



KPM32R14TX

32 位 ARM Cortex-M4F 微控制器

用户手册



目录

1	简介	1
2	系统架构和存储器	2
2.1	概述	2
2.2	主要功能	2
2.3	模块框图	3
2.4	功能描述	5
2.4.1	地址空间映射	5
2.4.2	Bit 操作引擎	7
3	Flash 存储器	9
3.1	概述	9
3.2	主要功能	9
3.3	模块框图	10
3.4	功能描述	10
3.4.1	Flash 读延迟	11
3.4.2	Flash 烧写和擦除	11
3.4.3	Flash 中断号	11
3.5	寄存器	12
3.5.1	Flash 寄存器	12
3.5.2	ICACHE 寄存器	13
3.5.3	DCACHE 寄存器	15
4	时钟 (Clock)	16
4.1	概述	16



4.2	主要功能.....	16
4.3	模块框图.....	17
4.4	功能描述.....	20
4.4.1	HOSC 时钟	20
4.4.2	HIRC 时钟	20
4.4.3	HIRC2 时钟	21
4.4.4	PLL 时钟	21
4.4.5	LIRC 时钟.....	21
4.4.6	系统时钟组	22
4.4.7	与系统时钟组成异步关系的 IP 时钟	23
4.4.8	时钟监视器	24
4.4.9	频率监测器	25
4.4.10	低功耗.....	26
5	复位 (Reset)	27
5.1	概述	27
5.2	模块框图.....	28
5.3	功能描述.....	29
5.3.1	外部 RC 复位	29
5.3.2	上电复位 (POR)	29
5.3.3	低电压检测 (LVD).....	29
5.3.4	看门狗定时器 (WDT)	30
5.3.5	软件复位 (SYS)	30
5.3.6	锁定复位 (LOCKUP).....	30
5.3.7	复位域.....	30
5.3.8	IP 软件复位控制位	30



6	功耗管理 (Power).....	31
6.1	概述	31
6.2	主要功能.....	31
6.3	功能描述.....	31
6.3.1	电源供应.....	31
6.3.2	电源供应管理.....	31
6.3.3	不同模式下的模块状态	32
6.3.4	低功耗模式	34
7	系统寄存器 (System Register)	41
7.1	寄存器	41
7.1.1	芯片 ID 寄存器 (CHIP_ID).....	44
7.1.2	复位状态寄存器 (RST_ST).....	44
7.1.3	系统 NMI 状态寄存器 (NMI_SR).....	46
7.1.4	系统控制寄存器 (SYS_CFG)	47
7.1.5	功耗模式控制寄存器 (PWR_MOD)	48
7.1.6	系统配置密钥寄存器 (SYS_SET_KR).....	48
7.1.7	模拟 PMU 控制寄存器 (ANA_PMU_CTRL)	49
7.1.8	IRC 控制寄存器 0 (IRC_CTRL_REG0)	50
7.1.9	IRC 控制寄存器 1 (IRC_CTRL_REG1)	50
7.1.10	IRC 控制寄存器 2 (IRC_CTRL_REG2)	51
7.1.11	IRC 控制寄存器 3 (IRC_CTRL_REG3)	52
7.1.12	HOSC 控制寄存器 (HOSC_CTRL_REG)	52
7.1.13	PLL 控制寄存器 (PLL_CTRL_REG)	54
7.1.14	温度传感器控制寄存器 (TS_CTRL_REG).....	56
7.1.15	时钟频率监测寄存器 0 (CLK_FRQ_MNT_REG0)	56



7.1.16	时钟频率监控寄存器 1 (CLK_FRQ_MNT_REG1)	57
7.1.17	时钟频率监控寄存器 2 (CLK_FRQ_MNT_REG2)	58
7.1.18	时钟监控器寄存器 (CLK_MNT_REG).....	58
7.1.19	电压检测模块 1 控制寄存器 (VD1_CTRL)	59
7.1.20	电压检测模块 1 状态寄存器 (VD1_ST).....	60
7.1.21	电压检测模块 2 控制寄存器 (VD2_CTRL)	61
7.1.22	电压检测模块 2 状态寄存器 (VD2_ST).....	62
7.1.23	复位控制寄存器 0 (RST_CTRL_REG0).....	62
7.1.24	复位控制寄存器 1 (RST_CTRL_REG1).....	64
7.1.25	时钟门控寄存器 0 (CLK_GAT_REG0).....	65
7.1.26	时钟门控寄存器 1 (CLK_GAT_REG1).....	67
7.1.27	时钟选择寄存器 0 (CLK_SEL_REG0).....	68
7.1.28	时钟选择寄存器 1 (CLK_SEL_REG1).....	70
7.1.29	时钟选择寄存器 2 (CLK_SEL_REG2).....	70
7.1.30	噪声滤波时钟控制寄存器 (NF_CLK_CTRL_REG)	71
7.1.31	IO 噪声滤波控制寄存器 0 (IO_NF_CTRL_REG0)	72
7.1.32	IO 噪声滤波控制寄存器 1 (IO_NF_CTRL_REG1)	75
7.1.33	端口输入使能控制寄存器 0 (PAD_IE_CTRL_REG0)	79
7.1.34	端口输入使能控制寄存器 1 (PAD_IE_CTRL_REG1)	82
7.1.35	引脚驱动控制寄存器 0 (IO_DS_CTRL_REG0).....	85
7.1.36	引脚驱动控制寄存器 1 (IO_DS_CTRL_REG1).....	87
7.1.37	引脚上拉控制寄存器 0 (IO_PU_CTRL_REG0).....	88
7.1.38	引脚上拉控制寄存器 1 (IO_PU_CTRL_REG1).....	90
7.1.39	引脚下拉控制寄存器 0 (IO_PD_CTRL_REG0).....	91
7.1.40	引脚下拉控制寄存器 1 (IO_PD_CTRL_REG1).....	93



7.1.41	端口模式控制寄存器 0 (IO_MOD_CTRL_REG0)	95
7.1.42	端口模式控制寄存器 1 (IO_MOD_CTRL_REG1)	96
7.1.43	端口模式控制寄存器 2 (IO_MOD_CTRL_REG2)	97
7.1.44	端口模式控制寄存器 3 (IO_MOD_CTRL_REG3)	98
7.1.45	功能复用选择寄存器 0 (AF_SEL_REG0)	99
7.1.46	功能复用选择寄存器 1 (AF_SEL_REG1)	100
7.1.47	功能复用选择寄存器 2 (AF_SEL_REG2)	101
7.1.48	功能复用选择寄存器 3 (AF_SEL_REG3)	102
7.1.49	功能复用选择寄存器 4 (AF_SEL_REG4)	103
7.1.50	功能复用选择寄存器 5 (AF_SEL_REG5)	104
7.1.51	功能复用选择寄存器 6 (AF_SEL_REG6)	105
7.1.52	功能复用选择寄存器 7 (AF_SEL_REG7)	106
7.1.53	端口开漏输出控制寄存器 0 (IO_OD_CTRL_REG0)	107
7.1.54	端口开漏输出控制寄存器 1 (IO_OD_CTRL_REG1)	109
7.1.55	调试控制寄存器 (DBG_CTRL_REG).....	111
7.1.56	SRAM 状态寄存器 (SRAM_ST)	111
7.1.57	SRAM 调试信息寄存器 (SRAM_DBG_INFO)	112
8	通用 I/O (GPIO).....	113
8.1	概述	113
8.2	主要功能.....	113
8.3	模块框图.....	113
8.4	功能描述.....	114
8.4.1	GPIO 地址分布	114
8.4.2	GPIO 输出控制	114
8.4.3	响应 EBUS 事件	115



8.5	寄存器	115
8.5.1	GPIOx 输入数据寄存器 (GPIOx_DATA_IN)	116
8.5.2	GPIOx 输出数据寄存器 (GPIOx_DATA_OUT)	116
8.5.3	GPIOx 输出置位清零寄存器 (GPIOx_DOUT_SET_CLR)	116
8.5.4	GPIOx 输出翻转寄存器 (GPIOx_TOG)	117
8.5.5	GPIOx 输出使能寄存器 (GPIOx_DOUT_EN)	117
8.5.6	GPIOx 输出使能清除寄存器 (GPIOx_DOUT_EN_CLR)	118
8.5.7	GPIOx 事件控制寄存器 (GPIOx_EVT_CFG)	118
9	直接存储器访问 (DMA)	120
9.1	概述	120
9.2	主要功能	120
9.3	模块框图	120
9.4	功能描述	121
9.4.1	DMA 通道优先级	121
9.4.2	DMA 仲裁速率	122
9.4.3	DMA 传输类型	122
9.4.4	DMA 通道控制	123
9.4.5	DMA 通道选择	124
9.4.6	DMA 中断号	128
9.5	寄存器	128
9.5.1	DMA 控制寄存器 (DMA_CTRL)	130
9.5.2	DMA 地址保持寄存器 (DMA_ADR_KEEP)	130
9.5.3	DMA 通道软件请求寄存器 (DMA_CH_SW_REQ)	132
9.5.4	DMA 中断使能寄存器 (DMA_INT_EN)	133
9.5.5	DMA 中断标志寄存器 (DMA_INT_FLAG)	134



9.5.6	DMA 状态寄存器 (DMA_INT_FLAG).....	135
9.5.7	DMA 错误状态寄存器 (DMA_ERR_STATUS)	136
9.5.8	DMA 通道屏蔽请求置位寄存器 (DMA_REQ_MASK_SET)	136
9.5.9	DMA 通道屏蔽请求清零寄存器 (DMA_REQ_MASK_CLR).....	138
9.5.10	DMA 通道使能置位寄存器 (DMA_CH_EN_SET).....	139
9.5.11	DMA 通道使能清零寄存器 (DMA_CH_EN_CLR)	140
9.5.12	DMA 通道优先级置位寄存器 (DMA_CH_PRT_SET).....	142
9.5.13	DMA 通道优先级清零寄存器 (DMA_CH_PRT_CLR).....	143
9.5.14	DMA 通道选择寄存器 (DMA_CH_SEL)	144
9.5.15	DMA EBUS 控制寄存器 (DMA_EBUS_CTRL).....	145
9.5.16	DMA 通道 0 源地址寄存器 (DMA_CH0_SRC).....	148
9.5.17	DMA 通道 0 目标地址寄存器 (DMA_CH0_DES)	148
9.5.18	DMA 通道 0 控制寄存器 (DMA_CH0_CTRL).....	148
10	循环冗余校验 (CRC).....	150
10.1	概述	150
10.2	主要功能.....	150
10.3	模块框图.....	150
10.4	功能描述.....	151
10.5	寄存器	151
10.5.1	CRC 控制寄存器 (CRC_CTRL).....	151
10.5.2	CRC 数据寄存器 (CRC_DATA).....	152
11	三角函数硬件加速器 (TMU)	153
11.1	概述	153
11.2	主要功能.....	153
11.3	模块框图	154



11.4	功能描述.....	154
11.4.1	Cordic 算法.....	154
11.4.2	单精度浮点数据类型	157
11.4.3	TMU 工作时序.....	157
11.5	寄存器	159
11.5.1	TMU 控制寄存器 (TMU_CTRL).....	160
11.5.2	TMU 角度浮点寄存器 (TMU_ANGLE).....	160
11.5.3	TMU 余弦浮点寄存器 (TMU_COSX).....	160
11.5.4	TMU 正弦浮点寄存器 (TMU_SINY).....	161
11.5.5	TMU 角度定点寄存器 (TMU_ANGLE_B)	161
11.5.6	TMU 误差定点寄存器 (TMU_ERR_B)	161
11.5.7	TMU 余弦定点寄存器 (TMU_COSX_B).....	161
11.5.8	TMU 正弦定点寄存器 (TMU_SINY_B).....	162
11.5.9	TMU 状态寄存器 (TMU_ST).....	162
12	模/数转换器 (ADC).....	163
12.1	概述	163
12.2	主要功能.....	163
12.3	模块框图.....	164
12.4	功能描述.....	165
12.4.1	相关配置	165
12.4.2	转换顺序优先级	165
12.4.3	突发转换模式.....	168
12.4.4	DMA 读取功能	171
12.4.5	硬件触发功能.....	171
12.4.6	零点偏移矫正.....	171

12.4.7	结果后处理	171
12.4.8	ADC 中断号	173
12.5	寄存器	173
12.5.1	ADC 软件触发寄存器 (ADC_TRIG)	176
12.5.2	ADC 控制寄存器 (ADC_CTRL)	178
12.5.3	ADC 转换单元优先级寄存器 (ADC_SOC_PRI)	179
12.5.4	ADC 中断使能寄存器 (ADC_INTEN)	180
12.5.5	ADC 中断状态寄存器 (ADC_STA)	184
12.5.6	ADC 中断状态清除寄存器 (ADC_STA_CLR)	187
12.5.7	ADC 转换单元触发溢出寄存器 (ADC_TRIG_OVF)	190
12.5.8	ADC 转换单元触发溢出清除寄存器 (ADC_OVF_CLR)	192
12.5.9	ADC 转换单元配置寄存器 (ADC_SOCx) (x=0,1,2...15)	194
12.5.10	ADC PPB0 设置寄存器 (ADC_PPB0_CFG)	196
12.5.11	ADC PPB0 偏移校正寄存器 (ADC_PPB0_OFFCAL)	197
12.5.12	ADC PPB0 偏移参考寄存器 (ADC_PPB0_OFFREF)	197
12.5.13	ADC PPB0 比较上限寄存器 (ADC_PPB0_COMPH)	198
12.5.14	ADC PPB0 比较下限寄存器 (ADC_PPB0_COMPL)	198
12.5.15	ADC PPB1 设置寄存器 (ADC_PPB1_CFG)	199
12.5.16	ADC PPB1 偏移校正寄存器 (ADC_PPB1_OFFCAL)	199
12.5.17	ADC PPB1 偏移参考寄存器 (ADC_PPB1_OFFREF)	200
12.5.18	ADC PPB1 比较上限寄存器 (ADC_PPB1_COMPH)	201
12.5.19	ADC PPB1 比较下限寄存器 (ADC_PPB1_COMPL)	201
12.5.20	ADC PPB2 设置寄存器 (ADC_PPB2_CFG)	202
12.5.21	ADC PPB2 偏移校正寄存器 (ADC_PPB2_OFFCAL)	202
12.5.22	ADC PPB2 偏移参考寄存器 (ADC_PPB2_OFFREF)	203

12.5.23	ADC PPB2 比较上限寄存器 (ADC_PPB2_COMPH).....	204
12.5.24	ADC PPB2 比较下限寄存器 (ADC_PPB2_COMPL).....	204
12.5.25	ADC PPB3 设置寄存器 (ADC_PPB3_CFG).....	205
12.5.26	ADC PPB3 偏移校正寄存器 (ADC_PPB3_OFFCAL)	205
12.5.27	ADC PPB3 偏移参考寄存器 (ADC_PPB3_OFFREF)	206
12.5.28	ADC PPB3 比较上限寄存器 (ADC_PPB3_COMPH).....	206
12.5.29	ADC PPB3 比较下限寄存器 (ADC_PPB3_COMPL).....	207
12.5.30	ADC 转换结果寄存器 (ADC_RESULTx) (x=0,1,2...15).....	207
12.5.31	ADC PPB0 转换结果寄存器 (ADC_PPB0_RESULT).....	208
12.5.32	ADC PPB1 转换结果寄存器 (ADC_PPB1_RESULT).....	208
12.5.33	ADC PPB2 转换结果寄存器 (ADC_PPB2_RESULT).....	208
12.5.34	ADC PPB3 转换结果寄存器 (ADC_PPB3_RESULT).....	208
12.5.35	ADC 模拟通道使能寄存器 (ADC_CHNEN).....	209
12.5.36	ADC 偏移校正寄存器 (ADC_OFFCAL)	211
12.5.37	ADC 转换器配置寄存器 (ADC_ANA_CFG).....	211
13	简版定时器 (TIMS).....	213
13.1	概述	213
13.2	主要功能.....	213
13.3	模块框图.....	214
13.4	功能描述.....	215
13.4.1	计数器介绍	215
13.4.2	计数器模式.....	216
13.4.3	时钟选择	223
13.4.4	捕获/比较通道.....	224
13.4.5	输入捕获模式.....	226



13.4.6	PWM 输入模式	227
13.4.7	强制输出模式.....	228
13.4.8	输出比较模式.....	228
13.4.9	PWM 模式	229
13.4.10	单脉冲模式.....	230
13.4.11	互补输出和死区插入	232
13.4.12	生成 6 步 PWM	233
13.4.13	使用断路功能.....	234
13.4.14	定时器与外部触发同步	238
13.4.15	定时器协同操作	240
13.4.16	定时器同步	241
13.4.17	TIMS 中断号	244
13.5	寄存器	244
13.5.1	TIM 控制寄存器 (TIM_CR).....	245
13.5.2	TIM 模式控制寄存器 (TIM_MCR)	248
13.5.3	TIM 捕获/比较模式寄存器 (TIM_CCMR)	250
13.5.4	TIM 捕获/比较模式使能寄存器 (TIM_CCER)	254
13.5.5	TIM 计数器 (TIM_CNT).....	255
13.5.6	TIM 预分频器 (TIM_PSC).....	256
13.5.7	TIM 自动重载寄存器 (TIM_ARR).....	256
13.5.8	TIM 重复计数器寄存器 (TIM_RCR).....	256
13.5.9	TIM 捕获/比较寄存器 1 (TIM_CCR1).....	257
13.5.10	TIM 捕获/比较寄存器 2 (TIM_CCR2).....	257
13.5.11	TIM 断路及死区寄存器 (TIM_BDTR)	258
13.5.12	TIM 配置密钥寄存器 (TIM_SET_KR).....	259



13.5.13	TIM 事件生成寄存器 (TIM_EGR)	260
13.5.14	TIM 中断使能寄存器 (TIM_IER)	261
13.5.15	TIM 状态寄存器 (TIM_SR)	262
14	通用定时器 (TIMG)	264
14.1	概述	264
14.2	主要功能	264
14.3	模块框图	265
14.4	功能描述	266
14.4.1	计数器介绍	266
14.4.2	计数器模式	266
14.4.3	时钟选择	273
14.4.4	捕获/比较通道	276
14.4.5	输入捕获模式	278
14.4.6	PWM 输入模式	279
14.4.7	强制输出模式	280
14.4.8	输出比较模式	280
14.4.9	PWM 模式	281
14.4.10	发生外部事件时清除 OCxREF 信号	284
14.4.11	单脉冲模式	285
14.4.12	编码器接口模式	286
14.4.13	定时器输入异或功能	292
14.4.14	定时器与外部触发同步	292
14.4.15	定时器同步	294
14.4.16	TIMG 中断号	295
14.5	寄存器	296



14.5.1	TIM 控制寄存器 0 (TIM_CR0)	297
14.5.2	TIM 控制寄存器 1 (TIM_CR1)	299
14.5.3	TIM 从属模式控制寄存器 (TIM_SMCR)	300
14.5.4	TIM DMA&中断使能寄存器 (TIM_DIER)	303
14.5.5	TIM 状态寄存器 (TIM_SR)	304
14.5.6	TIM 事件生成寄存器 (TIM_EGR)	306
14.5.7	TIM 捕获/比较模式寄存器 0 (TIM_CCMR0)	307
14.5.8	TIM 捕获/比较模式寄存器 1 (TIM_CCMR1)	311
14.5.9	TIM 捕获/比较模式使能寄存器 (TIM_CCER)	315
14.5.10	TIM 计数器 (TIM_CNT)	317
14.5.11	TIM 预分频器 (TIM_PSC)	317
14.5.12	TIM 自动重载寄存器 (TIM_ARR)	317
14.5.13	TIM 捕获/比较寄存器 0 (TIM_CCR0)	318
14.5.14	TIM 捕获/比较寄存器 1 (TIM_CCR1)	318
14.5.15	TIM 捕获/比较寄存器 2 (TIM_CCR2)	319
14.5.16	TIM 捕获/比较寄存器 3 (TIM_CCR3)	319
14.5.17	TIM 配置密钥寄存器 (TIM_SET_KR)	320
15	高级定时器 (TIMA)	321
15.1	概述	321
15.2	主要功能	321
15.3	模块框图	322
15.4	功能描述	323
15.4.1	计数器介绍	323
15.4.2	计数器模式	324
15.4.3	时钟选择	332



15.4.4	捕获/比较通道	335
15.4.5	输入捕获模式	336
15.4.6	PWM 输入模式	337
15.4.7	强制输出模式	338
15.4.8	输出比较模式	338
15.4.9	PWM 模式	339
15.4.10	互补输出和死区插入	344
15.4.11	使用断路功能	346
15.4.12	发生外部事件时清除 OCxREF 信号	349
15.4.13	生成 6 步 PWM	350
15.4.14	单脉冲模式	351
15.4.15	编码器接口模式	353
15.4.16	定时器输入异或功能	355
15.4.17	连接霍尔传感器	355
15.4.18	定时器与外部触发同步	356
15.4.19	定时器同步	359
15.4.20	TIMA 中断号	359
15.5	寄存器	360
15.5.1	TIM 控制寄存器 0 (TIM_CR0)	361
15.5.2	TIM 控制寄存器 1 (TIM_CR1)	362
15.5.3	TIM 从属模式控制寄存器 (TIM_SMCR)	366
15.5.4	TIM DMA&中断使能寄存器 (TIM_DIER)	368
15.5.5	TIM 状态寄存器 (TIM_SR)	371
15.5.6	TIM 事件生成寄存器 (TIM_EGR)	373
15.5.7	TIM 捕获/比较模式寄存器 0 (TIM_CCMR0)	375

15.5.8	TIM 捕获/比较模式寄存器 1 (TIM_CCMR1)	379
15.5.9	TIM 捕获/比较模式寄存器 2 (TIM_CCMR2)	383
15.5.10	TIM 捕获/比较模式使能寄存器 (TIM_CCER)	386
15.5.11	TIM 计数器 (TIM_CNT).....	388
15.5.12	TIM 预分频器 (TIM_PSC).....	389
15.5.13	TIM 自动重载寄存器 (TIM_ARR).....	389
15.5.14	TIM 重复计数器寄存器 (TIM_RCR).....	389
15.5.15	TIM 捕获/比较寄存器 0 (TIM_CCR0).....	390
15.5.16	TIM 捕获/比较寄存器 1 (TIM_CCR1).....	390
15.5.17	TIM 捕获/比较寄存器 2 (TIM_CCR2).....	391
15.5.18	TIM 捕获/比较寄存器 3 (TIM_CCR3).....	391
15.5.19	TIM 捕获/比较寄存器 4 (TIM_CCR4).....	392
15.5.20	TIM 捕获/比较寄存器 5 (TIM_CCR5).....	392
15.5.21	TIM 断路及死区寄存器 (TIM_BDTR).....	393
15.5.22	TIM 事件控制寄存器 (TIM_ECR)	395
15.5.23	TIM 配置密钥寄存器 (TIM_SET_KR).....	396
16	定时器选项 (TIMOR).....	397
16.1	概述.....	397
16.2	主要功能.....	397
16.3	模块框图.....	398
16.4	功能描述.....	399
16.4.1	断路输入控制.....	399
16.4.2	内部触发管理.....	399
16.4.3	通道输入管理.....	401
16.4.4	ETR 输入管理	401

16.4.5	OCREF_CLR 输入管理.....	402
16.5	寄存器.....	402
16.5.1	TIMA0 输入复用控制寄存器 (TIMA0_IMCR)	403
16.5.2	TIMG0 输入复用控制寄存器 (TIMG0_IMCR).....	404
16.5.3	TIMG1 输入复用控制寄存器 (TIMG1_IMCR).....	405
16.5.4	TIMG2 输入复用控制寄存器 (TIMG2_IMCR).....	406
16.5.5	TIMS 输入复用控制寄存器 (TIMS_IMCR).....	407
16.5.6	TIMA 内部触发控制寄存器 (TIMA_ITCR).....	408
16.5.7	TIMG 内部触发控制寄存器 (TIMG_ITCR).....	409
16.5.8	TIMS 内部触发控制寄存器 (TIMS_ITCR).....	410
16.5.9	TIMA0 断路控制寄存器 (TIMA0_SBCR).....	412
16.5.10	TIMA1 断路控制寄存器 (TIMA1_SBCR).....	413
16.5.11	TIMH 断路控制寄存器 (TIMH_SBCR).....	414
16.5.12	TIMS0 断路控制寄存器 (TIMS0_SBCR).....	414
16.5.13	TIMS1 断路控制寄存器 (TIMS1_SBCR).....	415
16.5.14	TIMS2 断路控制寄存器 (TIMS2_SBCR).....	416
16.5.15	TIMS3 断路控制寄存器 (TIMS3_SBCR).....	417
16.5.16	TIMA 外部触发控制寄存器 0 (TIMA_ETCR0)	418
16.5.17	TIMA OCREF 清除控制寄存器 (TIMA_OCCR).....	418
16.5.18	TIMG OCREF 清除控制寄存器 (TIMG_OCCR)	419
16.5.19	TIMS OCREF 清除控制寄存器 (TIMS_OCCR).....	419
17	高精度定时器 (TIMH)	421
17.1	概述.....	421
17.2	主要功能.....	421
17.3	模块框图.....	422



17.4	功能描述.....	424
17.4.1	PWM 时钟	424
17.4.2	时钟同步	428
17.4.3	最小 PWM 周期和脉宽	428
17.4.4	PWM 发生器 (PG) 特性	428
17.4.5	通用特性	464
17.4.6	寄存器锁定和写入限制	469
17.4.7	TIMH 中断号	470
17.5	寄存器	470
17.5.1	通用功能寄存器	470
17.5.2	PWM 发生器寄存器	478
18	基准定时器 (BTM).....	496
18.1	概述	496
18.2	主要功能.....	496
18.3	模块框图.....	496
18.4	功能描述.....	497
18.4.1	基本功能	497
18.4.2	时钟源.....	497
18.4.3	可编程中断	497
18.4.4	输出触发事件至 EBUS 系统.....	497
18.4.5	BTM 中断号	497
18.5	寄存器	498
18.5.1	TM0 控制寄存器 (TM0_CTRL)	498
18.5.2	TM0 加载寄存器 (TM0_LD)	499
18.5.3	TM0 计数器寄存器 (TM0_CNT).....	500



18.5.4	TM1 控制寄存器 (TM1_CTRL)	500
18.5.5	TM1 加载寄存器 (TM1_LD)	501
18.5.6	TM1 计数器寄存器 (TM1_CNT)	501
18.5.7	TM 状态寄存器 (TM_ST)	502
19	看门狗 (WDT)	503
19.1	概述	503
19.2	主要功能	503
19.3	模块框图	504
19.4	功能描述	504
19.4.1	预警中断功能	504
19.4.2	工作模式	505
19.4.3	普通模式 1 (不带预警功能)	505
19.4.4	普通模式 2 (带预警功能)	505
19.4.5	窗口模式 (不带预警功能)	506
19.4.6	窗口模式 (带预警功能)	506
19.4.7	寄存器安全锁功能	507
19.4.8	硬件看门狗	507
19.4.9	调试模式	507
19.4.10	时钟源	507
19.5	寄存器	507
19.5.1	WDT 控制寄存器 (WDT_CTRL)	508
19.5.2	WDT 加载寄存器 (WDT_LOAD)	509
19.5.3	WDT 窗口加载寄存器 (WDT_WIN_LOAD)	509
19.5.4	WDT 状态寄存器 (WDT_ST)	510
19.5.5	WDT 计数器寄存器 (WDT_VAL)	510



19.5.6	WDT 秘钥寄存器 (WDT_KR).....	511
20	串行通信接口 (SCI).....	512
20.1	概述	512
20.2	主要功能.....	512
20.3	模块框图.....	514
20.4	功能描述.....	515
20.4.1	工作模式	515
20.4.2	接口功能	515
20.4.3	SPI 模式.....	516
20.4.4	UART 通信模式.....	522
20.4.5	简易 I ² C 通信模式.....	527
20.4.6	SYNC 通信模式.....	529
20.4.7	Smart Card 通信模式	533
20.4.8	SCI 中断号	536
20.5	寄存器	537
20.5.1	时钟分频选择寄存器 (SCI_CLK_PRS)	537
20.5.2	波特率设置寄存器 (SCI_BAUD).....	538
20.5.3	模式控制寄存器 (SCI_MOD_CTL).....	538
20.5.4	UART 模式寄存器 (UART_MOD_CTL)	540
20.5.5	SPI 模式寄存器 (SPI_MOD_CTL)	541
20.5.6	IIC 模式寄存器 (IIC_MOD_CTL)	542
20.5.7	SMART CARD 模式寄存器 (SMC_MOD_CTL)	543
20.5.8	SYNC 模式寄存器 (SYNC_MOD_CTL).....	544
20.5.9	发送数据寄存器 (SCI_TX_DAT)	545
20.5.10	接收数据寄存器 (SCI_RX_DAT).....	545



20.5.11	中断使能寄存器 (SCI_IE)	545
20.5.12	中断状态寄存器 (SCI_STA).....	547
20.5.13	中断状态清除寄存器 (SCI_STA_CLR)	548
21	I ² C 总线 (I2CS).....	550
21.1	概述	550
21.2	主要功能.....	550
21.3	模块框图.....	551
21.4	功能描述.....	552
21.4.1	I ² C 串行接口功能.....	552
21.4.2	I ² C 总线模式功能.....	552
21.4.3	I ² C 总线定义及控制方法.....	553
21.4.4	I2CS 中断号	557
21.5	寄存器	558
21.5.1	I2CS 数据寄存器 (I2CS_DATA).....	559
21.5.2	I2CS 从机地址寄存器 (I2CS_SLAD)	559
21.5.3	I2CS 设备地址寄存器 (I2CS_DVAD).....	559
21.5.4	I2CS 移位寄存器 (I2CS_SHF).....	560
21.5.5	I2CS 控制寄存器 (I2C_CFG)	560
21.5.6	I2CS 时钟分频寄存器 (I2CS_CLK_DIV)	562
21.5.7	I2CS 电平宽度寄存器 (I2CS_WTH).....	562
21.5.8	I2CS 输出控制寄存器 (I2CS_OUT_CTRL).....	563
21.5.9	I2CS FIFO 控制寄存器 (I2CS_FIFO_CTRL).....	563
21.5.10	I2CS FIFO 状态寄存器 (I2CS_FIFO_ST)	564
21.5.11	I2CS 数据个数寄存器 (I2CS_DATA_CNT)	565
21.5.12	I2CS SDA 延迟寄存器 (I2CS_SDA_DLY).....	565



21.5.13	I2CS 状态寄存器 (I2CS_ST)	565
21.5.14	I2CS 中断控制寄存器 (I2C_INT_EN)	567
21.5.15	I2CS 中断状态寄存器 (I2C_INT_ST)	568
21.5.16	I2CS 超时控制寄存器 (I2CS_TMOUT_CFG)	569
21.5.17	I2CS 超时阈值寄存器 (I2CS_TMOUT_CNT)	570
22	串口总线 (SPI)	571
22.1	概述	571
22.2	主要功能	571
22.3	模块框图	572
22.4	功能描述	572
22.4.1	SPI 接口功能	572
22.4.2	SPI 传输模式	573
22.4.3	SPI 工作模式	574
22.4.4	SPI 通信波特率	575
22.4.5	SPI 中断号	576
22.5	寄存器	576
22.5.1	SPI 数据寄存器 (SPI_DATA)	577
22.5.2	SPI 控制寄存器 (SPI_CFG)	577
22.5.3	SPI 帧格式寄存器 (SPI_FRM_FMT)	578
22.5.4	SPI 帧长度寄存器 (SPI_FRM_WIDTH)	579
22.5.5	SPI 通信波特率寄存器 (SPI_BAUD)	579
22.5.6	SPI 片选控制寄存器 (SPI_SSLV_CTL)	579
22.5.7	SPI 片选时序寄存器 (SPI_SS_TIM)	580
22.5.8	SPI 时钟控制寄存器 (SPI_SCK_CTL)	581
22.5.9	SPI FIFO 控制寄存器 (SPI_FIFO_CTL)	582



22.5.10	SPI FIFO 状态寄存器 (SPI_FIFO_ST).....	583
22.5.11	SPI 状态寄存器 (SPI_ST).....	583
22.5.12	SPI DMA 发送寄存器 (SPI_DMA_TX).....	584
22.5.13	SPI DMA 接收寄存器 (SPI_DMA_RX).....	584
22.5.14	SPI 中断控制寄存器 (SPI_INT_CTL).....	586
22.5.15	SPI 中断状态寄存器 (SPI_INT_ST).....	587
23	模拟比较器 (CMP).....	589
23.1	概述.....	589
23.2	主要功能.....	589
23.3	模块框图.....	590
23.4	功能描述.....	590
23.4.1	CMP 工作时序.....	590
23.4.2	CMP 数字滤波.....	591
23.4.3	DAC 电压输出.....	591
23.4.4	DAC 斜坡发生.....	591
23.4.5	CMP 中断号.....	591
23.5	寄存器.....	592
23.5.1	CMPx 控制寄存器 (CMPx_CTRL).....	592
23.5.2	CMPx P/N 输入选择寄存器 (CMPx_PN_SEL).....	594
23.5.3	CMPx 滤波控制寄存器 (CMPx_FLT_CTRL).....	595
23.5.4	CMPx 状态寄存器 (CMPx_ST).....	596
23.5.5	DACx 控制寄存器 (DACx_CTRL).....	597
23.5.6	DACx 数据寄存器 (DACx_VAL).....	598
23.5.7	DACx 翻转数值寄存器 (DACx_SDAT).....	598
23.5.8	DACx 斜坡延迟寄存器 (DACx_RAMP_DLY).....	599



23.5.9	DACx 斜坡步长寄存器 (DACx_RAMP_STEP)	599
23.5.10	DACx 斜坡保持寄存器 (DACx_RAMP_SLOPE)	600
23.5.11	CMPx 输入 IO 选择 (CMPx_ASEL)	600
24	外设事件系统 (EBUS).....	601
24.1	概述	601
24.2	主要功能.....	601
24.3	模块框图.....	602
24.4	功能描述.....	602
24.4.1	通道工作模式.....	602
24.4.2	同步输出和输入边沿检测.....	602
24.4.3	事件延时	603
24.4.4	软件事件	603
24.4.5	组合逻辑功能.....	603
24.4.6	事件触发 CPU.....	603
24.4.7	IO 事件检测.....	603
24.4.8	通道事件输入源选择	604
24.4.9	通道事件输出从设备	609
24.4.10	EBUS 中断号	609
24.5	寄存器	610
24.5.1	EBUS 外设 ID 寄存器 (EBUS_ID).....	611
24.5.2	EBUS 软件事件触发寄存器 (EBUS_SOFT_EVT)	611
24.5.3	EBUS 事件触发 CPU 设置寄存器 (EBUS_EVT_CPU)	614
24.5.4	EBUS IO 边沿检测设置寄存器 (EBUS_IO_DET)	615
24.5.5	EBUS IO 检测中断状态寄存器 (EBUS_IO_STA).....	617
24.5.6	EBUS IO 检测中断状态清除寄存器 (IO_INT_STACLR).....	619



24.5.7	EBUS IO 源选择寄存器 1 (IO_SRC_SEL1)	621
24.5.8	EBUS IO 源选择寄存器 2 (IO_SRC_SEL2)	622
24.5.9	EBUS 通道 0 设置寄存器 (EBUS_CHN0_CFG)	624
24.5.10	EBUS 通道 1 设置寄存器 (EBUS_CHN1_CFG)	626
24.5.11	EBUS 通道 2 设置寄存器 (EBUS_CHN2_CFG)	627
24.5.12	EBUS 通道 3 设置寄存器 (EBUS_CHN3_CFG)	628
24.5.13	EBUS 通道 4 设置寄存器 (EBUS_CHN4_CFG)	629
24.5.14	EBUS 通道 5 设置寄存器 (EBUS_CHN5_CFG)	630
24.5.15	EBUS 通道 6 设置寄存器 (EBUS_CHN6_CFG)	631
24.5.16	EBUS 通道 7 设置寄存器 (EBUS_CHN7_CFG)	632
24.5.17	EBUS 通道 8 设置寄存器 (EBUS_CHN8_CFG)	633
24.5.18	EBUS 通道 9 设置寄存器 (EBUS_CHN9_CFG)	634
24.5.19	EBUS 通道 10 设置寄存器 (EBUS_CHN10_CFG)	635
24.5.20	EBUS 通道 11 设置寄存器 (EBUS_CHN11_CFG)	636
24.5.21	EBUS 通道 12 设置寄存器 (EBUS_CHN12_CFG)	637
24.5.22	EBUS 通道 13 设置寄存器 (EBUS_CHN13_CFG)	638
24.5.23	EBUS 通道 14 设置寄存器 (EBUS_CHN14_CFG)	639
24.5.24	EBUS 通道 15 设置寄存器 (EBUS_CHN15_CFG)	640
25	控制器局域网 (CAN).....	641
25.1	概述	641
25.2	主要功能.....	641
25.3	模块框图.....	642
25.4	功能描述.....	642
25.4.1	CAN 简述	642
25.4.2	操作模式	642



25.4.3	信息发送	643
25.4.4	信息接收	643
25.4.5	接收滤波	644
25.4.6	波特率.....	644
25.4.7	总线仲裁	645
25.4.8	错误处理	645
25.4.9	睡眠模式	645
25.4.10	CAN 中断号	646
25.5	寄存器	646
25.5.1	CAN 控制寄存器 (CAN_CTRL).....	647
25.5.2	CAN 配置寄存器 (CAN_CFG).....	648
25.5.3	CAN 中断使能寄存器 (CAN_IE).....	648
25.5.4	CAN 总线时序寄存器 (CAN_BUS_TIM).....	649
25.5.5	CAN 验收标识符 0 (CAN_ACPT_CODE0).....	651
25.5.6	CAN 验收屏蔽 0 (CAN_ACPT_MASK0).....	651
25.5.7	CAN 验收标识符 1 (CAN_ACPT_CODE1).....	651
25.5.8	CAN 验收屏蔽 1 (CAN_ACPT_MASK1).....	652
25.5.9	CAN 错误预警寄存器 (CAN_ERR_WARN_THR).....	652
25.5.10	CAN 输出控制寄存器 (CAN_OUT_CTRL).....	652
25.5.11	CAN 状态寄存器 (CAN_ST)	653
25.5.12	CAN 中断寄存器 (CAN_INT)	654
25.5.13	CAN 接收缓冲器状态 (RX_BUF_ST).....	655
25.5.14	CAN 发送帧信息寄存器 (TX_FRM_INFO).....	656
25.5.15	CAN 发送标识符 (TX_ID)	657
25.5.16	CAN 发送数据寄存器 0 (TX_DATA0).....	657

25.5.17	CAN 发送数据寄存器 1 (TX_DATA1).....	657
25.5.18	CAN 接收帧信息寄存器 (RX_FRM_INFO)	658
25.5.19	CAN 接收标识符 (RX_ID).....	658
25.5.20	CAN 接收数据寄存器 0 (RX_DATA0)	659
25.5.21	CAN 接收数据寄存器 1 (RX_DATA1)	659
25.5.22	CAN 仲裁丢失位 (CAN_ARBIT_LOST_BIT).....	659
25.5.23	CAN 错误信息寄存器 (RX_ERR_INFO)	663
25.5.24	CAN 错误计数寄存器 (ERR_CNT).....	664
25.5.25	CAN 接收缓冲器 (RX_FIFO).....	665
26	键入中断 (IOW)	665
26.1	概述	665
26.2	主要功能.....	665
26.3	模块框图.....	666
26.4	功能描述.....	666
26.4.1	通道输入源	666
26.4.2	IOW 中断号.....	666
26.5	寄存器	667
26.5.1	键入中断中断使能寄存器 (IOW_INTEN)	667
26.5.2	键入中断控制寄存器 (IOW_CTRL)	667
26.5.3	键入中断边沿选择寄存器 (IOW_EDGES).....	668
26.5.4	键入中断状态寄存器 (IOW_STA).....	669
27	低压检测 (LVD)	671
27.1	概述	671
27.2	主要功能.....	671
27.3	模块框图.....	673



27.4	功能描述.....	673
27.4.1	LVD 档位.....	673
27.4.2	LVD 功能.....	674
27.4.3	LVD 中断号.....	675
27.5	寄存器.....	675
27.5.1	电压检测模块 1 控制寄存器 (VD1_CTRL).....	675
27.5.2	电压检测模块 1 状态寄存器 (VD1_ST).....	677
27.5.3	电压检测模块 2 控制寄存器 (VD2_CTRL).....	677
27.5.4	电压检测模块 2 状态寄存器 (VD2_ST).....	679
28	嵌套中断向量控制器 (NVIC).....	680
29	附录.....	682
29.1	寄存器相关缩写词列表.....	682

1 简介

本产品采用高性能 ARM Cortex-M4F 32 位处理器，最高频率达到 100MHz，支持 FPU、MPU，性能高达 125DMIPS，内嵌存储容量达到 256KB 高速 Flash 存储器，支持 1KB 的指令 cache 和 512B 数据 cache，可实现 CPU 的零等待时间运行；芯片集成专用高性能 TMU（三角函数加速器），支持硬件 $\sin/\cos/\atan$ 等三角函数的浮点计算；内嵌 32KB SRAM 存储器，并且支持 SRAM 奇偶校验；芯片集成了丰富的 I/O 端口及多种功能外设。集成 2 个采样率高达 2.4Msps 的 12 位 ADC、2 个高精度高频 8MHz 时钟（全温度范围内达到 $\pm 1\%$ 精度）和 1 个 32kHz 低频时钟、1 个 100MHz PLL、4 个 DAC 和 CMP、1 个高精度定时器（最大分辨率 156ps，支持 3 对互补通道），2 个支持多通道互补输出的高级定时器（TIMA）、3 个多通道 PWM 输出支持正交编码和霍尔传感器的通用定时器（TIMG）、4 个单通道 PWM 输出的简单定时器（TIMS）、1 个基准定时器，此外还包含多个标准通信接口：3 个 SCI（支持 UART/SPI/简易 I²C）、2 个标准 I²C（支持主从模式和高速 I²C）、1 个标准 SPI 及 1 个标准 CAN 模块。

本产品供电电压为 2.4V ~ 5.5V，覆盖 -40°C 至 +105°C 宽温度范围，提供多种省电模式保证低功耗应用的要求，具有高可靠性、高整合度和高抗干扰性。

本产品提供多种封装形式，可以应用在多种应用场合。

- 工业应用
- 家电控制
- 电机驱动和应用控制
- EV 充电基础设施
- 可再生能源存储
- 混合动力、电动和动力总成系统
- 交流逆变器和变频驱动器
- 工业电源

2 系统架构和存储器

2.1 概述

系统总线由 AHB 和 APB 总线构成，支持 4 个 Master 和 6 个 Slave。

2.2 主要功能

- 支持 32 位 AHB-lite 总线
- 支持 32 位 APB 总线
- 支持 4 个 Master
 - Corex-M4F I-bus
 - Corex-M4F D-bus
 - Corex-M4F S-bus
 - DMA
- 支持 6 个 Slave
 - Flash 存储器 ICODE
 - Flash 存储器 DCODE
 - SRAM 存储器
 - AHB0, AHB to APB 总线，包含部分 APB 接口外设 (低速外设)
 - AHB1, 包含 GPIO、SYSROM、DMA 从属端、ADC、TMU、CRC 等 AHB 接口外设， AHB to APB 总线，包括部分 APB 内部 IP
 - AHB2, AHB to APB 总线，包含部分 APB 接口外设 (TIM 部分)
- 支持地址重映射

2.3 模块框图

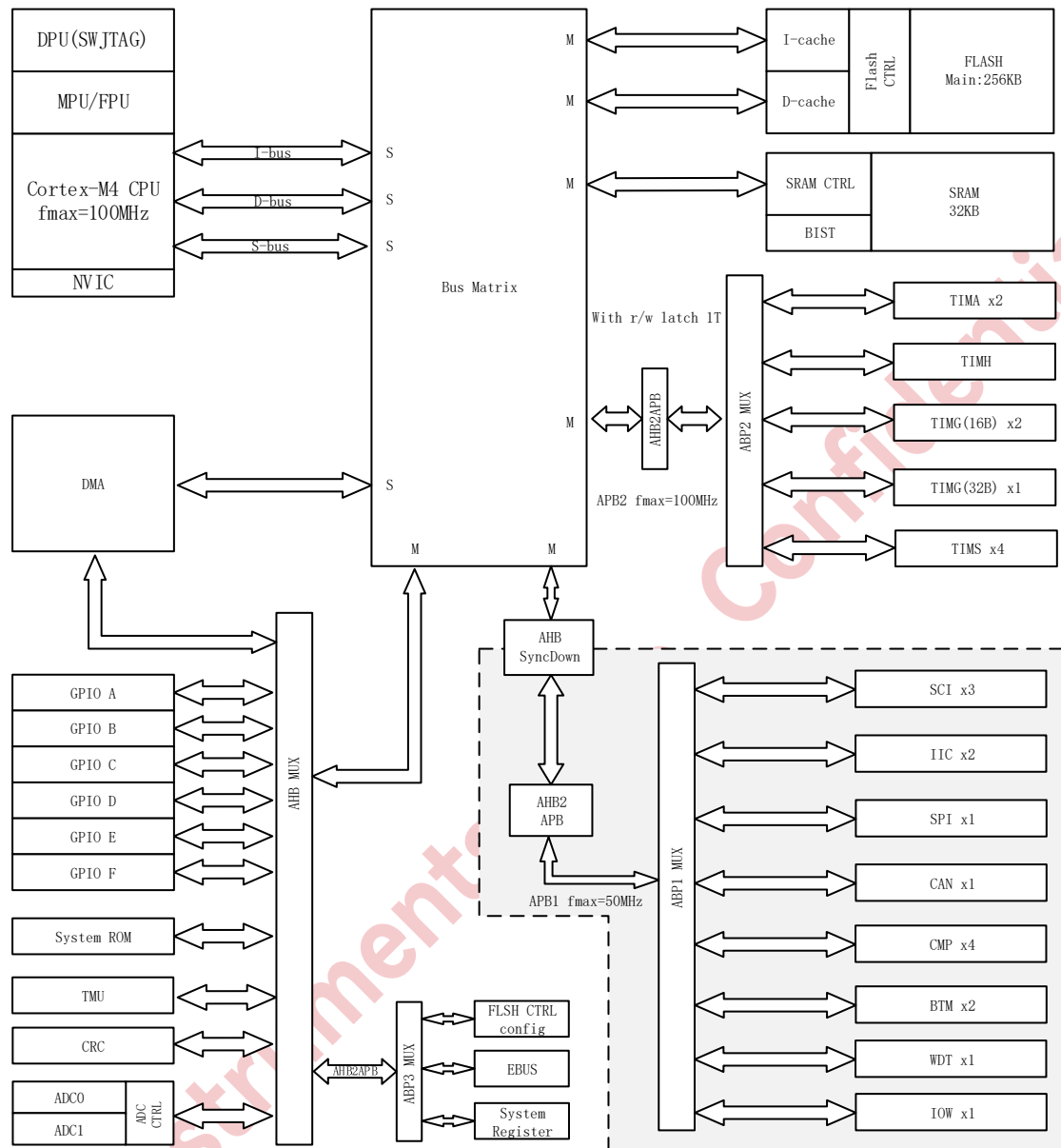


图 1 系统框图

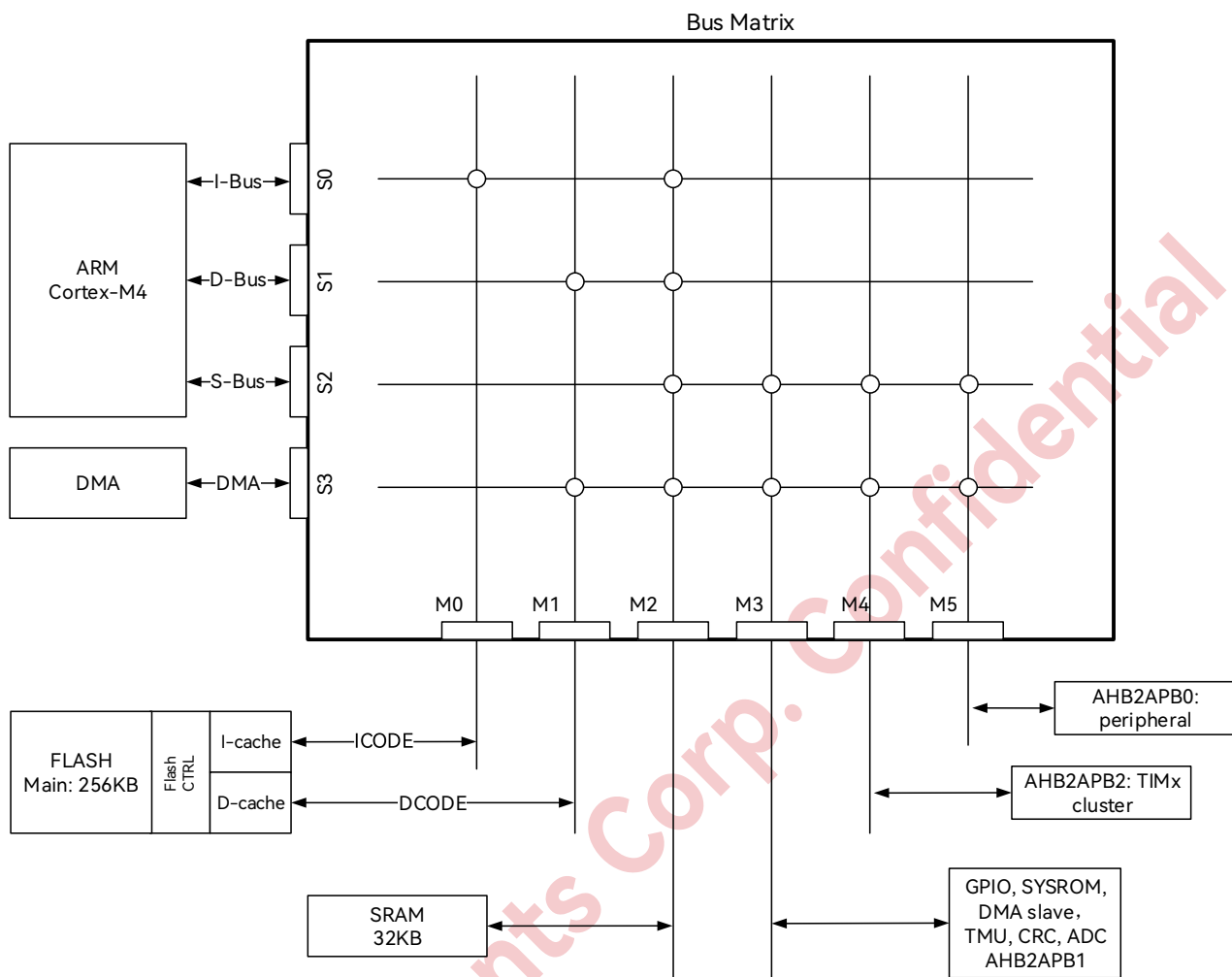


图 2 总线连接框图

2.4 功能描述

2.4.1 地址空间映射

本存储器和模块地址分配如下。

表 1 系统地址分配

Boundary Address	Size	Destination Slave
Code		
0x0000_0000-0x0000_0FFF	4KB	程序 Flash/SRAM/BootROM ⁽¹⁾
0x0000_1000-0x0000_7FFF	28KB	程序 Flash/ SRAM
0x0000_8000-0x0003_FFFF	224KB	程序 Flash
0x0800_0000-0x0803_FFFF	256KB	程序 Flash
0x1FFF_0000-0x1FFF_0FFF	4KB	BootROM/Chip_option
0x1FFF_1000-0x1FFF_1FFF	4KB	User Page
SRAM		
0x2000_0000-0x2000_7FFF	32KB	SRAM
0x2200_0000-0x23FF_FFFF	32MB	SRAM Bit-banding Alias Address
Peripheral		
0x4000_0000-0x4000_FFFF	64KB	AHB-to-APB Space
0x4001_0000-0x4001_FFFF	64KB	AHB-to-APB Space1
0x4002_0000-0x4002_0FFF	4KB	DMA
0x4002_1000-0x4002_1FFF	4KB	DNU
0x4003_0000-0x4003_0FFF	4KB	CRC
0x4003_1000-0x4003_1FFF	4KB	TMU
0x4003_2000-0x4003_2FFF	4KB	ADC0
0x4003_3000-0x4003_3FFF	4KB	ADC1
0x4004_0000-0x4004_FFFF	64KB	AHB-to-APB Space2
0x5000_0000-0x5000_0FFF	4KB	AHB Peripherals (GPIOA)
0x5000_1000-0x5000_1FFF	4KB	AHB Peripherals (GPIOB)
0x5000_2000-0x5000_2FFF	4KB	AHB Peripherals (GPIOC)
0x5000_3000-0x5000_3FFF	4KB	AHB Peripherals (GPIOD)
0x5000_4000-0x5000_4FFF	4KB	Reserve
0x5000_5000-0x5000_5FFF	4KB	Reserve
0x4200_0000-0x43FF_FFFF	32MB	IO Bit-banding Alias Address
0xF000_0000-0xF000_0FFF	4KB	System ROM Table

(1) 由 MEM_MODE_CTRL 寄存器配置此地址空间。

表 2 AHB-to-APB 地址空间

Boundary Address	Size	Destination Slave
0x4000_0000-0x4000_03FF	1KB	Reserve
0x4000_0400-0x4000_07FF	1KB	Reserve
0x4000_0800-0x4000_0FFF	2KB	Reserve
0x4000_1000-0x4000_13FF	1KB	CAN
0x4000_1400-0x4000_17FF	1KB	Reserve
0x4000_1800-0x4000_1FFF	2KB	Reserve
0x4000_2000-0x4000_23FF	1KB	SPI
0x4000_2400-0x4000_27FF	1KB	Reserve
0x4000_2800-0x4000_2BFF	1KB	Reserve
0x4000_2C00-0x4000_2FFF	1KB	WDT
0x4000_3000-0x4000_33FF	1KB	Basetimer
0x4000_3400-0x4000_37FF	1KB	Reserve
0x4000_3800-0x4000_3BFF	1KB	Reserve
0x4000_3C00-0x4000_3FFF	1KB	Reserve
0x4000_4000-0x4000_43FF	1KB	Reserve
0x4000_4400-0x4000_47FF	1KB	SCI0
0x4000_4800-0x4000_4BFF	1KB	SCI1
0x4000_4C00-0x4000_4FFF	1KB	SCI2
0x4000_5000-0x4000_53FF	1KB	Reserve
0x4000_5400-0x4000_57FF	1KB	I2C0
0x4000_5800-0x4000_5BFF	1KB	I2C1
0x4000_5C00-0x4000_5FFF	1KB	DAC
0x4000_6000-0x4000_63FF	1KB	CMP
0x4000_6400-0x4000_73FF	4KB	Reserve
0x4000_7400-0x4000_77FF	1KB	IOW
0x4000_7800-0x4000_FFFF	34KB	Reserve
0x4001_0000-0x4001_07FF	2KB	System Reg
0x4001_0800-0x4001_0BFF	1KB	EBUS
0x4001_0C00-0x4001_1FFF	5KB	Reserve
0x4001_2000-0x4001_23FF	1KB	FLASH controller
0x4001_2400-0x4001_27FF	1KB	I-cache

Boundary Address	Size	Destination Slave
0x4001_2800-0x4001_2BFF	1KB	D-cache
0x4001_2C00-0x4001_FFFF	53KB	Reserve
0x4004_0000-0x4004_03FF	1KB	TIMA0
0x4004_0400-0x4004_07FF	1KB	TIMA1
0x4004_0800-0x4004_0BFF	1KB	TIMH
0x4004_0C00-0x4004_0FFF	1KB	Reserve
0x4004_1000-0x4004_13FF	1KB	TIMG0
0x4004_1400-0x4004_17FF	1KB	TIMG1
0x4004_1800-0x4004_1BFF	1KB	TIMG2
0x4004_1C00-0x4004_1FFF	1KB	Reserve
0x4004_2000-0x4004_23FF	1KB	TIMS0
0x4004_2400-0x4004_27FF	1KB	TIMS1
0x4004_2800-0x4004_2BFF	1KB	TIMS2
0x4004_2C00-0x4004_2FFF	1KB	TIMS3
0x4004_3000-0x4004_3FFF	4KB	Reserve
0x4004_4000-0x4004_4FFF	4KB	Reserve
0x4004_5000-0x4004_5FFF	4KB	Reserve
0x4004_6000-0x4004_63FF	1KB	TIM_COM
0x4004_6400-0x4004_6FFF	3KB	Reserve
0x4004_7000-0x4004_FFFF	36KB	Reserve

2.4.2 Bit 操作引擎

Bit 操作引擎支持用户对特定地址进行 Bit 操作，当访问地址区间 0x2200_0000-0x23FF_FFFF 或者 0x4200_0000-0x43FF_FFFF 时，读写操作会被重映射到 0x2000_0000-0x200F_FFFF 或者 0x4000_0000-0x400F_FFFF。

如果进行 Bit 读操作，对应的 Bit 值会在读数据的最低位，如果进行 Bit 写操作，总线则会执行读改写的序列，写操作对应 Bit 位要在写数据的最低位。注意，进行 Bit 操作的地址需要 Word 对齐。

以下为 Bit 地址的计算公式：

$$\text{bit_word_offset} = (\text{byte_offset} \times 32) + (\text{bit_number} \times 4)$$

$$\text{bit_word_addr} = \text{bit_band_base} + \text{bit_word_offset}$$

其中：

bit_word_offset	bit 操作区域的偏移地址
bit_word_addr	bit 操作对应的字地址
bit_band_base	bit 操作区域的基地址 (0x2200_0000 或 0x4200_0000)
byte_offset	包含目标 bit 的偏移地址 (字节地址)
bit_number	目标 bit 的 bit 号, 0~7

举例说明：

1. 当要访问地址 0x2000_0010 的 bit[1] 时，对应的 bit 操作地址如下：

$$\text{bit_word_addr} = 0x2200_0000 + (0x10 \times 32) + 1 \times 4 = 0x2200_0204$$

2. 当要访问地址 0x4000_1003 的 bit[7] 时，对应的 bit 操作地址如下：

$$\text{bit_word_addr} = 0x4200_0000 + (0x1003 \times 32) + 7 \times 4 = 0x4202_007C$$

Kiwi Instruments Corp. Confidential

3 Flash 存储器

3.1 概述

Flash 存储器是非易失性的可重复编程的存储器，存储的数据或程序即使芯片掉电也可保存。Flash 的控制器接口支持 32 位的 AHB 和 APB 总线，支持指令 Cache 和数据 Cache 功能，在 Cache 功能打开的情况下，最高可实现 CPU 零等待时间访问。

3.2 主要功能

- 程序 Flash：存储空间分为如下两部分
 - 主存储区 (Main Flash)：总容量 256KB，分为 512 个页 (Sector)，每页容量为 512B (字节)
 - 信息存储区 (Info Flash)：总容量 8KB，分为 16 个页，每页容量为 512B
- 可以按 Byte (8 位)、Half-word (16 位)、Word (32 位) 烧写
- 支持指令 Cache
- 支持数据 Cache
- 支持页擦除和全擦除
- 支持多种页保护
- 支持选项字节 (Option Byte) 的读取和 CRC 校验
- 支持安全模式，保护代码内容
- 支持主存储区数据加密

3.3 模块框图

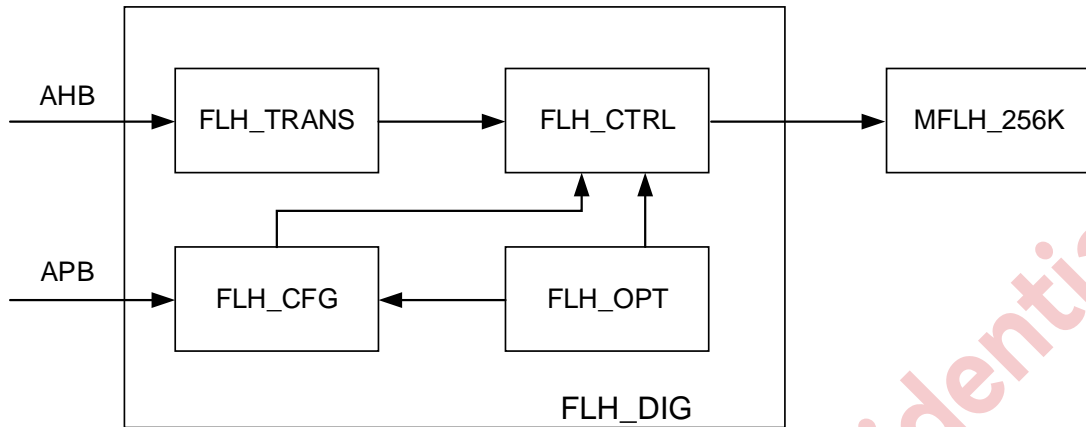


图 3 Flash 模块框图

FLH_TRANS 模块：将 AHB 接口转为内部 Flash 读接口；

FLH_CFG 模块：APB 的寄存器配置模块；

FLH_OPT 模块：完成芯片上电的 Option Byte 加载；

FLH_CTRL 模块：Flash 的状态控制模块，对 Flash 进行读、烧写和擦除操作。

3.4 功能描述

本芯片的 FLASH 分为主存储区和信息存储区组成，各存储空间的地址区间如下表所示。

表 3 系统地址分配

Flash	Flash 地址区间	容量 (字节)	序号	描述
程序 Flash (主存储区)	0x0800_0000-0x0800_01FF	512B	C_sector0	Code
	0x0800_0200-0x0800_03FF	512B	C_sector1	
	
	0x0803_FE00-0x0803_FFFF	512B	C_sector511	
程序 Flash (信息存储区)	0x1FFF_0200-0x1FFF_03FF	512B	C_info_sector1	BootROM
	0x1FFF_0400-0x1FFF_05FF	512B	C_info_sector2	
	
	0x1FFF_0C00-0x1FFF_0DFF	512B	C_info_sector6	
	0x1FFF_0E00-0x1FFF_0FFF	512B	C_info_sector7	Chip_option
	0x1FFF_1000-0x1FFF_1FFF	4KB	D_sector0-7	User Page

CPU 可以通过 AHB 总线对以上 Flash 地址空间进行读访问，但是烧写和擦除操作需要对 Flash 的寄存器进行特定的设置才能完成。

3.4.1 Flash 读延迟

当 CPU 工作在不同主频情况下，需要对 Flash 的读延迟寄存器进行设置，如下。

表 4 Flash 读延迟设置

CPU 时钟	读取延迟
CPU_CLK≤50MHz	1 时钟延迟
CPU_CLK≤100MHz	3 时钟延迟

3.4.2 Flash 烧写和擦除

Flash 支持烧写和擦除操作，通过 Flash 命令寄存器进行设置。当对 Flash 进行烧写或擦除时，不能再对此 Flash 进行读访问，否则读访问会被挂住，直到烧写或擦除全部完成，才会返回读数据，所以在进行烧写或擦除之前，需要把烧写擦除程序拷贝到 SRAM 上面，CPU 在 SRAM 上面执行烧写擦除程序，等到 Flash 完成烧写和擦除，再跳回到 Flash 上面执行程序。

下表是 Flash 烧写和擦除所耗费的时间。

表 5 Flash 烧写和擦除时间

Flash 操作	时间
烧写 (4 字节)	50μs
页擦除	4.5ms
全擦除	40ms

3.4.3 Flash 中断号

表 6 Flash 中断号

名称	中断号
FLH_INT	3

3.5 寄存器

3.5.1 Flash 寄存器

Flash 寄存器的基地址为 0x4001_2000，下表为 Flash 的各控制寄存器描述。

表 7 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x000	Flash 地址寄存器	32	r/w	0x00000000h
0x004	Flash 数据寄存器	32	r/w	0x00000000h
0x010	Flash 控制寄存器	32	r/w	0x00000000h

3.5.1.1 Flash 地址寄存器 (FLH_ADR)

偏移地址: 0x000

表 8 Flash 地址寄存器

位	访问	描述
31:0	r/w	FLH_ADR: Flash 烧写/擦除地址 页擦除: 配置为该页的首地址 烧写: Byte 模式 -地址应为 Byte 对齐 Half-Word 模式 -地址应为 Half-Word 对齐 Word 模式 -地址应为 Word 对齐 复位值: 0x0

3.5.1.2 Flash 数据寄存器 (FLASH_DATA)

偏移地址: 0x004

表 9 Flash 数据寄存器

位	访问	描述
31:0	r/w	FLH_DATA: 烧写数据 复位值: 0x0

3.5.1.3 Flash 控制寄存器 (FLH_CTRL)

偏移地址: 0x010

表 10 Flash 控制寄存器

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	PROG_MODE: Flash 烧写模式 0: Word 烧写/次 1: Half-Word 烧写/次 2: Byte 烧写/次 3: 保留 复位值: 0x0
3:2	Res	Reserved 复位值: 0x0
2:0	r/w	LATENCY: Flash 读取延迟 0: Reserved 1: 延迟 1 个时钟 (系统时钟为小于等于 50M) 2: 延迟 2 个时钟 3: 延迟 3 个时钟 (系统时钟为 100M) 4: 延迟 4 个时钟 5: 延迟 5 个时钟 6: 延迟 6 个时钟 (系统时钟为 200M) 7: 延迟 7 个时钟 复位值: 0x6

3.5.2 ICACHE 寄存器

指令 cache (ICACHE) 的基地址为 0x4001_2400, 下表为 ICACHE 的各控制寄存器描述。

表 11 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x000	ICACHE 控制寄存器	32	r/w	0x00000040h
0x004	ICACHE 状态寄存器	32	r/w	0x00000000h
0x014	ICACHE HIT 统计寄存器	32	r/w	0x00000000h
0x018	ICACHE MISS 统计寄存器	32	wo	0x00000000h

3.5.2.1 ICACHE 控制寄存器 (ICACHE_CTRL)

偏移地址: 0x000

表 12 ICACHE 控制寄存器

位	访问	描述
31:7	Res	Reserved 复位值: 0x0
6	r/w	CACHE_STATIS_EN: cache 统计使能 0: 关闭 1: 使能 注: 使能后将会统计 HIT 和 MISS 的数据。 复位值: 0x1
5	r/w	CACHE_PREF_EN: cache 预取使能 0: 关闭 1: 使能 复位值: 0x0
4:1	Res	Reserved 复位值: 0x0
0	r/w	CACHE_EN: cache 使能 0: 关闭 1: 使能 复位值: 0x0

3.5.2.2 ICACHE 状态寄存器 (ICACHE_STATUS)

偏移地址: 0x004

表 13 ICACHE 状态寄存器

位	访问	描述
31:2	Res	Reserved 复位值: 0x0
1:0	ro	CACHE_STA: cache 状态 2'b00: 关闭 2'b01: 使能中 2'b10: 已能中 2'b11: 关闭中 复位值: 0x0

3.5.2.3 ICACHE HIT 统计寄存器 (ICACHE_HIT_STATIC)

偏移地址: 0x014

表 14 ICACHE HIT 统计寄存器

位	访问	描述
31:0	r/w	CACHE_CSHR[31:0]: 统计 cache hit 的数量 复位值: 0x0

3.5.2.4 ICACHE MISS 统计寄存器 (ICACHE_MISS_STATIC)

偏移地址: 0x018

表 15 ICACHE MISS 统计寄存器

位	访问	描述
31:0	r/w	CACHE_CSMR[31:0]: 统计 cache miss 的数量 复位值: 0x0

3.5.3 DCACHE 寄存器

数据 cache (DCACHE) 的基地址为 0x4001_2800, 下表为 DCACHE 的各控制寄存器描述。

表 16 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x000	DCACHE 控制寄存器	32	r/w	0x00000040h
0x004	DCACHE 状态寄存器	32	r/w	0x00000000h
0x014	DCACHE HIT 统计寄存器	32	r/w	0x00000000h
0x018	DCACHE MISS 统计寄存器	32	wo	0x00000000h

DCACHE 的寄存器与 ICACHE 相同, 参见 ICACHE 寄存器描述。

4 时钟 (Clock)

4.1 概述

系统时钟的选择在启动的过程中就完成了相关的配置，其中内部 RC 8MHz 振荡器默认选为系统时钟。启动完成之后，外部 8MHz 时钟、备份内部 RC 8MHz 振荡器 2、内部 32kHz RC 振荡器和锁相环 100MHz 时钟可供选择使用，备份内部 RC 8MHz 振荡器 2 用于监测外部 8MHz 时钟、内部 RC 8MHz 振荡器和锁相环 100MHz 时钟，必要的时候开启时钟监视模块，一旦检测到时钟丢失，芯片会自动切换到备份内部 RC 8MHz 时钟继续运行。

4.2 主要功能

支持多种时钟源用作系统时钟 (FCLK)

- HIRC 8MHz RC 振荡器
- HIRC 8MHz RC 振荡器 2 (备份时钟)
- HOSC 晶振时钟
- PLL 100MHz 时钟

以上提及的每个时钟源均可独立地开启或者关闭，不使用时可关闭以节省功耗。

系统时钟支持预分频处理，AHB 和 APB 区域同属于系统时钟域，也就是最大的工作频率为 100MHz。

PLL 参考时钟可由软件选择，包括 HIRC 和 HOSC。

大多数 IP 时钟在初次启动之后都是处于门控状态（无时钟输出），除了 CPU、SRAM、FLASH、看门狗和总线矩阵模块。几乎所有的外设时钟均源于它们所在的总线时钟（HCLK 对应于 AHB，或者 PCLK 对应于 APB），然而以下例外。

- 选项字节 (Option Byte) 加载模块的时钟源于 MCLK 时钟
- LVD 模块的时钟源于 MCLK 时钟
- PMU&HIRC&LIRC 配置寄存器的时钟 AOS_CLK 源于 MCLK 时钟
- 看门狗时钟支持以下时钟源
 - LIRC 时钟
 - MCLK

- 基准定时器支持以下时钟源
 - LIRC 时钟
 - MCLK
- IO 噪声滤波时钟支持以下时钟源
 - 系统时钟
 - LIRC 时钟
 - MCLK
- MCO 时钟支持多个时钟源
 - LIRC 时钟
 - HIRC2 时钟
 - HIRC 时钟
 - HOSC 时钟
 - 系统时钟

4.3 模块框图

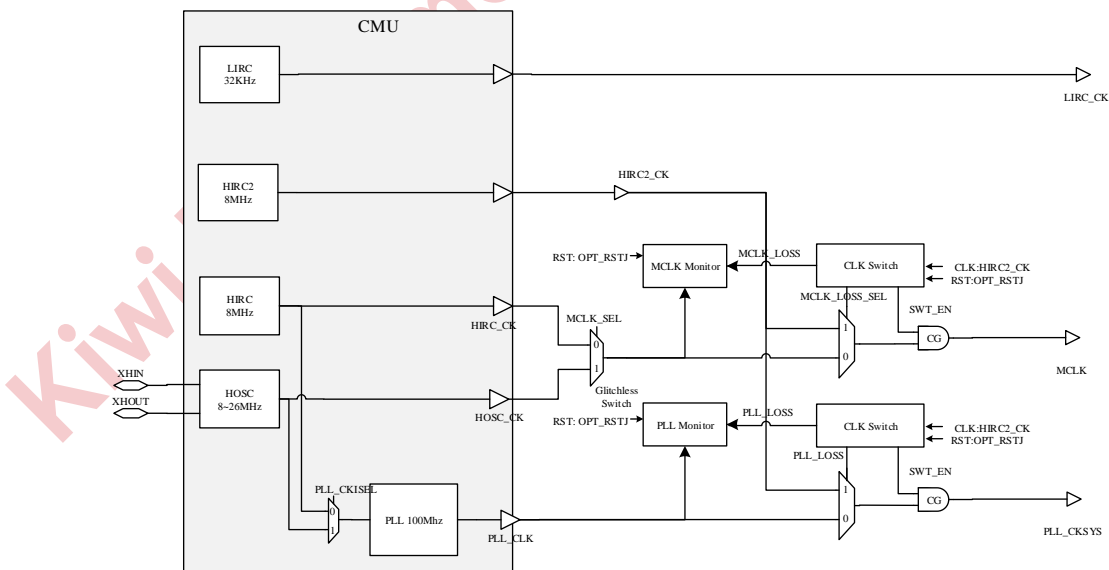


图 4 芯片时钟源示意图

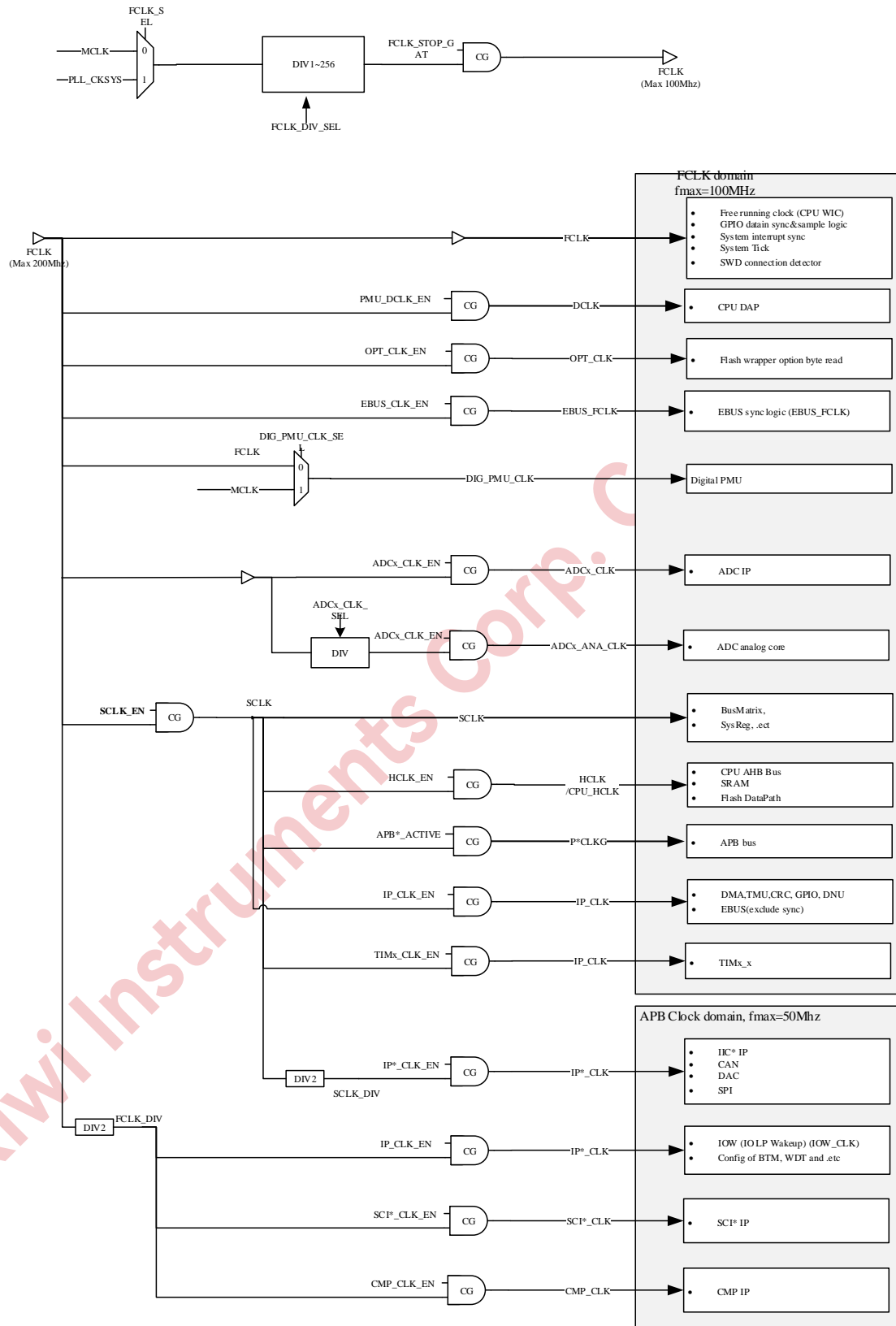


图 5 系统时钟组 (含同步 IP) 架构图

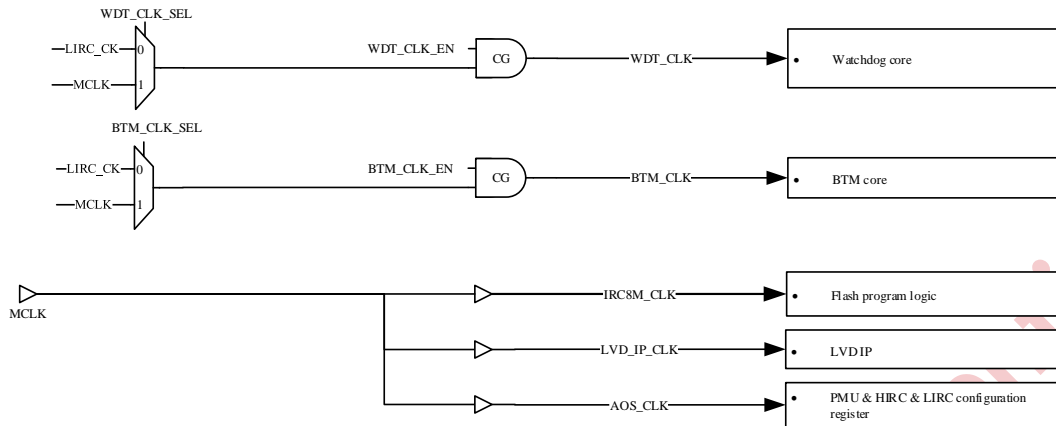


图 6 与系统时钟异步时钟架构图

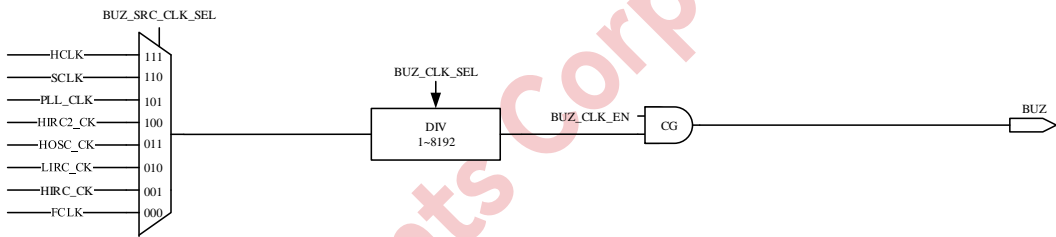


图 7 蜂鸣器时钟架构图

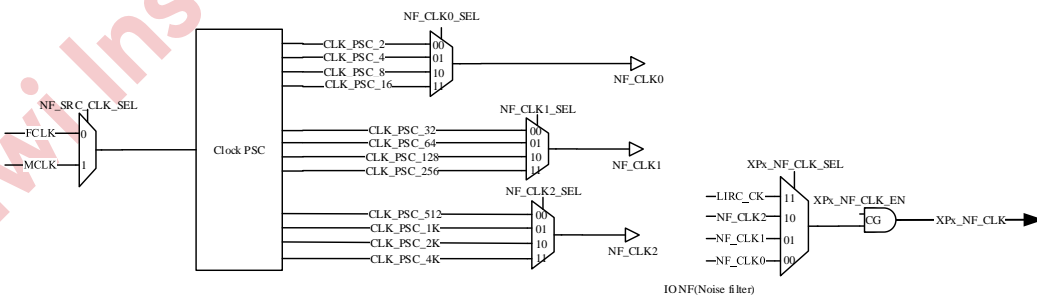


图 8 IO 噪声滤波时钟架构图

4.4 功能描述

4.4.1 HOSC 时钟

高速外部时钟信号 (HOSC) 可由这两个时钟源生成:

- HOSC 外部晶振或陶振
- HOSC 外部用户时钟

谐振器和负载电容必须放置在尽可能靠近振荡器引脚的位置, 以减少输出失真和启动稳定的时间。负载电容必须根据所选的振荡器进行调整。

外部晶振或陶瓷谐振器 (HOSC 晶振)

8MHz 外部振荡器的优势在于可产生非常精准的时钟信号。HOSC 振荡器稳定与否是体现在复位状态寄存器 (RST_ST) 的 HOSC_LOCK 标志。芯片复位启动时, HOSC 默认关闭。HOSC 晶振可通过时钟控制寄存器 (HOSC_CTRL_REG) 的 HOSC_EN 位进行开启或关闭。

注意:

如要开启 HOSC 振荡器, 需先使能 HOSC_EN, 经过大约 65536 个 HOSC 时钟脉冲后方可看到相关的内部稳定信号。即使在没有外挂晶振或谐振器的情况下, XHIN 引脚上的过多外部噪声仍可能导致振荡器启动。在系统时钟源于 HOSC 情况下, 如果碰到因为一些任何意想不到的原因导致 XHIN 引脚上的晶振失效或不复存在, 系统时钟跟着丢失, 此时振荡器不能被关闭, OSC 相关的引脚也被锁定不能用于其他用途, 并引入了不必要的功耗。为了避免这种情况, 强烈建议始终启用时钟监控系统, 即使在这种情况下, 系统仍能关闭 HOSC。

外部时钟源 (HOSC 旁路)

在这种模式下, 外部时钟源可用于供应芯片内部, 其频率可达 8MHz。外部时钟信号可以是 40~60%占空比的方波信号 (正弦或者三角波信号), 通过 XHIN 引脚进入芯片, 而 XHOUT 则可用作 GPIO。

4.4.2 HIRC 时钟

HIRC 时钟信号是由内部 8MHz 的 RC 振荡器产生, 可直接用作系统时钟或 PLL 输入。

HIRC 振荡器的优点是可提供一个低成本的时钟源 (无外部组件)。其启动时间也比 HOSC 晶体振荡器更快。然而, 即使经过校准, 其频率也不如外部晶体振荡器或陶瓷谐振器精确。

校准

由于制造工艺的差异, 每个芯片的 RC 振荡器频率都有差别。这也正是为什么每颗芯片都需要设计厂商在 $TA=25^{\circ}\text{C}$ 情况下实现 $\pm 1\%$ 精度工厂校准。芯片复位后, HIRC 所需要的工厂校准值从系统选项字节中加载。

HIRC 振荡器稳定与否是体现在复位状态寄存器 (RST_ST) 的 HIRC_LOCK 标志。在启动过程中，直至硬件设置了这个标志位，HIRC 输出时钟才被释放。HIRC 振荡器可以通过时钟控制寄存器 (IRC_CTRL_REG) 的 HIRC_PD 位进行开启或关闭。

特别地，HIRC 用于低功耗模式下的开关时钟，因此其可被硬件控制上电或下电。

4.4.3 HIRC2 时钟

HIRC2 时钟信号是由内部 8MHz 的 RC 振荡器 2 产生，用于监测系统源时钟 MCLK 和 PLL 时钟，并作为备份时钟替换监测时钟。

HIRC2 的物理特性与 HIRC 一致。

HIRC 振荡器稳定与否是体现在复位状态寄存器 (RST_ST) 的 HIRC2_LOCK 标志。在启动过程中，直至硬件设置了这个标志位，HIRC2 输出时钟才被释放。HIRC2 振荡器可以通过时钟控制寄存器 (IRC_CTRL_REG) 的 HIRC2_PD 位进行开启或关闭。

4.4.4 PLL 时钟

内部 PLL 是用于对 HIRC 或 HOSC 进行倍频输出更高频率。在使能 PLL 之前，PLL 相关配置（输入参考时钟选择，电荷泵电流和参考频率）必须配置完成。一旦 PLL 启动，这些配置参数不能运行修改。

如需修改 PLL 配置，请按照以下步骤进行：

- 将 PLL_EN 设置为 0 以关闭 PLL
- 更改所需参数
- 将 PLL_EN 设置为 1 以启动 PLL
- 等待 PLL 稳定状态位置位（复位状态寄存器的 PLL_LOCK）

4.4.5 LIRC 时钟

LIRC 作为一个低功耗时钟源，可以为看门狗、实时时钟和基准定时器在停止模式下运行。时钟频率在 32kHz 左右 ($\pm 15\%$)。有关详细信息，请参阅数据表的电气特性部分。

复位状态寄存器 (RST_ST) 的 LIRC_LOCK 标志指示 LIRC 振荡器稳定与否。在启动过程中，直到该标志位由硬件置位，LIRC 时钟方可释放。

4.4.6 系统时钟组

CM4F 集成时钟

系统时钟组提供 CPU 及其外围组件的时钟，总共三个时钟，包括 FCLK、HCLK 和 SCLK，并且这些时钟工作于同样的频率。

FCLK 在 CM 所有模式下（普通模式、睡眠模式、深度睡眠模式）一直运行。

SCLK 直接源于 FCLK。它在处理器处于 WIC 模式下深层睡眠而没有调试器连接的情况下始终保持运行。

HCLK 直接由 FCLK 派生，连接至处理器所连接的 AHB 层。HCLK 可在处理器处于睡眠模式下门控。

- FCLK 驱动 WIC 模块
- HCLK 驱动 CPU
- SCLK 驱动 NVIC，总线矩阵和系统寄存器

PMU 管理单元生成和管理以上时钟。在连接调试器时，始终驱动 DCLK。而没有调试器断开时，DCLK 可被门控。

对于其他时钟信号，当 MCU 工作于普通模式，时钟都是自由运行的。如果 MCU 切换至睡眠模式，HCLK 将被门控。假设 MCU 切换至深度睡眠模式，SCLK 和 HCLK 均被门控。关于 MCU 工作模式的详细信息，请参考功耗模式章节。

可使用以下各种时钟源用于驱动系统时钟组：

- MCLK 时钟
- PLL 时钟

系统时钟组有一个预分频器，支持 1/2/4/8/16/32/64/128/256 分频。其配置信息详见系统寄存器章节。

MCLK 时钟源于 HIRC 或 HOSC，是系统的源时钟，可作为系统时钟组其中一个来源。当 MCLK 发生时钟丢失的情况下，MCLK 的时钟源会切换至备份时钟 HIRC2，以保证 MCLK 有效时钟沿不间断，以保证系统后续正常处理。

系统时钟组中的 IP 时钟

DMA、TMU、CRC、GPIO、TIM、EBUS（同步逻辑除外）源自于 SCLK，并配备对应的门控使能控制位，可在该 IP 不使用的情况下进行门控以节省功耗。当 MCU 处于深度睡眠模式下，上述时钟均被门控。

IIC、CAN、DAC、CMP 和 SPI 源自于 SCLK 的分频时钟（2 分频，最快 100MHz），仅在总线接口部分与系统时钟同步，所以 CPU 访问相关的 IP 至少有 6 个 FCLK 时钟周期。相关 IP 配备对应的门控使能控制位，可在该 IP 不使用的情况下进行门控以节省功耗。当 MCU 处于深度睡眠模式下，上述时钟均被门控。

IOW、SCI、ADC IP 数字部分以及 EBUS 同步逻辑部分这三个部分的时钟信号源自于 FCLK，并配备对应的门控使能控制位，可在该 IP 不使用的情况下进行门控以节省功耗。

BTM、CMP 和看门狗 IP 的控制寄存器域的时钟信号源自于 FCLK 的分频时钟（2 分频，最快 100MHz），并配备对应的门控使能控制位，可在该 IP 不使用的情况下进行门控以节省功耗。

4.4.7 与系统时钟组成异步关系的 IP 时钟

系统中存在一些 IP 不与系统时钟组同步，因为这些 IP 时钟由自己的时钟生成器生成。

选择字节加载模块时钟

选项字节（Option Byte）加载模块的时钟 IRC8M_CLK 源于 MCLK 时钟。

LVD 模块时钟

LVD 模块的时钟源于 MCLK 时钟。

AOS_CLK

PMU&HIRC&LIRC 配置寄存器的时钟 AOS_CLK 源于 MCLK 时钟。

ADC 控制逻辑（与模拟部分紧密关联部分）时钟

ADC 控制逻辑时钟源于 FCLK 并运行于 FCLK 的分频时钟，具体分频基于软件配置。该 IP 可被门控以节省功耗。

WDT 内核时钟

WDT 时钟源于 LIRC 和 MCLK，默认配置运行于 LIRC 时钟。该 IP 可被门控以节省功耗。

BTM 内核时钟

BTM 时钟源于 LIRC 和 MCLK，默认配置运行于 LIRC 时钟。该 IP 可被门控以节省功耗。

IO 噪声滤波时钟

IO 中断和 IIC 端口所属的 IO 噪声滤波时钟源于以下 4 种源之一：

- NF_CLK0 (FCLK/HIRC 2/4/8/16 分频)
- NF_CLK1 (FCLK/HIRC 32/64/128/256 分频)
- NF_CLK2 (FCLK/HIRC 512/1024/2048/4096 分频)
- LIRC 时钟

以上 IO 噪声滤波时钟源相关独立，默认选择源于 NF_CLK0。

蜂鸣器时钟

蜂鸣器时钟源于以下 8 种之一：

- FCLK
- HIRC
- LIRC
- HOSC
- LOSC
- PLL_CLK
- SCLK
- HCLK

默认是运行于 FCLK，支持时钟预分频，包括 $2^0/2^1/.../2^{13}$ 。

注意：

在将 MCU 的主时钟源进行切换或者切换 PLL 参考时钟（HIRC 或 HOSC）之前，软件代码必须通过查验 RST_ST 系统寄存器相关状态位以确保目标时钟已完全稳定。欲了解更详细信息，请参考系统寄存器章节。

4.4.8 时钟监视器

KPM32R14TX 包含两个时钟监视器，分别用于 MCLK 和 PLL。以下文字会依次介绍。

4.4.8.1 MCLK 监视器

MCLK 监视器用于监测 MCLK 频率的准确性和稳定性。实际上，它就是一个时钟检测器，并且默认关闭。在启动该监视器之前需根据 MCLK_SEL 判定 MCLK 的时钟源的时钟状态是否为高（当 MCLK_SEL=0，MCLK 源于 HIRC，那应确定 HIRC_LOCK 是否为高，也就是 HIRC 处于非关电状态。当 MCLK_SEL=1，MCLK 源于 HOSC，那应确定 HOSC_LOCK 是否为高，也就是 HOSC 处于非关电状态）。如果 MCLK 丢失或损坏，HIRC2 时钟将替代它直至 MCLK 恢复。

启用之后，如果监视器监测到 MCLK 时钟停止或者 MCLK 的频率低于 $f_{HIRC2}/5$ ，便产生一个时钟失效事件发送至 NMI，并生成一个未决标志（可在系统寄存器中读取）。时钟失效事件驱动硬件由 HIRC2 时钟代替 MCLK 时钟输入。然等待软件清除 MCLK 丢失未决状态位，MCLK 时钟输入恢复。

4.4.8.2 PLL 监视器

PLL 监视器用于监测 PLL 频率的准确性和稳定性。实际上，它就是一个时钟检测器，并且默认关闭。在启动该监视器之前须确定 PLL 时钟状态 PLL_LOCK 是否为高，也就是 PLL 处于非关电状态。如果 PLL 丢失或损坏，HIRC2 时钟将替代它直至 PLL 恢复。

启用之后，如果监视器监测到 PLL 时钟停止或者 PLL 的频率低于 $f_{HIRC2} * 5$ ，便产生一个时钟失效事件发送至 NMI，并生成一个未决标志（可在系统寄存器中读取）。时钟失效事件驱动硬件由 HIRC2 时钟代替 PLL 时钟输入。然等待软件清除 PLL 丢失未决状态位，PLL 时钟输入恢复。

4.4.9 频率监测器

频率监测器用于监测系统各个时钟频率的准确性和稳定性。这是一个时钟频率监测电路模块，默认关闭。在启动该监测器之前需确保监测时钟和参考时钟均为稳定，可查看相关状态位进行确认。

频率监测的原理是：待测时钟在分频后的参考时钟的连续两个上升沿之间进行计数，所得数值则是待测时钟和参考时钟频率之间的比例关系，依据参考时钟频率、参考时钟的分频值和计数值则可反推出待测时钟的频率。

为了保证量测的精度和准确度，待测时钟的频率应大于分频之后的参考时钟频率，假设 $f_{OBJ} = N * f_{REF_DIV}$ （其中 $f_{REF_DIV} = f_{REF} / DIV$ ， $DIV = 1, 32, 64, 128, 256, 512, 1024$ 或 2048 ），建议 $100 \leq N \leq 65535$ 。

可监测的时钟包括以下几个：

- FCLK
- HIRC
- HIRC2
- HOSC

参考时钟可以有以下几种，鉴于监测时钟不同频率范围以及准确度的需求，该模块支持参考时钟分频扩大单位量测宽度。

- HIRC2
- HOSC
- LIRC
- EXT_REFCLK

时钟监测器的基本功能是基于精准的参考时钟对待测时钟进行时钟频率的监测。在完成基本时钟源配置之后启用检测器模块，然后等待量测结束标记 (MEASURE_END) 置位，量测结果在寄存器 CLK_PRD 获取，具体的频率转换关系是 $f_{OBJ} = f_{REF} * CLK_PRD / DIV$ ，其中 $DIV = 1, 32, 64, 128, 256, 512, 1024$ 或 2048 ，依据 REF_CLK_DIV 配置而定。

时钟监测器支持预警异常处理，在使用过程中需要设置上下限阈值，根据需要由软件激活。启用之后，如果监视器捕获的周期数不在设置阈值的周期范围内 ($CLK_PRD > PRD_THD_HIGH$ 或 $CLK_PRD < PRD_THD_LOW$)，便产生一个时钟频率错误事件发送至 NMI，并生成一个未决标志 (可在系统寄存器中读取)。

4.4.10 低功耗

关于低功耗模式下的时钟管理细节在功耗管理章节详细描述。

Kiwi Instruments Corp. Confidential

5 复位 (Reset)

5.1 概述

KPM32R14TX 包含两类复位源：系统复位和 POR 复位，如下表所列。

表 17 复位源

复位源	描述
POR 复位	外部 RC 复位 (PAD_RSTJ) 上电复位 (POR_RSTJ) VCCIO 低电压检测复位 (LVD_RSTJ)
系统复位	看门狗定时器复位 (WDT_RSTJ) 系统软件复位 (SYS_RSTJ) LOCKUP 复位 (LOCKUP_RSTJ)

每一个系统复位源在复位状态寄存器中都有一个相关的挂起状态位。低电压检测复位和上电复位在复位状态寄存器中共用一个挂起状态位。

无论功能模式下 CPU 是正处于执行代码或是处于调试停止状态，KPM32R14TX 都可以退出并复位。芯片支持几种启动选择可供配置，详见启动章节。

5.2 模块框图

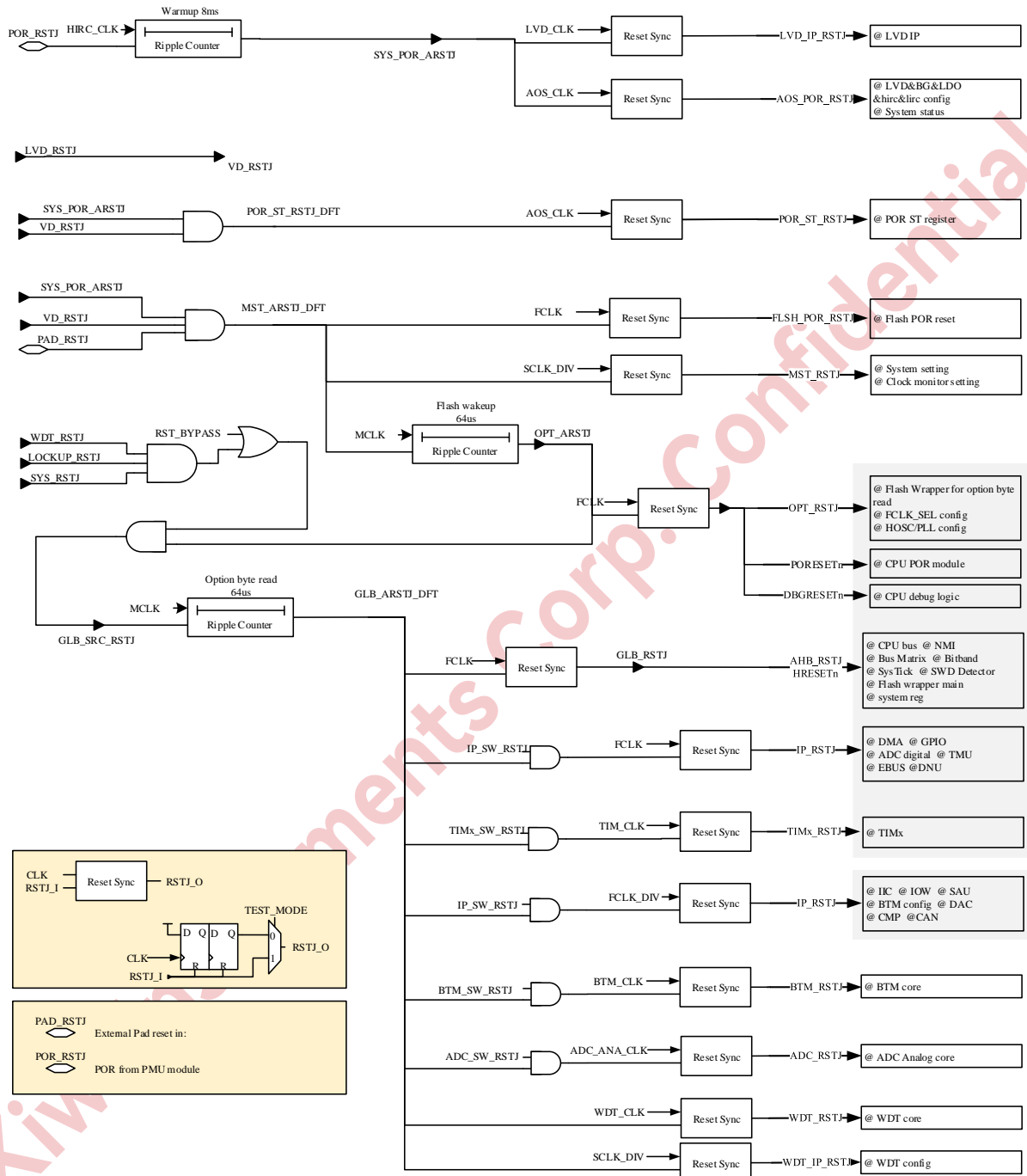


图 9 芯片复位架构图

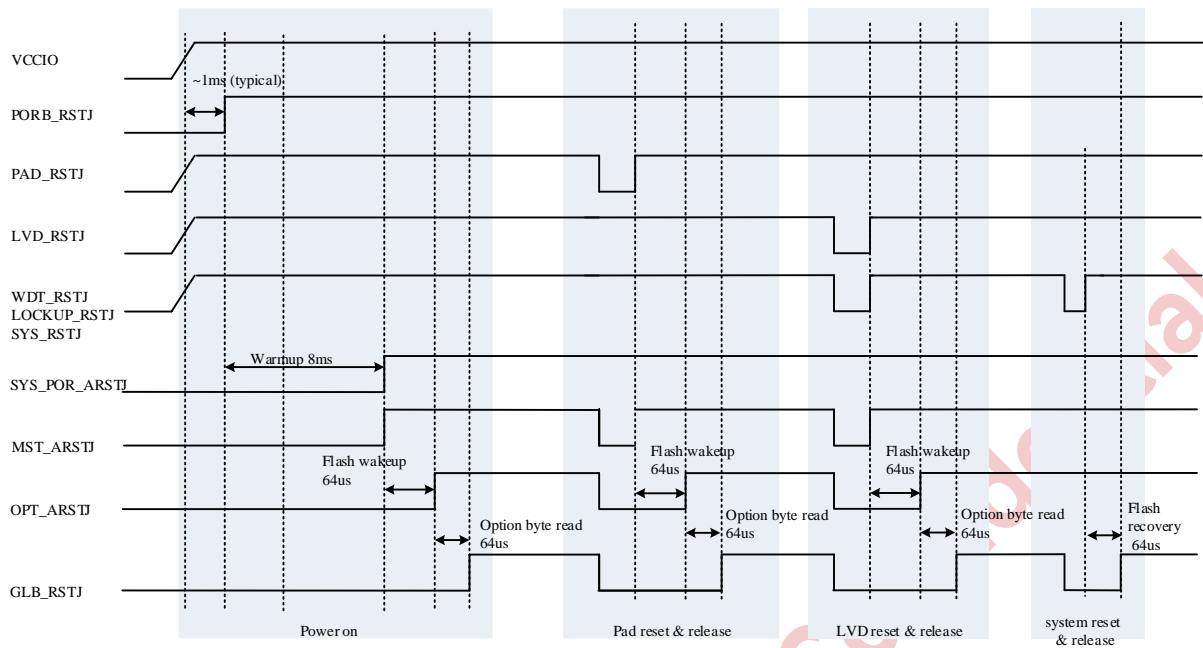


图 10 复位时序图

5.3 功能描述

本章节描述了基本的复位机制及复位源。所有的复位信号都是低有效。

5.3.1 外部 RC 复位

外部 RC 复位引脚是一个带有内部上拉电阻的专用复位 IO。它用于在任何模式下生效 PAD_RSTJ，从而将复位整个数字逻辑。

5.3.2 上电复位 (POR)

芯片支持上电复位。当发生以下事件之一，芯片触发上电复位 (POR_RSTJ)

- 电源从 0 开始给 MCU 上电
- 电源电压 VCCIO 低于复位电压阈值

上述任何一个事件都会导致 POR 复位电路产生 POR 复位条件。而上电复位将所有的寄存器重置为默认值。

5.3.3 低电压检测 (LVD)

KPM32R14TX 包含一个应对低电压条件的硬件系统。它可以保护存储器内容和 MCU 系统状态免受电源电压变化的影响。LVD 电路包括两个独立的电压检测模块：LVD1 用于 VCCIO 检测，默认开启；LVD2 是在 VCCIO 不稳定时发出警告信号，默认关闭。当 VCCIO 低于所设置的参考电压时，LVD_RSTJ 生效并复位芯片。请参阅 LVD 获取更多详细信息。

5.3.4 看门狗定时器 (WDT)

看门狗定时器是通过获得预期的软件定期通信，从而实现监控系统是否正常运行。这种通信通常被称为服务（或刷新）看门狗。如果不进行周期性刷新，看门狗电路便会产生系统复位 (WDT_RSTJ)。WDT_RSTJ 不能复位时钟源模块、实时时钟 (RTC) 和 Cortex-M4F 中的 DAP 逻辑。

5.3.5 软件复位 (SYS)

对 NVIC 应用中断和复位控制寄存器中的 SYSRESETREQ 位置位可强制产生设备上的软件复位（请参考 ARM 的 NVIC 文档了解相关寄存器的完整描述，尤其是 VECTKEY 字段的需求）。设置 SYSRESETREQ 会产生软件复位请求。和 WDT_RSTJ 一样，这个复位不能复位时钟源模块和 Cortex-M4F 中的 DAP 逻辑。

5.3.6 锁定复位 (LOCKUP)

LOCKUP 是对内核软件发生严重错误的立即显示。这是由于在激活处理器内置的系统状态保护硬件后，一个不可恢复的异常导致内核被锁定的结果。LOCKUP 发生会导致系统复位，同时 0x4001_0010[2] 也被置位。

5.3.7 复位域

每个复位源所能影响的区域如下表所示。

表 18 复位区域列表

复位源	CLK_GEN	WDT	OPT	EFLASH	IOW	DAP	其他
PAD_RSTJ	V	V	V	V	V	V	V
POR_RSTJ	V	V	V	V	V	V	V
LVD_RSTJ	V	V	V	V	V	V	V
WDT_RSTJ	V	V	X	V	V	X	V
SYS_RSTJ	V	V	X	V	V	X	V
LOCKUP_RSTJ	V	V	X	V	V	X	V

所有的复位源复位释放之后，系统总是从 0x00000000（具体物理地址请参照 MEM_MODE_CTRL）启动。

PAD_RSTJ、POR_RSTJ 和 LVD_RSTJ 释放后会触发选项字节 (Option Byte) 会重新加载，而其他不会。

5.3.8 IP 软件复位控制位

KPM32R14TX 为大多数数字 IP 提供了软件复位控制位 (0x4001_0280 & 0x4001_0284)，这些 IP 可由软件代码单独复位。有关于具体位的分配细节，请参阅系统寄存器章节。

6 功耗管理 (Power)

6.1 概述

本章描述了本芯片所支持的几种功耗模式，以及在这些功耗模式下相应模块的功能特性。

6.2 主要功能

KPM32R14TX 支持多种工作模式可用于节省功耗。相关特性如下：

- 支持基于 Cortex M4F 的运行、睡眠和深度睡眠模式
- 支持停止模式
- 支持运行模式下功耗策略

6.3 功能描述

6.3.1 电源供应

KPM32R14TX 内嵌一个电压调节器，用于向数字电源域提供 1.2V 电压。模拟电路、IO、内核、存储、数字外设的电源供应如下：

- 模拟电路模块部分电源供应

VCC 为 5V，用于供应 ADC、电压检测模块、内部 RC 振荡器、带隙基准、LDO、DAC、CMP 和 PLL

- IO 电源供应

VCCIO 为 5V

- 内核、存储、数字外设的电源供应

VDDCORE 为 1.2V

6.3.2 电源供应管理

该设备上集成了一个上电复位 (POR) 电路，复位电路一直活跃，确保所有的正确操作都高于一个阈值 (VPOR)。

对于 POR 而言，当监测到电源电压低于指定的阈值 (VPOR) 时，设备无需外部复位电路参与而保持在复位状态。POR 模块一直监测 VDDCORE 和 VCC 电源电压。

有关电源复位阈值的详细信息，请参阅数据表中的电气特性部分。

6.3.3 不同模式下的模块状态

下面的表格中展示了不同模式下的模块状态。

模块	时钟	工作模式 ⁽⁶⁾			
		普通	睡眠	深度睡眠	停止
内核模块					
CPU_CORE ⁽¹⁾	HCLK	ON	GT	GT	GT
WIC	FCLK	ON	ON	ON	GT
NVIC ⁽²⁾	SCLK	ON	ON	GT	GT
DAP ⁽³⁾	FCLK (DCLK)	GT			
SWCLKTCK	外灌时钟	SWCLKTCK 一般由外部调试所提供与 FCLK、HCLK、SCLK 和 DCLK 是异步关系。			
BUS_MATRIX	SCLK	ON	ON	GT	GT
DMA	DMA_CLK ⁽⁴⁾	ON	ON	GT	GT
存储模块					
Flash	FCLK	ON	ON	ON	GT
SRAM	SCLK	ON	ON	GT	GT
数字外设					
SYS_REG	SCLK	ON	ON	GT	GT
WDT	WDT_CLK ⁽⁵⁾	OPT	OPT	OPT	OPT
GPIO	GPIO_CLK ⁽⁴⁾	OPT	OPT	GT	GT
TIM	TIM_CLK	OPT	OPT	GT	GT
SCI	SCI_CLK	OPT	OPT	OPT	GT
SPI	SPI_CLK	OPT	OPT	GT	GT
CAN	CAN_CLK	OPT	OPT	GT	GT
IIC	IIC_CLK	OPT	OPT	GT	GT
CMP	CMP_CLK	OPT	OPT	OPT	GT
BTM	BTM_CLK ⁽⁵⁾	OPT	OPT	OPT	OPT
EBUS	EBUS_CLK	OPT	OPT	GT	GT

模拟模块					
PMU	-	-	-	-	-
ADC	ADC_CLK	OPT	OPT	ON	GT
LVD	LVD_CLK	ON	ON	ON	GT

- (1) HCLK主要用于内核 (CORE)，SCLK集中用于处理中断
- (2) SCLK主要用于NVIC模块，HCLK为CPU处理未决SV的异常
- (3) 当没有调试访问时，被门控
- (4) 时钟源于SCLK并且与其同频，可由系统寄存器门控
- (5) 默认源于FCLK，可通过系统寄存器控制位切换至其他时钟源
- (6) ON：表示正常运行；GT：表示模块时钟被关闭（门控）；SD：表示模块关电；OPT：表示该模块在进入现有状态前可配置

表 19 不同模式下的模块状态

下图说明了 CPU 时钟架构。

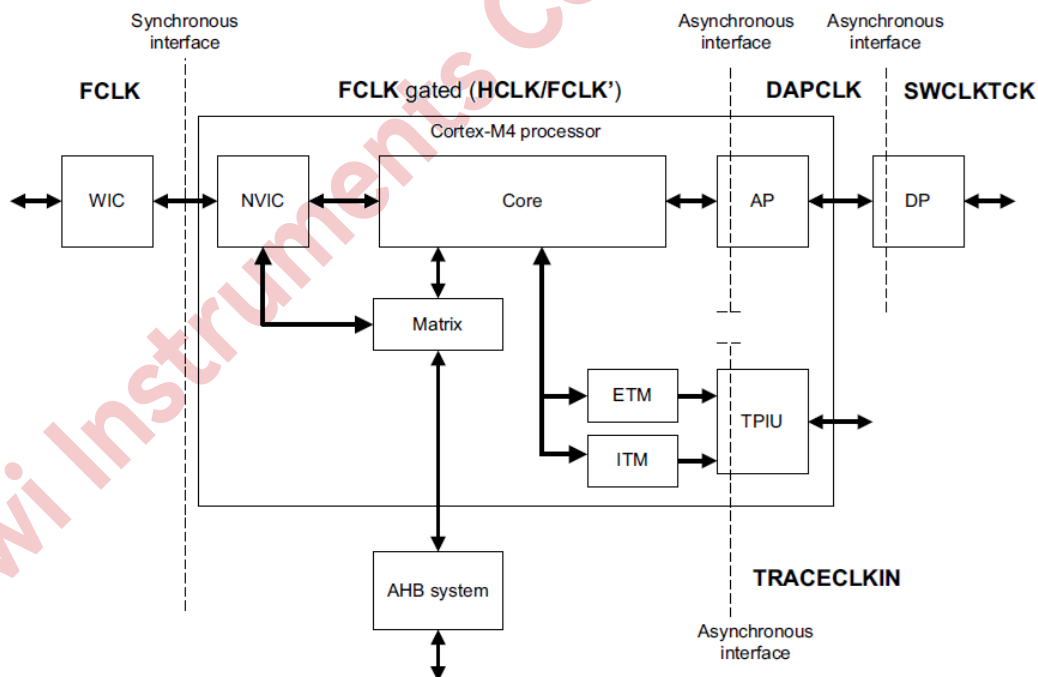


图 11 Cotex-M4F 时钟域

6.3.4 低功耗模式

KPM32R14TX 在每次系统复位或上电重启时，默认处于运行模式。当 CPU 不需要保持运行时（例如在等待外部事件时），可以使用几种低功耗模式来节省电源。用户可以在选择低功耗、短启动时间和可用的唤醒源之间做出最佳折衷的模式

KPM32R14TX MCU 引入了四种低功耗模式：

- 运行模式：运行模式下的功耗可通过以下方式进行
 - 放慢系统时钟
 - 外围时钟门控
- 睡眠模式：CPU 时钟关闭，但 Cortex®-M4F 核心外设如 NVIC，SysTick 仍然保持运行
- 深度睡眠模式：除了 WDT/BTM 所有时钟都关闭
- 停止模式：所有源于 FCLK 的时钟均停止，FCLK 也被门控，其他模块可依据配置关闭

6.3.4.1 运行模式

运行模式下功耗降低可通过以下方式降低：

- 降低系统时钟

在运行模式下，系统时钟（FCLK、HCLK 和 PCLK）的速度可以通过对 FCLK 的预分频器进行编程实现降速。这些预分频器也可以用来进入睡眠模式之前降低外围设备的速度。

- 外围时钟门控

运行模式下，独立的外设或存储所对应的 AHB 时钟（HCLK）和 APB 时钟（PCLK）可在在不访问的时钟停止以节省功耗。为了进一步降低功耗，可以通过操控系统寄存器对相关的 IP 或外设的时钟进行门控。

6.3.4.2 睡眠模式

进入睡眠模式

睡眠模式是通过执行 WFI (Wait for Interrupt) 或 WFE (Wait for Event) 进入。有两个方式可供选择进入睡眠模式，这取决于 Cortex®-M4F 系统寄存器中的 SLEEPONEXIT 位。

在睡眠模式中，所有 I/O 引脚保持与运行模式一样的状态。

- 立刻睡眠 (Sleep-now)

1. SLEEPDEEP = 0

2. SLEEPONEXIT = 0
3. 没有中断 (针对 WFI) 或事件 (针对 WFE) 未决
- 退出时睡眠 (Sleep-on-exit)
4. SLEEPDEEP = 0
5. SLEEPONEXIT = 1
6. 没有中断未决
7. WFI 或 WFE

MCU 在处理完最低优先级 ISR 之后立刻进入睡眠模式。

从睡眠模式唤醒

如果进入睡眠模式是使用 WFI 指令，那么任何被嵌套的矢量中断控制器 (NVIC) 确认的外设中断都可以将设备从睡眠模式唤醒。

如果进入睡眠模式是使用 WFE 指令，一旦发生唤醒事件，MCU 就会退出睡眠模式。唤醒事件可以是：通过外设控制寄存器启用中断，而不是在 NVIC，并在 Cortex®-M4F 系统控制寄存器中使能 SEVONPEND 位。当 MCU 从 WFE 指令恢复，外设的中断未决位和外设 NVIC IRQ 通道未决位 (在 NVIC 中断清除未决寄存器) 都必须清除。

以下表格详细说明了睡眠模式：

表 20 睡眠模式

睡眠模式	描述
模式进入	<p>WFI (Wait for Interrupt) 或 WFE (wait for event) 的同时：</p> <p>SLEEPDEEP = 0 位于 Cortex®-M4F 系统控制寄存器 (SCR) 位 2；</p> <p>无中断 (针对 WFI) 或 事件 (针对 WFE) 未决</p> <p>参考 Cortex®-M4F 系统寄存器</p>
	<p>处理完 ISR 的同时：</p> <p>SLEEPDEEP = 0 位于 Cortex®-M4F 系统控制寄存器 (SCR) 位 2；</p> <p>SLEEPONEXIT = 1 位于 Cortex®-M4F 系统控制寄存器 (SCR) 位 1；</p> <p>无中断未决</p> <p>参考 Cortex®-M4F 系统寄存器</p>

睡眠模式	描述
模式退出	<p>如果 WFI 或处理完 ISR 返回用于进入： 中断（在 NVIC 中相应的中断向量必须使能） 如果 WFE 用于进入而且 SEVONPEND = 0 唤醒事件：参照章节《唤醒事件管理》 如果 WFE 用于进入而且 SEVONPEND = 1 中断事件（NVIC 中不使能） 唤醒事件（章节《唤醒事件管理》）</p>
唤醒延迟	无

深度睡眠模式

深度睡眠模式是基于 Cortex®-M4F 深度睡眠模式加上外设时钟门控组合而成。

- 在该模式，所有源于 SCLK 的时钟均停止
- 与 FCLK 异步的模块可在进入该模式前配置好
- PLL 和内部 RC 振荡器默认都处于活跃状态
- SRAM 和寄存器的内容保留
- 所有 I/O 引脚保持与运行模式一样的状态

进入深度睡眠模式

如果 FLASH 存储烧写正在进行中，那进入深度睡眠就会推迟直至烧写结束。如果访问 APB 域正在进行中，那进入深度睡眠就会推迟直至 APB 访问结束。

1. 设置那些不与 FCLK 同步的 IP 时钟进入门控或者释放状态
2. 设置 FCLK 时钟源（切换至低速时钟或者其他）
3. 设置 DIG_PMU_EN = 1
4. 设置 SLEEPDEEP = 1
5. WFI 或 WFE

在深度睡眠模式，通过对独立模块编程可选择实现以下特性：

- 看门狗 (WDT): 看门狗在复位或上电后是否开启取决于存储 FLASH 信息页字节中的控制位。看门狗的配置寄存器可以修改, 但首先得输出正确的密钥。在进入低功耗模式之前用户可视应用场景将看门狗关闭。
- 基准定时器 (BTM): BTM 时钟在进入低功耗模式时默认开启, 但它也可在进入低功耗模式之前关闭

从深度睡眠模式中唤醒

当处理器检测任何具有足够优先级非屏蔽的中断时唤醒系统。系统返回活跃模式, CPU 和受影响模块继续运行。

如果 WFI 用于模式进入:

- 一些特别外设 (WDT/ BTM)的中断应编程为激活状态 (相关外设应编程处于激活状态及相关中断应在 NVIC 中使能)

如果 WFE 用于模式进入:

- 事件模式

KPM32R14TX 能够处理外部或内部事件以唤醒 CPU 内核 (WFE)。唤醒事件可通过以下方式产生: 在外设控制寄存器中使能中断, 而不是 NVIC; 在 Cortex®-M4F 系统控制寄存器中使能 SEVONPEND 位。当 MCU 从 WFE 恢复时, 必须清除外设 NVIC IRQ 通道未决位。

下面的表格详述了深度睡眠模式。

表 21 深度睡眠模式

深度睡眠模式	描述
模式进入	<p>WFI (Wait for Interrupt) 或 WFE (Wait for Event) 的同时: 无中断 (针对 WFI) 或事件 (针对 WFE) 未决; SLEEPDEEP = 1 在 Cortex®-M4F 系统控制寄存器 (SCR) 位 2; DIG_PMU_EN bit = 1 位于功耗模式寄存器 (PWR_MOD) 位 0;</p> <p>处理完 ISR 的同时: 无中断未决; SLEEPDEEP = 1 位于 Cortex®-M4F 系统控制寄存器 (SCR) 位 2; SLEEPONEXIT = 1 位于 Cortex®-M4F 系统控制寄存器 (SCR) 位 1; DIG_PMU_EN bit = 1 位于功耗模式寄存器 (PWR_MOD) 位 0;</p>
模式退出	<p>如果 WFI 或处理完 ISR 返回用于模式进入: 具备唤醒功能的 IP (WDT/BTM/IOW/SCI/ADC/CMP/IOW) 配置为中断模式 (相关的 IP 中断向量需要 NVIC 中使能) 如果 WFE 用于模式进入而且 SEVONPEND = 0 唤醒事件: 参照章节《唤醒事件管理》 如果 WFE 用于模式进入而且 SEVONPEND = 1 具备唤醒功能的 IP (WDT/BTM/IOW/SCI/ADC/CMP/IOW) 配置为中断模式 (相关的 IP 中断向量需要 NVIC 中关闭) 唤醒事件 (见章节《唤醒事件管理》)</p>
唤醒延迟	无

6.3.4.3 停止模式

相对比睡眠模式和深度睡眠模式来说，停止模式更加全面。用户可以使 MCU 在任何工作条件下进入停止模式以实现动态功耗降低。

- 在停止模式下，所有源于 FCLK 的时钟和 FCLK 本身都被门控，HOSC&PLL 均由软件关电，HIRC 由硬件状态机实现关电。对于 LIRC 来说,在进入停止模式前应该基于应用场景配置好。
- SRAM 和寄存器的内容保留
- 所有 I/O 引脚保持与进入停止模式前一样的状态
- 在系统寄存器中生成未决状态位 (0x40010010[18])
- 系统可被具备唤醒能力并配置为中断模式的外设 (WDT/BTM/IOW) 唤醒。对于 WDT/BTM 的时钟源不能配置为 HIRC，建议使用低速时钟 LIRC。
- 系统退出停止模式后继续运行

进入停止模式

进入停止模式按照以下步骤执行：

1. 设置 FCLK 源于 MCLK，MCLK 源于 HIRC，关闭 PLL 和 HOSC
2. 设置那些源于 LIRC 的 IP 时钟状态（门控或释放），例如 WDT 和 BTM
3. 如果有需要，使能 eFlash 深度待机控制位 (0x4001_0070[4]=1'b1)
4. 设置 STOP_MODE_EN 和 DIG_PMU_EN (0x4001_0070[1:0]=2'b11)
5. SLEEPDEEP = 1
6. WFI 或 WFE

在停止模式下，通过编程独立的功能模块可以选择实现以下功能：

- 看门狗 (WDT)：看门狗在复位或上电后是否开启取决于存储 FLASH 信息页字节中的控制位。看门狗的配置寄存器可以修改，但首先得输出正确的密钥。在停止模式中，WDT 不能通过编程关闭。
- 基准定时器 (BTM)：BTM 可以在停止模式下工作，可用作唤醒源或其他用途。BTM 的相关配置寄存器需要在进入停止模式前配置好
- IOW：IOW 可以在停止模式下工作，可用作唤醒源或其他用途，那需要在进入停止模式前使能。

退出停止模式

当处理器检测到来自 WDT、和 BTM 的唤醒中断或来自 IOW 的唤醒边沿时便唤醒系统。系统返回至活跃模式，CPU 和受影响的模块会继续运行。

停止模式如下表所示：

表 22 停止模式

停止模式	描述
模式进入	<p>WFI (Wait for Interrupt) 或 WFE (Wait for Event) 的同时：</p> <p>无中断（针对 WFI）或事件（针对 WFE）未决；</p> <p>设置 FCLK 源于 MCLK，MCLK 源于 HIRC，关闭 PLL 和 HOSC；</p> <p>SLEEPDEEP=1 位于 Cortex®-M4F 系统控制寄存器 (SCR) 位 2；</p> <p>STOP_MODE_EN=1&DIG_PMU_EN=1 位于功耗模式控制寄存器 (PWR_MOD)；</p>
	<p>处理完 ISR 的同时：</p> <p>无中断未决；</p> <p>设置 FCLK 源于 MCLK，MCLK 源于 HIRC，关闭 PLL 和 HOSC；</p> <p>SLEEPDEEP=1 位于 Cortex®-M4F 系统控制寄存器 (SCR) 位 2；</p> <p>SLEEPONEXIT=1 位于 Cortex®-M4F 系统控制寄存器 (SCR) 位 1；</p> <p>STOP_MODE_EN=1&DIG_PMU_EN=1 位于功耗模式控制寄存器 (PWR_MOD)；</p>
模式退出	<p>如果 WFI 或处理完 ISR 返回用于模式进入：</p> <p>具备唤醒功能的 IP (WDT/BTM/IOW) 配置为中断模式（相关的 IP 中断向量需要 NVIC 中使能）。</p> <p>如果 WFE 用于模式进入而且 SEVONPEND = 0</p> <p>唤醒事件：参照章节《唤醒事件管理》</p> <p>如果 WFE 用于模式进入而且 SEVONPEND = 1</p> <p>具备唤醒功能的 IP (WDT/BTM/IOW 配置为中断模式（相关的 IP 中断向量需要 NVIC 中关闭）</p> <p>唤醒事件（见章节《唤醒事件管理》）</p>
唤醒延迟	内部 RC 振荡器 (8MHz) 唤醒时间 + FLASH 唤醒时间

唤醒事件管理

唤醒事件可以通过以下方式生成：

- 在外设控制寄存器使能一个中断，但不在 NVIC，同时也在 Cortex®-M4F 系统控制寄存器 (SCR[4]) 使能 SEVONPEND 位。当 MCU 从 WFE 恢复，外设中断未决位和外设 NVIC IRQ 通道未决位（位于 NVIC 中断清除未决寄存器）必须清除。
- 或配置一个 EBUS 通道（参照 EBUS 规格书）至 CPU RXEV（一个输入高电平导致 Cortex®-M4F 处理器的事件寄存器置位）。当 MCU 从 WFE 恢复，外设中断未决位和外设 NVIC IRQ 通道未决位就没有必要清除，因为事件的相关未决位并未置位。（注：EBUS 事件不适用于停止模式）

Kiwi Instruments Corp. Confidential

7 系统寄存器 (System Register)

系统寄存器模块提供了用于系统控制和芯片配置的寄存器。该模块的主要功能如下：

- 芯片 ID 信息
- 系统配置信息
- 复位状态信息
- 时钟源选择、时钟分频和时钟门控
- 模拟 IP 控制
- 端口多路复用控制
- IOPAD 控制
- SRAM 控制

7.1 寄存器

系统寄存器的基地址是 0x40010000

表 23 系统寄存器映射表

偏移地址	寄存器列表	位宽	访问	复位值
0x000~0x004	Reserved	32	res	0x00000000
0x008	芯片 ID 寄存器	32	ro	0x0800FF00
0x00C	Reserved	32	res	0x00000000
0x010	复位状态寄存器	32	r/w	0x0B007001
0x018	Reserved	32	res	0x00000000
0x01C	Reserved	32	res	0x00000000
0x020	系统 NMI 状态寄存器	32	r/w	0x00000000
0x024~0x05C	Reserved	32	res	0x00000000
0x060	系统控制寄存器	32	r/w	0x00000003
0x064~0x06C	Reserved	32	res	0x00000000
0x070	功耗模式控制寄存器	32	r/w	0x00000000
0x074~0x0F8	Reserved	32	res	0x00000000
0x0FC	系统配置密钥寄存器	32	r/w	0x00000000
0x100	模拟PMU控制寄存器	32	r/w	0x001001A0

偏移地址	寄存器列表	位宽	访问	复位值
0x104~0x10C	Reserved	32	res	0x00000000
0x110	IRC 控制寄存器 0	32	r/w	0x00000000
0x114	IRC 控制寄存器 1	32	r/w	0x00842210
0x118	IRC 控制寄存器 2	32	r/w	0x00002210
0x11C	IRC 控制寄存器 3	32	r/w	0x00000011
0x120~0x12C	Reserved	32	res	0x00000000
0x130	HOSC 控制寄存器	32	r/w	0x00013420
0x130~0x14C	Reserved	32	res	0x00000000
0x150	PLL 控制寄存器	32	r/w	0x36448020
0x154~0x15C	Reserved	32	res	0x00000000
0x160	温度传感器控制寄存器	32	r/w	0x00000002
0x164~0x19C	Reserved	32	res	0x00000000
0x1A0	时钟频率监测寄存器 0	32	r/w	0x00000000
0x1A4	时钟频率监测寄存器 1	32	r/w	0xFFFF0000
0x1A8	时钟频率监测寄存器 2	32	r/w	0x00000000
0x1AC	Reserved	32	res	0x00000000
0x1B0	时钟监控器寄存器	32	r/w	0x00000000
0x1B4~0x1FC	Reserved	32	res	0x00000000
0x200	电压检测模块 1 控制寄存器	32	r/w	0x00000611
0x204	电压检测模块 1 状态寄存器	32	r/w	0x00000000
0x208	电压检测模块 2 控制寄存器	32	r/w	0x00000A00
0x20C	电压检测模块 2 状态寄存器	32	r/w	0x00000000
0x210~0x27C	Reserved	32	res	0x00000000
0x280	复位控制寄存器 0	32	r/w	0x00000000
0x284	复位控制寄存器 1	32	r/w	0x00000000
0x288~0x28C	Reserved	32	res	0x00000000
0x290	时钟门控寄存器 0	32	r/w	0x80000000
0x294	时钟门控寄存器 1	32	r/w	0x00000000
0x298~0x29C	Reserved	32	res	0x00000000
0x2A0	时钟选择寄存器 0	32	r/w	0x00000000
0x2A4	时钟选择寄存器 1	32	r/w	0x00000000
0x2A8	时钟选择寄存器 2	32	r/w	0x00000000

偏移地址	寄存器列表	位宽	访问	复位值
0x2AC~0x2BC	Reserved	32	res	0x00000000
0x2C0	噪声滤波时钟控制寄存器	32	r/w	0x00000000
0x2C4~0x2FC	Reserved	32	res	0x00000000
0x300	IO 噪声滤波控制寄存器 0	32	r/w	0xFFFFFFFF
0x304	IO 噪声滤波控制寄存器 1	32	r/w	0x3FFFFFFF
0x308~0x31C	Reserved	32	res	0x00000000
0x320	端口输入使能控制寄存器 0	32	r/w	0xFFFFFFFF
0x324	端口输入使能控制寄存器 1	32	r/w	0x3FFFFFFF
0x328~0x33C	Reserved	32	res	0x00000000
0x340	引脚驱动控制寄存器 0	32	r/w	0x00000000
0x344	引脚驱动控制寄存器 1	32	r/w	0x00000000
0x348~0x35C	Reserved	32	res	0x00000000
0x360	引脚上拉控制寄存器 0	32	r/w	0x00002000
0x364	引脚上拉控制寄存器 1	32	r/w	0x00000000
0x368~0x37C	Reserved	32	res	0x00000000
0x380	引脚下拉控制寄存器 0	32	r/w	0x00000000
0x384	引脚下拉控制寄存器 1	32	r/w	0x00000000
0x388~0x39C	Reserved	32	res	0x00000000
0x3A0	端口模式控制寄存器 0	32	r/w	0x04400000
0x3A4	端口模式控制寄存器 1	32	r/w	0x00000000
0x3A8	端口模式控制寄存器 2	32	r/w	0x00000000
0x3AC	端口模式控制寄存器 3	32	r/w	0x00000000
0x3B0~0x3DC	Reserved	32	res	0x00000000
0x3E0	功能复用选择寄存器 0	32	r/w	0x00000000
0x3E4	功能复用选择寄存器 1	32	r/w	0x00202000
0x3E8	功能复用选择寄存器 2	32	r/w	0x00000000
0x3EC	功能复用选择寄存器 3	32	r/w	0x00000000
0x3F0	功能复用选择寄存器 4	32	r/w	0x00000000
0x3F4	功能复用选择寄存器 5	32	r/w	0x00000000
0x3F8	功能复用选择寄存器 6	32	r/w	0x00000000
0x3FC	功能复用选择寄存器 7	32	r/w	0x00000000
0x400~0x45C	Reserved	32	res	0x00000000

偏移地址	寄存器列表	位宽	访问	复位值
0x460	端口开漏输出控制寄存器 0	32	r/w	0x00000000
0x464	端口开漏输出控制寄存器 1	32	r/w	0x00000000
0x468~0x4BC	Reserved	32	res	0x00000000
0x4C0	调试控制寄存器	32	r/w	0x00000003
0x500	SRAM 状态寄存器	32	r/w	0x00000000
0x504	SRAM 调试信息控制寄存器	32	r/w	0x20000000
0x508~0x7FC	Reserved	32	res	0x00000000

7.1.1 芯片 ID 寄存器 (CHIP_ID)

偏移地址: 0x0008

表 24 芯片 ID 寄存器

位	访问	描述
31:16	ro	CHIP_ID: 芯片身份 ID KPM32R14TX 则是 0x0800 复位值: 0x0800
15:8	ro	PKG_BOND_OPT: 封装选项信息 复位值: 0xFF
7:0	ro	CHIP_VER: 芯片版本信息 8'H00: 试验版本 复位值: 0x0

7.1.2 复位状态寄存器 (RST_ST)

偏移地址: 0x0010

表 25 复位状态寄存器

位	访问	描述
31:29	ro	Reserved 复位值: 0x0
28	ro	PLL_LOCK: PLL 输出时钟状态 0: PLL 输出时钟尚未稳定 1: PLL 输出时钟已稳定 复位值: 0x0
27	ro	HIRC2_LOCK: HIRC2 输出时钟状态 0: HIRC2 输出时钟尚未稳定 1: HIRC2 输出时钟已稳定 复位值: 0x1

位	访问	描述
26	ro	HOSC_LOCK: HOSC 输出时钟状态 0: HOSC 输出时钟尚未稳定 1: HOSC 输出时钟已稳定 复位值: 0x0
25	ro	LIRC_LOCK: LIRC 输出时钟状态 0: LIRC 输出时钟尚未稳定 1: LIRC 输出时钟已稳定 复位值: 0x1
24	ro	HIRC_LOCK: HIRC 输出时钟状态 0: HIRC 输出时钟尚未稳定 1: HIRC 输出时钟已稳定 复位值: 0x1
23:19	ro	Reserved 复位值: 0x0
18	ro	FROM_STOP_FLAG: 系统从停止 (STOP) 低功耗状态返回与否 0: 非 1: 是 复位值: 0x0
17:15	ro	Reserved 复位值: 0x0
14	ro	OPT_READ_CNT_FLAG: 选项信息读取计数器状态 0: 异常 1: 正常释放 复位值: 0x1
13	ro	FLSH_WU_CNT_FLAG: FLASH 唤醒计数器状态 0: 异常 1: 正常释放 复位值: 0x1
12	ro	POR_WARMUP_CNT_FLAG: POR 热身计数器状态 0: 异常 1: 正常释放 复位值: 0x1
11:4	Res	Reserved 复位值: 0x0
3	r/w	SYSRESETREQ_ST: 系统复位是否为软件置位 SYSRESETREQ 导致 0: 非 1: 是 注: 写 1 清零 复位值: 0x0
2	r/w	LOCKUP_ST: 系统复位是否为 ARM 核 LOCKUP 事件导致 0: 非

位	访问	描述
		1: 是 注: 写 1 清零 复位值: 0x0
1	r/w	WDT_RST_ST: 系统复位是否为看门狗复位事件导致 0: 非 1: 是 注: 写 1 清零 复位值: 0x0
0	r/w	POR_ST: 系统复位是否为 POR 或 LVD 复位事件导致 0: 非 1: 是 注: 写 1 清零 复位值: 0x1

7.1.3 系统 NMI 状态寄存器 (NMI_SR)

偏移地址: 0x0020

表 26 系统 NMI 状态寄存器

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5	r/w	PAD_NMI_PEND, PAD 不可屏蔽中断未决状态 0: 状态已清除或未检测到校验错误 1: 检测到 PAD 不可屏蔽中断 注: 该位写 1 清零 复位值: 0x0
4	r/w	CACHE_PARITY_ERR_PEND, CACHE 校验错误未决状态 0: 状态已清除或未检测到校验错误 1: 检测到 CACHE SRAM 校验错误 注: 该位写 1 清零 复位值: 0x0
3	r/w	SRAM_PARITY_ERR_PEND, SRAM 校验错误未决状态 0: 状态已清除或未检测到校验错误 1: 检测到 SRAM 校验错误 注: 该位写 1 清零 复位值: 0x0
2	r/w	FRQ_ERR_PEND, 待测时钟频率错误未决状态 0: 状态已清除或未检测到待测时钟频率错误 1: 检测到待测时钟频率错误 注: 该位写 1 清零

位	访问	描述
		复位值: 0x0
1	r/w	PLL_LOSS_PEND, PLL 时钟丢失未决状态 0: 状态已清除或未发生过时钟丢失 1: 发生过时钟丢失 注: 该位写 1 清零 复位值: 0x0
0	r/w	MCLK_LOSS_PEND, MCLK 时钟丢失未决状态 0: 状态已清除或未发生过时钟丢失 1: 发生过时钟丢失 注: 该位写 1 清零 复位值: 0x0

7.1.4 系统控制寄存器 (SYS_CFG)

偏移地址: 0x0060

表 27 系统控制寄存器

位	访问	描述
31:9	Res	Reserved 复位值: 0x0
8	r/w	PAD_NMI_EN: PAD 不可屏蔽中断使能控制 0: 关闭 1: 开启 复位值: 0x0
7:5	Res	Reserved 复位值: 0x0
4	r/w	LOCKUP_EN: ARM 核 LOCKUP 事件触发复位使能控制 0: 关闭 1: 开启 复位值: 0x0
3:2	Res	Reserved 复位值: 0x0
1:0	r/w	MEM_MODE_CTRL: 存储地址映射模式控制 00: 0x0000_0000 映射到 SRAM (0x2000_0000) 01, 10: 0x0000_0000 映射到 BootROM (0x1FFF_0000) 11: 0x0000_0000 映射到 flash 主页 (0x0800_0000) 注: 为防止误操作, 在更新该控制位的同时需要写数据的高 16 位输入 0xFEED 方能成功操作 (即 WDATA[31:16]= 0xFEED), 如若是其他值则不能更新该控制位。 复位值: 取决于 FLASH 信息页中的 BOOT_SEQUENCE_CFG [1:0]

7.1.5 功耗模式控制寄存器 (PWR_MOD)

偏移地址: 0x0070

表 28 功耗模式控制寄存器

位	访问	描述
31:5	Res	Reserved 复位值: 0x0
4	r/w	EFLH_STDBY_EN: 芯片低功耗模式下 Eflash 待机模式使能控制 0: 关闭 1: 开启 复位值: 0x0
3:2	Res	Reserved 复位值: 0x0
1	r/w	STOP_MODE_EN: 停止模式使能控制 0: 关闭 1: 开启 复位值: 0x0
0	r/w	DIG_PMU_EN: 数字 PMU 使能控制 0: 关闭 1: 开启 复位值: 0x0

7.1.6 系统配置密钥寄存器 (SYS_SET_KR)

偏移地址: 0x00FC

表 29 系统配置密钥寄存器 (写)

位	访问	描述
31:0	wo	SYS_SET_UNLOCK_KEY: 系统配置寄存器解锁密钥 注: 在默认情况下, 系统配置寄存器 (PMU, *IRC, LVD, *OSC, CLK_SEL_REG0 等) 是不能直接进行写操作, 在写之前需要输入正确密钥 0x2EE1D879 以解锁写权限, 即 SYS_SET_UNLOCK=1, 在更新完系统配置寄存器之后, 写入其他值便可重新锁定写权限, 即 SYS_SET_UNLOCK=0。 复位值: 0x0

表 30 系统配置密钥寄存器 (读)

位	访问	描述
31:30	Res	Reserved 复位值: 0x0
0	ro	SYS_SET_UNLOCK, 系统配置寄存器解锁与否 0: 锁定 1: 解锁 复位值: 0x0

7.1.7 模拟 PMU 控制寄存器 (ANA_PMU_CTRL)

偏移地址: 0x0100

表 31 模拟 PMU 控制寄存器

位	访问	描述
31:21	Res	Reserved 复位值: 0x0
20:16	r/w	VREF_TM: BandGAP Trimming 配置 复位值: 5'b10000
15:12	Res	Reserved 复位值: 0x0
11	r/w	LDO_DSSEL, LDO 输出驱动能力选择 0: 120mA 1: 210mA 复位值: 0x0
10:8	r/w	PMU_IBSEL: PMU 偏置电流选择控制 PMU_IBSEL [2], 带隙基准电流偏置设置 0: 250nA 1: 500nA PMU_IBSEL [1:0], LDO 电流偏置设置 00: 0.5μA 01: 1.0μA 10: 1.5μA 11: 2.0μA 复位值: 0x1
7	r/w	BG_TMT[1], 基准电压温度系数校正 (位 1) 复位值: 1
6:5	r/w	LDO_SEL[2:1], LDO输出电压控制 (位 1&2) 复位值: 01
4	r/w	BG_TMT[0], 基准电压温度系数校正 (位 0) 复位值: 0
3:0	Res	Reserved 复位值: 0x0

在更新该组寄存器之前需要确保完成系统配置寄存器安全锁解锁, 即 SYS_SET_UNLOCK 有效。

7.1.8 IRC 控制寄存器 0 (IRC_CTRL_REG0)

偏移地址: 0x0110

表 32 IRC 控制寄存器 0

位	访问	描述
31:17	Res	Reserved 复位值: 0x0
16	r/w	LIRC_PD: LIRC 控制 0: 开启 1: 断电 复位值: 0x0
15:2	Res	Reserved 复位值: 0x0
1	r/w	HIRC2_PD: HIRC2 控制 0: 开启 1: 断电 复位值: 0x0
0	r/w	HIRC_PD: HIRC 控制 0: 开启 1: 断电 注: 当 MCLK 源选择源于 HIRC 时, 该位不能置 1; 当 PLL 参考时钟源于 HIRC 并且 FCLK 源于 PLL 时, 该位不能置 1。 复位值: 0x0

在更新该组寄存器之前需要确保完成系统配置寄存器安全锁解锁, 即 SYS_SET_UNLOCK 有效。

7.1.9 IRC 控制寄存器 1 (IRC_CTRL_REG1)

偏移地址: 0x0114

表 33 IRC 控制寄存器 1

位	访问	描述
31:24	Res	Reserved 复位值: 0x0
23:20	r/w	LIRC_TMF: LIRC 精细调整 周期: 25.30 μ s ~ 36.80 μ s 调整步进: 0.75 μ s, 0.6% 频率: 27.18kHz ~ 39.53kHz 范围: -18.43% ~ 21.70% 复位值: 0x8 (31.25 μ s)
19	Res	Reserved 复位值: 0x0
18:16	r/w	LIRC_TMC, LIRC 粗略调整 频率: 22.03kHz ~ 53.64kHz 步进: 3.82 μ s, 3.06% 范围: -40.77% ~ 65.14%

位	访问	描述
		复位值: 0x4 (32K)
15:14	Res	Reserved 复位值: 0x0
13:10	r/w	HIRC_TMT: HIRC 温度调整 调整范围: -3.4%~2.66% 步进: 0.17% 复位值: 0x8
9:5	r/w	HIRC_TMF: HIRC 精细调整 Fine Trimming (T) 周期: 115.62ns ~ 129.76ns 调整步进: 0.456ns, 0.36% 频率: 7.70666MHz ~ 8.64878MHz 复位值: 0x10 (125ns)
4:0	r/w	HIRC_TMC, HIRC 粗略调整 (Coarse Trimming) 频率: 3.87MHz ~ 12.04MHz 步进: 0.25M 复位值: 0x10 (8MHz)

在更新该组寄存器之前需要确保完成系统配置寄存器安全锁解锁, 即 SYS_SET_UNLOCK 有效。

7.1.10 IRC 控制寄存器 2 (IRC_CTRL_REG2)

偏移地址: 0x0118

表 34 IRC 控制寄存器 2

位	访问	描述
31:14	Res	Reserved 复位值: 0x0
13:10	r/w	HIRC2_TMT: HIRC2 温度调整 调整范围: -3.4%~2.66% 步进: 0.17% 复位值: 0x8
9:5	r/w	HIRC2_TMF: HIRC2 精细调整 Fine Trimming (T) 周期: 115.62ns ~ 129.76ns 调整步进: 0.456ns, 0.36% 频率: 7.70666MHz ~ 8.64878MHz 复位值: 0x10 (125ns)
4:0	r/w	HIRC2_TMC, HIRC2 粗略调整 (Coarse Trimming) 频率: 3.87MHz ~ 12.04MHz 步进: 0.25M 复位值: 0x10 (8MHz)

7.1.11 IRC 控制寄存器 3 (IRC_CTRL_REG3)

偏移地址: 0x011C

表 35 IRC 控制寄存器 3

位	访问	描述
31:9	Res	Reserved 复位值: 0x0
8	r/w	LIRC_CKOE: LIRC 时钟输出使能 (测试模式) 0: 关闭 1: 开启 复位值: 0x0
7	Res	Reserved 复位值: 0x0
6	r/w	HIRC2_CKOE: HIRC2 时钟输出使能 (测试模式) 0: 关闭 1: 开启 复位值: 0x0
5:4	r/w	HIRC2_DIVT: HIRC2 分频选择 (测试模式) 00: 1MHz 01: 2MHz 10: 4MHz 11: 8MHz 复位值: 0x1
3	Res	Reserved 复位值: 0x0
2	r/w	HIRC_CKOE: HIRC 时钟输出使能 (测试模式) 0: 关闭 1: 开启 复位值: 0x0
1:0	r/w	HIRC_DIVT: HIRC 分频选择 (测试模式) 00: 1MHz 01: 2MHz 10: 4MHz 11: 8MHz 复位值: 0x1

7.1.12 HOSC 控制寄存器 (HOSC_CTRL_REG)

偏移地址: 0x0130

表 36 HOSC 控制寄存器

位	访问	描述
31	r/w	HOUT_ASEL: XHOUT 功能选择 0: 普通功能 1: XHOUT

位	访问	描述
		复位值: 0x0
30	r/w	HIN_ASEL: XHIN 功能选择 0: 普通功能 1: XHIN 复位值: 0x0
29:19	Res	Reserved 复位值: 0x0
18	r/w	HOSC_CKOE: HOSC 时钟输出使能 (测试模式) 0: 关闭 1: 开启 复位值: 0x0
17:16	r/w	HOSC_DIVT: HOSC 分频选择 (测试模式) 00: 1MHz 01: 2MHz 10: 4MHz 11: 8MHz 复位值: 0x1
15	Res	Reserved 复位值: 0x0
14:12	r/w	HOSC_LT, HOSC 稳定时间选择 000: $2^8/f_{HOSC}$ 100: $2^{12}/f_{HOSC}$ 001: $2^9/f_{HOSC}$ 101: $2^{13}/f_{HOSC}$ 010: $2^{10}/f_{HOSC}$ 110: $2^{14}/f_{HOSC}$ 011: $2^{11}/f_{HOSC}$ 111: $2^{15}/f_{HOSC}$ 复位值: 0x3
11	Res	Reserved 复位值: 0x0
10:9	r/w	HOSC_RESEL, HOSC 内部并联电阻选择 00: 1.8MΩ 01: 1.4MΩ 10: 1.0MΩ 11: 0.6MΩ 复位值: 0x2
8	r/w	HOSC_RESEN, 内部并联电阻使能 0: 关闭 1: 使能 复位值: 0x0
7:4	r/w	HOSC_GM: HOSC GM 参数调整 0000: 2.717m 1000: 8.155m 0001: 3.358m 1001: 8.832m 0010: 4.292m 1010: 9.511m 0011: 4.746m 1011: 10.160m 0100: 5.423m 1100: 10.838m 0101: 6.109m 1101: 11.535m 0110: 6.733m 1110: 12.192m 0111: 7.446m 1111: 12.864m 复位值: 0x2

位	访问	描述
3:2	Res	Reserved 复位值: 0x0
1	r/w	EXTCLK_EN, 外部时钟输入使能 0: 关闭 1: 使能 复位值: 0x0
0	r/w	HOSC_EN: HOSC 使能控制 0: 关闭 1: 开启 注意: 当 MCLK 源选择源于 HOSC 时, 该位不能置 0; 当 PLL 参考时钟源于 HOSC 并且 FCLK 源于 PLL 时, 该位不能置 0。 复位值: 0x0

在更新该组寄存器之前需要确保完成系统配置寄存器安全锁解锁, 即 SYS_SET_UNLOCK 有效。

7.1.13 PLL 控制寄存器 (PLL_CTRL_REG)

偏移地址: 0x0150

表 37 PLL 控制寄存器

位	访问	描述
31	r/w	PLL_CKOE: PLL 时钟输出使能 (测试模式) 0: 关闭 1: 开启 复位值: 0x0
30:28	r/w	PLL_LDO, PLL LDO 输出选择 000: 1.10V 001: 1.15V 010: 1.20V 011: 1.25V 100: 1.30V 101: 1.35V 110: 1.40V 111: 1.45V 复位值: 0x3
27:26	r/w	PLL_CPC: PLL 电荷泵电流选择 00: 8μA 01: 10μA 10: 12μA 11: 14μA 复位值: 0x1

位	访问	描述
25:24	r/w	PLL_KVCO, VCO 电压转电流增益调节控制 00: 14.13μA/V 01: 17.00μA/V 10: 21.60μA/V 11: 29.89μA/V 复位值: 0x2
23:20	r/w	PLL_DIVM, PLL 输入时钟分频选择 M 0000: 1 1000: 8 0001: 1 1001: 9 0010: 2 1010: 10 0011: 3 1011: 11 0100: 4 1100: 12 0101: 5 1101: 13 0110: 6 1110: 14 0111: 7 1111: 15 复位值: 0x4
19:17	Res	Reserved 复位值: 0x0
16:12	r/w	PLL_MUTN: PLL 输入时钟倍数 N 配置选择 PLL 的 VCO 工作频率是输入参考时钟频率乘以 N, $F_{vco}=F_{in}*N$ (其中 $N=128+PLL_MUTN$)。假设输入频率 2MHz, VCO 频率与 N 的关系如下: 000_0000: 128 000_0001: 128 + 1 000_0010: 128 + 2 000_0011: 128 + 3 ... 000_0000: 128 + Value ... 100_0000: 128 + 64 (192) ... 100_1000: 128 + 72 (200) ... 000_0000: 128 + 127 (255) 复位值: 0x48
11:9	Res	Reserved 复位值: 0x0
8:4	r/w	PLL_DIVSYS: PLL 输出分频控制 频率计算可表达为 $F_{PLL}=F_{vco}/PLL_DIVSYS$, 假设 VCO 频率为 400MHz: 0000_0000~0000_0011: Reserved 0000_0100: 400/4 = 100 0000_0101: 400/5 = 80 ... 0110_0100: 400/100 = 4 ... 1100_1000: 400/200 = 2 ... 1111_1111: 400/255 = 1.568 复位值: 0x2

位	访问	描述
3	Res	Reserved 复位值: 0x0
2:1	r/w	PLL_CKISEL: PLL 时钟输入源选择控制 00: HIRC 01: HOSC 10: HIRC2 11: Reserved 复位值: 0x0
0	r/w	PLL_EN: PLL 使能控制 0: 关闭 1: 开启 注: 当FCLK源选择源于 PLL 时, 该位不能置 0 复位值: 0x0

在更新该组寄存器之前需要确保完成系统配置寄存器安全锁解锁, 即 SYS_SET_UNLOCK 有效。

7.1.14 温度传感器控制寄存器 (TS_CTRL_REG)

偏移地址: 0x0160

表 38 温度传感器控制寄存器

位	访问	描述
31:3	Res	Reserved 复位值: 0x0
2	r/w	LDO_SEL[0], LDO输出电压控制 (位 0) 复位值: 0
1	r/w	LDO_SEL [3], LDO输出电压控制 (位 3) 复位值: 1
0	r/w	TS_EN: 温度传感器控制使能 0: 关闭 1: 开启 复位值: 0x0

7.1.15 时钟频率监测寄存器 0 (CLK_FRQ_MNT_REG0)

偏移地址: 0x01A0

表 39 时钟频率监测寄存器 0

位	访问	描述
31:19	Res	Reserved 复位值: 0x0
18:16	r/w	OBJ_CLK_SEL, 量测目标时钟选择 000: FCLK 001: HIRC 010: HIRC2 011: HOSC

位	访问	描述
		其他：保留 复位值：0x0
15	Res	Reserved 复位值：0x0
14:12	r/w	REF_CLK_DIV，参考时钟分频 000：不分频 001：32 分频 010：64 分频 011：128 分频 100：256 分频 101：512 分频 110：1024 分频 111：2048 分频 复位值：0x0
11	Res	Reserved 复位值：0x0
10:8	r/w	REF_CLK_SEL，参考时钟选择 00：HIRC2 01：HOSC 10：LIRC 11：EXT_REFCLK 复位值：0x0
7:2	Res	Reserved 复位值：0x0
1	r/w	FRQ_ERR_PEND_EN：频率错误事件至 NMI 使能控制 0：关闭 1：开启 复位值：0x0
0	r/w	FRQ_MNT_EN：频率监控器使能控制 0：关闭 1：开启 复位值：0x0

7.1.16 时钟频率监控寄存器 1 (CLK_FRQ_MNT_REG1)

偏移地址：0x01A4

表 40 时钟频率监测寄存器 1

位	访问	描述
31:16	r/w	PRD_THD_HIGH：量测时钟周期数比较上限值 注：基于参考时钟时钟周期内，量测时钟自行计数，如果 CLK_PRD 大于 PRD_THD_HIGH，FRQ_ERR_PEND 便会置位 复位值：0xFFFF
15:0	r/w	PRD_THD_LOW：量测时钟周期数比较下限值 注：基于参考时钟时钟周期内，量测时钟自行计数，如果 CLK_PRD 小于 PRD_THD_LOW，FRQ_ERR_PEND 便会置位 复位值：0x0

PRD_THD_HIGH 的设置值必须大于 PRD_THD_LOW。

7.1.17 时钟频率监控寄存器 2 (CLK_FRQ_MNT_REG2)

偏移地址: 0x01A8

表 41 时钟频率监测寄存器 2

位	访问	描述
31:17	Res	Reserved 复位值: 0x0
16	r/w	MEASURE_END: 量测结束标记 0: 量测尚在进行中 1: 已完成一次量测 注: 该位写 1 清零 复位值: 0x0
15:0	ro	CLK_PRD: 量测时钟周期数 基于参考时钟时钟周期内, 量测时钟自行计数值 复位值: 0x0

7.1.18 时钟监控器寄存器 (CLK_MNT_REG)

偏移地址: 0x01B0

表 42 时钟监控器寄存器

位	访问	描述
31:25	Res	Reserved 复位值: 0x0
24	r/w	PLL_BAK_VLD, PLL 备份时钟生效状态 0: PLL_CLK 源于 PLL 输出 1: PLL_CLK 源于备份时钟 HIRC2 注: 一旦该状态置位, 意味着 PLL 时钟可能受到破坏处于不稳定状态, PLL 输入至系统的时钟会自动切换到 HIRC2 时钟。该位写 1 清零, 同时输入时钟也会恢复位 PLL 时钟输入。 复位值: 0x0
23:18	Res	Reserved 复位值: 0x0
17	r/w	PLL_LOSS_PEND_EN: PLL 丢失事件至 NMI 使能控制 0: 关闭 1: 开启 复位值: 0x0
16	r/w	PLL_MNT_EN: PLL 监控器使能控制 0: 关闭 1: 开启 注: 该监控器仅适用于PLL输出频率为 50MHz~100MHZ 复位值: 0x0
15:9	Res	Reserved 复位值: 0x0
8	r/w	MCLK_BAK_VLD, MCLK 备份时钟生效状态 0: MCLK 源于正常 HOSC 或 HIRC 1: MCLK 源于备份时钟 HIRC2

位	访问	描述
		注：一旦该状态置位，意味着 MCLK 时钟可能受到破坏处于不稳定状态，MCLK输入至系统的时钟会自动切换到 HIRC2 时钟。该位写 1 清零，同时输入时钟也会恢复位 MCLK 时钟输入。 复位值：0x0
7:2	Res	Reserved 复位值：0x0
1	r/w	MCLK_LOSS_PEND_EN: MCLK 丢失事件至 NMI 使能控制 0: 关闭 1: 开启 复位值：0x0
0	r/w	MCLK_MNT_EN: MCLK 监控器使能控制 0: 关闭 1: 开启 复位值：0x0

7.1.19 电压检测模块 1 控制寄存器 (VD1_CTRL)

偏移地址：0x0200

表 43 电压检测模块 1 控制寄存器

位	访问	描述
31:12	Res	Reserved 复位值：0x0
11:10	r/w	VD1_SEL[3:2] 电压检测模块 1 档位选择控制 (位 2&3) 注：该位只能为 POR 复位或外部复位 复位值：0x1 注： VD1_SEL, 电压检测模块 1 档位选择控制 0000: 1.80+0.05V/-0.05V 0001: 1.90+0.05V/-0.05V 0010: 2.00+0.05V/-0.05V 0011: 2.20+0.05V/-0.05V 0100: 2.40+0.05V/-0.05V 0101: 2.60+0.05V/-0.05V 0110: 2.80+0.05V/-0.05V 0111: 3.00+0.05V/-0.05V 1000: 3.20+0.05V/-0.05V 1001: 3.40+0.05V/-0.05V 1010: 3.60+0.05V/-0.05V 1011: 3.80+0.05V/-0.05V 1100: 4.00+0.05V/-0.05V 1101: 4.20+0.05V/-0.05V 1110: 4.40+0.05V/-0.05V 1111: 4.50+0.05V/-0.05V
9	r/w	VD2_SEL[0]电压检测模块 2 档位选择控制 (位 0) 注： 该位只能为 POR 复位或外部复位 完整定义见 VD2_CTRL[11:9] 处描述

位	访问	描述
		复位值: 0x1
8	r/w	VD1_SEL[0] 电压检测模块1档位选择控制 (位 0) 注: 该位只能为 POR 复位或外部复位 完整定义见 VD1_CTRL[11:10] 处描述 复位值: 0x0
7:5	Res	Reserved 复位值: 0x0
4	Res	VD1_RST_EN: 电压检测模块 1 复位芯片使能控制 0: 关闭 1: 开启 注: 该位只能为 POR 复位或外部复位 复位值: 0x1
3:1	Res	Reserved 复位值: 0x0
0	r/w	VD1_EN: 电压检测模块 1 使能控制 0: 关闭 1: 开启 注: 该位只能为 POR 复位或外部复位 复位值: 0x1

在更新该组寄存器之前需要确保完成系统配置寄存器安全锁解锁, 即 SYS_SET_UNLOCK 有效。

7.1.20 电压检测模块 1 状态寄存器 (VD1_ST)

偏移地址: 0x0204

表 44 电压检测模块 1 状态寄存器

位	访问	描述
31:2	Res	Reserved 复位值: 0x0
1	r/w	VD1_FLG, 电压检测模块 1 标记 0: $VCC \geq VD1_SEL$ 1: $VCC < VD1_SEL$ 注: 当 VD1_OUT=1 时, 该位硬件置位, 支持写 1 清零 复位值: 0x0
0	ro	VD1_OUT, 电压检测模块 1 输出 0: $VCC \geq VD1_SEL$ 1: $VCC < VD1_SEL$ 复位值: 0x0

7.1.21 电压检测模块 2 控制寄存器 (VD2_CTRL)

偏移地址: 0x0208

表 45 电压检测模块 2 控制寄存器

位	访问	描述
31:24	Res	Reserved 复位值: 0x0
23:16	r/w	VD2_THR, 电压检测模块2滤波阈值 该值用于设置对于VD2_OUT输出电平毛刺滤波, 当输出电平有效地保持 VD2_THR个滤波周期, 才认为输出的电平有效 注: 该位只能为POR复位或外部复位。 复位值: 0x0
15	Res	Reserved 复位值: 0x0
14:12	r/w	VD2_FILTER_CLK_PSC, 电压检测模块 2 滤波时钟预分频选择 000: 不分频 001: 2 分频 010: 4 分频 011: 8 分频 100: 16 分频 101: 32 分频 110: 64 分频 111: 128 分频 注: 该位只能为 POR 复位或外部复位。 复位值: 0x0
11:9	r/w	VD2_SEL[3:1] 电压检测模块 2 档位选择控制 注: 该位只能为 POR 复位或外部复位。 复位值: 0x5 注: VD2_SEL, 电压检测模块 2 档位选择控制。 0000: 1.80+0.05V/-0.05V 0001: 1.90+0.05V/-0.05V 0010: 2.00+0.05V/-0.05V 0011: 2.20+0.05V/-0.05V 0100: 2.40+0.05V/-0.05V 0101: 2.60+0.05V/-0.05V 0110: 2.80+0.05V/-0.05V 0111: 3.00+0.05V/-0.05V 1000: 3.20+0.05V/-0.05V 1001: 3.40+0.05V/-0.05V 1010: 3.60+0.05V/-0.05V 1011: 3.80+0.05V/-0.05V 1100: 4.00+0.05V/-0.05V 1101: 4.20+0.05V/-0.05V 1110: 4.40+0.05V/-0.05V 1111: 4.50+0.05V/-0.05V
8	r/w	VD1_SEL[0]电压检测模块1档位选择控制 (位 1) 注: 1. 该位只能为 POR 复位或外部复位;

位	访问	描述
		2. 完整定义见 VD1_CTRL[11:10] 处描述。 复位值: 0x0
7:4	Res	Reserved 复位值: 0x0
3	r/w	VD2_IE: 电压检测模块 2 中断使能控制 0: 关闭 1: 开启 注: 该位只能为 POR 复位或外部复位。 复位值: 0x0
2:1	Res	Reserved 复位值: 0x0
0	r/w	VD2_EN: 电压检测模块 2 使能控制 0: 关闭 1: 开启 注: 该位只能为 POR 复位或外部复位。 复位值: 0x0

在更新该组寄存器之前需要确保完成系统配置寄存器安全锁解锁, 即 SYS_SET_UNLOCK 有效。

7.1.22 电压检测模块 2 状态寄存器 (VD2_ST)

偏移地址: 0x020C

表 46 电压检测模块 2 状态寄存器

位	访问	描述
31:2	Res	Reserved 复位值: 0x0
1	r/w	VD2_FLAG, 电压检测模块 2 标记 0: $VCC \geq VD2_SEL$ 1: $VCC < VD2_SEL$ 该位写 1 清零 复位值: 0x0
0	ro	VD2_OUT, 电压检测模块 2 输出 0: $VCC \geq VD2_SEL$ 1: $VCC < VD2_SEL$ 复位值: 0x0

7.1.23 复位控制寄存器 0 (RST_CTRL_REG0)

偏移地址: 0x0280

表 47 复位控制寄存器 0

位	访问	描述
31:23	Res	Reserved 复位值: 0x0
22	Res	Reserved

位	访问	描述
		复位值: 0x0
21	r/w	TIMG2_SW_RSTJ: TIMG2 软复位与否 0: 是 1: 否 复位值: 0x0
20	r/w	TIMG1_SW_RSTJ: TIMG1 软复位与否 0: 是 1: 否 复位值: 0x0
19	r/w	TIMG0_SW_RSTJ: TIMG0 软复位与否 0: 是 1: 否 复位值: 0x0
18	r/w	TIMH_SW_RSTJ: TIMH 软复位与否 0: 是 1: 否 复位值: 0x0
17	r/w	TIMA1_SW_RSTJ: TIMA1 软复位与否 0: 是 1: 否 复位值: 0x0
16	r/w	TIMA0_SW_RSTJ: TIMA0 软复位与否 0: 是 1: 否 复位值: 0x0
15:13	Res	Reserved 复位值: 0x0
12	r/w	BTM_SW_RSTJ: BTM 软复位与否 0: 是 1: 否 复位值: 0
11: 10	Res	Reserved 复位值: 0x0
9	r/w	ADC1_SW_RSTJ: ADC1 软复位与否 0: 是 1: 否 复位值: 0x0
8	r/w	ADC0_SW_RSTJ: ADC0 软复位与否 0: 是 1: 否 复位值: 0x0
7	r/w	EBUS_SW_RSTJ: EBUS 软复位与否 0: 是 1: 否 复位值: 0x0
6	r/w	TMU_SW_RSTJ: TMU 软复位与否 0: 是

位	访问	描述
		1: 否 复位值: 0x0
5	r/w	DMA_SW_RSTJ: DMA 软复位与否 0: 是 1: 否 复位值: 0x0
4	r/w	GPIO_SW_RSTJ: GPIO 软复位与否 0: 是 1: 否 复位值: 0x0
3:2	Res	Reserved 复位值: 0x0
1	r/w	I2C1_SW_RSTJ: I2C1 软复位与否 0: 是 1: 否 复位值: 0x0
0	r/w	I2C0_SW_RSTJ: I2C0 软复位与否 0: 是 1: 否 复位值: 0x0

7.1.24 复位控制寄存器 1 (RST_CTRL_REG1)

偏移地址: 0x0284

表 48 复位控制寄存器 1

位	访问	描述
31:20	Res	Reserved 复位值: 0x0
19	r/w	TIMS3_SW_RSTJ: TIMS3 软复位与否 0: 是 1: 否 复位值: 0x0
18	r/w	TIMS2_SW_RSTJ: TIMS2 软复位与否 0: 是 1: 否 复位值: 0x0
17	r/w	TIMS1_SW_RSTJ: TIMS1 软复位与否 0: 是 1: 否 复位值: 0x0
16	r/w	TIMS0_SW_RSTJ: TIMS0 软复位与否 0: 是 1: 否 复位值: 0x0
15	Res	Reserved 复位值: 0x0

位	访问	描述
14	r/w	CAN_SW_RSTJ: CAN 软复位与否 0: 是 1: 否 复位值: 0x0
13	r/w	CMP_SW_RSTJ: CMP 软复位与否 0: 是 1: 否 复位值: 0x0
12:11	Res	Reserved 复位值: 0x0
10	r/w	SCI2_SW_RSTJ: SCI2 软复位与否 0: 是 1: 否 复位值: 0x0
9	r/w	SCI1_SW_RSTJ: SCI1 软复位与否 0: 是 1: 否 复位值: 0x0
8	r/w	SCI0_SW_RSTJ: SCI0 软复位与否 0: 是 1: 否 复位值: 0x0
7:3	Res	Reserved 复位值: 0x0
2	r/w	SPI_SW_RSTJ: SPI 软复位与否 0: 是 1: 否 复位值: 0x0
1	r/w	IOW_SW_RSTJ: IOW 软复位与否 0: 是 1: 否 复位值: 0x0
0	Res	Reserved 复位值: 0x0

7.1.25 时钟门控寄存器 0 (CLK_GAT_REG0)

偏移地址: 0x0290

表 49 时钟门控寄存器 0

位	访问	描述
31	r/w	WDT_CLK_GAT: 看门狗时钟门控使能控制 0: 关闭 1: 开启 注: 更新此位前需要完成系统配置寄存器安全锁解锁, 即 SYS_SET_UNLOCK 有效 复位值: 0x1

位	访问	描述
30:26	Res	Reserved 复位值: 0x0
25	r/w	BUZ_CLK_GAT: 蜂鸣器时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
24:23	Res	Reserved 复位值: 0x0
22	Res	Reserved 复位值: 0x0
21	r/w	TIMG2_CLK_GAT: TIMG2 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
20	r/w	TIMG1_CLK_GAT: TIMG1 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
19	r/w	TIMG0_CLK_GAT: TIMG0 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
18	r/w	TIMH_CLK_GAT: TIMH 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
17	r/w	TIMA1_CLK_GAT: TIMA1 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
16	r/w	TIMA0_CLK_GAT: TIMA0 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
15:13	Res	Reserved 复位值: 0x0
12	r/w	BTM_CLK_GAT: BTM 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
11:10	Res	Reserved 复位值: 0x0
9	r/w	ADC1_CLK_GAT: ADC1 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
8	r/w	ADC0_CLK_GAT: ADC0 时钟门控使能控制

位	访问	描述
		0: 关闭 1: 开启 复位值: 0x0
7	r/w	EBUS_CLK_GAT: EBUS 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
6	r/w	TMU_CLK_GAT: TMU 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
5	r/w	DMA_CLK_GAT: DMA 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
4	r/w	GPIO_CLK_GAT: GPIO 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
3:2	Res	Reserved 复位值: 0x0
1	r/w	I2C1_CLK_GAT: I2C1 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
0	r/w	I2C0_CLK_GAT: I2C0 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0

7.1.26 时钟门控寄存器 1 (CLK_GAT_REG1)

偏移地址: 0x0294

表 50 时钟门控寄存器 1

位	访问	描述
31:20	Res	Reserved 复位值: 0x0
19	r/w	TIMS3_CLK_GAT: TIMS3 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
18	r/w	TIMS2_CLK_GAT: TIMS2 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
17	r/w	TIMS1_CLK_GAT: TIMS1 时钟门控使能控制

位	访问	描述
		0: 关闭 1: 开启 复位值: 0x0
16	r/w	TIMSO_CLK_GAT: TIMSO 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
15	Res	Reserved 复位值: 0x0
14	r/w	CAN_CLK_GAT: CAN 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
13	r/w	CMP_CLK_GAT: CMP 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
12:11	Res	Reserved 复位值: 0x0
10	r/w	SCI2_CLK_GAT: SCI2 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
9	r/w	SCI1_CLK_GAT: SCI1 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
8	r/w	SCI0_CLK_GAT: SCI0 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
7:3	Res	Reserved 复位值: 0x0
2	r/w	SPI_CLK_GAT: SPI 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
1	r/w	IOW_CLK_GAT: IOW 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
0	Res	Reserved 复位值: 0x0

7.1.27 时钟选择寄存器 0 (CLK_SEL_REG0)

偏移地址: 0x02A0

表 51 时钟选择寄存器 0

位	访问	描述
31:21	Res	Reserved 复位值: 0x0
20	r/w	FCLK_SWT_ERR, FCLK 源切换错误标记 当 FCLK_SEL_INFO 与 FCLK_SEL 不一致时, 此标记便会置位。该位写 1 清零。 复位值: 0x0
19	Res	Reserved 复位值: 0x0
18	ro	FCLK_SEL_INFO, FCLK 源时钟信息, 指示当前 FCLK 的时钟源: 0: MCLK 1: PLL 复位值: 0x0
17	Res	Reserved 复位值: 0x0
16	ro	MCLK_SEL_INFO, MCLK 源时钟信息, 指示当前 MCLK 的时钟源: 0: HIRC 1: HOSC 复位值: 0x0
15:8	Res	Reserved 复位值: 0x0
7:4	r/w	FCLK_DIV_SEL: FCLK 预分频控制 0: 不分频 (20) 1: 2 分频 (21) 2: 4 分频 (22) ... 7: 128 分频 (27) 8: 256 分频 (28) 其他: 保留 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2	r/w	FCLK_SEL: FCLK 源选择 0: MCLK 1: PLL 注: 如果切换的目标时钟源尚未准备好, 切换操作将会忽略, FCLK 仍将运行在原有的时钟源下。查看 FCLK_SEL_INFO 可获知当下正选中的时钟源。 复位值: 0x0
1	Res	Reserved 复位值: 0x0
0	r/w	MCLK_SEL: MCLK 源选择 0: HIRC 1: HOSC 注: 如果切换的目标时钟源尚未准备好, 切换操作将会忽略, FCLK 仍将运行在原有的时钟源下。查看 MCLK_SEL_INFO 可获知当下正选中的时钟源。 复位值: 0x0

在更新该组寄存器之前需要确保完成系统配置寄存器安全锁解锁, 即 SYS_SET_UNLOCK 有效。

7.1.28 时钟选择寄存器 1 (CLK_SEL_REG1)

偏移地址: 0x02A4

表 52 时钟选择寄存器 1

位	访问	描述
31:12	Res	Reserved 复位值: 0x0
11:8	r/w	BUZ_CLK_SEL: 蜂鸣器时钟选择 0000: BUZ_SRC_CLK 0001: BUZ_SRC_CLK 的 2 分频时钟 0010: BUZ_SRC_CLK 的 4 分频时钟 ... 1101: BUZ_SRC_CLK 的 2 ¹³ 分频时钟 其他值: BUZ_SRC_CLK 的 2 ¹³ 分频时钟 复位值: 0x0
7:0	Res	Reserved 复位值: 0x0

7.1.29 时钟选择寄存器 2 (CLK_SEL_REG2)

偏移地址: 0x02A8

表 53 时钟选择寄存器 2

位	访问	描述
31	Res	Reserved 复位值: 0x0
30:24	r/w	ADC1_CLK_SEL: ADC1 分频时钟选择 该控制位用于配置 FCLK 分频以作为 ADC1 的时钟源之一, 支持 1~128 分频 复位值: 0x0
23	Res	Reserved 复位值: 0x0
22	r/w	BTM_CLK_SEL: BTM 时钟源选择 0: LIRC 1: MCLK 复位值: 0x0
21	Res	Reserved 复位值: 0x0
20	r/w	WDT_CLK_SEL: WDT 时钟源选择 0: LIRC 1: MCLK 注: 更新此位前需要完成系统配置寄存器安全锁解锁, 即 SYS_SET_UNLOCK 有效。 复位值: 0x0
19	Res	Reserved 复位值: 0x0
18:16	r/w	BUZ_SRC_CLK_SEL: 蜂鸣器输出时钟源选择 000: FCLK

位	访问	描述
		001: HIRC 010: LIRC 011: HOSC 100: HIRC2 101: PLL_CLK 110: SCLK 111: HCLK 复位值: 0x0
15:14	r/w	NF_CLK2_SEL: 噪声滤波时钟 2 源选择 00: NF_SRC_CLK 的 512 分频时钟 01: NF_SRC_CLK 的 1024 分频时钟 10: NF_SRC_CLK 的 2048 分频时钟 11: NF_SRC_CLK 的 4096 分频时钟 复位值: 0x0
13:11	r/w	NF_CLK1_SEL: 噪声滤波时钟 1 源选择 000: NF_SRC_CLK 的 32 分频时钟 001: NF_SRC_CLK 的 64 分频时钟 010: NF_SRC_CLK 的 128 分频时钟 011: NF_SRC_CLK 的 256 分频时钟 其他: 保留 复位值: 0x0
10:9	r/w	NF_CLK0_SEL: 噪声滤波时钟 0 源选择 00: NF_SRC_CLK 的 2 分频时钟 01: NF_SRC_CLK 的 4 分频时钟 10: NF_SRC_CLK 的 8 分频时钟 11: NF_SRC_CLK 的 16 分频时钟 复位值: 0x0
8	r/w	NF_SRC_CLK_SEL: 噪声滤波源时钟选择 0: HIRC 1: FCLK 复位值: 0x0
7	Res	Reserved 复位值: 0x0
6:0	r/w	ADC0_CLK_SEL: ADC0 分频时钟选择 该控制位用于配置 FCLK 分频以作为 ADC0 的时钟源之一, 支持 1~128 分频。 复位值: 0x0

7.1.30 噪声滤波时钟控制寄存器 (NF_CLK_CTRL_REG)

偏移地址: 0x02C0

表 54 噪声滤波时钟选择寄存器

位	访问	描述
31:27	Res	Reserved 复位值: 0x0
26	r/w	PD_NF_CLK_EN, D 组 IO 噪声滤波时钟使能控制 0: 关闭

位	访问	描述
		1: 开启 复位值: 0x0
25:24	r/w	PD_NF_CLK_SEL, D 组 IO 噪声滤波时钟选择 00: NF_CLK0 01: NF_CLK1 10: NF_CLK2 11: LIRC 复位值: 0x0
23:19	Res	Reserved 复位值: 0x0
18	r/w	PC_NF_CLK_EN, C 组 IO 噪声滤波时钟使能控制 0: 关闭 1: 开启 复位值: 0x0
17:16	r/w	PC_NF_CLK_SEL, C 组 IO 噪声滤波时钟选择 00: NF_CLK0 01: NF_CLK1 10: NF_CLK2 11: LIRC 复位值: 0x0
15:11	Res	Reserved 复位值: 0x0
10	r/w	PB_NF_CLK_EN, B 组 IO 噪声滤波时钟使能控制 0: 关闭 1: 开启 复位值: 0x0
9:8	r/w	PB_NF_CLK_SEL, B 组 IO 噪声滤波时钟选择 00: NF_CLK0 01: NF_CLK1 10: NF_CLK2 11: LIRC 复位值: 0x0
7:3	Res	Reserved 复位值: 0x0
2	r/w	PA_NF_CLK_EN, A 组 IO 噪声滤波时钟使能控制 0: 关闭 1: 开启 复位值: 0x0
1:0	r/w	PA_NF_CLK_SEL, A 组 IO 噪声滤波时钟选择 00: NF_CLK0 01: NF_CLK1 10: NF_CLK2 11: LIRC 复位值: 0x0

7.1.31 IO 噪声滤波控制寄存器 0 (IO_NF_CTRL_REG0)

偏移地址: 0x0300

表 55 IO 噪声滤波控制寄存器 0

位	访问	描述
31	r/w	PB15_NF_DIS: PB15 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
30	r/w	PB14_NF_DIS: PB14 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
29	r/w	PB13_NF_DIS: PB13 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
28	r/w	PB12_NF_DIS: PB12 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
27	r/w	PB11_NF_DIS: PB11 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
26	r/w	PB10_NF_DIS: PB10 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
25	r/w	PB9_NF_DIS: PB9 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
24	r/w	PB8_NF_DIS: PB8 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
23	r/w	PB7_NF_DIS: PB7 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
22	r/w	PB6_NF_DIS: PB6 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
21	r/w	PB5_NF_DIS: PB5 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
20	r/w	PB4_NF_DIS: PB4 噪声滤波功能关闭控制 0: 开启

位	访问	描述
		1: 关闭 复位值: 0x1
19	r/w	PB3_NF_DIS: PB3 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
18	r/w	PB2_NF_DIS: PB2 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
17	r/w	PB1_NF_DIS: PB1 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
16	r/w	PB0_NF_DIS: PB0 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
15	r/w	PA15_NF_DIS: PA15 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
14	r/w	PA14_NF_DIS: PA14 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
13	r/w	PA13_NF_DIS: PA13 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
12	r/w	PA12_NF_DIS: PA12 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
11	r/w	PA11_NF_DIS: PA11 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
10	r/w	PA10_NF_DIS: PA10 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
9	r/w	PA9_NF_DIS: PA9 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
8	r/w	PA8_NF_DIS: PA8 噪声滤波功能关闭控制 0: 开启

位	访问	描述
		1: 关闭 复位值: 0x1
7	r/w	PA7_NF_DIS: PA7 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
6	r/w	PA6_NF_DIS: PA6 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
5	r/w	PA5_NF_DIS: PA5 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
4	r/w	PA4_NF_DIS: PA4 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
3	r/w	PA3_NF_DIS: PA3 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
2	r/w	PA2_NF_DIS: PA2 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
1	r/w	PA1_NF_DIS: PA1 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
0	r/w	PA0_NF_DIS: PA0 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1

7.1.32 IO 噪声滤波控制寄存器 1 (IO_NF_CTRL_REG1)

偏移地址: 0x0304

表 56 IO 噪声滤波控制寄存器 1

位	访问	描述
31:30	Res	Reserved 复位值: 0x0
29	r/w	PD13_NF_DIS: PD13 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1

位	访问	描述
28	r/w	PD12_NF_DIS: PD12 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
27	r/w	PD11_NF_DIS: PD11 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
26	r/w	PD10_NF_DIS: PD10 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
25	r/w	PD9_NF_DIS: PD9 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
24	r/w	PD8_NF_DIS: PD8 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
23	r/w	PD7_NF_DIS: PD7 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
22	r/w	PD6_NF_DIS: PD6 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
21	r/w	PD5_NF_DIS: PD5 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
20	r/w	PD4_NF_DIS: PD4 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
19	r/w	PD3_NF_DIS: PD3 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
18	r/w	PD2_NF_DIS: PD2 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
17	r/w	PD1_NF_DIS: PD1 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1

位	访问	描述
16	r/w	PD0_NF_DIS: PD0 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
15	r/w	PC15_NF_DIS: PC15 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
14	r/w	PC14_NF_DIS: PC14 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
13	r/w	PC13_NF_DIS: PC13 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
12	r/w	PC12_NF_DIS: PC12 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
11	r/w	PC11_NF_DIS: PC11 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
10	r/w	PC10_NF_DIS: PC10 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
9	r/w	PC9_NF_DIS: PC9 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
8	r/w	PC8_NF_DIS: PC8 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
7	r/w	PC7_NF_DIS: PC7 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
6	r/w	PC6_NF_DIS: PC6 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
5	r/w	PC5_NF_DIS: PC5 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1

位	访问	描述
4	r/w	PC4_NF_DIS: PC4 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
3	r/w	PC3_NF_DIS: PC3 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
2	r/w	PC2_NF_DIS: PC2 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
1	r/w	PC1_NF_DIS: PC1 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1
0	r/w	PC0_NF_DIS: PC0 噪声滤波功能关闭控制 0: 开启 1: 关闭 复位值: 0x1

Kiwi Instruments Corp. Confidential

7.1.33 端口输入使能控制寄存器 0 (PAD_IE_CTRL_REG0)

偏移地址: 0x0320

表 57 端口输入使能控制寄存器 0

位	访问	描述
31	r/w	PB15_IE, PB15 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
30	r/w	PB14_IE, PB14 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
29	r/w	PB13_IE, PB13 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
28	r/w	PB12_IE, PB12 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
27	r/w	PB11_IE, PB11 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
26	r/w	PB10_IE, PB10 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
25	r/w	PB9_IE, PB9 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
24	r/w	PB8_IE, PB8 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
23	r/w	PB7_IE, PB7 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
22	r/w	PB6_IE, PB6 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
21	r/w	PB5_IE, PB5 输入使能控制 0: 关闭

位	访问	描述
		1: 开启 复位值: 0x1
20	r/w	PB4_IE, PB4 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
19	r/w	PB3_IE, PB3 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
18	r/w	PB2_IE, PB2 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
17	r/w	PB1_IE, PB1 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
16	r/w	PB0_IE, PB0 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
15	r/w	PA15_IE, PA15 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
14	r/w	PA14_IE, PA14 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
13	r/w	PA13_IE, PA13 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
12	r/w	PA12_IE, PA12 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
11	r/w	PA11_IE, PA11 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
10	r/w	PA10_IE, PA10 输入使能控制 0: 关闭 1: 开启 复位值: 0x1

位	访问	描述
9	r/w	PA9_IE, PA9 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
8	r/w	PA8_IE, PA8 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
7	r/w	PA7_IE, PA7 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
6	r/w	PA6_IE, PA6 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
5	r/w	PA5_IE, PA5 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
4	r/w	PA4_IE, PA4 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
3	r/w	PA3_IE, PA3 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
2	r/w	PA2_IE, PA2 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
1	r/w	PA1_IE, PA1 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
0	r/w	PA0_IE, PA0 输入使能控制 0: 关闭 1: 开启 复位值: 0x1

7.1.34 端口输入使能控制寄存器 1 (PAD_IE_CTRL_REG1)

偏移地址: 0x0324

表 58 端口输入使能控制寄存器 1

位	访问	描述
31:30	Res	Reserved 复位值: 0x0
29	r/w	PD13_IE, PD13 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
28	r/w	PD12_IE, PD12 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
27	r/w	PD11_IE, PD11 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
26	r/w	PD10_IE, PD10 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
25	r/w	PD9_IE, PD9 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
24	r/w	PD8_IE, PD8 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
23	r/w	PD7_IE, PD7 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
22	r/w	PD6_IE, PD6 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
21	r/w	PD5_IE, PD5 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
20	r/w	PD4_IE, PD4 输入使能控制 0: 关闭 1: 开启

位	访问	描述
		复位值: 0x1
19	r/w	PD3_IE, PD3 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
18	r/w	PD2_IE, PD2 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
17	r/w	PD1_IE, PD1 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
16	r/w	PD0_IE, PD0 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
15	r/w	PC15_IE, PC15 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
14	r/w	PC14_IE, PC14 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
13	r/w	PC13_IE, PC13 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
12	r/w	PC12_IE, PC12 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
11	r/w	PC11_IE, PC11 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
10	r/w	PC10_IE, PC10 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
9	r/w	PC9_IE, PC9 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
8	r/w	PC8_IE, PC8 输入使能控制 0: 关闭

位	访问	描述
		1: 开启 复位值: 0x1
7	r/w	PC7_IE, PC7 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
6	r/w	PC6_IE, PC6 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
5	r/w	PC5_IE, PC5 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
4	r/w	PC4_IE, PC4 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
3	r/w	PC3_IE, PC3 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
2	r/w	PC2_IE, PC2 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
1	r/w	PC1_IE, PC1 输入使能控制 0: 关闭 1: 开启 复位值: 0x1
0	r/w	PC0_IE, PC0 输入使能控制 0: 关闭 1: 开启 复位值: 0x1

7.1.35 引脚驱动控制寄存器 0 (IO_DS_CTRL_REG0)

偏移地址: 0x0340

表 59 引脚驱动控制寄存器 0

位	访问	描述
31	r/w	PB15_DS, PB15 驱动能力选择 参照 PA0
30	r/w	PB14_DS, PB14 驱动能力选择 参照 PA0
29	r/w	PB13_DS, PB13 驱动能力选择 参照 PA0
28	r/w	PB12_DS, PB12 驱动能力选择 参照 PA0
27	r/w	PB11_DS, PB11 驱动能力选择 参照 PA0
26	r/w	PB10_DS, PB10 驱动能力选择 参照 PA0
25	r/w	PB9_DS, PB9 驱动能力选择 参照 PA0
24	r/w	PB8_DS, PB8 驱动能力选择 参照 PA0
23	r/w	PB7_DS, PB7 驱动能力选择 参照 PA0
22	r/w	PB6_DS, PB6 驱动能力选择 参照 PA0
21	r/w	PB5_DS, PB5 驱动能力选择 参照 PA0
20	r/w	PB4_DS, PB4 驱动能力选择 参照 PA0
19	r/w	PB3_DS, PB3 驱动能力选择 参照 PA0
18	r/w	PB2_DS, PB2 驱动能力选择 参照 PA0
17	r/w	PB1_DS, PB1 驱动能力选择 参照 PA0
16	r/w	PB0_DS, PB0 驱动能力选择 参照 PA0
15	r/w	PA15_DS, PA15 驱动能力选择 参照 PA0
14	r/w	PA14_DS, PA14 驱动能力选择 参照 PA0
13	r/w	PA13_DS, PA13 驱动能力选择 参照 PA0

位	访问	描述
12	r/w	PA12_DS, PA12 驱动能力选择 参照 PA0
11	r/w	PA11_DS, PA11 驱动能力选择 参照 PA0
10	r/w	PA10_DS, PA10 驱动能力选择 参照 PA0
9	r/w	PA9_DS, PA9 驱动能力选择 参照 PA0
8	r/w	PA8_DS, PA8 驱动能力选择 参照 PA0
7	r/w	PA7_DS, PA7 驱动能力选择 参照 PA0
6	r/w	PA6_DS, PA6 驱动能力选择 参照 PA0
5	r/w	PA5_DS, PA5 驱动能力选择 参照 PA0
4	r/w	PA4_DS: PA3 驱动能力选择 参照 PA0
3	r/w	PA3_DS, PA3 驱动能力选择 参照 PA0
2	r/w	PA2_DS, PA2 驱动能力选择 参照 PA0
1	r/w	PA1_DS, PA1 驱动能力选择 参照 PA0
0	r/w	PA0_DS, PA0 驱动能力选择 0: 5mA 1: 10mA 复位值: 0x0

7.1.36 引脚驱动控制寄存器 1 (IO_DS_CTRL_REG1)

偏移地址: 0x0344

表 60 引脚驱动控制寄存器 1

位	访问	描述
31:30	Res	Reserved 复位值: 0x0
29	r/w	PD13_DS, PD13 驱动能力选择 参照 PC0
28	r/w	PD12_DS, PD12 驱动能力选择 参照 PC0
27	r/w	PD11_DS, PD11 驱动能力选择 参照 PC0
26	r/w	PD10_DS, PD10 驱动能力选择 参照 PC0
25	r/w	PD9_DS, PD9 驱动能力选择 参照 PC0
24	r/w	PD8_DS, PD8 驱动能力选择 参照 PC0
23	r/w	PD7_DS, PD7 驱动能力选择 参照 PC0
22	r/w	PD6_DS, PD6 驱动能力选择 参照 PC0
21	r/w	PD5_DS, PD5 驱动能力选择 参照 PC0
20	r/w	PD4_DS, PD4 驱动能力选择 参照 PC0
19	r/w	PD3_DS, PD3 驱动能力选择 参照 PC0
18	r/w	PD2_DS, PD2 驱动能力选择 参照 PC0
17	r/w	PD1_DS, PD1 驱动能力选择 参照 PC0
16	r/w	PD0_DS, PD0 驱动能力选择 参照 PC0
15	r/w	PC15_DS, PC15 驱动能力选择 参照 PC0
14	r/w	PC14_DS, PC14 驱动能力选择 参照 PC0
13	r/w	PC13_DS, PC13 驱动能力选择 参照 PC0
12	r/w	PC12_DS, PC12 驱动能力选择 参照 PC0
11	r/w	PC11_DS, PC11 驱动能力选择 参照 PC0
10	r/w	PC10_DS, PC10 驱动能力选择 参照 PC0

位	访问	描述
9	r/w	PC9_DS, PC9 驱动能力选择 参照 PC0
8	r/w	PC8_DS, PC8 驱动能力选择 参照 PC0
7	r/w	PC7_DS, PC7 驱动能力选择 参照 PC0
6	r/w	PC6_DS, PC6 驱动能力选择 参照 PC0
5	r/w	PC5_DS, PC5 驱动能力选择 参照 PC0
4	r/w	PC4_DS, PC4 驱动能力选择 参照 PC0
3	r/w	PC3_DS, PC3 驱动能力选择 参照 PC0
2	r/w	PC2_DS, PC2 驱动能力选择 参照 PC0
1	r/w	PC1_DS, PC1 驱动能力选择 参照 PC0
0	r/w	PC0_DS, PC0 驱动能力选择 0: 5mA 1: 10mA 复位值: 0x0

7.1.37 引脚上拉控制寄存器 0 (IO_PU_CTRL_REG0)

偏移地址: 0x0360

表 61 引脚上拉控制寄存器 0

位	访问	描述
31	r/w	PB15_PU, PB15 上拉使能控制 参照 PA0_PU
30	r/w	PB14_PU, PB14 上拉使能控制 参照 PA0_PU
29	r/w	PB13_PU, PB13 上拉使能控制 参照 PA0_PU
28	r/w	PB12_PU, PB12 上拉使能控制 参照 PA0_PU
27	r/w	PB11_PU, PB11 上拉使能控制 参照 PA0_PU
26	r/w	PB10_PU, PB10 上拉使能控制 参照 PA0_PU
25	r/w	PB9_PU, PB9 上拉使能控制 参照 PA0_PU
24	r/w	PB8_PU, PB8 上拉使能控制 参照 PA0_PU

位	访问	描述
23	r/w	PB7_PU, PB7 上拉使能控制 参照 PA0_PU
22	r/w	PB6_PU, PB6 上拉使能控制 参照 PA0_PU
21	r/w	PB5_PU, PB5 上拉使能控制 参照 PA0_PU
20	r/w	PB4_PU, PB4 上拉使能控制 参照 PA0_PU
19	r/w	PB3_PU, PB3 上拉使能控制 参照 PA0_PU
18	r/w	PB2_PU, PB2 上拉使能控制 参照 PA0_PU
17	r/w	PB1_PU, PB1 上拉使能控制 参照 PA0_PU
16	r/w	PB0_PU, PB0 上拉使能控制 参照 PA0_PU
15	r/w	PA15_PU, PA15 上拉使能控制 参照 PA0_PU
14	r/w	PA14_PU, PA14 上拉使能控制 参照 PA0_PU
13	r/w	PA13_PU, PA13 上拉使能控制 参照 PA0_PU 复位值: 0x1
12	r/w	PA12_PU, PA12 上拉使能控制 参照 PA0_PU
11	r/w	PA11_PU, PA11 上拉使能控制 参照 PA0_PU
10	r/w	PA10_PU, PA10 上拉使能控制 参照 PA0_PU
9	r/w	PA9_PU, PA9 上拉使能控制 参照 PA0_PU
8	r/w	PA8_PU, PA8 上拉使能控制 参照 PA0_PU
7	r/w	PA7_PU, PA7 上拉使能控制 参照 PA0_PU
6	r/w	PA6_PU, PA6 上拉使能控制 参照 PA0_PU
5	r/w	PA5_PU, PA5 上拉使能控制 参照 PA0_PU
4	r/w	PA4_PU, PA4 上拉使能控制 参照 PA0_PU
3	r/w	PA3_PU, PA3 上拉使能控制 参照 PA0_PU

位	访问	描述
2	r/w	PA2_PU, PA2 上拉使能控制 参照 PA0_PU
1	r/w	PA1_PU, PA0 上拉使能控制 参照 PA0_PU
0	r/w	PA0_PU, PA0 上拉使能控制 0: 关闭 1: 开启 复位值: 0x0

7.1.38 引脚上拉控制寄存器 1 (IO_PU_CTRL_REG1)

偏移地址: 0x0364

表 62 引脚上拉控制寄存器 1

位	访问	描述
31:30	Res	Reserved 复位值: 0x0
29	r/w	PD13_PU, PD13 上拉使能控制 参照 PC0_PU
28	r/w	PD12_PU, PD12 上拉使能控制 参照 PC0_PU
27	r/w	PD11_PU, PD11 上拉使能控制 参照 PC0_PU
26	r/w	PD10_PU, PD10 上拉使能控制 参照 PC0_PU
25	r/w	PD9_PU, PD9 上拉使能控制 参照 PC0_PU
24	r/w	PD8_PU, PD8 上拉使能控制 参照 PC0_PU
23	r/w	PD7_PU, PD7 上拉使能控制 参照 PC0_PU
22	r/w	PD6_PU, PD6 上拉使能控制 参照 PC0_PU
21	r/w	PD5_PU, PD5 上拉使能控制 参照 PC0_PU
20	r/w	PD4_PU, PD4 上拉使能控制 参照 PC0_PU
19	r/w	PD3_PU, PD3 上拉使能控制 参照 PC0_PU
18	r/w	PD2_PU, PD2 上拉使能控制 参照 PC0_PU
17	r/w	PD1_PU, PD1 上拉使能控制 参照 PC0_PU
16	r/w	PD0_PU, PD0 上拉使能控制 参照 PC0_PU
15	r/w	PC15_PU, PC15 上拉使能控制 参照 PC0_PU

位	访问	描述
14	r/w	PC14_PU, PC14 上拉使能控制 参照 PC0_PU
13	r/w	PC13_PU, PC13 上拉使能控制 参照 PC0_PU
12	r/w	PC12_PU, PC12 上拉使能控制 参照 PC0_PU
11	r/w	PC11_PU, PC11 上拉使能控制 参照 PC0_PU
10	r/w	PC10_PU, PC10 上拉使能控制 参照 PC0_PU
9	r/w	PC9_PU, PC9 上拉使能控制 参照 PC0_PU
8	r/w	PC8_PU, PC8 上拉使能控制 参照 PC0_PU
7	r/w	PC7_PU, PC7 上拉使能控制 参照 PC0_PU
6	r/w	PC6_PU, PC6 上拉使能控制 参照 PC0_PU
5	r/w	PC5_PU, PC5 上拉使能控制 参照 PC0_PU
4	r/w	PC4_PU, PC4 上拉使能控制 参照 PC0_PU
3	r/w	PC3_PU, PC3 上拉使能控制 参照 PC0_PU
2	r/w	PC2_PU, PC2 上拉使能控制 参照 PC0_PU
1	r/w	PC1_PU, PC1 上拉使能控制 参照 PC0_PU
0	r/w	PC0_PU, PC0 上拉使能控制 0: 关闭 1: 开启 复位值: 0x0

7.1.39 引脚下拉控制寄存器 0 (IO_PD_CTRL_REG0)

偏移地址: 0x0380

表 63 引脚下拉控制寄存器 0

位	访问	描述
31	r/w	PB15_PD, PB15 下拉使能控制 参照 PA0_PD
30	r/w	PB14_PD, PB14 下拉使能控制 参照 PA0_PD
29	r/w	PB13_PD, PB13 下拉使能控制 参照 PA0_PD

位	访问	描述
28	r/w	PB12_PD, PB12 下拉使能控制 参照 PA0_PD
27	r/w	PB11_PD, PB11 下拉使能控制 参照 PA0_PD
26	r/w	PB10_PD, PB10 下拉使能控制 参照 PA0_PD
25	r/w	PB9_PD, PB9 下拉使能控制 参照 PA0_PD
24	r/w	PB8_PD, PB8 下拉使能控制 参照 PA0_PD
23	r/w	PB7_PD, PB7 下拉使能控制 参照 PA0_PD
22	r/w	PB6_PD, PB6 下拉使能控制 参照 PA0_PD
21	r/w	PB5_PD, PB5 下拉使能控制 参照 PA0_PD
20	r/w	PB4_PD, PB4 下拉使能控制 参照 PA0_PD
19	r/w	PB3_PD, PB3 下拉使能控制 参照 PA0_PD
18	r/w	PB2_PD, PB2 下拉使能控制 参照 PA0_PD
17	r/w	PB1_PD, PB1 下拉使能控制 参照 PA0_PD
16	r/w	PB0_PD, PB0 下拉使能控制 参照 PA0_PD
15	r/w	PA15_PD, PA15 下拉使能控制 参照 PA0_PD
14	r/w	PA14_PD, PA14 下拉使能控制 参照 PA0_PD
13	r/w	PA13_PD, PA13 下拉使能控制 参照 PA0_PD
12	r/w	PA12_PD, PA12 下拉使能控制 参照 PA0_PD
11	r/w	PA11_PD, PA11 下拉使能控制 参照 PA0_PD
10	r/w	PA10_PD, PA10 下拉使能控制 参照 PA0_PD
9	r/w	PA9_PD, PA9 下拉使能控制 参照 PA0_PD
8	r/w	PA8_PD, PA8 下拉使能控制

位	访问	描述
		参照 PA0_PD
7	r/w	PA7_PD, PA7 下拉使能控制 参照 PA0_PD
6	r/w	保留
5	r/w	PA5_PD, PA5 下拉使能控制 参照 PA0_PD
4	r/w	PA4_PD, PA4 下拉使能控制 参照 PA0_PD
3	r/w	PA3_PD, PA3 下拉使能控制 参照 PA0_PD
2	r/w	PA2_PD, PA2 下拉使能控制 参照 PA0_PD
1	r/w	PA1_PD, PA0 下拉使能控制 参照 PA0_PD
0	r/w	PA0_PD, PA0 下拉使能控制 0: 关闭 1: 开启 复位值: 0x0

7.1.40 引脚下拉控制寄存器 1 (IO_PD_CTRL_REG1)

偏移地址: 0x0384

表 64 引脚下拉控制寄存器 1

位	访问	描述
31:30	Res	Reserved 复位值: 0x0
29	r/w	PD13_PU, PD13 下拉使能控制 参照 PC0_PU
28	r/w	PD12_PU, PD12 下拉使能控制 参照 PC0_PU
27	r/w	PD11_PU, PD11 下拉使能控制 参照 PC0_PU
26	r/w	PD10_PU, PD10 下拉使能控制 参照 PC0_PU
25	r/w	PD9_PU, PD9 下拉使能控制 参照 PC0_PU
24	r/w	PD8_PU, PD8 下拉使能控制 参照 PC0_PU
23	r/w	PD7_PU, PD7 下拉使能控制 参照 PC0_PU
22	r/w	PD6_PU, PD6 下拉使能控制 参照 PC0_PU

位	访问	描述
21	r/w	PD5_PU, PD5 下拉使能控制 参照 PC0_PU
20	r/w	PD4_PU, PD4 下拉使能控制 参照 PC0_PU
19	r/w	PD3_PU, PD3 下拉使能控制 参照 PC0_PU
18	r/w	PD2_PU, PD2 下拉使能控制 参照 PC0_PU
17	r/w	PD1_PU, PD1 下拉使能控制 参照 PC0_PU
16	r/w	PD0_PU, PD0 下拉使能控制 参照 PC0_PU
15	r/w	PC15_PU, PC15 下拉使能控制 参照 PC0_PU
14	r/w	PC14_PU, PC14 下拉使能控制 参照 PC0_PU
13	r/w	PC13_PU, PC13 下拉使能控制 参照 PC0_PU
12	r/w	PC12_PU, PC12 下拉使能控制 参照 PC0_PU
11	r/w	PC11_PU, PC11 下拉使能控制 参照 PC0_PU
10	r/w	PC10_PU, PC10 下拉使能控制 参照 PC0_PU
9	r/w	PC9_PU, PC9 下拉使能控制 参照 PC0_PU
8	r/w	PC8_PU, PC8 下拉使能控制 参照 PC0_PU
7	r/w	PC7_PU, PC7 下拉使能控制 参照 PC0_PU
6	r/w	PC6_PU, PC6 下拉使能控制 参照 PC0_PU
5	r/w	PC5_PU, PC5 下拉使能控制 参照 PC0_PU
4	r/w	PC4_PU, PC4 下拉使能控制 参照 PC0_PU
3	r/w	PC3_PU, PC3 下拉使能控制 参照 PC0_PU
2	r/w	PC2_PU, PC2 下拉使能控制 参照 PC0_PU
1	r/w	PC1_PU, PC1 下拉使能控制 参照 PC0_PU
0	r/w	PC0_PU, PC0 下拉使能控制 0: 关闭 1: 开启 复位值: 0x0

7.1.41 端口模式控制寄存器 0 (IO_MOD_CTRL_REG0)

偏移地址: 0x03A0

表 65 端口模式控制寄存器 0

位	访问	描述
31:30	r/w	PA15_MOD, PA15 模式选择 参照 PA0_MOD
29:28	r/w	PA14_MOD, PA14 模式选择 参照 PA0_MOD
27:26	r/w	PA13_MOD, PA13 模式选择 参照 PA0_MOD 复位值: 0x1
25:24	r/w	PA12_MOD, PA12 模式选择 参照 PA0_MOD
23:22	r/w	PA11_MOD, PA11 模式选择 参照 PA0_MOD 复位值: 0x1
21:20	r/w	PA10_MOD, PA10 模式选择 参照 PA0_MOD
19:18	r/w	PA9_MOD, PA9 模式选择 参照 PA0_MOD
17:16	r/w	PA8_MOD, PA8 模式选择 参照 PA0_MOD
15:14	r/w	PA7_MOD, PA7 模式选择 参照 PA0_MOD
13:12	r/w	PA6_MOD, PA6 模式选择 参照 PA0_MOD
11:10	r/w	PA5_MOD, PA5 模式选择 参照 PA0_MOD
9:8	r/w	PA4_MOD, PA4 模式选择 参照 PA0_MOD
7:6	r/w	PA3_MOD, PA3 模式选择 参照 PA0_MOD
5:4	r/w	PA2_MOD, PA2 模式选择 参照 PA0_MOD
3:2	r/w	PA1_MOD, PA1 模式选择 参照 PA0_MOD
1:0	r/w	PA0_MOD, PA0 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0

7.1.42 端口模式控制寄存器 1 (IO_MOD_CTRL_REG1)

偏移地址: 0x03A4

表 66 端口模式控制寄存器 1

位	访问	描述
31:30	r/w	PB15_MOD, PB15 模式选择 参照 PB0_MOD
29:28	r/w	PB14_MOD, PB14 模式选择 参照 PB0_MOD
27:26	r/w	PB13_MOD, PB13 模式选择 参照 PB0_MOD
25:24	r/w	PB12_MOD, PB12 模式选择 参照 PB0_MOD
23:22	r/w	PB11_MOD, PB11 模式选择 参照 PB0_MOD
21:20	r/w	PB10_MOD, PB10 模式选择 参照 PB0_MOD
19:18	r/w	PB9_MOD, PB9 模式选择 参照 PB0_MOD
17:16	r/w	PB8_MOD, PB8 模式选择 参照 PB0_MOD
15:14	r/w	PB7_MOD, PB7 模式选择 参照 PB0_MOD
13:12	r/w	PB6_MOD, PB6 模式选择 参照 PB0_MOD
11:10	r/w	PB5_MOD, PB5 模式选择 参照 PB0_MOD
9:8	r/w	PB4_MOD, PB4 模式选择 参照 PB0_MOD
7:6	r/w	PB3_MOD, PB3 模式选择 参照 PB0_MOD
5:4	r/w	PB2_MOD, PB2 模式选择 参照 PB0_MOD
3:2	r/w	PB1_MOD, PB1 模式选择 参照 PB0_MOD
1:0	r/w	PB0_MOD, PB0 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0

7.1.43 端口模式控制寄存器 2 (IO_MOD_CTRL_REG2)

偏移地址: 0x03A8

表 67 端口模式控制寄存器 1

位	访问	描述
31:30	r/w	PC15_MOD, PC15 模式选择 参照 PC0_MOD
29:28	r/w	PC14_MOD, PC14 模式选择 参照 PC0_MOD
27:26	r/w	PC13_MOD, PC13 模式选择 参照 PC0_MOD
25:24	r/w	PC12_MOD, PC12 模式选择 参照 PC0_MOD
23:22	r/w	PC11_MOD, PC11 模式选择 参照 PC0_MOD
21:20	r/w	PC10_MOD, PC10 模式选择 参照 PC0_MOD
19:18	r/w	PC9_MOD, PC9 模式选择 参照 PC0_MOD
17:16	r/w	PC8_MOD, PC8 模式选择 参照 PC0_MOD
15:14	r/w	PC7_MOD, PC7 模式选择 参照 PC0_MOD
13:12	r/w	PC6_MOD, PC6 模式选择 参照 PC0_MOD
11:10	r/w	PC5_MOD, PC5 模式选择 参照 PC0_MOD
9:8	r/w	PC4_MOD, PC4 模式选择 参照 PC0_MOD
7:6	r/w	PC3_MOD, PC3 模式选择 参照 PC0_MOD
5:4	r/w	PC2_MOD, PC2 模式选择 参照 PC0_MOD
3:2	r/w	PC1_MOD, PC1 模式选择 参照 PC0_MOD
1:0	r/w	PC0_MOD, PC0 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0

7.1.44 端口模式控制寄存器 3 (IO_MOD_CTRL_REG3)

偏移地址: 0x03AC

表 68 端口模式控制寄存器 3

位	访问	描述
31:28	Res	Reserved 复位值: 0x0
27:26	r/w	PD13_MOD, PD13 模式选择 参照 PD0_MOD
25:24	r/w	PD12_MOD, PD12 模式选择 参照 PD0_MOD
23:22	r/w	PD11_MOD, PD11 模式选择 参照 PD0_MOD
21:20	r/w	PD10_MOD, PD10 模式选择 参照 PD0_MOD
19:18	r/w	PD9_MOD, PD9 模式选择 参照 PD0_MOD
17:16	r/w	PD8_MOD, PD8 模式选择 参照 PD0_MOD
15:14	r/w	PD7_MOD, PD7 模式选择 参照 PD0_MOD
13:12	r/w	PD6_MOD, PD6 模式选择 参照 PD0_MOD
11:10	r/w	PD5_MOD, PD5 模式选择 参照 PD0_MOD
9:8	r/w	PD4_MOD, PD4 模式选择 参照 PD0_MOD
7:6	r/w	PD3_MOD, PD3 模式选择 参照 PD0_MOD
5:4	r/w	PD2_MOD, PD2 模式选择 参照 PD0_MOD
3:2	r/w	PD1_MOD, PD1 模式选择 参照 PD0_MOD
1:0	r/w	PD0_MOD, PD0 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 模拟信号模式 复位值: 0x0

7.1.45 功能复用选择寄存器 0 (AF_SEL_REG0)

偏移地址: 0x03E0

表 69 功能复用选择寄存器 0

位	访问	描述
31:28	r/w	PA7_AF_SEL, PA7 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
27:24	r/w	PA6_AF_SEL, PA6 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
23:20	r/w	PA5_AF_SEL, PA5 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
19:16	r/w	PA4_AF_SEL, PA4 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
15:12	r/w	PA3_AF_SEL, PA3 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
11:8	r/w	PA2_AF_SEL, PA2 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
7:4	r/w	PA1_AF_SEL, PA1 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
3:0	r/w	PA0_AF_SEL, PA0 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0

7.1.46 功能复用选择寄存器 1 (AF_SEL_REG1)

偏移地址: 0x03E4

表 70 功能复用选择寄存器 1

位	访问	描述
31:28	r/w	PA15_AF_SEL, PA15 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
27:24	r/w	PA14_AF_SEL, PA14 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
23:20	r/w	PA13_AF_SEL, PA13 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x2
19:16	r/w	PA12_AF_SEL, PA12 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
15:12	r/w	PA11_AF_SEL, PA11 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x2
11:8	r/w	PA10_AF_SEL, PA10 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
7:4	r/w	PA9_AF_SEL, PA9 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
3:0	r/w	PA8_AF_SEL, PA8 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0

7.1.47 功能复用选择寄存器 2 (AF_SEL_REG2)

偏移地址: 0x03E8

表 71 功能复用选择寄存器 2

位	访问	描述
31:28	r/w	PB7_AF_SEL, PB7 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
27:24	r/w	PB6_AF_SEL, PB6 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0
23:20	r/w	PB5_AF_SEL, PB5 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
19:16	r/w	PB4_AF_SEL, PB4 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
15:12	r/w	PB3_AF_SEL, PB3 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
11:8	r/w	PB2_AF_SEL, PB2 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
7:4	r/w	PB1_AF_SEL, PB1 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
3:0	r/w	PB0_AF_SEL, PB0 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0

7.1.48 功能复用选择寄存器 3 (AF_SEL_REG3)

偏移地址: 0x03EC

表 72 功能复用选择寄存器 3

位	访问	描述
31:28	r/w	PB15_AF_SEL, PB15 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
27:24	r/w	PB14_AF_SEL, PB14 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
23:20	r/w	PB13_AF_SEL, PB13 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
19:16	r/w	PB12_AF_SEL, PB12 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
15:12	r/w	PB11_AF_SEL, PB11 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
11:8	r/w	PB10_AF_SEL, PB10 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
7:4	r/w	PB9_AF_SEL, PB9 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
3:0	r/w	PB8_AF_SEL, PB8 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0

7.1.49 功能复用选择寄存器 4 (AF_SEL_REG4)

偏移地址: 0x03F0

表 73 功能复用选择寄存器 4

位	访问	描述
31:28	r/w	PC7_AF_SEL, PC7 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
27:24	r/w	PC6_AF_SEL, PC6 功能复用选择 0~7: 保留 其他: 保留 复位值: 0x0
23:20	r/w	PC5_AF_SEL, PC5 功能复用选择 0~7: 保留 其他: 保留 复位值: 0x0
19:16	r/w	PC4_AF_SEL, PC4 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
15:12	r/w	PC3_AF_SEL, PC3 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
11:8	r/w	PC2_AF_SEL, PC2 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
7:4	r/w	PC1_AF_SEL, PC1 功能复用选择 0~7: 保留 其他: 保留 复位值: 0x0
3:0	r/w	PC0_AF_SEL, PC0 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0

7.1.50 功能复用选择寄存器 5 (AF_SEL_REG5)

偏移地址: 0x03F4

表 74 功能复用选择寄存器 5

位	访问	描述
31:28	r/w	PC15_AF_SEL, PC15 功能复用选择 0~7: 保留 其他: 保留 复位值: 0x0
27:24	r/w	PC14_AF_SEL, PC14 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
23:20	r/w	PC13_AF_SEL, PC13 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
19:16	r/w	PC12_AF_SEL, PC12 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
15:12	r/w	PC11_AF_SEL, PC11 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
11:8	r/w	PC10_AF_SEL, PC10 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
7:4	r/w	PC9_AF_SEL, PC9 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
3:0	r/w	PC8_AF_SEL, PC8 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0

7.1.51 功能复用选择寄存器 6 (AF_SEL_REG6)

偏移地址: 0x03F8

表 75 功能复用选择寄存器 6

位	访问	描述
31:28	r/w	PD7_AF_SEL, PD7 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
27:24	r/w	PD6_AF_SEL, PD6 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
23:20	r/w	PD5_AF_SEL, PD5 功能复用选择 0~7: 保留 其他: 保留 复位值: 0x0
19:16	r/w	PD4_AF_SEL, PD4 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
15:12	r/w	PD3_AF_SEL, PD3 功能复用选择 0~7: 保留 其他: 保留 复位值: 0x0
11:8	r/w	PD2_AF_SEL, PD2 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
7:4	r/w	PD1_AF_SEL, PD1 功能复用选择 0~7: 保留 其他: 保留 复位值: 0x0
3:0	r/w	PD0_AF_SEL, PD0 功能复用选择 0~7: 保留 其他: 保留 复位值: 0x0

7.1.52 功能复用选择寄存器 7 (AF_SEL_REG7)

偏移地址: 0x03FC

表 76 功能复用选择寄存器 7

位	访问	描述
31:28	Res	Reserved 复位值: 0x0
23:20	r/w	PD13_AF_SEL, PD13 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
19:16	r/w	PD12_AF_SEL, PD12 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
15:12	r/w	PD11_AF_SEL, PD11 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
11:8	r/w	PD10_AF_SEL, PD10 功能复用选择 0~7: 相关定义详见端口功能复用表格 其他: 保留 复位值: 0x0
7:4	r/w	PD9_AF_SEL, PD9 功能复用选择 0~7: 保留 其他: 保留 复位值: 0x0
3:0	r/w	PD8_AF_SEL, PD8 功能复用选择 0~7: 保留 其他: 保留 复位值: 0x0

7.1.53 端口开漏输出控制寄存器 0 (IO_OD_CTRL_REG0)

偏移地址: 0x0460

表 77 端口开漏输出控制寄存器 0

位	访问	描述
31	r/w	PB15_OD, PB15 开漏输出使能控制 参照 PA0_OD
30	r/w	PB14_OD, PB14 开漏输出使能控制 参照 PA0_OD
29	r/w	PB13_OD, PB13 开漏输出使能控制 参照 PA0_OD
28	r/w	PB12_OD, PB12 开漏输出使能控制 参照 PA0_OD
27	r/w	PB11_OD, PB11 开漏输出使能控制 参照 PA0_OD
26	r/w	PB10_OD, PB10 开漏输出使能控制 参照 PA0_OD
25	r/w	PB9_OD, PB9 开漏输出使能控制 参照 PA0_OD
24	r/w	PB8_OD, PB8 开漏输出使能控制 参照 PA0_OD
23	r/w	PB7_OD, PB7 开漏输出使能控制 参照 PA0_OD
22	r/w	PB6_OD, PB6 开漏输出使能控制 参照 PA0_OD
21	r/w	PB5_OD, PB5 开漏输出使能控制 参照 PA0_OD
20	r/w	PB4_OD, PB4 开漏输出使能控制 参照 PA0_OD
19	r/w	PB3_OD, PB3 开漏输出使能控制 参照 PA0_OD
18	r/w	PB2_OD, PB2 开漏输出使能控制 参照 PA0_OD
17	r/w	PB1_OD, PB1 开漏输出使能控制 参照 PA0_OD
16	r/w	PB0_OD, PB0 开漏输出使能控制 参照 PA0_OD
15	r/w	PA15_OD, PA15 开漏输出使能控制 参照 PA0_OD
14	r/w	PA14_OD, PA14 开漏输出使能控制 参照 PA0_OD

位	访问	描述
13	r/w	PA13_OD, PA13 开漏输出使能控制 参照 PA0_OD
12	r/w	PA12_OD, PA12 开漏输出使能控制 参照 PA0_OD
11	r/w	PA11_OD, PA11 开漏输出使能控制 参照 PA0_OD
10	r/w	PA10_OD, PA10 开漏输出使能控制 参照 PA0_OD
9	r/w	PA9_OD, PA9 开漏输出使能控制 参照 PA0_OD
8	r/w	PA8_OD, PA8 开漏输出使能控制 参照 PA0_OD
7	r/w	PA7_OD, PA7 开漏输出使能控制 参照 PA0_OD
6	r/w	PA6_OD, PA6 开漏输出使能控制 参照 PA0_OD
5	r/w	PA5_OD, PA5 开漏输出使能控制 参照 PA0_OD
4	r/w	PA4_OD, PA4 开漏输出使能控制 参照 PA0_OD
3	r/w	PA3_OD, PA3 开漏输出使能控制 参照 PA0_OD
2	r/w	PA2_OD, PA2 开漏输出使能控制 参照 PA0_OD
1	r/w	PA1_OD, PA1 开漏输出使能控制 参照 PA0_OD
0	r/w	PA0_OD, PA0 开漏输出使能控制 0: 关闭 1: 开启 复位值: 0x0

7.1.54 端口开漏输出控制寄存器 1 (IO_OD_CTRL_REG1)

偏移地址: 0x0464

表 78 端口开漏输出控制寄存器 1

位	访问	描述
31:30	Res	Reserved 复位值: 0x0
29	r/w	PD13_OD, PD13 开漏输出使能控制 参照 PC0_OD
28	r/w	PD12_OD, PD12 开漏输出使能控制 参照 PC0_OD
27	r/w	PD11_OD, PD11 开漏输出使能控制 参照 PC0_OD
26	r/w	PD10_OD, PD10 开漏输出使能控制 参照 PC0_OD
25	r/w	PD9_OD, PD9 开漏输出使能控制 参照 PC0_OD
24	r/w	PD8_OD, PD8 开漏输出使能控制 参照 PC0_OD
23	r/w	PD7_OD, PD7 开漏输出使能控制 参照 PC0_OD
22	r/w	PD6_OD, PD6 开漏输出使能控制 参照 PC0_OD
21	r/w	PD5_OD, PD5 开漏输出使能控制 参照 PC0_OD
20	r/w	PD4_OD, PD4 开漏输出使能控制 参照 PC0_OD
19	r/w	PD3_OD, PD3 开漏输出使能控制 参照 PC0_OD
18	r/w	PD2_OD, PD2 开漏输出使能控制 参照 PC0_OD
17	r/w	PD1_OD, PD1 开漏输出使能控制 参照 PC0_OD
16	r/w	PD0_OD, PD0 开漏输出使能控制 参照 PC0_OD
15	r/w	PC15_OD, PC15 开漏输出使能控制 参照 PC0_OD
14	r/w	PC14_OD, PC14 开漏输出使能控制 参照 PC0_OD
13	r/w	PC13_OD, PC13 开漏输出使能控制 参照 PC0_OD

位	访问	描述
12	r/w	PC12_OD, PC12 开漏输出使能控制 参照 PC0_OD
11	r/w	PC11_OD, PC11 开漏输出使能控制 参照 PC0_OD
10	r/w	PC10_OD, PC10 开漏输出使能控制 参照 PC0_OD
9	r/w	PC9_OD, PC9 开漏输出使能控制 参照 PC0_OD
8	r/w	PC8_OD, PC8 开漏输出使能控制 参照 PC0_OD
7	r/w	PC7_OD, PC7 开漏输出使能控制 参照 PC0_OD
6	r/w	PC6_OD, PC6 开漏输出使能控制 参照 PC0_OD
5	r/w	PC5_OD, PC5 开漏输出使能控制 参照 PC0_OD
4	r/w	PC4_OD, PC4 开漏输出使能控制 参照 PC0_OD
3	r/w	PC3_OD, PC3 开漏输出使能控制 参照 PC0_OD
2	r/w	PC2_OD, PC2 开漏输出使能控制 参照 PC0_OD
1	r/w	PC1_OD, PC1 开漏输出使能控制 参照 PC0_OD
0	r/w	PC0_OD, PC0 开漏输出使能控制 0: 关闭 1: 开启 复位值: 0x0

7.1.55 调试控制寄存器 (DBG_CTRL_REG)

偏移地址: 0x04C0

表 79 调试控制寄存器

位	访问	描述
31:2	Res	Reserved 复位值: 0x0
1	r/w	DBG_TIMH_BRK, 调试模式下高精度定时器刹车使能 0: 关闭 1: 开启 注: 在调试模式下, 即外部仿真器成功连接 CPU, 此时该控制位为 1 的情况下 TIMH 进入刹车状态, 避免调试过程中 TIMH 固定输出导致外设异常。该位只能上电复位。 复位值: 0x1
0	r/w	DBG_WDOG_STOP, 调试模式下看门狗停止工作使能控制 0: 关闭 1: 开启 注: 在调试模式下, 即外部仿真器成功连接 CPU, 此时该控制位为 1 的情况下看门狗会停止计数, 避免调试过程中发生看门狗复位从而仿真器调试中断。该位只能上电复位。 复位值: 0x1

7.1.56 SRAM 状态寄存器 (SRAM_ST)

偏移地址: 0x0500

表 80 SRAM 状态寄存器

位	访问	描述
31:2	Res	Reserved 复位值: 0x0
1	r/w	SRAM_PARITY_ERR_EN: SRAM 校验错误事件至 NMI 使能控制 0: 关闭 1: 开启 复位值: 0x0
0	r/w	SRAM_PARITY_ERR_INT, SRAM 校验错误中断 0: 无校验错误发生 1: 有校验错误发生 注: 该位写 1 清零 复位值: 0x0

7.1.57 SRAM 调试信息寄存器 (SRAM_DBG_INFO)

偏移地址: 0x0504

表 81 SRAM 调试信息寄存器

位	访问	描述
31:10	Res	Reserved 复位值: 0x800
9:0	r/w	SRAM_PARITY_ERR_ADDR, 记录发生校验错误的 SRAM 地址 复位值: 0x0

8 通用 I/O (GPIO)

8.1 概述

通用输入输出 (GPIO)模块可以通过 AHB_Lite 总线与 CPU 通信。GPIO 寄存器支持 8/16/32 位访问。

当引脚配置为 GPIO 功能时，GPIO 数据方向和输出数据寄存器控制每个引脚的方向和输出数据。在引脚相应端口已启用时，GPIO 输入数据寄存器可以显示在引脚配置为数字功能时的每个引脚上的逻辑值。

8.2 主要功能

KPM32R14TX GPIO 主要功能如下：

- GPIO 输出实现置 1、清 0、翻转
- 通过控制寄存器中置位和清零寄存器实现安全操作
- 输入采样通过使用 2 级触发器避免发生亚稳态
- 引脚上拉、下拉
- 推挽输出
- 支持事件触发实现置位、清零、翻转

8.3 模块框图

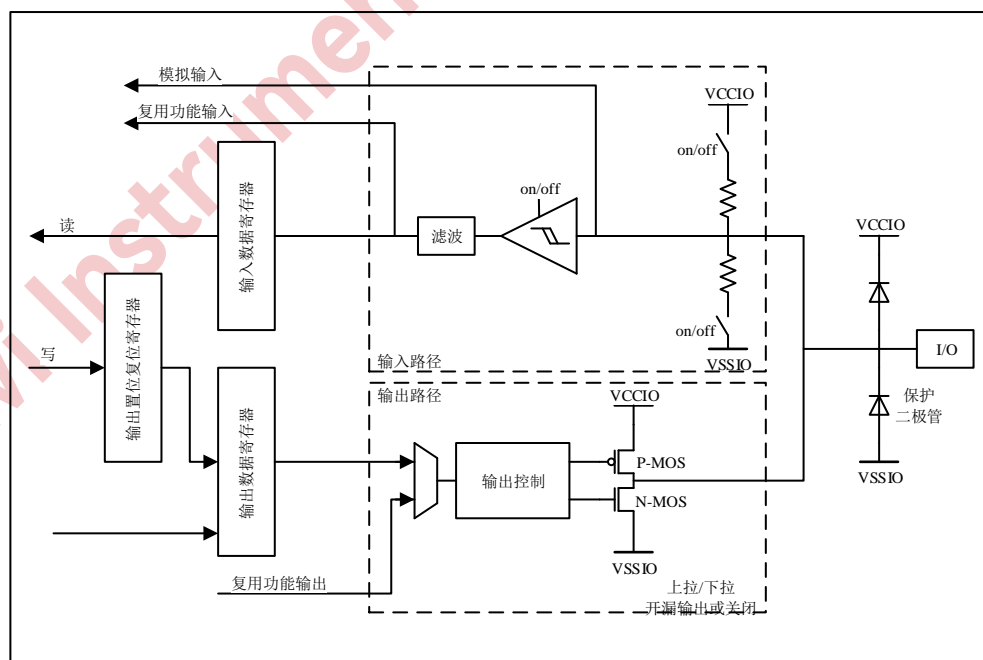


图 12 IO 结构框图

8.4 功能描述

每组 I/O 端口可以通过输出使能寄存器和输出使能清除寄存器 (GPIOx_DOUT_EN, GPIOx_DOUT_EN_CLR) 控制 I/O 信号传输方向, 2 个数据寄存器 (GPIOx_DATA_OUT, GPIOx_DATA_IN), 2 个输出控制寄存器 (GPIOx_DATA_OUT_SET_CLR, GPIOx_TOG) 和 1 个事件控制寄存器 (GPIOx_EVT_CFG)。

8.4.1 GPIO 地址分布

每组 GPIO 对应地址分布参考下方表格。

表 82 GPIO 地址分布

System 32-bit Address Range	Size	Destination Slave
0x5000_0000 - 0x5000_0FFC	4KB	AHB peripherals (GPIO A)
0x5000_1000 - 0x5000_1FFC	4KB	AHB peripherals (GPIO B)
0x5000_2000 - 0x5000_2FFC	4KB	AHB peripherals (GPIO C)
0x5000_3000 - 0x5000_3FFC	4KB	AHB peripherals (GPIO D)

8.4.2 GPIO 输出控制

通过输出使能寄存器 (DOUT_EN/DOUT_EN_CLR) 可以控制当前 IO 作为输入还是输出, 当输出使能关闭时, IO 处于输入功能, 通过输入数据寄存器可以查看 IO 状态; 当输出使能打开时, IO 处于输出功能, 通过输出数据寄存器可以更改 IO 状态。

以下 4 种方式是通过配置对应寄存器更改 IO 输出状态:

- 更改数据输出寄存器 (DATA_OUT)
- 通过数据输出置位寄存器将对应 IO 输出置 1 (DATA_OUT_SET)
- 通过数据输出清零寄存器将对应 IO 输出清 0 (DATA_OUT_CLR)
- 通过数据输出翻转寄存器将对应 IO 输出翻转 (DOUT_TOG)

8.4.3 响应 EBUS 事件

每个 GPIO 都可通过外设事件系统来对其进行控制；

当接收到 EBUS 发送过来的事件时，GPIO 可以进行以下操作：

- 置位：被选择的 IO 输出 1
- 清零：被选择的 IO 输出 0
- 翻转：被选择的 IO 输出翻转

通过事件动作寄存器 (EVT_ACT_SEL) 可选择当检测到事件时进行的动作，当检测到事件后，将会在下一个时钟周期发生相应动作。

8.5 寄存器

本章节对 GPIOx 的寄存器分布和内容进行详细描述。GPIOA，GPIOB，GPIOC，GPIOD 对应的基地址分别是 0x50000000，0x50001000，0x50002000，0x50003000。

GPIOD 有 14 个有效 IO，分别是 bit0 到 bit13。

表 83 GPIOx 寄存器分布

偏移地址	寄存器名称	位宽	访问	复位值
0x0010	GPIO 输入数据寄存器	32	r/w	0x00000000
0x0014	GPIO 输出数据寄存器	32	r/w	0x00000000
0x0018	GPIO 输出置位清零寄存器	32	r/w	0x00000000
0x001C	GPIO 输出翻转寄存器	32	r/w	0x00000000
0x0020	GPIO 输出使能寄存器	32	r/w	0x00000000
0x0024	GPIO 输出使能清除寄存器	32	r/w	0x00000000
0x003C	GPIO 事件控制寄存器	32	r/w	0x00000000

8.5.1 GPIOx 输入数据寄存器 (GPIOx_DATA_IN)

偏移地址: 0x0010

表 84 GPIO 输入数据寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	DATA_IN[15:0]: GPIO 输入数据 写: 将数据写入输出数据寄存器 读: IO 上采样值, 采用两级触发器同步采样, 存在 2 个周期的延迟 复位值: 0x0

8.5.2 GPIOx 输出数据寄存器 (GPIOx_DATA_OUT)

偏移地址: 0x0014

表 85 GPIO 输出数据寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	DATA_OUT[15:0]: GPIO 输出数据 写: 将数据写入输出数据寄存器 读: 当前输出寄存器数据 复位值: 0x0

8.5.3 GPIOx 输出置位清零寄存器 (GPIOx_DOUT_SET_CLR)

偏移地址: 0x0018

表 86 GPIO 输出置位清零寄存器

位	访问	描述
31:16	wo	DATA_OUT_SET [15:0]: GPIO 输出置位 写: 0: 无效操作 1: 将对应 IO 置位 读: 只写寄存器, 读回值为 0 复位值: 0x0
15:0	wo	DATA_OUT_CLR[15:0]: GPIO 输出复位 写: 0: 无效操作 1: 将对应 IO 复位 读: 只写寄存器, 读回值为 0 复位值: 0x0

8.5.4 GPIOx 输出翻转寄存器 (GPIOx_TOG)

偏移地址: 0x001C

表 87 GPIO 输出翻转寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	wo	DOUT_TOG[15:0]: GPIO 输出翻转 写: 0: 无效操作 1: 对应输出寄存器翻转 读: 只写寄存器, 读回值为 0 复位值: 0x0

8.5.5 GPIOx 输出使能寄存器 (GPIOx_DOUT_EN)

偏移地址: 0x0020

表 88 GPIO 输出使能寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	DOUT_EN[15:0]: GPIO 输出使能 写: 0: 无效操作 1: 打开输出使能 读: 0: 当前 IO 为输出使能关闭状态 1: 当前 IO 为输出使能状态 复位值: 0x0

8.5.6 GPIOx 输出使能清除寄存器 (GPIOx_DOUT_EN_CLR)

偏移地址: 0x0024

表 89 GPIO 输出使能清除寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	DOUT_EN_CLR[15:0]: GPIO 输出使能清除 写: 0: 无效操作 1: 清除输出使能 读: 0: 当前 IO 为输出使能关闭状态 1: 当前 IO 为输出使能状态 复位值: 0x0

8.5.7 GPIOx 事件控制寄存器 (GPIOx_EVT_CFG)

偏移地址: 0x003C

表 90 GPIO 事件控制寄存器

位	访问	描述
31:28	r/w	IO_SEL: 响应 IO 选择 配置 0~15 分别对应本组 IO 复位值: 0x0
27:19	Res	Reserved 复位值: 0x0
18:16	r/w	EVT_CHANL_SEL: EBUS 通道选择 000: 选择 EBUS 通道 0 001: 选择 EBUS 通道 1 010: 选择 EBUS 通道 2 011: 选择 EBUS 通道 3 100: 选择 EBUS 通道 4 101: 选择 EBUS 通道 5 110: 选择 EBUS 通道 6 111: 选择 EBUS 通道 7 复位值: 0x0
15:9	Res	Reserved 复位值: 0x0

位	访问	描述
8	r/w	CONFLICT_ST: (写 1 清 0) 冲突错误状态 如果 CPU 和 EBUS 同时作用于同一个 IO, 那么冲突状态位将会被置起, 写 1 清 0: 0: 无冲突发生 1: 冲突发生 复位值: 0x0
7:6	Res	Reserved 复位值: 0x0
5:4	r/w	EVT_ACT_SEL: 事件响应选择 00: 清零 01: 置位 10: 翻转 11: Reserved 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	EVT_ACT_EN: 事件触发使能 0: 关闭事件触发使能 1: 打开事件触发使能 复位值: 0x0

Kiwi Instruments Corp. Confidential

9 直接存储器访问 (DMA)

9.1 概述

DMA 模块用来提供在外设和存储器之间或者存储器和存储器之间的高速数据传输;传输过程中无须 CPU 参与;故 CPU 可同步进行其它操作。DMA 具有 8 条独立的 DMA 通道,每个通道专门用来管理来自于外设或存储器访问的请求。还有一个仲裁器来协调各个 DMA 请求的优先权。

9.2 主要功能

- 支持 8 个独立 DMA 通道,支持优先级配置
- 支持 32 位 AHB-lite 接口
- 支持多种传输类型
 - 存储器到存储器
 - 存储器到外设
 - 外设到存储器
- 支持 3 种数据位宽: 8-bit, 16-bit, 32-bit
- 传输数据长度可配置范围 1~65536

9.3 模块框图

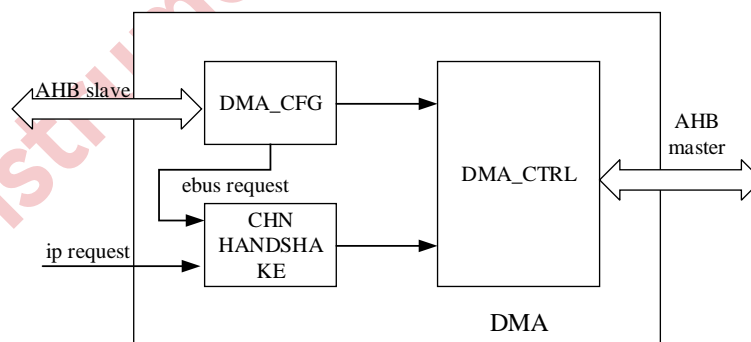


图 13 DMA 模块框图

9.4 功能描述

9.4.1 DMA 通道优先级

DMA 支持 4 个独立 DMA 通道 (简称 CH)，通道的仲裁机制取决于通道号及通道优先级，每个通道的优先级可以通过寄存器 CHN_PRIORITY_SET 设置为默认和高两种级别。CH0 具有最高优先级，随着通道号增加，通道优先级递减。

表 91 DMA 通道优先级

通道	优先级设置	优先级递减
CH0	高	最高优先级
CH1	高	-
CH2	高	-
CH3	高	-
CH4	高	
CH5	高	
CH6	高	
CH7	高	
CH0	默认	-
CH1	默认	-
CH2	默认	-
CH3	默认	-
CH4	默认	
CH5	默认	
CH6	默认	
CH7	默认	最低优先级

9.4.2 DMA 仲裁速率

DMA 的仲裁速率由寄存器 R_INTV 决定，表示为多少个 AHB 总线传输后进行通道仲裁。

表 92 DMA 仲裁间隔时间

R_INTV	R DMA 传输后仲裁
0	R=1
1	R=2
2	R=4
3	R=8

不要把较大的 R_INTV 赋值给低优先级通道，因为它会导致 DMA 无法及时响应高优先级通道。

9.4.3 DMA 传输类型

DMA 传输由寄存器 CYC_CTRL 控制，功能如下表所示。

表 93 DMA 传输类型

CYC_CTRL	功能描述
0	无效数据
1	基本 DMA 传输
2	自动请求
3	保留

无效数据

当 DMA 传输完成，DMA 控制器会将 CYC_CTRL 设置为无效数据，以此来阻止重复传输。

基本 DMA 传输

在此模式下，当使能 DMA 通道并且接收到 DMA 请求后，DMA 传输会按照如下方式进行：

1. DMA 进行 R 个数据传输，当 DMA 的剩余数据为 0 时进入第 3 步，否则进入第 2 步；
2. DMA 进行仲裁：如果有更高优先级通道请求时，响应它的请求，如果没有，当外设或者软件请求发出时，则继续进行第 1 步操作；
3. DMA 控制器置位 DMA_DONE (结束标志)，表示 DMA 传输结束。

自动请求

在此模式下，DMA 控制器只需接收一次请求即可自动完成全部的 DMA 传输。它可以完成大数据量传输，并且不会显著增加高优先级请求的响应时间。当使能 DMA 通道并且接收到请求，DMA 传输方式如下：

1. DMA 进行 R 个数据传输，当 DMA 的剩余数据为 0 时进入第 3 步，否则进入第 2 步；
2. DMA 进行仲裁：如果有更高优先级通道请求时，响应它的请求，如果没有，则继续进行第 1 步操作；
3. DMA 控制器置位 DMA_DONE (结束标志)，表示 DMA 传输结束。

9.4.4 DMA 通道控制

DMA 通道由寄存器 CHX_CTRL/CHX_SRC_EPIT/ CHX_DES_EPIT 进行配置，X 表示通道号；

CHX_CTRL

此寄存器包含 CYC_CTRL /R_INV/DATA_SIZE/DATA_LEN 信号，其中 CYC_CTRL 用来配置 DMA 传输类型，参见章节 9.4.3；R_INTV 用来配置仲裁间隔时间，参见章节 9.4.2。

DATA_SIZE 用来配置数据位宽，0: Byte, 1: Halfword, 2: Word, 3:保持；DMA_LEN 用来配置传输的数据长度。

CHX_SRC_EPIT

此寄存器用来配置 DMA 源数据的结束地址指针。

CHX_DES_EPIT

此寄存器用来配置 DMA 目标数据的结束地址指针。

地址计算

源地址和目标地址的计算由 DATA_SIZE\DATA_LEN\CHX_SRC_EPIT\CHX_DES_EPIT 得出，如下所示：

DATA_SIZE=0:

源地址= (CHX_SRC_EPIT - DATA_LEN)

目标地址= (CHX_DES_EPIT - DATA_LEN)

DATA_SIZE=1:

源地址= (CHX_SRC_EPIT - (DATA_LEN<<1))

目标地址= (CHX_DES_EPIT - (DATA_LEN<<1))

DATA_SIZE=2:

源地址= (CHX_SRC_EPIT - (DATA_LEN<<2))

目标地址= (CHX_DES_EPIT - (DATA_LEN<<2))

DATA_SIZE=3:

源地址= CHX_SRC_EPIT

目标地址= CHX_DES_EPIT

以上地址计算为源地址和目标地址均可以改变的情况。

9.4.5 DMA 通道选择

DMA 的每个独立通道可以响应 16 种设备请求，如下表所示：

表 94 DMA 通道映射

通道	设备请求
通道 0	Dev0: SCI0_DMA_TX
	Dev1: SCI1_DMA_TX
	Dev2: SCI2_DMA_TX
	Dev3:
	Dev4: IIC0_DMA_TX
	Dev5: IIC1_DMA_TX
	Dev6: SPI0_DMA_TX
	Dev7: DNU_CH0
	Dev8: ADC0_DMA_RX
	Dev9: TIMA0_UD
	Dev10:
	Dev11:
	Dev12: TIMG0_TD
	Dev13: TIMG1_CC3D
	Dev14: TIMG2_CC2D
	Dev15:
通道 1	Dev0: SCI0_DMA_RX
	Dev1: SCI1_DMA_RX

通道	设备请求
	Dev2: SCI2_DMA_RX
	Dev3:
	Dev4: IIC0_DMA_RX
	Dev5: IIC1_DMA_RX
	Dev6: SPI0_DMA_RX
	Dev7: DNU_CH1
	Dev8: ADC1_DMA_RX
	Dev9: TIMA0_CC0D
	Dev10: TIMA1_UD
	Dev11:
	Dev12:
	Dev13: TIMG1_TD
	Dev14: TIMG2_CC3D
	Dev15:
通道 2	Dev0: SCI0_DMA_TX
	Dev1: SCI1_DMA_TX
	Dev2: SCI2_DMA_TX
	Dev3:
	Dev4: IIC0_DMA_TX
	Dev5: IIC1_DMA_TX
	Dev6: SPI0_DMA_TX
	Dev7: DNU_CH2
	Dev8: ADC0_DMA_RX
	Dev9: TIMA0_CC1D
	Dev10: TIMA1_CC0D
	Dev11:
	Dev12:
	Dev13:
	Dev14: TIMG2_TD
	Dev15:
通道 3	Dev0: SCI0_DMA_RX
	Dev1: SCI1_DMA_RX

通道	设备请求
	Dev2: SCI2_DMA_RX
	Dev3:
	Dev4: IIC0_DMA_RX
	Dev5: IIC1_DMA_RX
	Dev6: SPI0_DMA_RX
	Dev7:
	Dev8: ADC1_DMA_RX
	Dev9: TIMA0_CC2D
	Dev10: TIMA1_CC1D
	Dev11:
	Dev12: TIMG0_UD
	Dev13:
	Dev14:
	Dev15:
通道 4	Dev0: SCI0_DMA_TX
	Dev1: SCI1_DMA_TX
	Dev2: SCI2_DMA_TX
	Dev3:
	Dev4: IIC0_DMA_TX
	Dev5: IIC1_DMA_TX
	Dev6: SPI0_DMA_TX
	Dev7:
	Dev8: ADC0_DMA_RX
	Dev9: TIMA0_CC3D
	Dev10: TIMA1_CC2D
	Dev11:
	Dev12: TIMG0_CC0D
	Dev13: TIMG1_UD
	Dev14:
	Dev15:
通道 5	Dev0: SCI0_DMA_RX
	Dev1: SCI1_DMA_RX

通道	设备请求
	Dev2: SCI2_DMA_RX
	Dev3:
	Dev4: IIC0_DMA_RX
	Dev5: IIC1_DMA_RX
	Dev6: SPI0_DMA_RX
	Dev7:
	Dev8: ADC1_DMA_RX
	Dev9: TIMA0_COMD
	Dev10: TIMA1_CC3D
	Dev11:
	Dev12: TIMG0_CC1D
	Dev13: TIMG1_CC0D
	Dev14: TIMG2_UD
	Dev15:
通道 6	Dev0: SCI0_DMA_TX
	Dev1: SCI1_DMA_TX
	Dev2: SCI2_DMA_TX
	Dev3:
	Dev4: IIC0_DMA_TX
	Dev5: IIC1_DMA_TX
	Dev6: SPI0_DMA_TX
	Dev7:
	Dev8: ADC0_DMA_RX
	Dev9: TIMA0_TD
	Dev10: TIMA1_COMD
	Dev11:
	Dev12: TIMG0_CC2D
	Dev13: TIMG1_CC1D
	Dev14: TIMG2_CC0D
	Dev15:
通道 7	Dev0: SCI0_DMA_RX
	Dev1: SCI1_DMA_RX

通道	设备请求
	Dev2: SCI2_DMA_RX
	Dev3:
	Dev4: IIC0_DMA_RX
	Dev5: IIC1_DMA_RX
	Dev6: SPI0_DMA_RX
	Dev7:
	Dev8: ADC1_DMA_RX
	Dev9:
	Dev10: TIMA1_TD
	Dev11:
	Dev12: TIMG0_CC3D
	Dev13: TIMG1_CC2D
	Dev14: TIMG2_CC1D
	Dev15:

9.4.6 DMA 中断号

表 95 DMA 中断

名称	中断号
DMA_INT	4

9.5 寄存器

DMA 寄存器的基地址为 0x4002_0000, 下表为 DMA 的各控制寄存器描述。

表 96 寄存器列表

偏移地址	寄存器名称	位宽	访问	复位值
0x000	DMA 控制寄存器	32	r/w	0x00000000h
0x004	DMA 地址保持寄存器	32	r/w	0x00000000h
0x008	DMA 通道软件请求寄存器	32	r/w	0x00000000h
0x010	DMA 中断使能寄存器	32	r/w	0x00000000h
0x014	DMA 中断标志寄存器	32	ro	0x00000000h
0x018	DMA 状态寄存器	32	r/w	0x00000000h
0x01C	DMA 错误状态寄存器	32	wo	0x00000000h

偏移地址	寄存器名称	位宽	访问	复位值
0x020	DMA 通道屏蔽请求置位寄存器	32	wo	0x00000000h
0x024	DMA 通道屏蔽请求清零寄存器	32	wo	0x00000000h
0x028	DMA 通道使能置位寄存器	32	wo	0x00000000h
0x02C	DMA 通道使能清零寄存器	32	wo	0x00000000h
0x030	DMA 通道优先级置位寄存器	32	wo	0x00000000h
0x034	DMA 通道优先级清零寄存器	32	wo	0x00000000h
0x040	DMA 通道选择寄存器	32	ro	0x00000000h
0x050	DMA EBUS 控制寄存器	32	r/w	0x00000000h
0x100	DMA 通道 0 源寄存器	32	r/w	0x00000000h
0x104	DMA 通道 0 目标寄存器	32	r/w	0x00000000h
0x108	DMA 通道 0 控制寄存器	32	r/w	0x00000000h
0x110	DMA 通道 1 源地址寄存器	32	r/w	0x00000000h
0x114	DMA 通道 1 目标地址寄存器	32	r/w	0x00000000h
0x118	DMA 通道 1 控制寄存器	32	r/w	0x00000000h
0x120	DMA 通道 2 源地址寄存器	32	r/w	0x00000000h
0x124	DMA 通道 2 目标地址寄存器	32	r/w	0x00000000h
0x128	DMA 通道 2 控制寄存器	32	r/w	0x00000000h
0x130	DMA 通道 3 源地址寄存器	32	r/w	0x00000000h
0x134	DMA 通道 3 目标地址寄存器	32	r/w	0x00000000h
0x138	DMA 通道 3 控制寄存器	32	r/w	0x00000000h
0x140	DMA 通道 4 源寄存器	32	r/w	0x00000000h
0x144	DMA 通道 4 目标寄存器	32	r/w	0x00000000h
0x148	DMA 通道 4 控制寄存器	32	r/w	0x00000000h
0x150	DMA 通道 5 源地址寄存器	32	r/w	0x00000000h
0x154	DMA 通道 5 目标地址寄存器	32	r/w	0x00000000h
0x158	DMA 通道 5 控制寄存器	32	r/w	0x00000000h
0x160	DMA 通道 6 源地址寄存器	32	r/w	0x00000000h
0x164	DMA 通道 6 目标地址寄存器	32	r/w	0x00000000h
0x168	DMA 通道 6 控制寄存器	32	r/w	0x00000000h

偏移地址	寄存器名称	位宽	访问	复位值
0x170	DMA 通道 7 源地址寄存器	32	r/w	0x00000000h
0x174	DMA 通道 7 目标地址寄存器	32	r/w	0x00000000h
0x178	DMA 通道 7 控制寄存器	32	r/w	0x00000000h

9.5.1 DMA 控制寄存器 (DMA_CTRL)

偏移地址: 0x000

表 97 DMA 控制寄存器

位	访问	描述
31:1	Res	Reserved 复位值: 0x0
0	r/w	DMA_EN: DMA 使能 0: 关闭 1: 使能 复位值: 0x0

9.5.2 DMA 地址保持寄存器 (DMA_ADR_KEEP)

偏移地址: 0x004

表 98 DMA 地址保持寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:14	r/w	CH7_ADR_CTRL: 通道地址控制 0: 源地址和目标地址都改变 1: 源地址不变, 目标地址改变 2: 源地址改变, 目标地址不变 3: 保留 复位值: 0x0
13:12	r/w	CH6_ADR_CTRL: 通道地址控制 0: 源地址和目标地址都改变 1: 源地址不变, 目标地址改变 2: 源地址改变, 目标地址不变 3: 保留 复位值: 0x0
11:10	r/w	CH5_ADR_CTRL: 通道地址控制 0: 源地址和目标地址都改变 1: 源地址不变, 目标地址改变 2: 源地址改变, 目标地址不变

位	访问	描述
		3: 保留 复位值: 0x0
9:8	r/w	CH4_ADR_CTRL: 通道地址控制 0: 源地址和目标地址都改变 1: 源地址不变, 目标地址改变 2: 源地址改变, 目标地址不变 3: 保留 复位值: 0x0
7:6	r/w	CH3_ADR_CTRL: 通道地址控制 0: 源地址和目标地址都改变 1: 源地址不变, 目标地址改变 2: 源地址改变, 目标地址不变 3: 保留 复位值: 0x0
5:4	r/w	CH2_ADR_CTRL: 通道地址控制 0: 源地址和目标地址都改变 1: 源地址不变, 目标地址改变 2: 源地址改变, 目标地址不变 3: 保留 复位值: 0x0
3:2	r/w	CH1_ADR_CTRL: 通道地址控制 0: 源地址和目标地址都改变 1: 源地址不变, 目标地址改变 2: 源地址改变, 目标地址不变 3: 保留 复位值: 0x0
1:0	r/w	CH0_ADR_CTRL: 通道地址控制 0: 源地址和目标地址都改变 1: 源地址不变, 目标地址改变 2: 源地址改变, 目标地址不变 3: 保留 复位值: 0x0

9.5.3 DMA 通道软件请求寄存器 (DMA_CH_SW_REQ)

偏移地址: 0x008

表 99 通道软件请求寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	wo	CH7_SW_REQ: 通道软件请求 0: 不发出请求 1: 发出此通道的软件请求 复位值: 0x0 注: 如果此通道未使能, 此位写 1 也无法发出请求; 软件写 1 后硬件会将此位自动清零。
6	wo	CH6_SW_REQ: 通道软件请求 0: 不发出请求 1: 发出此通道的软件请求 复位值: 0x0 注: 如果此通道未使能, 此位写 1 也无法发出请求; 软件写 1 后硬件会将此位自动清零。
5	wo	CH5_SW_REQ: 通道软件请求 0: 不发出请求 1: 发出此通道的软件请求 复位值: 0x0 注: 如果此通道未使能, 此位写 1 也无法发出请求; 软件写 1 后硬件会将此位自动清零。
4	wo	CH4_SW_REQ: 通道软件请求 0: 不发出请求 1: 发出此通道的软件请求 复位值: 0x0 注: 如果此通道未使能, 此位写 1 也无法发出请求; 软件写 1 后硬件会将此位自动清零。
3	wo	CH3_SW_REQ: 通道软件请求 0: 不发出请求 1: 发出此通道的软件请求 复位值: 0x0 注: 如果此通道未使能, 此位写 1 也无法发出请求; 软件写 1 后硬件会将此位自动清零。
2	wo	CH2_SW_REQ: 通道软件请求 0: 不发出请求 1: 发出此通道的软件请求 复位值: 0x0

位	访问	描述
		注：如果此通道未使能，此位写 1 也无法发出请求；软件写 1 后硬件会将此位自动清零。
1	wo	CH1_SW_REQ: 通道软件请求 0: 不发出请求 1: 发出此通道的软件请求 复位值: 0x0 注：如果此通道未使能，此位写 1 也无法发出请求；软件写 1 后硬件会将此位自动清零。
0	wo	CH0_SW_REQ: 通道软件请求 0: 不发出请求 1: 发出此通道的软件请求 复位值: 0x0 注：如果此通道未使能，此位写 1 也无法发出请求；软件写 1 后硬件会将此位自动清零。

9.5.4 DMA 中断使能寄存器 (DMA_INT_EN)

偏移地址: 0x010

表 100 DMA 中断使能寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	r/w	CH7_INT_EN: 通道中断使能 0: 关闭 1: 使能 复位值: 0x0
6	r/w	CH6_INT_EN: 通道中断使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	CH5_INT_EN: 通道中断使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	CH4_INT_EN: 通道中断使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	CH3_INT_EN: 通道中断使能 0: 关闭

位	访问	描述
		1: 使能 复位值: 0x0
2	r/w	CH2_INT_EN: 通道中断使能 0: 关闭 1: 使能 复位值: 0x0
1	r/w	CH1_INT_EN: 通道中断使能 0: 关闭 1: 使能 复位值: 0x0
0	r/w	CH0_INT_EN: 通道中断使能 0: 关闭 1: 使能 复位值: 0x0

9.5.5 DMA 中断标志寄存器 (DMA_INT_FLAG)

偏移地址: 0x014

表 101 DMA 中断标志寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	r/w	CH7_INT_FLAG: 通道传输完成标志 0: 传输未完成 1: 传输完成 复位值: 0x0 注: 此位软件写 1 清零
6	r/w	CH6_INT_FLAG: 通道传输完成标志 0: 传输未完成 1: 传输完成 复位值: 0x0 注: 此位软件写 1 清零
5	r/w	CH5_INT_FLAG: 通道传输完成标志 0: 传输未完成 1: 传输完成 复位值: 0x0 注: 此位软件写 1 清零
4	r/w	CH4_INT_FLAG: 通道传输完成标志 0: 传输未完成 1: 传输完成

位	访问	描述
		复位值: 0x0 注: 此位软件写 1 清零
3	r/w	CH3_INT_FLAG: 通道传输完成标志 0: 传输未完成 1: 传输完成 复位值: 0x0 注: 此位软件写 1 清零
2	r/w	CH2_INT_FLAG: 通道传输完成标志 0: 传输未完成 1: 传输完成 复位值: 0x0 注: 此位软件写 1 清零
1	r/w	CH1_INT_FLAG: 通道传输完成标志 0: 传输未完成 1: 传输完成 复位值: 0x0 注: 此位软件写 1 清零
0	r/w	CH0_INT_FLAG: 通道传输完成标志 0: 传输未完成 1: 传输完成 复位值: 0x0 注: 此位软件写 1 清零

9.5.6 DMA 状态寄存器 (DMA_INT_FLAG)

偏移地址: 0x018

表 102 DMA 状态寄存器

位	访问	描述
31:20	Res	Reserved 复位值: 0x0
19:16	ro	CH_NUM: DMA 通道数量 n: DMA 的通道数量为 n+1
15:8	Res	Reserved 复位值: 0x0
7:4	ro	STATUS: 传输状态 0: 空闲 1: 读通道控制数据 2: 读源数据地址指针 3: 读目标数据地址指针 4: 读源数据 5: 写目标数据

位	访问	描述
		6: 等待 DMA 请求 7: 写通道控制数据 8: 停顿 9: 完成 其他:保留 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	ro	MASTER_EN: DMA 传输使能状态 0: 未使能 1: 已使能 复位值: 0x0

9.5.7 DMA 错误状态寄存器 (DMA_ERR_STATUS)

偏移地址: 0x01C

表 103 DMA 错误状态寄存器

位	访问	描述
31:1	Res	Reserved 复位值: 0x0
0	ro	DMA_ERR_STATUS: DMA 错误状态 0: 未发生错误 1: 已发生错误 复位值: 0x0 注: 当 DMA 传输遇到总线错误时, 此位会置 1; 软件写 1 清 0。

9.5.8 DMA 通道屏蔽请求置位寄存器 (DMA_REQ_MASK_SET)

偏移地址: 0x020

表 104 DMA 通道屏蔽请求置位寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	r/w	CH7_REQ_MASK: 通道屏蔽请求置位 0: 使能通道请求 1: 屏蔽通道请求 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
6	r/w	CH6_REQ_MASK: 通道屏蔽请求置位 0: 使能通道请求

位	访问	描述
		1: 屏蔽通道请求 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
5	r/w	CH5_REQ_MASK: 通道屏蔽请求置位 0: 使能通道请求 1: 屏蔽通道请求 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
4	r/w	CH4_REQ_MASK: 通道屏蔽请求置位 0: 使能通道请求 1: 屏蔽通道请求 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
3	r/w	CH3_REQ_MASK: 通道屏蔽请求置位 0: 使能通道请求 1: 屏蔽通道请求 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
2	r/w	CH2_REQ_MASK: 通道屏蔽请求置位 0: 使能通道请求 1: 屏蔽通道请求 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
1	r/w	CH1_REQ_MASK: 通道屏蔽请求置位 0: 使能通道请求 1: 屏蔽通道请求 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
0	r/w	CH0_REQ_MASK: 通道屏蔽请求置位 0: 使能通道请求 1: 屏蔽通道请求 复位值: 0x0 注: 此位只能写 1, 写 0 无效果

9.5.9 DMA 通道屏蔽请求清零寄存器 (DMA_REQ_MASK_CLR)

偏移地址: 0x024

表 105 DMA 通道屏蔽请求清零寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	wo	CH7_REQ_MASK_CLR: 通道屏蔽请求清零 0: 无效果 1: 使能通道请求 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH3_REQ_MASK 清零, 写 0 无效果。
6	wo	CH6_REQ_MASK_CLR: 通道屏蔽请求清零 0: 无效果 1: 使能通道请求 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH3_REQ_MASK 清零, 写 0 无效果。
5	wo	CH5_REQ_MASK_CLR: 通道屏蔽请求清零 0: 无效果 1: 使能通道请求 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH3_REQ_MASK 清零, 写 0 无效果。
4	wo	CH4_REQ_MASK_CLR: 通道屏蔽请求清零 0: 无效果 1: 使能通道请求 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH3_REQ_MASK 清零, 写 0 无效果。
3	wo	CH3_REQ_MASK_CLR: 通道屏蔽请求清零 0: 无效果 1: 使能通道请求 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH3_REQ_MASK 清零, 写 0 无效果。
2	wo	CH2_REQ_MASK_CLR: 通道屏蔽请求清零 0: 无效果 1: 使能通道请求 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH2_REQ_MASK 清零, 写 0 无效果。
1	wo	CH1_REQ_MASK_CLR: 通道屏蔽请求清零 0: 无效果 1: 使能通道请求 复位值: 0x0

位	访问	描述
		注：此位只能写 1，可以使寄存器 CH1_REQ_MASK 清零，写 0 无效果。
0	wo	CH0_REQ_MASK_CLR: 通道屏蔽请求清零 0: 无效果 1: 使能通道请求 复位值: 0x0 注：此位只能写 1，可以使寄存器 CH0_REQ_MASK 清零，写 0 无效果。

9.5.10 DMA 通道使能置位寄存器 (DMA_CH_EN_SET)

偏移地址: 0x028

表 106 DMA 通道使能置位寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	r/w	CH7_EN: 通道使能 0: 通道关闭 1: 通道使能 复位值: 0x0 注：此位只能写 1，写 0 无效果
6	r/w	CH6_EN: 通道使能 0: 通道关闭 1: 通道使能 复位值: 0x0 注：此位只能写 1，写 0 无效果
5	r/w	CH5_EN: 通道使能 0: 通道关闭 1: 通道使能 复位值: 0x0 注：此位只能写 1，写 0 无效果
4	r/w	CH4_EN: 通道使能 0: 通道关闭 1: 通道使能 复位值: 0x0 注：此位只能写 1，写 0 无效果

位	访问	描述
3	r/w	CH3_EN: 通道使能 0: 通道关闭 1: 通道使能 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
2	r/w	CH2_EN: 通道使能 0: 通道关闭 1: 通道使能 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
1	r/w	CH1_EN: 通道使能 0: 通道关闭 1: 通道使能 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
0	r/w	CH0_EN: 通道使能 0: 通道关闭 1: 通道使能 复位值: 0x0 注: 此位只能写 1, 写 0 无效果

9.5.11 DMA 通道使能清零寄存器 (DMA_CH_EN_CLR)

偏移地址: 0x02C

表 107 DMA 通道使能清零寄存器

位	访问	描述
31:4	Res	Reserved 复位值: 0x0
7	wo	CH7_EN_CLR: 通道使能清零 0: 无效果 1: 关闭通道 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH3_EN 清零, 写 0 无效果; 当此通道 DMA 传输完成, 此位会被硬件清零。
6	wo	CH6_EN_CLR: 通道使能清零 0: 无效果 1: 关闭通道 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH3_EN 清零, 写 0 无效果; 当此通道 DMA 传输完成, 此位会被硬件清零。

位	访问	描述
5	wo	CH5_EN_CLR: 通道使能清零 0: 无效果 1: 关闭通道 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH3_EN 清零, 写 0 无效果; 当此通道 DMA 传输完成, 此位会被硬件清零。
4	wo	CH4_EN_CLR: 通道使能清零 0: 无效果 1: 关闭通道 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH3_EN 清零, 写 0 无效果; 当此通道 DMA 传输完成, 此位会被硬件清零。
3	wo	CH3_EN_CLR: 通道使能清零 0: 无效果 1: 关闭通道 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH3_EN 清零, 写 0 无效果; 当此通道 DMA 传输完成, 此位会被硬件清零。
2	wo	CH2_EN_CLR: 通道使能清零 0: 无效果 1: 关闭通道 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH2_EN 清零, 写 0 无效果; 当此通道 DMA 传输完成, 此位会被硬件清零。
1	wo	CH1_EN_CLR: 通道使能清零 0: 无效果 1: 关闭通道 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH1_EN 清零, 写 0 无效果; 当此通道 DMA 传输完成, 此位会被硬件清零。
0	wo	CH0_EN_CLR: 通道使能清零 0: 无效果 1: 关闭通道 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH0_EN 清零, 写 0 无效果; 当此通道 DMA 传输完成, 此位会被硬件清零。

9.5.12 DMA 通道优先级置位寄存器 (DMA_CH_PRT_SET)

偏移地址: 0x030

表 108 DMA 通道优先级置位寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	r/w	CH7_PRT: 通道优先级 0: 默认优先级 1: 高优先级 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
6	r/w	CH6_PRT: 通道优先级 0: 默认优先级 1: 高优先级 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
5	r/w	CH5_PRT: 通道优先级 0: 默认优先级 1: 高优先级 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
4	r/w	CH4_PRT: 通道优先级 0: 默认优先级 1: 高优先级 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
3	r/w	CH3_PRT: 通道优先级 0: 默认优先级 1: 高优先级 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
2	r/w	CH2_PRT: 通道优先级 0: 默认优先级 1: 高优先级 复位值: 0x0 注: 此位只能写 1, 写 0 无效果
1	r/w	CH1_PRT: 通道优先级 0: 默认优先级 1: 高优先级 复位值: 0x0

位	访问	描述
		注：此位只能写 1，写 0 无效果
0	r/w	CH0_PRT：通道优先级 0：默认优先级 1：高优先级 复位值：0x0 注：此位只能写 1，写 0 无效果

9.5.13 DMA 通道优先级清零寄存器 (DMA_CH_PRT_CLR)

偏移地址：0x034

表 109 DMA 通道优先级清零寄存器

位	访问	描述
31:8	Res	Reserved 复位值：0x0
7	wo	CH7_PRT_CLR：通道优先级清零 0：无效果 1：默认优先级 复位值：0x0 注：此位只能写 1，可以使寄存器 CH3_PRT 清零，写 0 无效果。
6	wo	CH6_PRT_CLR：通道优先级清零 0：无效果 1：默认优先级 复位值：0x0 注：此位只能写 1，可以使寄存器 CH3_PRT 清零，写 0 无效果。
5	wo	CH5_PRT_CLR：通道优先级清零 0：无效果 1：默认优先级 复位值：0x0 注：此位只能写 1，可以使寄存器 CH3_PRT 清零，写 0 无效果。
4	wo	CH4_PRT_CLR：通道优先级清零 0：无效果 1：默认优先级 复位值：0x0 注：此位只能写 1，可以使寄存器 CH3_PRT 清零，写 0 无效果。
3	wo	CH3_PRT_CLR：通道优先级清零 0：无效果 1：默认优先级 复位值：0x0 注：此位只能写 1，可以使寄存器 CH3_PRT 清零，写 0 无效果。
2	wo	CH2_PRT_CLR：通道优先级清零 0：无效果 1：默认优先级

位	访问	描述
		复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH2_PRT 清零, 写 0 无效果。
1	wo	CH1_PRT_CLR: 通道优先级清零 0: 无效果 1: 默认优先级 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH1_PRT 清零, 写 0 无效果。
0	wo	CH0_PRT_CLR: 通道优先级清零 0: 无效果 1: 默认优先级 复位值: 0x0 注: 此位只能写 1, 可以使寄存器 CH0_PRT 清零, 写 0 无效果。

9.5.14 DMA 通道选择寄存器 (DMA_CH_SEL)

偏移地址: 0x040

表 110 DMA 通道选择寄存器

位	访问	描述
31:28	r/w	CH7_SEL: 通道选择 0: 通道选择 dev0 1: 通道选择 dev1 ... 15: 通道选择 dev15 复位值: 0x0
27:24	r/w	CH6_SEL: 通道选择 0: 通道选择 dev0 1: 通道选择 dev1 ... 15: 通道选择 dev15 复位值: 0x0
23:20	r/w	CH5_SEL: 通道选择 0: 通道选择 dev0 1: 通道选择 dev1 ... 15: 通道选择 dev15 复位值: 0x0

位	访问	描述
19:16	r/w	CH4_SEL: 通道选择 0: 通道选择 dev0 1: 通道选择 dev1 ... 15: 通道选择 dev15 复位值: 0x0
15:12	r/w	CH3_SEL: 通道选择 0: 通道选择 dev0 1: 通道选择 dev1 ... 15: 通道选择 dev15 复位值: 0x0
11:8	r/w	CH2_SEL: 通道选择 0: 通道选择 dev0 1: 通道选择 dev1 ... 15: 通道选择 dev15 复位值: 0x0
7:4	r/w	CH1_SEL: 通道选择 0: 通道选择 dev0 1: 通道选择 dev1 ... 15: 通道选择 dev15 复位值: 0x0
3:0	r/w	CH0_SEL: 通道选择 0: 通道选择 dev0 1: 通道选择 dev1 ... 15: 通道选择 dev15 复位值: 0x0

9.5.15 DMA EBUS 控制寄存器 (DMA_EBUS_CTRL)

偏移地址: 0x050

表 111 DMA EBUS 控制寄存器

位	访问	描述
31	r/w	CH7_EBUS_DISABLE: 通道 EBUS 触发使能 0: 使能 EBUS 触发 1: 关闭 EBUS 触发 复位值: 0x1

位	访问	描述
30:28	r/w	CH7_EBUS_CTRL: 通道选择 EBUS 事件 0: 通道选择 EBUS 事件 0 1: 通道选择 EBUS 事件 1 ... 7: 通道选择 EBUS 事件 7 复位值: 0x0
27	r/w	CH6_EBUS_DISABLE: 通道 EBUS 触发使能 0: 使能 EBUS 触发 1: 关闭 EBUS 触发 复位值: 0x1
26:24	r/w	CH6_EBUS_CTRL: 通道选择 EBUS 事件 0: 通道选择 EBUS 事件 0 1: 通道选择 EBUS 事件 1 ... 7: 通道选择 EBUS 事件 7 复位值: 0x0
23	r/w	CH5_EBUS_DISABLE: 通道 EBUS 触发使能 0: 使能 EBUS 触发 1: 关闭 EBUS 触发 复位值: 0x1
22:20	r/w	CH5_EBUS_CTRL: 通道选择 EBUS 事件 0: 通道选择 EBUS 事件 0 1: 通道选择 EBUS 事件 1 ... 7: 通道选择 EBUS 事件 7 复位值: 0x0
19	r/w	CH4_EBUS_DISABLE: 通道 EBUS 触发使能 0: 使能 EBUS 触发 1: 关闭 EBUS 触发 复位值: 0x1
18:16	r/w	CH4_EBUS_CTRL: 通道选择 EBUS 事件 0: 通道选择 EBUS 事件 0 1: 通道选择 EBUS 事件 1 ... 7: 通道选择 EBUS 事件 7 复位值: 0x0
15	r/w	CH3_EBUS_DISABLE: 通道 EBUS 触发使能 0: 使能 EBUS 触发 1: 关闭 EBUS 触发 复位值: 0x1

位	访问	描述
14:12	r/w	CH3_EBUS_CTRL: 通道选择 EBUS 事件 0: 通道选择 EBUS 事件 0 1: 通道选择 EBUS 事件 1 ... 7: 通道选择 EBUS 事件 7 复位值: 0x0
11	r/w	CH2_EBUS_DISABLE: 通道 EBUS 触发使能 0: 使能 EBUS 触发 1: 关闭 EBUS 触发 复位值: 0x1
10:8	r/w	CH2_EBUS_CTRL: 通道选择 EBUS 事件 0: 通道选择 EBUS 事件 0 1: 通道选择 EBUS 事件 1 ... 7: 通道选择 EBUS 事件 7 复位值: 0x0
7	r/w	CH1_EBUS_DISABLE: 通道 EBUS 触发使能 0: 使能 EBUS 触发 1: 关闭 EBUS 触发 复位值: 0x1
6:4	r/w	CH1_EBUS_CTRL: 通道选择 EBUS 事件 0: 通道选择 EBUS 事件 0 1: 通道选择 EBUS 事件 1 ... 7: 通道选择 EBUS 事件 7 复位值: 0x0
3	r/w	CH0_EBUS_DISABLE: 通道 EBUS 触发使能 0: 使能 EBUS 触发 1: 关闭 EBUS 触发 复位值: 0x1
2:0	r/w	CH0_EBUS_CTRL: 通道选择 EBUS 事件 0: 通道选择 EBUS 事件 0 1: 通道选择 EBUS 事件 1 ... 7: 通道选择 EBUS 事件 7 复位值: 0x0

9.5.16 DMA 通道 0 源地址寄存器 (DMA_CH0_SRC)

偏移地址: 0x100

表 112 DMA 通道 0 源地址寄存器

位	访问	描述
31:0	r/w	CH0_SRC_EPIT: 通道源数据的结束地址指针 复位值: 0x0

9.5.17 DMA 通道 0 目标地址寄存器 (DMA_CH0_DES)

偏移地址: 0x104

表 113 DMA 通道 0 目标地址寄存器

位	访问	描述
31:0	r/w	CH0_DES_EPIT: 通道源数据的结束地址指针 复位值: 0x0

9.5.18 DMA 通道 0 控制寄存器 (DMA_CH0_CTRL)

偏移地址: 0x108

表 114 DMA 通道 0 控制寄存器

位	访问	描述
31:25	Res	Reserved 复位值: 0x0
24	r/w	CFG_HOLD_EN: 配置保持使能, 当使能此位时, DMA 传输完成后配置保持为上次配置值 0: 关闭 1: 使能 复位值: 0x0 注: 实际的数据长度为此寄存器值加 1。
23:8	r/w	DATA_LEN: 数据长度 复位值: 0x0 注: 实际的数据长度为此寄存器值加 1。
7:6	r/w	DATA_SIZE: 数据位宽 0: Byte 1: Halfword 2: Word 3: 保留 复位值: 0x0

位	访问	描述
5:4	r/w	R_INTV: 仲裁间隔时间 0: 1 DMA 传输 1: 2 DMA 传输 2: 4 DMA 传输 3: 8 DMA 传输 复位值: 0x0
3:2	Rev	Reserved 复位值: 0x0
1:0	r/w	CYC_CTRL: DMA 传输类型 0: 无效数据 1: 基本传输 2: 自动请求 3: 保留 复位值: 0x0 注: 当 DMA 传输完成, 此位会被硬件清零。

Kiwi Instruments Corp. Confidential

10 循环冗余校验 (CRC)

10.1 概述

CRC 是循环冗余校验模块，它可以根据生成多项式得到 CRC 校验码。

CRC 模块还可以对 Flash 存储的代码进行校验，以此来确保代码的正确性。

10.2 主要功能

- 支持 CRC-32 (以太网) 生成多项式: 0x4C11DB7
- 支持 32 位输入和输出
- 支持输入反转和输出反转
- 支持 0 延迟的 CRC 计算
- CRC 初始值可配置

10.3 模块框图

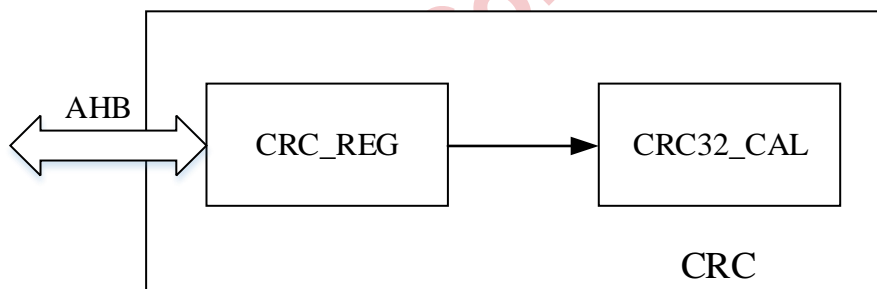


图 14 CRC 模块框图

CRC_REG 模块: 寄存器控制模块, 支持 AHB 总线访问;

CRC32_CAL 模块: 生成多项式 0x4C11DB7 的 CRC32 计算。

10.4 功能描述

CRC 模块支持 CRC-32 (以太网), 生成多项式位 0x4C11DB7, 如下:

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

CRC 时钟为系统 SCLK, 计算时间为 1 个时钟, CRC 模块只有一个数据寄存器, 可以完成初始值赋值、输入数据和输出结果的功能, 输当 CRC 使能之前, 写 CRC 数据寄存器可以进行初始值赋值, 在 CRC 使能之后, 写 CRC 数据寄存器可以输入数据, 读 CRC 寄存器可以得到 CRC 计算结果。

CRC 支持输入数据反转功能, 支持按字节、半字和字反转, 以输入数据 0xAB783412 数据为例,

按字节反转后, 进行 CRC 计算的数据为 0xD51E2C48

按半字节反转后, 进行 CRC 计算的数据为 0x1ED5482C

按字反转后, 进行 CRC 计算的数据为 0x482C1ED5

CRC 支持输出数据按字反转, 反转方法与输入按字反转一样。

10.5 寄存器

CRC 寄存器的基地址为 0x4003_0000, 下表为 CRC 的各控制寄存器描述。

表 115 CRC 寄存器列表

偏移地址	寄存器名称	位宽	访问	复位值
0x000	CRC 控制寄存器	32	r/w	0x00000000h
0x004	CRC 数据寄存器	32	r/w	0x00000000h

10.5.1 CRC 控制寄存器 (CRC_CTRL)

偏移地址: 0x000

表 116 CRC 地址寄存器

位	访问	描述
31:7	Res	Reserved 复位值: 0x0
6	r/w	CRC_OUT_REV: CRC 输出数据反转 0: 不反转 1: 按字 (Word) 反转 复位值: 0x0

位	访问	描述
5:4	r/w	CRC_IN_REV: CRC 输入数据反转 0: 不反转 1: 按字节 (Byte) 反转 2: 按半字 (Halfword) 反转 3: 按字 (Word) 反转 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	CRC_EN: CRC 使能 0: 关闭 1: 使能 复位值: 0x0 注: CRC 使能后, 才可以进行 CRC 计算

10.5.2 CRC 数据寄存器 (CRC_DATA)

偏移地址: 0x004

表 117 CRC 数据寄存器

位	访问	描述
31:0	r/w	CRC_DATA: CRC 数据寄存器 复位值: 0x0 注: 在 CRC 关闭时, 对此寄存器进行写操作的数据为 CRC 计算初始值; 在 CRC 使能时, 对此寄存器进行写操作的数据 (并依据 CRC_IN_REV 进行反转) 为 CRC 当前计算的输入值; 在 CRC 使能后, 对此寄存器进行读操作可以得到 CRC 的计算结果 (并依据 CRC_OUT_REV 进行反转)。

11 三角函数硬件加速器 (TMU)

11.1 概述

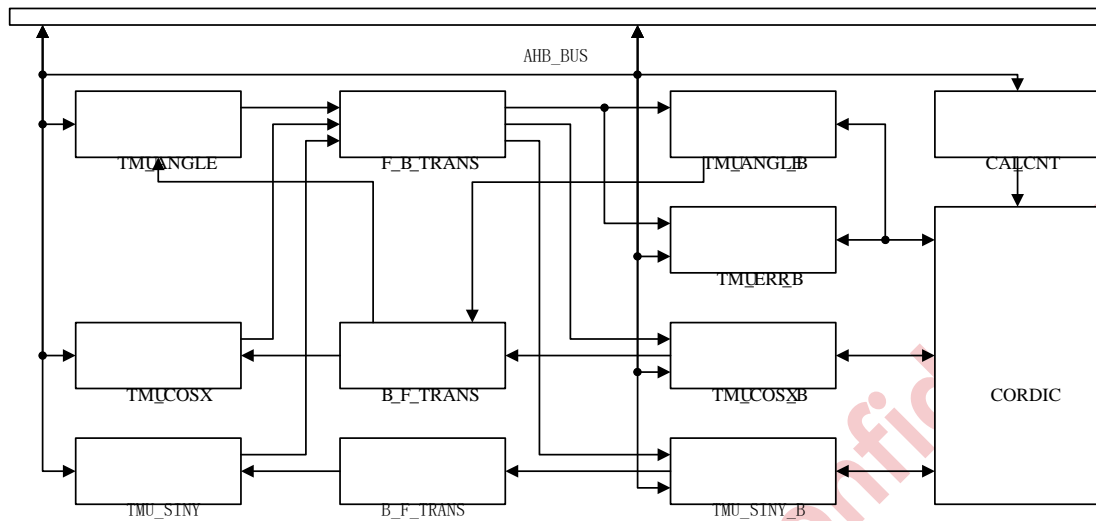
TMU 采用 CORDIC 算法实现正弦、余弦和反正切计算，数据计算类型支持单精度浮点数据及有符号定点数据。

11.2 主要功能

- $a = \sin(\text{angle})$
- $a = \cos(\text{angle})$
- $a = \arctan(x, y)$
- 支持计算模式自动切换
- 数据输入类型支持单精度浮点和有符号定点 (2^{24})
- 数据输出类型支持单精度浮点和有符号定点 (2^{24})
- 数据输出类型由输入类型决定
- 正余弦弧度输入范围 $[-2\pi, 2\pi]$
- 正余弦计算输出范围 $[-1, 1]$
- 反正切 X,Y 输入范围 $(-65536, 65536)$
- 反正切计算输出范围 $(-\pi, \pi)$
- 计算周期
 - $a = \sin(\text{angle})$ (5*cycles TMU_CLK)
 - $a = \cos(\text{angle})$ (5*cycles TMU_CLK)
 - $a = \arctan(x, y)$ (5*cycles TMU_CLK)

11.3 模块框图

图 15 TMU 模块框图



11.4 功能描述

11.4.1 Cordic 算法

Cordic 算法是通过运用基本的加减和移位运算来代替乘法运算，在计算过程中不断逼近目标值。

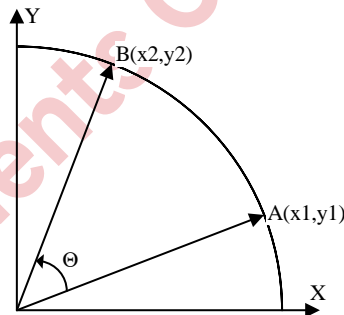


图 16 直角坐标系 A (x1, y1) B (x2, y2)

如上图所示，在直角坐标系中存在点 A (x1, y1)，将点 A 逆时针旋转角度 θ 得到点 B (x2, y2)；则存在以下关系：

$$x2 = x1 \cos \theta - y1 \sin \theta = \cos \theta (x1 - y1 \tan \theta)$$

$$y2 = x1 \sin \theta + y1 \cos \theta = \cos \theta (y1 + x1 \tan \theta)$$

消去 $\cos \theta$ 得到：

$$x2' = x1 - y1 \tan \theta$$

$$y2' = y1 + x1 \tan \theta$$

则存在下图关系：

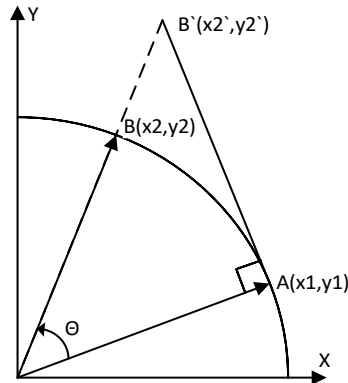


图 17 直角坐标系 B' (x2', y2')

可知向量 B' 的模值是向量 B 的模值的 $1/\cos\theta$ 倍，所以当计算得到 B'，只需对 B' 进行相应的比例变化，就可得到 B 的坐标：

存在一组角度值，存在关系：

$$\tan\theta_i = 2^{-i}$$

则存在：

$$x2' = x1 - y1\tan\theta = x1 - y1 * 2^{-i}$$

$$y2' = y1 + x1\tan\theta = y1 + x1 * 2^{-i}$$

通过这种方法使角度不断逼近计算目标值：

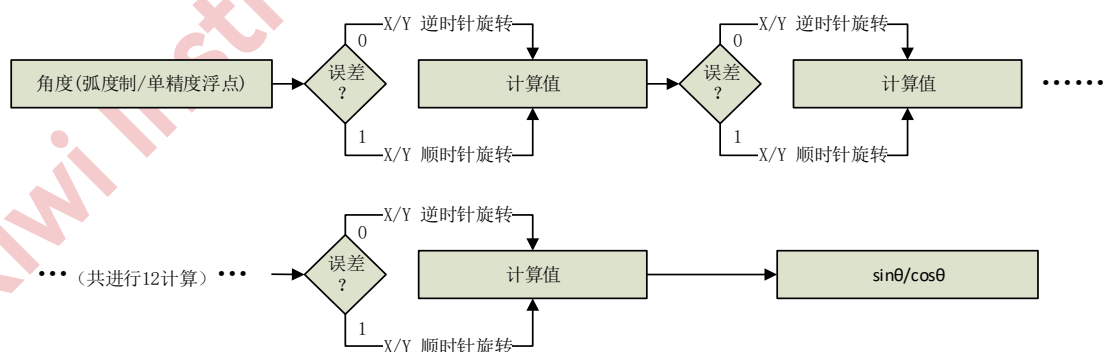


图 18 CORDIC 算法计算流程

将这组角度值整理如下，在经过 12 轮计算后，将结果乘 k(0.6072529591) 即可得到最终准确模值；下表为 k 的推导流程：

表 118 CORDIC 算法计算角度

序号	θ (角度制)	θ (弧度制)	$\tan\theta$	$2^{(-i)}$	$\arctan(2^{(-i)})$	$\cos(\arctan(2^{(-i)}))$
0	45.0000000000	0.7853981634	1.0000000000	1.0000000000	0.7853981634	0.7071067812
1	26.5650511771	0.4636476090	0.5000000000	0.5000000000	0.4636476090	0.8944271910
2	14.0362434679	0.2449786631	0.2500000000	0.2500000000	0.2449786631	0.9701425001
3	7.1250163489	0.1243549945	0.1250000000	0.1250000000	0.1243549945	0.9922778767
4	3.5763343750	0.0624188100	0.0625000000	0.0625000000	0.0624188100	0.9980525785
5	1.7899106082	0.0312398334	0.0312500000	0.0312500000	0.0312398334	0.9995120761
6	0.8951737102	0.0156237286	0.0156250000	0.0156250000	0.0156237286	0.9998779520
7	0.4476141709	0.0078123411	0.0078125000	0.0078125000	0.0078123411	0.9999694838
8	0.2238105004	0.0039062301	0.0039062500	0.0039062500	0.0039062301	0.9999923707
9	0.1119056771	0.0019531225	0.0019531250	0.0019531250	0.0019531225	0.9999980927
10	0.0559528919	0.0009765622	0.0009765625	0.0009765625	0.0009765622	0.9999995232
11	0.0279764526	0.0004882812	0.0004882813	0.0004882813	0.0004882812	0.9999998808
/	/	/	/	/	/	累乘计算结果= 0.6072529591

11.4.2 单精度浮点数据类型

TMU 支持单精度浮点数据计算，单精度浮点数据的格式如下图：单精度浮点数据共有 32bit，最高位是符号位，中间 8bit 为阶码，低 23bit 为尾数。

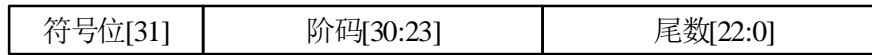


图 19 单精度浮点型数据

11.4.3 TMU 工作时序

当进行正余弦计算时，根据不同的数据类型将需要计算的角度输入至对应的角度寄存器，在 5 个计算周期后，通过访问正余弦寄存器获取计算值：

当数据类型为浮点计算时，将计算角度写入 TMU_ANGLE 寄存器中，通过 TMU_COSX 及 TMU_SINY 获取计算的正余弦值。

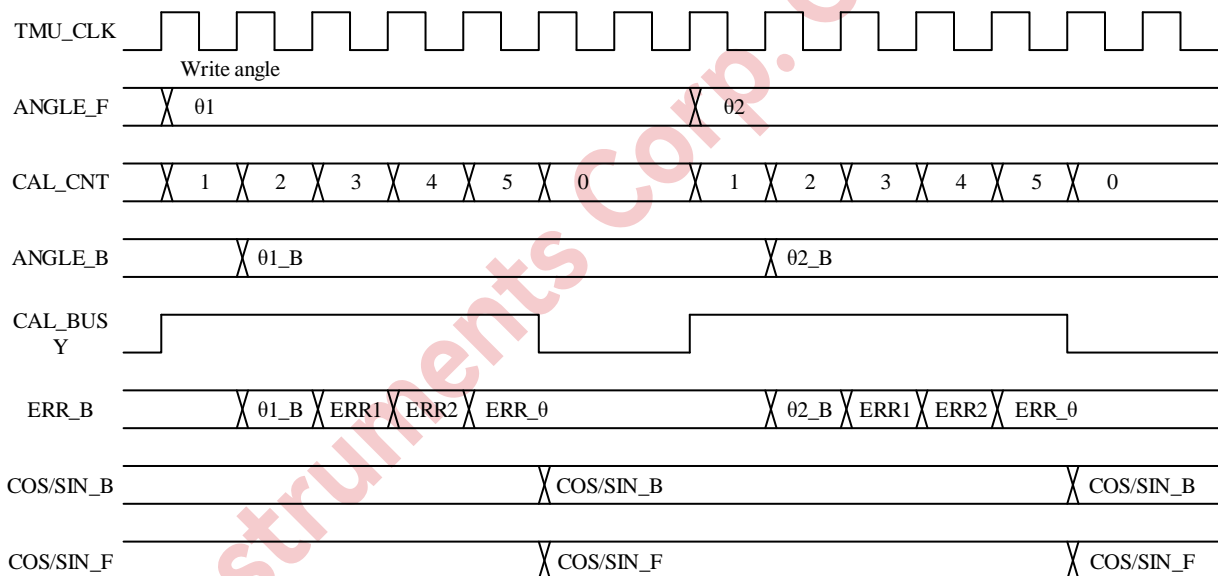


图 20 浮点输入正余弦计算时序

当数据类型为定点计算时，将计算角度写入 TMU_ANGLE_B 寄存器中，通过 TMU_COSX_B 及 TMU_SINY_B 获取计算的正余弦值。

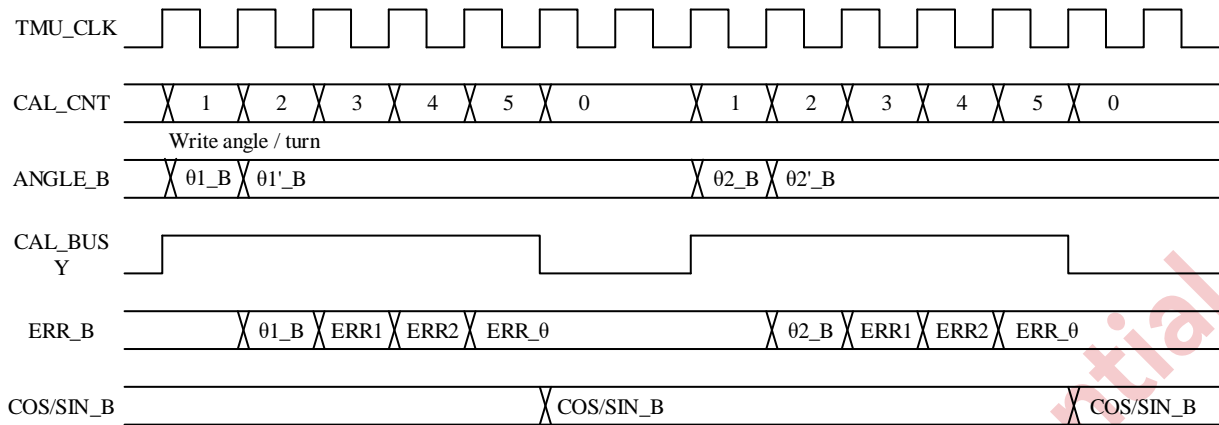


图 21 浮点输入反正切计算时序

当进行反正切计算时，根据不同的数据类型将 X 及 Y 输入至对应余弦及正弦寄存器（先写入 X，后写入 Y），在 5 个计算周期后，通过访问角度寄存器获取计算值；

当数据类型为浮点计算时，将 X 值写入 TMU_COSX 寄存器，将 Y 值写入 TMU_SINY 寄存器，通过 TMU_ANGLE 获取计算的角度值。

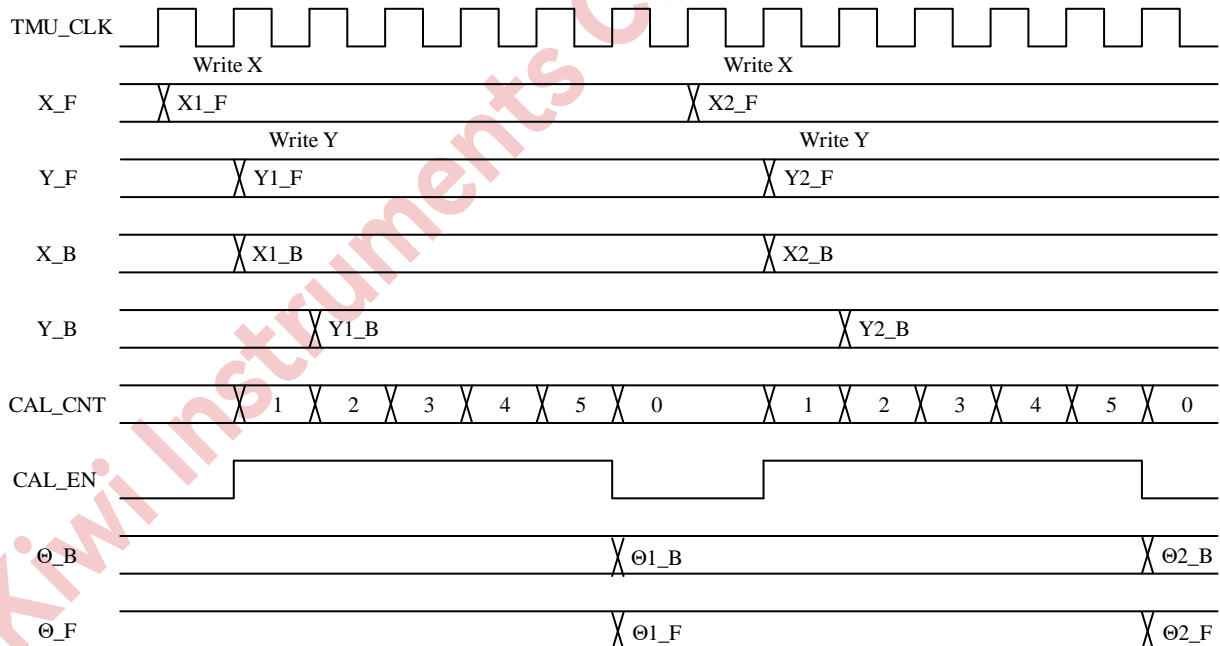


图 22 定点输入正余弦计算时序

当数据类型为定点计算时，将 X 值写入 TMU_COSX_B 寄存器，将 Y 值写入 TMU_SINY_B 寄存器，通过 TMU_ANGLE_B 获取计算的角度值。

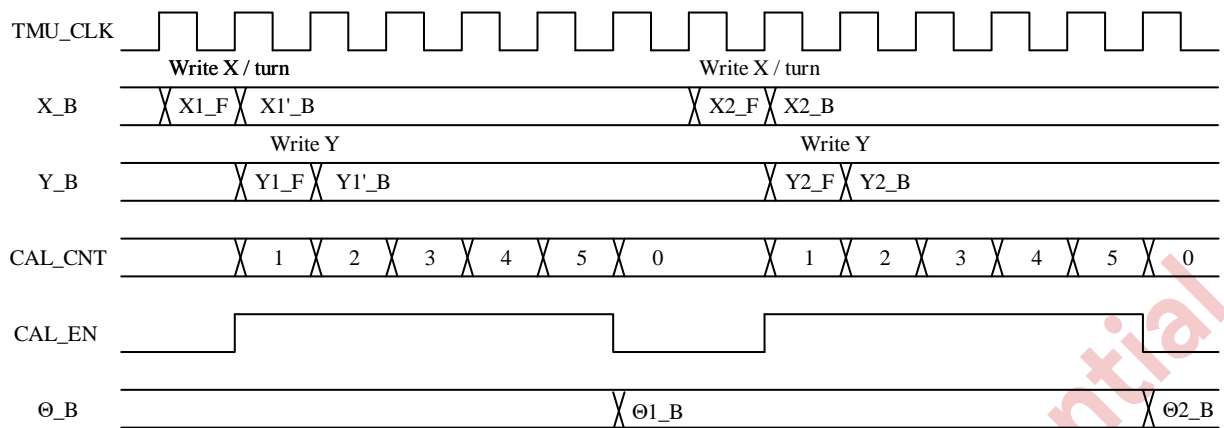


图 23 定点输入反正切计算时序

11.5 寄存器

TMU 寄存器的基地址为 0x4003_1000，下表为 TMU 的各控制寄存器描述。

表 119 寄存器列表

偏移地址	寄存器名称	位宽	访问	复位值
0x00	TMU 控制寄存器	32	r/w	0x00000000
0x04~0x0C	Reserved	32	res	0x00000000
0x10	TMU 角度浮点寄存器	32	r/w	0x00000000
0x14	Reserved	32	res	0x00000000
0x18	TMU 余弦浮点寄存器	32	r/w	0x00000000
0x1C	TMU 正弦浮点寄存器	32	r/w	0x00000000
0x20	TMU 角度定点寄存器	32	r/w	0x00000000
0x24	TMU 误差定点寄存器	32	r/w	0x00000000
0x28	TMU 余弦定点寄存器	32	r/w	0x00000000
0x2C	TMU 正弦定点寄存器	32	r/w	0x00000000
0x30	TMU 状态寄存器	32	r/w	0x00000000

11.5.1 TMU 控制寄存器 (TMU_CTRL)

偏移地址: 0x0000

表 120 TMU 控制寄存器

位	访问	描述
31:1	Res	Reserved 复位值: 0x0
0	r/w	TMU_EN: TMU 使能 0: 打开 TMU 1: 关闭 TMU 复位值: 0x0

11.5.2 TMU 角度浮点寄存器 (TMU_ANGLE)

偏移地址: 0x0010

表 121 TMU 角度浮点寄存器

位	访问	描述
31:0	r/w	TMU_ANGLE_F: 角度值 注: 进行 SIN/COS 运算时, 该寄存器用于单精度浮点型角度输入; 进行 ARCTAN 运算时, 该寄存器表示计算结果; 运算过程禁止操作该寄存器。 复位值: 0x0

11.5.3 TMU 余弦浮点寄存器 (TMU_COSX)

偏移地址: 0x0018

表 122 TMU 余弦浮点寄存器

位	访问	描述
31:0	r/w	TMU_COSX_F: 余弦值 注: 进行 SIN/COS 运算时, 该寄存器表示 COS 计算结果。进行 ARCTAN 运算时, 该寄存器用于 X 值的输入; 运算过程禁止操作该寄存器。 复位值: 0x0

11.5.4 TMU 正弦浮点寄存器 (TMU_SINY)

偏移地址: 0x001C

表 123 TMU 正弦浮点寄存器

位	访问	描述
31:0	r/w	TMU_SINY_F: 正弦值 注: 进行 SIN/COS 运算时, 该寄存器表示 SIN 计算结果; 进行 ARCTAN 运算时, 该寄存器用于 Y 值的输入; 运算过程禁止操作该寄存器。 复位值: 0x0

11.5.5 TMU 角度定点寄存器 (TMU_ANGLE_B)

偏移地址: 0x0020

表 124 TMU 角度定点寄存器

位	访问	描述
31:0	r/w	TMU_ANGLE_B: 角度值 注: 进行 SIN/COS 运算时, 该寄存器用于定点型角度输入; 进行 ARCTAN 运算时, 该寄存器表示计算结果; 运算过程禁止操作该寄存器。 复位值: 0x0

11.5.6 TMU 误差定点寄存器 (TMU_ERR_B)

偏移地址: 0x0014

表 125 TMU 误差定点寄存器

位	访问	描述
31:0	ro	TMU_ERR_B: 余弦值 (整型*2 ²⁴) 注: 进行 SIN/COS 运算时, 该寄存器表示角度误差; 进行 ARCTAN 运算时, 该寄存器表示 Y 值误差; 对该寄存器进行写, 无实际意义; 运算过程禁止操作该寄存器。 复位值: 0x0

11.5.7 TMU 余弦定点寄存器 (TMU_COSX_B)

偏移地址: 0x0028

表 126 TMU 余弦定点寄存器

位	访问	描述
31:0	r/w	TMU_COSX_B: 余弦值 (整型*2 ²⁴) 注: 进行 SIN/COS 运算时, 该寄存器表示 COS 计算结果; 进行 ARCTAN 运算时, 该寄存器用于 X 值的输入; 运算过程禁止操作该寄存器 复位值: 0x0

11.5.8 TMU 正弦定点寄存器 (TMU_SINY_B)

偏移地址: 0x002C

表 127 TMU 正弦定点寄存器

位	访问	描述
31:0	r/w	TMU_SINY_B: 正弦值 (整型*2 ²⁴) 注: 进行 SIN/COS 运算时, 该寄存器表示 SIN 计算结果; 进行 ARCTAN 运算时, 该寄存器用于 Y 值的输入; 运算过程禁止操作该寄存器。 复位值: 0x0

11.5.9 TMU 状态寄存器 (TMU_ST)

偏移地址: 0x0030

表 128 TMU 状态寄存器

位	访问	描述
31:10	Res	Reserved 复位值: 0x0
9:8	ro	ANGLE_LOAD: 角度所在象限 00: 第一象限 01: 第二象限 10: 第三象限 11: 第四象限 复位值: 0x0
7:5	Res	Reserved 复位值: 0x0
4	ro	OVER_ERR: 输入数据超出错误 0: 输入数据符合输入范围 1: 输入数据超出输入范围 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2	r	INPUT_MOD: 当前运算输入类型 0: binary 1: float 复位值: 0x0
1	r	CAL_MOD: 当前运算模式 0: SIN/COS 1: ARCTAN 复位值: 0x0
0	R	CAL_BUSY: TMU 忙标志 0: TMU 完成计算 1: TMU 正在计算 复位值: 0x0

12 模/数转换器 (ADC)

12.1 概述

KPM32R14TX 配备 2 个 12bit 分辨率的逐次逼近型模数转换器，每个模数转换器支持最多 13 个通道的模拟信号输入，其中，ADC0 的转换通道中，9 个通道连接到芯片外部，而 ADC1 有 11 个通道连接到芯片外部。模数转换器支持可配置最高 2.4Msps 的转换率，控制电路部分基于转换单元机制，16 个转换单元支持独立配置通道，采样时间等，并且，转换结果保存在基于转换单元的结果寄存器中。

12.2 主要功能

- 支持最高可达 2.4Msps 采样率
- 支持 9/11 个外部通道采样和 4/2 个内部通道采样
- 支持 12-bit/10-bit/8-bit 分辨率
- 支持 16 个可独立配置的转换单元
- 支持 16 个独立访问的转换结果寄存器
- 支持软件触发及硬件事件触发
- 支持突发连续转换模式
- 支持可配置高优先级转换单元
- 支持 4 个后处理模块
 - 偏移矫正
 - 偏移参考
 - 看门狗比较模式
 - 转换结果过零检测
- 支持转换完成事件及后处理事件发送
- 支持独立灵活配置中断使能

12.3 模块框图

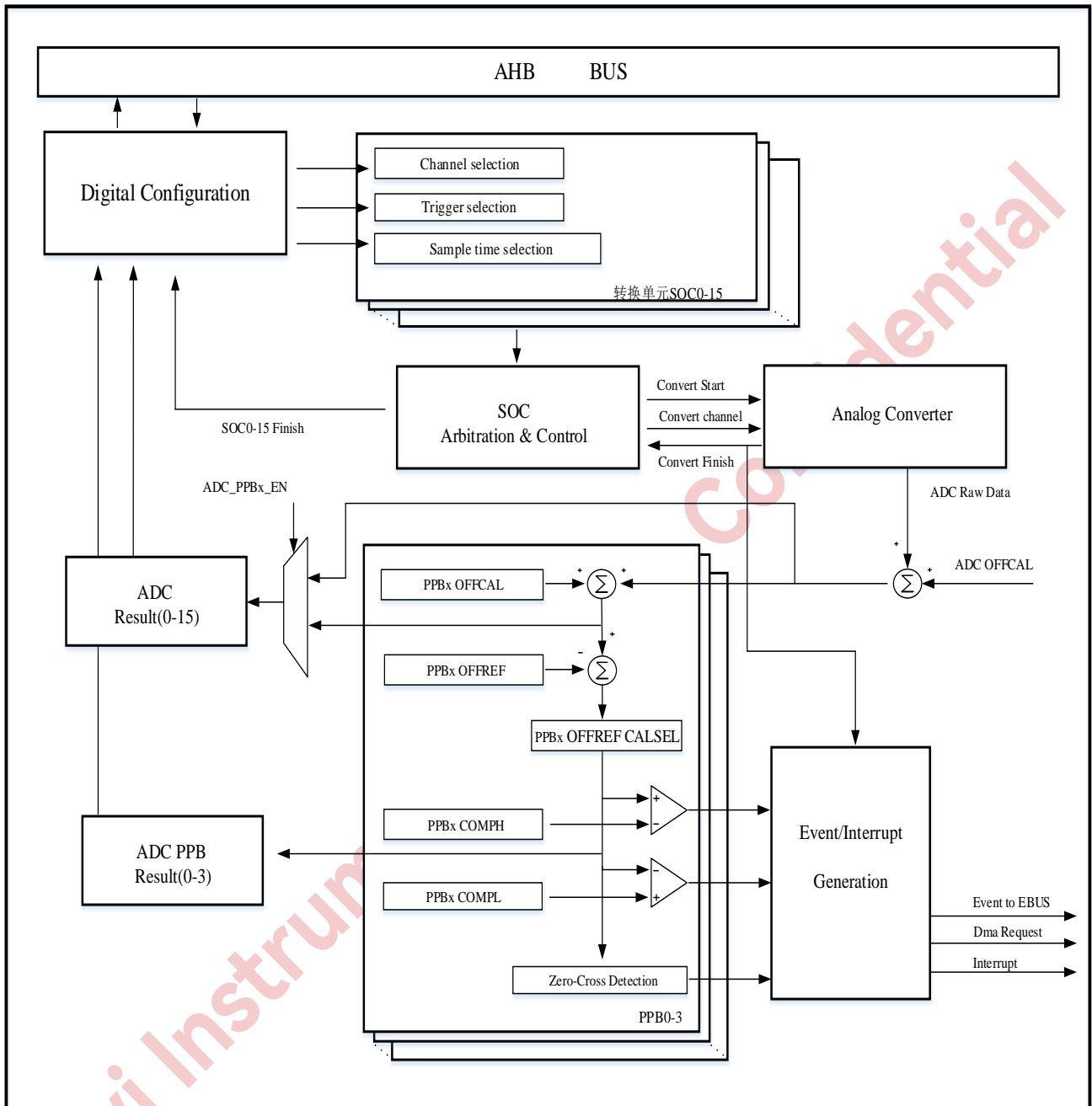


图 24 模数转换器模块架构框图

12.4 功能描述

12.4.1 相关配置

模数转换器的配置分为全局配置以及基于转换单元的配置，更改全局共用的相关配置，会对所有转换单元的转换都生效，基于转换单元的配置只会影响到相应转换单元的转换。

表 129 配置选项以及配置级别

配置选项	级别
时钟	全局
模拟通道使能	全局
硬件触发源	转换单元
采样时间	转换单元
转换通道	转换单元
突发转换模式	全局

12.4.2 转换顺序优先级

当多个转换单元 (SOC) 的触发信号同时被接收到时，这些 SOC 都将会变为待转换状态，此时，有两种机制决定这些 SOC 的转换顺序。

默认情况下，SOC 转换顺序采用轮询模式，所有的 SOC 具有相同的优先级别，转换顺序取决于当前转换单元轮询指针 (ADC_SOC_RRP) 的值，ADC_SOC_RRP 表示上一次被转换的 SOC，当前处于最高优先级的 SOC 为 ADC_SOC_RRP 的相邻下一个 SOC。当复位后，转换单元轮询指针 (ADC_SOC_RRP) 值为 16，此时表示 SOC0 具有最高优先级。

当高优先级转换单元 (ADC_HPRI_SEL) 所配置的值不为 0 时，此时，进入高优先级模式，部分 SOC (序号小于 ADC_HPRI_SEL 所配置的值) 具有高优先级别，其他 SOC 具有相同的优先级别；普通优先级的 SOC 部分，依然使用默认情况下的轮询机制，但当高优先级的 SOC 的触发信号被接收到时，普通优先级的轮询机制将会被打断 (当前 SOC 的转换会保持到完成)，高优先级的 SOC 将会优先被转换，当所有的被触发的高优先级 SOC 转换完成后，轮询模式被恢复，处于等待状态的普通优先级的 SOC 将会继续被转换，直到转换完成。需要注意的是，高优先级的 SOC 在转换过程中，转换单元轮询指针的值不会发生改变，高优先级的 SOC 内部顺序为按照序列从小到大进行转换。

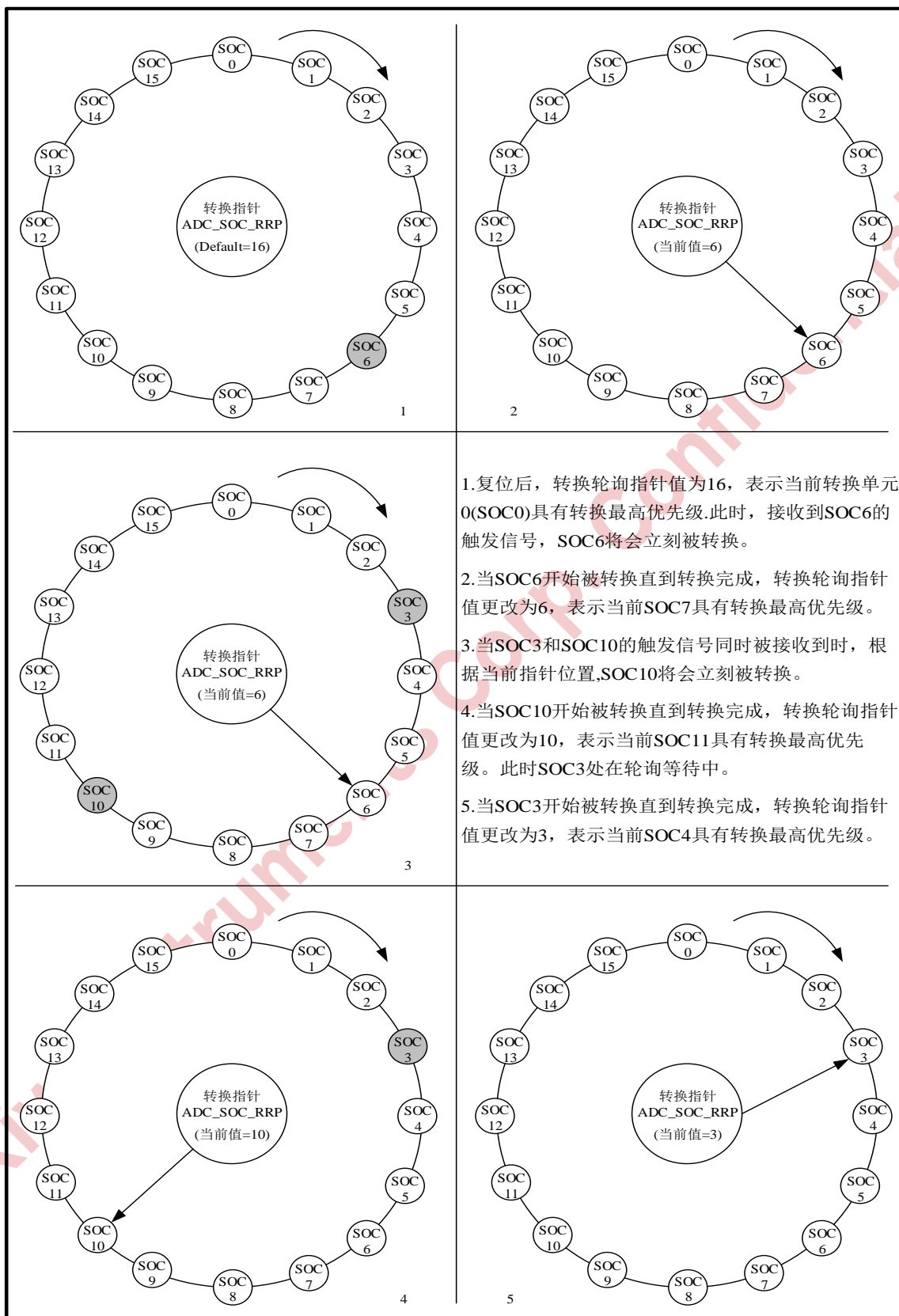


图 25 默认情况下的轮询模式

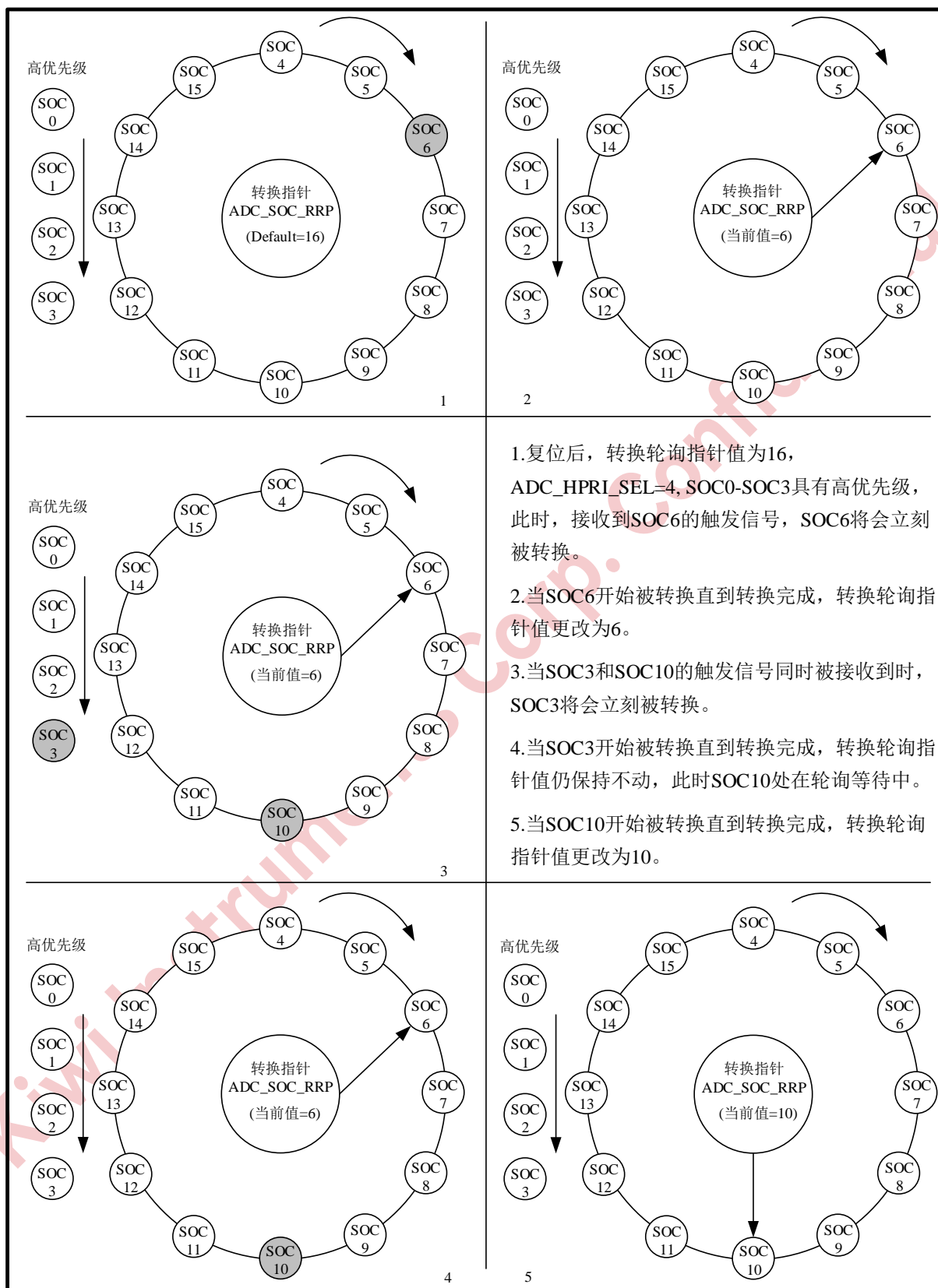


图 26 高优先级转换机制

12.4.3 突发转换模式

突发转换模式可以使一次触发转换固定数量的转换单元，通过配置 ADC 控制寄存器中 ADC_CTRL.ADC_BURST_EN 位使能突发转换模式，此时，在轮询模式下的转换单元（非高优先级转换单元）会进入突发转换模式，这些转换单元各自对应的触发信号将会失效，都由 ADC 控制寄存器中的突发转换模式下的触发信号来控制。

当接收到突发转换触发信号时，长度为 (ADC_CTRL.BURST_LEN+1) 的转换单元序列将会被转换，该序列的起始位置由当前轮询单元指针来决定。

处于高优先级的转换单元，不在突发转换模式的控制机制下，当高优先级转换单元的触发信号被接收到时，突发转换序列将会被打断（当前 SOC 的转换会保持到完成），高优先级的转换单元将会被逐一转换，当所有的被触发的高优先级转换单元转换完后，突发转换模式被恢复，处于等待状态的普通优先级的转换单元将会继续被转换，直到转换完成。需要注意的是，高优先级的转换单元在转换过程中，转换单元轮询指针的值不会发生改变，高优先级的转换单元内部顺序为按照序列从小到大进行转换。

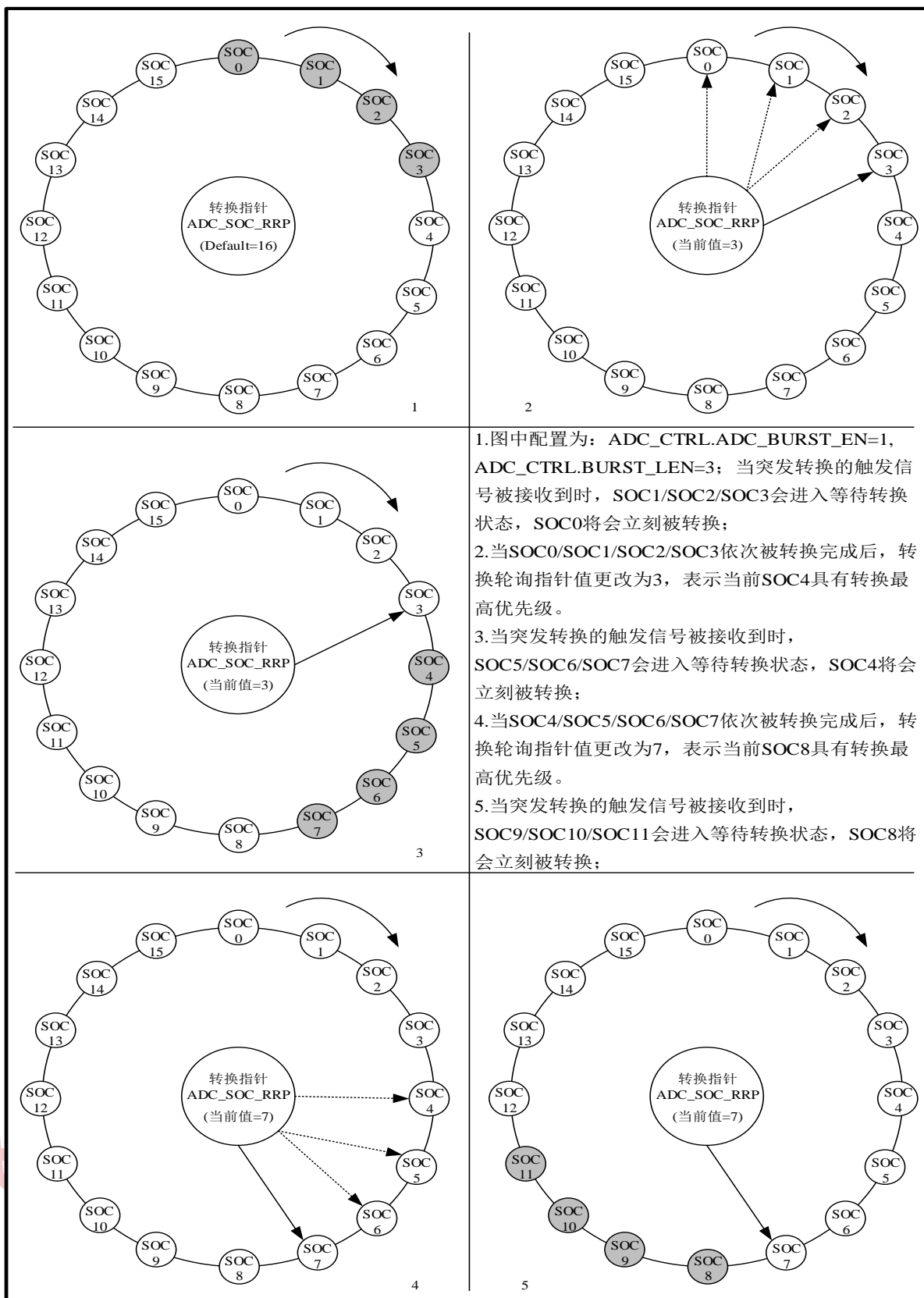


图 27 突发模式转换机制 (关闭高优先级模式)

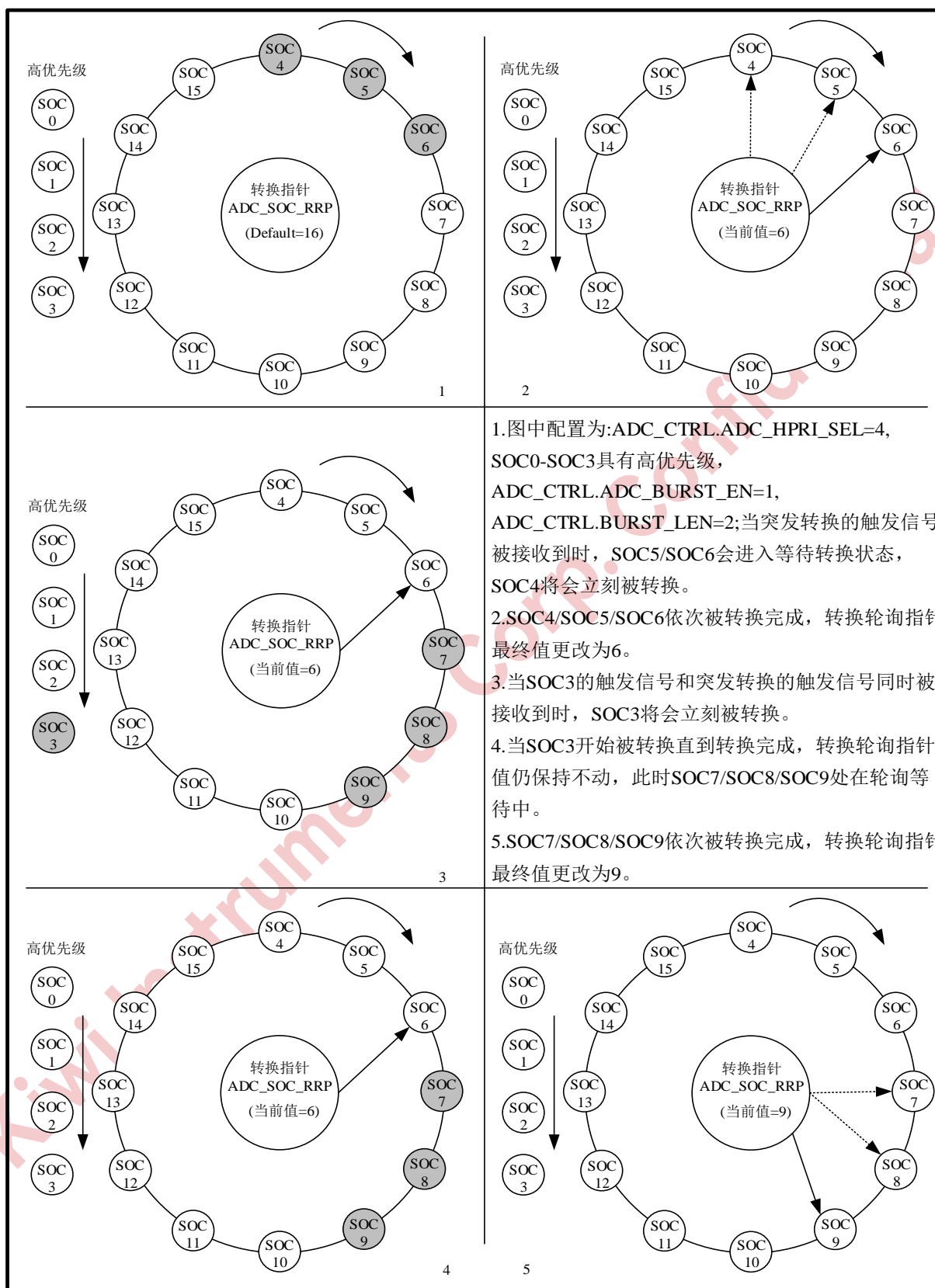


图 28 突发模式转换机制 (使能高优先级模式)

12.4.4 DMA 读取功能

模数转换器支持 DMA 功能，在控制寄存器使能 DMA 功能后，转换结果可以由 DMA 读取，存储到软件指定地址空间，当 DMA 读取功能使能时，每个转换单元的转换结果都会保存在转换单元 0 的结果寄存器中。

12.4.5 硬件触发功能

模数转换器支持硬件触发功能，每个转换单元都可以独立进行硬件触发功能使能及硬件事件源选择配置。

硬件触发功能在正常模式及突发转换模式下均可使能，在正常模式下，当单个或多个转换单元所选中的硬件触发源事件被接收到后，转换单元的转换顺序参考转换顺序优先级章节描述，在突发转换模式下，转换单元各自的配置寄存器中的硬件触发相关配置无效，由控制寄存器中的 `ADC_CTRL.BURST_HTRIG_EN` 以及 `ADC_CTRL.BURST_HTRIG_SEL` 位来决定（高优先级转换单元除外），当单个或多个转换单元所选中的硬件触发源事件被接收到后，转换单元的转换顺序参考突发转换模式章节描述。

12.4.6 零点偏移矫正

零点偏移错误发生在当模数转换器转换 0 值参考电压时，会出现结果不为 0 的情况。零点偏移量既可能是正值，也有可能为负值。因此，ADC 偏移矫正寄存器提供了 -64~63 的可配置矫正范围，用于矫正零点偏移错误。

ADC 偏移矫正寄存器中配置的矫正值会影响所有转换单元的结果，即每个转换单元对应的 ADC 结果寄存器中存储的是矫正后的结果。

12.4.7 结果后处理

模数转换器包含 4 个结果后处理模块，这 4 个后处理模块相应功能以及配置完全相同，每个模块都可以和任意一个转换单元的转换结果相连接，在 `ADC_PPbX_CFG` 寄存器中通过 `ADC_PPbX_CFG.PPBx_SOC_SEL` 位进行选择。

后处理模块包含以下四种功能：

- 对转换单元对应输入通道转换的电压值进行一个零点偏移电压的矫正
- 减去一个参考偏移电压
- 自动比较，产生电压阈值上限溢出状态和下限溢出状态
- 后处理结果过 0 检测

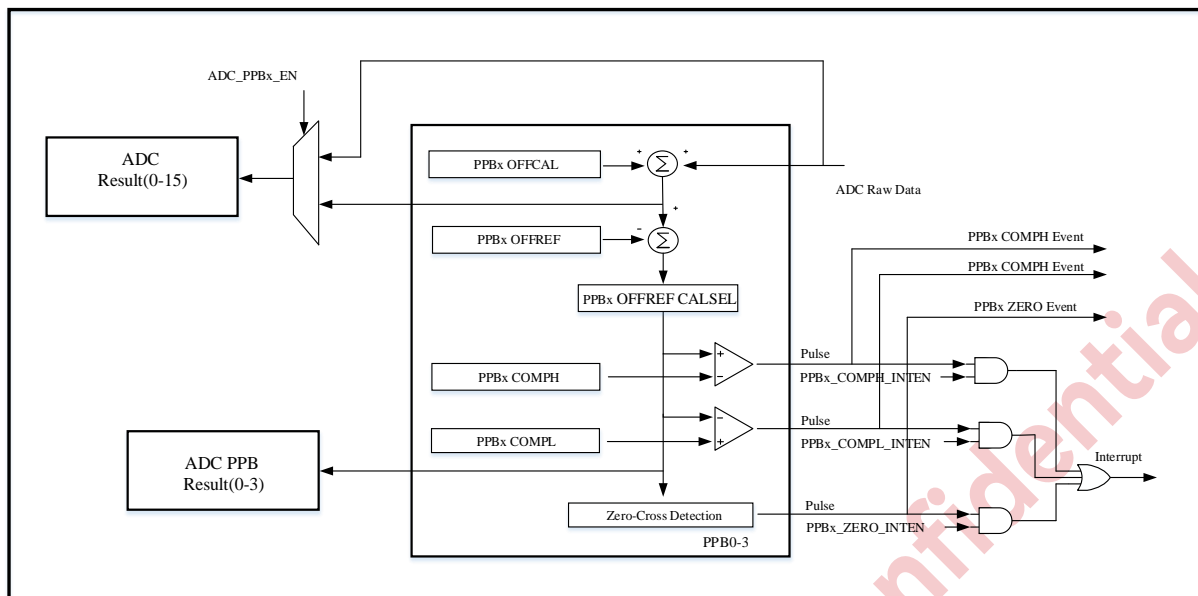


图 29 后处理模块示意图

12.4.7.1 偏移校正

在很多应用中，外部传感器或者一些信号源会产生不同的偏移电压，一个全局的偏移校正调整可能不足以适用这些不同的偏移从不同通道输入产生的影响，后处理模块可以对不同的通道进行不同的偏移校正，软件可以直接通过对应转换单元的 `ADC_RESULTx` 寄存器获取校正后的结果，以节省程序负担。

在通过 `ADC_PPbX_CFG.PPBx_SOC_SEL` 位选择相应的转换单元后，配置 `ADC_PPbX_OFFCAL` 寄存器，该转换单元的转换结果会自动加上该寄存器所配置的偏移电压值，结果由硬件进行饱和处理。

12.4.7.2 参考电压计算

在很多应用中，一个期望值需要在转换后的结果上进行计算，以及在一些必须要双极信号来进行控制计算或者双极信号可以简化运算的情况下，后处理模块可以自动对零点偏移校正后的结果进行参考电压的运算，以节省程序负担。

在通过 `ADC_PPbX_CFG.PPBx_SOC_SEL` 位选择相应的转换单元后，该转换单元的结果经过零点偏移校正后，后处理模块将自动对校正后的结果减去一个参考电压值，参考电压通过 `ADC_PPbX_OFFREF` 寄存器进行配置。同时，在将相减后的结果存放在 `ADC_PPbX_RESULT` 之前，软件可以通过 `ADC_PPbX_CFG.PPBx_OFFREF_CALSEL` 位来使能一个相反数的运算操作。

12.4.7.3 自动比较

结果后处理模块支持硬件自动比较功能，可作为模拟看门狗使用，比较后会产生上溢和下溢两种独立的状态标志位，通过配置中断使能寄存器相应位，可以产生比较中断。同时，自动比较的结果状态会作为硬件事件发送到 `EBUS` 模块中。

在通过 ADC_PPBx_CFG.PPBx_SOC_SEL 位选择相应的转换单元后，相应转换单元的转换结果经过偏移校正和参考电压的计算后，都会与 ADC_PPBx_COMPH 寄存器和 ADC_PPBx_COMPL 寄存器所配置的阈值进行比较，当计算结果超过上限值时，中断状态寄存器中的 ADC_STA.PPBx_COMPH_INT 状态位会被置位，当计算结果小于下限值时，中断状态寄存器的 ADC_STA.PPBx_COMPL_INT 状态位会被置位。

12.4.7.4 过零检测

结果后处理模块会对每次处理后的最终结果进行过零检测，每次当 ADC_PPBx_RESULT 中的结果符号发生变化时，相应后处理模块的过零检测状态标志位会被置位，通过配置中断使能寄存器相应位，可以产生比较中断。同时，过零检测状态会作为硬件事件发送到 EBUS 模块中。

在通过 ADC_PPBx_CFG.PPBx_SOC_SEL 位选择相应的转换单元后，过零检测会一直作用在相应转换单元，当处理后的结果满足过零检测条件时，中断状态寄存器中的 ADC_STA.PPBx_ZERO_INT 状态位会被置位。

12.4.8 ADC 中断号

表 130 ADC 中断号

名称	中断号
模数转换器 1	20
模数转换器 2	21

12.5 寄存器

本章节描述了模数转换器的相关寄存器，本器件中包含 2 个模数转换器模块，每个模数寄存器模块的寄存器分布及内容相同，通过基址区分。ADC0 和 ADC1 的基址地址分别为 0x4003_2000，0x4003_3000。

表 131 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	ADC 软件触发寄存器	32	wo	0x00000000
0x04	ADC 控制寄存器	32	r/w	0x00000000
0x10	ADC 转换单元优先级寄存器	32	r/w	0x00000000
0x20	ADC 中断使能寄存器	32	r/w	0x00000000
0x24	ADC 中断状态寄存器	32	ro	0x00000000
0x28	ADC 中断状态清除寄存器	32	wo	0x00000000
0x30	ADC 转换单元触发溢出寄存器	32	ro	0x00000000
0x34	ADC 转换单元触发溢出清除寄存器	32	wo	0x00000000

偏移地址	寄存器列表	位宽	访问	复位值
0x40	ADC 转换结果寄存器 0	32	r/w	0x00000000
0x44	ADC 转换结果寄存器 1	32	r/w	0x00000000
0x48	ADC 转换结果寄存器 2	32	r/w	0x00000000
0x4C	ADC 转换结果寄存器 3	32	r/w	0x00000000
0x50	ADC 转换结果寄存器 4	32	r/w	0x00000000
0x54	ADC 转换结果寄存器 5	32	r/w	0x00000000
0x58	ADC 转换结果寄存器 6	32	r/w	0x00000000
0x5C	ADC 转换结果寄存器 7	32	r/w	0x00000000
0x60	ADC 转换结果寄存器 8	32	r/w	0x00000000
0x64	ADC 转换结果寄存器 9	32	r/w	0x00000000
0x68	ADC 转换结果寄存器 10	32	r/w	0x00000000
0x6C	ADC 转换结果寄存器 11	32	r/w	0x00000000
0x70	ADC 转换结果寄存器 12	32	r/w	0x00000000
0x74	ADC 转换结果寄存器 13	32	r/w	0x00000000
0x78	ADC 转换结果寄存器 14	32	r/w	0x00000000
0x7C	ADC 转换结果寄存器 15	32	r/w	0x00000000
0x90	ADC PPB0 转换结果寄存器 0	32	r/w	0x00000000
0x94	ADC PPB1 转换结果寄存器 0	32	r/w	0x00000000
0x98	ADC PPB2 转换结果寄存器 0	32	r/w	0x00000000
0x9C	ADC PPB3 转换结果寄存器 0	32	r/w	0x00000000
0xB0	ADC 模拟通道使能寄存器	32	r/w	0x00000000
0xB4	ADC 偏移校正寄存器	32	r/w	0x00000000
0xB8	ADC 转换器配置寄存器	32	r/w	0x00000000
0xC0	ADC 转换单元配置寄存器 0	32	r/w	0x00000000
0xC4	ADC 转换单元配置寄存器 1	32	r/w	0x00000000
0xC8	ADC 转换单元配置寄存器 2	32	r/w	0x00000000
0xCC	ADC 转换单元配置寄存器 3	32	r/w	0x00000000
0xD0	ADC 转换单元配置寄存器 4	32	r/w	0x00000000

偏移地址	寄存器列表	位宽	访问	复位值
0xD4	ADC 转换单元配置寄存器 5	32	r/w	0x00000000
0xD8	ADC 转换单元配置寄存器 6	32	r/w	0x00000000
0xDC	ADC 转换单元配置寄存器 7	32	r/w	0x00000000
0xE0	ADC 转换单元配置寄存器 8	32	r/w	0x00000000
0xE4	ADC 转换单元配置寄存器 9	32	r/w	0x00000000
0xE8	ADC 转换单元配置寄存器 10	32	r/w	0x00000000
0xEC	ADC 转换单元配置寄存器 11	32	r/w	0x00000000
0xF0	ADC 转换单元配置寄存器 12	32	r/w	0x00000000
0xF4	ADC 转换单元配置寄存器 13	32	r/w	0x00000000
0xF8	ADC 转换单元配置寄存器 14	32	r/w	0x00000000
0xFC	ADC 转换单元配置寄存器 15	32	r/w	0x00000000
0x110	ADC PPB0 设置寄存器	32	r/w	0x00000000
0x114	ADC PPB0 偏移矫正寄存器	32	r/w	0x00000000
0x118	ADC PPB0 偏移参考寄存器	32	r/w	0x00000000
0x11C	ADC PPB0 比较上限寄存器	32	r/w	0x00000FFF
0x120	ADC PPB0 比较下限寄存器	32	r/w	0x00001000
0x130	ADC PPB1 设置寄存器	32	r/w	0x00000000
0x134	ADC PPB1 偏移矫正寄存器	32	r/w	0x00000000
0x138	ADC PPB1 偏移参考寄存器	32	r/w	0x00000000
0x13C	ADC PPB1 比较上限寄存器	32	r/w	0x00000FFF
0x140	ADC PPB1 比较下限寄存器	32	r/w	0x00001000
0x150	ADC PPB2 设置寄存器	32	r/w	0x00000000
0x154	ADC PPB2 偏移矫正寄存器	32	r/w	0x00000000
0x158	ADC PPB2 偏移参考寄存器	32	r/w	0x00000000
0x15C	ADC PPB2 比较上限寄存器	32	r/w	0x00000FFF
0x160	ADC PPB2 比较下限寄存器	32	r/w	0x00001000
0x170	ADC PPB3 设置寄存器	32	r/w	0x00000000
0x174	ADC PPB3 偏移矫正寄存器	32	r/w	0x00000000

偏移地址	寄存器列表	位宽	访问	复位值
0x178	ADC PPB3 偏移参考寄存器	32	r/w	0x00000000
0x17C	ADC PPB3 比较上限寄存器	32	r/w	0x00000FFF
0x180	ADC PPB3 比较下限寄存器	32	r/w	0x00001000

12.5.1 ADC 软件触发寄存器 (ADC_TRIG)

偏移地址: 0x0000

表 132 ADC 软件触发寄存器

位	访问	描述
31:18	Res	Reserved 复位值: 0x0
17	wo	ADC_STOP_TRIG: 软件停止信号 (硬件自动清 0) 0: 无影响 1: 停止当前工作 复位值: 0x0
16	wo	ADC_BURST_TRIG: 突发模式下, 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0
15	wo	ADC_SOC15_TRIG: 转换单元 15 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0
14	wo	ADC_SOC14_TRIG: 转换单元 14 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0
13	wo	ADC_SOC13_TRIG: 转换单元 13 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0
12	wo	ADC_SOC12_TRIG: 转换单元 12 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0
11	wo	ADC_SOC11_TRIG: 转换单元 11 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0

位	访问	描述
10	wo	ADC_SOC10_TRIG: 转换单元 10 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0
9	wo	ADC_SOC9_TRIG: 转换单元 9 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0
8	wo	ADC_SOC8_TRIG: 转换单元 8 软件触发(硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0
7	wo	ADC_SOC7_TRIG: 转换单元 7 软件触发(硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0
6	wo	ADC_SOC6_TRIG: 转换单元 6 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0
5	wo	ADC_SOC5_TRIG: 转换单元 5 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0
4	wo	ADC_SOC4_TRIG: 转换单元 4 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0
3	wo	ADC_SOC3_TRIG: 转换单元 3 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0
2	wo	ADC_SOC2_TRIG: 转换单元 2 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0
1	wo	ADC_SOC1_TRIG: 转换单元 1 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0

位	访问	描述
0	wo	ADC_SOC0_TRIG: 转换单元 0 软件触发 (硬件自动清 0) 0: 无触发 1: 触发开始 复位值: 0x0

12.5.2 ADC 控制寄存器 (ADC_CTRL)

偏移地址: 0x0004

表 133 ADC 控制寄存器

位	访问	描述
31:28	r/w	BURST_LEN: 突发模式转换长度 0000: 转换长度为 1 0001: 转换长度为 2 0010: 转换长度为 3 0011: 转换长度为 4 ... 1111: 转换长度为 16 复位值: 0x0
27:24	r/w	BURST_HTRIG_SEL: 突发模式硬件事件触发源选择 0000: EBUS 通道 0 事件输出 0001: EBUS 通道 1 事件输出 0010: EBUS 通道 2 事件输出 0011: EBUS 通道 3 事件输出 ... 1111: EBUS 通道 15 事件输出 复位值: 0x0
23:21	Res	Reserved 复位值: 0x0
20	r/w	BURST_HTRIG_EN: 突发模式硬件事件触发使能 0: 关闭 1: 使能 复位值: 0x0
19:17	Res	Reserved 复位值: 0x0
16	r/w	ADC_BURST_EN: 突发模式使能 0: 关闭突发模式 1: 使能突发模式 复位值: 0x0
15:13	Res	Reserved 复位值: 0x0

位	访问	描述
12	r/w	ADC_DMA_EN: DMA 功能使能 0: 关闭 DMA 功能 1: 使能 DMA 功能 复位值: 0x0
11:6	Res	Reserved 复位值: 0x0
5:4	r/w	ADC_RES: ADC 分辨率模式 00: 12-bit 模式 01: 10-bit 模式 10: 8-bit 模式 11: Reserved 复位值: 0x0
3:2	r/w	ADC_MODE: ADC 工作模式使能 00: 停止 01: 单次转换模式 10: 停止 11: Reserved 复位值: 0x0
1	Res	Reserved 复位值: 0x0
0	r/w	ADC_EN: ADC 使能 0: 关闭 1: 使能 复位值: 0x0

12.5.3 ADC 转换单元优先级寄存器 (ADC_SOC_PRI)

偏移地址: 0x0010

表 134 ADC 转换单元优先级控制寄存器

位	访问	描述
31:28	ro	ADC_SOC_CUR: 当前转换单元 当 ADC 处于繁忙时, 此部分表示正在工作的转换单元 当 ADC 处于空闲时, 此部分表示上一次工作的转换单元 复位值: 0x0
27:13	Res	Reserved 复位值: 0x0
12:8	ro	ADC_SOC_RRP: 转换单元轮询指针 00000: 转换单元 0 在上次被转换完成, 转换单元 1 具有最高优先级 00001: 转换单元 1 在上次被转换完成, 转换单元 2 具有最高优先级 00010: 转换单元 2 在上次被转换完成, 转换单元 3 具有最高优先级 00011: 转换单元 3 在上次被转换完成, 转换单元 4 具有最高优先级

位	访问	描述
		... 01110: 转换单元 14 在上次被转换完成, 转换单元 15 具有最高优先级 01111: 转换单元 15 在上次被转换完成, 转换单元 0 具有最高优先级 10000: 转换单元 0 具有最高优先级 其他值保留 复位值: 0x10 注意: 每次配置 ADC_HPRI_SEL 位时, 转换指针会被复位。
7:5	Res	Reserved 复位值: 0x0
4:0	r/w	ADC_HPRI_SEL: 高优先级转换单元选择 00000: 所有转换单元优先级相同, 都为轮询模式 00001: 转换单元 0 具有高优先级, 转换单元 1-15 为轮询模式 00010: 转换单元 0 和 1 具有高优先级, 转换单元 2-15 为轮询模式 00011: 转换单元 0,1 和 2 具有高优先级, 转换单元 3-15 为轮询模式 ... 01110: 转换单元 0-13 具有高优先级, 转换单元 14-15 为轮询模式 01111: 转换单元 0-14 具有高优先级, 转换单元 15 为轮询模式 10000: 所有转换单元均为高优先级模式 其他值保留 复位值: 0x0

12.5.4 ADC 中断使能寄存器 (ADC_INTEN)

偏移地址: 0x0020

表 135 ADC 中断使能寄存器

位	访问	描述
31:30	Res	Reserved 复位值: 0x0
29	r/w	SOC_OVF_INTEN: 任一转换单元触发溢出中断使能 0: 关闭 1: 使能 复位值: 0x0
28	r/w	SOC_FNS_INTEN: 任一转换单元转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
27	r/w	PPB3_ZERO_INTEN: PPB3 结果过零检测中断使能 0: 关闭 1: 使能 复位值: 0x0

位	访问	描述
26	r/w	PPB3_COMPL_INTEN: PPB3 结果比较小于下限值中断使能 0: 关闭 1: 使能 复位值: 0x0
25	r/w	PPB3_COMPH_INTEN: PPB3 结果比较大于上限值中断使能 0: 关闭 1: 使能 复位值: 0x0
24	r/w	PPB2_ZERO_INTEN: PPB2 结果过零检测中断使能 0: 关闭 1: 使能 复位值: 0x0
23	r/w	PPB2_COMPL_INTEN: PPB2 结果比较小于下限值中断使能 0: 关闭 1: 使能 复位值: 0x0
22	r/w	PPB2_COMPH_INTEN: PPB2 结果比较大于上限值中断使能 0: 关闭 1: 使能 复位值: 0x0
21	r/w	PPB1_ZERO_INTEN: PPB1 结果过零检测中断使能 0: 关闭 1: 使能 复位值: 0x0
20	r/w	PPB1_COMPL_INTEN: PPB1 结果比较小于下限值中断使能 0: 关闭 1: 使能 复位值: 0x0
19	r/w	PPB1_COMPH_INTEN: PPB1 结果比较大于上限值中断使能 0: 关闭 1: 使能 复位值: 0x0
18	r/w	PPB0_ZERO_INTEN: PPB0 结果过零检测中断使能 0: 关闭 1: 使能 复位值: 0x0
17	r/w	PPB0_COMPL_INTEN: PPB0 结果比较小于下限值中断使能 0: 关闭 1: 使能 复位值: 0x0

位	访问	描述
16	r/w	PPB0_COMPH_INTEN: PPB0 结果比较大于上限值中断使能 0: 关闭 1: 使能 复位值: 0x0
15	r/w	SOC15_FNS_INTEN: 转换单元 15 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
14	r/w	SOC14_FNS_INTEN: 转换单元 14 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
13	r/w	SOC13_FNS_INTEN: 转换单元 13 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
12	r/w	SOC12_FNS_INTEN: 转换单元 12 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
11	r/w	SOC11_FNS_INTEN: 转换单元 11 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
10	r/w	SOC10_FNS_INTEN: 转换单元 10 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
9	r/w	SOC9_FNS_INTEN: 转换单元 9 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
8	r/w	SOC8_FNS_INTEN: 转换单元 8 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
7	r/w	SOC7_FNS_INTEN: 转换单元 7 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0

位	访问	描述
6	r/w	SOC6_FNS_INTEN: 转换单元 6 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	SOC5_FNS_INTEN: 转换单元 5 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	SOC4_FNS_INTEN: 转换单元 4 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	SOC3_FNS_INTEN: 转换单元 3 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
2	r/w	SOC2_FNS_INTEN: 转换单元 2 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
1	r/w	SOC1_FNS_INTEN: 转换单元 1 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0
0	r/w	SOC0_FNS_INTEN: 转换单元 0 单次转换结束中断使能 0: 关闭 1: 使能 复位值: 0x0

12.5.5 ADC 中断状态寄存器 (ADC_STA)

偏移地址: 0x0024

表 136 ADC 中断状态寄存器

位	访问	描述
31	ro	ADC_BUSY: ADC 工作状态标志 0: 转换停止或者未开始 1: 转换正在进行 复位值: 0x0
30	Res	Reserved 复位值: 0x0
29	ro	SOC_OVF_STA: 任一转换单元触发溢出状态标志 0: 无状态发生 1: 触发事件溢出 复位值: 0x0
28	ro	SOC_FNS_STA: 任一转换单元转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0
27	ro	PPB3_ZERO_INT: PPB3 结果过零检测状态标志 0: 无状态发生 1: 转换结果符号改变 复位值: 0x0
26	ro	PPB3_COMPL_INT: PPB3 结果比较小于下限值状态标志 0: 无状态发生 1: 小于下限值 复位值: 0x0
25	ro	PPB3_COMPH_INT: PPB3 结果比较大于上限值状态标志 0: 无状态发生 1: 大于上限值 复位值: 0x0
24	ro	PPB2_ZERO_INT: PPB2 结果过零检测状态标志 0: 无状态发生 1: 转换结果符号改变 复位值: 0x0
23	ro	PPB2_COMPL_INT: PPB2 结果比较小于下限值状态标志 0: 无状态发生 1: 小于下限值 复位值: 0x0
22	ro	PPB2_COMPH_INT: PPB2 结果比较大于上限值状态标志 0: 无状态发生 1: 大于上限值

位	访问	描述
		复位值: 0x0
21	ro	PPB1_ZERO_INT: PPB1 结果过零检测状态标志 0: 无状态发生 1: 转换结果符号改变 复位值: 0x0
20	ro	PPB1_COMPL_INT: PPB1 结果比较小于下限值状态标志 0: 无状态发生 1: 小于下限值 复位值: 0x0
19	ro	PPB1_COMPH_INT: PPB1 结果比较大于上限值状态标志 0: 无状态发生 1: 大于上限值 复位值: 0x0
18	ro	PPB0_ZERO_INT: PPB0 结果过零检测状态标志 0: 无状态发生 1: 转换结果符号改变 复位值: 0x0
17	ro	PPB0_COMPL_INT: PPB0 结果比较小于下限值状态标志 0: 无状态发生 1: 小于下限值 复位值: 0x0
16	ro	PPB0_COMPH_INT: PPB0 结果比较大于上限值状态标志 0: 无状态发生 1: 大于上限值 复位值: 0x0
15	ro	SOC15_FNS_STA: 转换单元 15 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0
14	ro	SOC14_FNS_STA: 转换单元 14 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0
13	ro	SOC13_FNS_STA: 转换单元 13 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0

位	访问	描述
12	ro	SOC12_FNS_STA: 转换单元 12 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0
11	ro	SOC11_FNS_STA: 转换单元 11 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0
10	ro	SOC10_FNS_STA: 转换单元 10 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0
9	ro	SOC9_FNS_STA: 转换单元 9 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0
8	ro	SOC8_FNS_STA: 转换单元 8 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0
7	ro	SOC7_FNS_STA: 转换单元 7 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0
6	ro	SOC6_FNS_STA: 转换单元 6 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0
5	ro	SOC5_FNS_STA: 转换单元 5 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0
4	ro	SOC4_FNS_STA: 转换单元 4 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0
3	ro	SOC3_FNS_STA: 转换单元 3 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0

位	访问	描述
2	ro	SOC2_FNS_STA: 转换单元 2 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0
1	ro	SOC1_FNS_STA: 转换单元 1 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0
0	ro	SOC0_FNS_STA: 转换单元 0 转换结束状态标志 0: 无状态发生 1: 单次转换完成 复位值: 0x0

12.5.6 ADC 中断状态清除寄存器 (ADC_STA_CLR)

偏移地址: 0x0028

表 137 ADC 中断状态清除寄存器

位	访问	描述
31:30	Res	Reserved 复位值: 0x0
29	wo	SOC_OVF_STA: 任一转换单元触发溢出状态清除 0: 不清除 1: 清除 复位值: 0x0
28	wo	SOC_FNS_STA: 任一转换单元转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0
27	wo	PPB3_ZERO_INT: PPB3 结果过零检测状态清除 0: 不清除 1: 清除转换结果符号改变 复位值: 0x0
26	wo	PPB3_COMPL_INT: PPB3 结果比较小于下限值状态清除 0: 不清除 1: 清除 复位值: 0x0
25	wo	PPB3_COMPH_INT: PPB3 结果比较大于上限值状态清除 0: 不清除 1: 清除 复位值: 0x0

位	访问	描述
24	wo	PPB2_ZERO_INT: PPB2 结果过零检测状态清除 0: 不清除 1: 清除 复位值: 0x0
23	wo	PPB2_COMPL_INT: PPB2 结果比较小于下限值状态清除 0: 不清除 1: 清除 复位值: 0x0
22	wo	PPB2_COMPH_INT: PPB2 结果比较大于上限值状态清除 0: 不清除 1: 清除 复位值: 0x0
21	wo	PPB1_ZERO_INT: PPB1 结果过零检测状态清除 0: 不清除 1: 清除 复位值: 0x0
20	wo	PPB1_COMPL_INT: PPB1 结果比较小于下限值状态清除 0: 不清除 1: 清除 复位值: 0x0
19	wo	PPB1_COMPH_INT: PPB1 结果比较大于上限值状态清除 0: 不清除 1: 清除 复位值: 0x0
18	wo	PPB0_ZERO_INT: PPB0 结果过零检测状态清除 0: 不清除 1: 清除 复位值: 0x0
17	wo	PPB0_COMPL_INT: PPB0 结果比较小于下限值状态清除 0: 不清除 1: 清除 复位值: 0x0
16	wo	PPB0_COMPH_INT: PPB0 结果比较大于上限值状态清除 0: 不清除 1: 清除 复位值: 0x0
15	wo	SOC15_FNS_STA: 转换单元 15 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0

位	访问	描述
14	wo	SOC14_FNS_STA: 转换单元 14 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0
13	wo	SOC13_FNS_STA: 转换单元 13 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0
12	wo	SOC12_FNS_STA: 转换单元 12 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0
11	wo	SOC11_FNS_STA: 转换单元 11 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0
10	wo	SOC10_FNS_STA: 转换单元 10 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0
9	wo	SOC9_FNS_STA: 转换单元 9 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0
8	wo	SOC8_FNS_STA: 转换单元 8 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0
7	wo	SOC7_FNS_STA: 转换单元 7 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0
6	wo	SOC6_FNS_STA: 转换单元 6 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0
5	wo	SOC5_FNS_STA: 转换单元 5 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0

位	访问	描述
4	wo	SOC4_FNS_STA: 转换单元 4 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0
3	wo	SOC3_FNS_STA: 转换单元 3 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0
2	wo	SOC2_FNS_STA: 转换单元 2 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0
1	wo	SOC1_FNS_STA: 转换单元 1 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0
0	wo	SOC0_FNS_STA: 转换单元 0 转换结束状态清除 0: 不清除 1: 清除 复位值: 0x0

12.5.7 ADC 转换单元触发溢出寄存器 (ADC_TRIG_OVF)

偏移地址: 0x0030

表 138 ADC 转换单元触发溢出寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15	ro	ADC_SOC15_OVF: 转换单元 15 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0
14	ro	ADC_SOC14_OVF: 转换单元 14 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0
13	ro	ADC_SOC13_OVF: 转换单元 13 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0

位	访问	描述
12	ro	ADC_SOC12_OVF: 转换单元 12 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0
11	ro	ADC_SOC11_OVF: 转换单元 11 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0
10	ro	ADC_SOC10_OVF: 转换单元 10 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0
9	ro	ADC_SOC9_OVF: 转换单元 9 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0
8	ro	ADC_SOC8_OVF: 转换单元 8 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0
7	ro	ADC_SOC7_OVF: 转换单元 7 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0
6	ro	ADC_SOC6_OVF: 转换单元 6 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0
5	ro	ADC_SOC5_OVF: 转换单元 5 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0
4	ro	ADC_SOC4_OVF: 转换单元 4 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0
3	ro	ADC_SOC3_OVF: 转换单元 3 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0

位	访问	描述
2	ro	ADC_SOC2_OVF: 转换单元 2 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0
1	ro	ADC_SOC1_OVF: 转换单元 1 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0
0	ro	ADC_SOC0_OVF: 转换单元 0 触发事件溢出标志 0: 无触发溢出 1: 触发事件溢出 复位值: 0x0

12.5.8 ADC 转换单元触发溢出清除寄存器 (ADC_OVF_CLR)

偏移地址: 0x0034

表 139 ADC 转换单元触发溢出清除寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15	wo	ADC_SOC15_OVF: 清除转换单元 15 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0
14	wo	ADC_SOC14_OVF: 清除转换单元 14 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0
13	wo	ADC_SOC13_OVF: 清除转换单元 13 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0
12	wo	ADC_SOC12_OVF: 清除转换单元 12 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0
11	wo	ADC_SOC11_OVF: 清除转换单元 11 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0

位	访问	描述
10	wo	ADC_SOC10_OVF: 清除转换单元 10 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0
9	wo	ADC_SOC9_OVF: 清除转换单元 9 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0
8	wo	ADC_SOC8_OVF: 清除转换单元 8 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0
7	wo	ADC_SOC7_OVF: 清除转换单元 7 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0
6	wo	ADC_SOC6_OVF: 清除转换单元 6 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0
5	wo	ADC_SOC5_OVF: 清除转换单元 5 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0
4	wo	ADC_SOC4_OVF: 清除转换单元 4 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0
3	wo	ADC_SOC3_OVF: 清除转换单元 3 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0
2	wo	ADC_SOC2_OVF: 清除转换单元 2 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0
1	wo	ADC_SOC1_OVF: 清除转换单元 1 触发事件溢出标志 0: 无影响 1: 清除触发事件溢出 复位值: 0x0
0	wo	ADC_SOC0_OVF: 清除转换单元 0 触发事件溢出标志

位	访问	描述
		0: 无影响 1: 清除触发事件溢出 复位值: 0x0

12.5.9 ADC 转换单元配置寄存器 (ADC_SOCx) (x=0,1,2...15)

偏移地址: 0x00C0 0x00C4 0x00C8 0x00CC ... 0x00FC

表 140 ADC 转换单元配置寄存器

位	访问	描述
31:25	Res	Reserved 复位值: 0x0
24:20	r/w	<p>ADC_CHN_SEL: ADC 转换通道选择</p> <p>00000: 对于 ADC0, 选择外部 ADC0_IN0 通道进行转换; 对于 ADC1, 选择外部 ADC1_IN0 通道进行转换</p> <p>00001: 对于 ADC0, 选择外部 ADC0_IN1 通道进行转换; 对于 ADC1, 选择外部 ADC1_IN1 通道进行转换</p> <p>00010: 对于 ADC0, 选择外部 ADC0_IN2 通道进行转换; 对于 ADC1, 选择外部 ADC1_IN2 通道进行转换</p> <p>00011: 对于 ADC0, 选择外部 ADC0_IN3 通道进行转换; 对于 ADC1, 选择外部 ADC1_IN3 通道进行转换</p> <p>00100: 对于 ADC0, 选择外部 ADC0_IN4 通道进行转换; 对于 ADC1, 选择外部 ADC1_IN4 通道进行转换</p> <p>00101: 对于 ADC0, 选择外部 ADC0_IN5 通道进行转换; 对于 ADC1, 选择外部 ADC1_IN5 通道进行转换</p> <p>00110: 对于 ADC0, 选择外部 ADC0_IN6 通道进行转换; 对于 ADC1, 选择外部 ADC1_IN6 通道进行转换</p> <p>00111: 对于 ADC0, 选择外部 ADC0_IN7 通道进行转换; 对于 ADC1, 选择外部 ADC1_IN7 通道进行转换</p> <p>01000: 对于 ADC0, 选择外部 ADC0_IN8 通道进行转换; 对于 ADC1, 选择外部 ADC1_IN8 通道进行转换</p> <p>01001: 对于 ADC0, 选择内部 XVREFN0 通道进行转换; 对于 ADC1, 选择外部 ADC1_IN9 通道进行转换</p> <p>01010: 对于 ADC0, 选择内部 TS_OUT 通道进行转换; 对于 ADC1, 选择外部 ADC1_IN10 通道进行转换</p> <p>01011: 对于 ADC0, 选择内部 XVDD 通道进行转换; 对于 ADC1, 选择内部 TS_OUT 通道进行转换</p> <p>01100: 对于 ADC0, 选择内部 XVREFP0 通道进行转换; 对于 ADC1, 选择内部 XVDD 通道进行转换</p> <p>01101-11111: 保留</p>

位	访问	描述
		复位值: 0x0
19:16	Res	Reserved 复位值: 0x0
15:12	r/w	ADC_ST: 采样时间设置 0000: 3 个 ADC 模拟时钟 0001: 4 个 ADC 模拟时钟 0010: 6 个 ADC 模拟时钟 0011: 8 个 ADC 模拟时钟 0100: 12 个 ADC 模拟时钟 0101: 16 个 ADC 模拟时钟 0110: 24 个 ADC 模拟时钟 0111: 32 个 ADC 模拟时钟 1000: 48 个 ADC 模拟时钟 1001: 64 个 ADC 模拟时钟 1010: 96 个 ADC 模拟时钟 1011: 128 个 ADC 模拟时钟 1100: 192 个 ADC 模拟时钟 1101: 256 个 ADC 模拟时钟 1110: 384 个 ADC 模拟时钟 1111: 512 个 ADC 模拟时钟 复位值: 0x4
11:8	Res	Reserved 复位值: 0x0
7:4	r/w	ADC_HTRIG_SEL: ADC 硬件事件触发源选择 0000: EBUS 通道 0 事件输出 0001: EBUS 通道 1 事件输出 0010: EBUS 通道 2 事件输出 0011: EBUS 通道 3 事件输出 ... 1111: EBUS 通道 15 事件输出 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	ADC_HTRIG_EN: ADC 硬件事件触发使能 0: 关闭 1: 使能 复位值: 0x0

12.5.10 ADC PPB0 设置寄存器 (ADC_PPB0_CFG)

偏移地址: 0x0110

表 141 ADC PPB0 设置寄存器

位	访问	描述
31:9	Res	Reserved 复位值: 0x0
8	r/w	PPB0_OFFREF_CALSEL: PPB0 偏移参考计算方式选择 0: ADC_RESULTx - PPB0_OFFREF 1: PPB0_OFFREF - ADC_RESULTx 复位值: 0x0
7:4	r/w	PPB0_SOC_SEL: 转换单元选择 0000: 选择转换单元 0 0001: 选择转换单元 1 0010: 选择转换单元 2 0011: 选择转换单元 3 ... 1111: 选择转换单元 15 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	ADC_PPB0_EN: PPB0 使能控制 0: 关闭 1: 使能 复位值: 0x0

12.5.11 ADC PPB0 偏移校正寄存器 (ADC_PPB0_OFFCAL)

偏移地址: 0x0114

表 142 ADC PPB0 偏移校正寄存器

位	访问	描述
31:10	Res	Reserved 复位值: 0x0
9:0	r/w	PPB0_OFFCAL: PPB0 偏移校正值 000h: +0 001h: +1 002h: +2 ... 1ffh: +511 200h: -512 201h: -511 202h: -510 203h: -509 ... 3ffh: -1 复位值: 0x0

12.5.12 ADC PPB0 偏移参考寄存器 (ADC_PPB0_OFFREF)

偏移地址: 0x0118

表 143 ADC PPB0 偏移参考寄存器

位	访问	描述
31:12	Res	Reserved 复位值: 0x0
11:0	r/w	PPB0_OFFREF: PPB0 偏移参考值 (此部分没有符号位) 000h: 0 001h: 1 002h: 2 ... 7ffh: 2047 800h: 2048 801h: 2049 802h: 2050 803h: 2051 ... fffh: 4095 复位值: 0x0

12.5.13 ADC PPB0 比较上限寄存器 (ADC_PPB0_COMPH)

偏移地址: 0x011C

表 144 ADC PPB0 比较上限寄存器

位	访问	描述
31:13	ro	此部分为比较上限值的符号位 复位值: 0x0
12:0	r/w	PPB0_COMPH: PPB0 比较上限值 0000h: +0 0001h: +1 0002h: +2 ... 0fffh: +4095 1000h: -4096 1001h: -4095 1002h: -4094 1003h: -4093 ... 1fffh: -1 复位值: 0x1FFF

12.5.14 ADC PPB0 比较下限寄存器 (ADC_PPB0_COMPL)

偏移地址: 0x0120

表 145 ADC PPB0 比较下限寄存器

位	访问	描述
31:13	ro	此部分为比较上限值的符号位 复位值: 0x7FFFF
12:0	r/w	PPB0_CMPL: PPB0 比较下限值 0000h: +0 0001h: +1 0002h: +2 ... 0fffh: +4095 1000h: -4096 1001h: -4095 1002h: -4094 1003h: -4093 ... 1fffh: -1 复位值: 0x1000

12.5.15 ADC PPB1 设置寄存器 (ADC_PPB1_CFG)

偏移地址: 0x0130

表 146 ADC PPB1 设置寄存器

位	访问	描述
31:9	Res	Reserved 复位值: 0x0
8	r/w	PPB1_OFFREF_CALSEL: PPB1 偏移参考计算方式选择 0: ADC_RESULTx - PPB1_OFFREF 1: PB1_OFFREF - ADC_RESULTx 复位值: 0x0
7:4	r/w	PPB1_SOC_SEL: 转换单元选择 0000: 选择转换单元 0 0001: 选择转换单元 1 0010: 选择转换单元 2 0011: 选择转换单元 3 ... 1111: 选择转换单元 15 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	ADC_PPB1_EN: PPB1 使能控制 0: 关闭 1: 使能 复位值: 0x0

12.5.16 ADC PPB1 偏移校正寄存器 (ADC_PPB1_OFFCAL)

偏移地址: 0x0134

表 147 ADC PPB1 偏移校正寄存器

位	访问	描述
31:10	Res	Reserved 复位值: 0x0
9:0	r/w	PPB1_OFFCAL: PPB1 偏移校正 000h: +0 001h: +1 002h: +2 ... 1ffh: +511 200h: -512 201h: -511

位	访问	描述
		202h: -510 203h: -509 ... 3ffh: -1 复位值: 0x0

12.5.17 ADC PPB1 偏移参考寄存器 (ADC_PPB1_OFFREF)

偏移地址: 0x0138

表 148 ADC PPB1 偏移参考寄存器

位	访问	描述
31:12	Res	Reserved 复位值: 0x0
11:0	r/w	PPB1_OFFREF: PPB1 偏移参考值 (此部分没有符号位) 000h: 0 001h: 1 002h: 2 ... 7ffh: 2047 800h: 2048 801h: 2049 802h: 2050 803h: 2051 ... fffh: 4095 复位值: 0x0

12.5.18 ADC PPB1 比较上限寄存器 (ADC_PPB1_COMPH)

偏移地址: 0x013C

表 149 ADC PPB1 比较上限寄存器

位	访问	描述
31:13	ro	此部分为比较上限值的符号位 复位值: 0x0
12:0	r/w	PPB1_COMPH: PPB1 比较上限值 0000h: +0 0001h: +1 0002h: +2 ... 0fffh: +4095 1000h: -4096 1001h: -4095 1002h: -4094 1003h: -4093 ... 1fffh: -1 复位值: 0x1FFF

12.5.19 ADC PPB1 比较下限寄存器 (ADC_PPB1_COMPL)

偏移地址: 0x0140

表 150 ADC PPB1 比较下限寄存器

位	访问	描述
31:13	ro	此部分为比较上限值的符号位 复位值: 0x7FFFF
12:0	r/w	PPB1_CMPL: PPB1 比较下限值 0000h: +0 0001h: +1 0002h: +2 ... 0fffh: +4095 1000h: -4096 1001h: -4095 1002h: -4094 1003h: -4093 ... 1fffh: -1 复位值: 0x1000

12.5.20 ADC PPB2 设置寄存器 (ADC_PPB2_CFG)

偏移地址: 0x0150

表 151 ADC PPB2 设置寄存器

位	访问	描述
31:9	Res	Reserved 复位值: 0x0
8	r/w	PPB2_OFFREF_CALSEL: PPB2 偏移参考计算方式选择 0: ADC_RESULTx - PPB2_OFFREF 1: PPB2_OFFREF - ADC_RESULTx 复位值: 0x0
7:4	r/w	PPB2_SOC_SEL: 转换单元选择 0000: 选择转换单元 0 0001: 选择转换单元 1 0010: 选择转换单元 2 0011: 选择转换单元 3 ... 1111: 选择转换单元 15 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	ADC_PPB2_EN: PPB2 使能控制 0: 关闭 1: 使能 复位值: 0x0

12.5.21 ADC PPB2 偏移校正寄存器 (ADC_PPB2_OFFCAL)

偏移地址: 0x0154

表 152 ADC PPB2 偏移校正寄存器

位	访问	描述
31:10	Res	Reserved 复位值: 0x0
9:0	r/w	PPB2_OFFCAL: PPB2 偏移校正 000h: +0 001h: +1 002h: +2 ... 1ffh: +511 200h: -512 201h: -511

位	访问	描述
		202h: -510 203h: -509 ... 3ffh: -1 复位值: 0x0

12.5.22 ADC PPB2 偏移参考寄存器 (ADC_PP2_OFFREF)

偏移地址: 0x0158

表 153 ADC PPB2 偏移参考寄存器

位	访问	描述
31:12	Res	Reserved 复位值: 0x0
11:0	r/w	PPB2_OFFREF: PPB2 偏移参考值 (此部分没有符号位) 000h: 0 001h: 1 002h: 2 ... 7ffh: 2047 800h: 2048 801h: 2049 802h: 2050 803h: 2051 ... fffh: 4095 复位值: 0x0

12.5.23 ADC PPB2 比较上限寄存器 (ADC_PP2_COMPH)

偏移地址: 0x015C

表 154 ADC PPB2 比较上限寄存器

位	访问	描述
31:13	ro	此部分为比较上限值的符号位 复位值: 0x0
12:0	r/w	PPB2_COMPH: PPB2 比较上限值 0000h: +0 0001h: +1 0002h: +2 ... 0fffh: +4095 1000h: -4096 1001h: -4095 1002h: -4094 1003h: -4093 ... 1fffh: -1 复位值: 0x1FFF

12.5.24 ADC PPB2 比较下限寄存器 (ADC_PP2_COMPL)

偏移地址: 0x0160

表 155 ADC PPB2 比较下限寄存器

位	访问	描述
31:13	ro	此部分为比较上限值的符号位 复位值: 0x7FFFF
12:0	r/w	PPB2_CMPL: PPB2 比较下限值 0000h: +0 0001h: +1 0002h: +2 ... 0fffh: +4095 1000h: -4096 1001h: -4095 1002h: -4094 1003h: -4093 ... 1fffh: -1 复位值: 0x1000

12.5.25 ADC PPB3 设置寄存器 (ADC_PPB3_CFG)

偏移地址: 0x0170

表 156 ADC PPB3 设置寄存器

位	访问	描述
31:9	Res	Reserved 复位值: 0x0
8	r/w	PPB3_OFFREF_CALSEL: PPB3 偏移参考计算方式选择 0: ADC_RESULTx - PPB3_OFFREF 1: PPB3_OFFREF - ADC_RESULTx 复位值: 0x0
7:4	r/w	PPB3_SOC_SEL: 转换单元选择 0000: 选择转换单元 0 0001: 选择转换单元 1 0010: 选择转换单元 2 0011: 选择转换单元 3 ... 1111: 选择转换单元 15 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	ADC_PPB3_EN: PPB3 使能控制 0: 关闭 1: 使能 复位值: 0x0

12.5.26 ADC PPB3 偏移校正寄存器 (ADC_PPB3_OFFCAL)

偏移地址: 0x0174

表 157 ADC PPB3 偏移校正寄存器

位	访问	描述
31:10	Res	Reserved 复位值: 0x0
9:0	r/w	PPB3_OFFCAL: PPB3 偏移校正 000h: +0 001h: +1 002h: +2 ... 1ffh: +511 200h: -512 201h: -511

位	访问	描述
		202h: -510 203h: -509 ... 3ffh: -1 复位值: 0x0

12.5.27 ADC PPB3 偏移参考寄存器 (ADC_PP3_OFFREF)

偏移地址: 0x0178

表 158 ADC PPB3 偏移参考寄存器

位	访问	描述
31:12	Res	Reserved 复位值: 0x0
11:0	r/w	PPB3_OFFREF: PPB3 偏移参考值 (此部分没有符号位) 000h: 0 001h: 1 002h: 2 ... 7ffh: 2047 800h: 2048 801h: 2049 802h: 2050 803h: 2051 ... fffh: 4095 复位值: 0x0

12.5.28 ADC PPB3 比较上限寄存器 (ADC_PP3_COMPH)

偏移地址: 0x017C

表 159 ADC PPB3 比较上限寄存器

位	访问	描述
31:13	ro	此部分为比较上限值的符号位 复位值: 0x0
12:0	r/w	PPB3_COMPH: PPB3 比较上限值 0000h: +0 0001h: +1 0002h: +2 ... 0fffh: +4095

位	访问	描述
		1000h: -4096 1001h: -4095 1002h: -4094 1003h: -4093 ... 1ffffh: -1 复位值: 0x1FFF

12.5.29 ADC PPB3 比较下限寄存器 (ADC_PP3_COMPL)

偏移地址: 0x0180

表 160 ADC PPB3 比较下限寄存器

位	访问	描述
31:13	ro	此部分为比较上限值的符号位 复位值: 0x7FFFF
12:0	r/w	PPB3_CMPL: PPB3 比较下限值 0000h: +0 0001h: +1 0002h: +2 ... 0ffffh: +4095 1000h: -4096 1001h: -4095 1002h: -4094 1003h: -4093 ... 1ffffh: -1 复位值: 0x1000

12.5.30 ADC 转换结果寄存器 (ADC_RESULTx) (x=0,1,2...15)

偏移地址: 0x0040 0x0044 0x0048 0x004C ...0x007C

表 161 ADC 转换结果寄存器

位	访问	描述
31:12	Res	Reserved 复位值: 0x0
11:0	r/w	ADC_RESULTx: ADC 转换单元 x 对应的转换结果 x 复位值: 0x0

12.5.31 ADC PPB0 转换结果寄存器 (ADC_PPB0_RESULT)

偏移地址: 0x0090

表 162 ADC PPB0 转换结果寄存器

位	访问	描述
31:13	ro	与 ADC_PPB0_RESULT 符号位值保持一致 复位值: 0x0
12:0	ro	ADC_PPB0_RESULT: ADC PPB0 处理后的结果 复位值: 0x0

12.5.32 ADC PPB1 转换结果寄存器 (ADC_PPB1_RESULT)

偏移地址: 0x0094

表 163 ADC PPB1 转换结果寄存器

位	访问	描述
31:13	ro	与 ADC_PPB1_RESULT 符号位值保持一致 复位值: 0x0
12:0	ro	ADC_PPB1_RESULT: ADC PPB1 处理后的结果 复位值: 0x0

12.5.33 ADC PPB2 转换结果寄存器 (ADC_PPB2_RESULT)

偏移地址: 0x0098

表 164 ADC PPB2 转换结果寄存器

位	访问	描述
31:13	ro	与 ADC_PPB2_RESULT 符号位值保持一致 复位值: 0x0
12:0	ro	ADC_PPB2_RESULT: ADC PPB2 处理后的结果 复位值: 0x0

12.5.34 ADC PPB3 转换结果寄存器 (ADC_PPB3_RESULT)

偏移地址: 0x009C

表 165 ADC PPB3 转换结果寄存器

位	访问	描述
31:13	ro	与 ADC_PPB3_RESULT 符号位值保持一致 复位值: 0x0
12:0	ro	ADC_PPB3_RESULT: ADC PPB3 处理后的结果 复位值: 0x0

12.5.35 ADC 模拟通道使能寄存器 (ADC_CHNEN)

偏移地址: 0x00B0

表 166 ADC 模拟通道使能寄存器

位	访问	描述
31: 11	Res	Reserved 复位值: 0x0
10	r/w	ADC_CHN_EN: ADC 模拟通道 10 使能 0: 通道关闭 1: 通道使能 对于 ADC0: 保留 对于 ADC1: 此位控制外部通道 10 (ADC1_IN10) 复位值: 0x0
9	r/w	ADC_CHN_EN: ADC 模拟通道 9 使能 0: 通道关闭 1: 通道使能 对于 ADC0: 保留 对于 ADC1: 此位控制外部通道 9 (ADC1_IN9) 复位值: 0x0
8	r/w	ADC_CHN_EN: ADC 模拟通道 8 使能 0: 通道关闭 1: 通道使能 对于 ADC0: 此位控制外部通道 8 (ADC0_IN8) 对于 ADC1: 此位控制外部通道 8 (ADC1_IN8) 复位值: 0x0
7	r/w	ADC_CHN_EN: ADC 模拟通道 7 使能 0: 通道关闭 1: 通道使能 对于 ADC0: 此位控制外部通道 7 (ADC0_IN7) 对于 ADC1: 此位控制外部通道 7 (ADC1_IN7) 复位值: 0x0
6	r/w	ADC_CHN_EN: ADC 模拟通道 6 使能 0: 通道关闭 1: 通道使能 对于 ADC0: 此位控制外部通道 6 (ADC0_IN6) 对于 ADC1: 此位控制外部通道 6 (ADC1_IN6) 复位值: 0x0
5	r/w	ADC_CHN_EN: ADC 模拟通道 5 使能 0: 通道关闭 1: 通道使能 对于 ADC0: 此位控制外部通道 5 (ADC0_IN5) 对于 ADC1: 此位控制外部通道 5 (ADC1_IN5)

位	访问	描述
		复位值: 0x0
4	r/w	ADC_CHN_EN: ADC 模拟通道 4 使能 0: 通道关闭 1: 通道使能 对于 ADC0: 此位控制外部通道 4 (ADC0_IN4) 对于 ADC1: 此位控制外部通道 4 (ADC1_IN4) 复位值: 0x0
3	r/w	ADC_CHN_EN: ADC 模拟通道 3 使能 0: 通道关闭 1: 通道使能 对于 ADC0: 此位控制外部通道 3 (ADC0_IN3) 对于 ADC1: 此位控制外部通道 3 (ADC1_IN3) 复位值: 0x0
2	r/w	ADC_CHN_EN: ADC 模拟通道 2 使能 0: 通道关闭 1: 通道使能 对于 ADC0: 此位控制外部通道 2 (ADC0_IN2) 对于 ADC1: 此位控制外部通道 2 (ADC1_IN2) 复位值: 0x0
1	r/w	ADC_CHN_EN: ADC 模拟通道 1 使能 0: 通道关闭 1: 通道使能 对于 ADC0: 此位控制外部通道 1 (ADC0_IN1) 对于 ADC1: 此位控制外部通道 1 (ADC1_IN1) 复位值: 0x0
0	r/w	ADC_CHN_EN: ADC 模拟通道 0 使能 0: 通道关闭 1: 通道使能 对于 ADC0: 此位控制外部通道 0 (ADC0_IN0) 对于 ADC1: 此位控制外部通道 0 (ADC1_IN0) 复位值: 0x0

12.5.36 ADC 偏移校正寄存器 (ADC_OFFCAL)

偏移地址: 0x00B4

表 167 ADC 偏移校正寄存器

位	访问	描述
31:7	Res	Reserved 复位值: 0x0
6:0	r/w	ADC_OFFCAL: 零点偏移校正 00h: +0 01h: +1 02h: +2 ... 3fh: +63 40h: -64 41h: -63 42h: -62 43h: -61 ... 7fh: -1 复位值: 0x0

12.5.37 ADC 转换器配置寄存器 (ADC_ANA_CFG)

偏移地址: 0x00B8

表 168 ADC 转换器配置寄存器

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	ADC_BIAS_SEL: 转换器电流偏置选择 00: 10uA 01: 12uA 10: 14uA 11: 20uA 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	r/w	ADC_CHDET: 转换器通道短路/开路检测 00: Open/Short detection disable 01: Zero Scale 10: Full Scale 11: 1/2 Scale 复位值: 0x0

位	访问	描述
0	r/w	ADC_TEST_EN: 转换器测试模式使能 0: 关闭 1: 使能 复位值: 0x0

Kiwi Instruments Corp. Confidential

13 简版定时器 (TIMS)

13.1 概述

该简版定时器 (TIMS) 是由一个带自动加载功能的 16 位计数器构成，支持可编程预分频计数。定时器可用于多种用途：对输入信号脉冲宽度量测、比较输出、PWM 输出或带死区插入的互补 PWM 输出。

13.2 主要功能

- 16 位递增、递减、递增/递减自动重载计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数介于 1 到 65536 之间
- 通道配置
 - 1 对互补通道
 - 1 个单通道
- 通道功能
 - 输入捕获
 - 输出比较
 - PWM 输出
 - 单脉冲模式输出
- 带可编程死区的互补输出
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路
- 支持同类定时时间协作使能操作和换向操作
- 支持断路输入将输出信号置于复位状态或预写状态
- 支持以下事件的中断生成：
 - 更新：计数器溢出，由软件或内外部事件触发计数器初始化
 - 触发事件（计数器开始、停止、初始化或内外部触发导致的计数动作）
 - 输入捕获
 - 输出比较

13.3 模块框图

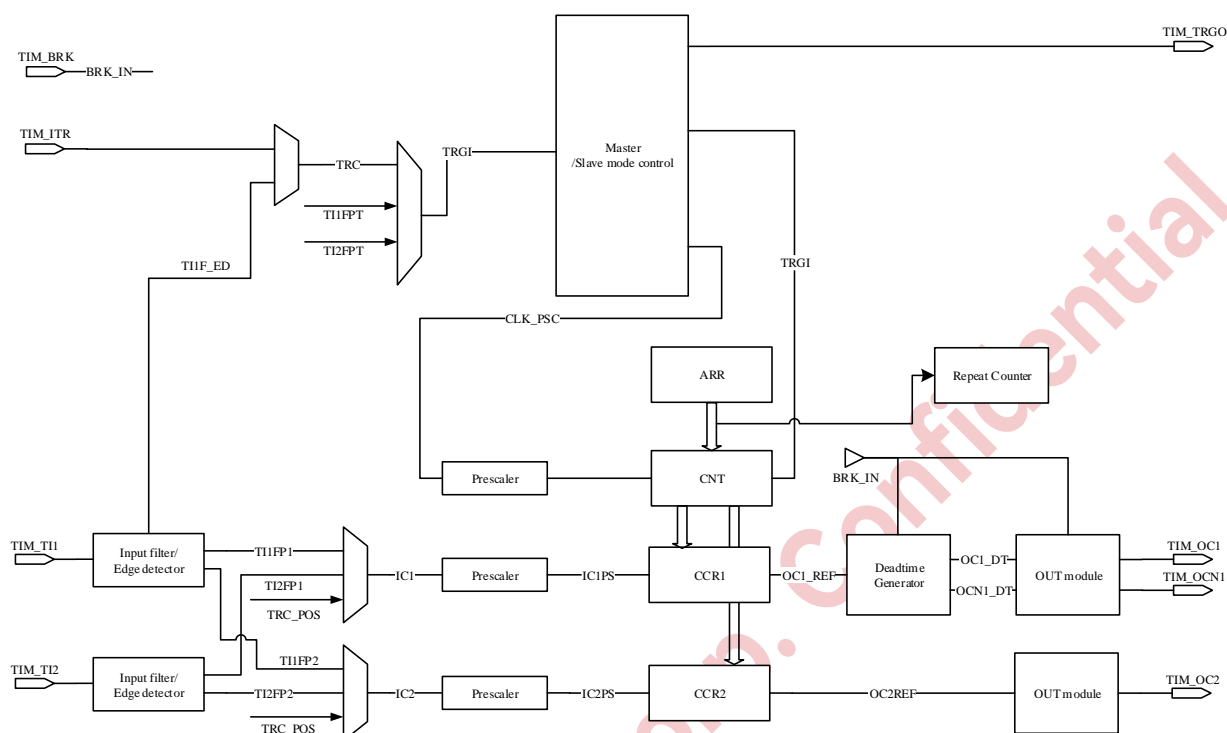


图 30 定时器 TMS 总体模块框图

13.4 功能描述

13.4.1 计数器介绍

可编程定时器的主要模块由一个 16 位计数器及其相关的自动重载寄存器组成。此计数器采用递增、递减或中心对齐的方式计数。计数器的时钟可通过预分频器进行分频。计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。时基单元包括计数器寄存器 (TIM_CNT)、预分频寄存器 (TIM_PSC) 和自动重载寄存器 (TIM_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器,也可以在每次发生更新事件 (UEV) 时传送到影子寄存器,这取决于 TIM_CR 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIM_CR 寄存器中的 UDIS 位为 0 时,将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CLK_CNT 提供时钟,仅当 TIM_CR 寄存器中的计数器启动位 (CEN) 置 1 时,才会启动计数器 (有关计数器使能的更多详细信息,另请参见从模式控制器的相关说明)。计数器是在 CEN 置 1 的一个时钟周期后方才真正开始计数。

预分频器可对计数器时钟频率进行分频,分频系数支持 1 到 65536。该预分频器基于 4 位寄存器 (TIM_PSC 寄存器) 所控制的 16 位计数器。由于该控制寄存器具有缓冲功能,因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

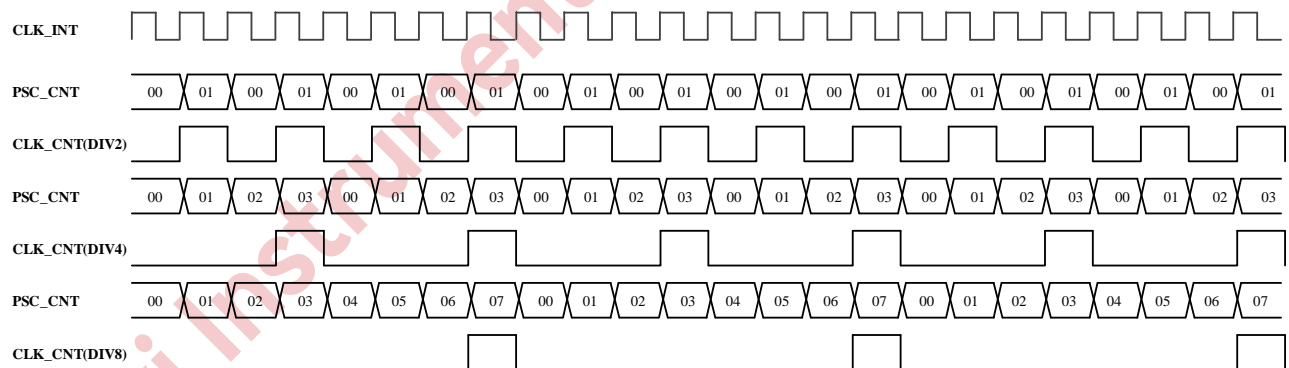


图 31 TIM 预分频器 N 分频时序图

13.4.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值 (TIM_ARR 寄存器的内容)，然后重新从 0 开始计数并生成计数器上溢事件。

如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中编程的次数加一次 (TIM_RCR+1) 后，将生成更新事件 (UEV)。否则，将在每次计数器上溢时产生更新事件。

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

通过软件将 TIM_CR 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数 (而预分频比保持不变)。此外，如果 TIM_CR 寄存器中的 URS 位 (更新请求选择) 已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1 (因此，不会发送任何中断或 DMA 请求)。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

以下各图以一些示例说明当 TIM_ARR=0x36 时不同时钟频率下计数器的行为。

- 重复计数器中将重新装载 TIM_RCR 寄存器的内容
- 自动重载影子寄存器将以预装载值 (TIM_ARR) 进行更新
- 预分频器的缓冲区中将重新装载预装载值 (TIM_PSC 寄存器的内容)

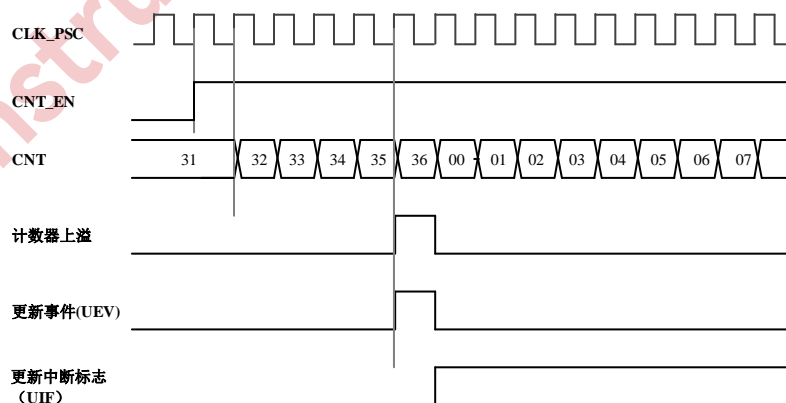


图 32 计数器时序图，1 分频内部时钟

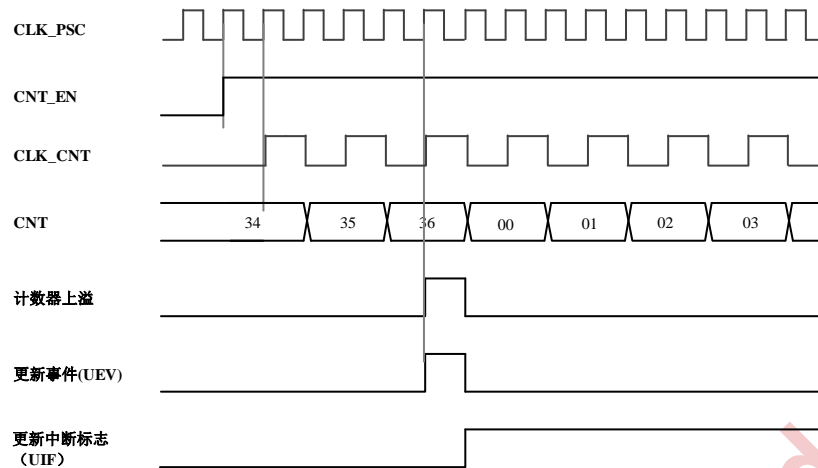


图 33 计数器时序图，2 分频内部时钟

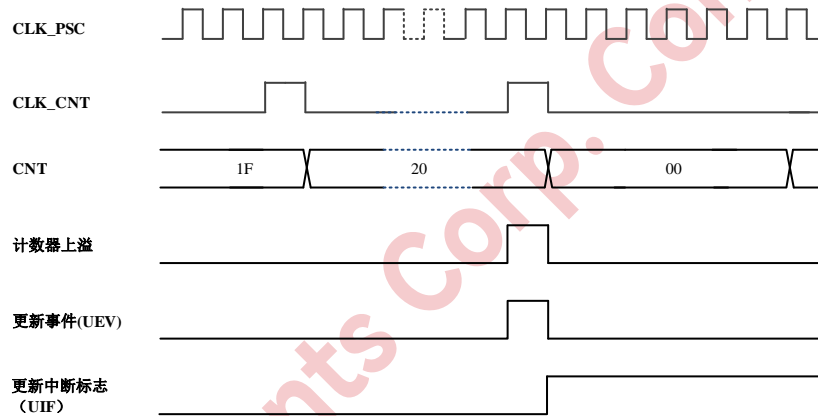


图 34 计数器时序图，N 分频内部时钟

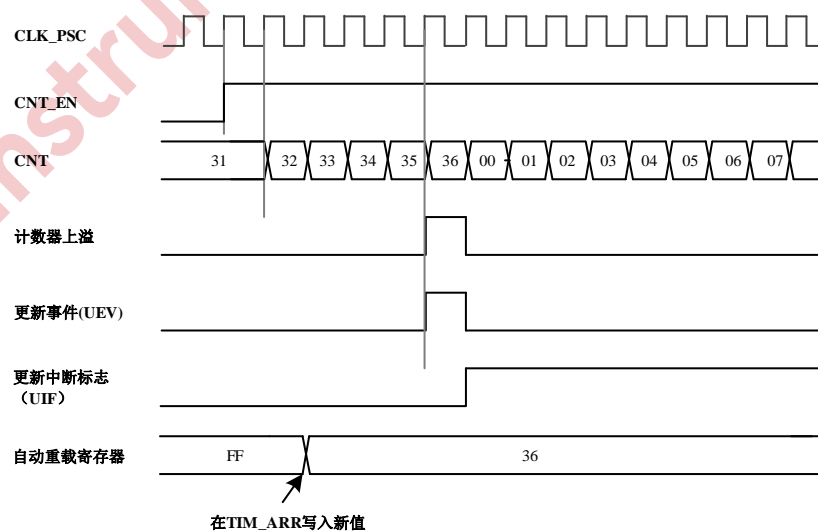


图 35 计数器时序图，ARPE=0 时更新事件 (TIM_ARR 未预装载)

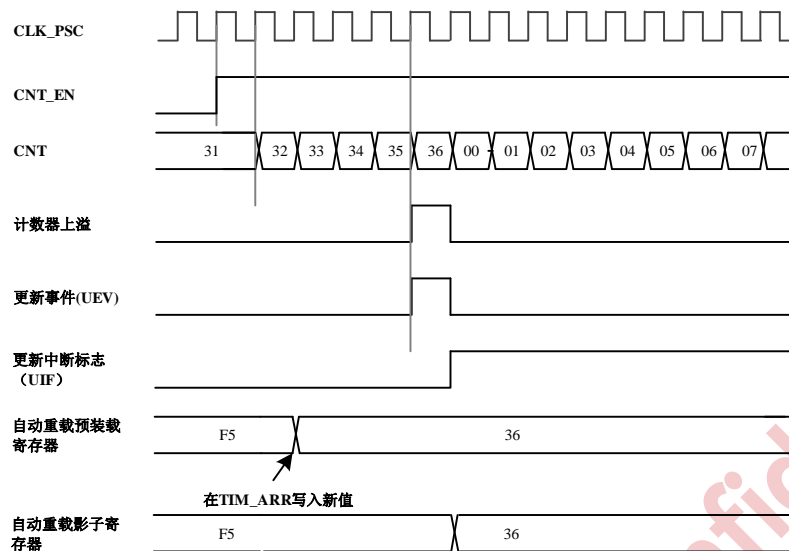


图 36 计数器时序图，ARPE=1 时更新事件 (TIM_ARR 预装载)

递减计数模式

在递减计数模式下，计数器从自动重载值 (TIM_ARR 寄存器的内容) 开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

如果使用重复计数器，则当递减计数的重复次数达到重复计数器寄存器中编程的次数加一次 (TIM_RCR+1) 后，将生成更新事件 (UEV)。否则，将在每次计数器下溢时产生更新事件。

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

通过软件将 TIM_CR 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数 (但预分频比保持不变)。

此外，如果 TIM_CR 寄存器中的 URS 位 (更新请求选择) 已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1 (因此，不会发送任何中断或 DMA 请求)。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 重复计数器中将重新装载 TIM_RCR 寄存器的内容
- 预分频器的缓冲区中将重新装载预装载值 (TIM_PSC 寄存器的内容)
- 自动重载活动寄存器将以预装载值 (TIM_ARR 寄存器的内容) 进行更新。注意，自动重载寄存器会在计数器重载之前得到更新，因此，下一个计数周期就是我们所希望的新的周期长度。

以下各图以一些示例说明当 TIM_ARR=0x36 时不同时钟频率下计数器的行为。

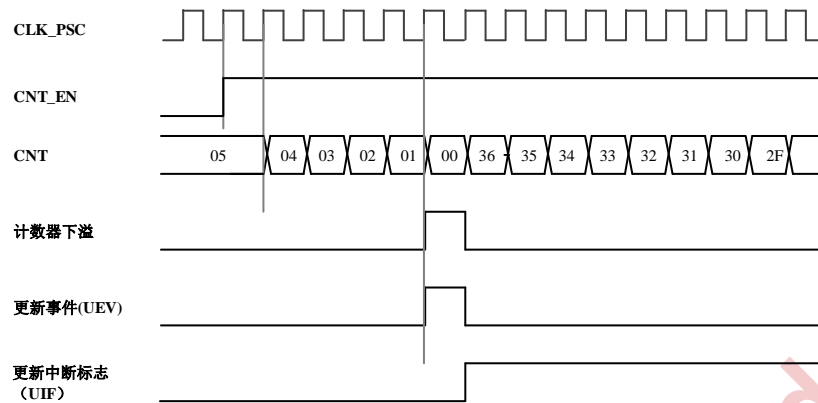


图 37 计数器时序图，1 分频内部时钟

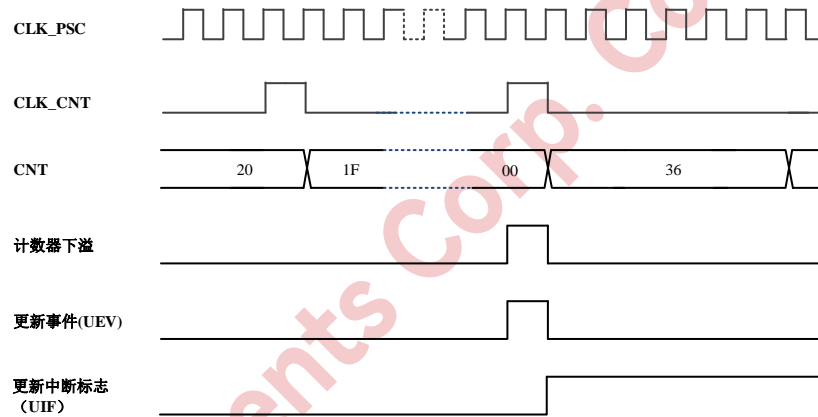


图 38 计数器时序图，N 分频内部时钟

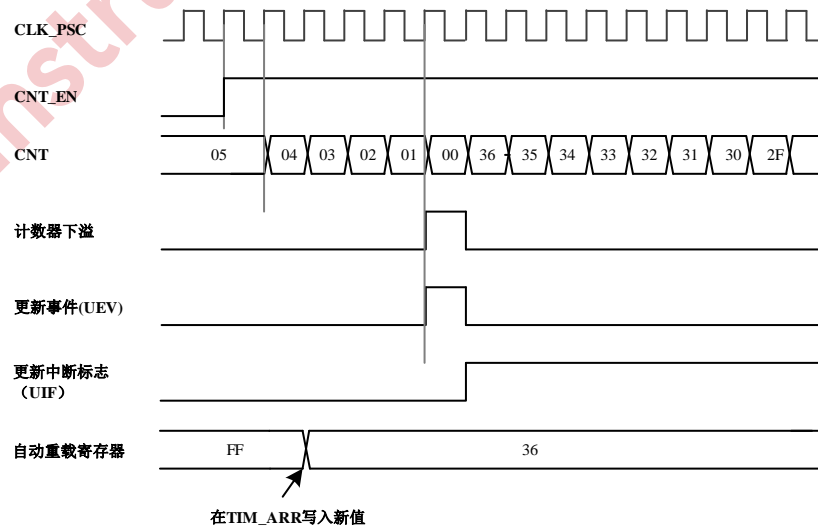


图 39 计数器时序图，未使用重复计数器时更新事件

中心对齐模式 (递增/递减计数)

在中心对齐模式下，计数器从 0 开始计数到自动重载值 (TIM_ARR 寄存器的内容) -1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIM_CR 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数 (中心对齐模式 1，CMS=“01”)、计数器递增计数 (中心对齐模式 2，CMS=“10”) 以及计数器递增/递减计数 (中心对齐模式 3，CMS=“11”)。

在此模式下，TIM_CR 寄存器的 DIR 方向位不可写入值，而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TIM_EGR 寄存器中的 UG 位置 1 (通过软件或使用从模式控制器) 也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

通过软件将 TIM_CR 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

此外，如果 TIM_CR 寄存器中的 URS 位 (更新请求选择) 已置 1，则将 UG 位置 1 会生成 UEV 更新事件，但不会将 UIF 标志置 1 (因此，不会发送任何中断或 DMA 请求)。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 重复计数器中将重新装载 TIM_RCR 寄存器的内容
- 预分频器的缓冲区中将重新装载预装载值 (TIM_PSC 寄存器的内容)
- 自动重载活动寄存器将以预装载值 (TIM_ARR 寄存器的内容) 进行更新。注意，如果更新操作是由计数器上溢触发的，则自动重载寄存器在重载计数器之前更新，因此，下一个计数周期就是我们所希望的新的周期长度 (计数器被重载新的值)。

以下各图以一些示例说明不同时钟频率下计数器的行为。

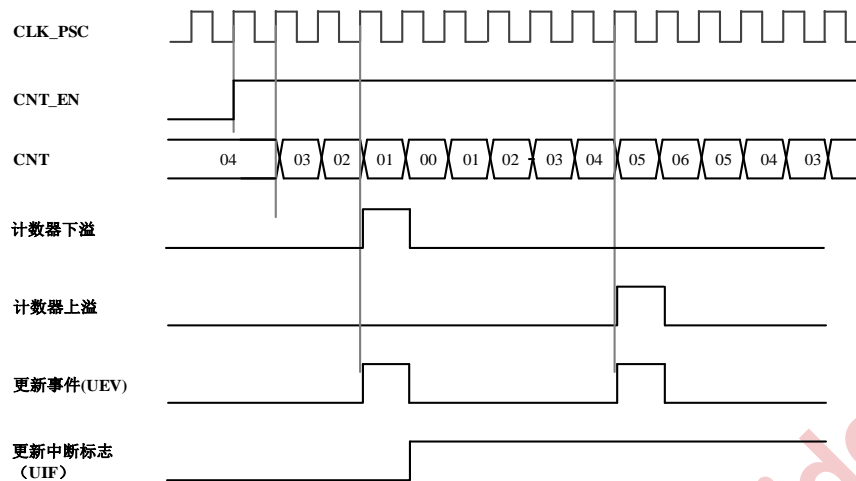


图 40 计数器时序图, 1 分频内部时钟, TIM_ARR = 0x6

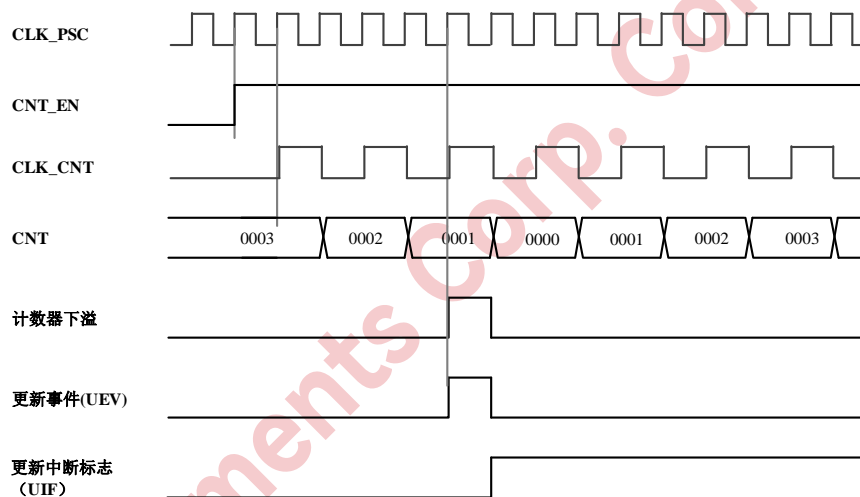


图 41 计数器时序图, 2 分频内部时钟

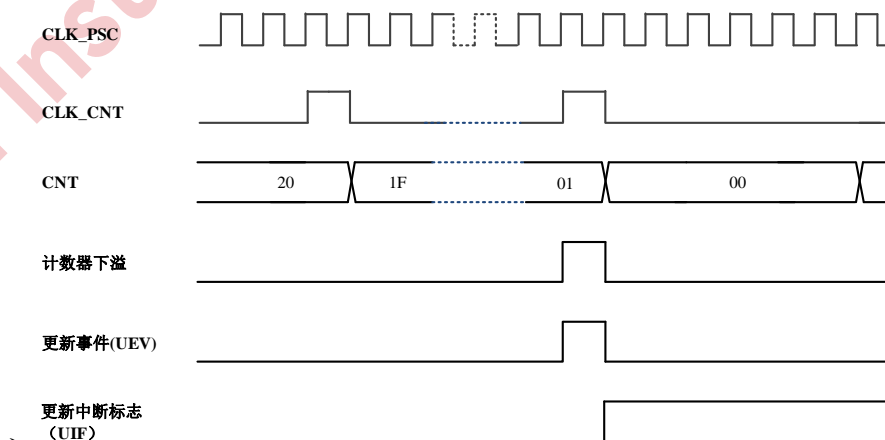


图 42 计数器时序图, N 分频内部时钟

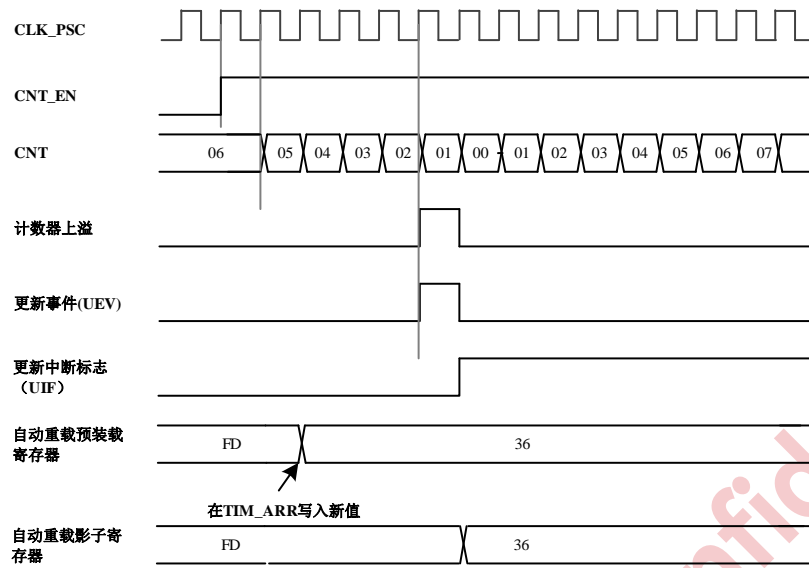


图 43 计数器时序图, ARPE=1 时的更新事件 (计数器下溢)

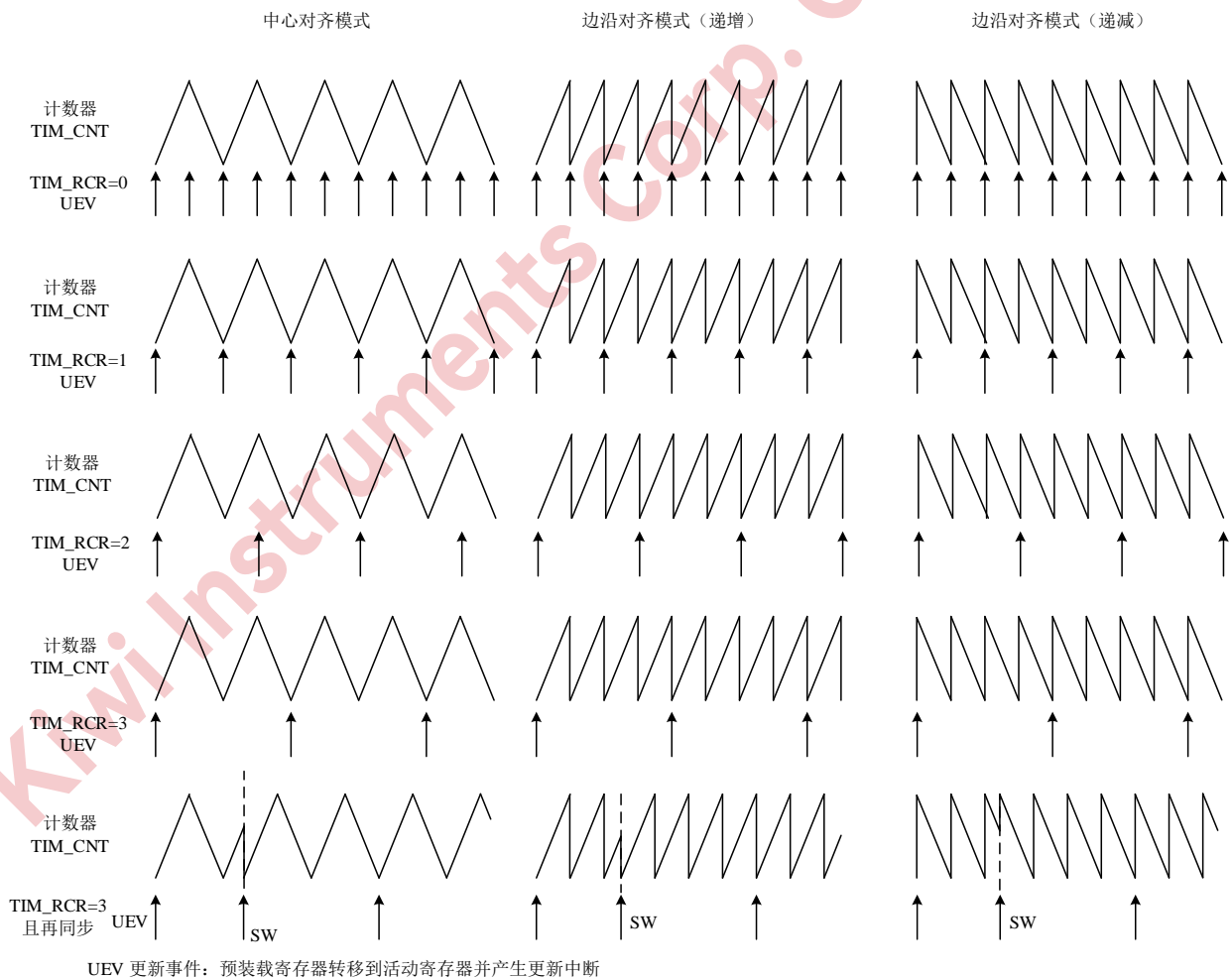


图 44 不同模式和 TIM_RCR 寄存器设置下的更新频率示例

13.4.3 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟 (TIM_CLK_INT)
- 外部时钟模式：外部输入引脚 (TIM_TiX)
- 内部触发输入 (TIM_ITR)：使用一个定时器作为另一个定时器的预分频器

13.4.3.1 内部时钟源

如果禁止从模式控制器 (TIM_CR 寄存器中 SMS=000)，则 CEN 位 (TIM_CR) 和 UG 位 (TIM_EGR) 为实际控制位，并且只能通过软件进行更改，但 UG 仍自动清零。当对 TIM_CEN 位写入 1 时，预分频器的时钟就由内部时钟 TIM_CLK_INT 提供。

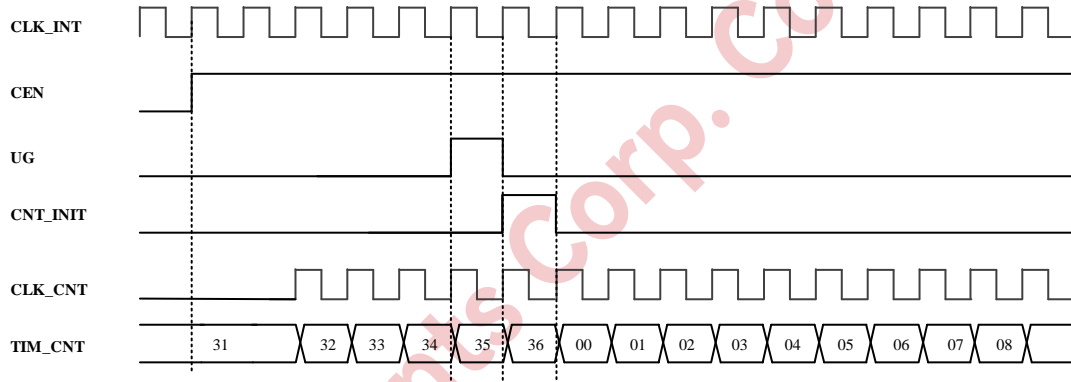


图 45 正常模式下的控制电路 1 分频内部时钟

13.4.3.2 外部时钟源模式

当 TIM_MCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

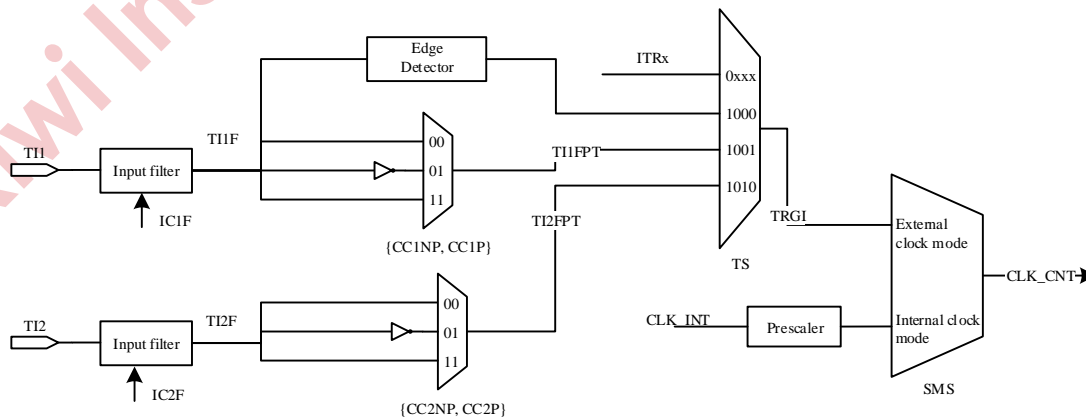


图 46 外部时钟输入连接关系

例如，要使计数器在 TI2 输入出现上升沿时计数，可执行以下步骤：

1. 通过在 TIM_CCMR 寄存器中写入 CC2S=01 来配置通道 2，使其能够检测 TI2 输入的上升沿。
2. 通过在 TIM_CCMR 寄存器中写入 IC2F[3:0] 位来配置输入滤波时间（如果不需要任何滤波，请保持 IC2F=0000）。
3. 通过在 TIM_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
4. 通过在 TIM_MCR 寄存器中写入 SMS=111，使定时器在外部时钟模式下工作。
5. 通过在 TIM_MCR 寄存器中写入 TS=110 来选择 TI2 作为输入源。
6. 通过在 TIM_CR 寄存器中写入 CEN=1 来使能计数器。

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

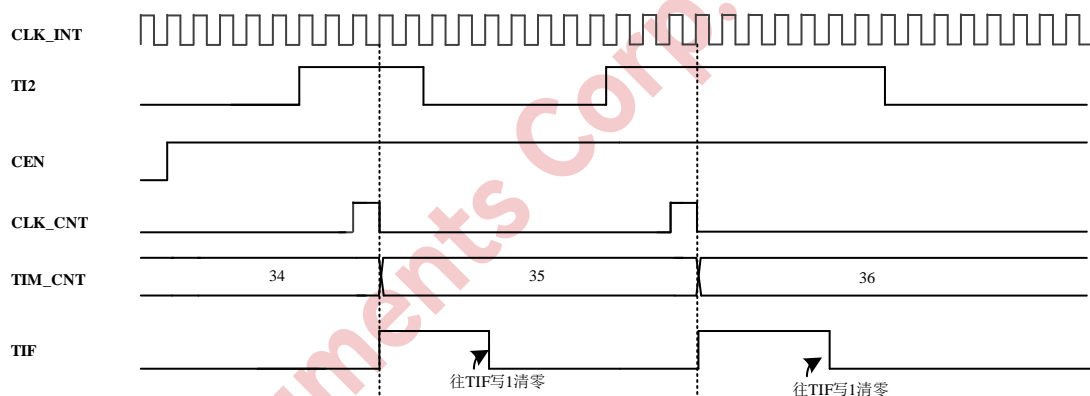


图 47 外部时钟模式下的控制电路

13.4.4 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

下图概括介绍了一个捕获/比较通道。输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

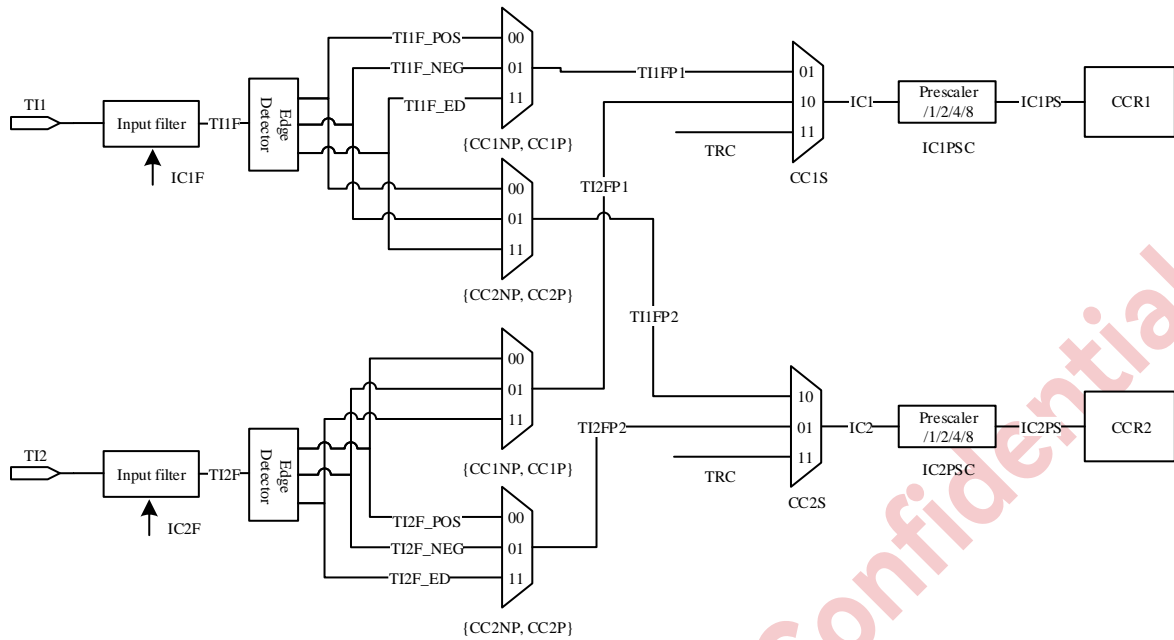


图 48 捕获/比较通道的输入阶段

输出阶段生成一个中间波形作为基准：OCxREF (高电平有效)。链的末端决定最终输出信号的极性。

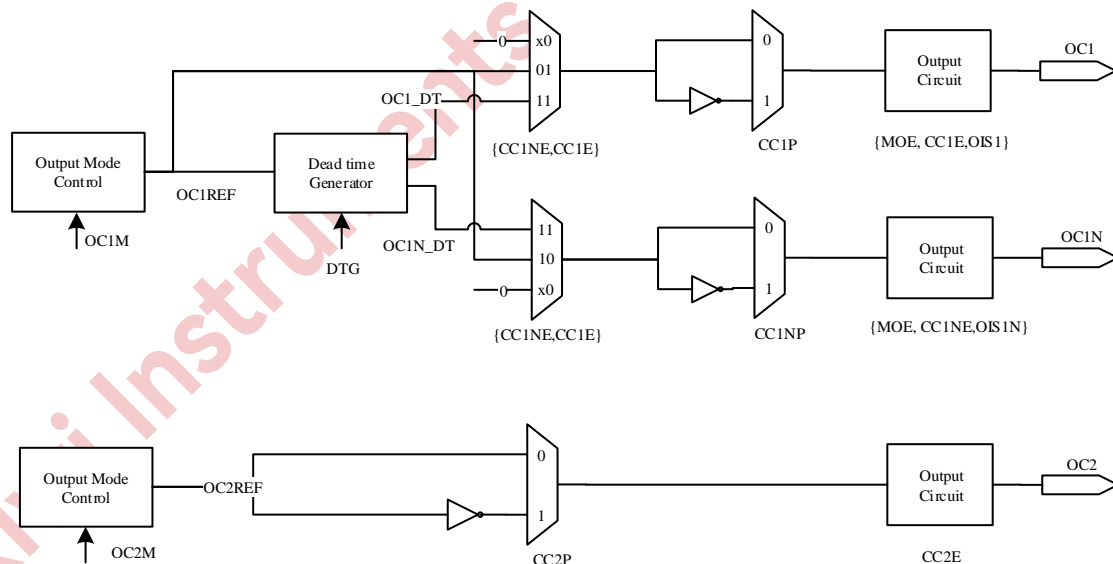


图 49 捕获/比较通道的输出阶段

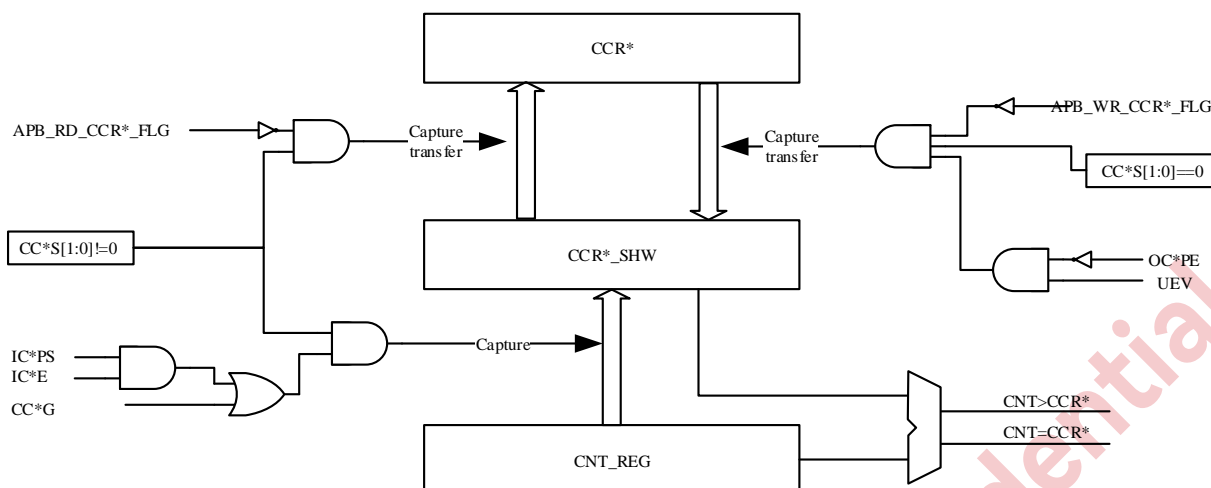


图 50 捕获/比较通道主要电路

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载寄存器。在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。在比较模式下，预装载寄存器的内容将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

13.4.5 输入捕获模式

在输入捕获模式下，当相应的 TIM_TIx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIM_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCxIF 标志 (TIM_SR) 置 1，并可发送中断请求 (如果已使能)。如果发生捕获事件时 CCxIF 标志已置位，则会将重复捕获标志 CCxOF (TIM_SR) 置 1。可通过软件向 CCxIF 写入 1 来给 CCxIF 清零，或读取存储在 TIM_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 1 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIM_CCR1 中。具体操作步骤如下：

1. 选择有效输入：TIM_CCR1 必须连接到 TI1 输入，因此向 TIM_CCMR 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIM_CCR1 寄存器将处于只读状态。
2. 根据连接到定时器的信号，对所需的输入滤波时间进行设置 (如果输入为 Tlx 输入之一，则对 TIM_CCMR 寄存器中的 ICxF 位进行设置)。输入的信号经过采样时钟进行采样，并根据连续 N 次的采样值保持一致方可判断为有效的电平输入，否则保持原来输入电平，其中 N 为 1 和 2 可选，采样频率可配置 (详见相关寄存器描述)。
3. 通过向 TIM_CCER 寄存器中的 CC1P 位和 CC1NP 位写入 0，选择 TI1 通道的有效转换边沿上升沿。
4. 对输入预分频器进行设置。本例中，倘若每次有效转换时都执行捕获操作，那需要禁止预分频器 (向 TIM_CCMR 寄存器中的 IC1PS 位写入 00)。

5. 通过将 TIM_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
6. 必要时可通过将 TIM_IER 寄存器中的 CC1IE 位置 1 来使能相关中断请求。
7. 当发生输入捕获时：
 - 发生有效跳变沿时，TIM_CCR1 寄存器会获取计数器的值。
 - 将 CC1IF 标志置 1 (中断标志)。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
 - 根据 CC1IE 位生成中断。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

注意：通过软件将 TIM_EGR 寄存器中的相应 CCxG 位置 1 可生成中断。

13.4.6 PWM 输入模式

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 TIM_TIx 信号被映射至同一个 ICx 输入。
- 这两个 TIM_TIx 信号在边沿处有效，但极性相反。
- 选择两个 TixFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIM_CCR1 寄存器中）和占空比（位于 TIM_CCR2 寄存器中）进行测量（取决于 TIM_CK_INT 频率和预分频器的值）

1. 选择 TIM_CCR1 的有效输入：向 TIM_CCMR1 寄存器中的 CC1S 位写入 01 (选择 TI1)。
2. 选择 TI1FP1 的有效极性（用于 TIM_CCR1 中的捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入 0 (上升沿有效)。
3. 选择 TIM_CCR2 的有效输入：向 TIM_CCMR 寄存器中的 CC2S 写入 10 (选择 TI1)。
4. 选择 TI1FP2 的有效极性（用于 TIM_CCR2 中的捕获）：向 CC2P 位和 CC2NP 位写入 1 (下降沿有效)。
5. 选择有效触发输入：向 TIM_MCR 寄存器中的 TS 位写入 1001 (选择 TI1FP1)。
6. 将从模式控制器配置为复位模式：向 TIM_MCR 寄存器中的 SMS 位写入 100。
7. 使能计数器开始计数 CEN=1。

8. 使能捕获：向 TIM_CCER 寄存器中的 CC1E 位和 CC2E 位分别写入 1。

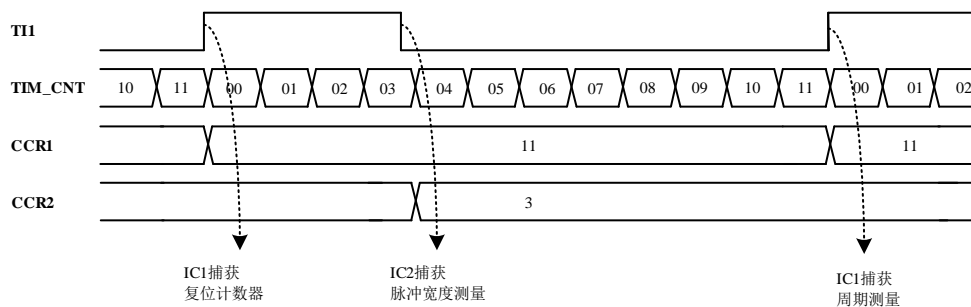


图 51 PWM 输入模式时序

13.4.7 强制输出模式

在输出模式 (TIM_CCMR 中的 CCxS = 00) 下，可直接由软件将每个输出比较信号 (OCxREF 和 OCx) 强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (OCxREF/OCx) 强制设置为有效电平，只需向相应 TIM_CCMR 寄存器中的 OCxM 位写入 101。OCxREF 进而强制设置为高电平 (OCxREF 始终为高电平有效)，同时 OCx 获取 CCxP 极性位的相反值。例如：CCxP=0 (OCx 高电平有效) => OCx 强制设置为高电平。通过向 TIM_CCMR 寄存器中的 OCxM 位写入 100，可将 OCxREF 信号强制设置为低电平。

无论如何，TIM_CCR 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断请求。输出比较模式一节对此进行了介绍。

13.4.8 输出比较模式

此功能用于控制输出波形，或指示已经过去某一段时间段计数器状态。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式 (TIM_CCMR 寄存器中的 OCxM 位) 和输出极性 (TIM_CCER 寄存器中的 CCxP 位) 定义。匹配时，输出引脚既可保持其电平 (OCxM=000)，也可设置为有效电平 (OCxM=001)、无效电平 (OCxM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1 (TIM_SR 寄存器中的 CCxIF 位)。
- 如果相应中断使能位 (TIM_IER 寄存器中的 CCxIE 位) 置 1，将生成中断。

使用 TIM_CCMR 寄存器中的 OCxPE 位，可将 TIM_CCRx 寄存器配置为带或不带预装载寄存器。

在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲 (在单脉冲模式下)。

步骤:

1. 选择计数器时钟 (内部、外部、预分频器)。
2. 在 TIM_ARR 和 TIM_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求, 将 CCxIE 位置 1。
4. 选择输出模式。例如, 当 CNT 与 CCRx 匹配、未使用预装载 CCRx 并且 OCx 使能且为高电平有效时, 必须写入 OCxM=011、OCxPE=0、CCxP=0 和 CCxE=1 来翻转 OCx 输出引脚。
5. 通过将 TIM_CR 中的 TIM_CEN 位置 1 来使能计数器。

可随时通过软件更新 TIM_CCRx 寄存器以控制输出波形, 前提是未使能预装载寄存器 (OCxPE=0, 否则仅当发生下一个更新事件 UEV 时, 才会更新 TIM_CCRx 影子寄存器)。下图列出了相关示例。

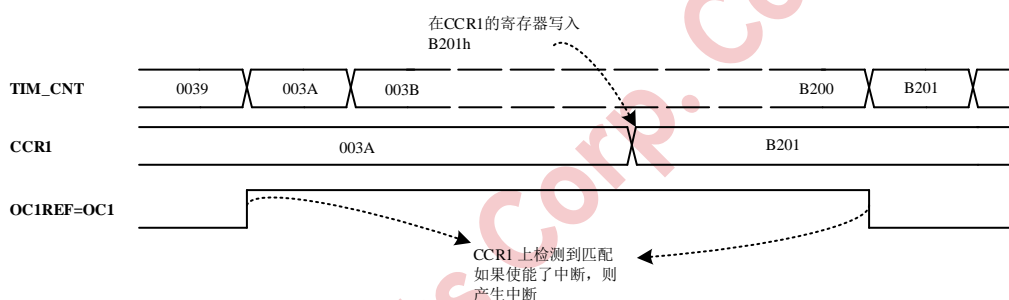


图 52 输出比较模式 (翻转 OC1)

13.4.9 PWM 模式

脉冲宽度调制模式可以生成一个信号, 该信号频率由 TIM_ARR 寄存器值决定, 其占空比则由 TIM_CCRx 寄存器值决定。

通过向 TIM_CCMR 寄存器中的 OCxM 位写入 110 (PWM 模式 1) 或 111 (PWM 模式 2), 可以独立选择各通道 (每个 OCx 输出对应一个 PWM) 的 PWM 模式。必须通过将 TIM_CCMR 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器, 最后通过将 TIM_CR 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器。

由于只有在发生更新事件时预装载寄存器才会传送到影子寄存器, 因此启动计数器之前, 必须通过将 TIM_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可使用 TIM_CCER 寄存器的 CCxP 位来编程。既可以设为高电平有效, 也可以设为低电平有效。OCx 输出通过将 TIM_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息, 请参见 TIM_CCER 寄存器说明。

在 PWM 模式 1 或 2 下, TIM_CNT 始终与 TIM_CCRx 进行比较, 以确定是 $TIM_CCR_x < TIM_CNT$ 还是 $TIM_CNT \leq TIM_CCR_x$ 。OCxREF 信号仅在以下情况下变为有效状态:

- 比较结果发生改变，或
- 输出比较模式 (TIM_CCMR 寄存器中的 OCxM 位) 从“冻结”配置 (OCxM=000) 切换为任一 PWM 模式 (OCxM=110 或 111)。

定时器运行期间，可以通过软件强制 PWM 输出。

我们以 PWM 模式 1 为例。只要 $TIM_CNT < TIM_CCR_x$ ，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIM_CCRx 中的比较值大于自动重载值 (TIM_ARR 中)，则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。下图举例介绍 PWM 模式 1 波形 (TIM_ARR=8)。

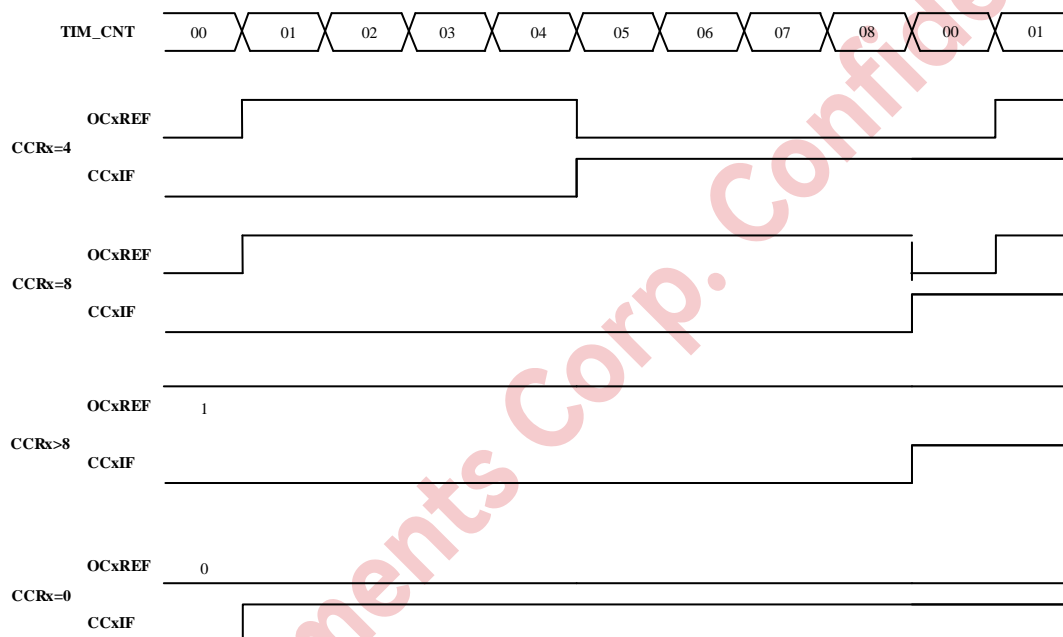


图 53 PWM 模式 1 波形 (ARR=8)

13.4.10 单脉冲模式

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

可通过从模式控制器启动计数器并在输出比较模式或 PWM 模式下生成波形。将 TIM_CR 寄存器中的 OPM 位置 1，即可选择单脉冲模式。在发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：递增计数模式下， $CNT < CCR_x \leq ARR$ (特别注意， $0 < CCR_x$)。

例如，用户希望达到这样的效果：在 TI2 输入引脚检测到正沿时，经过 tDELAY 的延迟，在 OC1 上产生一个长度为 tPULSE 的正脉冲。使用 TI2FP2 作为触发 1：

1. 在 TIM_CCMR 寄存器中写入 CC2S=01，将 TI2FP2 映射到 TI2。
2. 在 TIM_CCER 寄存器中写入 CC2P=0 和 CC2NP=0，使 TI2FP2 能够检测上升沿。
3. 在 TIM_MCR 寄存器中写入 TS=110，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
4. 在 TIM_MCR 寄存器中写入 SMS=110 (触发模式)，使用 TI2FP2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义 (考虑时钟频率和计数器预分频器)。

- t_{DELAY} 由写入 TIM_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值 (TIM_ARR-TIM_CCR1) 之差来定义。
- 欲生成如此波形：信号在发生比较匹配时从 0 变为 1，在计数器达到自动重载值时由 1 变为 0。为此，应在 TIM_CCMR 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果需要，可选择在 TIM_CCMR 寄存器的 OC1PE 和 TIM_CR 寄存器的 ARPE 中写入 1，以使能预装载寄存器。这种情况下，必须在 TIM_CCR1 寄存器中写入比较值并在 TIM_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为 0。

由于仅需要 1 个脉冲 (单脉冲模式)，因此应向 TIM_CR 寄存器的 OPM 位写入 1，以便在发生下一更新事件 (计数器从自动重载值返回到 0) 时使计数器停止计数。TIM_CR 寄存器中的 OPM 位设置为 0 时，即选择重复模式。

在单脉冲模式下，TIM_TIx 输入的边沿检测会将 TIM_CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟 (t_{DELAY} 最小值)。

如果要输出延迟时间最短的波形，可以将 TIM_CCMR 寄存器中的 OCxFE 位置 1。这样会强制 OCxREF (和 OCx) 对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM 模式 1 或 PWM 模式 2 时，OCxFE 才会起作用。

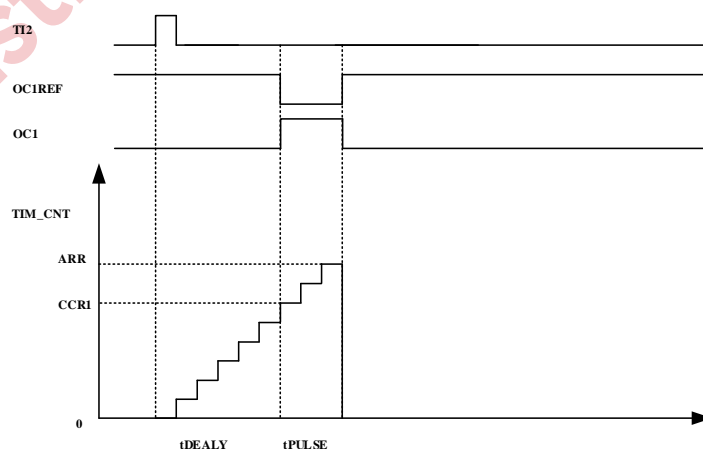


图 54 单脉冲模式示例

13.4.11 互补输出和死区插入

定时器可以输出两路互补信号，并管理输出的关断与接通瞬间。这段时间通常称为死区，用户必须根据与输出相连接的器件及其特性（电平转换器的固有延迟、开关器件产生的延迟...）来调整死区时间。

每路输出可以独立选择输出极性（主输出 OCx 或互补输出 OCxN）。可通过对 TIM_CCER 寄存器中的 CC1P 和 CC1NP 位执行写操作来完成极性选择。

互补信号 OC1 和 OC1N 通过以下多个控制位的组合进行激活：TIM_CCER 中的 CC1E 和 CC1NE 位以及 TIM_BDTR 和 TIMx_CR 中的 MOE、OIS1、OIS1N 位。更多详细信息，请参照下表：具有断路功能的互补通道 OC1 和 OC1N 的输出控制位。应当注意，切换至 IDLE (MOE 下降到 0) 的时刻，死区仍然有效。

CC1E 和 CC1NE 位同时置 1 并且 MOE 位置 1 (如果存在断路) 时，将使能死区插入。TIM_BDTR 寄存器中的 DTG[7:0] 位用于控制所有通道的死区生成。将基于参考波形 OC1REF 生成 2 个输出 OC1 和 OC1N。如果 OC1 和 OC1N 为高电平有效：

- 输出信号 OC1 与参考信号相同，只是其上升沿相对参考上升沿存在延迟。
- 输出信号 OC1N 与参考信号相反，并且其上升沿相对参考下降沿存在延迟。

如果延迟时间大于有效输出 (OC1 或 OC1N) 的宽度，则不会产生相应的脉冲。下图所示为死区发生器的输出信号与参考信号 OC1REF 之间的关系。(在这些示例中，假定 CC1P=0、CC1NP=0、MOE=1、CC1E=1 并且 CC1NE=1)

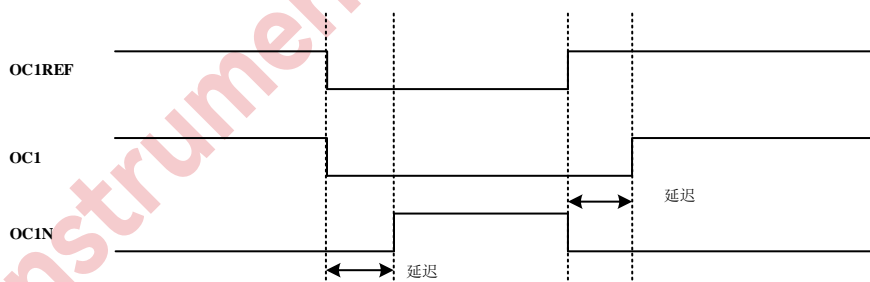


图 55 带死区插入的互补输出

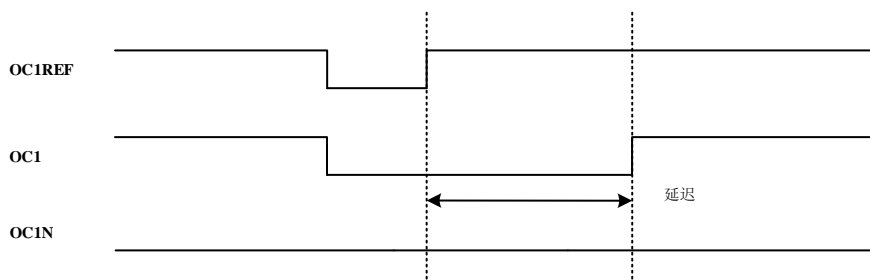


图 56 延迟时间大于负脉冲宽度的死区波形

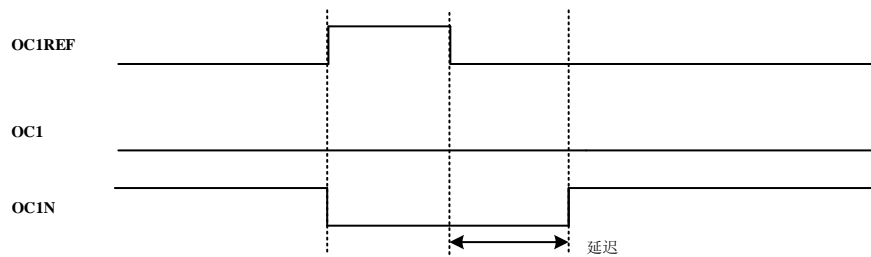


图 57 延迟时间大于正脉冲宽度的死区波形

死区延迟对于所有通道均相同，可通过 TIM_BDTR 寄存器中的 DTG 位进行编程。有关延迟时间计算的信息，请参见断路和死区寄存器 (TIM_BDTR) 描述。

将 OC1REF 重定向到 OC1 或 OC1N

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIM_CCER 寄存器中的 CC1E 和 CC1NE 位，可将 OC1REF 重定向到 OC1 输出或 OC1N 输出。

通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者，使两个输出同时保持无效电平，或者两个输出同时处于有效电平，两者互补并且带死区。

如果仅使能 OC1N (CC1E=0, CC1NE=1)，两者不互补，一旦 OC1REF 为高电平，OC1N 即变为有效。例如，如果 CC1NP=0，则 OC1N=OC1REF。另一方面，如果同时使能 OC1 和 OC1N (CC1E=CC1NE=1)，OC1 在 OC1REF 为高电平时变为有效，而 OC1N 则与之互补，在 OC1REF 为低电平时变为有效。

13.4.12 生成 6 步 PWM

当通道使用互补输出时，OCxM、CCxE 和 CCxNE 位上提供预装载位。发生 COM 换向事件时，这些预装载位将传输到影子位。因此，用户可以预先编程下一步骤的配置，并同时更改所有通道的配置。COM 可由软件通过将 TIM_EGR 寄存器中的 COM 位置 1 而生成，可以由硬件在 TRGI 上升沿生成，而且也可以由其他定时器 COM 置位生成。

发生 COM 事件时，某个标志位 (TIM_SR 寄存器中的 COMIF 位) 将会置 1。这时，如果 TIM_IER 寄存器中的 COMIE 位置 1，将产生中断；

下图以 3 种不同的编程配置为例，显示了发生 COM 事件时 OCx 和 OCxN 输出的行为。

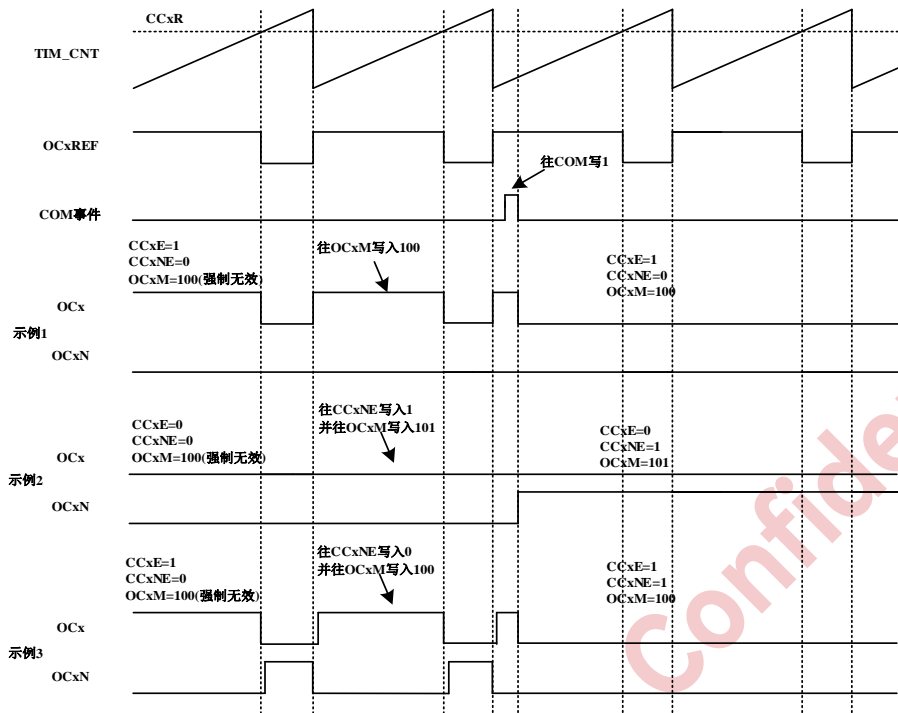


图 58 COM 事件生成 6 步 PWM 的示例

13.4.13 使用断路功能

使用断路功能时，根据相关控制位（TIM_BDTR 中的 MOE 位以及 TIM_CR 中的 OISx、OISxN、HIZx 和 HIZxN 位）修改输出使能信号、无效电平或者高阻输出。任何情况下，OCx 和 OCxN 输出都不能同时置为有效电平。更多详细信息，请参见表格：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位。

断路源可以是断路输入引脚，也可以是时钟故障事件。

退出复位状态后，断路功能处于禁止状态，MOE 位处于低电平。将 TIM_BDTR 寄存器中的 BKE 位置 1，可使得能断路功能。

由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 TIM_BDTR 寄存器中）之间插入了再同步电路，从而在异步信号与同步信号之间产生延迟。具体而言，如果在 MOE 处于低电平时向其写入 1，则必须首先插入延迟（空指令），才能准确进行读取。这是因为写入的是异步信号，而读取的却是同步信号。

发生断路（断路输入上出现所选电平）时：

- MOE 位异步清零，使输出处于无效状态、空闲状态或复位状态。即使定时器时钟关闭，该功能仍然有效。
- MOE=0 时，将以 TIM_CR 寄存器 OISx 和 OISxN 位中编程的电平驱动每个输出通道。
- MOE=0 时，将以 TIM_CR 寄存器 HIZx 和 HIZxN 位中编程的电平控制每个输出通道使能已达到高阻输出的目标。
- 将断路状态标志 (TIM_SR 寄存器中的 BIF 位) 置 1。如果 TIM_IER 寄存器中的 BIE 位置 1，可产生中断。
- 如果 TIM_BDTR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件 (UEV) 时自动再次置 1。这一特性有许多用处，比如，可用于实现调节器的功能。否则，MOE 将始终保持低电平，直到再次向该位写入 1。这种情况下，这一特性可用于确保安全。可以将断路输入连接到功率驱动器的警报、温度传感器或任何安全元件。

断路输入为电平有效。因此，当断路输入有效电平时，不能将 MOE 位置 1 (自动或通过软件)。同时，不能将状态标志 BIF 清零。

断路可由 BRK 输入生成，该输入具有可编程极性，其使能位 BKE 位于 TIM_BDTR 寄存器中。断路有以下两种生成方案：

- 使用 BRK 输入生成，该输入具有可编程极性，其使能位 BKE 位于 TIM_BDTR 中。
- 由软件通过 TIM_EGR 中的 BG 位生成。

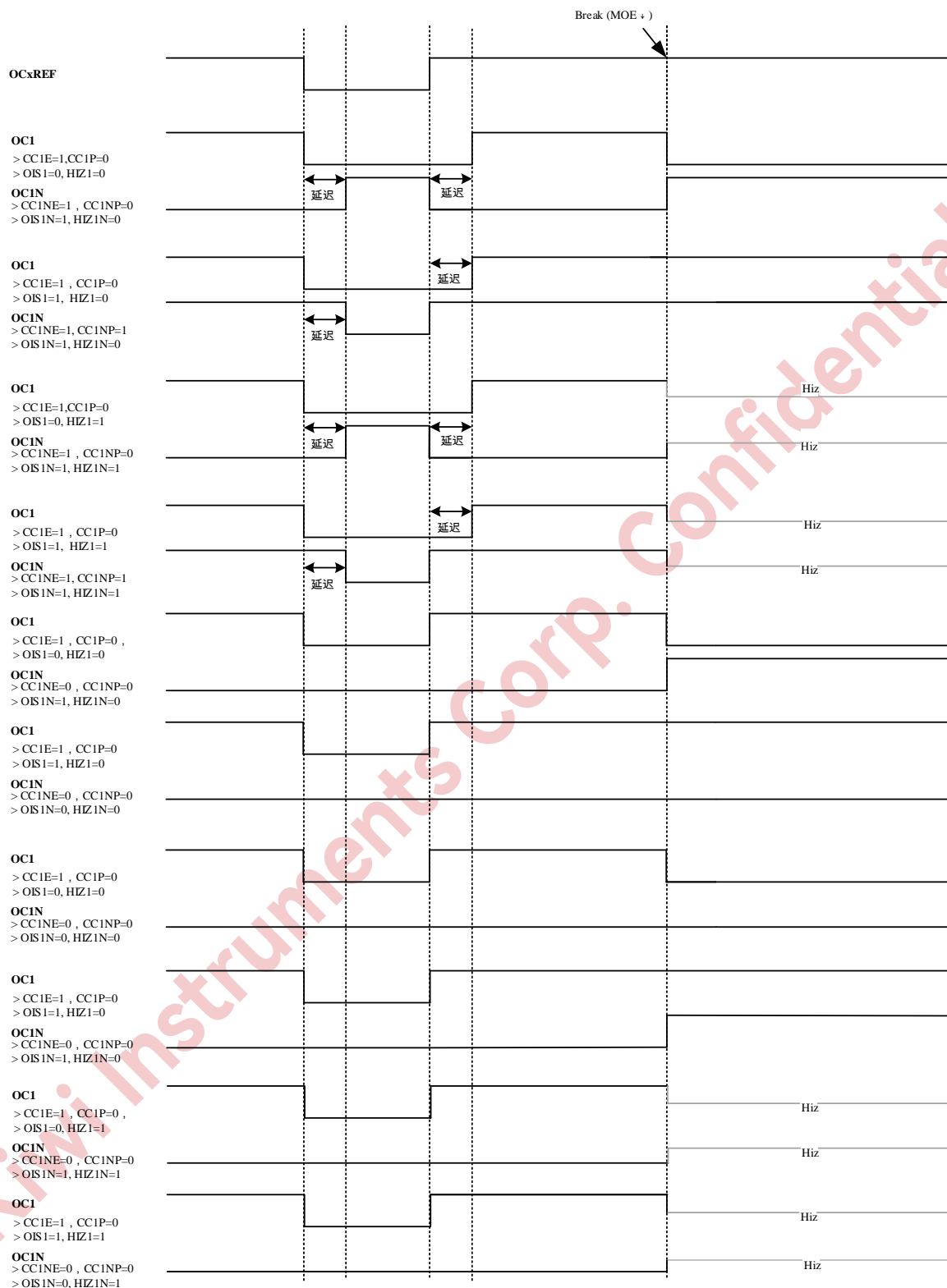


图 59 输出的断路响应行为 (通道 1)

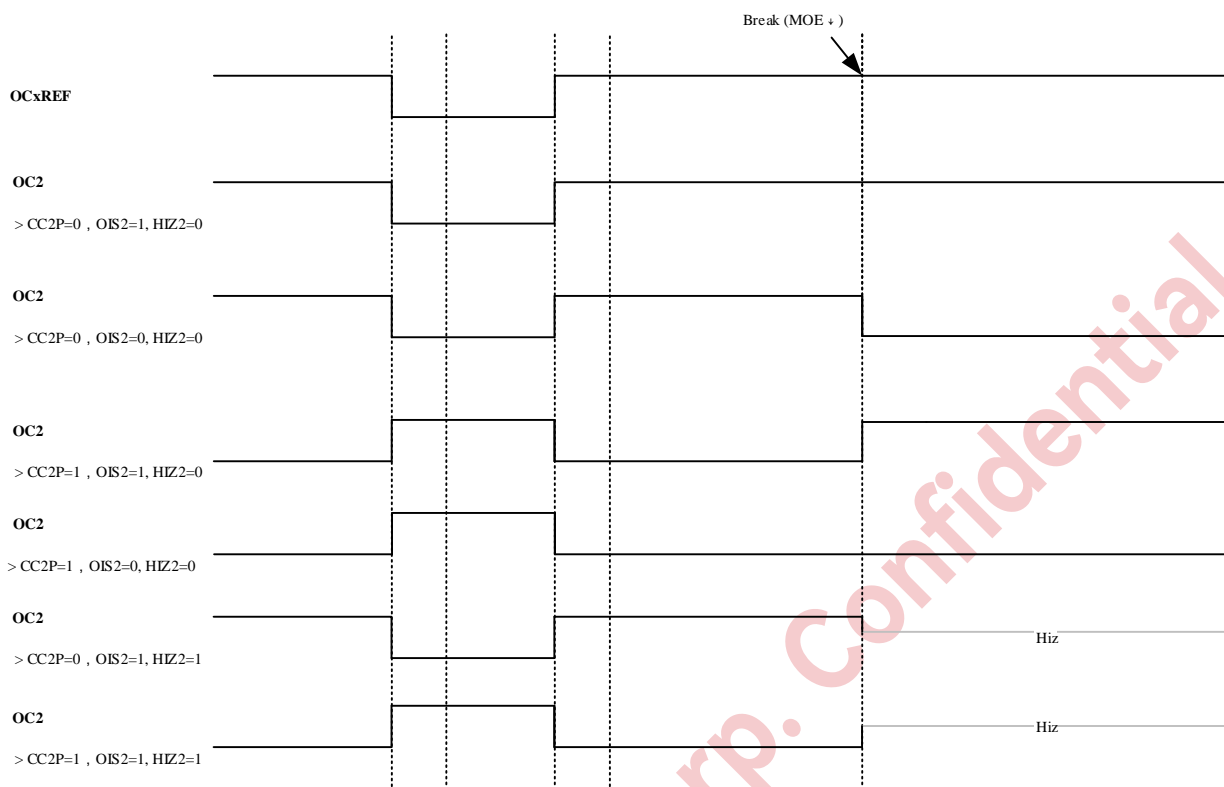


图 60 输出的断路响应行为 (通道 2)

断路输入的极性可通过该寄存器中的 BKP 位来选择。BKE 和 BKP 位可同时修改。对 BKE 和 BKP 位执行写操作时，写操作会在 1 个 APB 时钟周期的延迟后生效。因此，执行写操作后，需要等待 1 个 APB 时钟周期，才能准确回读该位。

定时器支持这几种断路源：外部输入断路、EBUS 事件、SRAM 奇偶校验错误和时钟丢失事件。每一类断路源都有独立的开关控制、8 路外部输入断路支持极性选择，而其他的断路源仅支持高电平有效。

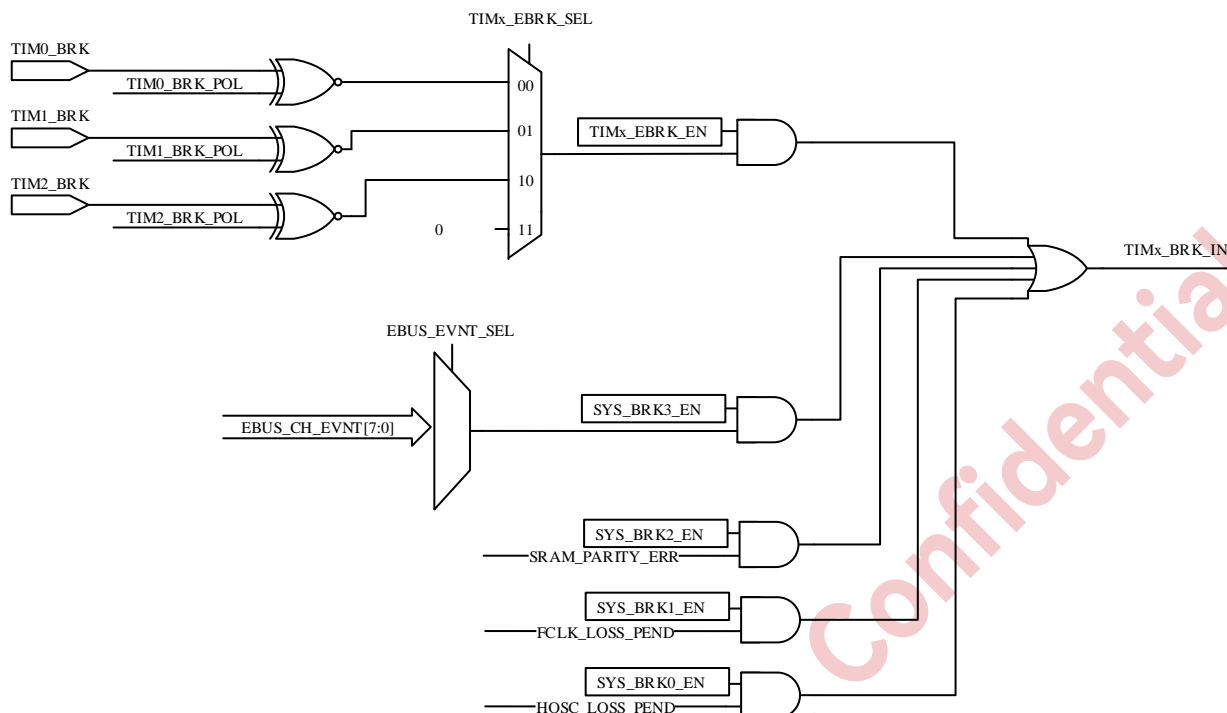


图 61 输出的断路响应行为 (通道 2)

13.4.14 定时器与外部触发同步

定时器可与外部触发以下列模式实现同步：复位模式、门控模式和触发模式。

13.4.14.1 复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIM_CR 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器 (TIM_ARR 和 TIM_CCRx) 都将更新。

在下例中，T11 输入上出现上升沿时，计数器清零：

1. 将通道 1 配置为检测 TIM1 的上升沿。配置输入滤波时间（本例中只需要默认采样滤波，故保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIM_CCMR 寄存器中的 CC1S=01。在 TIM_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，以验证极性（仅检测上升沿）。
2. 在 TIM_MCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIM_MCR 寄存器中写入 TS=1000，选择 TIM1 作为输入源。
3. 在 TIM_CR 寄存器中写入 TIM_CEN=1，启动计数器。

计数器使用内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时触发标志 (TIM_SR 寄存器中的 TIF 位) 置 1，使能中断后，还可发送中断请求 (如果 TIE 使能)。

下图显示了自动重载寄存器 TIM_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

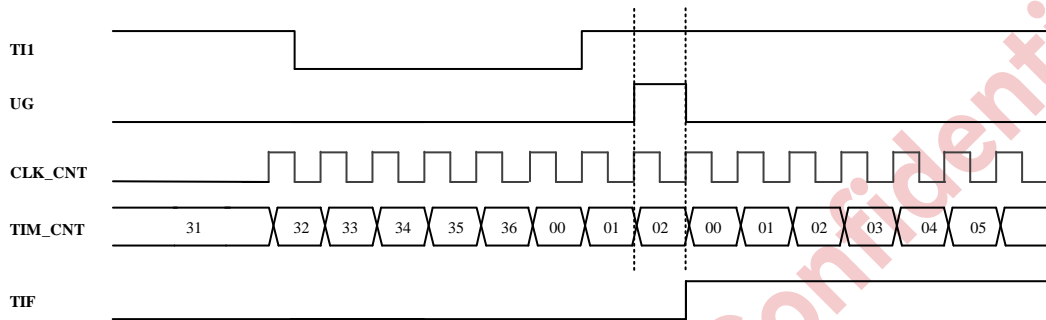


图 62 复位模式下的控制电路

13.4.14.2 门控模式

输入信号的电平可用来使能计数器。在以下示例中，计数器仅在 TI1 输入为低电平时计数：

1. 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间 (本例中不需要任何滤波, 因此保持 IC1F=0000)。由于捕获预分频器不用于触发操作, 因此无需对其进行配置。CC1S 位只选择输入捕获源, 即 TIM_CCMR 寄存器中的 CC1S=01。在 TIM_CCER 寄存器中写入 CC1P=1, 以确定极性 (仅检测低电平)。
2. 在 TIM_MCR 寄存器中写入 SMS=101, 将定时器配置为门控模式。在 TIM_MCR 寄存器中写入 TS=1000, 选择 TI1 作为输入源。
3. 在 TIM_CR 寄存器中写入 TIM_CEN=1, 使能计数器 (在门控模式下, 如果 CEN=0, 则无论触发输入电平如何, 计数器都不启动)。

只要 TI1 为低电平, 计数器就开始根据内部时钟计数, 直到 TI1 变为高电平时停止计数。计数器启动或停止时, TIM_SR 寄存器中的 TIF 标志都会置 1。TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

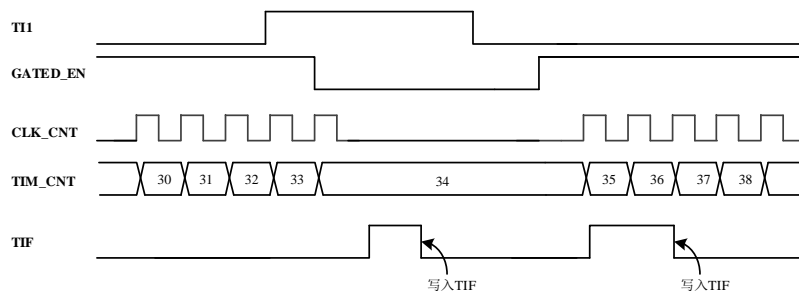


图 63 门控模式下的控制电路

13.4.14.3 触发模式

所选输入上发生某一事件时可以用以启动计数器。以下示例中，TI2 输入上出现上升沿时，计数器便启动：

1. 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIM_CCMR 寄存器中的 CC2S=01。在 TIM_CCER 中写入 CC2P=1，以确定极性（仅检测低电平）。
2. 在 TIM_MCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIM_MCR 寄存器中写入 TS=1010，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

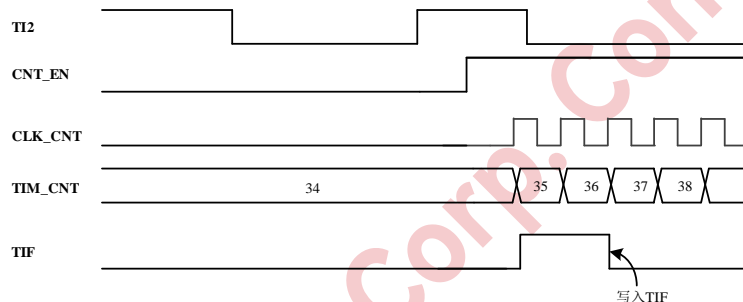


图 64 触发模式下的控制电路

13.4.15 定时器协同操作

13.4.15.1 使能

多个 TIMS 定时器可通过编程支持同时使能和关闭，从而实现同步的多通道操作。一旦使能了 CEC 控制位 (TIM_CR)，该定时器的 CEN 置位和清零可以通过其他定时器 (TIMS) 对 CEN 的操作完全同步实现，而且对该定时器的 CEN 操作也会同样可以同步操作其他定时器的 CEN。也就是说，使能了 CEC 控制位的所有定时器的 CEN 操作完全只需要操作其中一个定时的 CEN 便可。

该项功能可用于严格同步的多通道 PWM 拓展。

13.4.15.2 换向操作

多个 TIMS 定时器可通过编程支持同时软件触发换向事件，从而实现同步的多通道操作。一旦使能了 CCE 控制位 (TIM_CR)，该定时器的软件换向事件 COM 可以通过其他定时器 (TIMS) 软件触发换向事件生成，而且该定时器的软件触发换向事件同样可以作为其他定时器换向事件源。也就是说，使能了 CCE 控制位的所有定时器的软件触发换向操作完全只需要操作其中一个定时的 COM 便可。

该项功能可用于严格同步的多通道 BLDC 控制。

13.4.16 定时器同步

定时器从内部连接在一起，以实现定时器同步或级联。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

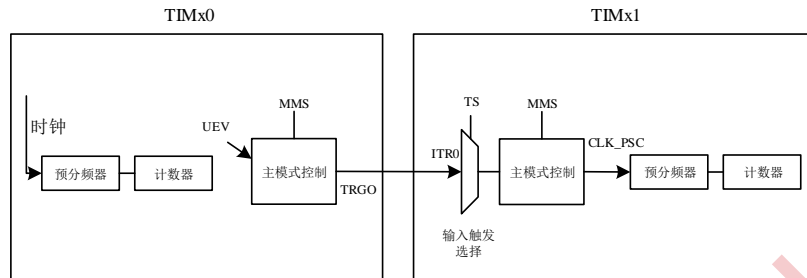


图 65 主/从定时器示例

将一个定时器用作另一个定时器的预分频器

例如，可以将定时器 0 配置为定时器 1 的预分频器。为此：

1. 将定时器 0 配置为主模式，以便每次发生更新事件 UEV 时都输出一个周期性触发信号。在 TIM1_CR 寄存器中写入 MMS=010，则每次生成更新事件时，TRGO 都会输出一个上升沿。
2. 要将定时器 0 的 TRGO 输出连接到定时器 1，必须将定时器 1 配置为从模式，使用 ITR0 作为内部触发。可通过设置定时器 1 的 TIM_SMCR 寄存器中的 TS 位（写入 TS=0000）可对此进行选择。
3. 然后将从模式控制器设为外部时钟模式 0（在定时器 1 的 TIM_MCR 寄存器中写入 SMS=111）。这样一来，定时器 1 的时钟将由定时器 0 周期性触发信号的上升沿（与定时器 0 的计数器上溢对应）提供。
4. 最后必须通过将这两个定时器的相应 CEN 位（TIM_CR 寄存器）置 1 同时使能二者。

使用一个定时器使能另一个定时器

本例中通过定时器 0 的输出比较 1 来使能定时器 1。相关连接图，请参见主/从定时器示例图。仅当定时器 0 的 OC1REF 为高电平时，定时器 1 才根据分频后的内部时钟进行计数。两个计数器的时钟频率都基于 TIM_CK_INT 通过预分频器执行 2 分频 ($f_{CK_CNT} = f_{CK_INT}/2$)。

- 将定时器 0 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出 (TIM0_CR 寄存器中的 MMS=100)；
- 配置定时器 0 的 OC1REF 波形 (TIM0_CCMR 寄存器)；
- 配置定时器 1 以接收来自定时器 0 的输入触发 (TIM1_MCR 寄存器中的 TS=0000)；
- 将定时器 1 配置为门控模式 (TIM1_MCR 寄存器中的 SMS=101)；
- 使能定时器 1；

- 使能定时器 0。

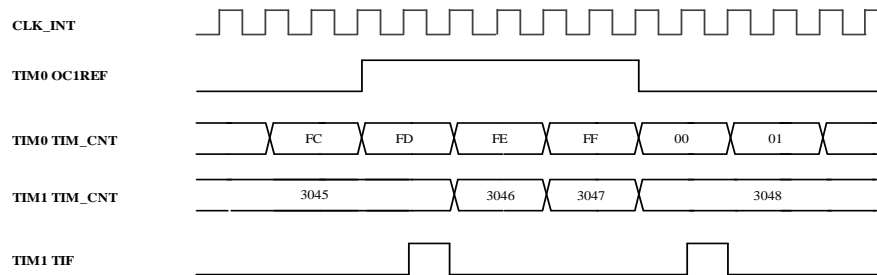


图 66 使用定时器 0 的 OC1REF 对定时器 1 实施门控控制

在上图的示例中，定时器 1 的计数器和预分频器在启动前未进行初始化。因此从各自的当前值开始计数。启动定时器 0 之前，通过复位这两个定时器可以从指定值开始计数。这样便可以在定时器计数器中写入所需的任意值。两个定时器都可通过软件使用 TIM_EGR 寄存器中的 UG 位轻松复位。

在下一示例中，定时器 0 与定时器 1 同步。定时器 0 为主模式，从 0 开始计数。定时器 2 为从模式，从 0xE7 开始计数。两个定时器的预分频比相同。在 TIM0_CR1 寄存器中通过向 TIM_CEN 位写入“0”来禁止定时器 0 时，定时器 1 将停止：

1. 将定时器 0 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出 (TIM1_CR 寄存器中的 MMS=100)。
2. 配置定时器 0 的 OC1REF 波形 (TIM0_CCMR 寄存器)。
3. 配置定时器 1 以接收来自定时器 0 的输入触发 (TIM1_MCR 寄存器中的 TS=0000)。
4. 将定时器 1 配置为门控模式 (TIM1_MCR 寄存器中的 SMS=101)。
5. 通过向 UG 位 (TIM0_EGR) 写入“1”复位定时器 0。
6. 通过向 UG 位 (TIM1_EGR 寄存器) 写入“1”复位定时器 1。
7. 通过在定时器 1 的计数器 (TIM1_CNT) 中写入“0xE7”使定时器 1 初始化为 0xE7。
8. 通过向 CEN 位 (TIM1_CR) 使能定时器 2。
9. 通过向 CEN 位 (TIM0_CR) 启动定时器 0。
10. 通过向 CEN 位 (TIM0_CR) 停止定时器 0。

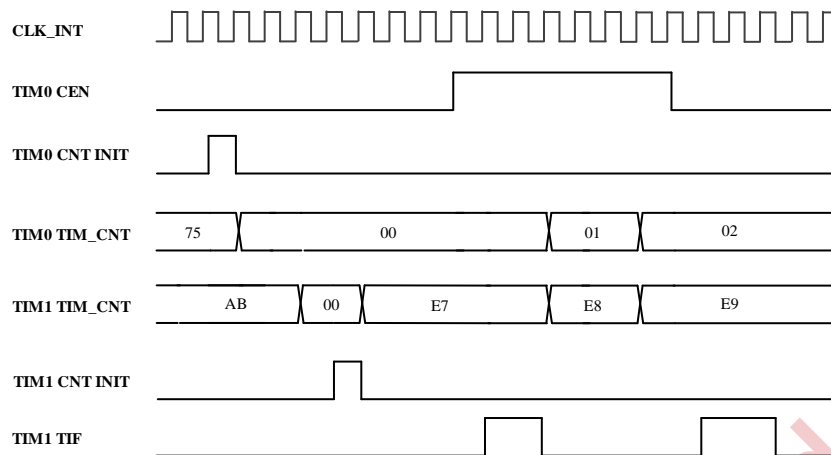


图 67 使用定时器 0 的使能信号对定时器 1 实施门控控制

使用一个定时器启动另一个定时器

本例中使用定时器 1 的更新事件使能定时器 2。相关连接图，请参见主/从定时器示例图。只要定时器 1 生成更新事件，定时器 2 便根据分频后的内部时钟从当前值（可以不为 0）开始计数。定时器 2 收到触发信号时，其 TIM_CEN 位自动置 1，并且计数器开始计数，直到向 TIM2_CR 寄存器的 TIM_CEN 位写入“0”后停止计数。两个计数器的时钟频率都基于 TIM_CLK_INT 通过预分频器执行 4 分频 ($f_{CK_CNT} = f_{CK_INT}/4$)。

1. 将定时器 1 配置为主模式，发送其更新事件 (UEV) 作为触发输出 (TIM1_CR 中的 MMS=010);
2. 配置定时器 1 的周期 (TIM1_ARR 寄存器);
3. 配置定时器 2 以接收来自定时器 1 的输入触发 (TIM2_MCR 寄存器中的 TS=0000);
4. 将定时器 2 配置为触发模式 (TIM2_MCR 中的 SMS=110);
5. 启动定时器 1。

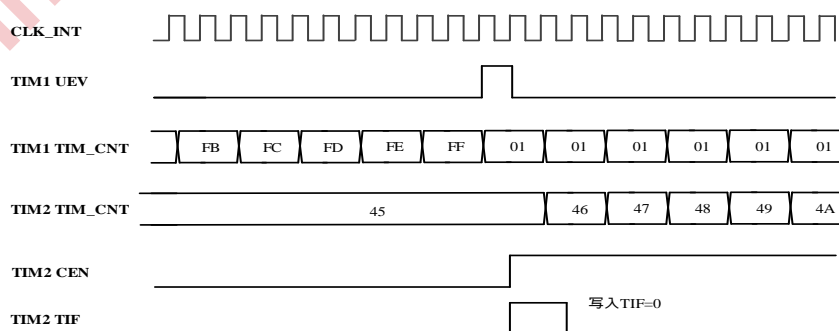


图 68 使用定时器 1 的更新事件触发定时器 2

如上述示例所示，用户可以在开始计数之前初始化两个计数器。

13.4.17 TIMS 中断号

所有 TIM 的中断连接至不可屏蔽中断处理模块 (NVIC) 的 IRQ 相关接口。为了启用该中断功能，需要按照以下流程进行操作：

- 使能 TIM 的中断使能位
- 使能 NVIC 中的 TIM IRQ 通道, TIM 的系统中断号如下表所示.
- 配置 TIM 以产生中断

表 169 TIMS 中断表

名称	中断号
TIMS0_INT	38
TIMS1_INT	39
TIMS2_INT	40
TIMS3_INT	41

13.5 寄存器

TIMS 寄存器基地址安排如下表所示。

表 170 基地址对应表

定时器	基地址地址
TIMS0	0x40042000
TIMS1	0x40042400
TIMS2	0x40042800
TIMS3	0x40042C00

下表为 TIMSx 的各控制寄存器描述。

表 171 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	TIM 控制寄存器	32	r/w	0x00000000
0x04	TIM 模式控制寄存器	32	r/w	0x00000000
0x08	TIM 捕获/比较模式寄存器	32	r/w	0x00000000
0x0c~0x1c	Reserved	32	res	0x00000000
0x20	TIM 捕获/比较使能寄存器	32	r/w	0x00000000
0x24~0x5c	Reserved	32	res	0x00000000

偏移地址	寄存器列表	位宽	访问	复位值
0x60	TIM 计数器	32	r/w	0x00000000
0x64	TIM 预分频器	32	r/w	0x00000000
0x68	TIM 自动重载寄存器	32	r/w	0x0000FFFF
0x6C	Reserved	32	res	0x00000000
0x70	TIM 捕获/比较寄存器 1	32	r/w	0x00000000
0x74	TIM 捕获/比较寄存器 2	32	r/w	0x00000000
0x78~0x8C	Reserved	32	res	0x00000000
0x90	TIM 断路及死区寄存器	32	r/w	0x00000000
0xB0	TIM 事件生成寄存器	32	r/w	0x00000000
0xC0	TIM 中断使能寄存器	32	r/w	0x00000000
0xC4	TIM 状态寄存器	32	r/w	0x00000000
0xC8~0xFC	Reserved	32	res	0x00000000

13.5.1 TIM 控制寄存器 (TIM_CR)

偏移地址: 0x0000

表 172 TIM 控制寄存器

位	访问	描述
31:27	Res	Reserved 复位值: 0x0
26	r/w	CCE: 协同换向使能 0: 关闭 1: 如果捕获/比较控制位 (CCPC=1) 进行预装载, 可通过其他 TIMS 的 COMG 的脉冲对这些位进行更新。 复位值: 0x0
25	r/w	TCE: 触发换向使能 0: 关闭 1: 如果捕获/比较控制位 (CCPC=1) 进行预装载, 可通过 TRGI 的上升沿对这些位进行更新。 复位值: 0x0
24	r/w	CCPC: 捕获/比较预装载控制 0: CCxE、CCxNE 和 OCxM 位未进行预装载 1: CCxE、CCxNE 和 OCxM 位进行了预装载, 写入这些位后, 仅当发生换向事件 (COM) 时才会对这些位进行更新。 换向事件包括以下三种: <ul style="list-style-type: none"> COMG 位置 1 TRGI 上检测到上升沿, 使能位为 TCS

位	访问	描述
		<ul style="list-style-type: none"> 检测到来自于其他 TIMS 的 COMG 脉冲, 使能位为 CCE 注意: 此位仅对具有互补输出的通道有效。 复位值: 0x0
23	Res	Reserved 复位值: 0x0
22	r/w	HIZ2: 高阻输出使能 2 (OC2 输出) 参照 HIZ1 定义 复位值: 0x0
21	r/w	HIZ1N: 高阻输出使能 1 (OC1N 输出) 0: 当 MOE=0 时, OC1N 依据 OIS10N 输出 1: 当 MOE=0 时, OC1N 输出高阻 注: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1、2 或 3, 此位即无法修改。 复位值: 0x0
20	r/w	HIZ1: 高阻输出使能 1 (OC1 输出) 0: 当 MOE=0 时, OC1 依据 OIS1 输出 1: 当 MOE=0 时, OC1 输出高阻 注: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1、2 或 3, 此位即无法修改。 复位值: 0x0
19	Res	Reserved 复位值: 0x0
18	r/w	OIS2: 输出空闲状态 2 (OC2 输出) 0: 当 MOE=0 和 HIZ2=0 时, OC2=0 1: 当 MOE=0 和 HIZ2=0 时, OC2=1 注: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1、2 或 3, 此位即无法修改。 复位值: 0x0
17	r/w	OIS1N: 输出空闲状态 1 (OC1N 输出) 0: 当 MOE=0 和 HIZ1N=0 时, OC1N=0 1: 当 MOE=0 和 HIZ1N=0 时, OC1N=1 注: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1、2 或 3, 此位即无法修改。 复位值: 0x0
16	r/w	OIS1: 输出空闲状态 1 (OC1 输出) 0: 当 MOE=0 和 HIZ1=0 时, OC1=0 1: 当 MOE=0 和 HIZ1=0 时, OC1=1 注: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1、2 或 3, 此位即无法修改。 复位值: 0x0

位	访问	描述
15:13	Res	Reserved 复位值: 0x0
12	r/w	CEC: 计数器协作使能控制 0: 关闭协作使能功能 1: 其他 TIMS 的 CEN 置位触发信号也会触发该 TIMS 的 CEN 置位 复位值: 0x0
11:10	Res	Reserved 复位值: 0x0
9:8	r/w	CKD: 时钟分频 此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器 (Tl _x) 所使用的死区及采样时钟 (t _{DTS}) 之间的分频比。 00: t _{DTS} = t _{CK_INT} 01: t _{DTS} = 2*t _{CK_INT} 10: t _{DTS} = 4*t _{CK_INT} 11: t _{DTS} = t _{CK_INT} 复位值: 0x0
7	r/w	ARPE: 自动重载预装载使能 0: TIM_ARR 寄存器不进行缓存 1: TIM_ARR 寄存器进行缓存 复位值: 0x0
6:5	r/w	CMS[1:0]: 中心对齐模式选择 00: 边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。 01: 中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时, 配置为输出的通道 (TIM_CCMR 寄存器中的 CCxS=00) 的输出比较中断标志才置 1。 10: 中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时, 配置为输出的通道 (TIM_CCMR 寄存器中的 CCxS=00) 的输出比较中断标志才置 1。 11: 中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时, 配置为输出的通道 (TIM_CCMR 寄存器中的 CCxS=00) 的输出比较中断标志都会置 1。 注: 只要计数器处于使能状态 (CEN=1), 就不得从边沿对齐模式切换为中心 复位值: 0x0
4	r/w	DIR: 方向 (Direction) 0: 计数器递增计数 1: 计数器递减计数 注: 当定时器配置为中心对齐模式或编码器模式时, 该位为只读状态。 复位值: 0x0
3	r/w	OPM: 单脉冲模式使能控制 0: 计数器在发生更新事件时不会停止计数

位	访问	描述
		1: 计数器在发生下一更新事件时停止计数 (将 CEN 位清零) 复位值: 0x0
2	r/w	URS: 更新请求源选择 0: 使能时, 所有以下事件都会生成更新中断请求。此类事件包括: <ul style="list-style-type: none"> 计数器上溢/下溢 将 UG 位置 1 通过从模式控制器生成的更新事件 1: 使能时, 只有计数器上溢/下溢会生成更新中断请求。 复位值: 0x0
1	r/w	UDIS: 更新禁止 此位由软件置 1 和清零, 用以使能/禁止 UEV 事件生成。 0: 使能 UEV。更新 (UEV) 事件可通过以下事件之一生成: <ul style="list-style-type: none"> 计数器上溢/下溢 将 UG 位置 1 通过从模式控制器生成的更新事件 然后更新影子寄存器的值。 1: 禁止 UEV。不生成更新事件, 各影子寄存器的值 (ARR、PSC 和 CCRx) 保持不变。但如果将 UG 位置 1, 或者从从模式控制器接收到硬件复位, 则会重新初始化计数器和预分频器。 复位值: 0x0
0	r/w	CEN: 计数器使能控制 0: 关闭 1: 开启 注意: 只有事先通过软件将 CEN 位置 1, 才可以使用外部时钟、门控模式。而触发模式可通过硬件自动将 CEN 位置 1。 复位值: 0x0

13.5.2 TIM 模式控制寄存器 (TIM_MCR)

偏移地址: 0x0004

表 173 TIM 模式控制寄存器

位	访问	描述
31:23	Res	Reserved 复位值: 0x0
22:20	r/w	TS: 触发信号选择 此位域可选择将要用于同步计数器的触发输入 这些控制位用于选择触发输入作为 TIM_TRGI, 其定义如下, 000~011: 内部触发信号 (TIM_ITR) 100: TI1 边沿检测器 (TIM_TI1F_ED) 101: 滤波后的 TI1 输入 (TIM_TI1FPT) 110: 滤波后的 TI2 输入 (TIM_TI2FPT) 其他: 保留

位	访问	描述
		<p>这些控制位同时也用于选择触发输入作为 TIM_TRC, 其定义如下,</p> <p>000~011: 内部触发信号 (TIM_ITR)</p> <p>100: TI1 边沿检测器 (TIM_TI1F_ED)</p> <p>其他保留</p> <p>注: 这些位只能在未使用的情况下 (例如, SMS=000 时) 进行更改, 以避免转换时出现错误的边沿检测。</p> <p>注: 各个定时器的内部定时器对应关系如表格内部触发对应关系所示</p> <p>复位值: 0x0</p>
19	Res	<p>Reserved</p> <p>复位值: 0x0</p>
18:16	r/w	<p>SMS: 从模式选择</p> <p>选择外部信号时, 触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关 (请参见输入控制寄存器和控制寄存器说明)。</p> <p>000: 从模式关闭</p> <p>如果 CEN = "1", 预分频器时钟直接由内部时钟提供。</p> <p>100: 复位模式</p> <p>在出现所选触发输入 (TRGI) 上升沿时, 重新初始化计数器并生成一个寄存器更新事件。</p> <p>101: 门控模式</p> <p>触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平, 计数器立即停止计数 (但不复位)。计数器的启动和停止都是受控的。</p> <p>110: 触发模式</p> <p>触发信号 TRGI 出现上升沿时启动计数器 (但不复位)。只控制计数器的启动。</p> <p>111: 外部时钟模式</p> <p>由所选触发信号 (TRGI) 的上升沿提供计数器时钟。</p> <p>其他: 保留</p> <p>复位值: 0x0</p>
15:9	Res	<p>Reserved</p> <p>复位值: 0x0</p>
8	r/w	<p>MSM: 主/从模式</p> <p>0: 不执行任何操作</p> <p>1: 当前定时器的触发输入事件 (TRGI) 的动作被推迟, 以使当前定时器与其从定时器实现完美同步 (通过 TRGO)。此设置适用于由单个外部事件对多个定时器进行同步的情况。</p> <p>复位值: 0x0</p>
7:3	Res	<p>Reserved</p> <p>复位值: 0x0</p>
2:0	r/w	<p>MMS: 主模式选择</p> <p>这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下:</p> <p>000: 复位</p>

位	访问	描述
		<p>TIM_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成 (从模式控制器配置为复位模式), 则 TRGO 上的信号相比实际复位会有延迟。</p> <p>001: 使能 计数器使能信号用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器, 或者控制在一段时间内使能从定时器。计数器使能信号可由 CEN 控制位产生。当配置为门控模式时, 也可由触发输入产生。当计数器使能信号由触发输入控制时, TRGO 上会存在延迟, 选择主/从模式时除外 (请参见 TIM_MCR 寄存器中 MSM 位的说明)。</p> <p>010: 更新 选择更新事件作为触发输出 (TRGO)。例如, 主定时器可用作从定时器的预分频器。</p> <p>011: 比较脉冲 一旦发生输入捕获或比较匹配事件, 当 CC1IF 被置 1 时 (即使已为高电平), 触发输出都会发送一个正脉冲 (TRGO)。</p> <p>100: 比较 OC1REF 信号用作触发输出 (TRGO)</p> <p>101: 比较 OC2REF 信号用作触发输出 (TRGO)</p> <p>其他: 保留</p> <p>复位值: 0x0</p>

13.5.3 TIM 捕获/比较模式寄存器 (TIM_CCMR)

偏移地址: 0x0008

表 174 TIM 捕获/比较模式寄存器 (输出比较模式)

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:12	r/w	OC2M: 输出比较 2 模式 参照 OC2M 描述 注: 对于只有 1 个通道的定时器, 此设置位保留 复位值: 0x0
11	r/w	OC2PE: 输出比较 2 预装载使能 参照 OC1PE 相关描述 注: 对于只有 1 个通道的定时器, 此设置位保留 复位值: 0x0
10	Res	Reserved 复位值: 0x0
9:8	r/w	CC2S: 捕获/比较 2 通道选择 00: CC2 通道配置为输出

位	访问	描述
		<p>01: CC2 通道配置为输入, IC2 映射到 TI2 上</p> <p>10: CC2 通道配置为输入, IC2 映射到 TI1 上</p> <p>11: CC2 通道配置为输入, IC2 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_MCR 寄存器) 选择内部触发输入时有效</p> <p>注:</p> <ol style="list-style-type: none"> 1. 仅当通道关闭时 (TIM_CCER 中的 CC2E=0), 才可向 CC2S 位写入数据。 2. 对于只有 1 个通道的定时器, 此设置位保留 <p>复位值: 0x0</p>
7	r/w	<p>OC1CE: 输出比较 1 清零使能</p> <p>0: OC1REF 不受 OCREF_CLR 输入影响</p> <p>1: OCREF_CLR 输入上检测到高电平时, OC1REF 立即清零</p> <p>注: OCREF_CLR 源自于 CMP 输出, 具体配置详见《定时器选项》规格书</p> <p>复位值: 0x0</p>
6:4	r/w	<p>OC1M: 输出比较 1 模式</p> <p>这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效, 而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。</p> <p>000: 冻结。输出比较寄存器 TIM_CCR1 与计数器 TIM_CNT 进行比较不会对输出造成任何影响。</p> <p>001: 将通道 1 设置为匹配时输出有效电平。当计数器 TIM_CNT 与捕获/比较寄存器 1 (TIM_CCR1) 匹配时, OC1REF 信号强制变为高电平。</p> <p>010: 将通道 1 设置为匹配时输出无效电平。当计数器 TIM_CNT 与捕获/比较寄存器 1 (TIM_CCR1) 匹配时, OC1REF 信号强制变为低电平。</p> <p>011: 翻转。TIM_CNT=TIM_CCR1 时, OC1REF 发生翻转。</p> <p>100: 强制变为无效电平。OC1REF 强制变为低电平。</p> <p>101: 强制变为有效电平。OC1REF 强制变为高电平。</p> <p>110: PWM 模式 1。在递增计数模式下, 只要 TIM_CNT<TIM_CCR0, 通道 0 便为有效状态, 否则为无效状态。在递减计数模式下, 只要 TIM_CNT>TIM_CCR0, 通道 0 便为无效状态 (OC0REF="0"), 否则为有效状态 (OC0REF="1")。</p> <p>111: PWM 模式 2。在递增计数模式下, 只要 TIM_CNT<TIM_CCR0, 通道 0 便为无效状态, 否则为有效状态。在递减计数模式下, 只要 TIM_CNT>TIM_CCR0, 通道 0 便为有效状态, 否则为无效状态。</p> <p>注:</p> <ol style="list-style-type: none"> 1. 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 3 且 CC1S="00" (通道配置为输出), 这些位即无法修改。 2. 在 PWM 模式下, 仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时, OC1REF 电平才会发生更改。 <p>1. 此位域将在具有互补输出的通道上进行预装载。如果 TIM_CR 寄存器中的 CCPC 位置 1, 则仅当生成 COM 事件时, OC1M 有效位才会从预装载位获取新值。</p> <p>复位值: 0x0</p>
3	r/w	<p>OC1PE: 输出比较 1 预装载使能</p>

位	访问	描述
		<p>0: 禁止与 TIM_CCR1 相关的预装载寄存器。可随时向 TIM_CCR1 写入数据, 写入后将立即使用新值。</p> <p>1: 使能与 TIM_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIM_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。</p> <p>注:</p> <ol style="list-style-type: none"> 只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式 (TIM_CR 寄存器中的 OPM 位置 1)。其它情况下则无法保证该行。 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 3 且 CC1S="00" (通道配置为输出), 这些位即无法修改。 <p>复位值: 0x0</p>
2	Res	<p>Reserved</p> <p>复位值: 0x0</p>
1:0	r/w	<p>CC1S: 捕获/比较 1 通道选择</p> <p>00: CC1 通道配置为输出</p> <p>01: CC1 通道配置为输入, IC1 映射到 TI1 上</p> <p>10: CC1 通道配置为输入, IC1 映射到 TI2 上</p> <p>11: CC1 通道配置为输入, IC1 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_MCR 寄存器) 选择内部触发输入时有效</p> <p>注: 仅当通道关闭时 (TIM_CCER 中的 CC1E=0), 才可向 CC1S 位写入数据。</p> <p>复位值: 0x0</p>

表 175 TIM 捕获/比较模式寄存器 (输入捕获模式)

位	访问	描述
31:16	Res	<p>Reserved</p> <p>复位值: 0x0</p>
15:12	r/w	<p>IC2F: 输入捕获 2 滤波器</p> <p>参照 IC1F 相关描述</p> <p>注: 对于只有 1 个通道的定时器, 此设置位保留</p> <p>复位值: 0x0</p>
11:10	r/w	<p>IC2PSC: 输入捕获 2 预分频器</p> <p>参照 IC1PSC 相关描述</p> <p>注: 对于只有 1 个通道的定时器, 此设置位保留</p> <p>复位值: 0x0</p>
9:8	r/w	<p>CC2S: 捕获/比较 2 通道选择</p> <p>00: CC2 通道配置为输出</p> <p>01: CC2 通道配置为输入, IC1 映射到 TI2 上</p> <p>10: CC2 通道配置为输入, IC1 映射到 TI1 上</p> <p>11: CC2 通道配置为输入, IC2 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_MCR 寄存器) 选择内部触发输入时有效</p> <p>注意:</p> <ol style="list-style-type: none"> 仅当通道关闭时 (TIM_CCER 中的 CC2E=0), 才可向 CC2S 位写入数据。

位	访问	描述
		2. 对于只有 1 个通道的定时器，此设置位保留 复位值：0x0
7:4	r/w	IC1F：输入捕获 1 滤波器 此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽 IC1F[2:0]=000: $f_{\text{SAMPLING}}=f_{\text{CK_INT}}$ IC1F[2:0]=001: $f_{\text{SAMPLING}}=f_{\text{DTS}}$ IC1F[2:0]=010: $f_{\text{SAMPLING}}=f_{\text{DTS}}/2$ IC1F[2:0]=011: $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$ IC1F[2:0]=100: $f_{\text{SAMPLING}}=f_{\text{DTS}}/8$ IC1F[2:0]=101: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$ IC1F[2:0]=110: $f_{\text{SAMPLING}}=f_{\text{DTS}}/32$ 数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿： IC1F[3]=0: N=1, IC1F[3]=1: N=2 复位值：0x0
3:2	r/w	IC1PSC：输入捕获 1 预分频器 此位域定义 CC1 输入 (IC1) 的预分频比。只要 CC1E=0 (TIM_CCER 寄存器)，预分频器便立即复位。 00: 无预分频器，捕获输入上每检测到一个边沿便执行捕获 01: 每发生 2 个事件便执行一次捕获 10: 每发生 4 个事件便执行一次捕获 11: 每发生 8 个事件便执行一次捕获 复位值：0x0
1:0	r/w	CC1S：捕获/比较 1 通道选择 00: CC1 通道配置为输出 01: CC1 通道配置为输入，IC1 映射到 TI1 上 10: CC1 通道配置为输入，IC1 映射到 TI2 上 (注：对于只有 1 个通道的定时器，此设置位保留) 11: CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_MCR 寄存器) 选择内部触发输入时有效 注意：仅当通道关闭时 (TIM_CCER 中的 CC1E=0)，才可向 CC1S 位写入数据。 复位值：0x0

13.5.4 TIM 捕获/比较模式使能寄存器 (TIM_CCER)

偏移地址： 0x0020

表 176 TIM 捕获/比较模式使能寄存器

位	访问	描述
31:8	Res	Reserved 复位值：0x0
7	r/w	TIM_CC2NP：捕获/比较 2 互补输出极性 参照 CC1NP 描述，该位只在输入模式下有意义 复位值：0x0
6	Res	Reserved 复位值：0x0
5	r/w	TIM_CC2P：捕获/比较 2 输出极性 参照 CC1P 描述 复位值：0x0
4	r/w	TIM_CC2E：捕获/比较 2 输出使能 参照 CC1E 描述 复位值：0x0
3	r/w	CC1NP：捕获/比较 1 互补输出极性 CC1 通道配置为输出： 0：OC1N 高电平有效。 1：OC1N 低电平有效。 CC1 通道配置为输入： 此位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。 注意：只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 2 或 3 且 CC1S="00" (通道配置为输出)，此位立即变为不可写状态。 复位值：0x0
2	r/w	CC1NE：捕获/比较 1 互补输出使能 0：关闭—OC1N 未激活。OC1N 电平是 MOE、OIS1N 和 CC1NE 位的函数。 1：开启—在相应输出引脚上输出 OC1N 信号，具体取决于 MOE、OIS1N、HIZ1N 和 CC1NE 位。 复位值：0x0
1	r/w	CC1P：捕获/比较 1 输出极性 CC1 通道配置为输出： 0：OC1 高电平有效 1：OC1 低电平有效 CC1 通道配置为输入： CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。

位	访问	描述
		<p>00: 非反相/上升沿触发电路对 TlxFP1 上升沿敏感 (在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TlxFP1 未反相 (在门控模式或编码器模式下执行触发操作)。</p> <p>01: 反相/下降沿触发电路对 TlxFP1 下降沿敏感 (在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TlxFP1 反相 (在门控模式或编码器模式下执行触发操作)。</p> <p>10: 保留, 不使用此配置。</p> <p>11: 未反相/边沿触发。电路对 TlxFP1 上升沿和下降沿都敏感 (在复位模式、外部时钟模式或触发模式下执行捕获 或触发操作), TlxFP1 未反相 (在门控模式下执行触发操作)。</p> <p>注意: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 2 或 3, 此位立即变为不可写状态。</p> <p>复位值: 0x0</p>
0	r/w	<p>CC1E: 捕获/比较 1 输出使能</p> <p>CC1 通道配置为输出:</p> <p>0: 关闭——OC1 未激活。OC1 电平是 MOE、OIS1 和 CC1E 位的函数。</p> <p>1: 开启——OC1 信号是相应输出引脚上的输出, 具体取决于 MOE、OIS1、HIZ1 和 CC1E 位。</p> <p>CC1 通道配置为输入:</p> <p>此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1 (TIM_CCR1) 中。</p> <p>0: 禁止捕获。</p> <p>1: 使能捕获。</p> <p>复位值: 0x0</p>

13.5.5 TIM 计数器 (TIM_CNT)

偏移地址: 0x0060

表 177 TIM 计数器

位	访问	描述
31:16	Res	<p>Reserved</p> <p>复位值: 0x0</p>
15:0	r/w	<p>CNT[15:0]: 计数器值</p> <p>注: 计数器值支持计数过程中更改修正, 但这是一个异步操作; 计数器亦可在使能信号起来之前进行软件初值加载。</p> <p>复位值: 0x0</p>

13.5.6 TIM 预分频器 (TIM_PSC)

偏移地址: 0x0064

表 178 TIM 预分频器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	PSC[15:0]: 预分频器值 计数器时钟频率 (CK_CNT) 等于 $f_{IP} / (PSC[15:0] + 1)$ 。PSC 包含每次发生更新事件 (包括计数器通过 TIM_EGR 寄存器中的 UG 位清零时, 或在配置为“复位模式”时通过触发控制器清零时) 时要装载到活动预分频器寄存器的值。 复位值: 0x0

13.5.7 TIM 自动重载寄存器 (TIM_ARR)

偏移地址: 0x0068

表 179 TIM 自动重载寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	ARR[15:0]: 自动重载值 ARR 为要装载到实际自动重载寄存器的值。当自动重载值为空时, 计数器不工作。 复位值: 0xFFFF

13.5.8 TIM 重复计数器寄存器 (TIM_RCR)

偏移地址: 0x006C

表 180 TIM 重复计数器寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7:0	r/w	REP[7:0]: 重复计数器值 使能预装载寄存器时, 用户可通过这些位设置比较寄存器的更新频率 (即, 从预装载寄存器向活动寄存器周期性传输数据); 使能更新中断时, 也可设置更新中断的生成速率。 与 REP_CNT 相关的减计数器每次计数到 0 时, 都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件时, REP_CNT 才会重载 REP 值, 因此在生成下一重复更新事件之前, 无论向 TIM_RCR 寄存器写入何值都无影响。这意味着 PWM 模式下 (REP+1) 相当于: <ul style="list-style-type: none"> 边沿对齐模式下的 PWM 周期数

位	访问	描述
		<ul style="list-style-type: none"> 中心对齐模式下的 PWM 半周期数 复位值：0x0

13.5.9 TIM 捕获/比较寄存器 1 (TIM_CCR1)

偏移地址：0x0070

表 181 TIM 捕获/比较寄存器 1

位	访问	描述
31:16	Res	Reserved 复位值：0x0
15:0	r/w	CCR1[15:0]：捕获/比较 1 值 通道 CC1 配置为输出。CCR1 是捕获/比较寄存器 1 的预装载值。如果没有通过 TIM_CCMR 寄存器中的 OC1PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 1）。实际捕获/比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC1 输出上发出信号的值。 通道 CC1 配置为输入。CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。此时该寄存器只读不可写。 复位值：0x0

13.5.10 TIM 捕获/比较寄存器 2 (TIM_CCR2)

偏移地址：0x0074

表 182 TIM 捕获/比较寄存器 2

位	访问	描述
31:16	Res	Reserved 复位值：0x0
15:0	r/w	CCR2[15:0]：捕获/比较 2 值 通道 CC2 配置为输出。CCR2 是捕获/比较寄存器 2 的预装载值。如果没有通过 TIM_CCMR 寄存器中的 OC2PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 2）。实际捕获/比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC2 输出上发出信号的值。 通道 CC2 配置为输入。CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。此时该寄存器只读不可写。 复位值：0x0

13.5.11 TIM 断路及死区寄存器 (TIM_BDTR)

偏移地址: 0x0090

表 183 TIM 断路及死区寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15	r/w	MOE: 主输出使能 只要断路输入变为有效状态, 此位便由硬件异步清零。此位由软件置 1, 也可根据 AOE 位状态自动置 1。此位仅对配置为输出的通道有效。 0: OC 和 OCN 输出禁止或被强制为空闲状态。 1: 如果 OC 和 OCN 输出的相应使能位 (TIM_CCER 寄存器中的 CCxE 和 CCxNE 位) 均置 1, 则使能 OC 和 OCN 输出。 复位值: 0x0
14	r/w	AOE: 自动输出使能 0: MOE 只能由软件置 1 1: MOE 可由软件置 1, 也可在发生下一更新事件时自动置 1 (如果断路输入无效) 注: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1, 此位即无法修改 复位值: 0
13	Res	Reserved 复位值: 0x0
12	r/w	BKE: 断路使能 0: 禁止断路输入 1: 使能断路输入 注意: 编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1 后, 此位即无法修改。注意: 对该位执行任何写操作后, 都需要经过 1 个 APB 时钟周期的延迟才生效。 复位值: 0x0
11:10	Res	Reserved 复位值: 0x0
9:8	r/w	LOCK[1:0]: 锁定配置 这些位用于针对软件错误提供写保护。 00: 关闭锁定--不对任何位提供写保护。 01: 锁定级别 1, 此时无法对 TIM_BDTR 寄存器中的 DTG*位、TIM_CR 寄存器中的 OISx 和 OISxN 位以及 TIM_BDTR 寄存器中的 BKE/BKP/AOE 位执行写操作。 10: 锁定级别 2, 此时无法对锁定级别 1 中适用的各位、CC 极性位 (TIM_CCER 寄存器中的 CCxP/CCxNP 位, 只要通过 CCxS 位将相关通道配置为输出) 执行写操作。 11: 锁定级别 3, 此时无法对锁定级别 2 中适用的各位、CC 控制位 (TIM_CCMRx 寄存器中的 OCxM 和 OCxPE 位, 只要通过 CCxS 位将相关通道配置为输出) 执行写操作。

位	访问	描述
		注意：复位后只能对 LOCK 位执行一次写操作。对 TIM_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。 复位值：0x0
7:0	r/w	DTG[7:0]：配置死区发生器 此位域定义插入到互补输出之间的死区持续时间。死区时间 DT 与该持续时间相对应： DTG[7:5]=0XX: $DT=DTG[7:0]*t_{dtg}$, 其中 $t_{dtg}=t_{DTS}$ 。 DTG[7:5]=10X: $DT=(64+DTG[5:0])*t_{dtg}$, 其中 $t_{dtg}=2*t_{DTS}$ 。 DTG[7:5]=110: $DT=(32+DTG[4:0])*t_{dtg}$, 其中 $t_{dtg}=8*t_{DTS}$ 。 DTG[7:5]=111: $DT=(32+DTG[4:0])*t_{dtg}$, 其中 $t_{dtg}=16*t_{DTS}$ 。 例如，如果 $t_{DTS}=125ns$ (8MHz)，则可能的死区值为： 0 到 15875ns (步长为 125ns) 16us 到 31750ns (步长为 250ns) 32us 到 63us (步长为 1us) 64us 到 126us (步长为 2us) 注意：只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1、2 或 3，此位域即无法修改。 复位值：0x0

13.5.12 TIM 配置密钥寄存器 (TIM_SET_KR)

偏移地址：0x0094

表 184 TIM 配置密钥寄存器 (写)

位	访问	描述
31:0	wo	TIM_SET_UNLOCK_KEY: TIM 配置寄存器解锁密钥 注：在默认情况下，TIM 配置寄存器 (TIM_CNT) 是不能直接进行写操作，在写之前需要输入正确密钥 0xD79B3EFD 以解锁写权限，即 TIM_SET_UNLOCK=1，在更新完系统配置寄存器之后，写入其他值便可重新锁定写权限，即 TIM_SET_UNLOCK=0。 复位值：0x0

表 185 TIM 配置密钥寄存器 (读)

位	访问	描述
31:1	Res	Reserved 复位值：0x0
0	ro	TIM_SET_UNLOCK, TIM 配置寄存器解锁与否 0: 锁定 1: 解锁 复位值：0x0

13.5.13 TIM 事件生成寄存器 (TIM_EGR)

偏移地址: 0x00B0

表 186 TIM 事件生成寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	r/w	BG: 断路生成 此位由软件置 1 以生成事件, 并由硬件自动清零。 0: 不执行任何操作 1: 生成断路事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断事件。 复位值: 0x0
6	Res	Reserved 复位值: 0x0
5	r/w	COMG: 捕获/比较控制更新生成 该位可通过软件置 1, 并由硬件自动清零 0: 不执行任何操作 1: CCPC 位置 1 时, 可更新 CCxE、CCxNE 和 OCxM 位 复位值: 0x0
4:3	Res	Reserved 复位值: 0x0
2	r/w	CC2G: 捕获/比较 2 生成 参照 CC1G 描述 复位值: 0x0
1	r/w	CC1G: 捕获/比较 1 生成 此位由软件置 1 以生成事件, 并由硬件自动清零。 0: 不执行任何操作 1: 通道 1 上生成捕获/比较事件: <ul style="list-style-type: none"> 通道 CC1 配置为输出: 使能时, CC1IF 标志置 1 并发送相应的中断。 通道 CC1 配置为输入: TIM_CCR1 寄存器中将捕获到计数器当前值。使能时, CC1IF 标志置 1 并发送相应的中断。如果 CC1IF 标志已为高电平, CC1OF 标志将置 1。 复位值: 0x0
0	r/w	UG: 更新生成 该位可通过软件置 1, 并由硬件自动清零。 0: 不执行任何操作 1: 重新初始化计数器并生成一个寄存器更新事件。 注: 预分频器计数器也将清零 (但预分频比不受影响), 而且计数器将清零。 复位值: 0x0

13.5.14 TIM 中断使能寄存器 (TIM_IER)

偏移地址: 0x00C0

表 187 TIM 中断使能寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	r/w	BIE: 断路中断使能 0: 关闭 1: 开启 复位值: 0x0
6	r/w	TIE: 触发信号 (TGRI) 中断使能 0: 关闭 1: 开启 复位值: 0x0
5	r/w	COMIE: COM 中断使能 0: 关闭 1: 开启 复位值: 0x0
4:3	Res	Reserved 复位值: 0x0
2	r/w	CC2IE: 捕获/比较 2 中断使能 0: 关闭 1: 开启 复位值: 0x0
1	r/w	CC1IE: 捕获/比较 1 中断使能 0: 关闭 1: 开启 复位值: 0x0
0	r/w	UIE: 更新中断使能 0: 关闭 1: 开启 复位值: 0x0

13.5.15 TIM 状态寄存器 (TIM_SR)

偏移地址: 0x00C4

表 188 TIM 状态寄存器

位	访问	描述
31:11	Res	Reserved 复位值: 0x0
10	r/w	CC2OF: 捕获/比较 2 重复捕获标志 0: 未检测到重复捕获。 1: TIM_CCR2 寄存器中已捕获到计数器值且 CC2IF 标志已置 1。 仅当将相应通道配置为输入捕获模式时, 此标志位才会由硬件置 1。该位写 1 清零。 复位值: 0x0
9	r/w	CC1OF: 捕获/比较 1 重复捕获标志 0: 未检测到重复捕获。 1: TIM_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。 仅当将相应通道配置为输入捕获模式时, 此标志位才会由硬件置 1。该位写 1 清零。 复位值: 0x0
8	Res	Reserved 复位值: 0x0
7	r/w	BIF: 断路中断标志 0: 未发生断路事件。 1: 在断路输入上检测到有效电平。 只要断路输入变为有效状态, 此标志便由硬件置 1。断路输入无效后可通过软件对其写 1 清零。 复位值: 0x0
6	r/w	TIF: 触发中断标志 在除门控模式以外的所有模式下, 当使能从模式控制器后在 TRGI 输入上检测到有效边沿时, 该标志将由硬件置 1。选择门控模式时, 该标志将在计数器启动或停止时置 1。但需要通过软件写 1 清零。 0: 未发生触发事件。 1: 触发中断挂起。 复位值: 0x0
5	r/w	COMIF: COM 中断标志 此标志在发生 COM 事件时 (捕获/比较控制位 CCxE、CCxNE 和 OCxM 已更新时)由硬件置 1。但需要通过软件清零。 0: 未发生 COM 事件。 1: COM 中断挂起。 复位值: 0x0
4:3	Res	Reserved 复位值: 0x0

位	访问	描述
2	r/w	CC2IF: 捕获/比较 2 中断标志 参照 CC1IF 描述 复位值: 0x0
1	r/w	CC1IF: 捕获/比较 1 中断标志 通道 CC1 配置为输出 当计数器与比较值匹配时, 此标志由硬件置 1, 软件写 1 清零。 0: 不匹配。 1: TIM_CNT 计数器的值与 TIM_CCR1 寄存器的值匹配。当 TIM_CCR1 的值大于 TIMx_ARR 的值时, CC1IF 位将在计数器发生上溢时变为高电平。 通道 CC1 配置为输入 此位将在发生捕获事件时由硬件置 1。通过软件写 1 清零或读取 TIM_CCR1 寄存器将该位清零。 0: 未发生输入捕获事件 1: TIM_CCR1 寄存器中已捕获到计数器值 (IC1 上已检测到与所选极性匹配的边沿) 复位值: 0x0
0	r/w	UIF: 更新中断标志 该位在发生更新事件时通过硬件置 1。但需要通过软件写 1 清零。 0: 未发生更新。 1: 更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1: TIM_CR 寄存器中的 URS=0 且 UDIS=0, 并且由软件使用 TIM_EGR 寄存器中的 UG 位重新初始化 CNT 时。 TIM_CR 寄存器中的 URS=0 且 UDIS=0, 并且 CNT 由触发事件重新初始化时 (请参见模式控制寄存器 (TIM_MCR))。 复位值: 0x0

14 通用定时器 (TIMG)

14.1 概述

该通用定时器 (TIMG) 是由一个带自动加载功能的 16/32 位计数器构成，支持可编程预分频计数。定时器可用于多种用途：对输入信号脉冲宽度量测、比较输出和 PWM 输出。其中 TIMG0 是 32 位定时器，TIMG1/2 是 16 位定时器。

14.2 主要功能

- 16/32 位递增、递减、递增/递减自动重载计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数介于 1 到 65536 之间
- 4 个独立通道功能，可用于
 - 输入捕获
 - 输出比较
 - PWM 输出（边沿和中心对齐模式）
 - 单脉冲模式输出
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路
- 支持以下事件的中断生成：
 - 更新：计数器溢出，由软件或内外部事件触发计数器初始化
 - 触发事件（计数器开始、停止、初始化或内外部触发导致的计数动作）
 - 输入捕获
 - 输出比较
 - 断路输入
- 支持定位用增量（正交）编码器和霍尔传感器电路
- 外部时钟触发输入或逐周期电流管理

14.3 模块框图

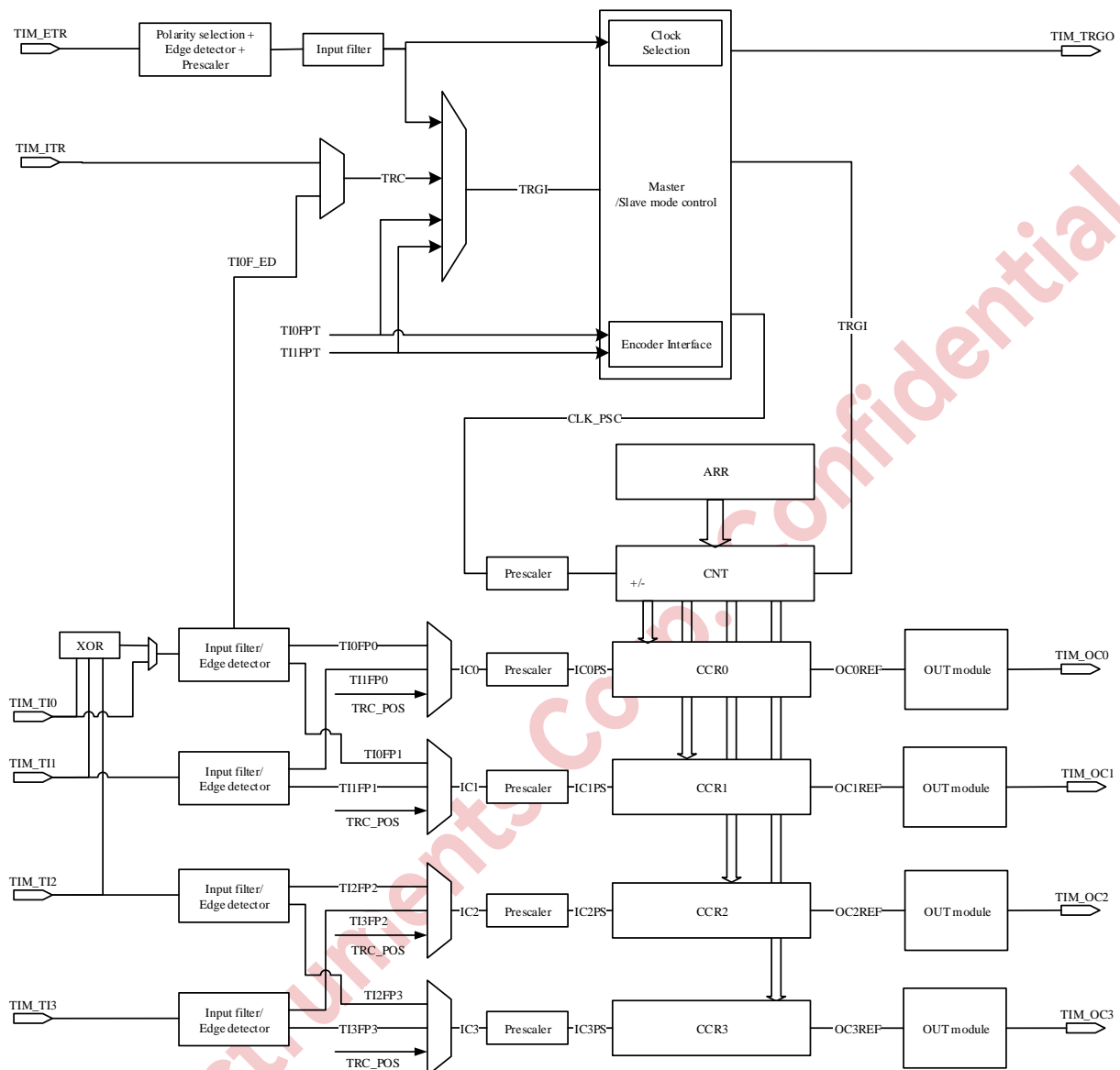


图 69 通用定时器 TIM 总体模块框图

14.4 功能描述

14.4.1 计数器介绍

可编程定时器的主要模块由一个 16 位计数器及其相关的自动重载寄存器组成。此计数器采用递增、递减或中心对齐方式计数。计数器的时钟可通过预分频器进行分频。计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。时基单元包括计数器寄存器 (TIM_CNT)、预分频寄存器 (TIM_PSC) 和自动重载寄存器 (TIM_ARR)。

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIM_CR0 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIM_CR0 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

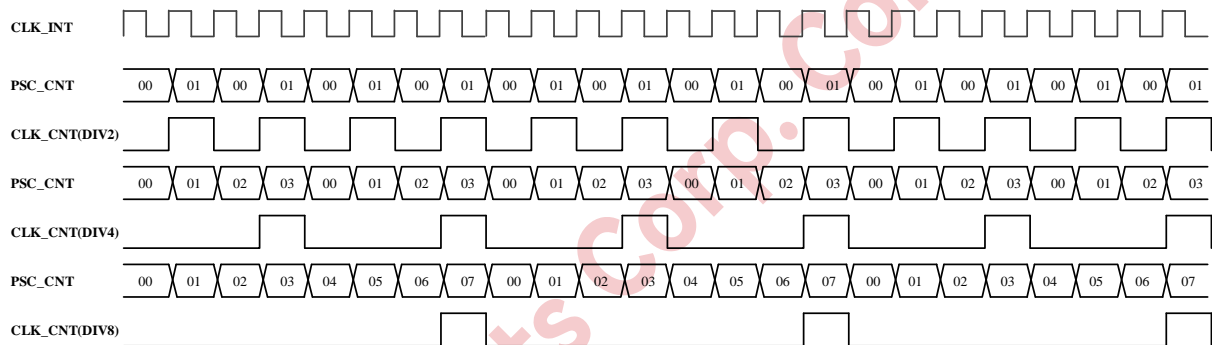


图 70 TIM 预分频器 N 分频时序图

14.4.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值 (TIM_ARR 寄存器的内容)，然后重新从 0 开始计数并生成计数器上溢事件。

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

通过软件将 TIM_CR0 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数 (而预分频比保持不变)。此外，如果 TIM_CR0 寄存器中的 URS 位 (更新请求选择) 已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1 (因此，不会发送任何中断或 DMA 请求)。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

以下各图以一些示例说明当 TIM_ARR=0x36 时不同时钟频率下计数器的行为。

- 自动重载影子寄存器将以预装载值（TIM_ARR）进行更新
- 预分频器的缓冲区中将重新装载预装载值（TIM_PSC 寄存器的内容）

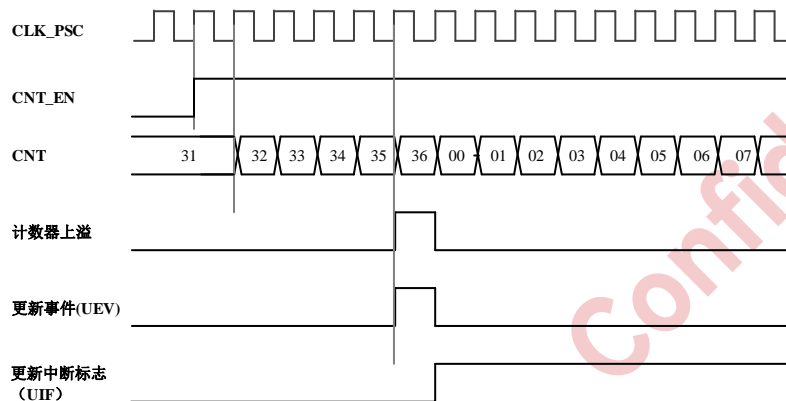


图 71 计数器时序图，1 分频内部时钟

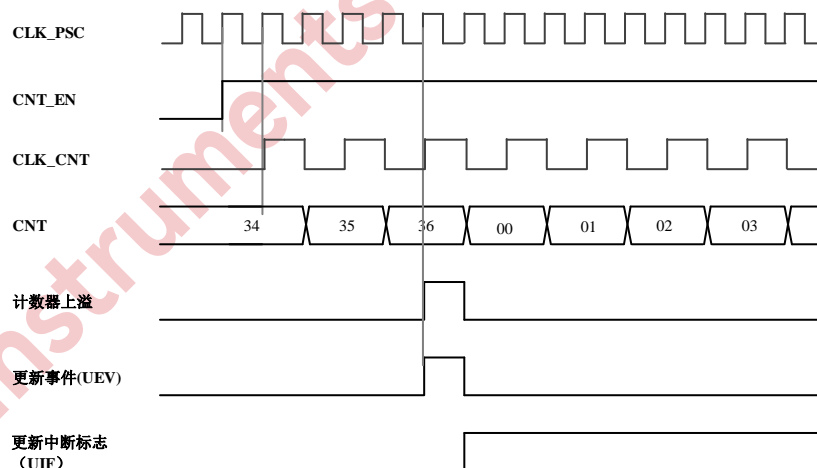


图 72 计数器时序图，2 分频内部时钟

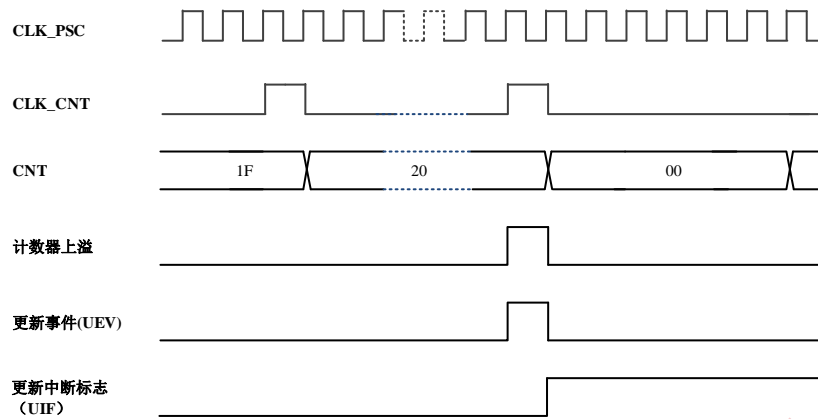


图 73 计数器时序图，N 分频内部时钟

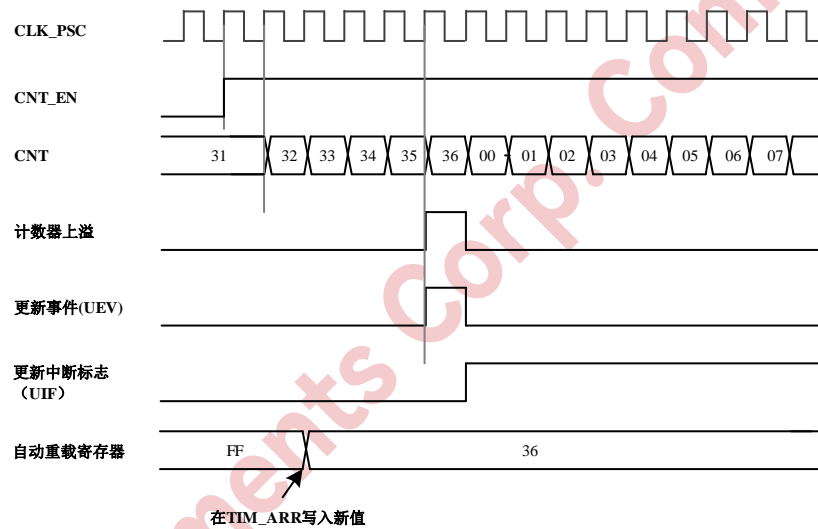


图 74 计数器时序图，ARPE=0 时更新事件 (TIM_ARR 未预装载)

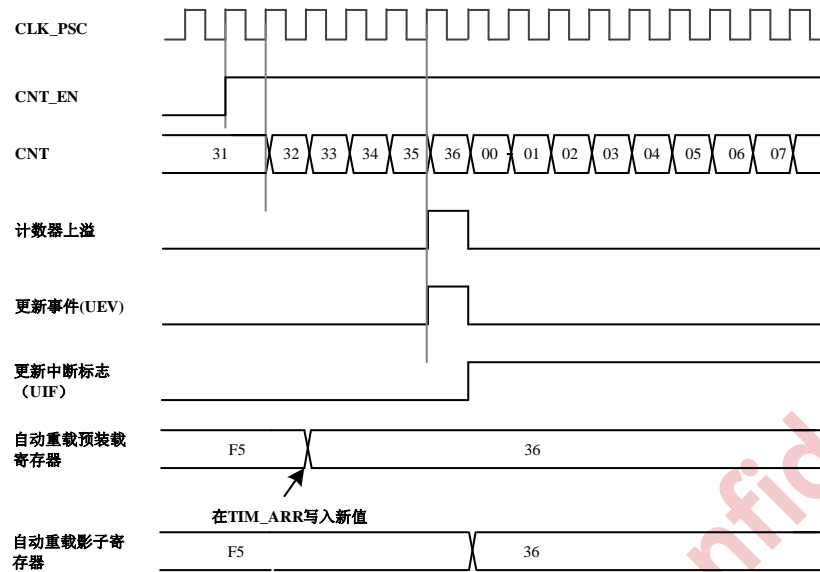


图 75 计数器时序图，ARPE=1 时更新事件 (TIM_ARR 预装载)

递减计数模式

在递减计数模式下，计数器从自动重载值 (TIM_ARR 寄存器的内容) 开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

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

通过软件将 TIM_CR0 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数 (但预分频比保持不变)。

此外，如果 TIM_CR0 寄存器中的 URS 位 (更新请求选择) 已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1 (因此，不会发送任何中断或 DMA 请求)。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 预分频器的缓冲区中将重新装载预装载值 (TIM_PSC 寄存器的内容)
- 自动重载活动寄存器将以预装载值 (TIM_ARR 寄存器的内容) 进行更新。注意，自动重载寄存器会在计数器重载之前得到更新，因此，下一个计数周期就是我们所希望的新的周期长度。

以下各图以一些示例说明当 TIM_ARR=0x36 时不同时钟频率下计数器的行为。

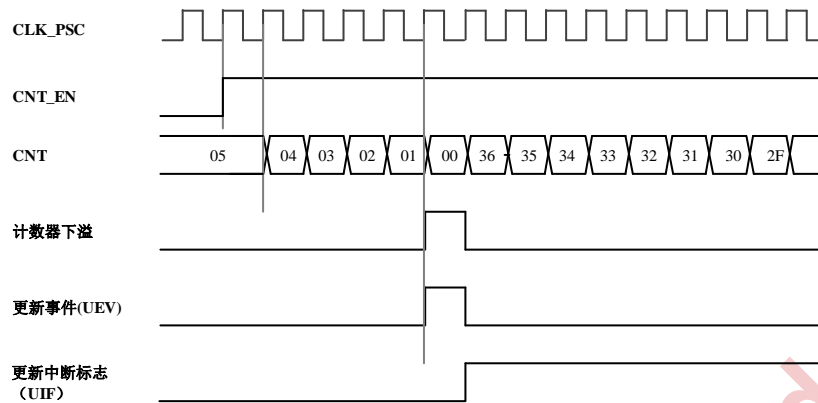


图 76 计数器时序图，1 分频内部时钟

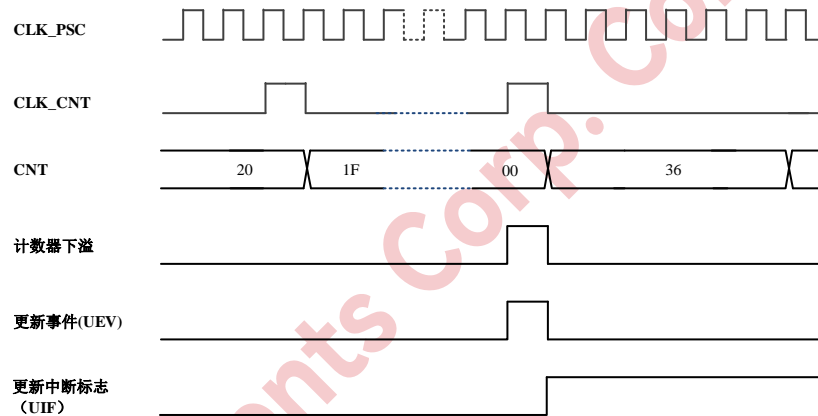


图 77 计数器时序图，N 分频内部时钟

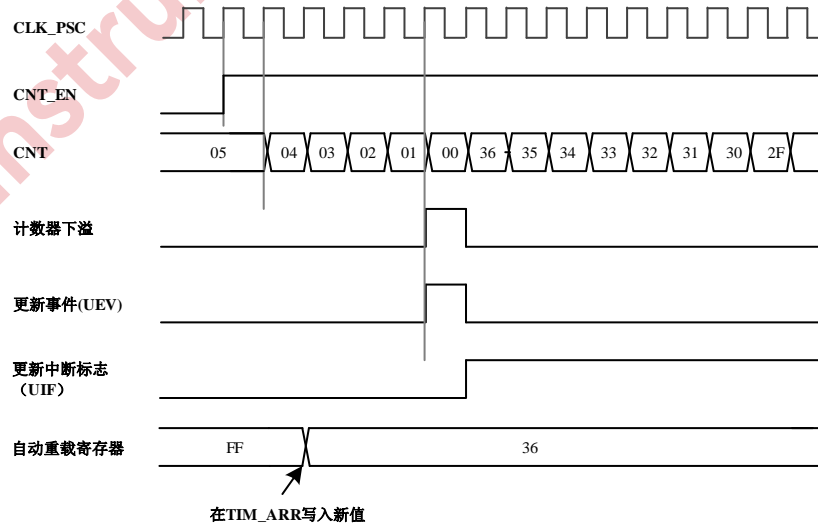


图 78 计数器时序图，未使用重复计数器时更新事件

中心对齐模式 (递增/递减计数)

在中心对齐模式下，计数器从 0 开始计数到自动重载值 (TIM_ARR 寄存器的内容) - 1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIM_CR0 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数 (中心对齐模式 1，CMS=“01”)、计数器递增计数 (中心对齐模式 2，CMS=“10”) 以及计数器递增/递减计数 (中心对齐模式 3，CMS=“11”)。

在此模式下，TIM_CR0 寄存器的 DIR 方向位不可写入值，而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TIM_EGR 寄存器中的 UG 位置 1 (通过软件或使用从模式控制器) 也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

通过软件将 TIM_CR0 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

此外，如果 TIM_CR0 寄存器中的 URS 位 (更新请求选择) 已置 1，则将 UG 位置 1 会生成 UEV 更新事件，但不会将 UIF 标志置 1 (因此，不会发送任何中断或 DMA 请求)。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 预分频器的缓冲区中将重新装载预装载值 (TIM_PSC 寄存器的内容)
- 自动重载活动寄存器将以预装载值 (TIM_ARR 寄存器的内容) 进行更新。注意，如果更新操作是由计数器上溢触发的，则自动重载寄存器在重载计数器之前更新，因此，下一个计数周期就是我们所希望的新的周期长度 (计数器被重载新的值)。

以下各图以一些示例说明不同时钟频率下计数器的行为。

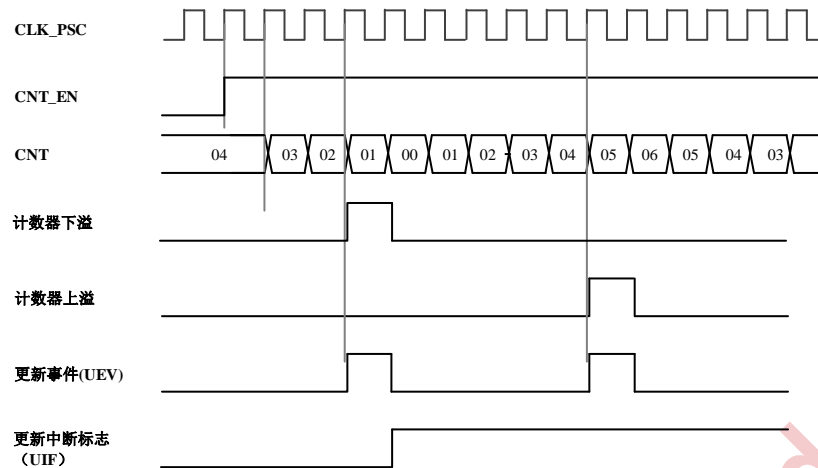


图 79 计数器时序图，1 分频内部时钟，TIM_ARR = 0x6

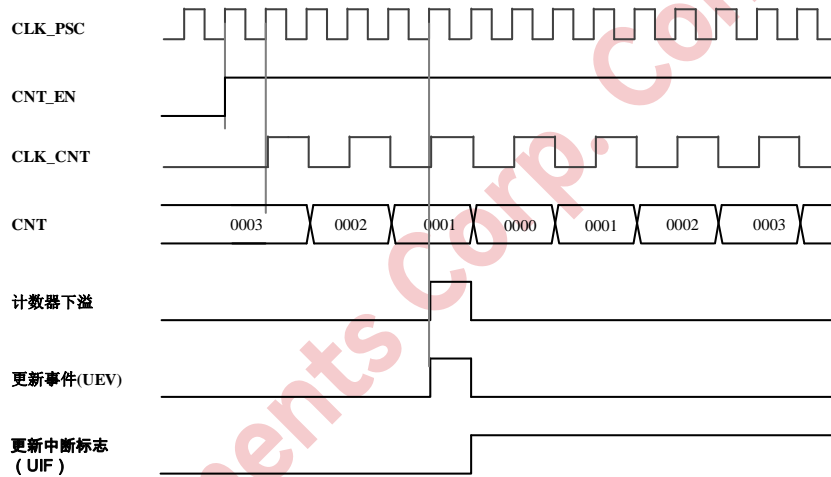


图 80 计数器时序图，2 分频内部时钟

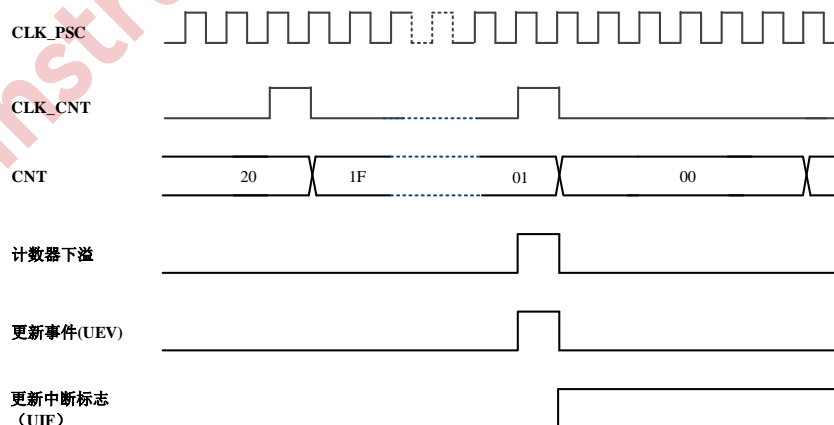


图 81 计数器时序图，N 分频内部时钟

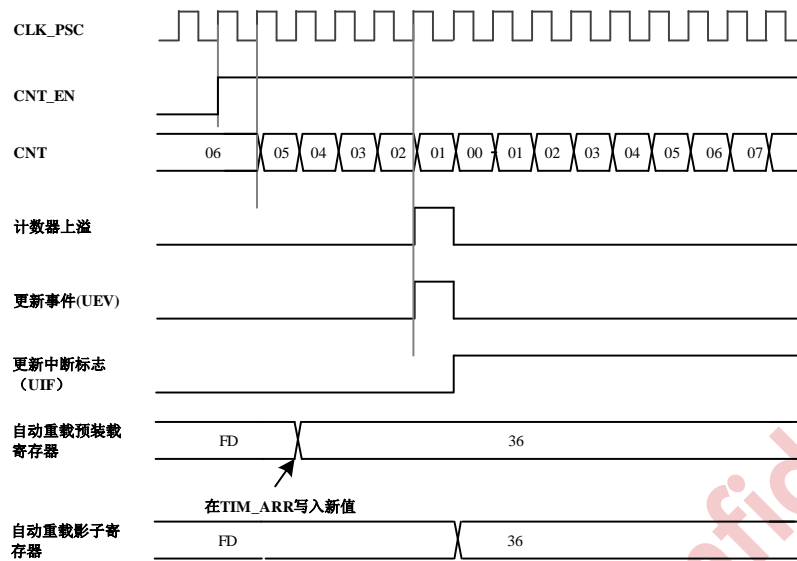


图 82 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

14.4.3 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟 (TIM_CLK_INT)
- 外部时钟模式 0：外部输入引脚 (TIM_TIx)
- 外部时钟模式 1：外部触发输入 ETR
- 内部触发输入 (TIM_ITRx)：使用一个定时器作为另一个定时器的预分频器

14.4.3.1 内部时钟源

如果禁止从模式控制器 (TIM_SMCR 寄存器中 SMS=000)，则 CEN 位 (TIM_CR0) 和 UG 位 (TIM_EGR) 为实际控制位，并且只能通过软件进行更改，但 UG 仍自动清零)。当对 TIM_CEN 位写入 1 时，预分频器的时钟就由内部时钟 TIM_CK_INT 提供。

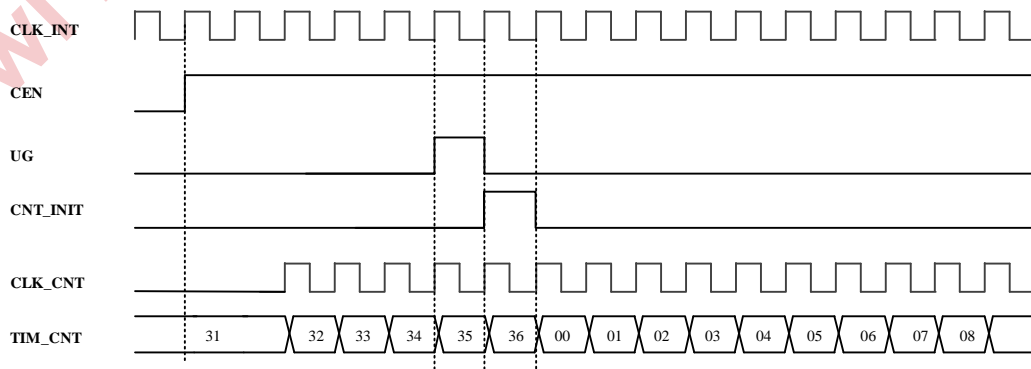


图 83 正常模式下的控制电路 1 分频内部时钟

14.4.3.2 外部时钟源模式 0

当 TIM_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

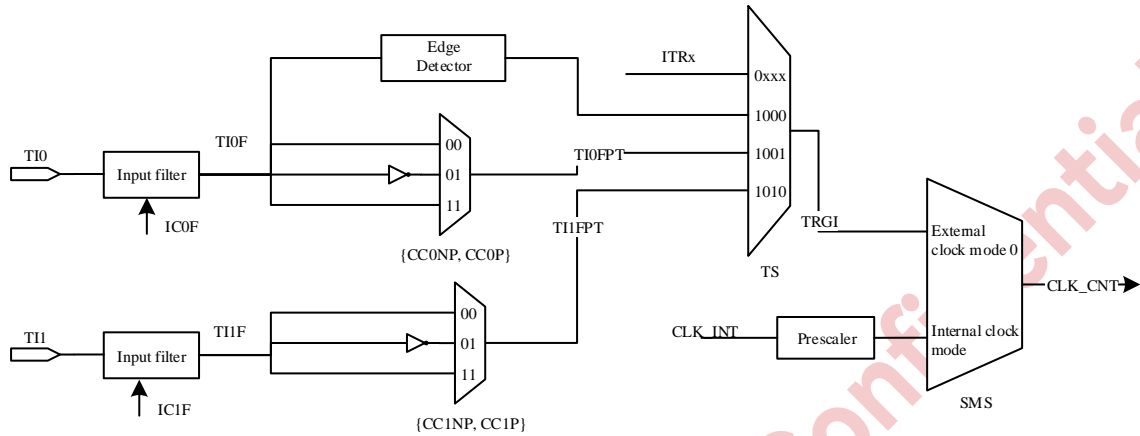


图 84 外部时钟输入连接关系

例如，要使计数器在 TI1 输入出现上升沿时计数，可执行以下步骤：

1. 通过在 TIM_CCMR0 寄存器中写入 CC1S=01 来配置通道 1，使其能够检测 TI1 输入的上升沿。
2. 通过在 TIM_CCMR0 寄存器中写入 IC1F[3:0] 位来配置输入滤波时间（如果不需要任何滤波，请保持 IC1F=0000）。
3. 通过在 TIM_CCER 寄存器中写入 CC1P=0 和 CC1NP=0 来选择上升沿极性。
4. 通过在 TIM_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式下工作。
5. 通过在 TIM_SMCR 寄存器中写入 TS=110 来选择 TI1 作为输入源。
6. 通过在 TIM_CR0 寄存器中写入 CEN=1 来使能计数器。

当 TI1 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。TI1 的上升沿与实际计数器时钟之间的延迟是由于 TI1 输入的重新同步电路引起的。

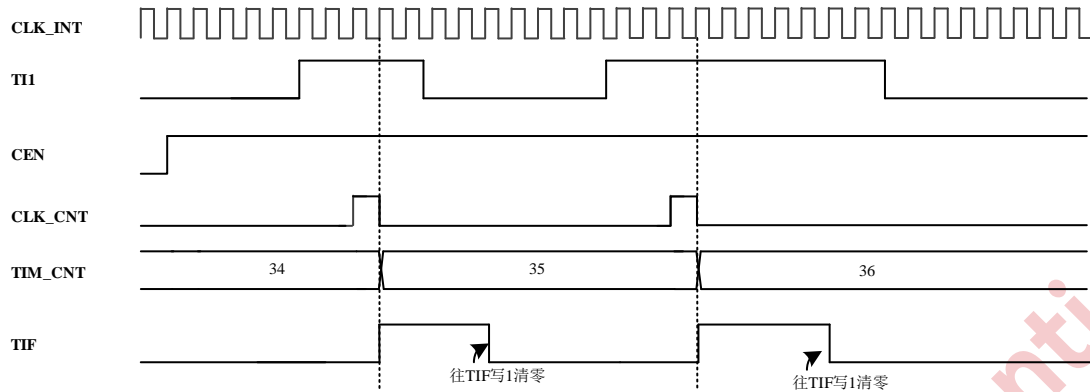


图 85 外部时钟模式下 1 的控制电路

14.4.3.3 外部时钟源模式 1

通过在 TIM_SMCR 寄存器中写入 ECE=1 可选择此模式。计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。下图简要介绍了外部触发输入模块。

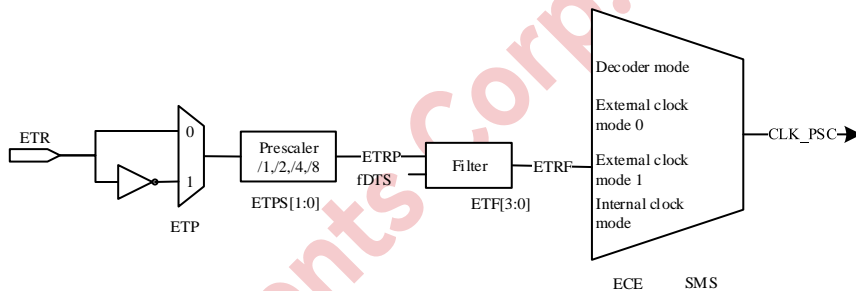


图 86 外部触发输入模块

例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 由于此例中不需滤波器，因此在 TIM_SMCR 寄存器中写入 ETF[3:0]=0000。
2. 通过在 TIM_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
3. 通过在 TIM_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
4. 通过在 TIM_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 1。
5. 通过在 TIM_CR0 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的。

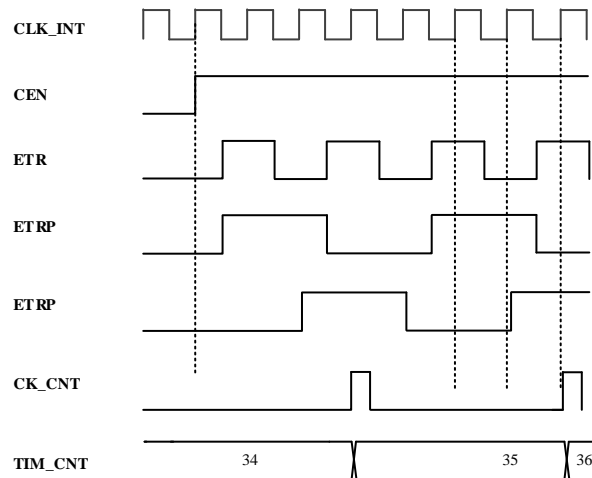


图 87 外部时钟模式 1 下的控制电路

14.4.4 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

下图概括介绍了一个捕获/比较通道。输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 $TIxF$ 。然后，带有极性选择功能的边沿检测器生成一个信号 ($TIxFPx$)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 ($ICxPS$)，而后再进入捕获寄存器。

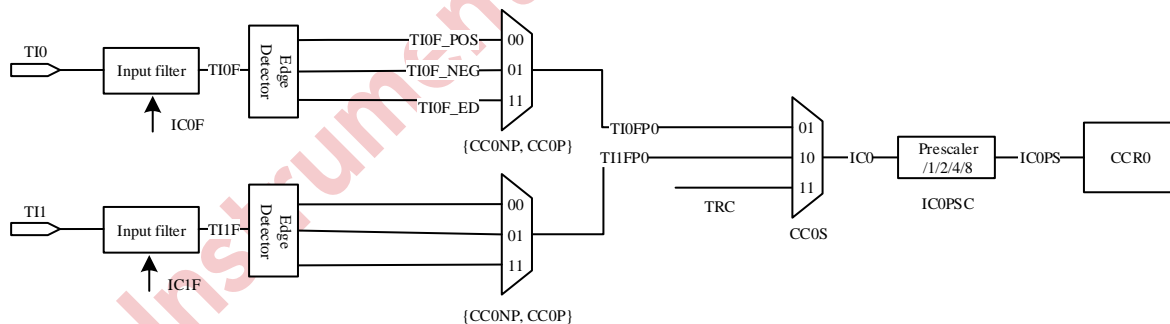


图 88 捕获/比较通道的输入阶段

输出阶段生成一个中间波形作为基准：OCxREF (高电平有效)。链的末端决定最终输出信号的极性。

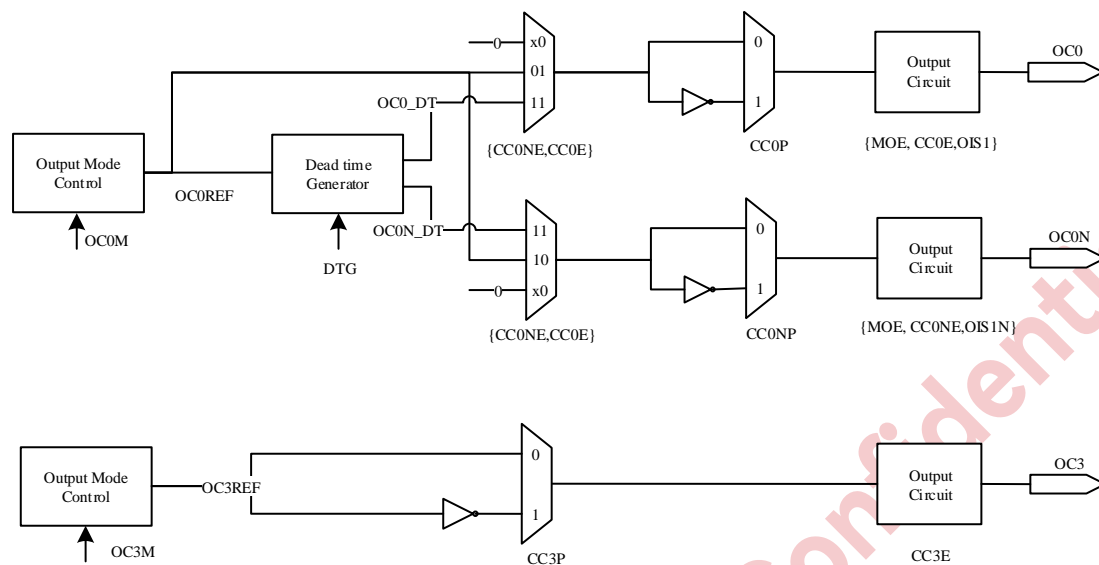


图 89 捕获/比较通道的输出阶段

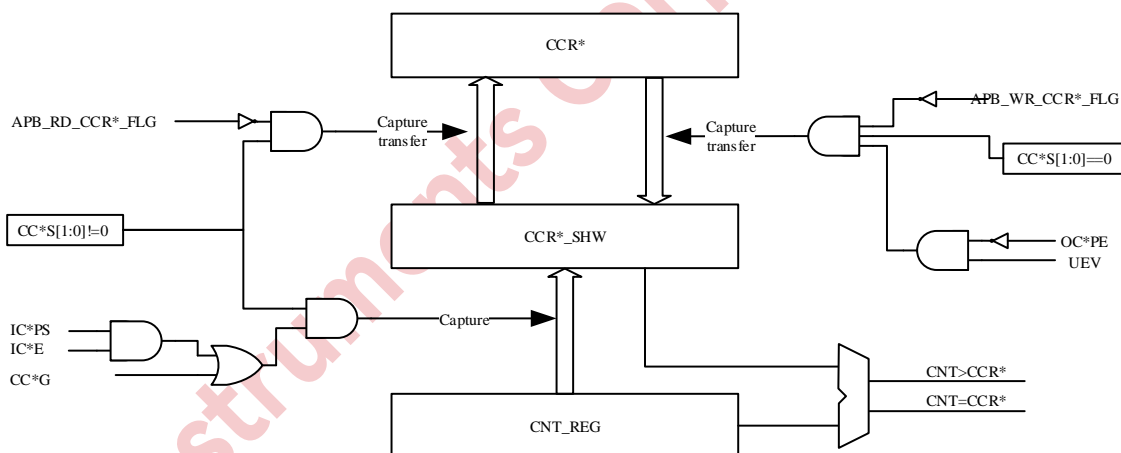


图 90 捕获/比较通道主要电路

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载寄存器。在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。在比较模式下，预装载寄存器的内容将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

14.4.5 输入捕获模式

在输入捕获模式下，当相应的 TIM_TIx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIM_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCxIF 标志 (TIM_SR) 置 1，并可发送中断请求 (如果已使能)。如果发生捕获事件时 CCxIF 标志已置位，则会将重复捕获标志 CCxOF (TIM_SR) 置 1。可通过软件向 CCxIF 写入 1 来给 CCxIF 清零，或读取存储在 TIM_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 1 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIM_CCR1 中。具体操作步骤如下：

1. 选择有效输入：TIM_CCR1 必须连接到 TI1 输入，因此向 TIM_CCMR 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIM_CCR1 寄存器将处于只读状态。
2. 根据连接到定时器的信号，对所需的输入滤波时间进行设置 (如果输入为 Tlx 输入之一，则对 TIM_CCMR 寄存器中的 ICxF 位进行设置)。输入的信号经过采样时钟进行采样，采样频率可配置 (详见相关寄存器描述)。
3. 通过向 TIM_CCER 寄存器中的 CC1P 位和 CC1NP 位写入 0，选择 TI1 通道的有效转换边沿上升沿。
4. 对输入预分频器进行设置。本例中，倘若每次有效转换时都执行捕获操作，那需要禁止预分频器 (向 TIM_CCMR0 寄存器中的 IC1PSC 位写入 0)。
5. 通过将 TIM_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
6. 必要时可通过将 TIM_IER 寄存器中的 CC1IE 位置 1 来使能相关中断请求。

当发生输入捕获时：

- 发生有效跳变沿时，TIM_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1 (中断标志)。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

通过软件将 TIM_EGR 寄存器中的相应 CCxG 位置 1 可生成中断。

14.4.6 PWM 输入模式

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 TIM_TIx 信号被映射至同一个 ICx 输入。
- 这两个 TIM_TIx 信号在边沿处有效，但极性相反。
- 选择两个 TixFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI0 的 PWM 的周期（位于 TIM_CCR0 寄存器中）和占空比（位于 TIM_CCR1 寄存器中）进行测量（取决于 TIM_CK_INT 频率和预分频器的值）

1. 选择 TIM_CCR0 的有效输入：向 TIM_CCMR0 寄存器中的 CC0S 位写入 01 (选择 TI0)。
2. 选择 TI0FP1 的有效极性（用于 TIM_CCR0 中的捕获和计数器清零）：向 CC0P 位和 CC0NP 位写入 0 (上升沿有效)。
3. 选择 TIM_CCR1 的有效输入：向 TIM_CCMR 寄存器中的 CC1S 写入 10 (选择 TI0)。
4. 选择 TI0FP1 的有效极性（用于 TIM_CCR1 中的捕获）：向 CC1P 位和 CC1NP 位写入 1 (下降沿有效)。
5. 选择有效触发输入：向 TIM_SMCR 寄存器中的 TS 位写入 101 (选择 TI0FPT)。
6. 将从模式控制器配置为复位模式：向 TIM_SMCR 寄存器中的 SMS 位写入 100。
7. 使能计数器开始计数 CEN=1。
8. 使能捕获：向 TIM_CCER 寄存器中的 CC0E 位和 CC1E 位分别写入 1。

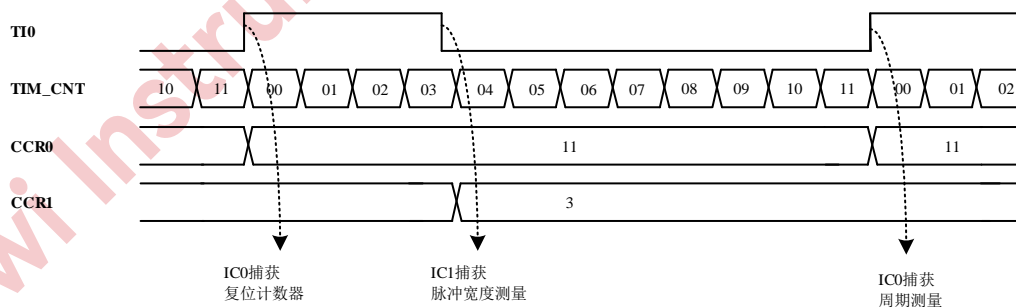


图 91 PWM 输入模式时序

14.4.7 强制输出模式

在输出模式 (TIM_CCMR 中的 CCxS=00) 下, 可直接由软件将每个输出比较信号 (OCxREF 和 OCx) 强制设置为有效电平或无效电平, 而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (OCxREF/OCx) 强制设置为有效电平, 只需向相应 TIM_CCMR 寄存器中的 OCxM 位写入 101。OCxREF 进而强制设置为高电平 (OCxREF 始终为高电平有效), 同时 OCx 获取 CCxP 极性位的相反值。例如: CCxP=0 (OCx 高电平有效) => OCx 强制设置为高电平。通过向 TIM_CCMR 寄存器中的 OCxM 位写入 100, 可将 OCxREF 信号强制设置为低电平。

无论如何, TIM_CCR 影子寄存器与计数器之间的比较仍会执行, 而且允许将标志置 1。因此可发送相应的中断请求。输出比较模式一节对此进行了介绍。

14.4.8 输出比较模式

此功能用于控制输出波形, 或指示已经过去某一时间段计数器状态。

当捕获/比较寄存器与计数器之间相匹配时, 输出比较功能:

- 将为相应的输出引脚分配一个可编程值, 该值由输出比较模式 (TIM_CCMR 寄存器中的 OCxM 位) 和输出极性 (TIM_CCER 寄存器中的 CCxP 位) 定义。匹配时, 输出引脚既可保持其电平 (OCxM=000), 也可设置为有效电平 (OCxM=001)、无效电平 (OCxM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1 (TIM_SR 寄存器中的 CCxIF 位)。
- 如果相应中断使能位 (TIM_IER 寄存器中的 CCxIE 位) 置 1, 将生成中断。

使用 TIM_CCMR 寄存器中的 OCxPE 位, 可将 TIM_CCRx 寄存器配置为带或不带预装载寄存器。

在输出比较模式下, 更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲 (在单脉冲模式下)。

步骤:

1. 选择计数器时钟 (内部、外部、预分频器)。
2. 在 TIM_ARR 和 TIM_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求, 将 CCxIE 位置 1。
4. 选择输出模式。例如, 当 CNT 与 CCRx 匹配、未使用预装载 CCRx 并且 OCx 使能且为高电平有效时, 必须写入 OCxM=011、OCxPE=0、CCxP=0 和 CCxE=1 来翻转 OCx 输出引脚。
5. 通过将 TIM_CR0 中的 TIM_CEN 位置 1 来使能计数器。

可随时通过软件更新 TIM_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器 (OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIM_CCRx 影子寄存器)。下图列出了相关示例。

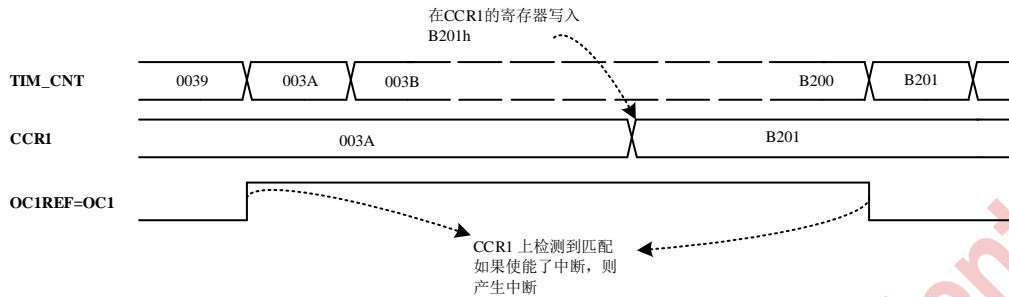


图 92 输出比较模式 (翻转 OC1)

14.4.9 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIM_ARR 寄存器值决定，其占空比则由 TIM_CCRx 寄存器值决定。

通过向 TIM_CCMR 寄存器中的 OCxM 位写入 0110 (PWM 模式 0) 或 0111 (PWM 模式 1)，可以独立选择各通道 (每个 OCx 输出对应一个 PWM) 的 PWM 模式。必须通过将 TIM_CCMR 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIM_CR0 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器。

由于只有在发生更新事件时预装载寄存器才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIM_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可使用 TIM_CCER 寄存器的 CCxP 位来编程。既可以设为高电平有效，也可以设为低电平有效。OCx 输出通过将 TIM_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 TIM_CCER 寄存器说明。

在 PWM 模式 0 或 1 下，TIM_CNT 始终与 TIM_CCRx 进行比较，以确定是 $TIM_CCR_x < TIM_CNT$ 还是 $TIM_CNT \leq TIM_CCR_x$ (取决于计数器计数方向)。

根据 TIM_CR0 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

14.4.9.1 PWM 边沿对齐模式

递增计数配置

当 TIM_CR0 寄存器中的 DIR 位为低时执行递增计数。请参见递增计数模式一节。

我们以 PWM 模式 0 为例。只要 $TIM_CNT < TIM_CCR_x$ ，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIM_CCRx 中的比较值大于自动重载值 (TIM_ARR 中)，则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。下图举例介绍 PWM 模式 0 波形 (TIM_ARR=8)。

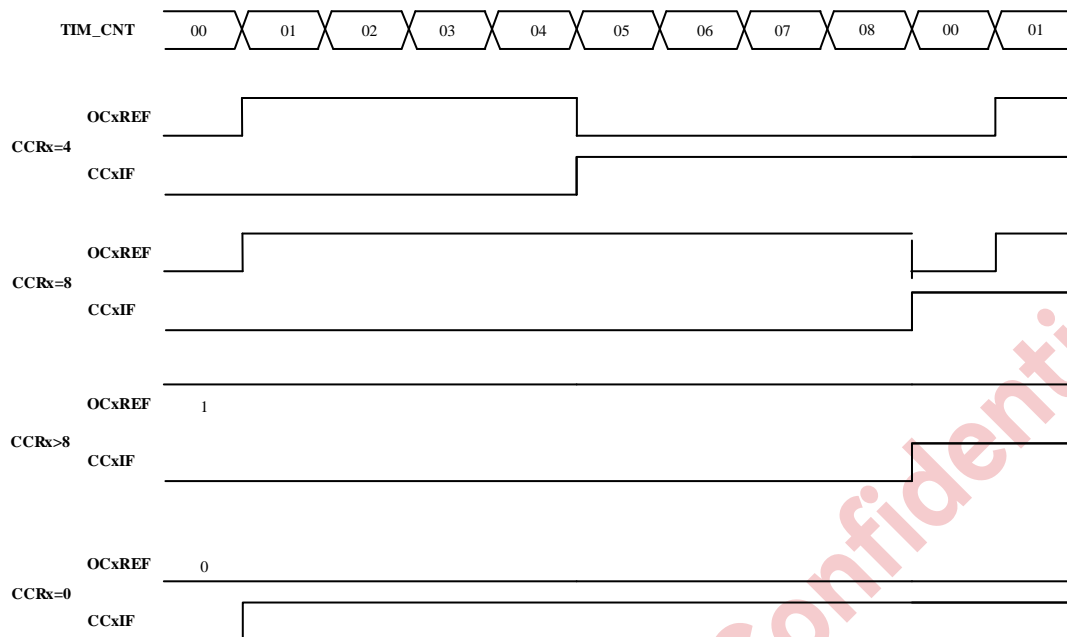


图 93 边沿对齐模式的 PWM 波形 (ARR=8)

递减计数配置

当 TIM_CR0 寄存器中的 DIR 位为高时执行递减计数。请参见递减计数模式一节。

在 PWM 模式 1 下,只要 $TIM_CNT < TIM_CCR_x$,参考信号 OCxREF 即为低电平,否则其为高电平。如果 TIM_CCR_x 中的比较值大于 TIM_ARR 中的自动重载值,则 OCxREF 保持为“1”。此模式下不可能产生 0%的 PWM 波形。

14.4.9.2 PWM 中心对齐模式

当 TIM_CR0 寄存器中的 CMS 位不为“00”(其余所有配置对 OCxREF/OCx 信号具有相同的作用),中心对齐模式生效。根据 CMS 位的配置,可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIM_CR0 寄存器中的方向位 (DIR) 由硬件更新,不得通过软件更改。请参见中心对齐模式 (递增/递减计数)。

下图显示了中心对齐模式的 PWM 波形,在此例中:

- TIM_ARR=8
- PWM 模式为 PWM 模式 0
- 在根据 TIM_CR0 寄存器中 CMS=01 而选择的中心对齐模式 0 下,当计数器递减计数时,比较标志置 1

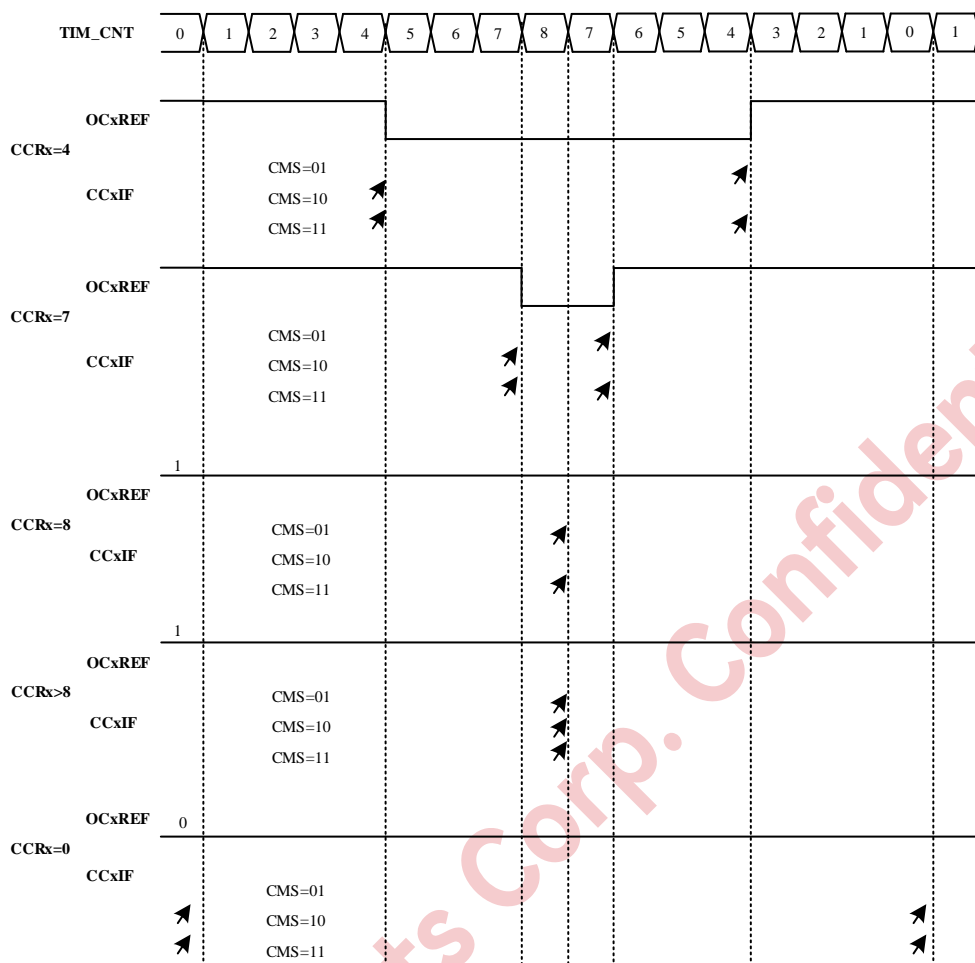


图 94 中心对齐模式 PWM 波形 (ARR=8)

14.4.9.3 中心对齐模式使用建议

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIM_CR0 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：
 - 如果写入计数器中的值大于自动重载值 (TIM_CNT>TIM_ARR)，计数方向不会更新。例如如果计数器之前递增计数，则继续递增计数。
 - 如果向计数器写入 0 或 TIM_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIM_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

14.4.10 发生外部事件时清除 OCxREF 信号

对于给定通道，在 ETRF 输入施加高电平（相应 TIM_CCMRx 寄存器中的 OCxCE 使能位置“1”），可使 OCxREF 信号变为低电平。OCxREF 信号将保持低电平，直到发生下一更新事件（UEV）。

此功能仅能用于输出比较模式和 PWM 模式，而不适用于强制输出模式。

例如，ETR 信号可以连接到比较器的输出，用于控制电流。此时，ETR 必须如下配置：

1. 必须关闭外部触发预分频器：TIM_SMCR 寄存器中的 ETPS[1:0]位置“00”。
2. 必须禁止外部时钟模式 1：TIM_SMCR 寄存器中的 ECE 位置“0”。
3. 外部触发极性 (ETP)和外部触发滤波器 (ETF)可根据用户需要进行配置。

下图对比了使能位 OCxCE 在不同值下的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 TIM 编程为 PWM 模式。

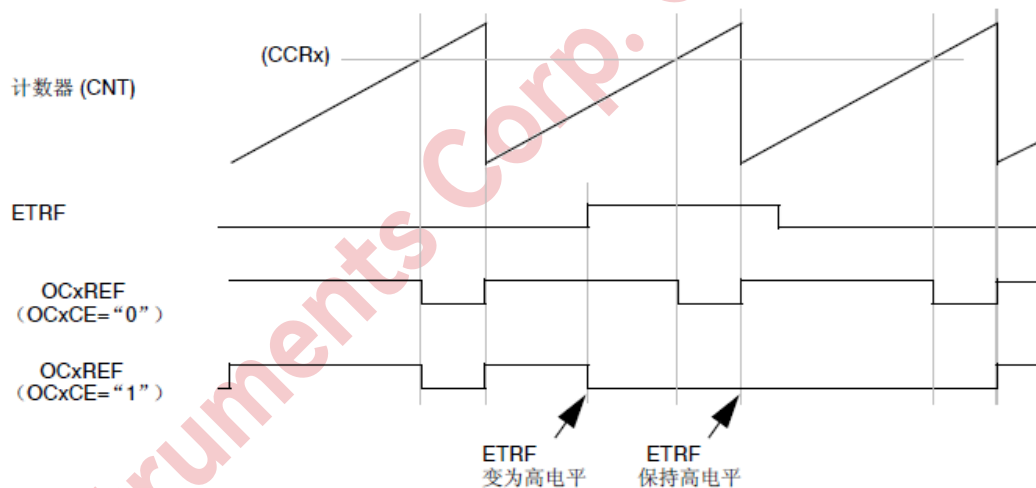


图 95 清除 TIM 的 OCxREF

如果 PWM 的占空比为 100% ($CCR_x > ARR$)，则下次计数器溢出时会再次使能 OCxREF。

14.4.11 单脉冲模式

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

可通过从模式控制器启动计数器并在输出比较模式或 PWM 模式下生成波形。将 TIM_CR0 寄存器中的 OPM 位置 1，即可选择单脉冲模式。在发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- 递增计数模式下， $CNT < CCRx \leq ARR$ (特别注意， $0 < CCRx$)
- 递减计数模式下， $CNT > CCRx$

例如，用户希望达到这样的效果：在 TI1 输入引脚检测到正沿时，经过 tDELAY 的延迟，在 OC1 上产生一个长度为 tPULSE 的正脉冲。使用 TI1FP1 作为触发：

1. 在 TIM_CCMR0 寄存器中写入 CC1S=01，将 TI1FP1 映射到 TI1；
2. 在 TIM_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，使 TI1FP1 能够检测上升沿；
3. 在 TIM_SMCR 寄存器中写入 TS=110，将 TI1FPT 配置为从模式控制器的触发 (TRGI)；
4. 在 TIM_SMCR 寄存器中写入 SMS=110 (触发模式)，使用 TI1FPT 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- tDELAY 由写入 TIM_CCR0 寄存器的值定义。
- tPULSE 由自动重载值与比较值 (TIM_ARR-TIM_CCR0) 之差来定义。
- 欲生成如此波形：信号在发生比较匹配时从 0 变为 1，在计数器达到自动重载值时由 1 变为 0。为此，应在 TIM_CCMR 寄存器中写入 OC0M=111，以使能 PWM 模式 1。如果需要，可选择在 TIM_CCMR 寄存器的 OC0PE 和 TIM_CR0 寄存器的 ARPE 中写入 1，以使能预装载寄存器。这种情况下，必须在 TIM_CCR0 寄存器中写入比较值并在 TIM_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI1 上的外部触发事件。本例中，CC1P 的值为 0。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIM_CR0 寄存器的 OPM 位写入 1，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIM_CR0 寄存器中的 OPM 位设置为 0 时，即选择重复模式。

在单脉冲模式下，TIM_Tl_x 输入的边沿检测会将 TIM_CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟 (t_{DELAY} 最小值)。

如果要输出延迟时间最短的波形，可以将 TIM_CCMR 寄存器中的 OCxFE 位置 1。这样会强制 OCxREF (和 OCx) 对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM 模式 0 或 PWM 模式 1 时，OCxFE 才会起作用。

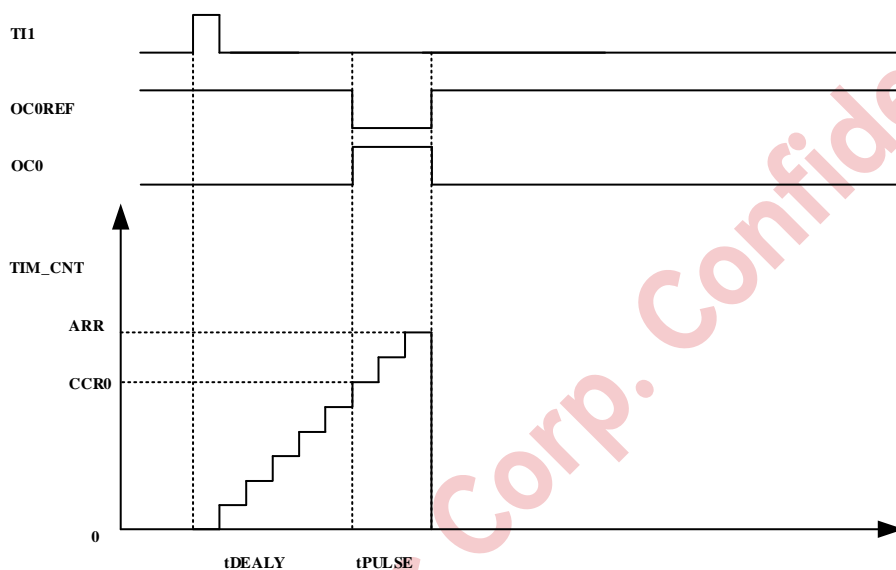


图 96 单脉冲模式示例

14.4.12 编码器接口模式

编码器支持 6 种模式，依据 TIM_SMCR 寄存器中 SMS 控制位进行选择。

14.4.12.1 编码器模式 1

表 189 编码器模式 1 下计数器方向与编码信号关系

有效边沿	对应电平 TI0	TI0 信号		TI1 信号	
		上升	下降	上升	下降
仅在 TI1 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增

通过编程 TIM_CCER 寄存器的 CC0P 和 CC1P 位，选择 TI0 和 TI1 极性。如果需要，还可对输入滤波器进行编程。CC0NP 和 CC1NP 必须保持低电平。下图示例展示了在 CC0P=0、CC1P=0 和输入滤波未启用的情况下编码工作方式。

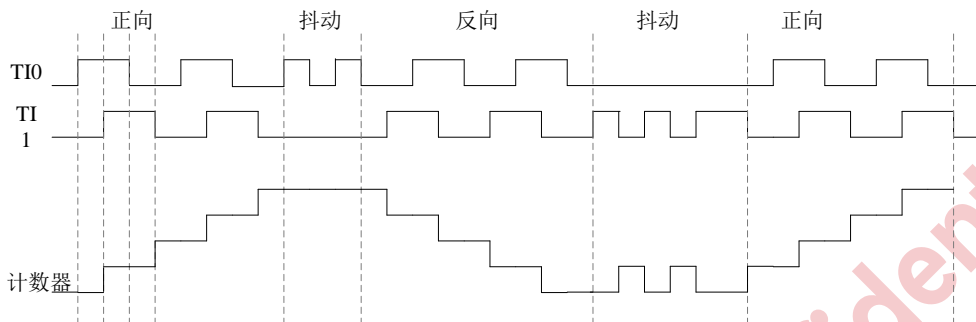


图 97 编码器模式 1 计数示例

14.4.12.2 编码器模式 2

当 SMS=4'b0010 时，编码器模式 2 启用，内部计数器根据 TI1 电平状态在 TI0 跳变边沿进行递增/递减计数，具体对应关系如下表所示。

表 190 编码器模式 2 下计数器方向与编码信号关系

有效边沿	对应电平 TI1	TI0 信号		TI1 信号	
		上升	下降	上升	下降
仅在 TI0 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数

通过编程 TIM_CCER 寄存器的 CC0P 和 CC1P 位，选择 TI0 和 TI1 极性。如果需要，还可对输入滤波器进行编程。CC0NP 和 CC1NP 必须保持低电平。下图示例展示了在 CC0P=0、CC1P=0 和输入滤波未启用的情况下编码工作方式。

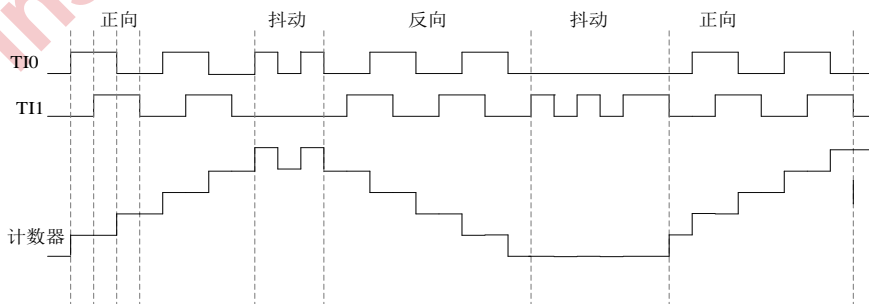


图 98 编码器模式 2 计数示例

14.4.12.3 编码器模式 3

当 SMS=4'b0011 时，编码器模式 3 启用，计数器在 TI0 和 TI1 的边沿计数，计数的方向取决于另外一个信号的电平，具体对应关系如下表所示。

表 191 编码器模式 3 下计数器方向与编码信号关系

有效边沿	TI0 对于 TI1 TI1 对应 TI0	TI0 信号		TI1 信号	
		上升	下降	上升	下降
在 TI0 和 TI1 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

通过编程 TIM_CCER 寄存器的 CC0P 和 CC1P 位，选择 TI0 和 TI1 极性。如果需要，还可对输入滤波器进行编程。CC0NP 和 CC1NP 必须保持低电平。下图示例展示了在 CC0P=0、CC1P=0 和输入滤波未启用的情况下编码工作方式。

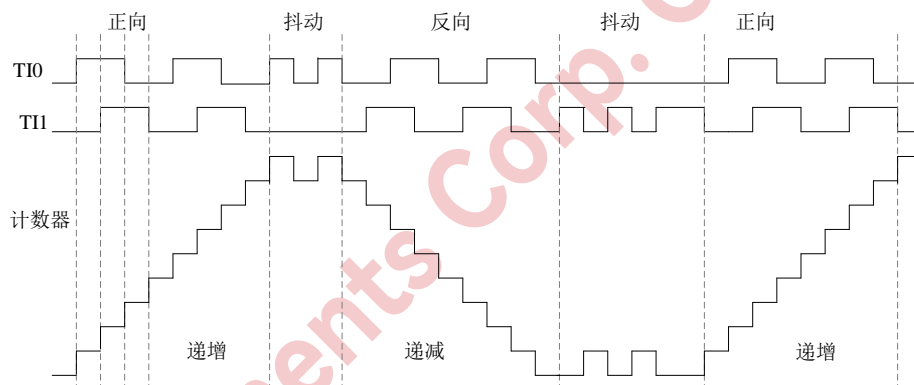


图 99 编码器模式 3 计数示例

14.4.12.4 编码器模式 4

当 SMS=4'b1000 时，编码器模式 4 启用，计数器根据 TI1 电平作为计数器计数方向，以 TI0 边沿进行递增/递减计数，具体对应关系如下表所示。

表 192 编码器模式 4 下计数器方向与编码信号关系

有效边沿	对应电平 TI1	TI0 信号		TI1 信号	
		上升	下降	上升	下降
仅在 TI0 处计数	高	递增	递增	不计数	不计数
	低	递减	递减	不计数	不计数

通过编程 TIM_CCER 寄存器的 CC0P 和 CC1P 位，选择 TI0 和 TI1 极性。如果需要，还可对输入滤波器进行编程。CC0NP 和 CC1NP 必须保持低电平。控制位 QEES 用于选择 TI0 的有效计数沿，此模式下支持上升沿、下降沿和双沿三种选择，下面的三幅图分别展示了对应有效计数沿下编码器计数。

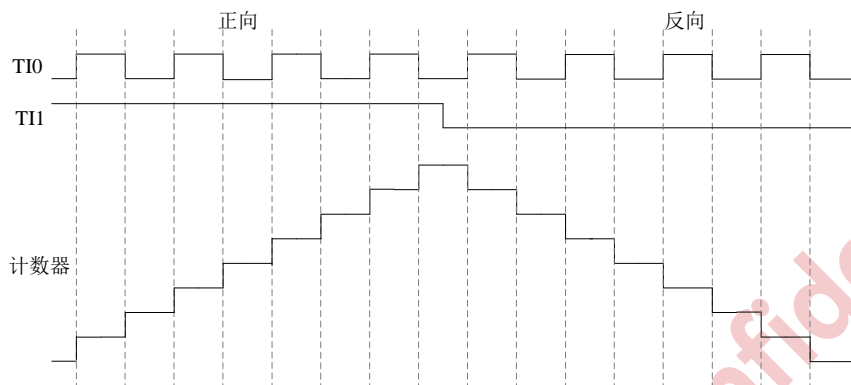


图 100 编码器模式 4 示例 (双沿有效)

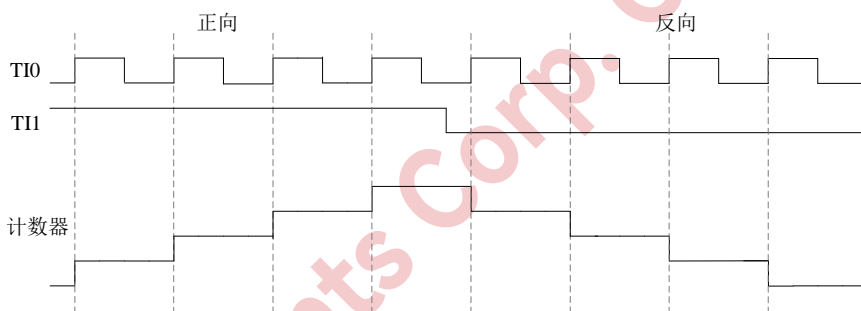


图 101 编码器模式 4 计数示例 (上升沿有效)

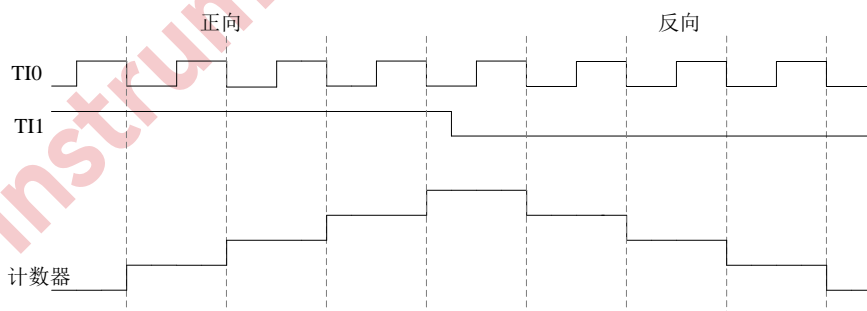


图 102 编码器模式 4 计数示例 (下降沿有效)

14.4.12.5 编码器模式 5

当 SMS=4'b1001 时，编码器模式 5 启用，计数器在 TI0 边沿进行递增计数，而在 TI1 的边沿进行递减计数，具体对应关系如下表所示。

表 193 编码器模式 5 下计数器方向与编码信号关系

有效边沿	对应电平	TI0 信号		TI1 信号	
		上升	下降	上升	下降
在 TI0 和 TI1 处均计数	-	递增	递增	递减	递减

通过编程 TIM_CCER 寄存器的 CC0P 和 CC1P 位，选择 TI0 和 TI1 极性。如果需要，还可对输入滤波器进行编程。CC0NP 和 CC1NP 必须保持低电平。控制位 QEES 用于选择 TI0 和 TI1 的有效计数沿，此模式下支持上升沿、下降沿和双沿三种选择，下面的三幅图分别展示了对应有效计数沿下编码器计数。

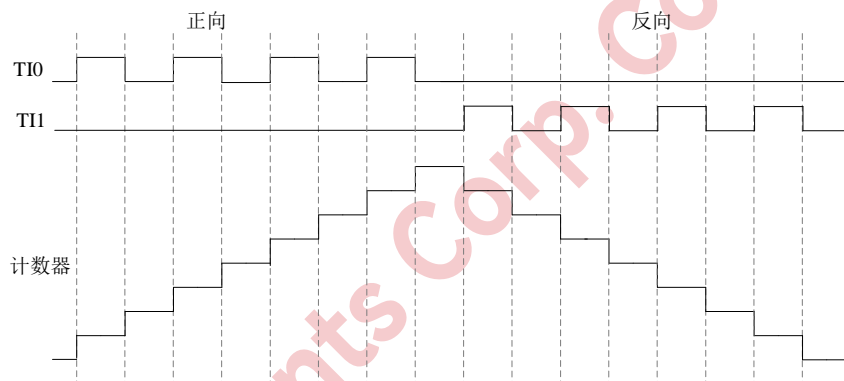


图 103 编码器模式 5 示例 (双沿有效)

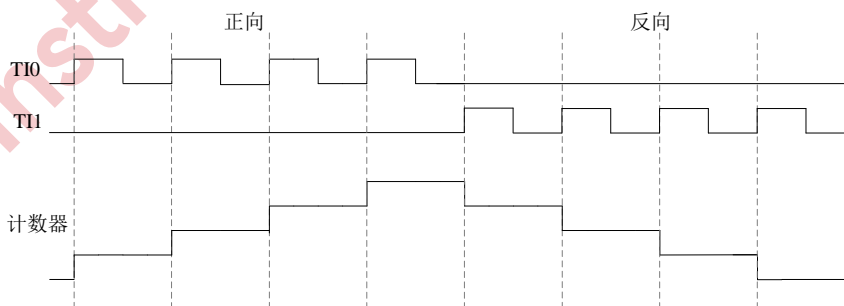


图 104 编码器模式 5 计数示例 (上升沿有效)

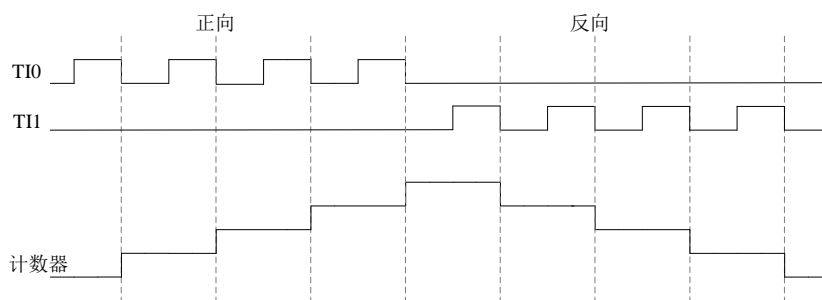


图 105 编码器模式 5 计数示例（下降沿有效）

14.4.12.6 编码器模式 6

当 SMS=4'b1010 时，编码器模式 6 启用，计数器在 TI0 和 TI1 的边沿进行递增计数，具体对应关系如下表所示。计数方向 DIR 始终为正向递增。

表 194 编码器模式 6 下计数器方向与编码信号关系

有效边沿	对应电平	TI0 信号		TI1 信号	
		上升	下降	上升	下降
在 TI0 和 TI1 处均计数	-	递增	递增	递增	递增

通过编程 TIM_CCER 寄存器的 CC0P 和 CC1P 位，选择 TI0 和 TI1 极性。如果需要，还可对输入滤波器进行编程。CC0NP 和 CC1NP 必须保持低电平。控制位 QEES 用于选择 TI0 和 TI1 的有效计数沿，此模式下支持上升沿、下降沿和双沿三种选择，下面的三幅图分别展示了对应有效计数沿下编码器计数。

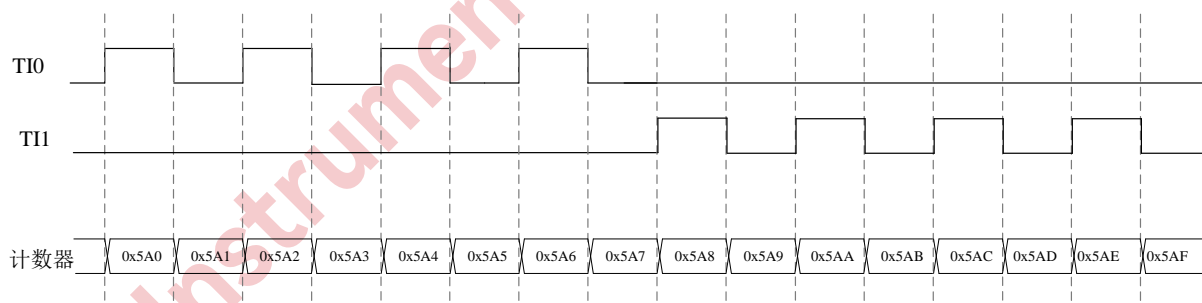


图 106 编码器模式 6 示例（双沿有效）

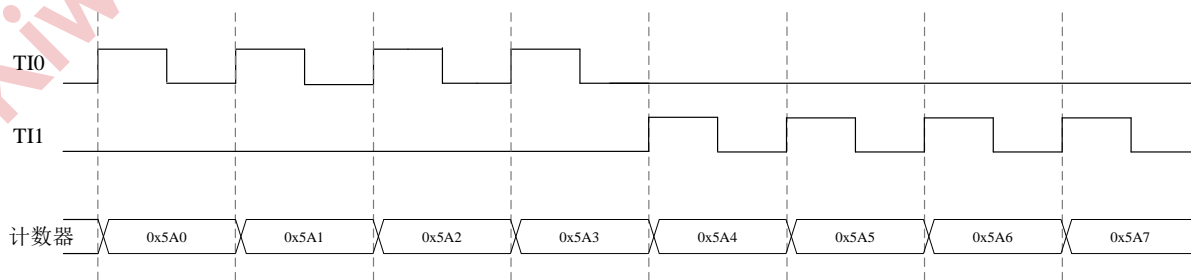


图 107 编码器模式 6 计数示例（上升沿有效）

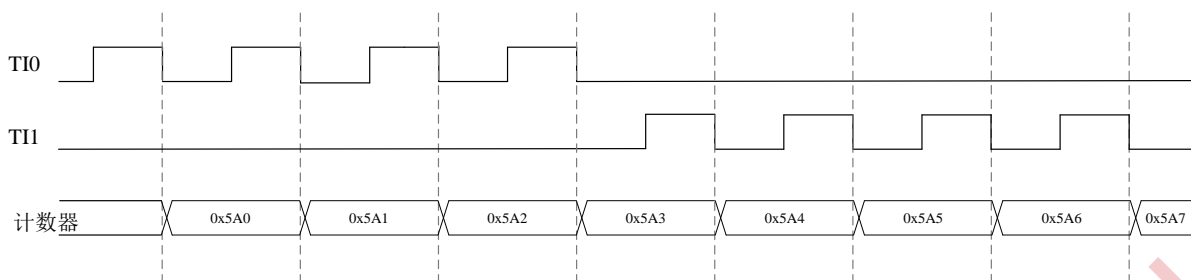


图 108 编码器模式 6 计数示例（下降沿有效）

定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）；还可以通过由实时时钟生成的 DMA 请求读取计数器值。

14.4.13 定时器输入异或功能

通过 TIM_CR0 寄存器中的 TI0S 位，可将通道 0 的输入滤波器连接到异或门的输出，从而将 TIM_CH0 到 TIM_CH2 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。高级定时器以连接霍尔传感器为例介绍了此功能。

14.4.14 定时器与外部触发同步

定时器可与外部触发以下列模式实现同步：复位模式、门控模式和触发模式。

14.4.14.1 复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIM_CR0 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIM_ARR 和 TIM_CCRx）都将更新。

在下例中，TI1 输入上出现上升沿时，计数器清零：

1. 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中只需要默认采样滤波，故保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIM_CCMR 寄存器中的 CC1S=01。在 TIM_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，以验证极性（仅检测上升沿）。
2. 在 TIM_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIM_SMCR 寄存器中写入 TS=110，选择 TI1 作为输入源。
3. 在 TIM_CR0 寄存器中写入 TIM_CEN=1，启动计数器。

计数器使用内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时触发标志 (TIM_SR 寄存器中的 TIF 位) 置 1，使能中断后，还可发送中断请求 (如果 TIE 使能)。

下图显示了自动重载寄存器 TIM_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

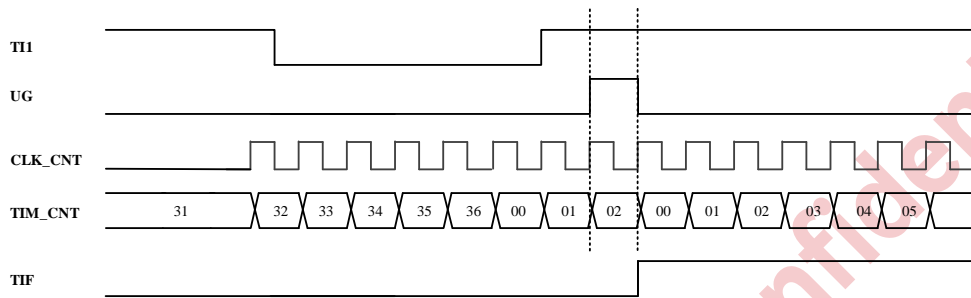


图 109 复位模式下的控制电路

14.4.14.2 门控模式

输入信号的电平可用来使能计数器。在以下示例中，计数器仅在 TI1 输入为低电平时计数。

1. 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间 (本例中不需要任何滤波, 因此保持 IC1F=0000)。由于捕获预分频器不用于触发操作, 因此无需对其进行配置。CC1S 位只选择输入捕获源, 即 TIM_CCMR 寄存器中的 CC1S=01。在 TIM_CCER 寄存器中写入 CC1P=1, 以确定极性 (仅检测低电平)。
2. 在 TIM_SMCR 寄存器中写入 SMS=101, 将定时器配置为门控模式。在 TIM_MCR 寄存器中写入 TS=110, 选择 TI1 作为输入源。
3. 在 TIM_CR0 寄存器中写入 TIM_CEN=1, 使能计数器 (在门控模式下, 如果 CEN=0, 则无论触发输入电平如何, 计数器都不启动)。

只要 TI1 为低电平, 计数器就开始根据内部时钟计数, 直到 TI1 变为高电平时停止计数。计数器启动或停止时, TIM_SR 寄存器中的 TIF 标志都会置 1。TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

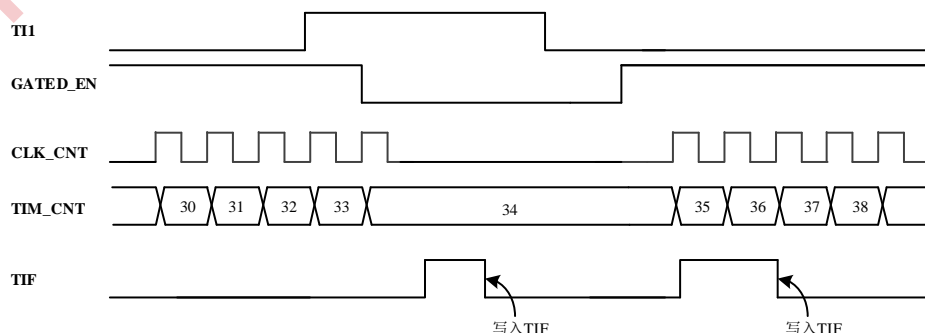


图 110 门控模式下的控制电路

14.4.14.3 触发模式

所选输入上发生某一事件时可以用以启动计数器。以下示例中，TI2 输入上出现上升沿时，计数器便启动：

1. 将通道 1 配置为检测 TI1 上的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIM_CCMR0 寄存器中的 CC1S=01。在 TIM_CCER 中写入 CC1P=1，以确定极性（仅检测低电平）。
2. 在 TIM_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIM_SMCR 寄存器中写入 TS=110，选择 TI1 作为输入源。

当 TI1 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。TI1 的上升沿与实际计数器启动之间的延迟是由于 TI1 输入的重新同步电路引起的。

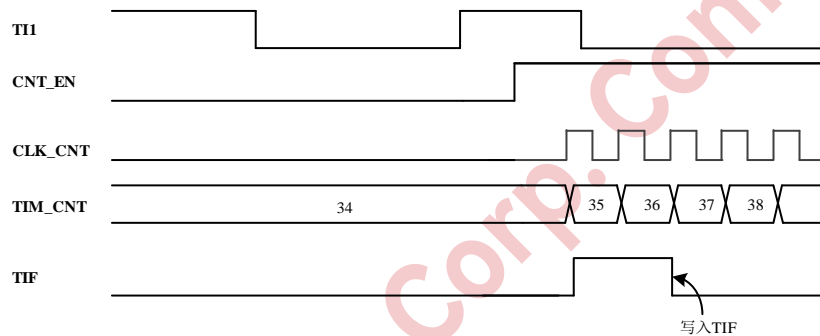


图 111 触发模式下的控制电路

14.4.15 定时器同步

定时器从内部连接在一起，以实现定时器同步或级联。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

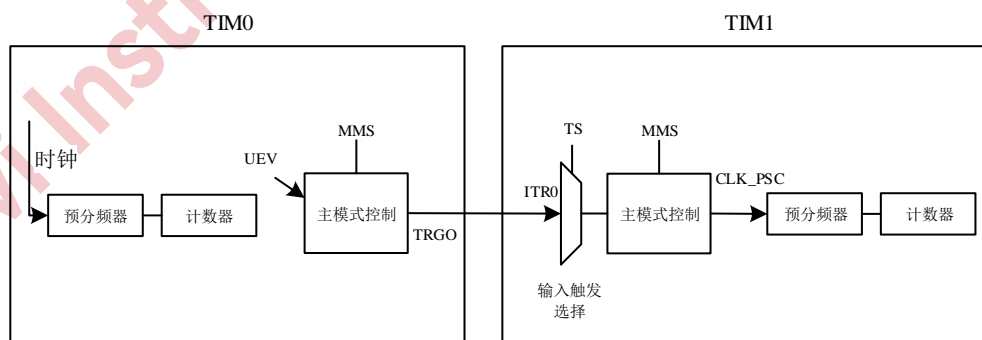


图 112 主/从定时器示例

定时器同步连接关系如下表所示。

表 195 定时器同步连接

从定时器	ITR0	ITR1	ITR2	ITR3
TIM1	TIM15_TRGO	-	TIM3_TRGO	TIM17_TRGO
TIM3	TIM1_TRGO	-	TIM14_TRGO	TIM15_TRGO
TIM14	-	-	-	-
TIM15	TIM3_TRGO	-	TIM16_TRGO	TIM17_TRGO
TIM16	-	-	-	-
TIM17	-	-	-	-

具体可参照简版定时器 TIMS 的定时器同步章节。

14.4.16 TIMG 中断号

TIM 的中断连接至不可屏蔽中断处理模块 (NVIC) 的 IRQ 相关接口。为了启用该中断功能，需要按照以下流程进行操作：

- 使能 TIM 的中断使能位
- 使能 NVIC 中的 TIM IRQ 通道, TIM 的系统中断号如下表所示.
- 配置 TIM 以产生中断

表 196 TIMG 中断表

名称	中断号
TIMG0_INT	35
TIMG1_INT	36
TIMG2_INT	37

14.5 寄存器

TIMG0~2 寄存器基地址分别是 0x40041000、0x40041400 和 0x40041800。下表为 TIMGx 的各控制寄存器描述。

表 197 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	TIM 控制寄存器 0	32	r/w	0x00000000
0x04	TIM 控制寄存器 1	32	r/w	0x00000000
0x08	TIM 从属模式控制寄存器	32	r/w	0x00000000
0x0C	TIM DMA&中断使能寄存器	32	r/w	0x00000000
0x10	TIM 状态寄存器	32	r/w	0x00000000
0x14	TIM 事件生成寄存器	32	r/w	0x00000000
0x18	TIM 捕获/比较模式寄存器 0	32	r/w	0x00000000
0x1C	TIM 捕获/比较模式寄存器 1	32	r/w	0x00000000
0x20	TIM 捕获/比较使能寄存器	32	r/w	0x00000000
0x24	TIM 计数器	32	r/w	0x00000000
0x28	TIM 预分频器	32	r/w	0x00000000
0x2C	TIM 自动重载寄存器	32	r/w	0x0000FFFF
0x30	Reserved	32	res	0x00000000
0x34	TIM 捕获/比较寄存器 0	32	r/w	0x00000000
0x38	TIM 捕获/比较寄存器 1	32	r/w	0x00000000
0x3C	TIM 捕获/比较寄存器 2	32	r/w	0x00000000
0x40	TIM 捕获/比较寄存器 3	32	r/w	0x00000000
0x44~0x60	Reserved	32	res	0x00000000
0x64	TIM 配置秘钥寄存器	32	r/w	0x00000000
0x68~0xFC	Reserved	32	res	0x00000000

关于输入复用关系、系统刹车控制和外部输入刹车控制的信息请查阅设计文档《简易定时器》中相关寄存器描述。

14.5.1 TIM 控制寄存器 0 (TIM_CR0)

偏移地址：0x0000

表 198 TIM 控制寄存器 0

位	访问	描述
31:11	Res	Reserved 复位值：0x0
10	r/w	CNT_ST：计数器停止运行标记 0：计数器尚在计数 1：计数器停止计数 复位值：0x0
9:8	r/w	CKD：时钟分频 此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器 (Tl _x) 所使用的死区及采样时钟 (t _{DTS}) 之间的分频比。 00: t _{DTS} = t _{CK_INT} 01: t _{DTS} = 2*t _{CK_INT} 10: t _{DTS} = 4*t _{CK_INT} 11: t _{DTS} = t _{CK_INT} 复位值：0x0
7	r/w	ARPE：自动重载预装载使能 0：TIM_ARR 寄存器不进行缓存 1：TIM_ARR 寄存器进行缓存 复位值：0x0
6:5	r/w	CMS[1:0]：中心对齐模式选择 00：边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。 01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道 (TIM_CCMRx 寄存器中的 CxS=00) 的输出比较中断标志才置 1。 10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道 (TIM_CCMRx 寄存器中的 CxS=00) 的输出比较中断标志才置 1。 11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道 (TIM_CCMRx 寄存器中的 CxS=00) 的输出比较中断标志都会置 1。 注： 只要计数器处于使能状态 (CEN=1)，就不得从边沿对齐模式切换为中心 复位值：0x0
4	r/w	DIR：方向 0：计数器递增计数 1：计数器递减计数 注：当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。 复位值：0x0

位	访问	描述
3	r/w	<p>OPM: 单脉冲模式使能控制</p> <p>0: 计数器在发生更新事件时不会停止计数</p> <p>1: 计数器在发生下一更新事件时停止计数 (将 CEN 位清零)</p> <p>复位值: 0x0</p>
2	r/w	<p>URS: 更新请求源选择</p> <p>0: 使能时, 所有以下事件都会生成更新中断请求。此类事件包括:</p> <ul style="list-style-type: none"> 计数器上溢/下溢 将 UG 位置 1 通过从模式控制器生成的更新事件 <p>1: 使能时, 只有计数器上溢/下溢会生成更新中断请求。</p> <p>复位值: 0x0</p>
1	r/w	<p>UDIS: 更新禁止</p> <p>此位由软件置 1 和清零, 用以使能/禁止 UEV 事件生成。</p> <p>0: 使能 UEV。更新(UEV)事件可通过以下事件之一生成:</p> <ul style="list-style-type: none"> 计数器上溢/下溢 将 UG 位置 1 通过从模式控制器生成的更新事件 然后更新影子寄存器的值。 <p>1: 禁止 UEV。不生成更新事件, 各影子寄存器的值 (ARR、PSC 和 CCRx) 保持不变。但如果将 UG 位置 1, 或者从从模式控制器接收到硬件复位, 则会重新初始化计数器和预分频器。</p> <p>复位值: 0x0</p>
0	r/w	<p>CEN: 计数器使能控制</p> <p>0: 关闭</p> <p>1: 开启</p> <p>注意: 只有事先通过软件将 CEN 位置 1, 才可以使用外部时钟、门控模式。而触发模式可通过硬件自动将 CEN 位置 1。</p> <p>复位值: 0x0</p>

14.5.2 TIM 控制寄存器 1 (TIM_CR1)

偏移地址: 0x0004

表 199 TIM 控制寄存器 1

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	r/w	TIOS:TI0 选择 (TI1 Selection) 0: TIM_CH0 引脚连接到 TI0 输入 1: TIM_CH0、CH1 和 CH2 引脚连接到 TI0 输入 (异或组合) 复位值: 0x0
6:4	r/w	MMS[1:0]: 主模式选择 (Master Mode Selection) 这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下: 000: 复位--TIM_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成 (从模式控制器配置为复位模式), 则 TRGO 上的信号相比实际复位会有延迟。 001: 使能--计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器, 或者控制在一段时间内使能从定时器。计数器使能信号可由 CEN 控制位产生。当配置为门控模式时, 也可由触发输入产生。当计数器使能信号由触发输入控制时, TRGO 上会存在延迟, 选择主/从模式时除外 (请参见 TIM_SMCR 寄存器中 MSM 位的说明)。 010: 更新--选择更新事件作为触发输出 (TRGO)。例如, 主定时器可用作从定时器的预分频器。 011: 比较脉冲--一旦发生输入捕获或比较匹配事件, 当 CC0IF 被置 1 时 (即使已为高电平), 触发输出都会发送一个正脉冲 (TRGO)。 100: 比较--OC0REF 信号用作触发输出 (TRGO) 101: 比较--OC1REF 信号用作触发输出 (TRGO) 110: 比较--OC2REF 信号用作触发输出 (TRGO) 111: 比较--OC3REF 信号用作触发输出 (TRGO) 复位值: 0x0
3	r/w	CCDS: 捕获/比较 DMA 选择 (Capture/Compare DMA Selection) 0: 发生 CCx 事件时发送 CCxDMA 请求 1: 发生更新事件时发送 CCxDMA 请求 复位值: 0x0
2:0	Res	Reserved 复位值: 0x0

14.5.3 TIM 从属模式控制寄存器 (TIM_SMCR)

偏移地址: 0x0008

表 200 TIM 从属模式控制寄存器

位	访问	描述
31:30	Res	Reserved 复位值: 0x0
29:28	r/w	QEES: 编码模式下有效时钟沿选择 00: 上升沿 01: 下降沿 1X: 上升沿或下降沿 注: 该控制位仅对编码模式 4~6 有意义 复位值: 0x0
27	Res	Reserved 复位值: 0x0
26:24	r/w	QEM, 编码模式 000: 编码器模式关闭。 001: 编码器模式 1--计数器根据 TI0FPT 电平在 TI1FPT 边沿递增/递减计数。 010: 编码器模式 2--计数器根据 TI1FPT 电平在 TI0FPT 边沿递增/递减计数。 011: 编码器模式 3--计数器在 TI0FPT 和 TI1FPT 的边沿计数, 计数的方向取决于另外一个信号的电平。 100: 编码器模式 4--计数器根据 TI1FPT 电平作为计数器计数方向, 以 TI0FPT 边沿进行递增/递减计数。 101: 编码器模式 5--计数器在 TI0FPT 边沿进行递增计数, 而在 TI1FPT 的边沿进行递减计数 110: 编码器模式 6--计数器在 TI0FPT 和 TI1FPT 的边沿进行递增计数 其余值: 保留 注意: 对于编码器模式 4~5, 需要配合控制位 QEES 选择有效计数沿 复位值: 0x0
23:21	Res	Reserved 复位值: 0x0
20	r/w	OCCS: OCREF 清除源选择 0: OCREF_CLR_INT 源于 OCREF_CLR 1: OCREF_CLR_INT 源于 ETRF 复位值: 0x0
19:17	Res	Reserved 复位值: 0x0
16	r/w	ETEN: 外部触发关闭使能 0: 关闭 1: 开启 复位值: 0x0
15	r/w	ETP: 外部触发极性

位	访问	描述
		<p>此位可选择将 ETR 还是 ETR 用于触发操作</p> <p>0: ETR 未反相, 高电平或上升沿有效。</p> <p>1: ETR 反相, 低电平或下降沿有效。</p> <p>复位值: 0x0</p>
14	r/w	<p>ECE: 外部时钟使能</p> <p>此位可使能外部时钟模式 1。</p> <p>0: 禁止外部时钟模式 1</p> <p>1: 使能外部时钟模式 1。计数器时钟由 ETRF 信号的任意有效边沿提供。</p> <p>注:</p> <p>1: 将 ECE 位置 1 与选择外部时钟模式 0 并将 TRGI 连接到 ETRF (SMS=111 且 TS=111) 具有相同效果。</p> <p>2: 外部时钟模式 1 可以和以下从模式同时使用: 复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF (TS 位不得为 111)。</p> <p>3: 如果同时使能外部时钟模式 0 和外部时钟模式 1, 则外部时钟输入为 ETRF。</p> <p>复位值: 0x0</p>
13:12	r/w	<p>ETPS[1:0]: 外部触发预分频器</p> <p>外部触发信号 ETRP 频率不得超过 TIM CLK 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。</p> <p>00: 预分频器关闭</p> <p>01: 2 分频 ETRP 频率</p> <p>10: 4 分频 ETRP 频率</p> <p>11: 8 分频 ETRP 频率</p> <p>复位值: 0x0</p>
11:8	r/w	<p>ETF[3:0]: 外部触发滤波器</p> <p>此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波时间。数字滤波器由事件计数器组成, 每 N 个事件才视为一个有效边沿:</p> <p>0000: 无滤波器, 按 f_{DTS} 频率进行采样</p> <p>0001: $f_{SAMPLING}=f_{CK_INT}$, $N=2$</p> <p>0010: $f_{SAMPLING}=f_{CK_INT}$, $N=4$</p> <p>0011: $f_{SAMPLING}=f_{CK_INT}$, $N=8$</p> <p>0100: $f_{SAMPLING}=f_{DTS}/2$, $N=6$</p> <p>0101: $f_{SAMPLING}=f_{DTS}/2$, $N=8$</p> <p>0110: $f_{SAMPLING}=f_{DTS}/4$, $N=6$</p> <p>0111: $f_{SAMPLING}=f_{DTS}/4$, $N=8$</p> <p>1000: $f_{SAMPLING}=f_{DTS}/8$, $N=6$</p> <p>1001: $f_{SAMPLING}=f_{DTS}/8$, $N=8$</p> <p>1010: $f_{SAMPLING}=f_{DTS}/16$, $N=5$</p> <p>1011: $f_{SAMPLING}=f_{DTS}/16$, $N=6$</p> <p>1100: $f_{SAMPLING}=f_{DTS}/16$, $N=8$</p> <p>1101: $f_{SAMPLING}=f_{DTS}/32$, $N=5$</p> <p>1110: $f_{SAMPLING}=f_{DTS}/32$, $N=6$</p>

位	访问	描述
		1111: $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, $N=8$ 复位值: 0x0
7	r/w	MSM: 主/从模式 0: 不执行任何操作 1: 当前定时器的触发输入事件 (TRGI) 的动作被推迟, 以使当前定时器与其从定时器实现完美同步 (通过 TRGO)。此设置适用于由单个外部事件对多个定时器进行同步的情况。 复位值: 0x0
6:4	r/w	TS[2:0]: 触发选择 此位域可选择将要用于同步计数器的触发输入。 000~011: 内部触发 (ITR) 100: TI1 边沿检测器 (TIOF_ED) 101: 滤波后的定时器输入 1 (TIOFPT) 110: 滤波后的定时器输入 2 (TI1FPT) 111: 外部触发输入 (ETRF) 有关各定时器 ITRx 含义的详细信息, 请参见表格: TIM 内部触发连接。 这些控制位也用于选择触发输入作为 TRC, 其定义如下, 000~011: 内部触发信号 (ITR) 100: TI1 边沿检测器 (TIOF_ED) 其他保留 注意: 这些位只能在未使用的情况下 (例如, SMS=000 时) 进行更改, 以避免转换时出现错误的边沿检测。 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:0	r/w	SMS: 从模式选择 选择外部信号时, 触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关 (请参见输入控制寄存器和控制寄存器说明)。 000: 禁止从模式--如果 CEN="1", 预分频器时钟直接由内部时钟提供。 001: 保留。 010: 保留。 011: 保留。 100: 复位模式--在出现所选触发输入 (TRGI) 上升沿时, 重新初始化计数器并生成一个寄存器更新事件。 101: 门控模式--触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平, 计数器立即停止计数 (但不复位)。计数器的启动和停止都是受控的。 110: 触发模式--触发信号 TRGI 出现上升沿时启动计数器 (但不复位)。只控制计数器的启动。 111: 外部时钟模式 0--由所选触发信号 (TRGI) 的上升沿提供计数器时钟。

位	访问	描述
		<p>注意：如果将 TI0F_ED 选作触发输入 (TS="100"), 则不得使用门控模式。实际上, TI0F 每次转换时, TI0F_ED 都输出 1 个脉冲, 而门控模式检查的则是触发信号的电平。</p> <p>复位值: 0x0</p>

14.5.4 TIM DMA&中断使能寄存器 (TIM_DIER)

偏移地址: 0x000C

表 201 TIM 中断使能寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14	r/w	<p>TDE: 触发 DMA 请求使能</p> <p>0: 关闭</p> <p>1: 开启</p> <p>复位值: 0x0</p>
13	Res	Reserved 复位值: 0x0
12	r/w	<p>CC3DE: 捕获/比较 3 DMA 请求使能</p> <p>0: 关闭</p> <p>1: 开启</p> <p>复位值: 0x0</p>
11	r/w	<p>CC2DE: 捕获/比较 2 DMA 请求使能</p> <p>0: 关闭</p> <p>1: 开启</p> <p>复位值: 0x0</p>
10	r/w	<p>CC1DE: 捕获/比较 1 DMA 请求使能</p> <p>0: 关闭</p> <p>1: 开启</p> <p>复位值: 0x0</p>
9	r/w	<p>CC0DE: 捕获/比较 0 DMA 请求使能</p> <p>0: 关闭</p> <p>1: 开启</p> <p>复位值: 0x0</p>
8	r/w	<p>UDE: 更新 DMA 请求使能</p> <p>0: 关闭</p> <p>1: 开启</p> <p>复位值: 0x0</p>
7	Res	Reserved 复位值: 0x0

位	访问	描述
6	r/w	TIE: 触发信号 (TGRI) 中断使能 0: 关闭 1: 开启 复位值: 0x0
5	Res	Reserved 复位值: 0x0
4	r/w	CC3IE: 捕获/比较 3 中断使能 0: 关闭 1: 开启 复位值: 0x0
3	r/w	CC2IE: 捕获/比较 2 中断使能 0: 关闭 1: 开启 复位值: 0x0
2	r/w	CC1IE: 捕获/比较 1 中断使能 0: 关闭 1: 开启 复位值: 0x0
1	r/w	CC0IE: 捕获/比较 0 中断使能 0: 关闭 1: 开启 复位值: 0x0
0	r/w	UIE: 更新中断使能 0: 关闭 1: 开启 复位值: 0x0

14.5.5 TIM 状态寄存器 (TIM_SR)

偏移地址: 0x0010

表 202 TIM 状态寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
12	r/w	CC3OF: 捕获/比较 3 重复捕获标志 请参照 CC1OF 说明 复位值: 0x0
11	r/w	CC2OF: 捕获/比较 2 重复捕获标志 请参照 CC1OF 说明 复位值: 0x0
10	r/w	CC1OF: 捕获/比较 1 重复捕获标志 请参照 CC1OF 说明

位	访问	描述
		复位值: 0x0
9	r/w	CC0OF: 捕获/比较 0 重复捕获标志 0: 未检测到重复捕获。 1: TIM_CCR0 寄存器中已捕获到计数器值且 CC0IF 标志已置 1。 仅当将相应通道配置为输入捕获模式时, 此标志位才会由硬件置 1。该位写 1 清零。 复位值: 0
8:7	Res	Reserved 复位值: 0x0
6	r/w	TIF: 触发中断标志 在除门控模式以外的所有模式下, 当使能从模式控制器后在 TRGI 输入上检测到有效边沿时, 该标志将由硬件置 1。选择门控模式时, 该标志将在计数器启动或停止时置 1。但需要通过软件写 1 清零。 0: 未发生触发事件。 1: 触发中断挂起。 复位值: 0x0
5	Res	Reserved 复位值: 0x0
4	r/w	CC3IF: 捕获/比较 3 中断标志 参照 CC1IF 描述 复位值: 0x0
3	r/w	CC2IF: 捕获/比较 2 中断标志 参照 CC0IF 描述 复位值: 0x0
2	r/w	CC1IF: 捕获/比较 1 中断标志 参照 CC0IF 描述 复位值: 0x0
1	r/w	CC0IF: 捕获/比较 0 中断标志 通道 CC1 配置为输出 当计数器与比较值匹配时, 此标志由硬件置 1, 软件写 1 清零。 0: 不匹配。 1: TIM_CNT 计数器的值与 TIM_CCR0 寄存器的值匹配。当 TIM_CCR0 的值大于 TIM_ARR 的值时, CC0IF 位将在计数器发生上溢时变为高电平。 通道 CC1 配置为输入 此位将在发生捕获事件时由硬件置 1。通过软件写 1 清零或读取 TIM_CCR0 寄存器将该位清零。 0: 未发生输入捕获事件 1: TIM_CCR0 寄存器中已捕获到计数器值 (IC0 上已检测到与所选极性匹配的边沿) 复位值: 0x0
0	r/w	UIF: 更新中断标志 该位在发生更新事件时通过硬件置 1。但需要通过软件写 1 清零。

位	访问	描述
		0: 未发生更新。 1: 更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1: <ul style="list-style-type: none"> TIM_CR 寄存器中的 URS=0 且 UDIS=0, 并且由软件使用 TIM_EGR 寄存器中的 UG 位重新初始化 CNT 时。 TIM_CR 寄存器中的 URS=0 且 UDIS=0, 并且 CNT 由触发事件重新初始化时 (请参见模式控制寄存器 (TIM_MCR))。 复位值: 0x0

14.5.6 TIM 事件生成寄存器 (TIM_EGR)

偏移地址: 0x0014

表 203 TIM 事件生成寄存器

位	访问	描述
31:7	Res	Reserved 复位值: 0x0
6	r/w	TG: 生成触发信号 此位由软件置 1 以生成事件, 并由硬件自动清零。 0: 不执行任何操作 1: TIM_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。 复位值: 0x0
5	Res	Reserved 复位值: 0x0
4	r/w	CC3G: 捕获/比较 3 生成 参照 CC0G 描述 复位值: 0x0
3	r/w	CC2G: 捕获/比较 2 生成 参照 CC0G 描述 复位值: 0x0
2	r/w	CC1G: 捕获/比较 1 生成 参照 CC0G 描述 复位值: 0x0
1	r/w	CC0G: 捕获/比较 0 生成 此位由软件置 1 以生成事件, 并由硬件自动清零。 0: 不执行任何操作 1: 通道 1 上生成捕获/比较事件: <ul style="list-style-type: none"> 通道 CC0 配置为输出: 使能时, CC0IF 标志置 1 并发送相应的中断或 DMA 请求。 通道 CC0 配置为输入: TIM_CCR0 寄存器中将捕获到计数器当前值。使能时, CC0IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC0IF 标志已为高电平, CC0OF 标志将置 1。 复位值: 0x0

位	访问	描述
0	r/w	<p>UG: 更新生成</p> <p>该位可通过软件置 1，并由硬件自动清零。</p> <p>0: 不执行任何操作</p> <p>1: 重新初始化计数器并生成一个寄存器更新事件。</p> <p>注: 预分频器计数器也将清零 (但预分频比不受影响)，而且计数器将清零</p> <p>复位值: 0x0</p>

14.5.7 TIM 捕获/比较模式寄存器 0 (TIM_CCMR0)

偏移地址: 0x0018

表 204 TIM 捕获/比较模式寄存器 0 (输出比较模式)

位	访问	描述
31:25	Res	<p>Reserved</p> <p>复位值: 0x0</p>
24	r/w	<p>OC1CE: 输出比较 1 清零使能</p> <p>参照 OC1CE 描述</p> <p>复位值: 0x0</p>
23:20	r/w	<p>OC1M: 输出比较 1 模式</p> <p>参照 OC0M 描述</p> <p>复位值: 0x0</p>
19	r/w	<p>OC1PE: 输出比较 1 预装载使能</p> <p>参照 OC0PE 相关描述</p> <p>复位值: 0x0</p>
18	r/w	<p>OC1FE: 输出比较 1 快速使能</p> <p>参照 OC0FE 描述</p> <p>复位值: 0x0</p>
17:16	r/w	<p>CC1S: 捕获/比较 1 通道选择</p> <p>00: CC1 通道配置为输出</p> <p>01: CC1 通道配置为输入, IC1 映射到 TI1 上</p> <p>10: CC1 通道配置为输入, IC1 映射到 TI0 上</p> <p>11: CC1 通道配置为输入, IC1 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效</p> <p>注:</p> <p>仅当通道关闭时 (TIM_CCER 中的 CC1E=0), 才可向 CC1S 位写入数据。</p> <p>复位值: 0x0</p>
15:9	Res	<p>Reserved</p> <p>复位值: 0x0</p>
8	r/w	<p>OC0CE: 输出比较 0 清零使能</p> <p>0: OC0Ref 不受 ETRF 输入影响</p> <p>1: ETRF 输入上检测到高电平时, OC0REF 立即清零</p> <p>复位值: 0x0</p>

位	访问	描述
7:4	r/w	<p>OC0M: 输出比较 0 模式</p> <p>这些位定义提供 OC0 和 OC0N 的输出参考信号 OC0REF 的行为。OC0REF 为高电平有效，而 OC0 和 OC0N 的有效电平则取决于 CC0P 位和 CC0NP 位。</p> <p>0000: 冻结。输出比较寄存器 TIM_CCR0 与计数器 TIM_CNT 进行比较不会对输出造成任何影响。</p> <p>0001: 将通道 0 设置为匹配时输出有效电平。当计数器 TIM_CNT 与捕获/比较寄存器 0 (TIM_CCR0) 匹配时，OC0REF 信号强制变为高电平。</p> <p>0010: 将通道 0 设置为匹配时输出无效电平。当计数器 TIM_CNT 与捕获/比较寄存器 0 (TIM_CCR0)匹配时，OC0REF 信号强制变为低电平。</p> <p>0011: 翻转。TIM_CNT=TIM_CCR0 时，OC0REF 发生翻转。</p> <p>0100: 强制变为无效电平。OC0REF 强制变为低电平。</p> <p>0101: 强制变为有效电平。OC0REF 强制变为高电平。</p> <p>0110: PWM 模式 0--在递增计数模式下，只要 TIM_CNT<TIM_CCR0，通道 0 便为有效状态，否则为无效状态。在递减计数模式下，只要 TIM_CNT>TIM_CCR0，通道 0 便为无效状态 (OC0REF="0")，否则为有效状态 (OC0REF="1")。</p> <p>0111: PWM 模式 1--在递增计数模式下，只要 TIM_CNT<TIM_CCR0，通道 0 便为无效状态，否则为有效状态。在递减计数模式下，只要 TIM_CNT>TIM_CCR0，通道 0 便为有效状态，否则为无效状态。</p> <p>其他：保留</p> <p>注：</p> <p>只要 CC1S="00" (通道配置为输出)，这些位即无法修改。在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。</p> <p>复位值：0x0</p>
3	r/w	<p>OC0PE: 输出比较 0 预装载使能</p> <p>0:禁止与 TIM_CCR0 相关的预装载寄存器。可随时向 TIM_CCR0 写入数据，写入后将立即使用新值。</p> <p>1: 使能与 TIM_CCR0 相关的预装载寄存器。可读/写访问预装载寄存器。TIM_CCR0 预装载值在每次生成更新事件时都会装载到活动寄存器中。</p> <p>注：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式 (TIM_CR1 寄存器中的 OPM 位置 1)。其它情况下则无法保证该行。</p> <p>复位值：0x0</p>
2	r/w	<p>OC0FE: 输出比较 0 快速使能</p> <p>此位用于加快触发输入事件对 CC0 输出的影响。</p> <p>0: 即使触发开启，CC0 也将根据计数器和 CCR0 值正常工作。触发输入出现边沿时，激活 CC0 输出的最短延迟时间为 5 个时钟周期。</p> <p>1: 触发输入上出现有效边沿相当于 CC0 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC0 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM0 或 PWM1 模式时，OC0FE 才会起作用。</p> <p>复位值：0x0</p>

位	访问	描述
1:0	r/w	<p>CC0S: 捕获/比较 0 通道选择</p> <p>00: CC0 通道配置为输出</p> <p>01: CC0 通道配置为输入, IC0 映射到 TI0 上</p> <p>10: CC0 通道配置为输入, IC0 映射到 TI1 上</p> <p>11: CC0 通道配置为输入, IC0 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_MCR 寄存器) 选择内部触发输入时有效</p> <p>注: 仅当通道关闭时 (TIM_CCER 中的 CC0E=0), 才可向 CC0S 位写入数据。</p> <p>复位值: 0x0</p>

表 205 TIM 捕获/比较模式寄存器 0 (输入捕获模式)

位	访问	描述
31:24	Res	<p>Reserved</p> <p>复位值: 0x0</p>
23:20	r/w	<p>IC1F: 输入捕获 1 滤波器</p> <p>参照 IC0F 相关描述</p> <p>复位值: 0x0</p>
19:18	r/w	<p>IC1PSC: 输入捕获 1 预分频器</p> <p>参照 IC0PSC 相关描述</p> <p>复位值: 0x0</p>
17:16	r/w	<p>CC1S: 捕获/比较 2 通道选择</p> <p>00: CC1 通道配置为输出</p> <p>01: CC1 通道配置为输入, IC1 映射到 TI1 上</p> <p>10: CC1 通道配置为输入, IC1 映射到 TI0 上</p> <p>11: CC1 通道配置为输入, IC1 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效</p> <p>注意: 仅当通道关闭时 (TIM_CCER 中的 CC1E=0), 才可向 CC1S 位写入数据。</p> <p>复位值: 0x0</p>
15:8	Res	<p>Reserved</p> <p>复位值: 0x0</p>

位	访问	描述
7:4	r/w	<p>IC0F: 输入捕获 0 滤波器 此位域可定义 TIO 输入的采样频率和适用于 TIO 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿： 0000: 无滤波器，按 f_{DTS} 频率进行采样 0001: $f_{SAMPLING}=f_{CK_INT}$, $N=2$ 0010: $f_{SAMPLING}=f_{CK_INT}$, $N=4$ 0011: $f_{SAMPLING}=f_{CK_INT}$, $N=8$ 0100: $f_{SAMPLING}=f_{DTS}/2$, $N=6$ 0101: $f_{SAMPLING}=f_{DTS}/2$, $N=8$ 0110: $f_{SAMPLING}=f_{DTS}/4$, $N=6$ 0111: $f_{SAMPLING}=f_{DTS}/4$, $N=8$ 1000: $f_{SAMPLING}=f_{DTS}/8$, $N=6$ 1001: $f_{SAMPLING}=f_{DTS}/8$, $N=8$ 1010: $f_{SAMPLING}=f_{DTS}/16$, $N=5$ 1011: $f_{SAMPLING}=f_{DTS}/16$, $N=6$ 1100: $f_{SAMPLING}=f_{DTS}/16$, $N=8$ 1101: $f_{SAMPLING}=f_{DTS}/32$, $N=5$ 1110: $f_{SAMPLING}=f_{DTS}/32$, $N=6$ 1111: $f_{SAMPLING}=f_{DTS}/32$, $N=8$ 复位值: 0x0</p>
3:2	r/w	<p>IC0PSC: 输入捕获 0 预分频器 此位域定义 CC1 输入 (IC0) 的预分频比。只要 CC0E=0 (TIM_CCER 寄存器)，预分频器便立即复位。 00: 无预分频器，捕获输入上每检测到一个边沿便执行捕获 01: 每发生 2 个事件便执行一次捕获 10: 每发生 4 个事件便执行一次捕获 11: 每发生 8 个事件便执行一次捕获 复位值: 0x0</p>
1:0	r/w	<p>CC0S: 捕获/比较 0 通道选择 00: CC0 通道配置为输出 01: CC0 通道配置为输入，IC0 映射到 TIO 上 10: CC0 通道配置为输入，IC0 映射到 TI1 上 11: CC0 通道配置为输入，IC0 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效 注意: 仅当通道关闭时 (TIM_CCER 中的 CC0E=0)，才可向 CC0S 位写入数据。 复位值: 0x0</p>

14.5.8 TIM 捕获/比较模式寄存器 1 (TIM_CCMR1)

偏移地址: 0x001C

表 206 TIM 捕获/比较模式寄存器 1 (输出比较模式)

位	访问	描述
31:25	Res	Reserved 复位值: 0x0
24	r/w	OC3CE: 输出比较 3 清零使能 参照 OC2CE 描述 复位值: 0x0
23:20	r/w	OC3M: 输出比较 3 模式 参照 OC2M 描述 复位值: 0x0
19	r/w	OC3PE: 输出比较 3 预装载使能 参照 OC2PE 相关描述 复位值: 0x0
18	r/w	OC3FE: 输出比较 3 快速使能 参照 OC2FE 描述 复位值: 0x0
17:16	r/w	CC3S: 捕获/比较 3 通道选择 00: CC3 通道配置为输出 01: CC3 通道配置为输入, IC3 映射到 TI3 上 10: CC3 通道配置为输入, IC3 映射到 TI2 上 11: CC3 通道配置为输入, IC3 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效 注: 仅当通道关闭时 (TIM_CCER 中的 CC3E=0), 才可向 CC3S 位写入数据。 复位值: 0x0
15:9	Res	Reserved 复位值: 0x0
8	r/w	OC2CE: 输出比较 2 清零使能 0: OC2REF 不受 ETRF 输入影响 1: ETRF 输入上检测到高电平时, OC2REF 立即清零 复位值: 0x0
7:4	r/w	OC2M: 输出比较 2 模式 这些位定义提供 OC2 和 OC2N 的输出参考信号 OC2REF 的行为。OC2REF 为高电平有效, 而 OC2 和 OC2N 的有效电平则取决于 CC2P 位和 CC2NP 位。 0000: 冻结。输出比较寄存器 TIM_CCR2 与计数器 TIM_CNT 进行比较不会对输出造成任何影响。 0001: 将通道 2 设置为匹配时输出有效电平。当计数器 TIM_CNT 与捕获/比较寄存器 1 (TIM_CCR2) 匹配时, OC2REF 信号强制变为高电平。 0010: 将通道 2 设置为匹配时输出无效电平。当计数器 TIM_CNT 与捕获/比较寄存器 2 (TIM_CCR2) 匹配时, OC2REF 信号强制变为低电平。

位	访问	描述
		<p>0011: 翻转。TIM_CNT=TIM_CCR2 时, OC2REF 发生翻转。</p> <p>0100: 强制变为无效电平。OC2REF 强制变为低电平。</p> <p>0101: 强制变为有效电平。OC2REF 强制变为高电平。</p> <p>0110: 对称 PWM 模式 0--在递增计数模式下, 只要 TIM_CNT<TIM_CCR2, 通道 2 便为有效状态, 否则为无效状态。在递减计数模式下, 只要 TIM_CNT>TIM_CCR1, 通道 2 便为无效状态 (OC2REF="0"), 否则为有效状态 (OC2REF="1")。</p> <p>111: PWM 模式 1--在递增计数模式下, 只要 TIM_CNT<TIM_CCR2, 通道 2 便为无效状态, 否则为有效状态。在递减计数模式下, 只要 TIM_CNT>TIM_CCR1, 通道 2 便为有效状态, 否则为无效状态。</p> <p>其他: 保留</p> <p>注: 只要编程了 CC1S="00" (通道配置为输出), 这些位即无法修改。在 PWM 模式下, 仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时, OCREF 电平才会发生更改。</p> <p>复位值: 0x0</p>
3	r/w	<p>OC2PE: 输出比较 2 预装载使能</p> <p>0: 禁止与 TIM_CCR0 相关的预装载寄存器。可随时向 TIM_CCR2 写入数据, 写入后将立即使用新值。</p> <p>1: 使能与 TIM_CCR2 相关的预装载寄存器。可读/写访问预装载寄存器。TIM_CCR2 预装载值在每次生成更新事件时都会装载到活动寄存器中。</p> <p>注: 只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式 (TIM_CR0 寄存器中的 OPM 位置 1)。其它情况下则无法保证该行。</p> <p>复位值: 0x0</p>
2	r/w	<p>OC2FE: 输出比较 2 快速使能</p> <p>此位用于加快触发输入事件对 CC 输出的影响。</p> <p>0: 即使触发开启, CC2 也将根据计数器和 CCR2 值正常工作。触发输入出现边沿时, 激活 CC2 输出的最短延迟时间为 5 个时钟周期。</p> <p>1: 触发输入上出现有效边沿相当于 CC2 输出上的比较匹配。随后, 无论比较结果如何, OC 都设置为比较电平。采样触发输入和激活 CC2 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM0 或 PWM1 模式时, OC2FE 才会起作用。</p> <p>复位值: 0x0</p>
1:0	r/w	<p>CC2S: 捕获/比较 2 通道选择</p> <p>00: CC2 通道配置为输出</p> <p>01: CC2 通道配置为输入, IC2 映射到 TI2 上</p> <p>10: CC2 通道配置为输入, IC2 映射到 TI3 上</p> <p>11: CC3 通道配置为输入, IC2 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效</p> <p>注: 仅当通道关闭时 (TIM_CCER 中的 CC2E=0), 才可向 CC2S 位写入数据。</p> <p>复位值: 0x0</p>

表 207 TIM 捕获/比较模式寄存器 1 (输入捕获模式)

位	访问	描述
31:24	Res	Reserved 复位值: 0x0
23:20	r/w	IC3F: 输入捕获 3 滤波器 参照 IC2F 相关描述 复位值: 0x0
19:18	r/w	IC3PSC: 输入捕获 3 预分频器 参照 IC2PSC 相关描述 复位值: 0x0
17:16	r/w	CC3S: 捕获/比较 3 通道选择 00: CC3 通道配置为输出 01: CC3 通道配置为输入, IC3 映射到 TI3 上 10: CC3 通道配置为输入, IC3 映射到 TI2 上 11: CC3 通道配置为输入, IC3 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效 注意: 仅当通道关闭时 (TIM_CCER 中的 CC3E=0), 才可向 CC3S 位写入数据。 复位值: 0x0
15:8	Res	Reserved 复位值: 0x0
7:4	r/w	IC2F: 输入捕获 2 滤波器 此位域可定义 TI2 输入的采样频率和适用于 TI2 的数字滤波器带宽。数字滤波器由事件计数器组成, 每 N 个事件才视为一个有效边沿: 0000: 无滤波器, 按 f_{DTS} 频率进行采样 0001: $f_{SAMPLING}=f_{CK_INT}$, N=2 0010: $f_{SAMPLING}=f_{CK_INT}$, N=4 0011: $f_{SAMPLING}=f_{CK_INT}$, N=8 0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 1111: $f_{SAMPLING}=f_{DTS}/32$, N=8 复位值: 0x0

位	访问	描述
3:2	r/w	<p>IC2PSC: 输入捕获 2 预分频器</p> <p>此位域定义 CC2 输入 (IC2)的预分频比。只要 CC2E=0 (TIM_CCER 寄存器), 预分频器便立即复位。</p> <p>00: 无预分频器, 捕获输入上每检测到一个边沿便执行捕获</p> <p>01: 每发生 2 个事件便执行一次捕获</p> <p>10: 每发生 4 个事件便执行一次捕获</p> <p>11: 每发生 8 个事件便执行一次捕获</p> <p>复位值: 0x0</p>
1:0	r/w	<p>CC2S: 捕获/比较 3 通道选择</p> <p>00: CC2 通道配置为输出</p> <p>01: CC2 通道配置为输入, IC2 映射到 TI2 上</p> <p>10: CC2 通道配置为输入, IC2 映射到 TI3 上</p> <p>11: CC2 通道配置为输入, IC2 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器)选择内部触发输入时有效</p> <p>注意: 仅当通道关闭时 (TIM_CCER 中的 CC2E=0), 才可向 CC2S 位写入数据。</p> <p>复位值: 0x0</p>

14.5.9 TIM 捕获/比较模式使能寄存器 (TIM_CCER)

偏移地址: 0x0020

表 208 TIM 捕获/比较模式使能寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15	r/w	CC3NP: 捕获/比较 3 互补输出极性 参照 CC0NP 描述, 该位只在输入模式下有意义 复位值: 0x0
14	Res	Reserved 复位值: 0x0
13	r/w	CC3P: 捕获/比较 3 输出极性 参照 CC0P 描述 复位值: 0x0
12	r/w	CC3E: 捕获/比较 3 输出使能 参照 CC0E 描述 复位值: 0x0
11	r/w	CC2NP: 捕获/比较 2 互补输出极性 参照 CC0NP 描述 复位值: 0x0
10	Res	Reserved 复位值: 0x0
9	r/w	CC2P: 捕获/比较 2 输出极性 参照 CC1P 描述 复位值: 0x0
8	r/w	CC2E: 捕获/比较 2 输出使能 参照 CC0E 描述 复位值: 0x0
7	r/w	CC1NP: 捕获/比较 1 互补输出极性 参照 CC0NP 描述 复位值: 0x0
6	Res	Reserved 复位值: 0x0
5	r/w	CC1P: 捕获/比较 1 输出极性 参照 CC0P 描述 复位值: 0x0
4	r/w	CC1E: 捕获/比较 1 输出使能 参照 CC0E 描述 复位值: 0x0

位	访问	描述
3	r/w	CC0NP: 捕获/比较 0 互补输出极性 CC0 通道配置为输入: 此位与 CC0P 配合使用, 用以定义 TI0FP0 和 TI1FP0 的极性。请参见 CC1P 说明。 复位值: 0x0
2	Res	Reserved 复位值: 0x0
1	r/w	CC0P: 捕获/比较 0 输出极性 CC0 通道配置为输出: 0: OC0 高电平有效 1: OC0 低电平有效 CC0 通道配置为输入: CC0NP/CC0P 位可针对触发或捕获操作选择 TI0FP0 和 TI1FP0 的有效极性。 00: 非反相/上升沿触发电路对 TlxFP0 上升沿敏感 (在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TlxFP0 未反相 (在门控模式或编码器模式下执行触发操作)。 01: 反相/下降沿触发电路对 TlxFP0 下降沿敏感 (在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), TlxFP0 反相 (在门控模式或编码器模式下执行触发操作)。 10: 保留, 不使用此配置。 11: 未反相/边沿触发。电路对 TlxFP0 上升沿和下降沿都敏感 (在复位模式、外部时钟模式或触发模式下执行捕获 或触发操作), TlxFP0 未反相 (在门控模式下执行触发操作)。 复位值: 0x0
0	r/w	CC0E: 捕获/比较 0 输出使能 CC0 通道配置为输出: 0: 关闭——OC0 未激活。OC0 电平是 MOE、OIS0 和 CC0E 位的函数。 1: 开启——OC0 信号是相应输出引脚上的输出, 具体取决于 MOE、OIS0 和 CC0E 位。 CC0 通道配置为输入: 此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 0 (TIM_CCR0) 中。 0: 禁止捕获。 1: 使能捕获。 复位值: 0x0

14.5.10 TIM 计数器 (TIM_CNT)

偏移地址: 0x0024

表 209 TIM 计数器

位	访问	描述
31:16	r/w	CNT[31:16]: 计数器值 (仅对于 32 位计数器有意义) 复位值: 0x0
15:0	r/w	CNT[15:0]: 计数器值 注: 计数器值支持计数过程中更改修正, 但这是一个异步操作; 计数器亦可在使能信号起来之前进行软件初值加载。 复位值: 0x0

14.5.11 TIM 预分频器 (TIM_PSC)

偏移地址: 0x0028

表 210 TIM 预分频器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	PSC[15:0]: 预分频器值 计数器时钟频率 (CK_CNT) 等于 $f_{IP}/(PSC[15:0]+1)$ 。PSC 包含每次发生更新事件 (包括计数器通过 TIM_EGR 寄存器中的 UG 位清零时, 或在配置为“复位模式”时通过触发控制器清零时) 时要装载到活动预分频器寄存器的值。 复位值: 0x0

14.5.12 TIM 自动重载寄存器 (TIM_ARR)

偏移地址: 0x002C

表 211 TIM 自动重载寄存器

位	访问	描述
31:16	r/w	ARR[31:16]: 自动重载值 注: 仅对于 32 位计数器有意义 复位值: 0x0
15:0	r/w	ARR[15:0]: 自动重载值 ARR 为要装载到实际自动重载寄存器的值。当自动重载值为空时, 计数器不工作。 复位值: 0xFFFF

14.5.13 TIM 捕获/比较寄存器 0 (TIM_CCR0)

偏移地址: 0x0034

表 212 TIM 捕获/比较寄存器 0

位	访问	描述
31:16	r/w	CCR0[31:16]: 捕获/比较 0 值 (仅对于 32 位计数器有意义) 复位值: 0x0
15:0	r/w	CCR0[15:0]: 捕获/比较 0 值 通道 CC0 配置为输出 CCR0 是捕获/比较寄存器 0 的预装载值。如果没有通过 TIM_CCMR 寄存器中的 OC0PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 0)。实际捕获/比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC0 输出上发出信号的值。 通道 CC0 配置为输入 CCR0 为上一个输入捕获 0 事件 (IC0) 发生时的计数器值。此时该寄存器只读不可写。 复位值: 0x0

14.5.14 TIM 捕获/比较寄存器 1 (TIM_CCR1)

偏移地址: 0x0038

表 213 TIM 捕获/比较寄存器 1

位	访问	描述
31:16	r/w	CCR1[31:16]: 捕获/比较 1 值 (仅对于 32 位计数器有意义) 复位值: 0x0
15:0	r/w	CCR1[15:0]: 捕获/比较 1 值 通道 CC1 配置为输出 CCR1 是捕获/比较寄存器 1 的预装载值。如果没有通过 TIM_CCMR 寄存器中的 OC1PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 1)。实际捕获/比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC1 输出上发出信号的值。 通道 CC1 配置为输入 CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。此时该寄存器只读不可写。 复位值: 0x0

14.5.15 TIM 捕获/比较寄存器 2 (TIM_CCR2)

偏移地址: 0x003C

表 214 TIM 捕获/比较寄存器 2

位	访问	描述
31:16	r/w	CCR2[31:16]: 捕获/比较 2 值 (仅对于 32 位计数器有意义) 复位值: 0x0
15:0	r/w	CCR2[15:0]: 捕获/比较 2 值 通道 CC2 配置为输出 CCR2 是捕获/比较寄存器 2 的预装载值。如果没有通过 TIM_CCMR 寄存器中的 OC2PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 2)。实际捕获/比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC2 输出上发出信号的值。 通道 CC2 配置为输入。CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。此时该寄存器只读不可写。 复位值: 0x0

14.5.16 TIM 捕获/比较寄存器 3 (TIM_CCR3)

偏移地址: 0x0040

表 215 TIM 捕获/比较寄存器 3

位	访问	描述
31:16	r/w	CCR3[31:16]: 捕获/比较 3 值 (仅对于 32 位计数器有意义) 复位值: 0x0
15:0	r/w	CCR3[15:0]: 捕获/比较 3 值 通道 CC3 配置为输出 CCR3 是捕获/比较寄存器 3 的预装载值。如果没有通过 TIM_CCMR 寄存器中的 OC3PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 3)。实际捕获/比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC3 输出上发出信号的值。 通道 CC3 配置为输入 CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。此时该寄存器只读不可写。 复位值: 0x0

14.5.17 TIM 配置密钥寄存器 (TIM_SET_KR)

偏移地址: 0x0064

表 216 TIM 配置安全锁寄存器 (写)

位	访问	描述
31:0	wo	<p>TIM_SET_UNLOCK_KEY: TIM 配置寄存器解锁密钥</p> <p>注: 在默认情况下, TIM 配置寄存器 (TIM_CNT) 是不能直接进行写操作, 在写之前需要输入正确密钥 0xD79B3EFD 以解锁写权限, 即 TIM_SET_UNLOCK=1, 在更新完系统配置寄存器之后, 写入其他值便可重新锁定写权限, 即 TIM_SET_UNLOCK=0。</p> <p>复位值: 0x0</p>

表 217 TIM 配置安全锁寄存器 (读)

位	访问	描述
31:1	Res	<p>Reserved</p> <p>复位值: 0x0</p>
0	ro	<p>TIM_SET_UNLOCK, TIM 配置寄存器解锁与否</p> <p>0: 锁定</p> <p>1: 解锁</p> <p>复位值: 0x0</p>

15 高级定时器 (TIMA)

15.1 概述

该高级定时器 (TIMA) 是由一个带自动加载功能的 16 位计数器构成, 支持可编程预分频计数。定时器可用于多种用途: 对输入信号脉冲宽度量测、比较输出、PWM 输出或带死区插入的互补 PWM 输出。

15.2 主要功能

- 16 位递增、递减、递增/递减自动重载计数器
- 16 位可编程预分频器, 用于对计数器时钟频率进行分频 (即运行时修改), 分频系数介于 1 到 65536 之间
- 3 个互补通道 (通道 0~2), 1 个单通道 (通道 3)
- 6 个独立通道功能, 可用于
 - 输入捕获 (除了通道 4&5)
 - 输出比较
 - PWM 输出 (边沿和中心对齐模式)
 - 单脉冲模式输出
- 带可编程死区的互补输出
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路
- 重复计数器, 用于仅在给定数目的计数器周期后更新定时器寄存器。
- 支持断路输入将输出信号置于复位状态或预写状态
- 支持以下事件的中断生成:
 - 更新: 计数器溢出, 由软件或内外部事件触发计数器初始化
 - 触发事件 (计数器开始、停止、初始化或内外部触发导致的计数动作)
 - 输入捕获
 - 输出比较
 - 断路输入
- 支持定位用增量 (正交) 编码器和霍尔传感器电路
- 外部时钟触发输入或逐周期电流管理

15.3 模块框图

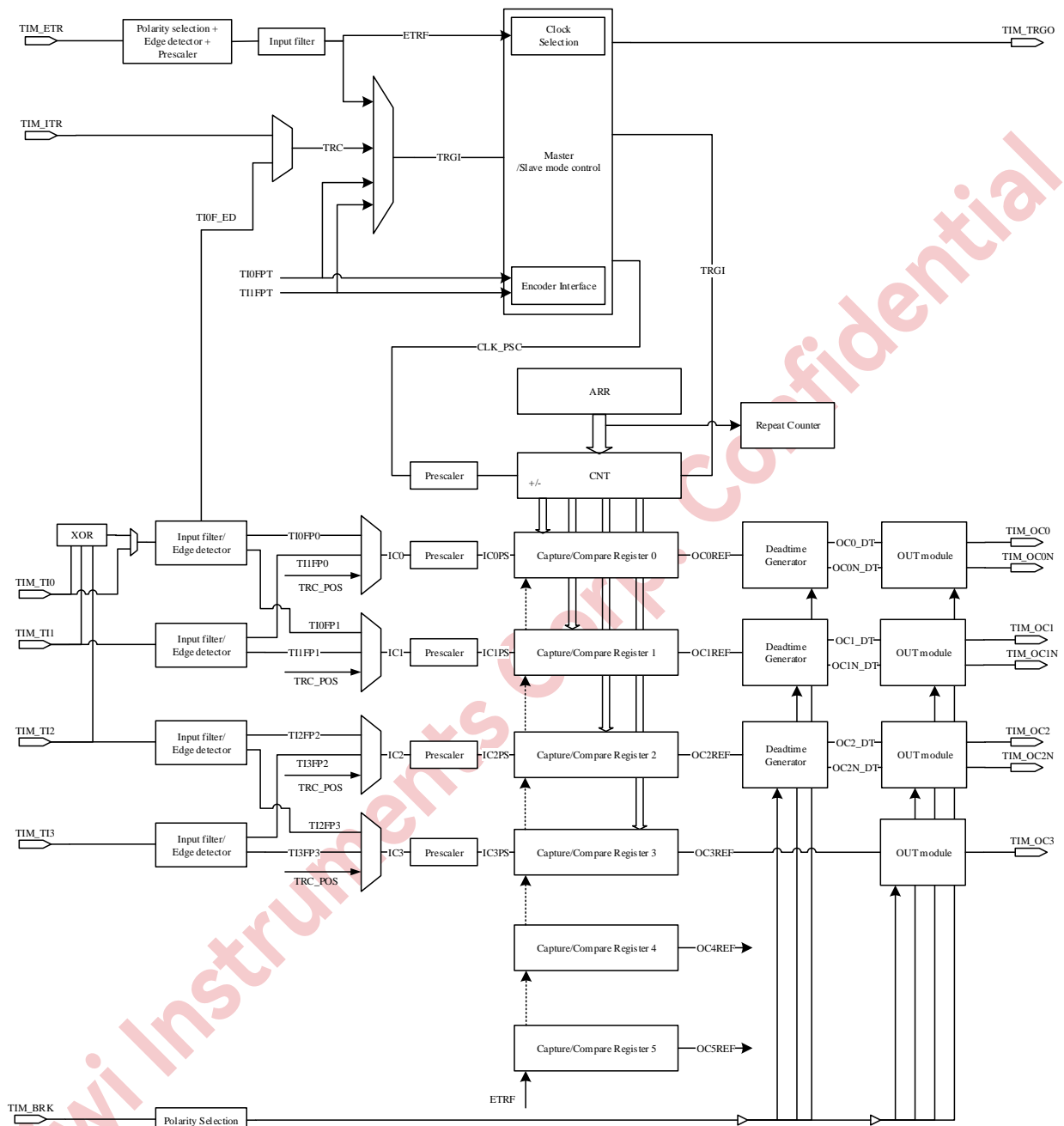


图 113 高级定时器 TIM 总体模块框图

15.4 功能描述

15.4.1 计数器介绍

可编程定时器的主要模块由一个 16 位计数器及其相关的自动重载寄存器组成。此计数器采用递增、递减或中心对齐方式计数。计数器的时钟可通过预分频器进行分频。计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。时基单元包括计数器寄存器 (TIM_CNT)、预分频寄存器 (TIM_PSC)、自动重载寄存器 (TIM_ARR) 和重复计数器寄存器 (TIM_RCR)。

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIM_CR0 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIM_CR0 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CLK_CNT 提供时钟，仅当 TIM_CR0 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。计数器是在 CEN 置 1 的一个时钟周期后方才真正开始计数。

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 TIM_PSC 寄存器中的 16 位寄存器所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此可对预分频器进行实时更改。而新的预分频比将在下一更新事件发生时被采用。

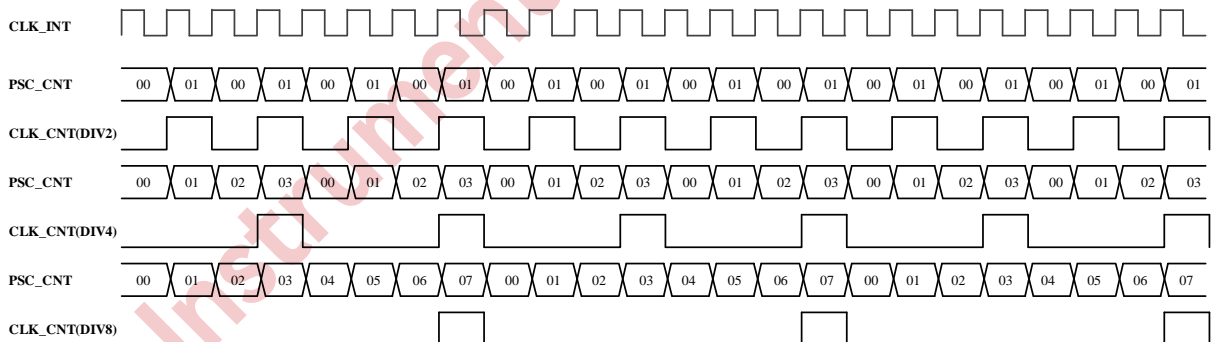


图 114 TIM 预分频器 N 分频时序图

15.4.2 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值 (TIM_ARR 寄存器的内容)，然后重新从 0 开始计数并生成计数器上溢事件。

如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中编程的次数加一次 (TIM_RCR+1) 后，将生成更新事件 (UEV)。否则，将在每次计数器上溢时产生更新事件。

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

通过软件将 TIM_CR0 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数 (而预分频比保持不变)。此外，如果 TIM_CR0 寄存器中的 URS 位 (更新请求选择) 已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1 (因此，不会发送任何中断或 DMA 请求)。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

以下各图以一些示例说明当 TIM_ARR=0x36 时不同时钟频率下计数器的行为。

- 重复计数器中将重新装载 TIM_RCR 寄存器的内容
- 自动重载影子寄存器将以预装载值 (TIM_ARR) 进行更新
- 预分频器的缓冲区中将重新装载预装载值 (TIM_PSC 寄存器的内容)

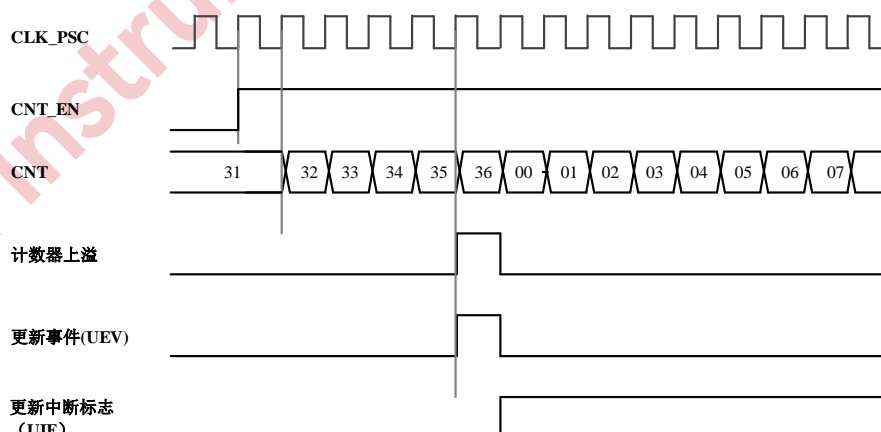


图 115 计数器时序图，1 分频内部时钟

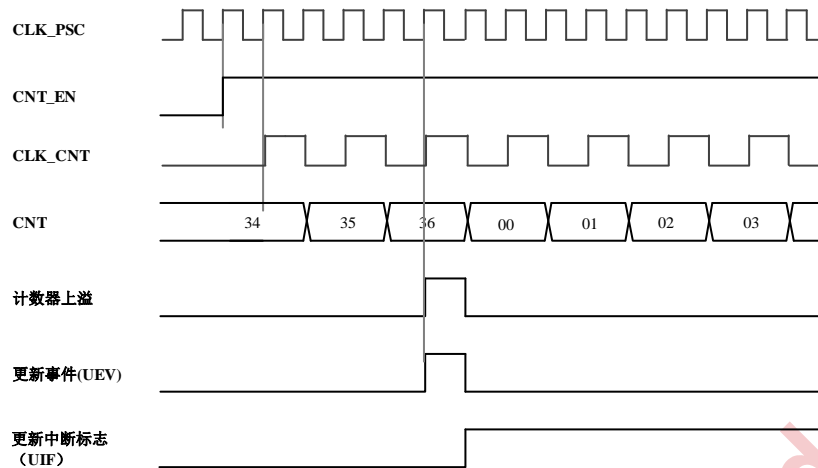


图 116 计数器时序图，2 分频内部时钟

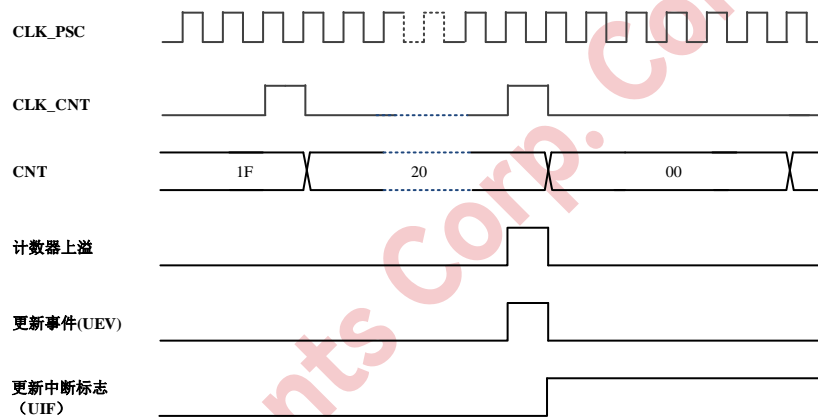


图 117 计数器时序图，N 分频内部时钟

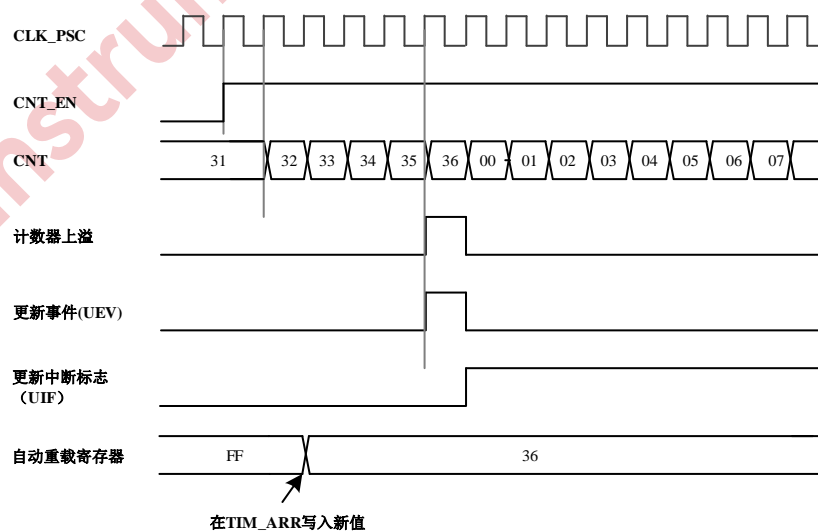


图 118 计数器时序图，ARPE=0 时更新事件 (TIM_ARR 未预装载)

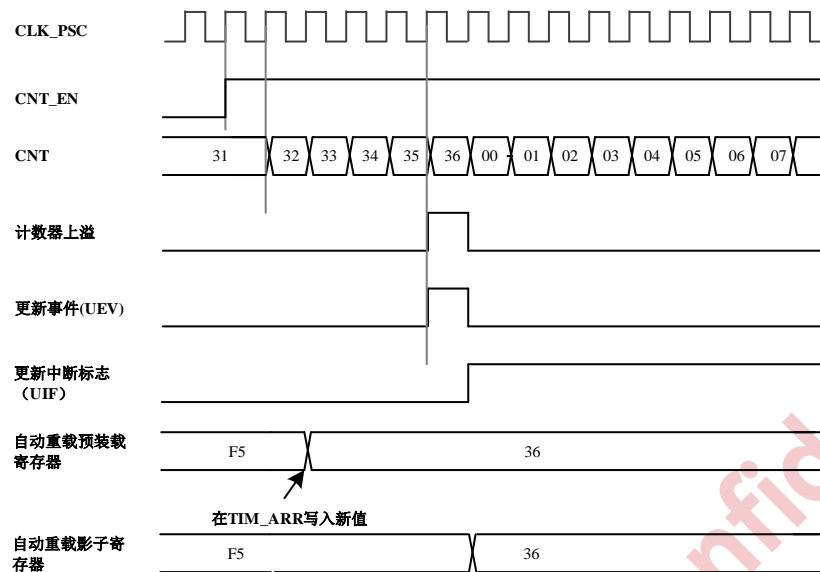


图 119 计数器时序图，ARPE=1 时更新事件 (TIM_ARR 预装载)

递减计数模式

在递减计数模式下，计数器从自动重载值 (TIM_ARR 寄存器的内容) 开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

如果使用重复计数器，则当递减计数的重复次数达到重复计数器寄存器中编程的次数加一次 (TIM_RCR+1) 后，将生成更新事件 (UEV)。否则，将在每次计数器下溢时产生更新事件。

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

通过软件将 TIM_CR0 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数 (但预分频比保持不变)。

此外，如果 TIM_CR0 寄存器中的 URS 位 (更新请求选择) 已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1 (因此，不会发送任何中断或 DMA 请求)。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 重复计数器中将重新装载 TIM_RCR 寄存器的内容
- 预分频器的缓冲区中将重新装载预装载值 (TIM_PSC 寄存器的内容)
- 自动重载活动寄存器将以预装载值 (TIM_ARR 寄存器的内容) 进行更新。注意，自动重载寄存器会在计数器重载之前得到更新，因此，下一个计数周期就是我们所希望的新的周期长度。

以下各图以一些示例说明当 TIM_ARR=0x36 时不同时钟频率下计数器的行为。

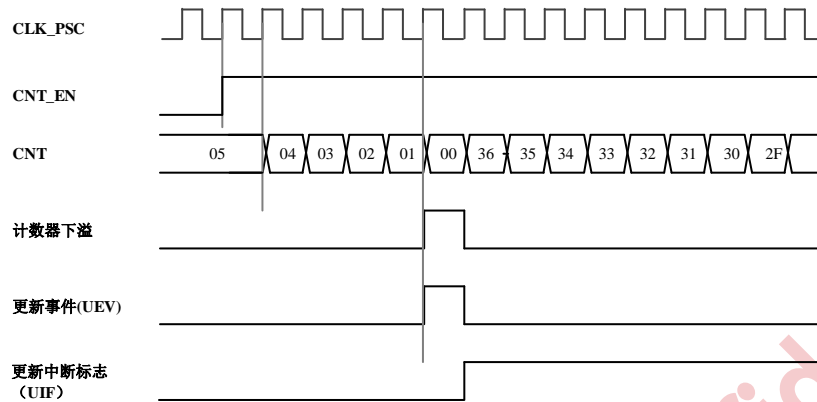


图 120 计数器时序图，1 分频内部时钟

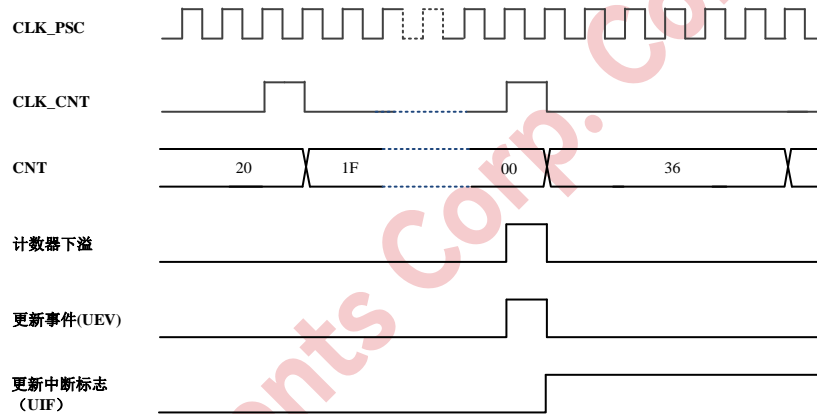


图 121 计数器时序图，N 分频内部时钟

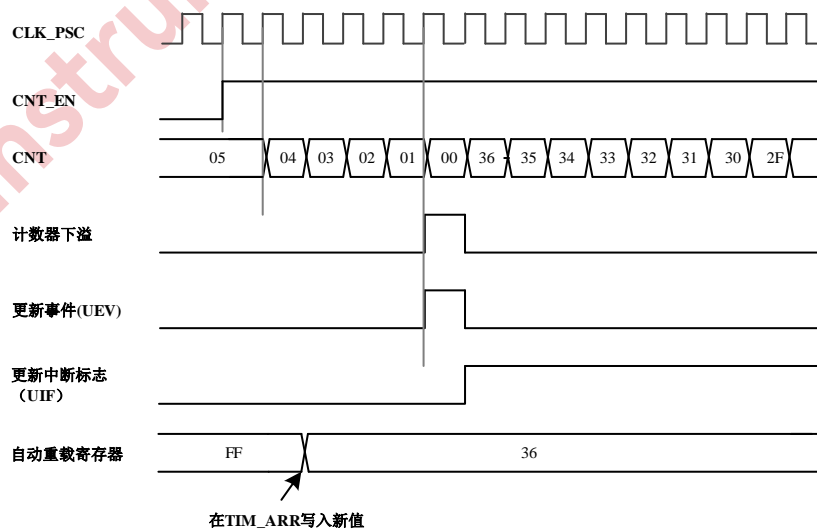


图 122 计数器时序图，未使用重复计数器时更新事件

中心对齐模式 (递增/递减计数)

在中心对齐模式下，计数器从 0 开始计数到自动重载值 (TIM_ARR 寄存器的内容) -1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIM_CR0 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数 (中心对齐模式 1，CMS=“01”)、计数器递增计数 (中心对齐模式 2，CMS=“10”) 以及计数器递增/递减计数 (中心对齐模式 3，CMS=“11”)。

在此模式下，TIM_CR0 寄存器的 DIR 方向位不可写入值，而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TIM_EGR 寄存器中的 UG 位置 1 (通过软件或使用从模式控制器) 也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

通过软件将 TIM_CR0 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

此外，如果 TIM_CR0 寄存器中的 URS 位 (更新请求选择) 已置 1，则将 UG 位置 1 会生成 UEV 更新事件，但不会将 UIF 标志置 1 (因此，不会发送任何中断或 DMA 请求)。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 重复计数器中将重新装载 TIM_RCR 寄存器的内容
- 预分频器的缓冲区中将重新装载预装载值 (TIM_PSC 寄存器的内容)
- 自动重载活动寄存器将以预装载值 (TIM_ARR 寄存器的内容) 进行更新。注意，如果更新操作是由计数器上溢触发的，则自动重载寄存器在重载计数器之前更新，因此，下一个计数周期就是我们所希望的新的周期长度 (计数器被重载新的值)。

以下各图以一些示例说明不同时钟频率下计数器的行为。

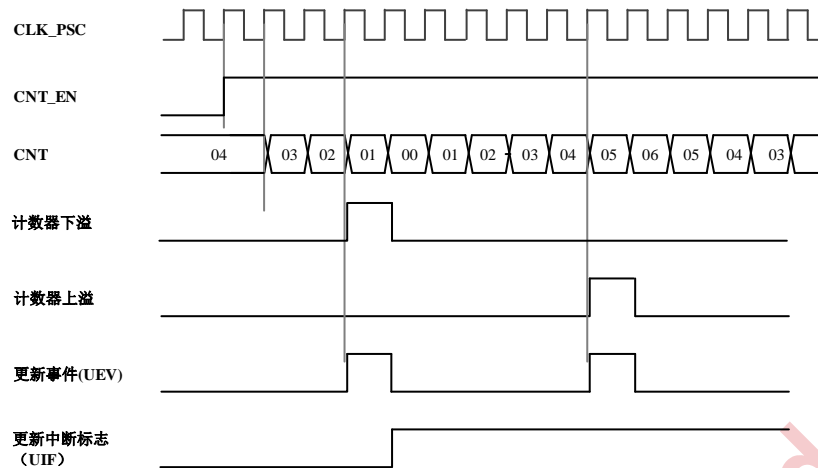


图 123 计数器时序图，1 分频内部时钟，TIM_ARR = 0x6

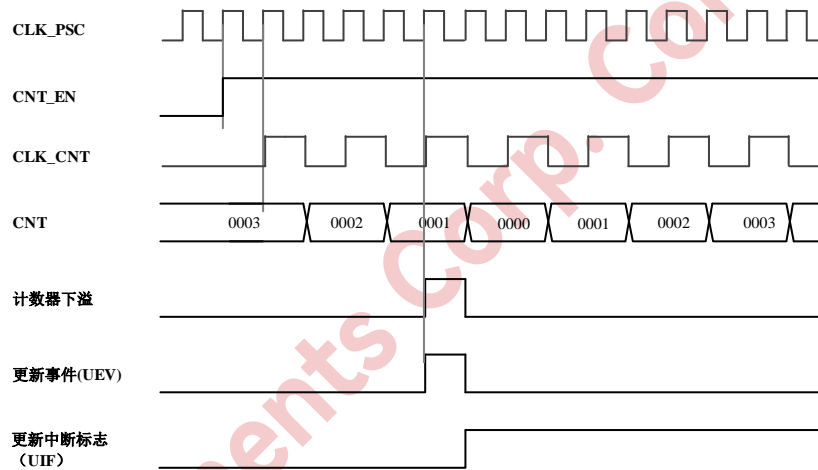


图 124 计数器时序图，2 分频内部时钟

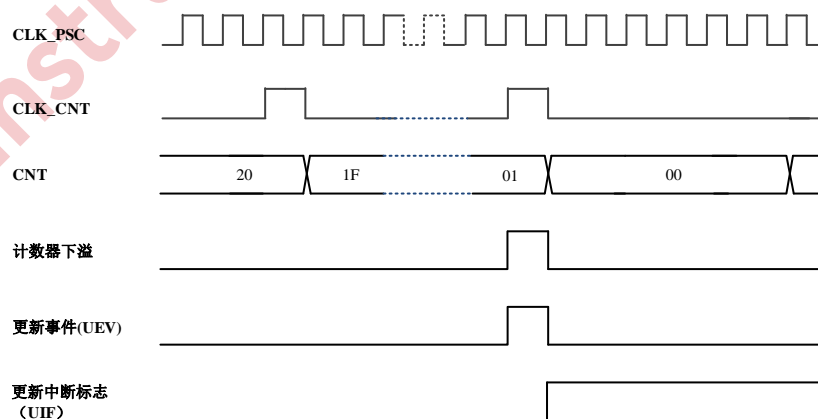


图 125 计数器时序图，N 分频内部时钟

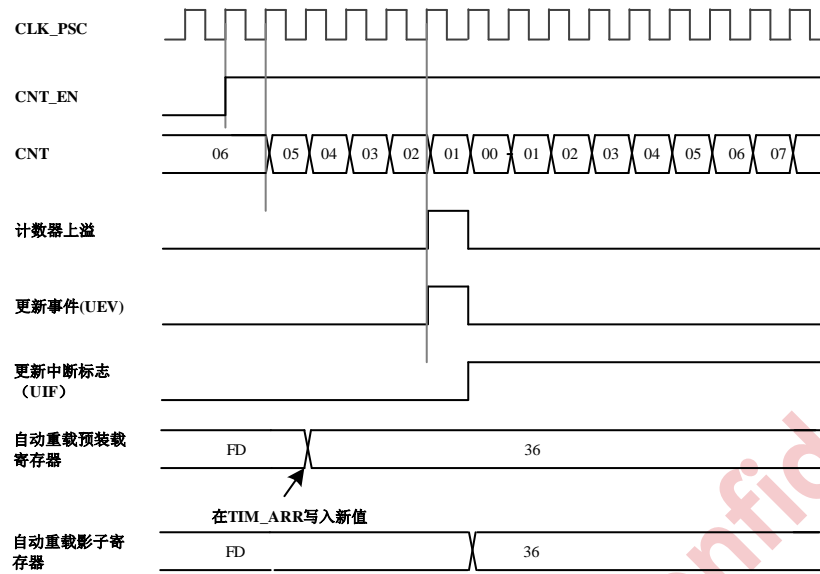


图 126 计数器时序图，ARPE=1 时的更新事件 (计数器下溢)

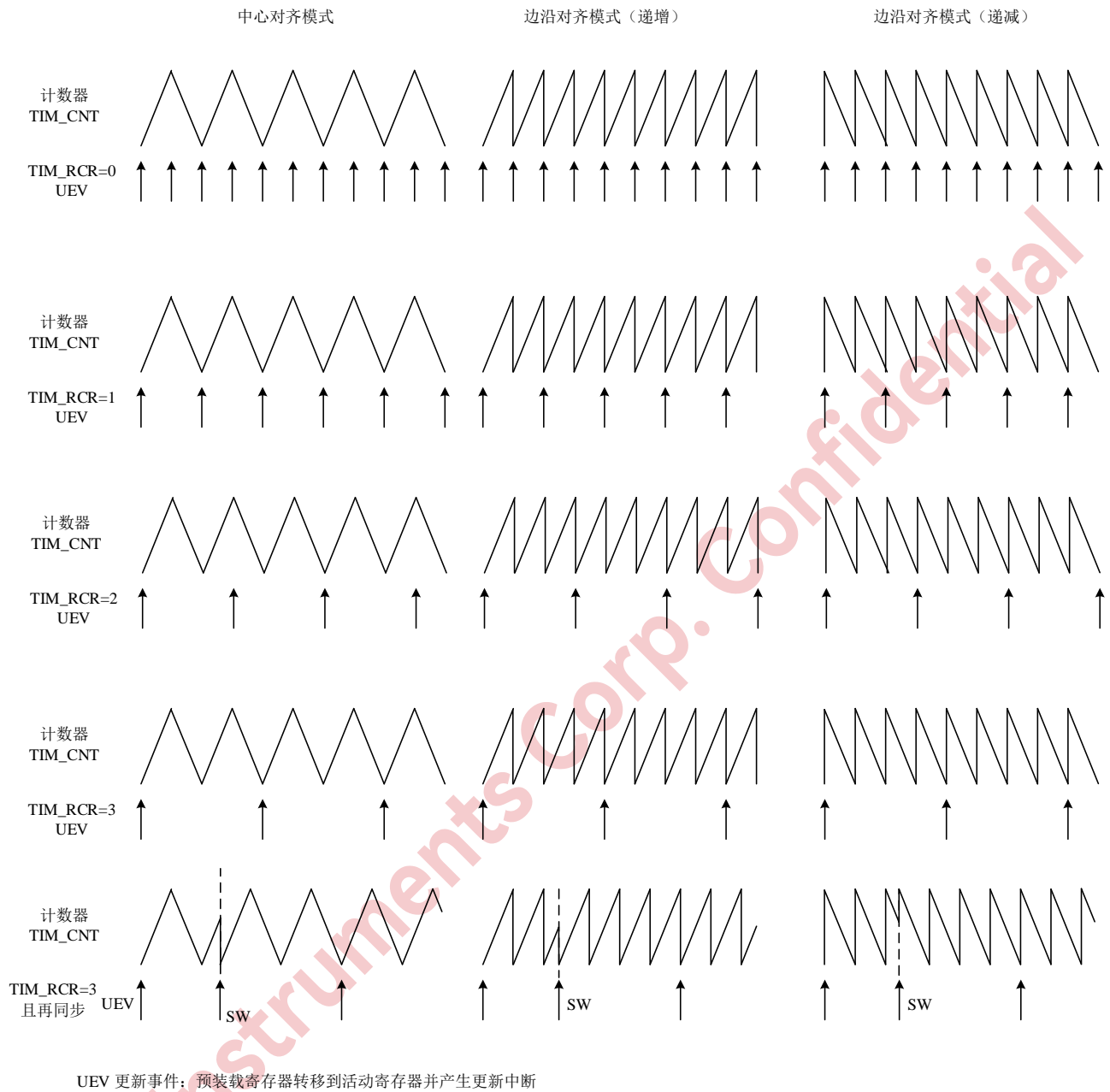


图 127 不同模式和 TIM_RCR 寄存器设置下的更新频率示例

15.4.3 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟 (TIM_CLK_INT)
- 外部时钟模式 0：外部输入引脚 (TIM_TIx)
- 外部时钟模式 1：外部触发输入 ETR
- 内部触发输入 (TIM_ITRx)：使用一个定时器作为另一个定时器的预分频器

15.4.3.1 内部时钟源

如果禁止从模式控制器 (TIM_CR 寄存器中 SMS=000)，则 CEN 位 (TIM_CR) 和 UG 位 (TIM_EGR) 为实际控制位，并且只能通过软件进行更改，但 UG 仍自动清零)。当对 TIM_CEN 位写入 1 时，预分频器的时钟就由内部时钟 TIM_CLK_INT 提供。

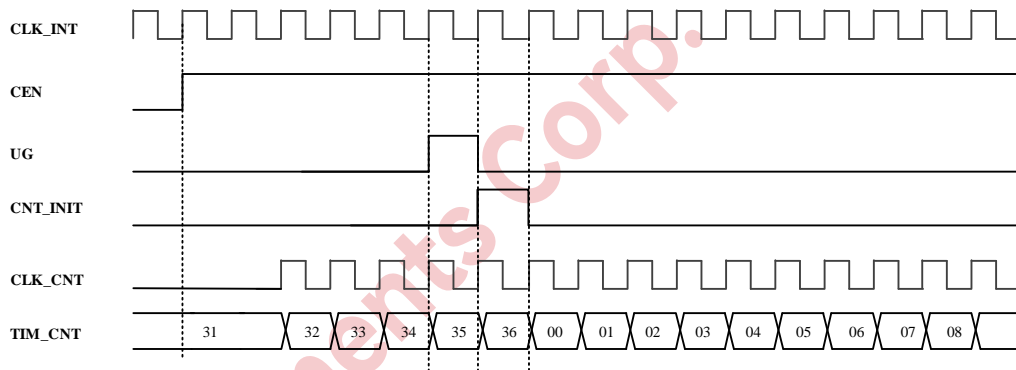


图 128 正常模式下的控制电路 1 分频内部时钟

15.4.3.2 外部时钟源模式 0

当 TIM_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

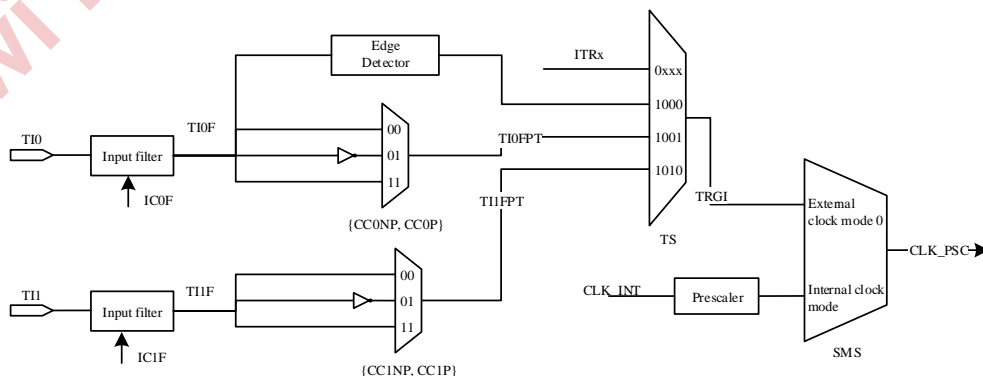


图 129 外部时钟输入连接关系

例如，要使计数器在 TI1 输入出现上升沿时计数，可执行以下步骤：

1. 通过在 TIM_CCMR 寄存器中写入 CC1S=01 来配置通道 1，使其能够检测 TI1 输入的上升沿；
2. 通过在 TIM_CCMR 寄存器中写入 IC1F[3:0] 位来配置输入滤波时间（如果不需要任何滤波，请保持 IC1F=0000）；
3. 通过在 TIM_CCER 寄存器中写入 CC1P=0 和 CC1NP=0 来选择上升沿极性；
4. 通过在 TIM_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式下工作；
5. 通过在 TIM_SMCR 寄存器中写入 TS=110 来选择 TI1 作为输入源；
6. 通过在 TIM_CR0 寄存器中写入 CEN=1 来使能计数器。

当 TI1 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。TI1 的上升沿与实际计数器时钟之间的延迟是由于 TI1 输入的重新同步电路引起的。

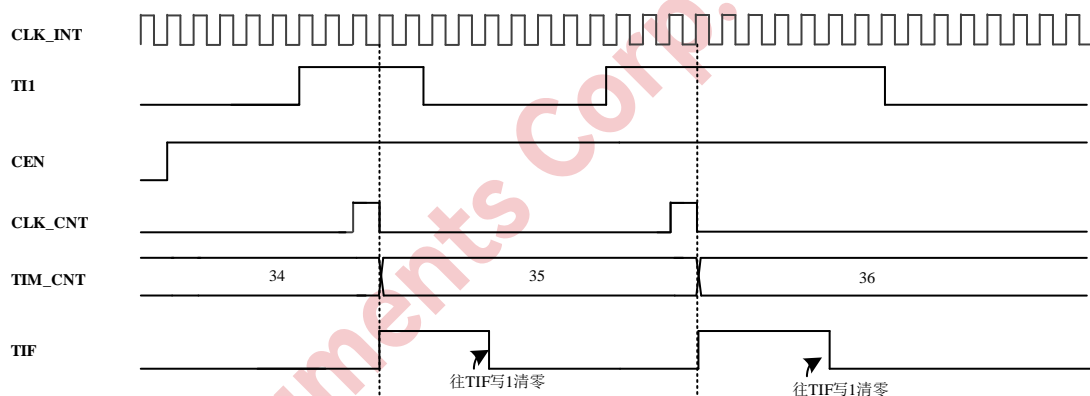


图 130 外部时钟模式下 1 的控制电路

15.4.3.3 外部时钟源模式 1

通过在 TIM_SMCR 寄存器中写入 ECE=1 可选择此模式。

计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

下图简要介绍了外部触发输入模块。

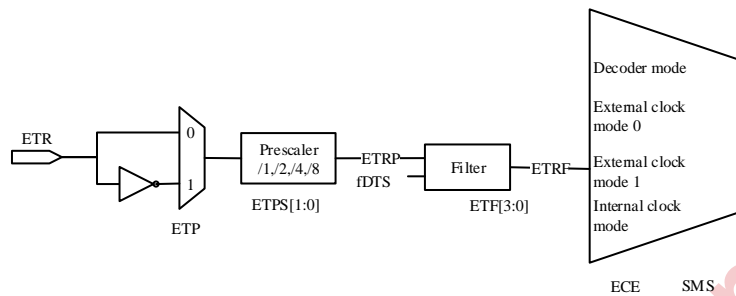


图 131 外部触发输入模块

例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 由于此例中不需滤波器，因此在 TIM_SMCR 寄存器中写入 ETF[3:0]=0000。
2. 通过在 TIM_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
3. 通过在 TIM_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
4. 通过在 TIM_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
5. 通过在 TIM_CR0 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的。

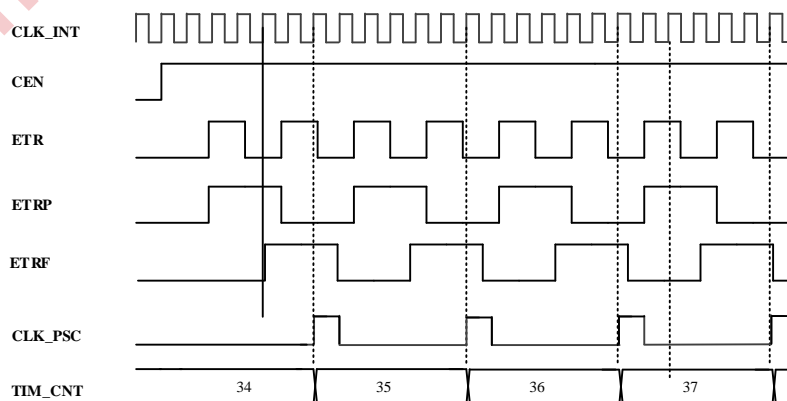


图 132 外部时钟模式 1 下的控制电路

15.4.4 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

下图概括介绍了一个捕获/比较通道。输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 $TIxF$ 。然后，带有极性选择功能的边沿检测器生成一个信号 ($TIxFPx$)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 ($ICxPS$)，而后再进入捕获寄存器。

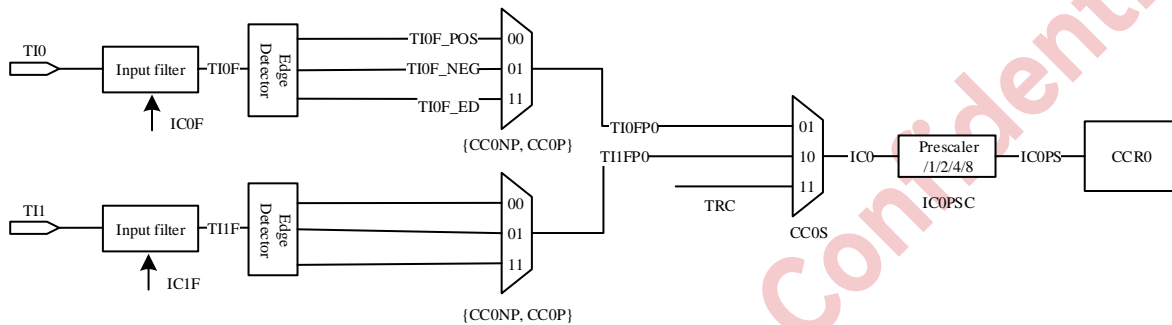


图 133 捕获/比较通道的输入阶段

输出阶段生成一个中间波形作为基准： $OCxREF$ （高电平有效）。链的末端决定最终输出信号的极性。

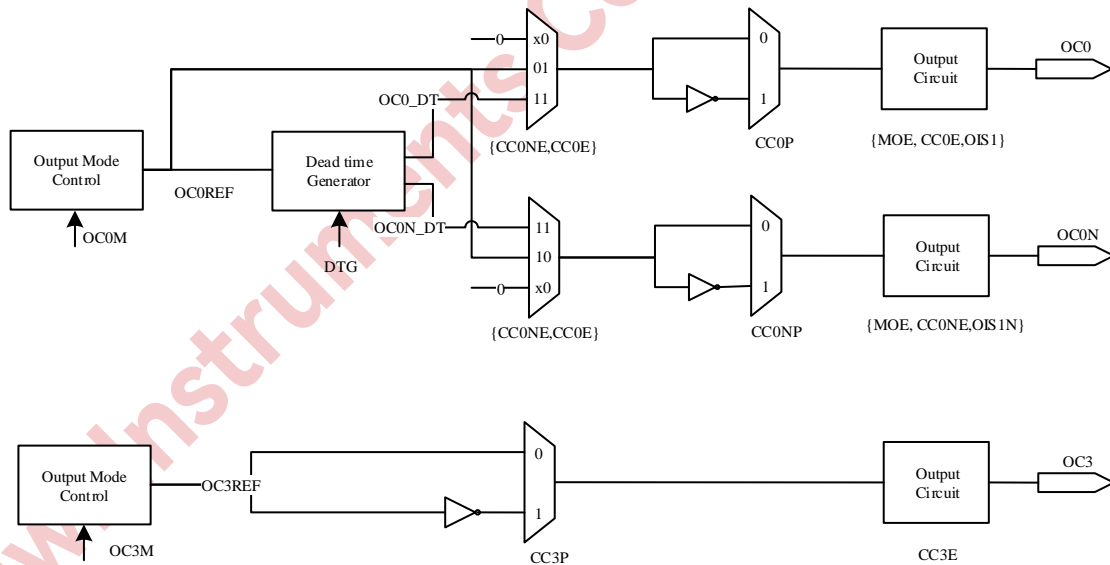


图 134 捕获/比较通道的输出阶段

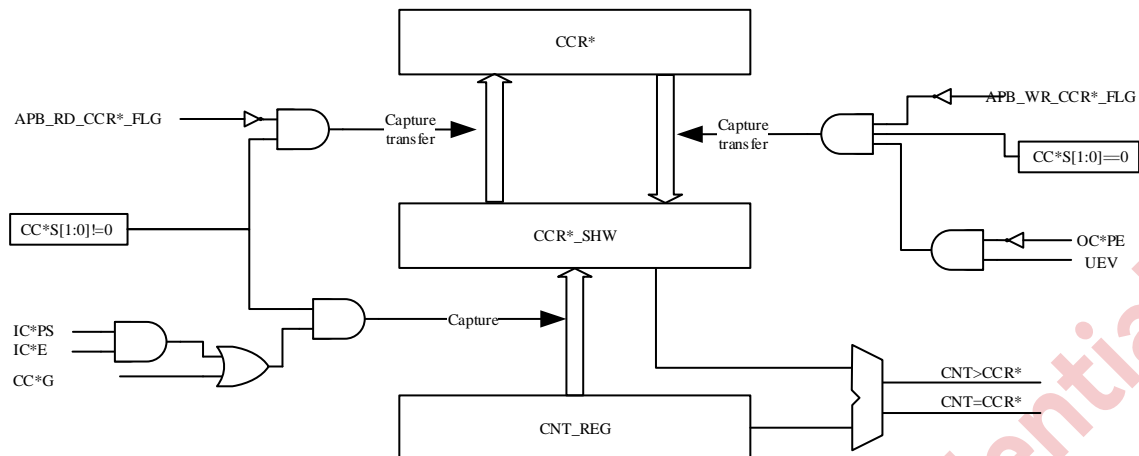


图 135 捕获/比较通道主要电路

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载寄存器。在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。在比较模式下，预装载寄存器的内容将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

15.4.5 输入捕获模式

在输入捕获模式下，当相应的 TIM_TIx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIM_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCxIF 标志 (TIM_SR) 置 1，并可发送中断请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已置位，则会将重复捕获标志 CCxOF (TIM_SR) 置 1。可通过软件向 CCxIF 写入 1 来给 CCxIF 清零，或读取存储在 TIM_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 1 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIM_CCR1 中。具体操作步骤如下：

1. 选择有效输入：TIM_CCR1 必须连接到 TI1 输入，因此向 TIM_CCMR 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIM_CCR1 寄存器将处于只读状态。
2. 根据连接到定时器的信号，对所需的输入滤波时间进行设置（如果输入为 Tlx 输入之一，则对 TIM_CCMR 寄存器中的 ICxF 位进行设置）。输入的信号经过采样时钟进行采样，采样频率可配置（详见相关寄存器描述）。
3. 通过向 TIM_CCER 寄存器中的 CC1P 位和 CC1NP 位写入 0，选择 TI1 通道的有效转换边沿上升沿。
4. 对输入预分频器进行设置。本例中，倘若每次有效转换时都执行捕获操作，那需要禁止预分频器（向 TIM_CCMR 寄存器中的 IC1PS 位写入 00）。
5. 通过将 TIM_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
6. 必要时可通过将 TIM_IER 寄存器中的 CC1IE 位置 1 来使能相关中断请求。

当发生输入捕获时：

- 发生有效跳变沿时，TIM_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1 (中断标志)。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

通过软件将 TIM_EGR 寄存器中的相应 CCxG 位置 1 可生成中断。

15.4.6 PWM 输入模式

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 TIM_TlX 信号被映射至同一个 ICx 输入。
- 这两个 TIM_TlX 信号在边沿处有效，但极性相反。
- 选择两个 TlXFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI0 的 PWM 的周期（位于 TIM_CCR0 寄存器中）和占空比（位于 TIM_CCR1 寄存器中）进行测量（取决于 TIM_CK_INT 频率和预分频器的值）

1. 选择 TIM_CCR0 的有效输入：向 TIM_CCMR0 寄存器中的 CC0S 位写入 01 (选择 TI0)。
2. 选择 TI0FP0 的有效极性（用于 TIM_CCR0 中的捕获和计数器清零）：向 CC0P 位和 CC0NP 位写入 0 (上升沿有效)。
3. 选择 TIM_CCR1 的有效输入：向 TIM_CCMR 寄存器中的 CC1S 写入 10 (选择 TI0)。
4. 选择 TI0FP1 的有效极性（用于 TIM_CCR1 中的捕获）：向 CC1P 位和 CC1NP 位写入 1 (下降沿有效)。
5. 选择有效触发输入：向 TIM_SMCR 寄存器中的 TS 位写入 101 (选择 TI0FPT)。
6. 将从模式控制器配置为复位模式：向 TIM_SMCR 寄存器中的 SMS 位写入 100。
7. 使能计数器开始计数 CEN=1。
8. 使能捕获：向 TIM_CCER 寄存器中的 CC0E 位和 CC1E 位分别写入 1。

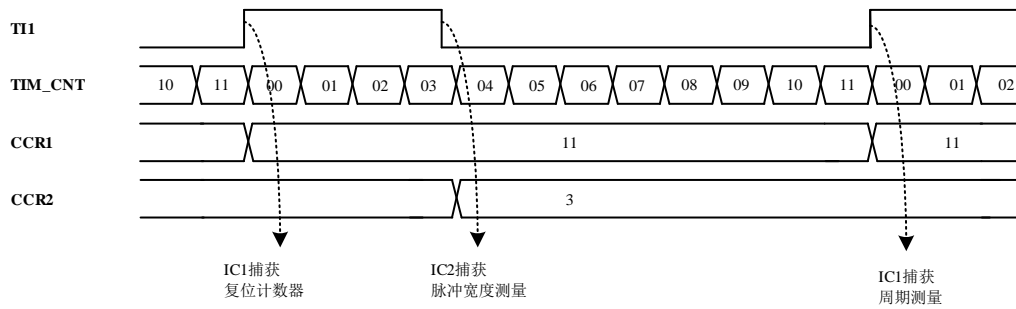


图 136 PWM 输入模式时序

15.4.7 强制输出模式

在输出模式 (TIM_CCMR 中的 CCxS=00) 下, 可直接由软件将每个输出比较信号 (OCxREF 和 OCx) 强制设置为有效电平或无效电平, 而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号 (OCxREF/OCx) 强制设置为有效电平, 只需向相应 TIM_CCMRx 寄存器中的 OCxM 位写入 101。OCxREF 进而强制设置为高电平 (OCxREF 始终为高电平有效), 同时 OCx 获取 CCxP 极性位的相反值。例如: CCxP=0 (OCx 高电平有效) =>OCx 强制设置为高电平。通过向 TIM_CCMRx 寄存器中的 OCxM 位写入 100, 可将 OCxREF 信号强制设置为低电平。

无论如何, TIM_CCRx 影子寄存器与计数器之间的比较仍会执行, 而且允许将标志置 1。因此可发送相应的中断请求。输出比较模式一节对此进行了介绍。

15.4.8 输出比较模式

此功能用于控制输出波形, 或指示已经过去某一时间段计数器状态。

当捕获/比较寄存器与计数器之间相匹配时, 输出比较功能:

- 将为相应的输出引脚分配一个可编程值, 该值由输出比较模式 (TIM_CCMR 寄存器中的 OCxM 位) 和输出极性 (TIM_CCER 寄存器中的 CCxP 位) 定义。匹配时, 输出引脚既可保持其电平 (OCxM=000), 也可设置为有效电平 (OCxM=001)、无效电平 (OCxM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1 (TIM_SR 寄存器中的 CCxIF 位)。
- 如果相应中断使能位 (TIM_IER 寄存器中的 CCxIE 位) 置 1, 将生成中断。

使用 TIM_CCMR 寄存器中的 OCxPE 位, 可将 TIM_CCRx 寄存器配置为带或不带预装载寄存器。

在输出比较模式下, 更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲 (在单脉冲模式下)。

步骤:

1. 选择计数器时钟 (内部、外部、预分频器)。
2. 在 TIM_ARR 和 TIM_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求, 将 CCxIE 位置 1。
4. 选择输出模式。例如, 当 CNT 与 CCRx 匹配、未使用预装载 CCRx 并且 OCx 使能且为高电平有效时, 必须写入 OCxM=011、OCxPE=0、CCxP=0 和 CCxE=1 来翻转 OCx 输出引脚。
5. 通过将 TIM_CR0 中的 TIM_CEN 位置 1 来使能计数器。

可随时通过软件更新 TIM_CCRx 寄存器以控制输出波形, 前提是未使能预装载寄存器 (OCxPE=0, 否则仅当发生下一个更新事件 UEV 时, 才会更新 TIM_CCRx 影子寄存器)。下图列出了相关示例。

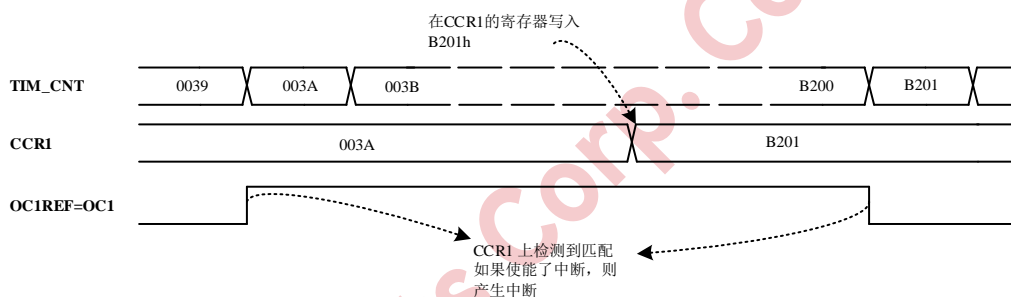


图 137 输出比较模式 (翻转 OC1)

15.4.9 PWM 模式

脉冲宽度调制模式可以生成一个信号, 该信号频率由 TIM_ARR 寄存器值决定, 其占空比则由 TIM_CCRx 寄存器值决定。

通过向 TIM_CCMR 寄存器中的 OCxM 位写入根据应用场景选择这六种 PMW 模式: 0110 (PWM 模式 0)、0111 (PWM 模式 1)、1100 (组合 PWM 模式 0)、1101 (组合 PWM 模式 1)、1110 (非对称 PWM 模式 0) 和 1111 (非对称 PWM 模式 1)。每个通道 (每个 OCx 输出对应一个 PWM) 的 PWM 模式可以独立选择。必须通过将 TIM_CCMR 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器, 最后通过将 TIM_CR 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器。

由于只有在发生更新事件时预装载寄存器才会传送到影子寄存器, 因此启动计数器之前, 必须通过将 TIM_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可使用 TIM_CCER 寄存器的 CCxP 位来编程。既可以设为高电平有效, 也可以设为低电平有效。OCx 输出通过将 TIM_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息, 请参见 TIM_CCER 寄存器说明。

在 PWM 模式下，TIM_CNT 始终与 CCRx 或 ACRx 进行比较，以确定是 $CCR_x < TIM_CNT$ 还是 $TIM_CNT \leq CCR_x$ ，或者 $ACR_x < TIM_CNT$ 还是 $TIM_CNT \leq ACR_x$ (取决于计数器计数方向)。

根据 TIM_CR0 寄存器中 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

15.4.9.1 PWM 边沿对齐模式

递增计数配置

当 TIM_CR0 寄存器中的 DIR 位为低时执行递增计数。请参见递增计数模式一节。适用于此配置的 PWM 模式有：0110 (PWM 模式 0)、0111 (PWM 模式 1)、1100 (组合 PWM 模式 0)、1101 (组合 PWM 模式 1)

对于 PWM 模式 0，只要 $TIM_CNT < CCR_x$ ，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 CCRx 中的比较值大于自动重载值 (TIM_ARR 中)，则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。

下图举例介绍 PWM 模式 0 波形 (TIM_ARR=8)。相对应的，同样的 TIM_CNT 与 TIM_CCRx 判断条件下，PWM 模式 0 的输出 OCxREF 波形与 PWM 模式 1 波形相反。

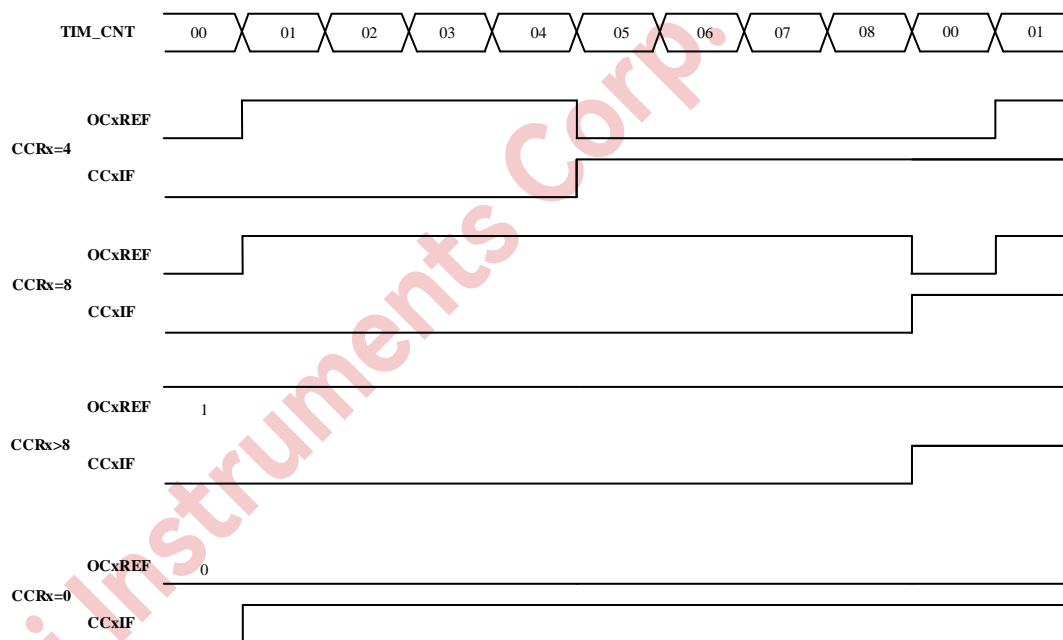


图 138 边沿对齐模式 (递增) 的 PWM0 波形

对于组合 PWM 模式 1，只要 $TIM_CNT < CCR_x$ 或者 $TIM_CNT > ACR_x$ ，PWM 参考信号 OCxREF 便为高电平，否则为低电平，其中要求 $CCR_x < ACR_x$ 。如果 CCRx 中的比较值大于自动重载值 (ARR 中)，则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。下图举例介绍组合 PWM 模式 1 波形。相对应的，同样的 TIM_CNT 与 CCRx 和 ACRx 判断条件下，组合 PWM 模式 0 的输出 OCxREF 波形与组合 PWM 模式 1 波形相反。

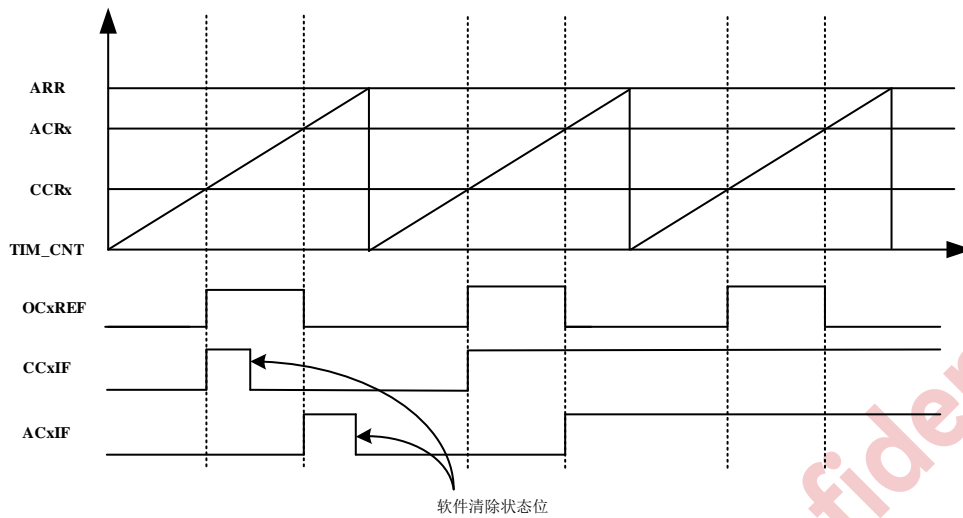


图 139 边沿对齐模式 (递增) 的组合 PWM1 波形

递减计数配置

当 TIM_CR0 寄存器中的 DIR 位为高时执行递减计数。请参见递减计数模式一节。

在 PWM 模式 0 下，只要 $TIM_CNT > CCRx$ ，参考信号 OCxREF 即为低电平，否则其为高电平。如果 CCRx 中的比较值大于 TIM_ARR 中的自动重载值，则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。

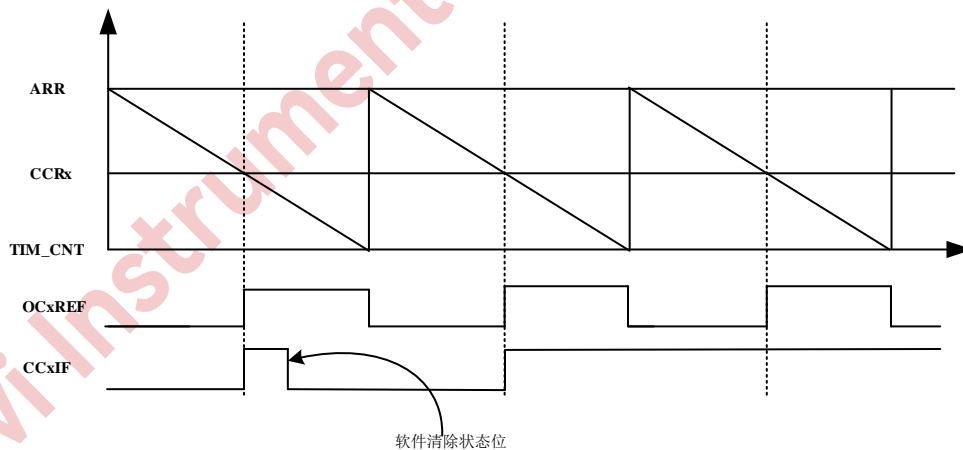


图 140 边沿对齐模式 (递减) 的 PWM0 波形

在组合 PWM 模式 0，只要 $TIM_CNT > CCRx$ 或者 $TIM_CNT \leq ACRx$ ，PWM 参考信号 OCxREF 便为高电平，否则为低电平,其中要求 $CCRx > ACRx$ 。下图展示了组合 PWM 模式 1 的 PWM 波形。相对应的，同样的 TIM_CNT 与 CCRx 和 ACRx 判断条件下，组合 PWM 模式 1 的输出 OCxREF 波形与组合 PWM 模式 0 波形相反。

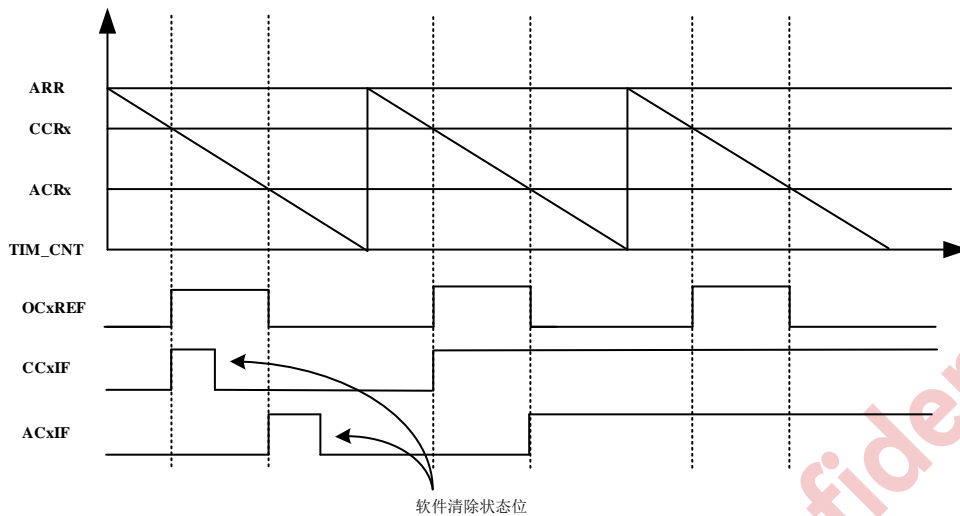


图 141 边沿对齐模式 (递减) 的组合 PWM0 波形

15.4.9.2 PWM 中心对齐模式

当 TIM_CR0 寄存器中的 CMS 位不为“00” (其余所有配置对 OCxREF/OCx 信号具有相同的作用), 中心对齐模式生效。根据 CMS 位的配置, 可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIM_CR0 寄存器中的方向位 (DIR) 由硬件更新, 不得通过软件更改。请参见中心对齐模式 (递增/递减计数)。

适用于此配置的 PWM 模式有: 0110 (PWM 模式 0)、0111 (PWM 模式 1)、1110 (非对称 PWM 模式 0)、1111 (非对称 PWM 模式 1)

在 PWM 模式 1 下, 当 $TIM_CNT < CCRx$, PWM 参考信号 OCxREF 便为高电平, 否则为低电平, 如下图所示。CCxIF 的产生方式依赖于配置 CMS, 图中所表示的是在 CMS=11 情况下, CCxIF 不论计数方向均会生成。

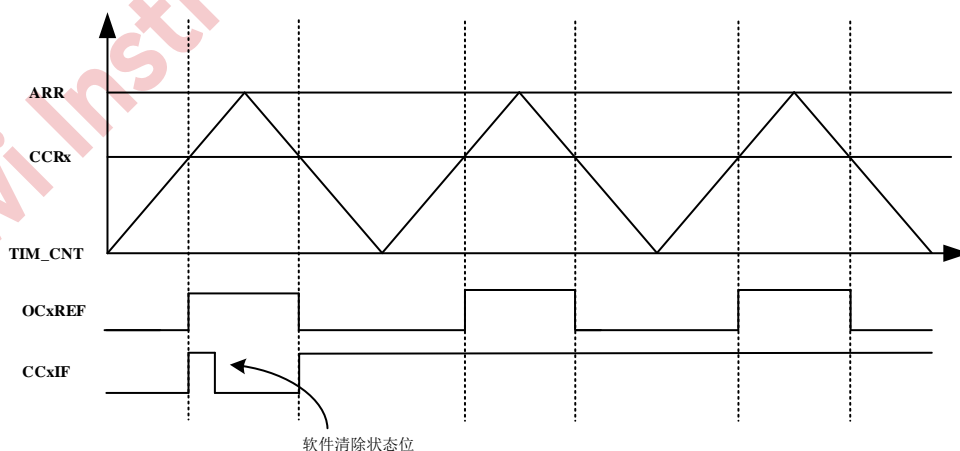


图 142 中心对齐模式的 PWM0 波形

下图列举了中心对齐模式的 PWM 波形，其中 $ARR=8$ ，PWM 模式为 PWM 模式 0，并根据 TIM_CR0 寄存器中 $CMS=01$ 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

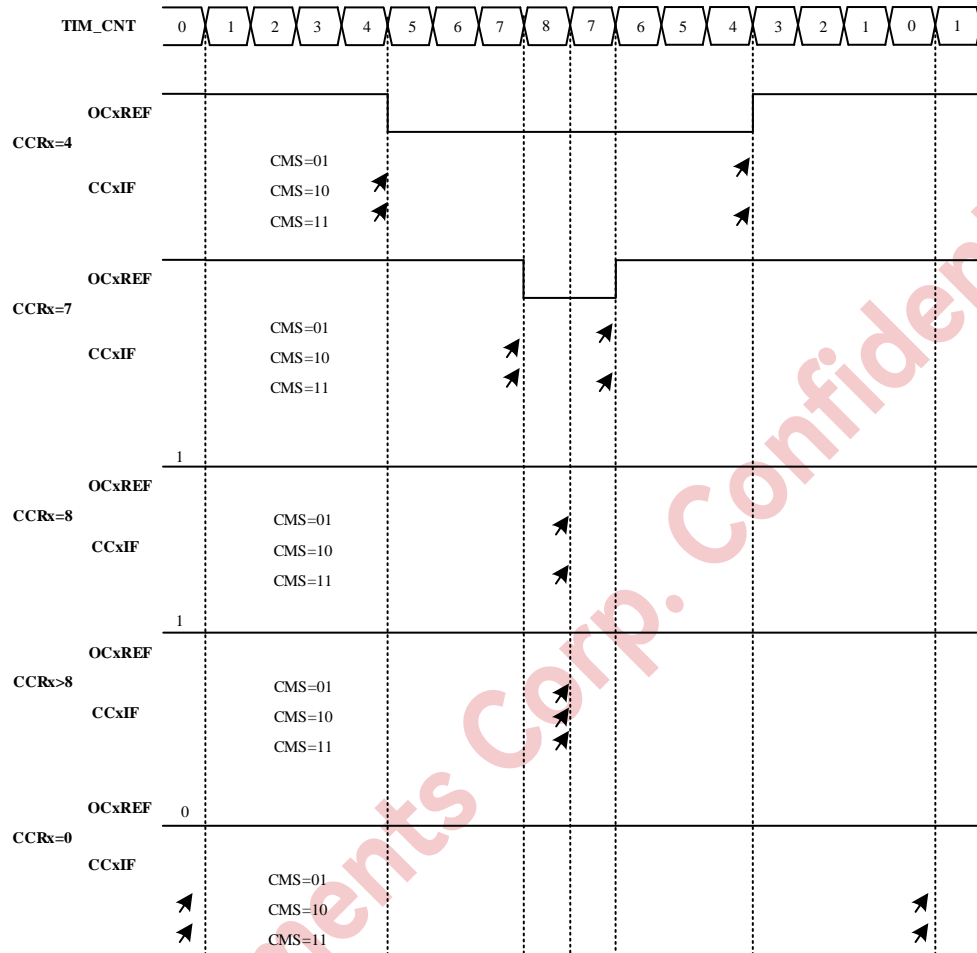


图 143 中心对齐模式 PWM 波形 (ARR=8)

对于非对称 PWM 模式 1，当计数器递增时，只要 $TIM_CNT < CCRx$ ，PWM 参考信号 $OCxREF$ 便为高电平，否则为低电平；当计数器递减时， $TIM_CNT < ACRx$ ，PWM 参考信号 $OCxREF$ 便为高电平，否则为低电平。如下图所示， $CCxIF$ 及 $ACxIF$ 的产生方式依赖于配置 CMS。

- CMS=01, $CCxIF$ 在计数器递增 $TIM_CNT=CCRx$ 时生成, $ACxIF$ 不生成。
- CMS=10, $ACxIF$ 在计数器递减 $TIM_CNT=ACRx$ 时生成, $CCxIF$ 不生成。
- CMS=11, $CCxIF$ 在计数器递增 $TIM_CNT=CCRx$ 时生成, $ACxIF$ 在计数器递减 $TIM_CNT=ACRx$ 时生成。

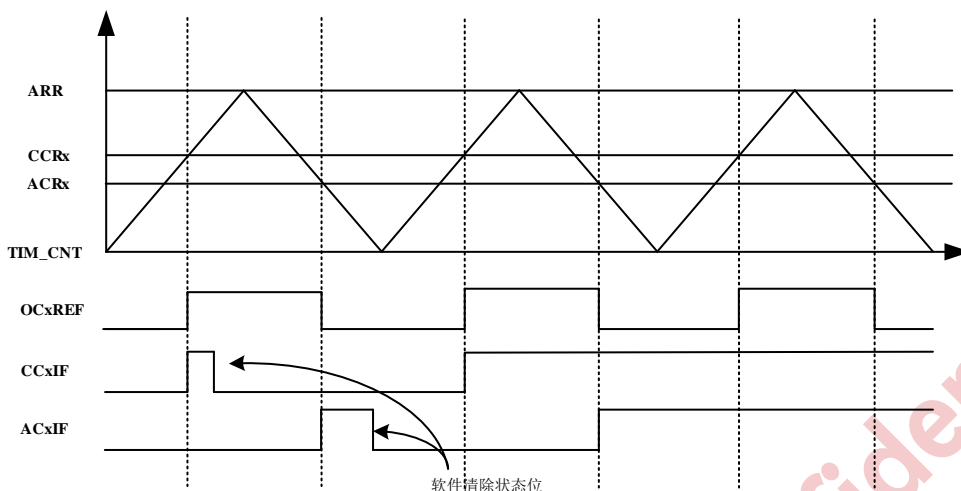


图 144 中心对齐模式的非对称 PWM0 波形

15.4.9.3 中心对齐模式使用建议

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIM_CR0 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：
 - 如果写入计数器中的值大于自动重载值 ($TIM_CNT > ARR$)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
 - 如果向计数器写入 0 或 ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIM_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

15.4.10 互补输出和死区插入

定时器可以输出两路互补信号，并管理输出的关断与接通瞬间。这段时间通常称为死区，用户必须根据与输出相连接的器件及其特性（电平转换器的固有延迟、开关器件产生的延迟...）来调整死区时间。

每路输出可以独立选择输出极性（主输出 OCx 或互补输出 OCxN）。可通过对 TIM_CCER 寄存器中的 CC1P 和 CC1NP 位执行写操作来完成极性选择。

互补信号 OC1 和 OC1N 通过以下多个控制位的组合进行激活：TIM_CCER 中的 CC1E 和 CC1NE 位以及 TIM_BDTR 和 TIM_CR1 中的 MOE、OISx、OISxN、HIZx 和 HIZxN 位。更多详细信息，请参照下表：具有断路功能的互补通道 OC1 和 OC1N 的输出控制位。应当注意，切换至 IDLE (MOE 下降到 0) 的时刻，死区仍然有效。

CC1E 和 CC1NE 位同时置 1 并且 MOE 位置 1 (如果存在断路) 时, 将使能死区插入。TIM_BDTR 寄存器中的 DTG[7:0]位和 DTGP[7:0] 用于控制所有通道的死区生成。其中, DTG 用于 OCxREF 下降沿处死区时间的插入, DTGP 用于 OCxREF 上升沿处死区时间的插入。OCxREF 上升沿和下降沿的死区时间配置也可统一由 DTG 配置控制, 这依赖于死区模式控制寄存器 DTGM, 默认情况下, 上下沿的死区时间配置是共用 DTG 配置。

DTG 和 DTGP 支持预装载功能, 通过配置 DTGPE 可启用该功能。每当 UEV 生效时, DTG 和 DTGP 的配置由影子寄存器加载到真正起作用的寄存器中。

将基于参考波形 OCxREF 生成 2 个输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效:

- 输出信号 OCx 与参考信号相同, 只是其上升沿相对参考上升沿存在延迟。
- 输出信号 OCxN 与参考信号相反, 并且其上升沿相对参考下降沿存在延迟。

如果延迟时间大于有效输出 (OCx 或 OCxN) 的宽度, 则不会产生相应的脉冲。下图所示为死区发生器的输出信号与参考信号 OCxREF 之间的关系。(在这些示例中, 假定 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1)

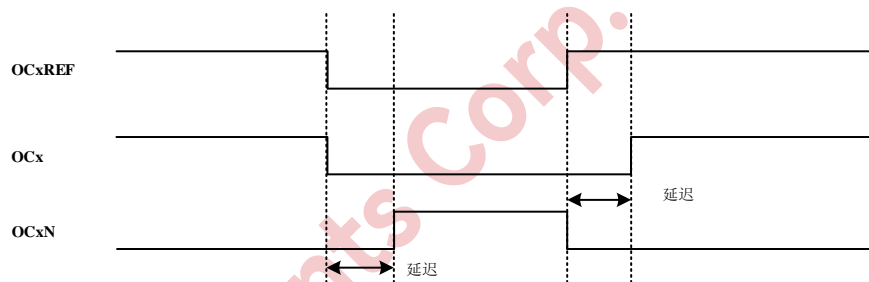


图 145 带死区插入的互补输出

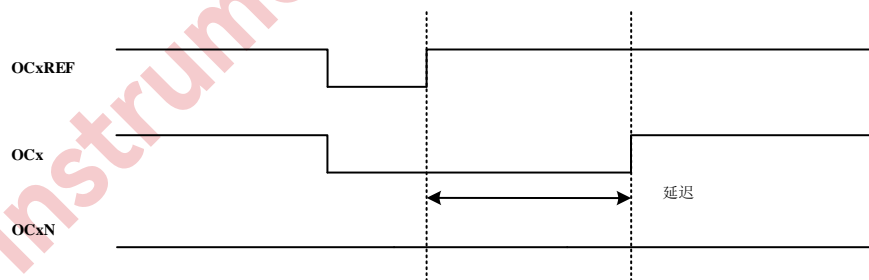


图 146 延迟时间大于负脉冲宽度的死区波形

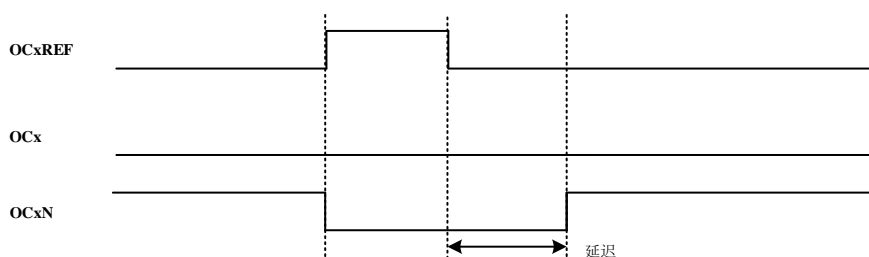


图 147 延迟时间大于正脉冲宽度的死区波形

死区延迟对于所有通道均相同，可通过 TIM_BDTR 寄存器中的 DTG 位进行编程。有关延迟时间计算的信息，请参见断路和死区寄存器 (TIM_BDTR) 描述。

将 OC1REF 重定向到 OC1 或 OC1N

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIM_CCER 寄存器中的 CC1E 和 CC1NE 位，可将 OC1REF 重定向到 OC1 输出或 OC1N 输出。

通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者，使两个输出同时保持无效电平，或者两个输出同时处于有效电平，两者互补并且带死区。

注：如果仅使能 OC1N (CC1E=0, CC1NE=1)，两者不互补，一旦 OC1REF 为高电平，OC1N 即变为有效。例如，如果 CC1NP=0，则 OC1N=OC1REF。另一方面，如果同时使能 OC1 和 OC1N (CC1E=CC1NE=1)，OC1 在 OC1REF 为高电平时变为有效，而 OC1N 则与之互补，在 OC1REF 为低电平时变为有效。

15.4.11 使用断路功能

使用断路功能时，根据相关控制位 (TIM_BDTR 中的 MOE 位以及 TIM_CR1 中的 OISx、OISxN、HIZx 和 HIZxN 位) 修改输出使能信号、无效电平或者高阻输出。任何情况下，OCx 和 OCxN 输出都不能同时置为有效电平。更多详细信息，请参见表格：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位。

断路源可以是断路输入引脚，也可以是时钟故障事件。

退出复位状态后，断路功能处于禁止状态，MOE 位处于低电平。将 TIM_BDTR 寄存器中的 BKE 位置 1，可使能断路功能。

由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 TIM_BDTR 寄存器中）之间插入了再同步电路，从而在异步信号与同步信号之间产生延迟。具体而言，如果在 MOE 处于低电平时向其写入 1，则必须首先插入延迟（空指令），才能准确进行读取。这是因为写入的是异步信号，而读取的却是同步信号。

发生断路（断路输入上出现所选电平）时：

- MOE 位异步清零，使输出处于无效状态、空闲状态或复位状态。即使定时器时钟关闭，该功能仍然有效。
- MOE=0 时，将以 TIM_CR1 寄存器 OISx 位中编程的电平驱动每个输出通道。
- MOE=0 时，将以 TIM_CR1 寄存器 HIZx 和 HIZxN 位中编程的电平控制每个输出通道使能已达到高阻输出的目标。
- 将断路状态标志 (TIM_SR 寄存器中的 BIF 位) 置 1。如果 TIM_IER 寄存器中的 BIE 位置 1，可产生中断。

- 如果 TIM_BDTR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件(UEV) 时自动再次置 1。这一特性有许多用处，比如，可用于实现调节器的功能。否则，MOE 将始终保持低电平，直到再次向该位写入 1。这种情况下，这一特性可用于确保安全。可以将断路输入连接到功率驱动器的警报、温度传感器或任何安全元件。

断路输入为电平有效。因此，当断路输入有效电平时，不能将 MOE 位置 1 (自动或通过软件)。同时，不能将状态标志 BIF 清零。

断路可由 BRK 输入生成，该输入具有可编程极性，其使能位 BKE 位于 TIM_BDTR 寄存器中。断路有以下两种生成方案：

- 使用 BRK 输入生成，该输入具有可编程极性，其使能位 BKE 位于 TIM_BDTR 中。
- 由软件通过 TIM_EGR 中的 BG 位生成。

除断路输入和输出管理外，断路电路内部还实施了写保护，用以保护应用的安全。通过该功能，用户可冻结多个参数配置（死区持续时间、OCx/OCxN 极性和禁止时的状态、OCxM 配置、断路使能和极性）。可以通过 TIM_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参见章节：断路和死区寄存器 (TIM_BDTR)。MCU 复位后只能对 LOCK 位执行一次写操作。

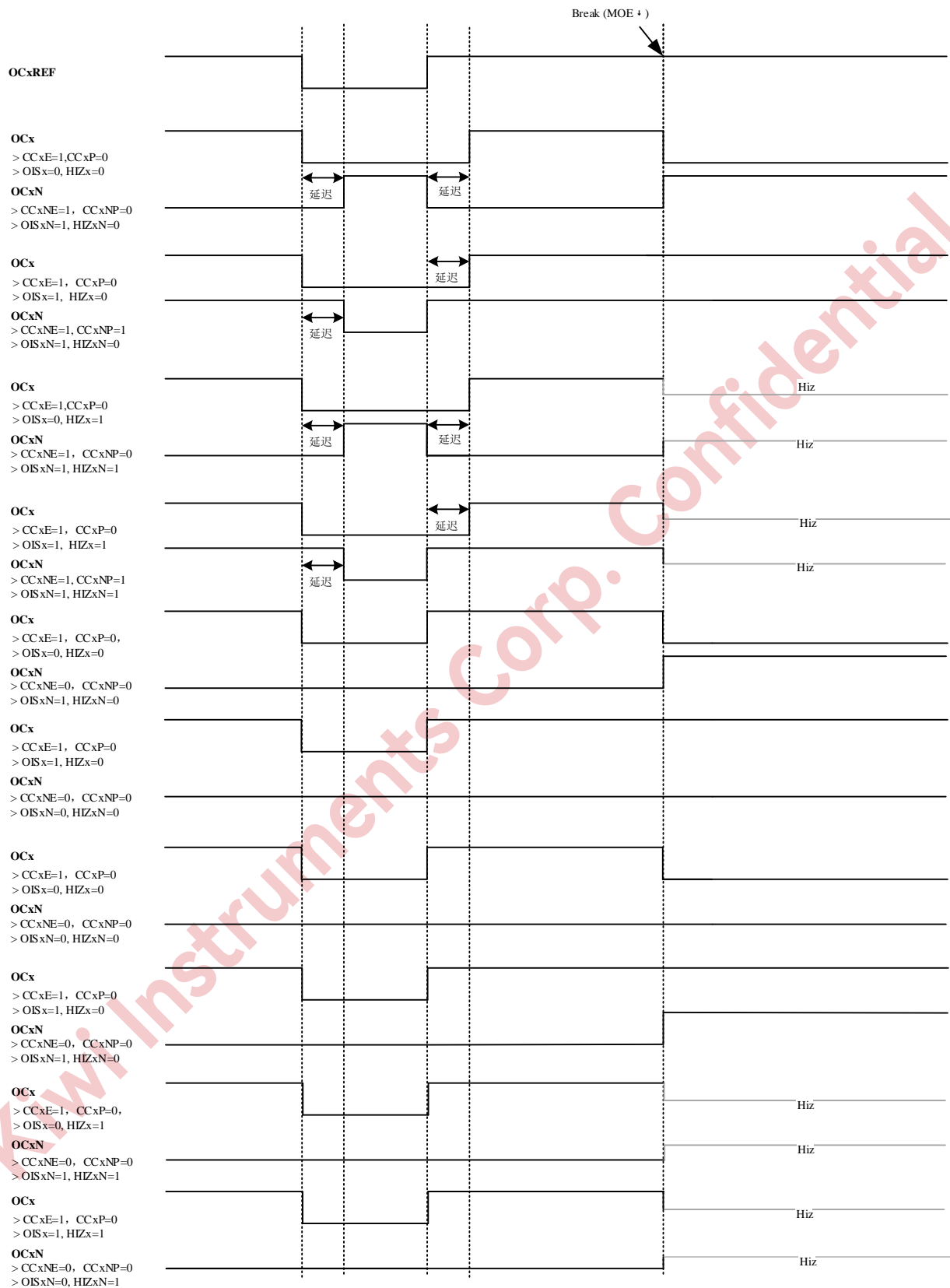


图 148 输出的断路响应行为 (通道 0~2)

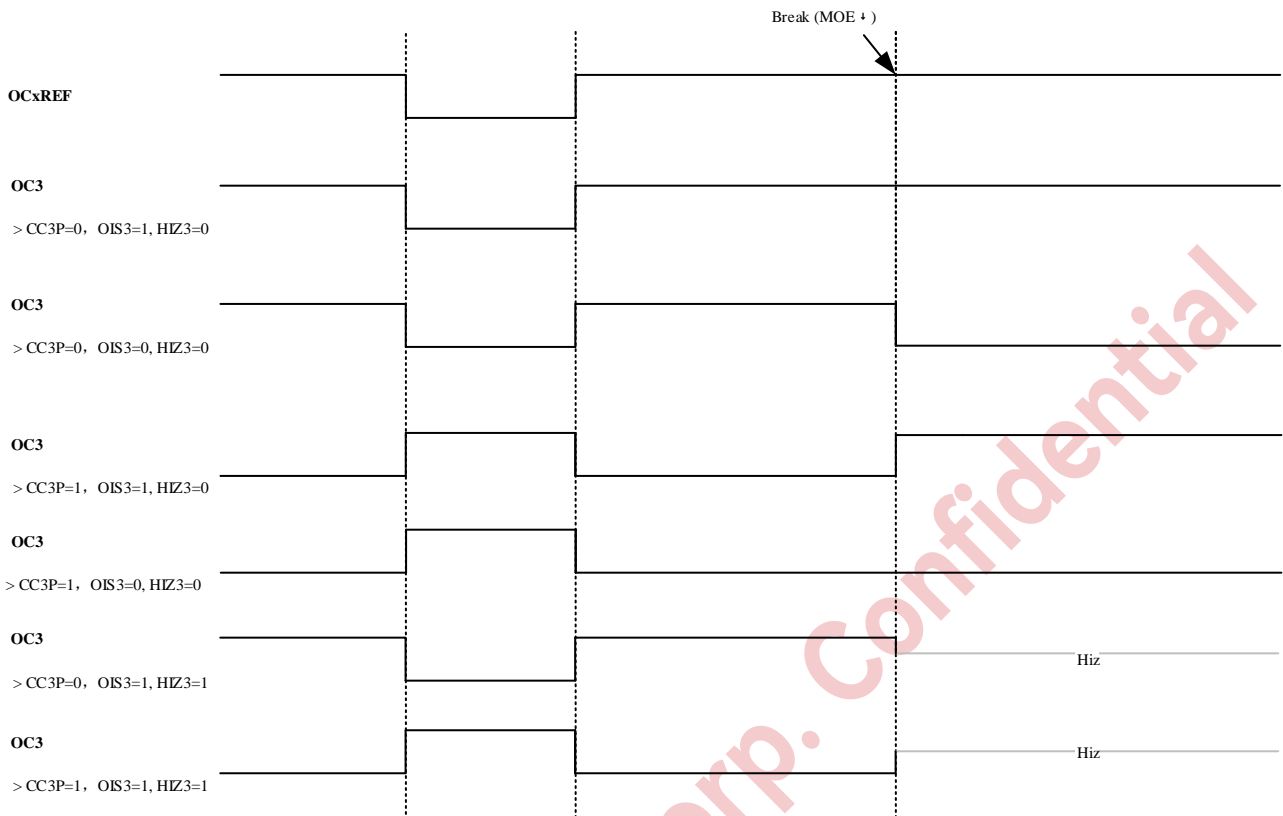


图 149 输出的断路响应行为 (通道 3)

15.4.12 发生外部事件时清除 OCxREF 信号

对于给定通道，在 ETRF 输入施加高电平（相应 TIM_CCMRx 寄存器中的 OCxCE 使能位置“1”），可使 OCxREF 信号变为低电平。OCxREF 信号将保持低电平，直到发生下一更新事件 (UEV)。

此功能仅能用于输出比较模式和 PWM 模式，而不适用于强制输出模式。

例如，ETR 信号可以连接到比较器的输出，用于控制电流。此时，ETR 必须如下配置：

1. 必须关闭外部触发预分频器：TIM_SMCR 寄存器中的 ETPS[1:0] 位置“00”。
2. 必须禁止外部时钟模式 1：TIM_SMCR 寄存器中的 ECE 位置“0”。
3. 外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可根据用户需要进行配置。

下图对比了使能位 OCxCE 在不同值下的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 TIM 编程为 PWM 模式。

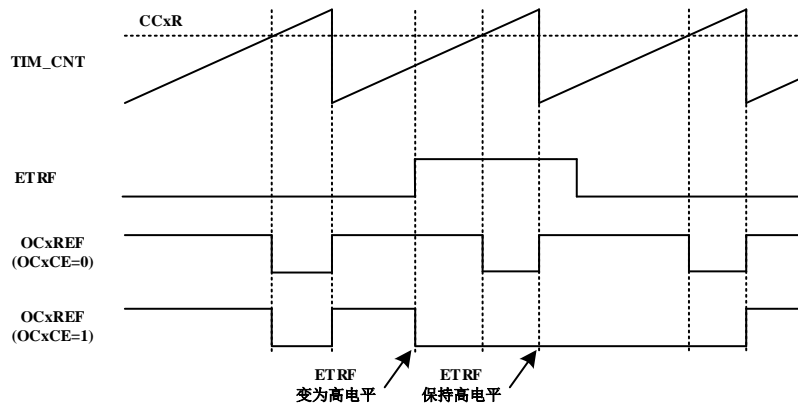


图 150 清除 TIM 的 OCxREF

如果 PWM 的占空比为 100% ($CCR_x > ARR$)，则下次计数器溢出时会再次使能 OCxREF。

15.4.13 生成 6 步 PWM

当通道使用互补输出时，OCxM、CCxE 和 CCxNE 位上提供预装载位。发生 COM 换向事件时，这些预装载位将传输到影子位。因此，用户可以预先编程下一步骤的配置，并同时更改所有通道的配置。COM 可由软件通过将 TIM_EGR 寄存器中的 COM 位置 1 而生成，也可以由硬件在 TRGI 上升沿生成。

发生 COM 事件时，某个标志位 (TIM_SR 寄存器中的 COMIF 位) 将会置 1。这时，如果 TIM_DIER 寄存器中的 COMIE 位置 1，将产生中断；如果 TIM_DIER 寄存器中的 COMDE 位置 1，则将产生 DMA 请求。

下图以 3 种不同的编程配置为例，显示了发生 COM 事件时 OCx 和 OCxN 输出的行为。

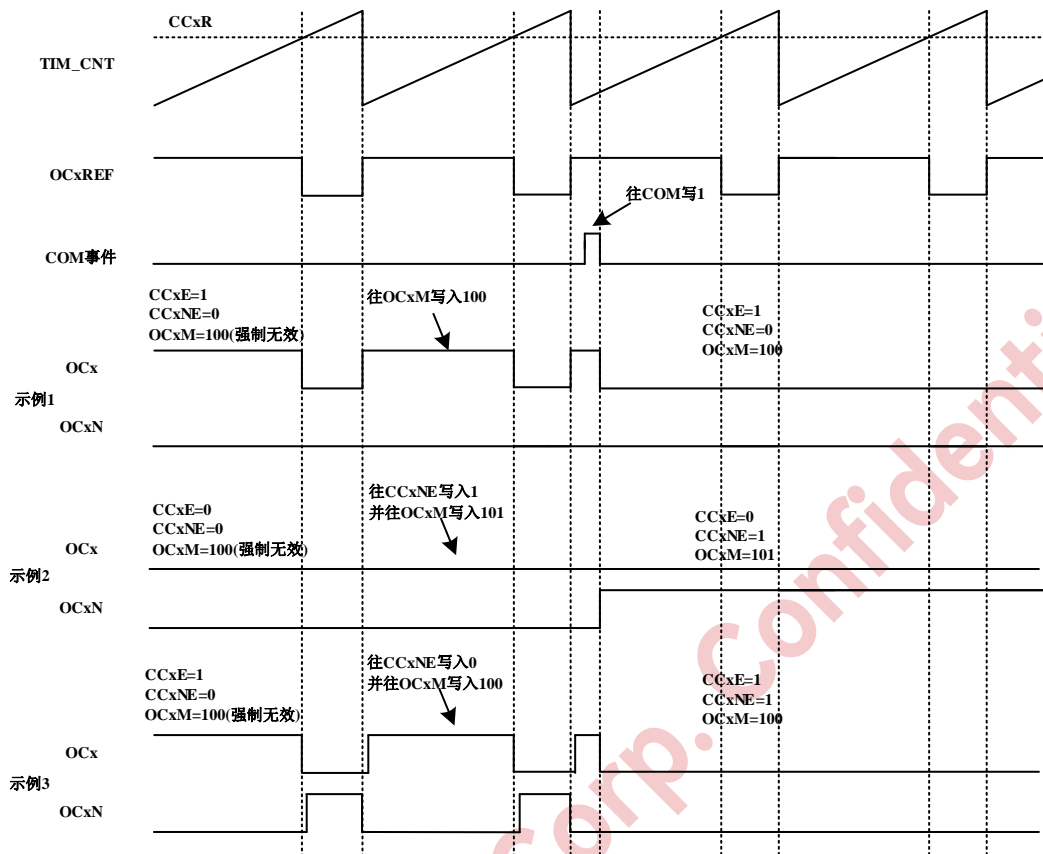


图 151 COM 事件生成 6 步 PWM 的示例

15.4.14 单脉冲模式

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

可通过从模式控制器启动计数器并在输出比较模式或 PWM 模式下生成波形。将 TIM_CR0 寄存器中的 OPM 位置 1，即可选择单脉冲模式。在发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- 递增计数模式下， $CNT < CCRx \leq ARR$ (特别注意， $0 < CCRx$)
- 递减计数模式下， $CNT > CCRx$

例如，用户希望达到这样的效果：在 TI1 输入引脚检测到正沿时，经过 tDELAY 的延迟，在 OC1 上产生一个长度为 tPULSE 的正脉冲。使用 TI1FP1 作为触发：

1. 在 TIM_CCMR0 寄存器中写入 CC1S=01，将 TI1FP1 映射到 TI1。
2. 在 TIM_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，使 TI1FP1 能够检测上升沿。

3. 在 TIM_SMCR 寄存器中写入 TS=110，将 TI1FPT 配置为从模式控制器的触发 (TRGI)。
4. 在 TIM_SMCR 寄存器中写入 SMS=110 (触发模式)，使用 TI1FPT 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义 (考虑时钟频率和计数器预分频器)。

- tDELAY 由写入 TIM_CCR0 寄存器的值定义。
- tPULSE 由自动重载值与比较值 (TIM_ARR-TIM_CCR0) 之差来定义。
- 欲生成如此波形：信号在发生比较匹配时从 0 变为 1，在计数器达到自动重载值时由 1 变为 0。为此，应在 TIM_CCMR0 寄存器中写入 OC0M=111，以使能 PWM 模式 1。如果需要，可选择在 TIM_CCMR0 寄存器的 OC0PE 和 TIM_CR0 寄存器的 ARPE 中写入 1，以使能预装载寄存器。这种情况下，必须在 TIM_CCR0 寄存器中写入比较值并在 TIM_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI1 上的外部触发事件。本例中，CC1P 的值为 0。

由于仅需要 1 个脉冲 (单脉冲模式)，因此应向 TIM_CR0 寄存器的 OPM 位写入 1，以便在发生下一更新事件 (计数器从自动重载值返回到 0) 时使计数器停止计数。TIM_CR0 寄存器中的 OPM 位设置为 0 时，即选择重复模式。

在单脉冲模式下，TIM_TIx 输入的边沿检测会将 TIM_CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟 (t_{DELAY} 最小值)。

如果要输出延迟时间最短的波形，可以将 TIM_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 OCxREF (和 OCx) 对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM 模式 0 或 PWM 模式 1 时，OCxFE 才会起作用。

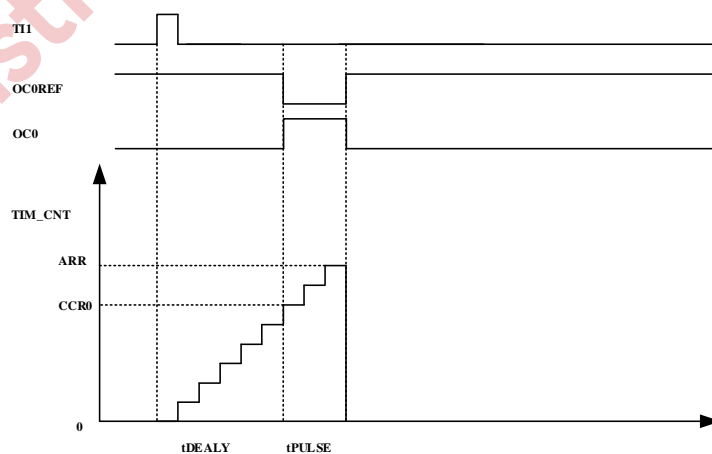


图 152 单脉冲模式示例

15.4.15 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI1 边沿处计数，在 TIM_SMCR 寄存器中写入 QEM=“001”；如果计数器仅在 TI0 边沿处计数，写入 QEM=“010”；如果计数器在 TI0 和 TI1 边沿处均计数，则写入 QEM=“011”。

通过编程 TIM_CCER 寄存器的 CC0P 和 CC1P 位，选择 TI0 和 TI1 极性。如果需要，还可对输入滤波器进行编程。CC0NP 和 CC1NP 必须保持低电平。

TI0 和 TI1 两个输入用于连接增量编码器。请参见下表。如果使能计数器（在 TIM_CR0 寄存器的 CEN 位中写入“1”），则计数器的时钟由 TIOFPT 或 TI1FPT 上的每次有效信号转换提供。TIOFPT 和 TI1FPT 是进行输入滤波器和极性选择后 TI0 和 TI1 的信号，如果不进行滤波和反相，则 TIOFPT=TI0，TI1FPT=TI1。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 TIM_CR0 寄存器的 DIR 位进行相应修改。任何输入（TI0 或 TI1）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI0 或 TI1 边沿处计数，还是同时在 TI0 和 TI1 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIM_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIM_ARR。同样，捕获、比较、预分频器、重复计数器及触发输出功能继续正常工作。编码器模式和外部时钟模式 1 不兼容，因此不能同时选择。

在此模式下，计数器会根据增量编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI0 和 TI1 不同时切换）。

表 218 计数方向与编码器信号的关系

有效边沿	相反信号的电平 (TIM_TI0->TIM_TI1, TIM_TI1->TIM_TI0)	TIM_TI0 信号		TIM_TI1 信号	
		上升	下降	上升	下降
仅在 TIM_TI0 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TIM_TI1 处计数	高	不计数	不计数	递减	递增
	低	不计数	不计数	递增	递减
在 TI0 和 TI1 处计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

外部增量编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码器输出可与外部中断输入相连，用以触发计数器复位。

下图以计数器工作为例，说明了计数信号的生成和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S="01" (TIM_CCMR0 寄存器，TI0FPT 映射到 TI0 上)。
- CC2S="01" (TIM_CCMR1 寄存器，TI0FPT 映射到 TI1 上)。
- CC1P="0"，CC1NP="0"，且 IC0F="0000" (TIM_CCER 寄存器，TI0FPT 未反相，TI0FPT=TI0)。
- CC1P="0"，CC1NP="0"，且 IC1F="0000" (TIM_CCER 寄存器，TI0FPT 未反相，TI0FPT=TI1)。
- QEM="011" (TIM_SMCR 寄存器，两个输入在上升沿和下降沿均有效)。
- CEN="1" (TIM_CR0 寄存器，使能计数器)。

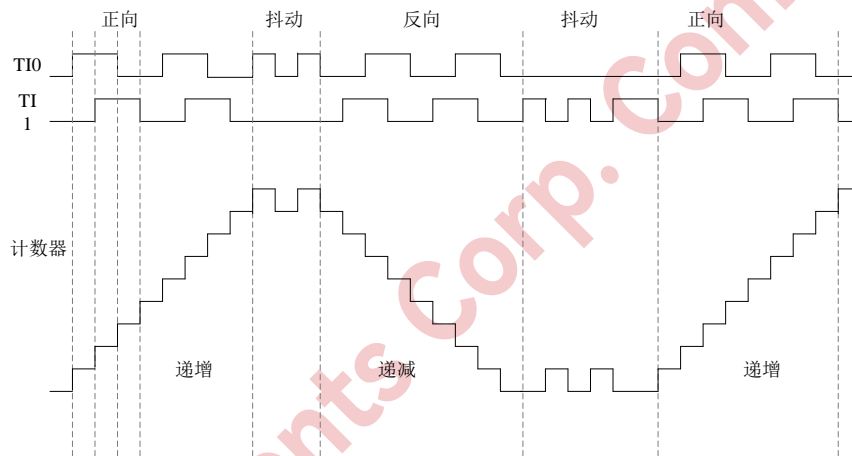


图 153 编码器接口模式下的计数器工作示例

下图举例说明 TI0FPT 极性反相时计数器的行为 (除 CC0P="1" 外，其它配置与上例相同)。

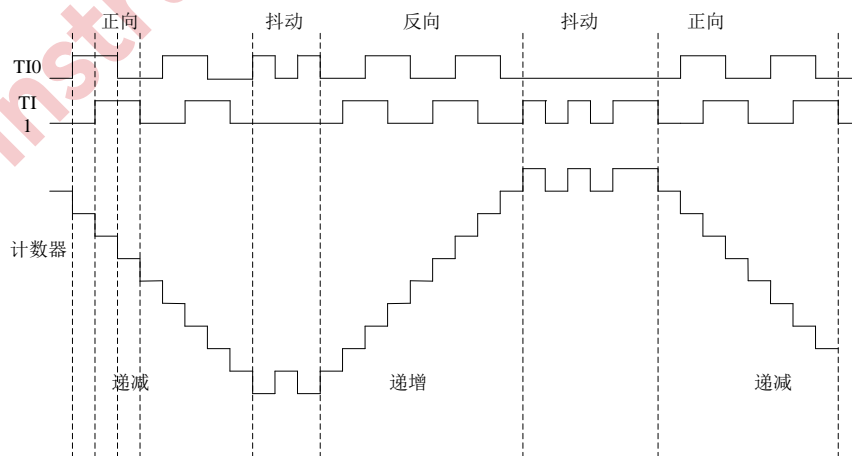


图 154 TI0FPT 极性反相时的编码器接口模式示例

定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）；还可以通过由实时时钟生成的 DMA 请求读取计数器值。

15.4.16 定时器输入异或功能

通过 TIM_CR1 寄存器中的 TI0S 位，可将通道 0 的输入滤波器连接到异或门的输出，从而将 TIM_CH0 到 TIM_CH2 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。下面章节以连接霍尔传感器为例介绍了此功能。

15.4.17 连接霍尔传感器

可通过用于生成电机驱动 PWM 信号的高级控制定时器 (TIM1) 以及下图中称为“接口定时器”的另一个定时器 TIM (TIM2)，实现与霍尔传感器的连接。3 个定时器输入引脚 (TIM_CH0、TIM_CH1 和 TIM_CH2) 通过异或门连接到 TI0 输入通道（通过将 TIM_CR1 寄存器中的 TI0S 位置 1 来选择），并由“接口定时器”进行捕获。

从模式控制器配置为复位模式：从输入为 TI0F_ED。这样，每当 3 个输入中有一个输入发生切换时，计数器会从 0 开始重新计数。这样将产生由霍尔输入的任何变化而触发的时基。

在“接口定时器”上，捕获/比较通道 0 配置为捕获模式，捕获信号为 TRC (请参见章节捕获/比较通道 (例如：通道 0 输入阶段))。捕获值对应于输入上两次变化的间隔时间，可提供与电机转速相关的信息。

“接口定时器”可用于在输出模式下产生脉冲，以通过触发 COM 事件更改高级控制定时器 (TIM1) 各个通道的配置。TIM1 定时器用于生成电机驱动 PWM 信号。为此，必须对接口定时器通道进行编程，以便在编程的延迟过后产生正脉冲（在输出比较或 PWM 模式中）。该脉冲通过 TRGO 输出发送到高级控制定时器 (TIM1)。

示例：霍尔输入与一个 TIM 定时器相连接，每当霍尔输入发生更改，需要在所编程的延迟过后更改高级控制定时器 TIM1 的 PWM 配置。

- 向 TIM_CR0 寄存器的 TI0S 位写入“1”，使 3 个定时器输入经过异或运算后进入 TI0 输入通道。
- 时基编程：向 TIM_ARR 写入其最大值（计数器必须通过 TI0 的变化清零）。设置预分频器，以得到最大计数器周期，该周期长于传感器上两次变化的间隔时间。
- 将通道 0 编程为捕获模式 (选择 TRC)：向 TIM_CCMR0 寄存器的 CC0S 位写入“11”。如果需要，还可以编程数字滤波器。
- 将通道 0 编程为 PWM1 模式，并具有所需延迟：向 TIM_CCMR0 寄存器的 OC0M 位写入“111”，CC1S 位写入“00”。

- 选择 OC0REF 作为 TRGO 上的触发输出：向 TIM_CR0 寄存器的 MMS 位写入“101”。

在高级控制定时器 TIM1 中，必须选择正确的 ITR 输入作为触发输入，定时器编程为可产生 PWM 信号，捕获/比较控制信号进行预装载 (TIM_CR1 寄存器的 CCPC=1)，并且 COM 事件由触发输入控制 (TIM_CR1 寄存器中 CCUS=1)。发生 COM 事件后，在 PWM 控制位 (CCxE、OCxM) 中写入下一步的配置，此操作可在由 OC2REF 上升沿产生的中断子程序中完成。

下图为本示例的示意图。

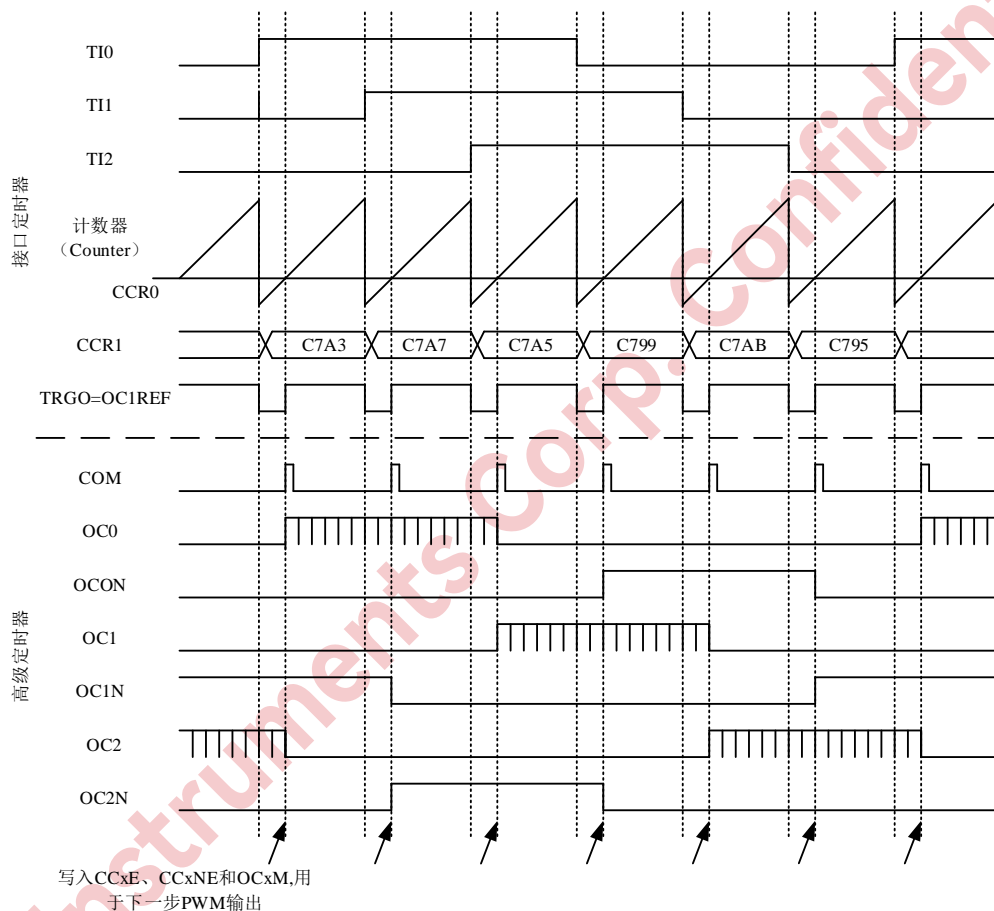


图 155 霍尔传感器接口的示例

15.4.18 定时器与外部触发同步

定时器可与外部触发以下列模式实现同步：复位模式、门控模式和触发模式。

15.4.18.1 复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIM_CR0 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器 (TIM_ARR 和 TIM_CCRx) 都将更新。

在下例中，TI1 输入上出现上升沿时，计数器清零：

1. 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中只需要默认采样滤波，故保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIM_CCMR1 寄存器中的 CC1S=01。在 TIM_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，以验证极性（仅检测上升沿）。
2. 在 TIM_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIM_SMCR 寄存器中写入 TS=110，选择 TI1 作为输入源。
3. 在 TIM_CR0 寄存器中写入 TIM_CEN=1，启动计数器。

计数器使用内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时触发标志（TIM_SR 寄存器中的 TIF 位）置 1，使能中断后，还可发送中断请求（如果 TIE 使能）。

下图显示了自动重载寄存器 TIM_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

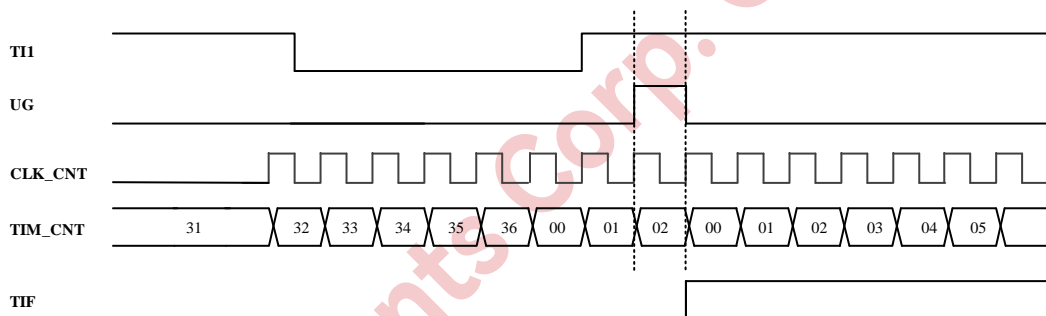


图 156 复位模式下的控制电路

15.4.18.2 门控模式

输入信号的电平可用来使能计数器。在以下示例中，计数器仅在 TI1 输入为低电平时计数：

1. 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIM_CCMR0 寄存器中的 CC1S=01。在 TIM_CCER 寄存器中写入 CC1P=1，以确定极性（仅检测低电平）。
2. 在 TIM_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIM_SMCR 寄存器中写入 TS=110，选择 TI1 作为输入源。
3. 在 TIM_CR0 寄存器中写入 TIM_CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIM_SR 寄存器中的 TIF 标志都会置 1。TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

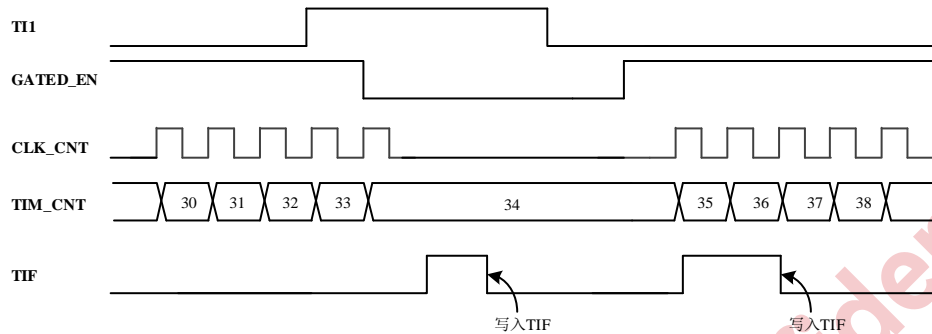


图 157 门控模式下的控制电路

15.4.18.3 触发模式

所选输入上发生某一事件时可以用以启动计数器。以下示例中，TI1 输入上出现上升沿时，计数器便启动：

1. 将通道 1 配置为检测 TI1 上的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIM_CCMR0 寄存器中的 CC1S=01。在 TIM_CCER 中写入 CC1P=1，以确定极性（仅检测低电平）。
2. 在 TIM_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIM_MCR 寄存器中写入 TS=110，选择 TI1 作为输入源。

当 TI1 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。TI1 的上升沿与实际计数器启动之间的延迟是由于 TI1 输入的重新同步电路引起的。

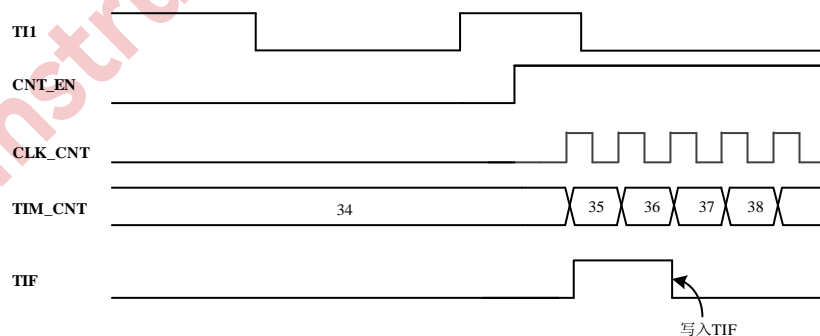


图 158 触发模式下的控制电路

15.4.19 定时器同步

定时器从内部连接在一起，以实现定时器同步或级联。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

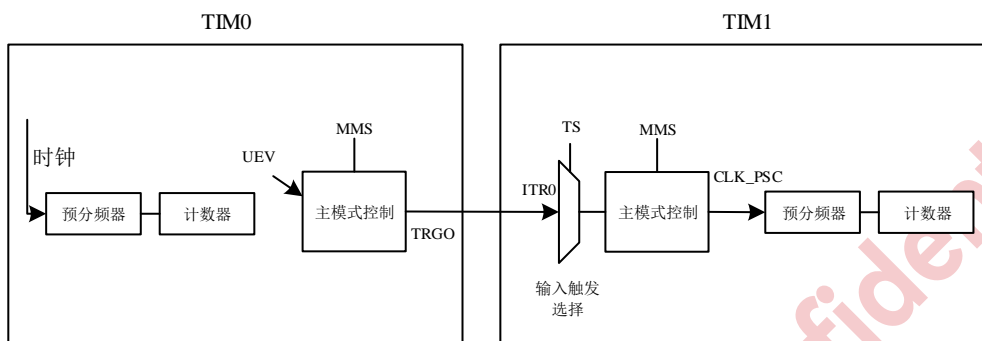


图 159 主/从定时器示例

具体可参照简版定时器 TIMS 的定时器同步章节。

15.4.20 TIMA 中断号

TIMA 的中断连接至不可屏蔽中断处理模块 (NVIC) 的 IRQ 相关接口。为了启用该中断功能，需要按照以下流程进行操作：

- 使能 TIM 的中断使能位
- 使能 NVIC 中的 TIM IRQ 通道, TIM 的系统中断号如下表所示.
- 配置 TIM 以产生中断
- TIM 对应的中断号如下

表 219 TIMA 中断表

名称	中断号
TIMA0_INT	32
TIMA1_INT	33

15.5 寄存器

TIMA0~1 寄存器基地址分别是 0x40040000 和 0x40040400。下表为 TIMAx 的各控制寄存器描述。

表 220 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	TIM 控制寄存器 0	32	r/w	0x00000000
0x04	TIM 控制寄存器 1	32	r/w	0x00000000
0x08	TIM 从属模式控制寄存器	32	r/w	0x00000000
0x0C	TIM DMA&中断使能寄存器	32	r/w	0x00000000
0x10	TIM 状态寄存器	32	r/w	0x00000000
0x14	TIM 事件生成寄存器	32	r/w	0x00000000
0x18	TIM 捕获/比较模式寄存器 0	32	r/w	0x00000000
0x1C	TIM 捕获/比较模式寄存器 1	32	r/w	0x00000000
0x20	TIM 捕获/比较模式寄存器 2	32	r/w	0x00000000
0x24	TIM 捕获/比较使能寄存器	32	r/w	0x00000000
0x28	TIM 计数器	32	r/w	0x00000000
0x2C	TIM 预分频器	32	r/w	0x00000000
0x30	TIM 自动重载寄存器	32	r/w	0x0000FFFF
0x34	TIM 重复计数器寄存器	32	r/w	0x00000000
0x38	TIM 捕获/比较寄存器 0	32	r/w	0x00000000
0x3C	TIM 捕获/比较寄存器 1	32	r/w	0x00000000
0x40	TIM 捕获/比较寄存器 2	32	r/w	0x00000000
0x44	TIM 捕获/比较寄存器 3	32	r/w	0x00000000
0x48	TIM 捕获/比较寄存器 4	32	r/w	0x00000000
0x4C	TIM 捕获/比较寄存器 5	32	r/w	0x00000000
0x50	TIM 断路及死区寄存器	32	r/w	0x00000000
0x54~0x5C	保留	32	RES	0x00000000
0x60	TIM 事件控制寄存器	32	r/w	0x00000000
0x64	TIM 配置秘钥寄存器	32	r/w	0x00000000
0x68~0xFC	保留	32	RES	0x00000000

关于输入复用关系和请查阅设计文档《定时器选项》中相关寄存器描述。

15.5.1 TIM 控制寄存器 0 (TIM_CR0)

偏移地址：0x0000

表 221 TIM 控制寄存器 0

位	访问	描述
31:11	Res	Reserved 复位值：0x0
10	ro	CNT_ST：计数器停止运行标记 0：计数器停止计数 1：计数器尚在计数 复位值：0x0
9:8	r/w	CKD：时钟分频 此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器 (Tl _x) 所使用的死区及采样时钟 (t _{DTS}) 之间的分频比。 00: t _{DTS} = t _{CK_INT} 01: t _{DTS} = 2*t _{CK_INT} 10: t _{DTS} = 4*t _{CK_INT} 11: t _{DTS} = t _{CK_INT} 复位值：0x0
7	r/w	ARPE：自动重载预装载使能 0：TIM_ARR 寄存器不进行缓存 1：TIM_ARR 寄存器进行缓存 复位值：0x0
6:5	r/w	CMS[1:0]：中心对齐模式选择 00：边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。 01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道 (TIM_CCMRx 寄存器中的 CxS=00) 的输出比较中断标志才置 1。 10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道 (TIM_CCMRx 寄存器中的 CxS=00) 的输出比较中断标志才置 1。 11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道 (TIM_CCMRx 寄存器中的 CxS=00) 的输出比较中断标志都会置 1。 注：只要计数器处于使能状态 (CEN=1)，就不得从边沿对齐模式切换为中心。 复位值：0x0
4	r/w	DIR：方向 (Direction) 0：计数器递增计数 1：计数器递减计数 注：当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。 复位值：0x0

位	访问	描述
3	r/w	OPM: 单脉冲模式使能控制 0: 计数器在发生更新事件时不会停止计数 1: 计数器在发生下一更新事件时停止计数 (将 CEN 位清零) 复位值: 0x0
2	r/w	URS: 更新请求源选择 0: 使能时, 所有以下事件都会生成更新中断请求。此类事件包括: <ul style="list-style-type: none"> 计数器上溢/下溢 将 UG 位置 1 通过从模式控制器生成的更新事件 1: 使能时, 只有计数器上溢/下溢会生成更新中断请求。 复位值: 0x0
1	r/w	UDIS: 更新禁止 此位由软件置 1 和清零, 用以使能/禁止 UEV 事件生成。 0: 使能 UEV。更新 UEV 事件可通过以下事件之一生成: <ul style="list-style-type: none"> 计数器上溢/下溢; 将 UG 位置 1; 通过从模式控制器生成的更新事件 然后更新影子寄存器的值。 1: 禁止 UEV。不生成更新事件, 各影子寄存器的值 (ARR、PSC 和 CCRx) 保持不变。但如果将 UG 位置 1, 或者从从模式控制器接收到硬件复位, 则会重新初始化计数器和预分频器。 复位值: 0x0
0	r/w	CEN: 计数器使能控制 0: 关闭 1: 开启 注意: 只有事先通过软件将 CEN 位置 1, 才可以使用外部时钟、门控模式。而触发模式可通过硬件自动将 CEN 位置 1。 复位值: 0x0

15.5.2 TIM 控制寄存器 1 (TIM_CR1)

偏移地址: 0x0004

表 222 TIM 控制寄存器 1

位	访问	描述
31:28	Res	Reserved 复位值: 0x0
27:24	Res	MMS1[3:0]: 主模式选择 1 (Master Mode Selection) 这些位可选择主模式下将要发送到 ADC 以实现同步的信息 (TRGO1)。这些位的组合如下: <ul style="list-style-type: none"> 0000: 复位-TIM_EGR 寄存器中的 UG 位用作触发输出 (TRGO1)。如果复位由触发输入生成 (从模式控制器配置为复位模式), 则 TRGO1 上的信号相比实际复位会有延迟。 0001: 使能-计数器使能信号 CNT_EN 用作触发输出 (TRGO1)。该触发输出可用于同时启动多个定时器, 或者控制在一段时间内使能从定时器。计数器使能信

位	访问	描述
		<p>号可由 CEN 控制位产生。当配置为门控模式时，也可由触发输入产生。当计数器使能信号由触发输入控制时，TRGO1 上会存在延迟，选择主/从模式时除外（请参见 TIM_SMCR 寄存器中 MSM 位的说明）。</p> <p>0010: 更新-选择更新事件作为触发输出 (TRGO1)。例如，主定时器可用作从定时器的预分频器。</p> <p>0011: 比较脉冲-一旦发生输入捕获或比较匹配事件，当 CC0IF 被置 1 时（即使已为高电平），触发输出都会发送一个正脉冲。(TRGO1)。</p> <p>0100: 比较-OC0REF 信号用作触发输出 (TRGO1)</p> <p>0101: 比较-OC1REF 信号用作触发输出 (TRGO1)</p> <p>0110: 比较-OC2REF 信号用作触发输出 (TRGO1)</p> <p>0111: 比较-OC3REF 信号用作触发输出 (TRGO1)</p> <p>1000: 比较-OC4REF 信号用作触发输出 (TRGO1)</p> <p>1001: 比较-OC5REF 信号用作触发输出 (TRGO1)</p> <p>1010: 比较脉冲-OC3REF 上下沿脉冲输出 (TRGO1)</p> <p>1011: 比较脉冲-OC5REF 上下沿脉冲输出 (TRGO1)</p> <p>1100: 比较脉冲-OC3REF 或 OC5REF 上沿脉冲输出 (TRGO1)</p> <p>1101: 比较脉冲-OC3REF 上沿脉冲或 OC5REF 下沿脉冲输出 (TRGO1)</p> <p>1110: 比较脉冲-OC4REF 或 OC5REF 上沿脉冲输出 (TRGO1)</p> <p>1111: 比较脉冲-OC4REF 上沿脉冲或 OC5REF 下沿脉冲输出 (TRGO1)</p> <p>复位值: 0x0</p>
23	Res	<p>Reserved</p> <p>复位值: 0x0</p>
22	r/w	<p>HIZ3: 高阻输出使能 3 (OC3 输出)</p> <p>参照 HIZ3 定义</p> <p>复位值: 0x0</p>
21	r/w	<p>HIZ2N: 高阻输出使能 2 (OC2N 输出)</p> <p>参照 HIZ0N 定义</p> <p>复位值: 0</p>
20	r/w	<p>HIZ2: 高阻输出使能 2 (OC2 输出)</p> <p>参照 HIZ0 定义</p> <p>复位值: 0x0</p>
19	r/w	<p>HIZ1N: 高阻输出使能 1 (OC1N 输出)</p> <p>参照 HIZ0N 定义</p> <p>复位值: 0x0</p>
18	r/w	<p>HIZ1: 高阻输出使能 1 (OC1 输出)</p> <p>参照 HIZ0 定义</p> <p>复位值: 0x0</p>
17	r/w	<p>HIZ0N: 高阻输出使能 0 (OC0N 输出)</p> <p>0: 当 MOE=0 时, OC0N 依据 OIS0N 输出</p> <p>1: 当 MOE=0 时, OC0N 输出高阻</p>

位	访问	描述
		注：只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1、2 或 3，此位即无法修改。 复位值：0x0
16	r/w	HIZ0：高阻输出使能 0 (OC0 输出) 0：当 MOE=0 时，OC0 依据 OIS0 输出 1：当 MOE=0 时，OC0 输出高阻 注：只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1、2 或 3，此位即无法修改。 复位值：0x0
15	Res	Reserved 复位值：0x0
14	r/w	OIS3：输出空闲状态 3 (OC3 输出) 参照 OIS1 位定义 复位值：0x0
13	r/w	OIS2N：输出空闲状态 2 (OC2N 输出) 参照 OIS1N 位定义 复位值：0x0
12	r/w	OIS2：输出空闲状态 2 (OC2 输出) 参照 OIS1 位定义 复位值：0x0
11	r/w	OIS1N：输出空闲状态 1 (OC1N 输出) 参照 OIS1N 位定义 复位值：0x0
10	r/w	OIS1：输出空闲状态 1 (OC1 输出) 参照 OIS1 位定义 复位值：0x0
9	r/w	OIS0N：输出空闲状态 0 (OC0N 输出) 0：当 MOE=0 和 HIZ0N=0 时，OC0N=0 1：当 MOE=0 和 HIZ0N=0 时，OC0N=1 注：只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1、2 或 3，此位即无法修改。 复位值：0x0
8	r/w	OIS0：输出空闲状态 0 (OC0 输出) 0：当 MOE=0 和 HIZ0=0 时，OC0=0 1：当 MOE=0 和 HIZ0=0 时，OC0=1 注：只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1、2 或 3，此位即无法修改。 复位值：0x0
7	r/w	TIOS：TI0 选择 (TI1 Selection) 0：TIM_CH0 引脚连接到 TI0 输入 1：TIM_CH0、CH1 和 CH2 引脚连接到 TI0 输入 (异或组合)

位	访问	描述
		复位值: 0x0
6:4	r/w	<p>MMS[1:0]: 主模式选择 (Master Mode Selection) 这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下: 000: 复位--TIM_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成 (从模式控制器配置为复位模式), 则 TRGO 上的信号相比实际复位会有延迟。 001: 使能--计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器, 或者控制在一段时间内使能从定时器。计数器使能信号可由 CEN 控制位产生。当配置为门控模式时, 也可由触发输入产生。当计数器使能信号由触发输入控制时, TRGO 上会存在延迟, 选择主/从模式时除外 (请参见 TIM_SMCR 寄存器中 MSM 位的说明)。 010: 更新--选择更新事件作为触发输出 (TRGO)。例如, 主定时器可用作从定时器的预分频器。 011: 比较脉冲--一旦发生输入捕获或比较匹配事件, 当 CC0IF 被置 1 时 (即使已为高电平), 触发输出都会发送一个正脉冲。(TRGO)。 100: 比较--OC0REF 信号用作触发输出 (TRGO) 101: 比较--OC1REF 信号用作触发输出 (TRGO) 110: 比较--OC2REF 信号用作触发输出 (TRGO) 111: 比较--OC3REF 信号用作触发输出 (TRGO) 复位值: 0x0</p>
3	r/w	<p>CCDS: 捕获/比较 DMA 选择(Capture/Compare DMA Selection) 0: 发生 CCx 事件时发送 CCxDMA 请求 1: 发生更新事件时发送 CCxDMA 请求 复位值: 0x0</p>
2	r/w	<p>CCUS: 捕获/比较控制更新选择 (Capture/Compare Control Update Selection) 0: 如果捕获/比较控制位 (CCPC=1) 进行预装载, 仅通过将 COMG 位置 1 来对这些位进行更新 1: 如果捕获/比较控制位 (CCPC=1) 进行预装载, 可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。 注意: 此位仅对具有互补输出的通道有效。 复位值: 0x0</p>
1	Res	<p>Reserved 复位值: 0x0</p>
0	r/w	<p>CCPC: 捕获/比较预装载控制 (Capture/Compare Preloaded Control) 0: CCxE、CCxNE 和 OCxM 位未进行预装载 1: CCxE、CCxNE 和 OCxM 位进行了预装载, 写入这些位后, 仅当发生换向事件 (COM) (COMG 位置 1 或在 TRGI 上检测到上升沿, 取决于 CCUS 位) 时才会对这些位进行更新。 注意: 此位仅对具有互补输出的通道有效。 复位值: 0x0</p>

15.5.3 TIM 从属模式控制寄存器 (TIM_SMCR)

偏移地址: 0x0008

表 223 TIM 从属模式控制寄存器

位	访问	描述
31:27	Res	Reserved 复位值: 0x0
26:24	r/w	QEM, 编码模式 000: 编码器模式关闭。 001: 编码器模式 1--计数器根据 TI0FP0 电平在 TI1FPT 边沿递增/递减计数。 010: 编码器模式 2--计数器根据 TI1FP1 电平在 TI0FPT 边沿递增/递减计数。 011: 编码器模式 3--计数器在 TI0FPT 和 TI1FPT 的边沿计数, 计数的方向取决于另外一个信号的电平。 其余值: 保留 复位值: 0x0
23:21	Res	Reserved 复位值: 0x0
20	r/w	OCCS: OCREF 清除源选择 0: OCREF_CLR_INT 源于 ETRF 1: OCREF_CLR_INT 源于 OCREF_CLR 复位值: 0x0
19:17	Res	Reserved 复位值: 0x0
16	r/w	ETEN: 外部触发使能 0: 关闭 1: 开启 复位值: 0x0
15	r/w	ETP: 外部触发极性 此位可选择将 ETR 还是 ETR 用于触发操作 0: ETR 未反相, 高电平或上升沿有效。 1: ETR 反相, 低电平或下降沿有效。 复位值: 0x0
14	r/w	ECE: 外部时钟使能 此位可使能外部时钟模式 1。 0: 禁止外部时钟模式 1 1: 使能外部时钟模式 1。计数器时钟由 ETRF 信号的任意有效边沿提供。 注: 1: 将 ECE 位置 1 与选择外部时钟模式 0 并将 TRGI 连接到 ETRF (SMS=111 且 TS=111) 具有相同效果。 2: 外部时钟模式 1 可以和以下从模式同时使用: 复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF (TS 位不得为 111)。 3: 如果同时使能外部时钟模式 0 和外部时钟模式 1, 则外部时钟输入为 ETRF。 复位值: 0x0

位	访问	描述
13:12	r/w	<p>ETPS[1:0]: 外部触发预分频器</p> <p>外部触发信号 ETRP 频率不得超过 TIM CLK 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。</p> <p>00: 预分频器关闭</p> <p>01: 2 分频 ETRP 频率</p> <p>10: 4 分频 ETRP 频率</p> <p>11: 8 分频 ETRP 频率</p> <p>复位值: 0x0</p>
11:8	r/w	<p>ETF[3:0]: 外部触发滤波器</p> <p>此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波时间。数字滤波器由事件计数器组成, 每 N 个事件才视为一个有效边沿:</p> <p>0000: 无滤波器, 按 f_{DTS} 频率进行采样</p> <p>0001: $f_{SAMPLING}=f_{CK_INT}$, $N=2$</p> <p>0010: $f_{SAMPLING}=f_{CK_INT}$, $N=4$</p> <p>0011: $f_{SAMPLING}=f_{CK_INT}$, $N=8$</p> <p>0100: $f_{SAMPLING}=f_{DTS}/2$, $N=6$</p> <p>0101: $f_{SAMPLING}=f_{DTS}/2$, $N=8$</p> <p>0110: $f_{SAMPLING}=f_{DTS}/4$, $N=6$</p> <p>0111: $f_{SAMPLING}=f_{DTS}/4$, $N=8$</p> <p>1000: $f_{SAMPLING}=f_{DTS}/8$, $N=6$</p> <p>1001: $f_{SAMPLING}=f_{DTS}/8$, $N=8$</p> <p>1010: $f_{SAMPLING}=f_{DTS}/16$, $N=5$</p> <p>1011: $f_{SAMPLING}=f_{DTS}/16$, $N=6$</p> <p>1100: $f_{SAMPLING}=f_{DTS}/16$, $N=8$</p> <p>1101: $f_{SAMPLING}=f_{DTS}/32$, $N=5$</p> <p>1110: $f_{SAMPLING}=f_{DTS}/32$, $N=6$</p> <p>1111: $f_{SAMPLING}=f_{DTS}/32$, $N=8$</p> <p>复位值: 0x0</p>
7	r/w	<p>MSM: 主/从模式</p> <p>0: 不执行任何操作</p> <p>1: 当前定时器的触发输入事件 (TRGI) 的动作被推迟, 以使当前定时器与其从定时器实现完美同步 (通过 TRGO)。此设置适用于由单个外部事件对多个定时器进行同步的情况。</p> <p>复位值: 0x0</p>
6:4	r/w	<p>TS[2:0]: 触发选择</p> <p>此位域可选择将要用于同步计数器的触发输入。</p> <p>000~011: 内部触发 (ITR)</p> <p>100: TI1 边沿检测器 (TIOF_ED)</p> <p>101: 滤波后的定时器输入 1 (TIOFPT)</p> <p>110: 滤波后的定时器输入 2 (TI1FPT)</p> <p>111: 外部触发输入 (ETRF)</p>

位	访问	描述
		有关各定时器 ITRx 含义的详细信息，请参见表格：TIM 内部触发连接。 这些控制位也用于选择触发输入作为 TRC，其定义如下， 000~011：内部触发信号 (ITR) 100：TI1 边沿检测器 (TIOF_ED) 其他保留 注意：这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。 复位值：0x0
3	Res	Reserved 复位值：0x0
2:0	r/w	SMS：从模式选择 选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关 (请参见输入控制寄存器和控制寄存器说明)。 000：禁止从模式--如果 CEN="1"，预分频器时钟直接由内部时钟提供。 001：保留 010：保留 011：保留 100：复位模式--在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。 101：门控模式--触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数 (但不复位)。计数器的启动和停止都是受控的。 110：触发模式--触发信号 TRGI 出现上升沿时启动计数器 (但不复位)。只控制计数器的启动。 111：外部时钟模式 0--由所选触发信号 (TRGI) 的上升沿提供计数器时钟。 注意： 如果将 TIOF_ED 选作触发输入 (TS="100")，则不得使用门控模式。实际上，TIOF 每次转换时，TIOF_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。 复位值：0x0

15.5.4 TIM DMA&中断使能寄存器 (TIM_DIER)

偏移地址：0x000C

表 224 TIM 中断使能寄存器

位	访问	描述
31:26	Res	Reserved 复位值：0x0
25	r/w	CC5IE：比较 5 中断使能 0：关闭 1：开启 复位值：0x0

位	访问	描述
24	r/w	CC4IE: 比较 4 中断使能 0: 关闭 1: 开启 复位值: 0x0
23:20	Res	Reserved 复位值: 0x0
19	r/w	AC3IE: 非对称比较 3 中断使能 0: 关闭 1: 开启 复位值: 0x0
18	r/w	AC2IE: 非对称比较 2 中断使能 0: 关闭 1: 开启 复位值: 0x0
17	r/w	AC1IE: 非对称比较 1 中断使能 0: 关闭 1: 开启 复位值: 0x0
16	r/w	AC0IE: 非对称比较 0 中断使能 0: 关闭 1: 开启 复位值: 0x0
15	Res	Reserved 复位值: 0x0
14	r/w	TDE: 触发 DMA 请求使能 0: 关闭 1: 开启 复位值: 0x0
13	r/w	COMDE: COM DMA 请求使能 0: 关闭 1: 开启 复位值: 0x0
12	r/w	CC3DE: 捕获/比较 3 DMA 请求使能 0: 关闭 1: 开启 复位值: 0x0
11	r/w	CC2DE: 捕获/比较 2 DMA 请求使能 0: 关闭 1: 开启 复位值: 0x0
10	r/w	CC1DE: 捕获/比较 1 DMA 请求使能

位	访问	描述
		0: 关闭 1: 开启 复位值: 0x0
9	r/w	CC0DE: 捕获/比较 0 DMA 请求使能 0: 关闭 1: 开启 复位值: 0x0
8	r/w	UDE: 更新 DMA 请求使能 0: 关闭 1: 开启 复位值: 0x0
7	r/w	BIE: 断路中断使能 0: 关闭 1: 开启 复位值: 0x0
6	r/w	TIE: 触发信号 (TGRI) 中断使能 0: 关闭 1: 开启 复位值: 0x0
5	r/w	COMIE: COM 中断使能 0: 关闭 1: 开启 复位值: 0x0
4	r/w	CC3IE: 捕获/比较 3 中断使能 0: 关闭 1: 开启 复位值: 0x0
3	r/w	CC2IE: 捕获/比较 2 中断使能 0: 关闭 1: 开启 复位值: 0x0
2	r/w	CC1IE: 捕获/比较 1 中断使能 0: 关闭 1: 开启 复位值: 0x0
1	r/w	CC0IE: 捕获/比较 0 中断使能 0: 关闭 1: 开启 复位值: 0x0
0	r/w	UIE: 更新中断使能 0: 关闭

位	访问	描述
		1: 开启 复位值: 0x0

15.5.5 TIM 状态寄存器 (TIM_SR)

偏移地址: 0x0010

表 225 TIM 状态寄存器

位	访问	描述
31:26	Res	Reserved 复位值: 0x0
27	r/w	EVNT1F: 事件 1 发生标志 0: 未发生 EVNT1 1: 已发生 EVNT1 此标志由硬件置 1, 可通过软件对其写 1 清零。 复位值: 0x0
26	r/w	EVNT0F: 事件 0 发生标志 0: 未发生 EVNT0 1: 已发生 EVNT0 此标志由硬件置 1, 可通过软件对其写 1 清零。 复位值: 0x0
25	r/w	CC5IF: 比较 5 中断标志 请参照 CCOIF 说明 复位值: 0x0
24	r/w	CC4IF: 比较 4 中断标志 请参照 CCOIF 说明 复位值: 0x0
23:20	Res	Reserved 复位值: 0x0
19	r/w	AC3IF: 非对称比较 3 中断标志 请参照 AC0IF 说明 复位值: 0x0
18	r/w	AC2IF: 非对称比较 2 中断标志 请参照 AC0IF 说明 复位值: 0x0
17	r/w	AC1IF: 非对称比较 1 中断标志 请参照 AC0IF 说明 复位值: 0x0
16	r/w	AC0IF: 非对称比较 0 中断标志 通道 CC0 配置为输出 当计数器与比较值匹配时, 此标志由硬件置 1, 软件写 1 清零。 0: 不匹配。

位	访问	描述
		1: TIM_CNT 计数器的值与 ACR0 寄存器的值匹配。当 ACR0 的值大于 TIM_ARR 的值时，AC0IF 位将在计数器发生上溢时变为高电平。 复位值: 0x0
15:13	Res	Reserved 复位值: 0x0
12	r/w	CC3OF: 捕获/比较 3 重复捕获标志 请参照 CC1OF 说明 复位值: 0x0
11	r/w	CC2OF: 捕获/比较 2 重复捕获标志 请参照 CC1OF 说明 复位值: 0x0
10	r/w	CC1OF: 捕获/比较 1 重复捕获标志 请参照 CC1OF 说明 复位值: 0x0
9	r/w	CC0OF: 捕获/比较 0 重复捕获标志 0: 未检测到重复捕获。 1: CCR0 寄存器中已捕获到计数器值且 CC0IF 标志已置 1。 仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。该位写 1 清零。 复位值: 0x0
8	Res	Reserved 复位值: 0x0
7	r/w	BIF: 断路中断标志 0: 未发生断路事件。 1: 在断路输入上检测到有效电平。 只要断路输入变为有效状态，此标志便由硬件置 1。断路输入无效后可通过软件对其写 1 清零。 复位值: 0x0
6	r/w	TIF: 触发中断标志 在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件写 1 清零。 0: 未发生触发事件。 1: 触发中断挂起。 复位值: 0x0
5	r/w	COMIF: COM 中断标志 此标志在发生 COM 事件时 (捕获/比较控制位 CCxE、CCxNE 和 OCxM 已更新时)由硬件置 1。但需要通过软件清零。 0: 未发生 COM 事件。 1: COM 中断挂起。 复位值: 0x0

位	访问	描述
4	r/w	CC3IF: 捕获/比较 3 中断标志 参照 CC1IF 描述 复位值: 0x0
3	r/w	CC2IF: 捕获/比较 2 中断标志 参照 CC0IF 描述 复位值: 0x0
2	r/w	CC1IF: 捕获/比较 1 中断标志 参照 CC0IF 描述 复位值: 0x0
1	r/w	CC0IF: 捕获/比较 0 中断标志 通道 CC0 配置为输出 当计数器与比较值匹配时, 此标志由硬件置 1, 软件写 1 清零。 0: 不匹配。 1: TIM_CNT 计数器的值与 CCR0 寄存器的值匹配。当 CCR0 的值大于 ARR 的值时, CC0IF 位将在计数器发生上溢时变为高电平。 通道 CC0 配置为输入 此位将在发生捕获事件时由硬件置 1。通过软件写 1 清零。 0: 未发生输入捕获事件 1: CCR0 寄存器中已捕获到计数器值 (IC0 上已检测到与所选极性匹配的边沿) 复位值: 0x0
0	r/w	UIF: 更新中断标志 该位在发生更新事件时通过硬件置 1。但需要通过软件写 1 清零。 0: 未发生更新。 1: 更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1 - TIM_CR 寄存器中的 URS=0 且 UDIS=0, 并且由软件使用 TIM_EGR 寄存器中的 UG 位重新初始化 CNT 时。 - TIM_CR 寄存器中的 URS=0 且 UDIS=0, 并且 CNT 由触发事件重新初始化时 (请参见模式控制寄存器(TIM_SMCR))。 复位值: 0x0

15.5.6 TIM 事件生成寄存器 (TIM_EGR)

偏移地址: 0x0014

表 226 TIM 事件生成寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0

位	访问	描述
7	r/w	<p>BG: 断路生成</p> <p>此位由软件置 1 以生成事件，并由硬件自动清零。</p> <p>0: 不执行任何操作</p> <p>1: 生成断路事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。</p> <p>复位值: 0x0</p>
6	r/w	<p>TG: 生成触发信号</p> <p>此位由软件置 1 以生成事件，并由硬件自动清零。</p> <p>0: 不执行任何操作</p> <p>1: TIM_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。</p> <p>复位值: 0x0</p>
5	r/w	<p>COMG: 捕获/比较控制更新生成</p> <p>该位可通过软件置 1，并由硬件自动清零</p> <p>0: 不执行任何操作 1: CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位</p> <p>复位值: 0x0</p>
4	r/w	<p>CC3G: 捕获/比较 3 生成</p> <p>参照 CC0G 描述</p> <p>复位值: 0x0</p>
3	r/w	<p>CC2G: 捕获/比较 2 生成</p> <p>参照 CC0G 描述</p> <p>复位值: 0x0</p>
2	r/w	<p>CC1G: 捕获/比较 1 生成</p> <p>参照 CC0G 描述</p> <p>复位值: 0x0</p>
1	r/w	<p>CC0G: 捕获/比较 0 生成</p> <p>此位由软件置 1 以生成事件，并由硬件自动清零。</p> <p>0: 不执行任何操作</p> <p>1: 通道 1 上生成捕获/比较事件:</p> <ul style="list-style-type: none"> 通道 CC0 配置为输出: 使能时, CC0IF 标志置 1 并发送相应的中断或 DMA 请求。 通道 CC0 配置为输入: TIM_CCR0 寄存器中将捕获到计数器当前值。使能时, CC0IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC0IF 标志已为高电平, CC0OF 标志将置 1。 <p>复位值: 0x0</p>
0	r/w	<p>UG: 更新生成</p> <p>该位可通过软件置 1，并由硬件自动清零。</p> <p>0: 不执行任何操作</p> <p>1: 重新初始化计数器并生成一个寄存器更新事件。</p> <p>注: 预分频器计数器也将清零 (但预分频比不受影响), 而且计数器将清零</p> <p>复位值: 0x0</p>

15.5.7 TIM 捕获/比较模式寄存器 0 (TIM_CCMR0)

偏移地址: 0x0018

表 227 TIM 捕获/比较模式寄存器 0 (输出比较模式)

位	访问	描述
31:25	Res	Reserved 复位值: 0x0
24	r/w	OC1CE: 输出比较 1 清零使能 参照 OC1CE 描述 复位值: 0x0
23:20	r/w	OC1M: 输出比较 1 模式 参照 OC0M 描述 复位值: 0x0
19	r/w	OC1PE: 输出比较 1 预装载使能 参照 OC0PE 相关描述 复位值: 0x0
18	r/w	OC1FE: 输出比较 1 快速使能 参照 OC0FE 描述 复位值: 0x0
17:16	r/w	CC1S: 捕获/比较 1 通道选择 00: CC1 通道配置为输出 01: CC1 通道配置为输入, IC1 映射到 TI1 上 10: CC1 通道配置为输入, IC1 映射到 TI0 上 11: CC1 通道配置为输入, IC1 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效 注: 仅当通道关闭时 (TIM_CCER 中的 CC1E=0), 才可向 CC1S 位写入数据。 复位值: 0x0
15:9	Res	Reserved 复位值: 0x0
8	r/w	OC0CE: 输出比较 0 清零使能 0: OC0REF 不受 OCREF_CLR_INT 输入影响 1: OCREF_CLR_INT 输入上检测到高电平时, OC0REF 立即清零 复位值: 0x0
7:4	r/w	OC0M: 输出比较 0 模式 这些位定义提供 OC0 和 OC0N 的输出参考信号 OC0REF 的行为。OC0REF 为高电平有效, 而 OC0 和 OC0N 的有效电平则取决于 CC0P 位和 CC0NP 位。 0000: 冻结。输出比较寄存器 TIM_CCR0 与计数器 TIM_CNT 进行比较不会对输出造成任何影响。 0001: 将通道 0 设置为匹配时输出有效电平。当计数器 TIM_CNT 与捕获/比较寄存器 0(TIM_CCR0)匹配时, OC0REF 信号强制变为高电平。

位	访问	描述
		<p>0010: 将通道 0 设置为匹配时输出无效电平。当计数器 TIM_CNT 与捕获/比较寄存器 0(TIM_CCR0)匹配时, OC0REF 信号强制变为低电平。</p> <p>0011: 翻转。TIM_CNT=TIM_CCR0 时, OC0REF 发生翻转。</p> <p>0100: 强制变为无效电平。OC0REF 强制变为低电平。</p> <p>0101: 强制变为有效电平。OC0REF 强制变为高电平。</p> <p>0110: PWM 模式 0--在递增计数模式下, 只要 TIM_CNT<TIM_CCR0, 通道 0 便为有效状态, 否则为无效状态。在递减计数模式下, 只要 TIM_CNT>TIM_CCR0, 通道 0 便为无效状态 (OC0REF="0"), 否则为有效状态 (OC0REF="1")。</p> <p>0111: PWM 模式 1--在递增计数模式下, 只要 TIM_CNT<TIM_CCR0, 通道 0 便为无效状态, 否则为有效状态。在递减计数模式下, 只要 TIM_CNT>TIM_CCR0, 通道 0 便为有效状态, 否则为无效状态。</p> <p>1100: 组合 PWM 模式 0--在递增计数模式下, 只要 TIM_CNT<TIM_CCR0 或者 TIM_CNT≥TIM_ACRO, 通道 0 便为有效状态, 否则为无效状态, 此时要求 CCR0<ACRO。在递减计数模式下, 只要 TIM_CNT>TIM_CCR0 或者 TIM_CNT≤TIM_ACRO, 通道 0 便为无效状态 (OC0REF="0"), 否则为有效状态 (OC0REF="1"), 此时要求 CCR0>ACRO。此模式仅支持计数器边沿对齐模式 (即 CMS=0)</p> <p>1101: 组合 PWM 模式 1--在递增计数模式下, 只要 TIM_CNT<TIM_CCR0 或者 TIM_CNT≥TIM_ACRO, 通道 0 便为无效状态, 否则为有效状态, 此时要求 CCR0<ACRO。在递减计数模式下, 只要 TIM_CNT>TIM_CCR0 或者 TIM_CNT≤TIM_ACRO, 通道 0 便为有效状态, 否则为无效状态, 此时要求 CCR0>ACRO。此模式仅支持计数器边沿对齐模式 (即 CMS=0)</p> <p>1110: 非对称 PWM 模式 0--在递增计数模式下, 只要 TIM_CNT<TIM_CCR0, 通道 0 便为有效状态, 否则为无效状态。在递减计数模式下, 只要 TIM_CNT>TIM_ACRO, 通道 0 便为无效状态 (OC0REF="0"), 否则为有效状态 (OC0REF="1")。此模式仅支持计数器中心对称模式 (即 CMS!=0)</p> <p>1111: 非对称 PWM 模式 1--在递增计数模式下, 只要 TIM_CNT<TIM_CCR0, 通道 0 便为无效状态, 否则为有效状态。在递减计数模式下, 只要 TIM_CNT>TIM_ACRO, 通道 0 便为有效状态, 否则为无效状态。此模式仅支持计数器中心对称模式 (即 CMS!=0)</p> <p>其他: 保留</p> <p>注: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 3 且 CC1S="00" (通道配置为输出), 这些位即无法修改。在 PWM 模式下, 仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时, OC0REF 电平才会发生更改。此位域将在具有互补输出的通道上进行预装载。如果 TIM_CR0 寄存器中的 CCPC 位置 1, 则仅当生成 COM 事件时, OC0M 有效位才会从预装载位获取新值。</p> <p>复位值: 0x0</p>
3	r/w	<p>OC0PE: 输出比较 0 预装载使能</p> <p>0: 禁止与 TIM_CCR0 相关的预装载寄存器。可随时向 TIM_CCR0 写入数据, 写入后将立即使用新值。</p>

位	访问	描述
		<p>1: 使能与 TIM_CCR0 相关的预装载寄存器。可读/写访问预装载寄存器。TIM_CCR0 预装载值在每次生成更新事件时都会装载到活动寄存器中。</p> <p>注:</p> <ol style="list-style-type: none"> 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 3 且 CC0S="00" (通道配置为输出), 这些位即无法修改。 只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式 (TIM_CR0 寄存器中的 OPM 位置 1)。其它情况下则无法保证该行为。 <p>复位值: 0x0</p>
2	r/w	<p>OC0FE: 输出比较 0 快速使能</p> <p>此位用于加快触发输入事件对 CC0 输出的影响。</p> <p>0: 即使触发开启, CC0 也将根据计数器和 CCR0 值正常工作。触发输入出现边沿时, 激活 CC0 输出的最短延迟时间为 5 个时钟周期。</p> <p>1: 触发输入上出现有效边沿相当于 CC0 输出上的比较匹配。随后, 无论比较结果如何, OC 都设置为比较电平。采样触发输入和激活 CC0 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM0 或 PWM1 模式时, OC0FE 才会起作用。</p> <p>复位值: 0x0</p>
1:0	r/w	<p>CC0S: 捕获/比较 0 通道选择</p> <p>00: CC0 通道配置为输出</p> <p>01: CC0 通道配置为输入, IC0 映射到 TI0 上</p> <p>10: CC0 通道配置为输入, IC0 映射到 TI1 上</p> <p>11: CC0 通道配置为输入, IC0 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效</p> <p>注: 仅当通道关闭时 (TIM_CCER 中的 CC0E=0), 才可向 CC0S 位写入数据。</p> <p>复位值: 0x0</p>

表 228 TIM 捕获/比较模式寄存器 0 (输入捕获模式)

位	访问	描述
31:24	Res	<p>Reserved</p> <p>复位值: 0x0</p>
23:20	r/w	<p>IC1F: 输入捕获 1 滤波器</p> <p>参照 IC0F 相关描述</p> <p>复位值: 0x0</p>
19:18	r/w	<p>IC1PSC: 输入捕获 1 预分频器</p> <p>参照 IC0PSC 相关描述</p> <p>复位值: 0x0</p>
17:16	r/w	<p>CC1S: 捕获/比较 1 通道选择</p> <p>00: CC1 通道配置为输出</p> <p>01: CC1 通道配置为输入, IC1 映射到 TI1 上</p> <p>10: CC1 通道配置为输入, IC1 映射到 TI0 上</p> <p>11: CC1 通道配置为输入, IC1 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效</p>

位	访问	描述
		注意：仅当通道关闭时 (TIM_CCER 中的 CC1E=0)，才可向 CC1S 位写入数据。 复位值：0x0
15:8	Res	Reserved 复位值：0x0
7:4	r/w	<p>IC0F：输入捕获 0 滤波器</p> <p>此位域可定义 TI0 输入的采样频率和适用于 TI0 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：</p> <p>0000：无滤波器，按 f_{DTS} 频率进行采样</p> <p>0001：$f_{SAMPLING}=f_{CK_INT}$，N=2</p> <p>0010：$f_{SAMPLING}=f_{CK_INT}$，N=4</p> <p>0011：$f_{SAMPLING}=f_{CK_INT}$，N=8</p> <p>0100：$f_{SAMPLING}=f_{DTS}/2$，N=6</p> <p>0101：$f_{SAMPLING}=f_{DTS}/2$，N=8</p> <p>0110：$f_{SAMPLING}=f_{DTS}/4$，N=6</p> <p>0111：$f_{SAMPLING}=f_{DTS}/4$，N=8</p> <p>1000：$f_{SAMPLING}=f_{DTS}/8$，N=6</p> <p>1001：$f_{SAMPLING}=f_{DTS}/8$，N=8</p> <p>1010：$f_{SAMPLING}=f_{DTS}/16$，N=5</p> <p>1011：$f_{SAMPLING}=f_{DTS}/16$，N=6</p> <p>1100：$f_{SAMPLING}=f_{DTS}/16$，N=8</p> <p>1101：$f_{SAMPLING}=f_{DTS}/32$，N=5</p> <p>1110：$f_{SAMPLING}=f_{DTS}/32$，N=6</p> <p>1111：$f_{SAMPLING}=f_{DTS}/32$，N=8</p> <p>复位值：0x0</p>
3:2	r/w	<p>IC0PSC：输入捕获 0 预分频器</p> <p>此位域定义 CC1 输入(IC0)的预分频比。只要 CC0E=0 (TIM_CCER 寄存器)，预分频器便立即复位。</p> <p>00：无预分频器，捕获输入上每检测到一个边沿便执行捕获</p> <p>01：每发生 2 个事件便执行一次捕获</p> <p>10：每发生 4 个事件便执行一次捕获</p> <p>11：每发生 8 个事件便执行一次捕获</p> <p>复位值：0x0</p>
1:0	r/w	<p>CC0S：捕获/比较 0 通道选择</p> <p>00：CC0 通道配置为输出</p> <p>01：CC0 通道配置为输入，IC0 映射到 TI0 上</p> <p>10：CC0 通道配置为输入，IC0 映射到 TI1 上</p> <p>11：CC0 通道配置为输入，IC0 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_MCR 寄存器)选择内部触发输入时有效</p> <p>注意：仅当通道关闭时 (TIM_CCER 中的 CC0E=0)，才可向 CC0S 位写入数据。</p> <p>复位值：0x0</p>

15.5.8 TIM 捕获/比较模式寄存器 1 (TIM_CCMR1)

偏移地址: 0x001C

表 229 TIM 捕获/比较模式寄存器 1 (输出比较模式)

位	访问	描述
31:25	Res	Reserved 复位值: 0x0
24	r/w	OC3CE: 输出比较 3 清零使能 参照 OC2CE 描述 复位值: 0x0
23:20	r/w	OC3M: 输出比较 3 模式 参照 OC2M 描述 复位值: 0x0
19	r/w	OC3PE: 输出比较 3 预装载使能 参照 OC2PE 相关描述 复位值: 0x0
18	r/w	OC3FE: 输出比较 3 快速使能 参照 OC2FE 描述 复位值: 0x0
17:16	r/w	CC3S: 捕获/比较 3 通道选择 00: CC3 通道配置为输出 01: CC3 通道配置为输入, IC3 映射到 TI3 上 10: CC3 通道配置为输入, IC3 映射到 TI2 上 11: CC3 通道配置为输入, IC3 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效 注: 仅当通道关闭时 (TIM_CCER 中的 CC3E=0), 才可向 CC3S 位写入数据。 复位值: 0x0
15:9	Res	Reserved 复位值: 0x0
8	r/w	OC2CE: 输出比较 2 清零使能 0: OC2REF 不受 OCREF_CLR_INT 输入影响 1: OCREF_CLR_INT 输入上检测到高电平时, OC2REF 立即清零 复位值: 0x0
7:4	r/w	OC2M: 输出比较 2 模式 这些位定义提供 OC2 和 OC2N 的输出参考信号 OC2REF 的行为。OC2REF 为高电平有效, 而 OC2 和 OC2N 的有效电平则取决于 CC2P 位和 CC2NP 位。 0000: 冻结。输出比较寄存器 TIM_CCR2 与计数器 TIM_CNT 进行比较不会对输出造成任何影响。 0001: 将通道 2 设置为匹配时输出有效电平。当计数器 TIM_CNT 与捕获/比较寄存器 1(TIM_CCR2) 匹配时, OC2REF 信号强制变为高电平。

位	访问	描述
		<p>0010: 将通道 2 设置为匹配时输出无效电平。当计数器 TIM_CNT 与捕获/比较寄存器 2(TIM_CCR2) 匹配时, OC2REF 信号强制变为低电平。</p> <p>0011: 翻转。TIM_CNT=TIM_CCR2 时, OC2REF 发生翻转。</p> <p>0100: 强制变为无效电平。OC2REF 强制变为低电平。</p> <p>0101: 强制变为有效电平。OC2REF 强制变为高电平。</p> <p>0110: PWM 模式 0--在递增计数模式下, 只要 TIM_CNT<TIM_CCR2, 通道 2 便为有效状态, 否则为无效状态。在递减计数模式下, 只要 TIM_CNT>TIM_CCR2, 通道 2 便为无效状态 (OC2REF="0"), 否则为有效状态 (OC2REF="1")。</p> <p>0111: PWM 模式 1--在递增计数模式下, 只要 TIM_CNT<TIM_CCR2, 通道 2 便为无效状态, 否则为有效状态。在递减计数模式下, 只要 TIM_CNT>TIM_CCR2, 通道 2 便为有效状态, 否则为无效状态。</p> <p>1100: 组合 PWM 模式 0--在递增计数模式下, 只要 TIM_CNT<TIM_CCR2 或者 TIM_CNT≥TIM_ACR2, 通道 2 便为有效状态, 否则为无效状态, 此时要求 CCR2<ACR2。在递减计数模式下, 只要 TIM_CNT>TIM_CCR2 或者 TIM_CNT≤TIM_ACR2, 通道 2 便为无效状态 (OC2REF="0"), 否则为有效状态 (OC2REF="1"), 此时要求 CCR2>ACR2。此模式仅支持计数器边沿对齐模式 (即 CMS=0)</p> <p>1101: 组合 PWM 模式 1--在递增计数模式下, 只要 TIM_CNT<TIM_CCR2 或者 TIM_CNT≥TIM_ACR2, 通道 2 便为无效状态, 否则为有效状态, 此时要求 CCR2<ACR2。在递减计数模式下, 只要 TIM_CNT>TIM_CCR2 或者 TIM_CNT≤TIM_ACR2, 通道 2 便为有效状态, 否则为无效状态, 此时要求 CCR2>ACR2。此模式仅支持计数器边沿对齐模式 (即 CMS=0)</p> <p>1110: 非对称 PWM 模式 0--在递增计数模式下, 只要 TIM_CNT<TIM_CCR2, 通道 2 便为有效状态, 否则为无效状态。在递减计数模式下, 只要 TIM_CNT>TIM_ACR2, 通道 2 便为无效状态 (OC2REF="0"), 否则为有效状态 (OC2REF="1")。此模式仅支持计数器中心对称模式 (即 CMS!=0)</p> <p>1111: 非对称 PWM 模式 1--在递增计数模式下, 只要 TIM_CNT<TIM_CCR2, 通道 2 便为无效状态, 否则为有效状态。在递减计数模式下, 只要 TIM_CNT>TIM_ACR2, 通道 2 便为有效状态, 否则为无效状态。此模式仅支持计数器中心对称模式 (即 CMS!=0)</p> <p>其他: 保留</p> <p>注: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位)级别 3 且 CC1S="00" (通道配置为输出), 这些位即无法修改。在 PWM 模式下, 仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时, OCREF 电平才会发生更改。此位域将在具有互补输出的通道上进行预装载。如果 TIM_CR1 寄存器中的 CCPC 位置 1, 则仅当生成 COM 事件时, OC3M 有效位才会从预装载位获取新值。</p> <p>复位值: 0x0</p>
3	r/w	<p>OC2PE: 输出比较 2 预装载使能</p> <p>0: 禁止与 TIM_CCR2 相关的预装载寄存器。可随时向 TIM_CCR2 写入数据, 写入后将立即使用新值。</p>

位	访问	描述
		<p>1: 使能与 TIM_CCR2 相关的预装载寄存器。可读/写访问预装载寄存器。TIM_CCR2 预装载值在每次生成更新事件时都会装载到活动寄存器中。</p> <p>注:</p> <p>1: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 3 且 CC2S="00" (通道配置为输出), 这些位即无法修改。</p> <p>2: 只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式 (TIM_CR0 寄存器中的 OPM 位置 1)。其它情况下则无法保证该行为。</p> <p>复位值: 0x0</p>
2	r/w	<p>OC2FE: 输出比较 2 快速使能</p> <p>此位用于加快触发输入事件对 CC 输出的影响。</p> <p>0: 即使触发开启, CC2 也将根据计数器和 CCR2 值正常工作。触发输入出现边沿时, 激活 CC2 输出的最短延迟时间为 5 个时钟周期。</p> <p>1: 触发输入上出现有效边沿相当于 CC2 输出上的比较匹配。随后, 无论比较结果如何, OC 都设置为比较电平。采样触发输入和激活 CC2 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM0 或 PWM1 模式时, OC2FE 才会起作用。</p> <p>复位值: 0x0</p>
1:0	r/w	<p>CC2S: 捕获/比较 2 通道选择</p> <p>00: CC2 通道配置为输出</p> <p>01: CC2 通道配置为输入, IC2 映射到 TI2 上</p> <p>10: CC2 通道配置为输入, IC2 映射到 TI3 上</p> <p>11: CC3 通道配置为输入, IC2 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效</p> <p>注: 仅当通道关闭时 (TIM_CCER 中的 CC2E=0), 才可向 CC2S 位写入数据。</p> <p>复位值: 0x0</p>

表 230 TIM 捕获/比较模式寄存器 1 (输入捕获模式)

位	访问	描述
31:24	Res	<p>Reserved</p> <p>复位值: 0x0</p>
23:20	r/w	<p>IC3F: 输入捕获 3 滤波器</p> <p>参照 IC2F 相关描述</p> <p>复位值: 0x0</p>
19:18	r/w	<p>IC3PSC: 输入捕获 3 预分频器</p> <p>参照 IC2PSC 相关描述</p> <p>复位值: 0x0</p>
17:16	r/w	<p>CC3S: 捕获/比较 3 通道选择</p> <p>00: CC3 通道配置为输出</p> <p>01: CC3 通道配置为输入, IC3 映射到 TI3 上</p> <p>10: CC3 通道配置为输入, IC3 映射到 TI2 上</p> <p>11: CC3 通道配置为输入, IC3 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效</p>

位	访问	描述
		注意：仅当通道关闭时 (TIM_CCER 中的 CC3E=0)，才可向 CC3S 位写入数据。 复位值：0x0
15:8	Res	Reserved 复位值：0x0
7:4	r/w	<p>IC2F：输入捕获 2 滤波器</p> <p>此位域可定义 TI2 输入的采样频率和适用于 TI2 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：</p> <p>0000：无滤波器，按 f_{DTS} 频率进行采样</p> <p>0001：$f_{SAMPLING}=f_{CK_INT}$，N=2</p> <p>0010：$f_{SAMPLING}=f_{CK_INT}$，N=4</p> <p>0011：$f_{SAMPLING}=f_{CK_INT}$，N=8</p> <p>0100：$f_{SAMPLING}=f_{DTS}/2$，N=6</p> <p>0101：$f_{SAMPLING}=f_{DTS}/2$，N=8</p> <p>0110：$f_{SAMPLING}=f_{DTS}/4$，N=6</p> <p>0111：$f_{SAMPLING}=f_{DTS}/4$，N=8</p> <p>1000：$f_{SAMPLING}=f_{DTS}/8$，N=6</p> <p>1001：$f_{SAMPLING}=f_{DTS}/8$，N=8</p> <p>1010：$f_{SAMPLING}=f_{DTS}/16$，N=5</p> <p>1011：$f_{SAMPLING}=f_{DTS}/16$，N=6</p> <p>1100：$f_{SAMPLING}=f_{DTS}/16$，N=8</p> <p>1101：$f_{SAMPLING}=f_{DTS}/32$，N=5</p> <p>1110：$f_{SAMPLING}=f_{DTS}/32$，N=6</p> <p>1111：$f_{SAMPLING}=f_{DTS}/32$，N=8</p> <p>复位值：0x0</p>
3:2	r/w	<p>IC2PSC：输入捕获 2 预分频器</p> <p>此位域定义 CC2 输入 (IC2) 的预分频比。只要 CC2E=0 (TIM_CCER 寄存器)，预分频器便立即复位。</p> <p>00：无预分频器，捕获输入上每检测到一个边沿便执行捕获</p> <p>01：每发生 2 个事件便执行一次捕获</p> <p>10：每发生 4 个事件便执行一次捕获</p> <p>11：每发生 8 个事件便执行一次捕获</p> <p>复位值：0x0</p>
1:0	r/w	<p>CC2S：捕获/比较 2 通道选择</p> <p>00：CC2 通道配置为输出</p> <p>01：CC2 通道配置为输入，IC2 映射到 TI2 上</p> <p>10：CC2 通道配置为输入，IC2 映射到 TI3 上</p> <p>11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效</p> <p>注意：仅当通道关闭时 (TIM_CCER 中的 CC2E=0)，才可向 CC2S 位写入数据。</p> <p>复位值：0x0</p>

15.5.9 TIM 捕获/比较模式寄存器 2 (TIM_CCMR2)

偏移地址: 0x0020

表 231 TIM 捕获/比较模式寄存器 2 (输出比较模式)

位	访问	描述
31:25	Res	Reserved 复位值: 0x0
24	r/w	OC5CE: 输出比较 5 清零使能 参照 OC4CE 描述 复位值: 0x0
23:20	r/w	OC5M: 输出比较 5 模式 参照 OC4M 描述 复位值: 0x0
19	r/w	OC5PE: 输出比较 5 预装载使能 参照 OC4PE 相关描述 复位值: 0x0
18	r/w	OC5FE: 输出比较 5 快速使能 参照 OC4FE 描述 复位值: 0x0
17:16	ro	CC5S: 捕获/比较 5 通道选择 注: 固定为输入 复位值: 0x0
15:9	Res	Reserved 复位值: 0x0
8	r/w	OC4CE: 输出比较 4 清零使能 0: OC4REF 不受 OCREF_CLR_INT 输入影响 1: OCREF_CLR_INT 输入上检测到高电平时, OC4REF 立即清零 复位值: 0x0
7:4	r/w	OC4M: 输出比较 4 模式 这些位定义提供参考信号 OC4REF 的行为。OC4REF 为高电平有效位。 0000: 冻结。输出比较寄存器 TIM_CCR4 与计数器 TIM_CNT 进行比较不会对输出造成任何影响。 0001: 将通道 4 设置为匹配时输出有效电平。当计数器 TIM_CNT 与捕获/比较寄存器 4 (TIM_CCR4) 匹配时, OC4REF 信号强制变为高电平。 0010: 将通道 4 设置为匹配时输出无效电平。当计数器 TIM_CNT 与捕获/比较寄存器 4 (TIM_CCR4) 匹配时, OC4REF 信号强制变为低电平。 0011: 翻转。TIM_CNT=TIM_CCR4 时, OC4REF 发生翻转。 0100: 强制变为无效电平。OC4REF 强制变为低电平。 0101: 强制变为有效电平。OC4REF 强制变为高电平。

位	访问	描述
		<p>0110: PWM 模式 0--在递增计数模式下, 只要 $TIM_CNT < TIM_CCR4$, 通道 4 便为有效状态, 否则为无效状态。在递减计数模式下, 只要 $TIM_CNT > TIM_CCR4$, 通道 4 便为无效状态 ($OC4REF = "0"$), 否则为有效状态 ($OC4REF = "1"$)。</p> <p>0111: PWM 模式 1--在递增计数模式下, 只要 $TIM_CNT < TIM_CCR4$, 通道 4 便为无效状态, 否则为有效状态。在递减计数模式下, 只要 $TIM_CNT > TIM_CCR4$, 通道 4 便为有效状态, 否则为无效状态。</p> <p>其他: 保留</p> <p>注:</p> <p>只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 3 且 CC1S="00" (通道配置为输出), 这些位即无法修改。</p> <p>在 PWM 模式下, 仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时, OCREF 电平才会发生更改。</p> <p>此位域将在具有互补输出的通道上进行预装载。如果 TIM_CR1 寄存器中的 CCPC 位置 1, 则仅当生成 COM 事件时, OC3M 有效位才会从预装载位获取新值。</p> <p>复位值: 0x0</p>
3	r/w	<p>OC4PE: 输出比较 4 预装载使能</p> <p>0: 禁止与 TIM_CCR4 相关的预装载寄存器。可随时向 TIM_CCR4 写入数据, 写入后将立即使用新值。</p> <p>1: 使能与 TIM_CCR4 相关的预装载寄存器。可读/写访问预装载寄存器。TIM_CCR4 预装载值在每次生成更新事件时都会装载到活动寄存器中。</p> <p>注:</p> <p>1: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 3 且 CC2S="00" (通道配置为输出), 这些位即无法修改。</p> <p>2: 只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式 (TIM_CR0 寄存器中的 OPM 位置 1)。其它情况下则无法保证该行为。</p> <p>复位值: 0x0</p>
2	r/w	<p>OC4FE: 输出比较 4 快速使能</p> <p>此位用于加快触发输入事件对 CC 输出的影响。</p> <p>0: 即使触发开启, CC4 也将根据计数器和 CCR4 值正常工作。触发输入出现边沿时, 激活 CC4 输出的最短延迟时间为 5 个时钟周期。</p> <p>1: 触发输入上出现有效边沿相当于 CC4 输出上的比较匹配。随后, 无论比较结果如何, OC 都设置为比较电平。采样触发输入和激活 CC4 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM0 或 PWM1 模式时, OC4FE 才会起作用。</p> <p>复位值: 0x0</p>
1:0	ro	<p>CC4S: 捕获/比较 4 通道选择</p> <p>注: 固定为输入</p> <p>复位值: 0x0</p>

表 232 TIM 捕获/比较模式寄存器 2 (输入捕获模式)

位	访问	描述
31:24	Res	Reserved 复位值: 0x0
23:20	r/w	IC3F:输入捕获 5 滤波器 参照 IC4F 相关描述 复位值: 0x0
19:18	r/w	IC5PSC: 输入捕获 5 预分频器 参照 IC4PSC 相关描述 复位值: 0x0
17:16	r/w	CC5S: 捕获/比较 5 通道选择 00: CC5 通道配置为输出 01: CC5 通道配置为输入, IC3 映射到 TI3 上 10: CC5 通道配置为输入, IC3 映射到 TI2 上 11: CC5 通道配置为输入, IC3 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效 注意: 仅当通道关闭时 (TIM_CCER 中的 CC5E=0), 才可向 CC5S 位写入数据。 复位值: 0x0
15:8	Res	Reserved 复位值: 0x0
7:4	r/w	IC4F: 输入捕获 4 滤波器 此位域可定义 TI4 输入的采样频率和适用于 TI4 的数字滤波器带宽。数字滤波器由事件计数器组成, 每 N 个事件才视为一个有效边沿: 0000: 无滤波器, 按 f_{DTS} 频率进行采样 0001: $f_{SAMPLING}=f_{CK_INT}$, $N=2$ 0010: $f_{SAMPLING}=f_{CK_INT}$, $N=4$ 0011: $f_{SAMPLING}=f_{CK_INT}$, $N=8$ 0100: $f_{SAMPLING}=f_{DTS}/2$, $N=6$ 0101: $f_{SAMPLING}=f_{DTS}/2$, $N=8$ 0110: $f_{SAMPLING}=f_{DTS}/4$, $N=6$ 0111: $f_{SAMPLING}=f_{DTS}/4$, $N=8$ 1000: $f_{SAMPLING}=f_{DTS}/8$, $N=6$ 1001: $f_{SAMPLING}=f_{DTS}/8$, $N=8$ 1010: $f_{SAMPLING}=f_{DTS}/16$, $N=5$ 1011: $f_{SAMPLING}=f_{DTS}/16$, $N=6$ 1100: $f_{SAMPLING}=f_{DTS}/16$, $N=8$ 1101: $f_{SAMPLING}=f_{DTS}/32$, $N=5$ 1110: $f_{SAMPLING}=f_{DTS}/32$, $N=6$ 1111: $f_{SAMPLING}=f_{DTS}/32$, $N=8$ 复位值: 0x0

位	访问	描述
3:2	r/w	<p>IC4PSC: 输入捕获 4 预分频器</p> <p>此位域定义 CC4 输入 (IC4) 的预分频比。只要 CC4E=0 (TIM_CCER 寄存器), 预分频器便立即复位。</p> <p>00: 无预分频器, 捕获输入上每检测到一个边沿便执行捕获</p> <p>01: 每发生 2 个事件便执行一次捕获</p> <p>10: 每发生 4 个事件便执行一次捕获</p> <p>11: 每发生 8 个事件便执行一次捕获</p> <p>复位值: 0x0</p>
1:0	r/w	<p>CC4S: 捕获/比较 4 通道选择</p> <p>00: CC4 通道配置为输出</p> <p>01: CC4 通道配置为输入, IC4 映射到 TI2 上</p> <p>10: CC4 通道配置为输入, IC4 映射到 TI3 上</p> <p>11: CC4 通道配置为输入, IC4 映射到 TRC 上。此模式仅在通过 TS 位 (TIM_SMCR 寄存器) 选择内部触发输入时有效</p> <p>注意: 仅当通道关闭时 (TIM_CCER 中的 CC4E=0), 才可向 CC4S 位写入数据。</p> <p>复位值: 0x0</p>

15.5.10 TIM 捕获/比较模式使能寄存器 (TIM_CCER)

偏移地址: 0x0024

表 233 TIM 捕获/比较模式使能寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15	r/w	<p>CC3NP: 捕获/比较 3 互补输出极性</p> <p>参照 CC0NP 描述, 该位只在输入模式下有意义</p> <p>复位值: 0x0</p>
14	Res	Reserved 复位值: 0x0
13	r/w	<p>CC3P: 捕获/比较 3 输出极性</p> <p>参照 CC0P 描述</p> <p>复位值: 0x0</p>
12	r/w	<p>CC3E: 捕获/比较 3 输出使能</p> <p>参照 CC0E 描述</p> <p>复位值: 0x0</p>
11	r/w	<p>CC2NP: 捕获/比较 2 互补输出极性</p> <p>参照 CC0NP 描述</p> <p>复位值: 0x0</p>
10	r/w	<p>CC2NE: 捕获/比较 2 互补输出使能</p> <p>参照 CC0NE 描述</p> <p>复位值: 0x0</p>

位	访问	描述
9	r/w	CC2P: 捕获/比较 2 输出极性 参照 CC1P 描述 复位值: 0x0
8	r/w	CC2E: 捕获/比较 2 输出使能 参照 CC0E 描述 复位值: 0x0
7	r/w	CC1NP: 捕获/比较 1 互补输出极性 参照 CC0NP 描述 复位值: 0x0
6	r/w	CC1NE: 捕获/比较 1 互补输出使能 参照 CC0NE 描述 复位值: 0x0
5	r/w	CC1P: 捕获/比较 1 输出极性 参照 CC0P 描述 复位值: 0x0
4	r/w	CC1E: 捕获/比较 1 输出使能 参照 CC0E 描述 复位值: 0x0
3	r/w	CC0NP: 捕获/比较 0 互补输出极性 CC0 通道配置为输出: 0: OC0N 高电平有效。 1: OC0N 低电平有效。 CC0 通道配置为输入: 此位与 CC0P 配合使用, 用以定义 TI0FP0 和 TI1FP0 的极性。请参见 CC1P 说明。 注意: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 2 或 3 且 CC0S="00" (通道配置为输出), 此位立即变为不可写状态。 复位值: 0x0
2	r/w	CC0NE: 捕获/比较 0 互补输出使能 0: 关闭—OC0N 未激活。OC0N 电平是 MOE、OIS0N、HIZ0N 和 CC0NE 位的函数。 1: 开启—在相应输出引脚上输出 OC0N 信号, 具体取决于 MOE、OIS0、HIZ0N 和 CC0NE 位。 复位值: 0x0
1	r/w	CC0P: 捕获/比较 0 输出极性 CC0 通道配置为输出: 0: OC0 高电平有效 1: OC0 低电平有效 CC0 通道配置为输入: CC0NP/CC0P 位可针对触发或捕获操作选择 TI0FP0 和 TI1FP0 的有效极性。

位	访问	描述
		<p>00: 非反相/上升沿触发电路对 T1xFP0 上升沿敏感 (在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), T1xFP0 未反相 (在门控模式或编码器模式下执行触发操作)。</p> <p>01: 反相/下降沿触发电路对 T1xFP0 下降沿敏感 (在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), T1xFP0 反相 (在门控模式或编码器模式下执行触发操作)。</p> <p>10: 保留, 不使用此配置。</p> <p>11: 未反相/边沿触发。电路对 T1xFP0 上升沿和下降沿都敏感 (在复位模式、外部时钟模式或触发模式下执行捕获或触发操作), T1xFP0 未反相 (在门控模式下执行触发操作)。</p> <p>注意: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 2 或 3, 此位立即变为不可写状态。</p> <p>复位值: 0x0</p>
0	r/w	<p>CC0E: 捕获/比较 0 输出使能</p> <p>CC0 通道配置为输出:</p> <p>0: 关闭——OC0 未激活。OC0 电平是 MOE、OIS0、HIZ0 和 CC0E 位的函数。</p> <p>1: 开启——OC0 信号是相应输出引脚上的输出, 具体取决于 MOE、OIS0、HIZ0 和 CC0E 位。</p> <p>CC0 通道配置为输入:</p> <p>此位决定是否可以将计数器值捕获到输入捕获/比较寄存器 0 (CCR0) 中。</p> <p>0: 禁止捕获。</p> <p>1: 使能捕获。</p> <p>复位值: 0x0</p>

15.5.11 TIM 计数器 (TIM_CNT)

偏移地址: 0x0028

表 234 TIM 计数器

位	访问	描述
31:16	Res	<p>Reserved</p> <p>复位值: 0x0</p>
15:0	r/w	<p>CNT[15:0]: 计数器值</p> <p>注: 计数器值支持计数过程中更改修正, 但这是一个异步操作。计数器亦可在使能信号起来之前进行软件初值加载。改写该值前需要解锁 TIM_SET_UNLOCK。</p> <p>复位值: 0x0</p>

15.5.12 TIM 预分频器 (TIM_PSC)

偏移地址: 0x002C

表 235 TIM 预分频器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	PSC[15:0]: 预分频器值 计数器时钟频率 (CK_CNT) 等于 $f_{IP} / (PSC[15:0] + 1)$ 。PSC 包含每次发生更新事件 (包括计数器通过 TIM_EGR 寄存器中的 UG 位清零时, 或在配置为“复位模式”时通过触发控制器清零时) 时要装载到活动预分频器寄存器的值。 复位值: 0x0

15.5.13 TIM 自动重载寄存器 (TIM_ARR)

偏移地址: 0x0030

表 236 TIM 自动重载寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	ARR[15:0]: 自动重载值 ARR 为要装载到实际自动重载寄存器的值。当自动重载值为空时, 计数器不工作。 复位值: 0xFFFF

15.5.14 TIM 重复计数器寄存器 (TIM_RCR)

偏移地址: 0x0034

表 237 TIM 重复计数器寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
7:0	r/w	REP[7:0]: 重复计数器值 使能预装载寄存器时, 用户可通过这些位设置比较寄存器的更新频率 (即, 从预装载寄存器向活动寄存器周期性传输数据); 使能更新中断时, 也可设置更新中断的生成速率。 与 REP_CNT 相关的减计数器每次计数到 0 时, 都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件时, REP_CNT 才会重载 REP 值, 因此在生成下一重复更新事件之前, 无论向 TIM_RCR 寄存器写入何值都无影响。这意味着 PWM 模式下 (REP+1) 相当于: <ul style="list-style-type: none"> 边沿对齐模式下的 PWM 周期数 中心对齐模式下的 PWM 半周期数 复位值: 0x0

15.5.15 TIM 捕获/比较寄存器 0 (TIM_CCR0)

偏移地址: 0x0038

表 238 TIM 捕获/比较寄存器 0

位	访问	描述
31:16	r/w	<p>ACR0[15:0]: 通道 0 非对称比较值</p> <p>ACR0 是非对称比较寄存器 0 的预装载值。如果没有通过 TIM_CCMR0 寄存器中的 OC0PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 0)。实际非对称比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC0 输出上发出信号的值。</p> <p>复位值: 0x0</p>
15:0	r/w	<p>CCR0[15:0]: 捕获/比较 0 值</p> <p>通道 CC0 配置为输出。CCR0 是捕获/比较寄存器 0 的预装载值。如果没有通过 TIM_CCMR0 寄存器中的 OC0PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 0)。实际捕获/比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC0 输出上发出信号的值。</p> <p>通道 CC0 配置为输入。CCR0 为上一个输入捕获 0 事件 (IC0) 发生时的计数器值。此时该寄存器只读不可写。</p> <p>复位值: 0x0</p>

15.5.16 TIM 捕获/比较寄存器 1 (TIM_CCR1)

偏移地址: 0x003C

表 239 TIM 捕获/比较寄存器 1

位	访问	描述
31:16	r/w	<p>ACR1[15:0]: 通道 1 非对称比较值</p> <p>ACR1 是非对称比较寄存器 1 的预装载值。如果没有通过 TIM_CCMR0 寄存器中的 OC1PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 1)。实际非对称比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC1 输出上发出信号的值。</p> <p>复位值: 0x0</p>
15:0	r/w	<p>CCR1[15:0]: 捕获/比较 1 值</p> <p>通道 CC1 配置为输出。CCR1 是捕获/比较寄存器 1 的预装载值。如果没有通过 TIM_CCMR0 寄存器中的 OC1PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 1)。实际捕获/比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC1 输出上发出信号的值。</p> <p>通道 CC1 配置为输入。CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。此时该寄存器只读不可写。</p> <p>复位值: 0x0</p>

15.5.17 TIM 捕获/比较寄存器 2 (TIM_CCR2)

偏移地址: 0x0040

表 240 TIM 捕获/比较寄存器 2

位	访问	描述
31:16	r/w	<p>ACR2[15:0]: 通道 2 非对称比较值</p> <p>ACR2 是非对称比较寄存器 2 的预装载值。如果没有通过 TIM_CCMR1 寄存器中的 OC2PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 2)。实际非对称比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC2 输出上发出信号的值。</p> <p>复位值: 0x0</p>
15:0	r/w	<p>CCR2[15:0]: 捕获/比较 2 值</p> <p>通道 CC2 配置为输出。CCR2 是捕获/比较寄存器 2 的预装载值。如果没有通过 TIM_CCMR1 寄存器中的 OC2PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 2)。实际捕获/比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC2 输出上发出信号的值。</p> <p>通道 CC2 配置为输入。CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。此时该寄存器只读不可写。</p> <p>复位值: 0x0</p>

15.5.18 TIM 捕获/比较寄存器 3 (TIM_CCR3)

偏移地址: 0x0044

表 241 TIM 捕获/比较寄存器 3

位	访问	描述
31:16	r/w	<p>ACR3[15:0]: 通道 3 非对称比较值</p> <p>ACR3 是非对称比较寄存器 3 的预装载值。如果没有通过 TIM_CCMR1 寄存器中的 OC3PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 3)。实际非对称比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC3 输出上发出信号的值。</p> <p>复位值: 0x0</p>
15:0	r/w	<p>CCR3[15:0]: 捕获/比较 3 值</p> <p>通道 CC3 配置为输出。CCR3 是捕获/比较寄存器 3 的预装载值。如果没有通过 TIM_CCMR1 寄存器中的 OC3PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 3)。实际捕获/比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC3 输出上发出信号的值。</p> <p>通道 CC3 配置为输入。CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。此时该寄存器只读不可写。</p>

位	访问	描述
		复位值: 0x0

15.5.19 TIM 捕获/比较寄存器 4 (TIM_CCR4)

偏移地址: 0x0048

表 242 TIM 捕获/比较寄存器 4

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	CCR4[15:0]: 捕获/比较 4 值 CCR4 是捕获/比较寄存器 4 的预装载值。如果没有通过 TIM_CCMR2 寄存器中的 OC4PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 4)。实际捕获/比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC3REF 输出上发出信号的值。 复位值: 0x0

15.5.20 TIM 捕获/比较寄存器 5 (TIM_CCR5)

偏移地址: 0x004C

表 243 TIM 捕获/比较寄存器 5

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	CCR5[15:0]: 捕获/比较 5 值 CCR5 是捕获/比较寄存器 5 的预装载值。如果没有通过 TIM_CCMR2 寄存器中的 OC5PE 位来使能预装载功能, 写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效 (拷贝到实际起作用的捕获/比较寄存器 5)。实际捕获/比较寄存器中包含要与计数器 TIM_CNT 进行比较并在 OC5REF 输出上发出信号的值。 复位值: 0x0

15.5.21 TIM 断路及死区寄存器 (TIM_BDTR)

偏移地址: 0x0050

表 244 TIM 断路及死区寄存器

位	访问	描述
31:24	r/w	DTGP[7:0]: 配置死区发生器 (上升沿) 此位域定义插入到互补输出之间的死区持续时间 (OCxREF 的上升沿)。相关描述参考 DTG。 当 DTGM =0 时, 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1、2 或 3, 此位域即无法修改。 复位值: 0x0
23:18	Res	Reserved 复位值: 0x0
17	r/w	DTGM: 死区时间配置模式选择 0: 上升沿和下降沿的死区时间配置共用 DTG 1: 上升沿和下降沿的死区时间配置分别对应 DTGP 和 DTG 注意: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1、2 或 3, 此位域即无法修改。 复位值: 0x0
16	r/w	DTGPE: 死区时间配置预装载使能 0: 禁止与 DTG&DTGP 相关的预装载寄存器。可随时向 DTG&DTGP 写入数据, 写入后将立即使用新值。 1: 使能与 DTG&DTGP 相关的预装载寄存器。可读/写访问预装载寄存器。DTG&DTGP 预装载值在每次生成更新事件时都会装载到活动寄存器中。 注意: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1、2 或 3, 此位域即无法修改。 复位值: 0x0
15	r/w	MOE: 主输出使能 只要断路输入变为有效状态, 此位便由硬件异步清零。此位由软件置 1, 也可根据 AOE 位状态自动置 1。此位仅对配置为输出的通道有效。 0: OC 和 OCN 输出禁止或被强制为空闲状态。 1: 如果 OC 和 OCN 输出的相应使能位 (TIM_CCER 寄存器中的 CCxE 和 CCxNE 位) 均置 1, 则使能 OC 和 OCN 输出。 复位值: 0x0
14	r/w	AOE: 自动输出使能 0: MOE 只能由软件置 1 1: MOE 可由软件置 1, 也可在发生下一更新事件时自动置 1 (如果断路输入无效) 注: 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1, 此位即无法修改 复位值: 0x0
13	res	保留 详见 TIM 外部断路控制寄存器 1 (TIM_EBCR1, 0x40014824) TIM8_BRK_POL 描述 复位值: 0x0

位	访问	描述
12	r/w	<p>BKE: 断路使能</p> <p>0: 禁止断路输入</p> <p>1: 使能断路输入</p> <p>注意: 编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1 后, 此位即无法修改。注意: 对该位执行任何写操作后, 都需要经过 1 个 APB 时钟周期的延迟才生效。</p> <p>复位值: 0x0</p>
11:10	Res	<p>Reserved</p> <p>复位值: 0x0</p>
9:8	r/w	<p>LOCK[1:0]: 锁定配置 (Lock Configuration)</p> <p>这些位用于针对软件错误提供写保护。</p> <p>00: 关闭锁定--不对任何位提供写保护。</p> <p>01: 锁定级别 1, 此时无法对 TIM_BDTR 寄存器中的 DTG*位 (DTGM=0)、TIM_CR1 寄存器中的 OISx 和 OISxN 位以及 TIM_BDTR 寄存器中的 BKE/BKP/AOE 位执行写操作。</p> <p>10: 锁定级别 2, 此时无法对锁定级别 1 中适用的各位、CC 极性位 (TIM_CCER 寄存器中的 CCxP/CCxNP 位, 只要通过 CCxS 位将相关通道配置为输出) 以及 OSSR 和 OSSI 位执行写操作。</p> <p>11: 锁定级别 3, 此时无法对锁定级别 2 中适用的各位、CC 控制位 (TIM_CCMRx 寄存器中的 OCxM 和 OCxPE 位, 只要通过 CCxS 位将相关通道配置为输出) 执行写操作。</p> <p>注意: 复位后只能对 LOCK 位执行一次写操作。对 TIM_BDTR 寄存器执行写操作后其中的内容将冻结, 直到下一次复位。</p> <p>复位值: 0x0</p>
7:0	r/w	<p>DTG[7:0]: 配置死区发生器 (下降沿)</p> <p>此位域定义插入到互补输出之间的死区持续时间 (OCxREF 的下降沿)。死区时间 DT 与该持续时间相对应:</p> <p>DTG[7:5]=0XX: $DT=DTG[7:0]*t_{dtg}$, 其中 $t_{dtg}=t_{DTS}$。</p> <p>DTG[7:5]=10X: $DT=(64+DTG[5:0])*t_{dtg}$, 其中 $t_{dtg}=2*t_{DTS}$。</p> <p>DTG[7:5]=110: $DT=(32+DTG[4:0])*t_{dtg}$, 其中 $t_{dtg}=8*t_{DTS}$。</p> <p>DTG[7:5]=111: $DT=(32+DTG[4:0])*t_{dtg}$, 其中 $t_{dtg}=16*t_{DTS}$。</p> <p>例如, 如果 $t_{DTS}=125ns(8MHz)$, 则可能的死区值为:</p> <p>0 到 15875ns (步长为 125ns)</p> <p>16us 到 31750ns (步长为 250ns)</p> <p>32us 到 63us (步长为 1μs)</p> <p>64us 到 126us (步长为 2μs)</p> <p>注意: 当 DTGM=0 时, 只要编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1、2 或 3, 此位域即无法修改。</p> <p>复位值: 0x0</p>

15.5.22 TIM 事件控制寄存器 (TIM_ECR)

偏移地址: 0x0060

表 245 TIM 事件控制寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:12	r/w	EVNT1PSC: EVNT1 事件预分频 该配置为 EVNT1 事件有效生成须累计有效事件源次数 (EVNT1PSC+1) 复位值: 0x0
11	Res	Reserved
10	r/w	EVNT1CNTCLR: EVNT1 事件计数器清零 0: 无 EVNT1 事件计数器清零请求 1: 有 EVNT1 事件计数器清零请求 注: 该位自动清零 复位值: 0x0
9:8	r/w	EVNT1SEL: EVNT1 事件源选择 00: OC5REF 上升沿用作 EVNT1 事件源 01: OC5REF 下降沿用作 EVNT1 事件源 1X: OC5REF 上升沿或下降沿用作 EVNT1 事件源 复位值: 0x0
7	Res	Reserved 复位值: 0x0
6:4	r/w	EVNT0PSC: EVNT0 事件预分频 该配置为 EVNT0 事件有效生成须累计有效事件源次数 (EVNT0PSC+1)。 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2	r/w	EVNT0CNTCLR: EVNT0 事件计数器清零 0: 无 EVNT0 事件计数器清零请求 1: 有 EVNT0 事件计数器清零请求 注: 该位自动清零 复位值: 0x0
1:0	r/w	EVNT0SEL: EVNT0 事件源选择 00: OC4REF 上升沿用作 EVNT0 事件源 01: OC4REF 下降沿用作 EVNT0 事件源 1X: OC4REF 上升沿或下降沿用作 EVNT0 事件源 复位值: 0x0

15.5.23 TIM 配置密钥寄存器 (TIM_SET_KR)

偏移地址: 0x0064

表 246 TIM 配置密钥寄存器 (写)

位	访问	描述
31:0	wo	<p>TIM_SET_UNLOCK_KEY: TIM 配置寄存器解锁密钥</p> <p>注: 在默认情况下, TIM 配置寄存器 (TIM_CNT)是不能直接进行写操作, 在写之前需要输入正确密钥 0xD79B3EFD 以解锁写权限, 即 TIM_SET_UNLOCK =1, 在更新完系统配置寄存器之后, 写入其他值便可重新锁定写权限, 即 TIM_SET_UNLOCK =0。</p> <p>复位值: 0x0</p>

表 247 TIM 配置密钥寄存器 (读)

位	访问	描述
31:1	Res	<p>Reserved</p> <p>复位值: 0x0</p>
0	ro	<p>TIM_SET_UNLOCK, TIM 配置寄存器解锁与否</p> <p>0: 锁定</p> <p>1: 解锁</p> <p>复位值: 0x0</p>

16 定时器选项 (TIMOR)

16.1 概述

该章节介绍了定时器输入选择控制，主要包括以下几个部分：刹车输入、通道输入、内部触发输入、ETR 输入和 OCREF_CLR 输入。

16.2 主要功能

- 外部刹车极性可编程
- 系统异常事件作为刹车源管理
- 通道输入管理
- 内部触发连接管理
- ETR 输入管理
- OCREF_CLR 输入管理

Kiwi Instruments Corp. Confidential

16.3 模块框图

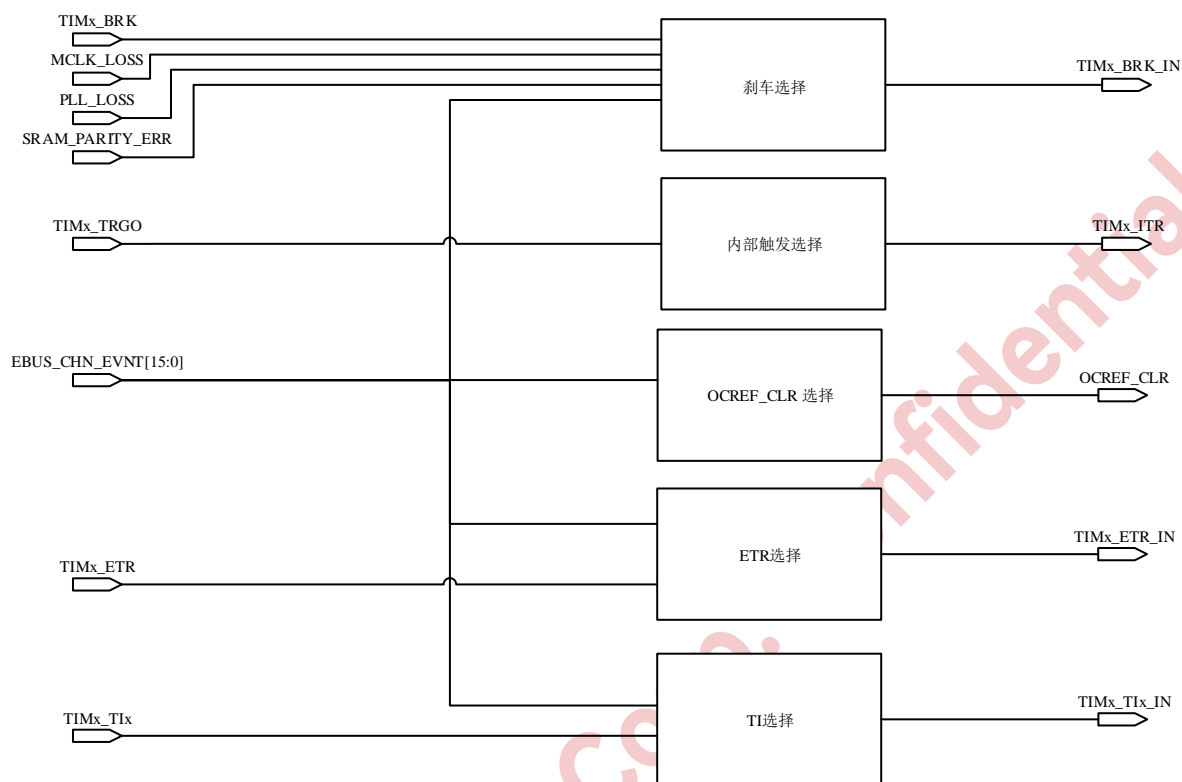


图 160 定时器选项总体模块框图

16.4 功能描述

16.4.1 断路输入控制

定时器支持这几种断路源：外部输入断路、EBUS 事件、SRAM 奇偶校验错误、时钟丢失事件和 CPU 进入 HALT 状态。所有的断路源都有独立的开关使能控制、外部断路输入支持极性选择，而 EBUS 事件极性则在 EBUS IP 内部完成极性配置，另外其他的断路源仅支持高电平有效。

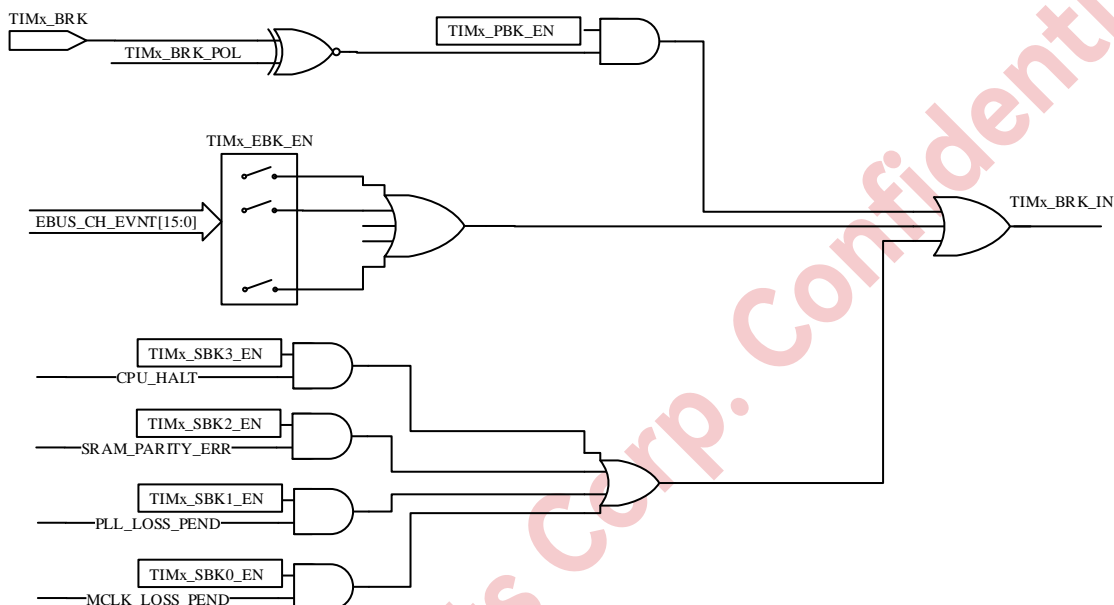


图 161 定时器断路源示意图

16.4.2 内部触发管理

多个定时器可以通过内部触发信号连接起来进行联动，从而实现预分频、通道扩展、精确同步等功能。每一类定时器的内部触发信号输入的对应关系如下面三个表格所示。

表 248 定时器 TIMAx 内部触发信号对应关系

从属端 主机端	TIMA0	TIMA1	TIMH
TIMA0	-	V	V
TIMA1	V	-	V
TIMH	V	V	
TIMG0	V	V	V
TIMG1	V	V	V
TIMG2	V	V	V

TIMS0	V	V	V
TIMS1	V	V	V
TIMS2	V	V	V
TIMS3	V	V	V

表 249 定时器 TIMGx 内部触发信号对应关系

从属端 主机端	TIMG0	TIMG1	TIMG2
TIMA0	V	V	V
TIMA1	V	V	V
TIMH	V	V	V
TIMG0	-	V	V
TIMG1	V	-	V
TIMG2	V	V	-
TIMS0	V	V	V
TIMS1	V	V	V
TIMS2	V	V	V
TIMS3	V	V	V

表 250 定时器 TIMSx 内部触发信号对应关系

从属端 主机端	TIMS0	TIMS1	TIMS2	TIMS3
TIMA0	V	V	V	V
TIMA1	V	V	V	V
TIMH	V	V	V	V
TIMG0	V	V	V	V
TIMG1	V	V	V	V
TIMG2	V	V	V	V
TIMS0	-	V	V	V
TIMS1	V	-	V	V
TIMS2	V	V	-	V
TIMS3	V	V	V	-

16.4.3 通道输入管理

定时器的通道输入除了源自于外部引脚正常输入，支持多个外部引脚的组合输入，而且 also 支持 EBUS 事件输入。下图展示了定时器通道输入所支持的输入功能。

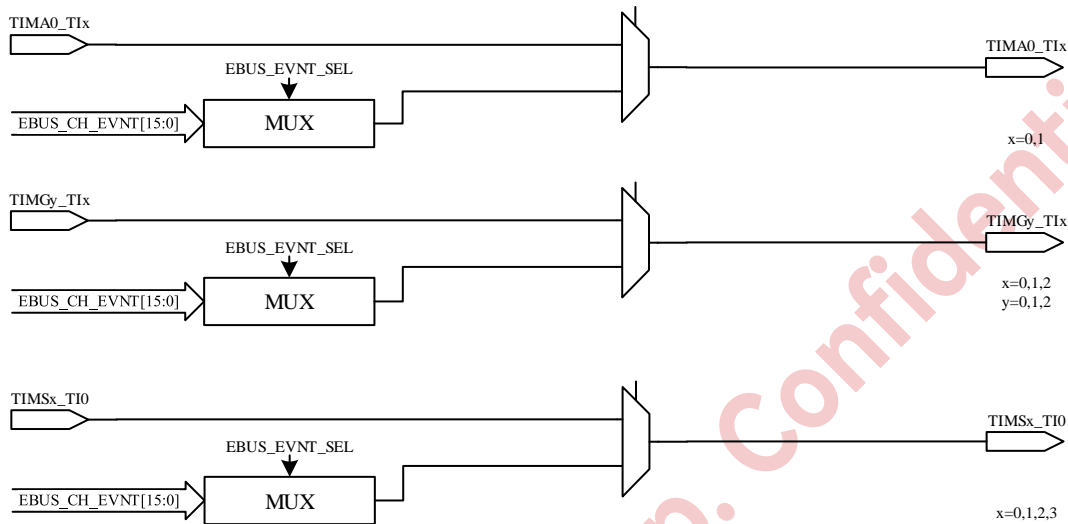


图 162 通道输入源示意图

16.4.4 ETR 输入管理

定时器的通道输入除了源自于外部引脚正常输入，而且 also 支持 EBUS 事件输入。下表展示了定时器通道输入所支持的输入功能。

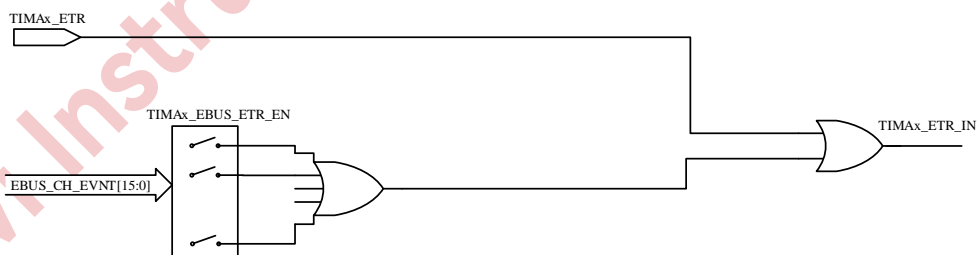


图 163 ETR 输入源示意图

16.4.5 OCREF_CLR 输入管理

定时器的 OCREF_CLR 主要就是来源于 CMP 事件输入。

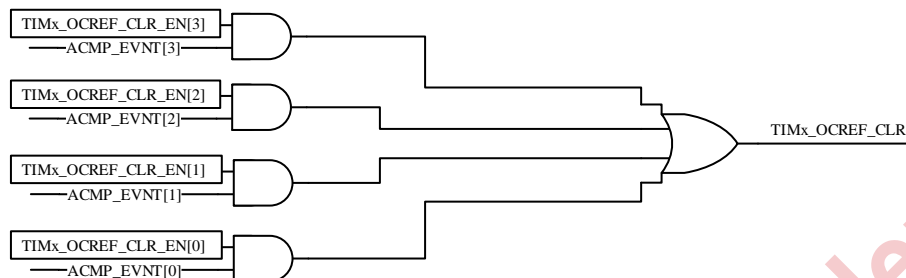


图 164 OCREF_CLR 输入源示意图

16.5 寄存器

TIMOR 寄存器基地址是 0x40046000。下表是 TIMOR 的寄存器寄存器列表。

表 251 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	TIMA0 输入复用控制寄存器	32	r/w	0x00000000
0x04~0x08	Reserved	32	RES	0x00000000
0x0C	TIMG0 输入复用控制寄存器	32	r/w	0x00000000
0x10	TIMG1 输入复用控制寄存器	32	r/w	0x00000000
0x14	TIMG2 输入复用控制寄存器	32	r/w	0x00000000
0x18	TIMS 输入复用控制寄存器	32	r/w	0x00000000
0x1C~0x3C	Reserved	32	RES	0x00000000
0x40	TIMA 内部触发控制寄存器	32	r/w	0x00000000
0x44	TIMG 内部触发控制寄存器	32	r/w	0x00000000
0x48	TIMS 内部触发控制寄存器	32	r/w	0x00000000
0x4C	Reserved	32	RES	0x00000000
0x50	TIMA0 断路控制寄存器	32	r/w	0x00000000
0x54	TIMA1 断路控制寄存器	32	r/w	0x00000000
0x58	TIMH 断路控制寄存器	32	r/w	0x00000000
0x5C	TIMS0 断路控制寄存器	32	r/w	0x00000000
0x60	TIMS1 断路控制寄存器	32	r/w	0x00000000
0x64	TIMS2 断路控制寄存器	32	r/w	0x00000000

偏移地址	寄存器列表	位宽	访问	复位值
0x68	TIMS3 断路控制寄存器	32	r/w	0x00000000
0x6C	Reserved	32	RES	0x00000000
0x70	TIMA 外部触发控制寄存器 0	32	r/w	0x00000000
0x74~0x7C	Reserved	32	RES	0x00000000
0x80	TIMA OCREF 清除控制寄存器	32	r/w	0x00000000
0x84	TIMG OCREF 清除控制寄存器	32	r/w	0x00000000
0x88	TIMS OCREF 清除控制寄存器	32	r/w	0x00000000
0x8C~0xFC	Reserved	32	RES	0x00000000

16.5.1 TIMA0 输入复用控制寄存器 (TIMA0_IMCR)

偏移地址: 0x0000

表 252 TIMA0 输入复用控制寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:12	r/w	TIMA0_TI1_EBUS_SEL: EBUS 事件源选择作为 TIMA0 TI1 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
11:9	Res	Reserved 复位值: 0x0
8	r/w	TIMA0_TI1_SEL: TIMA0 TI1 输入复用选择 0: 普通端口输入作为 TI1 输入 1: EBUS 事件作为 TI1 输入 复位值: 0x0
7:4	r/w	TIMA0_TI0_EBUS_SEL: EBUS 事件源选择作为 TIMA0 TI0 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	TIMA0_TI0_SEL: TIMA0 TI0 输入复用选择 0: 普通端口输入作为 TI0 输入 1: EBUS 事件作为 TI0 输入 复位值: 0x0

16.5.2 TIMG0 输入复用控制寄存器 (TIMG0_IMCR)

偏移地址: 0x000C

表 253 TIMG0 输入复用控制寄存器

位	访问	描述
31:24	Res	Reserved 复位值: 0x0
23:20	r/w	TIMG0_TI2_EBUS_SEL: EBUS 事件源选择作为 TIMG0 TI2 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
19:17	Res	Reserved 复位值: 0x0
16	r/w	TIMG0_TI2_SEL: TIMG0 TI2 输入复用选择 0: 普通端口输入作为 TI2 输入 1: EBUS 事件作为 TI2 输入 复位值: 0x0
15:12	r/w	TIMG0_TI1_EBUS_SEL: EBUS 事件源选择作为 TIMG0 TI1 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
11:9	Res	Reserved 复位值: 0x0
8	r/w	TIMG0_TI1_SEL: TIMG0 TI1 输入复用选择 0: 普通端口输入作为 TI1 输入 1: EBUS 事件作为 TI1 输入 复位值: 0x0
7:4	r/w	TIMG0_TIO_EBUS_SEL: EBUS 事件源选择作为 TIMG0 TIO 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	TIMG0_TIO_SEL: TIMG0 TIO 输入复用选择 0: 普通端口输入作为 TIO 输入 1: EBUS 事件作为 TIO 输入 复位值: 0x0

16.5.3 TIMG1 输入复用控制寄存器 (TIMG1_IMCR)

偏移地址: 0x0010

表 254 TIMG1 输入复用控制寄存器

位	访问	描述
31:24	Res	Reserved 复位值: 0x0
23:20	r/w	TIMG1_TI2_EBUS_SEL: EBUS 事件源选择作为 TIMG1 TI2 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
19:17	Res	Reserved 复位值: 0x0
16	r/w	TIMG1_TI2_SEL: TIMG1 TI2 输入复用选择 0: 普通端口输入作为 TI2 输入 1: EBUS 事件作为 TI2 输入 复位值: 0x0
15:12	r/w	TIMG1_TI1_EBUS_SEL: EBUS 事件源选择作为 TIMG1 TI1 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
11:9	Res	Reserved 复位值: 0x0
8	r/w	TIMG1_TI1_SEL: TIMG1 TI1 输入复用选择 0: 普通端口输入作为 TI1 输入 1: EBUS 事件作为 TI1 输入 复位值: 0x0
7:4	r/w	TIMG1_TI0_EBUS_SEL: EBUS 事件源选择作为 TIMG1 TI0 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	TIMG1_TI0_SEL: TIMG1 TI0 输入复用选择 0: 普通端口输入作为 TI0 输入 1: EBUS 事件作为 TI0 输入 复位值: 0x0

16.5.4 TIMG2 输入复用控制寄存器 (TIMG2_IMCR)

偏移地址: 0x0014

表 255 TIMG2 输入复用控制寄存器

位	访问	描述
31:24	Res	Reserved 复位值: 0x0
23:20	r/w	TIMG2_TI2_EBUS_SEL: EBUS 事件源选择作为 TIMG2 TI2 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
19:17	Res	Reserved 复位值: 0x0
16	r/w	TIMG2_TI2_SEL: TIMG2 TI2 输入复用选择 0: 普通端口输入作为 TI2 输入 1: EBUS 事件作为 TI2 输入 复位值: 0x0
15:12	r/w	TIMG2_TI1_EBUS_SEL: EBUS 事件源选择作为 TIMG2 TI1 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
11:9	Res	Reserved 复位值: 0x0
8	r/w	TIMG2_TI1_SEL: TIMG2 TI1 输入复用选择 0: 普通端口输入作为 TI1 输入 1: EBUS 事件作为 TI1 输入 复位值: 0x0
7:4	r/w	TIMG2_TI0_EBUS_SEL: EBUS 事件源选择作为 TIMG2 TI0 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	TIMG2_TI0_SEL: TIMG2 TI0 输入复用选择 0: 普通端口输入作为 TI0 输入 1: EBUS 事件作为 TI0 输入 复位值: 0x0

16.5.5 TIMS 输入复用控制寄存器 (TIMS_IMCR)

偏移地址: 0x0018

表 256 TIMS 输入复用控制寄存器

位	访问	描述
31:28	r/w	TIMS3_TI0_EBUS_SEL: EBUS 事件源选择作为 TIMS3 TI0 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
27:25	Res	Reserved 复位值: 0x0
24	r/w	TIMS3_TI0_SEL: TIMS3 TI0 输入复用选择 0: 普通端口输入作为 TI0 输入 1: EBUS 事件作为 TI0 输入 复位值: 0x0
23:20	r/w	TIMS2_TI0_EBUS_SEL: EBUS 事件源选择作为 TIMS2 TI0 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
19:17	Res	Reserved 复位值: 0x0
16	r/w	TIMS2_TI0_SEL: TIMS2 TI0 输入复用选择 0: 普通端口输入作为 TI0 输入 1: EBUS 事件作为 TI0 输入 复位值: 0x0
15:12	r/w	TIMS1_TI0_EBUS_SEL: EBUS 事件源选择作为 TIMS1 TI0 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
11:9	Res	Reserved 复位值: 0x0
8	r/w	TIMS1_TI0_SEL: TIMS1 TI0 输入复用选择 0: 普通端口输入作为 TI0 输入 1: EBUS 事件作为 TI0 输入 复位值: 0x0

位	访问	描述
7:4	r/w	TIMS0_TI0_EBUS_SEL: EBUS 事件源选择作为 TIMS0 TI0 输入 0: EBUS 事件通道 0 被选中 1: EBUS 事件通道 1 被选中 ... 15: EBUS 事件通道 15 被选中 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	TIMS0_TI0_SEL: TIMS0 TI0 输入复用选择 0: 普通端口输入作为 TI0 输入 1: EBUS 事件作为 TI0 输入 复位值: 0x0

16.5.6 TIMA 内部触发控制寄存器 (TIMA_ITCR)

偏移地址: 0x0040

表 257 TIMA 内部触发控制寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7:4	r/w	TIMA1_ITR_SEL, TIMA1 内部触发源选择 0000: TIMA0_TRGO 0001: 保留 0010: TIMH_EVNTA 0011: 保留 0100: TIMG0_TRGO 0101: TIMG1_TRGO 0110: TIMG2_TRGO 0111: 保留 1000: TIMS0_TRGO 1001: TIMS1_TRGO 1010: TIMS2_TRGO 1011: TIMS3_TRGO 其他: 保留 复位值: 0x0
3:0	r/w	TIMA0_ITR_SEL, TIMA0 内部触发源选择 0000: 保留 0001: TIMA1_TRGO 0010: TIMH_EVNTA 0011: 保留 0100: TIMG0_TRGO

位	访问	描述
		0101: TIMG1_TRGO 0110: TIMG2_TRGO 0111: 保留 1000: TIMS0_TRGO 1001: TIMS1_TRGO 1010: TIMS2_TRGO 1011: TIMS3_TRGO 其他: 保留 复位值: 0x0

16.5.7 TIMG 内部触发控制寄存器 (TIMG_ITCR)

偏移地址: 0x0044

表 258 TIMG 内部触发控制寄存器

位	访问	描述
31:12	Res	Reserved 复位值: 0x0
11:8	r/w	TIMG2_ITR_SEL, TIMG2 内部触发源选择 0000: TIMA0_TRGO 0001: TIMA1_TRGO 0010: TIMH_EVNTA 0011: 保留 0100: TIMG0_TRGO 0101: TIMG1_TRGO 0110: 保留 0111: 保留 1000: TIMS0_TRGO 1001: TIMS1_TRGO 1010: TIMS2_TRGO 1011: TIMS3_TRGO 其他: 保留 复位值: 0x0
7:4	r/w	TIMG1_ITR_SEL, TIMG1 内部触发源选择 0000: TIMA0_TRGO 0001: TIMA1_TRGO 0010: TIMH_EVNTA 0011: 保留 0100: TIMG0_TRGO 0101: 保留 0110: TIMG2_TRGO 0111: 保留

位	访问	描述
		1000: TIMS0_TRGO 1001: TIMS1_TRGO 1010: TIMS2_TRGO 1011: TIMS3_TRGO 其他: 保留 复位值: 0x0
3:0	r/w	TIMG0_ITR_SEL, TIMG0 内部触发源选择 0000: TIMA0_TRGO 0001: TIMA1_TRGO 0010: TIMH_EVNTA 0011: 保留 0100: 保留 0101: TIMG1_TRGO 0110: TIMG2_TRGO 0111: 保留 1000: TIMS0_TRGO 1001: TIMS1_TRGO 1010: TIMS2_TRGO 1011: TIMS3_TRGO 其他: 保留 复位值: 0x0

16.5.8 TIMS 内部触发控制寄存器 (TIMS_ITCR)

偏移地址: 0x0048

表 259 TIMS 内部触发控制寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:12	r/w	TIMS3_ITR_SEL, TIMS3 内部触发源选择 0000: TIMA0_TRGO 0001: TIMA1_TRGO 0010: TIMH_EVNTA 0011: 保留 0100: TIMG0_TRGO 0101: TIMG1_TRGO 0110: TIMG2_TRGO 0111: 保留 1000: TIMS0_TRGO 1001: TIMS1_TRGO 1010: TIMS2_TRGO

位	访问	描述
		1011: 保留 其他: 保留 复位值: 0x0
11:8	r/w	TIMS2_ITR_SEL, TIMS2 内部触发源选择 0000: TIMA0_TRGO 0001: TIMA1_TRGO 0010: TIMH_EVNTA 0011: 保留 0100: TIMG0_TRGO 0101: TIMG1_TRGO 0110: TIMG2_TRGO 0111: 保留 1000: TIMS0_TRGO 1001: TIMS1_TRGO 1010: 保留 1011: TIMS3_TRGO 其他: 保留 复位值: 0x0
7:4	r/w	TIMS1_ITR_SEL, TIMS1 内部触发源选择 0000: TIMA0_TRGO 0001: TIMA1_TRGO 0010: TIMH_EVNTA 0011: 保留 0100: TIMG0_TRGO 0101: TIMG1_TRGO 0110: TIMG2_TRGO 0111: 保留 1000: TIMS0_TRGO 1001: 保留 1010: TIMS2_TRGO 1011: TIMS3_TRGO 其他: 保留 复位值: 0x0
3:0	r/w	TIMS0_ITR_SEL, TIMS0 内部触发源选择 0000: TIMA0_TRGO 0001: TIMA1_TRGO 0010: TIMH_EVNTA 0011: 保留 0100: TIMG0_TRGO 0101: TIMG1_TRGO 0110: TIMG2_TRGO 0111: 保留

位	访问	描述
		1000: 保留 1001: TIMS1_TRGO 1010: TIMS2_TRGO 1011: TIMS3_TRGO 其他: 保留 复位值: 0x0

16.5.9 TIMA0 断路控制寄存器 (TIMA0_SBCR)

偏移地址: 0x0050

表 260 TIMA0 断路控制寄存器

位	访问	描述
31:16	r/w	TIMA0_EBK_EN[15:0], EBUS 事件刹车源使能控制 位 0 对应 EBUS 事件通道 0: 0, 关闭; 1, 开启 位 1 对应 EBUS 事件通道 1: 0, 关闭; 1, 开启 ... 位 15 对应 EBUS 事件通道 15: 0, 关闭; 1, 开启 复位值: 0x0
15:6	Res	Reserved 复位值: 0x0
5	r/w	TIMA0_BRK_POL, TIMA0 外部断路源极性选择 0: 低电平有效 1: 高电平有效 复位值: 0x0
4	r/w	TIMA0_PBK_EN, TIMA0 外部断路源使能控制 0: 关闭 1: 开启 复位值: 0x0
3	r/w	TIMA0_SBK3EN, TIMA0 系统断路源 3 能控制 0: 关闭 1: 开启 复位值: 0x0
2	r/w	TIMA0_SBK2_EN, TIMA0 系统断路源 2 使能控制 0: 关闭 1: 开启 复位值: 0x0
1	r/w	TIMA0_SBK1_EN, TIMA0 系统断路源 1 使能控制 0: 关闭 1: 开启 复位值: 0x0

位	访问	描述
0	r/w	TIMA0_SBK0_EN, TIMA0 系统断路源 0 使能控制 0: 关闭 1: 开启 复位值: 0x0

16.5.10 TIMA1 断路控制寄存器 (TIMA1_SBCR)

偏移地址: 0x0054

表 261 TIMA1 断路控制寄存器

位	访问	描述
31:16	r/w	TIMA1_EBK_EN[15:0], EBUS 事件刹车源使能控制 位 0 对应 EBUS 事件通道 0: 0, 关闭; 1, 开启 位 1 对应 EBUS 事件通道 1: 0, 关闭; 1, 开启 ... 位 15 对应 EBUS 事件通道 15: 0, 关闭; 1, 开启 复位值: 0x0
15:6	Res	Reserved 复位值: 0x0
5	r/w	TIMA1_BRK_POL, TIMA1 外部断路源极性选择 0: 低电平有效 1: 高电平有效 复位值: 0x0
4	r/w	TIMA1_PBK_EN, TIMA1 外部断路源使能控制 0: 关闭 1: 开启 复位值: 0x0
3	r/w	TIMA1_SBK3EN, TIMA1 系统断路源 3 能控制 0: 关闭 1: 开启 复位值: 0x0
2	r/w	TIMA1_SBK2_EN, TIMA1 系统断路源 2 使能控制 0: 关闭 1: 开启 复位值: 0x0
1	r/w	TIMA1_SBK1_EN, TIMA1 系统断路源 1 使能控制 0: 关闭 1: 开启 复位值: 0x0
0	r/w	TIMA1_SBK0_EN, TIMA1 系统断路源 0 使能控制 0: 关闭 1: 开启 复位值: 0x0

16.5.11 TIMH 断路控制寄存器 (TIMH_SBCR)

偏移地址: 0x0058

表 262 TIMH 断路控制寄存器

位	访问	描述
31:3	Res	Reserved 复位值: 0x0
2	r/w	TIMH_SBK2_EN, TIMH 系统断路源 2 使能控制 0: 关闭 1: 开启 复位值: 0x0
1	r/w	TIMH_SBK1_EN, TIMH 系统断路源 1 使能控制 0: 关闭 1: 开启 复位值: 0x0
0	r/w	TIMH_SBK0_EN, TIMH 系统断路源 0 使能控制 0: 关闭 1: 开启 复位值: 0x0

16.5.12 TIMS0 断路控制寄存器 (TIMS0_SBCR)

偏移地址: 0x005C

表 263 TIMS0 断路控制寄存器

位	访问	描述
31:16	r/w	TIMS0_EBK_EN[15:0], EBUS 事件刹车源使能控制 位 0 对应 EBUS 事件通道 0: 0, 关闭; 1, 开启 位 1 对应 EBUS 事件通道 1: 0, 关闭; 1, 开启 ... 位 15 对应 EBUS 事件通道 15: 0, 关闭; 1, 开启 复位值: 0x0
15:6	Res	Reserved 复位值: 0x0
5	r/w	TIMS0_BRK_POL, TIMS0 外部断路源极性选择 0: 低电平有效 1: 高电平有效 复位值: 0x0
4	r/w	TIMS0_PBK_EN, TIMS0 外部断路源使能控制 0: 关闭 1: 开启 复位值: 0x0

位	访问	描述
3	r/w	TIMS0_SBK3EN, TIMS0 系统断路源 3 能控制 0: 关闭 1: 开启 复位值: 0x0
2	r/w	TIMS0_SBK2_EN, TIMS0 系统断路源 2 使能控制 0: 关闭 1: 开启 复位值: 0x0
1	r/w	TIMS0_SBK1_EN, TIMS0 系统断路源 1 使能控制 0: 关闭 1: 开启 复位值: 0x0
0	r/w	TIMS0_SBK0_EN, TIMS0 系统断路源 0 使能控制 0: 关闭 1: 开启 复位值: 0x0

16.5.13 TIMS1 断路控制寄存器 (TIMS1_SBCR)

偏移地址: 0x0060

表 264 TIMS1 断路控制寄存器

位	访问	描述
31:16	r/w	TIMS1_EBK_EN[15:0], EBUS 事件刹车源使能控制 位 0 对应 EBUS 事件通道 0: 0, 关闭; 1, 开启 位 1 对应 EBUS 事件通道 1: 0, 关闭; 1, 开启 ... 位 15 对应 EBUS 事件通道 15: 0, 关闭; 1, 开启 复位值: 0x0
15:6	Res	Reserved 复位值: 0x0
5	r/w	TIMS1_BRK_POL, TIMS1 外部断路源极性选择 0: 低电平有效 1: 高电平有效 复位值: 0x0
4	r/w	TIMS1_PBK_EN, TIMS1 外部断路源使能控制 0: 关闭 1: 开启 复位值: 0x0
3	r/w	TIMS1_SBK3EN, TIMS1 系统断路源 3 能控制 0: 关闭 1: 开启

位	访问	描述
		复位值: 0x0
2	r/w	TIMS1_SBK2_EN, TIMS1 系统断路源 2 使能控制 0: 关闭 1: 开启 复位值: 0x0
1	r/w	TIMS1_SBK1_EN, TIMS1 系统断路源 1 使能控制 0: 关闭 1: 开启 复位值: 0x0
0	r/w	TIMS1_SBK0_EN, TIMS1 系统断路源 0 使能控制 0: 关闭 1: 开启 复位值: 0x0

16.5.14 TIMS2 断路控制寄存器 (TIMS2_SBCR)

偏移地址: 0x0064

表 265 TIMS2 断路控制寄存器

位	访问	描述
31:16	r/w	TIMS2_EBK_EN[15:0], EBUS 事件刹车源使能控制 位 0 对应 EBUS 事件通道 0: 0, 关闭; 1, 开启 位 1 对应 EBUS 事件通道 1: 0, 关闭; 1, 开启 ... 位 15 对应 EBUS 事件通道 15: 0, 关闭; 1, 开启 复位值: 0x0
15:6	Res	Reserved 复位值: 0x0
5	r/w	TIMS2_BRK_POL, TIMS2 外部断路源极性选择 0: 低电平有效 1: 高电平有效 复位值: 0x0
4	r/w	TIMS2_PBK_EN, TIMS2 外部断路源使能控制 0: 关闭 1: 开启 复位值: 0x0
3	r/w	TIMS2_SBK3EN, TIMS2 系统断路源 3 能控制 0: 关闭 1: 开启 复位值: 0x0
2	r/w	TIMS2_SBK2_EN, TIMS2 系统断路源 2 使能控制 0: 关闭

位	访问	描述
		1: 开启 复位值: 0x0
1	r/w	TIMS2_SBK1_EN, TIMS2 系统断路源 1 使能控制 0: 关闭 1: 开启 复位值: 0x0
0	r/w	TIMS2_SBK0_EN, TIMS2 系统断路源 0 使能控制 0: 关闭 1: 开启 复位值: 0x0

16.5.15 TIMS3 断路控制寄存器 (TIMS3_SBCR)

偏移地址: 0x0068

表 266 TIMS3 断路控制寄存器

位	访问	描述
31:16	r/w	TIMS3_EBK_EN[15:0], EBUS 事件刹车源使能控制 位 0 对应 EBUS 事件通道 0: 0, 关闭; 1, 开启 位 1 对应 EBUS 事件通道 1: 0, 关闭; 1, 开启 ... 位 15 对应 EBUS 事件通道 15: 0, 关闭; 1, 开启 复位值: 0x0
15:6	Res	Reserved 复位值: 0x0
5	r/w	TIMS3_BRK_POL, TIMS3 外部断路源极性选择 0: 低电平有效 1: 高电平有效 复位值: 0x0
4	r/w	TIMS3_PBK_EN, TIMS3 外部断路源使能控制 0: 关闭 1: 开启 复位值: 0x0
3	r/w	TIMS3_SBK3EN, TIMS3 系统断路源 3 能控制 0: 关闭 1: 开启 复位值: 0x0
2	r/w	TIMS3_SBK2_EN, TIMS3 系统断路源 2 使能控制 0: 关闭 1: 开启 复位值: 0x0
1	r/w	TIMS3_SBK1_EN, TIMS3 系统断路源 1 使能控制

位	访问	描述
		0: 关闭 1: 开启 复位值: 0x0
0	r/w	TIMS3_SBK0_EN, TIMS3 系统断路源 0 使能控制 0: 关闭 1: 开启 复位值: 0x0

16.5.16 TIMA 外部触发控制寄存器 0 (TIMA_ETCR0)

偏移地址: 0x0070

表 267 TIMA 外部触发寄存器 0

位	访问	描述
31:16	r/w	TIMA1_EBUS_ETR_EN [15:0], EBUS 事件刹车源使能控制 位 0 对应 EBUS 事件通道 0: 0, 关闭; 1, 开启 位 1 对应 EBUS 事件通道 1: 0, 关闭; 1, 开启 ... 位 15 对应 EBUS 事件通道 15: 0, 关闭; 1, 开启 复位值: 0x0
15:0	r/w	TIMA0_EBUS_ETR_EN[15:0], EBUS 事件选择作为外部触发事件输入 位 0 对应 EBUS 事件通道 0: 0, 关闭; 1, 开启 位 1 对应 EBUS 事件通道 1: 0, 关闭; 1, 开启 ... 位 15 对应 EBUS 事件通道 15: 0, 关闭; 1, 开启 复位值: 0x0

16.5.17 TIMA OCREF 清除控制寄存器 (TIMA_OCCR)

偏移地址: 0x0080

表 268 TIMA OCREF 清除控制寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7:4	r/w	TIMA1_OCREF_CLR_EN[3:0], TIMA1 OCREF 清除源使能控制 位 0 对应 CMP0 输出: 0, 关闭; 1, 开启 位 1 对应 CMP1 输出: 0, 关闭; 1, 开启 位 2 对应 CMP2 输出: 0, 关闭; 1, 开启 位 3 对应 CMP3 输出: 0, 关闭; 1, 开启 复位值: 0x0
3:0	r/w	TIMA0_OCREF_CLR_EN[3:0], TIMA0 OCREF 清除源使能控制 位 0 对应 CMP0 输出: 0, 关闭; 1, 开启 位 1 对应 CMP1 输出: 0, 关闭; 1, 开启

位	访问	描述
		位 2 对应 CMP2 输出：0，关闭；1，开启 位 3 对应 CMP3 输出：0，关闭；1，开启 复位值：0x0

16.5.18 TIMG OCREF 清除控制寄存器 (TIMG_OCCR)

偏移地址：0x0084

表 269 TIMG OCREF 清除控制寄存器

位	访问	描述
31:12	Res	Reserved 复位值：0x0
11:8	r/w	TIMG2_OCREF_CLR_EN[3:0]，TIMG2 OCREF 清除源使能控制 位 0 对应 CMP0 输出：0，关闭；1，开启 位 1 对应 CMP1 输出：0，关闭；1，开启 位 2 对应 CMP2 输出：0，关闭；1，开启 位 3 对应 CMP3 输出：0，关闭；1，开启 复位值：0x0
7:4	r/w	TIMG1_OCREF_CLR_EN[3:0]，TIMG1 OCREF 清除源使能控制 位 0 对应 CMP0 输出：0，关闭；1，开启 位 1 对应 CMP1 输出：0，关闭；1，开启 位 2 对应 CMP2 输出：0，关闭；1，开启 位 3 对应 CMP3 输出：0，关闭；1，开启 复位值：0x0
3:0	r/w	TIMG0_OCREF_CLR_EN[3:0]，TIMG0 OCREF 清除源使能控制 位 0 对应 CMP0 输出：0，关闭；1，开启 位 1 对应 CMP1 输出：0，关闭；1，开启 位 2 对应 CMP2 输出：0，关闭；1，开启 位 3 对应 CMP3 输出：0，关闭；1，开启 复位值：0x0

16.5.19 TIMS OCREF 清除控制寄存器 (TIMS_OCCR)

偏移地址：0x0088

表 270 TIMS OCREF 清除控制寄存器

位	访问	描述
31:16	Res	Reserved 复位值：0x0

位	访问	描述
15:12	r/w	<p>TIMS3_OCREF_CLR_EN[3:0], TIMS3 OCREF 清除源使能控制</p> <p>位 0 对应 CMP0 输出: 0, 关闭; 1, 开启</p> <p>位 1 对应 CMP1 输出: 0, 关闭; 1, 开启</p> <p>位 2 对应 CMP2 输出: 0, 关闭; 1, 开启</p> <p>位 3 对应 CMP3 输出: 0, 关闭; 1, 开启</p> <p>复位值: 0x0</p>
11:8	r/w	<p>TIMS2_OCREF_CLR_EN[3:0], TIMS2 OCREF 清除源使能控制</p> <p>位 0 对应 CMP0 输出: 0, 关闭; 1, 开启</p> <p>位 1 对应 CMP1 输出: 0, 关闭; 1, 开启</p> <p>位 2 对应 CMP2 输出: 0, 关闭; 1, 开启</p> <p>位 3 对应 CMP3 输出: 0, 关闭; 1, 开启</p> <p>复位值: 0x0</p>
7:4	r/w	<p>TIMS1_OCREF_CLR_EN[3:0], TIMS1 OCREF 清除源使能控制</p> <p>位 0 对应 CMP0 输出: 0, 关闭; 1, 开启</p> <p>位 1 对应 CMP1 输出: 0, 关闭; 1, 开启</p> <p>位 2 对应 CMP2 输出: 0, 关闭; 1, 开启</p> <p>位 3 对应 CMP3 输出: 0, 关闭; 1, 开启</p> <p>复位值: 0x0</p>
3:0	r/w	<p>TIMS0_OCREF_CLR_EN[3:0], TIMS0 OCREF 清除源使能控制</p> <p>位 0 对应 CMP0 输出: 0, 关闭; 1, 开启</p> <p>位 1 对应 CMP1 输出: 0, 关闭; 1, 开启</p> <p>位 2 对应 CMP2 输出: 0, 关闭; 1, 开启</p> <p>位 3 对应 CMP3 输出: 0, 关闭; 1, 开启</p> <p>复位值: 0x0</p>

17 高精度定时器 (TIMH)

17.1 概述

该章节描述了高精度定时器 (TIMH) 的相关功能特性及使用方式。该模块支持电机控制和电源控制方面的多种应用，包括：

- AC 转 DC
- DC 转 DC
- 交流或直流电机控制：有刷 DC、BLDC、PMSM、ACIM、SRM、步进电机等
- 逆变器
- 电池充电器
- 数字照明
- 功率因子校正 (PFC)

17.2 主要功能

- 多达 3 个独立的 PWM 生成模块，其中每个模块带双输出
- 运行模式
 - 独立边沿 PWM 模式
 - 可变相位 PWM 模式
 - 独立边沿 PWM 模式 (双输出)
 - 中心对齐 PWM 模式
 - 中心对齐 PWM 模式 (双重更新)
 - 双边沿中心对齐 PWM 模式
- 输出模式
 - 互补模式
 - 独立模式
 - 推挽模式

- 死区生成
- 死区补偿
- 前沿消隐功能 (Leading-Edge Blanking, LEB)
- 支持故障处理中将输出强行覆盖成预设值
- 支持灵活的周期、占空比更新方式
- 支持 PWM 控制输入 (PCI) 用作 PWM 引脚输出值覆盖和与外部其他定时器同步
- 支持硬件错误处理
- 高级的触发选项
- 组合 PWM 输出
- 支持 PWM 事件输出

17.3 模块框图

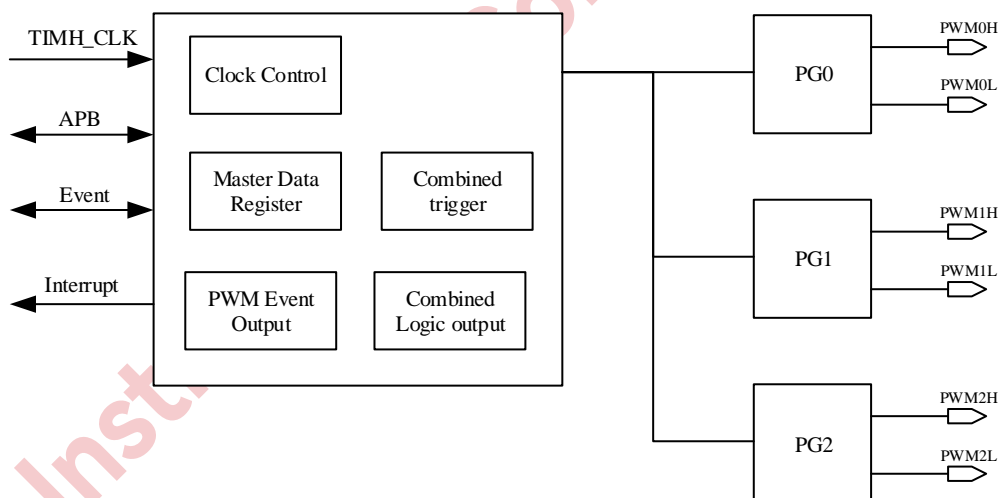


图 165 高精度定时器 TIMH 总体模块框图

PWM 模块包括一组通用的控制和功能，以及 PWM 发生器 (PGx) 的多个实例。可以独立配置每个 PWM 发生器，或者将多个 PWM 发生器组合使用以实现复杂的多相系统。此外，PWM 发生器还可用于实现复杂的触发、保护和逻辑功能。上图给出了其概要框图。

每个 PWM 发生器都是一个独立的外设，可以独立于其他 PWM 发生器进行使能。每个 PWM 发生器由一个信号发生器和一个输出控制模块组成。

PWM 发生器使用“事件”触发其他 PWM 发生器、ADC 转换和外部操作。每个 PWM 发生器接受一个触发输入并产生一个触发输出。触发输入通知 PWM 发生器何时启动新的 PWM 周期。当触发时间值等于 PWM 发生器定时器值时，将产生触发输出。

输出控制模块可改变发送到输出引脚的基本 PWM 信号，并且包含以下多种功能：

- 输出模式选择（互补、推挽和独立）
- 死区发生器
- PWM 控制输入 (PCI) 模块
- 前沿消隐 (LEB)
- 输出覆盖

每个 PWM 发生器输出模块都与两个 PWM 输出引脚的控制相关联。输出模块包含一个多用途的 PWM 控制输入 (PCI)，具体包括故障检测、外部触发以及与其他外设接口。LEB 模块与 PCI 模块配合使用，允许在 PWM 周期的某些时间段内忽略 PCI 输入。改写模块确定故障、限流和反馈控制等各类事件期间的 PWM 输出引脚状态。下图给出了单个 PWM 发生器的框图。

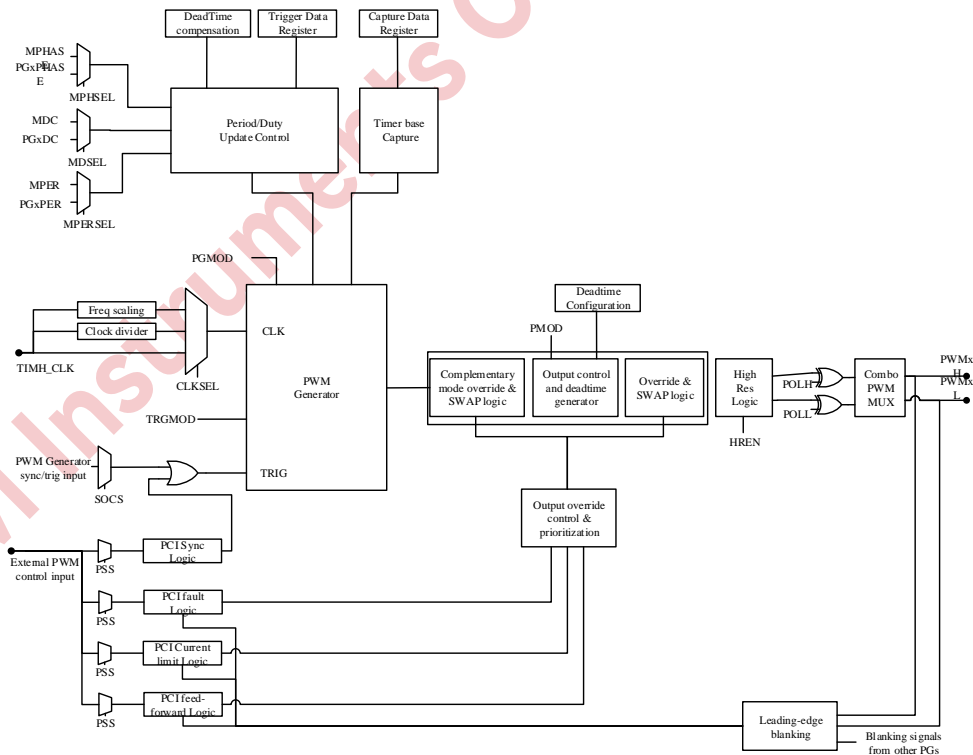


图 166 单个 PWM 发生器

PWM 发生器操作基于触发信号。要产生 PWM 周期，必须接收周期开始 (Start-of-Cycle, SOC) 触发信号；触发信号可以是自触发的，也可以是来自外部源的。章节《PWM 模式》所示为基本 PWM 波形，其中显示了 SOC 和周期结束 (End-of-Cycle, EOC) 事件。PWMxH 输出以“有效” (逻辑高电平) 状态开始周期，当内部计数器达到占空比值时，它将转换为“无效” (逻辑低电平) 状态。当计数器值达到周期值时，到达 EOC。某些工作模式和输出模式使用多个计数器周期来产生单个 PWM 周期。更多信息详见章节《PWM 模式》和《输出模式》。

17.4 功能描述

17.4.1 PWM 时钟

17.4.1.1 主时钟

PWM 模块在模块顶层提供了这几个时钟特性。每个 PWM 发生器可各自独立选择一个时钟源，如下图所示。系统时钟 TIMH_CLK 直接进入 PWM 模块，各个 PWM 发生器可通过配置 CLKSEL (PGxCON) 以选择各自的时钟，具体详见章节《PWM 发生器时钟》。而频率缩放和时钟分频器的功能在章节《共同时钟》中讨论。另外，在让 PWM 发生器开启正常功能之前需要控制位 CLKSELx 从默认更改为其他设定。

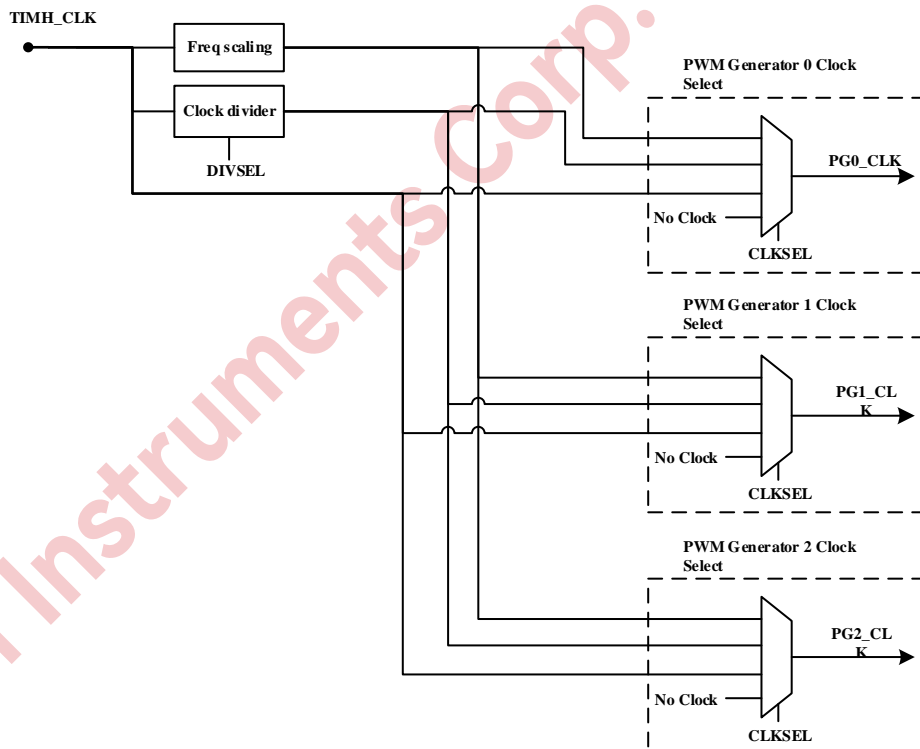


图 167 PWM 发生器时钟

17.4.1.2 标准精度时钟公式

在一些操作模式中，是利用多个时钟周期匹配以完成一个完整的 PWM 周期。下式提供了各种工作模式的时序计数公式：

公式：标准精度下 PWM 周期计算

边沿对齐模式或可变相位模式：

$$F_{PWM} = \frac{F_{PGx_CLK}}{PGxPER + 1}$$

$$PGxPER = \left(\frac{F_{PGx_CLK}}{F_{PWM}} \right) - 1$$

其中 F_{PWM} =开关频率，PWM 周期=1/ F_{PWM}

中心对齐模式，带推挽输出的边沿对齐模式或可变相位模式：

$$F_{PWM} = \frac{F_{PGx_CLK}}{2 \cdot (PGxPER + 1)}$$

$$PGxPER = \left(\frac{F_{PGx_CLK}}{2 \cdot F_{PWM}} \right) - 1$$

带推挽输出的中心对齐模式

$$F_{PWM} = \frac{F_{PGx_CLK}}{4 \cdot (PGxPER + 1)}$$

$$PGxPER = \left(\frac{F_{PGx_CLK}}{4 \cdot F_{PWM}} \right) - 1$$

公式：标准精度下 PWM 占空度、相位、触发和死区计算

MDC 或 PGxDC(A) = (PGxPER + 1) • 占空比，其中占空比是百分比值%。

MPHASE 或 PGxPHASE = F_{PGx_clk} • 相位

PGxTRIGy = F_{PGx_clk} • 触发偏移，(y = A, B or C)

PGxDTy = F_{PGx_clk} • 死区时间，(y = H or L)

上述公式中的相位、触发偏移和死区时间是以时间 (ms, μ s 或 ns)为单位。

17.4.1.3 高精度模式

PWM 发生器可在高精度模式下工作以提高相位、占空比和死区的分辨率，最高可达 156ps。高精度模式不能用于频率缩放或时钟分频启用的应用环境中。通过设置 HREN 控制位 (PGxCON) 便可为指定的 PWM 发生器启用高精度模式。状态位 HRRDY 反映了高精度相关电路是否已经准备好，而 HREER 则指示出现时钟错误。需要注意的是，当在高精度模式启用时，双 PWM 模式不能与互补输出模式结合使用。

注意：当使用高精度模式时，必须将 CLKSEL 设置为‘01’以直接选择 TIMH_CLK,并且 TIMH_CLK 需工作于系统最快的时钟频率 100MHZ。高精度模式下所能够实现的最小分辨率为 PG 时钟周期的 32 等分，也就是 $5\text{ns}/32 \approx 156\text{ps}$ ，从相关的周期、占空比等寄存器的设置基于 156ps 进行设定，配置寄存器增加 5 bit 以扩充高分辨率设定。

数据寄存器

当选择高精度模式时，一些 PWM 数据寄存器的精度存在一些限制。对于某些寄存器，无论写入的值是多少，最低有效 5 位都被强制为 0。在高精度模式下配置 PWM 相关的数据寄存器时，先设置 HREN 位，然后再对其功能与高精度模式相关的数据寄存器进行操作。高精度操作差异如下所示。

- PGxDCA: 高精度模式下，位 0~4 固定为 0
- PGxTRIGA/PGxTRIGB: 高精度模式下，位 0~4 在双 PWM 模式下可正常使用，而在其他运行模式固定为 0
- PGxTRIGC: 高精度模式下，位 0~4 固定为 0

高精度时钟公式

高精度模式下，周期计算公式如下所示：

公式：PWM 周期计算，高精度

边沿对齐模式或可变相位模式：

$$F_{PWM} = \frac{32 \cdot F_{PGxCLK}}{PGxPER + 32}$$

$$PGxPER = \left(\frac{32 \cdot F_{PGxCLK}}{F_{PWM}} \right) - 32$$

中心对齐模式，带推挽输出的边沿对齐模式或可变相位模式

$$F_{PWM} = \frac{16 \cdot F_{PGX_CLK}}{(PGXPER + 32)}$$

$$PGXPER = \left(\frac{16 \cdot F_{PGX_CLK}}{F_{PWM}} \right) - 32$$

带推挽输出的中心对齐模式

$$F_{PWM} = \frac{4 \cdot F_{PGX_CLK}}{(PGXPER + 32)}$$

$$PGXPER = \left(\frac{4 \cdot F_{PGX_CLK}}{F_{PWM}} \right) - 32$$

其中 F_{PWM} = 开关频率，PWM 周期 = $1/F_{PWM}$

公式：PWM 占空度、相位、触发和死区计算，标准精度：

MDC 或 PGxDC (A) = $(PGXPER + 32) \cdot$ 占空比，其中占空比是百分比值%。

MPHASE 或 PGxPHASE = $32 \cdot F_{PGx_clk} \cdot$ 相位

PGxTRIGy = $32 \cdot F_{PGx_clk} \cdot$ 触发偏移，(y = A, B or C)

PGxDTy = $32 \cdot F_{PGx_clk} \cdot$ 死区时间，(y = H or L)

上述公式中的相位、触发偏移和死区时间是以时间 (ms, μ s 或 ns) 为单位。

17.4.1.4 高精度周期同步

当在高精度模式下工作时，PWM 输出边缘可能不与 PWM 模块其余部分电路工作的 PGx_CLK 对齐。当 PGxPER (或 MPER) 值不能被 32 整除时，周期包含 PGx_CLK 的小数值。这种微小的时钟差异会导致其他事件，包括周期结束 (EOC)、触发等，与输出边缘不对齐。该模块包括一个累加电路可以在长时间内计算并最小化偏移量。

如果需要同步行为，建议使用位[4:0]为 0 的 PGxPER 配置。

与基础 PWM 信号相比，内嵌的精细边缘放置电路本身增加了 PWM 输出的延迟。在高精度模式下使用基础 PWM 信号进行门控或同步可能会在某些情况下对一些精细边缘时钟周期造成意想不到的结果。例如，使用 PCI 的自动终止功能将在 EOC 删除覆盖条件并将 PWM 输出恢复到其现有状态。对于经精细边缘放置电路的输出进行覆盖这种情况，在章节《输出覆盖》中的图有体现。由于 SOC 事件是基于基础 PWM 信号而来，所以在下一个 PWM 周期开始之前，可以观察到通过精细边缘电路的延迟。这种行为可以通过使用等于 PGxPER-32 的相移来缓解。

除了 EOC 事件之外，在某些情况下，使用基础 PWM 信号 (如 LEB 何 PGxTRGy) 或其他 PWM 发生器作为源也可能容易受到影响。

17.4.2 时钟同步

每个 PWM 发生器可以以不同的速度运行，这会影响到在 PWM 发生器间的交互动作。

17.4.3 最小 PWM 周期和脉宽

PWM 模块对 PWM 周期和脉宽的最小值有一定的要求。根据不同的工作模式，脉冲宽度不仅依赖于寄存器 DC，也依赖于寄存器 PHASE 和 TRIGy。最小脉冲宽度使用于 PWM 输出激活和非激活状态，同时也支持 0% 和 100% 占空比。相关限制如下表所示。

表 271 最小周期及脉宽

模式	最小周期 (PGxPER 或 MPER)	最小活跃脉冲宽度 (周期)	最小非活跃脉冲宽度 (周期)
标准精度	0x0020	0x0008	周期 - 0x0008
高精度	0x00400	0x00100	周期 - 0x00100

17.4.4 PWM 发生器 (PG) 特性

PWM 模块的大多数特性和控制是在 PWM 发生器层级通过控制其相关特殊功能寄存器实现。PWM 发生器是基于触发信号进行运作，其产生每一个 PWM 周期信号都必须接收到一个启动周期 (SOC) 触发信号。触发信号可以在 PWM 发生器外部产生，也可以是自触发的。当 PWM 发生器达到一个 PWM 周期的结束时，它便产生一个周期结束 (EOC) 触发信号，可供其他发生器使用。

如果多个 PWM 发生器以不同的频率运行，可基于 PCI 同步模块对触发信号进行同步处理。

17.4.4.1 PWM 发生器时钟

每一个 PWM 发生器都可由相互独立的时钟驱动，以获得最大的灵活性。设计中有四个时钟可供选择，由寄存器 CLKSEL (PGxCON) 控制：

1. 无时钟（最低功耗状态）
2. 原始 TIMH_CLK
3. 时钟分频器输出
4. 频率缩放输出

这种配置方式灵活，例如，允许一组 PWM 发生器在较高的频率下工作，而另一组则在较低的频率下工作。有关时钟输入的更多信息，详见章节《主时钟》。

当发生器正当运行时请勿更改 CLKSEL 的值。

17.4.4.2 PWM 模式

PWM 模块支持广泛的 PWM 模式的电机控制和电源设计。具体支持以下几种 PWM 模式：

- 独立边沿 PWM 模式（默认）
- 可变相位 PWM 模式
- 独立边沿 PWM 模式，双输出
- 中心对齐 PWM 模式
- 中心对齐 PWM 模式（双重更新）
- 双边沿中心对齐 PWM 模式

PWM 模式可通过设置 MODESEL (PGxCON) 来选择。有些模式利用多个时基周期来完成单个 PWM 周期。关于时序的细节可以参照前面章节提供的公式。

独立边沿 PWM 模式

独立边沿 PWM 模式可用于许多应用，可用于创建边沿对齐的 PWM 信号，以及具有任意相位偏移的 PWM 信号。该模式是 PWM 发生器默认的工作模式，或者将 MODESEL 配置为 000 则可选择此模式，并且必须两个数据寄存器以定义上升沿和下降沿。该模式下，PWM 信号的特性由这三个专用功能寄存器所定义：

- PGxPHASE: PWM 信号的上升沿的位置（相对于定时器计数周期开始）
- PGxDC: PWM 信号的下降沿的位置（相对于定时器计数周期开始）
- PGxPER: PWM 定时器计数周期结束点

通过设置 PGxPHASE=0，可以创建一个基本的边沿对齐 PWM 模式信号。或者，多个 PWM 发生器可通过使用相同的 PGxPHASE 值实现相互同步。等效于其他 PWM 发生器的 PGxPHASE 值的恒定值也可以用于同步多个发生器。另外，通过写入寄存器 PGxDC 可改变 PWM 信号的占空比。至此，任意相位的 PWM 信号可通过往 PGxPHASE 和 PGxDC 写入适当的值来产生。如果 PGxPHASE=PGxDC，则不会产生 PWM 脉冲。如果 $PGxDC \geq PGxPER$ ，则产生占空比为 100% 的周期信号。专用功能寄存器和输出波形之间的关系如下图所示。

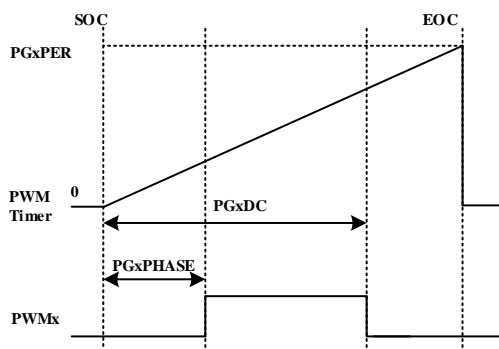


图 168 独立边沿 PWM 模式

可变相位 PWM 模式

可变相位 PWM 模式与独立边沿 PWM 模式的不同之处在于,有一个寄存器用于选择相对于周期开始时的相移,另一寄存器用于选择脉冲宽度。把 PGxDC 编程为一个固定值,调制 PGxPHASE,这种方式是很有用的。PWM 逻辑自动计算上升沿和下降沿的时间点,以保持恒定的脉冲宽度。类似地,用户也可以将 PGxPHASE 寄存器编程为一个恒定值以创建具有恒定相移和可变占空比的信号。当 MODSEL=001 时,可变相位 PWM 模式便被选中。该模式下, PWM 信号的特性由这三个专用功能寄存器所定义:

- PGxPHASE: PWM 信号的上升沿的位置 (相对于定时器计数周期开始)
- PGxDC: PWM 信号的脉冲宽度和下降沿的位置
- PGxPER: PWM 定时器计数周期结束点

专用功能寄存器和输出波形之间的关系如下图所示。

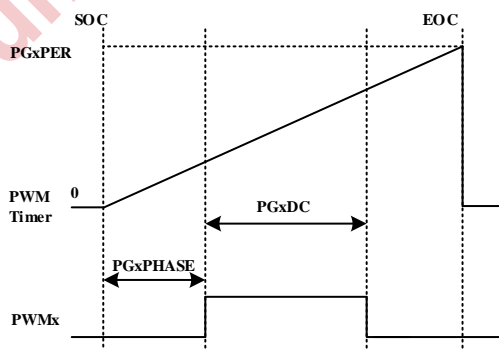


图 169 可变相位 PWM 模式

主占空比寄存器 (MDC) 也可以用于同时改变所有 PWM 信号的占空比,其中只要改写单一寄存器就可实现 (修改 MDC)。下图是一个多相位的例子。多相位模式不支持占空度跨越 EOC 的边界,要求 (相位+占空度) ≤ 周期。

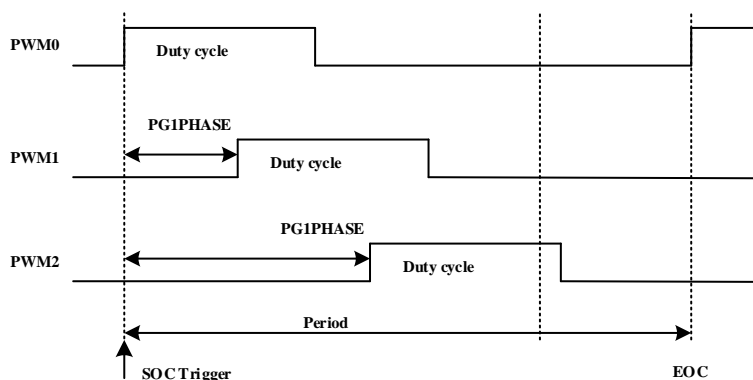


图 170 多相位 PWM 示例

双 PWM 模式

双 PWM 模式允许单个 PWM 发生器在 PWMxH 和 PWMxL 输出引脚上产生两个独立的脉宽信号。这种模式相当于独立边沿模式，只是有第二个 PWM 脉冲生成。当 MODSEL=010 时，该模式则被选中。PGxTRIGA 和 PGxTRIGB 寄存器用作第二组的 PGxPHASE 和 PGxDC 来控制第二个 PWM 信号产生。相关专用功能寄存器和输出波形之间的关系如下图所示。当工作在高精度模式时，双 PWM 模式不能与互补输出模式结合使用。

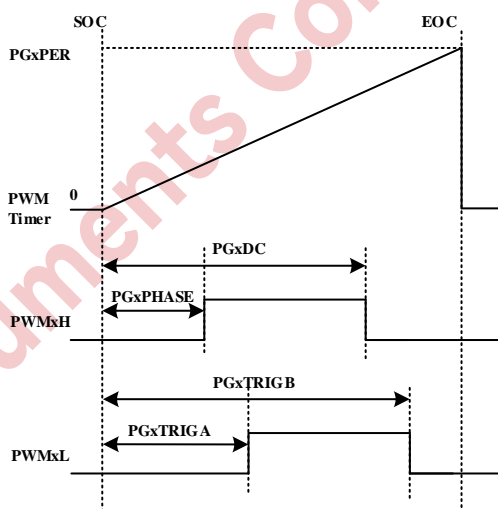


图 171 双 PWM 模式

在这个模式下，PGxTRIGA 和 PGxTRIGB 事件输出信号仍可继续正常工作，仍可作为其他 PWM 发生器的相位偏移触发、ADC 触发等。如果应用需要一个独立的触发事件，那可采用 PGxTRIGC 实现。有关 ADC 触发器的更多信号，详见章节《ADC 触发器》。

由于 PWMxH 和 PWMxL 引脚产生的 PWM 信号源自于同一个 PWM 发生器，它们将同样地受到任何激活的 PWM 控制输入 (PCI) 信号的影响，PWMxH 和 PWMxL 引脚将被改变并驱动到 PGxIOCON 寄存器定义的状态。因此，如果要使用 PCI 信号，那两个 PWM 输出用于的相关的应用功能也是很重要的。

中心对齐 PWM 模式

中心对齐 PWM 模式信号避免了不用占空比的情况下 PWM 发生器之间上升沿或下降沿的重合，减少了逆变器应用中过大的电流纹波和滤波要求。

PWM 脉冲在第一个定时器周期的结束和第二个周期的开始周围保持对称性。如果增加 PWM 信号的占空比，那么上升沿和下降沿的位置将发生变化，从而保持其对称性。当 $MODSEL=100$ 时，中心对齐 PWM 模式被选中，该模式使用两个定时器周期产生一个周期脉冲。PWM 信号特性由这两个专用功能寄存器所决定：

- PGxDC：从两个定时器周期的中心确定 PWM 脉冲宽度
- PGxPER：PWM 定时器计数周期结束点

PWM 脉冲的下降沿发生在 PWM 发生器内计数值等于 PGxDC 的时刻，而上升沿发生计数值等于 $PGxPER - PGxDC + 1$ 的时刻。增加偏移量 1 至上升沿计算中，以保证两个计数周期循环的对称性。例如，PGxDC 的值为 1 将产生持续时间为两个周期的脉冲。

基于 CAHALF 状态位 (PGxSTAT[1]) 跟踪计数器周期，状态位在周期的前半部分读作 0，在后半部分读作 1。发生器仅允许在第一个定时器开始对占空比或周期缓冲区更新，周期结束 (EOC) 中断只有在两个周期完成后才会产生。相关专用功能寄存器和输出波形之间的关系如下图所示。有关数据缓冲的更多信息，详见章节《数据缓冲》。

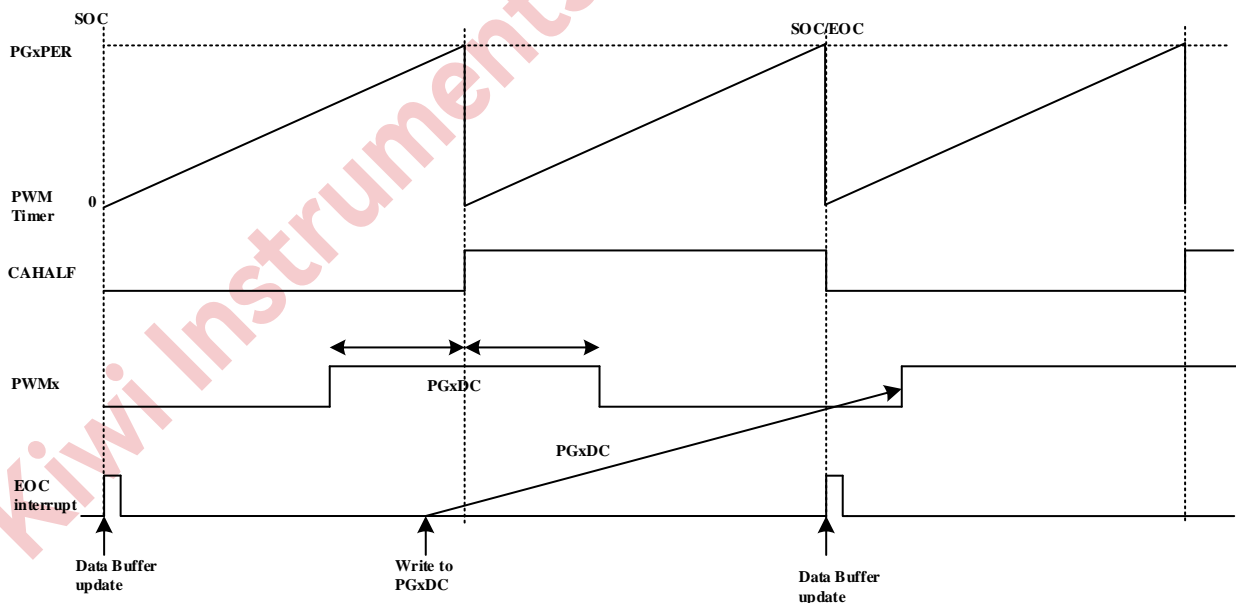


图 172 中心对齐 PWM 模式

双更新中心对齐 PWM 模式

双更新中心对齐 PWM 模式的工作原理与中心对齐模式相同，除了每个 PWM 周期内都会生成两次的 EOC 中断和两次数据缓存更新。当用户希望减少控制回路响应的延迟时，此模式非常有用。注意这种模式会消除中心对齐 PWM 模式脉冲自有的对称性，因为脉冲的上升沿和下降沿可以独立控制。当 MODSEL=101 时，模式选择双更新中心对齐 PWM 模式。相关专用功能寄存器和输出波形之间的关系如下图所示。

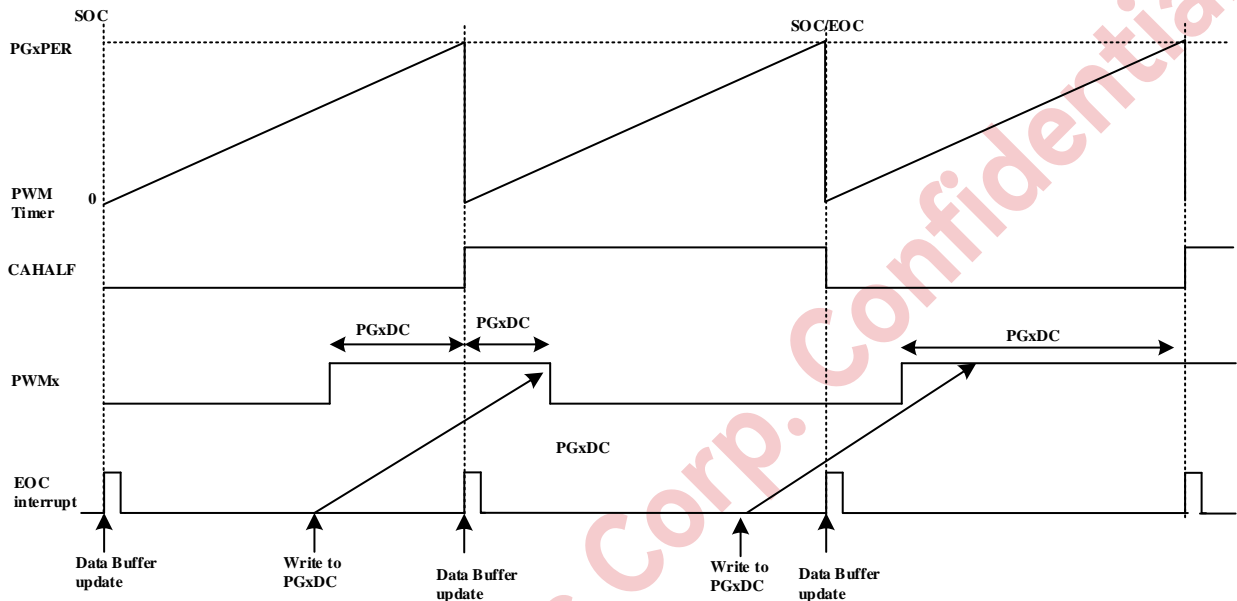


图 173 双更新中心对齐 PWM 模式

双边沿中心对齐 PWM 模式

双边沿中心对齐 PWM 模式的工作原理与双更新中心对齐 PWM 模式相同，但允许上升沿和下降沿的位置通过单独的数据寄存器进行控制。这种模式为用户提供了调整中心对齐脉冲的最大灵活性，而且降低中断事件频率。显然，这将消除中心对齐 PWM 脉冲的对称特性，除非 PGxPHASE=PGxDC。

- PGxPHASE: 决定 PWM 脉冲上升沿位置（相对于两个定时器周期的中心）
- PGxDC: 决定 PWM 脉冲下降沿位置（相对于两个定时器周期的中心）

在双边沿中心对齐 PWM 模式下，单数据缓存更新和双数据缓存更新都是可用的。当 MODSEL=110，此时选择单更新模式，而当 MODSEL=111，则选择双更新模式。在单更新模式下，用户可以在周期内的任何时候写入新的 PGxPHASE 和 PGxDC 值，以便在下一个中心对齐周期中使用。而在双更新模式下，中断事件和数据缓存更新在每个计时器周期发生。这为用户软件提供了修改下降沿事件的 PGxDC 值和上升沿事件的 PGxPHASE 值的机会。用户软件必须检查 CAHALF 位的状态，以确定要更新的寄存器。如果 CAHALF=0（中心对齐周期的前半部分），用户软件应该写入 PGxDC 寄存器。如果 CAHALF=1（中心对齐周期的后半部分），用户软件应该写入 PGxPHASE 寄存器。相关专用功能寄存器和输出波形之间的关系如下图所示。

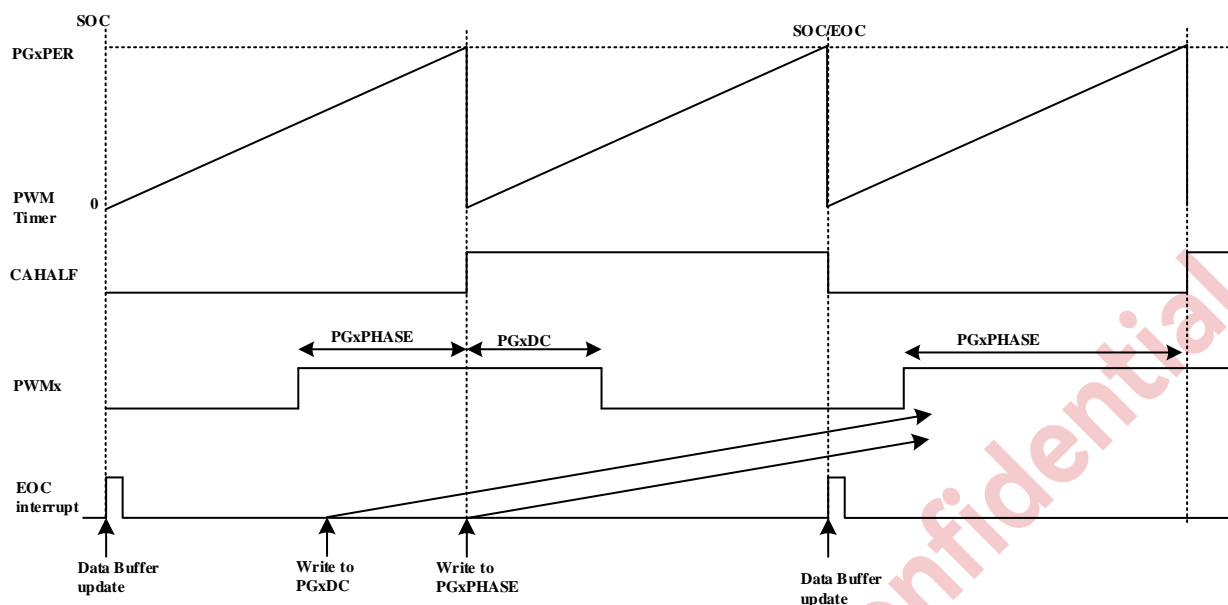


图 174 双边沿中心对齐 PWM 模式 (MODSEL=110)

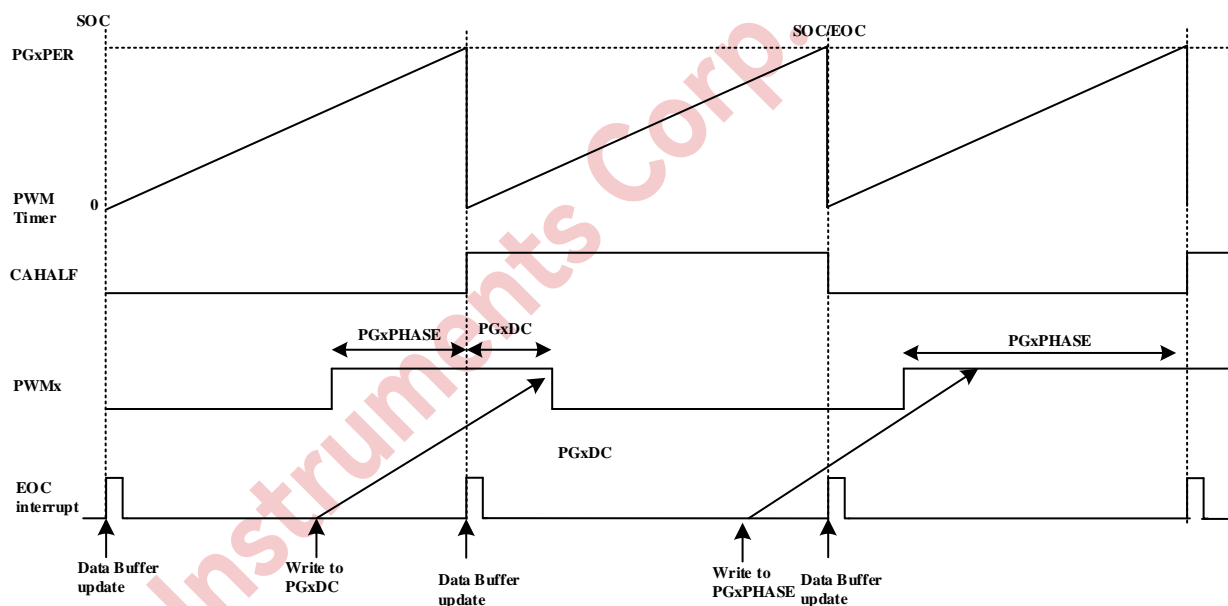


图 175 双边沿中心对齐 PWM 模式 (MODSEL=111)

17.4.4.3 输出模式

每个 PWM 发生器可以编程为三种输出模式之一以控制 PWMxH 和 PWMxL 引脚的行为。输出模式的选择与 PWM 模式相互独立。输出模式包括以下三种：

- 互补输出模式（默认）
- 独立输出模式
- 推挽输出模式

互补输出模式

在互补输出模式下，PWMxH 和 PWMxL 信号永远不会同时激活，可在两个信号之间插入死区切换延迟，死区时间长短可以通过 PGxDT 设置。

当 PMOD=00 时，选择互补输出模式。有关死区的更多信息，详见章节《死区时间》。

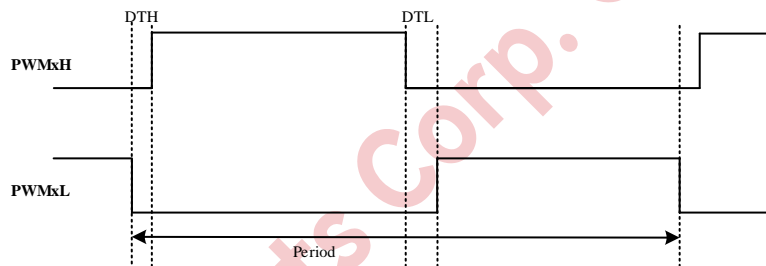


图 176 PWMxH/PWMxL 的上升沿和下降沿（带死区时间）

互补模式下输出覆盖行为：

PWMxH 和 PWMxL 输出可以由外部硬件信号或软件行为进行覆盖控制。输出引脚是被限制放在违反互补输出关系或违反死区插入的状态。由于硬件事件，输出引脚可能立刻被驱动为非活跃，然而在预设的死区时间结束之前，引脚状态不会被激活。有以下硬件和软件覆盖状态是可编程的：

- PCI 故障事件，FLT DAT[1:0]
- PCI 限流事件，CLD AT[1:0]
- PCI 前馈事件，FFD AT[1:0]
- 调试器停止，DBD AT[1:0]
- 软件覆盖，OVRENH/L, OVRD AT[1:0]
- 交换 PWMxH 和 PWMxL 引脚，SWAP

互补模式下覆盖行为的信号链如下图所示。从左往右，SWAP 控件率先应用，因此也可被其他控件所覆盖。随后是驱动引脚活跃的请求逻辑，接着是死区时间发生器。死区发生器是驱动引脚非活跃请求逻辑。这种安排允许非活跃请求优先于 SWAP 和活跃请求。最后，是极性控制逻辑应用于引脚。

PCI 覆盖操作存在优先处理机制，详见章节《输出控制 PCI 模块》。

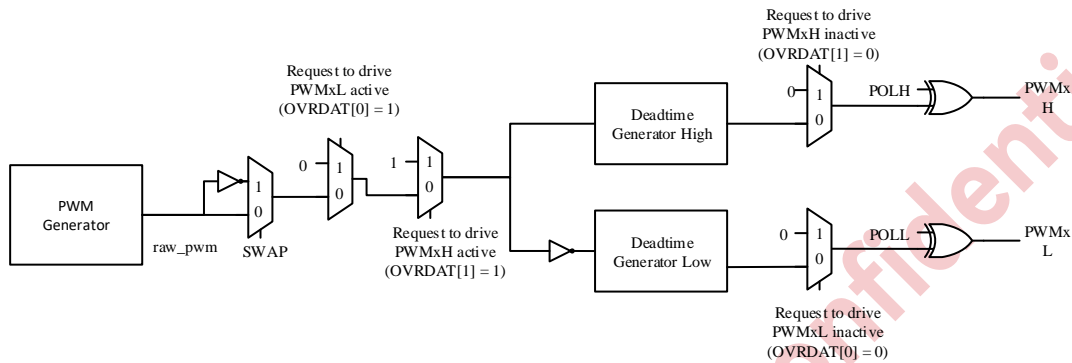


图 177 互补模式下覆盖和交换信号流

表 272 互补模式下覆盖行为

Source	SWAP	OVRENH	OVRENL	OVRDAT [1:0]	FFDAT [1:0]	CLDAT [1:0]	FLTDAT [1:0]	DBGDAT [1:0]	PWMxH Pin State	PWMxL Pin State	Note
DEBUG	x	x	x	xx	xx	xx	xx	00	Inactive	Inactive	Debug Override
								01	Inactive	Active	
								1x	Active	Inactive	
PCI FLT	x	x	x	xx	xx	xx	xx	00	Inactive	Inactive	Fault Override – Debug Override Must be Inactive
								01	Inactive	Active	
								1x	Active	Inactive	
PCI CL	x	x	x	xx	xx	00	xx	xx	Inactive	Inactive	Current Limit Override – Fault and Debug Overrides Must be Inactive
						01			Inactive	Active	
						1x			Active	Inactive	
PCI FF	x	0	0	xx	xx	00	xx	xx	Inactive	Inactive	Feed-Forward Override – Software, Current Limit, Fault and Debug Overrides Must be Inactive
						01			Inactive	Active	
						1x			Active	Inactive	
Software Override	0	0	1	x0	xx	xx	xx	xx	PWM	Inactive	Software Override – Current Limit, Fault and Debug Overrides Must be Inactive
		1	0	0x					Inactive	~PWM	
		0	0	00					~PWM	PWM	
		0	1	x0					~PWM	Inactive	
	1	0	1	x1					Inactive	Active	
		1	0	1x					Active	Inactive	
		1	1	00					Inactive	Inactive	
		1	1	01					Inactive	Active	
	X	1	1	1x					Active	Inactive	
		1	1	1x					Active	Inactive	
		1	1	1x					Active	Inactive	
		1	1	1x					Active	Inactive	

互补模式下启动时输出行为:

当 PWM 初始化并设置 ON 时, 输出立即进入互补状态。当信号通过 PWM 逻辑传播时存在一个输出延迟, 这导致活跃态出现存在延迟, PWMxL 输出转换到非活跃状态 (引脚为高), 延迟过程持续 4 个 TIMH_CLK 周期 (高精度模式下 8 个周期)。一旦活跃态正式开始, PWMx 引脚行为如上表所示。

独立输出模式

在独立输出模式下, PWM 发生器的输出连接至 PWMxH 和 PWMxL 引脚。在大多数应用场景中, 只启用 PWMxH 或 PWMxL 引脚, 另外一个引脚可以用 GPIO 或者其他外设功能。如果启用了双 PWM 模式, PWM 发生器在 PWMxH 和 PWMxL 上产生独立的脉冲信号, 如章节《双 PWM 模式》所述。在独立输出模式下, 没有死区时间要求, 而且对于 PWMxH 或 PWMxL 引脚的当下状态也没有特别要求, 可由外部硬件信号或软件操作实施覆盖控制。当 PMOD=01, 启用独立输出模式。

推挽输出模式

推挽输出模式类似于独立边沿模式, 但 PWM 周期 (MODSEL 所定义) 每收到 SOC 触发之后便重复两次。EOC 触发事件生成和数据缓存更新一直推迟到第二个 PWM 周期结束才出现。下图展示了推挽模式下调用的第二个周期。

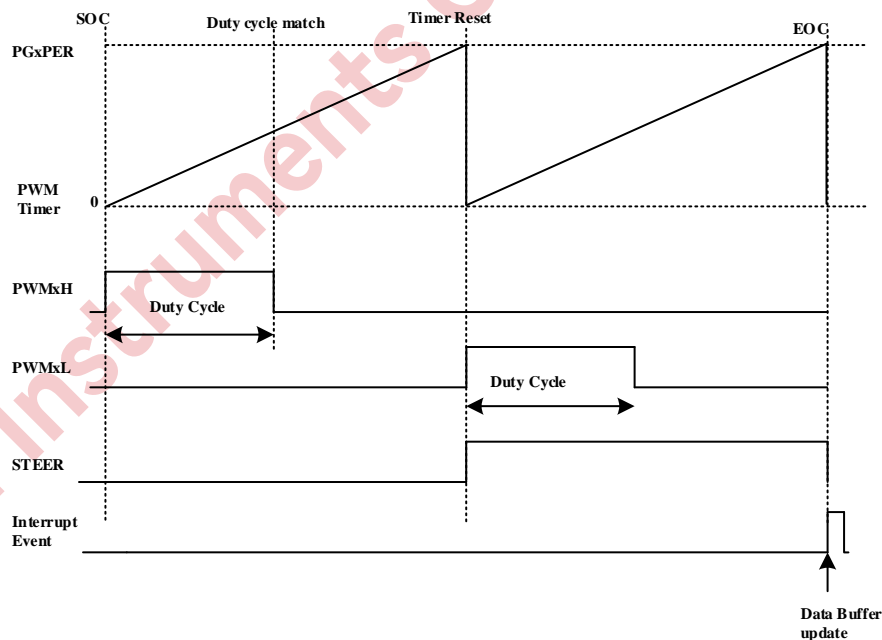


图 178 推挽输出 PWM

在推挽模式下操作 PWM 将使原有的完整周期翻倍, 因为每个推挽周期内存在两次定时器匹配动作。如果 PGxTRIGy 用于事件计时, 则 STEER 可用于门控应用计时。

推挽 PWM 模式通常用于变压器耦合电路，以确保无净直流电流流过变压器。推挽模式确保相同占空比的 PWM 脉冲交替应用到变压器绕组。推挽计数周期的相位可以通过读取 STEER 状态位来确定。如果 STEER=0，则 PWM 发生器正在产生第一个 PWM 脉冲。如果 STEER=1，则 PWM 发生器正在产生第二个 PWM 脉冲。

由于死区时间在推挽模式不可用，但其中的延迟可在推挽输出模式中通过 PGxPHASE 寄存器引入一个小的相移来模拟。类似地，软件中可以限制最大占空比，以避免脉冲结束时距离下一个 PWM 周期的开始太近。

中心对齐模式下推挽操作

当 PWM 发生器以任意一种（总共两种）中心对齐模式运行并选择于推挽模式，那一个完整的 PWM 周期将包括四个时基周期。

高精度模式不应用于中心对齐模式的推挽操作。

工作于中心对齐模式下的推挽输出如下图所示。这种模式组合将 PWM 数据缓存更新和中断事件限制为每四个时基周期内发生一次。因此，在允许改变占空比之前，PWMxH 和 PWMxL 引脚会产生相同的脉冲信号。当选择双边沿中心对齐模式（每个周期更新一次）时，也会发生类似的中断行为。

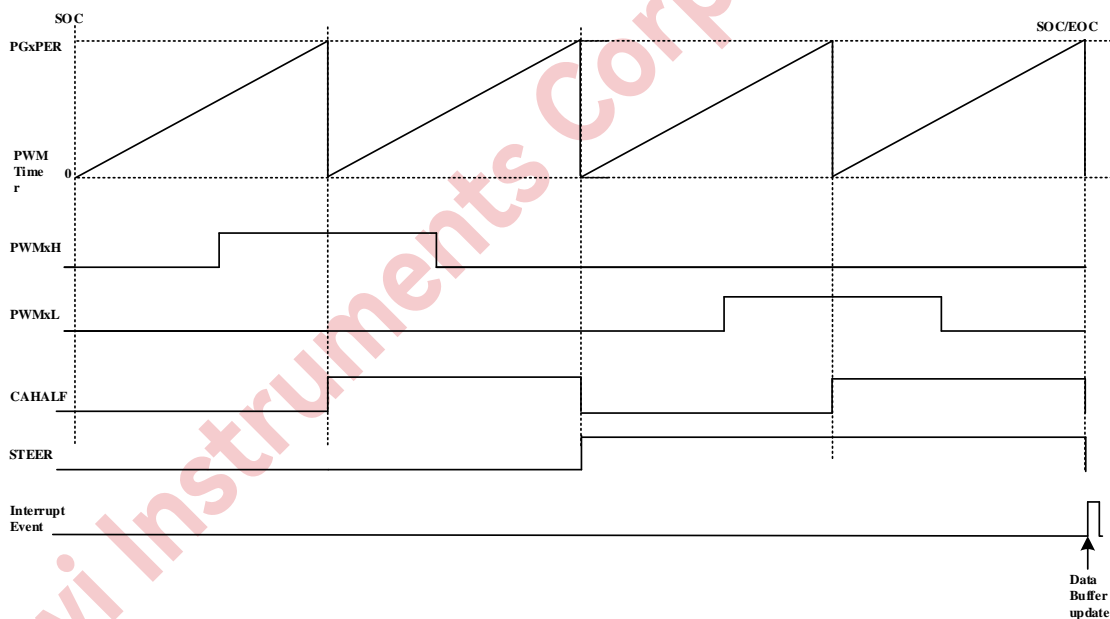


图 179 推挽 PWM：中心对齐模式，双边沿中心对齐模式 (MODSEL=110)

下图显示了双边沿中心对齐 PWM 模式（双更新）或双更新中心对齐模式推挽输出时的运行情况。这种组合模式允许在每个时基周期内进行数据缓存更新和中断事件。这种配置不试图保持输出波形上脉冲的对称性，这也是许多推挽应用程序的要求。用户软件可以在每个边沿事件之后改变中心对齐脉冲的跳变时间，这将控制回路延迟最小化。

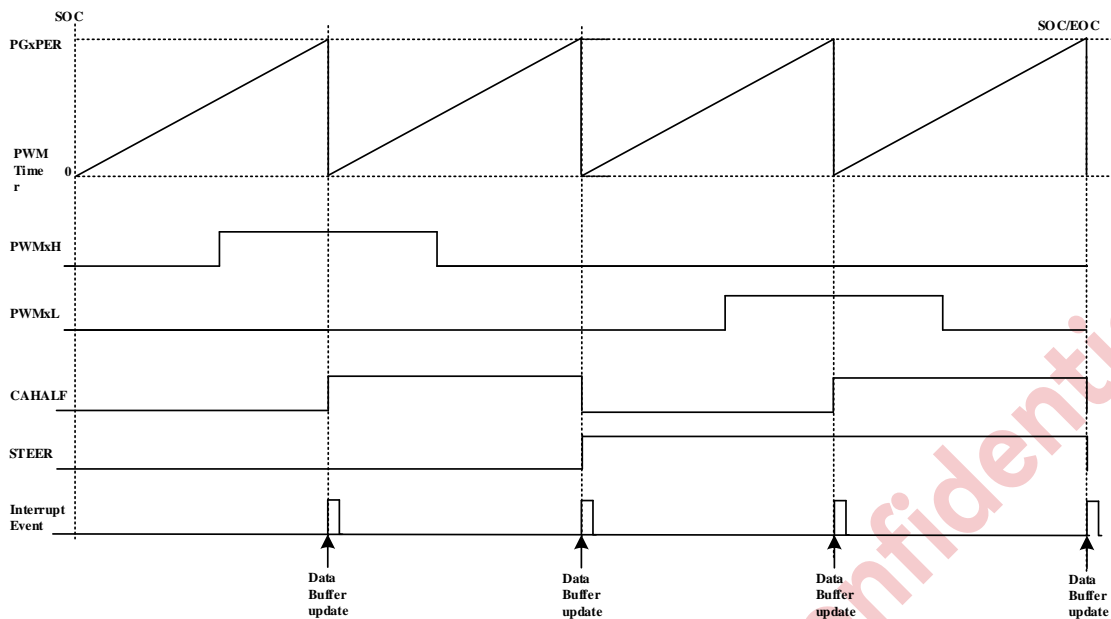


图 180 推挽 PWM：双更新中心对齐模式，双边沿中心对齐模式 (MODSEL=110)

推挽模式和独立模式下输出覆盖

当工作于推挽模式或独立输出模式时，PWMxH 和 PWMxL 信号没有强制的互补关系要求。可通过软件或硬件 (PCI) 覆盖机制将两个引脚都驱动到活跃状态。这种输出状态可能是理想的，也可能不是理想的，取决于由 PWM 发生器控制的外部电路。因此，必须小心选择引脚覆盖值。很多推挽应用程序要求在 PWMxH 和 PWMxL 输出相等的脉冲，以避免直流分量影响。如果应用程序对此敏感，则需在发生两个完整计数器周期后执行软件重写覆盖。硬件 PCI 覆盖应该配置为推挽序列中的两个计数器周期发生后生效。这可通过使用 STEER 信号来完成，配置该信号输出至外部引脚，同时也配置为 PCI 输入。

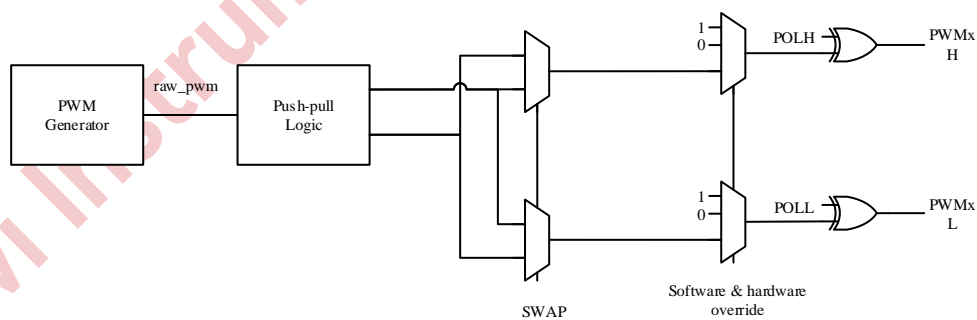


图 181 推挽模式下覆盖和 SWAP 信号流

表 273 推挽和独立模式下覆盖和 SWAP 行为

Source	SWAP	OVRENH	OVRENL	OVRDAT [1:0]	FFDAT [1:0]	CLDAT [1:0]	FLTDAT [1:0]	DBGDAT [1:0]	PWMxH Pin State	PWMxL Pin State	Note
DEBUG	x	x	x	xx	xx	xx	xx	00	Inactive	Inactive	Debug Override
								01	Inactive	Active	
								10	Active	Inactive	
								11	Active	Active	
PCI FLT	x	x	x	xx	xx	xx	00	xx	Inactive	Inactive	Fault Override - Debug Override Must be Inactive
							01		Inactive	Active	
							10		Active	Inactive	
							11		Active	Active	
PCI CL	x	x	x	xx	xx	00	xx	xx	Inactive	Inactive	Current Limit Override - Fault and Debug Overrides Must be Inactive
						01			Inactive	Active	
						10			Active	Inactive	
						11			Active	Active	
PCI FF	x	0	0	xx	00	xx	xx	xx	Inactive	Inactive	Feed- Forward Override - Software, Current Limit, Fault and Debug Overrides Must be Inactive
					01				Inactive	Active	
					10				Active	Inactive	
					11				Active	Active	

表 274 推挽和独立模式下覆盖和 SWAP 行为 (续)

Source	SWAP	OVRENH	OVRENH	OVRDAT [1:0]	FFDAT [1:0]	CLDAT [1:0]	FLTDAT [1:0]	DBGDAT [1:0]	PWMxH Pin State	PWMxL Pin State	Note
Software Override	0	0	1	x0	xx	xx	xx	xx	PWMH	Inactive	Software Override – Current Limit, Fault and Debug Overrides Must be Inactive
		0	1	x1					PWMH	Active	
		1	0	0X					Inactive	PWML	
		1	0	1X					Active	PWML	
	1	0	1	x0					PWML	Inactive	
		0	1	x1					PWML	Active	
		1	0	0X					Inactive	PWMH	
		1	0	1X					Active	PWMH	
	X	1	1	00					Inactive	Inactive	
				01					Inactive	Active	
				10					Active	Inactive	
				11					Active	Active	

17.4.4.4 PWM 产生器触发

每个 PWM 发生器必须接收到一个周期开始 (SOC) 触发信号以开始一个 PWM 周期。触发信号可以由 PWM 发生器本身 (自触发) 或另一个外部触发源提供。SOC 触发信号可来自以下三个来源:

- SOCS 所选的一个内部源 (所有时钟源相同)
- PCI 同步模块所选的外部源
- 软件触发请求 (写入 TRSET)

任何一个 PWM 发生器都可以充当主机，为其他发生器提供触发信号。由此可实现多种触发配置，包括

- 具有独立周期的多个 PWM 输出（彼此之间无同步）
- 具有同步周期的多个 PWM 输出（同步操作）
- 具有相位偏移关系的多个 PWM 输出（触发协作）

同步操作是通过将一个 PWM 发生器（从属）的 SOCS 位设置为另外一个 PWM 发生器（主机）的触发输出实现的，其中作为主机端的发生器的 PGTRGSEL 应设置为 000，这设置 EOC 作为从机端的触发信号。当采用 PGxTRIGy 实现发生器之间的相移时，其中因同步机制而存在的延迟高达 5 个 TIMH_CLK 时钟周期。如果 TRIG 的值为 0，仍然会有一个偏移量，应用中可能需要对 TRIGy 值进行补偿。

触发操作以类似的方式实现，设置主机端的 PGTRGSEL 选择 PGxTRIGy 其中之一作为触发。主机端 TRIGy 寄存器值表示了从机相对于主机的偏移量。

SOC 控制位有两个特殊的选择。当 SOCS=0000 时，PWM 发生器内部触发。当 SOCS=1111 时，没有选择触发源，但可以支持软件使用 TRSET 触发，或源自 PCI 同步模块输入。在这种模式下，直到收到另外一个触发才回开始下一个 PWM 周期。PCI 同步模块的可用源包括外部信号，如比较器事件、设备 I/O 引脚等。PCI 同步模块的一个重要功能就是将外部输入信号同步到发生器的时钟域。详见章节《PWM 控制输入 (PCI) 逻辑模块》获取更多 PCI 相关信息。只要启用了该模块，PCI 同步输入信号会与其他 SOC 或操作作为发生器最终的 SOC 信号。其他 PWM 发生器的输出触发信号也可用作 SOC 事件，配置选项请参考章节《事件选择块》。

触发操作

PWM 周期生成只有在接收到 SOC 触发事件才开始。当时基计数结束时，PWM 周期完成，PWM 发生器退出触发状态。发生器必须重新触发才能继续操作，这可通过以下几种方式实现：

- PWM 发生器是自触发的 (SOCS=0000，默认)
- 新接收触发脉冲，该脉冲与 PWM 周期结束事件一致。

TRIG 状态位表示 PWM 发生器是否处于触发状态。

EOC 信号是 SOC 触发多路复用器的默认选择，也就是允许自触发。当 PWM 发生器完成一个 PWM 周期时就会产生 EOC 触发信号。当与发生器相关联的 ON 位被设置时，EOC 也会产生。这允许所有接收 EOC 信号的 PWM 发生器在主机 PWM 发生器的 ON 位被设置时一致启动，而其他从属发生器的 ON 位需要预先设置，以实现同步启动。

触发模式

PWM 发生器提供了两种触发模式以决定 SOC 触发信号的使用方式。触发模式包括：

- 单触发模式（默认）
- 重触发模式

触发模式的选择由 TRGMOD 控制实现。

单触发模式

单触发模式用于当一个 PWM 发生器与另一个 PWM 发生器同时启动（或有一定时间偏移）。这种模式对于创建单个 PWM 脉冲或者基于外部事件创建单个延迟也很用。如果一个计数周期正在进行中，任何输入的 SOC 触发脉冲将被忽略。在另一个 SOC 触发重新启动发生器之前，要求整个计数周期必须完成。当 TRGMOD=0，选中单触发模式。

重触发模式

重触发模式不同于单触发模式，PWM 周期可以在正在进行的周期结束前重新启动。这样做的话，当接收到新的触发和当前 PWM 周期停止时，计数器将被重置。当 PWM 发生器与从不同时钟源操作的芯片外源同步时，这种模式特别有用。当 TRGMOD=1 时，选择重触发模式。加以配置 TRGCNT 可产生多周期 PWM 事件。

突发模式

在一些应用中，PWM 发生器被触发之后，希望产生的 PWM 周期可以重复一定的次数。TRGCNT 位则是用于配置触发事件后 PWM 周期重复的次数。如果发生器工作于单触发模式下，那么在所有的 PWM 周期完成之前，所有传入触发信号都被忽略。然而工作于重触发模式时，任何一个触发都将启动一个新的 PWM 周期并重置内部周期计数器。

中心对齐模式下触发寄存器

当使用任何中心对齐模式时，PGxTRIGy 寄存器的第 31 位用于指示比较事件产生是发生在第一阶段 (CAHALF=0) 还是第二阶段 (CAHALF=1)。在某些情况下，希望在两个计数阶段都发生一个触发事件，可通过启用两个触发寄存器来完成。

PWM 发生器输出信号跨 PWM 周期边界行为

在正常运行期间，将会设置 PWM 数据寄存器以创建一个在单个 PWM 周期内开始和结束的 PWM 脉冲。更改数据寄存器可能导致 100% 占空比或产生跨越 PWM 周期的活跃状态输出。PWM 发生器必须保持在连续触发状态，以便 PWM 输出在整个 PWM 周期保持活跃。为了保持触发，PWM 发生器的触发输入信号须与 EOC 输出信号保持一致。在以下情况会保持连续触发：

- PWM 发生器是自触发的 (SOCS=0000, 默认)
- 本地 PGxPER 的值与提供触发信号的其他发生器的 PGxPER 值是一样的。

如果 PWM 发生器的触发输入信号没有出现在 EOC 出现处或之前, 那么发生器将退出触发状态, 并且输出将被驱动为非活跃态。

17.4.4.5 PWM 控制输入 (PCI)逻辑块

PWM 控制输入 (PCI) 逻辑块是结构灵活的状态机, 可用于各种各样的应用目的。PCI 块控制输入信号并提供信号用于触发、门控和覆盖 PWM 输出。PCI 还允许将 PWM 发生器输入连接到另一个或外部输入信号。PCI 可在纯硬件上实现输出控制和触发算法, 无需软件参与。每个 PWM 发生器配备四个相同的 PCI 块。包括:

- 故障
- 限流
- 前馈
- 同步

PCI 块的命名不限制它们的使用, 只是被赋予唯一的名称来指定优先级别。同步 PCI 块用于触发, 特别来源是外部事件, 也包括其他 PWM 发生器。故障 PCI 块、限流 PCI 块和前馈 PCI 块是依据外部信号和其他外设来控制 PWM 输出。PWM 引脚的输出状态可以独立配置为每个 PCI 块的预定义状态, 如果有多个 PCI 块请求对 PWM 输出进行控制, 则在优先处理逻辑中进行仲裁。每个 PCI 块都有自己的控制寄存器 PGxyPCI (y=F、CL、FF 或 S), 其中包含与其操作相关的控制位。PCI 逻辑块中有三个主要组件用于构建逻辑功能:

- 输入
 - PCI 源
 - PCI 源界定符, 用于门控 PCI 源信号
 - 终止事件, 用于停止 PCI_active 输出信号
 - 终止界定符, 用于门控终止事件
- 接收逻辑
- 输出和旁路功能

可用的 PCI 源信号及映射关系是与设备相关的。请参照具体的设备数据表了解。典型的信号可能包括:

- PWM 发生器输出

- 组合触发
- ADC
- 模拟比较器 (CMP)
- TIM 事件输出
- 外部输入

PCI 块的输出 (PCI_active) 可用于 PWM 输出逻辑和其他 PWM 发生器。每个 PCI 块输出信号的状态都可在 PGxSTAT 寄存器中获得，包括当前状态和锁存状态。PCI 模块还可以生成中断，相关详细信息请参照章节《事件中断》。PCI 功能框图如下两图所示。

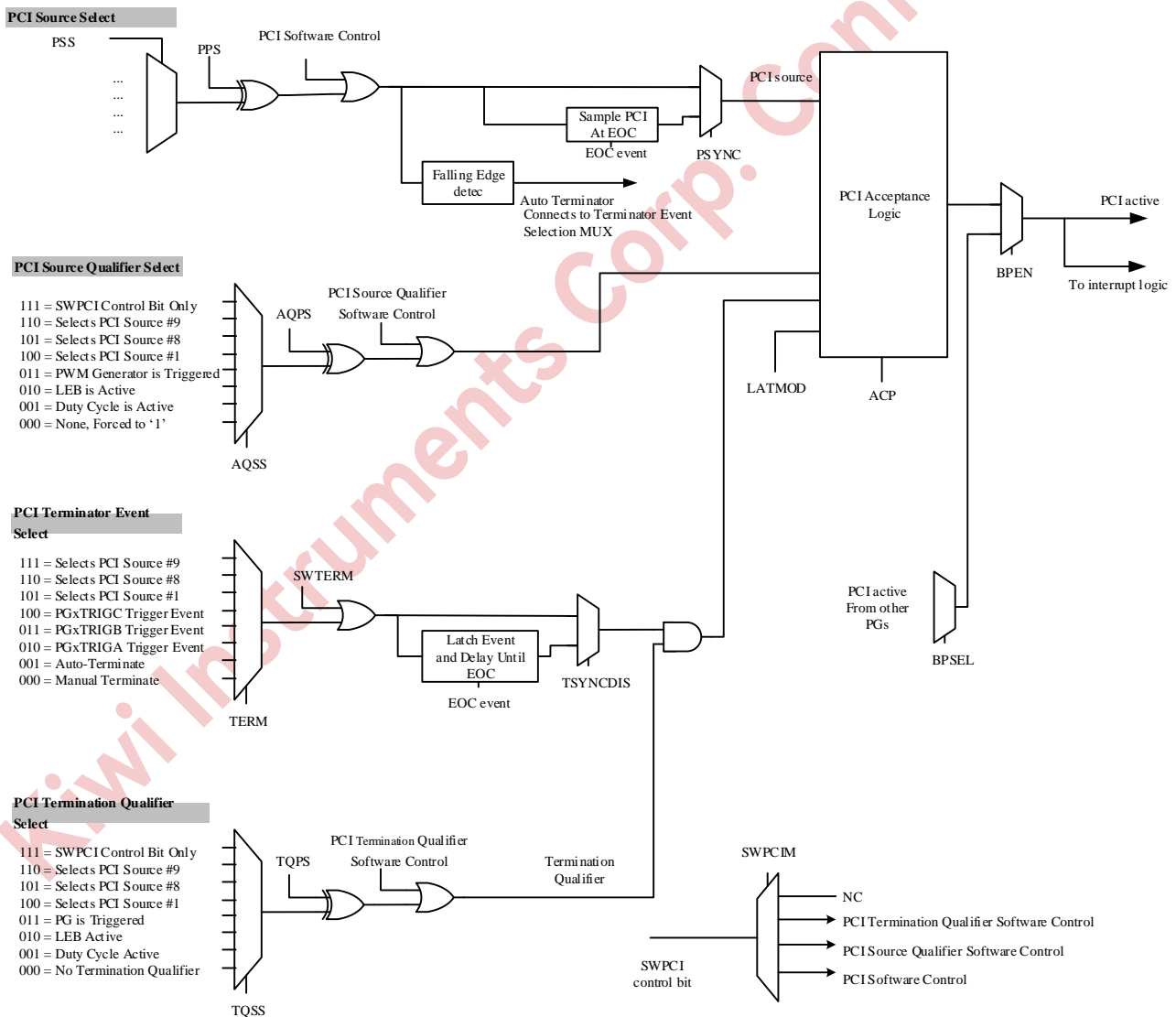


图 182 PCI 功能模块框图

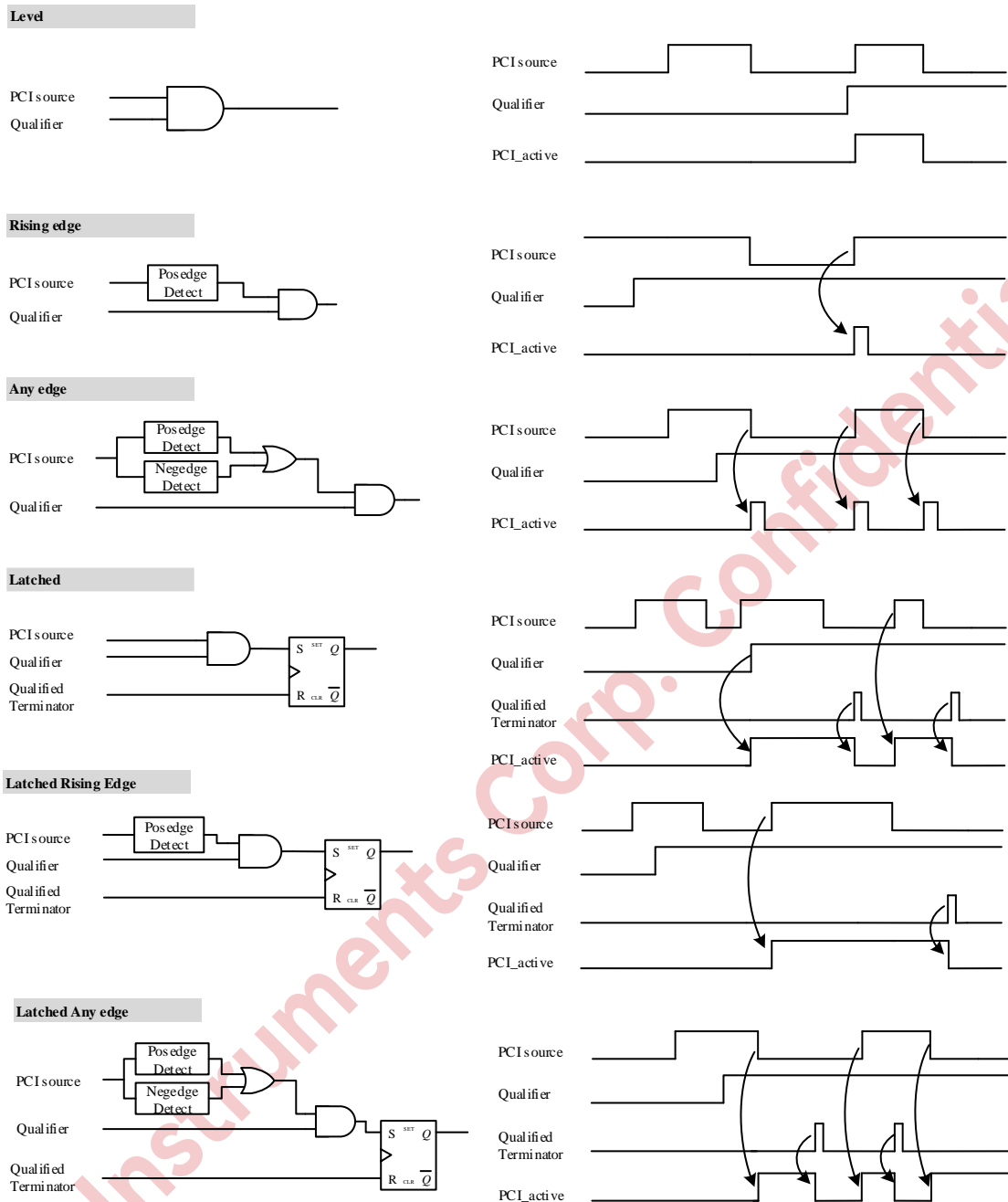


图 183 PCI 接收模式

同步 PCI

同步 PCI 块的主要用途是触发和同步外部事件到 PWM 时钟域，但也导入一个 PWM 时钟的延迟。同步 PCI 块是唯一可以启动周期开始的 PCI 块，可作为 SOCS 所控制的多路复用器的输入。只有同步 PCI 块和前馈 PCI 块可以触发改变死区时间（占空比调整），详情参照章节《死区补偿》。

输出控制 PCI 块

有三种输出控制类型 PCI 块可用于将 PWM 输出置于预设状态。其优先级如下排序：

1. 故障
2. 限流
3. 前馈

故障 PCI 块具有 PCI 块中的最高优先级，它将指定使用该块时输出的状态。故障状态通常认为是灾难性的，往往由软件清除状态。

限流 PCI 块旨在与电流限制传感电路一起使用，用于保护或用作控制回路。前沿消隐 (LEB) 在电流传感应用中通常用于忽略开关瞬态。

前馈 PCI 块用于作为电源应用的控制回路。如果传感电路检测到负载条件的快速变化，系统可配置为立即采取行动，而不必等待下一个 PWM 周期作出反应。

一旦 PCI_active 信号被断言，存储在 xDAT (x=FLT, CL 或 FF) 中的值将立刻应用到输出引脚。xDAT 位位于 PGxIOCON 寄存器中。

PCI 逻辑描述

PCI 块包含三个主要模块以支持广泛的应用程序。首先是输入，基于逻辑选择控制输入信号。其次是接收部分，采用可选逻辑功能应用于输入。最后是包括旁路功能的输出逻辑。

PCI 源：

PCI 源输入是 PCI 块的主要输入，具有以下特点：

- 选择输入多路复用器
- 极性控制
- 软件控制 (软件覆盖)
- 自动终止的边沿检测电路
- 周期终结 (EOC) 同步

使用 PSS 控制位选择 PCI 源。PCI 输入源的极性可以使用 PPS 控制位来选择。所选的 PCI 输入源可以使用 PSYNC 控制位可选地同步到 PWM 周期的末尾。当一个 PCI 信号被用来门控 PWM 脉冲时，这种同步时有用的，因为 PCI 信号可以延迟到下一个 PWM 边界，确保输出端不产生部门脉冲。配备有下降沿检测电路，可用于自动终止由终止事件选择复用器选择的 PCI 活跃信号。

PCI 源界定符：

PCI 源界定符用于界定第二个输入信号。PCI 源界定符与 PCI 接收逻辑中的 PCI 源进行 AND 操作。输入到 PCI 源界定符多路复用器的信号包括：

- PWM 周期信号（活跃部分）
- LEB 有效
- PWM 发生器已触发
- PWMPCI 所选 PWMx 输出
- 外部输入（其他外设或外部引脚）

与 PCI 源输入一样，PCI 源界定符输入具有极性和软件控制。PCI 源界定符用于所有的 PCU 接收逻辑类型。然而，如果不需要，可将 AQSS 设置为 000 以有效禁用该界定符。

PCI 终止事件及界定符：

PCI 终止事件源仅在 PCI 接收逻辑功能的锁存模式中使用，并用于重置锁存器。终止事件的输入包括：

- SWTERM 位
- 触发事件（触发 A、B、C）
- 自动终止（在 PCI 源上检测到的下降沿）
- PWMPCI 所选 PWMx 输出
- 外部输入（其他外设或外部引脚）

PCI 终止符的默认选择是 SWTERM。SWTERM 置位必须比 EOC 提前两个 PGx_CLK 周期。否则 PCI 终止事件将被延迟到下一个 EOC。PCI 触发选项（触发 A、B 或 C）允许在 PWM 周期的特定时间重置 PCI 逻辑。用户软件必须选择适当的 PGxTRIG 作为触发源。在时使用自动终止模式时，建议选择无终止限定符。默认情况下 EOC 同步是开启的，也可通过设置 TSYNCDIS 禁用。

终止界定符的输入和功能类似于 PCI 源界定符。终止界定符用于创建更高级的终止事件。

使用 PWMx 输出为 PCI 功能输入：

PWMPCI 用于选择 PWM 输出作为 PCI 块输入使用。在一些控制回路中，需要使用一个 PWM 发生器的输出来控制另外一个发生器。所选的 PWMx 输出可作为 PCI 源界定符源、PCI 终止事件源和 PCI 终止界定符源，如下图所示。

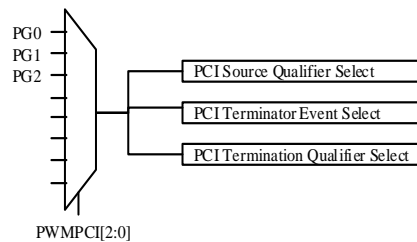


图 184 PCI 的 PWM 源选择

PCI 接收逻辑

PCI 接收逻辑是应用于 PCI 输入的可选逻辑功能。图 18- PCI 接收模式中有 6 种可用逻辑功能：

- 电平模式：PCI 信号直接通过给 PWM 发生器使用。PCI 信号可用接收界定符进行可选界定 (AND 与操作)。
- 上升沿模式：PCI 信号通过上升沿检测电路，产生脉冲事件。PCI 信号可用接收界定符进行可选界定 (AND 与操作)。
- 任意沿模式：PCI 信号通过边沿检测电路，在上升沿或下降沿产生脉冲事件。PCI 信号可用接收界定符进行可选界定 (AND 与操作)。
- 锁存模式：PIC 信号用于设置 SR 锁存。在这种模式下，终止信号和可选终止界定符是用来重置锁存器。异步锁存进入 PCI 活跃状态，但也可能被界定符信号所门控。退出 PCI 活跃状态是取决于终止信号和可能的终止限定符信号决定。从 PCI 活跃状态退出也可以由 PCI 信号本身的退场来界定。
- 上升沿锁存模式：PCI 信号通过上升沿检测电路，并可选界定而产生一个脉冲事件。此脉冲事件用于设置 SR 锁存器。SR 锁存器以类似于锁存模式的方式进行重置。锁存边沿检测模式允许 PCI 在界定符有效后在出现的 PCI 边沿事件后激活。
- 任意沿锁存模式：该模式类似于上升沿锁存模式，上升沿和下降沿都被用来创建设置锁存的脉冲事件。

PCI 逻辑的每种模式都旨在针对特定类型的功率控制功能，尽管这些功能可以应用于各种各样的应用程序。当 PCI 信号用来异步操作 PWM 输出状态时，电平模式是有用的。例如，电平模式可用于允许一个外部的消隐信号迫使 PWM 输出引脚在一段时间内处于特定状态。

边沿事件模式可用于 PCI 块将外部源信号同步到一个 PWM 发生器时基。当 PCI 逻辑用于同步功能，首要关注点是 PCI 信号的上升沿事件，边沿事件导致 PCI 逻辑产生一个内部脉冲可触发 PWM 发生器。

锁存模式对于故障和限流的应用是非常有用的。在这些应用程序中，重要的是 PCI 逻辑在符合条件（界定符）时异步进入活跃状态。PCI 逻辑将一直保持活跃状态，直到选定的终止事件发生为止。通常，终止事件是一个软件动作（手动）或 PWM 周期的结束（自动）。边沿检测锁存模式适用于某些类型的电流控制应用。除非 PCI 输入在一个界定条件满足之后发生翻转，否则 PCI 输出不能切换为活跃态。

锁存模式控制：

默认情况下，锁存接收模式下所用的 SR 锁存器是置位优先的，这可以防止当终止事件信号断言的情况下如果 PCI 信号也是活跃时 SR 锁存器被重置。LATMOD 控制位可用于位某些 PWM 控制功能创建重置优先的 SR 锁存器。当 PCI 逻辑用于处理故障事件时，不建议使用重置优先的 SR 锁存器，因为这可能允许在 PCI 输入信号仍处于活跃状态重置 PCI 逻辑的活动状态。锁存模式示例如下图所示。

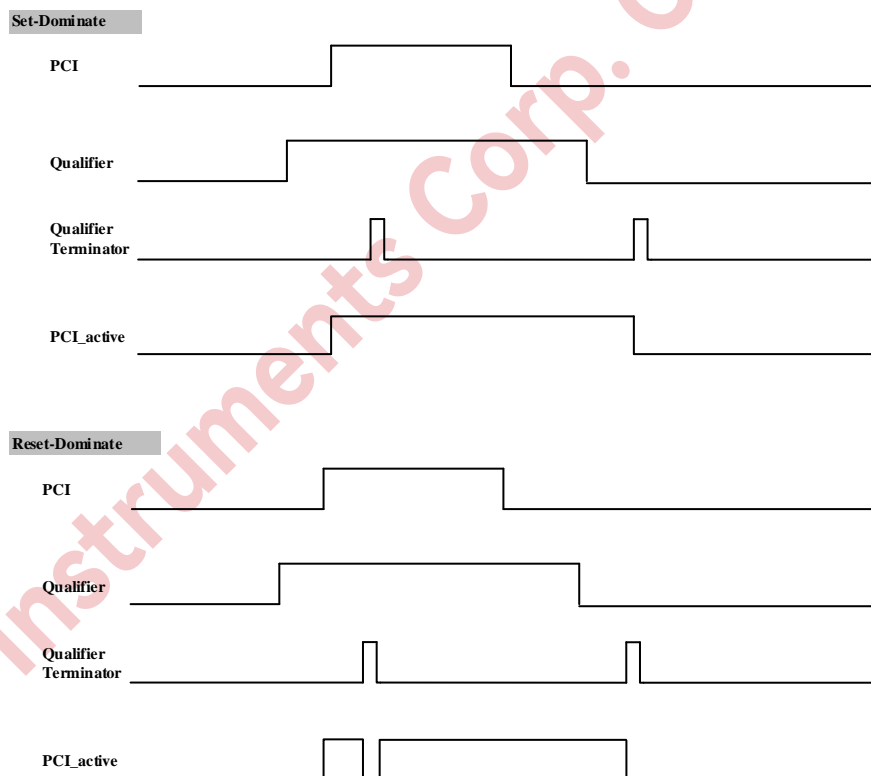


图 185 锁存模式控制

PCI 外部旁路

PCI 外部旁路功能支持使用另外一个 PWM 发生器的 PCI 输出。当辅助、从属或组合 PWM 发生器需要基于主机端 PWM 发生器的 PCI 功能时，此时旁路功能时很有必要的。本地的 PCI 逻辑可以被绕过，而使用来自另一个发生器的 PCI 输出。其中来自其他发生器的 PCI 输入必须是相同类型的 PCI 块（故障、限流、同步或前馈）才是

有用的。当 BPEN=1 时，BPSEL 位指定哪个发生器提供 PCI 控制，但 FLTDAT、CLDAT 和 FFDAT 控制位的覆盖状态不受影响，是由本发生器控制位所决定。

软件 PCI 控制

所有的 PCI 块都支持软件控制以产生和清除事件或调试开发。有三个控件可用于手动控制 PCI 输入：

- SWPCI 控制位
- SWPICM 分复用器 (用于 SWPCI 控制位)
- SWTERM 生成终止事件

SWPCI 控制位可编程为 0 或 1，并传导到三个目的地的其中一个，具体目的地由 SWPICM 指定。如下图所示。

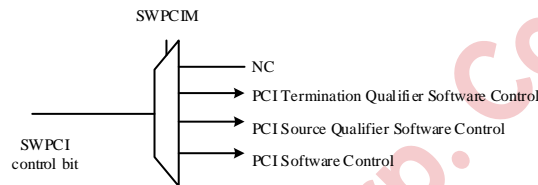


图 186 软件 PCI 控制位分配

SWTERM 位直接与终止事件输入绑定，可以通过向 SWTERM 写 1 并将 TERM 设置为 000 来手动终止 PCI 事件。此外，接收和终止界定符的输入多路复用器具有输出固定为 1 的选项，或者，当其与各自的极性控制一起使用时则输出固定为 0。这些固定状态可用于调试或关闭接收功能。

PCI 源 EOC 同步，电平模式：

当 PCI 接收逻辑运行于电平模式时，而且 PCI 源被同步至 EOC 事件后，此时没有逻辑保存先前的 PCI 源状态。因此，生成 PCI 输出只是同步到 EOC 事件的 PCI 源信号。这种配置对于 PWM 斩波应用非常有用，其中 PCI 源信号被用作门控信号，而门控信号自动对齐到 PWM 周期边界，如下图所示。

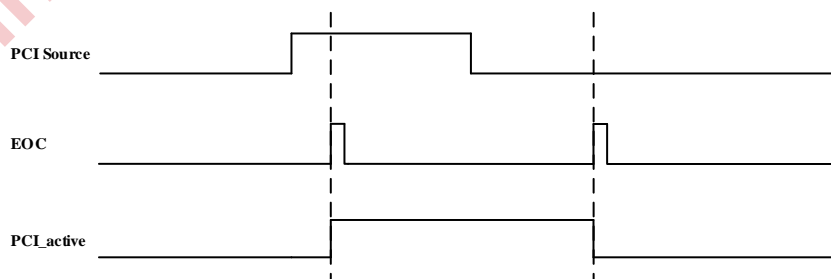


图 187 PCI 源 EOC 同步，电平接收模式

PCI 源 EOC 同步，边沿模式

当 PCI 接收逻辑运行于上升沿模式或任意沿模式而且 PSYNC=1 时，PCI 源被同步至 EOC 事件，如下图所示。如果检测到一个边沿事件，所产生的脉冲会延迟到下一个 EOC 事件到来时输出。在一个单一 PWM 周期内，如果 PCI 源信号经历由非活跃变为活跃，而后再回到非活跃，这种情况下 PCI 信号是不会置为有效。

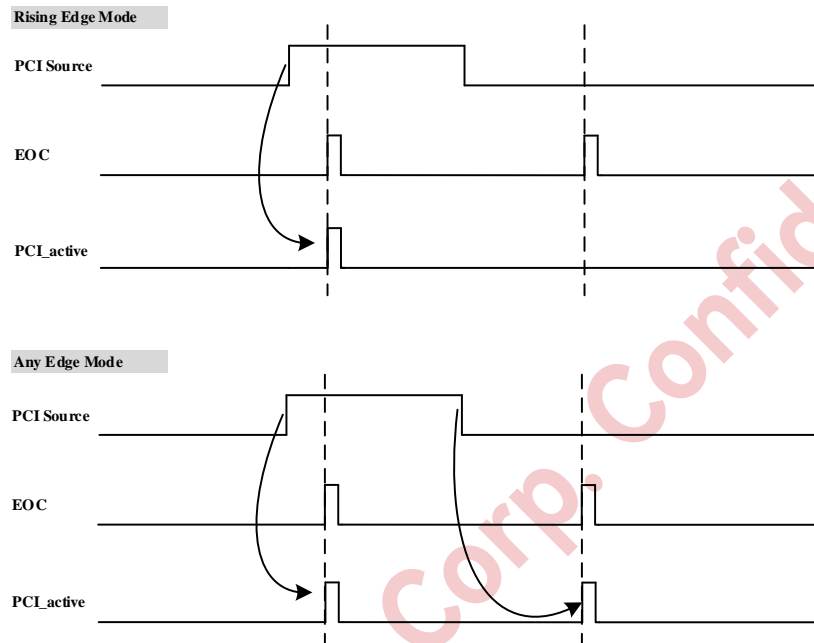


图 188 PCI 源 EOC 同步，边沿接收模式

PCI 源 EOC 同步，锁存模式

当 PCI 接收逻辑运行于锁存模式而且 PSYNC=1 时，PCI 源被同步至 EOC 事件，如下图所示。同步逻辑将 PCI 源信号的上升沿延迟到下一次 EOC 发生。同步逻辑输出在 PIC 源信号下降沿便失效，同步逻辑的输出用于置位 SR 锁存器，在 EOC 周期出现一个 PCI 输入脉冲并不能生效 PCI 活跃信号。这是因为 PCI 输入信号的下降沿会复位 EOC 同步逻辑。

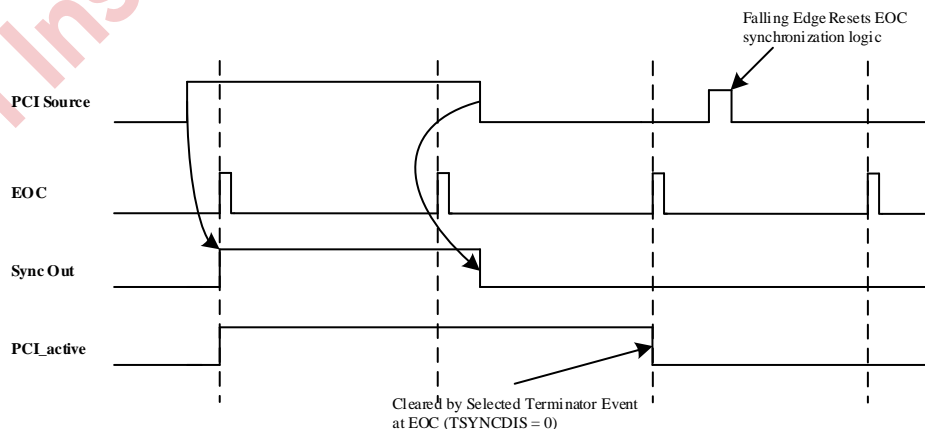


图 189 PCI 源 EOC 同步，锁存接收模式

PCI 源 EOC 同步，边沿锁存模式

当 PCI 接收逻辑运行于边沿锁存模式而且 PSYNC=1 时，PCI 源被同步至 EOC 事件，如下图所示。这种配置的操作类似于上升沿和任意沿模式，除了同步逻辑的事件输出是锁存的之外。在 EOC 周期出现一个 PCI 输入脉冲会激活 PCI 活跃信号。

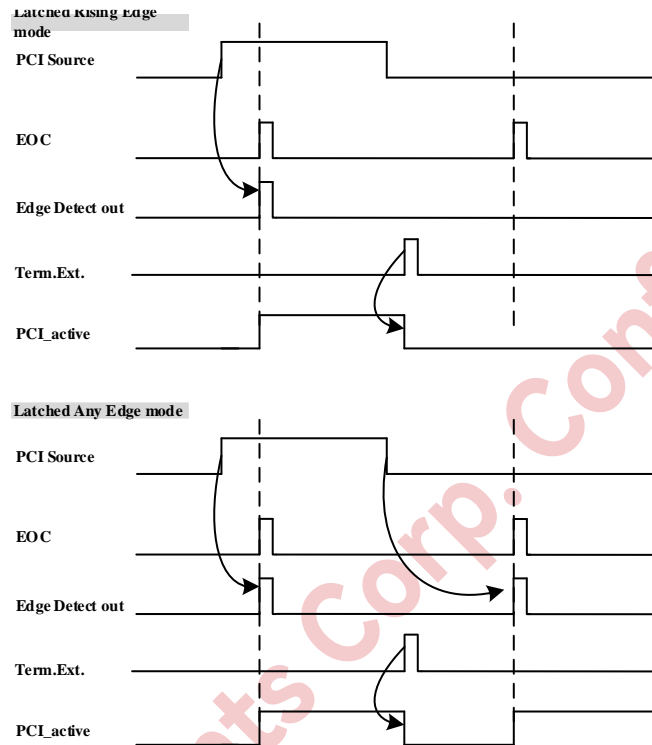


图 190 PCI 源 EOC 同步，边沿锁存接收模式

PCI 终止符 EOC 同步

默认情况下，PCI 逻辑将终止事件同步到 PWM EOC。这样 PWM 可在新周期时干净地恢复。终止信号的上升沿脉冲一直拖延到 EOC 事件出现才生效。终止信号通常是一个脉冲事件，用于重置 PIC 逻辑的锁存状态。如果在 EOC 事件发生之前就接收到短脉冲，则在 EOC 事件出现之时产生复位脉冲。如果终止信号是较长脉冲，则只要终止信号中存在，同步输出就保持活跃状态。必要的话可用此行为强制 PWM 进入复位状态。终止事件同步时序如下图所示。

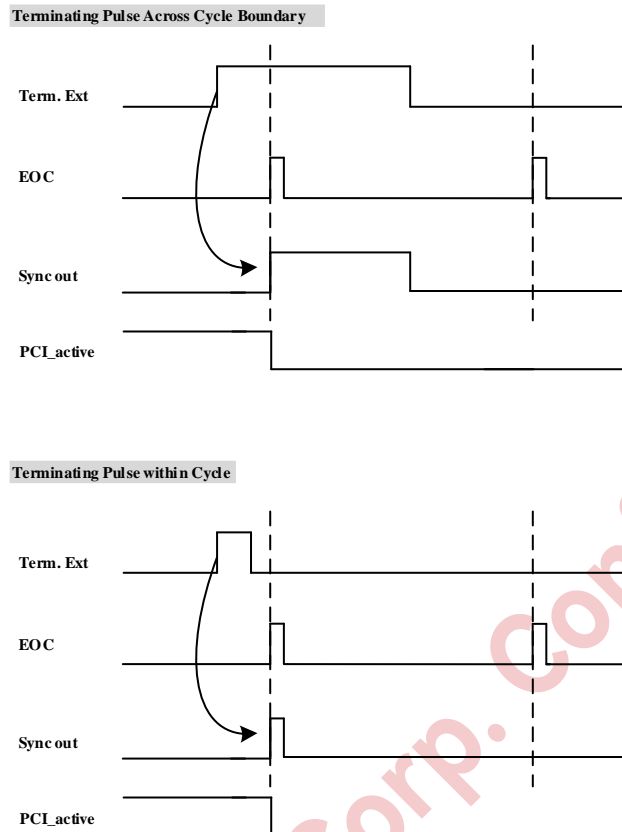


图 191 PCI 终止符 EOC 同步

17.4.4.6 死区时间

死区时间的特性就是用于设定一个时间段，期间两个互补输出都不能同时活动。这用于防止电流桥中的两个输出驱动设备（开关）同时导通，造成过大的电流。但输出开关的开启和关闭时间不是瞬时的，因此设置死区时间以确保只有一个设备处于活跃状态。死区时间是通过推延活跃状态输出来实现的。对于 PWMxH，这将延迟上升沿，而对于 PWMxL 则延迟下降沿。如下图所示。

死区时间是通过 PGxDTH 和 PGxDTL 配置实现，PGxDTH 和 PGxDTL 分别独立应用于 PWMxH 和 PWMxL 输出。控制寄存器的有效位依赖于是否启用高精度模式。在标准精度模式下，高 5 位 (15:11) 不使用。死区时间通常只在互补输出模式下使用。

示例：标准精度下死区时间计算

$$F_{PG_CLK} = 200 \text{ MHz}$$

$$\text{目标死区时间} = 2\mu\text{s}$$

$$PGxDTy = F_{PG_CLK} * \text{死区时间} \quad (y = H \text{ or } L)$$

$$PGxDTy = 200 \text{ MHz} * 2\mu\text{s} = 400 = 0x0190$$

示例：高精度下死区时间计算

$F_{PGx_CLK} = 200 \text{ MHz}$, $T_{PGx_CLK} = 5\text{ns}$, 最小精度 $\Delta = T_{PGx_CLK}/32$

目标死区时间 = $2\mu\text{s}$

$PGxDTy = \text{死区时间}(y = H \text{ or } L)/\Delta$

$PGxDTy = 2\mu\text{s}/(T_{PGx_CLK}/32) = 32 * 2\mu\text{s}/5\text{ns} = 12800 = 0x3200$

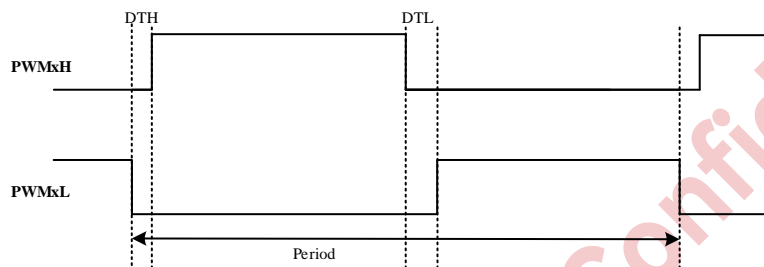


图 192 PWMxH/PWMxL 的上升沿和下降沿（带死区时间）

死区补偿

死区补偿功能允许由 PCI 输入选择性地控制占空比。通过向 PGxDCA (PWM 发生器 x 占空比调节) 寄存器写入非零值, 并设置 PCI 逻辑来控制补偿调节从而使能死区补偿。当 PCI 输入生效时, PGxDCA 的值将被加到 PGxDC 寄存器上生成有效的占空度值。如下图所示。高精度模式下, 低 5 位固定为 0。

DTCMPSEL 控制位选择用于死区补偿的 PCI 逻辑块。该逻辑块可以是前馈或同步 PCI 块。如果 PGxDCA 的值为 0, 则不管 DTCMPSEL 值如何, 死区补偿功能都将被禁用。来自 PCI 逻辑的死区补偿输入信号在 PWM 周期结束时被采样, 并在下一个 PWM 周期中使用。通过 PGxDCA 寄存器对占空比持续事件的修改应发生在占空比的末端。

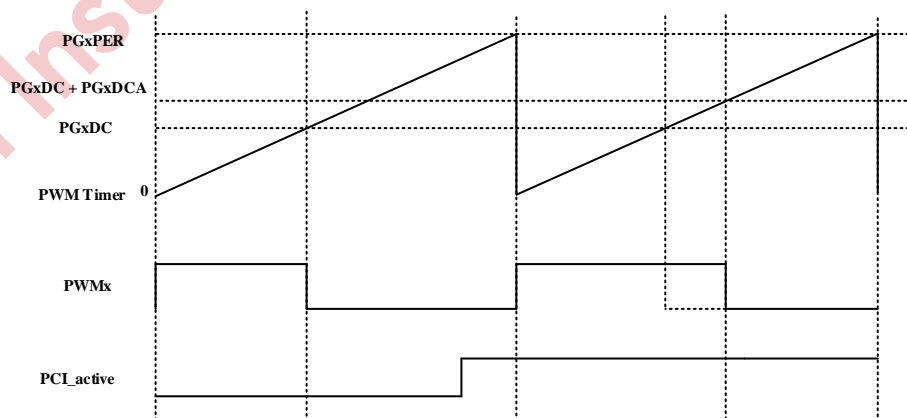


图 193 增加 PGxDCA 到 PGxDC 寄存器

17.4.4.7 前沿消隐

前沿消隐 (LEB) 特性是用于屏蔽可能导致错误故障状态的瞬态。前沿消隐可使用任何 PCI 块实现，基本上是在 PWM 边沿事件之后的指定时间内忽略输入信号。

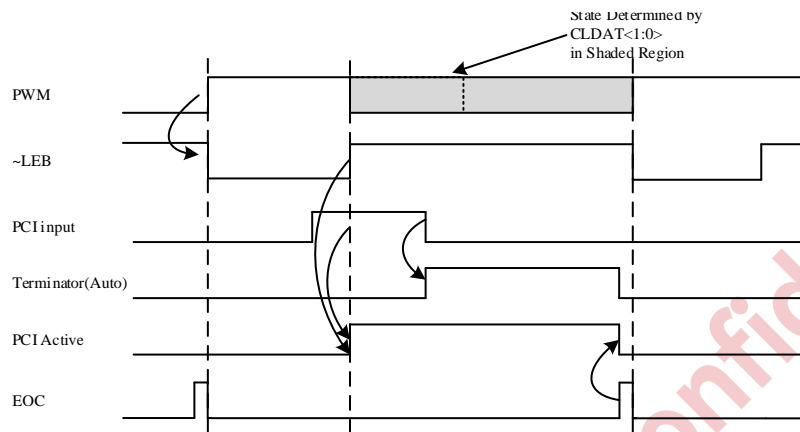


图 194 前沿消隐

PWMxH 和 PWMxL 的上升沿和下降沿都可用以启动 LEB 计时器。LEB 持续时间长度是通过设置 LEBP 实现。支持多个边沿 (PHR, PHR, PLR 或 PLF) 可供选择作为启动沿。但如果出现计时重叠, 则计数器将在每个有效边沿上重置。在大多数应用中, 只需选择 PWM 信号的一个边沿触发 LEB 定时器。

当 PCI 逻辑用于限流时, LEB 计数器通常用于避免错误跳闸。在这种情况下, 可在 PWM 信号的双沿均触发 LEB 计数。PCI 逻辑工作在锁存接收模式下, LEB 有效信号取反用作界定符限定 PCI 输入信号。上图中所示在 PWM 周期中, LEB 尚在计数, PCI 输入转为活跃, 但在 LEB 计数结束之前, 不会出现 PCI 生效事件或输出覆盖。

前沿消隐计数器周期计算

LEB 计数器周期值 (TLEB) 存储在寄存器 LEBP, 低三位只读且总是读回 0, 也就是最小 LEB 分辨率是 8 个 PGx_clk。关于周期计算的公式如下:

公式: 前沿消隐周期

$$T_{LEB} = \frac{8 * (LEBP[15:3] + 1)}{F_{PGx_CLK}}$$

$$LEBP[15:3] = \frac{T_{LEB} * F_{PGx_CLK}}{8} - 1$$

其中, TLEB 的单位为 ms, us 或 ns

示例:

$$F_{PGx_clk} = 200 \text{ MHz}$$

LEB 分辨率 = $8/200 \text{ MHz} = 40 \text{ ns}$ 每级

目标 LEB 时间长度 = TLEB = $1\mu\text{s}$

$$LEBP[15:3] = \frac{1\mu\text{s} * 200\text{Mhz}}{8} - 1 = 24 = 0x18$$

往低三位补 0, LEBP[15:0] = 0xC0

前沿消隐应用时 PCI 配置

LEB 计数器产生 LEB_active 信号,该信号可作为 PCI 块的接收界定符和/或终止界定符多路复用器的输入之一。允许 LEB 定时器用作选择 PCI 源或终止符的门控信号。接收界定符和终止界定符信号的极性可用 PCI 控制位实现翻转,所以“LEB 有效”信号可变成“LEB 无效”信号。当使用 LEB 模块时,建议使用 PCI 锁存接收模式,这样 LEB_active 信号就只会影响到 PCI 活跃状态的进入或退出。自动终止可在故障条件清除之后用于重置系统。

LEB 初始化序列示例:

1. 选择用于 LEB 的 PCI 类型
2. 配置 PSS 选择 PCI 输入。这通常连接到比较器以检测过流。
3. 配置 AQSS 选择 LEB 作为接收界定符
4. 配置 QAPS 设置翻转界定符极性
5. 配置 ACP 选择锁存模式
6. 配置 TERM 选择自动终止方式

17.4.4.8 输出覆盖

覆盖功能可以用来 PWM 输出并将某些条件强加到相应引脚上。用户软件可通过往位于 PGxIOCON 中的 OVRENH 和 OVRENH 位写 1 从而实现覆盖输出状态。当被覆盖时,引脚状态会变成 OVRDAT 的值,但也可能因为与所定输出模式(互补模式下,死区处理)相冲突而不一致。大多数情况是在互补模式,这在章节《互补输出模式》有讨论。

OVRDAT、OVRENH 和 OVRENH 控制位具备双缓存的灵活性。PGxIOCON 寄存器中的 OSYNC 控制位指定何时将用户覆盖值应用到 PWM 输出。手动软件覆盖可应用于以下时间点:

- 新的 PWM 周期开始
- 立即(尽快)
- 由 UPMOD 配置决定

关于 UPMOD 的细节在章节《数据缓存》中讨论。

17.4.4.9 ADC 触发

每个 PWM 发生器都有能力去触发多个 ADC，无论是内部还是外部的设备。ADC 触发器是基于 TRIGA、TRIGB 和 TRIGC 在 PWM 周期内的比较事件所得。触发器也可以用以触发其他外设和功能模块，例如 DAC、中断等。触发信号可通过事件输出模式供外部使用，或输出至 CPU 中断处理模块。

可启用多个 TRIGx 源来创建 ADC 触发输出，其中多个信号通过逻辑或一起。如果启用多个 TRIGx 寄存器来产生 ADC 触发事件，则必须将它们配置为允许到 ADC 的唯一触发事件。

每个 PWM 发生器可以生成两个 ADC 触发信号：ADC 触发 1 和 ADC 触发 2。这两个触发输出对于 SMPS 应用程序是非常有用的。在这种应用程序中，通常需要在在一个周期中测量两个量。每个触发信号连接到单独的 ADC 或 ADC 触发输入。ADC 触发 1 输出有一个额外的偏移和后分频功能，以允许以下功能：

- 后分频，降低 ADC 触发事件的频率。
- 偏移，一次性偏移，这允许经过后分频之后的触发事件与来自其他生成器的触发事件交错。

ADC 触发 2 的触发事件在每个 PWM 周都会产生。ADC 触发 1 输出可以使用 ADTR1PS 控制后分频，以减少 ADC 转换的频率。此外，ADC 触发 1 可抵消一定数量的触发事件（由 ADTR1OFS 控制）后再输出触发信号。这两组控制位允许客户基于多个 PWM 发生器建立一个交错的 ADC 触发事件集。此外，ADC 触发事件可以简单地后分频，以减少 ADC 测量的频率。如果 ADTR1PS 设置为 0，每个 PWM 周期将产生一个 ADC 触发事件；而当 ADTR1PS 不为 0 时，首先第一个触发事件将在 ON 位设置后的 PWM 周期中产生，此后每 N 个周期产生一次。ADTR1OFS 位用于建立 0~31 个比较事件偏移（在设置 ON 位后）。完成偏移量建立之后，后分频器依据 ADT1PS 值决定触发事件的数量。当从多个 PWM 发生器交错触发 ADC 时，所有的 PWM 发生器都应配置为相同的周期以确保触发事件之间的间隔一致。

17.4.4.10 事件选择块

每个 PWM 发生器都有一个用于事件、触发信号和中断的逻辑块。然后这些信号为事件输出模块所用或者作为触发源以开启新的 PWM 周期。事件选择块有三个主要功能：

- ADC 触发配置
- PWM 发生器触发
- 中断

PWM 发生器触发输出

在诸多 PWM 发生器内部事件中，可选其一来驱动触发输出。PWM 发生器触发输出信号是由 PGTRGSEL 控制位所选，选择包括在 EOC 或三个 TRIGx 比较事件之一。使用一个 TRIGX 作为另外一个发生器的 SOC 触发事件对于实现可变相位 PWM 是有用的。两个不同的 PWM 发生器之间的相位关系可以通过改写 TRIGx 值实现。

事件中断

输出至 CPU 的中断是灵活可编程的。IEVTSEL 允许用户选择下面事件之一作为中断输出：

- EOC (默认)
- TRIGA 比较事件
- TRIGB 比较事件
- TRIGC 比较事件
- 触发 1 事件
- 触发 2 事件
- 缓存更新事件
- 无

事件选择块还包含四个 PCI 块的中断使能，分别是 SIEN、FFIEN、CLIEN 和 FLTIEEN。

当 IEVTSEL 设置为禁用时，PCI 中断仍然可以独立使用。

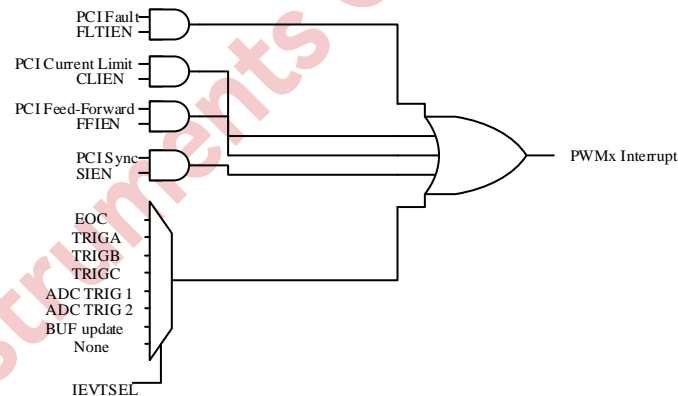


图 195 事件选择模块

17.4.4.11 数据缓存

PWM 模块允许特定专用功能寄存器的数据值被缓存，在后续的事件中应用到 PWM 输出。

有以下用户寄存器和位支持缓存，允许即时修改数据，而发生器正基于上一组设置好并已生效的数据工作。

- PGxPER
- PGxPHASE
- PGxDC
- PGxTRIGA
- PGxTRIGB
- PGxTRIGC
- PGxDTL/ PGxDTH
- SWAP
- OVRDAT (软件输出覆盖值)
- OVRENH/OVRENH (软件输出覆盖使能)

在 PWM 周期开始时，数据从专用功能寄存器 (SFR) 传输到内部 PWM 寄存器。该操作可能是每一个、两个或四个定时器周期执行一次，是取决于 PWM 发生器的工作模式和输出模式。可被要求立即更新寄存器，满足在功率转换器操作中产生立即更改的需求。在其他情况下，当需要多个 PWM 发生器之间数据的一致性时，缓存更新可能被推迟，直到一些外部事件发生。此模块支持用户指定何时将相关的专用功能寄存器传输到真正生效的内部寄存器，可支持的选项有：

- 立即
- 在下一个 PWM 周期开始
- 作为一个更大群体的一部分

PGxCON 寄存器中的 UPDMOD 控制位决定了数据更新的操作模式。PGxSTAT 中的 UPDATE 状态位允许查看何时完成数据寄存器更新和可能用到的更改。当 UPDATE=0 时，用户软件可以写入新的值到数据寄存器，配置完成后并设置 UPDREQ。设置 UPDREQ 是为了向 PWM 发生器提交新的配置值，此时用户软件不能修改数据寄存器，直到该位被硬件清除。

为了避免额外的 CPU 周期开销，数据更新可以配置为写入某个特定的数据寄存器时自动更新。配置 UPDTRG (位于 PGxEVT) 选定特定数据寄存器。

默认情况下，UPDREQ 必须在软件中手动设置。然对 PGxDC 寄存器的吸入可以触发更新，因为许多应用程序需要经常改变 PWM 的占空比。PGxPHASE 和 PGxTRIGA 也可以选作更新触发，在可变相位应用中，这些寄存器时经常修改。如果要更新几个 PWM 数据寄存器，那么选作更新触发的数据寄存器必须是最后一个被写入的寄存器。一旦 UPDATE 被置位，PWM 数据寄存器禁止修改，用户软件必须等待 PWM 硬件清除 UPDATE 位后才能再次修改相应数据寄存器。

同步多路 PWM 发生器的缓存更新

MSTEN 控制位 (PGxCON) 支持 PWM 发生器控制其他发生器中的数据寄存器更新。UPDREQ 控制和 UPDATE 状态位可以有效地广播到其他发生器，以允许一组控制共同功能的 PWM 发生器之间进行相关寄存器更新。当 MSTEN 已置位,用户软件 (或 PWM 发生器硬件本身) 设置 UPDREQ 控制位时,此事件将广播到所有其他 PWM 发生器。如果发生器 (UPDMOD=01x) 接收到请求，接收模块将对其本地 UPREQ 置位。本地寄存器完成更新后，相应的 UPDATE 状态位将被清除。当然用户也可能会手动设置本地 UPREQ 位。

表 275 PWM 数据寄存器更新模式

UPDMOD	模式	描述
000	SOC	当 UPDREQ=1 时，数据寄存器在接下来的 PWM 周期的起始处更新
001	立即	当 UPDREQ=1 时，数据寄存器立即更新。
010	从 SOC	当 UPDREQ=1 时，只有接收到主机端更新请求后，数据寄存器在接下来的 PWM 周期的起始处更新。
011	从立即	当 UPDREQ=1 时，只有接收到主机端更新请求后，数据寄存器立即更新。

更新数据寄存器的目的是因为 PWM 周期长度是可变的。一个 PWM 周期可能包含一个、两个或四个定时器周期，这取决于 PWM 工作模式及其所选的输出模式。当请求 SOC 更新时，PWM 数据寄存器可在下一个、第二个或第四个计时器周期结束更新。下表总结了每次 SOC 更新与 PWM 发生器工作模式和输出模式之间的定时器周期数。有关更新事件时序的附加信息，请参阅有关 PWM 模式的章节。

表 276 定时器周期数 (数据寄存器更新周期内)

PWM 模式	输出模式	定时器周期数 (PWM 周期内)	定时器周期数 (中断和数据更新周期内)
独立边沿 双 PWM 可变相位	独立 互补	1	1
独立边沿 双 PWM 可变相位	推挽	2	2
中心对齐	独立 互补	2	2

PWM 模式	输出模式	定时器周期数 (PWM 周期内)	定时器周期数 (中断和数据更新周期内)
中心对齐	推挽	4	4

立即更新

当使用即时更新模式时，从命令变更到真正实施之间可能会有延迟。该模式尽可能快地完成真正使能生效以防止出现意外结果。

周期值的立即更新将更新到 PGxPER 并立即生效。在周期缩短时应当注意。如果 PWM 的时基已经超过新的 PWM 周期值，那么计数器现在必须计数到 0xFFFF，然后翻转。对此，如果需要安全地完成立即更新，就最好的做法是在周期更新之间捕获时间基数值，这样就可以计算好一个安全的最小周期值并写入。

即时更新占空比和相移

如果 PWM 已经完成，那么立即更新占空比将延迟到下一个周期。如果通过写一个更小的占空比值来缩短 PWM 脉冲，并且时基已经超过了新的占空比值（但还没有到原有的占空比的计数值），则下降沿生成点将被错过。这将导致当前 PWM 周期的占空比为 100%。

对于相位更新，如果新的 PWM 脉冲仍在进行中，且该值大于现有的相位偏移值，则新值立即启用。然而当 PWM 脉冲的相位偏移减少或脉冲宽度延长时，此时当格外小心。所以如果需要立即更新，那么最佳方法是在占空比或相位更新之前捕获时基值，以便可以计算安全值并写入。如果相位偏移缩短，并且时基已经超过了新的相位偏移比较时间点，则当前 PWM 周期将产生 0% 的占空比。

下图是即时更新过程中的两个修正示例。在这些例子中，PWM 周期相对较短，并进行了较大的占空比调整以强调如何校正工作。在两个例子中，占空比都是大约在 PWM 周期中点位置处从 75% 下降到 25%，此时时基值都已经超过了 25% 的比较时间点。

在第一种情况下，立即更新写入大约发生在 55% 占空比下，PWM 脉冲被立即截断，因为 PWM 时基至少是 0x8F。而原设置占空比是 0x7F，所以在 0x7F 比较时，0x80 的值显然较大。在第二种情况下，立即的更新写操作发生在新编程的占空比之后，PWM 脉冲不会被截断，直到时基值为 0x0080 且“大于”比较为真。

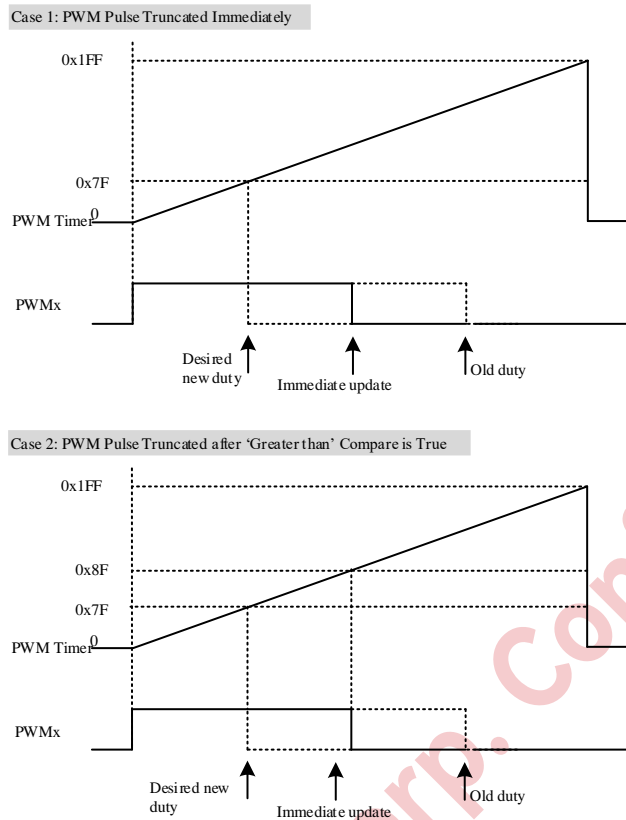


图 196 立即更新修正示例

即时更新死区时间:

如果死区窗口正在进行中并且发生了对死区时间的立即更新，则更新后的死区时间将延长。延长是由于死区计数器在结束之前被重置加载。如预期的那样，在立即更新之后的未来死区时间则是新的设定时间。

17.4.4.12 时基捕获

由于 PWM 定时器本身不是直接可读的，因此提供了一个时基捕获功能。当需要计数器值时，可通过 PGxCAP 寄存器捕获并读取。这里有两种方法可以捕获计数器值：手动软件触发或者硬件 PCI 事件触发。CAPSRC 则是用于选择手动捕获还是四个 PCI 块中哪一个作为触发捕获。

往 PGxCAP[0]位写 1 可完成手动捕获计数器值。CAP 状态位将设置为指示捕获已经完成，然后用户可以读取 PGxCAP 寄存器以确定事件发生时计数器值。PGxCAP 的读操作将清除 CAP 状态位。在用户软件读取 PGxCAP 之前，不允许进行新的捕获。类似地，当使用 PCI 块捕获一个时基值时，需要一个读操作重置逻辑以允许后续的捕获操作。建议在读取 PGxCAP 之前先确认 CAP 状态位是否为高。这是为了避免在硬件写入 PGxCAP 的同时读取它。另一种方法是用中断调度读取以避免并发访问。

在触发捕获事件生成与真正完成捕获之间存在最多四个时钟周期的延迟，这是由于同步和采样延迟造成的。

时基捕获示例:

1. 读取 CAP 状态位并验证为 0 (没有等待捕获)
2. 启动捕获事件 (软件触发或 PCI)
3. 轮询 CAP 状态位并等待它设置为表示数据已经准备好

17.4.4.13 调试模式运作

当使用调试器停止程序运行时, PWMx 输出引脚可能处于对硬件有害的状态。为了避免这种情况, 加入了将引脚强制到预定状态的逻辑, 所述状态由 DBDAT 位定义。引脚状态仍然受覆盖优先级影响, 如表格-输出覆盖所示。

17.4.5 通用特性

17.4.5.1 主数据寄存器

PWM 模块有一组通用数据寄存器, 可选地分配给多个 PWM 发生器:

- MDC: 主占空比寄存器
- MPER: 主周期寄存器
- MPHASE: 主相位寄存器

这些主寄存器允许用户软件通过写入一个数据寄存器来影响多个 PWM 发生器的操作。每个发生器 PGxCON 的 MDSEL, MPERSEL 和 MPHSEL 控制位决定发生器将使用本地数据寄存器还是主数据寄存器。

17.4.5.2 LFSR-线性反馈移位寄存器

线性反馈移位寄存器 (LFSR) 是一种伪随机数发生器, 可提供 15 位的随机值, 用于应用程序中少量修改占空比和/或周期, 以抖动应用电路中功率晶体管的相应开关边沿。这种抖动有利于减少峰值 EMI (电磁干扰) 排放。

LFSR 寄存器的每次读取都将触发其新一次更新。LFSR 运作基于多项式 $X^{15}+X^{14}+1$ (GF(2)) 产生随机值, 电路如下图所示。该电路经零检测电路修改, 使得 0x0000 后边跟着 0x0001, 然后遍历运算式得出所有结果, 周而复始。LFSR 复位初值是 0x0000。

如果需要使用相同的 LFSR 值进行多次运算, 那么需要将该值读取保存到不被硬件修改的临时缓存中。

对 LFSR 的一个特定位的连续读取, 该位会形成具有脉冲自相关函数的伪噪声 (PN) 序列, 可能很有用于抖动应用。整个 15 位 LFSR 值具有自相关性, 这在某些应用中作为伪随机噪声的来源可能是不需要的, 它的使用应在最终的应用程序中进行验证。

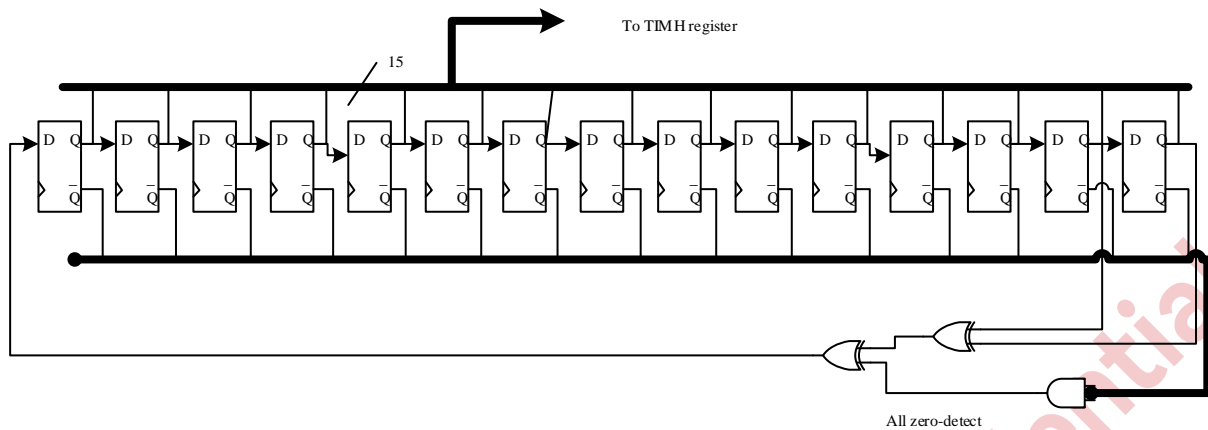


图 197 LFSR 模块图

17.4.5.3 共同时钟

PWM 时钟系统有两个额外的时钟功能以支持各种各样的应用。每个 PWM 发生器的时钟选择 (CLKSEL) 可以独立选择 TIMH_CLK，TIMH_CLK 的分频时钟或其频率缩放输出。

时钟分频器

一个通用的时钟分频电路可用于所有的 PWM 发生器，并允许 PWM 发生器在低频下运行。使用 DIVSEL 控制位可以选择四种不同分频。如果没有发生器有分频请求时，分频电路应保持在低功耗状态。

频率缩放

频率缩放模块具备降低时钟频率的能力，以有效地拉伸周期或占空比。这对于需要可变频率输入的谐振功率控制应用非常有用。频率缩放电路的输入时钟是 TIMH_CLK，其输出可用于每个 PWM 发动机，通过 CLKSEL 控制位进行选择应用。

FSCL(频率缩放) 和 FSMINPER(频率缩放最小周期) 寄存器指定了频率缩放的量，并且在任何时候都是可读写的。频率缩放电路内含一个累加器，不停地累加 FSCL 值，直到总和大于 FSMINPER 时，电路执行模运算，将累加值减去 FSMINPER 值，余数留在累加器接着运算，与此同时产生一个时钟脉冲，如下图所示。

注意：频率缩放时钟只应用于 PWM 时基计数器，不影响死区时间或 LEB 计数器的操作。如果没有任何 PWM 发生器选用，该电路保持低功耗状态。频率缩放时钟不能用于高精度模式。

示例：频率缩放计算

$$F_{FSCL} = (FSCL/FSMINPER) * F_{PWM}, \text{ 其中 } FSCL \leq FSMINPER.$$

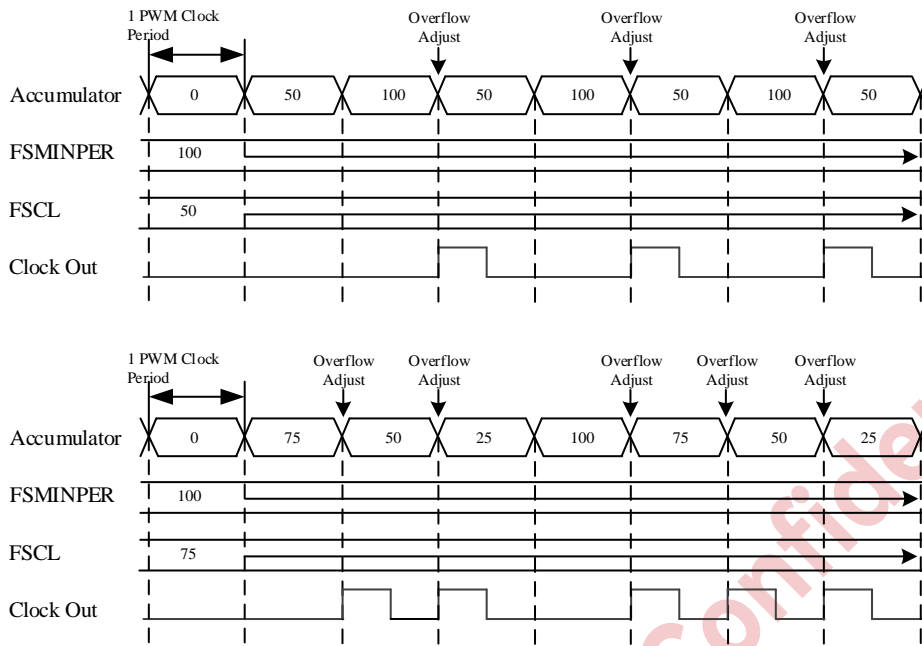


图 198 频率缩放示例

当选择频率缩放电路作为 PWM 发生器的时钟源时，PWM 发生器接收两个时钟，一个时钟是用来操作缩放频率缩放电路本身的原始时钟，用于操作 PWM 发生器内的死区时间计数器和 LEB 计数器。另一个时钟是缩放电路的时钟输出，其用于 PWM 时基计数器。

17.4.5.4 组合逻辑输出

组合逻辑输出功能可用于生成用于同步整流或其他应用的控制信号。基于一个或者多个 PWM 发生器构成一个逻辑功能输出，其中具有输入可选择和逻辑功能均可编程的特性。当启用该功能时，组合逻辑输出将取代原来发生器输出到引脚的 PWM 信号。组合路基输出块的数量与设备相关，具体参照数据说明书。功能实现包括以下控制。

- 输入源 (PWMSxy)
- 输入极性 (SxyPOL)
- 逻辑与、或、异或函数 (PWMLFy)
- 输出目的地 (PWMLFyD)

x 代表输入源 1 或 2，y 表示功能示例 (A, B)

下表所示是一个具有两个 LOGCONy 寄存器和组合逻辑输出功能的设备示例

表 277 组合逻辑例化映射

寄存器	组合逻辑例化	可用输出引脚选择
LOGCONA	A	PWM1H-PWM2H
LOGCONB	B	PWM1L-PWM2L

下图是组合逻辑功能实现框图。

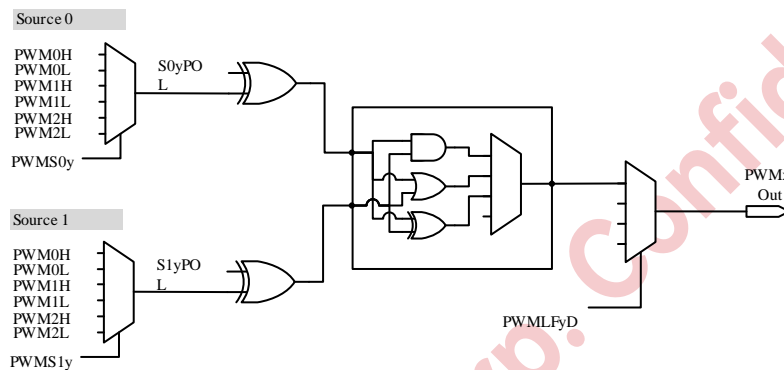


图 199 组合逻辑功能实现原理图

当使用组合逻辑输出功能时，来自 PWM 发生器的两个输入必须是相同的时钟源，否则可能输入无效。

组合输出的最小脉冲宽度是由具体设备所定，可能受输出引脚影响。

PWM 发生器的输出作为组合输出的输入源之前，PWMxH 和 PWMxL 的极性由 POLH 和 POLL 控制。如果没有选择输出目的地，则该组合功能关闭。输出目的地分组成对，LOGCONA 寄存器分配给 PWMH 输出控制，而 LOGCONB 寄存器分配给 PWML 输出控制。

只有 PWM1~2 可以使用组合逻辑输出，PWM0 不可用。

17.4.5.5 组合触发

使用组合触发功能可以创建复杂的触发算法。每个发生器中都有两个触发电路 A 和 B。该功能允许多个发生器的触发信号输出合成一个信号以用作另外一个发生器的触发源。

作为组合触发逻辑源的输入信号是由每个 PGxEVTL 控制器中的 PGTRGSEL 控制位所选的触发信号输出。这个触发信号输出可以是周期结束 (EOC)，也可以是 PGxTRIGy (y=A, B 或 C) 比较事件之一。通过设置 CMBTRIG 中的相关控制位，这些触发输出信号可被启用并逻辑 OR 组合在一起。经过组合触发后的组合触发 A 和 B 用于 PCI 的多路复用器输入，并通过 PCI 同步处理，最终可选作 PWM 发生器的输入触发。示例如下图所示。

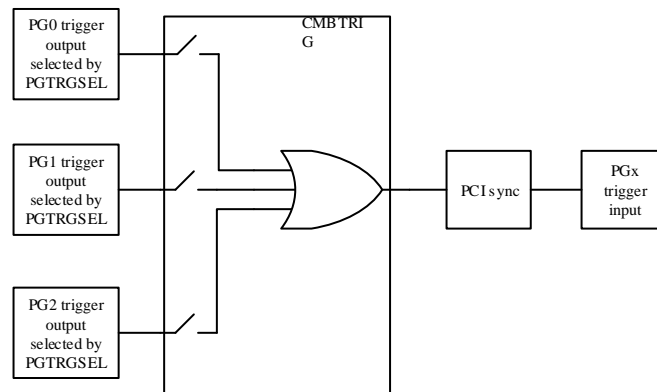


图 200 组合触发模块框图

17.4.5.6 PWM 事件输出

PWM 事件输出功能提供了一种机制将各种 PWM 信号和事件连接至其他外设或外部设备。逻辑模块支持从任何一个发生器中选择并处理事件信号的功能。相关控制包括以下配置选择：

- PWM 发生器 (PG0...PG2)选择
- 发生器信号选择
- 脉冲延展
- 输出信号极性
- 系统时钟同步

每个 PWMEVTy 寄存器包含用于 PWM 事件输出的控件。

EVTySEL 位选择输出块使用的信号，默认信号源时是由 PGTRGSEL 决定。有关这些信号和配置 ADC 触发的更多信息，请参考章节《事件选择块》。然后 EVTyPGS 用来选择使用三个发生器的事件信号。

一些高速运行的事件信号具有短脉冲，这可能不会被其他电路检测到，例如需要输出至片外的事件信号。脉冲延展电路可以通过设置 EVTySTRD 位来延长脉冲的持续时间。如果需要与主 PWM 时钟同步，则可以使能 EVTySYNC 位。EVTyPOL 控件用于反转事件信号的极性。

PWM 事件输出支持产生系统中断。

该功能模块框图如下图所示。

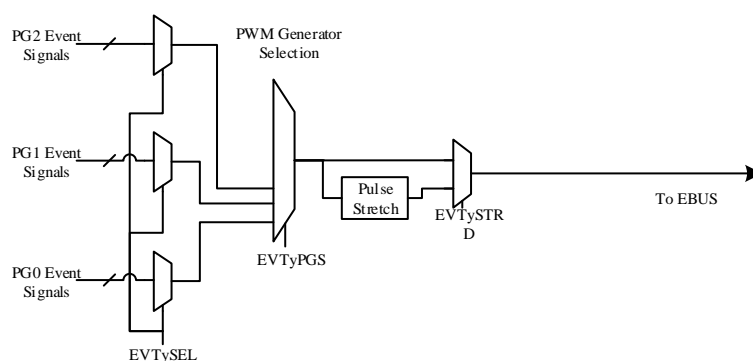


图 201 PWM 事件输出功能

17.4.6 寄存器锁定和写入限制

通过软件设置 LOCK 位 (PCLKCON[8]) 可阻止对某些寄存器的写入。然而解锁是需要执行一个特定的序列清除 LOCK 位。一般情况下, 禁止在模块运行时 (ON 位已置位) 对配置控制位进行修改。

谨慎修改数据寄存器（周期、占空比和相位），具体限制依赖于所使用的特定操作模式。根据 PWM 模式和数据缓存的相关要求合理修改相关寄存器。

寄存器的修改限制汇总如下所示。

表 278 寄存器锁定写入限制汇总

权限	受限寄存器或位
当 LOCK=1, 不可写	PCLKCON.DIVSEL PGxCON.HREN PGxIOCON.PMOD/PENH/PENL/POLH/POLL
当 UPDATE=1, 不可写	MPHASE/MDC PGxPHASE/PGxDC/PGxDCA/PGxPER PGxTRIGA/PGxTRIGB/PGxTRIGC PGxDTL/PGxDTH
当 ON=1, 避免修改	PCLKCON. DIVSEL PGxCON.HREN/CLKSEL/MDCSEL/MPERSEL/MPHSEL/MSTEN PGxIOCON.PMOD/PENH/PENL/POLH/POLL/DTCMPSEL CMBTRIG LOGCONy PGxEVT.ADTR1PS/ADTR1EN3/ADTR1EN2/ADTR1EN1/UPDTRG/PGTRGSEL PGxEVT.IEVTSEL/ADTR2EN3/ADTR2EN2/ADTR2EN1/ADTR1OFS PGxFPCI/PGxCLPCI/PGxFFPCI/PGxSPCI PGxLEB

17.4.7 TIMH 中断号

TIMH 的中断连接至不可屏蔽中断处理模块 (NVIC) 的 IRQ 相关接口。为了启用该中断功能，需要按照以下流程进行操作：

- 使能 TIMH 的中断使能位
- 使能 NVIC 中的 TIMH IRQ 通道, TIMH 的系统中断号如下表所示。
- 配置 TIMH 以产生中断
- TIMH 包括三个 PWM 生成模块，对应的中断号分别是：34，50，51。

表 279 TIMH 中断表

名称	中断号
TIMH0_INT	34
TIMH1_INT	50
TIMH2_INT	51

17.5 寄存器

17.5.1 通用功能寄存器

通用寄存器基地址是 0x40040800。下表为各控制寄存器描述。

表 280 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	PWM 时钟控制寄存器	32	r/w	0x00000000
0x04	频率标度寄存器	32	r/w	0x00000000
0x08	频率标度最小周期寄存器	32	r/w	0x00000000
0x0C	主相位寄存器	32	r/w	0x00000000
0x10	主占空度寄存器	32	r/w	0x00000000
0x14	主周期寄存器	32	r/w	0x00000000
0x18	线性反馈移位寄存器	32	r/w	0x00000000
0x1C	组合触发寄存器	32	r/w	0x00000000
0x20	组合 PWM 逻辑控制寄存器 y	32	r/w	0x00000000
0x24	PWM 事件输出控制寄存器 y	32	r/w	0x00000000
0x60	高精度电路控制寄存器	32	r/w	0x004F0000

17.5.1.1 PWM 时钟控制寄存器 (PCLKCON)

偏移地址: 0x0000

表 281 PWM 时钟控制寄存器

位	访问	描述
31	r/w	HRWORK: 高精度电路功能使能 0: 关闭 1: 开启 复位值: 0x0
30:16	Res	Reserved 复位值: 0x0
15	ro	HRRDY: 高精度功能可用状态 0: 高精度电路未准备好 1: 高精度电路已准备好 复位值: 0x0
14:9	Res	Reserved 复位值: 0x0
8	r/w	LOCK: 锁定 0: 具备写保护功能的寄存器和位处于解锁状态 1: 具备写保护功能的寄存器和位处于锁定状态 复位值: 0x0
7:2	Res	Reserved 复位值: 0x0
1:0	r/w	DIVSEL: PWM 时钟分频选择 00: 2 分频 01: 4 分频 10: 8 分频 11: 16 分频 复位值: 0x0

17.5.1.2 频率标度寄存器 (FSCL)

偏移地址: 0x0004

表 282 频率标度寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	FSCL: 频率标度 该寄存器值是作为频率标度累加器在每个 TIMH_CLK 时钟下的增加值。累加的结果大于 FSMINPER, 便产生一个有效时钟脉冲。 复位值: 0x0

17.5.1.3 频率标度最小周期寄存器 (FSMINPER)

偏移地址: 0x0008

表 283 频率标度寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	FSMINPER: 频率标度最小周期 该寄存器设定的是频率标度电路所能产生最小时钟周期。 复位值: 0

17.5.1.4 主相位寄存器 (MPHASE)

偏移地址: 0x000C

表 284 主相位寄存器

位	访问	描述
31:21	Res	Reserved 复位值: 0x0
20:0	r/w	MPHASE: 主相位 该寄存器设定的是多个 PWM 发生器所共享的相位偏移值。 注: 标准精度模式下, 位 16~20 保留, 无意义。 复位值: 0x0

17.5.1.5 主占空度寄存器 (MDC)

偏移地址: 0x0010

表 285 主占空度寄存器

位	访问	描述
31:21	Res	Reserved 复位值: 0x0
20:0	r/w	MDC: 主占空度 该寄存器设定的是多个 PWM 发生器所共享的占空度值。 注: 标准精度模式下, 位 16~20 保留, 无意义。要求占空度的值在标准精度模式大于等于 0x0008,而在高精度模式下应该大于等于 0x100。 复位值: 0x0

17.5.1.6 主周期寄存器 (MPER)

偏移地址: 0x0014

表 286 主周期寄存器

位	访问	描述
31:21	Res	Reserved 复位值: 0x0
20:0	r/w	MPER: 主周期 该寄存器设定的是多个 PWM 发生器所共享的周期值。 注: 标准精度模式下, 位 16~20 保留, 无意义。 要求占空度的值在标准精度模式大于等于 0x0020,而在高精度模式下应该大于等于 0x400 复位值: 0x0

17.5.1.7 线性反馈移位寄存器 (LFSR)

偏移地址: 0x0018

表 287 主周期寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:0	r/w	LFSR: 线性反馈移位寄存器 该寄存器的读操作会返回一个 15 位伪随机值 复位值: 0x0

17.5.1.8 组合触发寄存器 (CMBTRIG)

偏移地址: 0x001C

表 288 组合触发寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
18	r/w	CTB2EN: 使能 PG2 的触发输出作为组合触发 B 的输入源 0: 关闭 1: 使能指定触发信号以或操作并入组合触发 B 复位值: 0x0
17	r/w	CTB1EN: 使能 PG1 的触发输出作为组合触发 B 的输入源 0: 关闭 1: 使能指定触发信号以或操作并入组合触发 B 复位值: 0x0
16	r/w	CTB0EN: 使能 PG0 的触发输出作为组合触发 B 的输入源 0: 关闭 1: 使能指定触发信号以或操作并入组合触发 B 复位值: 0x0
15:3	Res	Reserved 复位值: 0x0
2	r/w	CTA2EN: 使能 PG2 的触发输出作为组合触发 A 的输入源 0: 关闭 1: 使能指定触发信号以或操作并入组合触发 A 复位值: 0x0
1	r/w	CTA1EN: 使能 PG1 的触发输出作为组合触发 A 的输入源 0: 关闭 1: 使能指定触发信号以或操作并入组合触发 A 复位值: 0x0
0	r/w	CTA0EN: 使能 PG0 的触发输出作为组合触发 A 的输入源 0: 关闭 1: 使能指定触发信号以或操作并入组合触发 A 复位值: 0x0

17.5.1.9 组合 PWM 逻辑控制寄存器 y (LOGCONy)

偏移地址: 0x0020~0x0024

备注: 其中 y 代表常见实例名称 (A-B)

表 289 组合 PWM 逻辑控制寄存器 y

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:12	r/w	PWMS1y: 组合 PWM 逻辑源 #1 选择: 0000: PWM0H 0001: PWM0L 0010: PWM1H 0011: PWM1L 0100: PWM2H 0101: PWM2L 其他: 保留 复位值: 0x0
11:8	r/w	PWMS0y: 组合 PWM 逻辑源 #0 选择: 0000: PWM0H 0001: PWM0L 0010: PWM1H 0011: PWM1L 0100: PWM2H 0101: PWM2L 其他: 保留 复位值: 0x0
7	r/w	S1yPOL: 组合 PWM 逻辑源 #1 极性 0: 正有效 1: 取反 复位值: 0x0
6	r/w	S0yPOL: 组合 PWM 逻辑源 #0 极性 0: 正有效 1: 取反 复位值: 0x0
5:4	r/w	PWMLFy: 组合 PWM 逻辑功能选择 00: PWMS1y PWMS0y (OR) 01: PWMS1y&PWMS0y (AND) 10: 保留 11: PWMS1y^PWMS0y (XOR) 复位值: 0x0
3	Res	Reserved 复位值: 0x0

位	访问	描述
2:0	r/w	PWMLFyD: 组合 PWM 逻辑输出目标选择 000: 关闭 001: 选择 PWM1 010: 选择 PWM2 其他: 保留 复位值: 0x0

17.5.1.10 PWM 事件输出控制寄存器 y (PWMEVTy)

偏移地址: 0x0040~48

备注: 其中 y 代表常见实例名称 (A-C)

表 290 组合 PWM 逻辑控制寄存器 y

位	访问	描述
31:14	Res	Reserved 复位值: 0x0
13	r/w	EVTySTRD: PWM 事件输出延展控制 0: 输出事件信号脉冲宽度延展至 8 个工作时钟宽度 1: 输出事件信号脉冲宽度未延展 复位值: 0x0
12:8	Res	Reserved 复位值: 0x0
7:4	r/w	EVTySE: PWM 事件选择 0000: PGTRGSEL 所选信号 0001: PWM 发生器输出 0010: PCI 同步有效输出 0011: PCI 前馈有效输出 0100: PCI 限流有效输出 0101: PCI 故障有效输出 0110: CAHALF (只在中心对称模式有效) 0111: STEER (只在推挽输出模式有效) 1000: ADC 触发 1 1001: ADC 触发 2 1010-1110: 保留 1111: 高精度错误事件 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:0	r/w	EVTyPGS: PWM 事件源选择 000: PG0 001: PG1 010: PG2 其他: 保留 复位值: 0x0

17.5.1.11 高精度电路控制寄存器 (HRCON)

偏移地址: 0x0060

表 291 高精度控制寄存器

位	访问	描述
31:24	Res	Reserved 复位值: 0x0
23:16	ro	DLYSEL: 延迟控制级数 (源自内部 PLL 调节所得) 复位值: 0x4F (不同芯片可能不一样, 同一芯片在不同电压温度可能也不一样)
15:8	r/w	SWDLYSEL: 软件延迟控制级数 在 SWDLYEN 生效时, 该值作为所有通道的延迟总级数控制 复位值: 0x0
7	Res	Reserved 复位值: 0x0
6:4	r/w	DLYTRIM: 微调 DLL 输出级数 000: 不调整 001: +1 级 010: +2 级 011: +3 级 100: 不调整 101: -1 级 110: -2 级 111: -3 级 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	SWDLYEN: 软件延迟控制使能 0: 关闭 1: 开启 复位值: 0x0

17.5.2 PWM 发生器寄存器

TIMH 包含 3 个 PWM 发生器，其基地址分别是 0x40040900，0x40040980，0x40040A00。下表为各控制寄存器描述。

说明：x=PWM 发生器编号 0、1、2。

表 292 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	PGx 控制寄存器	32	r/w	0x00000000
0x04	PGx 状态寄存器	32	r/w	0x00000000
0x08	PGx I/O 控制寄存器	32	r/w	0x00000000
0x0C	PGx 事件寄存器	32	r/w	0x00000000
0x10	PGx FPCI 寄存器	32	r/w	0x00000000
0x14	PGx CLPCI 寄存器	32	r/w	0x00000000
0x18	PGx FFPCI 寄存器	32	r/w	0x00000000
0x1C	PGx SPCI 寄存器	32	r/w	0x00000000
0x20	PGx 前沿消隐寄存器	32	r/w	0x00000000
0x24	PGx 相位寄存器	32	r/w	0x00000000
0x28	PGx 占空度寄存器	32	r/w	0x00000000
0x2C	PGx 占空度调整寄存器	32	r/w	0x0000FFFF
0x30	PGx 周期寄存器	32	r/w	0x00000000
0x34	PGx 触发 A 寄存器	32	r/w	0x00000000
0x38	PGx 触发 B 寄存器	32	r/w	0x00000000
0x3C	PGx 触发 C 寄存器	32	r/w	0x00000000
0x40	PGx 死区寄存器	32	r/w	0x00000000
0x44	PGx 捕获寄存器	32	r/w	0x00000000

17.5.2.1 PGx 控制寄存器 (PGxCON)

偏移地址：0x0000

表 293 PGx 控制寄存器

位	访问	描述
31	r/w	MDCSEL：主占空度寄存器选择 0：PG 使用 PGxDC 1：PG 使用 MDC 复位值：0x0

位	访问	描述
30	r/w	MPERSEL: 主周期寄存器选择 0: PG 使用 PGxPER 1: PG 使用 MPER 复位值: 0x0
29	r/w	MPHSEL: 主相位寄存器选择 0: PG 使用 PGxPHASE 1: PG 使用 MPHASE 复位值: 0x0
28	Res	Reserved 复位值: 0x0
27	r/w	MSTEN: 本 PG 往其他 PG 广播 UPDREQ 使能控制 0: 关闭 1: 开启 复位值: 0x0
26:24	r/w	UPDMOD: PWM 缓冲更新模式选择 000: SOC 模式——当 UPDREQ=1 时, 数据寄存器在接下来的 PWM 周期的起始处更新。 001: 立即模式——当 UPDREQ=1 时, 数据寄存器立即更新。 010: 从 SOC 模式——只有接收到主机端更新请求后, 数据寄存器在接下来的 PWM 周期的起始处更新。 011: 从立即模式——只有接收到主机端更新请求后, 数据寄存器立即更新。 其他: 保留 复位值: 0x0
23	Res	Reserved 复位值: 0x0
22	r/w	TRGMOD: PGx 触发模式选择 0: 单次触发模式 1: 重触发模式 复位值: 0x0
21:20	Res	Reserved 复位值: 0x0
19:16	r/w	SOCS: SOC (Start-of-cycle)选择 0000: 本地 EOC, PWM 发生器自动触发工作 0001: PG0 触发输出 (PGTRGSEL) 0010: PG1 触发输出 (PGTRGSEL) 0011: PG2 触发输出 (PGTRGSEL) 0100-1110: 保留 1111: TRIG 位或 PCI 同步功能 复位值: 0x0
15	r/w	ON: PGx 使能 0: 关闭

位	访问	描述
		1: 开启 复位值: 0x0
14:11	Res	Reserved 复位值: 0x0
10:8	r/w	TRGCNT: PGx 触发计数选择 000: PG 在触发后产生 1 个 PWM 周期 001: PG 在触发后产生 2 个 PWM 周期 010: PG 在触发后产生 3 个 PWM 周期 011: PG 在触发后产生 4 个 PWM 周期 100: PG 在触发后产生 5 个 PWM 周期 101: PG 在触发后产生 6 个 PWM 周期 110: PG 在触发后产生 7 个 PWM 周期 111: PG 在触发后产生 8 个 PWM 周期 复位值: 0x0
7	r/w	HREN: 高精度模式使能 0: 工作于标准精度模式 1: 工作于高精度模式 复位值: 0x0
6:5	Res	Reserved 复位值: 0x0
4:3	r/w	CLKSEL: 时钟选择 00: 无时钟选择, PG 处于低功耗模式 01: TIMH_CLK 10: TIMH_CLK 的分频时钟 11: TIMH_CLK 频率缩放时钟 (Frequency Scaling) 复位值: 0x0
2:0	r/w	MODSEL: PGx 模式选择 000: 独立边沿 PWM 模式 001: 可变相位 PWM 模式 010: 独立边沿 PWM 模式 (双输出) 011: 保留 100: 中心对齐 PWM 模式 101: 中心对齐 PWM 模式 (双重更新) 110: 双边沿中心对齐 PWM 模式 (每个 PWM 周期产生一次中断或者寄存器更新) 111: 双边沿中心对齐 PWM 模式 (每个 PWM 周期产生两次中断或者寄存器更新) 复位值: 0x0

17.5.2.2 PGx 状态寄存器 (PGxSTAT)

偏移地址: 0x0004

表 294 PGx 状态寄存器

位	访问	描述
31:29	Res	Reserved 复位值: 0x0
28	r/w	ADTRG2: ADC 触发 2 状态 0: 无 ADC 触发 2 事件发生 1: 发生 ADC 触发 2 事件 注: 该位写 1 清零 复位值: 0x0
27	r/w	ADTRG1: ADC 触发 1 状态 0: 无 ADC 触发 1 事件发生 1: 发生 ADC 触发 1 事件 注: 该位写 1 清零 复位值: 0x0
26	r/w	TRIGC: 触发 C 状态 0: 无触发 C 事件发生 1: 发生触发 C 事件 注: 该位写 1 清零 复位值: 0x0
25	r/w	TRIGB: 触发 B 状态 0: 无触发 B 事件发生 1: 发生触发 B 事件 注: 该位写 1 清零 复位值: 0x0
24	r/w	TRIGA: 触发 A 状态 0: 无触发 A 事件发生 1: 发生触发 A 事件 注: 该位写 1 清零 复位值: 0x0
23:19	Res	Reserved 复位值: 0x0
18	r/w	HFEVT: 硬件错误事件状态 0: 硬件错误事件发生 1: 发生硬件错误事件 注: 该位写 1 清零 复位值: 0x0
17	r/w	TOGERR: 翻转错误状态 (高精度模式下, PWM 生成电路错误) 0: 无翻转错误发生 1: 发生翻转错误 注: 该位写 1 清零 复位值: 0x0

位	访问	描述
16	r/w	IEVT: 普通中断事件状态 0: 无普通中断事件发生 1: 发生普通中断事件 注: 该位写 1 清零 复位值: 0x0
15	r/w	SEVT: PCI 同步事件状态 0: 无 PCI 同步事件发生 1: 发生 PCI 同步事件 (PCI 同步输出为高或出现脉冲) 注: 该位写 1 清零 复位值: 0x0
14	r/w	FLTEVT: PCI 故障状态 0: 无 PCI 故障事件发生 1: 发生 PCI 故障事件 (PCI 故障输出为高或出现脉冲) 注: 该位写 1 清零 复位值: 0x0
13	r/w	CLEVT: PCI 限流状态 0: 无 PCI 限流事件发生 1: 发生 PCI 限流事件 (PCI 限流输出为高或出现脉冲) 注: 该位写 1 清零 复位值: 0x0
12	r/w	FFEVT: PCI 前馈状态 0: 无 PCI 前馈事件发生 1: 发生 PCI 前馈事件 (PCI 前馈输出为高或出现脉冲) 注: 该位写 1 清零 复位值: 0x0
11	ro	SACT: PCI 同步状态 0: PCI 同步输出无效 1: PCI 同步输出有效 复位值: 0x0
10	ro	FLTACT: PCI 故障有效状态 0: PCI 故障输出无效 1: PCI 故障输出有效 复位值: 0x0
9	ro	CLACT: PCI 限流有效状态 0: PCI 限流输出无效 1: PCI 限流输出有效 复位值: 0x0
8	ro	FFACT: PCI 前馈有效状态 0: PCI 前馈输出无效 1: PCI 前馈输出有效 复位值: 0x0

位	访问	描述
7	r/w	<p>TRSET: PG 软件触发置位</p> <p>该位写 1 触发开启 PWM 周期信号, TRIG 位则响应并指示为 1.</p> <p>该位自动清零。</p> <p>复位值: 0x0</p>
6	r/w	<p>TRCLR: PG 软件触发清零</p> <p>该位写 1 停止 PWM 周期信号, TRIG 位则响应并指示为 0.</p> <p>该位自动清零。</p> <p>复位值: 0x0</p>
5	r/w	<p>CAP: 捕获状态</p> <p>0: 无捕获发生</p> <p>1: 发生捕获, 时基值已捕获存入 PGxCAP</p> <p>注: 该位写 1 清零</p> <p>复位值: 0x0</p>
4	ro	<p>UPDATE: PWM 数据寄存器更新状态或控制</p> <p>0: 无更新未决或已完成更新</p> <p>1: PWM 数据寄存器处于更新未决中, 此时不可写</p> <p>复位值: 0x0</p>
3	r/w	<p>UPDREQ: PWM 数据寄存器更新请求</p> <p>0: 无请求</p> <p>1: 更新请求</p> <p>注: 该位自动清零</p> <p>复位值: 0x0</p>
2	ro	<p>STEER: 输出导引状态 (仅推挽模式下有效)</p> <p>0: PG 处于推挽模式下的第一个周期</p> <p>1: PG 处于推挽模式下的第二个周期</p> <p>复位值: 0x0</p>
1	ro	<p>CAHALF: 半周期状态 (仅中心对称模式下有效)</p> <p>0: PG 处于第一个半时基周期</p> <p>1: PG 处于第二个半时基周期</p> <p>复位值: 0x0</p>
0	ro	<p>TRIG: 触发状态</p> <p>0: 无 PWM 周期信号产生</p> <p>1: PG 已触发, 正产生 PWM 周期信号中</p> <p>复位值: 0x0</p>

17.5.2.3 PGx I/O 控制寄存器 (PGxIOCON)

偏移地址: 0x0008

表 295 PGx I/O 控制寄存器

位	访问	描述
31	Res	Reserved 复位值: 0x0
30:28	r/w	CAPSRC: 时基捕获源选择 000: 无硬件触发源, 仅支持软件触发 001: PCI 同步信号生效 010: PCI 前馈信号生效 011: PCI 限流信号生效 100: PCI 故障信号生效 其他: 保留 复位值: 0x0
27:25	Res	Reserved 复位值: 0x0
24	r/w	DTCMPSEL: 死区补偿选择 0: 死区补偿由 PCI 同步逻辑控制 1: 死区补偿由 PCI 前馈逻辑控制 复位值: 0x0
23:22	r/w	IDAT: 输出关闭时 PWMxH/PWMxL 的预设值 当输出关闭时, IDAT[1] 值作为 PWMxH 输出 IDDAT[0] 值作为 PWMxL 输出 注: 当发生芯片故障时 (时钟丢失, SRAM 校验错误等), PEN*立即被清零, PWM 输出立即覆盖 复位值: 0x0
21:20	r/w	PMOD: 输出模式选择 00: 互补模式 01: 独立模式 10: 推挽模式 11: 保留 复位值: 0x0
19	r/w	PENH: PENH 输出使能 0: 关闭 1: 开启 注: 当发生芯片故障时 (时钟丢失, SRAM 校验错误等), 该控制位被立即清零, 输出状态由 IDAT[1]决定。 复位值: 0x0
18	r/w	PENL: PENL 输出使能

位	访问	描述
		0: 关闭 1: 开启 注: 当发生芯片故障时 (时钟丢失, SRAM 校验错误等), 该控制位被立即清零, 输出状态由 IDAT[0]决定。 复位值: 0x0
17	r/w	POLH: PWMxH 输出极性 0: 高有效 1: 低有效 复位值: 0x0
16	r/w	POLL: PWMxL 输出极性 0: 高有效 1: 低有效 复位值: 0x0
15	r/w	CLMOD: 限流模式选择 0: 当 PCI 限流事件生效时, CLDAT 值则是 PWM 输出电平 1: 当 PCI 限流事件生效时, PWM 输出信号翻转, 不由 CLDAT 决定。 复位值: 0x0
14	r/w	SWAP: 交换输出至 PWMxH 和 PWMxL 引脚的 PWM 信号 0: PWMxH/L 信号映射到对应到的 PWMxH/L 引脚 1: PWMxH 信号映射到 PWMxL 引脚, 而 PWMxL 信号映射到 PWMxH 引脚 复位值: 0x0
13	r/w	OVRENH: 针对 PWMxH 的用户覆盖控制 0: PWMxH 输出由 PWM 发生器提供 1: PWMxH 输出由 OVRDAT[1]提供 复位值: 0x0
12	r/w	OVRENH: 针对 PWMxL 的用户覆盖控制 0: PWMxL 输出由 PWM 发生器提供 1: PWMxL 输出由 OVRDAT[0]提供 复位值: 0x0
11:10	r/w	OVRDAT: 输出覆盖生效时 PWMxH/PWMxL 的预设值 当输出覆盖生效时, OVRDAT[1]值作为 PWMxH 输出 OVRDAT[0]值作为 PWMxL 输出 复位值: 0x0
9:8	r/w	OSYNC: 用户输出覆盖同步控制 实现用户输出覆盖是通过配置 SWAP、OVERENL/H 和 OVRDAT 实现, 而 OSYNC 用于控制完成更新上述配置后以何种方式生效, 00: 同步至当前 PWM 时基, 即下一个 SOC 生效 01: 立即生效 10: 依据 UPDMOD 定义的方式生效 11: 保留

位	访问	描述
		复位值: 0x0
7:6	r/w	FLTDAT: 故障事件生效时 PWMxH/PWMxL 的预设值 当故障事件生效时, FLTDAT[1]值作为 PWMxH 输出 FLTDAT[0]值作为 PWMxL 输出 复位值: 0x0
5:4	r/w	CLDAT: 限流事件生效时 PWMxH/PWMxL 的预设值 当限流事件生效时, CLDAT[1]值作为 PWMxH 输出 CLDAT[0]值作为 PWMxL 输出 复位值: 0x0
3:2	r/w	FFDAT: 前馈事件生效时 PWMxH/PWMxL 的预设值 当前馈生效时, FFDAT[1]值作为 PWMxH 输出 FFDAT[0]值作为 PWMxL 输出 注: 当前馈事件选择作为死区补偿控制信号时, 此时不作为输出覆盖事件 复位值: 0x0
1:0	r/w	DBDAT: 调试模式生效时 PWMxH/PWMxL 的预设值 当调试模式生效时, 并且 DBG_TIMH_BRK=1。 DBDAT[1]值作为 PWMxH 输出 DBDAT[0]值作为 PWMxL 输出 复位值: 0x0

17.5.2.4 PGx 事件寄存器 (PGxEVT)

偏移地址: 0x000C

表 296 PGx 事件寄存器

位	访问	描述
31	r/w	FLTIE: PCI 故障中断使能 0: 关闭 1: 使能 复位值: 0x0
30	r/w	CLIE: PCI 限流中断使能 0: 关闭 1: 使能 复位值: 0x0

位	访问	描述
29	r/w	FFIEN: PCI 前馈中断使能 0: 关闭 1: 使能 复位值: 0x0
28	r/w	SIEN: PCI 同步中断使能 0: 关闭 1: 使能 复位值: 0x0
27	Res	Reserved 复位值: 0x0
26:24	r/w	IEVTSEL: 中断事件选择 000: EOC 001: TRIGA 比较事件 010: TRIGB 比较事件 011: TRIGC 比较事件 100: ADC 触发 1 事件 101: ADC 触发 2 事件 110: 缓存更新事件 111: 关闭。(PCI 事件仍可正常配置为中断) 复位值: 0x0
23	r/w	ADTR2EN3: PGxTRIGC 比较事件作为 ADC 触发 2 源使能控制 0: 关闭 1: 使能 复位值: 0x0
22	r/w	ADTR2EN2: PGxTRIGB 比较事件作为 ADC 触发 2 源使能控制 0: 关闭 1: 使能 复位值: 0x0
21	r/w	ADTR2EN1: PGxTRIGA 比较事件作为 ADC 触发 2 源使能控制 0: 关闭 1: 使能 复位值: 0x0
20:16	r/w	ADTR1OFS: ADC 触发 1 偏移选择 00000: 无偏移 00001: 偏移 1 个触发事件 00010: 偏移 2 个触发事件 ... 11110: 偏移 30 个触发事件 11111: 偏移 31 个触发事件 复位值: 0x0
15:11	r/w	ADTR1PS: ADC 触发 1 后分频选择

位	访问	描述
		00000: 1 分频 00001: 2 分频 00010: 3 分频 ... 11110: 31 分频 11111: 32 分频 复位值: 0x0
10	r/w	ADTR1EN3: PGxTRIGC 比较事件作为 ADC 触发 1 源使能控制 0: 关闭 1: 使能 复位值: 0x0
9	r/w	ADTR1EN2: PGxTRIGB 比较事件作为 ADC 触发 1 源使能控制 0: 关闭 1: 使能 复位值: 0x0
8	r/w	ADTR1EN1: PGxTRIGA 比较事件作为 ADC 触发 1 源使能控制 0: 关闭 1: 使能 复位值: 0x0
7:6	Res	Reserved 复位值: 0x0
5	r/w	HFEN: 硬件错误响应使能 0: 关闭 1: 使能 复位值: 0x0
4:3	r/w	UPDTRG: 更新触发选择 00: 手动设置 UPDREQ 触发 01: 写寄存器 PGxDC 完成 UPDREQ 自动置位 10: 写寄存器 PGxPHASE 完成 UPDREQ 自动置位 11: 写寄存器 PGxTRIGA 完成 UPDREQ 自动置位 复位值: 0x0
2:0	r/w	PGTRGSEL: PWM 发生器触发输出 (PGxTRGO)选择 000: EOC (End-of-Cycle)事件 001: PGxTRIGA 比较事件 010: PGxTRIGB 比较事件 011: PGxTRIGC 比较事件 其他: 保留 复位值: 0x0

17.5.2.5 PGxyPCI 寄存器 (x=PG 编号, y=F,CL,FF 或 S)

偏移地址: 0x0010~0x001C

表 297 PWM 发生器 xyPCI 寄存器 (输出比较模式)

位	访问	描述
31	r/w	BPEN: PCI 旁路使能 0: PCI 功能未被旁路 1: 本地的 PCI 功能被旁路,PCI 输入可配置为其他 PG 的 PCI 输出 (BPSEL) 复位值: 0x0
30:28	r/w	BPSEL: PCI 旁路源选择 000: 当 BPEN=1, PCI 控制源于 PG0 PCI 逻辑 001: 当 BPEN=1, PCI 控制源于 PG1 PCI 逻辑 010: 当 BPEN=1, PCI 控制源于 PG2 PCI 逻辑 其他: 保留 注: 在 BPEN 已使能的情况下, 选择本地 PCI 功能, 则相当于没有 PCI 输出, 即关闭本 PCI 输出。 复位值: 0x0
27	Res	Reserved 复位值: 0x0
26:24	r/w	ACP:PCI 接收标准选择 000: 水平触发 001: 上升沿 010: 任意沿 011: 锁存 100: 上升沿锁存 101: 任意沿锁存 其他: 保留 复位值: 0x0
23	r/w	SWPCI: 软件 PCI 控制 0: 驱动 0 至 PCI 逻辑 (由 SWPCIM 所分配) 1: 驱动 1 至 PCI 逻辑 (由 SWPCIM 所分配) 复位值: 0x0
22:21	r/w	SWPCIM: 软件 PCI 控制模式 00: SWPCI 分配给 PCI 接收逻辑 01: SWPCI 分配给接收界定符逻辑 10: SWPCI 分配给终结界定符逻辑 11: 保留 复位值: 0x0
20	r/w	LATMOD: PCI SR 锁存模式 0: 置位优先 1: 重置优先 复位值: 0x0

位	访问	描述
19	r/w	TQPS: 终结界定符极性选择 0: 不取反 1: 取反 复位值: 0x0
18:16	r/w	TQSS: 终结界定符源选择 000: 无终结界定符 001: PWM 周期信号 (高) 010: LEB 生效状态 011: PWM 发生器已被触发工作 100: PCI 源 1 (PG 输出, PWMPCI) 101: PCI 源 8 110: PCI 源 9 111: 仅 SWPCI 控制位 复位值: 0x0
15	r/w	TSYNCDIS: 事件终结同步关闭 0: 终结锁存后的 PCI 有效信号会同步到 PWM EOC 处发生 1: 终结锁存后的 PCI 有效信号立刻发生 复位值: 0x0
14:12	r/w	TERM: 终结事件选择 000: 手动终结, 通过对 SWTERM 置位 001: 自动终结, 在 PCI 源从有效切换到无效时终结 010: PGxTRIGA 触发事件 011: PGxTRIGB 触发事件 100: PGxTRIGC 触发事件 101: PCI 源 1 (PG 输出, PWMPCI) 110: PCI 源 8 111: PCI 源 9 复位值: 0x0
11	r/w	AQPS: 接收界定符极性选择 0: 不取反 1: 取反 复位值: 0x0
10:8	r/w	AQSS: 接收界定符源选择 (Acceptance Qualifier Source Selection) 000: 无接收界定符 (界定符强制为 1) 001: PWM 周期信号 (高) 010: LEB 生效状态 011: PWM 发生器已被触发工作 100: PCI 源 1 (PG 输出, PWMPCI) 101: PCI 源 8 110: PCI 源 9 111: 仅 SWPCI 控制位

位	访问	描述
		复位值: 0x0
7	r/w	SWTERM: PCI 软件结束 写 1 产生结束事件, 自动清零 复位值: 0x0
6	r/w	PSYNC: PCI 同步控制 0: PCI 源直接输入 1: PCI 源同步至 PWM EOC 复位值: 0x0
5	r/w	PPS: PCI 极性选择 0: 不取反 1: 取反 复位值: 0x0
4:0	r/w	PSS: PCI 源选择 00000: 软件 PCI 控制位 (SWPCI) 00001: PCI 源 1 (PWMPCI 多路复用输出) 00010: PCI 源 2 (组合触发 A) 00011: PCI 源 3 (组合触发 B) 00100: PCI 源 4 (保留) 00100: PCI 源 5 (保留) ... 00111: PCI 源 7 (保留) 01000: PCI 源 8 (EBUS_CH_EV [8]) 01001: PCI 源 9 (EBUS_CH_EV [9]) ... 01111: PCI 源 15 (EBUS_CH_EV [15]) 10000: PCI 源 16 (保留) 10000: PCI 源 17 (保留) ... 11111: PCI 源 31 (保留) 复位值: 0x0

17.5.2.6 PGx 前沿消隐寄存器 (PGxLEB)

偏移地址: 0x0020

表 298 PGx 前沿消隐寄存器

位	访问	描述
31:27	Res	Reserved 复位值: 0x0
26:24	r/w	PWMPCI: PCI 的 PWM 源选择 000: PG0 输出 001: PG1 输出 010: PG2 输出

位	访问	描述
		其他：保留 复位值：0x0
23:20	Res	Reserved 复位值：0x0
19	r/w	PHR: PWMxH 上升沿触发使能 0: 关闭 1: 开启。PWMxH 上升沿触发 LEB 开始计数。 复位值：0x0
18	r/w	PHF: PWMxH 下降沿触发使能 0: 关闭 1: 开启。PWMxH 下降沿触发 LEB 开始计数。 复位值：0x0
17	r/w	PLR: PWMxL 上升沿触发使能 0: 关闭 1: 开启。PWMxL 上升沿触发 LEB 开始计数。 复位值：0x0
16	r/w	PLF: PWMxL 下降沿触发使能 0: 关闭 1: 开启。PWMxL 下降沿触发 LEB 开始计数。 复位值：0x0
15:0	r/w	LEBP: 前沿消隐周期 该配置作为消隐周期宽度设置，计量单位为 PGx_CLK，最小周期为 8，低三位固定为 0 复位值：0x0

17.5.2.7 PGx 相位寄存器 (PGxPHASE)

偏移地址：0x0024

表 299 PGx 相位寄存器

位	访问	描述
31:21	Res	Reserved 复位值：0x0
20:0	r/w	PGxPHASE: PWM 发生器 x 相位 该寄存器设定的是 PWM 发生器 x 的相位偏移值。 注：标准精度模式下，位 16~20 保留，无意义。 复位值：0x0

17.5.2.8 PGx 占空度寄存器 (PGxDC)

偏移地址: 0x0028

表 300 PGx 占空度寄存器

位	访问	描述
31:21	Res	Reserved 复位值: 0x0
20:0	r/w	PGxDC: PWM 发生器 x 占空度 该寄存器设定的是 PWM 发生器 x 的占空度值。 注: 标准精度模式下, 位 16~20 保留, 无意义。要求占空度的值在标准精度模式大于等于 0x0008, 而在高精度模式下应该大于等于 0x100。 复位值: 0x0

17.5.2.9 PGx 占空度调整寄存器 (PGxDCA)

偏移地址: 0x002C

表 301 PGx 占空度调整寄存器

位	访问	描述
31:13	Res	Reserved 复位值: 0x0
12:0	r/w	PGxDCA: PWM 发生器 x 占空度调整值 当 PCI 源生效时, PGxDCA 便加在 PGxDC 上生成真正有效的占空度, 从而改变 PWM 输出占空度。 注: 标准精度模式下, 位 8~12 保留, 无意义。高精度模式下, 位 0~4 固定为 0。 复位值: 0x0

17.5.2.10 PGx 周期寄存器 (PGxPER)

偏移地址: 0x0030

表 302 PGx 周期寄存器

位	访问	描述
31:21	Res	Reserved 复位值: 0x0
20:0	r/w	PGxPER: PWM 发生器 x 周期 该寄存器设定的是 PWM 发生器 x 的周期值。 注: 标准精度模式下, 位 16~20 保留, 无意义。要求周期值在标准精度模式大于等于 0x0020, 而在高精度模式下应该大于等于 0x400。 复位值: 0x0

17.5.2.11 PGx 触发 A 寄存器 (PGxTRIGA)

偏移地址: 0x0034

表 303 PGx 周期寄存器

位	访问	描述
31	r/w	TRIGASEL: 触发 A 事件发生选择 (仅在中心对称模式有效) 0: 在 CAHALF=0 进行比较 1: 在 CAHALF=1 进行比较 复位值: 0x0
30:21	Res	Reserved 复位值: 0x0
20:0	r/w	PGxTRIGA: PWM 发生器 x 触发 A 注: 标准精度模式下, 位 16~20 保留, 无意义。高精度模式下, 位 0~4 在双 PWM 模式下可正常使用, 而在其他运行模式固定为 0。 复位值: 0x0

17.5.2.12 PGx 触发 B 寄存器 (PGxTRIGB)

偏移地址: 0x0038

表 304 PGx 触发 B 寄存器

位	访问	描述
31	r/w	TRIGBSEL: 触发 B 事件发生选择 (仅在中心对称模式有效) 0: 在 CAHALF=0 进行比较 1: 在 CAHALF=1 进行比较 复位值: 0x0
30:21	Res	Reserved 复位值: 0x0
20:0	r/w	PGxTRIGB: PWM 发生器 x 触发 B 注: 标准精度模式下, 位 16~21 保留, 无意义。高精度模式下, 位 0~4 在双 PWM 模式下可正常使用, 而在其他运行模式固定为 0。 复位值: 0x0

17.5.2.13 PGx 触发 C 寄存器 (PGxTRIGC)

偏移地址: 0x003C

表 305 PGx 触发 C 寄存器

位	访问	描述
31	r/w	TRIGCSEL: 触发 C 事件发生选择 (仅在中心对称模式有效) 0: 在 CAHALF=0 进行比较 1: 在 CAHALF=1 进行比较 复位值: 0x0
30:21	Res	Reserved 复位值: 0x0
20:0	r/w	PGxTRIGC: PWM 发生器 x 触发 C 注: 标准精度模式下, 位 16~20 保留, 无意义。高精度模式下, 位 0~4 固定为 0。 复位值: 0x0

17.5.2.14 PGx 死区寄存器 (PGxDT)

偏移地址: 0x0040

表 306 PGx 触发 C 寄存器

位	访问	描述
31:16	r/w	PGxDTH[15:0]: PWMxH 死区时间延迟 注: 标准精度模式下, 位 11~15 保留, 无意义。 复位值: 0x0
15:0	r/w	PGxDTL[15:0]: PWMxL 死区时间延迟 注: 标准精度模式下, 位 11~15 保留, 无意义。 复位值: 0x0

17.5.2.15 PGx 捕获寄存器 (PGxCAP)

偏移地址: 0x0044

表 307 PGx 捕获寄存器

位	访问	描述
31:21	Res	Reserved 复位值: 0x0
20:0	r/w	PGxCAP: PGx 时基捕获 当捕获事件有效时, 时基计数器值锁存于此寄存器, CAP 位置 1。对此寄存器的读取, 于此同时硬件自动清除 CAP 状态位。 注: 在高精度模式下, PGxCAP[4:0] 固定为 0; 在标准精度模式下, 位 16~20 保留, 无意义; PGxCAP[20:1] 只读, PGxCAP[0] 可读可写; 对位 0 写 1 可触发时基捕获。 复位值: 0x0

18 基准定时器 (BTM)

18.1 概述

基准定时器 (BaseTimer, BTM) 是一个可自动加载初值的 32 位向下计数器。而且该定时器可以配置为两个独立的 16 位向下计数器，并且也同样具备自动加载初值的功能。系统中包括两个 BTM: BTM0 和 BTM1。

18.2 主要功能

- 单个 BTM 可作为两个可自动加载初值的 16 位向下计数器，各自独立
- 单个可作为一个可自动加载初值的 32 位向下计数器
- 当计数器下溢时，自动加载初值至计数器，产生中断
- 计数器工作时钟源可选
- 可编程中断，两个计数器的中断使能可编程
- 支持输出触发事件至 EBUS 系统

18.3 模块框图

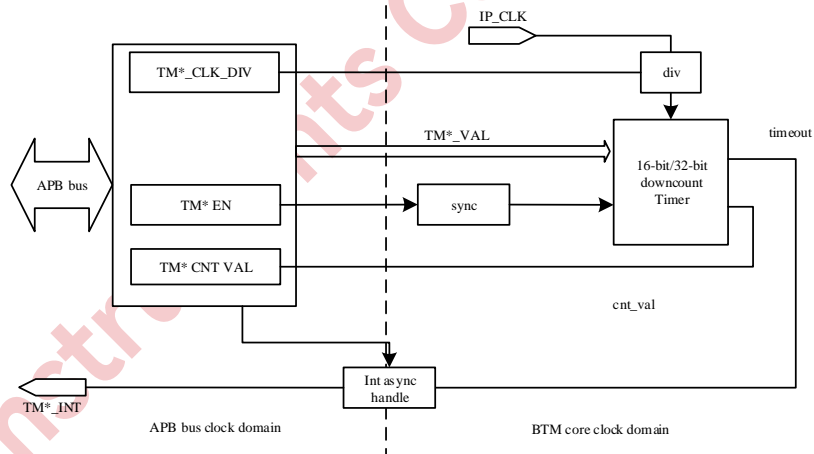


图 202 BTM 模块框图

18.4 功能描述

18.4.1 基本功能

当基准定时器使能之后，其内部计数器 (TM*_CNT) 基于模块分频时钟进行计数，分频时钟由 TM*_CLKDIV 控制分频所得。

当基准计时器使能后，计数器基于模块分频时钟进行向下计数，当计数器值减至 0 之后产生下溢，因此产生中断信号 (TM*_INT)，与此同时，硬件将加载寄存器 (TM*_LD) 的值自动加载到计数器中，计数器继续向下计数工作。

在计时器正常的使用过程中，软件可通过编写加载寄存器 (TM*_LD) 的值直接改变计数器的值，从而动态改变的定时器的计时长度。

基准计时器支持两种工作模式：

- 32 位计数器模式 (BTM_MOD=1)，基准计时器就是一个 32 位的计数器；
- 16 位计数器模式 (BTM_MOD=0)，基准计时器就是 2 个独立 16 位的计数器，可独立工作。

18.4.2 时钟源

计数器的时钟源可通过编写系统寄存器 0x400102A8 [22] 进行选择，具体选项如下：

- 0: LIRC_CLK
- 1: MCLK

18.4.3 可编程中断

计数器下溢并加载初值的时候，中断使能控制位 (TM*_IER) 可控制各个计数器的中断的产生与否。

18.4.4 输出触发事件至 EBUS 系统

计数器下溢并加载初值的时候，事件输出使能控制位 (TM*_EVNT_OE) 可控制计数器的触发事件的产生与否。

18.4.5 BTM 中断号

基准定时器的中断连接嵌套向量控制器 (NVIC) 的 IRQ 相关接口。为了启用该中断功能，需要按照以下流程进行操作：

- 使能 BTM 的中断使能位
- 使能 NVIC 中的 BTM IRQ 通道
- 配置 BTM 以产生中断

表 308 BTM 中断表

名称	中断号
BTM0_INT	22
BTM1_INT	23

18.5 寄存器

BTM0 寄存器的基地址为 0x4000_3000，BTM1 寄存器的基地址为 0x4000_3100，下表为 BTM 的各控制寄存器描述。

表 309 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	TM0 控制寄存器	32	r/w	0x00000000
0x04	TM0 加载寄存器	32	r/w	0xFFFFFFFF
0x08	TM0 计数器寄存器	32	r/w	0xFFFFFFFF
0x20	TM1 控制寄存器	32	r/w	0x00000000
0x24	TM1 加载寄存器	32	r/w	0x0000FFFF
0x28	TM1 计数器寄存器	32	r/w	0x0000FFFF
0x40	TM 状态寄存器	32	r/w	0x00000000

18.5.1 TM0 控制寄存器 (TM0_CTRL)

偏移地址：0x0000

表 310 BTM0 控制寄存器

位	访问	描述
31	r/w	BTM_MOD，基准计时器工作模式选择 0：支持两个独立的 16 位计数器 (TM0 & TM1) 1：支持一个 32 位计数器 (TM0) 复位值：0x0
30:17	Res	Reserved 复位值：0x0
16	ro	TM0_LD_BSY，TM0_LD 加载初值状态 0：无加载动作至 TM0 计数器 1：正在加载 TM0_LD 至 TM0 计数器 复位值：0x0
15:8	Res	Reserved 复位值：0x0
7	r/w	TM0_IE，TM0 计数器中断使能 0：关闭 1：开启 复位值：0x0

位	访问	描述
6	r/w	TM0_EVNT_OE, TM0 计数器事件输出使能 0: 关闭 1: 开启 复位值: 0x0
5:4	r/w	TM0_CLK_DIV[1:0]: TM0 计数器计数时钟分频选择 00: 计数器时钟是 IP 时钟的 16 分频 01: 计数器时钟是 IP 时钟的 32 分频 10: 计数器时钟是 IP 时钟的 64 分频 11: 计数器时钟是 IP 时钟的 256 分频 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	TM0_EN, TM0 计数器使能 0: TM0 计数器停止计数 1: TM0 计数器基于 IP 分频时钟向下计数 注: 1. 该使能位经由总线时钟域同步至 IP 时钟域方可生效。 2. 该位读回值是由 IP 时钟域同步到总线时钟域。 复位值: 0x0

18.5.2 TM0 加载寄存器 (TM0_LD)

偏移地址: 0x0004

表 311 TM0 加载寄存器

位	访问	描述
31:0	r/w	TM0_LOAD: TM0 计数器加载值 注: 1. 当 BTM_MOD=0 时, TM0 计数器是一个 16 位计数器, TM0_CNT[15:0] 从该值开始向下计数。若 TM0_CNT[15:0] 发生下溢, 硬件自动重新加载初值。 2. 当 BTM_MOD=1 时, TM0 计数器是一个 32 位计数器, TM0_CNT[31:0] 从该值开始向下计数。若 TM0_CNT[31:0] 发生下溢, 硬件自动重新加载初值。 3. 更新该寄存器的同时也会同步更新到 TM0_CNT 中 复位值: 0xFFFF

18.5.3 TM0 计数器寄存器 (TM0_CNT)

偏移地址: 0x0008

表 312 TM0 计数器寄存器

位	访问	描述
31:0	ro	<p>TM0_CNT, TM0 计数器当下计数值</p> <p>注:</p> <ol style="list-style-type: none"> 1.在 TM0_EN 开启下, TM0 计数器基于 IP 分频时钟向下计数。 2.当 BTM_MOD=0, TM0 计数器是一个 16 位计数器。TM0 计数器每次下溢时都会产生相应的中断信号 (TM0_INT), 同时又重新开始从初值开始向下计数。 3.当 BTM_MOD=1, TM0 计数器是一个 32 位计数器。TM0 计数器每次下溢时都会产生相应的中断信号 (TM0_INT), 同时又重新开始从初值开始向下计数。 4.该寄存器反映的是 TM0 计数器的值, 因为存在不同时钟域的跨度, 会存在 2~3 总线时钟周期的时间差 <p>复位值: 0xFFFF</p>

18.5.4 TM1 控制寄存器 (TM1_CTRL)

偏移地址: 0x0020

表 313 TM1 控制寄存器

位	访问	描述
31:17	Res	<p>Reserved</p> <p>复位值: 0x0</p>
16	ro	<p>TM1_LD_BSY, TM1_LD 加载初值状态</p> <p>0: 无加载动作至 TM1 计数器</p> <p>1: 正在加载 TM1_LD 至 TM1 计数器</p> <p>复位值: 0x0</p>
15:8	Res	<p>Reserved</p> <p>复位值: 0x0</p>
7	r/w	<p>TM1_IE, TM1 计数器中断使能</p> <p>0: 关闭</p> <p>1: 开启</p> <p>复位值: 0x0</p>
6	r/w	<p>TM1_EVNT_OE, TM1 计数器事件输出使能</p> <p>0: 关闭</p> <p>1: 开启</p> <p>复位值: 0x0</p>
5:4	r/w	<p>TM1_CLK_DIV[1:0]: TM1 计数器计数时钟分频选择</p> <p>00: 计数器时钟是 IP 时钟的 16 分频</p> <p>01: 计数器时钟是 IP 时钟的 32 分频</p> <p>10: 计数器时钟是 IP 时钟的 64 分频</p> <p>11: 计数器时钟是 IP 时钟的 256 分频</p> <p>复位值: 0x0</p>

位	访问	描述
3:1	Res	Reserved 复位值: 0x0
0	r/w	TM1_EN, TM1 计数器使能 0: TM1 计数器停止计数 1: TM1 计数器基于 IP 分频时钟向下计数 注: 1.该使能位经由总线时钟域同步至 IP 时钟域方可生效。 2.该位读回值是由 IP 时钟域同步到总线时钟域。 复位值: 0x0

18.5.5 TM1 加载寄存器 (TM1_LD)

偏移地址: 0x0024

表 314 TM1 加载寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	TM1_LOAD: TM1 计数器加载值 注: 1.该位只在基准计时器工作模式为 16 位计数器模式下才有意义 (BTM_MOD=1'b0) 2.TM1_CNT[15:0] 从该值开始向下计数。若 TM1_CNT[15:0] 发生下溢, 硬件自动重新加载初值 3.更新该寄存器的同时也会同步更新到 TM1_CNT 中 复位值: 0xFFFF

18.5.6 TM1 计数器寄存器 (TM1_CNT)

偏移地址: 0x0028

表 315 TM1 计数器寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	TM1_CNT, TM1 计数器当下计数值 注: 1.在 TM1_EN 开启下, TM1 计数器基于 IP 分频时钟向下计数 2.当 BTM_MOD=0, TM1 计数器是一个 16 位计数器。TM1 计数器每次下溢时都会产生相应的中断信号 (TM1_INT), 同时又重新开始从初值开始向下计数。然而 BTM_MOD=1, TM1 计数器不可启用。 3.该寄存器反映的是 TM1 计数器的值, 因为存在不同时钟域的跨度, 会存在 2~3 总线时钟周期的时间差 复位值: 0x0

18.5.7 TM 状态寄存器 (TM_ST)

偏移地址: 0x0040

表 316 TM 状态寄存器

位	访问	描述
31:17	Res	Reserved 复位值: 0x0
16	r/w	TM1_INT, TM1 计数器中断状态 0: 无中断 1: 有中断发生 注: 1.该中断在 TM1 计数器下溢并从 TM1_LD 加载初值的时刻产生。 2.该位写 1 清零。 复位值: 0x0
15:1	Res	Reserved 复位值: 0x0
0	r/w	TM0_INT, TM0 计数器中断状态 0: 无中断 1: 有中断发生 注: 1.该中断在 TM0 计数器下溢并从 TM0_LD 加载初值的时刻产生。 2.该位写 1 清零。 复位值: 0x0

19 看门狗 (WDT)

19.1 概述

看门狗 (Watchdog Timer, WDT) 是一个 24 位向下计数器，可提供预警中断和系统复位功能，方便于维持系统稳定，一旦软件出现异常时，系统可通过复位恢复到正常运行状态。

19.2 主要功能

- 24 位向下计数器
- 两种运行模式
 - 普通模式
 - 窗口模式
- 有条件复位
 - 计数器下溢
 - 在窗口期外喂狗
- 支持可编程预警中断产生机制
- 预警中断 (EWI)：预警阈值可编程
- 配置寄存器安全锁功能
 - WDT_KR 可关闭其他配置寄存器写操作
 - 输入正确的解锁码可以开启其他配置寄存器写操作功能
- 时钟源可编程选择
- WDT 可在系统所有功耗模式下运行
- 调试模式：在 CPU 调试模式下 WDT 可暂停工作

19.3 模块框图

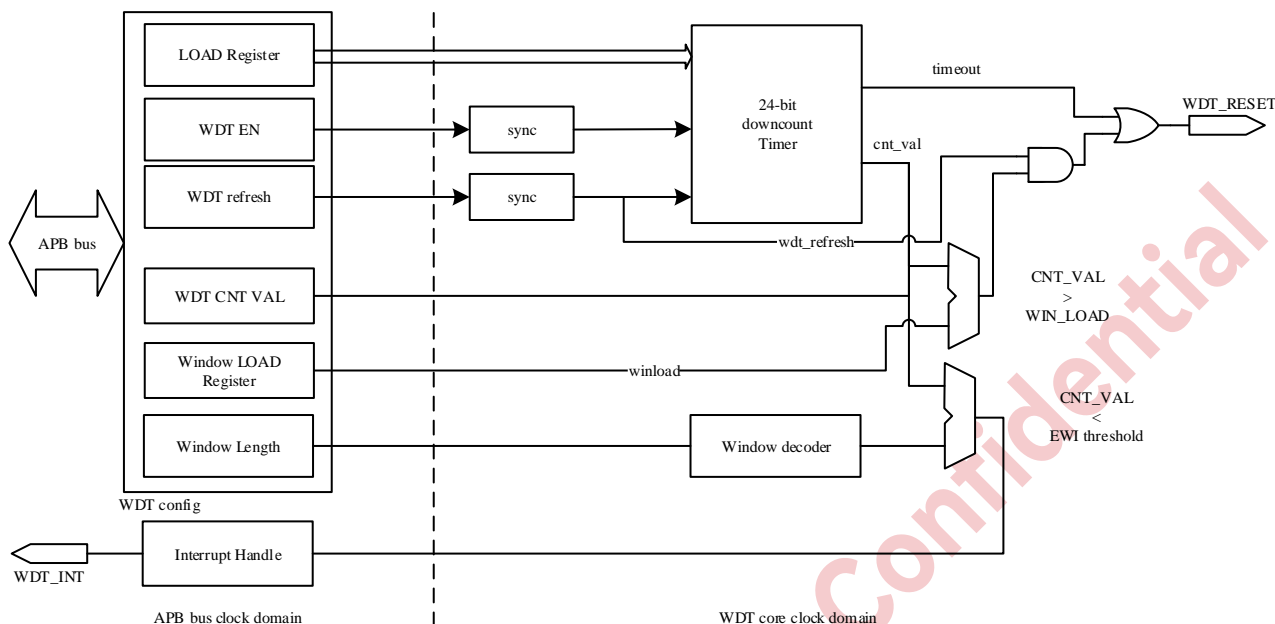


图 203 WDT 模块框图

19.4 功能描述

看门狗模块基于一个 24 位向下计数器实现，在每个 IP 时钟上升沿到来后减一，初始值源于加载寄存器 (WDT_LOAD)，通过比较计数器与可编程的数值从而产生常规中断信号 (WDT_INT)。在计数器下溢的时候，看门狗立刻产生复位请求信号 (WDT_RST)。具体相关功能详见以下章节。

19.4.1 预警中断功能

预警中断功能可用于在复位信号发生之前需要完成特别安全功能操作或者数据记录类似相关的应用场景。该中断功能可以通过置位 WDT_CTRL 中的 IE 进行使能，当计数器向下计数至预设值 (通过 EWI_THR 设定) 时，预警中断便产生，随后在复位生效之前进入相关的中断服务程序执行相关的操作。

在一些应用中，EWI 中断可用于软件系统检查、log 记录等，在这种情况下，相应的中断服务程序 (ISR) 应先重新加载 WDT 计数器以避免 WDT 复位而后才触发所需的操作。其中更新 WDT_LOAD 和喂狗操作并不清除中断状态位，可以通过在相应的状态位写 1 实现清零，并且重新加载初值至计数器。

WDT 的中断向量编号为 0。

19.4.2 工作模式

WDT 可通过控制位 WDT_EN 实现关闭及开启。在工作中支持以下四种工作模式：

- {WM_EN, IE} == 00, 普通模式 1, 不带预警中断功能
- {WM_EN, IE} == 01, 普通模式 2, 带预警中断功能
- {WM_EN, IE} == 10, 窗口模式 1, 不带预警中断功能
- {WM_EN, IE} == 11, 窗口模式 2, 带预警中断功能

19.4.3 普通模式 1 (不带预警功能)

在普通模式 1 中, 在加载寄存器 (WDT_LOAD) 中设定超时时间。一旦 WDT 使能, 软件可在超时前的任何时刻完成 WDT 计数器的刷新喂狗操作。

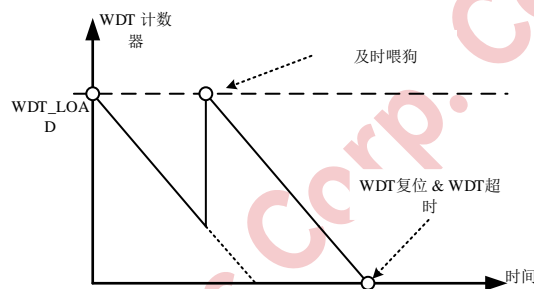


图 204 WDT 普通模式 1

19.4.4 普通模式 2 (带预警功能)

默认情况下, 一旦计数器超时, WDT 便会发起一个系统复位, 而预警中断功能是关闭的。如需要预警中断功能, 则需通过使能控制寄存器的里边 IE 控制位。并且通过设定 EWI_THR 选择好预警阈值, 详见寄存器描述。当计数器向下计数至预警阈值时, 预警中断便会生成。软件可以通过写 1 清除中断状态, 同时触发硬件自动加载初值寄存器至计数器。

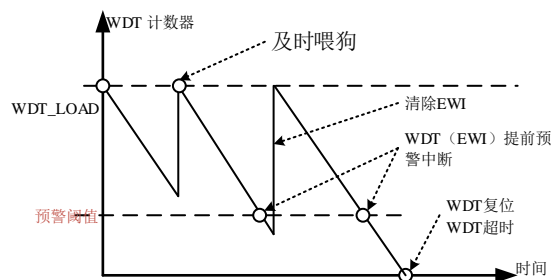


图 205 WDT 普通模式 2

19.4.5 窗口模式 (不带预警功能)

在窗口模式下，WDT 定义了两个超时周期：1. 窗口关闭期，计数器从加载值计数 (WDT_LOAD) 至窗口加载值 (WDT_WIN_LOAD)，设定为此期间不能执行喂狗操作，一旦喂狗则触发 WDT 系统复位；2. 窗口开启期，计数器从窗口加载值 (WDT_WIN_LOAD) 计数至超时之前，此计数周期允许执行喂狗操作。详见下图所示。在应用中，WDT_WIN_LOAD 的设定应小于 WDT_LOAD，具体窗口大小应结合应用需求进行设定。

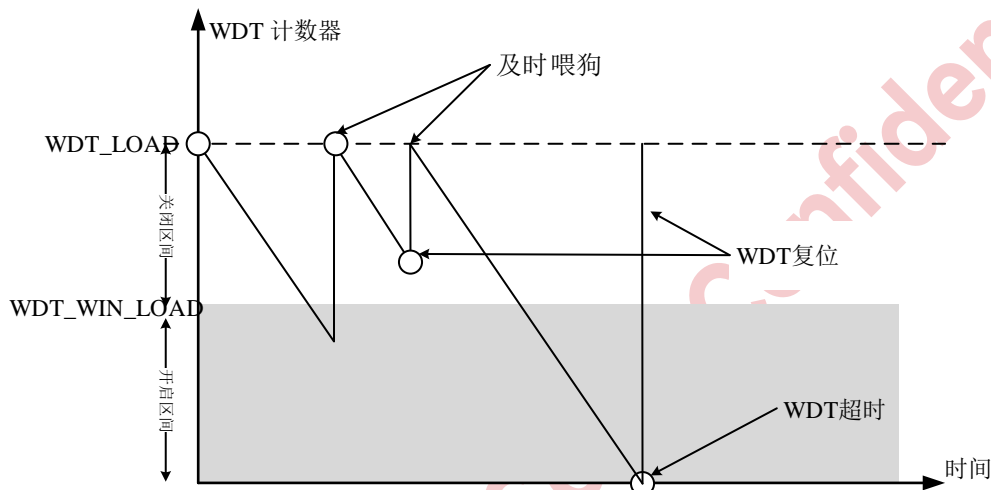


图 206 WDT 窗口模式 1

19.4.6 窗口模式 (带预警功能)

在窗口模式下可通过使能控制寄存器的 IE 控制位开启预警中断功能，并在 EW1_THR 选择好预警阈值，相关的工作机制如下图所示。预警工作方式与普通模式 2 一致。

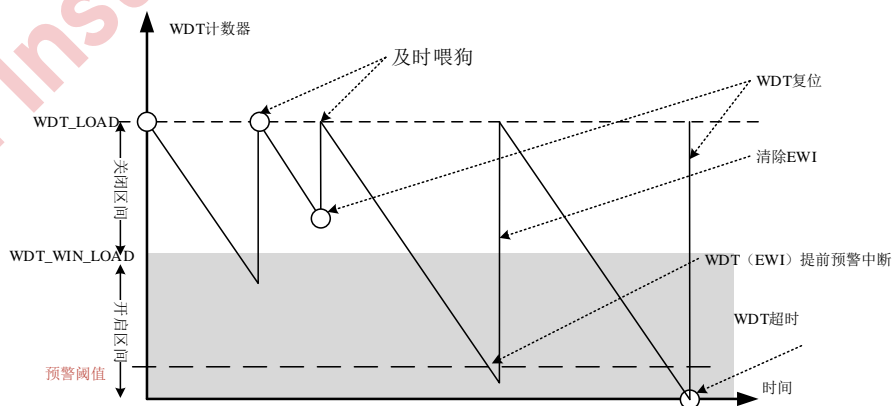


图 207 WDT 窗口模式 2

19.4.7 寄存器安全锁功能

操作 WDT_KR 可以用于关闭对于 WDT 相关寄存器 (包括 WDT_CTRL, WDT_LOAD、WDT_WIN_LOAD) 的写操作权限。此举可以避免软件误操作关闭看门狗功能。往 WDT_KR 写入正确的密钥可以实现上述写操作权限的解锁, 写入其他值则可以实现权限的关闭。相应的权限状态可通过读取 WDT_KR 位置获取。所以在操作 WDT 相关寄存器之前, 用户需输入正确密钥解锁写权限, 完成操作之后需要关闭写权限。

19.4.8 硬件看门狗

通过在选项字节 (Option Byte) 中烧写好 WDT_EN 和 WDT_LOAD 的预配置, 芯片可以在上电过程中自动加载预配置值至看门狗的相应的寄存器中, 看门狗根据配置进行工作。

19.4.9 调试模式

当 CPU 进入调试模式后, 看门狗可根据控制位 DBG_WDT_STOP (0x400104C0[0]) 的信息决定继续运行还是停止。

- DBG_WDT_STOP=0, 继续运行
- DBG_WDT_STOP=1, 停止

19.4.10 时钟源

看门狗有以下时钟源可供选择, 配置寄存器为 WDT_CLK_SEL (0x400102A8[20]), 如下所示, 具体操作详见系统寄存器。

- 0: LIRC
- 1: MCLK

19.5 寄存器

WDT 寄存器的基地址为 0x4000_2C00, 下表为 WDT 的各控制寄存器描述。

表 317 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	WDT 控制寄存器	32	r/w	0x00000000
0x04	WDT 加载寄存器	32	r/w	0x00000000
0x08	WDT 窗口加载寄存器	32	r/w	0x00FFFFFF
0x0C	WDT 状态寄存器	32	r/w	0x00000000
0x10	WDT 计数器寄存器	32	ro	0x00FFFFFF
0x20	WDT 密钥寄存器	32	r/w	0x00000001

19.5.1 WDT 控制寄存器 (WDT_CTRL)

偏移地址: 0x0000

表 318 WDT 控制寄存器

位	访问	描述
31	r/w	WDT_EN: 使能看门狗模块 0: 关闭 1: 开启 复位值: 取决于存储于内嵌 FLASH 中的芯片控制位 (Option Byte)
30:3	Res	Reserved 复位值: 0x0
5:3	r/w	EWI_THR: EWI 预警值设定 000: 0x80 001: 0x100 010: 0x200 011: 0x400 100: 0x800 101: 0x1000 110: 0x2000 111: 0x4000 复位值: 0x0
2	r	WIN_CLOSE_FLG: 窗口关闭标记 0: WDT 计数器值位于窗口开启区间 (WDT_CNT < WDT_WIN_LOAD) 或者窗口模式关闭 1: WDT 计数器值位于窗口关闭区间 注: 如果在窗口关闭区间进行喂狗操作则会触发 WDT 复位事件 复位值: 0x0
1	r/w	WM_EN: WDT 窗口模式使能控制 0: 关闭 1: 使能 注: 该位只能软件置位, 复位硬件清零 复位值: 0x0
0	r/w	IE: 中断使能控制 0: 关闭 1: 使能 注: 该位只能软件置位, 复位硬件清零 复位值: 0x0

19.5.2 WDT 加载寄存器 (WDT_LOAD)

偏移地址: 0x0004

表 319 WDT 加载寄存器

位	访问	描述
31:24	Res	Reserved 复位值: 0x0
23:0	r/w	WDT_LOAD[23:0], 看门狗加载值 该寄存器是 WDT 计数器向下计数的初值。 注: 1.当看门狗开启, 计数器会自动从该寄存器加载初值 2.当发生看门狗复位事件的同时, 计数器会自动重新从该寄存器加载初值 3.更新该寄存器, 内部的计数器也会同步更新 4.该寄存器最小值为 1 复位值: 取决于存储于内嵌 FLASH 中的芯片控制位 (Option Byte)

19.5.3 WDT 窗口加载寄存器 (WDT_WIN_LOAD)

偏移地址: 0x0008

表 320 WDT 窗口加载寄存器

位	访问	描述
31:24	Res	Reserved 复位值: 0x0
23:0	r/w	WDT window value: WDT 窗口加载寄存器 复位值: 0xFFFFFFFF

WDT_WIN_LOAD 寄存器值设定的是可进行 WDT 喂狗操作时计数器值的上限。正常情况下, 在计数器值仍然大于 WDT_WIN_LOAD 时, 有效的喂狗动作会导致一个看门狗事件发生, 也就是如前所述在窗口关闭期的喂狗动作会触发看门狗事件。通常情况下, WDT_WIN_LOAD 应小于 WDT_LOAD, 才能保证有效的窗口存在。

19.5.4 WDT 状态寄存器 (WDT_ST)

偏移地址: 0x000C

表 321 WDT 状态寄存器

位	访问	描述
31	ro	WDT_EN_SYNC: 看门狗启用状态 0: 关闭 1: 开启 复位值: 取决于存储于内嵌 FLASH 中的芯片控制位 (Option Byte)
30:17	Res	Reserved 复位值: 0x0
16	ro	WDT_FEED_BSY: 喂狗状态 0: 无喂狗动作 1: 喂狗正在进行中 复位值: 0x0
15:1	Res	Reserved 复位值: 0x0
0	r/w	WDT_INT, 看门狗中断状态 0: 空 1: 发生提前预警 注: 该位写 1 清零, 并且触发硬件刷新动作重新加载 WDT_LOAD 初值至计数器。 复位值: 0x0

19.5.5 WDT 计数器寄存器 (WDT_VAL)

偏移地址: 0x0010

表 322 WDT 计数器寄存器

位	访问	描述
31:24	Res	Reserved
23:3	ro	WDT_CNT[23:3], 内部计数器值观测 注: 因总线和 IP 分属于不同的时钟域, 该观测值是内部计数器数值的定期更新。 复位值: 0x1FFFFFFF
2:0	Res	Reserved

19.5.6 WDT 密钥寄存器 (WDT_KR)

对于该寄存器的操作可以用于关闭对于系统其他寄存器的写操作（包括 WDT_CTRL，WDT_LOAD，WDT_WIN_LOAD）。此举可以避免软件误操作关闭看门狗功能。此外往该寄存器写 0xAAAA 可以刷新 WDT 计数器，也就是喂狗。

偏移地址：0x0020

表 323 WDT 密钥寄存器 (写操作)

位	访问	描述
31:0	wo	寄存器 (WDT_CTRL、WDT_LOAD、WDT_WIN_LOAD) 写控制 写密钥 0x2EE1D879 可开启写许可 写其他值可关闭写许可 WDT 喂狗操作刷新计数器 → 写关键字 0xAAAA 触发喂狗动作

表 324 WDT 密钥寄存器 (读操作)

位	访问	描述
31:1	Res	Reserved 复位值：0x0
0	ro	寄存器 (WDT_CTRL、WDT_LOAD、WDT_WIN_LOAD) 写控制状态 0：允许写操作 1：禁止写操作 复位值：0x1

20 串行通信接口 (SCI)

20.1 概述

KPM32R14TX 包含三个串行通信接口 (SCI)，串行通信接口可以实现异步及同步通信，其中，异步通信主要包括 UART 及 Smart Card (ISO 7816-3 IC 卡标准)，同步通信接口支持 SPI 或者简易 IIC 协议。

20.2 主要功能

SPI 模式 (SCK、MISO、MOSI、CS)

- 数据长度 8bit
- 发送/接收数据相位控制
- 输入/输出时钟相位控制
- CS 控制可配置使能
- MSB/LSB 选择
- 发送/接收数据的电平设置及反相选择
- 主/从选择
- 可配置波特率
- 传输结束中断/缓冲器空中断
- 溢出错误检测中断

UART 模式 (RX、TX、CTS、RTS)

- 数据长度 7/8/9bit
- MSB/LSB 选择
- 发送/接收数据的电平设置及反相选择
- 奇偶检验位发送/校验功能
- 1bit/2bit 停止位
- 传输结束中断/缓冲器空中断
- 帧错误，奇偶校验错误，以及溢出错误检测中断

- 多处理器通信网络
- RTS/CTS 硬件流控

I²C 模式 (SCL、SDA)

- 主发送/主接收 (仅限单一节点主功能)
- 起始/重新起始条件及停止条件硬件产生
- 支持时钟同步
- 可配置输出保持
- ACK/NACK 输出功能以及 ACK/NACK 检测功能
- 数据长度 8bit
- 传输结束中断/缓冲器空中断
- ACK 错误, 溢出错误检测中断

SYNC 模式 (SCK、TX、RX、CTS/RTS)

- 数据长度 8/9bit
- MSB/LSB 选择
- 发送/接收数据相位控制
- 输入/输出时钟相位控制
- 发送/接收数据的电平设置及反相选择
- 传输结束中断/缓冲器空中断
- RTS/CTS 硬件流控
- 溢出错误检测中断

Smart Card 模式 (RX、TX)

- 数据长度 8
- 数据正向约定/反向约定
- 奇偶检验位发送/校验功能

- 自动重新发送
- 传输结束中断/缓冲器空中断
- 奇偶校验错误，DE 错误检测中断，以及溢出错误检测中断

20.3 模块框图

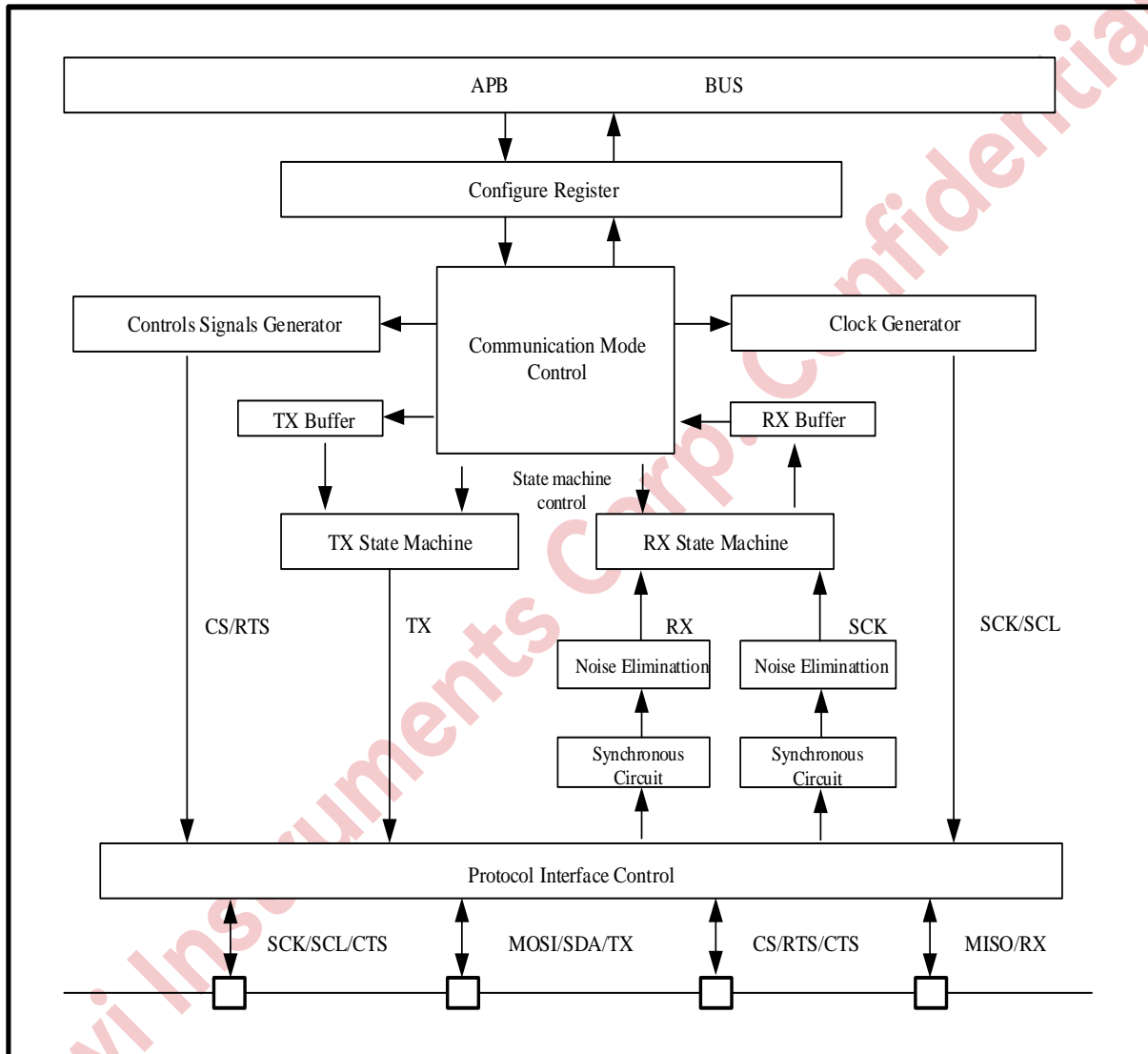


图 208 串行通信接口模块框图

20.4 功能描述

20.4.1 工作模式

串行通信接口可配置为以下 5 种通信模式：

- SPI 模式
- UART 模式
- 简易 I²C 模式
- SYNC 模式
- Smart Card 模式

每种模式下，协议相关性的差异性配置在各自的模式寄存器中，相同的部分在模式控制寄存器中。

20.4.2 接口功能

每个串行通信接口都有 4 个接口与外部设备进行通信，配置为不同模式时，每个接口的功能会相应进行改变，下表为在不同模式下接口功能的复用关系。

表 325 接口功能复用

通信模式		SCK/SCL/CTS	MOSI/SDA/TX	MISO/RX	CS/RTS/CTS
SPI	功能	SCK	MOSI	MISO	CS
	方向	I/O	I/O	I/O	I/O
UART	功能	CTS	TX	RX	RTS
	方向	Input	Output	Input	Output
IIC	功能	SCL	SDA		
	方向	Output	I/O		
SYNC	功能	SCK	TX	RX	CTS/RTS
	方向	I/O	Output	Input	I/O
Smart Card	功能	SCK	TX	RX	
	方向	Output	I/O	I/O	

20.4.3 SPI 模式

这是一种采用四条线路的时钟同步式通信功能：串行时钟 (SCK)，串行数据 (MOSI 和 MISO) 和片选 (CS) 线路。

在该模式下，主设备产生并输出串行时钟到从设备，通过片选控制通信有效性，单工，半双工以及双工通信模式都可以在此种模式下应用。因此，在 SPI 模式下，共有如下六种类型的通信操作。

20.4.3.1 主发送

主发送是指串行通信接口作为主设备，输出传输时钟 (SCK)，并将数据发送到外部器件；主发送支持单笔发送模式和连续发送模式，通过状态寄存器中的发送完成标志或发送缓冲器空标志来实现。具体的配置参考编程指南中的主发送部分。

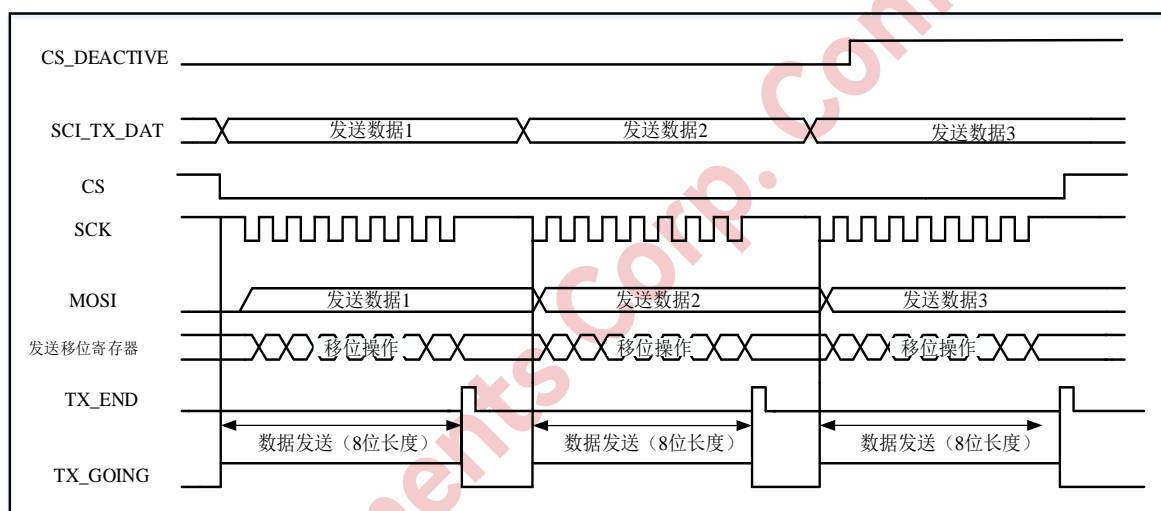


图 209 主发送 (单笔模式) 时序图

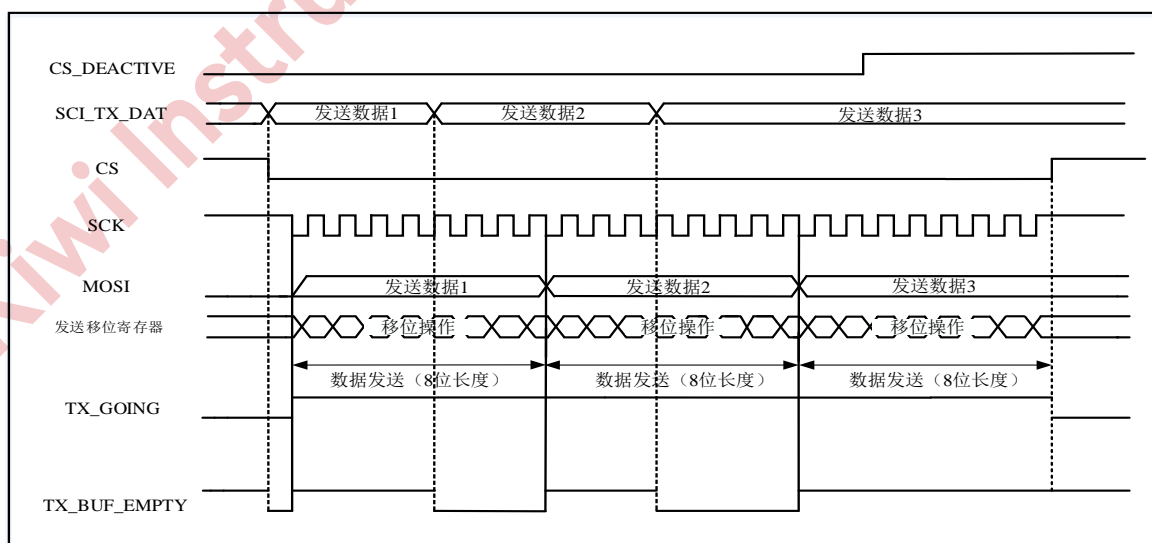


图 210 主发送 (连续模式) 时序图

20.4.3.2 主接收

主接收是指串行通信接口作为主设备，输出传输时钟 (SCK)，并从外部器件接收数据；主接收支持单笔接收模式和连续接收模式，通过状态寄存器中的接收完成标志或发送缓冲器空标志来实现。具体的配置参考编程指南中的主接收部分。

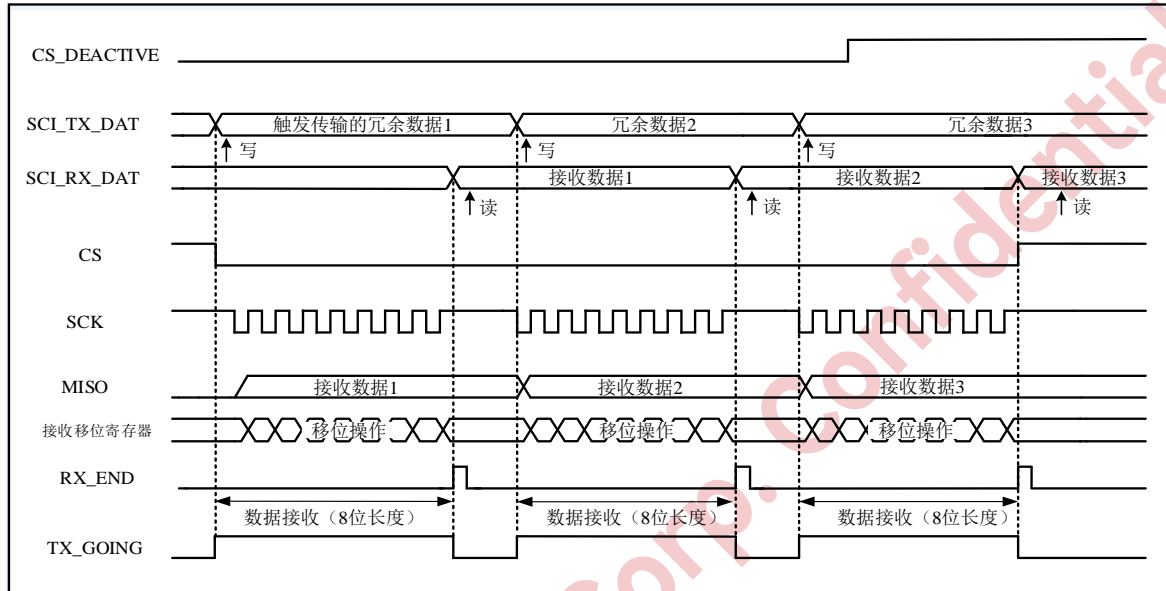


图 211 主接收（单次模式）时序图

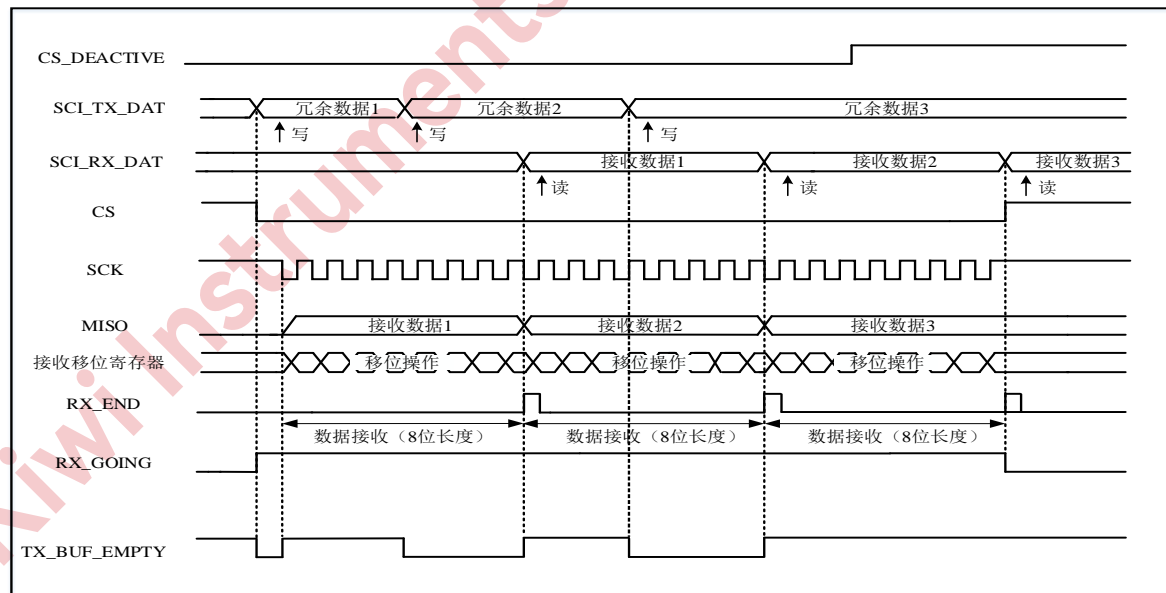


图 212 主接收（连续模式）时序图

20.4.3.3 主发送接收

主发送接收是指串行通信接口作为主设备，输出传输时钟（SCK），并从外部器件接收数据，同时将数据写入外部器件；主发送接收支持单笔模式和连续模式，通过状态寄存器中的完成状态标志或者缓冲器空满标志来实现。具体的配置参考编程指南中的主发送接收部分。

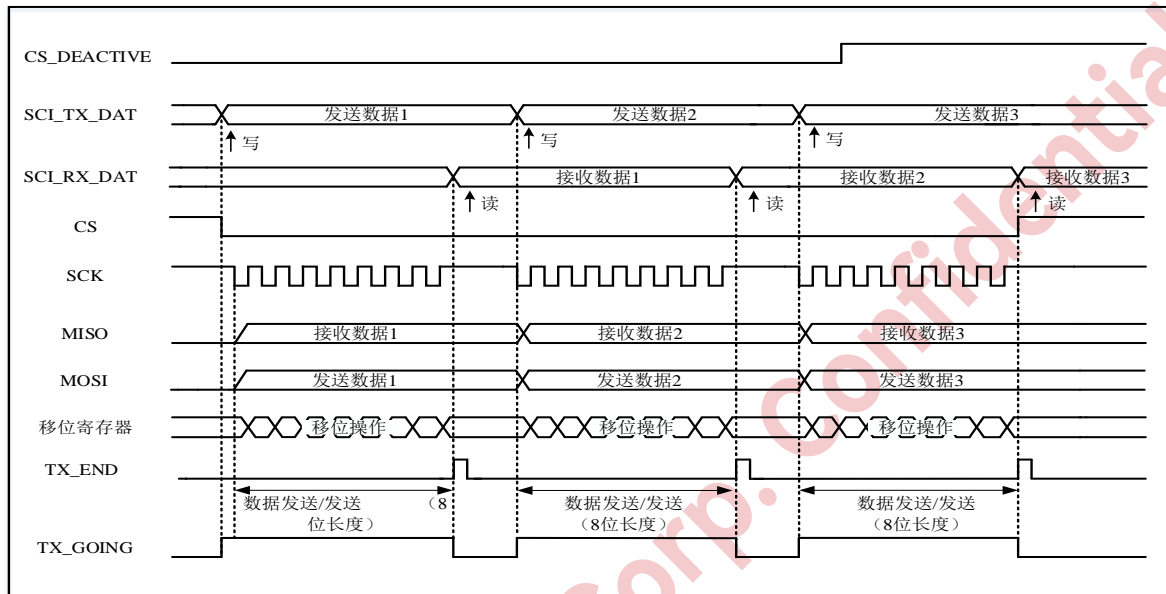


图 213 主发送接收（单笔模式）时序图

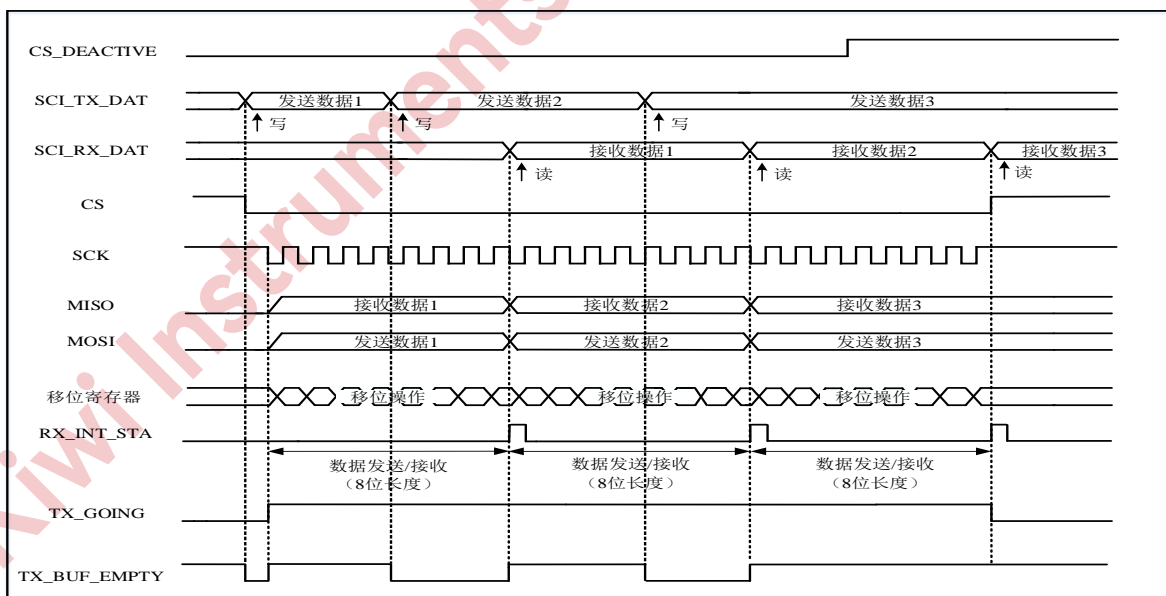


图 214 主发送接收（连续模式）时序图

20.4.3.4 从发送

从发送是指串行通信接口单元作为从设备，接收来自外部器件输入的传输时钟 (SCK)，并将数据发送到外部器件；从发送支持单笔发送模式和连续发送模式，通过中断状态寄存器完成状态标志和缓冲器空状态标志来实现。具体的配置参考编程指南中的从发送部分。

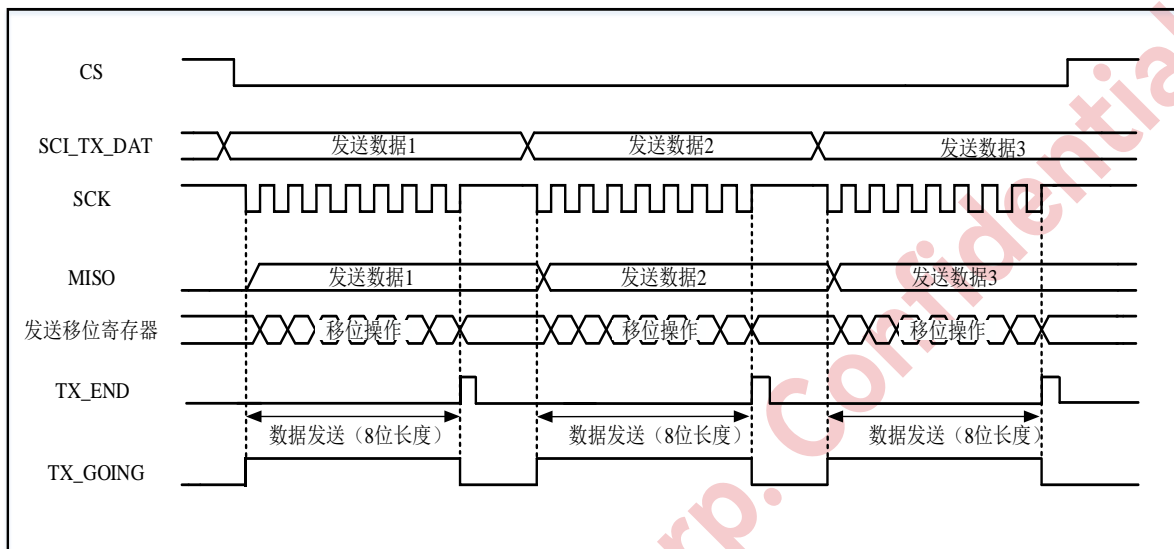


图 215 从发送（单笔模式）时序图

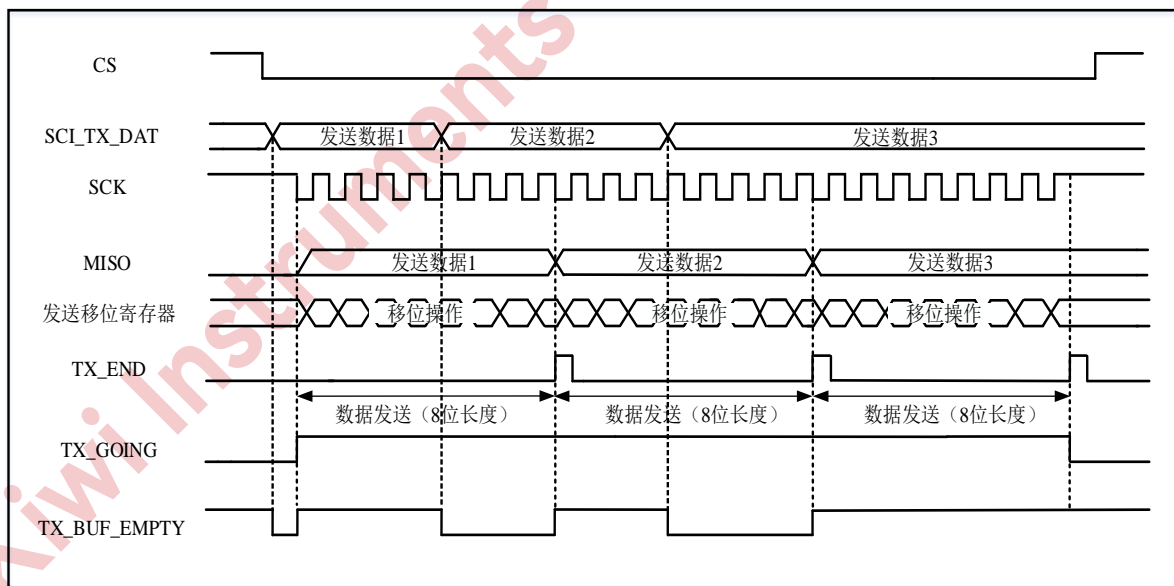


图 216 从发送（连续模式）时序图

20.4.3.5 从接收

从接收是指串行通信接口作为从设备，接收来自外部器件输入的传输时钟 (SCK)，并从外部器件接收数据；从接收仅支持单笔接收模式。具体的配置参考编程指南中的从接收部分。

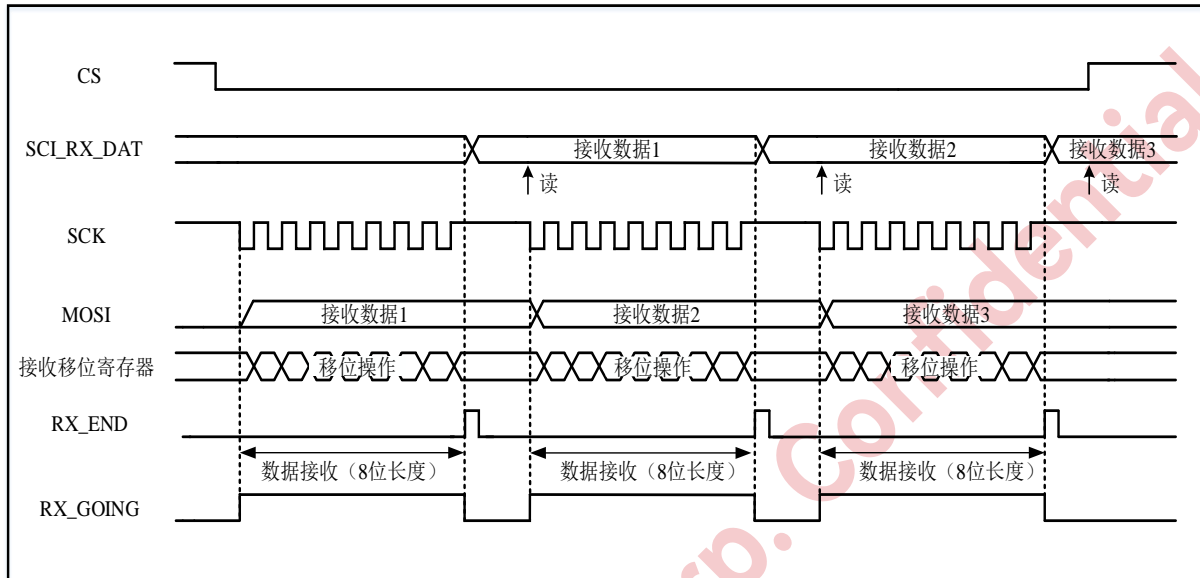


图 217 从接收（单笔模式）时序图

20.4.3.6 从发送接收

从发送接收是指串行通信接口作为从设备,接收来自外部器件输入的传输时钟 (SCK), 并从外部器件接收数据, 同时将数据写入外部器件; 从发送接收支持单笔模式和连续模式, 通过中断状态寄存器的完成状态标志及缓冲器空状态标志实现。具体的配置参考编程指南中的主发送接收部分。

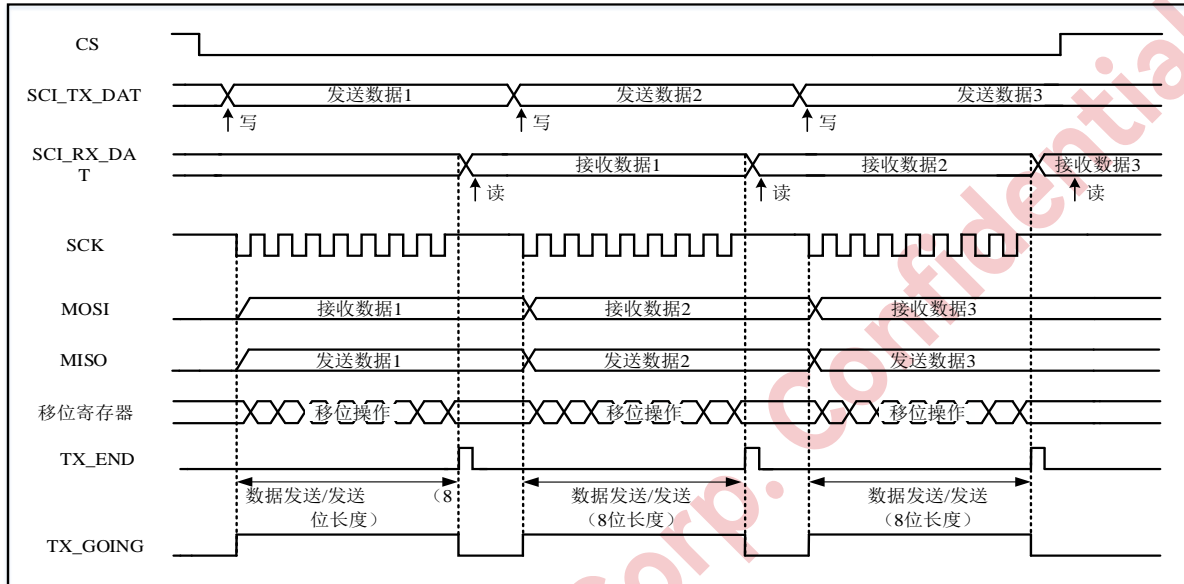


图 218 从发送接收 (单笔模式) 时序图

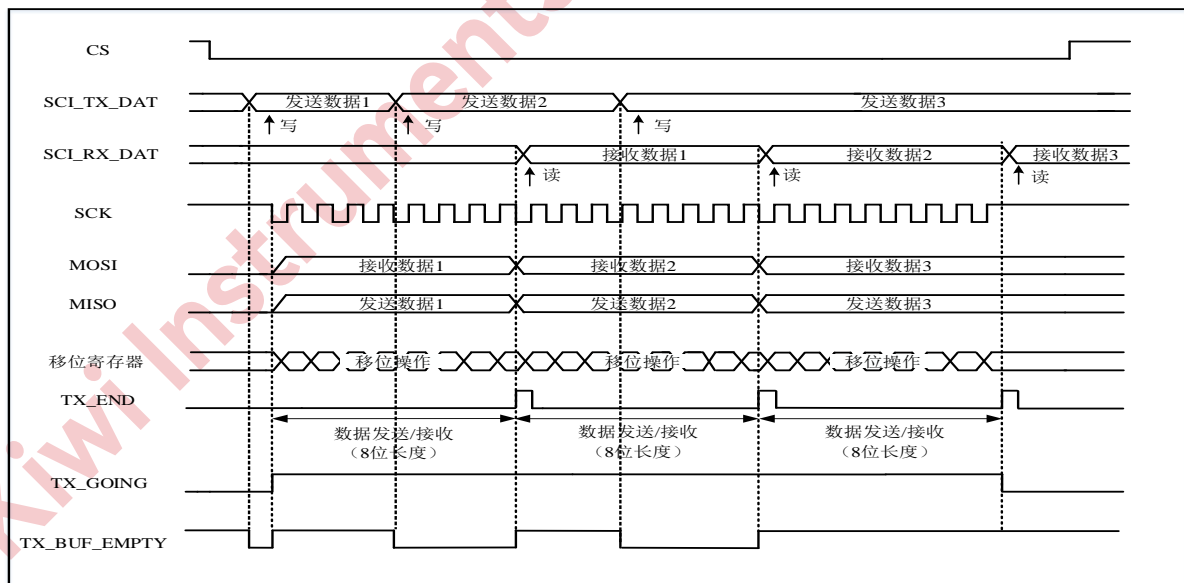


图 219 从发送接收 (连续模式) 时序图

20.4.3.7 传输时钟频率

SPI 模式通信的传输时钟频率计算表达式如下：

- 主设备：

- SCI 时钟频率 = 系统时钟频率/2

$$OP_CLK_DIV = (SCI_BAUD.CLK_DVI1+1) * (SCI_BAUD.CLK_DVI2+1)$$

$$传输频率 = (SCI 时钟频率 / 2^{CLK_PRS_SEL}) \div OP_CLK_DIV \div 2 \text{ (Hz)}$$

- 从设备：

- 传输频率 = 由外部主设备提供的串行时钟 (SCK) 频率 (Hz)

当串行单元作为从设备时，最大支持传输时钟频率为： $(SCI 时钟频率 / 2^{CLK_PRSx_SEL}) \div 6$ 。

20.4.4 UART 通信模式

串行通信接口可配置为 UART 异步通信模式，该模式下采用两条线路：串行数据传送 (TX) 和串行数据接收 (RX) 线路。利用这两条通信线路，各数据帧（由一个起始位、数据、奇偶校验位和停止位构成）在单片机与其他通信方之间（以内部波特率）异步传送。

20.4.4.1 UART 发送

UART 发送是将数据发送到外部器件；UART 发送支持单笔发送模式和连续发送模式，通过状态寄存器中的发送完成标志或发送缓冲器空标志来实现。具体的配置参考编程指南中的 UART 发送部分。

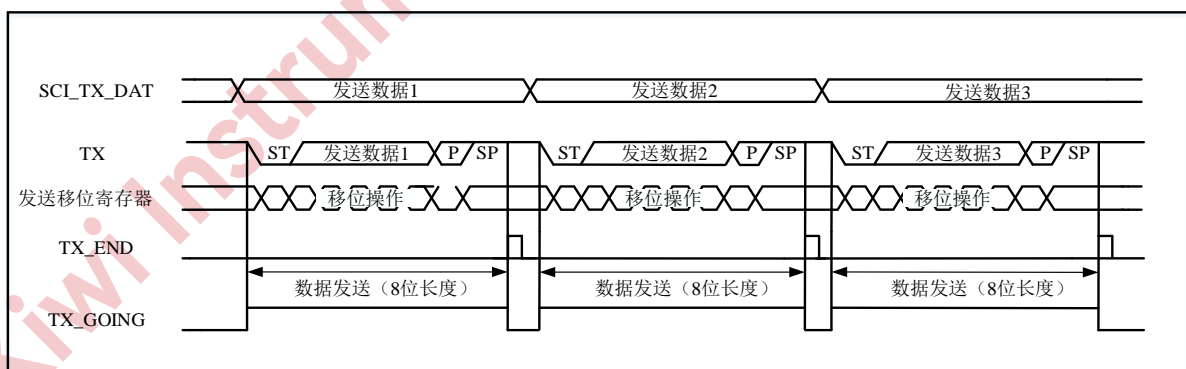


图 220 UART 发送 (单笔模式) 时序图

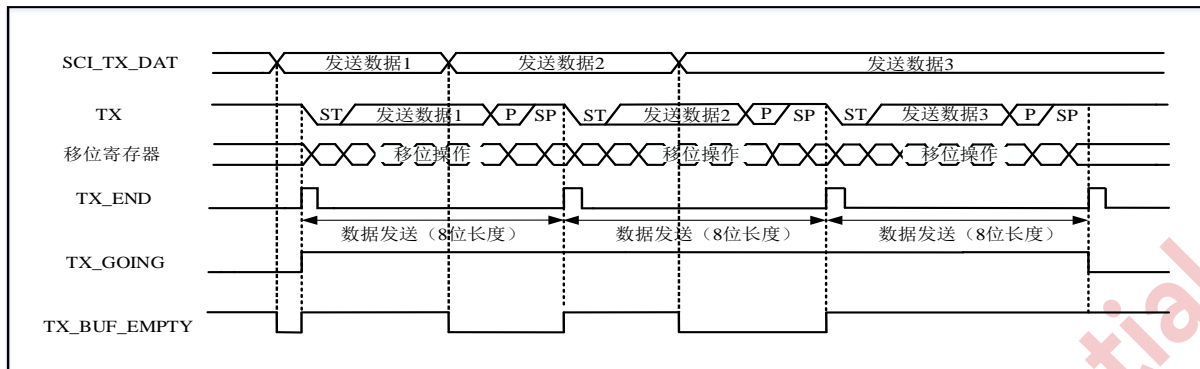


图 221 UART 发送 (连续模式) 时序图

20.4.4.2 UART 接收

UART 接收是接收外部器件发送的数据；UART 接收仅支持单笔接收模式，具体的配置参考编程指南中的 UART 接收部分。

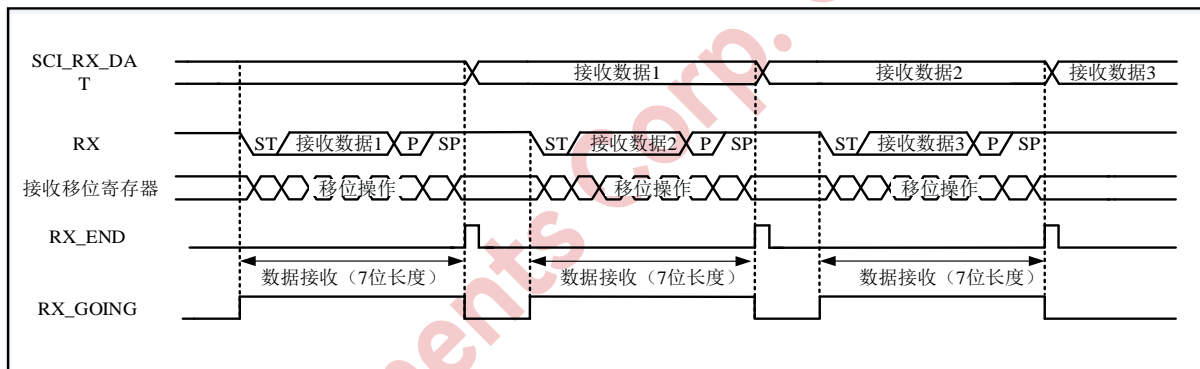


图 222 UART 接收 (单笔模式) 时序图

20.4.4.3 硬件流控

在 UART 模式下，输入信号 CTS 可以作为数据发送控制信号，通过 UART 模式寄存器中的 CTS_EN 位来使能该功能。当 CTS 电平由高变低后，串行通信接口开始进行数据发送到外部设备。

在数据发送过程中，将 CTS 电平拉高后，并不会影响当前正在发送的数据，当前帧数据发送完成后，串行通信接口停止发送动作。

同时，输出信号 RTS 可以作为数据接收控制信号通过 UART 模式寄存器中的 RTS_EN 位来使能该功能。当中断状态寄存器中的 RX_BUF_FULL 状态为低时，串行通信接口可以进行数据接收，此时 RTS 信号电平会被拉低，反之，则 RTS 会被置高，直到软件读取数据接收寄存器。

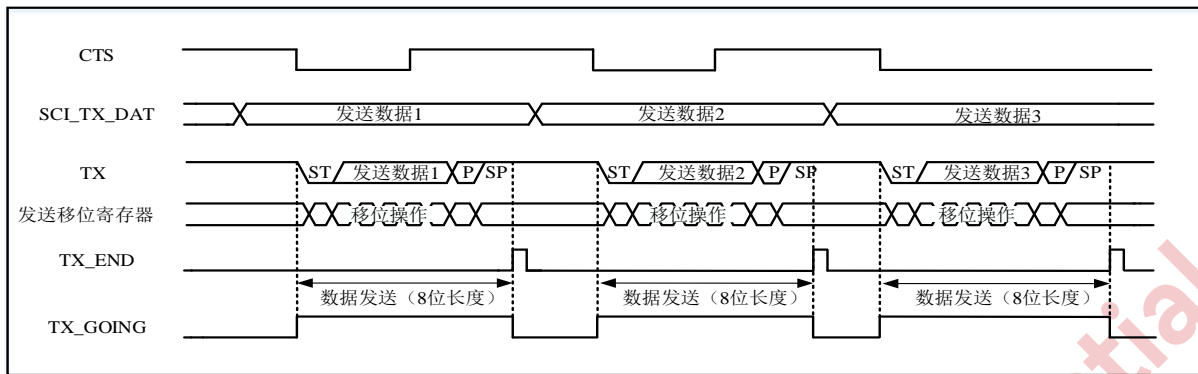


图 223 UART 发送 (使能 CTS 控制)

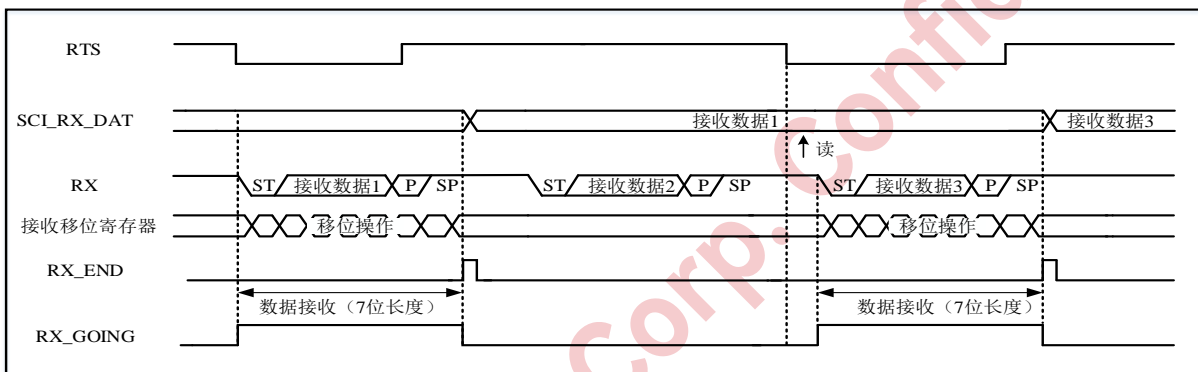


图 224 UART 接收 (使能 RTS 控制)

20.4.4.4 多处理器通信

使用多处理器通信功能，可以在共享通信线路的情况下，实现多个处理器之间通过异步串行通信来进行数据的发送和接收。在多核通信中，每一个接收设备都有一个唯一的 ID。串行通信周期是由指定接收设备的 ID 传输周期和向指定接收设备传输数据的数据周期组成。通过 UART_MOD_CTL.MP_BIT_FLG 位来区分 ID 周期和数据周期，当此位为 1 时，表示当前接收为 ID 信息，反之，则为数据传输。

下图展示了一个多处理器通信的示例，首先，发送设备发送 ID 信息到接收设备，此时，多处理器标志位值为 1，然后，发送设备进行数据发送，此时，多处理器标志位值保持为 0。当接收设备接收到多处理器标志位为 1 的 ID 信息时，与自身的 ID 进行比较，如果一致，则进行后续数据的接收，否则，将忽略后续的数据，直到下一次多处理器标志位为 1 的 ID 信息帧被接收到。通过 UART_MOD_CTL.MP_EN 位来使能多处理器通信功能。

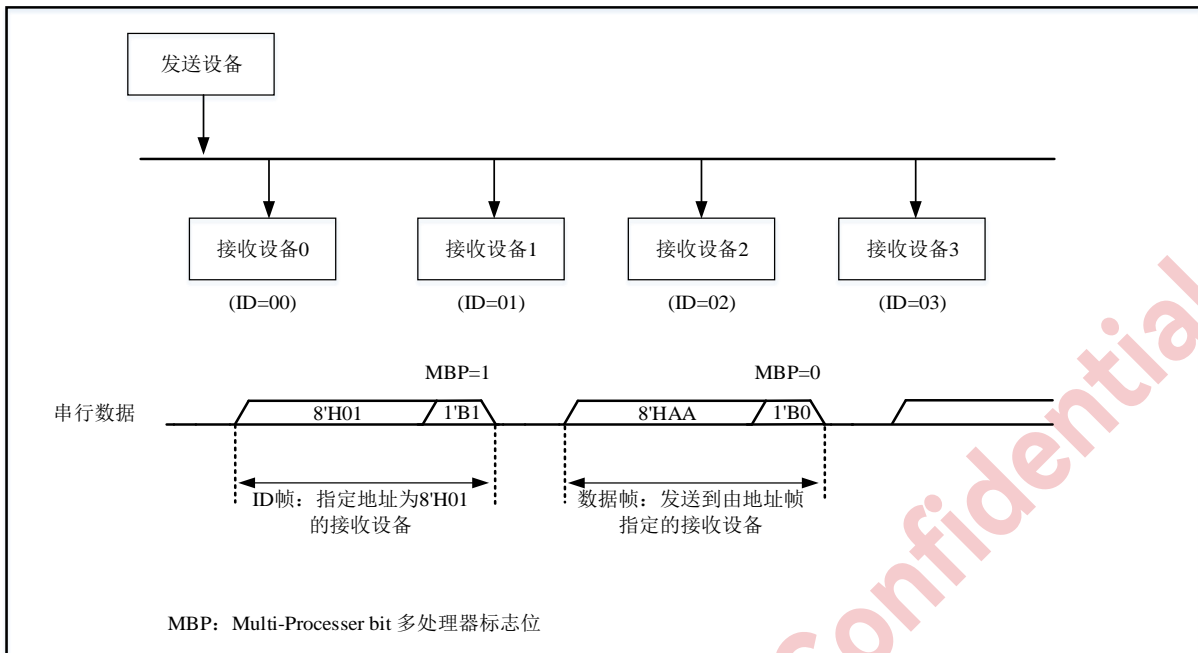


图 225 多处理器通信示例

20.4.4.5 多处理器发送

当串行通信接口作为发送设备时, 通过 `UART_MOD_CTL.MP_BIT` 来设置当前将要发送的数据是 ID 帧还是数据帧, 当 `UART_MOD_CTL.MP_BIT` 设置为 1 时, 表示将要发送的数据为 ID 帧, 反之, 则为数据帧。除此之外, 与前文的 UART 发送并无其他区别。

20.4.4.6 多处理器接收

当串行通信接口作为接收设备时, 通过 `UART_MOD_CTL.MP_BIT_FLG` 来判断当前接收数据是 ID 帧还是数据帧。当 `UART_MOD_CTL.MP_DET_EN` 设置为 1 时, 此时进入地址帧检测模式, 只有当接收到多处理器标志为 1 的 ID 帧时, 状态寄存器中的接收完成标志会被置 1, 其他数据帧会被忽略, 状态寄存器中的完成标志不会被置起, 并且接收数据寄存器中不会存储接收到的数据。

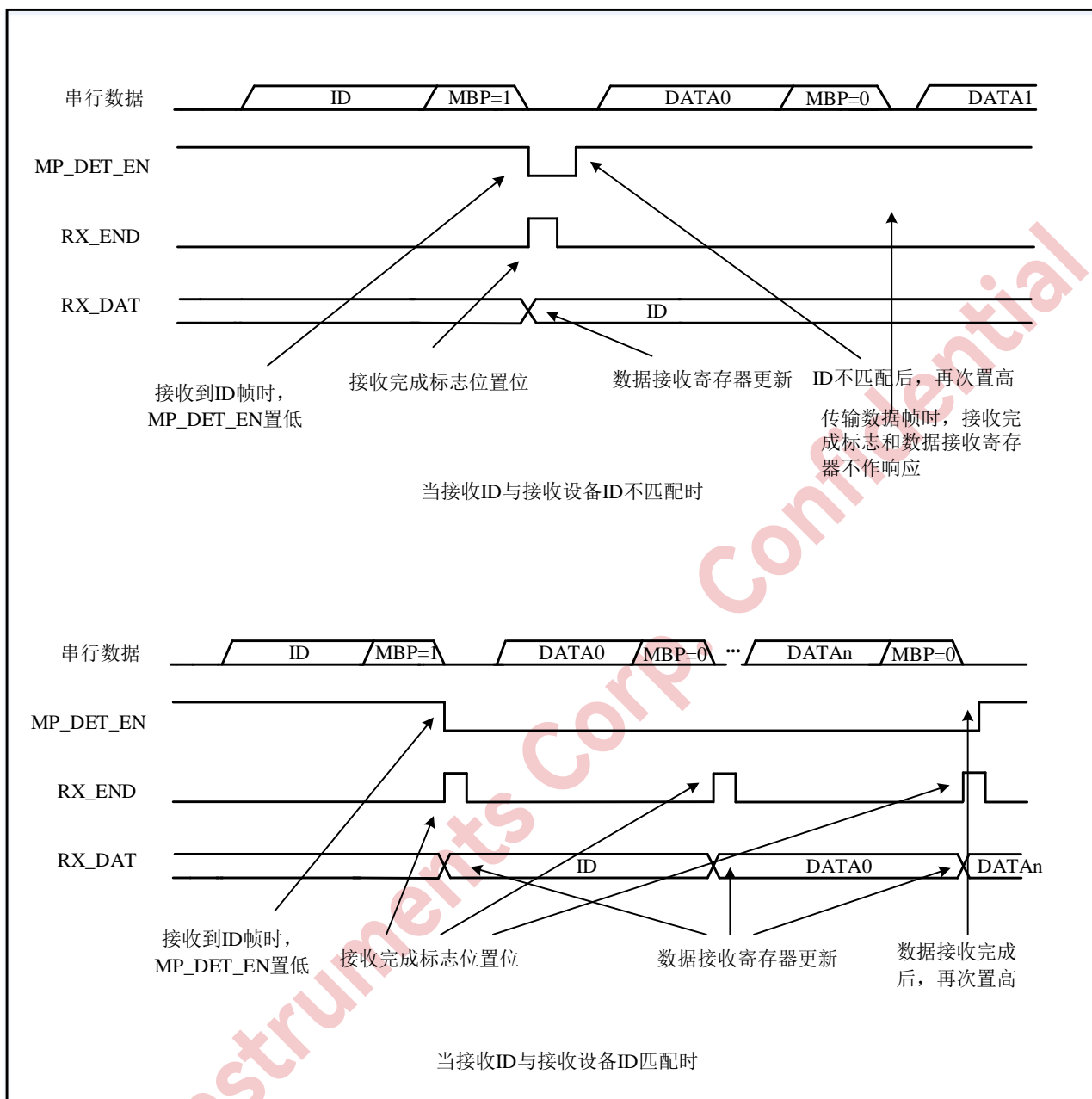


图 226 多处理器通信地址检测功能

20.4.4.7 波特率计算

用于 UART 通信的波特率计算公式如下：

- 发送波特率

- SCI 时钟频率 = 系统时钟频率/2

$$OP_CLK_DIV = (SCI_BAUD.CLK_DVI1+1) * (SCI_BAUD.CLK_DVI2+1)$$

$$\text{波特率} = (SCI \text{ 时钟频率} / 2^{CLK_PRx_SEL}) \div OP_CLK_DIV \div 2 \text{ (bps)}$$

在 UART 模式下时，OP_CLK_DIV 不能为 7'H1 和 7'H0。

- 接收波特率允许范围

$$(\text{最大可接收波特率}) = \frac{2 \times div \times N}{2 \times div \times N - div + 2} \times Brate$$

$$(\text{最小可接收波特率}) = \frac{2 \times div \times (N-1)}{2 \times div \times N - div - 2} \times Brate$$

Brate: 接收方的计算波特率值

div: OP_CLK_DIV

N: 1 帧数据长度

20.4.5 简易 I²C 通信模式

这是一种时钟同步通信功能，采用两线：串行时钟线 (SCL) 和串行数据线 (SDA)，进行两个或者多个器件之间的通信。串行通信接口的 I²C 模式仅支持主设备功能，不能作为从设备使用。

20.4.5.1 起始，重新开始，停止条件发送

I²C 模式寄存器中设有 ST_TRIG, RST_TRIG 和 SP_TRIG 三个可配置位，用于软件触发，相对地产生起始，重新开始以及停止条件。当软件将 ST_TRIG 置 1 后，串行通信接口开始发送起始条件到从设备，发送完成后，此位会被硬件自动清除，在此期间，软件禁止对该位进行改写。重新开始和停止条件的发送流程与起始条件相同。

20.4.5.2 数据发送

数据发送操作用于在发送完成地址字段之后将数据发送至从设备，在将所有数据全部发送到从设备后，产生停止条件并释放总线。

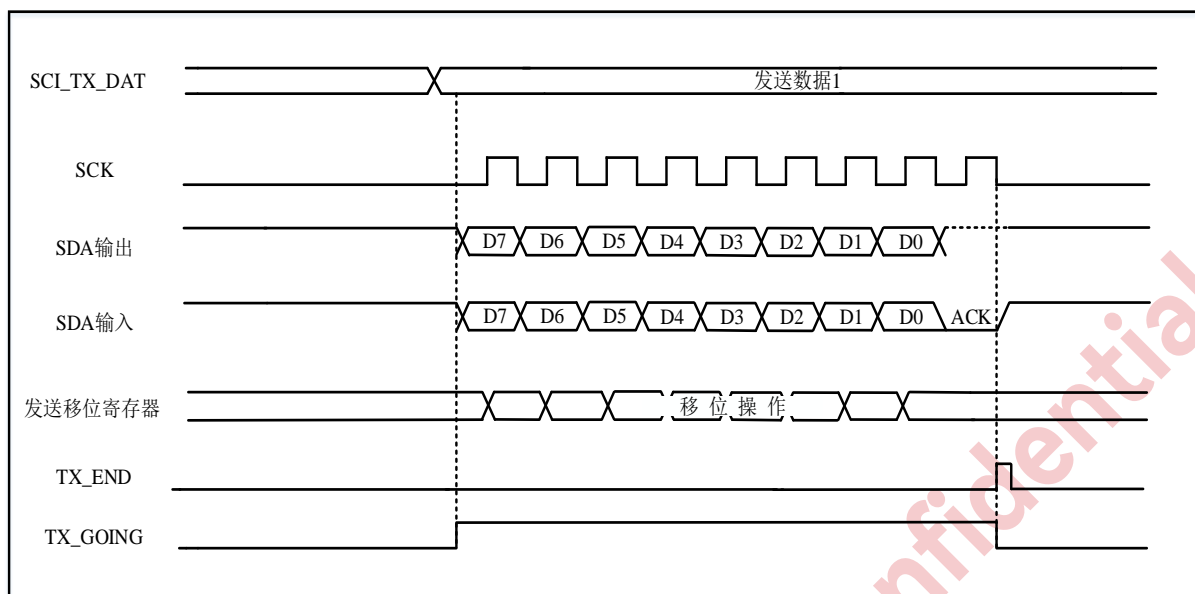


图 227 数据发送时序图

20.4.5.3 数据接收

数据接收操作用于在发送完成地址字段之后，接收从设备发出的数据，在将所有数据全部接收完成后，产生停止条件并释放总线。

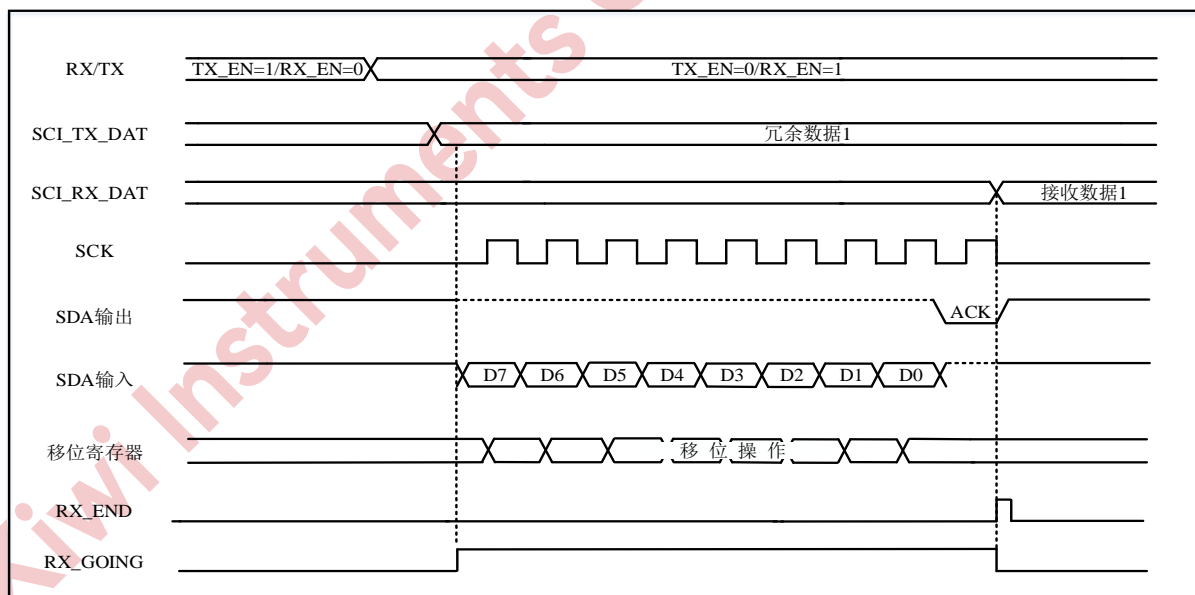


图 228 数据接收时序图

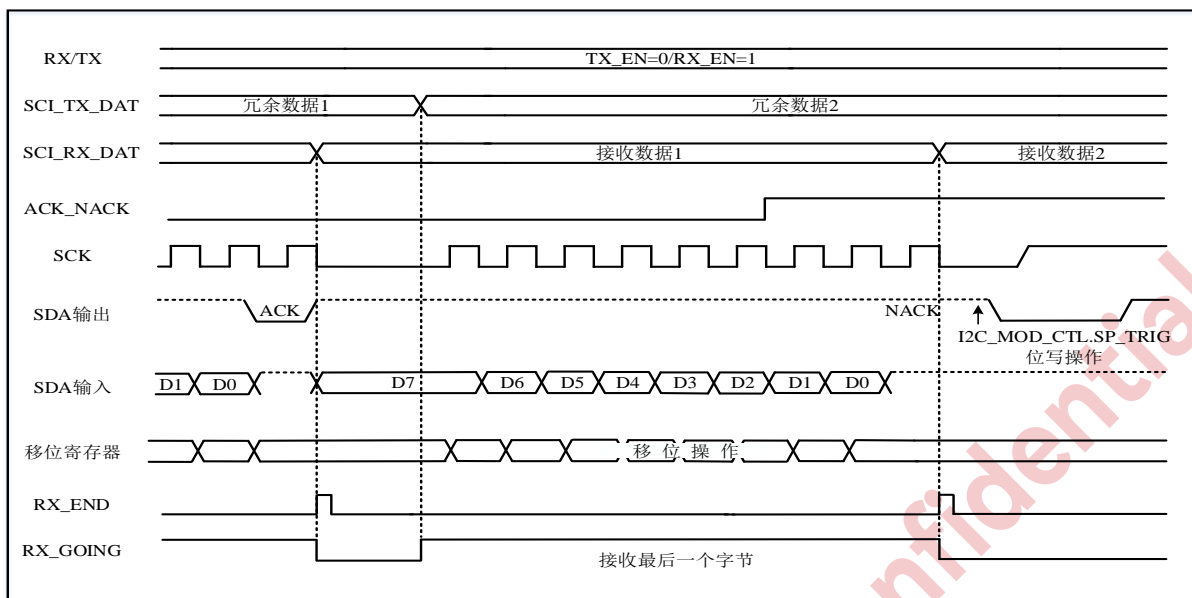


图 229 接收最后一笔数据时序图

20.4.5.4 时钟同步

在通信应用中，从设备有可能将时钟拉低并持续一段时间，当此种情况发生时，主设备需要等待从设备释放时钟后，继续进行传输，以保证传输的正确性。通过使能 I²C 模式寄存器中 CLK_SYNC_EN 位，使串行通信接口来支持此功能。

20.4.5.5 传输时钟频率

I²C 总线模式通信的传输时钟频率计算表达式如下：

- SCI 时钟频率 = 系统时钟频率/2

$$OP_CLK_DIV = (SCI_BAUD.CLK_DVI1+1) * (SCI_BAUD.CLK_DVI2+1)$$

$$传输频率 = (SCI 时钟频率 / 2^{CLK_PRx_SEL}) \div OP_CLK_DIV \div 2 (Hz)$$

20.4.6 SYNC 通信模式

串行通信接口可配置为 SYNC 通信模式，该模式下采用三条线路：同步时钟 (SCK)，串行数据传送 (TX) 和串行数据接收 (RX) 线路。利用这三条通信线路，数据在单片机与其他通信方之间实现同步传送。

此种模式与 SPI 模式相类似，串行通信接口既可以作为主设备（发送同步时钟 SCK 到外部设备），也可以作为从设备（接收来自外部设备的同步时钟 SCK），同时，时钟与数据的相位也可以进行配置，SYNC 模式下，只支持 SPI 模式下的 MODE3 与 MODE0 模式。

20.4.6.1 SYNC 模式主发送接收

主发送接收是指串行通信接口作为主设备，输出传输时钟 (SCK)，并从外部器件接收数据，同时将数据写入外部器件；主发送接收支持单笔模式和连续模式，通过状态寄存器中的完成状态标志或者缓冲器空满标志来实现。

具体的配置参考编程指南中的 SYNC 主发送接收部分。

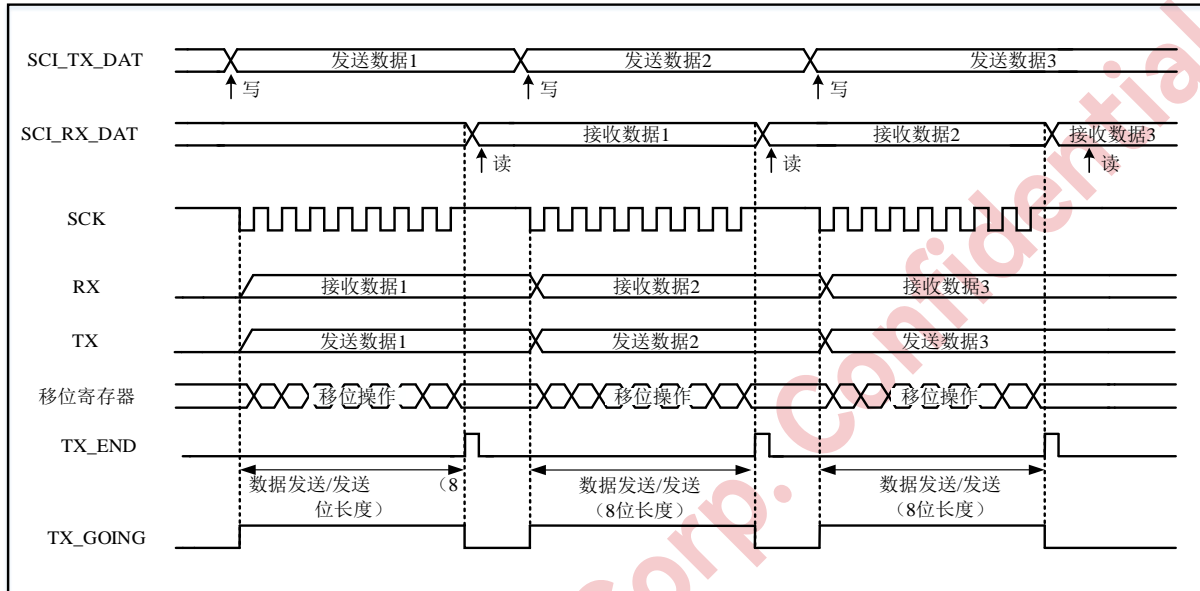


图 230 SYNC 主发送接收 (单笔模式) 时序图

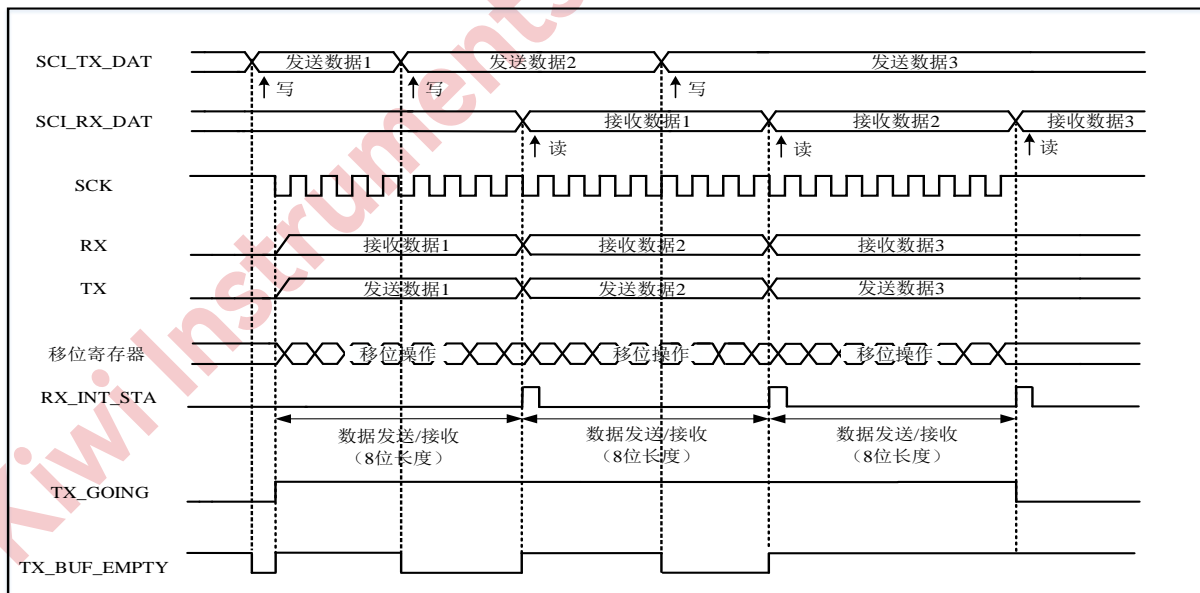


图 231 SYNC 主发送接收 (连续模式) 时序图

20.4.6.2 SYNC 模式从发送接收

从发送接收是指串行通信接口作为从设备，接收来自外部器件输入的传输时钟 (SCK)，并从外部器件接收数据，同时将数据写入外部器件；从发送接收支持单笔模式和连续模式，通过中断状态寄存器的完成状态标志及缓冲器空状态标志实现。具体的配置参考编程指南中的主发送接收部分。

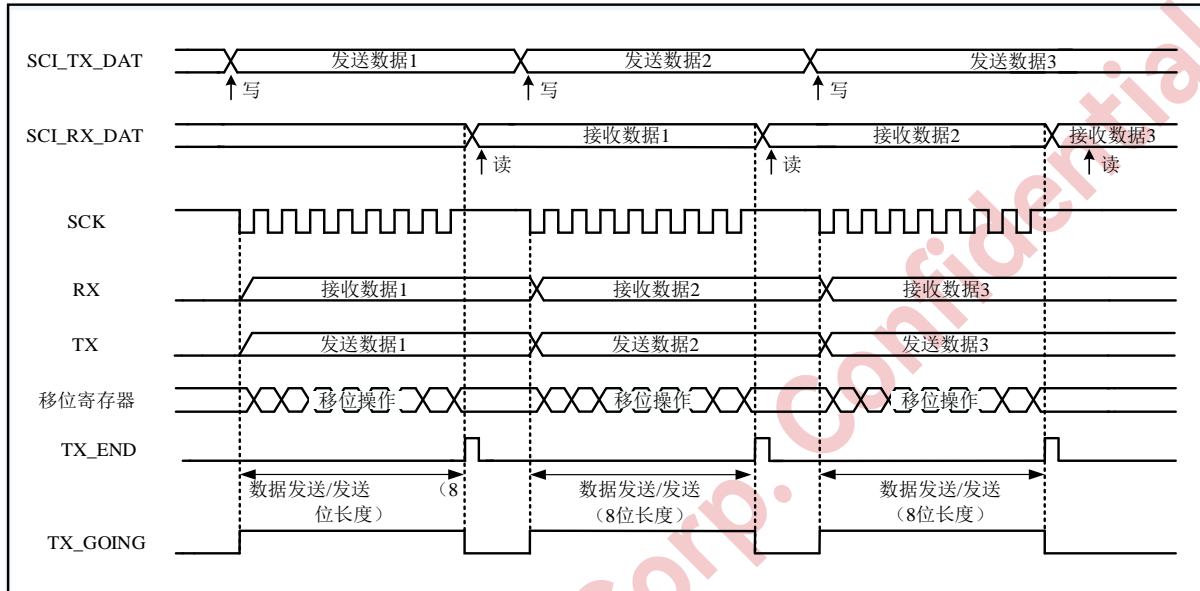


图 232 SYNC 从发送接收 (单笔模式) 时序图

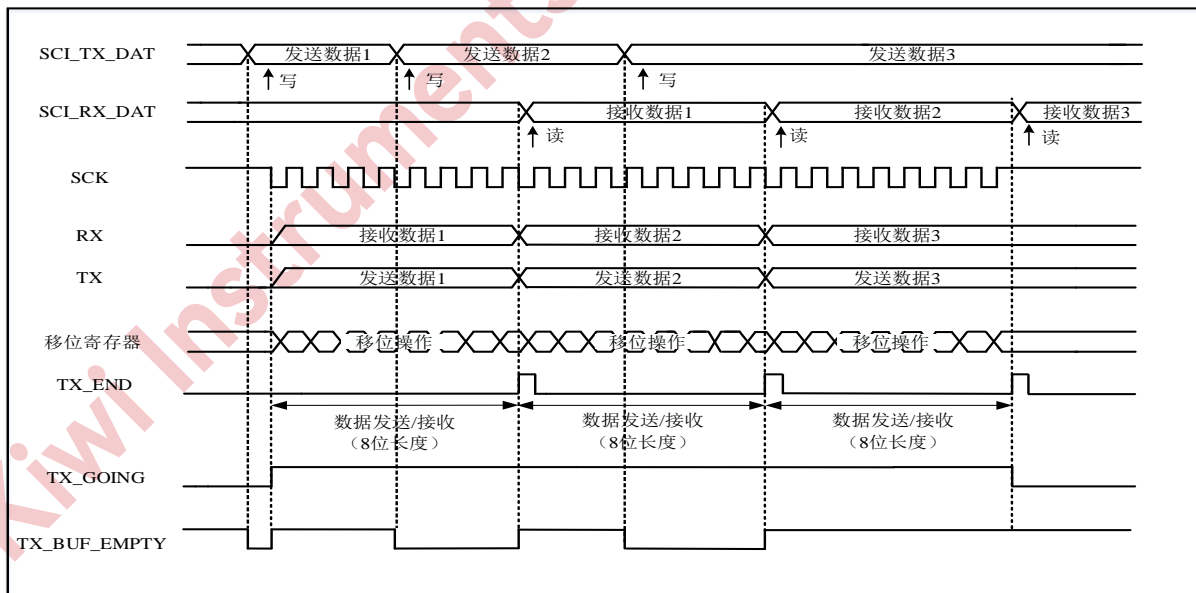


图 233 SYNC 从发送接收 (连续模式) 时序图

20.4.6.3 硬件流控

在 SYNCN 模式下，输入信号 CTS 可以作为数据发送控制信号，通过 SYNC 模式寄存器中的 CTS_EN 位来使能该功能。当 CTS 电平由高变低后，串行通信接口开始进行数据发送到外部设备。

在数据发送过程中，将 CTS 电平拉高后，并不会影响当前正在发送的数据，当前帧数据发送完成后，串行通信接口停止发送动作。

同时，输出信号 RTS 可以作为数据接收控制信号通过 SYNC 模式寄存器中的 RTS_EN 位来使能该功能。当中断状态寄存器中的 RX_BUF_FULL 状态为低时，串行通信接口可以进行数据接收，此时 RTS 信号电平会被拉低，反之，则 RTS 会被置高，直到软件读取数据接收寄存器。

注意，在 SYNC 模式下，CTS 和 RTS 信号是复用同一引脚，因此，在同一时间内只能使用两者功能中的一个。

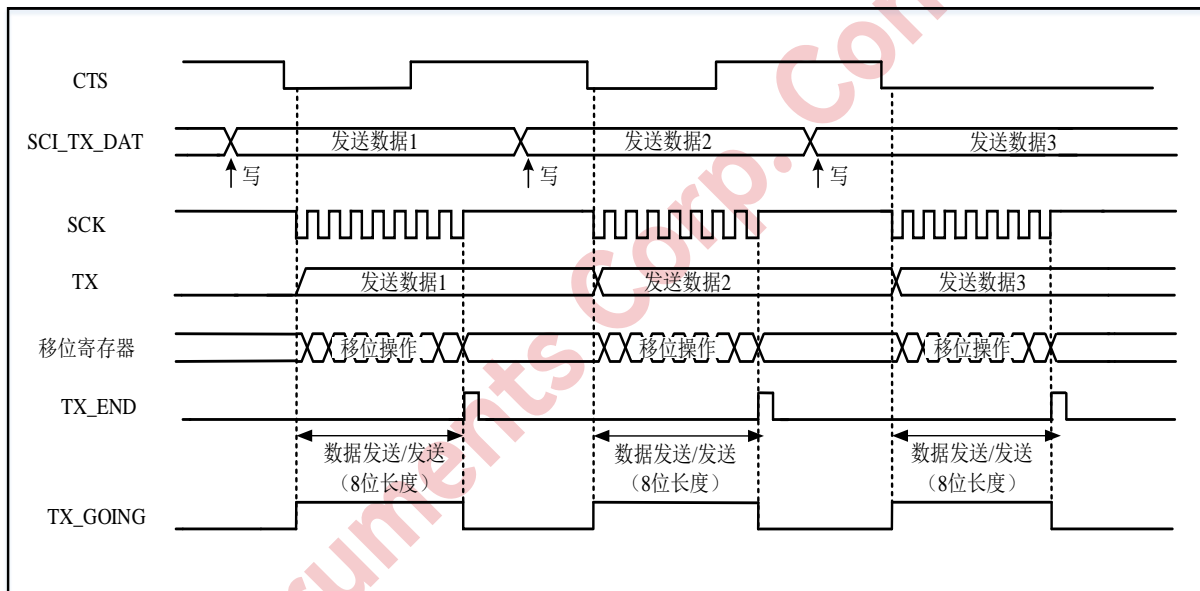


图 234 SYNC 发送 (使能 CTS 控制)

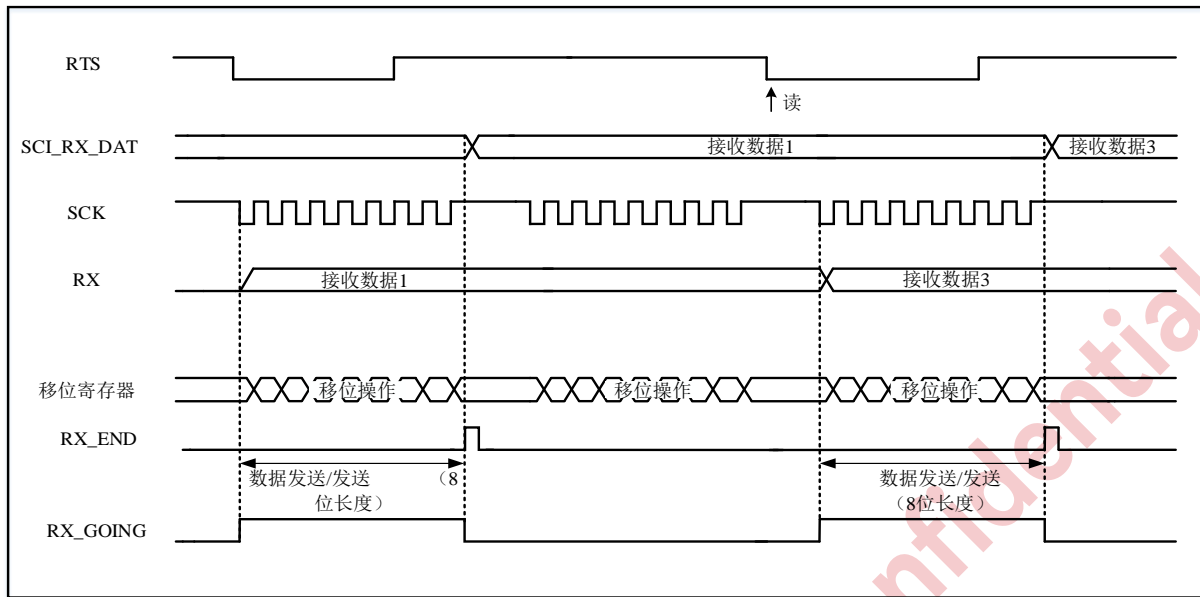


图 235 SYNC 接收 (使能 RTS 控制)

20.4.6.4 波特率计算

用于 SYNC 模式通信的波特率计算公式如下:

- 主设备

$$\text{SCI 时钟频率} = \text{系统时钟频率} / 2$$

$$\text{OP_CLK_DIV} = (\text{SCI_BAUD.CLK_DVI1} + 1) * (\text{SCI_BAUD.CLK_DVI2} + 1)$$

$$\text{通信波特率} = (\text{SCI 时钟频率} / 2^{\text{CLK_PRSx_SEL}}) \div \text{OP_CLK_DIV} \div 2 \text{ (bps)}$$

- 从设备

$$\text{传输频率} = \text{由外部主设备提供的串行时钟 (SCK) 频率 (Hz)}$$

当串行单元作为从设备时, 最大支持传输时钟频率为 400K。

20.4.7 Smart Card 通信模式

串行通信接口支持智能卡 (IC 卡) 模式, 参考 IOS/IEC 7816-3 (集成电路卡标准) 部分。该模式下采用三条线路: 输出时钟 (SCK), 数据发送 (TX) 和数据接收 (RX) 线路。

20.4.7.1 时钟输出

在 Smart Card 模式下, 串行通信接口作为主设备需要向外部从设备输出时钟, 输出时钟速率由波特率设置寄存器中的 CLK_DIV1 位来决定, 而通信波特率由波特率设置寄存器中的 CLK_DIV1 和 CLK_DIV2 共同决定。

- SCI 时钟频率 = 系统时钟频率/2
- 输出时钟频率

$$OP_CLK_DIV = (SCI_BAUD.CLK_DVI1+1)$$

$$\text{波特率} = (SCI \text{ 时钟频率} / 2^{CLK_PRSx_SEL}) \div OP_CLK_DIV \div 2 \text{ (bps)}$$

- 传输波特率

$$OP_CLK_DIV = (SCI_BAUD.CLK_DVI1+1) * (SCI_BAUD.CLK_DVI2+1)$$

$$\text{波特率} = (SCI \text{ 时钟频率} / 2^{CLK_PRSx_SEL}) \div OP_CLK_DIV \div 2 \text{ (bps)}$$

20.4.7.2 数据格式 (非 BLOCK 传输模式)

下图为 Smart Cart 模式数据传输特征。

1. 一帧数据由 8 位数据长度以及 1 位奇偶校验位组成。
2. 在发送过程中, 在当前帧奇偶检验位到下一帧的起始至少需要 2etu (传输 1 位数据需要的时间)的时间来作为 Guardtime。
3. 如果在传输过程中检测到奇偶检验错误, 则接收方会在数据起始 10.5etu 的位置发送 1etu 长度的低电平错误信号。
4. 当错误标志信号产生后, 发送方需要将发生错误的帧数据重新发送, 时间间隔至少 2etu。

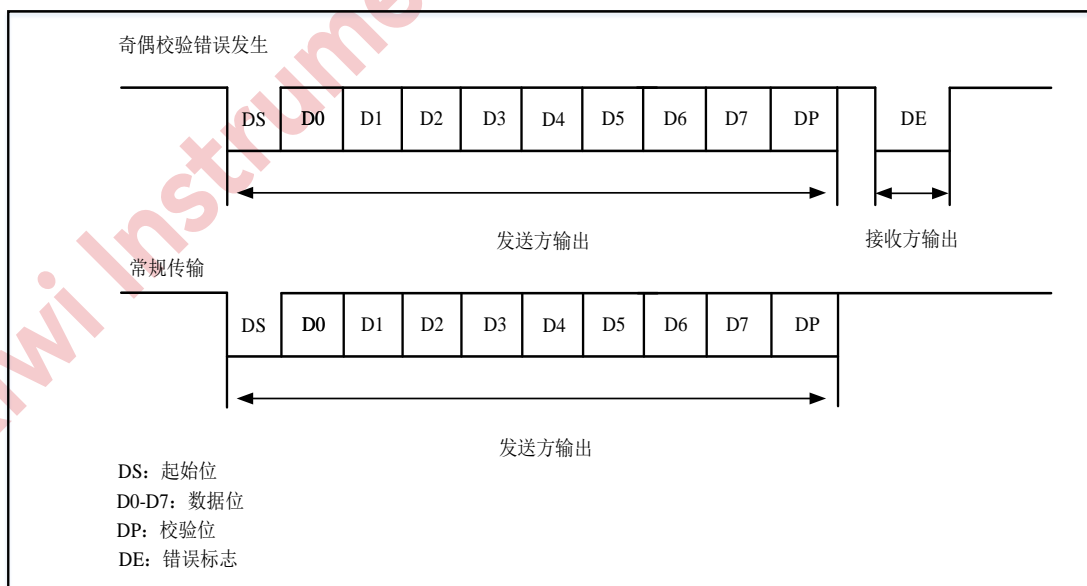


图 236 Smart Card 数据传输格式

Smart Card 通信协议中规定了两种数据位及校验位的传输约定:正向约定及反向约定。串行通信接口通过配置数据电平取反以及奇偶校验选择配置, 来实现两种约定格式。

正向约定

在正向约定中, 数据 1 用高电平表示, 数据 0 用低电平表示, 数据部分传输使用 LSB 模式, 奇偶校验位为偶校验结果, 下图以 9Bh 为例说明了正向约定的电平特征。

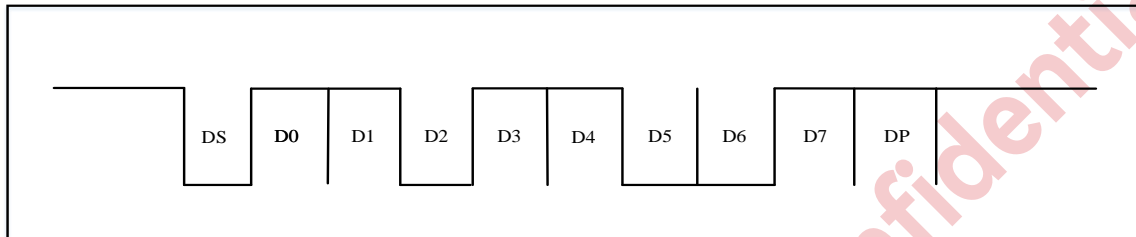


图 237 正向约定

反向约定

在正向约定中, 数据 1 用低电平表示, 数据 0 用高电平表示, 数据部分传输使用 MSB 模式, 奇偶校验位为奇校验结果, 下图以 3Ch 为例说明了反向约定的电平特征。

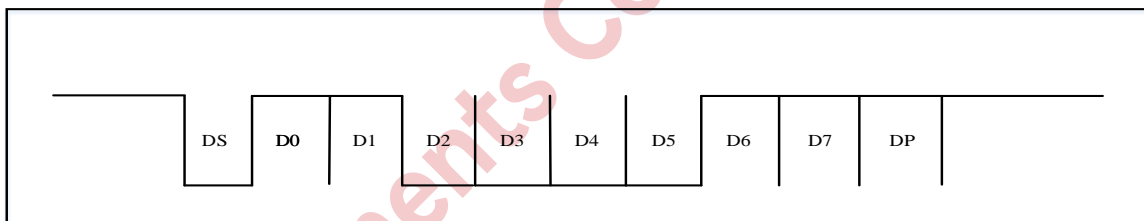


图 238 反向约定

20.4.7.3 Smart Card 发送 (非 BLOCK 传输模式)

Smart Card 发送是将数据发送到外部器件, 在非 Block 传输模式下, 如果接收到外部从设备反馈的错误标志后, 硬件会自动进行重新发送, 直到当前帧数据发送成功。

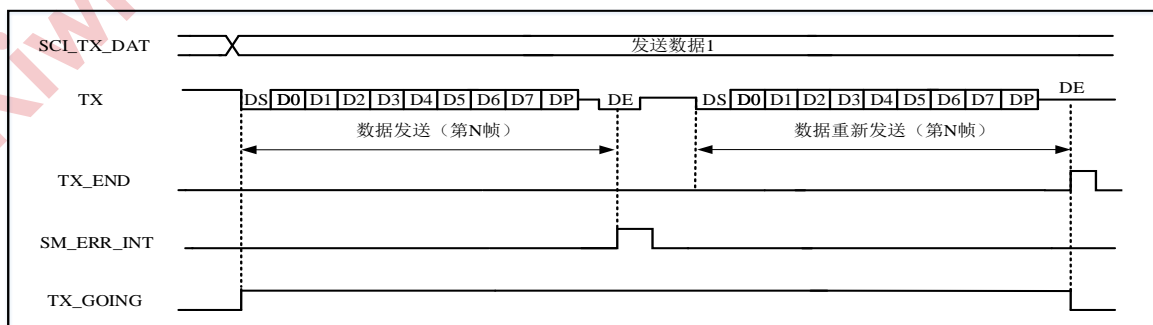


图 239 Smart Card 发送 (非 BLOCK 传输模式) 时序图

20.4.7.4 Smart Card 接收 (非 BLOCK 传输模式)

Smart Card 接收是接收外部器件发送的数据, 在非 Block 传输模式下, 当串行通信接口接收外部数据并进行校验出错后, 会发送错误标志, 并且将中断状态寄存器中的 FRM_ERR_INT 位置位, 注意, 此种情况下, 中断状态寄存器中的 RX_END 并不会被置位, 此帧数据会被硬件忽略。

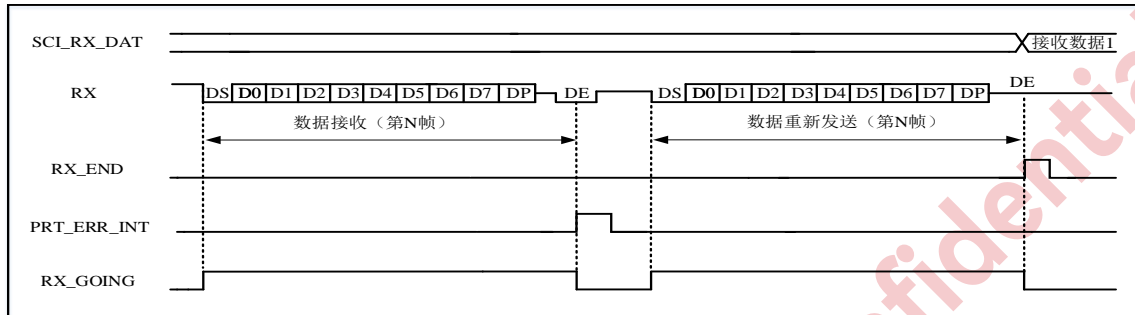


图 240 Smart Card 发送 (非 BLOCK 传输模式) 时序图

20.4.7.5 BLOCK 传输模式

在 Smart Card 的 BLOCK 传输模式下, 数据的发送和接收与常规模式下有如下不同:

1. 在接收过程中, 即使检测到奇偶校验位的错误, 也不会发送错误标志信号到线上, 但是中断状态寄存器中的 PRT_ERR_INT 位会被置位, 需要软件进行清除。
2. 在发送过程中, 中断状态寄存器中的 SMC_ERR_INT 不会被置位, 因为数据接收端不会发出错误标志信号。
3. 由于相同帧数据不会被重新发送, 当前帧的奇偶检验位到下一帧的起始信号之间至少要有 1ETU 的时间间隔。

20.4.8 SCI 中断号

本器件中包含 3 个串行通信接口模块, 每个都可以独立产生中断。

表 326 SCI 中断表

名称	中断号
SCI0_INT	12
SCI1_INT	13
SCI2_INT	14

20.5 寄存器

本章节描述了串行通信接口的相关寄存器，本器件中包含 3 个串行通信接口模块，每个串行通信接口模块的寄存器分布及内容相同，通过基址区分。三个模块的基址分别为 0x4000_4400, 0x4000_4800, 0x4000_4C00。

表 327 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x000	时钟分频选择寄存器	32	r/w	0x00000000
0x004	波特率设置寄存器	32	r/w	0x00000000
0x010	模式控制寄存器	32	r/w	0x00000000
0x020	UART 模式寄存器	32	r/w	0x00000000
0x024	SPI 模式寄存器	32	r/w	0x00000000
0x028	IIC 模式寄存器	32	r/w	0x00000000
0x02C	SMART CARD 模式寄存器	32	r/w	0x00000000
0x030	SYNC 模式寄存器	32	r/w	0x00000000
0x040	数据发送寄存器	32	r/w	0x00000000
0x050	数据接收寄存器	32	r/w	0x00000000
0x060	中断使能寄存器	32	r/w	0x00000000
0x064	中断状态寄存器	32	ro	0x00000000
0x068	中断状态清除寄存器	32	wo	0x00000000

20.5.1 时钟分频选择寄存器 (SCI_CLK_PRS)

偏移地址: 0x00

表 328 时钟分频选择寄存器

位	访问	描述
31:4	Res	Reserved 复位值: 0x0
3:0	r/w	CLK_PRS_SEL: CLK_PRS 时钟分频选择 0000: IPCLK 0001: IPCLK/2 ¹ 0010: IPCLK/2 ² 1111: IPCLK/2 ¹⁵ 复位值: 0x0 注意: IPCLK = FCLK / 2

20.5.2 波特率设置寄存器 (SCI_BAUD)

偏移地址: 0x04

表 329 波特率设置寄存器

位	访问	描述
31:25	Res	Reserved 复位值: 0x0
24:16	r/w	CLK_DIV2: 通信时钟分频设置 通信时钟频率=CLK_PRS/((CLK_DIV1+1)*(CLK_DIV2+1)*2) 复位值: 0x0
15:5	Res	Reserved 复位值: 0x0
4:0	r/w	CLK_DIV1: 通信时钟分频设置 SMART CARD 模式下, 输出时钟频率=CLK_PRS/((CLK_DIV1+1)*2) 通信时钟频率=CLK_PRS/((CLK_DIV1+1)*(CLK_DIV2+1)*2) 复位值: 0x0

20.5.3 模式控制寄存器 (SCI_MOD_CTL)

偏移地址: 0x10

表 330 模式控制寄存器

位	访问	描述
31:26	Res	Reserved 复位值: 0x0
25	r/w	CLK_NF_EN: 时钟输入滤波使能 0: 关闭 1: 使能 复位值: 0x0 注: 当选择 IIC 功能时, 此位控制 SCL 滤波功能 当选择 SPI slave 功能时, 此位控制 SCK 滤波功能
24	r/w	DAT_NF_EN: 数据输入滤波使能 0: 关闭 1: 使能 复位值: 0x0 注: 当选择 IIC 功能时, 此位控制 SDA 滤波功能 当选择 UART 功能时, 此位控制 RX 滤波功能 当选择 SPI master 功能时, 此位控制 MISO 滤波功能 当选择 SPI slave 功能时, 此位控制 MOSI 滤波功能 当选择 SYNC 功能时, 此位控制 RX 滤波功能 当选择 Smart card 时, 此位控制 RX 滤波功能

位	访问	描述
23:20	Res	Reserved 复位值: 0x0
19:18	r/w	SCK_OUT_SEL: 时钟输出选择 00: 硬件通信控制 01: Reserved 10: 固定低电平输出 11: 固定高电平输出 复位值: 0x0
17:16	r/w	DAT_OUT_SEL: 数据输出选择 00: 硬件通信控制 01: Reserved 10: 固定低电平输出 11: 固定高电平输出 复位值: 0x0
15:14	Res	Reserved 复位值: 0x0
13	r/w	DAT_INV: 数据位电平反相使能 0: 关闭 1: 使能 复位值: 0x0
12	r/w	DAT_DIR: 数据位方向选择 0: MSB 1: LSB 复位值: 0x0
11:10	Res	Reserved 复位值: 0x0
9	r/w	RX_EN: 接收使能 0: 关闭 1: 使能 复位值: 0x0
8	r/w	TX_EN: 发送使能 0: 关闭 1: 使能 复位值: 0x0
7:5	Res	Reserved 复位值: 0x0
4	r/w	DMA_EN: DMA 使能 0: 关闭 1: 使能 复位值: 0x0

位	访问	描述
3	Res	Reserved 复位值: 0x0
2:0	r/w	MOD_SEL: 模式选择 001: UART 模式 010: SPI 模式 011: IIC 模式 100: SYNC 模式 101: SMART CARD 模式 复位值: 0x0

20.5.4 UART 模式寄存器 (UART_MOD_CTL)

偏移地址: 0x20

表 331 UART 模式寄存器

位	访问	描述
31:17	Res	Reserved 复位值: 0x0
16	ro	MP_BIT_FLG: Multi-processor 模式下接收帧标志 0: 数据帧 1: ID/地址帧 复位值: 0x0
15:11	Res	Reserved 复位值: 0x0
10	r/w	MP_BIT: Multi-processor 模式下传输帧标志 0: 数据帧 1: ID/地址帧 复位值: 0x0
9	r/w	MP_DET_EN: Multi-processor 模式地址帧检测使能 0: 关闭 1: 使能 复位值: 0x0
8	r/w	MP_EN: Multi-processor 模式使能 0: 关闭 1: 使能 复位值: 0x0
7	Res	Reserved 复位值: 0x0
6	r/w	RTS_EN: RTS 使能 0: 关闭 1: 使能 复位值: 0x0

位	访问	描述
5	r/w	CTS_EN: CTS 使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	STOP_LEN: 停止位长度 0: 1bit 停止位 1: 2bit 停止位 复位值: 0x0
3:2	r/w	PRT_SEL: 校验位选择 2'b00: 不输出/不接收校验位 2'b01: 发送 0 校验位/接收校验位但不进行校验 2'b10: 发送奇校验/接收奇校验 2'b11: 发送偶校验/接收偶校验 复位值: 0x0
1:0	r/w	DAT_LEN: 数据位长度 2'b00: 8bits 2'b01: reserved 2'b10: 7bits 2'b11: 9bits 复位值: 0x0

20.5.5 SPI 模式寄存器 (SPI_MOD_CTL)

偏移地址: 0x24

表 332 SPI 模式寄存器

位	访问	描述
31:9	Res	Reserved 复位值: 0x0
8	r/w	CS_DEACTIVE: CS 引脚无效 0: 无影响 1: 当前传输结束后, 硬件将 CS 信号置为无效电平状态 复位值: 0x0
7:4	Res	Reserved 复位值: 0x0
3	r/w	CS_EN: CS 引脚控制使能 0: 关闭 1: 使能 复位值: 0x0
2	r/w	MS_SEL: Master/Slave 模式选择 0: Master 模式 1: Slave 模式

位	访问	描述
		复位值: 0x0
1	r/w	CLK_PHS: 时钟相位选择 0: 不反相 (在 SCK 的下降沿输出数据, 在其上升沿输入数据) 1: 反相 (在 SCK 的上升沿输出数据, 在其下降沿输入数据) 复位值: 0x0
0	r/w	DAT_PHS: 数据采样相位选择 0: 从开始串行时钟操作时开始数据输出 (输入) 1: 从开始串行时钟操作时的半个时钟前开始数据输出 (输入) 复位值: 0x0

20.5.6 IIC 模式寄存器 (IIC_MOD_CTL)

偏移地址: 0x28

表 333 IIC 模式寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14	r/w	SP_TRIG: 停止条件产生触发 0: 无动作 1: 产生停止开始条件 复位值: 0x0
13	r/w	RST_TRIG: 重新开始条件产生触发 0: 无动作 1: 产生重新开始条件 复位值: 0x0
12	r/w	ST_TRIG: 开始条件产生触发 0: 无动作 1: 产生开始条件 复位值: 0x0
11:10	Res	Reserved 复位值: 0x0
9:8	r/w	SDA_HOLD: SDA 保持时间选择 00: 关闭保持功能 01: 保持波特率周期的 1/4 10: 保持波特率周期的 1/2 11: 保持接近整个波特率周期 (相差一个 (SCI 时钟频率/2 ^{CLK_PRS_SEL}) 的周期时间) 复位值: 0x0
7	Res	Reserved 复位值: 0x0
6	r/w	FASE_MODE_EN: 快速模式使能 0: 关闭

位	访问	描述
		1: 使能 当使用 1Mbps 通信速率模式时，此位需要使能。 复位值: 0x0
5	r/w	CLK_SYNC_EN: 时钟同步使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	ACK_NACK: ACK/NACK 发送选择 0: 接收时发送 ACK 1: 接收时发送 NACK 复位值: 0x0
3:0	Res	Reserved 复位值: 0x0

20.5.7 SMART CARD 模式寄存器 (SMC_MOD_CTL)

偏移地址: 0x2C

表 334 SMART CARD 模式寄存器

位	访问	描述
31:4	Res	Reserved 复位值: 0x0
3:2	r/w	PRT_SEL: 校验位选择 2'b00: 不输出/不接收校验位 2'b01: 发送 0 校验位/接收校验位但不进行校验 2'b10: 发送奇校验/接收奇校验 2'b11: 发送偶校验/接收偶校验 复位值: 0x0 注意: 此位与 UART 模式寄存器中的 PRT_SEL 位同步被读写
1	Res	Reserved 复位值: 0x0
0	r/w	BLOCK_EN: 传输模式选择 0: 正常传输模式 1: 块传输模式 复位值: 0x0

20.5.8 SYNC 模式寄存器 (SYNC_MOD_CTL)

偏移地址: 0x30

表 335 SYNC 模式寄存器

位	访问	描述
31:17	Res	Reserved 复位值: 0x0
16	r/w	CLK_OUT_TRIG: 时钟输出触发信号 0: 无影响 1: 正在发送时钟 注意: 当时钟发送完成后, 此位自动清 0
15:7	Res	Reserved 复位值: 0x0
6	ro	RTS_EN: RTS 使能 0: 关闭 1: 使能 复位值: 0x0 注意: 此位通过 UART 模式寄存器中的 RTS_EN 位配置
5	ro	CTS_EN: CTS 使能 0: 关闭 1: 使能 复位值: 0x0 注意: 此位通过 UART 模式寄存器中的 CTS_EN 位配置
4:3	ro	DAT_LEN: 数据位长度 2'b00: 8bits 2'b01: reserved 2'b10: 7bits 2'b11: 9bits 注意: 此位通过 UART 模式寄存器中的 DAT_LEN 位配置, 对于 SYNC 模式, 只有 8bit 和 9bit 可以选择。 复位值: 0x0
2	ro	MS_SEL: Master/Slave 模式选择 0: Master 模式 (外设器件接收时钟) 1: Slave 模式 (外设器件发送时钟) 注意: 此位通过 SPI 模式寄存器中的 MS_SEL 位配置 复位值: 0x0
1	ro	CLK_PHS: 时钟相位选择 0: 不反相 (在 SCK 的下降沿输出数据, 在其上升沿输入数据) 1: 反相 (在 SCK 的上升沿输出数据, 在其下降沿输入数据) 复位值: 0x0 注意: 此位通过 SPI 模式寄存器中的 CLK_PHS 位配置

位	访问	描述
0	ro	DAT_PHS: 数据采样相位选择 0: 从开始串行时钟操作时开始数据输出 (输入) 1: 从开始串行时钟操作时的半个时钟前开始数据输出 (输入) 复位值: 0x0 注意: 此位通过 SPI 模式寄存器中的 DAT_PHS 位配置

20.5.9 发送数据寄存器 (SCI_TX_DAT)

偏移地址: 0x40

表 336 发送数据寄存器

位	访问	描述
31:9	Res	Reserved 复位值: 0x0
8:0	r/w	TX_DAT: 发送数据寄存器 发送数据寄存器 写操作被视为软件触发操作 复位值: 0x0

20.5.10 接收数据寄存器 (SCI_RX_DAT)

偏移地址: 0x50

表 337 接收数据寄存器

位	访问	描述
31:9	Res	Reserved 复位值: 0x0
8:0	r/w	RX_DAT: 串行数据寄存器 接收数据寄存器 复位值: 0x0

20.5.11 中断使能寄存器 (SCI_IE)

偏移地址: 0x60

表 338 中断使能寄存器

位	访问	描述
31:11	Res	Reserved 复位值: 0x0
10	r/w	IIC_FNS_INTEN: IIC 模式下, 开始, 重新开始, 停止条件发送完成中断使能 0: 关闭 1: 使能

位	访问	描述
		复位值: 0x0
9	r/w	SM_ERR_INTEN: SMART CARD 模式下收到错误反馈中断使能 0: 关闭 1: 使能 复位值: 0x0
8	r/w	MP_DET_INTEN: Multi-processor 模式下地址帧检测中断使能 0: 关闭 1: 使能 复位值: 0x0
7	r/w	FRM_ERR_INTEN: 帧错误中断标志使能 0: 关闭 1: 使能 复位值: 0x0
6	r/w	PRT_ERR_INTEN: 奇偶校验错误中断标志使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	UDR_ERR_INTEN: 下溢错误中断标志使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	OVR_ERR_INTEN: 溢出错误中断标志使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	RX_END_INTEN: 接收完成中断使能 0: 关闭 1: 使能 复位值: 0x0
2	r/w	TX_END_INTEN: 发送完成中断使能 0: 关闭 1: 使能 复位值: 0x0
1	r/w	RX_BUF_FULL_INTEN: 接收数据缓冲器满中断使能 0: 关闭 1: 使能 复位值: 0x0
0	r/w	TX_BUF_EMPTY_INTEN: 发送数据缓冲器空中断使能 0: 关闭 1: 使能 复位值: 0x0

20.5.12 中断状态寄存器 (SCI_STA)

偏移地址: 0x64

表 339 中断状态寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14	ro	RX_GOING: 接收状态显示标志 0: 接收停止或者暂停 1: 接收正在进行 复位值: 0x0
13	ro	TX_GOING: 发送状态显示标志 0: 发送停止或者暂停 1: 发送正在进行 复位值: 0x0
12	ro	ACK_NACK_FLAG: IIC 模式下 ACK/NACK 接收标志 0: 未进行传输或接收到 ACK 1: 接收到 NACK 复位值: 0x0
11	Res	Reserved 复位值: 0x0
10	ro	IIC_FNS_INT: IIC 模式下开始, 重新开始, 停止条件发送完成中断标志 0: 未发送或发送未完成 1: 发送完成 复位值: 0x0
9	ro	SM_ERR_INT: SMART CARD 模式下收到错误反馈中断标志 0: 未收到错误反馈 1: 收到错误反馈 复位值: 0x0
8	ro	MP_DET_INT: Multi-processor 模式下地址帧检测中断标志 0: 未检测到地址帧 1: 检测到地址帧 复位值: 0x0
7	ro	FRM_ERR_INT: 帧错误中断标志 0: 无错误发生 1: 帧中断发生 复位值: 0x0
6	ro	PRT_ERR_INT: 奇偶校验错误中断标志 0: 无错误发生 1: 奇偶校验错误

位	访问	描述
		复位值: 0x0
5	ro	UDR_ERR_INT: 下溢错误中断标志 0: 无错误发生 1: 下溢错误中断发生 复位值: 0x0
4	ro	OVR_ERR_INT: 溢出错误中断标志 0: 无错误发生 1: 溢出错误中断发生 复位值: 0x0
3	ro	RX_END: 接收完成标志 0: 接收未完成或未进行传输 1: 接收完成 复位值: 0x0
2	ro	TX_END: 发送完成标志 0: 发送未完成或未进行传输 1: 发送完成 复位值: 0x0
1	ro	RX_BUF_FULL: 接收数据缓冲器满标志 0: 接收数据缓冲器空 1: 接收数据缓冲器数据有效 复位值: 0x0
0	ro	TX_BUF_EMPTY: 发送数据缓冲器空标志 0: 发送数据缓冲器数据有效 1: 发送数据缓冲器空 复位值: 0x0

20.5.13 中断状态清除寄存器 (SCI_STA_CLR)

偏移地址: 0x68

表 340 中断状态清除寄存器

位	访问	描述
31:111	Res	Reserved 复位值: 0x0
10	wo	IIC_FNS_INT: IIC 模式下, 开始, 重新开始, 停止条件发送完成中断标志清除 0: 无影响 1: 清除发送完成中断标志 复位值: 0x0

位	访问	描述
9	wo	SM_ERR_INT: SMART CARD 模式下收到错误反馈中断标志清除 0: 无影响 1: 清除收到错误反馈中断标志 复位值: 0x0
8	wo	MP_DET_INT: Multi-processor 模式下地址帧检测中断标志清除 0: 无影响 1: 清除检测到地址帧中断标志 复位值: 0x0
7	wo	FRM_ERR_INT: 帧错误中断标志清除 0: 无影响 1: 清除帧中断标志 复位值: 0x0
6	wo	PRT_ERR_INT: 奇偶校验错误中断标志清除 0: 无影响 1: 清除奇偶校验错误中断标志 复位值: 0x0
5	wo	UDR_ERR_INT: 下溢错误中断标志清除 0: 无影响 1: 清除下溢错误中断标志 复位值: 0x0
4	wo	OVR_ERR_INT: 溢出错误中断标志清除 0: 无影响 1: 清除溢出错误中断标志 复位值: 0x0
3	wo	RX_END: 接收完成标志清除 0: 无影响 1: 清除接收完成标志 复位值: 0x0
2	wo	TX_END: 发送完成标志清除 0: 无影响 1: 清除发送完成标志 复位值: 0x0
1	wo	RX_BUF_FULL: 接收数据缓冲器满状态清除 此位清除通过读取接收数据寄存器 复位值: 0x0
0	wo	TX_BUF_EMPTY: 发送数据缓冲器空状态清除 此位清除通过写入发送数据寄存器 复位值: 0x0

21 I²C 总线 (I2CS)

21.1 概述

I²C 总线由两条线路组成，分别是串行时钟线 (SCL) 与串行数据线 (SDA)，SCL 与 SDA 都是双向传输。可以通过 I²C 总线与其他设备进行通信，将串行数据输入转换成并行或将并行数据转换成串行输出，传输过程可开启或禁止中断使能，并通过拉低串行时钟线电平暂停传输。支持 DMA 来搬运数据从而减少 CPU 工作。

21.2 主要功能

- I2CS 主设备功能
 - 产生串行通信时钟
 - 产生开始条件，停止条件
 - 通信数据收发
 - 响应信号收发
 - 配置通信从机地址
- I2CS 从设备功能
 - 串行通信时钟识别
 - 开始条件，停止条件检测
 - 通信数据收发
 - 响应信号收发
 - 配置自身从机地址
- 通信速度支持 10Kbps/100Kbps/400Kbps/1Mbps
- 支持 DMA 数据搬运
- 支持 I²C 总线仲裁
- 支持 I²C 总线多主通信
- 7/10 bit 地址格式通信
- 收发数据缓存 FIFO (8*8bit)
- 电平超时报警
- SMBUS 总线模式 (TTL 电平范围)

21.3 模块框图

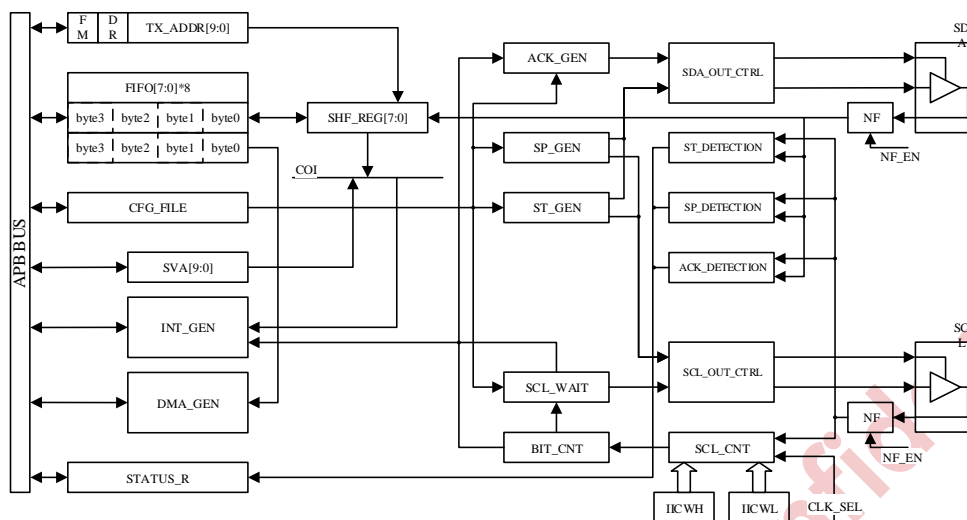
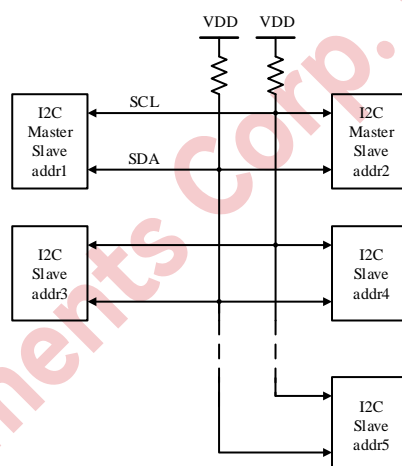


图 241 I2CS 模块框图

图 242 I²C 模块框图

21.4 功能描述

21.4.1 I²C 串行接口功能

操作停止模式

当不需要执行串行传输通信时使用该模式。关闭 I2CS 时钟使能可以降低功耗。

I²C 总线模式 (支持多主)

该模式下，使用两条线路（串行时钟线路 (SCL) 和串行数据总线线路 (SDA)）与多个设备之间进行 8 位数据传输。

该模式符合 I²C 总线格式，主设备可以产生并通过串行数据总线向从设备发送“开始条件”、“地址”、“传送方向指示”、“数据”和“停止条件”。从设备可以检测出这些接收到主设备发出的操作并完成响应回复。支持 10bit 地址模式的 I²C 协议。

由于 SCL 和 SDA 引脚为开漏输出模式，因此串行接口 I²C 中，串行时钟线和串行数据总线上需要连接上拉电阻，默认状态下 SCL 及 SDA 为高电平信号。

21.4.2 I²C 总线模式功能

引脚配置

串行时钟引脚 (SCL) 和串行数据总线引脚 (SDA) 的配置如下所示。

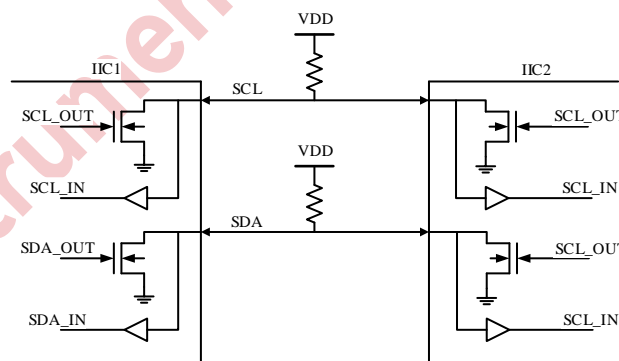


图 243 引脚配置图

- SCL 该引脚用于串行时钟输入和输出：

对于主设备和从设备，该引脚都是开漏输出，输入为施密特输入。

- SDA 该引脚用于串行数据输入和输出：

对于主设备和从设备，该引脚都是开漏输出，输入为施密特输入。

由于串行时钟线路和串行数据总线输出都是开漏输出，所以都需要外接上拉电阻，默认常态为高电平。

SMBUS 模式

当 I2CS 与总线连接选择 XPB1 (SCL) 及 XPB2 (SDA) 时，对应的 IO 口电平采用 TTL 电平格式，支持 SMBUS 总线电平规则

I2CS 时钟

$$f_{i2cs_clk} = \frac{Fclk/2}{I2CS_CLK_DIV + 1}$$

总线通信传送时钟

$$f_{i2c_bus} = \frac{Fclk/2}{(I2CS_CLK_DIV + 1) * (IICWL + IICWH) + 8 + Fclk(tR + tF)}$$

tR: SCL 信号由 0 变 1 时间；tF: SCL 信号由 1 变 0 时间。

21.4.3 I²C 总线定义及控制方法

以下章节描述 I²C 总线的串行数据通信格式和 I²C 总线使用的信号。通过 I²C 总线的串行数据总线输出“开始条件”、“地址”、“数据”和“停止条件”时的传送时序如下图所示。

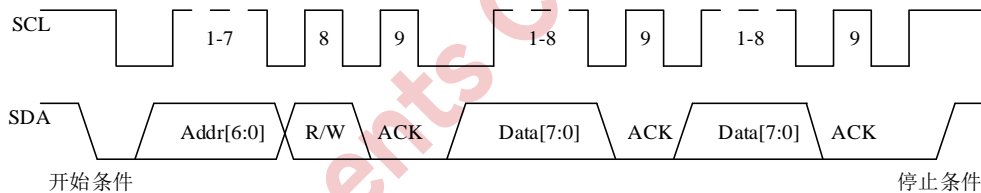


图 244 I²C 总线的串行数据传送时序

由主设备生成开始条件，从设备地址和停止条件。数据传输方向同样由主设备发送传输方向位来决定；

应答 (ACK) 信号既可由主设备产生，也可由从设备产生 (通常情况下，它由接收 8 位数据的器件输出)。串行时钟 (SCL) 由主设备连续输出。但是从设备可以延长 SCL 引脚的低电平时间，并可插入等待动作。

21.4.3.1 开始条件

当 SCL 引脚保持高电平并且 SDA 引脚由高电平变为低电平时，形成一个开始条件。SCL 和 SDA 引脚的开始条件是开始串行传送时主设备对从设备发送的信号。当器件作为从设备时，可以检测到开始条件。

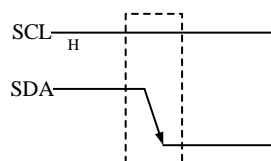


图 245 开始条件

21.4.3.2 地址发送及接收

开始条件的后续 7 位数据被定义为地址。

主设备通过向总线发送 7 位的地址数据段来选择一个目标从设备。所以，每一个接入总线的从设备必须具有一个独一无二的地址。

从设备通过硬件检测开始条件，并检查 7 位地址数据与从设备地址寄存器中保存的数据值是否匹配。如果地址数据和 I2CS_DVAD 寄存器内储存的数据值相匹配，这个从设备就被选择，并且和主设备进行通信，直到检测到总线上的结束条件。

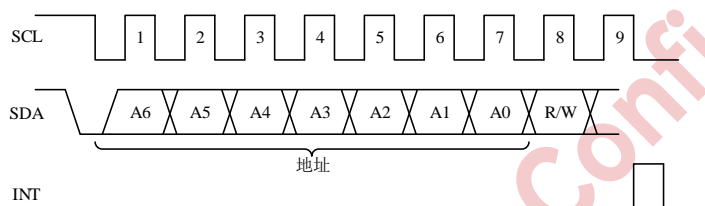


图 246 主机地址发送

当使用 10bit 地址模式，主设备在第一个地址字节的后 2bit 先发送 10bit 地址的高 2bit 地址，数据传输方向为低电平，从设备在接收到 10bit 地址的高 2bit 并与从设备本地地址匹配，从设备回复 ACK 信号。之后主设备进行第二个地址字节传输，完成剩余的 8bit 地址发送，从设备检测地址匹配并回复 ACK 信号，此时总线上的主从设备已完成地址匹配，主设备重新发送开始条件，重复第一地址字节的传输，通过该字节的数据传输方向控制位来决定后续数据传输的方向。

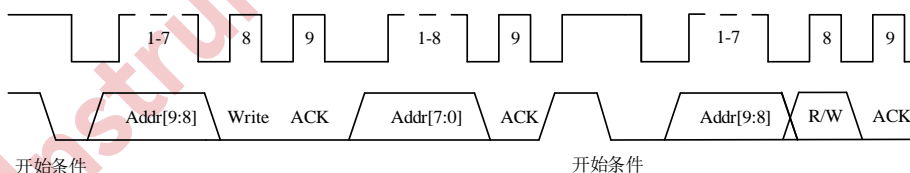


图 247 10bit 地址发送

该设备作为从机时支持地址屏蔽操作，当打开控制寄存器的 ADDR_MASK 后，无论总线在开始条件后的地址是否与设备地址匹配，都会在 8th 时钟处产生中断，并拉住 SCL 时钟总线，使总线传输进入等待状态，可通过 I2CS_SHF 寄存器读回总线上之前出现的地址数据，再判断是否回复 ACK。

从机地址寄存器 I2CS_SLAD 用于存放主机模式下发送地址，在 7bit 地址模式下，该寄存器的低 7bit 有效。在 10bit 地址模式下，I2CS_SLAD[14:8] 在第一地址字节发送，I2CS_SLAD[7:0] 在第二地址字节发送。

设备地址寄存器 I2CS_DVAD 用于存放本设备地址，当总线发出的地址与本地地址匹配时，回复 ACK 响应信号。在 7bit 地址模式下，该寄存器低 7bit 有效。在 10bit 地址模式下，I2CS_DVAD[14:8] 与第一地址字节进行匹配检验，I2CS_DVAD[7:0] 与第二地址字节进行匹配检验。

21.4.3.3 传输方向控制位

在第一个地址字节传输中，除了 7 位地址数据之外，主设备也会发送 1 位数据以指示数据传输方向。当该传送方向指示位为 0 时，表示主设备向从设备发送数据。当该传送方向指示位为 1 时，表示主设备接收从设备数据。

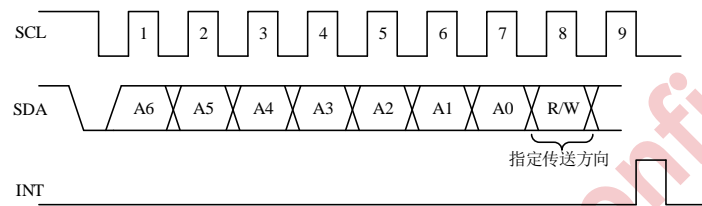


图 248 传输方向控制位

21.4.3.4 应答 (ACK)

ACK 用于检查发送和接收方的串行数据的状态。

每收到 8 位数据，接收方都会返回 ACK 信号。

每发送 8 位数据，发送方通常都会接收 ACK 信号。当从接收方返回 ACK 信号时，将假定接收已正确完成，继续处理。

主机完成数据接受后产生停止条件。从机接收时，如果主机接收到数据后的 NACK 信号，主设备会输出停止条件实现停止传输，或发送开始条件重新开始传输。不返回 ACK 信号的可能原因如下所示。

- 没有正常执行接收
- 已结束最后数据的接收
- 地址指定的接收方不存在

接收方通过在第 9 个时钟处使 SDA 线路处于低电平来产生 ACK (正常接收)；

无论作为主机还是从机，在作为数据接收方后发送的 ACK 值由 I²C 控制寄存器 I2CS_ACKE 决定。

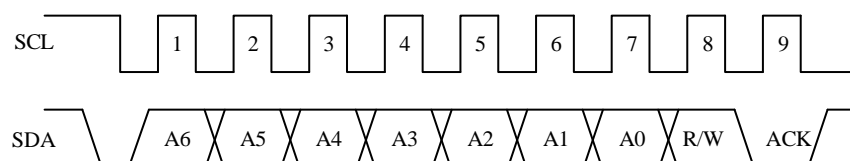


图 249 ACK 信号

21.4.3.5 停止条件

当 SCL 引脚保持高电平时并且 SDA 引脚由低电平变为高电平，即产生一个停止条件。停止条件是在完成串行传输时主设备向从设备发送的一种信号。当器件用作从设备时，可以检测到停止条件。

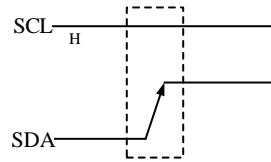


图 250 停止条件

21.4.3.6 等待

等待用来通知总线上其他期间，该器件（主设备或从设备）已经准备就绪可以收发数据（即：处于等待状态）。将 SCL 引脚设置为低电平，通知通信伙伴本机已经处于等待状态。当主设备和从设备的等待状态都被取消时，可以开始下一次传送。

可通过配置 I2CS_WTIM 寄存器控制等待行为出现在第 8 个时钟下降沿或者第 9 个时钟下降沿，等待阶段可以对发送或接收的数据进行处理，通过 I2CS_WRETR 寄存器可以退出等待，解除对 SCL 引脚的低电平设置。

21.4.3.7 中断请求

打开对应中断使能并发生以下情况时会产生中断请求，同时对应的中断标志位会置 1。

地址数据发送/接收期间

- 地址中断
 - 作为主设备，完成地址发送产生中断
 - 作为从设备，接收到与本地地址匹配的地址后产生中断
 - 打开地址屏蔽模式，作为从设备完成地址接收后产生中断
- 数据中断
 - 作为数据发送方，完成数据发送产生中断
 - 作为数据接收方，完成数据接收产生中断
 - 数据中断发生在 8th 时钟还是 9th 时钟由 I2CS_WTIM 位控制
- 停止条件中断
 - 检测到总线出现停止条件后产生中断

传输过程中 FIFO 状态

- 数据传输完成中断
 - 当传输数据的个数与预设值相等时产生中断
- FIFO 空中断
 - FIFO 为空时产生中断
- FIFO 满中断
 - FIFO 为满时产生中断
- FIFO 预警中断
 - 当数据传输方向是发送时，FIFO 数据个数小于等于预警值时产生中断
 - 当数据传输方向是接收时，FIFO 数据个数大于等于预警值时产生中断

传输过程异常状态

- 无应答中断
 - 作为发送方未检测到应答信号时产生中断
- 仲裁失败中断
 - 当发送数据与总线状态不符时产生中断
- 电平超时中断
 - 当总线保持某一状态时间超过阈值时产生中断

21.4.4 I2CS 中断号

表 341 I²C 中断号

名称	中断号
I2CS0_INT	10
I2CS1_INT	11

21.5 寄存器

I2CS0 寄存器的基地址为 0x4000_5400，I2CS1 寄存器的基地址为 0x4000_5800，下表为 I2CS 的各控制寄存器描述。

表 342 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x000	I2CS 数据寄存器	32	r/w	0x00000000
0x004	I2CS 从机地址寄存器	32	r/w	0x00000000
0x008	I2CS 设备地址寄存器	32	r/w	0x00000000
0x00C	I2CS 移位寄存器	32	r/w	0x00000000
0x010	I2CS 控制寄存器	32	r/w	0x07000000
0x014	I2CS 时钟分频寄存器	32	r/w	0x00000000
0x018	I2CS 电平宽度寄存器	32	r/w	0x00FF00FF
0x01C	I2CS 输出控制寄存器	32	r/w	0x00000030
0x020	I2CS FIFO 控制寄存器	32	r/w	0x00000000
0x024	I2CS FIFO 状态寄存器	32	r/w	0x00000000
0x028	I2CS 数据个数寄存器	32	r/w	0x00000001
0x02C	I2CS SDA 延迟寄存器	32	r/w	0x00000000
0x030	I2CS 状态寄存器	32	ro	0x00000000
0x034-0x03C	Reserved	32	ro	0x00000000
0x040	I2CS 中断控制寄存器	32	r/w	0x00000000
0x044	I2CS 中断状态寄存器	32	r/w	0x00000000
0x048-0x04C	Reserved	32	ro	0x00000000
0x050	I2CS 超时控制寄存器	32	r/w	0x00000000
0x054	I2CS 超时阈值寄存器	32	r/w	0x00000000
0x058~0x0FC	Reserved	32	res	0x00000000

21.5.1 I2CS 数据寄存器 (I2CS_DATA)

偏移地址: 0x0000

表 343 I2CS 数据寄存器

位	访问	描述
31:9	Res	Reserved 复位值: 0x0
8:0	r/w	I2C_DATA[7:0]: 数据寄存器 写: 在数据方向为发送时, 将发送数据写入该组寄存器 读: 在数据方向为接收时, 从该组寄存器读出接收数据 复位值: 0x0

21.5.2 I2CS 从机地址寄存器 (I2CS_SLAD)

偏移地址: 0x0004

表 344 I2CS 从机地址寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:0	r/w	I2C_SLAD[14:0]: 从机地址寄存器 该寄存器用于存放作为主机时发送的从机地址, 当进行 7bit 地址通信时, 低 7 位有效, 当进行 10bit 地址通信时, 高 7bit 用于第一地址字节发送, 低 8bit 用于第二地址字节发送。 复位值: 0x0

21.5.3 I2CS 设备地址寄存器 (I2CS_DVAD)

偏移地址: 0x0008

表 345 I2CS 设备地址寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:0	r/w	I2C_DVAD[14:0]: 从设备地址寄存器 该寄存器用于存放作为从机时的本设备地址, 当进行 7 位地址通信时, 低 7 位有效, 当进行 10 位地址通信时, 高 7bit 用于第一地址字节匹配, 低 8bit 用于第二地址字节匹配, 默认地址匹配后设备会自动回复 ACK。 复位值: 0x0

21.5.4 I2CS 移位寄存器 (I2CS_SHF)

偏移地址: 0x000C

表 346 I2CS 移位寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
7:0	ro	I2C_SHF [7:0]: 移位寄存器 可通过该寄存器实时读取移位寄存器数值; 复位值: 0x0

21.5.5 I2CS 控制寄存器 (I2C_CFG)

偏移地址: 0x0010

表 347 I2CS 控制寄存器

位	访问	描述
31:27	Res	Reserved 复位值: 0x0
26:24	r/w	DATA_LEN: 数据长度位宽 (不包含ACK) 0 0 0: 1 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits 复位值: 0x7
23:17	Res	Reserved 复位值: 0x0
16	r/w	DMA_EN: DMA 使能 0: DMA关闭 1: DMA打开 复位值: 0x0
15:14	Res	Reserved 复位值: 0x0
13	r/w	ADDR_MASK: 地址屏蔽 0: 关闭地址屏蔽功能 1: 打开地址屏蔽功能 注: 当打开该位后, 无论总线上主机发出什么地址都会在第 8th 时钟产生中断, 并拉住总线 SCL。 复位值: 0x0

位	访问	描述
12	r/w	DATA_DIR: 数据方向 0: 发送数据 1: 接收数据 复位值: 0x0
11	r/w	I2CS_FLTEN: 滤波使能 0: 关闭 1: 打开 复位值: 0x0
10	Res	Reserved 复位值: 0x0
9	r/w	I2CS_ACKE: 回复使能 0: 禁止应答 1: 应答 复位值: 0x0
8	r/w	I2CS_WTIM: 8/9th 时钟等待/中断选择 0: 8th时钟 1: 9th时钟 复位值: 0x0
7	r/w	I2CS_WRETR退出等待 0: 无效果 1: 退出等待, 继续通信 注: 该寄存器自动清零 复位值: 0x0
6	r/w	I2CS_CRETR退出通信 0: 无效果 1: 退出通信, 释放总线 注: 该寄存器自动清零 复位值: 0x0
5		I2CS_SPT结束触发 0: 无效果 1: 停止条件触发 注: 该寄存器自动清零 复位值: 0x0
4	r/w	I2CS_STT开始触发 0: 无效果 1: 开始条件触发 注: 该寄存器自动清零 复位值: 0x0
3	Res	Reserved 复位值: 0x0

位	访问	描述
2	r/w	I2CS_ADMOD_SEL: 地址模式选择 0: 7bit地址模式 1: 10bit地址模式 复位值: 0x0
1	Res	Reserved 复位值: 0x0
0	r/w	I2CS_EN使能 0: 关闭 I2CS 1: 打开 I2CS 复位值: 0x0

21.5.6 I2CS 时钟分频寄存器 (I2CS_CLK_DIV)

偏移地址: 0x0014

表 348 I2CS 时钟分频寄存器

位	访问	描述
31:10	Res	Reserved 复位值: 0x0
9:0	r/w	I2CS_CLK_DIV[9:0]: 时钟分频寄存器 $I2CS_CLK = (FCLK/2) / (I2CS_CLK_DIV+1)$ 当打开滤波使能后, I2CS_CLK 连续采到两次相同电平算作有效电平。 复位值: 0x0

21.5.7 I2CS 电平宽度寄存器 (I2CS_WTH)

偏移地址: 0x0018

表 349 I2CS 电平宽度寄存器

位	访问	描述
31:24	Res	Reserved 复位值: 0x0
23:16	r/w	I2CS_WH[7:0]: 高电平宽度设置 作为主设备时, 该寄存器用于设置输出的 SCL 引脚信号的高电平宽度; 在高电平设置宽度的 1/2 处进行总线数据采集。 复位值: 0xFF
15:8	Res	Reserved 复位值: 0x0
7:0	r/w	I2CS_WL[7:0]: 低电平宽度设置 作为主设备时, 该寄存器用于设置输出的 SCL 引脚信号的低电平宽度; 在低电平设置宽度的 1/4 处进行总线数据输出。 复位值: 0xFF

21.5.8 I2CS 输出控制寄存器 (I2CS_OUT_CTRL)

偏移地址: 0x001C

表 350 I2CS 输出控制寄存器

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5	r/w	SDA_CLRL_OUT: SDA 编程控制输出 0: SDA输出为低电平 1: SDA输出为开漏输出 复位值: 0x1
4	r/w	SCL_CTRL_OUT: SCL 编程控制输出 0: SCL 输出为低电平 1: SCL 输出为开漏输出 复位值: 0x1
3:2	Res	Reserved 复位值: 0x0
1	r/w	SDA_CTRL_EN: SDA 编程控制输出使能 0: 关闭 SDA 输出控制使能 1: 打开 SDA 输出控制使能 复位值: 0x0
0	r/w	SCL_CTRL_EN: SCL 编程控制输出使能 0: 关闭 SCL 输出控制使能 1: 打开 SCL 输出控制使能 复位值: 0x0

21.5.9 I2CS FIFO 控制寄存器 (I2CS_FIFO_CTRL)

偏移地址: 0x0020

表 351 I2CS FIFO 控制寄存器

位	访问	描述
31:9	Res	Reserved 复位值: 0x0
8	r/w	FIFO_CLR: 清空 FIFO 0: 无效果 1: 清空 FIFO 注: 该寄存器自动清零 复位值: 0x0
7:4	Res	Reserved 复位值: 0x0

位	访问	描述
3:0	r/w	FIFO_THR: FIFO 阈值 0: 0 1: 1 ... 8: 8 9~15: reserved 复位值: 0x0

21.5.10 I2CS FIFO 状态寄存器 (I2CS_FIFO_ST)

偏移地址: 0x0024

表 352 I2CS FIFO 状态寄存器

位	访问	描述
31:11	Res	Reserved 复位值: 0x0
10	Ro	FIFO_THR: FIFO 预警状态 0: FIFO 未处于预警状态 1: FIFO 处于预警状态 复位值: 0x0
9	ro	FIFO_FULL: FIFO 满状态 0: FIFO 非满 1: FIFO 满 复位值: 0x0
8	ro	FIFO_EMPTY: FIFO 空状态 0: FIFO 非空 1: FIFO 空 复位值: 0x0
7:4	Res	Reserved 复位值: 0x0
3:0	ro	FIFO_NUM: FIFO 有效数据个数 0: FIFO 空 1: FIFO 中存在 1 笔有效数据 8: FIFO 中存在 8 笔有效数据 9-15: Reserved 复位值: 0x0

21.5.11 I2CS 数据个数寄存器 (I2CS_DATA_CNT)

偏移地址: 0x0028

表 353 I2CS 数据个数寄存器

位	访问	描述
31:16	ro	TRANS_NUM: 当前数据传输个数 复位值: 0x0
15:0	r/w	I2CS_DATA_CNT: I2CS 数据传输个数寄存器 该寄存器用于配置总共需要传输的数据字节数 复位值: 0x1

21.5.12 I2CS SDA 延迟寄存器 (I2CS_SDA_DLY)

偏移地址: 0x002C

表 354 I2CS SDA 延迟寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7:0	r/w	I2CS_SDA_DLY: I2CS SDA 数据延迟 该寄存器用于配置 SDA 输出相对延后多少个 I2CS_DIV_CLK 时钟 复位值: 0x0

21.5.13 I2CS 状态寄存器 (I2CS_ST)

偏移地址: 0x0030

表 355 I2CS 状态寄存器

位	访问	描述
31:18	Res	Reserved 复位值: 0x0
17	ro	DATA_DONE: 数据传输个数达到预设值 0: 未达到预设值 1: 达到预设值 复位值: 0x0
16	ro	FRM_DONE: 帧结束标志 0: 数据/地址传送阶段 1: 数据/地址传送完成 注: 在检测到 SCL 上升沿时, 该位清零 复位值: 0x0
15:10	Res	Reserved 复位值: 0x0

位	访问	描述
11	ro	I2CS_ALD: 发生仲裁失败 0: 该状态表示不存在总线仲裁, 或仲裁结果为“成功”。 1: 该状态表示仲裁结果为“失败”, 总线当前状态与输出不符 注: 退出通信或者检测到开始条件停止条件后, 该位自动清零。 复位值: 0x0
10	ro	I2CS_WAIT: I2CS 等待状态 0: 未处于等待状态 1: 正处于等待状态 复位值: 0x0
9	ro	I2CS_DAD: SDA 引脚电平的检测 (仅限 I2CS_EN = 1 时有效) 0: SDA 引脚检测为低电平。 1: SDA 引脚检测为高电平。 复位值: 0x0
8	ro	I2CS_CLD: SCL 引脚电平的检测 (仅限 I2CS_EN = 1 时有效) 0: SCL 引脚检测为低电平。 1: SCL 引脚检测为高电平。 复位值: 0x0
7	ro	I2CS_SPD: 停止条件的检测 0: 未检测到停止条件。 1: 检测到停止条件。终结主通信, 并且释放总线。 复位值: 0x0
6	ro	I2CS_STD: 开始条件的检测 0: 未检测到开始条件。 1: 检测到开始条件。这表示处于地址传送期间。 复位值: 0x0
5	ro	I2CS_ACKD: 应答的检测 (ACK) 0: 未检测到应答。 1: 检测到应答。 复位值: 0x0
4	Res	Reserved 复位值: 0x0
3	ro	I2CS_COI: 匹配地址的检测 0: 地址不匹配。 1: 地址匹配。当接收到的地址和本地地址 (SVA) 匹配时 (在第8 个时钟的上升沿置1) 复位值: 0x0
2	Res	Reserved 复位值: 0x0
1	ro	I2CS_MST: 主设备检查标志 0: 从设备或通信待机状态 1: 主设备状态

位	访问	描述
		复位值: 0x0
0	ro	I2CS_BSY: I2C 总线状态标志 0: 总线释放状态 1: 总线通信状态 复位值: 0x0

21.5.14 I2CS 中断控制寄存器 (I2C_INT_EN)

偏移地址: 0x0040

表 356 I2CS 中断控制寄存器

位	访问	描述
31:11	Res	Reserved 复位值: 0x0
10	r/w	FIFO_THRE_INT_EN: FIFO 预警中断使能 0: 关闭中断使能 1: 打开中断使能 注: 当数据方向为发送时, FIFO 低于等于阈值发生中断; 当数据方向为接收时, FIFO 高于等于阈值发生中断。 复位值: 0x0
9	r/w	FIFO_FULL_INT_EN: FIFO 满中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
8	r/w	FIFO_EMPTY_INT_EN: FIFO 空中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
7	Res	Reserved 复位值: 0x0
6	r/w	TCNT_INT_EN: 传输计数完成中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
5	r/w	TIMO_INT_EN: 超时中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0

位	访问	描述
4	r/w	LALD_INT_EN: 仲裁失败中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
3	r/w	NACK_INT_EN: 未接收到应答中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
2	r/w	STOP_INT_EN: 停止条件中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
1	r/w	DATA_INT_EN: 数据中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
0	r/w	ADDR_INT_EN: 地址中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0

21.5.15 I2CS 中断状态寄存器 (I2C_INT_ST)

偏移地址: 0x0044

表 357 I2CS 中断状态寄存器

位	访问	描述
31:11	Res	Reserved 复位值: 0x0
10	r/w	FIFO_THRE_INT: FIFO 预警中断请求 0: 未发生中断请求 1: 发生中断请求 注: 当数据方向为发送时, FIFO 低于阈值发生中断; 当数据方向为接收时, FIFO 高于阈值发生中断。 复位值: 0x0
9	r/w	FIFO_FULL_INT: FIFO 满中断请求 0: 未发生中断请求 1: 发生中断请求 复位值: 0x0
8	r/w	FIFO_EMPTY_INT: FIFO 空中断请求 0: 未发生中断请求

位	访问	描述
		1: 发生中断请求 复位值: 0x0
7	Res	Reserved 复位值: 0x0
6	r/w	TCNT_INT: 传输计数完成中断请求 0: 未发生中断请求 1: 发生中断请求 复位值: 0x0
5	r/w	TIMO_INT: 超时中断请求 0: 未发生中断请求 1: 发生中断请求 复位值: 0x0
4	r/w	LALD_INT: 仲裁失败中断请求 0: 未发生中断请求 1: 发生中断请求 复位值: 0x0
3	r/w	NACK_INT: 未接收到应答中断请求 0: 未发生中断请求 1: 发生中断请求 复位值: 0x0
2	r/w	STOP_INT: 停止条件中断请求 0: 未发生中断请求 1: 发生中断请求 复位值: 0x0
1	r/w	DATA_INT: 数据中断请求 0: 未发生中断请求 1: 发生中断请求 复位值: 0x0
0	r/w	ADDR_INT: 地址中断请求 0: 未发生中断请求 1: 发生中断请求 复位值: 0x0

21.5.16 I2CS 超时控制寄存器 (I2CS_TMOUT_CFG)

偏移地址: 0x0050

表 358 I2CS 超时控制寄存器

位	访问	描述
31:3	Res	Reserved 复位值: 0x0

位	访问	描述
2	r/w	I2CS_LTMO_EN: 低电平超时使能 0: 关闭低电平超时检测使能 1: 打开低电平超时检测使能 复位值: 0x0
1	r/w	I2CS_HTMO_EN: 高电平超时使能 0: 关闭高电平超时检测使能 1: 打开高电平超时检测使能 复位值: 0x0
0	r/w	I2CS_TMO_EN: 超时检测使能 0: 关闭超时检测 1: 打开超时检测 复位值: 0x0

21.5.17 I2CS 超时阈值寄存器 (I2CS_TMOUT_CNT)

偏移地址: 0x0054

表 359 I2CS 超时阈值寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15:0	r/w	I2CS_TMO_CNT: 超时报警 当高/低电平时间超过 I2CS_TMO_CNT个I2CS_CLK 时钟后, 发生超时报警, 如需要更长时间的报警检测, 可与 BTM 进行联合使用。 复位值: 0xFFFF

22 串口总线 (SPI)

22.1 概述

串行外设接口 (SPI) 模块可以给 MCU 提供与其他微处理器、数模转换器、传感器或存储器等设备之间的全双工同步串行通信。

22.2 主要功能

- 支持主机模式和从机模式
- 支持全双工模式同时可实现半双工操作
- 支持单线模式收发 (主机情况下使用 MOSI, 从机情况下使用 MISO)
- 可配置通信波特率
- 支持两个片选信号
- 支持优先高位发送或低位发送
- 支持 DMA 数据接收及发送
- 拥有 4X32 的发送和接收缓存器
- 可编程 1-32bit 数据帧长度
- 片选信号时序可编程控制
- 每帧数据时间间隔可编程控制
- 支持摩托罗拉 SPI 协议, 四种工作模式可选择
- 支持片选及 SCK 可编程控制
- 支持 3 线 SPI 模式 (SCK/MOSI/MISO)

22.3 模块框图

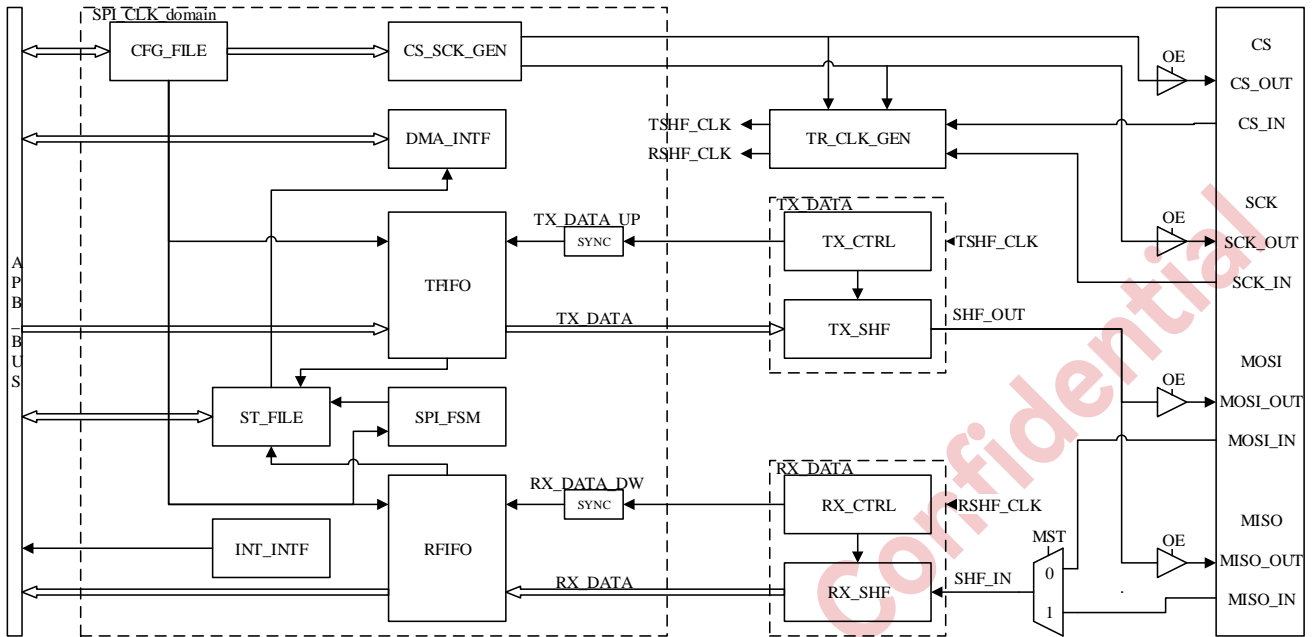


图 251 SPI 模块框图

22.4 功能描述

22.4.1 SPI 接口功能

SPI 模块可以与其他 MCU 进行双工同步串行通信。可以通过 SPI 状态寄存器来判断 SPI 当前工作状态或者打开中断使能，在中断中完成通信配置操作。SPI 有四个功能引脚：

- 从机选择 (CS)
- 串行时钟 (SCK)
- 主机输出/从机输入 (MOSI)
- 主机输入/从机输出 (MISO)

SPI 可配置为主机或从机与其他设备进行通信，通信同步时钟的相位与极性可以在不同配置下组合成 4 种不同格式的 SPI 通信。

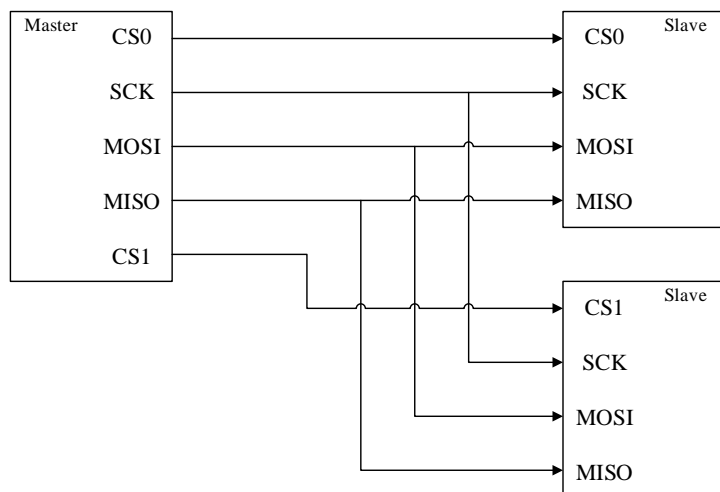


图 252 SPI 一主多从设备连接图

22.4.2 SPI 传输模式

SPI 可实现全双工通信，半双工通信及单工通信的不同工作模式。

SPI 全双工通信：

SPI 使用 MOSI 与 MISO 两根数据线进行数据传输，MOSI 用于主机发送数据及从机接收数据，MISO 用于主机接收数据及从机发送数据。发送数据与接收数据同时进行。

SPI 半双工通信：

SPI 使用 MOSI 与 MISO 两根数据线进行数据传输，MOSI 用于主机发送数据及从机接收数据，MISO 用于主机接受数据及从机发送数据。发送数据与接收数据分时进行，数据传输时钟由主机产生，在需要接收数据时，主机发送空闲数据即可。

SPI 单工通信：

SPI 使用单根数据线进行数据传输，主机情况下使用 MOSI，从机情况下使用 MISO。数据传输方向通过控制寄存器中的传输模式寄存器进行选择。

22.4.3 SPI 工作模式

通过配置 SPI 通信时钟 (SCK) 的时钟极性 (SCK_CPHA) 及时钟相位 (SCK_CPOL) 使 SPI 工作在不同的工作模式下，详见下表。

表 360 SPI 工作模式

SPI 模式	时钟极性	时钟相位	SCK 空闲状态	有效采样沿
Mode0	0	0	低电平	SCK 第一个边沿
Mode1	0	1	低电平	SCK 第二个边沿
Mode2	1	0	高电平	SCK 第一个边沿
Mode3	1	1	高电平	SCK 第二个边沿

SPI 工作模式 0:

该模式下配置时钟极性为 0，时钟相位为 0，SCK 在空闲时表现为低电平，主机从机在 SCK 的第一个 SCK 有效沿进行数据采样。如下图所示，主机在 SCK 的上升沿采 MISO 上从机发送来的数据，在 SCK 下降沿将需要发送的数据体现在 MOSI 上。

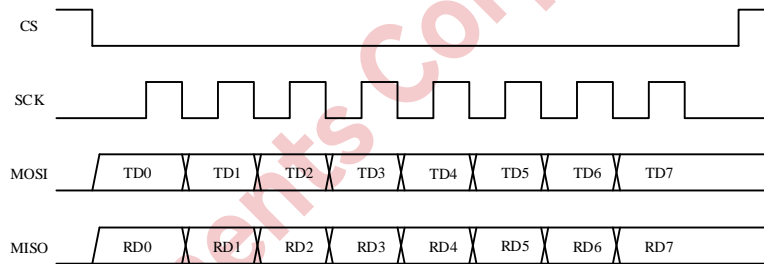


图 253 SPI 模式 0

SPI 工作模式 1:

该模式下配置时钟极性为 0，时钟相位为 1，SCK 在空闲时表现为低电平，主机从机在 SCK 的第二个 SCK 有效沿进行数据采样。如下图所示，主机在 SCK 的下升沿采 MISO 上从机发送来的数据，在 SCK 上降沿将需要发送的数据体现在 MOSI 上。

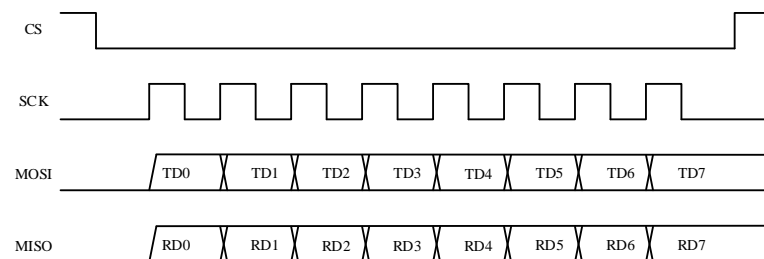


图 254 SPI 模式 1

SPI 工作模式 2:

该模式下配置时钟极性为 1，时钟相位为 0，SCK 在空闲时表现为高电平，主机从机在 SCK 的第一个 SCK 有效沿进行数据采样。如下图所示，主机在 SCK 的下升沿采 MISO 上从机发送来的数据，在 SCK 上降沿将需要发送的数据体现在 MOSI 上。

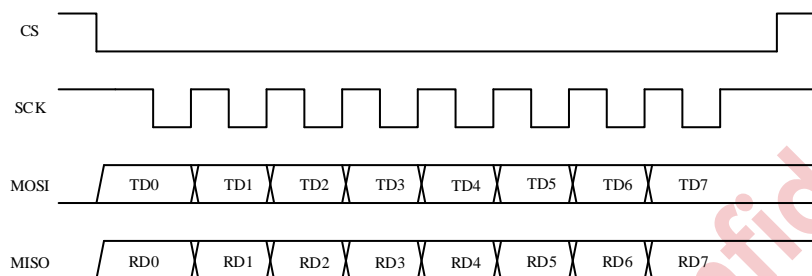


图 255 SPI 模式 2

SPI 工作模式 3:

该模式下配置时钟极性为 1，时钟相位为 1，SCK 在空闲时表现为高电平，主机从机在 SCK 的第二个 SCK 有效沿进行数据采样。如下图所示，主机在 SCK 的上升沿采 MISO 上从机发送来的数据，在 SCK 下降沿将需要发送的数据体现在 MOSI 上。

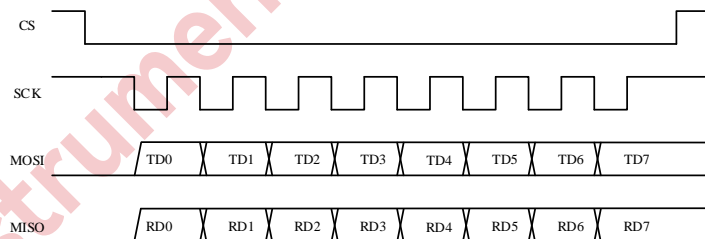


图 256 SPI 模式 3

22.4.4 SPI 通信波特率

SPI 总线通信时钟 SCK 由主机发出，通信波特率由主机控制；

通信传送时钟

$$F_{sck} = F_{spi_clk} / (SPI_BAUD_CTL + 1)$$

22.4.5 SPI 中断号

表 361 SPI 中断号

名称	中断号
SPI_INT	7

22.5 寄存器

SPI 寄存器的基地址为 0x4000_2000，下表为 SPI 的各控制寄存器描述。

表 362 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x000	SPI 数据寄存器	32	r/w	0x00000000
0x004-0x00C	Reserved	32	res	0x00000000
0x010	SPI 控制寄存器	32	r/w	0x00000000
0x014	SPI 帧格式寄存器	32	r/w	0x00000000
0x018	SPI 帧长度寄存器	32	r/w	0x00000000
0x01C	Reserved	32	res	0x00000000
0x020	SPI 通信波特率寄存器	32	r/w	0x00000003
0x024	SPI 片选控制寄存器	32	r/w	0x00030001
0x028	SPI 片选时序寄存器	32	r/w	0x00000000
0x02C	SPI 时钟控制寄存器	32	r/w	0x00000001
0x030	SPI FIFO 控制寄存器	32	r/w	0x00000000
0x034	SPI FIFO 状态寄存器	32	ro	0x00000000
0x038	SPI 状态寄存器	32	ro	0x00040700
0x03C	Reserved	32	res	0x00000000
0x040	SPI DMA 发送寄存器	32	r/w	0x00000000
0x044	SPI DMA 接收寄存器	32	r/w	0x00000000
0x048-0x04C	Reserved	32	res	0x00000000
0x050	SPI 中断控制寄存器	32	r/w	0x00000000
00054	SPI 中断状态寄存器	32	r/w	0x00000000

22.5.1 SPI 数据寄存器 (SPI_DATA)

偏移地址: 0x0000

表 363 SPI 数据寄存器

位	访问	描述
31:0	r/w	SPI_DATA: SPI 数据寄存器 写: 向 TFIFO 中填写数据 读: 从 RFIFO 中读取数据 复位值: 0x0

22.5.2 SPI 控制寄存器 (SPI_CFG)

偏移地址: 0x0010

表 364 SPI 配置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:12	r/w	FLT_SEL: 滤波时钟选择 $F_{flt} = F_{SCLK} / (FLT_SEL + 1)$ 复位值: 0x0
11	r/w	FLT_EN: 输入引脚滤波使能 (CS/SCK) 1'b0: 关闭输入滤波功能 1'b1: 打开输入滤波功能 复位值: 0x0
10	Res	Reserved 复位值: 0x0
9:8	r/w	TR_MOD: 传输模式 2'b00: 全双工模式收发 2'b01: Reserved 2'b10: 单线仅发送 2'b11: 单线仅接收 注: 当使用单线模式时, 主机使用 MOSI 进行通信, 从机使用 MISO 进行通信 复位值: 0x0
7	Res	Reserved 复位值: 0x0
6	r/w	SCK_CPOL: 通信时钟极性选择 0: 时钟极性为 0 1: 时钟极性为 1 注: 时钟极性配置需与其他通信设备相同 复位值: 0x0

位	访问	描述
5	r/w	SCK_CPHA: 通信时钟相位选择 0: 时钟第一个边沿为有效沿 1: 时钟第二个边沿为有效沿 注:时钟相位配置需与其他通信设备相同 复位值: 0x0
4	r/w	DATA_SHF_DIR: 数据移位方向 0: 高位先发送 1: 低位先发送 复位值: 0x0
3	r/w	MST_SLV_SEL: 主机从机选择 0: SPI 模块作为从机工作 1: SPI 模块作为主机工作 复位值: 0x0
2:1	Res	Reserved 复位值: 0x0
0	r/w	SPI_EN: SPI 模块使能 0: SPI 模块关闭 1: SPI 模块开启 复位值: 0x0

22.5.3 SPI 帧格式寄存器 (SPI_FRM_FMT)

偏移地址: 0x0014

表 365 SPI 帧格式寄存器

位	访问	描述
31:16	r/w	DATA_FRM_CNT: 数据帧传输个数 注: 当 MULT_DFRM_EN = 1 时该寄存器生效; 总传输数据帧个数为 DATA_FRM_CNT+1 复位值: 0x0
15:1	Res	Reserved 复位值: 0x0
0	r/w	MULT_DATA_EN: 多数据帧个数使能 0: 数据帧仅有一帧 1: 超过一帧数据传输 注: 在数据传输仅有一帧时, 将该位配置为 0。 复位值: 0x0

22.5.4 SPI 帧长度寄存器 (SPI_FRM_WIDTH)

偏移地址: 0x0018

表 366 SPI 帧长度寄存器

位	访问	描述
31:5	Res	Reserved 复位值: 0x0
4:0	r/w	DATA_FRM_WIDTH: 数据帧位宽 传输的每个数据帧长度为 (DATA_FRM_WIDTH+1) bit, 支持 1-32bit 位宽选择 注: 当数据帧小于 32bit 时, 接收到的数据会在接收缓存器中右对齐存储, 高位补 0。 复位值: 0x0

22.5.5 SPI 通信波特率寄存器 (SPI_BAUD)

偏移地址: 0x0020

表 367 SPI 通信波特率寄存器

位	访问	描述
31:12	Res	Reserved 复位值: 0x0
11:0	r/w	SPI_BAUD: 通信波特率设置 该寄存器用于配置通信时钟 SCK, 最小配置为 1 $F_{sck} = F_{spi_clk} / SPI_BAUD_CTL + 1$ 注: SPI_BAUD_CTL 不可配置为 0 复位值: 0x3

22.5.6 SPI 片选控制寄存器 (SPI_SSLV_CTL)

偏移地址: 0x0024

表 368 SPI 片选控制寄存器

位	访问	描述
31:18	Res	Reserved 复位值: 0x0
17	r/w	SS1_OUT_VAL: 片选 1 输出数值 0: 片选 1 输出低 1: 片选 1 输出高 注: 仅在 SS1_PROG_EN 打开后有效 复位值: 0x1
16	r/w	SS0_OUT_VAL: 片选 0 输出数值 0: 片选 0 输出低 1: 片选 0 输出高 注: 仅在 SS0_PROG_EN 打开后有效

位	访问	描述
		复位值: 0x1
15:10	Res	Reserved 复位值: 0x0
9	r/w	SS1_PROG_EN: 片选 1 可编程使能 0: 关闭编程使能 1: 打开编程使能 注: 片选信号硬件控制与软件控制不可同时开启 复位值: 0x0
8	r/w	SS0_PROG_EN: 片选 0 可编程使能 0: 关闭编程使能 1: 打开编程使能 注: 片选信号硬件控制与软件控制不可同时开启 复位值: 0x0
7:4	Res	Reserved 复位值: 0x0
1	r/w	SS1_HDW_EN: 片选 1 硬件控制使能 0: 关闭硬件控制使能 1: 打开硬件控制使能 注: 片选信号硬件控制与软件控制不可同时开启 复位值: 0x0
0	r/w	SS0_HDW_EN: 片选 0 硬件控制使能 0: 关闭硬件控制使能 1: 打开硬件控制使能 注: 片选信号硬件控制与软件控制不可同时开启 复位值: 0x1

22.5.7 SPI 片选时序寄存器 (SPI_SS_TIM)

偏移地址: 0x0028

表 369 SPI 片选时序寄存器

位	访问	描述
31:24	Res	Reserved 复位值: 0x0

位	访问	描述
23:16	r/w	MIN_FRM_DLY: 数据帧之间延迟时间 8'h0: 无额外要求 8'h1: 多插入 1 SCK 8'h2: 多插入 2 SCK ... 8'hFF: 多插入 255 SCK 复位值: 0x0
15:8	r/w	HLD_TIME: 片选保持时间 8'h0: 无额外要求 8'h1: 多插入 1 SCK 8'h2: 多插入 2 SCK ... 8'hFF: 多插入 255 SCK 复位值: 0x0
7:0	r/w	SET_TIME: 片选建立时间 8'h0: 无额外要求 8'h1: 多插入 1 SCK 8'h2: 多插入 2 SCK ... 8'hFF: 多插入 255 SCK 复位值: 0x0

22.5.8 SPI 时钟控制寄存器 (SPI_SCK_CTL)

偏移地址: 0x002C

表 370 SPI 时钟控制寄存器

位	访问	描述
31:17	Res	Reserved 复位值: 0x0
16	r/w	SCK_OUT_VAL: 通信时钟输出数值 0: 片选 0 输出低 1: 片选 0 输出高 注: 仅在 SCK_PROG_EN 打开后有效 复位值: 0x0
15:9	Res	Reserved 复位值: 0x0
8	r/w	SCK_PROG_EN: 通信时钟可编程使能 0: 关闭编程使能 1: 打开编程使能 注: 时钟信号硬件控制与软件控制不可同时开启

位	访问	描述
		复位值: 0x0
7:1	Res	Reserved 复位值: 0x0
0	r/w	SCK_HDW_EN: 使能 0: 关闭硬件控制使能 1: 打开硬件控制使能 注: 时钟信号硬件控制与软件控制不可同时开启 复位值: 0x1

22.5.9 SPI FIFO 控制寄存器 (SPI_FIFO_CTL)

偏移地址: 0x0030

表 371 SPI FIFO 控制寄存器

位	访问	描述
31:20	Res	Reserved 复位值: 0x0
24	r/w	RFIFO_CLR: 发送 FIFO 清空操作 对该寄存器写 1 可使接收 FIFO 内部清空, 有效数据个数为 0 注: 该寄存器写 1 清零。 复位值: 0x0
23:20	Res	Reserved 复位值: 0x0
19:16	r/w	RFIFO_THR: 接收缓存阈值 设置接收 FIFO 存有数据个数的报警阈值, 当接收缓存器有效数据个数等于或大于阈值时并在中断使能打开后会触发接收 FIFO 阈值报警中断请求。 复位值: 0x0
15:9	Res	Reserved 复位值: 0x0
8	r/w	TFIFO_CLR: 发送 FIFO 清空操作 对该寄存器写 1 可使发送 FIFO 内部清空, 有效数据个数为 0 注: 该寄存器写 1 清零。 复位值: 0x0
7:4	Res	Reserved 复位值: 0x0
3:0	r/w	TFIFO_THR: 发送缓存个数阈值 设置发送 FIFO 存有数据个数的报警阈值, 当发送缓存器有效数据个数等于或小于阈值时并在中断使能打开后会触发发送 FIFO 阈值报警中断请求。 复位值: 0x0

22.5.10 SPI FIFO 状态寄存器 (SPI_FIFO_ST)

偏移地址: 0x0034

表 372 SPI FIFO 状态寄存器

位	访问	描述
31:20	Res	Reserved 复位值: 0x0
19:16	ro	RFIFO_LVL: 接收缓存器有效数据个数 该寄存器表示当前接收 FIFO 中存有的有效数据的个数 复位值: 0x0
15:4	Res	Reserved 复位值: 0x0
3:0	ro	TFIFO_LVL: 发送缓存器有效数据个数 该寄存器表示当前发送 FIFO 中存有的有效数据的个数 复位值: 0x0

22.5.11 SPI 状态寄存器 (SPI_ST)

偏移地址: 0x0038

表 373 SPI 状态寄存器

位	访问	描述
31:19	Res	Reserved 复位值: 0x0
18	ro	RFIFO_THR_WR: 接收 FIFO 阈值报警标志 0: 接收 FIFO 中有效数据的个数未达到报警阈值 1: 接收 FIFO 中有效数据的个数等于或大于报警阈值 复位值: 0x1
17	ro	RFIFO_NEMPTY: 接收 FIFO 非空标志 0: 接收 FIFO 空 1: 接收 FIFO 非空 复位值: 0x0
16	ro	RFIFO_FULL: 接收 FIFO 满标志 0: 接收 FIFO 非满 1: 接收 FIFO 满 复位值: 0x0
15:11	Res	Reserved 复位值: 0x0
10	ro	TFIFO_THR_WR: 发送 FIFO 阈值报警 0: 发送 FIFO 中有效数据的个数未达到报警阈值 1: 发送 FIFO 中有效数据的个数等于或大于报警阈值

位	访问	描述
		复位值: 0x1
9	ro	TFIFO_NFULL: 发送 FIFO 非满 0: 发送 FIFO 满 1: 发送 FIFO 非满 复位值: 0x1
8	ro	TFIFO_EMPTY: 发送缓存器空 0: 发送 FIFO 非空 1: 发送 FIFO 空 复位值: 0x1
7:0	Res	Reserved 复位值: 0x0

22.5.12 SPI DMA 发送寄存器 (SPI_DMA_TX)

偏移地址: 0x0040

表 374 SPI DMA 发送寄存器

位	访问	描述
31:20	Res	Reserved 复位值: 0x0
19:16	r/w	TDMA_LVL: 发送 DMA 触发等级设置 当发送缓存器中有效数据个数小于等于 TDMA_LVL 时请求 DMA 对数据寄存器进行写行为 复位值: 0x0
15:1	Res	Reserved 复位值: 0x0
0	r/w	TDMA_EN: 发送 DMA 触发使能 0: 关闭触发使能 1: 打开触发使能 复位值: 0x0

22.5.13 SPI DMA 接收寄存器 (SPI_DMA_RX)

偏移地址: 0x0044

表 375 SPI DMA 接收寄存器

位	访问	描述
31:20	Res	Reserved 复位值: 0x0
19:16	r/w	RDMA_LVL: 接收 DMA 触发等级 当发送缓存器中有效数据个数大于等于 RDMA_LVL 时请求 DMA 对数据寄存器进行读行为

位	访问	描述
		复位值: 0x0
15:1	Res	Reserved 复位值: 0x0
0	r/w	RDMA_EN: 接收 DMA 触发使能 0: 关闭触发使能 1: 打开触发使能 复位值: 0x0

Kiwi Instruments Corp. Confidential

22.5.14 SPI 中断控制寄存器 (SPI_INT_CTL)

偏移地址: 0x0050

表 376 SPI 中断控制寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15	r/w	TRANS_DONE_INTEN: 传输完成中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
14	r/w	SLV_TXUR_INTEN: 从机模式发送下溢中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
13	r/w	SLV_RXOR_INTEN: 从机模式接收上溢中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
12:11	Res	Reserved 复位值: 0x0
10	r/w	RFIFO_THRWR_INTEN: 接收缓存器阈值报警中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
9	r/w	RFIFO_OVF_INTEN: 接收缓存器溢出中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
8	r/w	RFIFO_UDF_INTEN: 接收缓存器下溢中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
7	Res	Reserved 复位值: 0x0
6	r/w	TFIFO_THRWR_INTEN: 发送缓存器阈值报警中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
5	r/w	TFIFO_OVF_INTEN: 发送缓存器溢出中断使能 0: 关闭中断使能 1: 打开中断使能

位	访问	描述
		复位值: 0x0
4	r/w	TFIFO_UDF_INTEN: 发送缓存器下溢中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	SPI_INTEN: SPI 总中断使能 0: 关闭中断使能 1: 打开中断使能 复位值: 0x0

22.5.15 SPI 中断状态寄存器 (SPI_INT_ST)

偏移地址: 0x0054

表 377 SPI 中断状态寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15	r/w	TRANS_DONE_INT: 传输完成中断标志 0: 未发生 1: 发生 注: 该状态位写 1 清零 复位值: 0x0
14	r/w	SLV_TXUR_INT: 从机模式发送下溢中断标志 0: 未发生 1: 发生 注: 该状态位写 1 清零 复位值: 0x0
13	r/w	SLV_RXOR_INT: 从机模式接收上溢中断标志 0: 未发生 1: 发生 注: 该状态位写 1 清零 复位值: 0x0
12:11	Res	Reserved 复位值: 0x0

位	访问	描述
10	r/w	RFIFO_THRWR_INT: 接收缓存器阈值报警中断标志 0: 未发生 1: 发生 注: 该状态位写 1 清零 复位值: 0x0
9	r/w	RFIFO_OVF_INT: 接收缓存器溢出中断标志 0: 未发生 1: 发生 注: 该状态位写 1 清零 复位值: 0x0
8	r/w	RFIFO_UDF_INT: 接收缓存器下溢中断标志 0: 未发生 1: 发生 注: 该状态位写 1 清零 复位值: 0x0
7	Res	Reserved 复位值: 0x0
6	Res	TFIFO_THRWR_INT: 发送缓存器阈值报警中断标志 0: 未发生 1: 发生 注: 该状态位写 1 清零 复位值: 0x0
5	r/w	TFIFO_OVF_INT: 发送缓存器溢出中断标志 0: 未发生 1: 发生 注: 该状态位写 1 清零 复位值: 0x0
4	r/w	TFIFO_UDF_INT: 发送缓存器下溢中断标志 0: 未发生 1: 发生 注: 该状态位写 1 清零 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	ro	SPI_INTST: SPI 总中断状态 0: SPI 没有中断请求发生 1: SPI 有中断请求发生 复位值: 0x0

23 模拟比较器 (CMP)

23.1 概述

模拟比较器用于比较两个模拟电压输入，比较器电路设计应用于整个电源电压范围内；共有 4 个模拟比较器，每个比较器可根据使用需求进行比较电压输入档位选择。

23.2 主要功能

- 共有 4 路独立支持 DAC 参考的 CMP
- 电压比较器多输入选择
- 比较器迟滞能力选择
 - 无迟滞
 - 25mV
 - 50mV
 - 100mV
- 支持数字滤波功能
- 支持比较器输出翻转
- 支持 EBUS 触发屏蔽功能
- 支持有效沿触发中断响应
- 支持发送 EBUS 触发事件
- 比较器可输出至 EBUS&TIM&PAD
- 比较器四种输出模式选择
 - 异步输出
 - 同步输出
 - 滤波输出
 - 边沿检测输出
- 12-bit DAC 转换器
- 软件更新 & 硬件更新 DAC
- EBUS 多通道选择触发 DAC 更新
- DAC 斜坡发生器

23.3 模块框图

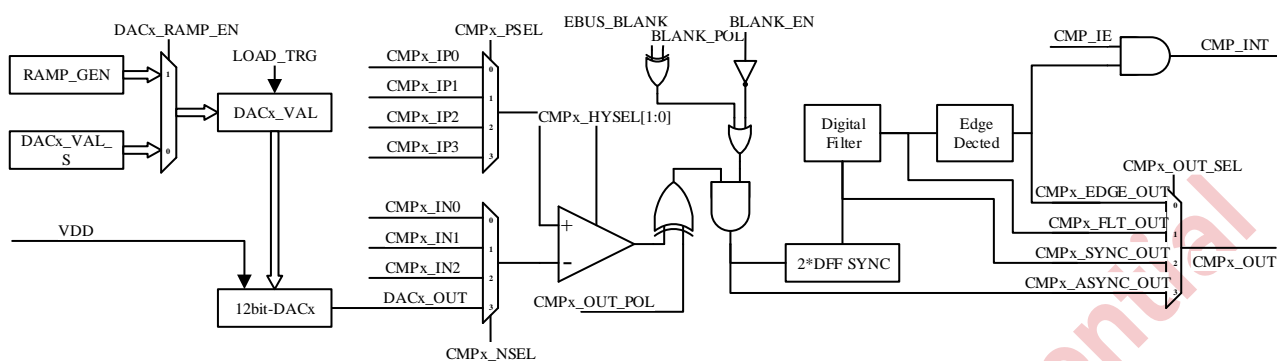


图 257 CMP 模块框图

23.4 功能描述

23.4.1 CMP 工作时序

CMP 可对输入端电压值进行比较，如下图所示，当正端输入电压大于负端输入电压时，比较器输出逻辑高，反之输出逻辑低。负端可以有四个通道进行选择，分别是 DAC 输出及外部 IO。当选择 DAC 输出时，负端电压可通过 DAC 进行控制。

工作时可选择比较器输出上升沿、下降沿或双沿作为比较器的有效输出沿，打开对应中断使能和 **EBUS** 使能，当出现有效沿时会产生对应中断事件和 **EBUS** 事件。

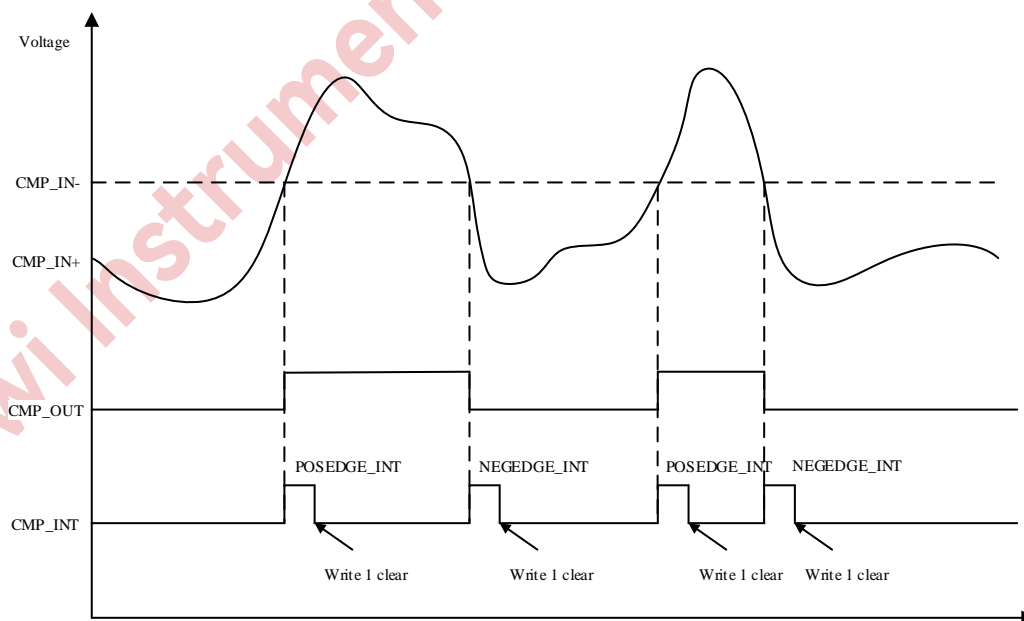


图 258 CMP 工作时序图

23.4.2 CMP 数字滤波

CMP 内部包含 32bit 的 FIFO 用于数字滤波，可选择参与数字滤波的 DFF 数量，在采样时钟的上升沿连续采到目标数量个相同电平信号时认为该信号有效；在配置滤波参数时需关闭滤波使能并清空滤波 FIFO，否则会造成不必要的毛刺干扰。滤波时钟源为 CMP_CLK，可配置对应的分频寄存器进行滤波时钟频率选择。

23.4.3 DAC 电压输出

DAC 输出电压可通过软件和硬件两种模式进行更新，通过 DAC_MD 寄存器进行选择。在使用软件更新输出电压时，向 DAC_CODE 寄存器写入数值即可更新转换电压，使用硬件更新时，先将更新电压值写入 DAC_CODE 寄存器，在 EBUS 触发信号出现后的第三个周期输出电压进行更新。

该 DAC 拥有 12bit 分辨率，相应转换电压公式如下：

$$DAC_{out} = V_{ref} \times CODE_{in} / 4096$$

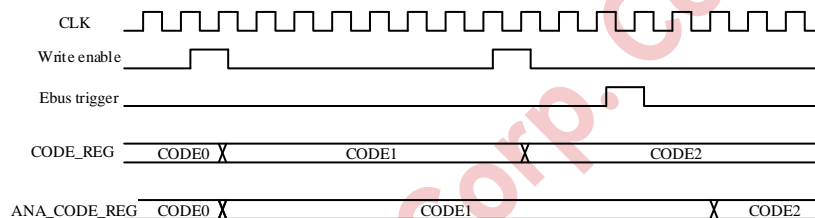


图 259 DAC 输出更新

23.4.4 DAC 斜坡发生

DAC 输出电压可通过软件和硬件两种模式触发斜坡发生器，由预设高电压开始下降，当 CMP 输出出现上升翻转，恢复初始值，并将翻转对应电压值保存。

23.4.5 CMP 中断号

表 378 CMP 中断号

名称	中断号
CMP0_INT	16
CMP1_INT	17
CMP2_INT	18
CMP3_INT	19

23.5 寄存器

CMP0 寄存器的基地址为 0x4000_6000，CMP1 寄存器的基地址为 0x4000_6040，CMP2 寄存器的基地址为 0x4000_6080，CMP3 寄存器的基地址为 0x4000_60C0，下表为 CMP 的各控制寄存器描述。

表 379 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	CMPx 控制寄存器	32	r/w	0x00000108
0x04	CMPx P/N 输入选择寄存器	32	r/w	0x00000003
0x08	CMPx 滤波控制寄存器	32	r/w	0x00000000
0x0c	CMPx 状态寄存器	32	r/w	0x00000000
0x10	DACx 控制寄存器	32	r/w	0x00000000
0x18	DACx 数据寄存器	32	r/w	0x00000000
0x1C	DACx 翻转数值寄存器	32	r/w	0x00000000
0x20	DACx 斜坡延迟寄存器	32	r/w	0x00000000
0x24	DACx 斜坡步长寄存器	32	r/w	0x00000000
0x28	DACx 斜坡斜率寄存器	32	r/w	0x00000000
0x30	CMPx 输入 IO 选择	32	r/w	0x00000000

23.5.1 CMPx 控制寄存器 (CMPx_CTRL)

偏移地址：0x000

表 380 CMPx 控制寄存器

位	访问	描述
31:20	Res	Reserved 复位值：0x0
19:16	r/w	CMPx_EBUS_BLK_SEL: blanking 信号的 EBUS 通道选择 0000: 通道 0 0001: 通道 1 0010: 通道 2 0011: 通道 3 0100: 通道 4 0101: 通道 5 0110: 通道 6 0111: 通道 7 1000: 通道 8 1001: 通道 9 1010: 通道 10 1011: 通道 11 1100: 通道 12 1101: 通道 13 1110: 通道 14 1111: 通道 15 复位值：0x0
15:14	r/w	CMPx_EOUT_SEL: 电压比较器输出至 EBUS 信号类型选择 00: 异步输出 01: 同步输出

位	访问	描述
		10: 滤波输出 11: 边沿检测输出 复位值: 0x0
13:12	r/w	CMPx_POUT_SEL: 电压比较器输出至 PIN 信号类型选择 00: 异步输出 01: 同步输出 10: 滤波输出 11: 边沿检测输出 复位值: 0x0
11:10	r/w	CMPx_TOUT_SEL: 电压比较器输出至 TIM 信号类型选择 00: 异步输出 01: 同步输出 10: 滤波输出 11: 边沿检测输出 复位值: 0x0
9:8	r/w	CMPx_HYSEL: 电压比较器迟滞选择 00: 无迟滞 01: 25mV (default) 10: 50mV 11: 100mV 复位值: 0x1
7	r/w	CMP_FLT_EN: 滤波使能 0: 关闭滤波 1: 打开滤波 复位值: 0x0
6	r/w	CMPx_EDG_DET: 电压比较器边沿检测 0: 单边沿检测 1: 双边沿检测 复位值: 0x0
5	r/w	CMPx_EDG_POL: 电压比较器边沿检测极性 0: 上升沿检测 1: 下降沿检测 注: 该位仅当配置为单边沿模式时有效 复位值: 0x0
4	r/w	CMPx_IE: 电压比较器中断使能 0: 关闭电压比较器中断 1: 打开电压比较器中断 复位值: 0x0
3	r/w	CMPx_BLK_POL: blanking 信号有效电平选择 0: blanking 低电平生效 1: blanking 高电平生效

位	访问	描述
		复位值: 0x1
2	r/w	CMPx_BLK_EN: blanking 使能 0: 关闭 1: 开启 注: 当 blanking 信号生效时, 将输出信号锁定为低电平 复位值: 0x0
1	r/w	CMPx_OUT_POL: 电压比较器输出极性 0: 比较器输出 1: 比较器输出反相 复位值: 0x0
0	r/w	CMPx_EN: 电压比较器使能 0: 关闭电压比较器 1: 打开电压比较器 复位值: 0x0

23.5.2 CMPx P/N 输入选择寄存器 (CMPx_PN_SEL)

偏移地址: 0x004

表 381 CMPx 控制寄存器

位	访问	描述
31:20	Res	Reserved 复位值: 0x0
17:16	r/w	CMPx_PSEL: 电压比较器 P 端输入选择 CMP0 P 端输入选择: 00: P 端选择 XP42 引脚 01: P 端选择 XP21 引脚 10: P 端选择 XP40 引脚 11: P 端选择 XP45 引脚 CMP1 P 端输入选择: 00: P 端选择 XP42 引脚 01: P 端选择 XP21 引脚 10: P 端选择 XP40 引脚 11: P 端选择 XP45 引脚 CMP2 P 端输入选择: 00: P 端选择 XP51 引脚 01: P 端选择 XP44 引脚 10: P 端选择 XP53 引脚 11: P 端选择 XP46 引脚 CMP3 P 端输入选择: 00: P 端选择 XP51 引脚 01: P 端选择 XP44 引脚 10: P 端选择 XP53 引脚 11: P 端选择 XP46 引脚 复位值: 0x0
15:2	Res	Reserved 复位值: 0x0
1:0	r/w	CMPx_NSEL: 电压比较器 N 端输入选择 CMP0 N 端输入选择: 00: N 端选择 XP42 引脚 01: N 端选择 XP20 引脚

位	访问	描述
		10: N 端选择 XP40 引脚 CMP1 N 端输入选择: 00: N 端选择 XP42 引脚 10: N 端选择 XP40 引脚 CMP2 N 端输入选择: 00: N 端选择 XP21 引脚 10: N 端选择 XP53 引脚 CMP3 N 端输入选择: 00: N 端选择 XP21 引脚 10: N 端选择 XP53 引脚 复位值: 0x0
		11: DAC 输出 01: N 端选择 XP20 引脚 11: DAC 输出 01: N 端选择 XP44 引脚 11: DAC 输出 01: N 端选择 XP44 引脚 11: DAC 输出

23.5.3 CMPx 滤波控制寄存器 (CMPx_FLT_CTRL)

偏移地址: 0x0008

表 382 CMPx 滤波控制寄存器

位	访问	描述
31	r/w	CMPx_FLT_CLR: CMPx 滤波 FIFO 清零 CMP 内部采用 32bit DFF 进行滤波采样, 对该寄存器写 1, 将这 32bit DFF 全部清零。 复位值: 0x0
30:29	Res	Reserved 复位值: 0x0
28:24	r/w	CMPx_FLT_WIND: 滤波采样窗口 CMP 内部采用 32bit DFF 进行滤波采样, 该寄存器决定有几个 bit DFF 参与采样滤波判断 复位值: 0x0
23:21	Res	Reserved 复位值: 0x0
20:16	r/w	CMPx_FLT_NUM: 滤波采样有效个数配置 注: 在采样时钟的上升沿连续采到 (CMPx_FLT_NUM+1)个相同电平信号时认为该信号有效; 配置滤波参数时需关闭滤波使能并清空滤波 DFF, 否则会造成不必要的毛刺干扰。 注: 配置时 CMPx_FLT_NUM 需要大于 CMPx_FLT_WIND/2 并小于等于 CMPx_FLT_WIND 复位值: 0x0
15:10	Res	Reserved 复位值: 0x0
9:0	r/w	CMPx_FLT_DIV: 滤波时钟分频 滤波时钟符合 $FLT_CLK = CMP_CLK / (CMPx_FLT_DIV+1)$ 规律 注: 配置滤波参数时需关闭滤波使能, 否则会造成不必要的毛刺干扰 复位值: 0x0

23.5.4 CMPx 状态寄存器 (CMPx_ST)

偏移地址: 0x00C

表 383 CMPx 滤波控制寄存器

位	访问	描述
30:4	Res	Reserved 复位值: 0x0
3	r/w	CMPx_NEG: 下降沿检测标志 0: 未检测到下降沿 1: 检测到下降沿 该状态位写 1 清零, 仅当进行下降沿检测后该位才有效 复位值: 0x0
2	r/w	CMPx_POS: 上升沿发生标志 0: 未检测到上升沿 1: 检测到上升沿 该状态位写 1 清零, 仅当进行上升沿检测后该位才有效 复位值: 0x0
1	r/w	CMPx_INT: 比较器 0 中断标志 0: 未检测到 CMP0 电压符合中断条件 1: 检测到 CMP0 电压符合中断条件 该状态位写 1 清零 复位值: 0x0
0	ro	CMPx_OUT: 比较器 0 实时输出 0: 比较器正端电压 < 参考电压 1: 比较器正端电压 > 参考电压 复位值: 0x0

23.5.5 DACx 控制寄存器 (DACx_CTRL)

偏移地址: 0x010

表 384 DACx 控制寄存器

位	访问	描述
30:20	Res	Reserved 复位值: 0x0
19:16	r/w	DACx_EBUS_SEL: EBUS 通道选择 0000: 通道 0 0001: 通道 1 0010: 通道 2 0011: 通道 3 0100: 通道 4 0101: 通道 5 0110: 通道 6 0111: 通道 7 1000: 通道 8 1001: 通道 9 1010: 通道 10 1011: 通道 11 1100: 通道 12 1101: 通道 13 1110: 通道 14 1111: 通道 15 注: 触发信号上升沿有效 复位值: 0x0
15:5	Res	Reserved 复位值: 0x0
4	r/w	DACx_RAMP_TRG: 斜坡发生器触发 0: 无效 1: 触发斜坡发生 注: 仅在打开斜坡发生器使能时生效 复位值: 0x0
3	r/w	DACx_RAMP_MD: 斜坡发生器模式 0: DACx_RAMP_TRG 触发模式 1: EBUS 触发模式 注: 仅在打开斜坡发生器使能时生效 复位值: 0x0
2	r/w	DACx_RAMP_EN: 斜坡发生器使能 0: 关闭 1: 开启 注: 当开启斜坡发生器使能后, 复位值: 0x0
1	r/w	DACx_MD: D/A 转换模式 0: 正常模式 (数据寄存器更新触发) 1: EBUS 触发模式 复位值: 0x0
0	r/w	DACx_EN: D/A 转换器使能 0: 关闭 D/A 转换器 1: 打开 D/A 转换器 复位值: 0x0

23.5.6 DACx 数据寄存器 (DACx_VAL)

偏移地址: 0x018

表 385 DACx 数据寄存器

位	访问	描述
30:28	Res	Reserved 复位值: 0x0
27:16	r/w	DACx_VAL_B: DACx 数值预装载寄存器 注: 在使用斜坡发生器模式或 DAC 触发模式更新时使用, 该寄存器用于存放预装载数值。 复位值: 0x0
15:12	Res	Reserved 复位值: 0x0
11:0	r/w	DACx_VAL: DACx 数值 该寄存器用于存放当前 DACx 输出电压值, 与输出电压值的关系符合公式: $V_{dacx} = VCC * DACx_CODE / 4096$ 注: 当打开斜坡发生器或 DAC 触发模式更新时, 该寄存器不可写, 需将转换的数据填写至 DACx_VAL_B 中, 当发生触发信号时, 会将 DACx_VAL_B 中的数据自动装载到 DACx_VAL。 复位值: 0x0

23.5.7 DACx 翻转数值寄存器 (DACx_SDAT)

偏移地址: 0x01C

表 386 DACx 翻转数值寄存器

位	访问	描述
30:12	Res	Reserved 复位值: 0x0
11:0	r/w	DACx_SDAT: DACx 翻转数值寄存器 当使用斜坡发生器并检测到 CMPx 出现检测目标边沿时, 会将当前 DAC_VAL 保存至该寄存器, 最终保存值为最后一个有效边沿对应的 DAC_VAL。 注: 对该寄存器进行写操作可以更改该寄存器数值 复位值: 0x0

23.5.8 DACx 斜坡延迟寄存器 (DACx_RAMP_DLY)

偏移地址: 0x020

表 387 DACx 斜坡延迟寄存器

位	访问	描述
31:30	Res	Reserved 复位值: 0x0
29:16	r/w	DACx_RAMP_DLY_B: DACx 斜坡延迟预装载寄存器 注: 在使用斜坡发生器模式时使用, 该寄存器用于存放预装载数值。 复位值: 0x0
15:14	Res	Reserved 复位值: 0x0
13:0	r/w	DACx_RAMP_DLY: DACx 斜坡延迟 该寄存器用于存放触发斜坡发生后当前斜坡经过多少个时钟后 DACx 输出电压值发生第一次变化, 其中的变化关系符合公式: $T_{first_dly} = T_{cmp_clk} * DACx_RAMP_DLY$ 注: DAC 转换时长为 1μs, 步长时间需大于 1μs。 复位值: 0x0

23.5.9 DACx 斜坡步长寄存器 (DACx_RAMP_STEP)

偏移地址: 0x024

表 388 DACx 斜坡步长寄存器

位	访问	描述
31:28	Res	Reserved 复位值: 0x0
27:16	r/w	DACx_RAMP_STEP_B: DACx 斜坡步长预装载寄存器 注: 在使用斜坡发生器模式时使用, 该寄存器用于存放预装载数值。 复位值: 0x0
15:14	Res	Reserved 复位值: 0x0
11:0	r/w	DACx_RAMP_STEP: DACx 斜坡步长 该寄存器用于存放当前斜坡发生 DACx 输出电压值变化时, 每一次变化的数值是多少 复位值: 0x0

23.5.10 DACx 斜坡保持寄存器 (DACx_RAMP_SLOPE)

偏移地址: 0x028

表 389 DACx 斜坡斜率寄存器

位	访问	描述
31:30	Res	Reserved 复位值: 0x0
29:16	r/w	DACx_RAMP_SLOPE_B: DACx 斜坡保持预装载寄存器 注: 在使用斜坡发生器模式时使用, 该寄存器用于存放预装载数值。 复位值: 0x0
15:14	Res	Reserved 复位值: 0x0
13:0	r/w	DACx_RAMP_SLOPE: DACx 斜坡保持 该寄存器用于存放当前斜坡输出电压经过多少个时钟后 DACx 输出电压值发生一次变化, 其中的变化关系符合公式: $T_{dly} = T_{cmp_clk} * DACx_RAMP_SLOPE$ 注: DAC 转换时长为 1us, 步长时间需大于 1μs。 复位值: 0x0

23.5.11 CMPx 输入 IO 选择 (CMPx_ASEL)

偏移地址: 0x0030

表 390 DACx 斜坡斜率寄存器

位	访问	描述
31:20	Res	Reserved 复位值: 0x0
19:16	r/w	CMPx_INP_ASEL: 比较器 P 端 PIN 选择 0000: P 端浮空 ***1: P 端第一个输入使能 **1*: P 端第二个输入使能 *1**: P 端第三个输入使能 1***: P 端第四个输入使能 复位值: 0x0
15:3	Res	Reserved 复位值: 0x0
2:0	r/w	CMPx_INN0_ASEL: 比较器 N 端 PIN 选择 000: N 端引脚无使能 **1: N 端第一个输入使能 *1*: N 端第二个输入使能 1**: N 端第三个输入使能 复位值: 0x0

24 外设事件系统 (EBUS)

24.1 概述

外设事件系统能够通过多种不同的配置,来实现设备输入,输出以及片上外设模块之间的事件进行灵活地互连。事件通过此模块在不同外设之间的传递通信是独立于 CPU 的。片上多个外设支持编程为发送事件/接收事件,产生事件的具体条件以及接收事件后的响应行为取决于外设模块的工作方式。接收事件的外设被称为从设备,发送事件的外设被称作主设备。一个外设模块可以既是从设备同时又是主设备。

通过外设事件系统的通信不需要 CPU 干涉并且不消耗系统资源,比如总线或者 RAM 带宽,与传统的基于中断的方式相比,可以减轻 CPU 的负担以及节省系统资源。

24.2 主要功能

- 时钟与系统时钟同步
- 支持最多 16 个独立可配置的事件通道
- 每个通道事件异步传输模式以及同步采样模式
- 在同步采样模式中,具有以下边沿检测功能
 - 将输入事件源同步后发送
 - 在输入事件源的上升沿产生事件
 - 在输入事件源的下降沿产生事件
 - 在输入事件源的上升和下降沿都产生事件
- 支持软件触发电平或脉冲事件
- 支持相邻通道之间的逻辑组合输出
- 支持发送事件到 CPU 作为 WFE
- 支持最多 16 个独立的 IO 电平检测通道
- 可配置 16 个独立的 IO 检测中断
- 可配置 IO 上升沿或者下降沿检测

24.3 模块框图

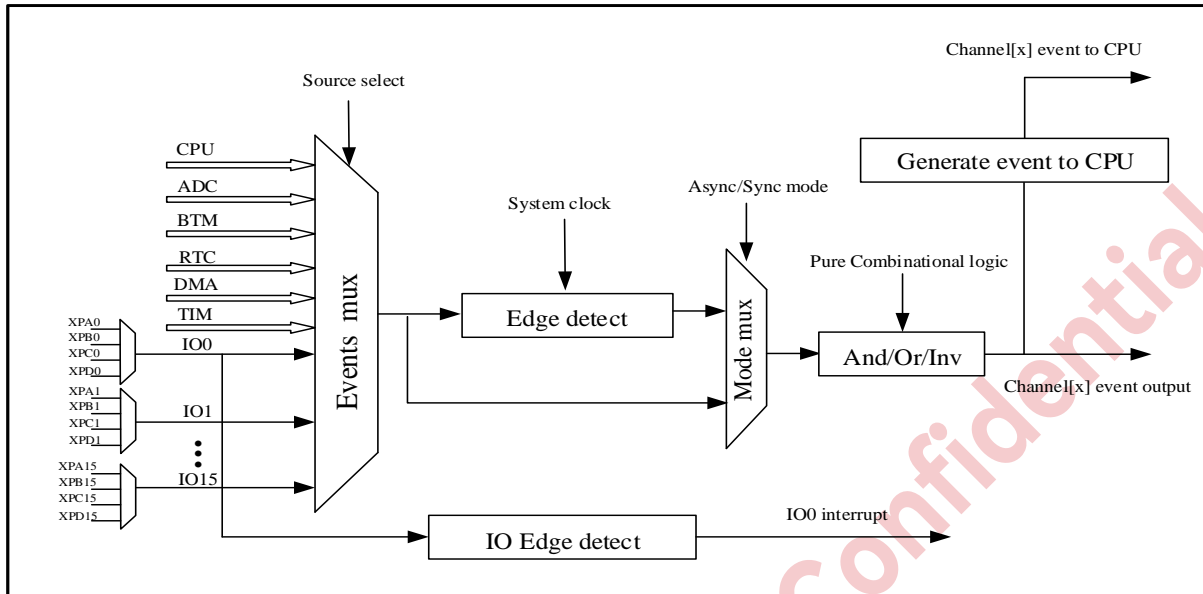


图 260 EBUS 模块架构图

24.4 功能描述

24.4.1 通道工作模式

每个通道可以各自独立地工作在两种模式下，异步模式或同步模式。模式设置控制可以通过通道 X 设置寄存器的 EBUS_ASYNC_EN 信号。

当使用异步模式时，事件在主设备和从设备之间的传播不经过外设事件系统的干涉，此时该模块的时钟可以不开启，也就是事件在主从之间传播没有任何的时钟延时。当异步模式被选择时，边沿检测功能无效。

当使用同步模式时，边沿检测功能被使能，主设备输出的事件会经过同步后，再发送到从设备。

24.4.2 同步输出和输入边沿检测

当使用同步模式时，外设事件系统可以执行同步输出或者边沿检测，通过以下四种不同方式：

- 同步输入事件源后直接输出
- 只在输入事件源的上升沿输出事件
- 只在输入事件源的下降沿输出事件
- 在输入事件源的上升沿或下降沿都输出事件

同步模式的具体方式由通道 X 设置寄存器的 EBUS_DET_SEL[1:0] 来决定，当 EBUS_DET_SEL 不等于 2'B00 时，边沿检测电路被使能，此时，通道输出为系统时钟域的单周期脉冲事件。

24.4.3 事件延时

当通道工作在不同模式下时，事件从主设备到从设备之间的传播延时是不同的。

- 异步模式时，事件延时主要取决于主从设备之间的线路延时
- 同步模式时，事件延时为固定的两个系统时钟

24.4.4 软件事件

CPU 可以控制通道直接发出事件，通过设置软件事件触发寄存器的相应位。设置 CHX_SW_PUL 位时，可以使通道发出一个系统时钟宽度的脉冲事件，设置 CHX_SW_LVL 位时，可以使通道发出一个持续的高电平事件，直到 CPU 将 CHX_SW_LVL 位清除。

当通道所选择事件输入源和软件事件控制同时发生作用时，通道输出是由两者异或之后的结果。例如，当 CPU 设置了 CHX_SW_LVL 后，通道输出保持为高电平，此时当所选择的事件输入源也输入一个高电平事件时，通道输出会变为低电平。

24.4.5 组合逻辑功能

每个通道的输出都可以与前后两个相邻通道的输出进行逻辑组合后变为最终输出。此功能由通道 X 设置寄存器的 EBUS_OR_EN, EBUS_AND_EN, EBUS_INV_EN 位来使能或者关闭。

当使能 EBUS_OR_EN 时，当前通道的输出会与前一通道的输出进行逻辑或；

当使能 EBUS_AND_EN 时，当前通道的输出会与后一通道的输出进行逻辑与；

当使能 EBUS_INV_EN 时，当前通道的输出会对当前值取反；

该功能的先后执行顺序为，逻辑与->逻辑或->取反；

24.4.6 事件触发 CPU

外设事件系统可以单独地发送 RXEV 事件到 CPU，通过事件触发 CPU 设置寄存器进行使能，事件的触发源可以在所有通道中选择一个，此项功能可以和 CPU 的 WFE (Wait for Event) 指令相配合。

24.4.7 IO 事件检测

外设事件系统可以独立地对 16 个 IO 输入进行边沿检测，同时产生独立的 16 个检测中断信号，检测方式有以下三种：

- 输入电平的上升沿检测
- 输入电平的下降沿检测
- 输入电平的上升或者下降沿检测

在 IO 边沿检测设置寄存器中，通过配置 IOX_EDGE_SEL[1:0] 对检测方式进行选择，当输入电平符合检测条件时，相应的 IO 检测结果会以中断/事件的方式发出。

24.4.8 通道事件输入源选择

EBUS 所有通道都有相同的 256 个事件输入源，每个通道的事件输入源可单独配置，配置寄存器在通道 x 设置寄存器中。

表 391 通道事件输入源

输入源序号	事件源	输出事件类型
0	CPU 事件	脉冲 (系统时钟域)
1-3	保留	保留
4	DMA 中断事件	电平 (系统时钟域)
5-7	保留	保留
8	TIMA0 OC0	电平 (系统时钟域)
9	TIMA0 OC1	电平 (系统时钟域)
10	TIMA0 OC2	电平 (系统时钟域)
11	TIMA0 OC3	电平 (系统时钟域)
12	TIMA0 EVT0	电平 (系统时钟域)
13	TIMA0 EVT1	电平 (系统时钟域)
14	TIMA0 TRGO1	电平 (系统时钟域)
15	TIMA0 TRGO	电平 (系统时钟域)
16	TIMA1 OC0	电平 (系统时钟域)
17	TIMA1 OC1	电平 (系统时钟域)
18	TIMA1 OC2	电平 (系统时钟域)
19	TIMA1 OC3	电平 (系统时钟域)
20	TIMA1 EVT0	电平 (系统时钟域)
21	TIMA1 EVT1	电平 (系统时钟域)
22	TIMA1 TRGO1	电平 (系统时钟域)
23	TIMA1 TRGO	电平 (系统时钟域)
24	TIMH eventA	电平 (系统时钟域)
25	TIMH eventB	电平 (系统时钟域)
26	TIMH eventC	电平 (系统时钟域)
27	保留	电平 (系统时钟域)
28	保留	电平 (系统时钟域)
29	保留	电平 (系统时钟域)
30	保留	电平 (系统时钟域)

输入源序号	事件源	输出事件类型
31	保留	电平 (系统时钟域)
32	TIMG0 OC0	电平 (系统时钟域)
33	TIMG0 OC1	电平 (系统时钟域)
34	TIMG0 OC2	电平 (系统时钟域)
35	TIMG0 OC3	电平 (系统时钟域)
36	TIMG0 TRGO	电平 (系统时钟域)
39-37	保留	电平 (系统时钟域)
40	TIMG1 OC0	电平 (系统时钟域)
41	TIMG1 OC1	电平 (系统时钟域)
42	TIMG1 OC2	电平 (系统时钟域)
43	TIMG1 OC3	电平 (系统时钟域)
44	TIMG1 TRGO	电平 (系统时钟域)
47-45	保留	保留
48	TIMG2 OC0	电平 (系统时钟域)
49	TIMG2 OC1	电平 (系统时钟域)
50	TIMG2 OC2	电平 (系统时钟域)
51	TIMG2 OC3	电平 (系统时钟域)
52	TIMG2 TRGO	电平 (系统时钟域)
55-53	保留	保留
56	TIMS0 OC0	电平 (系统时钟域)
57	TIMS0 OC1	电平 (系统时钟域)
58	TIMS0 TRGO	电平 (系统时钟域)
59	保留	保留
60	TIMS1 OC0	电平 (系统时钟域)
61	TIMS1 OC1	电平 (系统时钟域)
62	TIMS1 TRGO	电平 (系统时钟域)
63	保留	保留
64	TIMS2 OC0	电平 (系统时钟域)
65	TIMS2 OC1	电平 (系统时钟域)
66	TIMS2 TRGO	电平 (系统时钟域)
67	保留	保留
68	TIMS3 OC0	电平 (系统时钟域)

输入源序号	事件源	输出事件类型
69	TIMS3 OC1	电平 (系统时钟域)
70	TIMS3 TRGO	电平 (系统时钟域)
71	保留	保留
72	BTM 输出事件 1	电平 (系统时钟域)
73	BTM 输出事件 2	电平 (系统时钟域)
74	BTM 输出事件 3	电平 (系统时钟域)
75	BTM 输出事件 4	电平 (系统时钟域)
87:76	保留	保留
88	CMP0 输出事件	异步事件
89	CMP1 输出事件	异步事件
90	CMP2 输出事件	异步事件
91	CMP3 输出事件	异步事件
95:92	保留	保留
96	ADC0 转换单元 0 完成事件	脉冲 (系统时钟域)
97	ADC0 转换单元 1 完成事件	脉冲 (系统时钟域)
98	ADC0 转换单元 2 完成事件	脉冲 (系统时钟域)
99	ADC0 转换单元 3 完成事件	脉冲 (系统时钟域)
100	ADC0 转换单元 4 完成事件	脉冲 (系统时钟域)
101	ADC0 转换单元 5 完成事件	脉冲 (系统时钟域)
102	ADC0 转换单元 6 完成事件	脉冲 (系统时钟域)
103	ADC0 转换单元 7 完成事件	脉冲 (系统时钟域)
104	ADC0 转换单元 8 完成事件	脉冲 (系统时钟域)
105	ADC0 转换单元 9 完成事件	脉冲 (系统时钟域)
106	ADC0 转换单元 10 完成事件	脉冲 (系统时钟域)
107	ADC0 转换单元 11 完成事件	脉冲 (系统时钟域)
108	ADC0 转换单元 12 完成事件	脉冲 (系统时钟域)
109	ADC0 转换单元 13 完成事件	脉冲 (系统时钟域)
110	ADC0 转换单元 14 完成事件	脉冲 (系统时钟域)
111	ADC0 转换单元 15 完成事件	脉冲 (系统时钟域)
112	ADC0 PPB0 比较值上限事件	脉冲 (系统时钟域)
113	ADC0 PPB0 比较值下限事件	脉冲 (系统时钟域)
114	ADC0 PPB0 结果 Zero-crossing 事件	脉冲 (系统时钟域)

输入源序号	事件源	输出事件类型
115	ADC0 PPB1 比较值上限事件	脉冲 (系统时钟域)
116	ADC0 PPB1 比较值下限事件	脉冲 (系统时钟域)
117	ADC0 PPB1 结果 Zero-crossing 事件	脉冲 (系统时钟域)
118	ADC0 PPB2 比较值上限事件	脉冲 (系统时钟域)
119	ADC0 PPB2 比较值下限事件	脉冲 (系统时钟域)
120	ADC0 PPB2 结果 Zero-crossing 事件	脉冲 (系统时钟域)
121	ADC0 PPB3 比较值上限事件	脉冲 (系统时钟域)
122	ADC0 PPB3 比较值下限事件	脉冲 (系统时钟域)
123	ADC0 PPB3 结果 Zero-crossing 事件	脉冲 (系统时钟域)
124	ADC0 任一转换单元完成事件	脉冲 (系统时钟域)
125	ADC0 转换单元触发溢出事件	脉冲 (系统时钟域)
127-126	保留	保留
128	ADC1 转换单元 0 完成事件	脉冲 (系统时钟域)
129	ADC1 转换单元 1 完成事件	脉冲 (系统时钟域)
130	ADC1 转换单元 2 完成事件	脉冲 (系统时钟域)
131	ADC1 转换单元 3 完成事件	脉冲 (系统时钟域)
132	ADC1 转换单元 4 完成事件	脉冲 (系统时钟域)
133	ADC1 转换单元 5 完成事件	脉冲 (系统时钟域)
134	ADC1 转换单元 6 完成事件	脉冲 (系统时钟域)
135	ADC1 转换单元 7 完成事件	脉冲 (系统时钟域)
136	ADC1 转换单元 8 完成事件	脉冲 (系统时钟域)
137	ADC1 转换单元 9 完成事件	脉冲 (系统时钟域)
138	ADC1 转换单元 10 完成事件	脉冲 (系统时钟域)
139	ADC1 转换单元 11 完成事件	脉冲 (系统时钟域)
140	ADC1 转换单元 12 完成事件	脉冲 (系统时钟域)
141	ADC1 转换单元 13 完成事件	脉冲 (系统时钟域)
142	ADC1 转换单元 14 完成事件	脉冲 (系统时钟域)
143	ADC1 转换单元 15 完成事件	脉冲 (系统时钟域)
144	ADC1 PPB0 比较值上限事件	脉冲 (系统时钟域)
145	ADC1 PPB0 比较值下限事件	脉冲 (系统时钟域)
146	ADC1 PPB0 结果 Zero-crossing 事件	脉冲 (系统时钟域)
147	ADC1 PPB1 比较值上限事件	脉冲 (系统时钟域)

输入源序号	事件源	输出事件类型
148	ADC1 PPB1 比较值下限事件	脉冲 (系统时钟域)
149	ADC1 PPB1 结果 Zero-crossing 事件	脉冲 (系统时钟域)
150	ADC1 PPB2 比较值上限事件	脉冲 (系统时钟域)
151	ADC1 PPB2 比较值下限事件	脉冲 (系统时钟域)
152	ADC1 PPB2 结果 Zero-crossing 事件	脉冲 (系统时钟域)
153	ADC1 PPB3 比较值上限事件	脉冲 (系统时钟域)
154	ADC1 PPB3 比较值下限事件	脉冲 (系统时钟域)
155	ADC1 PPB3 结果 Zero-crossing 事件	脉冲 (系统时钟域)
156	ADC1 任一转换单元完成事件	脉冲 (系统时钟域)
157	ADC1 转换单元触发溢出事件	脉冲 (系统时钟域)
159-158	保留	保留
239-160	保留	保留
240	IO 输入选择源 0	异步信号
241	IO 输入选择源 1	异步信号
242	IO 输入选择源 2	异步信号
243	IO 输入选择源 3	异步信号
244	IO 输入选择源 4	异步信号
245	IO 输入选择源 5	异步信号
246	IO 输入选择源 6	异步信号
247	IO 输入选择源 7	异步信号
248	IO 输入选择源 8	异步信号
249	IO 输入选择源 9	异步信号
250	IO 输入选择源 10	异步信号
251	IO 输入选择源 11	异步信号
252	IO 输入选择源 12	异步信号
253	IO 输入选择源 13	异步信号
254	IO 输入选择源 14	异步信号
255	IO 输入选择源 15	异步信号

24.4.9 通道事件输出从设备

片上其他外设可作为从设备接收 EBUS 通道事件作为硬件触发源或其他用途。

表 392 可接收通道事件的从设备

外设	电平/脉冲	EBUS 模式配置
ADC	正常模式：脉冲（系统时钟域）	同步模式
GPIO	脉冲（系统时钟域）	同步模式
	电平	异步模式
TIMER	电平（系统时钟域）	异步模式
BTM	电平（BTM 时钟域）	异步模式
DMA	脉冲（系统时钟域）	同步模式

24.4.10 EBUS 中断号

表 393 EBUS 中断号

输入源序号	中断号
240	24
241	25
242	26
243	27
244	28
245	29
246	30
247	31
248	42
249	43
250	44
251	45
252	46
253	47
254	48
255	49

24.5 寄存器

EBUS 寄存器的基地址为 0x4001_0800, 下表为 EBUS 的相关寄存器描述。

表 394 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	EBUS 外设 ID 寄存器	32	ro	0x05070100
0x04	EBUS 软件事件触发寄存器	32	r/w	0x00000000
0x08	EBUS 事件触发 CPU 设置寄存器	32	r/w	0x00000000
0x0C	EBUSIO 边沿检测设置寄存器	32	r/w	0x00000000
0x10	EBUS 检测中断状态寄存器	32	ro	0x00000000
0x14	EBUS 检测中断状态清除寄存器	32	wo	0x00000000
0x18	EBUSI IO 滤波使能寄存器	32	r/w	0x00000000
0x1C	EBUS IO 源选择寄存器 1	32	r/w	0x00000000
0x20	EBUS IO 源选择寄存器 2	32	r/w	0x00000000
0x40	EBUS 通道 0 设置寄存器	32	r/w	0x00000000
0x44	EBUS 通道 1 设置寄存器	32	r/w	0x00000000
0x48	EBUS 通道 2 设置寄存器	32	r/w	0x00000000
0x4C	EBUS 通道 3 设置寄存器	32	r/w	0x00000000
0x50	EBUS 通道 4 设置寄存器	32	r/w	0x00000000
0x54	EBUS 通道 5 设置寄存器	32	r/w	0x00000000
0x58	EBUS 通道 6 设置寄存器	32	r/w	0x00000000
0x5C	EBUS 通道 7 设置寄存器	32	r/w	0x00000000
0x60	EBUS 通道 8 设置寄存器	32	r/w	0x00000000
0x64	EBUS 通道 9 设置寄存器	32	r/w	0x00000000
0x68	EBUS 通道 10 设置寄存器	32	r/w	0x00000000
0x6C	EBUS 通道 11 设置寄存器	32	r/w	0x00000000
0x70	EBUS 通道 12 设置寄存器	32	r/w	0x00000000
0x74	EBUS 通道 13 设置寄存器	32	r/w	0x00000000
0x78	EBUS 通道 14 设置寄存器	32	r/w	0x00000000
0x7C	EBUS 通道 15 设置寄存器	32	r/w	0x00000000

24.5.1 EBUS 外设 ID 寄存器 (EBUS_ID)

偏移地址: 0x0000

表 395 EBUS 外设 ID 寄存器

位	访问	描述
31:0	ro	EBUS 外设 ID 版本号: 0x05070100 复位值: 0x0

24.5.2 EBUS 软件事件触发寄存器 (EBUS_SOFT_EVT)

偏移地址: 0x0004

表 396 EBUS 软件事件触发寄存器

位	访问	描述
31	r/w	CH15_SW_LVL: 通道 15 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
30	r/w	CH14_SW_LVL: 通道 14 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
29	r/w	CH13_SW_LVL: 通道 13 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
28	r/w	CH12_SW_LVL: 通道 12 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
27	r/w	CH11_SW_LVL: 通道 11 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
26	r/w	CH10_SW_LVL: 通道 10 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
25	r/w	CH9_SW_LVL: 通道 9 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0

位	访问	描述
24	r/w	CH8_SW_LVL: 通道 8 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
23	r/w	CH7_SW_LVL: 通道 7 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
22	r/w	CH6_SW_LVL: 通道 6 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
21	r/w	CH5_SW_LVL: 通道 5 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
20	r/w	CH4_SW_LVL: 通道 4 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
19	r/w	CH3_SW_LVL: 通道 3 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
18	r/w	CH2_SW_LVL: 通道 2 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
17	r/w	CH1_SW_LVL: 通道 1 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
16	r/w	CH0_SW_LVL: 通道 0 软件触发电平事件 0: 不触发事件 1: 软件触发一个电平事件 复位值: 0x0
15	r/w	CH15_SW_PUL: 通道 15 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0

位	访问	描述
14	r/w	CH14_SW_PUL: 通道 14 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0
13	r/w	CH13_SW_PUL: 通道 13 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0
12	r/w	CH12_SW_PUL: 通道 12 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0
11	r/w	CH11_SW_PUL: 通道 11 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0
10	r/w	CH10_SW_PUL: 通道 10 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0
9	r/w	CH9_SW_PUL: 通道 9 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0
8	r/w	CH8_SW_PUL: 通道 8 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0
7	r/w	CH7_SW_PUL: 通道 7 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0
6	r/w	CH6_SW_PUL: 通道 6 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0
5	r/w	CH5_SW_PUL: 通道 5 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0

位	访问	描述
4	r/w	CH4_SW_PUL: 通道 4 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0
3	r/w	CH3_SW_PUL: 通道 3 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0
2	r/w	CH2_SW_PUL: 通道 2 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0
1	r/w	CH1_SW_PUL: 通道 1 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0
0	r/w	CH0_SW_PUL: 通道 0 软件触发脉冲事件 0: 不触发事件 1: 软件触发一个事件脉冲, 硬件自动清 0 复位值: 0x0

24.5.3 EBUS 事件触发 CPU 设置寄存器 (EBUS_EVT_CPU)

偏移地址: 0x0008

表 397 EBUS 事件触发 CPU 设置寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7:4	r/w	EVT_CPU_CHNSEL: EBUS 事件触发 CPU 通道选择 0000: 通道 0 事件输出触发 CPU 0001: 通道 1 事件输出触发 CPU 0010: 通道 2 事件输出触发 CPU ... 1111: 通道 15 事件输出触发 CPU 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	EVT_CPU_EN: EBUS 事件触发 CPU 使能 0: 关闭 1: 使能 复位值: 0x0

24.5.4 EBUS IO 边沿检测设置寄存器 (EBUS_IO_DET)

偏移地址: 0x000C

表 398 EBUS IO 边沿检测设置寄存器

位	访问	描述
31:30	r/w	IO15_EDGE_SEL: IO15 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0
29:28	r/w	IO14_EDGE_SEL: IO14 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0
27:26	r/w	IO13_EDGE_SEL: IO13 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0
25:24	r/w	IO12_EDGE_SEL: IO12 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0
23:22	r/w	IO11_EDGE_SEL: IO11 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0
21:20	r/w	IO10_EDGE_SEL: IO10 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0

位	访问	描述
19:18	r/w	IO9_EDGE_SEL: IO9 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0
17:16	r/w	IO8_EDGE_SEL: IO8 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0
15:14	r/w	IO7_EDGE_SEL: IO7 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0
13:12	r/w	IO6_EDGE_SEL: IO6 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0
11:10	r/w	IO5_EDGE_SEL: IO5 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0
9:8	r/w	IO4_EDGE_SEL: IO4 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0
7:6	r/w	IO3_EDGE_SEL: IO3 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测

位	访问	描述
		复位值: 0x0
5:4	r/w	IO2_EDGE_SEL: IO2 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0
3:2	r/w	IO1_EDGE_SEL: IO1 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0
1:0	r/w	IO0_EDGE_SEL: IO0 边沿检测选择配置 00: 不使能检测功能 01: IO 上升沿检测 10: IO 下降沿检测 11: IO 上升或者下降沿检测 复位值: 0x0

24.5.5 EBUS IO 检测中断状态寄存器 (EBUS_IO_STA)

偏移地址: 0x0010

表 399 EBUS IO 检测中断状态寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15	ro	IO15_INT_STA: IO15 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0
14	ro	IO14_INT_STA: IO14 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0
13	ro	IO13_INT_STA: IO13 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0
12	ro	IO12_INT_STA: IO12 检测中断状态 0: 未检测到目标边沿

位	访问	描述
		1: 已检测到目标边沿 复位值: 0x0
11	ro	IO11_INT_STA: IO11 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0
10	ro	IO10_INT_STA: IO10 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0
9	ro	IO9_INT_STA: IO9 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0
8	ro	IO8_INT_STA: IO8 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0
7	ro	IO7_INT_STA: IO7 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0
6	ro	IO6_INT_STA: IO6 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0
5	ro	IO5_INT_STA: IO5 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0
4	ro	IO4_INT_STA: IO4 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0
3	ro	IO3_INT_STA: IO3 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0

位	访问	描述
2	ro	IO2_INT_STA: IO2 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0
1	ro	IO1_INT_STA: IO1 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0
0	ro	IO0_INT_STA: IO0 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 复位值: 0x0

24.5.6 EBUS IO 检测中断状态清除寄存器 (IO_INT_STACLR)

偏移地址: 0x0014

表 400 EBUS IO 检测中断状态清除寄存器

位	访问	描述
31:16	Res	Reserved 复位值: 0x0
15	wo	IO15_INT_STACLR: IO15 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0
14	wo	IO14_INT_STACLR: IO14 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0
13	wo	IO13_INT_STACLR: IO13 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0
12	wo	IO12_INT_STACLR: IO12 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0
11	wo	IO11_INT_STACLR: IO11 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0

位	访问	描述
10	wo	IO10_INT_STACLR: IO10 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0
9	wo	IO9_INT_STACLR: IO9 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0
8	wo	IO8_INT_STACLR: IO8 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0
7	wo	IO7_INT_STACLR: IO7 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0
6	wo	IO6_INT_STACLR: IO6 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0
5	wo	IO5_INT_STACLR: IO5 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0
4	wo	IO4_INT_STACLR: IO4 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0
3	wo	IO3_INT_STACLR: IO3 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0
2	wo	IO2_INT_STACLR: IO2 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0
1	wo	IO1_INT_STACLR: IO1 检测中断状态清除 0: 不清除 1: 清除中断状态 复位值: 0x0
0	wo	IO0_INT_STACLR: IO0 检测中断状态清除

位	访问	描述
		0: 不清除 1: 清除中断状态 复位值: 0x0

24.5.7 EBUS IO 源选择寄存器 1 (IO_SRC_SEL1)

偏移地址: 0x001C

表 401 EBUS IO 源选择寄存器 1

位	访问	描述
31:28	r/w	IO7_SRC_SEL: IO7 源选择 0000: PA7 0001: PB7 0010: PC7 0011: PD7 其他值保留 复位值: 0x0
27:24	r/w	IO6_SRC_SEL: IO6 源选择 0000: PA6 0001: PB6 0010: PC6 0011: PD6 其他值保留 复位值: 0x0
23:20	r/w	IO5_SRC_SEL: IO5 源选择 0000: PA5 0001: PB5 0010: PC5 0011: PD5 其他值保留 复位值: 0x0
19:16	r/w	IO4_SRC_SEL: IO4 源选择 0000: PA4 0001: PB4 0010: PC4 0011: PD4 其他值保留 复位值: 0x0
15:12	r/w	IO3_SRC_SEL: IO3 源选择 0000: PA3

位	访问	描述
		0001: PB3 0010: PC3 0011: PD3 其他值保留 复位值: 0x0
11:8	r/w	IO2_SRC_SEL: IO2 源选择 0000: PA2 0001: PB2 0010: PC2 0011: PD2 其他值保留 复位值: 0x0
7:4	r/w	IO1_SRC_SEL: IO1 源选择 0000: PA1 0001: PB1 0010: PC1 0011: PD1 其他值保留 复位值: 0x0
3:0	r/w	IO0_SRC_SEL: IO0 源选择 0000: PA0 0001: PB0 0010: PC0 0011: PD0 其他值保留 复位值: 0x0

24.5.8 EBUS IO 源选择寄存器 2 (IO_SRC_SEL2)

偏移地址: 0x0020

表 402 EBUS IO 源选择寄存器 2

位	访问	描述
31:28	r/w	IO15_SRC_SEL: IO15 源选择 0000: PA15 0001: PB15 0010: PC15 0011: PD15 其他值保留 复位值: 0x0
27:24	r/w	IO14_SRC_SEL: IO14 源选择 0000: PA14

位	访问	描述
		0001: PB14 0010: PC14 0011: PD14 其他值保留 复位值: 0x0
23:20	r/w	IO13_SRC_SEL: IO13 源选择 0000: PA13 0001: PB13 0010: PC13 0011: PD13 其他值保留 复位值: 0x0
19:16	r/w	IO12_SRC_SEL: IO12 源选择 0000: PA12 0001: PB12 0010: PC12 0011: PD12 其他值保留 复位值: 0x0
15:12	r/w	IO11_SRC_SEL: IO11 源选择 0000: PA11 0001: PB11 0010: PC11 0011: PD11 其他值保留 复位值: 0x0
11:8	r/w	IO10_SRC_SEL: IO10 源选择 0000: PA10 0001: PB10 0010: PC10 0011: PD10 其他值保留 复位值: 0x0
7:4	r/w	IO9_SRC_SEL: IO9 源选择 0000: PA9 0001: PB9 0010: PC9 0011: PD9 其他值保留 复位值: 0x0
3:0	r/w	IO8_SRC_SEL: IO8 源选择

位	访问	描述
		0000: PA8 0001: PB8 0010: PC8 0011: PD8 其他值保留 复位值: 0x0

24.5.9 EBUS 通道 0 设置寄存器 (EBUS_CHN0_CFG)

偏移地址: 0x0040

表 403 EBUS 通道 0 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 0 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 0 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 0 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 0 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 0 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 0 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿

位	访问	描述
		11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 0 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

Kiwi Instruments Corp. Confidential

24.5.10 EBUS 通道 1 设置寄存器 (EBUS_CHN1_CFG)

偏移地址: 0x0044

表 404 EBUS 通道 1 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 1 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 1 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 1 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 1 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 1 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 1 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 1 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

24.5.11 EBUS 通道 2 设置寄存器 (EBUS_CHN2_CFG)

偏移地址: 0x0048

表 405 EBUS 通道 2 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 2 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 2 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 2 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 2 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 2 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 2 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 2 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

24.5.12 EBUS 通道 3 设置寄存器 (EBUS_CHN3_CFG)

偏移地址: 0x004C

表 406 EBUS 通道 3 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 3 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 3 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 3 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 3 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 3 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 3 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 3 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

24.5.13 EBUS 通道 4 设置寄存器 (EBUS_CHN4_CFG)

偏移地址: 0x0050

表 407 EBUS 通道 4 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 4 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 4 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 4 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 4 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 4 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 4 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 4 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

24.5.14 EBUS 通道 5 设置寄存器 (EBUS_CHN5_CFG)

偏移地址: 0x0054

表 408 EBUS 通道 5 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 5 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 5 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 5 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 5 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 5 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 5 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 5 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

24.5.15 EBUS 通道 6 设置寄存器 (EBUS_CHN6_CFG)

偏移地址: 0x0058

表 409 EBUS 通道 6 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 6 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 6 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 6 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 6 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 6 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 6 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 6 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

24.5.16 EBUS 通道 7 设置寄存器 (EBUS_CHN7_CFG)

偏移地址: 0x005C

表 410 EBUS 通道 7 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 7 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 7 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 7 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 7 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 7 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 7 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 7 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

24.5.17 EBUS 通道 8 设置寄存器 (EBUS_CHN8_CFG)

偏移地址: 0x005C

表 411 EBUS 通道 8 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 8 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 8 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 8 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 8 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 8 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 8 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 8 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

24.5.18 EBUS 通道 9 设置寄存器 (EBUS_CHN9_CFG)

偏移地址: 0x005C

表 412 EBUS 通道 9 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 9 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 9 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 9 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 9 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 9 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 9 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 9 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

24.5.19 EBUS 通道 10 设置寄存器 (EBUS_CHN10_CFG)

偏移地址: 0x005C

表 413 EBUS 通道 10 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 10 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 10 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 10 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 10 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 10 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 10 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 10 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

24.5.20 EBUS 通道 11 设置寄存器 (EBUS_CHN11_CFG)

偏移地址: 0x005C

表 414 EBUS 通道 11 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 11 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 11 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 11 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 11 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 11 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 11 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 11 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

24.5.21 EBUS 通道 12 设置寄存器 (EBUS_CHN12_CFG)

偏移地址: 0x005C

表 415 EBUS 通道 12 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 12 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 12 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 12 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 12 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 12 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 12 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 12 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

24.5.22 EBUS 通道 13 设置寄存器 (EBUS_CHN13_CFG)

偏移地址: 0x005C

表 416 EBUS 通道 13 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 13 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 13 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 13 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 13 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 13 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 13 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 13 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

24.5.23 EBUS 通道 14 设置寄存器 (EBUS_CHN14_CFG)

偏移地址: 0x005C

表 417 EBUS 通道 14 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 14 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 14 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 14 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 14 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 14 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 14 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 14 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

24.5.24 EBUS 通道 15 设置寄存器 (EBUS_CHN15_CFG)

偏移地址: 0x005C

表 418 EBUS 通道 15 设置寄存器

位	访问	描述
31:15	Res	Reserved 复位值: 0x0
14:7	r/w	EBUS_SRC_SEL: EBUS 通道 15 输入源选择 'h00: 选择输入源 0 'h01: 选择输入源 1 'h02: 选择输入源 2 ... 'hFF: 选择输入源 255 复位值: 0x0
6	r/w	EBUS_CHN_OE: EBUS 通道 15 输出到 PAD 使能 0: 关闭 1: 使能 复位值: 0x0
5	r/w	EBUS_INV_EN: EBUS 通道 15 输出反相使能 0: 关闭 1: 使能 复位值: 0x0
4	r/w	EBUS_OR_EN: EBUS 通道 15 输出与前一通道输出逻辑或使能 0: 关闭 1: 使能 复位值: 0x0
3	r/w	EBUS_AND_EN: EBUS 通道 15 输出与后一通道输出逻辑与使能 0: 关闭 1: 使能 复位值: 0x0
2:1	r/w	EBUS_DET_SEL: EBUS 通道 15 输入检测方式选择 00: 只同步通道输入源 01: 检测通道输入源的上升沿 10: 检测通道输入源的下降沿 11: 检测通道输入源的上升沿或下降沿 复位值: 0x0
0	r/w	EBUS_ASYNC_EN: EBUS 通道 15 输入异步使能 0: 关闭 1: 使能 复位值: 0x0

25 控制器局域网 (CAN)

25.1 概述

CAN 总线系统广泛应用于汽车工业及其它工业环境中，它给微处理器之间提供了可进行数据编码解码、信息管理、通信同步、发送及接收信息的 CAN 总线接口。

该 CAN 控制器支持 BOSCH CAN2.0A 和 CAN2.0B；可进行标准报文格式传输 (11 位标识符)和扩展报文格式传输 (29 位标识符)。

25.2 主要功能

- 支持 BOSCH CAN2.0A 和 CAN2.0B
- 支持 5Kbps~1Mbps 波特率传输
- 支持 11 位标识符和 29 位标识符
- 有 16*32bit 接收缓冲
- 支持信息过滤
- 支持监听模式
- 接收自身报文
- 自测模式
- 总线错误中断
- 保存位信息的仲裁丢失中断
- 读写错误计数
- 错误状态信息记录
- 可配置错误个数预警

25.3 模块框图

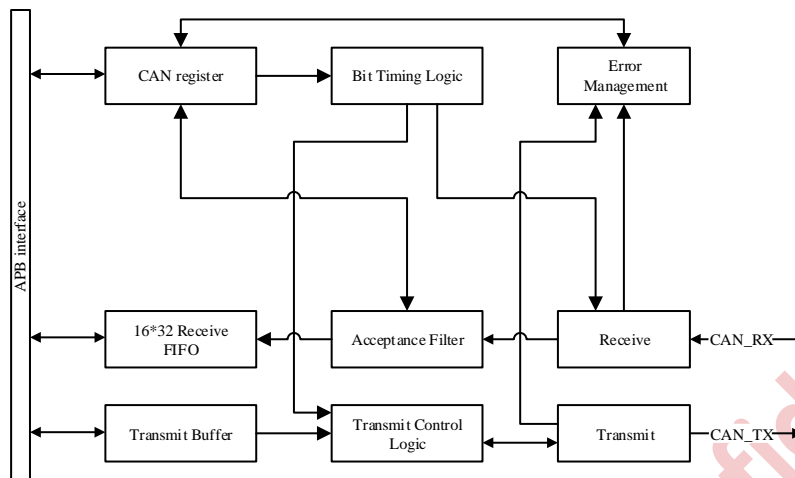


图 261 CAN 模块架构图

25.4 功能描述

25.4.1 CAN 简述

CPU 通过 APB 总线对 CAN 进行访问。接收到的通信报文在经过过滤处理后存储在传输缓冲区以方便 CPU 取出；CPU 可取出缓冲区存有的数据，接收缓冲器一次最多可容纳 4 笔扩展帧格式的报文。

配置位时序逻辑模块可更改通信波特率，通信波特率可实现 5Kbps~1Mbps 范围的调节。

CAN 控制器拥有 CAN_TX (发送)、CAN_RX (接收) 和 CAN_DBG (调试) 接口，通常 CAN_DBG 与 CAN_TX 相反，也可将 CAN_DBG 调整为发送时钟来方便调试。

25.4.2 操作模式

CAN 有初始化模式和标准模式两种主要的操作模式；初始化模式可以设置总线波特率参数和过滤器配置，更改 REC (接收错误计数)、TEC (发送错误计数) 和 ERR_WARN_THR (错误警告阈值)；当工作模式切换到初始化模式时将会执行一个硬件复位；在标准模式下可进行发送和接收信息。

CAN 也支持监听模式和自测模式；在监听模式下，仅可进行数据接收，无法实现数据发送，当数据接收成功后也不会回复响应信号。它还被强制为被动错误；在自测模式下，CAN 使用自动收发功能，无需其他 CAN 节点回复。

CAN 同时也提供时钟输出功能，可将发送时钟通过 CAN_DBG 引脚输出。

25.4.3 信息发送

发送的信息需要按照 SFF (标准帧格式) 或者 EFF (扩展帧格式) 写入发送缓冲器中，在将数据写入缓冲器之前请检查发送缓冲器未处于锁定状态 (CAN_ST[2])，否则将造成数据丢失；

在进行配置发送请求寄存器 (CAN_CTRL[0]) 或自发自收请求寄存器 (CAN_CTRL[8]) 时，CAN 会将发送缓冲器中的数据发出，当开始数据发送后传输状态位置起 (CAN_ST[0]) 并且发送请求寄存器自动清零；

发送序列通过 CAN_TX 端口输出，在标准模式下，反相信号在 CAN_DBG 端口输出。如果发生仲裁失败或传输错误，CAN 控制器将自动尝试再次发送信息；

如果传输未开始可以通过终止传输寄存器 (CAN_CTRL[1]) 取消信息发送，信息一旦开始，就不可能中止；

注：

1. 要查看信息是否已发送或中止，可等待发送缓冲器处于未锁定 (CAN_ST[2]) 或产生发送中断，之后可通过发送完成寄存器 (CAN_ST[1]) 查看发送状态；
2. 信息发送由于发送缓冲器状态变为未锁定而中止，当发送中断使能时也会产生发送中断；
3. 在发送请求的同时发送中止发送请求会导致当前消息单次发送，即使发生仲裁丢失或错误不会重新尝试发送。

25.4.4 信息接收

CAN 接收到的信息先通过接收过滤器，然后传递到接收缓冲器，过滤器只会将与预设匹配的信息通过。在接收到信息后，如果打开接收中断使能则会产生接收中断。

接收缓冲器拥有 16*32 比特的空间，最多可存储 4 笔扩展帧格式的信息；如果接收缓冲器没有足够空间接收信息，会发生接收缓冲溢出状态，并丢失接收到的信息，同时如果打开溢出中断使能也会产生接收缓冲溢出中断。

在每读出一笔信息后，需要对接收缓冲器进行释放动作，如果接收缓冲器仍存在未读信息并接收中断使能未关闭，会再一次产生接收中断。

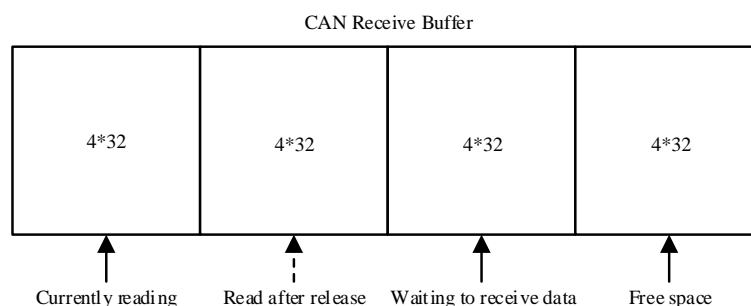


图 262 释放 FIFO 之前

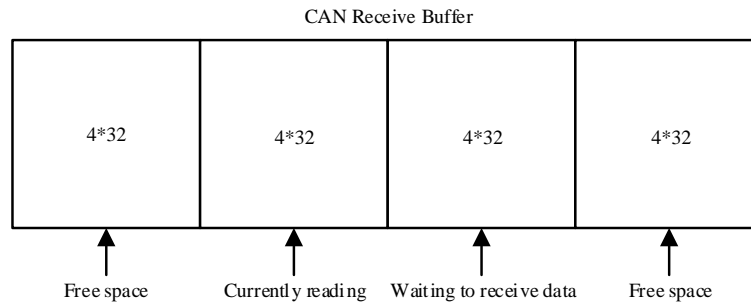


图 263 释放 FIFO 之后

CAN 控制器的一个功能是在将信息发送至其他节点 CAN 设备的同时接收信息，该功能可通过自接收发送请求实现。

25.4.5 接收滤波

在 CAN 总线网络中所有的节点都接收总线上传输的信息，为了忽略与本身无关的信息，可通过接收滤波器进行预过滤。只有标识符位与过滤器匹配的消息才会传输至接收缓冲器，可以配置两套接收过滤器。

25.4.6 波特率

CAN 总线上使用的波特率是通过设定周期与 TQ (Time Quanta) 之间的关系来指定；TQ 的长度与 CAN 工作时钟频率有关；

位周期允许定义为 3 到 25 个 TQ，然而实际使用要遵循 BOSCH 标准，该标准定义位周期长度在 8 到 25 个 TQ 之间。

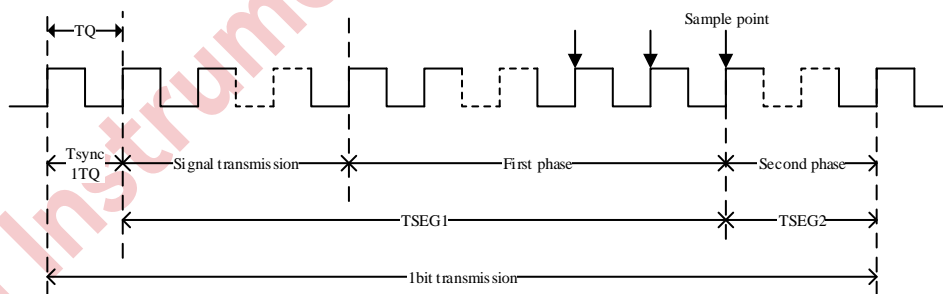


图 264 波特率

可以通过设置相位缓冲段 1 (TSEG1) 和相位缓冲段 2 (TSEG2) 的长度来更改位周期与位采样点，位采样次数可选择 1 次或 3 次。

25.4.7 总线仲裁

CAN 总线的控制按照 CAN2.0 的规则管理；任何时候，控制总线的 CAN 节点都是标识符最低的节点，仲裁失败的节点必须退出，并且知道 CAN 总线空闲之前不再尝试控制 CAN 总线；总线仲裁失败而且仲裁失败中断使能打开时会产生仲裁失败中断，并将仲裁失败的位置记录下来；

仲裁失败寄存器中记录的详细信息直到该寄存器被读取后才会被清除，因此在从寄存器中读取之前记录的详细信息之前，无法记录有关仲裁失败的更多信息。

25.4.8 错误处理

CAN 总线传输错误按照 CAN2.0 的规则管理；CAN 包含两个错误计数器，一个用于记录接收错误 (REC)，一个记录发送错误 (TEC)，当发生错误时，错误计数根据 CAN2.0 协议递增；错误信息会存储在错误信息寄存器中，错误信息直到该寄存器被读取之前将会一直保留；可设置错误报警阈值并打开错误报警中断使能，在错误计数器达到报警阈值后会产生报警中断。

如果发送或接收错误计数器超过 127，CAN 将进入被动错误状态并在错误报警中断使能情况下发生被动错误中断。如果超过 255，将会关闭总线并进入到初始化模式，同时在错误报警中断使能情况下发生错误报警中断。

CAN 在 CPU 配置工作模式寄存器之前一直保持在初始化模式。另外在回到初始化模式后，CAN 将等待 128 个总线空闲时序，之后再变回总线开启。

25.4.9 睡眠模式

当没有通信需求时可以将 CAN 配置为睡眠模式来降低功耗；

以下任何一种情况均可将 CAN 唤醒：

1. 关闭睡眠模式使能
2. CAN 总线上有通信动作

在总线通信唤醒后，会产生一个唤醒中断（唤醒中断使能打开）

注：

1. 初始化模式下无法切换至睡眠模式；
2. 如果 CAN 是被总线通信唤醒，直到检测到总线上 11 个隐形位之前它将不会收到任何信息。

25.4.10 CAN 中断号

表 419 CAN 中断号

名称	中断号
CAN_INT	26

25.5 寄存器

CAN 寄存器的基地址为 0x4001_2000,下表为 CAN 的相关寄存器描述。

表 420 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x08	CAN 控制寄存器	32	r/w	0x00000000
0x10	CAN 配置寄存器	32	r/w	0x00000000
0x14	CAN 中断使能寄存器	32	r/w	0x00000000
0x18	CAN 总线时序寄存器	32	r/w	0x00000000
0x20	CAN 验收标识符 0	32	r/w	0x00000000
0x24	CAN 验收屏蔽 0	32	r/w	0x00000000
0x28	CAN 验收标识符 1	32	r/w	0x00000000
0x2C	CAN 验收屏蔽 1	32	r/w	0x00000000
0x30	CAN 错误预警寄存器	32	r/w	0x00000060
0x38	CAN 输出控制寄存器	32	r/w	0x00000002
0x40	CAN 状态寄存器	32	ro	0x00000103
0x44	CAN 中断寄存器	32	r/w	0x00000000
0x48	CAN 接收缓冲器状态	32	r/w	0x00000000
0x50	CAN 发送帧信息寄存器	32	r/w	0x00000000
0x54	CAN 发送标识符	32	r/w	0x00000000
0x58	CAN 发送数据寄存器 0	32	r/w	0x00000000
0x5C	CAN 发送数据寄存器 1	32	r/w	0x00000000
0x60	CAN 接收帧信息寄存器	32	ro	0x00000000
0x64	CAN 接收标识符	32	ro	0x00000000
0x68	CAN 接收数据寄存器 0	32	ro	0x00000000
0x6C	CAN 接收数据寄存器 1	32	ro	0x00000000
0x70	CAN 仲裁丢失位	32	ro	0x00000000
0x74	CAN 错误信息寄存器	32	ro	0x00000000

偏移地址	寄存器列表	位宽	访问	复位值
0x78	CAN 错误计数寄存器	32	r/w	0x00000000
0x80~0xBC	CAN 接收缓冲器	32	r/w	0x00000000

25.5.1 CAN 控制寄存器 (CAN_CTRL)

偏移地址: 0x0008

表 421 CAN 控制寄存器

位	访问	描述
31:18	Res	Reserved 复位值: 0x0
17	r/w	CLR_DATA_OV: 清除数据溢出 0: 无请求 1: 请求清除数据溢出状态位发出的数据溢出条件 复位值: 0x0
16	r/w	RLSE_RX_BUF: 释放接收缓冲 0: 无请求 1: 请求释放接收缓冲 复位值: 0x0
15:9	Res	Reserved 复位值: 0x0
8	r/w	SELF_RX_REQ: 自接收请求 0: 无请求 1: 请求发送信息并同时接收 复位值: 0x0
7:2	Res	Reserved 复位值: 0x0
1	r/w	ABORT_TX: 中止发送 0: 无影响 1: 中止重新发送 复位值: 0x0
0	r/w	TX_REQ: 发送请求 0: 无信息发送请求 1: 信息发送请求 复位值: 0x0

注意:

1. 同时设置 CAN_CTRL[0] 和 CAN_CTRL[1] 会导致信息单次发送, 在发生错误或仲裁失败后不会重新发送;
2. 同时将 CAN_CTRL[0] 和 CAN_CTRL[8] 配置 1, 则 CAN_CTRL[8] 位会被忽略。

25.5.2 CAN 配置寄存器 (CAN_CFG)

偏移地址: 0x0010

表 422 CAN 配置寄存器

位	访问	描述
31:9	Res	Reserved 复位值: 0x0
8	r/w	SLEEP_MODE_EN: 睡眠模式使能 0: 睡眠模式唤醒 1: 进入睡眠模式 注: 在没有中断挂起并且总线无信息传输可进入睡眠模式; 如果有中断挂起或总线信息传输将执行唤醒; 复位值: 0x0
7:2	Res	Reserved 复位值: 0x0
1:0	r/w	WORK_MD: 工作模式 2'B00: 初始化模式 注: 中止当前正在发送或接收的任何消息并进入复位模式 2'B01: 标准模式 2'B10: 监听模式 注: 在这种模式下及时成功接收也不会发送应答 2'B11: 自接收模式 注: 可在总线没有任何活跃节点的情况下进行测试, 及时没有应答也能发送成功 复位值: 0x0

25.5.3 CAN 中断使能寄存器 (CAN_IE)

偏移地址: 0x0014

表 423 CAN 中断使能寄存器

位	访问	描述
31:25	Res	Reserved 复位值: 0x0
24	r/w	WAKEUP_INT_EN: 唤醒中断使能 0: 关闭唤醒中断使能 1: 打开唤醒中断使能 复位值: 0x0
23:20	Res	Reserved 复位值: 0x0
19	r/w	ERR_ERR_INT_EN: 总线错误中断使能 0: 关闭总线错误中断使能 1: 打开总线错误中断使能 复位值: 0x0

位	访问	描述
18	r/w	ERR_PASSIVE_INT_EN: 被动错误中断使能 0: 关闭被动错误中断使能 1: 打开被动错误中断使能 当CAN由主动错误变成被动错误时发生; 复位值: 0x0
17	r/w	ERR_WARN_INT_EN: 错误报警中断使能 0: 关闭错误报警中断使能 1: 打开错误报警中断使能 复位值: 0x0
16	r/w	ARBIT_LOST_INT_EN: 仲裁丢失中断使能 0: 关闭仲裁丢失中断使能 1: 打开仲裁丢失中断使能 复位值: 0x0
15:11	Res	Reserved 复位值: 0x0
10	r/w	DATA_OV_INT_EN: 溢出中断使能 0: 关闭溢出中断使能 1: 打开溢出中断使能 复位值: 0x0
9	r/w	TX_INT_EN: 发送中断使能 0: 关闭发送中断使能 1: 打开发送中断使能 复位值: 0x0
8	r/w	RX_INT_EN: 接收中断使能 0: 关闭接收中断使能 1: 打开接收中断使能 复位值: 0x0
7:1	Res	Reserved 复位值: 0x0
0	r/w	CAN_INT_EN: CAN中断使能 0: 关闭CAN中断使能 1: 打开CAN中断使能 复位值: 0x0

25.5.4 CAN 总线时序寄存器 (CAN_BUS_TIM)

偏移地址: 0x0018

表 424 CAN 总线时序寄存器

位	访问	描述
31:27	Res	Reserved 复位值: 0x0

位	访问	描述
26	r/w	SAM: 采样次数 0: 采样 1 次 1: 采样 3 次 复位值: 0x0
25:24	r/w	SJW: 同步跳转宽度 2'B00: 1*TQ 2'B01: 2*TQ 2'B10: 3*TQ 2'B11: 4*TQ 复位值: 0x0
23	Res	Reserved 复位值: 0x0
22:20	r/w	TSEG2: 相位缓冲段 2 3'B000: 1*TQ 3'B001: 2*TQ 3'B010: 3*TQ ... 3'B110: 7*TQ 3'B111: 8*TQ 复位值: 0x0
19:16	r/w	TSEG1: 相位缓冲段 1 4'B0000: 1*TQ 4'B0001: 2*TQ 4'B0010: 3*TQ ... 4'B1110: 15*TQ 4'B1111: 16*TQ 复位值: 0x0
15:12	Res	Reserved 复位值: 0x0
11:0	r/w	BRP: 波特率预分频 注: 定义TQ与CAN时钟分频关系, 遵循以下公式: $TQ = 2 * Tclk * (BRP[11:0] + 1)$ 复位值: 0x0

注: 该寄存器仅可在初始化模式下配置。

$$tsyncseg = 1 * TQ$$

$$ttseg1 = TQ * (TSEG1[3:0] + 1)$$

$$ttseg2 = TQ * (TSEG2[2:0] + 1)$$

25.5.5 CAN 验收标识符 0 (CAN_ACPT_CODE0)

偏移地址: 0x0020

表 425 CAN 验收标识符 0

位	访问	描述
31:0	r/w	ACPT_CODE0: 验收标识符 0 注: 用于记录需要接收的标识符, 配置方式取决于数据是标准帧格式还是扩展帧格式 复位值: 0x0

注: 该寄存器仅可在初始化模式下配置。

25.5.6 CAN 验收屏蔽 0 (CAN_ACPT_MASK0)

偏移地址: 0x0024

表 426 CAN 验收屏蔽 0

位	访问	描述
31:0	r/w	ACPT_MASK0: 验收屏蔽 0 注: 对应比特设置为 0 表示接收标识符需要与验收标识符对应, 如不对应则会被过滤; 对应比特设置为 1 则表示该比特不关心; 复位值: 0x0

注: 该寄存器仅可在初始化模式下配置。

25.5.7 CAN 验收标识符 1 (CAN_ACPT_CODE1)

偏移地址: 0x0028

表 427 CAN 验收标识符 1

位	访问	描述
31:0	r/w	ACPT_CODE1: 验收标识符 1 注: 用于记录需要接收的标识符, 配置方式取决于数据是标准帧格式还是扩展帧格式。 复位值: 0x0

注: 该寄存器仅可在初始化模式下配置。

25.5.8 CAN 验收屏蔽 1 (CAN_ACPT_MASK1)

偏移地址：0x002C

表 428 CAN 验收屏蔽 1

位	访问	描述
31:0	r/w	ACPT_MASK1: 验收屏蔽 1 注：对应比特设置为 0 表示接收标识符需要与验收标识符对应，如不对应则会被过滤；对应比特设置为 1 则表示该比特不关心。 复位值：0x0

注：该寄存器仅可在初始化模式下配置。

25.5.9 CAN 错误预警寄存器 (CAN_ERR_WARN_THR)

偏移地址：0x0030

表 429 CAN 错误预警寄存器

位	访问	描述
31:8	Res	Reserved 复位值：0x0
7:0	r/w	ERR_WARN_THR: 错误报警阈值 注：用于定义发生错误报警中断的错误阈值； 复位值：0x60

注：该寄存器仅可在初始化模式下配置。

25.5.10 CAN 输出控制寄存器 (CAN_OUT_CTRL)

偏移地址：0x0038

表 430 CAN 输出控制寄存器

位	访问	描述
31:1	Res	Reserved 复位值：0x0
0	r/w	OC_MODE: 输出控制模式 2'B0x: Reserved 2'B10: 正常输出模式 2'B11: 时钟输出模式 注：在正常输出模式，CAN_TX发送信息，CAN_DBG 的输出与之相反；在时钟输出模式，CAN_TX 发送信息，CAN_DBG 输出发送时钟。 复位值：0x2

注：该寄存器仅可在初始化模式下配置。

25.5.11 CAN 状态寄存器 (CAN_ST)

偏移地址: 0x0040

表 431 CAN 状态寄存器

位	访问	描述
31:18	Res	Reserved 复位值: 0x0
17	ro	BUS_OFF: 总线关闭 0: 当前 CAN 可参与总线通信 1: 当前 CAN 处于总线关闭 复位值: 0x0
16	ro	ERR_CNT_ST: 错误计数器状态 0: 接收错误计数与发送错误计数低于错误预警值 1: 接收错误计数或发送错误计数高于或等于错误预警值 复位值: 0x0
15:11	Res	Reserved 复位值: 0x0
10	ro	RX_DATA_OV: 接收缓冲器溢出 0: 接收缓冲器有空间存储信息 1: 接收缓冲器无空间存储信息 复位值: 0x0
9	ro	RX_FIFO_NOT_EMPTY: 接收缓冲器非空 0: 当前接收缓冲器无有效数据 1: 当前接收缓冲器有有效数据 复位值: 0x0
8	ro	RX_BUSY: 接收忙 0: 当前无信息接收 1: 当前正在接收信息 复位值: 0x1
7:3	Res	Reserved 复位值: 0x0
2	ro	TX_BUF_LOCK: 发送缓冲器锁定 0: 发送缓冲器未锁定 1: 发送缓冲器已锁定 复位值: 0x0
1	ro	TX_DONE: 发送完成 0: 当前发送未完成 1: 当前发送已完成 复位值: 0x1

位	访问	描述
0	ro	TX_BUSY: 发送忙 0: 当前无信息发送 1: 当前正在发送信息 复位值: 0x1

注:

- 如果 TX_BUSY 和 RX_BUSY 同时为 0 则表示总线处于空闲, 如果都是 1 则等待再次处于空闲。在硬件复位后, 一旦检测到总线出现空闲序列 (11 个连续隐形位) 就进入空闲状态。在总线关闭后, 进入空闲状态前必须接收 128 个总线空闲序列;
- 溢出状态仅在接收到整个消息后显示, 如果消息没有完成, 则不会显示溢出情况。

25.5.12 CAN 中断寄存器 (CAN_INT)

偏移地址: 0x0044

表 432 CAN 中断寄存器

位	访问	描述
31:25	Res	Reserved 复位值: 0x0
24	r/w	WAKEUP_INT: 唤醒中断 0: 未发生唤醒中断请求 1: 发生唤醒中断请求 复位值: 0x0
23:20	Res	Reserved 复位值: 0x0
19	r/w	BUS_ERR_INT: 总线错误中断 0: 未发生总线错误中断请求 1: 发生总线错误中断请求 复位值: 0x0
18	r/w	ERR_PASSIVE_INT: 被动错误中断 0: 未发生被动错误中断请求 1: 发生被动错误中断请求 复位值: 0x0
17	r/w	ERR_WARN_INT: 错误报警中断 0: 未发生错误报警中断请求 1: 发生错误报警中断请求 复位值: 0x0
16	r/w	ARBIT_LOST_INT: 仲裁丢失中断 0: 未发生仲裁丢失中断请求 1: 发生仲裁丢失中断请求 复位值: 0x0

位	访问	描述
15:11	Res	Reserved 复位值: 0x0
10	r/w	DATA_OV_INT : 数据溢出中断 0: 未发生数据溢出中断请求 1: 发生数据溢出中断请求 复位值: 0x0
9	r/w	TX_INT : 发送中断 0: 未发生发送中断请求 1: 发生发送中断请求 复位值: 0x0
8	r/w	RX_INT : 接收中断 0: 未发生接收中断请求 1: 发生接收中断请求 注: 当对接收缓冲器释放时, 如果缓冲器中还有数据未读走则会再一次产生接收中断请求 复位值: 0x0
7:1	Res	Reserved 复位值: 0x0
0	r/w	CAN_INT: CAN 中断 0: CAN 中断未发生 1: CAN 中断已发生 注: 该位表示当前是否有中断请求 <ul style="list-style-type: none"> ● BUS_ERR_INT ● ARBIT_LOST_INT ● ERR_PASSIVE_INT ● WAKEUP_INT ● DATA_OV_INT ● ERR_WARN_INT ● TX_INT ● RX_INT 复位值: 0x0

25.5.13 CAN 接收缓冲器状态 (RX_BUF_ST)

偏移地址: 0x0048

表 433 CAN 接收缓冲器状态

位	访问	描述
31:22	Res	Reserved 复位值: 0x0
21:16	r/w	RX_BUF_START_ADDR: 接收缓冲器开始地址 注: 用于表示当前接收缓冲器读指针所指向的地址; 复位值: 0x0

位	访问	描述
15:5	Res	Reserved 复位值: 0x0
4:0	ro	RX_MSG_CNT: 接收信息计数 注: 用于记录接收缓冲器当前存在多少有效信息。当接收到信息后该寄存器自动增加, 在释放接收缓冲器后自动减少。 复位值: 0x0

注: 该寄存器仅可在初始化模式下配置。

25.5.14 CAN 发送帧信息寄存器 (TX_FRM_INFO)

偏移地址: 0x0050

表 434 CAN 发送帧信息寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	r/w	FF: 帧格式 0: 标准帧格式 1: 扩展帧格式 复位值: 0x0
6	r/w	RTR: 远程帧请求 0: 数据帧格式传输 1: 远程帧格式传输 复位值: 0x0
5:4	Res	Reserved 复位值: 0x0
3:0	r/w	DLC: 发送数据长度 4'b0000: 0 - 字节 4'b0001: 1 - 字节 4'b0010: 2 - 字节 4'b0011: 3 - 字节 4'b0100: 4 - 字节 4'b0101: 5 - 字节 4'b0110: 6 - 字节 4'b0111: 7 - 字节 4'b1000: 8 - 字节 4'b1001-4'b1111: Reserved 复位值: 0x0

25.5.15 CAN 发送标识符 (TX_ID)

偏移地址: 0x0054

表 435 CAN 发送标识符

位	访问	描述
31:29	Res	Reserved 复位值: 0x0
28:18	r/w	ID[28:18]: 标识符 28~18 比特 注: 应用于标准帧格式和扩展帧格式 复位值: 0x0
17:0	r/w	ID[17:0]: 标识符 17~0 比特 注: 应用于扩展帧格式 复位值: 0x0

25.5.16 CAN 发送数据寄存器 0 (TX_DATA0)

偏移地址: 0x0058

表 436 CAN 发送数据寄存器 0

位	访问	描述
31:24	r/w	TX_DATA_BYTE3: 发送字节 3 复位值: 0x0
23:16	r/w	TX_DATA_BYTE2: 发送字节 2 复位值: 0x0
15:8	r/w	TX_DATA_BYTE1: 发送字节 1 复位值: 0x0
7:0	r/w	TX_DATA_BYTE0: 发送字节 0 复位值: 0x0

25.5.17 CAN 发送数据寄存器 1 (TX_DATA1)

偏移地址: 0x005C

表 437 CAN 发送数据寄存器 1

位	访问	描述
31:24	r/w	TX_DATA_BYTE7: 发送字节7 复位值: 0x0
23:16	r/w	TX_DATA_BYTE6: 发送字节6 复位值: 0x0
15:8	r/w	TX_DATA_BYTE5: 发送字节5 复位值: 0x0
7:0	r/w	TX_DATA_BYTE4: 发送字节4 复位值: 0x0

25.5.18 CAN 接收帧信息寄存器 (RX_FRM_INFO)

偏移地址: 0x0060

表 438 CAN 接收帧信息寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	ro	FF: 帧格式 0: 标准帧格式 1: 扩展帧格式 复位值: 0x0
6	ro	RTR: 远程帧请求 0: 数据帧格式传输 1: 远程帧格式传输 复位值: 0x0
5:4	Res	Reserved 复位值: 0x0
3:0	ro	DLC: 数据长度 4'b0000: 0 -字节 4'b0001: 1 -字节 4'b0010: 2 -字节 4'b0011: 3 -字节 4'b0100: 4 -字节 4'b0101: 5 -字节 4'b0110: 6 -字节 4'b0111: 7 -字节 4'b1000: 8 -字节 4'b1001-4'b1111: Reserved 复位值: 0x0

25.5.19 CAN 接收标识符 (RX_ID)

偏移地址: 0x0064

表 439 CAN 接收标识符

位	访问	描述
31:29	Res	Reserved 复位值: 0x0
28:18	ro	ID[28:18]: 标识符 28~18 比特 注: 应用于标准帧格式和扩展帧格式 复位值: 0x0
17:0	ro	ID[17:0]: 标识符 17~0 比特 注: 应用于扩展帧格式 复位值: 0x0

25.5.20 CAN 接收数据寄存器 0 (RX_DATA0)

偏移地址: 0x0068

表 440 CAN 接收数据寄存器 0

位	访问	描述
31:24	ro	RX_DATA_BYTE3: 接收字节 3 复位值: 0x0
23:16	ro	RX_DATA_BYTE2: 接收字节 2 复位值: 0x0
15:8	ro	RX_DATA_BYTE1: 接收字节 1 复位值: 0x0
7:0	ro	RX_DATA_BYTE0: 接收字节 0 复位值: 0x0

25.5.21 CAN 接收数据寄存器 1 (RX_DATA1)

偏移地址: 0x006C

表 441 CAN 接收数据寄存器 1

位	访问	描述
31:24	ro	RX_DATA_BYTE7: 接收字节 7 复位值: 0x0
23:16	ro	RX_DATA_BYTE6: 接收字节 6 复位值: 0x0
15:8	ro	RX_DATA_BYTE5: 接收字节 5 复位值: 0x0
7:0	ro	RX_DATA_BYTE4: 接收字节 4 复位值: 0x0

25.5.22 CAN 仲裁丢失位 (CAN_ARBIT_LOST_BIT)

偏移地址: 0x0070

表 442 CAN 仲裁丢失位

位	访问	描述
31	ro	RTR_ARBIT_ST: RTR 位仲裁状态 0: CAN 在 RTR 位未发生仲裁丢失 1: CAN 在 RTR 位发生仲裁丢失 复位值: 0x0
30	ro	ID0_ARBIT_ST: ID0 位仲裁状态 0: CAN 在 ID0 位未发生仲裁丢失 1: CAN 在 ID0 位发生仲裁丢失 复位值: 0x0

位	访问	描述
29	ro	ID1_ARBIT_ST: ID1 位仲裁状态 0: CAN 在 ID1 位未发生仲裁丢失 1: CAN 在 ID1 位发生仲裁丢失 复位值: 0x0
28	ro	ID2_ARBIT_ST: ID2 位仲裁状态 0: CAN 在 ID2 位未发生仲裁丢失 1: CAN 在 ID2 位发生仲裁丢失 复位值: 0x0
27	ro	ID3_ARBIT_ST: ID3 位仲裁状态 0: CAN 在 ID3 位未发生仲裁丢失 1: CAN 在 ID3 位发生仲裁丢失 复位值: 0x0
26	ro	ID4_ARBIT_ST: ID4 位仲裁状态 0: CAN 在 ID4 位未发生仲裁丢失 1: CAN 在 ID4 位发生仲裁丢失 复位值: 0x0
25	ro	ID5_ARBIT_ST: ID5 位仲裁状态 0: CAN 在 ID5 位未发生仲裁丢失 1: CAN 在 ID5 位发生仲裁丢失 复位值: 0x0
24	ro	ID6_ARBIT_ST: ID6 位仲裁状态 0: CAN 在 ID6 位未发生仲裁丢失 1: CAN 在 ID6 位发生仲裁丢失 复位值: 0x0
23	ro	ID7_ARBIT_ST: ID7 位仲裁状态 0: CAN 在 ID7 位未发生仲裁丢失 1: CAN 在 ID7 位发生仲裁丢失 复位值: 0x0
22	ro	ID8_ARBIT_ST: ID8 位仲裁状态 0: CAN 在 ID8 位未发生仲裁丢失 1: CAN 在 ID8 位发生仲裁丢失 复位值: 0x0
21	ro	ID9_ARBIT_ST: ID9 位仲裁状态 0: CAN 在 ID9 位未发生仲裁丢失 1: CAN 在 ID9 位发生仲裁丢失 复位值: 0x0
20	ro	ID10_ARBIT_ST: ID10 位仲裁状态 0: CAN 在 ID10 位未发生仲裁丢失 1: CAN 在 ID10 位发生仲裁丢失 复位值: 0x0

位	访问	描述
19	ro	ID11_ARBIT_ST: ID11 位仲裁状态 0: CAN 在 ID11 位未发生仲裁丢失 1: CAN 在 ID11 位发生仲裁丢失 复位值: 0x0
18	ro	ID12_ARBIT_ST: ID12 位仲裁状态 0: CAN 在 ID12 位未发生仲裁丢失 1: CAN 在 ID12 位发生仲裁丢失 复位值: 0x0
17	ro	ID13_ARBIT_ST: ID13 位仲裁状态 0: CAN 在 ID13 位未发生仲裁丢失 1: CAN 在 ID13 位发生仲裁丢失 复位值: 0x0
16		ID14_ARBIT_ST: ID14 位仲裁状态 0: CAN 在 ID14 位未发生仲裁丢失 1: CAN 在 ID14 位发生仲裁丢失 复位值: 0x0
15	ro	ID15_ARBIT_ST: ID15 位仲裁状态 0: CAN 在 ID15 位未发生仲裁丢失 1: CAN 在 ID15 位发生仲裁丢失 复位值: 0x0
14	ro	ID16_ARBIT_ST: ID16 位仲裁状态 0: CAN 在 ID16 位未发生仲裁丢失 1: CAN 在 ID16 位发生仲裁丢失 复位值: 0x0
13	ro	ID17_ARBIT_ST: ID17 位仲裁状态 0: CAN 在 ID17 位未发生仲裁丢失 1: CAN 在 ID17 位发生仲裁丢失 复位值: 0x0
12	ro	IDE_ARBIT_ST: IDE 位仲裁状态 0: CAN 在 IDE 位未发生仲裁丢失 1: CAN 在 IDE 位发生仲裁丢失 复位值: 0x0
11	ro	SRTR_ARBIT_ST: SRTR 位仲裁状态 0: CAN 在 SRTR 位未发生仲裁丢失 1: CAN 在 SRTR 位发生仲裁丢失 复位值: 0x0
10	ro	ID18_ARBIT_ST: ID18 位仲裁状态 0: CAN 在 ID18 位未发生仲裁丢失 1: CAN 在 ID18 位发生仲裁丢失 复位值: 0x0

位	访问	描述
9	ro	ID19_ARBIT_ST: ID19 位仲裁状态 0: CAN 在 ID19 位未发生仲裁丢失 1: CAN 在 ID19 位发生仲裁丢失 复位值: 0x0
8	ro	ID20_ARBIT_ST: ID20 位仲裁状态 0: CAN 在 ID20 位未发生仲裁丢失 1: CAN 在 ID20 位发生仲裁丢失 复位值: 0x0
7	ro	ID21_ARBIT_ST: ID21 位仲裁状态 0: CAN 在 ID21 位未发生仲裁丢失 1: CAN 在 ID21 位发生仲裁丢失 复位值: 0x0
6	ro	ID22_ARBIT_ST: ID22 位仲裁状态 0: CAN 在 ID22 位未发生仲裁丢失 1: CAN 在 ID22 位发生仲裁丢失 复位值: 0x0
5	ro	ID23_ARBIT_ST: ID23 位仲裁状态 0: CAN 在 ID23 位未发生仲裁丢失 1: CAN 在 ID23 位发生仲裁丢失 复位值: 0x0
4	ro	ID24_ARBIT_ST: ID24 位仲裁状态 0: CAN 在 ID24 位未发生仲裁丢失 1: CAN 在 ID24 位发生仲裁丢失 复位值: 0x0
3	ro	ID25_ARBIT_ST: ID25 位仲裁状态 0: CAN 在 ID25 位未发生仲裁丢失 1: CAN 在 ID25 位发生仲裁丢失 复位值: 0x0
2	ro	ID26_ARBIT_ST: ID26 位仲裁状态 0: CAN 在 ID26 位未发生仲裁丢失 1: CAN 在 ID26 位发生仲裁丢失 复位值: 0x0
1	ro	ID27_ARBIT_ST: ID27 位仲裁状态 0: CAN 在 ID27 位未发生仲裁丢失 1: CAN 在 ID27 位发生仲裁丢失 复位值: 0x0
0	ro	ID28_ARBIT_ST: ID28 位仲裁状态 0: CAN 在 ID28 位未发生仲裁丢失 1: CAN 在 ID28 位发生仲裁丢失 复位值: 0x0

25.5.23 CAN 错误信息寄存器 (RX_ERR_INFO)

偏移地址: 0x0074

表 443 CAN 错误信息寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7:6	ro	ECT: 错误类型 2'B00: 位错误 2'B01: 格式错误 2'B10: 填充错误 2'B11: 其他类型错误 复位值: 0x0
5	r/w	DIR: 通信方向 0: 错误发生在发送阶段 1: 错误发生在接收阶段 复位值: 0x0
4:0	ro	SC: 信息区间段 5'b00011: 帧起始 5'b00010: ID28 ~ ID21 5'b00110: ID20 ~ ID18 5'b00100: SRTR 位 5'b00101: IDE 位 5'b00111: ID17 ~ ID13 5'b01111: ID12 ~ ID5 5'b01110: ID4 ~ ID0 5'b01100: RTR 位 5'b01101: Reserved 1 5'b01001: Reserved 0 5'b01011: DLC 5'b01010: 数据值 5'b01000: CRC 值 5'b11000: CRC 界定符 5'b11001: 应答 5'b11011: 应答界定符 5'b11010: 帧结尾 5'b10010: 帧间空间 5'b10001: 主动错误标志 5'b10110: 被动错误标志 5'b10011: 容许显性位 5'b10111: 错误界定符 5'b11100: 过载标志 Others: Reserved

位	访问	描述
		复位值: 0x0

25.5.24 CAN 错误计数寄存器 (ERR_CNT)

偏移地址: 0x0078

表 444 CAN 错误计数寄存器

位	访问	描述
31:24	Res	Reserved 复位值: 0x0
23:16	r/w	REC: 接收错误计数 注: 根据 CAN2.0 规范, 在接收存在错误时计数器自动增加, 无错误时会自动减少。接收错误计数器与发送错误计数器一起作为 CAN 总线传输质量的指示; 当接收错误计数器达到以下等级会触发特殊事件: 当计数器达到错误报警阈值时如果错误报警中断已使能则会产生一个错误报警中断; 当计数超过 127 时, CAN 进入被动错误状态并发送一个主动错误。如果被动错误中断使能已打开则会产生一个被动错误中断 硬件复位或发生总线关闭后, 计数器自动设置为 0。该寄存器可在初始化模式下进行配置, 在工作模式下为只读; 复位值: 0x0
15:8	Res	Reserved 复位值: 0x0
7:0	r/w	TEC: 发送错误计数 注: 根据 CAN2.0 规范, 在发送存在错误时计数器自动增加, 无错误时会自动减少。接收错误计数器与发送错误计数器一起作为 CAN 总线传输质量的指示; 当发送错误计数器达到以下等级会触发特殊事件: 当计数器达到错误报警阈值时如果错误报警中断已使能则会产生一个错误报警中断; 当计数超过 127 时, CAN 进入被动错误状态并发送一个主动错误。如果被动错误中断使能已打开则会产生一个被动错误中断 当计数超过 255 时, CAN 进入总线关闭状态并自动进入初始化模式。如果错误报警中断使能已打开则会产生一个错误报警中断; 硬件复位或发生总线关闭后, 计数器自动设置为 0。该寄存器可在初始化模式下进行配置, 在工作模式下为只读; 在总线关闭状态下向 TEC 写入 0-254 的数值可以清除总线关闭标志, 在退出初始化模式后, CAN 将等待一个总线空闲序列; 在初始化模式下向 TEC 写入 255 将启动 CPU 驱动的总线关闭事件, 在 CAN 重新进入工作模式之前, 不会发生新的错误和总线状态改变的响应, 此时总线关闭事件将完全像是由总线错误强制执行, 这意味着再次进入初始化模式时发送错误计数器被初始化为 127, 接收计数器清零, 相关状态和中断将被置位。现在退出复位模式将执行总线断开恢复, 等待总线出现 128 次空闲序列; 复位值: 0x0

注：该寄存器仅可在初始化模式下配置。

25.5.25 CAN 接收缓冲器 (RX_FIFO)

表 445 CAN 接收缓冲器

偏移地址	缓冲器地址	访问	位宽
80H	0H	r/w	32bit
84H	4H	r/w	32bit
88H	8H	r/w	32bit
8CH	CH	r/w	32bit
90H	10H	r/w	32bit
94H	14H	r/w	32bit
98H	18H	r/w	32bit
9CH	1CH	r/w	32bit
A0H	20H	r/w	32bit
A4H	24H	r/w	32bit
A8H	28H	r/w	32bit
ACH	2CH	r/w	32bit
B0H	30H	r/w	32bit
B4H	34H	r/w	32bit
B8H	38H	r/w	32bit
BCH	3CH	r/w	32bit

注：该寄存器仅可在初始化模式下配置。

26 键入中断 (IOW)

26.1 概述

键入中断模块提供了 8 个独立的 IO 输入检测通道，在所有低功耗模式下，当目标检测边沿发生时，都可以发出中断唤醒系统。

26.2 主要功能

- 支持 8 个独立的通道进行按键检测
- 每个通道独立可编程上升沿检测或者下降沿检测
- 每个通道独立可编程中断产生使能

- Sleep/Deep Sleep/Stop 模式下唤醒系统

26.3 模块框图

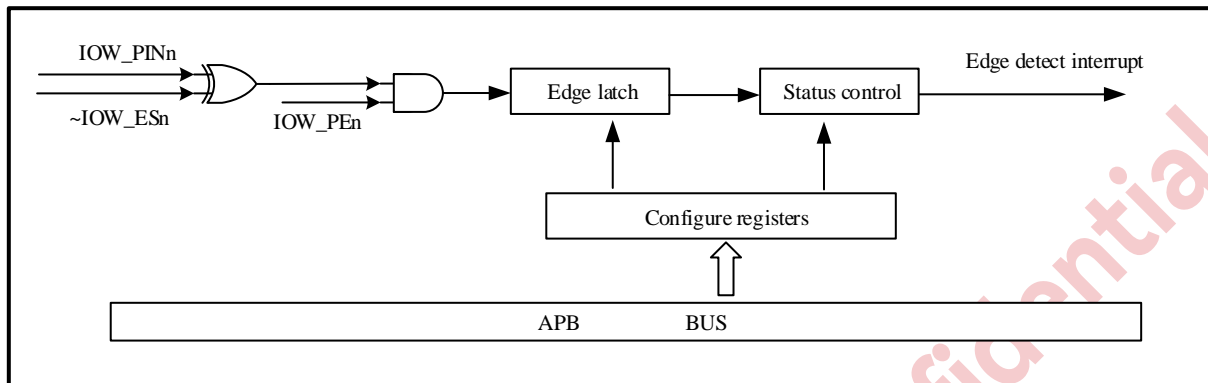


图 265 IOW 模块架构图

26.4 功能描述

26.4.1 通道输入源

表 446 键入中断通道输入源

通道	输入源
PIN0_IN0	XP37
PIN1_IN0	XP36
PIN2_IN0	XPE2
PIN3_IN0	XPD6
PIN4_IN0	XPD4
PIN5_IN0	XP96
PIN6_IN0	XP95
PIN7_IN0	XP70

26.4.2 IOW 中断号

表 447 IOW 中断号

名称	中断号
IOW_INT	23

26.5 寄存器

键入中断寄存器的基地址为 0x4000_7400,下表为键入中断的相关寄存器描述。

表 448 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	键入中断中断使能寄存器	32	r/w	0x00000000
0x04	键入中断控制寄存器	32	r/w	0x00000000
0x08	键入中断边沿选择寄存器	32	r/w	0x00000000
0x0C	键入中断状态寄存器	32	r/w	0x00000000

26.5.1 键入中断中断使能寄存器 (IOW_INTEN)

偏移地址: 0x0000

表 449 键入中断中断使能寄存器

位	访问	描述
31:1	Res	Reserved 复位值: 0x0
0	r/w	IOW_INT_EN: 键入中断中断使能控制 0: 关闭 1: 使能 复位值: 0x0

26.5.2 键入中断控制寄存器 (IOW_CTRL)

偏移地址: 0x0004

表 450 键入中断控制寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	r/w	IOW_PIN7_EN: 键入中断通道 7 使能控制 0: 关闭 1: 使能 复位值: 0x0
6	r/w	IOW_PIN6_EN: 键入中断通道 6 使能控制 0: 关闭 1: 使能 复位值: 0x0

位	访问	描述
5	r/w	IOW_PIN5_EN: 键入中断通道 5 使能控制 0: 关闭 1: 使能 复位值: 0x0
4	r/w	IOW_PIN4_EN: 键入中断通道 4 使能控制 0: 关闭 1: 使能 复位值: 0x0
3	r/w	IOW_PIN3_EN: 键入中断通道 3 使能控制 0: 关闭 1: 使能 复位值: 0x0
2	r/w	IOW_PIN2_EN: 键入中断通道 2 使能控制 0: 关闭 1: 使能 复位值: 0x0
1	r/w	IOW_PIN1_EN: 键入中断通道 1 使能控制 0: 关闭 1: 使能 复位值: 0x0
0	r/w	IOW_PIN0_EN: 键入中断通道 0 使能控制 0: 关闭 1: 使能 复位值: 0x0

26.5.3 键入中断边沿选择寄存器 (IOW_EDGES)

偏移地址: 0x0008

表 451 键入中断边沿选择寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	r/w	PIN7_EDGE_SEL: 键入中断通道 7 检测边沿选择 0: 下降沿 1: 上升沿 复位值: 0x0
6	r/w	PIN6_EDGE_SEL: 键入中断通道 6 检测边沿选择 0: 下降沿 1: 上升沿 复位值: 0x0

位	访问	描述
5	r/w	PIN5_EDGE_SEL: 键入中断通道 5 检测边沿选择 0: 下降沿 1: 上升沿 复位值: 0x0
4	r/w	PIN4_EDGE_SEL: 键入中断通道 4 检测边沿选择 0: 下降沿 1: 上升沿 复位值: 0x0
3	r/w	PIN3_EDGE_SEL: 键入中断通道 3 检测边沿选择 0: 下降沿 1: 上升沿 复位值: 0x0
2	r/w	PIN2_EDGE_SEL: 键入中断通道 2 检测边沿选择 0: 下降沿 1: 上升沿 复位值: 0x0
1	r/w	PIN1_EDGE_SEL: 键入中断通道 1 检测边沿选择 0: 下降沿 1: 上升沿 复位值: 0x0
0	r/w	PIN0_EDGE_SEL: 键入中断通道 0 检测边沿选择 0: 下降沿 1: 上升沿 复位值: 0x0

26.5.4 键入中断状态寄存器 (IOW_STA)

偏移地址: 0x000C

表 452 键入中断状态寄存器

位	访问	描述
31:8	Res	Reserved 复位值: 0x0
7	r/w	IOW_INT_STA: 键入中断通道 7 检测状态寄存器, 软件写 1 清 0 0: 未检测到目标沿 1: 检测到目标沿 复位值: 0x0
6	r/w	IOW_INT_STA: 键入中断通道 6 检测状态寄存器, 软件写 1 清 0 0: 未检测到目标沿 1: 检测到目标沿 复位值: 0x0

位	访问	描述
5	r/w	IOW_INT_STA: 键入中断通道 5 检测状态寄存器, 软件写 1 清 0 0: 未检测到目标沿 1: 检测到目标沿 复位值: 0x0
4	r/w	IOW_INT_STA: 键入中断通道 4 检测状态寄存器, 软件写 1 清 0 0: 未检测到目标沿 1: 检测到目标沿 复位值: 0x0
3	r/w	IOW_INT_STA: 键入中断通道 3 检测状态寄存器, 软件写 1 清 0 0: 未检测到目标沿 1: 检测到目标沿 复位值: 0x0
2	r/w	IOW_INT_STA: 键入中断通道 2 检测状态寄存器, 软件写 1 清 0 0: 未检测到目标沿 1: 检测到目标沿 复位值: 0x0
1	r/w	IOW_INT_STA: 键入中断通道 1 检测状态寄存器, 软件写 1 清 0 0: 未检测到目标沿 1: 检测到目标沿 复位值: 0x0
0	r/w	IOW_INT_STA: 键入中断通道 0 检测状态寄存器, 软件写 1 清 0 0: 未检测到目标沿 1: 检测到目标沿 复位值: 0x0

27 低压检测 (LVD)

27.1 概述

低电压检测电路用于比较电源电压和目标检测电压，当发生低电压情况时产生内部复位或内部中断。

LVD1 检测到低电压后发生 LVD Reset，默认开启；

LVD2 检测到低电压后发生 LVD Interrupt，可配置滤波功能，默认关闭。

27.2 主要功能

- 低电压检测模块 1
 - 检测 VCCIO 电压
 - 通过软件打开或关闭电压检测使能
 - 电压检测等级选择
 - ◆ $1.80+0.05V/-0.05V$
 - ◆ $1.90+0.05V/-0.05V$
 - ◆ $2.00+0.05V/-0.05V$
 - ◆ $2.20+0.05V/-0.05V$
 - ◆ $2.40+0.05V/-0.05V$
 - ◆ $2.60+0.05V/-0.05V$
 - ◆ $2.80+0.05V/-0.05V$
 - ◆ $3.00+0.05V/-0.05V$
 - ◆ $3.20+0.05V/-0.05V$
 - ◆ $3.40+0.05V/-0.05V$
 - ◆ $3.60+0.05V/-0.05V$
 - ◆ $3.80+0.05V/-0.05V$
 - ◆ $4.00+0.05V/-0.05V$
 - ◆ $4.20+0.05V/-0.05V$
 - ◆ $4.40+0.05V/-0.05V$
 - ◆ $4.50+0.05V/-0.05V$
 - 当 VCCIO 电压等级低于检测电压时可产生芯片复位信号

- 低电压检测模块 2
 - 检测 VCCIO 电压
 - 通过软件打开或关闭电压检测使能
 - 电压检测等级选择
 - ◆ $1.80+0.05V/-0.05V$
 - ◆ $1.90+0.05V/-0.05V$
 - ◆ $2.00+0.05V/-0.05V$
 - ◆ $2.20+0.05V/-0.05V$
 - ◆ $2.40+0.05V/-0.05V$
 - ◆ $2.60+0.05V/-0.05V$
 - ◆ $2.80+0.05V/-0.05V$
 - ◆ $3.00+0.05V/-0.05V$
 - ◆ $3.20+0.05V/-0.05V$
 - ◆ $3.40+0.05V/-0.05V$
 - ◆ $3.60+0.05V/-0.05V$
 - ◆ $3.80+0.05V/-0.05V$
 - ◆ $4.00+0.05V/-0.05V$
 - ◆ $4.20+0.05V/-0.05V$
 - ◆ $4.40+0.05V/-0.05V$
 - ◆ $4.50+0.05V/-0.05V$
 - 当 VCCIO 电压等级低于检测电压时可产生中断请求信号

27.3 模块框图

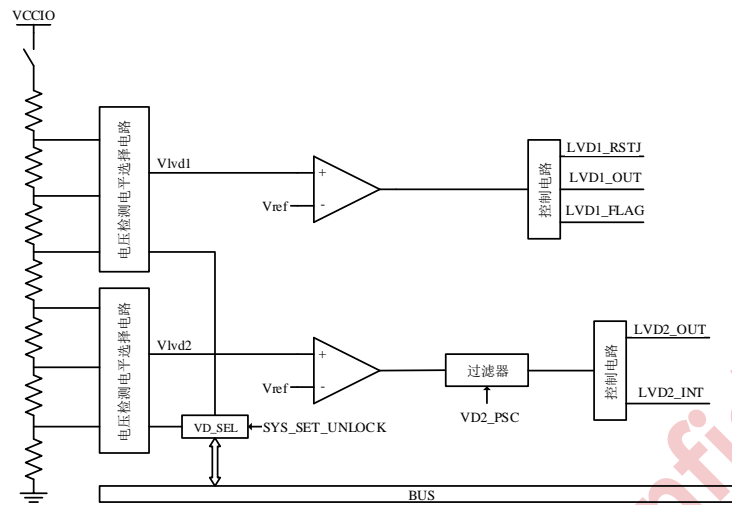


图 266 LVD 模块框图

27.4 功能描述

27.4.1 LVD 档位

LVD1 及 LVD2 分别对应的低电压检测档位，详见下表。

表 453 LVD 档位

序号	档位选择	LVD1 档位电压(V)	LVD2 档位电压(V)
1	4'B1111	4.5	4.5
2	4'B1110	4.4	4.4
3	4'B1101	4.2	4.2
4	4'B1100	4	4
5	4'B1011	3.8	3.8
6	4'B1010	3.6	3.6
7	4'B1001	3.4	3.4
8	4'B1000	3.2	3.2
9	4'B0111	3	3
10	4'B0110	2.8	2.8
11	4'B0101	2.6	2.6
12	4'B0100	2.4	2.4
13	4'B0011	2.2	2.2
14	4'B0010	2	2

序号	档位选择	LVD1 档位电压(V)	LVD2 档位电压(V)
15	4'B0001	1.9	1.9
16	4'B0000	1.8	1.8

注：

1. 当需要更改 LVD 检测电压档位时，需要确保完成系统配置寄存器安全锁解锁，即 SYS_SET_UNLOCK 有效。
2. LVD1 系统默认打开，默认档位为 2.8V，LVD2 系统默认关闭，默认档位为 3.6V。

27.4.2 LVD 功能

LVD 用于检测 VCCIO 电压，当 VCCIO 电压低于预设检测电压时，会产生相应的报警中断或复位动作，LVD1 用于低电压复位，LVD2 用于低电压报警；当 LVD 检测到低电压并发生相应动作后，可通过对应的低电压检测标志位查看；具体工作波形可详见下图。

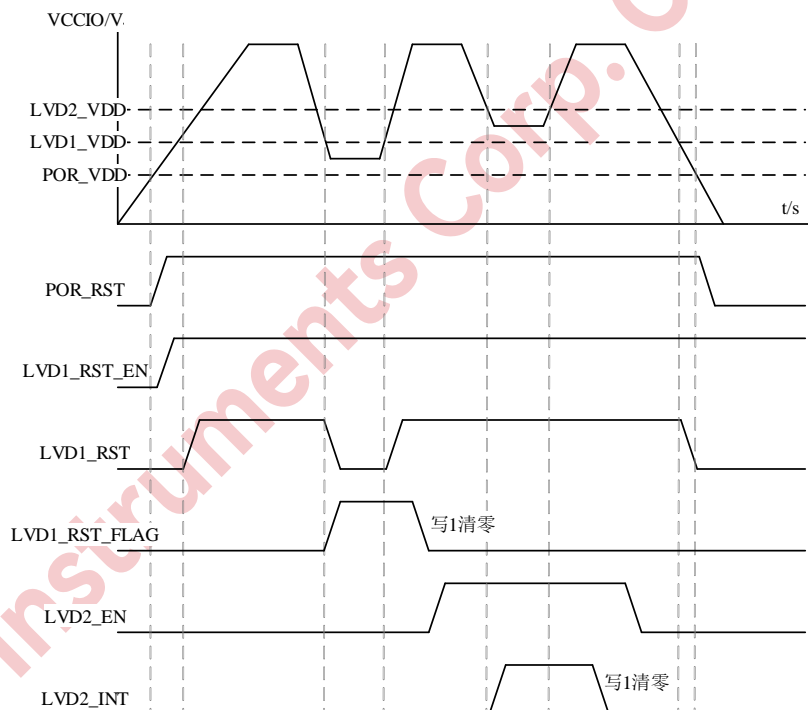


图 267 LVD 工作波形图

27.4.3 LVD 中断号

打开 LVD2 及相应中断使能，LVD2 在检测到 VCCIO 电压低于预设档位电压时，发生 LVD 中断；

表 454 LVD 中断号

名称	中断号
LVD_INT	1

27.5 寄存器

LVD 寄存器的基地址为 0x4001_0000,下表为键入中断的相关寄存器描述。

表 455 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x0200	电压检测模块 1 控制寄存器	32	r/w	0x00000611
0x0204	电压检测模块 1 状态寄存器	32	r/w	0x00000000
0x0208	电压检测模块 2 控制寄存器	32	r/w	0x00000A00
0x020C	电压检测模块 2 状态寄存器	32	r/w	0x00000000

27.5.1 电压检测模块 1 控制寄存器 (VD1_CTRL)

偏移地址：0x0200

表 456 电压检测模块 1 控制寄存器

位	访问	描述
31:12	Res	Reserved 复位值：0x0
11:10	r/w	VD1_SEL[3:2] 电压检测模块 1 档位选择控制 (位 2&3) 注：该位只能为 POR 复位或外部复位 复位值：0x1 注： VD1_SEL，电压检测模块 1 档位选择控制 0000: 1.80+0.05V/-0.05V 0001: 1.90+0.05V/-0.05V 0010: 2.00+0.05V/-0.05V 0011: 2.20+0.05V/-0.05V 0100: 2.40+0.05V/-0.05V 0101: 2.60+0.05V/-0.05V 0110: 2.80+0.05V/-0.05V 0111: 3.00+0.05V/-0.05V 1000: 3.20+0.05V/-0.05V 1001: 3.40+0.05V/-0.05V

		1010: 3.60+0.05V/-0.05V 1011: 3.80+0.05V/-0.05V 1100: 4.00+0.05V/-0.05V 1101: 4.20+0.05V/-0.05V 1110: 4.40+0.05V/-0.05V 1111: 4.50+0.05V/-0.05V
9	r/w	VD2_SEL[0] 电压检测模块 2 档位选择控制 (位 0) 注: 1. 该位只能为 POR 复位或外部复位 2. 完整定义见 VD2_CTRL[11:9] 处描述 复位值: 0x1
8	r/w	VD1_SEL[0] 电压检测模块 1 档位选择控制 (位 0) 注: 1. 该位只能为 POR 复位或外部复位 2. 完整定义见 VD1_CTRL[11:10] 处描述 复位值: 0x0
7:5	Res	Reserved 复位值: 0x0
4	Res	VD1_RST_EN: 电压检测模块 1 复位芯片使能控制 0: 关闭 1: 开启 注: 该位只能为 POR 复位或外部复位 复位值: 0x1
3:1	Res	Reserved 复位值: 0x0
0	r/w	VD1_EN: 电压检测模块 1 使能控制 0: 关闭 1: 开启 注: 该位只能为 POR 复位或外部复位 复位值: 0x1

注: 在更新该组寄存器之前需要确保完成系统配置寄存器安全锁解锁, 即 SYS_SET_UNLOCK 有效。

27.5.2 电压检测模块 1 状态寄存器 (VD1_ST)

偏移地址: 0x0204

表 457 电压检测模块 1 状态寄存器

位	访问	描述
31:2	Res	Reserved 复位值: 0x0
1	r/w	VD1_FLG, 电压检测模块 1 标记 0: $VCC \geq VD1_SEL$ 1: $VCC < VD1_SEL$ 注: 当 VD1_OUT =1 时, 该位硬件置位, 支持写 1 清零 复位值: 0x0
0	ro	VD1_OUT, 电压检测模块 1 输出 0: $VCC \geq VD1_SEL$ 1: $VCC < VD1_SEL$ 复位值: 0x0

27.5.3 电压检测模块 2 控制寄存器 (VD2_CTRL)

偏移地址: 0x0208

表 458 电压检测模块 2 控制寄存器

位	访问	描述
31:24	Res	Reserved 复位值: 0x0
23:16	r/w	VD2_THR, 电压检测模块 2 滤波阈值 该值用于设置对于 VD2_OUT 输出电平毛刺滤波, 当输出电平有效地保持 VD2_THR 个滤波周期, 才认为输出的电平有效。 注: 该位只能为 POR 复位或外部复位。 复位值: 0x0
15	Res	Reserved 复位值: 0x0
14:12	r/w	VD2_FILTER_CLK_PSC, 电压检测模块 2 滤波时钟预分频选择 000: 不分频 001: 2 分频 010: 4 分频 011: 8 分频 100: 16 分频 101: 32 分频 110: 64 分频 111: 128 分频 注: 该位只能为 POR 复位或外部复位 复位值: 0x0

位	访问	描述
11:9	r/w	<p>VD2_SEL[3:1]电压检测模块 2 档位选择控制</p> <p>注：该位只能为 POR 复位或外部复位</p> <p>复位值：0x5</p> <p>注：</p> <p>VD2_SEL，电压检测模块 2 档位选择控制</p> <p>0000: 1.80+0.05V/-0.05V</p> <p>0001: 1.90+0.05V/-0.05V</p> <p>0010: 2.00+0.05V/-0.05V</p> <p>0011: 2.20+0.05V/-0.05V</p> <p>0100: 2.40+0.05V/-0.05V</p> <p>0101: 2.60+0.05V/-0.05V</p> <p>0110: 2.80+0.05V/-0.05V</p> <p>0111: 3.00+0.05V/-0.05V</p> <p>1000: 3.20+0.05V/-0.05V</p> <p>1001: 3.40+0.05V/-0.05V</p> <p>1010: 3.60+0.05V/-0.05V</p> <p>1011: 3.80+0.05V/-0.05V</p> <p>1100: 4.00+0.05V/-0.05V</p> <p>1101: 4.20+0.05V/-0.05V</p> <p>1110: 4.40+0.05V/-0.05V</p> <p>1111: 4.50+0.05V/-0.05V</p>
8	r/w	<p>VD1_SEL[0]电压检测模块 1 档位选择控制 (位 1)</p> <p>注：</p> <p>1. 该位只能为 POR 复位或外部复位</p> <p>2. 完整定义见 VD1_CTRL[11:10] 处描述</p> <p>复位值：0x0</p>
7:1	Res	<p>Reserved</p> <p>复位值：0x0</p>
3	r/w	<p>VD2_IE: 电压检测模块 2 中断使能控制</p> <p>0: 关闭</p> <p>1: 开启</p> <p>注：该位只能为 POR 复位或外部复位</p> <p>复位值：0x0</p>
2:1	Res	<p>Reserved</p> <p>复位值：0x0</p>
0	r/w	<p>VD2_EN: 电压检测模块 2 使能控制</p> <p>0: 关闭</p> <p>1: 开启</p> <p>注：该位只能为 POR 复位或外部复位</p> <p>复位值：0x0</p>

注：在更新该组寄存器之前需要确保完成系统配置寄存器安全锁解锁，即 SYS_SET_UNLOCK 有效。

27.5.4 电压检测模块 2 状态寄存器 (VD2_ST)

偏移地址: 0x020C

表 459 电压检测模块 2 状态寄存器

位	访问	描述
31:2	Res	Reserved 复位值: 0x0
1	r/w	VD2_FLAG, 电压检测模块 2 标记 0: $VCC \geq VD2_SEL$ 1: $VCC < VD2_SEL$ 该位写 1 清零 复位值: 0x0
0	ro	VD2_OUT, 电压检测模块 2 输出 0: $VCC \geq VD2_SEL$ 1: $VCC < VD2_SEL$ 复位值: 0x0

28 嵌套中断向量控制器 (NVIC)

Cortex-M4F 集成嵌套向量中断控制器 (NVIC)，NVIC 对系统异常及外设中断提供的控制包括使能、优先级、清除、挂起、有效状态及向量表重置等功能，NVIC 能够处理多达 256 个可屏蔽中断通道，更多信息请参考 Cortex-M4F 技术参考手册。本产品的中断向量表如下表所示。

表 460 L 中断向量表

中断号	优先级	名称	向量地址	描述
-	-	-	0x00	初始堆栈指针
-	-3 (最高)	复位	0x04	复位
-14	-2	NMI	0x08	不可屏蔽中断
-13	-1	硬件故障	0x0C	所有故障类型
-12	可配置	MemManage	0x10	存储管理故障
-11	可配置	BusFault	0x14	总线故障
-10	可配置	UsageFault	0x18	指令故障
-	-	保留	-	-
-5	可配置	SVCall	0x2C	SVC 指令的可挂起请求
-	-	-	-	-
-2	可配置	PendSV	0x38	系统服务的可挂起请求
-1	可配置	SysTick	0x3C	SysTick 定时器递减到零
0	可配置	WDG	0x40	看门狗中断
1	可配置	LVD	0x44	低压检测中断
2	-	-	-	-
3	可配置	Flash	0x4C	Flash 中断
4	可配置	DMA	0x50	DMA 中断
5	可配置	CAN	0x54	CAN 中断
6	-	-	-	-
7	可配置	SPI	0x5C	SPI 中断
8	-	-	-	-
9	可配置	IOW	0x64	IOW 中断
10	可配置	I2C0	0x68	I2C0 中断
11	可配置	I2C1	0x6C	I2C1 中断

中断号	优先级	名称	向量地址	描述
12	可配置	SCI0	0x70	SCI0 中断
13	可配置	SCI1	0x74	SCI1 中断
14	可配置	SCI2	0x78	SCI2 中断
15	-	-	-	-
16	可配置	CMP0	0x80	CMP0 中断
17	可配置	CMP1	0x84	CMP1 中断
18	可配置	CMP2	0x88	CMP2 中断
19	可配置	CMP3	0x8C	CMP3 中断
20	可配置	ADC0	0x90	ADC0 中断
21	可配置	ADC1	0x94	ADC1 中断
22	可配置	BTM0	0x98	BTM0 中断
23	可配置	BTM1	0x9C	BTM1 中断
24	可配置	EBUS_INT0	0xA0	EBUS 中断 0
25	可配置	EBUS_INT1	0xA4	EBUS 中断 1
26	可配置	EBUS_INT2	0xA8	EBUS 中断 2
27	可配置	EBUS_INT3	0xAC	EBUS 中断 3
28	可配置	EBUS_INT4	0xB0	EBUS 中断 4
29	可配置	EBUS_INT5	0xB4	EBUS 中断 5
30	可配置	EBUS_INT6	0xB8	EBUS 中断 6
31	可配置	EBUS_INT7	0xBC	EBUS 中断 7
32	可配置	TIMA0	0xC0	TIMA0 中断
33	可配置	TIMA1	0xC4	TIMA1 中断
34	可配置	TIMH_PG0	0xC8	TIMH_PG0 中断
35	可配置	TIMG0	0xCC	TIMG0 中断
36	可配置	TIMG1	0xD0	TIMG1 中断
37	可配置	TIMG2	0xD4	TIMG2 中断
38	可配置	TIMS0	0xD8	TIMS0 中断
39	可配置	TIMS1	0xDC	TIMS1 中断
40	可配置	TIMS2	0xE0	TIMS2 中断
41	可配置	TIMS3	0xE4	TIMS3 中断
42	可配置	EBUS_INT8	0xE8	EBUS 中断 8
43	可配置	EBUS_INT9	0xEC	EBUS 中断 9

中断号	优先级	名称	向量地址	描述
44	可配置	EBUS_INT10	0xF0	EBUS 中断 10
45	可配置	EBUS_INT11	0xF4	EBUS 中断 11
46	可配置	EBUS_INT12	0xF8	EBUS 中断 12
47	可配置	EBUS_INT13	0xFC	EBUS 中断 13
48	可配置	EBUS_INT14	0x100	EBUS 中断 14
49	可配置	EBUS_INT15	0x104	EBUS 中断 15
50	可配置	TIMH_PG1	0x108	TIMH_PG1 中断
51	可配置	TIMH_PG2	0x10C	TIMH_PG2 中断
52	可配置	DNU	0x110	DNU 中断

29 附录

29.1 寄存器相关缩写词列表

寄存器说明中使用以下缩写词：

rw, r/w 读写，软件可以读写这些位

ro 只读，软件只能读取这些位

wo 只写，软件只能写入此位，读取此位返回无效数据

Res/res 保留位

声明

必易微保留在没有通知的情况下对其产品和产品说明书或规格书进行任何修改的权利。客户下单前请获取最新资料。产品说明书或规格书不用于作任何明示或暗示的保证包括但不限于产品的商用性、目的适用性或不侵犯他人权利等，也不用于作任何授权包括但不限于对必易微或第三方知识产权的授权。使用者在将必易微的产品整合到应用中时或使用过程中应确保该具体应用或使用不侵犯他人知识产权或其他权利，因该应用或使用引起纠纷或造成任何损失的，必易微不承担任何法律责任包括但不限于间接责任或偶然损失责任。未经必易微书面说明，必易微的产品非为用于人体植入器械和提供生命支持系统的目的而设计。本声明替代以往版本的声明。

Kiwi Instruments Corp. Confidential