



KPM32K07XX

32 位 ARM Cortex-M0+ 微控制器

用户手册



目录

1 简介	1
2 系统架构和存储器	1
2.1 概述	1
2.2 主要功能	1
2.3 模块框图	2
2.4 功能描述	2
2.4.1 地址空间映射	2
3 FLASH 存储器 (FLASH)	5
3.1 概述	5
3.2 主要功能	5
3.3 模块框图	5
3.4 功能描述	6
3.4.1 Flash 读延迟	6
3.4.2 Flash 烧写和擦除	6
3.4.3 Cache 功能	7
3.4.4 Flash 中断号	7
3.5 寄存器描述	7
3.5.1 Flash 地址寄存器 (FLH_ADR)	8
3.5.2 Flash 数据寄存器 (FLASH_DATA)	8
3.5.3 Flash 状态寄存器 (FLASH_ST)	8
3.5.4 Flash 控制寄存器 (FLH_CTRL)	10
3.5.5 Flash 命令寄存器 (FLH_CMD)	11
3.5.6 Flash 秘钥寄存器 (FLASH_KEY)	12
3.5.7 Flash 页保护寄存器 (FLH_PROT)	12
4 时钟 (CLOCK)	13
4.1 概述	13
4.2 主要功能	13
4.3 模块框图	14



4.4 功能描述.....	17
4.4.1 HIRC 时钟.....	17
4.4.2 PLL 时钟.....	17
4.4.3 LIRC 时钟.....	17
4.4.4 系统时钟组.....	17
4.4.5 与系统时钟组成异步关系的 IP 时钟.....	18
4.4.6 低功耗.....	19
5 复位 (RESET).....	20
5.1 概述.....	20
5.2 模块框图.....	21
5.3 功能描述.....	23
5.3.1 外部 RC 复位.....	23
5.3.2 上电复位 (POR).....	23
5.3.3 低电压检测 (LVD).....	24
5.3.4 看门狗定时器 (WDT).....	24
5.3.5 软件复位 (SYS).....	24
5.3.6 锁定复位 (LOCKUP).....	24
5.3.7 复位域.....	24
5.3.8 IP 软件复位控制位.....	24
6 功耗模式 (POWER_MODE).....	25
6.1 概述.....	25
6.2 主要功能.....	25
6.3 功能描述.....	25
6.3.1 电源供应.....	25
6.3.2 电源供应管理.....	25
6.3.3 不同模式下的模块状态.....	26
6.3.4 低功耗模式.....	27
7 系统寄存器 (SYSTEM_REG).....	33
7.1 概述.....	33
7.2 寄存器描述.....	33



7.2.1 芯片 ID 寄存器 (CHIP_ID).....	36
7.2.2 复位状态寄存器 (RST_ST).....	36
7.2.3 系统控制寄存器 (SYS_CFG).....	38
7.2.4 功耗模式控制寄存器 (PWR_MOD).....	39
7.2.5 系统配置密钥寄存器 (SYS_SET_KR).....	39
7.2.6 模拟 PMU 控制寄存器 (ANA_PMU_CTRL).....	40
7.2.7 IRC 控制寄存器 0 (IRC_CTRL_REG0).....	41
7.2.8 IRC 控制寄存器 1 (IRC_CTRL_REG1).....	41
7.2.9 IRC 控制寄存器 2 (IRC_CTRL_REG2).....	42
7.2.10 PLL 控制寄存器 (PLL_CTRL_REG).....	43
7.2.11 温度传感器控制寄存器 (TS_CTRL_REG).....	45
7.2.12 电压检测模块控制寄存器 (LVD_CTRL).....	45
7.2.13 电压检测模块状态寄存器 (LVD_ST).....	46
7.2.14 系统复位控制寄存器 (SYSRST_CTRL_REG).....	46
7.2.15 复位控制寄存器 0 (RST_CTRL_REG0).....	47
7.2.16 复位控制寄存器 1 (RST_CTRL_REG1).....	48
7.2.17 时钟门控寄存器 0 (CLK_GAT_REG0).....	49
7.2.18 时钟门控寄存器 1 (CLK_GAT_REG1).....	50
7.2.19 时钟选择寄存器 0 (CLK_SEL_REG0).....	51
7.2.20 时钟选择寄存器 1 (CLK_SEL_REG1).....	52
7.2.21 时钟选择寄存器 2 (CLK_SEL_REG2).....	52
7.2.22 噪声滤波时钟控制寄存器 (NF_CLK_CTRL_REG0).....	54
7.2.23 IO 噪声控制寄存器 0 (IO_NF_CTRL_REG0).....	55
7.2.24 IO 噪声控制寄存器 1 (IO_NF_CTRL_REG1).....	58
7.2.25 端口输入使能控制寄存器 0 (PAD_IE_CTRL_REG0).....	58
7.2.26 端口输入使能控制寄存器 1 (PAD_IE_CTRL_REG1).....	61
7.2.27 引脚驱动控制寄存器 0 (IO_DS_CTRL_REG0).....	61
7.2.28 引脚驱动控制寄存器 1 (IO_DS_CTRL_REG1).....	63
7.2.29 引脚上拉控制寄存器 0 (IO_PU_CTRL_REG0).....	63
7.2.30 引脚上拉控制寄存器 1 (IO_PU_CTRL_REG1).....	65



7.2.31	引脚下拉控制寄存器 0 (IO_PD_CTRL_REG0)	65
7.2.32	引脚下拉控制寄存器 1 (IO_PD_CTRL_REG1)	67
7.2.33	端口模式控制寄存器 0 (IO_MOD_CTRL_REG0)	67
7.2.34	端口模式控制寄存器 1 (IO_MOD_CTRL_REG1)	69
7.2.35	端口模式控制寄存器 2 (IO_MOD_CTRL_REG2)	71
7.2.36	功能复用选择寄存器 0 (AF_SEL_REG0)	71
7.2.37	功能复用选择寄存器 1 (AF_SEL_REG1)	72
7.2.38	功能复用选择寄存器 2 (AF_SEL_REG2)	73
7.2.39	功能复用选择寄存器 3 (AF_SEL_REG3)	74
7.2.40	功能复用选择寄存器 4 (AF_SEL_REG4)	75
7.2.41	端口开漏输出控制寄存器 0 (IO_OD_CTRL_REG0)	75
7.2.42	端口开漏输出控制寄存器 1 (IO_OD_CTRL_REG1)	77
7.2.43	调试控制寄存器 (DBG_CTRL_REG)	77
8	通用 I/O (GPIO)	78
8.1	概述	78
8.2	主要功能	78
8.3	模块框图	78
8.4	功能描述	79
8.4.1	GPIO 地址分布	79
8.4.2	GPIO 输出控制	79
8.5	寄存器描述	79
8.5.1	GPIOx 输入值 (GPIOx_DATA_IN)	80
8.5.2	GPIOx 输出值 (GPIOx_DATA_OUT)	80
8.5.3	GPIOx 输出值置位清零 (GPIOx_DATA_OUT_SET_CLR)	80
8.5.4	GPIOx 输出翻转 (GPIOx_TOG)	81
8.5.5	GPIOx 输出使能 (GPIOx_DOUT_EN)	81
8.5.6	GPIOx 输出使能清除 (GPIOx_DOUT_EN_CLR)	81
9	硬件计算加速器 (HAU)	82
9.1	概述	82
9.2	主要功能	82



9.3 模块框图.....	83
9.4 功能描述.....	83
9.4.1 Cordic 算法.....	83
9.4.2 TMU 工作时序.....	86
9.4.3 DIV 工作时序.....	87
9.5 寄存器描述.....	87
9.5.1 HAU 控制寄存器 (HAU_CTRL).....	88
9.5.2 HAU 数据寄存器 A (HAU_DATA_A).....	88
9.5.3 HAU 数据寄存器 B (HAU_DATA_B).....	88
9.5.4 HAU 数据寄存器 C (HAU_DATA_C).....	89
9.5.5 DIV 数据寄存器 A (DIV_DATA_A).....	89
9.5.6 DIV 数据寄存器 B (DIV_DATA_B).....	89
9.5.7 HAU 状态寄存器 (HAU_ST).....	90
10 模数转换器 (ADC).....	91
10.1 概述.....	91
10.2 主要功能.....	91
10.3 模块框图.....	92
10.4 功能描述.....	92
10.4.1 相关配置.....	92
10.4.2 转换顺序优先级.....	93
10.4.3 突发转换模式.....	96
10.4.4 硬件触发功能.....	99
10.4.5 结果后处理.....	99
10.4.6 ADC 中断号.....	99
10.5 寄存器描述.....	100
10.5.1 ADC 软件触发寄存器 (ADC_TRIG).....	101
10.5.2 ADC 控制寄存器 (ADC_CTRL).....	102
10.5.3 ADC 转换单元优先级寄存器 (ADC_SOC_PRI).....	103
10.5.4 ADC 中断使能寄存器 (ADC_INTEN).....	104
10.5.5 ADC 中断状态寄存器 (ADC_STA).....	105



10.5.6 ADC 中断状态清除寄存器 (ADC_STA_CLR)	106
10.5.7 ADC 转换单元触发溢出寄存器 (ADC_TRIG_OVF)	107
10.5.8 ADC 转换单元触发溢出清除寄存器 (ADC_OVF_CLR)	108
10.5.9 ADC 转换单元配置寄存器 (ADC_SOCx) (x=0, 1, 2, 3, 4, 5)	108
10.5.10 ADC PPB0 设置寄存器 (ADC_PPB0_CFG)	110
10.5.11 ADC PPB0 比较上限寄存器 (ADC_PPB0_COMPH)	110
10.5.12 ADC PPB0 比较下限寄存器 (ADC_PPB0_COMPL)	111
10.5.13 ADC 转换结果寄存器 (ADC_RESULTx) (x=0, 1, 2, 3, 4, 5)	111
10.5.14 ADC 模拟通道使能寄存器 (ADC_CHNEN)	111
11 高级定时器 (TIMP)	113
11.1 概述	113
11.2 主要功能	113
11.3 模块框图	114
11.4 功能描述	114
11.4.1 计数器介绍	114
11.4.2 计数器模式	115
11.4.3 时钟选择	123
11.4.4 强制输出模式	124
11.4.5 输出比较模式	125
11.4.6 PWM 模式	126
11.4.7 互补输出和死区插入	128
11.4.8 使用断路功能	130
11.4.9 发生外部事件时清除 OCxREF 信号	133
11.4.10 生成 6 步 PWM	134
11.4.11 单脉冲模式	135
11.4.12 连接霍尔传感器	137
11.4.13 定时器与外部触发同步	139
11.4.14 TIMP 中断号	140
11.5 寄存器描述	141
11.5.1 TIM 控制寄存器 0 (TIM_CR0)	142



11.5.2 TIM 控制寄存器 1 (TIM_CR1).....	143
11.5.3 TIM 从属模式控制寄存器 (TIM_SMCR).....	145
11.5.4 TIM 中断使能寄存器 (TIM_IER).....	147
11.5.5 TIM 状态寄存器 (TIM_SR).....	148
11.5.6 TIM 事件生成寄存器 (TIM_EGR).....	149
11.5.7 TIM 比较模式寄存器 0 (TIM_CM0).....	150
11.5.8 TIM 比较模式寄存器 1 (TIM_CM1).....	152
11.5.9 TIM 比较模式使能寄存器 (TIM_CER).....	154
11.5.10 TIM 计数器 (TIM_CNT).....	155
11.5.11 TIM 预分频器 (TIM_PSC).....	156
11.5.12 TIM 自动重载寄存器 (TIM_ARR).....	156
11.5.13 TIM 重复计数器寄存器 (TIM_RCR).....	156
11.5.14 TIM 比较寄存器 0 (TIM_CCR0).....	157
11.5.15 TIM 比较寄存器 1 (TIM_CCR1).....	157
11.5.16 TIM 比较寄存器 2 (TIM_CCR2).....	158
11.5.17 TIM 比较寄存器 3 (TIM_CCR3).....	158
11.5.18 TIM 比较寄存器 4 (TIM_CCR4).....	159
11.5.19 TIM 断路及死区寄存器 (TIM_BDTR).....	159
12 简版定时器 (TIMS).....	161
12.1 概述.....	161
12.2 主要功能.....	161
12.3 模块框图.....	162
12.4 功能描述.....	163
12.4.1 计数器介绍.....	163
12.4.2 计数器模式.....	163
12.4.3 时钟选择.....	164
12.4.4 捕获/比较通道.....	166
12.4.5 输入捕获模式.....	168
12.4.6 PWM 输入模式.....	169
12.4.7 强制输出模式.....	170



12.4.8 输出比较模式.....	170
12.4.9 PWM 模式.....	171
12.4.10 单脉冲模式.....	172
12.4.11 互补输出和死区插入	174
12.4.12 使用断路功能	175
12.4.13 定时器与外部触发同步	178
12.4.14 定时器同步	180
12.4.15 TIMS 中断号.....	183
12.5 寄存器描述.....	184
12.5.1 TIM 控制寄存器 (TIM_CR).....	185
12.5.2 TIM 模式控制寄存器 (TIM_MCR).....	187
12.5.3 TIM 捕获/比较模式寄存器 (TIM_CCMR).....	189
12.5.4 TIM 捕获/比较模式使能寄存器 (TIM_CCER).....	192
12.5.5 TIM 计数器 (TIM_CNT).....	194
12.5.6 TIM 预分频器 (TIM_PSC).....	194
12.5.7 TIM 自动重载寄存器 (TIM_ARR).....	195
12.5.8 TIM 捕获/比较寄存器 0 (TIM_CCRO)	195
12.5.9 TIM 捕获/比较寄存器 1 (TIM_CCR1)	195
12.5.10 TIM 断路及死区寄存器 (TIM_BDTR).....	196
12.5.11 TIM 事件生成寄存器 (TIM_EGR).....	196
12.5.12 TIM 中断使能寄存器 (TIM_IER).....	197
12.5.13 TIM 状态寄存器 (TIM_SR).....	198
12.5.14 TIM 输入复用控制寄存器 (TIM_IMCR).....	200
12.5.15 TIM 断路控制寄存器 (TIM_BCR0).....	201
12.5.16 TIM 断路控制寄存器 (TIM_BCR1).....	202
12.5.17 TIM 霍尔状态寄存器 (TIM_HSR).....	203
13 看门狗 (WDT)	204
13.1 概述	204
13.2 主要功能	204
13.3 模块框图	205



13.4 功能描述	205
13.4.1 预警中断功能	205
13.4.2 工作模式	205
13.4.3 寄存器安全锁功能	208
13.4.4 调试模式	208
13.4.5 时钟源	208
13.5 寄存器描述	208
13.5.1 WDT 控制寄存 (WDT_CTRL)	209
13.5.2 WDT 加载寄存器 (WDT_LOAD)	210
13.5.3 WDT 窗口加载寄存器 (WDT_WIN_LOAD)	210
13.5.4 WDT 状态寄存器 (WDT_ST)	210
13.5.5 WDT 计数器寄存器 (WDT_VAL)	211
13.5.6 WDT 密钥寄存器 (WDT_KR)	211
14 串行通信接口 (SCI)	212
14.1 概述	212
14.2 主要功能	212
14.3 模块框图	213
14.4 功能描述	214
14.4.1 工作模式	214
14.4.2 接口功能	214
14.4.3 SPI 模式	214
14.4.4 UART 通信模式	220
14.4.5 简易 I ² C 通信模式	222
14.4.6 SCI 中断号	224
14.5 寄存器描述	225
14.5.1 时钟分频选择寄存器 (SCI_CLK_PRS)	225
14.5.2 波特率设置寄存器 (SCI_BAUD)	226
14.5.3 模式控制寄存器 (SCI_MOD_CTL)	226
14.5.4 UART 模式寄存器 (UART_MOD_CTL)	227
14.5.5 SPI 模式寄存器 (SPI_MOD_CTL)	228



14.5.6 IIC 模式寄存器 (IIC_MOD_CTL)	229
14.5.7 发送数据寄存器 (SCI_TX_DAT)	230
14.5.8 接收数据寄存器 (SCI_RX_DAT)	230
14.5.9 中断使能寄存器 (SCI_IE)	230
14.5.10 中断状态寄存器 (SCI_STA)	231
14.5.11 中断状态清除寄存器 (SCI_STA_CLR)	232
15 模拟电压比较器 (CMP)	234
15.1 概述	234
15.2 主要功能	234
15.3 模块框图	235
15.4 功能描述	236
15.4.1 CMP 功能	236
15.4.2 PGA 功能	237
15.4.3 CMP 数字滤波	239
15.4.4 CMP 中断号	239
15.5 寄存器描述	239
15.5.1 CMP 模式控制寄存器 (CMP_MD_CTRL)	240
15.5.2 CMP 滤波控制寄存器 (CMP_FLT_CTRL)	241
15.5.3 CMP 输出控制寄存器 (CMP_OUT_CTRL)	243
15.5.4 CMP 状态寄存器 (CMP_ST)	243
15.5.5 DAC 控制寄存器 (DAC_CFG)	244
15.5.6 PGA0 控制寄存器 (PGA0_CFG)	245
15.5.7 PGA1 控制寄存器 (PGA1_CFG)	247
15.5.8 PGA 滤波控制寄存器 (PGA_FLT_CTRL)	248
15.5.9 PGA 状态寄存器 (PGA_ST)	249
15.5.10 CMP 管脚使能 (CMP_ASEL)	249
15.5.11 PGA0 管脚使能 (PGA0_ASEL)	250
15.5.12 PGA1 管脚使能 (PGA1_ASEL)	251
16 事件系统 (EBUS)	252
16.1 概述	252



16.2 主要功能	252
16.3 模块框图	252
16.4 功能描述	253
16.4.1 IO 检测通道	253
16.4.2 通道事件输入源	254
16.4.3 输出通道	255
16.4.4 通道事件输入源	255
16.4.5 软件事件	256
16.4.6 EBUS 中断号	256
16.5 寄存器描述	257
16.5.1 EBUS IO 检测通道设置寄存器 0 (EBUS_IO_DET0)	258
16.5.2 EBUS IO 检测通道设置寄存器 1 (EBUS_IO_DET1)	258
16.5.3 EBUS IO 检测通道设置寄存器 2 (EBUS_IO_DET2)	259
16.5.4 EBUS IO 检测通道设置寄存器 3 (EBUS_IO_DET3)	259
16.5.5 EBUS IO 检测通道设置寄存器 4 (EBUS_IO_DET4)	260
16.5.6 EBUS IO 检测通道设置寄存器 5 (EBUS_IO_DET5)	260
16.5.7 EBUS IO 检测通道设置寄存器 6 (EBUS_IO_DET6)	261
16.5.8 EBUS IO 检测通道设置寄存器 7 (EBUS_IO_DET7)	261
16.5.9 EBUS IO 检测通道设置寄存器 8 (EBUS_IO_DET8)	262
16.5.10 EBUS IO 检测通道设置寄存器 9 (EBUS_IO_DET9)	262
16.5.11 EBUS IO 检测通道设置寄存器 10 (EBUS_IO_DET10)	263
16.5.12 EBUS IO 检测通道设置寄存器 11 (EBUS_IO_DET11)	264
16.5.13 EBUS IO 检测通道设置寄存器 12 (EBUS_IO_DET12)	264
16.5.14 EBUS IO 检测通道设置寄存器 13 (EBUS_IO_DET13)	265
16.5.15 EBUS IO 检测通道设置寄存器 14 (EBUS_IO_DET14)	265
16.5.16 EBUS IO 检测通道设置寄存器 15 (EBUS_IO_DET15)	266
16.5.17 EBUS IO 检测中断状态寄存器 (EBUS_IO_STA)	266
16.5.18 EBUS 输出通道 0 设置寄存器 (EBUS_OUT0_CFG)	268
16.5.19 EBUS 输出通道 1 设置寄存器 (EBUS_OUT1_CFG)	269
16.5.20 EBUS 输出通道 2 设置寄存器 (EBUS_OUT2_CFG)	269



16.5.21 EBUS 输出通道 3 设置寄存器 (EBUS_OUT3_CFG)	270
16.5.22 EBUS 输出通道 4 设置寄存器 (EBUS_OUT4_CFG)	270
16.5.23 EBUS 输出通道 5 设置寄存器 (EBUS_OUT5_CFG)	271
17 键入中断 (IOW)	272
17.1 概述	272
17.2 主要功能	272
17.3 模块框图	272
17.4 功能描述	272
17.4.1 通道输入源	272
17.4.2 IOW 中断号	273
17.5 寄存器描述	273
17.5.1 键入中断中断使能寄存器 (IOW_INTEN)	274
17.5.2 键入中断控制寄存器 (IOW_CTRL)	274
17.5.3 键入中断边沿选择寄存器 (IOW_EDGES)	275
17.5.4 键入中断状态寄存器 (IOW_STA)	276
17.5.5 键入中断输入选择寄存器 (IOW_INS)	277
18 低电压检测 (LVD)	279
18.1 概述	279
18.2 主要功能	279
18.3 模块框图	280
18.4 功能描述	280
18.4.1 LVD 档位	280
18.4.2 LVD 功能	281
18.5 寄存器描述	281
18.5.1 电压检测模块控制寄存器 (LVD_CTRL)	282
18.5.2 电压检测模块状态寄存器 (LVD_ST)	283
19 嵌套中断向量控制器 (NVIC)	284
20 附录	286
20.1 寄存器相关缩写词列表	286

1 简介

本产品采用高性能 ARM Cortex-M0+ 32 位处理器，最高频率达到 96MHz，内嵌存储容量 64KB 高速 Flash 存储器和 4KB SRAM* 存储器。芯片集成了丰富的 I/O 端口及多种功能外设。集成 1 个硬件加速模块（含 32 位除法器 and 常用三角函数）、1 个采样率高达 1.26MSPS 的 12 位 ADC、1 个高精度 8MHz 高频 RC 时钟、1 个 32kHz 低频 RC 时钟、一个 96MHz 高速 PLL 时钟、2 个 DAC 和 CMP、2 个 PGA、1 个 16 位高级定时器和 3 个 16 位简单定时器。此外还包含多个标准通信接口：3 个 SCI（支持 UART/SPI/简易 I²C）。

本产品供电电压为 2.4V ~ 5.5V，工作温度范围为 -40°C 至 +105°C，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

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 1 个 Master 和 4 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 1 个 Master
 - Cortex-M0+
- 支持 4 个 Slave
 - Flash 存储器
 - SRAM 存储器
 - AHB0, AHB to APB 总线，包含部分 APB 接口外设
 - AHB2, 包含所有的 AHB 接口外设
- 支持地址重映射

2.3 模块框图

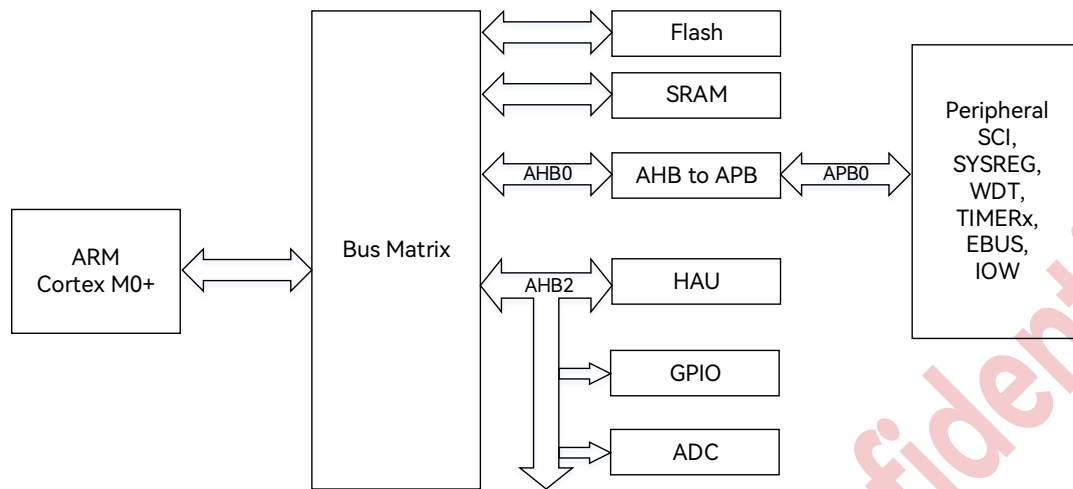


图 1 系统框图

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

地址空间	大小	从属分配
程序空间		
0x0000_0000-0x0000_07FF	2KB	程序 Flash/BootROM ⁽¹⁾
0x0000_0800-0x0000_FFFF	62KB	程序 Flash/保留 ⁽²⁾
0x0800_0000-0x0800_FFFF	64KB	程序 Flash ⁽³⁾
0x0801_0000-0x080F_FFFF	960KB	保留
0x0810_0000-0x0FFF_FFFF	127MB	保留
0x1000_0000-0x1FEF_FFFF	255MB	保留
0x1FF0_0000-0x1FFE_FFFF	960KB	保留
0x1FFF_0000-0x1FFF_05FF	1.5KB	BootROM ⁽⁴⁾
0x1FFF_0600-0x1FFF_07FF	0.5KB	选项字节
0x1FFF_0800-0x1FFF_FFFF	62KB	保留
SRAM		
0x2000_0000-0x2000_0FFF	4KB	SRAM
0x2000_1000-0x200F_FFFF	1020KB	保留
0x2010_0000-0x2FFF_FFFF	255MB	保留
0x3000_0000-0x3FFF_FFFF	256MB	保留

地址空间	大小	从属分配
片上外设		
0x4000_0000-0x4000_FFFF	64KB	AHB-to-APB 空间
0x4001_0000-0x4001_FFFF	64KB	保留
0x4002_0000-0x4002_0FFF	4KB	ADC
0x4002_1000-0x4002_FFFF	60KB	保留
0x4003_0000-0x4003_0FFF	4KB	保留
0x4003_1000-0x4003_1FFF	4KB	HAU
0x4003_2000-0x4003_FFFF	56KB	保留
0x4004_0000-0x400F_FFFF	768KB	保留
0x4010_0000-0x4FFF_FFFF	255MB	保留
0x5000_0000-0x5000_0FFF	4KB	GPIOA
0x5000_1000-0x5000_1FFF	4KB	GPIOB
0x5000_2000-0x5000_2FFF	4KB	GPIOC
0x5000_3000-0x5000_3FFF	4KB	保留
0x5000_4000-0x5000_4FFF	4KB	保留
0x5000_4000-0x5000_5FFF	4KB	保留
0x5000_6000-0x5000_FFFF	40KB	保留
0x5001_0000-0x500F_FFFF	960KB	保留
0x5010_0000-0x5FFF_FFFF	255MB	保留
其余空间		
0x6000_0000-0xFFFF_FFFF	2560MB	保留 ⁽⁵⁾

- (1) 该地址空间是 CPU 启动地址空间，由 MEM_MODE_CTRL 寄存器配置此地址空间是映射到程序 Flash 空间还是 BootROM 空间。
- (2) 该地址空间是 CPU 启动地址空间，当 MEM_MODE_CTRL 为 0 时，映射到程序 Flash 空间。
- (3) 该地址空间是程序 Flash 的物理地址空间。
- (4) 该地址空间是 BootROM 的物理地址空间。
- (5) 该地址空间是包含片外存储 & 外设拓展空间和 CPU 核专用空间。片外存储 & 外设拓展空间在本芯片为保留，CPU 核专用空间相关信息参照 ARM Cortex-M0+ 相关技术文档。

外部数据空间分配如下。

表 2 AHB-to-APB 地址空间

地址空间	大小	从属分配
0x4000_0000-0x4000_03FF	1KB	Flash 控制器
0x4000_0400-0x4000_07FF	1KB	TIM0
0x4000_0800-0x4000_0BFF	1KB	TIM1
0x4000_0C00-0x4000_0FFF	1KB	TIM2
0x4000_1000-0x4000_03FF	1KB	TIM8
0x4000_1400-0x4000_17FF	1KB	TIM_COM
0x4000_1800-0x4000_1BFF	1KB	保留
0x4000_1C00-0x4000_1FFF	1KB	IOW
0x4000_2000-0x4000_23FF	1KB	EBUS
0x4000_2400-0x4000_27FF	1KB	保留
0x4000_2800-0x4000_2BFF	1KB	SCI0
0x4000_2C00-0x4000_2FFF	1KB	WDT
0x4000_3000-0x4000_33FF	1KB	CMP
0x4000_3400-0x4000_37FF	1KB	SCI1
0x4000_3800-0x4000_3BFF	1KB	SCI2
0x4000_3C00-0x4000_3FFF	1KB	保留
0x4000_4000-0x4000_47FF	2KB	System Reg
0x4000_4800-0x4000_53FF	3KB	保留
0x4000_5400-0x4000_57FF	1KB	保留
0x4000_5800-0x4000_5FFF	2KB	保留
0x4000_6000-0x4000_63FF	1KB	保留
0x4000_6400-0x4000_73FF	4KB	保留
0x4000_7400-0x4000_77FF	1KB	保留
0x4000_7800-0x4000_FFFF	34KB	保留

3 Flash 存储器 (FLASH)

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。

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 64KB，分为 64 个页 (Sector)，每页容量为 1KB (字节)
 - 信息存储区 (Info Flash)：总容量 2KB，分为 2 个页，每页容量为 1KB
- 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
- 支持安全模式，保护代码内容
- 支持 Cache 功能

3.3 模块框图

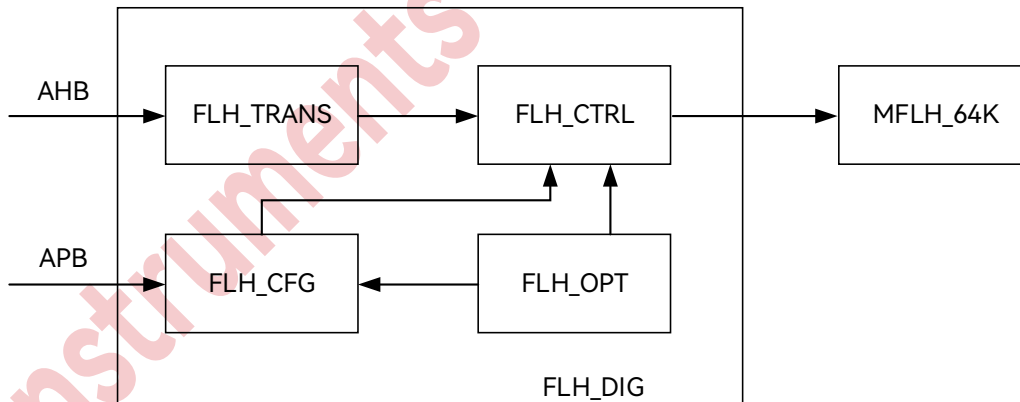


图 2 Flash 模块框图

FLH_TRANS 模块：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，各存储空间的地址区间如下表所示。

表 3 Flash 存储空间

Flash	Flash 地址区间	容量 (字节)	序号	描述
程序 Flash (主存储区)	0x0800_0000–0x0800_03FF	1KB	C_sector0	code
	0x0800_0400–0x0800_07FF	1KB	C_sector1	
	...	...	...	
	0x0800_FC00–0x0800_FFFF	1KB	C_sector63	
程序 Flash (信息存储区)	0x1FFF_0000–0x1FFF_03FF	1KB	C_info_sector0	bootrom
	0x1FFF_0400–0x1FFF_07FF	1KB	C_info_sector1	Bootrom/ chip_option

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，CPU 频率与读取延迟的配置如下。

表 4 Flash 读延迟设置

CPU 时钟	读取延迟
CPU_CLK ≤ 24MHz	1 时钟延迟
CPU_CLK=48MHz	3 时钟延迟
CPU_CLK=96MHz	5 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 Cache 功能

Flash 支持 Cache 功能以提高 CPU 的执行速度，当此功能打开后，Flash 读取时会先跟 Cache 中的缓存数据做比较，如果地址匹配则使用 Cache 的数据，如果不匹配会到 Flash 中进行读取。

注：Flash 此功能只有在 Flash 读延迟大于 0 时才会有效果。

3.4.4 Flash 中断号

表 6 Flash 中断

名称	中断号
FLH_INT	3

3.5 寄存器描述

Flash 寄存器的基地址为 0x4000_0000，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	0x00000000
0x004	Flash 数据寄存器	32	r/w	0x00000000
0x008	Flash 状态寄存器	32	r/w	0x00000000
0x010	Flash 控制寄存器	32	r/w	0x00000000
0x014	Flash 命令寄存器	32	r/w	0x00000000
0x018	Flash 密钥寄存器	32	wo	0x00000000
0x01C	Flash 页保护寄存器	32	ro	0x00000000

3.5.1 Flash 地址寄存器 (FLH_ADR)

偏移地址: 0x0000

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: 置为该页的首地址 烧写: Byte 模式: 地址应为 Byte 对齐 Halfword 模式: 地址应为 Half-Word 对齐 Word 模式: 地址应为 Word 对齐 复位值: 0x0

3.5.2 Flash 数据寄存器 (FLASH_DATA)

偏移地址: 0x0004

表 9 Flash 数据寄存器

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

3.5.3 Flash 状态寄存器 (FLASH_ST)

偏移地址: 0x0008

表 10 Flash 状态寄存器

位	访问	描述
31:27	res	Reserved 复位值: 0x0
28	r/w	FLH_ADR_ERR: Flash 地址选择错误状态 0: 地址选择无错误 1: 地址选择错误 注: 软件写 1 清 0。 复位值: 0x0
27:25	res	Reserved 复位值: 0x0
24	r/w	VEFY_PROG_ERR: 烧写预校验错误状态 0: 校验无错误 1: 校验错误 在烧写前读取 Flash 地址对应的数据, 如果数据不为全 1, 此状态会置位 注: 软件写 1 清 0。 复位值: 0x0
23:21	res	Reserved 复位值: 0x0

位	访问	描述
20	r/w	PROT_ERASE_ERR: 被保护页擦除错误 0: 擦除无错误 1: 擦除错误 当擦除被保护页时, 此状态会置位, 该页受保护, 不会被擦除 注: 软件写 1 清 0。 复位值: 0x0
19:17	res	Reserved 复位值: 0x0
16	r/w	PROT_PROG_ERR: 被保护页烧写错误 0: 被保护页烧写无错误 1: 被保护页烧写错误 当烧写被保护页时, 此状态会置位, 该页受保护, 不会被烧写 注: 软件写 1 清 0。 复位值: 0x0
15:13	res	Reserved 复位值: 0x0
12	r/w	MASS_FINISH: 全擦命令完成标志 0: 全擦命令未完成或未执行 1: 全擦命令已完成 注: 软件写 1 清 0。 复位值: 0x0
11:9	res	Reserved 复位值: 0x0
8	r/w	ERASE_FINISH: 页擦除完成标志 0: 页擦除未完成或未执行 1: 页擦除已完成 注: 软件写 1 清 0。 复位值: 0x0
7:5	res	Reserved 复位值: 0x0
4	r/w	PROG_FINISH: 烧写完成标志 0: 烧写未完成或未执行 1: 烧写已完成 注: 软件写 1 清 0。 复位值: 0x0
3:1	res	Reserved 复位值: 0x0
0	r	IDLE : Flash 空闲状态 0: 空闲 1: 忙 复位值: 0x0

3.5.4 Flash 控制寄存器 (FLH_CTRL)

偏移地址: 0x0010

表 11 Flash 控制寄存器

位	访问	描述
31:28	res	Reserved 复位值: 0x0
27:26	r/w	FLH_RDCK_CNT: Flash 读校验计数 (Flash Read Check Counter) 0: 校验 4 次 1: 校验 8 次 2: 校验 16 次 3: 校验 32 次 复位值: 0x0
25	r/w	FLH_RDCK_MOD: Flash 读校验模式 (Flash Read Check Mode) 0: 校验码放在 SRAM (0x2000_0800-0x2000_0FFF) 1: 校验码放在 FLASH (0x0800_7000-0x0800_7FFF) 注: 当此功能使能后, 在读取 Flash 时会同时读取校验 复位值: 0x0
24	r/w	FLH_RDCK_EN: Flash 读校验使能 (Flash Read Check Enable) 0: 关闭 1: 使能 复位值: 0x0
23:21	res	Reserved 复位值: 0x0
20	r/w	FLH_CACHE_EN: Flash Cache 功能使能 0: 关闭 1: 使能 复位值: 0x0
19:13	res	Reserved 复位值: 0x0
12	r/w	FLH_ERROR_INTEN: Flash 错误中断使能 (烧写/擦除/全擦) 0: 关闭 1: 使能 复位值: 0x0
11:9	res	Reserved 复位值: 0x0
8	r/w	FLH_FINISH_INTEN: Flash 结束中断使能 (烧写/擦除/全擦) 0: 关闭 1: 使能 复位值: 0x0
7:6	res	Reserved
5:4	r/w	PROG_MODE: Flash 烧写模式 0: Word 烧写/次 1: Halfword 烧写/次 2: Byte 烧写/次 3.保留 复位值: 0

位	访问	描述
3	res	Reserved
2:0	r/w	LATENCY: Flash 读取延迟 0: 延迟 0 个时钟 (系统时钟小于 24M) 1: 延迟 1 个时钟 (系统时钟为 24M) 2: 延迟 2 个时钟 3: 延迟 3 个时钟 (系统时钟为 48M) 4: 延迟 4 个时钟 5: 延迟 5 个时钟 (系统时钟为 96M) 6: 延迟 6 个时钟 (系统时钟>96M) 其他: 保留 复位值: 0x0

3.5.5 Flash 命令寄存器 (FLH_CMD)

偏移地址: 0x0014

表 12 Flash 命令寄存器

位	访问	描述
31:9	res	Reserved 复位值: 0x0
12	r/w	LOCK_EN: 寄存器锁使能 0: 未使能 1: 使能 注: 当软件使能此位时, Flash 寄存器不可被修改, 当通过 Flash 密钥寄存器进行解锁操作后, 此位会被清零, Flash 寄存器才可被修改。 复位值: 0x1
11:6	res	Reserved 复位值: 0x0
5:4	r/w	COMMAND: 命令 00: 烧写 01: 页擦除 10: 全擦除程序 Flash 11: Reserved 复位值: 0x0
3:1	res	Reserved 复位值: 0x0
0	r/w	START: 命令 (烧写/页擦除/全擦) 启动 0: 误操作 1: 触发启动 注: 软件写 1, 硬件会自动清 0。 复位值: 0x0

3.5.6 Flash 密钥寄存器 (FLASH_KEY)

偏移地址: 0x0018

表 13 Flash 密钥寄存器

位	访问	描述
31:0	wo	FLH_UNLOCK_KEY: Flash 寄存器密钥寄存器 当对此寄存器写值 0x53414645 时, Flash 寄存器会被解锁

3.5.7 Flash 页保护寄存器 (FLH_PROT)

偏移地址: 0x001C

表 14 Flash 页保护寄存器

位	访问	描述
31:0	r	SECTOR_PROT[31:0]: 页保护状态 SECTOR_PROT[0]: C_sector0~1 页保护状态, 此位为 1 表示受保护, 该页不可以被烧写和擦除, 否则会产生页保护错误状态在 Flash 状态寄存器中 SECTOR_PROT[1]: C_sector2~3 页保护状态 以此类推 SECTOR_PROT[31]: C_sector62~63 页保护状态 注: SECTOR_PROT 的值由硬件根据程序 Flash 区地址 0x400 的值取反得出, 如果此页被保护, 只能通过全擦程序 Flash 区并且在芯片下一次上电后才可被修改, 当修改了地址 0x400 的值后也需要芯片重新上电页保护才能生效。

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，内部 32kHz RC 振荡器和锁相环 96MHz 时钟可供选择使用。

4.2 主要功能

支持多种时钟源以用作系统时钟 (FCLK)

- 8MHz 高速内部 RC (HIRC)
- 96MHz 内部高速时钟 (PLL)
- 32kHz 低速内部 RC (LIRC)，可用于驱动看门狗模块从深度睡眠或停止模式下自动唤醒

以上提及的每个时钟源均可独立地开启或者关闭，一旦不使用的時候可关闭以节省功耗。然而这种情况除外，当 PLL 启用时，因其参考时钟为 HIRC，所以此时不能关闭 HIRC 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 96MHz。

大多数 IP 时钟在初次启动之后都是处于被控状态，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 对于 AHB，或者 PCLK 对于 APB)，然而以下例外。

- 选项字节 (Option Byte) 加载模块的时钟总是源于 HIRC 时钟
- LVD 模块的时钟源于 HIRC 时钟
- PMU & HIRC & LIRC 配置寄存器的时钟 AOS_CLK 总是源于 HIRC 时钟
- 看门狗时钟支持以下时钟源
 - LIRC 时钟
 - HIRC 时钟
- IO 噪声滤波时钟支持以下时钟源
 - 系统时钟
 - LIRC 时钟
 - HIRC 时钟
- ADC 控制逻辑部分的时钟支持以下时钟源
 - 系统时钟

- 蜂鸣器时钟支持多个时钟源

- LIRC 时钟
- HIRC 时钟
- 系统时钟

4.3 模块框图

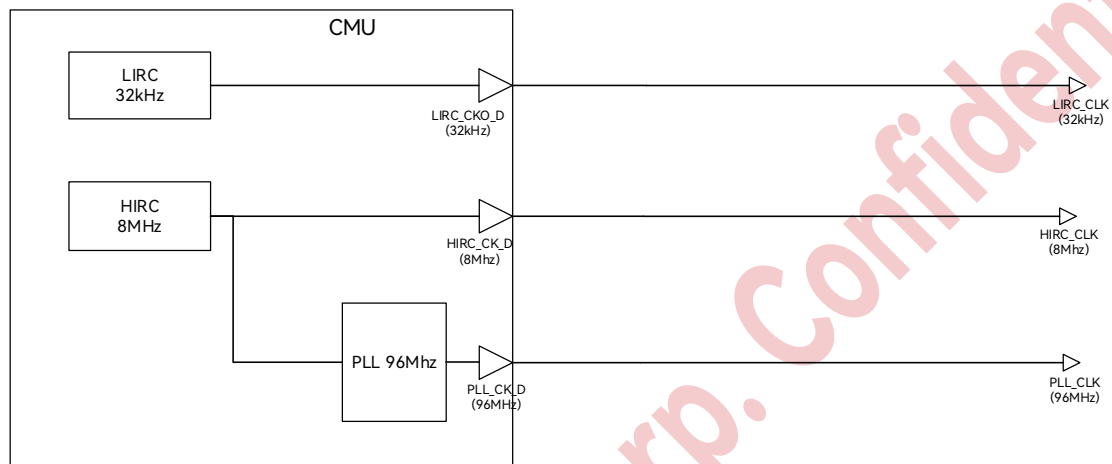


图 3 芯片时钟源示意图

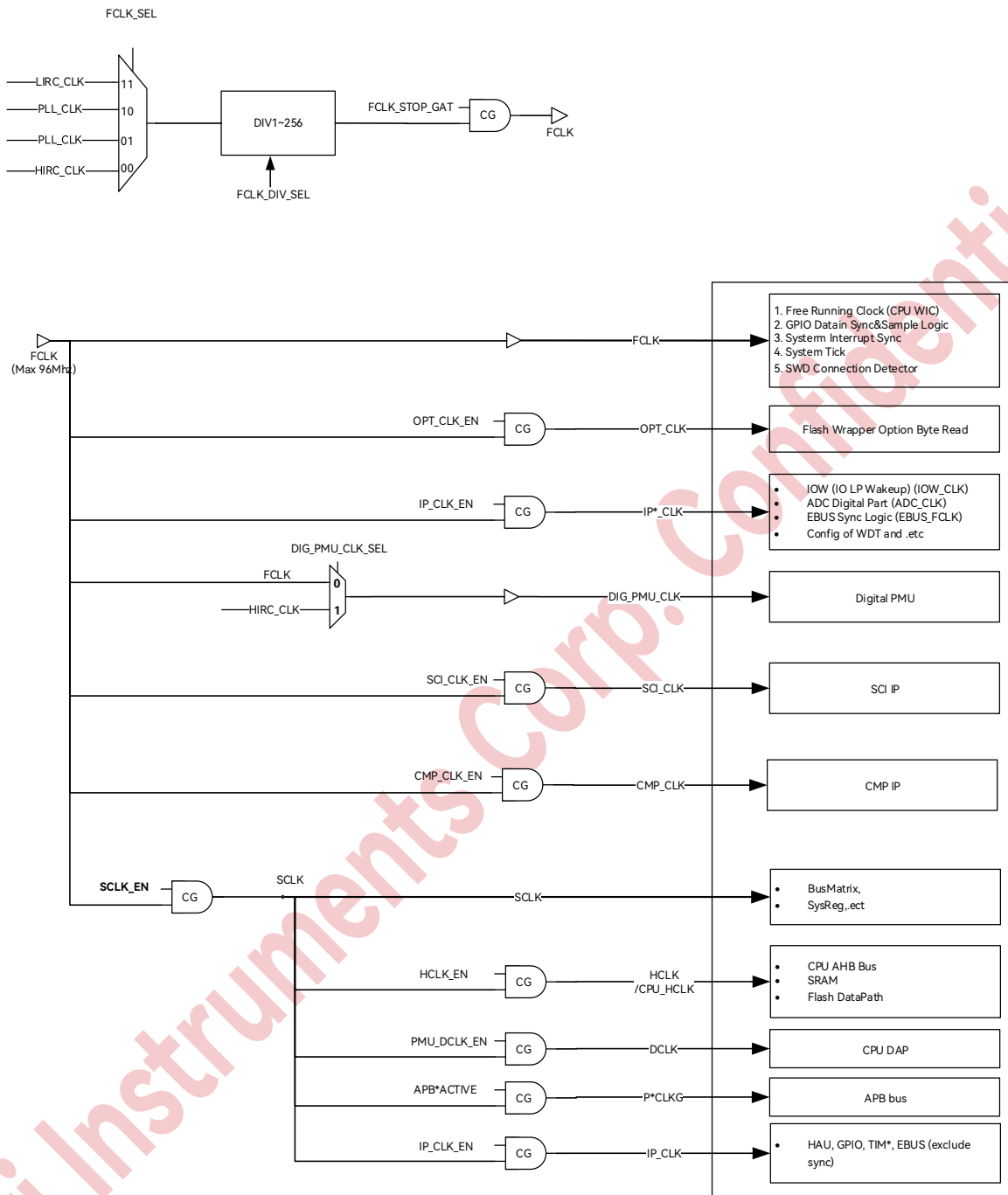


图 4 系统时钟组 (含同步 IP) 架构图

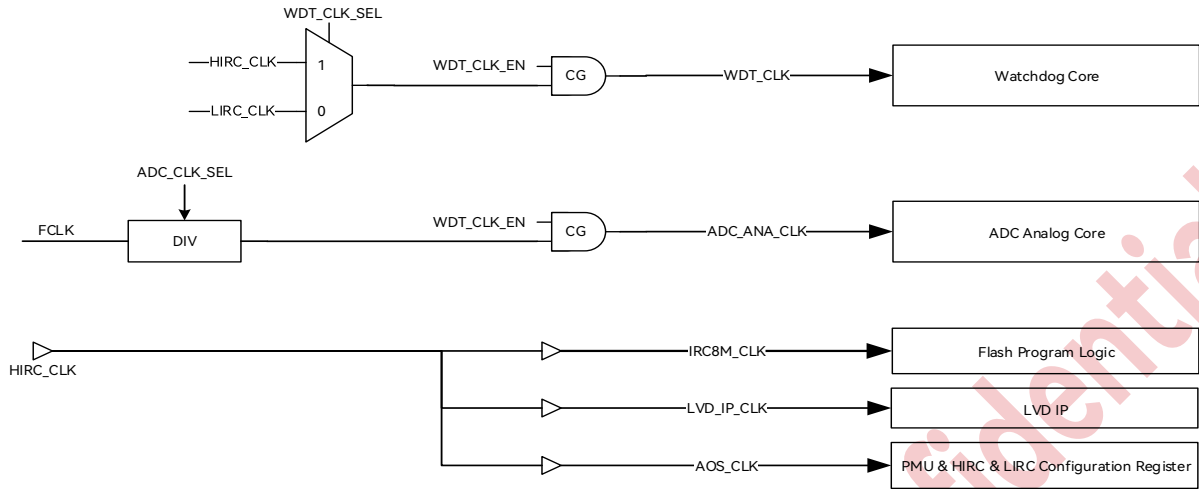


图 5 与系统时钟组异步时钟架构图

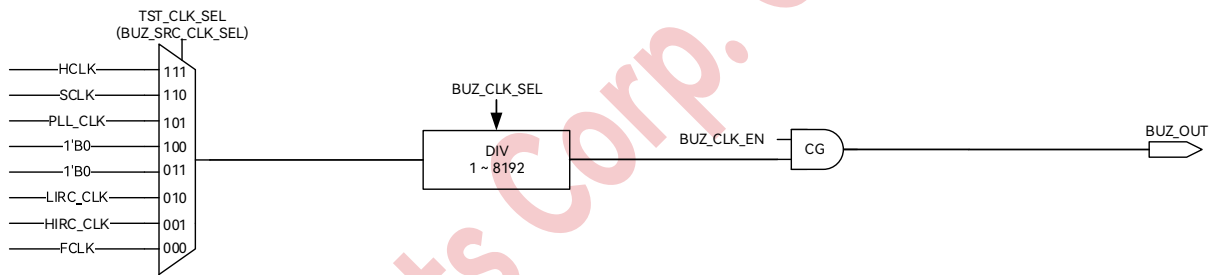


图 6 蜂鸣器时钟架构图

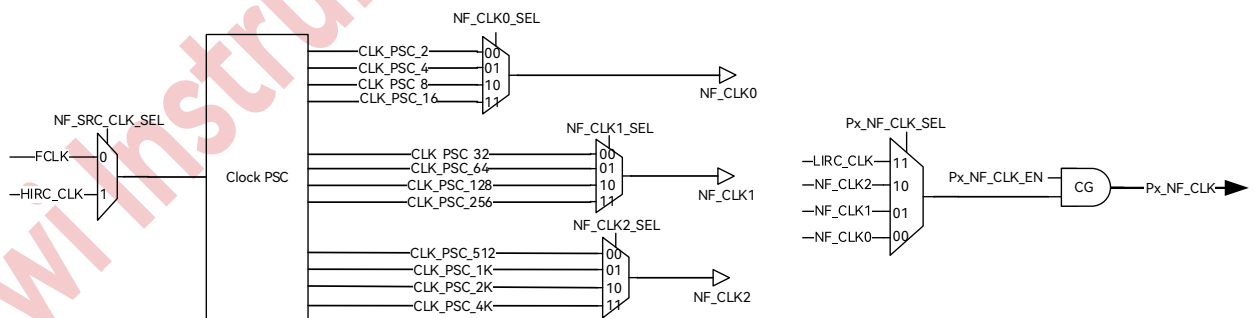


图 7 IO 噪声滤波时钟架构图

4.4 功能描述

4.4.1 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 参考时钟输入。

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）。其启动时间也比外部晶体振荡器更快。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。该模块支持出厂校正功能，常温下精度可校正至 $\pm 0.5\%$ 以内，在 $-40^{\circ}\text{C} \sim 105^{\circ}\text{C}$ 温度范围内精度 $\pm 1\%$ 以内。芯片复位后，HIRC 所需要的工厂校准值从系统选项字节（Option Byte）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_REG0）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。

4.4.2 PLL 时钟

内部 PLL 是用于对于 HIRC 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。

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

4.4.3 LIRC 时钟

LIRC 作为一个低功耗时钟源，支持看门狗和基准定时器在停止模式下运行。时钟频率在 32kHz 左右，该模块也具有出厂校正功能，常温下精度可校正至 $\pm 3\%$ 以内，在 $-40^{\circ}\text{C} \sim 105^{\circ}\text{C}$ 温度范围内精度 $\pm 10\%$ 以内。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。

4.4.4 系统时钟组

CM0+ 集成时钟

系统时钟组提供 CPU 及其外围组件的时钟，总共四个时钟，包括 FCLK、HCLK、SCLK 和 DCLK，并且这些

时钟工作于同样的频率。

FCLK 在 CM0+所有模式下（普通模式、睡眠模式、深度睡眠模式）一直运行。

SCLK 必须直接源于 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 必须直接由 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。

DCLK 必须直接由 FCLK 派生，在连接调试器时，必须始终驱动 DCLK。然而没有调试器连接时，它是处于门控状态。

- FCLK 驱动 WIC 模块
- HCLK 驱动 CPU
- SCLK 驱动 NVIC，总线矩阵和系统寄存器
- DCLK 驱动 CM0+调试域

PMU 管理单元生成和管理以上时钟。在连接调试器时，必须始终驱动 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，请参考功耗模式文档。

可使用以下各种时钟源用于驱动系统时钟组。

- HIRC 时钟
- PLL 时钟
- LIRC 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。

系统时钟组中的 IP 时钟

HAU、GPIO、TIM 和 EBUS（同步逻辑除外）的时钟信号源自于 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。

IOW、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。

看门狗 IP 的控制寄存器域的时钟信号源自于 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。

SCI 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。

4.4.5 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 时钟由专属的时钟生成器生成。

选项字节加载模块时钟

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

LVD 模块时钟

LVD 模块的时钟源于 HIRC 时钟。

AOS_CLK

PMU & HIRC & LIRC 配置寄存器的时钟 AOS_CLK 总是源于 HIRC 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 时钟，默认配置运行于 FCLK 的分频时钟，具体分频基于软件配置。

WDT 内核时钟

WDT 内核时钟源于 LIRC 和 HIRC，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之一：

- FCLK
- HIRC
- LIRC
- PLL
- SCLK
- HCLK

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

4.4.6 低功耗

关于低功耗模式下的时钟管理细节在功耗模式章节详细描述。

5 复位 (RESET)

5.1 概述

KPM32K07XX 包含两类复位源：系统复位和 POR 复位，如下表所列。

表 15 复位源

复位源	描述
POR 复位	外部 RC 复位 (PAD_RSTJ) 上电复位 (POR_RSTJ) VCC 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，KPM32K07XX 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

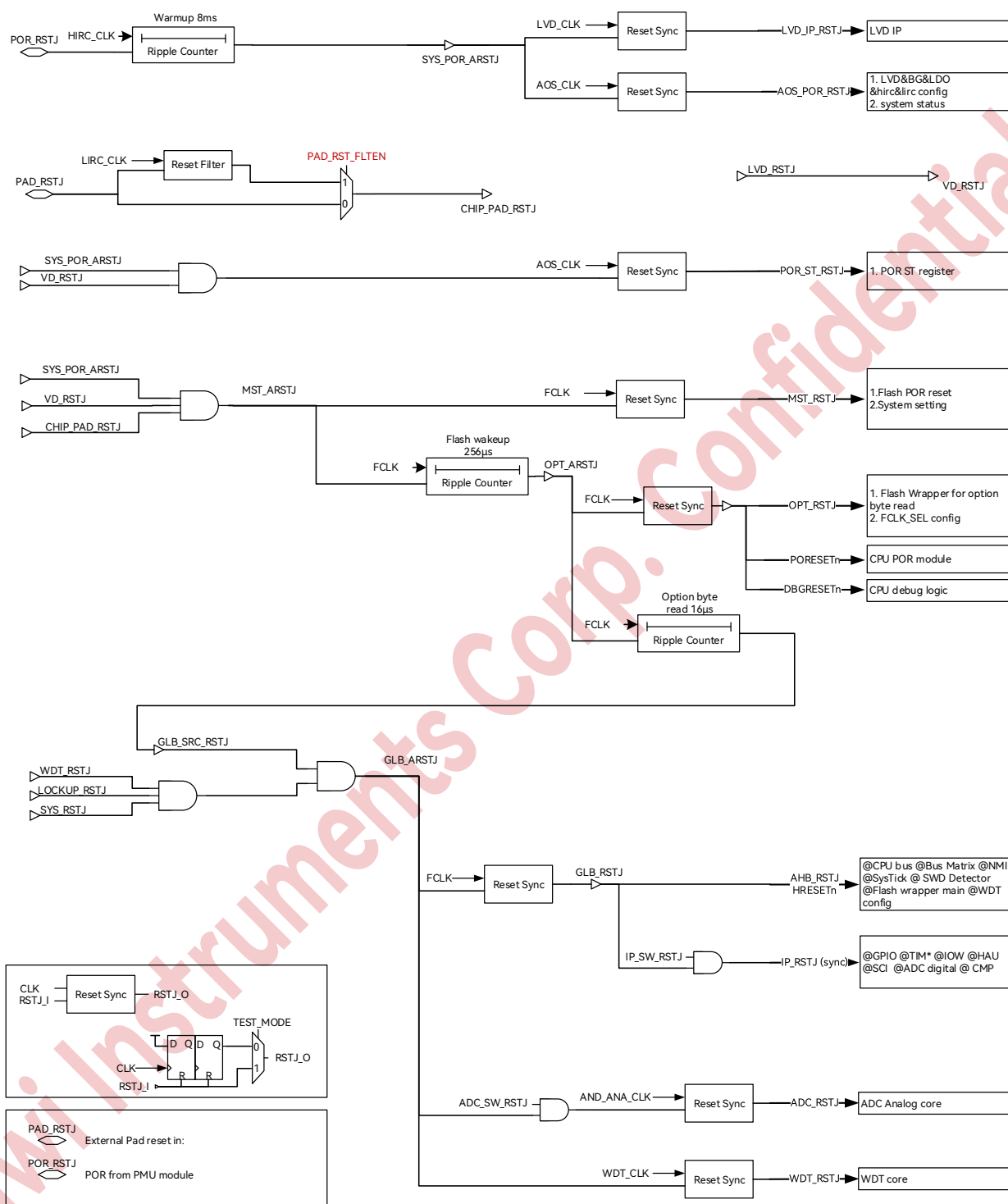


图 8 芯片复位架构图

复位架构图中的关键复位信号释义如下。

- **POR_RSTJ**: 由芯片 PMU 模块产生, 当检测到 VCC 成功上电, 该信号由低转高, 同时依据 V_{POR} & V_{PDR} 的值复位整个芯片。
- **PAD_RSTJ**: 外部复位信号 RSTJ 经 IO 输入所得。
- **LVD_RSTJ**: 电压检测模块复位请求信号。
- **WDT_RSTJ**: 看门狗复位请求信号。
- **LOCKUP_RSTJ**: LOCKUP 复位请求信号。
- **SYS_RSTJ**: 系统软件复位请求信号。
- **SYS_POR_ARSTJ**: 系统上电复位信号。
- **MST_ARSTJ**: 系统主复位信号, 也用于触发 Flash 预热操作。
- **OPT_ARSTJ**: 用于触发加载选择字节复位信号。
- **GLB_RSTJ**: 系统全局复位信号。

上述关键复位信号的时序关系如下图所示, 图中的参数 (V_{LVD} 、 V_{POR} 、 V_{PDR} 、 V_{IH} 和 V_{IL}) 值详见数据手册。

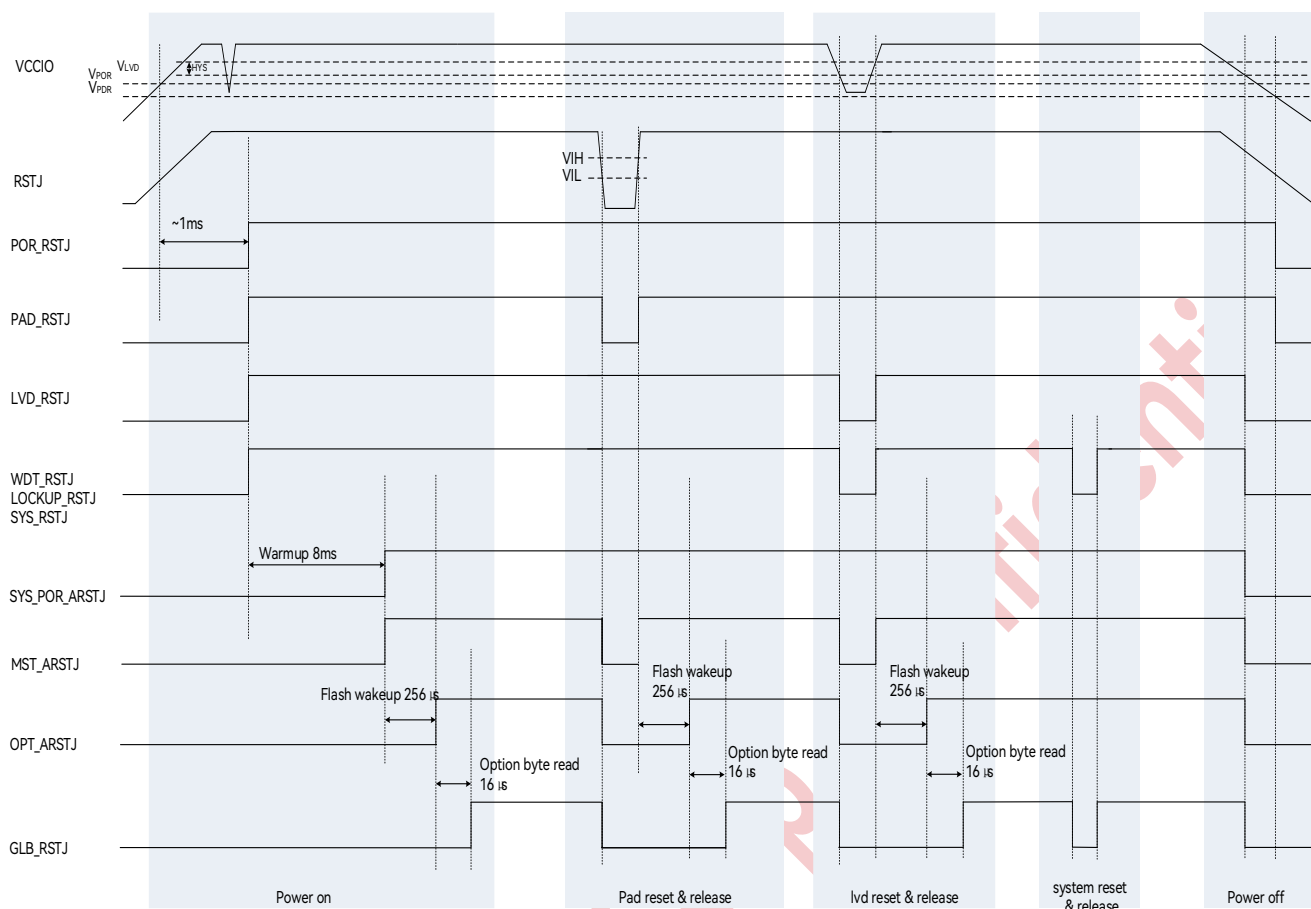


图 9 复位时序图

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 VCC 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)

KPM32K07XX 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 电路包含电压检测模块 LVD 用于 VCC 检测，默认开启。当 VCC 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和 Cortex-M0+ 中的 DAP 逻辑。

5.3.5 软件复位 (SYS)

对 NVIC 应用中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-M0+ 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 0x4000_4010[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 16 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

注：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

KPM32K07XX 为大多数数字 IP 提供了软件复位控制位 (0x4000_4280 & 0x4000_4284)，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 功耗模式 (POWER_MODE)

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

KPM32K07XX 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 M0+ 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

KPM32K07XX 内嵌一个电压调节器，用于向数字电源域提供 1.5V 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
 - VCC 为 5V
- 内核、存储、数字外设的电源供应
 - VDD 为 1.5V

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 (V_{POR})。

对于 POR 而言，当监测到电源电压低于指定的阈值 (V_{POR}) 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 VDD 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。

表 17 不同模式下的模块状态

模块	时钟	工作模式			
		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⁾	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
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
HAU	HAU_CLK	OPT	OPT	GT	GT
TIM	TIM_CLK	OPT	OPT	GT	GT
SCI	SCI_CLK	OPT	OPT	OPT	GT
CMP	CMP_CLK	OPT	OPT	OPT	GT
EBUS	EBUS_CLK	OPT	OPT	GT	GT
模拟模块					
PMU	-	-	-	-	-
ADC	ADC_CLK	OPT	OPT	OPT	GT
LVD	LVD_CLK	ON	ON	ON	GT

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

下图说明了 CPU 时钟架构。

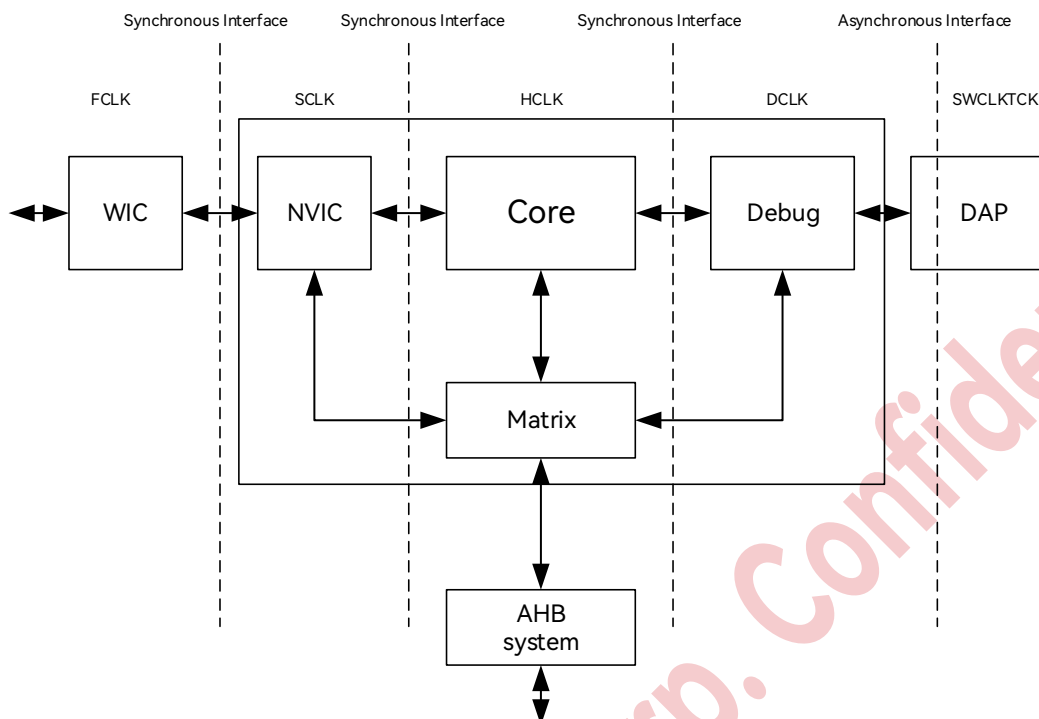


图 10 Cortex-M0+ 时钟域

6.3.4 低功耗模式

KPM32K07XX 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。

KPM32K07XX 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®-M0+ 核心外设如 NVIC, SysTick 仍然保持运行
- 深度睡眠模式：除了 WDT 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®-M0+ 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)
 - SLEEPDEEP=0
 - SLEEPONEXIT=0
 - 没有中断 (针对 WFI) 或事件 (针对 WFE) 未决
- 退出时睡眠 (Sleep-on-exit)
 - SLEEPDEEP=0
 - SLEEPONEXIT=1
 - 没有中断未决
 - 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®-M0+ 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 18 不同模式下的模块状态

睡眠模式	描述
模式进入	WFI (Wait for Interrupt) 或 WFE (wait for event) 的同时： ■ SLEEPDEEP=0 位于 Cortex®-M0+ 系统控制寄存器 (SCR) 位 2 ■ 无中断（针对 WFI）或事件（针对 WFE）未决 参考 Cortex®-M0+ 系统寄存器。
	处理完 ISR 的同时： ■ SLEEPDEEP=0 位于 Cortex®-M0+ 系统控制寄存器 (SCR) 位 2 ■ SLEEPONEXIT=1 位于 Cortex®-M0+ 系统控制寄存器 (SCR) 位 1 ■ 无中断未决 参考 Cortex®-M0+ 系统寄存器。
模式退出	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》）。
唤醒延迟	无

深度睡眠模式

深度睡眠模式是基于 Cortex®-M0+ 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
- 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。

- 设置那些不与 FCLK 同步的 IP 时钟进入门控或者释放状态
- 设置 FCLK 时钟源（切换至低速时钟或者其他）

- 设置 PMUENABLE=1
- 设置 SLEEPDEEP=1
- 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默认关闭。看门狗的配置寄存器可以修改，但首先得输入正确的密钥。在进入低功耗模式之前用户可视应用场景将看门狗关闭。

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) 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

KPM32K07XX 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-M0+ 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 19 深度睡眠模式

深度睡眠模式	描述
模式进入	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®-M0+ 系统控制寄存器 (SCR) 位 2 ■ PMUENABLE bit=1 位于功耗模式寄存器 (PWR_MOD) 位 0
	处理完 ISR 的同时： ■ 无中断未决 ■ SLEEPDEEP=1 位于 Cortex®-M0+ 系统控制寄存器 (SCR) 位 2 ■ SLEEPONEXIT=1 位于 Cortex®-M0+ 系统控制寄存器 (SCR) 位 1 ■ PMUENABLE bit=1 位于功耗模式寄存器 (PWR_MOD) 位 0

深度睡眠模式	描述
模式退出	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/IOW/SCI/ADC/CMP) 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/IOW/SCI/ADC/CMP) 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》)
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，HIRC 分频时钟均由软件关电，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
- 系统可被具备唤醒能力并配置为中断模式的外设 (WDT/IOW) 唤醒。对于 WDT 的时钟源不能配置为 HIRC 及相关时钟，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 HIRC；
2. 设置那些源于 LIRC 或 IP 时钟状态 (门控或释放)，例如 WDT；
3. 如果有需要，使能 eFlash 深度待机控制位 (0x4000_4070[4]=1'b1)；
4. 设置 STOP_MODE 和 PMUENABLE (0x4000_4070[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默认关闭。看门狗的配置寄存器可以修改，但首先得输入正确的密钥
- 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 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 20 停止模式

停止模式	描述
模式进入	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 ■ 设置 FCLK 源于 HIRC ■ SLEEPDEEP=1 位于 Cortex®-M0+ 系统控制寄存器 (SCR) 位 2 ■ STOP_MODE=1 & PMUENABLE=1 位于功耗模式控制寄存器 (PWR_MOD)
	处理完 ISR 的同时： ■ 无中断未决 ■ 设置 FCLK 源于 HIRC ■ SLEEPDEEP=1 位于 Cortex®-M0+ 系统控制寄存器 (SCR) 位 2 ■ SLEEPONEXIT=1 位于 Cortex®-M0+ 系统控制寄存器 (SCR) 位 1 ■ STOP_MODE=1 & PMUENABLE=1 位于功耗模式控制寄存器 (PWR_MOD)
模式退出	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/ 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/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》)
唤醒延迟	内部 RC 振荡器唤醒时间 + 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®-M0+ 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
- 或者配置一个 EBUS 通道 (参照 EBUS 规格书) 至 CPU RXEV (一个输入高电平导致 Cortex-M0+ 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)

7 系统寄存器 (SYSTEM_REG)

7.1 概述

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

7.2 寄存器描述

系统寄存器的基地址是 0x4000_4000

表 21 系统寄存器映射表

偏移地址	寄存器列表	位宽	访问	复位值
0x0000 ~0x0004	Reserved	32	res	0x00000000
0x0008	芯片 ID 寄存器	32	ro	0x01200000
0x000C	Reserved	32	res	0x00000000
0x0010	复位状态寄存器	32	ro	0x03007001
0x0014 ~0x005C	Reserved	32	res	0x00000000
0x0060	系统控制寄存器	32	r/w	0x00000003
0x0064 ~0x006C	Reserved	32	res	0x00000000
0x0070	功耗模式控制寄存器	32	r/w	0x00000000
0x0074 ~0x00F8	Reserved	32	res	0x00000000
0x00FC	系统配置密钥寄存器	32	w	0x00000000
0x0100	模拟 PMU 控制寄存器	32	r/w	0x04200180
0x0104 ~0x010C	Reserved	32	res	0x00000000

偏移地址	寄存器列表	位宽	访问	复位值
0x0110	IRC 控制寄存器 0	32	r/w	0x00000000
0x0114	IRC 控制寄存器 1	32	r/w	0x00844210
0x0118	IRC 控制寄存器 2	32	r/w	0x00000001
0x011C ~0x014C	Reserved	32	res	0x00000000
0x0150	PLL 控制寄存器	32	r/w	0x36420020
0x0154 ~0x015C	Reserved	32	res	0x00000000
0x0160	温度传感器控制寄存器	32	r/w	0x00000000
0x0164 ~0x01FC	Reserved	32	res	0x00000000
0x0200	电压检测模块控制寄存器	32	r/w	0x00000411
0x0204	电压检测模块状态寄存器	32	r/w	0x00000000
0x0208 ~0x0274	Reserved	32	res	0x00000000
0x0278	系统复位控制寄存器	32	r/w	0x00000000
0x027C	Reserved	32	res	0x00000000
0x0280	复位控制寄存器 0	32	r/w	0x00000000
0x0284	复位控制寄存器 1	32	r/w	0x00000000
0x0288 ~0x028C	Reserved	32	res	0x00000000
0x0290	时钟门控寄存器 0	32	r/w	0x80000000
0x0294	时钟门控寄存器 1	32	r/w	0x00000000
0x0298 ~0x029C	Reserved	32	res	0x00000000
0x02A0	时钟选择寄存器 0	32	r/w	0x00000000
0x02A4	时钟选择寄存器 1	32	r/w	0x00000000
0x02A8	时钟选择寄存器 2	32	r/w	0x00000000
0x02AC ~0x02BC	Reserved	32	res	0x00000000
0x02C0	噪声滤波时钟控制寄存器	32	r/w	0x00000000
0x02C4 ~0x02FC	Reserved	32	res	0x00000000
0x0300	IO 噪声滤波控制寄存器 0	32	r/w	0xFFFFFFFF

偏移地址	寄存器列表	位宽	访问	复位值
0x0304	IO 噪声滤波控制寄存器 1	32	r/w	0x00000003
0x0308 ~0x031C	Reserved	32	res	0x00000000
0x0320	端口输入使能控制寄存器 0	32	r/w	0xFFFFFFFF
0x0324	端口输入使能控制寄存器 1	32	r/w	0x00000003
0x0328 ~0x033C	Reserved	32	res	0x00000000
0x0340	引脚驱动控制寄存器 0	32	r/w	0x00000000
0x0344	引脚驱动控制寄存器 1	32	r/w	0x00000000
0x0348 ~0x035C	Reserved	32	res	0x00000000
0x0360	引脚上拉控制寄存器 0	32	r/w	0x00000000
0x0364	引脚上拉控制寄存器 1	32	r/w	0x00000000
0x0368 ~0x037C	Reserved	32	res	0x00000000
0x0380	引脚下拉控制寄存器 0	32	r/w	0x00000040
0x0384	引脚下拉控制寄存器 1	32	r/w	0x00000000
0x0388 ~0x039C	Reserved	32	res	0x00000000
0x03A0	端口模式控制寄存器 0	32	r/w	0x00000000
0x03A4	端口模式控制寄存器 1	32	r/w	0x00000000
0x03A8	端口模式控制寄存器 2	32	r/w	0x00000000
0x03AC ~0x03DC	Reserved	32	res	0x00000000
0x03E0	功能复用选择寄存器 0	32	r/w	0x00000000
0x03E4	功能复用选择寄存器 1	32	r/w	0x00000000
0x03E8	功能复用选择寄存器 2	32	r/w	0x00000000
0x03EC	功能复用选择寄存器 3	32	r/w	0x00000000
0x03F0	功能复用选择寄存器 4	32	r/w	0x00000000
0x03F4 ~0x045C	Reserved	32	res	0x00000000
0x0460	端口开漏输出控制寄存器 0	32	r/w	0x00000000
0x0464	端口开漏输出控制寄存器 1	32	r/w	0x00000000

偏移地址	寄存器列表	位宽	访问	复位值
0x0468 ~0x04BC	Reserved	32	res	0x00000000
0x04C0	调试控制寄存器	32	r/w	0x00000000
0x04C4 ~0x07FC	Reserved	32	res	0x00000000

7.2.1 芯片 ID 寄存器 (CHIP_ID)

偏移地址: 0x0008

表 22 芯片 ID 寄存器

位	访问	描述
31:16	ro	CHIP_ID, 芯片身份 ID KPM32K07XX 则是 0x0120. 复位值: 0x0120
15:8	ro	Reserved 复位值: 0x0
7:0	ro	CHIP_VER, 芯片版本信息 复位值: 0x0

7.2.2 复位状态寄存器 (RST_ST)

偏移地址: 0x0010

表 23 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
27:26	ro	Reserved 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

位	访问	描述
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: 0x0
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: 0x0
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: 0x0
11:4	ro	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 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

7.2.3 系统控制寄存器 (SYS_CFG)

偏移地址: 0x0060

表 24 系统控制寄存器

位	访问	描述
31:9	res	Reserved 复位值: 0x0
8	r/w	DP_SW_CFG_EN, 软件配置调试口 pinmux 使能 0: 调试口默认开启 (硬件自动设定) 1: 通过软件控制系统相关寄存器实现调试口开启或切换到其他复用功能 注: 1. 在更新该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; 2. 在 DP_SW_CFG_EN 为 0 时, 相关调试口的端口功能复用控制寄存器 (IO_PU_CTRL_REG*, IO_PD_CTRL_REG*, IO_MODE_CTRL_REG*, AF_SEL_REG* 的具体控制位不起作用。 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: 保留 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]。

7.2.4 功耗模式控制寄存器 (PWR_MOD)

偏移地址: 0x0070

表 25 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, 停止模式使能控制 0: 关闭 1: 开启 复位值: 0x0
0	r/w	PMUENABLE, 数字 PMU 使能控制 0: 关闭 1: 开启 复位值: 0x0

7.2.5 系统配置密钥寄存器 (SYS_SET_KR)

偏移地址: 0x00FC

表 26 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 和 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 27 系统配置密钥寄存器 (读)

位	访问	描述
31:1	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

7.2.6 模拟 PMU 控制寄存器 (ANA_PMU_CTRL)

偏移地址: 0x0100

表 28 模拟 PMU 控制寄存器

位	访问	描述
31:27	res	Reserved 复位值: 0x0
26:24	r/w	VREF_TMT, BandGAP 温度 trimming 配置 注: 调整步进精度 2.5%, 范围为 -10% ~ 6.5%。 复位值: 0x4
23:22	res	Reserved 复位值: 0x0
21:16	r/w	VREF_TM, BandGAP trimming 配置 注: 调整步进精度 12mv, 范围为 +/-30%, 默认是 1.2V。 复位值: 0x20
15:11	res	Reserved 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: 200nA 1: 400nA PMU_IBSEL[1:0], LDO 电流偏置设置 00: 300nA 01: 500nA 10: 800nA 11: 1000nA 复位值: 0x1
7:4	r/w	LDO_SEL, LDO 输出电压控制 输出电压值: $1350 + \text{LDO_SEL} * 25$ (mv) 注: 默认情况下 1.550V。 复位值: 0x8
3:2	r/w	LDO_DSSEL, LDO 输出驱动能力选择 00: 110mA 01: 160mA 10: 220mA 11: 270mA 复位值: 0x0
1:0	res	Reserved 复位值: 0x0

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。

7.2.7 IRC 控制寄存器 0 (IRC_CTRL_REG0)

偏移地址: 0x0110

表 29 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 注: 当 FCLK 源选择源于 LIRC 时, 该位不能置 1 复位值: 0x0
15:1	res	Reserved 复位值: 0x0
0	r/w	HIRC_PD, HIRC 控制 0: 开启 1: 断电 注: 当 FCLK 源选择源于 HIRC 时, 该位不能置 1 复位值: 0x0

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。

7.2.8 IRC 控制寄存器 1 (IRC_CTRL_REG1)

偏移地址: 0x0114

表 30 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: 26.8μS ~ 39.3μS 调整步进: 0.8μS 频率: 25.5kHz ~ 37.4kHz 范围: -18.8% ~ 22.3% 复位值: 0x8 (31.25μs)
19	res	Reserved 复位值: 0x0
18:16	r/w	LIRC_TMC, LIRC 粗略调整 周期: 19.6μS ~ 48.4μS 步进: 4μS 频率: 20.6kHz ~ 51kHz 范围: -41% ~ 66.3% 复位值: 0x4 (32K)
15	res	Reserved 复位值: 0x0

位	访问	描述
14:10	r/w	HIRC_TMT, HIRC 温度调整 调整范围: -5.3% ~ +5.3% 步进: 0.33% 复位值: 0x10
9:5	r/w	HIRC_TMF, HIRC 精细调整 Fine trimming (T) 周期: 116.35nS ~ 130.55nS, 调整步进: 0.45nS, 0.36% 频率: 7.629511MHz ~ 8.594693MHz, -5.26% ~ 6.31% 复位值: 0x10 (125ns)
4:0	r/w	HIRC_TMC, HIRC 粗略调整 (coarse trimming) 频率: 4.00MHz ~ 11.78MHz, -79% ~ 62% 步进: 0.25MHz, 3.12%, 复位值: 0x10 (8MHz)

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。

7.2.9 IRC 控制寄存器 2 (IRC_CTRL_REG2)

偏移地址: 0x0118

表 31 IRC 控制寄存器 2

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 (仅在测试模式使用) 0: 关闭 1: 开启 复位值: 0x0
7:3	res	Reserved 复位值: 0x0
2	r/w	HIRC_CKOE, HIRC 时钟输出使能 (仅在测试模式使用) 0: 关闭 1: 开启 复位值: 0x0
1:0	r/w	HIRC_DIVT, HIRC 分频选择 (仅在测试模式使用) 00: 1MHz 01: 2MHz 10: 4MHz 11: 8MHz 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。

7.2.10 PLL 控制寄存器 (PLL_CTRL_REG)

偏移地址: 0x0150

表 32 PLL 控制寄存器

位	访问	描述
31	r/w	PLL_CKOE, PLL 时钟输出使能 (仅在测试模式使用) 0: 关闭 1: 开启 复位值: 0x0
30:28	r/w	PLL_LDO, PLL LDO 选择 3'b000: 1.50V 3'b001: 1.55V 3'b010: 1.60V 3'b011: 1.65V 3'b100: 1.70V 3'b101: 1.75V 3'b110: 1.80V 3'b111: 1.85V 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
25:24	r/w	PLL_KVCO, VCO 电压转电流增益调节控制 2'b00: 6.49μA/V 2'b01: 7.89μA/V 2'b10: 10.16μA/V 2'b11: 14.40μA/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 注: PLL 的 VCO 输入时钟频率 $F_{in}=F_{ref}/PLL_DIVM$, 其中 F_{ref} 是 PLL 参考时钟的频率 复位值: 0x4
19:18	res	Reserved 复位值: 0x0

位	访问	描述
17:12	r/w	PLL_MUTN, PLL 输入时钟倍数 N 配置选择 PLL 的 VCO 工作频率是输入时钟频率 F_{in} 乘以 N, $F_{vco}=F_{in}*N$ (其中 $N=64+PLL_MUTN$)。假设输入频率 $F_{in}=2MHz$, VCO 频率与 N 的关系如下: 000 00: 128MHz 000 01: 130MHz 000 10: 132MHz 100 00: 192MHz 111101: 250MHz 111110: 252MHz 111111: 254MHz 复位值: 0x20
11:9	res	Reserved 复位值: 0x0
9:4	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 192MHz, 000000: $192/2=96MHz$ 000001: $192/2=96MHz$ 000010: $192/2=96MHz$ 000011: $192/3=64MHz$ 111100: $192/60=3.2MHz$ 111101: $192/61=3.148MHz$ 111110: $192/62=3.097MHz$ 111111: $192/63=3.048MHz$ 复位值: 0x2
3:2	res	Reserved 复位值: 0x0
1	r/w	PLL_CKISEL, PLL 时钟输入源选择控制 0: HIRC 1: EXT_CLK 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

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。

7.2.11 温度传感器控制寄存器 (TS_CTRL_REG)

偏移地址: 0x0160

表 33 温度传感器控制寄存器

位	访问	描述
31:3	res	Reserved 复位值: 0x0
2:1	r/w	TS_SEL, 温度传感器输出电压选择控制 00: 0.93V 01: 0.86V 10: 0.79V 11: 0.72V 复位值: 0x1
0	r/w	TS_EN, 温度传感器控制使能 0: 关闭 1: 开启 复位值: 0x0

7.2.12 电压检测模块控制寄存器 (LVD_CTRL)

偏移地址: 0x0200

表 34 电压检测模块控制寄存器

位	访问	描述
31:12	res	Reserved 复位值: 0x0
11:8	r/w	LVD_SEL, 电压检测模块档位选择控制 0000: 保留 0001: 保留 0010: 保留 0011: 保留 0100: 2.4V±0.05V 0101: 2.6V±0.05V 0110: 2.8V±0.05V 0111: 3.0V±0.05V 1000: 3.2V±0.05V 1001: 3.4V±0.05V 1010: 3.6V±0.05V 1011: 3.8V±0.05V 1100: 4.0V±0.05V 1101: 4.2V±0.05V 1110: 4.4V±0.05V 1111: 4.5V±0.05V 注: 该位只能被 POR 复位或外部复位。 复位值: 0x4
7:5	res	Reserved 复位值: 0x0

位	访问	描述
4	res	LVD_RST_EN, 电压检测模块复位芯片使能控制 0: 关闭 1: 开启 注: 该位只能被 POR 复位或外部复位。 复位值: 0x1
3:1	res	Reserved 复位值: 0x0
0	r/w	LVD_EN, 电压检测模块使能控制 0: 关闭 1: 开启 注: 该位只能被 POR 复位或外部复位。 复位值: 0x1

7.2.13 电压检测模块状态寄存器 (LVD_ST)

偏移地址: 0x0204

表 35 电压检测模块状态寄存器

位	访问	描述
31:16	res	Reserved 复位值: 0x0
15:2	res	Reserved 复位值: 0x0
1	r/w	LVD_FLG, 电压检测模块标记 0: $VCC \geq LVD_SEL$ 1: $VCC < LVD_SEL$ 注: 当 LVD_OUT=1 时, 该位硬件置位, 支持写 1 清零。 复位值: 0x0
0	ro	LVD_OUT, 电压检测模块输出 0: $VCC \geq LVD_SEL$ 1: $VCC < LVD_SEL$ 复位值: 0x0

7.2.14 系统复位控制寄存器 (SYSRST_CTRL_REG)

偏移地址: 0x0278

表 36 系统复位控制寄存器

位	访问	描述
31:1	res	Reserved 复位值: 0x0
0	r/w	PAD_RST_FLTEN, 外部复位的数字滤波使能 0: 关闭 1: 开启 注: 在更新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 复位值: 0x0

7.2.15 复位控制寄存器 0 (RST_CTRL_REG0)

偏移地址: 0x0280

表 37 复位控制寄存器 0

位	访问	描述
31:20	res	Reserved 复位值: 0x0
19	r/w	TIM8_SW_RSTJ, TIM8 软复位与否 0: 是 1: 否 复位值: 0x0
18	r/w	TIM2_SW_RSTJ, TIM2 软复位与否 0: 是 1: 否 复位值: 0x0
17	r/w	TIM1_SW_RSTJ, TIM1 软复位与否 0: 是 1: 否 复位值: 0x0
16	r/w	TIM0_SW_RSTJ, TIM0 软复位与否 0: 是 1: 否 复位值: 0x0
15:9	res	Reserved 复位值: 0x0
8	r/w	ADC_SW_RSTJ, ADC 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	HAU_SW_RSTJ, HAU 软复位与否 0: 是 1: 否 复位值: 0x0
5	res	Reserved 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	r/w	SCI2_SW_RSTJ, SCI2 软复位与否 0: 是 1: 否 复位值: 0x0

位	访问	描述
2	r/w	SCI1_SW_RSTJ, SCI1 软复位与否 0: 是 1: 否 复位值: 0x0
1	r/w	SCI0_SW_RSTJ, SCI0 软复位与否 0: 是 1: 否 复位值: 0x0
0	res	Reserved 复位值: 0x0

7.2.16 复位控制寄存器 1 (RST_CTRL_REG1)

偏移地址: 0x0284

表 38 复位控制寄存器 1

位	访问	描述
31:14	res	Reserved 复位值: 0x0
13	r/w	CMP_SW_RSTJ, CMP 软复位与否 0: 是 1: 否 复位值: 0x0
12:2	res	Reserved 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

7.2.17 时钟门控寄存器 0 (CLK_GAT_REG0)

偏移地址: 0x0290

表 39 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
30:26	res	Reserved 复位值: 0x0
25	r/w	BUZ1_CLK_GAT, 蜂鸣器 1 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
24:20	res	Reserved 复位值: 0x0
19	r/w	TIM8_CLK_GAT, TIM8 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
18	r/w	TIM2_CLK_GAT, TIM2 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
17	r/w	TIM1_CLK_GAT, TIM1 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
16	r/w	TIM0_CLK_GAT, TIM0 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
15:9	res	Reserved 复位值: 0x0
8	r/w	ADC_CLK_GAT, ADC 时钟门控使能控制 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	HAU_CLK_GAT, HAU 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0

位	访问	描述
5	res	Reserved 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	r/w	SCI2_CLK_GAT, SCI2 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
2	r/w	SCI1_CLK_GAT, SCI1 时钟门控使能控制 0: 关闭 1: 开启 复位值: 0x0
1	r/w	SCI0_CLK_GAT, SCI0 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

7.2.18 时钟门控寄存器 1 (CLK_GAT_REG1)

偏移地址: 0x0294

表 40 时钟门控寄存器 1

位	访问	描述
31:14	res	Reserved 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
12:2	res	Reserved 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

7.2.19 时钟选择寄存器 0 (CLK_SEL_REG0)

偏移地址: 0x02A0

表 41 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
19:18	res	Reserved 复位值: 0x0
17:16	ro	FCLK_SEL_INFO, FCLK 源时钟信息, 指示当前 FCLK 的时钟源: 00: HIRC (8MHz) 01: PLL (96MHz) 10: PLL (96MHz) 11: LIRC (32kHz) 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 (2^0) 1: 2 分频 (2^1) 2: 4 分频 (2^2) ... 7: 128 分频 (2^7) 8: 256 分频 (2^8) 其他: 保留 复位值: 0x0
3:2	res	Reserved 复位值: 0x0
1:0	r/w	FCLK_SEL, FCLK 源选择 00: HIRC (8MHz) 01: PLL (96MHz) 10: PLL (96MHz) 11: LIRC (32kHz) 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

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。

7.2.20 时钟选择寄存器 1 (CLK_SEL_REG1)

偏移地址: 0x02A4

表 42 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/2 0010: BUZ_SRC_CLK/4 ... 1101: BUZ_SRC_CLK/2 ¹³ 其他值: BUZ_SRC_CLK/2 ¹³ 注: BUZ_SRC_CLK 为 CLK_TST_MUX_SEL 选择所得。 复位值: 0x0
7:0	res	Reserved 复位值: 0x0

7.2.21 时钟选择寄存器 2 (CLK_SEL_REG2)

偏移地址: 0x02A8

表 43 时钟选择寄存器 2

位	访问	描述
31:22	res	Reserved 复位值: 0x0
21:20	r/w	WDT_CLK_SEL, WDT 时钟源选择 00: LIRC 01: HIRC 其他: 保留 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	CLK_TST_MUX_SEL, 测试输出时钟源选择 000: FCLK 001: HIRC 010: LIRC 011: 保留 100: 保留 101: PLL 110: SCLK 111: HCLK 注: 该位时钟也作为 BUZ 的时钟源。 复位值: 0x0
15:14	r/w	NF_CLK2_SEL, 噪声滤波时钟 2 源选择

位	访问	描述
		00: NF_SRC_CLK_PSC_8K 01: NF_SRC_CLK_PSC_16K 10: NF_SRC_CLK_PSC_32K 11: NF_SRC_CLK_PSC_64K 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_PSC_32 001: NF_SRC_CLK_PSC_64 010: NF_SRC_CLK_PSC_128 011: NF_SRC_CLK_PSC_256 100: NF_SRC_CLK_PSC_512 101: NF_SRC_CLK_PSC_1K 110: NF_SRC_CLK_PSC_2K 111: NF_SRC_CLK_PSC_4K 复位值: 0x0
10:9	r/w	NF_CLK0_SEL, 噪声滤波时钟 0 源选择 2'B00: NF_SRC_CLK_PSC_2 2'B01: NF_SRC_CLK_PSC_4 2'B10: NF_SRC_CLK_PSC_8 2'B11: NF_SRC_CLK_PSC_16 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	ADC_CLK_SEL, ADC 分频时钟选择 该控制位用于支持 FCLK 1 ~ 128 分频 复位值: 0x0

7.2.22 噪声滤波时钟控制寄存器 (NF_CLK_CTRL_REG0)

偏移地址: 0x02C0

表 44 噪声滤波时钟选择寄存器

位	访问	描述
31:19	res	Reserved 复位值: 0x0
18	r/w	PC_NF_CLK_EN, PC IO 输入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, PC IO 输入噪声滤波时钟选择 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, PB IO 输入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, PB IO 输入噪声滤波时钟选择 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, PA IO 输入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, PA IO 输入噪声滤波时钟选择 00: NF_CLK0 01: NF_CLK1 10: NF_CLK2 11: LIRC 复位值: 0x0

7.2.23 IO 噪声控制寄存器 0 (IO_NF_CTRL_REG0)

偏移地址: 0x0300

表 45 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

位	访问	描述
20	r/w	PB4_NF_DIS, PB4 噪声滤波功能关闭控制 0: 开启 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

位	访问	描述
8	r/w	PA8_NF_DIS, PA8 噪声滤波功能关闭控制 0: 开启 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

7.2.24 IO 噪声控制寄存器 1 (IO_NF_CTRL_REG1)

偏移地址: 0x0304

表 46 IO 噪声滤波控制寄存器 1

位	访问	描述
31:2	res	Reserved 复位值: 0x0
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

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

偏移地址: 0x0320

表 47 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

位	访问	描述
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 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

位	访问	描述
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
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

位	访问	描述
0	r/w	PA0_IE, PA0 输入使能控制 0: 关闭 1: 开启 复位值: 0x1

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

偏移地址: 0x0324

表 48 端口输入使能控制寄存器 1

位	访问	描述
31:2	res	Reserved 复位值: 0x0
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

7.2.27 引脚驱动控制寄存器 0 (IO_DS_CTRL_REG0)

偏移地址: 0x0340

表 49 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_DS
30	r/w	PB14_DS, PB14 驱动能力选择 参照 PA0_DS
29	r/w	PB13_DS, PB13 驱动能力选择 参照 PA0_DS
28	r/w	PB12_DS, PB12 驱动能力选择 参照 PA0_DS
27	r/w	PB11_DS, PB11 驱动能力选择 参照 PA0_DS
26	r/w	PB10_DS, PB10 驱动能力选择 参照 PA0_DS
25	r/w	PB9_DS, PB9 驱动能力选择 参照 PA0_DS
24	r/w	PB8_DS, PB8 驱动能力选择 参照 PA0_DS
23	r/w	PB7_DS, PB7 驱动能力选择 参照 PA0_DS
22	r/w	PB6_DS, PB6 驱动能力选择 参照 PA0_DS

位	访问	描述
21	r/w	PB5_DS, PB5 驱动能力选择 参照 PA0_DS
20	r/w	PB4_DS, PB4 驱动能力选择 参照 PA0_DS
19	r/w	PB3_DS, PB3 驱动能力选择 参照 PA0_DS
18	r/w	PB2_DS, PB2 驱动能力选择 参照 PA0_DS
17	r/w	PB1_DS, PB1 驱动能力选择 参照 PA0_DS
16	r/w	PB0_DS, PB0 驱动能力选择 参照 PA0_DS
15	r/w	PA15_DS, PA15 驱动能力选择 参照 PA0_DS
14	r/w	PA14_DS, PA14 驱动能力选择 参照 PA0_DS
13	r/w	PA13_DS, PA13 驱动能力选择 参照 PA0_DS
12	r/w	PA12_DS, PA12 驱动能力选择 参照 PA0_DS
11	r/w	PA11_DS, PA11 驱动能力选择 参照 PA0_DS
10	r/w	PA10_DS, PA10 驱动能力选择 参照 PA0_DS
9	r/w	PA9_DS, PA9 驱动能力选择 参照 PA0_DS
8	r/w	PA8_DS, PA8 驱动能力选择 参照 PA0_DS
7	r/w	PA7_DS, PA7 驱动能力选择 参照 PA0_DS
6	r/w	PA6_DS, PA6 驱动能力选择 参照 PA0_DS
5	r/w	PA5_DS, PA5 驱动能力选择 参照 PA0_DS
4	r/w	PA4_DS: PA3 驱动能力选择 参照 PA0_DS
3	r/w	PA3_DS, PA3 驱动能力选择 参照 PA0_DS
2	r/w	PA2_DS, PA2 驱动能力选择 参照 PA0_DS
1	r/w	PA1_DS, PA1 驱动能力选择 参照 PA0_DS
0	r/w	PA0_DS, PA0 驱动能力选择 0: 5mA 1: 10mA 复位值: 0x0

7.2.28 引脚驱动控制寄存器 1 (IO_DS_CTRL_REG1)

偏移地址: 0x0344

表 50 引脚驱动控制寄存器 1

位	访问	描述
31:2	res	Reserved 复位值: 0x0
1	r/w	PC1_DS, PC1 驱动能力选择 参照 PC0_DS
0	r/w	PC0_DS, PC0 驱动能力选择 0: 5mA 1: 10mA 复位值: 0x0

7.2.29 引脚上拉控制寄存器 0 (IO_PU_CTRL_REG0)

偏移地址: 0x0360

表 51 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
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

位	访问	描述
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
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
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

7.2.30 引脚上拉控制寄存器 1 (IO_PU_CTRL_REG1)

偏移地址: 0x0364

表 52 引脚上拉控制寄存器 1

位	访问	描述
31:2	res	Reserved 复位值: 0x0
1	r/w	PC1_PU, PC0 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

7.2.31 引脚下拉控制寄存器 0 (IO_PD_CTRL_REG0)

偏移地址: 0x0380

表 53 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
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

位	访问	描述
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 参照 PA0_PD
7	r/w	PA7_PD, PA7 下拉使能控制 参照 PA0_PD
6	r/w	PA6_PD, PA6 下拉使能控制 参照 PA0_PD
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

7.2.32 引脚下拉控制寄存器 1 (IO_PD_CTRL_REG1)

偏移地址: 0x0384

表 54 引脚下拉控制寄存器 1

位	访问	描述
31:2	res	Reserved 复位值: 0x0
1	r/w	PC1_PD, PC0 下拉使能控制 参照 PC0_PD
0	r/w	PC0_PD, PC0 下拉使能控制 0: 关闭 1: 开启 复位值: 0x0

7.2.33 端口模式控制寄存器 0 (IO_MOD_CTRL_REG0)

偏移地址: 0x03A0

表 55 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 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
29:28	r/w	PA14_MOD, PA14 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
27:26	r/w	PA13_MOD, PA13 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
25:24	r/w	PA12_MOD, PA12 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
23:22	r/w	PA11_MOD, PA11 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
21:20	r/w	PA10_MOD, PA10 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
19:18	r/w	PA9_MOD, PA9 模式选择

位	访问	描述
		00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
17:16	r/w	PA8_MOD, PA8 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
15:14	r/w	PA7_MOD, PA7 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
13:12	r/w	PA6_MOD, PA6 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
11:10	r/w	PA5_MOD, PA5 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
9:8	r/w	PA4_MOD, PA4 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
7:6	r/w	PA3_MOD, PA3 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
5:4	r/w	PA2_MOD, PA2 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
3:2	r/w	PA1_MOD, PA1 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
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

7.2.34 端口模式控制寄存器 1 (IO_MOD_CTRL_REG1)

偏移地址: 0x03A4

表 56 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 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
29:28	r/w	PB14_MOD, PB14 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
27:26	r/w	PB13_MOD, PB13 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
25:24	r/w	PB12_MOD, PB12 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
23:22	r/w	PB11_MOD, PB11 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
21:20	r/w	PB10_MOD, PB10 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
19:18	r/w	PB9_MOD, PB9 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
17:16	r/w	PB8_MOD, PB8 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0

位	访问	描述
15:14	r/w	PB7_MOD, PB7 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
13:12	r/w	PB6_MOD, PB6 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
11:10	r/w	PB5_MOD, PB5 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
9:8	r/w	PB4_MOD, PB4 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
7:6	r/w	PB3_MOD, PB3 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
5:4	r/w	PB2_MOD, PB2 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
3:2	r/w	PB1_MOD, PB1 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
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

7.2.35 端口模式控制寄存器 2 (IO_MOD_CTRL_REG2)

偏移地址: 0x03A8

表 57 端口模式控制寄存器 2

位	访问	描述
31:4	res	Reserved 复位值: 0x0
3:2	r/w	PC1_MOD, PC1 模式选择 00: GPIO 模式 01: 功能复用模式 1x: 保留 复位值: 0x0
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

7.2.36 功能复用选择寄存器 0 (AF_SEL_REG0)

偏移地址: 0x03E0

表 58 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

位	访问	描述
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

7.2.37 功能复用选择寄存器 1 (AF_SEL_REG1)

偏移地址: 0x03E4

表 59 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: 保留 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: 0x0
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: 0x0
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

7.2.38 功能复用选择寄存器 2 (AF_SEL_REG2)

偏移地址: 0x03E8

表 60 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: 保留 其他: 保留 复位值: 0x0
23:20	r/w	PB5_AF_SEL, PB5 功能复用选择 0 ~ 7: 保留 其他: 保留 复位值: 0x0
19:16	r/w	PB4_AF_SEL, PB4 功能复用选择 0 ~ 7: 保留 其他: 保留 复位值: 0x0
15:12	r/w	PB3_AF_SEL, PB3 功能复用选择 0 ~ 7: 保留 其他: 保留 复位值: 0x0
11:8	r/w	PB2_AF_SEL, PB2 功能复用选择 0 ~ 7: 保留 其他: 保留 复位值: 0x0
7:4	r/w	PB1_AF_SEL, PB1 功能复用选择 0 ~ 7: 保留 其他: 保留 复位值: 0x0
3:0	r/w	PB0_AF_SEL, PB0 功能复用选择 0 ~ 7: 保留 其他: 保留 复位值: 0x0

7.2.39 功能复用选择寄存器 3 (AF_SEL_REG3)

偏移地址: 0x03EC

表 61 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: 保留 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

7.2.40 功能复用选择寄存器 4 (AF_SEL_REG4)

偏移地址: 0x03F0

表 62 功能复用选择寄存器 4

位	访问	描述
31:8	res	Reserved 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: 保留 其他: 保留 复位值: 0x0

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

偏移地址: 0x0460

表 63 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

位	访问	描述
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
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

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

偏移地址: 0x0464

表 64 端口开漏输出控制寄存器 1

位	访问	描述
31:3	res	Reserved
2	res	Reserved 复位值: 0x0
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

7.2.43 调试控制寄存器 (DBG_CTRL_REG)

偏移地址: 0x04C0

表 65 调试控制寄存器

位	访问	描述
31:1	res	Reserved 复位值: 0x0
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: 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 总线进行访问。

当引脚配置为 GPIO 功能时，GPIO 输出使能寄存器和输出值寄存器分别控制每个引脚的方向和输出状态。当引脚相应端口被启用并且其数字输入功能激活时，GPIO 输入值寄存器可以显示每个引脚上的逻辑值。

8.2 主要功能

KPM32K07XX GPIO 主要功能如下，GPIO 提供 16 位 I/O 接口。

- GPIO 输出实现置 1、清 0 和翻转
- 通过控制寄存器中置位寄存器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

8.3 模块框图

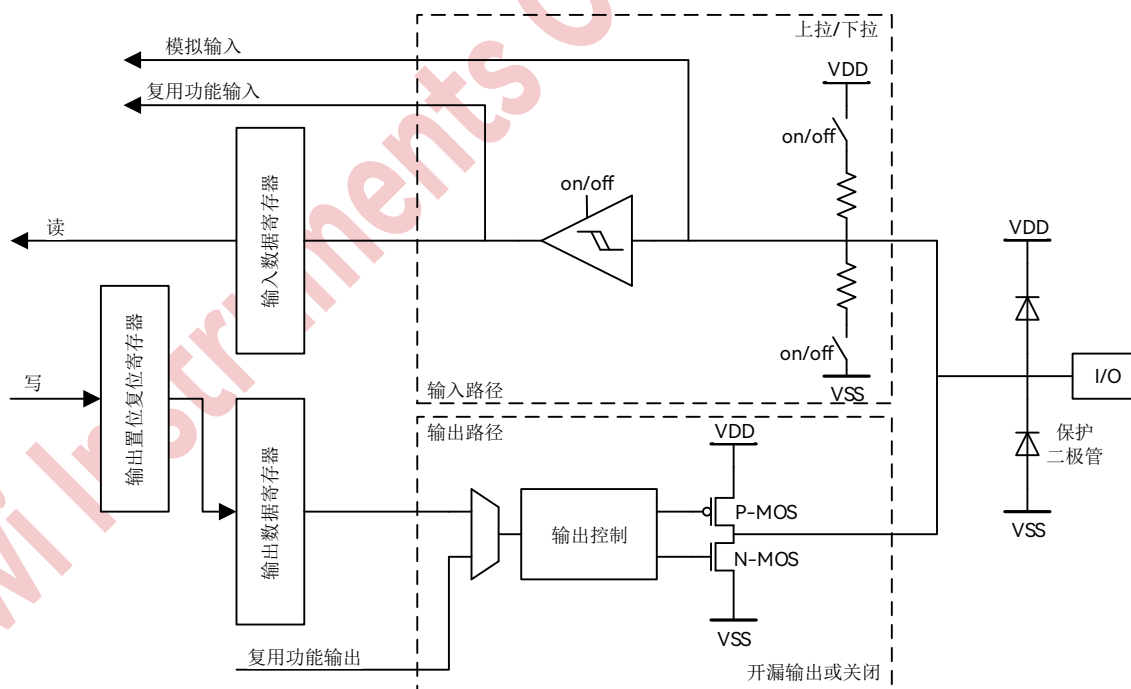


图 11 I/O 结构图

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, 还拥有 2 个数据寄存器 (GPIOx_DATA_OUT, GPIOx_DATA_IN) 和 2 个输出控制寄存器 (GPIOx_DATA_OUT_SET_CLR, GPIOx_TOG)。

8.4.1 GPIO 地址分布

每组 GPIO 对应地址分布参考下方表格。

表 66 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)

8.4.2 GPIO 输出控制

通过输出使能寄存器 (GPIOx_DOUT_EN) 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 (GPIOx_DATA_IN) 可以查看 IO 逻辑状态; 当输出使能打开时, IO 处于输出功能, 通过输出数据寄存器 (GPIOx_DATA_OUT) 可以更改 IO 状态。

以下 4 种方式可以更改 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)

8.5 寄存器描述

GPIO 基地址: GPIOA: 0x5000_0000; GPIOB: 0x5000_1000; GPIOC: 0x5000_2000。

注: GPIOA 有 15 个有效 IO, 分别是 bit0-bit14。GPIOB 有 14 个有效 IO, 分别是 bit0 ~ bit12, bit15。GPIOC 有 2 个有效 IO, 分别是 bit0 ~ bit1。

表 67 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x10	GPIO 输入值	32	r/w	0x00000000
0x14	GPIO 输出值	32	r/w	0x00000000
0x18	GPIO 输出置位清零	32	r/w	0x00000000

偏移地址	寄存器列表	位宽	访问	复位值
0x1C	GPIO 输出翻转	32	r/w	0x00000000
0x20	GPIO 输出使能	32	r/w	0x00000000
0x24	GPIO 输出使能清除	32	r/w	0x00000000

8.5.1 GPIOx 输入值 (GPIOx_DATA_IN)

偏移地址: 0x0010

表 68 GPIOx 输入值

位	访问	描述
31:16	res	Reserved 复位值: 0x0
15:0	r/w	DATA_IN: IO 状态输入值 0: IO 为低电平 1: IO 为高电平 读: IO 实际输入值 写: 更改 IO 输出设置 复位值: 0x0

8.5.2 GPIOx 输出值 (GPIOx_DATA_OUT)

偏移地址: 0x0014

表 69 GPIOx 输出值

位	访问	描述
31:16	res	Reserved 复位值: 0x0
15:0	r/w	DATA_OUT: IO 状态输出值 0: 将对应 IO 设置为低电平 1: 将对应 IO 设置为高电平 读: IO 输出设置值 写: 更改 IO 输出设置 复位值: 0x0

8.5.3 GPIOx 输出值置位清零 (GPIOx_DATA_OUT_SET_CLR)

偏移地址: 0x0018

表 70 GPIOx 输出值置位清零

位	访问	描述
31:16	r/w	DATA_OUT_SET: IO 状态输出值置 1 0: 无效动作 1: 将对应 IO 输出设置值置 1 注: 读一直为 0。 复位值: 0x0

位	访问	描述
15:0	r/w	DATA_OUT_CLR: IO 状态输出值清 0 0: 无效动作 1: 将对应 IO 输出设置值清 0 注: 读一直为 0。 复位值: 0x0

8.5.4 GPIOx 输出翻转 (GPIOx_TOG)

偏移地址: 0x001C

表 71 GPIOx 输出翻转

位	访问	描述
31:16	res	Reserved 复位值: 0x0
15:0	r/w	DOUT_TOG: GPIO 输出使能 0: 无效动作 1: 将对应 IO 输出设置值翻转 注: 读一直为 0。 复位值: 0x0

8.5.5 GPIOx 输出使能 (GPIOx_DOUT_EN)

偏移地址: 0x0020

表 72 GPIOx 输出使能

位	访问	描述
31:16	res	Reserved 复位值: 0x0
15:0	r/w	DOUT_EN: GPIO 输出使能 0: 无效操作 1: GPIO 输出使能打开 复位值: 0x0

8.5.6 GPIOx 输出使能清除 (GPIOx_DOUT_EN_CLR)

偏移地址: 0x0024

表 73 GPIOx 输出使能清除

位	访问	描述
31:16	res	Reserved 复位值: 0x0
15:0	r/w	DOUT_EN_CLR: GPIO 输出使能清零 0: 无效操作 1: GPIO 输出使能关闭 注: 读该寄存器的数值为 GPIO 输出使能状态 复位值: 0x0

9 硬件计算加速器 (HAU)

9.1 概述

该硬件加速器包含一个除法计算器 (DIV) 及一个三角函数计算器 (TMU)。TMU 通过 CORDIC 算法实现正弦、余弦和反正切计算，数据类型支持 Q15 格式输入输出。DIV 支持有符号型及无符号型 32 位定点类型数据计算。

9.2 主要功能

- $a = \sin(\text{angle})$
- $a = \cos(\text{angle})$
- $a = \arctan(x, y)$
- $a/b = c \dots d$
- 支持 $\sin/\cos/\arctan/\text{div}$ 计算模式自动切换
- DIV 支持无符号型及有符号型 32 位除法计算，
- 通过寄存器选择 DIV 运算类型 (无符号型，有符号型)
- TMU 数据输入类型 Q15
- TMU 数据输出类型 Q15
- 正余弦弧度输入范围 $(-\pi, \pi)$ (按比例分配至 $(-32767, 32767)$)
- 正余弦计算输出范围 $(-1, 1)$
- 反正切计算 $\sin(y)$ 、 $\cos(x)$ 输入范围 $(-1, 1)$ (按比例分配至 $(-32767, 32767)$)
- 反正切计算输出范围 $(-\pi, \pi)$ (按比例分配至 $(-32767, 32767)$)
- TMU 计算周期
 - $a = \sin(\text{angle})$ (12 个 HAU 工作时钟周期)
 - $a = \cos(\text{angle})$ (12 个 HAU 工作时钟周期)
 - $a = \arctan(x, y)$ (12 个 HAU 工作时钟周期)
- DIV 计算周期
 - $a/b = c \dots d$ (16 个 HAU 工作时钟周期)

9.3 模块框图

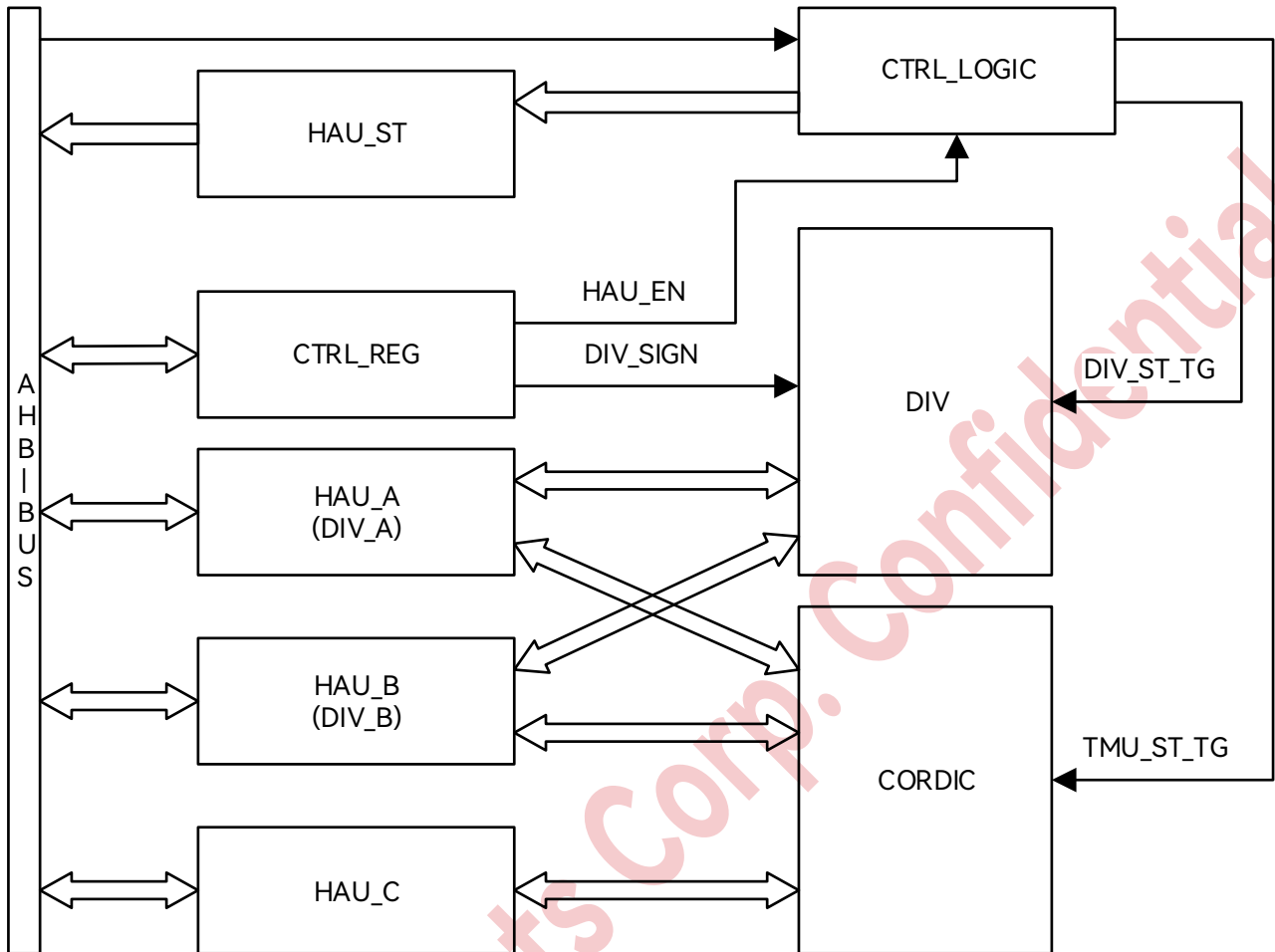


图 12 HAU 模块框图

9.4 功能描述

9.4.1 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。

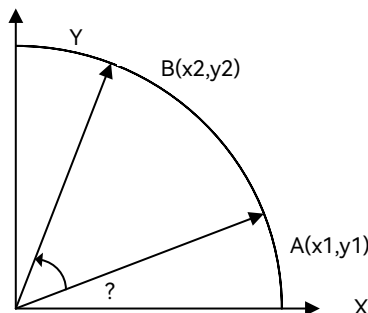


图 13 A(x1,y1) B(x2,y2)

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。

$$x_2 = x_1 \cos \theta - y_1 \sin \theta = \cos \theta (x_1 - y_1 \tan \theta)$$

$$y_2 = x_1 \sin \theta + y_1 \cos \theta = \cos \theta (y_1 + x_1 \tan \theta)$$

消去 $\cos$ 得到:

$$x_2' = x_1 - y_1 \tan \theta$$

$$y_2' = y_1 + x_1 \tan \theta$$

则存在下图关系:

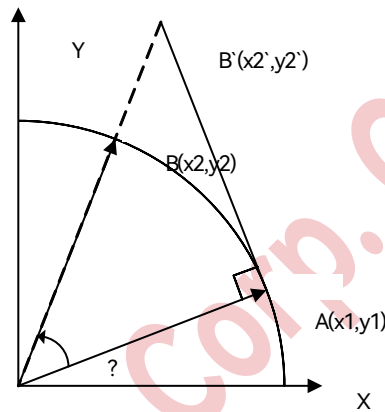


图 14 B'(x2',y2')

可知向量 B' 的模值是向量 B 的模值的 $1/\cos$ 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:

$$x_2' = x_1 - y_1 \tan \theta = x_1 - y_1 * 2^{-i}$$

$$y_2' = y_1 + x_1 \tan \theta = y_1 + x_1 * 2^{-i}$$

通过这种方法使角度不断逼近计算目标值。

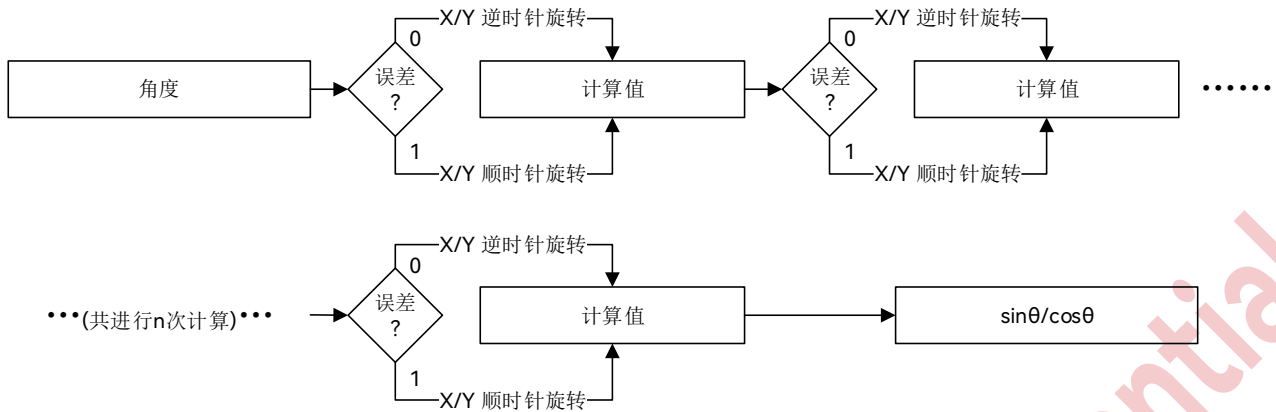


图 15 Cordic 算法计算流程

将这组角度值整理如下，因为每次旋转计算将会使向量 B 的模值放大 $1/\cos\theta$ 倍，所以在经过 n 轮计算后，将结果乘 0.60725294 即可得到最终准确模值。

表 74 Cordic 算法计算角度 (十进制)

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
**	**	**	**	**	**	**
i	θ_i (角度制)	θ_i (弧度制)	$\tan\theta_i$	2^{-i}	$\arctan(2^{-i})$	$\cos(\arctan(2^{-i}))$

9.4.2 TMU 工作时序

当进行正余弦计算时，先将需要计算的角度写入 HAU_DATA_A 寄存器中，硬件自动识别进行正余弦计算，在 12 个计算周期后，通过 HAU_DATA_B 及 HAU_DATA_C 获取计算的正余弦值。

如下图所示为计算正弦余弦值的过程，第一个周期是确定输入角度象限及将输入角度转换到第一象限进行计算。中间十个周期按照 **cordic** 算法逐渐逼近计算角度，得到正余弦。最后一个周期将所得到的正余弦值根据角度所在的象限进行转换，完成最终计算。

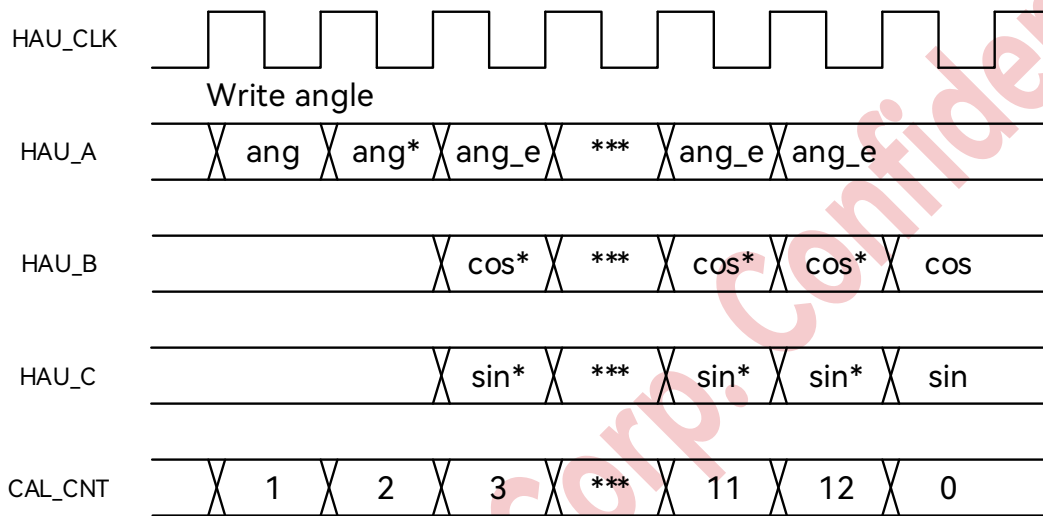


图 16 正余弦计算时序

当进行反正切计算时，先将余弦值写入 HAU_DATA_B 寄存器中后再将正弦值写入 HAU_DATA_C 寄存器，硬件自动识别进行反正切计算，在 12 个计算周期后，读取 HAU_DATA_A 寄存器得到反正切计算结果。

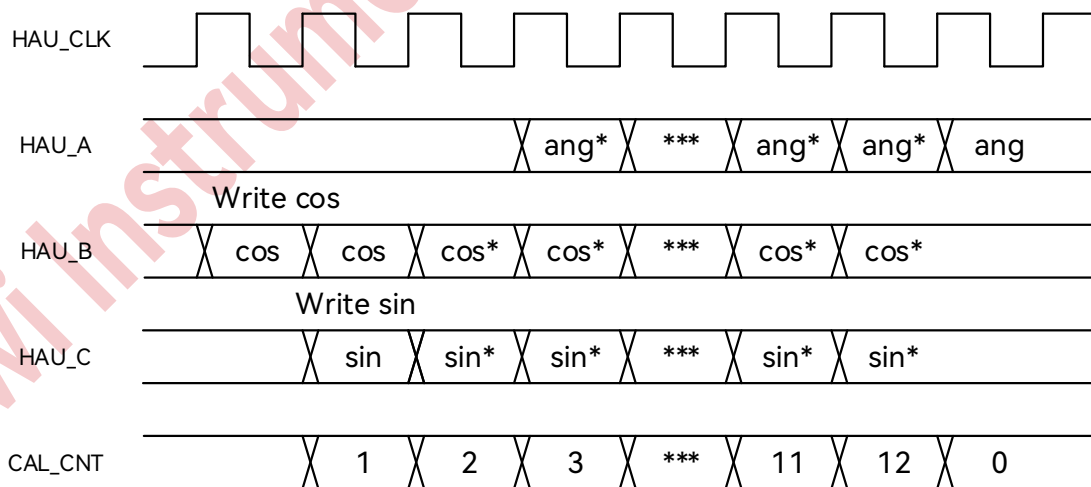


图 17 反正切计算时序

9.4.3 DIV 工作时序

HAU 支持 32 位有符号/无符号型除法计算，写入除数开始计算。可通过控制寄存器决定当前进行有符号型还是无符号型的除法计算，计算完成后商存放在被除数寄存器 (DIV_DATA_A) 中，余数存放在除数寄存器 (DIV_DATA_B) 中。当输出的除数为 0 时，除数为 0 状态位会被置 1，在进行下次计算开始时，该状态位会自动清零。完成一次除法运算需要 16 个时钟周期。

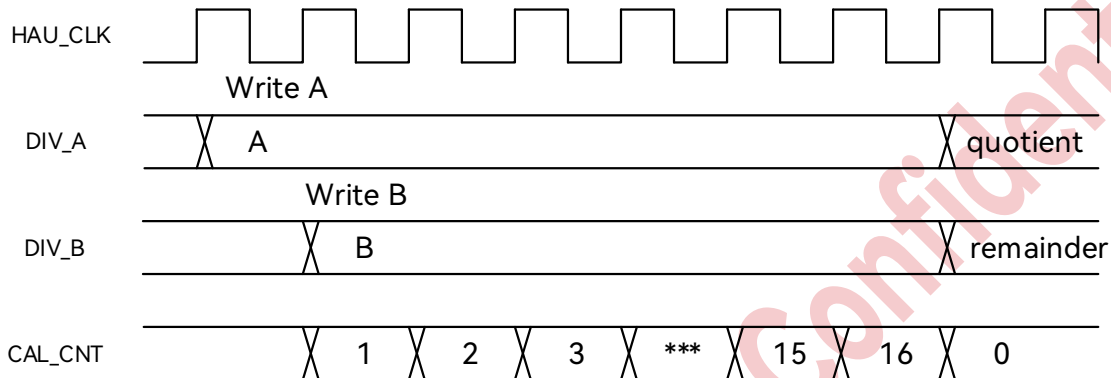


图 18 除法器计算时序

9.5 寄存器描述

HAU 基地址: 0x4003_1000

表 75 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	HAU 控制寄存器	32	r/w	0x00000000
0x04-0x0C	Reserved	32	res	0x00000000
0x10	HAU 数据寄存器 A	32	r/w	0x00000000
0x14	HAU 数据寄存器 B	32	r/w	0x00000000
0x18	HAU 数据寄存器 C	32	r/w	0x00000000
0x1C	Reserved	32	res	0x00000000
0x20	DIV 数据寄存器 A	32	r/w	0x00000000
0x24	DIV 数据寄存器 B	32	r/w	0x00000000
0x28-0x2C	Reserved	32	res	0x00000000
0x30	HAU 状态寄存器	32	r/w	0x00000000

9.5.1 HAU 控制寄存器 (HAU_CTRL)

偏移地址: 0x0000

表 76 HAU 控制寄存器

位	访问	描述
31:2	res	Reserved 复位值: 0x0
1	r/w	HAU_F_IN: HAU 计算输入模式 当前进行 DIV 运算时生效 0: 无符号型除法器 1: 有符号型除法器 复位值: 0x0
0	r/w	HAU_EN: HAU 使能 0: 打开 HAU 1: 关闭 HAU 复位值: 0x0

9.5.2 HAU 数据寄存器 A (HAU_DATA_A)

偏移地址: 0x0010

表 77 HAU 数据寄存器 A

位	访问	描述
31:16	res	Reserved 复位值: 0x0
15:0	r/w	HAU_DATA_A: 数据寄存器 A TMU 模式: 当进行 SIN/COS 运算时, 向该寄存器填入计算角度即开始计算 当进行 ARCTAN 运算时, 从该寄存器读取计算的角度值 注: 计算过程中不可访问。 复位值: 0x0

9.5.3 HAU 数据寄存器 B (HAU_DATA_B)

偏移地址: 0x0014

表 78 HAU 数据寄存器 B

位	访问	描述
31:16	res	Reserved 复位值: 0x0
15:0	r/w	HAU_DATA_B: 数据寄存器 B TMU 模式: 当进行 SIN/COS 运算时, 从该寄存器读取计算的余弦值 当进行 ARCTAN 运算时, 向该寄存器填入计算 X 数值 注: 计算过程中不可访问。 复位值: 0x0

9.5.4 HAU 数据寄存器 C (HAU_DATA_C)

偏移地址: 0x0018

表 79 HAU 数据寄存器 C

位	访问	描述
31:16	res	Reserved 复位值: 0x0
15:0	r/w	HYAU_DATA_C: 数据寄存器 C TMU 模式: 当进行 SIN/COS 运算时, 从该寄存器读取计算的正弦值 当进行 ARCTAN 运算时, 向该寄存器填入计算 Y 数值 DIV 模式: Reserved 注: 计算过程中不可访问。 复位值: 0x0

9.5.5 DIV 数据寄存器 A (DIV_DATA_A)

偏移地址: 0x0020

表 80 DIV 数据寄存器 A

位	访问	描述
31:0	r/w	DIV_DATA_A: 除法器被除数 写: 写入被除数 读: 读取计算结果商 注: 计算过程中不可访问。 复位值: 0x0

9.5.6 DIV 数据寄存器 B (DIV_DATA_B)

偏移地址: 0x0024

表 81 DIV 数据寄存器 B

位	访问	描述
31:0	r/w	DIV_DATA_B: 除法器除数 写: 写入除数 读: 读取计算结果余数 注: 计算过程中不可访问。 复位值: 0x0

9.5.7 HAU 状态寄存器 (HAU_ST)

偏移地址: 0x0030

表 82 HAU 状态寄存器

位	访问	描述
31:5	res	Reserved 复位值: 0x0
4	ro	DIV_DIV0: 除数为 0 0: 除数不等于 0 1: 除数等于 0 注: 在完成 DIV 运算时自动更新。 复位值: 0x0
3	res	Reserved 复位值: 0x0
2	ro	TMU_MOD: TMU 当前运算模式 0: sin/cos 1: arctan 注: TMU 支持运算模式自检测, 在进行 SIN/COS 运算时, 请将输入角度写入 HAU_DATA_A 寄存器, 在写入后 TMU 自动开始进行计算, 计算后正余弦值的结果分别存储在 HAU_DATA_B/C 寄存器中。在进行 ARCTAN 运算时, 请先将输入的 X 数值写入 HAU_DATA_B 寄存器, 再将输入的 Y 数值写入 HAU_DATA_C 寄存器, 在填写完成 Y 值后 TMU 自动开始进行计算, 计算后角度值的结果存储在 HAU_DATA_A 寄存器中。该状态寄存器在进行新一轮 TMU 运算时自动更新。 复位值: 0x0
1	ro	HAU_MOD: HAU 当前运算模式 0: TMU 计算模式 1: DIV 计算模式 注: HAU 支持运算模式自检测, 在进行 TMU 运算时, 输入数据及结果通过 HAU_DATA_A/B/C 参与运算。在进行 DIV 运算时, 输入数据及结果通过 DIV_DATA_A/B 参与运算。该状态寄存器在进行新一轮运算时自动更新。 HAU 包含 TMU 及 DIV 两种运算模式, 注意这两种模式不可同时使用, 需等一种模式结束再开始下一轮计算。 复位值: 0x0
0	ro	HAU_BUSY: HAU 忙标志 0: HAU 完成计算 1: HAU 正在计算 复位值: 0x0

10 模数转换器 (ADC)

10.1 概述

KPM32K07XX 配备 1 个 12bit 分辨率的逐次逼近型模数转换器，支持最多 16 个通道的模拟信号输入，其中，11 个通道连接到芯片外部。模数转换器支持可配置最高 1.25Msps 的转换率，控制电路部分基于转换单元机制，6 个转换单元支持独立配置通道，采样触发源等，并且，转换结果保存在基于转换单元的结果寄存器中。

10.2 主要功能

- 支持最高可达 1.25Msps 采样率
- 支持 11 个外部通道采样和 5 个内部通道采样
- 支持 12-bit/10-bit/8-bit 分辨率
- 支持 6 个可独立配置的转换单元
- 支持 6 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 1 个后处理模块
 - 看门狗比较模式
- 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

10.3 模块框图

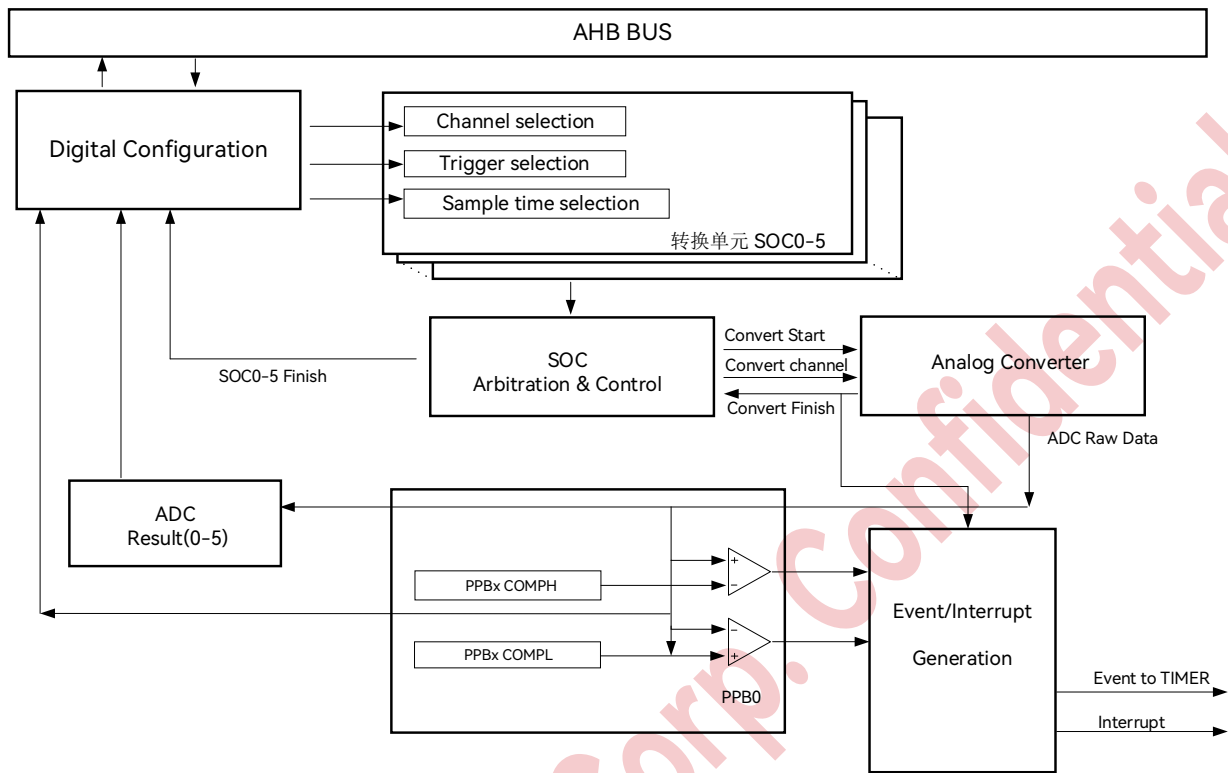


图 19 ADC 模块框图

10.4 功能描述

10.4.1 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 83 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

10.4.2 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 6, 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。

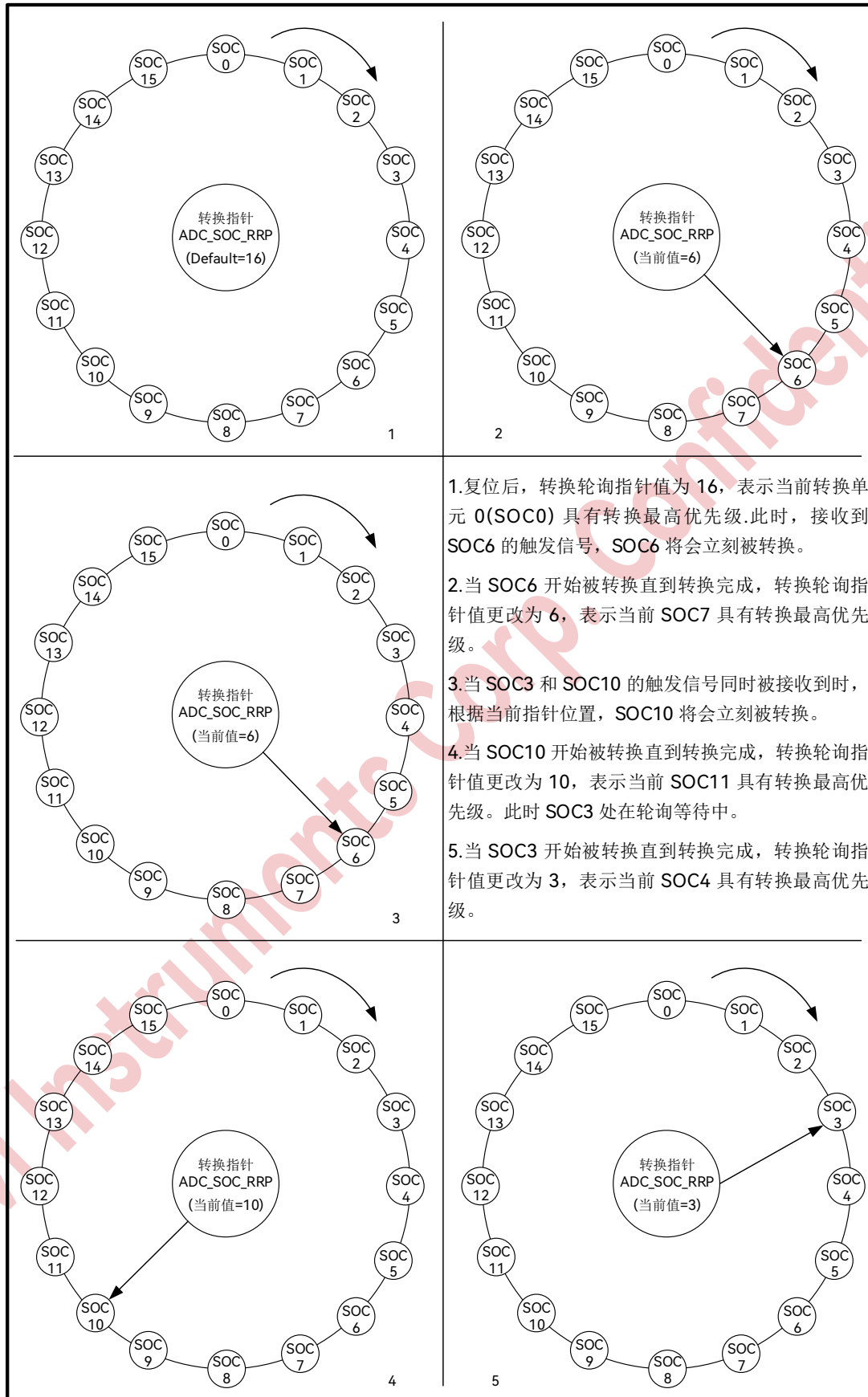


图 20 默认情况下的轮询模式 (图中以 16 个转换单元为例)

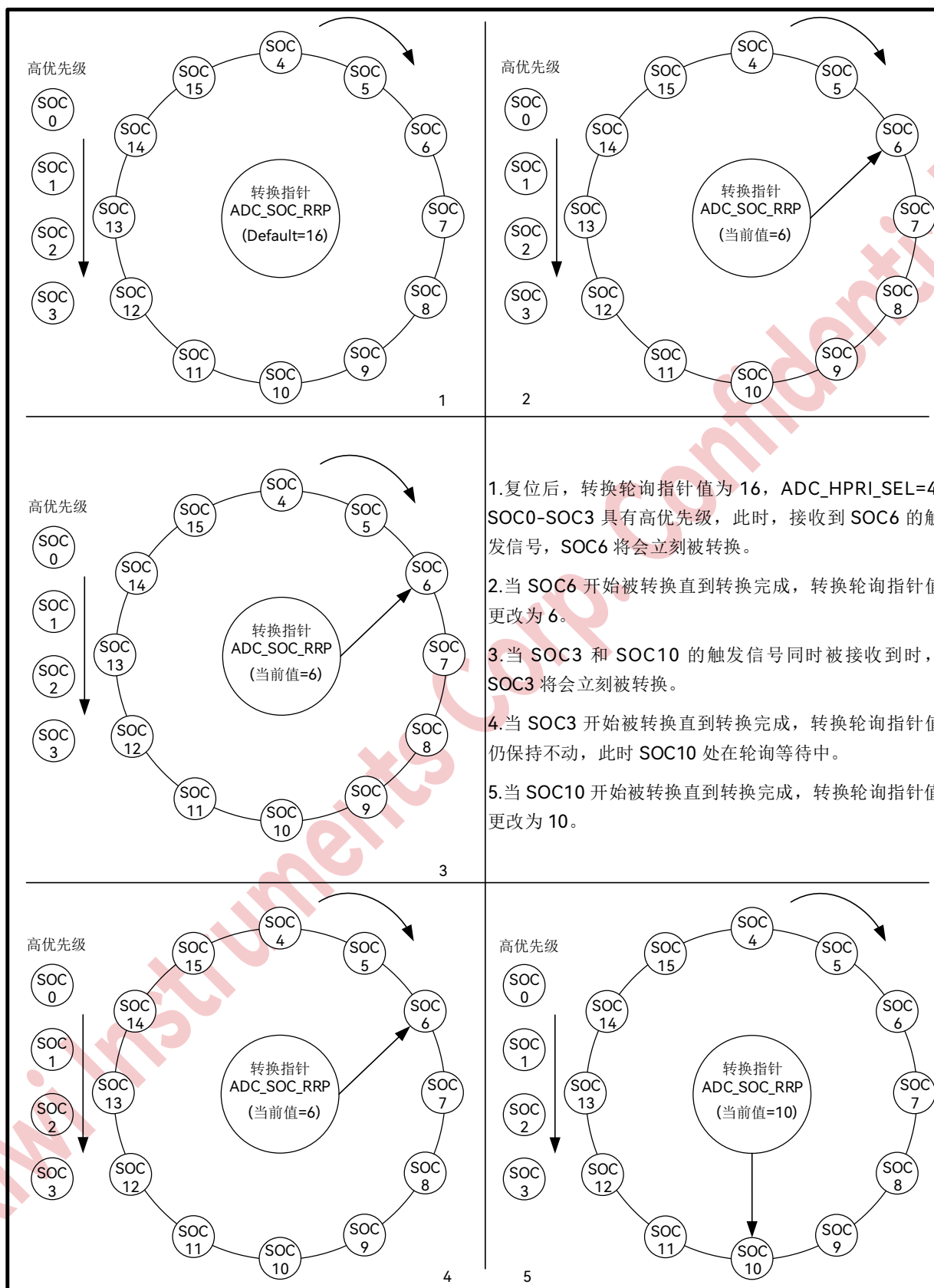


图 21 高优先级转换机制 (图中以 16 个转换单元为例)

10.4.3 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完成，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。

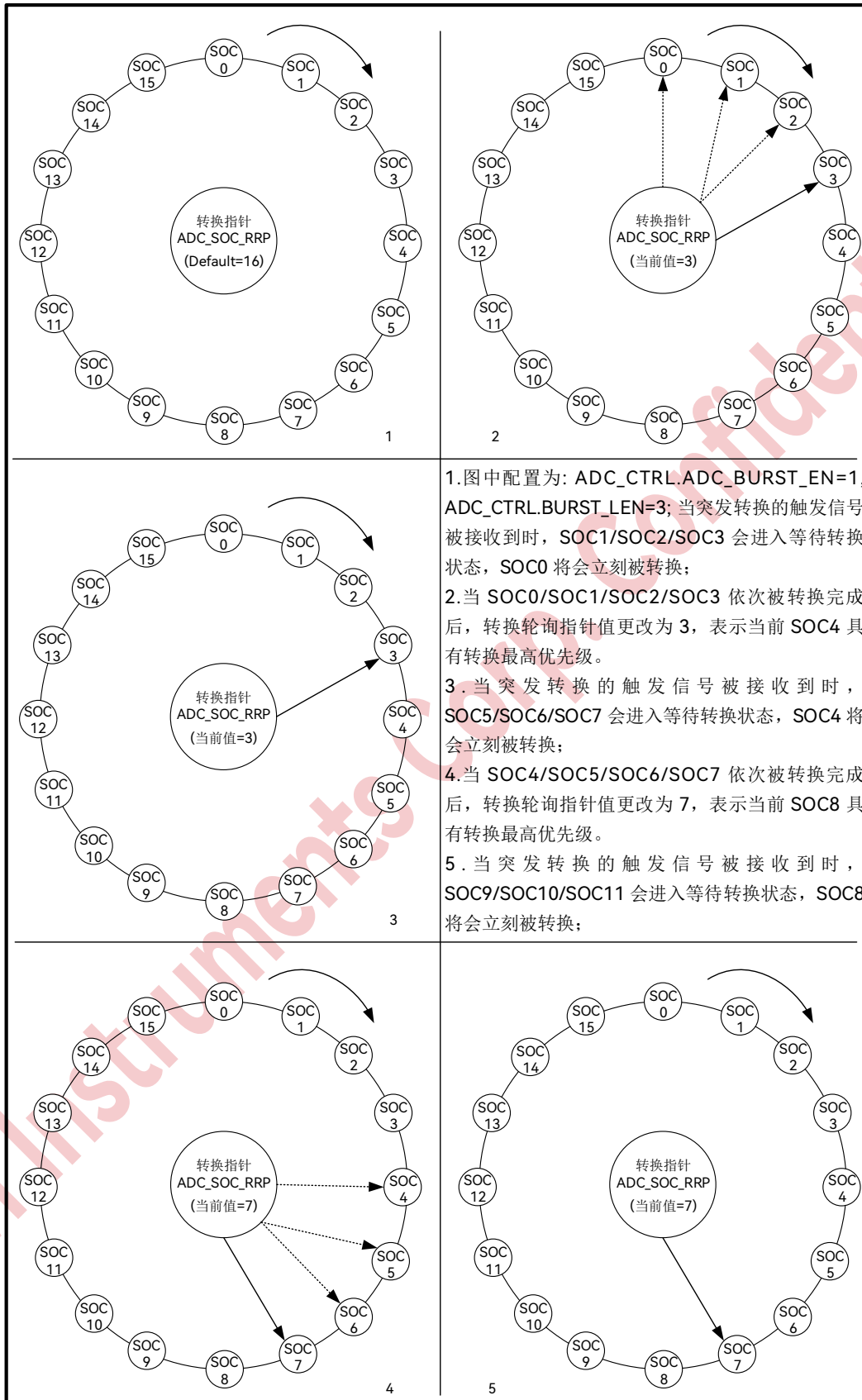


图 22 突发模式转换机制 (关闭高优先级模式) (图中以 16 个转换单元为例)

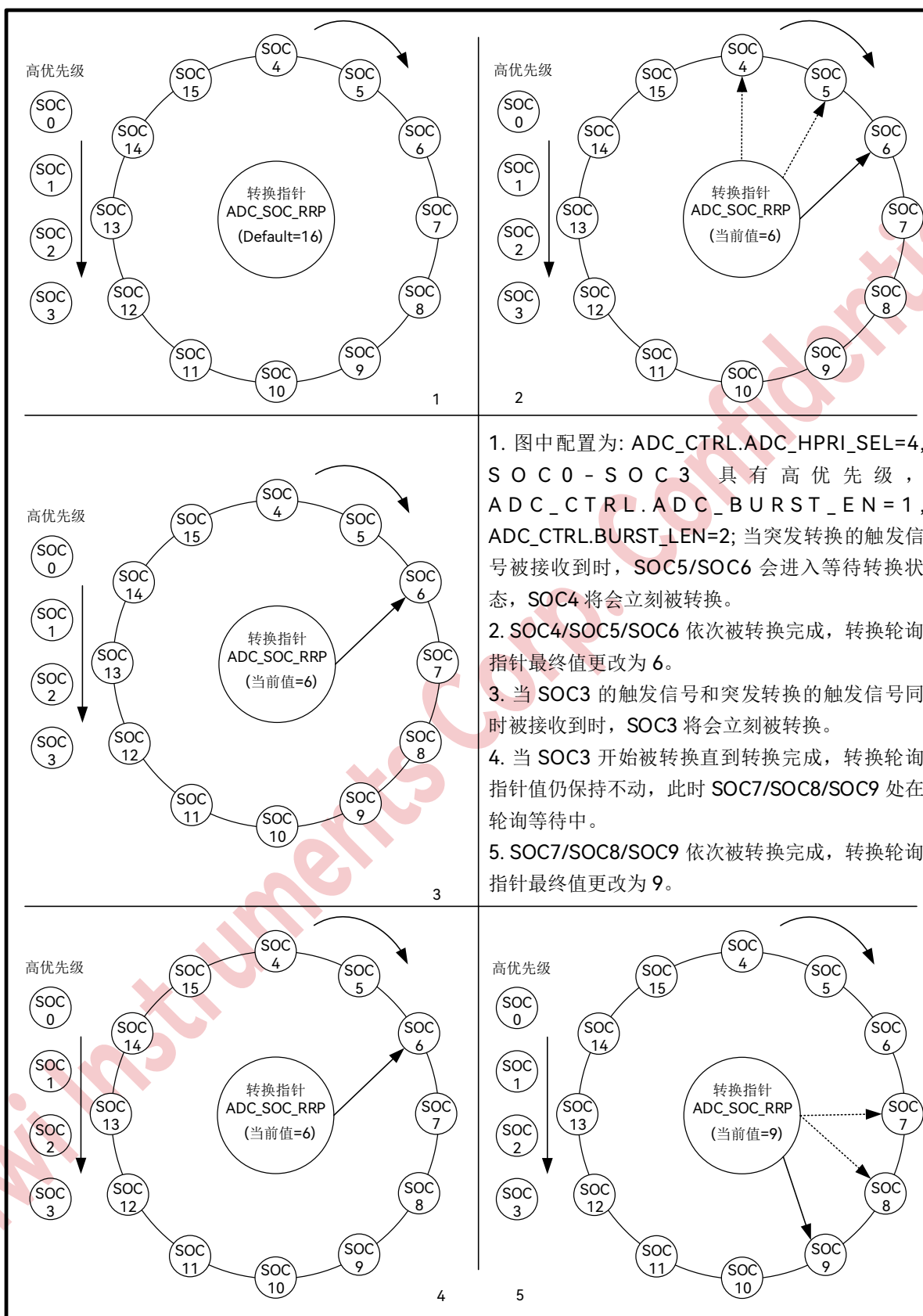


图 23 突发模式转换机制 (使能高优先级模式) (图中以 16 个转换单元为例)

10.4.4 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。

10.4.5 结果后处理

模数转换器包含 1 个结果后处理模块，后处理模块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功能。

- 自动比较，产生电压阈值上限溢出状态和下限溢出状态

10.4.5.1 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，比较后会产生比较匹配的状态标志位，通过配置中断使能位，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 `TIMER` 模块中。

在通过 `ADC_PPbX_CFG.PPBx_SOC_SEL` 位选择相应的转换单元后，该转换单元的转换结果会与 `ADC_PPbX_COMPH` 寄存器和 `ADC_PPbX_COMPL` 寄存器所配置的阈值进行比较，比较模式通过 `PPB0` 设置寄存器中的 `ADC_COMP_MOD` 进行配置。

表 84 模数转换器比较模式

比较模式配置	比较方式
2'B00	结果 < 下限值
2'B01	结果 > 上限值
2'B10	结果 ≥ 下限值并且结果 ≤ 上限值
2'B11	结果 < 下限值或者结果 > 上限值

10.4.6 ADC 中断号

模数转换器的中断状态寄存器中的状态都可独立使能是否产生中断，当中断产生后，软件可通过中断状态清除寄存器将中断状态清除。

表 85 模数转换器中断号

名称	中断号
模数转换器	9

10.5 寄存器描述

ADC 寄存器的基地址为 0x4002_0000，下表为 ADC 的相关寄存器描述。

表 86 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	0x00000600
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
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
0xB0	ADC 模拟通道使能寄存器	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
0xD4	ADC 转换单元配置寄存器 5	32	r/w	0x00000000
0x110	ADC PPB0 设置寄存器	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

10.5.1 ADC 软件触发寄存器 (ADC_TRIG)

偏移地址: 0x0000

表 87 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
15:6	res	Reserved 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
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

10.5.2 ADC 控制寄存器 (ADC_CTRL)

偏移地址: 0x0004

表 88 ADC 控制寄存器

位	访问	描述
31	res	Reserved 复位值: 0x0
30:28	r/w	BURST_LEN: 突发模式转换长度 000: 转换长度为 1 001: 转换长度为 2 010: 转换长度为 3 011: 转换长度为 4 100: 转换长度为 5 101: 转换长度为 6 其他: 保留 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: TIM0 OC1 0001: TIM0 OC2 0010: TIM0 TRGO 0011: TIM1 OC1 0100: TIM2 OC1 0101: TIM8 事件 0 0110: TIM8 事件 1 0111: TIM8 TRGO 1000: 转换单元 0 完成事件 1001: 转换单元 1 完成事件 1010: 转换单元 2 完成事件 1011: 转换单元 3 完成事件 1100: 转换单元 4 完成事件 1101: 转换单元 5 完成事件 1110: 任一转换单元事件 1111: 保留 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
15:6	res	Reserved 复位值: 0x0

位	访问	描述
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: 保留 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: 保留 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

10.5.3 ADC 转换单元优先级寄存器 (ADC_SOC_PRI)

偏移地址: 0x0010

表 89 ADC 转换单元优先级控制寄存器

位	访问	描述
31	res	Reserved 复位值: 0x0
30:28	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
27:11	res	Reserved 复位值: 0x0
10:8	ro	ADC_SOC_RRP: 转换单元轮询指针 000: 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 001: 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 010: 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 011: 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 100: 转换单元 4 在上次被转换完成, 转换单元 5 具有最高优先级 101: 转换单元 5 在上次被转换完成, 转换单元 0 具有最高优先级 110: 转换单元 0 具有最高优先级 其他: 保留 注: 每次配置 ADC_HPRI_SEL 位时, 转换指针会被复位。 复位值: 0x6
7:3	res	Reserved 复位值: 0x0
2:0	r/w	ADC_HPRI_SEL: 高优先级转换单元选择 000: 所有转换单元优先级相同, 都为轮询模式

位	访问	描述
		001: 转换单元 0 具有高优先级, 转换单元 1-5 为轮询模式 010: 转换单元 0 和 1 具有高优先级, 转换单元 2-5 为轮询模式 011: 转换单元 0, 1 和 2 具有高优先级, 转换单元 3-5 为轮询模式 100: 转换单元 0, 1, 2 和 3 具有高优先级, 转换单元 4-5 为轮询模式 101: 转换单元 0, 1, 2, 3 和 4 具有高优先级, 转换单元 5 为轮询模式 110: 所有转换单元均为高优先级模式 其他: 保留 复位值: 0x0

10.5.4 ADC 中断使能寄存器 (ADC_INTEN)

偏移地址: 0x0020

表 90 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
27:17	res	Reserved 复位值: 0x0
16	r/w	PPB0_COMP_INTEN: PPB0 结果比较匹配中断使能 0: 关闭 1: 使能 复位值: 0x0
15:6	res	Reserved 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

位	访问	描述
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

10.5.5 ADC 中断状态寄存器 (ADC_STA)

偏移地址: 0x0024

表 91 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
27:17	res	Reserved 复位值: 0x0
16	ro	PPB0_COMP_INT: PPB0 结果比较匹配状态标志 0: 无状态发生 1: 比较结果匹配 复位值: 0x0
15:6	res	Reserved 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

10.5.6 ADC 中断状态清除寄存器 (ADC_STA_CLR)

偏移地址: 0x0028

表 92 ADC 中断状态清除寄存器

位	访问	描述
31	res	Reserved 复位值: 0x0
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
27:17	res	Reserved 复位值: 0x0
16	wo	PPB0_COMP_INT: PPB0 结果比较匹配状态清除 0: 不清除 1: 清除 复位值: 0x0
15:6	res	Reserved 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
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

位	访问	描述
		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

10.5.7 ADC 转换单元触发溢出寄存器 (ADC_TRIG_OVF)

偏移地址: 0x0030

表 93 转换单元触发溢出寄存器

位	访问	描述
31:6	res	Reserved 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
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

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

偏移地址: 0x0034

表 94 转换单元触发溢出清除寄存器

位	访问	描述
31:6	res	Reserved 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 0: 无影响 1: 清除触发事件溢出 复位值: 0x0

10.5.9 ADC 转换单元配置寄存器 (ADC_SOCx) (x=0, 1, 2, 3, 4, 5)

偏移地址: 0x00C0, 0x00C4, 0x00C8, 0x00CC, 0x00D0, 0x00D4

表 95 ADC 转换单元配置寄存器

位	访问	描述
31:24	res	Reserved 复位值: 0x0
23:20	r/w	ADC_CHN_SEL: ADC 转换通道选择 0000: 选择外部通道 0 进行转换 0001: 选择外部通道 1 进行转换 0010: 选择外部通道 2 进行转换 0011: 选择外部通道 3 进行转换 0100: 选择外部通道 4 进行转换 0101: 选择外部通道 5 进行转换 0110: 选择外部通道 6 进行转换 0111: 选择外部通道 7 进行转换

位	访问	描述
		1000: 选择外部通道 8 进行转换 1001: 选择外部通道 9 进行转换 1010: 选择外部通道 10 进行转换 1011: 选择内部通道 PGA0_OUT 进行转换 1100: 选择内部通道 PGA1_OUT 进行转换 1101: 选择内部通道 TS_OUT 进行转换 1110: 选择内部通道 VDD 进行转换 1111: 选择内部通道 VCC 进行转换 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: TIM0 OC1 0001: TIM0 OC2 0010: TIM0 TRGO 0011: TIM1 OC1 0100: TIM2 OC1 0101: TIM8 事件 0 0110: TIM8 事件 1 0111: TIM8 TRGO 1000: 转换单元 0 完成事件 1001: 转换单元 1 完成事件 1010: 转换单元 2 完成事件 1011: 转换单元 3 完成事件 1100: 转换单元 4 完成事件 1101: 转换单元 5 完成事件 1110: 任一转换单元事件 1111: 保留 复位值: 0x0

位	访问	描述
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

10.5.10 ADC PPB0 设置寄存器 (ADC_PPB0_CFG)

偏移地址: 0x0110

表 96 ADC PPB0 设置寄存器

位	访问	描述
31:7	res	Reserved 复位值: 0x0
6:4	r/w	PPB0_SOC_SEL: 转换单元选择 000: 选择转换单元 0 001: 选择转换单元 1 010: 选择转换单元 2 011: 选择转换单元 3 100: 选择转换单元 4 101: 选择转换单元 5 复位值: 0x0
3	res	Reserved 复位值: 0x0
2:1	r/w	ADC_COMP_MOD: ADC 比较模式 00: 结果 < 下限值 01: 结果 > 上限值 10: 结果 >= 下限值 并且 结果 <= 上限值 11: 结果 < 下限值 或者 结果 > 上限值 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

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

偏移地址: 0x011C

表 97 ADC PPB0 比较上限寄存器

位	访问	描述
31:12	res	Reserved 复位值: 0x0
11:0	r/w	PPB0_CMPH: PPB0 比较上限值 0x000: +0 0x001: +1 0x002: +2 ... 0xFFFF: +4095 复位值: 0xFFFF

10.5.12 ADC PPB0 比较下限寄存器 (ADC_PPBO_COMPL)

偏移地址: 0x0120

表 98 ADC PPB0 比较下限寄存器

位	访问	描述
31:12	res	Reserved 复位值: 0x0
11:0	r/w	PPB0_CMPL: PPB0 比较下限值 0x000: +0 0x001: +1 0x002: +2 ... 0xFFF: +4095 复位值: 0x0

10.5.13 ADC 转换结果寄存器 (ADC_RESULTx) (x=0, 1, 2, 3, 4, 5)

偏移地址: 0x0040, 0x0044, 0x0048, 0x004C, 0x0050, 0x0054

表 99 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

10.5.14 ADC 模拟通道使能寄存器 (ADC_CHNEN)

偏移地址: 0x00B0

表 100 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_CHN10_EN: ADC 模拟通道 10 使能 0: 通道关闭 1: 通道使能 复位值: 0x0
9	r/w	ADC_CHN9_EN: ADC 模拟通道 9 使能 0: 通道关闭 1: 通道使能 复位值: 0x0
8	r/w	ADC_CHN8_EN: ADC 模拟通道 8 使能 0: 通道关闭 1: 通道使能 复位值: 0x0

位	访问	描述
7	r/w	ADC_CHN7_EN: ADC 模拟通道 7 使能 0: 通道关闭 1: 通道使能 复位值: 0x0
6	r/w	ADC_CHN6_EN: ADC 模拟通道 6 使能 0: 通道关闭 1: 通道使能 复位值: 0x0
5	r/w	ADC_CHN5_EN: ADC 模拟通道 5 使能 0: 通道关闭 1: 通道使能 复位值: 0x0
4	r/w	ADC_CHN4_EN: ADC 模拟通道 4 使能 0: 通道关闭 1: 通道使能 复位值: 0x0
3	r/w	ADC_CHN3_EN: ADC 模拟通道 3 使能 0: 通道关闭 1: 通道使能 复位值: 0x0
2	r/w	ADC_CHN2_EN: ADC 模拟通道 2 使能 0: 通道关闭 1: 通道使能 复位值: 0x0
1	r/w	ADC_CHN1_EN: ADC 模拟通道 1 使能 0: 通道关闭 1: 通道使能 复位值: 0x0
0	r/w	ADC_CHN0_EN: ADC 模拟通道 0 使能 0: 通道关闭 1: 通道使能 复位值: 0x0

11 高级定时器 (TIMP)

11.1 概述

该高级定时器 (TIMP) 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。系统中有多个定时器，该高级定时器以 TIM8 编号以示区分。定时器可用于多种用途：比较输出、PWM 输出或带死区插入的互补 PWM 输出。

11.2 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)
- 3 个独立通道功能，可用于
 - 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

11.3 模块框图

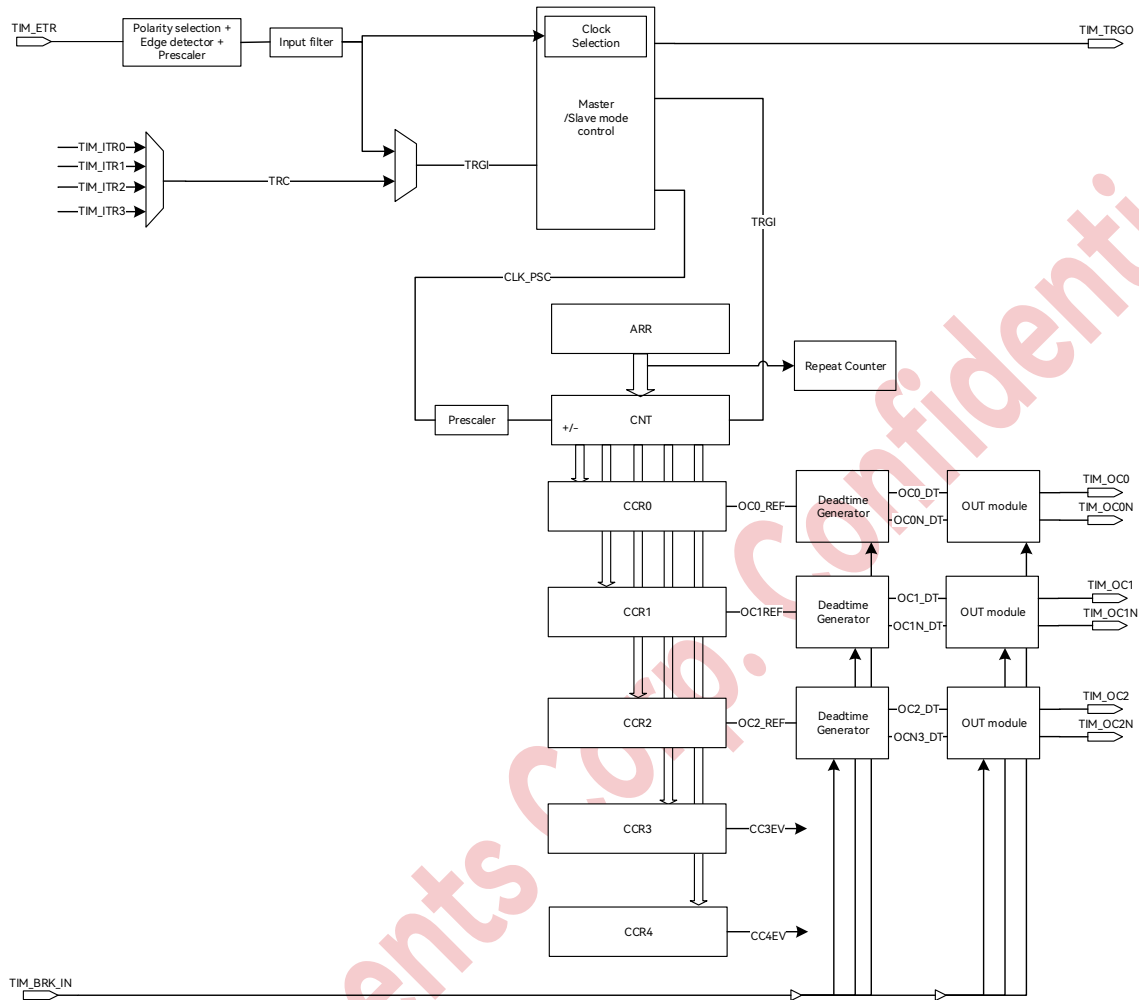


图 24 高级定时器 TIMP 总体模块框图

11.4 功能描述

11.4.1 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_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。

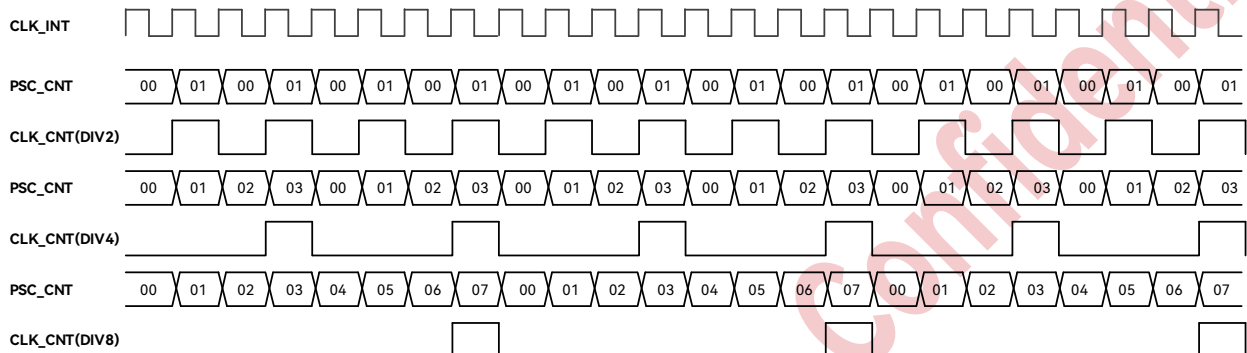


图 25 TIM 预分频器 N 分频时序图

11.4.2 计数器模式

11.4.2.1 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)

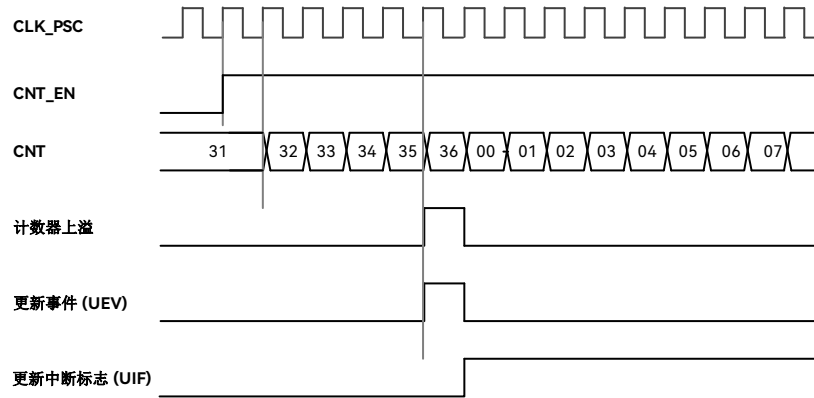


图 26 计数器时序图，1 分频内部时钟

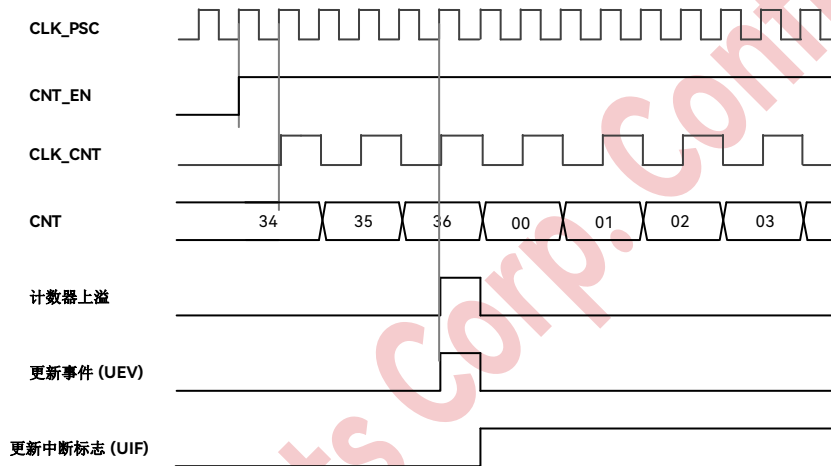


图 27 计数器时序图，2 分频内部时钟

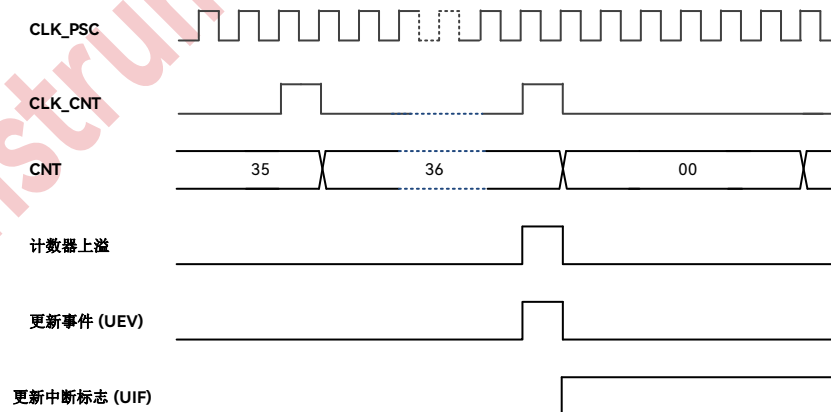


图 28 计数器时序图，N 分频内部时钟

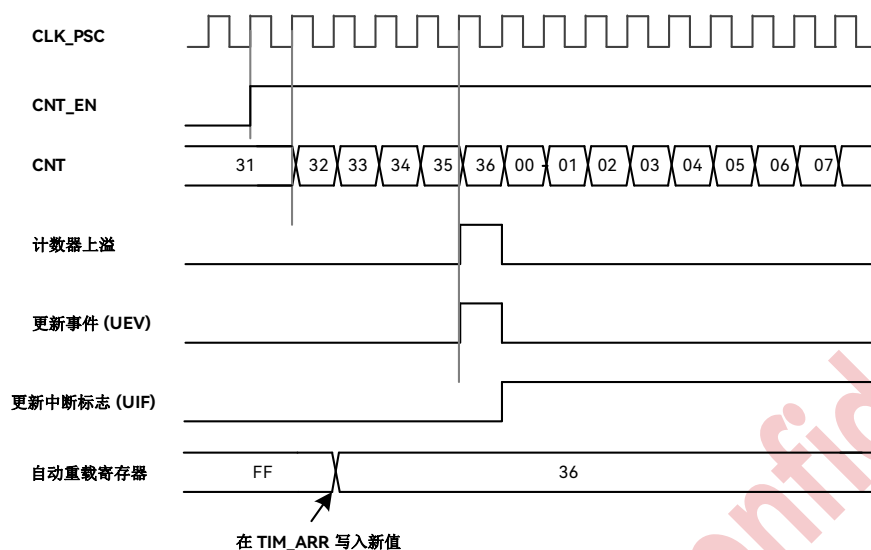


图 29 计数器时序图，ARPE=0 时更新事件 (TIM_ARR 未预装载)

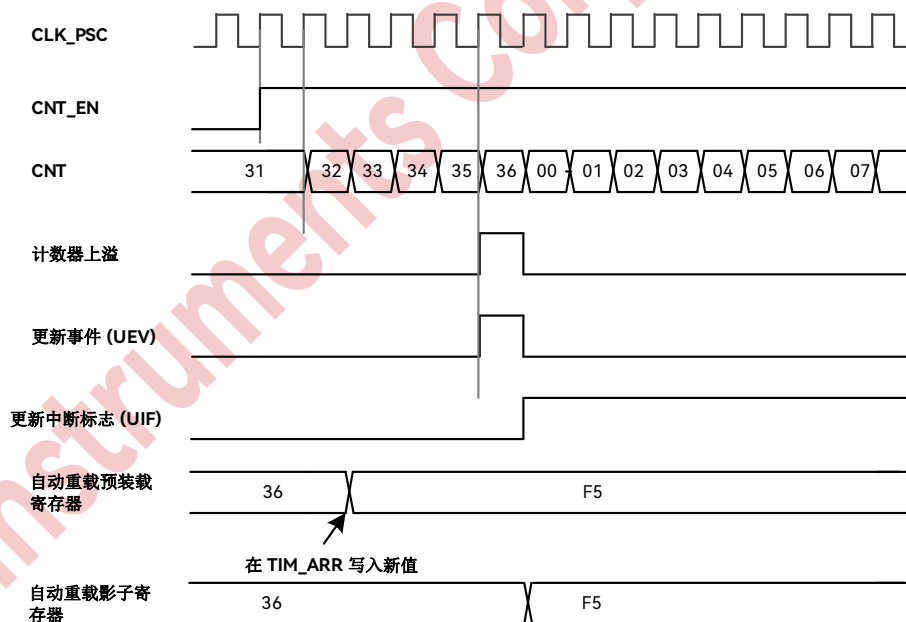


图 30 计数器时序图，ARPE=1 时更新事件 (TIM_ARR 预装载)

11.4.2.2 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。

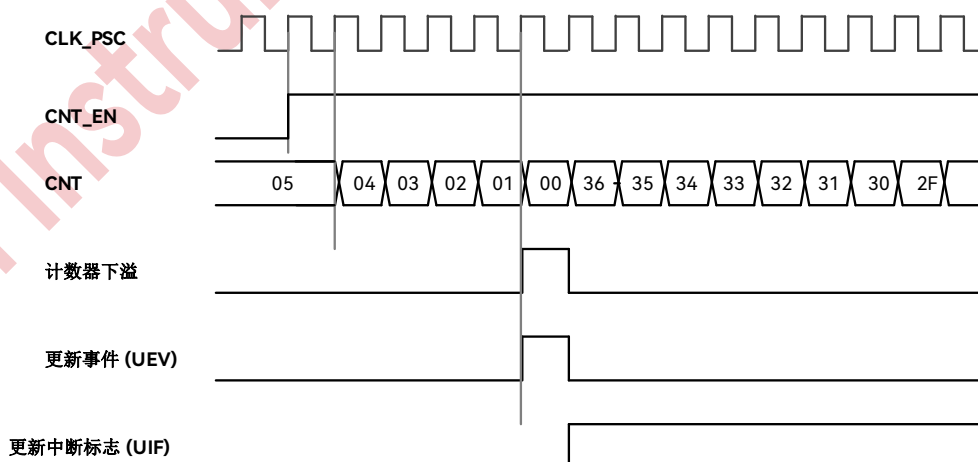


图 31 计数器时序图，1 分频内部时钟

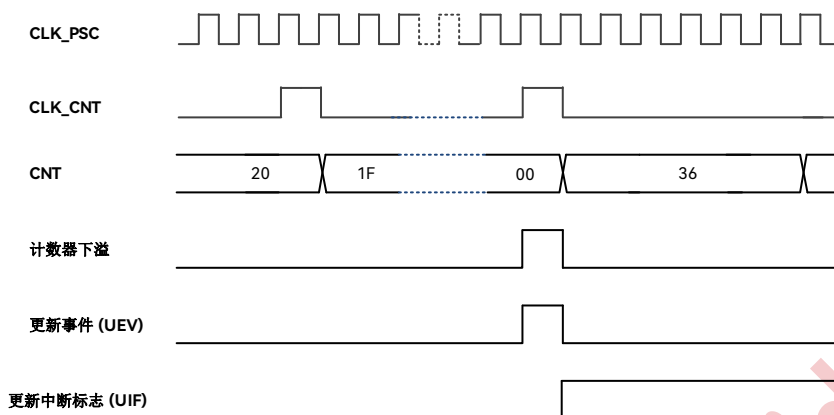


图 32 计数器时序图，N 分频内部时钟

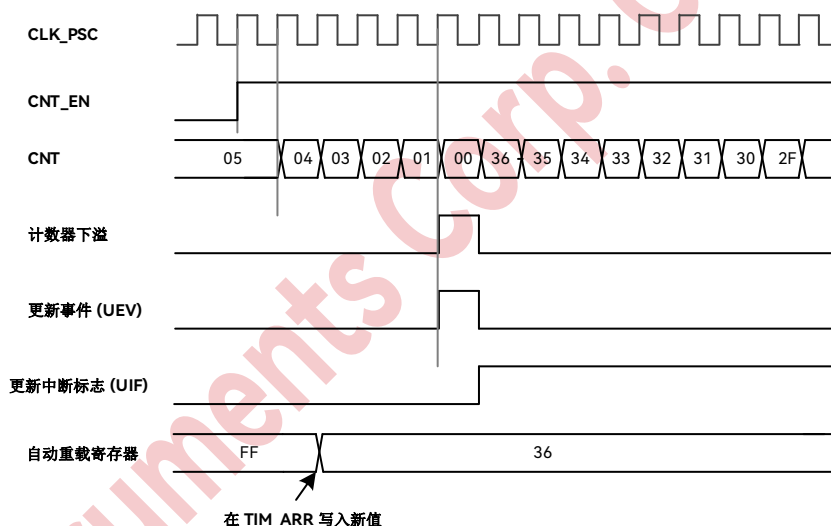


图 33 计数器时序图，未使用重复计数器时更新事件

11.4.2.3 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。

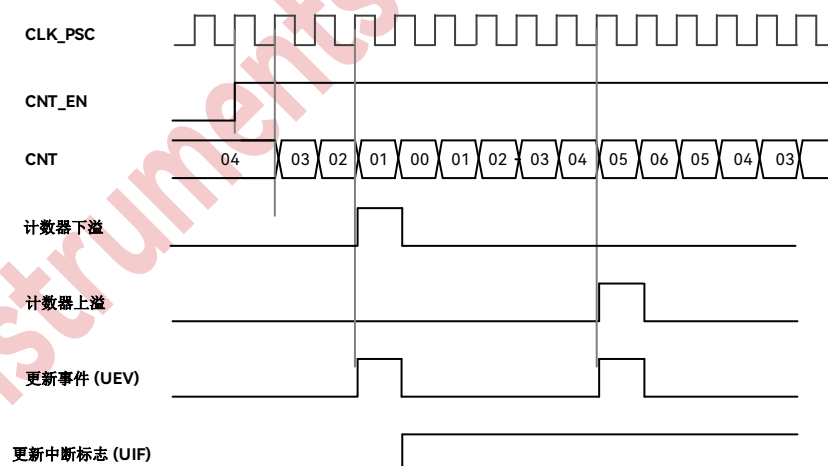


图 34 计数器时序图，1 分频内部时钟，TIM_ARR=0x6

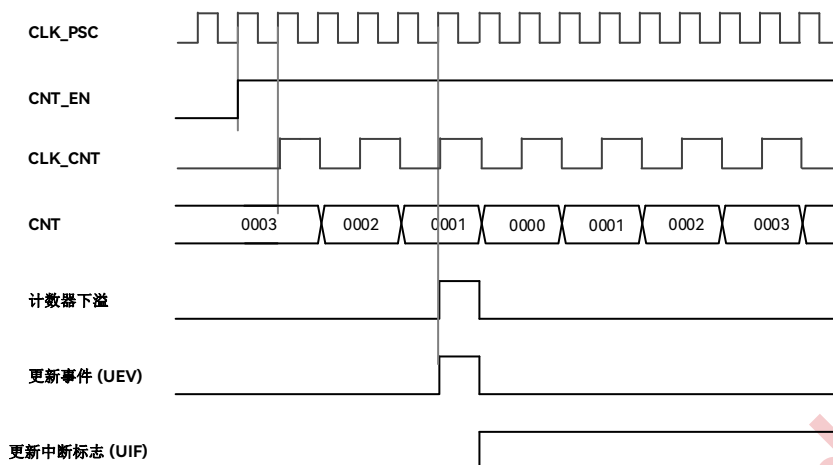


图 35 计数器时序图，2 分频内部时钟

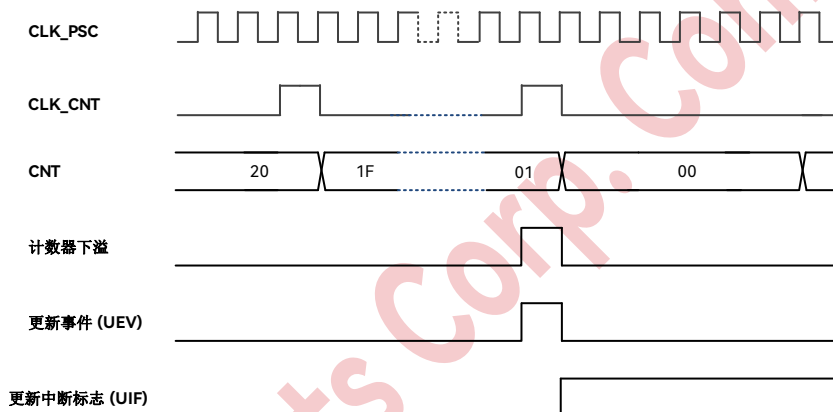


图 36 计数器时序图，N 分频内部时钟

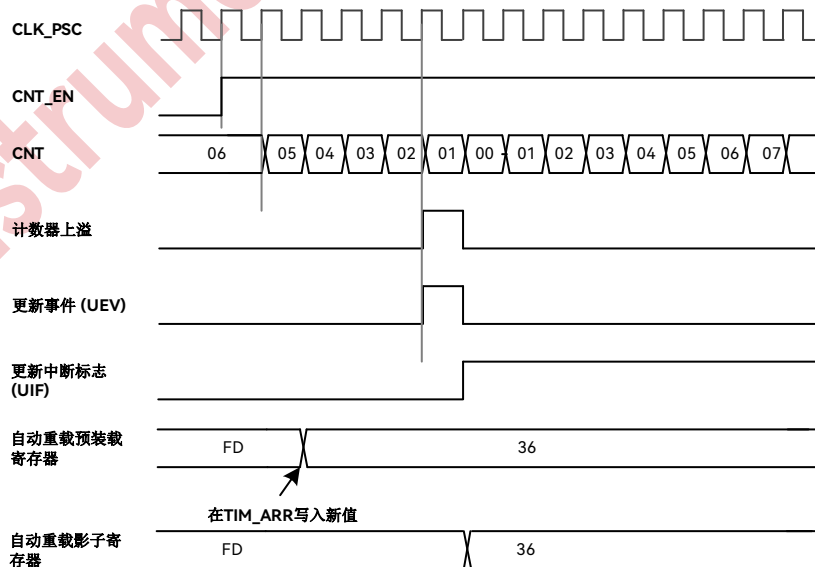
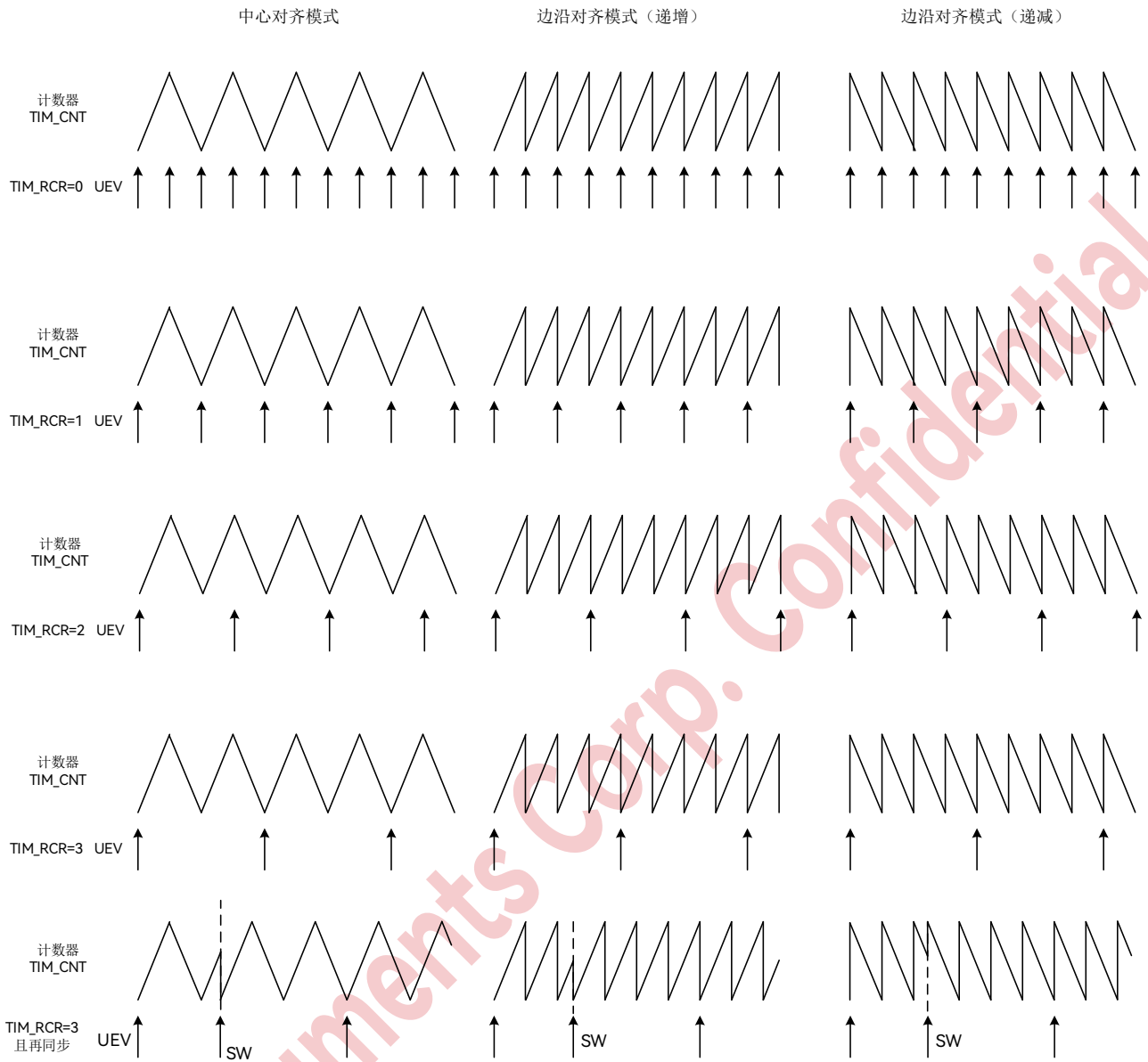


图 37 计数器时序图，ARPE=1 时的更新事件（计数器下溢）



UEV 更新事件：预装载寄存器转移到活动寄存器并产生更新中断

图 38 不同模式和 TIM_RCR 寄存器设置下的更新频率示例

11.4.3 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

11.4.3.1 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。

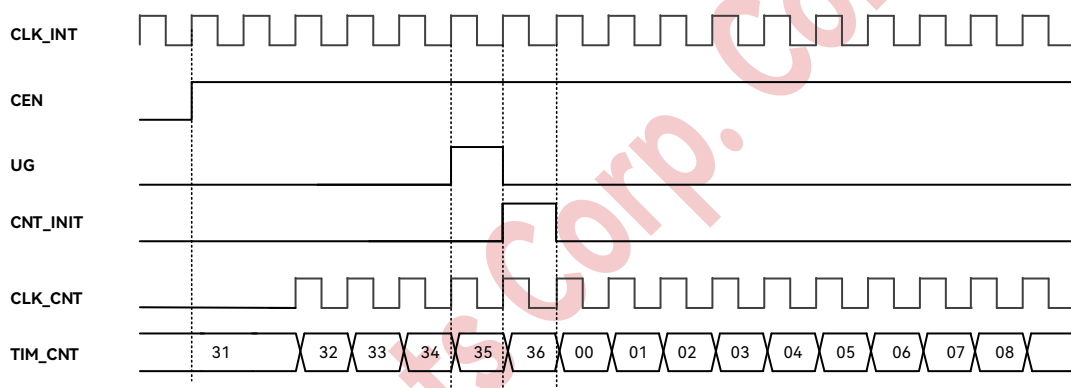


图 39 正常模式下的控制电路 1 分频内部时钟

11.4.3.2 外部时钟源模式

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。

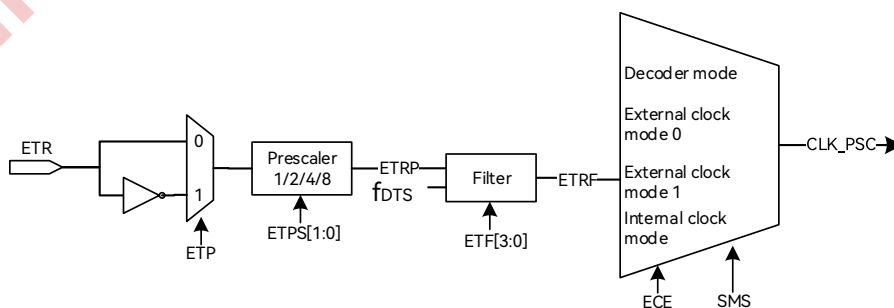


图 40 外部触发输入模块

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 ETRF[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；
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。

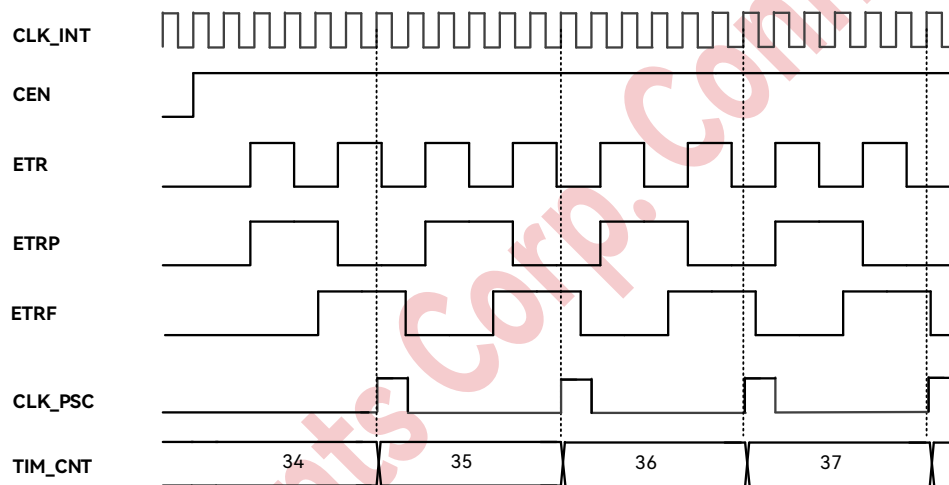


图 41 外部时钟模式 1 下的控制电路

11.4.4 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。

11.4.5 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_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。

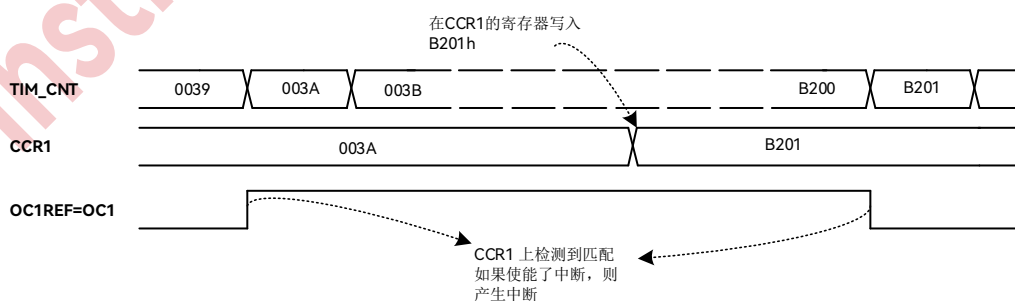


图 42 输出比较模式 (翻转 OC1)

11.4.6 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 0) 或 111 (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_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 0 或 1 下，TIM_CNT 始终与 TIM_CCRx 进行比较，以确定是 $TIM_CCRx < TIM_CNT$ 还是 $TIM_CNT \leq TIM_CCRx$ (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。

11.4.6.1 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 1 为例。只要 $TIM_CNT < TIM_CCRx$ ，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)。

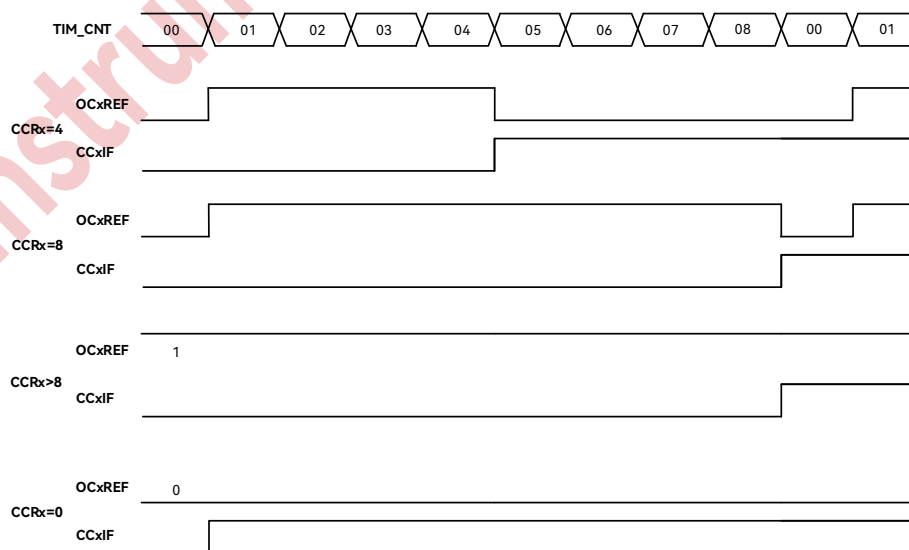


图 43 边沿对齐模式的 PWM 波形 (ARR=8)

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 0 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_CCRx 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。

11.4.6.2 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 (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 1 下，当计数器递减计数时，比较标志置 1。

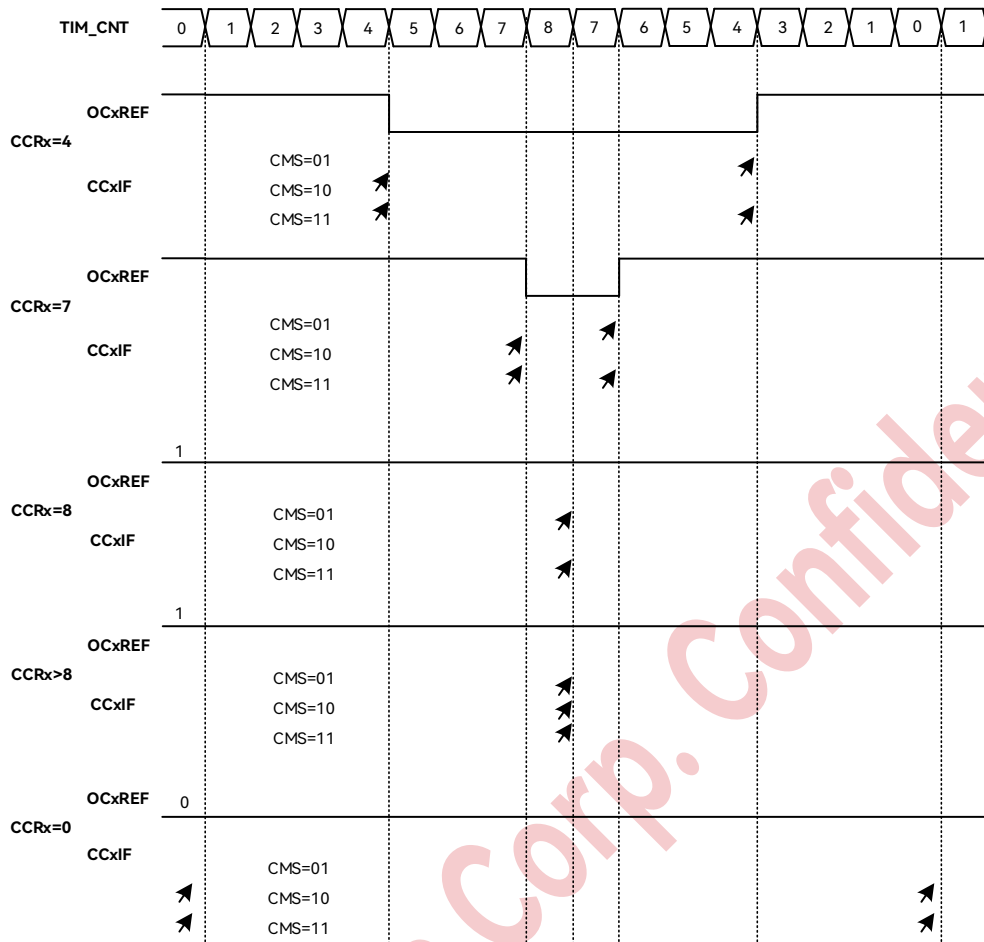


图 44 中心对齐模式 PWM 波形 (ARR=8)

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。

11.4.7 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、OSSI 和 OSSR 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)

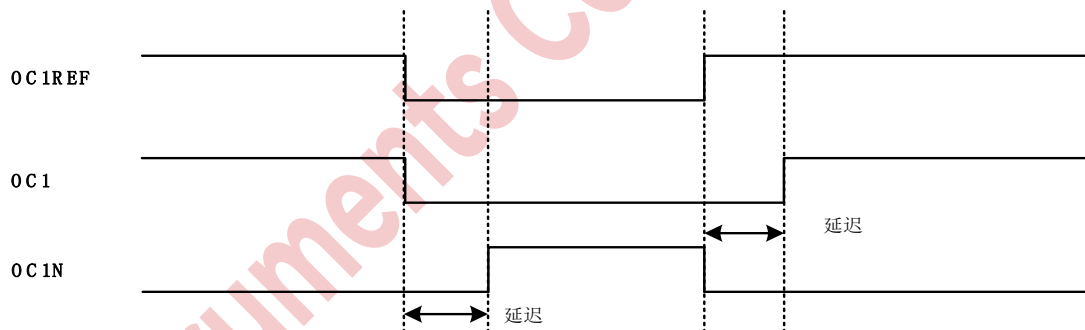


图 45 带死区插入的互补输出

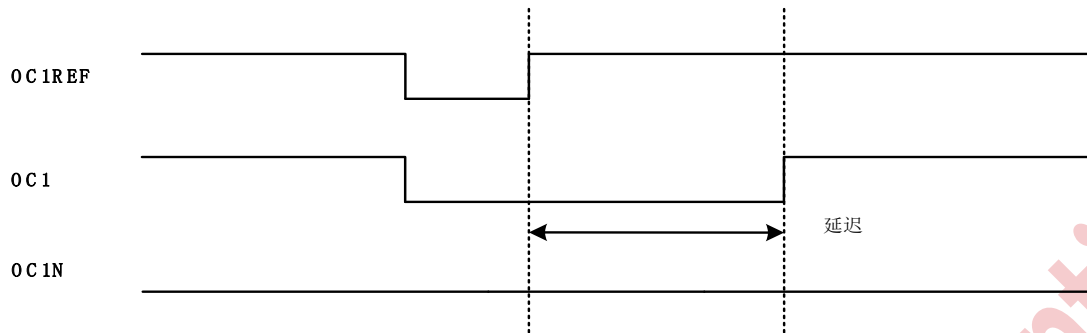


图 46 延迟时间大于负脉冲宽度的死区波形

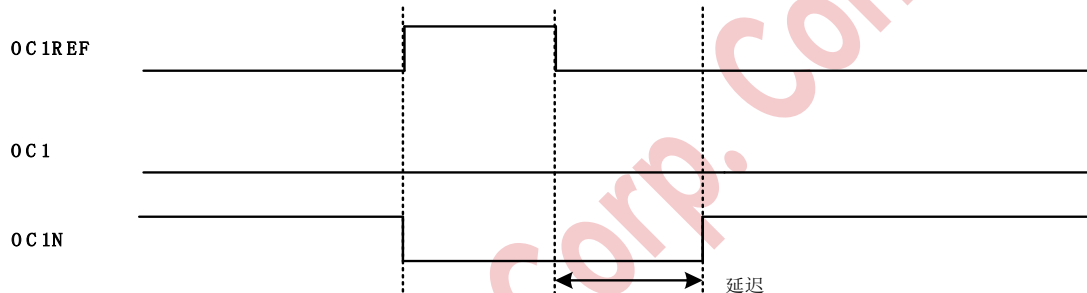


图 47 延迟时间大于正脉冲宽度的死区波形

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。

注意：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。

11.4.8 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 和 OISxN 位) 修改输

出使能信号和无效电平。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_CR 寄存器 OISx 位中编程的电平驱动每个输出通道。
- 使用互补输出时：
 - 以 OISx 和 OISxN 位中编程的电平驱动输出。请注意，MOE 进行再同步，因此死区的持续时间会比通常情况长一些（约 2 个定时器时钟周期）。
- 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。

注意：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。

断路源支持多种输入，配备相应的源使能位（位于 TIM_BSR1），而总的使能位 BKE 位于 TIM_BDTR 寄存器中。断路有以下三种生成方案：

- 使用外部 IO BRK 或者 EBUS IO 输入生成，该输入具有可编程极性
- 使用 ADC 事件 (ADC_EV_TIM)
- 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。

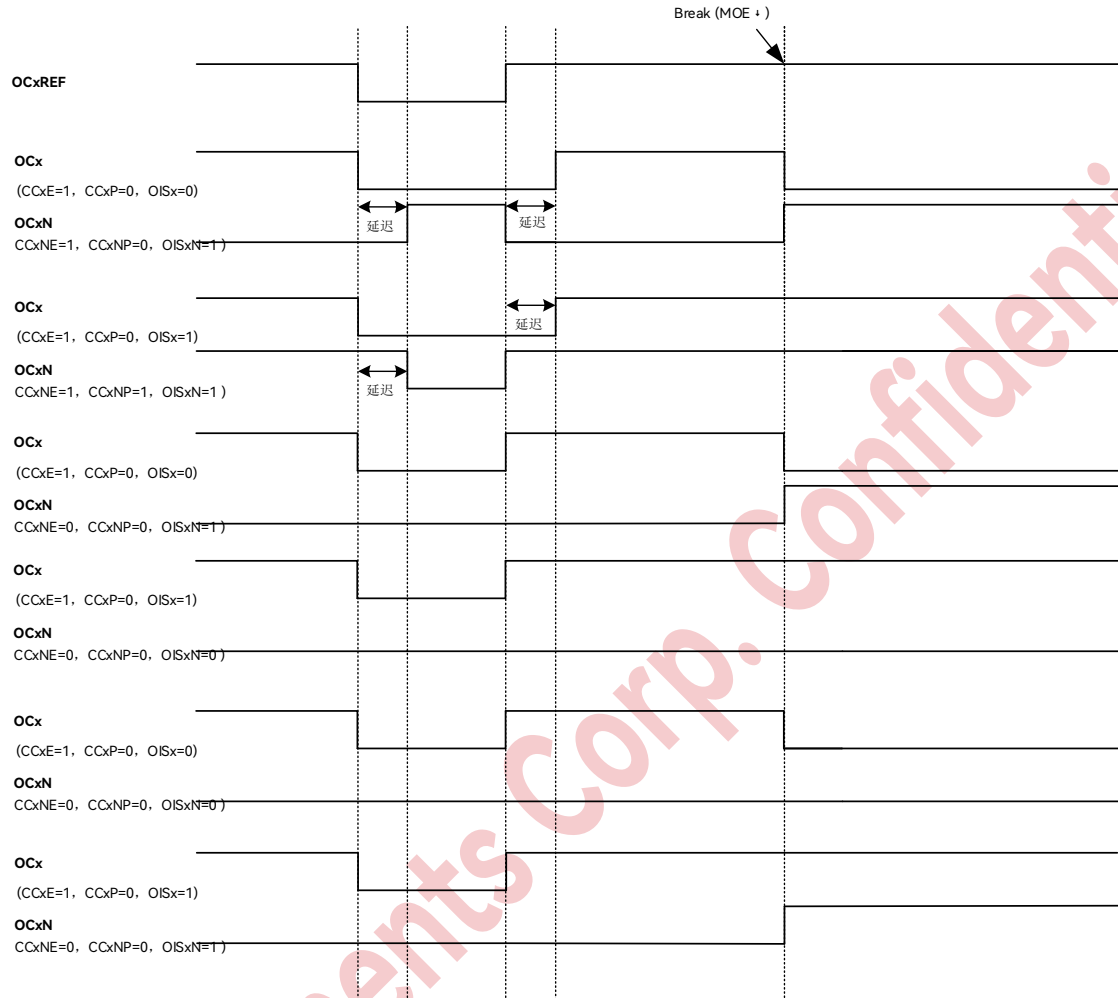


图 48 输出的断路响应行为

外部断路源包括外部 IO 输入断路由 EBUS 输入至 TIM8, (TIM8_EBUS_IO_SEL 用于选择输入通道并支持极性可编程), 专用断路源输入, CMP 输出以及 ADC 事件 (ADC_EV_TIM), 具体原理如下图所示。

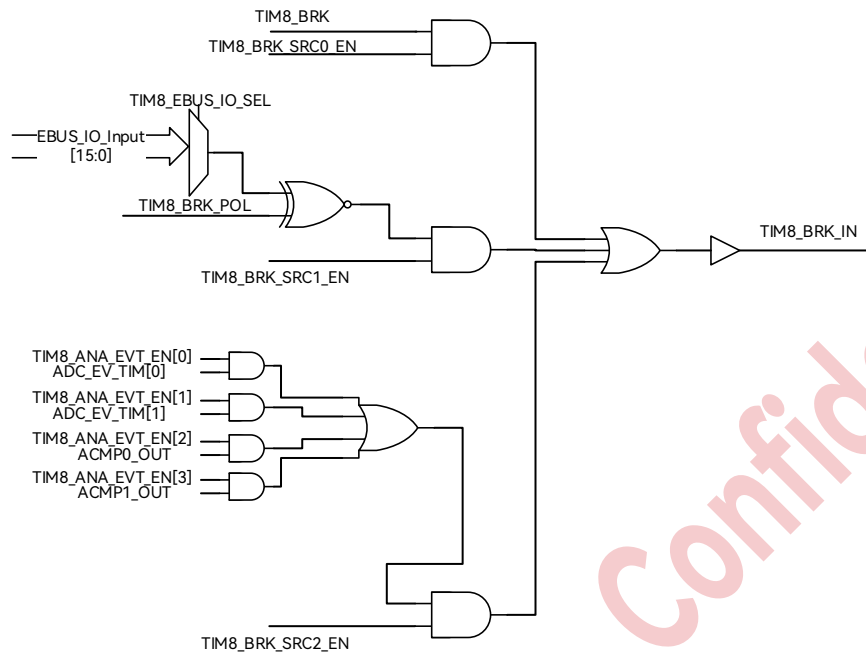


图 49 定时器断路源示意图

11.4.9 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：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。

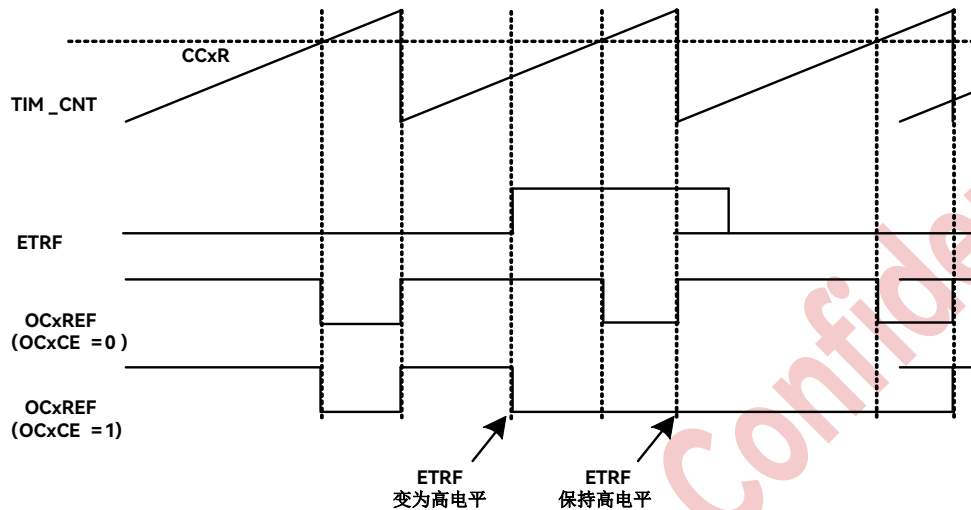


图 50 清除 TIM 的 OCxREF

注意：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。

11.4.10 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。

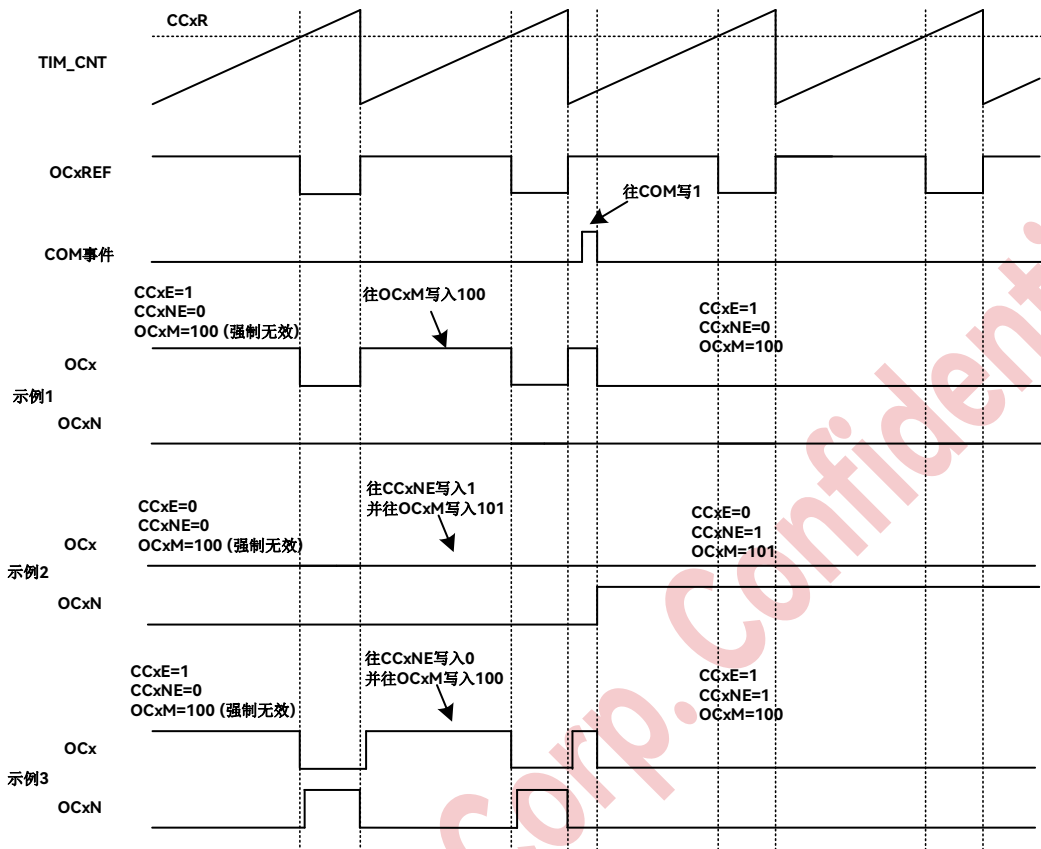


图 51 COM 事件生成 6 步 PWM 的示例

11.4.11 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_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， $CNT < CCRx \leq ARR$ (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 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。使用 TI1FP1 作为触发：

1. 在 TIM_CCMR 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_MCR 寄存器中写入 TS=110, 将 TI1FP1 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 TI1FP1 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_CCR0 寄存器的值定义;
- t_{PULSE} 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_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_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_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_Tlx 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。

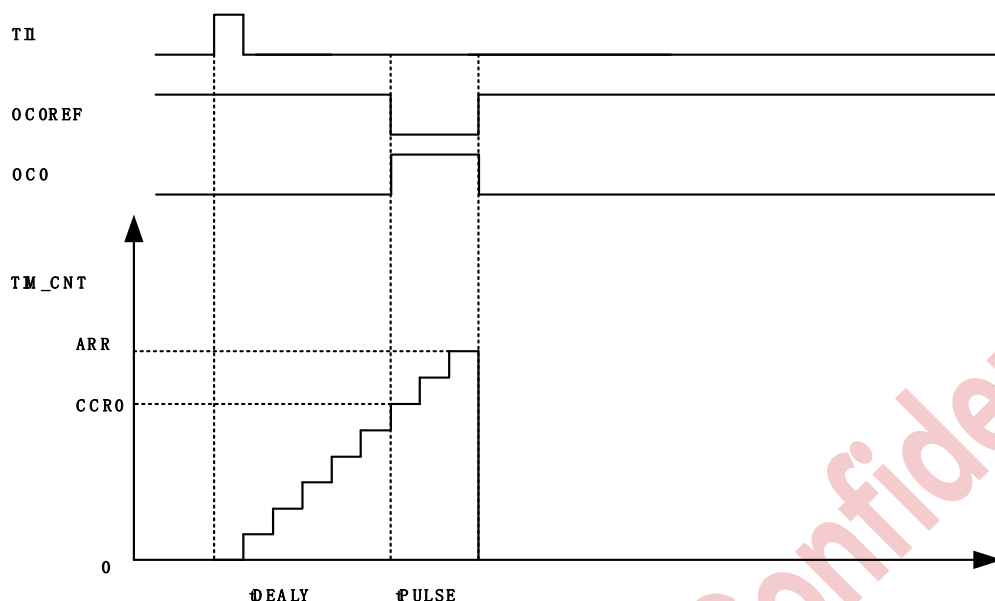


图 52 单脉冲模式示例

11.4.12 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。

在接口定时器的捕获/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。

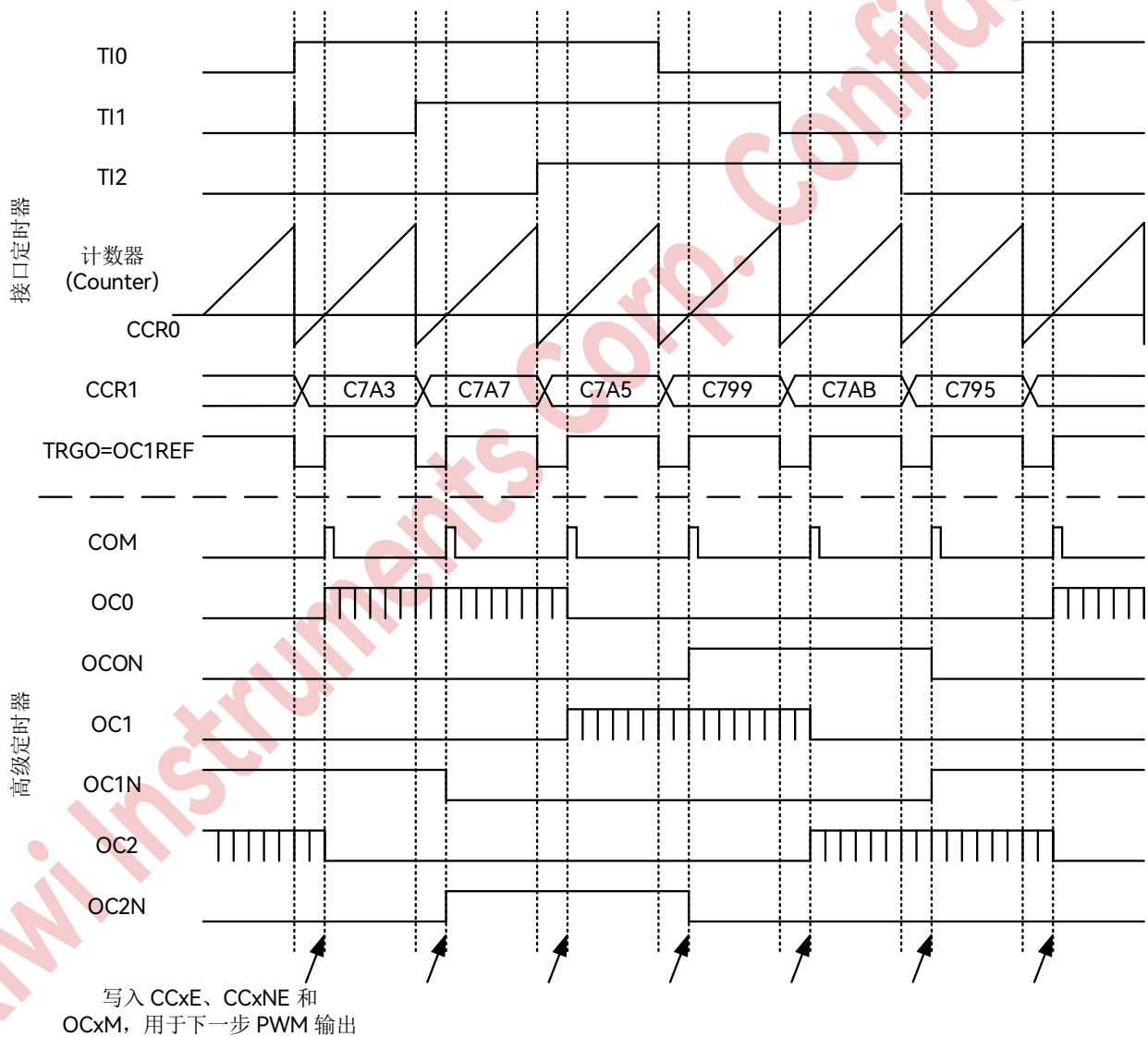


图 53 霍尔传感器接口的示例

11.4.13 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。

11.4.13.1 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，ITR0 输入上出现上升沿时，计数器清零：

1. 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=000，选择 ITR0 作为输入源。
2. 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 ITR0 上升沿。当 ITR0 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。ITR0 的上升沿与实际计数器复位之间的延迟是由于 ITR0 输入的重新同步电路引起的。

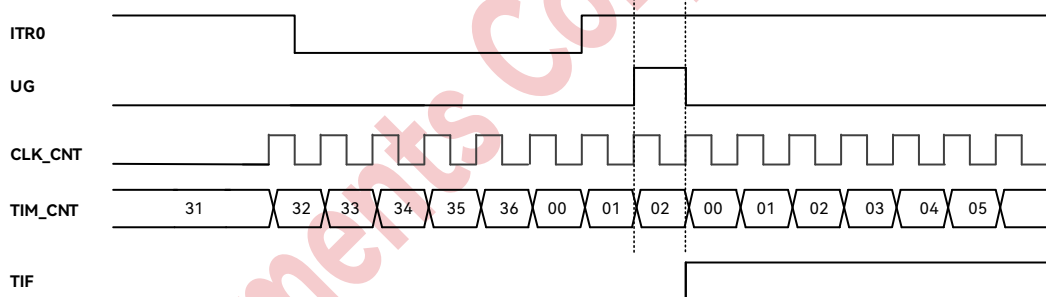


图 54 复位模式下的控制电路

11.4.13.2 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 ITR0 输入为低电平时计数：

1. 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=000，选择 ITR0 作为输入源。
2. 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 TIM1 为低电平，计数器就开始根据内部时钟计数，直到 ITR0 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。ITR0 的上升沿与实际计数器停止之间的延迟是由于 ITR0 输入的重新同步电路引起的。

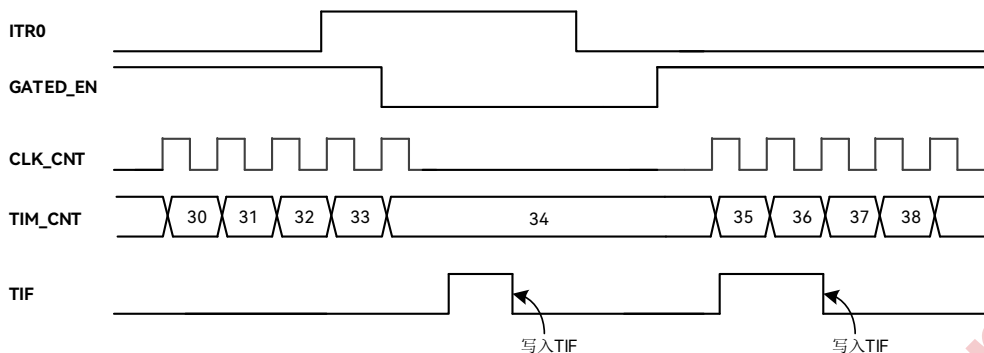


图 55 门控模式下的控制电路

11.4.13.3 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，ITR1 输入上出现上升沿时，计数器便启动：

1. 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=001，选择 ITR1 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。ITR1 的上升沿与实际计数器启动之间的延迟是由于 ITR1 输入的重新同步电路引起的。

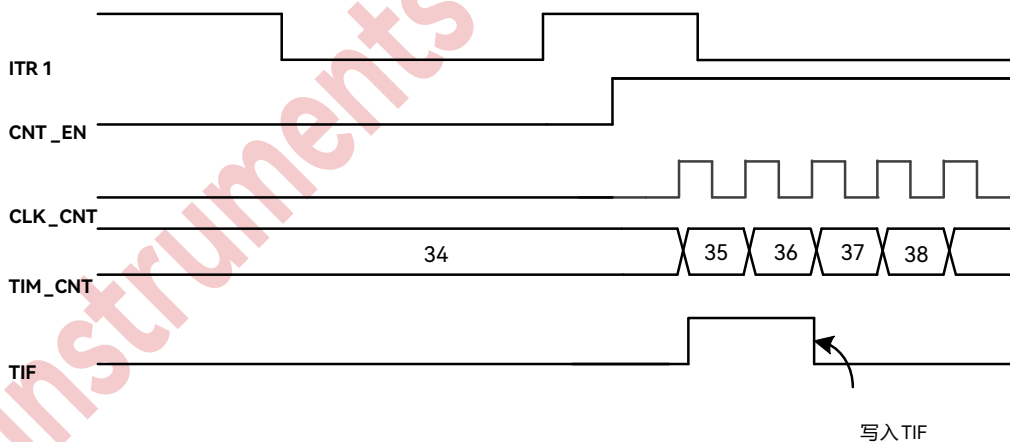


图 56 触发模式下的控制电路

11.4.14 TIMP 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；
- TIM 对应的中断号为：29。

表 101 TIM 中断号

名称	中断号
TIMP_INT	29

11.5 寄存器描述

TIM8 寄存器基地址是 0x4000_1000。下表为 TIM 的各控制寄存器描述。

表 102 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 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 比较模式寄存器 0	32	r/w	0x00000000
0x1C	TIM 比较模式寄存器 1	32	r/w	0x00000000
0x20	TIM 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	TIM 重复计数器寄存器	32	r/w	0x00000000
0x34	TIM 比较寄存器 0	32	r/w	0x00000000
0x38	TIM 比较寄存器 1	32	r/w	0x00000000
0x3C	TIM 比较寄存器 2	32	r/w	0x00000000
0x40	TIM 比较寄存器 3	32	r/w	0x00000000
0x44	TIM 比较寄存器 4	32	r/w	0x00000000
0x48~0x4C	保留	32	r/w	0x00000000
0x50	TIM 断路及死区寄存器	32	r/w	0x00000000
0x54~0xFC	保留	32	RES	0x00000000

注：关于输入复用关系和请查阅设计文档《简易定时器》中相关寄存器描述。

11.5.1 TIM 控制寄存器 0 (TIM_CR0)

偏移地址: 0x0000

表 103 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_CMR _x 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_CMR _x 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_CMR _x 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
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

位	访问	描述
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: ■ 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: ■ 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: 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

11.5.2 TIM 控制寄存器 1 (TIM_CR1)

偏移地址: 0x0004

表 104 TIM 控制寄存器 1

位	访问	描述
31:24	res	Reserved 复位值: 0x0
23	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
22:20	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)。 注: EVNT1 事件源于通道 4 的比较事件有效。 复位值: 0x0
19	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
18:16	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)。 注: EVNT0 事件源于通道 3 的比较事件有效。

位	访问	描述
		复位值: 0x0
15:14	res	Reserved 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 时, 经过死区时间之后 OC0N=0 1: 当 MOE=0 时, 经过死区时间之后 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 时, (如果 OC0N 有效, 则经过死区时间之后) OC0=0 1: 当 MOE=0 时, (如果 OC0N 有效, 则经过死区时间之后) 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	res	Reserved 复位值: 0x0
6:4	r/w	MMS[1:0], 主模式选择 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: 保留

位	访问	描述
		复位值: 0x0
3	res	Reserved 复位值: 0x0
2	r/w	CCUS, 捕获/比较控制更新选择 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: 此位仅对具有互补输出的通道有效。 复位值: 0x0
1	res	Reserved 复位值: 0x0
0	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) (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: 此位仅对具有互补输出的通道有效。 复位值: 0x0

11.5.3 TIM 从属模式控制寄存器 (TIM_SMCR)

偏移地址: 0x0008

表 105 TIM 从属模式控制寄存器

位	访问	描述
31:17	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。 0: 禁止外部时钟模式 1: 使能外部时钟模式。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)。 复位值: 0x0
13:12	r/w	ETPS, 外部触发预分频器

位	访问	描述
		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。 00: 预分频器关闭 01: 2 分频 ETRP 频率 10: 4 分频 ETRP 频率 11: 8 分频 ETRP 频率 复位值: 0x0
11:8	r/w	ETF[3:0], 外部触发滤波器 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: 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
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: 内部触发 0 (ITR0)→TIM0_TRGO 001: 内部触发 1 (ITR1)→TIM1_TRGO 010: 内部触发 2 (ITR2)→TIM2_TRGO 011: 内部触发 3 (ITR3)→保留 100: 保留 101: 保留 110: 保留 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。 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。 复位值: 0x0
3	res	Reserved 复位值: 0x0

位	访问	描述
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——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。 复位值: 0x0

11.5.4 TIM 中断使能寄存器 (TIM_IER)

偏移地址: 0x000C

表 106 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	res	Reserved 复位值: 0x0
3	r/w	C2IE, 比较 2 中断使能 0: 关闭 1: 开启 复位值: 0x0
2	r/w	C1IE, 比较 1 中断使能 0: 关闭 1: 开启 复位值: 0x0
1	r/w	C0IE, 比较 0 中断使能 0: 关闭 1: 开启

位	访问	描述
		复位值: 0x0
0	r/w	UIE, 更新中断使能 0: 关闭 1: 开启 复位值: 0x0

11.5.5 TIM 状态寄存器 (TIM_SR)

偏移地址: 0x0010

表 107 TIM 状态寄存器

位	访问	描述
31:14	res	Reserved 复位值: 0x0
13	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
12	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
11:8	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	r/w	C3IF, 比较 3 中断标志 参照 C1IF 描述

位	访问	描述
		复位值: 0x0
3	r/w	C2IF, 比较 2 中断标志 参照 C0IF 描述 复位值: 0x0
2	r/w	C1IF, 比较 1 中断标志 参照 C0IF 描述 复位值: 0x0
1	r/w	C0IF, 比较 0 中断标志 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, C0IF 位将在计数器发生上溢时变为高电平。 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

11.5.6 TIM 事件生成寄存器 (TIM_EGR)

偏移地址: 0x0014

表 108 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或 DMA 传输事件。 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。 复位值: 0x0
5	r/w	COMG, 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	res	Reserved 复位值: 0x0

位	访问	描述
3	r/w	C2G, 比较 2 生成 参照 C0G 描述 复位值: 0x0
2	r/w	C1G, 比较 1 生成 参照 C0G 描述 复位值: 0x0
1	r/w	C0G, 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 上生成比较事件: 使能时, C0IF 标志置 1 并发送相应的中断。 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

11.5.7 TIM 比较模式寄存器 0 (TIM_CMRO)

偏移地址: 0x0018

表 109 TIM 比较模式寄存器 0

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, 比较 1 通道选择 CC1 通道固定为输出 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 ETRF 输入影响 1: ETRF 输入上检测到高电平时, OC0REF 立即清零 复位值: 0x0

位	访问	描述
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。 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。 0011: 翻转——TIM_CNT=TIM_CCR0 时, OC0REF 发生翻转。 0100: 强制变为无效电平——OC0REF 强制变为低电平。 0101: 强制变为有效电平——OC0REF 强制变为高电平。 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”)。 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。 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_CCR0, 通道 0 便为无效状态 (OC0REF=“0”), 否则为有效状态 (OC0REF=“1”)。 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_CCR0, 通道 0 便为有效状态, 否则为无效状态。 其他: 保留 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_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。 复位值: 0x0
3	r/w	OC0PE, 输出比较 0 预装载使能 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。 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。 注: - 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。 复位值: 0x0

位	访问	描述
2	r/w	OC0FE, 输出比较 0 快速使能 此位用于加快触发输入事件对 CC0 输出的影响。 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。 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。 复位值: 0x0
1:0	r/w	CC0S, 比较 0 通道选择 CC0 通道固定为输出 复位值: 0x0

11.5.8 TIM 比较模式寄存器 1 (TIM_CMR1)

偏移地址: 0x001C

表 110 TIM 比较模式寄存器 1

位	访问	描述
31:28	res	Reserved 复位值: 0x0
27	r/w	OC4PE, 输出比较 4 预装载使能 参照 OC2PE 相关描述 复位值: 0x0
26	res	Reserved 复位值: 0x0
25:24	r/w	CC4S, 比较 4 通道选择 CC4 通道固定为输出 复位值: 0x0
23:20	res	Reserved 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	res	Reserved 复位值: 0x0
17:16	r/w	CC3S, 比较 3 通道选择 CC3 通道固定为输出 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

位	访问	描述
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。 0011: 翻转——TIM_CNT=TIM_CCR2 时, OC2REF 发生翻转。 0100: 强制变为无效电平——OC2REF 强制变为低电平。 0101: 强制变为有效电平——OC2REF 强制变为高电平。 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”)。 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_CCR2, 通道 2 便为有效状态, 否则为无效状态。 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_CCR2, 通道 2 便为无效状态 (OC2REF=“0”), 否则为有效状态 (OC2REF=“1”)。 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_CCR2, 通道 2 便为有效状态, 否则为无效状态。 其他: 保留 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。 复位值: 0x0
3	r/w	OC2PE, 输出比较 2 预装载使能 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。 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。 注: - 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。 - 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。 复位值: 0x0

位	访问	描述
2	r/w	OC2FE, 输出比较 2 快速使能 此位用于加快触发输入事件对 CC 输出的影响。 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。 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, OCFE 才会起作用。 复位值: 0x0
1:0	r/w	CC2S, 比较 2 通道选择 CC2 通道固定为输出 复位值: 0x0

11.5.9 TIM 比较模式使能寄存器 (TIM_CER)

偏移地址: 0x0020

表 111 TIM 捕获/比较模式使能寄存器

位	访问	描述
31:17	res	Reserved 复位值: 0x0
16	r/w	C4E, 比较 4 使能 参照 C0E 描述 复位值: 0x0
15:13	res	Reserved 复位值: 0x0
12	r/w	C3E, 比较 3 使能 参照 C0E 描述 复位值: 0x0
11	r/w	C2NP, 比较 2 互补输出极性 参照 C0NP 描述 复位值: 0x0
10	r/w	C2NE, 比较 2 互补输出使能 参照 C0NE 描述 复位值: 0x0
9	r/w	C2P, 比较 2 输出极性 参照 C1P 描述 复位值: 0x0
8	r/w	C2E, 比较 2 输出使能 参照 C0E 描述 复位值: 0x0
7	r/w	C1NP, 比较 1 互补输出极性 参照 C0NP 描述 复位值: 0x0
6	r/w	C1NE, 比较 1 互补输出使能 参照 C0NE 描述 复位值: 0x0

位	访问	描述
5	r/w	C1P, 比较 1 输出极性 参照 C0P 描述 复位值: 0x0
4	r/w	C1E, 比较 1 输出使能 参照 C0E 描述 复位值: 0x0
3	r/w	C0NP, 比较 0 互补输出极性 0: OC0N 高电平有效。 1: OC0N 低电平有效。 注: 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	C0NE, 比较 0 互补输出使能 0: 关闭——OC0N 未激活。OC0N 电平是 MOE、OIS0、OIS0N 和 CC0E 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、OIS0N 和 CC0E 位。 复位值: 0x0
1	r/w	C0P, 比较 0 输出极性 0: OC0 高电平有效 1: OC0 低电平有效 注: 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。 复位值: 0x0
0	r/w	C0E, 比较 0 输出使能 0: 关闭——OC0 未激活。OC0 电平是 MOE、OIS0、OIS0N 和 CC0NE 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、OIS0N 和 CC0NE 位。 复位值: 0x0

11.5.10 TIM 计数器 (TIM_CNT)

偏移地址: 0x0024

表 112 TIM 计数器

位	访问	描述
31:16	res	Reserved 复位值: 0x0
15:0	r/w	CNT[15:0]: 计数器值 复位值: 0x0

11.5.11 TIM 预分频器 (TIM_PSC)

偏移地址: 0x0028

表 113 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

11.5.12 TIM 自动重载寄存器 (TIM_ARR)

偏移地址: 0x002C

表 114 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

11.5.13 TIM 重复计数器寄存器 (TIM_RCR)

偏移地址: 0x0030

表 115 TIM 自动重载寄存器

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 该控制位为更新中断的预分频值。实现方式是与 REP_CNT 相关的减计数器每次计数到 0 时, 都将生成一个重复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: ■ 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

11.5.14 TIM 比较寄存器 0 (TIM_CCR0)

偏移地址: 0x0034

表 116 TIM 比较寄存器 0

位	访问	描述
31:16	r/w	ACR0[15:0], 通道 0 非对称比较值 ACR0 是通道 0 非对称比较寄存器 0 的预装载值。如果没有通过 TIM_CMR 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。 注: 该比较值仅用于在非对称 PWM 模式下计数器向下计数时通道 0 的比较点。 复位值: 0x0
15:0	r/w	CCR0[15:0], 通道 0 比较值 CCR0 是通道 0 比较寄存器的预装载值。如果没有通过 TIM_CMR 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比较寄存器 0)。实际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。 复位值: 0x0

11.5.15 TIM 比较寄存器 1 (TIM_CCR1)

偏移地址: 0x0038

表 117 TIM 比较寄存器 1

位	访问	描述
31:16	r/w	ACR1[15:0], 通道 1 非对称比较值 ACR1 是通道 1 非对称比较寄存器的预装载值。如果没有通过 TIM_CMR 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。 注: 该比较值仅用于在非对称 PWM 模式下计数器向下计数时通道 1 的比较点。 复位值: 0x0
15:0	r/w	CCR1[15:0], 通道 1 比较值 CCR1 是通道 1 比较寄存器的预装载值。如果没有通过 TIM_CMR 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比较寄存器 1)。实际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。 复位值: 0x0

11.5.16 TIM 比较寄存器 2 (TIM_CCR2)

偏移地址: 0x003C

表 118 TIM 比较寄存器 2

位	访问	描述
31:16	r/w	CCR2[15:0], 通道 2 非对称比较值 CCR2 是通道 2 非对称比较寄存器的预装载值。如果没有通过 TIM_CMR 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。 注: 该比较值仅用于在非对称 PWM 模式下计数器向下计数时通道 2 的比较点。 复位值: 0x0
15:0	r/w	CCR2[15:0], 通道 2 比较值 CCR2 是通道 2 捕获/比较寄存器的预装载值。如果没有通过 TIM_CMR 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比较寄存器 2)。实际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。 复位值: 0x0

11.5.17 TIM 比较寄存器 3 (TIM_CCR3)

偏移地址: 0x0040

表 119 TIM 比较寄存器 3

位	访问	描述
31:17	res	Reserved 复位值: 0x0
16:0	r/w	CCR3[16:0], 通道 3 比较值 CCR3 是通道 3 比较寄存器的预装载值。如果没有通过 TIM_CMR 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比较寄存器 3)。 注: 1. 该比较值是通道 3 用于与 TIM_CNT 比较产生事件 EVNT0。 2. 该比较值在边沿模式下仅有低 16 位有效。而中心对齐模式启用时, 所设置值: $0 \leq CCR3 < (2 * (ARR + 1))$。 复位值: 0x0

11.5.18 TIM 比较寄存器 4 (TIM_CCR4)

偏移地址: 0x0044

表 120 TIM 比较寄存器 4

位	访问	描述
31:17	res	Reserved 复位值: 0x0
16:0	r/w	CCR4[16:0], 通道 4 比较值 CCR3 是通道 4 比较寄存器的预装载值。如果没有通过 TIM_CMR 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比较寄存器 4)。 注: 1. 该比较值是通道 4 用于与 TIM_CNT 比较产生事件 EVNT1。 2. 该比较值在边沿模式下仅有低 16 位有效。而最高位用于中心对齐模式, 所设置值: $0 \leq \text{CCR4} < (2 * (\text{ARR} + 1))$ 。 复位值: 0x0

11.5.19 TIM 断路及死区寄存器 (TIM_BDTR)

偏移地址: 0x0050

表 121 TIM 断路及死区寄存器

位	访问	描述
30:16	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_CER 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_BCR1, 0x40001408) TIM8_BRK_POL 描述 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: 编程了 LOCK (TIM_BDTR 寄存器中的 LOCK 位) 级别 1 后, 此位即无法修改。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

位	访问	描述
11	ro	OSSR, 运行模式下的关闭状态选择 固定为 0x0
10	ro	OSSI, 空闲模式下的关闭状态选择 固定为 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_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。 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。 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。 注: 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 $tdtg=t_{DTS}$ 。 DTG[7:5]=10X: $DT=(64+DTG[5:0])*t_{dtg}$, 其中 $tdtg=2*t_{DTS}$ 。 DTG[7:5]=110: $DT=(32+DTG[4:0])*t_{dtg}$, 其中 $tdtg=8*t_{DTS}$ 。 DTG[7:5]=111: $DT=(32+DTG[4:0])*t_{dtg}$, 其中 $tdtg=16*t_{DTS}$ 。 例如, 如果 $t_{DTS}=125ns$ (8MHz), 则可能的死区值为: 0 到 15875ns (步长为 125ns) 16 μs 到 31750ns (步长为 250ns) 32 μs 到 63 μs (步长为 1 μs) 64 μs 到 126 μs (步长为 2 μs) 注: 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

12 简版定时器 (TIMS)

12.1 概述

该简版定时器 (TIM) 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。

12.2 主要功能

- 16 位递增自动重载计数器
- 支持 1/2/4/8/16/32/64/128/256/512/1024 计数器时钟预分频 (可运行时修改)
- 通道配置
 - 支持 1 对互补通道 (TIM0)
 - 支持 1 单通道 (TIM0/1/2)
- 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 (仅 TIM0)
- 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

12.3 模块框图

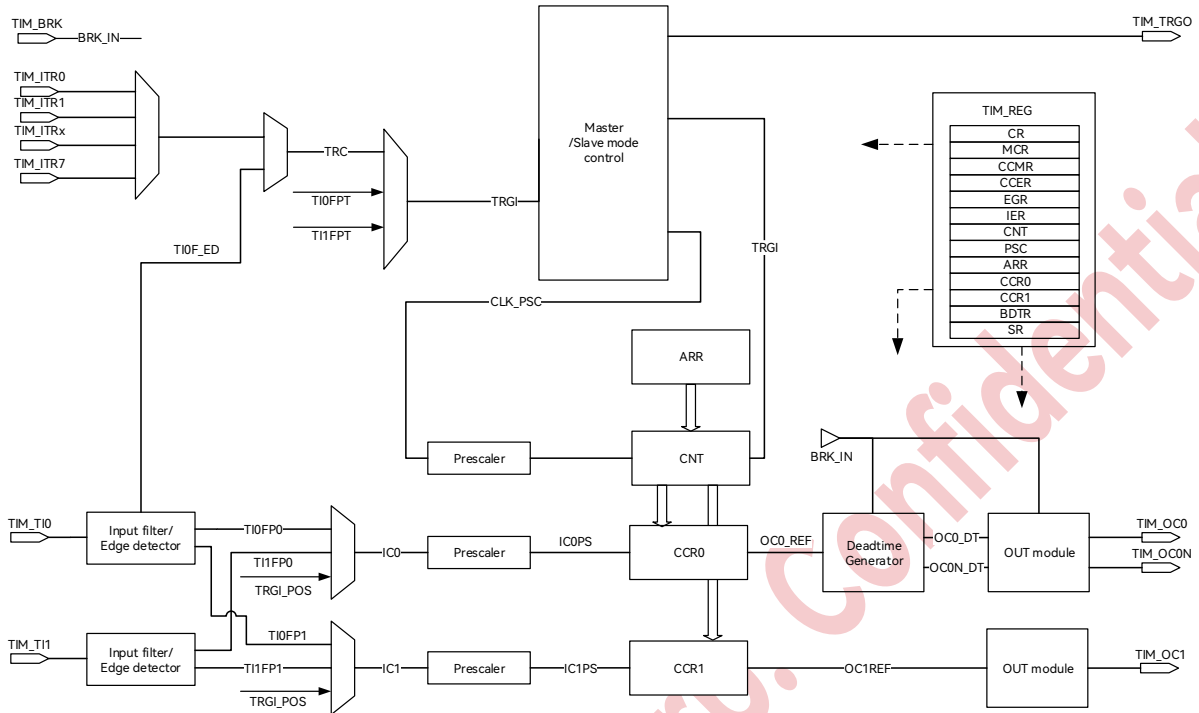


图 57 定时器 TIM0 总体模块框图

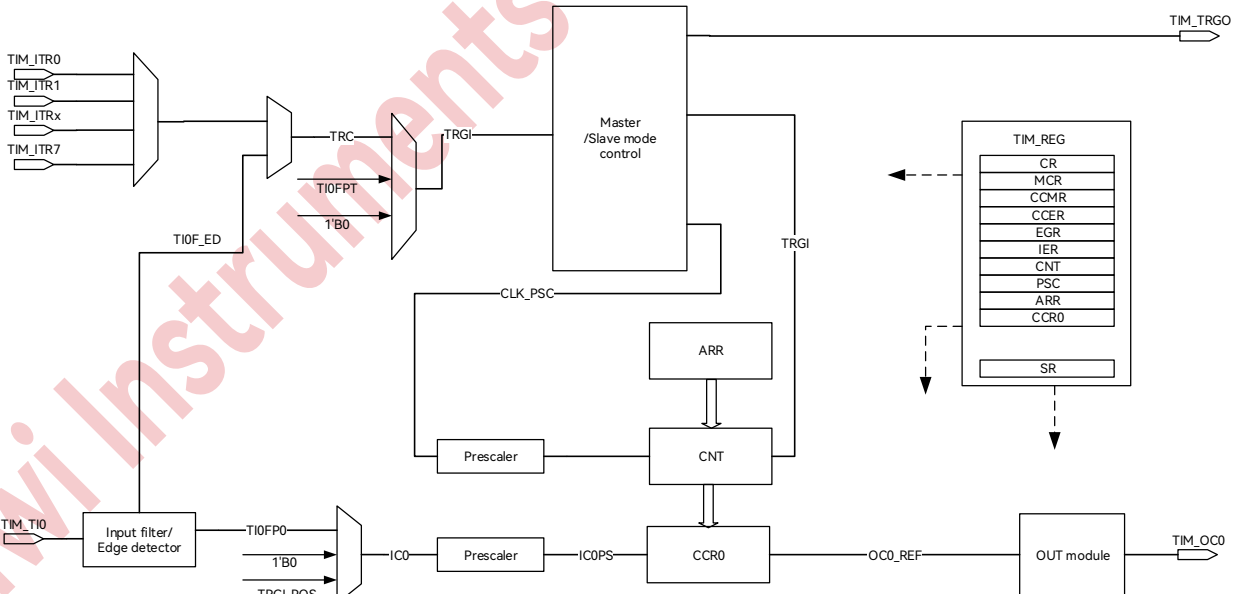


图 58 定时器 TIM1/2 总体模块框图

12.4 功能描述

12.4.1 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 1/2/4/8/16/32/64/128/256/512/1024。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。

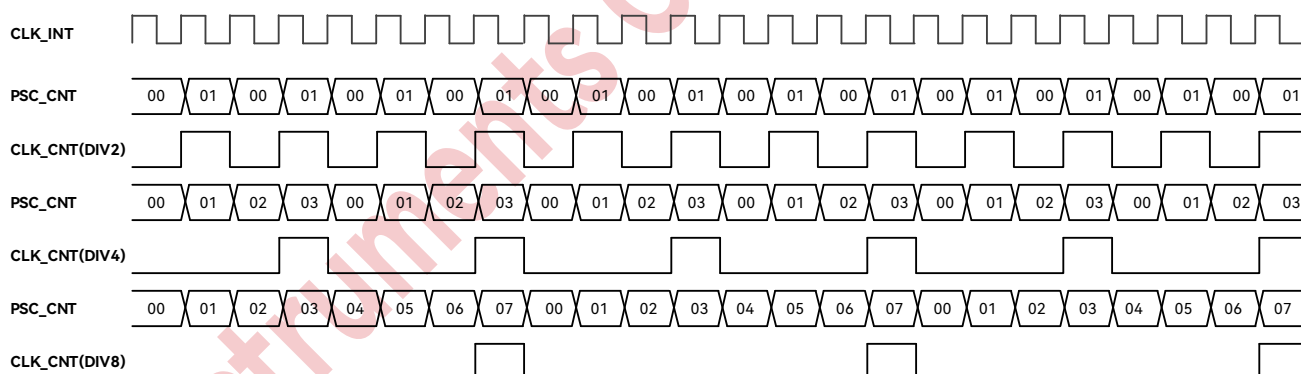


图 59 TIM 预分频器 N 分频时序图

12.4.2 计数器模式

计数器以递增方式进行计数，从 0 计数到自动重载值 (TIM_ARR)，然后重新从 0 开始计数并生成计数器上溢事件。每次发生计数器上溢时会生成更新事件，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。

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)：

- 自动重载影子寄存器将以预装载值进行更新 (TIM_ARR)
- 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)

以下各图以一些示例说明当 TIM_ARR=3 时不同时钟频率下计数器的行为。

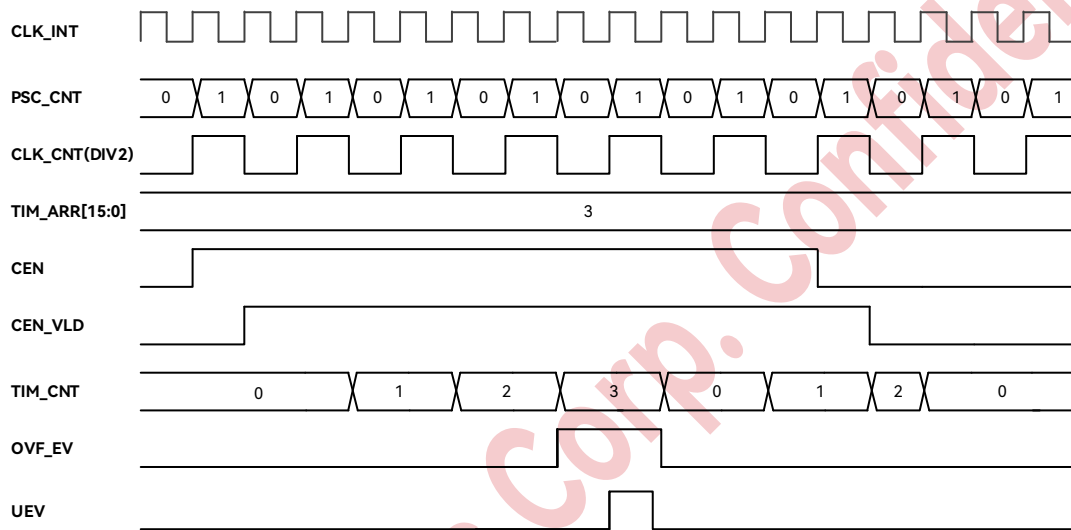


图 60 计数器时序图 (TIM_ARR=3)

12.4.3 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_ITRx)：使用一个定时器作为另一个定时器的预分频器

12.4.3.1 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_CK_INT 提供。

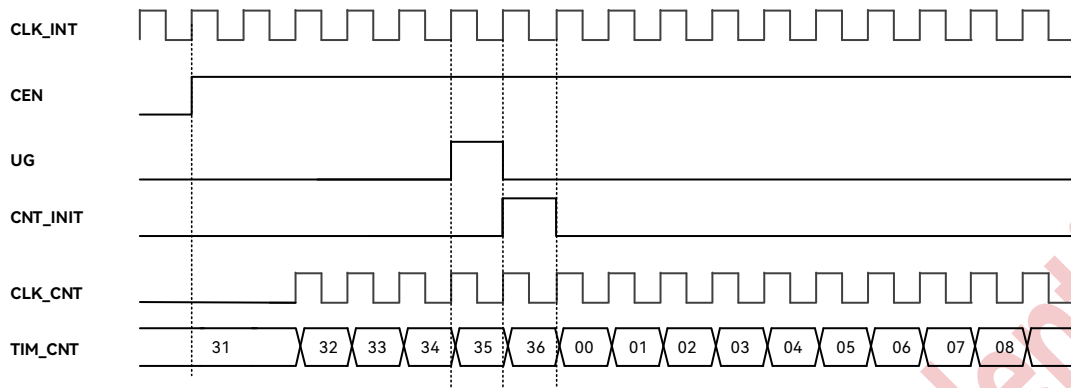


图 61 正常模式下的控制电路 1 分频内部时钟

12.4.3.2 外部时钟源

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。

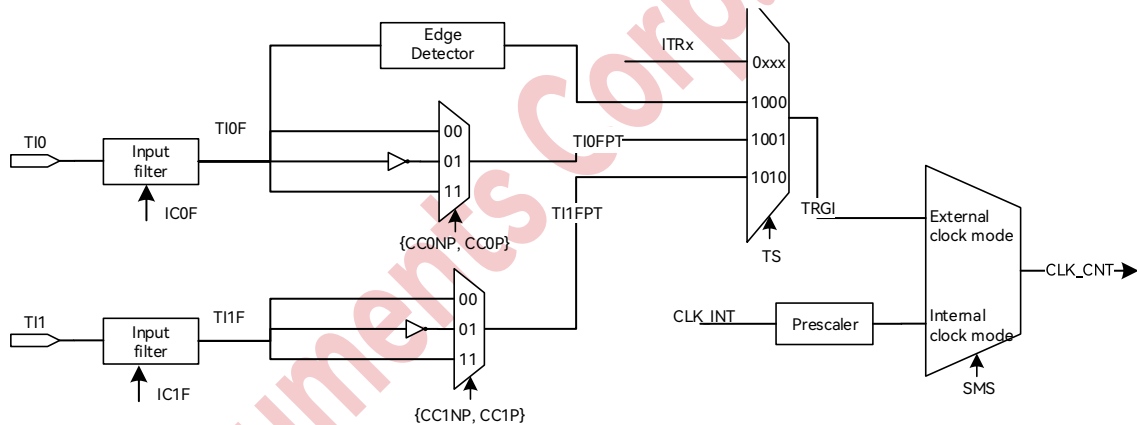


图 62 外部时钟输入连接关系

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_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 TI1 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 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。

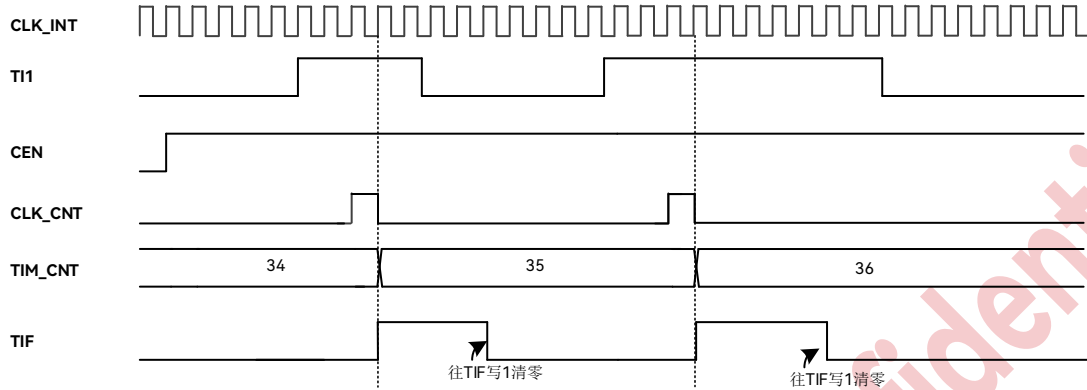


图 63 外部时钟模式下的控制电路

12.4.4 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 TIx_F 。然后，带有极性选择功能的边沿检测器生成一个信号 (TIx_{FPx})，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。

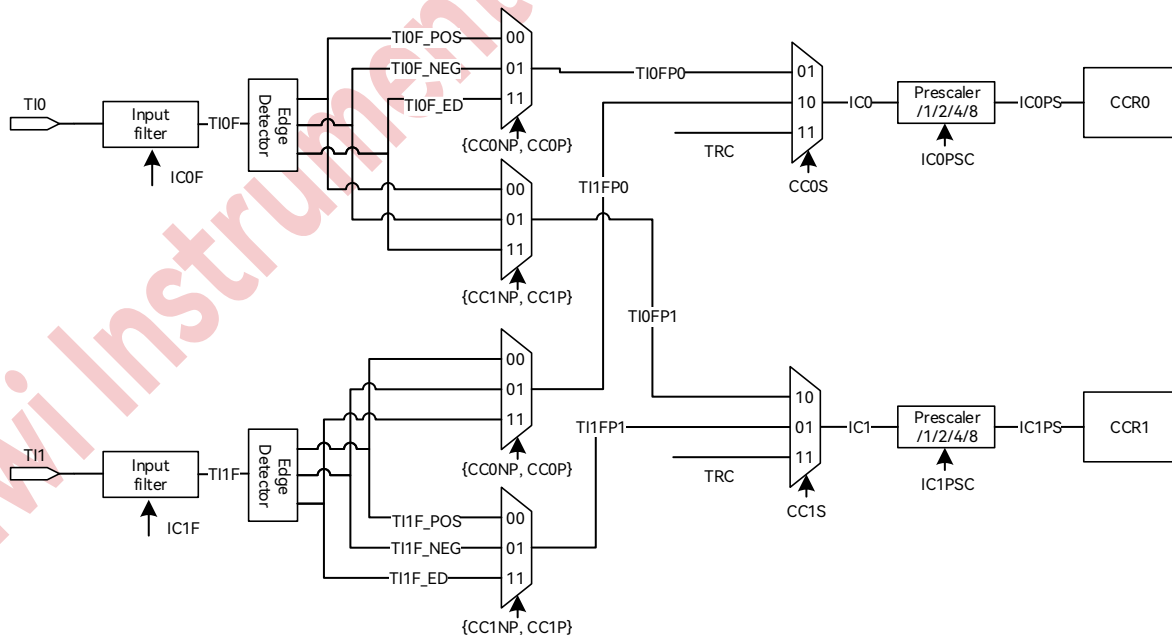


图 64 捕获/比较通道的输入阶段

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。

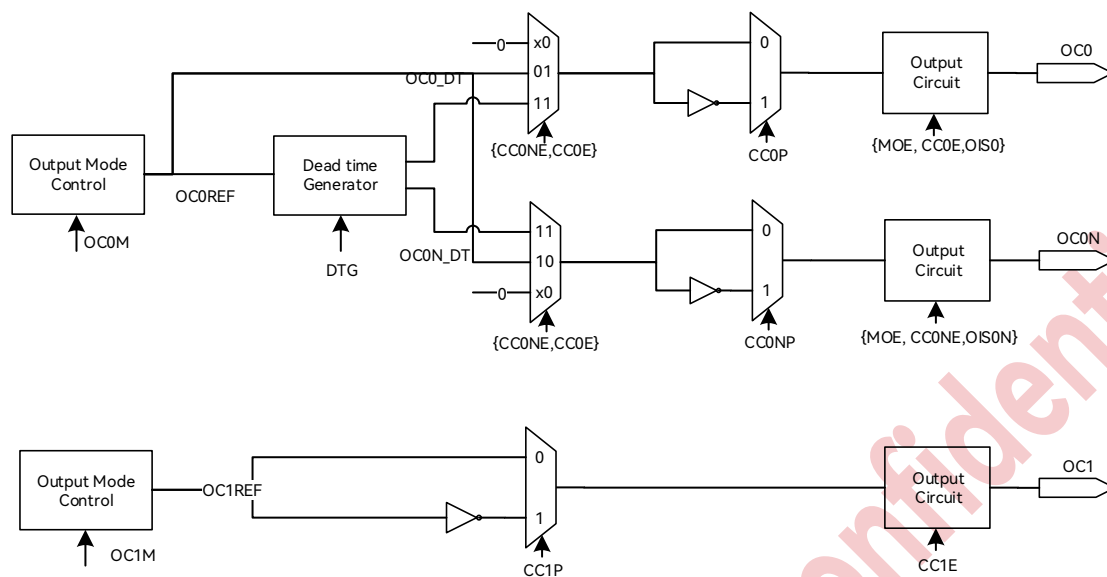


图 65 捕获/比较通道的输出阶段

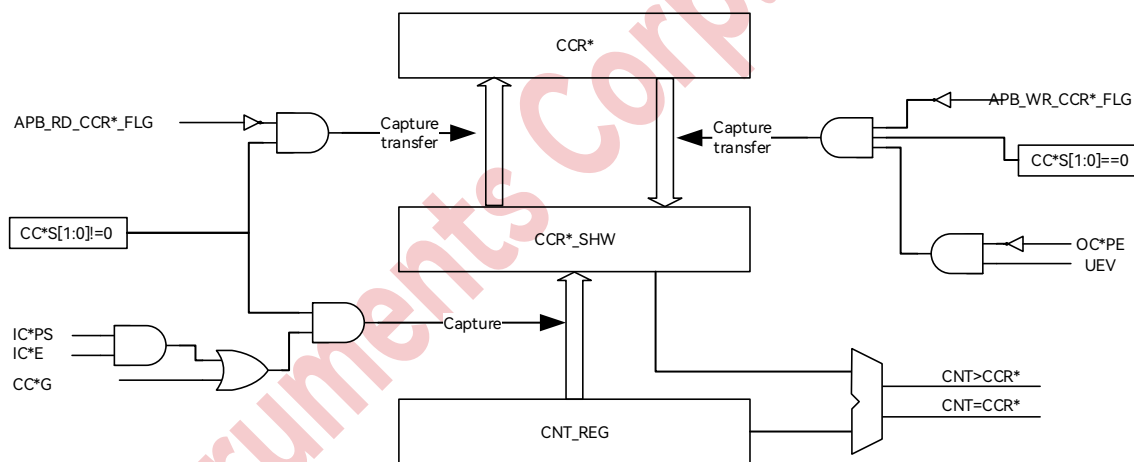


图 66 捕获/比较通道主要电路

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。

12.4.5 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 TIO 输入出现上升沿时将计数器的值捕获到 TIM0_CCR 中。具体操作步骤如下：

1. 选择有效输入：TIM_CCR0 必须连接到 TIO 输入，因此向 TIM_CCMR 寄存器中的 CC0S 位写入 01。
只要 CC0S 不等于 00，就会将通道配置为输入模式，并且 TIM_CCR0 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 CC0P 位和 CC0NP 位写入 0，选择 TIO 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 IC0PS 位写入 00）。
5. 通过将 TIM_CCER 寄存器中的 CC0E 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 CC0IE 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_CCR0 寄存器会获取计数器的值。
- 将 CC0IF 标志置 1 (中断标志)。如果至少发生了两次连续捕获，但 CC0IF 标志未被清零，这样 CC0OF 捕获溢出标志会被置 1。
- 根据 CC0IE 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。

12.4.6 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_CCMR 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_MCR 寄存器中的 TS 位写入 1001 (选择 TI0FP0)。
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 CC0E 位和 CC1E 位分别写入 1。

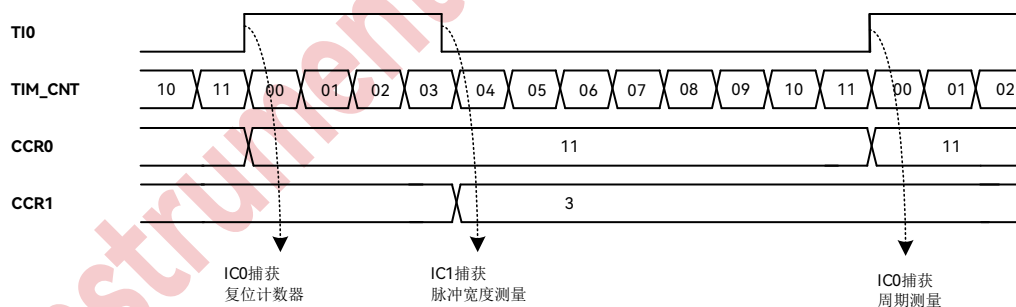


图 67 PWM 输入模式时序

12.4.7 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。

12.4.8 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。

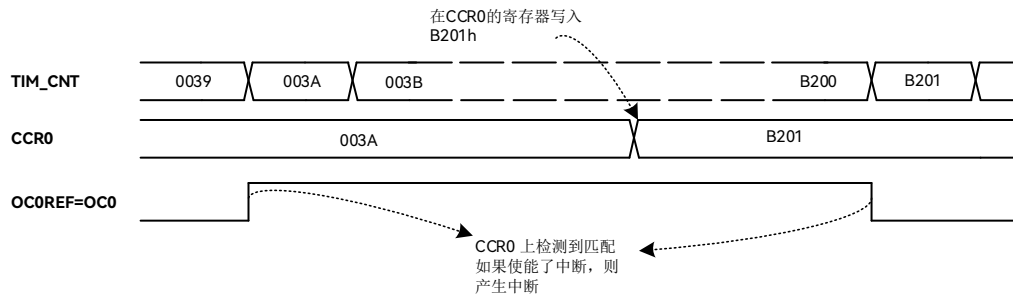


图 68 输出比较模式 (翻转 OC0)

12.4.9 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 $TIM_CCRx < TIM_CNT$ 还是 $TIM_CNT \leq TIM_CCRx$ 。OCREF 信号仅在以下情况下变为有效状态:

- 比较结果发生改变;
- 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 $TIM_CNT < TIM_CCRx$, 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 69 PWM 模式 1 波形 (ARR=8)

12.4.10 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， $CNT < CCRx \leq ARR$ (特别注意， $0 < CCRx$)。

例如，用户希望达到这样的效果：在 TI0 输入引脚检测到正沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。使用 TI0FP0 作为触发 1：

1. 在 TIM_CCMR 寄存器中写入 CC0S=01，将 TI0FP0 映射到 TI0。
2. 在 TIM_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，使 TI0FPT 能够检测上升沿。
3. 在 TIM_MCR 寄存器中写入 TS=1001，将 TI0FPT 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 TI0FPT 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 TIO 上的外部触发事件。本例中，CC0P 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。

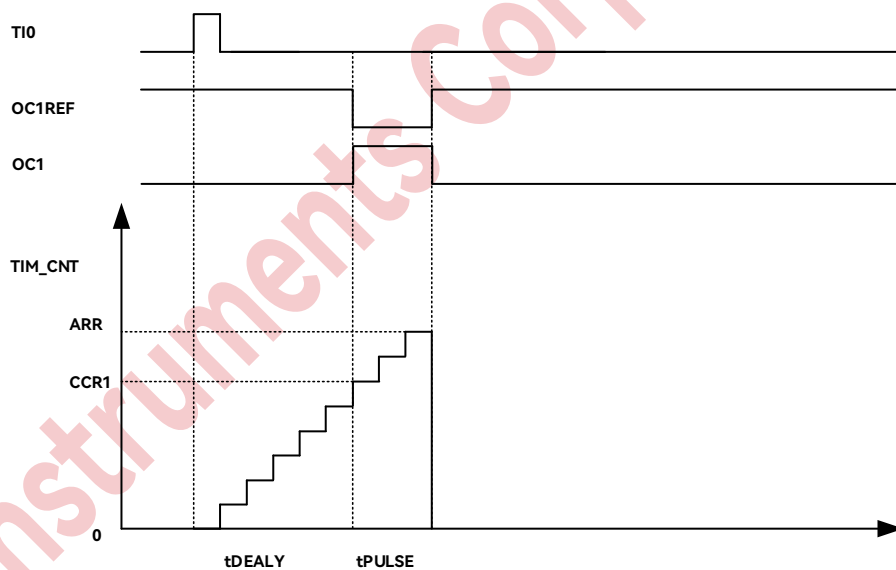


图 70 单脉冲模式示例

12.4.11 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 CC0P 和 CC0NP 位执行写操作来完成极性选择。

互补信号 OC0 和 OC0N 通过以下多个控制位的组合进行激活：TIM_CCER 中的 CC0E 和 CC0NE 位以及 TIM_BDTR 和 TIMx_CR 中的 MOE、OIS0、OIS0N 位。更多详细信息，请参照下表：具有断路功能的互补通道 OC0 和 OC0N 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。

CC0E 和 CC0NE 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 OC0REF 生成 2 个输出 OC0 和 OC0N。如果 OC0 和 OC0N 为高电平有效：

- 输出信号 OC0 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 OC0N 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 (OC0 或 OC0N) 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 CC0P=0、CC0NP=0、MOE=1、CC0E=1 并且 CC0NE=1)

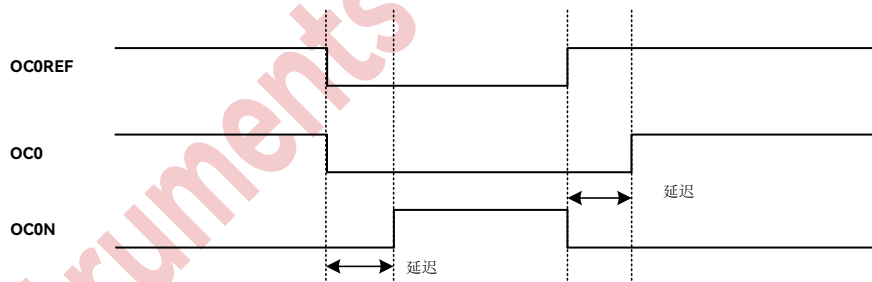


图 71 带死区插入的互补输出

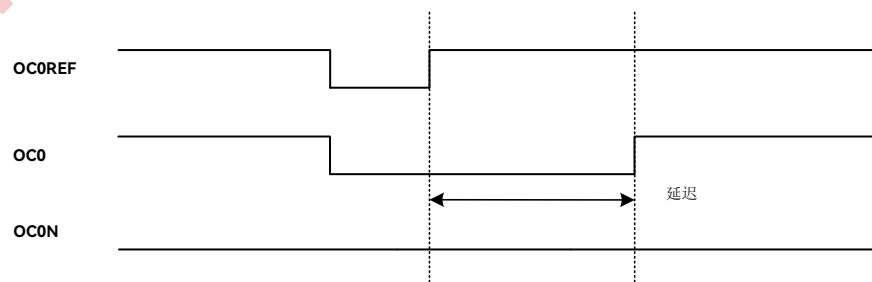


图 72 延迟时间大于负脉冲宽度的死区波形

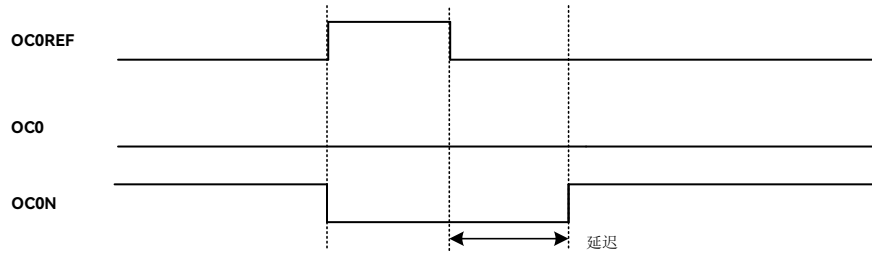


图 73 延迟时间大于正脉冲宽度的死区波形

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 OC0REF 重定向到 OC0 或 OC0N

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 CC0E 和 CC0NE 位，可将 OC0REF 重定向到 OC0 输出或 OC0N 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。

注意：如果仅使能 OC0N (CC0E=0, CC0NE=1)，两者不互补，一旦 OC0REF 为高电平，OC0N 即变为有效。例如，如果 CC0NP=0，则 OC0N=OC0REF。另一方面，如果同时使能 OC0 和 OC0N (CC0E=CC0NE=1)，OC0 在 OC0REF 为高电平时变为有效，而 OC0N 则与之互补，在 OC0REF 为低电平时变为有效。

12.4.12 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 和 OISxN 位) 修改输出使能信号和无效电平。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_CR 寄存器 OISx 位中编程的电平驱动每个输出通道。
- 使用互补输出时：
 - 输出首先置于复位状态或无效状态（取决于极性）。这是异步操作，因此即使没有为定时器提供时钟，该操作仍有效。
 - 如果定时器时钟仍存在，则将重新激活死区发生器，进而在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，也不能同时将 OCx 和 OCxN 驱动至其有效电平。请注意，MOE 进行再同步，因此死区的持续时间会比通常情况长一些（约 2 个定时器时钟周期）。
- 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。

注意：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。

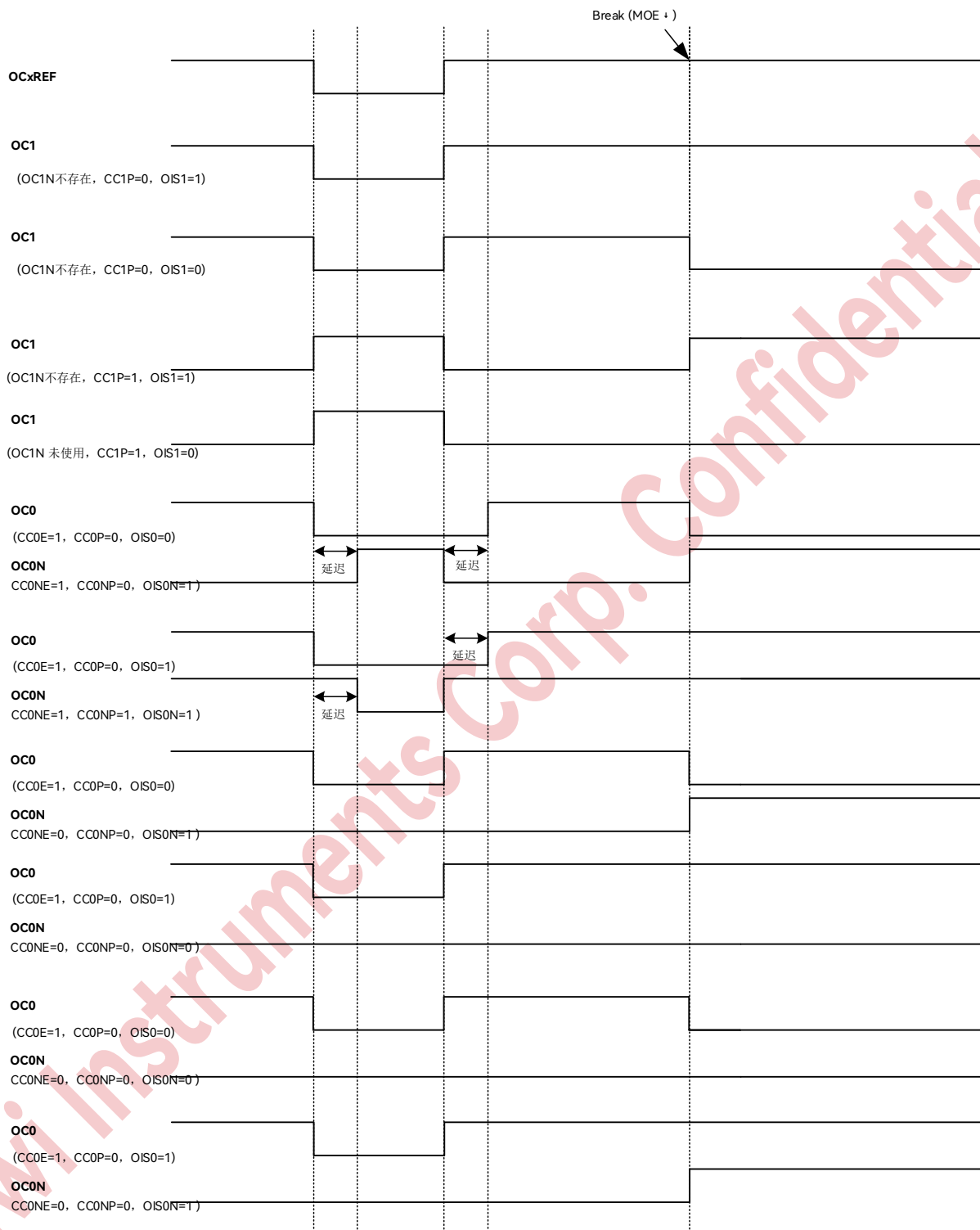


图 74 输出的断路响应行为

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。

定时器支持外部 IO 输入断路由 EBUS 输入至 TIM0, TIM0_EBUS_IO_SEL 用于选择输入通道并支持极性可编程, 也支持 CMP 输出事件和 ADC 事件 (ADC_EV_TIM) 用于断路, 具体原理如下图所示。

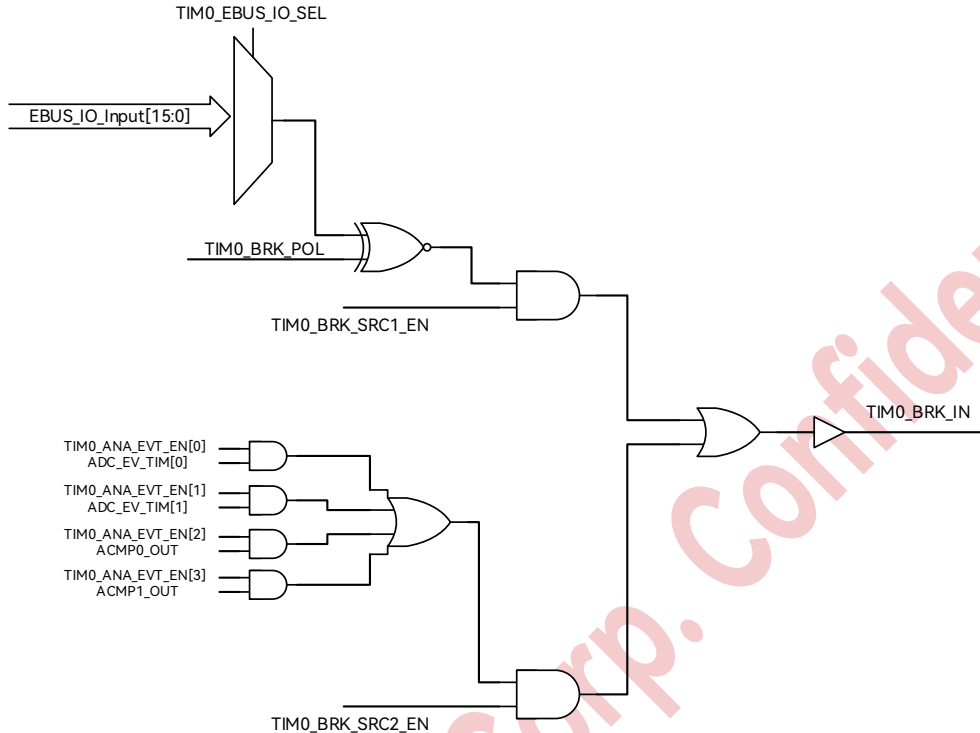


图 75 定时器断路源示意图

12.4.13 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。

12.4.13.1 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，TIM0 输入上出现上升沿时，计数器清零：

1. 将通道 1 配置为检测 TIM0 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 IC0F=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。CC0S 位只选择输入捕获源，即 TIM_CCMR 寄存器中的 CC0S=01。在 TIM_CCER 寄存器中写入 CC0P=0 和 CC0NP=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 TIM0 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 TIM0 上升沿。当 TIM0 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。TIO 的上升沿与实际计数器复位之间的延迟是由于 TIO 输入的重新同步电路引起的。

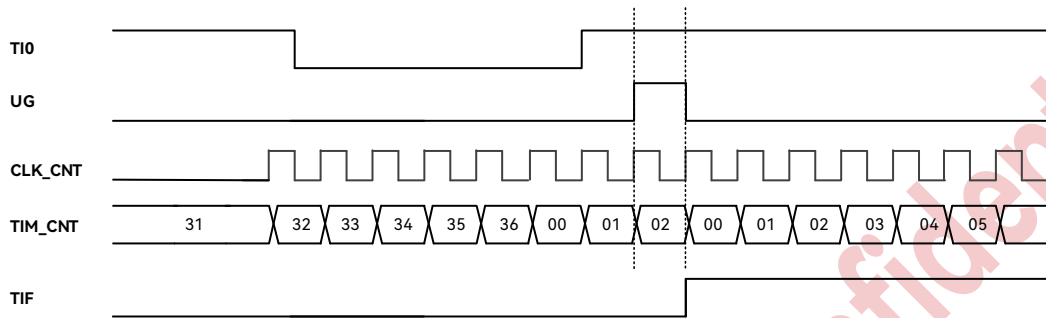


图 76 复位模式下的控制电路

12.4.13.2 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 TIO 输入为低电平时计数。

1. 将通道 0 配置为检测 TIO 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 IC0F=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。CC0S 位只选择输入捕获源，即 TIM_CCMR 寄存器中的 CC0S=01。在 TIM_CCER 寄存器中写入 CC0P=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 TIO 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 TIO 为低电平，计数器就开始根据内部时钟计数，直到 TIO 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。TIO 的上升沿与实际计数器停止之间的延迟是由于 TIO 输入的重新同步电路引起的。

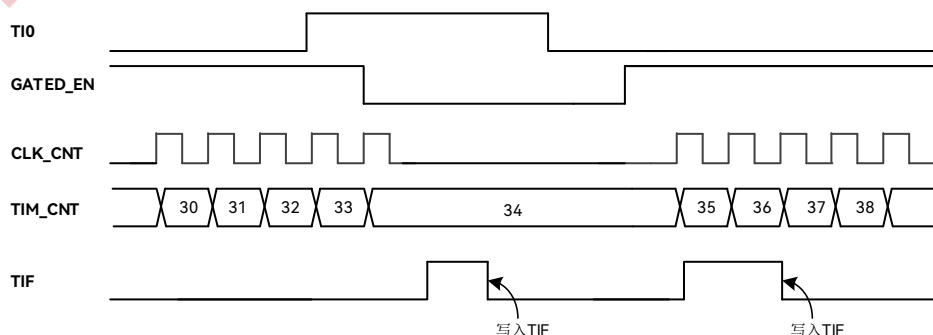


图 77 门控模式下的控制电路

12.4.13.3 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_CCMR 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_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 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。

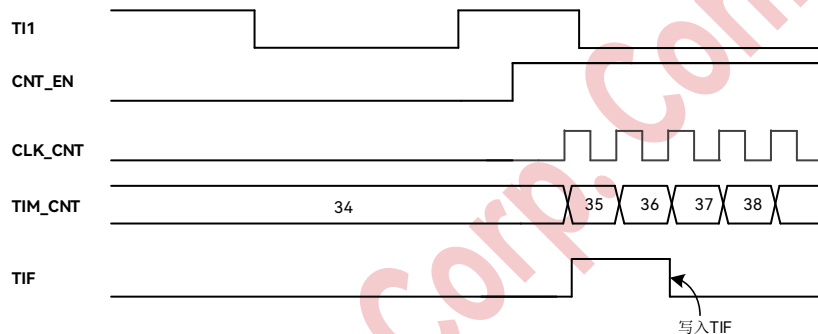


图 78 触发模式下的控制电路

12.4.14 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。

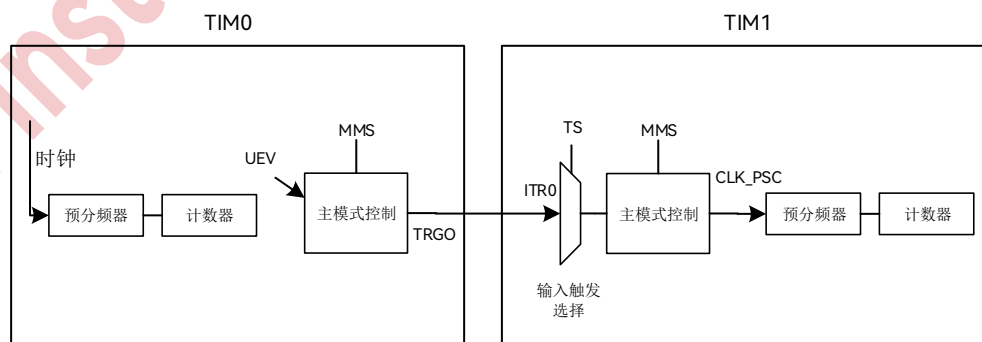


图 79 主/从定时器示例

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 ITRO 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. 最后必须通过在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 0 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 OC0REF 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 ($f_{CK_CNT}=f_{CK_INT}/2$)。

1. 将定时器 0 配置为主模式，发送其输出比较 0 参考信号 (OC1REF) 作为触发输出 (TIM0_CR 寄存器中的 MMS=100)。
2. 配置定时器 0 的 OC0REF 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. 使能定时器 1
6. 使能定时器 0

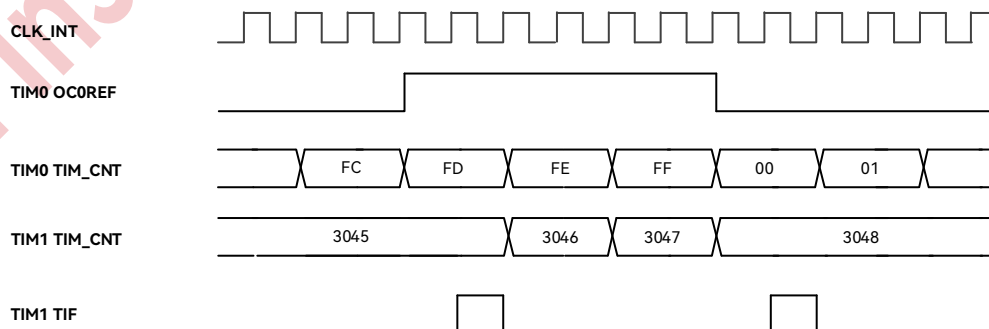


图 80 使用定时器 0 的 OC1REF 对定时器 1 实施门控控制

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_CR 寄存器中通过向 TIM_CEN 位写 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 0 参考信号 (OC0REF) 作为触发输出 (TIM1_CR 寄存器中的 MMS=100)。
2. 配置定时器 0 的 OC0REF 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。

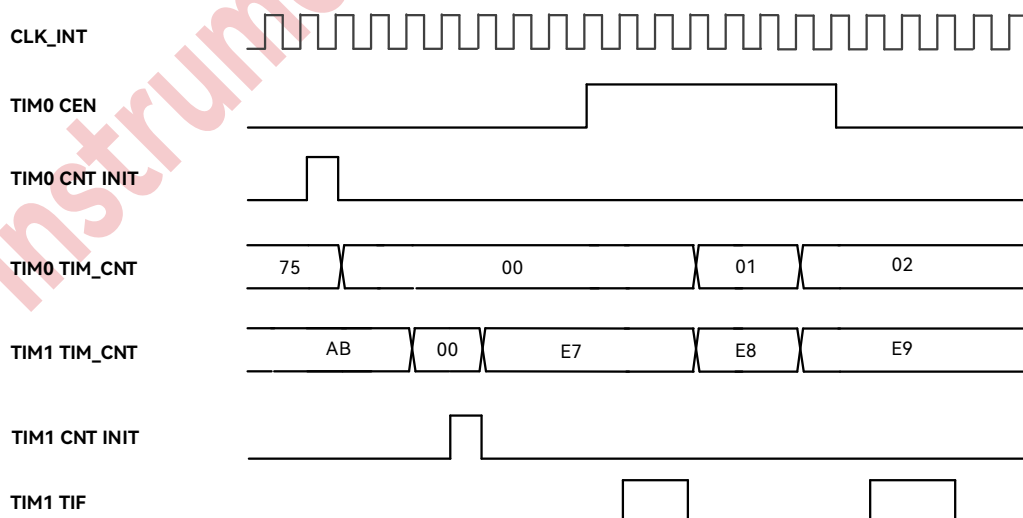


图 81 使用定时器 0 的使能信号对定时器 1 实施门控控制

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。

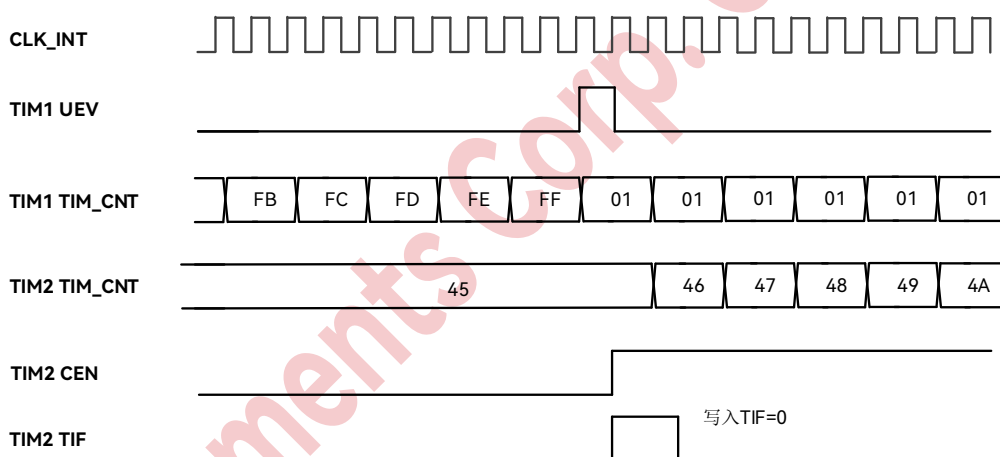


图 82 使用定时器 1 的更新事件触发定时器 2

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。

12.4.15 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 122 TIMS 中断号

名称	中断号
TIM0_INT	19
TIM1_INT	31
TIM2_INT	30

12.5 寄存器描述

TIM 寄存器基地址安排如下表所示。

表 123 基地址对应表

定时器	基地址地址
TIM0	0x4000_0400
TIM1	0x4000_0800
TIM2	0x4000_0C00

下表为 TIM 的各控制寄存器描述。

表 124 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
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	0x00000000
0x6C	Reserved	32	RES	0x00000000
0x70	TIM 捕获/比较寄存器 0	32	r/w	0x00000000
0x74	TIM 捕获/比较寄存器 1	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

偏移地址	寄存器列表	位宽	访问	复位值
0xC4	TIM 状态寄存器	32	r/w	0x00000000
0xC8~0xFC	Reserved	32	RES	0x00000000

上述是 3 个 TIM 的寄存器寄存器列表，除此之外，这有一些控制寄存器用于控制这些定时器的输入，其地址为 0x4000_1400。

表 125 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	TIM 输入复用控制寄存器	32	r/w	0x00000000
0x04	TIM 断路控制寄存器	32	r/w	0x00000000
0x08	TIM 霍尔状态寄存器	32	ro	0x00000000
0x0C~0xFC	Reserved	32	RES	0x00000000

12.5.1 TIM 控制寄存器 (TIM_CR)

偏移地址：0x0000

表 126 TIM 控制寄存器

位	访问	描述
31:19	res	Reserved 复位值：0x0
18	r/w	OIS1，输出空闲状态 1 (OC1 输出) 0：当 MOE=0 时，OC1=0 1：当 MOE=0 时，OC1=1 注：只有 1 个单通道的定时器 (TIM1/2) 不包含此位定义 复位值：0x0
17	r/w	OIS0N，输出空闲状态 0 (OC0N 输出) 0：当 MOE=0 时，经过死区时间之后 OC0N=0 1：当 MOE=0 时，经过死区时间之后 OC0N=1 注：只有 1 个单通道的定时器 (TIM1/2) 不包含此位定义 复位值：0x0
16	r/w	OIS0，输出空闲状态 0 (OC0 输出) 0：当 MOE=0 时，(如果 OC0N 有效，则经过死区时间之后) OC0=0 1：当 MOE=0 时，(如果 OC0N 有效，则经过死区时间之后) OC0=1 注：只有 1 个单通道的定时器 (TIM1/2) 不包含此位定义 复位值：0x0
15:10	res	Reserved 复位值：0x0

位	访问	描述
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 (TlX) 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 注: 定时器 (TIM1/2) 不包含此位定义 复位值: 0x0
6:4	res	Reserved 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 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: - 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: - 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: 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

12.5.2 TIM 模式控制寄存器 (TIM_MCR)

偏移地址: 0x0004

表 127 TIM 模式控制寄存器

位	访问	描述
31:24	res	Reserved 复位值: 0x0
23:20	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, 0000: 内部触发信号 0 (TIM_ITR0) 0001: 内部触发信号 1 (TIM_ITR1) 0010: 内部触发信号 2 (TIM_ITR2) 0011: 内部触发信号 3 (TIM_ITR3) 0100: 内部触发信号 4 (TIM_ITR4) 0101: 内部触发信号 5 (TIM_ITR5) 0110: 内部触发信号 6 (TIM_ITR6) 0111: 内部触发信号 7 (TIM_ITR7) 1000: TI0 边沿检测器 (TIM_TI0F_ED) 1001: 滤波后的 TI0 输入 (TIM_TI0FPT) 1010: 滤波后的 TI1 输入 (TIM_TI1FPT) (注: 对于只有 1 个通道的定时器, 此设置位保留) 其他: 保留 这些控制位用于选择触发输入作为 TIM_TRC, 其定义如下, 0000: 内部触发信号 0 (TIM_ITR0) 0001: 内部触发信号 1 (TIM_ITR1) 0010: 内部触发信号 2 (TIM_ITR2) 0011: 内部触发信号 3 (TIM_ITR3) 0100: 内部触发信号 4 (TIM_ITR4) 0101: 内部触发信号 5 (TIM_ITR5) 0110: 内部触发信号 6 (TIM_ITR6) 0111: 内部触发信号 7 (TIM_ITR7) 其他:内部触发信号 0 (TIM_ITR0) 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。 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 复位值: 0x0
19	res	Reserved 复位值: 0x0

位	访问	描述
18:16	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: 从模式关闭——如果 CEN="1", 预分频器时钟直接由内部时钟提供。 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——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。 其他: 保留 复位值: 0x0
15:9	res	Reserved 复位值: 0x0
8	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
7:3	res	Reserved 复位值: 0x0
2:0	r/w	MMS, 主模式选择 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)。 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: 保留 复位值: 0x0

表 128 定时器内部触发信号对应关系

内部触发信号	TIM0	TIM1	TIM2	TIM8
TIM_ITR0	-	TIM0_TRGO	TIM0_TRGO	TIM0_TRGO
TIM_ITR1	TIM1_TRGO	-	TIM1_TRGO	TIM1_TRGO
TIM_ITR2	TIM2_TRGO	TIM2_TRGO	-	TIM8_TRGO
TIM_ITR3	TIM8_TRGO	TIM8_TRGO	TIM8_TRGO	-
TIM_ITR4	-	-	-	-
TIM_ITR5	-	-	-	-
TIM_ITR6	-	-	-	-
TIM_ITR7	-	-	-	-

12.5.3 TIM 捕获/比较模式寄存器 (TIM_CCMR)

偏移地址: 0x0008

输出比较模式:

表 129 TIM 捕获/比较模式寄存器

位	访问	描述
31:15	res	Reserved 复位值: 0x0
14:12	r/w	OC1M, 输出比较 1 模式 参照 OC1M 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	OC1PE, 输出比较 1 预装载使能 参照 OC0PE 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	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_MCR 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。 对于只有 1 个通道的定时器, 此设置位保留 复位值: 0x0
7	res	Reserved 复位值: 0x0

位	访问	描述
6:4	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。 000: 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。 001: 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。 010: 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。 011: 翻转——TIM_CNT=TIM_CCR0 时, OC0REF 发生翻转。 100: 强制变为无效电平——OC0REF 强制变为低电平。 101: 强制变为有效电平——OC0REF 强制变为高电平。 110: PWM 模式 1——在计数模式下, 只要 TIM_CNT>TIM_CCR0, 通道 0 便为有效状态, 否则为无效状态。 111: PWM 模式 2——在计数模式下, 只要 TIM_CNT≤TIM_CCR0, 通道 0 便为有效状态, 否则为无效状态。 注: 在 PWM 模式 1 或 PWM 模式 2 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。 复位值: 0x0
3	r/w	OC0PE: 输出比较 0 预装载使能 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。 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。 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_CR 寄存器中的 OPM 位置 1)。其它情况下则无法保证该行。 复位值: 0x0
2	res	Reserved 复位值: 0x0
1:0	r/w	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 TI0 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_MCR 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 CC0S 位写入数据。 复位值: 0x0

输入捕获模式下。

表 130 TIM 捕获/比较模式寄存器

位	访问	描述
31:16	res	Reserved 复位值: 0x0
15:12	r/w	IC1F, 输入捕获 1 滤波器 参照 IC0F 相关描述 注: 对于只有 1 个通道的定时器, 此设置位保留

位	访问	描述
		复位值: 0x0
11:10	r/w	IC1PSC, 输入捕获 1 预分频器 参照 IC0PSC 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
9:8	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_MCR 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。 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	IC0F, 输入捕获 0 滤波器 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 IC0F[2:0]=000: $f_{\text{SAMPLING}}=f_{\text{CK_INT}}$ IC0F[2:0]=001: $f_{\text{SAMPLING}}=f_{\text{DTS}}$ IC0F[2:0]=010: $f_{\text{SAMPLING}}=f_{\text{DTS}}/2$ IC0F[2:0]=011: $f_{\text{SAMPLING}}=f_{\text{DTS}}/4$ IC0F[2:0]=100: $f_{\text{SAMPLING}}=f_{\text{DTS}}/8$ IC0F[2:0]=101: $f_{\text{SAMPLING}}=f_{\text{DTS}}/16$ IC0F[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: IC0F[3]=0: N=1 IC0F[3]=1: N=2 复位值: 0x0
3:2	r/w	IC0PSC, 输入捕获 0 预分频器 此位域定义 CC0 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
1:0	r/w	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 TI0 上 10: CC0 通道配置为输入, IC0 映射到 TI1 上 (注: 对于只有 1 个通道的定时器, 此设置位保留) 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

位	访问	描述
		注：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

12.5.4 TIM 捕获/比较模式使能寄存器 (TIM_CCER)

偏移地址：0x0020

表 131 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_CC1NP, 捕获/比较 1 互补输出极性 参照 CC0NP 描述, 该位只在输入模式下有意义 注：对于只有 1 个通道的定时器, 此设置位保留 复位值：0x0
6	res	Reserved 复位值：0x0
5	r/w	TIM_CC1P, 捕获/比较 1 输出极性 参照 CC0P 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
4	r/w	TIM_CC1E, 捕获/比较 1 输出使能 参照 CC0E 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
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 CC0P 说明。 注：对于定时器 (TIM1/2), 该位只在输入条件下有意义 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、OIS0、OIS0N 和 CC0E 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、OIS0N 和 CC0E 位。 注：对于定时器 (TIM1/2), 该位保留 复位值：0x0

位	访问	描述
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、OIS0N 和 CC0NE 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、OIS0N 和 CC0NE 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

12.5.5 TIM 计数器 (TIM_CNT)

偏移地址: 0x0060

表 132 TIM 计数器

位	访问	描述
31:16	res	Reserved 复位值: 0x0
15:0	r/w	CNT[15:0], 计数器值 复位值: 0x0

12.5.6 TIM 预分频器 (TIM_PSC)

偏移地址: 0x0064

表 133 TIM 预分频器

位	访问	描述
31:4	res	Reserved 复位值: 0x0
3: 0	r/w	PSC, 预分频器 计数器的工作时钟 (TIM_CLK_CNT) 频率等于 $f_{CK_INT}/(2^{PSC})$。但对于该预分频器, TIM_CLK_CNT 所能支持的最小为 $f_{CK_INT}/1024$。相关控制位定义如下: 0: $f_{CLK_CNT}=f_{CK_INT}$ 1: $f_{CLK_CNT}=f_{CK_INT}/2$ 2: $f_{CLK_CNT}=f_{CK_INT}/4$ 3: $f_{CLK_CNT}=f_{CK_INT}/8$ 4: $f_{CLK_CNT}=f_{CK_INT}/16$ 5: $f_{CLK_CNT}=f_{CK_INT}/32$ 6: $f_{CLK_CNT}=f_{CK_INT}/64$ 7: $f_{CLK_CNT}=f_{CK_INT}/128$ 8: $f_{CLK_CNT}=f_{CK_INT}/256$ 9: $f_{CLK_CNT}=f_{CK_INT}/512$ 10: $f_{CLK_CNT}=f_{CK_INT}/1024$ 其他: $f_{CLK_CNT}=f_{CK_INT}/1024$ 注: 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

12.5.7 TIM 自动重载寄存器 (TIM_ARR)

偏移地址: 0x0068

表 134 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

12.5.8 TIM 捕获/比较寄存器 0 (TIM_CCR0)

偏移地址: 0x0070

表 135 TIM 捕获/比较寄存器 0

位	访问	描述
31:16	res	Reserved 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

12.5.9 TIM 捕获/比较寄存器 1 (TIM_CCR1)

偏移地址: 0x0074

表 136 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。 注: 对于定时器 (TIM1/2), 该位保留 复位值: 0x0

12.5.10TIM 断路及死区寄存器 (TIM_BDTR)

偏移地址: 0x0090

表 137 TIM 断路及死区寄存器

位	访问	描述
30:16	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) 复位值: 0x0
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 复位值: 0x0
11:8	res	Reserved 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) 16 μs 到 31750ns (步长为 250ns) 32 μs 到 63 μs (步长为 1 μs) 64 μs 到 126 μs (步长为 2 μs) 复位值: 0x0

注: 对于定时器 1~2, 此寄存器保留。

12.5.11 TIM 事件生成寄存器 (TIM_EGR)

偏移地址: 0x00B0

表 138 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。 注: 对于定时器 1~2, 此寄存器保留 复位值: 0x0
6:3	res	Reserved 复位值: 0x0
2	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: 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。 注: 对于定时器 1~2, 此寄存器保留 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 0 上生成捕获/比较事件: 通道 CC0 配置为输出: 使能时, CC0IF 标志置 1 并发送相应的中断。 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。如果 CC0IF 标志已为高电平, CC0OF 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

12.5.12 TIM 中断使能寄存器 (TIM_IER)

偏移地址: 0x00C0

表 139 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 注: 对于定时器 1~2, 此寄存器保留

位	访问	描述
		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
5:3	res	Reserved 复位值: 0x0
2	r/w	CC1IE, 捕获/比较 1 中断使能 0: 关闭 1: 开启 注: 对于定时器 1~2, 此寄存器保留 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

12.5.13 TIM 状态寄存器 (TIM_SR)

偏移地址: 0x00C4

表 140 TIM 状态寄存器

位	访问	描述
31:11	res	Reserved 复位值: 0x0
10	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。 注: 对于定时器 1~2, 此寄存器保留 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: 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。

位	访问	描述
		注：对于定时器 1~2，此寄存器保留 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
5:3	res	Reserved 复位值：0x0
2	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
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 TIM_CCR0 寄存器的值匹配。当 TIM_CCR0 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 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。 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

12.5.14 TIM 输入复用控制寄存器 (TIM_IMCR)

偏移地址: 0x0000

表 141 TI 复用控制寄存器

位	访问	描述
31:17	res	Reserved 复位值: 0x0
17:16	r/w	TIM8_ETR_SEL, TIM8 ETR 输入复用选择 0: TIM8_ETR 1: TIM8_ETR 2: CMP0_OUT 3: CMP1_OUT 复位值: 0x0
15:12	r/w	EBUS_SELW, EBUS IO 输入复用选择 (用作霍尔输入) 选择 EBUS_IO_INPUT[EBUS_SELW] 用作霍尔输入 复位值: 0x0
11:8	r/w	EBUS_SELV, EBUS IO 输入复用选择 (用作霍尔输入) 选择 EBUS_IO_INPUT[EBUS_SELV]用作霍尔输入 复位值: 0x0
7:4	r/w	EBUS_SELU, EBUS IO 输入复用选择 (用作霍尔输入) 选择 EBUS_IO_INPUT[EBUS_SELU] 用作霍尔输入 复位值: 0x0
3:2	res	Reserved 复位值: 0x0
1:0	r/w	TIM2_TI_SEL, TIM2 TI 输入复用选择 0: TIM2_CH0 1: EBUS_IO_INPUT[EBUS_SELU]^EBUS_IO_INPUT[EBUS_SELV]^ EBUS_IO_INPUT[EBUS_SELW] 2: PGA0_OUT^PGA1_OUT^CMP0_OUT 3: PGA0_OUT^PGA1_OUT^CMP1_OUT 复位值: 0x0

12.5.15 TIM 断路控制寄存器 (TIM_BCR0)

偏移地址: 0x0004

表 142 TIM 断路使能寄存器 0

位	访问	描述
31:16	res	Reserved 复位值: 0x0
15:12	r/w	TIM0_ANA_EVT_EN[3:0], TIM0 模拟异常事件源使能 位 0: 1, 启用 ADC 事件 0 (ADC 转换值大于比较上限或小于比较下限); 0, 关闭 位 1: 1, 启用 ADC 事件 1 (ADC 转换值小于比较上限且大于比较下限); 0, 关闭 位 2: 1, 启用 CMP0_OUT 事件; 0, 关闭 位 3: 1, 启用 CMP1_OUT 事件; 0, 关闭 复位值: 0x0
11:8	r/w	TIM0_EBUS_IO_SEL, EBUS IO 输入选择控制 (TIM0) 0: 选择通道 0 作为 TIM0 断路源 1: 选择通道 1 作为 TIM0 断路源 ... 14: 选择通道 14 作为 TIM0 断路源 15: 选择通道 15 作为 TIM0 断路源 复位值: 0x0
7:5	res	Reserved 复位值: 0x0
4	r/w	TIM0_BRK_POL, TIM0_BRK 极性 0: 外部断路输入 BRK 为低电平有效 1: 外部断路输入 BRK 为高电平有效 复位值: 0x1
3	res	Reserved 复位值: 0x0
2	r/w	TIM0_BRK_SRC2_EN, TIM0 断路源 2 (ANA_EVT 模拟事件) 使能 0: 关闭 1: 开启 复位值: 0x0
1	r/w	TIM0_BRK_SRC1_EN, TIM0 断路源 1 (EBUS IO 输入) 使能 0: 关闭 1: 开启 复位值: 0x0
0	res	Reserved 复位值: 0x0

12.5.16 TIM 断路控制寄存器 (TIM_BCR1)

偏移地址: 0x0008

表 143 TIM 断路使能寄存器 1

位	访问	描述
31	r/w	TIM8_EBCR_LOCK, TIM8 外部断路控制寄存器锁定 0: 未锁定 1: 锁定 注: 该位只能从 0 写 1 复位值: 0x0
30:16	res	Reserved 复位值: 0x0
15:12	r/w	TIM8_ANA_EVT_EN[3:0], TIM0 模拟异常事件源使能 位 0: 1, 启用 ADC 事件 0 (ADC 转换值大于比较上限或小于比较下限); 0, 关闭 位 1: 1, 启用 ADC 事件 1 (ADC 转换值小于比较上限且大于比较下限); 0, 关闭 位 2: 1, 启用 CMP0_OUT 事件; 0, 关闭 位 3: 1, 启用 CMP1_OUT 事件; 0, 关闭 复位值: 0x0
11:8	r/w	TIM8_EBUS_IO_SEL, EBUS IO 输入选择控制 (TIM8) 0: 选择通道 0 作为 TIM8 断路源 1: 选择通道 1 作为 TIM8 断路源 ... 14: 选择通道 14 作为 TIM8 断路源 15: 选择通道 15 作为 TIM8 断路源 复位值: 0x0
7:5	res	Reserved 复位值: 0x0
4	r/w	TIM8_BRK_POL, TIM8_BRK 极性 0: 外部断路输入 BRK 为低电平有效 1: 外部断路输入 BRK 为高电平有效 复位值: 0x1
3	res	Reserved 复位值: 0x0
2	r/w	TIM8_BRK_SRC2_EN, TIM8 断路源 2 (ANA_EVT 模拟事件) 使能 0: 关闭 1: 开启 复位值: 0x0
1	r/w	TIM8_BRK_SRC1_EN, TIM8 断路源 1 (EBUS IO 输入) 使能 0: 关闭 1: 开启 复位值: 0x0
0	r/w	TIM8_BRK_SRC0_EN, TIM8 断路源 0 (外部 IO) 使能 0: 关闭 1: 开启 复位值: 0x0

12.5.17 TIM 霍尔状态寄存器 (TIM_HSR)

偏移地址: 0x0010

表 144 TIM 霍尔状态寄存器

位	访问	描述
31:3	res	Reserved 复位值: 0x0
2	ro	W, W 路状态 复位值: 0x0
1	ro	V, V 路状态 复位值: 0x0
0	ro	U, U 路状态 复位值: 0x0

注意: U、V 和 W 对应通道的关系如下。

表 145 U、V 和 W 对应通道

通道	TIM2_TI_SEL=1	TIM2_TI_SEL=2	TIM2_TI_SEL=3
U	EBUS_IO_INPUT[EBUS_SELU]	PGA0_OUT	PGA0_OUT
V	EBUS_IO_INPUT[EBUS_SELV]	PGA1_OUT	PGA1_OUT
W	EBUS_IO_INPUT[EBUS_SELW]	CMP0_OUT	CMP1_OUT

13 看门狗 (WDT)

13.1 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。

13.2 主要功能

WDT 主要功能如下。

- 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

13.3 模块框图

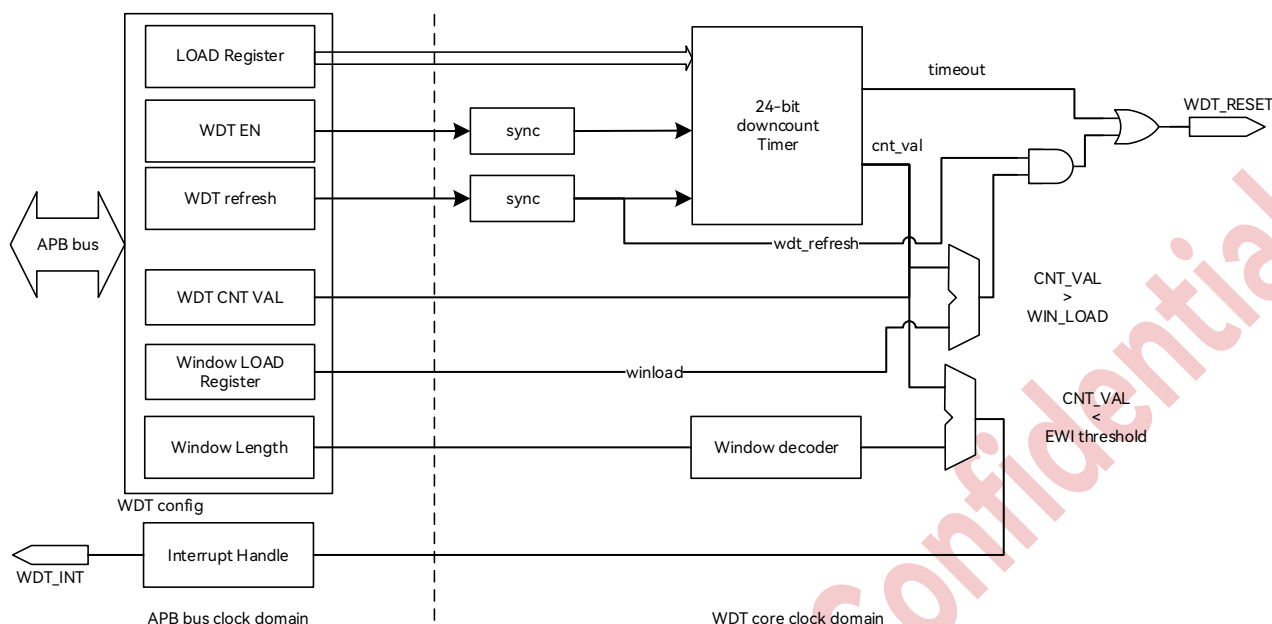


图 83 WDT 模块框图

13.4 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_RSTJ)。具体相关功能详见以下章节。

13.4.1 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。

13.4.2 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

13.4.2.1 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。

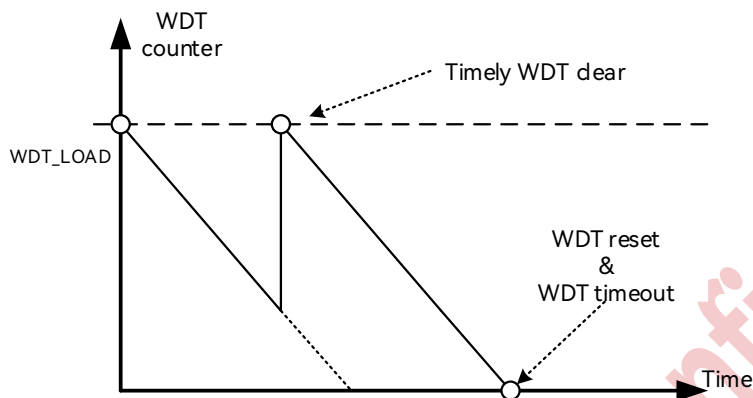


图 84 WDT 普通模式 1

13.4.2.2 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 EWL_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。

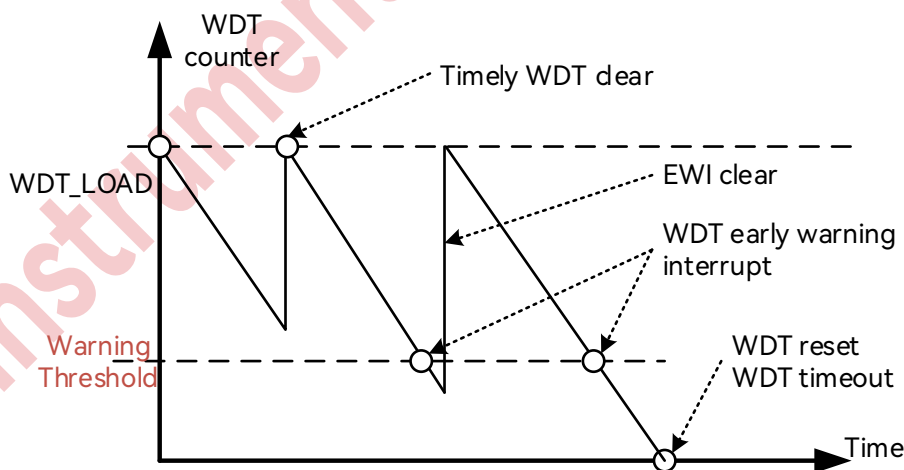


图 85 WDT 普通模式 2

13.4.2.3 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。

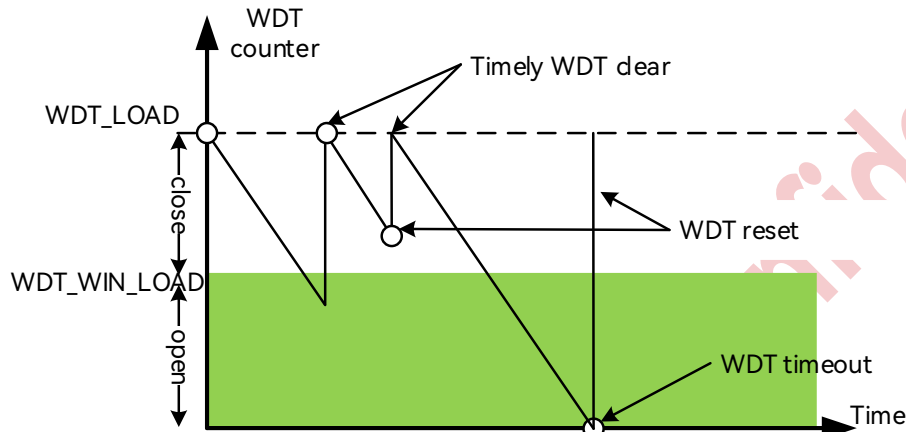


图 86 WDT 窗口模式 1

13.4.2.4 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 EWI_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。

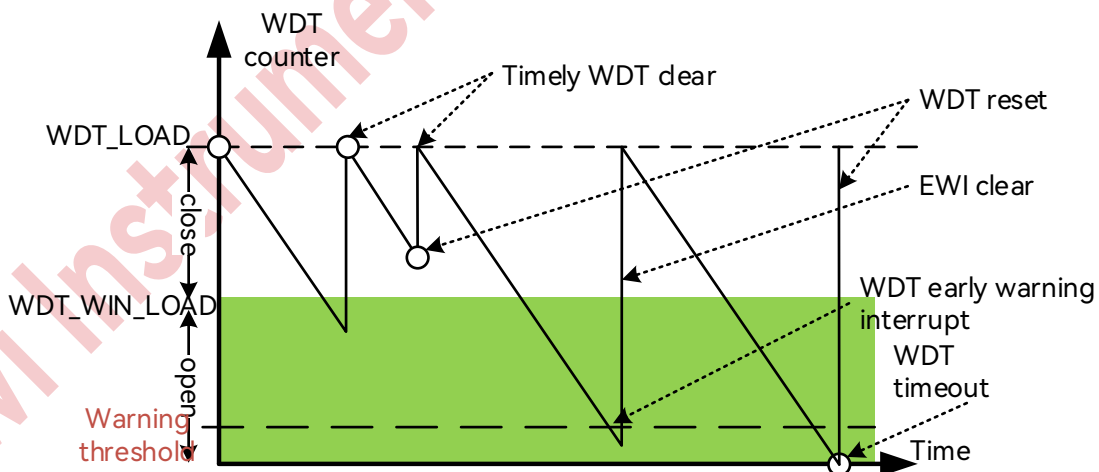


图 87 WDT 窗口模式 2

13.4.3 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。

13.4.4 调试模式

当 CPU 进入调试模式后, 看门狗可根据控制位 DBG_WDT_STO (0x4000_44C0[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

13.4.5 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 (0x400042A8[21:20]), 如下所示, 具体操作详见系统寄存器。

- 00: LIRC_CLK (32kHz)
- 01: HIRC (8MHz)

13.5 寄存器描述

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 146 寄存器列表

绝对地址	寄存器列表	位宽	访问	复位值
0x40002C00	WDT 控制寄存器	32	r/w	-
0x40002C04	WDT 加载寄存器	32	r/w	-
0x40002C08	WDT 窗口加载寄存器	32	r/w	0x00FF_FFFF
0x40002C0C	WDT 状态寄存器	32	r/w	0x0000_0000
0x40002C10	WDT 计数器寄存器	32	ro	0x00FF_FFFF
0x40002C20	WDT 密钥寄存器	32	r/w	0x0000_0001

13.5.1 WDT 控制寄存 (WDT_CTRL)

偏移地址: 0x0000

表 147 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: 0x0
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

13.5.2 WDT 加载寄存器 (WDT_LOAD)

偏移地址: 0x0004

表 148 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: 当看门狗开启, 计数器会自动从该寄存器加载初值 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 更新该寄存器, 内部的计数器也会同步更新 该寄存器最小值为 1 复位值: 0xFFFFFF

13.5.3 WDT 窗口加载寄存器 (WDT_WIN_LOAD)

偏移地址: 0x0008

表 149 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 窗口值。 复位值: 0xFFFFFF

注: 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。

13.5.4 WDT 状态寄存器 (WDT_ST)

偏移地址: 0x000C

表 150 WDT 状态寄存器

位	访问	描述
31	r/w	WDT_EN_SYNC, 看门狗启用状态 0: 关闭 1: 开启 复位值: 0x0
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

位	访问	描述
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

13.5.5 WDT 计数器寄存器 (WDT_VAL)

偏移地址: 0x0010

表 151 WDT 计数器寄存器

位	访问	描述
31:24	res	Reserved 复位值: 0x0
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: 0xFFFFFFFF
2:0	res	Reserved 复位值: 0x0

13.5.6 WDT 秘钥寄存器 (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 152 WDT 秘钥寄存器 (写操作)

位	访问	描述
31:0	w	寄存器 (WDT_CTRL、WDT_LOAD、WDT_WIN_LOAD) 写控制 写秘钥 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 153 WDT 秘钥寄存器 (读操作)

位	访问	描述
31:1	res	Reserved 复位值: 0x0
0	r	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

14 串行通信接口 (SCI)

14.1 概述

KPM32K07XX 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，同步通信接口支持 SPI 或者简易 IIC 协议。

14.2 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)

- 数据长度 7/8/9 bit
- 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
- 1/2 bit 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

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

14.3 模块框图

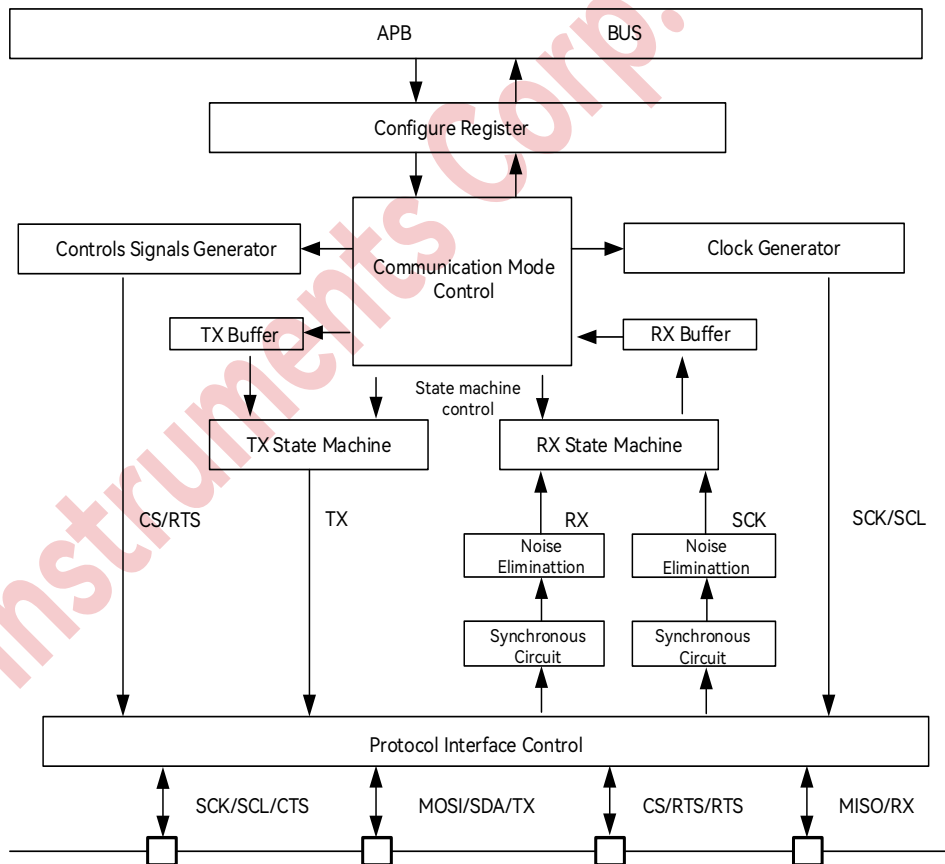


图 88 串行通信接口架构图

14.4 功能描述

14.4.1 工作模式

串行通信接口可配置为以下 3 种通信模式。

- SPI 模式
- UART 模式
- 简易 I²C 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。

14.4.2 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 154 接口功能复用

通信模式		SCK/SCL	MOSI/SDA/TX	MISO/RX	CS
SPI	功能	SCK	MOSI	MISO	CS
	方向	I/O	I/O	I/O	I/O
UART	功能		TX	RX	
	方向		Output	Input	
IIC	功能	SCL	SDA		
	方向	Output	I/O		

14.4.3 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。

14.4.3.1 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。

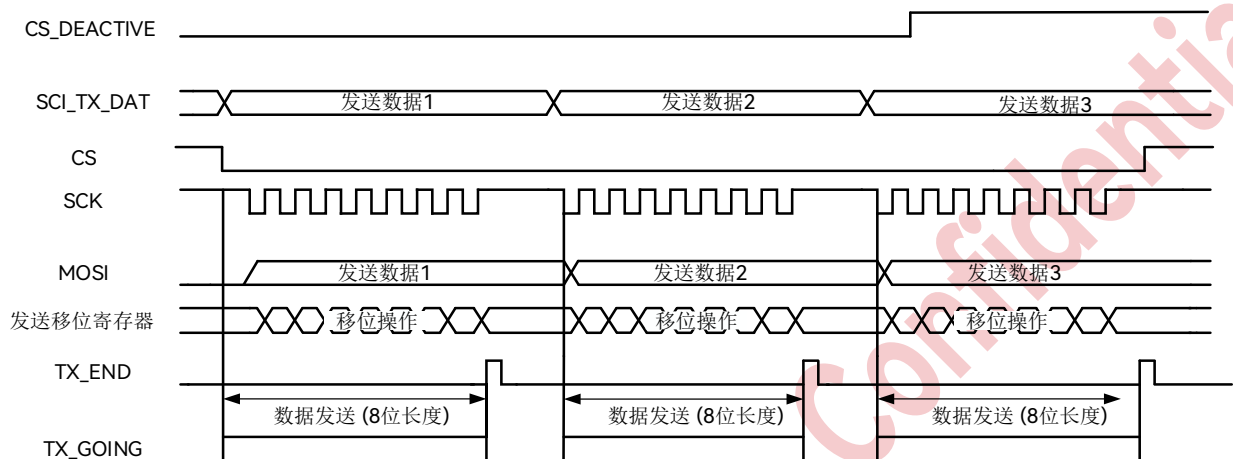


图 89 主发送 (单笔模式) 时序图

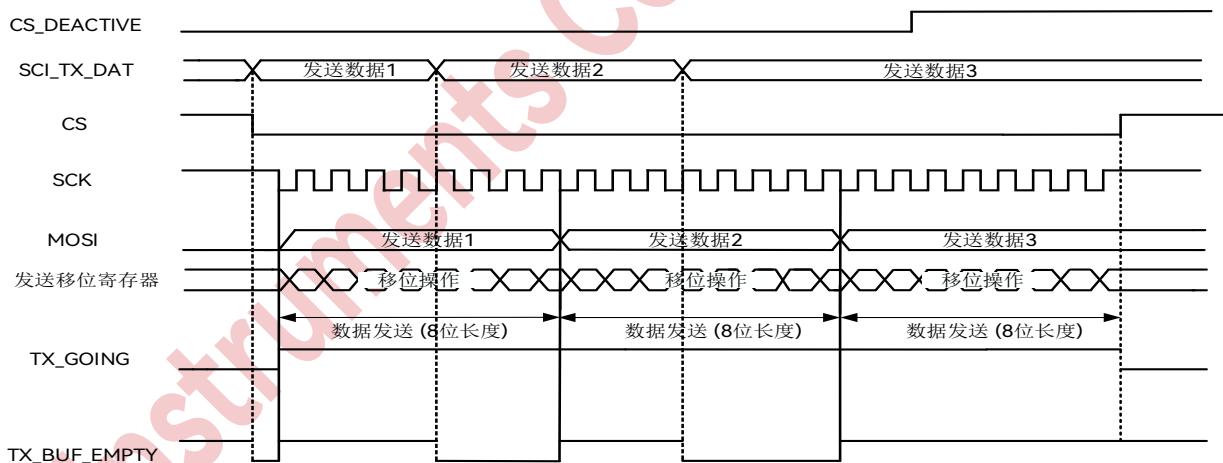


图 90 主发送 (连续模式) 时序图

14.4.3.2 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。

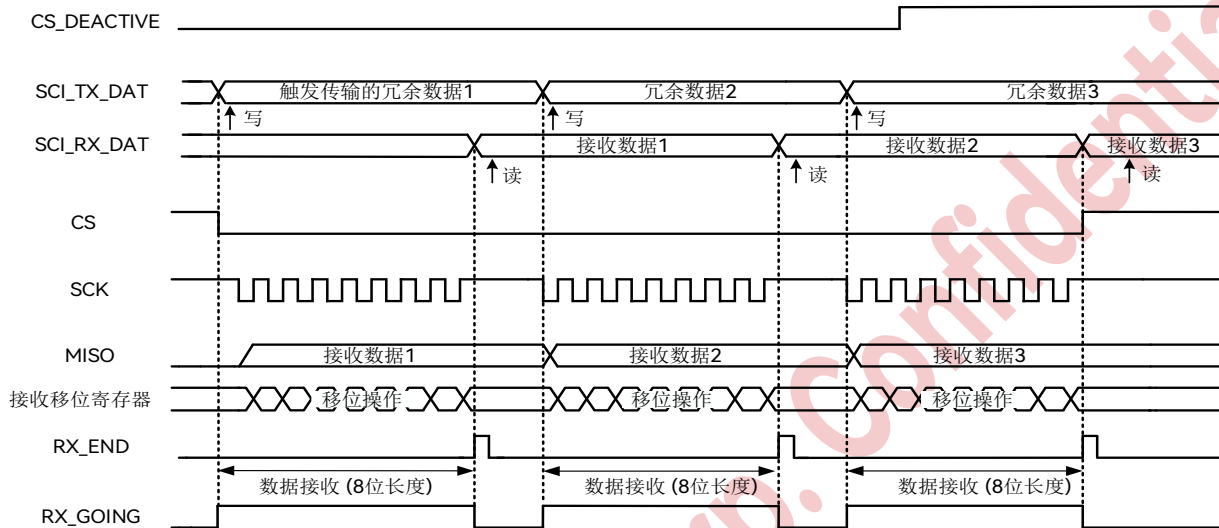


图 91 主接收 (单次模式) 时序图

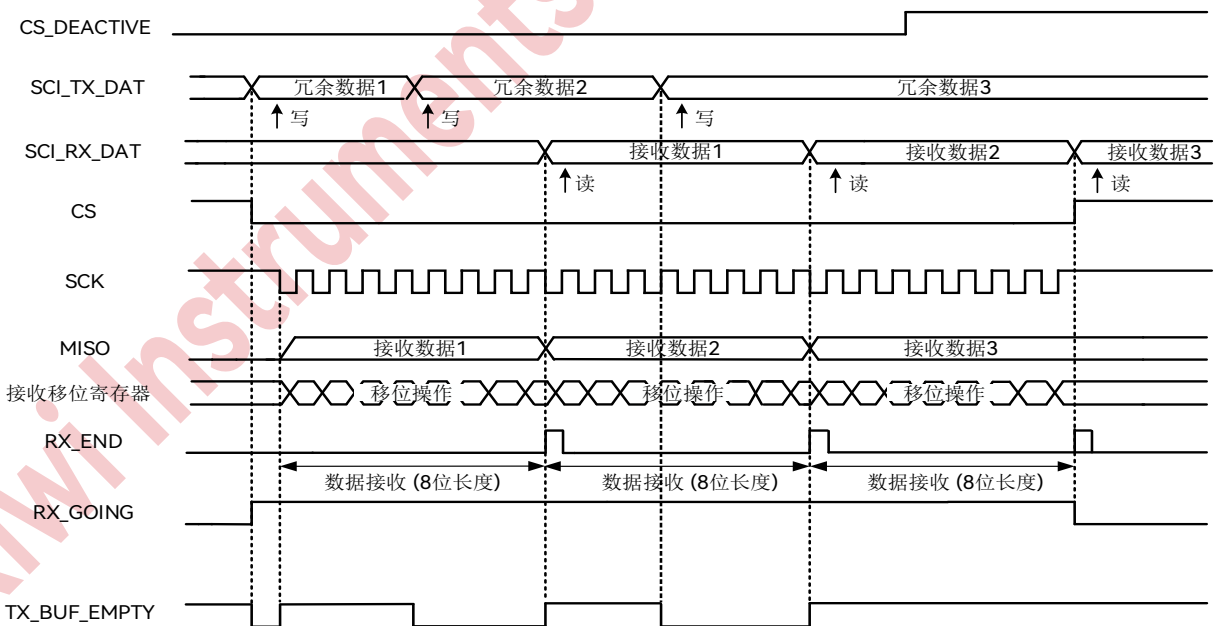


图 92 主接收 (连续模式) 时序图

14.4.3.3 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。

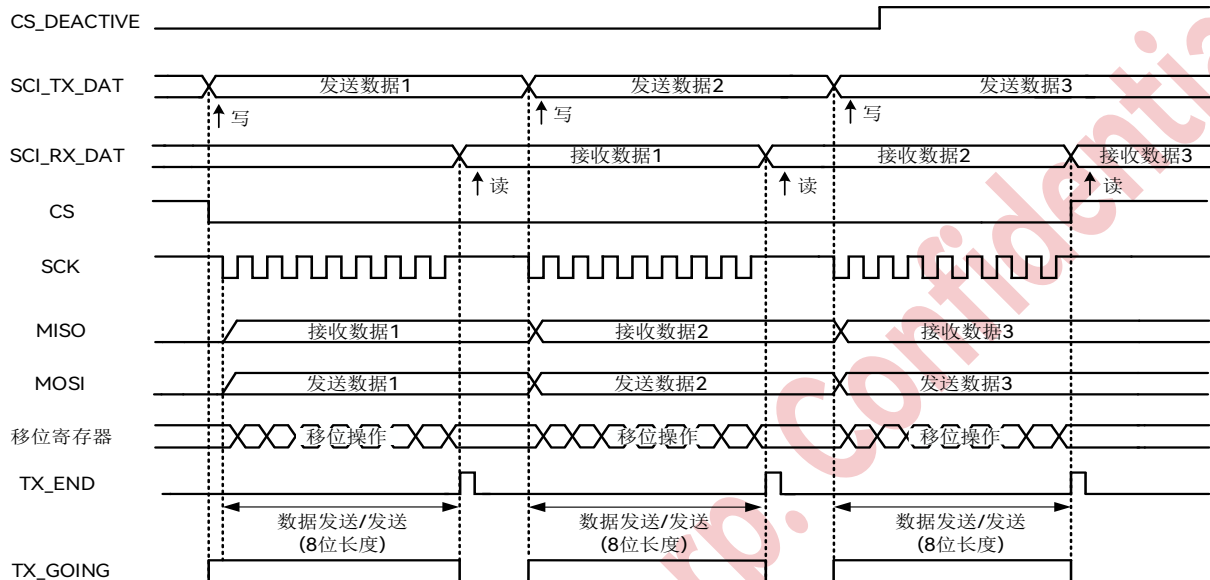


图 93 主发送接收 (单笔模式) 时序图

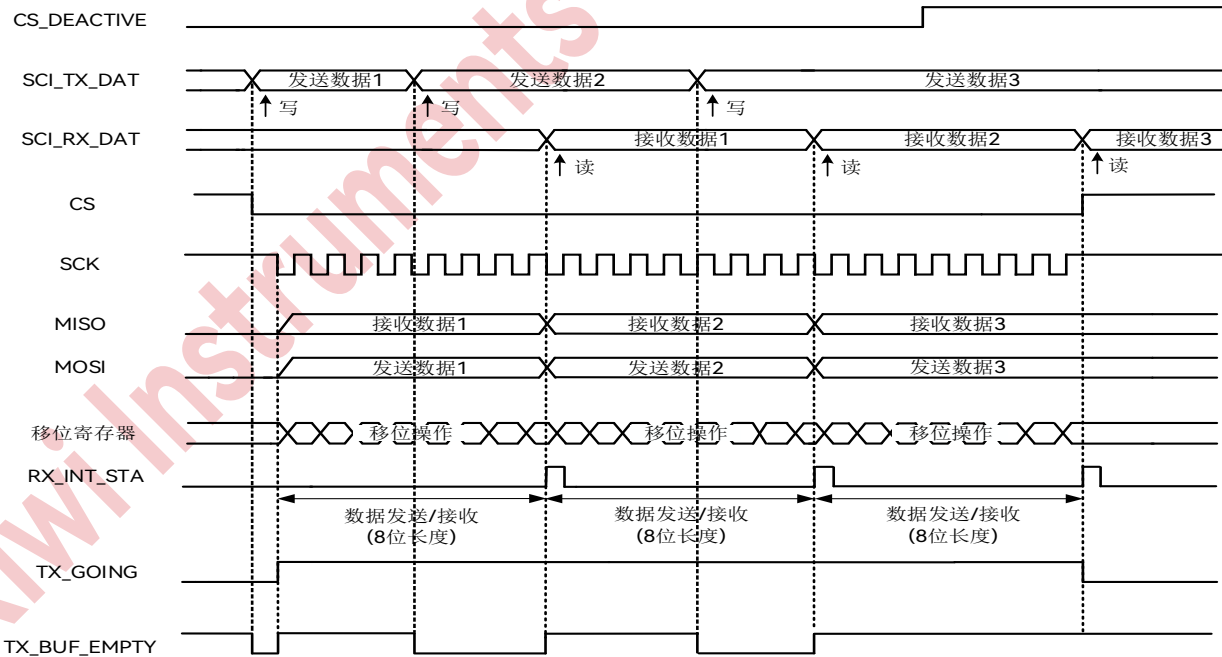


图 94 主发送接收 (连续模式) 时序图

14.4.3.4 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。

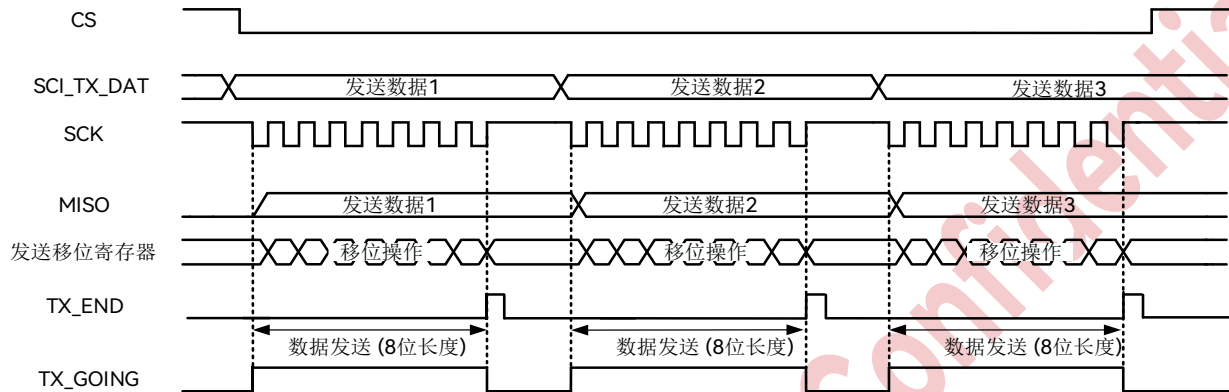


图 95 从发送 (单笔模式) 时序图

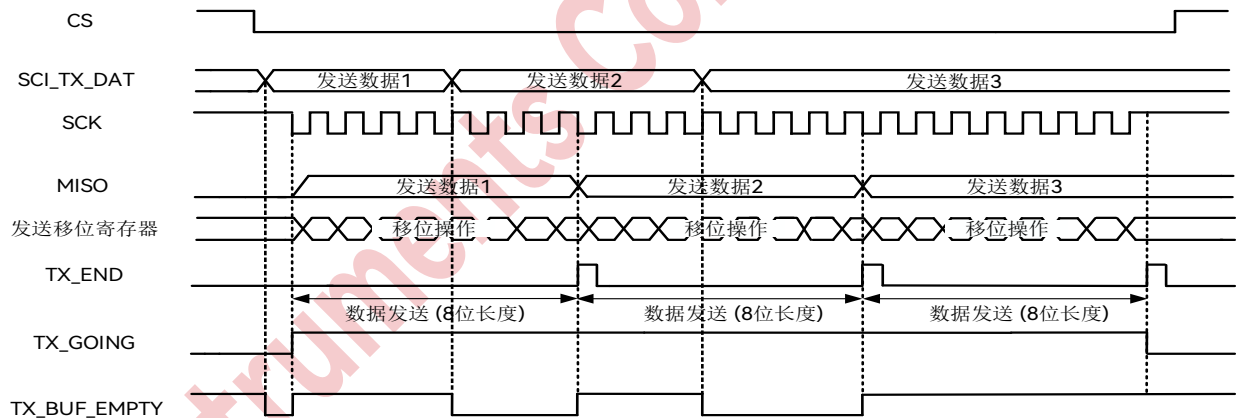


图 96 从发送 (连续模式) 时序图

14.4.3.5 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。

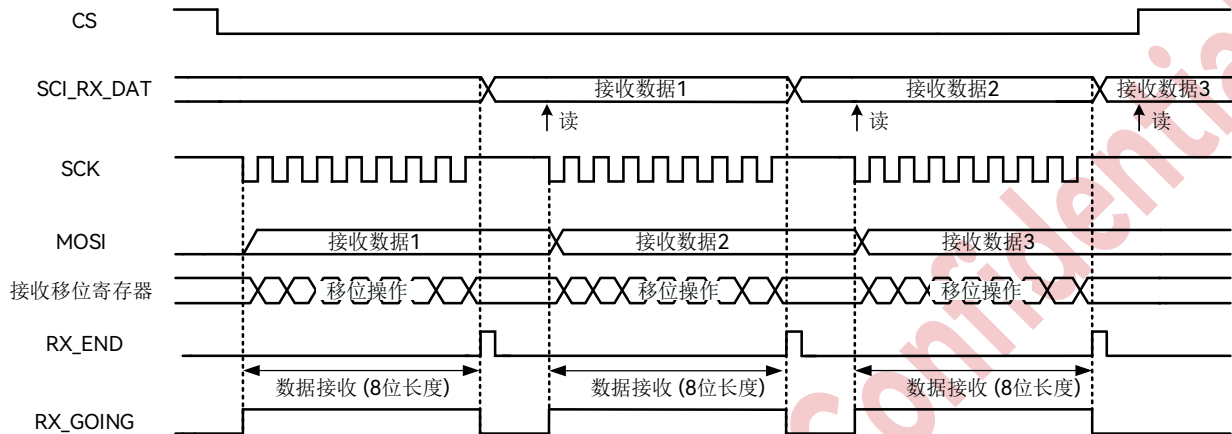


图 97 从接收 (单笔模式) 时序图

14.4.3.6 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。

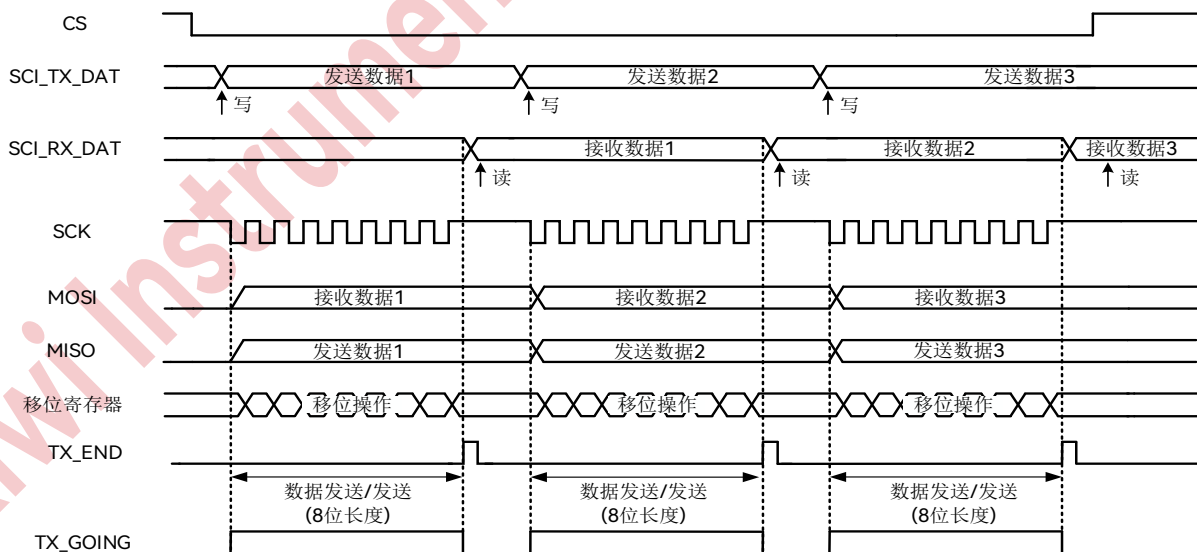


图 98 从发送接收 (单笔模式) 时序图

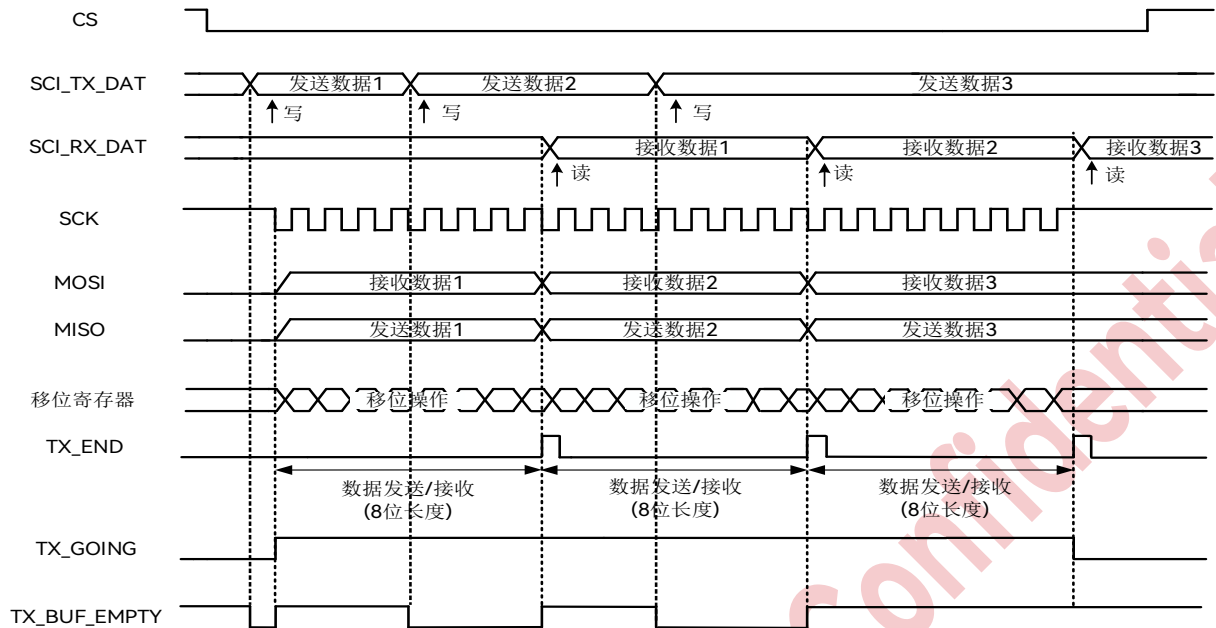


图 99 从发送接收 (连续模式) 时序图

14.4.3.7 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

$$OP_CLK_DIV = (SCI_BAUD.CLK_DVI1 + 1) \times (SCI_BAUD.CLK_DVI2 + 1)$$

$$\text{传输频率} = \frac{\text{系统时钟频率}}{2^{CLK_PRS_SEL} \times OP_CLK_DIV \times 2} \text{ (Hz)}$$

● 从设备

传输频率 = 由外部主设备提供的串行时钟 (SCK) 频率 (Hz)

注：当串行单元作为从设备时，最大支持传输时钟频率为 $(\text{系统时钟频率} / 2^{CLK_PRS_SEL}) \div 6$ 。

14.4.4 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) 在芯片与其他通信方之间 (以内部波特率) 异步传送。

14.4.4.1 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。

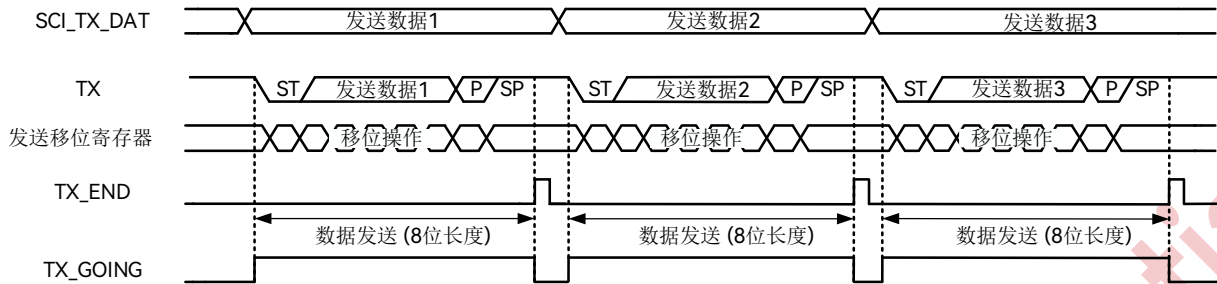


图 100 UART 发送 (单笔模式) 时序图

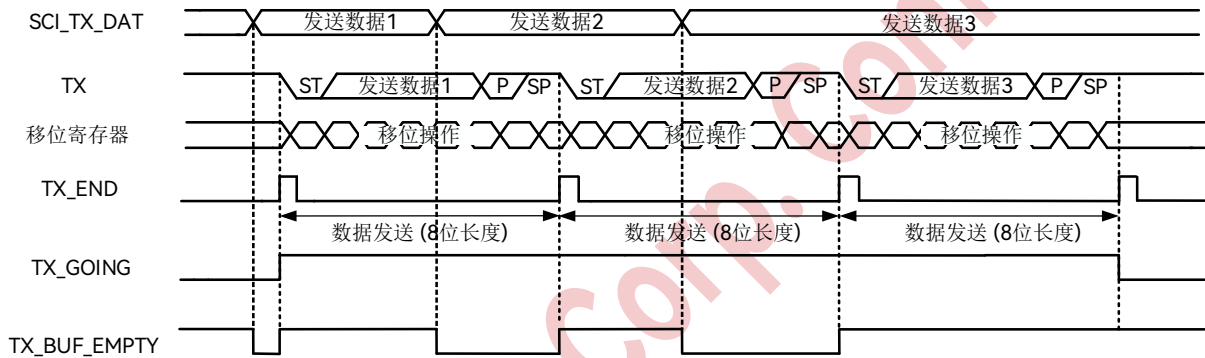


图 101 UART 发送 (连续模式) 时序图

14.4.4.2 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。

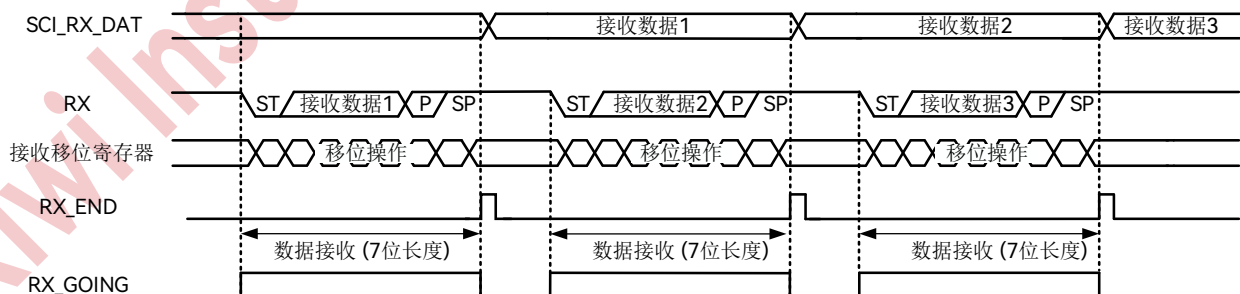


图 102 UART 接收 (单笔模式) 时序图

14.4.4.3 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

$$OP_CLK_DIV = (SCI_BAUD.CLK_DVI1 + 1) \times (SCI_BAUD.CLK_DVI2 + 1)$$

$$\text{波特率} = \frac{\text{系统时钟频率}}{2^{CLK_PRS_SEL} \times OP_CLK_DIV \times 2} \text{ (bps)}$$

注：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

$$\text{最大可接收波特率} = \frac{2 \times div \times N}{2 \times div \times N - div + 2} \times Brate$$

$$\text{最小可接收波特率} = \frac{2 \times div \times (N - 1)}{2 \times div \times N - div - 2} \times Brate$$

注：Brate：接收方的计算波特率值

div：OP_CLK_DIV

N：1 帧数据长度

14.4.5 简易 I²C 通信模式

这是一种时钟同步通信功能，通过两线：串行时钟线（SCL）和串行数据线（SDA）与两个或者多个器件进行通信。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。

14.4.5.1 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，相对应地产生起始，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。

14.4.5.2 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。

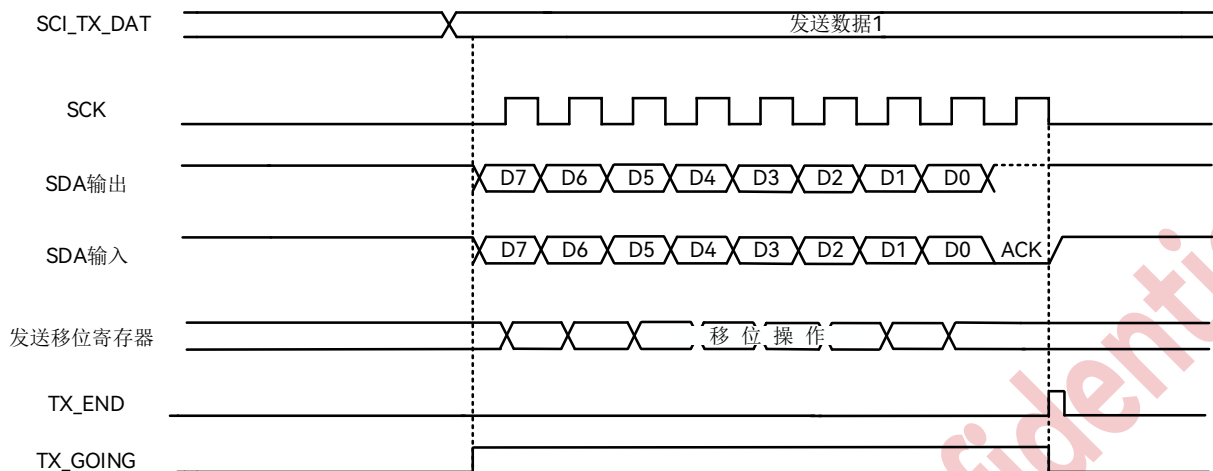


图 103 数据发送时序图

14.4.5.3 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。

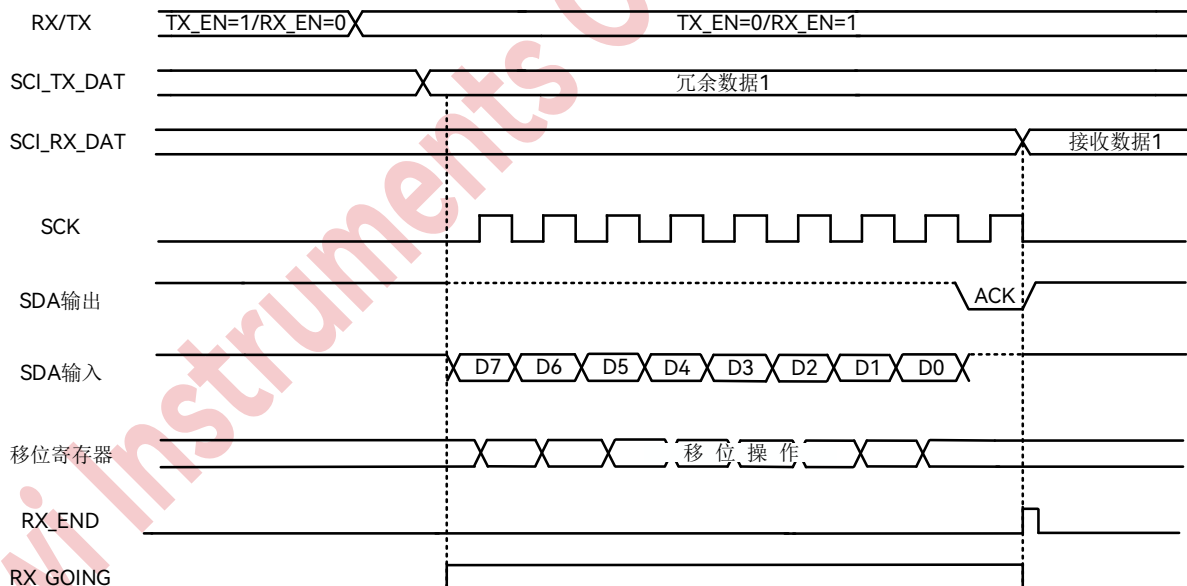


图 104 数据接收时序图

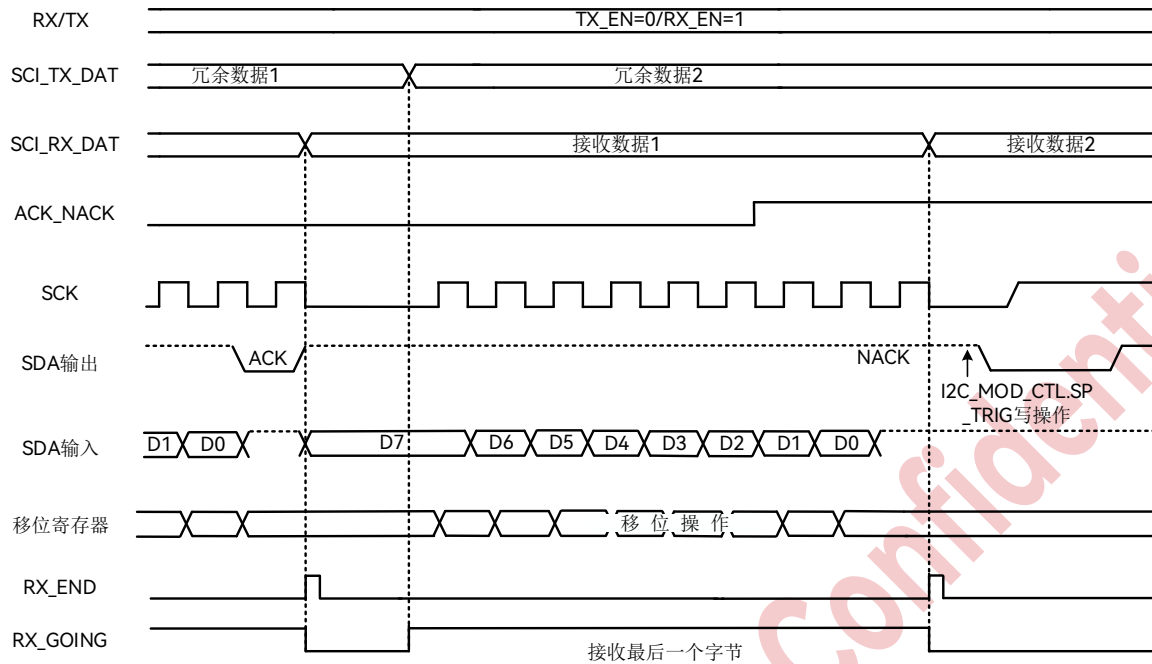


图 105 接收最后一笔数据时序图

14.4.5.4 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。

14.4.5.5 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。

$$OP_CLK_DIV = (SCI_BAUD.CLK_DVI1 + 1) \times (SCI_BAUD.CLK_DVI2 + 1)$$

$$\text{传输频率} = \frac{\text{系统时钟频率}}{2^{CLK_PRsx_SEL} \times OP_CLK_DIV \times 2} \text{ (Hz)}$$

14.4.6 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 155 SCI 中断号

名称	中断号
SCI0	12
SCI1	13
SCI2	14

14.5 寄存器描述

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，通过基址以区分。SCI0~2 基址分别为 0x4000_2800, 0x4000_3400, 0x4000_3800。

表 156 寄存器列表

偏移地址	寄存器列表	宽度	访问	复位值
0x00	时钟分频选择寄存器	32	r/w	0x00000000
0x04	波特率设置寄存器	32	r/w	0x00000000
0x10	模式控制寄存器	32	r/w	0x00000000
0x20	UART 模式寄存器	32	r/w	0x00000000
0x24	SPI 模式寄存器	32	r/w	0x00000000
0x28	IIC 模式寄存器	32	r/w	0x00000000
0x40	数据发送寄存器	32	r/w	0x00000000
0x50	数据接收寄存器	32	r/w	0x00000000
0x60	中断使能寄存器	32	r/w	0x00000000
0x64	中断状态寄存器	32	ro	0x00000000
0x68	中断状态清除寄存器	32	wo	0x00000000

14.5.1 时钟分频选择寄存器 (SCI_CLK_PRS)

偏移地址：0x0000

表 157 时钟分频选择寄存器描述

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 注: IPCLK=FCLK 复位值: 0x0

14.5.2 波特率设置寄存器 (SCI_BAUD)

偏移地址: 0x0004

表 158 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 通信时钟频率=CLK_PRS/((CLK_DIV1+1)×(CLK_DIV2+1)×2) 复位值: 0x0

14.5.3 模式控制寄存器 (SCI_MOD_CTL)

偏移地址: 0x0010

表 159 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 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 复位值: 0x0
24	r/w	DAT_NF_EN: 数据输入滤波使能 0: 关闭 1: 使能 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 复位值: 0x0
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: 固定高阻输出 复位值: 0x0

位	访问	描述
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: 固定高阻输出 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
7:3	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 其他值保留 复位值: 0x0

14.5.4 UART 模式寄存器 (UART_MOD_CTL)

偏移地址: 0x0020

表 160 UART 模式寄存器

位	访问	描述
31:5	res	Reserved 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

位	访问	描述
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

14.5.5 SPI 模式寄存器 (SPI_MOD_CTL)

偏移地址: 0x0024

表 161 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: CS 保持有效 1: 当前传输结束后, CS 无效 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 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

14.5.6 IIC 模式寄存器 (IIC_MOD_CTL)

偏移地址: 0x0028

表 162 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 (系统时钟频率/ $2^{\text{CLK_PRS_SEL}}$) 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 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

14.5.7 发送数据寄存器 (SCI_TX_DAT)

偏移地址: 0x0040

表 163 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

14.5.8 接收数据寄存器 (SCI_RX_DAT)

偏移地址: 0x0050

表 164 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

14.5.9 中断使能寄存器 (SCI_IE)

偏移地址: 0x0060

表 165 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 复位值: 0x0
9:8	res	Reserved 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

位	访问	描述
		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

14.5.10 中断状态寄存器 (SCI_STA)

偏移地址: 0x0064

表 166 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

位	访问	描述
9:8	res	Reserved 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/未检测到 ACK 错误中断标志 0: 无错误发生 1: 奇偶校验错误/未检测到 ACK 错误中断发生 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 注: 此位清除通过读取接收数据寄存器 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 注: 此位清除通过写入发送数据寄存器 复位值: 0x0

14.5.11 中断状态清除寄存器 (SCI_STA_CLR)

偏移地址: 0x0068

表 167 中断状态清除寄存器

位	访问	描述
31:12	res	Reserved 复位值: 0x0
11	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

位	访问	描述
		1: 清除发送完成中断标志 复位值: 0x0
9:8	res	Reserved 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/未检测到 ACK 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/未检测到 ACK 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	res	Reserved 复位值: 0x0
0	res	Reserved 复位值: 0x0

15 模拟电压比较器 (CMP)

15.1 概述

CMP 模块内含有 2 个模拟比较器、2 个 DAC 和 2 个可编程增益放大器 PGA。

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，每个比较器可根据使用需求选择模拟信号输入源。

15.2 主要功能

- 2 个模拟电压比较器
- 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
- 电压比较器支持数字滤波功能
- 电压比较器支持输出逻辑翻转
- 电压比较器支持可选择有效沿触发中断响应
- 电压比较器支持独立可编程 8 位 DAC 参考电压
- 2 个可编程增益放大器 (PGA)
- 可编程增益放大器输出逻辑支持数字滤波功能
- 支持 CMP、PGA 输出至 TIM

15.3 模块框图

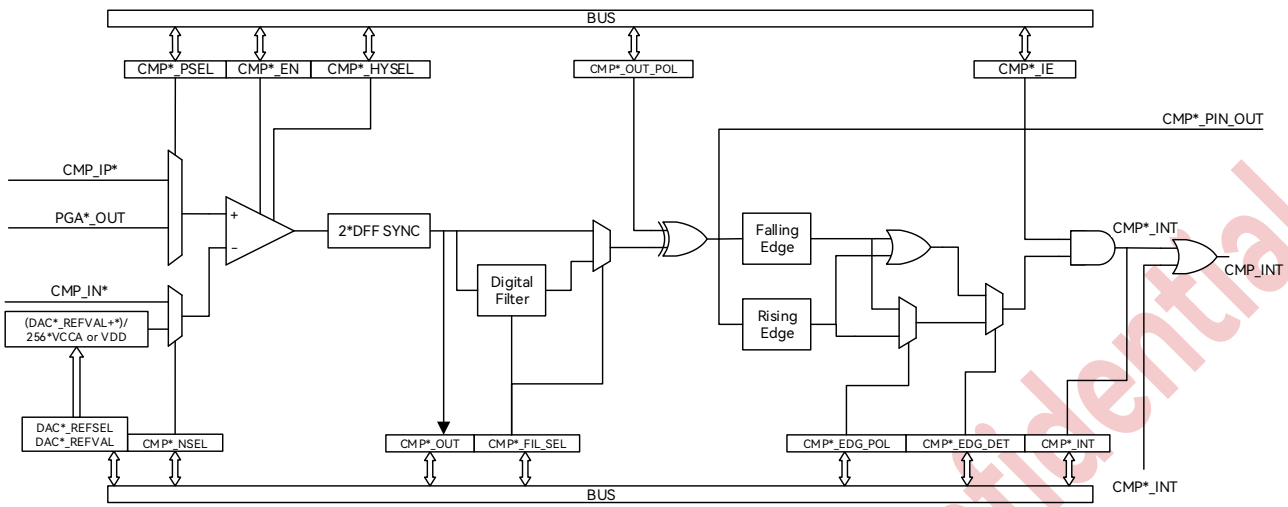


图 106 电压比较器框图

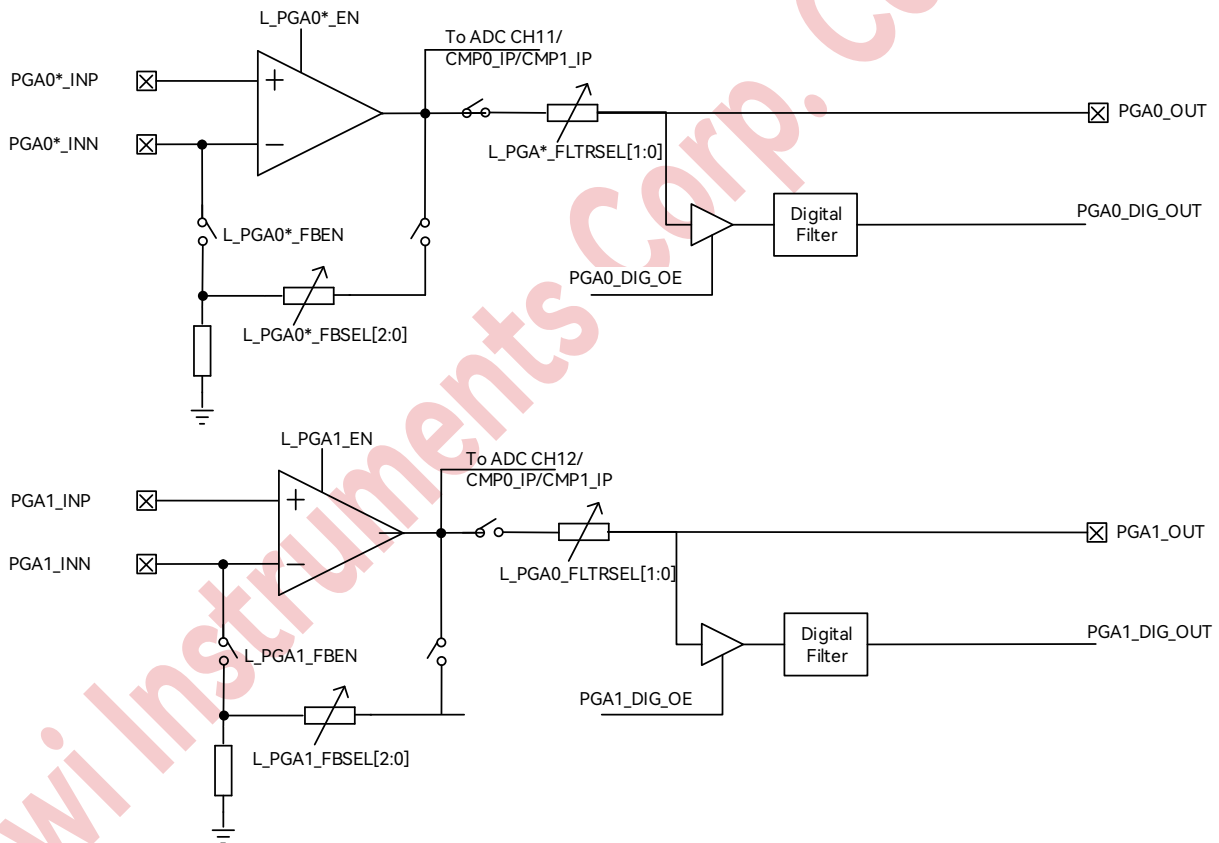


图 107 可编程增益放大器框图

15.4 功能描述

15.4.1 CMP 功能

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。

CMP 比较器 P 端可选择通过外部 IO 输出或 PGA 输出，N 端可选择 IO 输入或内部 DAC 输出，DAC 参考电压可选择 VCC 或 VDD。

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，当出现有效沿时会产生对应中断请求。

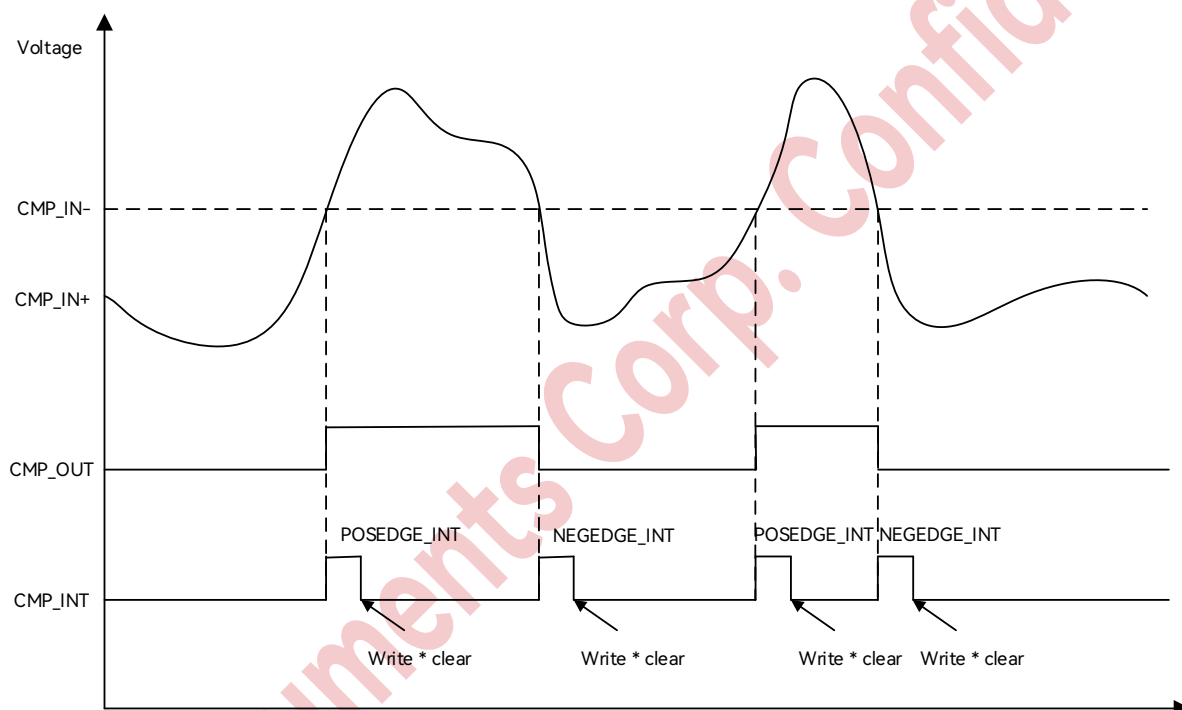


图 108 CMP 工作时序图

15.4.2 PGA 功能

15.4.2.1 PGA 放大增益选择

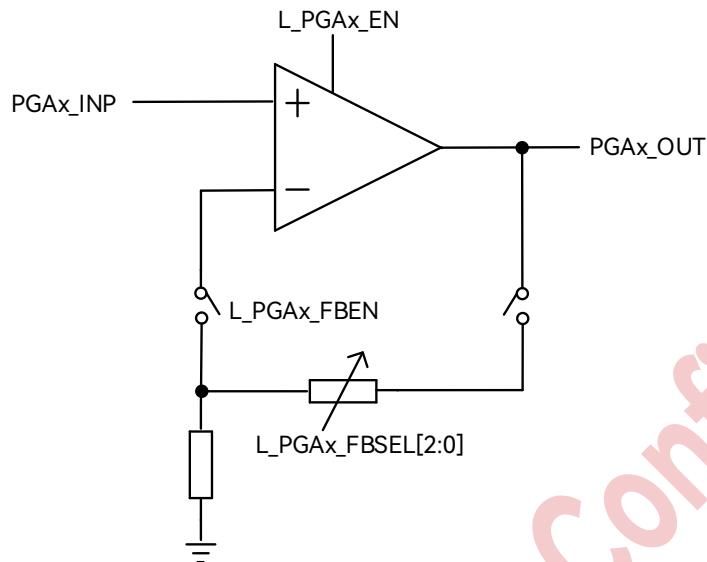


图 109 PGA 放大增益

使用内部反馈可对 PGA 的 P 端输入电压实现比例放大，具体放大比例如下表所示。

表 168 PGA 比例放大

L_PGAX_FBEN=1'b1		
L_PGAX_FBSEL<2:0>	Gain	$R_{network}=R1/R2$
000	2	11.1k/11.1k
001	3	22.2k/11.1k
010	4	33.3k/11.1k
011	8	77.7k/11.1k
100	16	82.5k/5.5k
Other	Forbidden	

15.4.2.2 差分输入单端输出

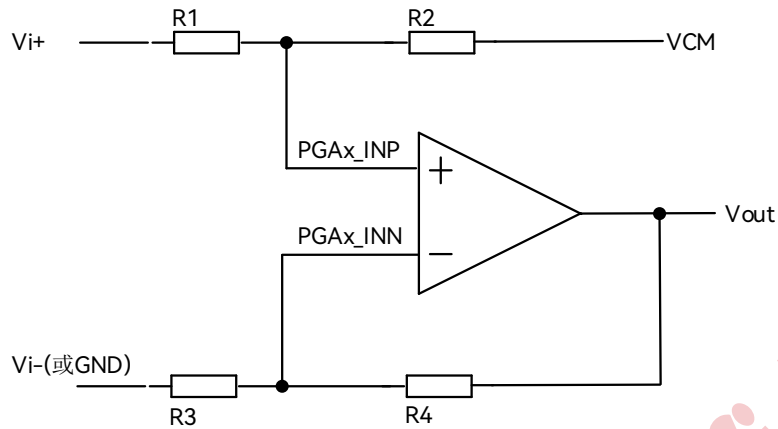


图 110 差分输入单端输出

放大关系如下：

如果 $R1=R3$, $R2=R4$, $V_{out}=R2/R1*(Vi+ - Vi-)+V_{CM}$

$$\begin{aligned} V_p &= \frac{R2}{R1 + R2} V_{ip} + \frac{R1}{R1 + R2} V_{cm} \\ &= V_n = \frac{R4}{R3 + R4} V_{in} + \frac{R3}{R3 + R4} V_{out} \\ V_{out} &= \frac{R3 + R4}{R3} \left(\frac{R2}{R1 + R2} V_{ip} - \frac{R4}{R3 + R4} V_{in} \right) + \frac{R3 + R4}{R3} * \frac{R1}{R1 + R2} V_{cm} \end{aligned}$$

15.4.2.3 共模偏置应用

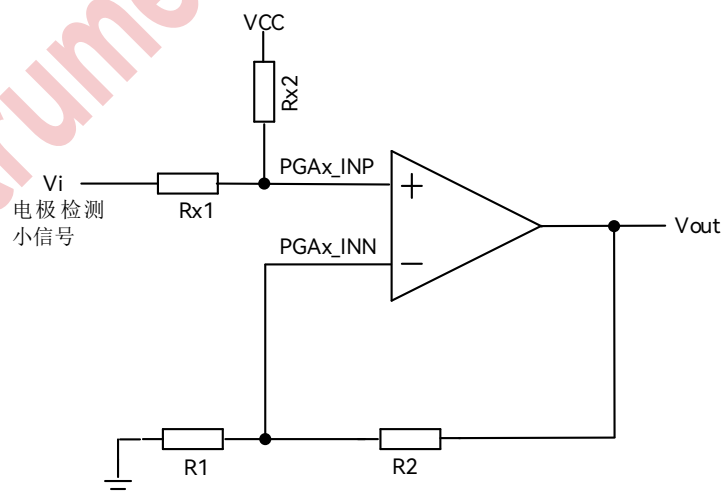


图 111 共模偏置应用

存在如下关系：

$$V_{out} = \left(\frac{R_{x1}}{R_{x1} + R_{x2}} V_{CC} + \frac{R_{x2}}{R_{x1} + R_{x2}} V_i \right) * \left(\frac{R_2}{R_1} + 1 \right)$$

若 $R_{x1}=10K$, $R_{x2}=50K$; $R_1=10K$, $R_2=20K$ (或者内部 $G=3$): $V_{out}=0.5V_{CC}+2.5V_i$

若 $R_{x1}=2K$, $R_{x2}=20K$; $R_1=2K$, $R_2=10K$: $V_{out}=0.545V_{CC}+5.454V_i$

若 $R_{x1}=2K$, $R_{x2}=28K$; $R_1=2K$, $R_2=14K$ (或者内部 $G=8$): $V_{out}=0.533V_{CC}+7.467V_i$

15.4.3 CMP 数字滤波

CMP 具有滤波功能，当无需该功能时可将其关闭。可以通过信号的宽度以 FCLK 作为基准，使用 CMPx_SMP_SEL 寄存器调整采样周期的 FCLK 时钟周期数，配置 CMPx_FLT_SEL 寄存器选择最小可以通过的信号宽度，可配置为 4、16、32、64、128、256 和 512 个采样周期。

PGA 在作为比较器使用时可配置 4、16、32、64、128、256 和 512 个采样周期，采样周期固定为 FCLK 时钟。

15.4.4 CMP 中断号

表 169 CMP 中断号

名称	中断号
CMP_INT	27

15.5 寄存器描述

CMP 寄存器的基地址为 0x4000_3000，下表为 CMP 的各控制寄存器描述。

表 170 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	CMP 模式控制寄存器	32	r/w	0x13021602
0x04	CMP 滤波控制寄存器	32	r/w	0x00000000
0x08	CMP 输出控制寄存器	32	r/w	0x00000000
0x0c	CMP 状态寄存器	32	r/w	0x00000000
0x10	DAC 控制寄存器	32	r/w	0x08000800
0x20	PGA0 控制寄存器	32	r/w	0x00000000
0x24	PGA1 控制寄存器	32	r/w	0x00000000
0x28	PGA 滤波控制寄存器	32	r/w	0x00000000
0x2c	PGA 状态寄存器	32	r/w	0x00000000
0x30	CMP 管脚使能	32	r/w	0x00000000
0x34	DAC 管脚使能	32	Wo	0x00000000
0x38	PGA0 管脚使能	32	r/w	0x00000000
0x3c	PGA1 管脚使能	32	r/w	0x00000000

15.5.1 CMP 模式控制寄存器 (CMP_MD_CTRL)

偏移地址: 0x0000

表 171 CMP 模式控制寄存器

位	访问	描述
31:29	res	Reserved 复位值: 0x0
28	r/w	CMP1_NSEL: 电压比较器 1 -N 端输入电压选择 0: 外部引脚 PA2 (ACMP1_IN) 1: 内部 DAC1 输出 $x/256 \times VCC$ or VDD ($x=0, 1, \dots, 255$) 复位值: 0x1
27:26	res	Reserved 复位值: 0x0
25:24	r/w	CMP1_PSEL: 电压比较器 1 -P 端输入电压选择 00: 外部引脚 PA3 (ACMP1_IP0) 01: 外部引脚 PA6 (ACMP1_IP1) 10: PGA0_OUT 11: PGA1_OUT 复位值: 0x3
23:20	res	Reserved 复位值: 0x0
19	r/w	CMP1_IBSEL: 电压比较器 1 基准电流选择 0: 4uA 1: 8uA 复位值: 0x0
18:17	r/w	CMP1_HYSEL: 电压比较器 1 迟滞选择 00: No hysteresis 01: 25mV 10: 50mV 11: 100mV 复位值: 0x1
16	r/w	CMP1_EN: 电压比较器 1 使能 0: 关闭电压比较器 1 1: 打开电压比较器 1 复位值: 0x0
15:13	res	Reserved 复位值: 0x0
12	r/w	CMP0_NSEL: 电压比较器 0 -N 端输入电压选择 0: 外部引脚 PB9 (XCMP0_IN) 1: 内部 DAC0 输出 $x/256 \times VCC$ or VDD ($x=0, 1, 2, \dots, 255$) 复位值: 0x1
11	res	Reserved 复位值: 0x0

位	访问	描述
10:8	r/w	CMP0_PSEL: 电压比较器 0 -P 端输入电压选择 000: 外部引脚 PA0 (ACMP0_IP0) 001: 外部引脚 PB8 (ACMP0_IP1) 010: 外部引脚 PB10 (ACMP0_IP2) 011: 外部引脚 PB7 (ACMP0_IP3) 100: PGA0_OUT 101: PGA1_OUT 110: VDD 111: VDD 复位值: 0x6
7:4	res	Reserved 复位值: 0x0
3	r/w	CMP0_IBSEL: 电压比较器 0 基准电流选择 0: 4μA 1: 8μA 复位值: 0x0
2:1	r/w	CMP0_HYSEL: 电压比较器 0 迟滞选择 00: No hysteresis 01: 25mV 10: 50mV 11: 100mV 复位值: 0x1
0	r/w	CMP0_EN: 电压比较器 0 使能 0: 关闭电压比较器 0 1: 打开电压比较器 0 复位值: 0x0

15.5.2 CMP 滤波控制寄存器 (CMP_FLT_CTRL)

偏移地址: 0x0004

表 172 CMP 滤波控制寄存器

位	访问	描述
31:28	res	Reserved 复位值: 0x0
27:24	r/w	CMP1_SMP_SEL: 采样周期选择 采样周期为 CMP1_SMP_SEL+1 个 CMP 时钟 复位值: 0x0
23	res	Reserved 复位值: 0x0
22:20	r/w	CMP1_FLT_SEL: 电压比较器 1 滤波选择 000: 无滤波 001: 4 采样周期 010: 16 采样周期 011: 32 采样周期

位	访问	描述
		100: 64 采样周期 101: 128 采样周期 110: 256 采样周期 111: 512 采样周期 复位值: 0x0
19:18	res	Reserved 复位值: 0x0
17	r/w	CMP1_EDG_DET: 电压比较器 1 边沿检测 0: 单边沿检测 1: 双边沿检测 复位值: 0x0
16	r/w	CMP1_EDG_POL: 电压比较器 1 边沿检测极性 0: 上升沿检测 1: 下降沿检测 复位值: 0x0
15:12	res	Reserved 复位值: 0x0
11:8	r/w	CMP0_SMP_SEL: 采样周期选择 采样周期为 CMP0_SMP_SEL+1 个 CMP 时钟 复位值: 0x0
7	res	Reserved 复位值: 0x0
6:4	r/w	CMP0_FLT_SEL: 电压比较器 0 滤波选择 000: 无滤波 001: 4 周期 010: 16 周期 011: 32 周期 100: 64 周期 101: 128 周期 110: 256 周期 111: 512 周期 复位值: 0x0
3:2	res	Reserved 复位值: 0x0
1	r/w	CMP0_EDG_DET: 电压比较器 0 边沿检测 0: 单边沿检测 1: 双边沿检测 复位值: 0x0
0	r/w	CMP0_EDG_POL: 电压比较器 0 边沿检测极性 0: 上升沿检测 1: 下降沿检测 复位值: 0x0

15.5.3 CMP 输出控制寄存器 (CMP_OUT_CTRL)

偏移地址: 0x0008

表 173 滤波控制寄存器

位	访问	描述
31:18	res	Reserved 复位值: 0x0
17	r/w	CMP1_IE: 电压比较器 1 中断使能 0: 关闭电压比较器 1 中断 1: 打开电压比较器 1 中断 复位值: 0x0
16	r/w	CMP1_OUT_POL: 电压比较器 1 输出极性 0: 比较器 1 输出 1: 比较器 1 输出反相 复位值: 0x0
15:2	res	Reserved 复位值: 0x0
1	r/w	CMP0_IE: 电压比较器 0 中断使能 0: 关闭电压比较器 0 中断 1: 打开电压比较器 0 中断 复位值: 0x0
0	r/w	CMP0_OUT_POL: 电压比较器 0 输出极性 0: 比较器 0 输出 1: 比较器 0 输出反相 复位值: 0x0

15.5.4 CMP 状态寄存器 (CMP_ST)

偏移地址: 0x000C

表 174 CMP 状态寄存器

位	访问	描述
31:18	res	Reserved 复位值: 0x0
17	r/w	CMP1_INT: 比较器 1 中断标志 0: 未检测到 CMP1 电压符合条件 1: 检测到 CMP1 电压符合条件 该状态位写 1 清零 复位值: 0x0
16	ro	CMP1_OUT: 比较器 1 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
15:2	res	Reserved 复位值: 0x0
1	r/w	CMP0_INT: 比较器 0 中断标志 0: 未检测到 CMP0 电压符合条件 1: 检测到 CMP0 电压符合条件 该状态位写 1 清零 复位值: 0x0

位	访问	描述
0	ro	CMP0_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

15.5.5 DAC 控制寄存器 (DAC_CFG)

偏移地址: 0x0010

表 175 DAC 控制寄存器

位	访问	描述
31:28	res	Reserved 复位值: 0x0
27:20	r/w	DAC1_VAL: DAC1 配置数值 当 DAC1_REFSEL=0 时 $V_{out}=DAC1_VAL/256 \times VCC$ 当 DAC1_REFSEL=1 时 $V_{out}=DAC1_VAL/256 \times VDD$ 复位值: 0x80
19:18	res	Reserved 复位值: 0x0
17	r/w	DAC1_REFSEL: DAC1 参考电压选择 0: VCC 作为参考电压 1: VDD 作为参考电压 复位值: 0x0
16	ro	DAC1_EN: DAC1 使能 复位值: 0x0
15:12	res	Reserved 复位值: 0x0
11:4	r/w	DAC0_VAL: DAC0 配置数值 当 DAC0_REFSEL=0 时 $V_{out}=DAC0_VAL/256 \times VCC$ 当 DAC0_REFSEL=1 时 $V_{out}=DAC0_VAL/256 \times VDD$ 复位值: 0x80
3:2	res	Reserved 复位值: 0x0
1	r/w	DAC0_REFSEL: DAC0 参考电压选择 0: VCC 作为参考电压 1: VDD 作为参考电压 复位值: 0x0
0	ro	DAC0_EN: DAC0 使能 复位值: 0x0

15.5.6 PGA0 控制寄存器 (PGA0_CFG)

偏移地址: 0x0020

表 176 PGA0 控制寄存器

位	访问	描述
31:29	res	Reserved 复位值: 0x0
28:24	r/w	PGA0_TRIMN: PGA0 NMOS 校准偏移电压调节 位 20: 校准偏移电压符号 1: 负数 0: 正数 位 19:16: 校准偏移电压绝对值 0000: No calibration 0001: 0.602 / -0.573 0010: 1.174 / -1.146 0011: 1.735 / -1.705 0100: 2.281 / -2.251 0101: 2.813 / -2.784 0110: 3.334 / -3.305 0111: 3.843 / -3.814 1000: 4.341 / -4.311 1001: 4.827 / -4.798 1010: 5.304 / -5.275 1011: 5.771 / -5.742 1100: 6.228 / -6.199 1101: 6.676 / -6.647 1110: 7.114 / -7.087 1111: 7.540 / -7.518 复位值: 0x0
23:21	res	Reserved 复位值: 0x0
20:16	r/w	PGA0_TRIMP: PGA0 PMOS 校准偏移电压调节 位 20: 校准偏移电压符号 1: 负数 0: 正数 位 19:16: 校准偏移电压绝对值 0000: No calibration 0001: 0.602 / -0.573 0010: 1.175 / -1.146 0011: 1.734 / -1.705 0100: 2.280 / -2.250 0101: 2.813 / -2.784 0110: 3.334 / -3.305 0111: 3.844 / -3.815 1000: 4.343 / -4.313 1001: 4.831 / -4.801 1010: 5.308 / -5.279 1011: 5.776 / -5.747 1100: 6.235 / -6.206 1101: 6.685 / -6.656

位	访问	描述
		1110: 7.126 / -7.097 1111: 7.559 / -7.530 复位值: 0x0
15:14	res	Reserved 复位值: 0x0
13:12	r/w	PGA0_FLTSEL: PGA0 输入滤波选择 00: 300Ω 01: 600Ω 10: 1000Ω 11: 2000Ω 复位值: 0x0
11:9	res	Reserved 复位值: 0x0
8	r/w	PGA0_DIG_OE: PGA0 数字逻辑输出使能 0: 关闭 PGA0 逻辑输出 1: 打开 PGA0 逻辑输出 复位值: 0x0
7	res	Reserved 复位值: 0x0
6:4	r/w	PGA0_FBSEL: PGA0 反馈增益选择 (只有当 PGA0_FBEN=1 时有效) 000: GAIN=2 001: GAIN=3 010: GAIN=4 011: GAIN=8 100: GAIN=16 Other: Reserved 复位值: 0x0
3:2	res	Reserved 复位值: 0x0
1	r/w	PGA0_FBEN: PGA0 增益反馈使能 0: 关闭 1: 开启 复位值: 0x0
0	r/w	PGA0_EN: PGA0 控制使能 0: 关闭 1: 开启 复位值: 0x0

15.5.7 PGA1 控制寄存器 (PGA1_CFG)

偏移地址: 0x0024

表 177 PGA1 控制寄存器

位	访问	描述
31:29	res	Reserved 复位值: 0x0
28:24	r/w	PGA1_TRIMN: PGA1 NMOS 校准偏移电压调节 位 20: 校准偏移电压符号 1: 负数 0: 正数 位 19:16: 校准偏移电压绝对值 0000: No calibration 0001: 0.602 / -0.573 0010: 1.174 / -1.146 0011: 1.735 / -1.705 0100: 2.281 / -2.251 0101: 2.813 / -2.784 0110: 3.334 / -3.305 0111: 3.843 / -3.814 1000: 4.341 / -4.311 1001: 4.827 / -4.798 1010: 5.304 / -5.275 1011: 5.771 / -5.742 1100: 6.228 / -6.199 1101: 6.676 / -6.647 1110: 7.114 / -7.087 1111: 7.540 / -7.518 复位值: 0x0
23:21	res	Reserved 复位值: 0x0
20:16	r/w	PGA1_TRIMP: PGA1 PMOS 校准偏移电压调节 位 20: 校准偏移电压符号 1: 负数 0: 正数 位 19:16: 校准偏移电压绝对值 0000: No calibration 0001: 0.602 / -0.573 0010: 1.175 / -1.146 0011: 1.734 / -1.705 0100: 2.280 / -2.250 0101: 2.813 / -2.784 0110: 3.334 / -3.305 0111: 3.844 / -3.815 1000: 4.343 / -4.313 1001: 4.831 / -4.801 1010: 5.308 / -5.279 1011: 5.776 / -5.747 1100: 6.235 / -6.206 1101: 6.685 / -6.656

位	访问	描述
		1110: 7.126 / -7.097 1111: 7.559 / -7.530 复位值: 0x0
15:14	res	Reserved 复位值: 0x0
13:12	r/w	PGA1_FLTSEL: PGA1 输入滤波选择 00: 300Ω 01: 600Ω 10: 1000Ω 11: 2000Ω 复位值: 0x0
11:9	res	Reserved 复位值: 0x0
8	r/w	PGA1_DIG_OE: PGA1 数字逻辑输出使能 0: 关闭 PGA1 逻辑输出 1: 打开 PGA1 逻辑输出 复位值: 0x0
7	res	Reserved 复位值: 0x0
6:4	r/w	PGA1_FBSEL: PGA1 反馈增益选择(只有当 PGA1_FBEN=1 时有效) 000: GAIN=2 001: GAIN=3 010: GAIN=4 011: GAIN=8 100: GAIN=16 Other: Reserved 复位值: 0x0
3:2	res	Reserved 复位值: 0x0
1	r/w	PGA1_FBEN: PGA1 增益反馈使能 0: 关闭 1: 开启 复位值: 0x0
0	r/w	PGA1_EN: PGA1 控制使能 0: 关闭 1: 开启 复位值: 0x0

15.5.8PGA 滤波控制寄存器 (PGA_FLT_CTRL)

偏移地址: 0x0028

表 178 PGA 滤波控制寄存器

位	访问	描述
31:19	res	Reserved 复位值: 0x0
18:16	r/w	PGA1_FLT_SEL: 电压比较器 1 滤波选择 000: 无滤波

位	访问	描述
		001: 4 周期 010: 16 周期 011: 32 周期 100: 64 周期 101: 128 周期 110: 256 周期 111: 512 周期 复位值: 0x0
15:3	res	Reserved 复位值: 0x0
2:0	r/w	PGA0_FLT_SEL: 电压比较器 0 滤波选择 000: 无滤波 001: 4 周期 010: 16 周期 011: 32 周期 100: 64 周期 101: 128 周期 110: 256 周期 111: 512 周期 复位值: 0x0

15.5.9 PGA 状态寄存器 (PGA_ST)

偏移地址: 0x002C

表 179 PGA 状态寄存器

位	访问	描述
31:17	res	Reserved 复位值: 0x0
16	ro	PGA1_OUT: 放大器 1 实时逻辑输出 0: P 端电压 < N 端电压 1: P 端电压 > N 端电压 复位值: 0x0
15:1	res	Reserved 复位值: 0x0
0	ro	PGA0_OUT: 放大器 0 实时逻辑输出 0: P 端电压 < N 端电压 1: P 端电压 > N 端电压 复位值: 0x0

15.5.10 CMP 管脚使能 (CMP_ASEL)

偏移地址: 0x0030

表 180 CMP 管脚使能

位	访问	描述
31:25	res	Reserved 复位值: 0x0

位	访问	描述
24	r/w	CMP1_INN0_ASEL: 比较器 1N 端端口输入 0 使能 (PA2) 0: 关闭 1: 打开 复位值: 0x0
23:18	res	Reserved 复位值: 0x0
17	r/w	CMP1_INP1_ASEL: 比较器 1P 端端口输入 1 使能 (PA6) 0: 关闭 1: 打开 复位值: 0x0
16	r/w	CMP1_INP0_ASEL: 比较器 1P 端端口输入 0 使能 (PA3) 0: 关闭 1: 打开 复位值: 0x0
15:9	res	Reserved 复位值: 0x0
8	r/w	CMP0_INN0_ASEL: 比较器 0N 端端口输入 0 使能 (PB9) 0: 关闭 1: 打开 复位值: 0x0
7:4	res	Reserved 复位值: 0x0
3	r/w	CMP0_INP3_ASEL: 比较器 0P 端端口输入 3 使能 (PB7) 0: 关闭 1: 打开 复位值: 0x0
2	r/w	CMP0_INP2_ASEL: 比较器 0P 端端口输入 2 使能 (PB10) 0: 关闭 1: 打开 复位值: 0x0
1	r/w	CMP0_INP1_ASEL: 比较器 0P 端端口输入 1 使能 (PB8) 0: 关闭 1: 打开 复位值: 0x0
0	r/w	CMP0_INP0_ASEL: 比较器 0P 端端口输入 0 使能 (PA0) 0: 关闭 1: 打开 复位值: 0x0

15.5.11 PGA0 管脚使能 (PGA0_ASEL)

偏移地址: 0x0038

表 181 PGA0 管脚使能

位	访问	描述
31:9	Res	Reserved 复位值: 0x0
8	r/w	PGA0_OUT_ASEL: 放大器 0 端口输出 0 使能 (PA3)

位	访问	描述
		0: 关闭 1: 打开 复位值: 0x0
7:5	Res	Reserved 复位值: 0x0
4	r/w	PGA0_INN0_ASEL: 放大器 0N 端端口输入 0 使能 (PA4) 0: 关闭 1: 打开 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	PGA0_INP0_ASEL: 放大器 0P 端端口输入 0 使能 (PA5) 0: 关闭 1: 打开 复位值: 0x0

15.5.12 PGA1 管脚使能 (PGA1_ASEL)

偏移地址: 0x003C

表 182 PGA1 管脚使能

位	访问	描述
31:9	Res	Reserved 复位值: 0x0
8	r/w	PGA1_OUT_ASEL: 放大器 1 端口输出 0 使能 (PA1) 0: 关闭 1: 打开 复位值: 0x0
7:5	Res	Reserved 复位值: 0x0
4	r/w	PGA1_INN0_ASEL: 放大器 1N 端端口输入 0 使能 (PB10) 0: 关闭 1: 打开 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	PGA1_INP0_ASEL: 放大器 1P 端端口输入 0 使能 (PA0) 0: 关闭 1: 打开 复位值: 0x0

16 事件系统 (EBUS)

16.1 概述

事件系统能够通过各种不同的编程配置，灵活地将设备的输入、输出以及内部外设事件进行灵活的连接。

事件系统主要包括以下两部分：IO 检测通道、输出通道。

IO 检测通道可以对芯片管脚的电平输入进行目标边沿检测，并发出中断事件；输出通道事件可以将 IO 输入或外设事件与输出通道的 IO 输出进行连接，达到输出/输入重定向的目的。

16.2 主要功能

- 时钟与系统时钟同步
- 支持最多 16 个独立的 IO 检测通道
- 支持最多 6 个独立可配置的输出通道
- 支持软件触发脉冲事件
- 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

16.3 模块框图

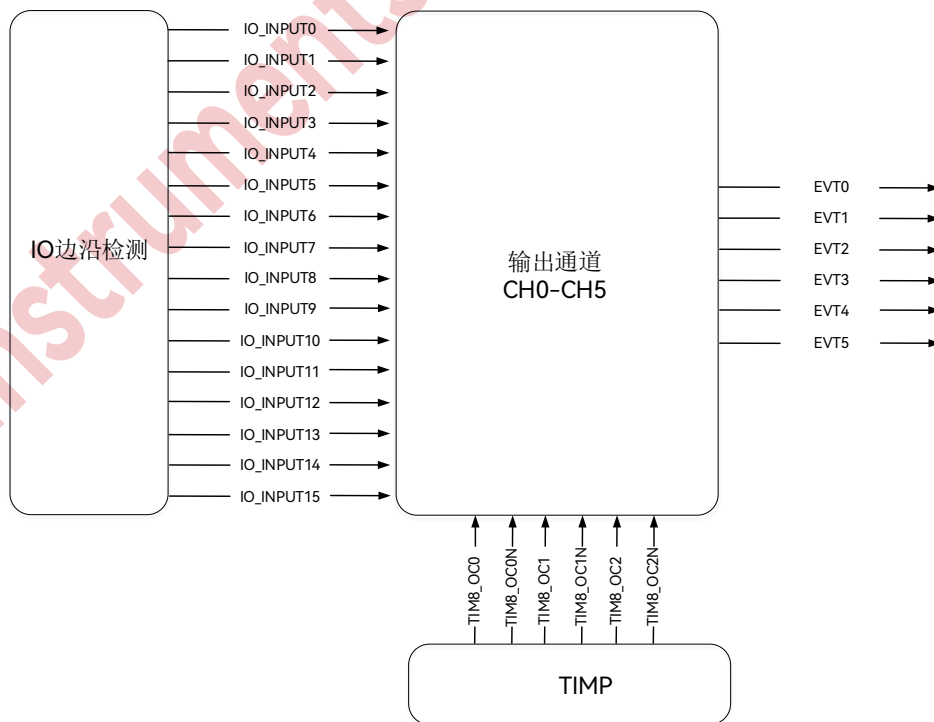


图 112 EBUS 模块框图

16.4 功能描述

16.4.1 IO 检测通道

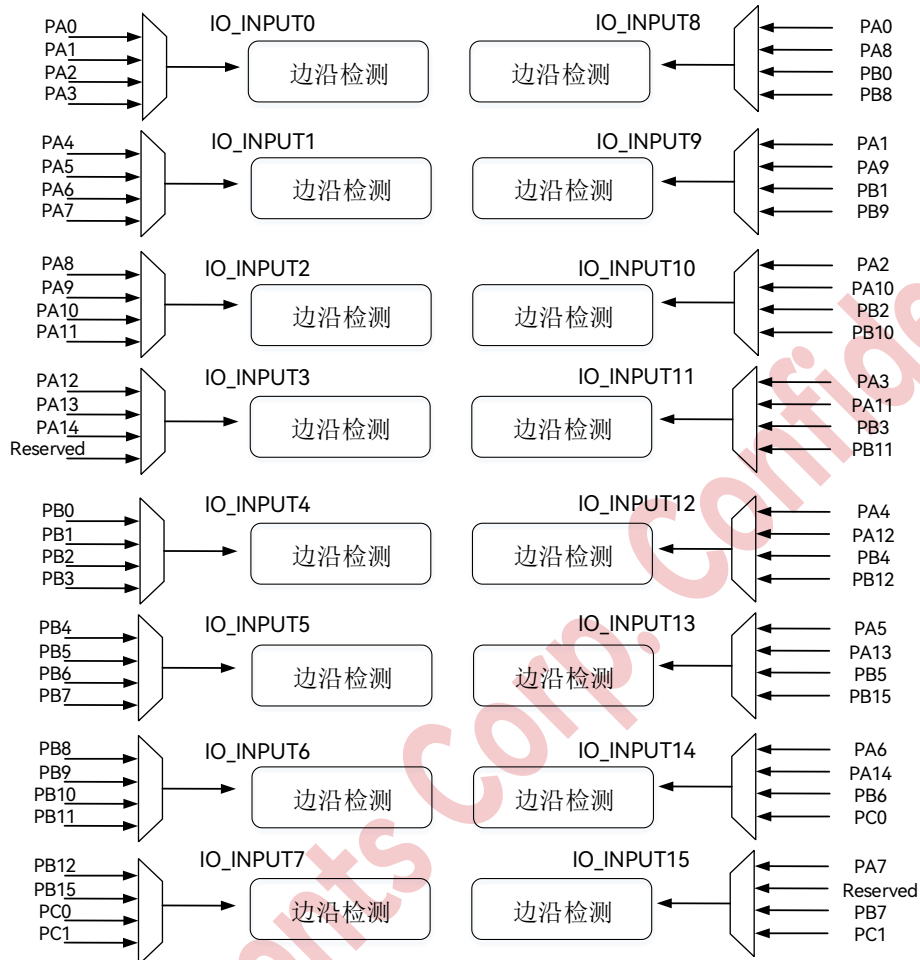


图 113 EBUS IO 检测通道架构图

事件系统包含 16 路 IO 检测通道，每个通道可以独立地对 IO 输入源进行边沿检测，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
- 输入电平的上升沿或下降沿检测

在 IO 检测通道设置寄存器中，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 检测结果会以中断或事件的方式发出。

每个 IO 检测通道的 IO 输入源通过 IO 通道设置寄存器进行配置，同时，配置后得到的 IO 输入源 (IO_INPUT0-IO_INPUT15)，可以作为 IO 输入事件源连接到事件通道。

16.4.2 通道事件输入源

表 183 IO 检测通道输入源

	Mux_in0	Mux_in1	Mux_in2	Mux_in3
检测通道 0	PA0	PA1	PA2	PA3
检测通道 1	PA4	PA5	PA6	PA7
检测通道 2	PA8	PA9	PA10	PA11
检测通道 3	PA12	PA13	PA14	Reserved
检测通道 4	PB0	PB1	PB2	PB3
检测通道 5	PB4	PB5	PB6	PB7
检测通道 6	PB8	PB9	PB10	PB11
检测通道 7	PB12	PB15	PC0	PC1
检测通道 8	PA0	PA8	PB0	PB8
检测通道 9	PA1	PA9	PB1	PB9
检测通道 10	PA2	PA10	PB2	PB10
检测通道 11	PA3	PA11	PB3	PB11
检测通道 12	PA4	PA12	PB4	PB12
检测通道 13	PA5	PA13	PB5	PB15
检测通道 14	PA6	PA14	PB6	PC0
检测通道 15	PA7	Reserved	PB7	PC1

16.4.3 输出通道

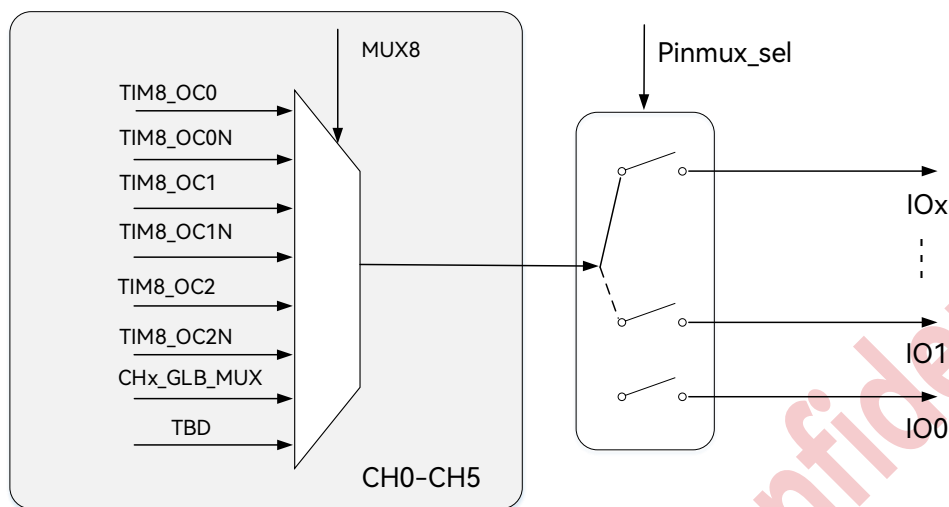


图 114 输出通道架构图

事件系统包含了 6 路输出通道，每个输出通道架构相同，由一个八选一的选择器构成，通过输出事件通道 x 设置寄存器的 `EBUS_OUTx_CFG.OUT_SRC_SEL` 位进行配置，选择器的输出信号会连接到 IO 管脚上进行片外输出。

输出通道的事件输入源由高级定时器的互补 PWM 波、事件通道的全局事件选择器输出以及其他信号组成，具体参考下表内容。

输出通道工作在异步模式下，事件在主设备和从设备之间的传播不经过事件系统的干涉，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。

16.4.4 通道事件输入源

表 184 输出通道输入源

	输出通道 0	输出通道 1	输出通道 2	输出通道 3	输出通道 4	输出通道 5
Mux_in0	TIM8_OC0	TIM8_OC0	TIM8_OC0	TIM8_OC0	TIM8_OC0	TIM8_OC0
Mux_in1	TIM8_OC0N	TIM8_OC0N	TIM8_OC0N	TIM8_OC0N	TIM8_OC0N	TIM8_OC0N
Mux_in2	TIM8_OC1	TIM8_OC1	TIM8_OC1	TIM8_OC1	TIM8_OC1	TIM8_OC1
Mux_in3	TIM8_OC1N	TIM8_OC1N	TIM8_OC1N	TIM8_OC1N	TIM8_OC1N	TIM8_OC1N
Mux_in4	TIM8_OC2	TIM8_OC2	TIM8_OC2	TIM8_OC2	TIM8_OC2	TIM8_OC2
Mux_in5	TIM8_OC2N	TIM8_OC2N	TIM8_OC2N	TIM8_OC2N	TIM8_OC2N	TIM8_OC2N
Mux_in6	IO_INPUT0	IO_INPUT1	IO_INPUT2	IO_INPUT3	IO_INPUT4	IO_INPUT5
Mux_in7	IO_INPUT6	IO_INPUT7	SOC0_EVT_SEL	SOC1_EVT_SEL	SOC2_EVT_SEL	SOC3_EVT_SEL

16.4.5 软件事件

CPU 可以控制输出通道的输出，通过配置输出通道 x 配置寄存器 CHX_SW_PUL 位，可以使通道发出一个系统时钟宽度的脉冲事件。

当输出或输出通道所选择事件输入源和软件事件控制同时发生作用时，通道输出信号是由两者异或之后的结果。例如，当 CPU 设置了 CHX_SW_PUL 后，期望结果是通道会输出一个高电平的脉冲，此时当所选择的事件输入源也输入一个持续的高电平事件时，通道输出实际变为低电平脉冲。

16.4.6 EBUS 中断号

EBUS 的 16 个 IO 检测通道可以独立的产生 16 个检测中断。

表 185 EBUS 中断号

IO 检测通道	中断号
0	5
1	6
2	7
3	8
4	11
5	20
6	21
7	22
11:8	24
15:12	25

16.5 寄存器描述

EBUS 寄存器的基地址为 0x4000_2000，下表为 EBUS 的相关寄存器描述。

表 186 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x00	EBUS IO 检测通道设置寄存器 0	32	r/w	0x05070100
0x04	EBUS IO 检测通道设置寄存器 1	32	r/w	0x00000000
0x08	EBUS IO 检测通道设置寄存器 2	32	r/w	0x00000000
0x0C	EBUS IO 检测通道设置寄存器 3	32	r/w	0x00000000
0x10	EBUS IO 检测通道设置寄存器 4	32	r/w	0x00000000
0x14	EBUS IO 检测通道设置寄存器 5	32	r/w	0x00000000
0x18	EBUS IO 检测通道设置寄存器 6	32	r/w	0x00000000
0x1C	EBUS IO 检测通道设置寄存器 7	32	r/w	0x00000000
0x20	EBUS IO 检测通道设置寄存器 8	32	r/w	0x00000000
0x24	EBUS IO 检测通道设置寄存器 9	32	r/w	0x00000000
0x28	EBUS IO 检测通道设置寄存器 10	32	r/w	0x00000000
0x2C	EBUS IO 检测通道设置寄存器 11	32	r/w	0x00000000
0x30	EBUS IO 检测通道设置寄存器 12	32	r/w	0x00000000
0x34	EBUS IO 检测通道设置寄存器 13	32	r/w	0x00000000
0x38	EBUS IO 检测通道设置寄存器 14	32	r/w	0x00000000
0x3C	EBUS IO 检测通道设置寄存器 15	32	r/w	0x00000000
0x40	EBUS IO 检测中断状态寄存器	32	r/w	0x00000000
0x44	EBUS 输出通道 0 设置寄存器	32	r/w	0x00000000
0x48	EBUS 输出通道 1 设置寄存器	32	r/w	0x00000000
0x4C	EBUS 输出通道 2 设置寄存器	32	r/w	0x00000000
0x50	EBUS 输出通道 3 设置寄存器	32	r/w	0x00000000
0x54	EBUS 输出通道 4 设置寄存器	32	r/w	0x00000000
0x58	EBUS 输出通道 5 设置寄存器	32	r/w	0x00000000

16.5.1 EBUS IO 检测通道设置寄存器 0 (EBUS_IO_DET0)

偏移地址: 0x0000

表 187 EBUS IO 检测通道设置寄存器 0

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO0_SRC_SEL: IO0 源选择 00: PA0 01: PA1 10: PA2 11: PA3 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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
0	r/w	IO0_DB_EN: IO0 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.2 EBUS IO 检测通道设置寄存器 1 (EBUS_IO_DET1)

偏移地址: 0x0004

表 188 EBUS IO 检测通道设置寄存器 1

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO1_SRC_SEL: IO1 源选择 00: PA4 01: PA5 10: PA6 11: PA7 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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
0	r/w	IO1_DB_EN: IO1 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.3 EBUS IO 检测通道设置寄存器 2 (EBUS_IO_DET2)

偏移地址: 0x0008

表 189 EBUS IO 检测通道设置寄存器 2

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO2_SRC_SEL: IO2 源选择 00: PA8 01: PA9 10: PA10 11: PA11 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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
0	r/w	IO2_DB_EN: IO2 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.4 EBUS IO 检测通道设置寄存器 3 (EBUS_IO_DET3)

偏移地址: 0x000C

表 190 EBUS IO 检测通道设置寄存器 3

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO3_SRC_SEL: IO0 源选择 00: PA12 01: PA13 10: PA14 11: 保留 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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 复位值: 0x0
0	r/w	IO3_DB_EN: IO3 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.5 EBUS IO 检测通道设置寄存器 4 (EBUS_IO_DET4)

偏移地址: 0x0010

表 191 EBUS IO 检测通道设置寄存器 4

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO4_SRC_SEL: IO4 源选择 00: PB0 01: PB1 10: PB2 11: PB3 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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
0	r/w	IO4_DB_EN: IO4 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.6 EBUS IO 检测通道设置寄存器 5 (EBUS_IO_DET5)

偏移地址: 0x0014

表 192 EBUS IO 检测通道设置寄存器 5

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO5_SRC_SEL: IO5 源选择 00: PB4 01: PB5 10: PB6 11: PB7 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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

位	访问	描述
0	r/w	IO5_DB_EN: IO5 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.7 EBUS IO 检测通道设置寄存器 6 (EBUS_IO_DET6)

偏移地址: 0x0018

表 193 EBUS IO 检测通道设置寄存器 6

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO6_SRC_SEL: IO6 源选择 00: PB8 01: PB9 10: PB10 11: PB11 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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
0	r/w	IO6_DB_EN: IO6 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.8 EBUS IO 检测通道设置寄存器 7 (EBUS_IO_DET7)

偏移地址: 0x001C

表 194 EBUS IO 检测通道设置寄存器

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO7_SRC_SEL: IO7 源选择 00: PB12 01: PB15 10: PC0 11: PC1 复位值: 0x0
3	Res	Reserved 复位值: 0x0

位	访问	描述
2:1	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
0	r/w	IO7_DB_EN: IO7 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.9 EBUS IO 检测通道设置寄存器 8 (EBUS_IO_DET8)

偏移地址: 0x0020

表 195 EBUS IO 检测通道设置寄存器 8

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO8_SRC_SEL: IO0 源选择 00: PA0 01: PA8 10: PB0 11: PB8 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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
0	r/w	IO8_DB_EN: IO8 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.10 EBUS IO 检测通道设置寄存器 9 (EBUS_IO_DET9)

偏移地址: 0x0024

表 196 EBUS IO 检测通道设置寄存器 9

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO9_SRC_SEL: IO0 源选择 00: PA1 01: PA9

位	访问	描述
		10: PB1 11: PB9 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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
0	r/w	IO9_DB_EN: IO9 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.11 EBUS IO 检测通道设置寄存器 10 (EBUS_IO_DET10)

偏移地址: 0x0028

表 197 EBUS IO 检测通道设置寄存器 10

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO10_SRC_SEL: IO10 源选择 00: PA2 01: PA10 10: PB2 11: PB10 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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
0	r/w	IO10_DB_EN: IO10 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.12 EBUS IO 检测通道设置寄存器 11 (EBUS_IO_DET11)

偏移地址: 0x002C

表 198 EBUS IO 检测通道设置寄存器 11

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO11_SRC_SEL: IO11 源选择 00: PA3 01: PA11 10: PB3 11: PB11 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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
0	r/w	IO11_DB_EN: IO11 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.13 EBUS IO 检测通道设置寄存器 12 (EBUS_IO_DET12)

偏移地址: 0x0030

表 199 EBUS IO 检测通道设置寄存器 12

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO12_SRC_SEL: IO12 源选择 00: PA4 01: PA12 10: PB4 11: PB12 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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
0	r/w	IO12_DB_EN: IO12 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.14 EBUS IO 检测通道设置寄存器 13 (EBUS_IO_DET13)

偏移地址: 0x0034

表 200 EBUS IO 检测通道设置寄存器 13

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO13_SRC_SEL: IO13 源选择 00: PA5 01: PA13 10: PB5 11: PB15 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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
0	r/w	IO13_DB_EN: IO13 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.15 EBUS IO 检测通道设置寄存器 14 (EBUS_IO_DET14)

偏移地址: 0x0038

表 201 EBUS IO 检测通道设置寄存器 14

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO14_SRC_SEL: IO14 源选择 00: PA6 01: PA14 10: PB6 11: PC0 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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
0	r/w	IO14_DB_EN: IO14 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.16 EBUS IO 检测通道设置寄存器 15 (EBUS_IO_DET15)

偏移地址: 0x003C

表 202 EBUS IO 检测通道设置寄存器 15

位	访问	描述
31:6	Res	Reserved 复位值: 0x0
5:4	r/w	IO15_SRC_SEL: IO15 源选择 00: PA7 01: 保留 10: PB7 11: PC1 复位值: 0x0
3	Res	Reserved 复位值: 0x0
2:1	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
0	r/w	IO15_DB_EN: IO15 输入滤波使能 0: 关闭 1: 使能 复位值: 0x0

16.5.17 EBUS IO 检测中断状态寄存器 (EBUS_IO_STA)

偏移地址: 0x0040

表 203 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	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0
14	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0
13	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0

位	访问	描述
12	r/w	IO12_INT_STA: IO12 检测中断状态 0: 未检测到目标边沿 1: 已检测到目标边沿 注: 此位软件写 1 清 0。 复位值: 0x0
11	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0
10	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0
9	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0
8	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0
7	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0
6	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0
5	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0
4	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0

位	访问	描述
3	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0
2	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0
1	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0
0	r/w	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 注: 此位软件写 1 清 0。 复位值: 0x0

16.5.18 EBUS 输出通道 0 设置寄存器 (EBUS_OUT0_CFG)

偏移地址: 0x0044

表 204 EBUS 输出通道 0 设置寄存器

位	访问	描述
31	r/w	OUT_SW_PUL: 输出通道 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
30:3	Res	Reserved 复位值: 0x0
2:0	r/w	OUT_SRC_SEL: 输出通道 0 输入源选择 000: 选择输入源 0 001: 选择输入源 1 010: 选择输入源 2 ... 111: 选择输入源 7 复位值: 0x0

16.5.19 EBUS 输出通道 1 设置寄存器 (EBUS_OUT1_CFG)

偏移地址: 0x0048

表 205 EBUS 输出通道 1 设置寄存器

位	访问	描述
31	r/w	OUT_SW_PUL: 输出通道 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
30:3	Res	Reserved 复位值: 0x0
2:0	r/w	OUT_SRC_SEL: 输出通道 1 输入源选择 000: 选择输入源 0 001: 选择输入源 1 010: 选择输入源 2 ... 111: 选择输入源 7 复位值: 0x0

16.5.20 EBUS 输出通道 2 设置寄存器 (EBUS_OUT2_CFG)

偏移地址: 0x004C

表 206 EBUS 输出通道 2 设置寄存器

位	访问	描述
31	r/w	OUT_SW_PUL: 输出通道 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
30:3	Res	Reserved 复位值: 0x0
2:0	r/w	OUT_SRC_SEL: 输出通道 2 输入源选择 000: 选择输入源 0 001: 选择输入源 1 010: 选择输入源 2 ... 111: 选择输入源 7 复位值: 0x0

16.5.21 EBUS 输出通道 3 设置寄存器 (EBUS_OUT3_CFG)

偏移地址: 0x0050

表 207 EBUS 输出通道 3 设置寄存器

位	访问	描述
31	r/w	OUT_SW_PUL: 输出通道 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
30:3	Res	Reserved 复位值: 0x0
2:0	r/w	OUT_SRC_SEL: 输出通道 3 输入源选择 000: 选择输入源 0 001: 选择输入源 1 010: 选择输入源 2 ... 111: 选择输入源 7 复位值: 0x0

16.5.22 EBUS 输出通道 4 设置寄存器 (EBUS_OUT4_CFG)

偏移地址: 0x0054

表 208 EBUS 输出通道 4 设置寄存器

位	访问	描述
31	r/w	OUT_SW_PUL: 输出通道 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
30:3	Res	Reserved 复位值: 0x0
2:0	r/w	OUT_SRC_SEL: 输出通道 4 输入源选择 000: 选择输入源 0 001: 选择输入源 1 010: 选择输入源 2 ... 111: 选择输入源 7 复位值: 0x0

16.5.23 EBUS 输出通道 5 设置寄存器 (EBUS_OUT5_CFG)

偏移地址: 0x0058

表 209 EBUS 输出通道 5 设置寄存器

位	访问	描述
31	r/w	OUT_SW_PUL: 输出通道 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
30:3	Res	Reserved 复位值: 0x0
2:0	r/w	OUT_SRC_SEL: 输出通道 5 输入源选择 000: 选择输入源 0 001: 选择输入源 1 010: 选择输入源 2 ... 111: 选择输入源 7 复位值: 0x0

17 键入中断 (IOW)

17.1 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。

17.2 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

17.3 模块框图

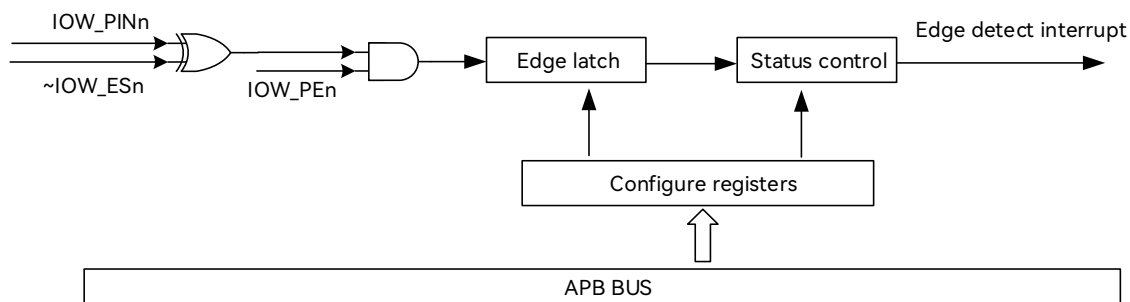


图 115 IOW 模块架构图

17.4 功能描述

17.4.1 通道输入源

表 210 键入中断通道输入源

通道	输入源
PIN0_IN0	PA0
PIN1_IN0	PA1
PIN2_IN0	PA5
PIN3_IN0	PA6
PIN4_IN0	PA7
PIN5_IN0	PB3
PIN6_IN0	PB9

通道	输入源
PIN7_IN0	Reserved
PIN0_IN1	Reserved
PIN1_IN1	Reserved
PIN2_IN1	Reserved
PIN3_IN1	Reserved
PIN4_IN1	Reserved
PIN5_IN1	Reserved
PIN6_IN1	Reserved
PIN7_IN0	Reserved

17.4.2 IOW 中断号

键入中断的中断状态寄存器中的状态都可独立使能是否产生中断，当中断产生后，软件可通过状态清除寄存器将中断状态清除。

表 211 IOW 中断号

名称	中断号
IOW_INT	23

17.5 寄存器描述

键入中断寄存器的基地址为 0x4000_1C00，下表为键入中断的相关寄存器描述。

表 212 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
0x10	键入中断输入选择寄存器	32	r/w	0x00000000

17.5.1 键入中断中断使能寄存器 (IOW_INTEN)

偏移地址: 0x0000

表 213 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

17.5.2 键入中断控制寄存器 (IOW_CTRL)

偏移地址: 0x0004

表 214 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
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

位	访问	描述
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

17.5.3 键入中断边沿选择寄存器 (IOW_EDGES)

偏移地址: 0x0008

表 215 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
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

17.5.4 键入中断状态寄存器 (IOW_STA)

偏移地址: 0x000C

表 216 键入中断状态寄存器

位	访问	描述
31:3	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 0: 未检测到目标沿 1: 检测到目标沿 注: 该位软件写 1 清 0。 复位值: 0x0
6	r/w	IOW_INT_STA: 键入中断通道 6 检测状态寄存器 0: 未检测到目标沿 1: 检测到目标沿 注: 该位软件写 1 清 0。 复位值: 0x0
5	r/w	IOW_INT_STA: 键入中断通道 5 检测状态寄存器 0: 未检测到目标沿 1: 检测到目标沿 注: 该位软件写 1 清 0。 复位值: 0x0
4	r/w	IOW_INT_STA: 键入中断通道 4 检测状态寄存器 0: 未检测到目标沿 1: 检测到目标沿 注: 该位软件写 1 清 0。 复位值: 0x0
3	r/w	IOW_INT_STA: 键入中断通道 3 检测状态寄存器 0: 未检测到目标沿 1: 检测到目标沿 注: 该位软件写 1 清 0。 复位值: 0x0
2	r/w	IOW_INT_STA: 键入中断通道 2 检测状态寄存器 0: 未检测到目标沿 1: 检测到目标沿 注: 该位软件写 1 清 0。 复位值: 0x0
1	r/w	IOW_INT_STA: 键入中断通道 1 检测状态寄存器 0: 未检测到目标沿 1: 检测到目标沿 注: 该位软件写 1 清 0。 复位值: 0x0
0	r/w	IOW_INT_STA: 键入中断通道 0 检测状态寄存器 0: 未检测到目标沿 1: 检测到目标沿 注: 该位软件写 1 清 0。 复位值: 0x0

17.5.5 键入中断输入选择寄存器 (IOW_INS)

偏移地址: 0x0010

表 217 键入中断输入选择寄存器

位	访问	描述
31:29	Res	Reserved 复位值: 0x0
28	r/w	IOW_PIN7_SEL: 键入中断通道 7 输入选择 0: IOW_PIN7_IN0 1: IOW_PIN7_IN1 复位值: 0x0
27:25	Res	Reserved 复位值: 0x0
24	r/w	IOW_PIN6_SEL: 键入中断通道 6 输入选择 0: IOW_PIN6_IN0 1: IOW_PIN6_IN1 复位值: 0x0
23:21	Res	Reserved 复位值: 0x0
20	r/w	IOW_PIN5_SEL: 键入中断通道 5 输入选择 0: IOW_PIN5_IN0 1: IOW_PIN5_IN1 复位值: 0x0
19:17	Res	Reserved 复位值: 0x0
16	r/w	IOW_PIN4_SEL: 键入中断通道 4 输入选择 0: IOW_PIN4_IN0 1: IOW_PIN4_IN1 复位值: 0x0
15:13	Res	Reserved 复位值: 0x0
12	r/w	IOW_PIN3_SEL: 键入中断通道 3 输入选择 0: IOW_PIN3_IN0 1: IOW_PIN3_IN1 复位值: 0
11:9	Res	Reserved 复位值: 0x0
8	r/w	IOW_PIN2_SEL: 键入中断通道 2 输入选择 0: IOW_PIN2_IN0 1: IOW_PIN2_IN1 复位值: 0x0

位	访问	描述
7:5	Res	Reserved 复位值: 0x0
4	r/w	IOW_PIN1_SEL: 键入中断通道 1 输入选择 0: IOW_PIN1_IN0 1: IOW_PIN1_IN1 复位值: 0x0
3:1	Res	Reserved 复位值: 0x0
0	r/w	IOW_PIN0_SEL: 键入中断通道 0 输入选择 0: IOW_PIN0_IN0 1: IOW_PIN0_IN1 复位值: 0x0

18 低电压检测 (LVD)

18.1 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。LVD 检测到低电压后发生低电压检测复位 (LVD_RSTJ)，该功能默认开启。

18.2 主要功能

- 低电压检测模块

- 检测 VCC 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

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 VCC 电压等级低于检测电压时可产生芯片复位信号

18.3 模块框图

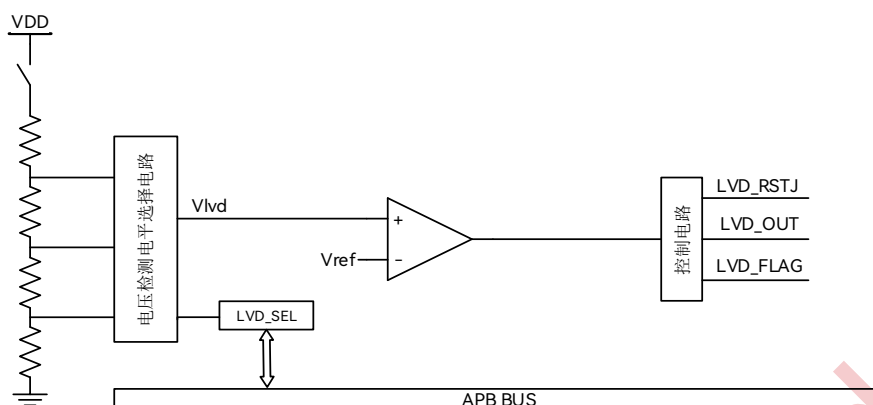


图 116 LVD 电路示意图

18.4 功能描述

18.4.1 LVD 档位

LVD 低电压检测档位可选择，详见下表。

表 218 低电压检测档位

	LVD_SEL	LVD Level(V)
1	1111	4.5+0.05V/-0.05V
2	1110	4.4+0.05V/-0.05V
3	1101	4.2+0.05V/-0.05V
4	1100	4+0.05V/-0.05V
5	1011	3.8+0.05V/-0.05V
6	1010	3.6+0.05V/-0.05V
7	1001	3.4+0.05V/-0.05V
8	1000	3.2+0.05V/-0.05V
9	0111	3+0.05V/-0.05V
10	0110	2.8+0.05V/-0.05V
11	0101	2.6+0.05V/-0.05V
12	0100	2.4+0.05V/-0.05V
13	0011-0000	保留

注：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。LVD 系统默认打开，默认档位为 2.4V。

18.4.2 LVD 功能

LVD 用于检测 VCC 电压，当 VCC 电压低于预设检测电压时，会产生相应的复位动作，LVD 用于低电压复位；当 LVD 检测到低电压并发生相应动作后，可通过相应低电压检测标志位查看；具体工作波形可详见下图。

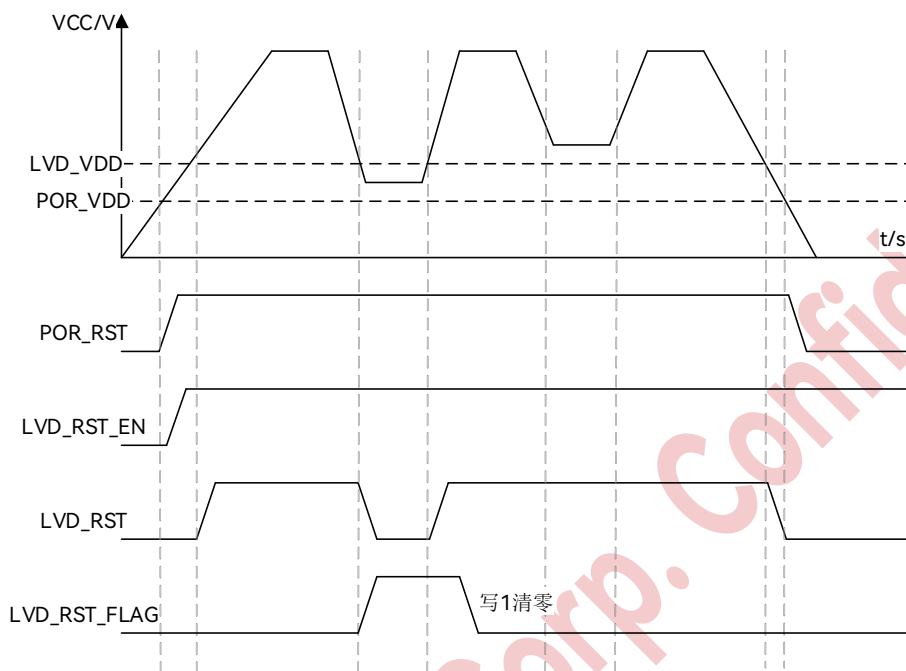


图 117 LVD 工作波形图

18.5 寄存器描述

LVD 基地址: 0x4000_4000

表 219 寄存器列表

偏移地址	寄存器列表	位宽	访问	复位值
0x0200	电压检测模块控制寄存器	32	r/w	0x00000411
0x0204	电压检测模块状态寄存器	32	r/w	0x00000000

18.5.1 电压检测模块控制寄存器 (LVD_CTRL)

偏移地址: 0x0200

表 220 电压检测模块控制寄存器

位	访问	描述
31:12	Res	Reserved 复位值: 0x0
11:8	r/w	LVD_SEL, 电压检测模块档位选择控制 0000: 保留 0001: 保留 0010: 保留 0011: 保留 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 注: 该位只能被 POR 复位或外部复位 复位值: 0x4
7:5	Res	Reserved 复位值: 0x0
4	r/w	LVD_RST_EN: 电压检测模块复位芯片使能控制 0: 关闭 1: 开启 注: 该位只能被 POR 复位或外部复位 复位值: 0x1
3:1	Res	Reserved 复位值: 0x0
0	r/w	LVD_EN: 电压检测模块使能控制 0: 关闭 1: 开启 注: 该位只能被 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。

18.5.2 电压检测模块状态寄存器 (LVD_ST)

偏移地址: 0x0204

表 221 电压检测模块状态寄存器

位	访问	描述
31:2	Res	Reserved 复位值: 0x0
1	r/w	LVD_FLG, 电压检测模块标记 0: VCC ≥ LVD_SEL 1: VCC < LVD_SEL 注: 当 LVD_OUT = 1 时, 该位硬件置位, 支持写 1 清零 复位值: 0x0
0	ro	LVD_OUT, 电压检测模块输出 0: VCC ≥ LVD_SEL 1: VCC < LVD_SEL 复位值: 0x0

19 嵌套中断向量控制器 (NVIC)

Cortex-M0+ 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 32 个可屏蔽中断通道和 4 个可编程优先级，更多信息请参考 Cortex-M0+ 技术参考手册。

本产品的中断向量表如下表所示。

表 222 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
-	-	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	保留	0x44	-
2	可配置	保留	0x48	-
3	可配置	Flash	0x4C	Flash 中断
4	可配置	保留	0x50	-
5	可配置	EBUS_INT[0]	0x54	EBUS 中断 0
6	可配置	EBUS_INT[1]	0x58	EBUS 中断 1
7	可配置	EBUS_INT[2]	0x5C	EBUS 中断 2
8	可配置	EBUS_INT[3]	0x60	EBUS 中断 3
9	可配置	ADC	0x64	ADC 中断
10	可配置	保留	0x68	-
11	可配置	EBUS_INT[4]	0x6C	EBUS 中断 4
12	可配置	SCI0_INT	0x70	SCI0 中断

中断号	优先级	名称	向量地址	描述
13	可配置	SCI1_INT	0x74	SCI1 中断
14	可配置	SCI2_INT	0x78	SCI2 中断
15	可配置	保留	0x7C	-
16	可配置	保留	0x80	-
17	可配置	保留	0x84	-
18	可配置	保留	0x88	-
19	可配置	TIM0_INT	0x8C	TIM0 中断
20	可配置	EBUS_INT[5]	0x90	EBUS 中断 5
21	可配置	EBUS_INT[6]	0x94	EBUS 中断 6
22	可配置	EBUS_INT[7]	0x98	EBUS 中断 7
23	可配置	IOW	0x9C	IOW 中断
24	可配置	EBUS_INT[11:8]	0xA0	EBUS 中断 11-8
25	可配置	EBUS_INT[15:12]	0xA4	EBUS 中断 15-12
26	可配置	保留	0xA8	-
27	可配置	CMP	0xAC	CMP 中断
28	可配置	保留	0xB4	-
29	可配置	TIM8	0xB4	TIM8 中断
30	可配置	TIM2	0xB8	TIM2 中断
31	可配置	TIM1	0xBC	TIM1 中断

20 附录

20.1 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