



# ZH5210 用户手册

Rev 1.21



<http://www.enpowermicro.com>

# 产品特征

- 电机控制专用 32 位 MCU，单周期乘法
- 工作电压范围: 2.8V~5.5V
- MCU 最高工作主频 96Mhz
- 32K 字节 FLASH 存储器
- 4K 字节 RAM 存储器
- 集成 CORDIC 运算协处理和除法器
- 集成 EPWM(三路互补带死区、故障保护、PWM 移相)
- 集成 PWM 频率和占空比检测模块
- 集成正交编码检测 QEP 模块
- 集成 4 个 16bit 定时器和 1 个 32bit 定时器
- 1 个独立看门狗定时器
- 1 路主从模式 SPI
- 1 路主从模式 I2C
- 2 路 UART，支持 LIN 同步帧检测
- 32 个高速 GPIO，支持位操作
- 集成 3 路可编程增益运放
- 集成 3 路反电动势比较器
- 集成 1 路保护检测比较器
- 双通道同步、异步采样 ADC，同时采样两路电流
- 10 位数模转换器 DAC
- 集成高精度低温漂的高频 RC 时钟
- 集成 PLL，最高频率 96Mhz
- 集成温度传感器
- 集成低功耗 LDO 和电源监测电路
- ZH5212 集成 PN 预驱,工作电压最高可到 24V

## 目录

1. 功能描述 .....	13
2. 结构框图 .....	15
3. 存储器架构 .....	16
3.1. EFLASH 存储器 .....	16
3.2. 数据存储器 .....	16
3.3. APB 外设寄存器 .....	16
4. 32Bit 处理器 .....	17
4.1. 概述 .....	17
4.2. 功能特性 .....	17
4.3. 内核控制模块(SCB) .....	18
4.3.1. 模块寄存器总表 .....	18
4.3.2. 中断控制和状态寄存器(ICSR: 0xE000ED04) .....	18
4.3.3. 应用中断控制和复位寄存器(AIRCR: 0xE000ED0C) .....	20
4.3.4. 系统控制寄存器(SCR: 0xE000ED10) .....	20
4.3.5. 配置和控制寄存器(CCR: 0xE000ED14) .....	21
4.3.6. 系统中断处理优先级寄存器(SHPR2: 0xE000ED1C) .....	21
4.3.7. 系统中断处理优先级寄存器(SHPR3: 0xE000ED20) .....	21
4.4. 系统控制(SYSCTRL) .....	22
4.4.1. 模块寄存器总表 .....	22
4.4.2. 内核睡眠控制寄存器(DPSLEEPCTRL: 0x4001F004) .....	22
4.4.3. 内核锁定复位使能寄存器(LOCKUPRESET: 0x4001F008) .....	23
4.4.4. 内核复位状态标志位(RESETIFO: 0x4001F010) .....	23
4.5. 系统定时器(SysTick) .....	23
4.5.1. 模块寄存器总表 .....	23
4.5.2. SysTick 控制和状态寄存器(CTRL: 0xE000E010) .....	23
4.5.3. SysTick 重加载寄存器(LOAD: 0xE000E014) .....	24
4.5.4. SysTick 当前值寄存器(VAL: 0xE000E018) .....	24
4.5.5. SysTick 校准值寄存器(CALIB: 0xE000E01C) .....	24
5. 系统控制(SYSCFG) .....	25
5.1. 模块寄存器总表 .....	25
5.1.1. 系统保护寄存器(SYSKEY: 0x4000E018) .....	25
5.2. 系统时钟(SYSCLK) .....	26
5.2.1. 系统时钟框图 .....	26
5.2.2. 系统时钟寄存器 .....	27
5.2.3. 锁相环倍频器(PLL) .....	33
5.2.4. 系统工作时钟 .....	33
5.2.5. 时钟切换 .....	33
5.3. 电源管理(PMU) .....	35
5.3.1. 电源管理寄存器 .....	35

5.3.2.	睡眠工作模式 (SLEEP 工作模式)	38
5.3.3.	深度睡眠工作模式 (DPSLEEP 工作模式)	38
5.3.4.	停机工作模式 (STOP 工作模式)	39
5.4.	内部参考电压	40
5.4.1.	内部参考电压框图	40
5.4.2.	内部基准电平校准控制位寄存器 (VBGTRIM: 0x4000E00C)	40
5.4.3.	模拟功能控制寄存器 (ANA_CON: 0x4000E048)	41
5.5.	温度传感器 (VTS)	41
5.5.1.	温度传感器框图	41
5.5.2.	模拟功能控制寄存器 (ANA_CON: 0x4000E048)	42
5.6.	复位系统 (RESET)	43
5.6.1.	系统状态寄存器 (SYSSTA: 0x4000E024)	43
5.6.2.	功能描述	44
6.	中断控制器 (NVIC)	45
6.1.	中断和异常向量表	46
6.2.	模块寄存器总表	47
6.3.	NVIC 控制寄存器	48
6.3.1.	中断允许寄存器 (ISER)	48
6.3.2.	中断禁止寄存器 (ICER)	48
6.3.3.	中断挂起寄存器 (ISPR)	49
6.3.4.	清除中断挂起寄存器 (ICPR)	49
6.3.5.	中断优先级寄存器	49
6.4.	外部引脚中断	51
6.4.1.	PA 中断使能寄存器 (PAIEN/ PAIENCLR: 0x40010020 /0x40010024)	51
6.4.2.	PA 中断类型寄存器 (PAITYP/ PAITYPCLR: 0x40010028 /0x4001002C)	51
6.4.3.	PA 中断极性控制寄存器 (PAIPOL/ PAIPOLCLR: 0x40010030/0x40010034)	51
6.4.4.	PA 中断状态寄存器 (PAISTA: 0x40010038)	52
6.4.5.	PB 中断使能寄存器 (PBIEN/ PBIENCLR: 0x40011020/0x40011024)	52
6.4.6.	PB 中断类型寄存器 (PBITYP/ PBITYPCLR: 0x40011028/0x4001102C)	52
6.4.7.	PB 中断极性控制寄存器 (PBIPOL/ PBIPOLCLR: 0x40011030/0x40011034)	53
6.4.8.	PB 中断状态寄存器 (PBISTA: 0x40011038)	53
6.4.9.	外部引脚可唤醒中断控制寄存器 (EINTCON: 0x4000E01C)	53
6.4.10.	外部引脚可唤醒中断标志位寄存器 (EINTFCN: 0x4000E020)	56
7.	通用输入输出端口 (GPIO)	57
7.1.	端口结构	58
7.2.	模块寄存器总表	58
7.3.	PA 端口寄存器	60
7.3.1.	PA 端口输入寄存器 (PAIN: 0x40010000)	60
7.3.2.	PA 端口输出寄存器 (PAOUT: 0x40010004)	61
7.3.3.	PA 端口单独输出寄存器	61
7.3.4.	PA 端口模式寄存器 (PAMD: 0x40010008)	64

7.3.5. PA 上下拉电阻使能寄存器(PAPDREN: 0x4001000C).....	65
7.3.6. PA 输入输出控制寄存器(PAOEN / PAOENCLR: 0x40010010/0x40010014).....	65
7.3.7. PA 复用控制寄存器(PAALT/ PAALTCLR: 0x40010018/0x4001001C).....	65
7.3.8. PA 复用功能选择寄存器 0(PAFMUX70: 0x4000E030).....	66
7.3.9. PA 复用功能选择寄存器 1(PAFMUXF8: 0x4000E034).....	67
7.3.10. PA 上拉下拉选择寄存器(PAPDRS: 0x4000E050).....	67
7.3.11. PA 开漏开源输出选择寄存器(PAODOSS: 0x4000E054).....	68
7.3.12. PA 输出压摆率输入迟滞配置寄存器(PACSSR: 0x4000E058).....	68
7.3.13. PA 输出驱动力选择寄存器(PADRS: 0x4000E05C).....	68
7.3.14. PA 模拟通道 1 配置寄存器(PAAEN1: 0x4000E060).....	68
7.3.15. PA 模拟通道 2 配置寄存器(PAAEN2: 0x4000E064).....	69
7.3.16. PA 上拉电阻选择寄存器(PAPUS: 0x4000E068).....	69
7.4. PB 端口寄存器.....	69
7.4.1. PB 端口输入寄存器(PBIN: 0x40011000).....	69
7.4.2. PB 端口输出寄存器(PBOUT: 0x40011004).....	69
7.4.3. PB 端口单独输出寄存器.....	70
7.4.4. PB 端口模式寄存器 (PBMD: 0x40011008).....	73
7.4.5. PB 上下拉电阻使能寄存器(PBPDREN: 0x4001100C).....	73
7.4.6. PB 输入输出控制寄存器(PBOEN/ PBOENCLR: 0x40011010/0x40011014).....	73
7.4.7. PB 复用控制寄存器(PBALT/ PBALTCLR: 0x40011018/0x4001101C).....	74
7.4.8. PB 复用功能选择寄存器 0(PBFMUX70: 0x4000E038).....	74
7.4.9. PB 复用功能选择寄存器 1(PBFMUXF8: 0x4000E03C).....	75
7.4.10. PB 上拉下拉选择寄存器(PBPDRS: 0x4000E070).....	76
7.4.11. PB 开漏开源配置寄存器(PBODOSS: 0x4000E074).....	76
7.4.12. PB 输出压摆率输入迟滞配置寄存器(PBCSSR: 0x4000E078).....	76
7.4.13. PB 输出驱动力配置寄存器(PBDRS: 0x4000E07C).....	77
7.4.14. PB 模拟通道 1 配置寄存器(PBAEN1: 0x4000E080).....	77
7.4.15. PB 模拟通道 2 配置寄存器(PBAEN2: 0x4000E084).....	77
7.4.16. PB 上拉电阻选择寄存器(PBPUS: 0x4000E088).....	77
7.5. IO 功能映射寄存器.....	78
7.5.1. IO 映射寄存器 0(REMAP0: 0x4000E028).....	78
7.5.2. IO 映射寄存器 1(REMAP1: 0x4000E02C).....	79
7.6. 功能描述.....	80
7.6.1. IO 模式设置 (PxALT= 0).....	80
7.6.2. IO 数字功能复用.....	81
7.6.3. IO 模拟功能复用.....	85
8. 定时器(BASICTIMER).....	87
8.1. 模块寄存器总表.....	87
8.2. 寄存器说明.....	87
8.2.1. BasicTimer 控制寄存器 (BTCTRL: 0x40000000).....	87
8.2.2. BasicTimer 计数寄存器 (BTCURR: 0x40000004).....	88

8.2.3. BasicTimer 重载寄存器 (BTRELOAD: 0x40000008)	88
8.2.4. BasicTimer 中断清零寄存器 (BTINTCLR: 0x4000000C)	88
9. 定时器(TIMER0/1)	88
9.1. 模块寄存器总表	89
9.2. 寄存器说明	89
9.2.1. 定时器 T0 控制寄存器(T0CON: 0x40000800)	89
9.2.2. 定时器 T0 周期寄存器 (T0PER: 0x40000804)	91
9.2.3. 定时器 T0 计数器 (T0CNT: 0x40000808)	91
9.2.4. 定时器 T0 捕获比较寄存器 (T0CCR: 0x4000080C)	91
9.2.5. 定时器 T1 控制寄存器(T1CON: 0x40001000)	92
9.2.6. 定时器 T1 周期寄存器 (T1PER: 0x40001004)	93
9.2.7. 定时器 T1 计数寄存器 (T1CNT: 0x40001008)	94
9.2.8. 定时器 T1 捕获比较寄存器 (T1CCR: 0x4000100C)	94
9.3. 功能描述	94
9.3.1. 定时模式	94
9.3.2. 捕获模式	97
9.3.3. PWM 模式	99
9.3.4. 事件更新	101
9.3.5. 中断源	101
10. 定时器(TIMER2)	102
10.1. 模块寄存器总表	102
10.2. 寄存器说明	102
10.2.1. 定时器 T2 控制寄存器(T2CON: 0x40001800)	102
10.2.2. 定时器 T2 配置寄存器(T2CFG: 0x40001804)	104
10.2.3. 定时器 T2 中断寄存器(T2ICON: 0x40001808)	106
10.2.4. 定时器 T2 周期寄存器 (T2PER: 0x4000180C)	106
10.2.5. 定时器 T2 计数寄存器 (T2CNT: 0x40001810)	106
10.2.6. 定时器 T2 捕获/比较寄存器 (T2CCR0: 0x40001814)	107
10.2.7. 定时器 T2 捕获/比较寄存器 (T2CCR1: 0x40001818)	107
10.2.8. 定时器 T2 捕获/比较寄存器 (T2CCR2: 0x4000181C)	107
10.3. 功能描述	108
10.3.1. 定时模式	108
10.3.2. PWM 模式	109
10.3.3. 捕获模式	111
10.3.4. 事件更新	112
10.3.5. 捕获滤波	112
10.3.6. 中断源	113
11. 定时器(RT)	113
11.1. 系统控制寄存器(SYSCON:0x4000E000)	113
11.2. 系统状态寄存器(SYSSTA:0x4000E024)	114
12. 看门狗(WDOG)	115

12.1. 模块寄存器总表 .....	115
12.2. 寄存器说明 .....	115
12.2.1. WDOG 计数值载入寄存器 (WDOGLOAD: 0x4000C000) .....	115
12.2.2. WDOG 当前计数值寄存器 (WDOGVALUE: 0x4000C004) .....	116
12.2.3. WDOG 控制寄存器 (WDOGCONTROL: 0x4000C008) .....	116
12.2.4. WDOG 中断清除寄存器 (WDOGINTCLR: 0x4000C00C) .....	116
12.2.5. WDOG 中断标志寄存器 (WDOGMIS: 0x4000C014) .....	116
12.2.6. WDOG 写使能寄存器 (WDOGLOCK: 0x4000CC00) .....	116
12.3. 看门狗应用 .....	117
12.3.1. 示例代码 .....	117
12.3.2. 看门狗载入值对应复位时间 .....	117
13. 编码器检测器(EQEP) .....	117
13.1. 模块寄存器总表 .....	117
13.2. 寄存器说明 .....	118
13.2.1. 控制寄存器(QEPCTL: 0x40002800) .....	118
13.2.2. 初始值寄存器(QPOSINIT: 0x40002804) .....	120
13.2.3. 最大值寄存器(QPOSMAX: 0x40002808) .....	120
13.2.4. 比较值寄存器(QPOSCMP: 0x4000280C) .....	120
13.2.5. 锁存寄存器(QPOSLAT: 0x40002810) .....	120
13.2.6. 锁存寄存器(QPOSILAT: 0x40002814) .....	120
13.2.7. 锁存寄存器(QPOSSLAT: 0x40002818) .....	121
13.2.8. 计数寄存器(QPOSCNT: 0x4000281C) .....	121
13.2.9. 状态寄存器(QEPSTS: 0x40002820) .....	121
13.2.10. 中断控制寄存器(QEPICTL: 0x40002824) .....	122
13.3. 功能说明 .....	122
13.3.1. 正交解码单元(QDU) .....	123
13.3.2. 位置计数和控制单元(PCCU) .....	125
14. 脉冲宽度调制器(EPWM) .....	132
14.1. 模块寄存器总表 .....	132
14.2. 寄存器说明 .....	133
14.2.1. 控制寄存器(PWMCON: 0x40003800) .....	133
14.2.2. 周期控制寄存器(PWMPER: 0x40003804) .....	135
14.2.3. 死区控制寄存器(DTCON: 0x40003808) .....	135
14.2.4. 输出控制寄存器(OUTCFG: 0x4000380C) .....	135
14.2.5. 捕获比较配置寄存器(CICOCFG: 0x40003810) .....	136
14.2.6. 中断控制寄存器(INTCON: 0x40003814) .....	138
14.2.7. 故障控制寄存器(FTCFG: 0x40003818) .....	141
14.2.8. 通道 0 数据寄存器 (CCR0N: 0x4000381C) .....	142

14.2.9.通道 0 数据寄存器 (CCR0P: 0x40003820).....	142
14.2.10. 通道 1 数据寄存器 (CCR1N: 0x40003824) .....	142
14.2.11. 通道 1 数据寄存器 (CCR1P: 0x40003828).....	143
14.2.12. 通道 2 数据寄存器 (CCR2N: 0x4000382C).....	143
14.2.13. 通道 2 数据寄存器 (CCR2P: 0x40003830).....	143
14.2.14. 通道 3 数据寄存器 (CCR3N: 0x40003834) .....	143
14.2.15. 通道 3 数据寄存器 (CCR3P: 0x40003838).....	143
14.2.16. PWMTRGC 控制寄存器 (TRGC: 0x4000383C) .....	143
14.2.17. PWM 计数寄存器(PWMCNT: 0x40003840) .....	144
14.2.18. PWM 输出极性配置锁定寄存器(PWMLOCK: 0x40003844) .....	144
14.2.19. PWM 单电阻采样状态寄存器(PWMSSTA: 0x40003848) .....	145
14.3. 功能描述.....	145
14.3.1.计数器工作方式 .....	146
14.3.2.捕获模式 .....	149
14.3.3.PWM 模式.....	152
14.3.4.PWM 输出控制 .....	155
14.3.5.中断 .....	160
15. 频率检测器(TF).....	161
15.1. 设计框图.....	161
15.2. 模块寄存器总表 .....	161
15.3. 寄存器说明 .....	162
15.3.1.频率捕获控制寄存器(TFCON: 0x40009800).....	162
15.3.2.频率捕获数据(TFDAT: 0x40009804) .....	162
15.4. 功能描述.....	162
16. PWM 占空比检测器(TD) .....	163
16.1. 功能框图.....	164
16.2. 模块寄存器总表 .....	164
16.3. 寄存器说明 .....	164
16.3.1.PWM 占空比捕获控制寄存器 (TDCON: 0x40004800).....	164
16.3.2.PWM 占空比捕获数据 (TDDAT: 0x40004804).....	165
16.4. 功能描述.....	165
16.4.1.PWM 占空比捕获功能.....	165
17. 数学运算协处理器(MATH) .....	166
17.1. 模块寄存器总表 .....	167
17.2. 寄存器说明.....	168
17.2.1.控制寄存器 (DIVCON: 0x40015000) .....	168
17.2.2.除法状态寄存器 (DIVSTA : 0x40015004) .....	169
17.2.3.除法状态清零寄存器 (DIVSTA_CLR : 0x40015018) .....	170
17.2.4.被除数寄存器 (DIVDVD: 0x40015008) .....	170
17.2.5.除数寄存器 (DIVDVS: 0x4001500C) .....	170
17.2.6.商寄存器 (DIVQUOT : 0x40015010) .....	170

17.2.7.余数寄存器(DIVRMD : 0x40015014) .....	170
17.2.8.CORDIC 控制寄存器(CORDCON : 0x40015020) .....	171
17.2.9.CORDIC 状态寄存器(CORDSTA : 0x40015024) .....	172
17.2.10. ORDIC 状态清零寄存器(CORDSTA_CLR : 0x40015040).....	173
17.2.11. XI 寄存器(CORDXI : 0x40015028) .....	173
17.2.12. YI 寄存器(CORDYI : 0x4001502C).....	173
17.2.13. ZI 寄存器(CORDZI: 0x40015030).....	173
17.2.14. XO 寄存器(CORDXO: 0x40015034).....	174
17.2.15. YO 寄存器(CORDYO: 0x40015038).....	174
17.2.16. ZO 寄存器(CORDZO: 0x4001503C).....	174
17.3.除法.....	174
17.4.CORDIC.....	174
17.4.1.结构框图.....	176
17.4.2.工作模式和对应结果 .....	177
17.4.3.X、Y、Z 数据格式 .....	178
17.4.4.收敛域.....	179
17.4.5.反正切和双曲反正切查找表.....	180
17.4.6.线性函数的模拟查找表.....	181
17.5.结果链接.....	181
18.基础异步通信收发器(BUART).....	182
18.1.功能框图 .....	183
18.2.模块寄存器总表 .....	183
18.3.寄存器说明 .....	184
18.3.1.发送/接收数据寄存器(BUARTBUF: 0x40006000).....	184
18.3.2.状态寄存器 (BUARTSTA : 0x40006004) .....	184
18.3.3.控制寄存器 (BUARTCON: 0x40006008).....	184
18.3.4.中断标志寄存器 (BUARTFLG : 0x4000600C).....	185
18.3.5.波特率寄存器 (BUARTBAUD: 0x40006010).....	185
19.通用异步通信收发器(UART).....	185
19.1.功能框图.....	186
19.2.模块寄存器总表.....	187
19.3.寄存器说明 .....	188
19.3.1.UART 控制寄存器(UARTCON: 0x40008800) .....	188
19.3.2.UART 缓存控制寄存器(BUFCON: 0x4000880C) .....	188
19.3.3.LIN 控制寄存器(LINCON: 0x40008810).....	189
19.3.4.LINPID 寄存器(LINPID: 0x40008814).....	190
19.3.5.LIN BAUD 寄存器(LINBAUD: 0x40008818).....	190
19.3.6.LIN BFTime 寄存器(LINBFT: 0x4000881C) .....	190
19.3.7.UART 通讯数据寄存器(UARTBUF: 0x40008804).....	190
19.3.8.UART 波特率寄存器(BAUDCFG: 0x40008808).....	190
19.3.9.UART 发送缓存寄存器 1(WBUF1: 0x40008820) .....	190

19.3.10.	UART 发送缓存寄存器 2(WBUF2: 0x40008824) .....	191
19.3.11.	UART 发送缓存寄存器 3(WBUF3: 0x40008828) .....	191
19.3.12.	UART 接收缓存寄存器 1(RBUF1: 0x40008830) .....	191
19.3.13.	UART 接收缓存寄存器 2(RBUF2: 0x40008834) .....	191
19.3.14.	UART 接收缓存寄存器 3(RBUF3: 0x40008838) .....	192
19.4.	功能描述 .....	192
19.5.	工作方式 .....	192
19.5.1.	工作方式 0—同步移位寄存器 .....	192
19.5.2.	工作方式 1—10 位数据异步通信 .....	193
19.5.3.	工作方式 2—11 位数据异步通信 .....	194
19.5.4.	多机通信 .....	195
19.5.5.	波特率设置 .....	196
19.5.6.	LIN 同步帧检测 .....	197
20.	I2C 串行通信收发器(I2C) .....	199
20.1.	模块寄存器总表 .....	199
20.2.	寄存器说明 .....	199
20.2.1.	波特率控制寄存器(I2CBAUD: 0x40008000) .....	199
20.2.2.	从机地址寄存器(I2CSADR: 0x40008004) .....	200
20.2.3.	控制寄存器(I2CCON: 0x40008008) .....	200
20.2.4.	中断屏蔽寄存器(I2CINTMASK: 0x4000800C) .....	202
20.2.5.	中断标志清除寄存器(I2CINTCLR: 0x40008010) .....	202
20.2.6.	多字节传输控制寄存器(I2CTNUM: 0x40008014) .....	203
20.2.7.	传输状态寄存器(I2CSTA: 0x40008018) .....	203
20.2.8.	发送数据寄存器(I2CSDAT3_0: 0x40008020) .....	204
20.2.9.	发送数据寄存器(I2CSDAT7_4: 0x40008024) .....	204
20.2.10.	接收数据寄存器(I2CRDAT3_0: 0x40008030) .....	204
20.2.11.	接收数据寄存器(I2CRDAT7_4: 0x40008034) .....	205
20.3.	功能说明 .....	205
20.3.1.	单字节收发设置: .....	205
20.3.2.	多字节收发设置: .....	205
20.3.3.	中断 .....	206
20.3.4.	IIC 总线时序图 .....	206
20.3.5.	寻址方式 .....	207
20.3.6.	开始 .....	207
20.3.7.	停止 .....	207
20.3.8.	重启 .....	208
20.4.	示例代码 .....	208
20.4.1.	I2C 管脚配置示例 .....	208
20.4.2.	示例程序: 主机模式, 单字节发送 .....	209
20.4.3.	示例程序: 主机模式, 多字节发送 .....	209
20.4.4.	示例程序: 主机模式, 单字节接收 .....	210

20.4.5.示例程序：主机模式，多字节接收.....	210
20.4.6.示例程序：主机模式，重启.....	211
20.4.7.示例程序：从机模式，单字节发送.....	212
20.4.8.示例程序：从机模式，多字节发送.....	213
20.4.9.示例程序：从机模式，单字节接收.....	214
20.4.10. 示例程序：从机模式，多字节接收.....	215
20.4.11. 示例程序：从机模式，重启 .....	216
21. 串行同步通信收发器(SPI).....	217
21.1.功能框图.....	218
21.2.模块寄存器总表 .....	218
21.3.寄存器说明 .....	219
21.3.1.SPI 控制寄存器(SPICON: 0x40009000) .....	219
21.3.2.SPI 状态寄存器(SPISTA: 0x40009004) .....	220
21.3.3.SPI 数据寄存器(SPIDAT: 0x40009008).....	220
21.4.功能描述.....	221
22. 模拟比较器(ACMP0/1/2/3).....	223
22.1.功能框图.....	224
22.1.1.ACMP0/1/2 框图 .....	224
22.1.2.ACMP3 框图 .....	225
22.1.3.内置分压模块框图.....	225
22.2.模块寄存器总表 .....	226
22.3.寄存器说明 .....	227
22.3.1.比较器 012 控制寄存器(CMP012CON: 0x40005810) .....	227
22.3.2.ACMP012 输出配置寄存器(CMPOCFG: 0x40005800).....	229
22.3.3.ACMP3 控制寄存器(CMP3CON: 0x40005814) .....	230
22.3.4.ACMP0/1/2/3 滤波配置寄存器 0(CMPFILT: 0x40005804).....	231
22.3.5.ACMP0/1/2/3 输出状态寄存器(CMPOSTA: 0x40005808).....	232
22.3.6.ACMP0/1/2/3 中断控制寄存器(CMPINTC: 0x4000580C) .....	232
22.3.7.模拟功能控制寄存器 (ANA_CON:0x4000E048) .....	234
22.4.功能描述.....	234
22.4.1.ACMP0/1/2/3 比较器迟滞功能 .....	234
22.4.2.ACMP0/1/2/3 的输出滤波配置 .....	236
22.4.3.比较器输出中断 .....	238
23. 运算放大器(OPA0/1/2).....	238
23.1.功能框图.....	239
23.2.模块寄存器总表 .....	240
23.3.寄存器说明 .....	240
23.3.1.OPA0 控制寄存器(OPA0CON: 0x40006800) .....	240
23.3.2.OPA1 控制寄存器(OPA1CON: 0x40006804) .....	241
23.3.3.OPA2 控制寄存器(OPA2CON: 0x40006808).....	242
23.3.4.模拟功能控制寄存器 (ANA_CON:0x4000E048) .....	243

23.4. 功能描述.....	244
23.4.1. 失调校准.....	244
24. 模数转换器(ADC).....	244
24.1. 功能框图.....	245
24.2. 模块寄存器总表.....	245
24.3. 寄存器说明.....	246
24.3.1. ADC 控制寄存器(ADCON: 0x40005000).....	246
24.3.2. ADC 自检配置寄存器(ADCHK: 0x40005004).....	248
24.3.3. ADC 延时配置寄存器(ADDLY: 0x40005008).....	248
24.3.4. ADC 通道控制寄存器(ADSEL: 0x4000500C).....	249
24.3.5. ADC 数据缓存寄存器( ADCDATA0: 0x40005010).....	250
24.3.6. ADC 数据缓存寄存器( ADCDATA1: 0x40005014).....	250
24.3.7. ADC 数据缓存寄存器( ADCDATA2: 0x40005018).....	250
24.3.8. ADC 数据缓存寄存器( ADCDATA3: 0x4000501C).....	250
24.3.9. ADC 数据缓存寄存器( ADCDATA4: 0x40005020).....	250
24.3.10. ADC 数据缓存寄存器( ADCDATA5: 0x40005024).....	250
24.3.11. ADC 数据缓存寄存器( ADCDATA6: 0x40005028).....	251
24.3.12. ADC 数据缓存寄存器( ADCDATA7: 0x4000502C).....	251
24.3.13. Iu 数据寄存器( IUDATA: 0x40005030).....	251
24.3.14. Iv 数据寄存器( IVDATA: 0x40005034).....	251
24.3.15. Iw 数据寄存器( IWDATA: 0x40005038).....	251
24.3.16. OFFSET 数据寄存器(OFFSET: 0x4000503C).....	251
24.4. 功能描述.....	252
24.4.1. 单次单通道转换模式.....	252
24.4.2. 连续单通道转换模式.....	253
24.4.3. 单次组转换模式.....	253
24.4.4. 连续组转换模式.....	254
24.4.5. ADC 启动.....	254
24.4.6. ADC 自检.....	254
24.4.7. ADC 中断.....	254
24.4.8. ADC 同步双采样.....	255
24.4.9. ADC 异步双采样.....	255
25. 数模转换器(DAC).....	257
25.1. 功能框图.....	257
25.2. 模块寄存器总表.....	257
25.3. 寄存器说明.....	258
25.3.1. DAC 数据寄存器(DACDATA: 0x40007800).....	258
25.3.2. DAC 控制寄存器(DACCON: 0x40007804).....	258
25.4. DAC 功能描述.....	258
25.4.1. DAC 模块使能.....	258
25.4.2. DAC 输出电压计算.....	258

25.4.3.DAC 输出与其他模块互联关系.....	259
26. 存储器(FLASH).....	259
26.1. 模块寄存器总表.....	259
26.2. 寄存器说明.....	260
26.2.1.系统控制寄存器 (SYSCON: 0x4000E000) .....	260
26.2.2.FLASH 编程地址寄存器 (FSADDR: 0x40012000) .....	260
26.2.3.FLASH 数据寄存器 (FSDATA: 0x40012004) .....	261
26.2.4.FLASH 编程控制寄存器 (FSCON: 0x40012008) .....	261
26.2.5.FLASH 状态寄存器 (FSSTA: 0x4001200C) .....	262
26.2.6.FLASH 主存储区安全锁寄存器(FSMINKEY: 0x40012014).....	263
26.2.7.FLASH 信息区安全锁寄存器(FSINFOKEY: 0x40012018).....	263
26.2.8.FLASH 访问控制寄存器(FSACR: 0x4001201C) .....	264
26.2.9.IAP 保护配置寄存器 (FSIAP: 0x40012020) .....	264
26.2.10. 写保护配置寄存器 (FSWRP: 0x40012024) .....	265
26.3. 读操作.....	265
26.3.1.指令预取.....	265
26.3.2.Flash 读操作等待.....	265
26.3.3.Flash 读低功耗控制.....	266
26.4. Flash 编程和擦除操作.....	266
26.4.1.安全开关打开/关闭.....	266
26.4.2.Flash 主存储区字节编程.....	267
26.4.3.Flash 主存储区页擦除.....	268
26.4.4.Flash 信息区字节编程.....	268
26.4.5.Flash 信息区页擦除.....	268
26.4.6.Flash 全片擦除.....	269
26.5. Flash 读保护.....	269
26.6. Flash 写保护.....	269
27. 用户配置.....	270
28. 调试模式.....	271
28.1. 系统控制寄存器 (SYSCON: 0x4000E000) .....	271
28.2. SWD 复用成 GPIO .....	271
29. 版本历史.....	272

## 1. 功能描述

- 工作电压: 2.8V~5.5V
- 1 个 PWM 频率检测 TF 模块
- 工作温度: -40℃~105℃

- 高速 32 位 MCU 核,最高工作频率: 96Mhz
- 存储器:  
32K 字节 EFLASH  
4K 字节 SRAM
- PLL 可倍频到 96Mhz±1%
- 片内集成晶体振荡电路, 可以外接 8Mhz-16Mhz 的晶振
- 支持在线调试(Debug), 在线编程(ISP), 在应用编程(IAP)
- 支持外部复位, 内置上电复位电路, 低压复位电路
- 支持程序加密
- 32 个高速通用 GPIO, 支持位操作
- 1 个 32 位基本定时器 BasicTimer
- 1 个 16 位低频时钟驱动的 RT 定时器
- 2 个 16 位高级定时器 Timer0、Timer1  
支持计数, 捕获, 生成 PWM
- 1 个 16 位高级定时器 Timer2  
支持计数, 3 路捕获, 生成 3 路不同占空比的 PWM
- 1 个单周期 32 位乘法器
- MATH 协处理器, 支持除法、CORDIC 运算
- 1 路正交编码检测 QEP 模块
- 1 个 16 位 PWM 发生模块  
支持输出 3 路互补 PWM, 死区设置, 硬件故障保护。
- 1 个 PWM 占空比检测 TD 模块, 占空比最高精度 0.2%, 支持检测 40hz~100khz 的 PWM
- 3 种通信接口:  
1 路基本 BUART  
1 路通用 UART, 带 9Bytes 缓冲, 支持 LIN 同步帧检测  
1 路主从模式 SPI  
1 路主从模式 I2C, 带 8Bytes 缓冲
- 1 个 12 位高精度模数转换器 ADC, 最快转换时间为 1us  
支持 8 路组转换, 最多支持 14 路外部 ADC 通道。  
支持同步/异步双通道采样两路电流。
- 3 个高性能模拟比较器 ACMP0/1/2,  
支持 3 相反电动势检测  
内置中心点生成电路
- 1 个多功能高性能独立模拟比较器 ACMP3  
输入信号可选择管脚、OPA、DAC 相关信号。
- 3 个高性能运算放大器 OPA  
支持外置放大电路、内置放大电路、差分输入
- 1 个 10 位数模转换器 DAC
- 支持多种低功耗待机模式, 最低待机电流小于 5uA

硬件支持单电阻采样 PWM 移相功能。

- 1 个 PWM 频率检测 TF 模块
- 1 个 PWM 占空比检测 TD 模块, 占空比最高精度 0.2%, 支持检测 40hz~100khz 的 PWM

■ 应用方向:

通用电机控制, 伺服系统, 风扇、水泵、压缩机, 电动工具等

浙江英能电子科技有限公司(初稿, 内容会有改动)

## 2. 结构框图

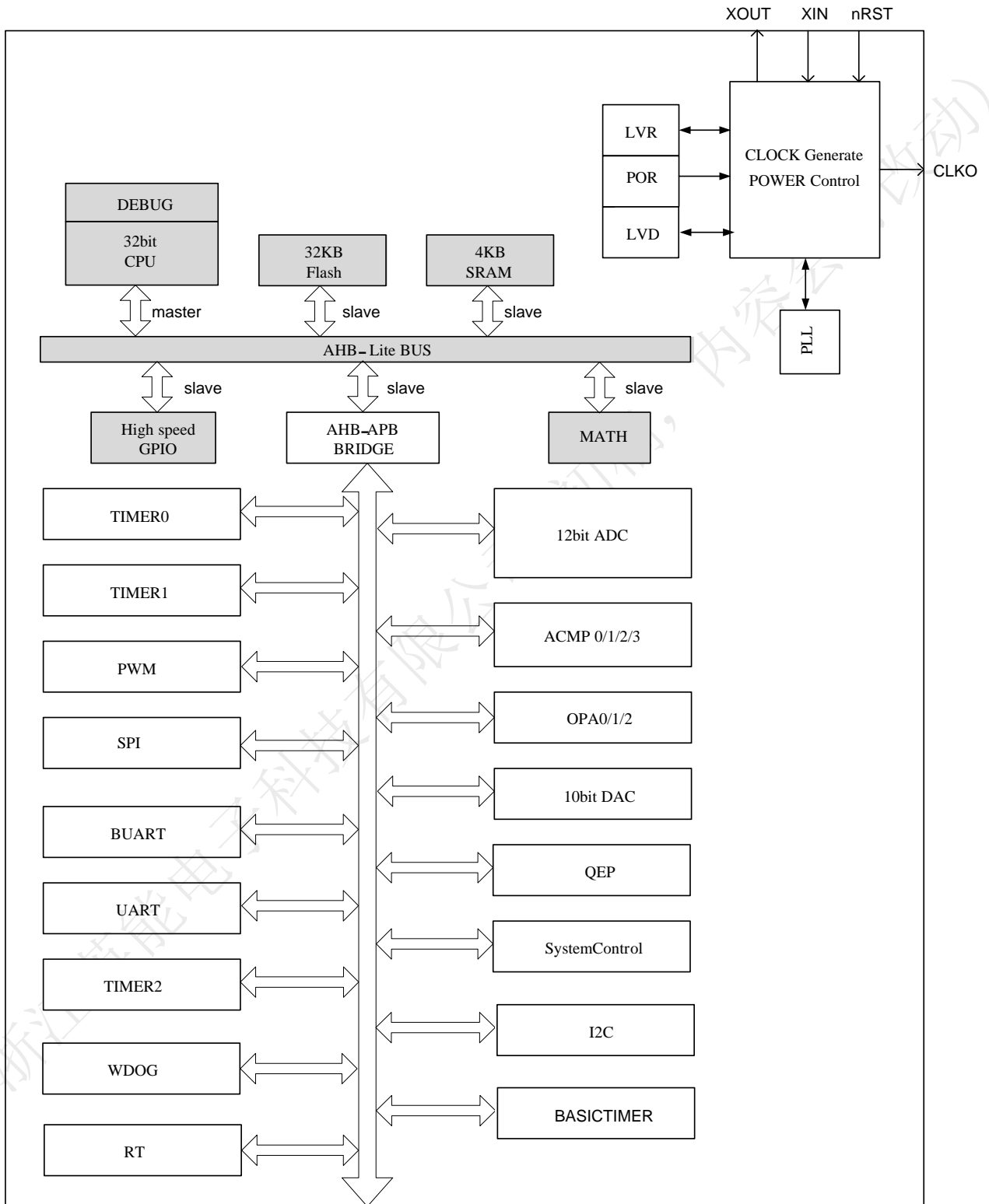


图 2 结构框图

## 3. 存储器架构

地址范围	功能
0x00000000 - 0x00007FFF	32K bytes EFLASH程序区
0x00008000 - 0x000087FF	2K bytes EFLASH信息区
0x20000000 - 0x20000FFF	4K bytes SRAM
0x40000000 - 0x4000FFFF	APB外设访问空间
0x40010000 - 0x40010FFF	AHB GPIOA寄存器访问空间
0x40011000 - 0x40011FFF	AHB GPIOB寄存器访问空间
0x40012000 - 0x40012FFF	AHB EFLASH寄存器访问空间
0x40015000 - 0x40015FFF	AHB MATH协处理器访问空间
0x4001F000 - 0x4001FFFF	AHB SYSCTRL寄存器访问空间
0xE0000000 - 0xE000FFFF	SCS寄存器访问空间
0xF0000000 - 0xF0000FFF	SYSTEM ROM TABLE

### 3.1. EFLASH 存储器

EFLASH 程序存储器空间为 32K，访问地址 0x00000000 - 0x00007FFF。复位后，程序从 0x00000000 地址开始执行程序，没有 Boot ROM。

EFLASH 信息区空间为 2K，访问地址 0x00008000-0x000087FF。其中 0x00008000-0x000085FF 用户可以修改，可用于存储数据，配置代码读保护等功能。0x00008600-0x000087FF 为芯片出厂配置区域，用户禁止修改。

### 3.2. 数据存储器

内部数据存储地址空间 0x20000000 - 0x20000FFF，集成 4K 字节 SRAM。

### 3.3. APB 外设寄存器

SFR 寄存器主要分布在 0x40000000-0x4000FFFF 范围内的 APB 总线上，只有 EFLASH 和 GPIO 控制寄存器放在 AHB 总线上。APB 上外设 SFR 地址分布如下：

地址范围	模块
0x40000000 - 0x400007FF	BASICTIMER
0x40000800 - 0x40000FFF	TIMER0
0x40001000 - 0x400017FF	TIMER1
0x40001800 - 0x40001FFF	TIMER2

0x40002800 - 0x40002FFF	QEP
0x40003800 - 0x40003FFF	EPWM
0x40004800 - 0x40004FFF	TD
0x40005000 - 0x400057FF	ADC
0x40005800 - 0x40005FFF	ACMP0/1/2, ACMP3
0x40006000 - 0x400067FF	BUART
0x40006800 - 0x40006FFF	OPA0/1/2
0x40007800 - 0x40007FFF	DAC
0x40008000 - 0x400087FF	I2C
0x40008800 - 0x40008FFF	UART
0x40009000 - 0x400097FF	TF
0x4000C000 - 0x4000C7FF	WDOG
0x4000C800 - 0x4000CFFF	WDOG
0x4000E000 - 0x4000E7FF	SYSCFG, GPIOCFG

## 4. 32Bit 处理器

### 4.1. 概述

处理器是 32 位可配置的多级流水线 RISC 处理器，内嵌 AMBA-Lite 接口和嵌套向量中断控制器(NVIC)。具有可选的硬件调试功能，可以执行 Thumb 指令。它工作在两种模式：Thread 模式和 Handler 模式。当系统异常发生时，就进入 Handler 模式。从 Handler 模式返回时，执行异常返回。复位时系统进入 Thread 模式。Thread 模式也可由异常返回时进入。

### 4.2. 功能特性

- 32-位硬件单周期乘法器
- 系统接口支持小端(little-endian)数据访问
- 准确而及时的中断处理能力
- 加载、存储多个数据和多周期乘法指令可被终止然后重新开始从而实现快速中断处理
- 使用中断唤醒(WFI)与事件唤醒(WFE)指令进入低功耗的休眠模式，或者从中断退出休眠模式
- NVIC 特征
  - 32 个外部中断，每个中断具有 4 级优先级
  - 专用的不可屏蔽中断(NMI)
  - 同时支持电平和脉冲中断触发
  - 中断唤醒控制器(WIC)，支持极低功耗休眠模式
- 调试支持
  - 四个硬件断点
  - 两个观察点

- 用于非侵入式代码分析的程序计数采样寄存器(PCSR)
- 单步和向量捕获能力
- 总线接口
  - 提供简单的集成到所有系统外设和存储器的单一 32 AHB-Lite 系统接口
  - 支持 DAP(Debug Access Port)的单一 32 位的从机端口

### 4.3. 内核控制模块(SCB)

内核控制模块(SCB) 包括了内核相关信息和系统控制, 如系统配置、控制、异常报告, 涉及休眠模式控制和复位状态。

#### 4.3.1. 模块寄存器总表

寄存器名	地址	结构体(.h)	描述	复位值
ICSR	0xE000ED04	SCB	NMI 中断控制和状态寄存器	0x00000000
AIRCR	0xE000ED0C	SCB	应用程序中断控制和复位寄存器	0xFA050000
SCR	0xE000ED10	SCB	系统控制寄存器	0x00000000
CCR	0xE000ED14	SCB	配置和控制寄存器	0x00000208
SHPR2	0xE000ED1C	SCB	系统中断处理优先级寄存器2	0x00000000
SHPR3	0xE000ED20	SCB	系统中断处理优先级寄存器3	0x00000000

#### 4.3.2. 中断控制和状态寄存器(ICSR: 0xE000ED04)

中断控制和状态寄存器(ICSR) 用于标记和屏蔽可以屏蔽的中断。

描述位	名称	读写	描述	复位值
5: 0	VECTACTIVE	RO	当前活动异常代码。 0 = Thread 模式 非0: 当前活动异常代码。 (注: 当前值减16便是CMSIS IRQ 中断号, 对应到中断允许寄存器, 中断禁止寄存器, 中断清除挂起寄存器, 中断挂起寄存器和中断优先级寄存器)	0x0
11: 6			保留.	
17: 12	VECTPENDING	RO	最高优先级挂起中断异常代码。 0: 无挂起中断异常 非0: 最高优先级挂起中断异常代码	0x00
21: 18			保留.	
22	ISR_PENDING	RO	中断挂起标识, 不包含NMI和故障。 0: 没有挂起的中断 1: 有中断正在挂起.	0

描述位	名称	读写	描述	复位值
24: 23			保留.	
25	PENDSTCLR	WO	系统时钟(SysTick)异常清除挂起控制位。 写: 0: 无效 1: 清除系统时钟中断挂起。只写位。	0
26	PENDSTSET	RW	系统时钟(SysTick)异常允许挂起控制位。 写: 0: 无效 1: 修改系统时钟(SysTick)异常到挂起状态 读: 0: 系统时钟(SysTick)异常不在挂起状态 1: 系统时钟(SysTick)异常在挂起状态。	0
27	PENDSVCLR	WO	PendSV异常清除挂起控制位。 写: 0: 无效 1: 清除PendSV中断挂起。	0
28	PENDSVSET	RW	PendSV异常允许挂起控制位。 写: 0: 无效 1: 强制PendSV异常到挂起状态 读: 0: PendSV异常不在挂起状态 1: PendSV异常在挂起状态。	0
30: 29			保留.	
31	NMIPENDSET	RW	NMI 异常允许挂起控制位。 写: 0: 无效 1: 强制NMI异常到挂起状态 读: 0: NMI异常不在挂起状态 1: NMI异常在挂起状态 因为NMI是最高优先级中断, 在正常情况下当此位变成1时, 处理器会立即进入中	0

描述位	名称	读写	描述	复位值
			断处理程序并清除此位。如在中断处理程序中检测到此位为1，即意味着NMI中断异常再次发生。	

### 4.3.3. 应用中断控制和复位寄存器(AIRCR: 0xE00ED0C)

应用中断控制和复位寄存器(AIRCR) 包含数据读写大小端控制和系统重启控制。

描述位	名称	读写	描述	复位值
0	-	-	保留.	
1	VECTCLRACTIVE	WO	保留为调试用。此位只允许写0。否则CPU行为不可预知。	0
2	SYSRESETREQ	WO	系统复位请求: 0: 无效 1: 触发系统级别复位。	0
14: 3	-	-	保留	
15	ENDIANESS	RO	数据访问大小端模式设定: 0: 小端模式(Little-endian) 1: 大端模式(Big-endian).	0
31: 16	VECTKEY	WO	寄存器启动钥匙: 只有写入0x05FA 到VECTKEY时, 寄存器修改才有效。否则, 写入无效。	0xFA05

### 4.3.4. 系统控制寄存器(SCR: 0xE00ED10)

系统控制寄存器(SCR) 控制系统进入/退出省电模式。

描述位	名称	读写	描述	复位值
0	-		保留	
1	SLEEPONEXIT	R/W	标志当系统从Handler模式返回到Thread模式是否进入睡眠模式: 0: 不进入睡眠状态当系统返回到Thread模式。 1: 进入睡眠或深度睡眠当系统从中断处理程序(ISR)返回到Thread模式。设置此位为1可以防止中断驱动的应用返回到一个空的主程序中。	0
2	SLEEPDEEP	R/W	控制处理器在低功耗模式下进入睡眠或深度睡眠: 0: 睡眠 1: 深度睡眠.	
3	-		保留.	

描述位	名称	读写	描述	复位值
4	SEVONPEND	R/W	挂起时发送事件位: 0: 只允许中断或事件唤醒处理器, 禁止中断被排除。 1: 允许所有中断或事件唤醒处理器, 包括禁止中断。当一个事件或中断进入挂起状态, 事件信号会把处理器从WFE状态中唤醒。如处理器不在等待事件唤醒, 那么这个事件将会被记录并影响下一个WFE。SEV命令也会唤醒处理器。	0
31: 5	-		保留	

SCR(系统控制)寄存器可以选择使用立即休眠还是退出时休眠, 当 SCR 寄存器的 SLEEPONEXIT 位为 0 的时候使用立即休眠, 当为 1 的时候使用退出时休眠。

#### 4.3.5. 配置和控制寄存器(CCR: 0xE00ED14)

配置和控制寄存器(CCR) 是只读寄存器。

描述位	名称	读写	描述	复位值
2: 0	-		保留	
3	UNALIGN_TRP		始终读为1。指示所有非对齐的数据访问将导致HardFault故障。	1
8: 4	-		保留	
9	STKALIGN		始终读为1。指示进入异常时堆栈按8 字节对齐。进入异常时, 处理器使用入栈的PSR 的位[9] 来指示栈对齐。从异常中返回时, 处理器使用这个入栈的位来恢复正确的栈对齐。	1
31: 10	-		保留	

#### 4.3.6. 系统中断处理优先级寄存器(SHPR2: 0xE00ED1C)

描述位	名称	读写	描述	复位值
29: 0	-		保留	
31: 30	PRI_11		中断11 – SVCall 的优先级: “0”表示最高优先级, “3”表示最低优先级	00

#### 4.3.7. 系统中断处理优先级寄存器(SHPR3: 0xE00ED20)

描述位	名称	读写	描述	复位值
21: 0	-		保留	

描述位	名称	读写	描述	复位值
23: 22	PRI_14		中断14 – PendSV 的优先级: “0”表示最高优先级, “3”表示最低优先级	00
29: 24	-		保留	
31: 30	PRI_15		中断15 – SysTick 的优先级: “0”表示最高优先级, “3”表示最低优先级	00

SVC、PendSV 和 SysTick 可编程的中断多用于在操作系统之上的软件开发中。SVC 用于产生系统函数的调用请求，例如，操作系统不让用户程序直接访问硬件，而是通过提供一些系统服务函数，用户程序使用 SVC 发出对系统服务函数的呼叫请求，以这种方法调用它们来间接访问硬件。因此，当用户程序想要控制特定的硬件时，它就会产生一个 SVC 异常，然后操作系统提供的 SVC 异常服务例程得到执行，它再调用相关的操作系统函数，后者完成用户程序请求的服务。

另一个相关的异常是 PendSV，它和 SVC 协同使用。一方面，SVC 异常是必须立即得到响应的(若因优先级不比当前正处理的高，或是其它原因使之无法立即响应，将上访成 Hardfault)，应用程序执行 SVC 时都是希望所需的请求立即得到响应。另一方面，PendSV 则不同，它是可以像普通的中断一样被挂起的(不像 SVC 那样会上访)。OS 可以利用它“延期执行”一个异常——直到其它重要的任务完成后才执行动作。挂起 PendSV 的方法是：程序中往 NVIC 的 PendSV 挂起寄存器中写 1。挂起后，如果优先级不够高，则将延期等待执行。PendSV 的典型使用场合是在上下文切换时(在不同任务之间切换)。有两个就绪的任务，上下文切换被触发的场合可以是：

执行一个系统调用系统滴答定时器(SysTick)中断，(轮转调度中需要)

## 4.4. 系统控制(SYSCTRL)

### 4.4.1. 模块寄存器总表

寄存器名	地址	结构体(.h)	寄存器描述	复位值
DPSLEEPCTRL	0x4001F004		深度睡眠控制寄存器	0x00000000
LOCKUPRESET	0x4001F008		LOCKUP复位控制寄存器	0x00000000
RESETIFO	0x4001F010		复位状态信息标志寄存器	0x00000000

### 4.4.2. 内核睡眠控制寄存器(DPSLEEPCTRL: 0x4001F004)

描述位	名称	读写	描述	复位值
0	DPSLEEP	RW	控制处理器在低功耗模式下进入睡眠或深度睡眠	0

描述位	名称	读写	描述	复位值
			0: 睡眠 1: 深度睡眠 要进入深度睡眠, 错误!未找到引用源。 SCR寄存器的SLEEPDEEP也要置1	
31: 1	-	-	保留	

#### 4.4.3. 内核锁定复位使能寄存器(LOCKUPRESET: 0x4001F008)

描述位	名称	读写	描述	复位值
0	LOCKUPRSTEN	RW	为1表示允许LOCKUP触发系统复位	0
31: 1	-	-	保留	

#### 4.4.4. 内核复位状态标志位(RESETIFO: 0x4001F010)

描述位	名称	读写	描述	复位值
0	SYSRESETFLG	RW	为1表示由SYSRESET触发了复位。即AIRCR的SYSRESETREQ置1触发。写1清除标志位	0
1	WDTRSTFLG	RW	为1表示由WDT触发了复位, 写1清除标志位	0
2	LOCKUPRSTFLG	RW	为1表示LOCKUP信号触发了复位, 写1清除标志位	0
31: 3	-	-	保留	

### 4.5. 系统定时器(SysTick)

#### 4.5.1. 模块寄存器总表

寄存器名	地址	结构体(.h)	寄存器描述	复位值
CTRL	0xE000E010	SysTick	SysTick控制和状态寄存器	0x00000000
LOAD	0xE000E014	SysTick	SysTick重加载寄存器	0x00FFFFFF
VAL	0xE000E018	SysTick	SysTick当前计数值寄存器	0x00000000
CALIB	0xE000E01C	SysTick	SysTick校准值寄存器	0x40000000

#### 4.5.2. SysTick 控制和状态寄存器(CTRL: 0xE000E010)

描述位	名称	读写	描述	复位值
0	EN	RW	SysTick 计数器使能位。 0: 禁止 1: 使能	0
1	INT	RW	SysTick中断使能, 为1时SysTick计数到0会触发异常。	0
2	CLKSEL	RW	为1表示SysTick使用内核系统时钟。	0
15: 3	-	-	保留	
16	COUNTFLAG	R	当SysTick计数到0时, 该位置1。读寄存器时, 该位清零。	0
31: 17	-	-	保留	

#### 4.5.3. SysTick 重加载寄存器(LOAD: 0xE000E014)

描述位	名称	读写	描述	复位值
23: 0	RELOAD	RW	SysTick装载寄存器, 当计数器减到0时, 此值会重载到SYST_CVR寄存器。	0xFFFFFFFF
31: 24			保留	

#### 4.5.4. SysTick 当前值寄存器(VAL: 0xE000E018)

描述位	名称	读写	描述	复位值
23: 0	CURRENT	RW	SysTick计数当前值。写入任何值都会清零寄存器, COUNTFLAG也会清零。	0x000000
31: 24			保留	

#### 4.5.5. SysTick 校准值寄存器(CALIB: 0xE000E01C)

描述位	名称	读写	描述	复位值
23: 0	TENMS	R	10ms校准值, 如果这个值读出为0, 则此校准值是不确定的。	0
31: 24	-	-	保留	0
30	SKEW	R	为1表示TENMS不准确。	1
31	NOREF	R	为1表示SysTick使用内部系统时钟, 为0表示有外部参考时钟可供使用。	0

## 5. 系统控制(SYSCFG)

### 5.1. 模块寄存器总表

系统控制的主要寄存器列表如下:

寄存器名	地址	结构体(.h)	寄存器描述	复位值
SYSCON	0x4000E000	SYSCFG	系统控制寄存器	0x0307136C
CLKCON	0x4000E004	SYSCFG	时钟控制寄存器	0x00000003
IRHTRIM	0x4000E008	SYSCFG	内部振荡器校准控制寄存器	0x00000000
VBGTRIM	0x4000E00C	SYSCFG	基准电平校准控制寄存器	0x00
LDOCON	0x4000E010	SYSCFG	LDO控制寄存器	0x03
LVCON	0x4000E014	SYSCFG	低压复位控制寄存器	0x0089
SYSKEY	0x4000E018	SYSCFG	系统保护寄存器	0x00000000
EINTCON	0x4000E01C	SYSCFG	外部IO中断控制寄存器	0x00000000
EINTFCON	0x4000E020	SYSCFG	外部IO中断控制寄存器	0x00000000

SYSSTA	0x4000E024	SYSCFG	系统状态寄存器	0x00000000
PERIEN	0x4000E040	SYSPEN	外设使能寄存器	0x0000
PCLKSEL	0x4000E044	SYSPEN	外设时钟控制寄存器	0x0000
ANA_CON	0x4000E048	SYSPEN	模拟功能控制寄存器	0x0000

### 5.1.1. 系统保护寄存器(SYSKEY: 0x4000E018)

描述位	名称	读写	描述	复位值
7:0	SYSKEY	R/W	往该寄存器写入正确KEY值0xA5后才能修改重要寄存器，该寄存器值在7个时钟周期后自动复位。受保护的寄存器为SYSCON、CLKCON、LVCON、IRHTRIM、VREFTRIM、LDOCON。	0x00

## 5.2. 系统时钟(SYSCLK)

### 5.2.1. 系统时钟框图

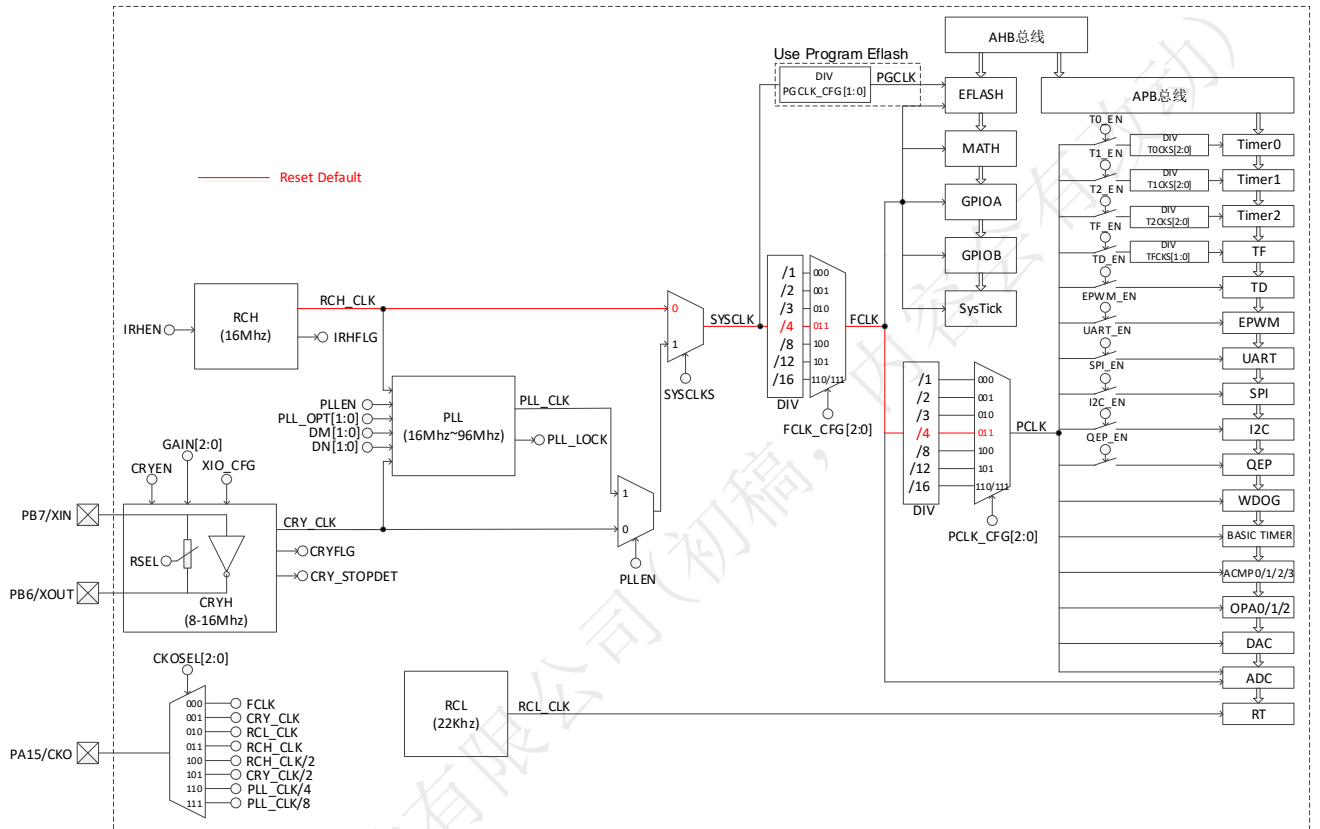


图 5-2-1 结构框图

### 5.2.2. 系统时钟寄存器

#### 5.2.2.1. 系统控制寄存器 SYSCON (0x4000E000)

描述位	名称	读写	描述	复位值
0	SYSCLKS	R/W	CPU系统时钟选择位。 0: 选择内部高频振荡器作为系统时钟源 1: 选择外部晶振或PLL作为系统时钟源	0
1	CSSSEN	R/W	外部时钟失效检测使能。 0: 禁止外部时钟失效检测 1: 使能外部时钟失效检测	0

描述位	名称	读写	描述	复位值
4: 2	FCLK_CFG [2: 0]	R/W	FCLK时钟频率控制。 000: SYSCLK时钟不分频 001: SYSCLK 2分频 010: SYSCLK 3分频 011: SYSCLK 4分频 100: SYSCLK 8分频 101: SYSCLK 12分频 110: SYSCLK 16分频 111: SYSCLK 16分频	011
7: 5	PCLK_CFG [2: 0]	R/W	外设PCLK时钟频率控制。 000: FCLK时钟不分频 001: FCLK 2分频 010: FCLK 3分频 011: FCLK 4分频 100: FCLK 8分频 101: FCLK 12分频 110: FCLK 16分频 111: FCLK 16分频	011
10: 8			保留	
11	STOP_CFG	R/W	见5.3.1章节	
12	WKEN_EINT0	R/W		
13	WKEN_EINT1	R/W		
14	WKEN_EINT2	R/W		
15	WKEN_EINT3	R/W		
16	WKEN_LVD	R/W		
17	WKEN_RT	R/W		
18	RTEN	R/W	见11.1章节	
21: 19	RT_CFG[2: 0]	R/W		
23: 22	PGCLK_CFG [1: 0]	R/W	见26.2.1章节	
24	PGCLKEN	R/W		
25	DEBUG_MODE	R/W	见28.1章节	
31: 26			保留	

### 5.2.2.2. 时钟控制寄存器 CLKCON (0x4000E004)

描述位	名称	读写	描述	复位值
0	IRHEN	R/W	内部高频振荡器使能。 0: 内部高频振荡器禁止	1

描述位	名称	读写	描述	复位值
			1: 内部高频振荡器使能	
1	IRHFLG	RO	内部高频振荡器稳定标志位。 0: 内部高频振荡器未稳定 1: 内部高频振荡器稳定	1
2	CRYEN	R/W	晶振使能。 0: 晶振禁止 1: 晶振使能	0
3	CRYFLG	RO	在使用RCH作为PLL时钟源: 0: PLL未稳定 1: PLL稳定 在使用CRY作为PLL时钟源: 0: 晶振未稳定 1: 晶振稳定 PLL稳定标志为PLL_LOCK, 见5.2.2.4章节	0
4	RSEL	R/W	晶振电路内置反馈电阻选择。 0: 10M 1: 20M	0
7: 5	GAIN[2: 0]	R/W	CRY增益选择。 000: 增益0 001: 增益1 010: 增益2 011: 增益3 100: 增益4 101: 增益5 110: 增益6 111: 增益7	000
9: 8	STABLE[1: 0]	R/W	停振检测时钟选择: 00: RCH 65536分频, 检测时钟周期为4ms 01: RCH 32768分频, 检测时钟周期为2ms 10: RCH 16384分频, 检测时钟周期约为1ms 11: RCH 8192分频, 检测时钟周期约为0.5ms	00
10	XIO_CFG	R/W	晶振电路模式控制。 0: 正常模式 1: BYPASS模式, 从XIN(PB7)脚外灌时钟	0
13: 11	CKOSEL[2: 0]	R/W	选择 CKO时钟输出源。 000: FCLK 001: CRY 010: RCL 011: RCH	000

描述位	名称	读写	描述	复位值
			100: RCH/2 101: CRY/2 110: PLL/4 111: PLL/8	
15: 14	DM[1: 0]	R/W	PLL输入时钟分频控制信号。 00: 输入时钟1分频 01: 输入时钟2分频 10: 输入时钟3分频 11: 输入时钟4分频	00
20: 16	DN[4: 0]	R/W	反馈时钟分频控制信号，分频比为DN[4: 0]+1,例如: 00000: 反馈时钟1分频 ... 11111: 反馈时钟32分频	00000
22: 21	PLL_OPT[1: 0]	R/W	PLL基准时钟选择。 00: 内部16MHz高频振荡器时钟2分频 01: 内部16MHz高频振荡器时钟 10: 外部高频晶振时钟2分频 11: 外部高频晶振时钟	00
23	PLLEN	R/W	PLL开启使能位	0
31: 24			保留	

### 5.2.2.3.内部振荡器校准控制位寄存器 IRHTRIM (0x4000E008)

描述位	名称	读写	描述	复位值
7: 0	IRHTRIM[7: 0]	R/W	内部高频振荡器校验控制，上电复位后自动从EFLASH信息区载入。	0x00
11: 8	IRHTRIM_PT [3: 0]	R/W	内部高频振荡器校验控制，上电复位后自动从EFLASH信息区载入。	0000
17: 12	IRLTRIM[5: 0]	R/W	内部低频振荡器校验控制，上电复位后自动从EFLASH信息区载入。	000000
31: 18			保留	

### 5.2.2.4.系统状态寄存器 SYSSTA (0x4000E024)

描述位	名称	读写	描述	复位值
0	PORFLG	R	见5.6.1章节	0
1	LVRFLG	R	见5.6.1章节	0
2	EXTFLG	R	见5.6.1章节	0

描述位	名称	读写	描述	复位值
3	RTFLG	R	见11.2章节	0
4	LVDFLG	R	见5.3.1.4章节	0
5	CRY_STOPDET	R	外部晶振停振标志位 CRYEN=0时, STOPDET=1 CRYEN=1时, 发生停振, STOPDET=1	0
6	PLL_LOCK	R	使用CRY作为PLL时钟源时, PLL锁定标志位。 0: PLL未锁定 1: PLL锁定, LOCK输出高电平	0
7	LVD	R	见5.3.1.4章节	0
8	WTDFLG	R	见5.6.1章节	0
15: 9			保留	
16	PORFLG_CLR	W	见5.6.1章节	0
17	LVRFLG_CLR	W	见5.6.1章节	0
18	EXTFLG_CLR	W	见5.6.1章节	0
19	RTFLG_CLR	W	见11.2章节	0
20	LVDFLG_CLR	W	见5.3.1.4章节	0
23: 21			保留	0
24	WTDFLG_CLR	W	见5.6.1章节	0
31: 25			保留	

### 5.2.2.5. 外设使能寄存器 PERIEN (0x4000E040)

描述位	名称	读写	描述	复位值
0	T0_EN	R/W	Timer0模块时钟使能。 0: 禁止 1: 使能	0
1	T1_EN	R/W	Timer1模块时钟使能。 0: 禁止 1: 使能	0
2	T2_EN	R/W	Timer2模块时钟使能。 0: 禁止 1: 使能	0
3	TF_EN	R/W	TF模块时钟使能。 0: 禁止 1: 使能	0
4	ADC_EN	R/W	ADC模块时钟使能。 0: 禁止 1: 使能	0
5	EPWM_EN	R/W	EPWM模块时钟使能。	0

描述位	名称	读写	描述	复位值
			0: 禁止 1: 使能	
6		R/W	保留	
7	TD_EN	R/W	TD模块时钟使能(使用TD模块时, TDCKEN也要使能)。 0: 禁止 1: 使能	0
8	UART_EN	R/W	UART模块时钟使能。 0: 禁止 1: 使能	0
9	SPI_EN	R/W	SPI模块时钟使能。 0: 禁止 1: 使能	0
10	I2C_EN	R/W	I2C模块时钟使能。 0: 禁止 1: 使能	0
12: 11			保留	
13	QEP_EN	R/W	QEP模块时钟使能。 0: 禁止 1: 使能	0
15: 14			保留	

### 5.2.2.6. 外设时钟控制寄存器 PCLKSEL (0x4000E044)

描述位	名称	读写	描述	复位值
2: 0	T0CKS[2: 0]	R/W	Timer0模块时钟分频选择寄存器。 000: PCLK 001: PCLK/2 010: PCLK/4 011: PCLK/8 100: PCLK/16 101: PCLK/32 110: PCLK/64 111: PCLK/128	000
5: 3	T1CKS[2: 0]	R/W	Timer1模块时钟分频选择寄存器。 000: PCLK 001: PCLK/2 010: PCLK/4 011: PCLK/8	000

描述位	名称	读写	描述	
			100: PCLK/16 101: PCLK/32 110: PCLK/64 111: PCLK/128	
8: 6	T2CKS[2: 0]	R/W	Timer2模块时钟分频选择寄存器。 000: PCLK 001: PCLK/2 010: PCLK/4 011: PCLK/8 100: PCLK/16 101: PCLK/32 110: PCLK/64 111: PCLK/128	000
10: 9	TFCKS[1: 0]	R/W	TF模块时钟分频选择寄存器。 00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	00
15: 11			保留	

### 5.2.3. 锁相环倍频器(PLL)

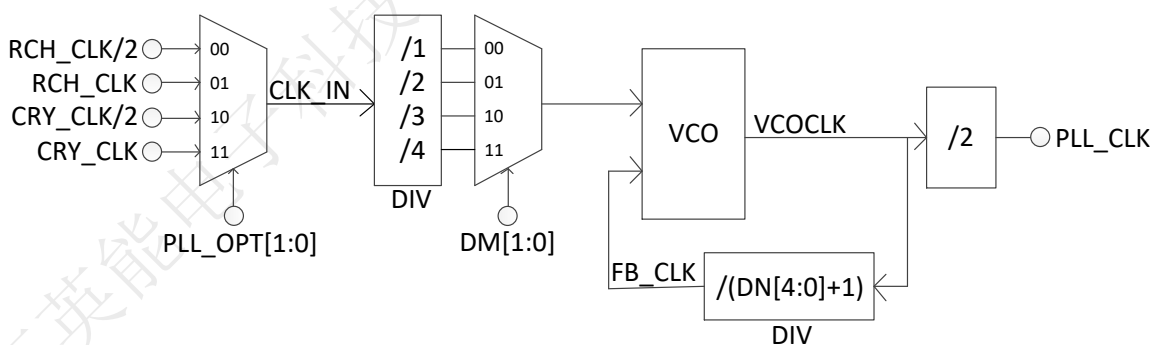


图 5-2-3-1 结构框图

锁相环倍频器的时钟源可以选择内置振荡器 RCH 和外置晶体振荡器 CRY，有丰富的倍频系数设置，典型的倍频范围: 16Mhz 到 96Mhz。PLL 锁相稳定后，SYSSTA 寄存器的 PLL\_LOCK 位会置 1。在系统时钟 SYSCLK 切换为 PLL 时钟时，PLL\_LOCK 必须为 1。

$$F_{PLL\_CLK} = \frac{F_{CLK\_IN} \times (DN[4:0] + 1)}{(DM[1:0] + 1) \times 2}$$

图 5-2-3-2 结构框图

### 5.2.4. 系统工作时钟

支持多种系统时钟分频，用户可以通过系统控制寄存器 **SYSCON** 中的 **FCLK\_CFG[2:0]**、**PCLK\_CFG[2:0]** 时钟分频控制位来选择各种分频。当用户希望低功耗应用的时候，**FCLK** 可以采用系统时钟 **SYSCLK** 的 16 分频。默认 **FCLK** 采用系统时钟 **SYSCLK** 的 4 分频，

用户可以通过外设时钟使能寄存器配置相应的外设是否使能，默认所有的外设均未使能。也可以单独配置某个外设时钟分频以减少系统功耗。

### 5.2.5. 时钟切换

共支持 3 种时钟源作为系统时钟 **SYSCLK**，内部高频振荡器 **RCH**、外部晶振 **CRY**、**PLL** 时钟。用户可以通过 **SYSCON** 中的 **SYSCLKS** 位来选择。**SYSCLKS** 为 0 时选择内部高频振荡器，为 1 时选择外部晶振或 **PLL** 时钟。当 **PLL** 使能时，**SYSCLKS** 为 1 选择 **PLL** 时钟，当 **PLL** 不使能时，**SYSCLKS** 为 1 选择外部晶振 **CRY**。

系统默认启动选择内部高频振荡器作为系统时钟，需要切换成其他时钟时，可以配置 **CLKCON** 寄存器的相应寄存器位使得相应时钟使能。通过查询 **CLKCON** 中的时钟标志位判断该时钟是否稳定。在切换时钟时需要确保目标时钟已经稳定才可以开始切换时钟操作。如果时钟切换完成后需要关闭原有时钟，需要三个指令周期后才能关闭。

当时钟从旧时钟切换到新时钟刚好发生在进 **STOP** 模式之前时，需要确保时钟已经切换完成，否则可能导致丢失系统时钟。

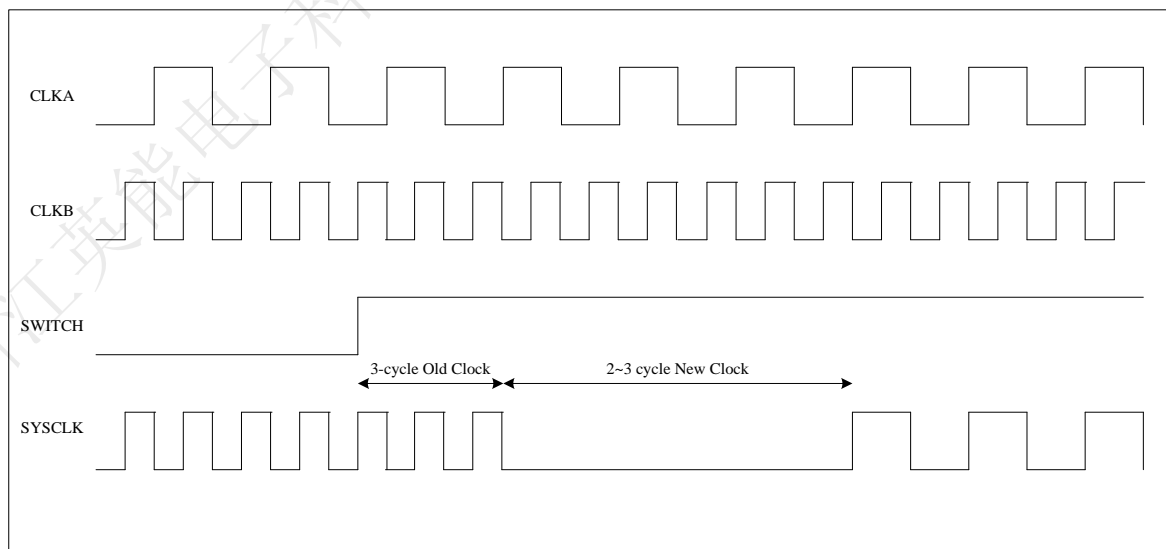


图 5-2-5-1 时钟切换示意图

● 时钟安全系统:

当系统时钟采用外部晶振时，为防止外部晶振失效，内部集成停振检测电路。可以通过 **SYSCON** 中的 **CSSSEN** 位使能停振检测电路。停振检测的使能位只有在系统时钟选用外部晶振时才能配置成功。一旦 **CSSSEN** 位被配置为 1，除复位操作外，不能再将该位清零。使能 **CSSSEN** 后，内部高频振荡器将不能关闭，即使 **IRHEN** 位被清除为 0。

● 停振功能的使能操作:

切换外部晶振时钟作为系统时钟

等待外部晶振时钟稳定时间后，使能 **CSSSEN** 位

检测外部晶振时钟是否发生失效

停振检测的机制，如果使用 **RCH** 的 65536 分频作为采样时钟，**RCH** 为 16MHz，则采样时钟的周期约为 4ms，在采样时钟的上升沿，检测信号被拉低，当停振检测功能使能时，外部时钟的上升沿将会置位检测信号。在采样时钟的下降沿检测该检测信号的高低电平，若为高电平，则表示当前外部时钟正常，若为低电平，则表示外部时钟已经停振。此时若停振检测中断使能，则进入中断程序。当检测到外部时钟停振后，系统将会自动切换回 **RCH** 作为系统时钟，同时 **SYSCON** 的 **SYSCCLKS** 位变为 0。

停振检测的机制的采样时钟周期可以由 **STABLE[1:0]** 来配置，**RCH** 为 16MHz:

00: **RCH** 65536 分频，则采样时钟的周期约为 4ms

01: **RCH** 32768 分频，则采样时钟的周期约为 2ms

10: **RCH** 16384 分频，则采样时钟的周期约为 1ms

11: **RCH** 8192 分频，则采样时钟的周期约为 0.5ms

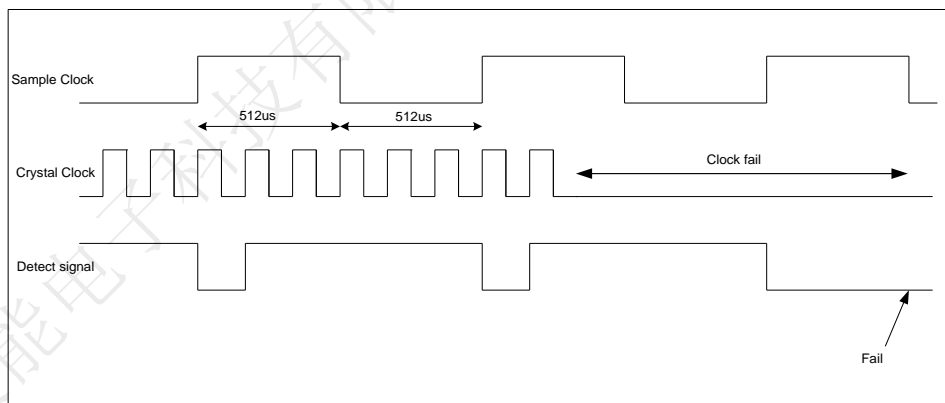


图 5-2-5-2 时钟切换示意图

## 5.3. 电源管理(PMU)

### 5.3.1. 电源管理寄存器

#### 5.3.1.1. 系统控制寄存器 SYSCON (0x4000E000)

描述位	名称	读写	描述	复位值
0	SYSCCLKS	R/W	见5.2.2章节。	
1	CSSSEN	R/W		
4: 2	FCLK_CFG[2: 0]	R/W		
7: 5	PCLK_CFG[2: 0]	R/W		
10: 8			保留	
11	STOP_CFG	R/W	DEEPSLEEP模式下，关闭高频时钟源控制。 0: 不关闭高频时钟源 1: 关闭高频时钟源	0
12	WKEN_EINT0	R/W	外部中断0唤醒使能。 0: 不允许外部中断0唤醒时钟； 1: 允许外部中断0唤醒时钟。	1
13	WKEN_EINT1	R/W	外部中断1唤醒使能。 0: 不允许外部中断1唤醒时钟； 1: 允许外部中断1唤醒时钟。	0
14	WKEN_EINT2	R/W	外部中断2唤醒使能。 0: 不允许外部中断2唤醒时钟； 1: 允许外部中断2唤醒时钟。	0
15	WKEN_EINT3	R/W	外部中断3唤醒使能。 0: 不允许外部中断3唤醒时钟； 1: 允许外部中断3唤醒时钟。	0
16	WKEN_LVD	R/W	LVD事件唤醒使能。 0: 不允许LVD事件唤醒时钟； 1: 允许LVD事件唤醒时钟。	1
17	WKEN_RT	R/W	RT定时器中断唤醒使能。 0: 不允许RTIMER计数溢出中断唤醒时钟； 1: 允许RTIMER计数溢出中断唤醒时钟。	1
18	RTEN	R/W	见11.1章节	
21: 19	RT_CFG[2: 0]	R/W		
23: 22	PGCLK_CFG [1: 0]	R/W	见26.2.1章节	
24	PGCLKEN	R/W		
25	DEBUG_MODE	R/W	见28.1章节	

描述位	名称	读写	描述	复位值
31: 26			保留	

### 5.3.1.2.LDO 控制寄存器 LDOCON(0x4000E010)

描述位	名称	读写	描述	复位值
1: 0	DRV[1: 0]	R/W	LDO_SOC的驱动模式选择。 00: 10UA工作模式 01: 50UA工作模式 10: 150UA工作模式 11: 200UA工作模式	11
2	MODE	R/W	STOP模式时，LDO的工作模式。 0: LDO工作在正常工作模式(默认状态) 1: LDO工作在LPVR低功耗模式。	0
4: 3	VSEL[1: 0]	R/W	调节LPVR输出电压值。 00: LPVR输出1.5V 01: LPVR输出1.2V 10: LPVR输出1.1V 11: LPVR输出1.0V	00
7: 5			保留	

### 5.3.1.3.低压复位控制寄存器 LVCON(0x4000E014)

描述位	名称	读写	描述	复位值
0	LVREN	R/W	低压复位使能位。 0: 低压复位禁止 1: 低压复位使能	1
3: 1	LVRSEL[2: 0]	R/W	低压复位电平选择位。 000: 1.6V 001: 1.8V 010: 2V 011: 2.5V 100: 2.8V 101: 3V 110: 3.5V 111: 4V	100
4	LVDEN	R/W	低压检测使能位。 0: 低压检测禁止 1: 低压检测使能	0

描述位	名称	读写	描述	复位值
7: 5	LVDSEL[2: 0]	R/W	低压检测电平选择位。 000: 2.0V 001: 2.2V 010: 2.4V 011: 2.7V 100: 2.9V 101: 3.1V 110: 3.6V 111: 4.5V	100
8	LVDFSEL	R/W	低压检测滤波使能位。 0: 低压检测不滤波 1: 低压检测滤波	0
15: 9			保留	

#### 5.3.1.4. 系统状态寄存器 SYSSTA (0x4000E024)

描述位	名称	读写	描述	复位值
0	PORFLG	R	见5.6.1章节	0
1	LVRFLG	R	见5.6.1章节	0
2	EXTFLG	R	见5.6.1章节	0
3	RTFLG	R	见11.2章节	0
4	LVDFLG	R	LVD检测标志位，对LVDFLG_CLR写1清零。 0: 未发生LVD事件 1: 发生LVD事件	0
5	CRY_STOPDET	R	见5.2.2.4章节	0
6	PLL_LOCK	R	见5.2.2.4章节	0
7	LVD	R	LVD模块状态输出。 0: 未发生LVD事件 1: 发生LVD事件	0
8	WTDFLG	R	见5.6.1章节	0
15: 9			保留	
16	PORFLG_CLR	W	见5.6.1章节	0
17	LVRFLG_CLR	W	见5.6.1章节	0
18	EXTFLG_CLR	W	见5.6.1章节	0
19	RTFLG_CLR	W	见11.2章节	0
20	LVDFLG_CLR	W	写1操作，产生清零脉冲，清除LVDFLG标志位。 写0无效	0
23: 21			保留	0
24	WTDFLG_CLR	W	见5.6.1章节	0

描述位	名称	读写	描述	复位值
31: 25			保留	

### 5.3.2. 睡眠工作模式 (SLEEP 工作模式)

用户可以通过将 WFE、WFI 命令进入待机模式。在 SLEEP 模式下，CPU 时钟将会停止，外设的时钟依然存在，内部中断和外设等依然可以工作，各个 I/O 口依然保持为原来的工作状态。

有两种方式可以退出 SLEEP 模式：

1. 执行复位操作，外部复位，上电复位，LVR 复位可以退出 SLEEP 模式。采用复位操作退出 SLEEP 模式时，系统和外围控制寄存器将全部复位到默认状态。如果全局中断未使能，那么复位是退出 SLEEP 模式的唯一方法。

2. 使能的中断发生中断事件，也会退出 SLEEP 模式。退出 SLEEP 模式后，系统会优先处理中断服务程序，中断服务处理程序执行完成后返回执行下一条指令。

有一种特殊情况：当 CPU 运行在 EFLASH 指令空间，且执行字节编程和 SECTOR 擦除时，这时候必须停止 CPU 的时钟。待完成字节编程和 SECTOR 擦除后，硬件再恢复 CPU 时钟。

### 5.3.3. 深度睡眠工作模式 (DPSLEEP 工作模式)

配置 SCR 的 SLEEPDEEP 为 1 且 DPSLEEPCTRL 的 DPSLEEP 为 1 时，用户可以通过将 WFE、WFI 命令进入深度待机模式。在 DPSLEEP 模式下，CPU 和调试模块时钟将会停止，外设的时钟依然存在，各个 I/O 口依然保持为原来的工作状态。

有两种方式可以退出 DPSLEEP 模式：

1. 执行复位操作，外部复位，上电复位，LVR 复位可以退出 DPSLEEP 模式。采用复位操作退出 DPSLEEP 模式时，系统和外围控制寄存器将全部复位到默认状态。如果全局中断未使能，那么复位是退出 DPSLEEP 模式的唯一方法。

2. 使能的中断发生中断事件，也会退出 DPSLEEP 模式。退出 DPSLEEP 模式后，系统会优先处理中断服务程序，中断服务处理程序执行完成后返回执行下一条指令。

### 5.3.4. 停机工作模式 (STOP 工作模式)

配置 SCR 的 SLEEPDEEP 为 1，DPSLEEPCTRL 的 DPSLEEP 为 1 且 SYSCON 的 STOP\_CFG 为 1 时，用户可以通过将 WFE、WFI 命令进入 STOP 模式。在 STOP 模式下，CPU 和外设的时钟都被停止，系统时钟 SYSCLK 会被关闭，内置高频振荡器 RCH 也会关闭。如果进 STOP 前，SYSCON 的 SYSCLKS 为 1，进入 STOP 模式后 SYSCLKS 会被置 0，系统时钟切换到 RCH 时钟，同时晶振 CRY 和 PLL 的使能会硬件自动关闭。STOP 唤醒后软件切换到别的时钟，需要重新使能晶振 CRY 和 PLL。

有三种方式可以退出 STOP 模式:

1、执行复位操作，外部复位，上电复位，LVR 复位都可以退出 STOP 模式。采用复位操作退出 STOP 模式后，系统和外围控制寄存器将全部复位到默认状态。在复位退出后，需要等待一段时间才能正常读取指令并运行。

2、采用外部中断唤醒退出 STOP 模式，外部中断不能选择数字滤波，否则将无法唤醒 STOP 模式。利用外部中断退出 STOP 模式时，系统和外围控制寄存器的值保持不变。

3、采用 RT 计时溢出唤醒退出 STOP 模式。

4、采用 LVD 检测事件唤醒退出 STOP 模式。

在停机模式下，LVR、ADC、DAC、ACMP0/1/2/3、OPA 等模拟外设，保持停机前的状态。

## 5.4. 内部参考电压

### 5.4.1. 内部参考电压框图

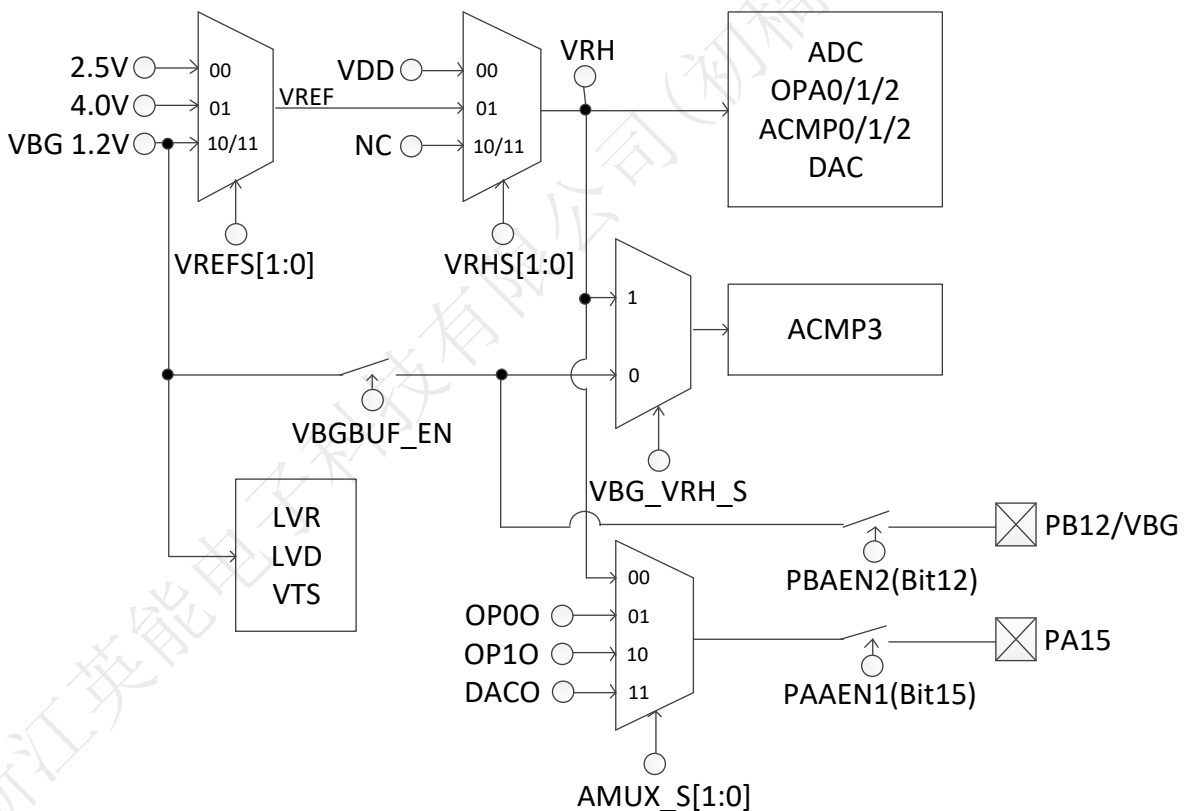


图 5-4-1 内部参考电压示意图

### 5.4.2. 内部基准电平校准控制位寄存器 (VBGTRIM: 0x4000E00C)

描述位	名称	读写	描述	复位值
5: 0	VBGTRIM[5: 0]	R/W	VBG校验控制，上电复位后自动从EFLASH信息区载入。	000000
7: 6			保留	

### 5.4.3. 模拟功能控制寄存器 (ANA\_CON: 0x4000E048)

描述位	名称	读写	描述	复位值
4: 0			保留	
6: 5	VRHS[1: 0]	R/W	内部参考电压VRH选择位: 00: 供电电压VDD(默认值) 01: 内置基准电压VREF(1.2V/2.5V/4V) 1X: 无效	00
8: 7	VREFS[1: 0]	R/W	内部基准电压VREF选择位: 00: 2.5V 01: 4.0V 1X: VBG, 1.2V	00
9	VTSEN	R/W	见5.5.2章节	0
10	AMUX2_S	R/W	见23.3.4章节	
11	VTS_SEL	R/W	见5.5.2章节	0
12	VBGBUF_EN	R/W	VBG 输出使能，高有效 0: 禁止 1: 使能	0
13	VBG_VRH_S	R/W	见22.3.7章节	0
15: 14	AMUX_S[1: 0]	R/W	AMUX_OUT连接到PA15的YA1,做模拟信号测试。 00: VRH 01: OP00 10: OP10 11: DAC0	00

## 5.5. 温度传感器(VTS)

### 5.5.1. 温度传感器框图

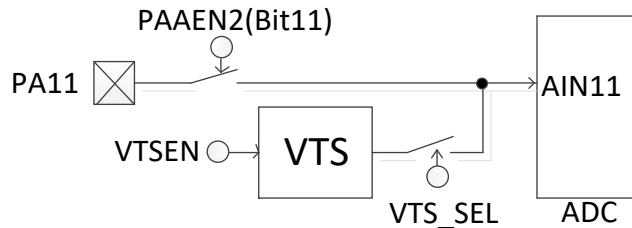


表 5-5-1 温度传感器示意图

### 5.5.2. 模拟功能控制寄存器(ANA\_CON: 0x4000E048)

描述位	名称	读写	描述	复位值
4: 0			保留	
6: 5	VRHS[1: 0]	R/W	见5.4.3章节	00
8: 7	VREFS[1: 0]	R/W	见5.4.3章节	0
9	VTS_EN	R/W	VTS使能控制位。 0: 关闭; 1: 开启。	0
10	AMUX2_S	R/W	见23.3.4章节	
11	VTS_SEL	R/W	温度传感器输出与ADC的AIN11通断使能位: 0: VTS输出与AIN11断开 1: VTS输出与AIN11闭合 注: 当AIN11选择VTS功能时, 除了VTS_SEL要置1, PAAEN2的Bit11要置0, 否则PA11的电压会影响VTS电压。 反过来当AIN11选择PA11输入时, 除了PAAEN2的Bit11要置1, VTS_SEL也要置0。	0
12	VBGBUF_EN	R/W	见5.4.3章节	0
13	VBG_VRH_S	R/W	见22.3.7章节	0
15: 14	AMUX_S[1: 0]	R/W	见5.4.3章节	0

### 5.5.3. 温度传感器计算公式

$$\text{温度}(\text{°C}) = -256.61 \times V_{\text{vts}} + 354.84$$

温度(°C)与  $V_{VTS}$ (V)电压对应表:

温度	$V_{VTS}$	温度	$V_{VTS}$	温度	$V_{VTS}$	温度	$V_{VTS}$	温度	$V_{VTS}$
-40	1.49	-20	1.41	0	1.34	20	1.26	40	1.18
-38	1.48	-18	1.40	2	1.33	22	1.25	42	1.18
-36	1.47	-16	1.40	4	1.32	24	1.24	44	1.17
-34	1.46	-14	1.39	6	1.31	26	1.24	46	1.16
-32	1.46	-12	1.38	8	1.31	28	1.23	48	1.15
-30	1.45	-10	1.37	10	1.30	30	1.22	50	1.15
-28	1.44	-8	1.37	12	1.29	32	1.21	52	1.14
-26	1.43	-6	1.36	14	1.28	34	1.21	54	1.13
-24	1.43	-4	1.35	16	1.27	36	1.20	56	1.12
-22	1.42	-2	1.34	18	1.27	38	1.19	58	1.12

温度	$V_{VTS}$	温度	$V_{VTS}$	温度	$V_{VTS}$	温度	$V_{VTS}$
60	1.11	80	1.03	100	0.96	120	0.88
62	1.10	82	1.02	102	0.95	122	0.87
64	1.09	84	1.02	104	0.94	124	0.86
66	1.08	86	1.01	106	0.93	126	0.86
68	1.08	88	1.00	108	0.93	128	0.85
70	1.07	90	0.99	110	0.92	130	0.84
72	1.06	92	0.99	112	0.91	132	0.83
74	1.05	94	0.98	114	0.90	134	0.83
76	1.05	96	0.97	116	0.89	136	0.82
78	1.04	98	0.96	118	0.89	138	0.81

## 5.6. 复位系统(RESET)

### 5.6.1. 系统状态寄存器(SYSSTA: 0x4000E024)

描述位	名称	读写	描述	复位值
0	PORFLG	R	上电复位状态位, 对PORFLG_CLR写1清零。 0: 未发生上电复位 1: 发生上电复位	0
1	LVRFLG	R	低压复位状态位, 对LVRFLG_CLR写1清零。 0: 未发生低压复位 1: 发生低压复位	0
2	EXTFLG	R	外部复位状态位, 对EXTFLG_CLR写1清零。 0: 未发生外部复位 1: 发生外部复位	0

描述位	名称	读写	描述	复位值
3	RTFLG	R	见11.2章节	0
4	LVDFLG	R	见5.3.1.4章节	0
5	CRY_STOPDET	R	见5.2.2.4章节	0
6	PLL_LOCK	R	见5.2.2.4章节	0
7	LVD	R	见5.3.1.4章节	0
8	WTDFLG	R	WTD0G复位标志位，对WTDFLG_CLR写1清零。 0: WTD0G未触发复位 1: WTD0G触发复位	0
15: 9			保留	
16	PORFLG_CLR	W	写1操作，产生清零脉冲，清除PORFLG标志位。写0无效。	0
17	LVRFLG_CLR	W	写1操作，产生清零脉冲，清除LVRFLG标志位。写0无效。	0
18	EXTFLG_CLR	W	写1操作，产生清零脉冲，清除EXTFLG标志位。写0无效。	0
19	RTFLG_CLR	W	见11.2章节	0
20	LVDFLG_CLR	W	见5.3.1.4章节	0
23: 21			保留	0
24	WTDFLG_CLR	W	写1操作，产生清零脉冲，清除WTDFLG标志位。写0无效。	0
31: 25			保留	

## 5.6.2. 功能描述

支持 5 种复位方式，包括：上电复位、外部复位、低压复位、看门狗复位、软件复位。上电复位可以保证系统在上电初始时系统可以工作在初始状态。

外部复位可以通过复位管脚来使系统恢复到初始状态，外部复位信号为低电平有效信号，外部复位脉宽必须长于 1us。

低压复位使能后，当电压低于设定的电压值时，系统发生复位。由于这个特性，可以保证应用安全性的同时，为用户省去外部复位电路，当电源电压低于设定电压，芯片内部将处于复位状态，当电源电压高于设定电压时，系统又会重新工作。

看门狗复位，可以保证程序跑飞以后系统可以恢复到初始状况，确保不出现不能挽救的意外发生。如果程序在规定的时段内没有给看门狗定时器执行重载操作，看门狗定时器将会溢出，启动芯片复位操作。

软件复位，可以使程序复位到初始状态。对 AIRCR 寄存器的 SYSRESETREQ 写 1 触发一次软件复位(包括 debug 时)。

上述五种复位机制，可以分为三种类型，一种是硬复位，包括上电复位，低压复位和外部复位，硬复位发生后需要等待  $4 * 65536$  个 SYSCLK 时钟。第二种是软复位，包括看门狗复位和软件复位，软复位发生后系统不会执行等待操作，程序将直接跳转回起始地址，开始执行程序。

复位的状态标志可以通过复位状态寄存器 RESETIFO 和 SYSSTA 来查询相应的复位状态。

## 6. 中断控制器(NVIC)

最多支持 32 个中断向量:

- NMI: WDT 中断
- 0: RT 定时器中断，可以用于唤醒 STOP 模式;
- 1: 晶振时钟异常检测中断
- 2: LVD 低压检测中断，可以用于唤醒 STOP 模式;
- 3: 外部管脚 EINT0 中断，可以用于唤醒 STOP 模式;
- 4: BUART 接收中断
- 5: BUART 发送中断
- 6: 外部管脚 EINT1 中断，可以用于唤醒 STOP 模式;
- 7: 外部管脚 EINT2 中断，可以用于唤醒 STOP 模式;
- 8: Basic Timer 中断;
- 9: 外部管脚 EINT3 中断，可以用于唤醒 STOP 模式;
- 10: TIMER0 定时器匹配中断;
- 11: TIMER0 定时器溢出中断;
- 12: TIMER1 定时器匹配中断;
- 13: TIMER1 定时器溢出中断;
- 14: BUART OVF 中断;
- 15: TIMER2 定时器匹配中断;
- 16: TIMER2 定时器溢出中断;
- 17: EPWM 匹配中断;
- 18: EPWM 溢出中断;
- 19: ACMP0/1/2/3 输出中断;
- 20: ADC 中断;
- 21: I2C 中断;
- 22: SPI 中断;
- 23: UART 中断;
- 24: LIN 同步帧检测中断;
- 25: MATH 除法中断;

- 26: MATH CORDIC 中断;
- 27: QEP 中断;
- 28: PA 中断, PA 的 15 个 IO 中断源或操作后触发;
- 29: PB 中断, PB 的 15 个 IO 中断源或操作后触发;
- 30: ADC 组转换模式时 0、1 通道转换完成中断;
- 31: 保留。

## 6.1. 中断和异常向量表

向量编号	默认优先级	优先级类型	模块	说明	向量地址
-15	-3	固定	Reset	复位	0x00000004
-14	-2	固定	NMI	WDT 溢出中断	0x00000008
-13	-1	固定	HardFault	硬件错误	0x0000000C
-5	0	可配	SVCall	由 SWI 指令发起的服务程序	0x0000002C
-2	1	可配	PendSV		0x00000038
-1	2	可配	SysTick		0x0000003C
0	3	可配	RT		0x00000040
1	4	可配	CSS		0x00000044
2	5	可配	LVD		0x00000048
3	6	可配	EINT0		0x0000004C
4	7	可配	BUARTRX		0x00000050
5	8	可配	BUARTTX		0x00000054
6	9	可配	EINT1		0x00000058
7	10	可配	EINT2		0x0000005C
8	11	可配	BASICTIMER		0x00000060
9	12	可配	EINT3		0x00000064
10	13	可配	TIMER0_CCR		0x00000068
11	14	可配	TIMER0_OVF		0x0000006C
12	15	可配	TIMER1_CCR		0x00000070
13	16	可配	TIMER1_OVF		0x00000074
14	17	可配	BUART_OVF		0x00000078
15	18	可配	TIMER2_CCR		0x0000007C
16	19	可配	TIMER2_OVF		0x00000080
17	20	可配	PWM_CCR		0x00000084
18	21	可配	PWM_OVF		0x00000088
19	22	可配	ACMP		0x0000008C

20	23	可配	ADC		0x0000090
21	24	可配	I2C		0x0000094
22	25	可配	SPI		0x0000098
23	26	可配	UART		0x000009C
24	27	可配	LIN		0x00000A0
25	28	可配	DIV		0x00000A4
26	29	可配	CORDIC		0x00000A8
27	30	可配	QEP		0x00000AC
28	31	可配	PA		0x00000B0
29	32	可配	PB		0x00000B4

## 6.2. 模块寄存器总表

寄存器名	地址	结构体(.h)	寄存器描述	复位值
ISER	0xE000E100	NVIC	设置中断使能0~31, 写1置1, 写0无效。	0x00000000
ICER	0xE000E180	NVIC	清零中断使能0~31, 写1置0, 写0无效。	0x00000000
ISPR	0xE000E200	NVIC	设置中断挂起0~31, 写1置1, 写0无效。	0x00000000
ICPR	0xE000E280	NVIC	清零中断挂起0~31, 写1置0, 写0无效。	0x00000000
IP0	0xE000E400	NVIC	中断0(bit7~6)优先级 中断1(bit15~14)优先级 中断2(bit23~22)优先级 中断3(bit31~30)优先级	0x00000000
IP1	0xE000E404	NVIC	中断7~4优先级	0x00000000
IP2	0xE000E408	NVIC	中断11~8优先级	0x00000000
IP3	0xE000E40C	NVIC	中断15~12优先级	0x00000000
IP4	0xE000E410	NVIC	中断19~16优先级	0x00000000
IP5	0xE000E414	NVIC	中断23~20优先级	0x00000000
IP6	0xE000E418	NVIC	中断27~24优先级	0x00000000
IP7	0xE000E41C	NVIC	中断31~28优先级	0x00000000
PRIMASK				
PAIEN	0x40010020	PA	PAIEN[15: 0]: PA外部中断的使能控制位。	0x0000
PAITYP	0x40010028	PA	PAITYP[15: 0]: PA外部中断的电平触发或边沿触发控制位。	0x0000
PAITYPCLR	0x4001102C			
PAIPOL	0x40010030	PA	PAIPOL[15: 0]: PA外部中断的高/低电平、上升沿/下降沿控制位。	0x0000
PAIPOLCLR	0x40011034			
PAISTA	0x40010038	PA	PAISTA[15: 0]: PA外部中断的标志位。	0x0000
PBIEN	0x40011020	PB	PBIEN[15: 0]: PB外部中断的使能控制位。	0x0000
PBITYP	0x40011028	PB		0x0000

PBITYPCLR	0x4001102C		PBITYP[15: 0]: PB外部中断的电平触发或边沿触发控制位。	
PBIPOL	0x40011030	PB	PBIPOL[15: 0]: PB外部中断的高/低电平、上升沿/下降沿控制位。	0x0000
PBIPOLCLR	0x40011034			
PBISTA	0x40011038	PB	PBISTA[15: 0]: PB外部中断的标志位。	0x0000
EINTCON	0x4000E01C	SYSCFG	EINTCON[31: 0]: IO外部中断控制寄存器(EINT3~0)。	0x00000000
EINTFCON	0x4000E020	SYSCFG	EINTCLR[3: 0]: 写1清除EINT3~0对应标志位。	0x0

## 6.3.NVIC 控制寄存器

### 6.3.1. 中断允许寄存器(ISER)

中断允许寄存器(ISER)用于允许中断设置，同时可返回当前允许中断设置。

中断允许寄存器(ISER, 地址: 0xE000 E100)

位	名称	描述	初始值
31: 0	SETENA	IRQ0~IRQ31 允许中断位。 写: 0: 无效 1: 允许中断 读: 0: 中断禁止 1: 中断允许	0x00000000

如果中断被允许，并且相应中断挂起被设置，NVIC 将会根据中断优先级触发中断。反之，中断被禁止，中断源只会改变中断挂起状态，而NVIC不会对中断源信号采取任何动作，不论中断优先级任何。

### 6.3.2. 中断禁止寄存器(ICER)

中断禁止寄存器(ICER) 用于禁止中断，同时可返回当前允许中断设置。

中断禁止寄存器(ICER, 地址: 0xE000 E180)

位	名称	描述	初始值
31: 0	CLRENA	IRQ0~IRQ31中断禁止位。 写:	0x00000000

位	名称	描述	初始值
		0: 无效 1: 禁止中断 读: 0: 中断禁止 1: 中断允许	

### 6.3.3. 中断挂起寄存器(ISPR)

中断挂起寄存器(ISPR) 强制中断进入挂起状态，同时可返回当前中断挂起设置。

中断挂起寄存器(ISPR, 地址: 0xE000 E200)

位	名称	描述	初始值
31: 0	SETPEND	IRQ0~IRQ31中断挂起位。 写: 0: 无效 1: 强制中断进入挂起状态 读: 0: 没有中断挂起 1: 中断挂起	0x00000000

### 6.3.4. 清除中断挂起寄存器(ICPR)

清除中断挂起寄存器(ICPR)用于清除中断的挂起状态或返回中断的挂起状态。

清除中断挂起寄存器( ICPR, 地址: 0xE000 E280)

位	名称	描述	初始值
31: 0	CLRPEND	中断 IRQ0~IRQ31 清除挂起控制位。 写: 0: 无效 1: 清除该中断挂起状态 读: 0: 该中断不在挂起状态 1: 该中断处于挂起状态	0x00000000

### 6.3.5. 中断优先级寄存器

中断优先级寄存器 IP0-IP7 提供给每个中断(IRQ0~IRQ31)2 位 4 种优先级设置。每一个寄存器包含 4 个中断的优先级。

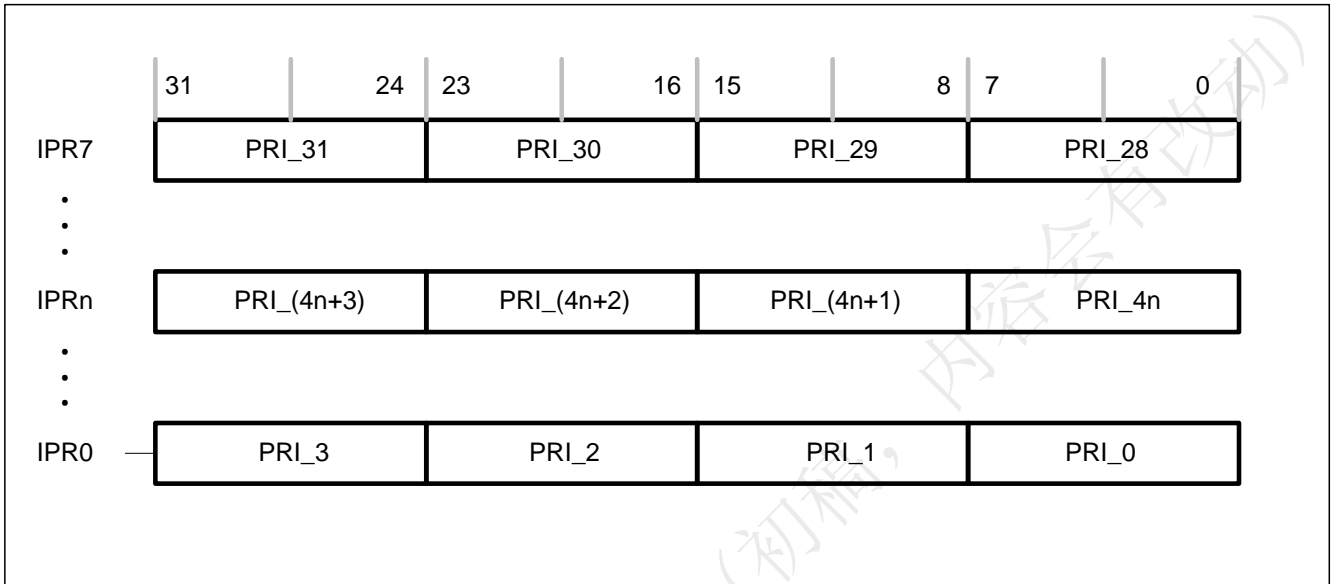


表 6-3-5 时钟切换示意图

IPn 寄存器(IP0~7, 地址: 0xE000-E400~0xE000-E41C)

位	名称	描述	初始值
7: 0	优先级, 字节偏移量0	每个中断的优先级设定放在字节的高两位: 位7和位6。 每个中断的优先级值为: 0-3。越低的值, 表示越高的 优先级。 位5~0 读为0, 写无效。	0x00
15: 8	优先级, 字节偏移量1		0x00
23: 16	优先级, 字节偏移量2		0x00
31: 24	优先级, 字节偏移量3		0x00

用下面的方法计算中断 M 的 IP 寄存器号:

- 计算对应的 IP 寄存器号,  $N, N = M \text{ DIV } 4$
- 计算 IP 寄存器内的字节偏移量  $M \text{ MOD } 4$ , 其中:
  - 字节偏移量 0 对应寄存器的位 7: 0
  - 字节偏移量 1 对应寄存器的位 15: 8
  - 字节偏移量 2 对应寄存器的位 23: 16
  - 字节偏移量 3 对应寄存器的位 31: 24

## 6.4. 外部引脚中断

### 6.4.1. PA 中断使能寄存器(PAIEN/ PAIENCLR: 0x40010020

#### /0x40010024)

描述位	名称	读写	描述	复位值
15: 0	PAIEN [15: 0]	R/W	PAXiEN: PAI/O口作为外部中断输入使能控制位，MCU对PAIEN写1置1，写0由PAIENCLR控制。其中x = 15~0。 0: GPIO输入中断无效 1: GPIO输入中断使能，高电平/低电平/上升沿/下降沿触发中断。	0x0000
15: 0	PAIENCLR [15: 0]	R/W	写1将对应的PAIEN的位清0	0x0000

### 6.4.2. PA 中断类型寄存器(PAITYP/ PAITYPCLR: 0x40010028

#### /0x4001002C)

描述位	名称	读写	描述	复位值
15: 0	PAITYP [15: 0]	R/W	PAXiTYP: PAI/O口作为外部中断输入控制位，MCU对PAITYP写1置1，写0由PAITYPCLR控制。其中x = 15~0。 0: 电平触发中断 1: 沿触发中断	0x0000
15: 0	PAITYPCLR [15: 0]	R/W	写1将对应的PAITYP的位清0	0x0000

### 6.4.3. PA 中断极性控制寄存器(PAIPOL/ PAIPOLCLR:

**0x40010030/0x40010034)**

描述位	名称	读写	描述	复位值
15: 0	PAIPOL [15: 0]	R/W	PAxIPOL: PAI/O口作为外部中断输入控制位, MCU对PAIPOL写1置1, 写0由PAIPOLCLR控制。其中x = 15~0 0: 低电平/下降沿触发中断 1: 高电平/上升沿触发中断	0x0000
15: 0	PAIPOLCLR [15: 0]	R/W	写1将对应的PAIPOL的位清0	0x0000

### 6.4.4. PA 中断状态寄存器(PAISTA: 0x40010038)

描述位	名称	读写	描述	复位值
15: 0	PAISTA [15: 0]	R/W	PAxISTA: PAI/O口作为外部中断的状态位, 其中x = 15~0, 写1清0。 0: 无触发中断 1: 触发中断	0x0000

### 6.4.5. PB 中断使能寄存器(PBIEN/ PBIENCLR:

**0x40011020/0x40011024)**

描述位	名称	读写	描述	复位值
15: 0	PBIEN [15: 0]	R/W	PBxIEN: PBI/O口作为外部中断输入使能控制位, 其中x = 15~0 0: GPIO输入中断无效 1: GPIO输入中断使能, 高电平/低电平/上升沿/下降沿触发中断 MCU对PBIEN写1置1, 写0由PBIENCLR控制。	0x0000
15: 0	PBIENCLR [15: 0]	R/W	写1将对应的PBIEN的位清0	0x0000

### 6.4.6. PB 中断类型寄存器(PBITYP/ PBITYPCLR:

**0x40011028/0x4001102C)**

描述位	名称	读写	描述	复位值
15: 0	PBITYP [15: 0]	R/W	PBxITYP: PB I/O口作为外部中断输入控制位, MCU对PBITYP写1置1, 写0由PBITYPCLR控制。其中x = 15~0。 0: 电平触发中断 1: 沿触发中断	0x0000
15: 0	PBITYPCLR [15: 0]	R/W	写1将对应的PBITYP的位清0	0x0000

### 6.4.7. PB 中断极性控制寄存器(PBIPOL/ PBIPOLCLR:

**0x40011030/0x40011034)**

描述位	名称	读写	描述	复位值
15: 0	PBIPOL [15: 0]	R/W	PBxIPOL: PB I/O口作为外部中断输入控制位, MCU对PBIPOL写1置1, 写0由PBIPOLCLR控制。其中x = 15~0 0: 低电平/下降沿触发中断 1: 高电平/上升沿触发中断	0x0000
15: 0	PBIPOLCLR [15: 0]	R/W	写1将对应的PBIPOL的位清0	0x0000

### 6.4.8. PB 中断状态寄存器(PBISTA: 0x40011038)

描述位	名称	读写	描述	复位值
15: 0	PBISTA [15: 0]	R/W	PBxISTA: PB I/O口作为外部中断的状态位, 其中x = 15~0, 写1清0。 0: 无触发中断 1: 触发中断	0x0000

### 6.4.9. 外部引脚可唤醒中断控制寄存器(EINTCON: 0x4000E01C)

描述位	名称	读写	描述		复位值
2: 0	EINT0_IOPINSEL[2: 0]	R/W	IO组的EINT0_IOGSEL =x0时, 000: 选择IO组中的PIN0 001: 选择IO组中的PIN1 010: 选择IO组中的PIN2 011: 选择IO组中的PIN3 100: 选择IO组中的PIN4 101: 选择IO组中的PIN5 110: 选择IO组中的PIN6 111: 选择IO组中的PIN7	IO组的EINT0_IOGSEL =x1时, 000:选择IO组中的PIN8 001:选择IO组中的PIN9 010:选择IO组中的PIN10 011:选择IO组中的PIN11 100:选择IO组中的PIN12 101:选择IO组中的PIN13 110:选择IO组中的PIN14 111: 选择IO组中的PIN15	000
4: 3	EINT0_IOGSEL[1: 0]	R/W	IO组选择位。 00: 选择PA[7: 0] 01: 选择PA[15: 8] 10: 选择PB[7: 0] 11: 选择PB[15: 8]		00
5	EINT0_FSEL	R/W	数字滤波使能位。 0: 禁止滤波, 外部信号直通, 在deepsleep模式时钟关闭时可由此唤醒时钟 1: 使能滤波, 固定滤4个SYSCLK时间宽度的毛刺		0
7: 6	EINT0_PSEL [1: 0]	R/W	中断触发边沿选择位。 00: 低电平触发 01: 上升沿触发 10: 下降沿触发 11: 上升沿或者下降沿触发		00
10: 8	EINT1_IOPINSEL [2: 0]	R/W	IO组的EINT1_IOGSEL =x0时, 000: 选择IO组中的PIN0 001: 选择IO组中的PIN1 010: 选择IO组中的PIN2 011: 选择IO组中的PIN3 100: 选择IO组中的PIN4 101: 选择IO组中的PIN5 110: 选择IO组中的PIN6 111: 选择IO组中的PIN7	IO组的EINT1_IOGSEL =x1时, 000:选择IO组中的PIN8 001:选择IO组中的PIN9 010:选择IO组中的PIN10 011:选择IO组中的PIN11 100:选择IO组中的PIN12 101:选择IO组中的PIN13 110:选择IO组中的PIN14 111:选择IO组中的PIN15	000
12: 11	EINT1_IOGSEL[1: 0]	R/W	IO组选择位。 00: 选择PA[7: 0]		00

描述位	名称	读写	描述		复位值
			01: 选择PA[15: 8] 10: 选择PB[7: 0] 11: 选择PB[15: 8]		
13	EINT1_FSEL	R/W	数字滤波使能位。 0: 禁止滤波, 外部信号直通, 在deepsleep模式时钟关闭时可由此唤醒时钟 1: 使能滤波, 固定滤4个SYSCLK时间宽度的毛刺		0
15: 14	EINT1_PSEL[1: 0]	R/W	中断触发边沿选择位。 00: 低电平触发 01: 上升沿触发 10: 下降沿触发 11: 上升沿或者下降沿触发		00
18: 16	EINT2_IOPINSEL [2: 0]	R/W	IO组的EINT2_IOGSEL =x0时, 000: 选择IO组中的PIN0 001: 选择IO组中的PIN1 010: 选择IO组中的PIN2 011: 选择IO组中的PIN3 100: 选择IO组中的PIN4 101: 选择IO组中的PIN5 110: 选择IO组中的PIN6 111: 选择IO组中的PIN7	IO组的EINT2_IOGSEL =x1时, 000:选择IO组中的PIN8 001:选择IO组中的PIN9 010:选择IO组中的PIN10 011: 选择IO组中的PIN11 100:选择IO组中的PIN12 101:选择IO组中的PIN13 110:选择IO组中的PIN14 111: 选择IO组中的PIN15	000
20: 19	EINT2_IOGSEL[1: 0]	R/W	IO组选择位。 00: 选择PA[7: 0] 01: 选择PA[15: 8] 10: 选择PB[7: 0] 11: 选择PB[15: 8]		00
21	EINT2_FSEL	R/W	数字滤波使能位。 0: 禁止滤波, 外部信号直通, 在deepsleep模式时钟关闭时可由此唤醒时钟 1: 使能滤波, 固定滤4个SYSCLK时间宽度的毛刺		0
23: 22	EINT2_PSEL[1: 0]	R/W	中断触发边沿选择位。 00: 低电平触发 01: 上升沿触发 10: 下降沿触发 11: 上升沿或者下降沿触发		00
26: 24	EINT3_IOPINSEL [2: 0]	R/W	IO组的EINT3_IOGSEL =x0时,	IO组的EINT3_IOGSEL =x1时,	000

描述位	名称	读写	描述	复位值	
			000: 选择IO组中的PIN0 001: 选择IO组中的PIN1 010: 选择IO组中的PIN2 011: 选择IO组中的PIN3 100: 选择IO组中的PIN4 101: 选择IO组中的PIN5 110: 选择IO组中的PIN6 111: 选择IO组中的PIN7	000:选择IO组中的PIN8 001:选择IO组中的PIN9 010:选择IO组中的PIN10 011: 选择IO组中的PIN11 100:选择IO组中的PIN12 101:选择IO组中的PIN13 110:选择IO组中的PIN14 111:选择IO组中的PIN15	
28: 27	EINT3_I0GSEL[1: 0]	R/W	IO组选择位。 00: 选择PA[7: 0] 01: 选择PA[15: 8] 10: 选择PB[7: 0] 11: 选择PB[11: 8]	00	
29	EINT3_FSEL	R/W	数字滤波使能位。 0: 禁止滤波, 外部信号直通, 在deepsleep模式时钟关闭时可由此唤醒时钟 1: 使能滤波, 固定滤4个SYSCLK时间宽度的毛刺	0	
31: 30	EINT3_PSEL[1: 0]	R/W	中断触发边沿选择位。 00: 低电平触发 01: 上升沿触发 10: 下降沿触发 11: 上升沿或者下降沿触发	00	

#### 6.4.10. 外部引脚可唤醒中断标志位寄存器(EINTFCON:

**0x4000E020)**

描述位	名称	读写	描述	复位值
0	EINT0CLR	W	写1操作, 产生清零脉冲, 清零EINT0_FLAG。写0无效。	
1	EINT1CLR	W	写1操作, 产生清零脉冲, 清零EINT1_FLAG。写0无效。	
2	EINT2CLR	W	写1操作, 产生清零脉冲, 清零EINT2_FLAG。写0无效。	
3	EINT3CLR	W	写1操作, 产生清零脉冲, 清零EINT3_FLAG。写0无效。	

描述位	名称	读写	描述	复位值
4	LVDINTCLR	W	写1操作，产生清零脉冲，清零LVDINT_FLAG。写0无效。	
7: 5			保留	
8	EINT0_FLAG	R	中断事件0标志位，1表示IO中断事件发生	0
9	EINT1_FLAG	R	中断事件1标志位，1表示IO中断事件发生	0
10	EINT2_FLAG	R	中断事件2标志位，1表示IO中断事件发生	0
11	EINT3_FLAG	R	中断事件3标志位，1表示IO中断事件发生	0
12	LVDINT_FLAG	R	LVD事件标志位，1表示LVD中断事件发生	0
15: 13			保留	
16	EINT0_EN	R/W	1表示允许IO中断事件0触发标志位，缺省为0	0
17	EINT1_EN	R/W	1表示允许IO中断事件1触发标志位，缺省为0	0
18	EINT2_EN	R/W	1表示允许IO中断事件2触发标志位，缺省为0	0
19	EINT3_EN	R/W	1表示允许IO中断事件3触发标志位，缺省为0	0
31: 20			保留	

## 7. 通用输入输出端口(GPIO)

共有 2 组可位操作的 I/O 口，最大支持 32 个 IO，每个 IO 都可以根据需要配置成：

- 悬空输入
- 上、下拉输入
- 推挽输出(驱动能力可调、压摆率可调)
- 开漏输出
- 模拟输入输出

其中，PA、PB 分别各有 16 个 IO，各个 IO 根据具体应用也可以配置为相应的复用功能。

## 7.1. 端口结构

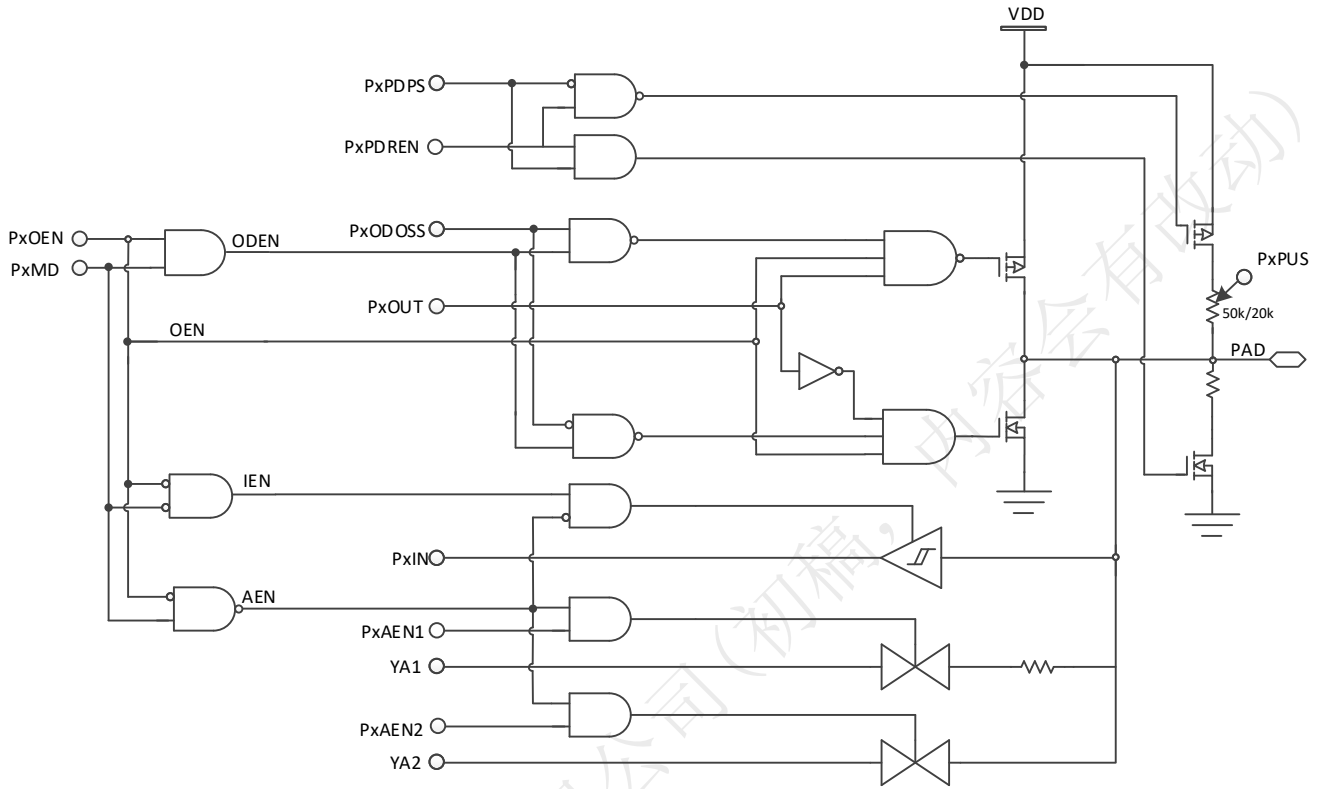


图 7-1 通用 I/O 口结构

## 7.2. 模块寄存器总表

GPIO 控制寄存器列表如下:

寄存器名	地址	结构体(.h)	寄存器描述	特殊说明
PAIN	0x40010000	PA	PA端口管脚输入电平	0x0000
PAOUT	0x40010004	PA	PA端口输出寄存器	0x0000
PA0OUT	0x40010404	PABIT	单独控制PA输出寄存器	0x0000
PA1OUT	0x40010408	PABIT	单独控制PA输出寄存器	0x0000
PA2OUT	0x40010410	PABIT	单独控制PA输出寄存器	0x0000
PA3OUT	0x40010420	PABIT	单独控制PA输出寄存器	0x0000
PA4OUT	0x40010440	PABIT	单独控制PA输出寄存器	0x0000
PA5OUT	0x40010480	PABIT	单独控制PA输出寄存器	0x0000
PA6OUT	0x40010500	PABIT	单独控制PA输出寄存器	0x0000
PA7OUT	0x40010600	PABIT	单独控制PA输出寄存器	0x0000
PA8OUT	0x40010804	PABIT	单独控制PA输出寄存器	0x0000
PA9OUT	0x40010808	PABIT	单独控制PA输出寄存器	0x0000

PA10OUT	0x40010810	PABIT	单独控制PA输出寄存器	0x0000
PA11OUT	0x40010820	PABIT	单独控制PA输出寄存器	0x0000
PA12OUT	0x40010840	PABIT	单独控制PA输出寄存器	0x0000
PA13OUT	0x40010880	PABIT	单独控制PA输出寄存器	0x0000
PA14OUT	0x40010900	PABIT	单独控制PA输出寄存器	0x0000
PA15OUT	0x40010A00	PABIT	单独控制PA输出寄存器	0x0000
PAMD	0x40010008	PA	PA输入输出模式控制位	0x0000
PAPDREN	0x4001000C	PA	PA上下拉控制位	0x0000
PAOEN	0x40010010	PA	PA输入/输出控制位	0x0000
PAOENCLR	0x40010014			
PAALT	0x40010018	PA	PA功能模块IO复用控制位	0x0000
PAALTCLR	0x4001001C			
PAIEN	0x40010020	PA	PA外部中断输入使能控制位	0x0000
PAIENCLR	0x40010024			
PAITYP	0x40010028	PA	PA外部中断输入控制位	0x0000
PAITYPCLR	0x4001002C			
PAIPOL	0x40010030	PA	PA外部中断控制寄存器	0x0000
PAIPOLCLR	0x40010034			
PAISTA	0x40010038	PA	PA外部中断的状态位	0x0000
PBIN	0x40011000	PB	PB端口管脚输入电平	0x0000
PBOUT	0x40011004	PB	PB端口输出寄存器	0x0000
PB0OUT	0x40011404	PBBIT	单独控制PB输出寄存器	0x0000
PB1OUT	0x40011408	PBBIT	单独控制PB输出寄存器	0x0000
PB2OUT	0x40011410	PBBIT	单独控制PB输出寄存器	0x0000
PB3OUT	0x40011420	PBBIT	单独控制PB输出寄存器	0x0000
PB4OUT	0x40011440	PBBIT	单独控制PB输出寄存器	0x0000
PB5OUT	0x40011480	PBBIT	单独控制PB输出寄存器	0x0000
PB6OUT	0x40011500	PBBIT	单独控制PB输出寄存器	0x0000
PB7OUT	0x40011600	PBBIT	单独控制PB输出寄存器	0x0000
PB8OUT	0x40011804	PBBIT	单独控制PB输出寄存器	0x0000
PB9OUT	0x40011808	PBBIT	单独控制PB输出寄存器	0x0000
PB10OUT	0x40011810	PBBIT	单独控制PB输出寄存器	0x0000
PB11OUT	0x40011820	PBBIT	单独控制PB输出寄存器	0x0000
PB12OUT	0x40011840	PBBIT	单独控制PB输出寄存器	0x0000
PB13OUT	0x40011880	PBBIT	单独控制PB输出寄存器	0x0000
PB14OUT	0x40011900	PBBIT	单独控制PB输出寄存器	0x0000
PB15OUT	0x40011A00	PBBIT	单独控制PB输出寄存器	0x0000
PBMD	0x40011008	PB	PB输入输出模式控制位	0x0000
PBPDREN	0x4001100C	PB	PB上下拉控制位	0x0000

PBOEN	0x40011010	PB	PB输入/输出控制位	0x0000
PBOENCLR	0x40011014			
PBALT	0x40011018	PB	PB功能模块IO复用控制位	0x0000
PBALTCLR	0x4001101C			
PBIEN	0x40011020	PB	PB外部中断输入使能控制位	0x0000
PBIENCLR	0x40011024			
PBITYP	0x40011028	PB	PB外部中断输入控制位	0x0000
PBITYPCLR	0x4001102C			
PBIPOL	0x40011030	PB	PB外部中断控制寄存器	0x0000
PBIPOLCLR	0x40011034			
PBISTA	0x40010038	PB	PB外部中断的状态位	0x0000
REMAP0	0x4000E028	IOALT	功能模块IO复用控制位	0x0000
REMAP1	0x4000E02C	IOALT	功能模块IO复用控制位	0x0000
PAFMUX70	0x4000E030	IOALT	功能模块IO复用控制位	0x0000
PAFMUXF8	0x4000E034	IOALT	功能模块IO复用控制位	0x0000
PBFMUX70	0x4000E038	IOALT	功能模块IO复用控制位	0x0000
PBFMUXF8	0x4000E03C	IOALT	功能模块IO复用控制位	0x0000
PAPDRS	0x4000E050	PACFG	PA上下拉选择位	0x0000
PAODOSS	0x4000E054	PACFG	PA开漏、开源输出控制位	0x0000
PACSSR	0x4000E058	PACFG	PA输出压摆率和输入迟滞控制位	0x0000
PADRS	0x4000E05C	PACFG	PA输出驱动能力控制位	0xFFFF
PAAEN1	0x4000E060	PACFG	PA模拟通道1控制位	0x0000
PAAEN2	0x4000E064	PACFG	PA模拟通道2控制位	0x0000
PAPUS	0x4000E068	PACFG	上拉电阻选择	0x0000
PBPDRS	0x4000E070	PBCFG	PB上下拉选择位	0x0000
PBODOSS	0x4000E074	PBCFG	PB开漏、开源输出控制位	0x0000
PBCSSR	0x4000E078	PBCFG	PB输出压摆率和输入迟滞控制位	0x0000
PBDRS	0x4000E07C	PBCFG	PB输出驱动能力控制位	0xFFFF
PBAEN1	0x4000E080	PBCFG	PB模拟通道1控制位	0x0000
PBAEN2	0x4000E084	PBCFG	PB模拟通道2控制位	0x0000
PBPUS	0x4000E088	PBCFG	上拉电阻选择	0x0000

## 7.3. PA 端口寄存器

### 7.3.1. PA 端口输入寄存器(PAIN: 0x40010000)

描述位	名称	读写	描述	复位值
15: 0	PAIN[15: 0]	R	PA端口管脚输入电平	0x0000

### 7.3.2. PA 端口输出寄存器(PAOUT: 0x40010004)

描述位	名称	读写	描述	复位值
15: 0	PAOUT [15: 0]	R/W	PA端口输出寄存器 对PA端口输出寄存器实现写操作。	0x0000

### 7.3.3. PA 端口单独输出寄存器

为了实现 PAOUT [15: 0]的位操作，定义了 2 个位操作地址 0x40010\_b01xx\_xxxx\_xx00(寻址 PAOUT [7: 0])、0x40010\_b10xx\_xxxx\_xx00(寻址 PAOUT [15: 8])。其中 xx\_xxxx\_xx 对 PAOUT [7: 0]、PAOUT [15: 8]进行 bit 寻址，某 x 位为 1 时，可以对该 bit 写操作；某 x 位为 0 时，该 bit 保持不变。

对这 2 个地址读操作时，读取的是 PA[15: 0] 管脚输入电平。

0x40010\_b01xx\_xxxx\_xx00(寻址 PAIN [7: 0])，0x40010\_b10xx\_xxxx\_xx00(寻址 PAIN [15: 8])。某 x 位为 1 时，可以对该 bit 读操作；某 x 位为 0 时，该 bit 读结果为 0。0x40010xxx: 三个 x 对应 12bit，高 11 位和高 10 位为 01 时，对应 PAOUT [7: 0]，第 2-9 位分别对应 PA0-PA7；高 11 位和高 10 位为 10 时，对应 PAOUT [15: 8]，第 2-9 位分别对应 PA8-PA15。

PAOUT [15: 0]支持字节写操作，当总线为字节模式，地址为 0x40010000/0x40010004 时，则只对 PAOUT [7: 0]写操作；

地址为 0x40010001/0x40010005 时，则只对 PAOUT [15: 8]写操作。

#### 7.3.3.1.PA0 端口输出寄存器 PA0OUT: 0x40010404

描述位	名称	读写	描述	复位值
	PA0OUT	R/W	端口PA0输出寄存器。 对PA0OUT的第0位写1，PA0输出高电平； 对PA0OUT的第0位写0，PA0输出低电平。	0x0000

#### 7.3.3.2.PA1 端口输出寄存器 PA1OUT: 0x40010408

描述位	名称	读写	描述	复位值
	PA1OUT	R/W	端口PA1输出寄存器。 对PA1OUT的第1位写1，PA1输出高电平； 对PA1OUT的第1位写0，PA1输出低电平。	0x0000

#### 7.3.3.3.PA2 端口输出寄存器 PA2OUT: 0x40010410

描述位	名称	读写	描述	复位值
	PA2OUT	R/W	端口PA2输出寄存器。 对PA2OUT的第2位写1，PA2输出高电平；	0x0000

描述位	名称	读写	描述	复位值
			对PA2OUT的第2位写0, PA2输出低电平。	

#### 7.3.3.4.PA3 端口输出寄存器 PA3OUT: 0x40010420

描述位	名称	读写	描述	复位值
	PA3OUT	R/W	端口PA3输出寄存器。 对PA3OUT的第3位写1, PA3输出高电平; 对PA3OUT的第3位写0, PA3输出低电平。	0x0000

#### 7.3.3.5.PA4 端口输出寄存器 PA4OUT: 0x40010440

描述位	名称	读写	描述	复位值
	PA4OUT	R/W	端口PA4输出寄存器。 对PA4OUT的第4位写1, PA4输出高电平; 对PA4OUT的第4位写0, PA4输出低电平。	0x0000

#### 7.3.3.6.PA5 端口输出寄存器 PA5OUT: 0x40010480

描述位	名称	读写	描述	复位值
	PA5OUT	R/W	端口PA5输出寄存器。 对PA5OUT的第5位写1, PA5输出高电平; 对PA5OUT的第5位写0, PA5输出低电平。	0x0000

#### PA6端口输出寄存器PA6OUT: 0x40010500

描述位	名称	读写	描述	复位值
	PA6OUT	R/W	端口PA6输出寄存器。 对PA6OUT的第6位写1, PA6输出高电平; 对PA6OUT的第6位写0, PA6输出低电平。	0x0000

#### 7.3.3.7.PA7 端口输出寄存器 PA7OUT: 0x40010600

描述位	名称	读写	描述	复位值
	PA7OUT	R/W	端口PA7输出寄存器。 对PA7OUT的第7位写1, PA7输出高电平; 对PA7OUT的第7位写0, PA7输出低电平。	0x0000

#### 7.3.3.8.PA8 端口输出寄存器 PA8OUT: 0x40010804

描述位	名称	读写	描述	复位值
	PA8OUT	R/W	端口PA8输出寄存器。 对PA8OUT的第8位写1，PA8输出高电平； 对PA8OUT的第8位写0，PA8输出低电平。	0x0000

### 7.3.3.9. PA9 端口输出寄存器 PA9OUT: 0x40010808

描述位	名称	读写	描述	复位值
	PA9OUT	R/W	端口PA9输出寄存器。 对PA9OUT的第9位写1，PA9输出高电平； 对PA9OUT的第9位写0，PA9输出低电平。	0x0000

### 7.3.3.10. PA10 端口输出寄存器 PA10OUT: 0x40010810

描述位	名称	读写	描述	复位值
	PA10OUT	R/W	端口PA10输出寄存器。 对PA10OUT的第10位写1，PA10输出高电 平； 对PA10OUT的第10位写0，PA10输出低电 平。	0x0000

### 7.3.3.11. PA11 端口输出寄存器 PA11OUT: 0x40010820

描述位	名称	读写	描述	复位值
	PA11OUT	R/W	端口PA11输出寄存器。 对PA11OUT的第11位写1，PA11输出高电 平； 对PA11OUT的第11位写0，PA11输出低电 平。	0x0000

### 7.3.3.12. PA12 端口输出寄存器 PA12OUT: 0x40010840

描述位	名称	读写	描述	复位值
	PA12OUT	R/W	端口PA12输出寄存器。 对PA12OUT的第12位写1，PA12输出高电 平；	0x0000

描述位	名称	读写	描述	复位值
			对PA12OUT的第12位写0，PA12输出低电平。	

### 7.3.3.13. PA13 端口输出寄存器 PA13OUT: 0x40010880

描述位	名称	读写	描述	复位值
	PA13OUT	R/W	端口PA13输出寄存器。 对PA13OUT的第13位写1，PA13输出高电平； 对PA13OUT的第13位写0，PA13输出低电平。	0x0000

### 7.3.3.14. PA14 端口输出寄存器 PA14OUT: 0x40010900

描述位	名称	读写	描述	复位值
	PA14OUT	R/W	端口PA14输出寄存器。 对PA14OUT的第14位写1，PA14输出高电平； 对PA14OUT的第14位写0，PA14输出低电平。	0x0000

### 7.3.3.15. PA15 端口输出寄存器 PA15OUT: 0x40010A00

描述位	名称	读写	描述	复位值
	PA15OUT	R/W	端口PA15输出寄存器。 对PA15OUT的第15位写1，PA15输出高电平； 对PA15OUT的第15位写0，PA15输出低电平。	0x0000

## 7.3.4. PA 端口模式寄存器 (PAMD: 0x40010008)

描述位	名称	读写	描述	复位值
15: 0	PAMD[15: 0]	R/W	PA 系列I/O口的模式控制位，其中x = 15~0 当PAxOEN为0时， 0: 作为数字输入口 1: 作为模拟输入输出口	0x0000

描述位	名称	读写	描述	复位值
			当PAxOEN为1时， 0: 作为数字推挽输出 1: 作为数字开源或开漏输出	

### 7.3.5. PA 上下拉电阻使能寄存器(PAPDREN: 0x4001000C)

描述位	名称	读写	描述	复位值
15: 0	PAPDREN [15: 0]	R/W	PAxPDREN: PA I/O口x的上拉/下拉控制位，其中x = 15~0。 0: 禁止上拉、下拉 1: 上下拉使能	0x0000

### 7.3.6. PA 输入输出控制寄存器(PAOEN / PAOENCLR: 0x40010010/0x40010014)

描述位	名称	读写	描述	复位值
15: 0	PAOEN [15: 0]	R/W	PAxOEN: PA I/O口x的输入、输出控制位，其中x=15~0。 0: 输入 1: 输出 MCU对PAOEN写1置1，写0由PAOENCLR控制。	0x0000
15: 0	PAOENCLR [15: 0]	R/W	写1将对应的PAOEN的位清0。	0x0000

### 7.3.7. PA 复用控制寄存器(PAALT/ PAALTCLR: 0x40010018/0x4001001C)

描述位	名称	读写	描述	复位值
15: 0	PAALT [15: 0]	R/W	PAxALT: PA I/O口作为功能模块输入、输出的控制位，MCU对PAALT写1置1，写0由PAALTCLR控制。其中x = 15~0。 0: GPIO输入输出 1: 功能模块输入输出	0x0000

描述位	名称	读写	描述	复位值
15: 0	PAALTCR [15: 0]	R/W	写1将对应的PAALT的位清0	0x0000

### 7.3.8. PA 复用功能选择寄存器 0(PAFMUX70: 0x4000E030)

描述位	名称	读写	描述	复位值
1: 0	PA0ALT	R/W	PA0ALT: PA0复用功能选择。 00: 复用成T1CCR 01: 复用成T0CK 10: 复用成PWM6 11: 复用成SDA	00
3: 2	PA1ALT	R/W	PA1ALT: PA1复用功能选择。 00: 复用成T2CCR0 其它: 保留	00
5: 4	PA2ALT	R/W	PA2ALT: PA2复用功能选择。 01: 复用成QEPA 其它: 保留	00
7: 6	PA3ALT	R/W	PA3ALT: PA3复用功能选择。 00: 复用成T2CCR1 01: 复用成QEPB 其它: 保留	00
9: 8	PA4ALT	R/W	PA4ALT: PA4复用功能选择。 01: 复用成QEPI 其它: 保留	00
11: 10	PA5ALT	R/W	PA5ALT: PA5复用功能选择。 00: 复用成T2CCR2 01: 复用成QEPS 其它: 保留	00
13: 12	PA6ALT	R/W	PA6ALT: PA6复用功能选择, 保留。	00
15: 14	PA7ALT	R/W	PA7ALT: PA7复用功能选择。 00: 复用成T0CCR 01: 复用成EPWM模块的FLT输入 10: 复用成TF模块的TF输入 11: 复用成TD模块TD输入	00

### 7.3.9. PA 复用功能选择寄存器 1(PAFMUXF8: 0x4000E034)

描述位	名称	读写	描述	复位值
1: 0	PA8ALT	R/W	PA8ALT: PA8复用功能选择, 保留	00
3: 2	PA9ALT	R/W	PA9ALT: PA9复用功能选择, 保留	00
5: 4	PA10ALT	R/W	PA10ALT: PA10复用功能选择。 00: 复用成CMP00 其它: 保留	00
7: 6	PA11ALT	R/W	PA11ALT: PA11复用功能选择。 00: 复用成CMP10 其它: 保留	00
9: 8	PA12ALT	R/W	PA12ALT: PA12复用功能选择。 00: 复用成CMP20 其它: 保留	00
11: 10	PA13ALT	R/W	PA13ALT: PA13复用功能选择。 00: 复用成CMP30 01: 复用成ADC模块的ADCSF输出 10: 复用成EPWM模块的FLT输入 11: 复用成Timer2模块的T2CCR0	00
13: 12	PA14ALT	R/W	PA14ALT: PA14复用功能选择。 00: 复用成Timer0模块的T0CCR输入 01: 复用成Timer1模块的T1CCR输入 10: 复用成TF模块的TF输入 11: 复用成TD模块的TD输入	00
15: 14	PA15ALT	R/W	PA15ALT: PA15复用功能选择。 00: 复用成时钟观测输出CLKO 01: 复用成UART模块的TXD 10: 复用成ADC模块的ADCSF输出 11: 保留	00

### 7.3.10. PA 上拉下拉选择寄存器(PAPDRS: 0x4000E050)

描述位	名称	读写	描述	复位值
15: 0	PAPDRS [15: 0]	R/W	PA0~PA15上拉/下拉配置寄存器。 0: 下拉电阻 1: 上拉电阻	0x00

### 7.3.11. PA 开漏开源输出选择寄存器(PAODOSS: 0x4000E054)

描述位	名称	读写	描述	复位值
15: 0	PAODOSS [15: 0]	R/W	PA0~PA15开漏/开源配置寄存器。 0: 开源 1: 开漏	0x00

### 7.3.12. PA 输出压摆率输入迟滞配置寄存器(PACSSR: 0x4000E058)

描述位	名称	读写	描述	复位值
15: 0	PACSSR [15: 0]	R/W	PA0~PA15作为数字输出时，PAxCSSR用于配置压摆率选择。 0: 高压摆率 1: 低压摆率 PA0~PA15作为数字输入时，PAxCSSR用于配置迟滞使能选择。 0: 施密特迟滞输入 1: 无迟滞输入	0x00

### 7.3.13. PA 输出驱动力选择寄存器(PADRS: 0x4000E05C)

描述位	名称	读写	描述	复位值
15: 0	PADRS [15: 0]	R/W	PA0~PA15输出驱动力配置寄存器，缺省为0xFFFF。 0: 大驱动 1: 小驱动	0xFFFF

### 7.3.14. PA 模拟通道 1 配置寄存器(PAAEN1: 0x4000E060)

描述位	名称	读写	描述	复位值
15: 0	PAAEN1 [15: 0]	R/W	PA0~PA15模拟通道1配置寄存器。 0: 模拟通道1关闭 1: 模拟通道1开启，如果该管脚被配置为模拟通道时。	0x0000

### 7.3.15. PA 模拟通道 2 配置寄存器(PAAEN2: 0x4000E064)

描述位	名称	读写	描述	复位值
15: 0	PAAEN2 [15: 0]	R/W	PA0~PA15模拟通道2配置寄存器。 0: 模拟通道2关闭 1: 模拟通道2开启, 如果该管脚被配置为模拟通道时。	0x0000

注意: AEN1、AEN2 可以同时为 1, 这时模拟通道 1、模拟通道 2 和 PAD 连通。

### 7.3.16. PA 上拉电阻选择寄存器(PAPUS: 0x4000E068)

描述位	名称	读写	描述	复位值
15: 0	PAPUS [15: 0]	R/W	PA0~PA15上拉电阻配置寄存器。 0: 上拉电阻50K 1: 上拉电阻20K。	0x0000

## 7.4. PB 端口寄存器

### 7.4.1. PB 端口输入寄存器(PBIN: 0x40011000)

描述位	名称	读写	描述	复位值
15: 0	PBIN[15: 0]	R	PB端口管脚输入电平	0x0000

### 7.4.2. PB 端口输出寄存器(PBOUT: 0x40011004)

描述位	名称	读写	描述	复位值
15: 0	PBOUT [15: 0]	R/W	PB端口输出寄存器 对PB端口输出寄存器实现写操作。	0x0000

### 7.4.3. PB 端口单独输出寄存器

为了实现 PBOUT [15: 0]的位操作，定义了 2 个位操作地址 0x40011\_b01xx\_xxxx\_xx00(寻址 PBOUT [7: 0])、0x40011\_b10xx\_xxxx\_xx00(寻址 PBOUT [15: 8])。其中 xx\_xxxx\_xx 对 PBOUT [7: 0]、PBOUT [15: 8]进行 bit 寻址，某 x 位为 1 时，可以对该 bit 写操作；某 x 位为 0 时，该 bit 保持不变。

对这 2 个地址读操作时，读取的是 PB[15: 0] 管脚输入电平。0x40011\_b01xx\_xxxx\_xx00(寻址 PBIN [7: 0])，0x40011\_b10xx\_xxxx\_xx00(寻址 PBIN [15: 8])。某 x 位为 1 时，可以对该 bit 读操作；某 x 位为 0 时，该 bit 读结果为 0。0x40011xxx: 三个 x 对应 12bit，高 11 位和高 10 位为 01 时，对应 PBOUT [7: 0]，第 2-9 位分别对应 PB0-PB7；高 11 位和高 10 位为 10 时，对应 PBOUT [15: 8]，第 2-9 位分别对应 PB8-PB15。

PBOUT [15: 0]支持字节写操作，当总线为字节模式，地址为 0x40011000/0x40011004 时，则只对 PBOUT [7: 0]写操作；

地址为 0x40011001/0x40011005 时，则只对 PBOUT [15: 8]写操作。

#### 7.4.3.1.PB0 端口输出寄存器 (PB0OUT: 0x40011404)

描述位	名称	读写	描述	复位值
	PB0OUT	R/W	端口PB0输出寄存器。 对PB0OUT的第0位写1，PB0输出高电平 对PB0OUT的第0位写0，PB0输出低电平	0x0000

#### 7.4.3.2.PB1 端口输出寄存器 (PB1OUT: 0x40011408)

描述位	名称	读写	描述	复位值
	PB1OUT	R/W	端口PB1输出寄存器。 对PB1OUT的第1位写1，PB1输出高电平 对PB1OUT的第1位写0，PB1输出低电平	0x0000

#### 7.4.3.3.PB2 端口输出寄存器 (PB2OUT: 0x40011410)

描述位	名称	读写	描述	复位值
	PB2OUT	R/W	端口PB2输出寄存器。 对PB2OUT的第2位写1，PB2输出高电平 对PB2OUT的第2位写0，PB2输出低电平	0x0000

#### 7.4.3.4.PB3 端口输出寄存器 (PB3OUT: 0x40011420)

描述位	名称	读写	描述	复位值
	PB3OUT	R/W	端口PB3输出寄存器。 对PB3OUT的第3位写1，PB3输出高电平 对PB3OUT的第3位写0，PB3输出低电平	0x0000

#### 7.4.3.5.PB4 端口输出寄存器 (PB4OUT: 0x40011440)

描述位	名称	读写	描述	复位值
	PB4OUT	R/W	端口PB4输出寄存器。 对PB4OUT的第4位写1，PB4输出高电平 对PB4OUT的第4位写0，PB4输出低电平	0x0000

#### 7.4.3.6.PB5 端口输出寄存器 (PB5OUT: 0x40011480)

描述位	名称	读写	描述	复位值
	PB5OUT	R/W	端口PB5输出寄存器。 对PB5OUT的第5位写1，PB5输出高电平 对PB5OUT的第5位写0，PB5输出低电平	0x0000

#### 7.4.3.7.PB6 端口输出寄存器 (PB6OUT: 0x40011500)

描述位	名称	读写	描述	复位值
	PB6OUT	R/W	端口PB6输出寄存器。 对PB6OUT的第6位写1，PB6输出高电平 对PB6OUT的第6位写0，PB6输出低电平	0x0000

#### 7.4.3.8.PB7 端口输出寄存器 (PB7OUT: 0x40011600)

描述位	名称	读写	描述	复位值
	PB7OUT	R/W	端口PB7输出寄存器。 对PB7OUT的第7位写1，PB7输出高电平 对PB7OUT的第7位写0，PB7输出低电平	0x0000

#### 7.4.3.9.PB8 端口输出寄存器 (PB8OUT: 0x40011804)

描述位	名称	读写	描述	复位值
	PB8OUT	R/W	端口PB8输出寄存器。 对PB8OUT的第8位写1，PB8输出高电平 对PB8OUT的第8位写0，PB8输出低电平	0x0000

#### 7.4.3.10. PB9 端口输出寄存器 (PB9OUT: 0x40011808)

描述位	名称	读写	描述	复位值
	PB9OUT	R/W	端口PB9输出寄存器。 对PB9OUT的第9位写1, PB9输出高电平 对PB9OUT的第9位写0, PB9输出低电平	0x0000

#### 7.4.3.11. PB10 端口输出寄存器(PB10OUT: 0x40011810)

描述位	名称	读写	描述	复位值
	PB10OUT	R/W	端口PB10输出寄存器。 对PB10OUT的第10位写1, PB10输出高电平 对PB10OUT的第10位写0, PB10输出低电平	0x0000

#### 7.4.3.12. PB11 端口输出寄存器(PB11OUT: 0x40011820)

描述位	名称	读写	描述	复位值
	PB11OUT	R/W	端口PB11输出寄存器。 对PB11OUT的第11位写1, PB11输出高电平 对PB11OUT的第11位写0, PB11输出低电平	0x0000

#### 7.4.3.13. PB12 端口输出寄存器(PB12OUT: 0x40011840)

描述位	名称	读写	描述	复位值
	PB12OUT	R/W	端口PB12输出寄存器。 对PB12OUT的第12位写1, PB12输出高电平 对PB12OUT的第12位写0, PB12输出低电平	0x0000

#### 7.4.3.14. PB13 端口输出寄存器(PB13OUT: 0x40011880)

描述位	名称	读写	描述	复位值
	PB13OUT	R/W	端口PB13输出寄存器。 对PB13OUT的第13位写1, PB13输出高电平 对PB13OUT的第13位写0, PB13输出低电平	0x0000

#### 7.4.3.15. PB14 端口输出寄存器(PB14OUT: 0x40011900)

描述位	名称	读写	描述	复位值
	PB14OUT	R/W	端口PB14输出寄存器。 对PB14OUT的第14位写1, PB14输出高电平 对PB14OUT的第14位写0, PB14输出低电平	0x0000

#### 7.4.3.16. PB15 端口输出寄存器(PB15OUT: 0x40011A00)

描述位	名称	读写	描述	复位值
	PB15OUT	R/W	端口PB15输出寄存器。 对PB15OUT的第15位写1，PB15输出高电平 对PB15OUT的第15位写0，PB15输出低电平	0x0000

#### 7.4.4. PB 端口模式寄存器 (PBMD: 0x40011008)

描述位	名称	读写	描述	复位值
15: 0	PBMD[15: 0]	R/W	PB I/O口的模式控制位，其中x = 15~0。 当PBxOEN为0时 0: 作为数字输入口 1: 作为模拟输入输出 当PBxOEN为1时 0: 作为数字推挽输出 1: 作为数字开源或开漏输出	0x0000

#### 7.4.5. PB 上下拉电阻使能寄存器(PBPDREN: 0x4001100C)

描述位	名称	读写	描述	复位值
15: 0	PBPDREN [15: 0]	R/W	PBxPDREN: PB I/O口x的上下拉控制位，其中x = 15~0。 0: 禁止上拉、下拉 1: 上下拉使能	0x0000

#### 7.4.6. PB 输入输出控制寄存器(PBOEN/ PBOENCLR: 0x40011010/0x40011014)

描述位	名称	读写	描述	复位值
15: 0	PBOEN [15: 0]	R/W	PBxOEN: PB I/O口x的输入、输出控制位，其中x = 15~0。 0: 输入 1: 输出 MCU对PBOEN写1置1，写0由PBOENCLR控制。	0x0000
15: 0	PBOENCLR [15: 0]	R/W	写1将对应的PAIPOL的位清0	0x0000

### 7.4.7. PB 复用控制寄存器(PBALT/ PBALTCLR:

#### 0x40011018/0x4001101C)

描述位	名称	读写	描述	复位值
15: 0	PBALT [15: 0]	R/W	PBxALT: PB I/O口作为功能模块输入、输出的控制位，其中x = 15~0。 0: GPIO输入输出 1: 功能模块输入输出 MCU对PBALT写1置1，写0由PBALTCLR控制。	0x0000
15: 0	PBALTCLR [15: 0]	R/W	写1将对应的PBALT的位清0	0x0000

### 7.4.8. PB 复用功能选择寄存器 0(PBFMUX70: 0x4000E038)

描述位	名称	读写	描述	复位值
1: 0	PB0ALT	R/W	PB0ALT: PB0复用功能选择。 00: 复用成EPWM模块的PWM0 其它: 保留	00
3: 2	PB1ALT	R/W	PB1ALT: PB1复用功能选择。 00: 复用成EPWM模块的PWM1 其它: 保留	00
5: 4	PB2ALT	R/W	PB2ALT: PB2复用功能选择。 00: 复用成EPWM模块的PWM2 其它: 保留	00
7: 6	PB3ALT	R/W	PB3ALT: PB3复用功能选择。 00: 复用成EPWM模块的PWM3 其它: 保留	00
9: 8	PB4ALT	R/W	PB4ALT: PB4复用功能选择。 00: 复用成EPWM模块的PWM4 其它: 保留	00
11: 10	PB5ALT	R/W	PB5ALT: PB5复用功能选择。 00: 复用成EPWM模块的PWM5 其它: 保留	00
13: 12	PB6ALT	R/W	PB6ALT: PB6复用功能选择。 00: 复用成BUART的TXD 01: 复用成UART的TXD 10: 复用成I2C模块的SCL	00

描述位	名称	读写	描述	复位值
			11: 复用成SPI模块的SCLK	
15: 14	PB7ALT	R/W	PB7ALT: PB7复用功能选择。 00: 复用成BUART的RXD 01: 复用成UART的RXD 10: 复用成I2C模块的SDA 11: 复用成SPI模块的MOSI	00

#### 7.4.9. PB 复用功能选择寄存器 1(PBFMUXF8: 0x4000E03C)

描述位	名称	读写	描述	复位值
1: 0	PB8ALT	R/W	PB8ALT: PB8复用功能选择。 00: 复用成Timer0的外部时钟输入T0CK 01: 复用成UART的TXD 10: 复用成Timer2模块的T2CCR1 11: 复用成SPI模块的MISO	00
3: 2	PB9ALT	R/W	PB9ALT: PB9复用功能选择。 00: 复用成Timer1的T1CCR 01: 复用成UART的RXD 10: 复用成SPI模块的从机NSS 11: 复用成Timer2模块的T2CCR2	00
5: 4	PB10ALT	R/W	PB10ALT: PB10复用功能选择。 00: 复用成CMP00 01: 复用成Timer0的T0CCR 10: 复用成EPWM模块的FLT输入 11: 复用成SPI模块的SCLK	00
7: 6	PB11ALT	R/W	PB11ALT: PB11复用功能选择。 00: 复用成CMP10 01: 复用成CMP00 10: 复用成BUART模块的TXD 11: 复用成SPI模块的MOSI	00
9: 8	PB12ALT	R/W	PB12ALT: PB12复用功能选择。 00: 复用成CMP20 01: 复用成CMP30 10: 复用成BUART模块的RXD 11: 复用成SPI模块的MISO	00
11: 10	PB13ALT	R/W	PB13ALT: PB13复用功能选择。 00: 复用成CMP30	00

描述位	名称	读写	描述	复位值
			01: 复用成Timer1的T1CCR 10: 复用成I2C模块的SCL 11: 复用成SPI模块的从机NSS	
13: 12	PB14ALT	R/W	PB14ALT: PB14复用功能选择。 00: 复用成EPWM模块的PWM6输出 01: 复用成Timer0的外部时钟输入T0CK 10: 复用成ADC模块的ADCSF输出 11: 复用成EPWM模块的FLT输入	00
15: 14	PB15ALT	R/W	PB15ALT: PB15复用功能选择。 00: 复用成Timer0模块的T0CCR 01: 复用成UART的RXD 10: 复用成TF模块的TF输入 11: 复用成TD模块TD输入	00

#### 7.4.10. PB 上拉下拉选择寄存器(PBPDRS: 0x4000E070)

描述位	名称	读写	描述	复位值
15: 0	PBPDRS [15: 0]	R/W	PB0~PB15上拉/下拉配置寄存器。 0: 下拉电阻 1: 上拉电阻	0x0000

#### 7.4.11. PB 开漏开源配置寄存器(PBODOSS: 0x4000E074)

描述位	名称	读写	描述	复位值
15: 0	PBODOSS [15: 0]	R/W	PB0~PB15开漏/开源配置寄存器。 0: 开源 1: 开漏	0x0000

#### 7.4.12. PB 输出压摆率输入迟滞配置寄存器(PBCSSR: 0x4000E078)

描述位	名称	读写	描述	复位值
15: 0	PBCSSR [15: 0]	R/W	PB0~PB15作为数字输出时, PBxCSSR用于配置压摆率选择。 0: 高压摆率 1: 低压摆率 PB0~PB15作为数字输入时, PBxCSSR用于配置迟滞选择。 0: 施密特迟滞输入	0x0000

描述位	名称	读写	描述	复位值
			1: 无迟滞输入	

#### 7.4.13. PB 输出驱动力配置寄存器(PBDRS: 0x4000E07C)

描述位	名称	读写	描述	复位值
15:0	PBDRS [15:0]	R/W	PB0~PB15驱动力配置寄存器。 0: 大驱动 1: 小驱动	0xFFFF

#### 7.4.14. PB 模拟通道 1 配置寄存器(PBAEN1: 0x4000E080)

描述位	名称	读写	描述	复位值
15:0	PBAEN1 [15:0]	R/W	PB0~PB15模拟通道1配置寄存器。 0: 模拟通道1关闭 1: 模拟通道1开启, 如果该管脚被配置为模拟通道时。	0x0000

#### 7.4.15. PB 模拟通道 2 配置寄存器(PBAEN2: 0x4000E084)

描述位	名称	读写	描述	复位值
15:0	PBAEN2 [15:0]	R/W	PB0~PB15模拟通道2配置寄存器。 0: 模拟通道2关闭 1: 模拟通道2开启, 如果该管脚被配置为模拟通道时。	0x0000

注意: AEN1、AEN2 可以同时为 1, 这时模拟通道 1、模拟通道 2 和 PAD 连通。

#### 7.4.16. PB 上拉电阻选择寄存器(PBPUS: 0x4000E088)

描述位	名称	读写	描述	复位值
15:0	PBPUS [15:0]	R/W	PB0~PB15上拉电阻配置寄存器。 0: 上拉电阻50K 1: 上拉电阻20K。	0x0000

## 7.5. IO 功能映射寄存器

### 7.5.1. IO 映射寄存器 0(REMAP0: 0x4000E028)

描述位	名称	读写	描述	复位值
1: 0	T0CCALT	R/W	Timer0的T0CCR映射。 00: 映射到PA7 01: 映射到PA14 10: 映射到PB10 11: 映射到PB15	00
3: 2	T1CCALT	R/W	Timer1的T1CCR映射。 00: 映射到PA0 01: 映射到PA14 10: 映射到PB9 11: 映射到PB13	00
5: 4	T2CC0ALT	R/W	Timer2的T2CCR0映射。 00: 映射到PA1 01: 映射到PA13 10: 保留 11: 映射到CMP00	00
7: 6	T2CC1ALT	R/W	Timer2的T2CCR1映射。 00: 映射到PA3 01: 映射到PB8 10: 保留 11: 映射到CMP10	00
9: 8	T2CC2ALT	R/W	Timer2的T2CCR2映射。 00: 映射到PA5 01: 映射到PB9 10: 保留 11: 映射到CMP20	00
11: 10	TFALT	R/W	TF的频率捕获输入映射。 00: 映射到PA7 01: 映射到PA14 10: 映射到PB15 11: 保留	00
13: 12	TDALT	R/W	TD的占空比捕获输入映射。 00: 映射到PA7	00

描述位	名称	读写	描述	复位值
			01: 映射到PA14 10: 映射到PB15 11: 保留	
15: 14	T0CKALT	R/W	Timer0的外部时钟输入T0CK映射。 00: 映射到PA0 01: 映射到PB8 10: 映射到PB14 11: 保留	00

### 7.5.2. I0 映射寄存器 1(REMAP1: 0x4000E02C)

描述位	名称	读写	描述	复位值
2: 0	FLTALT	R/W	PWM的FLT映射关系。 000: 映射到PA7 001: 映射到PA13 010: 映射到PB10 011: 映射到PB14 其它: CMP3O	000
4: 3	PWM6ALT	R/W	EPWM模块的PWM6映射关系。 00: PWM6映射到PA0 01: PWM6映射到PB14 10: 保留 11: 保留	00
6: 5	PWMALT		EPWM模块的PWM0~PWM5映射关系。 00: PWM0映射到PB0, PWM1映射到PB1, PWM2映射到PB2, PWM3映射到PB3, PWM4映射到PB4, PWM5映射到PB5 其它: 保留	00
8: 7	I2CALT	R/W	I2C的SCL/SDA映射关系。 00: SCL映射到PB6, SDA映射到PB7 01: SCL映射到PB13, SDA映射到PA0 1X: 保留	00
10: 9	UARTALT	R/W	UART的TXD/RXD映射关系。 00: TXD映射到PB6, RXD映射到PB7	00

描述位	名称	读写	描述	复位值
			01: TXD映射到PB8, RXD映射到PB9 10: TXD映射到PA15, RXD映射到PB15 11: 保留	
12: 11	SPIALT	R/W	SPI的SCLK/MOSI/MISO映射关系。 00: SCLK映射到PB6, MOSI映射到PB7, MISO映射到PB8, NSS映射到PB9 01: SCLK映射到PB10, MOSI映射到PB11, MISO映射到PB12, NSS映射到PB13 1X: 保留	00
14: 13	BUARTALT	R/W	BUART的TXD/RXD映射关系。 00: TXD映射到PB6, RXD映射到PB7 01: TXD映射到PB11, RXD映射到PB12 1X: 保留	00
15			保留	

## 7.6. 功能描述

内部包括 2 组 I/O 口，最大支持 32 个管脚。其中，PA 口支持 16 个 I/O，PB 口支持 16 个 I/O。

### 7.6.1. I0 模式设置 (PxALT= 0)

通过设置模式寄存器 PxMD、PxOEN，每个 I/O 口都可以配置成 4 种输入输出模式：数字输入、模拟输入输出、数字开漏输出、数字推挽输出。

通过设置上拉寄存器 PxPEN，每个 I/O 口都可以使能/禁止内部上拉或下拉电阻。

模式寄存器和上拉寄存器的设置方法如下：(PxALT= 0，非复用功能)

序号	寄存器 PxMD、PxOEN	输入输出模式	寄存器位 PxEN 设置	上拉/下拉电阻	模式
1	PxMD = 0 PxOEN=0	数字输入	PxPEN = 0	禁止	悬空输入
			PxPEN = 1	使能	上拉/下拉输入
2	PxMD = 1 PxOEN=0	模拟输入输出	PxPEN = 0	禁止	模拟输入输出
3	PxMD = 1 PxOEN=1	开漏/开源输出	PxPEN = 0	禁止	无上拉/下拉，开漏/开源输出

			PxPEN = 1	使能	上拉/下拉，开漏/开源输出
4	PxMD = 0 PxOEN=1	数字推挽输出	PxPEN = 0	禁止	数字推挽输出
				禁止	数字推挽输出

注 1: 模拟输入输出模式下，必须禁止相应 I/O 口的上拉/下拉电阻，对应 PxALT 必须设置为 00。

数字推挽输出模式下，必须禁止相应 I/O 口的上拉/下拉电阻。

注 2: 特殊情况 1，复用功能配置为 SPI、I2C、UART 等时，输入输出模式与 PxMD、PxOEN 无关，硬件会自动将 IO 口配置为输入或输出状态。

注 3: 特殊情况 2，复用功能配置为 PWM00~PWM06 时，PxMD、PxOEN 可以配置为输入模式。此时由 OUTCFG 中 PWMxOEN 决定 PWM 通道为捕获输入或 PWM 输出，当 PWMxOEN 位为高电平时，表示 PWM 输出；PWMxOEN 位为低电平时，表示 PWM 捕获输入。

## 7.6.2. IO 数字功能复用

### 7.6.2.1. IO 管脚对应数字复用功能

IO 口	数字功能复用			
	复用 0	复用 1	复用 2	复用 3
PA5	T2CCR2			
PA7	T0CCR	FLT	TFIN	TDIN
PA10	CP00			
PA11	CP10			
PA12	CP20			
PA13	CP30	ADCSF	FLT	T2CCR0
PA14	T0CCR	T1CCR	TFIN	TDIN
PA15	CKO	TXD1	ADCSF	
PB0	PWM0			
PB1	PWM1			
PB2	PWM2			
PB3	PWM3			
PB4	PWM4			
PB5	PWM5			
PB6	TXD0	TXD1	SCL	SCLK
PB7	RXD0	RXD1	SDA	MOSI

PB8	T0CK	TXD1	T2CCR1	MISO
PB9	T1CCR	RXD1	T2CCR2	NSS
PB10	CP0O	T0CCR	FLT	SCLK
PB11	CP1O	CP0O	TXD0	MOSI
PB12	CP2O	CP3O	RXD0	MISO
PB13	CP3O	T1CCR	SCL	NSS
PB14	PWM6	T0CK	ADCSF	FLT
PB15	T0CCR	RXD1	TFIN	TDIN
PA0	T1CCR	T0CK	PWM6	SDA
PA1	T2CCR0			
PA3	T2CCR1			

#### 7.6.2.2. 定时器 PWM 输出 IO 配置示例

```

PA->MD_b.bit7 = 0;
PA->OEN_b.bit7 = 1;
PA->ALT_b.bit7 = 1;
IOALT->REMAP0_b.T0CCALT = 0;
IOALT->PAFMUX70_b.PA7ALT = 0;
    
```

#### 7.6.2.3. 定时器捕获 IO 配置示例

```

PB->MD_b.bit15 = 0;
PB->OEN_b.bit15 = 0;
PB->ALT_b.bit15 = 1;
IOALT->REMAP0_b.T0CCALT = 3;
IOALT->PBFMUXF8_b.PB15ALT = 0;
    
```

#### 7.6.2.4. EPWM 模块故障检测 FLT 信号 IO 配置示例

```

PA->MD_b.bit7 = 0;
PA->OEN_b.bit7 = 1;
PA->ALT_b.bit7 = 1;
IOALT->REMAP1_b.FLTALT = 0;
IOALT->PAFMUX70_b.PA7ALT = 1;
    
```

#### 7.6.2.5. EPWM 模块 PWM 输出 IO 配置示例

```
PA->MD_b.bit0 = 0;  
PA->OEN_b.bit0 = 1;  
PA->ALT_b.bit0 = 1;  
IOALT->REMAP1_b.PWM6ALT = 0;  
IOALT->PAFMUX70_b.PA0ALT = 2;
```

#### 7.6.2.6. TF 模块对应 IO 配置示例

```
PA->MD_b.bit7 = 0;  
PA->ALT_b.bit7 = 1;  
IOALT->PAFMUX70_b.PA7ALT = 2;  
IOALT->REMAP0_b.TFALT = 0;
```

#### 7.6.2.7. TD 模块对应 IO 配置示例

```
PB->MD_b.bit15 = 0;  
PB->ALT_b.bit15 = 1;  
IOALT->PBFMUXF8_b.PB15ALT = 3;  
IOALT->REMAP0_b.TDALT = 2;
```

#### 7.6.2.8. 内部时钟输出 IO 配置示例

```
PA->MD_b.bit15 = 0;  
PA->OEN_b.bit15 = 1;  
PA->ALT_b.bit15 = 1;  
IOALT->PAFMUXF8_b.PA15ALT = 0;
```

#### 7.6.2.9. UART 对应 IO 配置示例

```
PB->ALT_b.bit6 = 1;  
PB->ALT_b.bit7 = 1;  
  
IOALT->REMAP1_b.UARTALT = 0;  
IOALT->PBFMUX70_b.PB6ALT = 1;  
IOALT->PBFMUX70_b.PB7ALT = 1;
```

#### 7.6.2.10. SPI 对应 IO 配置示例

```
PB->ALT_b.bit6 = 1;
```

```
PB->ALT_b.bit7 = 1;  
PB->ALT_b.bit8 = 1;  
PB->ALT_b.bit9 = 1;
```

```
IOALT->REMAP1_b.SPIALT = 0;  
IOALT->PBFMUX70_b.PB6ALT = 3;  
IOALT->PBFMUX70_b.PB7ALT = 3;  
IOALT->PBFMUXF8_b.PB8ALT = 3;  
IOALT->PBFMUXF8_b.PB9ALT = 2;
```

#### 7.6.2.11. IIC 对应 IO 配置示例

```
PBCFG->PDRS_b.bit6 = 1;  
PBCFG->ODOSS_b.bit6 = 1;  
PB->ALT_b.bit6 = 1;
```

```
PBCFG->PDRS_b.bit7 = 1;  
PBCFG->ODOSS_b.bit7 = 1;  
PB->ALT_b.bit7 = 1;
```

```
IOALT->REMAP1_b.I2CALT = 0;  
IOALT->PBFMUX70_b.PB6ALT = 2;  
IOALT->PBFMUX70_b.PB7ALT = 2;
```

#### 7.6.2.12. 比较器输出对应 IO 配置示例

```
PA->MD_b.bit10 = 0;  
PA->OEN_b.bit10 = 1;  
PA->ALT_b.bit10 = 1;  
IOALT->PAFMUXF8_b.PA10ALT = 0x00;
```

#### 7.6.2.13. ADCSF 输出 IO 配置示例

```
PA->MD_b.bit15 = 0;  
PA->OEN_b.bit15 = 1;  
PA->ALT_b.bit15 = 0;  
IOALT->PAFMUXF8.PA15ALT = 2;
```

### 7.6.3. IO 模拟功能复用

#### 7.6.3.1. IO 管脚对应模拟复用功能

IO 模拟功能复用		
IO 名称	AEN1 通道	AEN2 通道
PA4	CP2N	AIN4
PA5	CP2P	AIN5
PA6	CP3P	AIN6
PA7	OP0N	AIN7 CP3N
PA8	OP0P	AIN8
PA9	OP1N	AIN9
PA10	OP1P	AIN10
PA11	OP2N	AIN11
PA12	OP2P	AIN12
PA15	DACO VRH OP0O OP1O	AIN12
PB6		XOUT
PB7		XIN

PB15	OP00 OP20	CP3N AIN7
PA0	CP0N	AIN0
PA1	CP0P	AIN1
PA2	CP1N	AIN2
PA3	CP1P	AIN3

模拟通道的寄存器设置方式

配置端口用作 ADC、DAC、OPA、CMP、时钟输出、晶振接口时：

1. 配置寄存器 PxOEN，设置 I/O 口的输入输出状态
2. 配置寄存器 PxMD，设置 I/O 口为模拟输入输出状态
3. 配置寄存器 PxCFG->AENx，设置对应的模拟通道；

### 7.6.3.2.ADC 输入 IO 配置示例

```
PA->MD_b.bit12 = 1;
PA->OENCLR_b.bit12 = 1;
PACFG->AEN2_b.bit12 = 1;
```

### 7.6.3.3.DAC 输出 IO 配置示例

```
PA->MD_b.bit15 = 1;
PA->OENCLR_b.bit15 = 1;
PACFG->AEN1_b.bit15 = 1;
```

### 7.6.3.1.VREF 电压输出 IO 配置示例

```
PA->MD_b.bit15 = 1;
PA->OENCLR_b.bit15 = 1;
PACFG->AEN1_b.bit15 = 1;
```

### 7.6.3.2.晶振输入 IO 配置示例

```
PB->MD_b.bit6 = 1;
PB->MD_b.bit7 = 1;

PB->OENCLR_b.bit6 = 1;
PB->OENCLR_b.bit7 = 1;

PBCFG->AEN2_b.bit6 = 1;
PBCFG->AEN2_b.bit7 = 1;
```

### 7.6.3.3.比较器输入 IO 配置示例

PA->MD\_b.bit0 = 1;  
PA->OENCLR\_b.bit0 = 1;  
PACFG->AEN1\_b.bit0 = 1;

### 7.6.3.4.运算放大器输入 IO 配置示例

PA->MD\_b.bit8 = 1;  
PA->OENCLR\_b.bit8 = 1;  
PACFG->AEN1\_b.bit8 = 1;

### 7.6.3.5.运算放大器输出 IO 配置示例

PA->MD\_b.bit15 = 1;  
PA->OEN\_b.bit15 = 1;  
PACFG->AEN1\_b.bit15 = 1;

## 8. 定时器(BASICTIMER)

定时器 0 为 32 位定时器，提供基本定时功能，控制寄存器的主要寄存器列表如下：

### 8.1. 模块寄存器总表

寄存器名	地址	结构体(.h)	寄存器描述	复位值
BTCTRL	0x40000000	BTIM	定时器控制寄存器	0x00
BTCURR	0x40000004	BTIM	定时器计数寄存器	0x00000000
BTRELOAD	0x40000008	BTIM	定时器RELOAD寄存器	0x00000000
BTINTCLR	0x4000000C	BTIM	定时器中断清0寄存器	0x00

### 8.2. 寄存器说明

#### 8.2.1. BasicTimer 控制寄存器 (BTCTRL: 0x40000000)

描述位	名称	读写	描述	复位值
0	BTEN	R/W	计数器使能位。 0: 禁止，计数器不工作 1: 允许，计数时钟触发计数器减1	0

描述位	名称	读写	描述	复位值
2: 1			保留	
3	BTITEN	R/W	中断屏蔽位。 0: 禁止中断 1: 允许中断	0
7: 4			保留	

### 8.2.2. BasicTimer 计数寄存器 (BTCURR: 0x40000004)

描述位	名称	读写	描述	复位值
31: 0	BTCURR[31: 0]	R/W	32位计数器当前计数值。	0x00000000

### 8.2.3. BasicTimer 重载寄存器 (BTRELOAD: 0x40000008)

描述位	名称	读写	描述	复位值
31: 0	BTRELOAD [31: 0]	R/W	当计数器减1到0时，重新从RELOAD_VAL载入初始值。	0x00000000

### 8.2.4. BasicTimer 中断清零寄存器 (BTINTCLR: 0x4000000C)

描述位	名称	读写	描述	复位值
0	INTCLR	R/W	写1清零中断标志。	0
7: 1			保留	

## 9. 定时器(TIMER0/1)

由 2 个独立的 16 位定时器 T0、T1 组成，具有以下特性：

- 可编程的预分频器
- 可独立实现 16 位定时模式
- 可独立实现 16 位比较输出模式(PWM 模式)
- 可独立实现 16 位捕获输入模式
- T0、T1 可配置为级联模式
- 计数时钟可以由系统外设时钟(PCLK)分频时钟输入，或外部信号(T0)输入
- 捕获可以由外部事件上升沿、下降沿、上升/下降沿触发
- T0 产生 2 个中断源(2 个中断向量)，中断标志软件清零

- T1 产生 2 个中断源(2 个中断向量), 中断标志软件清零
- T0 支持 N(1~256)周期溢出中断
- T1 只支持单周期溢出中断

## 9.1. 模块寄存器总表

定时器 T0、T1 控制寄存器的主要寄存器列表如下:

寄存器名	地址	结构体(.h)	寄存器描述	复位值
T0CON	0x40000800	TIM0	定时器0控制寄存器	0x00000000
T0PER	0x40000804	TIM0	定时器0周期寄存器	0xFFFF
T0CNT	0x40000808	TIM0	定时器0计数器	0x0000
T0CCR	0x4000080C	TIM0	定时器0捕获/比较寄存器	0x0000
T1CON	0x40001000	TIM1	定时器1控制寄存器	0x0000
T1PER	0x40001004	TIM1	定时器1周期寄存器	0xFFFF
T1CNT	0x40001008	TIM1	定时器1计数器	0x0000
T1CCR	0x4000100C	TIM1	定时器1捕获/比较寄存器	0x0000

## 9.2. 寄存器说明

### 9.2.1. 定时器 T0 控制寄存器(T0CON: 0x40000800)

描述位	名称	读写	描述	复位值
0	T0EN	R/W	定时器T0使能位。 0: 定时器关闭 1: 定时器使能, T0EN由0→1过程, 计数器值会被复位清零。	0
1	OVFCLREN	R/W	T0周期溢出时清零使能位, 该位只有捕获模式且CAPCLREN = 1时有意义。 0: 周期溢出时计数器不清零。即计数器暂停计数, 直到发生新的捕获事件, 计数器将重新从0开始计数。 1: 周期溢出时计数器立即清零, 并重新从0开始计数。	0
2	CAPCLREN	R/W	T0捕获清零使能位, 该位只有捕获模式有意义。 0: 捕获发生时计数器不清零。计数器将继续向上计数, 直到周期溢出时, 计数器重新从0开始计数(不论OVFCLREN等于0或1)。 1: 捕获发生时计数器清零, 计数器将重新从0开始计数。	0

描述位	名称	读写	描述	复位值
4: 3	MODE[1: 0]	R/W	T0模式控制位。 00: 定时模式 01: 捕获输入模式 10: 比较输出模式(PWM模式) 11: 无效	00
6: 5	T0CFG[1: 0]	R/W	T0通道配置位。 配置为捕获模式时，捕获边沿选择： 00: 不使能 01: 上升沿捕获 10: 下降沿捕获 11: 上升/下降沿捕获 配置为比较输出时，输出极性选择： 0X: 正向输出 1X: 反向输出	00
7	T0CLKS	R/W	T0时钟选择位。 0: 系统外设时钟(PCLK)分频时钟输入 1: 外部时钟输入脚(T0CK)	0
9: 8	FILTER[1: 0]	R/W	FILTER[1: 0]: T0外部事件消抖时间设置。 00: 关闭消抖功能 01: 消抖时间为3个系统外设时钟(PCLK) 10: 消抖时间为7个系统外设时钟(PCLK) 11: 消抖时间为15个系统外设时钟(PCLK)	00
10	UEV	R/W	T0缓冲更新事件控制。 0: 软件触发缓冲更新(UEV = 0, SUPD = 1同时满足时) 1: 比较匹配时，触发缓冲更新	0
11	PUPDEN	R/W	T0计数器周期溢出时，触发周期寄存器的缓冲更新。 0: 禁止缓冲更新 1: 使能缓冲更新	0
12	NOVFLG	R/W	T0的N周期溢出中断标志。当N次周期溢出时，该标志位置1，NOVFLG需要软件写1清零。 0: 未发生“N周期溢出事件” 1: “N周期溢出事件”发生	0
13	COMFLG	R/W	T0捕获或比较匹配中断标志。COMFLG需要软件写1清零。 0: 未发生捕获事件或比较匹配事件 1: 发生捕获事件或比较匹配事件	0
14	T0CLR	R/W	T0计数器清零位。 0: 无效	0

描述位	名称	读写	描述	复位值
			1: 计数器清零, 清零完成后, T0CLR位自动清零	
15	SUPD	R/W	T0软件触发缓冲更新。SUPD置1触发缓冲更新, 更新动作完成后, SUPD位自动清零。 0: 无效 1: 触发缓冲更新(UEV = 0, SUPD = 1同时满足时), 周期寄存器T0PER和捕获/比较寄存器T0CCR都有1个16位缓冲寄存器。	0
23: 16	T0OVPN[7: 0]	R/W	定时器T0溢出次数设置。每N周期溢出后, 产生1次“N周期溢出事件”。 $N = T0OVPN[7: 0] + 1$ 。	0x00
31: 24			保留	

### 9.2.2. 定时器 T0 周期寄存器 (T0PER: 0x40000804)

描述位	名称	读写	描述	复位值
15: 0	T0PER[15: 0]	R/W	T0周期寄存器。	0xFFFF

### 9.2.3. 定时器 T0 计数器 (T0CNT: 0x40000808)

描述位	名称	读写	描述	复位值
15: 0	T0CNT[15: 0]	R/W	定时器T0计数器当前计数值。	0x0000

### 9.2.4. 定时器 T0 捕获比较寄存器 (T0CCR: 0x4000080C)

描述位	名称	读写	描述	复位值
15: 0	T0CCR [15: 0]	R/W	T0捕获/比较寄存器, 仅在PWM模式下可读可写(R/W), 其他模式下只读(RO)。 定时模式时, T0计数器的当前计数值可以由T0CCR读取。	0x0000

### 9.2.5. 定时器 T1 控制寄存器(T1CON: 0x40001000)

描述位	名称	读写	描述	复位值
0	T1EN	R/W	定时器T1使能信号。 0: 定时器关闭 1: 定时器使能, T1EN由0→1过程, 计数器值会被复位清零。	0
1	OVFCLREN	R/W	T1周期溢出清零使能位。该位只有捕获模式且CAPCLREN = 1时有意义。 0: 周期溢出时计数器不清零。即计数器暂停计数, 直到发生新的捕获事件, 计数器将重新从0开始计数。 1: 周期溢出时计数器清零, 并重新从0开始计数。	0
2	CAPCLREN	R/W	T1捕获清零使能位。该位只有捕获模式有意义。 0: 捕获发生时计数器不清零。计数器将继续向上计数, 直到周期溢出时, 计数器重新从0开始计数(不论OVFCLREN等于0或1)。 1: 捕获发生时计数器清零, 计数器将重新从0开始计数。	0
4: 3	MODE[1: 0]	R/W	T1模式控制位。 00: 定时模式 01: 捕获输入模式 10: 比较输出模式(PWM模式) 11: 无效, 不能用	00
6: 5	T1CFG[1: 0]	R/W	T1通道配置位。 通道作为捕获模式时, 捕获边沿选择: 00: 不使能 01: 上升沿捕获 10: 下降沿捕获 11: 上升/下降沿捕获 通道作为比较输出时, 输出极性选择: 0X: 正向输出 1X: 反向输出	00
7	T1CLKS	R/W	T1时钟选择位。 0: 系统外设时钟(PCLK)分频时钟输入, PCLK分频倍数的设置, 见SYSCFG控制章节的PCLKSEL寄存器。	0

描述位	名称	读写	描述	复位值
			1: T0计数溢出信号(用于T0、T1级联模式, T0溢出作T1的时钟)	
9: 8	FILTER[1: 0]	R/W	T1消抖时间设置。 00: 关闭消抖功能 01: 消抖时间为3个系统外设时钟(PCLK) 10: 消抖时间为7个系统外设时钟(PCLK) 11: 消抖时间为15个系统外设时钟(PCLK)	00
10	UEV		T1缓冲更新事件控制。 0: 软件触发缓冲更新(UEV = 0, SUPD = 1同时满足时) 1: 通道1比较匹配时, 触发缓冲更新	0
11	PUPDEN	R/W	T1计数器周期溢出时, 触发缓冲更新。 0: 禁止 1: 使能	0
12	OVFLG	R/W	T1的周期溢出中断标志。中断标志OVFLG需要软件写1清零。 0: 未产生周期溢出事件 1: 周期溢出事件发生	0
13	COMFLG	R/W	T1捕获或者比较匹配中断标志。CPFLG需要软件写1清零。 0: 未发生捕获事件或比较匹配事件 1: 发生捕获事件或比较匹配事件	0
14	T1CLR	R/W	T1计数器清零, 清零完成后, T1CLR自动清零。 写0: 无效 写1: 计数器清零	0
15	SUPD	R/W	T1软件触发缓冲更新。SUPD置1触发缓冲更新, 更新动作完成后, SUPD位自动清零。 写0: 无效 写1: 触发缓冲更新(UEV = 0, SUPD = 1同时满足时), 周期寄存器T1PER和捕获/比较寄存器T1CCR都有1个16位缓冲寄存器。	0

### 9.2.6. 定时器 T1 周期寄存器 (T1PER: 0x40001004)

描述位	名称	读写	描述	复位值
15: 0	T1PER[15: 0]	R/W	T1周期寄存器。	0xFFFF

### 9.2.7. 定时器 T1 计数寄存器 (T1CNT: 0x40001008)

描述位	名称	读写	描述	复位值
15: 0	T1CNT[15: 0]	R/W	定时器T1计数器当前计数值。	0x0000

### 9.2.8. 定时器 T1 捕获比较寄存器 (T1CCR: 0x4000100C)

描述位	名称	读写	描述	复位值
15: 0	T1CCR [15: 0]	R/W	T1捕获/比较寄存器，仅在PWM模式下可读可写(R/W)，其他模式下只读(RO)。定时模式时，T1计数器的当前计数值可以由T1CCR读取。	0x0000

## 9.3. 功能描述

### 9.3.1. 定时模式

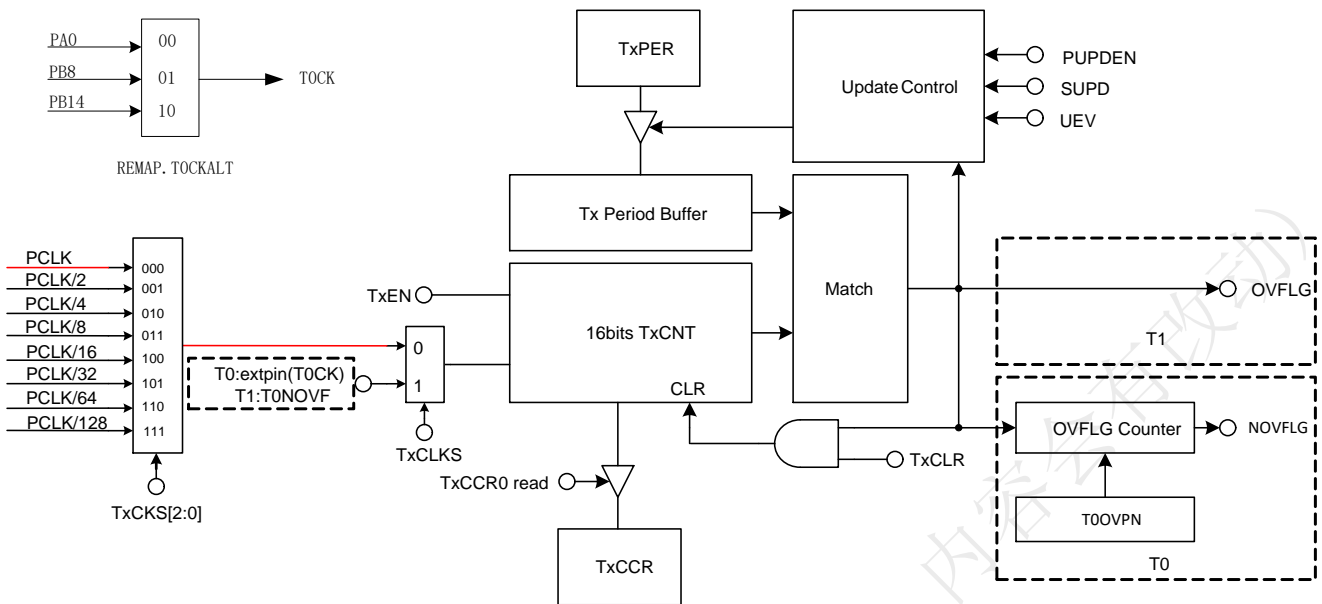


图 9-3-1-1 T0/T1 定时模式

### ● 模式配置、时钟源配置

定时器 T0/T1 控制寄存器(T0CON/T1CON)的 MODE[1: 0]位用于选择工作模式。当 MODE[1: 0] = 00 时, 定时器工作在定时模式。

控制寄存器(T0CON/T1CON)的 T0CLKS/T1CLKS 位, 用于配置计数时钟源。当 T0CLKS/T1CLKS = 0 时, 计数时钟源选择系统外设时钟(PCLK)预分频; 当 T0CLKS = 1 时, T0 计数时钟源选择外部时钟脚 T0CK (PA0、PB4、PB8 复用脚)。当 T1CLKS = 1 时, 计数时钟源选择 T0 的 NOVFLG 信号。

T0EN/T1EN 置 1, 定时器 T0/T1 启动工作, 计数器从 0 开始向上计数, 当计数器值等于周期寄存器值时(周期溢出), 计数器值清零, 并重新从 0 开始计数。

计数时钟源选择系统外设时钟(PCLK)时, 可以通过 TxCLKS[2: 0]设置分频系数, 设置方法详见 SYSCFG 控制章节的 PCLKSEL 寄存器。

### ● 周期溢出中断和 N 周期溢出中断

定时模式下, 定时器 T1 只有周期溢出中断; 定时器 T0 有 N 周期溢出中断, 当 N = 1 时, N 周期溢出中断相当于周期溢出中断。

#### (1)T1 的周期溢出中断

计数器与周期寄存器(T1PER)的值相等时, 周期溢出中断标志(OVFLG)将会被置 1。此时如果周期溢出中断已经使能, CPU 将响应中断。周期溢出中断标志(OVFLG)只能通过软件写 1 清除。

#### (2)T0 的 N 周期溢出中断

每出现 N 次周期溢出时, 就将“N 周期溢出中断标志 NOVFLG”置 1。此时如果“N 周期溢出中断”已经使能, CPU 将响应中断。N 周期溢出中断标志(NOVFLG)只能通过软件写 1 清除。

● 周期缓冲寄存器的数据更新

为了实现 16 位周期寄存器(TxPER)的同步更新，定时器 T0/T1 设计了周期寄存器缓冲 (Period Buffer)。周期寄存器缓冲的更新，有以下 2 种方法: (1)软件触发缓冲更新；(2)计数器与周期寄存器的值相等时更新，也叫计数器溢出时刻更新。

(1)软件触发缓冲更新，寄存器设置的方法: 当 UEV = 0 时，每次 SUPD 置 1，触发 1 次缓冲更新，更新动作完成后，SUPD 自动清零。

(2)计数器与周期寄存器相等时更新，寄存器设置的方法: 寄存器 T0CON/T1CON 的 PUPDEN 位设置为 1，每次周期溢出时，即计数器与周期寄存器(T0PER/T1PER)相等时，更新周期缓冲寄存器值。

● 计数器当前计数值的读取

定时模式下，计数器(Counter)的当前计数值可以通过读取“T0 捕获比较寄存器 T0CCR”或“T1 捕获比较寄存器 T1CCR”来获取。

定时模式举例

定时模式举例 1: T0CON = 0x01, T0PER = 0x0002, T0OVPN = 0x02。本例中，计数时钟源选择系统外设时钟(PCLK)，无分频；计数器累加到 0x02，会产生 1 次周期溢出事件，计数器(Counter)清零并重新从“0”开始计数，每 3 次周期溢出会产生 1 次“N 周期溢出事件”，其中 N = 3。如下图所示:

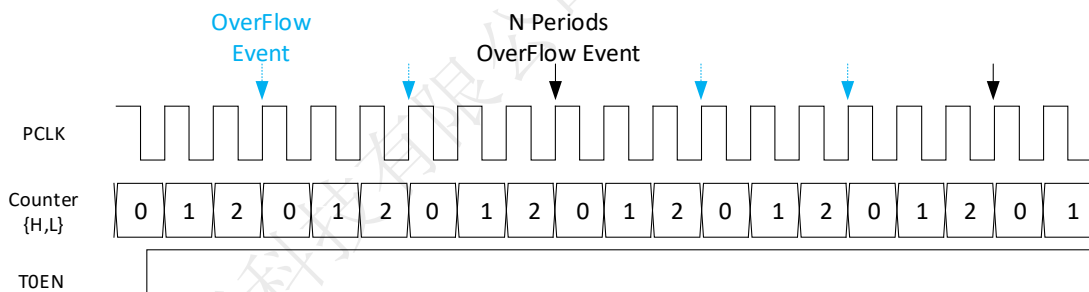


图 9-3-1-2 定时器 T0 定时模式时序图

### 9.3.2. 捕获模式

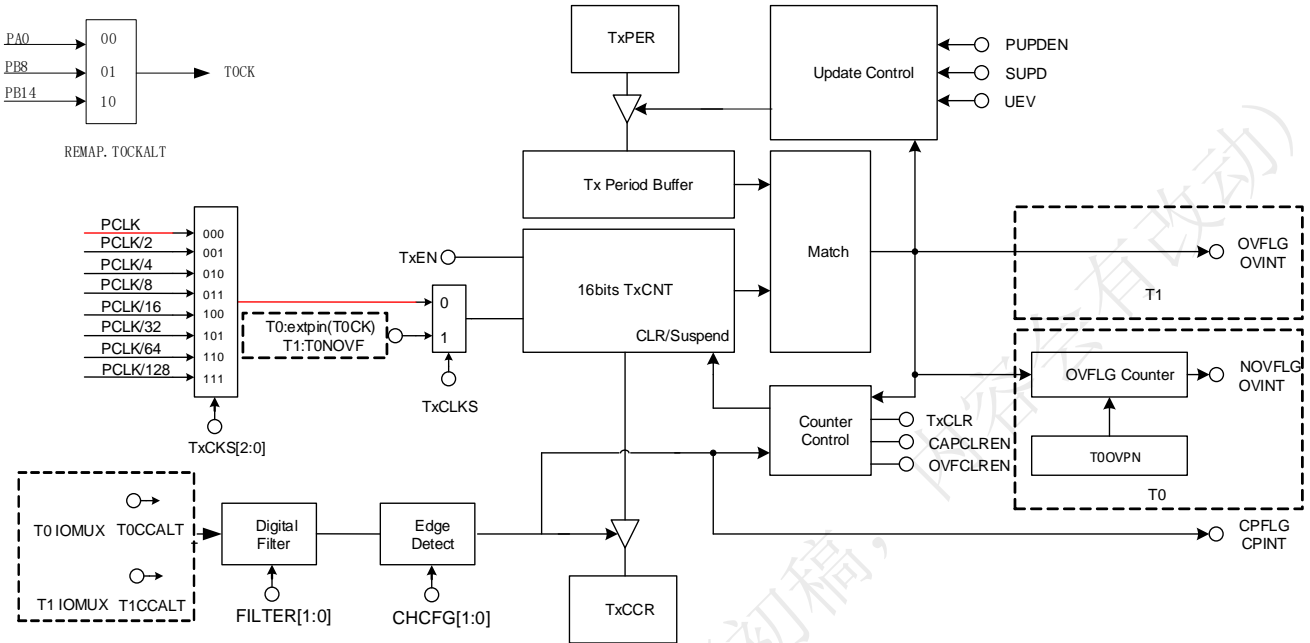


图 9-3-2-1 定时器 T0/T1 捕获模式

#### ● 模式配置、时钟源配置

当控制寄存器 1(TxCON)的 MODE[1: 0] = 01 时，定时器 Tx 的通道 0 配置为捕获输入模式(Capture mode)。这时，外部触发信号由 TxCCR 输入，TxCCR 与外部 GPIO 的对应关系，见 GPIO 配置及引脚布局图等相应章节。

控制寄存器 1(TxCON)的 TxCLKS 位，用于配置计数时钟源。当 TxCLKS = 0 时，计数时钟源选择系统外设时钟(PCLK)预分频；当 T0CLKS = 1 时，计数时钟源选择外部时钟脚 T0 (PA0 复用脚)。当 T1CLKS = 1 时，计数时钟源选择 T0 的 NOVFLG 信号。

TxEN 置 1，定时器 T0/T1 启动工作，计数器(Counter)从 0 开始计数。

计数时钟源选择系统外设时钟(PCLK)时，可以通过 TxCLKS[2: 0]设置分频系数，设置方法详见见 SYSCFG 控制章节的 PCLKSEL 寄存器。

#### ● 捕获模式的计数器清零和暂停

捕获模式下，TxEN 置 1，计数器启动，从 0 开始计数，在捕获事件或周期溢出事件发生之前，计数器的值将会一直累加。标志 CAPCLREN 和 OVFLREN 的设置决定了“捕获或周期溢出”时刻，计数器的工作模式是“继续累加、暂停累加或清零计数值”。详见寄存器描述。

#### ● 捕获输入信号源

定时器 T0/T1 的捕获输入通道从外部 IO 复用脚 TxCCR 输入，TxCCR 与外部 GPIO 的对应关系，见 GPIO 配置及引脚布局图等相应章节。

#### ● 捕获输入信号消抖

定时器 T0/T1 的捕获输入通道 TxCCR 设有消抖模块，其作用是对捕获输入信号进行采样，采样周期为系统外设时钟周期(PCLK)，并对比连续“0/4/8/16”个采样信号。当输入信号发生翻转，且 0/3/7/15 个采样信号的电平相同时，则消抖模块的输出信号翻转。

● 捕获中断和周期溢出中断

捕获模式下，当定时器 T0/T1 发生捕获事件时，捕获中断标志(CPFLG)将会被置 1。如果此时捕获中断使能(详见寄存器 IE1)，CPU 将响应捕获中断。捕获中断标志(CPFLG)需要软件置 1 清零。

捕获模式下，定时器 T0 具有“N 周期溢出中断”，定时器 T1 具有周期溢出中断。详细描述，参考 T0/T1 定时模式的相关描述。

● 捕获数据读取的注意事项

捕获事件发生的时刻，计数器的值存入相应通道的捕获/比较寄存器(TxCCR)，读取捕获/比较寄存器值时。

● 捕获模式举例

捕获模式举例 1: T0CON = 0x29，定时器 T0 的通道 T0CCR 设置为上升沿捕获模式，捕获发生时，计数器不清零；T0PER = 0x00FF，周期寄存器设置为 0x00FF。当计数到 0x0013 时，检测到 T0CCR 上升沿触发捕获(Edge Detect 为高电平)，计数器值 0x0013 被捕获，并存入寄存器 T0CCR。

捕获模式举例 2: T0CON = 0x2D，定时器 T0 的通道 T0CCR 设置为上升沿捕获模式，捕获发生时，计数器清零；T0PER = 0x00FF，周期寄存器设置为 0x00FF。当计数到 0x0013 时，检测到 T0CCR 上升沿触发捕获(Edge Detect 为高电平)，计数器值 0x0013 被捕获，并存入寄存器 T0CCR；计数器重新从 0 开始计数。

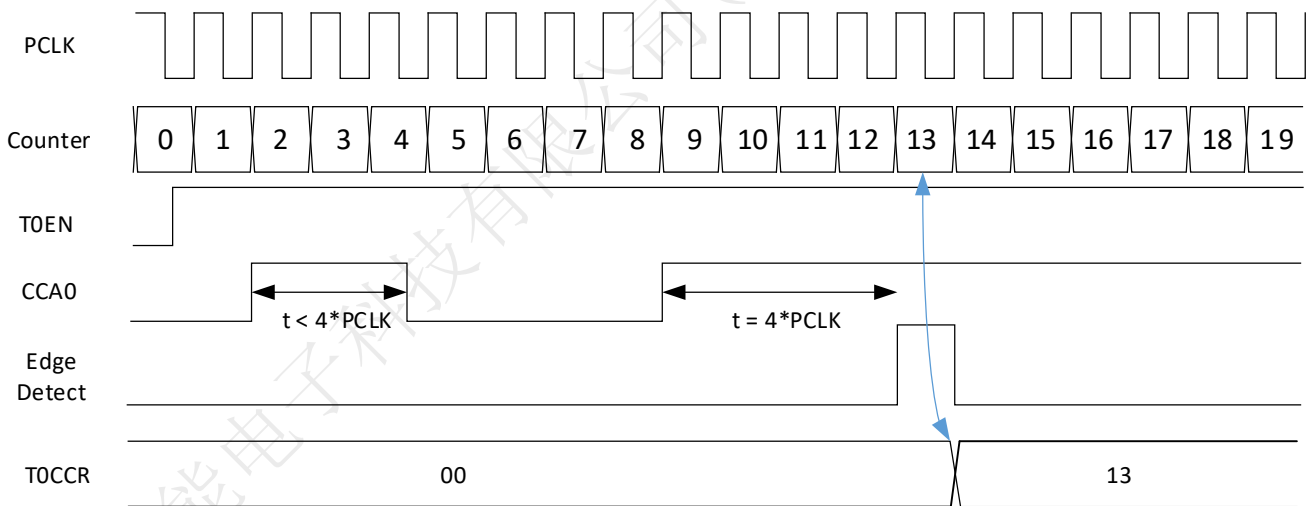


图 9-3-2-2 捕获模式时序图(举例 1)

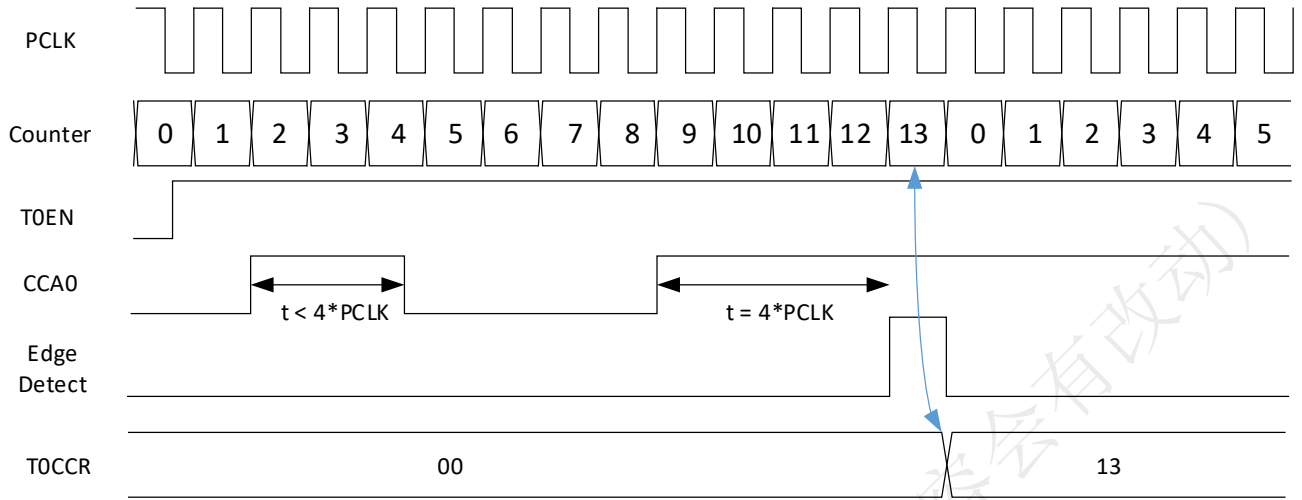


图 9-3-2-3 捕获模式时序图(举例 2)

### 9.3.3. PWM 模式

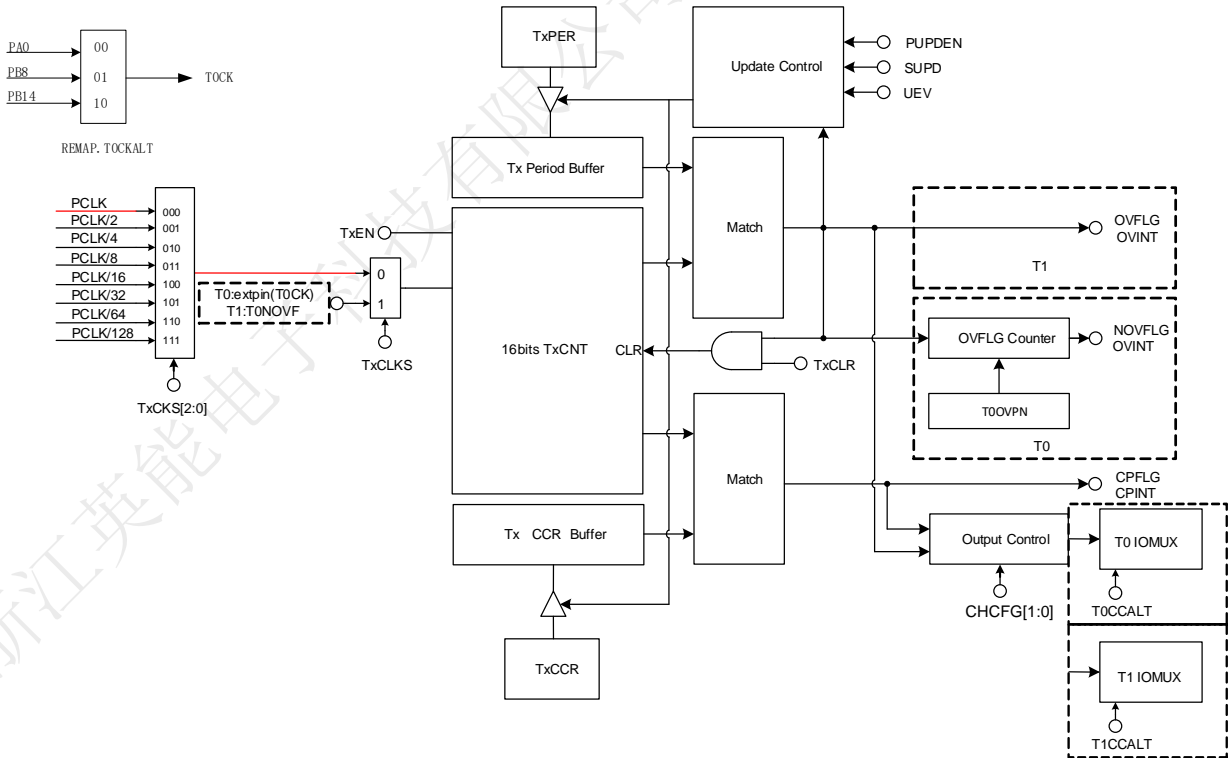


图 9-3-3-1 定时器 T0/T1 PWM 模式

### ● 模式配置、时钟源配置

当控制寄存器 1(TxCON)的 MODE[1: 0] = 10 时, 定时器 Tx 的通道 0 配置为比较输出模式(PWM mode), PWM 波形可由 TxCCR 输出, TxCCR 与外部 GPIO 对应关系, 见 GPIO 配置及引脚布局图等相应章节。

控制寄存器 1(TxCON)的 TxCLKS 位, 用于配置计数时钟源。当 TxCLKS = 0 时, 计数时钟源选择系统外设时钟(PCLK)预分频; 当 T0CLKS = 1 时, 计数时钟源选择外部时钟脚 T0 (P10 复用脚)。当 T1CLKS = 1 时, 计数时钟源选择 T0 的 NOVFLG 信号。

TxEN 置 1, 定时器 Tx 启动工作, 计数器从 0 开始计数。

计数时钟源选择系统外设时钟(PCLK)时, 可以通过 TxCLKS[2: 0]设置分频系数, 设置方法详见 SYSCFG 控制章节的 PCLKSEL 寄存器。

### ● 比较匹配中断和周期溢出中断

PWM 模式下, 当计数器(Counter)的值和捕获/比较缓冲寄存器(TxCCR)的值相等时, 比较匹配中断标志 CH0FLG 将会被置 1。如果此时比较匹配中断使能, CPU 将响应比较匹配中断。比较匹配发生时不会清零计数器, 而是继续累加, 直到计数器与周期寄存器的值相等时发生周期溢出, 计数器重新从 0x0000 开始计数。

PWM 模式下, 定时器 T0 具有 N 周期溢出中断, 定时器 T1 具有单周期溢出中断。

### ● 输出极性

定时器 T0/T1 控制寄存器 1(TxCON)的 TxCFG 默认为 0x, 此时比较输出为正向输出, 当计数器的值小于等于捕获/比较寄存器的值(Counter ≤ TxCCR)时, 通道 TxCCR 输出为高电平, 当计数器的值大于捕获/比较缓冲寄存器的值(Counter > TxPER)时, 通道 TxCCR 输出为低电平; 周期寄存器的值或者捕获/比较寄存器的值都为 0 时, 通道 TxCCR 输出为恒低电平; 捕获/比较缓冲寄存器的值大于等于周期寄存器的值(TxCCR ≥ TxPER)时, 通道 TxCCR 输出为恒高电平。

定时器 T0/T1 控制寄存器 1(TxCON)的 TxCFG 设置为 1x, 此时比较输出为反向输出, 当计数器的值小于等于捕获/比较寄存器的值(Counter ≤ TxCCR)时, 通道 TxCCR 输出为低电平, 当计数器的值大于捕获/比较寄存器的值(Counter > TxPER)时, 通道 TxCCR 输出为高电平; 周期寄存器的值或者捕获/比较寄存器的值都为 0 时, 通道 TxCCR 输出为恒高电平; 捕获/比较寄存器的值大于等于周期寄存器的值(TxCCR ≥ TxPER)时, 通道 TxCCR 输出为恒低电平。

### ● 缓冲寄存器更新

周期缓冲寄存器和捕获/比较缓冲寄存器的更新, 可以由相应的事件进行触发。具体可以由以下几种方式实现:

- (1)软件触发缓冲更新;
- (2)周期溢出时, 即计数器与周期缓冲寄存器(TxPER)的值相等时缓冲更新;
- (3)通道匹配时, 即计数器与捕获/比较寄存器(TxCCR)的值相等时, 缓冲更新。

软件触发缓冲更新，寄存器设置的方法：当  $UEV = 0$  时，每次  $SUPD$  设置为 1 时，触发 1 次缓冲更新，更新动作完成后， $SUPD$  自动清零。

计数器与周期寄存器匹配时更新，寄存器设置的方法：寄存器  $TxCON$  的  $PUPDEN$  位设置为 1，每次溢出时，即计数器(Counter)与周期寄存器( $TxPER$ )相等时，更新周期缓冲寄存器和捕获/比较缓冲寄存器。

通道匹配时缓冲更新，寄存器设置的方法：寄存器  $TxCON$  的  $UEV$  位设置为 1，通道  $TxCCR$  匹配时(即  $Counter = TxCCR$ )，触发缓冲更新。

### ● PWM 模式举例

PWM 模式举例 1:  $T1CON = 0x51$ ,  $T1CCR0 = 0x0050$ ,  $T1PER = 0xE0$ 。通道  $T1CCR$  为反向输出，当计数器的值小于等于  $0x50$  时， $T1CCR$  输出为低电平，当计数器的值大于  $0x50$  时， $T1PER$  输出为高电平。计数器的值等于  $0xE0$  时，计数清零。

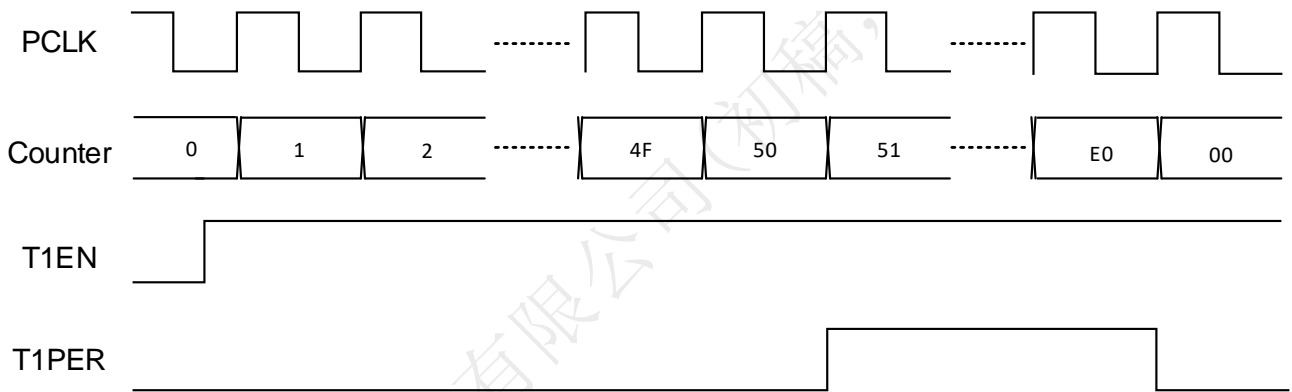


图 9-3-3-2 PWM 模式时序图

### 9.3.4. 事件更新

$T0/T1$  对周期寄存器、捕获/比较寄存器都做了缓冲处理，触发缓冲更新的事件发生时，配置在以上寄存器的值才会更新到缓冲寄存器中。更新事件可以选择软件触发更新、周期溢出更新、捕获/匹配更新。寄存器  $TxCON$  的  $SUPD$ 、 $PUPDEN$ 、 $UEV$  可以对更新方式进行配置。

### 9.3.5. 中断源

定时器  $T0/T1$  分别有两个中断源，对应两个中断向量，分别是计数器(N 或 1)周期溢出中断、通道 0 的比较匹配/捕获中断。详见“中断异常控制”章节。

## 10. 定时器(TIMER2)

16 位定时器:

- 定时模式
- 比较输出模式(PWM mode)
- 捕获模式
- 系统外设时钟(PCLK)分频时钟输入
- 最大支持 3 路外部 I/O 通道，3 路通道可以同时配置为“捕获模式”，或同时配置为“比较输出模式”
- 捕获模式支持 4 种硬件捕获方式(不捕获、上升沿、下降沿、上升/下降沿)；支持软件触发捕获方式
- 硬件捕获可选 2 种捕获信号源：外部 I/O 输入或内部比较器输出；具有输入信号数字滤波功能，可以有效防止干扰信号导致的捕获误动作
- 捕获模式支持 3 个捕获中断和 1 个周期溢出中断
- 比较输出模式支持 3 个比较匹配中断和 1 个周期溢出中断

### 10.1. 模块寄存器总表

定时器 timer2 控制的主要寄存器列表如下:

寄存器名	地址	结构体(.h)	寄存器描述	复位值
T2CON	0x40001800	TIM2	定时器T2控制寄存器	0x0000
T2CFG	0x40001804	TIM2	定时器T2配置寄存器	0x0000
T2ICON	0x40001808	TIM2	定时器T2中断控制与标志寄存器	0x00
T2PER	0x4000180C	TIM2	定时器T2周期寄存器	0xFFFF
T2CNT	0x40001810	TIM2	定时器T2计数器	0x0000
T2CCR0	0x40001814	TIM2	定时器T2捕获/比较寄存器0	0x0000
T2CCR1	0x40001818	TIM2	定时器T2捕获/比较寄存器1	0x0000
T2CCR2	0x4000181C	TIM2	定时器T2捕获/比较寄存器2	0x0000

### 10.2. 寄存器说明

#### 10.2.1. 定时器 T2 控制寄存器(T2CON: 0x40001800)

描述位	名称	读写	描述	复位值
0	T2EN	R/W	定时器T2使能位。 0: 定时器关闭	0

描述位	名称	读写	描述	复位值
			1: 定时器使能, T2EN由0→1过程, 计数器值会被复位清零。	
1	SCAPEN	R/W	软件触发捕获功能使能。 0: 无效 1: 使能	0
2	OVFCLREN	R/W	周期溢出清零使能位。该位只有捕获模式且CAPCLREN = 1时有意义。 0: 周期溢出时计数器不清零。即计数器暂停计数, 直到发生新的捕获事件, 计数器将重新从0开始计数。 1: 周期溢出时计数器清零, 并重新从0开始计数。	0
3	CAPCLREN	R/W	捕获清零使能位。该位只有捕获模式有意义, 即MODE=1。 0: 捕获发生时计数器不清零。计数器将继续向上计数, 直到周期溢出时, 计数器重新从0开始计数(不论OVFCLREN等于0或1)。 1: 捕获发生时计数器清零, 计数器将重新从0开始计数。	0
4	MODE	R/W	工作模式选择位。 0: 定时模式 1: 捕获/比较模式。若CHXCEN = 0, 则通道T2CCR0/ T2CCR1/ T2CCR2为捕获输入模式; 若CHXCEN = 1, 则通道T2CCR0/ T2CCR1/ T2CCR2为比较输出模式	0
5	CHXCEN	R/W	通道T2CCR0/ T2CCR1/ T2CCR2模式选择位, 捕获/比较选择。只有MODE = 1时, 设置才有意义。 0: 3路通道全部设置为捕获输入 1: 3路通道全部设置为比较输出	0
6	SCAP	R/W	软件触发捕获。 0: 无效 1: 软件触发捕获。SCAPEN=1时, SCAP置1, 将会触发1次捕获功能, 捕获动作完成后, SCAP自动清零。 软件触发捕获时, 每次捕获的计数器值都会存入T2CCR1。	0
7	T2CLKS	R/W	T2CLKS: 时钟信号源选择。 0: 系统外设时钟(PCLK)分频时钟 1: 无效	0

描述位	名称	读写	描述	复位值
10: 8	UEV[2: 0]	R/W	比较输出模式缓冲更新事件控制。 000: 软件触发缓冲更新(UEV[2: 0] = 000, SUPD = 1同时满足时) 001: 通道0匹配时, 触发缓冲更新 010: 通道1匹配时, 触发缓冲更新 011: 通道2匹配时, 触发缓冲更新 100: 通道0/1匹配时, 触发缓冲更新 101: 通道0/2匹配时, 触发缓冲更新 110: 通道1/2匹配时, 触发缓冲更新 111: 通道0/1/2匹配时, 触发缓冲更新	000
11	PUPDEN	R/W	计数器溢出时, 即计数器值与周期寄存器(T2PER)的值相等的时刻, 触发缓冲(buffer)更新。 0: 禁止 1: 使能	0
12	CAPDT0	RO	捕获通道0的输入电平(滤波后)。 0: 低电平 1: 高电平	0
13	CAPDT1	RO	捕获通道1的输入电平(滤波后)。 0: 低电平 1: 高电平	0
14	CAPDT2	RO	捕获通道2的输入电平(滤波后)。 0: 低电平 1: 高电平	0
15	SUPD	R/W	软件触发缓冲更新控制位。SUPD置1触发缓冲更新, 更新动作完成后, SUPD自动清零。 写0: 无效 写1: 触发缓冲更新(UEV[2: 0] = 000, SUPD = 1同时满足时)	0

### 10.2.2. 定时器 T2 配置寄存器(T2CFG: 0x40001804)

描述位	名称	读写	描述	复位值
1: 0	CH0CFG[1: 0]	R/W	捕获模式时, 通道0的捕获边沿选择。 00: 不使能 01: 上升沿捕获 10: 下降沿捕获 11: 上升/下降沿捕获 比较输出模式时, 通道0的输出极性选择。	00

描述位	名称	读写	描述	复位值
			0X: 正向输出 1X: 反向输出	
3: 2	CH1CFG[1: 0]	R/W	捕获模式时，通道1的捕获边沿选择。 00: 不使能 01: 上升沿捕获 10: 下降沿捕获 11: 上升/下降沿捕获 比较输出模式时，通道1的输出极性选择。 0X: 正向输出 1X: 反向输出	00
5: 4	CH2CFG[1: 0]	R/W	捕获模式时，通道2的捕获边沿选择。 00: 不使能 01: 上升沿捕获 10: 下降沿捕获 11: 上升/下降沿捕获 比较输出模式时，通道2的输出极性选择。 0X: 正向输出 1X: 反向输出	00
6	T2CLR	R/W	定时器T2计数器清零，清零动作完成后，T2CLR自动清零。 0: 无效 1: 计数器清零	0
7	CAPINIT	R/W	捕获寄存器初始化。初始化动作完成后，CAPINIT标志自动清零。 0: 无效 1: 捕获寄存器初始化。包括: T2CCR0/1/2初始化为0xFFFF。	
13: 8	FSVAL[5: 0]	R/W	捕获滤波的时间设置，最长63个采样时钟。 000000: 滤波功能关闭; 000001: 滤波功能相当于定时采样。	000000
15: 14	FSCLK[1: 0]	R/W	捕获滤波的滤波器采样时钟配置。 00: 1个系统外设时钟(PCLK) 01: 2个系统外设时钟(PCLK) 10: 4个系统外设时钟(PCLK) 11: 16个系统外设时钟(PCLK)	00

### 10.2.3. 定时器 T2 中断寄存器(T2ICON: 0x40001808)

描述位	名称	读写	描述	复位值
0	CH0IEN	R/W	通道0中断使能。 0: 禁止 1: 使能	0
1	CH1IEN	R/W	通道1中断使能。 0: 禁止 1: 使能	0
2	CH2IEN	R/W	通道2中断使能。 0: 禁止 1: 使能	0
3			保留	
4	CH0FLG	R/W	通道0捕获或者比较匹配中断标志。中断标志需要软件置1清零。 0: 未发生捕获事件或者比较匹配事件 1: 发生捕获事件或者比较匹配事件	0
5	CH1FLG	R/W	通道1捕获或者比较匹配中断标志。中断标志需要软件置1清零。 0: 未发生捕获事件或者比较匹配事件 1: 发生捕获事件或者比较匹配事件	0
6	CH2FLG	R/W	通道2捕获或者比较匹配中断标志。中断标志需要软件置1清零。 0: 未发生捕获事件或者比较匹配事件 1: 发生捕获事件或者比较匹配事件	0
7	OVFLG	R/W	周期溢出中断标志。中断标志需要软件置1清零。 0: 未产生周期溢出事件 1: 周期溢出事件发生	0

### 10.2.4. 定时器 T2 周期寄存器 (T2PER: 0x4000180C)

描述位	名称	读写	描述	复位值
15: 0	T2PER[15: 0]	R/W	T2周期寄存器。	0xFFFF

### 10.2.5. 定时器 T2 计数寄存器 (T2CNT: 0x40001810)

描述位	名称	读写	描述	复位值
15: 0	T2CNT[15: 0]	R/W	T2计数寄存器。	0x0000

### 10.2.6. 定时器 T2 捕获/比较寄存器 (T2CCR0: 0x40001814)

描述位	名称	读写	描述	复位值
15: 0	T2CCR0[15: 0]	R/W	定时器T2捕获/比较寄存器0，仅在PWM模式下可读可写(R/W)，其他模式下只读(RO)。	0x0000

### 10.2.7. 定时器 T2 捕获/比较寄存器 (T2CCR1: 0x40001818)

描述位	名称	读写	描述	复位值
15: 0	T2CCR1[15: 0]	R/W	定时器T2捕获/比较寄存器1，仅在PWM模式下可读可写(R/W)，其他模式下只读(RO)。软件触发捕获时，每次捕获的计数器值都会存入T2CCR1。	0x0000

### 10.2.8. 定时器 T2 捕获/比较寄存器 (T2CCR2: 0x4000181C)

描述位	名称	读写	描述	复位值
15: 0	T2CCR2[15: 0]	R/W	定时器T2捕获/比较寄存器1，仅在PWM模式下可读可写(R/W)，其他模式下只读(RO)。	0x0000

## 10.3.功能描述

### 10.3.1. 定时模式

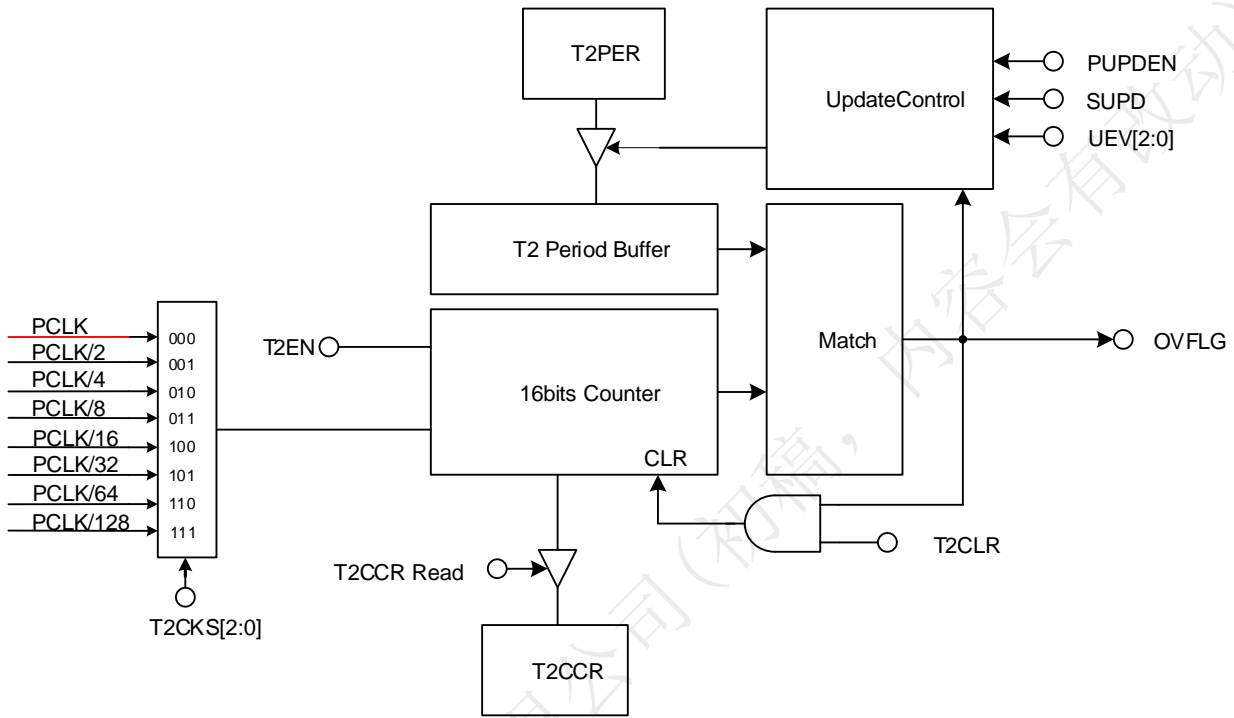


图 10-3-1 定时器 T2 定时模式

#### ● 模式配置、时钟源配置

定时器 T2 控制寄存器(T2CON)的 MODE 位用于选择工作模式。当 MODE = 0 时，定时器工作在定时模式。T2CLKS 位用于配置计数时钟源，计数时钟源选择系统外设时钟(PCLK)预分频；

T2EN 置 1，定时器 T2 启动工作，计数器从 0 开始计数，直到周期溢出时，计数器清零并重新从 0 开始计数。

计数时钟源选择系统外设时钟(PCLK)时，可以通过 T2CLK[2: 0]设置分频系数，设置方法详见 SYSCFG 控制章节的 PCLKSEL 寄存器。

#### ● 周期溢出中断

定时模式下，计数器与周期寄存器(T2PER)相等时，周期溢出中断标志(OVFLG)将会被置 1。此时如果周期溢出中断已经使能(详见寄存器 IE1)，CPU 将响应溢出中断。周期溢出中断标志(OVFLG)只能通过软件写 1 清除。

#### ● 周期缓冲寄存器的数据更新

周期寄存器缓冲的更新，有以下 2 种方法：(1)软件触发缓冲更新；(2)计数器与周期寄存器的值相等时更新，也叫计数器溢出时刻更新。

(1)软件触发缓冲更新，寄存器设置的方法：当 UEV[2: 0] = 000 时，每次 SUPD 置 1 时，

触发 1 次缓冲更新，更新动作完成后，SUPD 自动清零。

(2)计数器与周期寄存器相等时更新，寄存器设置的方法: 寄存器 T2CON 的 PUPDEN 位设置为 1，每次周期溢出时，即计数器与周期寄存器(T2PER)相等时，更新周期缓冲寄存器值。

### 10.3.2. PWM 模式

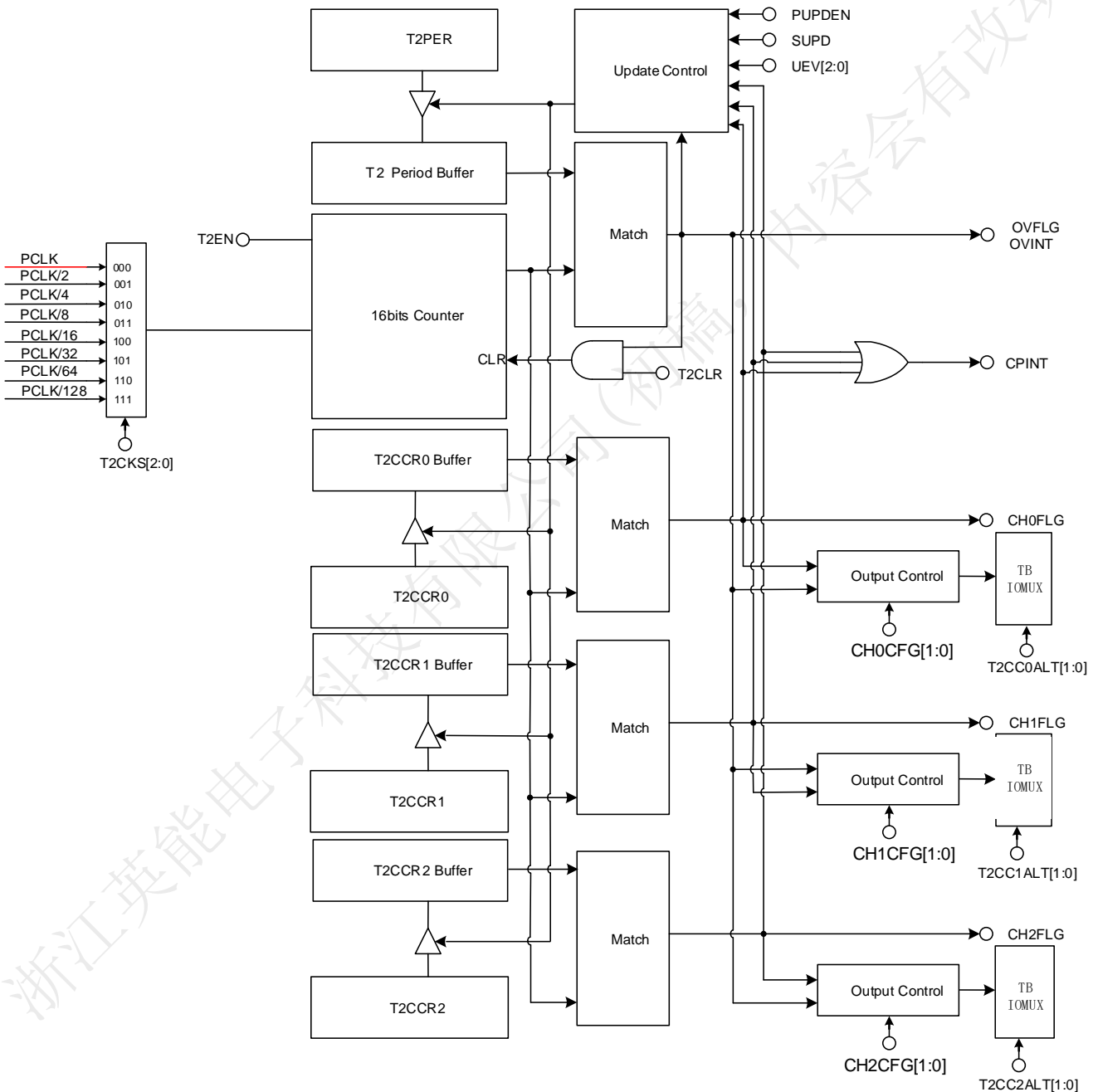


图 10-3-2-1 定时器 T2 PWM 模式

- 模式配置、时钟源配置

当控制寄存器 1(T2CON)的 MODE=1 且 CHXCEN=1 时, 通道 0/1/2 全部配置为比较输出模式(PWM mode)。这时 3 路 PWM 通道与外部 GPIO 复用功能 T2CCR0/1/2 相对应。

控制寄存器(T2CON)的 T2CLKS 位, 用于配置计数时钟源。

#### ● 匹配中断

PWM 模式下, 当计数器的值和捕获/比较缓冲寄存器(T2CCR0/1/2)的值相等时, 比较匹配中断标志 CH0FLG/CH1FLG/CH2FLG 将会被置 1。如果相应的中断使能(CH0IEN / CH1IEN/CH2IEN = 1), 将向 CPU 发出匹配中断请求。比较匹配中断标志(CHxFLG)通过软件置 1 清零。比较匹配事件发生时, 不会清零计数器, 而是继续累加, 直到计数器与周期缓冲寄存器的值相等时发生周期溢出, 计数器清零, 并重新从 0x0000 开始计数。

#### ● 周期溢出中断

PWM 模式下, 计数器与周期寄存器(T2PER)相等时, 周期溢出中断标志(OVFLG)将会被置 1。向 CPU 发出溢出中断请求。溢出中断标志(OVFLG)通过软件写 1 清除。

#### ● 输出极性

定时器 T2 配置寄存器(T2CFG)的 CHxCFG[1: 0]默认为 00, 此时比较输出为正向输出, 当计数器的值小于等于捕获/比较缓冲寄存器的值(Counter  $\leq$  T2CCRx)时, 相应通道的输出为高电平, 当计数器的值大于捕获/比较缓冲寄存器的值(Counter > T2CCRx)时, 相应通道的输出为低电平; 周期缓冲寄存器的值或者捕获/比较缓冲寄存器的值都为 0 时, 相应通道的输出为恒低电平; 捕获/比较缓冲寄存器的值大于等于周期缓冲寄存器的值(T2CCRx  $\geq$  T2PER)时, 相应通道的输出为恒高电平。

定时器 T2 配置寄存器(T2CFG)的 CHxCFG[1: 0]设置为 10 或 11, 此时比较输出为反向输出, 当计数器的值小于等于捕获/比较缓冲寄存器的值(Counter  $\leq$  T2CCRx)时, 相应通道的输出为低电平, 当计数器的值大于捕获/比较缓冲寄存器的值(Counter > T2CCRx)时, 相应通道的输出为高电平; 周期缓冲寄存器的值或者捕获/比较缓冲寄存器的值都为 0 时, 相应通道的输出为恒高电平; 捕获/比较缓冲寄存器的值大于等于周期缓冲寄存器的值(T2CCRx  $\geq$  T2PER)时, 相应通道的输出为恒低电平。

#### ● 缓冲寄存器更新

周期缓冲寄存器和捕获/比较缓冲寄存器的更新, 可以由相应的事件进行触发。具体可以由以下几种方式实现: (1)软件触发缓冲更新; (2)周期溢出时, 即计数器与周期缓冲寄存器(T2PER)的值相等时缓冲更新; (3)通道匹配时, 即计数器与捕获/比较寄存器(T2CCRx)的值相等时, 缓冲更新。

(1)软件触发缓冲更新, 寄存器设置的方法: 当 UEV[2: 0] = 000 时, 每次 SUPD 设置为 1 时, 触发 1 次缓冲更新, 更新动作完成后, SUPD 自动清零。

(2)计数器与周期寄存器相等时更新, 寄存器设置的方法: 寄存器 T2CON 的 PUPDEN 位设置为 1, 每次溢出时, 即计数器(Counter)与周期寄存器(T2PER)相等时, 更新周期缓冲寄存器和捕获/比较缓冲寄存器。

(3)通道匹配时缓冲更新, 寄存器设置的方法: 设置寄存器 T2CON 的 UEV[2: 0] = 001/010/011/100/ 101/110/111, 相应的通道 x 匹配时(即 Counter = T2CCRx), 触发缓冲更新。

#### ● PWM 模式举例

PWM 模式举例 1: T2CON = 0x31, T2CCR1 = 0x0050, T2PER = 0x00E0, T2CFG = 0x08, 当计数器的值小于等于 0x50 时, T2CCR 输出为低电平, 当计数器的值大于 0x50 时,

T2CCR 输出为高电平。

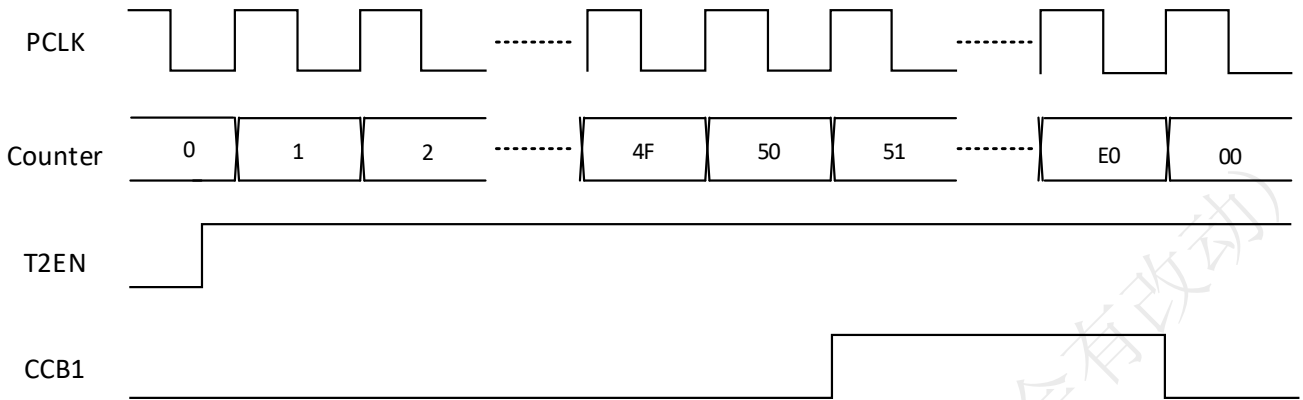


图 10-3-2-2 PWM 模式举例 1 的时序图

### 10.3.3. 捕获模式

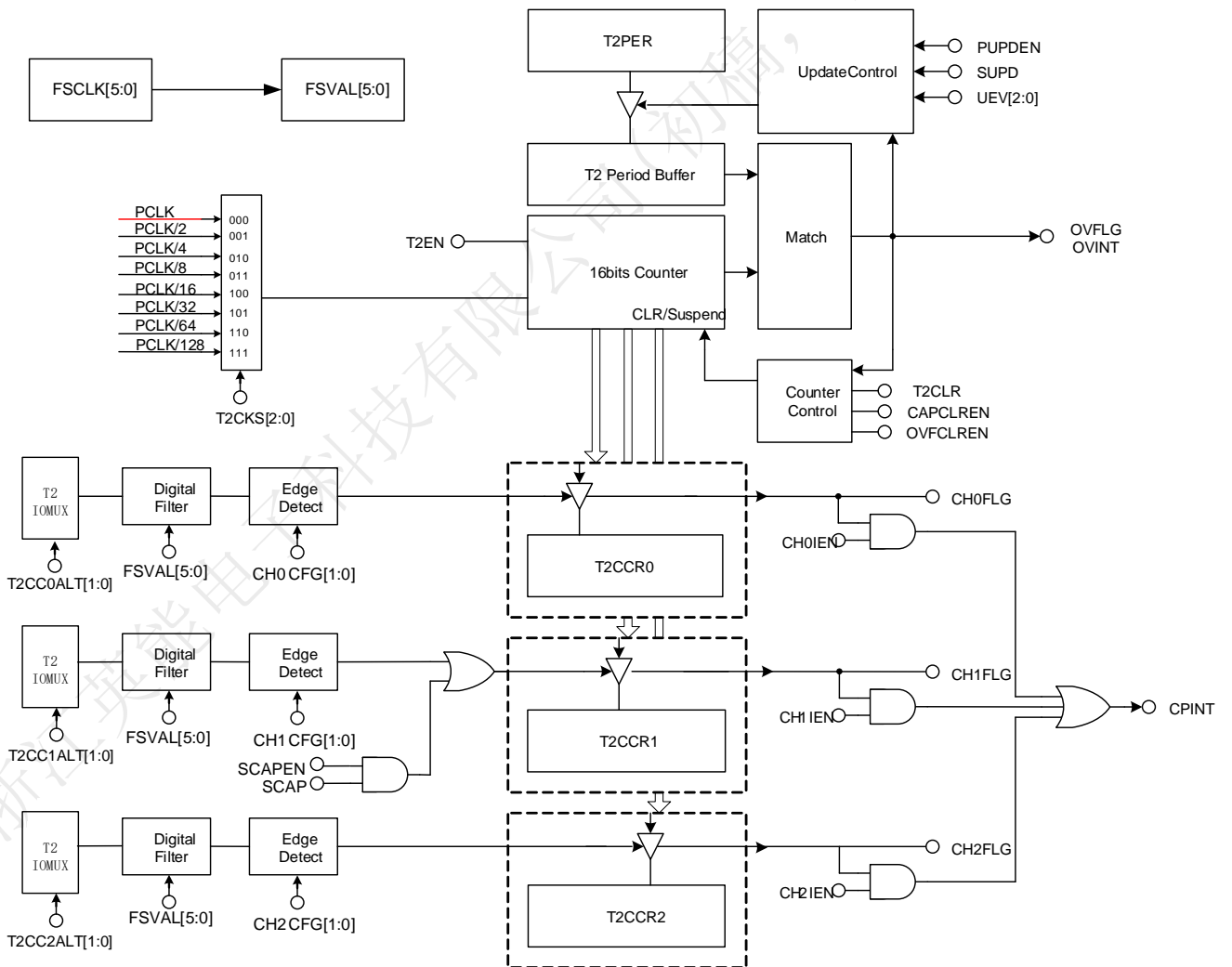


图 10-3-3 定时器 T2 捕获模式

### 模式配置、时钟源配置

当控制寄存器(T2CON)的 MODE = 1 且 CHXCEN = 0 时, 通道 0/1/2 全部配置为捕获模式。这时 3 路捕获通道与外部 GPIO 复用功能 T2CC0/1/2 互连, 也可以与内部比较器输出口 CMP00/CMP10/CMP20 互连。

控制寄存器(T2CON)的 T2CLKS 位, 用于配置计数时钟源。计数时钟源选择系统外设时钟(PCLK)预分频。

T2EN 置 1, 定时器 T2 启动工作, 计数器从 0 开始计数。

#### ● 计数器清零

捕获事件或周期溢出事件发生时, 根据 CAPCLREN 和 OVFLREN 的不同配置, 计数器将会选择清零、暂停计数或继续向上计数。

	CAPCLREN	OVFLREN	捕获事件	溢出事件
模式1	0	0或1	计数器不清零, 继续向上计数	计数器清零, 并重新从0计数
模式2	1	0	计数器清零, 并重新从0计数	计数器暂停计数, 但不清零, 直到发生新的捕获, 计数器清零并重新计数
模式3	1	1	计数器清零, 并重新从0计数	计数器清零, 并重新从0计数

#### ● 捕获中断

捕获模式下, 当捕获通道 0/1/2 出现信号翻转时, 捕获中断标志(CH0FLG /CH1FLG /CH2FLG)将会被置 1。如果此时相应通道的捕获中断使能(CH0IEN/CH1IEN/CH2IEN = 1), 将向 CPU 发出捕获中断请求。捕获中断标志 CH0FLG/CH1FLG/CH2FLG 需要软件清零。

#### ● 周期溢出中断

捕获模式下, 计数器与周期寄存器(T2PER)相等时, 周期溢出中断标志(OVFLG)将会被置 1。并向 CPU 发出溢出中断请求。溢出中断标志(OVFLG)只能通过软件写 1 清除。

#### ● 软件触发捕获功能

捕获模式使能(SCAPEN = 1)时, 可以通过软件对 SCAP 置 1 触发捕获, 来获取计数器的当前计数值。寄存器 T2CON 的 SCAP 置 1, 单次触发捕获事件, 计数器的当前计数值存入寄存器 T2CCR1, 捕获完成后, SCAP 标志自动清零。

### 10.3.4. 事件更新

周期寄存器、通道 0 / 1/2 的捕获/比较寄存器都做了缓冲处理, 触发缓冲更新的事件发生时, 配置在以上 4 个寄存器的值才会更新到缓冲寄存器中。更新事件可以选择软件触发更新、周期溢出更新、比较匹配更新。寄存器 T2CON 的 SUPD、PUPDEN、UEV[2: 0]可以对更新方式进行配置。

### 10.3.5. 捕获滤波

捕获输入信号有时候会受到严重电磁干扰, 定时器 T2 可以对捕获输入信号进行了数字滤波处理, 通过配置寄存器进行配置。FSCLK 位选择滤波采样时钟分频系数。FSCLK = 0, 每个系统外设时钟周期(PCLK)采样一次; FSCLK = 1, 2 个系统外设时钟周期(PCLK)采样一

次；FSCLK = 2，4 个系统外设时钟周期(PCLK)采样一次；FSCLK = 3，16 个系统外设时钟周期(PCLK)采样一次。

捕获信号源的数字滤波原理如下：如果 T2CC0/1/2 滤波前的信号为高电平，则“采样计数器”值加 1；如果 T2CC0/1/2 滤波前的信号为低电平，则“采样计数器”值减 1。“采样计数器”的最大值为 FSVAL，“采样计数器”的最小值为 0。当“采样计数器”值计数到 FSVAL 时，T2CC0/1/2 滤波后的信号(CAPDT0/1/2)输出高电平；当“采样计数器”值减到 0 时，T2CC0/1/2 滤波后的信号(CAPDT0/1/2)输出低电平；当“采样计数器”值介于 0 和 FSVAL 之间时，CAPDT0/1/2 的输出电平保持不变。

### 10.3.6. 中断源

定时器 T2 有四个中断源(T2CC0/1/2 的比较匹配中断/捕获中断，计数器周期溢出中断)，分别对应两个中断向量。

通道 T2CC0/1/2 的比较匹配中断/捕获中断由寄存器 T2ICON 的 bit0~bit2 使能。中断状态可以查询寄存器 T2ICON 的 bit4~bit6 中断标志位，中断标志位写 1 清除，不支持硬件自动清除。

计数器周期溢出中断的中断标志不支持硬件自动清除，需要由软件清除。

## 11. 定时器(RT)

内部集成一个 RT 定时器，可用于精确定时。

该定时器主要有以下特性：

- 使用内部 RCL 作为计数时钟
- 内置 16 位计数器
- 支持定时溢出中断，产生定时功能
- 在停机模式下，如果选择 RCL 作为时钟源，可以在规定的时间内唤醒 STOP 模式

### 11.1. 系统控制寄存器(SYSCON:0x4000E000)

描述位	名称	读写	描述	复位值
0	SYSCCLKS	R/W	见5.2.2章节	
1	CSSSEN	R/W		
4: 2	FCLK_CFG [2: 0]	R/W		
7: 5	PCLK_CFG [2: 0]	R/W		
10: 8			保留	
11	STOP_CFG	R/W	见5.3.1章节	
12	WKEN_EINT0	R/W		
13	WKEN_EINT1	R/W		

描述位	名称	读写	描述	复位值
14	WKEN_EINT2	R/W		
15	WKEN_EINT3	R/W		
16	WKEN_LVD	R/W		
17	WKEN_RT	R/W		
18	RTEN	R/W	RTimer定时使能寄存器。 0: 禁止 1: 使能	1
21: 19	RT_CFG[2: 0]	R/W	RT_CFG[2: 0]: RTimer定时溢出控制寄存器。 000: 计时到0xFFFF溢出 001: 计时到0x7FFF溢出 010: 计时到0x3FFF溢出 011: 计时到0x1FFF溢出 100: 计时到0x0FFF溢出 101: 计时到0x07FF溢出 110: 计时到0x03FF溢出 111: 计时到0x01FF溢出	000
23: 22	PGCLK_CFG [1: 0]	R/W	见26.2.1章节	
24	PGCLKEN	R/W		
25	DEBUG_ MODE	R/W	见28.1章节	
31: 26			保留	

## 11.2. 系统状态寄存器(SYSSTA:0x4000E024)

描述位	名称	读写	描述	复位值
0	PORFLG	R	见5.6.1章节	0
1	LVRFLG	R	见5.6.1章节	0
2	EXTFLG	R	见5.6.1章节	0
3	RTFLG	R	RT定时器溢出标志位。对RTFLG_CLR写1清零。 0: 未发生溢出 1: 发生溢出	0
4	LVDFLG	R	见5.3.1.4章节	0
5	CRY_ STOPDET	R	见5.2.2.4章节	0
6	PLL_LOCK	R	见5.2.2.4章节	0
7	LVD	R	见5.3.1.4章节	0
8	WTDFLG	R	见5.6.1章节	0

描述位	名称	读写	描述	复位值
15: 9			保留	
16	PORFLG_CLR	W	见5.6.1章节	0
17	LVRFLG_CLR	W	见5.6.1章节	0
18	EXTFLG_CLR	W	见5.6.1章节	0
19	RTFLG_CLR	W	写1操作，产生清零脉冲，清除RTFLG标志位。写0无效。	0
20	LVDFLG_CLR	W	见5.3.1.4章节	0
23: 21			保留	0
24	WTDFLG_CLR	W	见5.6.1章节	0
31: 25			保留	

## 12. 看门狗(WDOG)

WATCHDOG 模块是一个基于 32 比特向下计数的计数器，可以通过 WDOGLOAD 设置初始器；当计数器第一次从设置的 WDOGLOAD 值计到 0 时，会产生一个看门狗中断计数器会重新从看门狗的载入值计数；当计数器第二次从 WDOGLOAD 计到 0 时，会产生一个看门狗复位；当处于 DEBUG 状态，暂停运行，WDT 计数器停止计数。

### 12.1. 模块寄存器总表

寄存器名	地址	结构体(.h)	寄存器描述	复位值
WDOGLOAD	0x4000C000	WDOG	看门狗的载入值。	0xFFFFFFFF
WDOGVALUE	0x4000C004	WDOG	当前看门狗计数器的值。	0xFFFFFFFF
WDOG CONTROL	0x4000C008	WDOG	看门狗控制。	0x00
WDOGINTCLR	0x4000C00C	WDOG	看门狗中断清除。	0x00000000
WDOG MIS	0x4000C014	WDOG	看门狗中断标志位。	0x00
WDOGLOCK	0x4000CC00	WDOGLOCK	看门狗寄存器写使能。	0x00000000

### 12.2. 寄存器说明

#### 12.2.1. WDOG 计数值载入寄存器 (WDOGLOAD: 0x4000C000)

描述位	名称	读写	描述	复位值
31: 0	WDOGLOAD	R/W	看门狗的载入值。	0xFFFFFFFF

看门狗的载入值，看门狗每次都从这个值开始向下计数；当该寄存器被写入后，看门狗计数器马上重新载入这个值，然后开始计数；该寄存器的最小的值为 1。

### 12.2.2. WDOG 当前计数值寄存器 (WDOGVALUE: 0x4000C004)

描述位	名称	读写	描述	复位值
31: 0	WDOGVALUE	R	当前看门狗计数器的值。	0xFFFFFFFF

### 12.2.3. WDOG 控制寄存器 (WDOGCONTROL: 0x4000C008)

描述位	名称	读写	描述	复位值
0	INTEN	R/W	看门狗中断和计数使能位。 0: 禁止看门狗中断, 关闭计数器; 1: 使能看门狗中断, 使能计数器;	0
1	RESEN	R/W	看门狗复位使能位。 0: 禁止看门狗复位; 1: 使能看门狗复位;	0
7: 2			保留	

### 12.2.4. WDOG 中断清除寄存器 (WDOGINTCLR: 0x4000C00C)

描述位	名称	读写	描述	复位值
31: 0	WDOGINTCLR	W	写入任何值都可以清除看门狗中断, 同时看门狗计数器重新从WDOGLOAD寄存器值开始计数。	

### 12.2.5. WDOG 中断标志寄存器(WDOGMIS: 0x4000C014)

描述位	名称	读写	描述	复位值
0	WDOGMINT	R	看门狗中断标志位。 0: 看门狗未产生中断 1: 看门狗产生中断	0
7: 1			保留	

### 12.2.6. WDOG 写使能寄存器 (WDOGLOCK: 0x4000CC00)

描述位	名称	读写	描述	复位值
31: 0	LOCK	W	看门狗寄存器写使能密钥。 写: 0x1ACCE551: 开启寄存器写功能; 其他 : 关闭寄存器写功能;	0x00000000

描述位	名称	读写	描述	复位值
			读: 0: 对其他所有的寄存器写是有效; 1: 对其他所有的寄存器写都是无效。	

## 12.3.看门狗应用

### 12.3.1. 示例代码

```
WDOGLOCK->LOCK=0x1ACCE551;
WDOG->WDOGCONTROL_b.INTEN = 1;
WDOG->WDOGLOAD = 16000000;
WDOGLOCK->LOCK=0x00000000;
```

```
WDOGLOCK->LOCK=0x1ACCE551;
WDOG->WDOGCONTROL_b.RESEN = 1;
WDOGLOCK->LOCK=0x00000000;
```

### 12.3.2. 看门狗载入值对应复位时间

看门狗溢出复位时间  $T = (2 * WDOGLOAD / PCLK)S$ 。

## 13. 编码器检测器(EQEP)

### 13.1. 模块寄存器总表

寄存器名	地址	结构体(.h)	寄存器描述	复位值
QEPCTL	0x40002800	EQEP	QEP控制寄存器	0x00000000
QPOSINIT	0x40002804	EQEP	初始值寄存器	0x0000
QPOSMAX	0x40002808	EQEP	最大值寄存器	0x0000
QPOSCMP	0x4000280C	EQEP	捕获比较寄存器	0x0000
QPOSLAT	0x40002810	EQEP	定时器触发锁存寄存器	0x0000
QPOSILAT	0x40002814	EQEP	Index触发锁存寄存器	0x0000
QPOSSLAT	0x40002818	EQEP	Strobe触发计数锁存寄存器	0x0000

QPOSCNT	0x4000281C	EQEP	计数寄存器	0x0000
QEPSTS	0x40002820	EQEP	状态寄存器	0x0000
QEPICTL	0x40002824	EQEP	中断控制和状态位清0	0x0000

## 13.2. 寄存器说明

### 13.2.1. 控制寄存器(QEPCTL: 0x40002800)

描述位	名称	读写	描述	复位值
0	QAP	R/W	QEPA极性选择。 0: 正相 1: 反相	0
1	QBP	R/W	QEPB极性选择。 0: 正相 1: 反相	0
2	SWAP	R/W	QEPA、QEPB交换模式。 0: QEPA作为clock, QEPB作为方向 1: QEPB作为clock, QEPA作为方向	0
3	QIP	R/W	QEPI极性选择。 0: 正相 1: 反相	0
4	QSP	R/W	QEPS极性选择。 0: 正相 1: 反相	0
5	QIGATE	R/W	QI门控开关。 0: QI直通 1: QS=1, QI输入开启; QS=0, QI输入关闭	0
7: 6	QSRC[1: 0]	R/W	QEP计数模式选择。 00: 正交计数模式 01: 直接计数模式 10: 向上计数模式 11: 向下计数模式	00
8	XCR	R/W	计数时钟配置。 0: QEPA 信号上升沿计数	0

描述位	名称	读写	描述	复位值
			1: QEPA 信号上升沿或下降沿计数(2倍频率)	
9	SWI	R/W	软件触发计数器复位。 0: 不触发计数器初始化 1: 触发计数器初始化, 硬件自动清0	0
11: 10	IEI[1: 0]	R/W	Index event 复位配置。 00: 不触发计数器初始化 01: 允许SWI触发计数器初始化 10: Index上升沿触发计数器初始化 11: Index下降沿触发计数器初始化	00
13: 12	SEI[1: 0]	R/W	Strobe event 复位配置。 0X: 不触发计数器初始化 10: Strobe 上升沿触发计数器初始化 11: Strobe 上升沿(加1)或下降沿(减1)触发计数器初始化	00
14	CMPEN	R/W	比较匹配开关控制, 高有效	0
15	QEPEN	R/W	QEP模块开关控制, 高有效	0
17: 16	TS[1: 0]	R/W	定时器输入源选择。 00: T0计数溢出 01: T1计数溢出 10: T2计数溢出 11: BasicTimer溢出	00
19: 18	QPCRM[1: 0]	R/W	计数器复位模式。 00: Index事件触发计数器复位 01: 计数到QPOS MAX时触发计数器复位 10: 第一个Index事件发生时, 位置计数器复位 11: t0_ovf、t1_ovf、t2_ovf触发计数器复位	00
21: 20	IEL[1: 0]	R/W	Index event 锁存配置。 00: 不触发计数器锁存 01: Index上升沿触发计数器锁存 10: Index下降沿触发计数器锁存 11: Index标识事件/软件触发计数器锁存	00
22	SWL	R/W	软件触发计数器锁存。 0: 不触发计数器锁存 1: 触发计数器锁存, 硬件自动清0	0
23	SEL	R/W	Strobe event 锁存配置。 0: Strobe 上升沿触发计数器锁存 1: Strobe 上升沿(加1)或下降沿(减1)触发计数器锁存	0

描述位	名称	读写	描述	复位值
25: 24	FILTER[1: 0]	R/W	FILTER[1: 0]: 消抖时间设置。 00: 关闭消抖功能 01: 消抖时间为3个系统外设时钟(PCLK) 10: 消抖时间为7个系统外设时钟(PCLK) 11: 消抖时间为15个系统外设时钟(PCLK)	00
31: 26			保留	

### 13.2.2. 初始值寄存器(QPOSINIT: 0x40002804)

描述位	名称	读写	描述	复位值
15: 0	QPOSINIT [15: 0]	R/W	QEP初始值寄存器。	0x0000

### 13.2.3. 最大值寄存器(QPOSMAX: 0x40002808)

描述位	名称	读写	描述	复位值
15: 0	QPOSMAX [15: 0]	R/W	QEP最大值寄存器。	0x0000

### 13.2.4. 比较值寄存器(QPOSCMP: 0x4000280C)

描述位	名称	读写	描述	复位值
15: 0	QPOSCMP [15: 0]	R/W	QEP匹配比较寄存器。	0x0000

### 13.2.5. 锁存寄存器(QPOSLAT: 0x40002810)

描述位	名称	读写	描述	复位值
15: 0	QPOSLAT [15: 0]	R	QEP计数锁存寄存器, 由定时器溢出触发。	0x0000

### 13.2.6. 锁存寄存器(QPOSILAT: 0x40002814)

描述位	名称	读写	描述	复位值
15: 0	QPOSILAT [15: 0]	R	QEP计数锁存寄存器, 由index信号触发。	0x0000

### 13.2.7. 锁存寄存器(QPOSSLAT: 0x40002818)

描述位	名称	读写	描述	复位值
15: 0	QPOSSLAT [15: 0]	R	QEP计数QPHE锁存寄存器,由Strobe信号触发	0x0000

### 13.2.8. 计数寄存器(QPOSCNT: 0x4000281C)

描述位	名称	读写	描述	复位值
15: 0	QPOSCNT [15: 0]	R	QEP计数寄存器	0x0000

### 13.2.9. 状态寄存器(QEPSTS: 0x40002820)

描述位	名称	读写	描述	复位值
0	QPHE	R	QEPA、QEPB相位错误标志,同时跳变。	0
1	FIMF	R	第一个Index标志位	0
3: 2	FIPH[1: 0]	R	第一个Index触发时, QEPA、QEPB的相位。	00
4	FIDF	R	第一个Index触发时的计数方向。	0
5	QDIRLAT	R	Index、Strobe信号触发的方向锁存位。	0
6	QDIR	R	实时的计数方向标志位。	0
7	TIMER_OVF	R	t0_ovf、t1_ovf、t2_ovf标志位。	0
8	PCEF	R/W	计数错误中断标志位。 对QEPICTL的PCEF写1清0	0
9	OVF	R/W	计数溢出中断标志位。 对QEPICTL的OVF对应位写1清0	0
10	MTF	R/W	计数匹配中断标志位。 对QEPICTL的MTF对应位写1清0	0
11	INXF	R/W	Index标识事件中断标志位。 对QEPICTL的INXF对应位写1清0	0
12	PHEF	R/W	相位错误中断标志位。 对QEPICTL的PHEF写1清0	0
13	IELF	R/W	Index锁存的中断标志位。 对QEPICTL的IELF写1清0	0
14	SELF	R/W	Strobe锁存的中断标志位。 对QEPICTL的SELF写1清0	0
15	QDCF	R/W	计数器方向改变的中断标志位。 对QEPICTL的QDCF写1清0	0

### 13.2.10. 中断控制寄存器(QEPICTL: 0x40002824)

描述位	名称	读写	描述	复位值
0	PCEF_EN	R/W	计数错误中断使能, 高有效	0
1	OVF_EN	R/W	计数溢出中断使能, 高有效	0
2	MTF_EN	R/W	计数匹配中断使能, 高有效	0
3	INXF_EN	R/W	Index事件中断使能, 高有效	0
4	PHEF_EN	R/W	相位错误中断使能, 高有效	0
5	IELF_EN	R/W	Index锁存的中断使能, 高有效	0
6	SELF_EN	R/W	Strobe锁存的中断使能, 高有效	0
7	QDCF_EN	R/W	计数器方向改变的中断使能, 高有效	0
8	PCEF	R/W	计数错误中断标志位写1清0	0
9	OVF	R/W	计数溢出中断标志位写1清0	0
10	MTF	R/W	计数匹配中断标志位写1清0	0
11	INXF	R/W	Index事件中断标志位写1清0	0
12	PHEF	R/W	相位错误中断标志位写1清0	0
13	IELF	R/W	Index锁存的中断标志位写1清0	0
14	SELF	R/W	Strobe锁存的中断标志位写1清0	0
15	QDCF	R/W	计数器方向改变的中断标志位写1清0	0

### 13.3.功能说明

EQEP 模块有 4 个输入管脚, 分别是 QEPA/XCLK、QEPB/XDIR、QEPI 和 QEPS。其中前两个引脚被使用在正交时钟模式或者直接计数模式。QEP 包含正交解码单元(QDU)和位置计数器和位置计算控制单元(PCCU)。

- 正交时钟模式

正交编码器提供两路相位差为 90 度的脉冲, 相位关系决定了电机旋转方向信息, 脉冲的个数可以决定电机的绝对位置信息。超前或者顺时钟旋转时, A 路信号超前 B 路信号, 滞后或者逆时针旋转时, B 路信号超前 A 路信号。正交编码器使用这两路输入引脚可以产生正交时钟和方向信号。

- 直接计数模式

在直接计数模式中, 方向和时钟信号直接来自外部, 此时 QEPA 引脚提供时钟输入, QEPB 引脚提供方向输入。

后两个引脚 QEPI 是索引或者起始标记脚, 正交编码器使用索引信号来确定一个绝对的

起始位置，此引脚直接与正交编码器的索引输出端相连，当此信号到来时，可以将位置计数器复位清零，也可以初始化或者锁存位置计数器的值。

QEPS 是锁存输入引脚，锁存引脚输入的主要作用是当规定时间信号到来时，初始或者锁存位置计数器的值，该引脚通常和传感器或者限制开关连接，用于通知电机是否达到预定位置。

### 13.3.1. 正交解码单元(QDU)

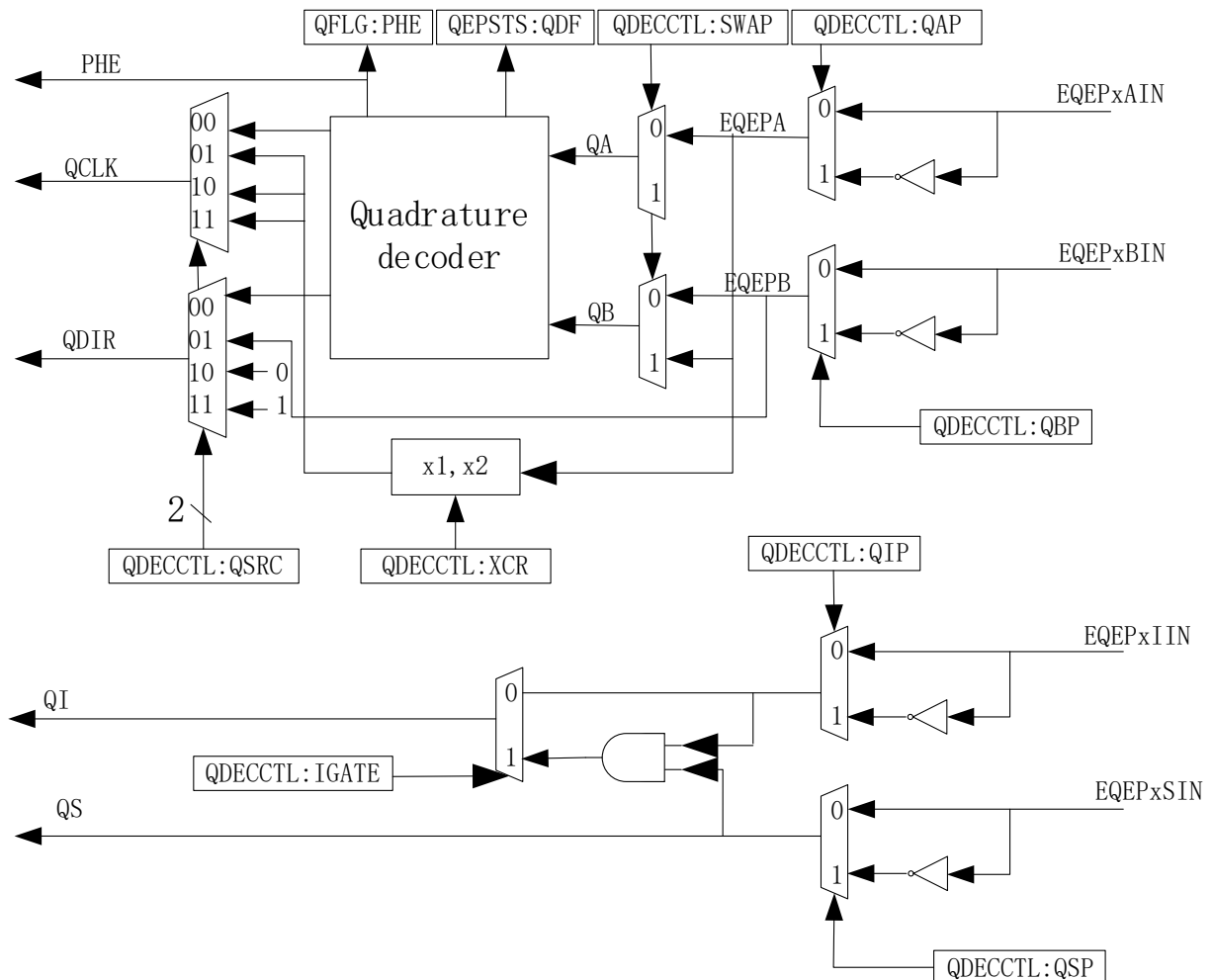


图 13-3-1-1 QDU 功能框图

QEP 输入极性选择，每个 QEP 输入都可以通过 QDECCTL 寄存器来配置极性。正交解码单元产生位置计数器时钟信号和方向信号，位置计数器的计数模式由 QDECCTL

寄存器中的 QSRC 位决定，主要有如下 4 种计数模式：

正交计数模式

在正交-计数模式下，正交解码器产生方向信号和时钟信号送给位置计数器。通过确定 QEPA 和 QEPB 两个脉冲信号哪一个超前，来解码出旋转方向逻辑，并且将此方向逻辑更新到 QEPSTS 的 QDIR 位。下图是方向解码逻辑的真值表以及状态机。

先前边沿	当前边沿	QDIR	QPOSCNT
QA ↑	QB ↑	上	递增
	QB ↓	下	递减
	QA ↓	翻转	递增或递减
QA ↓	QB ↓	上	递增
	QB ↑	下	递减
	QA ↑	翻转	递增或递减
QB ↑	QA ↑	下	递增
	QA ↓	上	递减
	QB ↓	翻转	递增或递减
QB ↓	QA ↓	下	递增
	QA ↑	上	递减
	QB ↑	翻转	递增或递减

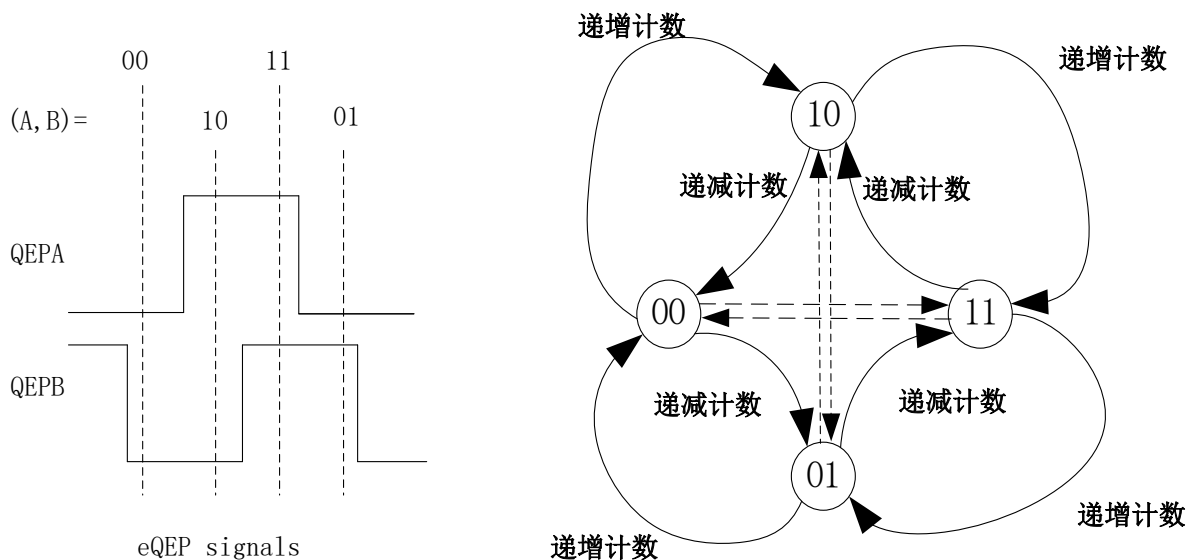


图 13-3-1-2 正交解码状态机

QEP 还有相位错误检测机制，当 QEPA 和 QEPB 信号同步到来时，QEP 中的

相位错误标志会置位，而且会申请中断。QEP 中的解码模块会对 QEPA 和 QEPB 脉冲的上升沿以及下降沿进行计数，所以最后解码的时钟频率将会是实际输入 QEPA 或者 QWPB 的 4 倍。

相位错误标识: 在正常操作条件下，正交输入 QEPA 与 QEPB 在相位上相差 90 度，边沿信号是不会同时到达的，当同时检测到两者的边沿信号时，那么 QFLG 寄存器的相位错误标志位(PHE)将置位。上图中状态机的虚线就表示产生的相位错误的可能。

反向计数: 在正常正交计数模式操作时，QEPA 输入送到正交解码器的 QA 输入，QEPB 输入送到正交解码器的 QB 输入。反向计数通过设置 QDECCTL 寄存器的 SWAP 位被使能，此时正交解码器的输入取反，从而计数方向也取反。

### 1. 直接计数模式

有些位置编码器提供了方向和时钟输出代替正交输出，这种情况，可以使用直接计数模式。此时 QEPA 只能作为时钟输入，QEPB 只能作为方向输入。如果方向为高时，那么计数器会在 QEPA 输入的上升沿时递增计数，当方向为低时，那么计数器会在 QEPA 输入的上升沿自动递减计数。

### 2. 向上计数模式

计数器的方向信号被硬件规定为递增计数，此时位置计数器根据 QDECCTL 中的 XCR 位规定，对 QEPA 信号计数或者 2 倍关系计数。

### 3. 向下计数模式

计数器的方向信号被硬件规定为递减计数，此时位置计数器根据 QDECCTL 中的 XCR 位规定，对 QEPA 信号计数或者 2 倍关系计数。

## 13.3.2. 位置计数和控制单元(PCCU)

控制单元通过控制寄存器 QEPCTL 设置位置计数器工作模式、位置计数器初始化/锁存模式和比较匹配逻辑。工作过程中会产生各种事件，并触发 MCU 中断。

### 13.3.2.1. 计数工作模式

位置计数有 4 种工作模式，用于控制计数器的计数溢出、复位和位置计数锁存。

- 在索引事件到来时，位置计数器复位。
- 在计数值达到设定的最大值时，位置计数器复位。
- 第一个索引事件到来时，位置计数器复位。
- 在定时事件到达时，位置计数器复位。

上述所有的操作模式，位置计数器都会在上溢时复位到 0，在下溢时复位到 QPOS MAX 最大值。

1. 在索引事件到来时，位置计数器复位，当前计数值和计数方向锁存到寄存器。

如果索引事件发生在正向运动时，那么位置计数器在下一个 QEP 时钟复位到 0，如果索引事件发生在反向运动时，那么位置计数器在下一个 QEP 时钟复位到 QPOS MAX 最大设定值。

在每次索引事件发生时，位置计数器的值锁存到 QPOSILAT 寄存器中，方向信号更新到 QEPSTS[QDLF]中。如果锁存的值不是 0、QPOSINIT 或者 QPOS MAX 值时，位置计数错误的中断标志会置 1。在索引事件被标识时，位置计数器错误中断标志(QEPSTS[PCEF])被更新，只有通过软件才能清 0 这个错误标志位。

在这种模式下，索引事件锁存配置 QEPCTL[IEL]位被忽略，位置计数器错误中断标志只有在该模式下产生。

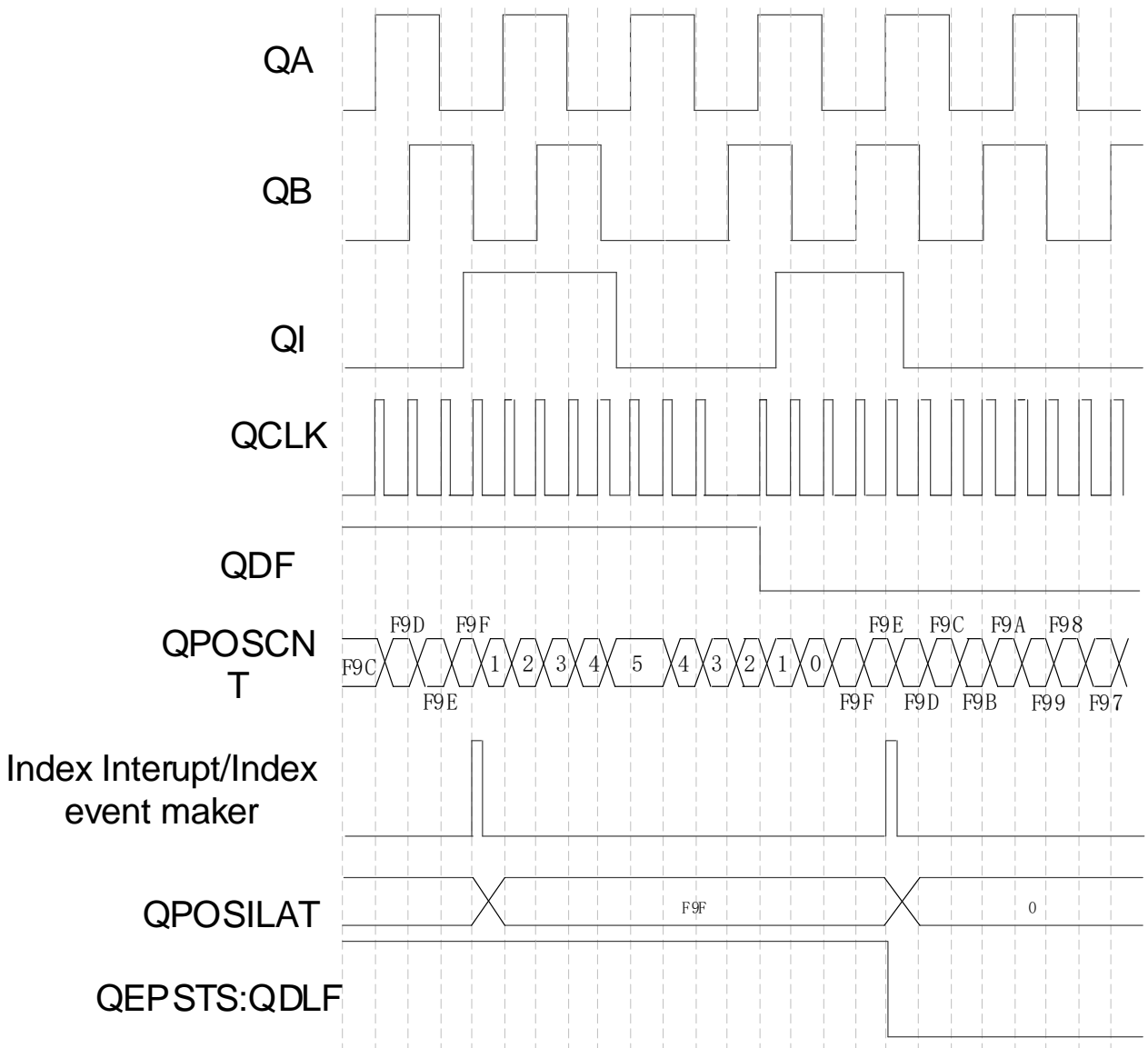


图 13-3-2-1 Index 事件触发位置计数器复位

2. 在计数值达到设定的最大值时，位置计数器复位。

如果位置计数器达到 **QPOSMAX** 时，那么在正向运动过程中，位置计数器在下一个 **QEP** 时钟时置 0，而且位置计数器溢出标识位置位；如果位置计数器等于 0，而且在反向运动的条件下，那么位置计数器在 **QEP** 的下一个时钟复位到 **QPOSMAX**，且位置计数器溢出标识位置位，如下图所示：

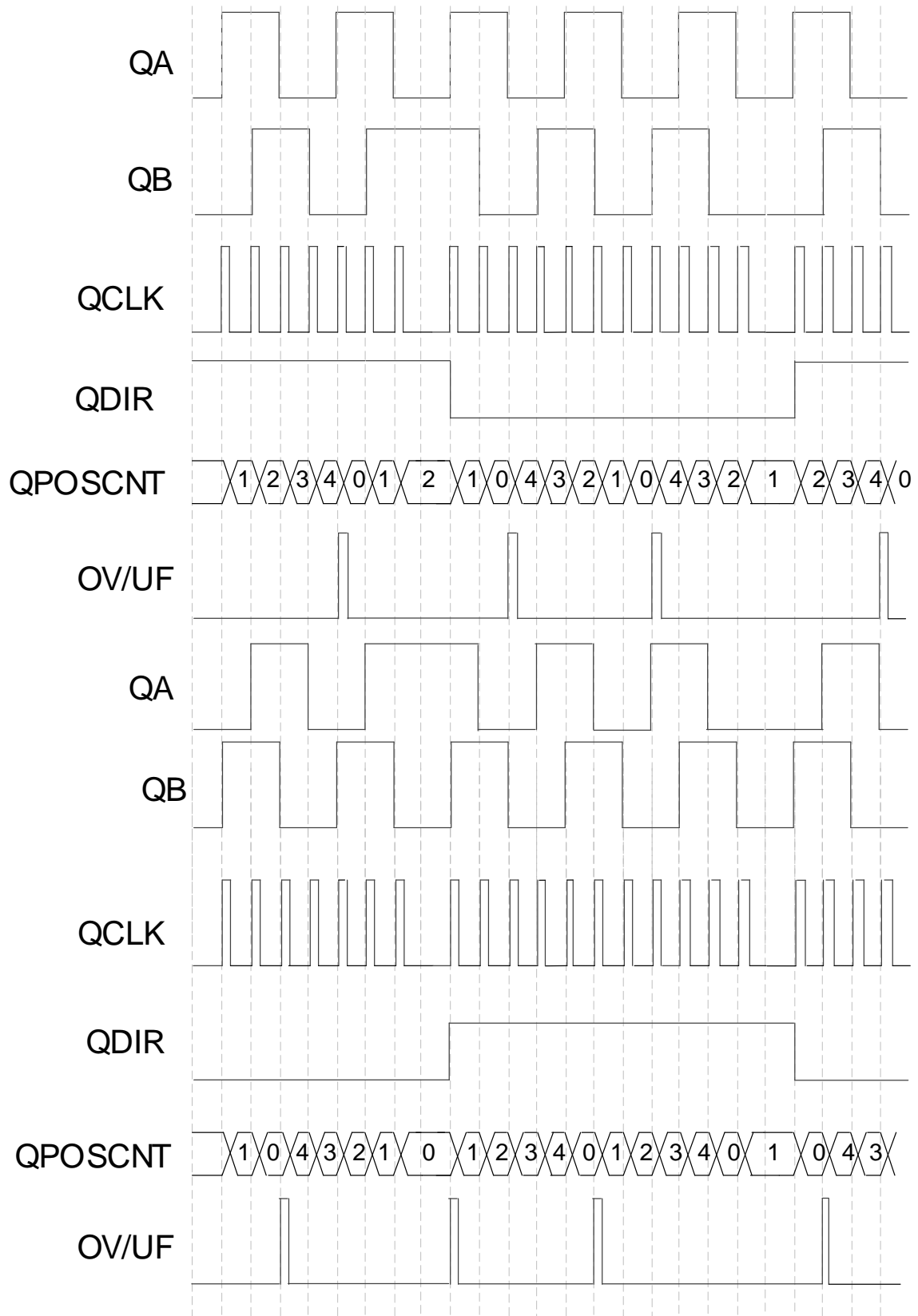


图 13-3-2-2 计数值达到设定的最大值时位置计数器复位

3. 第一个索引事件到来时，位置计数器复位到初始值。

如果第一个索引事件发生在正向运动时，位置计数器会在下一个 QEP 时钟复位为 0；如果第一个索引事件发生在反向运动时，位置计数器会在下一个 QEP 时钟复位为 QPOSMAX。注意只有在第一个索引事件发生时，位置计数器才会复位，而后面的索引事件不会使位置计数器复位。

第一个索引标识为在第一个索引边沿后的正交边沿。EQEP 外设记录了第一个索引标识 (QEPSTS[FIMF]) 的发生以及第一个索引事件标识方向 (QEPSTS[FIDF])，它也记录了第一个索引标识的正交边沿，从而可使这个相关的正交转换用于索引事件的复位操作。比如在正向运动过程中，第一次复位操作发生在 QEPB 的下降沿，那么所有后来的复位必须分配在正向运动中的 QEPB 的下降沿或者反向运动中的 QEPB 的上升沿。

4. 在定时事件到达时，位置计数器复位到初始值，当前计数值和计数方向锁存到寄存器。

在此种模式下，事件发生时 QPOSCNT 的值会锁存到 QPOSLAT 寄存器中，然后位置计数器会复位到 0 或者 QPOSMAX，取决于计数方向。这种模式对测量频率很有效。

QEPCTL[TS]选择 T0、T1、T2 的计时溢出作为定时事件。

### 13.3.2.2. 位置计数器锁存

QEP 的 Index 和 Strobe 事件发生时，将位置计数器的值锁存到 QPOSILAT 和 QPOSSLAT 中。

#### 1. Index 事件锁存

在某些应用中，并不要求在每次索引事件到来时复位位置计数器 (PCRM=01 和 PCRM=10) 下，在这种情况下，位置计数器可以在如下事件锁存，并且在每次索引事件标识时方向信息被记录到 QEPSTS[QDLF] 中。

##### 1) 上升沿锁存 (QEPCTL[IEL]=01)

在输入索引事件的每次上升沿到来时，位置计数器值 (QPOSCNT) 锁存到 QPOSILAT 寄存器中。

##### 2) 下降沿锁存 (QEPCTL[IEL]=10)

在输入索引事件的每次下降沿到来时，位置计数器值 (QPOSCNT) 锁存到 QPOSILAT 寄存器中。

##### 3) 索引事件标识锁存 (QEPCTL[IEL]=11)

利用第一个索引标识，在第一个索引边沿后记录的正交边沿 (QEPSTS[FIPH]) 和方向

(QEPSTS[FIDF]) 检测索引事件，在匹配的正交转换位置将计数器值 (QPOSCNT) 锁存到 QPOSILAT 寄存器中。

这个作为错误检查机制很有用，可以用来检查位置计数器在索引事件之间能否正确累加。比如，1000 线的编

码器当按照相同方向运动时，在两个索引事件之间应当有 4000 个脉冲，进行累加 4000

次。

位置计数器的值锁存到 QPOSILAT 寄存器中时，索引事件中断标志位 (QEPSTS[IELF])被置位。

索引事件触发的位置计数器锁存时序图如下：

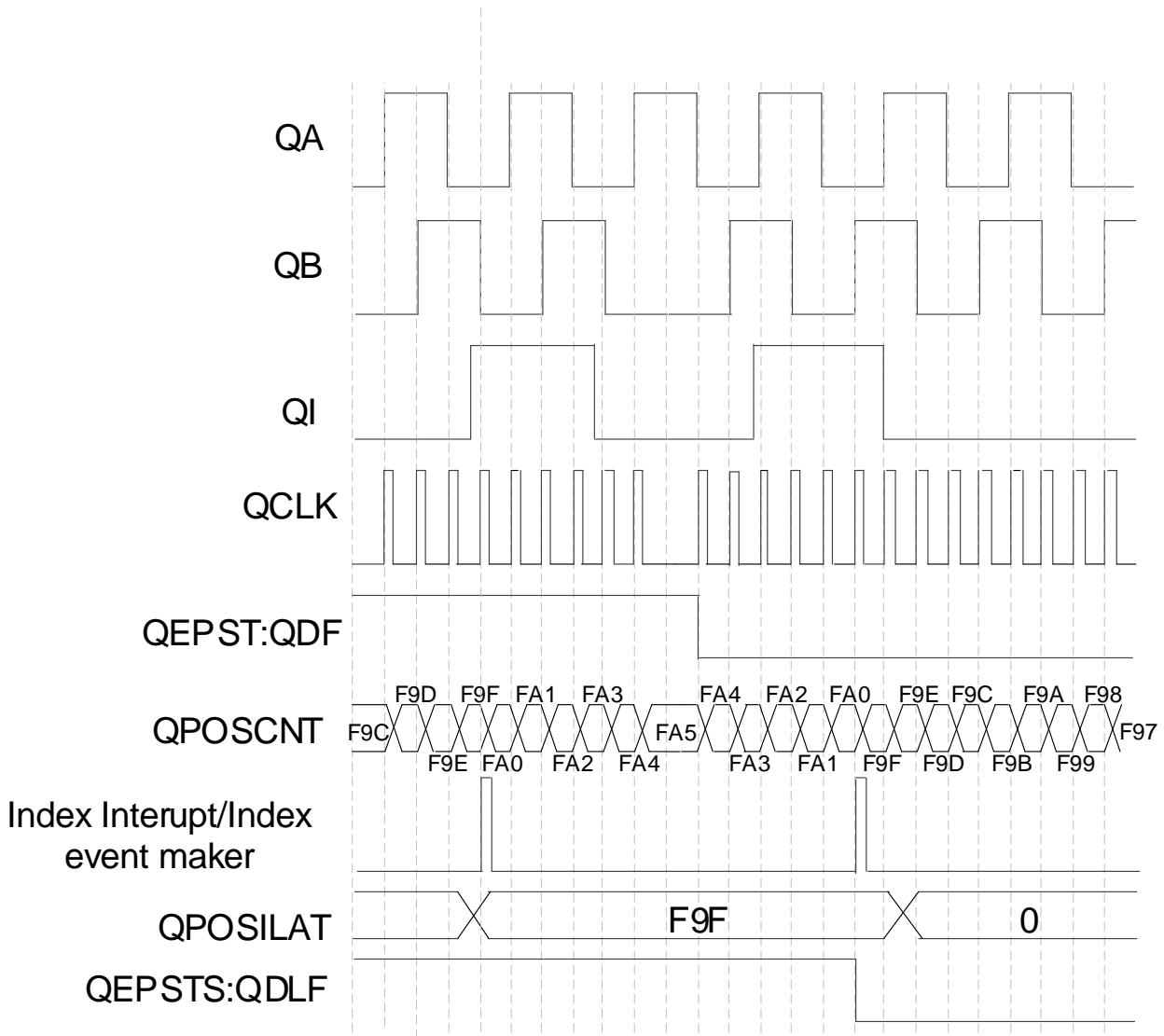


图 13-3-2-3 索引事件触发位置计数器锁存

## 2. Strobe 事件锁存

当位置计数器的值锁存到 QPOSSLAT 寄存器后，Strobe 事件锁存中断标志 (QEPSTS[SELF])被置位。

1) 上升沿锁存(QEPCTL[SEL]=0)

位置计数器的值在 Strobe 信号的上升沿被锁存到 QPOSSLAT 寄存器中

2) 下降沿锁存(QEPCTL[SEL]=1)

位置计数器的值在正向运动时，Strobe 信号的上升沿进行锁存；反向运动时，Strobe 信号的下降沿进行锁存。如下图所示：

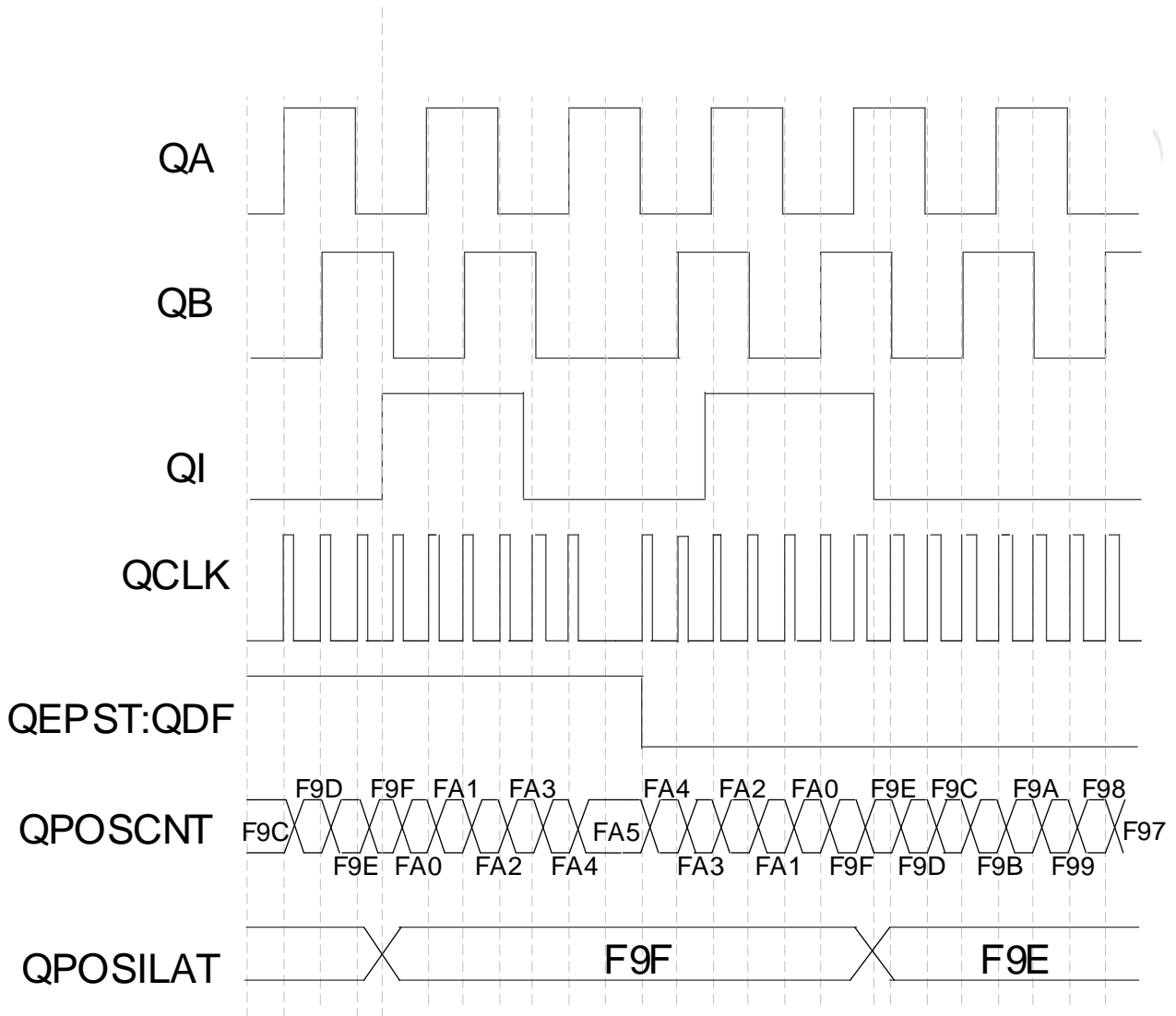


图 13-3-2-4 Strobe 事件触发位置计数器锁存

### 13.3.2.3. 位置计数器初始化

位置计数器可以由以下事件进行初始化：

- Index 事件
- Strobe 事件
- 软件初始化

#### 1. Index 事件初始化

在索引输入上升沿(QEPCTL[IEI]=10)或者下降沿(QEPCTL[IEI]=11)的时候可以触发位置计数器的初始化为寄存器 QPOSINIT 的值。

## 2. Strobe 事件初始化

如果 QEPCTL[SEI]=10, 那么位置计数器在 Strobe 输入上升沿时会初始化为寄存器 QPOSINIT 的值。

如果 QEPCTL[SEI]=11, 位置计数器的值在正转时会在 Strobe 输入的上升沿初始化为 QPOSINIT 的值, 否则在反转时, 在 Strobe 输入的下降沿初始化。

3. 软件初始化(SWI): 将 QEPCTL(SWI)位设置为 1 时, 位置计数器由软件初始化, 而 QEPCTL(SWI)位在初始化后会自动清除。

### 13.3.2.4. 位置比较

当 QEPCTL[CMPEM]=1 时, 如果位置计数器 QPOSCNT 值与比较值寄存器 QPOSCMP 匹配, 则位置匹配中断

标志(QEPSTS[MTF])被置位。

## 14. 脉冲宽度调制器(EPWM)

16 位 PWM 波形发生器, 具有四个比较/捕获通道, 最大支持 7 路 PWM 输出, 最大支持 3 组互补输出, 支持死区时间控制, 支持故障控制。

- 具有边沿计数模式或者中央对齐模式
- 具有故障触发功能, 故障发生后可支持软故障模式(即逐周期保护)或硬故障模式
- 互补输出死区时间设置, 支持上升沿下降沿死区独立设置
- 各路 PWM 输出极性可以独立设置
- 支持 4 路捕获输入
- PWM0/1、PWM2/3、PWM4/5 可作为 3 路互补输出
- PWM6 的输出可作为 ADC 的启动触发源
- PWM\_FLT 的输入信号源可选外部 IO 输入 FLT 或 ACMP3 输出

### 14.1. 模块寄存器总表

EPWM 控制的主要寄存器列表如下:

寄存器名	地址	结构体(.h)	寄存器描述	复位值
PWMCON	0x40003800	EPWM	PWM控制寄存器	0x0000
PWMPER	0x40003804	EPWM	PWM周期寄存器	0xFFFF
DTCON	0x40003808	EPWM	PWM死区控制寄存器	0x0000
OUTCFG	0x4000380C	EPWM	PWM输出控制寄存器	0x00000000
CICOCFG	0x40003810	EPWM	捕获比较设置寄存器	0x00000000
INTCON	0x40003814	EPWM	PWM中断控制寄存器	0x00000000
FTCFG	0x40003818	EPWM	PWM故障控制寄存器	0x0000
CCR0N	0x4000381C	EPWM	通道0下降沿匹配寄存器	0xFFFF
CCR0P	0x40003820	EPWM	通道0上升沿匹配寄存器	0xFFFF

CCR1N	0x40003824	EPWM	通道1下降沿匹配寄存器	0xFFFF
CCR1P	0x40003828	EPWM	通道1上升沿匹配寄存器	0xFFFF
CCR2N	0x4000382C	EPWM	通道2下降沿匹配寄存器	0xFFFF
CCR2P	0x40003830	EPWM	通道2上升沿匹配寄存器	0xFFFF
CCR3N	0x40003834	EPWM	通道3下降沿匹配寄存器	0xFFFF
CCR3P	0x40003838	EPWM	通道3上升沿匹配寄存器	0xFFFF
TRGC	0x4000383C	EPWM	PWM缓存控制寄存器	0x0000
PWMCNT	0x40003840	EPWM	PWM计数寄存器	0x0000
PWMLOCK	0x40003844	EPWM	PWM输出极性配置锁定保护寄存器	0x00
PWMSSTA	0x40003848	EPWM	单电阻采样状态寄存器	0x00

## 14.2. 寄存器说明

### 14.2.1. 控制寄存器(PWMCON: 0x40003800)

描述位	名称	读写	描述	复位值
0	PWM_EN	R/W	PWM使能信号。 0: PWM禁止, 清零PWM计时器 1: PWM使能	0
1	STRG	R/W	写操作: 仅在单次模式有效。STRG置1, 触发计数器单次计数, 触发动作完成后, STRG自动清零。 0: 无效 1: 单次触发 读操作: 读取计数器的计数方向。边沿对齐方式时, 读取的STRG一直为0。 0: 上升计数 1: 下降计数	0
3: 2	TMOD[1: 0]	R/W	计数器模式控制。 00: 连续边沿对齐模式 01: 连续中央对齐模式 (上升奇数计数, 下降偶数计数) 10: 单次边沿对齐模式 11: 单次中央对齐模式 (上升奇数计数, 下降偶数计数)	00
4	CH0MS	R/W	通道0工作模式选择。 0: 捕获模式 1: 比较模式	0
5	CH1MS	R/W	通道1工作模式选择。	0

描述位	名称	读写	描述	复位值
			0: 捕获模式 1: 比较模式	
6	CH2MS	R/W	通道2工作模式选择。 0: 捕获模式 1: 比较模式	0
7	CH3MS	R/W	通道3工作模式选择。 0: 捕获模式 1: 比较模式	0
10: 8	CLKS[2: 0]		PWM时钟选择控制位，系统外设时钟分频后， 作为PWM时钟。 000: 不分频 001: 2分频 010: 3分频 011: 4分频 100: 5分频 101: 6分频 110: 7分频 111: 8分频	000
11	SUPD		软件更新控制位，写1更新PWM缓冲寄存器，动 作完成后，SUPD自动清零。 0: 无效 1: 更新缓冲寄存器	0
13: 12	UPD_NC[1: 0]		缓冲更新间隔。 00: 每个更新事件发生即更新缓冲寄存器 01: 更新事件发生2次时更新缓冲寄存器 10: 更新事件发生4次时更新缓冲寄存器 11: 更新事件发生8次时更新缓冲寄存器	00
15: 14	BF_UPD[1: 0]		缓冲更新控制。 00: SUPD写1时即更新 01: 过零溢出更新，仅中央对齐模式有效 10: 周期匹配更新，中央对齐模式或者边沿 对齐有效 11: 周期匹配或者过零溢出更新，中央对齐 模式下，周期和过零均有效，边沿对齐模式下， 仅周期有效	00

### 14.2.2. 周期控制寄存器(PWMPER: 0x40003804)

描述位	名称	读写	描述	复位值
15: 0	PWMPER[15: 0]	R/W	PWM周期数据寄存器	0xFFFF

### 14.2.3. 死区控制寄存器(DTCON: 0x40003808)

描述位	名称	读写	描述	复位值
7: 0	DTR[7: 0]	R/W	PWM上沿死区控制时间，死区时间为系统 外设时钟乘以DTR。	0x00
15: 8	DTF[7: 0]	R/W	PWM下沿死区控制时间，死区时间为系统 外设时钟乘以DTF。	0x00

### 14.2.4. 输出控制寄存器(OUTCFG: 0x4000380C)

描述位	名称	读写	描述	复位值
0	PWM0OEN	R/W	PWM0通道输出使能，捕获模式时设置为0。 0: 禁止 1: 使能	0
1	PWM1OEN	R/W	PWM1通道输出使能，捕获模式时设置为0。 0: 禁止 1: 使能	0
2	PWM2OEN	R/W	PWM2通道输出使能，捕获模式时设置为0。 0: 禁止 1: 使能	0
3	PWM3OEN	R/W	PWM3通道输出使能，捕获模式时设置为0。 0: 禁止 1: 使能	0
4	PWM4OEN	R/W	PWM4通道输出使能，捕获模式时设置为0。 0: 禁止 1: 使能	0
5	PWM5OEN	R/W	PWM5通道输出使能，捕获模式时设置为0。 0: 禁止 1: 使能	0
6	PWM6OEN	R/W	PWM6通道输出使能，捕获模式时设置为0。 0: 禁止 1: 使能	0
7	PROTDIS	R/W	PWM防直通配置。 0: 使能PWM防直通功能，不允许互补输出的 两个通道同时为1	0

描述位	名称	读写	描述	复位值
			1: 禁止PWM防直通功能, 允许互补输出的两个通道同时为1	
8	PWM0POL	R/W	PWM0输出极性控制位。 0: 同相输出 1: 反相输出	0
9	PWM1POL	R/W	PWM1输出极性控制位。 0: 同相输出 1: 反相输出	0
10	PWM2POL	R/W	PWM2输出极性控制位。 0: 同相输出 1: 反相输出	0
11	PWM3POL	R/W	PWM3输出极性控制位。 0: 同相输出 1: 反相输出	0
12	PWM4POL	R/W	PWM4输出极性控制位。 0: 同相输出 1: 反相输出	0
13	PWM5POL	R/W	PWM5输出极性控制位。 0: 同相输出 1: 反相输出	0
14	PWM6POL	R/W	PWM6输出极性控制位。 0: 同相输出 1: 反相输出	0
15	OUTMOD	R/W	输出REMAP选择控制位。 0: UH输出到PWM0, UL输出到PWM1, VH输出到PWM2, VL输出到PWM3, WH输出到PWM4, WL输出到PWM5; 1: UH输出到PWM0, VH输出到PWM1, WH输出到PWM2, UL输出到PWM3, VL输出到PWM4, WL输出到PWM5。	0
16	OUTCFG	R/W	PWM输出配置位。 0: CPU DEBUG断点后输出关闭 1: CPU DEBUG断点后输出不关闭	0
31: 17			保留	

#### 14.2.5. 捕获比较配置寄存器(CICOCFG: 0x40003810)

描述位	名称	读写	描述	复位值
1:0	UHS[1:0]	R/W	当通道0作为捕获输入时,	00

描述位	名称	读写	描述	复位值
			00: 保留 01: 上升沿捕获 10: 下降沿捕获 11: 上升沿捕获或者下降沿捕获 当通道0作为UH比较输出时, 00: UH输出恒为逻辑“0” 01: UH输出恒为逻辑“1” 10: UH输出OC0L 11: UH输出OC0H	
3: 2	VHS[1: 0]	R/W	当通道1作为捕获输入时, 00: 保留 01: 上升沿捕获 10: 下降沿捕获 11: 上升沿捕获或者下降沿捕获 当通道1作为VH比较输出时, 00: VH输出恒为逻辑“0” 01: VH输出恒为逻辑“1” 10: VH输出OC1L 11: VH输出OC1H	00
5: 4	WHS[1: 0]	R/W	当通道2作为捕获输入时, 00: 保留 01: 上升沿捕获 10: 下降沿捕获 11: 上升沿捕获或者下降沿捕获 当通道2作为WH比较输出时, 00: WH输出恒为逻辑“0” 01: WH输出恒为逻辑“1” 10: WH输出OC2L 11: WH输出OC2H	00
7: 6	CH3S[1: 0]	R/W	当通道3作为捕获输入时, 00: 保留 01: 上升沿捕获 10: 下降沿捕获 11: 上升沿捕获或者下降沿捕获 当通道3作为比较输出时, 00: 输出恒为逻辑“0” 01: 输出恒为逻辑“1” 10: 保留 11: 输出OC3H	00
9: 8	ULS[1: 0]	R/W	当通道0作为UL比较输出时,	00

描述位	名称	读写	描述	复位值
			00: UL输出恒为逻辑“0” 01: UL输出恒为逻辑“1” 10: UL输出OC0H 11: UL输出OC0L	
11: 10	VLS[1: 0]	R/W	当通道1作为VL比较输出时, 00: VL输出恒为逻辑“0” 01: VL输出恒为逻辑“1” 10: VL输出OC1H 11: VL输出OC1L	00
13: 12	WLS[1: 0]	R/W	当通道2作为WL比较输出时, 00: WL输出恒为逻辑“0” 01: WL输出恒为逻辑“1” 10: WL输出OC2H 11: WL输出OC2L	00
15: 14	CH3MOD[1: 0]	R/W	当通道3作为比较输出时, 00: 输出默认方波 01: 中央对齐, 上升时输出方波(要保证CC3N小于CC3P) 10: 中央对齐, 下降时输出方波(要保证CC3N小于CC3P) 11: 中央对齐, 上升、下降均输出方波(要保证CC3N小于CC3P)	00
16	UPDATE_TRIG	W	写1触发buffer值更新到控制寄存器, 硬件清零。 MCU写该寄存器其它位时, 只写入buffer, MCU也是从buffer读出。当UPDATE_TRIG=1时才会更新到实际的控制寄存器中。	
31: 17			保留	

● 注意:

写 CICOCFG 寄存器, UPDATE\_TRIG 要同时写 1, 不要分开写, 否则只是写入 buffer, 不会对 PWM 产生作用, 因为还没有写入到实际的控制寄存器。

#### 14.2.6. 中断控制寄存器(INTCON: 0x40003814)

描述位	名称	读写	描述	复位值
0	UDFIEN	R/W	过零溢出中断使能。 0: 禁止 1: 使能	0
1	OVFIEN	R/W	周期溢出中断使能。	0

描述位	名称	读写	描述	复位值
			0: 禁止 1: 使能	
2	FLTIEN	R/W	故障中断使能。 0: 禁止 1: 使能	0
3	UEVIEN	R/W	更新中断使能。 0: 禁止 1: 使能	0
4	UDFSTA	R/W	过零溢出标志，软件写1清零。 0: 未发生过零溢出 1: 发生过零溢出	0
5	OVFSTA	R/W	周期溢出标志，软件写1清零。 0: 未发生周期溢出 1: 发生周期溢出	0
6	FLTSTA	R/W	故障标志，软件写1清零。 0: 未发生故障 1: 发生故障中断	0
7	UEVSTA	R/W	更新事件标志，软件写1清零。 0: 未发生更新 1: 发生更新中断	0
8	CH0RIEN	R/W	0通道上升计数匹配中断使能，或捕获中断使能。（CCR寄存器设置成奇数） 0: 禁止 1: 使能	0
9	CH0FIEN	R/W	0通道下降计数匹配中断使能，捕获模式下无效。（CCR寄存器设置成偶数） 0: 禁止 1: 使能	0
10	CH1RIEN	R/W	1通道上升计数匹配中断使能，或捕获中断使能。（CCR寄存器设置成奇数） 0: 禁止 1: 使能	0
11	CH1FIEN	R/W	1通道下降计数匹配中断使能，捕获模式下无效。（CCR寄存器设置成偶数） 0: 禁止 1: 使能	0
12	CH2RIEN	R/W	2通道上升计数匹配中断使能，或捕获中断使能。（CCR寄存器设置成奇数） 0: 禁止 1: 使能	0

描述位	名称	读写	描述	复位值
13	CH2FIEN	R/W	2通道下降计数匹配中断使能，捕获模式下无效。（CCR寄存器设置成偶数） 0: 禁止 1: 使能	0
14	CH3RIEN	R/W	3通道上升计数匹配中断使能，或捕获中断使能。（CCR寄存器设置成奇数） 0: 禁止 1: 使能	0
15	CH3FIEN	R/W	3通道下降计数匹配中断使能，捕获模式下无效。（CCR寄存器设置成偶数） 0: 禁止 1: 使能	0
16	CH0RSTA	R/W	比较模式通道0计数上升匹配中断标志，写1清零。 0: 未发生匹配中断 1: 发生匹配中断 捕获模式通道0捕获中断标志，写1清零 0: 未发生捕获中断 1: 发生捕获中断	0
17	CH0FSTA	R/W	比较模式通道0计数下降匹配中断标志，写1清零。 0: 未发生匹配中断 1: 发生匹配中断	0
18	CH1RSTA	R/W	比较模式通道1计数上升匹配中断标志，写1清零。 0: 未发生匹配中断 1: 发生匹配中断 捕获模式通道1捕获中断标志，写1清零。 0: 未发生捕获中断 1: 发生捕获中断	0
19	CH1FSTA	R/W	比较模式时通道1计数下降匹配中断标志，写1清零。 0: 未发生匹配中断 1: 发生匹配中断	0
20	CH2RSTA	R/W	比较模式通道2计数上升匹配中断标志，写1清零。 0: 未发生匹配中断 1: 发生匹配中断 捕获模式通道2捕获中断标志，写1清零 0: 未发生捕获中断	0

描述位	名称	读写	描述	复位值
			1: 发生捕获中断	
21	CH2FSTA	R/W	比较模式时通道2计数下降匹配中断标志, 写1清零。 0: 未发生匹配中断 1: 发生匹配中断	0
22	CH3RSTA	R/W	比较模式通道3计数上升匹配中断标志, 写1清零。 0: 未发生匹配中断 1: 发生匹配中断 捕获模式通道3捕获中断标志, 写1清零 0: 未发生捕获中断 1: 发生捕获中断	0
23	CH3FSTA	R/W	比较模式时通道3计数下降匹配中断标志, 写1清零。 0: 未发生匹配中断 1: 发生匹配中断	0
31: 24			保留	

#### 14.2.7. 故障控制寄存器(FTCFG: 0x40003818)

描述位	名称	读写	描述	复位值
6: 0	FCENx	R/W	通道故障控制寄存器, x=0~6。 0: 禁止, 此时对应通道不受故障保护控制 1: 使能, 故障发生后根据故障模式选择软故障或者硬故障保护。	0x00
7	FLTEN		故障输入使能。 0: 禁止 1: 使能	0
8	FLTMOD		故障保护模式选择。 0: 软故障模式(逐周期保护) 1: 硬故障模式(硬件保护)	0
9	FLTSEL		故障信号输入引脚的极性选择。 0: 故障输入信号上升沿时, 触发故障标志 FLTSTA; 1: 故障输入信号下降沿时, 触发故障标志 FLTSTA。	0
10	SCAP		软件触发捕获。SCAP置1, 触发捕获, 动作完成后, SCAP自动清零。	0

描述位	名称	读写	描述	复位值
			0: 禁止; 1: 触发捕获	
13: 11	FLTDB[2: 0]		FLT管脚滤波选择。 000: 不滤波 001: 4个PWM时钟滤波 010: 8个PWM时钟滤波 011: 16个PWM时钟滤波 100: 32个PWM时钟滤波 101: 64个PWM时钟滤波 110: 128个PWM时钟滤波 111: 256个PWM时钟滤波	000
15: 14	STRCH[1: 0]		设置软件触发捕获时, 捕获数据的存储位置。 00: 存储到通道0寄存器 01: 存储到通道1寄存器 10: 存储到通道2寄存器 11: 存储到通道3寄存器	00

#### 14.2.8. 通道 0 数据寄存器 (CCR0N: 0x4000381C)

描述位	名称	读写	描述	复位值
15: 0	CCR0N[15: 0]	R/W	通道0下降匹配数据寄存器。	0xFFFF

#### 14.2.9. 通道 0 数据寄存器 (CCR0P: 0x40003820)

描述位	名称	读写	描述	复位值
15: 0	CCR0P[15: 0]	R/W	通道0上升匹配数据寄存器。	0xFFFF

#### 14.2.10. 通道 1 数据寄存器 (CCR1N: 0x40003824)

描述位	名称	读写	描述	复位值
15: 0	CCR1N[15: 0]	R/W	通道1下降匹配数据寄存器。	0xFFFF

### 14.2.11. 通道 1 数据寄存器 (CCR1P: 0x40003828)

描述位	名称	读写	描述	复位值
15: 0	CCR1P[15: 0]	R/W	通道1上升匹配数据寄存器。	0xFFFF

### 14.2.12. 通道 2 数据寄存器 (CCR2N: 0x4000382C)

描述位	名称	读写	描述	复位值
15: 0	CCR2N[15: 0]	R/W	通道2下降匹配数据寄存器。	0xFFFF

### 14.2.13. 通道 2 数据寄存器 (CCR2P: 0x40003830)

描述位	名称	读写	描述	复位值
15: 0	CCR2P[15: 0]	R/W	通道2上升匹配数据寄存器。	0xFFFF

### 14.2.14. 通道 3 数据寄存器 (CCR3N: 0x40003834)

描述位	名称	读写	描述	复位值
15: 0	CCR3N[15: 0]	R/W	通道3下降匹配数据寄存器。	0xFFFF

### 14.2.15. 通道 3 数据寄存器 (CCR3P: 0x40003838)

描述位	名称	读写	描述	复位值
15: 0	CCR3P[15: 0]	R/W	通道3上升匹配数据寄存器。	0xFFFF

### 14.2.16. PWMTRGC 控制寄存器 (TRGC: 0x4000383C)

描述位	名称	读写	描述	复位值
3: 0	BUFC[3: 0]	R/W	某些情况下CICOCFG需要预存在BUF中，在某 个事件触发后才会载入PWM功能模块。 0000: 实时(与MCU写同步) 0001: 0通道上升沿 0010: 0通道下降沿 0011: 0通道上升或下降沿 0100: 1通道上升沿 0101: 1通道下降沿	0000

描述位	名称	读写	描述	复位值
			0110: 1通道上升或下降沿 0111: 2通道上升沿 1000: 2通道下降沿 1001: 2通道上升或下降沿 1010: 计数器过零 1011: 计数器溢出 1100: 过零或溢出 其它: 实时(与MCU写同步)	
6: 4	RSVD[2: 0]	R/W	保留	
7	SAMP_MODE	R/W	PWM单电阻采样模式。 0: 非单电阻采样 1: 单电阻采样	0
15: 8	TADC[7: 0]	R/W	电流稳定时间+ADC采样时间，用于单电阻 模式调整PWM波形。	0x00
16	SAMP_OPT	R/W	PWM 单电阻采样模式选项 0: U、V、W 调整关闭，但是 PWMSSTA [7:4] 可以指示出 PWM 占空比是否满足 ADC 采样。 1: U、V、W调整开启	0

#### 14.2.17. PWM 计数寄存器(PWMCNT: 0x40003840)

描述位	名称	读写	描述	复位值
15: 0	PWMCNT[15: 0]	R/W	PWM计数寄存器。	0x0000

#### 14.2.18. PWM 输出极性配置锁定寄存器(PWMLOCK: 0x40003844)

描述位	名称	读写	描述	复位值
0	PWMLOCK	R/W	只有当PWMLOCK写入0xB3时，LOCK位置位为 高。LOCK位为高时，用户无法修改。 OUTCFG.PWMxPOL寄存器，只有发生外部复 位，上电复位或者掉电复位时，LOCK位方可清 零。	0x00
7: 1			保留	

### 14.2.19. PWM 单电阻采样状态寄存器(PWMSSTA: 0x40003848)

描述位	名称	读写	描述	复位值
2:0	ORD[2:0]	R	U、V、W三相PWM占空比排序。 000: U>V>W, 采样电流依次是-lw、lu 001: U>W>V, 采样电流依次是-lv、lu 011: W>U>V, 采样电流依次是-lv、lw 100: V>U>W, 采样电流依次是-lw、lv 110: V>W>U, 采样电流依次是-lu、lv 111: W>V>U, 采样电流依次是-lu、lw 其它无效。	000
3			保留	
7~4	STA[3:0]	R	PWM波形调整标志。 0000: U、V、W没有调整, 波形正常 0001: U、V、W中间值右移 0010: U、V、W中间值左移 0100: U、V、W中间值左移, 最大值右移 1000: U、V、W中间值右移, 最大值右移 1111: U、V、W没有调整, 三相占空比都不大于Tadc*2。	0000

### 14.3.功能描述

集成一个 16 位的 PWM 波形发生器模块, 具有 4 个通道, 每个通道可独立配置为捕获输入模式或者 PWM 输出模式。EPWM 模块内部集成了死区控制单元, 可输出 3 组互补的 PWM 波形和一路单独的 PWM 输出。工作在 PWM 模式时, 支持四种工作模式: 连续边沿对齐、连续中央对齐、单次边沿对齐、单次中央对齐模式。工作在捕获模式时, 最多支持 4 个捕获通道。此外该模块还支持 1 个故障处理单元, 支持逐周期保护(软故障)或者禁止 PWM 输出(硬故障)。故障保护时, PWM 输出通道关闭, 对应通道的输出状态由 GPIO 的配置决定(通常建议相应的 GPIO 设置成输入口, 即故障保护时, 通道输出状态为高阻态)。

### 14.3.1. 计数器工作方式

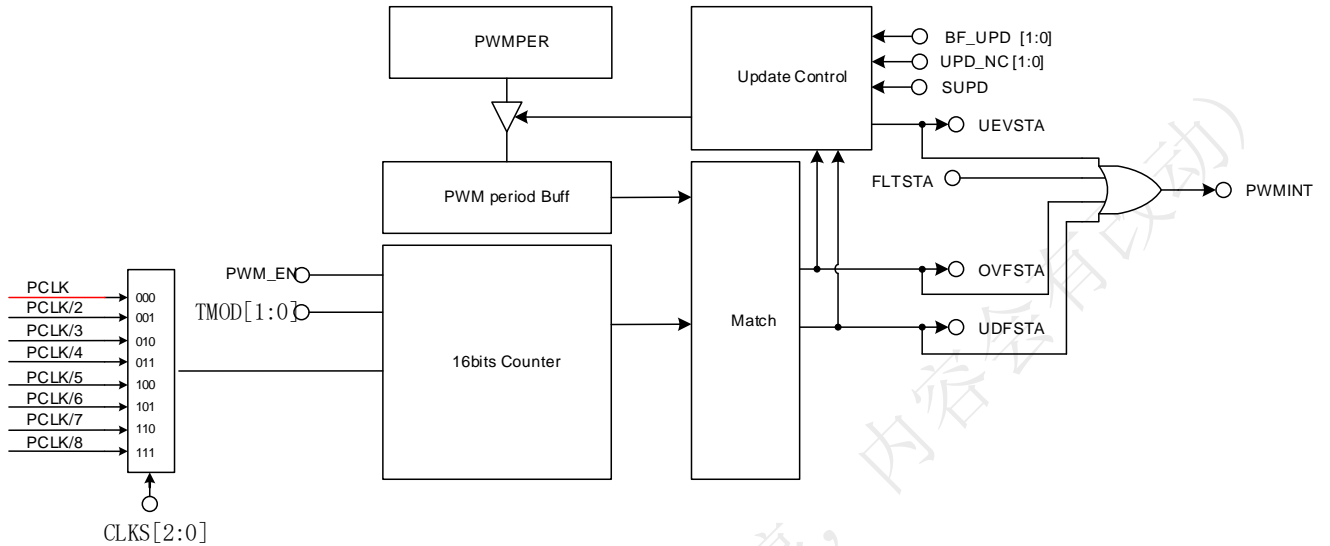
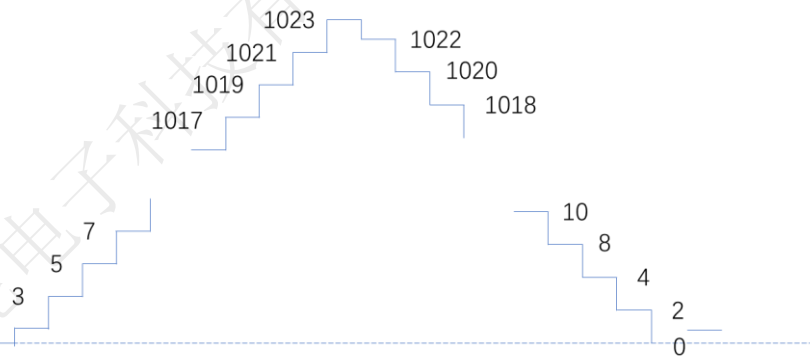


图 14-3-1-1 PWM 计数模式

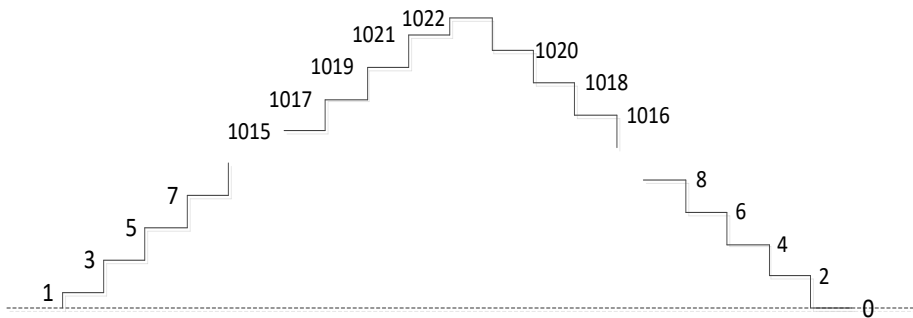
当计数器将工作在边沿对齐模式时，Counter 还是每次加 1、减 1。

当计数器将工作在中央对齐模式时，Counter 每次增加、减小两个数字，上升时奇数，下降时偶数。这样，当 duty 的数值发生变化 1 的时候，输出的占空比左边增加一个，右边增加一个，而不是同时增加两个。这样增加了分辨率，解决了采用中心对齐方式时输出 PWM 分辨率损失了一半的问题。

- 周期为奇数的中央对齐模式



- 周期为偶数的中央对齐模式



注意：当需要设置 100% PWM 占空比输出时，CCRxP 寄存器的值需要比 PWMPER 寄存器的值大（不能等于）。

● 连续边沿对齐模式

当 PWM 控制寄存器 PWMCON 中的 TMOD [1: 0] 为 00 时，计数器将工作在连续边沿对齐模式。内部 16 位计数器将循环从 0 计数到周期缓冲寄存器，然后从 0 开始重新计数。PWM 周期寄存器 PWMPER 在预设的时刻更新到周期缓冲寄存器。

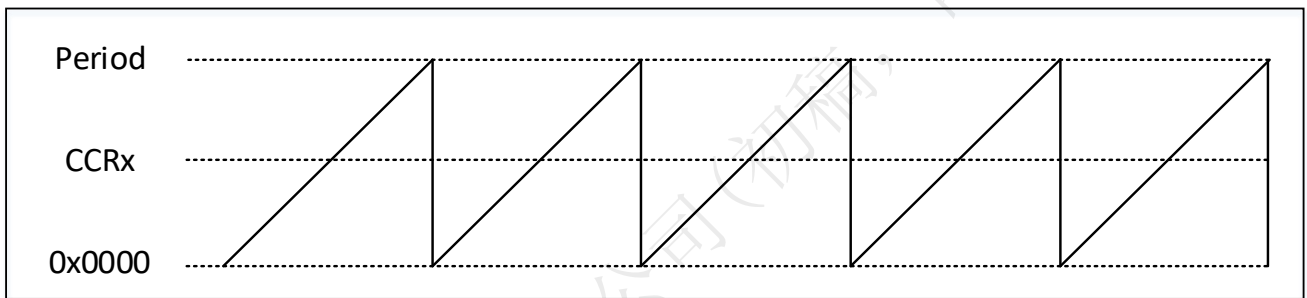


图 14-3-1-2 连续边沿对齐模式

● 连续中央对齐模式

当 PWM 控制寄存器 PWMCON 中的 TMOD [1: 0] 为 01 时，PWM 将工作在连续中央对齐模式。内部 16 位计数器将循环从 0 上升计数到周期缓冲寄存器，然后再下降计数到 0 后再重新开始向上计数。PWM 周期寄存器 PWMPER 在预设的时刻会更新到周期缓冲寄存器。工作在该模式下，可通过读取 PWMCON 中的 STRG 位判断当前计数器是处于上升计数状态还是下降计数状态。

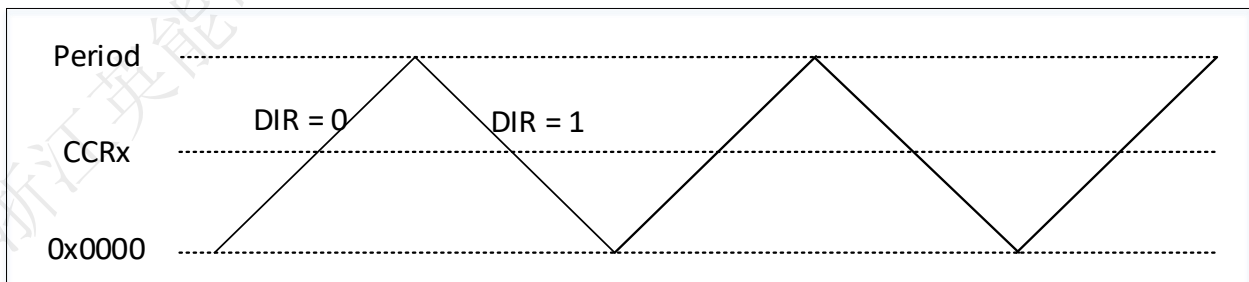


图 14-3-1-3 连续中央对齐模式

● 单次边沿对齐模式

当 PWM 控制寄存器 (PWMCON) 中的 TMOD [1: 0] 为 10 时, PWM 将工作在单次边沿对齐模式。启动后, 内部 16 位计数器将从 0 计数到周期缓冲寄存器, 然后计数器的值恢复为 0, 并停止计数。PWM 单次边沿对齐模式的启动由 PWMCON 寄存器中的 STRG 位来触发, 该位在写 1 后自动清零。

- **单次中央对齐模式**

当 PWM 控制寄存器 (PWMCON) 中的 TMOD [1: 0] 位为 11 时, PWM 将工作在单次中央对齐模式。启动后, 内部 16 位计数器将从 0 计数到周期缓冲寄存器, 然后向下计数到 0 后, 停止计数。PWM 单次中央对齐模式的启动由 PWMCON 寄存器中的 STRG 位来触发, 该位在写 1 后自动清零。通过读取 PWMCON 中的 STRG 位判断当前计数器是处于上升计数状态还是处于下降计数状态。

- **缓冲寄存器更新**

PWM 周期寄存器 PWMPER、数据寄存器 (CCR<sub>xP</sub>/CCR<sub>xN</sub>) 和死区寄存器 DTCON 都有缓冲寄存器, 用户不能直接操作这些缓冲寄存器。在写 PWM 周期寄存器、匹配寄存器和死区控制寄存器时, 写入的值不会立即更新到缓冲寄存器。用户可以通过 PWM 配置寄存器 PWMCON 中的 UPD\_NC 和 BF\_UPD 位来配置更新时刻。

连续中央对齐模式下, 更新事件可以选择软件更新、过零溢出更新、周期匹配更新、过零溢出或周期匹配更新; 连续边沿对齐模式下, 更新时间可以选择软件更新或者周期匹配更新。

采用过零溢出/周期匹配更新时, UPD\_NC 位用来设置更新频率, 每当出现 1 次/2 次/4 次/8 次更新事件时, 产生一次更新动作。

采用软件更新时, 更新频率不受 UPD\_NC 位控制, 只要使能 PWMCON 中的 SUPD 位就会立即更新缓冲寄存器。对于单次边沿或者单次中央对齐模式, 缓冲寄存器更新只能选择软件更新。

只有通道作为比较输出时, 更新操作才会更新相应的通道缓冲寄存器。无论工作在何种状态, PWM 周期缓冲寄存器和死区缓冲寄存器只要发生更新事件都会更新。

举例 1: 下图是 BF\_UPD[1: 0]=10 选择周期匹配作为更新源, UPD\_NC[1: 0]=01 选择两次更新事件发生时更新缓冲寄存器。

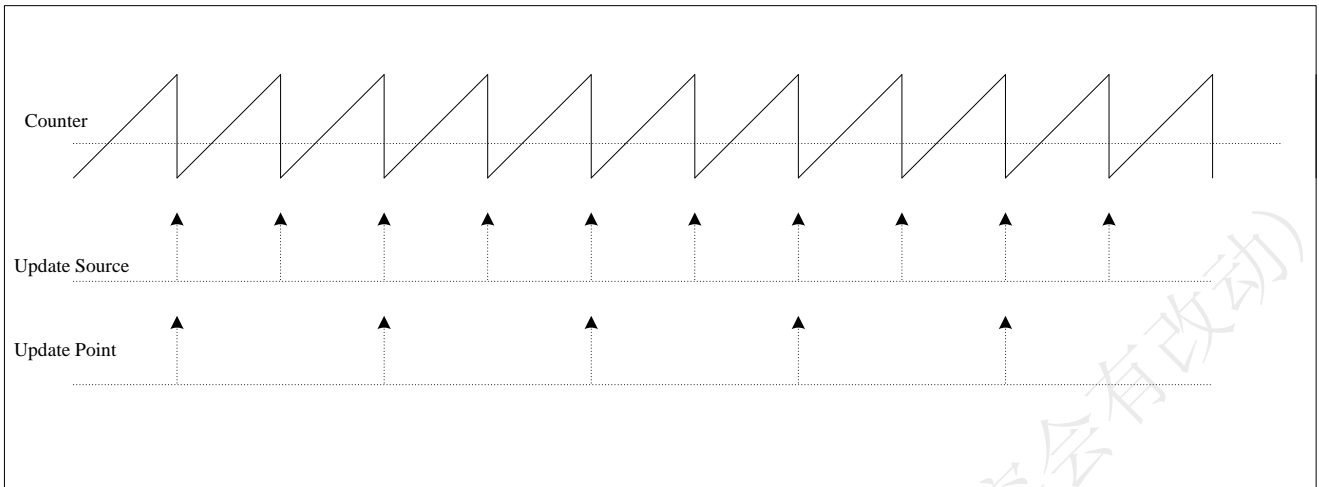


图 14-3-1-4 边沿对齐模式的周期匹配更新

举例 2: 下图是  $BF\_UPD[1: 0]=11$  选择周期匹配和过零匹配作为更新源,  $UPD\_NC[1: 0]=01$  选择两次更新事件发生时更新缓冲寄存器。

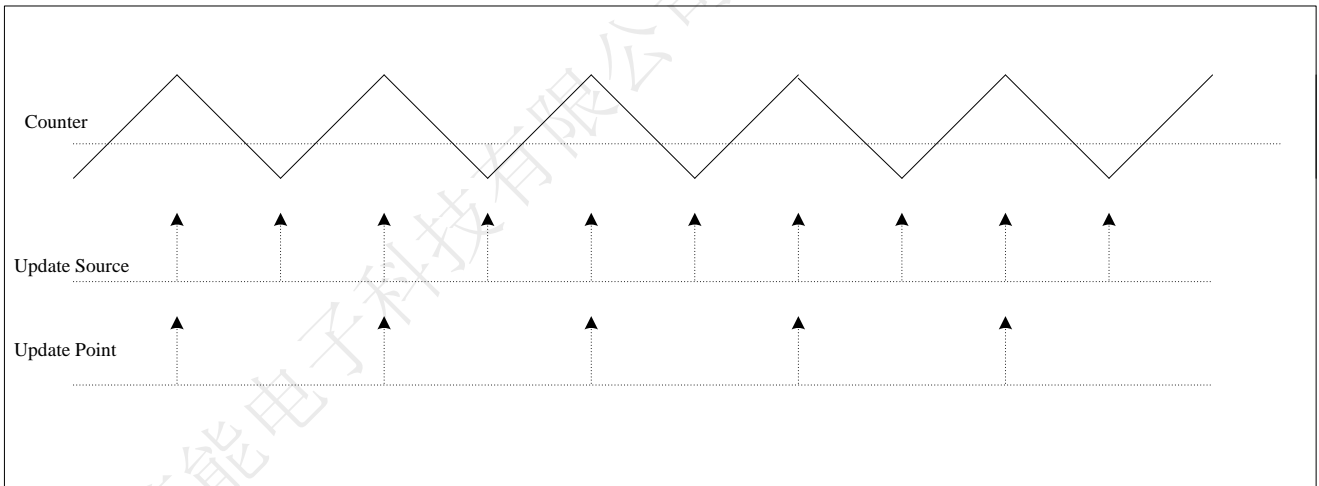


图 14-3-1-5 中央对齐模式的周期匹配和过零匹配更新

### 14.3.2. 捕获模式



时全局中断使能，CPU 将响应捕获中断。

- **捕获模式下计数器工作模式**

捕获模式下，计数器的工作模式可由控制寄存器的 **TMOD** 位配置为边沿对齐还是中央对齐模式。配置为中央对齐模式时，可通过读取控制寄存器的 (**STRG**) 位判断计数是处于上升计数状态还是处于下降计数状态，当读取的 **STRG** 位是 **0** 时，表明处于上升计数状态，当读取的 **STRG** 位是 **1** 时，表明处于下降计数状态。当配置为边沿对齐模式时，读取 **STRG** 位一直为 **0**。

- **软件触发捕获功能**

捕获模式下，可以通过软件触发捕获事件，来获取计数器的当前计数值。寄存器 **FTCFG** 的 **STRCH** 用于配置软件捕获的数据存储通道；寄存器 **FTCFG** 的 **SCAP** 置 **1**，单次触发捕获事件，计数器的当前计数值存入 **STRCH** 配置的相应通道 **CCRxP**，捕获完成后，**SCAP** 标志自动清零。

- **捕获数据读取的注意事项**

若当前计数模式配置为中央对齐模式，需要通过读取控制寄存器中的 **STRG** 位判断当前计数是处于上升计数状态还是下降计数状态。再结合捕获/比较寄存器 **CCRxP** 来判断外部事件的触发间隔时间。



图等相应章节。

PWM 的分频可由配置寄存器 PWMCON 中的 CLKS 位选择。

PWM\_EN 置 1，EPWM 模块启动工作，计数器开始计数。

### ● 边沿对齐模式

当 PWM 控制寄存器 PWMCON 的 TMOD[1: 0] = 00 时，计数器工作在边沿对齐模式。计数器从 0 开始计数到周期缓冲寄存器后，重复开始从 0 开始计数。

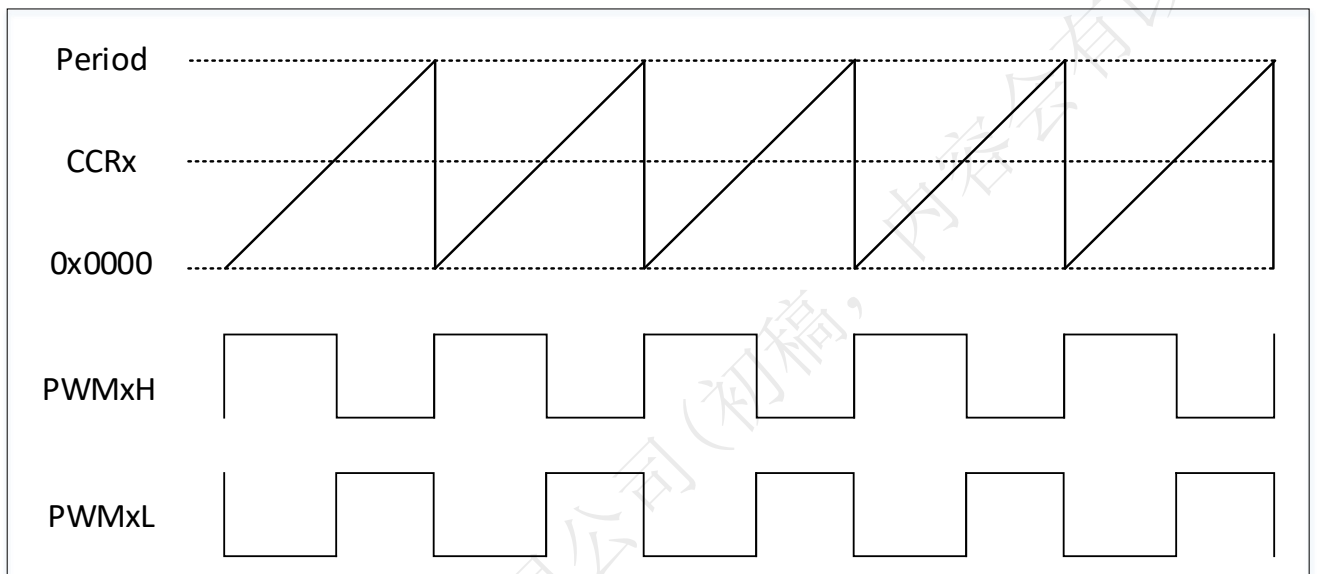


图 14-3-3-2 边沿对齐模式

### 边沿模式匹配中断

工作在边沿对齐模式下，当计数器的值和捕获/比较缓冲寄存器的值相等时，若 INTCON 中的上升沿匹配中断使能位 (CHxRIEN) 使能，PWMCCISR 中的匹配中断标志 (CHxCPRSTA) 将会被置 1。如果此时全局中断使能，CPU 将响应匹配中断。匹配发生后不会清零计数器，而是继续累加，直到计数器与周期缓冲寄存器的值相等时，发生溢出从 0x0000 开始重新计数。

边沿对齐模式下，INTCON 中的下降沿匹配中断无效。

### 边沿模式输出

当计数器的值小于等于捕获/比较缓冲寄存器的值时，相应通道的 PWMxH 输出为高电平，当计数器的值大于捕获/比较缓冲寄存器的值时，相应通道的 PWMxH 输出为低电平；周期缓冲寄存器的值或者捕获/比较缓冲寄存器的值为 0 时，相应通道的 PWMxH 输出为恒低电平；捕获/比较缓冲寄存器的值大于等于周期缓冲寄存器的值时，相应通道 PWMxH 输出为恒高电平。PWMxL 的输出与 PWMxH 的输出互补。

● 中央对齐模式

当 PWM 控制寄存器(PWMCON)的 TMOD[1: 0] = 01 时, 计数器工作在中央对齐模式。计数器从 0 开始计数到周期缓冲寄存器后, 开始下降计数直到计数到 0, 然后再重新开始从 0 计数。

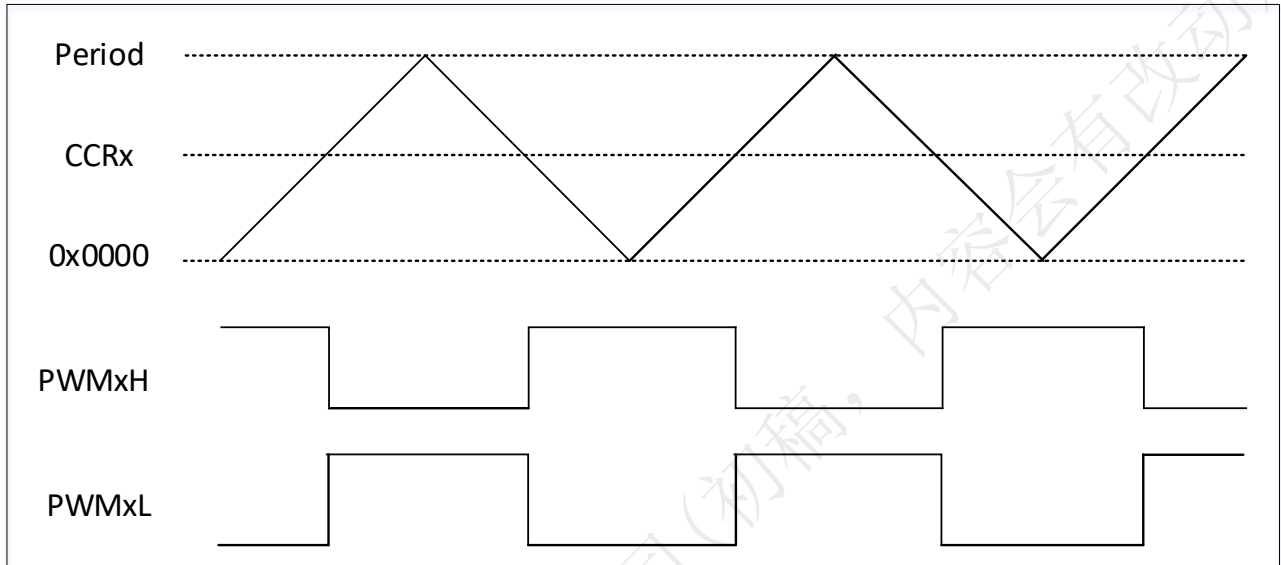


图 14-3-3-3 中央对齐模式

中央对齐模式匹配中断

工作在中央对齐模式下, 在上升计数时, 当计数器的值和捕获/比较缓冲寄存器的值相等, 若 INTCON 中的上升沿匹配中断使能位(CHxRIEN)使能, INTCON 中的匹配中断标志 CHxRSTA 将会被置 1。在下降计数时, 当计数器的值和捕获/比较缓冲寄存器的值相等, 若 INTCON 中的下降沿匹配中断使能位(CHxFIEN)使能, INTCON 中的匹配中断标志(CHxFSTA)将会被置 1。如果此时全局中断使能, CPU 将响应匹配中断。匹配发生后不会清零计数器, 而是继续计数。

中央对齐模式输出

当计数器的值小于等于捕获/比较缓冲寄存器的值时, 相应通道的 PWMxH 输出为高电平, 当计数器的值大于捕获/比较缓冲寄存器的值时, 相应通道的 PWMxH 输出为低电平; 周期缓冲寄存器的值或者捕获/比较缓冲寄存器的值都为 0 时, 相应通道的 PWMxH 输出为恒低电平; 捕获/比较缓冲寄存器的值大于等于周期缓冲寄存器的值时, 相应通道的 PWMxH 输出为恒高电平。PWMxL 的输出与 PWMxH 的输出互补。

### 14.3.4. PWM 输出控制

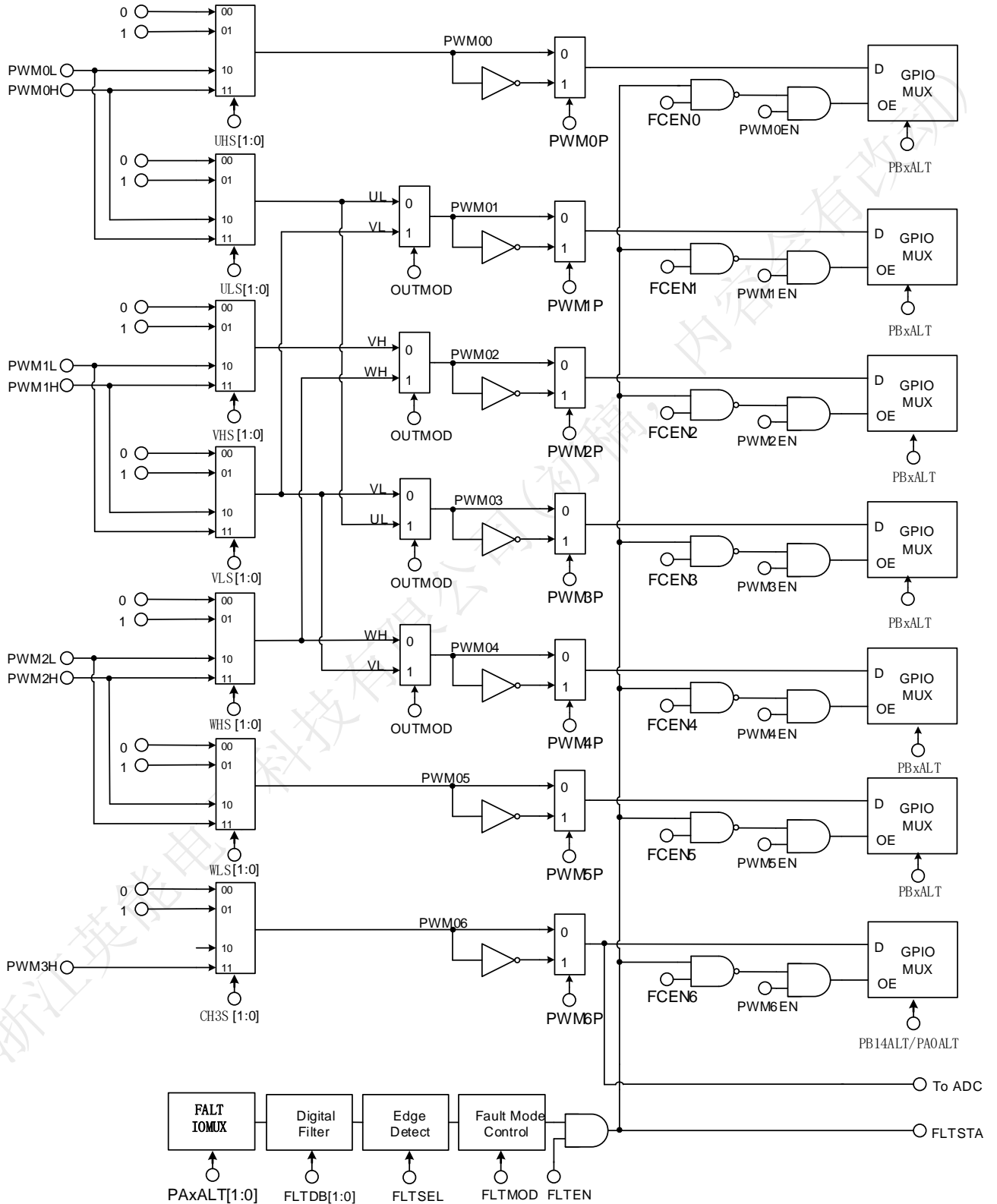


图 14-3-4-1 PWM 输出控制

● 输出模式配置

EPWM 模块的输出可以由寄存器 CICOCFG、OUTCFG.PWMxPOL 以及寄存器 OUTCFG 的 PROTDIS 位配置。

● 输出信号源

UH/UL/VH/VL/WH/WL 的输出可以有 4 种信号源: 恒低电平、恒高电平、PWMxH 或 PWMxL。寄存器 CICOCFG 用来设置 UH/UL/VH/VL/WH/WL 的输出信号源。

恒低电平对应 0% 占空比, 恒高电平对应 100% 占空比, PWMxH 的占空比与匹配寄存器设置的占空比 Duty 一致, PWMxL 的占空比与“1 - Duty”一致。下面将以上图所示的 4 种情况为例, 介绍如何设置输出信号源。

		UH	UL	VH	VL	WH	WL
例 1	选择的信号源	PWM0H	恒“0”	恒“0”	恒“1”	恒“0”	恒“0”
	寄存器设置	UHS = 3	ULS = 0	VHS = 0	VLS = 1	WHS = 0	WLS = 0
例 2	选择的信号源	恒“0”	PWM0H	恒“1”	恒“0”	恒“0”	恒“0”
	寄存器设置	UHS = 0	ULS = 2	VHS = 1	VLS = 0	WHS = 0	WLS = 0
例 3	选择的信号源	PWM0H	PWM0L	PWM1H	PWM1L	PWM2H	PWM2L
	寄存器设置	UHS = 3	ULS = 3	VHS = 3	VLS = 3	WHS = 3	WLS = 3
例 4	选择的信号源	PWM0L	PWM0H	PWM1L	PWM1H	PWM2L	PWM2H
	寄存器设置	UHS = 2	ULS = 2	VHS = 2	VLS = 2	WHS = 2	WLS = 2

表 14-3-4-1 PWM 信号源设置

● 输出极性设置

通过寄存器 OUTCFG 可以设置 PWM 的输出极性。输出极性的设置与预驱动和功率管的类型有关。

序号	OUTCFG 设置	PWM 输出极性	预驱动类型	功率管类型
1	PWMxP = 0	同相输出	同相输出	NMOS
2	PWMxP = 0	同相输出	反相输出	PMOS
3	PWMxP = 1	反相输出	反相输出	NMOS
4	PWMxP = 1	反相输出	同相输出	PMOS

表 14-3-4-2 PWM 输出极性设置

● 输出极性配置锁定

为了防止误动作导致输出极性翻转, 进而导致功率管直通故障, 用户可以通过设置寄存器 PWMLOCK, 锁定寄存器 OUTCFG.PWMxPOL 的设置。寄存器 PWMLOCK 的用法如下: 当 PWMLOCK 写入 0xB3 时, LOCK 位被置 1。LOCK = 1 时, 用户无法修改 OUTCFG.PWMxPOL 寄存器, 只有发生外部复位, 上电复位或者掉电复位时, LOCK 位才可以清零。

● 输出信号重新排列(REMAP)

EPWM 模块内部信号(UH/UL/VH/VL/WH/WL)与外部引脚(PWM00~05)的对应关系, 见下表所示。(OUTMOD 为寄存器 OUTCFG 中的 BIT15 位。)

寄存器设置	PWM00	PWM01	PWM02	PWM03	PWM04	PWM05
OUTMOD = 0	UH	UL	VH	VL	WH	WL
OUTMOD = 1	UH	VH	WH	UL	VL	WL

● 防直通功能

寄存器 OUTCFG 的 PROTDIS = 0 时，使能 PWM 防直通功能，此时组成互补 PWM 的 2 路输出不能在同一时刻出现逻辑 1。如果出现同一个通道的互补输出同时输出逻辑 1 时，将通过硬件确保管脚输出逻辑 0。

注：此处的逻辑 1 和逻辑 0 与电平高/低没有直接对应关系。对于极性设置的前端信号，逻辑 1 与高电平对应，逻辑 0 与低电平对应。对于极性设置的后端信号，输出极性设置为正相输出时，逻辑 1 与高电平对应，逻辑 0 与低电平对应；输出极性设置为反相输出时，逻辑 1 与低电平对应，逻辑 0 与高电平对应。

● ADC 的触发源

PWM06 可作为 ADC 的触发源。

● 死区控制

无论 PWM 工作在中央对齐还是边沿对齐模式，EPWM 模块都支持死区控制。信号 PWMx 作为参考输入，信号 PWMxH/PWMxL 作为死区控制输出，死区时间由寄存器 DTCON 设置。

当 DTR 设置不为 0 时，信号 PWMxH 的上升沿比 PWMx 的上升沿延时输出，信号 PWMxH 的下降沿和 PWMx 的下降沿同时输出。PWMx 上升沿延时输出的时间长度为  $(PWMDR+1)*2/Freq\_SYSCLK$ 。当 PWMDR 设置为 0 时，PWMxH 死区控制无效，PWMxH 直接输出 PWMx 信号。

当 DTF 设置不为 0 时，信号 PWMxL 的上升沿比 PWMx 的下降沿延时输出，信号 PWMxL 的下降沿和 PWMx 的上升沿同时输出。PWMx 下降沿延时输出的时间长度为  $(PWMDF+1)*2/Freq\_SYSCLK$ 。当 PWMDF 设置为 0 时，PWMxL 死区控制无效，PWMxL 直接输出 PWMx 的互补信号。

如果 PWMx 的脉冲宽度小于 DTR 或者 DTF，相应的脉冲信号将不生成。

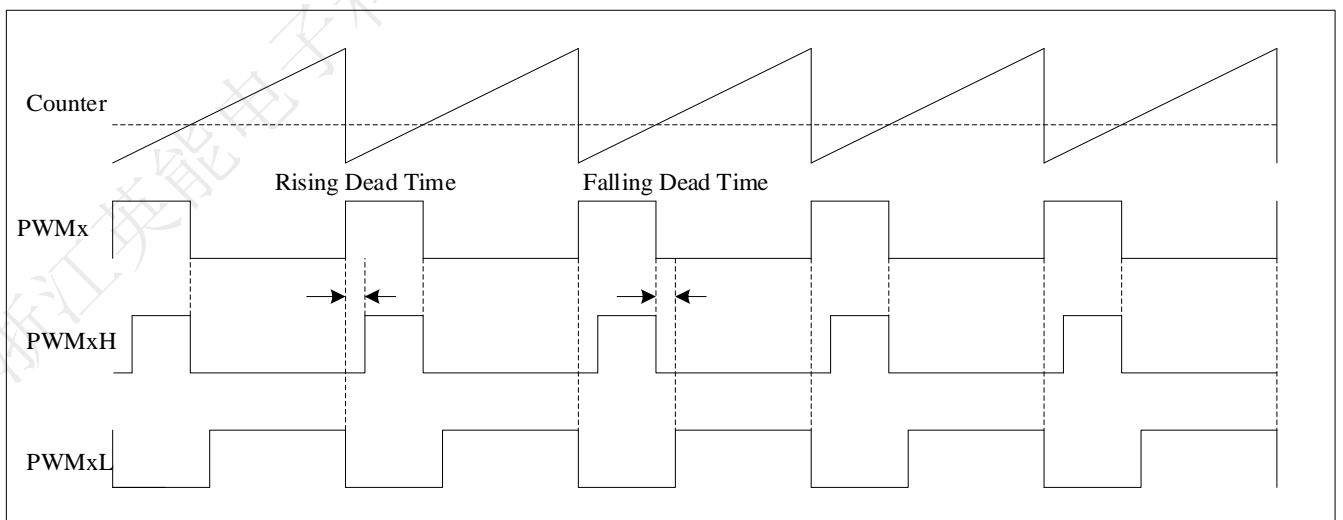


图 14-3-4-2 当 PWMDR 和 PWMDF 不为 0 时，死区控制逻辑

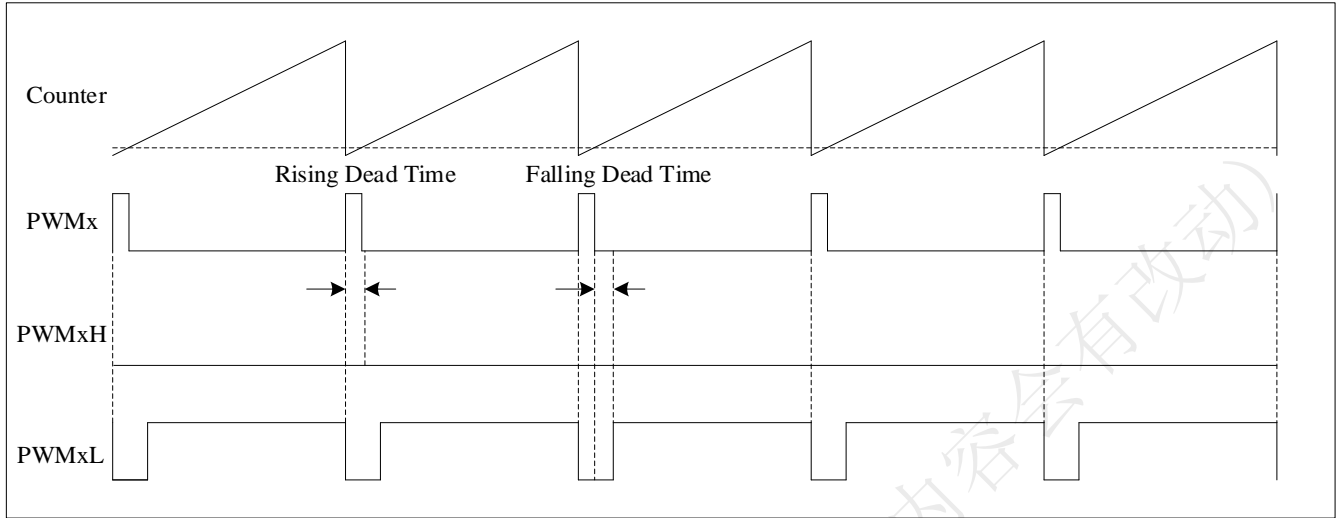


图 14-3-4-3 当 PWMxH 脉冲宽度小于 PwMDR 设定的死区时间

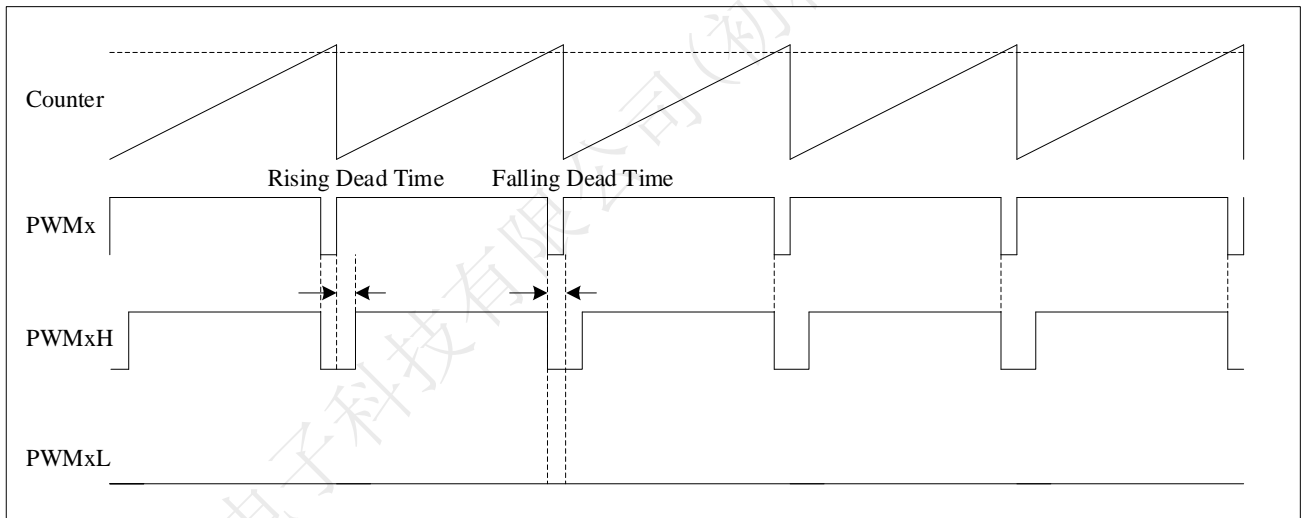


图 14-3-4-4 当 PWMxL 宽度小于 PwMDF 设定的死区时间

### ● 故障保护

FLT 管脚输入故障保护可设置为软故障保护或硬故障保护。故障控制寄存器 FTCFG 的 FLTSEL 位可以选择故障信号的极性，FLTSEL = 0 时，故障输入信号的上升沿，触发故障标志 FLTSTA；FLTSEL = 1 时，故障输入信号的下降沿，触发故障标志 FLTSTA。

软故障保护，又叫逐周期故障保护，故障发生时，寄存器 OUTCFG 的 PWMxEN 标志清零，PWMx 输出关闭。故障消失后的下一个 PWM 周期，寄存器 OUTCFG 的 PWMxEN 标志自动恢复，PWMx 恢复正常输出。边沿对齐模式下，PWM 计数器与周期寄存器匹配时恢复输出；中心

对齐模式下，PWM 计数器与周期寄存器匹配或计数器归 0 时恢复正常输出。故障标志 FLTSTA 由故障信号 FLT 的有效沿触发，需要手动清零。

对于各个通道是否启用故障保护功能，可由 FTCFG 寄存器来配置。每个通道都可以独立配置为故障保护使能或者禁止。

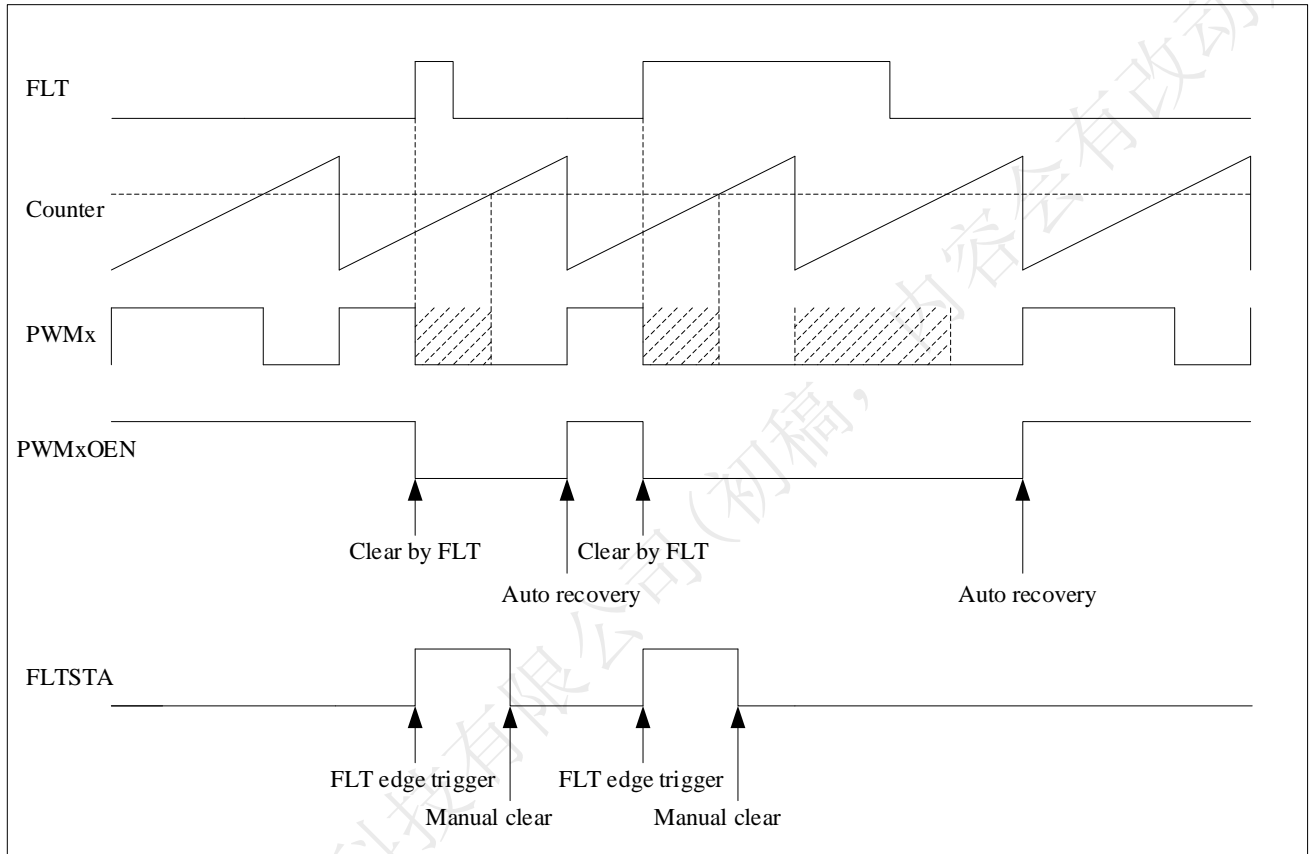


图 14-3-4-5 软故障保护(逐周期保护)

硬故障保护发生时，寄存器 OUTCFG 的 PWMxEN 标志清零，PWMx 输出关闭。故障消失后，寄存器 OUTCFG 的 PWMxEN 标志不会自动恢复，需要重新写 OUTCFG 寄存器才能恢复正常输出。故障标志 FLTSTA 由故障信号 FLT 的有效沿触发，需要手动清零。

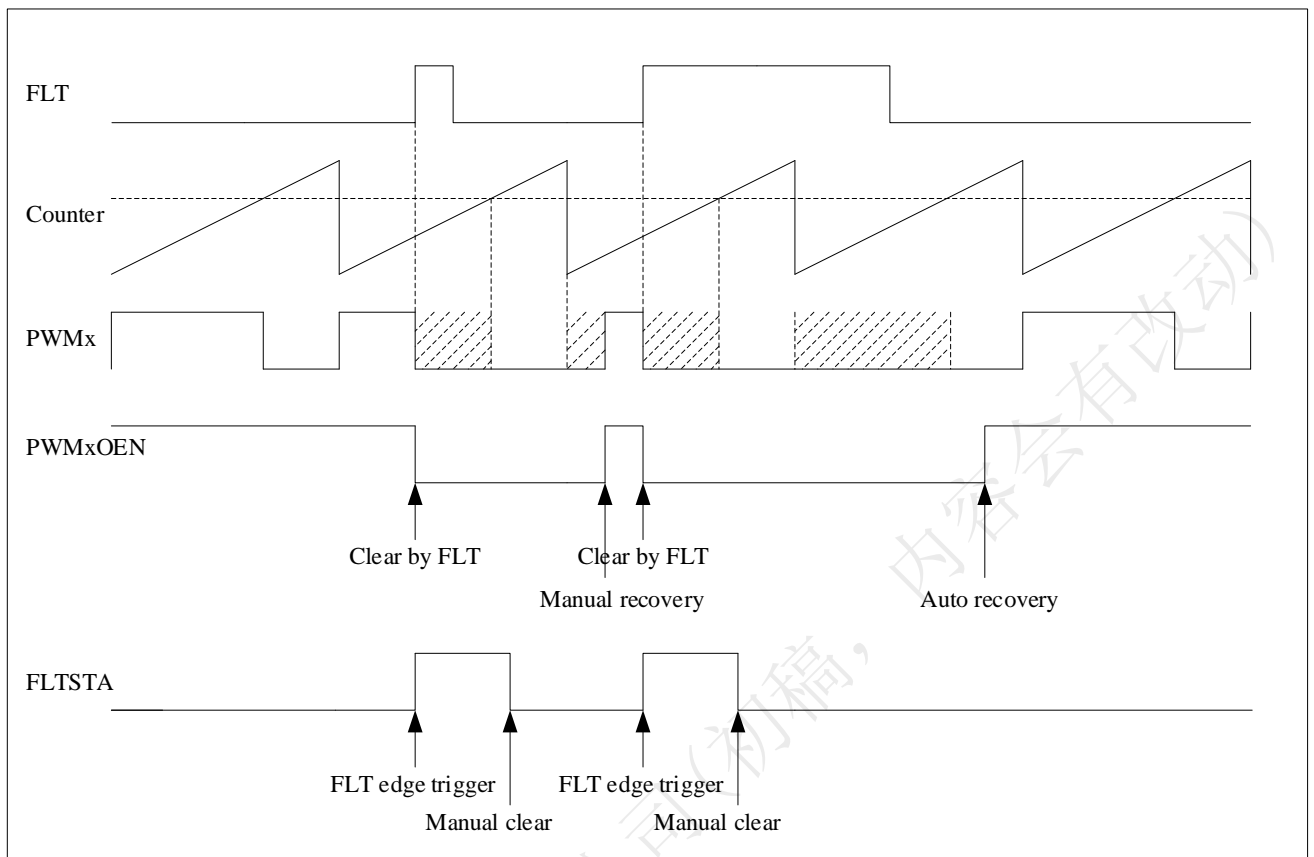


图 14-3-4-6 硬故障保护

注意事项: 发生故障时, PWM 输出端口关闭, 切换为由 IO 口设置的状态。建议在程序初始化时, 将复用 PWM 输出功能的 IO 口设置为输入模式, 并在外部引脚设置上拉电阻 (低电平有效时) 或下拉电阻 (高电平有效时), 那么发生故障时, PWM 输出端口会自动切换为 IO 口模式, 即输入高阻态, 端口的电平将由上拉电阻或下拉电阻决定。

### 14.3.5. 中断

PWM 中断源共有 5 个: 周期匹配中断、过零中断、故障保护中断、缓冲寄存器更新中断、比较匹配中断/捕获中断。

- **周期匹配中断**

无论计数器是工作在边沿对齐还是中央对齐模式, 在计数器与周期缓冲寄存器相等时产生溢出事件, 若此时 INTCON 中的溢出中断使能位使能, 将会产生 PWM 周期匹配中断。

- **过零中断**

中央对齐模式下, 当计数器下降计数到 0 时将会产生过零事件, 若此时 INTCON 中的过零中断使能位使能, 将会产生 PWM 过零中断。

- **故障保护中断**

如果 INTCON 中的故障中断使能位使能, 那么发生故障时, 将会产生 PWM 故障中断。

● **缓冲寄存器更新中断**

如果 INTCON 中的缓冲寄存器更新中断使能位使能，那么发生缓冲寄存器更新时，将会产生 PWM 缓冲寄存器更新中断。

● **比较匹配中断和捕获中断**

PWM 匹配或捕获中断详见捕获模式和比较模式中的相关描述。

● **故障标志清零**

PWM 的所有中断标志，都不会硬件自动清除，需要手动往 INTCON 的相应位写 1 才能清除相关中断。

## 15. 频率检测器(TF)

PWM 频率捕获模块(TF)用于捕获输入 PWM 信号的频率，计数时钟可调。

### 15.1. 设计框图

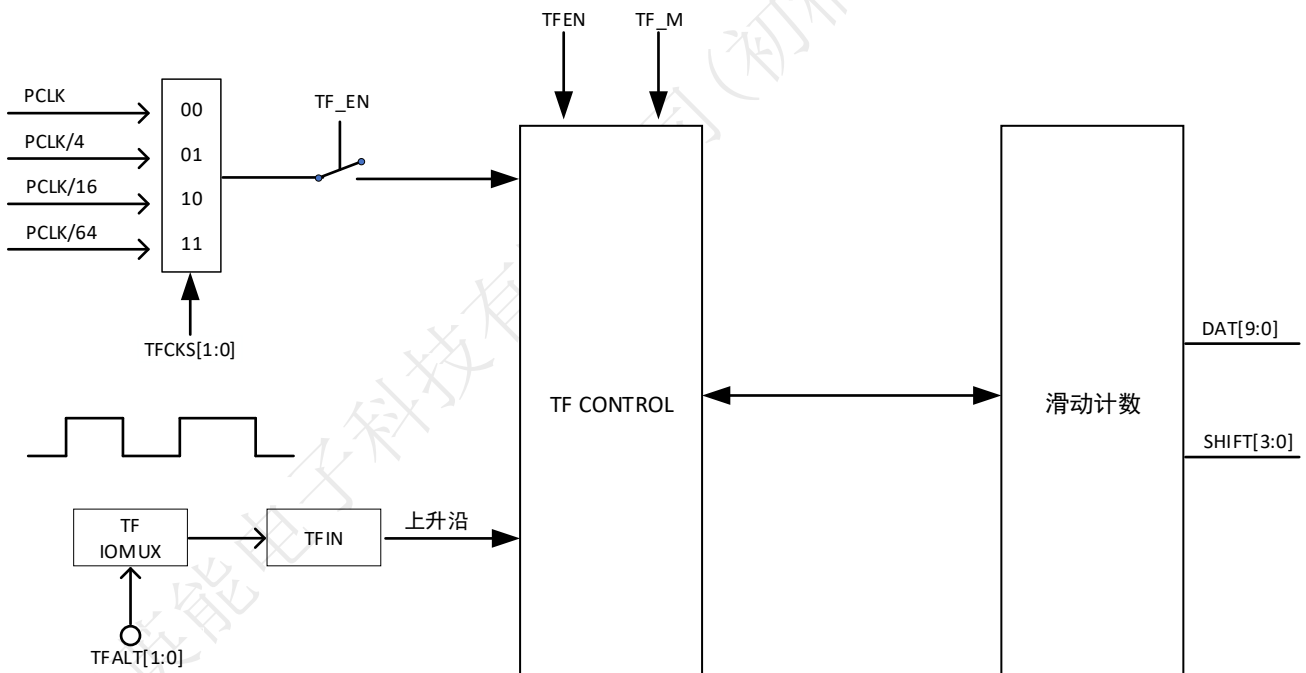


图 15-1 TF 频率捕获模块的原理框图

### 15.2. 模块寄存器总表

寄存器名	地址	结构体(.h)	描述	复位值
TFCON	0x40009800	TF	频率捕获控制寄存器	0x00
TFDAT	0x40009804	TF	频率捕获数据寄存器	0x0000

## 15.3. 寄存器说明

### 15.3.1. 频率捕获控制寄存器(TFCON: 0x40009800)

描述位	名称	读写	描述	复位值
0	TFEN	R/W	TFEN: 频率捕获模块使能。 0: 关闭 1: 使能	0
1	TF_M	R/W	TF_M: 频率捕获控制。 0: 8bit模式, 底数范围0~255; 1: 10bit模式, 底数范围0~1023;	0
7: 2			保留	

### 15.3.2. 频率捕获数据(TFDAT: 0x40009804)

描述位	名称	读写	描述	复位值
9: 0	DAT[9: 0]	R	DAT[9: 0]: 频率捕获底数。 8位模式时, DAT[7: 0]构成捕获底数; 10位模式时, DAT[9: 0]构成捕获底数。	0x000
13: 10	SHIFT[3: 0]:	R	频率捕获指数	0x0
15: 14			保留	

## 15.4. 功能描述

使用滑动计数器进行计数, 所谓滑动计数器就是计数最大值固定(本应用为 255), 当时间超过这个值后, 计数器会自动移位, 同时记录下移位大小。

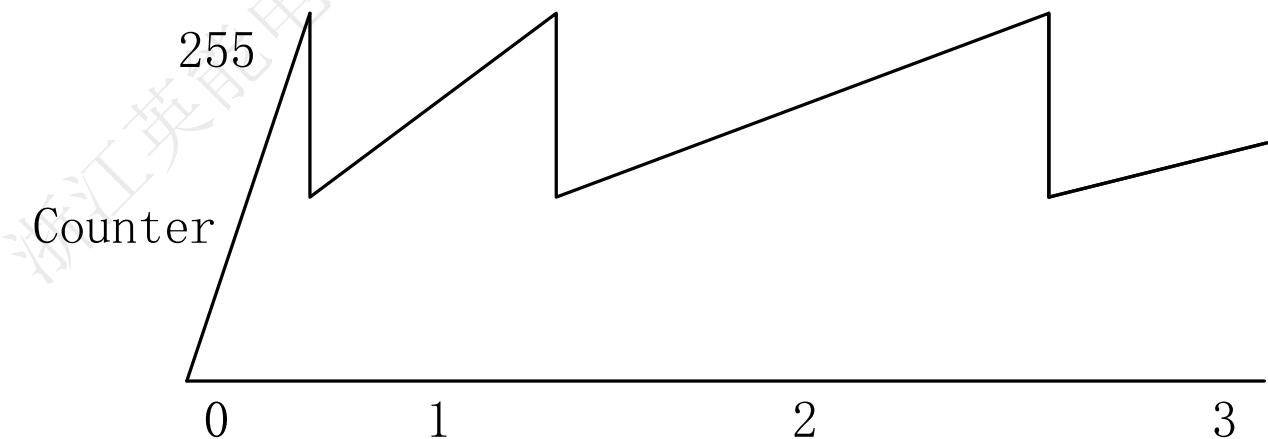


图 15-4 频率捕获计数过程

比如上面这个例子，采用 8MHz 时钟，计数器溢出了三次，三次减半(移位)，移位量为 3。最终的结果是  $150 \times 2^3$ ，150 是最终 counter 的结果，3 是移位次数。

最后要做一次除法，8000000 除以周期，就是电机的转动频率，带入闭环的初始化速度。

这个模块支持 10 位或 8 位底数，4 位 2 幂计数，客户可以通过 TDC\_M 控制位选择读取 8 位或者 10 位。比如：

PWM 频率是 1M，结果是：10'd8、4'd0； 或者 8'd8、4'd0。

PWM 频率是 8k，结果是：10'd1000、4'd0； 或者 8'd250、4'd2。

PWM 频率是 1k，结果是：10'd1000、4'd3； 或者 8'd250、4'd5。

PWM 频率是 100Hz，结果是：10'd625、4'd7； 或者 8'd156、4'd9。

以此类推。

外部输入 PWM 频率等于 TF\_CLK 除以底数，再除以 2 的指数次方。

$$PWM = TF\_CLK / (TFDAT.DAT \times 2^{TFDAT.SHIFT})$$

如果 PWM 信号很长时间为固定电平，没有上升沿跳变。则指数、底数计数至最大后保持不变，但是这个数据也不会更新到寄存器 SHIFT[3: 0]、DAT[9: 0]。

## 16. PWM 占空比检测器(TD)

TD 模块用于捕获输入 PWM 信号的占空比，主要特点如下：

- 9 位精度，用 0~1023 代表 0~100% 占空比
- 输入 PWM 信号的采样频率可支持 4 档选择，由外设时钟选择寄存器 CKSEL1 中的 TDCKS[1: 0] 寄存器设置，可以选择 PCLK、PCLK/4、PCLK/16、PCLK/64。

## 16.1. 功能框图

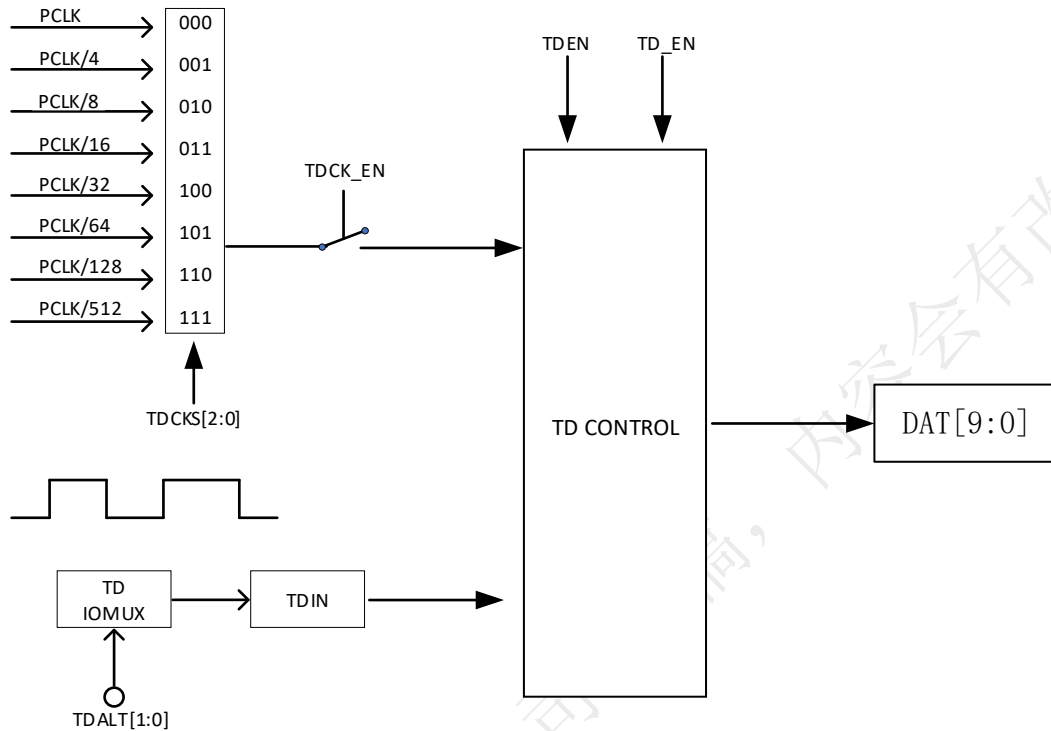


图 16-1 TD 功能框图

## 16.2. 模块寄存器总表

寄存器名	地址	结构体(.h)	描述	复位值
TDCON	0x40004800	TD	PWM占空比捕获控制寄存器	0x00
TDDAT	0x40004804	TD	PWM占空比捕获数据寄存器	0x0000

## 16.3. 寄存器说明

### 16.3.1. PWM 占空比捕获控制寄存器 (TDCON: 0x40004800)

描述位	名称	读写	描述	复位值
0	TDEN	R/W	PWM占空比捕获模块使能。 0: 关闭 1: 使能	0
3: 1	TDCKS[2: 0]	R/W	TD时钟频率选择。	000

描述位	名称	读写	描述	复位值
			000: PCLK 001: PCLK/4 010: PCLK/8 011: PCLK/16 100: PCLK/32 101: PCLK/64 110: PCLK/128 111: PCLK/512	
4	TDCKEN	R/W	TD时钟使能, 1有效(使用TD模块时, TD_EN也要使能)。	0
7: 5			保留	

### 16.3.2. PWM 占空比捕获数据 (TDDAT: 0x40004804)

描述位	名称	读写	描述	复位值
9: 0	DAT[9: 0]	R	10位精度, DAT[9: 0]构成捕获数据。	0x000
15: 10			保留	

## 16.4.功能描述

### 16.4.1. PWM 占空比捕获功能

PWM 占空比捕获模块可以捕获“40~100Khz”频率范围的 PWM 信号占空比,用于电机 PWM 调速。

TDCKS[2: 0]	采样时钟	可精确捕获 PWM 频率范围
000	时钟选择 PCLK	5khz~100Khz (TDCK=16Mhz)
001	时钟选择 PCLK/4	
010	时钟选择 PCLK/8	1khz~6Khz (TDCK=4Mhz)
011	时钟选择 PCLK/16	
100	时钟选择 PCLK/32	200hz~1.2khz(TDCK=1Mhz)
101	时钟选择 PCLK/64	
110	时钟选择 PCLK/128	

111	时钟选择 PCLK/512	40hz~250hz (TDCK=250khz)
-----	---------------	--------------------------

## 17. 数学运算协处理器(MATH)

MATH 模块包含两个独立的运算模块(除法、CORDIC)，以支持密集型数学运算。

具有以下特性:

- 由操作数预处理和结果后处理能力
- 支持除法模块和 CORDIC 模块之间数据结果链接

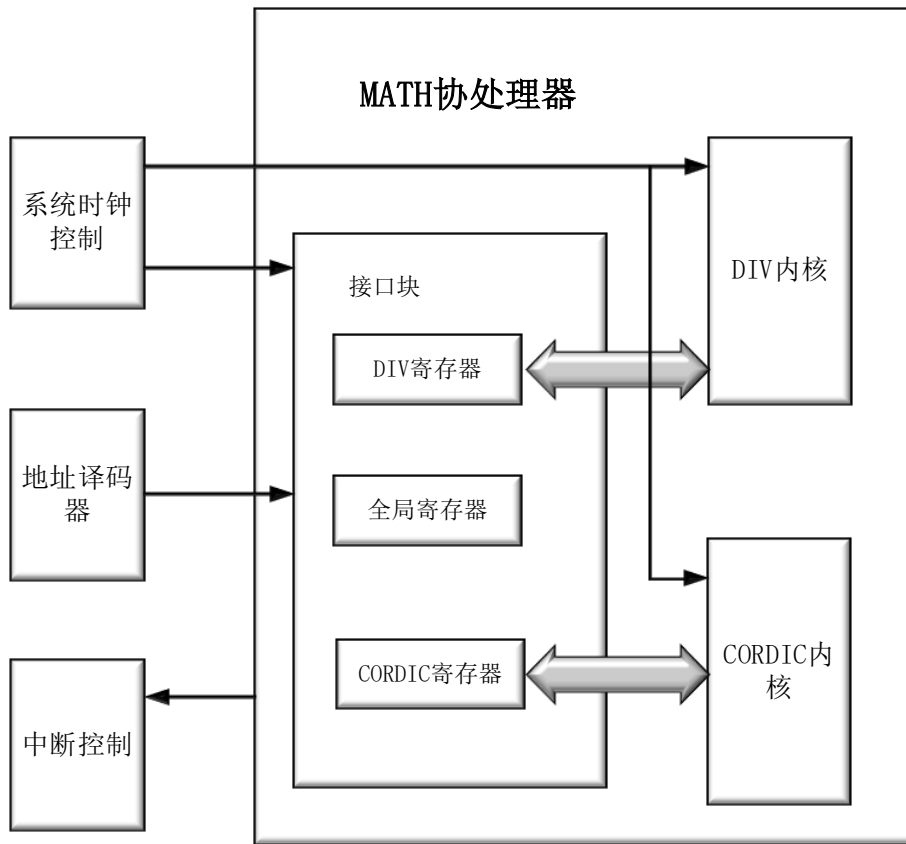


图 17 MATH 协处理系统框图

## 17.1. 模块寄存器总表

寄存器名	地址	结构体(.h)	寄存器描述	复位值
DIVCON	0x40015000	DIV	除法控制寄存器	0x00000000
DIVSTA	0x40015004	DIV	除法状态寄存器	0x00
DIVDVD	0x40015008	DIV	被除数寄存器	0x00000000
DIVDVS	0x4001500C	DIV	除数寄存器	0x00000000
DIVQUOT	0x40015010	DIV	商寄存器	0x00000000
DIVRMD	0x40015014	DIV	余数寄存器	0x00000000
DIVSTA_CLR	0x40015018	DIV	除法状态清零寄存器	0x00000000
CORDCON	0x40015020	CORDIC	CORDIC控制寄存器	0x00000000
CORDSTA	0x40015024	CORDIC	CORDIC状态寄存器	0x00
CORDXI	0x40015028	CORDIC	CORDIC X输入数据寄存器	0x00000000
CORDYI	0x4001502C	CORDIC	CORDIC Y输入数据寄存器	0x00000000
CORDZI	0x40015030	CORDIC	CORDIC Z输入数据寄存器	0x00000000
CORDXO	0x40015034	CORDIC	CORDIC X结果数据寄存器	0x00000000
CORDYO	0x40015038	CORDIC	CORDIC Y结果数据寄存器	0x00000000
CORDZO	0x4001503C	CORDIC	CORDIC Z结果数据寄存器	0x00000000

CORDSTA_CLR	0x40015040	CORDIC	CORDIC状态清零寄存器	0x00000000
-------------	------------	--------	---------------	------------

## 17.2. 寄存器说明

### 17.2.1. 控制寄存器 (DIVCON: 0x40015000)

描述位	名称	读写	描述	复位值
0	SOC	R/W	除法计算启动位，硬件1时钟周期后自动清零。 0: 无效 1: 启动除法计算，	0
1	USIGN	R/W	符号除法使能。 0: 无符号除法计算 1: 有符号除法计算	0
2			保留	-
3	QSDIR	R/W	商移位方向。 0: 左移 1: 右移	0
7: 4	QSC[3: 0]	R/W	商移位计数。 0: 不移位 除法计算结束后，对商进行移位的位数	0
11: 8	NSC[3: 0]	R/W	被除数移位计数，左移。 0: 不移位 除法计算开始前，对被除数进行移位的位数	0
15: 12	DSC[3: 0]	R/W	除数移位计数，右移。 0: 不移位 除法计算开始前，对除数进行移位的位数	0
18: 16	DVDRC[2: 0]	R/W	DIV被除数寄存器链接控制，当发生结果链接触发事件时，DIV的DVD操作数寄存器将被所选择的结果寄存器值更新： 000: 没有选择结果链接寄存器 001: QUOT是结果链接寄存器，除法EOC触发 010: RMD是结果链接寄存器，除法EOC触发 011: XO是结果链接寄存器，左移8位，CORDIC EOC触发 100: YO是结果链接寄存器，左移8位，CORDIC EOC触发	000

描述位	名称	读写	描述	复位值
			101: ZO是结果链接寄存器，左移8位，CORDIC EOC触发 110: 保留 111: 保留	
21: 19	DVSR[2: 0]	R/W	DIV除数寄存器链接控制，当发生结果链接触发事件时，DIV的DVS操作数寄存器将被所选择的结果寄存器值更新： 000: 没有选择结果链接寄存器 001: QUOT是结果链接寄存器，除法EOC触发 010: RMD是结果链接寄存器，除法EOC触发 011: XO是结果链接寄存器，左移8位，CORDIC EOC触发 100: YO是结果链接寄存器，左移8位，CORDIC EOC触发 101: ZO是结果链接寄存器，左移8位，CORDIC EOC触发 110: 保留 111: 保留	000
22	DIVEOCIEN	R/W	除法计算结束事件中断使能。 0: 禁止除法计算结束事件触发中断 1: 允许除法计算结束事件触发中断	0
23	DIVERRIEN	R/W	除法计算错误事件标志。 0: 禁止除法错误事件触发中断 1: 允许除法错误事件触发中断	0
31: 24			保留	

### 17.2.2. 除法状态寄存器 (DIVSTA : 0x40015004)

描述位	名称	读写	描述	复位值
0	DIVBUSY	R	运算忙标志。 0: 空闲 1: 正在进行运算	0
2: 1			保留	
3	DIVEOC	R/W	计算结束事件标志，写1清零。 0: 未检测到计算结束事件 1: 检测到计算结束事件	0

描述位	名称	读写	描述	复位值
6: 4			保留	
7	DIVERR	R/W	计算错误事件标志，写1清零。 0: 未检测到错误事件 1: 检测到错误事件	0

### 17.2.3. 除法状态清零寄存器 (DIVSTA\_CLR : 0x40015018)

描述位	名称	读写	描述	复位值
2: 0			保留	
3	DIVEOC_CLR	W	除法计算结束事件标志位写1清零。	
6: 4			保留	
7	DIVERR_CLR	W	除法计算错误事件标志位写1清零。	

### 17.2.4. 被除数寄存器 (DIVDVD: 0x40015008)

该寄存器可由软件写入，如果结果链接使能，则由硬件写入。

描述位	名称	读写	描述	复位值
31: 0	DVD[31: 0]	R/W	被除数寄存器。	0x00000000

### 17.2.5. 除数寄存器 (DIVDVS: 0x4001500C)

该寄存器可由软件写入，如果结果链接使能，则由硬件写入。

描述位	名称	读写	描述	复位值
31: 0	DVS[31: 0]	R/W	除数寄存器。	0x00000000

### 17.2.6. 商寄存器 (DIVQUOT : 0x40015010)

该寄存器在除法运算完成后，由硬件自动更新。

描述位	名称	读写	描述	复位值
31: 0	QUOT[31: 0]	R	商寄存器。	0x00000000

### 17.2.7. 余数寄存器 (DIVRMD : 0x40015014)

该寄存器在除法运算完成后，由硬件自动更新。

描述位	名称	读写	描述	复位值
31: 0	RMD[31: 0]	R	余数寄存器。	0x00000000

### 17.2.8. CORDIC 控制寄存器(CORDCON : 0x40015020)

描述位	名称	读写	描述	复位值
0	SOC	R/W	计算启动位，硬件1时钟周期后自动清零。 0: 无效 1: 启动CORDIC计算，	0
2: 1	MODE[1: 0]	R/W	功能模式选择。 00: reserved(无效) 01: circle 10: atanth 11: line	00
3	ROTVEC	R/W	旋转向量模式选择。 0: vector 1: rotate	0
4	USIGN		符号除法使能。 0: 无符号计算 1: 有符号计算	0
6: 5	MPS[1: 0]	R/W	CORDIC迭代计算完成后，XO、YO除以该系数得到最终结果。 00: 除以1 01: 除以2 10: 除以4 11: 除以8	00
7	LINEAROPT	R/W	线性函数运算控制。 0: 有理数格式 1: 整数格式	0
8	KPX	R/W	最后一次XO结果作为XI操作数进行新的计算。 0: XI操作数由软件写入XI寄存器 1: XI操作数来之XO寄存器	0
9	KPY	R/W	最后一次YO结果作为YI操作数进行新的计算。 0: YI操作数由软件写入YI寄存器 1: YI操作数来之YO寄存器	0
10	KPZ	R/W	最后一次ZO结果作为ZI操作数进行新的计算。 0: ZI操作数由软件写入ZI寄存器 1: ZI操作数来之ZO寄存器	0
12: 11	XRC[1: 0]	R/W	CORDIC XI寄存器链接控制，当发生结果链接触发事件时，CORDIC的XI操作数寄存器将被所选择的结果寄存器值更新:	00

描述位	名称	读写	描述	复位值
			00: 没有选择结果链接寄存器 01: QUOT是结果链接寄存器, 右移8位, 除法EOC触发 10: RMD是结果链接寄存器, 右移8位, 除法EOC触发 11: 保留	
14: 13	YRC[1: 0]	R/W	CORDIC YI寄存器链接控制, 当发生结果链接触发事件时, CORDIC的YI操作数寄存器将被所选择的结果寄存器值更新: 00: 没有选择结果链接寄存器 01: QUOT是结果链接寄存器, 右移8位, 除法EOC触发 10: RMD是结果链接寄存器, 右移8位, 除法EOC触发 11: 保留	00
16: 15	ZRC[1: 0]	R/W	CORDIC ZI寄存器链接控制, 当发生结果链接触发事件时, CORDIC的ZI操作数寄存器将被所选择的结果寄存器值更新: 00: 没有选择结果链接寄存器 01: QUOT是结果链接寄存器, 右移8位, 除法EOC触发 10: RMD是结果链接寄存器, 右移8位, 除法EOC触发 11: 保留	00
17	CORDEOC IEN	R/W	CORDIC计算结束事件标志。 0: 禁止CORDIC计算结束事件触发中断 1: 允许CORDIC计算结束事件触发中断	0
18	CORDERRIEN	R/W	除法CORDIC错误事件标志。 0: 禁止CORDIC错误事件触发中断 1: 允许CORDIC错误事件触发中断	0
31: 19			保留	

### 17.2.9. CORDIC 状态寄存器 (CORDSTA : 0x40015024)

描述位	名称	读写	描述	复位值
0	BUSY	R	运算忙标志。 0: 空闲 1: 正在进行运算	0
2: 1			保留	

描述位	名称	读写	描述	复位值
3	EOC	R/W	计算结束事件标志，写1清零。 0: 未检测到计算结束事件 1: 检测到计算结束事件	0
6: 4			保留	
7	ERR	R/W	计算错误事件标志，写1清零。 0: 未检测到错误事件 1: 检测到错误事件	0

### 17.2.10. ORDIC 状态清零寄存器(CORDSTA\_CLR : 0x40015040)

描述位	名称	读写	描述	复位值
2: 0			保留	
3	EOC	W	CORDIC计算结束事件标志位，写1清零。	0
6: 4			保留	
7	ERR	W	CORDIC计算错误事件标志位，写1清零。	0

### 17.2.11. XI 寄存器(CORDXI : 0x40015028)

该寄存器可由软件写入，如果结果链接使能，则由硬件写入。在向量模式，XI 不能为负数，需要取补码。

描述位	名称	读写	描述	复位值
23: 0	XI[23: 0]	R/W	XI数据寄存器	0x00000000
31: 24			保留	

### 17.2.12. YI 寄存器(CORDYI : 0x4001502C)

该寄存器可由软件写入，如果结果链接使能，则由硬件写入。

描述位	名称	读写	描述	复位值
23: 0	YI[23: 0]	R/W	YI数据寄存器	0x00000000
31: 24			保留	

### 17.2.13. ZI 寄存器(CORDZI: 0x40015030)

该寄存器可由软件写入，如果结果链接使能，则由硬件写入。

描述位	名称	读写	描述	复位值
23: 0	ZI[23: 0]	R/W	ZI数据寄存器	0x00000000

描述位	名称	读写	描述	复位值
31: 24			保留	

### 17.2.14. XO 寄存器(CORDXO: 0x40015034)

该寄存器在运算完成后，由硬件自动更新。

描述位	名称	读写	描述	复位值
23: 0	XO[23: 0]	R	X结果寄存器	0x00000000
31: 24			保留	

### 17.2.15. YO 寄存器(CORDYO: 0x40015038)

该寄存器在运算完成后，由硬件自动更新。

描述位	名称	读写	描述	复位值
23: 0	YO[23: 0]	R	Y结果寄存器	0x00000000
31: 24			保留	

### 17.2.16. ZO 寄存器(CORDZO: 0x4001503C)

该寄存器在运算完成后，由硬件自动更新。

描述位	名称	读写	描述	复位值
23: 0	ZO[23: 0]	R	Z结果寄存器	0x00000000
31: 24			保留	

## 17.3.除法

支持 32bit 有/无符号操作，32 个周期完成；向下支持 32/16 的有符号除法。

## 17.4.CORDIC

CORDIC 协处理器能计算三角函数、线性函数、双曲线函数及相关函数。

CORDIC 算法是一个迭代过程，截断误差是这个过程的固有性质。CORDIC 协处理器每次计算使用 27 次迭代，并且使用至少为 32 位的内核数据宽度，因而可获得较高的精度。使用该算法的主要优点在于比软件有更快的计算速度，并且可获得高精度。通用的 CORDIC 算法使用以下公式，因子  $m$  控制矢量旋转并且选择三角函数、线性函数和双曲函数的角度：

$$x_{i+1} = x_i - m * d_i * y_i * 2^{-i}$$

$$y_{i+1} = y_i + d_i * x_i * 2^{-i}$$

$$z_{i+1} = z_i - d_i * e_i$$

其中:

m=1 三角函数(基本 CORDIC),  $e_i = \text{atan}(2^{-i})$

m=0 线性函数,  $e_i = 2^{-i}$

m=-1 双曲函数,  $e_i = \text{atanh}(2^{-i})$

CORDIC 协处理器的关键特性如下:

- 运算模式  
支持所有的 CORDIC 运算模式, 用于解决圆函数(三角函数)、线性(乘-加, 除-加)函数和双曲函数问题。集成的查找表(LUT)可用于所有的运算模式
- 圆向量模式: 支持满范围 $[-2^{23}, (2^{23} - 1)]$ 内的初始 X 和 Y 数据值, 来计算角度和幅值。
- 圆旋转模式: 支持满范围 $[-2^{23}, (2^{23} - 1)]$ 内的初始 Z 数据值, 代表范围 $[-\pi, ((2^{23} - 1)/2^{23})\pi]$ 内的角度, 来求解三角学问题。
- 24 位可访问的数据宽度  
X/Y: 32 位内核数据宽度外加 2 个符号扩展位以及 X 和 Y 各一位溢出位  
Z: 29 位内核数据宽度外加 1 个符号扩展位和 Z 的一位溢出位  
KEEP: 保持最近一次结果值, 用于在内核寄存器中为进行一次新计算
- 每次运算进行 27 次迭代; 从启动位(ST)置 1 到计算结束标志置 1 需要 30 个内核时钟周期或更少。
- 补码数据处理  
唯一的例外: 具有用户可选项的 X 结果数据(对于无符号结果)。
- 接受的 X 和 Y 数据一般为整数或有理数; X 和 Y 必须是相同的数据格式。
- 截断误差  
由于低有效位的截断效应, 一次 CORDIC 计算的结果可能返回一个近似值。  
CORDIC 计算出的结果数据精度高, 尤其是在圆模式下。
- 中断  
一次计算完成时可触发中断, 具有中断使能和中断标志
- CORDIC 应用

用例	应用
角度计算	EPS, 电机控制
派克变换	电机控制
Y+XZ	数字滤波, PI控制回路

### 17.4.1. 结构框图

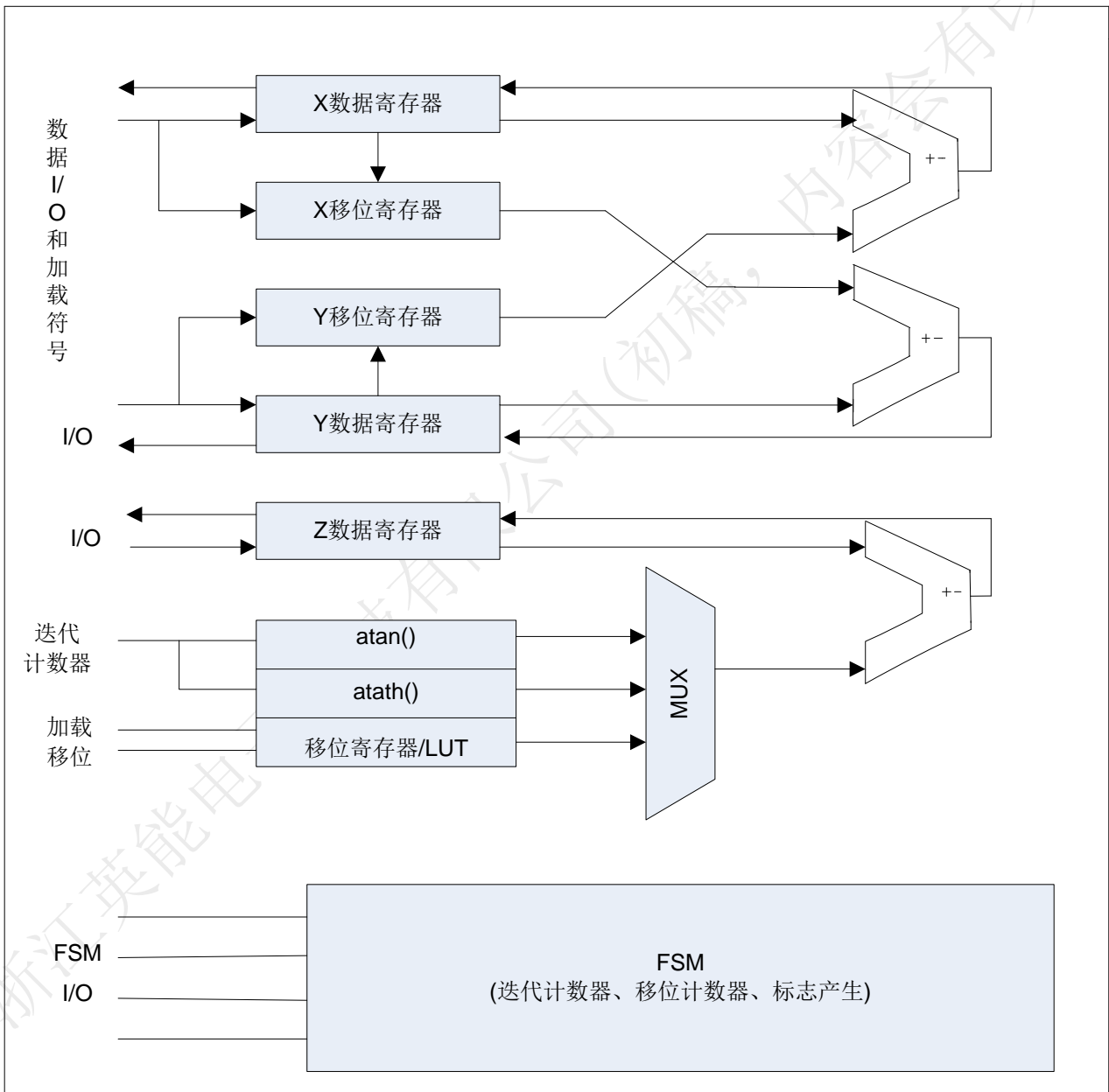


图 17-4-1 MATH 协处理功能框图

- CORDIC 计算出的数据包含一个固有的由旋转或向量引起的增益系数  $K$ 。每个 CORDIC 函数的  $K$  值是不同的:

区域	描述
三角函数	1.646760258121
双曲函数	0.828159360960
线性函数	1

### 17.4.2. 工作模式和对应结果

工作模式的概览表中,  $X$ 、 $Y$  和  $Z$  代表初始化数据,  $X_{\text{final}}$ 、 $Y_{\text{final}}$  和  $Z_{\text{final}}$  代表当所有处理过程都结束并且 **BUSY** 不再有效使得最终结果数据。

函数	旋转模式	向量模式
	$d_i = \text{sign}(z_i), z_i \rightarrow 0$	$d_i = -\text{sign}(y_i), y_i \rightarrow 0$ $X > 0$
<b>圆函数</b> $m = 1$ $e_i = \text{atan}(2^{-i})$	$X_{\text{final}} = K[X \cos(Z) - Y \sin(Z)] / \text{MPS}$ $Y_{\text{final}} = K[Y \cos(Z) + X \sin(Z)] / \text{MPS}$ $Z_{\text{final}} = 0$ 其中 $K \approx 1.646760258121$	$X_{\text{final}} = K \sqrt{X^2 + Y^2} / \text{MPS}$ $Y_{\text{final}} = 0$ $Z_{\text{final}} = Z + \text{atan}(Y/X)$ 其中 $K \approx 1.646760258121$
	为了计算 $\cos(Z)$ and $\sin(Z)$ , 令 $X = 1/K, Y = 0$ 。 定义域: 由于有预处理逻辑, 所以支持满范围的 $X, Y$ 和 $Z$ 。	为了计算向量的模 ( $\sqrt{x^2+y^2}$ ), 令 $X = x/K, Y = y/K$ 。 定义域: 由于有预处理和后处理逻辑, 所以支持满范围的 $X$ 和 $Y$ 。  用于计算 $\text{atan}(Y/X)$ , 令 $Z = 0$ 。 定义域: 满范围的 $X$ 和 $Y, X = 0$ 除外。
	关系: $\tan(v) = \sin(v) / \cos(v)$	关系: $\text{acos}(w) = \text{atan}[\sqrt{1-w^2} / w]$ $\text{asin}(w) = \text{atan}[w / \sqrt{1-w^2}]$
<b>线性函数</b> $m = 0$ $e_i = 2^{-i}$	$X_{\text{final}} = X / \text{MPS}$ $Y_{\text{final}} = [Y + X Z] / \text{MPS}$ $Z_{\text{final}} = 0$	$X_{\text{final}} = X / \text{MPS}$ $Y_{\text{final}} = 0$ $Z_{\text{final}} = Z + Y/X$
	为了计算 $X \cdot Z$ , 令 $Y = 0$ 。 定义域: $ Z  \leq 2$ 。	为了计算比值 $Y/X$ , 令 $Z = 0$ 。 定义域: $ Y/X  \leq 2, X > 0$ 。

函数	旋转模式	向量模式
<b>双曲函数</b> $m = -1$ $e_1 = \operatorname{atanh}(2^{-1})$	$X_{\text{final}} = k[X \cosh(Z) + Y \sinh(Z)] / \text{MPS}$ $Y_{\text{final}} = k[Y \cosh(Z) + X \sinh(Z)] / \text{MPS}$ $Z_{\text{final}} = 0$ 其中 $k \approx 0.828159360960$	$X_{\text{final}} = k \sqrt{X^2 - Y^2} / \text{MPS}$ $Y_{\text{final}} = 0$ $Z_{\text{final}} = Z + \operatorname{atanh}(Y / X)$ 其中 $k \approx 0.828159360960$
	为了计算 $\cosh(Z)$ 、 $\sinh(Z)$ 和 $e^Z$ , 令 $X = 1 / k, Y = 0$ 。 定义域: $ Z  \leq 1.11\text{rad}, Y = 0$ 。	为了计算 $\sqrt{x^2 - y^2}$ , 令 $X = x / k, Y = y / k$ 。 定义域: $ y  <  x , X > 0$ 。  为了计算 $\operatorname{atanh}(Y / X)$ , 令 $Z = 0$ 。 定义域: $ \operatorname{atanh}(Y / X)  \leq 1.11\text{rad}, X > 0$ 。
	关系: $\tanh(v) = \sinh(v) / \cosh(v)$ $e^v = \sinh(v) + \cosh(v)$ $w^t = e^{t \ln(w)}$	关系: $\ln(w) = 2 \operatorname{atanh}[(w-1) / (w+1)]$ $\sqrt{w} = \sqrt{(w+0.25)^2 - (w-0.25)^2}$ $\operatorname{acosh}(w) = \ln[w + \sqrt{1-w^2}]$ $\operatorname{asinh}(w) = \ln[w + \sqrt{1+w^2}]$

### 17.4.3. X、Y、Z 数据格式

函数	数据		模式	
			旋转	向量
圆函数	X	初始	24位补码	
		结果	24位补码	24位补码/24位无符号数 (CON.X_USIGN=1)
	Y	初始	24位补码	
		结果		
	Z	初始		
		结果		
线性函数	X	初始	24位补码	
		结果		
	Y	初始		
		结果		
	Z	初始	S4.19(1个符号位+4位整数+19位小数)	
		结果		
双曲函数	X	初始	24位补码	
		结果		

函数	数据		模式	
			旋转	向量
	Y	初始		
		结果		
Z	初始			
	结果			

在所有操作模式下，返回一个规格化的结果数据到 X 和 Y，如下面的方程所示：

$$x \text{ 或 } y \text{ 结果数据} = \frac{\text{CORDIC 计算数据}}{\text{MPS}}$$

图 17-4-3

另一方面，根据所使用的 CORDIC 函数不同，对 Z 结果数据的解读是不同的：

对于线性函数，对 CORDIC 计算的 Z 数据没有额外的处理过程，它本身就被直接当作结果数据。可访问的 Z 结果是一个实数，表示为有符号的 4Q19。Z 数据总是被作为有符号小数 S4.24 处理(可按 S4.19 访问，格式是有符号的 4Q19)。

对于三角函数和双曲函数，可访问的 Z 结果数据是一个规格化的整数值，其范围为 $[-2^{23}, (2^{23}-1)]$ ，代表的角度范围为 $[-\pi, ((2^{31}-1)/2^{31})\pi]$ 。服务请求处理要求将 Z 数据解读如下： $2^{23} = 8388608$

$$\begin{aligned} \text{输入 Z 初始值} &= \text{实际 Z 初始值(弧度)} \times (2^{23}/\pi) \\ \text{实际 Z 初始值(弧度)} &= \text{Z 输出结果} \times (\pi/2^{23}) \end{aligned}$$

#### 17.4.4. 收敛域

对于结果数据的收敛域，根据所有的工作模式不同，初始数据的值或模是有限制的，相应的数据格式也有限制。以下是关于 CORDIC 结果数据收敛的一般适用结论。

##### 旋转模式：

Z 数据必须收敛于 0。初始 Z 数据必须等于或小于  $\sum d_i \cdot e_i$ ，其中  $e_i$  总是随迭代次数  $i$  的增大而减小。换句话说， $|Z| \leq \text{LUT}$  的和。对于圆函数，这意味着  $|Z| \leq$  代表 1.74 弧度的整数值。对于线性函数， $|Z| < 2$ 。对于双曲函数， $|Z| \leq$  代表 1.11 弧度的整数值。

##### 向量模式：

Y 数据必须收敛于 0。X 和 Y 的初始值被 Z 函数限定，而 Z 函数取决于对应的 LUT。对于圆函数，这意味着  $|\text{atan}(Y/X)| \leq 1.74$  弧度。在向量模式，还要求  $X > 0$ 。

##### 圆旋转模式：

支持满范围  $[-2^{31}, (2^{31}-1)]$  的 Z 输入，代表的角度范围是  $[-\pi, ((2^{31}-1)/2^{31})\pi]$ 。要得到收敛的结果，则必须  $|Z| \leq 8388608 \times 1.74/\pi$ 。

对 X 和 Y 初始输入没有限制，但要注意溢出，可以通过 MPS 克服溢出。

**圆向量模式:**

支持满范围 $[-2^{31}, (2^{31} - 1)]$ 的 X 和 Y 输入,但要注意溢出,可以通过 MPS 克服溢出。  
 Z 初始值应当满足 $|Z| \leq \pi/2$ ,以防止可能的 Z 结果数据溢出。

### 17.4.5. 反正切和双曲反正切查找表

反正切查找表(atan LUT)和双曲反正切查找表(atanh LUT)分别为 29 位和 30 位宽。反正切 LUT 的每一项被分成一个符号位(MSB)和后面跟随的 28 位整数部分。对于双曲反正切 LUT。每一项有一个重复位(MSB),后面跟随一个符号位,然后是 28 位的整数部分。

- 数据格式为 S29 的反正切 LUT( $2^{28} \times \text{弧度}/\pi$ )

迭代序号	十六进制格式的 atan( $2^{-i}$ )定标值	迭代次数	十六进制格式的 atan( $2^{-i}$ )定标值
i=0	4000000(45°)	i=14	145F
i=1	25C80A4(26.5651°)	i=15	A30
i=2	13F670B	i=16	518
i=3	A2223B	i=17	28C
i=4	5161AB	i=18	146
i=5	28BAFC	i=19	A3
i=6	145EC4	i=20	51
i=7	A2F8B	i=21	29
i=8	517CA	i=22	14
i=9	28BE6	i=23	A
i=10	145F3	i=24	5
i=11	A2FA	i=25	3
i=12	517D	i=26	1
i=13	28BE		

- 数据格式为 S29 的双曲反正切 LUT

迭代序号	十六进制格式的 atanh( $2^{-i}$ )定标值	迭代次数	十六进制格式的 atan( $2^{-i}$ )定标值
i=0	-	i=14	145F
i=1	2CC2F12	i=15	A30
i=2	14D01AC	i=16	518
i=3	A3D4E0	i=17	28C
i=4	5197FC	i=18	146
i=5	28C1C7	i=19	A3
i=6	145F9D	i=20	51
i=7	A2FA6	i=21	29
i=8	517CE	i=22	14
i=9	28BE6	i=23	A

迭代序号	十六进制格式的 $\operatorname{atanh}(2^{-i})$ 定标值	迭代次数	十六进制格式的 $\operatorname{atan}(2^{-i})$ 定标值
i=10	145F3	i=24	5
i=11	A2FA	i=25	3
i=12	517D	i=26	1
i=13	28BE		

Z 数据是实际角度的规格化表示。例如内部换算将  $[-\pi, ((2^{23} - 1)/2^{23})\pi]$  等价于  $[-2^{28}, (2^{28} - 1)]$ 。当被寻址时，最后 5 个 LSB 位被截断，24 位数据被传送到数据总线。

从用户的角度看，角度  $[-\pi, ((2^{23} - 1)/2^{23})\pi]$  由范围  $[-2^{23}, (2^{23} - 1)]$  来代表。

#### 17.4.6. 线性函数的模拟查找表

线性函数的模拟 LUT 实际是一个移位寄存器。该模拟 LUT 有一个整数位(MSB)，后面跟随 23 位 1Q23 格式的小数位。

在线性函数中，Z 是一个实数，内部的 Z 数据是格式为 4Q19 的有符号数。外部读取的数据是小数部分最后 5 位被截断后的结果，这就导致读取的数据是一个符号位后面跟随 4 位整数部分，最后是 19 位的小数部分。

### 17.5. 结果链接

MATH 协处理支持 DIV 和 CORDIC 之间的结果链接。

对于 DIV，操作数 DVD、DVS 可以被结果寄存器(DIV 的 QUOT、RMD, CORDIC 的 XO、YO、ZO)值更新。

对于 CORDIC，操作数 XI、YI、ZI 可以被结果寄存器(DIV 的 QUOT、RMD)值更新。

- DIV 被除数寄存器链接

当发生结果链接触发事件时，DIV 的 DVD 操作数寄存器将被所选择的结果寄存器值更新：

000: 没有选择结果链接寄存器

001: QUOT 是结果链接寄存器

010: RMD 是结果链接寄存器

011: XO 是结果链接寄存器

100: YO 是结果链接寄存器

101: ZO 是结果链接寄存器

- DIV 除数寄存器链接

当发生结果链接触发事件时，DIV 的 DVS 操作数寄存器将被所选择的结果寄存器值更新：

000: 没有选择结果链接寄存器

001: QUOT 是结果链接寄存器

010: RMD 是结果链接寄存器

011: XO 是结果链接寄存器

100: YO 是结果链接寄存器

101: ZO 是结果链接寄存器

- CORDIC XI 寄存器链接

当发生结果链接触发事件时，CORDIC 的 XI 操作数寄存器将被所选择的结果寄存器值更新:

00: 没有选择结果链接寄存器

01: QUOT 是结果链接寄存器

10: RMD 是结果链接寄存器

- CORDIC YI 寄存器链接

当发生结果链接触发事件时，CORDIC 的 YI 操作数寄存器将被所选择的结果寄存器值更新:

00: 没有选择结果链接寄存器

01: QUOT 是结果链接寄存器

10: RMD 是结果链接寄存器

- CORDIC ZI 寄存器链接

当发生结果链接触发事件时，CORDIC 的 ZI 操作数寄存器将被所选择的结果寄存器值更新:

00: 没有选择结果链接寄存器

01: QUOT 是结果链接寄存器

10: RMD 是结果链接寄存器

## 18. 基础异步通信收发器(BUART)

UART 模块，只支持 10 位异步全双工收发通信方式。支持发送中断、接收中断、发送错误中断、接收错误中断。

### 18.1. 功能框图

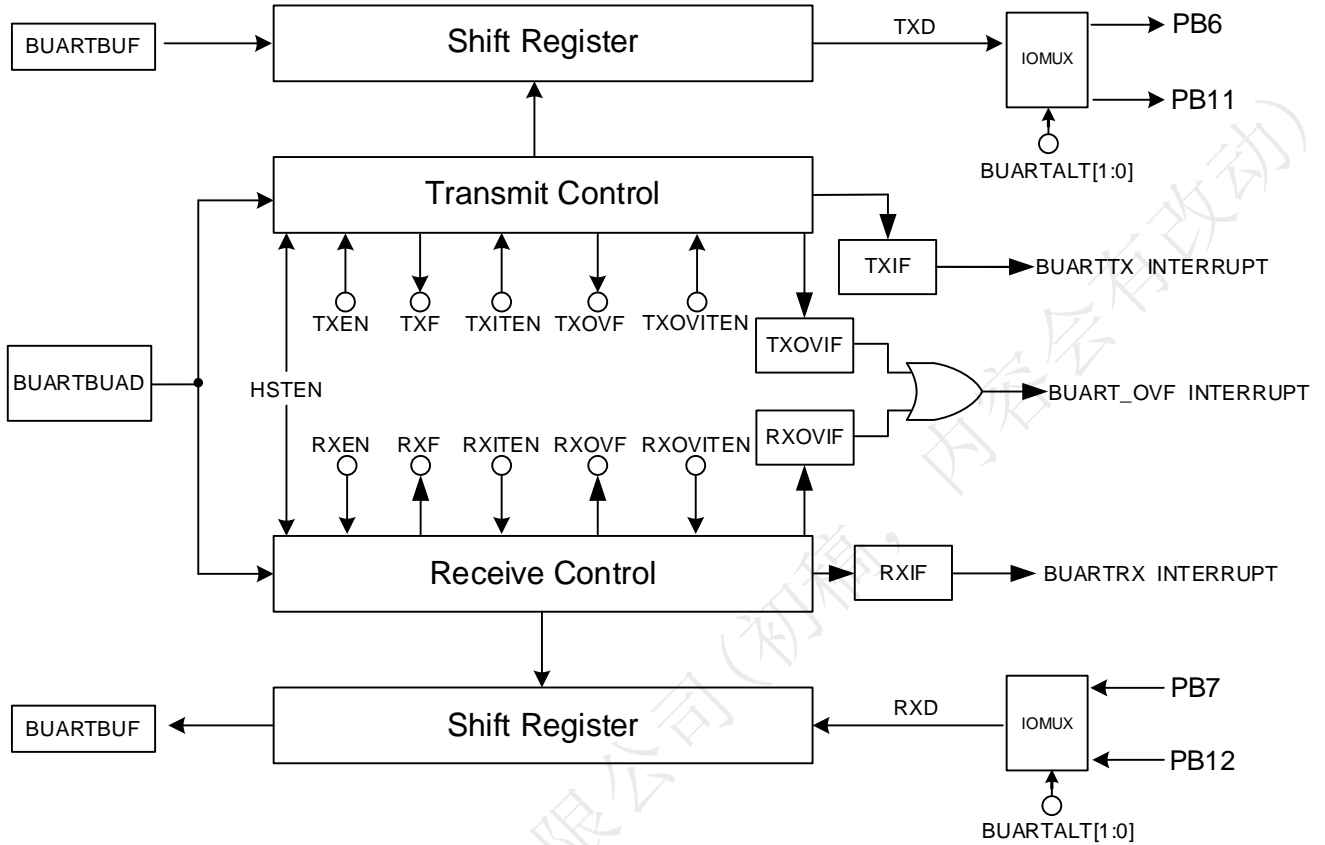


图 18-1 BUART 功能框图

### 18.2. 模块寄存器总表

寄存器名	地址	结构体(.h)	寄存器描述	复位值
BUARTBUF	0x40006000	BUART	接收/发送数据寄存器	0x00
BUARTSTA	0x40006004	BUART	状态寄存器	0x00
BUARTCON	0x40006008	BUART	控制寄存器	0x00
BUARTFLG	0x4000600C	BUART	中断寄存器	0x00
BUARTBAUD	0x40006010	BUART	波特率分频寄存器	0x00000000

## 18.3. 寄存器说明

### 18.3.1. 发送/接收数据寄存器(BUARTBUF: 0x40006000)

描述位	名称	读写	描述	复位值
7: 0	BUARTBUF[7: 0]	R/W	发送/接收数据寄存器。	0x00

### 18.3.2. 状态寄存器 (BUARTSTA : 0x40006004)

描述位	名称	读写	描述	复位值
0	TXF	R/W	1表示待发送数据已写入缓冲，载入发送移位寄存器时硬件自动清零。	0
1	RXF	R/W	1表示接收的数据已写入缓冲，读取该寄存器时硬件自动清零。	0
2	TXOVF	R/W	1表示发送数据写入错误(在上一笔数据载入移位寄存器前又对tx_buf写数据)，写1清零，或对tx_overflow_intr写1清零时一起清零。	0
3	RXOVF	R/W	1表示接收数据缓冲满(在上一笔数据被读取前又有新数据接收进来)，写1清零，或对rx_overflow_intr写1清零时一起清零。	0
7: 4			保留	

### 18.3.3. 控制寄存器 (BUARTCON: 0x40006008)

描述位	名称	读写	描述	复位值
0	TXEN	R/W	发送使能。 0: 禁止发送 1: 允许发送	0
1	RXEN	R/W	接收使能。 0: 禁止接收 1: 允许接收	0
2	TXITEN	R/W	发送中断触发使能。 0: 禁止触发发送中断 1: 允许触发发送中断	0
3	RXITEN	R/W	接收中断触发使能。 0: 禁止触发接收中断 1: 允许触发接收中断	0
4	TXOVITEN	R/W	TxOverRun中断触发使能。	0

描述位	名称	读写	描述	复位值
			0: 禁止触发中断 1: 允许触发中断	
5	RXOVITEN	R/W	RxOverRun中断触发使能。 0: 禁止触发中断 1: 允许触发中断	0
6	HSTEN	R/W	高速测试模式使能。 0: 禁止 1: 允许	0
7			保留	

#### 18.3.4. 中断标志寄存器 (BUARTFLG : 0x4000600C)

描述位	名称	读写	描述	复位值
0	TXIF	R/W	1表示待发送数据已载入发送移位寄存器，写1清零。	0
1	RXIF	R/W	1表示新接收的数据写入缓冲，写1清零。	0
2	TXOVIF	R/W	1表示发送数据写入错误中断，写1清零。	0
3	RXOVIF	R/W	1表示接收数据缓冲溢出中断，写1清零。	0
7: 4			保留	

#### 18.3.5. 波特率寄存器 (BUARTBAUD: 0x40006010)

描述位	名称	读写	描述	复位值
19: 0	BUARTBAUD	R/W	波特率计数器载入baud_div[19: 0]作为初始值，自减1到0则再次载入baud_div[19: 0]。 baud_div[19: 0]最小值为16。	0x00000
31: 20			保留	

## 19. 通用异步通信收发器(UART)

集成一个 LIN 同步间隔段、同步段、PID 段的自动检测逻辑，可以实现波特率同步和 PID 接收校验，并产生中断，实现唤醒功能。

集成一个可编程的全双工串行通信接口，可以作为通用异步接收/发送器，也可以作为同步移位寄存器，包括四种串行工作方式：

- 0: 移位寄存器方式，波特率 PCLK/12；
- 1: 10 位数据异步通信方式，波特率可变；
- 2: 11 位数据异步通信方式，波特率 PCLK/32；
- 3: 11 位数据异步通信方式，波特率可变。

波特率可以选择系统外设时钟预分频，也可以选择由 TimerA0 PWM 模式输出的 CCA00。

### 19.1.功能框图

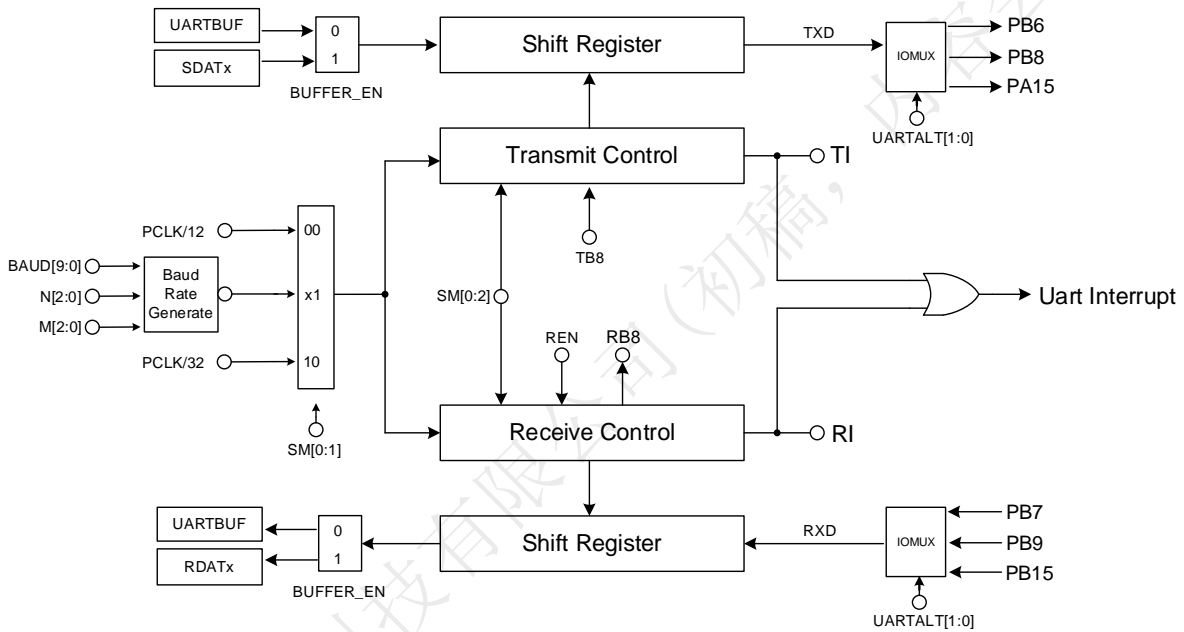


图 19-1-1 UART 功能框图

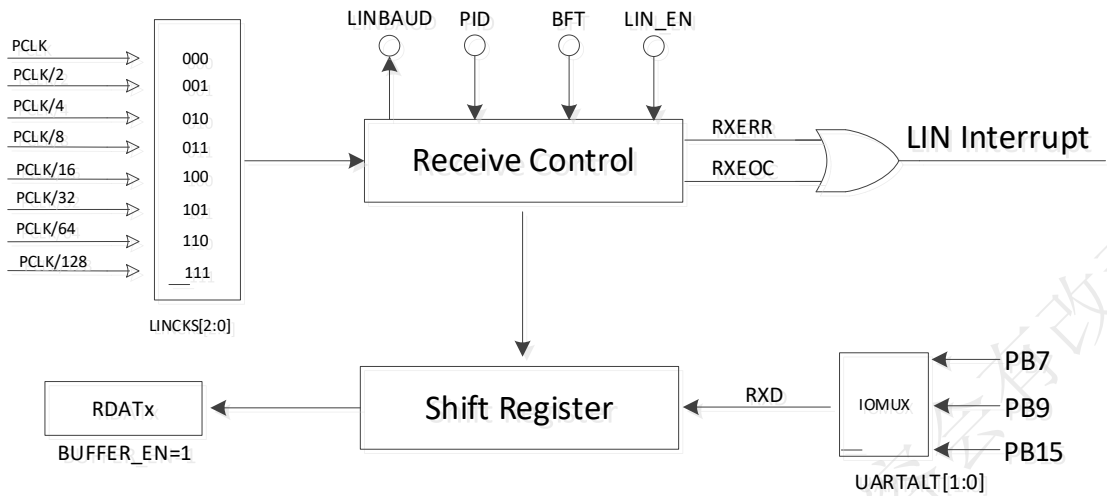


图 19-1-2 LIN 功能框图

## 19.2. 模块寄存器总表

寄存器名	地址	结构体(.h)	寄存器描述	复位值
UARTCON	0x40008800	UART	UART控制寄存器	0x00
UARTBUF	0x40008804	UART	UART数据寄存器	0x00
BAUDCFG	0x40008808	UART	UART波特率配置寄存器	0x0000
BUFCON	0x4000880C	UART	UART缓存控制寄存器	0x0000
LINCON	0x40008810	UART	LIN控制寄存器	0x0000
LINPID	0x40008814	UART	LIN PID寄存器	0x00
LINBAUD	0x40008818	UART	LIN波特率检测结果寄存器	0x0000
LINBFT	0x4000881C	UART	LIN间隔场检测时间寄存器	0xFFFF
WBUF1	0x40008820	UART	UART发送缓冲寄存器1	0x00
WBUF2	0x40008824	UART	UART发送缓冲寄存器2	0x00
WBUF3	0x40008828	UART	UART发送缓冲寄存器3	0x00
RBUF1	0x40008830	UART	UART接收缓冲寄存器1	0x00
RBUF2	0x40008834	UART	UART接收缓冲寄存器2	0x00
RBUF3	0x40008838	UART	UART接收缓冲寄存器3	0x00

## 19.3. 寄存器说明

### 19.3.1. UART 控制寄存器(UARTCON: 0x40008800)

描述位	名称	读写	描述	复位值
0	RI	R/W	接收中断标志, 写0清零。	0
1	TI	R/W	发送中断标志, 写0清零。	0
2	RB8	R/W	方式2和方式3中要接收的第9位数据, 在方式1时, 如SM2=0, RB8是接收到的停止位。方式0中, 不使用RB8。	0
3	TB8	R/W	在方式2和方式3中要发送的第9位数据, 需要时由软件置位或复位。	0
4	REN	R/W	允许接收控制位, 软件置位。 0: 禁止接收 1: 允许接收	0
5	SM2	R/W	多机通信控制器(用于方式10和11)。 0: 不使能多机通信。将8位数据送入SBUF中, 并使RI置位1, 产生中断请求 1: 使能多机通信。只有当接收到的第9位数据(RB8)为1时, 才置位RI, 产生中断请求; 当接收到的数(RB8)为0时, 将接收到的前8位数据丢弃; 方式00时, SM2必须为0; 方式01时, 如果SM2=1, 则只有收到有效的停止位时才会激活RI。	0
7: 6	SM[0: 1]	R/W	工作方式选择。 00: 同步移位寄存器方式, 波特率PCLK/12, 01: 10位数据异步通信方式, 波特率可调; 10: 11位数据异步通信方式, 波特率PCLK/32 11: 11位数据异步通信方式, 波特率可调	00

### 19.3.2. UART 缓存控制寄存器(BUFCON: 0x4000880C)

描述位	名称	读写	描述	复位值
3: 0	BUFFER_N[3: 0]	R/W	发送\接收的buffer字节数, 最大为9个字节。	0000
4	RX_BUFFER_STA	R	1表示接收完成(buffer满), 需要MCU读取数据。	0

描述位	名称	读写	描述	复位值
5	TX_BUFFER_STA	R	1表示发送完成(buffer空), MCU可以写入数据。	0
6	RX_BUFFER_CLR	W	写1清零接收完成(buffer满)标志位, 硬件自清零, 写0无效。	
7	TX_BUFFER_CLR	W	写1清零发送完成(buffer空)标志位, 硬件自清零, 写0无效。	
8	TX_BUFFER_SET	W	写1清零发送完成(buffer空)标志位, 并启动发送。硬件自清零, 写0无效。	
9	BUFFER_EN	R/W	发送\接收buffer使能位, 高有效。	0
15: 10			保留	

### 19.3.3. LIN 控制寄存器(LINCON: 0x40008810)

描述位	名称	读写	描述	复位值
0	LIN_EN	R/W	LIN检测使能位。 0: 禁止LIN检测与响应 1: 允许LIN检测与响应	0
1	LIN_RST	R/W	LIN检测重启控制。 0: 无效 1: LIN检测重启, 硬件自动清零	0
4: 2	LINCKS[2: 0]	R/W	LIN模块时钟分频选择寄存器。 000: PCLK 001: PCLK/2 010: PCLK/4 011: PCLK/8 100: PCLK/16 101: PCLK/32 110: PCLK/64 111: PCLK/128	000
5	RXERR	R	1表示PID帧通讯错误。	0
6	RXEOC	R	1表示PID帧通讯完成。	0
7	PIDERR	R	PID校验标志位。 0: LIN PID奇偶校验正确 1: LIN PID奇偶校验错误	0
8	RXERR_CLR	W	写1清零RXERR标志位。	
9	RXEOC_CLR	W	写1清零RXEOC标志位。	
10	PIDERR_CLR	W	写1清零PID校验标志位。	
15: 11			保留	

### 19.3.4. LINPID 寄存器(LINPID: 0x40008814)

描述位	名称	读写	描述	复位值
7: 0	PID[7: 0]	R	LIN PID寄存器。	0x00

### 19.3.5. LIN BAUD 寄存器(LINBAUD: 0x40008818)

描述位	名称	读写	描述	复位值
11: 0	LINBAUD[11: 0]	R	LIN 同步波特率寄存器。等于LIN模块时钟除以外部输入的LIN通讯波特率，可以通过LINBAUD反推外部输入的波特率。	0x000
15: 12			保留	

### 19.3.6. LIN BFTIME 寄存器(LINBFT: 0x4000881C)

描述位	名称	读写	描述	复位值
15: 0	BFT[15: 0]	R/W	LIN 间隔场检测时间寄存器，8bit通讯时间 < 时间 < 13bit通讯时间。即(8*LIN模块时钟)/波特率 < BFT < (13*LIN模块时钟)/波特率	0xFFFF

### 19.3.7. UART 通讯数据寄存器(UARTBUF: 0x40008804)

描述位	名称	读写	描述	复位值
7: 0	UARTBUF [7: 0]	R/W	8位数据接收/发送缓存器。	0x00

### 19.3.8. UART 波特率寄存器(BAUDCFG: 0x40008808)

描述位	名称	读写	描述	复位值
9: 0	BAUD[9: 0]	R/W	10位波特率。	0x0FF
12: 10	N[2: 0]	R/W	小数分频波特率分子位。	0x07
15: 13	M[2: 0]	R/W	小数分频波特率分母位。	0x07

### 19.3.9. UART 发送缓存寄存器 1(WBUF1: 0x40008820)

描述位	名称	读写	描述	复位值
7: 0	SDAT0[7: 0]	R/W	发送缓存寄存器，写入要发送的数据。	0x00
15: 8	SDAT1[15: 8]	R/W		0x00
23: 16	SDAT2[23: 16]	R/W		0x00

描述位	名称	读写	描述	复位值
31: 24	SDAT3[31: 24]	R/W		0x00

### 19.3.10. UART 发送缓存寄存器 2(WBUF2: 0x40008824)

描述位	名称	读写	描述	复位值
7: 0	SDAT4[7: 0]	R/W	发送缓存寄存器，写入要发送的数据。	0x00
15: 8	SDAT5[15: 8]	R/W		0x00
23: 16	SDAT6[23: 16]	R/W		0x00
31: 24	SDAT7[31: 24]	R/W		0x00

### 19.3.11. UART 发送缓存寄存器 3(WBUF3: 0x40008828)

描述位	名称	读写	描述	复位值
7: 0	SDAT8[7: 0]	R/W	发送缓存寄存器，写入要发送的数据。	0x00

发送时，UART 依次从 SDAT0、SDAT1、SDAT2、SDAT3、SDAT4、SDAT5、SDAT6、SDAT7、SDAT8 取出要发送数据。

### 19.3.12. UART 接收缓存寄存器 1(RBUF1: 0x40008830)

描述位	名称	读写	描述	复位值
7: 0	RDAT0[7: 0]	R	接收缓存寄存器。	0x00
15: 8	RDAT1[15: 8]	R		0x00
23: 16	RDAT2[23: 16]	R		0x00
31: 24	RDAT3[31: 24]	R		0x00

### 19.3.13. UART 接收缓存寄存器 2(RBUF2: 0x40008834)

描述位	名称	读写	描述	复位值
7: 0	RDAT4[7: 0]	R	接收缓存寄存器。	0x00
15: 8	RDAT5[15: 8]	R		0x00
23: 16	RDAT6[23: 16]	R		0x00
31: 24	RDAT7[31: 24]	R		0x00

### 19.3.14. UART 接收缓存寄存器 3(RBUF3: 0x40008838)

描述位	名称	读写	描述	复位值
7:0	RDAT8[7:0]	R	接收缓存寄存器。	0x00

接收时, UART 依次向 RDAT0、RDAT1、RDAT2、RDAT3、RDAT4、RDAT5、RDAT6、RDAT7、RDAT8 写入接收到的数据。

## 19.4.功能描述

发送和接收的数据缓存寄存器在物理上是相互独立的, 一个用于发送数据只能写入不能读取、一个用于接收数据只能读取不能写入, 两个缓存寄存器共用 UARTBUF 寄存器, 通过读写指令区别。

发送控制器的作用是波特率时钟的配合下, 将发送数据缓冲器中的并行数据转换为串行数据, 并自动添加起始位、可编程位和停止位。这一过程结束后可使发送中断请求标志位 TI 自动置 1, 如果中断使能, 将会触发 Uart 中断。

接收控制器的作用是波特率时钟的配合下, 使来自 RXD 的串行数据转为并行数据, 并自动过滤掉起始位、可编程位和停止位。这一过程结束后可使接收中断请求标志位 RI 自动置 1, 如果中断使能, 将会触发 Uart 中断。

## 19.5.工作方式

### 19.5.1. 工作方式 0—同步移位寄存器

当 SM[0:1] 设置成 00 时, Uart 工作在同步移位寄存器工作模式, 固定波特率是 PCLK/12, TXD 输出同步时钟, RXD 发送或接收数据。

将要发送的数据写到 UARTBUF 寄存器, 会产生一个启动发送传输脉冲, 发送控制器会把 UARTBUF 中的 8 位数据以 PCLK/12 的固定波特率从 RXD 引脚串行输出, 低位在先, TXD 引脚同时输出移位脉冲, 当发送完 8 位数据后 TI 标志位会置 1。如果中断使能, 将会触发 Uart 中断。

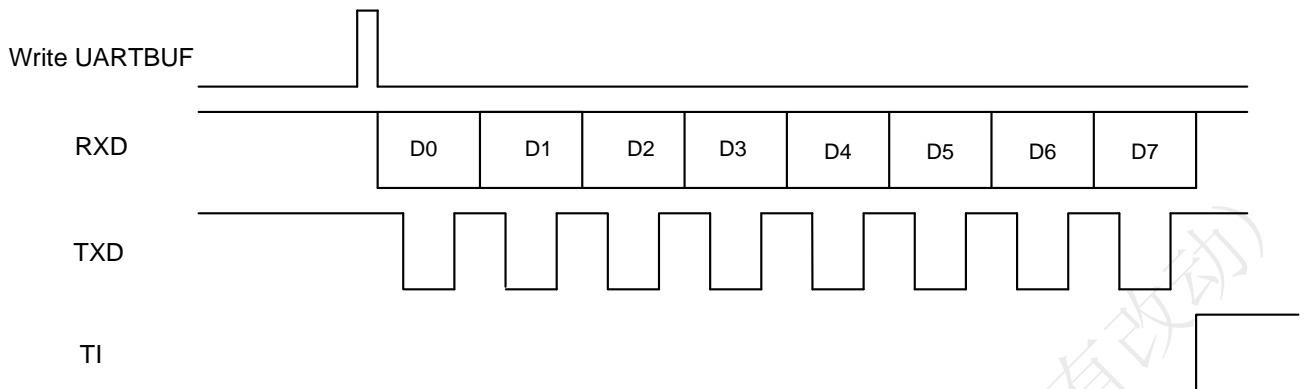


图 19-5-1-1 方式 0 发送时序

REN=1 时，允许接收数据；REN=0 禁止接收数据。当允许接收数据时将 RI 标志位写 0 时会产生一个启动接收传输脉冲，接收控制器以 PCLK/12 的固定波特率从 RXD 引脚串行读取数据，低位在先，TXD 引脚同时输出移位脉冲，当接收完 8 位数据后 RI 标志位会置 1。如果中断使能，将会触发 Uart 中断。

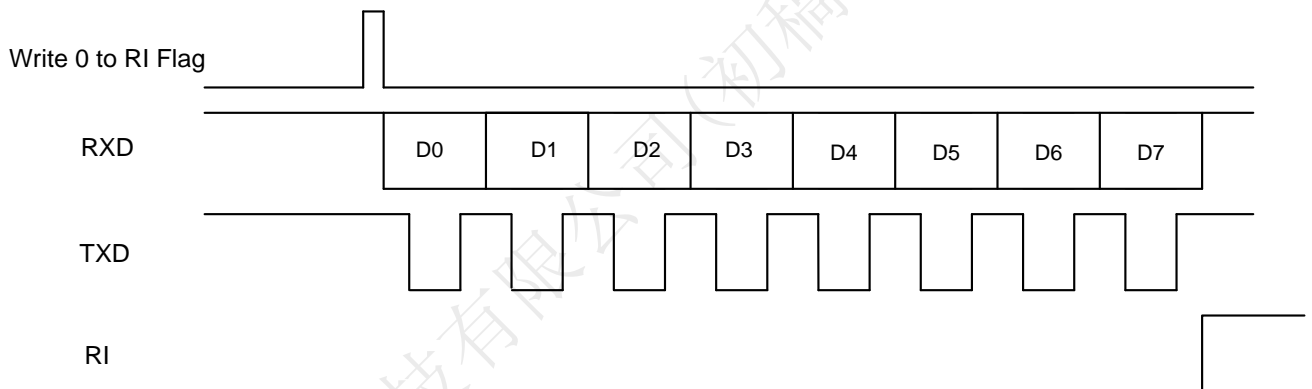


图 19-5-1-2 方式 0 接收时序

## 19.5.2. 工作方式 1—10 位数据异步通信

当 SM[0: 1]设置成 01 时，Uart 工作在 10 位数据异步通信工作模式。通信时钟波特率是可调的，波特率配置见“波特率设置”章节。

该工作模式下发送数据由 TXD 输出，接收数据由 RXD 输入。数据帧由 10 位组成，包括 1 位起始位，8 位数据位，1 位停止位。

将要发送的数据写到 UARTBUF 寄存器，会产生一个启动发送传输脉冲，发送控制器会把 UARTBUF 中的 8 位数据以设定的波特率从 TXD 引脚串行输出，低位在先。先发送 1 位起始位，然后是 8 位数据，最后是 1 位停止位。当发送完停止位后 TI 标志位会置 1。如果中断使能，将会触发 Uart 中断。

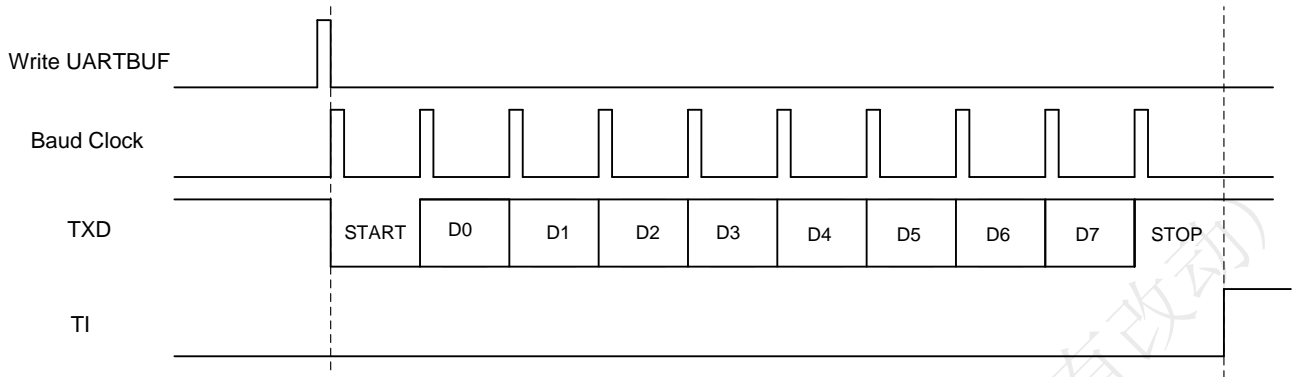


图 19-5-2-1 方式 1 发送时序

REN=1 时，允许接收数据；REN=0 禁止接收数据。当允许接收数据时，采样时钟会不断采样 RXD 引脚的信号，当检测到低电平时，且低电平宽度满足设定的波特率要求，认为检测到起始位，然后开始接收 8 位数据，最后接收完高电平的停止位后 RI 标志位会置 1。如果中断使能，将会触发 Uart 中断。

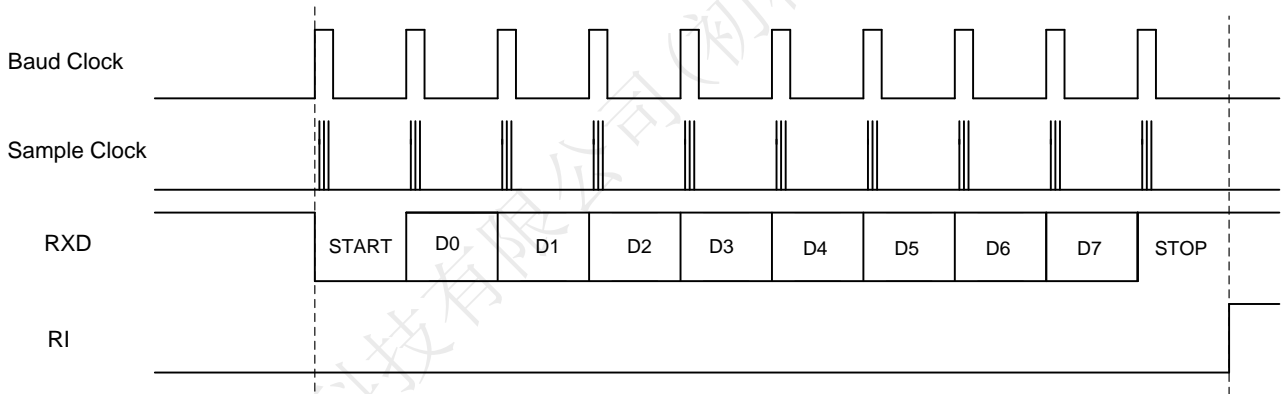


图 19-5-2-2 方式 1 接收时序

### 19.5.3. 工作方式 2—11 位数据异步通信

当 SM[0: 1]设置成 10 时，Uart 工作在 11 位数据异步通信工作模式。通信时钟波特率固定为 PCLK/32。

该工作模式下发送数据由 TXD 输出，接收数据由 RXD 输入。数据帧由 11 位组成，包括 1 位起始位，9 位数据位，1 位停止位。其中数据位的第 9 位发送时由 TB8 设定，接收时保存在 RB8 寄存器位。

将要发送的数据写到 UARTBUF 寄存器，会产生一个启动发送传输脉冲，发送控制器会把 UARTBUF 中的 8 位数据以设定的波特率从 TXD 引脚串行输出，低位在先。先发送 1 位起始位，接着是 8 位数据，然后是 TB8 位，最后是 1 位停止位。当发送完停止位后 TI 标志位会置 1。如果中断使能，将会触发 Uart 中断。

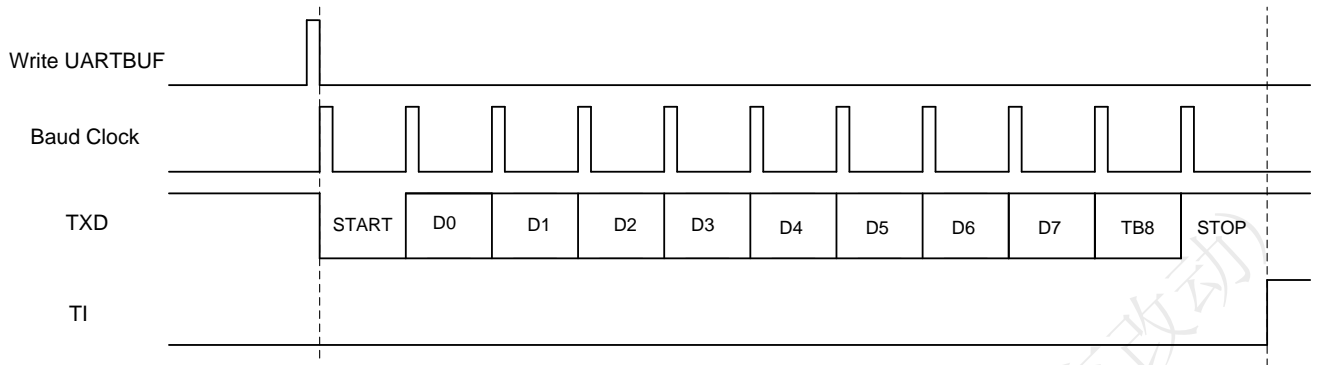


图 19-5-3-1 方式 2 发送时序

REN=1 时, 允许接收数据; REN=0 禁止接收数据。当允许接收数据时, 采样时钟会不断采样 RXD 引脚的信号, 当检测到低电平时, 且低电平宽度满足设定的波特率要求, 认为检测到起始位, 接着接收 8 位数据, 然后接收 1 位数据保存在 RB8 寄存器位, 最后接收完高电平的停止位后 RI 标志位会置 1。如果中断使能, 将会触发 Uart 中断。

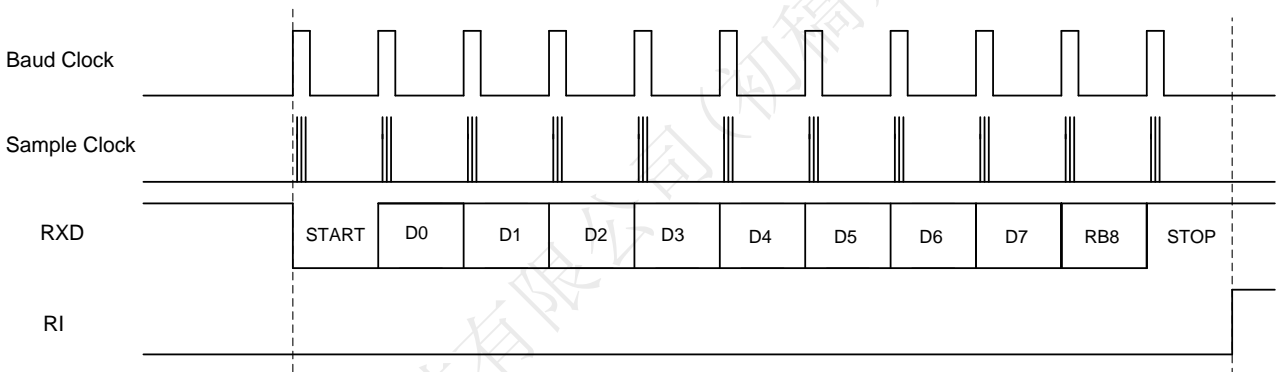


图 19-5-3-2 方式 2 接收时序

### 工作方式 3—11 位数据异步通信

方式 3 与方式 2 差别, 除了波特率可调外, 其他控制方式与方式 2 一样。波特率配置见“波特率设置”章节。

#### 19.5.4. 多机通信

当 SM2 置 1, 且工作模式为方式 2 或方式 3 时, Uart 工作在多机通信模式。只有当接收到的第 9 位数据(RB8)为 1 时, 才置位 RI, 产生中断请求; 当接收到的数(RB8)为 0 时, 将接收到的前 8 位数据丢弃。

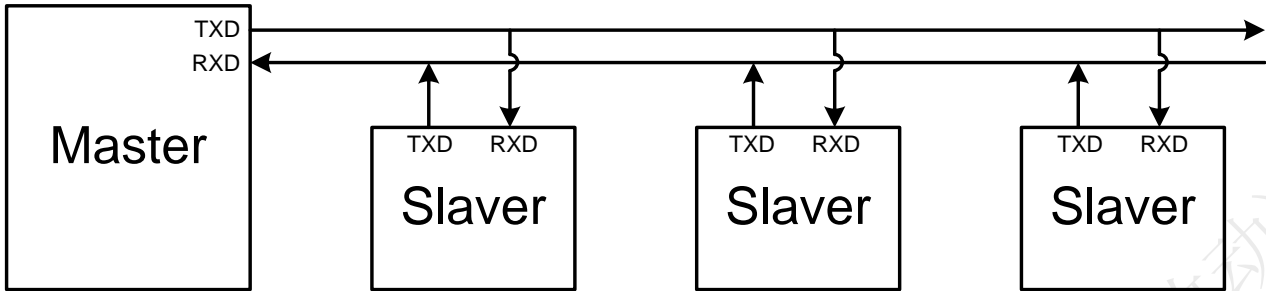


图 19-5-4 多机通信

多机通信过程:

1. 所有从机 SM2 设置为 1，只接收 RB8 为 1 的数据帧，TXD 脚配置为开漏输出。
2. 主机发送地址数据，且 TB8 为 1。所有从机都接收到了地址数据，并进行判断地址与从机本身地址是否匹配。如果地址匹配，则从机将 SM2 设置为 0，否则保持 SM2 为 1。
3. 从机地址匹配后，回复数据给主机确认有从机地址匹配上了。
4. 主机与地址匹配的从机进行数据通信，通信过程中主机发送数据的 TB8 需要都为 0，这样 SM2 为 1 的从机不会接收主机的数据。
5. 通信结束后，从机将 SM2 设置为 1。重新开始下一次通信。

### 19.5.5. 波特率设置

波特率计算公式如下所示: 
$$\left( \frac{F_{osc}}{16 \times (1024 - BAUD) \times (1 + N/M)} \right)$$
, 在配置时必须确保 N 小于等于

于 M。

以系统外设时钟是 16MHz 为例，典型波特率的配置如下所示:

目标波特率	N	M	BAUD	实际波特率	误差
256000	2	7	1021	259259.3	1.27%
230400	3	7	1021	233333.3	1.27%
128000	4	7	1019	127272.7	-0.56%
115200	3	7	1018	116666.7	1.27%
57600	3	7	1012	58333.3	1.27%
56000	1	5	1009	55555.6	-0.8%
38400	1	4	1003	38095.2	-0.8%
19200	1	4	982	19047.6	0.02%
14400	2	7	970	14403.3	0.02%
9600	2	7	943	9602.2	0.02%
4800	2	7	862	4801.1	0.02%

表 19-5-5 典型波特率设置

### 19.5.6. LIN 同步帧检测

LIN 报文帧包括帧头(header)与应答(response)两部分。主机负责发送至帧头;从机负责接收帧头并作出解析, 然后决定是发送应答, 还是接收应答或不回复。

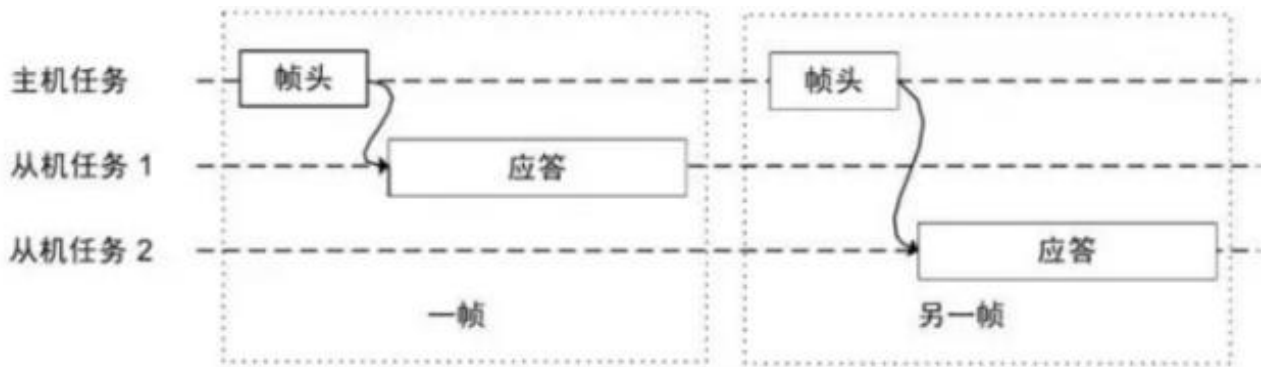


图 19-5-6

帧头结构包括同步间隔段、同步段、PID 段(受保护 ID)段, 应答部分包括数据段与校验和段。

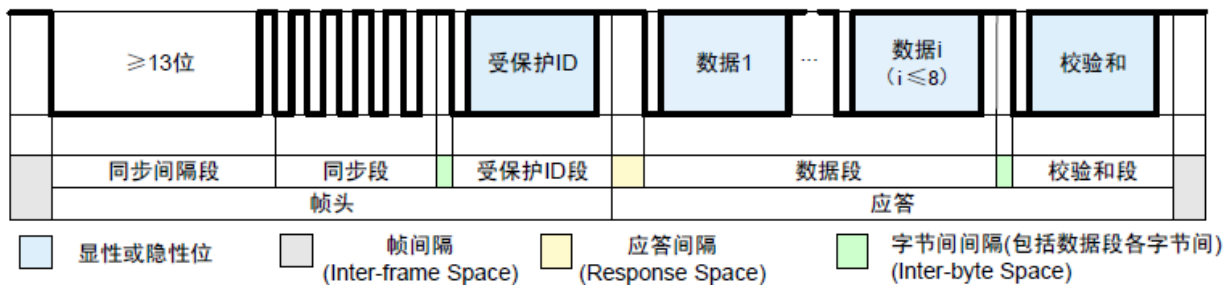


图 19-5-6-1

- LIN 间隔段时间寄存器(LINBFT: 0x4000881C)

同步间隔段同步间隔段至少是由 13 位的显性电平组成, 由于帧中的所有间隙或者总线空闲时总线均保持隐性电平状态。所以同步间隔段可以标志一个帧的开始。其中同步间隔段的间隔符至少为 1 位隐性电平。

硬件检测低电平持续时间, 如果持续时间大于 LINBFT 寄存器设定阈值, 则判定为有效的同步间隔段。

- LIN 波特率寄存器(LINBAUD: 0x40008818)

同步段 LIN 同步以下降沿为判断标志, 采用字节 0x55(01010101b)进行同步。在从机节点上可以不采用高精度的时钟, 由此带来的偏差, 需要通过同步段来进行调整。

下降沿后检测 5 个上升沿，计算 5 个上升沿的间隔时间，如果间隔时间小于 LINBFT 寄存器设定阈值，则除以 8 写入到 LINBAUD 寄存器。需要 MCU 根据 LINBAUD 寄存器去配置 UART 波特率寄存器(BAUDCFG: 0x40008808)。

以系统外设时钟是 16MHz 为例，典型 LIN 通讯波特率的配置如下所示：

目标波特率	LINBAUD	N	M	BAUD	实际波特率	误差
38400	417	1	4	1003	38095.2	-0.8%
19200	833	1	4	982	19047.6	0.02%
14400	1111	2	7	970	14403.3	0.02%
9600	1667	2	7	943	9602.2	0.02%
4800	3333	2	7	862	4801.1	0.02%

表 19-5-6 LIN 波特率设置

● PID 寄存器(LINPID: 0x40008814)

PID 段受保护的 ID 的前 6 位叫做帧 ID，加上两个奇偶校验位后称作受保护的 ID。帧 ID 的取值范围为 0x00~0x3F 总共 64 个，帧 ID 标识了帧的类别和目的地。从机任务会根据帧头 ID 作出反应(接收/发送/忽略应答)。

其中校验位如下：

$$P0 = ID0 \oplus ID1 \oplus ID2 \oplus ID4$$

$$P1 = \neg (ID1 \oplus ID3 \oplus ID4 \oplus ID5)$$

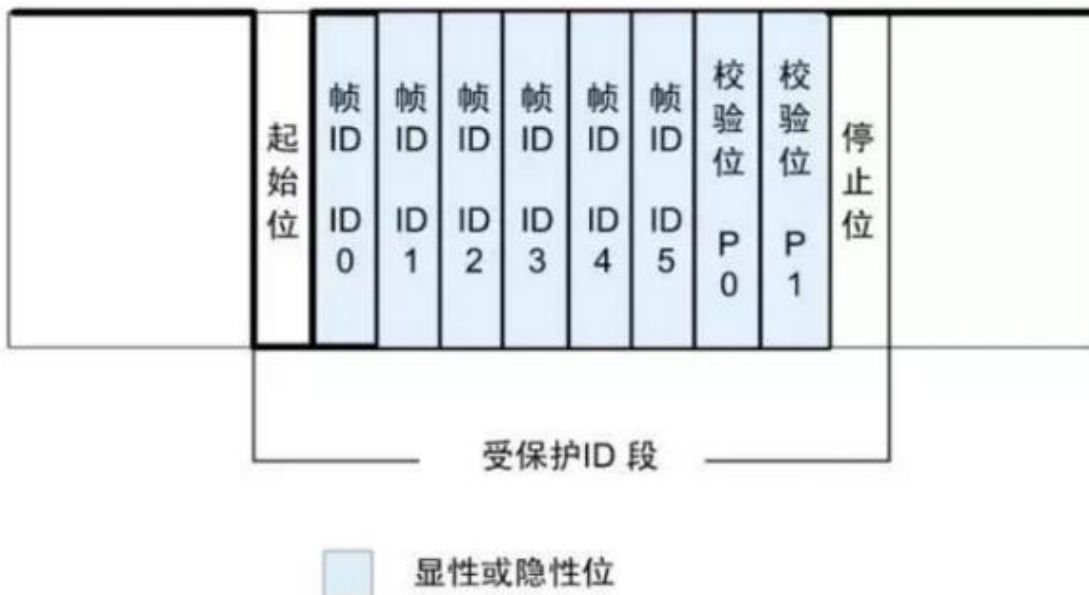


图 19-5-6-2

在 CPU 待机时，硬件自动检测 LIN 总线的帧头(同步间隔段、PID)，检测到有效的帧头后，可以产生中断唤醒 MCU，MCU 根据 PID 做出应答。

## 20. I2C 串行通信收发器(I2C)

I2C 模块同时有 8 Byte 的发送、接收缓存器，可以一次多个数据的传输，不需要 CPU 的频繁参与，节省 CPU 指令。支持 master、slave 模式，支持标准模式、快速模式。

### 20.1. 模块寄存器总表

寄存器名	地址	结构体(.h)	寄存器描述	复位值
I2CBAUD	0x40008000	I2C	波特率寄存器	0x0000
I2CSADR	0x40008004	I2C	Slave地址寄存器	0x00
I2CCON	0x40008008	I2C	控制寄存器	0x0000
I2CINTMASK	0x4000800C	I2C	中断屏蔽寄存器	0x0000
I2CINTCLR	0x40008010	I2C	中断标志清除寄存器	0x0000
I2CTRNUM	0x40008014	I2C	多字节传输控制寄存器	0x01
I2CSTA	0x40008018	I2C	传输状态寄存器	0x00000000
I2CSDAT3_0	0x40008020	I2C	发送数据寄存器	0x00000000
I2CSDAT7_4	0x40008024	I2C	发送数据寄存器	0x00000000
I2CRDAT3_0	0x40008030	I2C	接收数据寄存器	0x00000000
I2CRDAT7_4	0x40008034	I2C	接收数据寄存器	0x00000000

### 20.2. 寄存器说明

#### 20.2.1. 波特率控制寄存器(I2CBAUD: 0x40008000)

描述位	名称	读写	描述	复位值
15: 0	I2CBAUD[15: 0]	R/W	波特率设置寄存器。	0x0000

SCL 的频率由 I2CBAUD[15: 0]控制，占空比由 SCLDUTY 控制。

- SCLDUTY=0

$$I2CBAUD = (PCLK / SCL) / 2$$

SCL 的高低电平时间控制如下:

SCL\_HI\_CNT = I2CBAUD - 1, 考虑 SCL 上升快慢影响，高电平时间会增加 3 个左右 PCLK 周期

$$SCL_LO_CNT = I2CBAUD + 3$$

- SCLDUTY=1

理论上:  $I2CBAUD = (PCLK / SCL) / 3$

SCL 的高低电平时间控制如下:

SCL\_HI\_CNT = I2CBAUD + 2, 考虑 SCL 上升快慢影响, 高电平时间会增加 3 个左右 PCLK 周期

SCL\_LO\_CNT = I2CBAUD\*2

### 20.2.2. 从机地址寄存器(I2CSADR: 0x40008004)

描述位	名称	读写	描述	复位值
6: 0	I2CSADR[6: 0]	R/W	Slave地址寄存器, 作为Slave器件时, 用作器件地。(备注: 从机接收的第一个字节, 高七位和从机地址寄存器进行匹配, 低0位是读写标志位)	0x00
7			保留	

### 20.2.3. 控制寄存器(I2CCON: 0x40008008)

描述位	名称	读写	描述	复位值
0	RST	R/W	总线重新启动控制位。 0: disable; 1: enable。 如果不想停止总线, 但需要重新启动时, 可以把这个命令设置为1, 当此命令有效后, 会由硬件来清零。	0
1	TAK	R/W	Ack应答控制位。 0: ack; 1: no ack。	0
2	MT	R/W	主机发送/接收控制位。 0: 接收; 1: 发送。	0
3	MSTA	R/W	传输控制位。 0->1: 主机开始启动传输; 1->0: 主机停止传输。 当仲裁失败后, 此寄存器被清零, 表示master没有总线使用权。	0
4	MIEN	R/W	中断使能控制位。 0: 关闭中断; 1: 使能中断。	0
5	MEN	R/W	模块使能控制位。	0

描述位	名称	读写	描述	复位值
			0: I <sup>2</sup> C模块关闭; 1: I <sup>2</sup> C模块使能。 在开始启动I <sup>2</sup> C模块前, 应该置位此位。	
6	MBEN	R/W	多字节传输控制位。 0: 禁止多字节传输, 只能单字节传输; 1: 多字节传输使能。	0
7	GCD	R/W	广播地址开启。 0: 可以产生广播地址; 1: 不能产生广播地址。	0
8	SCLDUTY	R/W	SCL占空比控制位。 0: low/high=1; 1: low/high=2。	0
12: 9	DHDT[3: 0]	R/W	总线时序控制	0000
13	MODE	R/W	模式控制位。 0: 快速模式(fast mode); 1: 标准模式(standard mode)。	0
14	SCLIN	R	监视总线SCL数据。	
15	SDAIN	R	监视总线SDA数据。	

I<sup>2</sup>C 总线时序控制由 DHDT[3: 0]控制, 在快速模式(fast mode)、标准模式(standard mode), 时序的控制也不一样:

- I<sup>2</sup>CBAUD[15: 0] 要保证大于 DHDT[3: 0] \* 2
- SDA Hold time(SDA 相对于 SCL 下降沿的延迟时间, 下图 T1)  
根据 DHDT[3: 0]设置, 硬件会将 sda 保持时间控制为大于或等于 5。
  1. i2c\_mode = 0 (Fast mode) : DHDT[3: 0] \* 2
  2. i2c\_mode = 1 (Standard mode) : DHDT[3: 0] \* 5
- Hold time (repeated) START condition(下图 T3)
  1. i2c\_mode = 0 (Fast mode) : DHDT[3: 0] \* 2
  2. i2c\_mode = 1 (Standard mode) : DHDT[3: 0] \* 14
- Set-up time for repeated START condition(下图 T2)
  1. i2c\_mode = 0 (Fast mode) : DHDT[3: 0] \* 2
  2. i2c\_mode = 1 (Standard mode) : DHDT[3: 0] \* 16
- Set-up time for STOP condition(下图 T4)
  1. i2c\_mode = 0 (Fast mode) : DHDT[3: 0] \* 2
  2. i2c\_mode = 1 (Standard mode) : DHDT[3: 0] \* 14
- bus free time between a STOP and START condition
  1. i2c\_mode = 0 (Fast mode) : DHDT[3: 0] \* 5
  2. i2c\_mode = 1 (Standard mode) : DHDT[3: 0] \* 16
- Rise time both SDA and SCL signals(SDA 输出低电平时, 检测 SCL 跳变的 Blanking 时间)

- 1. i2c\_mode = 0 (Fast mode) : DHDT[3: 0]
- 2. i2c\_mode = 1 (Standard mode) : DHDT[3: 0] \* 4
- Fall time both SDA and SCL signals(SDA 输出高阻时, 检测 SCL 跳变的 Blanking 时间)
  - 1. i2c\_mode = 0 (Fast mode) : DHDT[3: 0]
  - 2. i2c\_mode = 1 (Standard mode) : DHDT[3: 0]

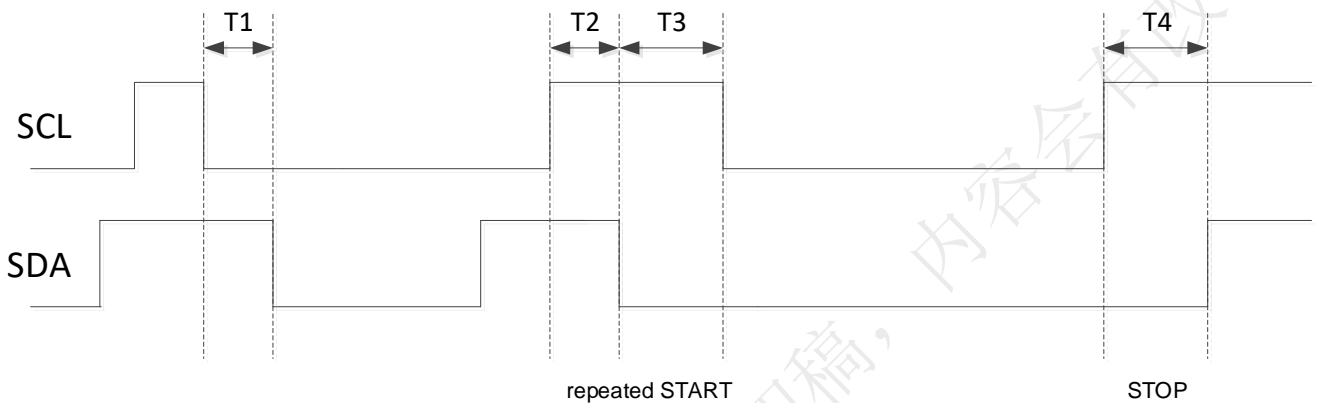


图 20-2-3

#### 20.2.4. 中断屏蔽寄存器(I2CINTMASK: 0x4000800C)

描述位	名称	读写	描述	复位值
15: 0	INTMASK[15: 0]	R/W	中断屏蔽寄存器低字节。 0: 屏蔽对应中断, 不能向MCU发出中断请求; 1: 不屏蔽中断。	0x0000

INTMASK 和中断状态位的对应关系见“传输状态寄存器”。

#### 20.2.5. 中断标志清除寄存器(I2CINTCLR: 0x40008010)

描述位	名称	读写	描述	复位值
15: 0	INTCLR [15: 0]	R/W	当某一位设置为1时, 中断状态中的相应标志位就会被清零, 以清除中断。 该寄存器只保持2个pclk时钟周期, 硬件会自动清0。	0x0000

INTCLR 和中断状态的对应关系见“传输状态寄存器”。

### 20.2.6. 多字节传输控制寄存器(I2CTNUM: 0x40008014)

描述位	名称	读写	描述	复位值
6: 0	TRNUM[6: 0]	R/W	需要发送接收的字节数，最大为8个字节。(使能MBEN时有效)	0x01
7	MRXLAST	R/W	设置为1，表示master receiver完设定的字节接收后，master给出No Ack，结束接收。(使能MBEN时有效)	0

### 20.2.7. 传输状态寄存器(I2CSTA: 0x40008018)

描述位	名称	读写	描述	复位值
7: 0	TRN[7: 0]	R	已经传输完的字节数目。	0
8	MIF	R	中断标志位， 0: 无中断， 1: 有中断。	0
9	TBUF	R	0: master允许生成START状态， 1: master不允许生成START状态。	0
10	RAK	R	接收到的Ack。	0
11	MBB	R	0: I <sup>2</sup> C总线处于空闲， 1: I <sup>2</sup> C总线处于忙。	0
12	SRW	R	R/W标志位	0
13	MAAS	R	1: 被寻址作为slave，并且地址匹配。	0
14	MCT	R	0: 表示多字节传输还在进行中 或者处于空闲状态， 1: 表示多个字节已经传输完成。	0
15	MCF	R	0: 一个字节还在传输中， 1: 一个字节传输完成。	0
31: 16	INTSTAS[15: 0]	R	I <sup>2</sup> C传输状态寄存器	0

- 当有中断产生时，如果不把中断清除的话，SCL 就处于等待状态，即时钟保持中断前一个的状态不变。所以建议要对中断进行处理，把中断清除。
- 仲裁时，当冲突发生在字节的第八个 bit 时，会有问题出现，就是 SCL 还会有时钟输出，而不是被拉高。
- INTSTAS : I<sup>2</sup>C 传输状态寄存器，在启动一个传输后，在状态寄存器中会有相应位的变化，来指示传输所处于何种状态。在下列传输状态，并且 INTMASK[15: 0] 相应位为 1 ，中断状态位会被置 1；当 Int\_clear[15: 0] 相应位为 1 时 ，中断状态位会被清 0。

INTSTAS[0]=1: 在 master 模式下，仲裁失败。

INTSTAS[1]=1: 进入 slave transmitter 状态。

INTSTAS[2]=1: 进入 slave receive 状态。

INTSTAS[3]=1: 接收到 No Ack。

INTSTAS[4]=1: master transmitter, 设定的字节数目传输完成。

INTSTAS[5]=1: master receive, 设定的字节数目接收完成

INTSTAS[6]=1: slave transmitter, 设定的字节数目已经传输完成

INTSTAS[7]=1: slave receive, 在单字节设置时, 完成接收一个字节;

在多字节设置时, 完成接收 8 个字节, 或者 I<sup>2</sup>C 停止,

或者 repeat start。

INTSTAS[8]=1: slave 模式下, 探测到 repeat start。

INTSTAS[9]=1: slave 模式下, 探测到 stop。

INTSTAS[10]=1: slave receiver, 广播地址。

INTSTAS[11: 15]=00, 这些状态位保留。

### 20.2.8. 发送数据寄存器(I2CSDAT3\_0: 0x40008020)

描述位	名称	读写	描述	复位值
7: 0	SDAT0[7: 0]	R/W	发送数据寄存器0	0x00
15: 8	SDAT1[7: 0]	R/W	发送数据寄存器1	0x00
23: 16	SDAT2[7: 0]	R/W	发送数据寄存器2	0x00
31: 24	SDAT3[7: 0]	R/W	发送数据寄存器3	0x00

### 20.2.9. 发送数据寄存器(I2CSDAT7\_4: 0x40008024)

描述位	名称	读写	描述	复位值
7: 0	SDAT4[7: 0]	R/W	发送数据寄存器4	0x00
15: 8	SDAT5[7: 0]	R/W	发送数据寄存器5	0x00
23: 16	SDAT6[7: 0]	R/W	发送数据寄存器6	0x00
31: 24	SDAT7[7: 0]	R/W	发送数据寄存器7	0x00

### 20.2.10. 接收数据寄存器(I2CRDAT3\_0: 0x40008030)

描述位	名称	读写	描述	复位值
7: 0	I2CRDAT0[7: 0]	R/W	接收数据寄存器0	0x00
15: 8	I2CRDAT1[7: 0]	R/W	接收数据寄存器1	0x00
23: 16	I2CRDAT2[7: 0]	R/W	接收数据寄存器2	0x00
31: 24	I2CRDAT3[7: 0]	R/W	接收数据寄存器3	0x00

### 20.2.11. 接收数据寄存器(I2CRDAT7\_4: 0x40008034)

描述位	名称	读写	描述	复位值
7: 0	I2CRDAT4[7: 0]	R/W	接收数据寄存器4	0x00
15: 8	I2CRDAT5[7: 0]	R/W	接收数据寄存器5	0x00
23: 16	I2CRDAT6[7: 0]	R/W	接收数据寄存器6	0x00
31: 24	I2CRDAT7[7: 0]	R/W	接收数据寄存器7	0x00

## 20.3.功能说明

### 20.3.1. 单字节收发设置:

可以只设置控制寄存器的  $MBEN=0$ ，禁止多字节传送；也可以只设置  $I2CTRNUM==0x01$ ；或者两个同时设置为单字节。

### 20.3.2. 多字节收发设置:

必须先使能控制寄存器的多字节使能位  $MBEN=1$ ，然后设置要传输  $I2CTRNUM$  多少字节，否则设置的无效。

作为 **slave** 或者 **master** 时，内部都有 8 个字节的数据寄存器。但是作为 **master** 的时候，**master** 需要发送器件地址给 **slave**，所以需要一字节数据来存储 7bits 地址和一个读写标志位，这样就剩下 7 字节作为有效的数据字节，当发送和接收一字节数据时，需要设置要  $I2CTRNUM==0x02$ (传输结束后，不给出 **ack**)。作为 **slave** 的时候，除接收器件地址外，还最多可以接收和发送 8byte 有效数据。如果时钟线被 **slave** 拉低，**master** 会停止传输，并保持等待状态。直到时钟线变高才继续传输。

1. 可以通过中断或者查询状态寄存器来判断目前传输处于何种状态，然后作出相应的处理。
2. 如果不想用中断方式，而用查询方式时，可以设置  $I2CINTMASK==0x0000$ ，否则把相应需要中断的位设置为 1。
3. 仲裁:

在启动  $I^2C$  之前，最好能够先使能  $I^2C$ ，这样如果可以检测到总线处于忙状态，可以不申请总线，等总线空闲时再申请；如果需要申请总线，可以启动  $I^2C$ ，之后通过仲裁来决定。

当仲裁失败后，如果是 **sda** 上的数据发送的与接收的不同，就会停止传输并产生中断，还需要重新启动才能继续申请总线；如果是其他原因，会有中断产生，但是有可能传输不会停止。

### 20.3.3. 中断

当中断产生时，如果是 **slave** 模式，**slave** 会把时钟线拉低，来延迟 **master** 的操作，直到 **slave** 已经通过中断服务程序对中断进行处理并清除中断源；如果在 **master** 模式下，时钟保持中断前一个的状态不变，直到中断清除为止。

当中断产生时，如果中断使能控制位 **I2CCON[4](MIEN)**有效的的话，中断请求信号就会有效，可以通过此信号来通知处理器进行中断处理，否则无效。

### 20.3.4. IIC 总线时序图

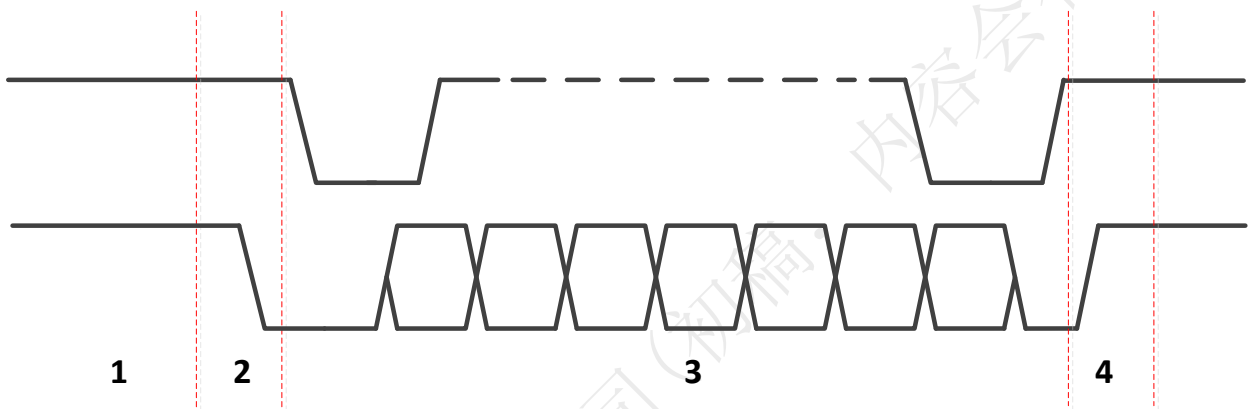


图 20-3-4

上图中#1 表示“总线空闲状态”，在此状态下时钟信号 **SCL** 和数据信号 **SDA** 均保持高电平，此时无 **IIC** 设备工作。

上图中#2 表示“起始信号”，在 **IIC** 总线处于“空闲状态”时，**SCL** 依旧保持高电平时，**SDA** 出现由高电平转为低电平的下降沿，产生一个起始信号，此时与总线相连的所有 **IIC** 设备在检测到起始信号后，均跳出空闲状态，等待控制字节的输入。

上图中#3 表示“数据读/写状态”，在串行时钟线 **SCL** 为低电平状态时，**SDA** 允许改变传输的数据位(1 为高电平，0 为低电平)，在 **SCL** 为高电平状态时，**SDA** 要求保持稳定，相当于一个时钟周期传输 **1bit** 数据，经过 8 个时钟周期后，传输了 **8bit** 数据，即一个字节。第 8 个时钟周期末，主机释放 **SDA** 以使从机应答，在第 9 个时钟周期，从机将 **SDA** 拉低以应答；如果第 9 个时钟周期，**SCL** 为高电平时，**SDA** 未被检测到为低电平，视为非应答，表明此次数据传输失败。需要注意数据以 **8bit** 即一个字节为单位串行发出，其最先发送的是字节的最高位。

上图中#4 表示“停止信号”，完成数据读写后，时钟 **SCL** 保持高电平，当数据 **SDA** 产生一个由低电平到高电平的上升沿时，产生一个停止信号，**IIC** 总线跳转回“总线空闲状态”。

### 20.3.5. 寻址方式

作为从机时，接收的第一个字节的高 7 位为主机发送的从机地址，高 7 位(bit[7: 1])定义从机的地址。第 0 位为读写标志位。

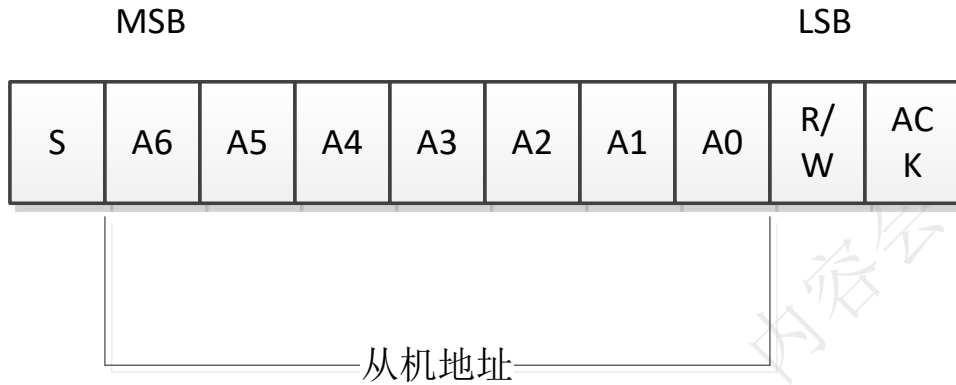


图 20-3-5

### 20.3.6. 开始

作为 master 时，控制寄存器 Msta 由 0 到 1，之后保持为 1。SDA 出现由高电平转为低电平的下降沿，产生一个起始信号。

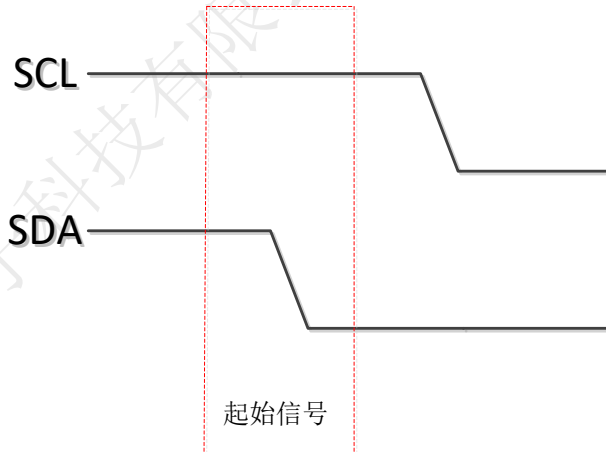


图 20-3-6

### 20.3.7. 停止

作为 master 时，控制寄存器 Msta 由 1 到 0，之后保持为 0。时钟 SCL 保持高电平，当数据 SDA 产生一个由低电平到高电平的上升沿时，产生一个停止信号。

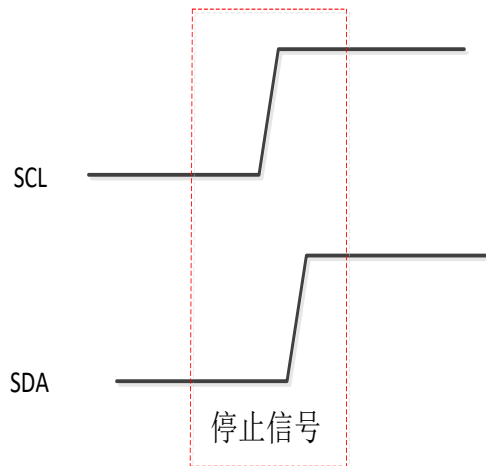


图 20-3-7

### 20.3.8. 重启

作为 **master** 时，把 **RST** 设置为 1 即可，不需要用软件来把该位再设置为 0，因为硬件在

检测到 **RST** 后，会把该位清零。

当然，也先让 **Msta** 有个下降沿，停止总线，然后再让 **Msta** 有个上升沿，之后保持高电平，由此开启总线传输。

注意: I<sup>2</sup>C 协议规定: 如果总线在忙，不允许其他 **master** 请求，否则可能出错；如果总线空闲，其他的 **master** 都可以申请总线使用权。

## 20.4. 示例代码

### 20.4.1. I<sup>2</sup>C 管脚配置示例

```

/***** I2C 复用功能 PB6 SCL PB7 SDA *****/
IOALT->REMAP1_b.I2CALT = 0;
IOALT->PBFMUX70_b.PB6ALT = 2;
IOALT->PBFMUX70_b.PB7ALT = 2;
PB->ALT_b.bit6 = 1;
PB->ALT_b.bit7 = 1;
PB->MD_b.bit6 = 0;
PB->MD_b.bit7 = 0;
PB->OEN_b.bit6 = 1;
PBCFG->PDRS_b.bit6 = 1;
PBCFG->PDRS_b.bit7 = 1;
    
```

```
PBCFG->ODOSS_b.bit6 = 1;
PBCFG->ODOSS_b.bit7 = 1;
```

### 20.4.2. 示例程序：主机模式，单字节发送

```
SYSPEN->PERIEN_b.I2C_EN = 1;
I2C->I2CBAUD = 4000;
I2C->I2CSADR = 0xA0;    //从机地址
I2C->I2CINTMASK = 0xFFFF;
I2C->I2CSDAT3_0_b.SDAT0 = 0xA0;
I2C->I2CSDAT3_0_b.SDAT1 = 0xCD;
I2C->I2CTRNUM = 0x01;    //单字节
I2C->I2CCON = 0x34;
I2C->I2CCON = 0x3C;
while(1){
    if(I2C->I2CSTA_b.INTSTAS & 0x10) //发送
    {
        num++;
        I2C->I2CSTA_b.MIF = 0;    //中断标志位，1: 有中断，0: 无中断。
        I2C->I2CINTCLR=0x10;
        if(num == 2)
        {
            num = 0;
            I2C->I2CCON_b.MSTA = 0;
        }
    }
}
```

### 20.4.3. 示例程序：主机模式，多字节发送

```
SYSPEN->PERIEN_b.I2C_EN = 1;
I2C->I2CBAUD = 4000;
I2C->I2CSADR = 0xA0;    //从机地址
I2C->I2CINTMASK = 0xFFFF;
I2C->I2CSDAT3_0_b.SDAT0 = 0xA0;
I2C->I2CSDAT3_0_b.SDAT1 = 0xA1;
I2C->I2CSDAT3_0_b.SDAT2 = 0x11;
I2C->I2CSDAT3_0_b.SDAT3 = 0x22;
I2C->I2CSDAT7_4_b.SDAT4 = 0x33;
I2C->I2CCON = 0x74;
I2C->I2CCON = 0x7C;
I2C->I2CTRNUM = 0x05;
```

```

while(1){
    if(I2C->I2CSTA_b.INTSTAS & 0x10)
    {
        I2C->I2CSTA_b.MIF = 0;    //中断标志位，1: 有中断，0: 无中断。
        I2C->I2CINTCLR = 0x10;
        I2C->I2CCON_b.MSTA = 0;
    }
}
    
```

#### 20.4.4. 示例程序：主机模式，单字节接收

```

num = 0;
SYSPEN->PERIEN_b.I2C_EN = 1;
I2C->I2CBAUD = 4000;
I2C->I2CSADR = 0xA1;    //从机地址
I2C->I2CINTMASK = 0xFFFF;
I2C->I2CSDAT3_0_b.SDAT0 = 0xA1;
I2C->I2CCON = 0x30;
I2C->I2CCON = 0x38;
while(1){
    if((I2C->I2CSTA_b.INTSTAS & 0x20) == 0x20)    //设定的字节数目接收完成
    {
        num++;
        I2C->I2CINTCLR=0x20;
        if(num == 2){
            num = 0;
            I2C->I2CCON_b.MSTA = 0;    //不加这句不会自动产生停止位
        }
    }
}
    
```

#### 20.4.5. 示例程序：主机模式，多字节接收

```

SYSPEN->PERIEN_b.I2C_EN = 1;
I2C->I2CBAUD = 4000;
I2C->I2CSADR = 0xA1;    //从机地址
I2C->I2CINTMASK = 0xFFFF;
I2C->I2CSDAT3_0_b.SDAT0 = 0xA1;
I2C->I2CCON = 0x70;
    
```

```

I2C->I2CCON = 0x78;
I2C->I2CTRNUM = 0x87;
while(1){
    if((I2C->I2CSTA_b.INTSTAS & 0x20) == 0x20) // 设定的字节数目接收完成
    {
        I2C->I2CINTCLR = 0x20;
        I2C->I2CCON_b.MSTA = 0;    //写为 0 后会产生停止位，不设置不会产生停止位
    }
}
    
```

#### 20.4.6. 示例程序：主机模式，重启

```

SYSPEN->PERIEN_b.I2C_EN = 1;
I2C->I2CBAUD = 4000;
I2C->I2CINTMASK = 0xFFFF;
I2C->I2CINTCLR = 0xFFFF;
I2C->I2CSDAT3_0_b.SDAT0 = 0xA0;
I2C->I2CSDAT3_0_b.SDAT1 = 0x11;
I2C->I2CSDAT3_0_b.SDAT2 = 0x22;
I2C->I2CCON = 0x0EF4;
I2C->I2CCON = 0x0EFC;
I2C->I2CTRNUM = 0x02;
while(I2C->I2CSTA != 0x00100902){}
I2C->I2CINTCLR = 0x0010;
I2C->I2CSDAT3_0_b.SDAT0 = 0xA1;
I2C->I2CCON = 0x0EF9;
I2C->I2CTRNUM = 0x83;
while(I2C->I2CSTA != 0x00201d03){}
I2C->I2CINTCLR = 0x0020;
I2C->I2CCON = 0x0EB0;
SoftDelay(300);
I2C->I2CCON = 0x0EB8;
I2C->I2CCON_b.MSTA = 1;
while(1);
    
```

### 20.4.7. 示例程序：从机模式，单字节发送

如果没有用 I2CINTMASK 来屏蔽中断时，当中断产生的时候需要对中断进行处理，并需要通过写 Int\_clear 来把中断清除掉，否则 slave 会把把时钟线拉低，用来告诉 master，此时不能多总线上的数据进行处理，让 master 处于等待状态。

#### 1. 单字节单次模式

```

numm = 0;
    SYSPEN->PERIEN_b.I2C_EN = 1;
    I2C->I2CBAUD = 4000;
    I2C->I2CSADR = 0x50;    //从机地址注意此处的地址需要处理，得到真实地址。将
    真实地址右移一位为此处设置
    I2C->I2CINTMASK = 0x200;
    I2C->I2CSDAT3_0_b.SDAT0 = numm;
    I2C->I2CCON = 0x24B0;
    I2C->I2CTRNUM = 0x01;
    while(1){
        if(I2C->I2CSTA_b.INTSTAS & BIT9){ //slave transmitter, 设定的字节数目已经
    发送完成
            I2C->I2CINTCLR = BIT9;
            numm++;
            I2C->I2CSDAT3_0_b.SDAT0 = numm;
        }
    }
    
```

#### 2. 单字节连续模式

```

SYSPEN->PERIEN_b.I2C_EN = 1;
    I2C->I2CINTMASK = 0xFFFF;
    I2C->I2CINTCLR = 0xFFFF;
    I2C->I2CSADR = 0x50;    //从机地址注意此处的地址需要处理，得到真实地址。将
    真实地址右移一位为此处设置
    I2C->I2CTRNUM = 0x01;
    I2C->I2CCON = 0xB6;
    while(I2C->I2CSTA == 0x00023900)
    {
        I2C->I2CINTCLR = 0x0002;
    }
    while(I2C->I2CSTA == 0x00403901)
    {
    
```

```

        I2C->I2CINTCLR = 0x0040;
    }
    while(I2C->I2CSTA == 0x00403901)
    {
        I2C->I2CINTCLR = 0x0040;
    }
    while(I2C->I2CSTA == 0x00483D01)
    {
        I2C->I2CINTCLR = 0x0048;
    }
    while(I2C->I2CSTA == 0x02001100)
    {
        I2C->I2CINTCLR = 0x0200;
    }

```

#### 20.4.8. 示例程序：从机模式，多字节发送

##### 1. 多字节单次模式

```

SYSPEN->PERIEN_b.I2C_EN = 1;
I2C->I2CINTMASK = 0xFFFF;
I2C->I2CINTCLR = 0xFFFF;
I2C->I2CSADR = 0x50;
I2C->I2CTRNUM = 0x03;
I2C->I2CCON = 0x74;
while(I2C->I2CSTA == 0x00023900)
{
    I2C->I2CINTCLR = 0x0002;
}
while(I2C->I2CSTA == 0x00483D03)
{
    I2C->I2CINTCLR = 0x0048;
}
while(I2C->I2CSTA == 0x02001100)
{
    I2C->I2CINTCLR = 0x0200;
}

```

##### 2. 多字节连续模式

### 20.4.9. 示例程序：从机模式，单字节接收

#### 1. 单字节单次模式

```

SYSPEN->PERIEN_b.I2C_EN = 1;
I2C->I2CBAUD = 4000;
I2C->I2CSADR = 0x50; //从机地址注意此处的地址需要处理，得到真实地址。将真实
    地址右移一位为此处设置
I2C->I2CINTMASK = 0x200;
I2C->I2CCON = 0x24B0;
I2C->I2CTRNUM = 0x01; //单字节
I2C_Master = 1;
while(1){
    if(I2C->I2CSTA_b.INTSTAS & BIT9)
    {
        I2C->I2CINTCLR = BIT9;
        RECV2_BUF[BufPtr] = I2C->I2CRDAT3_0_b.RDAT0;
        BufPtr++;
    }
}
    
```

#### 2. 单字节连续模式

```

SYSPEN->PERIEN_b.I2C_EN = 1;
I2C->I2CINTMASK = 0xFFFF;
I2C->I2CINTCLR = 0xFFFF;
I2C->I2CSADR = 0x50; //从机地址注意此处的地址需要处理，得到真实地址。将
    真实地址右移一位为此处设置
I2C->I2CTRNUM = 0x01;
I2C->I2CCON = 0xB0;
while(I2C->I2CSTA == 0x00042900)
{
    I2C->I2CINTCLR = 0x0004;
}
while(I2C->I2CSTA == 0x00802901)
{
    I2C->I2CINTCLR = 0x0080;
}
while(I2C->I2CSTA == 0x00802901)
    
```

```

{
    I2C->I2CINTCLR = 0x0080;
}
while(I2C->I2CSTA == 0x00802D01)
{
    I2C->I2CINTCLR = 0x0080;
}
while(I2C->I2CSTA == 0x02000100)
{
    I2C->I2CINTCLR = 0x0200;
}
    
```

#### 20.4.10. 示例程序：从机模式，多字节接收

##### 1. 多字节单次模式

```

SYSPEN->PERIEN_b.I2C_EN = 1;
I2C->I2CBAUD = 4000;
I2C->I2CSADR = 0x50;
I2C->I2CINTMASK = 0x200;
I2C->I2CSDAT3_0_b.SDAT0 = 0xAA;
I2C->I2CSDAT3_0_b.SDAT1 = 0xA5;
I2C->I2CSDAT3_0_b.SDAT2 = 0x0F;
I2C->I2CSDAT3_0_b.SDAT3 = 0xF0;
I2C->I2CSDAT7_4_b.SDAT4 = 0x00;
I2C->I2CSDAT7_4_b.SDAT5 = 0xFF;
I2C->I2CCON = 0x24F0;
I2C->I2CTRNUM = 0x08;
while(1){
    if(I2C->I2CSTA_b.INTSTAS & BIT9)
    {
        I2C->I2CINTCLR = BIT9;
        RECV2_BUF[BufPtr] = I2C->I2CRDAT3_0_b.RDAT0;
        BufPtr++;
        RECV2_BUF[BufPtr] = I2C->I2CRDAT3_0_b.RDAT1;
        BufPtr++;
        RECV2_BUF[BufPtr] = I2C->I2CRDAT3_0_b.RDAT2;
        BufPtr++;
        RECV2_BUF[BufPtr] = I2C->I2CRDAT3_0_b.RDAT3;
    }
}
    
```

```

        BufPtr++;
        RECV2_BUF[BufPtr] = I2C->I2CRDAT7_4_b.RDAT4;
        BufPtr++;
        RECV2_BUF[BufPtr] = I2C->I2CRDAT7_4_b.RDAT5;
        BufPtr++;
        RECV2_BUF[BufPtr] = I2C->I2CRDAT7_4_b.RDAT6;
        BufPtr++;
        RECV2_BUF[BufPtr] = I2C->I2CRDAT7_4_b.RDAT7;
        BufPtr++;
    }
}

```

## 2. 多字节连续模式

```

SYSPEN->PERIEN_b.I2C_EN = 1;
I2C->I2CINTMASK = 0xFFFF;
I2C->I2CINTCLR = 0xFFFF;
I2C->I2CSADR = 0x50;    //从机地址注意此处的地址需要处理，得到真实地址。将
真实地址右移一位为此处设置
I2C->I2CCON = 0xF0;
I2C->I2CTRNUM = 0x03;
while(I2C->I2CSTA == 0x00042900)
{
    I2C->I2CINTCLR = 0x0004;
}
while(I2C->I2CSTA == 0x02800106)
{
    I2C->I2CINTCLR = 0x0280;
}

```

### 20.4.11. 示例程序：从机模式，重启

```

SYSPEN->PERIEN_b.I2C_EN = 1;
I2C->I2CINTMASK = 0xFFFF;
I2C->I2CINTCLR = 0xFFFF;
I2C->I2CSADR = 0x50;
I2C->I2CCON = 0xB0;
I2C->I2CTRNUM = 0x01;
while(I2C->I2CSTA == 0x00042900)

```

```

{
    I2C->I2CINTCLR = 0x0004;
}
while(I2C->I2CSTA == 0x00802901)
{
    I2C->I2CINTCLR = 0x0080;
}
while(I2C->I2CSTA == 0x01000900)
{
    I2C->I2CINTCLR = 0x0100;
    I2C->I2CCON = 0xF0;
    I2C->I2CTRNUM = 0x02;
}
while(I2C->I2CSTA == 0x00023900)
{
    I2C->I2CINTCLR = 0x0002;
}
while(I2C->I2CSTA == 0x00483d02)
{
    I2C->I2CINTCLR = 0x0048;
}
while(I2C->I2CSTA == 0x02001100)
{
    I2C->I2CINTCLR = 0x0200;
}
    
```

## 21. 串行同步通信收发器(SPI)

SPI 可以进行高速的同步数据传输，其特点如下：

- 全双工、同步数据传输，支持 4 种极性的通信协议
- 支持主机或从机操作
- 波特率可选
- 写冲突检测

## 21.1. 功能框图

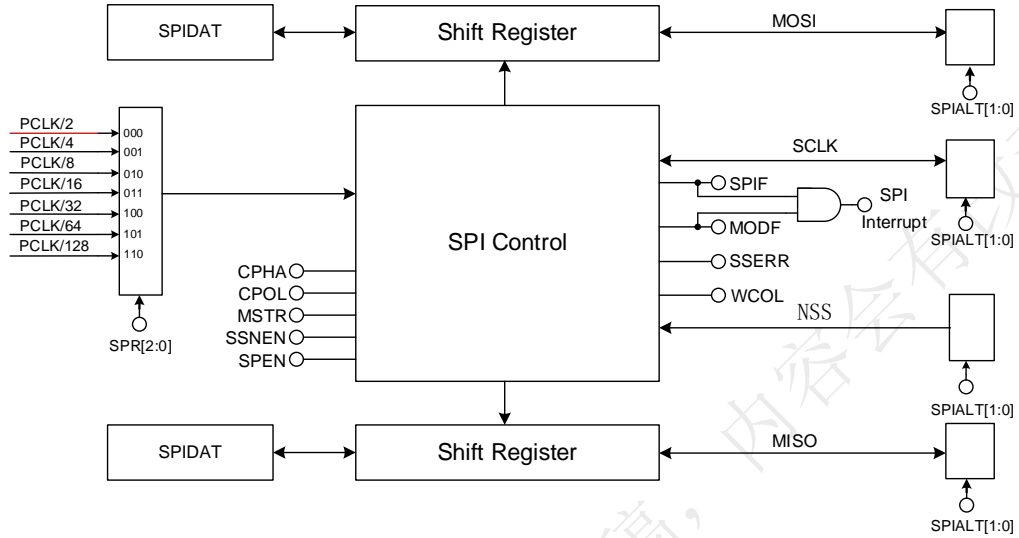


图 21-1 SPI 功能框图

## 21.2. 模块寄存器总表

SPI 控制寄存器的主要寄存器列表如下:

寄存器名	地址	结构体(.h)	寄存器描述	复位值
SPICON	0x40009000	SPI	SPI控制寄存器	0x00
SPISTA	0x40009004	SPI	SPI状态寄存器	0x01
SPIDAT	0x40009008	SPI	SPI数据寄存器	0x00

## 21.3. 寄存器说明

### 21.3.1. SPI 控制寄存器(SPICON: 0x40009000)

描述位	名称	读写	描述	复位值
1:0	SPR1_0	R/W	SPR[2:0]波特率选择位，配合第7位SPR。 000: PCLK /2 001: PCLK /4 010: PCLK /8 011: PCLK /16 100: PCLK /32 101: PCLK /64 110: PCLK /128	000
2	CPHA	R/W	时钟相位。 CPOL=0时: 0: SCLK上升沿采样数据 1: SCLK下降沿采样数据 CPOL=1时: 0: SCLK下降沿采样数据 1: SCLK上升沿采样数据 片选控制: 0: SSN下降沿表示从机片选有效 1: SSN低电平或SSNEN=1表示从机片选有效	0
3	CPOL	R/W	时钟极性选择位。 0: 时钟空闲时SCLK为低电平 1: 时钟空闲时SCLK为高电平	0
4	MSTR	R/W	主/从选择位。 0: 从机模式 1: 主机模式	0
5	SSNEN	R/W	SS片选内部使能。(作为从机时，设置此寄存器为1时，无需外部给片选使能信号，可节约一个引脚) 0: 无效 1: 有效	0
6	SPEN	R/W	SPI使能信号。 0: 无效 1: 有效	0
7	SPR2		波特率选择位。配合第0位和第1位	

### 21.3.2. SPI 状态寄存器(SPISTA: 0x40009004)

描述位	名称	读写	描述	复位值
0	SSTATE	R	从机状态位。 0: 无效 1: 有效	0
1	MSTATE	R	主机状态位。 0: 无效 1: 有效	0
3: 2			保留	
4	MODF	R	模式错误标志位,主机模式时,当外部给NSS低电平,此位置1。只能通过先读SPSTA再写SPICON(任意一bit)来清0。 0: 未出现模式错误 1: 出现模式错误	0
5	SSERR	R	串行同步错误标志位,从机模式,在数据接收完成之前NSS拉高,此位置位1。 只能通过写SPICON中的SPEN=0来清0。 0: 未出现串行同步错误 1: 出现串行同步错误	0
6	WCOL	R	写冲突标志位,在发送当中对SPI数据寄存器SPIDAT写数据将置位WCOL。可以通过先读SPISTA再读SPIDAT来清零。 0: 未出现写冲突 1: 出现写冲突	0
7	SPIF	R/W	SPI中断标志位,串行发送结束后,SPIF置位。若中断使能,SPI中断响应后自动清零标志位。或者通过先读SPISTA寄存器,再访问SPIDAT来清零SPIF。 0: 未产生SPI中断 1: 产生SPI中断	0

### 21.3.3. SPI 数据寄存器(SPIDAT: 0x40009008)

描述位	名称	读写	描述	复位值
7: 0	SPIDAT	R/W	SPI发送/接收数据。	0x00

## 21.4. 功能描述

配置为 SPI 主机时，SPI 接口不自动控制 NSS，必须由软件配置其它 GPIO 来实现从机 NSS，同时不要将 PB9\PB13 复用为 NSS 功能。对 SPIDAT 数据寄存器写入数据即启动 SPI 时钟，将 8 位数据通过 MOSI 引脚移入从机，同时通过 MISO 引脚读取从机的 8 位数据。传输结束后 SPI 时钟停止，传输结束标志 SPIF 置位。如果此时 SPI 中断使能，就会产生中断。主机可以继续往 SPIDAT 写入数据以移位到从机中去，或者将从机的 NSS 拉高说明数据包发送完成。最后进来的数据将一直保存与缓冲器。

配置为从机时，可以将 PB9\PB13 复用为 NSS 功能(或者 SSNEN=1 直接使 NSS 有效)，只要 NSS 为高，SPI 接口将一直保持睡眠状态。在这个状态下，软件可以将需要发送给主机的数据写到 SPI 数据寄存器 SPIDAT 中。即使此时 SCLK 有输入时钟，SPIDAT 的数据也不会移出，直至 NSS 被拉低。当 NSS 被主机拉低后，且从机接收到 SCLK 时钟，SPIDAT 寄存器中的 8 位数据通过 MISO 引脚输出给主机，同时 MOSI 引脚读取主机的 8 位数据。当 8 位数据传输完成后，传输结束标志 SPIF 置位。如果此时 SPI 中断使能，将产生中断。在读取移入的数据之前从机可以继续往 SPDAT 写入数据。最后进来的数据将一直保存于缓冲器。

SPI 中断源有两个：分别是 SPIF 标志置 1 和 MODF 标志置 1。如果 SPI 中断使能，都将产生 SPI 中断。

SPI 模块支持 4 种类型的通信时序，由 CPOL 和 CPHA 两位设定。主从机通信时需要配置成相同的通信时序，否则无法正确发送接收数据。

{CPOL,CPHA}	起始沿	结束沿
00	采样(上升沿)	设置(下降沿)
01	设置(上升沿)	采样(下降沿)
10	采样(下降沿)	设置(上升沿)
11	设置(下降沿)	采样(上升沿)

表 21-4 SPI 时序设置

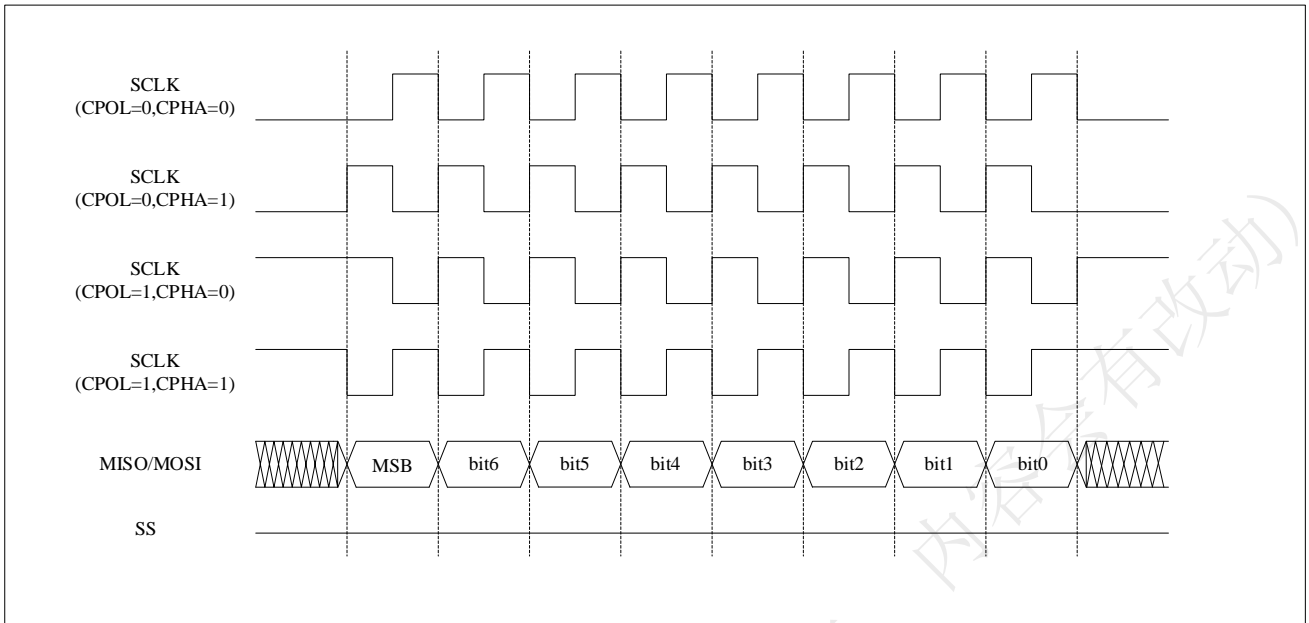


图 21-4-1 主机数据时序图

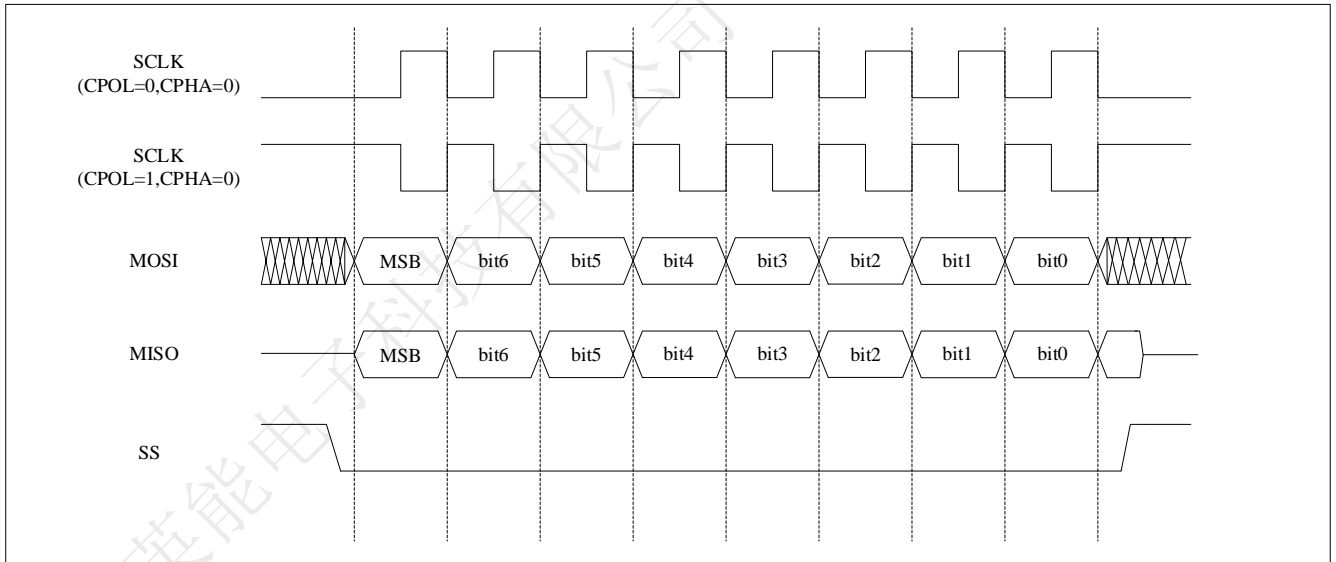


图 21-4-2 从机数据时序 (CPHA=0)

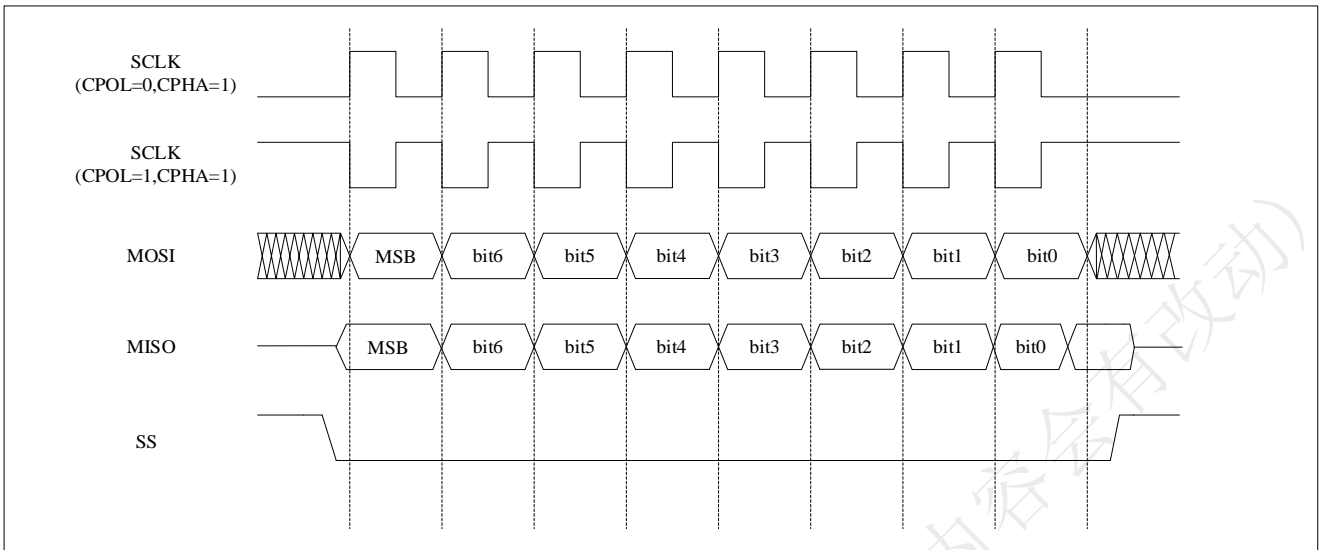


图 21-4-3 从机数据时序 (CPHA=1)

## 22. 模拟比较器(ACMP0/1/2/3)

共有 4 个模拟比较器，分别是 ACMP0/1/2/3。ACMP3 为独立的模拟比较器，可用检测母线电流。ACMP0/1/2 共用同一个模拟模块，通过扫描的方式实现 3 个模拟比较器的功能。ACMP0/1/2 可用于三路霍尔元件(Hall element)过零信号检测，或者反电动势(BEMF)过零信号检测。

### ACMP3 功能介绍:

- 轨至轨输入
- 输入迟滞可选:
  - CMP3\_FREN =0 时，无迟滞， $\pm 5\text{mv}$ ， $\pm 10\text{mv}$ ， $\pm 30\text{mv}$
  - CMP3\_FREN =1 时，无迟滞， $\pm 10\text{mv}$ ， $\pm 25\text{mv}$ ， $\pm 60\text{mv}$
- 内置多档分压可选，可以用作比较器正端输入
- 比较器输入可选择外部管脚，DAC 输出，OPA 输出
- 支持输出数字滤波
- 比较器输出可以用于触发中断
- 用于母线电流限流检测。比较器输出 CP30 可以作为 PWM 模块的故障输入信号，通过 REMAP1 的 FLTALT 进行设置

### ACMP0/1/2 功能介绍:

- 轨至轨输入
- 输入迟滞可选:
  - CMP012\_FREN =0 时, 无迟滞,  $\pm 5\text{mv}$ ,  $\pm 10\text{mv}$ ,  $\pm 30\text{mv}$
  - CMP012\_FREN =1 时, 无迟滞,  $\pm 10\text{mv}$ ,  $\pm 25\text{mv}$ ,  $\pm 60\text{mv}$
- 内置多档分压可选, 可以用作比较器正端输入
- 当共享模式使能时, 可以作为 3 个独立工作的模拟比较器
- 内置一个假中心点生成电路
- 比较器输入可选择外部管脚, 假中心点
- 支持输出数字滤波
- 支持对 PWM 输出进行消隐(Blank)控制
- 比较器输出可以用于触发中断
- 比较器输出 CP0/1/2O 可映射到外部唤醒中断 EINT3, 作为外部中断信号
- 用于三路“Hall element”输入检测或“反电势过零检测”输入检测

## 22.1.功能框图

### 22.1.1. ACMP0/1/2 框图

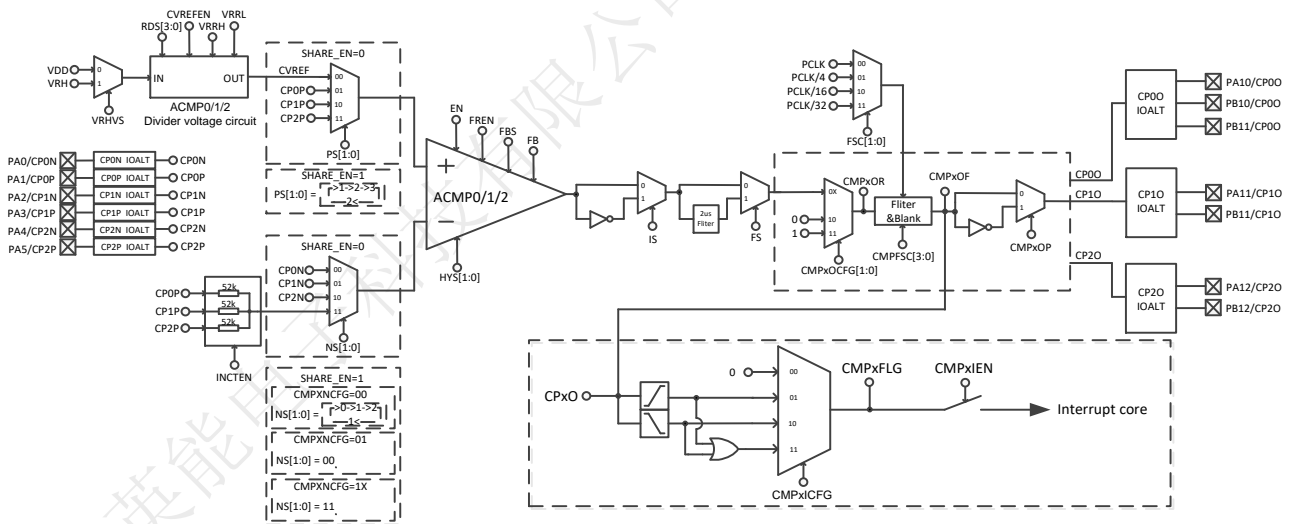


图 22-1-1 比较器 012 模块功能框图

### 22.1.2. ACMP3 框图

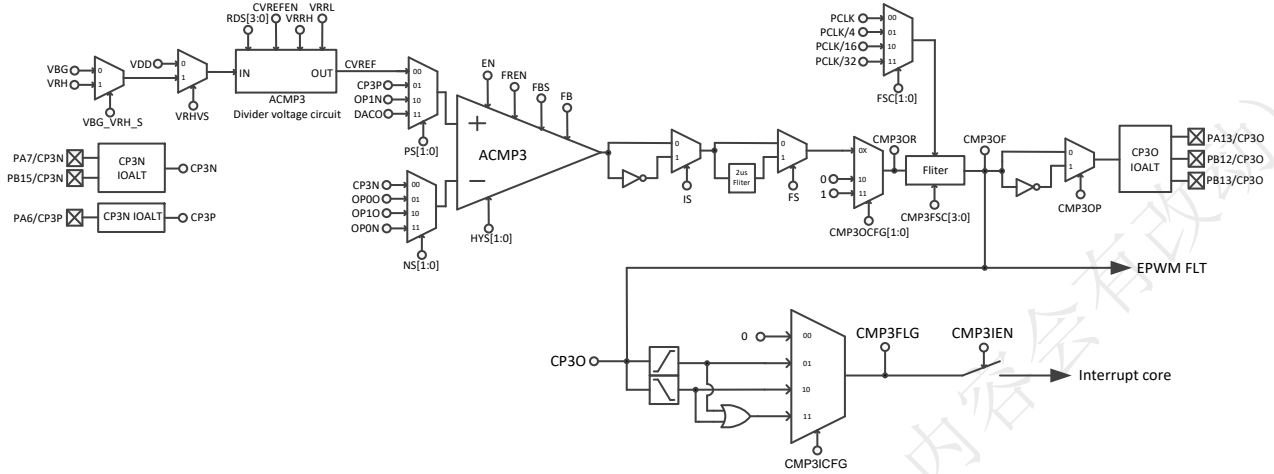


图 22-1-2 比较器 3 模块功能框图

### 22.1.3. 内置分压模块框图

浙江英能电子科技有限公司 (初稿)

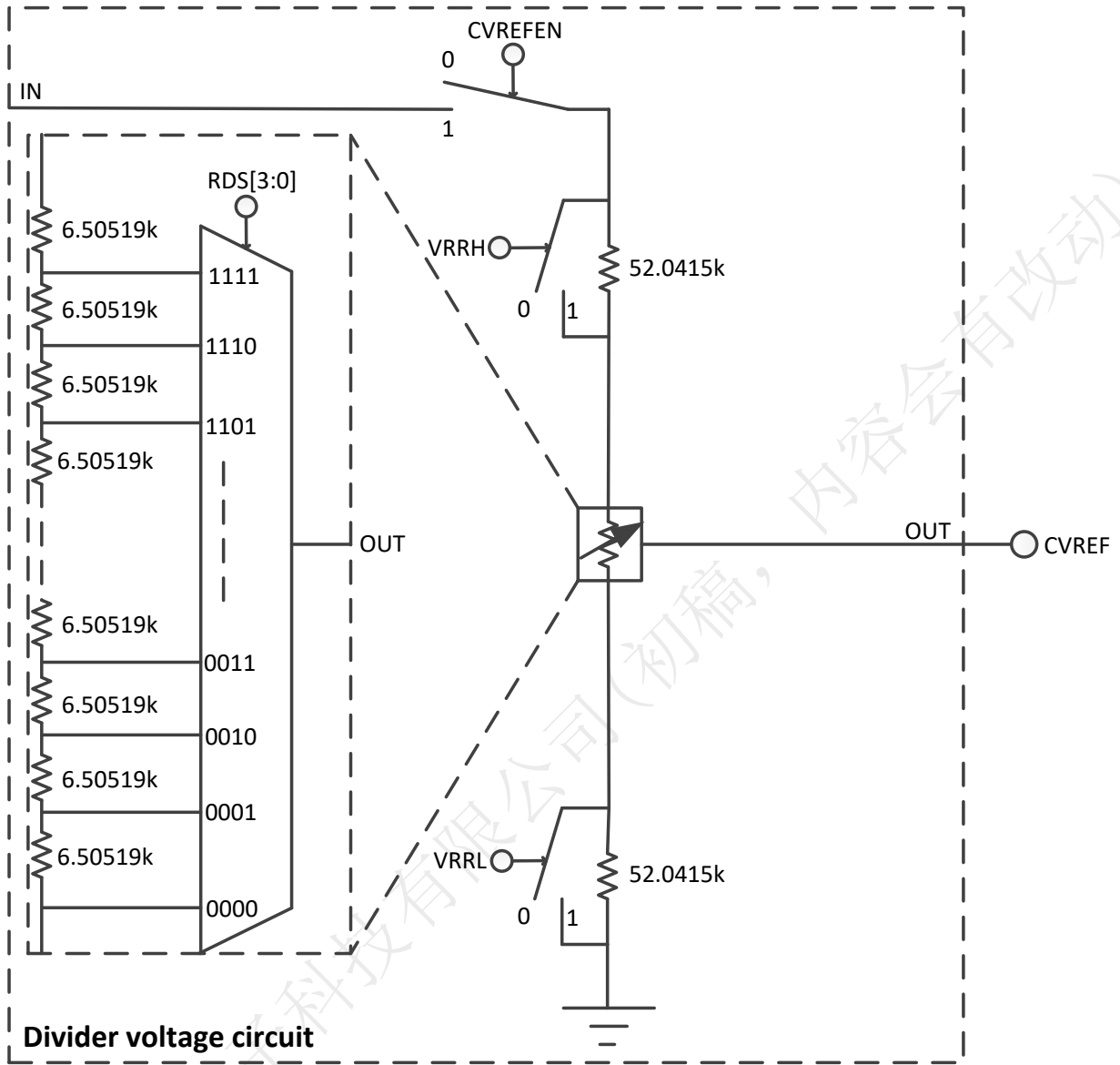


图 22-2-3 比较器分压模块功能框图

## 22.2. 模块寄存器总表

比较器 CMP0/1/2/3 寄存器列表如下：

寄存器名	地址	结构体(.h)	寄存器描述	复位值
CMPOCFG	0x40005800	ACMP	比较器输出配置寄存器	0x0000
CMPFILT	0x40005804	ACMP	比较器滤波配置寄存器	0x0000
CMPOSTA	0x40005808	ACMP	比较器输出状态寄存器	0x00
CMPINTC	0x4000580C	ACMP	比较器中断控制寄存器	0x0000
CMP012CON	0x40005810	ACMP	ACMP012控制寄存器	0x0000000

CMP3CON	0x40005814	ACMP	ACMP3控制寄存器	0x000000
---------	------------	------	------------	----------

## 22.3.寄存器说明

### 22.3.1. 比较器 012 控制寄存器(CMP012CON: 0x40005810)

描述位	名称	读写	描述	复位值
0	CMP012EN	R/W	ACMP0/1/2使能。 0: ACMP0/1/2关闭 1: ACMP0/1/2使能	0
1	CMP012_FREN	R/W	ACMP0/1/2响应时间配置信号。 0: 正常响应 1: 响应时间减小	0
2	CMP012_VRRH	R/W	正端分压电阻范围选择位, 工作原理可参考22.1.3章节。	0
3	CMP012_VRRL	R/W	负端分压电阻范围选择位, 工作原理可参考22.1.3章节。	0
4	CMP012_VRHVS	R/W	ACMP0/1/2的分压电阻供电源选择。 0: VDD作为供电源 1: VRH作为供电源	0
5	CMP012_CVREFEN	R/W	ACMP0/1/2内部分压电路使能位。 0: 不开启内部分压电路 1: 开启内部分压电路	0
6	CMP012_IS	R/W	ACMP0/1/2输出极性选择位。 0: 正常输出 1: 反向输出	0
7	CMP012_FS	R/W	ACMP0/1/2输出滤波选择位。 0: 不滤波 1: 使能滤波	0
8	CMP012_FBS	R/W	ACMP0/1/2迟滞反馈方式选择。 0: 选择数字反馈 1: 选择比较器输出直接反馈	0
9	CMP012_FB	R/W	ACMP0/1/2迟滞反馈信号, 在CMP_SHARE_EN和CMP012_FBS为0有效。 0: 比较点强制偏低 1: 比较点强制偏高	0
10	INCTEN	R/W	中心电压模块使能控制位 0: 关闭 1: CP0P/CP1P/CP2P输入通过电阻形成一个中心点电压	0

描述位	名称	读写	描述	复位值
11	CMP_SHARE_EN	R/W	ACMP0/1/2 轮换模式使能寄存器。 0: 比较器轮换模式关闭。(CMP012_FB, CMP012_FB, CMP012_NS[1: 0], CMP012_PS[1: 0] 由寄存器控制, 通过 ACMP0 路径输出。)  1: 比较器轮换模式使能。ACMP0/1/2 输入自动轮换, 实现 3 个比较器的功能。	0
15: 12	CMP012_RDS[3: 0]	R/W	内部分压模块分压点选择位, 工作原理可参考 22.1.3 章节。	0000
17: 16	CMP012_NS[1: 0]	R/W	ACMP0/1/2 负端输入选择。 00: CP0N(PA0 模拟通道1) 01: CP1N(PA2 模拟通道1) 10: CP2N(PA4 模拟通道1) 11: 选择中心点电压(需要使能 INCTEN)	00
19: 18	CMP012_PS[1: 0]	R/W	ACMP0/1/2 正端输入选择。 00: CVREF 01: CP0P(PA1 模拟通道1) 10: CP1P(PA3 模拟通道1) 11: CP2P(PA5 模拟通道1)	00
21: 20	CMP012HYS[1: 0]	R/W	ACMP0/1/2 的迟滞参数配置。 CMP012_FREN=0 时: 00: 迟滞电压为 0mV, 即关闭迟滞功能 01: 迟滞电压为 20mV, 即 ±10mV 10: 迟滞电压为 50mV, 即 ±25mV 11: 迟滞电压为 120mV, 即 ±60mV CMP012_FREN=1 时: 00: 迟滞电压为 0mV, 即关闭迟滞功能 01: 迟滞电压为 10mV, 即 ±5mV 10: 迟滞电压为 20mV, 即 ±10mV 11: 迟滞电压为 60mV, 即 ±30mV	00
23: 22	CMPXNCFG	R/W	ACMP0/1/2 负端信号源互连配置。 00: ACMP0/1/2 负端信号独立输入 01: ACMP0/1/2 负端信号互相短路, 输入到 ACMP0/1/2 的负端(适用于三路负端相同, 可节约两个 IO)  1X: 选择中心点模块产生的电压(INCTEN 需要置 1)	00
25: 24	CMP012HYSM[1: 0]	R/W	ACMP0/1/2 的迟滞模式配置(在 CMP_SHARE_EN=1 有效)。	00

描述位	名称	读写	描述	复位值
			0X: 正常迟滞连接 10: 比较点强制偏高 11: 比较点强制偏高低	
31: 26			保留	

### 22.3.2. ACMP012 输出配置寄存器(CMPOCFG: 0x40005800)

描述位	名称	读写	描述	复位值
0	CMP0OP	R/W	ACMP0输出极性设置。 0: CP00由CMP00F的正常极性输出 1: CP00由CMP00F的反向极性输出	0
1	CMP1OP	R/W	ACMP1输出极性设置。 0: CP10由CMP10F的正常极性输出 1: CP10由CMP10F的反向极性输出	0
2	CMP2OP	R/W	ACMP2输出极性设置。 0: CP20由CMP20F的正常极性输出 1: CP20由CMP20F的反向极性输出	0
3	CMP3OP	R/W	ACMP3输出极性设置。 0: CP30由CMP30F的正常极性输出 1: CP30由CMP30F的反向极性输出	0
5: 4	CMP0OCFG [1: 0]	R/W	ACMP0输出信号配置。 0X: CP00正常输出 10: CP00强制为低电平输出 11: CP00强制为高电平输出	00
7: 6	CMP1OCFG [1: 0]	R/W	ACMP1输出信号配置。 0X: CP10正常输出 10: CP10强制为低电平输出 11: CP10强制为高电平输出	00
9: 8	CMP2OCFG [1: 0]	R/W	ACMP2输出信号配置。 0X: CP20正常输出 10: CP20强制为低电平输出 11: CP20强制为高电平输出	00
11: 10	CMP3OCFG [1: 0]	R/W	ACMP3输出信号配置。 0X: CP30正常输出 10: CP30强制为低电平输出 11: CP30强制为高电平输出	00
15: 12			保留	

### 22.3.3. ACMP3 控制寄存器(CMP3CON: 0x40005814)

描述位	名称	读写	描述	复位值
0	CMP3EN	R/W	ACMP3使能。 0: ACMP3关闭 1: ACMP3使能	0
1	CMP3_FREN	R/W	ACMP3响应时间配置信号。 0: 正常响应 1: 响应时间减小	0
2	CMP3_VRRH	R/W	正端分压电阻范围选择位，工作原理可参考22.1.3章节。	0
3	CMP3_VRRL	R/W	负端分压电阻范围选择位，工作原理可参考22.1.3章节。	0
4	CMP3_VRHVS	R/W	ACMP3供电源选择。 0: VDD作为供电源 1: VRH作为供电源	0
5	CMP3_CVREFEN	R/W	ACMP3内部参考电压使能位。 0: 不开启内部参考电压 1: 开启内部参考电压	0
6	CMP3_IS	R/W	ACMP3输出极性选择位。 0: 正常输出 1: 反向输出	0
7	CMP3_FS	R/W	ACMP3输出滤波选择位。 0: 不滤波 1: 使能滤波	0
8	CMP3_FBS	R/W	ACMP3反馈方式选择。 0: 选择数字反馈 1: 选择比较器直接反馈	0
9	CMP3_FB	R/W	ACMP3迟滞反馈信号，在CMP3_FBS为0有效。 0: 负迟滞 1: 正迟滞	0
11: 10			保留	
15: 12	CMP3_RDS[3: 0]	R/W	内部参考电压VDD源分压选择位，工作原理可参考22.1.3章节。	0
17: 16	CMP3_NS[1: 0]	R/W	ACMP3负端输入选择。 00: CP3N(PA7模拟通道2、PB15模拟通道2) 01: OP00 10: OP10 11: OP0N	00

19: 18	CMP3_PS[1: 0]	R/W	ACMP3正端输入选择。 00: CVREF 01: CP3P(PA6 模拟通道1) 10: OP1N 11: DAC_OUT	00
21: 20	CMP3HYS[1: 0]	R/W	ACMP3的迟滞参数配置。 CMP3_FREN=0时: 00: 迟滞电压为0mV, 即关闭迟滞功能 01: 迟滞电压为20mV, 即±10mV 10: 迟滞电压为50mV, 即±25mV 11: 迟滞电压为120mV, 即±60mV CMP3_FREN=1时: 00: 迟滞电压为0mV, 即关闭迟滞功能 01: 迟滞电压为10mV, 即±5mV 10: 迟滞电压为20mV, 即±10mV 11: 迟滞电压为60mV, 即±30mV	00
31: 22			保留	

#### 22.3.4. ACMP0/1/2/3 滤波配置寄存器 0(CMPFILT: 0x40005804)

描述位	名称	读写	描述	复位值
5: 0	BLANKTIME [5: 0]	R/W	PWM边沿Blanking Time时间设置, 最长63个采样时钟(CMP_CLK)。当BLANKTIME[5: 0] = 000000时, PWM边沿Blanking Time功能关闭。采样时钟的频率由寄存器CMPFILT的FSC[1: 0]设置。	
7: 6	FSC[1: 0]	R/W	滤波器采样时钟(CMP_CLK)配置。 00: 1个系统外设时钟(PCLK) 01: 4个系统外设时钟(PCLK) 10: 16个系统外设时钟(PCLK) 11: 32个系统外设时钟(PCLK)	00
11: 8	CMP3FSC[3: 0]	R/W	ACMP3的滤波时间设置, 最长15个采样时钟(CMP_CLK)。 0000: 滤波功能关闭; 0001: 滤波功能相当于定时采样。采样时钟的频率由寄存器CMPFILT的FSC[1: 0]设置。	0000
15: 12	CMP012FSC[3: 0]	R/W	ACMP0/1/2的滤波时间设置, 最长15个采样时钟(CMP_CLK)。 0000: 滤波功能关闭;	0000

描述位	名称	读写	描述	复位值
			0001: 滤波功能相当于定时采样。采样时钟的频率由寄存器CMPFILT的FSC[1: 0]设置。	

### 22.3.5. ACMP0/1/2/3 输出状态寄存器(CMPOSTA: 0x40005808)

描述位	名称	读写	描述	复位值
0	CMP0OF	R	比较器输出CMP0O滤波结果。 0: 低电平 1: 高电平	0
1	CMP1OF	R	比较器输出CMP1O滤波结果。 0: 低电平 1: 高电平	0
2	CMP2OF	R	比较器输出CMP2O滤波结果。 0: 低电平 1: 高电平	0
3	CMP3OF	R	比较器输出CMP3O滤波结果。 0: 低电平 1: 高电平	0
4	CMP0OR	R	比较器输出CMP0O原始信号。 0: 低电平 1: 高电平	0
5	CMP1OR	R	比较器输出CMP1O原始信号。 0: 低电平 1: 高电平	0
6	CMP2OR	R	比较器输出CMP2O原始信号。 0: 低电平 1: 高电平	0
7	CMP3OR	R	比较器输出CMP3O原始信号。 0: 低电平 1: 高电平	0

### 22.3.6. ACMP0/1/2/3 中断控制寄存器(CMPINTC: 0x4000580C)

描述位	名称	读写	描述	复位值
1: 0	CMP0ICFG[1: 0]	R/W	ACMP0输出CMP0OF的中断配置。 00: 上升/下降沿都不产生中断 01: 仅上升沿产生中断 10: 仅下降沿产生中断	00

描述位	名称	读写	描述	复位值
			11: 上升/下降沿都产生中断	
3: 2	CMP1ICFG[1: 0]	R/W	ACMP1输出CMP1OF的中断配置。 00: 上升/下降沿都不产生中断 01: 仅上升沿产生中断 10: 仅下降沿产生中断 11: 上升/下降沿都产生中断	00
5: 4	CMP2ICFG[1: 0]	R/W	ACMP2输出CMP2OF的中断配置。 00: 上升/下降沿都不产生中断 01: 仅上升沿产生中断 10: 仅下降沿产生中断 11: 上升/下降沿都产生中断	00
7: 6	CMP3ICFG[1: 0]	R/W	ACMP3输出CMP3OF的中断配置。 00: 上升/下降沿都不产生中断 01: 仅上升沿产生中断 10: 仅下降沿产生中断 11: 上升/下降沿都产生中断	00
8	CMP0IEN	R/W	ACMP0输出中断使能。 0: 禁止 1: 使能	0
9	CMP1IEN	R/W	ACMP1输出中断使能。 0: 禁止 1: 使能	0
10	CMP2IEN	R/W	ACMP2输出中断使能。 0: 禁止 1: 使能	0
11	CMP3IEN	R/W	ACMP3输出中断使能。 0: 禁止 1: 使能	0
12	CMP0FLG	R/W	ACMP0输出中断标志。中断标志需要软件写1清零。 0: 未产生ACMP0输出中断 1: 产生ACMP0输出中断	0
13	CMP1FLG	R/W	ACMP1输出中断标志。中断标志需要软件写1清零。 0: 未产生ACMP1输出中断 1: 产生ACMP1输出中断	0
14	CMP2FLG	R/W	ACMP2输出中断标志。中断标志需要软件写1清零。 0: 未产生ACMP2输出中断 1: 产生ACMP2输出中断	0

描述位	名称	读写	描述	复位值
15	CMP3FLG	R/W	ACMP3输出中断标志。中断标志需要软件写1清零。 0: 未产生ACMP3输出中断 1: 产生ACMP3输出中断	0

### 22.3.7. 模拟功能控制寄存器 (ANA\_CON:0x4000E048)

描述位	名称	读写	描述	复位值
4: 0			保留	
6: 5	VRHS[1: 0]	R/W	见5.4.3章节	
8: 7	VREFS[1: 0]	R/W	见5.4.3章节	
9	VTSEN	R/W	见5.5.2章节	
10	AMUX2_S	R/W	见23.3.4章节	
11	VTS_SEL	R/W	见5.5.2章节	
12	VBGBUF_EN	R/W	见5.4.3章节	
13	VBG_VRH_S	R/W	ACMP0/1/2/3内部分压电路供电选择: 0: 选择VBG 1: 选择VRH	0
15: 14	AMUX_S[1: 0]	R/W	见5.4.3章节	

## 22.4.功能描述

### 22.4.1. ACMP0/1/2/3 比较器迟滞功能

ACMP0/1/2 比较器迟滞电压设置轮换模式和非轮换模式下设置是不一样的。

#### ➤ 轮换模式(CMP\_SHARE\_EN=1)

由 CMP012HYSM[1: 0]设置。CMP012HYSM[1: 0] = 0X 时为正常迟滞模式下，迟滞可以避免输入噪声造成比较器的输出跳变。

CMP012HYSM[1: 0] = 10 或 11 为强制模式，该模式下比较器偏置电压工作在强制高或低的状态，不随比较器输出变化。

迟滞电压由 CMP012HYS[1: 0]设置。

CMP012HYS[1: 0]	FREN=0	FREN=1
00	无迟滞	无迟滞
01	±10mv	±5mv
10	±25mv	±10mv
11	±60mv	±30mv

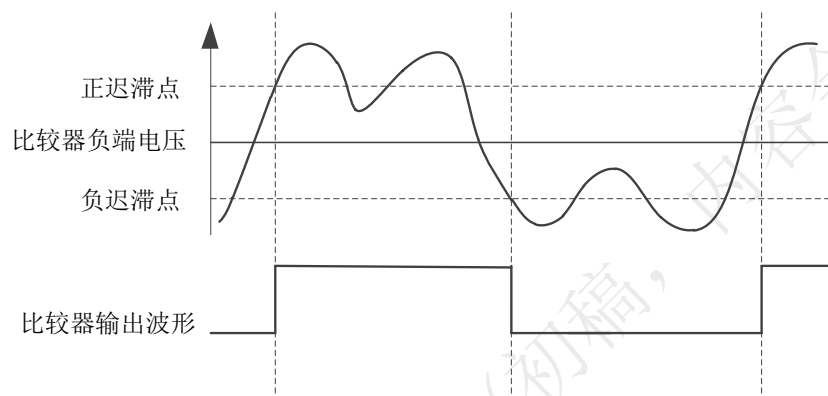


图 22-4-1-1 比较器正常迟滞功能

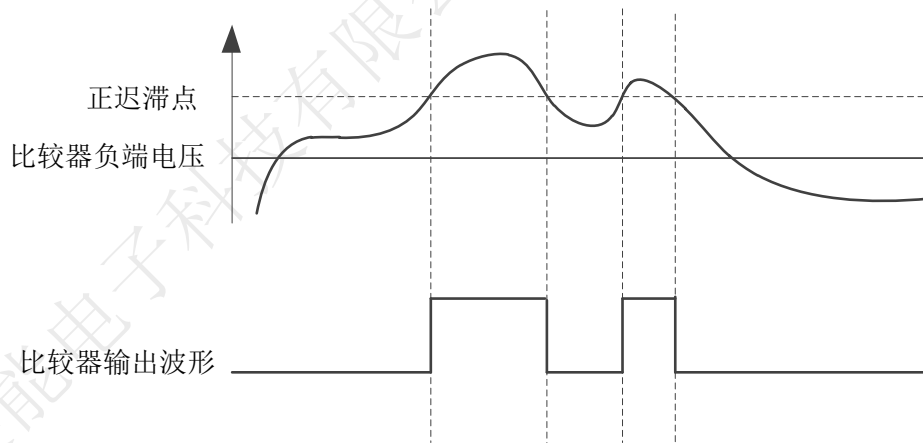


图 22-4-1-2 比较点强制偏高

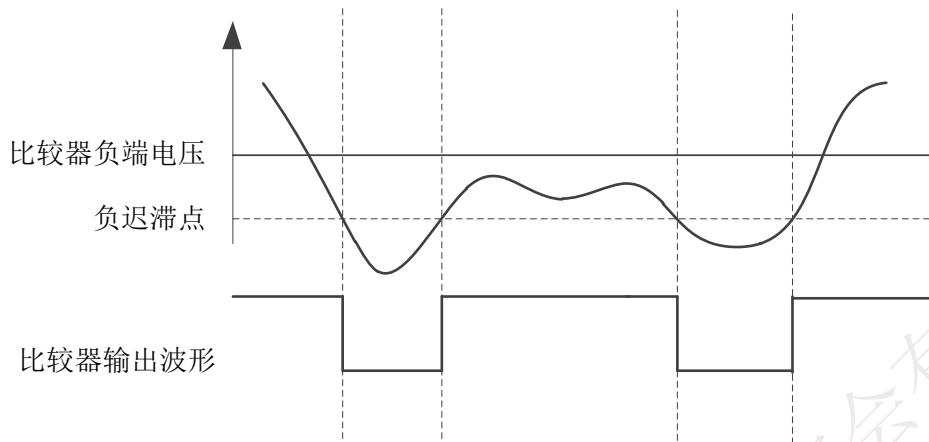


图 22-4-1-3 比较点强制偏低

➤ 非轮换模式(CMP\_SHARE\_EN=0)

在非轮换模式下当 CMP012\_FBS=1 时，迟滞功能选择比较器输出直接反馈，也就是正常迟滞功能，迟滞电压由 CMP012HYS[1: 0]设置。

在非轮换模式下当 CMP012\_FBS=0 时，迟滞反馈由 CMP012\_FB 决定。当 CMP012\_FB=1 时，比较器的比较点强制偏高；当 CMP012\_FB=0 时，比较器的比较点强制偏低。

ACMP3 比较器迟滞功能只支持非轮换模式，即由 CMP3\_FB，CMP3\_FBS，CMP3HYS[1: 0]决定迟滞的工作方式。

### 22.4.2. ACMP0/1/2/3 的输出滤波配置

比较器滤波配置寄存器 CMPFILT 的 FSC[1: 0]用于设置消隐时间(Blanking Time)和滤波器的时钟源 PCLK 的分频系数。

FSC[1: 0]	BlankTime和滤波器的时钟(CMP_CLK)
00	PCLK
01	PCLK/4
10	PCLK/16
11	PCLK/32

比较器滤波配置寄存器 CMPFILT 的 CMPFSC [3: 0]用于设置 ACMP0/1/2 输出滤波器的滤波时间。当 CMPFSC [3: 0] = 0000 时，ACMP0/1/2 的输出滤波功能关闭；当 CMPFSC [3: 0]>0000 时，ACMP0/1/2 的输出滤波为(CMPFSC [3: 0] + 1)/ CMP\_CLK。滤波器的输入/输出信号时序图见下图所示。

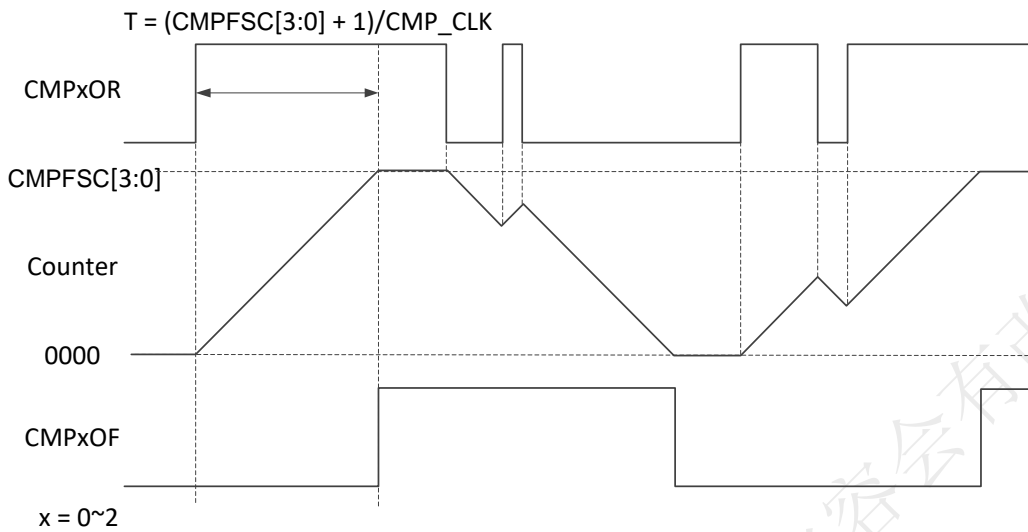


图 22-4-2-1 ACMP0/1/2/3 输出滤波器时序图(CMPxOCFG[1: 0]=0X)

比较器滤波配置寄存器 CMPFILT 的 BLANKTIME[5: 0]用于设置 PWM 边沿消隐 Blanking Time 的计数时间，当 BLANKTIME[5: 0] = 000 000 时，Blanking Time 的 PWM 边沿锁定功能关闭。

当 BLANKTIME[5: 0] > 000 000 时，PWM 边沿 Blanking Time 功能打开。消隐模块的输入信号是 PWM 边沿检测信号，由 PWM 模块提供，只要 PWM00~PWM05 出现信号翻转，PWM 边沿检测信号就会产生一个脉冲启动消隐动作。消隐过程中滤波器计数器将停止工作，只有 BlankingTime 结束后，滤波器才会继续工作，也就是增加了滤波时间，延时长度  $T\_blank = (BLANKTIME[5: 0] + 1) / CMP\_CLK$ 。该功能非常适合“方波无感控制的反电势检测过程”，用于避开 PWM 上升沿或 PWM 下降沿引起的干扰信号。滤波器和 Blanking Time 功能的输入/输出信号时序图见下图所示。

BlankTime 的时间设置与 PWM 边沿导致的干扰信号持续时间有关。一般而言，当 ACMP0/1/2 用于检测反电势过零点时，建议 BlankTime 设置为 10us 左右；滤波时间常数设置为 2~5us 即可。

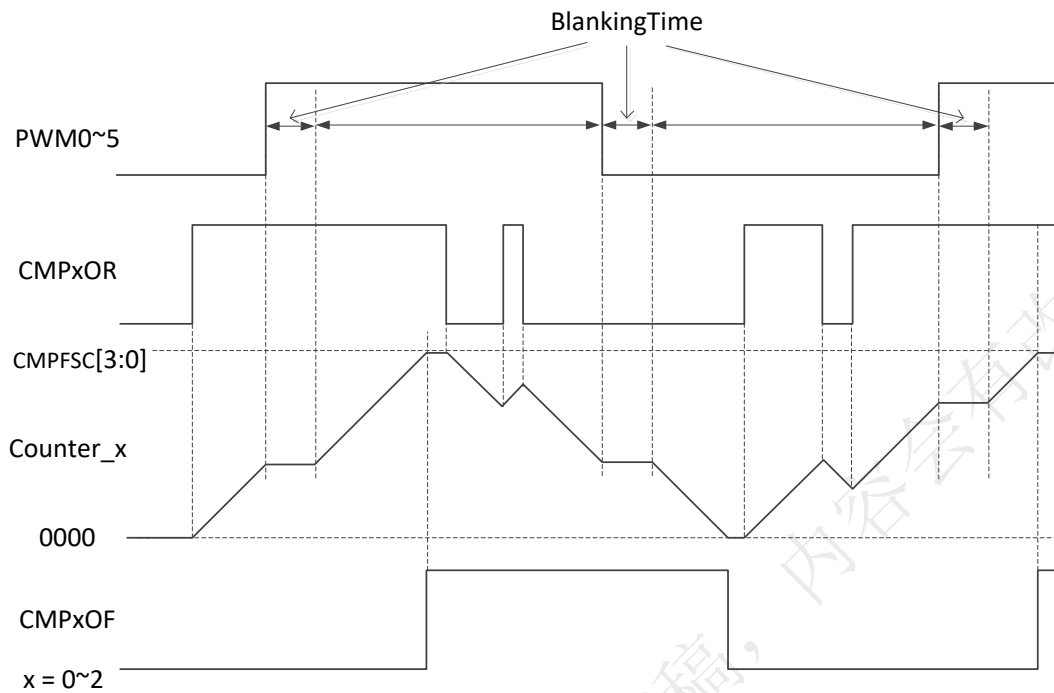


图 22-4-2-2 ACMP0~2 输出滤波器时序图

比较器滤波配置寄存器CMPFILT的CMP3FSC [3: 0]用于设置ACMP3输出滤波器的滤波时间，工作原理与上面描述一致。ACMP3输出没有消隐(Blank)功能。

### 22.4.3. 比较器输出中断

比较器输出出现信号翻转，并且相应的中断沿配置位 CMPxICFG[1: 0]使能时，比较器输出中断标志 CMPxFLG 将会被置 1。如果此时相应的中断使能(CMPxIEN= 1)，CPU 将响应中断。比较器输出中断标志 CMPxFLG 需要由软件清零。

ACMP0/1/2/3 比较器模块一共有四个中断源，对应 1 个中断向量(ACMP0/1/2/3 中断向量)。

ACMP0/1/2 的输出还可以被引入到外部唤醒中断 EINT3，做为外部中断的中断源。详见：[错误!未找到引用源。](#) 章节。

## 23. 运算放大器(OPA0/1/2)

集成 3 个运放，分别是 OPA0/1/2，都可独立配置。用于模拟信号放大，输出送到比较器或 ADC，实现模拟信号检测。

运算放大器(OPA)特点如下：

- 轨至轨输入输出

- 内置可编程增益  
同相单端放大模式：5/9/13/17/21/25/33/41  
差分放大模式：4/8/12/16/20/24/32/40
- 增益系数精度 +/- 1%
- 输入失调电压：  
未校准前：±5mV  
校准后：±1mV
- 失调电压可校准

### 23.1.功能框图

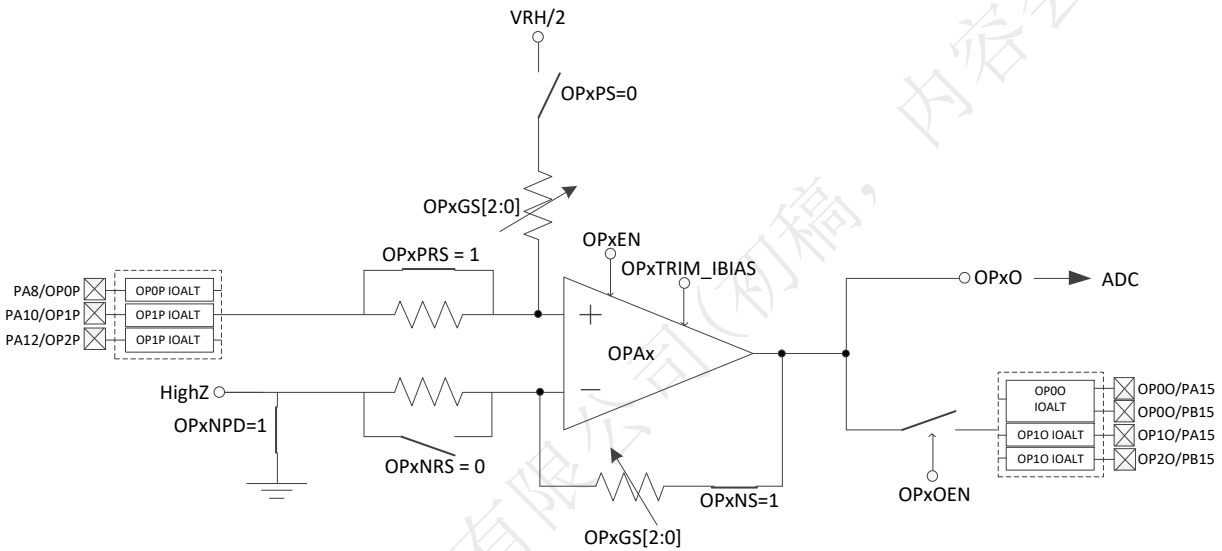
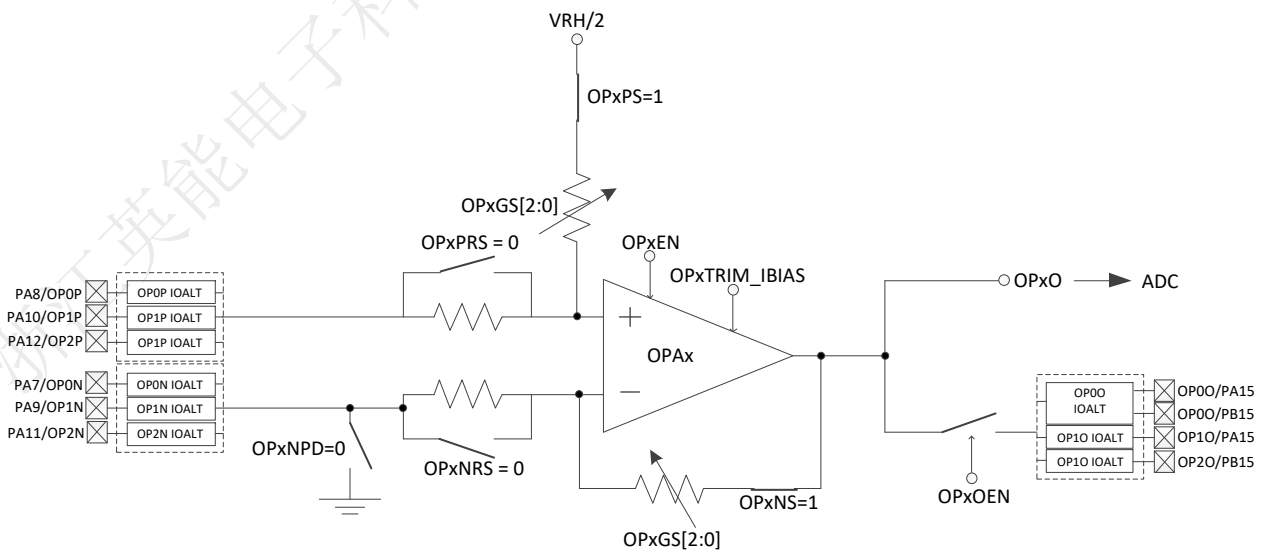


图 23-1-1 同向单端放大模式



图

## 23-1-2 差分放大模式

## 23.2. 模块寄存器总表

OPA0、OPA1 控制寄存器列表如下:

寄存器名	地址	结构体(.h)	寄存器描述	复位值
OPA0CON	0x40006800	OPA	OPA0控制寄存器	0x00000000
OPA1CON	0x40006804	OPA	OPA1控制寄存器	0x00000000
OPA2CON	0x40006808	OPA	OPA2控制寄存器	0x00000000

## 23.3. 寄存器说明

### 23.3.1. OPA0 控制寄存器(OPA0CON: 0x40006800)

描述位	名称	读写	描述	复位值	
0	OP0EN	R/W	OPA0运放模式使能。 0: 关闭 1: 使能	0	
1	OP0NS	R/W	放大器反馈通路开关。 0: 断开负反馈回路 1: 连接	0	
2	OP0NPD	R/W	负输入端接地。 0: 不使能 1: 将OP0N接地	0	
3	OP0PS	R/W	放大器正端偏置开关。 0: 关闭 1: 使能	0	
4	OP0OEN	R/W	OP0O输出到端口使能位。 0: 禁止; 1: 使能。	0	
7: 5	OP0GS[2: 0]	R/W	OP0内部增益选择位		000
			同相单端输入模式: 000: 5x 001: 9x 010: 13x 011: 17x 100: 21x 101: 25x 110: 33x 111: 41x	差分输入模式: 000: 4x 001: 8x 010: 12x 011: 16x 100: 20x 101: 24x 110: 32x 111: 40x	

描述位	名称	读写	描述	复位值
8	OP0PRS	R/W	放大器正相输入电阻短路。	0
9	OP0NRS	R/W	放大器反相输入电阻短路。	0
14: 10	OP0OS[4: 0]	R/W	放大器失调电压修调位，默认全0。 00000: OP0N-OP0P最大 10000: 不引入失调修调 11111: OP0P-OP0N最大	00000
15	OP0TRIMEN	R/W	修调使能信号，默认TRIMEN=0。 0: 关闭TRIM功能 1: 开启TRIM功能	0
16	OP0TRIMO	R	修调指示信号。 TRIMEN=0, OP0TRIMO=0; TRIMEN=1, OP0TRIMO跳转为1, 代表修调完成	0
17	OP0TRIM_IBIAS	R/W	运放偏置电流控制位，默认0, 为小电流模式。 0: 小电流模式 1: 大电流模式	0
31: 18			保留	

### 23.3.2. OPA1 控制寄存器(OPA1CON: 0x40006804)

描述位	名称	读写	描述	复位值
0	OP1EN	R/W	OPA1运放模式使能。 0: 关闭 1: 使能	0
1	OP1NS	R/W	放大器反馈通路开关。 0: 断开负反馈回路 1: 连接	0
2	OP1NPD	R/W	负输入端接地。 0: 不使能 1: 将OP1N接地	0
3	OP1PS	R/W	放大器正端偏置开关。 0: 关闭 1: 使能	0
4	OP1OEN	R/W	OP1O输出到端口使能位。 0: 禁止; 1: 使能。	0
7: 5	OP1GS[2: 0]	R/W	OPA1内部增益选择位。 同相单端输入模式:      差分输入模式:	000

描述位	名称	读写	描述	复位值
			000: 5x 001: 9x 010: 13x 011: 17x 100: 21x 101: 25x 110: 33x 111: 41x	000: 4x 001: 8x 010: 12x 011: 16x 100: 20x 101: 24x 110: 32x 111: 40x
8	OP1PRS	R/W	放大器正相输入电阻短路。	0
9	OP1NRS	R/W	放大器反相输入电阻短路。	0
14: 10	OP1OS[4: 0]	R/W	放大器失调电压修调位，默认全0。 00000: OP1N-OP1P最大 10000: 不引入失调修调 11111: OP1P-OP1N最大	00000
15	OP1TRIMEN	R/W	修调使能信号，默认TRIMEN=0，关闭TRIM功能。	0
16	OP1TRIMO	R	修调指示信号。 TRIMEN=0，OP1TRIMO=0； TRIMEN=1，OP1TRIMO跳转为1，代表修调完成。	0
17	OP1TRIM_IBIAS	R/W	运放偏置电流控制位，默认0。 0: 小电流模式 1: 大电流模式	0
31: 18			保留	

### 23.3.3. OPA2 控制寄存器(OPA2CON: 0x40006808)

描述位	名称	读写	描述	复位值
0	OP2EN	R/W	OPA2运放模式使能。 0: 关闭 1: 使能	0
1	OP2NS	R/W	放大器反馈通路开关。 0: 断开负反馈回路 1: 连接	0
2	OP2NPD	R/W	负输入端接地。 0: 不使能 1: 将OP2N接地	0
3	OP2PS	R/W	放大器正端偏置开关。 0: 关闭	0

描述位	名称	读写	描述	复位值
			1: 使能	
4	OP2OEN	R/W	OP2O输出到端口使能位。 0: 禁止; 1: 使能。	0
7: 5	OP2GS[2: 0]	R/W	OP2内部增益选择位。 同相单端输入模式: 000: 5x 001: 9x 010: 13x 011: 17x 100: 21x 101: 25x 110: 33x 111: 41x 差分输入模式: 000: 4x 001: 8x 010: 12x 011: 16x 100: 20x 101: 24x 110: 32x 111: 40x	000
8	OP2PRS	R/W	放大器正相输入电阻短路。	0
9	OP2NRS	R/W	放大器反相输入电阻短路。	0
14: 10	OP2OS[4: 0]	R/W	放大器失调电压修调位。 00000: OP2N-OP2P最大 10000: 不引入失调修调 11111: OP2P-OP2N最大	00000
15	OP2TRIMEN	R/W	修调使能信号。 0: 关闭TRIM功能 1: 开启TRIM功能	0
16	OP2TRIMO	R	修调指示信号。 TRIMEN=0, OP2TRIMO=0; TRIMEN=1, OP2TRIMO跳转为1, 代表修调完成。	0
17	OP2TRIM_IBIAS	R/W	运放偏置电流控制位。 0: 小电流模式 1: 大电流模式	0
31: 18			保留	

### 23.3.4. 模拟功能控制寄存器 (ANA\_CON: 0x4000E048)

描述位	名称	读写	描述	复位值
4: 0			保留	
6: 5	VRHS[1: 0]	R/W	见5.4.3章节	00
8: 7	VREFS[1: 0]	R/W	见5.4.3章节	0

描述位	名称	读写	描述	复位值
9	VTSEN	R/W	见5.5.2章节	0
10	AMUX2_S	R/W	AMUX2_OUT连接到PB15的YA1, 做模拟信号测试。 0: OP00 1: OP20	
11	VTS_SEL	R/W	见5.5.2章节	0
12	VBGBUF_EN	R/W	见5.4.3章节	0
13	VBG_VRH_S	R/W	见22.3.7章节	0
15: 14	AMUX_S[1: 0]	R/W	AMUX_OUT连接到PA15的YA1,做模拟信号测试。 00: VRH 01: OP00 10: OP10 11: DAC0	0

## 23.4.功能描述

### 23.4.1. 失调校准

通过 OPxTRIMEN 可以对 OPA 的输入失调电压进行校准, 使输入失调电压降到最低值。通过调整 OPxOS [4: 0]值大小, 读 OPxTRIMO 标志位, 直到 OPxTRIMO 由 0 变 1 表示校准结束, OPxOS [4: 0]保持设置值。

在配置成失调校准模式时, 运放的正负端接地。运放不能从引脚进入信号 OPxP 和 OPxN, 对应的 IO 口最好配置成数字端口。

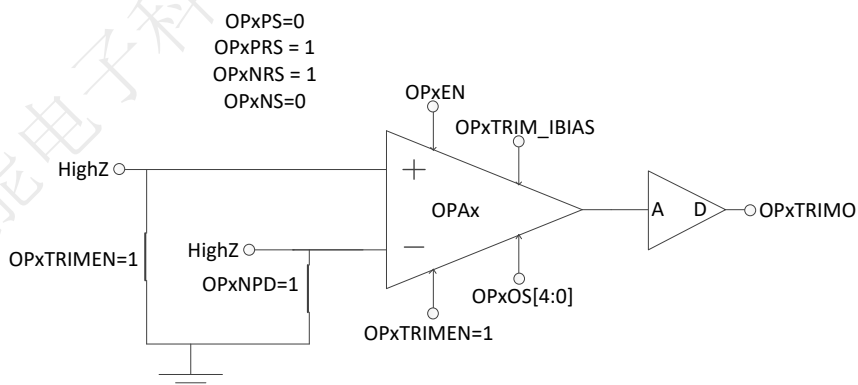


图 23-4-1 OPA 失调校准

## 24. 模数转换器(ADC)

12 位的模拟/数字转换器(ADC), 最大支持 16 个通道输入。基本特征为:

- 16 个输入选择
- 12bit 数据转换
- 可选择软件触发启动 ADC
- 可选择 PWM06 触发启动 ADC，可选择上升沿或者下降沿触发
- 可选择单次转换或者连续转换两种模式
- 支持单通道输入或者组转换，一次组转换最大包含 8 个不同的输入通道
- 支持双采样和流水采样

## 24.1. 功能框图

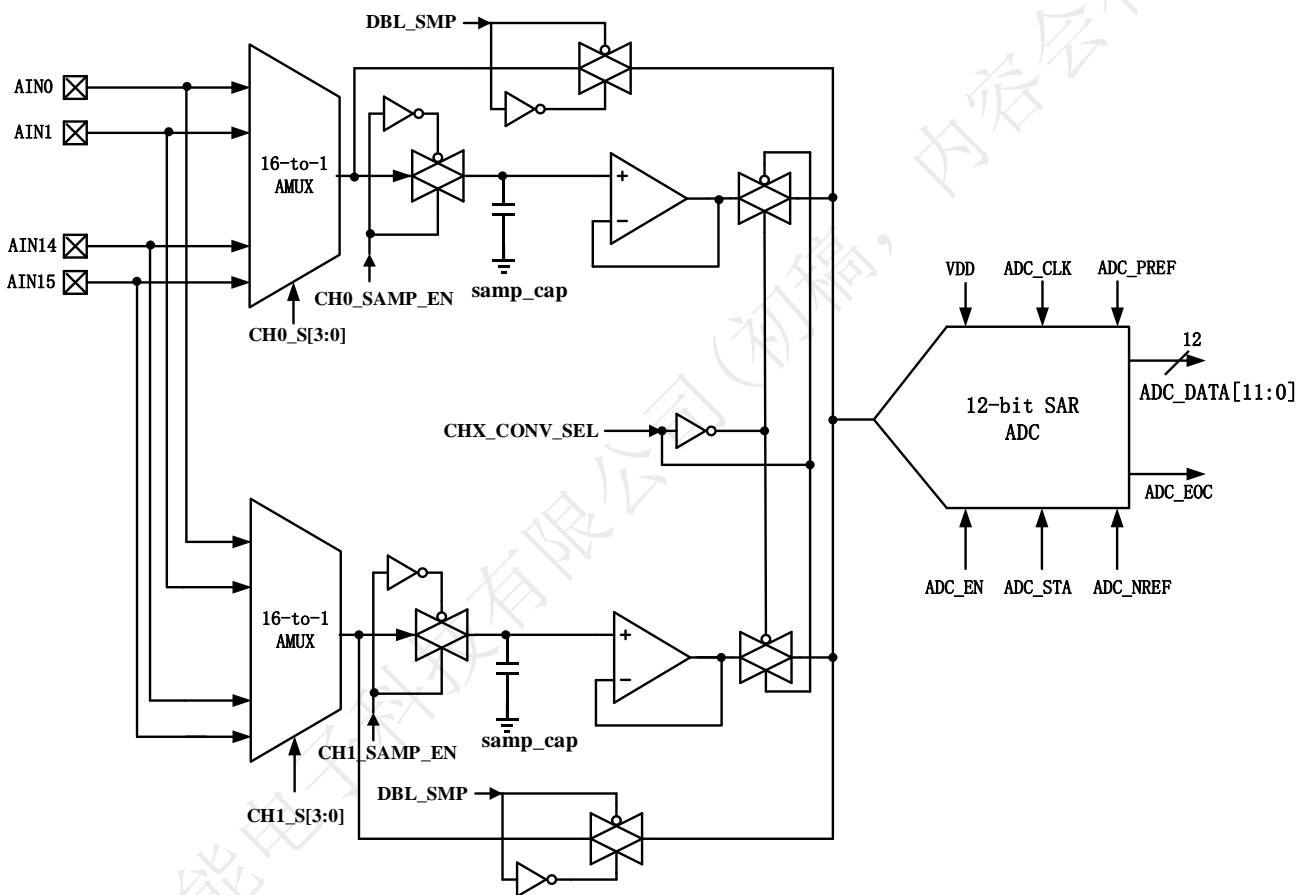


图 24-1 ADC 功能框图

## 24.2. 模块寄存器总表

寄存器列表如下:

寄存器名	地址	结构体(.h)	寄存器描述	复位值
ADCON	0x40005000	ADC	ADC控制寄存器	0x00000000

ADCHK	0x40005004	ADC	ADC自检控制寄存器	0x0000FF00
ADDLY	0x40005008	ADC	ADC延时控制寄存器	0x00000000
ADSEL	0x4000500C	ADC	ADC通道选择寄存器	0x00000000
ADCDATA0	0x40005010	ADC	ADC数据缓存寄存器0	0x0000
ADCDATA1	0x40005014	ADC	ADC数据缓存寄存器1	0x0000
ADCDATA2	0x40005018	ADC	ADC数据缓存寄存器2	0x0000
ADCDATA3	0x4000501C	ADC	ADC数据缓存寄存器3	0x0000
ADCDATA4	0x40005020	ADC	ADC数据缓存寄存器4	0x0000
ADCDATA5	0x40005024	ADC	ADC数据缓存寄存器5	0x0000
ADCDATA6	0x40005028	ADC	ADC数据缓存寄存器6	0x0000
ADCDATA7	0x4000502C	ADC	ADC数据缓存寄存器7	0x0000
IUDATA	0x40005030	ADC	Iu数据寄存器	0x0000
IVDATA	0x40005034	ADC	Iv数据寄存器	0x0000
IWDATA	0x40005038	ADC	Iw数据寄存器	0x0000
OFFSET	0x4000503C	ADC	offset数据寄存器	0x0000

## 24.3. 寄存器说明

### 24.3.1. ADC 控制寄存器(ADCON: 0x40005000)

描述位	名称	读写	描述	复位值
0	ADC_EN	R/W	ADC使能。 0: ADC关闭 1: ADC工作	0
2: 1	MODE[1: 0]	R/W	ADC模式选择位。 00: 单次单通道转换 01: 连续单通道转换 10: 单次组转换模式 11: 连续组转换模式	00
3	SSTA	R/W	既作为ADC软件启动位(写1有效), 硬件自动清零。 0: 不启动 1: 开启 又作为ADC转换状态位(只读)。 0: ADC空闲 1: ADC在转换中	0
6: 4	GM_N[2: 0]	R/W	ADC组转换控制位。 0~x通道进行组转换, x由GM_N[2: 0]确定(0~7)	000
7	EOC_FLAG	R/W	ADC转换标志, 写1清零。	0

描述位	名称	读写	描述	复位值
			0: ADC未转换结束 1: 单次单通道、连续单通道转换模式时单通道转换结束, 或者单次组转换、连续组转换模式时组转换结束	
9: 8	TRGS0[1: 0]	R/W	ADC启动触发信号沿选择。 00: 软件启动 01: PWM事件的上升沿启动 10: PWM事件的下降沿启动 11: PWM事件的上升沿或下降沿启动	00
12: 10		R/W	保留	
13	EOC_IEN	R/W	ADC转换结束标志中断使能。 0: 禁止 1: 使能	0
14	CHK_IEN	R/W	ADC自检监控中断使能。 0: 禁止 1: 使能	0
15	CHK_FLG	R/W	ADC自检溢出标志, 写1清零。 0: ADC自检ok 1: 自检超出阈值范围(大于高阈值, 或低于低阈值)	0
17: 16	SMP_MODE [1: 0]	R/W	ADC组转换, 双采样控制。 00: 单采样, AD转换 01/10: 异步双采样, 前2次采样由PWM Uh/Vh/Wh前2个下降沿触发, 支持单电阻采样应用。 11: 同步双采样	00
20: 18	CLK_CFG[2: 0]	R/W	ADC模拟电路时钟控制寄存器, 必须是50%占空比。ADC模拟电路的采样时间为11个时钟周期, 转换时间为13个时钟周期。 000: PCLK/2 001: PCLK/4 010: PCLK/8 011: PCLK/16 100: FCLK/2 101: FCLK/3 110: FCLK/4 111: 保留	000
23: 21	TRGS1[2: 0]	R/W	选择PWM事件触发ADC启动。	000

描述位	名称	读写	描述	复位值
			000: PWM_06 (由EPWM模块中的CCR3P上升计数匹配, CCR3N下降计数匹配事件触发) 001: PWM_FALT 010: PWM_UEV 011: PWM_UDF 100: PWM_OVF 其它: PWM_06	
31: 24			保留	

### 24.3.2. ADC 自检配置寄存器(ADCHK: 0x40005004)

描述位	名称	读写	描述	复位值
9: 0	CHK_REFL [9: 0]	R/W	ADC自检低阈值寄存器。	0x0000
19: 10	CHK_REFH [9: 0]	R/W	ADC自检高阈值寄存器。	0x3FFFF
22: 20	CHK_CHS[2: 0]	R/W	ADC自检通道选择寄存器 000: 通道0缓冲寄存器 001: 通道1缓冲寄存器 010: 通道2缓冲寄存器 011: 通道3缓冲寄存器 100: 通道4缓冲寄存器 101: 通道5缓冲寄存器 110: 通道6缓冲寄存器 111: 通道7缓冲寄存器	000
31: 23			保留	

### 24.3.3. ADC 延时配置寄存器(ADDLY: 0x40005008)

描述位	名称	读写	描述	复位值
7: 0	DLY1[7: 0]	R/W	ADC默认延迟时间寄存器, 用于组转换的0~1通道转换, 控制触发事件→开始采样的延时, 以PCLK时钟周期为单位。	0x00
15: 8	SDLY1 [7: 0]	R/W	ADC默认采样时间寄存器, 用于组转换的0~1通道转换, 控制采样脉冲的宽度。以PCLK时钟周期为单位。	0x00

描述位	名称	读写	描述	复位值
23: 16	DLY2 [7: 0]	R/W	ADC第2延迟时间寄存器, 仅用于组转换的2~x通道转换, 控制触发事件→开始采样的延时, 以PCLK时钟周期为单位。	0x00
31: 24	SDLY2 [7: 0]	R/W	ADC第2采样时间寄存器, 仅用于组转换的2~x通道采样, 控制采样脉冲的宽度. 以PCLK时钟周期为单位。	0x00

#### 24.3.4. ADC 通道控制寄存器(ADSEL: 0x4000500C)

描述位	名称	读写	描述	复位值
3: 0	CH0S	R/W	ADC通道0选择寄存器。 0000: PA0 0001: PA1 0010: PA2 0011: PA3 0100: PA4 0101: PA5 0110: PA6 0111: PA7 1000: PA8 1001: PA9 1010: PA10 1011: PA11/ VTS 1100: PA12/ PA15 1101: OP20 1110: OP00 1111: OP10	0000
7: 4	CH1S	R/W	ADC通道1选择寄存器, 同上	0000
11: 8	CH2S	R/W	ADC通道2选择寄存器, 同上	0000
15: 12	CH3S	R/W	ADC通道3选择寄存器, 同上	0000
19: 16	CH4S	R/W	ADC通道4选择寄存器, 同上	0000
23: 20	CH5S	R/W	ADC通道5选择寄存器, 同上	0000
27: 24	CH6S	R/W	ADC通道6选择寄存器, 同上	0000
31: 28	CH7S	R/W	ADC通道7选择寄存器, 同上	0000

### 24.3.5. ADC 数据缓存寄存器( ADCDATA0: 0x40005010)

描述位	名称	读写	描述	复位值
11: 0	ADCDATA0[11: 0]	R	ADC数据缓存寄存器0	0x000
15: 12			保留	

组转换模式下，ADCDATA0[11: 0] 对应 CH0S[3: 0]选择通道的转换结果。

### 24.3.6. ADC 数据缓存寄存器( ADCDATA1: 0x40005014)

描述位	名称	读写	描述	复位值
11: 0	ADCDATA1[11: 0]	R	ADC数据缓存寄存器1	0x000
15: 12			保留	

组转换模式下，ADCDATA1[11: 0] 对应 CH1S[3: 0]选择通道的转换结果。

### 24.3.7. ADC 数据缓存寄存器( ADCDATA2: 0x40005018)

描述位	名称	读写	描述	复位值
11: 0	ADCDATA2[11: 0]	R	ADC数据缓存寄存器2	0x000
15: 12			保留	

组转换模式下，ADCDATA2[11: 0] 对应 CH2S[3: 0]选择通道的转换结果。

### 24.3.8. ADC 数据缓存寄存器( ADCDATA3: 0x4000501C)

描述位	名称	读写	描述	复位值
11: 0	ADCDATA3[11: 0]	R	ADC数据缓存寄存器3	0x000
15: 12			保留	

组转换模式下，ADCDATA3[11: 0] 对应 CH3S[3: 0]选择通道的转换结果。

### 24.3.9. ADC 数据缓存寄存器( ADCDATA4: 0x40005020)

描述位	名称	读写	描述	复位值
11: 0	ADCDATA4[11: 0]	R	ADC数据缓存寄存器4	0x000
15: 12			保留	

组转换模式下，ADCDATA4[11: 0] 对应 CH4S[3: 0]选择通道的转换结果。

### 24.3.10. ADC 数据缓存寄存器( ADCDATA5: 0x40005024)

描述位	名称	读写	描述	复位值
11: 0	ADCDATA5[11: 0]	R	ADC数据缓存寄存器5	0x000
15: 12			保留	

组转换模式下，ADCDATA5[11: 0] 对应 CH5S[3: 0]选择通道的转换结果。

### 24.3.11. ADC 数据缓存寄存器( ADCDATA6: 0x40005028)

描述位	名称	读写	描述	复位值
11: 0	ADCDATA6[11: 0]	R	ADC数据缓存寄存器6	0x000
15: 12			保留	

组转换模式下, ADCDATA6[11: 0] 对应 CH6S[3: 0]选择通道的转换结果。

### 24.3.12. ADC 数据缓存寄存器( ADCDATA7: 0x4000502C)

描述位	名称	读写	描述	复位值
11: 0	ADCDATA7[11: 0]	R	ADC数据缓存寄存器7	0x000
15: 12			保留	

组转换模式下, ADCDATA7[11: 0] 对应 CH7S[3: 0]选择通道的转换结果。

### 24.3.13. Iu 数据寄存器( IUDATA: 0x40005030)

描述位	名称	读写	描述	复位值
11: 0	IUDATA[11: 0]	R	12位Iu数据	0x000
15: 12			保留	

### 24.3.14. Iv 数据寄存器( IVDATA: 0x40005034)

描述位	名称	读写	描述	复位值
11: 0	IVDATA[11: 0]	R	12位Iv数据	0x000
15: 12			保留	

### 24.3.15. Iw 数据寄存器(IWDATA: 0x40005038)

描述位	名称	读写	描述	复位值
11: 0	IWDATA [11: 0]	R	12位Iw数据	0x000
15: 12			保留	

### 24.3.16. OFFSET 数据寄存器(OFFSET: 0x4000503C)

描述位	名称	读写	描述	复位值
11:0	OFFSET [11:0]	R/W	12位OFFSET数据	0x000
15: 12			保留	

## 24.4.功能描述

### 24.4.1. 单次单通道转换模式

当 ADC 工作在单次单通道模式时，只要 ADC 触发启动，ADC 只转换一次就结束转换，等待下一次触发。在该模式下，ADC 的输入通道只能通过 ADSEL 寄存器的低四位 CH0S[3:0]选择。触发源选择软件触发或 PWM 事件，配置如下所述：

- 配置 ADSEL 寄存器低四位 CH0S[3:0]选择模拟输入通道。
- 配置 ADCON 中的 ADC\_EN 位为 1
- 配置 ADCON 中的 ADC\_MODE[1:0]为 00
- 配置 ADCON 中的 SMP\_MODE[1:0]为 00，单采样模式(bypass S/H 和 buffer 电路)。
- 配置 ADDLY 寄存器，T<sub>dly</sub> 时间由 AD\_DLY1[7:0]设置，T<sub>soc</sub> 时间由 AD\_SDLY1 [7:0] 设置。
- 配置 ADCON 中的 AD\_SFT 位为 1 或者设置为 PWM 事件触发，启动 ADC 转换过程。
- 等待 ADDLY 的延迟后开始实际转换过程
- 读 ADCON 中的 ADC\_BUSY 位是否为 0，或者 EOC\_FLAG 位是否为 1，或者判断是否进入 EOC 中断。

单次单通道转换流程如下：使能 ADC 和 PWM，当 PWM 信号输出时 ADC 触发信号(图中 trg\_edge)置 1，在经过 T<sub>dly1</sub> 时间后 ADC\_STA 置 1，ADC 开始对 CH0 采样，采样时间为 T<sub>soc1</sub>，然后对 CH0S 通道电压进行 AD 转换，一次转换完成产生 EOC\_FLAG，如果需要再次采集信号，则需要再次产生 PWM 事件触发启动 ADC 转换。

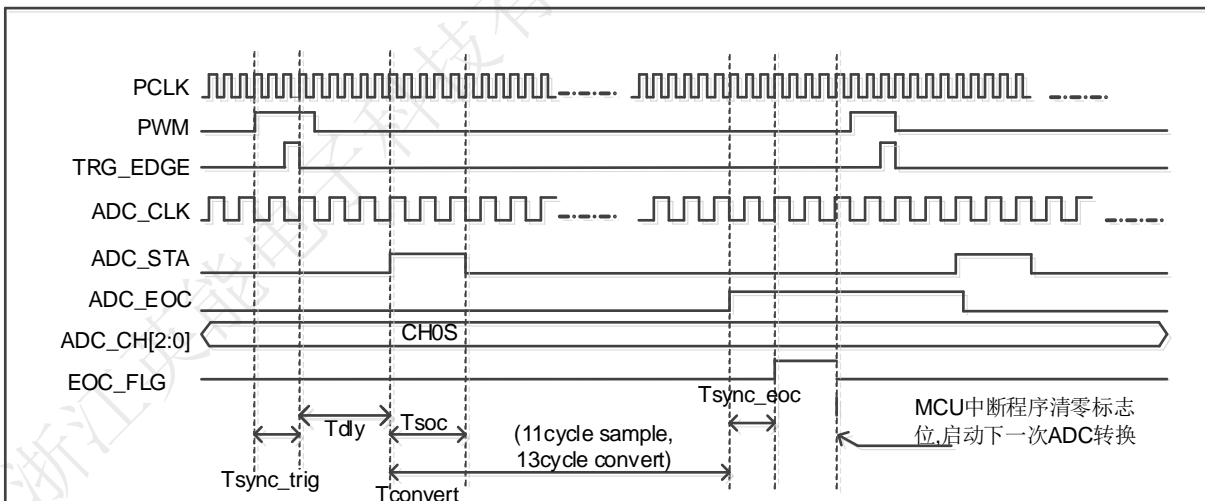


图 24-4-1 ADC 单通道转换时序

### 24.4.2. 连续单通道转换模式

当 ADC 工作在连续单通道转换模式时，ADC\_MODE[1: 0] 设定为 01，只要 ADC 触发启动，ADC 将会连续转换，每次转换结束均会产生转换结束中断。ADC 的输入通道只能通过 ADSEL 寄存器的低四位 CH0S[3: 0] 选择。停止 ADC 连续单通道转换模式的方法有两种，一种是直接使 ADCON 中的 ADC\_EN 位置 0，一种是将 ADCON 中的模式重新切换成单次转换模式。

如果采用 ADCON 中 ADC\_EN 位置 0 的模式，ADC 转换立即结束。如果采用重新切换成单次转换模式的方式，那么 ADC 在最后一次转换结束后停止转换。

### 24.4.3. 单次组转换模式

当 ADC 工作在单次组转换模式时，只要 ADC 触发启动，ADC 将会根据 ADSEL 中由 CH0S[3: 0]、CH1S[3: 0]、CH2S[3: 0]、CH3S[3: 0]、CH4S[3: 0]、CH5S[3: 0]、CH6S[3: 0]、CH7S[3: 0] 选定的 8 个通道依次连续转换。转换后的结果将依次存储在缓冲寄存器 0/1/2/3/4/5/6/7 中，用户可以直接从对应 ADC 数据缓存寄存器 0~7 读取转换结果寄存器值。

ADC 工作在单次组转换模式时，如果设定 GM\_N=3，则只有当 4 个通道(CH0S[3: 0]、CH1S[3: 0]、CH2S[3: 0]、CH3S[3: 0]选定的)完全转换完成后才产生组转换结束中断标志。配置如下：

- 配置 ADSEL 寄存器 CH0S[3: 0]、CH1S[3: 0]、CH2S[3: 0]、CH3S[3: 0] 选择 4 个模拟输入通道。
- 配置 ADCON 中的 ADC\_EN 位为 1
- 配置 ADCON 中的 ADC\_MODE[1: 0] 为 10
- 配置 ADCON 中的 GM\_N[2: 0] 为 011
- 配置 ADCON 中的 SMP\_MODE[1: 0] 为 00，单采样模式(bypass S/H 和 buffer 电路)。
- 配置 ADDLY 寄存器，第 0、1 通道的 Tdly1 时间由 AD\_DLY1[7: 0] 设置，Tsoc1 时间由 AD\_SDLY1 [7: 0] 设置；后续通道的 Tdly2 时间由 AD\_DLY2[7: 0] 设置，Tsoc2 时间由 AD\_SDLY2 [7: 0] 设置
- 配置 ADCON 中的 AD\_SFT 位为 1 或者设置为 PWM 事件触发，启动 ADC 转换过程。
- 等待 ADDLY 的延迟后开始实际转换过程
- 读 ADCON 中的 ADC\_BUSY 位是否为 0，或者 EOC\_FLAG 位是否为 1，或者判断是否进入 EOC 中断。
- 第 0、1 通道转换完成后，会产生一个中断，用于实现电流采样数据的快速处理。

单次组转换流程如下：使能 ADC 和 PWM，当 PWM 信号输出时 ADC 触发信号(图中 trg\_edge)置 1，在经过 Tdly1 时间后 ADC\_STA 置 1，ADC 开始对 CH0 采样，采样时间为 Tsoc1，然后对 CH0S 通道电压进行 AD 转换，在转换完 CH0S 通道后接着对 CH1S 通道进行采样转换和转换，依次将 CH2S、CH3S 的数据转换完成产生 EOC\_FLAG。

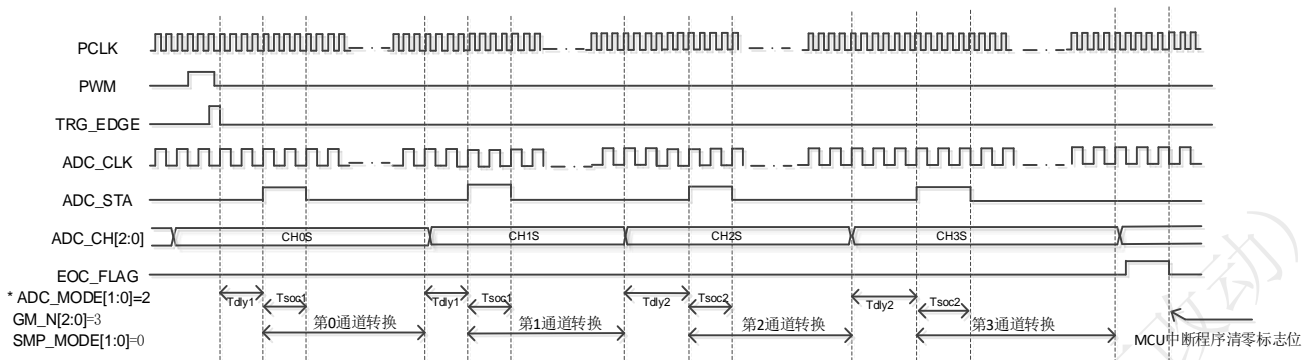


图 24-4-3 单次组转换时序

#### 24.4.4. 连续组转换模式

当 ADC 工作在连续组转换模式时，ADC\_MODE[1: 0]设定为 11，只要 ADC 触发启动，ADC 将会连续组转换，每次组转换结束均会产生组转换结束中断。停止 ADC 连续组转换模式的方法有两种，一种是直接使 ADCON 中的 ADC\_EN 位置 0，一种是将 ADCON 中的模式重新切换成单次转换模式。

如果采用 ADCON 中 ADC\_EN 位置 0 的模式，ADC 转换立即结束。如果采用重新切换成单次转换模式的方式，那么 ADC 在最后一次转换结束后停止转换。

#### 24.4.5. ADC 启动

ADC 的启动可以采用软件触发启动、PWM 事件触发启动，可以通过 ADCON 中的 ADTRG[2: 0]位来选择。可以通过 ADDLY 寄存器来配置启动延迟时间，启动延迟时间可以通过  $T_{SCLK} \times (ADDLY + 1)$  来计算。

#### 24.4.6. ADC 自检

ADC 支持自检功能，当相应通道 ADC 的转换数值高 8 位高于 ADC 自检高阈值 ADREFH 值时，或低于低阈值 ADREFL 值时，可触发中断。ADCHK 寄存器的 CHK\_CHS[2: 0]选择自检监控通道。

举例说明：如将 CHK\_CHS[2: 0]设置为 010，则 ADC 工作在连续组转换模式下，采用通道 2 作为自检监控通道，当通道 2 缓冲值对应的 ADC 转换值高 8 位高于 ADCREFH 值时产生中断。

#### 24.4.7. ADC 中断

ADC 中断包括转换结束中断和自检溢出中断，可以通过 ADCON 寄存器中的 EOC\_IEN 和 CHK\_IEN 位使能。

ADC 转换结束中断在单通道模式和组转换模式有所不同，单通道模式下每次转换结束都会产生一次中断，组转换模式时只有组转换结束后才产生中断。ADC 中断的清除可以通过往 ADCON 寄存器相应的中断标志位写 1 清除。

### 24.4.8. ADC 同步双采样

ADC 双采样仅在组转换时有效，且仅仅作用在前 2 次采样转换，SAMP\_MODE[1:0]=3 时，ADC 双采样功能使能。

ADC 双采样功能可同时存储 CH0 和 CH1 信号的电压值，在完成对 CH0 的 AD 转换后再进行 CH1 通道的 AD 转换，以此保证在电机运行是电流采样可以在同一时刻采集不同的信号。

具体流程如下图所示，将 ADCCON 寄存器设置为 ADC\_MODE[1:0]=3，SMP\_MODE[1:0]=3，GM\_N[2:0]=3，ADSEL 的 CH0S = 8，CH1S=7，CH2S=6，CH3S=5，并使能 ADC 和 PWM，当 PWM 信号输出时 ADC 触发信号(图中 trg\_edge)置 1，在经过 Tdly1 后 ADC\_STA 置 1，ADC 开始对 CH0S 和 CH1S 采样，采样时间为 Tsoc1，然后对第 0 通道电压进行 AD 转换，在转换完 CH0S 通道后接着对 CH1S 通道进行转换，在完成通道 0 和 1 的双采样后，再根据 Tdly2，和 Tsoc2 的时间对 CH2S 和 CH3S 通道依次进行采样和转换。下图中 Tdly1 时间由 ADDLY 的 AD\_DLY1[7:0]寄存器配置，Tdly2 时间由 ADDLY 的 AD\_DLY2[7:0]寄存器配置，Tsoc1 的时间由 ADDLY 的 AD\_SDLY1 [7:0]寄存器配置，Tsoc2 的时间由 ADDLY 的 AD\_SDLY2[7:0]寄存器配置。

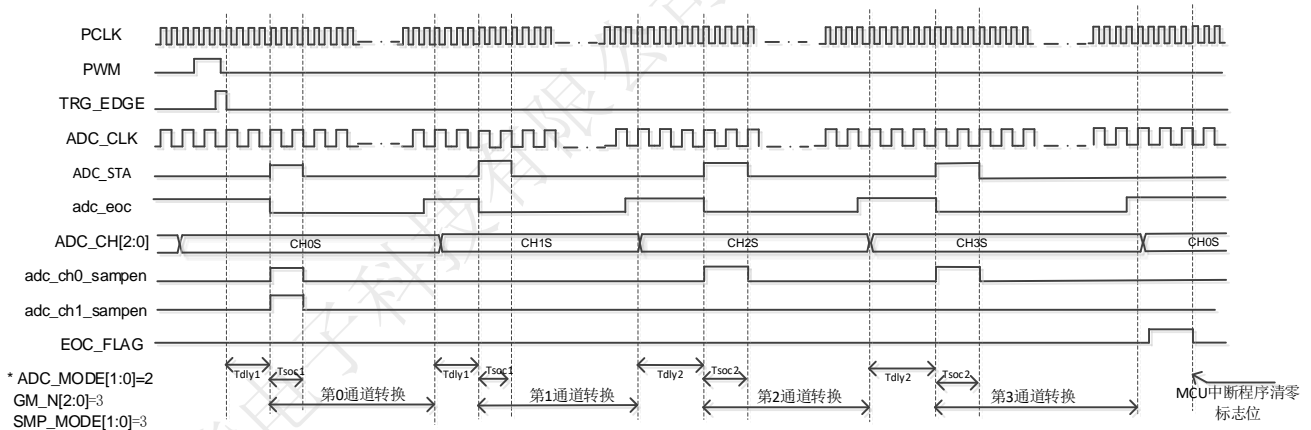


图 24-4-8 同步双采样时序

注意：adc\_ch0\_sampen 和 adc\_ch1\_sampen，adc\_eoc 为 AD 模块内部信号。adc\_ch0\_sampen 为双采样时 0 通道采样信号，adc\_ch1\_sampen 为双采样时 1 通道采样信号。

### 24.4.9. ADC 异步双采样(单电阻采样)

仅在组转换时有效，SMP\_MODE 为 01/10 时，触发组转换的异步双采样功能，用于单电阻采样的电机方案。

在异步触发组转换模式下，组转换的第 1 次、第 2 次采样是由 PWM UH/VH/WH 第 1 次、第 2 次下降沿触发，其它采样则由 ADC 控制器自动依次触发。

ADC 异步双采样功能必须与 EPWM 模块的 PWMTRGC 寄存器的 Samp\_mode=1 一起使用。且电流采样结果会存入 ADC 的 lu lv lw 数据寄存器。lu lv lw 寄存器数据会由 ADC 模块自动计算得到，需要用户对 offset 数据寄存器写入电流采样时运放的偏置电压值，lu lv lw 寄存器数据可被直接用于 FOC 运算。

- 配置 PWMTRGC 的 Samp\_mode = 1 为单电阻模式。
- 配置 PWMTRGC 的 Tadc[7:0]电流采样所需的时间。
- 配置 ADSEL 寄存器 CH0S[3:0]、CH1S[3:0]、CH2S[3:0]、CH3S[3:0]选择 4 个模拟输入通道。其中 CH0S[3:0]、CH1S[3:0]需要写入单电阻电流采样通道。
- 配置 ADCON 中的 ADC\_MODE[1:0]为 2。
- 配置 ADCON 中的 GM\_N[2:0] 为 3。
- 配置 ADCON 中的 SMP\_MODE[1:0]为 1，异步双采样模式。
- 配置 ADDLY 寄存器，第一次电流采样的 Tdly1 时间由 AD\_DLY1[7:0]设置(运放输出电流稳定时间)，Tsoc1 时间由 AD\_SDLY1 [7:0]设置(ADC 采样时间)；第二次电流采样的 Tdly1 时间由 AD\_DLY1[7:0]设置(运放输出电流稳定时间)，Tsoc1 时间由 AD\_SDLY1 [7:0]设置(ADC 采样时间)。
- 配置 ADCON 中的 ADC\_EN 位为 1。

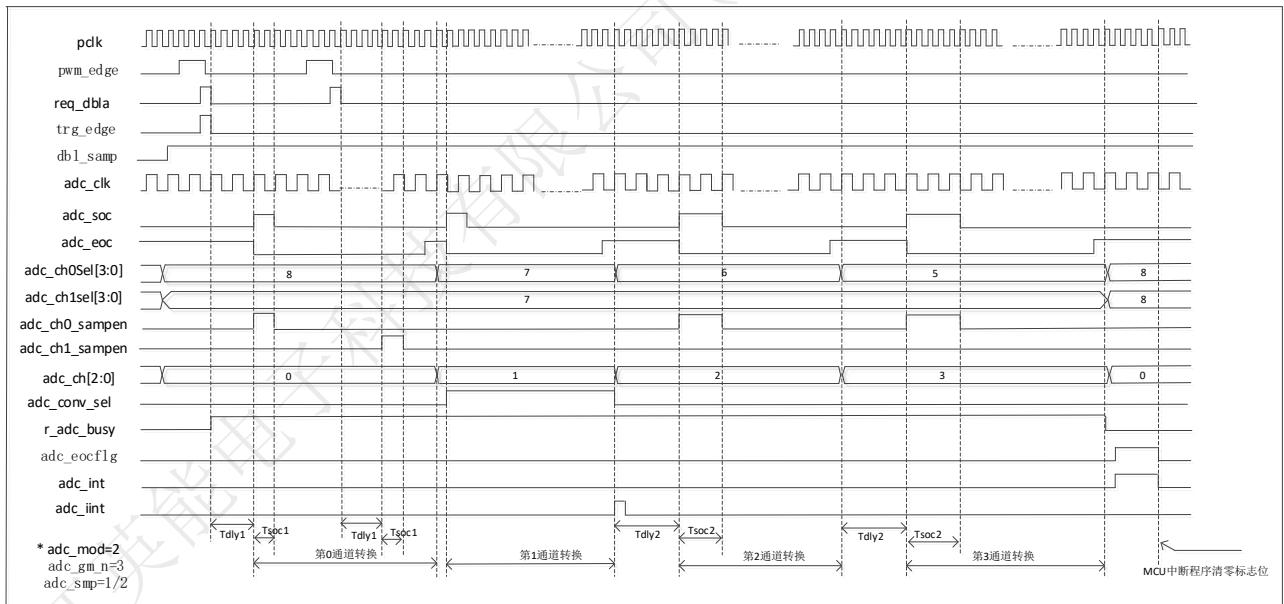


图 24-4-9 异步双采样时序

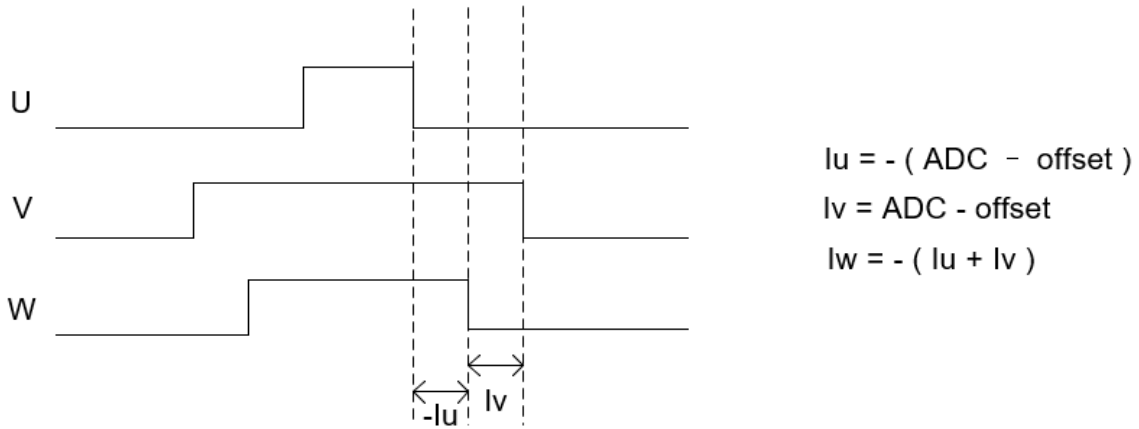


图 24-4-9-1 驱动信号波形和采样电流示例

## 25. 数模转换器(DAC)

DAC 模块具有以下特点:

- 输入参考电压可选 1.2V, 2.5V, 4V, VDD
- 精度为 10 位, 1023 档可调电压
- DAC 输出, 输出电压范围 0 到 参考电压。

### 25.1.功能框图

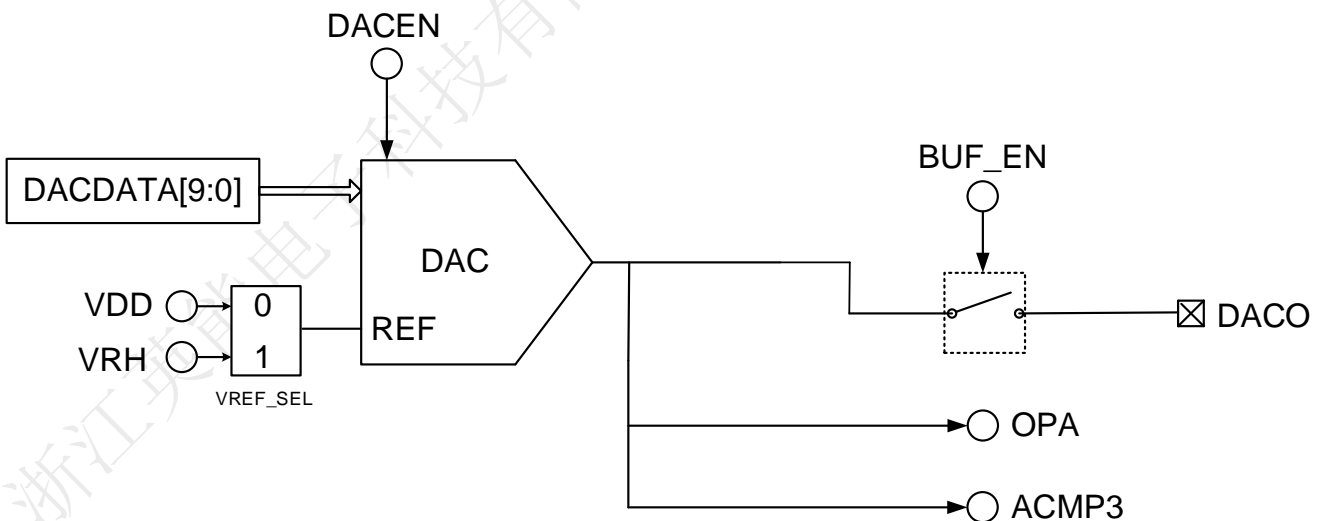


图 25-1 DAC 功能框图

### 25.2.模块寄存器总表

寄存器名	地址	结构体(.h)	寄存器描述	复位值
------	----	---------	-------	-----

DACDATA	0x40007800	DAC	DAC数据寄存器	0x0000
DACCON	0x40007804	DAC	DAC控制寄存器	0x00

## 25.3. 寄存器说明

### 25.3.1. DAC 数据寄存器(DACDATA: 0x40007800)

描述位	名称	读写	描述	复位值
9: 0	DACDATA [9: 0]	R/W	DAC数据寄存器	0x000
15: 10			保留	

### 25.3.2. DAC 控制寄存器(DACCON: 0x40007804)

描述位	名称	读写	描述	复位值
0	DACEN	R/W	DAC模块使能配置，不使能时，DACO 输出悬空。 0: 不使能 1: 使能	0
1			保留	
2	BUF_EN	R/W	DAC管脚输出使能配置。 0: DACO输出不使能，输出悬空 1: DACO输出使能	0
3	VREF_SEL	R/W	DAC参考电压选择。 0: VDD 1: VRH	0
7: 4			保留	

## 25.4. DAC 功能描述

### 25.4.1. DAC 模块使能

将寄存器 DACCON 的标志 DACEN 置 1，即可打开对 DAC 模块的供电。经过一段启动时间后，DAC 模块即被使能。

### 25.4.2. DAC 输出电压计算

当 DAC 模块被开启后，如果向寄存器 DACDATA 写入 10 位数据，硬件自动触发 DAC 转换。

DAC模拟输出电压满足以下关系:

$$VDAC00 = VREF * (DACDATA+1) / 1024$$

### 25.4.3. DAC 输出与其他模块互联关系

DAC 输出 DAC00 可以直接输出或通过缓冲(Buffer)输出到外部管脚 PA15;  
也可以输出到 ACMP3 的输入脚 CMP3P;  
也可以输出到运放作为运放输入的偏置电压 OPAVBIA。

## 26. 存储器(FLASH)

FLASH 特点:

- 32k 字节 Flash 主存储区, 总共 64 个 Sector, 地址范围: 0x0000-0x7FFF
- 2K 字节 Flash 信息区, 总共 4 个 Sector, 地址范围: 0x8000-0x87FF
- 带预取缓冲器的读接口
- 字节编程, 页擦除, 片擦除操作
- 读保护
- 写保护

### 26.1. 模块寄存器总表

寄存器名	地址	结构体 (.h)	寄存器描述	复位值
SYSCON	0x4000E000	SYSCFG	系统控制寄存器, 设置Flash编程时钟	0x0307136C
FSADDR	0x40012000	FLASH	Flash编程地址寄存器	0x0000
FSDATA	0x40012004	FLASH	Flash数据寄存器	0x00000000
FSCON	0x40012008	FLASH	Flash编程控制寄存器	0x0000
FSSTA	0x4001200C	FLASH	Flash状态寄存器	0x0000
FSMAINKEY:	0x40012014	FLASH	Flash主存储器安全锁寄存器	0x00000000
FSINFOKEY	0x40012018	FLASH	Flash信息区安全锁寄存器	0x00000000
FSACR	0x4001201C	FLASH	Flash访问控制寄存器	0x00
FSIAP	0x40012020	FLASH	IAP保护配置寄存器	0x00000000
FSWRP	0x40012024	FLASH	写保护配置寄存器	0x00000000

## 26.2. 寄存器说明

### 26.2.1. 系统控制寄存器 (SYSCON: 0x4000E000)

描述位	名称	读写	描述	复位值
0	SYSCCLKS	R/W	见5.2.2章节	
1	CSSSEN	R/W		
4: 2	FCLK_CFG[2:0]	R/W		
7: 5	PCLK_CFG[2:0]	R/W		
10: 8			保留	
11	STOP_CFG	R/W	见5.3.1章节	
12	WKEN_EINT0	R/W		
13	WKEN_EINT1	R/W		
14	WKEN_EINT2	R/W		
15	WKEN_EINT3	R/W		
16	WKEN_LVD	R/W		
17	WKEN_RT	R/W	见11.1章节	
18	RTEN	R/W		
21: 19	RT_CFG[2: 0]	R/W		
23: 22	PGCLK_CFG [1: 0]	R/W	PGCLK_CFG[1: 0]: EFLASH 编程时钟频率控制。 00: SYSCLK 4 分频, SYSCLK $\approx$ 16M 01: SYSCLK 12 分频, SYSCLK $\approx$ 48M 10: SYSCLK 16 分频, SYSCLK $\approx$ 64M 11: SYSCLK 20 分频, SYSCLK $\approx$ 80M (建议在对Flash进行编程时, SYSCLK选择内置RCH)	00
24	PGCLKEN	R/W	EFLASH编程时钟开关控制, 缺省1。 0: 关闭 1: 开启	1
25	DEBUG_MODE	R/W	见28.1章节	1
31: 26			保留	

### 26.2.2. FLASH 编程地址寄存器 (FSADDR: 0x40012000)

描述位	名称	读写	描述	复位值
0~15	FSADDR[15: 0]	RW	Flash编程地址寄存器。 当进行Byte/Word编程时, 为Flash的Byte/Word地址。在Word编程时, 地址低2位必须为0。	0x0000

描述位	名称	读写	描述	复位值
			当进行Sector擦除时，为Flash的Sector地址，低9位无效。每个Sector为512字节。	

### 26.2.3. FLASH 数据寄存器(FSDATA: 0x40012004)

描述位	名称	读写	描述	复位值
0~31	FSDATA[31: 0]	RW	Flash字节编程时的数据寄存器。 Word编程时，FSDATA写入与FSADDR匹配的32位数据 Byte编程时： FSADDR[1:0]=00，则数据写入FSDATA[7:0] FSADDR[1:0]=01，则数据写入FSDATA[15:8] FSADDR[1:0]=10，则数据写入FSDATA[23:16] FSADDR[1:0]=11，则数据写入FSDATA[31:24]	0x00000000

### 26.2.4. FLASH 编程控制寄存器(FSCON: 0x40012008)

描述位	名称	读写	描述	复位值
0~2	PROGEN [2: 0]	RW	编程/擦除操作使能位。 101: 允许启动编程/擦除操作 其它值: 禁止启动编程/擦除操作	000
3	WRST	RW	Byte/Word 编程启动控制位。 0: 无效 1: 启动Byte/Word编程	0
4	PERST	RW	Sector 擦除启动控制位。 0: 无效 1: 启动Sector擦除动作	0
5	MERST	RW	片擦除启动控制位。 0: 无效 1: 启动片擦除	0
6	OPSEL	RW	Flash 存储区选择位。 0: 主存储区 1: 信息区	0
7	WRMODE	RW	编程模式选择位。 0: Byte 编程模式 1: Word(32位)编程模式	0
8	MERMODE	RW	片擦除模式选择位。 0: 无效	0

描述位	名称	读写	描述	复位值
			1: 擦除整个FLASH主存储区和信息区的Sector0~2	
9	STB	RW	待机模式控制位。 0: 无效 1: 表示Flash进入待机状态(CEb=1);	0
10	DPSTB	RW	深度待机模式控制位。 0: 无效 1: 表示Flash进入深度待机状态(DPSTB=1);	0
11~14			保留	
15	CPU_EN	RW	在编程/擦除过程中 CPU 控制位。 0: CPU 不停止 1: CPU停止, 编程/擦除结束后继续运行	0

### 26.2.5. FLASH 状态寄存器 (FSSTA: 0x4001200C)

描述位	名称	读写	描述	复位值
0	BSY	R	Flash忙标志位, 只读, 硬件置1或清零。 0: Flash处在空闲状态 1: Flash处在忙状态,	0
1	IAPSEQERR	R	IAP操作序列错误标志位, 只读, 硬件置1, 系统复位清零。 0: 1: 发生操作时序错误, 置1	0
2	IAPLOCKSTA	R	安全开关状态, 只读, 硬件置1, 写开关序列清零。 0: 打开状态, 允许进行IAP操作 1: 关闭状态	1
3	INFOSEQERR	R	信息区安全开关序列错误标志位, 只读, 硬件置1, 系统复位清零。 0: 1: 发生操作时序错误, 置1	0
4	INFOLOCKSTA	R	信息区安全开关状态, 只读, 硬件置1, 写信息区开关序列清零。 0: 信息区安全开关打开, 允许进行IAP操作 1: 信息区安全开关关闭状态	1
5	MAINSEQERR	R	主存储区安全开关序列错误标志位, 只读, 硬件置1, 系统复位清零。 0: 1: 发生操作时序错误, 置1	0

描述位	名称	读写	描述	复位值
6	MAINLOCKSTA	R	主存储区安全开关状态，只读，硬件置1，写主存储区开关序列清零。 0: 主存储区安全开关打开，允许进行IAP操作 1: 主存储区安全开关关闭状态	1
7	WPERR	R	写保护错误标志位，只读，硬件置1或清零。 0: 写1清零 1: 对写保护页进行写或擦除，操作无效，置1	0
8			保留	
9	IAPFLG	R	结束操作标志位，只读，硬件置1或清零。 0: Flash IAP操作开始时，清0 1: Flash IAP操作结束时，置1	1
15: 10			保留	

### 26.2.6. FLASH 主存储区安全锁寄存器(FSMINKEY: 0x40012014)

描述位	名称	读写	描述	复位值
31~0	FSMAINKEY [31: 0]	W	往FSMAINKEY寄存器中连续写 0x0123CDEF ,0x89AB4567 安全开关打开； 往FSMAINKEY寄存器中连续写 0x4567CDEF ,0x89AB0123 安全开关关闭； 往FSMAINKEY寄存器0x0123CDEF， 0x4567CDEF 以外的数据，安全开关锁定关闭只能 复位恢复。	0x00000000

### 26.2.7. FLASH 信息区安全锁寄存器(FSINFOKEY: 0x40012018)

描述位	名称	读写	描述	复位值
31~0	FSINFOKEY [31: 0]	W	往FSINFOKEY寄存器中连续写 0x0123CDEF ,0x89AB4567 安全开关打开； 往FSINFOKEY寄存器中连续写 0x4567CDEF ,0x89AB0123 安全开关关闭； 往FSINFOKEY寄存器0x0123CDEF， 0x4567CDEF 以外的数据，安全开关锁定关闭，只能复位恢复。	0x00000000

### 26.2.8. FLASH 访问控制寄存器(FSACR: 0x4001201C)

描述位	名称	读写	描述	复位值
1:0	LATENCY[1:0]	RW	Flash 读操作等待周期。 00: 读数据无延迟 (FCLK≤32Mhz) 01: 读数据延迟 1 个总线周期 (32Mhz<FCLK≤64Mhz) 10: 读数据延迟 2 个总线周期 (64Mhz<FCLK≤96Mhz) 11: 读数据延迟 3 个总线周期 (64Mhz<FCLK≤96Mhz) (在LATENCY[1:0]从大往小调时, 需要确保FCLK满足要求, 否则会带来风险。)	00
2	PRFETCH_EN	RW	指令预取使能控制。 0: 预取禁止 1: 预取使能	0
3	PRFETCH_MOD	RW	指令预取提前判断控制。 0: 预取提前判断关闭 1: 预取提前判断使能	0
5:4	LPCON[1:0]	RW	读低功耗控制位。 00: 读低功耗等级0, OEb关闭2个系统时钟周期, AHB总线时钟4分频 01: 读低功耗等级1, OEb关闭6个系统时钟周期, AHB总线时钟8分频 10: 读低功耗等级2, OEb关闭10个系统时钟周期, AHB总线时钟12分频 11: 读低功耗等级3, OEb关闭14个系统时钟周期, AHB总线时钟16分频	00
6	LPEN	RW	低功耗读使能。 0: 低功耗读关闭, CEB、OEb一直开启 1: 低功耗读开启	0
7			保留	

### 26.2.9. IAP 保护配置寄存器(FSIAP: 0x40012020)

描述位	名称	读写	描述	复位值
0~7	RDP[7:0]	R	读保护设置字节。 当RDP设置为0xA5时读保护有效, 其他值芯片处于不保护状态。	0

描述位	名称	读写	描述	复位值
			读保护有效后，只有通过全片擦除命令擦除读写保护配置区数据，然后复位重启后才能解除保护。	
15~8	IAPKEY[7: 0]	R	IAP保护设置字节。 当IAPKEY设置为0x69时IAP保护有效，其他值芯片处于不保护状态。 写保护有效后，只有通过全片擦除命令擦除读写保护配置区数据，然后复位重启后才能解除保护。	0x00
16~31	保留			0x0000

### 26.2.10. 写保护配置寄存器 (FSWRP: 0x40012024)

描述位	名称	读写	描述	复位值
0~31	FSWRP[31: 0]	R	写保护位，对应编号 Block(0~31)的写保护配置位，一个 Block 包含 2 个 Sector。 0: 写保护有效 1: 不写保护 只有通过片擦除命令擦除读写保护配置区数据，然后复位重启后才能解除Block写保护。	0x00000000

## 26.3. 读操作

### 26.3.1. 指令预取

内核与 Flash 之间有指令预取缓冲区，预取缓冲区的工作使得更快速的 CPU 执行成为可能，因为 CPU 取一个字指令的同时下一个字的指令内容页已经在预取缓冲区中。指令预取功能具有使能/禁止两种设置。

### 26.3.2. Flash 读操作等待

为了保证 Flash 的正确读取，必须在 Flash 访问控制寄存器中的 LATENCY[1: 0] 中指定预取指控制器的速度比，这个数值等于每次访问 Flash 后到下次访问之间所需插入的等待周期的个数。复位后，这个值默认为零，也就是没有插入等待周期的状态。为了在系统时钟超过访问 Flash 的最高速度时，将访问 Flash 的速度降下来，保证正确读取数据。

在 LATENCY[1: 0]从大往小调时，需要确保 FCLK 满足要求，否则会带来风险。先把 FCLK 切换到目标档位，然后对应 FCLK 频率选择 LATENCY[1: 0]。

### 26.3.3. Flash 读低功耗控制

当系统时钟处于低频运行模式时，低于 Flash 最高访问速度的一半以下。在低频运行时，通过读低功耗控制来减少访问 Flash 的时间，也就是减少了 Flash 工作时间，从而降低了芯片整体运行功耗。

通过控制 Flash 的 OEB 来实现。

## 26.4. Flash 编程和擦除操作

芯片 Flash 存储器支持在系统编程和在应用编程。

在系统编程是指使用 SWD 或 BootLoader 的方法在板下载程序到 Flash 中，这样免去使用芯片烧录座进行烧录的问题，简化了生产流程。

在应用编程(IAP)是指允许用户在运行程序的过程中改写 Flash 数据，比如记录一个掉电保存的数据，或通过一段已经烧录好的引导程序对另外一块 Flash 空间进行程序升级。支持下面的命令：

- 1) FLASH 主存储区 Byte/Word 编程
- 2) FLASH 主存储区 Sector 擦除
- 3) FLASH 信息区 Byte/Word 编程
- 4) FLASH 信息区 Sector 擦除

对 Flash 进行编程/擦除操作时，必须要保证工作电压稳定。由于 Flash 编程/擦除操作需要使用内部 RCH，所以编程擦除时 RCH 必须处于开启状态下。

CPU 运行在 FLASH 区域，对 Flash 进行编程/擦除操作时，CPU 必须暂停运行，通过关闭 CPU 时钟来实现。

FLASH 编程擦除操作需要以下 7 个寄存器配合操作完成。

寄存器地址	寄存器名称	寄存器功能
0x40010000	FSADDR	FLASH编程地址寄存器
0x40010004	FSDATA	FLASH数据输入寄存器
0x40010008	FSCON	FLASH编程控制寄存器
0x4001000C	FSSTA	FLASH状态寄存器
0x40010014	FMAINKEY	主存储区安全锁寄存器
0x40010018	FSINFOKEY	信息区安全锁寄存器
0x4001001C	FSACR	FLASH访问控制寄存器
0x40010020	FSIAP	IAP保护设置寄存器
0x40010024	FSWRP	IAP写保护设置寄存器

#### 26.4.1. 安全开关打开/关闭

由于 FLASH 数据的正确性非常重要，不能被意外改写，所以需要有一个安全开关。当安全开关打开时可以进行 FLASH 编程擦除操作，当安全开关闭合时禁止 FLASH 编程擦除操作。

由于 FLASH 主存储区和信息区是分开的，且在应用时信息区数据不能被更改的。所以安全开关分为主存储区安全开关和信息区安全开关，分别由 FSMAINKEY 和 FSINFOKEY 两个寄存器控制。

- **主存储区安全开关打开/关闭:**

往 FSMAINKEY 寄存器中连续写 0x0123CDEF , 0x89AB4567 安全开关打开

往 FSMAINKEY 寄存器中连续写 0x4567CDEF , 0x89AB0123 安全开关关闭

信息区安全开关打开/关闭:

往 FSINFOKEY 寄存器中连续写 0x0123CDEF , 0x89AB4567 安全开关打开

往 FSINFOKEY 寄存器中连续写 0x4567CDEF , 0x89AB0123 安全开关关闭

当安全开关打开后，FSSTA 寄存器的 IAPLOCKSTA 位清 0。当安全开关闭合后，FshSta 寄存器的 IAPLOCKSTA 位置 1。

安全开关打开过程如果出现非法操作序列，则 IAP 操作会被锁住不能再进行 IAP 操作，IAPSEQERR 会置 1 且 IAPLOCKSTA 位一直为 1，直到下一次复位才能恢复。

- **安全开关打开非法操作序列:**

1. 往 FSMAINKEY/FSINFOKEY 写入非 0x0123CDEF , 0x89AB4567 , 0x4567CDEF, 0x89AB0123 数据

2. 往 FSMAINKEY/FSINFOKEY 写入 0x0123CDEF 后，规定时间内必须对 FSMAINKEY/FSINFOKEY 写 0x89AB4567，否则超时出错

3. 往 FSMAINKEY/FSINFOKEY 写入 0x4567CDEF 后，规定时间内必须对 FSMAINKEY/FSINFOKEY 写 0x89AB0123，否则超时出错

4. 信息区安全开关打开的情况下，对主存储区安全开关进行打开操作

5. 主存储区安全开关打开的情况下，对信息区安全开关进行打开操作

6. 在主存储区、信息区安全开关都关闭时(IAPLOCKSTA=1)，对 FSCON 寄存器进行写操作

7. 在 PROGEN[2: 0]不是 101 状态下，对 FSCON[7: 3]写非 0 数据

8. PROGEN[2: 0]为 101 的状态下，对 FSCON[5: 3]写大于 1 的数据

9. PROGEN[2: 0]为 101 的状态下，对 FSCON[14: 13]写 01、10 的数据

10. FSCON[8]为 0 的状态下，不能改变 FSCON[9]的状态

11. IAP 保护有效(擦除 EFLASH)时，不能设置 FSCON[5](片擦除)

12. IAP 保护有效(编程、擦除 EFLASH)时，不能设置 FSCON[6](不能选信息区)

## 26.4.2. Flash 主存储区字节编程

主存储区字节编程流程:

步骤一: 判断 IAPLOCKSTA 是否为 1，为 1 则往 FSMAINKEY 寄存器中连续写 0x0123CDEF , 0x89AB4567 安全开关打开，然后继续步骤二。 为 0 则直接进行步骤二。

步骤二: 将 PROGEN[2: 0]置成 101，允许启动编程/擦除操作。

步骤三: 如果要编程一个 word(4 个字节), 则将 WRMODE 置成 1, 硬件自动连续编程 4 个字节; 如果只需编程某一个字节(编程字节由 FSADDR[1: 0]决定), 则将 WRMODE 置成 0。

步骤四: 往 FSADDR 寄存器写入目标地址

步骤五: 往 FSDATA 寄存器写入目标数据

步骤六: 将 WRST 置 1, 启动字节编程操作同时等待忙标志位 BSY 置 1

步骤七: 如果是最后一个字节编程结束, 则 IAPFLG 会置 1, 硬件自动清零 FSCON[7: 0];

### 26.4.3.Flash 主存储区页擦除

主存储区页擦除流程:

步骤一: 判断 IAPLOCKSTA 是否为 1, 为 1 则往 FSMINKEY 寄存器中连续写 0x0123CDEF, 0x89AB4567 安全开关打开, 然后继续步骤二。 为 0 则直接进行步骤二。

步骤二: 将 PROGEN[2: 0]置成 101, 允许启动编程/擦除操作。

步骤三: 往 FSADDR 寄存器写入目标地址

步骤四: 将 PERST 置 1, 启动页擦除操作同时等待忙标志位 BSY 置 1

步骤五: 等待 BSY 变为 0, 变为 0 则硬件自动清零 FSCON[5: 3]

步骤六: 页擦除结束, IAPFLG 会置 1, 同时硬件清零 FSCON[7: 0]

### 26.4.4.Flash 信息区字节编程

信息区字节编程流程:

步骤一: 判断 IAPLOCKSTA 是否为 1, 为 1 则往 FSINFOKEY 寄存器中连续写 0x0123CDEF, 0x89AB4567 安全开关打开, 然后继续步骤二。 为 0 则直接进行步骤二。

步骤二: 将 PROGEN[2: 0]置成 101, 允许启动编程/擦除操作。

步骤三: 如果要编程同一个 word(4 个字节), 则将 WRMODE 置成 1, 硬件自动连续编程 4 个字节; 如果只需编程某一个字节(编程字节由 FSADDR[1: 0]决定), 则将 WRMODE 置成 0。

步骤四: 往 FSADDR 寄存器写入目标地址, 将 FSCON[6]置 1

步骤五: 往 FSDATA 寄存器写入目标数据

步骤六: 将 WRST 置 1, 启动字节编程操作同时等待忙标志位 BSY 置 1

步骤七: 如果是最后一个字节编程结束, 则 IAPFLG 会置 1, 硬件自动清零 FSCON[7: 0];

### 26.4.5.Flash 信息区页擦除

信息区页擦除流程:

步骤一: 判断 IAPLOCKSTA 是否为 1, 为 1 则往 FSMINKEY 寄存器中连续写 0x0123CDEF, 0x89AB4567 安全开关打开, 然后继续步骤二。 为 0 则直接进行步骤二。

步骤二: 将 PROGEN[2: 0]置成 101, 允许启动编程/擦除操作。

步骤三: 往 FSADDR 寄存器写入目标地址, 将 FSCON[6]置 1

步骤四: 将 PERST 置 1, 启动页擦除操作同时等待忙标志位 BSY 置 1

步骤五: 等待 BSY 变为 0, 变为 0 则硬件自动清零 FSCON[5: 3]

步骤六: 页擦除结束, IAPFLG 会置 1, 同时硬件清零 FSCON[7: 0]

### 26.4.6. Flash 全片擦除

全片擦除流程: 擦除主存储区、信息区 sector0~2, 不会擦除存放校准数据的信息区 sector3。

步骤一: 判断 IAPLOCKSTA 是否为 1, 为 1 则往 FSMINKEY 寄存器中连续写 0x0123CDEF, 0x89AB4567 安全开关打开, 然后继续步骤二。 为 0 则直接进行步骤二。

步骤二: 将 PROGEN[2: 0]置成 101, 允许启动编程/擦除操作。

步骤三: 将 MERMODE 置 1, 允许全片擦除操作。

步骤四: 将 MERST 置 1, 启动片擦除操作同时等待忙标志位 BSY 置 1

步骤五: 等待 BSY 变为 0, 变为 0 则硬件自动清零 FSCON[5: 3]

步骤六: 片擦除结束, IAPFLG 会置 1, 同时硬件清零 FSCON[7: 0]

## 26.5. Flash 读保护

Flash 读保护状态下是不能通过任何接口(SWD、MO 程序)读取到 Flash 数据, 保护程序的知识产权。将信息区读保护字节 RDP (0x85FC) 写成 0xA5, 然后复位芯片既可以使读保护生效。

- 禁止 SWD 读 flash;
- 禁止 SWD 写内存, 防止改写内存程序读取 flash;

## 26.6. Flash 写保护

写保护是指不能通过 ISP 或 IAP 操作对保护区进行编程/擦除, 以保护重要的 FLASH 数据不会被意外破坏掉。才用以下两种方式实现写保护:

- 将信息区写保护字节 IAPKEY (0x85FD) 写成 0x69, 然后复位芯片就可以使 IAP 写保护生效, 生效后就不能通过 IAP 进行主程序区片擦除、信息区编程或页擦除。
- 将信息区 BLOCK 写保护字 FSWRP(地址:0x85F8-0x85FB), 然后复位芯片就能够以 BLOCK 为单位对主程序区进行保护(禁止页擦除和编程), 每个 BLOCK 为 1kBytes, 最大支持 32kBytes。

BLOCK 序号与 FLASH 地址的关系:

BLOCK序号	FLASH地址范围
---------	-----------

0	0x0000 0000-0x0000 03FF
1	0x0000 0400-0x0000 07FF
2	0x0000 0800-0x0000 0BFF
3	0x0000 0C00-0x0000 0FFF
4	0x0000 1000-0x0000 13FF
5	0x0000 1400-0x0000 15FF
6	0x0000 1800-0x0000 16FF
7	0x0000 1C00-0x0000 17FF
8	0x0000 2000-0x0000 28FF
9	0x0000 2400-0x0000 29FF
10	0x0000 2800-0x0000 2AFF
11	0x0000 2C00-0x0000 2BFF
12	0x0000 3000-0x0000 33FF
13	0x0000 3400-0x0000 37FF
14	0x0000 3800-0x0000 3BFF
15	0x0000 3C00-0x0000 3FFF
16	0x0000 4000-0x0000 43FF
17	0x0000 4400-0x0000 47FF
18	0x0000 4800-0x0000 4BFF
19	0x0000 4C00-0x0000 4FFF
20	0x0000 5000-0x0000 53FF
21	0x0000 5400-0x0000 57FF
22	0x0000 5800-0x0000 5BFF
23	0x0000 5C00-0x0000 5FFF
24	0x0000 6000-0x0000 63FF
25	0x0000 6400-0x0000 67FF
26	0x0000 6800-0x0000 6BFF
27	0x0000 6C00-0x0000 6FFF
28	0x0000 7000-0x0000 73FF
29	0x0000 7400-0x0000 77FF
30	0x0000 7800-0x0000 7BFF
31	0x0000 7C00-0x0000 7FFF

不能  
进行  
外改

## 27. 用户配置

Flash 读保护，保护后  
读出 FLASH 数据

Flash 写保护，保护后  
对保护区的空间用程序  
编程和擦除，防止程序被意  
写。

上面两种保护设置，可以通过 SWD 接口片擦除操作恢复成芯片初始状态。  
详细功能描述见第 26 章。

## 28. 调试模式

### 28.1. 系统控制寄存器 (SYSCON:0x4000E000)

描述位	名称	读写	描述	复位值
0	SYSCCLKS	R/W	见5.2.2章节	
1	CSSSEN	R/W		
4: 2	FCLK_CFG[2: 0]	R/W		
7: 5	PCLK_CFG[2: 0]	R/W		
10: 8			保留	
11	STOP_CFG	R/W	见5.3.1章节	
12	WKEN_EINT0	R/W		
13	WKEN_EINT1	R/W		
14	WKEN_EINT2	R/W		
15	WKEN_EINT3	R/W		
16	WKEN_LVD	R/W		
17	WKEN_RT	R/W	见11.1章节	
18	RTEN	R/W		
21: 19	RT_CFG[2: 0]	R/W	见26.2.1章节	
23: 22	PGCLK_CFG[1: 0]	R/W		
24	PGCLKEN	R/W		
25	DEBUG_MODE	R/W	DEBUG模式控制位。 0: PB[12: 11]复用为GPIO功能 1: PB[12: 11]复用为SWD调试接口（此时对应IO配置无效，RCH无法关闭，VBG无法关闭）	1
31: 26			保留	

### 28.2. SWD 复用成 GPIO

芯片默认 PB[12: 11]为 SWD 功能，支持 4/5 线调试(有无 NRST 管脚都可以)。

当程序将 PB[12: 11]改为 GPIO 功能，即 SYSCON 的 DEBUG\_MODE 置 0。这种情况下，必须要 5 线调试，需要调试器控制 NRST 将 PB[12: 11]恢复成 SWD 功能，然后进入调试模式。

## 29. 版本历史

版本号	修改内容	修改时间
V1.0	初始版本	
V1.1	1. 修改了DAC的描述	2024.03.14
V1.11	1. 添加 PB15 复用 AIN7。 2. 修改 EPWM 里的表 14-3-4-1	2024.05.06
V1.12	1. 修改 EPWM 里触发中断功能描述	2024.05.07
V1.13	1. 修改 WDOG 里复位时间公式	2024.05.11
V1.14	1. 修改 5.2.3 PLL 的框图和 PLL 时钟计算公式	2024.05.15
V1.15	1. 增加 5.5.3 温度传感器计算公式	2024.05.23
V1.16	1. 增加 14.2.5 捕获比较配置寄存器 UPDATE_TRIG 的补充说明 2. 15.4 功能描述部分增加 PWM 长时间固定电平的计数结果说明	2024.05.24
V1.17	1. 修改 24.4.9 异步采样电流时的 Tdly1、Tsoc1 说明 2. 增加异步双采样时序图	2024.05.23
V1.18	1. 增加中断 30 说明：ADC 组转换模式时 0、1 通道转换完成中断	2024.06.21
V1.19	1. 更改内核复位状态标志位的读写说明	2024.06.25
V1.20	1. 修改了 timer0 的 UEV 寄存器	2024.07.03
V1.21	1. 系统时钟图里晶振输入改为 8-16mHZ	2024.07.15