主要特性

JMT51 MCU 内核:

- 8 位 MCU，兼容 MCS51 指令集
- 1T 指令周期
- 内核运行最高频率 24.576MHz
- 支持 JTAG 调试

时钟:

- 内置高精度 73.728MHz RC 振荡器，精度为 $\pm 1\%$ @5v, 25°C
- 内置 32KHz RC 振荡器，精度为 $\pm 10\%$ @5v, 25°C
- 支持 8~24MHz 外部晶体振荡器时钟输入

外设特性:

- 上电复位、掉电复位、低压复位
- 3 个 Timer、1 个 WDT
- 17 个双向 I/O
 - 8 个具有唤醒功能的输入口
 - 所有 I/O 都可作为外部中断源
- 内置 8K 字节 Flash
 - 擦写次数：至少 100,000 次
 - 保持时间：至少 10 年
- 内置 SRAM

- 128 字节内部 RAM
- 1024 字节外部 RAM
- PWM0 模块
 - 支持 3 路互补加 1 路独立 PWM
 - PWM 为 16 位计数器
 - 最高运行频率 73.728MHz
 - 支持内部、外部刹车(软硬件)
 - 支持比较器刹车
 - 支持死区可编程
 - 可以触发 ADC 采样
 - 支持正交解码
 - 支持霍尔传感器解码
 - 支持 4 路捕获
 - 支持各路 PWM 输出的相位偏移
- PWM1 模块
 - 支持 1 路互补 PWM
 - PWM 为 16 位计数器
 - 最高运行频率 73.728MHz
 - 支持内部、外部刹车(软硬件)
 - 支持比较器刹车
 - 支持死区可编程

- 可以触发 ADC 采样
 - 支持正交解码
 - 支持霍尔传感器解码
 - 支持 3 路捕获
 - 1 个 12 通道 10 位 ADC
 - 支持 12 通道输入
 - 支持外部管脚/PWM 触发
 - 采样时间延时可编程
 - 1 个 UART
 - 支持标准 UART 通信协议
 - 支持 IrDA (SIR 标准)
 - 支持输出 38kHz 红外调制
 - 支持 LIN 模式
 - 1 个 SPI 接口总线
 - 支持 4 线全双工传输
 - 支持 3 线半双工传输
 - 1-8 位数据帧格式可编程
 - 两级发送 FIFO
 - 数据 MSB 在前和 LSB 在前可选择
 - 支持主机或从机模式
 - 1 个可编程运算放大器 (PGA)
 - 支持 1、2、4、8、16 放大倍数可配置
 - 支持同相、反相极性
 - 支持差分输入四路可选
 - 1 个 I2C 接口总线
 - 支持标准模式: 100Kbps
 - 支持快速模式: 400Kbps
 - 支持 master/slave 工作模式
 - 支持单 master/多 master 操作
 - 支持 7/10 位设备地址寻址
 - 1 个蜂鸣器输出:
 - 蜂鸣器输出信号的频率和占空比可编程控制
 - 1 个模拟比较器
 - 支持低压检测功能
 - 比较器输入可选择: PA3/PGA 输出/VDDA
- 低功耗特性:**
- 芯片工作电压
 - V_{DD} : 2.75V~5.5V
 - 支持 5 种工作模式
 - Normal、Idle、Stop、Sleep 和 Deep Sleep
- 封装:**
- TSSOP20L

1.3 框图

JMT18F003PLUS 框图如 Figure 1 所示:

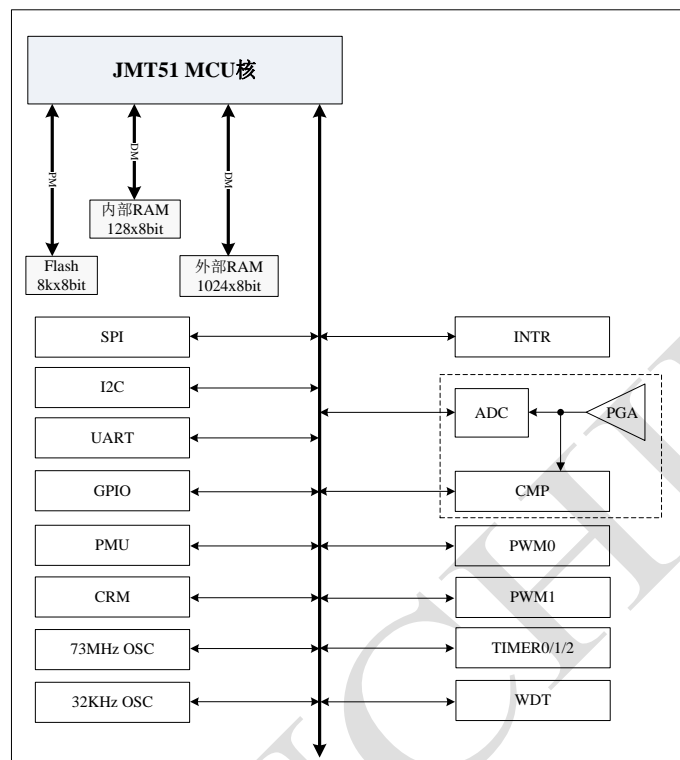


Figure 1 JMT18F003PLUS 结构框图

1.4 应用范围

JMT18F003PLUS 可应用于无线充电、电子烟、电机控制、逆变器、小家电等产品中。

1.5 引脚配置

1.5.1 TSSOP20L 引脚图

JMT18F003PLUS 芯片 TSSOP20L 封装如 Figure 2 所示:

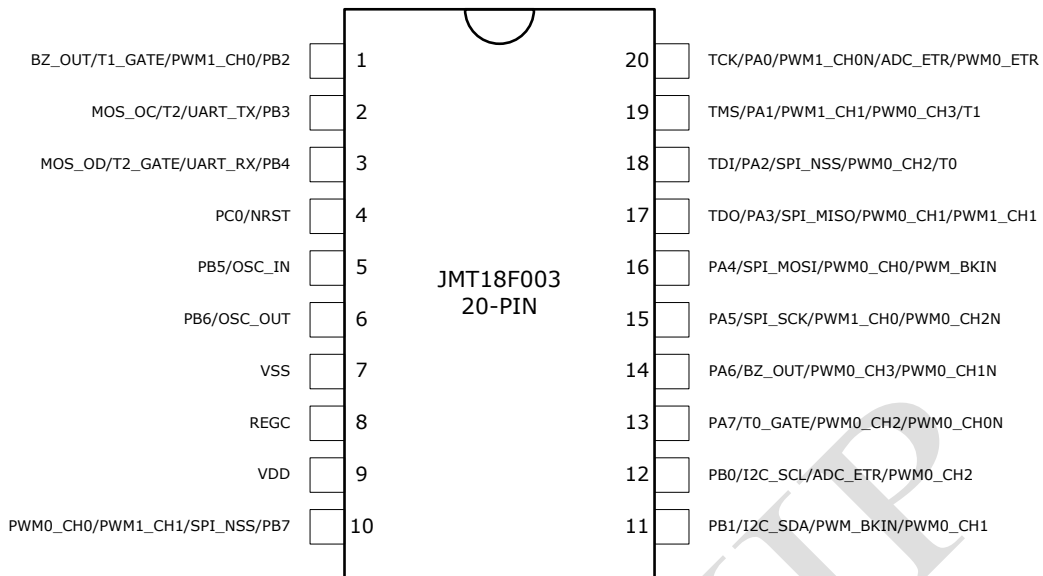


Figure 2 JMT18F003PLUS 芯片 TSSOP20L 封装

1.5.2 引脚说明

JMT18F003PLUS 芯片 I/O 分为两种类型，一种为数字、模拟复用 I/O，另一种为纯数字 I/O。PA0~PA7、PB2~PB4、PC0 为数字、模拟复用 I/O；PB0~PB1、PB5~PB7 为纯数字 I/O。管脚的功能复用见 Table 1：

Table 1 JMT18F003PLUS 引脚说明

序号 TSSOP20L	引脚 名	默认功能	复用功能 1	复用功能 2	复用功能 3	JTAG 引脚
1	PB2	GPIO	PWM1_CH0	T1_GATE	BZ_OUT	
2	PB3	GPIO	UART_TX	T2	MOS_OC	
3	PB4	GPIO	UART_RX	T2_GATE	MOS_OD	
4	NRST	NRST	GPIO			
5	PB5	OSC_IN	GPIO			
6	PB6	OSC_OUT	GPIO			
7	VSS	VSS				
8	REGC	REGC				
9	VDD	VDD				
10	PB7	GPIO	SPI_NSS	PWM1_CH1	PWM0_CH0	
11	PB1	GPIO	I2C_SDA	PWM_BKIN	PWM0_CH1	
12	PB0	GPIO	I2C_SCL	ADC_ETR	PWM0_CH2	
13	PA7	GPIO	T0_GATE	PWM0_CH2	PWM0_CH0N	
14	PA6	GPIO	BZ_OUT	PWM0_CH3	PWM0_CH1N	

序号 TSSOP20L	引脚 名	默认功能	复用功能 1	复用功能 2	复用功能 3	JTAG 引脚
15	PA5	GPIO	SPI_SCK	PWM1_CH0	PWM0_CH2N	
16	PA4	GPIO	SPI_MOSI	PWM0_CH0	PWM_BKIN	
17	PA3	GPIO	SPI_MISO	PWM0_CH1	PWM1_CH1	TDO
18	PA2	GPIO	SPI_NSS	PWM0_CH2	T0	TDI
19	PA1	GPIO	PWM1_CH1	PWM0_CH3	T1	TMS
20	PA0	GPIO	PWM1_CH0N	ADC_ETR	PWM0_ETR	TCK

2 电源管理（PMU）

2.1 概述

JMT18F003PLUS 的工作电压范围为 2.75V~5.5V，内置两个电压调节器：

- 主电压调节器（MAIN LDO）
- 常开电压调节器（AO LDO）

MAIN LDO 可以通过低功耗控制关闭，关闭后通过 AO LDO 为看门狗（WDT）以及一些控制电路供电。Figure 3 为 JMT18F003PLUS 供电示意图。

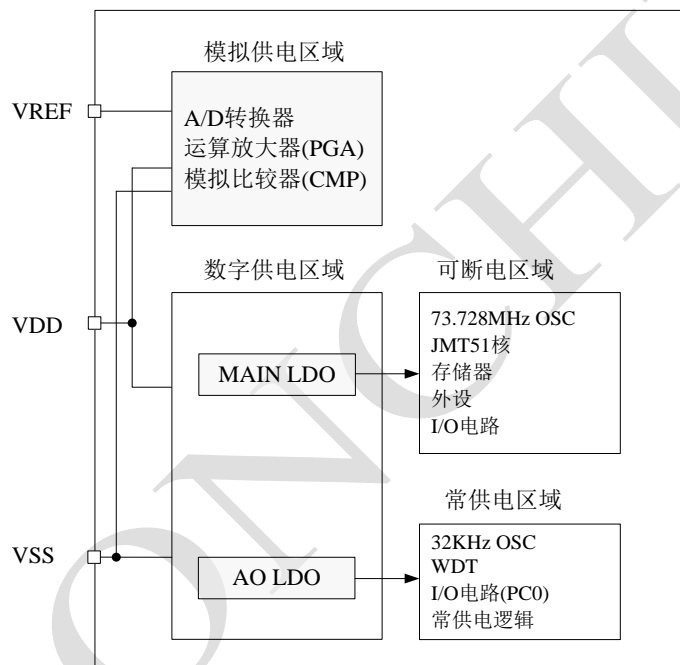


Figure 3 JMT18F003PLUS 芯片供电示意图

Figure 3 中主要分成模拟供电区域和数字供电区域。

- 模拟供电部分为 A/D 转换器，运算放大器及模拟比较器等模拟模块供电，VSS 为模拟地，VREF 为芯片外部参考电压。
- 数字供电部分一方面通过 MAIN LDO 为可断电区域供电，主要包含 73.728MHz OSC、JMT51 核、存储器、外设和 I/O 电路等；另一方面通过 AO LDO 为 32KHz OSC、WDT 和常供电逻辑供电，保证在 MAIN LDO 关闭状态下这些模块能够正常工作。

2.2 低功耗控制

在芯片上电后，微控制器处于正常运行模式（Normal）。根据用户需要，可以利用多种低功耗模式来节省功耗。用户可以根据最低功耗、唤醒时间和可用的唤醒源等条件，选定一

个最佳的低功耗模式。

JMT18F003PLUS 有四种低功耗模式：

- Idle 模式：JMT51 核时钟关闭，其他所有模块正常工作。
- Stop 模式：JMT51 核和外设时钟关闭；内部 73.728MHz OSC 关闭；MAIN LDO 及模拟模块正常供电，32KHz OSC 以及看门狗（WDT）正常工作。
- Sleep 模式：在 Stop 模式基础上进一步关闭 MAIN LDO 及模拟模块供电。
- Deep Sleep 模式：在 Sleep 模式基础上进一步关闭 32KHz OSC，此时 WDT 停止工作。
- 四种低功耗模式下对应功耗大小排序为： $P_{Idle} > P_{Stop} > P_{Sleep} > P_{Deep Sleep}$ 。
- 四种低功耗模式所需的唤醒时间长短排序为： $T_{Deep Sleep} = T_{Sleep} > T_{Stop} > T_{Idle}$ 。

此外，在 Normal 模式下，可以通过以下方式中的一种降低功耗：

- 降低系统时钟频率
- 关闭未被使用的外设时钟
- 关闭不工作的模拟模块，如 ADC，PGA，比较器等

各工作模式下工作时钟及供电开关如 Table 2 所示：

Table 2 各模式下时钟及供电开关场景表

工作模式	JMT51 时钟	外设时钟	73.728MHz OSC	32KHz OSC	MAIN LDO	AO LDO
Normal	开	开	开	开	开	开
Idle	关	开	开	开	开	开
Stop	关	关	关	开	开	开
Sleep	关	关	关	开	关	开
Deep Sleep	关	关	关	关	关	开

2.2.1 降低系统时钟频率

在正常运行模式下，可以通过配置系统时钟分频系数寄存器 SYS_CLK_DIV 来降低系统时钟频率，从而降低系统功耗；在进入 Idle 之前，也可以通过配置该寄存器降低外设的时钟。

2.2.2 外设时钟门控

JMT18F003PLUS 提供了外设的时钟门控功能，用户可以根据自己使用的外设资源及应用场景灵活的打开关闭相应外设，这样可以有效地降低系统的动态功耗，外设门控的控制可参见 3.1.4 章节门控时钟寄存器的说明。

2.2.3 关闭模拟模块

JMT18F003PLUS 模拟模块提供了低功耗模式管理，用户可根据具体使用情况对模拟模块进行开关，包括 ADC，运算放大器 PGA，模拟比较器 CMP。具体参见各自模块章节说明。

2.2.4 四种低功耗模式

PMU 控制全芯片进入或退出四种低功耗模式，即 Idle、Stop、Sleep 和 Deep Sleep 模式。通过配置寄存器 PMUCTR 进入四种低功耗模式。

寄存器 PMUCTR 管理四种不同的低功耗模式，若同时配置多种低功耗模式时，为了不引起混淆，将进入功耗最低的一种模式。例如同时配置 Idle 和 Sleep 模式时，则进入 Sleep 模式。同样唤醒时也要满足 Sleep 唤醒的条件才能够唤醒。

JMT18F003PLUS 四种低功耗模式与正常运行模式的状态转换图如 Figure 4 所示¹。

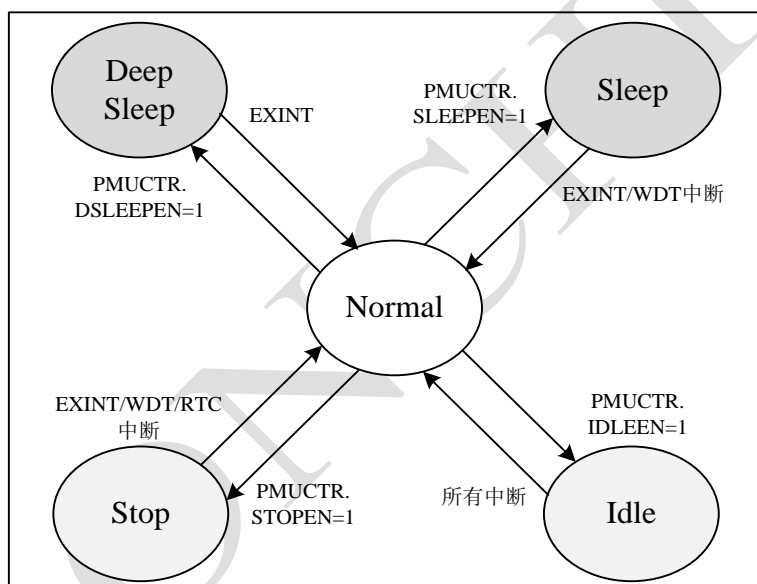


Figure 4 JMT18F003PLUS 工作模式切换场景图

2.2.4.1 Idle 模式

2.2.4.1.1 进入 Idle 模式

通过配置寄存器 PMUCTR.IDLEEN 为 1，全芯片进入 Idle 模式。

在此模式下，JMT51 核时钟关闭，除 JMT51 核外所有模块正常工作。

2.2.4.1.2 退出 Idle 模式

退出 Idle 模式至 Normal 模式的唤醒源为所有中断。

该模式唤醒所需时间为 1 个系统时钟周期，退出 Idle 模式后切换至 Normal 模式，所有模块正常工作，JMT51 核继续取指向下执行应用程序。

¹ Figure 4 中的 EXINT 表示外部中断，对应管脚为 PC0。

2.2.4.2 Stop 模式

2.2.4.2.1 进入 Stop 模式

通过配置寄存器 PMUCTR.STOPEN 为 1，全芯片进入 Stop 模式。

在此模式下，JMT51 核和外设时钟关闭；内部 73.728MHz OSC 关闭，MAIN LDO 及模拟模块正常供电，32KHz OSC 以及看门狗（WDT）正常工作。

2.2.4.2.2 退出 Stop 模式

退出 Stop 模式至 Normal 模式的唤醒源有：

- 看门狗（WDT）中断唤醒或溢出复位
- 外部中断 EXINT（PC0 中断）

该模式唤醒所需时间为 20us，退出 Stop 模式后切换至 Normal 模式，所有模块正常工作，JMT51 核继续取指向下执行应用程序。

内部 73.728MHz 和 32KHz OSC 自动打开。系统时钟硬件自动切换至内部 73.728MHz 时钟源。

2.2.4.3 Sleep 模式

2.2.4.3.1 进入 Sleep 模式

通过配置寄存器 PMUCTR.SLEEPEN 为 1，全芯片进入 Sleep 模式。

在此模式下，JMT51 核和外设时钟关闭；内部 73.728MHz OSC 关闭，MAIN LDO 及模拟模块关闭供电，内部 32KHz OSC、WDT 及 PC0 口由 AO LDO 供电正常工作。

2.2.4.3.2 退出 Sleep 模式

退出 Sleep 模式至 Normal 模式的唤醒源有：

- WDT 中断唤醒或溢出复位
- 外部中断 EXINT（PC0 中断）

该模式唤醒所需时间为 24.5us，退出 Sleep 模式后全芯片将自动复位进入 Normal 模式，JMT51 核从 FLASH 的 0 地址开始执行应用程序。

退出 Sleep 模式后系统时钟硬件自动切换至内部 73.728MHz 时钟源。

2.2.4.4 Deep Sleep 模式

2.2.4.4.1 进入 Deep Sleep 模式

通过配置寄存器 PMUCTR.DSLEEPEN 为 1，全芯片进入 Deep Sleep 模式。

在此模式下，JMT51 核和外设时钟关闭；内部 73.728MHz OSC、32KHz OSC 关闭，MAIN LDO 及模拟模块关闭供电，WDT 停止工作。

2.2.4.4.2 退出 Deep Sleep 模式

退出 Deep Sleep 模式至 Normal 模式的唤醒源只有外部中断 EXINT（PC0 中断）。

该模式唤醒所需时间为 24.5us，退出 Deep Sleep 模式后全芯片将自动复位进入 Normal

模式，JMT51 核从 FLASH 的 0 地址开始执行应用程序。

退出 Deep Sleep 模式后系统时钟硬件自动切换至内部 73.728MHz 时钟源。

2.2.5 OSC 开关说明

OSC 的打开和关闭可通过软件和硬件两种方式实现：

软件开关：

- JMT18F003PLUS 有三个时钟，其中 73.728MHz OSC 和外部 OSC 可以通过软件开关。
- 配置寄存器 OSCPD.OSC73MPD，控制 73.728MHz OSC 的打开及关闭。配置为 0 时打开（OSC73MPD 复位状态为 0），配置为 1 时关闭。
- 配置寄存器 OSCPD.OSC24MEN，控制外部 OSC 工作模式。配置为 1 时正常工作，配置为 0 时进入低功耗模式（时钟关闭）。
- 对某个 OSC 正常软件开关时，要保证系统时钟使用的时钟源与关闭的时钟源无关，这样才可以正常的通过配置寄存器实现软件开关操作。

硬件开关：

- 根据配置的低功耗模式寄存器 STOPEN、SLEEPEN 或 DSLEEPEN，硬件自动关闭相应的 OSC。由唤醒逻辑或复位，硬件自动打开 OSC，具体参见 2.2.4 四种低功耗模式中的说明。

2.3 寄存器

PMU 有 1 个寄存器，对应的 SFR PAGE 为 0，具体列表如 Table 3 所示。

Table 3 PMU 寄存器（SFRPAGE=0x00）

地址	寄存器名	属性	复位值	功能描述
0xA1	PMUCTR	W	0x00	低功耗控制寄存器

属性：W/R—可读写，W—只写，R—只读（下同）

2.3.1 低功耗控制寄存器(PMUCTR) [0xA1]

Table 4 低功耗控制寄存器(PMUCTR)

参数名	比特位	属性	复位值	描述
IDLEEN	0	W	0	IDLE 模式使能： 1：进入 IDLE 模式 0：无效
STOPEN	1	W	0	STOP 模式使能： 1：进入 STOP 模式 0：无效

参数名	比特位	属性	复位值	描述
SLEEPEN	2	W	0	Sleep 模式使能: 1: 进入 Sleep 模式 0: 无效
DSLEEPEN	3	W	0	Deep Sleep 模式使能: 1: 进入 Deep Sleep 模式 0: 无效
Reserved	7-4	R	0	保留

3 时钟复位管理 (CRM)

3.1 时钟管理

3.1.1 概述

- 芯片包含三个振荡器：一个输出频率为 73.728MHz 的片内高频 RC 振荡器，一个输出频率为 32.768KHz 的片内低频 RC 振荡器和一个最高频率为 24MHz 的外部晶体振荡器。
- 时钟管理模块管理时钟选择、时钟分频和时钟门控，为各个模块提供所需的时钟。
- 看门狗 (WDT) 和 PMU 模块的时钟直接由片内 32.768KHz 低频振荡器产生。
- 除 PWM 外的其他模块的时钟由片内 24.576MHz (由片内 73.728MHz 高频振荡器产生的时钟经过 3 分频得到) 或外部晶振 (最高 24MHz) 或片内低频振荡器产生的 32.768KHz 时钟提供。
- PWM 模块时钟片内 73.728MHz 高频振荡器或外部晶振 (最高 24MHz) 提供。

3.1.2 框图

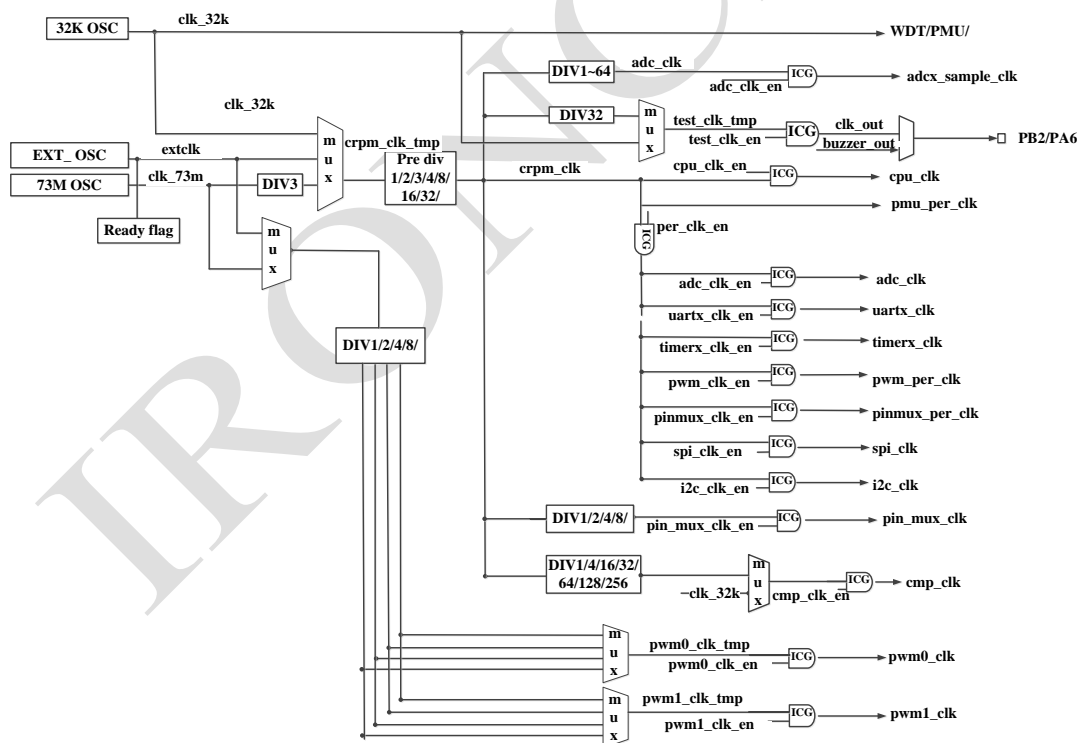


Figure 5 JMT18F003PLUS 时钟树

3.1.3 功能描述

3.1.3.1 时钟源

JMT18F003PLUS 时钟模块包含三个振荡器：输出频率为 32.768 KHz 的片内低频振荡器、

输出频率为 73.728MHz 的片内高频振荡器和最高频率为 24MHz 的外部振荡器。

3.1.3.1.1 片内 73.728MHz RC 振荡器

片内高频 RC 振荡器提供 73.728MHz 的时钟，它可以直接作为 PWM 时钟，由此时钟 3 分频得到的 24.576MHz 时钟可以作为系统时钟。

制造工艺决定了不同芯片的 RC 振荡器频率不同，为了减小 73.728MHz 时钟频率的误差，在芯片出厂前需要对振荡器进行校准，通过将校准值写入寄存器 TRIMVAL73M[6:0]，可以调整片内高频振荡器时钟的频率。出厂时，片内高频振荡器的时钟频率误差被校准到 1% 以内（25℃）。

片内高频振荡器可以由软件控制启动和停止，配置寄存器 OSCPD 的 OSC73MPD 位为 1，关闭片内高频振荡器，配置寄存器 OSCPD 的 OSC73MPD 位为 0，打开片内高频振荡器。

3.1.3.1.2 片内 32.768KHz RC 振荡器

片内低频 RC 振荡器提供 32.768KHz 时钟，它可以直接作为 PMU 和 WDT 模块的时钟，也可以作为系统时钟。它能够在不需要任何外部器件的条件下提供时钟。

片内低频振荡器只能由硬件控制启动和停止。芯片工作在 DEEP SLEEP 模式下，片内低频振荡器自动关闭，其他工作模式下，片内低频振荡器一直工作。

制造工艺决定了不同芯片的 RC 振荡器频率不同，通过配置寄存器 TRIMVAL32K[3:0] 可以调整片内低频振荡器的输出时钟频率，出厂时，片内低频振荡器的时钟频率误差被校准到 10% 以内（25℃）。

3.1.3.1.3 片外晶体振荡器

片外晶体振荡器可为系统提供更为精确的主时钟，相关的硬件配置可参考 Figure 6，进一步信息可参考电气特性部分的 18.2.7.3 章节。

片外晶体振荡器的开关受寄存器 OSCPD 的 OSC22MEN 位和寄存器 OSC22_CTRL 的 CLKOUTEN 位控制，当 OSCPD 的 OSC22MEN 位和 OSC22_CTRL 寄存器的 CLKOUTEN 位同时配置为 1 时，外部振荡器为芯片正常提供最高 24MHz 的时钟；当寄存器 OSCPD 的 OSC22MEN 位配置为 0 时，外部振荡器停止工作；当 OSC22_CTRL 寄存器的 CLKOUTEN 配置为 0 时，外部振荡器产生时钟被门控掉。

寄存器 CLK_READY 的 OSC22MRDY 位指示外部振荡器的输出时钟是否稳定。启动外部振荡器后，当时钟稳定时，寄存器 CLK_READY 的 OSC22MRDY 位被硬件置 1，此时时钟才能被使用。

可以通过配置寄存器 OSC_STABL 和寄存 OSC_STABH 决定等待外部振荡器时钟稳定的时间。

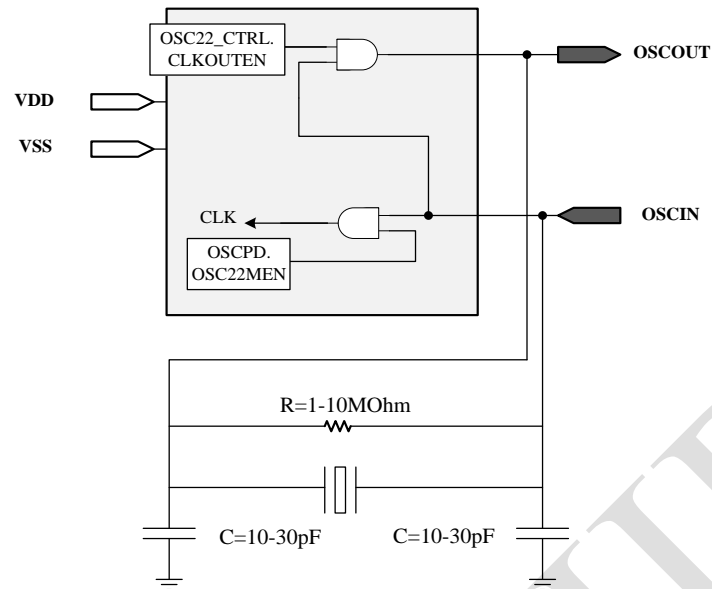


Figure 6 外部晶体振荡器连接示意图

3.1.3.2 系统时钟

系统时钟的最高工作频率为 24.576MHz，复位后默认工作时钟频率为 6.144MHz。系统时钟 sys_clk 可以通过以下两个步骤得到：

首先，做时钟选择。时钟源 MUX 可通过配置 SYS_CLK_SEL 寄存器选择以下三个时钟源：

- 片内 32.768KHz 振荡器输出的时钟。
- 片外振荡器输出的最高 24MHz 时钟。
- 片内 73.728MHz 振荡器产生的时钟做 3 分频得到的 24.567MHz 时钟。

该时钟源 MUX 默认选择 24.576MHz 时钟，可以配置寄存器 SYS_CLK_SEL 选择片外振荡器输出的时钟，步骤如下：

- 配置寄存器 OSCPD 的 OSC22MEN 为 1 并且配置寄存器 OSC22_CTRL 的 CLKOUTEN 位为 1，使能外部振荡器时钟。
- 读取寄存器 CLK_READY，检查 CLK_READY 的 OSC22MRDY 位，若 OSC22MRDY 为 0，不能切换时钟；若 OSC22MRDY 为 1，表示外部时钟稳定，此时可以配置 SYS_CLK_SEL 为 0x02，时钟切换至外部时钟。

可以配置寄存器 SYS_CLK_SEL 为 0x03 选择片内 32.768KHz 振荡器时钟作为系统时钟源。

其次，做时钟分频，过程如下：

- 时钟源 MUX 输出连接至分频器。
- 分频器可通过配置寄存器 SYS_CLK_DIV 得到 1/2/4/8/16/32 分频。

分频后即得到 sys_clk。

3.1.3.3 时钟输出

JMT18F003PLUS 允许输出时钟信号到外部引脚，产生过程如下：

首先，做时钟选择，可通过配置寄存器 CRMCTRL 选择以下三个时钟：

- 片内 32.768KHz 振荡器输出的时钟。
- 系统时钟 sys_clk。

复位时，默认选系统时钟 sys_clk 作为时钟输出。

其次，对选出的时钟做时钟门控，配置寄存器 CRMCTRL 的 TESTCEN 位为 1，打开时钟输出，配置寄存器 CRMCTRL 的 TESTCEN 位为 0，关闭时钟输出。

最后，对门控后的时钟做 32 分频。默认情况下，最终输出的时钟频率为 192KHz。

3.1.3.4 蜂鸣器控制

当 BUZZER_CTRL1 的 BUZEN 位配置为 1 时，管脚 PA1、PB0 和 PC6 输出蜂鸣器控制信号。

蜂鸣器控制信号在 32KHz 时钟的基础上通过分频得到，蜂鸣器控制信号的频率为：

$$f_{buzzer} = \frac{32KHz}{BUZZER_CTRL0[4:0] + BUZZER_CTRL1[4:0]},$$

占空比为：

$$duty = \frac{BUZZER_CTRL1[4:0]}{BUZZER_CTRL0[4:0] + BUZZER_CTRL1[4:0]},$$

例如，配置寄存器 BUZZER_CTRL0 为 0x04，配置寄存器 BUZZER_CTRL1 为 0x84，则蜂鸣器控制信号的频率为 4KHz，占空比为 50%；若配置寄存器 BUZZER_CTRL0 为 0x04，配置寄存器 BUZZER_CTRL1 为 0x8C，则蜂鸣器控制信号的频率为 2KHz，占空比为 75%，输出的波形如图 Figure 7 蜂鸣器输出波形图所示。

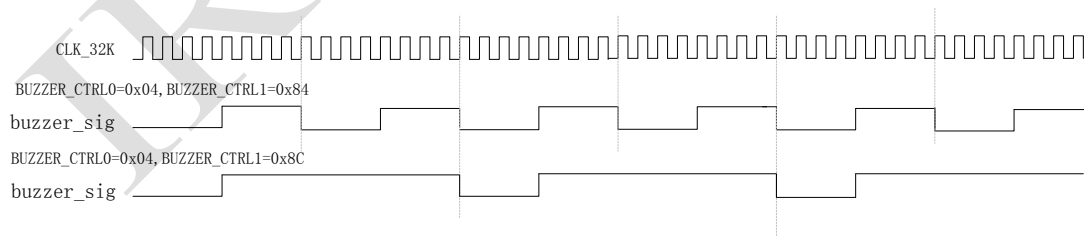


Figure 7 蜂鸣器输出波形图

3.1.3.5 JMT51 MCU 核时钟

JMT51 核时钟的最高工作频率为 24.576MHz，复位后默认工作时钟频率为 6.144MHz，JMT51 核时钟 jmt51_clk 由系统时钟 sys_clk 经过门控得到，该时钟门控由硬件根据芯片的低功耗工作模式自动控制，具体参见 2.2 节。

3.1.3.6 外设时钟

3.1.3.6.1 WDT 和 PMU 模块时钟

此两个模块的时钟频率为 32.768KHz，由片内 32.768KHz 振荡器直接提供。

3.1.3.6.2 AD 采样时钟和 AD 控制模块时钟

AD 采样时钟产生过程如下：

首先，做时钟分频：

- 将 sys_clk 连接至分频器。
- 分频器可通过配置寄存器 ADC_CLK_DIV 得到 sys_clk 时钟的 1~64 分频。

其次，对分频得到的时钟做门控：

- 将寄存器 CLK_EN1 的 ADCCEN 位配置为 1，打开 AD 采样时钟，配置为 0，关闭 AD 采样时钟。

AD 控制模块 ADC 的时钟 adc_clk 由系统时钟 sys_clk 经过门控得到，通过配置 CLK_EN1.ADCCEN 为 0，可关闭 adc_clk，将 CLK_EN1.ADCCEN 配置为 1，打开 adc_clk。

3.1.3.6.3 UART 时钟

UART 时钟 uart_clk 由系统时钟 sys_clk 经过门控得到，通过配置 CLK_EN0.UARTCEN 为 0，关闭 uart_clk，将 CLK_EN0.UARTCEN 配置为 1，打开 uart_clk。

3.1.3.6.4 TIMER 时钟

TIMER 时钟 timer_clk 由 sys_clk 经过门控得到，通过配置 CLK_EN0.TIMERCEN 为 0，关闭 timer_clk，将 CLK_EN0.TIMERCEN 配置为 1，打开 timer_clk。

3.1.3.6.5 PWMx (x=0, 1) 模块时钟

PWMx (x=0, 1) 模块的最高工作时钟为 73.728MHz，默认工作时钟为 73.728MHz，PWMx 模块的工作时钟产生过程如下：

首先，做时钟选择，时钟源 MUX 可通过配置 PWM_CLK_CTRL 的 PWMCLKSEL 寄存器选择以下两个时钟源：

- 片外振荡器输出时钟。
- 片内 73.728MHz 振荡器产生的时钟。

该时钟源 MUX 默认选择 73.728MHz 时钟，可以配置寄存器 PWM_CLK_CTRL 的 PWMCLKSEL 选择片外振荡器输出的时钟，步骤如下：

- 配置寄存器 OSCPD 的 OSC22MEN 为 1 并且配置寄存器 OSC22_CTRL 的 CLKOUTEN 位为 1，使能外部振荡器时钟。
- 读取寄存器 CLK_READY，检查 CLK_READY 的 OSC22MRDY 位，若 OSC22MRDY 为 0，不能切换时钟；若 OSC22MRDY 为 1，表示外部时钟稳定，此时可以配置 PWM_CLK_CTRL 的 PWMCLKSEL 位为 0x01，时钟切换至外部时钟。

其次，做时钟分频：

- 将选择后的时钟连接至分频器。
- PWM0 分频器可通过配置寄存器 PWM_CLK_CTRL 的 PWM0CLKDIV 位得到时钟的 1/2/4/8 分频。
- PWM1 分频器可通过配置寄存器 PWM_CLK_CTRL 的 PWM1CLKDIV 位得到时钟的 1/2/4/8 分频。

最后，做时钟门控：

- 将寄存器 CLK_EN1 的 PWM0CEN 位配置为 0，关闭 PWM0 模块时钟，配置为 1，打开 PWM0 模块时钟。
- 将寄存器 CLK_EN1 的 PWM1CEN 位配置为 0，关闭 PWM1 模块时钟，配置为 1，打开 PWM1 模块时钟。

3.1.3.6.6 GPIO 时钟

GPIO 模块的最高工作时钟为 24.576MHz，默认工作时钟为 6.144MHz，GPIO 模块的工作时钟 gpio_clk 产生过程如下：

首先，对 sys_clk 做时钟分频：

- 将时钟 sys_clk 链接至 GPIO 分频器。
- GPIO 分频器可通过配置寄存器 GPIO_CLK_DIV 得到 1/2/4/8/分频。

其次，对 GPIO 分频器输出的时钟做门控：

- 将 GPIO 分频器的输出连接至 GPIO 门控单元。
- 配置 CLK_EN0.GPIOCEN 为 0，关闭 gpio_clk，配置 CLK_EN0.GPIOCEN 为 1，打开 gpio_clk。

3.1.3.6.7 SPI 时钟

SPI 模块的最高工作时钟为 24.576MHz，默认工作时钟为 6.144MHz，SPI 模块的工作时钟 spi_clk 产生过程如下：

对系统时钟 sys_clk 做门控，配置 CLK_EN1.SPICEN 为 0，关闭 spi_clk，配置 CLK_EN1.SPICEN 为 1，打开 spi_clk。

3.1.3.6.8 I2C 时钟

I2C 模块的最高工作时钟为 24.576MHz，默认工作时钟为 6.144MHz，I2C 模块的工作时钟 i2c_clk 产生过程如下：

对系统时钟 sys_clk 做门控，配置 CLK_EN2.I2CCEN 为 0，关闭 i2c_clk，配置 CLK_EN2.I2CCEN 为 1，打开 i2c_clk。

3.1.3.6.9 模拟比较器滤波时钟

模拟比较器滤波时钟的最高工作时钟为 24.576MHz，默认工作时钟为 6.144MHz，模拟比较器滤波时钟产生过程如下：

首先，对 sys_clk 做时钟分频：

- 将时钟 sys_clk 链接至模拟比较器分频器。
- 分频器可通过配置寄存器 CMP_CLK_CTRL 得到 1/4/16/32/64/128/256 分频。
- 将寄存器 CMP_CLK_CTRL 的低 3 位配置为 111, 可以选择 32.768KHz 作为模拟比较器的滤波时钟。

其次, 对分频器输出的时钟做门控:

- 将分频器的输出连接至门控单元。
- 配置 CLKEN0.CMPCLKEN 为 0, 关闭 cmp_clk, 配置 CLKEN0.CMPCLKEN 为 1, 打开 cmp_clk。

3.1.4 寄存器描述

该模块共有 25 个寄存器, 对应的 SFR PAGE 为 2, 具体列表如下:

Table 5 时钟模块寄存器 (SFRPAGE=0x02)

地址	寄存器名	类型	复位值	功能描述
0xDA	SYS_CLK_SEL	W/R	0x00	系统时钟选择寄存器
0xD7	SYS_CLK_DIV	W/R	0x03	系统时钟分频系数选择寄存器
0xDB	ADC_CLK_DIV	W/R	0x09	AD 采样时钟分频系数选择寄存器
0xF2	GPIO_CLK_DIV	W/R	0x00	GPIO 时钟分频系数选择寄存器
0xF1	CMP_CLK_DIV	W/R	0x00	模拟比较器滤波时钟控制寄存器
0xED	PWM_CLK_CTRL	W/R	0x00	PWM0 时钟分频系数选择寄存器
0xE9	CLK_EN0	W/R	0x0F	时钟门控寄存器 0
0xEA	CLK_EN1	W/R	0x00	时钟门控寄存器 1
0xD6	OSCPD	W/R	0x00	振荡器时钟控制寄存器
0xE2	OSC24_CTRL	W/R	0x00	片外振荡器输出控制寄存器
0xE3	OSC_STABL	W/R	0xE0	外部振荡器稳定时间寄存器低 8 位
0xE4	OSC_STABH	W/R	0x2E	外部振荡器稳定时间寄存器高 8 位
0xE1	CLK_READY	R	0x00	时钟稳定标识寄存器

地址	寄存器名	类型	复位值	功能描述
0xCF	TRMVAL73M	W/R	0x40	73.728MHz 时钟校准值
0xD1	TRMVAL32K	W/R	0x08	32.768KHz 时钟校准值
0xD9	BANDGAP_CTRL	W/R	0x10	BANDGAP 校准值
0xD3	ADCTRIM	W/R	0x08	ADC 内部参考电压校准值
0xD2	CRPMCTRL	W/R	0x00	时钟输出控制寄存器
0xDC	JTAGEN	W/R	0x00	JTAG 模式使能寄存器
0xDE	BUZZER_CTRL0	W/R	0x01	蜂鸣器控制寄存器 0
0xDF	BUZZER_CTRL1	W/R	0x01	蜂鸣器控制寄存器 1
0xDD	TRIMKEY	W	0x00	时钟 trimming 启动寄存器
0xD4	TRIM_CNT_73ML	R	0x00	时钟 trimming 计数器低 8 位
0xD5	TRIM_CNT_73MH	R	0x00	时钟 trimming 计数器高 8 位
0xE5	FLTCKSEL	W/R	0x00	唤醒管脚滤波时钟选择寄存器。

3.1.4.1 系统时钟选择寄存器(SYS_CLK_SEL) [0xDA]

Table 6 系统时钟选择寄存器(SYS_CLK_SEL)

参数名	比特位	属性	复位值	描述
SYSCLKSEL	1-0	W/R	00	系统时钟选择寄存器： 00/01：选择片内 24.576MHz 时钟源； 10：选择片外晶振时钟（最高 24MHz）； 11：选择片内 32.768KHz 时钟。
Reserved	7-2	R	0	保留

3.1.4.2 系统时钟分频系数选择寄存器(SYS_CLK_DIV) [0xD7]

Table 7 系统时钟分频系数选择寄存器(SYS_CLK_DIV)

参数名	比特位	属性	复位值	描述
SYSCLKDIV	2-0	W/R	011	系统时钟分频系数选择寄存器： 000：1 分频； 001：2 分频； 010：4 分频； 011：8 分频；

参数名	比特位	属性	复位值	描述
				100: 16 分频; 其他: 32 分频。
Reserved	7-3	R	0	保留

3.1.4.3 AD 采样时钟分频系数选择寄存器(ADC_CLK_DIV) [0xDB]

Table 8 AD 采样时钟分频系数选择寄存器(ADC_CLK_DIV)

参数名	比特位	属性	复位值	描述
ADCCLKDIV	5-0	W/R	001001	AD 采样时钟分频系数选择: 0-63: 对系统时钟 sys_clk 进行 1-64 分频, 得到 AD 采样时钟。
Reserved	7-6	R	0	保留

3.1.4.4 GPIO 时钟分频系数选择寄存器(GPIO_CLK_DIV) [0xF2]

Table 9 GPIO 时钟分频系数选择寄存器(GPIO_CLK_DIV)

参数名	比特位	属性	复位值	描述
GPIOCLKDIV	1-0	W/R	000	GPIO 模块时钟分频系数选择寄存器: 00: 1 分频; 01: 2 分频; 10: 4 分频; 11: 8 分频;
Reserved	7-3	R	0	保留

3.1.4.5 模拟比较器滤波时钟控制寄存器(CMP_CLK_DIV) [0xF1]

Table 10 模拟比较器滤波时钟控制寄存器(CMP_CLK_DIV)

参数名	比特位	属性	复位值	描述
CMPCLKDIV	2-0	W/R	000	模拟比较器滤波时钟分频系数选择寄存器 000: sys_clk 1 分频; 001: sys_clk 4 分频; 010: sys_clk 16 分频; 011: sys_clk 32 分频; 100: sys_clk 64 分频; 101: sys_clk 128 分频。 110: sys_clk 256 分频。

参数名	比特位	属性	复位值	描述
				111: 选择 32.768KHz 时钟作为模拟比较器滤波时钟。
Reserved	7-3	R	0	保留

3.1.4.6 PWM 时钟分频系数选择寄存器(PWM_CLK_CTRL) [0xED]

Table 11 PWM0 时钟分频系数选择寄存器(PWM_CLK_CTRL)

参数名	比特位	属性	复位值	描述
PWMCLKSEL	0	W/R	0	PWM 时钟选择寄存器: 0: 选择片内 73.728MHz 时钟作为 PWM 时钟; 1: 选择外部晶振时钟作为 PWM 时钟。
PWM0CLKDIV	2-1	W/R	00	PWM0 模块时钟分频系数选择寄存器: 00: 1 分频; 01: 2 分频; 10: 4 分频; 11: 8 分频;
PWM1CLKDIV	4-3	W/R	00	PWM1 模块时钟分频系数选择寄存器: 00: 1 分频; 01: 2 分频; 10: 4 分频; 11: 8 分频;
Reserved	7-5	R	0	保留

3.1.4.7 时钟门控寄存器 0(CLK_EN0) [0xE9]

Table 12 时钟门控寄存器 0(CLK_EN0)

参数名	比特位	属性	复位值	描述
CMPCEN	0	W/R	1	比较器滤波时钟门控: 1: 打开输出时钟; 0: 关闭输出时钟。
UARTCEN	1	W/R	1	UART 模块时钟门控: 1: 打开 UART0 模块时钟; 0: 关闭 UART0 模块时钟。

参数名	比特位	属性	复位值	描述
TIMERCEN	2	W/R	1	TIMER 模块时钟门控： 1：打开 TIMER 模块时钟； 0：关闭 TIMER 模块时钟。
GPIOCEN	3	W/R	1	GPIO 模块时钟门控： 1：打开 GPIO 模块时钟； 0：关闭 GPIO 模块时钟。
Reserved	7-4	R	0	保留

3.1.4.8 时钟门控寄存器 1(CLK_EN1) [0xEA]

Table 13 时钟门控寄存器 1(CLK_EN1)

参数名	比特位	属性	复位值	描述
I2CCEN	0	W/R	0	I2C 模块时钟门控： 1：打开 I2C 模块时钟 0：关闭 I2C 模块时钟
PWM0CEN	1	W/R	0	PWM0 模块时钟门控： 1：打开 PWM 0 模块时钟； 0：关闭 PWM 0 模块时钟。
PWM1CEN	2	W/R	0	PWM1 模块时钟门控： 1：打开 PWM 1 模块时钟； 0：关闭 PWM 1 模块时钟。
SPICEN	3	W/R	0	SPI 模块时钟门控： 1：打开 SPI 模块时钟 0：关闭 SPI 模块时钟
ADCCEN	4	W/R	1	ADC 模块时钟门控： 1：打开 ADC 模块时钟； 0：关闭 ADC 模块时钟。
Reserved	7-5	R	0	保留

3.1.4.9 振荡器时钟控制寄存器(OSCPD) [0xD6]

Table 14 振荡器时钟控制寄存器(OSCPD)

参数名	比特位	属性	复位值	描述
Reserved	0	R	0	保留
OSC73MPD	1	W/R	0	片内 73.728MHz 振荡器使能：

参数名	比特位	属性	复位值	描述
				0: 片内 73.728MHz 振荡器正常工作; 1: 关闭片内 73.728MHz 振荡器。
OSC24MEN	2	W/R	0	片外振荡器使能: 0: 片外 OSC 不使能; 1: 片外 OSC 使能。
Reserved	7-3	R	0	保留

3.1.4.10 片外晶体振荡器输出控制寄存器(OSC24_CTRL) [0xE2]

Table 15 片外晶体振荡器输出控制寄存器(OSC24_CTRL)

参数名	比特位	属性	复位值	描述
CLKOUTEN	0	W/R	0	片外振荡器时钟门控: 0: 关闭片外振荡器时钟; 1: 打开片外振荡器时钟。
Reserved	7-1	R	0	保留

3.1.4.11 片外晶体振荡器稳定时间寄存器低 8 位(OSC_STABL) [0xE3]

Table 16 片外晶体振荡器稳定时间寄存器低 8 位(OSC_STABL)

参数名	比特位	属性	复位值	描述
OSCSTBTL	7-0	W/R	0xe0	片外晶体振荡器稳定时间寄存器低 8 位。

3.1.4.12 片外晶体振荡器稳定时间寄存器高 8 位(OSC_STABH) [0xE4]

Table 17 片外晶体振荡器稳定时间寄存器高 8 位(OSC_STABH)

参数名	比特位	属性	复位值	描述
OSCSTBTH	7-0	W/R	0x2e	片外晶体振荡器稳定时间寄存器高 8 位。

3.1.4.13 时钟稳定标识寄存器(CLK_READY) [0xE1]

Table 18 时钟稳定标识寄存器(CLK_READY)

参数名	比特位	属性	复位值	描述
OSC24MRDY	0	R	0	片外晶体振荡器时钟稳定标识寄存器。 0: 片外晶体振荡器时钟未稳定; 1: 片外晶体振荡器时钟稳定。
Reserved	7-1	R	0	保留

3.1.4.14 73.728MHz 时钟校准值(TRMVAL73M) [0xCF]

Table 19 73.728MHz 时钟校准值(TRMVAL73M)

参数名	比特位	属性	复位值	描述
TRMVAL73M	6-0	W/R	1000000	73.728MHz 时钟校准时写入的校准值。
Reserved	7	R	0	保留

3.1.4.15 32.768KHz 时钟校准值(TRMVAL32K) [0xD1]

Table 20 32.768KHz 时钟校准值(TRMVAL32K)

参数名	比特位	属性	复位值	描述
TRMVAL32K	3-0	W/R	1000	32.768KHz 时钟校准时写入的校准值。
Reserved	7-4	R	0	保留

3.1.4.16 BANDGAP 输出值校准(BANDGAP_CTRL) [0xD9]

Table 21 BANDGAP 输出值校准(BANDGAP_CTRL)

参数名	比特位	属性	复位值	描述
BGTRIM	4-0	W/R	10000	BANDGAP 输出值校准。
Reserved	7-5	R	0	保留

3.1.4.17 ADC 参考电压校准值寄存器(ADCTRIM) [0xD3]

Table 22 ADC 参考电压校准值寄存器(ADCTRIM)

参数名	比特位	属性	复位值	描述
ADCTRIM	3-0	W/R	1000	ADC 参考电压校准值寄存器。
Reserved	7-4	R	0	保留

3.1.4.18 CRM 模块控制寄存器(CRMCTRL) [0xD2]

Table 23 CRM 模块控制寄存器(CRMCTRL)

参数名	比特位	属性	复位值	描述
TRIM32KSEL	0	W/R	0	时钟 trimming 32.768k 时钟选择： 0：选择外部晶振频率为 22.1184MHz 时，经分频得到的 32.768KHz 时钟； 1：选择片内 32.768KHz 时钟。
BORENCTRL	1	W/R	0	BOR 控制方式寄存器： 0：由软件配置 BORSWEN 寄存器，控制 BOR 模块的使能； 1：由硬件控制 BOR 模块的使能。

参数名	比特位	属性	复位值	描述
				建议使用软件配置 BORSWEN 寄存器，使 BOR 一直处于工作状态。
BORSWEN	2	W/R	0	BOR 模块使能寄存器： 0: BOR 模块正常工作； 1: 关闭 BOR 模块。
LVRSTEN	3	W/R	0	LVR 复位信号使能控制： 0: LVR 复位信号无效； 1: LVR 复位信号有效。
TESTCLKSEL	5-4	W/R	00	输出测试时钟选择寄存器： 00/10: 选择系统时钟作为输出时钟； 其他: 选择 32KHz 时钟作为输出时钟；
TESTCEN	6	W/R	0	输出测试时钟使能寄存器： 1: 打开测试时钟； 0: 关闭测试时钟。
Reserved	7	R	0	保留

3.1.4.19 JTAG 使能寄存器(JTAGEN) [0xDC]

Table 24 ADC 参考电压校准值寄存器(JTAGEN)

参数名	比特位	属性	复位值	描述
JTAGEN	0	W/R	1	JTAG 使能寄存器： 0: 关闭 JTAG 功能； 1: 打开 JTAG 功能。
Reserved	7-1	R	0	保留

3.1.4.20 蜂鸣器控制寄存器 0 (BUZZER_CTRL0) [0xDE]

Table 25 蜂鸣器控制寄存器 0 (BUZZER_CTRL0)

参数名	比特位	属性	复位值	描述
LOW_DUTY	4-0	W/R	1	蜂鸣器控制信号中低电平持续的 32.768KHz 时钟数
RESERVED	7-5	R	0	保留

3.1.4.21 蜂鸣器控制寄存器 1 (BUZZER_CTRL1) [0x DF]

Table 26 蜂鸣器控制寄存器 1 (BUZZER_CTRL1)

参数名	比特位	属性	复位值	描述
HIGH_DUTY	4-0	W/R	1	蜂鸣器控制信号高电平持续的 32.768KHz 时钟数
RESERVED	6-5	R	0	保留
BUZZER_EN	7	W/R	0	蜂鸣器使能寄存器: 0: 关闭蜂鸣器; 1: 使能蜂鸣器。

3.1.4.22 时钟 trimming 启动寄存器 (TRIMKEY) [0x DD]

Table 27 时钟 trimming 启动寄存器(TRIMKEY)

参数名	比特位	属性	复位值	描述
TRIMKEY	7-0	W	0x00	时钟 trimming 启动寄存器: 先配置 0xA5 再配置 0xF1 时启动时钟 trimming. 注: trimming 在出厂前已经完成, 不建议用于配置。

3.1.4.23 时钟 trimming 计数寄存器 低 8 位(TRIM_CNT_73ML) [0x D4]

Table 28 时钟 trimming 计数寄存器 低 8 位(TRIM_CNT_73ML)

参数名	比特位	属性	复位值	描述
TRIM_CNT_73ML	7-0	R	0x00	时钟 trimming 计数寄存器 低 8 位. 注: trimming 在出厂前已经完成, 不建议用于配置。

3.1.4.24 时钟 trimming 计数寄存器 高 8 位(TRIM_CNT_73MH) [0x D5]

Table 29 时钟 trimming 计数寄存器 高 8 位(TRIM_CNT_73MH)

参数名	比特位	属性	复位值	描述
TRIM_CNT_73MH	7-0	R	0x00	时钟 trimming 计数寄存器 高 8 位. 注: trimming 在出厂前已经完成, 不建议用于配置。

参数名	比特位	属性	复位值	描述
				配置。

3.1.4.25 唤醒管脚滤波时钟选择寄存器(FLTCKSEL) [0xE5]

Table 30 唤醒管脚滤波时钟选择寄存器(FLTCKSEL)

参数名	比特位	属性	复位值	描述
FLTCKSEL	0	W/R	0	唤醒管脚滤波时钟选择寄存器： 0：选择系统时钟作为唤醒管脚的滤波时钟； 1：选择 32.768KHz 时钟作为唤醒管脚的滤波时钟。
Reserved	7-1	R	0	保留

3.2 复位管理

3.2.1 概述

JMT18F003PLUS 有 7 种复位方式：

- 上电复位（Power On Reset）
- 掉电复位（Brown Out Reset）
- 管脚硬复位（Hardware Reset）
- 低电压检测复位（LVR Reset）
- 看门狗复位（Watch Dog Reset）
- 全局软复位（Global Software Reset）
- 电源分区复位（Power Reset）
- 模块软复位（Module Software Reset）

JMT18F003PLUS 中各个复位信号均为低有效。将各个复位源信号（低有效）送入 CRM 模块，在 CRM 模块产生各模块的复位信号。只要上电复位、掉电复位、管脚硬复位、低电压检测复位、看门狗复位和全局软复位中的任何一个复位源变为低电平，CRM 模块均会为各模块产生复位信号。退出 SLEEP 或 DEEP SLEEP 低功耗模式后，PMU 模块产生电源分区复位，对断电的模块进行复位。

3.2.2 框图

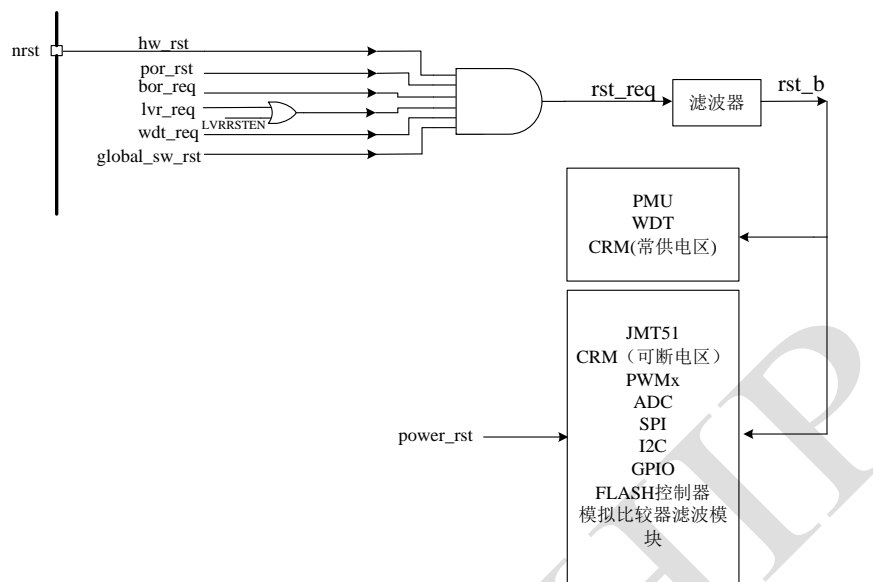


Figure 8 JMT18F003PLUS 复位

3.2.3 功能描述

3.2.3.1 上电复位

上电复位（Power On Reset）：上电后，芯片内部产生上电复位信号送到 CRM 模块，经过滤波后产生全芯片复位信号，对全芯片所有模块进行复位，复位结束后芯片进入 boot 流程。

3.2.3.2 掉电复位(BOR)

掉电复位电路监控主电源 VDD，当 VDD 发生掉电，电压小于限位电压 V_{BOR} 时，掉电复位信号有效。

可选择由硬件或软件控制掉电复位检测电路，配置寄存器 CRMCTRL.BORENCTRL 为 1，由硬件控制掉电复位检测电路的启停；当 CRMCTRL.BORENCTRL 配置为 0 时，由软件控制掉电复位检测电路的启停。默认为软件控制掉电复位检测电路的启停。

若掉电复位检测电路由软件控制，当 CRMCTRL.BORSWEN 配置为 0，打开掉电复位检测电路，当 CRMCTRL.BORSWEN 配置为 1，关闭掉电复位检测电路，复位后，掉电复位检测电路默认打开。

若掉电复位检测电路由硬件控制，芯片工作在 DEEP SLEEP 模式下，关闭掉电复位检测电路，其他工作模式下，掉电复位检测电路正常工作。

当掉电复位信号产生后，CRM 模块为各个模块产生复位信号，对全芯片进行复位，复位结束后，JMT51 核从 FLASH 的 0 地址开始执行应用程序。

3.2.3.3 管脚硬复位

硬复位信号由管脚 `nrst` 输入，通过配置 `NRSTFQS` 寄存器的 `NRSTFQS` 位可以选择对输入的硬复位信号进行滤波，硬复位输入到 `CRM` 模块，`CRM` 模块为各个模块产生复位信号，对全芯片进行复位，复位结束后芯片进入 `boot` 流程。

3.2.3.4 低电压检测复位

低电压检测电路对全芯片电压 `VDD` 进行监控，当电压小于可编程的限位电压 V_{LVR} 时，低电压检测电路发出复位信号。

为了防止检测电路产生毛刺，在检测电路中增加了滤波功能。

低电压检测电路产生复位请求信号后，信号进入 `CRM` 模块，在 `CRM` 模块为低电压复位信号设计了使能信号，若寄存器 `CRMCTRL` 的 `LVRSTEN` 位配置为 0，复位请求信号被屏蔽，不会对全芯片进行复位；若寄存器 `CRMCTRL` 的 `LVRSTEN` 位配置为 1，则低电压检测电路发送过来的复位请求信号有效，`CRM` 模块为各个模块产生复位信号，复位结束后，`JMT51` 核从 `FLASH` 的 0 地址开始执行应用程序。

3.2.3.5 看门狗(WDT)复位

上电复位后，`WDT` 默认处于工作状态，当 `WDT` 产生复位请求信号后，将复位请求信号送入 `CRM` 模块，`CRM` 模块为各个模块产生复位信号，对全芯片进行复位，复位结束后，`JMT51` 核从 `FLASH` 的 0 地址开始执行应用程序。

3.2.3.6 全局软复位

`JMT18F003PLUS` 支持软件配置全局复位功能，将 `GLOBALRST` 寄存器配置为 0x01，对全芯片进行复位，硬件自动清除 `GLOBALRST` 寄存器，复位结束后，`JMT51` 核从 `FLASH` 的 0 地址开始执行应用程序。

3.2.3.7 电源分区复位

在 `SLEEP` 或 `DEEP SLEEP` 模式下，可断电区域的模块断电，重新唤醒后，芯片退出 `SLEEP` 或 `DEEP SLEEP` 模式，产生电源分区复位，对断电区域进行复位，`JMT51` 核从 `FLASH` 的 0 地址开始执行应用程序。

3.2.3.8 模块软复位

各个模块均具有了软件复位功能，当某模块的复位寄存器置 1 后，将会产生复位信号对该模块进行复位，硬件自动清除该复位寄存器。软件复位寄存器的详细描述参见 Table 35 和 Table 36。

3.2.3.9 复位信号对各模块的作用

表 Table 31 列出了复位信号对各模块的复位情况，表中(1)表示电源分区复位只对 `CRM`

模块的部分单元复位，其中复位的单元和寄存器有：

- GPIO 时钟分频寄存器 GPIO_CLK_DIV、时钟使能寄存器 CLK_EN1.GPIOCEN 和 GPIO 时钟产生单元。
- SPI 时钟分频寄存器 SPI_CLK_DIV、时钟使能寄存器 CLK_EN2.SPICEN 和 SPI 时钟产生单元。
- I2C 时钟分频寄存器 I2C_CLK_DIV、时钟使能寄存器 CLK_EN2.I2CCEN 和 SPI 时钟产生单元。
- 模拟比较器滤波时钟控制寄存器 CMP_CLK_CTRL 和模拟比较器滤波时钟产生单元。
- PWM0 时钟分频寄存器 PWM_CLK_DIV、时钟使能寄存器 CLK_EN2.PWM0CEN 和 PWM0 时钟产生单元。
- PWM1 时钟分频寄存器 PWM_CLK_DIV、时钟使能寄存器 CLK_EN2.PWM1CEN 和 PWM1 时钟产生单元。
- TIMER 和 UART 的时钟使能寄存器及时钟门控单元。

表中（2）表示硬复位不对 nrst 管脚的滤波模块和滤波控制寄存器 NRSTFQS 复位，对 GPIO 的其他的部分复位。

Table 31 复位信号对各模块的作用一览表

模块	POR_ RST	HW_ RST	BOR_ RST	LVR_ RST	WDT_ RST	GALOBAL_ RST	POWER_ RST
PMU 模块	复位	复位	复位	复位	复位	复位	不复位
WDT 模块	复位	复位	复位	复位	复位	复位	不复位
CRM 模块	复位	复位	复位	复位	复位	复位	部分复位 (1)
GPIO 模块	复位	部分复位 (2)	复位	复位	复位	复位	复位
PWM 模块	复位	复位	复位	复位	复位	复位	复位
JMT51 模块	复位	复位	复位	复位	复位	复位	复位
SPI 模块	复位	复位	复位	复位	复位	复位	复位
I2C 模块	复位	复位	复位	复位	复位	复位	复位
ADC 模块	复位	复位	复位	复位	复位	复位	复位
FLASH 控制 模块	复位	复位	复位	复位	复位	复位	复位
模拟比较器	复位	复位	复位	复位	复位	复位	复位

模块	POR_ RST	HW_ RST	BOR_ RST	LVR_ RST	WDT_ RST	GALOBAL_ RST	POWER_ RST
滤波模块							
是否进入 boot 流程	是	是	否	否	否	否	否

3.2.4 寄存器

该模块共有 5 个寄存器，对应的 SFR PAGE 为 2，具体列表如下。

Table 32 复位模块寄存器(SFRPAGE = 0x02)

地址	寄存器名	属性	复位值	功能描述
0xD2	CRMCTRL	W/R	0x04	CRM 模块控制
0xCE	RSTFLAG	W/R	0x20	复位标志
0xCB	SWRST0	W	0x00	模块软复位 0
0xCC	SWRST1	W	0x00	模块软复位 1
0xCD	GLOBALRST	W/R	0x00	全局软复位

3.2.4.1 CRM 模块控制寄存器(CRMCTRL) [0xD2]

Table 33 CRM 模块控制寄存器(CRMCTRL)

参数名	比特位	属性	复位值	描述
TRIM32KSEL	0	W/R	0	时钟 trimming 32.768k 时钟选择： 0: 选择外部晶振频率为 22.1184MHz 时，经分频得到的 32.768KHz 时钟； 1: 选择片内 32.768KHz 时钟。
BORENCTRL	1	W/R	0	BOR 控制方式寄存器： 0: 由软件配置 BORSWEN 寄存器，控制 BOR 模块的使能； 1: 由硬件控制 BOR 模块的使能。 建议使用软件配置 BORSWEN 寄存器，使 BOR 一直处于工作状态。
BORSWEN	2	W/R	0	BOR 模块使能寄存器： 0: BOR 模块正常工作； 1: 关闭 BOR 模块。
LVRIRSTEN	3	W/R	0	LVR 复位信号使能控制： 0: LVR 复位信号无效；

参数名	比特位	属性	复位值	描述
				1: LVR 复位信号有效。
TESTCLKSEL	5-4	W/R	00	输出测试时钟选择寄存器: 00/10: 选择系统时钟作为输出时钟; 其他: 选择 32KHz 时钟作为输出时钟;
TESTCEN	6	W/R	0	输出测试时钟使能寄存器: 1: 打开测试时钟; 0: 关闭测试时钟。
Reserved	7	R	0	保留

3.2.4.2 复位标识寄存器(RSTFLAG) [0xCE]

Table 34 复位标识寄存器(RSTFLAG)

参数名	比特位	属性	复位值	描述
WDTRSTFLAG	0	W/R	0	看门狗 (WDT) 复位标志位: 0: 没有 WDT 复位请求; 1: 有 WDT 复位请求; 软件写 0 清除该标志位。
BORRSTFLAG	1	W/R	0	掉电复位 (BOR) 标志位: 0: 没有 BOR 复位请求; 1: 有 BOR 复位请求; 软件写 0 清除该标志位。
LVRRSTFLAG	2	W/R	0	低电压复位 (LVR) 标志位: 0: 没有 LVR 复位请求; 1: 有 LVR 复位请求; 软件写 0 清除该标志位。
HWRSTFLAG	3	W/R	0	硬复位标志位: 0: 没有硬复位请求; 1: 有硬复位请求; 软件写 0 清除该标志。
GLBRSTFLAG	4	W/R	0	全局软复位标识位: 0: 没有全局软复位请求; 1: 有全局软复位请求; 软件写 0 清除该标识位。

参数名	比特位	属性	复位值	描述
PORRSTFLAG	5	W/R	1	上电复位 (POR_RST) 标识位: 0: 没有 POR 复位请求; 1: 有 POR 复位请求; 软件写 0 清除该标识位。
POWRSTFLAG	6	W/R	0	电源分区复位标识位: 0: 没有电源分区复位请求; 1: 有电源分区复位请求; 软件写 0 清除该标识位。
Reserved	7	R	0	保留

3.2.4.3 模块软复位寄存器 0(SWRST0) [0xCB]

Table 35 模块软复位寄存器 0(SWRST0)

参数名	比特位	属性	复位值	描述
WDTSWRST	0	W	0	WDT 模块软复位寄存器: 0: 无效; 1: 对 WDT 模块进行复位; 硬件自动将此位清 0。
Reserved	1	R	0	保留
PWM0SWRST	2	W	0	PWM0 模块软复位寄存器: 0: 无效; 1: 对 PWM0 模块进行复位; 硬件自动将此位清 0。
PWM1SWRST	3	W	0	PWM1 模块软复位寄存器: 0: 无效; 1: 对 PWM1 模块进行复位; 硬件自动将此位清 0。
Reserved	4	R	0	保留
SPISWRST	5	W	0	SPI 模块软复位寄存器: 0: 无效; 1: 对 SPI 模块进行复位; 硬件自动将此位清 0。
I2CSWRST	6	W	0	I2C 模块软复位寄存器:

参数名	比特位	属性	复位值	描述
				0: 无效; 1: 对 I2C 模块进行复位; 硬件自动将此位清 0。
Reserved	7	R	0	保留

3.2.4.4 模块软复位寄存器 1(SWRST1) [0xCC]

Table 36 模块软复位寄存器 1(SWRST1)

参数名	比特位	属性	复位值	描述
GPIO SWRST	0	W	0	GPIO 模块软复位寄存器: 0: 无效; 1: 对 GPIO 模块进行复位; 硬件自动将此位清 0。
ADC SWRST	1	W	0	ADC 模块软复位寄存器: 0: 无效; 1: 对 ADC 模块进行复位; 硬件自动将此位清 0。
Reserved	7-2	R	0	保留

3.2.4.5 全局软复位寄存器(GLOBALRST) [0xCD]

Table 37 全局软复位寄存器(GLOBALRST)

参数名	比特位	属性	复位值	描述
GLOB SWRST	0	W	0	全芯片软复位寄存器: 0: 无效; 1: 对全芯片进行复位; 硬件自动将此位清 0。
Reserved	7-1	R	0	保留

4 JMT51 MCU 核

4.1 概述

JMT51 MCU 内核为增强型 51 内核，与 MCS-51 指令集完全兼容，可以使用标准 803x/805x 的汇编器和编译器进行软件开发。JMT51 采用流水线结构，与标准的 8051 结构相比指令执行速度有很大的提高。标准 8051，除 MUL 和 DIV 以外的所有指令都需要 12 或 24 个系统时钟周期，对于 JMT51 内核，大部分指令的执行时间为 1 或 2 个系统时钟周期。JMT51 时钟最高频率为 24.576MHz。

JMT51 MCU 支持通过 JTAG 接口对 JMT51 核进行调试。具体使用方法可参考 JMT18F003PLUS JTAG 使用说明文档。需要说明的是，在 FLASH 的 NVR0 中存放了 JTAG 的两个使能标志（地址 0x2002 和 0x2003）。上电之后，Boot 程序会比较该使能标志，如果不等于 0x5A 和 0xA5，则将管脚 PA0、PA1、PA2、PA3 分别复用为 TCK、TMS、TDI、TDO。如果等于 0x5A 和 0xA5，PA0、PA1、PA2、PA3 四个管脚则被复用为正常功能。因此在进行量产的时候，可通过修改 NVR0 中的 JTAG 使能标志，将管脚复用为正常功能。

JMT51 MCU 内核堆栈位于内部 128 字节的数据存储器，因此堆栈深度最大可以达到 128。通过堆栈指针 SP 指示当前堆栈位置，当子程序调用或中断响应时，PC 会被压入堆栈中。在子程序或中断服务程序结束后，通过 RET 或 RETI 指令，PC 会重新从堆栈中获取到原先的值。如果数据存储器被用作堆栈区域，将不能作为通用数据存储器使用。

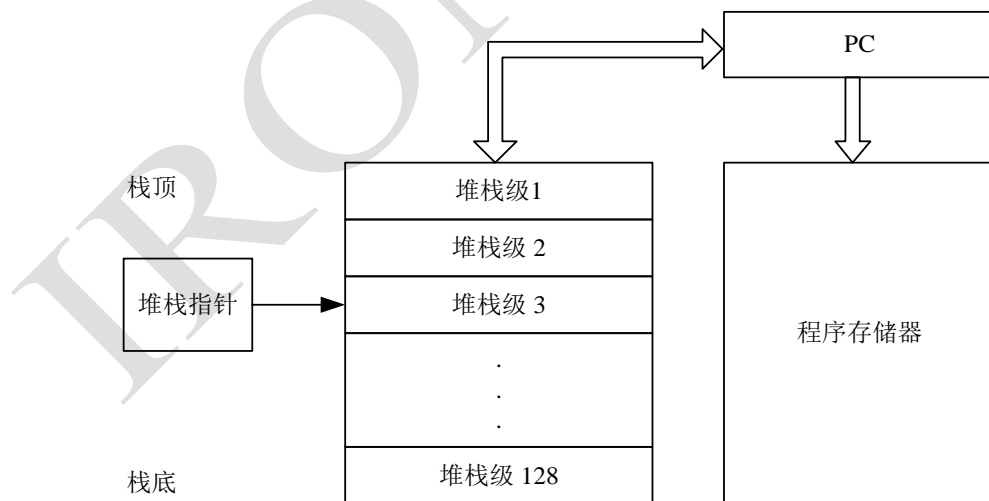


Figure 9 堆栈示意图

4.2 指令集

JMT51 MCU 核指令与标准 51 指令集完全兼容，可以使用标准 8051 的开发工具开发 JMT51 的软件。所有的 JMT51 指令在二进制码和功能上与同类的 MCS-51 产品完全等价，

包括操作码、寻址方式和对 PSW 标志的影响。

指令共分为五大类，算术运算类、逻辑运算类、数据传输类、程序控制类、布尔处理类：

- 算术运算类包括加法、带进位加法、带借位减法、自增、自减及乘除法指令，运算都是按字节进行的。
- 逻辑运算类包含按位与、按位或、按位异或、左移、右移、对累加器清 0、取反等操作。
- 数据传输类包括立即数到直接地址、间接地址、外部数据存储器的传输，内部数据存储器到外部数据存储器之间的传输，内部数据存储器到程序存储器之间的传输，压栈/出栈。
- 程序控制类指令包括绝对跳转、相对跳转、子程序调用、有条件跳转、子程序返回。
- 布尔处理类指令包括位清 0、置 1、取反、与、或、传输等。

Table 38 指令集中符号说明

Rn	通用寄存器 R0~R7
Direct	8 比特直接地址，内部数据存储器低 128 字节和 SFR 区域地址 00~FFH
@Ri	间接寻址，Ri 中存放的是内部数据存储器的地址
#data	8 位立即数
#data 16	16 位立即数
addr 16	16 位地址，用于跳转或调用，指示整个 32K 字节程序空间
addr 11	11 位地址，用于跳转或调用，地址范围为 2K 字节
Rel	相对跳转 8 位有符号地址，地址范围-128~+127
Bit	位地址，内部数据存储器 128 位和 SFR 中可位寻址区域

Table 39 算术类指令

助记符	描述	指令码	字节
ADD A,Rn	寄存器与 A 相加	28~2F	1
ADD A,direct	直接地址中的数与 A 相加	25	2
ADD A,@Ri	间接地址中的数与 A 相加	26~27	1
ADD A,#data	立即数与 A 相加	24	2
ADDC A,Rn	寄存器与 A 带进位相加	38~3F	1
ADDC A,direct	直接地址中的数与 A 带进位相加	35	2
ADDC A,@Ri	间接地址中的数与 A 带进位相加	36~37	1
ADDC A,#data	立即数与 A 带进位相加	34	2
SUBB A,Rn	A 带借位减寄存器	98~9F	1
SUBB A,direct	A 带借位减直接地址中的数	95	2

助记符	描述	指令码	字节
SUBB A,@Ri	A 带借位减间接地址中的数	96~97	1
SUBB A,#data	A 带借位减立即数	94	2
INC A	A 自增 1	04	1
INC Rn	寄存器中的数自增 1	08~0F	1
INC direct	直接地址中的数自增 1	05	2
INC @Ri	间接地址中的数自增 1	06~07	1
INC DPTR	数据指针自增 1	A3	1
DEC A	A 自减 1	14	1
DEC Rn	寄存器中的数自减 1	18~1F	1
DEC direct	直接地址中的数自减 1	15	2
DEC @Ri	间接地址中的数自减 1	16~17	1
MUL AB	A 与 B 相乘	A4	1
DIV	A 除与 B	84	1
DA A	A 中数十进制调整	D4	1

Table 40 逻辑运算类指令

助记符	描述	指令码	字节
ANL A,Rn	A 与寄存器按位与	58~5F	1
ANL A,direct	A 与直接地址中的数按位与	55	2
ANL A,@Ri	A 与间接地址中的数按位与	56~57	1
ANL A,#data	A 与立即数按位与	54	2
ANL direct,A	直接地址与 A 按位与	52	2
ANL direct,#data	直接地址与立即数按位与	53	3
ORL A,Rn	A 与寄存器按位或	48~4F	1
ORL A,direct	A 与直接地址中的数按位或	45	2
ORL A,@Ri	A 与间接地址中的数按位或	46~47	1
ORL A,#data	A 与立即数按位或	44	2
ORL direct,A	直接地址与 A 按位或	42	2
ORL direct,#data	直接地址与立即数按位或	43	3
XRL A,Rn	A 与寄存器按位异或	68~6F	1
XRL A,direct	A 与直接地址中的数按位异或	65	2
XRL A,@Ri	A 与间接地址中的数按位异或	66~67	1

助记符	描述	指令码	字节
XRL A,#data	A 与立即数按位异或	64	2
XRL direct,A	直接地址与 A 按位异或	62	2
XRL direct,#data	直接地址与立即数按位异或	63	3
CLR A	A 清 0	E4	1
CPL A	A 按位取反	F4	1
RL A	A 左移 1 位	23	1
RLC A	A 带进位左移 1 位	33	1
RR A	A 右移 1 位	03	1
RRC A	A 带进位右移 1 位	13	1
SWAP A	A 中高/低半字节数据交换	C4	1

Table 41 数据传输类指令

助记符	描述	指令码	字节
MOV A,Rn	寄存器数到 A	E8~EF	1
MOV A,direct	直接地址中的数到 A	E5	2
MOV A,@Ri	间接地址中的数到 A	E6~E7	1
MOV A,#data	立即数数到 A	E4	2
MOV Rn,A	A 到寄存器	F8~FF	1
MOV Rn,direct	直接地址到寄存器	A8~AF	2
MOV Rn,#data	立即数到寄存器	78~7F	2
MOV direct,A	A 到直接地址	F5	2
MOV direct,Rn	寄存器到直接地址	88~8F	2
MOV direct1,direct2	直接地址 2 到直接地址 1	85	3
MOV direct,@Ri	间接地址中的数到直接地址	86~87	2
MOV direct,#data	立即数到直接地址	75	3
MOV @Ri,A	A 到间接地址	F6~F7	1
MOV @Ri,direct	直接地址中的数到间接地址	A6~A7	2
MOV @Ri,#data	立即数到间接地址	76~77	2
MOV DPTR,#data16	16 位立即数到 DPTR	90	3
MOVC A,@A+DPTR	A+DPTR 程序地址中的数到 A	93	1
MOVC A,@A+PC	A+PC 程序地址中的数到 A	83	1
MOVX A,@Ri	外部间接地址空间(8 位)数到 A	E2~E3	1

助记符	描述	指令码	字节
MOVX A,@DPTR	外部间接地址空间(16 位)数到 A	E0	1
MOVX @Ri,A	A 到外部间接地址空间(8 位)	F2~F3	1
MOVX @DPTR,A	A 到外部间接地址空间(16 位)	F0	1
PUSH direct	直接地址中的数压栈	C0	2
POP direct	堆栈中数出栈到直接地址	D0	2
XCH A,Rn	A 与寄存器数交换	C8~CF	1
XCH A,direct	A 与直接地址中的数交换	C5	2
XCH A,@Ri	A 与间接地址中的数交换	C6~C7	1
XCHD A,@Ri	A 与间接地址中的数的低半字节数交换	C6~C7	1

Table 42 程序控制类指令

助记符	描述	指令码	字节
ACALL addr11	绝对子程序调用, 地址范围 2K 字节	xxx10001	2
LCALL addr16	长调用, 地址范围 64K 字节	12	3
RET	子程序返回	22	1
RETI	中断子程序返回	32	1
AJMP addr11	绝对跳转, 地址范围 2K 字节	xxx00001	2
LJMP addr16	长跳转, 地址范围 64K 字节	02	3
SJMP rel	相对跳转, 地址范围-128~+127	80	2
JMP @A+DPTR	跳转到 A+DPTR 地址	73	1
JZ rel	如果 A=0, 跳转到相对地址	60	2
JNZ rel	如果 A! =0, 跳转到相对地址	70	2
JC rel	如果 C=1, 跳转到相对地址	40	2
JNC rel	如果 C=0, 跳转到相对地址	50	2
JB bit,rel	如果位地址数=1, 跳转到相对地址	20	3
JNB bit,rel	如果位地址数! =1, 跳转到相对地址	30	3
JBC bit, el	如果位地址数=1, 清除后, 跳转到相对地址	10	3
CJNE A,direct,rel	如果 A 与直接地址中数不等, 跳转到相对地址	B5	3
CJNE A,#data,rel	如果 A 与立即数不等, 跳转到相对地址	B4	3
CJNE Rn,#data,rel	如果寄存器数与立即数不等, 跳转到相对地址	B8~BF	3

助记符	描述	指令码	字节
CJNE @Ri,#data,rel	如果间接地址的数与立即数不等, 跳转到相对地址	B6~B7	3
DJNZ Rn,rel	寄存器数自减 1, 不等于 0, 跳转到相对地址	D8~DF	2
DJNZ direct,rel	直接地址中的数自减 1, 不等于 0, 跳转到相对地址	D5	3
NOP	空操作	00	1

Table 43 布尔类指令

助记符	描述	指令码	字节
CLR C	C 置 0	C3	1
CLR bit	位地址的数置 0	C2	2
SETB C	C 置 1	D3	1
SETB bit	位地址的数置 1	D2	2
CPL C	C 取反	B3	1
CPL bit	位地址的数取反	B2	2
ANL C,bit	位地址的数与 C 相与	82	2
ANL C,/bit	位地址的数取反与 C 相与	B0	2
ORL C,bit	位地址的数与 C 相或	72	2
ORL C,/bit	位地址的数取反与 C 相或	A0	2
MOV C,bit	位地址的数赋值到 C	A2	2
MOV bit,C	C 赋值到位地址	92	2

5 中断

5.1 概述

JMT51 包含一个扩展的中断系统，支持 13 个中断源，每个中断源有 4 个优先级。每个中断源在特殊功能寄存器（SFR）中有相应的中断标志。当一个中断源满足有效的中断条件时，相应的中断标志被置 1。

中断使能采用多级控制：

- 第一级控制为总控制，通过配置 IEN0.7 中断使能位 EA，对所有中断进行开启或屏蔽。
- 第二级控制为 13 个中断源分别控制，通过配置寄存器 IEN0、IEN1 及 IEN2 的相应比特位开启或屏蔽相应中断。
- 第三级控制为连接到第二级 13 个中断源的外设中断的控制，具体寄存器参见各外设模块寄存器说明。

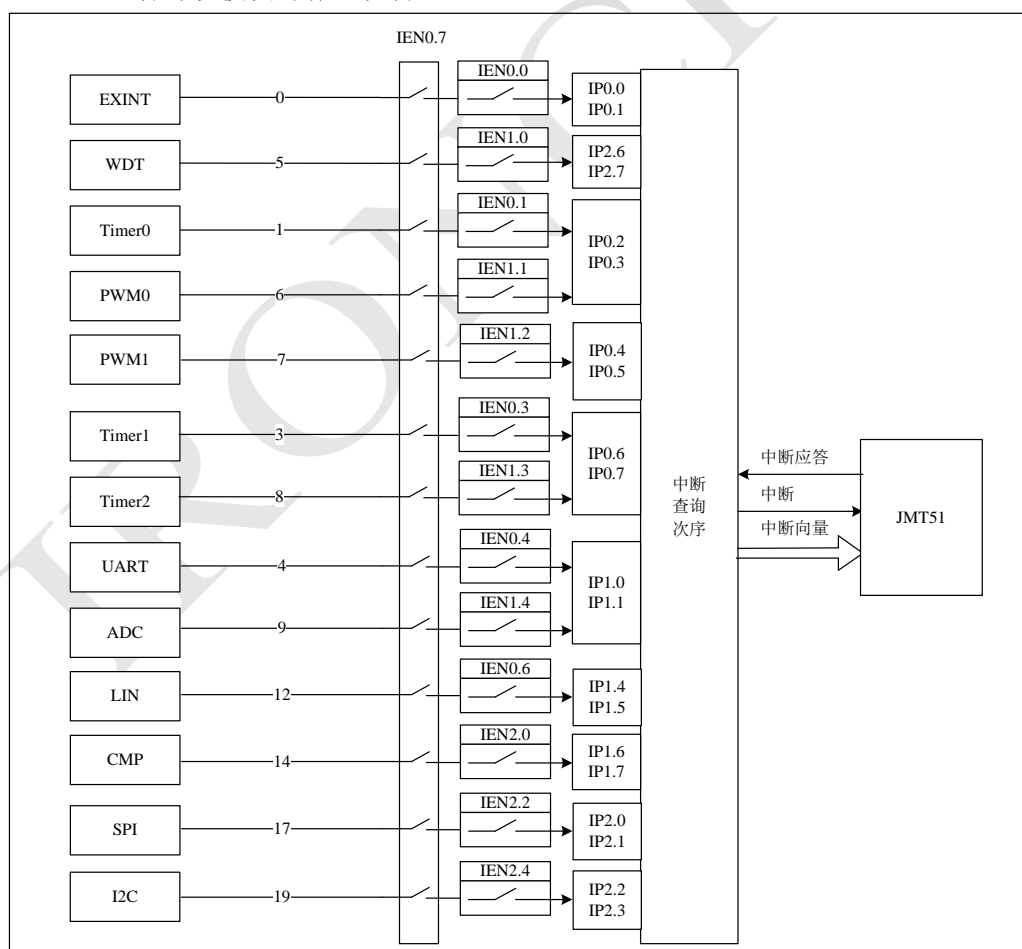


Figure 10 中断各级使能框图

Timer 中断标志位在 JMT51 进入中断服务程序后自动清除，其他中断标志硬件不会自

动清除，必须在中断返回之前软件配置寄存器清除，具体见 Table 44 中“断标志位清除说明”。

如果一个中断标志位在 JMT51 执行完中断返回指令（RETI）后仍然保持有效状态，则会立即产生一个新的中断请求，JMT51 执行完下一条指令后再次进入该中断。

中断分布如 Table 44 所示。

Table 44 中断分布情况

中断号	中断向量	中断源	默认中断优先级	中断标志位清除说明
0	0x03	EXINT 中断	最高 0	软件清除
1	0x0B	Timer 0 中断	2	硬件自动清除
3	0x1B	Timer 1 中断	5	硬件自动清除
4	0x23	UART 中断	7	软件清除
5	0x2B	WDT 中断	1	软件清除
6	0x33	PWM0 中断	3	软件清除
7	0x3B	PWM1 中断	4	软件清除
8	0x43	Timer2 中断	6	硬件自动清除
9	0x4B	ADC 中断	8	软件清除
12	0x63	LIN 中断	10	软件清除
14	0x73	CMP 中断	11	软件清除
17	0x8B	SPI 中断	12	软件清除
19	0x9B	I2C 中断	13	软件清除

5.2 外部管脚中断（EXINT）

JMT51 共有 1 个外部管脚中断：EXINT，由多个 GPIO 中断合并而成，每个 GPIO 中断又可独立配置为上升沿触发、下降沿触发或双沿触发。有关外部管脚中断 EXINT 的详细说明参见 17.7 章节描述。

5.3 中断优先级

中断优先级描述如下：

- 每个中断可以通过配置中断优先级寄存器 IP0、IP1、IP2 选择中断优先级。
- 同优先级的中断源同时申请中断，硬件会根据 Table 44 中各中断源的默认中断优先级顺序进行响应。
- 不同优先级的中断源同时申请中断时，JMT51 将响应较高优先级的中断申请。
- JMT51 在处理中断服务程序时，可被更高优先级的中断打断，但不能被同级或低优先级的中断打断。
- JMT51 在处理最高级中断服务程序时，不可被其他任何中断打断。

Table 45 中断优先级分组

组	中断源		描述
0	EXINT 中断		通过 IP0.0、IP0.1 设置优先级
1	Timer 0 中断	PWM0 中断	通过 IP0.2、IP0.3 设置优先级
2	PWM1 中断		通过 IP0.4、IP0.5 设置优先级
3	Timer 1 中断	Timer2 中断	通过 IP0.6、IP0.7 设置优先级
4	UART 中断	ADC 中断	通过 IP1.0、IP1.1 设置优先级
5	保留		通过 IP1.2、IP1.3 设置优先级
6	LIN 中断		通过 IP1.4、IP1.5 设置优先级
7	CMP 中断		通过 IP1.6、IP1.7 设置优先级
8	SPI 中断		通过 IP2.0、IP2.1 设置优先级
9	I2C 中断		通过 IP2.2、IP2.3 设置优先级
10	保留		通过 IP2.4、IP2.5 设置优先级
11	WDT 中断		通过 IP2.6、IP2.7 设置优先级

5.4 中断处理过程

每个系统时钟周期 JMT51 都将去检测是否存在中断申请, JMT51 捕获到中断标志位后, 将利用长转移指令 (LCALL) 调用该中断的中断服务程序, 下列情况下中断将会延迟处理:

- 同级或更高级的优先级中断在运行中。
- JMT51 正在执行的指令是 RETI 指令。在 RETI 之后, 至少再执行一条其他指令之后才会响应中断。
- JMT51 正在访问中断相关寄存器的指令, 在读写中断相关寄存器之后, 至少再执行一条其他指令之后才会响应中断。

若检测到某一中断, LCALL 指令会把程序计数器 (PC 值) 中的内容进行压栈, 将相应中断源的向量地址存入程序计数器 (PC 值), 中断服务程序从该中断向量指定地址开始运行, 直到 RETI 指令, RETI 指令通知 JMT51 中断服务程序结束, 将堆栈数据重载入程序计数器 (PC 值)。

5.5 中断响应时间

中断响应时间取决于中断发生时 JMT51 状态。

中断系统在每个 JMT51 时钟周期采样中断标志位, 最快响应时间为 5 个 JMT51 时钟周期: 1 个周期用于检测中断, 4 个周期完成对中断服务程序长调用 LCALL。如果中断标志有效时 JMT51 正在执行 RETI 指令, 则需要再执行 1 条指令才能进入中断服务程序。如果 RETI 后面跟 1 个多周期指令, 则响应时间会更长。

5.6 中断寄存器

中断模块共有 6 个寄存器，对应的 SFRPAGE 为所有 PAGE（公共寄存器），也即在所有 PAGE 配置中断寄存器均生效。具体列表如 Table 46 所示。

Table 46 中断模块寄存器（所有 PAGE）

地址	寄存器名	类型	复位值	功能描述
0xA8	IEN0	W/R	0x00	中断使能寄存器 0
0xB8	IEN1	W/R	0x00	中断使能寄存器 1
0xC0	IEN2	W/R	0x00	中断使能寄存器 2
0xB9	IP0	W/R	0x00	中断优先级寄存器 0
0xBA	IP1	W/R	0x00	中断优先级寄存器 1
0xBB	IP2	W/R	0xC0	中断优先级寄存器 2

5.6.1 中断使能寄存器 0(IEN0) [0xA8]

Table 47 中断使能寄存器 0(IEN0)

参数名	比特位	属性	复位值	描述
EXINT	0	W/R	0	EXINT 中断允许位： 0：禁止 EXINT 中断 1：允许 EXINT 中断
ET0	1	W/R	0	Timer0 中断允许位： 0：禁止 Timer0 中断 1：允许 Timer0 中断
Reserved	2	R	0	保留
ET1	3	W/R	0	Timer1 中断允许位： 0：禁止 Timer1 中断 1：允许 Timer1 中断
ES	4	W/R	0	UART 发送/接收中断允许位： 0：禁止 UART 发送/接收中断 1：允许 UART 发送/接收中断
Reserved	5	R	0	保留
LIN	6	W/R	0	LIN 中断允许位： 0：禁止 LIN 中断 1：允许 LIN 中断
EA	7	W/R	0	总中断允许位：

参数名	比特位	属性	复位值	描述
				0: 禁止所有中断 1: 允许所有中断

5.6.2 中断使能寄存器 1(IEN1) [0xB8]

Table 48 中断使能寄存器 1(IEN1)

参数名	比特位	属性	复位值	描述
WDTINT	0	W/R	0	WDT 中断允许位: 0: 禁止 WDT 中断 1: 允许 WDT 中断
PWM0INT	1	W/R	0	PWM0 中断允许位: 0: 禁止 PWM0 中断 1: 允许 PWM0 中断
PWM1INT	2	W/R	0	PWM1 中断允许位: 0: 禁止 PWM1 中断 1: 允许 PWM1 中断
ET2	3	W/R	0	Timer2 中断允许位: 0: 禁止 Timer2 中断 1: 允许 Timer2 中断
ADCINT	4	W/R	0	ADC 中断允许位: 0: 禁止 ADC 中断 1: 允许 ADC 中断
Reserved	7-5	R	0	保留

5.6.3 中断使能寄存器 2 (IEN2) [0xC0]

Table 49 中断使能寄存器 2(IEN2)

参数名	比特位	属性	复位值	描述
CMPINT	0	W/R	0	CMP 中断允许位: 0: 禁止 CMP 中断 1: 允许 CMP 中断
Reserved	1	R	0	保留
SPIINT	2	W/R	0	SPI 中断允许位: 0: 禁止 SPI 中断 1: 允许 SPI 中断

参数名	比特位	属性	复位值	描述
Reserved	3	R	0	保留
I2CINT	4	W/R	0	I2C 中断允许位： 0：禁止 I2C 中断 1：允许 I2C 中断
Reserved	7-5	R	0	保留

5.6.4 中断优先级寄存器 0(IP0) [0xB9]

Table 50 中断优先级寄存器 0(IP0)

参数名	比特位	属性	复位值	描述
IP0.1~IP0.0	1-0	W/R	0	EXINT 中断优先级控制： 00：第 0 级（最低） 01：第 1 级 10：第 2 级 11：第 3 级（最高）
IP0.3~IP0.2	3-2	W/R	0	Timer0 中断和 PWM0 中断优先级控制： 00：第 0 级（最低） 01：第 1 级 10：第 2 级 11：第 3 级（最高）
IP0.5~IP0.4	5-4	W/R	0	PWM1 中断优先级控制： 00：第 0 级（最低） 01：第 1 级 10：第 2 级 11：第 3 级（最高）
IP0.7~IP0.6	7-6	W/R	0	Timer1 中断和 Timer2 中断优先级控制： 00：第 0 级（最低） 01：第 1 级 10：第 2 级 11：第 3 级（最高）

5.6.5 中断优先级寄存器 1(IP1) [0xBA]

Table 51 中断优先级寄存器 1(IP1)

参数名	比特位	属性	复位值	描述
-----	-----	----	-----	----

参数名	比特位	属性	复位值	描述
IP1.1~IP1.0	1-0	W/R	0	UART 中断和 ADC 中断优先级控制： 00: 第 0 级（最低） 01: 第 1 级 10: 第 2 级 11: 第 3 级（最高）
IP1.3~IP1.2	3-2	W/R	0	保留
IP1.5~IP1.4	5-4	W/R	0	LIN 中断优先级控制： 00: 第 0 级（最低） 01: 第 1 级 10: 第 2 级 11: 第 3 级（最高）
IP1.7~IP1.6	7-6	W/R	0	CMP 中断优先级控制： 00: 第 0 级（最低） 01: 第 1 级 10: 第 2 级 11: 第 3 级（最高）

5.6.6 中断优先级寄存器 2(IP2) [0xBB]

Table 52 中断优先级寄存器 2(IP2)

参数名	比特位	属性	复位值	描述
IP2.1~IP2.0	1-0	W/R	0	SPI 中断优先级控制： 00: 第 0 级（最低） 01: 第 1 级 10: 第 2 级 11: 第 3 级（最高）
IP2.3~IP2.2	3-2	W/R	0	I2C 中断优先级控制： 00: 第 0 级（最低） 01: 第 1 级 10: 第 2 级 11: 第 3 级（最高）
IP2.5~IP2.4	5-4	W/R	0	保留
IP2.7~IP2.6	7-6	W/R	11	WDT 中断优先级控制： 00: 第 0 级（最低）

参数名	比特位	属性	复位值	描述
				01: 第 1 级 10: 第 2 级 11: 第 3 级 (最高)

6 存储器

6.1 JMT51 存储器

JMT51 和标准的 8051 类似，存储器由三部分组成：程序存储器、外部数据存储器以及内部数据存储器。其中，程序存储器和外部数据存储器共享相同的地址空间，通过使用不同的指令类型进行访问。

JMT51 的存储器组织如 Figure 11 所示。

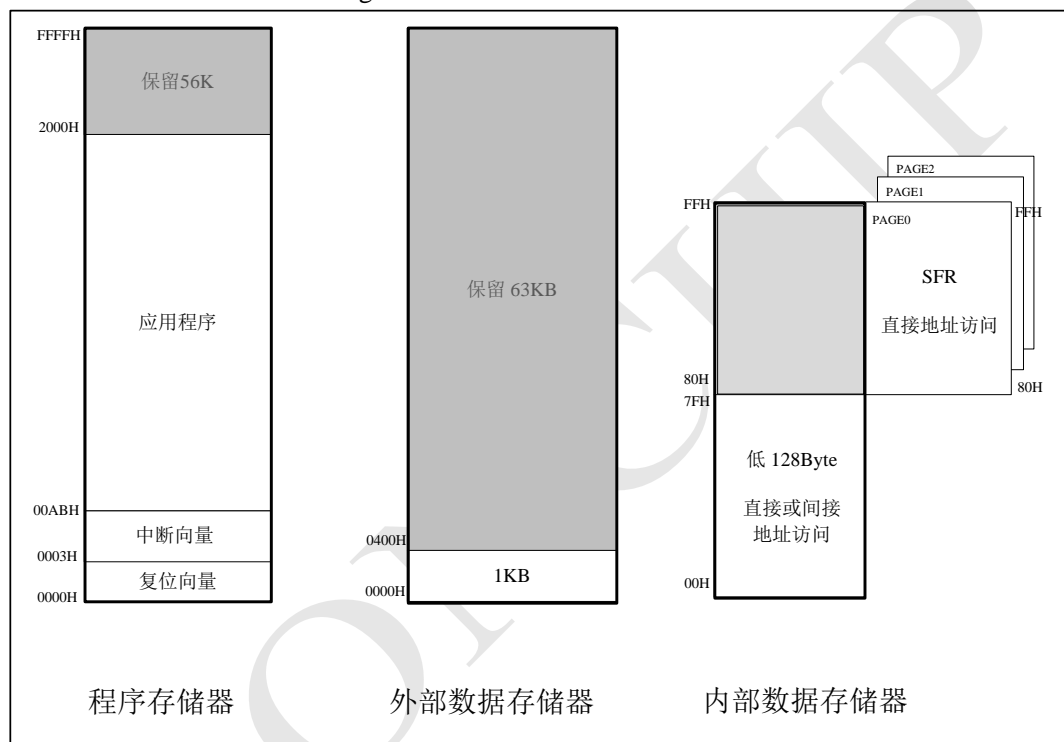


Figure 11 JMT51 存储器分布图

6.1.1 程序存储器

JMT51 有 8K 字节的程序存储空间，该程序存储器为可编程的 FLASH 存储器，地址范围为 0x0000~0x1fff，如 Figure 11 中程序存储器所示。

JMT51 内部含有 BOOT ROM，用于用户应用程序的烧录，该 BOOT ROM 对用户不可见。

FLASH 存储器一般用于存储 JMT51 运行的应用程序，该应用程序为通过 BOOT ROM 下载至 FLASH 中的。上电之后，JMT51 自动从 FLASH 的 0x0000 地址开始运行所下载的应用程序。

FLASH 存储器也可用于在线存储用户的非易失性数据，具体参见章节 7Flash 存储器。

6.1.2 外部数据存储器

JMT51 外部数据存储器，如 Figure 11 中外部数据存储器所示，大小为 1K 字节，地址空间为：0x0000~0x03ff；

配置寄存器 MOVXCON.PMW 位为 0，JMT51 即可使用 MOVX 指令对外部数据存储器进行读写操作，该存储器的读写有两种寻址方式：DPTR 地址、间接地址。

6.1.3 内部数据存储器

JMT51 内部数据存储器大小为 128 字节，地址空间为 0x00~0x7f，如 Figure 12 所示。
128 字节内部数据存储器可用于通用寄存器和临时存储器，可通过直接或间接寻址方式进行访问。
地址从 0x00 到 0x1f 为 4 个通用寄存器区，每个区有 8 个字节的寄存器。
从地址 0x20 到 0x2f 的 16 个字节，既可以字节寻址，又可作为 128 个位地址利用直接位寻址方式进行访问。

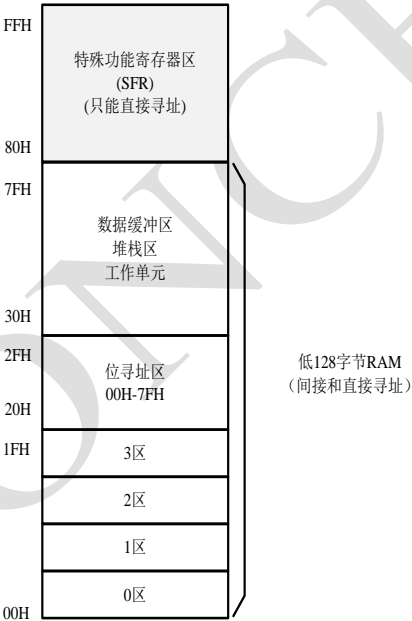


Figure 12 内部数据存储器分区

6.1.3.1 通用寄存器区

通用寄存器区(0x00~0x1f)，共 32 个单元分为四组，每组包含 8 个 8 位寄存器，均以 R0~R7 命名。通过状态寄存器 PSW 的 RS0 和 RS1 位决定使用哪一组寄存器，某一时刻只能选择其中的一组寄存器，其余的可以作为正常的数据存储器使用。间接寻址方式使用 R0 和 R1 作为间接寄存器。JMT51 复位后，默认选中第 0 组寄存器。

Table 53 通用寄存器地址

组	RS1	RS0	R0	R1	R2	R3	R4	R5	R6	R7
---	-----	-----	----	----	----	----	----	----	----	----

0	0	0	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
1	0	1	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
2	1	0	0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
3	1	1	0x20	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F

6.1.3.2 位寻址区

位寻址区(0x20~0x2F)，内部数据存储器的 0x20~0x2F 单元为位寻址区，即可作为一般单元用字节寻址，也可对它们的位进行寻址。位寻址共有 16 个字节，128 位，位地址为 0x00~0x7F。位地址分配如 Table 54 所示。

特殊功能寄存器（SFR）中，直接地址可被 8 整除的寄存器，也可以进行位寻址。

JMT51 能直接访问位地址对应的 BIT，执行如置 1、清 0、求反、转移、传送、逻辑等操作。

Table 54 位寻址地址

单元地址	MSB 位地址 LSB							
0x2F	0x7F	0x7E	0x7D	0x7C	0x7B	0x7A	0x79	0x78
0x2E	0x77	0x76	0x75	0x74	0x73	0x72	0x71	0x70
⋮	---							
0x2B	0x0F	0x0E	0x0D	0x0C	0x0B	0x0A	0x09	0x08
0x20	0x07	0x06	0x05	0x04	0x03	0x02	0x01	0x00

6.1.3.3 堆栈

程序的堆栈可以位于 128 字节数据存储器中的任何位置，JMT51 堆栈深度参见 4.1 描述。

堆栈区域用堆栈指针 SP 指定，SP 指向最后使用的位置。下一个压入堆栈的数据将被存放在 SP+1 位置，然后 SP 加 1。复位以后堆栈指针被初始化为 0x07，因此第一个被压入堆栈的数据将被存放在地址 0x08，0x08 地址的内容是通用寄存器组 1 的第一个寄存器 R0。如果使用不止一个通用寄存器组，SP 值应被初始化为 128 字节内部数据存储器中不用于数据存储的位置。

一般用户要设置 SP 初值，堆栈是向上生成的。

6.1.3.4 一般存储区

片内 RAM 128 字节中，通用寄存器占有 32 个字节，位寻址区占有 16 个字节，剩下的 80 个字节为可供用户使用的一般寄存器区，对这部分区域的使用不作任何规定和限制，堆栈一般开辟在这个区域。

6.1.4 特殊功能寄存器

从 0x80 到 0xFF 的直接寻址存储空间为特殊功能寄存器(SFR),通过配置这些寄存器,可以设置外设及 JMT51 的功能及特性。

任何时刻用直接寻址访问从 0x80 到 0xFF 的存储器空间将访问特殊功能寄存器。地址以 0x0 或 0x8 结尾的 SFR (例如 P0、TCON、SCON 等)既可以字节寻址也可以位寻址,其他所有 SFR 只能字节寻址。

SFR 空间中未使用的地址保留以后使用,访问这些会产生不确定结果,应避免访问。

6.1.4.1 特殊功能寄存器(SFR)分页

特殊功能寄存器(SFR)分页,使得从 0x80 到 0xFF 的 SFR 地址空间可以映射更多的寄存器。JMT18F003PLUS 芯片 SFR 分为 3 页(PAGE): PAGE0/PAGE1/PAGE2。通过配置寄存器 SFRPAGE 来选择所属的页。

读写特殊功能寄存器地址空间的寄存器步骤如下:

- 配置寄存器 SFRPAGE,选择合适的 PAGE 号;
- 通过直接寻址方式,使用 MOV 指令读/写特殊功能寄存器(SFR)。

6.1.4.2 中断和 SFR 分页

若 JMT51 检测某一个中断,寄存器 SFRPAGE 将自动的切换至该中断模块对应的 PAGE 号,该功能极大的降低了软件在中断服务程序中对 PAGE 进行切换的负担。中断服务程序响应完成之后,退出中断,寄存器 SFRPAGE 又将自动的切换至进入中断服务程序之前的 PAGE 号。该功能通过一个 5 级 SFR PAGE 堆栈完成,该堆栈不占用 JMT51 内部数据存储器。

5 级 SFR PAGE 堆栈,最上面一级的堆栈为当前使用的 PAGE,也即寄存器 SFRPAGE,往下依次为 PGSTACK0、PGSTACK1、PGSTACK2、PGSTACK3,如 Figure 13 所示:

- 当检测到某一中断,寄存器 SFRPAGE 将自动压栈至 PGSTACK0,PGSTACK0 将自动压栈至 PGSTACK1,PGSTACK1 将自动压栈至 PGSTACK2,PGSTACK2 将自动压栈至 PGSTACK3,然后硬件自动将该中断所属模块对应的 PAGE 号加载至寄存器 SFRPAGE。
- 当退出某一中断时,之前存储至寄存器 PGSTACK0 的 PAGE 号将自动加载至寄存器 SFRPAGE,PGSTACK1 将自动出栈至 PGSTACK0,PGSTACK2 将自动出栈至 PGSTACK1,PGSTACK3 将自动出栈至 PGSTACK2,PGSTACK3 将自动赋值 0x00。
- 在中断服务程序中,如果需要修改当前 PAGE 号,用户可配置寄存器 SFRPAGE 进行修改。配置寄存器 SFRPAGE 不会发生压栈/出栈的动作,仅会更改当前使用的 PAGE 号。
- 在中断服务程序中,如果需要,用户可通过配置栈寄存器 PGSTACK0、PGSTACK1、PGSTACK2 和 PGSTACK3,修改中断返回之后的 PAGE。配置栈寄存器不会发生

压栈/出栈的动作，仅当发生中断时，才会发生压栈/出栈的动作。

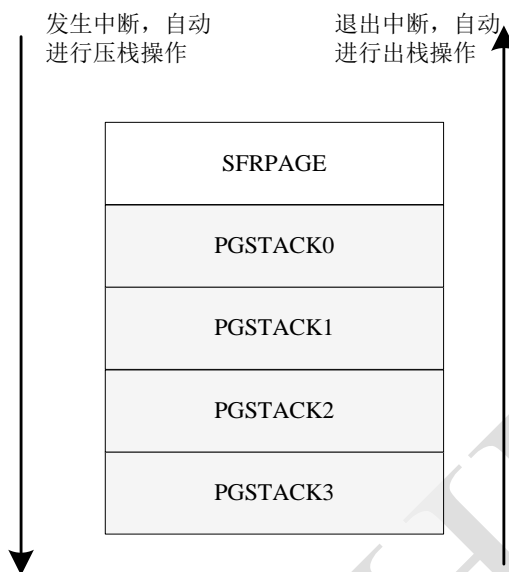


Figure 13 SFR PAGE 堆栈

通过配置寄存器 SFRPGEN，可以对硬件基于发生中断自动压栈，退出中断自动出栈的动作进行开关，默认为使能该功能。

Table 55 列出了 JMT18F003PLUS 各模块对应的 PAGE 号：

Table 55 各模块对应的 PAGE 号

PAGE0	PAGE1	PAGE2
Timer0/1	PWM0	GPIO
Timer2	PWM1	PAGE 堆栈
UART		CRM
SPI		
I2C		
PMU		
WDT		
CMP		
ADC		
FLASH		

Table 57、Table 58 和 Table 59 分别列出了 PAGE0、PAGE1 和 PAGE2 地址对应的寄存器名，从中可以看出，部分寄存器在所有 PAGE 中均可以访问，称之为公共寄存器，公共寄存器所包括的寄存器见 Table 56 所示。

Table 56 特殊功能寄存器（公共寄存器）

公共寄存器								
F8H	IP2							
F0H	B							
E8H	IP1							
E0H	ACC							
D8H	IP0							
D0H	PSW							
C8H	P4							
C0H	IEN2							
B8H	IEN1							
B0H	P3							
A8H	IEN0							
A0H	P2							
98H								
90H	P1							
88H								
80H	P0	SP	DPL	DPH	SFRPAGE	SFRPGEN	MOVXCON	

Table 57 特殊功能寄存器(PAGE0)

PAGE0								
F8H	IP2	FLSC	FLSK					
F0H	B							
E8H	IP1	I2CDUTYH	I2CHOLD	I2CWBUF	I2CRBUF	I2CSTS	I2CISC	I2CIEN
E0H	ACC	I2CCON	I2CSADDRL	I2CSADDRH	I2COADDRL	I2COADDRH	I2CDIV	I2CDUTYL
D8H	IP0	WDTCKDIV	WDTINT	WDTINTF	WDTLD	WDTST		
D0H	PSW	T2CON	RL2	RH2	TL2	TH2	T2PSC	
C8H	P4	SPICON	SPIIE	SPIF	SPIBR	SPIBUF	SPISPC	SPISPS
C0H	IEN2	OCPRSEL	ODPRSEL	BKPRSEL				
B8H	IEN1	CMPEN	CMPFT	CMPVDD	CMPREF	CMPINT	CMPFLAG	
B0H	P3							
A8H	IEN0							
A0H	P2	PMUCTR						

PAGE0								
98H	SCON	SBUF	SRELL	SRELH	SIRCON	LINCON		
90H	P1	ADCCON0	ADCCON1	ADCPGAC	ADCETC	ADCPLY	ADCPL	ADCDH
88H	TCON	TMOD	TL0	TL1	TH0	TH1	TPSC	
80H	P0	SP	DPL	DPH	SFRPAGE	SFRPGEN	MOVXCON	

Table 58 特殊功能寄存器(PAGE1)

PAGE1								
F8H	IP2	TIM0_PHASEH	TIM0_CNTEN					
F0H	B	TIM0_EGR	TIM0_PHACON	TIM0_ISR1	TIM0_ISR2	TIM0_CNTL	TIM0_CNTH	TIM0_PHASEL
E8H	IP1	TIM0_CC2R1L	TIM0_CC2R1H	TIM0_CC3R1L	TIM0_CC3R1H	TIM0_CC4R1L	TIM0_CC4R1H	TIM0_BRKC
E0H	ACC	TIM0_ARRL	TIM0_ARRH	TIM0_PSCL	TIM0_PSCH	TIM0_RCR	TIM0_CC1R1L	TIM0_CC1R1H
D8H	IP0	TIM0_CC1MR	TIM0_CC2MR	TIM0_CC3MR	TIM0_CC4MR	TIM0_CCENR	TIM0_CCPS	TIM0_DTC
D0H	PSW	TIM0_CONR1	TIM0_CONR2	TIM0_CONR3	TIM0_CONR4	TIM0_TGICR1	TIM0_TGICR2	TIM0_IER
C8H	P4		TIM1_CNTEN					
C0H	IEN2	TIM1_EGR		TIM1_ISR1	TIM1_ISR2	TIM1_CNTL	TIM1_CNTH	
B8H	IEN1	TIM1_CC2R1L	TIM1_CC2R1H	TIM1_CC3R1L	TIM1_CC3R1H			TIM1_BRKC
B0H	P3	TIM1_ARRL	TIM1_ARRH	TIM1_PSCL	TIM1_PSCH	TIM1_RCR	TIM1_CC1R1L	TIM1_CC1R1H
A8H	IEN0	TIM1_CC1MR	TIM1_CC2MR	TIM1_CC3MR		TIM1_CCENR	TIM1_CCPS	TIM1_DTC
A0H	P2	TIM1_CONR1	TIM1_CONR2	TIM1_CONR3	TIM1_CONR4	TIM1_TGICR1		TIM1_IER
98H								
90H	P1							
88H								
80H	P0	SP	DPL	DPH	SFRPAGE	SFRPGEN	MOVXCON	

Table 59 特殊功能寄存器(PAGE2)

PAGE2								
F8H	IP2	PGSTACK0	PGSTACK1	PGSTACK2	PGSTACK3			
F0H	B	CMP_CLK_DIV	GPIO_CLK_DIV					
E8H	IP1	CLK_EN0	CLK_EN1			PWM_CLK_DIV		
E0H	ACC	CLK_READY	OSC22_CTRL	OSC_STABL	OSC_STABH	FLTCKSEL	EXTCLK_DIVL	EXTCLK_DIVH
D8H	IP0	BANDGAP_CTRL	SYS_CLK_SEL	ADC_CLK_DIV	JTAGEN	TRIMKEY	BUZZER_CTRL0	BUZZER_CTRL1
D0H	PSW	TRMVAL32K	CRMCTRL	ADCTRIM	TRIM_CNT_72ML	TRIM_CNT_72MH	OSCPD	SYS_CLK_DIV
C8H	P4	LDO_CTRL	INSTRST	SWRST0	SWRST1	GLOBALRST	RSTFLAG	TRMVAL72M

PAGE2								
C0H	IEN2							
B8H	IEN1	PC0CTRL	PB5CTRL	PB6CTRL				
B0H	P3							
A8H	IEN0							
A0H	P2	INT0SEL0	INT0SEL1	INTPE	INTNE	INTEN	INTCON	
98H		PA7CTRL	PB7CTRL	PADSRRDR	AIOEN0	AIOEN1	FPSC	
90H	P1	PB0CTRL	PB1CTRL	PB2CTRL	PB3CTRL	PB4CTRL		
88H		PA0CTRL	PA1CTRL	PA2CTRL	PA3CTRL	PA4CTRL	PA5CTRL	PA6CTRL
80H	P0	SP	DPL	DPH	SFRPAGE	SFRPGEN	MOVXCON	

6.2 JMT51 核寄存器

JMT51 核有 7 个寄存器，对应的 SFRPAGE 为所有 PAGE（公共寄存器），也即在所有 PAGE 配置 JMT51 核寄存器均生效。具体说明如下。

Table 60 JMT51 核寄存器（所有 PAGE）

地址	寄存器名	属性	复位值	功能描述
0xE0	ACC	W/R	0x00	累加器 ACC
0xF0	B	W/R	0x00	B 寄存器
0x81	SP	W/R	0x07	堆栈指针(SP)
0xD0	PSW	W/R	0x00	程序状态字(PSW)寄存器
0x82	DPL	W/R	0x00	数据指针低字节(DPL)
0x83	DPH	W/R	0x00	数据指针高字节(DPH)
0x86	MOVXCON	W/R	0x01	MOVX 指令设置寄存器(MOVXCON)

6.2.1 ACC 寄存器(A) [0xE0]

Table 61 ACC 寄存器(A)

参数名	比特位	属性	复位值	描述
ACC	7-0	W/R	0	ACC 可以直接寻址，也可以位寻址，位地址范围 0xe0~0xe7

6.2.2 B 寄存器(B) [0xF0]

Table 62 B 寄存器(B)

参数名	比特位	属性	复位值	描述
B	7-0	W/R	0	B 可以直接寻址，也可以位寻址，位地址范围

参数名	比特位	属性	复位值	描述
				0xf0~0xf7; 乘法运算中存放乘数, 运算结束后, 存放计算结果高 8 位; 除法运算中存放除数, 运算结束后, 存放余数。

6.2.3 堆栈指针(SP) [0x81]

Table 63 堆栈指针寄存器(SP)

参数名	比特位	属性	复位值	描述
SP	7-0	W/R	0111	在执行 PUSH、各种子程序调用、中断响应指令时, SP 先加 1, 再将数据压栈; 执行 POP、RET、RETI 指令时, 数据退出堆栈后 SP 再减 1。 堆栈栈顶可以是 128 字节内部数据存储器任意地址。

6.2.4 程序状态字寄存器(PSW) [0xD0]

程序状态字寄存器包含了程序状态信息。

PSW 可以直接寻址, 可以位寻址, 位地址范围 (0xd0~0xd7)。

Table 64 程序状态字寄存器(PSW)

参数名	比特位	属性	复位值	描述
P	0	W/R	0	奇偶标志位: 0: 累加器 A 中 1 的位数为偶数 1: 累加器 A 中 1 的位数为奇数
F1	1	W/R	0	用户标志位: 受软件控制的通用标志位, 用户可以自己定义。
OV	2	W/R	0	溢出标志位: 0: 没有溢出发生 1: 下列情况下溢出: <ul style="list-style-type: none"> ● ADD、ADDC 或 SUBB 指令引起符号位变化溢出 ● MUL 指令结果大于 255 ● DIV 指令除数为 0

参数名	比特位	属性	复位值	描述
RS1~RS0	4-3	W/R	0	R0~R7 寄存器区选择: RS1 RS0 0 0 : 组 0 0 1 : 组 1 1 0 : 组 2 1 1 : 组 3
F0	5	W/R	0	用户标志位: 受软件控制的通用标志位, 用户可以自己定义。
AC	6	W/R	0	辅助进位标志位: 0: 算术运算中, 没有低字节向高字节进位或借位 1: 算术运算中, 有低字节向高字节进位或借位
C	7	W/R	0	进位标志位: 0: 算术运算中, 没有进位或借位发生 1: 算术运算中, 有进位或借位发生

6.2.5 数据指针低字节(DPL) [0x82]

Table 65 数据指针低字节(DPL)

参数名	比特位	属性	复位值	描述
DPL	7-0	W/R	0	数据指针低字节, 为 16 位数据指针 DPTR 的低字节。 DPTR 用于访问间接寻址的外部数据存储器或 FLASH 存储器。

6.2.6 数据指针高字节(DPH) [0x83]

Table 66 数据指针高字节(DPH)

参数名	比特位	属性	复位值	描述
DPH	7-0	W/R	0	数据指针高字节, 为 16 位数据指针 DPTR 的高字节。 DPTR 用于访问间接寻址的外部数据存储器或 FLASH 存储器。

6.2.7 MOVX 指令设置寄存器(MOVXCON) [0x86]

Table 67 MOVX 指令设置寄存器(MOVXCON)

参数名	比特位	属性	复位值	描述
XMCON	2-0	W/R	001	MOVX 存储器访问等待周期： 000: 读 1 个周期，写 1 个周期 001: 读 2 个周期，写 1 个周期 010: 读 3 个周期，写 2 个周期 011: 读 4 个周期，写 3 个周期 100: 读 5 个周期，写 4 个周期 101: 读 6 个周期，写 5 个周期 110: 读 7 个周期，写 6 个周期 111: 读 8 个周期，写 7 个周期
Reversed	6-3	R	0	保留
PMW	7	W/R	0	MOVX 指令访问空间选择： 0: MOVX 指令访问空间为外部数据存储器 1: MOVX 指令访问空间为 FLASH 存储器

其他特殊功能寄存器（SFR）详见各个模块的说明。

6.3 PAGE 堆栈寄存器

PAGE 堆栈寄存器分为两部分：

- SFRPAGE 和 SFRPGEN。这两个寄存器对应的 SFRPAGE 为所有 PAGE（公共寄存器），也即在所有 PAGE 配置这两个寄存器均生效。
- 堆栈寄存器 PGSTACK0、PGSTACK1、PGSTACK2 和 PGSTACK3。对应的 SFRPAGE 为 2。

Table 68 PAGE 堆栈寄存器

地址	寄存器名	属性	复位值	功能描述	PAGE 号
0x84	SFRPAGE	W/R	0x00	SFR PAGE 寄存器，指示当前特殊功能寄存器所属的 PAGE 号	所有 PAGE
0x85	SFRPGEN	W/R	0x01	PAGE 堆栈使能	所有 PAGE
0xF9	PGSTACK0	W/R	0x00	PAGE 堆栈寄存器 0	PAGE2
0xFA	PGSTACK1	W/R	0x00	PAGE 堆栈寄存器 1	PAGE2
0xFB	PGSTACK2	W/R	0x00	PAGE 堆栈寄存器 2	PAGE2
0xFC	PGSTACK3	W/R	0x00	PAGE 堆栈寄存器 3	PAGE2

6.3.1 SFR PAGE 寄存器(SFRPAGE) [0x84]

Table 69 SFR PAGE 寄存器(SFRPAGE)

参数名	比特位	属性	复位值	描述
SFRPAGE	2-0	W/R	0	SFR PAGE 寄存器, 指示当前特殊功能寄存器所属的 PAGE 号: 000: PAGE0 001: PAGE1 010: PAGE2 其它: PAGE0
Reserved	7-3	R	0	保留

6.3.2 PAGE 堆栈使能寄存器(SFRPGEN) [0x85]

Table 70 PAGE 堆栈使能寄存器(SFRPGEN)

参数名	比特位	属性	复位值	描述
SFRPGEN	0	W/R	1	PAGE 堆栈使能: 0: 不使能 1: 使能
Reserved	7-1	R	0	保留

6.3.3 PAGE 堆栈寄存器 0(PGSTACK0) [0xF9]

Table 71 PAGE 堆栈寄存器 0(PGSTACK0)

参数名	比特位	属性	复位值	描述
PGSTACK0	2-0	W/R	0	PAGE 堆栈寄存器 0
Reserved	7-3	R	0	保留

6.3.4 PAGE 堆栈寄存器 1(PGSTACK1) [0xFA]

Table 72 PAGE 堆栈寄存器 1(PGSTACK1)

参数名	比特位	属性	复位值	描述
PGSTACK1	2-0	W/R	0	PAGE 堆栈寄存器 1
Reserved	7-3	R	0	保留

6.3.5 PAGE 堆栈寄存器 2(PGSTACK2) [0xFB]

Table 73 PAGE 堆栈寄存器 2(PGSTACK2)

参数名	比特位	属性	复位值	描述
PGSTACK2	2-0	W/R	0	PAGE 堆栈寄存器 2

参数名	比特位	属性	复位值	描述
Reserved	7-3	R	0	保留

6.3.6 PAGE 堆栈寄存器 3(PGSTACK3) [0xFC]

Table 74 PAGE 堆栈寄存器 3(PGSTACK3)

参数名	比特位	属性	复位值	描述
PGSTACK3	2-0	W/R	0	PAGE 堆栈寄存器 3
Reserved	7-3	R	0	保留

7 Flash 存储器

7.1 概述

JMT18F003PLUS 采用 FLASH 作为程序存储空间，FLASH 程序存储空间（MAIN）大小为 8K Bytes，并包括 2 个非易失性扇区（NVR0 和 NVR1，每个 512 Bytes）：

- NVR0 用户可用，0x2000~0x203f 存放出厂前的 trimming 数据（共 64 个 Byte），其余的 448 个 Byte 可用作数据 FLASH 供用户使用。注意：由于 NVR0 是整个扇区擦除，用户写入新的数据时，先读出其它地址已写入的数据，NVR0 扇区擦除后，读出的数据再和新数据一起写入。
- NVR1 用于 FLASH 的系统参数，用户不能更改。

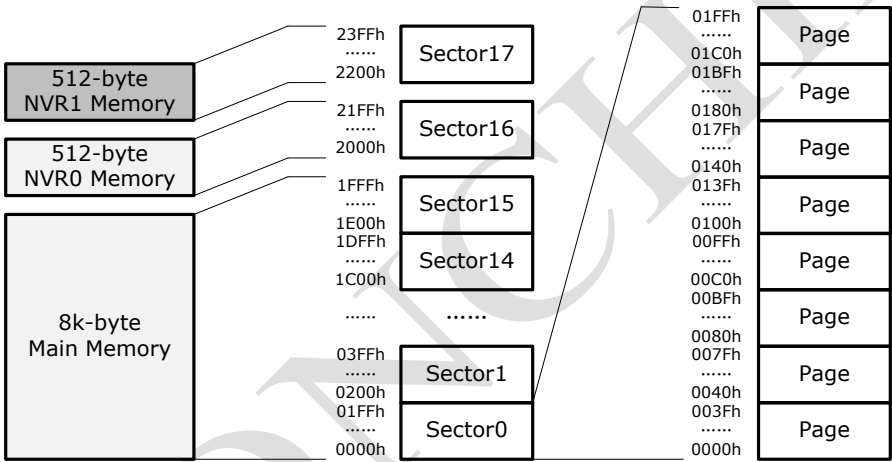


Figure 14 FLASH Memory

FLASH 如 Table 75 所示。

Table 75 FLASH 特性

特性参数	大小
扇区（Sector）大小	512Bytes
页（PAGE）大小	64Bytes
数据保持时间	最少 10 年
擦除次数	最少 100,000 次

7.2 FLASH 操作

FLASH 在编程之前要先进行擦除，支持对 FLASH 的以下操作：

- 支持 MAIN 和 NVR0 读取
- 支持 MAIN 和 NVR0 扇区（SECTOR）擦除
- 支持 MAIN 和 NVR0 编程

7.2.1 FLASH 数据读取

读取 Flash 中的数据，在汇编程序中可以使用以下两种方式：

- “MOVX A,@DPTR”方式如下：
 - 1) 寄存器 MOVXCON.PMW 置 1，选择 MOVX 指令对 FLASH 的操作
 - 2) 寄存器 FLSC.OM 设置为“000”（默认），选择对 FLASH 读取操作
 - 3) 将要读取的 FLASH 地址写入 DPTR，使用“MOVX A,@DPTR”指令，将数据存入累加器 A
- “MOVC A,@A+DPTR”方式如下：
 - 1) FLSC.OM 设置为“000”（默认），选择对 FLASH 读取操作
 - 2) 将要读取的 FLASH 地址写入 DPTR，清零累加器 A，使用“MOVC A,@ A+DPTR”指令，将数据存入累加器 A

读取 Flash 中的数据，C 语言实现例程如下：

- “MOVX A,@DPTR”方式：

```
MOVXCON |= 0x80;
buffer = *((unsigned char xdata *) address);
MOVXCON &= 0x7F;
```
- “MOVC A,@ A+DPTR”方式：

```
buffer = *((unsigned char code *) address);
```

7.2.2 FLASH 扇区擦除

操作流程如下：

- 1) 配置寄存器 WDTST 为 0，关闭 WDT
 - 2) 寄存器 MOVXCON.PMW 置 1，选择“MOVX @DPTR,A”对 FLASH 进行操作
 - 3) 向 FLSK 先写入 0xA5，再写入 0xF1，解除对 FLASH 操作模式的锁定
 - 4) 寄存器 FLSC.OM 设置为“011”，选择对 FLASH 扇区擦除操作
 - 5) 将要擦除的 FLASH 扇区首地址写入 DPTR，向累加器 A 写入任意数据，使用“MOVX @DPTR,A”指令，启动对 FLASH 该扇区的擦除操作
 - 6) FLASH 擦除过程中，MCU 处于等待状态，擦除完成后，MCU 继续取指工作
- FLASH 擦除时间较长，擦除前需要关闭 WDT，防止在擦除过程中产生 WDT 复位。

7.2.3 FLASH 编程

操作流程如下：

- 1) 配置寄存器 WDTST 为 0，关闭 WDT
- 2) 寄存器 MOVXCON.PMW 置 1，选择“MOVX @DPTR,A”对 FLASH 进行操作
- 3) 向 FLSK 先写入 0xA5，再写入 0xF1，解除对 FLASH 操作模式的锁定

- 4) 寄存器 FLSC.OM 设置为“001”，选择对 FLASH 的编程操作
- 5) 将要编程的 FLASH 地址写入 DPTR，向累加器 A 写入需要编程的数据，使用“MOVX @DPTR,A”指令，启动对 FLASH 的按字节编程操作
- 6) FLASH 编程过程中，MCU 处于等待状态，编程完成后，MCU 继续取指工作
- 7) 重复操作 3) 到 6)，直到完成所有字节的编程

FLASH 编程时间较长，编程前需要关闭 WDT，防止在编程过程中产生 WDT 复位。

7.3 FLASH 寄存器

该模块共有 2 个寄存器，对应的 SFRPAGE 为 0，具体列表如 Table 76 所示。

Table 76 FLASH 寄存器（SFRPAGE=0x00）

地址	寄存器名	类型	复位值	功能描述
0xF9	FLSC	W/R	0x00	FLASH 控制寄存器
0xFA	FLSK	W/R	0x00	FLASH 关键字寄存器

7.3.1 FLASH 控制寄存器(FLSC) [0xF9]

Table 77 FLASH 控制寄存器(FLSC)

参数名	比特位	属性	复位值	描述
OM	2-0	W/R	0	FLASH 操作模式控制位： 000：读操作（默认） 001：编程 011：扇区擦除 其他：保留
Reserved	7-3	R	0	保留

7.3.2 FLASH 关键字寄存器(FLSK) [0xFA]

Table 78 FLASH 关键字寄存器(FLSK)

参数名	比特位	属性	复位值	描述
FLSK	7-0	W/R	0	只有向 FLSK 中顺序写入 0xA5 和 0xF1 后，FLSC.OM 比特才能更新；OM 写入 0 时，没有此限制； FLSK.0 回读值指示 0xA5 是否写入，若继续写入 0xF1，该位回读值清零； FLSK.1 回读值指示 0xA5 和 0xF1 已经连续写入，当对 FLSC.OM 写入更新后，FLSK.0 和 FLSK.1 自动清零。

8 看门狗定时器（WDT）

8.1 概述

看门狗定时器（WDT）可用来检测 and 解决由软件错误引起的故障。

看门狗定时器（WDT）主要功能如下：

- 16 位计数器；
- 定时时钟可由 32KHz OSC 及其分频得到（1-8 分频可配置）；
- 可产生中断和复位；

8.2 功能描述

芯片正常工作时，每隔一段时间需要配置 WDTLD 寄存器为 1 对 WDT 进行喂狗操作，使得 WDT 计数器重新初始化并开始新一轮计数。若不及时喂狗，计数溢出后则会产生复位信号进行全芯片复位。

WDT 有两种工作模式：

- 先中断后复位模式：配置 WDTCKDIV.WDTINTS 为 0 时 WDT 工作于先中断后复位模式。该模式下当 WDT 从 0xFFFF 递减至 16 位比较值 WDTINT 时，产生中断，软件需要根据该中断配置 WDTLD 为 1 进行喂狗操作，若不及时进行喂狗操作，当计数器递减至 0 时，产生全芯片复位。
- 复位模式：配置 WDTCKDIV.WDTINTS 为 1 时 WDT 工作于复位模式。该模式下 WDT 从 0xFFFF 递减至 16 位比较值 WDTINT 时，产生全芯片复位。

8.3 寄存器

WDT 模块共有 5 个寄存器，对应的 SFRPAGE 为 0，具体说明如 Table 79 所示。

Table 79 WDT 寄存器（SFRPAGE=0x00）

地址	寄存器名	属性	复位值	功能描述
0xD9	WDTCKDIV	W/R	0x00	看门狗分频系数寄存器
0xDA	WDTINT	W/R	0x00	看门狗比较值寄存器
0xDB	WDTINTF	W/R	0x00	看门狗中断标志寄存器
0xDC	WDTLD	W/R	0x00	看门狗喂狗寄存器
0xDD	WDTST	W/R	0x01	看门狗启停寄存器

8.3.1 看门狗分频系数寄存器(WDTCKDIV) [0xD9]

Table 80 看门狗分频系数寄存器(WDTCKDIV)

参数名	比特位	属性	复位值	描述
WDTINTS	0	W/R	0	计数达到 WDTINT 时产生复位或中断： 0：产生中断 1：产生复位 其中 WDTINT=[WDTINTH:0xFF]
CKDIV	3-1	W/R	0	工作时钟分频系数： 000：1 分频 001：2 分频 010：3 分频 011：4 分频 100：5 分频 101：6 分频 110：7 分频 111：8 分频
Reserved	7-4	R	0	保留

8.3.2 看门狗比较值寄存器(WDTINT) [0xDA]

Table 81 看门狗比较值寄存器(WDTINT)

参数名	比特位	属性	复位值	描述
WDTINTH	7-0	W/R	0	比较值寄存器高 8 位，低 8 位固定为 0xFF，可配置最大值为 0xFE，即当该寄存器配置为以下值时对应的 16 位比较值 WDTINT 如下所示： 0xFE：真实比较值为 0xFEFF 0xFD：真实比较值为 0xFDFE 0x01：真实比较值为 0x01FF 0x00：真实比较值为 0x00FF 该寄存器配置后需要配置 WDTLD 为 1 才会加载生效

8.3.3 看门狗中断标志寄存器(WDTINTF) [0xDB]

Table 82 看门狗中断标志寄存器(WDTINTF)

参数名	比特位	属性	复位值	描述
WDTINTFLG	0	W/R	0	看门狗中断标志位： <ul style="list-style-type: none"> 当看门狗有中断申请时，该位置为 1 软件配置该位为 1 可产生软中断，在中断处理程序中需要软件写 0 来清除
Reserved	7-1	R	0	保留

8.3.4 看门狗喂狗寄存器(WDTLD) [0xDC]

Table 83 看门狗喂狗寄存器(WDTLD)

参数名	比特位	属性	复位值	描述
WDTLD	0	W/R	0	WDT 喂狗信号： 1：进行喂狗操作，重加载比较值，重加载时钟分频计数值，看门狗初始计数从 0xFFFF 重新开始递减计数，硬件自清 0：不进行喂狗
Reserved	7-1	R	0	保留

8.3.5 看门狗启停寄存器(WDTST) [0xDD]

Table 84 看门狗启停寄存器(WDTST)

参数名	比特位	属性	复位值	描述
WDTST	1-0	W/R	01	WDT 启动关闭信号： 01/10：启动看门狗 其他：关闭看门狗
Reserved	7-2	R	0	保留

9 定时器/计数器 (TIMER)

9.1 概述

JMT18F003PLUS 芯片有 3 个 16 位定时器/计数器: Timer0, Timer1 和 Timer2, 都有两种模式:

- 定时器模式: 以 timer_clk 时钟周期为基准进行定时。该模式下, 通过配置 TPSC 或 T2PSC 寄存器, 可对每个 Timer 的时钟进行预分频 (1/8/64/256 分频)。
- 计数器模式: 对外部输入管脚 T0、T1 或 T2 的下降沿进行计数。

Timer0 有 4 种工作模式:

- 模式 0: 13 位定时器/计数器
- 模式 1: 16 位定时器/计数器
- 模式 2: 8 位自动重装定时器/计数器
- 模式 3: 两个 8 位定时器/计数器

Timer1 有 3 种工作模式:

- 模式 0: 13 位定时器/计数器
- 模式 1: 16 位定时器/计数器
- 模式 2: 8 位自动重装定时器/计数器

Timer2 有 4 种工作模式:

- 模式 0: 13 位定时器/计数器
- 模式 1: 16 位定时器/计数器
- 模式 2: 16 位自动重装定时器/计数器
- 模式 3: 1 个 8 位定时器/计数器

9.2 TIMER 管脚配置

TIMER 的管脚配置可参见 Table 1 引脚说明表格, 管脚配置说明见 GPIO 寄存器说明。

9.3 Timer0 工作模式

配置寄存器 TMOD 中的 M1(TM0D.1)、M0(TM0D.0), 设置 Timer0 的工作模式。

9.3.1 模式 0(13 位定时器/计数器)

配置寄存器 TMOD 的标志位 $TMOD[1:0]^2=00$, 使得 Timer0 工作在模式 0。配置寄存器 TMOD[2] (C/T 位) 可选择为定时器模式或计数器模式。

Timer0 被分为两个 8 位寄存器, 低字节 TL0 和高字节 TH0, TL0 的低 5 位和 TH0 组成

² TMOD[1:0]代表寄存器 TMOD 的低两位

一个 13 位的计数器，当 Timer0 溢出时寄存器位 TF0 置位，同时产生 Timer0 溢出中断。程序进入中断后，该位被自动清零。

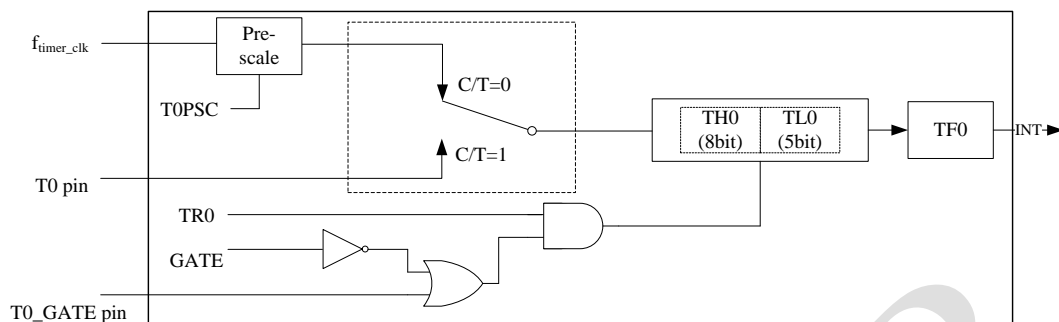


Figure 15 Timer0 模式 0 实现框图

当 C/T 为 0，Timer0 用作定时器时， $X=\{TH0[7:0], TL0[4:0]\}$ ，定时时间计算公式为：

$$T_d = (2^{13} - X) \times T_{timer_clk} \times \text{预分频系数}$$

9.3.2 模式 1(16 位定时器/计数器)

配置寄存器 TMOD 的标志位 TMOD[1:0]=01，使得 Timer0 工作在模式 1。配置寄存器 TMOD[2] (C/T 位) 可选择为定时器模式或计数器模式。

模式 1 为 TL0 和 TH0 组成一个 16 位的计数器，当 Timer0 溢出时寄存器位 TF0 置位，同时产生 Timer0 溢出中断。程序进入中断后，该位被自动清零。

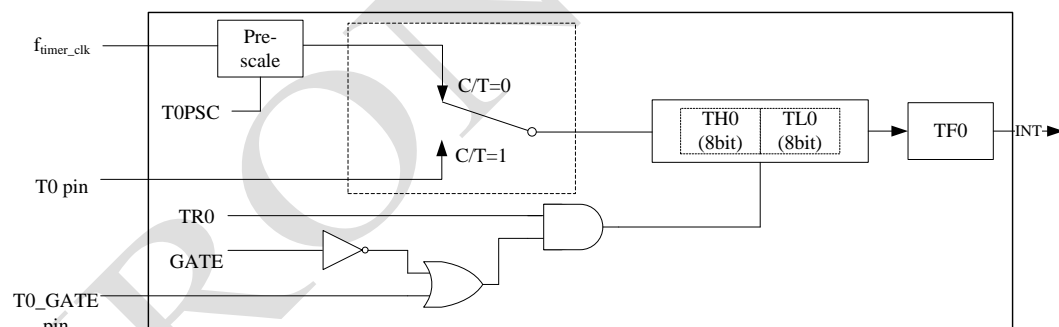


Figure 16 timer0 模式 1 实现框图

当 C/T 为 0，Timer0 用作定时器时， $X=\{TH0[7:0], TL0[7:0]\}$ ，定时时间计算公式为：

$$T_d = (2^{16} - X) \times T_{timer_clk} \times \text{预分频系数}$$

9.3.3 模式 2(8 位自动重装定时器/计数器)

配置寄存器 TMOD 的标志位 TMOD[1:0]=10，使得 Timer0 工作在模式 2。配置寄存器 TMOD[2] (C/T 位) 可选择为定时器模式或计数器模式。

模式 2 中只有低字节寄存器 TL0 计数。当 Timer0 溢出时，TH0 中的值会自动装载到 TL0 中，TF0 置位，同时产生 Timer0 溢出中断。程序进入中断后，该位被自动清零。

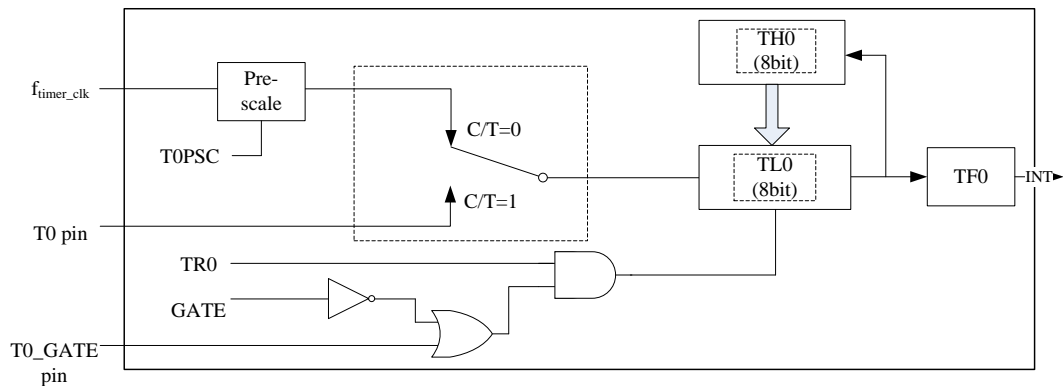


Figure 17 timer0 模式 2 实现框图

当 C/T 为 0, Timer0 用作定时器时, $X=TH0[7:0]$, 定时时间计算公式为:

$$T_d = (2^8 - X) \times T_{timer_clk} \times \text{预分频系数}$$

9.3.4 模式 3(两个 8 比特定器)

配置寄存器 TMOD 的标志位 $TMOD[1:0]=11$, 使得 Timer0 工作在模式 3。配置寄存器 TMOD[2] (C/T 位) 可选择为定时器模式或计数器模式。

模式 3 中, 定时器/计数器的低字节受 TR0 控制, 高字节受 TR1 控制。低字节可作定时器/计数器, 高字节只能作定时器。

当 Timer0 低字节溢出时, TF0 置位, 同时产生 Timer0 溢出中断, 程序进入中断后, 该位被自动清零。

当 Timer0 的高字节溢出时, TF1 置位, 同时产生 Timer1 溢出中断, 程序进入中断后, 该位被自动清零。

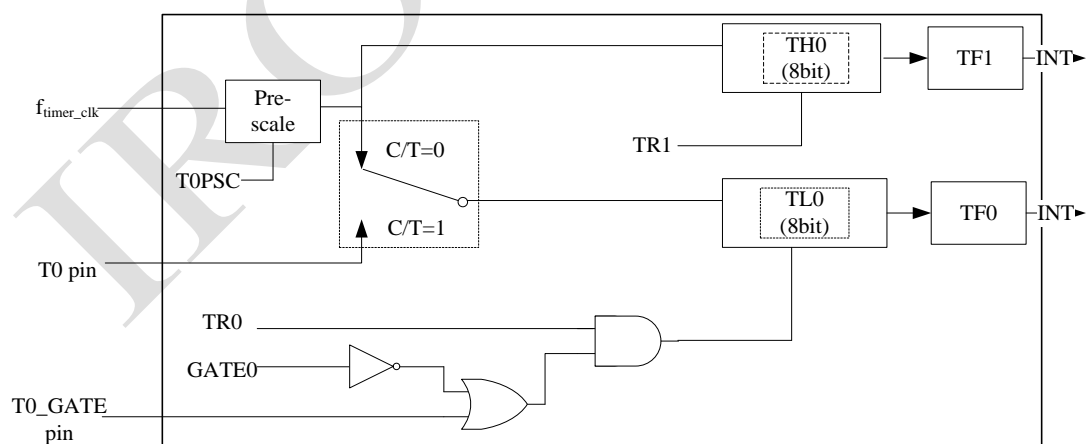


Figure 18 timer0 模式 3 实现框图

当 C/T 为 0, Timer0 用作定时器时, $X=TH0[7:0]$ 或 $TL0[7:0]$, 定时时间计算公式为:

$$T_d = (2^8 - X) \times T_{timer_clk} \times \text{预分频系数}$$

9.4 Timer1 工作模式

配置寄存器 TMOD 中的 M1(TMOD.5)、M0(TMOD.4)，设置 Timer1 的工作模式。

9.4.1 模式 0(13 位定时器/计数器)

配置寄存器 TMOD 的标志位 TMOD[5:4]=00，使得 Timer1 工作在模式 0。配置寄存器 TMOD[6] (C/T 位) 可选择为定时器模式或计数器模式。

Timer1 被分为两个 8 位寄存器，低字节 TL1 和高字节 TH1，TL1 的低 5 位和 TH1 组成一个 13 位的计数器，当 Timer1 溢出时寄存器位 TF1 置位，同时产生 Timer1 溢出中断。程序进入中断后，该位被自动清零。

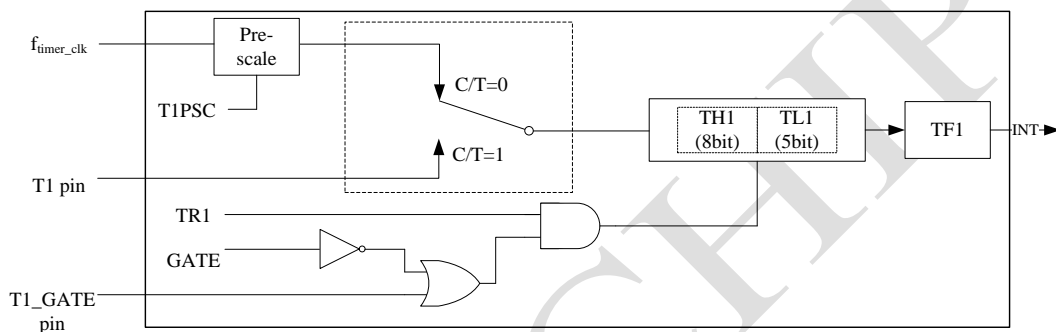


Figure 19 Timer1 模式 0 实现框图

当 C/T 为 0，Timer1 用作定时器， $X=\{TH1[7:0], TL1[4:0]\}$ ，定时时间计算公式为：

$$T_d = (2^{13} - X) \times T_{timer_clk} \times \text{预分频系数}$$

9.4.2 模式 1(16 位定时器/计数器)

配置寄存器 TMOD 的标志位 TMOD[5:4]=01，使得 Timer1 工作在模式 1。配置寄存器 TMOD[6] (C/T 位) 可选择为定时器模式或计数器模式。

模式 1 为 TL1 和 TH1 组成一个 16 位的定时器/计数器，当 Timer1 溢出时寄存器位 TF1 置位，同时产生 Timer1 溢出中断。程序进入中断后，该位被自动清零。

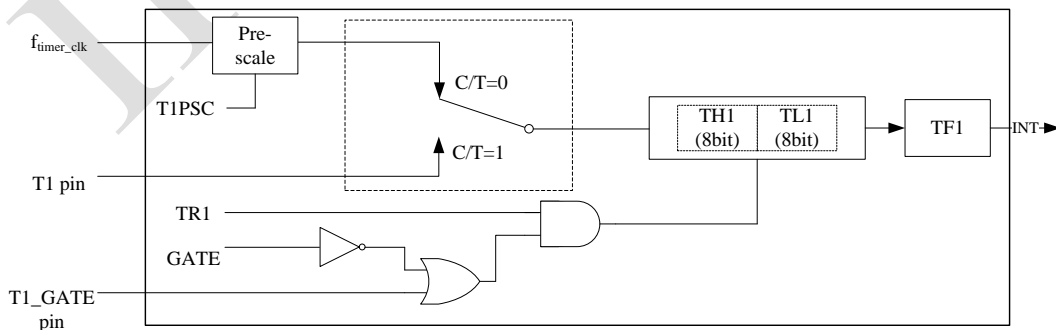


Figure 20 Timer1 模式 1 实现框图

当 C/T 为 0，Timer1 用作定时器时， $X=\{TH1[7:0], TL1[7:0]\}$ ，定时时间计算公式为：

$$T_d = (2^{16} - X) \times T_{timer_clk} \times \text{预分频系数}$$

9.4.3 模式 2(8 位自动重装定时器/计数器)

配置寄存器 TMOD 的标志位 TMOD[5:4]=10, 使得 Timer1 工作在模式 2。配置寄存器 TMOD[6] (C/T 位) 可选择为定时器模式或计数器模式。

模式 2 中只有低字节寄存器 TL1 进行计数。当 Timer1 溢出时, TH1 中的值会自动装载到 TL1 中, TF1 置位, 同时产生 Timer1 溢出中断。程序进入中断后, 该位被自动清零。

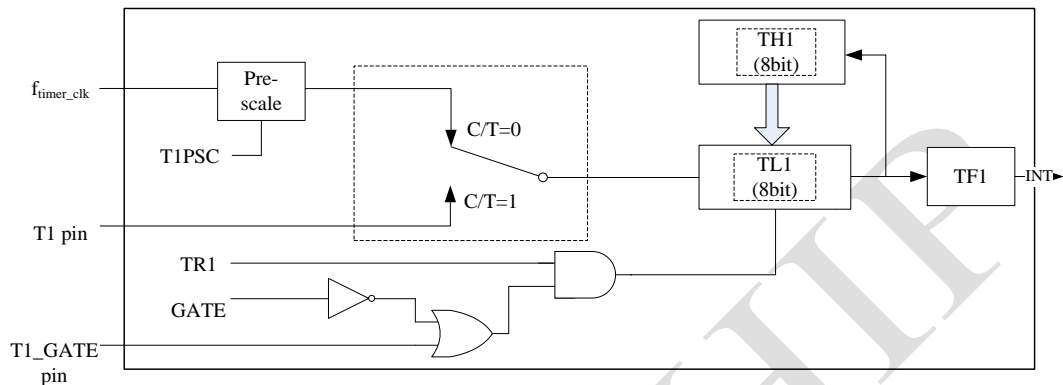


Figure 21 Timer1 模式 2 实现框图

当 C/T 为 0, Timer1 用作定时器时, $X=TH1[7:0]$, 定时时间计算公式为:

$$T_d = (2^8 - X) \times T_{timer_clk} \times \text{预分频系数}$$

9.5 Timer2 工作模式

配置寄存器 T2CON 中的 M1(T2CON.1)、M0(T2CON.0), 设置 Timer2 的工作模式。

9.5.1 模式 0(13 位定时器/计数器)

配置寄存器 T2CON 的标志位 T2CON[1:0]=00, 使得 Timer2 工作在模式 0。配置寄存器 T2CON[2] (C/T 位) 可选择为定时器模式或计数器模式。

Timer2 被分为两个 8 位寄存器, 低字节 TL2 和高字节 TH2, TL2 的低 5 位和 TH2 组成一个 13 位的计数器, 当 Timer2 溢出时寄存器位 TF2 置位, 同时产生 Timer2 溢出中断。程序进入中断后, 该位被自动清零。

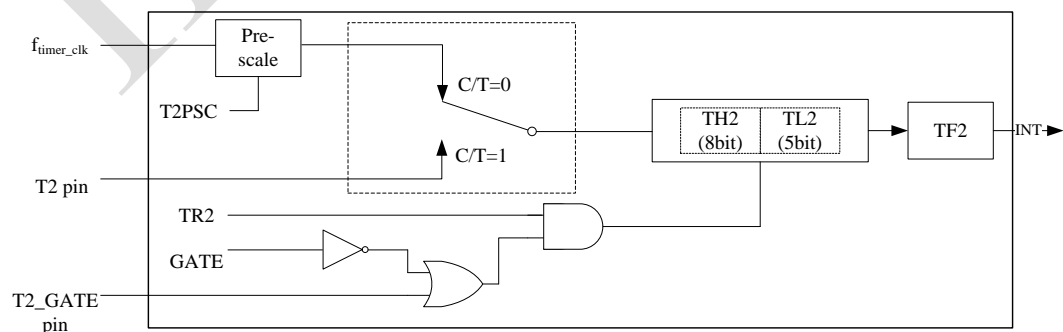


Figure 22 Timer2 模式 0 实现框图

当 C/T 为 0, Timer2 用作定时器, $X=\{TH2[7:0], TL2[4:0]\}$, 定时时间计算公式为:

$$T_d = (2^{13} - X) \times T_{timer_clk} \times \text{预分频系数}$$

9.5.2 模式 1(16 位定时器/计数器)

配置寄存器 T2CON 的标志位 T2CON[1:0]=01, 使得 Timer2 工作在模式 1。配置寄存器 T2CON[2] (C/T 位) 可选择为定时器模式或计数器模式。

模式 1 为 TL2 和 TH2 组成一个 16 位的计数器, 当 Timer2 溢出时寄存器位 TF2 置位, 同时产生 Timer2 溢出中断。程序进入中断后, 该位被自动清零。

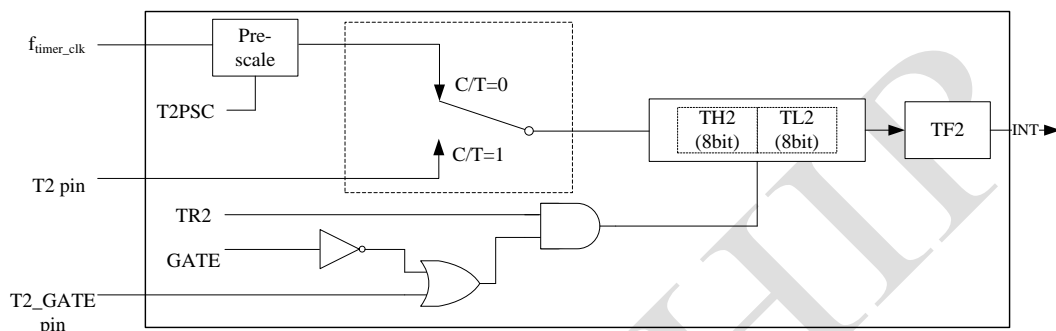


Figure 23 timer2 模式 1 实现框图

当 C/T 为 0, Timer2 用作定时器时, $X=\{TH2[7:0], TL2[7:0]\}$, 定时时间计算公式为:

$$T_d = (2^{16} - X) \times T_{timer_clk} \times \text{预分频系数}$$

9.5.3 模式 2(16 位自动重装定时器/计数器)

配置寄存器 T2CON 的标志位 T2CON[1:0]=10, 使得 Timer2 工作在模式 2。配置寄存器 T2CON[2] (C/T 位) 可选择为定时器模式或计数器模式。

模式 2 中 TL2 和 TH2 组成一个 16 位的计数器, 当 Timer2 溢出时, RH2 中的值会自动装载到 TH2 中, RL2 中的值会自动装载到 TL2 中, TF2 置位, 同时产生 Timer2 溢出中断。程序进入中断后, 该位被自动清零。

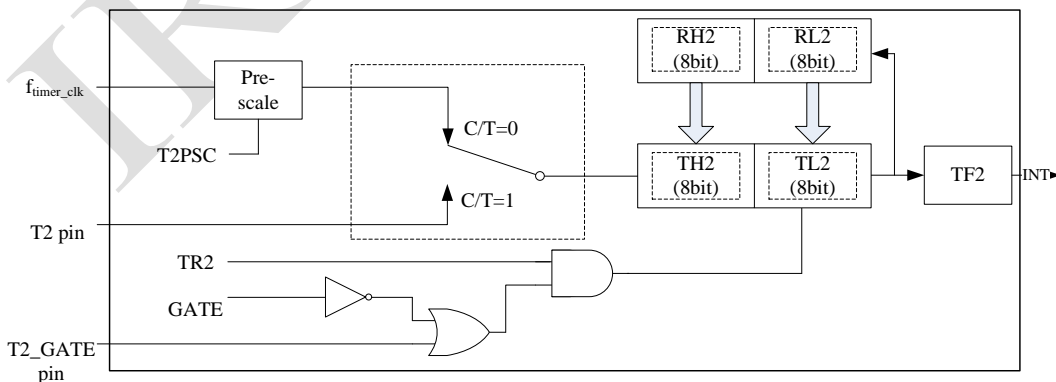


Figure 24 timer2 模式 2 实现框图

当 C/T 为 0, Timer2 用作定时器时, $X=\{RH2[7:0], RL2[7:0]\}$, 定时时间计算公式为:

$$T_d = (2^{16} - X) \times T_{timer_clk} \times \text{预分频系数}$$

9.5.4 模式 3(1 个 8 比特定计时器/计数器)

配置寄存器 T2CON 的标志位 T2CON [1:0]=11，使得 Timer2 工作在模式 3。配置寄存器 T2CON[2]（C/T 位）可选择为定时器模式或计数器模式。

模式 3 中，低字节可作定时器/计数器。

当 Timer2 低字节溢出时，TF2 置位，同时产生 Timer2 溢出中断，程序进入中断后，该位被自动清零。

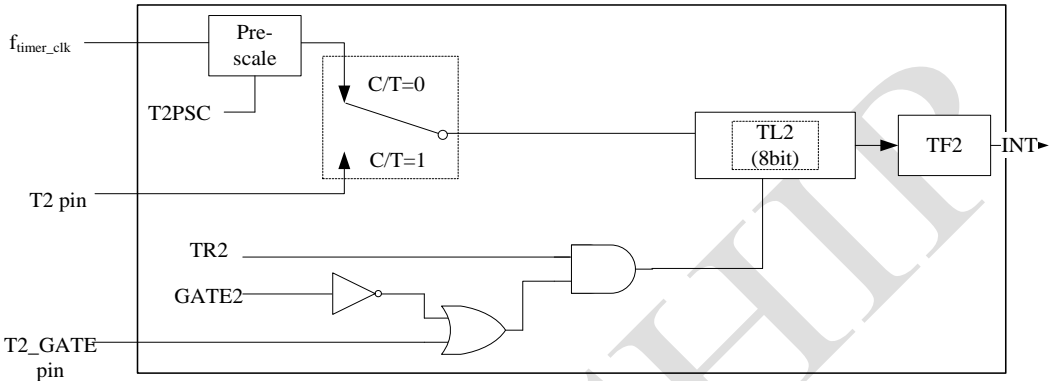


Figure 25 timer2 模式 3 实现框图

当 C/T 为 0，Timer2 用作定时器时，X=TL2[7:0]，定时时间计算公式为：

$$T_d = (2^8 - X) \times T_{timer_clk} \times \text{预分频系数}$$

9.6 TIMER 寄存器

TIMER 模块有 13 个寄存器，对应的 SFRPAGE 为 0，具体说明如 Table 85 所示。

Table 85 TIMER 寄存器（SFRPAGE=0x00）

地址	寄存器名	类型	复位值	功能描述
0x88	TCON	W/R	0x00	Timer0/1 控制寄存器
0x89	TMOD	W/R	0x00	Timer0/1 模式寄存器
0x8A	TL0	W/R	0x00	Timer0 低 8 位
0x8B	TL1	W/R	0x00	Timer1 低 8 位
0x8C	TH0	W/R	0x00	Timer0 高 8 位
0x8D	TH1	W/R	0x00	Timer1 高 8 位
0x8E	TPSC	W/R	0x00	Timer0/1 预分频控制器
0xD1	T2CON	W/R	0x00	Timer2 控制寄存器
0xD2	RL2	W/R	0x00	Timer2 重载寄存器低 8 位
0xD3	RH2	W/R	0x00	Timer2 重载寄存器高 8 位
0xD4	TL2	W/R	0x00	Timer2 低 8 位

地址	寄存器名	类型	复位值	功能描述
0xD5	TH2	W/R	0x00	Timer2 高 8 位
0xD6	T2PSC	W/R	0x00	Timer2 预分频控制器

9.6.1 Timer0/1 控制寄存器(TCON)[0x88]

Table 86 Timer0/1 控制寄存器(TCON)

参数名	比特位	属性	复位值	描述
UBSEL	0	W/R	0	UART 波特率产生选择位： 0: 使用 Timer 1 产生波特率 1: 使用 SRELL 和 SRELH 寄存器计数产生波特率
Reserved	1	R	0	保留
SMOD	2	W/R	0	UART 双倍波特率使能控制位： 0: 正常模式 1: 使用双倍波特率
Reserved	3	R	0	保留
TR0	4	W/R	0	Timer0 运行控制位： 0: 停止 1: 启动
TF0	5	W/R	0	Timer0 溢出中断标志： 0: 未溢出 1: 溢出
TR1	6	W/R	0	Timer1 运行控制位： 0: 停止 1: 启动
TF1	7	W/R	0	Timer1 溢出中断标志： 0: 未溢出 1: 溢出

9.6.2 Timer0/1 模式寄存器(TMOD)[0x89]

Table 87 Timer0/1 模式寄存器(TMOD)

参数名	比特位	属性	复位值	描述
M0	0	W/R	0	Timer0 模式，见 Table 88 Timer0/1 工作模

参数名	比特位	属性	复位值	描述
M1	1	W/R	0	式选择描述
C/T	2	W/R	0	Timer0 用作定时器/计数器选择位： 0: 定时器 1: 计数器
GATE	3	W/R	0	Timer0 门控控制位： 0: T0_GATE 不起作用 1: T0_GATE 可起作用
M0	4	W/R	0	Timer1 模式，见 Table 88 Timer0/1 工作模式选择描述
M1	5	W/R	0	
C/T	6	W/R	0	Timer1 用作定时器/计数器选择位： 0: 定时器 1: 计数器
GATE	7	W/R	0	Timer1 门控控制位： 0: T1_GATE 不起作用 1: T1_GATE 可起作用

Table 88 Timer0/1 工作模式选择

M1	M0	模式	说明
0	0	0	13 位定时器/计数器，TL0 (TL1) 中的低 5 位和 TH0(TH1)中的 8 位。
0	1	1	16 位定时器/计数器。
1	0	2	8 位定时器/计数器，具有自动装载功能，在 TL0(TL1)溢出时装载 TH0(TH1)的值。
1	1	3	Timer0 设置为该模式时，分成两个 8 位的独立计数器：TL0 使用控制位 TR0，溢出时将 TF0 置位；TH0 使用控制位 TR1，溢出时将 TF1 置位，该模式下 Timer1 不能工作。 Timer1 设置为该模式时停止工作。

9.6.3 Timer0 低 8 位(TL0)[0x8A]

Table 89 Timer0 低 8 位(TL0)

参数名	比特位	属性	复位值	描述
TL0	7-0	W/R	0	Timer0 低 8 位

9.6.4 Timer1 低 8 位(TL1)[0x8B]

Table 90 Timer1 低 8 位(TL1)

参数名	比特位	属性	复位值	描述
TL1	7-0	W/R	0	Timer1 低 8 位

9.6.5 Timer0 高 8 位(TH0)[0x8C]

Table 91 Timer0 高 8 位(TH0)

参数名	比特位	属性	复位值	描述
TH0	7-0	W/R	0	Timer0 高 8 位

9.6.6 Timer1 高 8 位(TH1)[0x8D]

Table 92 Timer1 高 8 位(TH1)

参数名	比特位	属性	复位值	描述
TH1	7-0	W/R	0	Timer1 高 8 位

9.6.7 Timer0/1 预分频控制器(TPSC)[0x8E]

Table 93 Timer0/1 预分频控制器(TPSC)

参数名	比特位	属性	复位值	描述
T0PSC	1-0	W/R	0	Timer0 时钟预分频系数： 00: 1 分频 01: 8 分频 10: 64 分频 11: 256 分频
T1PSC	3-2	W/R	0	Timer1 时钟预分频系数： 00: 1 分频 01: 8 分频 10: 64 分频 11: 256 分频
Reserved	7-4	R	0	保留

9.6.8 Timer2 控制寄存器(T2CON)[0xD1]

Table 94 Timer2 控制寄存器(T2CON)

参数名	比特位	属性	复位值	描述
M0	0	W/R	0	Timer2 模式，见 Table 95 Timer2 工作模式选择
M1	1	W/R	0	

参数名	比特位	属性	复位值	描述
C/T	2	W/R	0	Timer2 用作定时器/计数器选择位： 0: 定时器 1: 计数器
GATE	3	W/R	0	Timer2 门控控制位： 0: T2_GATE 不起作用 1: T2_GATE 可起作用
TR2	4	W/R	0	Timer2 运行控制位： 0: 停止 1: 启动
TF2	5	W/R	0	Timer2 溢出中断标志： 0: 未溢出 1: 溢出
Reserved	7-6	R	0	保留

Table 95 Timer2 工作模式选择

M1	M0	模式	说明
0	0	0	13 位定时器/计数器，TL2 中的低 5 位和 TH2 中的 8 位。
0	1	1	16 位定时器/计数器。
1	0	2	16 位定时器/计数器，具有自动装载功能，在 Timer2 溢出时，{TH2,TL2}自动装载{RH2,RL2}的值。
1	1	3	8 位定时器/计数器。

9.6.9 Timer2 重载寄存器低 8 位(RL2)[0xD2]

Table 96 Timer2 重载寄存器低 8 位(RL2)

参数名	比特位	属性	复位值	描述
RL2	7-0	W/R	0	Timer2 重载寄存器低 8 位

9.6.10 Timer2 重载寄存器高 8 位(RH2)[0xD3]

Table 97 Timer2 重载寄存器高 8 位(RH2)

参数名	比特位	属性	复位值	描述
RH2	7-0	W/R	0	Timer2 重载寄存器高 8 位

9.6.11 Timer2 低 8 位(TL2)[0xD4]

Table 98 Timer2 低 8 位(TL2)

参数名	比特位	属性	复位值	描述
TL2	7-0	W/R	0	Timer2 低 8 位

9.6.12 Timer2 高 8 位(TH2)[0xD5]

Table 99 Timer2 高 8 位(TH2)

参数名	比特位	属性	复位值	描述
TH2	7-0	W/R	0	Timer2 高 8 位

9.6.13 Timer2 预分频控制器(T2PSC)[0xD6]

Table 100 Timer2 预分频控制器(T2PSC)

参数名	比特位	属性	复位值	描述
T2PSC	1-0	W/R	0	Timer2 时钟预分频系数: 00: 1 分频 01: 8 分频 10: 64 分频 11: 256 分频
Reserved	7-2	R	0	保留

10 高级定时器 0 (PWM0)

10.1 概述

JMT18F003PLUS 高级定时器 0 主要包含一个 16 位的向上/向下自动重装载计数器、八个 16 位的比较/捕获寄存器、一个可编程的时钟预分频器和多个控制寄存器。

本定时器主要用途有：

- 产生多种输出波形（输出比较波形、PWM 波形、嵌入死区的互补 PWM 波形等）
- 测量输入信号的脉冲宽度（输入捕获功能）
- 作为通用定时器使用
- 作为霍尔传感器接口
- 作为正交增量编码器接口

本定时器可以产生 3 对互补的 PWM 输出加 1 路独立的 PWM 输出。

通过控制定时器的时钟分频（两种方式实现：控制 CRM 模块中 PWM 时钟分频器和控制定时器自身的时钟分频器），本定时器可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

10.2 特性

本定时器的特性：

- 16 向上、向下和向上/向下自动装载计数器。
- 16 位可编程分频器，计数器时钟频率的分频系数为 1~65536 之间的任意值。
- 4 个独立通道，每个通道支持以下功能：
 - 输入捕获功能
 - 输出比较功能
 - PWM 波形生成（边沿或中央对齐模式）
 - 单脉冲模式输出
 - 死区时间可编程的互补输出
- 在互补模式下可以输出 3 对互补的 PWM 波形和 1 路独立的 PWM 波形。
- 可以在指定数目的计数器周期之后更新定时器中的预装载寄存器。
- 相位偏移功能：可以选择 CC0~CC3 中的任一通道，使其输出波形比其他通道的输出延迟一定的时间，延迟时间可由软件编程控制。
- 刹车功能：外部刹车信号可以将定时器输出信号置于复位状态或者一个已知状态，也可以通过软件配置刹车事件实现刹车功能。
- 如下事件发生时产生中断

- 更新：计数器向上溢出、向下溢出，计数器初始化（通过软件或内部/外部触发）
 - 触发事件（计数器启动，停止，初始化或由内部/外部触发计数）产生触发中断
 - 输入捕获，当捕获数据时产生捕获中断
 - 输出比较，当计数器与输出比较寄存器匹配时产生比较中断
 - 刹车信号输入
 - 软件配置 COM 事件，更新通道输出控制寄存器。
- 正交增量编码器接口功能
 - 霍尔传感器接口功能

10.3 框图

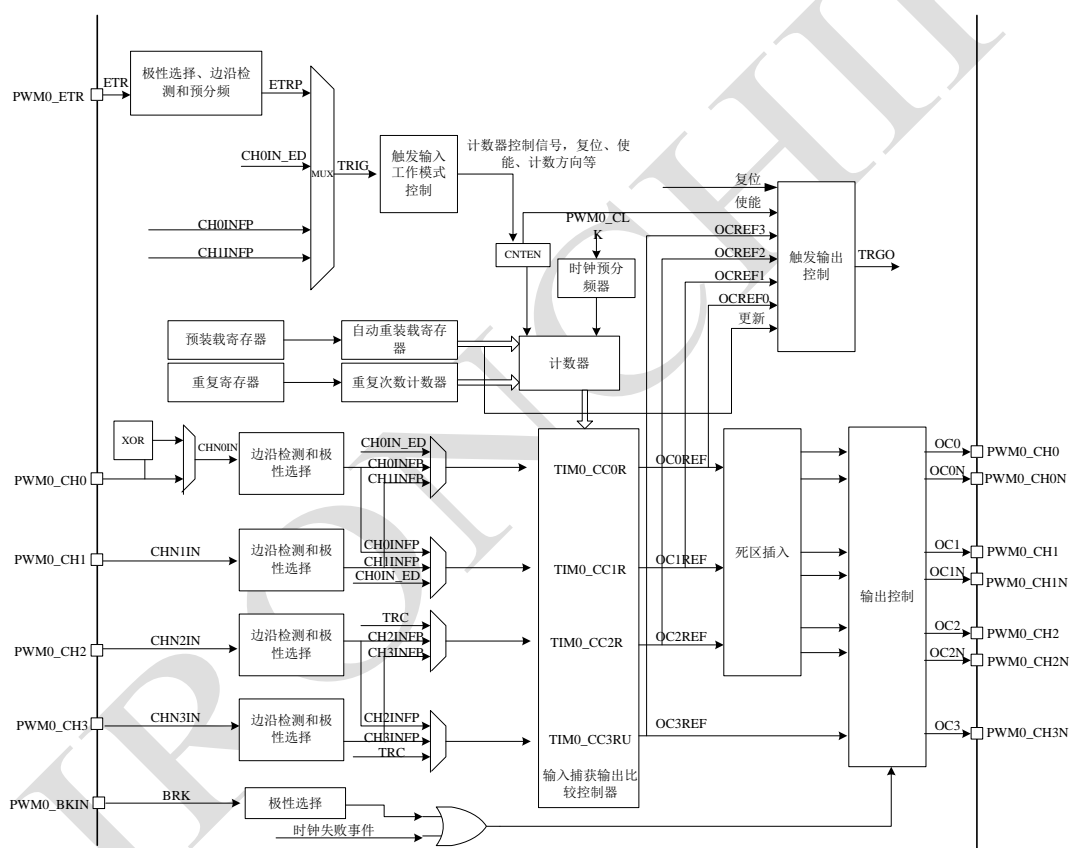


Figure 26 JMT18F003PLUS 高级定时器 0 (TIM0/PWM0) 框图

10.4 输入输出引脚配置

有些引脚可以复用为高级定时器的输入输出功能，详细描述见 17.5 节。

当高级定时器配置为输入捕获功能时（寄存器 TIM0_CCxMR³的 CCxMS 位配置为 01/10/11），外部信号通过 PWM0_CHx 或 PWM0_ETR 通道输入，刹车信号通过 PWM0_BKIN 通道输入；当高级定时器配置为输出比较功能时（寄存器 TIM0_CCxMR 的 CCxMS 位配置

³ CCxMR 中的 x 比较通道标号，x=0, 1, 2, 3。

为 00)，高级定时器的输出信号通过 PWM0_CHx 和 PWM0_CHxN 输出。

当高级定时器输出互补 PWM 波形时，PWM0_CHx 和 PWM0_CHxN 是通道 CCx 的一对互补 PWM 输出。

10.5 功能描述

高级定时器的结构主要分为五部分：计数器部分、从模式控制器、主模式控制器、四个通道的输入捕获控制器和四个通道的输出比较控制器。下面分别介绍高级定时器各个部分的功能和各种工作模式的控制方法。

10.5.1 计数器

10.5.1.1 计数器的计数周期

高级定时器 TIM0 包含一个 16 位的计数器，计数器的周期值保存在自动重装载寄存器中，自动重装载寄存器是预先装载的（TIM0_ARR 是其预装载寄存器），根据 TIM0_CONR2 寄存器中的自动装载使能位(ARPLE)的设置，预装载寄存器的值被立即或在更新事件时传送到影子寄存器（软件不可见）。写或读自动重装载寄存器将访问预装载寄存器(TIM0_ARR)。

10.5.1.2 计数器的使能控制

计数器使能受到寄存器 TIM0_CNTEN 的 CNTEN 位和从模式控制器控制，具体控制方式如下：

- 当寄存器 TIM0_TGICR0 的 SMS 位配置为 000 时，高级定时器关闭从模式，此时计数器使能只受 TIM0_CNTEN 的 CNTEN 位控制，当 CNTEN 配置为 1 时，计数器计数，当 CNTEN 配置为 0 时，计数器停止计数。
- 当寄存器 TIM0_TGICR0 的 SMS 位配置为 110 时，高级定时器工作在触发模式下，此时将寄存器 TIM0_CONR2 的 CNTEN 位配置为 0，计数器在触发输入的有效沿启动（但不复位）。
- 当寄存器 TIM0_TGICR0 的 SMS 位配置为其他值时，计数器使能同时受 TIM0_CNTEN 的 CNTEN 位和触发信号的控制，此时需要将寄存器 CNTEN 位配置为 1，计数器在输入触发信号的控制下工作。

寄存器 TIM0_CNTEN 的 CNTEN 位置位后意味着计数器即将开始计数，高级定时器也即将进入工作状态，此时其他寄存器应该完成配置，所以建议 TIM0_CNTEN 的 CNTEN 位最后配置。

10.5.1.3 计数器预分频控制器

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。通过配置寄存器 TIM0_PSCL 和 TIM0_PSCH 可以改变分频系数，此寄存器带有缓存器，所以可以在工作时配置，新的预分频器的参数在下一次更新事件（UEV）到来时被采用。Figure 27 为预分频

器的参数由 1 分频变为 4 分频时，计数器的时序图。

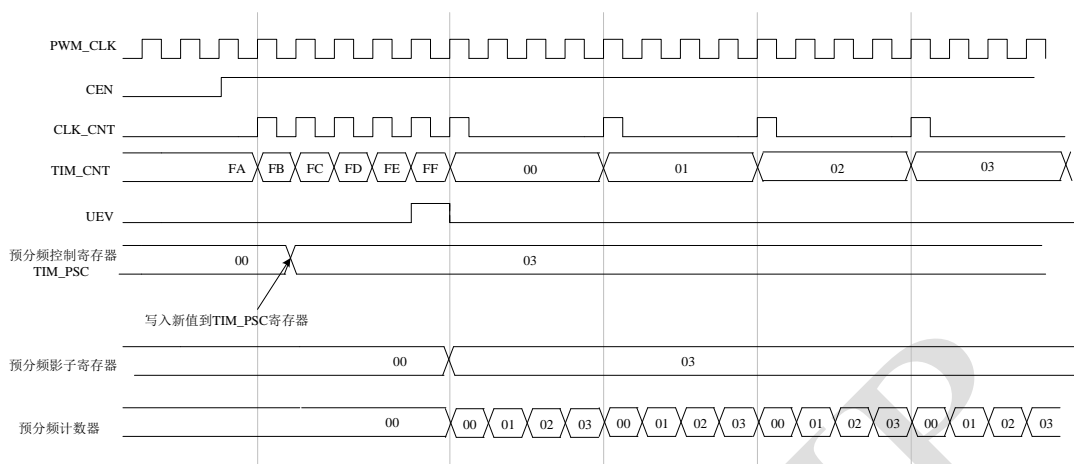


Figure 27 预分频器的参数由 1 分频变为 4 分频时，计数器的时序图

10.5.1.4 计数器计数模式控制

计数器有三种计数模式：向上计数模式、向下计数模式和中央对齐计数模式（向上向下交替计数），配置 TIM0_CONR2 的 CNTMC 位可以设置计数器的计数模式：

当 CNTMC 配置为 00 时，计数器工作在边沿计数模式下，此时可以通过配置 TIM0_CONR2 寄存器的 CNTDIR 位设置计数器的计数方向。

当 CNTMC 配置为 01/10/11 时，计数器工作在中央对齐模式下，此时不可以配置 TIM0_CONR2 寄存器的 CNTDIR 位，计数器的方向由硬件自动设置。

当寄存器 TIM0_TGICR0 的 SMS 位配置为 001/010/011 时，即高级定时器工作在增量编码器接口模式下，TIM0_CONR2 的 CNTDIR 位也是不可写的，此时计数器的计数方向由硬件根据触发信号确定。

10.5.1.4.1 向上计数模式

当寄存器 TIM0_CONR2 的 CNTMC 位配置为 00 且 CNTDIR 位配置为 0 时，计数器向上计数，计数器的值从 0 开始累加，一直累加到计数器的重载值（TIM0_ARR 中的值）。一旦计数器达到了重载值，寄存器将从 0 开始重新计数。若设置了重复周期计数的值（TIM0_RCR），在计数器向上溢出次数达到设置的重复周期次数时才产生更新事件；否则每当计数器溢出时会产生更新事件。在此种计数模式下，设置 TIM0_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，计数器初始化为 0；

更新事件产生的规则：

- 设置寄存器 TIM0_CONR3 中的 UPOUDIS 位为 1，可以禁止上溢更新事件的产生，若更新事件被禁止，预分频寄存器的数值不变，同时计数器按预分频的频率进行计数。禁止更新事件产生的目的为：避免在向预装载寄存器中写入新值时更新影子寄存器。

- 设置 TIM0_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，可以产生一个更新事件。
- 如果寄存器 TIM0_CONR2 中的 URC 位被设置为 1，设置 UG 位为 1 将产生一个更新事件 UEV，但硬件不更新中断标志位 SOFTUIF（即不产生软更新中断）。这是为了避免在捕获模式下初始化计数器时产生更新中断。

当发生更新事件时，硬件做如下处理：

- 上溢更新标志位 UOVERIF 自动置 1，若上溢更新中断使能（TIM0_CONR3 寄存器的 UPOIE 位为 1）有效，则产生上溢更新中断。
- 若通过软件配置 UG 位为 1 产生更新事件，硬件根据 URC 位的配置更新中断标志位 SOFTUIF，若软更新中断使能（TIM0_CONR3 寄存器的 SOFTUPIE 位）有效，则产生软件更新中断。
- 自动重载寄存器被重新置入预装载寄存器的值。
- 重复周期计数器被重新加载为 TIM0_RCR 寄存器的内容。
- 预分频器的影子寄存器被置入预装载寄存器（TIM0_PSCL 和 TIM0_PSCH 寄存器）的值。

Figure 28 和 Figure 29 为向上计数模式下计数器的时序，计数器时钟的预分频系数为 1（2 分频）。

Figure 28 中自动装载寄存器的预装载功能使能，新配置的计数器周期值在更新事件时传递到影子寄存器中，下一周期生效。Figure 29 的自动装载寄存器的预装载功能关闭，新配置的计数器周期值立即传送到影子寄存器中，在本周期即生效。

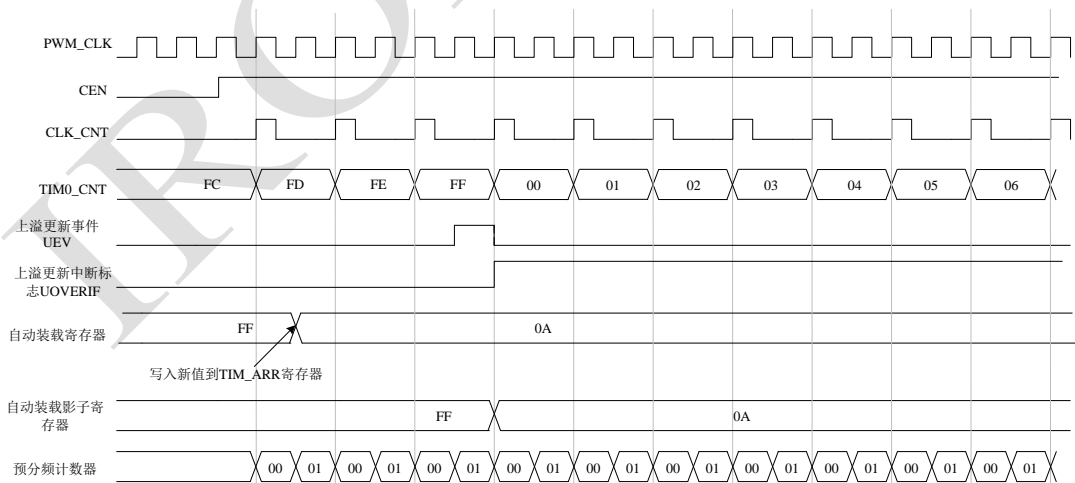


Figure 28 向上计数模式下，自动装载寄存器的预装载功能使能时计数器的时序图

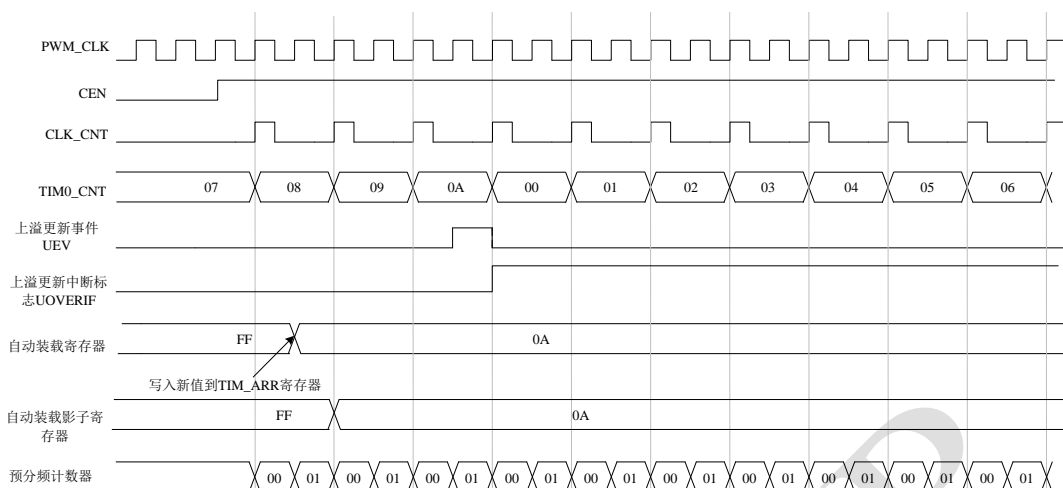


Figure 29 向上计数模式下，当自动装载寄存器的预装载功能不使能时计数器的时序图

10.5.1.4.2 向下计数模式

当寄存器 TIM0_CONR2 的 CNTMC 位配置为 00 且 CNTDIR 位配置为 1 时，计数器向下计数，计数器的值从自动装入的值 TIM0_ARR 开始递减到 0，然后从自动装入的值重新开始计数并且产生一个计数器向下溢出事件。若设置了重复周期计数的值，在计数器向下溢出次数达到设置的重复周期次数时才产生更新事件；否则每当计数器向下溢出时会产生更新事件。在此种计数模式下，当设置 TIM0_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，计数器初始化为周期值 TIM0_ARR。

更新事件产生的规则：

- 设置寄存器 TIM0_CONR3 中的 DOWNOUDIS 位为 1，可以禁止下溢更新事件的产生。若更新事件被禁止，预分频寄存器的数值不变，同时计数器按预分频的频率进行计数。禁止更新事件产生的目的为：避免在向预装载寄存器中写入新值时更新影子寄存器。
- 设置 TIM0_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，可以产生一个更新事件。
- 如果 TIM0_CONR3 中的 URC 位被设置为 1，设置 UG 位为 1 将产生一个更新事件 UEV，但硬件不更新中断标志 SOFTUIF（即软更新中断标志）。这是为了避免在捕获模式下初始化计数器时产生更新中断。

当发生更新事件时，硬件做如下处理：

- 下溢更新标志位 DOVERIF 自动置 1，若下溢更新中断使能（TIM0_CONR3 寄存器的 DOWNOIE 位）有效，则产生下溢更新中断。
- 若通过软件配置 UG 位为 1 产生更新事件，硬件根据 URC 位的配置更新中断标志位 SOFTUIF，若软更新中断使能（TIM0_CONR3 寄存器的 SOFTUPIE 位）有效，则产生软件更新中断。

- 自动重载寄存器被重新置入预装载寄存器的值。
- 重复周期计数器被重新加载为 TIM0_RCR 寄存器的内容。
- 预分频器的影子寄存器被置入预装载寄存器(TIM0_PSCL 和 TIM0_PSCH 寄存器)的值。

值得注意的是，在向下计数模式下，自动重载寄存器的值在计数器重载入之前更新，因此一下个周期将是预期的值。Figure 30 为向下计数模式下计数器的时序图。

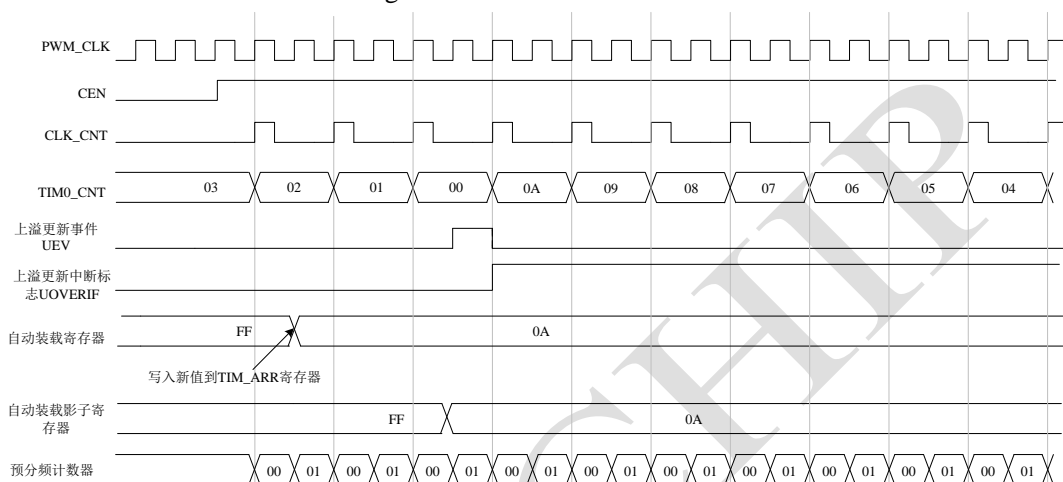


Figure 30 向上计数时计数器的时序图

10.5.1.4.3 中央对齐计数模式

寄存器 TIM0_CONR2 的 CNTMC 位配置为 00/01/11 时，计数器的计数模式为中央对齐模式，中央对齐模式下，寄存器 TIM0_CONR2 的 CNTDIR 位是不可写的，它由硬件更新并指示当前的计数方向。

在这种模式下，计数器从 0 开始计数到自动加载值（TIM0_ARR 寄存器）减 1，产生一个计数器上溢事件，然后向下计数到 1 并且产生一个计数器下溢事件，然后再从 0 开始计数。在此种计数模式下，当设置 TIM0_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，计数器初始化为 0；

每一次计数上溢和每一次计数下溢都可以产生更新事件。

更新事件产生的规则：

- 设置寄存器 TIM0_CONR3 中的 UPOUDIS 位为 1，可以禁止上溢更新事件的产生，设置寄存器 TIM0_CONR3 中的 DOWNOUDIS 位为 1，可以禁止下溢更新事件的产生。若更新事件被禁止，预分频寄存器的数值不变，同时计数器按预分频的频率进行计数。禁止更新事件产生的目的为：避免在向预装载寄存器中写入新值时更新影子寄存器。
- 设置 TIM0_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时可以产生一个软件更新事件。

- 如果 TIM0_CONR3 中的 URC 位被设置为 1，设置 UG 位为 1 将产生一个更新事件 UEV，但硬件不更新 SOFTUIF 标志（即不产生软更新中断）。这是为了避免在捕获模式下初始化计数器时产生更新。

当发生更新事件时，硬件做如下处理：

- 若更新事件是上溢更新事件，上溢更新标志位 UOVERIF 自动置位，若上溢更新中断使能（TIM0_CONR3 寄存器的 UPOIE 位）有效，则产生上溢更新中断。若更新事件是下溢更新事件，下溢更新标志位 DOVERIF 自动置位，若设置了下溢更新中断使能（TIM0_CONR3 寄存器的 DOWNOIE 位）有效，则产生下溢更新中断。
- 若配置 UG 位为 1 产生更新事件时，硬件根据 URC 位更新中断标志位 SOFTUIF，若软更新中断使能（TIM0_CONR3 寄存器的 SOFTUPIE 位）有效，则产生软件更新中断。
- 自动重载寄存器被重新置入预装载寄存器的值。
- 重复周期计数器被重新加载为 TIM0_RCR 寄存器的值。
- 预分频器的影子寄存器被置入预装载寄存器的值（TIM0_PSCL 和 TIM0_PSCH 寄存器的内容）。

值得注意的是，如果因为计数器下溢而产生更新，自动重载寄存器的值在计数器重载入之前更新，因此一下个周期将是预期的值。Figure 31 和 Figure 32 为中央计数模式下计数器的时序。Figure 31 的自动装载寄存器的预装载功能使能，新配置的计数器周期值在更新事件时传递到影子寄存器中，下一周期生效。Figure 32 的自动装载寄存器的预装载功能关闭，新配置的计数器周期值立即传送到影子寄存器中，在本周期即生效。

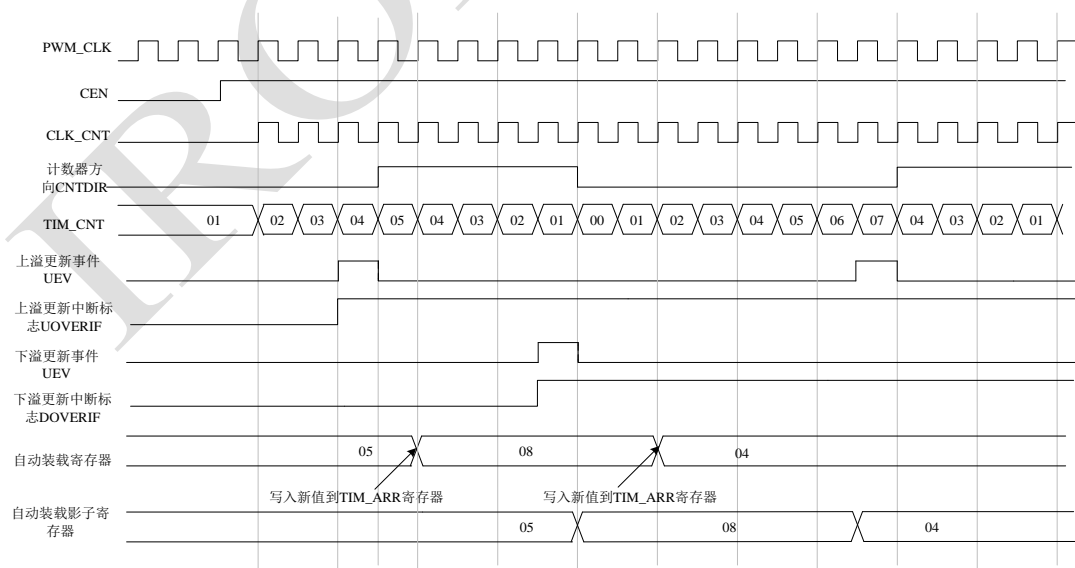


Figure 31 中央计数模式下，自动装载寄存器的预装载功能使能时计数器的时序图

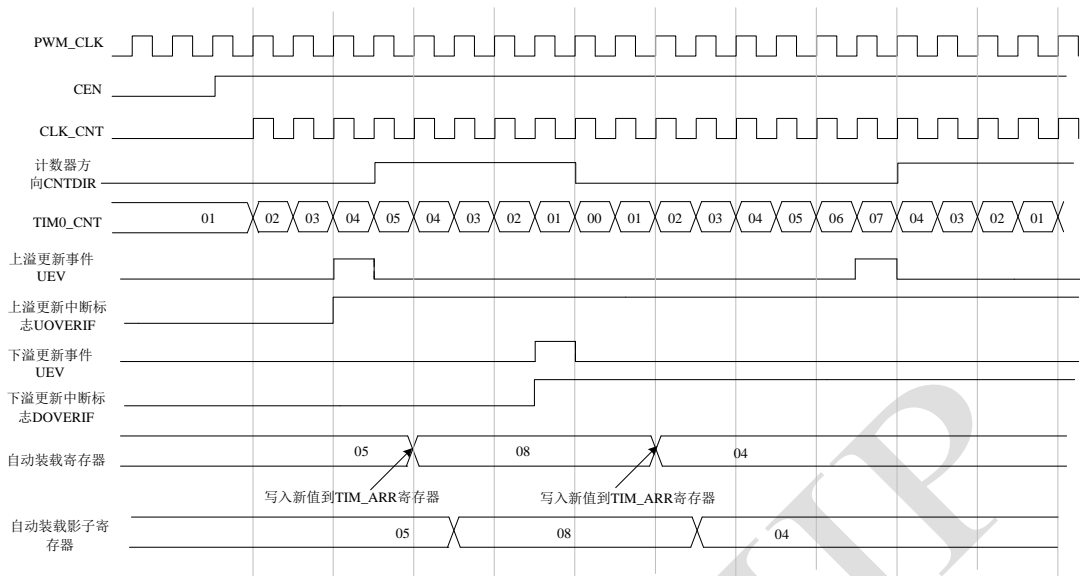


Figure 32 中央计数模式下，自动装载寄存器的预装载功能不使能时计数器的时序图

10.5.1.5 计数器的重复周期计数控制器

借助重复周期计数功能，可以在计数器连续上溢/下溢 N 次后，才会产生计数器更新事件，更新各寄存器的内容。

周期计数器递减的条件有：

- 向上计数模式下的每次计数器上溢；
- 向下计数模式下的每次计数器下溢；
- 中央对齐模式下的每次计数器上溢/下溢。

周期计数器是自动加载的向下计数器，重复次数由寄存器 TIM0_RCR 的值定义。当更新事件由软件产生或者通过硬件的从模式控制器产生时，无论周期计数器的值是多少，立即发生更新事件，并且寄存器 TIM0_RCR 的内容被重新载入到周期计数器。

Figure 33 为向上计数模式下周期计数控制器对更新事件的控制，图中箭头表示产生更新事件，当寄存器 TIM0_CRC 配置为 0 时，每当计数器溢出时产生更新事件，当寄存器 TIM0_CRC 配置为 1 时，计数器连续 2 次溢出时产生更新事件，当 TIM0_CRC 配置为 N 时，计数器连续 N+1 次溢出时产生更新事件。若软件配置更新事件时，立即发生更新事件。

Figure 34 为向下计数模式下周期计数控制器对更新事件的控制。

Figure 35 为中央对齐计数模式下周期计数控制器对更新事件的控制。

Figure 36 为中央对齐模式下，当重复周期次数为 3 时，更新事件产生的时序图。

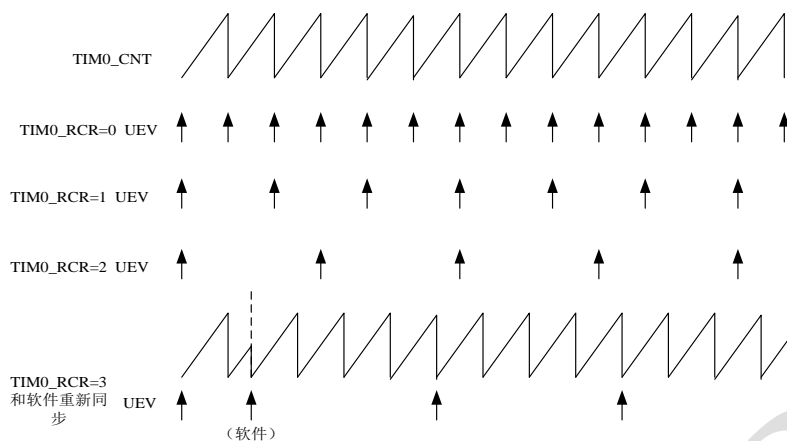


Figure 33 向上计数模式下，重复计数次数分别为 0、1、2、3 时更新事件的产生

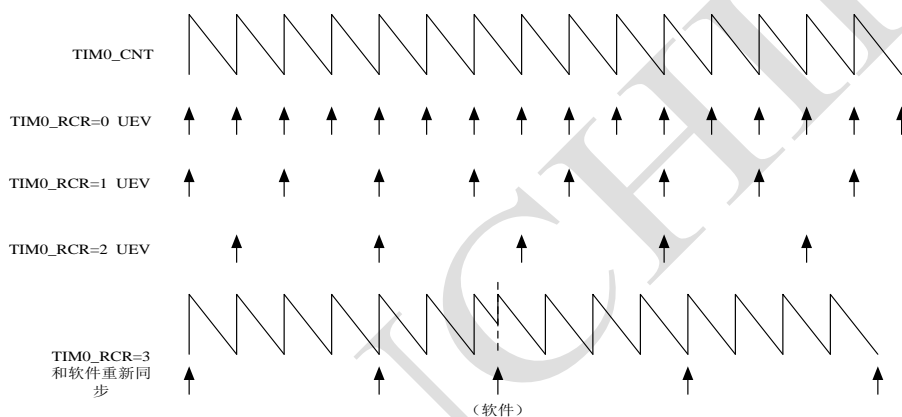


Figure 34 向下计数模式下，重复计数次数分别为 0、1、2、3 时更新事件的产生

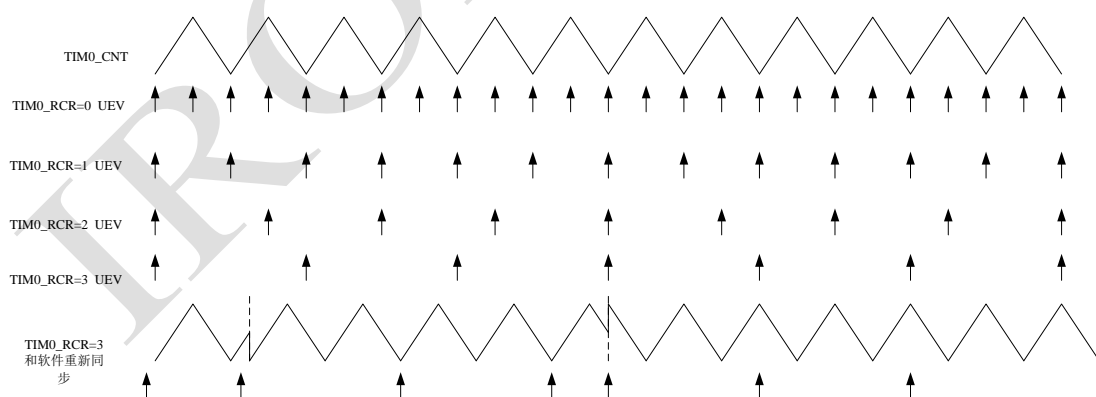


Figure 35 中央对齐计数模式下，重复计数次数分别为 0、1、2、3 时更新事件的产生

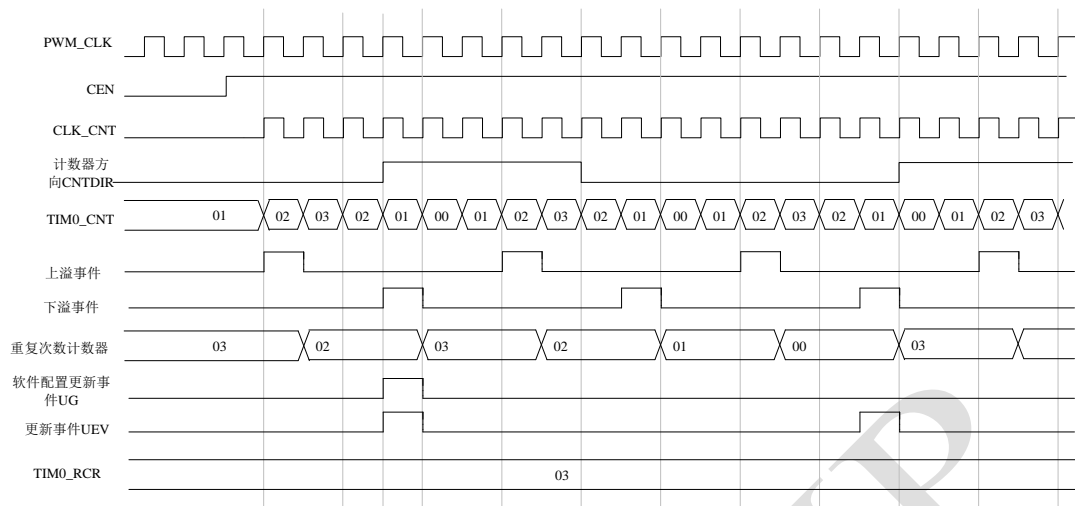


Figure 36 中央对齐模式下，更新事件的产生时序图

10.5.1.6 计数器时钟选择

计数器的时钟可由下列时钟源提供：

- 内部时钟。
- 外部输入脚 PWM0_CHx (x=1、2) 提供的输入信号（外部时钟源模式 1）。
- 外部触发输入 PWM0_ETR 提供的输入信号（外部时钟模式 2）。

10.5.1.6.1 内部时钟

内部时钟由 CRM 模块提供，最高频率为 73.728MHz，可以通过配置 CRM 模块中的 PWM_CLK_CTRL 寄存器对时钟进行 1/2/4/8 分频，预分频器可以对输入的 pwm0_clk 再进行 1~65536 分频，可以通过配置寄存器 TIM0_PSCL 和 TIM0_PSCH 实现。

将 TIM0_CONR2 寄存器的 SMS 位配置为 000，则从模式控制器被关闭，预分频器的驱动时钟由内部时钟 pwm0_clk 提供。计数器的使能、方向和软更新事件由软件配置确定。当 TIM0_CNTEN 寄存器的 CNTEN 位配置为 1 时，计数器开始计数。

10.5.1.6.2 外部时钟源模式 1

当寄存器 TIM0_TGICR0 的 SMS 位配置为 111 时，外部时钟源模式 1 被选中，计数器可在选定的输入脚 PWM0_CHx 输入信号的每个上升沿或下降沿计数，通过配置 TIM0_TGICR0 寄存器的 TRGS 位可以选择输入管脚 PWM0_CHx 的信号。Figure 37 为选择输入脚 PWM0_CH0 的信号作为外部时钟的例子，具体配置步骤如下：

- 外部信号输入后，通过配置寄存器 TIM0_CC0MR 中的 CC0MS 位可以从 CC0 的输入信号 CH0IN 和通道 CC1 的输入信号 CH1IN 中选择一个作为通道 CC0 的输入信号，本例中选择 CH0INF 作为 CC0 的输入信号，本例中选择 CH0IN。
- 将信号 CH0IN 送入边沿检测器做边沿检测，得到上升沿信号 CH0INF_R、下降沿信号 CH0INF_D 和边沿信号 CH0INF_ED（包括上升沿信号和下降沿信号）。

- 通过配置寄存器 TIM0_CCPS 寄存器的 CC0P 位可以选择输入信号的有效沿，当 CC0P 配置为 0 时，选择信号的上升沿 CH0INF_R 作为有效边沿；当 CC0P 配置为 1 时，选择信号的下降沿 CH0INF_D 作为有效边沿，得到边沿选择后的输入信号 CH0INFP，本例中选择上升沿为有效边沿。
- 配置寄存器 TIM0_TGICR0 的 TRGS 位为 101，选择 CH0INFP 作为从模式控制器的输入触发信号。
- 配置寄存器 TIM0_TGICR0 的 SMS 位为 111，使从模式控制器工作在外部时钟模式 1 下。
- 配置寄存器 TIM0_CNTEN 的 CNTEN 位为 1，启动计时器。

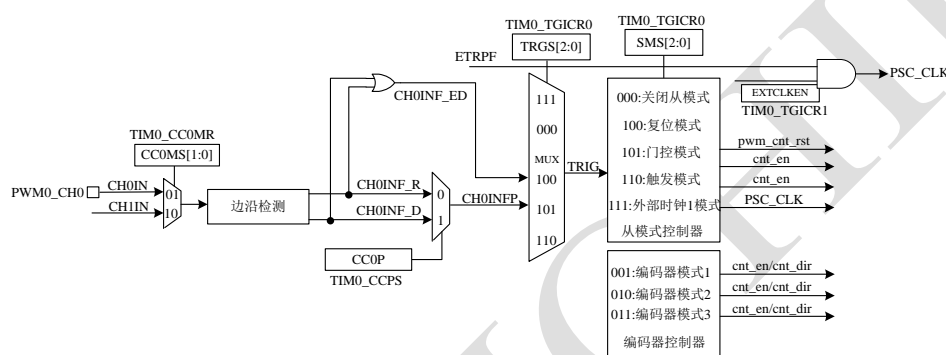


Figure 37 输入脚 PWM0_CH0 的信号作为外部时钟的连接例子

Figure 38 为输入信号 CH0IN 的上升沿触发计数器计数的时序图，当输入信号 CH0IN 出现一次上升沿时，计数器记一次数。输入信号 CH0IN 的上升沿和实际的计数器时钟之间的延时取决于 CH0IN 输入通道的重同步电路。

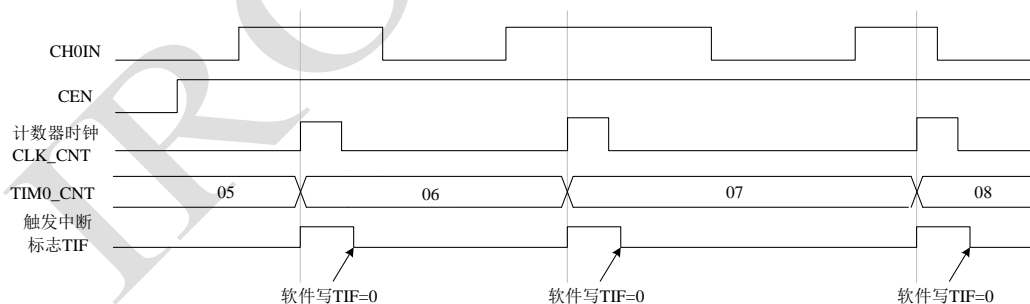


Figure 38 输入脚 PWM0_CH0 的信号作为外部时钟的时序图

10.5.1.6.3 外部时钟源模式 2

将寄存器 TIM0_TGICR1 的 EXTCLKEN 位配置为 1，即采用此种模式，计数器可以由外部管脚 PWM0_ETR 的每个上升沿或下降沿驱动计数。Figure 39 为外部触发信号 PWM0_ETR 作为外部时钟源的例子，具体配置为：

- 配置寄存器 TIM0_TGICR1 的 EXTP 位，选择输入信号的极性。若 EXTP 位配置为

0，不对输入的信号反相，输入信号的上升沿或高电平有效；若 EXTP 位配置为 1，则对输入的信号反相，输入信号的下降沿或低电平有效，极性选择后得到信号 PWM0_ETRP。

- 配置寄存器 TIM0_TGICR1 的 EXTDIV 位，对 PWM0_ETRP 进行分频，得到信号 PWM0_ETRPDIV。
- 配置寄存器 TIM0_TGICR1 的 EXTCLKEN 位为 1，将计数器时钟配置为外部时钟源模式 2。

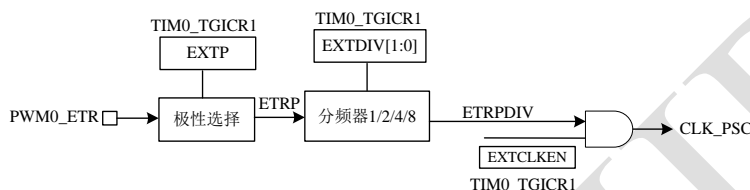


Figure 39 外部触发信号作为外部时钟源

值得注意的是，将计数器的时钟配置为外部时钟模式 2 所达到的效果与将时钟配置为外部时钟模式 1 同时把外部输入信号 PWM0_ETR 作为信号源效果是一样的，都是在 PWM0_ETR 的有效边沿触发计数器计数。

外部时钟模式 2 可以与各种从模式配合使用，例如，当时钟配置为外部时钟模式 2 时，从模式可以配置为门控模式。外部时钟模式 1 则不可以和从模式配合使用。Figure 40 为外部时钟模式 2 下计数器的时序图，外部信号 PWM0_ETR 出现一次上升沿，则计数器计数一次。PWM0_ETR 信号的上升沿和实际的时钟上升沿的延时取决于 PWM0_ETR 输入通道的重同步电路。

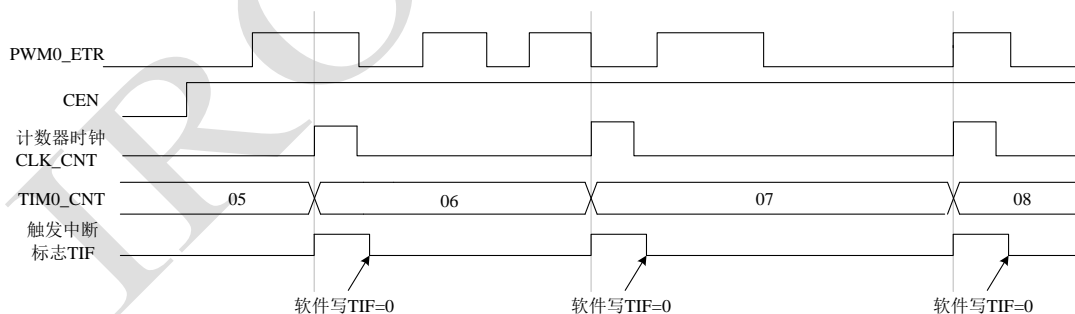


Figure 40 外部时钟模式 2 下计数器的时序图

10.5.1.7 寄存器更新及安全级别设置

10.5.1.7.1 16 位寄存器的读写

本定时器中有些寄存器是 16 位寄存器，而芯片的数据总线是 8 位，为了便于寄存器读写，将 16 位寄存器拆分为两个 8 位寄存器，即高字节寄存器 MS 和低字节寄存器 LS，例如

通道 CCx 的捕获比较寄存器 TIM0_CCxR 分为高字节寄存器 TIM0_CCxRH 和低字节寄存器 TIM0_CCxRL。在本定时器中读写 16 位寄存器应遵循以下规则：先读写高字节寄存器 MS 再读写低字节寄存器 LS。

10.5.1.7.2 寄存器的预装载模式和立即更新模式

定时器中有些寄存器有两种更新模式：预装载模式和立即更新模式。

立即更新模式指：当向寄存器写入新值时，新值立即生效。

预装载模式指：具有预装载功能的寄存器有两个寄存器--预装载寄存器和影子寄存器，TIM0 使用的是影子寄存器值，而软件只能读写预装载寄存器。

若开启了预装载功能，当软件配置寄存器时，只能将新值写入预装载寄存器，当发生寄存器的更新事件时，硬件才将预装载寄存器中的值装载到影子寄存器，此时新配置的值才会生效。

TIM0 中所有具有预装载功能的寄存器有：TIM0_ARR、TIM0_PSC（无立即更新模式）、TIM0_CCxR、TIM0_CCENR、TIM0_CCPS 和寄存器 TIM0_CCxMR 的 OCxMS 位，根据预装载模式的使能信号和更新信号的不同，可以将这些寄存器分为三类：

- 由寄存器 TIM0_CONR2 的 ARPLE 位控制预装载模式是否使能，由更新事件 UEV 更新寄存器，这些寄存器有：TIM0_ARR、TIM0_PSC（无立即更新模式）。
- 由 TIM0_CCxMR 寄存器的 OCxPEN 位控制预装载模式是否使能，由更新事件 UEV 更新寄存器，这些寄存器有：TIM0_CCxRL、TIM0_CCxRH。
- 由 TIM0_CONR0 寄存器的 CCPE 位控制预装载模式是否使能，在预装载模式下由 COM 事件或从模式的触发输入更新（由 TIM0_CONR0 寄存器的 CCUS 位决定）寄存器，这些寄存器有：TIM0_CCENR、TIM0_CCPS 和 TIM0_CCxMR 寄存器的 OCxMS 位。

具有预装载功能寄存器的预装载功能可以关闭（TIM0_PSC 除外），当预装载功能关闭时，具有预装载功能的寄存器使用立即更新模式。

10.5.1.7.3 寄存器的安全级别控制

为了防止寄存器被误配，高级定时器为寄存器设置了配置保护，可以通过配置寄存器 TIM0_BRKC 的 LOCK 位设置对不同寄存器的保护，复位后 LOCK 位为 00，关闭对寄存器的保护，所有寄存器均可配置，LOCK 位只能配置一次非 0 值，一旦配置为非 0 值后不可以再更改直到高级定时器模块被复位。

寄存器 TIM0_BRKC.LOCK 对寄存器的保护分为三个级别：

- 当寄存器 TIM0_BRKC.LOCK 配置为 01 时，将寄存器的安全级别配置为锁定级别 1，不能写入 TIM0_DTG 寄存器、TIM0_BRKC 的 BRKE、BRKP、AOE 位、TIM0_CONR2 寄存器。
- 当寄存器 TIM0_BRKC.LOCK 配置为 10 时，将寄存器的安全级别配置为锁定级别

2, 不能写入锁定级别 1 中的所有寄存器, 也不能写入 TIM0_CCPS 寄存器, 不能写入 TIM0_BRKC 的 ROSS 和 IOSS 位。

- 当寄存器 TIM0_BRKC.LOCK 配置为 11 时，将寄存器的安全级别配置为锁定级别 3，不能写入锁定级别 1 和锁定级别 2 中的所有寄存器，不能写入 TIM0_CCxMR 的 OCxMS 和 OCxPEN 位。

10.5.2 输入捕获功能

高级定时器包含四个通道，可以分别对四个通道独立控制，每个通道可以配置为输入捕获模式或输出比较模式，下面以通道 CC0 为例介绍当通道配置为输入捕获功能时的使用方法。

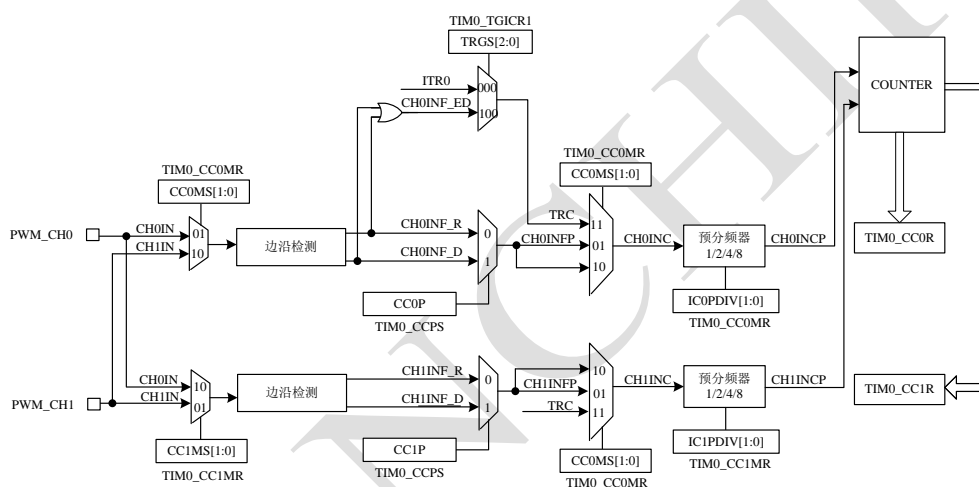


Figure 41 输入捕获通道

如 Figure 41 所示，当寄存器 TIM0_CC0MR 的 CC0MS 位配置为 01/10/11 时，通道 CC0 配置为输入捕获功能，外部输入信号的有效边沿可以捕获计数器的值，并将捕获的值存入通道 CC0 的输入捕获寄存器 TIM0_CC0R 中，具体控制方法如下：

- 配置 GPIO 模块中的寄存器,将相应的管脚配置为高级定时器的信号输入输出功能。
- 通过配置寄存器 TIM0_CC0MR 中的 CC0MS 位可以选 CH0IN 或通道 CC1 的输入信号 CH1IN 作为通道 CC0 的输入信号。
- 将滤波后的信号送入边沿检测器做边沿检测,得到上升沿信号 CH0INF_R、下降沿信号 CH0INF_D 和边沿信号 CH0INF_ED (包括上升沿信号和下降沿信号)。
- 将边沿信号 CH0INF_ED 送入从模式控制器,通过配置寄存器 TIM_TGICR0 的 TRGI 位可以从 CH0INF_ED 和 ITR0 中选择一个信号作为触发捕获信号 TRC。通过配置寄存器 TIM_CCPS 的 CC0P 位可以选择输入信号的有效沿,当 CC0P 配置为 0 时,选择信号的上升沿 CH0INF_R 作为有效边沿;当 CC0P 配置为 1 时,选择信号的下降沿 CH0INF_D 作为有效边沿,得到边沿选择后的输入信号 CH0INFP。

- 通过配置寄存器 TIM0_CC0MR 中的 CC0MS 位可以从 TRC 信号和 CH0INFP 中选择一个作为通道 CC0 原始的捕获信号 CH0INC。
- 配置寄存器 TIM0_CC0MR 的 ICPDIV 位,对原始的捕获信号 CH0INC 进行预分频,得到输入捕获信号 CH0INCP。
- 将输入捕获信号 CH0INCP 送入计数器部分进行数据捕获,捕获后将数据存入输入捕获寄存器 TIM0_CC0RU。

以上为 CC0 通道的输入捕获模式的控制步骤,CC1、CC2 和 CC3 通道的控制步骤相同。在软件访问 16 位寄存器时,先访问高 8 位部分再访问低 8 位部分,按照此规则,在软件读取 TIM0_CC0R 中的捕获值时,先读取寄存器 TIM0_CC0RH 再读取 TIM0_CC0RL, Figure 42 为软件读取捕获寄存器 TIM0_CC0R 的过程。

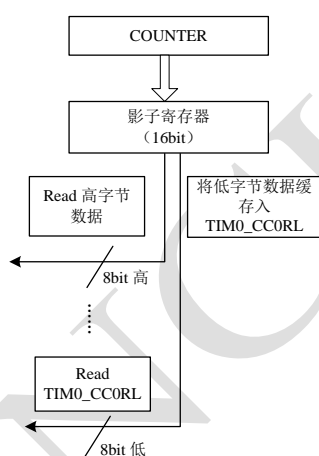


Figure 42 向软件读取捕获寄存器

为了防止在读取捕获值的高 8 位部分时再次发生捕获,软件一旦开始读取高 8 位部分数据时,硬件将低 8 位数据从影子寄存器缓存入预装载寄存器 TIM0_CC0RL。

当产生捕获信号 CH0INCP 时,寄存器 TIM0_ISR0 中捕获中断标志位 CC0IF 置位,若捕获/比较中断使能打开,则产生捕获中断。读取捕获值后捕获中断标志位 CC0IF 自动清除,也可以通过软件配 0 将捕获中断标志位 CC0IF 清除。

若捕获的数据没有及时读取,此时又有捕获信号来到,则寄存器 TIM0_ISR1 中的过捕获标志位 CC0OIF 置位,新捕获的数据将原来的数据覆盖。需要软件配置 CC0OIF 位为 0 清除过捕获标志 CC0OIF。

10.5.3 从模式控制

10.5.3.1 从模式控制器的信号源

从模式是指高级定时器受外部触发信号驱动的工作模式, Figure 43 所示为从模式控制器,外部触发信号的来源有:

- 外部信号输入通道 PWM0_ETR，触发信号产生过程为：
 - 配置寄存器 TIM0_TGICR1 的 EXTP 位，选择输入信号的极性，若 EXTP 位配置为 0，不对输入的信号反相，输入信号的上升沿或高电平有效；若 EXTP 位配置为 1，则对输入的信号反相，输入信号的下降沿或低电平有效，极性选择后得到信号 EXTRP。
 - 配置寄存器 TIM0_TGICR1 的 EXTDIV 位，对 EXTRP 进行分频，得到信号 PWM0_ETRPDIV。
 - PWM0_ETRPDIV 信号有两个用途：i) 作为外部触发信号，进入从模式控制器驱动高级定时器工作。ii) 当寄存器 TIM0_TGICR1 的 EXTCLKEN 位配置为 1 时，外部时钟模式 2 打开，PWM0_ETRPF 也可作为计数器预分频器的时钟 PSC_CLK。
- 通道 CC0 的双边沿信号 CH0IN_ED，产生过程见 10.5.2 部分说明。
- 通道 CC0 的单边沿信号 CH0INFP，产生过程见 10.5.2 部分说明，此信号一方面送入从模式控制器驱动高级定时器工作，另一方面作为编码器控制器的输入，在编码器接口模式下控制计数器计数。
- 通道 CC1 的单边沿信号 CH1INFP，产生过程见 10.5.2 部分说明，此信号一方面送入从模式控制器驱动高级定时器工作，另一方面作为编码器控制器的输入，在编码器接口模式下控制计数器工作。

其中，TIM0_TRGO 和 CH0IN_ED 信号还可以作为捕获信号。

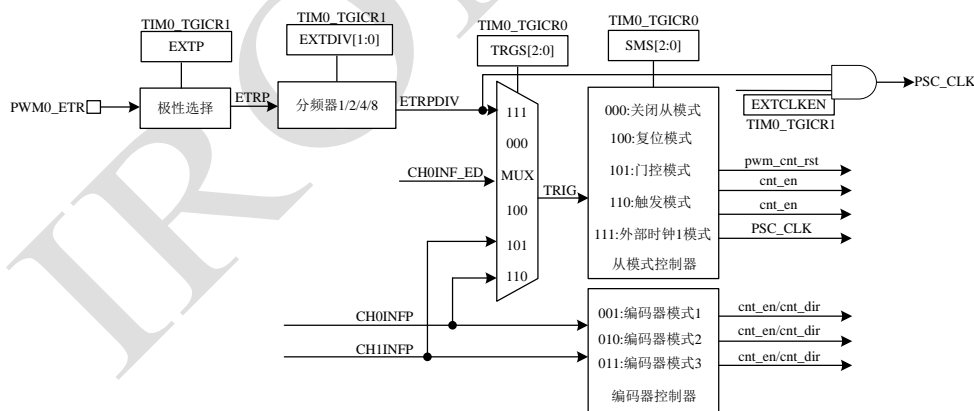


Figure 43 从模式控制器示意图

10.5.3.2 从模式控制

配置寄存器 TIM0_TGICR0 的 TRGS 位，从 ETRPF、CH0INF_ED、CH0INFP 和 CH1INFP 中选择一个作为触发信号送入从模式控制器，在不同的控制模式下驱动高级定时器工作。对高级定时器的控制模式有：

- 关闭从模式：将寄存器 TIM0_TGICR0 的 SMS 位配置为 000，关闭从模式，此时预

分频器的驱动时钟由内部时钟 `pwm0_clk` 提供。计数器的使能、方向和软更新事件由软件配置确定。当 `TIM0_CNTEN` 寄存器的 `CNTEN` 位配置为 1 时，计数器开始计数。

- 复位模式：将寄存器 `TIM0_TGICR0` 的 `SMS` 位配置为 100 时，从模式控制器工作在复位模式下，选中的触发信号 `TRGI` 的有效沿重新初始化计数器，并且产生一个更新事件来更新寄存器。若 `TIM0_CONR2` 的 `URC` 位配置为 0，则寄存器 `TIM0_ISR0` 的中断标志 `SOFTUIF` 置位，若软更新中断使能打开，则产生一个软更新中断，此时，相当于配置了寄存器 `TIM0_EGR` 的 `UG` 事件；若 `TIM0_CONR2` 的 `URC` 位配置为 1，则在触发信号 `TRGI` 的有效沿仅初始化计数器，并且产生一个更新事件来更新寄存器，而不产生中断和中断标志。
- 门控模式：将寄存器 `TIM0_TGICR0` 的 `SMS` 位配置为 101 时，从模式控制器工作在门控模式下，高级定时器的计数器使能受触发信号 `TRGI` 的控制，当 `TRGI` 为有效电平时，计数器开始计数，当 `TRGI` 为无效电平时，计数器停止计数（但不复位）。此时 `TIM0_CNTEN` 的 `CNTEN` 位仍需要配置为 1，否则，即使 `TRGI` 为有效电平，计数器也停止计数。
- 触发模式：将寄存器 `TIM0_TGICR0` 的 `SMS` 位配置为 110 时，从模式控制器工作在触发模式下，选中的触发信号 `TRGI` 的有效沿启动计数器计数（但不复位）。此时 `TIM0_CNTEN` 的 `CNTEN` 位不需要配置为 1，触发信号 `TRGI` 的有效沿会将 `CNTEN` 位置 1。计数器启动后配置 `TIM0_CNTEN` 的 `CNTEN` 位为 0，可以停止计数器计数。
- 外部时钟源模式 1：将寄存器 `TIM0_TGICR0` 的 `SMS` 位配置为 111 时，选择触发信号 `TRGI` 的有效沿驱动计数器计数。此时 `TIM0_CNTEN` 的 `CNTEN` 位需要配置为 1。
- 外部时钟源模式 2：配置寄存器 `TIM0_TGICR1` 的 `EXTCLKEN` 位为 1，`PWM0_ETR` 的有效沿驱动计数器计数。此时 `TIM0_CNTEN` 的 `CNTEN` 位需要配置为 1。

有三点需要注意：

第一，在 `CH0IN_ED` 被选为触发信号 `TRGI` 时，不要使用门控模式，因为 `CH0IN_ED` 是一个脉冲信号，而门控模式时要检查触发输入的电平。

第二，从模式控制器配置为外部时钟源模式 1 且选 `PWM0_ETR` 作为触发信号 `TRGI` 与直接将定时器配置为外部时钟源模式 2 效果是一样的，都是使用 `PWM0_ETR` 的有效沿驱动计数器计数。

第三，外部时钟源模式 2 和各种从模式可以配合使用，当定时器配置为外部时钟源模式 2 时，从模式控制器也可以配置为各种从模式，此时从模式控制器的输入触发信号 `TRGI` 不要选择 `PWM0_ETR`。

10.5.3.3 编码器接口模式

使用通道 CC0 和 CC1 作为正交增量编码器两路信号的接口，在通道 CC0 和 CC1 对信号做必要的处理后得到信号 CH0INFP 和 CH1INFP，再将 CH0INFP 和 CH1INFP 直接送入编码器控制器，在编码器控制器中根据两路信号驱动计数器计数。

编码器接口模式的配置步骤为：

- 根据实际需要配置寄存器 TIM0_CCPS 的 CC0P 位，对输入的编码器信号进行极性选择，得到信号 CH0INFP。
- 根据实际需要配置寄存器 TIM0_CCPS 的 CC1P 位，对输入的编码器信号进行极性选择，得到信号 CH1INFP。
- 配置寄存器 TIM0_TGICR0 的 SMS 位：若将 SMS 位配置为 001，定时器工作在编码器模式 1 下，计数器以 CH0INFP 作为相对信号由 CH1INFP 的边沿驱动计数；若将 SMS 位配置为 010，定时器工作在编码器模式 2 下，计数器以 CH1INFP 作为相对信号由 CH0INFP 的边沿驱动计数；若将 SMS 位配置为 011，定时器工作在编码器模式 3 下，计数器以 CH1INFP 和 CH0INFP 作为相对信号由 CH0INFP 和 CH1INFP 的边沿驱动计数。计数器的计数方向也在编码器控制器中根据一定的规则产生。
- 配置寄存器 TIM0_CCENR 的 CC0EN 和 CC1EN 位为 1，使能通道 CC0 和通道 CC1。
- 将定时器配置为编码器模式后，需要将 TIM0_CTENR 寄存器的 CTEN 位置 1 来启动计数器。

假设计数器已经使能，则计数器由 CH0INFP 和 CH1INFP 的有效跳变沿驱动。编码器控制器根据两个输入信号的跳变顺序，产生计数器的计数脉冲和计数方向，使计数器向上或向下计数，无论计数器是由 CH0INFP 驱动计数还是由 CH1INFP 驱动计数或者由 CH0INFP 和 CH1INFP 的边沿同时驱动，任一输入端跳变都会重新计算计数器的方向。计数器的计数方向产生规则如 Table 101 所示，表中相对信号的含义为：若计数器由 CH0INFP 的有效跳变沿驱动计数时，相对信号为 CH1INFP；若计数器由 CH1INFP 的有效跳变沿驱动计数时，相对信号为 CH0INFP；若计数器由 CH0INFP 和 CH1INFP 的有效跳变沿同时驱动计数时，互为相对信号。

Table 101 增量编码器接口模式下计数器计数规则

有效边沿	相对信号的 电平	CH0INFP 信号		CH1INFP 信号	
		上升沿	下降沿	上升沿	下降沿
仅在 CH0INFP 边 沿计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在 CH1INFP 边	高	不计数	不计数	向上计数	向下计数

有效边沿	相对信号的 电平	CH0INFP 信号		CH1INFP 信号	
		上升沿	下降沿	上升沿	下降沿
沿计数	低	不计数	不计数	向下计数	向上计数
在 CH0INFP 和 CH1INFP 边沿计 数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

编码器接口模式相当于使用了一个带有方向选择的外部时钟，这意味着计数器只有在 0 到 TIM0_ARR 寄存器的自动装载值之间连续计数（根据 Table 101 增量编码器接口模式下计数器计数规则中的方向，或者是 0 到 TIM0_ARR 计数，或者是 TIM0_ARR 到 0 计数），所以在开始计数之前必须配置 TIM0_ARR。

在这种模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置，计数器方向与相连的传感器旋转的方向对应。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发一个计数器复位。Figure 44 为在增量编码器模式 3 下对输入信号 CH0INFP 和 CH1INFP 的编码。

例子：在增量编码器模式下，计数器在输入信号 CH0INFP 和 CH1INFP 的驱动下计数器，计数过程如 Figure 44 所示。

- 配置 GPIO 模块寄存器，将相应的引脚复用为高级定时器的输入管脚 PWM0_CH0 和 PWM0_CH1，并且将增量编码器的输出信号连接至 PWM0_CH0 和 PWM0_CH1，这样增量编码器的信号由定时器的通道 CC0 和 CC1 输入。
- 配置寄存器 TIM0_CCENR 的 CC0EN 和 CC1EN 位为 1，使能通道 CC0 和 CC1 输入。
- 配置寄存器 TIM0_CCPS 的 CC0P 和 CC1P 位为 0，通道 CC0 和 CC1 输入信号不反相。
- 配置寄存器 TIM0_TGICR0 的 SMS 位配置为 011，将定时器从模式配置为编码器模式 3。
- 配置寄存器 TIM0_CNTEN 的 CNTEN 位为 1，使能计数器。

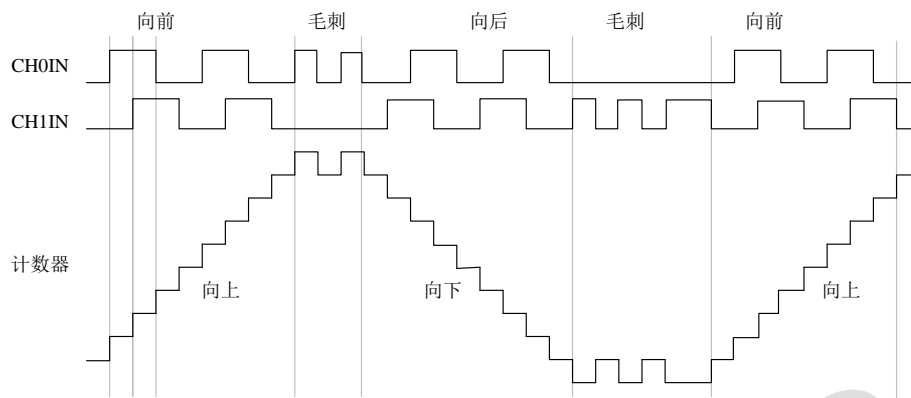


Figure 44 高级定时器作为增量编码接口的例子

10.5.3.4 PWM 输入模式

PWM 输入模式是输入捕获模式的一个特例,可以测量一个 PWM 信号的占空比和周期,假设 PWM 信号由通道 CC0 输入,具体配置如下:

- 将 TIM0_ARR 配置为最大值。
- 配置寄存器 TIM0_CCPS 的 CC0P 位为 0, CC0 通道使用信号的上升沿捕获数据。
- 配置寄存器 TIM0_CC0MR 的 CC0MS 位为 01, 选择 CH0INF 的上升沿捕获数据。
- 不对捕获信号分频, 配置寄存器 TIM0_CC0MR 的 IC0PDIV 位为 00。
- 配置寄存器 TIM0_CCPS 的 CC1PS 位为 1, 配置寄存器 TIM0_CC0MR 的 CC0MS 位为 10, 选择通道 CC0 信号 CH0INF 的下降沿作为通道 CC1 的捕获信号。
- 配置寄存器 TIM0_TGICR0 的 TRGS 位为 101, 选择 CH0INFP 作为从模式的触发信号 TRGI。
- 配置寄存器 TIM0_TGICR0 的 SMS 位为 100, 将从模式配置为复位模式, CH0INFP 的上升沿将计数器清零。
- 配置寄存器 TIM0_CCENR 的 CC0EN 位和 CC1EN 位为 1, 使能通道 CC0 和通道 CC1 的捕获使能。
- 配置寄存器 TIM0_EGR 的 UG 位和 COMG 位为 1, 初始化计数器并更新相应寄存器。
- 配置寄存器 TIM0_CNTEN 的 CNTEN 位为 1, 启动计数器。

因为只有 CH0INFP 和 CH1INFP 连接到了从模式控制器,所以 PWM 输入模式只能使用 CC0 和 CC1 通道。PWM 输入模式捕获时序如 Figure 45 所示。

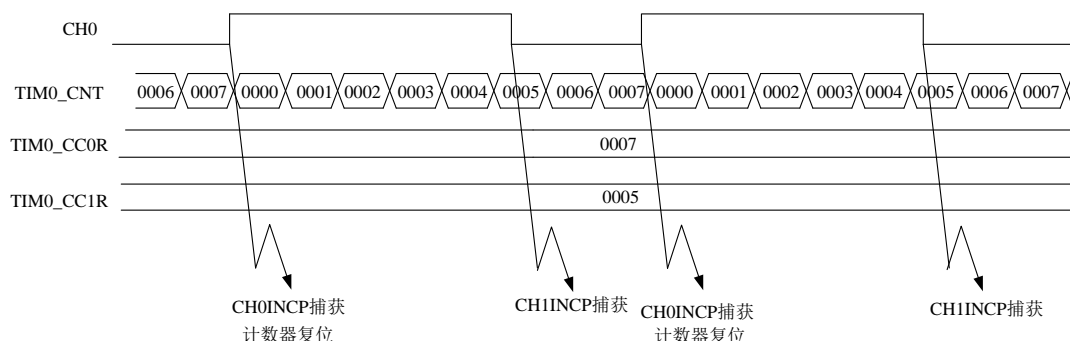


Figure 45 PWM 输入模式时序图

10.5.4 输出比较功能

将寄存器 TIM0_CCxMR 的 CCxMS 位配置为 00，通道 CCx 工作在输出比较模式下。在输出比较模式下，四个通道是相互独立的，可以独立配置各自的工作模式。通道 CC0、CC1 和 CC2 配置为比较输出时可以输出两路互补的 PWM 波形，输出控制过程如 Figure 46 所示，通道 CC3 只能输出一路 PWM 波形，输出控制过程如 Figure 47 所示。

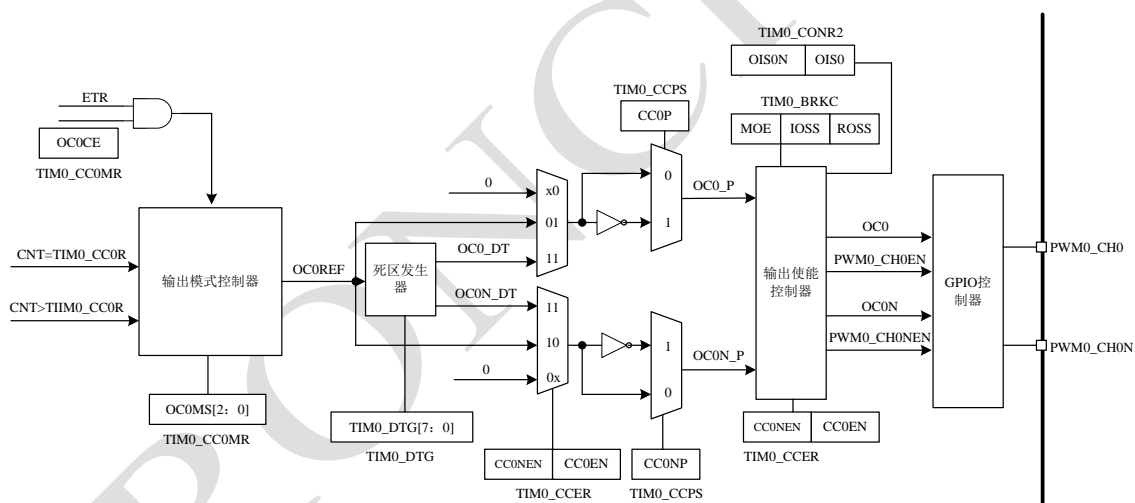


Figure 46 通道 CC0 输出控制

以通道 CC0 为例说明通道 CC0、CC1、CC2 的输出控制过程如下：

- 配置寄存器 TIM0_CC0MR 的 CC0MS 位为 00，将 CC0 通道配置为输出功能。
- 配置寄存器 TIM0_CC0MR 的 OC0MS 位域，选择输出模式，可选的输出模式有：冻结模式、匹配时输出有效电平、匹配时输出无效电平、强制输出有效电平、强制输出无效电平、翻转、PWM 模式 1、PWM 模式 2，各种模式的详细描述见 10.5.4.1 节，输出基准参考信号 OC0REF，OC0REF 高电平为有效电平、低电平为无效电平。当寄存器 TIM0_CC0MR 的 OC0CE 位配为 1 时，外部信号 PWM0_ETR 的有效电平清除 OC0REF。
- 若输出模式配置为 PWM 模式 1 或 PWM 模式 2 且主路输出和互补输出的使能均有

效 (CC0ENN=1, CC0NEN=1), 则根据寄存器 TIM0_DTG 的配置插入死区, 输出两路互补信号 OC0_DT 和 OC0N_DT。若两路使能不同时有效, 则不插入死区, 直接在使能打开的一路输出 OC0REF, 使能关闭的一路输出低电平。

- 对输出信号进行极性控制, 配置寄存器 TIM0_CCPS 的 CC0P 位为 0 时, 主路输出高电平为有效电平, 当 CC0P 位为 1 时, 主路输出低电平为有效电平, 极性控制后得到输出信号 OC0_P; 配置寄存器 TIM0_CCPS 的 CC0NP 位为 0 时, 互补路输出高电平为有效电平, 当 CC0NP 位为 1 时, 互补路输出低电平为有效电平, 极性控制后得到输出信号 OC0N_P。
- 根据寄存器 TIM0_BRKC 中 MOE 位、IOSS、ROSS 位、寄存器 TIM0_CCER 中的 CC0EN、CC0NEN、寄存器 TIM0_CONR1 中的 OIS0 和 OIS0N 位的配置控制主路输出和互补路的输出, 并控制主路和互补路的输出使能, 详细控制方法见 10.5.4.4 节所述。最终从高级定时器中输出信号 OC0 和 OC0N 以及两路的输出使能信号 PWM0_CH0EN 和 PWM0_CH0NEN 至 GPIO 模块, 在 GPIO 模块中选择 PWM 输出功能将 OC0 和 OC0N 输出到相应的管脚。
- 若计数器与输出比较寄存器 TIM0_CC0R 比较成功, 则寄存器 TIM0_ISR0 的中断标志位 CC0IF 置位。若设置了通道 CC0 的输入输出中断使能, 则产生通道 CC0 的输出比较中断。

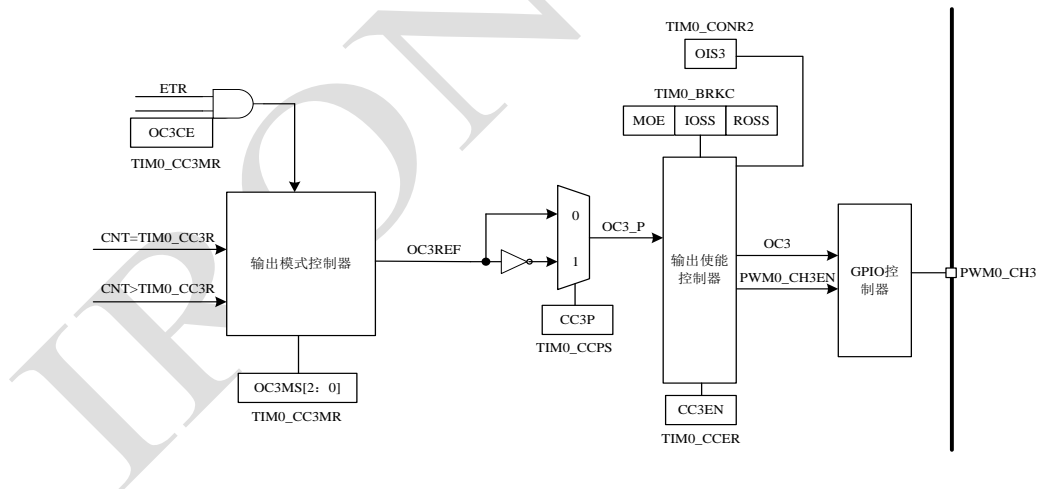


Figure 47 通道 CC3 输出控制

通道 CC3 的输出与通道 CC0-CC2 不同, 通道 CC3 只有主路输出 OC3, 没有互补路输出, 输出控制如 Figure 47 所示, 通道 CC3 的输出控制过程为:

- 配置寄存器 TIM0_CC3MR 的 CC3MS 位为 00, 将 CC3 通道配置为输出功能。
- 配置寄存器 TIM0_CC3MR 的 OC3MS 位, 选择输出模式, 可选的输出模式有: 冻结模式、匹配时输出有效电平、匹配时输出无效电平、强制为有效电平、强制为无效电平、翻转、PWM 模式 1、PWM 模式 2, 各种模式的详细描述见 10.5.4.1 节,

输出基准参考信号 OC3REF, OC3REF 高电平为有效电平、低电平为无效电平。当寄存器 TIM0_CC3MR 的 OC3CE 位配置为 1 时, 外部信号 PWM0_ETR 的有效电平清除 OC3REF。

- 对输出信号进行极性控制, 配置寄存器 TIM0_CCPS 的 CC3P 位为 0 时, 主路输出高电平为有效电平, 当 CC3P 位为 1 时, 主路输出低电平为有效电平, 极性控制后得到输出信号 OC3_P。
- 根据寄存器 TIM0_BRKC 中 MOE 位、IOSS、ROSS 位、寄存器 TIM0_CCER 中的 CC3EN、寄存器 TIM0_CONR1 中的 OIS3 位的配置控制主路输出, 并控制主路输出使能, 详细控制方法见 10.5.4.4 节所述。最终从高级定时器中输出信号 OC3 以及输出使能信号 PWM0_CH3EN, 将 OC3 和 PWM0_CH3EN 送入 GPIO 控制模块, 在 GPIO 模块做相应的控制后输出到相应的管脚。
- 若计数器与输出比较寄存器 TIM0_CC3R 比较成功, 则寄存器 TIM0_ISR0 的中断标志位 CC3IF 置位。若设置了通道 CC3 的输入输出中断使能, 则产生通道 CC3 的输出比较中断。

10.5.4.1 输出模式控制

10.5.4.1.1 强置输出模式

在输出模式下(寄存器 TIM0_CCxMR 的 CCxMS 位配置为 00), 输出比较信号(OCxREF 和相应的 OCx/OCxN)能够直接由软件强置为有效或无效状态, 而不依赖于输出比较寄存器和计数器的比较结果。

将寄存器 TIM0_CCxMR 的 OCxMS 位配置为 101, 可将输出比较信号(OCxREF/OCx)强置为有效状态, 这样 OCxREF 被强置为高电平(OCxREF 始终为高电平有效), 同时 OCx 得到与寄存器 TIM0_CCPS 的 CCxP 位相反的信号。例如, CCxP 位配置为 1 (将 OCx 配置为低电平有效), 则 OCx 被强置为低电平(与 CCxP 值相反)。

将寄存器 TIM0_CCxMR 寄存器的 OCxMS 位配置为 100, 可将输出比较信号(OCxREF/OCx)强置为无效状态, 这样 OCxREF 被强置为低电平(OCxREF 始终为高电平有效), 同时 OCx 得到与寄存器 TIM0_CCPS 的 CCxP 位相同的信号, 例如, CCxP 配置为 1 (将 OCx 配置为低电平有效), 则 OCx 被强置为高电平(与 OCxP 的值相同)。

在强置输出模式下, 比较寄存器 TIM0_CCxR 和计数器之间的比较仍在进行, 相应的标志也会被修改。因此仍然会产生相应的中断标志和中断请求。

Figure 48 为当高级定时器配置为强置输出模式时的时序图。

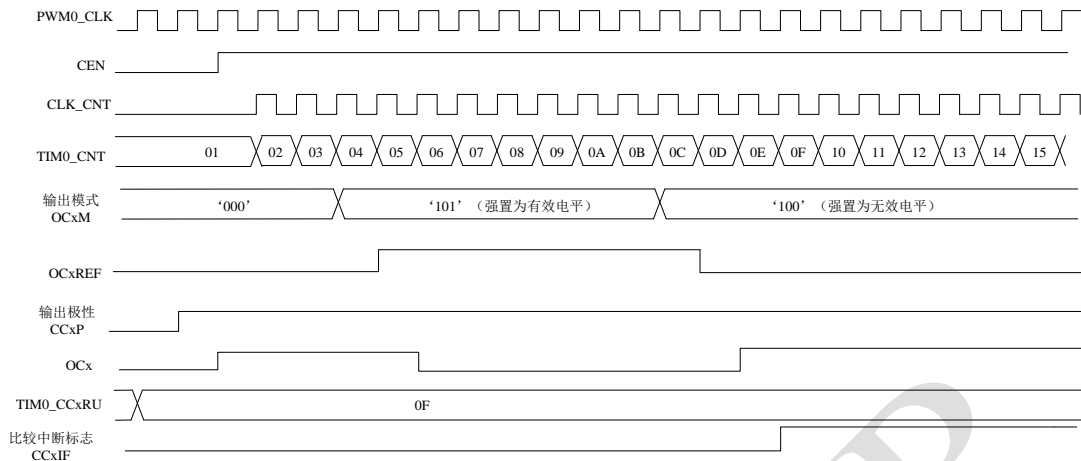


Figure 48 高级定时器强置输出模式时序图

10.5.4.1.2 输出比较模式

当计数器与比较寄存器的内容相同时，输出比较功能做如下操作：

- 定义基准参考波形 OCxREF 和定时器输出 OCx 的变化，当计数器与比较寄存器匹配时 ($TIM0_CNT = TIM0_CCxR$):
 - 若寄存器 TIM0_CCxMR 的 OCxMS 位配置为 000, 则 OCxREF 保持不变, OCx 保持不变。
 - 若寄存器 TIM0_CCxMR 的 OCxMS 位配置为 001, 则 OCxREF 被设置为高电平, 输出 OCx 被设置为有效电平, 即若 CCxP=0, 则 OCx 输出高电平, 若 CCxP=1, 则 OCx 输出低电平。
 - 若寄存器 TIM0_CCxMR 的 OCxMS 位配置为 010, 则 OCxREF 被设置为低电平, 输出 OCx 被设置为无效电平, 即若 CCxP=0, 则 OCx 输出低电平, 若 CCxP=1, 则 OCx 输出高电平。
 - 若寄存器 TIM0_CCxMR 的 OCxMS 位配置为 011, 则 OCxREF 波形翻转, 输出 OCx 的电平翻转。
- 设置中断状态寄存器 TIM0_ISR0 中的中断标志位 CCxIF。
- 若设置了相应的中断使能, 则产生一个比较中断。

在输出比较模式下, 更新事件 UEV 对 OCxREF 输出没有影响。

输出比较模式的配置步骤为:

- 选择计数器时钟和预分频器 (内部时钟, 外部时钟, 预分频器)。
- 配置寄存器 TIM0_CONR2 的 ARPLE 位和 TIM0_CCxMR 的 OCxPEN 位, 确定 TIM0_ARR 寄存器、TIM0_CCxR 寄存器的预装载功能是否打开。
- 配置自动装载寄存器 TIM0_ARR、捕获比较寄存器 TIM0_CCxR, 都是遵循先配置高 8 位再配置低 8 位的规则。

- 配置寄存器 TIM0_CCxMR 的 OCxMS 位，选择输出模式，配置 TIM0_CCPS 寄存器，选择输出极性。
- 配置寄存器 TIM0_CCENR，使能相应的输出通道。
- 若要产生一个输出比较中断请求，设置寄存器 TIM0_IER 的 CCxIE 位。
- 配置寄存器 TIM0_EGR 的 UG 位和 COMG 位为 1，更新具有预装载功能的寄存器。
- 设置 TIM0_CNTEN 寄存器的 CNTEN 位，启动计数器。

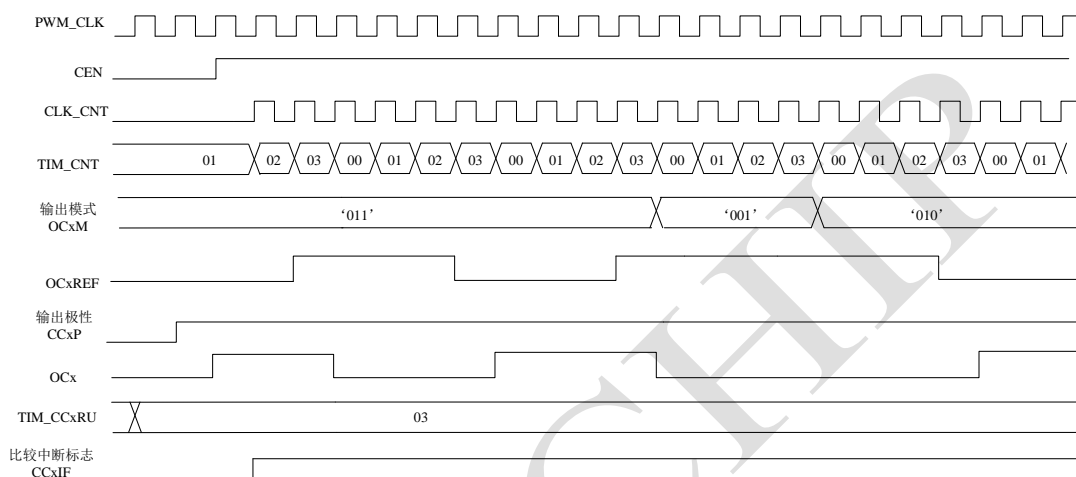


Figure 49 高级定时器比较输出模式

10.5.4.1.3 PWM 输出模式

脉冲宽度调制模式(PWM)可以产生一个由 TIM0_ARR 寄存器确定周期、由 TIM0_CCxR 寄存器确定占空比的信号。

配置 TIM0_CCxMR 寄存器中的 OCxMS 位为“110”，定时器输出 PWM 模式 1 的波形；配置 TIM0_CCxMR 寄存器中的 OCxMS 位为“111”，定时器输出 PWM 模式 2 的波形。

TIM0_CCMRx 寄存器 OCxPEN 位控制比较寄存器 TIM0_CCxR 的预装载使能。TIM0_CONR2 的 ARPLE 位控制自动重载寄存器 TIM0_ARR 的预装载使能。若使能了预装载功能，仅当发生一个更新事件时，预装载寄存器才能被传送到影子寄存器，因此在计数器计数之前，必须通过设置 TIM0_EGR 寄存器的 UG 位来更新具有预装载功能寄存器。

输出波形的极性可以通过软件在 TIM0_CCPS 中设置，可以设置高电平为有效电平或低电平为有效电平。高级定时器在管脚上的输出 PWM0_CHx 或 PWM0_CHxN 通过 CCxEN、CCxNEN、MOE、IOSS、ROSS、OISx 和 OISxN 位的组合控制，详见 10.5.4.4 节的描述。

PWM 模式分为 PWM 模式 1 和 PWM 模式 2，PWM 模式 1 波形产生的规则为：当 TIM0_CNT < TIM0_CCxR 时，PWM 基准参考信号 OCxREF 为有效电平（高电平），否则为无效电平（低电平）；PWM 模式 2 波形产生的规则为：当 TIM0_CNT < TIM0_CCxR 时，PWM 基准参考信号 OCxREF 为无效电平（低电平），否则为有效电平（高电平）；

下面以 PWM 模式 1 为例说明在不同的计数器计数模式下 PWM 输出控制。

● 向上计数模式下的 PWM 模式 1

当 TIM0_CONR2 寄存器中的 CNTMC 位配置为 00 且 CNTDIR 位配置为 0 时，计数器向上计数。此时，PWM 模式 1 的输出为：当 TIM0_CNT < TIM0_CCxR 时，PWM 基准参考信号 OCxREF 为有效电平（高电平），否则为无效电平（低电平）。如果 TIM0_CCxR 中的比较值大于自动重装载值，则 OCxREF 保持为有效电平（高电平）。如果比较值为 0，则 OCxREF 保持为无效电平（低电平）。Figure 50 为 TIM0_ARR=8 且计数器向上计数时 PWM 模式 1 的输出波形。

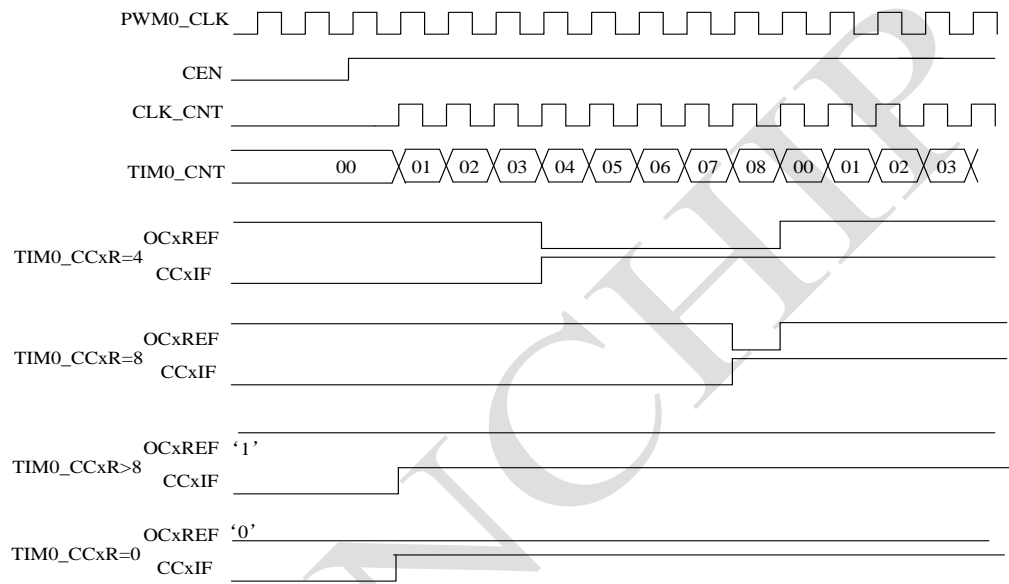


Figure 50 向上计数模式下，PWM 模式 1 输出

● 向下计数模式下的 PWM 模式 1

当 TIM0_CONR2 寄存器中的 CNTMC 位配置为 00 且 CNTDIR 位配置为 1 时，计数器向上计数。此时，PWM 模式 1 的输出为：当 TIM0_CNT > TIM0_CCxR 时，PWM 基准参考信号 OCxREF 为无效电平（低电平），否则为有效电平（高电平）。如果 TIM0_CCxR 中的比较值大于自动重装载值，则 OCxREF 保持为有效电平（高电平）。该模式下不能产生 0% 的 PWM 波形。Figure 51 为 TIM0_ARR=8 时向上计数器时 PWM 模式 1 的输出波形。

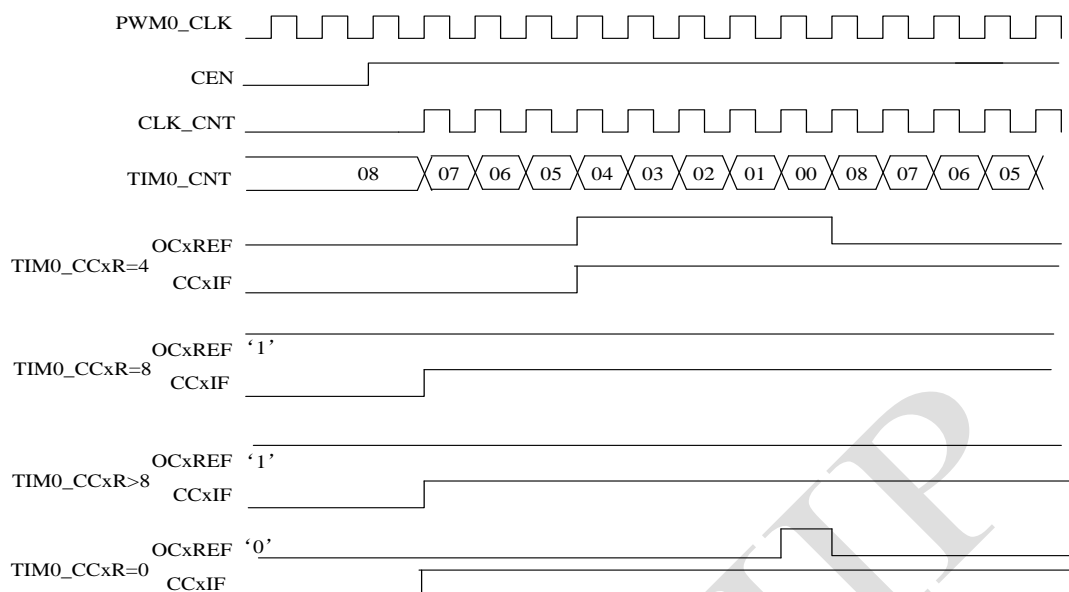


Figure 51 向下计数模式下，PWM 模式 1 输出

● 中央对齐计数模式下的 PWM 模式 1

当 TIM0_CONR2 寄存器中的 CNTMC 位配置为 01/10/11 时，计数器工作在中央对齐模式下，TIM0_CONR2 寄存器中的计数方向 CNTDIR 不能用软件配置。

在此种计数模式下，PWM 模式 1 输出波形规则为：在向上计数过程中，当 $TIM0_CNT < TIM0_CCxR$ 时，PWM 基准参考信号 OCxREF 为有效电平（高电平），否则为无效电平（低电平）；在向下计数过程中，当 $TIM0_CNT > TIM0_CCxR$ 时，PWM 基准参考信号 OCxREF 为无效电平（低电平），否则为有效电平（高电平）。当 CNTMC 位配置为 01 时，比较标志只在计数器向上计数时被置位；当 CNTMC 位配置为 10 时，比较标志只在计数器向下计数时被置位；当 CNTMC 位配置为 11 时，比较标志在计数器向上和向下计数时都被置位。Figure 52 为中央对齐的 PWM 模式 1 输出波形，图中箭头表示将比较中断标志置位。

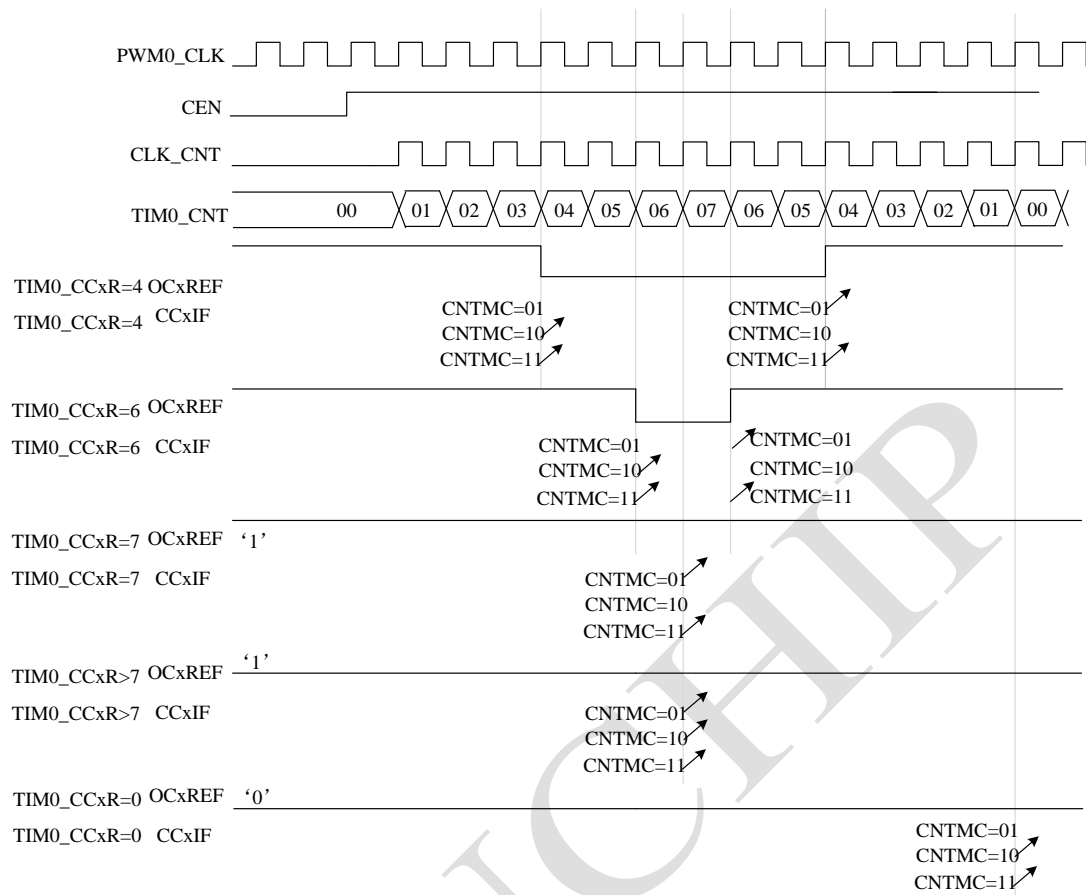


Figure 52 中央对齐计数模式下，PWM 模式 1 输出

使用中央对齐模式的注意事项：

- 在中央对齐计数模式下，若配置了计数器的复位功能（通过配置 TIM0_EGR 寄存器的 UG 位或通过从模式控制器产生复位信号），计数器开始向上计数，若没有配置计数器的复位功能，计数器按照以前保留的计数方向开始计数。此外，CNTDIR 和 CNTMC 位不能同时被软件修改，在边沿计数模式和中央对齐计数模式间转换时需按照以下步骤配置计数模式控制寄存器：

- 从边沿计数模式向中央对齐计数模式转换

首先，配置寄存器 TIM0_CONR2 的 CNTDIR 位，若希望中央计数模式下首先向下计数，则配置 CNTDIR 位为 1，若希望中央计数模式下首先向上计数，则配置 CNTDIR 位为 0，若希望中央计数模式下首先向下计数，将 CNTDIR 位配置为 1 后不可再配置 TIM0_EGR 寄存器的 UG 位。

然后，配置寄存器 TIM0_CONR2 的 CNTMC 位为 01/10/11。

- 从中央对齐计数模式向边沿计数模式转换

首先，配置寄存器 TIM0_CONR2 的 CNTMC 位为 00，将计数器切换至边沿计数模式。

然后，配置寄存器 TIM0_CONR2 的 CNTDIR 位，确定计数方向。

- 不建议当运行在中央对齐模式时改写计数器，因为会产生不可预知的结果：
 - 如果写入计数器的值大于自动重加载的值，则方向不会被更新，若计数器正在向上计数，计数器将继续向上计数。
 - 如果将 0 或者 TIM0_ARR 的值写入计数器，方向被更新，但是不产生更新事件。
- 使用中央对齐模式推荐的方法为：在启动计数器之前产生一个软件更新（设置 TIM0_EGR 寄存器中的 UG 位），不要在计数过程中修改计数器的值。

10.5.4.2 互补输出和死区插入

高级定时器能够输出两路互补信号 OCx_DT 和 OCxN_DT（相反的两路信号），也能控制输出在一段时间内无效，然后再接通，这段时间通常称为死区，本定时器的死区时间是可编程的，在具体应用中应该根据连接到输出的器件和他们的特性（电平转换的延时、电源开关的延时等）来调整死区时间。

互补输出 OCx_DT 和 OCxN_DT 在参考信号 OCxREF 基础上产生，定时器通过以下方式控制互补输出 OCx_DT 和 OCxN_DT：

- 配置 TIM0_CCPS 寄存器的 CCxP 和 CCxNP 位，可以为每一路 PWM 输出独立地选择极性；
- 通过配置 TIM0_CCxENR 寄存器的 CCxEN 和 CCxNEN 位，控制互补输出的使能。
- 当定时器工作在不同的工作状态时（如刹车状态），互补输出还受到 TIM0_BRKC 寄存器的 MOE 位、IOSS、ROSS 位和 TIM_CONR1 寄存器的 OISx、OISxN 位的控制，详见 10.5.4.4 章节。

同时设置 CCxE、CCxNE 位，则 PWM 输出将插入死区。每一个通道都有一个死区发生器，定时器根据参考信号 OCxREF 产生两路带死区的互补输出 OCx_DT 和 OCxN_DT，若两路互补输出都配置为高电平有效：

- 输出信号 OCx_DT 与参考信号 OCxREF 相同，只是他的上升沿相对于参考信号的上升沿有一个延迟。
- 输出信号 OCxN_DT 与参考信号 OCxREF 相反，只是他的上升沿相对于参考信号的下降沿有一个延迟。

如果死区时间大于当前有效的输出宽度，则不会产生相应的脉冲。

Figure 53—Figure 55 显示了带死区的互补输出与当前参考信号 OCxREF 之间的关系。

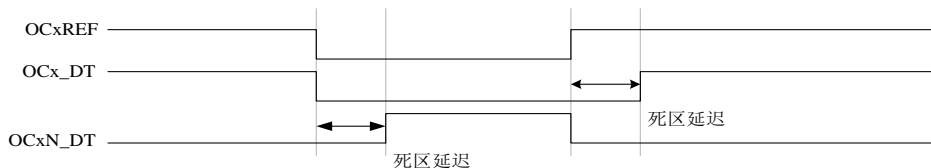


Figure 53 正常死区插入



Figure 54 死区时间大于 OCxREF 无效电平宽度时的死区插入

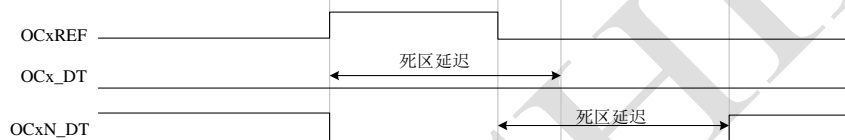


Figure 55 死区时间大于 OCxREF 有效电平宽度时的死区插入

值得注意的是：只要同时设置 CCxE 和 CCxNE 位为 1，无论在 PWM 输出模式下还是在非 PWM 输出模式下(强置输出模式和输出比较模式)，OCx 和 OCxN 均输出互补的波形，因此，在非 PWM 输出模式下，若不需要输出互补的波形，只需使能主路输出即可（即只设置 CCxE）。

10.5.4.3 相位偏移功能

在本高级定时器中，通过配置相应的寄存器可以使某一通道的 PWM 输出比其他通道的 PWM 输出具有一定的相位偏移。

将寄存器 TIM0_PHACON 的 PHASEEN 位配置为 1，打开定时器的相位偏移功能，将 PHASEEN 位配置为 0，关闭定时的相位偏移功能。

通过配置 TIM0_PHACON 的 PHASECH 位选择具有相位偏移功能（向后延迟）的输出通道，例如将 PHASECH 位配置为 00，则通道 CC0 的 PWM 输出 OCx 和 OCxN 比其他通道的 PWM 输出有一定的相位偏移。

配置 TIM0_PHASEL 和 TIM0_PHASEH 决定相位偏移的长度，当配置的相位偏移长度大于等于计数器的周期值时，输出信号不偏移，当配置的相位偏移长度为 0 时，输出信号也不偏移。

Figure 56 为通道 CC1 的 PWM 输出具有相位偏移的时序图。

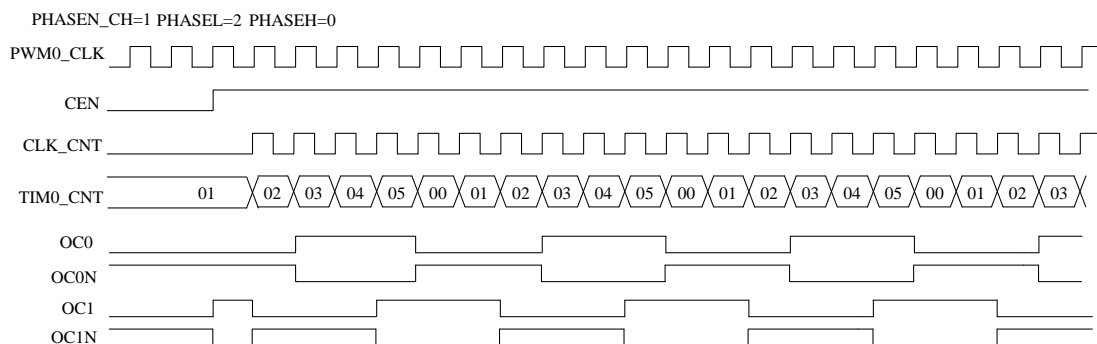


Figure 56 具有相位偏移的 PWM 输出时序图

10.5.4.4 输出使能控制

高级定时器最终向引脚输出每一通道的两路输出使能信号 PWM0_CHxEN、PWM0_CHxNEN 和每一通道互补输出 OCx、OCxN。

若输出使能信号 PWM0_CHxEN 为 1，则相应的引脚输出 OCx，若 PWM0_CHxEN 为 0，则相应的引脚不输出 OCx，而是输出高阻；输出使能信号 PWM0_CHxNEN 对引脚的控制同理。

输出使能信号 PWM0_CHxEN 和 PWM0_CHxNEN 受到 TIM0_BRKC 寄存器的 MOE、IOSS、ROSS 位和 TIM0_CCENR 寄存器的 CCxEN、CCxNEN 位控制，具体控制方式如 Table 102 和 Table 103 所示。

高级定时器的管脚输出 PWM0_CHx 和 PWM0_CHxN 受到 TIM0_BRKC 寄存器的 MOE 位、IOSS 位、ROSS 位、TIM0_CCENR 寄存器的 CCxEN、CCxNEN 位和计数器使能 CEN 的控制，当 CEN=0 时，计数器不工作，此时 PWM0_CHx 和 PWM0_CHxN 的输出值如 Table 102 所示。当 CEN=1 时，计数器正常计数，高级定时器处于正常工作状态，此时 PWM0_CHx 和 PWM0_CHxN 的输出值如 Table 103 所示。

寄存器 TIM0_BRKC 的 MOE 位为 1，表示定时器处于运行模式，MOE 为 0 表示定时器处于空闲模式。

由 Table 102 和 Table 103 可以看出，当 MOE 位为 1 时，管脚的 PWM 输出使能 (PWM0_CHxEN/PWM0_CHxNEN) 受寄存器 TIM0_BRKC 的 ROSS 位和 TIM0_CCENR 的 CCxEN、CCxNEN 位控制，ROSS 位表示运行模式下 (MOE=1) 关闭状态 (TIM0_CCENR 的 CCxEN 位或 CCxNEN 位为 0) 选择。当 ROSS 位为 0 时，表示运行模式下管脚的输出使能 PWM0_CHxEN/PWM0_CHxNEN 与 CCxEN/CCxNEN 的值一致，当 CCxE/CCxNE 配置为 0 时，相应管脚的输出使能 PWM0_CHxEN/PWM0_CHxNEN 为 0，即管脚的输出使能关闭，此时管脚输出高阻。若 CCxE/CCxNE 配置为 1，则相应管脚的输出使能 PWM0_CHxEN/PWM0_CHxNEN 为 1，管脚的输出使能有效，管脚输出 PWM 模块的输出值 OCx/OCxN。当 ROSS 位为 1 时，无论 CCxEN/CCxNEN 位配置为何值，管脚的输出使能

PWM0_CHxEN/PWM0_CHxNEN 均有效，管脚输出 PWM 模块的输出值。

当 MOE 为 0 时，管脚上的 PWM 输出和管脚的输出使能受寄存器 TIM0_BRKC 的 IOSS 位和 TIM0_CCENR 的 CCxEN、CCxNEN 位控制，IOSS 位表示空闲模式下（MOE=0）关闭状态（TIM0_CCENR 的 CCxEN 位或 CCxNEN 位为 0）选择。当 IOSS 配置为 0 时，无论 CCxEN/CCxNEN 位配置为何值，管脚的输出使能 PWM0_CHxEN/PWM0_CHxNEN 均无效，PWM 输出管脚输出高阻。当 IOSS 配置为 1 时，管脚的输出使能 PWM0_CHxEN/PWM0_CHxNEN 均有效，管脚输出 PWM 模块的输出值 OCx/OCxN。

Table 102 互补模式下，计数器不使能时 PWM 各个通道的输出及输出使能控制

CEN=0						
控制位					输出状态	
MOE 位	IOSS 位	ROSS 位	CCxEN 位	CCxNEN 位	PWM0_CHx 输出状态	PWM0_CHxN 输出状态
1	X	0	0	0	输出禁止（引脚输出高阻） PWM0_CHxEN =0; PWM0_CHx=高阻	输出禁止（引脚输出高阻） PWM0_CHxNEN =0; PWM0_CHxN=高阻;
		0	0	1	输出禁止（引脚输出高阻） PWM0_CHxEN =0 PWM0_CHx=高阻	PWM0_CHxNEN =1; PWM0_CHxN=OCxN =CCxNP;
		0	1	0	PWM0_CHxEN =1; PWM0_CHx=OCx =CCxP;	输出禁止（引脚输出高阻） PWM0_CHxNEN =0; PWM0_CHxN=高阻;
		0	1	1	PWM0_CHxEN =1 PWM0_CHx=OCx =死区+CCxP;	PWM0_CHxNEN =1; PWM0_CHxN=OCxN =死区+!CCxNP;
		1	0	0	关闭状态（输出使能有效，输出无效电平） PWM0_CHxEN =1 PWM0_CHx=OCx = CCxP;	关闭状态（输出使能有效，输出无效电平） PWM0_CHxNEN=1; PWM0_CHxN=OCxN = CCxNP;

CEN=0						
控制位					输出状态	
MOE 位	IOSS 位	ROSS 位	CCxEN 位	CCxNEN 位	PWM0_CHx 输出状 态	PWM0_CHxN 输出状态
		1	0	1	关闭状态（输出使能 有效，输出无效电 平） PWM0_CHxEN =1 PWM0_CHx=OCx = CCxP;	PWM0_CHxNEN =1; PWM0_CHxN=OCxN =CCxNP;
		1	1	0	PWM0_CHxEN =1 PWM0_CHx=OCx =CCxP;	关闭状态（输出使能有 效，输出无效电平） PWM0_CHxNEN =1; PWM0_CHxN=OCxN = CCxNP;
		1	1	1	PWM0_CHxEN =1 PWM0_CHx=OCx =CCxP;	PWM0_CHxNEN =1; PWM0_CHxN=OCxN =死区+!CCxNP;
0	0	X	0	0	输出禁止（引脚输出高阻） PWM0_CHxEN = 0; PWM0_CHxNEN = 0; PWM0_CHx =高阻; PWM0_CHx N=高阻;	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0	关闭状态（输出使能有效，输出无效电平） PWM0_CHxEN = 1, PWM0_CHxNEN = 1; ● 当 CCxP=CCxNP 时: 若 OISx=OISx, PWM0_CHx=OCx =CCxP, PWM0_CHxN=OCxN =CCxNP; 若 OISxOISxN, PWM0_CHx=OCx =OISx, PWM0_CHxN=OCxN =OISxN。 ● 当 CCxP≠CCxNP 时: 若 OISx=OISxN, 则 PWM0_CHx=OCx=OIS, PWM0_CHxN=OCxN =OISxN; 若 OIS≠OISxN, 则 PWM0_CHx=OCx =CCxP	
	1		0	1		
	1		1	0		
	1		1	1		

CEN=0						
控制位					输出状态	
MOE 位	IOSS 位	ROSS 位	CCxEN 位	CCxNEN 位	PWM0_CHx 输出状态	PWM0_CHxN 输出状态
					PWM0_CHxN=OCxN =CCxNP。	

Table 103 互补模式下，计数器使能时 PWM 各个通道的输出及输出使能控制

CEN=1						
控制位					输出状态	
MOE 位	IOSS 位	ROSS 位	CCxE N 位	CCxNE N 位	PWM0_CHx 输出状态	PWM0_CHxN 输出状态
1	X	0	0	0	输出禁止（引脚输出高阻） PWM0_CHxEN =0; PWM0_CHx=高阻;	输出禁止（引脚输出高阻） PWM0_CHxNEN =0; PWM0_CHxN=高阻;
		0	0	1	输出禁止（引脚输出高阻） PWM0_CHxEN =0; PWM0_CHx=高阻;	PWM0_CHxNEN =1; PWM0_CHxN=OCxN =OCxREF xor CCxNP
		0	1	0	PWM0_CHxEN =1; PWM0_CHx=OCx =OCxREF xor CCxP	输出禁止（引脚输出高阻） PWM0_CHxNEN =0; PWM0_CHxN=高阻
		0	1	1	PWM0_CHxEN =1; PWM0_CHx=OCx =OCxREF +死区+ 极性	PWM0_CHxNEN =1; PWM0_CHxN=OCxN =OCxREF +死区+ 极性
		1	0	0	关闭状态（输出使能有效，输出无效电平） PWM0_CHxEN =1; PWM0_CHx=OCx = CCxP	关闭状态（输出使能有效，输出无效电平） PWM0_CHxNEN =1; PWM0_CHxN=OCxN = CCxNP
		1	0	1	关闭状态（输出使能有效，输出无效电平）	PWM0_CHxNEN =1;

CEN=1						
控制位					输出状态	
MO E 位	IOSS 位	ROS S 位	CCxE N 位	CCxNE N 位	PWM0_CHx 输出状态	PWM0_CHxN 输出状态
					效, 输出无效电平) PWM0_CHxEN = 1; PWM0_CHx = OCx = CCxP	PWM0_CHxN = OCxN = OCxREF xor CCxNP
		1	1	0	PWM0_CHxEN = 1; PWM0_CHx = OCx = OCxREF xor CCxP	关闭状态 (输出使能有效, 输出无效电平) PWM0_CHxNEN = 1; PWM0_CHxN = OCxN = CCxNP
		1	1	1	PWM0_CHxEN = 1; PWM0_CHx = OCx = OCxREF + 死区 + 极性	PWM0_CHxNEN = 1; PWM0_CHxN = OCxN = OCxREF + 死区 + 极性
0	0	X	0	0	输出禁止 (引脚输出高阻)	
	0		0	1	PWM0_CHxEN = 0; PWM0_CHxNEN = 0;	
	0		1	0	PWM0_CHx = 高阻;	
	0		1	1	PWM0_CHxN = 高阻;	
	1		0	0	关闭状态 (输出使能有效, 输出无效电平)	
	1		0	1	PWM0_CHxEN = 1, PWM0_CHxNEN = 1;	
	1		1	0	● 当 CCxP=CCxNP 时: 若 OISx=OISxN, PWM0_CHx=OCx =CCxP, PWM0_CHxN=OCxN =CCxNP; 若 OISxOISxN, PWM0_CHx=OCx =OISx, PWM0_CHxN=OCxN =OISxN。	
	1		1	1	● 当 CCxPCCxNP 时: 若 OISx=OISxN, 则 PWM0_CHx=OCx =OISx, PWM0_CHxN=OCxN =OISxN; 若 OISxOISxN, 则 PWM0_CHx=OCx =CCxP, PWM0_CHxN=OCxN =CCxNP。	

由 Table 102 和 Table 103 可知,管脚上的 PWM 输出使能和 PWM 输出受 MOE、ROSS、IOSS、CCxE、CCxNEN、CCxP、CCxNP 和 CEN 控制,这些寄存器初始值都为 0,所以初始状态下 PWM 管脚输出高阻,为了保证 PWM 平稳启动,建议按照如下顺序配置寄存器:

- 1) 首先配置极性寄存器 TIM0_CCxPS,若需要控制空闲模式(MOE=0)下的输出电平,可以配置寄存器 TIM0_CONR1。
- 2) 配置 TIM0_CONR2 的 ARPLE 位和 TIM0_CONR0 的 CCPE 位,控制相应寄存器的预装载功能。
- 3) 配置各个通道输入捕获/输出比较模式控制寄存器 TIM0_CCxMR。
- 4) 配置周期寄存器 TIM0_ARR、比较/捕获寄存器 TIM0_CCxR、通道使能寄存器 TIM0_CCxEN、死区控制等各种寄存器。
- 5) 配置软更新事件,将新配置的参数装载到影子寄存器中(配置 TIM0_EGR 的 UG 位为 1, COMG 位为 1,装载各个预装载寄存器)。
- 6) 配置 TIM0_BRKC 寄存器,对 MOE 位、ROSS 位、和 IOSS 位进行配置(建议配置 ROSS 和 IOSS 位为 1)。
- 7) 配置 TIM0_CNTEN 寄存器的 CNTEN 位,启动 PWM 计数器。

10.5.4.5 刹车功能

当使用刹车功能时,依据相应的控制位(TIM0_BRKC 寄存器的 MOE、IOSS 和 ROSS, TIM0_CONR1 寄存器的 OISx 和 OISxN 位),输出使能和输出信号都会被修改。但无论何时,均保证输出信号 PWM0_CHx 和 PWM0_CHxN 不会同时处于有效电平。

10.5.4.5.1 刹车源

刹车源包括三种:

- 引脚刹车事件。
- 软刹车事件。通过配置刹车事件寄存器(配置寄存器 TIM0_EGR 的 BG 位为 1)对 PWM 输出进行刹车。
- 比较器刹车,比较器的输出信号可以作为 PWM 的刹车信号,详见 0 节。

系统复位后,刹车电路被禁止,MOE 位为低,定时器工作在空闲模式下。设置 TIM0_BRKC 寄存器的 BRKEN 位可以使能刹车功能(软刹车不受 BRKEN 位控制),刹车输入信号的极性可以通过配置寄存器 TIM0_BRKC 中的 BKP 位选择。

10.5.4.5.2 刹车过程

当刹车事件发生时,硬件做如下处理:

- 非互补模式下:

MOE 位被异步地立即清零,每一个通道输出由寄存器 TIM0_CONR2 的 OISx 位或 OISxN 位设定电平,如果 IOSS=0,则定时器管脚输出使能无效,否则管脚输出使能始

终有效。这个特性在系统时钟关闭时依然有效。

- 互补模式下：
 - 输出首先被置于无效状态，即使没有时钟，此功能也有效。
 - 如果定时器的时钟存在，死区生成器会重新生效，在死区之后根据 OISx、OISxN 位以及极性指示的电平驱动输出端口，详见 Table 102 和 Table 103 的 MOE=0 部分。即使在这种情况下，PWM0_CHx 和 PWM0_CHxN 也不会被同时驱动到有效电平。因为重新同步 MOE，死区时间比通常情况长一些（大约 2 个定时器时钟周期）。
 - 如果 IOSS=0，定时器管脚输出使能无效，管脚输出高阻，否则管脚输出使能有效。
- 如果设置了寄存器 TIM0_IER 中的 BIE 位，则产生一个中断。
- 如果寄存器 TIM0_BRKC 中的 AOE 位为 1，在下一个更新事件时 MOE 位被自动置 1；否则，MOE 始终保持为低直到被软件再次置 1；

当刹车输入有效时，软件无法配置 MOE，同时，状态标志不能被清除。

10.5.4.5.3 刹车后输出信号控制

因为当外部刹车信号来到时，MOE 必须立即变为低电平，即 MOE 的下降沿可以是异步的，所以在实际信号（作用在定时器的输出端）和寄存器的控制位之间设置了一个同步电路，当向 MOE 位写入数据时，硬件会将软件写入的值作为异步信号对待，对其做同步后再写入相应的寄存器位。因此，如果当 MOE 为 0 时写 MOE 为 1，则读出它之前必须先插入一个延时（空指令）才能读到正确的值，这是因为写入的是异步信号而读的是同步信号。

寄存器 TIM0_BRKC 的 MOE 位控制 PWM 在管脚上的输出，当 MOE 为 1 时，PWM 处于运行模式下，若 TIM0_BRKC 寄存器的 ROSS 位设置为 1，则无论 CCxE 和 CCxNE 配置为使能与否，PWM0_CHx 和 PWM0_CHxN 管脚的输出使能都为高；若 TIM0_BRKC 寄存器的 ROSS 位设置为 0，则当 CCxE 配置为有效时，PWM0_CHx 管脚的输出使能有效，否则，PWM0_CHx 管脚的输出使能为无效，PWM0_CHx 管脚输出高阻，PWM0_CHxN 的输出管脚控制同理。

当 MOE 为 0 位，PWM 输出处于空闲模式（刹车后的状态），若 TIM0_BRKC 寄存器的 IOSS 位设置为 1，则无论 CCxE 和 CCxNE 配置为使能与否，PWM0_CHx 和 PWM0_CHxN 管脚的输出使能都有效；若 TIM0_BRKC 寄存器的 IOSS 位设置为 0，则 PWM0_CHx 和 PWM0_CHxN 管脚的输出使能设置为无效，PWM0_CHx 和 PWM0_CHxN 管脚输出高阻。当 CNTEN 为 0 时，计数器不计数，定时器处于初始化状态，PWM0_CHx 和 PWM0_CHxN 管脚上的输出如 Table 102 所示，当 CNTEN 为 1 时，定时器处于工作状态，PWM0_CHx 和 PWM0_CHxN 管脚上的输出如 Table 103 所示。

Figure 57 显示了响应刹车的输出实例。

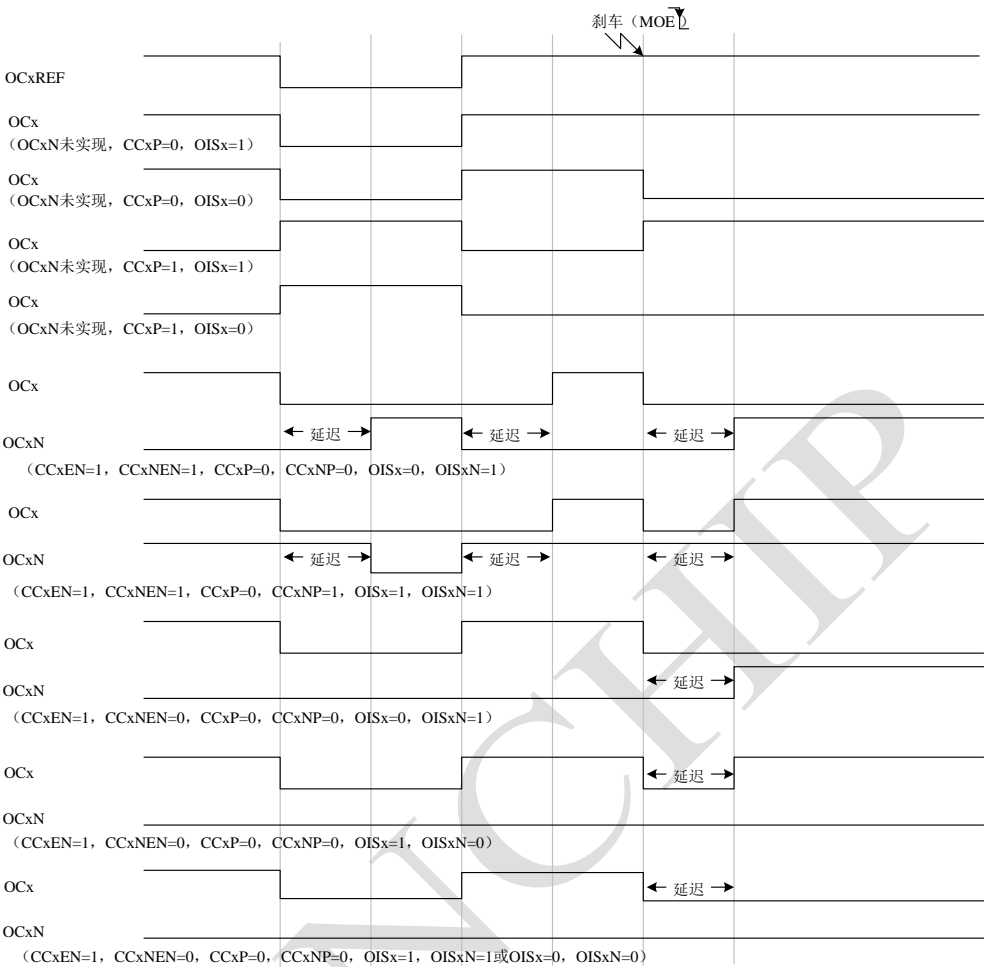


Figure 57 高级定时器刹车输出

10.5.4.6 外部事件 (PWM0_ETR) 清除 OCxREF 信号

对于一个给定的通道，设置寄存器 TIM0_CCxMR 中的 OCxCE 位为 1，能够用 PWM0_ETR 通道输入信号的有效电平把 OCxREF 信号清为低电平，PWM0_ETR 信号无效后 OCxREF 信号将保持为无效电平直到发生下一次更新事件，该功能只能用于输出比较和 PWM 模式，而不能用于强置模式。PWM0_ETR 通道配置如下：

- 关闭 PWM0_ETR 通道的预分频器。
- 禁止外部时钟模式 2。
- 配置外部触发信号的极性(EXTP)。

Figure 58 显示了当 PWM0_ETR 输入变为高时，对应不同 OCxCE 的值，OCxREF 信号的变化。在这个例子中，定时器被设置为 PWM 模式。

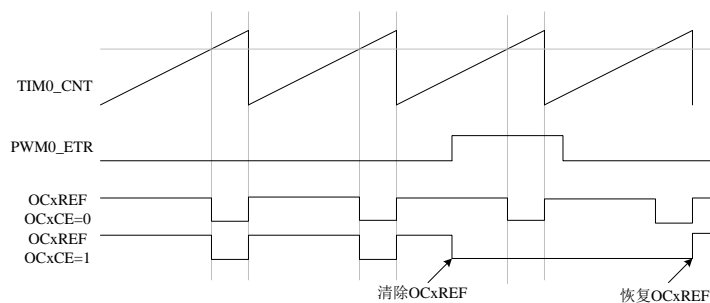


Figure 58 外部信号 PWM0_ETR 清除 OCxREF

10.5.4.7 霍尔传感器接口功能

高级定时器 TIM0 可以作为“接口定时器”来连接霍尔传感器。配置 TIM0_CONR0 寄存器的 CH0INSEL 位为 1, 选择定时器的三个输入脚(PWM0_CH0、PWM0_CH1、PWM0_CH2) 异或后的信号连接至 CC0 输入通道, 高级定时器捕获这个信号。

配置“接口定时器”的从模式控制器为复位模式, 选择 CH0INF_ED 作为从模式控制器的触发输入, 每当 3 个霍尔信号中任意一个发生翻转时, 计数器重新从 0 开始计数, 这样产生一个由霍尔输入端的任何变化而触发的时间基准。

“接口定时器”TIM1 上的通道 CC0 配置为捕获模式, 捕获信号为 CH0INF_ED, 捕获值反映了两个输入变化间的时间延迟, 给出了马达速度的信息。

“接口定时器”TIM1 可以将 CC1-CC3 中的某一通道配置为输出模式, 在输出模式下产生一个比较输出波形, 将此波形通过 TRGO 送到另一个高级定时器 TIM0, 在 TIM0 中利用此波形触发 TIM0 的 COM 事件。

10.5.4.8 六步 PWM

通过高级定时器 TIM0 的 COM 事件产生六步 PWM。COM 事件可由两种方式产生: 一种通过霍尔信号产生, 另外一种通过无霍尔应用中的反电动势产生。下面以 TIM0 作为 PWM 输出, 使用 TIM1 作为霍尔信号接口产生 COM 事件为例阐述六步 PWM 的控制过程。

霍尔信号产生 COM 事件的方式有以下两种:

- 第一种方式: 将三路霍尔信号连接到“接口定时器”TIM1, 在 TIM1 产生一个触发信号, 然后通过 TRGO 传送到 TIM0。若寄存器 TIM0_CONR0 的 CCUS 为 0, TRGO 送入 TIM0 后引起触发中断, 在中断处理程序中设置六步 PWM 相应相位的输出控制寄存器, 并配置 TIM0_EGR 寄存器的 COM 位进行换相; 若寄存器 TIM0_CONR0 的 CCUS 位为 1, TRGO 进入 TIM0 后触发 COM 事件进行 PWM 换相, 同时引起触发中断, 在中断处理程序中配置相应相位的输出控制寄存器。
- 第二种方式: 将三路霍尔信号连接到芯片的输入引脚, 打开三个引脚脚的中断使能, 当霍尔信号发生跳变时, 管脚发出中断, 在中断中设置 TIM0_EGR 寄存器的 COM

位进行换相。

每个六步 PWM 相位的输出控制有以下两种方法：

- 通过不同的 OCxMS、CCxE 和 CCxNE 的设置可以得到六步 PWM 每个相位的输出，这些控制位具有预装载功能，更新事件为 COM 事件。
- 通过设置 CCxE、CCxNE、CCxP 和 CCxNP 也可以得到六步 PWM 每个相位的输出，这些控制位具有预装载功能，更新事件为 COM 事件。

在发生 COM 事件时，这些预装载位被传送到影子寄存器，这样就可以预先设置好相应相位的输出控制寄存器，并在同一个时刻更新所有通道的配置。

例 1：霍尔信号连接到定时器 TIM1，在任一霍尔输入变化指定的延迟后，改变高级定时器 TIM0 的配置，实现 TIM0 的六步 PWM 输出。Figure 59 为 TIM1 作为霍尔传感器接口驱动 TIM0 产生六步 PWM 的示意图。

- 将 TIM1 的计数器周期值配置为最大值，即配置寄存器 TIM1_ARRH 为 0xFF，配置寄存器 TIM1_ARRL 为 0xFF。
- 配置寄存器 TIM1_CONR0 的 CH0INSEL 位为 1，将 TIM1_CH0、TIM1_CH1 和 TIM1_CH2 管脚上的信号异或后连接到通道 CC0。
- 配置寄存器 TIM1_TGICR0 的 TRGS 位为 100，选择 CH0INF_ED 作为从模式控制器的触发输入信号。
- 配置寄存器 TIM1_TGICR0 的 SMS 位为 100，将从模式控制器配置为复位模式。
- 配置寄存器 TIM1_CC0MR 的 CC0MS 位为 0x70，将通道 CC0 配置为 PWM2 模式。
- 配置寄存器 TIM1_CC1MR 的 CC1MS 位为 0x73，将通道 CC1 配置输入捕获模式。
- 配置寄存器 TIM1_CC0R，指定延迟产生一个正脉冲，该寄存器的取值范围为 0x01 至两个相邻霍尔信号跳变的间隔。
- 配置寄存器 TIM1_CONR0 的 MMC 位为 100，将正脉冲 OC0REF 选择为 TRGO 输出。
- 配置 TIM0 的周期值寄存器 TIM0_ARR、输出模式控制寄存器 TIM0_CCxMR、极性控制寄存器 TIM0_CCPS、通道使能控制寄存器 TIM0_CCENR、比较值寄存器 TIM0_CCxR 等。
- 配置寄存器 TIM0_TGICR0 的 TRGS 位为 000，选择 ITR0（TIM1 的 TRGO 信号）作为 TIM0 从模式控制器的触发输入信号。
- 配置寄存器 TIM0_CONR0 的 CCPE 位为 1，使能比较输出控制寄存器的预装载功能。
- 配置寄存器 TIM0_IER 的 TRIE 位为 1，使能触发中断。在触发中断处理程序中通过配置寄存器 TIM0_CCENR、TIM0_CCPS 实现对六步 PWM 每个相位的控制，同时在触发中断服务程序中配置 TIM0_EGR 的 COM 位，产生 COM 事件，更新 TIM0

的输出控制寄存器。

- 配置寄存器 TIM0_TGICR0 的 SMS 位为 110，将从模式控制器配置为触发模式。

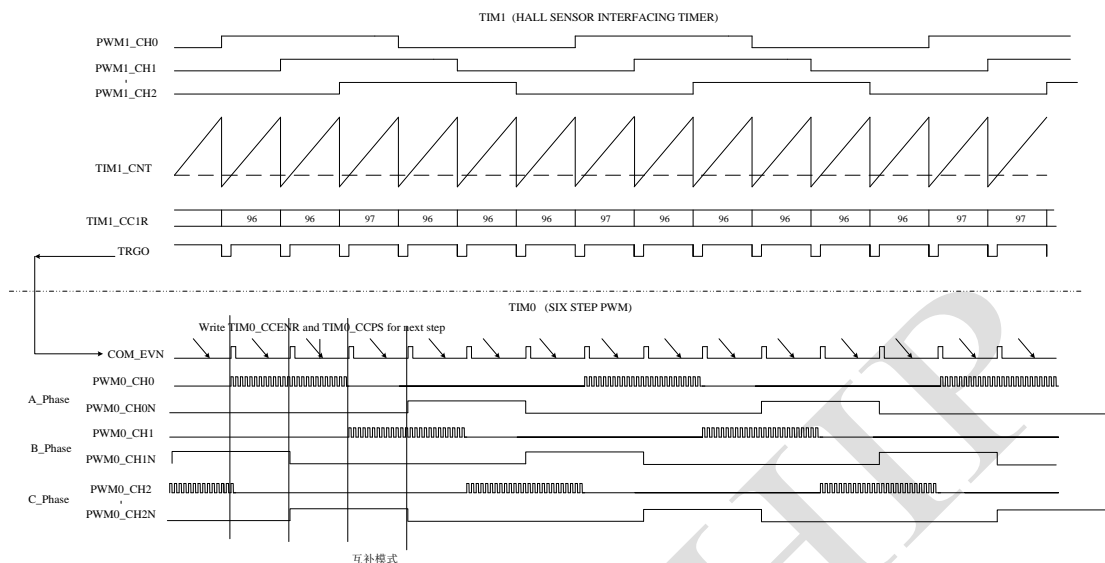


Figure 59 TIM1 作为霍尔传感器接口驱动 TIM0 产生六步 PWM 的示意图

例 2：将三路霍尔信号连接到 GPIO 的 PA0，PA1 和 PA2 管脚，打开 PA0、PA1 和 PA2 的中断使能，在 GPIO 中断中软件触发 TIM0 的 COM 事件来改变 TIM0 的输出配置。

- 配置 TIM0 的周期值寄存器 TIM0_ARR、输出模式控制寄存器 TIM0_CCxMR、极性控制寄存器 TIM0_CCPS、通道使能控制寄存器 TIM0_CCENR、比较值寄存器 TIM0_CCxRU 或 TIM0_CCxRD 等。
- 配置寄存器 TIM0_CONR0 的 CCPE 位为 1，使能比较输出控制寄存器的预装载功能。
- 将三路霍尔信号连接到 PA0，PA1 和 PA2 管脚，配置 PA0，PA1 和 PA2 为 GPIO 管脚模式。
- 配置寄存器 INT0EN 为 0x07，配置寄存器 INT0PE 为 0x07，配置寄存器 INT0NE 为 0x07，将 PA0、PA1 和 PA2 的上升沿下降沿中断使能打开。
- 在中断处理函数中通过配置寄存器 TIM0_CCENR、TIM0_CCPS 实现对六步 PWM 每个相位的控制，在触发中断处理函数中配置 TIM0_EGR 的 COM 位，产生 COM 事件，更新 TIM0 的输出控制寄存器。

10.5.4.9 单脉冲模式

将寄存器 TIM0_CONR2 的 OPM 位置 1，则定时器工作在单脉冲模式下，在此种模式下，计数器计数到周期值，定时器输出一个单脉冲，然后停止输出。

这种模式允许计数器响应一个激励，并在一个程序可控的延时后产生一个脉宽可程序控

制的脉冲。

将寄存器 TIM0_CNTEN 的 CNTEN 位置 1 或通过触发信号触发，都可以使能计数器，定时器在输出比较模式或 PWM 模式下产生波形，若发生更新事件或软件清除计数器使能，则计数器停止计数，假如软件清除计数器使能导致计数器停止计数，则计数器的值保持不变，假如计数器在发生更新事件时停止，则计数器处于初始化状态。

在计数器启动时，只有比较值与计数器的初始值不同时，才能产生一个脉冲，计数器启动前，配置的比较值 (TIM0_CCxR) 需满足以下条件：

- 向上计数模式：TIM0_CNT < TIM0_CCxR < TIM0_ARR
- 向下计数模式：TIM0_CNT > TIM0_CCxR

例 1：从 PWM0_CHx 输入脚上检测到一个上升沿开始，延时 t1 (DELAY) 后，在 OC1 上产生一个长度为 t2 (PULSE) 的正脉冲。

- 配置寄存器 TIM0_CC1MR 的 CC1MS 位为 10，选择 CC0 通道的输入信号作为 CC1 通道的触发信号。
- 配置寄存器 TIM0_CCPS 的 CC1P 位为 0，将输入信号的上升沿作为有效边沿。
- 配置寄存器 TIM0_TGICR0 的 TRGS 位为 110，选择 CH0INF_ED 作为从模式控制器的触发输入信号。
- 配置寄存器 TIM0_CC1R，TIM0_CC1R = t1 (DELAY)。
- 配置寄存器 TIM0_ARR，TIM0_ARR = t1 (DELAY) + t2 (PULSE)。
- 假设在比较匹配时需要产生由 0 到 1 的波形，计数器计到周期值时产生由 1 到 0 的波形，将寄存器 TIM0_CC1MR 的 OC1MS 位配置为 111 (即 PWM 模式 2)；假设在比较匹配时需要产生由 1 到 0 的波形，计数器计到周期值时产生由 0 到 1 的波形，将寄存器 TIM0_CC1MR 的 OC1MS 位配置为 110 (即 PWM 模式 1)。
- 将寄存器 TIM0_CONR2 的 OPM 位配置为 1，使能单脉冲模式。
- 配置寄存器 TIM0_TGICR0 的 SMS 位为 110，将从模式控制器配置为触发模式。

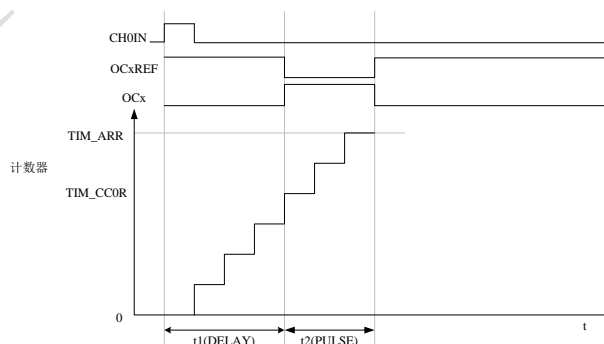


Figure 60 单脉冲模式的例子

10.5.5 主模式控制

定时器工作在主模式下，可产生触发信号 TRGO 去控制 AD 采样模块。Figure 61 为主模式控制器示意图。

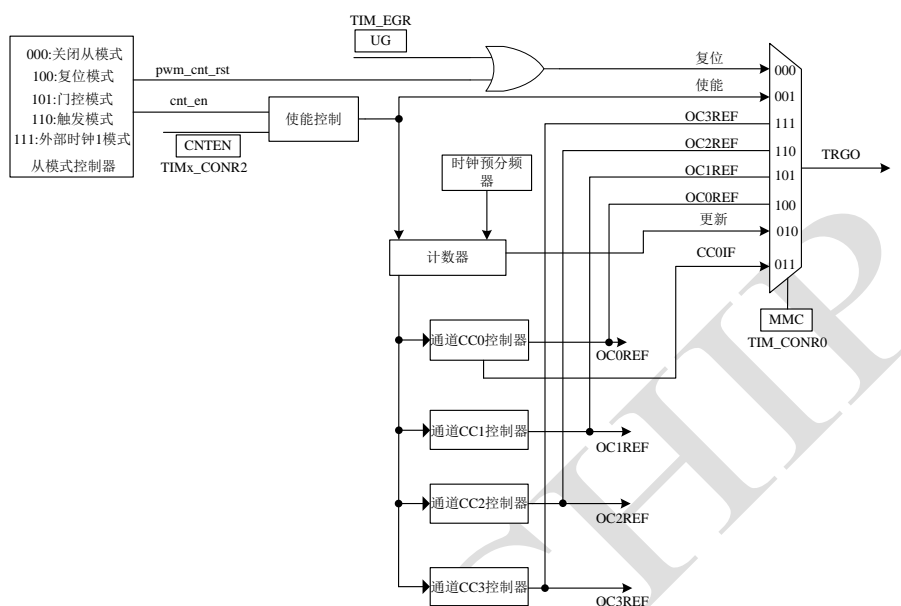


Figure 61 主模式控制示意图

主模式控制器的主要功能是选择定时器的相应信号作为触发输出 TRGO，控制过程如下：

- 配置寄存器 TIM0_CONR0 的 MMC 位为 000，选择计数器的复位信号作为触发输出，复位信号包括由从模式控制器产生的复位信号和 TIM0_EGR 寄存器的 UG 位。当 UG 为配置为 1 时，产生一个初始化信号去初始化寄存器，同时将此信号选为触发输出。
- 配置寄存器 TIM0_CONR0 的 MMC 位为 001，选择计数器的使能信号作为触发输出 TIM0_TRGO。
- 配置寄存器 TIM0_CONR0 的 MMC 位域为 010，选择通道 CC0 的比较标志 CC0IF 作为触发输出 TIM0_TRGO。若通道 CC0 配置为输入捕获功能，当发生一次捕获时，CC0IF 置位；若通道 CC0 配置为输出比较功能，当计数器的值和 TIM0_CC0R 匹配时，CC0IF 置位。在这两种情况下都可以将 CC0IF 信号作为触发输出 TIM0_TRGO。
- 配置寄存器 TIM0_CONR0 的 MMC 位为 011，选择更新事件 UEV 作为触发输出。
- 配置寄存器 TIM0_CONR0 的 MMC 位为 100，选择通道 CC0 的基准参考信号 OC0REF 作为触发输出。
- 配置寄存器 TIM0_CONR0 的 MMC 位为 101，选择通道 CC1 的基准参考信号 OC1REF 作为触发输出。

- 配置寄存器 TIM0_CONR0 的 MMC 位为 110，选择通道 CC2 的基准参考信号 OC2REF 作为触发输出。
- 配置寄存器 TIM0_CONR0 的 MMC 位域为 111，选择通道 CC3 的基准参考信号 OC3REF 作为触发输出。

定时器发出的触发 AD 采样的 CC 事件有：

- 定时器 TIM0 的触发输出 TIM0_TRGO；
- 定时器 TIM0 的通道 CC0 的主路输出 TIM0_OC0；
- 定时器 TIM0 的通道 CC1 的主路输出 TIM0_OC1；
- 定时器 TIM0 的通道 CC2 的主路输出 TIM0_OC2；
- 定时器 TIM0 的通道 CC3 的主路输出 TIM0_OC3；

10.5.6 中断

10.5.6.1 中断源

高级定时器共有 17 个中断源，每个中断产生的条件如下：

1) 刹车中断。若已打开刹车中断使能，刹车中断产生的条件有：

- 刹车使能有效时外部输入刹车信号。
- 软件配置寄存器 TIM0_EGR 的 BG 位为 1，产生软刹车事件。
- 产生比较器刹车信号。

2) 触发中断。若已打开触发中断使能，触发中断产生的条件有：

- 软件配置寄存器 TIM0_EGR 的 TG 位为 1，产生触发事件。
- 在从模式控制器的触发输入端 TRGI 检测到有效边沿。

3) COM 中断。若已打开 COM 中断使能，COM 中断产生的条件有：

- 软件配置寄存器 TIM0_EGR 的 COM 位为 1，产生 COM 事件。

4) 通道 CCx 的捕获/比较中断。若已打开通道 CCx 的捕获/比较中断使能，通道 CCx 的捕获/比较中断产生的条件有：

- 若通道 CCx 配置为输入捕获功能（寄存器 TIM0_CCxMR 的 CCxMS 位配置为 01/10/11），软件配置寄存器 TIM0_EGR 的 CCxG 位为 1，产生捕获事件。
- 若通道 CCx 配置为输入捕获功能（寄存器 TIM0_CCxMR 的 CCxMS 位配置为 01/10/11），检测到通道 CCx 的捕获信号 CHxINCP 的有效边沿。
- 若通道 CCx 配置为输出比较功能（寄存器 TIM0_CCxMR 的 CCxMS 位配置为 00），软件配置寄存器 TIM0_EGR 的 CCxG 位为 1，产生比较事件。
- 若通道 CCx 配置为输出比较功能（寄存器 TIM0_CCxMR 的 CCxMS 位配置为 00），计数器与比较值匹配时。

5) 计数器上溢更新中断。若已打开上溢更新中断使能，上溢更新中断产生的条件

有：

- 若计数器的上溢事件没有被屏蔽（配置寄存器 TIM0_CONR3 的 UPOUDIS 位为 0），当计数器上溢时产生计数器上溢更新中断。

6) 计数器下溢更新中断。若已打开下溢更新中断使能，下溢更新中断产生的条件有：

有：

- 若计数器的下溢事件没有被屏蔽（配置寄存器 TIM0_CONR3 的 DOWNOUDIS 位为 0），当计数器下溢时产生计数器下溢更新中断。

7) 复位更新中断。若已打开通复位更新中断使能，复位更新中断产生的条件有：

- 设置 TIM0_EGR 寄存器的 UG 位为 1。
- 从模式控制器产生的复位信号。

各个中断源产生后，在 TIM0 中进行中断合并得到中断 PWM0_INT，将 PWM0_INT 送入 JMT51，在 JMT51 中经过一级门控后得到 JMT51 中的 TIM0 中断，Figure 62 为中断产生过程。

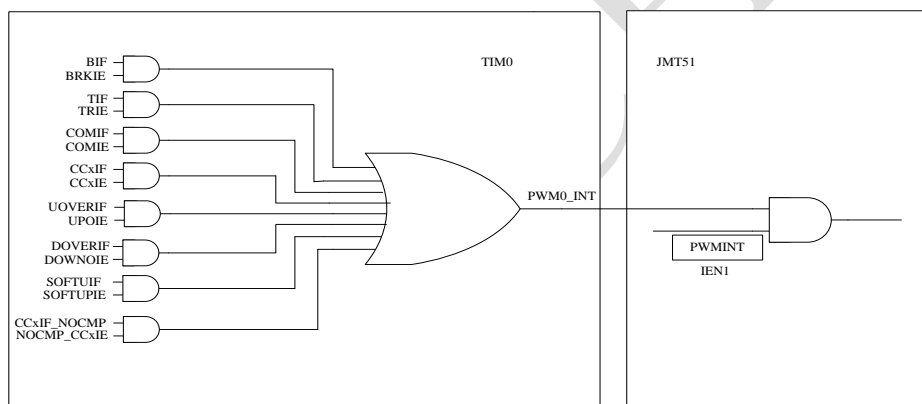


Figure 62 JMT18F003PLUS 高级定时器中断

10.5.6.2 中断配置及处理

TIM 中断配置及处理过程如下：

- 配置中断使能寄存器 TIM0_IER 和 TIM0_CONR3 的中断使能位，使能相应的中断源。
- 配置寄存器 IEN1 的 PWMINT 位，使能高级定时器中断。
- 当产生 TIM0 中断后，软件需要回读中断标志寄存器 TIM0_ISR0、TIM0_ISR1，来判断高级定时器的中断源，根据不同中断源，向中断标志寄存器相应位写 0 清除中断，并做相应处理。

10.6 寄存器描述

PWM0 模块有 38 个寄存器，对应的 SFRPAGE 为 1，具体说明如 Table 104 所示。

Table 104 PWM0 寄存器 (SFRPAGE=0x01)

地址	寄存器名	属性	复位值	功能描述
0XD1	TIM0_CONR0	W/R	0x00	TIM0 控制寄存器 0
0XD2	TIM0_CONR1	W/R	0x00	TIM0 控制寄存器 1
0XD3	TIM0_CONR2	W/R	0x00	TIM0 控制寄存器 2
0XD4	TIM0_CONR3	W/R	0x20	TIM0 控制寄存器 3
0XD5	TIM0_TGICR0	W/R	0x00	从模式控制寄存器 0
0XD6	TIM0_TGICR1	W/R	0x00	从模式控制寄存器 1
0XD7	TIM0_IER0	W/R	0x00	TIM0 中断使能寄存器 0
0XD9	TIM0_CC0MR	W/R	0x00	TIM0 通道 CC0 输入捕获/输出比较模式控制寄存器
0XDA	TIM0_CC1MR	W/R	0x00	TIM0 通道 CC1 输入捕获/输出比较模式控制寄存器
0XDB	TIM0_CC2MR	W/R	0x00	TIM0 通道 CC2 输入捕获/输出比较模式控制寄存器
0XDC	TIM0_CC3MR	W/R	0x00	TIM0 通道 CC3 输入捕获/输出比较模式控制寄存器
0XDD	TIM0_CCENR	W/R	0x00	通道使能控制寄存器
0XDE	TIM0_CCPS	W/R	0x00	通道极性控制寄存器
0XDF	TIM0_DTG	W/R	0x00	TIM0 死区时间寄存器
0XE1	TIM0_ARRL	W/R	0x00	TIM0 周期值自动装载寄存器的低 8 位
0XE2	TIM0_ARRH	W/R	0x00	TIM0 周期值自动装载寄存器的高 8 位
0XE3	TIM0_PSCL	W/R	0x00	TIM0 预分频寄存器的低 8 位
0XE4	TIM0_PSCH	W/R	0x00	TIM0 预分频寄存器的高 8 位
0XE5	TIM0_RCR	W/R	0x00	TIM0 重复计数寄存器
0XE6	TIM0_CC0RL	W/R	0x00	通道 CC0 捕获/比较寄存器的低 8 位
0XE7	TIM0_CC0RH	W/R	0x00	通道 CC0 捕获/比较寄存器的高 8 位
0XE9	TIM0_CC1RL	W/R	0x00	通道 CC1 捕获/比较寄存器的低 8 位
0XEA	TIM0_CC1RH	W/R	0x00	通道 CC1 捕获/比较寄存器的高 8 位
0XEB	TIM0_CC2RL	W/R	0x00	通道 CC2 捕获/比较寄存器的低 8 位

地址	寄存器名	属性	复位值	功能描述
0XEC	TIM0_CC2RH	W/R	0x00	通道 CC2 捕获/比较寄存器的高 8 位
0XED	TIM0_CC3RL	W/R	0x00	通道 CC3 捕获/比较寄存器的低 8 位
0XEE	TIM0_CC3RH	W/R	0x00	通道 CC3 捕获/比较寄存器的高 8 位
0XEF	TIM0_BRKC	W/R	0x00	TIM0 刹车控制寄存器
0XF1	TIM0_EGR	W	0x00	TIM0 事件产生寄存器
0XF2	TIM0_PHACON	W/R	0X00	相位偏移控制寄存器
0XF3	TIM0_ISR0	W/R	0x00	TIM0 中断标志寄存器 0
0XF4	TIM0_ISR1	W/R	0x00	TIM0 中断标志寄存器 1
0XF5	TIM0_CNTL	W/R	0x00	TIM0 计数器的低 8 位
0XF6	TIM0_CNTH	W/R	0x00	TIM0 计数器的高 8 位
0XF7	TIM0_PHASEL	W/R	0x00	相位偏移长度的低 8 位
0XF9	TIM0_PHASEH	W/R	0x00	相位偏移长度的高 8 位
0XFA	TIM0_CNTEN	W/R	0x00	TIM0 计数器使能寄存器
0XFB	TIM0_IER1	W/R	0x00	TIM0 中断使能寄存器 1

10.6.1 TIM0 控制寄存器 0 (TIM0_CONR0) [0xD1]

Table 105 TIM0 控制寄存器 0 (TIM0_CONR0)

参数名	比特位	属性	复位值	描述
CLKDIV	1-0	W/R	0	<p>时钟分频系数，定义了定时器时钟（pwm0_clk）与死区及滤波时钟之间的分频系数。时钟 dts_clk 用于死区时间发生器以及数字滤波器。</p> <p>00: $T_{dts_clk} = T_{pwm0_clk}$</p> <p>01: $T_{dts_clk} = 2 \times T_{pwm0_clk}$</p> <p>10: $T_{dts_clk} = 4 \times T_{pwm0_clk}$</p> <p>11: 保留。</p>
CHOINSEL	2	W/R	0	<p>通道 CC0 输入信号选择寄存器：</p> <p>0: PWM0_CH0 管脚上的信号连接到通道 CC0</p> <p>1: PWM0_CH0、PWM0_CH1 和 PWM0_CH2 管脚上的信号异或后连接到通道 CC0</p>

参数名	比特位	属性	复位值	描述
MMC	5-3	W/R	0	<p>主模式下输出信号选择寄存器，当定时器用作主定时器时，此寄存器用于选择送到从定时器的同步信号（TRGO）：</p> <p>000：复位信号</p> <ul style="list-style-type: none"> 当 TIM0_EGR 寄存器的 UG 置位时，或由从模式控制器产生复位时，复位信号作为触发输出（TRGO）。 <p>001：使能信号</p> <ul style="list-style-type: none"> 计数器使能信号被用于作为触发输出（TRGO）。 <p>若需要在同一时间启动多个定时器，或者要实现在一定的时间窗口内使能从定时器，那么这种模式将非常有用。</p> <p>当 CNTEN 控制位被置位，或当从模式下输入有效信号时，计数器使能信号将会被置位。</p> <p>当计数器使能信号受控于触发输入时，TRGO 比触发输入信号有一个延迟，除非选择了主/从模式，详述参看寄存器 TIM_TGICR0 的 MSM 位。</p> <p>010：更新事件</p> <ul style="list-style-type: none"> 更新事件（上溢更新事件、下溢更新事件和软件更新事件）被选为触发输出（TRGO）。 <p>011：通道 CC0 的捕获/比较脉冲</p> <ul style="list-style-type: none"> 一旦发生一次捕获或一次比较成功，送出一个正脉冲。 <p>100：通道 CC0 的基准参考信号 OC0REF</p> <p>101：通道 CC1 的基准参考信号 OC1REF</p> <p>110：通道 CC2 的基准参考信号 OC2REF</p> <p>111：通道 CC3 的基准参考信号 OC3REF</p>

参数名	比特位	属性	复位值	描述
CCUS	6	W/R	0	捕获/比较控制位更新条件： 0：如果捕获/比较控制位是预装载的 (CCPE =1)，只能通过设置 COM 位更新。 1：如果捕获/比较控制位是预装载的 (CCPE =1)，可以通过设置 COM 位或 TRGI 的有效沿更新。
CCPE	7	W/R	0	捕获/比较控制位预装载使能控制寄存器： 0：捕获/比较控制位 CCxE, CCxNE, CCxP, CCxNP 和 OCxMS 不是预装载的。 1：捕获/比较控制位 CCxE, CCxNE, CCxP, CCxNP 和 OCxMS 位是预装载的，设置该位后，这些寄存器只在设置了 COM 位或 TRGI 的有效沿更新。

10.6.2 TIM0 控制寄存器 1 (TIM0_CONR1) [0xD2]

Table 106 TIM0 控制寄存器 1 (TIM0_CONR1)

参数名	比特位	属性	复位值	描述
OIS0	0	W/R	0	通道 CC0 输出的空闲状态 (OC0 输出状态)： 0：当 MOE=0 时 (空闲状态)，死区后 OC0=0。 1：当 MOE=0 时，死区后 OC0=1。 注：若设置了 LOCK (TIM0_BRKC 寄存器) 级别 1、2 或 3，该位不能被修改。
OIS0N	1	W/R	0	通道 CC0 输出的空闲状态 (OC0N 输出状态)： 0：当 MOE=0 时，死区后 OC0N=0； 1：当 MOE=0 时，死区后 OC0N=1； 注：若设置了 LOCK (TIM0_BKR 寄存器) 级别 1、2 或 3，该位不能被修改。
OIS1	2	W/R	0	通道 CC1 输出的空闲状态 (OC1 输

参数名	比特位	属性	复位值	描述
				出状态), 参见 OIS0。
OIS1N	3	W/R	0	通道 CC1 输出的空闲状态 (OC1N 输出状态), 参见 OIS0N。
OIS2	4	W/R	0	通道 CC2 输出的空闲状态 (OC2 输出状态), 参见 OIS0。
OIS2N	5	W/R	0	通道 CC2 输出的空闲状态 (OC2N 输出状态), 参见 OIS0N。
OIS3	6	W/R	0	通道 CC3 输出的空闲状态 (OC3 输出状态), 参见 OIS0。
RESERVED	7	R	0	保留。

10.6.3 TIM0 控制寄存器 2 (TIM0_CONR2) [0xD3]

Table 107 TIM0 控制寄存器 2 (TIM0_CONR2)

参数名	比特位	属性	复位值	描述
ARPLE	0	W/R	0	计数器周期值预装载使能位: 0: 关闭 TIM0_ARR 寄存器预装载功能。 1: 打开 TIM0_ARR 寄存器预装载功能。
CNTMC	2-1	W/R	0	计数器模式控制: 00: 边沿对齐模式, 计数器依据方向位 CNTDIR 的配置向上或向下计数。 01: 中央对齐模式 1, 计数器交替向上向下计数, 只在计数器向下计数时更新输出通道的输出比较中断标志位。 10: 中央对齐模式 2, 计数器交替向上向下计数, 只在计数器向上计数时更新输出通道的输出比较中断标志位。 11: 中央对齐模式 3, 计数器交替向上向下计数, 在计数器向上计数和向下计数时更新输出通道的输出比较中断标志位。 注: 当计数器正在计数时, 不允许从边沿对齐模式转换到中央对齐模式。
CNTDIR	3	W/R	0	计数器计数方向:

参数名	比特位	属性	复位值	描述
				0: 计数器向上计数。 1: 计数器向下计数。 注: 当计数器配置为中央对齐模式或编码器模式时, 该位只读。
OPM	4	W/R	0	单脉冲模式配置寄存器: 0: 不是单脉冲模式, 在发生更新事件时, 计数器不停止。 1: 配置为单脉冲模式, 在发生下一次更新事件时, 计数器停止。
URC	5	W/R	0	更新源选择寄存器, 软件通过该位选择更新事件的源: 0: 下述任一事件产生一个更新事件, 如果更新中断使能, 则同时产生一个更新中断: --计数器上溢/下溢 --设置 TIM_EGR 寄存器的 UG 位 --从模式控制器产生的复位信号 1: 下述任一事件产生一个更新事件, 如果更新中断使能, 只有计数器上溢/下溢时产生一个更新中断: --计数器上溢/下溢 --设置 TIM_EGR 寄存器的 UG 位 --从模式控制器产生的复位信号。
RESERVED	7-6	W/R	0	保留。

10.6.4 TIM0 控制寄存器 3 (TIM0_CONR3) [0xD4]

Table 108 TIM0 模式选择寄存器 3(TIM0_CONR3)

参数名	比特位	属性	复位值	描述
UPOUDIS	0	W/R	0	上溢无效寄存器: 0: 当计数器上溢时产生更新事件。 1: 当计数器上溢时不产生更新事件。
DOWNOUDIS	1	W/R	0	下溢无效寄存器: 0: 当计数器下溢时产生更新事件。

参数名	比特位	属性	复位值	描述
				1: 当计数器下溢时不产生更新事件。
RESERVED	4-2	R	0	保留
LOCK	6-5	W/R	0	寄存器锁定设置, 该位为防止软件错误配置而提供写保护。 00: 锁定关闭, 寄存器无写保护。 01: 锁定级别 1, 不能写入 TIM0_DTG 寄存器、TIM0_BRKC 的 BRKE、BRKP、AOE 位、TIM0_CONR2 寄存器。 10: 锁定级别 2, 不能写入锁定级别 1 中的各位, 也不能写入 TIM0_CCPS 寄存器, 不能写入 TIM0_BRKC 的 ROSS 和 IOSS 位。 11: 锁定级别 3, 不能写入锁定级别 2 中的所有寄存器, 不能写入 TIM0_CCxMR 的 OCxMS 和 OCxPEN 位。 注: 只有 LOCK 值为 0 时才可以配置为其他非 0 值, 其内容冻结直到高级定时器被复位。
RESERVED	7	R	0	保留

10.6.5 从模式控制寄存器 0 (TIM0_TGICR0) [0xD5]

Table 109 从模式控制寄存器 0(TIM0_TGICR0)

参数名	比特位	属性	复位值	描述
MSM	0	R	0	主从模式寄存器: 0: TIM0 和 TIM1 独立启动。 1: TIM0 处于主模式下, TIM0 和 TIM1 同时启动。
TRGS	3-1	W/R	0	从模式控制器触发信号选择寄存器: 000: 通道 CC0 的边沿检测信号 CH0INFP_ED;

参数名	比特位	属性	复位值	描述
				001: 通道 CC0 滤波后的定时器输入 CH0INFP; 010: 通道 CC1 滤波后的定时器输入 CH1INFP; 100: 外部触发输入 PWM_ETR。 其他: 保留
SMS	6-4	W/R	0	从模式选择寄存器: 000: 关闭从模式 <ul style="list-style-type: none"> 通过配置计数器能寄存器 (TIM0_CNTEN.CNTEN) 启动计数器。 001: 编码器模式 1 <ul style="list-style-type: none"> 根据 CH0INFP0 的电平, 计数器在 CH1INFP1 的边沿向上/向下计数。 010: 编码器模式 2 <ul style="list-style-type: none"> 根据 CH1INFP1 的电平, 计数器在 CH0INFP0 的边沿向上/向下计数。 011: 编码器模式 3 <ul style="list-style-type: none"> 根据 CH1INFP1 和 CH0INFP0 的电平, 计数器在 CH0INFP0 和 CH1INFP1 的边沿向上/向下计数。 100: 复位模式 <ul style="list-style-type: none"> 选中的触发输入的有效沿重新初始化计数器, 并且产生一个更新事件。 101: 门控模式 <ul style="list-style-type: none"> 当触发输入为有效电平时, 计数器计数。一旦触发输入变为无效电平, 计数器停止计数

参数名	比特位	属性	复位值	描述
				<p>(但不复位)。</p> <p>110: 触发模式</p> <ul style="list-style-type: none"> 计数器在触发输入的有效沿启动 (但不复位), 仅控制计数器启动。 <p>111: 外部时钟模式 1</p> <ul style="list-style-type: none"> 选中的触发输入的有效沿驱动计数器计数。 <p>注: 如果 CH1INFP1_ED 被选为触发输入时, 不要使用门控模式, 这是因为 CH1INFP1_ED 是一个脉冲信号, 而门控模式是要检查触发输入的电平。</p>
RESERVED	7	R	0	保留

10.6.6 从模式控制寄存器 1 (TIM0_TGICR1) [0xD6]

Table 110 从模式控制寄存器 1 (TIM0_TGICR1)

参数名	比特位	属性	复位值	描述
EXTP	0	W/R	0	<p>外部触发信号极性选择:</p> <p>0: PWM0_ETR 不反相, 高电平或上升沿有效。</p> <p>1: PWM0_ETR 被反相, 低电平或下降沿有效。</p>
EXTCLKEN	1	W/R	0	<p>外部时钟模式 2 使能:</p> <p>0: 禁止外部时钟模式 2</p> <p>1: 启动外部时钟模式 2, 计数器由 PWM0_ETRF 信号的有效沿驱动。</p> <p>注 1: 设置 EXTCLKEN 位与选择外部时钟模式 1 并将 TRGI 连到 PWM0_ETRF 具有相同功效。</p> <p>注 2: 下述从模式可以与外部时钟模式 2 同时使用: 复位模式、门控模式和触发模式; 但是这时 TRGI 不能连到 PWM0_ETRF。</p>
EXTDIV	3-2	W/R	0	外部触发信号 PWM0_ETR 预分频:

参数名	比特位	属性	复位值	描述
				00: 1 分频 01: 2 分频 10: 4 分频 11: 8 分频
RESERVED	7-4	R	0	保留

10.6.7 TIM0 中断使能寄存器 0 (TIM0_IER0) [0xD7]

Table 111 TIM0 中断使能寄存器 0 (TIM0_IER0)

参数名	比特位	属性	复位值	描述
BRKIE	0	W/R	0	刹车中断使能: 0: 禁止刹车中断 1: 允许刹车中断
TRIE	1	W/R	0	触发中断使能: 0: 禁止触发中断 1: 允许触发中断
COMIE	2	W/R	0	COM 中断使能: 0: 禁止 COM 中断 1: 允许 COM 中断
CC0IE	3	W/R	0	互补模式下通道 CC0 捕获/比较中断使能: 0: 禁止捕获/比较中断 1: 允许捕获/比较中断
CC1IE	4	W/R	0	互补模式下通道 CC1 捕获/比较中断使能: 0: 禁止捕获/比较中断 1: 允许捕获/比较中断
CC2IE	5	W/R	0	互补模式下通道 CC2 捕获/比较中断使能: 0: 禁止捕获/比较中断 1: 允许捕获/比较中断
CC3IE	6	W/R	0	互补模式下通道 CC3 捕获/比较中断使能: 0: 禁止捕获/比较中断 1: 允许捕获/比较中断
RESERVED	7	R	0	保留

10.6.8 TIM0 中断使能寄存器 1 (TIM0_IER1) [0xFB]

Table 112 TIM0 中断使能寄存器 1 (TIM0_IER1)

参数名	比特位	属性	复位值	描述
UPOIE	0	W/R	0	上溢中断使能寄存器： 0：当计数器上溢时不允许产生更新中断。 1：当计数器上溢时允许产生更新中断。
DOWMOIE	1	W/R	0	下溢中断使能寄存器： 0：当计数器下溢时不允许产生更新中断。 1：当计数器下溢时允许产生更新中断。
SOFTUPIE	2	W/R	0	复位更新中断使能寄存器： 0：不允许产生复位更新中断。 1：当产生以下更新事件时允许产生更新中断： <ul style="list-style-type: none"> ● 设置 TIM0_EGR 寄存器的 UG 位为 1。 ● 从模式控制器产生的复位信号。
RESERVED	7-3	R	0	保留

10.6.9 TIM0 通道 CC0 输入捕获/输出比较模式控制寄存器 (TIM0_CC0MR) [0xD9]

Table 113 TIM0 通道 CC0 输入捕获/输出比较模式控制寄存器（输出比较模式）
(TIM0_CC0MR)

参数名	比特位	属性	复位值	描述
CC0MS	1-0	W/R	00	捕获/比较功能选择： 00：CC0 通道被配置为输出比较功能； 01：CC0 通道被配置为输入捕获功能，输入捕获信号 CH0INC 来自通道 CC0 的输入信号 CH0IN。 10：CC0 通道被配置为输入捕获功能，输入捕获信号 CH0INC 来自通道 CC1 的输入信号 CH1IN。 11：CC0 通道被配置为输入捕获功能，输入捕获信号 CH0INC 来自 TRC。 注：CC0MS 仅在通道关闭（寄存器

参数名	比特位	属性	复位值	描述
				TIMx_CCENR 的 CC0EN=0) 时才是可写的，因此在使用通道 CC0 时，应该先配置 CC0MS，再配置 CC0EN 为 1。
RESERVED	2	R	0	保留
OC0PEN	3	W/R	0	<p>通道 CC0 的输出比较寄存器的预装载使能：</p> <p>0: 禁止寄存器 TIM_CC0R 的预装载功能，可随时将数据写入寄存器 TIM_CC0R，且新值立即生效。</p> <p>1: 开启寄存器 TIM_CC0R 的预装载功能，读写操作仅对预装载寄存器操作，TIM_CC0R 的预装载值在更新事件产生时被载入影子寄存器。</p> <p>注 1：一旦 LOCK 级别设为 3，并且通道 CC0 被配置为输出比较模式时，该位不能被修改。</p> <p>注 2：若未使能输出比较寄存器的预装载功能，当改变输出比较寄存器的值时，PWM 模式下可能会出现不规则的输出。单脉冲模式下的 PWM 输出不会出现这种情况。</p>
OC0MS	6-4	W/R		<p>输出模式选择寄存器，该位定义了输出参考信号 OC0REF 的动作，而 OC0REF 决定了 OC0、OC0N 的输出，从而决定输出管脚上的 PWM 输出波形。OC0REF 为高电平有效，而 OC0 和 OC0N 的有效电平取决于 CC0P 和 CC0NP 位的配置。</p> <p>000: 冻结</p> <ul style="list-style-type: none"> 输出比较寄存器 TIM0_CC0RU 与计数器间的比较结果对 OC0REF 不起作用。 <p>001: 匹配时输出有效电平</p>

参数名	比特位	属性	复位值	描述
				<ul style="list-style-type: none"> 当输出比较寄存器 TIM0_CC0R 与计数器的值相同时, OC0REF 输出有效电平 (即高电平)。 <p>010: 匹配时输出无效电平</p> <ul style="list-style-type: none"> 当输出比较寄存器 TIM0_CC0R 与计数器的值相同时, OC0REF 输出无效电平 (即低电平)。 <p>011: 翻转</p> <ul style="list-style-type: none"> 当输出比较寄存器 TIM0_CC0R 与计数器的值相同时, OC0REF 信号翻转。 <p>100: 强制输出无效电平</p> <ul style="list-style-type: none"> 当配置此种输出模式时, 立即将 OC0REF 设置为无效电平。 <p>101: 强制输出有效电平</p> <ul style="list-style-type: none"> 当配置此种输出模式时, 立即将 OC0REF 设置为有效电平。 <p>110: PWM 模式 1</p> <ul style="list-style-type: none"> 边沿计数模式下: 向上计数时, 若 $TIM0_CNT < TIM0_CC0R$, OC0REF 为有效电平 (OC0REF=1), 否则为无效电平 (OC0REF=0); 向下计数时, 若 $TIM0_CNT > TIM0_CC0R$, OC0REF 为无效电平 (OC0REF=0), 否则为有效电平 (OC0REF=1)。 中央对齐计数模式下: 向上计数时, 若 $TIM0_CNT < TIM0_CC0R$, OC0REF 为有效电平 (OC0REF=1), 否则为无效电平 (OC0REF=0); 向下计数时, 若 $TIM0_CNT > TIM0_CC0R$, OC0REF 为无效电平 (OC0REF=0), 否则为有效电平 (OC0REF=1)。

参数名	比特位	属性	复位值	描述
				<p>(OC0REF=1)。</p> <p>111: PWM 模式 2</p> <ul style="list-style-type: none"> 边沿计数模式下: 向上计数时, 若 $TIM0_CNT < TIM0_CC0R$, OC0REF 为无效电平 (OC0REF=0), 否则为有效电平 (OC0REF=1); 向下计数时, 若 $TIM_CNT0 > TIM0_CC0R$, OC0REF 为有效电平 (OC0REF=1), 否则为无效电平 (OC0REF=0)。 中央对齐计数模式下: 向上计数时, 若 $TIM0_CNT < TIM0_CC0R$, OC0REF 为无效电平 (OC0REF=0), 否则为有效电平 (OC0REF=1); 向下计数时, 若 $TIM0_CNT > TIM0_CC0R$, OC0REF 为有效电平 (OC0REF=1), 否则为无效电平 (OC0REF=0)。
OC0CE	7	W/R	0	<p>比较输出 OC0REF 清除控制:</p> <p>0: OC0REF 不受 PWM0_ETRF 输入的影响。</p> <p>1: PWM0_ETRF 输入有效电平时, OC0REF 输出无效电平。</p>

PWM 通道 CC0 输入捕获/输出比较模式控制寄存器 (输入捕获模式) (TIM0_CC0MR)

参数名	比特位	属性	复位值	描述
CC0MS	1-0	W/R	00	<p>捕获/比较功能选择寄存器:</p> <p>00: CC0 通道被配置为输出比较功能;</p> <p>01: CC0 通道被配置为输入捕获功能, 输入捕获信号 CH0INC 来自通道 CC0 的输入信号 CH0IN。</p> <p>10: CC0 通道被配置为输入捕获功能, 输入捕获信号 CH0INC 来自通道 CC1 的输入信号 CH1IN。</p>

参数名	比特位	属性	复位值	描述
				11: CC0 通道被配置为输入捕获功能, 输入捕获信号 CH0INC 来自 TRC。 注: CC0MS 仅在通道关闭 (寄存器 TIMx_CCENR 的 CC0EN=0) 时才是可写的, 因此在使用通道 CC0 时, 应该先配置 CC0MS, 再配置 CC0EN 为 1。
IC0PDIV	3-2	W/R	0	通道 CC0 输入捕获信号预分频控制寄存器, 这两位定义了 CC0 输入信号的预分频系数。 00: 不分频。 01: 每 2 个事件触发一次捕获。 10: 每 4 个事件触发一次捕获。 11: 每 8 个事件触发一次捕获。
RESERVED	7-4	R	0	保留

10.6.10TIM0 通道 CC1 输入捕获/输出比较模式控制寄存器 (TIM0_CC1MR)

[0xDA]

Table 114 TIM0 通道 CC1 输入捕获/输出比较模式控制寄存器 (输出比较模式)

(TIM0_CC1MR)

参数名	比特位	属性	复位值	描述
CC1MS	1-0	W/R	00	捕获/比较功能选择寄存器: 00: CC1 通道被配置为输出比较功能; 01: CC1 通道被配置为输入捕获功能, 输入捕获信号 CH1INC 来自通道 CC1 的输入信号 CH1IN。 10: CC1 通道被配置为输入捕获功能, 输入捕获信号 CH1INC 来自通道 CC0 的输入信号 CH0IN。 11: CC1 通道被配置为输入捕获功能, 输入捕获信号 CH1INC 来自 TRC。 注: CC1MS 仅在通道关闭 (寄存器 TIMx_CCENR 的 CC1E=0) 时才是可写的, 因此在使用通道 CC1 时, 应该先配

参数名	比特位	属性	复位值	描述
				置 CC1MS，再配置 CC1E 为 1。
RESERVED	2	R	0	保留
OC1PEN	3	W/R	0	<p>输出比较寄存器的预装载使能控制寄存器：</p> <p>0：禁止寄存器 TIM0_CC1R 的预装载功能，可随时将数据写入寄存器 TIM0_CC1R，且新值立即生效。</p> <p>1：开启寄存器 TIM0_CC1R 的预装载功能，读写操作仅对预装载寄存器操作，TIM0_CC1R 的预装载值在更新事件产生时被载入影子寄存器中。</p> <p>注 1：一旦 LOCK 级别设为 3，并且通道被配置为输出比较模式时，该位不能被修改。</p> <p>注 2：若未使能输出比较寄存器的预装载功能，当改变输出比较寄存器的值时，PWM 模式下可能会出现不规则的输出。单脉冲模式下的 PWM 输出不会出现这种情况。</p>
OC1MS	6-4	W/R		<p>输出模式选择寄存器，该位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1、OC1N 的输出，从而决定输出管脚上的 PWM 输出波形。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平取决于 CC1P 和 CC1NP 位的配置。</p> <p>000：冻结。</p> <ul style="list-style-type: none"> ● 输出比较寄存器 TIM0_CC1R 与计数器间的比较结果对 OC1REF 不起作用。 <p>001：匹配时输出有效电平。</p> <ul style="list-style-type: none"> ● 当输出比较寄存器 TIM0_CC1R 与计数器的值相同时，设置 OC1REF 为

参数名	比特位	属性	复位值	描述
				<p>有效电平（即高电平）。</p> <p>010：匹配时输出无效电平。</p> <ul style="list-style-type: none"> 当输出比较寄存器 TIM0_CC1R 与计数器的值相同时，设置 OC1REF 为无效电平（即低电平）。 <p>011：翻转。</p> <ul style="list-style-type: none"> 当输出比较寄存器 TIM0_CC1R 与计数器的值相同时，OC1REF 信号翻转。 <p>100：强制输出无效电平。</p> <ul style="list-style-type: none"> 当配置此种输出模式时，立即将 OC1REF 设置为无效电平。 <p>101：强制输出有效电平。</p> <ul style="list-style-type: none"> 当配置此种输出模式时，立即将 OC1REF 设置为有效电平。 <p>110：PWM 模式 1</p> <ul style="list-style-type: none"> 边沿计数模式下：向上计数时，若 $TIM0_CNT < TIM0_CC1R$，OC1REF 为有效电平（OC1REF=1），否则为无效电平（OC1REF=0）；向下计数时，若 $TIM0_CNT > TIM0_CC1R$，OC1REF 为无效电平（OC1REF=0），否则为有效电平（OC1REF=1）。 中央对齐计数模式下：向上计数时，若 $TIM0_CNT < TIM0_CC1R$，OC1REF 为有效电平（OC1REF=1），否则为无效电平（OC1REF=0）；向下计数时，若 $TIM0_CNT > TIM0_CC1R$，OC1REF 为无效电平（OC1REF=0），否则为有效电平（OC1REF=1）。

参数名	比特位	属性	复位值	描述
				<p>(OC1REF=1)。</p> <p>111: PWM 模式 2</p> <ul style="list-style-type: none"> 边沿计数模式下: 向上计数时, 若 $TIM0_CNT < TIM0_CC1R$, OC1REF 为无效电平 (OC1REF=0), 否则为有效电平 (OC1REF=1); 向下计数时, 若 $TIM0_CNT > TIM0_CC1R$, OC1REF 为有效电平 (OC1REF=1), 否则为无效电平 (OC1REF=0)。 中央对齐计数模式下: 向上计数时, 若 $TIM0_CNT < TIM0_CC1R$, OC1REF 为无效电平 (OC1REF=0), 否则为有效电平 (OC1REF=1); 向下计数时, 若 $TIM0_CNT > TIM0_CC1R$, OC1REF 为有效电平 (OC1REF=1), 否则为无效电平 (OC1REF=0)。
OC1CE	7	W/R	0	<p>比较输出 OC1REF 清除控制寄存器:</p> <p>0: OC1REF 不受 PWM0_ETRF 输入的影响。</p> <p>1: PWM0_ETRF 输入有效电平时, 设置 OC1REF 为无效电平。</p>

PWM 通道 CC1 输入捕获/输出比较模式控制寄存器 (输入捕获模式) (TIM0_CC1MR)

参数名	比特位	属性	复位值	描述
CC1MS	1-0	W/R	00	<p>捕获/比较功能选择寄存器:</p> <p>00: CC1 通道被配置为输出比较功能;</p> <p>01: CC1 通道被配置为输入捕获功能, 输入捕获信号 CH0INC 来自通道 CC1 的输入信号 CH1IN。</p> <p>10: CC1 通道被配置为输入捕获功能, 输入捕获信号 CH1INC 来自通道 CC0 的输</p>

参数名	比特位	属性	复位值	描述
				入信号 CH0IN。 11: CC1 通道被配置为输入捕获功能，输入捕获信号 CH1INC 来自 TRC。 注：CC1MS 仅在通道关闭（寄存器 TIMx_CCENR 的 CC1E=0）时才是可写的，因此在使用通道 CC1 时，应该先配置 CC1MS，再配置 CC1E 为 1。
IC1PDIV	3-2	W/R	0	通道 CC1 输入捕获信号预分频控制寄存器，这两位定义了 CC1 输入信号的预分频系数。 00: 不分频。 01: 每 2 个事件触发一次捕获。 10: 每 4 个事件触发一次捕获。 11: 每 8 个事件触发一次捕获。
RESERVED	7-4	R	0	保留

10.6.11TIM0 通道 CC2 输入捕获/输出比较模式控制寄存器（TIM0_CC2MR） [0xDB]

Table 115 TIM0 通道 CC2 输入捕获/输出比较模式控制寄存器（输出比较模式）
(TIM0_CC2MR)

参数名	比特位	属性	复位值	描述
CC2MS	1-0	W/R	00	捕获/比较功能选择寄存器： 00: CC2 通道被配置为输出比较功能； 01: CC2 通道被配置为输入捕获功能，输入捕获信号 CH2INC 来自通道 CC2 的输入信号 CH2IN。 10: CC2 通道被配置为输入捕获功能，输入捕获信号 CH2INC 来自通道 CC3 的输入信号 CH3IN。 11: CC2 通道被配置为输入捕获功能，输入捕获信号 CH2INC 来自 TRC。 注：CC2MS 仅在通道关闭（寄存器 TIMx_CCENR 的 CC2E=0）时才是可写

参数名	比特位	属性	复位值	描述
				的，因此在使用通道 CC2 时，应该先配置 CC2MS，再配置 CC2E 为 1。
RESERVED	2	R	0	保留
OC2PEN	3	W/R	0	<p>输出比较寄存器的预装载使能控制寄存器：</p> <p>0：禁止寄存器 TIM0_CC2R 的预装载功能，可随时将数据写入寄存器 TIM0_CC2R，且新值立即生效。</p> <p>1：开启寄存器 TIM0_CC2R 的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CC2R 的预装载值在更新事件产生时被载入影子寄存器中。</p> <p>注 1：一旦 LOCK 级别设为 3，并且通道被配置为输出比较模式时，该位不能被修改。</p> <p>注 2：若未使能输出比较寄存器的预装载功能，当改变输出比较寄存器的值时，PWM 模式下可能会出现不规则的输出。单脉冲模式下的 PWM 输出不会出现这种情况。</p>
OC2MS	6-4	W/R		<p>输出模式选择寄存器，该位定义了输出参考信号 OC2REF 的动作，而 OC2REF 决定了 OC2、OC2N 的输出，从而决定输出管脚上的 PWM 输出波形。OC2REF 为高电平有效，而 OC2 和 OC2N 的有效电平取决于 CC2P 和 CC2NP 位的配置。</p> <p>000：冻结。</p> <ul style="list-style-type: none"> ● 输出比较寄存器 TIM0_CC2R 与计数器间的比较结果对 OC2REF 不起作用。 <p>001：匹配时输出有效电平</p> <ul style="list-style-type: none"> ● 当输出比较寄存器 TIM0_CC2R 与计

参数名	比特位	属性	复位值	描述
				<p>计数器的值相同时，设置 OC2REF 为有效电平（即高电平）。</p> <p>010：匹配时输出无效电平</p> <ul style="list-style-type: none"> 当输出比较寄存器 TIM0_CC2R 与计数器的值相同时，设置 OC2REF 为无效电平（即低电平）。 <p>011：翻转。</p> <ul style="list-style-type: none"> 当输出比较寄存器 TIM0_CC2R 与计数器的值相同时，OC2REF 信号翻转。 <p>100：强制输出无效电平。</p> <ul style="list-style-type: none"> 当配置此种输出模式时，立即将 OC2REF 设置为无效电平。 <p>101：强制输出有效电平。</p> <ul style="list-style-type: none"> 当配置此种输出模式时，立即将 OC2REF 设置为有效电平。 <p>110：PWM 模式 1</p> <ul style="list-style-type: none"> 边沿计数模式下：向上计数时，若 $TIM0_CNT < TIM0_CC2R$，OC2REF 为有效电平（OC2REF=1），否则为无效电平（OC2REF=0）；向下计数时，若 $TIM0_CNT > TIM0_CC2R$，OC2REF 为无效电平（OC2REF=0），否则为有效电平（OC2REF=1）。 中央对齐计数模式下：向上计数时，若 $TIM0_CNT < TIM0_CC2R$，OC2REF 为有效电平（OC2REF=1），否则为无效电平（OC2REF=0）；向下计数时，若 $TIM0_CNT > TIM0_CC2R$，OC2REF 为无效电平

参数名	比特位	属性	复位值	描述
				<p>(OC2REF=0), 否则为有效电平 (OC2REF=1)。</p> <p>111: PWM 模式 2</p> <ul style="list-style-type: none"> 边沿计数模式下: 向上计数时, 若 $TIM0_CNT < TIM0_CC2R$, OC2REF 为无效电平 (OC2REF=0), 否则为有效电平 (OC2REF=1); 向下计数时, 若 $TIM0_CNT > TIM0_CC2R$, OC2REF 为有效电平 (OC2REF=1), 否则为无效电平 (OC2REF=0)。 中央对齐计数模式下: 向上计数时, 若 $TIM0_CNT < TIM0_CC2R$, OC2REF 为无效电平 (OC2REF=0), 否则为有效电平 (OC2REF=1); 向下计数时, 若 $TIM0_CNT > TIM0_CC2R$, OC2REF 为有效电平 (OC2REF=1), 否则为无效电平 (OC2REF=0)。
OC2CE	7	W/R	0	<p>比较输出 OC2REF 清除控制寄存器:</p> <p>0: OC2REF 不受 PWM_ETRF 输入的影响。</p> <p>1: PWM0_ETRF 输入有效电平时, 设置 OC2REF 为无效电平。</p>

PWM 通道 CC2 输入捕获/输出比较模式控制寄存器 (输入捕获模式) (TIM0_CC2MR)

参数名	比特位	属性	复位值	描述
CC2MS	1-0	W/R	00	<p>捕获/比较功能选择寄存器:</p> <p>00: CC2 通道被配置为输出比较功能;</p> <p>01: CC2 通道被配置为输入捕获功能, 输入捕获信号 CH2INC 来自通道 CC2 的输入信号 CH2IN。</p> <p>10: CC2 通道被配置为输入捕获功能, 输入捕获信号 CH2INC 来自通道 CC3 的输入信号 CH3IN。</p>

参数名	比特位	属性	复位值	描述
				入信号 CH3IN。 11: CC2 通道被配置为输入捕获功能，输入捕获信号 CH2INC 来自 TRC。 注：CC2MS 仅在通道关闭（寄存器 TIM_CCENR 的 CC2E=0）时才是可写的，因此在使用通道 CC2 时，应该先配置 CC2MS，再配置 CC2E 为 1。
IC2PDIV	3-2	W/R	0	通道 CC2 输入捕获信号预分频控制寄存器，这两位定义了 CC2 输入信号的预分频系数。 00: 不分频。 01: 每 2 个事件触发一次捕获。 10: 每 4 个事件触发一次捕获。 11: 每 8 个事件触发一次捕获。
RESERVED	7-4	R	0	保留

10.6.12TIM0 通道 CC3 输入捕获/输出比较模式控制寄存器（TIM0_CC3MR） [0xDC]

Table 116 TIM0 通道 CC3 输入捕获/输出比较模式控制寄存器（输出比较模式）
(TIM0_CC3MR)

参数名	比特位	属性	复位值	描述
CC3MS	1-0	W/R	00	捕获/比较功能选择寄存器： 00: CC3 通道被配置为输出比较功能； 01: CC3 通道被配置为输入捕获功能，输入捕获信号 CH3INC 来自通道 CC3 的输入信号 CH3IN。 10: CC3 通道被配置为输入捕获功能，输入捕获信号 CH3INC 来自通道 CC2 的输入信号 CH2IN。 11: CC3 通道被配置为输入捕获功能，输入捕获信号 CH3INC 来自 TRC。 注：CC3MS 仅在通道关闭（寄存器 TIM0_CCENR 的 CC3E=0）时才是可写

参数名	比特位	属性	复位值	描述
				的，因此在使用通道 CC3 时，应该先配置 CC3MS，再配置 CC3E 为 1。
RESERVED	2	R	0	保留
OC3PEN	3	W/R	0	<p>输出比较寄存器的预装载使能控制寄存器：</p> <p>0：禁止寄存器 TIM0_CC3R 的预装载功能，可随时将数据写入寄存器 TIM0_CC3R，且新值立即生效。</p> <p>1：开启寄存器 TIM0_CC3R 的预装载功能，读写操作仅对预装载寄存器操作，TIM0_CC3R 的预装载值在更新事件产生时被载入影子寄存器中。</p> <p>注 1：一旦 LOCK 级别设为 3，并且通道被配置为输出比较模式时，该位不能被修改。</p> <p>注 2：若未使能输出比较寄存器的预装载功能，当改变输出比较寄存器的值时，PWM 模式下可能会出现不规则的输出。单脉冲模式下的 PWM 输出不会出现这种情况。</p>
OC3MS	6-4	W/R		<p>输出模式选择寄存器，该位定义了输出参考信号 OC3REF 的动作，而 OC3REF 决定了 OC3、OC3N 的输出，从而决定输出管脚上的 PWM 输出波形。OC3REF 为高电平有效，而 OC3 和 OC3N 的有效电平取决于 CC3P 和 CC3NP 位的配置。</p> <p>000：冻结。</p> <ul style="list-style-type: none"> ● 输出比较寄存器 TIM0_CC3R 与计数器间的比较结果对 OC3REF 不起作用。 <p>001：匹配时输出有效电平</p> <ul style="list-style-type: none"> ● 当输出比较寄存器 TIM0_CC3R 与计

参数名	比特位	属性	复位值	描述
				<p>计数器的值相同时，设置 OC3REF 为有效电平（即高电平）。</p> <p>010：匹配时输出无效电平</p> <ul style="list-style-type: none"> 当输出比较寄存器 TIM0_CC3R 与计数器的值相同时，设置 OC3REF 为无效电平（即低电平）。 <p>011：翻转。</p> <ul style="list-style-type: none"> 当输出比较寄存器 TIM0_CC3R 与计数器的值相同时，OC3REF 信号翻转。 <p>100：强制输出无效电平。</p> <ul style="list-style-type: none"> 当配置此种输出模式时，立即将 OC3REF 设置为无效电平。 <p>101：强制输出有效电平。</p> <ul style="list-style-type: none"> 当配置此种输出模式时，立即将 OC3REF 设置为有效电平。 <p>110：PWM 模式 1</p> <ul style="list-style-type: none"> 边沿计数模式下：向上计数时，若 $TIM0_CNT < TIM0_CC3R$，OC3REF 为有效电平（OC3REF=1），否则为无效电平（OC3REF=0）；向下计数时，若 $TIM0_CNT > TIM0_CC3R$，OC3REF 为无效电平（OC3REF=0），否则为有效电平（OC3REF=1）。 中央对齐计数模式下：向上计数时，若 $TIM0_CNT < TIM0_CC3R$，OC3REF 为有效电平（OC3REF=1），否则为无效电平（OC3REF=0）；向下计数时，若 $TIM0_CNT > TIM0_CC3R$，OC2REF 为无效电平

参数名	比特位	属性	复位值	描述
				<p>(OC3REF=0), 否则为有效电平 (OC3REF=1)。</p> <p>111: PWM 模式 2</p> <ul style="list-style-type: none"> 边沿计数模式下: 向上计数时, 若 $TIM0_CNT < TIM0_CC3R$, OC3REF 为无效电平 (OC3REF=0), 否则为有效电平 (OC3REF=1); 向下计数时, 若 $TIM0_CNT > TIM0_CC3R$, OC3REF 为有效电平 (OC3REF=1), 否则为无效电平 (OC3REF=0)。 中央对齐计数模式下: 向上计数时, 若 $TIM0_CNT < TIM0_CC3R$, OC3REF 为无效电平 (OC3REF=0), 否则为有效电平 (OC3REF=1); 向下计数时, 若 $TIM0_CNT > TIM0_CC3R$, OC3REF 为有效电平 (OC3REF=1), 否则为无效电平 (OC3REF=0)。
OC3CE	7	W/R	0	<p>比较输出 OC3REF 清除控制寄存器:</p> <p>0: OC3REF 不受 PWM0_ETRF 输入的影响。</p> <p>1: PWM0_ETRF 输入有效电平时, 设置 OC3REF 为无效电平。</p>

PWM 通道 CC3 输入捕获/输出比较模式控制寄存器 (输入捕获模式) (TIM_CC3MR)

参数名	比特位	属性	复位值	描述
CC3MS	1-0	W/R	00	<p>捕获/比较功能选择寄存器:</p> <p>00: CC3 通道被配置为输出比较功能;</p> <p>01: CC3 通道被配置为输入捕获功能, 输入捕获信号 CH3INC 来自通道 CC3 的输入信号 CH3IN。</p> <p>10: CC3 通道被配置为输入捕获功能, 输入捕获信号 CH3INC 来自通道 CC2 的输</p>

参数名	比特位	属性	复位值	描述
				入信号 CH2IN。 11: CC3 通道被配置为输入捕获功能，输入捕获信号 CH3INC 来自 TRC。 注：CC3MS 仅在通道关闭（寄存器 TIM_CCENR 的 CC3E=0）时才是可写的，因此在使用通道 CC3 时，应该先配置 CC3MS，再配置 CC3E 为 1。
IC3PDIV	3-2	W/R	0	通道 CC3 输入捕获信号预分频控制寄存器，这两位定义了 CC3 输入信号的预分频系数。 00: 不分频。 01: 每 2 个事件触发一次捕获。 10: 每 4 个事件触发一次捕获。 11: 每 8 个事件触发一次捕获。
RESERVED	7-4	R	0	保留

10.6.13 通道使能控制寄存器 (TIM0_CCENR) [0xDD]

Table 117 通道使能控制寄存器(TIM0_CCENR)

参数名	比特位	属性	复位值	描述
CC0EN	0	W/R	0	通道 CC0 输入捕获/输出比较使能： CC0 通道配置为输出时： 0: 关闭 <ul style="list-style-type: none"> OC0 禁止输出，输出引脚 PWM0_CH0 的信号受寄存器 MOE、IOSS、ROSS、OIS0、OIS0N、CC0NEN、CC0P 位的控制。 1: 开启 <ul style="list-style-type: none"> OC0 信号输出到对应的输出引脚 PWM0_CH0，其输出受寄存器 MOE、OSSI、OSSR、OIS0、OIS0N、CC0NEN、CC0P 的控制。 CC0 通道配置为输入时：

参数名	比特位	属性	复位值	描述
				0: 捕获禁止。 1: 捕获使能 ● 当捕获信号产生有效沿时, 将计数器的值捕获到 TIM0_CC0R 中。
CC0NEN	1	W/R	0	通道 CC0 互补路输出使能控制: 0: 关闭 ● OC0N 禁止输出, 输出引脚 PWM0_CH0N 的信号受寄存器 MOE、IOSS、ROSS、OIS0、OIS0N、CC0EN、CC0NP 位的控制。 1: 开启 ● OC0N 信号输出到对应的输出引脚 PWM0_CH0N, 其输出受寄存器 MOE、OSSI、OSSR、OIS0、OIS0N、CC0EN、CC0NP 的控制。
CC1EN	2	W/R	0	通道 CC1 输入捕获/输出比较使能控制, 参见 CC0EN 描述。
CC1NEN	3	W/R	0	通道 CC1 互补输出使能控制, 参见 CC0NEN 描述。
CC2EN	4	W/R	0	通道 CC2 输入捕获/输出比较使能控制, 参见 CC0EN 描述。
CC2NEN	5	W/R	0	通道 CC2 互补输出使能控制, 参见 CC0NEN 描述。
CC3EN	6	W/R	0	通道 CC3 输入捕获/输出比较使能控制, 参见 CC0EN 描述。
RESERVED	7	R	0	保留

10.6.14 通道极性控制寄存器 (TIM0_CCPS) [0xDE]

Table 118 通道极性控制寄存器(TIM0_CCPS)

参数名	比特位	属性	复位值	描述
CC0P	0	W/R	0	通道 CC0 的极性: ● 通道 CC0 配置为输出时:

参数名	比特位	属性	复位值	描述
				<p>0: OC0 (PWM0_CH0) 的高电平为有效电平。</p> <p>1: OC0 (PWM0_CH0) 的低电平为有效电平。</p> <p>● CC0 通道配置为输入时:</p> <p>该位选择是 CH0IN 还是 CH0IN 的反相信号作为触发或捕获信号。</p> <p>0: 不反相: CH0IN 的上升沿作为有效边沿, CH0IN 的高电平作为有效电平。</p> <p>1: 反相: CH0IN 的下降沿作为有效边沿, CH0IN 的低电平作为有效电平。</p> <p>注: 一旦 LOCK 级别设置为 2 或 3, 该位不能被修改。</p>
CC0NP	1	W/R	0	<p>通道 CC0 互补输出极性:</p> <p>0: OC0N (PWM0_CH0N) 的高电平为有效电平。</p> <p>1: OC0N (PWM0_CH0N) 的低电平为有效电平。</p> <p>注: 一旦 LOCK 级别设置为 2 或 3 或 CC3MS =0 时, 则该位不能被修改。</p>
CC1P	2	W/R	0	通道 CC1 极性, 参见 CC0P 描述。
CC1NP	3	W/R	0	通道 CC1 互补输出极性控制寄存器, 参见 CC0NP 描述。
CC2P	4	W/R	0	通道 CC2 极性, 参见 CC0P 描述。
CC2NP	5	W/R	0	通道 CC2 互补输出极性, 参见 CC0NP 描述。
CC3P	6	W/R	0	通道 CC3 极性, 参见 CC0P 描述。
RESERVED	7	R	0	保留

10.6.15TIM0 死区时间寄存器 (TIM0_DTG) [0xDF]

Table 119 TIM0 死区时间寄存器(TIM0_DTG)

参数名	比特位	属性	复位值	描述
-----	-----	----	-----	----

参数名	比特位	属性	复位值	描述
DTG	7-0	W/R	0	<p>死区发生器死区时间设置寄存器, 这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间:</p> <p>若 $DTG[7:5] = "0xx"$: $DT = DTG[6:0] \times T_{dts_clk}$</p> <p>若 $DTG[7:5] = "10x"$: $DT = (64 + DTG[5:0]) \times 2 \times T_{dts_clk}$</p> <p>若 $DTG[7:5] = "110"$: $DT = (32 + DTG[4:0]) \times 8 \times T_{dts_clk}$</p> <p>若 $DTG[7:5] = "111"$: $DT = (32 + DTG[4:0]) \times 16 \times T_{dts_clk}$</p>

10.6.16 TIM0 周期值自动装载寄存器的低 8 位 (TIM0_ARRL) [0xE1]

Table 120 TIM0 周期值自动装载寄存器的低 8 位 (TIM0_ARRL)

参数名	比特位	属性	复位值	描述
ARRL	7-0	W/R	0	<p>周期值自动重载值的低 8 位。若寄存器 TIM0_CONR2 的 ARPLE 位配置为 1, 寄存器的预装载功能有效, 当产生更新事件时, 该寄存器的值被装载到影子寄存器, 更新事件包括:</p> <ul style="list-style-type: none"> 计数器上溢/下溢 配置寄存器 TIM0_EGR 的 UG 位为 1 从模式控制器产生的复位信号 <p>当自动重载的值为 0 时, 计数器不工作。</p>

10.6.17 TIM0 周期值自动装载寄存器的高 8 位 (TIM0_ARRH) [0xE2]

Table 121 TIM0 周期值自动装载寄存器的高 8 位 (TIM0_ARRH)

参数名	比特位	属性	复位值	描述
ARRH	7-0	W/R	0	<p>周期值自动重载值的高 8 位。若寄存器 TIM0_CONR2 的 ARPLE 配置为 1, 寄存器的预装载功能有效, 当产生更新事件时, 该寄存器的值被装载到影子寄存器, 更新事件包括:</p> <ul style="list-style-type: none"> 计数器上溢/下溢 配置寄存器 TIM0_EGR 的 UG 位为

参数名	比特位	属性	复位值	描述
				<p>1</p> <ul style="list-style-type: none"> 从模式控制器产生的复位信号 <p>当自动重装载的值为 0 时，计数器不工作。</p>

10.6.18TIM0 预分频寄存器的低 8 位 (TIM0_PSCL) [0xE3]

Table 122 TIM0 预分频寄存器的低 8 位 (TIM0_PSCL)

参数名	比特位	属性	复位值	描述
CNTPSCL	7-0	W/R	0	<p>预分频寄存器的低 8 位,该寄存器包含了当更新事件产生时装入预分频影子寄存器的值,该寄存器的预装载功能始终有效,更新事件包括:</p> <ul style="list-style-type: none"> 计数器上溢/下溢 设置 UG 位为 1 从模式控制器产生的复位信号

10.6.19TIM0 预分频寄存器的高 8 位 (TIM0_PSCH) [0xE4]

Table 123 TIM0 预分频寄存器的高 8 位 (TIM0_PSCH)

参数名	比特位	属性	复位值	描述
CNTPSCH	7-0	W/R	0	<p>预分频器值的高 8 位,该寄存器包含了当更新事件产生时装入预分频影子寄存器的值,该寄存器的预装载功能始终有效,更新事件包括:</p> <ul style="list-style-type: none"> 计数器上溢/下溢 设置 UG 位为 1 从模式控制器产生的复位信号

10.6.20TIM0 重复计数寄存器 (TIM0_RCR) [0xE5]

Table 124 TIM0 重复计数寄存器(TIM0_RCR)

参数名	比特位	属性	复位值	描述
RCR	7-0	W/R	0	<p>周期计数次数的值,开启了预装载功能后,这些位用于设置比较寄存器的更新速率,如果允许产生更新中断,则会同时影响产生更新中断的速率。</p> <p>每次向下计数器 REP_CNT 达到 0,会产生</p>

参数名	比特位	属性	复位值	描述
				<p>一个更新事件并且计数器 REP_CNT 重新从 RCR 开始计数。由于 REP_CNT 只有在周期更新事件发生时才重载重复值，因此对此寄存器写入的新值只在下次更新事件发生时才起作用。</p> <p>这意味着在 PWM 模式下，RCR +1 对应着：</p> <p>--在边沿对齐模式下，PWM 周期的数目。</p> <p>--在中央对齐模式下，PWM 半周期的数目。</p>

10.6.21 通道 CC0 捕获/比较寄存器的低 8 位 (TIM0_CC0RL) [0xE6]

Table 125 通道 CC0 捕获/比较寄存器的低 8 位(TIM0_CC0RL)

参数名	比特位	属性	复位值	描述
CC0RL	7-0	W/R	0	<p>通道 CC0 捕获/向上计数比较寄存器的低 8 位：</p> <ul style="list-style-type: none"> 若 CC0 通道配置为输出： <p>此寄存器包含了装入比较影子寄存器的值（预装载值）。</p> <p>如果未选择预装载功能，其立即被装入捕获/比较影子寄存器。否则，只有当更新事件发生时，此预装载值才被装载入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值，根据比较结果在 OC0 端口上输出信号。</p> 若 CC0 通道配置为输入： <p>此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。</p>

10.6.22 通道 CC0 捕获/比较寄存器的高 8 位 (TIM0_CC0RH) [0xE7]

Table 126 通道 CC0 捕获/比较寄存器的高 8 位(TIM0_CC0RH)

参数名	比特位	属性	复位值	描述
CC0RH	7-0	W/R	0	<p>通道 CC0 捕获/向上计数比较寄存器的高 8 位：</p>

参数名	比特位	属性	复位值	描述
				<ul style="list-style-type: none"> ● 若 CC0 通道配置为输出： 此寄存器包含了装入比较影子寄存器的值（预装载值）。 如果未选择预装载功能，其立即被装入捕获/比较影子寄存器中。否则，只有当更新事件发生时，此预装载值才被装载入捕获/比较影子寄存器中。影子寄存器包含了与计数器比较的值，根据比较结果在 OC0 端口上输出信号。 ● 若 CC0 通道配置为输入： 此寄存器包含了由上一次输入捕获事件捕获到的计数器值的高 8 位。

10.6.23 通道 CC1 捕获/比较寄存器的低 8 位 (TIM0_CC1RL) [0xE9]

Table 127 通道 CC1 捕获/比较寄存器的低 8 位(TIM0_CC1RL)

参数名	比特位	属性	复位值	描述
CC1RL	7-0	W/R	0	<p>通道 CC1 捕获寄存器/向上计数比较寄存器的低 8 位：</p> <ul style="list-style-type: none"> ● 若 CC1 通道配置为输出： 此寄存器包含了装入比较影子寄存器的值（预装载值）。 如果未选择预装载功能，其立即被装入捕获/比较影子寄存器。否则，只有当更新事件发生时，此预装载值才被装载入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值，根据比较结果在 OC1 端口上输出信号。 ● 若 CC1 通道配置为输入： 此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。

10.6.24 通道 CC1 捕获/比较寄存器的高 8 位 (TIM0_CC1RH) [0xEA]

Table 128 通道 CC1 捕获/比较寄存器的高 8 位(TIM0_CC1RH)

参数名	比特位	属性	复位值	描述
CC1RH	7-0	W/R	0	<p>通道 CC1 捕获寄存器/向上计数比较寄存器的高 8 位：</p> <ul style="list-style-type: none"> ● 若 CC1 通道配置为输出： 此寄存器包含了装入比较影子寄存器的值（预装载值）。 如果未选择预装载功能，其立即被装入捕获/比较影子寄存器。否则，只有当更新事件发生时，此预装载值才被装载入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值，根据比较结果在 OC1 端口上输出信号。 ● 若 CC1 通道配置为输入： 此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。

10.6.25 通道 CC2 捕获/比较寄存器的低 8 位 (TIM0_CC2RL) [0xEB]

Table 129 通道 CC2 捕获/比较寄存器的低 8 位(TIM0_CC2RL)

参数名	比特位	属性	复位值	描述
CC2RL	7-0	W/R	0	<p>通道 CC2 捕获寄存器/向上计数比较寄存器的低 8 位：</p> <ul style="list-style-type: none"> ● 若 CC2 通道配置为输出： 此寄存器包含了装入比较影子寄存器的值（预装载值）。 如果未选择预装载功能，其立即被装入捕获/比较影子寄存器。否则，只有当更新事件发生时，此预装载值才被装载入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值，根据比较结果在 OC2 端口上输出信号。 ● 若 CC2 通道配置为输入： 此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。

10.6.26 通道 CC2 捕获/比较寄存器的高 8 位 (TIM0_CC2RH) [0xEC]

Table 130 通道 CC2 捕获/比较寄存器的高 8 位(TIM0_CC2RH)

参数名	比特位	属性	复位值	描述
CC2RH	7-0	W/R	0	<p>通道 CC2 捕获寄存器/向上计数比较寄存器的高 8 位：</p> <ul style="list-style-type: none"> ● 若 CC2 通道配置为输出： 此寄存器包含了装入比较影子寄存器的值（预装载值）。 如果未选择预装载功能，其立即被装入捕获/比较影子寄存器。否则，只有当更新事件发生时，此预装载值才被装载入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值，根据比较结果在 OC2 端口上输出信号。 ● 若 CC2 通道配置为输入： 此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。

10.6.27 通道 CC3 捕获/比较寄存器的低 8 位 (TIM0_CC3RL) [0xED]

Table 131 通道 CC3 捕获/比较寄存器的低 8 位(TIM0_CC3RL)

参数名	比特位	属性	复位值	描述
CC3RL	7-0	W/R	0	<p>通道 CC3 捕获寄存器/向上计数比较寄存器的低 8 位：</p> <ul style="list-style-type: none"> ● 若 CC3 通道配置为输出： 此寄存器包含了装入比较影子寄存器的值（预装载值）。 如果未选择预装载功能，其立即被装入捕获/比较影子寄存器。否则，只有当更新事件发生时，此预装载值才被装载入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值，根据比较结果在 OC3 端口上输出信号。 ● 若 CC3 通道配置为输入： 此寄存器包含了由上一次输入捕获事件

参数名	比特位	属性	复位值	描述
				捕获到的计数器值的低 8 位。

10.6.28 通道 CC3 捕获/比较寄存器的高 8 位 (TIM0_CC3RH) [0xEE]

Table 132 通道 CC3 捕获/比较寄存器的高 8 位(TIM0_CC3RH)

参数名	比特位	属性	复位值	描述
CC3RH	7-0	W/R	0	<p>通道 CC3 捕获寄存器/向上计数比较寄存器的高 8 位：</p> <ul style="list-style-type: none"> 若 CC3 通道配置为输出： 此寄存器包含了装入比较影子寄存器的值（预装载值）。 如果未选择预装载功能，其立即被装入捕获/比较影子寄存器。否则，只有当更新事件发生时，此预装载值才被装载入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值，根据比较结果在 OC3 端口上输出信号。 若 CC3 通道配置为输入： 此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。

10.6.29 TIM0 刹车控制寄存器 (TIM0_BRKC) [0xEF]

Table 133 TIM0 刹车控制寄存器(TIM0_BRKC)

参数名	比特位	属性	复位值	描述
BRKZEN	0	W/R	0	<p>刹车时输出高阻使能寄存器：</p> <p>0：刹车时不输出高阻；</p> <p>1：刹车时输出高阻。</p>
IOSS	1	W/R	0	<p>空闲模式（MOE=0）下“关闭状态”（CCxEN=0 或 CCxNEN=0）输出控制，该位用于当 MOE=0 且通道设置为输出时。</p> <p>0：当定时器 CCx 通道处于关闭状态时（CCxEN=0 或 CCxNEN=0），输出脚 PWM_CHx 或 PWM0_CHxN 的输出使能 PWM_CHxEN 和 PWM0_CHxNEN 无效，</p>

参数名	比特位	属性	复位值	描述
				<p>PWM0_CHx 或 PWM0_CHxN 输出高阻。</p> <p>详见 Table 103 中 MOE=0 且 IOSS=0 的部分。</p> <p>1: 当定时器 CCx 通道处于关闭状态时 (CCxEN=0 或 CCxNEN=0), 输出脚 PWM0_CHx 和 PWM0_CHxN 的输出使能 PWM0_CHxEN 和 PWM0_CHxNEN 有效, PWM0_CHx 和 PWM0_CHxN 输出其空闲电平, 详见 Table 103 中 MOE=0 且 IOSS=1 的部分。</p> <p>注: 一旦 LOCK 级别设置为 2 或 3, 则该位不能被修改。</p>
ROSS	2	W/R	0	<p>运行模式 (MOE=1) 下“关闭状态” (CCxEN=0 或 CCxNEN=0) 输出控制, 该位用于当 MOE=1 且通道位互补输出时。没有互补输出的定时器没有 OSSR 位。</p> <p>0: 当定时器 CCx 通道处于关闭状态时 (CCxEN=0 或 CCxNEN=0), 输出脚 PWM0_CHx 或 PWM0_CHxN 的输出使能 PWM0_CHxEN 和 PWM0_CHxNEN 无效, PWM0_CHx 或 PWM0_CHxN 输出高阻。</p> <p>详见 Table 103 中 MOE=1 且 ROSS=0 的部分。</p> <p>1: 当定时器 CCx 通道处于关闭状态时 (CCxEN=0 或 CCxNEN=0), 输出脚 PWM0_CHx 和 PWM0_CHxN 的输出使能 PWM0_CHxEN 和 PWM0_CHxNEN 有效, PWM0_CHx 和 PWM0_CHxN 输出其空闲电平, 详见 Table 103 中 MOE=1 且 ROSS=1 的部分。</p> <p>注: 一旦 LOCK 级别设置为 2, 则该位不能被修改。</p>

参数名	比特位	属性	复位值	描述
CMPBRKE	4	W/R	0	比较器刹车功能使能寄存器： 0：禁止刹车输入 1：开启刹车输入 注 1：一旦 LOCK 级别设置为 1、2 或 3，该位不能修改。 注 2：此位只控制外部输入的刹车信号，对寄存器配置的刹车事件 BG 无效。 注 3：通过配置 16.3 节的寄存器 BKPRSEL 可以改变比较器器刹车信号的极性。
EXTBRKE	4	W/R	0	外部刹车功能使能寄存器： 0：禁止刹车输入 1：开启刹车输入 注 1：一旦 LOCK 级别设置为 1、2 或 3，该位不能修改。 注 2：此位只控制外部输入的刹车信号，对寄存器配置的刹车事件 BG 无效。
BRKP	5	W/R	0	刹车输入极性： 0：刹车输入高电平为有效电平 1：刹车输入低电平为有效电平 注：一旦 LOCK 级别设置为 1、2 或 3，该位不能修改。
AOE	6	W/R	0	自动输出使能： 0：MOE 只能被软件置 1； 1：MOE 能被软件置 1 或在刹车信号无效后的更新事件自动置 1。 注：一旦 LOCK 级别设置为 1、2 或 3，该位不能修改。
MOE	7	W/R	0	主输出使能寄存器，一旦刹车输入有效，该位被硬件异步清 0，根据 AOE 的设置，可由软件置 1 或自动置 1，它仅对配置为输出通道有效。 0：使定时器处于空闲状态。定时器的输出

参数名	比特位	属性	复位值	描述
				<p>受 IOSS、OIS0、OIS0N、CC0EN、CC0NEN、CC0P 和 CC0NP 位的控制。</p> <p>1: 使定时器处于运行状态。定时器的输出受 ROSS、CC0EN、CC0NEN、CC0P 和 CC0NP 位的控制。</p>

10.6.30TIM0 事件产生寄存器 (TIM0_EGR) [0xF1]

Table 134 TIM0 事件产生寄存器(TIM0_EGR)

参数名	比特位	属性	复位值	描述
BG	0	W	0	<p>产生刹车事件，该位由软件置 1，硬件自动清 0:</p> <p>0: 无动作;</p> <p>1: 产生一个刹车事件。此时 MOE=0, BIF=1, 若打开刹车中断使能，则产生刹车中断。</p>
TG	1	W	0	<p>产生触发事件，该位由软件置 1，硬件自动清 0:</p> <p>0: 无动作;</p> <p>1: 产生一个触发事件。此时 TIF=1, 若打开了触发中断使能，则产生触发中断。</p>
COMG	2	W	0	<p>产生寄存器 TIM0_CCENR、TIM0_CCPS 和 TIM0_CCxMR.OCxMS 的更新事件:</p> <p>0: 无动作;</p> <p>1: 当 CCxPE=1 时，更新 TIM0_CCENR、TIM0_CCPS 和 TIM0_CCxMR.OCxMS。</p>
CC0G	3	W	0	<p>产生捕获/比较事件，该位由软件置 1，用于产生一个捕获/比较事件，由硬件自动清 0:</p> <p>0: 无动作;</p> <p>1: 在通道 CC0 上产生一个捕获/比较事件</p> <p>若通道 CC0 配置位输出:</p> <p>设置 CC0IF=1, 若打开比较中断使能，则产生比较中断。</p> <p>若通道 CC0 配置为输入:</p> <p>当前的计数器值捕获至 TIM0_CC0R 寄存</p>

参数名	比特位	属性	复位值	描述
				器, 捕获中断标志位 CC0IF 自动置 1, 若打开捕获中断使能, 则产生捕获中断, 若 CC0IF 已经为 1, 则过捕获标志位 CC0OIF 自动置 1。
CC1G	4	W	0	参见 CC0G 描述。
CC2G	5	W	0	参见 CC0G 描述。
CC3G	6	W	0	参见 CC0G 描述。
UG	7	W	0	产生更新事件, 该位由软件置 1, 由硬件自动清 0: 0: 无动作; 1: 初始化计数器、初始化预分频计数器、产生一个更新事件: <ul style="list-style-type: none"> ● 若在中央对齐模式或向上计数模式下, 计数器被清 0; ● 若在向下计数模式下, 计数器初始化为 TIM0_ARR 的值; ● 初始化预分频计数器为 0。

10.6.31 相位偏移控制寄存器(TIM0_PHACON) [0XF2]

Table 135 相位偏移控制寄存器(TIM0_PHACON)

参数名	比特位	属性	复位值	描述
PHASEEN	0	W/R	0	不同通道间 PWM 输出相位偏移使能寄存器: 0: 不同通道间 PWM 输出无相位偏移; 1: 打开通道间 PWM 输出相位偏移功能。
PHASECH	2-1	W/R	00	具有相位偏移的通道选择寄存器: 00: 通道 CC0 的 PWM 输出具有相位偏移; 01: 通道 CC1 的 PWM 输出具有相位偏移; 10: 通道 CC2 的 PWM 输出具有相位偏移; 11: 通道 CC3 的 PWM 输出具有相位偏移。
RESERVED	7-3	R	0	保留。

10.6.32TIM0 中断标志寄存器 1 (TIM0_ISR0) [0xF3]

Table 136 TIM0 中断标志寄存器 0(TIM0_ISR0)

参数名	比特位	属性	复位值	描述
BIF	0	W/R	0	刹车中断标志。一旦发生刹车事件，由硬件将该位置 1。刹车输入无效时，该位可由软件清 0。 0: 无刹车事件产生； 1: 产生刹车事件。
TIF	1	W/R	0	触发中断标志，当发生触发事件时（当在 TRGI 输入端检测到有效边沿或软件配置寄存器 TIM0_EGR 的 TG 位）由硬件将该位置 1，由软件清 0： 0: 无触发事件产生 1: 有触发事件产生
COMIF	2	W/R	0	COM 中断标志，一旦 COM 事件来到，该位由硬件置 1，由软件清 0： 0: 无 COM 事件产生 1: 有 COM 事件产生
CC0IF	3	W/R	0	互补模式下，通道 CC0 的捕获/比较中断标志： ● 如果通道 CC0 配置为输出模式： 当计数器值与比较值匹配时由硬件置 1，由软件清 0 0: 无匹配发生 1: 计数器与比较值匹配 ● 如果通道 CC0 配置为输入捕获模式： 当捕获事件发生时该位由硬件置 1，由软件清 0 或通过读 TIM0_CC0R 清 0 0: 无输入捕获产生 1: 输入捕获产生
CC1IF	4	W/R	0	通道 CC1 捕获/比较中断标志寄存器，参见

参数名	比特位	属性	复位值	描述
				CC0IF 描述
CC2IF	5	W/R	0	通道 CC2 捕获/比较中断标志寄存器，参见 CC0IF 描述
CC3IF	6	W/R	0	通道 CC3 捕获/比较中断标志寄存器，参见 CC0IF 描述
RESERVED	7	R	0	保留。

10.6.33TIM0 中断标志寄存器 2 (TIM0_ISR1) [0xF4]

Table 137 TIM0 中断标志寄存器 1(TIM0_ISR1)

参数名	比特位	属性	复位值	描述
RESERVED	3-0	R	0	保留。
UOVERIF	4	W/R	0	计数器上溢更新中断标志： 0：没有上溢事件发生； 1：发生上溢事件。
DOVERIF	5	W	0	计数器下溢更新中断标志： 0：没有下溢事件发生； 1：发生下溢事件。
SOFTUIF	6	W/R	0	复位（配置 TIM0_EGR 的 UG 位或复位模式下的从模式控制器产生的复位信号）更新中断标志： 0：没有软件或复位更新事件发生； 1：发生软件或复位更新事件。
RESERVED	7	R	0	保留。

10.6.34TIM0 计数器的低 8 位 (TIM0_CNTL) [0xF5]

Table 138 TIM0 计数器的低 8 位 (TIM0_CNTL)

参数名	比特位	属性	复位值	描述
CNTL	7-0	W	0	计数器的低 8 位。

10.6.35TIM0 计数器的高 8 位 (TIM0_CNTH) [0xF6]

Table 139 TIM0 计数器的高 8 位(TIM0_CNTH)

参数名	比特位	属性	复位值	描述
CNTH	7-0	W	0	计数器的高 8 位。

10.6.36 相位偏移长度寄存器的低 8 位(TIM0_PHASEL) [0XF7]

Table 140 相位偏移长度寄存器的低 8 位(TIM0_PHASEL)

参数名	比特位	属性	复位值	描述
PHASEL	7-0	W/R	0	相位偏移长度寄存器的低 8 位。

10.6.37 相位偏移长度寄存器的高 8 位(TIM0_PHASEH) [0XF9]

Table 141 相位偏移长度寄存器的高 8 位(TIM0_PHASEH)

参数名	比特位	属性	复位值	描述
PHASEH	7-0	W/R	0	相位偏移长度寄存器的高 8 位。

10.6.38 TIM 计数器使能寄存器(TIM0_CNTEN) [0xFA]

Table 142 TIM 计数器使能寄存器 (TIM0_CNTEN)

参数名	比特位	属性	复位值	描述
CNTEN	0	W/R	0	计数器使能软件控制位： 0：计数器不计数。 1：计数器计数。 注：触发模式下不需要配置，硬件自动设置 CNTEN 位。
RESERVED	7-1	R	0	保留。

11 高级定时器 1 (PWM1)

11.1 概述

JMT18F003PLUS 高级定时器 1 主要包含一个 16 位的向上/向下自动重装载计数器、六个 16 位的比较/捕获寄存器、一个可编程的时钟预分频器和多个控制寄存器。

本定时器主要用途有：

- 产生多种输出波形（输出比较波形、PWM 波形、嵌入死区的互补 PWM 波形等）
- 测量输入信号的脉冲宽度（输入捕获功能）
- 作为通用定时器使用
- 作为霍尔传感器接口
- 作为正交增量编码器接口

本定时器可以产生 1 对互补的 PWM 输出。

通过控制定时器的时钟分频（两种方式实现：控制 CRM 模块中 PWM 时钟分频器和控制定时器自身的时钟分频器），本定时器可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

11.2 特性

本定时器的特性：

- 16 位向上、向下和向上/向下自动装载计数器。
- 16 位可编程分频器，计数器时钟频率的分频系数为 1~65536 之间的任意值。
- 3 个输入捕获通道。
- 在互补模式下可以输出 1 对互补的 PWM 波形。
- 可以在指定数目的计数器周期之后更新定时器中的预装载寄存器。
- 刹车功能：外部刹车信号可以将定时器输出信号置于复位状态，也可以通过软件配置刹车事件实现刹车功能。
- 如下事件发生时产生中断：
 - 更新：计数器向上溢出、向下溢出，计数器初始化（通过软件或内部/外部触发）
 - 触发事件（计数器启动，停止，初始化或由内部/外部触发计数）产生触发中断
 - 输入捕获，当捕获数据时产生捕获中断
 - 输出比较，当计数器与输出比较寄存器匹配时产生比较中断
 - 刹车信号输入
 - 软件配置 COM 事件，更新通道输出控制寄存器。
- 正交增量编码器接口功能

● 霍尔传感器接口功能

11.3 框图

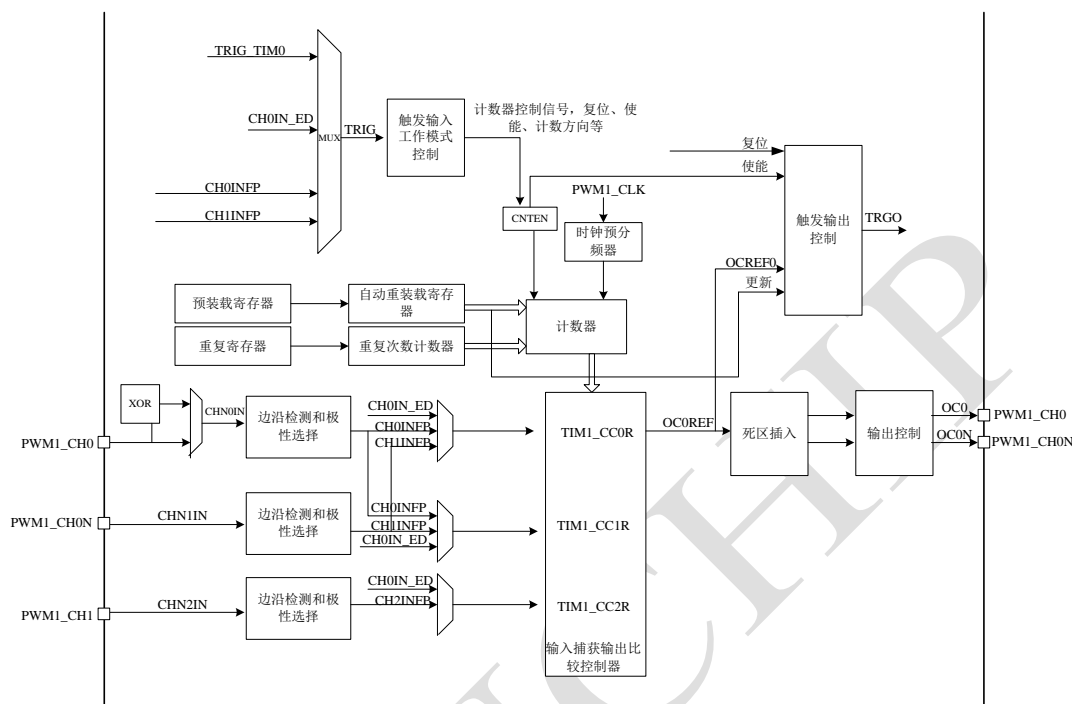


Figure 63 JMT18F003PLUS 高级定时器 1 (TIM1/PWM1) 框图

11.4 输入输出引脚配置

有些引脚可以复用为高级定时器的输入输出功能，详细描述见 17.5 节。

当高级定时器配置为输入捕获功能时（寄存器 TIM1_CCxMR⁴的 CCxMS 位配置为 01/10/11），外部信号通过 PWM1_CH0、PWM1_CH0N、PWM1_CH1 通道输入；当高级定时器配置为输出比较功能时（寄存器 TIM1_CC0MR 的 CC0MS 位配置为 00），高级定时器的输出信号通过 PWM1_CH0 和 PWM1_CH0N 输出。

当高级定时器输出互补 PWM 波形时，PWM1_CH0 和 PWM1_CH0N 是通道 CC0 的一对互补 PWM 输出。

11.5 功能描述

高级定时器的结构主要分为五部分：计数器部分、从模式控制器、主模式控制器、三个通道的输入捕获控制器和一个通道的输出比较控制器。下面分别介绍高级定时器各个部分的

⁴ CC_xMR 中的 x 比较通道标号, x=0, 1, 2。

功能和各种工作模式的控制方法。

11.5.1 计数器

11.5.1.1 计数器的计数周期

高级定时器 TIM1 包含一个 16 位的计数器，计数器的周期值保存在自动重载寄存器中，自动重载寄存器是预先装载的（TIM1_ARR 是其预装载寄存器），根据 TIM1_CONR1 寄存器中的自动装载使能位(ARPLE)的设置，预装载寄存器的值被立即或在更新事件时传送到影子寄存器（软件不可见）。写或读自动重载寄存器将访问预装载寄存器(TIM1_ARR)。

11.5.1.2 计数器的使能控制

计数器使能受到寄存器 TIM1_CNTEN 的 CNTEN 位和从模式控制器控制，具体控制方式如下：

- 当寄存器 TIM1_TGICR0 的 SMS 位配置为 000 时，高级定时器关闭从模式，此时计数器使能只受 TIM1_CNTEN 的 CNTEN 位控制，当 CNTEN 配置为 1 时，计数器计数，当 CNTEN 配置为 0 时，计数器停止计数。
- 当寄存器 TIM1_TGICR0 的 SMS 位配置为 110 时，高级定时器工作在触发模式下，此时将寄存器 TIM1_CONR2 的 CNTEN 位配置为 0，计数器在触发输入的有效沿启动（但不复位）。
- 当寄存器 TIM1_TGICR0 的 SMS 位配置为其他值时，计数器使能同时受 TIM1_CNTEN 的 CNTEN 位和触发信号的控制，此时需要将寄存器 CNTEN 位配置为 1，计数器在输入触发信号的控制下工作。

寄存器 TIM1_CNTEN 的 CNTEN 位置位后意味着计数器即将开始计数，高级定时器也即将进入工作状态，此时其他寄存器应该完成配置，所以建议 TIM1_CNTEN 的 CNTEN 位最后配置。

11.5.1.3 计数器预分频控制器

预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。通过配置寄存器 TIM1_PSCL 和 TIM1_PSCH 可以改变分频系数，此寄存器带有缓存器，所以可以在工作时配置，新的预分频器的参数在下一次更新事件（UEV）到来时被采用。Figure 27 为预分频器的参数由 1 分频变为 4 分频时，计数器的时序图。

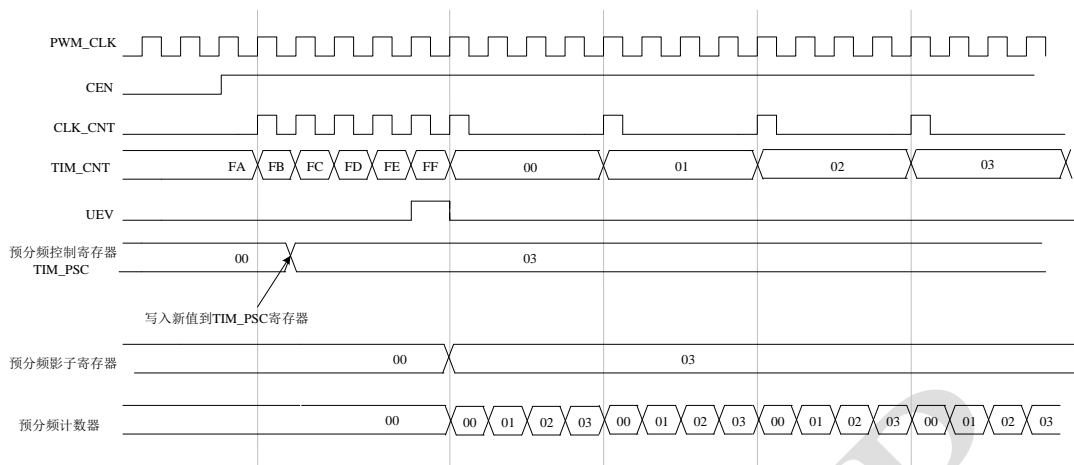


Figure 64 预分频器的参数由 1 分频变为 4 分频时，计数器的时序图

11.5.1.4 计数器计数模式控制

计数器有三种计数模式：向上计数模式、向下计数模式和中央对齐计数模式（向上向下交替计数），配置 TIM1_CONR2 的 CNTMC 位可以设置计数器的计数模式：

当 CNTMC 配置为 00 时，计数器工作在边沿计数模式下，此时可以通过配置 TIM1_CONR2 寄存器的 CNTDIR 位设置计数器的计数方向。

当 CNTMC 配置为 01/10/11 时，计数器工作在中央对齐模式下，此时不可以配置 TIM1_CONR2 寄存器的 CNTDIR 位，计数器的方向由硬件自动设置。

当寄存器 TIM1_TGICR0 的 SMS 位配置为 001/010/011 时，即高级定时器工作在增量编码器接口模式下，TIM1_CONR2 的 CNTDIR 位也是不可写的，此时计数器的计数方向由硬件根据触发信号确定。

11.5.1.4.1 向上计数模式

当寄存器 TIM1_CONR2 的 CNTMC 位配置为 00 且 CNTDIR 位配置为 0 时，计数器向上计数，计数器的值从 0 开始累加，一直累加到计数器的重载值（TIM1_ARR 中的值）。一旦计数器达到了重载值，寄存器将从 0 开始重新计数。若设置了重复周期计数的值 (TIM1_RCR)，在计数器向上溢出次数达到设置的重复周期次数时才产生更新事件；否则每当计数器溢出时会产生更新事件。在此种计数模式下，设置 TIM1_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，计数器初始化为 0；

更新事件产生的规则：

- 设置寄存器 TIM1_CONR3 中的 UPOUDIS 位为 1，可以禁止上溢更新事件的产生，若更新事件被禁止，预分频寄存器的数值不变，同时计数器按预分频的频率进行计数。禁止更新事件产生的目的为：避免在向预装载寄存器中写入新值时更新影子寄存器。
- 设置 TIM1_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，可以产

生一个更新事件。

- 如果寄存器 TIM1_CONR2 中的 URC 位被设置为 1，设置 UG 位为 1 将产生一个更新事件 UEV，但硬件不更新中断标志位 SOFTUIF（即不产生软更新中断）。这是为了避免在捕获模式下初始化计数器时产生更新中断。

当发生更新事件时，硬件做如下处理：

- 上溢更新标志位 UOVERIF 自动置 1，若上溢更新中断使能（TIM1_CONR3 寄存器的 UPOIE 位为 1）有效，则产生上溢更新中断。
- 若通过软件配置 UG 位为 1 产生更新事件，硬件根据 URC 位的配置更新中断标志位 SOFTUIF，若软更新中断使能（TIM1_CONR3 寄存器的 SOFTUPIE 位）有效，则产生软件更新中断。
- 自动重载寄存器被重新置入预装载寄存器的值。
- 重复周期计数器被重新加载为 TIM1_RCR 寄存器的内容。
- 预分频器的影子寄存器被置入预装载寄存器（TIM1_PSCL 和 TIM1_PSCH 寄存器）的值。

Figure 28 和 Figure 29 为向上计数模式下计数器的时序，计数器时钟的预分频系数为 1（2 分频）。

Figure 28 中自动装载寄存器的预装载功能使能，新配置的计数器周期值在更新事件时传递到影子寄存器中，下一周期生效。Figure 29 的自动装载寄存器的预装载功能关闭，新配置的计数器周期值立即传送到影子寄存器中，在本周期即生效。

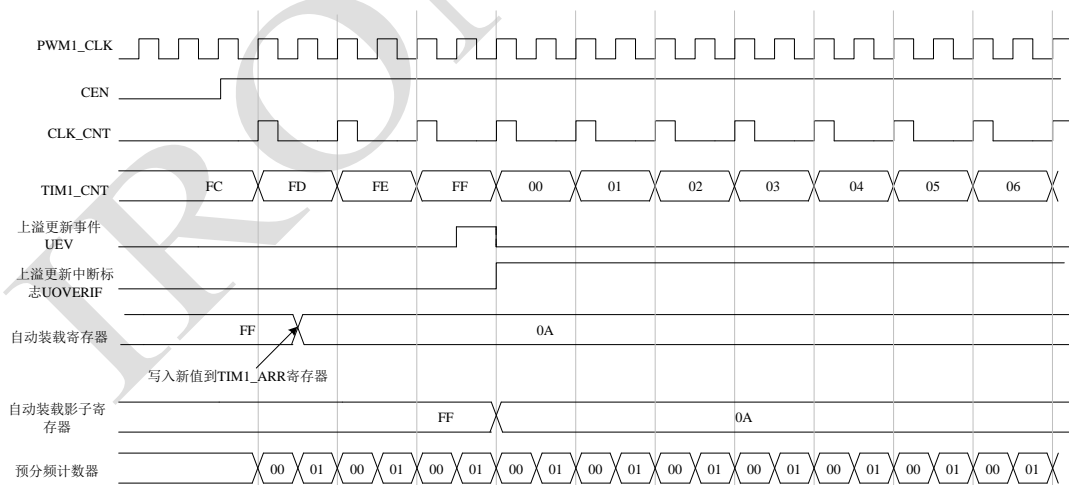


Figure 65 向上计数模式下，自动装载寄存器的预装载功能使能时计数器的时序图

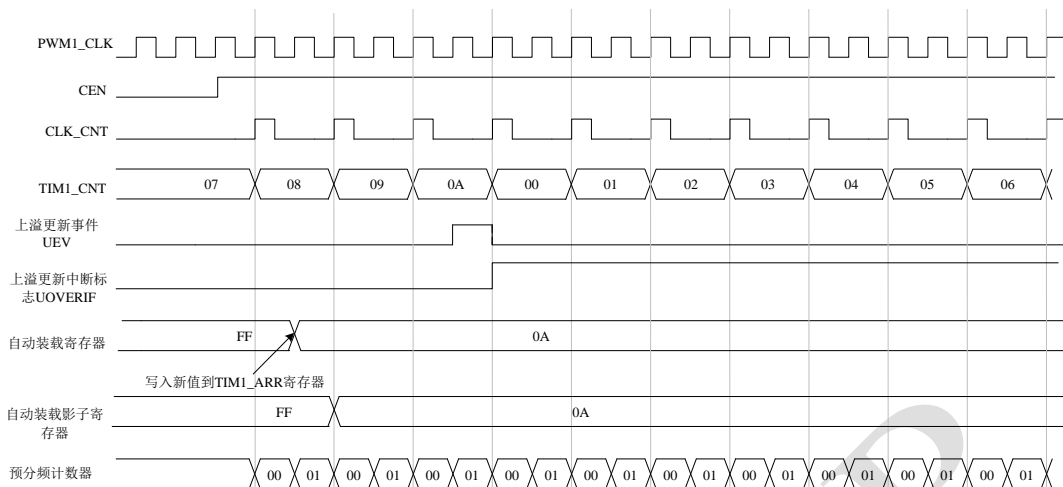


Figure 66 向上计数模式下，当自动装载寄存器的预装载功能不使能时计数器的时序图

11.5.1.4.2 向下计数模式

当寄存器 TIM1_CONR2 的 CNTMC 位配置为 00 且 CNTDIR 位配置为 1 时，计数器向下计数，计数器的值从自动装入的值 TIM1_ARR 开始递减到 0，然后从自动装入的值重新开始计数并且产生一个计数器向下溢出事件。若设置了重复周期计数的值，在计数器向下溢出次数达到设置的重复周期次数时才产生更新事件；否则每当计数器向下溢出时会产生更新事件。在此种计数模式下，当设置 TIM1_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，计数器初始化为周期值 TIM1_ARR。

更新事件产生的规则：

- 设置寄存器 TIM1_CONR3 中的 DOWNOUDIS 位为 1，可以禁止下溢更新事件的产生。若更新事件被禁止，预分频寄存器的数值不变，同时计数器按预分频的频率进行计数。禁止更新事件产生的目的为：避免在向预装载寄存器中写入新值时更新影子寄存器。
- 设置 TIM1_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，可以产生一个更新事件。
- 如果 TIM1_CONR3 中的 URC 位被设置为 1，设置 UG 位为 1 将产生一个更新事件 UEV，但硬件不更新中断标志 SOFTUIF（即软更新中断标志）。这是为了避免在捕获模式下初始化计数器时产生更新中断。

当发生更新事件时，硬件做如下处理：

- 下溢更新标志位 DOVERIF 自动置 1，若下溢更新中断使能（TIM1_CONR3 寄存器的 DOWNOIE 位）有效，则产生下溢更新中断。
- 若通过软件配置 UG 位为 1 产生更新事件，硬件根据 URC 位的配置更新中断标志位 SOFTUIF，若软更新中断使能（TIM1_CONR3 寄存器的 SOFTUPIE 位）有效，则产生软件更新中断。

- 自动重载寄存器被重新置入预装载寄存器的值。
- 重复周期计数器被重新加载为 TIM1_RCR 寄存器的内容。
- 预分频器的影子寄存器被置入预装载寄存器(TIM1_PSCL 和 TIM1_PSCH 寄存器)的值。

值得注意的是，在向下计数模式下，自动重载寄存器的值在计数器重载入之前更新，因此一下个周期将是预期的值。Figure 30 为向下计数模式下计数器的时序图。

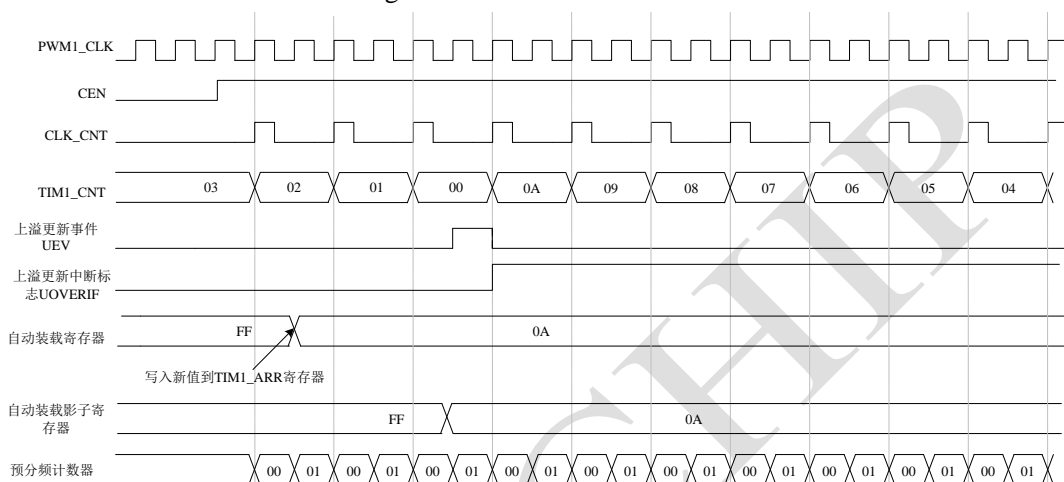


Figure 67 向上计数时计数器的时序图

11.5.1.4.3 中央对齐计数模式

寄存器 TIM1_CONR2 的 CNTMC 位配置为 00/01/11 时，计数器的计数模式为中央对齐模式，中央对齐模式下，寄存器 TIM1_CONR2 的 CNTDIR 位是不可写的，它由硬件更新并指示当前的计数方向。

在这种模式下，计数器从 0 开始计数到自动加载值（TIM1_ARR 寄存器）减 1，产生一个计数器上溢事件，然后向下计数到 1 并且产生一个计数器下溢事件，然后再从 0 开始计数。在此种计数模式下，当设置 TIM1_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时，计数器初始化为 0；

每一次计数上溢和每一次计数下溢都可以产生更新事件。

更新事件产生的规则：

- 设置寄存器 TIM1_CONR3 中的 UPOUDIS 位为 1，可以禁止上溢更新事件的产生，设置寄存器 TIM1_CONR3 中的 DOWNOUDIS 位为 1，可以禁止下溢更新事件的产生。若更新事件被禁止，预分频寄存器的数值不变，同时计数器按预分频的频率进行计数。禁止更新事件产生的目的为：避免在向预装载寄存器中写入新值时更新影子寄存器。
- 设置 TIM1_EGR 寄存器中的 UG 位为 1，或从模式控制器产生复位信号时可以产生一个软件更新事件。

- 如果 TIM1_CONR3 中的 URC 位被设置为 1，设置 UG 位为 1 将产生一个更新事件 UEV，但硬件不更新 SOFTUIF 标志（即不产生软更新中断）。这是为了避免在捕获模式下初始化计数器时产生更新。

当发生更新事件时，硬件做如下处理：

- 若更新事件是上溢更新事件，上溢更新标志位 UOVERIF 自动置位，若上溢更新中断使能（TIM1_CONR3 寄存器的 UPOIE 位）有效，则产生上溢更新中断。若更新事件是下溢更新事件，下溢更新标志位 DOVERIF 自动置位，若设置了下溢更新中断使能（TIM1_CONR3 寄存器的 DOWNOIE 位）有效，则产生下溢更新中断。
- 若配置 UG 位为 1 产生更新事件时，硬件根据 URC 位更新中断标志位 SOFTUIF，若软更新中断使能（TIM1_CONR3 寄存器的 SOFTUPIE 位）有效，则产生软件更新中断。
- 自动重载寄存器被重新置入预装载寄存器的值。
- 重复周期计数器被重新加载为 TIM1_RCR 寄存器的值。
- 预分频器的影子寄存器被置入预装载寄存器的值（TIM1_PSCL 和 TIM1_PSCH 寄存器的内容）。

值得注意的是，如果因为计数器下溢而产生更新，自动重载寄存器的值在计数器重载入之前更新，因此一下个周期将是预期的值。Figure 31 和 Figure 32 为中央计数模式下计数器的时序。Figure 31 的自动装载寄存器的预装载功能使能，新配置的计数器周期值在更新事件时传递到影子寄存器中，下一周期生效。Figure 32 的自动装载寄存器的预装载功能关闭，新配置的计数器周期值立即传送到影子寄存器中，在本周期即生效。

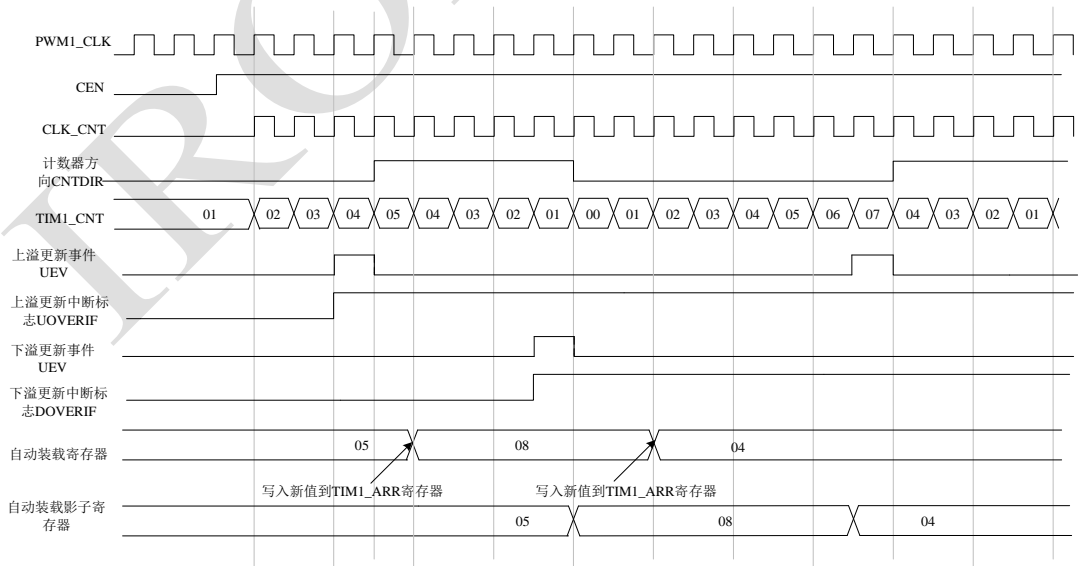


Figure 68 中央计数模式下，自动装载寄存器的预装载功能使能时计数器的时序图

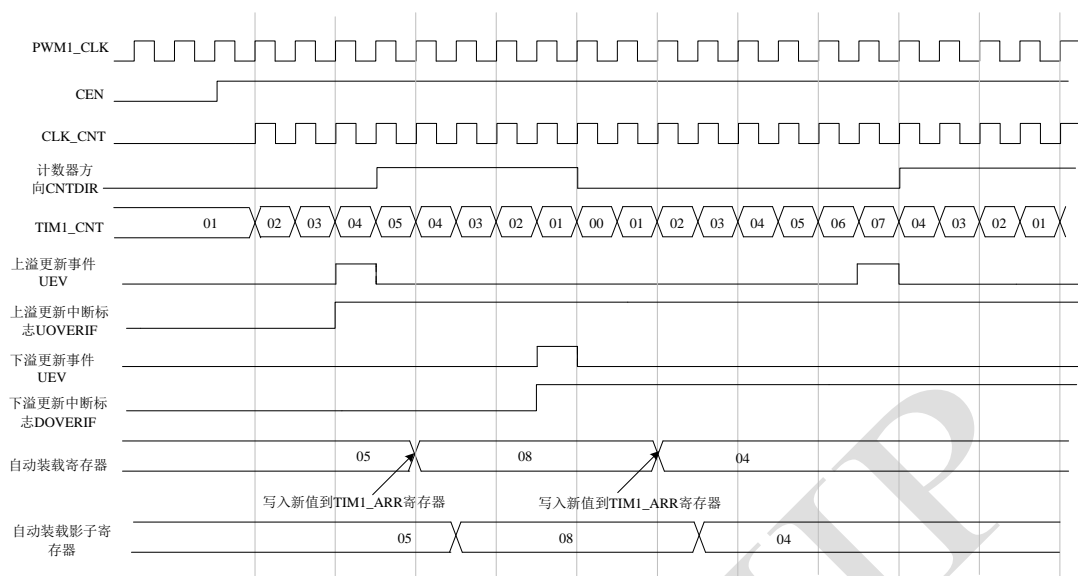


Figure 69 中央计数模式下，自动装载寄存器的预装载功能不使能时计数器的时序图

11.5.1.5 计数器的重复周期计数控制器

借助重复周期计数功能，可以在计数器连续上溢/下溢 N 次后，才会产生计数器更新事件，更新各寄存器的内容。

周期计数器递减的条件有：

- 向上计数模式下的每次计数器上溢；
- 向下计数模式下的每次计数器下溢；
- 中央对齐模式下的每次计数器上溢/下溢。

周期计数器是自动加载的向下计数器，重复次数由寄存器 TIM1_RCR 的值定义。当更新事件由软件产生或者通过硬件的从模式控制器产生时，无论周期计数器的值是多少，立即发生更新事件，并且寄存器 TIM1_RCR 的内容被重新载入到周期计数器。

Figure 33 为向上计数模式下周期计数控制器对更新事件的控制，图中箭头表示产生更新事件，当寄存器 TIM1_CRC 配置为 0 时，每当计数器溢出时产生更新事件，当寄存器 TIM1_CRC 配置为 1 时，计数器连续 2 次溢出时产生更新事件，当 TIM1_CRC 配置为 N 时，计数器连续 N+1 次溢出时产生更新事件。若软件配置更新事件时，立即发生更新事件。

Figure 34 为向下计数模式下周期计数控制器对更新事件的控制。

Figure 35 为中央对齐计数模式下周期计数控制器对更新事件的控制。

Figure 36 为中央对齐模式下，当重复周期次数为 3 时，更新事件产生的时序图。

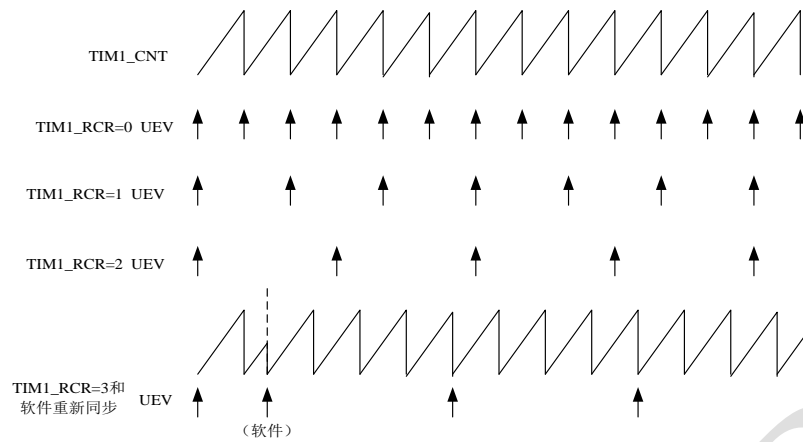


Figure 70 向上计数模式下，重复计数次数分别为 0、1、2、3 时更新事件的产生

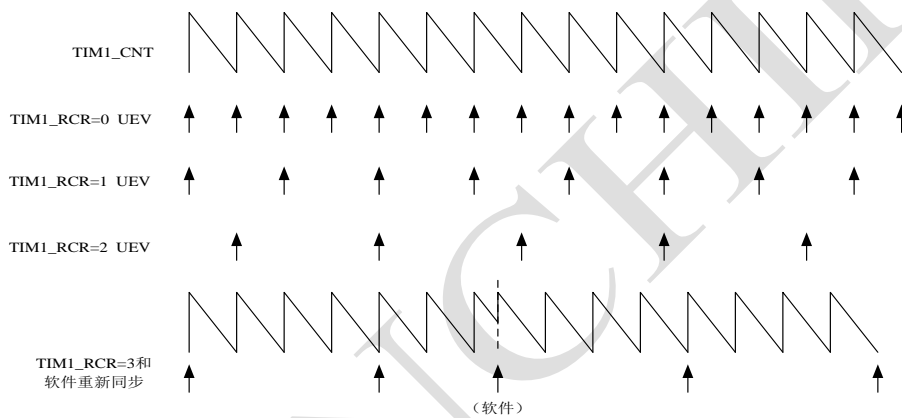


Figure 71 向下计数模式下，重复计数次数分别为 0、1、2、3 时更新事件的产生

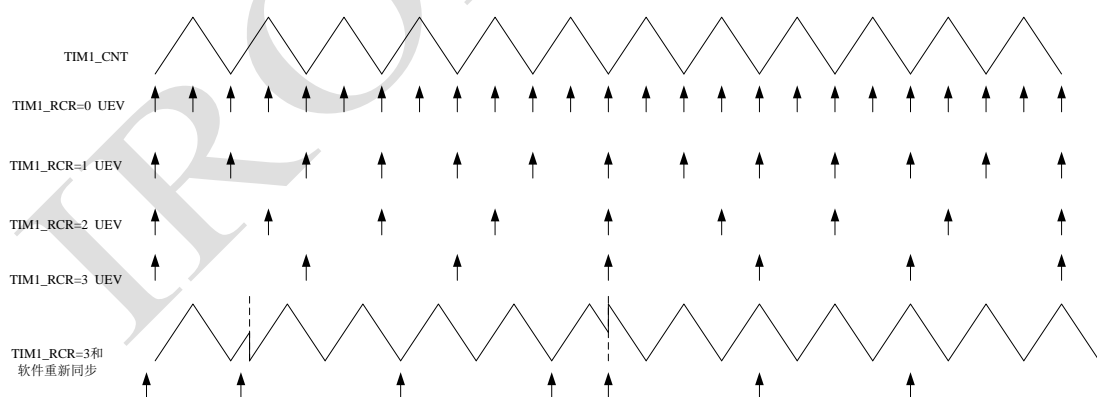


Figure 72 中央对齐计数模式下，重复计数次数分别为 0、1、2、3 时更新事件的产生

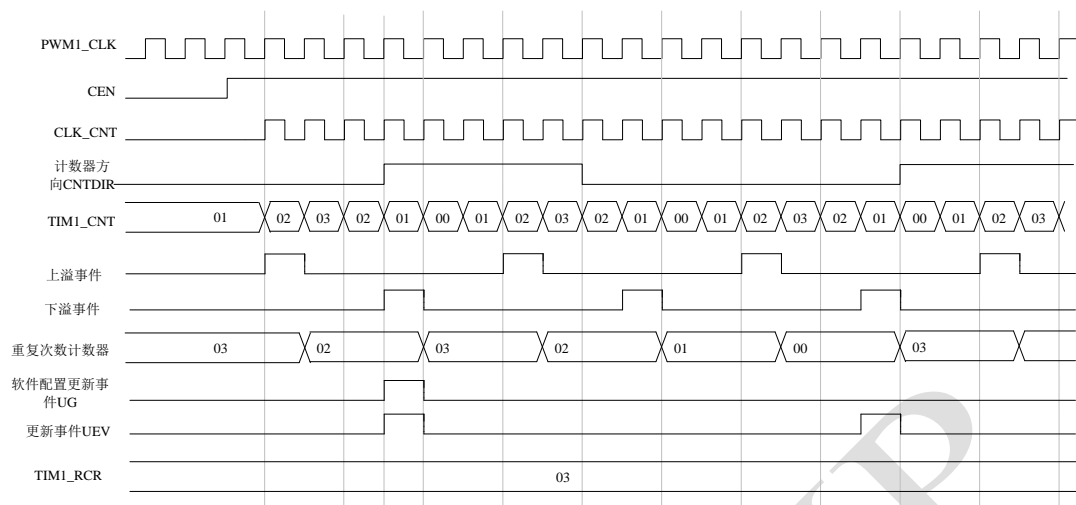


Figure 73 中央对齐模式下，更新事件的产生时序图

11.5.1.6 计数器时钟选择

计数器的时钟可由下列时钟源提供：

- 内部时钟。
- 外部输入脚 PWM1_CH0、PWM1_CH0N 提供的输入信号（外部时钟源模式 1）。

11.5.1.6.1 内部时钟

内部时钟由 CRM 模块提供，最高频率为 73.728MHz，可以通过配置 CRM 模块中的 PWM_CLK_CTRL 寄存器对时钟进行 1/2/4/8 分频，预分频器可以对输入的 pwm1_clk 再进行 1~65536 分频，可以通过配置寄存器 TIM1_PSCL 和 TIM1_PSCH 实现。

将 TIM1_CONR2 寄存器的 SMS 位配置为 000，则从模式控制器被关闭，预分频器的驱动时钟由内部时钟 pwm1_clk 提供。计数器的使能、方向和软更新事件由软件配置确定。当 TIM0_CNTEN 寄存器的 CNTEN 位配置为 1 时，计数器开始计数。

11.5.1.6.2 外部时钟源模式 1

当寄存器 TIM1_TGICR0 的 SMS 位配置为 111 时，外部时钟源模式 1 被选中，计数器可在选定的输入脚 PWM1_CH0、PWM1_CH0N 输入信号的每个上升沿或下降沿计数，通过配置 TIM1_TGICR0 寄存器的 TRGS 位可以选择输入管脚 PWM1_CH0、PWM1_CH0N 的信号。Figure 37 为选择输入脚 PWM1_CH0 的信号作为外部时钟的例子，具体配置步骤如下：

- 外部信号输入后，通过配置寄存器 TIM1_CC0MR 中的 CC0MS 位可以从 CC0 的输入信号 CH0IN 和通道 CC1 的输入信号 CH1IN 中选择一个作为通道 CC0 的输入信号，本例中选择 CH0INF 作为 CC0 的输入信号，本例中选择 CH0IN。
- 将信号 CH0IN 送入边沿检测器做边沿检测，得到上升沿信号 CH0INF_R、下降沿信号 CH0INF_D 和边沿信号 CH0INF_ED（包括上升沿信号和下降沿信号）。
- 通过配置寄存器 TIM1_CCPS 寄存器的 CC0P 位可以选择输入信号的有效沿，当

CC0P 配置为 0 时，选择信号的上升沿 CH0INF_R 作为有效边沿；当 CC0P 配置为 1 时，选择信号的下降沿 CH0INF_D 作为有效边沿，得到边沿选择后的输入信号 CH0INFP，本例中选择上升沿为有效边沿。

- 配置寄存器 TIM1_TGICR0 的 TRGS 位为 101，选择 CH0INFP 作为从模式控制器的输入触发信号。
- 配置寄存器 TIM1_TGICR0 的 SMS 位为 111，使从模式控制器工作在外部时钟模式 1 下。
- 配置寄存器 TIM1_CNTEN 的 CNTEN 位为 1，启动计数器。

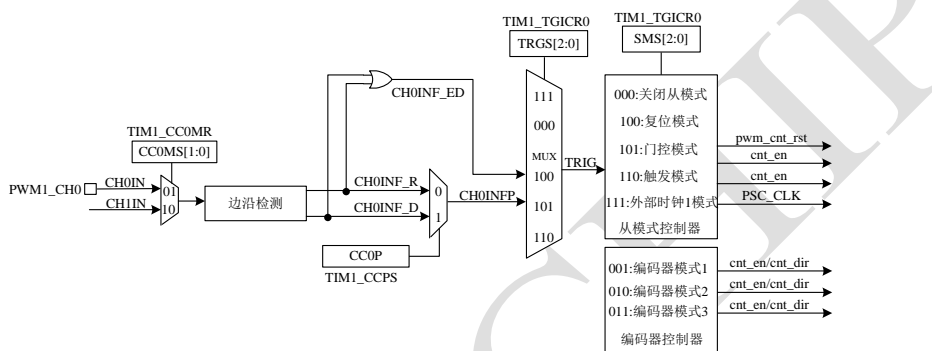


Figure 74 输入脚 PWM1_CH0 的信号作为外部时钟的连接例子

Figure 38 为输入信号 CH0IN 的上升沿触发计数器计数的时序图，当输入信号 CH0IN 出现一次上升沿时，计数器记一次数。输入信号 CH0IN 的上升沿和实际的计数器时钟之间的延时取决于 CH0IN 输入通道的重同步电路。

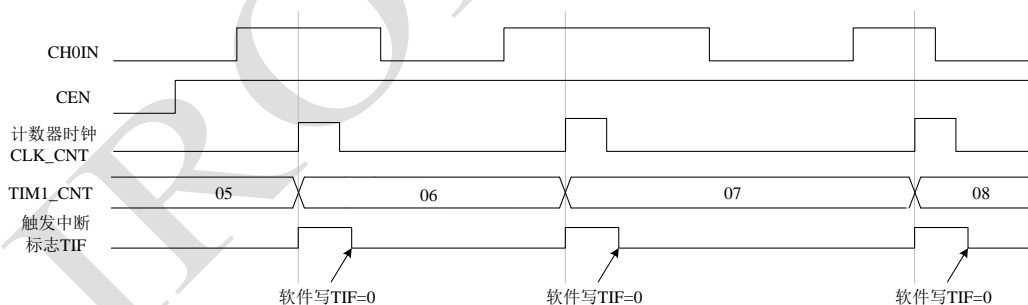


Figure 75 输入脚 PWM1_CH0 的信号作为外部时钟的时序图

11.5.1.7 寄存器更新

11.5.1.7.1 16 位寄存器的读写

本定时器中有些寄存器是 16 位寄存器，而芯片的数据总线是 8 位，为了便于寄存器读写，将 16 位寄存器拆分为两个 8 位寄存器，即高字节寄存器 MS 和低字节寄存器 LS，例如通道 CCx 的捕获比较寄存器 TIM1_CCxR 分为高字节寄存器 TIM1_CCxRH 和低字节寄存器 TIM1_CCxRL。在本定时器中读写 16 位寄存器应遵循以下规则：先读写高字节寄存器 MS

再读写低字节寄存器 LS。

11.5.1.7.2 寄存器的预装载模式和立即更新模式

定时器中有些寄存器有两种更新模式：预装载模式和立即更新模式。

立即更新模式指：当向寄存器写入新值时，新值立即生效。

预装载模式指：具有预装载功能的寄存器有两个寄存器--预装载寄存器和影子寄存器，TIM 使用的是影子寄存器值，而软件只能读写预装载寄存器。

若开启了预装载功能，当软件配置寄存器时，只能将新值写入预装载寄存器，当发生寄存器的更新事件时，硬件才将预装载寄存器中的值装载到影子寄存器，此时新配置的值才会生效。

TIM1 中所有具有预装载功能的寄存器有：TIM1_ARR、TIM1_PSC（无立即更新模式）、TIM1_CCxR、TIM1_CCENR、TIM1_CCPS 和寄存器 TIM1_CCxMR 的 OCxMS 位，根据预装载模式的使能信号和更新信号的不同，可以将这些寄存器分为三类：

- 由寄存器 TIM1_CONR2 的 ARPLE 位控制预装载模式是否使能，由更新事件 UEV 更新寄存器，这些寄存器有：TIM1_ARR、TIM1_PSC（无立即更新模式）。
- 由 TIM1_CCxMR 寄存器的 OCxPEN 位控制预装载模式是否使能，由更新事件 UEV 更新寄存器，这些寄存器有：TIM1_CCxRL、TIM1_CCxRH。
- 由 TIM1_CONR0 寄存器的 CCPE 位控制预装载模式是否使能，在预装载模式下由 COM 事件或从模式的触发输入更新（由 TIM1_CONR0 寄存器的 CCUS 位决定）寄存器，这些寄存器有：TIM1_CCENR、TIM1_CCPS 和 TIM1_CCxMR 寄存器的 OCxMS 位。

具有预装载功能寄存器的预装载功能可以关闭（TIM1_PSC 除外），当预装载功能关闭时，具有预装载功能的寄存器使用立即更新模式。

11.5.2 输入捕获功能

高级定时器包含三个通道，可以分别对三个通道独立控制，每个通道可以配置为输入捕获模式或输出比较模式，下面以通道 CC0 为例介绍当通道配置为输入捕获功能时的使用方法。

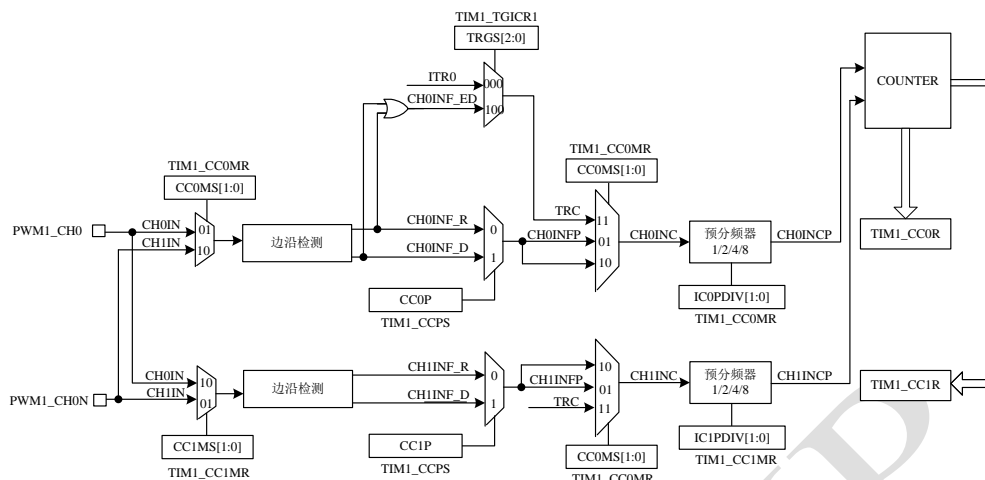


Figure 76 输入捕获通道

如 Figure 41 所示，当寄存器 TIM1_CC0MR 的 CC0MS 位配置为 01/10/11 时，通道 CC0 配置为输入捕获功能，外部输入信号的有效边沿可以捕获计数器的值，并将捕获的值存入通道 CC0 的输入捕获寄存器 TIM1_CC0R 中，具体控制方法如下：

- 配置 GPIO 模块中的寄存器，将相应的管脚配置为高级定时器的信号输入输出功能。
- 通过配置寄存器 TIM1_CC0MR 中的 CC0MS 位可以选 CH0IN 或通道 CC1 的输入信号 CH1IN 作为通道 CC0 的输入信号。
- 将滤波后的信号送入边沿检测器做边沿检测，得到上升沿信号 CH0INF_R、下降沿信号 CH0INF_D 和边沿信号 CH0INF_ED（包括上升沿信号和下降沿信号）。
- 将边沿信号 CH0INF_ED 送入从模式控制器，通过配置寄存器 TIM1_TGICR0 的 TRGI 位可以从 CH0INF_ED 和 ITR0 中选择一个信号作为触发捕获信号 TRC。通过配置寄存器 TIM1_CCPS 的 CC0P 位可以选择输入信号的有效沿，当 CC0P 配置为 0 时，选择信号的上升沿 CH0INF_R 作为有效边沿；当 CC0P 配置为 1 时，选择信号的下降沿 CH0INF_D 作为有效边沿，得到边沿选择后的输入信号 CH0INFP。
- 通过配置寄存器 TIM1_CC0MR 中的 CC0MS 位可以从 TRC 信号和 CH0INFP 中选择一个作为通道 CC0 原始的捕获信号 CH0INC。
- 配置寄存器 TIM1_CC0MR 的 ICPDIV 位，对原始的捕获信号 CH0INC 进行预分频，得到输入捕获信号 CH0INCP。
- 将输入捕获信号 CH0INCP 送入计数器部分进行数据捕获，捕获后将数据存入输入捕获寄存器 TIM1_CC0RU。

以上为 CC0 通道的输入捕获模式的控制步骤，CC1 和 CC2 通道的控制步骤相同。在软件访问 16 位寄存器时，先访问高 8 位部分再访问低 8 位部分，按照此规则，在软件读取 TIM1_CC0R 中的捕获值时，先读取寄存器 TIM1_CC0RH 再读取 TIM1_CC0RL，Figure 42

为软件读取捕获寄存器 TIM1_CC0R 的过程。

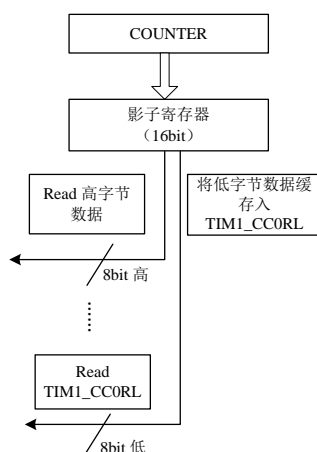


Figure 77 向软件读取捕获寄存器

为了防止在读取捕获值的高 8 位部分时再次发生捕获, 软件一旦开始读取高 8 位部分数据时, 硬件将低 8 位数据从影子寄存器缓存入预装载寄存器 TIM1_CC0RL。

当产生捕获信号 CH0INCP 时, 寄存器 TIM1_ISR0 中捕获中断标志位 CC0IF 置位, 若捕获/比较中断使能打开, 则产生捕获中断。读取捕获值后捕获中断标志位 CC0IF 自动清除, 也可以通过软件配 0 将捕获中断标志位 CC0IF 清除。

11.5.3 从模式控制

11.5.3.1 从模式控制器的信号源

从模式是指高级定时器受外部触发信号驱动的工作模式, Figure 43 所示为从模式控制器, 外部触发信号的来源有:

- 通道 CC0 的双边沿信号 CH0IN_ED, 产生过程见 10.5.2 部分说明。
- 通道 CC0 的单边沿信号 CH0INFP, 产生过程见 10.5.2 部分说明, 此信号一方面送入从模式控制器驱动高级定时器工作, 另一方面作为编码器控制器的输入, 在编码器接口模式下控制计数器计数。
- 通道 CC1 的单边沿信号 CH1INFP, 产生过程见 10.5.2 部分说明, 此信号一方面送入从模式控制器驱动高级定时器工作, 另一方面作为编码器控制器的输入, 在编码器接口模式下控制计数器工作。

其中, TIM1_TRGO 和 CH0IN_ED 信号还可以作为捕获信号。

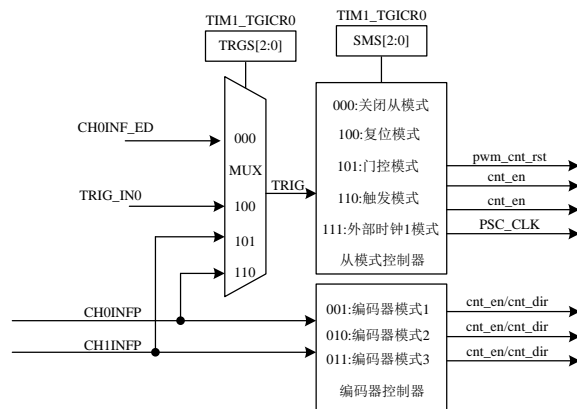


Figure 78 从模式控制器示意图

11.5.3.2 从模式控制

配置寄存器 TIM1_TGICR0 的 TRGS 位，从 CH0INF_ED、CH0INFP、TRIG_IN0 和 CH1INFP 中选择一个作为触发信号送入从模式控制器，在不同的控制模式下驱动高级定时器工作。对高级定时器的控制模式有：

- 关闭从模式：将寄存器 TIM1_TGICR0 的 SMS 位配置为 000，关闭从模式，此时预分频器的驱动时钟由内部时钟 pwm1_clk 提供。计数器的使能、方向和软更新事件由软件配置确定。当 TIM1_CNTEN 寄存器的 CNTEN 位配置为 1 时，计数器开始计数。
- 复位模式：将寄存器 TIM1_TGICR0 的 SMS 位配置为 100 时，从模式控制器工作在复位模式下，选中的触发信号 TRGI 的有效沿重新初始化计数器，并且产生一个更新事件来更新寄存器。若 TIM1_CONR2 的 URC 位配置为 0，则寄存器 TIM1_ISR0 的中断标志 SOFTUIF 置位，若软更新中断使能打开，则产生一个软更新中断，此时，相当于配置了寄存器 TIM1_EGR 的 UG 事件；若 TIM1_CONR2 的 URC 位配置为 1，则在触发信号 TRGI 的有效沿仅初始化计数器，并且产生一个更新事件来更新寄存器，而不产生中断和中断标志。
- 门控模式：将寄存器 TIM1_TGICR0 的 SMS 位配置为 101 时，从模式控制器工作在门控模式下，高级定时器的计数器使能受触发信号 TRGI 的控制，当 TRGI 为有效电平时，计数器开始计数，当 TRGI 为无效电平时，计数器停止计数（但不复位）。此时 TIM1_CNTEN 的 CNTEN 位仍需要配置为 1，否则，即使 TRGI 为有效电平，计数器也停止计数。
- 触发模式：将寄存器 TIM1_TGICR0 的 SMS 位配置为 110 时，从模式控制器工作在触发模式下，选中的触发信号 TRGI 的有效沿启动计数器计数（但不复位）。此时 TIM1_CNTEN 的 CNTEN 位不需要配置为 1，触发信号 TRGI 的有效沿会将 CNTEN 位置 1。计数器启动后配置 TIM1_CNTEN 的 CNTEN 位为 0，可以停止计

数器计数。

- 外部时钟源模式 1：将寄存器 TIM1_TGICR0 的 SMS 位配置为 111 时，选择触发信号 TRGI 的有效沿驱动计数器计数。此时 TIM1_CNTEN 的 CNTEN 位需要配置为 1。

有三点需要注意：

第一，在 CH0IN_ED 被选为触发信号 TRGI 时，不要使用门控模式，因为 CH0IN_ED 是一个脉冲信号，而门控模式时要检查触发输入的电平。

编码器接口模式

使用通道 CC0 和 CC1 作为正交增量编码器两路信号的接口，在通道 CC0 和 CC1 对信号做必要的处理后得到信号 CH0INFP 和 CH1INFP，再将 CH0INFP 和 CH1INFP 直接送入编码器控制器，在编码器控制器中根据两路信号驱动计数器计数。

编码器接口模式的配置步骤为：

- 根据实际需要配置寄存器 TIM1_CCPS 的 CC0P 位，对输入的编码器信号进行极性选择，得到信号 CH0INFP。
- 根据实际需要配置寄存器 TIM1_CCPS 的 CC1P 位，对输入的编码器信号进行极性选择，得到信号 CH1INFP。
- 配置寄存器 TIM1_TGICR0 的 SMS 位：若将 SMS 位配置为 001，定时器工作在编码器模式 1 下，计数器以 CH0INFP 作为相对信号由 CH1INFP 的边沿驱动计数；若将 SMS 位配置为 010，定时器工作在编码器模式 2 下，计数器以 CH1INFP 作为相对信号由 CH0INFP 的边沿驱动计数；若将 SMS 位配置为 011，定时器工作在编码器模式 3 下，计数器以 CH1INFP 和 CH0INFP 作为相对信号由 CH0INFP 和 CH1INFP 的边沿驱动计数。计数器的计数方向也在编码器控制器中根据一定的规则产生。
- 配置寄存器 TIM1_CCENR 的 CC0EN 和 CC1EN 位为 1，使能通道 CC0 和通道 CC1。
- 将定时器配置为编码器模式后，需要将 TIM1_CTEN 寄存器的 CTEN 位置 1 来启动计数器。

假设计数器已经使能，则计数器由 CH0INFP 和 CH1INFP 的有效跳变沿驱动。编码器控制器根据两个输入信号的跳变顺序，产生计数器的计数脉冲和计数方向，使计数器向上或向下计数，无论计数器是由 CH0INFP 驱动计数还是由 CH1INFP 驱动计数或者由 CH0INFP 和 CH1INFP 的边沿同时驱动，任一输入端跳变都会重新计算计数器的方向。计数器的计数方向产生规则如 Table 101 所示，表中相对信号的含义为：若计数器由 CH0INFP 的有效跳变沿驱动计数时，相对信号为 CH1INFP；若计数器由 CH1INFP 的有效跳变沿驱动计数时，相对信号为 CH0INFP；若计数器由 CH0INFP 和 CH1INFP 的有效跳变沿同时驱动计数时，互为相对信号。

Table 143 增量编码器接口模式下计数器计数规则

有效边沿	相对信号的 电平	CH0INFP 信号		CH1INFP 信号	
		上升沿	下降沿	上升沿	下降沿
仅在 CH0INFP 边沿计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在 CH1INFP 边沿计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在 CH0INFP 和 CH1INFP 边沿计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

编码器接口模式相当于使用了一个带有方向选择的外部时钟，这意味着计数器只有在 0 到 TIM1_ARR 寄存器的自动装载值之间连续计数（根据 Table 101 增量编码器接口模式下计数器计数规则中的方向，或者是 0 到 TIM1_ARR 计数，或者是 TIM1_ARR 到 0 计数），所以在开始计数之前必须配置 TIM1_ARR。

在这种模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置，计数器方向与相连的传感器旋转的方向对应。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发一个计数器复位。Figure 44 为在增量编码器模式 3 下对输入信号 CH0INFP 和 CH1INFP 的编码。

例子：在增量编码器模式下，计数器在输入信号 CH0INFP 和 CH1INFP 的驱动下计数器，计数过程如 Figure 44 所示。

- 配置 GPIO 模块寄存器，将相应的引脚复用为高级定时器的输入管脚 PWM1_CH0 和 PWM1_CH1，并且将增量编码器的输出信号连接至 PWM1_CH0 和 PWM1_CH1，这样增量编码器的信号由定时器的通道 CC0 和 CC1 输入。
- 配置寄存器 TIM1_CCENR 的 CC0EN 和 CC1EN 位为 1，使能通道 CC0 和 CC1 输入。
- 配置寄存器 TIM1_CCPS 的 CC0P 和 CC1P 位为 0，通道 CC0 和 CC1 输入信号不反相。
- 配置寄存器 TIM1_TGICR0 的 SMS 位配置为 011，将定时器从模式配置为编码器模式 3。
- 配置寄存器 TIM1_CNTEN 的 CNTEN 位为 1，使能计数器。

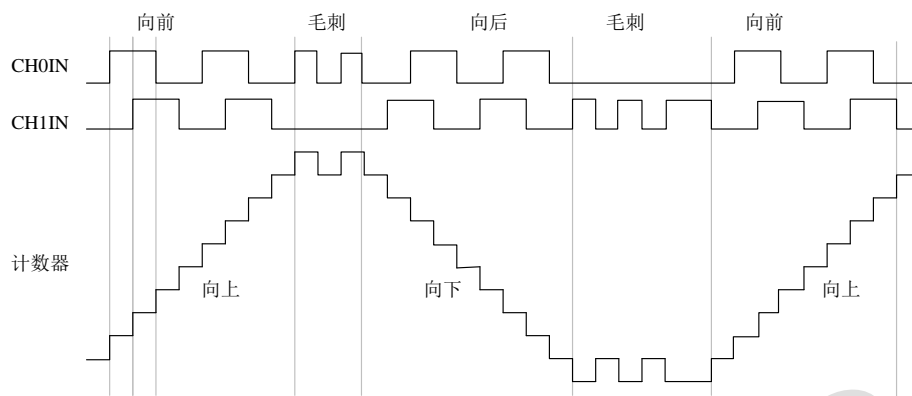


Figure 79 高级定时器作为增量编码接口的例子

11.5.3.3 PWM 输入模式

PWM 输入模式是输入捕获模式的一个特例,可以测量一个 PWM 信号的占空比和周期,假设 PWM 信号由通道 CC0 输入,具体配置如下:

- 将 TIM1_ARR 配置为最大值。
- 配置寄存器 TIM1_CCPS 的 CC0P 位为 0, CC0 通道使用信号的上升沿捕获数据。
- 配置寄存器 TIM1_CC0MR 的 CC0MS 位为 01, 选择 CH0INF 的上升沿捕获数据。
- 不对捕获信号分频, 配置寄存器 TIM1_CC0MR 的 IC0PDIV 位为 00。
- 配置寄存器 TIM1_CCPS 的 CC1PS 位为 1, 配置寄存器 TIM1_CC0MR 的 CC0MS 位为 10, 选择通道 CC0 信号 CH0INF 的下降沿作为通道 CC1 的捕获信号。
- 配置寄存器 TIM1_TGICR0 的 TRGS 位为 101, 选择 CH0INFP 作为从模式的触发信号 TRGI。
- 配置寄存器 TIM1_TGICR0 的 SMS 位为 100, 将从模式配置为复位模式, CH0INFP 的上升沿将计数器清零。
- 配置寄存器 TIM1_CCENR 的 CC0EN 位和 CC1EN 位为 1, 使能通道 CC0 和通道 CC1 的捕获使能。
- 配置寄存器 TIM1_EGR 的 UG 位和 COMG 位为 1, 初始化计数器并更新相应寄存器。
- 配置寄存器 TIM1_CNTEN 的 CNTEN 位为 1, 启动计数器。

因为只有 CH0INFP 和 CH1INFP 连接到了从模式控制器,所以 PWM 输入模式只能使用 CC0 和 CC1 通道。PWM 输入模式捕获时序如 Figure 45 所示。

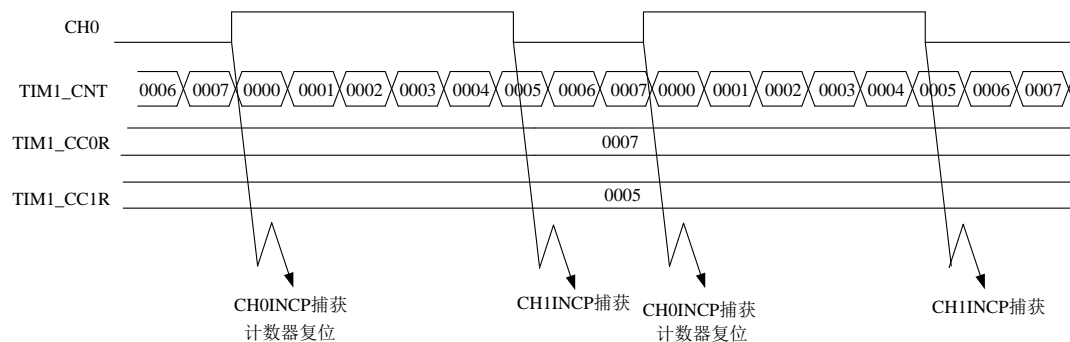


Figure 80 PWM 输入模式时序图

11.5.4 输出比较功能

将寄存器 TIM1_CC0MR 的 CC0MS 位配置为 00，通道 CC0 工作在输出比较模式下。在输出比较模式下，可以独立配置各自的工作模式。通道 CC0 为比较输出时可以输出两路互补的 PWM 波形，输出控制过程如 Figure 46 所示。

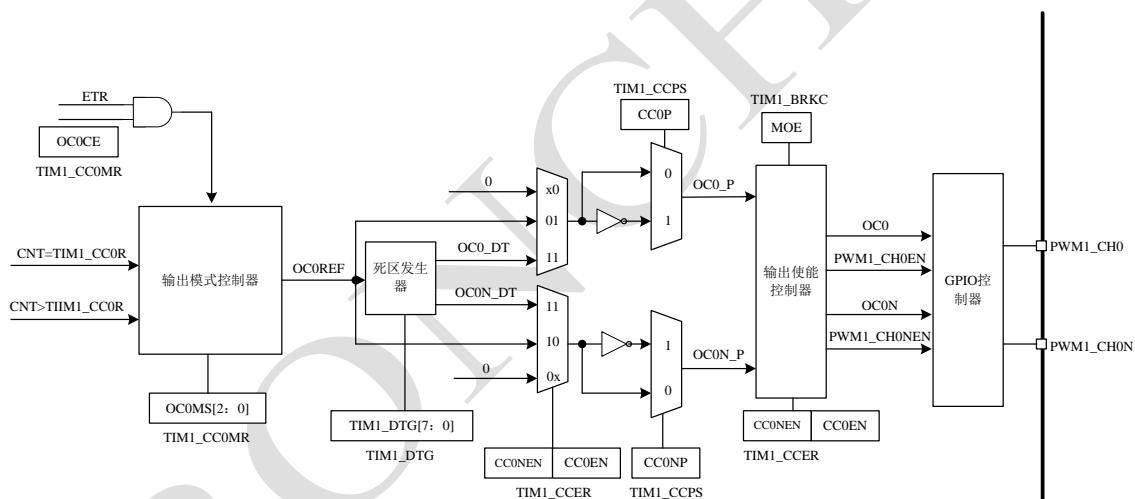


Figure 81 通道 CC0 输出控制

通道 CC0 的输出控制过程如下：

- 配置寄存器 TIM1_CC0MR 的 CC0MS 位为 00，将 CC0 通道配置为输出功能。
- 配置寄存器 TIM1_CC0MR 的 OC0MS 位域，选择输出模式，可选的输出模式有：冻结模式、匹配时输出有效电平、匹配时输出无效电平、强制输出有效电平、强制输出无效电平、翻转、PWM 模式 1、PWM 模式 2，各种模式的详细描述见 10.5.4.1 节，输出基准参考信号 OC0REF，OC0REF 高电平为有效电平、低电平为无效电平。
- 若输出模式配置为 PWM 模式 1 或 PWM 模式 2 且主路输出和互补输出的使能均有效（CC0ENN=1，CC0NEN=1），则根据寄存器 TIM1_DTG 的配置插入死区，输出两路互补信号 OC0_DT 和 OC0N_DT。若两路使能不同时有效，则不插入死区，直接在使能打开的一路输出 OC0REF，使能关闭的一路输出低电平。

- 对输出信号进行极性控制，配置寄存器 TIM1_CCPS 的 CC0P 位为 0 时，主路输出高电平为有效电平，当 CC0P 位为 1 时，主路输出低电平为有效电平，极性控制后得到输出信号 OC0_P；配置寄存器 TIM1_CCPS 的 CC0NP 位为 0 时，互补路输出高电平为有效电平，当 CC0NP 位为 1 时，互补路输出低电平为有效电平，极性控制后得到输出信号 OC0N_P。
- 根据寄存器 TIM1_BRKC 中 MOE 位、寄存器 TIM1_CCER 中的 CC0EN、CC0NEN 位的配置控制主路输出和互补路的输出，并控制主路和互补路的输出使能，详细控制方法见 10.5.4.4 节所述。最终从高级定时器中输出信号 OC0 和 OC0N 以及两路的输出使能信号 PWM1_CH0EN 和 PWM1_CH0NEN 至 GPIO 模块，在 GPIO 模块中选择 PWM 输出功能将 OC0 和 OC0N 输出到相应的管脚。
- 若计数器与输出比较寄存器 TIM1_CC0R 比较成功，则寄存器 TIM1_ISR0 的中断标志位 CC0IF 置位。若设置了通道 CC0 的输入输出中断使能，则产生通道 CC0 的输出比较中断。

11.5.4.1 输出模式控制

11.5.4.1.1 强置输出模式

在输出模式下(寄存器 TIM1_CC0MR 的 CC0MS 位配置为 00)，输出比较信号(OC0REF 和相应的 OC0/OC0N)能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器的比较结果。

将寄存器 TIM1_CC0MR 的 OC0MS 位配置为 101，可将输出比较信号(OC0REF/OC0)强置为有效状态，这样 OC0REF 被强置为高电平(OC0REF 始终为高电平有效)，同时 OCx 得到与寄存器 TIM1_CCPS 的 CC0P 位相反的信号。例如，CC0P 位配置为 1(将 OC0 配置为低电平有效)，则 OC0 被强置为低电平(与 CC0P 值相反)。

将寄存器 TIM1_CC0MR 寄存器的 OC0MS 位配置为 100，可将输出比较信号(OC0REF/OC0)强置为无效状态，这样 OC0REF 被强置为低电平(OC0REF 始终为高电平有效)，同时 OC0 得到与寄存器 TIM1_CCPS 的 CC0P 位相同的信号，例如，CC0P 配置为 1(将 OC0 配置为低电平有效)，则 OC0 被强置为高电平(与 OC0P 的值相同)。

在强置输出模式下，比较寄存器 TIM1_CC0R 和计数器之间的比较仍在进行，相应的标志也会被修改。因此仍然会产生相应的中断标志和中断请求。

Figure 48 为当高级定时器配置为强置输出模式时的时序图。

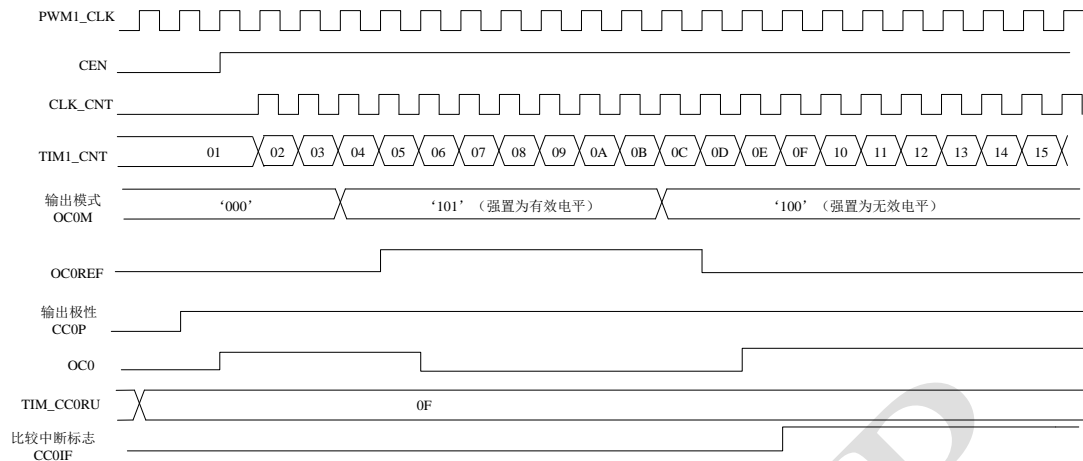


Figure 82 高级定时器强置输出模式时序图

11.5.4.1.2 输出比较模式

当计数器与比较寄存器的内容相同时，输出比较功能做如下操作：

- 定义基准参考波形 OC0REF 和定时器输出 OC0 的变化，当计数器与比较寄存器匹配时 ($TIM1_CNT = TIM1_CC0R$):
 - 若寄存器 TIM1_CC0MR 的 OC0MS 位配置为 000, 则 OC0REF 保持不变, OC0 保持不变。
 - 若寄存器 TIM1_CC0MR 的 OC0MS 位配置为 001, 则 OC0REF 被设置为高电平, 输出 OC0 被设置为有效电平, 即若 CC0P=0, 则 OC0 输出高电平, 若 CC0P=1, 则 OC0 输出低电平。
 - 若寄存器 TIM1_CC0MR 的 OC0MS 位配置为 010, 则 OC0REF 被设置为低电平, 输出 OC0 被设置为无效电平, 即若 CC0P=0, 则 OC0 输出低电平, 若 CC0P=1, 则 OC0 输出高电平。
 - 若寄存器 TIM1_CC0MR 的 OC0MS 位配置为 011, 则 OC0REF 波形翻转, 输出 OC0 的电平翻转。
- 设置中断状态寄存器 TIM1_ISR0 中的中断标志位 CC0IF。
- 若设置了相应的中断使能, 则产生一个比较中断。

在输出比较模式下, 更新事件 UEV 对 OC0REF 输出没有影响。

输出比较模式的配置步骤为:

- 选择计数器时钟和预分频器 (内部时钟, 外部时钟, 预分频器)。
- 配置寄存器 TIM1_CONR2 的 ARPLE 位和 TIM1_CC0MR 的 OC0PEN 位, 确定 TIM1_ARR 寄存器、TIM1_CC0R 寄存器的预装载功能是否打开。
- 配置自动装载寄存器 TIM1_ARR、捕获比较寄存器 TIM1_CC0R, 都是遵循先配置高 8 位再配置低 8 位的规则。

- 配置寄存器 TIM1_CC0MR 的 OC0MS 位，选择输出模式，配置 TIM1_CCPS 寄存器，选择输出极性。
- 配置寄存器 TIM1_CCENR，使能相应的输出通道。
- 若要产生一个输出比较中断请求，设置寄存器 TIM1_IER 的 CC0IE 位。
- 配置寄存器 TIM1_EGR 的 UG 位和 COMG 位为 1，更新具有预装载功能的寄存器。
- 设置 TIM1_CNTEN 寄存器的 CNTEN 位，启动计数器。

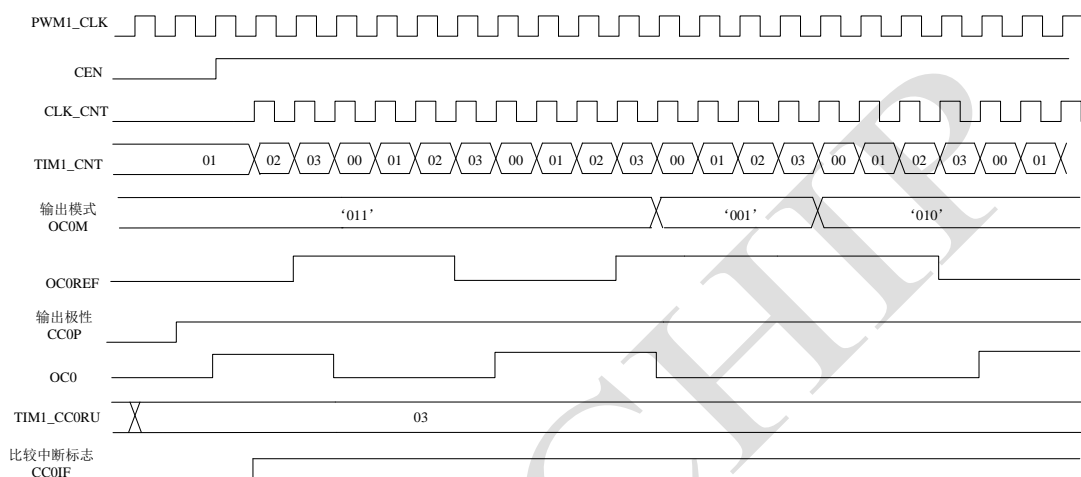


Figure 83 高级定时器比较输出模式

11.5.4.1.3 PWM 输出模式

脉冲宽度调制模式(PWM)可以产生一个由 TIM1_ARR 寄存器确定周期、由 TIM1_CC0R 寄存器确定占空比的信号。

配置 TIM1_CC0MR 寄存器中的 OC0MS 位为“110”，定时器输出 PWM 模式 1 的波形；配置 TIM1_CC0MR 寄存器中的 OC0MS 位为“111”，定时器输出 PWM 模式 2 的波形。

TIM1_CC0MR 寄存器 OC0PEN 位控制比较寄存器 TIM1_CC0R 的预装载使能。TIM1_CONR2 的 ARPLE 位控制自动重装载寄存器 TIM1_ARR 的预装载使能。若使能了预装载功能，仅当发生一个更新事件时，预装载寄存器才能被传送到影子寄存器，因此在计数器计数之前，必须通过设置 TIM1_EGR 寄存器的 UG 位来更新具有预装载功能寄存器。

输出波形的极性可以通过软件在 TIM1_CCPS 中设置，可以设置高电平为有效电平或低电平为有效电平。高级定时器在管脚上的输出 PWM1_CH0 或 PWM1_CH0N 通过 CC0EN、CC0NEN、MOE 位的组合控制，详见 10.5.4.4 节的描述。

PWM 模式分为 PWM 模式 1 和 PWM 模式 2，PWM 模式 1 波形产生的规则为：当 $TIM1_CNT < TIM1_CC0R$ 时，PWM 基准参考信号 OC0REF 为有效电平（高电平），否则为无效电平（低电平）；PWM 模式 2 波形产生的规则为：当 $TIM1_CNT < TIM1_CC0R$ 时，PWM 基准参考信号 OC0REF 为无效电平（低电平），否则为有效电平（高电平）；

下面以 PWM 模式 1 为例说明在不同的计数器计数模式下 PWM 输出控制。

● 向上计数模式下的 PWM 模式 1

当 TIM1_CONR2 寄存器中的 CNTMC 位配置为 00 且 CNTDIR 位配置为 0 时，计数器向上计数。此时，PWM 模式 1 的输出为：当 TIM1_CNT < TIM1_CC0R 时，PWM 基准参考信号 OC0REF 为有效电平（高电平），否则为无效电平（低电平）。如果 TIM1_CC0R 中的比较值大于自动重装载值，则 OC0REF 保持为有效电平（高电平）。如果比较值为 0，则 OC0REF 保持为无效电平（低电平）。Figure 50 为 TIM1_ARR=8 且计数器向上计数时 PWM 模式 1 的输出波形。

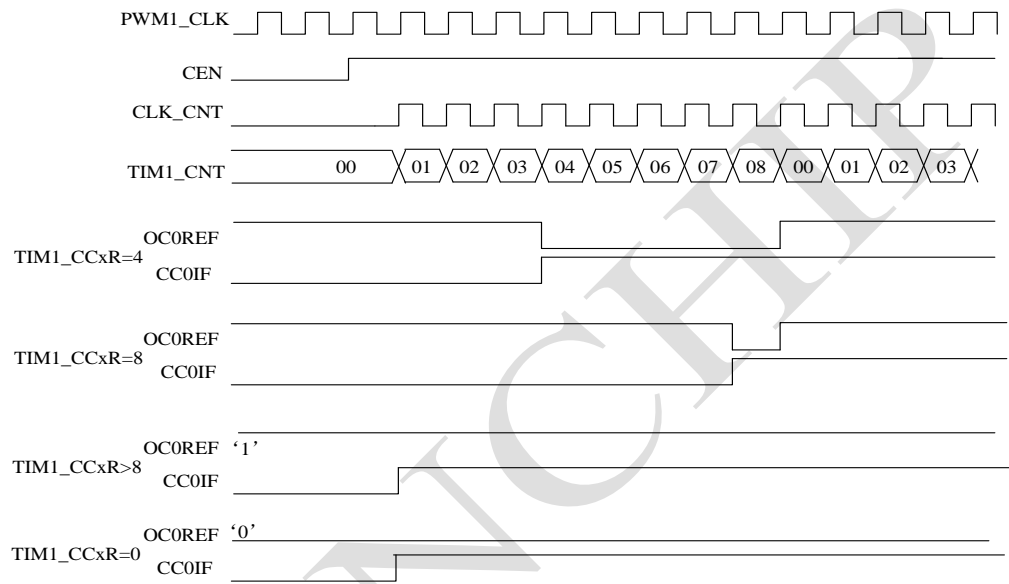


Figure 84 向上计数模式下，PWM 模式 1 输出

● 向下计数模式下的 PWM 模式 1

当 TIM1_CONR2 寄存器中的 CNTMC 位配置为 00 且 CNTDIR 位配置为 1 时，计数器向上计数。此时，PWM 模式 1 的输出为：当 TIM1_CNT > TIM1_CC0R 时，PWM 基准参考信号 OC0REF 为无效电平（低电平），否则为有效电平（高电平）。如果 TIM1_CC0R 中的比较值大于自动重装载值，则 OC0REF 保持为有效电平（高电平）。该模式下不能产生 0% 的 PWM 波形。Figure 51 为 TIM1_ARR=8 时向上计数器时 PWM 模式 1 的输出波形。

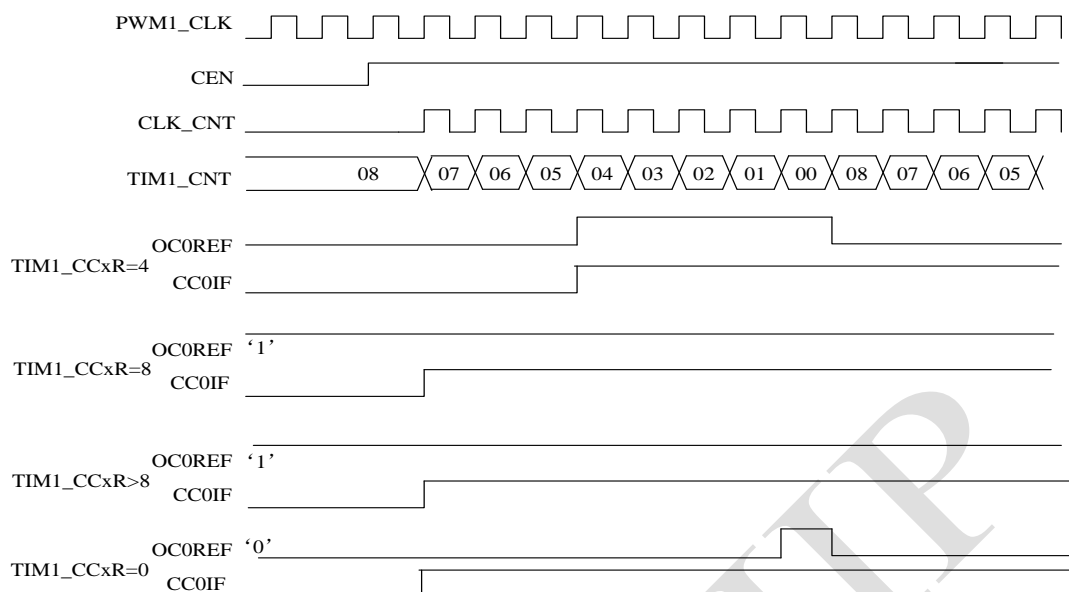


Figure 85 向下计数模式下，PWM 模式 1 输出

● 中央对齐计数模式下的 PWM 模式 1

当 TIM1_CONR2 寄存器中的 CNTMC 位配置为 01/10/11 时，计数器工作在中央对齐模式下，TIM1_CONR2 寄存器中的计数方向 CNTDIR 不能用软件配置。

在此种计数模式下，PWM 模式 1 输出波形规则为：在向上计数过程中，当 $TIM1_CNT < TIM1_CCxR$ 时，PWM 基准参考信号 OC0REF 为有效电平（高电平），否则为无效电平（低电平）；在向下计数过程中，当 $TIM1_CNT > TIM1_CCxR$ 时，PWM 基准参考信号 OC0REF 为无效电平（低电平），否则为有效电平（高电平）。当 CNTMC 位配置为 01 时，比较标志只在计数器向上计数时被置位；当 CNTMC 位配置为 10 时，比较标志只在计数器向下计数时被置位；当 CNTMC 位配置为 11 时，比较标志在计数器向上和向下计数时都被置位。Figure 52 为中央对齐的 PWM 模式 1 输出波形，图中箭头表示将比较中断标志置位。

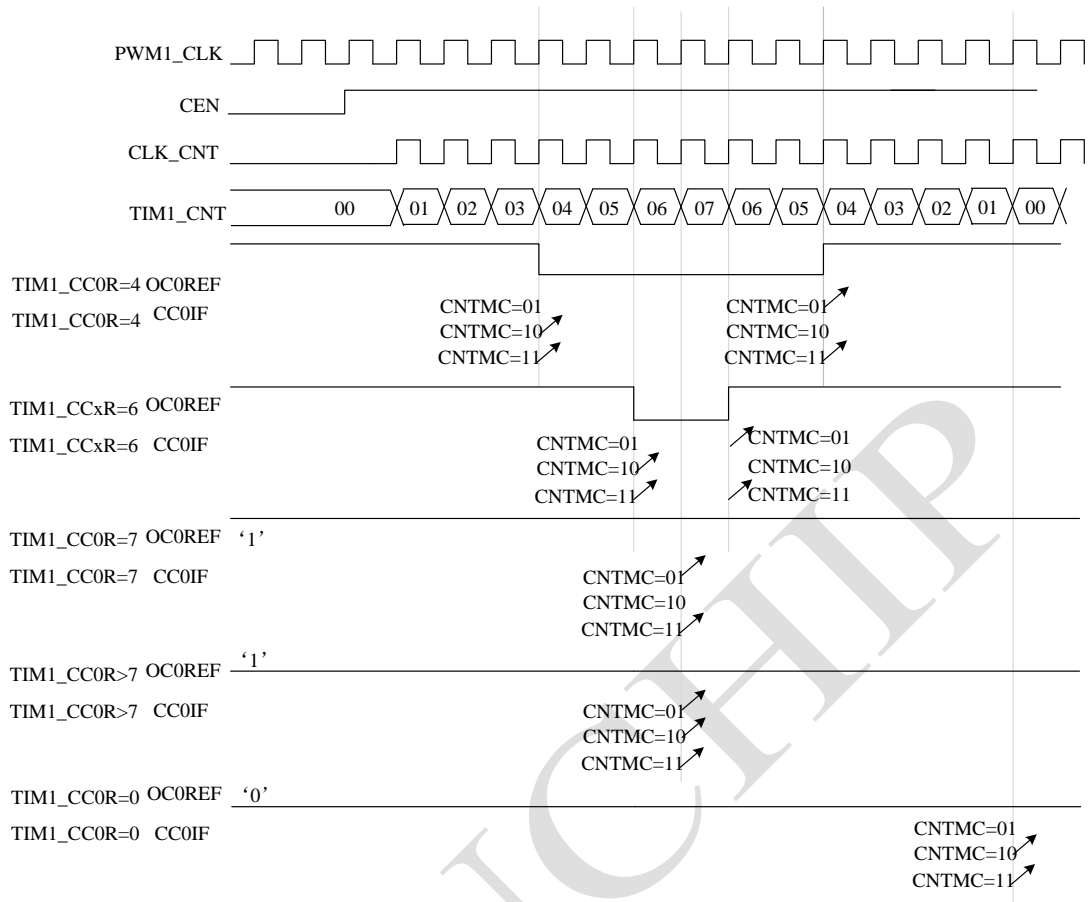


Figure 86 中央对齐计数模式下，PWM 模式 1 输出

使用中央对齐模式的注意事项:

- 在中央对齐计数模式下，若配置了计数器的复位功能（通过配置 TIM_EGR 寄存器的 UG 位或通过从模式控制器产生复位信号），计数器开始向上计数，若没有配置计数器的复位功能，计数器按照以前保留的计数方向开始计数。此外，CNTDIR 和 CNTMC 位不能同时被软件修改，在边沿计数模式和中央对齐计数模式间转换时需按照以下步骤配置计数模式控制寄存器：
 - 从边沿计数模式向中央对齐计数模式转换
首先，配置寄存器 TIM1_CONR2 的 CNTDIR 位，若希望中央计数模式下首先向下计数，则配置 CNTDIR 位为 1，若希望中央计数模式下首先向上计数，则配置 CNTDIR 位为 0，若希望中央计数模式下首先向下计数，将 CNTDIR 位配置为 1 后不可再配置 TIM1_EGR 寄存器的 UG 位。
然后，配置寄存器 TIM1_CONR2 的 CNTMC 位为 01/10/11。
 - 从中央对齐计数模式向边沿计数模式转换
首先，配置寄存器 TIM1_CONR2 的 CNTMC 位为 00，将计数器切换至边沿计数模式。

然后，配置寄存器 TIM1_CONR2 的 CNTDIR 位，确定计数方向。

- 不建议当运行在中央对齐模式时改写计数器，因为会产生不可预知的结果：
 - 如果写入计数器的值大于自动重加载的值，则方向不会被更新，若计数器正在向上计数，计数器将继续向上计数。
 - 如果将 0 或者 TIM1_ARR 的值写入计数器，方向被更新，但是不产生更新事件。
- 使用中央对齐模式推荐的方法为：在启动计数器之前产生一个软件更新（设置 TIM1_EGR 寄存器中的 UG 位），不要在计数过程中修改计数器的值。

11.5.4.2 互补输出和死区插入

高级定时器能够输出两路互补信号 OC0_DT 和 OC0N_DT（相反的两路信号），也能控制输出在一段时间内无效，然后再接通，这段时间通常称为死区，本定时器的死区时间是可编程的，在具体应用中应该根据连接到输出的器件和他们的特性（电平转换的延时、电源开关的延时等）来调整死区时间。

互补输出 OC0_DT 和 OC0N_DT 在参考信号 OC0REF 基础上产生，定时器通过以下方式控制互补输出 OC0_DT 和 OC0N_DT：

- 配置 TIM1_CCPS 寄存器的 CC0P 和 CC0NP 位，可以为每一路 PWM 输出独立地选择极性；
- 通过配置 TIM1_CCENR 寄存器的 CC0EN 和 CC0NEN 位，控制互补输出的使能。
- 当定时器工作在不同的工作状态时（如刹车状态），互补输出还受到 TIM1_BRKC 寄存器的 MOE 位控制，详见 10.5.4.4 章节。

同时设置 CC0E、CC0NE 位，则 PWM 输出将插入死区。每一个通道都有一个死区发生器，定时器根据参考信号 OC0REF 产生两路带死区的互补输出 OC0_DT 和 OC0N_DT，若两路互补输出都配置为高电平有效：

- 输出信号 OC0_DT 与参考信号 OC0REF 相同，只是他的上升沿相对于参考信号的上升沿有一个延迟。
- 输出信号 OC0N_DT 与参考信号 OC0REF 相反，只是他的上升沿相对于参考信号的下降沿有一个延迟。

如果死区时间大于当前有效的输出宽度，则不会产生相应的脉冲。

Figure 53—Figure 55 显示了带死区的互补输出与当前参考信号 OC0REF 之间的关系。

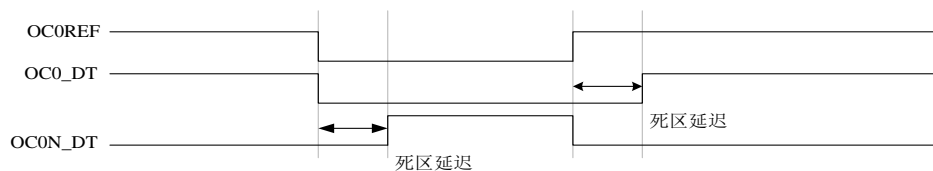


Figure 87 正常死区插入



Figure 88 死区时间大于 OC0REF 无效电平宽度时的死区插入



Figure 89 死区时间大于 OC0REF 有效电平宽度时的死区插入

值得注意的是：只要同时设置 CC0E 和 CC0NE 位为 1，无论在 PWM 输出模式下还是在非 PWM 输出模式下(强置输出模式和输出比较模式)，OC0 和 OC0N 均输出互补的波形，因此，在非 PWM 输出模式下，若不需要输出互补的波形，只需使能主路输出即可（即只设置 CC0E）。

11.5.4.3 输出使能控制

高级定时器最终向引脚输出第一通道的两路输出使能信号 PWM1_CH0EN、PWM1_CH0NEN 和第一通道互补输出 OC0、OC0N。

若输出使能信号 PWM1_CH0EN 为 1，则相应的引脚输出 OC0，若 PWM1_CH0EN 为 0，则相应的引脚不输出 OC0，而是输出高阻；输出使能信号 PWM1_CH0NEN 对引脚的控制同理。

输出使能信号 PWM1_CH0EN 和 PWM1_CH0NEN 受到 TIM1_BRKC 寄存器的 MOE、和 TIM1_CCENR 寄存器的 CC0EN、CC0NEN 位控制，具体控制方式为：若 MOE 位为 1，当 CC0E/CC0NE 配置为 0 时，相应管脚的输出使能 PWM1_CH0EN/PWM1_CH0NEN 为 0，即管脚的输出使能关闭，此时管脚输出高阻；当 CC0E/CC0NE 配置为 1，则相应管脚的输出使能 PWM1_CH0EN/PWM1_CH0NEN 为 1，管脚的输出使能有效，管脚输出 PWM 模块的输出值 OC0/OC0N。若 MOE 位为 0，无论 CC0E/CC0NE 配置为何值，相应管脚的输出使能 PWM1_CH0EN/PWM1_CH0NEN 为 0，即管脚的输出使能关闭，此时管脚输出高阻。

管脚上的 PWM 输出使能和 PWM 输出受 MOE、CC0E、CC0NEN、CC0P、CC0NP 和 CEN 控制，这些寄存器初始值都为 0，所以初始状态下 PWM 管脚输出高阻，为了保证 PWM 平稳启动，建议按照如下顺序配置寄存器：

- 1) 首先配置极性寄存器 TIM1_CC0PS。
- 2) 配置 TIM1_CONR1 的 ARPLE 位和 TIM1_CONR0 的 CCPE 位，控制相应寄存器的

预装载功能。

3) 配置各个通道输入捕获/输出比较模式控制寄存器 TIM1_CC0MR。

4) 配置周期寄存器 TIM1_ARR、比较/捕获寄存器 TIM1_CC0R、通道使能寄存器 TIM1_CC0EN、死区控制等各种寄存器。

5) 配置软更新事件，将新配置的参数装载到影子寄存器中（配置 TIM1_EGR 的 UG 位为 1，COMG 位为 1，装载各个预装载寄存器）。

6) 配置 TIM1_BRKC 寄存器，对 MOE 位进行配置。

7) 配置 TIM1_CNTEN 寄存器的 CNTEN 位，启动 PWM 计数器。

11.5.4.4 刹车功能

当使用刹车功能时，依据相应的控制位（TIM1_BRKC 寄存器的 MOE），输出使能和输出信号都会被修改。但无论何时，均保证输出信号 PWM1_CH0 和 PWM1_CH0N 不会同时处于有效电平。

11.5.4.4.1 刹车源

刹车源包括三种：

- 引脚刹车事件。
- 软刹车事件。通过配置刹车事件寄存器（配置寄存器 TIM1_EGR 的 BG 位为 1）对 PWM 输出进行刹车。
- 比较器刹车，比较器的输出信号可以作为 PWM 的刹车信号。

系统复位后，刹车电路被禁止，MOE 位为低，定时器工作在空闲模式下。设置 TIM1_BRKC 寄存器的 BRKEN 位可以使能刹车功能（软刹车不受 BRKEN 位控制），刹车输入信号的极性可以通过配置寄存器 TIM1_BRKC 中的 BKP 位选择。

11.5.4.4.2 刹车过程

当刹车事件发生时，硬件做如下处理：

- 互补模式下：
 - 输出首先被置于无效状态，即使没有时钟，此功能也有效。
 - 定时器管脚输出使能无效，管脚输出高阻。
- 如果设置了寄存器 TIM1_IER 中的 BIE 位，则产生一个中断。
- 如果寄存器 TIM1_BRKC 中的 AOE 位为 1，在下一个更新事件时 MOE 位被自动置 1；否则，MOE 始终保持为低直到被软件再次置 1；

当刹车输入有效时，软件无法配置 MOE，同时，状态标志不能被清除。

11.5.4.4.3 刹车后输出信号控制

因为当外部刹车信号来到时，MOE 必须立即变为低电平，即 MOE 的下降沿可以是异步的，所以在实际信号（作用在定时器的输出端）和寄存器的控制位之间设置了一个同步电

路，当向 MOE 位写入数据时，硬件会将软件写入的值作为异步信号对待，对其做同步后再写入相应的寄存器位。因此，如果当 MOE 为 0 时写 MOE 为 1，则读出它之前必须先插入一个延时（空指令）才能读到正确的值，这是因为写入的是异步信号而读的是同步信号。

寄存器 TIM1_BRKC 的 MOE 位控制 PWM 在管脚上的输出，当 MOE 为 1 时，PWM 处于运行模式下，则当 CC0E 配置为有效时，PWM1_CH0 管脚的输出使能有效，否则，PWM1_CH0 管脚的输出使能为无效，PWM1_CH0 管脚输出高阻，PWM1_CH0N 的输出管脚控制同理。

当 MOE 为 0 位，PWM 输出处于空闲模式（刹车后的状态），则 PWM1_CH0 和 PWM1_CH0N 管脚的输出使能设置为无效，PWM1_CH0 和 PWM1_CH0N 管脚输出高阻。

Figure 57 显示了响应刹车的输出实例。

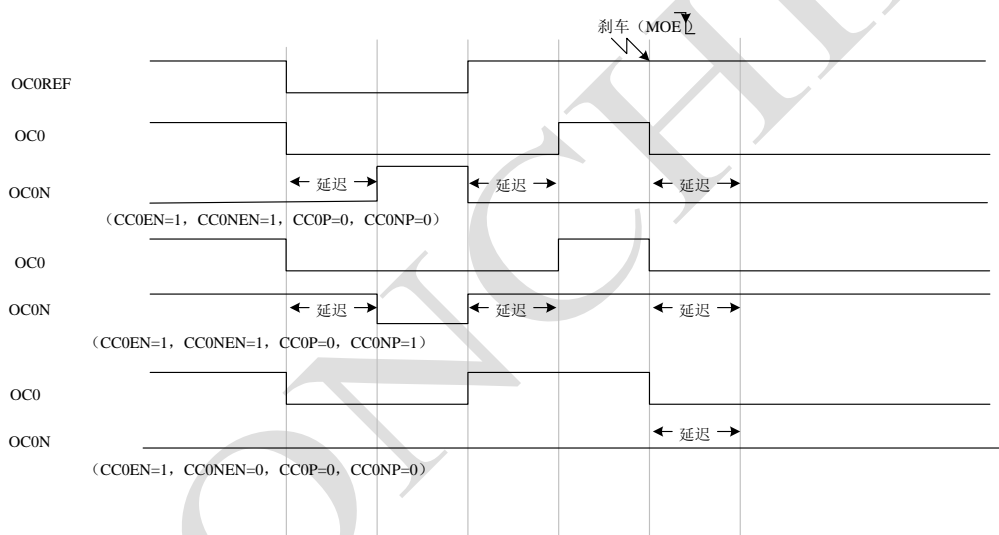


Figure 90 高级定时器刹车输出

11.5.4.5 霍尔传感器接口功能

高级定时器 TIM1 可以作为“接口定时器”来连接霍尔传感器。配置 TIM1_CONR0 寄存器的 CH0INSEL 位为 1, 选择定时器的三个输入脚(PWM1_CH0、PWM1_CH1、PWM1_CH2) 异或后的信号连接至 CC0 输入通道，高级定时器捕获这个信号。

配置高级定时器的从模式控制器为复位模式，选择 CH0INF_ED 作为从模式控制器的触发输入，每当 3 个霍尔信号中任意一个发生翻转时，计数器重新从 0 开始计数，这样产生一个由霍尔输入端的任何变化而触发的时间基准。

高级定时器的通道 CC0 配置为捕获模式，捕获信号为 CH0INF_ED，捕获值反映了两个输入变化间的时间延迟，给出了马达速度的信息。

高级定时器可以将 CC1-CC3 中的某一通道配置为输出模式，在输出模式下产生一个比

较输出波形，将此波形通过 TRGO 送出。

11.5.5 主模式控制

定时器工作在主模式下，可产生触发信号 TRGO 去控制 AD 采样模块。Figure 61 为主模式控制器示意图。

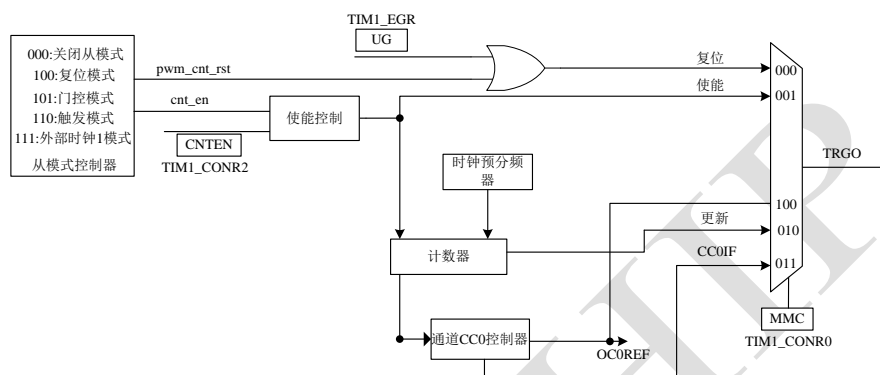


Figure 91 主模式控制示意图

主模式控制器的主要功能是选择定时器的相应信号作为触发输出 TRGO，控制过程如下：

- 配置寄存器 TIM1_CONR0 的 MMC 位为 000，选择计数器的复位信号作为触发输出，复位信号包括由从模式控制器产生的复位信号和 TIM1_EGR 寄存器的 UG 位。当 UG 为配置为 1 时，产生一个初始化信号去初始化寄存器，同时将此信号选为触发输出。
- 配置寄存器 TIM1_CONR0 的 MMC 位为 001，选择计数器的使能信号作为触发输出 TIM1_TRGO。
- 配置寄存器 TIM1_CONR0 的 MMC 位域为 010，选择通道 CC0 的比较标志 CC0IF 作为触发输出 TIM1_TRGO。若通道 CC0 配置为输入捕获功能，当发生一次捕获时，CC0IF 置位；若通道 CC0 配置为输出比较功能，当计数器的值和 TIM1_CC0R 匹配时，CC0IF 置位。在这两种情况下都可以将 CC0IF 信号作为触发输出 TIM1_TRGO。
- 配置寄存器 TIM1_CONR0 的 MMC 位为 011，选择更新事件 UEV 作为触发输出。
- 配置寄存器 TIM1_CONR0 的 MMC 位为 100，选择通道 CC0 的基准参考信号 OC0REF 作为触发输出。

定时器发出的触发 AD 采样的 CC 事件有：

- 定时器 TIM1 的触发输出 TIM1_TRGO；
- 定时器 TIM1 的通道 CC0 的主路输出 TIM1_OC0；

11.5.6 中断

11.5.6.1 中断源

高级定时器共有 9 个中断源，每个中断产生的条件如下：

- 1) 刹车中断。若已打开刹车中断使能，刹车中断产生的条件有：
 - 刹车使能有效时外部输入刹车信号。
 - 软件配置寄存器 TIM1_EGR 的 BG 位为 1，产生软刹车事件。
 - 产生比较器刹车信号。
- 2) 触发中断。若已打开触发中断使能，触发中断产生的条件有：
 - 软件配置寄存器 TIM1_EGR 的 TG 位为 1，产生触发事件。
 - 在从模式控制器的触发输入端 TRGI 检测到有效边沿。
- 3) COM 中断。若已打开 COM 中断使能，COM 中断产生的条件有：
 - 软件配置寄存器 TIM1_EGR 的 COM 位为 1，产生 COM 事件。
- 4) 通道 CC0 的捕获/比较中断。若已打开通道 CC0 的捕获/比较中断使能，通道 CC0 的捕获/比较中断产生的条件有：
 - 若通道 CC0 配置为输入捕获功能（寄存器 TIM1_CC0MR 的 CC0MS 位配置为 01/10/11），软件配置寄存器 TIM1_EGR 的 CC0G 位为 1，产生捕获事件。
 - 若通道 CC0 配置为输入捕获功能（寄存器 TIM1_CC0MR 的 CC0MS 位配置为 01/10/11），检测到通道 CC0 的捕获信号 CH0INCP 的有效边沿。
 - 若通道 CC0 配置为输出比较功能（寄存器 TIM1_CC0MR 的 CC0MS 位配置为 00），软件配置寄存器 TIM1_EGR 的 CC0G 位为 1，产生比较事件。
 - 若通道 CC0 配置为输出比较功能（寄存器 TIM1_CC0MR 的 CC0MS 位配置为 00），计数器与比较值匹配时。
- 5) 通道 CCx（x=1, 2）的捕获/比较中断。若已打开通道 CCx 的捕获/比较中断使能，通道 CCx 的捕获/比较中断产生的条件有：
 - 若通道 CCx 配置为输入捕获功能（寄存器 TIM1_CCxMR 的 CCxMS 位配置为 01/10/11），软件配置寄存器 TIM1_EGR 的 CCxG 位为 1，产生捕获事件。
 - 若通道 CCx 配置为输入捕获功能（寄存器 TIM1_CCxMR 的 CCxMS 位配置为 01/10/11），检测到通道 CCx 的捕获信号 CHxINCP 的有效边沿。
- 6) 计数器上溢更新中断。若已打开上溢更新中断使能，上溢更新中断产生的条件有：
 - 若计数器的上溢事件没有被屏蔽（配置寄存器 TIM1_CONR3 的 UPOUDIS 位为 0），当计数器上溢时产生计数器上溢更新中断。
- 7) 计数器下溢更新中断。若已打开下溢更新中断使能，下溢更新中断产生的条件有：
 - 若计数器的下溢事件没有被屏蔽（配置寄存器 TIM1_CONR3 的 DOWNOUDIS

位为 0)，当计数器下溢时产生计数器下溢更新中断。

8) 复位更新中断。若已打开复位更新中断使能，复位更新中断产生的条件有：

- 设置 TIM1_EGR 寄存器的 UG 位为 1。
- 从模式控制器产生的复位信号。

各个中断源产生后，在 TIM1 中进行中断合并得到中断 PWM1_INT，将 PWM1_INT 送入 JMT51，在 JMT51 中经过一级门控后得到 JMT51 中的 TIM 中断，Figure 62 为中断产生过程。

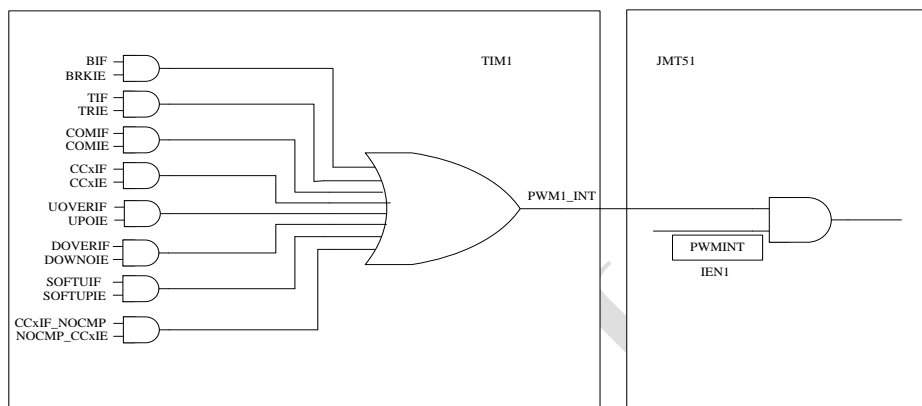


Figure 92 JMT18F003PLUS 高级定时器中断

11.5.6.2 中断配置及处理

TIM1 中断配置及处理过程如下：

- 配置中断使能寄存器 TIM1_IER 和 TIM1_CONR3 的中断使能位，使能相应的中断源。
- 配置寄存器 IEN1 的 PWMINT 位，使能高级定时器中断。
- 当产生 TIM1 中断后，软件需要回读中断标志寄存器 TIM1_ISR0、TIM1_ISR1，来判断高级定时器的中断源，根据不同中断源，向中断标志寄存器相应位写 0 清除中断，并做相应处理。

11.6 寄存器描述

PWM1 模块有 28 个寄存器，对应的 SFRPAGE 为 1，具体说明如 Table 104 所示。

Table 144 PWM1 寄存器（SFRPAGE=0x01）

地址	寄存器名	属性	复位值	功能描述
0XA1	TIM1_CONR0	W/R	0x00	TIM1 控制寄存器 0
0XA3	TIM1_CONR1	W/R	0x00	TIM1 控制寄存器 1
0XA4	TIM1_CONR2	W/R	0x20	TIM1 控制寄存器 3
0XA5	TIM1_TGICR0	W/R	0x00	从模式控制寄存器 0
0XA7	TIM1_IER	W/R	0x00	TIM1 中断使能寄存器 0

地址	寄存器名	属性	复位值	功能描述
0XA9	TIM1_CC0MR	W/R	0x00	TIM1 通道 CC0 输入捕获/输出比较模式控制寄存器
0XAA	TIM1_CC1MR	W/R	0x00	TIM1 通道 CC1 输入捕获/输出比较模式控制寄存器
0XAB	TIM1_CC2MR	W/R	0x00	TIM1 通道 CC2 输入捕获/输出比较模式控制寄存器
0XAD	TIM1_CCENR	W/R	0x00	通道使能控制寄存器
0XAE	TIM1_CCPS	W/R	0x00	通道极性控制寄存器
0XAF	TIM1_DTG	W/R	0x00	TIM1 死区时间寄存器
0XB1	TIM1_ARRL	W/R	0x00	TIM1 周期值自动装载寄存器的低 8 位
0XB2	TIM1_ARRH	W/R	0x00	TIM1 周期值自动装载寄存器的高 8 位
0XB3	TIM1_PSCL	W/R	0x00	TIM1 预分频寄存器的低 8 位
0XB4	TIM1_PSCH	W/R	0x00	TIM1 预分频寄存器的高 8 位
0XB5	TIM1_RCR	W/R	0x00	TIM1 重复计数寄存器
0XB6	TIM1_CC0RL	W/R	0x00	通道 CC0 捕获/比较寄存器的低 8 位
0XB7	TIM1_CC0RH	W/R	0x00	通道 CC0 捕获/比较寄存器的高 8 位
0XB9	TIM1_CC1RL	W/R	0x00	通道 CC1 捕获/比较寄存器的低 8 位
0XBA	TIM1_CC1RH	W/R	0x00	通道 CC1 捕获/比较寄存器的高 8 位
0XBB	TIM1_CC2RL	W/R	0x00	通道 CC2 捕获/比较寄存器的低 8 位
0XBC	TIM1_CC2RH	W/R	0x00	通道 CC2 捕获/比较寄存器的高 8 位
0XBF	TIM1_BRKC	W/R	0x00	TIM1 刹车控制寄存器
0XC1	TIM1_EGR	W	0x00	TIM1 事件产生寄存器
0XC3	TIM1_ISR0	W/R	0x00	TIM1 中断标志寄存器 0
0XC4	TIM1_ISR1	W/R	0x00	TIM1 中断标志寄存器 1
0XCA	TIM1_CNTEN	W/R	0x00	TIM1 计数器使能寄存器

11.6.1 TIM1 控制寄存器 0 (TIM1_CONR0) [0xA1]

Table 145 TIM1 控制寄存器 0 (TIM1_CONR0)

参数名	比特位	属性	复位值	描述
CLKDIV	1-0	W/R	0	时钟分频系数，定义了定时器时钟

参数名	比特位	属性	复位值	描述
				<p>(pwm0_clk) 与死区及滤波时钟之间的分频系数。时钟 dts_clk 用于死区时间发生器以及数字滤波器。</p> <p>00: $T_{dts_clk} = T_{pwm1_clk}$</p> <p>01: $T_{dts_clk} = 2 \times T_{pwm1_clk}$</p> <p>10: $T_{dts_clk} = 4 \times T_{pwm1_clk}$</p> <p>11: 保留。</p>
CHOINSEL	2	W/R	0	<p>通道 CC0 输入信号选择寄存器:</p> <p>0: PWM0_CH0 管脚上的信号连接到通道 CC0</p> <p>1: PWM0_CH0、PWM0_CH1 和 PWM0_CH2 管脚上的信号异或后连接到通道 CC0</p>
MMC	5-3	W/R	0	<p>主模式下输出信号选择寄存器, 当定时器用作主定时器时, 此寄存器用于选择送到从定时器的同步信号 (TRGO):</p> <p>000: 复位信号</p> <ul style="list-style-type: none"> 当 TIM0_EGR 寄存器的 UG 置位时, 或由从模式控制器产生复位时, 复位信号作为触发输出 (TRGO)。 <p>001: 使能信号</p> <ul style="list-style-type: none"> 计数器使能信号被用于作为触发输出 (TRGO)。 <p>若需要在同一时间启动多个定时器, 或者要实现在一定的时间窗口内使能从定时器, 那么这种模式将非常有用。</p> <p>当 CNTEN 控制位被置位, 或当从模式下输入有效信号时, 计数器使能信号将会被置位。</p> <p>当计数器使能信号受控于触发输入时, TRGO 比触发输入信号有一个延迟,</p>

参数名	比特位	属性	复位值	描述
				<p>除非选择了主/从模式，详述参看寄存器 TIM_TGICR0 的 MSM 位。</p> <p>010: 更新事件</p> <ul style="list-style-type: none"> 更新事件（上溢更新事件、下溢更新事件和软件更新事件）被选为触发输出（TRGO）。 <p>011: 通道 CC0 的捕获/比较脉冲</p> <ul style="list-style-type: none"> 一旦发生一次捕获或一次比较成功，送出一个正脉冲。 <p>其他: 通道 CC0 的基准参考信号 OC0REF</p>
CCUS	6	W/R	0	<p>捕获/比较控制位更新条件:</p> <p>0: 如果捕获/比较控制位是预装载的 (CCPE =1), 只能通过设置 COM 位更新。</p> <p>1: 如果捕获/比较控制位是预装载的 (CCPE =1), 可以通过设置 COM 位或 TRGI 的有效沿更新。</p>
CCPE	7	W/R	0	<p>捕获/比较控制位预装载使能控制寄存器:</p> <p>0: 捕获/比较控制位 CCxE, CCxNE, CCxP, CCxNP 和 OCxMS 不是预装载的。</p> <p>1: 捕获/比较控制位 CCxE, CCxNE, CCxP, CCxNP 和 OCxMS 位是预装载的, 设置该位后, 这些寄存器只在设置了 COM 位或 TRGI 的有效沿更新。</p>

11.6.2 TIM1 控制寄存器 1 (TIM1_CONR1) [0xA3]

Table 146 TIM1 控制寄存器 1 (TIM1_CONR1)

参数名	比特位	属性	复位值	描述
ARPLE	0	W/R	0	<p>计数器周期值预装载使能位:</p> <p>0: 关闭 TIM1_ARR 寄存器预装载功能。</p> <p>1: 打开 TIM1_ARR 寄存器预装载功能。</p>
CNTMC	2-1	W/R	0	<p>计数器模式控制:</p> <p>00: 边沿对齐模式, 计数器依据方向位</p>

参数名	比特位	属性	复位值	描述
				<p>CNTDIR 的配置向上或向下计数。</p> <p>01: 中央对齐模式 1, 计数器交替向上向下计数, 只在计数器向下计数时更新输出通道的输出比较中断标志位。</p> <p>10: 中央对齐模式 2, 计数器交替向上向下计数, 只在计数器向上计数时更新输出通道的输出比较中断标志位。</p> <p>11: 中央对齐模式 3, 计数器交替向上向下计数, 在计数器向上计数和向下计数时更新输出通道的输出比较中断标志位。</p> <p>注: 当计数器正在计数时, 不允许从边沿对齐模式转换到中央对齐模式。</p>
CNTDIR	3	W/R	0	<p>计数器计数方向:</p> <p>0: 计数器向上计数。</p> <p>1: 计数器向下计数。</p> <p>注: 当计数器配置为中央对齐模式或编码器模式时, 该位只读。</p>
RESERVED	4	R	0	保留。
URC	5	W/R	0	<p>更新源选择寄存器, 软件通过该位选择更新事件的源:</p> <p>0: 下述任一事件产生一个更新事件, 如果更新中断使能, 则同时产生一个更新中断:</p> <ul style="list-style-type: none"> --计数器上溢/下溢 --设置 TIM_EGR 寄存器的 UG 位 --从模式控制器产生的复位信号 <p>1: 下述任一事件产生一个更新事件, 如果更新中断使能, 只有计数器上溢/下溢时产生一个更新中断:</p> <ul style="list-style-type: none"> --计数器上溢/下溢 --设置 TIM_EGR 寄存器的 UG 位 --从模式控制器产生的复位信号。
RESERVED	7	R	0	保留。

11.6.3 TIM1 控制寄存器 2 (TIM1_CONR2) [0xA4]

Table 147 TIM1 模式选择寄存器 2(TIM1_CONR2)

参数名	比特位	属性	复位值	描述
UPOUDIS	0	W/R	0	上溢无效寄存器: 0: 当计数器上溢时产生更新事件。 1: 当计数器上溢时不产生更新事件。
DOWNOUDIS	1	W/R	0	下溢无效寄存器: 0: 当计数器下溢时产生更新事件。 1: 当计数器下溢时不产生更新事件。
RESERVED	7-2	R	0	保留

11.6.4 从模式控制寄存器 0 (TIM1_TGICR0) [0xA5]

Table 148 从模式控制寄存器 0(TIM1_TGICR0)

参数名	比特位	属性	复位值	描述
MSM	0	R	0	主从模式寄存器: 0: TIM0 和 TIM1 独立启动。 1: TIM1 处于主模式下, TIM0 和 TIM1 同时启动。
TRGS	3-1	W/R	0	从模式控制器触发信号选择寄存器: 000: 通道 CC0 的边沿检测信号 CH0INFP_ED; 001: 通道 CC0 滤波后的定时器输入 CH0INFP; 010: 通道 CC1 滤波后的定时器输入 CH1INFP; 100 : PWM0 的输出信号 PWM0_TRGO。 其他: 保留
SMS	6-4	W/R	0	从模式选择寄存器: 000: 关闭从模式 ● 通过配置计数器能寄存器 (TIM0_CNTEN.CNTEN)启动

参数名	比特位	属性	复位值	描述
				<p>计数器。</p> <p>001: 编码器模式 1</p> <ul style="list-style-type: none"> ● 根据 CH0INFP0 的电平, 计数器在 CH1INFP1 的边沿向上/向下计数。 <p>010: 编码器模式 2</p> <ul style="list-style-type: none"> ● 根据 CH1INFP1 的电平, 计数器在 CH0INFP0 的边沿向上/向下计数。 <p>011: 编码器模式 3</p> <ul style="list-style-type: none"> ● 根据 CH1INFP1 和 CH0INFP0 的电平, 计数器在 CH0INFP0 和 CH1INFP1 的边沿向上/向下计数。 <p>100: 复位模式</p> <ul style="list-style-type: none"> ● 选中的触发输入的有效沿重新初始化计数器, 并且产生一个更新事件。 <p>101: 门控模式</p> <ul style="list-style-type: none"> ● 当触发输入为有效电平时, 计数器计数。一旦触发输入变为无效电平, 计数器停止计数 (但不复位)。 <p>其他: 触发模式</p> <ul style="list-style-type: none"> ● 计数器在触发输入的有效沿启动 (但不复位), 仅控制计数器启动。 <p>注: 如果 CH1INFP1_ED 被选为触发输入时, 不要使用门控模式, 这是因为 CH1INFP1_ED 是一个脉冲信号, 而门控模式是要检查触发输入的电平。</p>
RESERVED	7	R	0	保留

11.6.5 TIM1 中断使能寄存器 0 (TIM1_IER0) [0xA7]

Table 149 TIM1 中断使能寄存器 0 (TIM1_IER0)

参数名	比特位	属性	复位值	描述
BRKIE	0	W/R	0	刹车中断使能： 0：禁止刹车中断 1：允许刹车中断
TRIE	1	W/R	0	触发中断使能： 0：禁止触发中断 1：允许触发中断
COMIE	2	W/R	0	COM 中断使能： 0：禁止 COM 中断 1：允许 COM 中断
CC0IE	3	W/R	0	互补模式下通道 CC0 捕获/比较中断使能： 0：禁止捕获/比较中断 1：允许捕获/比较中断
CC1IE	4	W/R	0	互补模式下通道 CC1 捕获中断使能： 0：禁止捕获/比较中断 1：允许捕获/比较中断
CC2IE	5	W/R	0	互补模式下通道 CC2 捕获中断使能： 0：禁止捕获/比较中断 1：允许捕获/比较中断
RESERVED	7-6	R	0	保留

11.6.6 TIM1 中断使能寄存器 1 (TIM1_IER1) [0xCB]

Table 150 TIM1 中断使能寄存器 1 (TIM1_IER1)

参数名	比特位	属性	复位值	描述
UPOIE	0	W/R	0	上溢中断使能寄存器： 0：当计数器上溢时不允许产生更新中断。 1：当计数器上溢时允许产生更新中断。
DOWMOIE	1	W/R	0	下溢中断使能寄存器： 0：当计数器下溢时不允许产生更新中断。 1：当计数器下溢时允许产生更新中断。
SOFTUPIE	2	W/R	0	复位更新中断使能寄存器：

参数名	比特位	属性	复位值	描述
				0: 不允许产生复位更新中断。 1: 当产生以下更新事件时允许产生更新中断: <ul style="list-style-type: none"> ● 设置 TIM0_EGR 寄存器的 UG 位为 1。 ● 从模式控制器产生的复位信号。
RESERVED	7-3	R	0	保留

11.6.7 TIM1 通道 CC0 输入捕获/输出比较模式控制寄存器 (TIM1_CC0MR) [0xA9]

Table 151 TIM1 通道 CC0 输入捕获/输出比较模式控制寄存器（输出比较模式）
(TIM1_CC0MR)

参数名	比特位	属性	复位值	描述
CC0MS	1-0	W/R	00	捕获/比较功能选择: 00: CC0 通道被配置为输出比较功能; 01: CC0 通道被配置为输入捕获功能, 输入捕获信号 CH0INC 来自通道 CC0 的输入信号 CH0IN。 10: CC0 通道被配置为输入捕获功能, 输入捕获信号 CH0INC 来自通道 CC1 的输入信号 CH1IN。 11: CC0 通道被配置为输入捕获功能, 输入捕获信号 CH0INC 来自 TRC。 注: CC0MS 仅在通道关闭 (寄存器 TIM1_CCENR 的 CC0EN=0) 时才是可写的, 因此在使用通道 CC0 时, 应该先配置 CC0MS, 再配置 CC0EN 为 1。
RESERVED	2	R	0	保留
OC0PEN	3	W/R	0	通道 CC0 的输出比较寄存器的预装载使能: 0: 禁止寄存器 TIM1_CC0R 的预装载功能, 可随时将数据写入寄存器 TIM1_CC0R, 且新值立即生效。

参数名	比特位	属性	复位值	描述
				<p>1: 开启寄存器 TIM1_CC0R 的预装载功能，读写操作仅对预装载寄存器操作，TIM1_CC0R 的预装载值在更新事件产生时被载入影子寄存器。</p> <p>注 2: 若未使能输出比较寄存器的预装载功能，当改变输出比较寄存器的值时，PWM 模式下可能会出现不规则的输出。单脉冲模式下的 PWM 输出不会出现这种情况。</p>
OC0MS	6-4	W/R		<p>输出模式选择寄存器，该位定义了输出参考信号 OC0REF 的动作，而 OC0REF 决定了 OC0、OC0N 的输出，从而决定输出管脚上的 PWM 输出波形。OC0REF 为高电平有效，而 OC0 和 OC0N 的有效电平取决于 CC0P 和 CC0NP 位的配置。</p> <p>000: 冻结</p> <ul style="list-style-type: none"> ● 输出比较寄存器 TIM1_CC0RU 与计数器间的比较结果对 OC0REF 不起作用。 <p>001: 匹配时输出有效电平</p> <ul style="list-style-type: none"> ● 当输出比较寄存器 TIM1_CC0R 与计数器的值相同时，OC0REF 输出有效电平（即高电平）。 <p>010: 匹配时输出无效电平</p> <ul style="list-style-type: none"> ● 当输出比较寄存器 TIM1_CC0R 与计数器的值相同时，OC0REF 输出无效电平（即低电平）。 <p>011: 翻转</p> <ul style="list-style-type: none"> ● 当输出比较寄存器 TIM1_CC0R 与计数器的值相同时，OC0REF 信号翻转。 <p>100: 强制输出无效电平</p>

参数名	比特位	属性	复位值	描述
				<ul style="list-style-type: none"> 当配置此种输出模式时，立即将 OC0REF 设置为无效电平。 <p>101: 强制输出有效电平</p> <ul style="list-style-type: none"> 当配置此种输出模式时，立即将 OC0REF 设置为有效电平。 <p>110: PWM 模式 1</p> <ul style="list-style-type: none"> 边沿计数模式下：向上计数时，若 $TIM1_CNT < TIM1_CC0R$，OC0REF 为有效电平（OC0REF=1），否则为无效电平（OC0REF=0）；向下计数时，若 $TIM1_CNT > TIM1_CC0R$，OC0REF 为无效电平（OC0REF=0），否则为有效电平（OC0REF=1）。 中央对齐计数模式下：向上计数时，若 $TIM1_CNT < TIM1_CC0R$，OC0REF 为有效电平（OC0REF=1），否则为无效电平（OC0REF=0）；向下计数时，若 $TIM1_CNT > TIM1_CC0R$，OC0REF 为无效电平（OC0REF=0），否则为有效电平（OC0REF=1）。 <p>111: PWM 模式 2</p> <ul style="list-style-type: none"> 边沿计数模式下：向上计数时，若 $TIM1_CNT < TIM1_CC0R$，OC0REF 为无效电平（OC0REF=0），否则为有效电平（OC0REF=1）；向下计数时，若 $TIM1_CNT > TIM1_CC0R$，OC0REF 为有效电平（OC0REF=1），否则为无效电平（OC0REF=0）。 中央对齐计数模式下：向上计数时，若 $TIM1_CNT < TIM1_CC0R$，OC0REF 为无效电平（OC0REF=0），

参数名	比特位	属性	复位值	描述
				否则为有效电平（OC0REF=1）；向下计数时，若 TIM1_CNT > TIM1_CC0R，OC0REF 为有效电平（OC0REF=1），否则为无效电平（OC0REF=0）。
RESERVED	7	R	0	保留

PWM 通道 CC0 输入捕获/输出比较模式控制寄存器（输入捕获模式）（TIM1_CC0MR）

参数名	比特位	属性	复位值	描述
CC0MS	1-0	W/R	00	捕获/比较功能选择寄存器： 00：CC0 通道被配置为输出比较功能； 01：CC0 通道被配置为输入捕获功能，输入捕获信号 CH0INC 来自通道 CC0 的输入信号 CH0IN。 10：CC0 通道被配置为输入捕获功能，输入捕获信号 CH0INC 来自通道 CC1 的输入信号 CH1IN。 11：CC0 通道被配置为输入捕获功能，输入捕获信号 CH0INC 来自 TRC。 注：CC0MS 仅在通道关闭（寄存器 TIMx_CCENR 的 CC0EN=0）时才是可写的，因此在使用通道 CC0 时，应该先配置 CC0MS，再配置 CC0EN 为 1。
IC0PDIV	3-2	W/R	0	通道 CC0 输入捕获信号预分频控制寄存器，这两位定义了 CC0 输入信号的预分频系数。 00：不分频。 01：每 2 个事件触发一次捕获。 10：每 4 个事件触发一次捕获。 11：每 8 个事件触发一次捕获。
RESERVED	7-4	R	0	保留

11.6.8 TIM1 通道 CC1 输入捕获控制寄存器 (TIM1_CC1MR) [0xAA]

Table 152 TIM1 通道 CC1 输入捕获模式控制寄存器（输入捕获模式）（TIM1_CC1MR）

参数名	比特位	属性	复位值	描述
CC1MS	1-0	W/R	00	捕获/比较功能选择寄存器： 00: CC1 通道被配置为输出比较功能； 01: CC1 通道被配置为输入捕获功能，输入捕获信号 CH0INC 来自通道 CC1 的输入信号 CH1IN。 10: CC1 通道被配置为输入捕获功能，输入捕获信号 CH1INC 来自通道 CC0 的输入信号 CH0IN。 11: CC1 通道被配置为输入捕获功能，输入捕获信号 CH1INC 来自 TRC。 注：CC1MS 仅在通道关闭（寄存器 TIMx_CCENR 的 CC1E=0）时才是可写的，因此在使用通道 CC1 时，应该先配置 CC1MS，再配置 CC1E 为 1。
IC1PDIV	3-2	W/R	0	通道 CC1 输入捕获信号预分频控制寄存器，这两位定义了 CC1 输入信号的预分频系数。 00: 不分频。 01: 每 2 个事件触发一次捕获。 10: 每 4 个事件触发一次捕获。 11: 每 8 个事件触发一次捕获。
RESERVED	7-4	R	0	保留

11.6.9 TIM1 通道 CC2 输入捕获控制寄存器 (TIM1_CC2MR) [0xAB]

Table 153 TIM1 通道 CC2 输入捕获模式控制寄存器（输入捕获模式）(TIM1_CC2MR)

参数名	比特位	属性	复位值	描述
RESERVED	1-0	R	0	保留
IC2PDIV	3-2	W/R	0	通道 CC2 输入捕获信号预分频控制寄存器，这两位定义了 CC2 输入信号的预分频系数。 00: 不分频。 01: 每 2 个事件触发一次捕获。 10: 每 4 个事件触发一次捕获。

参数名	比特位	属性	复位值	描述
				11: 每 8 个事件触发一次捕获。
RESERVED	7-4	R	0	保留

11.6.10 通道使能控制寄存器 (TIM1_CCENR) [0xAD]

Table 154 通道使能控制寄存器(TIM1_CCENR)

参数名	比特位	属性	复位值	描述
CC0EN	0	W/R	0	<p>通道 CC0 输入捕获/输出比较使能:</p> <p>CC0 通道配置为输出时:</p> <p>0: 关闭</p> <ul style="list-style-type: none"> OC0 禁止输出, 输出引脚 PWM1_CH0 的信号受寄存器 MOE、IOSS、ROSS、OIS0、OIS0N、CC0EN、CC0P 位的控制。 <p>1: 开启</p> <ul style="list-style-type: none"> OC0 信号输出到对应的输出引脚 PWM1_CH0, 其输出受寄存器 MOE、OSSI、OSSR、OIS0、OIS0N、CC0EN、CC0P 的控制。 <p>CC0 通道配置为输入时:</p> <p>0: 捕获禁止。</p> <p>1: 捕获使能</p> <ul style="list-style-type: none"> 当捕获信号产生有效沿时, 将计数器的值捕获到 TIM1_CC0R 中。
CC0NEN	1	W/R	0	<p>通道 CC0 互补路输出使能控制:</p> <p>0: 关闭</p> <ul style="list-style-type: none"> OC0N 禁止输出, 输出引脚 PWM1_CH0N 的信号受寄存器 MOE、IOSS、ROSS、OIS0、OIS0N、CC0EN、CC0NP 位的控制。 <p>1: 开启</p> <ul style="list-style-type: none"> OC0N 信号输出到对应的输出引脚 PWM1_CH0N, 其输出受寄存器 MOE、

参数名	比特位	属性	复位值	描述
				OSSI、OSSR、OIS0、OIS0N、CC0EN、CC0NP 的控制。
CC1EN	2	W/R	0	通道 CC1 输入捕获/输出比较使能控制，参见 CC0EN 描述。
CC2EN	3	W/R	0	通道 CC2 输入捕获/输出比较使能控制，参见 CC0EN 描述。
RESERVED	7-4	R	0	保留

11.6.11 通道极性控制寄存器 (TIM1_CCPS) [0xAE]

Table 155 通道极性控制寄存器(TIM1_CCPS)

参数名	比特位	属性	复位值	描述
CC0P	0	W/R	0	<p>通道 CC0 的极性：</p> <ul style="list-style-type: none"> 通道 CC0 配置为输出时： <p>0：OC0（PWM1_CH0）的高电平为有效电平。</p> <p>1：OC0（PWM1_CH0）的低电平为有效电平。</p> CC0 通道配置为输入时： <p>该位选择是 CH0IN 还是 CH0IN 的反相信号作为触发或捕获信号。</p> <p>0：不反相：CH0IN 的上升沿作为有效边沿，CH0IN 的高电平作为有效电平。</p> <p>1：反相：CH0IN 的下升沿作为有效边沿，CH0IN 的低电平作为有效电平。</p> <p>注：一旦 LOCK 级别设置为 2 或 3，该位不能被修改。</p>
CC0NP	1	W/R	0	<p>通道 CC0 互补输出极性：</p> <p>0：OC0N（PWM1_CH0N）的高电平为有效电平。</p> <p>1：OC0N（PWM1_CH0N）的低电平为有效电平。</p> <p>注：一旦 LOCK 级别设置为 2 或 3 或</p>

参数名	比特位	属性	复位值	描述
				CC3MS =0 时，则该位不能被修改。
CC1P	2	W/R	0	通道 CC1 极性，参见 CC0P 描述。
CC2P	3	W/R	0	通道 CC2 极性，参见 CC0P 描述。
RESERVED	7-4	R	0	保留

11.6.12TIM1 死区时间寄存器 (TIM1_DTG) [0xAF]

Table 156 TIM1 死区时间寄存器(TIM1_DTG)

参数名	比特位	属性	复位值	描述
DTG	7-0	W/R	0	死区发生器死区时间设置寄存器,这些位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间: 若 $DTG[7:5] = "0xx"$: $DT = DTG[6:0] \times T_{dts_clk}$ 若 $DTG[7:5] = "10x"$: $DT = (64 + DTG[5:0]) \times 2 \times T_{dts_clk}$ 若 $DTG[7:5] = "110"$: $DT = (32 + DTG[4:0]) \times 8 \times T_{dts_clk}$ 若 $DTG[7:5] = "111"$: $DT = (32 + DTG[4:0]) \times 16 \times T_{dts_clk}$

11.6.13TIM1 周期值自动装载寄存器的低 8 位 (TIM1_ARRL) [0xB1]

Table 157 TIM1 周期值自动装载寄存器的低 8 位 (TIM1_ARRL)

参数名	比特位	属性	复位值	描述
ARRL	7-0	W/R	0	周期值自动重载值的低 8 位。若寄存器 TIM1_CONR2 的 ARPLE 位配置为 1，寄存器的预装载功能有效，当产生更新事件时，该寄存器的值被装载到影子寄存器，更新事件包括： <ul style="list-style-type: none"> 计数器上溢/下溢 配置寄存器 TIM0_EGR 的 UG 位为 1 从模式控制器产生的复位信号 当自动重载的值为 0 时，计数器不工作。

11.6.14TIM1 周期值自动装载寄存器的高 8 位 (TIM1_ARRH) [0xB2]

Table 158 TIM1 周期值自动装载寄存器的高 8 位 (TIM1_ARRH)

参数名	比特位	属性	复位值	描述
-----	-----	----	-----	----

参数名	比特位	属性	复位值	描述
ARRH	7-0	W/R	0	<p>周期值自动重装载值的高 8 位。若寄存器 TIM0_CONR2 的 ARPLE 配置为 1，寄存器的预装载功能有效，当产生更新事件时，该寄存器的值被装载到影子寄存器，更新事件包括：</p> <ul style="list-style-type: none"> ● 计数器上溢/下溢 ● 配置寄存器 TIM0_EGR 的 UG 位为 1 ● 从模式控制器产生的复位信号 <p>当自动重装载的值为 0 时，计数器不工作。</p>

11.6.15 TIM1 预分频寄存器的低 8 位 (TIM1_PSCL) [0xB3]

Table 159 TIM1 预分频寄存器的低 8 位 (TIM1_PSCL)

参数名	比特位	属性	复位值	描述
CNTPSCL	7-0	W/R	0	<p>预分频寄存器的低 8 位，该寄存器包含了当更新事件产生时装入预分频影子寄存器的值，该寄存器的预装载功能始终有效，更新事件包括：</p> <ul style="list-style-type: none"> ● 计数器上溢/下溢 ● 设置 UG 位为 1 ● 从模式控制器产生的复位信号

11.6.16 TIM1 预分频寄存器的高 8 位 (TIM1_PSCH) [0xB4]

Table 160 TIM1 预分频寄存器的高 8 位 (TIM1_PSCH)

参数名	比特位	属性	复位值	描述
CNTPSCH	7-0	W/R	0	<p>预分频器值的高 8 位，该寄存器包含了当更新事件产生时装入预分频影子寄存器的值，该寄存器的预装载功能始终有效，更新事件包括：</p> <ul style="list-style-type: none"> ● 计数器上溢/下溢 ● 设置 UG 位为 1 ● 从模式控制器产生的复位信号

11.6.17TIM1 重复计数寄存器 (TIM1_RCR) [0xB5]

Table 161 TIM1 重复计数寄存器(TIM1_RCR)

参数名	比特位	属性	复位值	描述
RCR	7-0	W/R	0	<p>周期计数次数的值，开启了预装载功能后，这些位用于设置比较寄存器的更新速率，如果允许产生更新中断，则会同时影响产生更新中断的速率。</p> <p>每次向下计数器 REP_CNT 达到 0，会产生一个更新事件并且计数器 REP_CNT 重新从 RCR 开始计数。由于 REP_CNT 只有在周期更新事件发生时才重载重复值，因此对此寄存器写入的新值只在下次更新事件发生时才起作用。</p> <p>这意味着在 PWM 模式下，RCR +1 对应着：</p> <ul style="list-style-type: none"> --在边沿对齐模式下，PWM 周期的数目。 --在中央对齐模式下，PWM 半周期的数目。

11.6.18通道 CC0 捕获/比较寄存器的低 8 位 (TIM1_CC0RL) [0xB6]

Table 162 通道 CC0 捕获/比较寄存器的低 8 位(TIM1_CC0RL)

参数名	比特位	属性	复位值	描述
CC0RL	7-0	W/R	0	<p>通道 CC0 捕获/向上计数比较寄存器的低 8 位：</p> <ul style="list-style-type: none"> ● 若 CC0 通道配置为输出： <p>此寄存器包含了装入比较影子寄存器的值（预装载值）。</p> <p>如果未选择预装载功能，其立即被装入捕获/比较影子寄存器。否则，只有当更新事件发生时，此预装载值才被装载入捕获/比较影子寄存器。影子寄存器包含了与计数器比较的值，根据比较结果在 OC0 端口上输出信号。</p> ● 若 CC0 通道配置为输入： <p>此寄存器包含了由上一次输入捕获</p>

参数名	比特位	属性	复位值	描述
				事件捕获到的计数器值的低 8 位。

11.6.19 通道 CC0 捕获/比较寄存器的高 8 位 (TIM1_CC0RH) [0xB7]

Table 163 通道 CC0 捕获/比较寄存器的高 8 位(TIM1_CC0RH)

参数名	比特位	属性	复位值	描述
CC0RH	7-0	W/R	0	<p>通道 CC0 捕获/向上计数比较寄存器的高 8 位：</p> <ul style="list-style-type: none"> 若 CC0 通道配置为输出： 此寄存器包含了装入比较影子寄存器的值（预装载值）。 如果未选择预装载功能，其立即被装入捕获/比较影子寄存器中。否则，只有当更新事件发生时，此预装载值才被装入捕获/比较影子寄存器中。影子寄存器包含了与计数器比较的值，根据比较结果在 OC0 端口上输出信号。 若 CC0 通道配置为输入： 此寄存器包含了由上一次输入捕获事件捕获到的计数器值的高 8 位。

11.6.20 通道 CC1 捕获寄存器的低 8 位 (TIM1_CC1RL) [0xB9]

Table 164 通道 CC1 捕获寄存器的低 8 位(TIM1_CC1RL)

参数名	比特位	属性	复位值	描述
CC1RL	7-0	W/R	0	通道 CC1 捕获寄存器/向上计数比较寄存器的低 8 位，此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。

11.6.21 通道 CC1 捕获寄存器的高 8 位 (TIM1_CC1RH) [0xBA]

Table 165 通道 CC1 捕获寄存器的高 8 位(TIM1_CC1RH)

参数名	比特位	属性	复位值	描述
CC1RH	7-0	W/R	0	通道 CC1 捕获寄存器/向上计数比较寄存器的高 8 位，此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。

11.6.22通道 CC2 捕获/比较寄存器的低 8 位 (TIM1_CC2RL) [0xBB]

Table 166 通道 CC2 捕获/比较寄存器的低 8 位(TIM1_CC2RL)

参数名	比特位	属性	复位值	描述
CC2RL	7-0	W/R	0	通道 CC2 捕获寄存器/向上计数比较寄存器的低 8 位, 此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。

11.6.23通道 CC2 捕获/比较寄存器的高 8 位 (TIM1_CC2RH) [0xBC]

Table 167 通道 CC2 捕获/比较寄存器的高 8 位(TIM1_CC2RH)

参数名	比特位	属性	复位值	描述
CC2RH	7-0	W/R	0	通道 CC2 捕获寄存器/向上计数比较寄存器的高 8 位, 此寄存器包含了由上一次输入捕获事件捕获到的计数器值的低 8 位。

11.6.24TIM1 刹车控制寄存器 (TIM1_BRKC) [0xBF]

Table 168 TIM1 刹车控制寄存器(TIM1_BRKC)

参数名	比特位	属性	复位值	描述
BRKZEN	0	W/R	0	刹车时输出高阻使能寄存器: 0: 刹车时不输出高阻; 1: 刹车时输出高阻。
RESERVED	2-1	R	0	保留
CMPBRKE	3	W/R	0	比较器刹车功能使能寄存器: 0: 禁止刹车输入 1: 开启刹车输入 注 1: 一旦 LOCK 级别设置为 1、2 或 3, 该位不能修改。 注 2: 此位只控制外部输入的刹车信号, 对寄存器配置的刹车事件 BG 无效。 注 3: 通过配置 16.3 节的寄存器 BKPRSEL 可以改变比较器刹车信号的极性。
EXTBRKE	4	W/R	0	外部刹车功能使能寄存器: 0: 禁止刹车输入 1: 开启刹车输入 注 1: 一旦 LOCK 级别设置为 1、2 或 3,

参数名	比特位	属性	复位值	描述
				该位不能修改。 注 2: 此位只控制外部输入的刹车信号, 对寄存器配置的刹车事件 BG 无效。
BRKP	5	W/R	0	刹车输入极性: 0: 刹车输入高电平为有效电平 1: 刹车输入低电平为有效电平 注: 一旦 LOCK 级别设置为 1、2 或 3, 该位不能修改。
AOE	6	W/R	0	自动输出使能: 0: MOE 只能被软件置 1; 1: MOE 能被软件置 1 或在刹车信号无效后的更新事件自动置 1。 注: 一旦 LOCK 级别设置为 1、2 或 3, 该位不能修改。
MOE	7	W/R	0	主输出使能寄存器, 一旦刹车输入有效, 该位被硬件异步清 0, 根据 AOE 的设置, 可由软件置 1 或自动置 1, 它仅对配置为输出通道有效。 0: 使定时器处于空闲状态。定时器的输出受 IOSS、OIS0、OIS0N、CC0EN、CC0NEN、CC0P 和 CC0NP 位的控制。 1: 使定时器处于运行状态。定时器的输出受 ROSS、CC0EN、CC0NEN、CC0P 和 CC0NP 位的控制。

11.6.25TIM1 事件产生寄存器 (TIM1_EGR) [0xC1]

Table 169 TIM1 事件产生寄存器(TIM1_EGR)

参数名	比特位	属性	复位值	描述
BG	0	W	0	产生刹车事件, 该位由软件置 1, 硬件自动清 0: 0: 无动作; 1: 产生一个刹车事件。此时 MOE=0, BIF=1, 若打开刹车中断使能, 则产生刹车中断。

参数名	比特位	属性	复位值	描述
TG	1	W	0	产生触发事件，该位由软件置 1，硬件自动清 0： 0：无动作； 1：产生一个触发事件。此时 TIF=1，若打开了触发中断使能，则产生触发中断。
COMG	2	W	0	产生寄存器 TIM1_CCENR、TIM1_CCPS 和 TIM1_CCxMR.OCxMS 的更新事件： 0：无动作； 1：当 CCxPE=1 时，更新 TIM1_CCENR、TIM0_CCPS 和 TIM1_CCxMR.OCxMS。
CC0G	3	W	0	产生捕获/比较事件，该位由软件置 1，用于产生一个捕获/比较事件，由硬件自动清 0： 0：无动作； 1：在通道 CC0 上产生一个捕获/比较事件 若通道 CC0 配置位输出： 设置 CC0IF=1，若打开比较中断使能，则产生比较中断。 若通道 CC0 配置为输入： 当前的计数器值捕获至 TIM1_CC0R 寄存器，捕获中断标志位 CC0IF 自动置 1，若打开捕获中断使能，则产生捕获中断，若 CC0IF 已经为 1，则过捕获标志位 CC0OIF 自动置 1。
CC1G	4	W	0	参见 CC0G 描述。
CC2G	5	W	0	参见 CC0G 描述。
RESERVED	6	R	0	保留
UG	7	W	0	产生更新事件，该位由软件置 1，由硬件自动清 0： 0：无动作； 1：初始化计数器、初始化预分频计数器、产生一个更新事件： ● 若在中央对齐模式或向上计数模式下，

参数名	比特位	属性	复位值	描述
				计数器被清 0； <ul style="list-style-type: none"> ● 若在向下计数模式下,计数器初始化为 TIM1_ARR 的值； ● 初始化预分频计数器为 0。

11.6.26TIM1 中断标志寄存器 1 (TIM1_ISR0) [0xC3]

Table 170 TIM1 中断标志寄存器 0(TIM1_ISR0)

参数名	比特位	属性	复位值	描述
BIF	0	W/R	0	刹车中断标志。一旦发生刹车事件,由硬件将该位置 1。刹车输入无效时,该位可由软件清 0。 0: 无刹车事件产生; 1: 产生刹车事件。
TIF	1	W/R	0	触发中断标志,当发生触发事件时(当在 TRGI 输入端检测到有效边沿或软件配置寄存器 TIM0_EGR 的 TG 位)由硬件将该位置 1,由软件清 0: 0: 无触发事件产生 1: 有触发事件产生
COMIF	2	W/R	0	COM 中断标志,一旦 COM 事件来到,该位由硬件置 1,由软件清 0: 0: 无 COM 事件产生 1: 有 COM 事件产生
CC0IF	3	W/R	0	互补模式下,通道 CC0 的捕获/比较中断标志: <ul style="list-style-type: none"> ● 如果通道 CC0 配置为输出模式: 当计数器值与比较值匹配时由硬件置 1,由软件清 0 0: 无匹配发生 1: 计数器与比较值匹配 ● 如果通道 CC0 配置为输入捕获模式:

参数名	比特位	属性	复位值	描述
				当捕获事件发生时该位由硬件置 1，由软件清 0 或通过读 TIM1_CC0R 清 0 0：无输入捕获产生 1：输入捕获产生
CC1IF	4	W/R	0	通道 CC1 捕获/比较中断标志寄存器，参见 CC0IF 描述
CC2IF	5	W/R	0	通道 CC2 捕获/比较中断标志寄存器，参见 CC0IF 描述
RESERVED	7-6	R	0	保留。

11.6.27TIM1 中断标志寄存器 2 (TIM1_ISR1) [0xC4]

Table 171 TIM1 中断标志寄存器 1(TIM1_ISR1)

参数名	比特位	属性	复位值	描述
RESERVED	3-0	R	0	保留。
UOVERIF	4	W/R	0	计数器上溢更新中断标志： 0：没有上溢事件发生； 1：发生上溢事件。
DOVERIF	5	W	0	计数器下溢更新中断标志： 0：没有下溢事件发生； 1：发生下溢事件。
SOFTUIF	6	W/R	0	复位（配置 TIM0_EGR 的 UG 位或复位模式下的从模式控制器产生的复位信号）更新中断标志： 0：没有软件或复位更新事件发生； 1：发生软件或复位更新事件。
RESERVED	7	R	0	保留。

11.6.28TIM1 计数器使能寄存器(TIM1_CNTEN) [0xCA]

Table 172 TIM1 计数器使能寄存器 (TIM1_CNTEN)

参数名	比特位	属性	复位值	描述
CNTEN	0	W/R	0	计数器使能软件控制位： 0：计数器不计数。 1：计数器计数。

参数名	比特位	属性	复位值	描述
				注：触发模式下不需要配置，硬件自动设置 CNTEN 位。
RESERVED	7-1	R	0	保留。

IRONCHIP

IRONCHIP

IRONCHIP

12 通用异步收发传输器 (UART)

12.1 概述

JMT18F003PLUS 的通用异步收发传输器 (UART) 支持三种接口:

- 标准 UART 通信接口
- 38KHz 红外调制发送接口
- IrDA(SIR)红外通信接口
- LIN 模式

12.2 框图

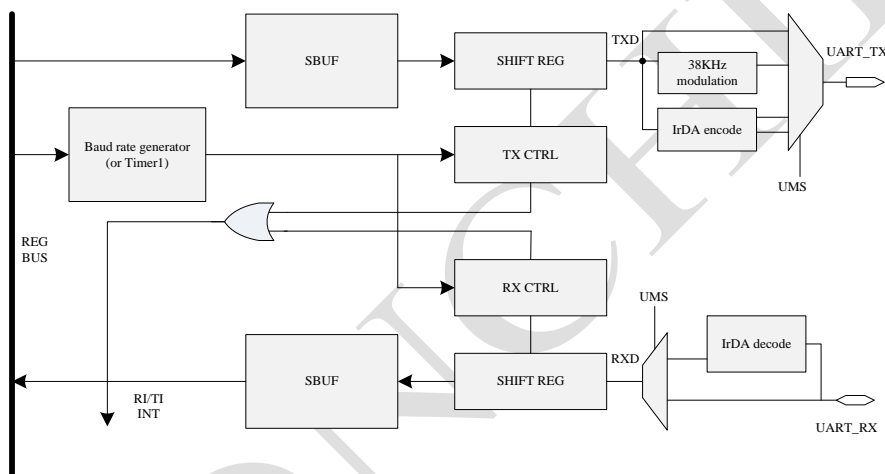


Figure 93 UART 结构框图

UART 通信接口由两个数据缓冲器、一个移位寄存器、一个串行控制寄存器、一个波特率发生器、一个 38KHz 调制和一个 IrDA 编解码器等组成。

标准 UART 通信接口共有 4 种工作模式，以供不同应用场景选用：

- 模式 0：同步移位寄存器。通过 UART_TX 发送同步时钟，通过 UART_RX 接收或发送数据，波特率是处理器频率 $f_{\text{sys_clk}}$ 的 1/12。
- 模式 1：8 位 UART。通过 UART_TX 发送或通过 UART_RX 接收 8 个数据位，波特率是可变的。
- 模式 2：9 位 UART。通过 UART_TX 发送或通过 UART_RX 接收 9 个数据位，波特率可编程为处理器频率 $f_{\text{sys_clk}}$ 的 1/32 或 1/64。
- 模式 3：9 位 UART。通过 UART_TX 发送或通过 UART_RX 接收 9 个数据位，波特率是可变的。

38KHz 红外调制发送接口，可通过标准 UART 通信接口的模式 1 和 3 实现，具体参加 12.4.5。

IrDA(SIR)红外通信接口，可通过标准 UART 通信接口的模式 1、2 和 3 实现，支持 3/16 以及 low-power(1.41-2.23us)脉宽格式，具体参见 12.4.6。

LIN 模式包括主机发送断开符及从机检测断开符功能，具体参见 12.4.7。

12.3 UART 管脚配置

UART 的管脚配置可参见 Table 1 引脚说明表格，管脚配置说明见 GPIO 寄存器说明。

12.4 UART 工作模式

12.4.1 标准 UART 工作模式 0：同步移位寄存器

配置 SCON 的 SM0、SM1 为 00 时，UART 工作于模式 0。此模式为同步移位寄存器，波特率固定为系统时钟频率 $f_{\text{sys_clk}}$ 的 1/12。同步时钟由 UART_TX 输出，数据由 UART_RX 端输入或输出，发送、接收的是 8 位数据，低位在先。接口时序如 Figure 94 和 Figure 95 所示。图中 baud_clk 表示波特率示意时钟，buf_write 为软件写 SBUF 动作，SCON.TI 为发送中断标志，SCON.RI 为接收中断标志，此后类同。

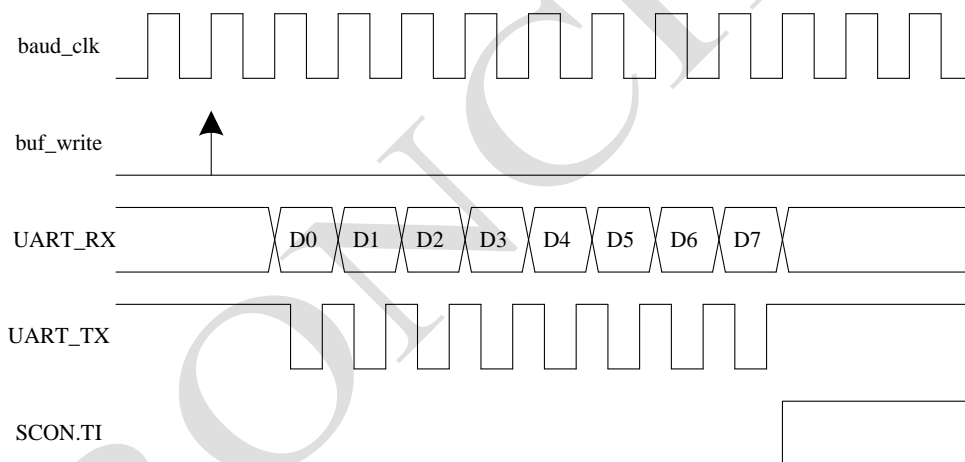


Figure 94 UART 模式 0 发送时序

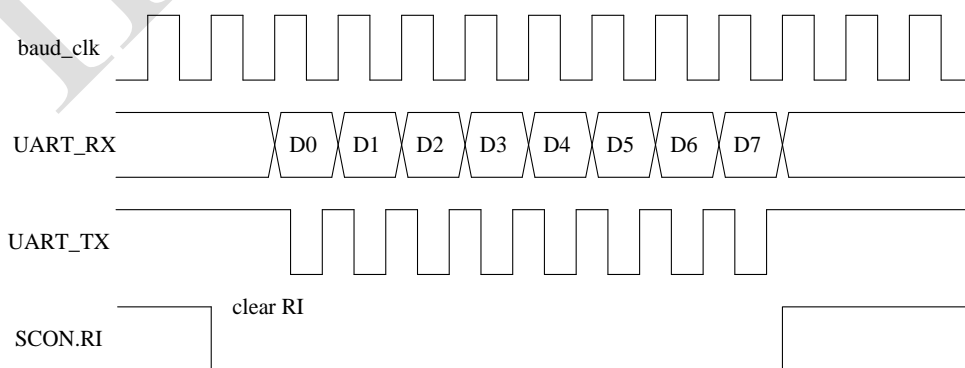


Figure 95 UART 模式 0 接收时序

模式 0 发送：当处理器执行将数据写入发送缓冲器 SBUF 时启动发送，硬件将 8 位数据

以 $f_{\text{sys_clk}}/12$ 的波特率从 UART_RX 管脚输出。一帧(8 位)数据发送完毕时, SCON.TI 置 1, 申请中断。再次发送数据前, 必须用软件将 SCON.TI 清 0。

模式0接收: 首先将接收中断请求标志 SCON.RI 置 1, 然后将接收中断请求标志 SCON.RI 清 0, 且将允许接收控制位 SCON.REN 置 1, 启动串行模式 0 接收过程。启动接收过程后, UART_RX 为串行输入端, UART_TX 为同步时钟。串行接收的波特率为 $f_{\text{sys_clk}}/12$ 。当接收完成一帧数据(8 位)后, 硬件将会把 SCON.RI 置 1, 并发出中断申请。当再次接收时, 必须通过软件将 SCON.RI 清 0。

工作于模式 0 时, 必须将多机通信控制位 SCON.SM2 清 0, 使得对 SCON.TB8 位和 SCON.RB8 位没有影响。

模式 0 发送中断和模式 0 接收中断合并后送至 CPU, 请求中断, 软件响应中断后必须判别是 SCON.TI 中断请求还是 SCON.RI 中断请求, 并清除相应的标志位。

12.4.2 标准 UART 工作模式 1: 8 位 UART, 波特率可变

配置寄存器 SCON 的 SM0、SM1 为 01 时, UART 工作于模式 1。此模式为 8 位 UART 工作模式, 一帧数据为 10 位: 1 位起始位, 8 位数据位(低位在先)和 1 位停止位。

该模式下, 波特率可变, 即可根据需要进行设置。

UART_TX 为发送信息, UART_RX 为接收信息, 以全双工模式进行接收/发送。接口时序如 Figure 96 和 Figure 97 所示。

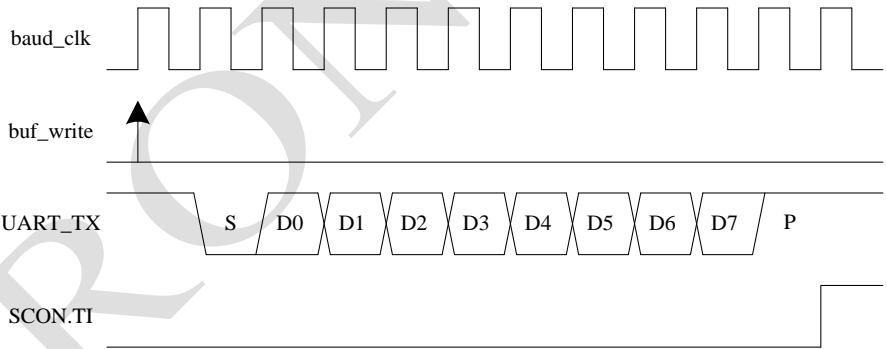


Figure 96 UART 模式 1 发送时序

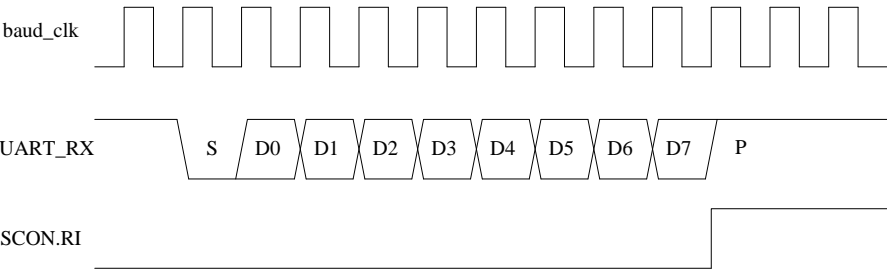


Figure 97 UART 模式 1 接收时序

UART 模式 1 中,波特率可以由内部的波特率产生器或 Timer 1 产生,具体参见 12.7 节。

在模式 1 中,可以用中断方式进行数据通信。当发送完一帧数据后,硬件自动将发送中断标志 `SCON.TI` 置 1;当接收完一帧数据后,硬件自动将接收中断标志 `SCON.RI` 置 1。在响应中断后,必须由软件清 0。

模式 1 发送中断和模式 1 接收中断合并后送至 CPU,请求中断,软件响应中断后必须判别是 `SCON.TI` 中断请求还是 `SCON.RI` 中断请求,并清除相应的标志位。

12.4.3 标准 UART 工作模式 2: 9 位 UART, 波特率固定

配置寄存器 `SCON` 的 `SM0`、`SM1` 为 10 时, UART 工作于模式 2。此模式为 9 位 UART 工作模式,一帧信息为 11 位: 1 位起始位, 8 位数据位(低位在先), 1 位可编程位(`SCON.TB8/SCON.RB8`, 第 9 位数据)和 1 位停止位。`SCON.TB8/SCON.RB8` 可以作为多机通信地址标志或奇偶校验位使用。`UART_TX` 为发送信息, `UART_RX` 为接收信息, 以全双工模式进行接收/发送。接口时序如 Figure 98 和 Figure 99 所示。

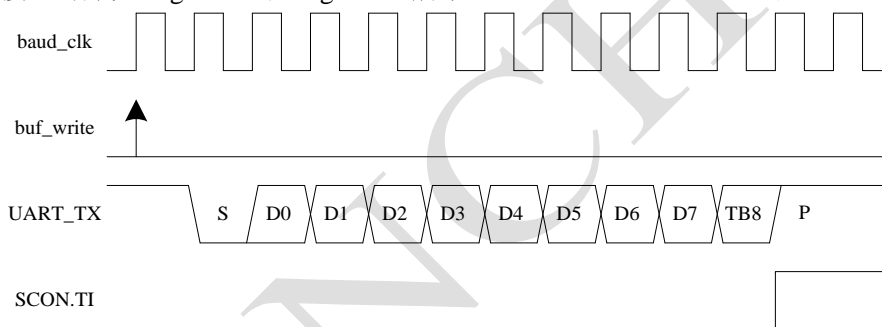


Figure 98 UART 模式 2 发送时序

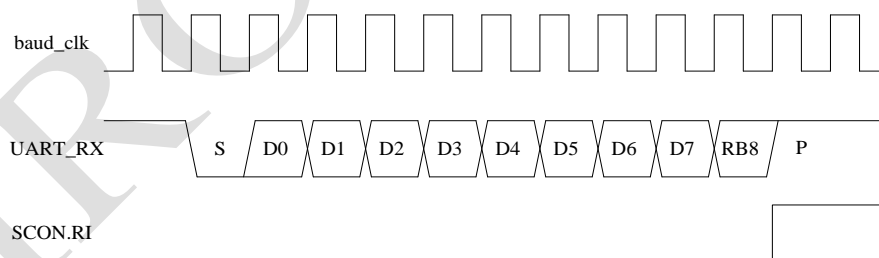


Figure 99 UART 模式 2 接收时序

配置寄存器 `SCON` 中的 `SM2`、`TB8` 可实现多机通信和奇偶校验,具体见 12.4.7.1 和 12.6 节。

模式 2 发送中断和模式 2 接收中断合并后送至 CPU,请求中断,软件响应中断后必须判别是 `SCON.TI` 中断请求还是 `SCON.RI` 中断请求,并清除相应的标志位。

12.4.4 标准 UART 工作模式 3: 9 位 UART, 波特率可变

配置寄存器 `SCON` 的 `SM0`、`SM1` 为 11 时, UART 工作于模式 3。此模式为 9 位 UART

工作模式，一帧信息为 11 位：1 位起始位，8 位数据位(低位在先)，1 位可编程位(SCON.TB8/SCON.RB8，第 9 位数据)和 1 位停止位。SCON.TB8/SCON.RB8 可以作为多机通信地址标志或奇偶校验位使用。UART_TX 为发送信息，UART_RX 为接收信息，以全双工模式进行接收/发送。接口时序如 Figure 100 和 Figure 101 所示。

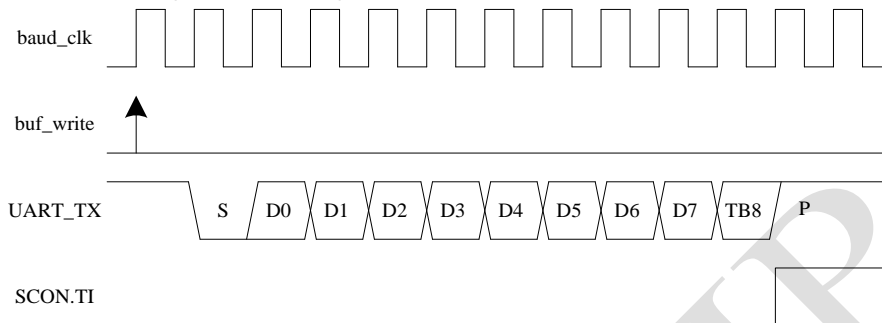


Figure 100 UART 模式 3 发送时序

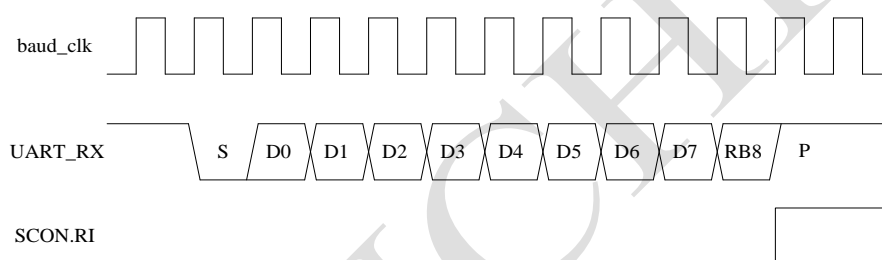


Figure 101 UART 模式 3 接收时序

UART 模式 3 中，波特率可以由内部的波特率产生器或 Timer 1 产生，具体参见 12.7 节。

配置寄存器 SCON 中的 SM2、TB8 可实现多机通信和奇偶校验，具体见 12.4.7.1 和 12.6 节。

模式 3 发送中断和模式 3 接收中断合并后送至 CPU，请求中断，软件响应中断后必须判别是 SCON.TI 中断请求还是 SCON.RI 中断请求，并清除相应的标志位。

12.4.5 38KHz 红外调制发送

UART 支持 38KHz 的红外调制发送功能。38KHz 红外调制原理为：将 TXD 的数据‘0’调制为占空比为 1/3 的 38KHz 信号，数据‘1’不调制。

驱动红外发送管的三极管可以是 NPN 型或 PNP 型，通过设置 SIRCON.IRINV 控制寄存器来选择，具体见 Figure 102，IRINV=0 对应 NPN 型，IRINV=1 对应 PNP 型。

通过改变寄存器 IRPD(见寄存器 SRELH 和 SIRCON)可调整至 38KHz 红外载波频率，分频系数寄存器 IRPD 计算公式为：

$$IRPD = \left(\frac{f_{sys_clk}(KHz)}{(3 * 38)(KHz)} \right)$$

在该工作模式下，发送为 38KHz 的红外调制信号，接收为标准 UART 信号。

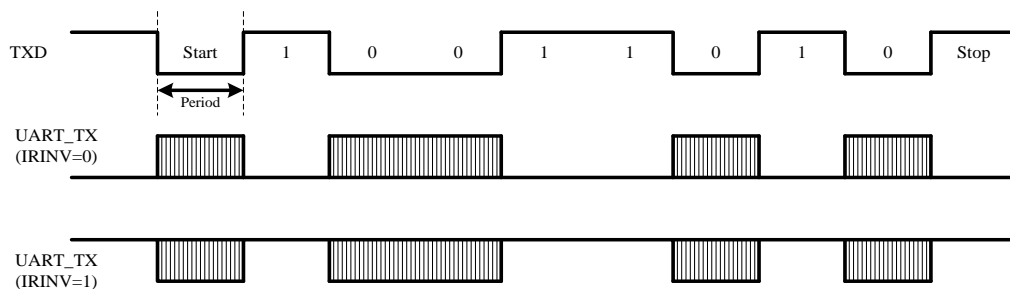


Figure 102 38KHz 红外调制发送时序

12.4.6 IrDA(SIR)红外通信

IrDA 是红外数据组织 (Infrared Data Association) 的简称, 在 IrDA 物理层中, 将数据通信按发送速率分为三类: SIR、MIR 和 FIR。串行红外 (SIR) 的速率覆盖了 RS-232 端口通常支持的速率 (9600bps~115.2Kbps)。

IrDA(SIR)为半双工通信, 有两种模式:

- 正常模式:
 - 发送时, '0'被调制成 3/16 比特宽度的高电平脉冲输出, 而'1'则被转换为持续的低电平输出。
 - 接收时, '0'对应的输入信号为 3/16 比特宽度的低电平脉冲, '1'对应的输入信号为整个比特宽度的高电平。
- 低功耗模式 (low-power IrDA):
 - 发送时, '0'被调制成 3 倍 $f_{SLPBaud}$ 比特宽度的高电平脉冲输出, 而'1'则被转换为持续的低电平输出。
 - 接收时, '0'对应的输入信号为 3 倍 $f_{SLPBaud}$ 比特宽度的低电平脉冲, '1'对应的输入信号为整个比特宽度的高电平。
 - 分频系数寄存器 IRPD 可根据 $f_{SLPBaud}$ 和 f_{sys_clk} 得到, 具体计算公式为:

$$IRPD = \left(\frac{f_{sys_clk}}{f_{SLPBaud}} \right)$$

其中 $1.35MHz < f_{SLPBaud} < 2.12MHz$ 。

IrDA(SIR)通信在发送和接收之间至少需要 10ms 的延迟, 这种延迟也被称为接收器建立时间。

IrDA 接收和发送时序参见 Figure 103。

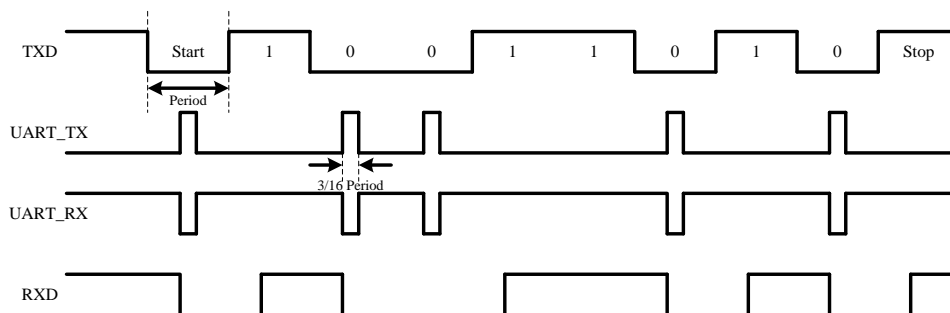


Figure 103 IrDA 接收和发送时序

12.4.7 LIN 模式

12.4.7.1 LIN 发送

UART 实现发送 13bit 断开符功能，操作如下：

- 配置寄存器 LINCON.LINEN 为 1，使能 LIN 模式；
- 向寄存器 LINCON.LBK 写 1，UART_TX 首先发送 13bit ‘0’ 作为断开符，再发送 1bit ‘1’，该过程完成后 SCON.TI 置位。

12.4.7.2 LIN 接收

当配置寄存器 LINCON.LINEN 为 1 时，LIN 模式使能，断开符检测电路被激活。该电路独立于 UART 接收器，不管 UART 总线是否空闲，断开符只要一出现就能被检测到。

同时，配置 SCON.REN 为 1，使能 UART 接收功能，若检测到 UART_RX 上的起始信号，断开符检测电路对接下来的每个位进行采样，如果连续检测到 10 个（LINCON.LBKL 设置为 0）或 11 个（LINCON.LBKL 设置为 1）采样位为‘0’，且随后又采样到一个‘1’，则 LINCON.LBKI 置位。如果 LINCON.LBKIEEN 置位，且 IEN0.6 和 IEN0.7 同时设置为 1，可使能 LIN 中断。

如果在第 10 或 11 个采样点之前采样到‘1’，检查电路放弃当前检测，重新开始监测起始位。

以 10bit 断开符为例，UART_RX 接收到的断开符长度和 LINCON.LBKI 之间关系如下：Figure 104 所示断开符长度不足情况，Figure 105 所示断开符长度正好情况，Figure 106 所示断开符长度足够长情况。

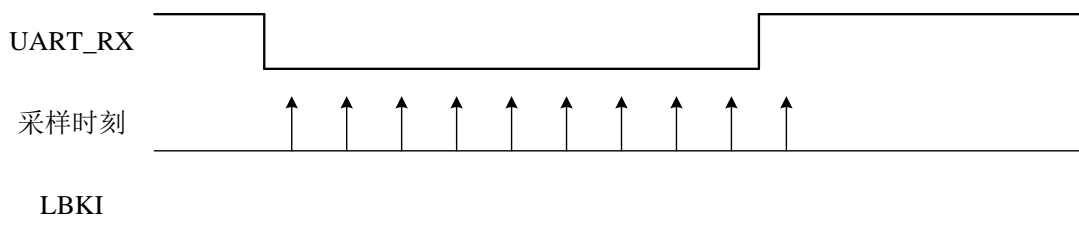


Figure 104 断开符长度不足

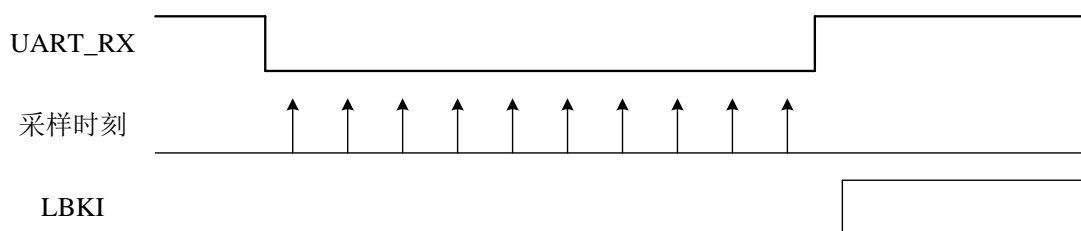


Figure 105 断开符长度正好

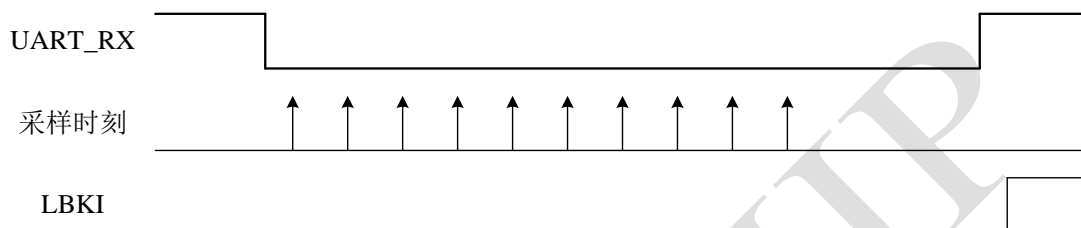


Figure 106 断开符长度足够长

12.5 UART 多机通信

UART 模式 2 和模式 3 通过使用第 9 数据位 (SCON.TB8/SCON.RB8) 可以支持一个主机与多个从机之间的多机通信。当主机要发送数据给多个从机时, 它先发送一个用于选择目标从机的地址字节。

地址字节与数据字节的区别是: 地址字节的第 9 位为逻辑 1, 数据字节的第 9 位为逻辑 0。

如果从机的寄存器 SCON.SM2 位被置 1, 则只有当 UART 接收到的第 9 位为 1 (SCON.RB8 = 1) 时, UART 才会产生接收中断。在 UART 的中断处理程序中, 软件将接收到的地址与从机自身的 8 位地址进行比较, 如果地址匹配, 从机将它的寄存器 SCON.SM2 清零, 从而允许后面接收数据字节时产生中断。未被寻址的从机保持 SCON.SM2 位为 1, 在收到后续的数据字节时不产生中断, 从而忽略接收到的数据。一旦数据块传输完毕, 被寻址的从机将它的 SCON.SM2 位重新置 1, 以忽略所有的数据传输, 直到它收到下一个地址字节。

可以将一个地址分配给多个从机, 从而允许同时向多个从机“广播”发送。

可以将多个地址分配给一个从机。

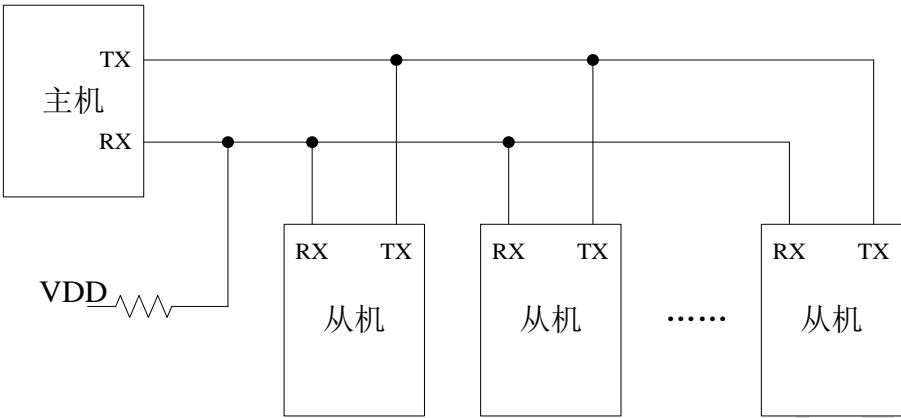


Figure 107 UART 多机通信连接方式

12.6 UART 奇偶校验

UART 模式 2 和模式 3 中每帧数据中含有第 9 数据位 (SCON.TB8/SCON.RB8)，该位可以实现奇偶校验的功能。

配置寄存器 SCON.SM2 为 0，使得 UART 工作在非多机通信模式。

发送时：SCON 中的 TB8 配置建议：将发送数据写入寄存器 A，然后将寄存器 PSW 中的奇/偶校验位 P 写入 SCON.TB8。

接收时：硬件自动把 UART 接收到的第 9 位数据写入 SCON 寄存器的 RB8，软件可将接收数据写入寄存器 A，然后将寄存器 PSW 中的奇/偶校验位 P 和 SCON 寄存器的 RB8 进行比较，来判断奇偶校验是否正确。

12.7 UART 波特率计算

UART 波特率计算：

- UART 模式 0

$$baud\ rate = \frac{f_{sys_clk}}{12}$$

- UART 模式 1, 其中 UART 的 UBSEL 对应 TCON.0; UART 的 SMOD 对应 TCON.2。

➤ 寄存器 UBSEL=0，使用 Timer1 的模式 2 作为波特率产生器：

$$baud\ rate = \frac{2^{SMOD} * f_{sys_clk}}{32 * (256 - TH1)}$$

➤ 寄存器 UBSEL=1，使用 UART 自带波特率产生器：

$$baud\ rate = \frac{2^{SMOD} * f_{sys_clk}}{32 * (1024 - SREL)}$$

Table 173 UART 模式 1 波特率 (f_{sys_clk}=24.576M)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)

4800	96	4800.0	0.0000	x	x	x
9600	176	9600.0	0.0000	96	9600.0	0.0000
19200	216	19200.0	0.0000	176	19200.0	0.0000
38400	236	38400.0	0.0000	216	38400.0	0.0000
56000	242	54857.1	-2.0408	229	56888.9	1.5873
57600	243	59076.9	2.5641	229	56888.9	-1.2346
115200	249	109714.3	-4.7619	243	118153.8	2.5641
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
1200	384	1200.0	0.0000	x	x	x
2400	704	2400.0	0.0000	384	2400.0	0.0000
4800	864	4800.0	0.0000	704	4800.0	0.0000
9600	944	9600.0	0.0000	864	9600.0	0.0000
19200	984	19200.0	0.0000	944	19200.0	0.0000
38400	1004	38400.0	0.0000	984	38400.0	0.0000
56000	1010	54857.1	-2.0408	997	56888.9	1.5873
57600	1011	59076.9	2.5641	997	56888.9	-1.2346
115200	1017	109714.3	-4.7619	1011	118153.8	2.5641

 Table 174 UART 模式 1 波特率 ($f_{sys_clk}=12.288M$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
2400	96	2400.0	0.0000	x	x	x
4800	176	4800.0	0.0000	96	4800.0	0.0000
9600	216	9600.0	0.0000	176	9600.0	0.0000
19200	236	19200.0	0.0000	216	19200.0	0.0000
38400	246	38400.0	0.0000	236	38400.0	0.0000
56000	249	54857.1	-2.0408	242	54857.1	-2.0408
57600	249	54857.1	-4.7619	243	59076.9	2.5641
115200	x	x	x	249	109714.3	-4.7619
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
600	384	600.0	0.0000	x	x	x

1200	704	1200.0	0.0000	384	1200.0	0.0000
2400	864	2400.0	0.0000	704	2400.0	0.0000
4800	944	4800.0	0.0000	864	4800.0	0.0000
9600	984	9600.0	0.0000	944	9600.0	0.0000
19200	1004	19200.0	0.0000	984	19200.0	0.0000
38400	1014	38400.0	0.0000	1004	38400.0	0.0000
56000	1017	54857.1	-2.0408	1010	54857.1	-2.0408
57600	1017	54857.1	-4.7619	1011	59076.9	2.5641
115200	x	x	x	1017	109714.3	-4.7619

Table 175 UART 模式 1 波特率 ($f_{\text{sys_clk}}=6.144\text{M}$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
1200	96	1200.0	0.0000	x	x	x
2400	176	2400.0	0.0000	96	2400.0	0.0000
4800	216	4800.0	0.0000	176	4800.0	0.0000
9600	236	9600.0	0.0000	216	9600.0	0.0000
19200	246	19200.0	0.0000	236	19200.0	0.0000
38400	251	38400.0	0.0000	246	38400.0	0.0000
56000	x	x	x	249	54857.1	-2.0408
57600	x	x	x	249	54857.1	-4.7619
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
300	384	300.0	0.0000	x	x	x
600	704	600.0	0.0000	384	600.0	0.0000
1200	864	1200.0	0.0000	704	1200.0	0.0000
2400	944	2400.0	0.0000	864	2400.0	0.0000
4800	984	4800.0	0.0000	944	4800.0	0.0000
9600	1004	9600.0	0.0000	984	9600.0	0.0000
19200	1014	19200.0	0.0000	1004	19200.0	0.0000
38400	1019	38400.0	0.0000	1014	38400.0	0.0000
56000	x	x	x	1017	54857.1	-2.0408
57600	x	x	x	1017	54857.1	-4.7619

Table 176 UART 模式 1 波特率 ($f_{\text{sys_clk}}=3.072\text{M}$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
600	96	600.0	0.0000	x	x	x
1200	176	1200.0	0.0000	96	1200.0	0.0000
2400	216	2400.0	0.0000	176	2400.0	0.0000
4800	236	4800.0	0.0000	216	4800.0	0.0000
9600	246	9600.0	0.0000	236	9600.0	0.0000
19200	251	19200.0	0.0000	246	19200.0	0.0000
38400	x	x	x	251	38400.0	0.0000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
300	704	300.0	0.0000	384	300.0	0.0000
600	864	600.0	0.0000	704	600.0	0.0000
1200	944	1200.0	0.0000	864	1200.0	0.0000
2400	984	2400.0	0.0000	944	2400.0	0.0000
4800	1004	4800.0	0.0000	984	4800.0	0.0000
9600	1014	9600.0	0.0000	1004	9600.0	0.0000
19200	1019	19200.0	0.0000	1014	19200.0	0.0000
38400	x	x	x	1019	38400.0	0.0000

Table 177 UART 模式 1 波特率 ($f_{\text{sys_clk}}=1.536\text{M}$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
300	96	300.0	0.0000	x	x	x
600	176	600.0	0.0000	96	600.0	0.0000
1200	216	1200.0	0.0000	176	1200.0	0.0000
2400	236	2400.0	0.0000	216	2400.0	0.0000
4800	246	4800.0	0.0000	236	4800.0	0.0000
9600	251	9600.0	0.0000	246	9600.0	0.0000
19200	x	x	x	251	19200.0	0.0000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)

300	864	300.0	0.0000	704	300.0	0.0000
600	944	600.0	0.0000	864	600.0	0.0000
1200	984	1200.0	0.0000	944	1200.0	0.0000
2400	1004	2400.0	0.0000	984	2400.0	0.0000
4800	1014	4800.0	0.0000	1004	4800.0	0.0000
9600	1019	9600.0	0.0000	1014	9600.0	0.0000
19200	x	x	x	1019	19200.0	0.0000

Table 178 UART 模式 1 波特率 ($f_{sys_clk}=0.768M$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
300	176	300.0	0.0000	96	300.0	0.0000
600	216	600.0	0.0000	176	600.0	0.0000
1200	236	1200.0	0.0000	216	1200.0	0.0000
2400	246	2400.0	0.0000	236	2400.0	0.0000
4800	251	4800.0	0.0000	246	4800.0	0.0000
9600	x	x	x	251	9600.0	0.0000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
300	944	300.0	0.0000	864	300.0	0.0000
600	984	600.0	0.0000	944	600.0	0.0000
1200	1004	1200.0	0.0000	984	1200.0	0.0000
2400	1014	2400.0	0.0000	1004	2400.0	0.0000
4800	1019	4800.0	0.0000	1014	4800.0	0.0000
9600	x	x	x	1019	9600.0	0.0000

Table 179 UART 模式 1 波特率 ($f_{sys_clk}=22.1184M$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
4800	112	4800.0	0.0000	x	x	x
9600	184	9600.0	0.0000	112	9600.0	0.0000
19200	220	19200.0	0.0000	184	19200.0	0.0000
38400	238	38400.0	0.0000	220	38400.0	0.0000
51200	244	57600.0	2.8571	231	55296.0	-1.2571

57600	244	57600.0	0.0000	232	57600.0	0.0000
115200	250	115200.0	0.0000	244	115200.0	0.0000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
1200	448	1200.0	0.0000	x	x	x
2400	736	2400.0	0.0000	448	2400.0	0.0000
4800	880	4800.0	0.0000	736	4800.0	0.0000
9600	952	9600.0	0.0000	880	9600.0	0.0000
19200	988	19200.0	0.0000	952	19200.0	0.0000
38400	1006	38400.0	0.0000	988	38400.0	0.0000
51200	1012	57600.0	2.8571	999	55296.0	-1.2571
57600	1012	57600.0	0.0000	1000	57600.0	0.0000
115200	1018	115200.0	0.0000	1012	115200.0	0.0000

Table 180 UART 模式 1 波特率 ($f_{sys_clk}=11.0592M$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
2400	112	2400.0	0.0000	x	x	x
4800	184	4800.0	0.0000	112	4800.0	0.0000
9600	220	9600.0	0.0000	184	9600.0	0.0000
19200	238	19200.0	0.0000	220	19200.0	0.0000
38400	247	38400.0	0.0000	238	38400.0	0.0000
56000	250	57600.0	2.8571	244	57600.0	2.8571
57600	250	57600.0	0.0000	244	57600.0	0.0000
115200	253	115200.0	0.0000	250	115200.0	0.0000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
600	448	600.0	0.0000	x	x	x
1200	736	1200.0	0.0000	448	1200.0	0.0000
2400	880	2400.0	0.0000	736	2400.0	0.0000
4800	952	4800.0	0.0000	880	4800.0	0.0000
9600	988	9600.0	0.0000	952	9600.0	0.0000
19200	1006	19200.0	0.0000	988	19200.0	0.0000

38400	1015	38400.0	0.0000	1006	38400.0	0.0000
56000	1018	57600.0	2.8571	1012	57600.0	2.8571
57600	1018	57600.0	0.0000	1012	57600.0	0.0000
115200	1021	115200.0	0.0000	1018	115200.0	0.0000

Table 181 UART 模式 1 波特率 ($f_{sys_clk}=5.5296M$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
1200	112	1200.0	0.0000	x	x	x
2400	184	2400.0	0.0000	112	2400.0	0.0000
4800	220	4800.0	0.0000	184	4800.0	0.0000
9600	238	9600.0	0.0000	220	9600.0	0.0000
19200	247	19200.0	0.0000	238	19200.0	0.0000
38400	x	x	x	247	38400.0	0.0000
56000	253	57600.0	2.8571	250	57600.0	2.8571
57600	253	57600.0	0.0000	250	57600.0	0.0000
115200	x	x	x	253	115200.0	0.0000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
300	448	300.0	0.0000	x	x	x
600	736	600.0	0.0000	448	600.0	0.0000
1200	880	1200.0	0.0000	736	1200.0	0.0000
2400	952	2400.0	0.0000	880	2400.0	0.0000
4800	988	4800.0	0.0000	952	4800.0	0.0000
9600	1006	9600.0	0.0000	988	9600.0	0.0000
19200	1015	19200.0	0.0000	1006	19200.0	0.0000
38400	x	x	x	1015	38400.0	0.0000
56000	1021	57600.0	2.8571	1018	57600.0	2.8571
57600	1021	57600.0	0.0000	1018	57600.0	0.0000
115200	x	x	x	1021	115200.0	0.0000

Table 182 UART 模式 1 波特率 ($f_{sys_clk}=2.7648M$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)

600	112	600.0	0.0000	x	x	x
1200	184	1200.0	0.0000	112	1200.0	0.0000
2400	220	2400.0	0.0000	184	2400.0	0.0000
4800	238	4800.0	0.0000	220	4800.0	0.0000
9600	247	9600.0	0.0000	238	9600.0	0.0000
19200	x	x	x	247	19200.0	0.0000
56000	x	x	x	253	57600.0	2.8571
57600	x	x	x	253	57600.0	0.0000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
300	736	300.0	0.0000	448	300.0	0.0000
600	880	600.0	0.0000	736	600.0	0.0000
1200	952	1200.0	0.0000	880	1200.0	0.0000
2400	988	2400.0	0.0000	952	2400.0	0.0000
4800	1006	4800.0	0.0000	988	4800.0	0.0000
9600	1015	9600.0	0.0000	1006	9600.0	0.0000
19200	x	x	x	1015	19200.0	0.0000
56000	x	x	x	1021	57600.0	2.8571
57600	x	x	x	1021	57600.0	0.0000

Table 183 UART 模式 1 波特率 ($f_{sys_clk}=1.3824M$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
300	112	300.0	0.0000	x	x	x
600	184	600.0	0.0000	112	600.0	0.000
1200	220	1200.0	0.0000	184	1200.0	0.000
2400	238	2400.0	0.0000	220	2400.0	0.000
4800	247	4800.0	0.0000	238	4800.0	0.000
9600	x	x	x	247	9600.0	0.000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
300	880	300.0	0.0000	736	300.0	0.0000
600	952	600.0	0.0000	880	600.0	0.0000

1200	988	1200.0	0.0000	952	1200.0	0.0000
2400	1006	2400.0	0.0000	988	2400.0	0.0000
4800	1015	4800.0	0.0000	1006	4800.0	0.0000
9600	x	x	x	1015	9600.0	0.0000

Table 184 UART 模式 1 波特率 ($f_{sys_clk}=0.6912M$)

波特率	UBSEL=0, SMOD=0			UBSEL=0, SMOD=1		
	TH1(十进制)	实际值	偏差(%)	TH1(十进制)	实际值	偏差(%)
300	184	300.0	0.0000	112	300.0	0.0000
600	220	600.0	0.0000	184	600.0	0.0000
1200	238	1200.0	0.0000	220	1200.0	0.0000
2400	247	2400.0	0.0000	238	2400.0	0.0000
4800	x	x	x	247	4800.0	0.0000
波特率	UBSEL=1, SMOD=0			UBSEL=1, SMOD=1		
	SREL(十进制)	实际值	偏差(%)	SREL(十进制)	实际值	偏差(%)
300	952	300.0	0.0000	880	300.0	0.0000
600	988	600.0	0.0000	952	600.0	0.0000
1200	1200.0	0.0000	0.0000	988	1200.0	0.0000
2400	1015	2400.0	0.0000	1006	2400.0	0.0000
4800	x	x	x	1015	4800.0	0.0000

- UART 模式 2，其中 UART 的 SMOD 对应 TCON.2

$$baud\ rate = \frac{2^{SMOD} * f_{sys_clk}}{64}$$

- UART 模式 3

计算方法和模式 1 相同

12.8 UART 寄存器

UART 模块有 6 个寄存器，对应的 SFRPAGE 为 0，具体说明如 Table 185 所示。

Table 185 UART 寄存器 (SFRPAGE=0x00)

地址	寄存器名	属性	复位值	功能描述
0x98	SCON	W/R	0x00	UART 控制寄存器
0x99	SBUF	W/R	0x00	UART 数据缓冲寄存器
0x9A	SRELL	W/R	0x00	UART 波特率重载寄存器低 8 位
0x9B	SRELH	W/R	0x00	UART 波特率重载寄存器高 2 位

地址	寄存器名	属性	复位值	功能描述
0x9C	SIRCON	W/R	0x00	UART 红外控制寄存器
0x9D	LINCON	W/R	0x00	LIN 控制寄存器

12.8.1 UART 控制寄存器(SCON) [0x98]

Table 186 UART 控制寄存器(SCON)

参数名	比特位	属性	复位值	描述
RI	0	W/R	0	接收中断标志： <ul style="list-style-type: none"> 当有接收中断申请时，该位置 1 软件配置该位为 1 可产生软中断，在中断处理程序中需要软件写 0 来清除
TI	1	W/R	0	发送中断标志： <ul style="list-style-type: none"> 当有发送中断申请时，该位置为 1 软件配置该位为 1 可产生软中断，在中断处理程序中需要软件写 0 来清除
RB8	2	W/R	0	接收数据位 RB8： 模式 1 中若 SM2=0，则为收到的停止位 模式 2、3 中是收到的第 9 数据位
TB8	3	W/R	0	发送数据位 TB8： 模式 2、3 中是被发出去的第 9 数据位 由软件置 1/清 0
REN	4	W/R	0	允许接收位： 0：不允许接收 1：允许接收
SM2	5	W/R	0	多机交互使能位 0：不使能 1：使能
SM1	6	W/R	0	具体说明参见 Table 187
SM0	7	W/R	0	

Table 187 UART 工作方式选择

SM0	SM1	模式	描述	波特率
0	0	0	移位寄存器	$f_{\text{sys_clk}}/12$

0	1	1	8 位 UART	可变
1	0	2	9 位 UART	$f_{\text{sys_clk}}/32$ 或 $f_{\text{sys_clk}}/64$
1	1	3	9 位 UART	可变

12.8.2 UART 数据缓冲寄存器(SBUF) [0x99]

Table 188 UART 数据缓冲寄存器(SBUF)

参数名	比特位	属性	复位值	描述
SBUF	7-0	W/R	0	UART 数据缓冲寄存器： 对寄存器 SBUF 写操作，则 UART 将开始向外传输发送缓存数据； 对寄存器 SBUF 读操作，则 UART 将从串行接收缓存中读取数据

12.8.3 UART 波特率重载寄存器低 8 位(SRELL) [0x9A]

Table 189 UART 波特率重载寄存器低 8 位(SRELL)

参数名	比特位	属性	复位值	描述
SRELL	7-0	W/R	0	UART 波特率重载寄存器低 8 位： 通过配置该寄存器改变 UART 的通讯波特率。

12.8.4 UART 波特率重载寄存器高 2 位(SRELH) [0x9B]

Table 190 UART 波特率重载寄存器高 2 位(SRELH)

参数名	比特位	属性	复位值	描述
SRELH	1-0	W/R	0	UART 波特率重载寄存器高 2 位： 通过配置该寄存器改变 UART 的通讯波特率。
IRPD[9:4]	7-2	W/R	0	UART 红外调制频率控制位高 6 位

12.8.5 UART 红外控制寄存器(SIRCON) [0x9C]

Table 191 UART 红外控制寄存器(SIRCON)

参数名	比特位	属性	复位值	描述
IRPD[3:0]	3-0	W/R	0	UART 红外调制频率控制位低 4 位： 当工作在 low-power IrDA 模式时，用作 SIR 的 low-power 模式时钟分频系数，选择分频系数使 $1.35\text{MHz} < f_{\text{SLPBaud}} < 2.12\text{MHz}$ ；

参数名	比特位	属性	复位值	描述
				当工作在 38KHz 红外调制发送模式时，用作 38KHz 时钟分频控制。 具体见 38KHz 红外调制发送和 IrDA (SIR) 红外通信描述
IRINV	4	W/R	0	38KHz 红外调试模式，数据‘1’不调制输出时电平： 0：低电平 1：高电平
LBEN	5	W/R	0	回环测试使能位： 0：正常功能模式 1：回环测试模式
UMS	7-6	W/R	0	UART 工作模式选择控制： 00：标准 UART 模式 01：UART 38KHz 红外调试发送模式 10：3/16 IrDA 模式 11：low-power IrDA 模式

12.8.6 LIN 控制寄存器(LINCON) [0x9D]

Table 192LIN 控制寄存器(LINCON)

参数名	比特位	属性	复位值	描述
LINEN	0	W/R	0	LIN 使能： 0：不使能 1：使能
LBK	1	W/R	0	发送 13bit 断开符： 0：无效 1：发送断开帧
LBKL	2	W/R	0	检测断开帧长度： 0：10 1：11
LBKIEN	3	W/R	0	断开帧中断使能： 0：不使能 1：使能

参数名	比特位	属性	复位值	描述
LBKI	4	W/R	0	断开帧中断控制： <ul style="list-style-type: none">● 检测到断开帧后，该位置为 1● 需要软件写 0 清除● 支持写 1 软中断
Reserved	7-5	R	0	保留

13 I2C 接口

13.1 概述

JMT18F003PLUS 的 I2C 接口支持：

- master/slave 工作模式；
- 单 master/多 master 操作；
- 7/10 位设备地址寻址；
- master/slave 模式下简单的读写操作；
- master/slave 模式下组合读写操作；
- SCL 高低电平宽度可编程；
- I2C 总线的 SCL 时钟频率为：
 - I2C 总线标准模式：100KHz
 - I2C 总线快速模式：400KHz

13.2 框图

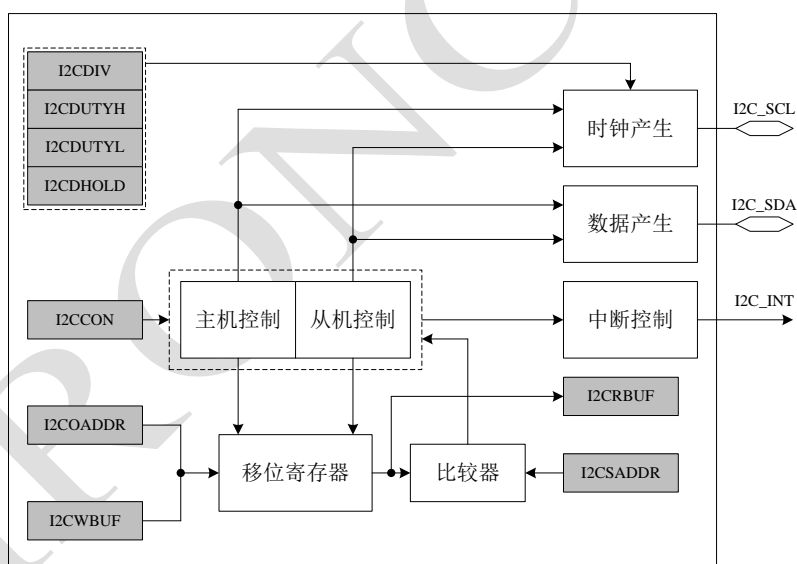


Figure 108 I2C 结构框图

13.3 I2C 管脚配置

I2C 的管脚配置可参见 Table 1 引脚说明表格，管脚配置说明见 GPIO 寄存器说明。

13.4 I2C 模块连接

每个连接到 I2C 总线的设备都有一个独一无二的地址识别。连接到 I2C 的设备都可以作为主机或从机。主机设备启动总线数据传输并产生传输的时钟信号。传输过程中，被主机寻址的设备是从机。

为了数据传输，I2C 接口有一根串行数据管脚（SDA）和一根串行时钟管脚（SCL）。两个管脚都具有漏极开路特性，来实现所需要的线与功能。它们都要用一个上拉电阻连接到正电压。总线空闲时，两个管脚都是高电平。

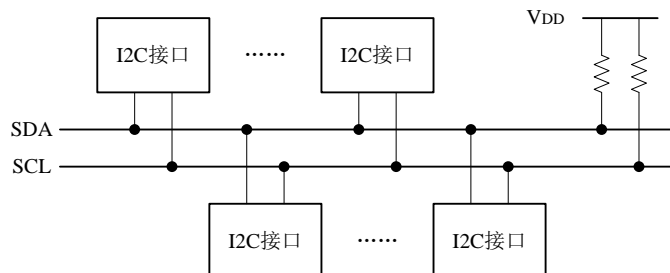


Figure 109 I2C 模块连接

13.5 I2C 数据格式

I2C 接口的时序图如下所示：

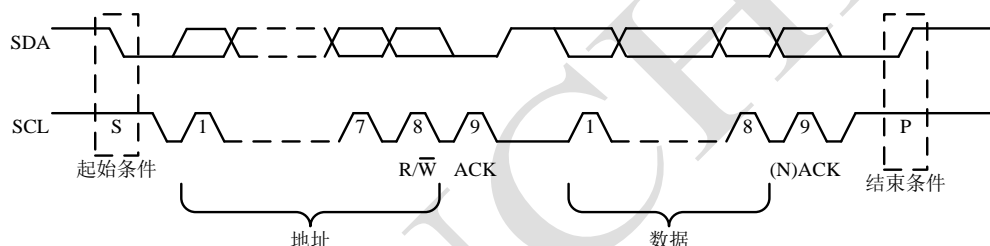


Figure 110 I2C 接口时序

13.5.1 数据有效

在传输数据的时候，SDA 线必须在时钟的高电平周期保持稳定，SDA 的高或低电平状态只有在 SCL 线是低电平时才能改变。

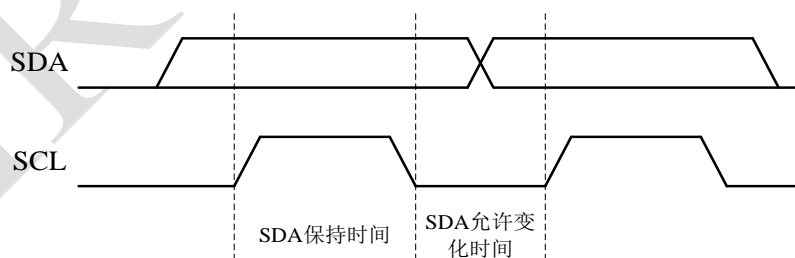


Figure 111 I2C 位传输

13.5.2 读写控制比特

在 I2C 总线中，首字节（地址或帧头字节）的第 8 位为读写控制比特（R/W）。

读写控制比特为 0 时，表示主机的写模式；读写控制比特为 1 时，表示主机的读模式。

13.5.3 反馈 (ACK)

不管是主机还是从机,在发送完 1 个字节后,都会等待接下来的一个比特作为反馈信号。

接收到的反馈信号为 0 时,表示发送字节被应答 (ACK);接收到的反馈信号为 1 时,表示发送字节未被应答 (NACK)。

13.5.4 起始条件和结束条件

在 I2C 总线中,起始条件 (S) 和结束条件 (P) 的情况:

- 起始条件指当 SCL 线是高电平时,SDA 线从高电平向低电平切换。
- 结束条件指当 SCL 线是高电平时,SDA 线由低电平向高电平切换。

起始和结束条件一般由主机产生。总线在起始条件后被认为处于忙的状态。在结束条件的某段时间后,总线被认为再次处于空闲状态。

如果产生重复起始 (Sr) 条件而不产生结束条件,总线会一直处于忙的状态。此时的起始条件 (S) 和重复起始 (Sr) 条件在功能上是一样的。

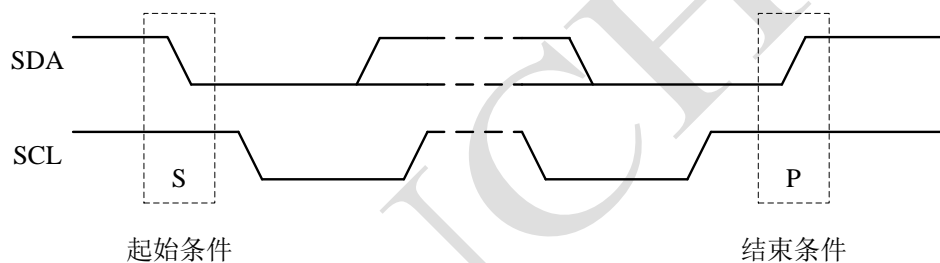


Figure 112 起始和结束条件

13.5.5 数据格式

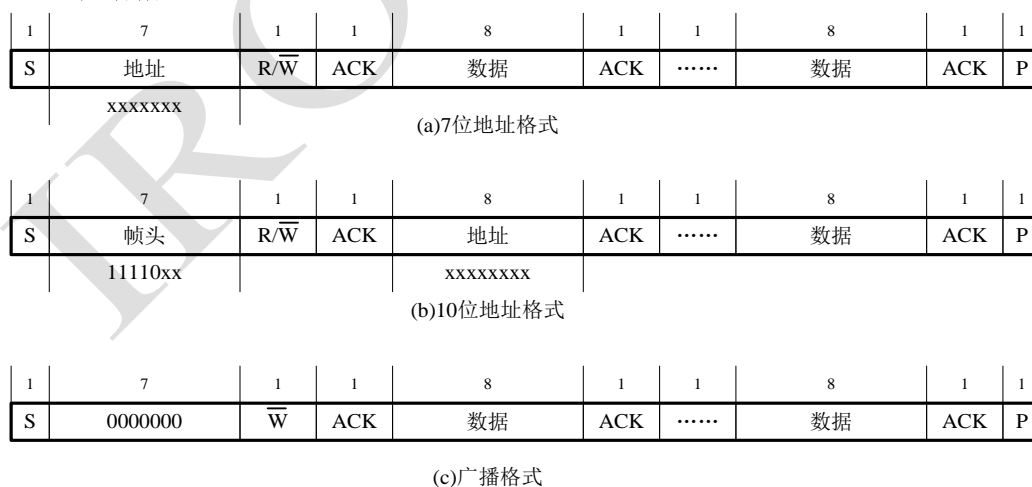


Figure 113 数据格式

7 位地址格式,首字节由“地址”+“读写控制比特”组成。

10 位地址格式,首字节由“帧头”+“读写控制比特”组成,其中帧头由“11110”+“10 位地址的高 2 比特”组成;第二个字节由 10 位地址的低 8 比特组成。

广播格式，首字节由“0000000”+“写比特”组成。

13.5.6 时钟同步

所有主机在 SCL 线上产生它们自己的时钟来传输 I2C 总线上的信息。数据只在时钟的高电平周期有效。因此需要一个确定的时钟进行逐位仲裁。

时钟同步通过线与连接 I2C 接口到 SCL 线来执行。这就是说：SCL 线的高到低切换会使器件开始数它们的低电平周期，而且一旦器件的时钟变低电平，它会使 SCL 线保持这种状态直到到达时钟的高电平。但是，如果另一个时钟仍处于低电平周期，这个时钟的低到高切换不会改变 SCL 线的状态。因此，SCL 线被有最长低电平周期的器件保持低电平。此时，低电平周期短的器件会进入高电平的等待状态。

当所有有关的器件数完了它们的低电平周期后，时钟线被释放并变成高电平。之后，器件时钟和 SCL 线的状态没有差别。而且所有器件会开始数它们的高电平周期。首先完成高电平周期的器件会再次将 SCL 线拉低。

这样产生的同步 SCL 时钟的低电平周期由低电平时钟周期最长的器件决定而高电平周期由高电平时钟周期最短的器件决定。

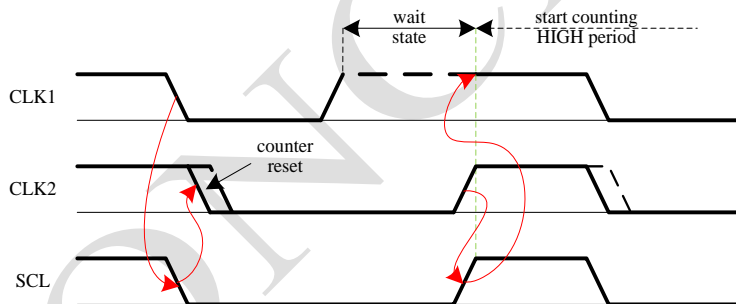


Figure 114 仲裁过程中的时钟同步

13.5.7 仲裁

主机只能在总线空闲的时候启动传输，两个或多个主机可能同时产生一个起始条件。

当 SCL 线是高电平时，仲裁在 SDA 线发生；这样在其他主机发送低电平时，发送高电平的主机将断开它的数据输出级，因为总线上的电平与它自己的电平不相同。

仲裁可以持续多位。它的第一个阶段是比较地址位。如果每个主机都尝试寻址相同的器件，仲裁会继续比较数据位（如果是主机发送模式），或者比较响应位（如果是主机接收模式）。因为 I2C 总线的地址和数据信息由赢得仲裁的主机决定，在仲裁过程中不会丢失信息。

丢失仲裁的主机立即切换到它的从机模式，置位寄存器 I2CISC.ALINT。

Figure 115 显示了两个主机的仲裁过程。图中产生 DATA1 的主机的内部数据电平与 SDA 线的实际电平不同，关断数据输出就意味着总线连接了一个高电平，这不会影响赢得仲裁的主机的数据传输。

必须特别注意的是：在串行传输时当重复起始条件或停止条件发送到 I2C 总线的时候，仲裁过程仍在进行。如果可能产生这样的情况，有关的主机必须在帧格式相同位置发送重复起始条件或停止条件。也就是说，仲裁不能在下面情况之间进行：

- 重复起始条件和数据位
- 停止条件和数据位
- 重复起始条件和停止条件

从机不被卷入仲裁过程。

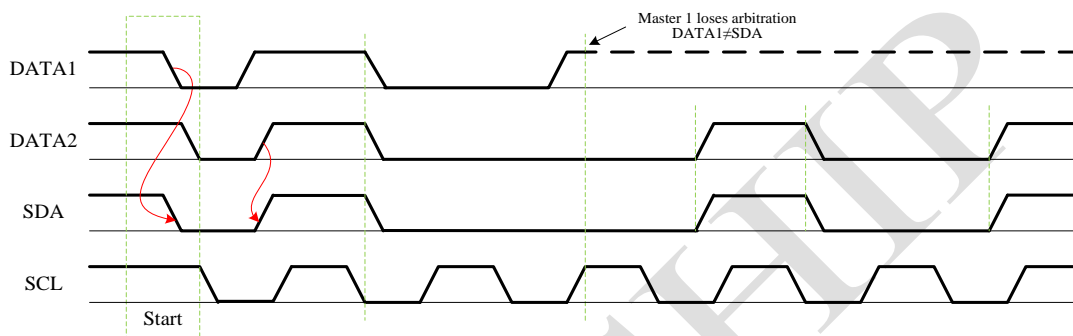


Figure 115 两个主机的仲裁过程

13.6 I2C 时钟产生

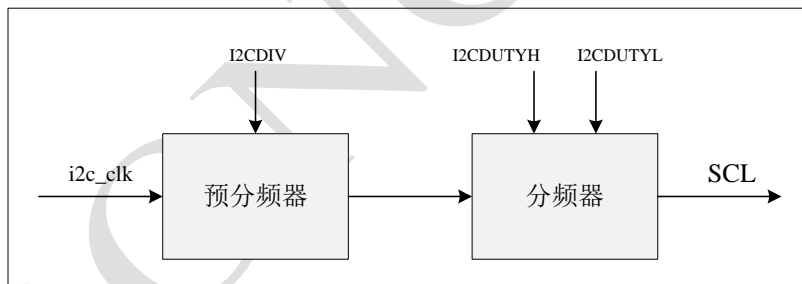


Figure 116 I2C 时钟产生示意图

主机模式下，I2C 串行时钟 SCL 由模块工作时钟 i2c_clk 经过分频得到。

SCL 高电平周期： $T_{SCL_HIGH} = T_{i2c_clk} * NDIV * (NHIGH + 1)$

SCL 低电平周期： $T_{SCL_LOW} = T_{i2c_clk} * NDIV * (NLOW + 1)$

SCL 下降沿到 SDA 变化时间： $T_{HOLD} = T_{i2c_clk} * NDIV * (NHOLD + 1)$

其中，NDIV 见寄存器 I2CDIV 说明，NHIGH 见寄存器 I2CDUTYH 说明，NLOW 见寄存器 I2CDUTYL 说明，NHOLD 见寄存器 I2CHOLD 说明。

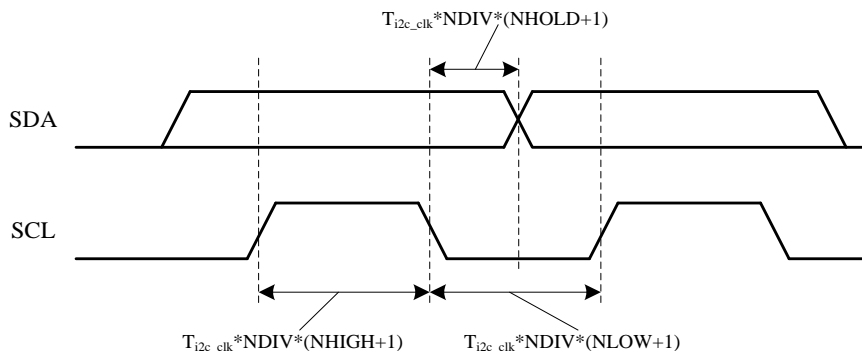


Figure 117 I2C 时钟分频

13.7 I2C 工作模式

13.7.1 从机模式

13.7.1.1 从机地址匹配

I2C 接口默认条件下处于从机模式，一旦检测到起始条件，在 SDA 线上接收到的地址被送到移位寄存器，并与 I2C 本机地址寄存器 I2COADDH 和 I2COADDL 比较，或者和广播呼叫地址比较，若比对结果一致，则主机要对本从机进行访问。

器件地址匹配的标准：

- 若 I2COADDRH.O10BEN=0，即 7 位地址时，接收的第一个字节的地址部分和 I2COADDL[6:0]一致时，器件地址匹配。
- 若 I2COADDRH.O10BEN=1，即 10 位地址时，接收的第一个字节的帧头部分为“11110”+“2 位地址”+“读写控制比特”，其中“2 位地址”和 I2COADDH[1:0]一致；接收的第二个字节和 I2COADDL[7:0]一致。当两个条件都满足时，器件地址匹配。
- 若 I2CCON.ADDR0EN=1，即广播模式使能时，接收的第一个字节为“00000000”时，器件地址匹配。

13.7.1.2 从机接收模式

当从机地址匹配后，且“读写控制比特”为 0，I2C 进入从机接收模式。

每接收一个数据，都会产生 RXINT 中断，需要读取 I2CRBUF 中数据，才能释放 SCL 线。若需继续从主机接收数据，软件参照 EV1 进行配置，产生 ACK 条件；若需停止接收数据，软件参照 EV2 进行配置，产生 NACK 条件。

图中 EV_x 代表软件响应中断之后所要完成的任务，以下类同。

EV1：（1）清除 RXINT；（2）配置 I2CCON.NACK 为 0；（3）读取 I2CRBUF 数据。

EV2：（1）清除 RXINT；（2）配置 I2CCON.NACK 为 1；（3）读取 I2CRBUF 数据。

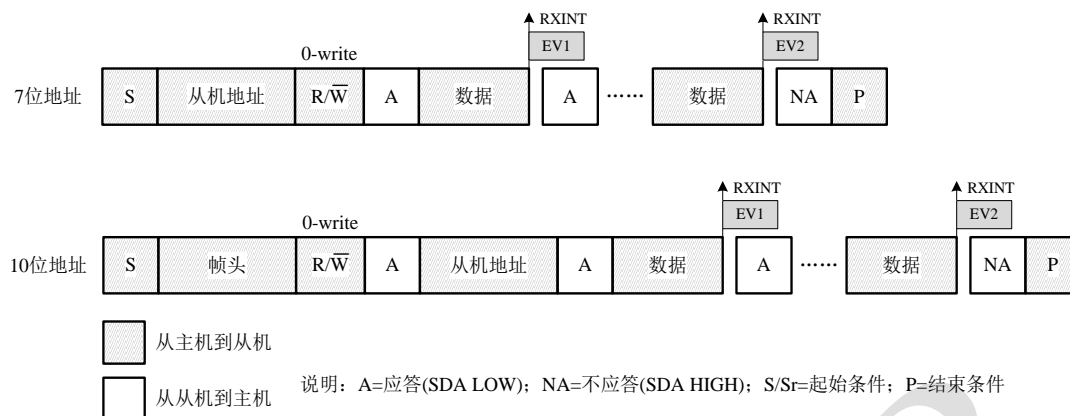


Figure 118 从机接收模式

13.7.1.3 从机发送模式

当从机地址匹配后，且“读写控制比特”为1，I2C 进入从机发送模式。

地址匹配后，从机发送反馈信号，同时产生 TXINT 中断，需软件向寄存器 I2CWBUF 中写入数据，才能释放 SCL 线。

EV3: (1) 清除 TXINT; (2) 向 I2CWBUF 写入数据。

EV4: (1) 清除 TXINT; (2) 向 I2CWBUF 写入任意数据。

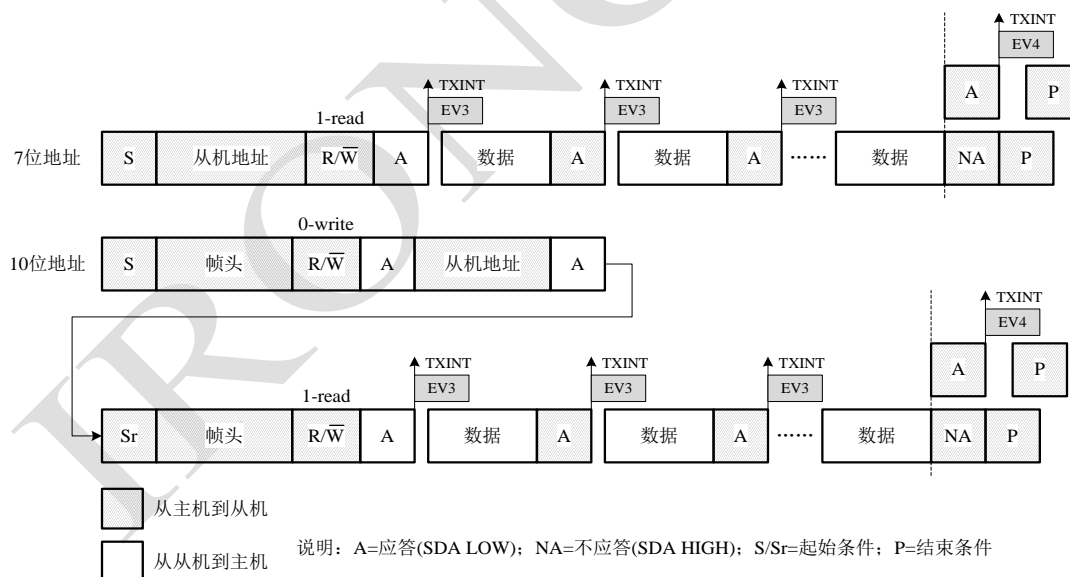


Figure 119 从机发送模式

13.7.1.4 从机混合模式

当从机地址匹配后，且“读写控制比特”为0，I2C 进入从机接收模式。每接收一个数据，都会产生 RXINT 中断，需要读取 I2CRBUF 中数据，才能释放 SCL 线。若需继续从主机接收数据，软件参照 EV1 进行配置，产生 ACK 条件。

在从机接收数据过程中，未发送 NACK 条件，而再次检测到 Sr 条件，且从机地址匹配，“读写控制比特”为 1，I2C 进入从机发送模式。从机发送反馈信号，同时产生 TXINT 中断，需软件向寄存器 I2CWBUF 中写入数据，才能释放 SCL 线。

- EV1: (1) 清除 RXINT; (2) 读取 I2CRBUF 数据。
- EV3: (1) 清除 TXINT; (2) 向 I2CWBUF 写入数据。
- EV4: (1) 清除 TXINT; (2) 向 I2CWBUF 写入任意数据。

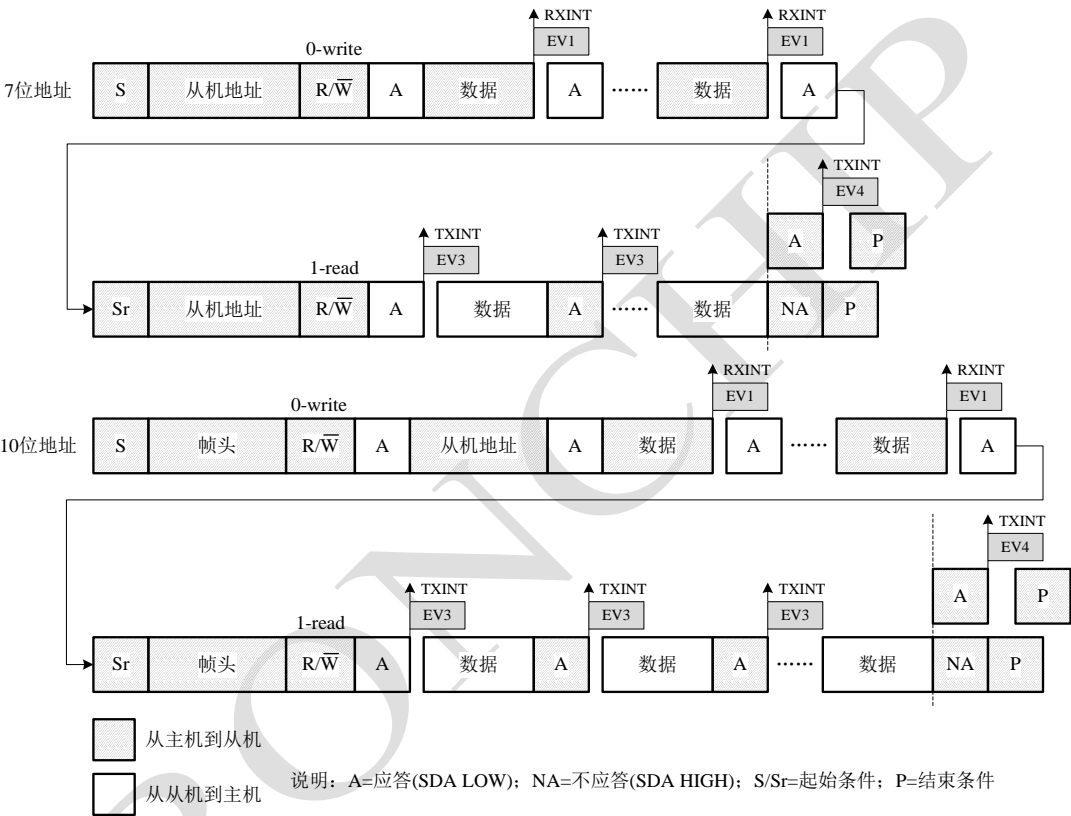


Figure 120 从机混合模式

13.7.2 主机模式

13.7.2.1 主机发送从机地址

配置 I2CCON.MSSEL 为 1，I2C 接口进入主机模式。
根据地址格式，配置 I2CSADDRH 和 I2CSADDRL：

- 7 位地址：配置 I2CSADDRH.S10BEN 为 0，向 I2CSADDRL[6:0]中写入要访问的从机 7 位地址。
- 10 位地址：配置 I2CSADDRH.S10BEN 为 1，向 I2CSADDRH[1:0]和 I2CSADDRL[7:0]中写入要访问的从机 10 位地址。
- 广播模式：向 I2CSADDRL[6:0]中写入“0000000”。

13.7.2.2 主机发送模式

配置寄存器 I2CCON.RW 为 0，配置寄存器 I2CCON.START 为 1，I2C 主机启动主机发送模式。若发送的从机地址为“0000000”，则为广播模式。

若 I2C 总线中存在相应地址的从机，并发出反馈，I2C 模块产生 TXINT 中断，向 I2CWBUF 中写入数据，才能释放 SCL 线。若接收到 ACK 条件并需向从机发送数据，软件参照 EV1 进行配置；若接收到 ACK 条件并需停止发送数据，产生 STOP 条件，则参照 EV2 进行配置；若接收到 NACK 条件，硬件自动产生 STOP 条件。

EV1: (1) 清除 TXINT; (2) 向 I2CWBUF 写入主机向从机发送的数据。

EV2: (1) 清除 TXINT (2) 配置 I2CCON.STOP 为 1，主机产生 STOP 条件。

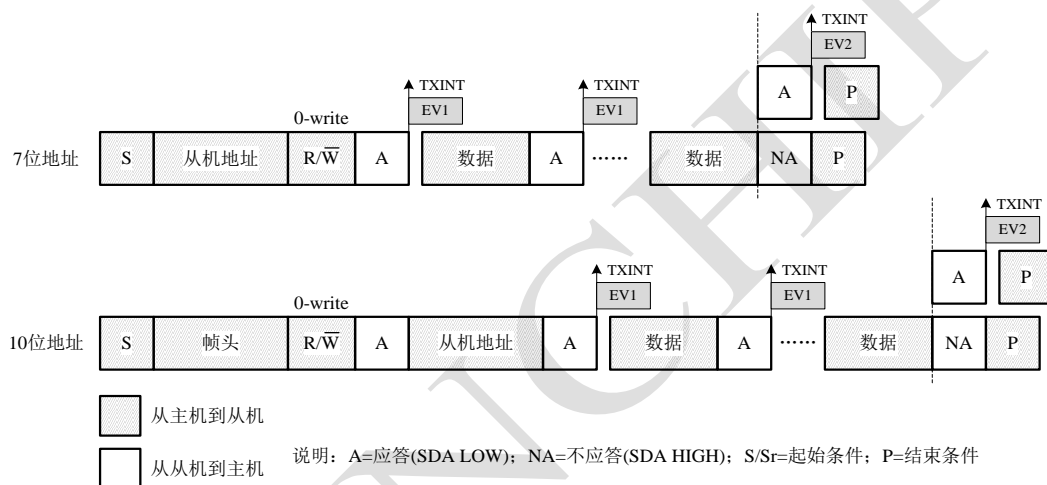


Figure 121 主机发送模式

13.7.2.3 主机接收模式

若为 7 位地址接收模式，配置寄存器 I2CCON.RW 为 1，配置寄存器 I2CCON.START 为 1，I2C 主机启动主机接收模式；若为 10 位地址接收模式，配置寄存器 I2CCON.RW 为 0，配置寄存器 I2CCON.START 为 1，I2C 总线中存在相应地址的从机，并发出反馈，帧头和从机地址发送完成后，会产生 TXINT 中断，软件参照 EV5 进行配置，产生 Sr 条件，并发送帧头信息，I2C 主机启动主机接收模式。

若 I2C 总线中存在相应地址的从机，并发出反馈，I2C 主机开始接收数据，每接收一个数据，都会产生 RXINT 中断，需要读取 I2CRBUF 中数据，才能释放 SCL 线。若需继续接收数据参照 EV3 进行配置；若需产生 STOP 条件，则参照 EV4 进行配置。

EV3: (1) 清除 RXINT; (2) 配置 I2CCON.NACK 为 0，产生 ACK 条件; (3) 读取 I2CRBUF 数据。

EV4: (1) 清除 RXINT; (2) 配置 I2CCON.NACK 为 0 或 1，产生 ACK 或 NACK 条件; (3) 配置 I2CCON.STOP 为 1，产生 STOP 条件; (4) 读取 I2CRBUF 数据。

EV5: (1) 清除 TXINT; (2) 配置 I2CCON.START 为 1; (3) 配置 I2CCON.RW 为 1。

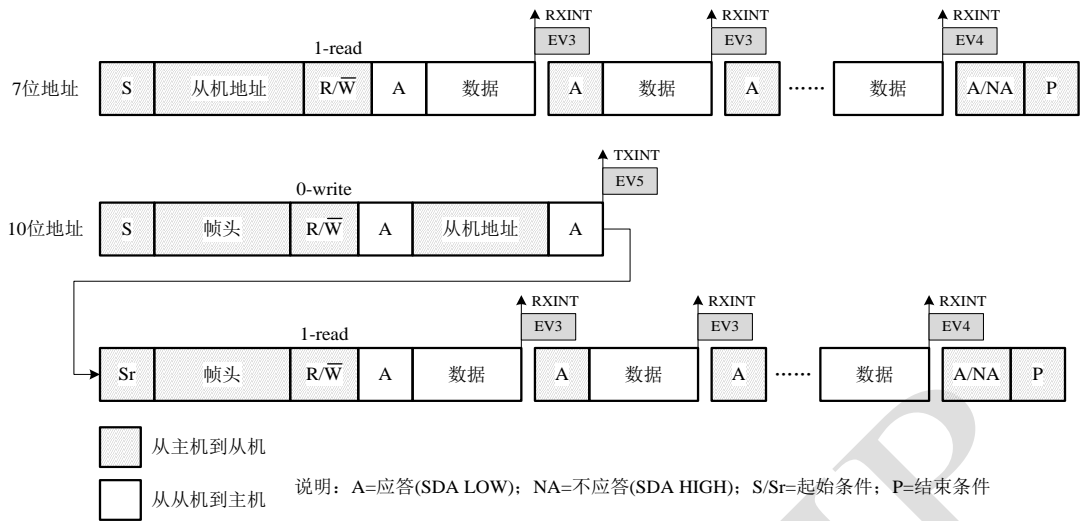


Figure 122 主机接收模式

13.7.2.4 主机混合模式

配置寄存器 I2CCON.RW 为 0，配置寄存器 I2CCON.START 为 1，I2C 主机启动主机发送模式。

若 I2C 总线中存在相应地址的从机，并发出反馈，I2C 模块产生 TXINT 中断，向 I2CWBUF 中写入数据，才能释放 SCL 线。若需向从机发送数据，软件参照 EV1 进行配置。

数据发送完后，参照 EV5 进行配置，产生 Sr（重新启动）条件，使 I2C 主机再次启动进入主机接收模式。

I2C 主机接收到从机反馈，并开始接收数据，每接收一个数据，都会产生 RXINT 中断，需要读取 I2CRBUF 中数据，才能释放 SCL 线。若需继续接收数据参照 EV3 进行配置，若需产生 STOP 条件，则参照 EV4 进行配置。

- EV1: (1) 清除 TXINT; (2) 向 I2CWBUF 写入数据。
- EV5: (1) 清除 TXINT; (2) 配置 I2CCON.START 为 1; (3) 配置 I2CCON.RW 为 1。
- EV3: (1) 清除 RXINT; (2) 配置 I2CCON.NACK 为 0; (3) 读取 I2CRBUF 数据。
- EV4: (1) 清除 RXINT; (2) 配置 I2CCON.NACK 为 0 或 1; (3) 配置 I2CCON.STOP 为 1; (4) 读取 I2CRBUF 数据。

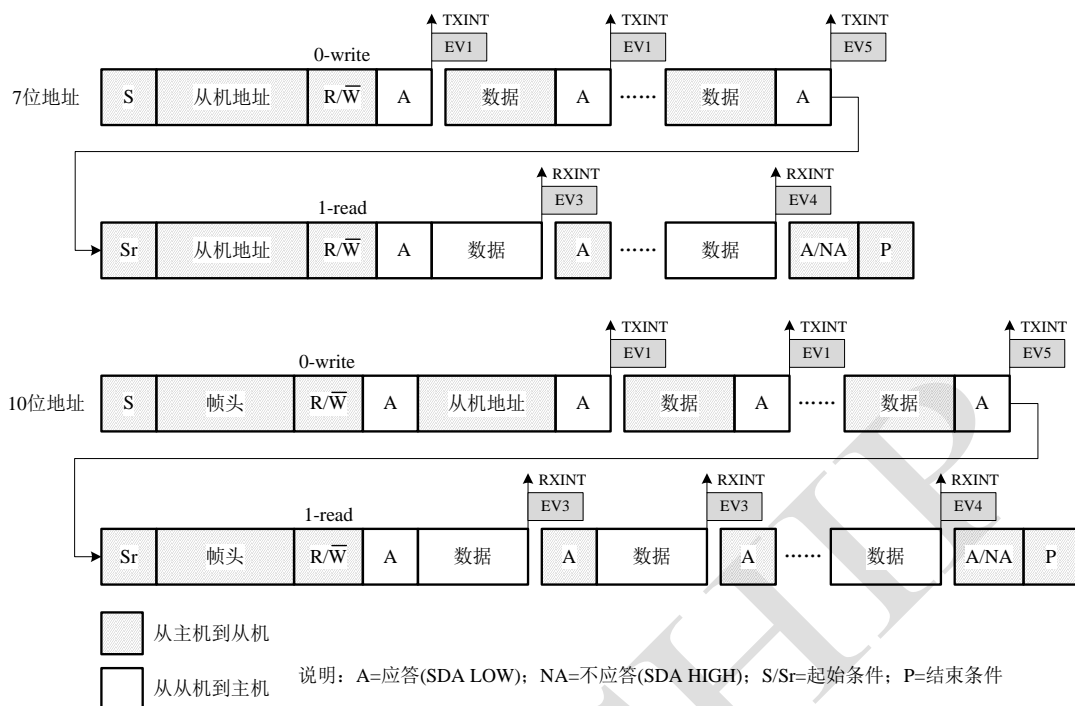


Figure 123 主机混合模式

13.8 I2C 中断

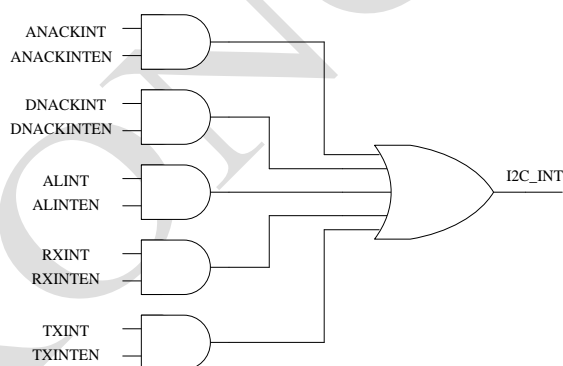


Figure 124 I2C 中断

I2C 有 5 个中断源: 地址 NACK 中断 (ANACKINT)、数据 NACK 中断 (DNACKINT)、仲裁丢失中断 (ALINT)、接收中断 (RXINT) 和发送中断 (TXINT)。

I2C 中断配置及处理:

- 配置 I2CIEN 寄存器使能相关中断源;
- 配置 IEN2. I2CINT 寄存器为 1, 使能 I2C 中断;
- 当产生 I2C 中断后, 软件需要回读 I2CISC 寄存器, 来判断 I2C 中断源, 根据不同中断源, 向 I2CISC 寄存器相应位写 0 清除, 并做相应处理。

13.9 I2C 寄存器

I2C 模块有 14 个寄存器，对应的 SFRPAGE 为 0，具体说明如 Table 193 所示。

Table 193 I2C 寄存器 (SFRPAGE=0x00)

地址	寄存器名	类型	复位值	功能描述
0xE1	I2CCON	W/R	0x00	I2C 控制寄存器
0xE2	I2CSADDRL	W/R	0x00	I2C 从机地址寄存器低 8 位
0xE3	I2CSADDRH	W/R	0x00	I2C 从机地址寄存器高 2 位
0xE4	I2COADDRL	W/R	0x00	I2C 本机地址寄存器低 8 位
0xE5	I2COADDRH	W/R	0x00	I2C 本机地址寄存器高 2 位
0xE6	I2CDIV	W/R	0x02	I2C 工作时钟分频控制寄存器
0xE7	I2CDUTYL	W/R	0x0e	I2C 时钟 SCL 低电平时间配置寄存器
0xE9	I2CDUTYH	W/R	0x0e	I2C 时钟 SCL 高电平时间配置寄存器
0xEA	I2CHOLD	W/R	0x06	I2C 数据 SDA 保持时间配置寄存器
0xEB	I2CWBUF	W/R	0x00	I2C 数据写缓冲寄存器
0xEC	I2CRBUF	R	0x00	I2C 数据读缓冲寄存器
0xED	I2CSTS	R	0x42	I2C 状态寄存器
0xEE	I2CISC	W/R	0x00	I2C 中断标志寄存器
0xEF	I2CIEN	W/R	0x00	I2C 中断使能寄存器

13.9.1 I2C 控制寄存器(I2CCON) [0xE1]

Table 194 I2C 控制寄存器(I2CCON)

参数名	比特位	属性	复位值	描述
MSSEL	0	W/R	0	主机/从机模式选择： 0: 从机模式 1: 主机模式
START	1	W/R	0	产生起始条件，仅在主机模式下有效，当检测到总线上的起始条件后，硬件自动清除该位： 0: 无效 1: 产生起始条件
STOP	2	W/R	0	产生停止条件，仅在主机模式下有效，当检测到总线上的停止条件后，硬件自动清除该位：

参数名	比特位	属性	复位值	描述
				0: 无效 1: 产生停止条件
NACK	3	W/R	0	产生无效应答, 当检测到总线上的起始条件或停止条件后, 硬件自动清除该位: 0: 无效 1: 产生无效应答
RW	4	W/R	0	I2C 总线读写传输控制, 仅在主机模式下有效: 0: 写(发送)操作 1: 读(接收)操作
ADDR0EN	5	W/R	0	广播呼叫使能, 仅在从机模式下有效: 0: 禁止, 不产生应答 1: 使能, 产生应答
LBM	6	W/R	0	在主机模式下, I2C 回环测试模式使能: 0: 不使能 1: 使能
Reserved	7	R	0	保留

13.9.2 I2C 从机地址寄存器低 8 位(I2CSADDRL) [0xE2]

Table 195 I2C 从机地址寄存器低 8 位(I2CSADDRL)

参数名	比特位	属性	复位值	描述
I2CSADDRL	7-0	W/R	0	从机地址: 7 位从机地址模式时, 低 7 位为从机地址; 10 位从机地址模式时, 为从机地址的低 8 位

13.9.3 I2C 从机地址寄存器高 2 位(I2CSADDRH) [0xE3]

Table 196 I2C 从机地址寄存器高 2 位(I2CSADDRH)

参数名	比特位	属性	复位值	描述
I2CSADDRH	1-0	W/R	0	从机地址: 7 位从机地址模式时, 该位无效; 10 位从机地址模式时, 为从机地址的高 2 位

参数名	比特位	属性	复位值	描述
Reserved	6-2	R	0	保留
S10BEN	7	W/R	0	10 位从机地址模式使能： 0: 7 位从机地址模式 1: 10 位从机地址模式

13.9.4 I2C 本机地址寄存器低 8 位(I2COADDRL) [0xE4]

Table 197 I2C 本机地址寄存器低 8 位(I2COADDRL)

参数名	比特位	属性	复位值	描述
I2COADDRL	7-0	W/R	0	本机地址： 7 位本机地址模式时，低 7 位为本机地址； 10 位本机地址模式时，为本机地址的低 8 位

13.9.5 I2C 本机地址寄存器高 2 位(I2COADDRH) [0xE5]

Table 198 I2C 本机地址寄存器高 2 位(I2COADDRH)

参数名	比特位	属性	复位值	描述
I2COADDRH	1-0	W/R	0	本机地址： 7 位本机地址模式时，该位无效； 10 位本机地址模式时，为本机地址的高 2 位
Reserved	6-2	R	0	保留
O10BEN	7	W/R	0	10 位本机地址模式使能： 0: 7 位本机地址模式 1: 10 位本机地址模式

13.9.6 I2C 工作时钟分频控制寄存器(I2CDIV) [0xE6]

Table 199 I2C 工作时钟分频控制寄存器(I2CDIV)

参数名	比特位	属性	复位值	描述
NDIV	7-0	W/R	00000010	I2C 工作时钟分频控制： 00000000: 1 分频 00000001: 1 分频 00000010: 2 分频 00000011: 3 分频

参数名	比特位	属性	复位值	描述
			 11111110: 254 分频 11111111: 255 分频

13.9.7 I2C 时钟 SCL 低电平时间配置寄存器(I2CDUTYL) [0xE7]

Table 200 I2C 时钟 SCL 低电平时间配置寄存器(I2CDUTYL)

参数名	比特位	属性	复位值	描述
NLOW	7-0	W/R	00001110	I2C 时钟 SCL 低电平时间配置: 00000000: $3 * NDIV * T_{i2c_clk}$ 00000001: $3 * NDIV * T_{i2c_clk}$ 00000010: $3 * NDIV * T_{i2c_clk}$ 00000011: $4 * NDIV * T_{i2c_clk}$ 11111110: $255 * NDIV * T_{i2c_clk}$ 11111111: $256 * NDIV * T_{i2c_clk}$

13.9.8 I2C 时钟 SCL 高电平时间配置寄存器(I2CDUTYH) [0xE9]

Table 201 I2C 时钟 SCL 高电平时间配置寄存器(I2CDUTYH)

参数名	比特位	属性	复位值	描述
NHIGH	7-0	W/R	00001110	I2C 时钟 SCL 高电平时间配置: 00000000: $3 * NDIV * T_{i2c_clk}$ 00000001: $3 * NDIV * T_{i2c_clk}$ 00000010: $3 * NDIV * T_{i2c_clk}$ 00000011: $4 * NDIV * T_{i2c_clk}$ 11111110: $255 * NDIV * T_{i2c_clk}$ 11111111: $256 * NDIV * T_{i2c_clk}$

13.9.9 I2C 数据 SDA 保持时间配置寄存器(I2CHOLD) [0xEA]

Table 202 I2C 数据 SDA 保持时间配置寄存器(I2CHOLD)

参数名	比特位	属性	复位值	描述
NHOLD	7-0	W/R	00001110	I2C 数据 SDA 保持时间配置, 即 SDA 在 SCL 拉低后数据的保持时间, 在时序满足的情况下, 尽量小于(NLOW+1)/2:

参数名	比特位	属性	复位值	描述
				00000000: $1 * NDIV * T_{i2c_clk}$ 00000001: $1 * NDIV * T_{i2c_clk}$ 00000010: $2 * NDIV * T_{i2c_clk}$ 00000011: $3 * NDIV * T_{i2c_clk}$ 11111110: $254 * NDIV * T_{i2c_clk}$ 11111111: $255 * NDIV * T_{i2c_clk}$

13.9.10 I2C 数据写缓冲寄存器(I2CWBUF) [0xEB]

Table 203 I2C 数据写缓冲寄存器(I2CWBUF)

参数名	比特位	属性	复位值	描述
I2CWBUF	7-0	W/R	0	I2C 数据写缓冲

13.9.11 I2C 数据读缓冲寄存器(I2CRBUF) [0xEC]

Table 204 I2C 数据读缓冲寄存器(I2CRBUF)

参数名	比特位	属性	复位值	描述
I2CRBUF	7-0	R	0	I2C 数据读缓冲

13.9.12 I2C 状态寄存器(I2CSTS) [0xED]

Table 205 I2C 状态寄存器(I2CSTS)

参数名	比特位	属性	复位值	描述
BUSY	0	R	0	I2C 总线忙状态指示: 0: 总线空闲 1: 总线忙
WBUFS	1	R	1	写缓冲状态指示: 0: 写缓冲满 1: 写缓冲空
RBUFS	2	R	0	读缓冲状态指示: 0: 读缓冲空 1: 读缓冲满
ADDR0	3	R	0	广播呼叫使能且在从机模式的情况下, 指示是否发生广播呼叫: 0: 未发生广播呼叫

参数名	比特位	属性	复位值	描述
				1: 发生广播呼叫
AAS	4	R	0	本机被作为从机寻址标志位, 发生广播呼叫时, 该位置 1: 0: 本机未被寻址 1: 本机被寻址
SDIR	5	R	0	本机被作为从机寻址时, 本机传输方向指示: 0: 接收 1: 发送
SCL	6	R	1	I2C 时钟 SCL 线状态指示: 0: 低电平 1: 高电平
Reserved	7	R	0	保留

13.9.13 I2C 中断标志寄存器(I2CISC) [0xEE]

Table 206 I2C 中断标志寄存器(I2CISC)

参数名	比特位	属性	复位值	描述
ANACKINT	0	W/R	0	地址未被应答中断标志位: <ul style="list-style-type: none"> 主机模式发送从机地址后, 未被应答时, 该位置为 1 需要软件写 0 清除 不支持写 1 软中断
DNACKINT	1	W/R	1	数据未被应答中断标志位: <ul style="list-style-type: none"> 数据发送后, 未被应答时, 该位置为 1 需要软件写 0 清除 不支持写 1 软中断
ALINT	2	W/R	0	丢失仲裁中断标志位: <ul style="list-style-type: none"> 主机模式时, 丢失仲裁后, 该位置为 1 需要软件写 0 清除 不支持写 1 软中断
RXINT	3	W/R	0	数据接收中断标志位: <ul style="list-style-type: none"> 接收到数据后, 该位置为 1

参数名	比特位	属性	复位值	描述
				<ul style="list-style-type: none"> ● 需要软件写 0 清除 ● 不支持写 1 软中断
TXINT	4	W/R	0	数据发送请求中断标志位： <ul style="list-style-type: none"> ● 存在数据发送请求时，该位置为 1 ● 需要软件写 0 清除 ● 不支持写 1 软中断
Reserved	7-5	R	0	保留

13.9.14 I2C 中断使能寄存器(I2CIEN) [0xEF]

Table 207 I2C 中断使能寄存器(I2CIEN)

参数名	比特位	属性	复位值	描述
ANACKINTEN	0	W/R	0	地址未被应答中断使能： 0：不使能 1：使能
DNACKINTEN	1	W/R	1	数据未被应答中断使能： 0：不使能 1：使能
ALINTEN	2	W/R	0	丢失仲裁中断使能： 0：不使能 1：使能
RXINTEN	3	W/R	0	数据接收中断使能： 0：不使能 1：使能
TXINTEN	4	W/R	0	数据发送请求中断使能： 0：不使能 1：使能
Reserved	7-5	R	0	保留

14 串行外设接口 (SPI)

14.1 概述

JMT18F003PLUS 的串行外设接口 (SPI) 具有以下特征:

- 同步串行传输
- 支持 4 线全双工或 3 线半双工传输
- 1~8 位数据帧格式可设置
- 数据 MSB 在前和 LSB 在前可选择
- 支持主机或从机模式
- 时钟极性和相位可配置
- 片选极性可配置
- 可触发发送中断, 接收中断和接收溢出中断
- 两级 8 位发送 FIFO
- 支持回环测试模式
- 主机波特率可配置
- 主机发送最高频率为 $f_{\text{spi_clk}}/2$, 主机接收最高频率为 $f_{\text{spi_clk}}/4$
- 从机发送最高频率为 $f_{\text{spi_clk}}/4$, 从机接收最高频率为 $f_{\text{spi_clk}}/4$

14.2 框图

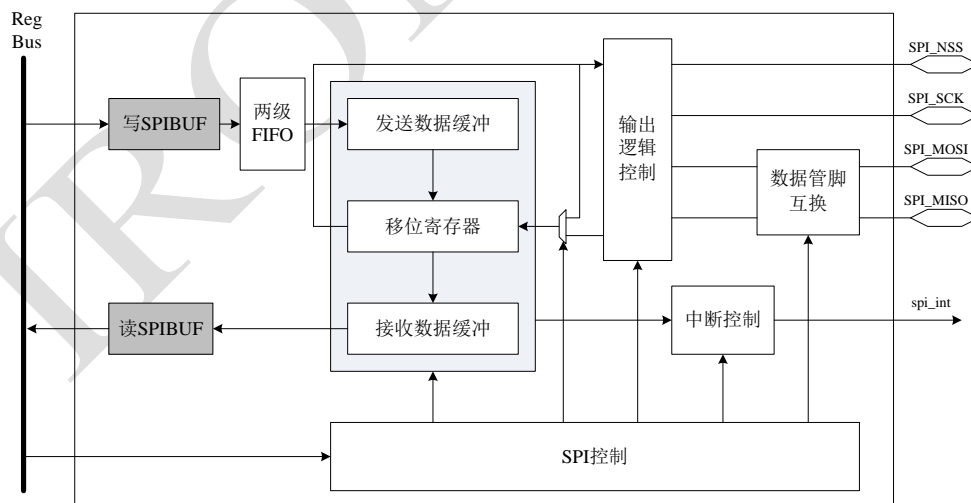


Figure 125 SPI 结构框图

14.3 SPI 管脚配置

SPI 的管脚配置可参见 Table 1 引脚说明表格, 管脚配置说明见 GPIO 寄存器说明。

14.4 SPI 主/从机互连

SPI 的互连方式有 4 种：4 线单从机，4 线多从机，3 线单从机和 3 线多从机。

14.4.1 4 线互连模式

4 线互连模式分为单从机和多从机方式。

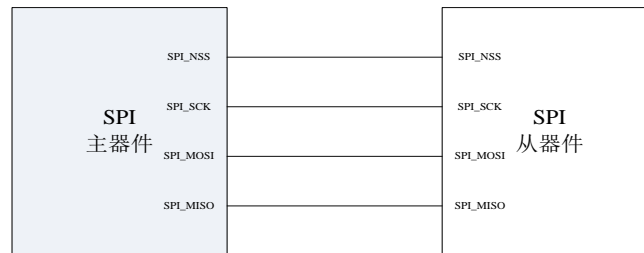


Figure 126 4 线单从机互连

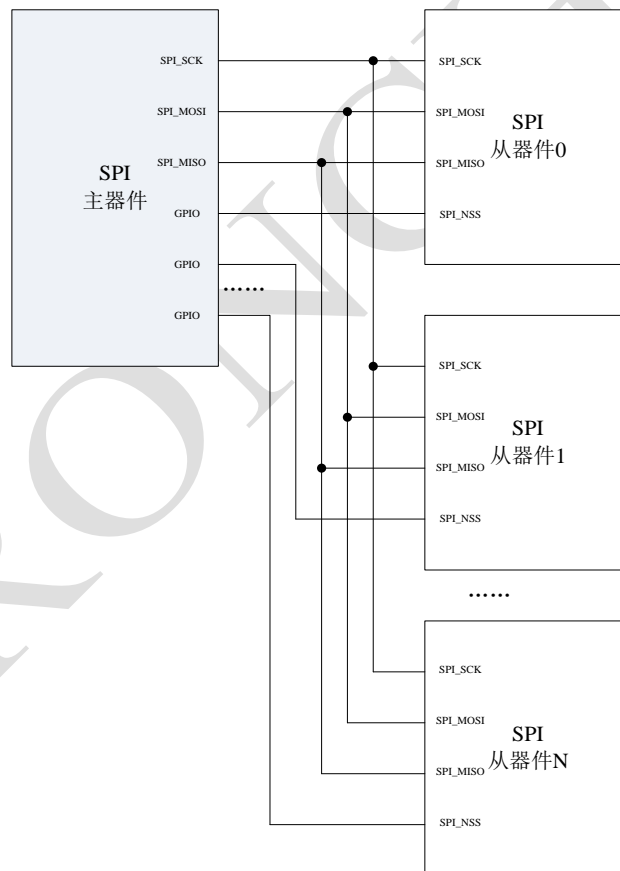


Figure 127 4 线多从机互连

14.4.2 3 线互连模式

3 线互连模式分为单从机和多从机方式。在 3 线互连模式下，寄存器 SPIBR.SDIOSWAP 可以将 SPI 的 SPI_MOSI 和 SPI_MISO 管脚功能进行互换。

JMT18F003PLUS SPI 作为主机时，SPI_MOSI 默认作为主机数据收发管脚，可配置寄存器 SPISTS.SDIOSWAP 为 1，使用 SPI_MISO 作为主机数据收发管脚。

JMT18F003PLUS SPI 作为从机时，SPI_MISO 默认作为从机的数据收发管脚，可配置寄存器 SPISTS.SDIOSWAP 为 1，使用 SPI_MOSI 作为主机数据收发管脚。

如 Figure 128 所示为 3 线默认连接方式：

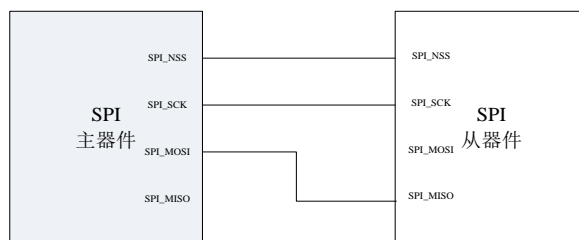


Figure 128 3 线单从机互连

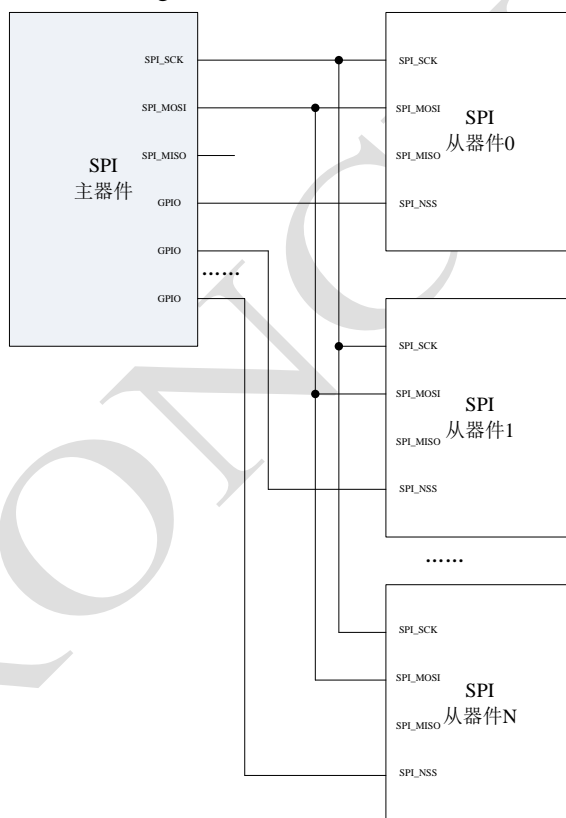


Figure 129 3 线多从机互连

14.5 SPI 时序设置

SPI 的时序配置如下：

- 配置寄存器 SPICON.PHA 选择时钟 SPI_SCK 相位；
- 配置寄存器 SPICON.POL 选择时钟 SPI_SCK 极性；
- 配置寄存器 SPICON.FRP 选择片选 SPI_NSS 极性；
- 配置寄存器 SPICON.LSB 选择数据 SPI_MOSI 和 SPI_MISO 格式。

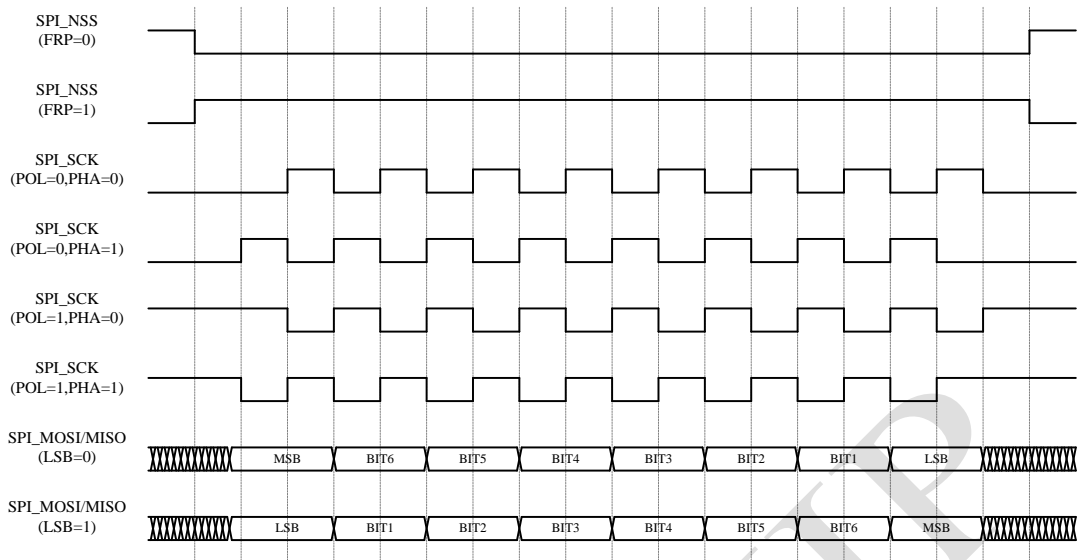


Figure 130 SPI 时序

SPI 从机根据主机提供的 SPI_SCK 时钟发送数据，由于管脚和线路的延时，主机按照默认的相位去采样 SPI_MISO 数据可能来不及，SPI 模块通过配置 SPISPC.SPC 寄存器，可以选择不同的主机采样相位，如图 Figure 131 所示。

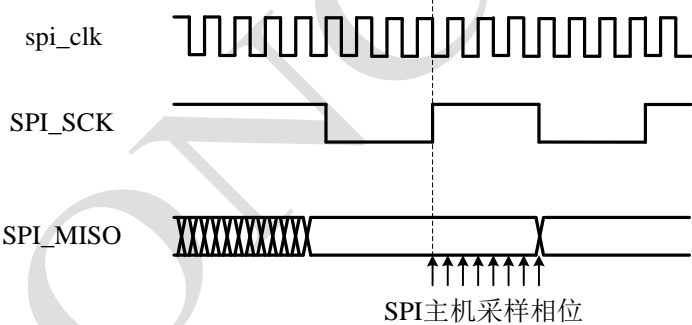


Figure 131 SPI 主机采样相位

14.6 数据发送和接收过程

14.6.1 主机模式

设置寄存器 SPICON.MS 为 0，使 SPI 工作于主机模式。

SPI 通过 SPI_SCK 管脚为整个串行通信提供时钟，数据从 SPI_MOSI 管脚输出，从 SPI_MISO 管脚读入，并通过 SPI_NSS 管脚发送从机 SPI 设备的片选，SPI_NSS 管脚在数据传输之前被置为有效值，在数据传输完毕后被置为无效值。

SPI 总线上的数据传输都由 SPI 主机启动。数据写入 SPIBUF 寄存器来启动在 SPI_MOSI 管脚的数据传输，数据发送顺序可通过 SPICON.LSB 寄存器设置，同时，SPI_MISO 管脚上的接收数据延时 SPC（通过寄存器 SPISPC 配置）个 spi_clk 时钟后，移位到内部移位寄存器的最低比特。当数据的 8 个比特发送完后，接收数据转移到接收缓冲中，可以通过 SPIBUF

寄存器读取。

14.6.1.1 4 线主机模式

4 线主机模式配置：

- 设置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 设置寄存器 SPICON.MS 为 0，使 SPI 工作于主机模式
- 设置寄存器 SPICON.TWI 为 0，选择 4 线 SPI 模式
- 设置寄存器 SPICON.SOD 为 0，使能 SPI 数据输出
- 设置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 设置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 设置寄存器 SPICON.FRP，选择 SPI_NSS 极性
- 设置寄存器 SPIBR，选择 SPI 波特率
- 设置寄存器 SPICON.LSB，选择 SPI 数据比特顺序
- 设置寄存器 SPISPC.SPC，选择主机采样延时
- 设置寄存器 SPISPC.DSS，选择数据帧比特数

4 线主机模式单帧收发流程：

- 向寄存器 SPIBUF 中写入发送数据，启动 SPI 传输
- 等待 SPI 中断或查询 SPIIF.RXINT 标志位置 1 后，传输完成，可通过 SPIBUF 读取接收的从机数据

4 线主机模式连续帧收发流程：

- 判断 SPIIF.TXINT 标志是否为 1，若为 1，则表示发送 FIFO 非满，可向寄存器 SPIBUF 中写入发送数据，启动 SPI 传输
- 等待 SPI 中断或查询 SPIIF.RXINT 标志位置 1 后，表示一帧数据传输完成，可通过 SPIBUF 读取接收的从机数据
- 持续监测 SPIIF.TXINT，直到写入需求的数据个数后停止；并根据 SPIIF.RXINT 回读 SPIBUF 中的从机数据

14.6.1.2 3 线主机模式

14.6.1.2.1 3 线主机模式发送

3 线主机模式发送配置：

- 设置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 设置寄存器 SPICON.MS 为 0，使 SPI 工作于主机模式
- 设置寄存器 SPICON.TWI 为 1，选择 3 线 SPI 模式
- 设置寄存器 SPICON.SOD 为 0，使能 SPI 数据输出
- 设置寄存器 SPICON.POL，选择 SPI_SCK 极性

- 设置寄存器 SPICON.PHA, 选择 SPI_CLK 相位
- 设置寄存器 SPICON.FRP, 选择 SPI_NSS 极性
- 设置寄存器 SPIBR, 选择 SPI 波特率
- 设置寄存器 SPICON.LSB, 选择 SPI 数据比特顺序
- 设置寄存器 SPISPC.DSS, 选择数据帧比特数

3 线主机模式单帧发送流程:

- 向寄存器 SPIBUF 中写入发送数据, 启动 SPI 传输
- 等待 SPI 中断或查询 SPIIF.RX INT 标志位置 1 后, 传输完成

3 线主机模式连续帧发送流程:

- 判断 SPIIF.TXINT 标志是否为 1, 若为 1, 则表示发送 FIFO 非满, 可向寄存器 SPIBUF 中写入发送数据, 启动 SPI 传输
- 等待 SPI 中断或查询 SPIIF.RXINT 标志位置 1 后, 表示一帧数据传输完成
- 持续监测 SPIIF.TXINT, 直到写入需求的数据个数后停止

14.6.1.2.2 3 线主机模式接收

3 线主机模式接收配置:

- 设置寄存器 SPICON.SPIEN 为 1, 使能 SPI 模块
- 设置寄存器 SPICON.MS 为 0, 使 SPI 工作于主机模式
- 设置寄存器 SPICON.TWI 为 1, 选择 3 线 SPI 模式
- 设置寄存器 SPICON.SOD 为 1, 关闭 SPI 数据输出
- 设置寄存器 SPICON.POL, 选择 SPI_SCK 极性
- 设置寄存器 SPICON.PHA, 选择 SPI_CLK 相位
- 设置寄存器 SPICON.FRP, 选择 SPI_NSS 极性
- 设置寄存器 SPIBR, 选择 SPI 波特率
- 设置寄存器 SPICON.LSB, 选择 SPI 数据比特顺序
- 设置寄存器 SPISPC.SPC, 选择主机采样延时
- 设置寄存器 SPISPC.DSS, 选择数据帧比特数

3 线主机模式接收流程:

- 向寄存器 SPIBUF 中写入任意数据, 启动 SPI 传输
- 等待 SPI 中断或查询 SPIIF.RXINT 标志位置 1 后, 传输完成, 可通过 SPIBUF 读取接收的从机数据

14.6.2 从机模式

设置寄存器 SPICON.MS 为 1, 使 SPI 工作于从机模式。

SPI 通过 SPI_SCK 管脚接收串行时钟, 并只在 SPI_NSS 管脚为有效值时, 从 SPI_MOSI

管脚读入数据，并向 SPI_MISO 管脚输出数据。

SPI 从机只能被动的收发数据，不能作为通信的发起者。当检测到 SPI_NSS 片选有效时，SPI 从机开始根据 SPI_SCK 管脚的时钟从 SPI_MOSI 管脚读取数据，移位到内部移位寄存器的最低比特，同时，启动将 SPIBUF 中数据在 SPI_MISO 管脚上的传输，数据发送顺序可通过 SPICON.LSB 寄存器设置。当数据接收完后，接收数据转移到接收缓冲中，可以通过 SPIBUF 寄存器读取。

14.6.2.1 4 线从机模式

4 线从机模式配置：

- 设置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 设置寄存器 SPICON.MS 为 1，使 SPI 工作于从机模式
- 设置寄存器 SPICON.TWI 为 0，选择 4 线 SPI 模式
- 设置寄存器 SPICON.SOD 为 0，使能 SPI 数据输出
- 设置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 设置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 设置寄存器 SPICON.FRP，选择 SPI_NSS 极性
- 设置寄存器 SPICON.LSB，选择 SPI 数据比特顺序
- 设置寄存器 SPISPC.DSS，选择数据帧比特数

4 线从机模式收发流程：

- 向寄存器 SPIBUF 中写入发送数据，硬件将根据 SPI 总线 SPI_NSS 和 SPI_SCK 状态，从 SPI_MOSI 接收数据，向 SPI_MISO 发送数据
- 等待 SPI 中断或查询 SPIIF.RXINT 标志位置 1 后，传输完成，可通过寄存器 SPIBUF 读取接收的主机数据

14.6.2.2 3 线从机模式

14.6.2.2.1 3 线从机模式发送

3 线从机模式发送配置：

- 设置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 设置寄存器 SPICON.MS 为 1，使 SPI 工作于从机模式
- 设置寄存器 SPICON.TWI 为 1，选择 3 线 SPI 模式
- 设置寄存器 SPICON.SOD 为 0，使能 SPI 数据输出
- 设置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 设置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 设置寄存器 SPICON.FRP，选择 SPI_NSS 极性
- 设置寄存器 SPICON.LSB，选择 SPI 数据比特顺序

- 设置寄存器 SPISPC.DSS，选择数据帧比特数

3 线从机模式发送流程：

- 向 SPIBUF 中写入发送数据，硬件将根据 SPI 总线 SPI_NSS 和 SPI_SCK 状态，向 SPI_MISO 发送数据
- 等待 SPI 中断或查询 SPIIF.RXINT 标志位置 1 后，传输完成

14.6.2.2.2 3 线从机模式接收

3 线从机模式接收配置：

- 设置寄存器 SPICON.SPIEN 为 1，使能 SPI 模块
- 设置寄存器 SPICON.MS 为 1，使 SPI 工作于从机模式
- 设置寄存器 SPICON.TWI 为 1，选择 3 线 SPI 模式
- 设置寄存器 SPICON.SOD 为 1，关闭 SPI 数据输出
- 设置寄存器 SPICON.POL，选择 SPI_SCK 极性
- 设置寄存器 SPICON.PHA，选择 SPI_CLK 相位
- 设置寄存器 SPICON.FRP，选择 SPI_NSS 极性
- 设置寄存器 SPICON.LSB，选择 SPI 数据比特顺序
- 设置寄存器 SPISPC.DSS，选择数据帧比特数

3 线从机模式接收流程：

- 硬件根据 SPI 总线 SPI_NSS 和 SPI_SCK 状态，从 SPI_MISO 接收数据
- 等待 SPI 中断或查询 SPIIF.RXINT 标志位置 1 后，传输完成，可通过寄存器 SPIBUF 读取接收的主机数据

14.7 SPI 中断

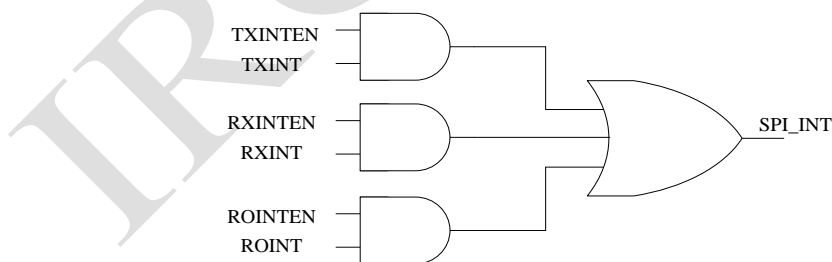


Figure 132 SPI 中断

SPI 有 3 个中断源：发送请求中断（TXINT），接收完成中断（RXINT）和接收溢出中断（ROINT）。其中 TXINT 使能为寄存器 SPIIE.TXINTEN，RXINT 使能为寄存器 SPIIE.RXINTEN，ROINT 使能为寄存器 SPIIE.ROINTEN。

当 SPI FIFO 非满时，TXINT 标志位置 1。

当一帧数据传输完成时，RXINT 标志位置 1。

当 SPI 接收数据缓存发生溢出时，即数据接收后放入 SPIBUF 未被读取，另一个接收数据又将之前接收到的数据覆盖了，ROINT 标志位置 1。

SPI 中断配置及处理：

- 配置 SPIIE 寄存器使能相关中断源；
- 配置 IEN2.SPIINT 寄存器为 1，使能 SPI 中断；
- 当产生 SPI 中断后，软件需要回读 SPIIF 寄存器，来判断 SPI 中断源，根据不同中断源，向 SPIIF 寄存器相应位写 0 清除，并做相应处理。

14.8 SPI 寄存器

SPI 模块有 7 个寄存器，对应的 SFRPAGE 为 0，具体说明如 Table 208 所示。

Table 208 SPI 寄存器（SFRPAGE=0x00）

地址	寄存器名	类型	复位值	功能描述
0xC9	SPICON	W/R	0x02	SPI 控制寄存器
0xCA	SPIIE	W/R	0x00	SPI 中断使能寄存器
0xCB	SPIF	W/R	0x00	SPI 中断标志寄存器
0xCC	SPIBR	W/R	0x01	SPI 波特率控制寄存器
0xCD	SPIBUF	W/R	0x00	SPI 数据缓冲寄存器
0xCE	SPISPC	W/R	0x70	SPI 主机采样控制寄存器
0xCF	SPISTS	W/R	0x04	SPI 状态寄存器

14.8.1 SPI 控制寄存器(SPICON)[0xC9]

Table 209 SPI 控制寄存器(SPICON)

参数名	比特位	属性	复位值	描述
SPIEN	0	W/R	0	SPI 使能控制： 0：不使能 SPI 1：使能 SPI
MS	1	W/R	1	主机/从机模式选择控制： 0：主机模式 1：从机模式
POL	2	W/R	0	SPI 时钟极性控制： 0：SPI_SCK 在非活动期间的极性为低电平 1：SPI_SCK 在非活动期间的极性为高电平
PHA	3	W/R	0	SPI 相位控制： 0：采样时钟第一个跳变出现在每帧第一比

参数名	比特位	属性	复位值	描述
				特传输中间处 1: 采样时钟第一个跳变出现在每帧第一比特传输开始处
FRP	4	W/R	0	SPI 片选信号极性控制: 0: SPI 片选低电平有效 1: SPI 片选高电平有效
TWI	5	W/R	0	SPI 3/4 线模式选择: 0: 4 线模式 1: 3 线模式
LSB	6	W/R	0	SPI 数据 LSB 使能控制: 0: MSB, 先发送数据高位 1: LSB, 先发送数据低位
SOD	7	W/R	0	SPI 输出关闭控制: 0: SPI 输出使能 1: SPI 输出关闭

寄存器 TWI（三线 SPI 模式使能），需要寄存器 SOD 来配合实现三线 SPI：

- 当 SPI 工作于主机模式时，SPI_MOSI 作为输入输出，主机发送时需要配置寄存器 SOD 为 0，主机接收时需要配置寄存器 SOD 为 1；
- 当 SPI 工作于从机模式时，SPI_MISO 作为输入输出，从机发送时需要配置寄存器 SOD 为 0，从机接收时需要配置寄存器 SOD 为 1。

Figure 133 所示为两个 JMT18F003PLUS SPI 工作于 3 线模式下的主从连接关系，当主机发送从机接收，则设置主机 SOD 为 0，从机 SOD 为 1；当主机接收从机发送，则设置主机 SOD 为 1，从机 SOD 为 0。

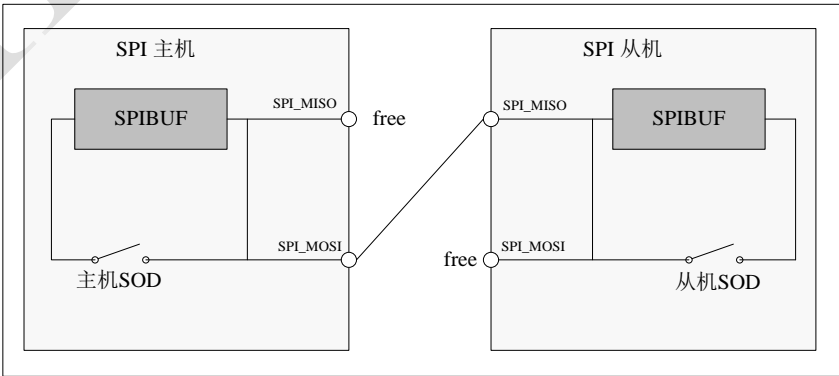


Figure 133 3 线 SPI 连接

14.8.2 SPI 中断使能寄存器(SPIIE)[0xCA]

Table 210 SPI 中断使能寄存器(SPIIE)

参数名	比特位	属性	复位值	描述
TXINTEN	0	W/R	0	SPI 发送请求中断使能: 0: 不使能 1: 使能
RXINTEN	1	W/R	0	SPI 接收完成中断使能: 0: 不使能 1: 使能
ROINTEN	2	W/R	0	SPI 接收溢出中断使能: 0: 不使能 1: 使能
Reserved	7-3	R	0	保留

14.8.3 SPI 中断标志寄存器(SPIIF)[0xCB]

Table 211 SPI 中断标志寄存器(SPIIF)

参数名	比特位	属性	复位值	描述
TXINT	0	W/R	1	SPI 发送请求中断标志位: <ul style="list-style-type: none"> 当 SPI FIFO 非满时, 会产生发送请求中断申请, 该位置为 1 当 SPI FIFO 满时, 该位置为 0 不支持写 1 软中断
RXINT	1	W/R	0	SPI 接收完成中断标志位: <ul style="list-style-type: none"> 当有 SPI 接收完成中断申请时, 该位置为 1 软件写 0 清除 不支持写 1 软中断
ROINT	2	W/R	0	SPI 接收溢出中断标志位: <ul style="list-style-type: none"> 当有 SPI 接收溢出中断申请时, 该位置为 1 软件写 0 清除 不支持写 1 软中断
Reserved	7-3	R	0	保留

14.8.4 SPI 波特率控制寄存器(SPIBR)[0xCC]

Table 212 SPI 波特率控制寄存器(SPIBR)

参数名	比特位	属性	复位值	描述
SPIBR	7-0	W/R	0x01	SPI 波特率控制，计算公式如下： $f_{SPI} = \frac{f_{spi_clk}}{2 * (SPIBR + 1)}$

14.8.5 SPI 数据缓冲寄存器(SPIBUF) [0xCD]

Table 213 SPI 数据缓冲寄存器(SPIBUF)

参数名	比特位	属性	复位值	描述
SPIBUF	7-0	W/R	0	SPI 数据缓冲寄存器： <ul style="list-style-type: none"> 主机模式下，对寄存器 SPIBUF 写操作，则 SPI 将开始向外传输发送缓存数据，从机模式下，则为待发送数据 对寄存器 SPIBUF 读操作，则 SPI 将从接收缓存中读取数据

14.8.6 SPI 主机采样控制寄存器(SPISPC)[0xCE]

Table 214 SPI 主机采样控制寄存器(SPISPC)

参数名	比特位	属性	复位值	描述
SPC	2-0	W/R	0	SPI 主机采样 SPI_MISO 上数据的延时控制： 000: 不延时采样 001: 延时 1 个 spi_clk 采样 010: 延时 2 个 spi_clk 采样 011: 延时 3 个 spi_clk 采样 100: 延时 4 个 spi_clk 采样 101: 延时 5 个 spi_clk 采样 110: 延时 6 个 spi_clk 采样 111: 延时 7 个 spi_clk 采样
Reserved	3	R	0	保留
DSS	6-4	W/R	111	SPI 每帧数据比特数： 000: 1 001: 2 010: 3

参数名	比特位	属性	复位值	描述
				011: 4 100: 5 101: 6 110: 7 111: 8
Reserved	7	R	0	保留

14.8.7 SPI 状态寄存器(SPISTS)[0xCF]

Table 215 SPI 状态寄存器(SPISTS)

参数名	比特位	属性	复位值	描述
FIFOUSED	1-0	R	0	SPI 发送 FIFO 使用深度: 00: FIFO 中没有发送数据 01: FIFO 中有 1 个发送数据 10: FIFO 中有 2 个发送数据 11: 保留
FIFOEMPTY	2	R	1	SPI 发送 FIFO 空状态标志位: 0: 非空 1: 空
FIFOFULL	3	R	0	SPI 发送 FIFO 满状态标志位: 0: 非满 1: 满
Reserved	4	R	0	保留
SPIBUSY	5	R	0	SPI 忙状态标志位: 0: 空闲 1: 工作
SDIOSWAP	6	W/R	0	SPI 数据信号 SPI_MISO 和 SPI_MOSI 功能 互换使能: 0: 不互换 1: 互换
LBM	7	W/R	0	SPI 回环模式使能, 只在 Master 模式下有 效: 0: 不使能

参数名	比特位	属性	复位值	描述
				1: 使能

输入端 ANx	对应模拟输入
AN3	PA3
AN4	PA4
AN5	PA5
AN6	PA6
AN7	PC0
AN8	PB2
AN9	PB3
AN10	PB4
AN11	PGA 输出
VREF	PA7 或内部参考电压

其中 PA0~PA6, PB2~PB4 和 PC0 为芯片模拟输入管脚, PA7 作为外部参考电压管脚 VREF。当这些管脚作为模拟管脚使用时, 需将相应通道的寄存器 AIOEN 置位, 隔断数字对模拟的影响, 具体见 AIOEN0, AIOEN1 寄存器。

15.3 可编程增益放大器 (PGA) 说明

芯片内置 1 个可编程增益放大器 (PGA), PA0, PA1, PA6 和 PB3 经过四选一后连接到 PGA 的正输入端, 其中, PA0 对应的负输入端为 PB2, PA1 对应的负输入端为 PA2, PA6 对应的负输入端为 PA5, PB3 对应的负输入端为 PB4。PGA 具体实现如 Figure 135 所示:

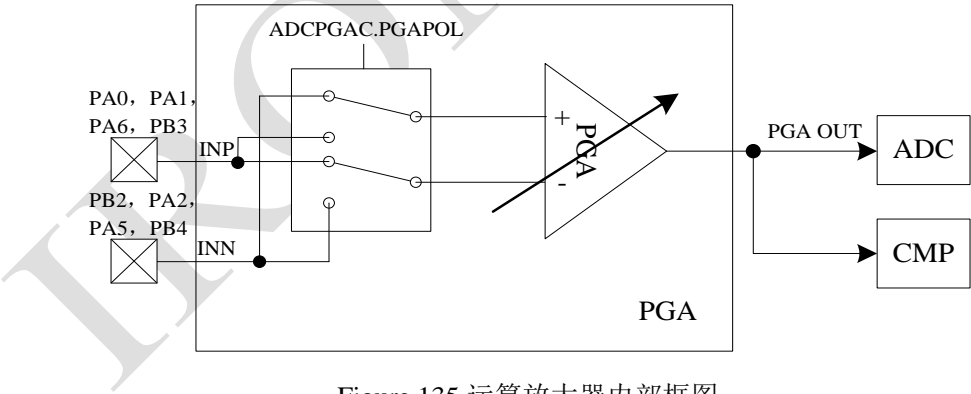


Figure 135 运算放大器内部框图

PGA 支持 1、2、4、8、16 倍放大, 配置寄存器 ADCPGAC.PGAMUL 可选择运放的放大倍数, 具体参见该寄存器说明。

配置寄存器 ADCPGAC.PGAPOL 可切换 PGA 的输入极性, 也即调整 PGA 正负端的输入连接关系, 如 Figure 135 所示:

- 寄存器 ADCPGAC.PGAPOL 为 0 (反相): PGA 正端接 INN, 负端接 INP;
- 寄存器 ADCPGAC.PGAPOL 为 1 (同相): PGA 正端接 INP, 负端接 INN。

PGA 输入电压为 0 时, 对应的 PGA 输出电压为 ADC 参考电压 VREF 的一半, 也即

VREF/2, PGA 输入输出关系为:

- 反相: $V_OUT = VREF/2 - (PGA_MUL_NUM * V_IN)$
- 同相: $V_OUT = VREF/2 + (PGA_MUL_NUM * V_IN)$

其中, V_OUT 为 PGA 的输出电压, $VREF$ 为 ADC 的参考电压, PGA_MUL_NUM 为 PGA 的放大倍数, V_IN 为 PGA 的输入端 INP 和 INN 之间的电压差值 (取值范围 $-300mV \sim VREF$)。

Figure 136 给出了 PGA 配置为同相时, 放大倍数为 1 和 2 场景下, PGA 输入和 PGA 输出的电压关系图。

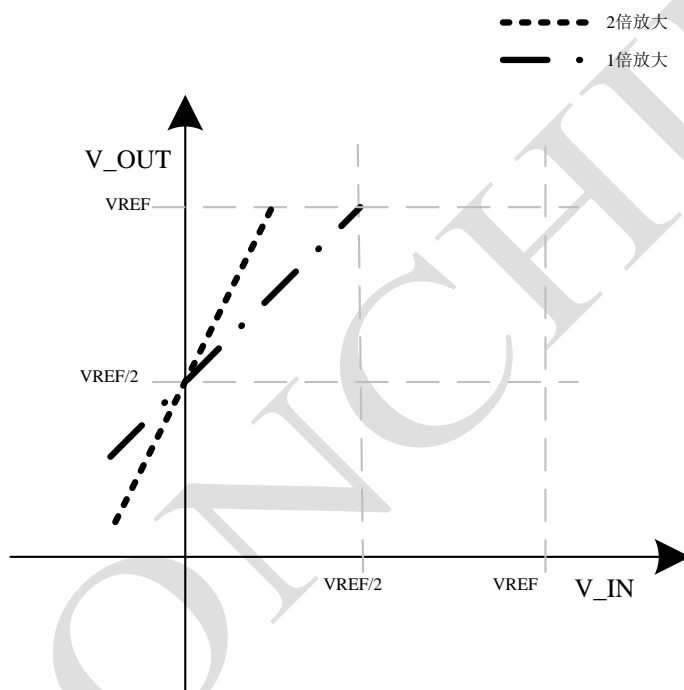


Figure 136 运算放大器同相, 输入输出关系

Figure 137 给出了 PGA 配置为反相时, 放大倍数为 1 和 2 场景下, PGA 输入和 PGA 输出的电压关系图。

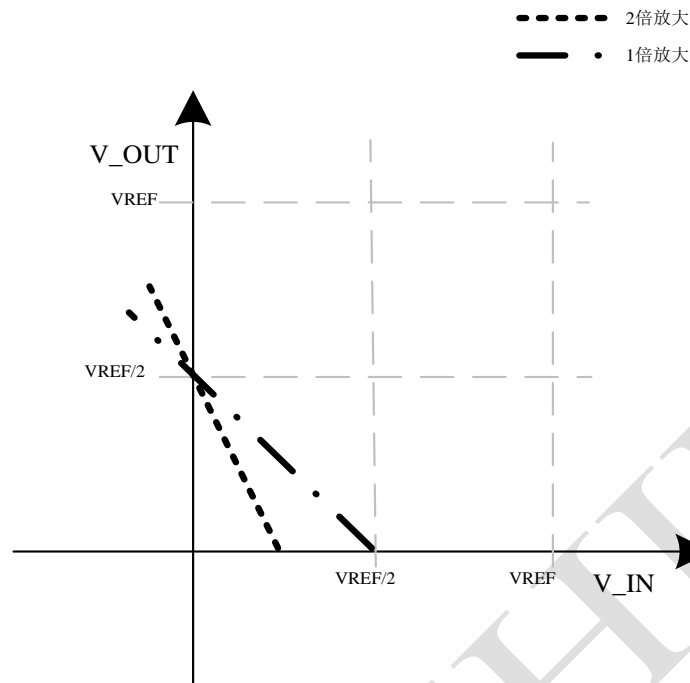


Figure 137 运算放大器反相，输入输出关系

配置寄存器 ADCPGAC.PGAPD，可对 PGA 工作模式进行设置：

- 0：配置 PGA 进入正常工作模式（默认）。
- 1：配置 PGA 进入低功耗模式。

15.4 ADC 参考电压配置

通过配置 ADCCON1.ECVREF_SEL 寄存器，可选择 ADC 的参考电压（VREF）。

ADC 可供选择的参考电压（VREF）有：内部 2.5V 参考电压、芯片模拟供电 VDDA、芯片外部输入参考电压（PA7）。

15.5 ADC 转换功能

芯片的 10 位 A/D 转换器，转换的最大值可达 0x3FF。由于模拟输入最大值等于 VREF 的电压值，因此每一位可表示 $VREF/1024$ 的模拟输入值。下图显示 A/D 转换器模拟输入值和数字输出值之间理想的转换功能。

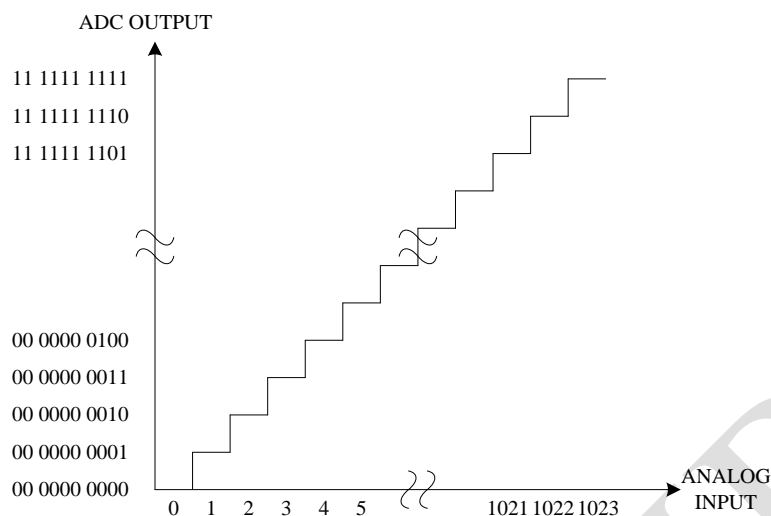


Figure 138 ADC 转换功能

15.6 ADC 转换结果数据格式

ADC 原始转换结果为 10 位数据，该 10 位数据经过数据位扩展至 16 位后放入结果寄存器。扩展后的 16 位数据有 4 种数据格式，可通过配置寄存器 ADCCON1.ADCDF 进行选择，4 种数据格式参见 Figure 139 所示。

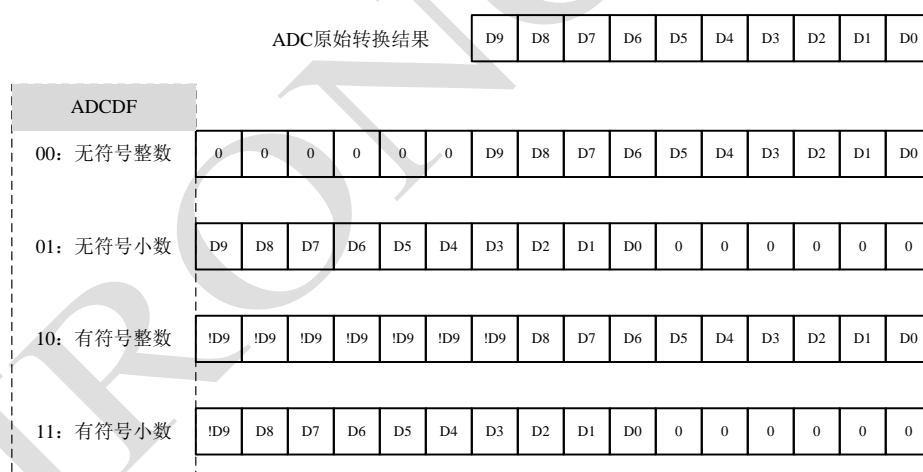


Figure 139 ADC 数据格式

15.7 ADC 转换时序

每次 ADC 转换包括 14 个 `adc_sample_clk` 时钟周期。

单次 ADC 时序如 Figure 140 所示。

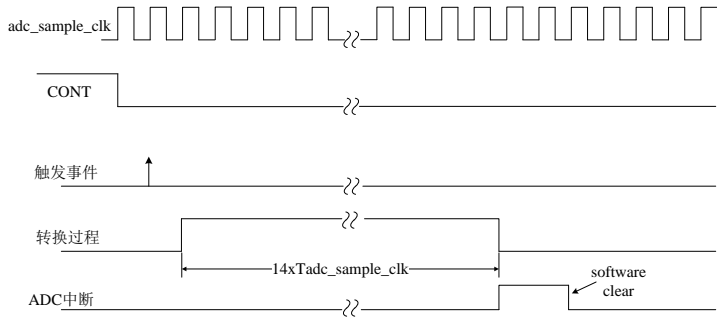


Figure 140 单次 ADC 时序图

连续 ADC 时序如 Figure 141 所示。

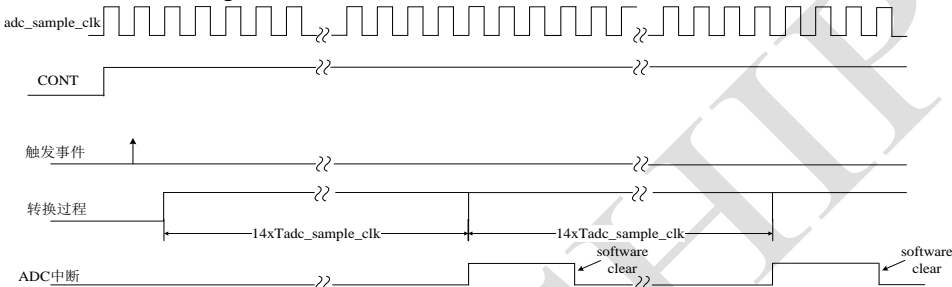


Figure 141 连续 ADC 时序图

15.8 触发事件

触发事件包括软件触发事件和外部触发事件。

15.8.1 软件触发事件

通过配置寄存器 ADCCON0.SWST 为 1，可以立即触发 ADC 转换。

15.8.2 外部触发事件

外部触发事件包括 PWM 触发、管脚边沿触发和 TRGSW 软件触发，配置过程如下：

- 配置寄存器 ADCETC.TRGSEL[3:0]，选择外部触发事件；
- 配置寄存器 ADCETC.TRGPE 和 ADCETC.TRGNE，选择外部触发事件的有效沿；
- 配置寄存器 ADCDLY，选择外部触发事件启动 ADC 采样的延时；
- 配置寄存器 ADCETC.TRGEN 为 1，使能外部触发事件。

外部触发事件启动 ADC 采样延时功能，如 Figure 142 所示。

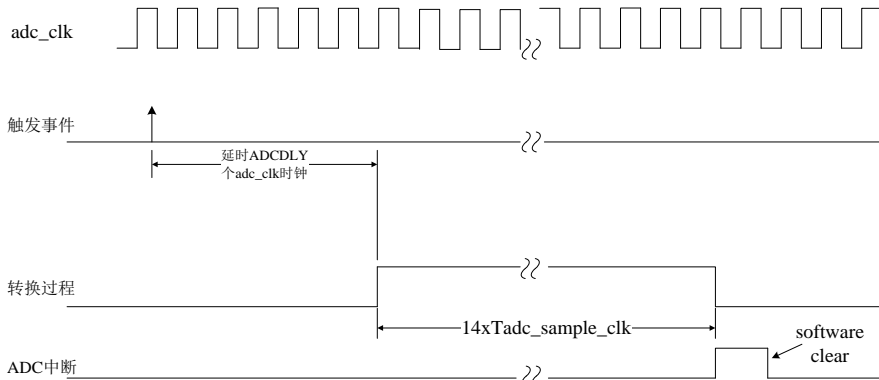


Figure 142 外部触发事件延时功能

Table 217 ADC 外部触发事件

触发源	类型	TRGSEL
ADC_ETR	管脚复用功能	0000
TIM0_OC0	PWM0 内部信号	0001
TIM0_OC1		0010
TIM0_OC2		0011
TIM0_OC3		0100
TIM0_TRGO		0101
TIM1_OC0	PWM1 内部信号	0110
TIM1_TRGO		0111
TRGSW	软件控制	1000

15.9 ADC 操作流程

A/D 转换的操作流程如下：

- 配置 ADCETC 寄存器中的 TRGSEL，选择触发事件
- 配置 ADCETC 寄存器中的 TRGPE 和 TRGNE，选择触发事件有效沿
- 配置 ADCON0 寄存器中的 SEL，选择模拟输入通道
- 配置 ADCON0 寄存器中的 CONT，选择单次或者连续模式
- 配置 ADCON0 寄存器中的 ADCEN，使能 ADC 模块
- 软件触发事件或外部触发事件启动 AD 转换
- 完成 AD 转换后，ADCINT 信号置位，产生 ADC 中断（也可作为状态查询）；同时，AD 转换的结果存放在结果寄存器 ADCRL 和 ADCRH 中

15.10 ADC 寄存器

ADC 模块有 7 个寄存器，对应的 SFRPAGE 为 0，具体说明如 Table 218 所示。ADC 的结果寄存器是 16 位，而 MCU 的数据总线是 8 位，为了便于寄存器读写，将 16 位寄存器拆分为两个 8 位寄存器，即高字节寄存器 ADCRH 和低字节寄存器 ADCRL。读写时需遵循以下原则：先读写高字节寄存器再读写低字节寄存器。

Table 218 JMT51 访问 ADC 寄存器（SFRPAGE=0x00）

地址	寄存器名	类型	复位值	功能描述
0x91	ADCCON0	W/R	0x00	ADC 控制寄存器 0
0x92	ADCCON1	W/R	0x00	ADC 控制寄存器 1
0x93	ADCPGAC	W/R	0x02	放大器控制寄存器

地址	寄存器名	类型	复位值	功能描述
0x94	ADCETC	W/R	0x60	ADC 外部触发事件控制寄存器
0x95	ADCDLY	W/R	0x00	ADC 外部触发延时控制寄存器
0x96	ADCRL	R	0x00	ADC 结果寄存器低 8 位
0x97	ADCRH	R	0x00	ADC 结果寄存器高 8 位

15.10.1 ADC 控制寄存器 0(ADCCON0)[0x91]

Table 219 ADC 控制寄存器 0(ADCCON0)

参数名	比特位	属性	复位值	描述
SEL	3-0	W/R	0	ADC 通道选择: 0000: PA0 0001: PA1 0010: PA2 0011: PA3 0100: PA4 0101: PA5 0110: PA6 0111: PC0 1000: PB2 1001: PB3 1010: PB4 1011: PGA 输出 其他: 保留
ADCEN	4	W/R	0	ADC 使能位: 0: 关闭 ADC 1: 打开 ADC
CONT	5	W/R	0	ADC 模式控制: 0: 单次模式 1: 连续模式
ADCINT	6	W/R	0	ADC 中断标志位: <ul style="list-style-type: none"> 当有 ADC 中断申请时, 该 bit 置为 1 软件配置该 bit 为 1 可产生软中断, 在中断处理程序中需要软件写 0 来清除
SWST	7	W/R	0	ADC 软件触发事件启动:

参数名	比特位	属性	复位值	描述
				0: 不启动 1: 启动 ADC 转换

15.10.2 ADC 控制寄存器 1(ADCCON1)[0x92]

Table 220 ADC 控制寄存器 1(ADCCON1)

参数名	比特位	属性	复位值	描述
ADCDF	1-0	W/R	0	ADC 结果数据格式: 00: 无符号整数 01: 无符号小数 10: 有符号整数 11: 有符号小数
ADCPD	2	W/R	0	软件控制 ADC 低功耗使能: 0: 不使能 1: 使能
PDVREF	3	W/R	0	内部参考电压的低功耗控制: 0: 正常工作模式 1: 低功耗模式 注: 该内部参考电压同时用于 ADC、PGA 和模拟比较器, 需要在以上模块均不工作时才能配置内部参考电压进入低功耗模式
ECVREF_SEL	5-4	W/R	0	ADC 和 PGA 的参考电压选择: 00: 内部 2.5V 参考电压 01: 模拟供电 VDDA 1x: 芯片外部输入参考电压 (PA7 管脚)
ADREFTRIM	6	W/R	0	内部参考电压校准使能: 0: 不使能 1: 使能 注: 仅用于出厂前校准
Reserved	7	R	0	保留

15.10.3 放大器控制寄存器(ADCPGAC)[0x93]

Table 221 放大器控制寄存器(ADCPGAC)

参数名	比特位	属性	复位值	描述
-----	-----	----	-----	----

参数名	比特位	属性	复位值	描述
PGAMUL	2-0	W/R	010	PGA 运放倍数选择（软件调零）： 001: 1 倍 010: 2 倍 011: 4 倍 100: 8 倍 101: 16 倍 其它: 保留
PGAPOL	3	W/R	0	PGA 极性寄存器： 0: 反相 1: 同相
PGAMUX	5-4	W/R	0	PGA 输入引脚选择： 00: INP—PA0; INN—PB2 01: INP—PA1; INN—PA2 10: INP—PA6; INN—PA5 11: INP—PB3; INN—PB4
PGATRIM	6	W/R	0	PGA 校准使能： 0: 不使能 1: 使能 注: 仅用于出厂前校准
PGAPD	7	W/R	0	PGA 的低功耗控制： 0: 正常工作模式 1: 低功耗模式

15.10.4ADC 外部触发事件控制寄存器(ADCETC)[0x94]

Table 222ADC 外部触发事件控制寄存器(ADCETC)

参数名	比特位	属性	复位值	描述
TRGSEL	3-0	W/R	0	外部触发事件选择： 0000: ADC_ETR 0001: TIM0_OC0 0010: TIM0_OC1 0011: TIM0_OC2 0100: TIM0_OC3 0101: TIM0_TRGO

参数名	比特位	属性	复位值	描述
				0110: TIM1_OC0 0111: TIM1_TRGO 1000: TRGSW 其他: 保留
TRGEN	4	W/R	0	外部触发事件使能: 0: 不使能 1: 使能
TRGPE	5	W/R	1	外部触发事件上升沿使能: 0: 不使能 1: 使能
TRGNE	6	W/R	1	外部触发事件下降沿使能: 0: 不使能 1: 使能
TRGSW	7	W/R	0	配置 TRGSW 为 1 作为外部触发事件: 0: 不启动 ADC 转换 1: 启动 ADC 转换 注: 该启动仅在 TRGSEL 选中 TRGSW 为外部触发事件, 且 TRGEN 有效时生效

15.10.5 ADC 外部触发延时控制寄存器(ADCDLY)[0x95]

Table 223 ADC 外部触发延时控制寄存器(ADCDLY)

参数名	比特位	属性	复位值	描述
ADCDLY	7-0	W/R	0	外部触发事件启动 ADC 采样延时 adc_clk 周期个数, 当延时时间大于外部触发事件周期时间时, 外部触发事件将不起作用。

15.10.6 ADC 结果寄存器低 8 位(ADCRL)[0x96]

Table 224 ADC 结果寄存器低 8 位(ADCRL)

参数名	比特位	属性	复位值	描述
ADCRL	7-0	R	0	ADC 结果低 8 位

15.10.7 ADC 结果寄存器高 8 位(ADCRH)[0x97]

Table 225 ADC 结果寄存器高 8 位(ADCRH)

参数名	比特位	属性	复位值	描述
ADCRH	7-0	R	0	ADC 结果高 8 位

16 模拟比较器（CMP）

16.1 模拟比较器概述

JMT18F003PLUS 内置 1 个模拟比较器（CMP），该比较器工作原理为输入电压和参考电压进行比较，若选定比较器输入源的电压值大于选定参考电压的电压值，则比较器输出 1，反之则输出 0。

模拟比较器输出结果可以用于 PWM 刹车、OC/OD 控制、低压检测等功能。

模拟比较器参考电压来自于参考电压 PA7/VREF 管脚或内部参考电压。

模拟比较器的输入可在芯片模拟供电 VDDA、芯片管脚 PA3 和 PGA 输出之间进行选择。

模拟比较器的比较输出，经过滤波防毛刺处理之后，送至比较结果寄存器，软件可以通过回读结果寄存器获取比较的结果。

模拟比较器的框图如 Figure 143 所示，下面具体阐述模拟比较器参考电压选择、比较器输入源选择、防毛刺滤波：

- 比较器输入源选择：配置寄存器 CMPSEL，选择比较器输入源：
 - 00：选择芯片模拟供电 VDDA 作为该比较器输入（默认）。
 - 01：选择芯片管脚 PA3 作为比较器输入。
 - 1x：选择 PGA 输出作为比较器输入。
 - 选择芯片模拟供电 VDDA 作为比较器输入时，比较器输入端电压可通过内部 DAC 对 VDDA 分压得到。通过配置 VDDDAC 寄存器，可修改比较器输入端电压大小，具体大小为： $(V_{VDDA}/64) \times (VDDDAC+1)$ ，其中 V_{VDDA} 为芯片模拟供电的电压值，VDDDAC 为软件配置该寄存器的数值。
 - 若选择 PA3 作为比较器输入，需要配置管脚 PA3 为模拟输入模式，具体配置参见 GPIO 章节说明。

- 配置参考电压大小：通过配置寄存器 CMPREFDAC，可以修改比较器的参考电压大小，具体大小为：

$$(V_{REF}/64) \times (CMPREFDAC+1)$$

其中 V_{REF} 为参考电压源的电压值，CMPREFDAC 为软件配置寄存器的数值。

- 比较器输出滤波防毛刺：
 - 硬件对模拟比较器的比较输出进行了滤波防毛刺处理；
 - 配置比较器滤波时钟控制寄存器 CMP_CLK_CTRL，选择滤波时钟频率；
 - 配置比较器滤波选择寄存器 CMPFT，选择滤波的周期数。

模拟比较器比较结果，滤波之后，做如下处理：

- 配置极性选择寄存器 BKSEL，选择合适的信号作为 PWM 的比较器刹车信号，

控制 PWM 的输出。

- 配置极性选择寄存器 OCSEL，再配置 MOS 管选型寄存器 OCMOSEL，选择合适的信号作为 OC 控制信号，可用于板级硬件保护。
- 配置极性选择寄存器 ODSEL，再配置 MOS 管选型寄存器 ODMOSEL，选择合适的信号作为 OD 控制信号，可用于板级硬件保护。
- 输出结果直接用于 LVR 低压检测，若 LVR 低压复位使能，则产生低压复位信号复位全芯片。

另外，比较器的比较结果送至结果寄存器 CMPFLAG，软件可通过回读该结果寄存器，获取比较的结果。

如果中断被使能了（寄存器 IEN1.CMPINT=1），则硬件按照以下规则产生中断，配置寄存器 CMPINT 为 0 可清除比较器中断：

- 比较器比较结果经滤波之后，和上次的结果不一致，即产生中断；
- 软件可通过查询中断类型寄存器 CMPINTTYPE，获取该中断为比较器结果从 1 变为 0 的中断，还是从 0 变为 1 的中断。

模拟比较器有两种工作模式，正常工作模式（Normal）和低功耗模式（power down），通过配置寄存器 CMPEN.CMPPD，可以选择模拟比较器的工作模式。

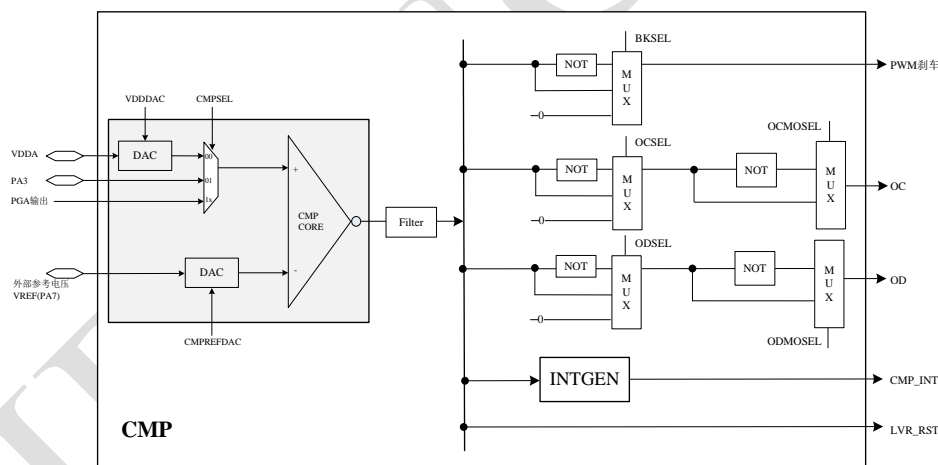


Figure 143 比较器 CMP 内部框图

16.2 低压检测

通过模拟比较器，可实现低压检测功能（LVR），该功能用于监视芯片供电，若芯片供电电压低于设定的值可上报一个中断或产生全芯片复位信号。模块实现如 Figure 144 所示。

如果芯片供电满足如下关系，也即芯片供电电压低于设定的门限值，则会产生 LVR 低压检测中断或者低压检测复位：

$$V_{VDDA} * (VDDDAC+1)/64 < V_{REF} * (CMPREFDAC+1)/64$$

其中：V_{VDDA} 为芯片供电电压，VDDDAC 为 V_{VDDA} 的分压 DAC 的寄存器值，V_{REF} 为

模拟比较器参考电压的电压值，CMPREFDAC 为比较器参考电压分压 DAC 的寄存器值。

假设比较器的参考电压为 2.5V，LVR 低压检测电平为 2.75V。则配置的寄存器 VDDDAC 和 CMPREFDAC 需要满足如下关系，才能实现芯片供电小于 2.75V 时产生低压检测中断或复位：

$$2.75 * (VDDDAC+1)/64 \geq 2.5 * (CMPREFDAC +1)/64$$

可配置 VDDDAC 为 9，CMPREFDAC 配置为 10。

比较器作为 LVR 功能的配置包括：

- 根据比较器小节 16.1 描述，选择比较器输入源为芯片模拟供电 VDDA、选择参考电压、配置两个 DAC 的值、配置滤波周期数。
- 若仅需要监控全芯片供电电压是否低于门限电压，可配置比较器中断使能，中断使能之后，若电压低于监控的门限电压，则产生中断。
- 当全芯片电压低于门限电压时，若需要产生全芯片复位信号，配置寄存器 LVRRSTEN，启动 LVR 复位功能。

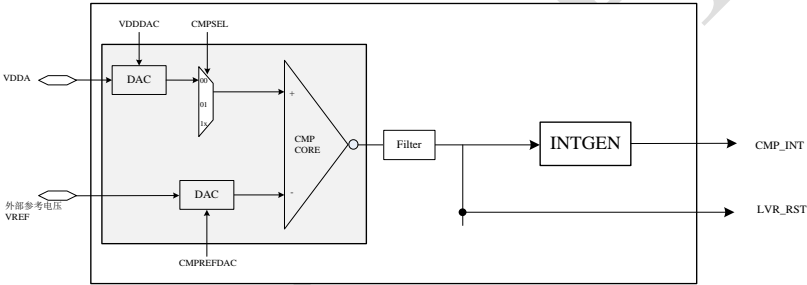


Figure 144 LVR 实现框图

16.3 模拟比较器寄存器

模拟比较器（CMP）共有 9 个寄存器。对应的 SFR PAGE 为 0，具体列表如 Table 226 所示。

Table 226 模拟比较器寄存器（SFRPAGE=0x00）

地址	寄存器名	类型	复位值	功能描述
0xB9	CMPEN	W/R	0x01	模拟比较器使能寄存器
0xBA	CMPFT	W/R	0x03	模拟比较器滤波选择寄存器
0xBB	CMPVDD	W/R	0x20	模拟比较器 VDD 分压寄存器
0xBC	CMPREF	W/R	0x12	模拟比较器参考电压选择寄存器
0xBD	CMPINT	W/R	0x00	模拟比较器中断寄存器
0xBE	CMPFLAG	W/R	0x00	模拟比较器比较结果寄存器
0xC1	OCPRSEL	W/R	0x03	OC 控制寄存器
0xC2	ODPRSEL	W/R	0x03	OD 控制寄存器

地址	寄存器名	类型	复位值	功能描述
0xC3	BKPRSEL	W/R	0x03	PWM 刹车控制寄存器

16.3.1 模拟比较器使能寄存器(CMPEN) [0xB9]

Table 227 模拟比较器使能寄存器(CMPEN)

参数名	比特位	属性	复位值	描述
CMPPD	0	W/R	1	模拟比较器工作模式： 0：正常工作模式 1：低功耗模式
CMPSEL	2-1	W/R	0	模拟比较器输入源选择： 00：芯片供电 VDDA 01：管脚 PA3 10/11：PGA 输出
Reserved	7-3	R	0	保留

16.3.2 模拟比较器滤波选择寄存器(CMPFT) [0xBA]

Table 228 模拟比较器滤波选择寄存器(CMPFT)

参数名	比特位	属性	复位值	描述
CMPFT	1-0	W/R	11	模拟比较器输出滤波周期： 00：不滤波 01：1 个滤波时钟周期 10：2 个滤波时钟周期 11：3 个滤波时钟周期
Reserved	7-2	R	0	保留

16.3.3 模拟比较器 VDD 分压寄存器(CMPVDD) [0xBB]

Table 229 模拟比较器 VDD 分压寄存器(CMPVDD)

参数名	比特位	属性	复位值	描述
VDDDAC	5-0	W/R	0x20	VDD 的 DAC 数字输入码。VDD 经过该 DAC 分压之后输出至比较器输入端，比较器输入端电压为： $(V_{VDD}/64) \times (VDDDAC+1)$ 其中 V_{VDD} 为芯片供电电压值
Reserved	7-6	R	0	保留

16.3.4 模拟比较器参考电压选择寄存器(CMPREF) [0xBC]

Table 230 模拟比较器参考电压选择寄存器(CMPREF)

参数名	比特位	属性	复位值	描述
CMPREFDAC	5-0	W/R	0x12	参考电压的 DAC 数字输入码。参考电压经过该 DAC 分压之后输出至比较器参考电压输入端，比较器参考电压输入端电压为： $(V_{REF}/64) \times (CMPREFDAC+1)$ 其中 V_{REF} 为参考电压源的电压值
CMPREFSEL	6	W/R	1	比较器参考电压选择寄存器： 0：外部参考电压 1：内部参考电压
Reserved	7	R	0	保留

16.3.5 模拟比较器中断寄存器(CMPINT) [0xBD]

Table 231 模拟比较器中断寄存器(CMPINT)

参数名	比特位	属性	复位值	描述
CMPINT	0	W/R	0	模拟比较器中断标志位： <ul style="list-style-type: none"> 若比较器输出结果和之前结果不一致则产生中断 软件配置该位为 1 可产生软中断，在中断处理程序中需要软件写 0 来清除
Reserved	7-1	R	0	保留

16.3.6 模拟比较器比较结果寄存器(CMPFLAG) [0xBE]

Table 232 模拟比较器比较结果寄存器(CMPFLAG)

参数名	比特位	属性	复位值	描述
CMPINTTYPE	0	W/R	0	比较器中断类型寄存器： 0：比较器结果从 1 变为 0 中断 1：比较器结果从 0 变为 1 中断
CMPOUT	1	R	0	模拟比较器输出结果： 0：比较器输入电压小于参考电压 1：比较器输入电压大于参考电压

参数名	比特位	属性	复位值	描述
Reserved	7-3	R	0	保留

16.3.7 OC 控制寄存器(OCPRSEL) [0xC1]

Table 233 OC 控制寄存器(OCPRSEL)

参数名	比特位	属性	复位值	描述
OCSEL	1-0	W/R	11	OC 基准信号极性选择: 00: 滤波之后结果取反 01: 滤波之后直通 其他: 选取 0 作为基准信号
OCMOSEL	2	W/R	0	OC 控制信号 MOS 类型选择: 0: 基准信号取反产生 OC 信号 1: 基准信号作为 OC 信号
Reserved	7-3	R	0	保留

16.3.8 OD 控制寄存器(ODPRSEL) [0xC2]

Table 234 OD 控制寄存器(ODPRSEL)

参数名	比特位	属性	复位值	描述
ODSEL	1-0	W/R	11	OD 基准信号极性选择: 00: 滤波之后结果取反 01: 滤波之后直通 其他: 选取 0 作为基准信号
ODMOSEL	2	W/R	0	OD 控制信号 MOS 类型选择: 0: 基准信号取反产生 OD 信号 1: 基准信号作为 OD 信号
Reserved	7-3	R	0	保留

16.3.9 PWM 刹车控制寄存器(BKPRSEL) [0xC3]

Table 235 PWM 刹车控制寄存器(BKPRSEL)

参数名	比特位	属性	复位值	描述
BKSEL	1-0	W/R	11	PWM 刹车基准信号极性选择: 00: 滤波之后结果取反 01: 滤波之后直通 其他: 选取 0 作为基准信号

参数名	比特位	属性	复位值	描述
Reserved	7-2	R	0	保留

17 通用输入/输出 (GPIO)

GPIO 实现的主要功能有：

- 管脚的模拟/数字选择、输入输出选择、开漏、驱动强度、上下拉等设置
- 管脚功能复用
- 管脚滤波
- 外部中断合并

正常情况下，GPIO 模块的最高输入时钟为 24.576MHz。

17.1 管脚设置

数字 I/O 实现框图如 Figure 145 所示，数字、模拟复用 I/O 实现框图如 Figure 146 所示。

为防止数字 I/O 信号对模拟 I/O 信号的影响，当数字、模拟复用 I/O 当做模拟 I/O 使用时，需将相应管脚的寄存器 AIOEN 置位。

输入使能 (IE) 由硬件自动控制：

- 数字 I/O 的 IE 固定接 1，使能输入
- 数字、模拟复用 I/O 用作模拟 I/O 时，IE 接 0，关断输入
- 数字、模拟复用 I/O 用作数字 I/O 时，IE 接 1，使能输入

管脚信号说明：

- PU：弱上拉使能
- PD：弱下拉使能
- OD：开漏输出模式使能
- IE：输入使能
- OE：输出使能
- A：数字输出 Buffer 输入
- Y：数字输入 Buffer 输出
- YA：模拟信号管脚
- SR：转换速率快慢选择
- DR：驱动能力强弱选择

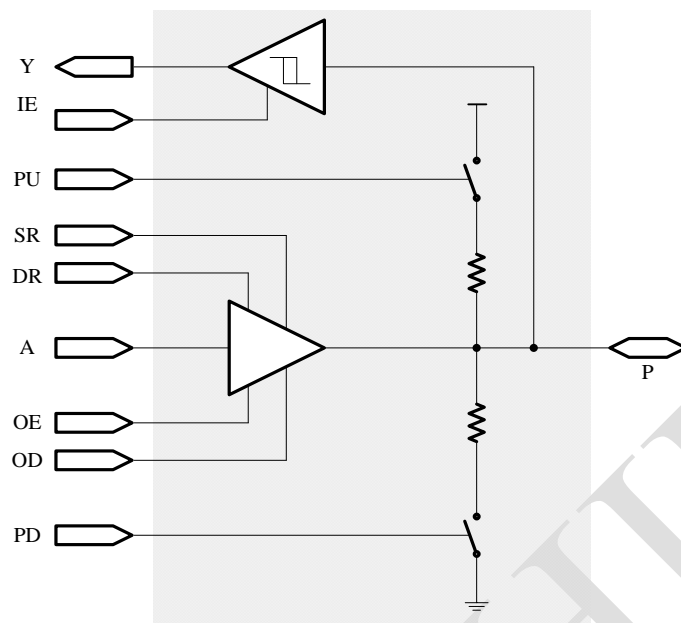


Figure 145 数字 I/O 实现框图

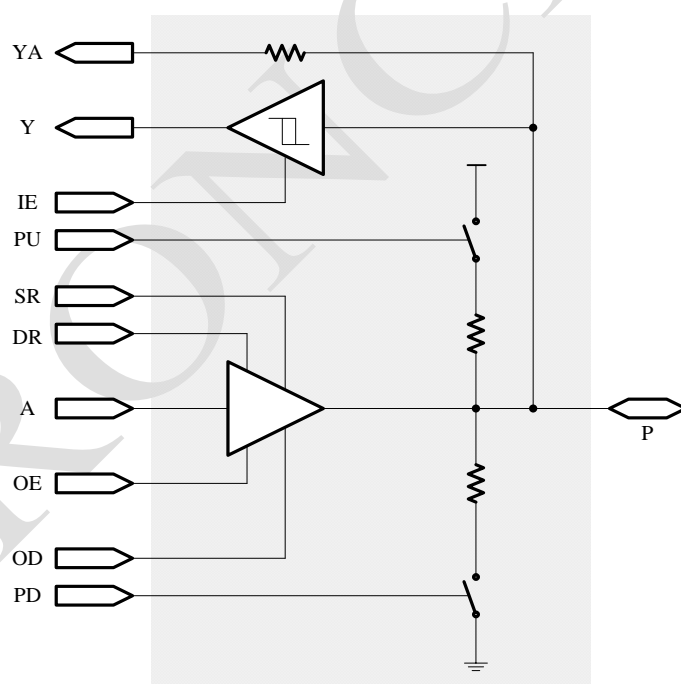


Figure 146 数字/模拟混合 I/O 实现框图

管脚真值表如 Table 236 所示，其中“W”为任意值，“Z”为高阻态，“X”为不确定态：

Table 236 管脚真值表

I/O	输入引脚								输出引脚	
	OE	IE	PU/PD	A	OD	SR	DR	P/YA	P/YA	Y
输入模式	0	1	W	W	W	W	W	1	1	1
	0	1	W	W	W	W	W	0	0	0
	0	1	1	W	W	W	W	-	H(PU) L(PD)	1(PU) 0(PD)
	0	1	0	W	W	W	W	-	Z	X
	0	0	W	W	W	W	W	1	1	0
	0	0	W	W	W	W	W	0	0	0
	0	0	1	W	W	W	W	-	H(PU) L(PD)	0
	0	0	0	W	W	W	W	-	Z	0
输出模式	1	0	W	1	0	W	W	-	1	0
	1	0	W	0	0	W	W	-	0	0
	1	0	W	0	1	W	W	-	0	0
	1	0	1	1	1	W	W	-	H(PU) L(PD)	0
	1	0	0	1	1	W	W	-	Z	0
	1	1	W	1	0	W	W	-	1	1
	1	1	W	0	0	W	W	-	0	0
	1	1	W	0	1	W	W	-	0	0
	1	1	1	1	1	W	W	-	H(PU) L(PD)	1(PU) 0(PD)
	1	1	0	1	1	W	W	-	Z	X

上升/下降转变时间：

- 上升转变时间 T_r ：信号从逻辑 0 变为逻辑 1 所需时间，逻辑 0 标准为小于供电电压的 10%，逻辑 1 的标准为大于供电电压的 90%。
- 下降转变时间 T_f ：信号从逻辑 1 变为逻辑 0 所需时间，逻辑 0 标准为小于供电电压的 10%，逻辑 1 的标准为大于供电电压的 90%。

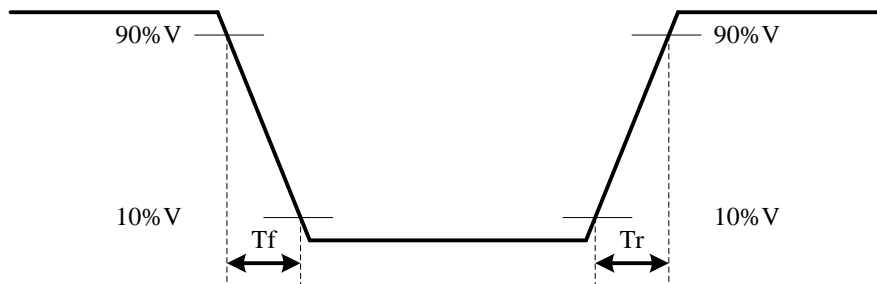


Figure 147 上升/下降转变时间示意图

传播延迟:

- 上升传播延迟 T_{dr} : 上升信号从 PAD 的 P 端输入到 Y 端传输所需时间
- 下降传播延迟 T_{df} : 下降信号从 PAD 的 P 端输入到 Y 端传输所需时间

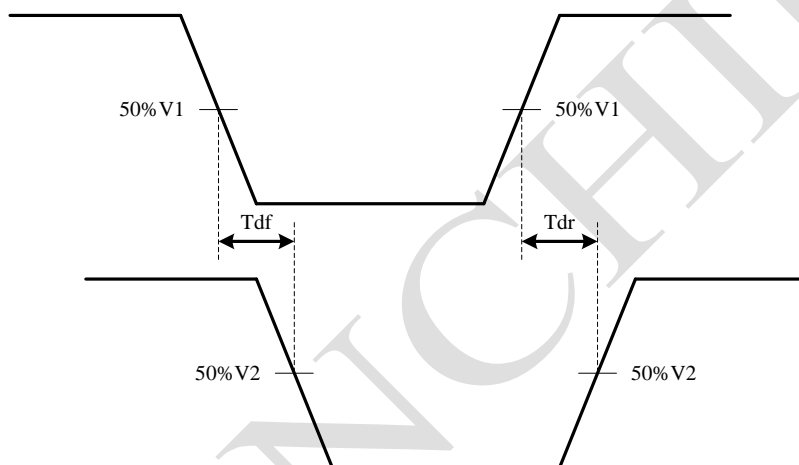


Figure 148 传播延迟示意图

转换速率 SR 对上升/下降转变时间和传播延迟影响:

- 转换速率 SR 越快, 上升/下降转变时间越小
- 转换速率 SR 越快, 传播延迟越小

驱动能力 DR 对上升/下降转变时间和传播延迟影响:

- 驱动能力 DR 越高, 上升/下降转变时间越小
- 驱动能力 DR 越高, 传播延迟越小

17.2 管脚复用

PA0~PA7, PB2~PB4, NRST/PC0 作为数字模拟复用管脚, 控制参见 Figure 149 和 Figure 150。虚线框中内容为 GPIO 模块寄存器中的控制位, 具体参见 17.9 GPIO 寄存器说明。

以上管脚作为模拟 IO 使用时, 该管脚的 IE, OE, PD, PU, OD 由硬件自动强制清零; 作为数字 IO 使用时, 管脚的 IE, OE, PD, PU, OD 由软件控制, 参见 Figure 149 和 Figure 150 中 AIOEN 所示控制。

Figure 149 为 PA0~PA7, PB2~PB4 管脚复用框图, 产生 EXINT 中断, 其中 PA7 作为模拟 IO 使用时, 作为 VREF 功能。

Figure 149 PA0~PA7, PB2~PB4 管脚复用框图

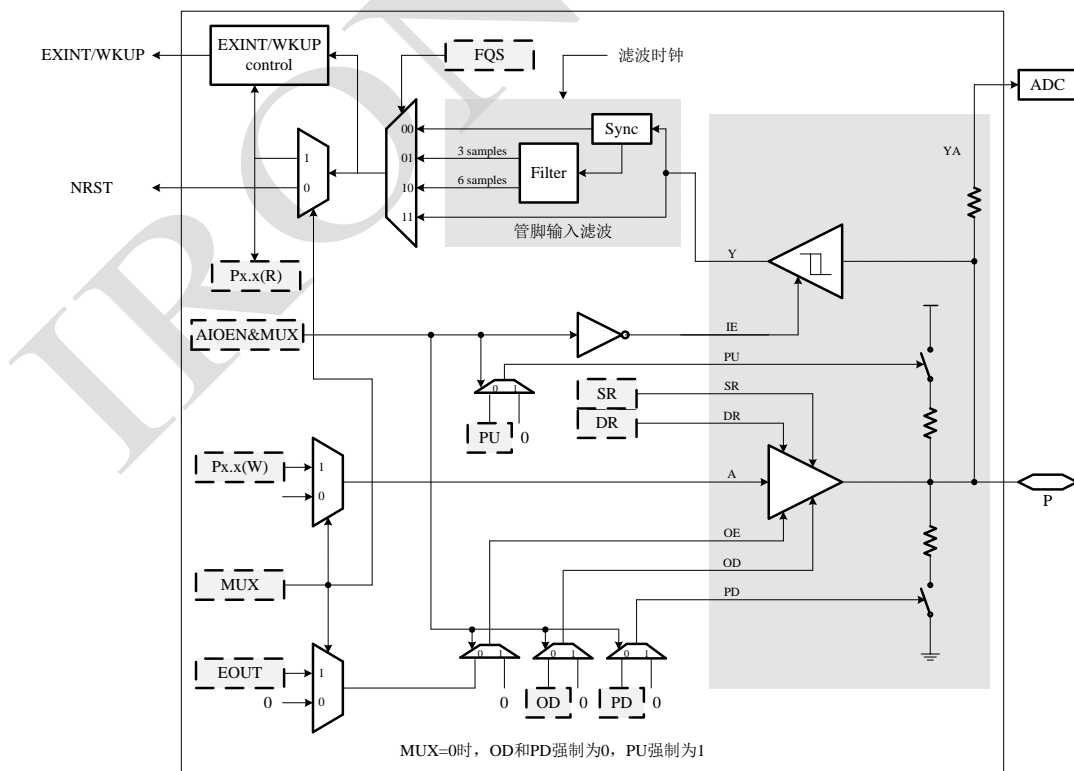


Figure 150 PC0/NRST 管脚复用框图

PB0~PB1, PB5~PB7 作为纯数字管脚, 控制参见 Figure 151, 产生 EXINT 中断。

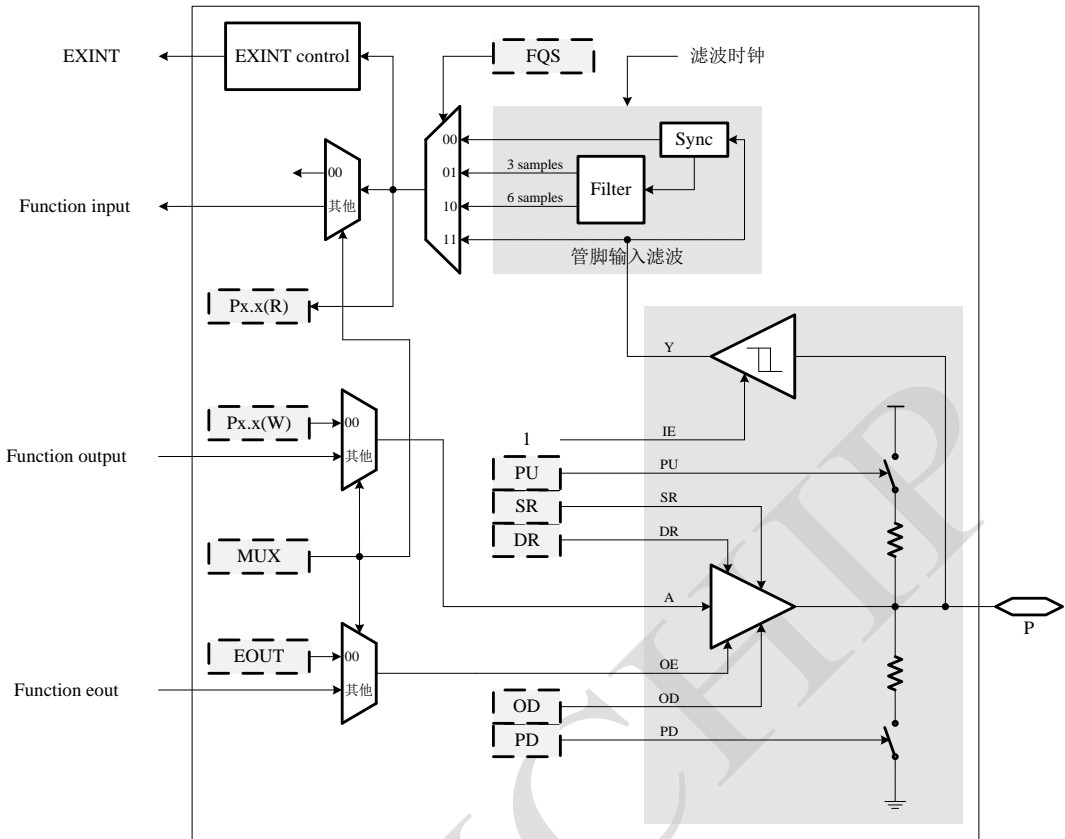


Figure 151 PB0~PB1, PB5~PB7 管脚复用框图

其中，NRST/PC0，PB5 和 PB6 三个管脚出去常供电区域，若管脚设置为 GPIO 输出功能，进入 SLEEP 或 DEEPSLEEP 低功耗模式时，管脚的输出状态可保持。

17.3 GPIO 功能复用

通过配置 MUX 寄存器，每个管脚都可以选择复用功能。

Table 237 管脚功能复用

管脚	GPIO (Default)	复用功能 1	复用功能 2	复用功能 3
PA0	GPIO-PA0	PWM1_CH0N	ADC_ETR	PWM0_ETR
PA1	GPIO-PA1	PWM1_CH1	PWM0_CH3	T1
PA2	GPIO-PA2	SPI_NSS	PWM0_CH2	T0
PA3	GPIO-PA3	SPI_MISO	PWM0_CH1	PWM1_CH1
PA4	GPIO-PA4	SPI_MOSI	PWM0_CH0	PWM_CKIN
PA5	GPIO-PA5	SPI_SCK	PWM1_CH0	PWM0_CH2N
PA6	GPIO-PA6	BZ_OUT	PWM0_CH3	PWM0_CH1N
PA7	GPIO-PA7	T0_GATE	PWM0_CH2	PWM0_CH0N
PB0	GPIO-PB0	I2C_SCL	ADC_ETR	PWM0_CH2
PB1	GPIO-PB1	I2C_SDA	PWM_BKIN	PWM0_CH1

PB2	GPIO-PB2	PWM1_CH0	T1_GATE	BZ_OUT
PB3	GPIO-PB3	UART_TX	T2	MOS_OC
PB4	GPIO-PB4	UART_RX	T2_GATE	MOS_OD
PB5	OSC_IN	GPIO-PB5		
PB6	OSC_OUT	GPIO-PB6		
PB7	GPIO-PB7	SPI_NSS	PWM1_CH1	PWM0_CH0
NRST	NRST	GPIO-PC0		

17.4 管脚数据寄存器读写

管脚数据寄存器（P0、P1 和 P2）读写遵循以下规则：

管脚数据寄存器写入：

- GPIO 输出模式下：写入将要输出到管脚的电平；
- 其它模式下：只将数据锁存到数据寄存器，不影响管脚电平。

管脚数据寄存器读取：

- 对管脚数据寄存器执行“读取-修改-写入”指令时，读取管脚数据寄存器值；
- 执行其它指令时，读取管脚输入电平值。

17.5 外设管脚复用配置

Table 238 PWM 管脚配置

管脚复用功能	说明	配置需求
PWM_BKIN	PWM 刹车输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
PWM0_ETR	PWM0 外部时钟输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
PWM0_CHx	PWM0 通道 x 输入捕获	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0；

管脚复用功能	说明	配置需求
		PU 和 PD 可根据需要配置。
	PWM0 通道 x 输出比较	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
PWM0_CHxN	PWM0 通道 x 互补输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
PWM1_CH0	PWM1 通道 0 输入捕获	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
	PWM1 通道 0 输出比较	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
PWM1_CH0N	PWM1 通道 1 输入捕获	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
	PWM1 通道 0 互补输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
PWM1_CH1	PWM1 通道 2 输入捕获	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。

Table 239 ADC 外部触发管脚配置

管脚复用功能	说明	配置需求
ADC_ETR	ADC 外部触发输入	相应管脚配置： 复用选择配置为该复用功能；

管脚复用功能	说明	配置需求
		滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。

Table 240 蜂鸣器管脚配置

管脚复用功能	说明	配置需求
BZ_OUT	蜂鸣器输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。

Table 241 UART 管脚配置

管脚复用功能	说明	配置
UART_TX	UART 模式数据输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
	同步移位寄存器模式时钟输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
UART_RX	UART 模式数据输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； PD 和 OD 配置为 0； PU 可根据需要配置。
	同步移位寄存器模式数据输入/输出	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。

Table 242 TIMER 管脚配置

管脚复用功能	说明	配置
T0	Timer0 外部时钟输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0；

管脚复用功能	说明	配置
		PU 和 PD 可根据需要配置。
T0_GATE	Timer0 外部门控输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； PD 和 OD 配置为 0； PU 可根据需要配置。
T1	Timer1 外部时钟输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
T1_GATE	Timer1 外部门控输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； PD 和 OD 配置为 0； PU 可根据需要配置。
T2	Timer2 外部时钟输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
T2_GATE	Timer2 外部门控输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； PD 和 OD 配置为 0； PU 可根据需要配置。

Table 243 SPI 管脚配置

管脚复用功能	说明	配置
SPI_SCK	主机模式 SPI 时钟输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
	从机模式 SPI 时钟输入	相应管脚配置：

管脚复用功能	说明	配置
		复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
SPI_NSS	主机模式 SPI 使能输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
	从机模式 SPI 使能输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
SPI_MOSI	主机模式 SPI 数据输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
	从机模式 SPI 数据输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
SPI_MISO	主机模式 SPI 数据输入	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 0； PU 和 PD 可根据需要配置。
	从机模式 SPI 数据输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。

Table 244 I2C 管脚配置

管脚复用功能	说明	配置
I2C_SCL	I2C 时钟开漏输出	相应管脚配置： 复用选择配置为该复用功能；

管脚复用功能	说明	配置
		滤波 FQS 可根据需要配置； OD 配置为 1； PD 配置为 0； PU 可根据需要配置。
I2C_SDA	I2C 数据开漏输出	相应管脚配置： 复用选择配置为该复用功能； 滤波 FQS 可根据需要配置； OD 配置为 1； PD 配置为 0； PU 可根据需要配置。

Table 245 MOS 管脚配置

管脚复用功能	说明	配置
MOS_OC	MOS 的 OC 控制输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。
MOS_OD	MOS 的 OD 控制输出	相应管脚配置： 复用选择配置为该复用功能； PU、PD 和 OD 配置为 0。

Table 246 JTAG 管脚配置

管脚复用功能	说明	配置
TCK	JTAG 时钟输入	相应管脚配置： 滤波 FQS 配置为 11，选择直通模式； 配置寄存器 JTAGEN 为 1，使管脚为 JTAG 功能，硬件自动将 PD 置 1，PU 和 OD 清 0。
TMS	JTAG 控制输入	相应管脚配置： 滤波 FQS 配置为 11，选择直通模式； 配置寄存器 JTAGEN 为 1，使管脚为 JTAG 功能，硬件自动将 PU 置 1，PD 和 OD 清 0。
TDI	JTAG 数据输入	相应管脚配置： 滤波 FQS 配置为 11，选择直通模式；

管脚复用功能	说明	配置
		配置寄存器 JTAGEN 为 1, 使管脚为 JTAG 功能, 硬件自动将 PU 置 1, PD 和 OD 清 0。
TDO	JTAG 数据输出	相应管脚配置: 配置寄存器 JTAGEN 为 1, 使管脚为 JTAG 功能, 硬件自动将 PU, PD 和 OD 清 0。

若 JTAGEN 使能, PA0, PA1, PA2, PA3 作为 JTAG 管脚使用, PA0 弱下拉, PA1 弱上拉, PA2 弱上拉; 此时 PA0CTRL, PA1CTRL, PA2CTRL, PA3CTRL 寄存器不起作用。

17.6 管脚滤波

为了消除外部环境对 I/O 的干扰而产生对全芯片工作状态的影响, 芯片所有数字输入管脚均可选择滤波处理 (消除干扰)。滤波实现架构框图如 Figure 152 所示。

输入管脚可经过四种不同处理方式, 默认选择同步 (sync) 模式:

- 直通模式 (bypass) 模式
- 同步 (Sync) 模式, 利用滤波时钟进行同步
- 3 个滤波时钟周期滤波 (可滤除小于 2 个滤波时钟周期的脉冲)
- 6 个滤波时钟周期滤波 (可滤除小于 5 个滤波时钟周期的脉冲)

其中的滤波时钟为 gpio_clk 经过预分频后的时钟。

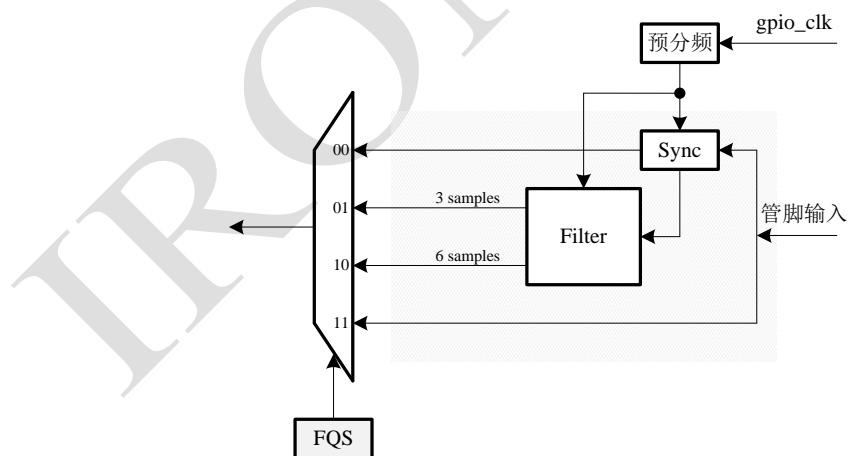


Figure 152 输入管脚滤波

17.7 EXINT 中断

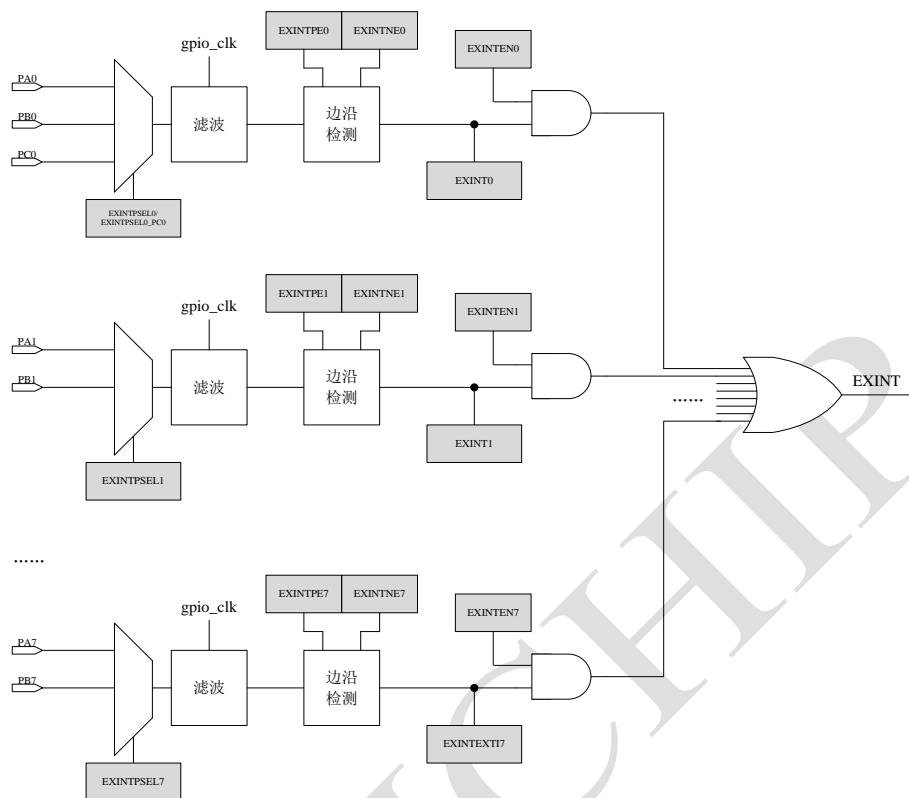


Figure 153 EXINT 中断

EXINT 中断包括 8 个外部管脚中断：

- 外部管脚中断 EXINT0:
 - 1) 配置寄存器 EXINTPSE0 和 EXINTPSE0_PC0, 从 PA0, PB0 和 PC0 中选择 EXINT0 的中断管脚 PX0(其中 X=A, B, C)
 - 2) 配置寄存器 PX0CTRL.FQS, 设置管脚 PX0 输入滤波数
 - 3) 配置寄存器 PX0CTRL.MUX 为 0, 配置寄存器 PX0CTRL.EOUT 为 0, 设置管脚 PX0 为 GPIO 输入
 - 4) 配置寄存器 EXINTPE0 和 EXINTNE0, 选择 EXINT0 中断类型
 - 5) 配置寄存器 EXINTEN0, 选择打开或关断 EXINT0 中断
 - 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
 - 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
- 外部管脚中断 EXINT1:
 - 1) 配置寄存器 EXINTPSEL1, 从 PA1 和 PB1 中选择 EXINT1 的中断管脚 PX1(其中 X= A, B)
 - 2) 配置寄存器 PX1CTRL.FQS, 设置管脚 PX1 输入滤波数

- 3) 配置寄存器 PX1CTRL.MUX 为 0, 配置寄存器 PX1CTRL.EOUT 为 0, 设置管脚 PX1 为 GPIO 输入
 - 4) 配置寄存器 EXINTPE1,和 EXINTNE1, 选择 EXINT1 中断类型
 - 5) 配置寄存器 EXINTEN1, 选择打开或关断 EXINT1 中断
 - 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
 - 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
- 外部管脚中断 EXINT2:
 - 1) 配置寄存器 EXINTPSEL2, 从 PA2 和 PB2 中选择 EXINT2 的中断管脚 PX2(其中 X= A, B)
 - 2) 配置寄存器 PX2CTRL.FQS, 设置管脚 PX2 输入滤波数
 - 3) 配置寄存器 PX2CTRL.MUX 为 0, 配置寄存器 PX2CTRL.EOUT 为 0, 设置管脚 PX2 为 GPIO 输入
 - 4) 配置寄存器 EXINTPE2 和 EXINTNE2, 选择 EXINT2 中断类型
 - 5) 配置寄存器 EXINTEN2, 选择打开或关断 EXINT2 中断
 - 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
 - 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
 - 外部管脚中断 EXINT3:
 - 1) 配置寄存器 EXINTPSEL3, 从 PA3 和 PB3 中选择 EXINT3 的中断管脚 PX3(其中 X= A, B)
 - 2) 配置寄存器 PX3CTRL.FQS, 设置管脚 PX3 输入滤波数
 - 3) 配置寄存器 PX3CTRL.MUX 为 0, 配置寄存器 PX3CTRL.EOUT 为 0, 设置管脚 PX3 为 GPIO 输入
 - 4) 配置寄存器 EXINTPE3 和 EXINTNE3, 选择 EXINT3 中断类型
 - 5) 配置寄存器 EXINTEN3, 选择打开或关断 EXINT3 中断
 - 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
 - 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
 - 8) 若使用查询方式, 则 5) ~7) 不用配置
 - 外部管脚中断 EXINT4:
 - 1) 配置寄存器 EXINTPSEL4, 从 PA4 和 PB4 中选择 EXINT4 的中断管脚 PX4(其中 X= A, B)
 - 2) 配置寄存器 PX4CTRL.FQS, 设置管脚 PX4 输入滤波数
 - 3) 配置寄存器 PX4CTRL.MUX 为 0, 配置寄存器 PX4CTRL.EOUT 为 0, 设置管

脚 PX4 为 GPIO 输入

- 4) 配置寄存器 EXINTPE4 和 EXINTNE4, 选择 EXINT4 中断类型
- 5) 配置寄存器 EXINTEN4, 选择打开或关断 EXINT4 中断
- 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
- 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
- 8) 若使用查询方式, 则 5) ~7) 不用配置

● 外部管脚中断 EXINT5:

- 1) 配置寄存器 EXINTPSEL5, 从 PA5 和 PB5 中选择 EXINT5 的中断管脚 PX5(其中 X= A, B)
- 2) 配置寄存器 PX5CTRL.FQS, 设置管脚 PX5 输入滤波数
- 3) 配置寄存器 PX5CTRL.MUX 为 0, 配置寄存器 PX5CTRL.EOUT 为 0, 设置管脚 PX5 为 GPIO 输入
- 4) 配置寄存器 EXINTPE5 和 EXINTNE5, 选择 EXINT5 中断类型
- 5) 配置寄存器 EXINTEN5, 选择打开或关断 EXINT5 中断
- 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
- 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
- 8) 若使用查询方式, 则 5) ~7) 不用配置

● 外部管脚中断 EXINT6:

- 1) 配置寄存器 EXINTPSEL6, 从 PA6 和 PB6 中选择 EXINT6 的中断管脚 PX6(其中 X= A, B)
- 2) 配置寄存器 PX6CTRL.FQS, 设置管脚 PX6 输入滤波数
- 3) 配置寄存器 PX6CTRL.MUX 为 0, 配置寄存器 PX6CTRL.EOUT 为 0, 设置管脚 PX6 为 GPIO 输入
- 4) 配置寄存器 EXINTPE6 和 EXINTNE6, 选择 EXINT6 中断类型
- 5) 配置寄存器 EXINTEN6, 选择打开或关断 EXINT6 中断
- 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
- 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
- 8) 若使用查询方式, 则 5) ~7) 不用配置

● 外部管脚中断 EXINT7:

- 1) 配置寄存器 EXINTPSEL7, 从 PA7 和 PB7 中选择 EXINT7 的中断管脚 PX7(其中 X= A, B)
- 2) 配置寄存器 PX7CTRL.FQS, 设置管脚 PX7 输入滤波数
- 3) 配置寄存器 PX7CTRL.MUX 为 0, 配置寄存器 PX7CTRL.EOUT 为 0, 设置管脚 PX7 为 GPIO 输入

- 4) 配置寄存器 EXINTPE7 和 EXINTNE7, 选择 EXINT7 中断类型
- 5) 配置寄存器 EXINTEN7, 选择打开或关断 EXINT7 中断
- 6) 配置寄存器 IEN0.2, 选择打开或关断 EXINT 中断
- 7) 配置寄存器 IEN0.7, 选择打开或关断总中断
- 8) 若使用查询方式, 则 5) ~7) 不用配置

8 个外部管脚中断 EXINT0~ EXINT7 经过或逻辑组合成 EXINT 中断, 每个中断都有各自的中断使能, 组合后中断使能由寄存器 IEN0.2 控制。

硬件实现 8 个外部管脚中断 EXINT0~ EXINT7 时没有优先级, 软件可根据需要灵活编程。

17.8 PC0 唤醒

当 PC0 设置为 GPIO 功能时, 可将芯片从任意低功耗模式唤醒。唤醒使能、输入滤波和唤醒有效沿可配置:

- a) 配置寄存器 PC0CTRL.FQS, 设置管脚 PC0 输入滤波数;
- b) 配置寄存器 EXINTPE0 和 EXINTNE0, 设置 PC0 唤醒有效沿, 可选择上升沿、下降沿或双沿, PC0 唤醒有效沿和 PC0 中断有效沿一致。

若需要支持唤醒 DEEPSLEEP 模式, PC0 管脚的滤波需要软件配置为直通模式。

17.9 GPIO 寄存器

GPIO 寄存器对应的 PAGE 如下:

- 管脚数据寄存器 P0, P1 和 P2 对应的 SFRPAGE 为所有 PAGE (公共寄存器), 也即在所有 PAGE 配置这 3 个寄存器均生效;
- 其他寄存器对应的 SFRPAGE 为 2。

Table 247 GPIO 寄存器 (SFRPAGE=0x02)

地址	寄存器名	类型	复位值	功能描述
0x89	PA0CTRL	W/R	0x30	PA0 控制寄存器
0x8A	PA1CTRL	W/R	0x30	PA1 控制寄存器
0x8B	PA2CTRL	W/R	0x30	PA2 控制寄存器
0x8C	PA3CTRL	W/R	0x00	PA3 控制寄存器
0x8D	PA4CTRL	W/R	0x00	PA4 控制寄存器
0x8E	PA5CTRL	W/R	0x00	PA5 控制寄存器
0x8F	PA6CTRL	W/R	0x00	PA6 控制寄存器
0x99	PA7CTRL	W/R	0x00	PA7 控制寄存器
0x91	PB0CTRL	W/R	0x00	PB0 控制寄存器

地址	寄存器名	类型	复位值	功能描述
0x92	PB1CTRL	W/R	0x00	PB1 控制寄存器
0x93	PB2CTRL	W/R	0x00	PB2 控制寄存器
0x94	PB3CTRL	W/R	0x04	PB3 控制寄存器
0x95	PB4CTRL	W/R	0x04	PB4 控制寄存器
0xBA	PB5CTRL	W/R	0x00	PB5 控制寄存器
0xBB	PB6CTRL	W/R	0x00	PB6 控制寄存器
0x9A	PB7CTRL	W/R	0x00	PB7 控制寄存器
0xB9	PC0CTRL	W/R	0x00	PC0 控制寄存器
0x9B	PADSRDR	W/R	0x3F	PAD 转换速率和驱动能力寄存器
0x9C	AIOEN0	W/R	0x00	模拟管脚使能寄存器 0
0x9D	AIOEN1	W/R	0x00	模拟管脚使能寄存器 1
0x9E	FPSC	W/R	0x00	滤波时钟预分频寄存器
0xA3	EXINTPE	W/R	0x00	EXINT 上升沿中断使能寄存器
0xA4	EXINTNE	W/R	0x00	EXINT 下降沿中断使能寄存器
0xA5	EXINTEN	W/R	0x00	EXINT 中断使能寄存器
0xA6	EXINTCON	W/R	0x00	EXINT 中断控制寄存器
0xA1	EXINTSEL0	W/R	0x00	EXINT 中断管脚选择寄存器 0
0xA2	EXINTSEL1	W/R	0x00	EXINT 中断管脚选择寄存器 1
0x80	P0	W/R	0xFF	PA 数据寄存器
0x90	P1	W/R	0xFF	PB 数据寄存器
0xA0	P2	W/R	0xFF	PC 数据寄存器

17.9.1 PA0 控制寄存器(PA0CTRL) [0x89]

Table 248 PA0 控制寄存器(PA0CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA0 开漏使能： 0：不使能 1：使能
EOUT	1	W/R	0	PA0 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PA0 上拉使能：

参数名	比特位	属性	复位值	描述
				0: 不使能 1: 使能
PD	3	W/R	0	PA0 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	11	PA0 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	7-6	W/R	0	PA0 功能选择: 00: GPIO-PA0 (P0.0) 01: PWM1_CH0N 10: ADC_ETR 11: PWM0_ETR

17.9.2 PA1 控制寄存器(PA1CTRL) [0x8A]

Table 249 PA1 控制寄存器(PA1CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA1 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PA1 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PA1 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PA1 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	11	PA1 输入滤波选择: 00: 仅同步

参数名	比特位	属性	复位值	描述
				01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	7-6	W/R	0	PA1 功能选择: 00: GPIO-PA1 (P0.1) 01: PWM1_CH1 10: PWM0_CH3 11: T1

17.9.3 PA2 控制寄存器(PA2CTRL) [0x8B]

Table 250 PA2 控制寄存器(PA2CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA2 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PA2 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PA2 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PA2 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	11	PA2 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	7-6	W/R	0	PA2 功能选择: 00: GPIO-PA2 (P0.2) 01: SPI_NSS 10: PWM0_CH2

参数名	比特位	属性	复位值	描述
				11: T0

17.9.4 PA3 控制寄存器(PA3CTRL) [0x8C]

Table 251 PA3 控制寄存器(PA3CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA3 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PA3 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PA3 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PA3 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PA3 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	7-6	W/R	0	PA3 功能选择: 00: GPIO-PA3 (P0.3) 01: SPI_MISO 10: PWM0_CH1 11: PWM1_CH1

17.9.5 PA4 控制寄存器(PA4CTRL) [0x8D]

Table 252 PA4 控制寄存器(PA4CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA4 开漏使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
EOUT	1	W/R	0	PA4 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PA4 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PA4 下拉使能： 0：不使能 1：使能
FQS	5-4	W/R	0	PA4 输入滤波选择： 00：仅同步 01：可滤除小于 2 个滤波时钟周期的脉冲 10：可滤除小于 5 个滤波时钟周期的脉冲 11：直通
MUX	7-6	W/R	0	PA4 功能选择： 00：GPIO-PA4 (P0.4) 01：SPI_MOSI 10：PWM0_CH0 11：PWM_BKIN

17.9.6 PA5 控制寄存器(PA5CTRL) [0x8E]

Table 253 PA5 控制寄存器(PA5CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA5 开漏使能： 0：不使能 1：使能
EOUT	1	W/R	0	PA5 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PA5 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PA5 下拉使能：

参数名	比特位	属性	复位值	描述
				0: 不使能 1: 使能
FQS	5-4	W/R	0	PA5 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	7-6	W/R	0	PA5 功能选择: 00: GPIO-PA5 (P0.5) 01: SPI_SCK 10: PWM1_CH0 11: PWM0_CH2N

17.9.7 PA6 控制寄存器(PA6CTRL) [0x8F]

Table 254 PA6 控制寄存器(PA6CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA6 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PA6 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PA6 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PA6 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PA6 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通

参数名	比特位	属性	复位值	描述
MUX	7-6	W/R	0	PA6 功能选择: 00: GPIO-PA6 (P0.6) 01: BZ_OUT 10: PWM0_CH3 11: PWM0_CH1N

17.9.8 PA7 控制寄存器(PA7CTRL) [0x99]

Table 255 PA7 控制寄存器(PA7CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PA7 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PA7 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	1	PA7 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PA7 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PA7 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	7-6	W/R	0	PA7 功能选择: 00: GPIO-PA7 (P0.7) 01: T0_GATE 10: PWM0_CH2 11: PWM0_CH0N

17.9.9 PB0 控制寄存器(PB0CTRL) [0x91]

Table 256 PB0 控制寄存器(PB0CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB0 开漏使能： 0：不使能 1：使能
EOUT	1	W/R	0	PB0 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PB0 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PB0 下拉使能： 0：不使能 1：使能
FQS	5-4	W/R	0	PB0 输入滤波选择： 00：仅同步 01：可滤除小于 2 个滤波时钟周期的脉冲 10：可滤除小于 5 个滤波时钟周期的脉冲 11：直通
MUX	7-6	W/R	0	PB0 功能选择： 00：GPIO-PB0 (P1.0) 01：I2C_SCL 10：ADC_ETR 11：PWM0_CH2

17.9.10 PB1 控制寄存器(PB1CTRL) [0x92]

Table 257 PB1 控制寄存器(PB1CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB1 开漏使能： 0：不使能 1：使能
EOUT	1	W/R	0	PB1 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PB1 上拉使能：

参数名	比特位	属性	复位值	描述
				0: 不使能 1: 使能
PD	3	W/R	0	PB1 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PB1 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	7-6	W/R	0	PB1 功能选择: 00: GPIO-PB1 (P1.1) 01: I2C_SDA 10: PWM_BKIN 11: PWM0_CH1

17.9.11 PB2 控制寄存器(PB2CTRL) [0x93]

Table 258 PB2 控制寄存器(PB2CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB2 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PB2 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PB2 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PB2 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PB2 输入滤波选择: 00: 仅同步

参数名	比特位	属性	复位值	描述
				01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	7-6	W/R	0	PB2 功能选择: 00: GPIO-PB2 (P1.2) 01: PWM1_CH0 10: T1_GATE 11: BZ_OUT

17.9.12 PB3 控制寄存器(PB3CTRL) [0x94]

Table 259 PB3 控制寄存器(PB3CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB3 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PB3 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	1	PB3 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PB3 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PB3 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	7-6	W/R	0	PB3 功能选择: 00: GPIO-PB3 (P1.3) 01: UART_TX 10: T2

参数名	比特位	属性	复位值	描述
				11: MOS_OC

17.9.13 PB4 控制寄存器(PB4CTRL) [0x95]

Table 260 PB4 控制寄存器(PB4CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB4 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PB4 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	1	PB4 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PB4 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PB4 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	7-6	W/R	0	PB4 功能选择: 00: GPIO-PB4 (P1.4) 01: UART_RX 10: T2_GATE 11: MOS_OD

17.9.14 PB5 控制寄存器(PB5CTRL) [0xBA]

Table 261 PB5 控制寄存器(PB5CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB5 开漏使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
EOUT	1	W/R	0	PB5 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PB5 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PB5 下拉使能： 0：不使能 1：使能
FQS	5-4	W/R	0	PB5 输入滤波选择： 00：仅同步 01：可滤除小于 2 个滤波时钟周期的脉冲 10：可滤除小于 5 个滤波时钟周期的脉冲 11：直通
MUX	6	W/R	0	PB5 功能选择： 0：OSC_IN 1：GPIO-PB5（P1.5）
Reserved	7	R	0	保留

17.9.15 PB6 控制寄存器(PB6CTRL) [0xBB]

Table 262 PB6 控制寄存器(PB6CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB6 开漏使能： 0：不使能 1：使能
EOUT	1	W/R	0	PB6 作为 GPIO 时，输出使能： 0：不使能 1：使能
PU	2	W/R	0	PB6 上拉使能： 0：不使能 1：使能
PD	3	W/R	0	PB6 下拉使能： 0：不使能

参数名	比特位	属性	复位值	描述
				1: 使能
FQS	5-4	W/R	0	PB6 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PB6 功能选择: 0: OSC_OUT 1: GPIO-PB6 (P1.6)
Reserved	7	R	0	保留

17.9.16 PB7 控制寄存器(PB7CTRL) [0x9A]

Table 263 PB7 控制寄存器(PB7CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PB7 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PB7 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PB7 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PB7 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PB7 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	7-6	W/R	0	PB7 功能选择: 00: GPIO-PB7 (P1.7)

参数名	比特位	属性	复位值	描述
				01: SPI_NSS 10: PWM1_CH1 11: PWM0_CH0

17.9.17 PC0 控制寄存器(PC0CTRL) [0xB9]

Table 264 PC0 控制寄存器(PC0CTRL)

参数名	比特位	属性	复位值	描述
OD	0	W/R	0	PC0 开漏使能: 0: 不使能 1: 使能
EOUT	1	W/R	0	PC0 作为 GPIO 时, 输出使能: 0: 不使能 1: 使能
PU	2	W/R	0	PC0 上拉使能: 0: 不使能 1: 使能
PD	3	W/R	0	PC0 下拉使能: 0: 不使能 1: 使能
FQS	5-4	W/R	0	PC0 输入滤波选择: 00: 仅同步 01: 可滤除小于 2 个滤波时钟周期的脉冲 10: 可滤除小于 5 个滤波时钟周期的脉冲 11: 直通
MUX	6	W/R	0	PC0 功能选择: 0: NRST 1: GPIO-PC0 (P2.0)
Reserved	7	R	0	保留

17.9.18 PAD 转换速率和驱动能力寄存器(PADSRDR) [0x9B]

Table 265 PAD 转换速率和驱动能力寄存器(PADSRDR)

参数名	比特位	属性	复位值	描述
SRPA	0	W/R	1	PA0~7 转换速率控制位: 0: 转换速率快

参数名	比特位	属性	复位值	描述
				1: 转换速率慢
DRPA	1	W/R	1	PA0~7 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
SRPB	2	W/R	1	PB0~7 转换速率控制位: 0: 转换速率快 1: 转换速率慢
DRPB	3	W/R	1	PB0~7 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
SRPC	4	W/R	1	PC0 转换速率控制位: 0: 转换速率快 1: 转换速率慢
DRPC	5	W/R	1	PC0 驱动能力控制位: 0: 驱动能力高 1: 驱动能力低
Reserved	7-6	R	0	保留

17.9.19 模拟管脚使能寄存器 0(AIOEN0) [0x9C]

Table 266 模拟管脚使能寄存器 0(AIOEN0)

参数名	比特位	属性	复位值	描述
AIOEN_PA0	0	W/R	0	PA0 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PA0 的 IE,OE,PD,PU,OD 自动由硬件强制为低
AIOEN_PA1	1	W/R	0	PA1 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PA1 的 IE,OE,PD,PU,OD 自动由硬件强制为低
AIOEN_PA2	2	W/R	0	PA2 作为模拟输入 IO 使能: 0: 不使能

参数名	比特位	属性	复位值	描述
				1: 使能 该比特使能时, PA2 的 IE,OE,PD,PU,OD 自动由硬件强制为低
AIOEN_PA3	3	W/R	0	PA3 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PA3 的 IE,OE,PD,PU,OD 自动由硬件强制为低
AIOEN_PA4	4	W/R	0	PA4 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PA4 的 IE,OE,PD,PU,OD 自动由硬件强制为低
AIOEN_PA5	5	W/R	0	PA5 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PA5 的 IE,OE,PD,PU,OD 自动由硬件强制为低
AIOEN_PA6	6	W/R	0	PA6 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PA6 的 IE,OE,PD,PU,OD 自动由硬件强制为低
AIOEN_PA7	7	W/R	0	PA7 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PA7 的 IE,OE,PD,PU,OD 自动由硬件强制为低

17.9.20 模拟管脚使能寄存器 1(AIOEN1) [0x9D]

Table 267 模拟管脚使能寄存器 1(AIOEN1)

参数名	比特位	属性	复位值	描述
AIOEN_PC0	0	W/R	0	当 PC0 设置为 GPIO 功能时, 作为模拟输

参数名	比特位	属性	复位值	描述
				入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PC0 的 IE,OE,PD,PU,OD 自动由硬件强制为低
AIOEN_PB2	1	W/R	0	PB2 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PB2 的 IE,OE,PD,PU,OD 自动由硬件强制为低
AIOEN_PB3	2	W/R	0	PB3 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PB3 的 IE,OE,PD,PU,OD 自动由硬件强制为低
AIOEN_PB4	3	W/R	0	PB4 作为模拟输入 IO 使能: 0: 不使能 1: 使能 该比特使能时, PB4 的 IE,OE,PD,PU,OD 自动由硬件强制为低
Reserved	7-4	R	0	保留

17.9.21 滤波时钟预分频寄存器(FPSC) [0x9E]

Table 268 滤波时钟预分频寄存器(FPSC)

参数名	比特位	属性	复位值	描述
FPSC	7-0	W/R	0	滤波时钟预分频系数: 滤波时钟周期 = $(FPSC + 1) \times T_{\text{gpio_clk}}$

17.9.22 EXINT 上升沿中断使能寄存器(EXINTPE) [0xA3]

Table 269 EXINT 上升沿中断使能寄存器(EXINTPE)

参数名	比特位	属性	复位值	描述
EXINTPE0	0	W/R	0	EXINT0 上升沿中断使能: 0: 不使能

参数名	比特位	属性	复位值	描述
				1: 使能
EXINTPE1	1	W/R	0	EXINT1 上升沿中断使能: 0: 不使能 1: 使能
EXINTPE2	2	W/R	0	EXINT2 上升沿中断使能: 0: 不使能 1: 使能
EXINTPE3	3	W/R	0	EXINT3 上升沿中断使能: 0: 不使能 1: 使能
EXINTPE4	4	W/R	0	EXINT4 上升沿中断使能: 0: 不使能 1: 使能
EXINTPE5	5	W/R	0	EXINT5 上升沿中断使能: 0: 不使能 1: 使能
EXINTPE6	6	W/R	0	EXINT6 上升沿中断使能: 0: 不使能 1: 使能
EXINTPE7	7	W/R	0	EXINT7 上升沿中断使能: 0: 不使能 1: 使能

17.9.23 EXINT 下降沿中断使能寄存器(EXINTNE) [0xA4]

Table 270 EXINT 下降沿中断使能寄存器(EXINTNE)

参数名	比特位	属性	复位值	描述
EXINTNE0	0	W/R	0	EXINT0 下降沿中断使能: 0: 不使能 1: 使能
EXINTNE1	1	W/R	0	EXINT1 下降沿中断使能: 0: 不使能 1: 使能

参数名	比特位	属性	复位值	描述
EXINTNE2	2	W/R	0	EXINT2 下降沿中断使能: 0: 不使能 1: 使能
EXINTNE3	3	W/R	0	EXINT3 下降沿中断使能: 0: 不使能 1: 使能
EXINTNE4	4	W/R	0	EXINT4 下降沿中断使能: 0: 不使能 1: 使能
EXINTNE5	5	W/R	0	EXINT5 下降沿中断使能: 0: 不使能 1: 使能
EXINTNE6	6	W/R	0	EXINT6 下降沿中断使能: 0: 不使能 1: 使能
EXINTNE7	7	W/R	0	EXINT7 下降沿中断使能: 0: 不使能 1: 使能

17.9.24 EXINT 中断使能寄存器(EXINTEN) [0xA5]

Table 271 EXINT 中断使能寄存器(EXINTEN)

参数名	比特位	属性	复位值	描述
EXINTEN0	0	W/R	0	EXINT0 中断使能: 0: 不使能 1: 使能
EXINTEN1	1	W/R	0	EXINT1 中断使能: 0: 不使能 1: 使能
EXINTEN2	2	W/R	0	EXINT2 中断使能: 0: 不使能 1: 使能
EXINTEN3	3	W/R	0	EXINT3 中断使能:

参数名	比特位	属性	复位值	描述
				0: 不使能 1: 使能
EXINTEN4	4	W/R	0	EXINT4 中断使能: 0: 不使能 1: 使能
EXINTEN5	5	W/R	0	EXINT5 中断使能: 0: 不使能 1: 使能
EXINTEN6	6	W/R	0	EXINT6 中断使能: 0: 不使能 1: 使能
EXINTEN7	7	W/R	0	EXINT7 中断使能: 0: 不使能 1: 使能

17.9.25 EXINT 中断控制寄存器(EXINTCON) [0xA6]

Table 272 EXINT 中断控制寄存器(EXINTCON)

参数名	比特位	属性	复位值	描述
EXINT0	0	W/R	0	EXINT0 中断标志位: <ul style="list-style-type: none"> ● 当有 EXINT0 中断申请时, 该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
EXINT1	1	W/R	0	EXINT1 中断标志位: <ul style="list-style-type: none"> ● 当有 EXINT1 中断申请时, 该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
EXINT2	2	W/R	0	EXINT2 中断标志位: <ul style="list-style-type: none"> ● 当有 EXINT2 中断申请时, 该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断

参数名	比特位	属性	复位值	描述
EXINT3	3	W/R	0	EXINT3 中断标志位： <ul style="list-style-type: none"> ● 当有 EXINT3 中断申请时，该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
EXINT4	4	W/R	0	EXINT4 中断标志位： <ul style="list-style-type: none"> ● 当有 EXINT4 中断申请时，该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
EXINT5	5	W/R	0	EXINT5 中断标志位： <ul style="list-style-type: none"> ● 当有 EXINT5 中断申请时，该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
EXINT6	6	W/R	0	EXINT6 中断标志位： <ul style="list-style-type: none"> ● 当有 EXINT6 中断申请时，该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断
EXINT7	7	W/R	0	EXINT7 中断标志位： <ul style="list-style-type: none"> ● 当有 EXINT7 中断申请时，该 bit 置为 1 ● 需要软件写 0 来清除 ● 不支持写 1 软中断

17.9.26 EXINT 中断管脚选择寄存器 0(EXINTSEL0) [0xA1]

Table 273 EXINT 中断管脚选择寄存器 0(EXINTSEL0)

参数名	比特位	属性	复位值	描述
EXINTPSEL0	0	W/R	0	EXINT0 中断管脚选择, EXINTPSEL0_PC0 为 1 时无效: 0: PA0 1: PB0

参数名	比特位	属性	复位值	描述
EXINTPSEL1	1	W/R	0	EXINT1 中断管脚选择: 0: PA1 1: PB1
EXINTPSEL2	2	W/R	0	EXINT2 中断管脚选择: 0: PA2 1: PB2
EXINTPSEL3	3	W/R	0	EXINT3 中断管脚选择: 0: PA3 1: PB3
EXINTPSEL4	4	W/R	0	EXINT4 中断管脚选择: 0: PA4 1: PB4
EXINTPSEL5	5	W/R	0	EXINT5 中断管脚选择: 0: PA5 1: PB5
EXINTPSEL6	6	W/R	0	EXINT6 中断管脚选择: 0: PA6 1: PB6
EXINTPSEL7	7	W/R	0	EXINT7 中断管脚选择: 0: PA7 1: PB7

17.9.27 EXINT 中断管脚选择寄存器 1(EXINTSEL1) [0xA2]

Table 274 EXINT 中断管脚选择寄存器 1(EXINTSEL1)

参数名	比特位	属性	复位值	描述
EXINTPSEL0_PC0	0	W/R	0	EXINT0 中断管脚选择: 0: PA0 或 PB0 1: PC0
Reserved	7-1	R	0	保留

17.9.28 PA 数据寄存器(P0) [0x80]

Table 275 PA 数据寄存器(P0)

参数名	比特位	属性	复位值	描述
-----	-----	----	-----	----

参数名	比特位	属性	复位值	描述
P0	7-0	W/R	0xFF	PA 数据寄存器

17.9.29 PB 数据寄存器(P1) [0x90]

Table 276 PB 数据寄存器(P1)

参数名	比特位	属性	复位值	描述
P1	7-0	W/R	0xFF	PB 数据寄存器

17.9.30 PC 数据寄存器(P2) [0xA0]

Table 277 PC 数据寄存器(P2)

参数名	比特位	属性	复位值	描述
P2	7-0	W/R	0xFF	PC 数据寄存器

18 电气特性

18.1 极限参数

Table 278 极限参数

符号	参数	最小值	最大值	单位
V_{DD}	供电电压	-0.3	5.5	V
V_{IN}	端口输入电压	-0.3	5.8	V
$I_{VDD-VSS}$	最大总电流	-	100	mA
I_{IO}	单个管脚注入电流	-10	10	mA
$I_{INJ(PIN)}$	总注入电流	-50	50	mA
T_A	环境温度	-40	125	°C
T_J	结温度	-40	150	°C
T_{STG}	储存温度	-55	150	°C

备注：超过该“极限参数”可能导致器件永久性损坏。长时间在最大允许值或超过最大允许值的条件下工作可能影响器件的可靠性。这里只是给出能承受的最大载荷，并不意味着在此条件下器件的功能性操作无误。

18.2 工作条件

18.2.1 芯片供电

Table 279 芯片供电

符号	参数	测试条件	最小	典型	最大	单位
V_{DD}	工作电压		2.75		5.5	V
f_{sys_clk}	系统工作时钟	$V_{DD}: 2.75V \sim 5.5V$		24.576		MHz

18.2.2 功耗特性

18.2.2.1 电流特性

Table 280 电流特性(芯片供电 5V)

符号	参数	测试条件	最小	典型	最大	单位
I_{DD1}	工作电流 (正常模式电流)	$f_{sys_clk} = 24.576MHz$, 关闭所有外设	6	6.6	7	mA
		$f_{sys_clk} = 6.144MHz$, 关闭所有外设	3.5	3.8	4.2	mA
I_{DD2}	IDLE 模式电流	$f_{sys_clk} = 24.576MHz$, 关闭所有外设	3	3.4	3.8	mA
I_{DD3}	IDLE 模式电流	$f_{sys_clk} = 6.144 MHz$, 关闭所有外设	2.3	2.7	3.1	mA

符号	参数	测试条件	最小	典型	最大	单位
I_{DD4}	STOP 模式电流	关闭所有外设	80	82	85	μA
I_{DD5}	SLEEP 模式电流		2.9	3.1	3.3	μA
I_{DD6}	Deep sleep 模式电流		2.4	2.6	2.8	μA

18.2.2.2 低功耗唤醒时间

Table 281 低功耗唤醒时间

符号	参数	测试条件	典型	单位
t_{WUIDLE}	从 IDLE 模式唤醒时间	使用内部 24.576MHz 时钟	40.7	ns
t_{WUSTOP}	从 STOP 模式唤醒时间	使用内部 WDT 唤醒 若使用 PC0 唤醒, 不考虑管脚滤波延时	20	μs
$t_{WUSLEEP}$	从 SLEEP 模式唤醒时间	使用内部 WDT 唤醒 若使用 PC0 唤醒, 不考虑管脚滤波延时	24.5	μs
$t_{WUDPSLEEP}$	从 Deep sleep 模式唤醒时间	使用 PC0 唤醒	24.5	μs

18.2.3 I/O 特性

Table 282 I/O 特性

符号	参数	测试条件	最小	典型	最大	单位
V_{IL}	I/O 输入低电平电压		-0.3		$0.35V_{DD}$	V
V_{IH}	I/O 输入高电平电压		$0.65 V_{DD}$		$V_{DD} + 0.3$	V
V_{OL}	I/O 输出低电平电压				$0.1 V_{DD}$	V
V_{OH}	I/O 输出高电平电压		$0.8 V_{DD}$			V
I_{IO}	I/O 灌电流				20	mA
	I/O 源电流				-20	mA
R_{PU}	I/O 上拉电阻		20		100	$K\Omega$
R_{PD}	I/O 下拉电阻		20		100	$K\Omega$
C_{IN}	I/O 输入电容				10	pF

18.2.4 REGC 引脚特性

REGC 引脚连接芯片内部 MAIN LDO 的外部负载电容。

需在该引脚固定连接一个 1 μF 的钽电容, 使内部 LDO 输出稳定, 如 Figure 154:

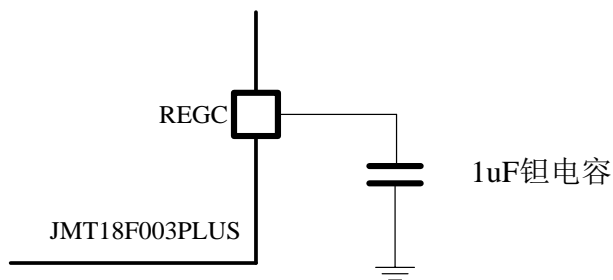


Figure 154 REGC 引脚连接示意图

18.2.5 NRST 引脚特性

NRST 引脚在芯片内部连接了一个不能断开的上拉电阻 (R_{PU})，滤波后生成内部复位信号。NRST 外部连接示意图参见 Figure 155，具体滤波配置参见寄存器 NRSTFQS 描述。

复位信号为低电平有效，输入的低电平必须满足以下两个条件，否则不会复位芯片：

- 小于 Table 282 定义的 V_{IL} 的最大值；
- 低电平宽度大于滤波的宽度。

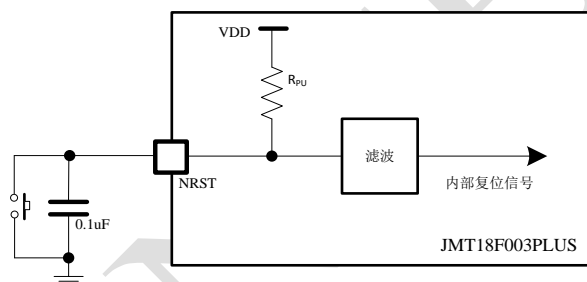


Figure 155 NRST 引脚连接示意图

18.2.6 上电/掉电条件

Table 283 上电/掉电工作条件

符号	参数	测试条件	最小	最大	单位
V_{POR}	产生 POR 复位的电压			500	mV
V_{BOR}	产生 BOR 复位的电压		2.6	2.7	V
t_{RT}	BOR 迟滞时间		30	150	us
t_{VDD}	V_{DD} 上升速率		3		V/ms

18.2.7 时钟特性

18.2.7.1 内部高速 RC 振荡器

Table 284 内部高速 RC 振荡器

符号	参数	测试条件	最小	典型	最大	单位
----	----	------	----	----	----	----

符号	参数	测试条件	最小	典型	最大	单位
f_{HRC}	HRC 的频率		72.99	73.728	74.46	MHz
ACC_{HRC}	HRC 的精度	$T_A = -40 \sim 125^\circ\text{C}$	-1		+1	%
t_{HRC}	HRC 的启动时间		2	5	9	us
$I_{DD(HRC)}$	HRC 的功耗			0.8	1	mA

18.2.7.2 内部低速 RC 振荡器

Table 285 内部低速 RC 振荡器

符号	参数	测试条件	最小	典型	最大	单位
f_{LRC}	LRC 的频率		27	32	36	KHz
T_{LRC}	LRC 的启动时间				90	us
$I_{DD(LRC)}$	LRC 的功耗			0.5	1	uA

18.2.7.3 外部时钟源

外部时钟源有两种方式：（1）连接一个谐振器至管脚 OSC_IN 和 OSC_OUT；（2）连接一个外部时钟至管脚 OSC_IN。

18.2.7.3.1 晶体/陶瓷谐振器

使用一个 8~24MHz 的晶体/陶瓷谐振器构成振荡器产生外部时钟。在应用中，谐振器和负载电容必须尽可能的靠近振荡器的引脚，以减少输出失真和启动的稳定时间。

Table 286 晶体/陶瓷谐振器

符号	参数	测试条件	最小	典型	最大	单位
f_{OSC_IN}	振荡器频率		8	22.1184	24	MHz
R_F	反馈电阻		1		10	MΩ
C_{L1} C_{L2}	建议负载电容		10		30	pF
I_2	内部振荡器驱动电流				2	mA
t_{SU}	启动时间	V_{DD} 稳定之后		0.7	1	ms

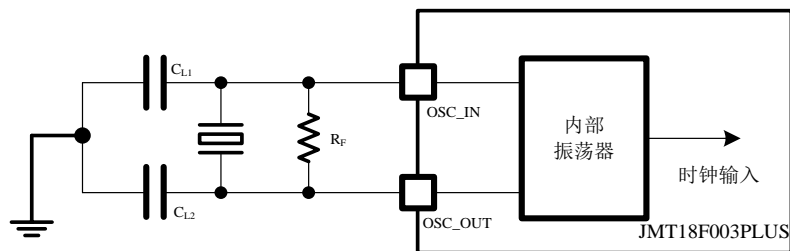


Figure 156 外部谐振器输入 OSC_IN/OSC_OUT 连接示意图

18.2.7.3.2 外部时钟

连接一个高速的外部时钟源至管脚 OSC_IN，可以给芯片提供工作时钟。

Table 287 外部时钟

符号	参数	测试条件	最小	典型	最大	单位
f_{EXT}	OSC_IN 输入的时钟频率		0		24	MHz
V_{EXTH}	OSC_IN 输入的时钟高电平		$0.7V_{DD}$		V_{DD}	V
V_{EXTL}	OSC_IN 输入的时钟低电平		0		$0.3V_{DD}$	V
$DUCY_{EXT}$	占空比		40		60	%
I_L	OSC_IN 输入漏电流				1	μA

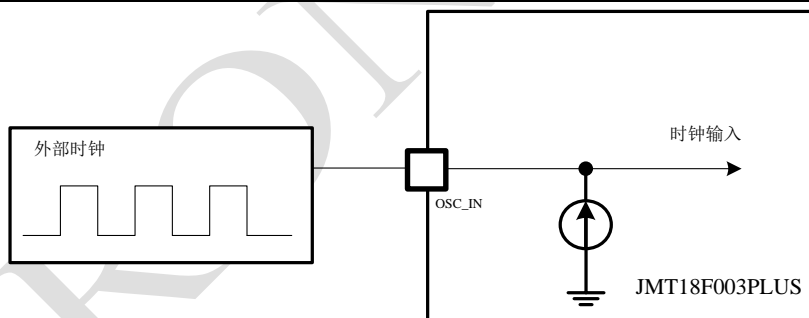


Figure 157 外部时钟输入 OSC_IN 连接示意图

18.2.8 通信接口

18.2.8.1 I2C

JMT18F003PLUS 的 I2C 接口符合标准 I2C 通信协议。SDA 和 SCL 不是真开漏的引脚，当配置为开漏输出时，在引脚和 VDD 之间的 PMOS 管被关闭，但仍然存在。

Table 288 I2C 接口特性

符号	参数	标准模式		快速模式		单位
		最小	最大	最小	最大	

符号	参数	标准模式		快速模式		单位
		最小	最大	最小	最大	
$t_{w(SCLL)}$	SCL 时钟低电平时间	4.7		1.3		us
$t_{w(SCLH)}$	SCL 时钟高电平时间	4.0		0.6		us
$t_{su(SDA)}$	SDA 建立时间	250		100		ns
$t_h(SDA)$	SDA 数据保持时间	0		0	0.9	us
$t_r(SDA)$ $t_r(SCL)$	SDA 和 SCL 上升时间		1000	$20+0.1C_b$	300	ns
$t_f(SDA)$ $t_f(SCL)$	SDA 和 SCL 下降时间		300		300	ns
$t_h(START)$	起始条件保持时间	4.0		0.6		us
$t_{su}(START)$	重复起始条件建立时间	4.7		0.6		us
$t_{su}(STOP)$	停止条件建立时间	4.0		0.6		us
$t_w(STOP:START)$	停止条件至起始条件的时间（总线空闲）	4.7		1.3		us
C_b	每条总线的容性负载		400		400	pF

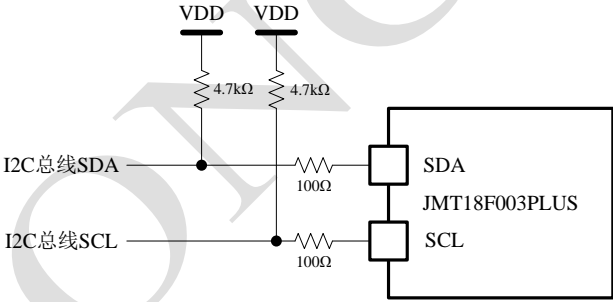


Figure 158 I2C 总线连接电路

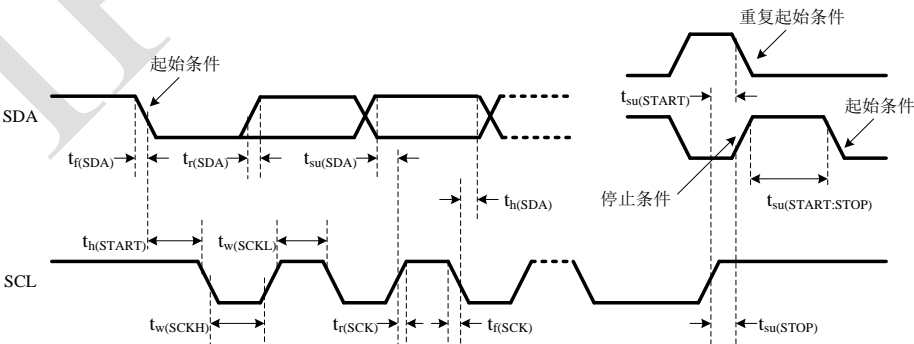


Figure 159 I2C 总线波形示意图

18.2.8.2 SPI

有关输入输出复用功能引脚（SPI_NSS、SPI_SCK、SPI_MOSI 和 SPI_MISO）的特性详情，参见 Table 282。

Table 289 SPI 特性

符号	参数	测试条件	最小	最大	单位
f_{SCK}	SPI 时钟频率	主机模式	0	12.288	MHz
$1/t_{c(SCK)}$		从机模式	0	12.288	
$t_{r(SCK)}$ $t_{f(SCK)}$	SPI 时钟上升和下降时间	负载电容: C=25pF		9	ns
$t_{su(NSS)}$	NSS 建立时间	从机模式	$4t_{sys_clk}$		ns
$t_{h(NSS)}$	NSS 保持时间	从机模式	$2t_{sys_clk}+20$		ns
$t_{w(SCKH)}$ $t_{w(SCKL)}$	SCK 高电平和低电平时间	主机模式, $f_{sys_clk}=73.728\text{MHz}$, 预分频系数为 1	50	60	ns
$t_{su(MI)}$	数据输入建立时间	主机模式	6		ns
$t_{su(SI)}$	数据输入建立时间	从机模式	6		ns
$t_{h(MI)}$	数据输入保持时间	主机模式	6		ns
$t_{h(SI)}$	数据输入保持时间	从机模式	6		ns
$t_{a(SO)}$	数据输出访问时间	从机模式		$4t_{sys_clk}$	ns
$t_{dis(SO)}$	数据输出禁止时间	从机模式	10		ns
$t_{v(SO)}$	数据输出有效时间	从机模式（使能边沿之后）		26	ns
$t_{v(MO)}$	数据输出有效时间	主机模式（使能边沿之后）		3	ns
$t_{h(SO)}$	数据输出保持时间	从机模式（使能边沿之后）	26		ns
$t_{h(MO)}$	数据输出保持时间	主机模式（使能边沿之后）	3		ns

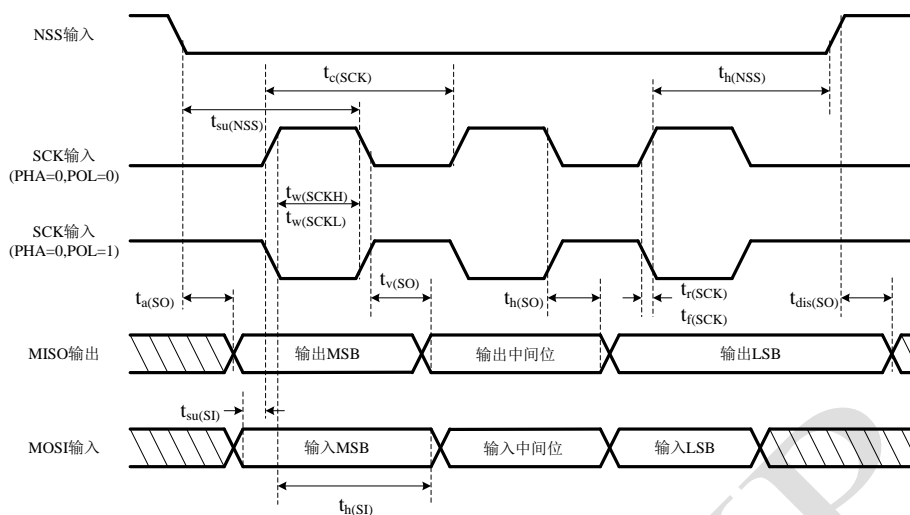


Figure 160 SPI 时序图 (从模式, PHA=0)

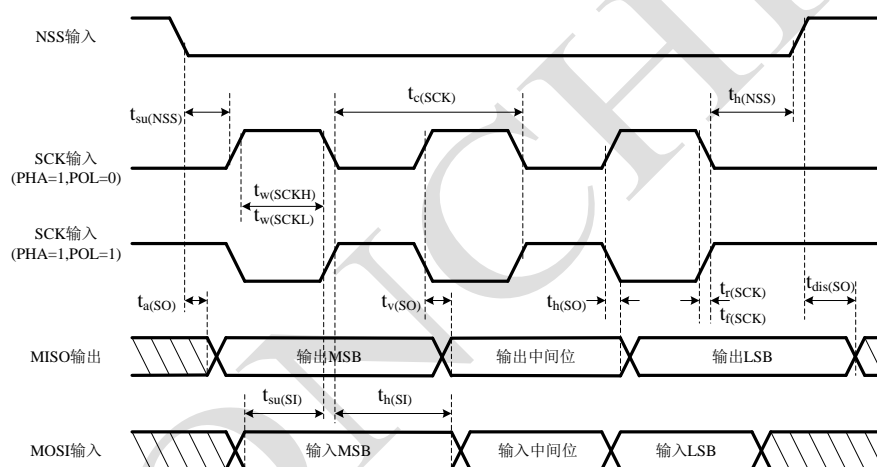


Figure 161 SPI 时序图 (从模式, PHA=1)

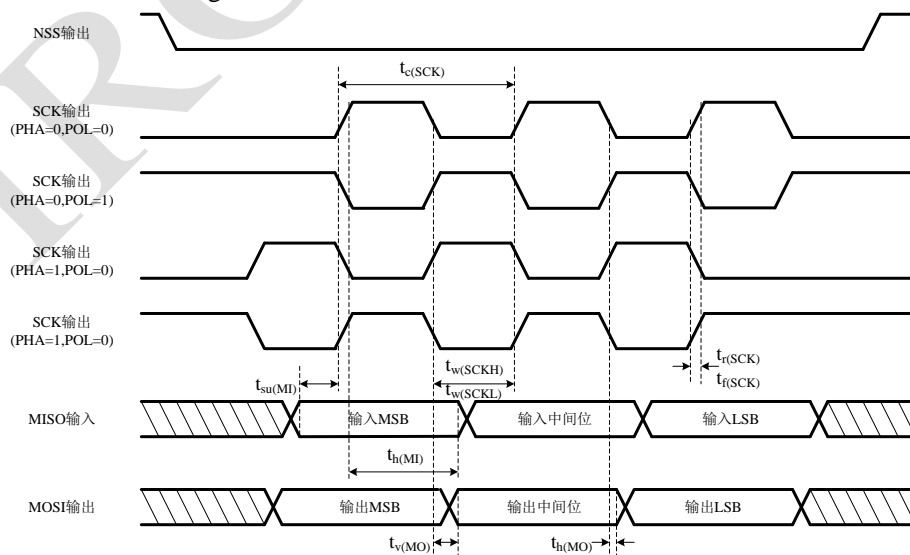


Figure 162 SPI 时序图 (主模式)

18.2.9 FLASH 特性

Table 290 FLASH 特性

符号	参数	测试条件	最小	典型	最大	单位
t_{prog}	8 位的编程时间	$T_A = -40 \sim 125^\circ\text{C}$	16.5	18.5	20	us
t_{erase}	扇区（512 字节）擦除时间	$T_A = -40 \sim 125^\circ\text{C}$	4	5	6	ms
V_{prog}	编程电压		1.62	1.8	1.98	V

Table 291 FLASH 寿命和数据保存期限

符号	参数	测试条件	最小	典型	最大	单位
N_{END}	寿命	$V_{\text{prog}} = 1.8\text{V}$	100000			次
T_{DR}	数据保存期限	$T_A = 25^\circ\text{C}$	10			年

18.2.10 ADC 特性

Table 292 ADC 电气特性

符号	参数	最小	典型	最大	单位
N_R	ADC 分辨率			10	BIT
DNL	ADC 非线性微分误差			± 1.2	LSB
INL	ADC 非线性积分误差			± 1.5	LSB
E_{OFF}	ADC 偏移误差			± 2	LSB
E_{GN}	ADC 增益误差			± 1.5	LSB
V_{REF}	ADC 参考电压	0		V_{DDA}	V
V_{INADC}	ADC 输入电压范围	0		V_{REF}	V
f_{SAMPLE}	ADC 采样速率	0		585	KSPS
$f_{\text{adcx_sample_clk}}$	ADC 时钟频率	0		8.192	MHz
$T_{\text{adcx_sample_clk}}$	ADC 转换器时钟周期			122	ns
t_{adc}	A/D 转换时间		14		$T_{\text{adcx_sample_clk}}$

为使得 ADC 达到更好的性能，模拟供电和参考电压的 PCB 设计建议按照 Figure 163 所示连接，其中电容建议选择瓷介电容，并且使得他们尽可能的靠近芯片。

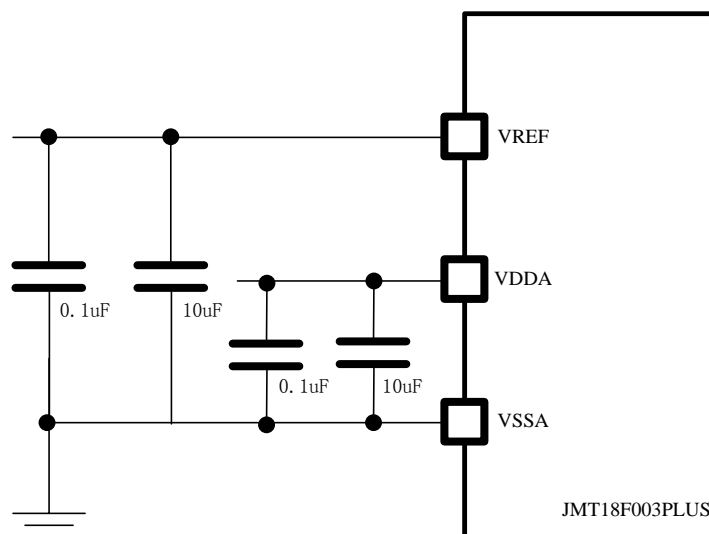


Figure 163 供电和参考电压去耦线路

18.2.11PGA 特性

Table 293 PGA 电气特性

符号	参数	测试条件	最小	典型	最大	单位
V_{opos}	输入失调电压	校准后	-4		+4	mV
V_{cm}	共模电压范围		0		$V_{\text{DDA}}-1.4$	V
V_{INPGA}	输入电压范围		0		V_{DDA}	V
V_{REFPGA}	参考电压范围			$1/2V_{\text{REF}}$		V
PSRR	电源电压抑制比			60		dB
CMRR	共模抑制比			60		dB
SR	转换速率			2.5		V/us
GAIN	增益		1/2		16	倍
GBW	单位增益带宽			10		MHz
GAIN ERROR	增益误差		-1		+1	%
Voltage Spectral Density	电压噪声频谱密度		20		55	nV/ $\sqrt{\text{Hz}}$

18.2.12CMP 特性

Table 294 CMP 电气特性

符号	参数	测试条件	最小	典型	最大	单位
V_{cpos}	输入失调电压		-15		+15	mV
V_{cm}	共模电压范围		0		$V_{\text{DDA}}-1.4$	V
V_{INCMP}	输入电压范围		0		V_{DDA}	V
V_{REFCMP}	参考电压范围		0		V_{REF}	V
PSRR	电源电压抑制比			60		dB
CMRR	共模抑制比			60		dB
t_{pd}	比较器响应时间			200		ns

18.2.13电气敏感性

18.2.13.1 ESD

ESD HBM 模型测试，参照“MIL-STD-883H Method 3015.8”标准。

ESD CDM 模型测试，参照“JEDEC EIA/JESD22-C101E”标准。

Table 295 ESD

符号	参数	测试条件	最大	单位
$V_{\text{ESD(HBM)}}$	Electrostatic discharge voltage (Human body model)	$T_A=25^\circ\text{C}$, MIL-STD-883J / Method 3015.9	6000	V

符号	参数	测试条件	最大	单位
$V_{ESD(CDM)}$	Electrostatic discharge voltage (Charge device model)	$T_A=25^{\circ}\text{C}$, JEDEC EIA/JESD22-C101E	1000	V

18.2.13.2 Latch Up

Latch Up（栓锁性能）测试，照“JEDEC78D”标准。

Table 296 Latch Up

符号	参数	测试条件	单位
LU	静态栓锁类	$T_A=125^{\circ}\text{C}$, JESD78D	200mA

19 封装特性

19.1 TSSOP20L

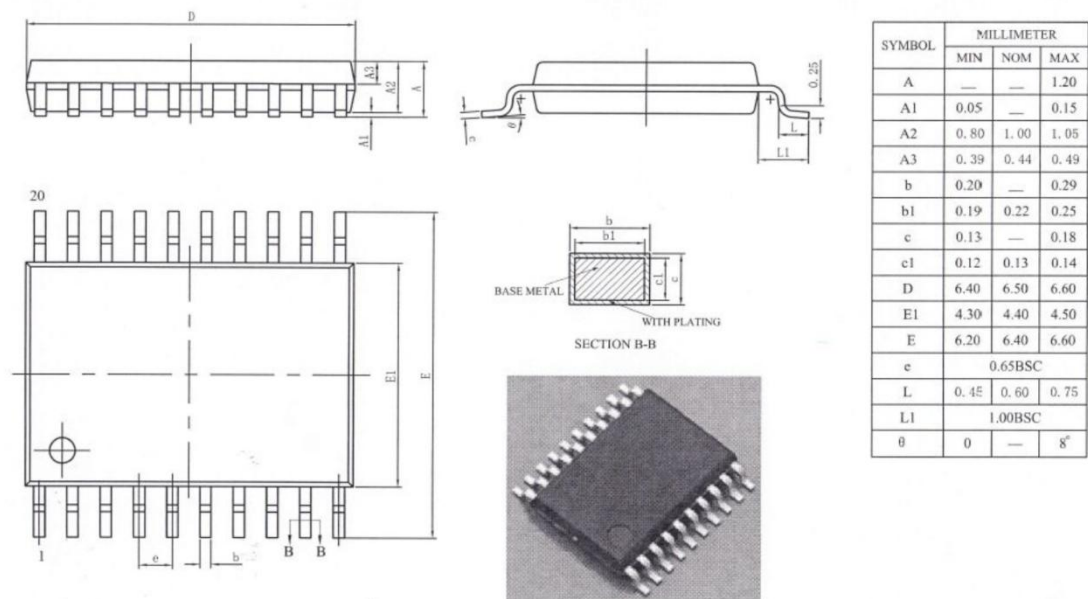


Figure 164 TSSOP20, 6.50X4.40mm 封装图

重要提示

➤ 江苏宏云技术有限公司保留对本手册中产品在可靠性、功能和设计方面的改进作进一步说明的权利。本手册内容如有变动，恕不另行通知。订购前建议用户咨询销售代表。

➤ 江苏宏云技术有限公司的公司名称和标识都是江苏宏云技术有限公司的注册商标。

➤ 本手册中的信息仅供参考，诸如功能概要和应用电路示例旨在说明江苏宏云技术有限公司半导体器件的使用方法和操作示例；江苏宏云技术有限公司不保证参考运用时器件的正常工作。如果用户根据该信息使用器件实行相关开发，用户应承担因此引发的责任。基于上述信息的使用引起任何损失，江苏宏云技术有限公司不承担任何责任。

➤ 本手册内的任何技术信息，包括功能介绍和原理图，不应理解为使用和执行任何知识产权的许可，诸如专利权或著作权，或江苏宏云技术有限公司的其他权利。任何经由不当手段侵害江苏宏云技术有限公司专利权的公司、组织或个人，江苏宏云技术有限公司将采取一切可能的法律行动，遏止侵权行为，并追讨江苏宏云技术有限公司所受的损失、或侵权者所得的不法利益。江苏宏云技术有限公司也不保证使用该信息不侵犯任何第三方知识产权或其他权利。因使用该信息引起的对第三方知识产权或其他权利的侵权行为，江苏宏云技术有限公司不承担任何责任。

➤ 本手册所介绍的产品旨在一般用途而设计、开发和制造，包括但不限于一般的工业使用、通常办公使用、个人使用和家庭使用，不授权在以下设计、开发和制造：使用中伴随着致命风险或危险，若不加以特别高度安全保障，有可能导致对公众产生危害，甚至直接死亡、人身伤害、严重物质损失或其他损失（例如：核设施的核反应控制、航空飞行控制、空中交通控制、公共交通控制、医用维系生命系统、核武器系统的导弹发射控制等）；需要极高可靠性的应用领域（比如海底中转器和人造卫星）。注意上述领域内对使用该产品引起的用户和/或第三方的任何索赔或者损失，江苏宏云技术有限公司不承担任何责任。

➤ 半导体器件存在一定的故障发生概率。请用户对器件和设备采取冗余设计、消防设计、过电流等防护措施，及其它异常操作防护措施等安全设计，保证即使半导体器件发生故障的情况下，也不会造成人身伤害、社会损害或者重大损失。