

## 概述

OB6625 是一款高集成、高性能、低成本的 3 相直流无刷电机 (BLDC) 或永磁同步电机(PMSM)控制芯片, 只需要少量的外围器件即可搭建完整的磁场定向控制 (FOC) 电路。集成了 32bit ARM Cortex-M0 MCU , 3P+3N 全桥驱动器, 高精度 5V 稳压电源。

OB6625 提供完整芯片保护和外部电机驱动电流保护, 包括芯片工作电源欠压保护、过压保护, 过温保护, 桥臂防直通保护; 驱动电路内置死区; 电机驱动过流保护等。内置 MCU 工作主频可以达到 72Mhz, 提供 32kB FLASH 和 4kB SRAM, 集成了坐标旋转数字计算机 (Cordic) 和乘法除法单元 (MDU), 专为电机控制优化了脉宽调制 (PWM), 运算放大器 (Amplifier), 模拟比较器 (Comparator), 模拟数字采样器 (ADC), 和不同模块间联动, 易于开发高性能算法。

OB6625 应用于低成本, 高性能, 高效率 BLDC/PMSM 驱动电路, 通过使用少量外围组件即可轻松构建具有电流环路和速度环路控制电路。兼具灵活性和低成本要求。

OB6625 提供 TSSOP20/TSSOP24/QFN28 封装, 有利于 PCB 小型化设计。

## 应用

- 5V 至 28V 直流输入, 通用 BLDC/PMSM 控制器, 包括扇类/泵类/按摩器/运动机构等。

## 特性

### 三相 P+N MOSFET 驱动器

- 高压侧 P-MOSFET 驱动耐压 40V
- 高压侧 P -MOSFET 驱动电压 11V, 驱动电流 0.3/0.22A
- 低压侧 N-MOSFET 驱动电压 -11V, 驱动电流 0.43/0.6A
- 低压侧驱动器电源欠压保护 (UVLO)、过压保护 (OVP)
- 内置死区
- 内置防桥臂直通逻辑

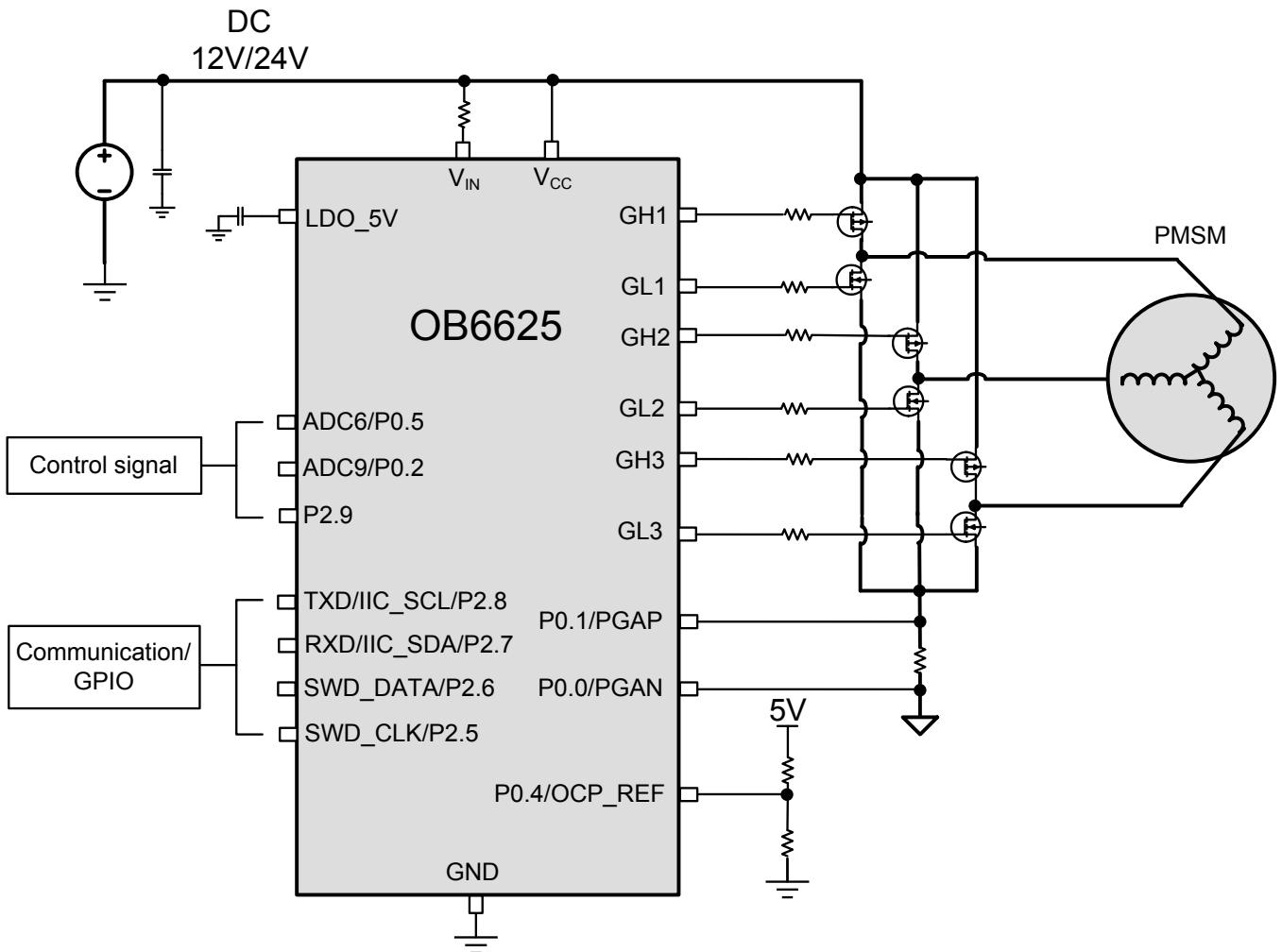
### LDO

- 输入电压最大 35V
- 输出电压精度  $5V \pm 2\%$ , 最大输出电流 100mA
- 内置短路保护和过温保护

### ARM Cortex-M0 内核

- 工作主频 72MHz , 内置嵌套向量中断控制器 (NVIC)
- 宽工作电压范围: 1.8V 至 5.5V
- 片上闪存编程存储器 32KB
- 4KB SRAM
- 1 组 32bit 定时器/计数器; 2 组 16bit 定时器/计数器
- 6 通道 16bit PWM
- 9 通道 12bit ADC
- 3 组 8MHz, 10V/us 运算放大器
- 3 组通用比较器
- 串行在线调试(SWD)
- 坐标旋转数字计算机 (Cordic)
- 乘除法单元 (MDU)

### 典型应用图



**目录**

概述 .....	1
应用 .....	1
特性 .....	1
典型应用图 .....	5
1. 基本信息 .....	14
2. 功能方框图 .....	10
3. 电气参数 .....	11
4. 特征曲线图 .....	12
5. 工作模式描述 .....	14
6. 系统配置 .....	15
6.1 存储器映射 .....	15
6.2 寄存器说明 .....	16
6.3 重置控制器 .....	17
6.3.1 重置源和框图 .....	17
6.3.2 重置注释项目 .....	17
6.4 时钟控制器 .....	18
6.4.1 概述 .....	18
6.4.2 时钟发生器 .....	18
6.4.3 CPU 时钟配置寄存器(CCLKCFG – 0x5001 0010) .....	20
6.4.4 电源/时钟门控外设寄存器(PCGP – 0x5001 0020) .....	20
6.4.5 功率控制使用说明 .....	21
6.4.6 DPLL 控制寄存器(DPLLCON – 0x5001 0080) .....	21
6.4.7 DPLL2 控制寄存器(DPLL2CON – 0x5001 0084) .....	22
6.4.8 DPLL 调试模式寄存器(DPLLDM – 0x5001 0090) .....	22
6.4.9 CPU / CHIP 控制 0 寄存器(CPUCHIPCTR0 – 0x5001 00B0) .....	23
6.4.10 LVR / LVI 控制/状态 0 寄存器(LVRLVICTRSTA0 – 0x5001 00C0) .....	23
6.4.11 USR/Flash 控制 0 寄存器(USRFLSHCTR0 – 0x5001 00D0) .....	24
6.5 功率控制器 .....	26
6.5.1 睡眠模式 .....	26
6.5.2 深度睡眠模式 .....	26
6.5.3 功率模式控制寄存器(PCON – 0x5001 0024) .....	27
6.5.4 从低功耗模式中唤醒 .....	27
7. 嵌套向量中断控制器(NVIC) .....	28
7.1 描述 .....	28
7.2 特性 .....	28
7.3 中断源 .....	28
8. I2C-总线控制 .....	30
8.1 特性 .....	30
8.2 应用 .....	30
8.3 概述 .....	30
8.4 引脚描述 .....	31
8.5 寄存器描述 .....	31
8.5.1 控制置位寄存器 (CONSET – 0x4000 0000) .....	32
8.5.2 状态寄存器 (STAT – 0x4000 0004) .....	33
8.5.3 数据寄存器(DAT – 0x4000 0008) .....	33
8.5.4 从地址寄存器 0 (ADR0 – 0x4000 000C) .....	33
8.5.5 SCL 高低电平占空比寄存器(SCLH – 0x4000 0010, SCLL – 4000 0014) .....	33
8.5.6 选择合适的 I2C 总线速率和占空比 .....	34
8.5.7 控制清零寄存器(CONCLR – 0x4000 0018) .....	34
8.5.8 从地址寄存器 1(ADR1 – 0x4000 0020) .....	34

8.5.9	数据缓冲器寄存器(DATABUFFER – 0x4000 002C).....	35
8.5.10	屏蔽寄存器 (MASK0/1 – 0x4000 0030/34).....	35
8.5.11	读数据寄存器(RDDAT – 0x4000 0038).....	35
8.6	I2C 操作模式 .....	35
8.6.1	主发送模式 .....	35
8.6.2	主接收模式 .....	36
8.6.3	从接收模式 .....	37
8.6.4	从发送模式 .....	37
8.7	I2C 的实现和操作 .....	38
8.7.1	输入滤波器和输出级 .....	40
8.7.2	地址寄存器, ADDR0 与 ADDR1 .....	40
8.7.3	地址屏蔽寄存器, ADDR0 与 ADDR1 .....	40
8.7.4	比较器 .....	40
8.7.5	移位寄存器, DAT .....	40
8.7.6	仲裁和同步逻辑 .....	40
8.7.7	串行时钟发生器 .....	41
8.7.8	时序和控制 .....	41
8.7.9	控制寄存器, CONSET 和 CONCLR.....	41
8.7.10	状态译码器和状态寄存器.....	42
8.8	I2C 操作模式的详解.....	42
8.8.1	主发送模式 .....	42
8.8.2	主接收模式 .....	45
8.8.3	从接收模式 .....	47
8.8.4	从发送模式 .....	50
8.8.5	其他状态.....	51
8.8.6	一些特殊情况.....	52
8.8.7	I2C 状态服务程序.....	54
8.8.8	初始化 .....	55
8.8.9	I2C 中断服务 .....	55
8.8.10	状态服务程序.....	55
8.8.11	实际应用中的状态服务 .....	55
8.9	软件示例.....	55
8.9.1	初始化程序 .....	55
8.9.2	启动主机发送功能.....	55
8.9.3	启动主机接收功能.....	55
8.9.4	I2C 中断程序.....	56
8.9.5	无指定模式的状态.....	56
8.9.6	主发送状态 .....	56
8.9.7	主接收状态 .....	57
8.9.8	从接收状态 .....	58
8.9.9	从发送状态 .....	59
9.	串行外设接口(SPI).....	61
9.1	特性.....	61
9.2	简介.....	61
9.3	引脚描述.....	61
9.4	寄存器描述 .....	61
9.4.1	控制寄存器 0 (CR0 – 0x5001 8000) .....	62
9.4.2	控制寄存器 1 (CR1 – 0x5001 8004) .....	62
9.4.3	数据寄存器 (DR – 0x5001 8008).....	63
9.4.4	状态寄存器 (SR – 0x5001 800C).....	63
9.4.5	时钟预分频寄存器(CPSR – 0x5001 8010).....	63



9.4.6	中断掩码设置 / 清除寄存器(IMSC – 0x5001 8014)	64
9.4.7	原始中断状态寄存器(RIS – 0x5001 8018)	64
9.4.8	掩码后中断状态寄存器(MIS – 0x5001 801C)	64
9.4.9	中断清除寄存器(ICR – 0x5001 8020)	65
9.5	功能说明	65
9.5.1	德州仪器同步串行帧格式	65
9.5.2	SPI 帧格式	65
9.5.3	半导体 Microwire 帧格式	69
9.5.4	SSNP 功能	71
10.	UART	72
10.1	特性	72
10.2	引脚描述	72
10.3	寄存器描述	72
10.3.1	UART 接收器缓冲寄存器(URBR – 0x5002 8000)	73
10.3.2	UART 发送器保持寄存器(UTHR – 0x5002 8000)	73
10.3.3	UART 除数锁存寄存器(UDL – 0x5002 8008)	73
10.3.4	UART 中断使能寄存器(UIER – 0x5002 800C)	73
10.3.5	UART 中断识别寄存器(UIIR – 0x5002 8010)	74
10.3.6	UART FIFO 控制寄存器(UFCR – 0x5002 8014)	75
10.3.7	UART FIFO RX 计数器寄存器(UFRC – 0x5002 8018)	75
10.3.8	UART FIFO TX 计数器寄存器(UFTC – 0x5002 801C)	76
10.3.9	UART 线路控制寄存器(ULCR – 0x5002 8020)	76
10.3.10	UART 线路状态寄存器(ULSR – 0x5002 8024)	76
10.3.11	UART 自动波特率控制寄存器(UACR – 0x5002 8028)	77
10.3.12	自动波特率	77
10.3.13	自动波特率模式	78
10.3.14	波特率计算	79
10.3.15	UART RS485 控制寄存器(URS485CTRL – 0x5002 804C)	79
10.3.16	UART RS485 地址匹配寄存器(URS485ADRMATCH – 0x5002 8050)	79
10.3.17	RS-485/EIA-485 模式的操作	80
10.4	架构	80
11.	16 位计数器/定时器	82
11.1	特性	82
11.2	应用	82
11.3	描述	82
11.4	引脚描述	82
11.5	寄存器描述	82
11.5.1	中断寄存器(TMR16IR – 0x4000 C000, 0x4001 0000)	83
11.5.2	定时器控制寄存器(TMR16TCR – 0x4000 C004, 0x4001 0004)	83
11.5.3	定时器计数器寄存器(TMR16TC – 0x4000 C008, 0x4001 0008)	84
11.5.4	预分频寄存器(TMR16PR – 0x4000 C00C, 0x4001 000C)	84
11.5.5	预分频计数器寄存器(TMR16PC – 0x4000 C010, 0x4001 0010)	84
11.5.6	匹配控制寄存器(TMR16MCR – 0x4000 C014, 0x4001 0014)	84
11.5.7	匹配寄存器(TMR16MR0/1/2/3 – 0x4000 C018/1C/20/24, 0x4001 0018/1C/20/24)	85
11.5.8	捕获控制寄存器(TMR16CCR – 0x4000 C028, 0x4001 0028)	85
11.5.9	捕获寄存器(TMR16CR0/1/2 – 0x4000 C02C/30/34, 0x4001 002C/30/34)	86
11.5.10	CAP0/1/2 数字滤波器寄存器(TMR16CDF0/1/2 – 0x4000 C050/54/58, 0x4001 0050/54/58)	86
11.5.11	计数控制寄存器(TMR16CTCR – 0x4000 C070, 0x4001 0070)	87
11.6	定时器操作示例	87
11.7	架构	90
12.	32 位计数器/定时器	91

12.1	特性.....	91
12.2	应用.....	91
12.3	简介.....	91
12.4	寄存器描述.....	92
12.4.1	中断寄存器(TMR32IR – 0x5002 C000).....	92
12.4.2	定时器控制寄存器(TMR32TCR – 0x5002 C004).....	92
12.4.3	定时器计数器寄存器(TMR32TC – 0x5002 C008).....	93
12.4.4	预分频寄存器(TMR32PR – 0x5002 C00C).....	93
12.4.5	预分频计数器寄存器(TMR32PC – 0x5002 C010).....	93
12.4.6	匹配控制寄存器(TMR32MCR – 0x5002 C014).....	93
12.4.7	匹配寄存器(TMR32MR0/1/2/3 – 0x5002 C018/1C/20/24).....	94
12.4.8	Trigger ADC Register (TMR32TADC0 – 0x5002 C038).....	94
12.5	功能描述.....	95
12.5.1	定时器操作示例.....	95
12.6	架构.....	97
13.	IO 配置(端口输入/输出).....	98
13.1	特征.....	98
13.2	应用.....	98
13.3	寄存器说明.....	98
13.3.1	GPIO 端口 X 数据寄存器(DATA – 0x5000 0000, 0x5000 4000, 0x5000 8000).....	99
13.3.2	GPIO 端口 X 引脚值寄存器(PIN – 0x5000 0004, 0x5000 4004, 0x5000 8004).....	99
13.3.3	GPIO 端口 X 模式寄存器(MODE – 0x5000 0008, 0x5000 4008, 0x5000 8008).....	100
13.3.4	GPIO 端口 X 数据输出写屏蔽寄存器(WMASK – 0x5000 000C, 0x5000 400C, 0x5000 800C).....	100
13.3.5	GPIO 端口 X 中断使能设置 0 寄存器(IEENS0 – 0x5000 0014, 0x5000 4014, 0x5000 8014).....	101
13.3.6	GPIO 端口 X 中断使能设置 1 寄存器(IEENS1 – 0x5000 0018, 0x5000 4018, 0x5000 8018).....	102
13.3.7	GPIO 端口 X 中断去抖动使能寄存器(IDBEN – 0x5000 001C, 0x5000 401C, 0x5000 801C).....	102
13.3.8	GPIO 端口 X 中断去抖计数器寄存器(IDBCU – 0x5000 0020, 0x5000 4020, 0x5000 8020).....	102
13.3.9	GPIO 端口 X 中断状态寄存器(IEST – 0x5000 0024, 0x5000 4024, 0x5000 8024).....	103
13.3.10	GPIO 端口 X 中断状态清除寄存器(IESTC – 0x5000 0028, 0x5000 4028, 0x5000 8028).....	103
13.3.11	GPIO 端口 X 触发 ADC 0 设置 0 寄存器(TADC0S0 – 0x5000 002C, 0x5000 402C, 0x5000 802C).....	103
13.3.12	GPIO 端口 X 触发 ADC 0 设置 1 寄存器(TADC0S1 – 0x5000 0030, 0x5000 4030, 0x5000 8030).....	104
13.3.13	GPIO 端口 0 多功能 0 寄存器(MF0 – 0x5000 003C).....	104
13.3.14	GPIO 端口 1 多功能 0 寄存器(MF0 – 0x5000 403C).....	105
13.3.15	GPIO 端口 2 多功能 0 寄存器(MF0 – 0x5000 803C).....	106
13.3.16	GPIO 端口 2 多功能 1 寄存器(MF1 – 0x5000 8040).....	106
14.	闪存控制器(FMC).....	108
14.1	概述.....	108
14.2	特征.....	108
14.3	闪存组织.....	108
14.4	框图.....	109
14.5	闪存控制 (FMC) 功能.....	110
14.5.1	启动选择功能.....	110
14.5.2	在线更新(ISP).....	111
14.5.3	ISP 程序.....	111
14.5.4	User Code(IAP) 功能 带/不带 ISP 代码.....	113
14.5.5	ISP Code 功能.....	115
14.6	用户选项 (配置).....	115
14.6.1	程序内存读取保护 (MMCRP).....	116

14.6.2	ISP 启动(ISPET).....	116
14.6.3	IAP 开始/结束地址(IAPSA/IAPEA).....	116
14.7	寄存器说明.....	117
14.7.1	ISP 控制/命令寄存器(ISPCC – 0x4003 C000).....	118
14.7.2	ISP 地址寄存器(ISPADR – 0x4003 C004).....	118
14.7.3	ISP 数据寄存器(ISPDAT – 0x4003 C008).....	118
14.7.4	ISP 触发器/状态寄存器(ISPTS – 0x4003 C00C).....	118
14.7.5	ICP 启动寄存器(ICPENTRY – 0x4003 C040).....	119
14.7.6	主存储器代码读保护寄存器(MMCRPR – 0x4003 C080).....	119
14.7.7	引导选择控制/状态寄存器(BSCSR – 0x4003 C084).....	119
15.	窗口看门狗定时器(WWDT).....	121
15.1	概述.....	121
15.2	特征.....	121
15.3	一般描述.....	121
15.4	寄存器概述.....	123
15.4.1	看门狗定时器重载计数器寄存器(WWDTRLD – 0x4000 4000).....	123
15.4.2	看门狗定时器控制寄存器(WWDTCR – 0x4000 4004).....	124
15.4.3	看门狗定时器状态寄存器(WWDTSR – 0x4000 4008).....	124
15.4.4	看门狗定时器计数器值寄存器(WWDTCVR – 0x4000 400C).....	124
16.	功率控制 PWM 模块.....	125
16.1	介绍.....	125
16.2	特征.....	125
16.3	模块功能.....	125
16.4	PWM 时基.....	126
16.4.1	自由运行模式.....	126
16.4.2	单发模式.....	127
16.4.3	连续递增/递减计数模式.....	128
16.5	PWM 时钟选择.....	128
16.6	PWM 计数器选择.....	129
16.7	PWM 中断.....	130
16.8	PWM 周期比较.....	131
16.9	PWM ADC 触发器.....	132
16.10	PWM 触发 ADC 的运作时序.....	134
16.11	PWM 触发 ADC 注意事项.....	134
16.12	PWM 死区.....	135
16.13	PWM 输出.....	136
16.14	PWM 故障输入.....	138
16.12.1	故障引脚使能位.....	139
16.12.2	故障状态时的 PWM 输出.....	139
16.15	PWM 更新.....	140
16.16	使用 PWM 触发模拟比较器方式.....	141
16.17	寄存器描述.....	143
16.15.1	PWM 控制寄存器(PWMCON – 0x5001 C000).....	145
16.15.2	PWM 最大比较寄存器(MAX0/1 – 0x5001 C004/08).....	146
16.15.3	PWM 计数器选择寄存器(PWMCS – 0x5001 C00C).....	146
16.15.4	PWM 输出使能寄存器(OUTEN – 0x5001 C010).....	146
16.15.5	PWM 输出初始寄存器(OUTINI – 0x5001 C014).....	147
16.15.6	PWM 0A 输出控制寄存器(OUTCON0A – 0x5001 C018).....	147
16.15.7	PWM 0B 输出控制寄存器(OUTCON0B – 0x5001 C01C).....	149
16.15.8	PWM 1A 输出控制寄存器(OUTCON1A – 0x5001 C020).....	151
16.15.9	PWM 1B 输出控制寄存器(OUTCON1B – 0x5001 C024).....	153

16.15.10	PWM 2A 输出控制寄存器(OUTCON2A – 0x5001 C028).....	155
16.15.11	PWM 2B 输出控制寄存器(OUTCON2B – 0x5001 C02C).....	156
16.15.12	PWM 死区寄存器(DB0/1/2 – 0x5001 C048/4C/50).....	158
16.15.13	PWM 比较寄存器(PWMCMP0/1/2/3/4/5 – 0x5001 C060/64/68/6C/70/74).....	159
16.15.14	PWM 当前计数器寄存器(PWMCU0/1 – 0x5001 C078/7C).....	159
16.15.15	PWM 周期比较寄存器(CYCMP0/1 – 0x5001 C080/84).....	159
16.15.16	PWM 周期计数器寄存器(CYCU0/1 – 0x5001 C088/8C).....	159
16.15.17	PWM ADC 触发延迟 0U 寄存器(ADCTDEL0U – 0x5001 C090).....	159
16.15.18	PWM ADC 延时当前计数器 0 寄存器(DELCU0 – 0x5001 C098).....	160
16.15.19	PWM 中断使能设置寄存器(IEENS – 0x5001 C0A0).....	160
16.15.20	PWM 中断使能清除寄存器(IEENC – 0x5001 C0A4).....	161
16.15.21	PWM 事件状态寄存器(EVST – 0x5001 C0A8).....	162
16.15.22	PWM 事件状态清除寄存器(EVSTC – 0x5001 C0AC).....	163
16.15.23	PWM 故障配置寄存器(FLTCON – 0x5001 C0B0).....	163
16.15.24	PWM 故障输出使能寄存器(FLTOTEN – 0x5001 C0B4).....	165
16.15.25	PWM 故障输出电平寄存器(FLTOL – 0x5001 C0B8).....	165
16.15.26	PWM 故障噪声滤波器寄存器(FLTNFA – 0x5001 C0BC).....	166
16.15.27	PWM 故障 B 噪声滤波器寄存器(FLTNFB – 0x5001 C0C0).....	166
16.15.28	PWM 故障 C 噪声滤波器寄存器(FLTNFC – 0x5001 C0C4).....	166
16.15.29	PWM 状态寄存器(STA – 0x5001 C0C8).....	166
16.15.30	PWM 触发器 ADC 0 使能寄存器(TADC0 – 0x5001 C0CC).....	167
16.15.31	PWM 更新模式使能寄存器(UPDATE – 0x5001 C0D4).....	169
16.15.32	PWM 故障释放寄存器(FLTREL – 0x5001 C0D8).....	171
16.15.33	PWM 故障状态寄存器(FLTST – 0x5001 C0DC).....	173
16.15.34	PWM 仿真模式寄存器(ICE – 0x5001 C0E0).....	174
16.15.35	PWM 计数器初始寄存器(CUINIT0/1 – 0x5001 C0E4/E8).....	174
16.15.36	PWM 比较器触发控制寄存器(TACMP – 0x5001 C0EC).....	175
16.15.37	PWM 当前计数器 COMP 寄存器(CUCOMP – 0x5001 C0F0).....	175
16.15.38	PWM 输出固定寄存器(OUTFIX – 0x5001 C100).....	175
16.15.39	PWM 输出反相寄存器(OUTIVT – 0x5001 C104).....	175
16.15.40	PWM ADC 触发延迟 0D 寄存器(ADCTDEL0D – 0x5001 C108).....	176
17.	坐标旋转数字计算机(CORDIC).....	177
17.1	概观.....	177
17.2	特性.....	177
17.3	CORDIC 方块图.....	177
17.4	功能说明.....	178
17.4.1	圆形坐标系(CORDIC : m=1).....	178
17.4.2	双曲型坐标系 (CORDIC : m=-1).....	179
17.4.3	CORDIC 转换实现.....	180
17.4.4	广义 CORDIC 算法.....	181
17.4.5	CORDIC 操作模式和功能.....	181
17.4.6	例 1 : 圆形旋转.....	185
17.4.7	例 2 : 圆形旋转.....	186
17.4.8	例 1 : 圆矢量.....	187
17.4.9	例 2 : 圆矢量.....	188
17.4.10	CORDIC 圆形旋转角度.....	189
17.5	寄存器说明.....	192
17.5.1	CORDIC 控制寄存器(CORDIC_CTR – 0x5001 4000).....	194
17.5.2	CORDIC 数据格式寄存器(CORDIC_X/Y/Z – 0x5001 4010/14/18).....	194
18.	乘法除法单元(MDU).....	195
18.1	概观.....	195

18.2	MDU 操作说明.....	195
18.2.1	装载 MDx 寄存器 .....	195
18.2.2	执行计算.....	195
18.2.3	从 MDx 寄存器读取结果.....	195
18.2.4	标准化 .....	195
18.2.5	移位 .....	195
18.3	寄存器说明 .....	195
18.3.1	算术控制寄存器(ARCON – 0x5001 4400(MDU0),0x5001 4800(MDU1)).....	196
18.3.2	乘法 / 除法寄存器((MD0/1/2 – 0x5001 4404/08/0C)(MDU0), 0x5001 4804/08/0C)(MDU1))	196
19.	模拟数字转换器(ADC).....	198
19.1	介绍.....	198
19.2	特征.....	198
19.3	功能描述.....	198
19.3.1	校准功能.....	198
19.3.2	ADC 时钟.....	199
19.3.3	ADC 通道选择.....	199
19.3.4	开始转换.....	199
19.3.5	数据管理.....	199
19.3.6	ADC 转换时间.....	202
19.3.7	ADC 中断进入方式 .....	202
19.4	管脚描述.....	204
19.5	寄存器描述 .....	204
19.5.1	ADC 控制寄存器(ADCCON – 0x5002 0000) .....	205
19.5.2	ADC 0 采样序列 0 寄存器 (A0SC0 - 0x5002 0004) .....	205
19.5.3	ADC 0 采样序列 1 寄存器 (A0SC1 - 0x5002 0008) .....	206
19.5.4	ADC 0 样本序列长度寄存器(SL0 – 0x5002 000C).....	207
19.5.5	ADC 0 采样序列计数器寄存器(CU0 – 0x5002 0010).....	207
19.5.6	ADC 0 采样结果 0 寄存器(A0RS0 – 0x5002 0014).....	208
19.5.7	ADC 0 采样结果 1 寄存器(A0RS1 – 0x5002 0018).....	208
19.5.8	ADC 0 采样结果 2 寄存器(A0RS2 – 0x5002 001C) .....	208
19.5.9	ADC 0 采样结果 3 寄存器(A0RS3 – 0x5002 0020).....	208
19.5.10	ADC 0 采样结果 4 寄存器(A0RS4 – 0x5002 0024).....	208
19.5.11	ADC 0 采样结果 5 寄存器(A0RS5 – 0x5002 0028).....	209
19.5.12	ADC 0 采样结果 6 寄存器(A0RS6 – 0x5002 002C) .....	209
19.5.13	ADC 0 采样结果 7 寄存器(A0RS7 – 0x5002 0030).....	209
19.5.14	ADC 0 采样和保持寄存器(SH0 – 0x5002 0034) .....	210
19.5.15	ADC 0 中断使能设置寄存器(IEENS – 0x5002 0070).....	210
19.5.16	ADC 0 中断使能清除寄存器(IEENC – 0x5002 0074).....	210
19.5.17	ADC 0 中断状态寄存器(IEST – 0x5002 0078).....	210
19.5.18	ADC 0 中断状态清除寄存器(IESTC – 0x5002 007C).....	211
19.5.19	ADC 0 结果索引寄存器(RSIN0 – 0x5002 0088) .....	211
19.5.20	ADC 0 时钟分频寄存器(ADC0CD – 0x5002 0090).....	211
19.5.21	ADC 0 校准控制寄存器(CALCON0 – 0x5002 00A0).....	211
19.5.22	ADC 0 校准状态寄存器(CALSTA0 – 0x5002 00A4).....	212
19.5.23	ADC 0 锁存计数器寄存器(LATCON – 0x5002 00E8).....	212
20.	模拟比较器(ACMP).....	213
20.1	简介 .....	213
20.2	特征.....	213
20.3	功能描述.....	214
20.3.1	COMP 引脚和内部信号.....	216
20.3.2	中断和事件 .....	216



20.3.3	迟滞功能.....	216
20.4	引脚描述.....	216
20.5	寄存器描述.....	216
20.5.1	比较器控制寄存器(ACMPCON – 0x5002 4000).....	217
20.5.2	比较器迟滞寄存器(HYS – 0x5002 4008).....	218
20.5.3	比较器输出寄存器(OUT – 0x5002 4004).....	218
20.5.4	比较器输出状态寄存器(OUTSTA – 0x4005 000C).....	220
20.5.5	比较器触发 ADC 0 寄存器(TADC0 – 0x5002 4010).....	220
20.5.6	比较器中断使能设置寄存器(IEENS – 0x5002 4018).....	220
20.5.7	比较器中断使能清除寄存器(IEENC – 0x5002 401C).....	221
20.5.8	比较器中断状态寄存器(IEST – 0x5002 4020).....	221
20.5.9	比较器中断状态清除寄存器(IESTC – 0x5002 4024).....	221
20.5.10	比较器去反弹计数器寄存器(DBCNT0/1/2/3 – 0x5002 4030/34/38).....	223
21.	运算放大器(OPA).....	224
21.1	引脚描述.....	224
21.2	寄存器描述.....	224
21.2.1	OPAMP 控制寄存器(OPACON – 0x5002 4080).....	224
22.	循环冗余校验码(CRC).....	226
22.1	介绍.....	226
22.2	特征.....	226
22.3	寄存器描述.....	227
22.3.1	CRC 控制寄存器(CTL – 0x4010 00D0).....	227
22.3.2	CRC 偏移周期寄存器(OCR – 0x4010 00D4).....	229
22.3.3	CRC 传输周期计数寄存器(TCR – 0x4010 00D8).....	229
22.3.4	CRC 写数据寄存器(WD – 0x4010 00E4).....	229
22.3.5	CRC 种子寄存器(SEED – 0x4010 00E8).....	229
22.3.6	CRC 校验码寄存器(CS – 0x4010 00EC).....	229
23.	内部电压(IV).....	230
23.1	特征.....	230
23.2	功能说明.....	230
23.3	引脚说明.....	230
23.4	寄存器描述.....	230
23.4.1	内部电压控制寄存器(IVCON – 0x5002 00FC).....	230
24.	电气特性.....	231
24.1	绝对最大额定值.....	231
24.2	运行条件.....	231
24.3	DC 电气特性.....	232
24.4	AC 电气特性.....	234
24.4.1	Internal High Speed RC Oscillator (HIRC).....	234
24.4.2	20 KHz Internal RC Oscillator (LIRC1).....	234
24.4.3	SAR ADC.....	235
24.4.4	Low Voltage Reset.....	236
24.4.5	Low Voltage Interrupt.....	236
24.4.6	Comparator.....	236
24.4.7	Internal Voltage.....	236
24.4.8	OPAMP.....	236
25.	包装尺寸.....	238

## 1. 基本信息

### 订购信息

型号	描述
OB6625VIP	TSSOP20, Halogen-free in Tube
OB6625VIPA	TSSOP20, Halogen-free in T&R
OB6625VKP	TSSOP24, Halogen-free in Tube
OB6625VKPA	TSSOP24, Halogen-free in T&R
OB6625FLP	QFN28, Halogen-free in Tray

### 额定封装散热

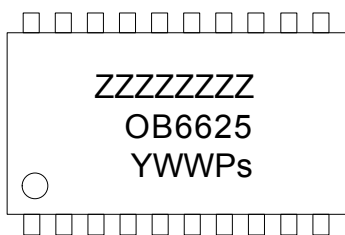
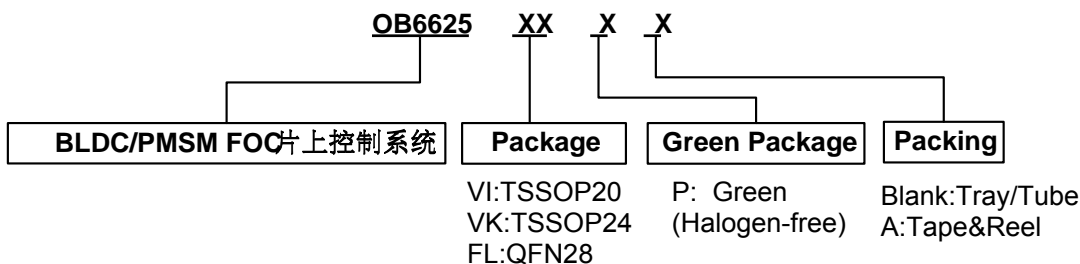
封装	R $\theta$ JA (°C/W)
TSSOP20	
TSSOP24	86
QFN28	30

### 应用极限参数

符号	描述	最大值
VCC, Vin	DC Supply Voltage	40V
GH1, GH2, GH3	High Side P <sub>MOS</sub> Gatedriver	40V
MCU_IO	MCU General IO Input Voltage	5.5V
T <sub>J</sub>	Min/Max Operating Junction Temperature	-40 to 150°C
T <sub>STG</sub>	Min/Max Storage Temperature	-40 to 150°C

备注: 芯片的实际应用条件超出规定的“应用极限值”将会对芯片造成永久性损伤。以上应用极限值标志了芯片可承受应力等级, 但并不建议芯片在此极限条件或超出“推荐工作条件”下工作。芯片长时间处于最大额定工作条件, 会影响芯片的可靠性。

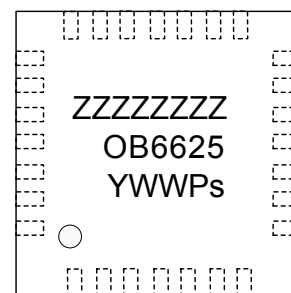
### 标识信息



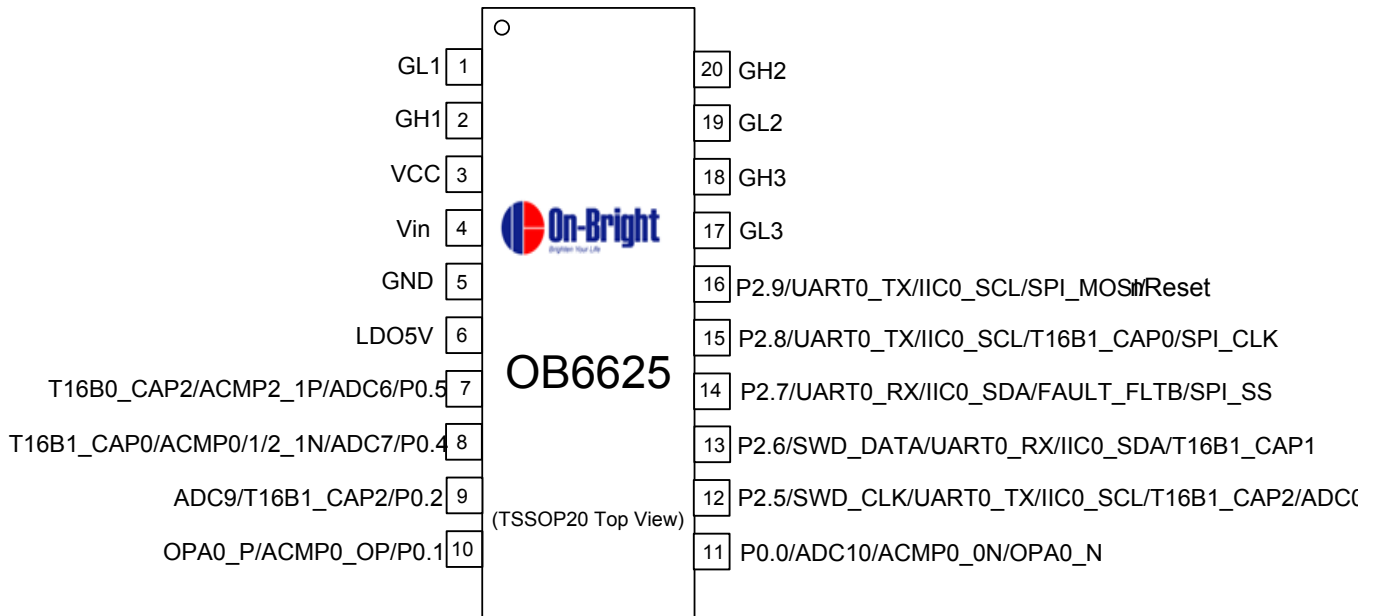
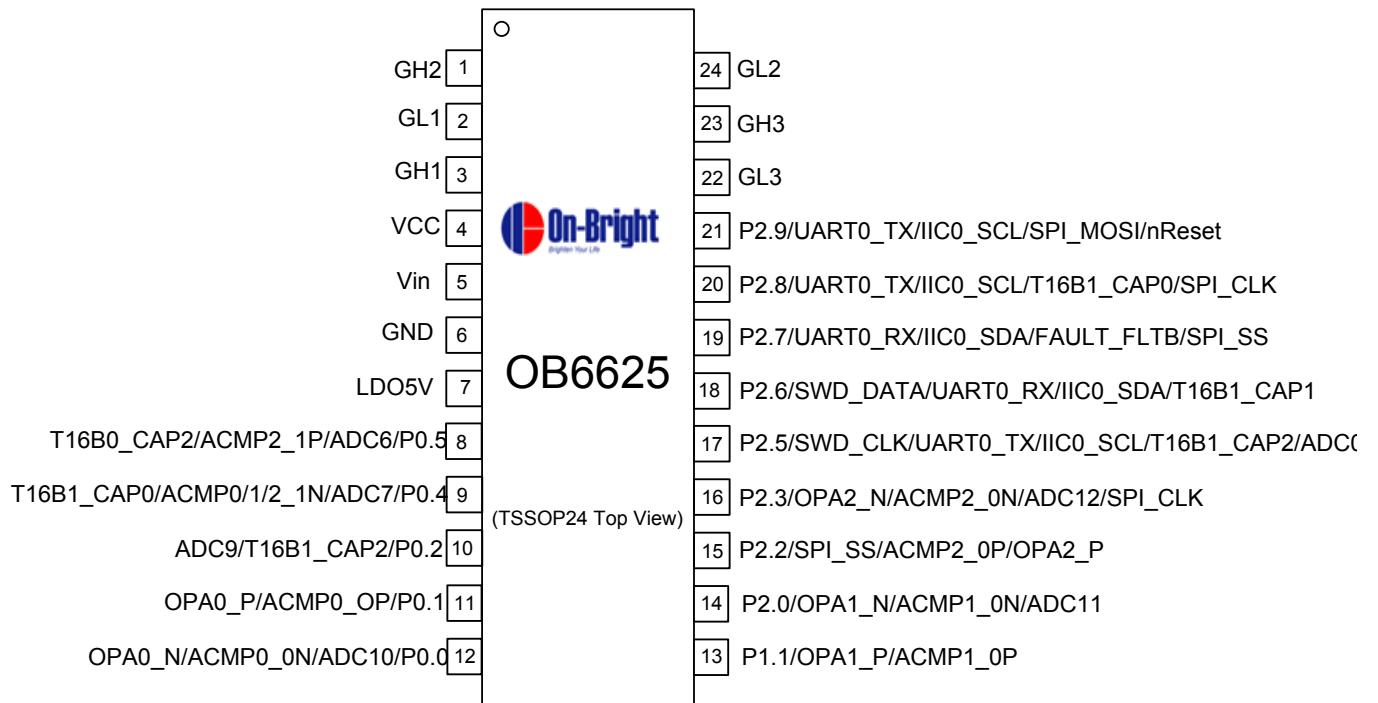
ZZZZZZZZ: Lot Code  
 Y: Year Code  
 WW: Week Code(01-52)  
 P: Halogen-free Package  
 S: Internal Code(Optional)



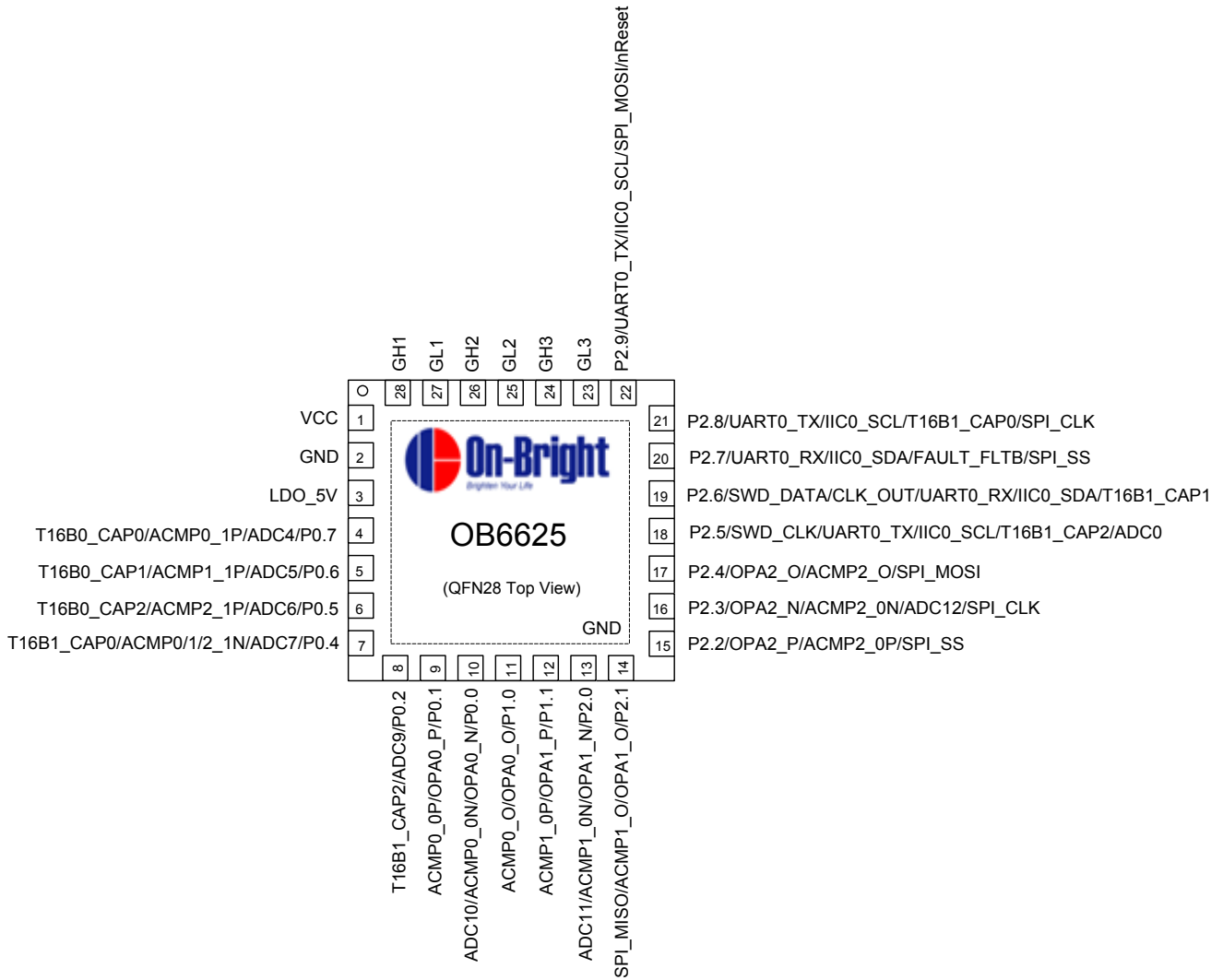
ZZZZZZZZ: Lot Code  
 Y: Year Code  
 WW: Week Code(01-52)  
 P: Halogen-free Package  
 S: Internal Code(Optional)



ZZZZZZZZ: Lot Code  
 Y: Year Code  
 WW: Week Code(01-52)  
 P: Halogen-free Package  
 S: Internal Code(Optional)

**引脚排布及说明 (TSSOP20)**

**引脚排布及说明 (TSSOP24)**




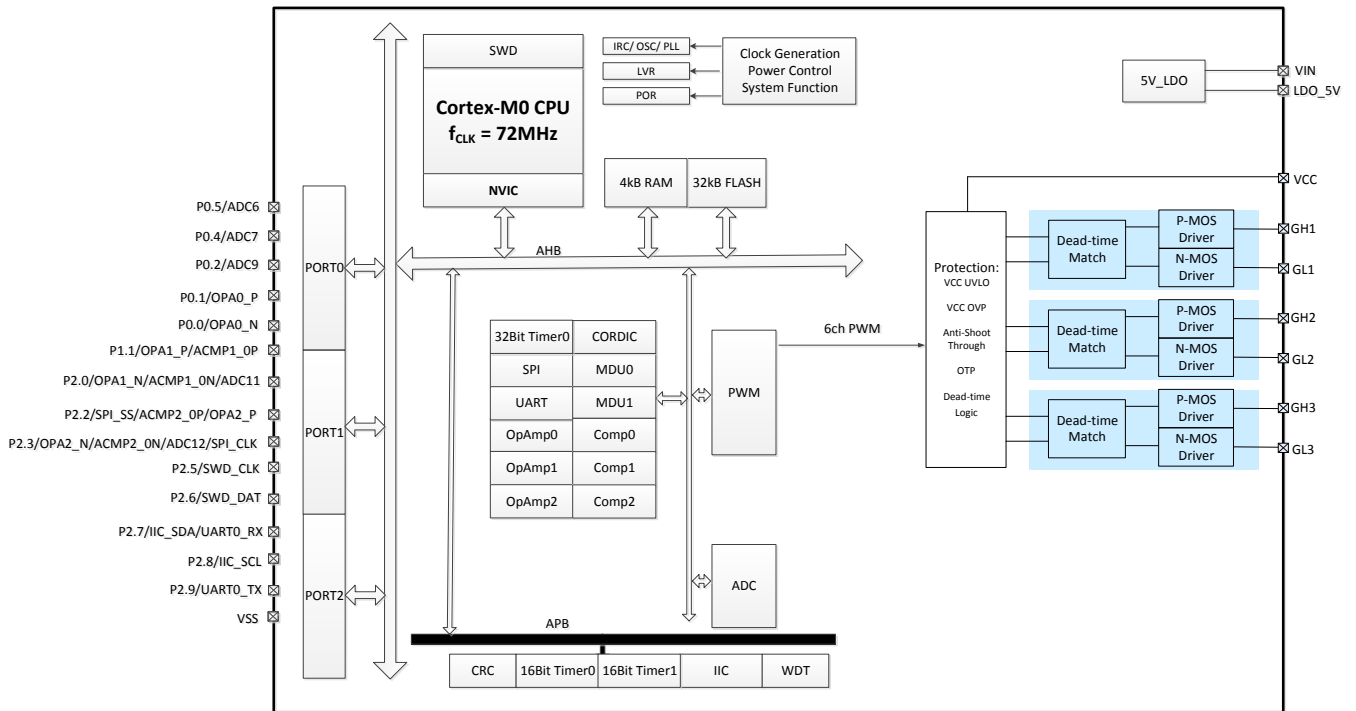
**引脚排布及说明 (QFN28)**


Pin Num			Pin Name	Description		
28	24	20				
4			P0.7/ADC4/ACMP0_1P/T16B0_CAP0	P0.7	I/O	General purpose digital input/output pin
				ADC4	I	ADC input channel 4
				ACMP0_1P	I	Analog comparator 0 positive Pin
				T16B0_CAP0	I	Capture input 0 for 16-bit timer 0
5			P0.6/ADC5/ACMP1_1P/T16B0_CAP1	P0.6	I/O	General purpose digital input/output pin
				ADC5	I	ADC input channel 5
				ACMP1_1P	I	Analog comparator 1 positive Pin
				T16B0_CAP1	I	Capture input 1 for 16-bit timer 0
6	8	7	P0.5/ADC6/ACMP2_1P/T16B0_CAP2	P0.5	I/O	General purpose digital input/output pin
				ADC6	I	ADC input channel 6
				ACMP2_1P	I	Analog comparator 2 positive Pin
				T16B0_CAP2	I	Capture input 2 for 16-bit timer 0
7	9	8	P0.4/ADC7/ACMP0/1/2_N/T16B1_CAP0	P0.4	I/O	General purpose digital input/output pin
				ADC7	I	ADC input channel 7
				ACMP0/1/2_N	I	Analog comparator 0/1/2 negative Pin
				T16B1_CAP0	I	Capture input 0 for 16-bit timer 1
8	10	9	P0.2/ADC9/T16B1_CAP2	P0.2	I/O	General purpose digital input/output pin
				ADC9	I	ADC input channel 9
				T16B1_CAP2	I	Capture input 2 for 16-bit timer 1
9	11	10	P0.1/OPA0_P/ACMP0_0P	P0.1	I/O	General purpose digital input/output pin
				OPA0_P	I	OPA 0 positive Pin.
				ACMP0_0P	I	Analog comparator 0 positive Pin
10	12	11	P0.0/ADC10/OPA0_N/ACMP0_0N	P0.0	I/O	General purpose digital input/output pin
				ADC10	I	ADC input channel 10
				OPA0_N	I	OPA 0 negative Pin
				ACMP0_0N	I	Analog comparator 0 negative Pin
11			P1.0/OPA0_O/ACMP0_O	P1.0	I/O	General purpose digital input/output pin
				OPA0_O	O	OPA 0 output Pin
				ACMP0_O	O	Analog comparator 0 output Pin
12	13		P1.1/OPA1_P/ACMP1_0P	P1.1	I/O	General purpose digital input/output pin
				OPA1_P	I	OPA 1 positive Pin
				ACMP1_0P	I	Analog comparator 1 positive Pin
13	14		P2.0/OPA1_N/ACMP1_0N/ADC11	P2.0	I/O	General purpose digital input/output pin
				OPA1_N	I	OPA 1 negative Pin
				ACMP1_0N	I	Analog comparator 1 negative Pin
				ADC11	I	ADC input channel 11
14			P2.1/OPA1_O/ACMP1_O/ SPI_MISO	P2.1	I/O	General purpose digital input/output pin
				OPA1_O	O	OPA 1 output Pin
				ACMP1_O	O	Analog comparator 1 output Pin
				SPI_MISO	I/O	Master In Slave Out for SPI

15	15		P2.2/SPI_SS/ACMP2_0P/OPA2_P	P2.2	I/O	General purpose digital input/output pin
				SPI_SS	O	Slave Select for SPI0.
				ACMP2_0P	I	Analog comparator 2 positive Pin
				OPA2_P	I	OPA 2 positive Pin
16	16		P2.3/OPA2_N/ACMP2_0N/ADC12/SPI_CLK	P2.3	I/O	General purpose digital input/output pin
				OPA2_N	I	OPA 2 negative Pin
				ACMP2_0N	I	Analog comparator 2 negative Pin
				ADC12	I	ADC input channel 12
17			P2.4/OPA2_O/ACMP2_0/SPI_MOSI	P2.4	I/O	General purpose digital input/output pin
				OPA2_O	O	OPA 2 output Pin
				ACMP2_0	O	Analog comparator 2 output Pin
				SPI_MOSI	I/O	Master Out Slave In for SPI
18	17	12	P2.5/SWD_CLK/UART0_TX/IIC0_SCL/T16B1_CAP2/ADC0	P2.5	I/O	General purpose digital input/output pin
				SWD_CLK	I/O	Serial wire debug input/output.
				UART0_TX	I/O	Transmitter output for UART 0.
				IIC0_SCL	I/O	I2C-bus clock
				T16B1_CAP2	I	Capture input 2 for 16-bit timer 1.
19	18	13	P2.6/SWD_DATA/CLK_OUT/UART0_RX/IIC0_SDA/T16B1_CAP1	P2.6	I/O	General purpose digital input/output pin
				SWD_DATA	I/O	Serial wire debug input/output.
				CLK_OUT	O	System frequency output pin
				UART0_RX	I/O	Receiver input for UART 0.
				IIC0_SDA	I/O	I2C-bus data
20	19	14	P2.7/UART0_RX/IIC0_SDA/FAULT_FLTB/SPI_SS	P2.7	I/O	General purpose digital input/output pin
				UART0_RX	I/O	Receiver input for UART 0.
				IIC0_SDA	I/O	I2C-bus data
				FAULT_FLTB	I	PWM input fault
				SPI_SS	I/O	Slave Select for SPI
21	20	15	P2.8/UART0_TX/IIC0_SCL/T16B1_CAP0/SPI_CLK	P2.6	I/O	General purpose digital input/output pin
				UART0_TX	I/O	Transmitter output for UART 0.
				IIC0_SCL	I/O	I2C-bus clock
				T16B1_CAP0	I	Capture input 0 for 16-bit timer 1.
				SPI_CLK	I/O	Serial clock for SPI
22	21	16	P2.9/UART0_TX/IIC0_SCL/SPI_MOSI/nReset	P2.9	I/O	General purpose digital input/output pin
				UART0_TX	I/O	Receiver input for UART 0.
				IIC0_SCL	I/O	I2C-bus clock
				SPI_MOSI	I/O	Master Out Slave In for SPI
				nReset	I	External reset input
23,25,27	22,24,2	17,19,1	GL3,GL2,GL1	GL3,GL2,GL1	O	低压侧栅极驱动器 3, 2, 1 输出
24,26,28	23,3,1	18,20,2	GH3,GH2,GH1	GH3,GH2,GH1	O	高压侧栅极驱动器 3, 2, 1 输出
1	4	3	VCC	VCC	P	工作电源电压输入
/	5	4	VIN	VIN	P	5VLDO 输入电压
3	7	6	LDO_5V	LDO_5V	P	5VLDO 输出电压; 与地间连接去耦电容

2	6	5	VSS	VSS	P	MCU Ground
---	---	---	-----	-----	---	------------

## 2. 功能方框图



### 3. 电气参数

 (如无特殊说明, 环境温度  $T_A = 25^\circ\text{C}$ ,  $V_{CC} = 18\text{V}$ )

符号	参数描述	测试条件	最小值	典型值	最大值	单位
<b>Analog Supply Voltage(<math>V_{CC}</math> Pin)</b>						
$I_{VCC}$	$V_{CC}$ quiescent supply current	$V_{CC} = V_{IN} = 18\text{V}$ , no load of output		2.7		mA
UVLO(OFF)	$V_{CC}$ under voltage lockout exit		4	4.5	5	V
UVLO(ON)	$V_{CC}$ under voltage lockout enter		3.5	4	4.5	V
OVP_on	VCC over voltage protection enter		38	39	40	V
OVP_off	VCC over voltage protection exit		37	38	39	V
<b>5V LDO (LDO_5V pin)</b>						
$I_{LDO-5V}$	LDO-5V max output current	$V_{IN} = 7\text{V} \sim 35\text{V}$	100			mA
$V_{LDO-5V}$	LDO output voltage	$V_{IN} = 7\text{V} \sim 35\text{V}$ $I_{LDO-5V} = 20\text{mA}$	4.9	5.0	5.1	V
$V_{LDO-5V}$	LDO output voltage	$V_{IN} = 5.2\text{V} \sim 7\text{V}$		5		V
$V_{LDO-5V}$	LDO output voltage	$V_{IN} \leq 5.2\text{V}$		$V_{IN} - 0.2$		V
<b>High Side P-MOSFET Gate Driver (GH1, GH2, GH3 pin)</b>						
$V_{GH} - V_{CC}$	$V_P - V_{CC}$ voltage after PWM on	PWM = 5V		-11		V
$V_{GH} - V_{CC}$	$V_P - V_{CC}$ voltage after PWM off	PWM = 0V		0		V
$I_{source}$	GH output short circuit peak current	CL = 10nF		330		mA
$I_{sink}$	GH input short circuit peak current	CL = 10nF		200		mA
$t_{on}$	GL Turn on propagation delay			120		ns
$t_{off}$	GL Turn off propagation delay			130		ns
<b>Low Side N-MOSFET Gate Driver (GL1, GL2, GL3 pin)</b>						
$V_{GL}$	$V_{GL}$ voltage after PWM on	PWM = 5V		11		V
$V_{GL}$	$V_{GL}$ voltage after PWM off	PWM = 0V		0		V
$I_{source}$	GL output short circuit peak current	CL = 10nF		400		mA
$I_{sink}$	GL input short circuit peak current	CL = 10nF		410		mA
$t_{on}$	GH Turn on propagation delay			190		ns
$t_{off}$	GH Turn off propagation delay			120		ns
<b>Deadtime</b>						
DT	Dead time			160		ns
<b>Over Temperature Protection</b>						
OTP(ON)	Over temperature protection enter			150		$^\circ\text{C}$
OTP(OFF)	Over temperature protection exit			130		$^\circ\text{C}$

#### 4. 特征曲线图

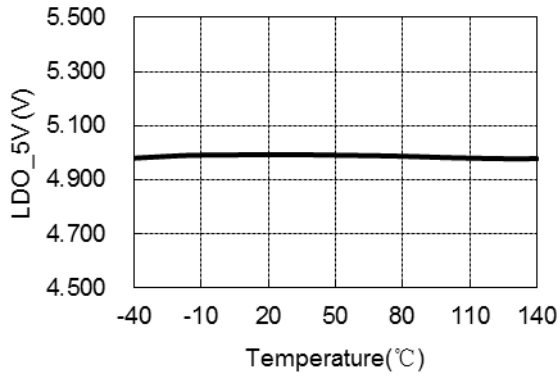
**LDO\_5V (V) vs Temperature**


Figure4.1 LDO\_5V vs Temperature

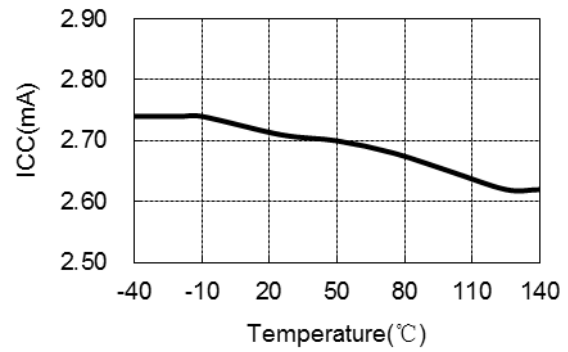
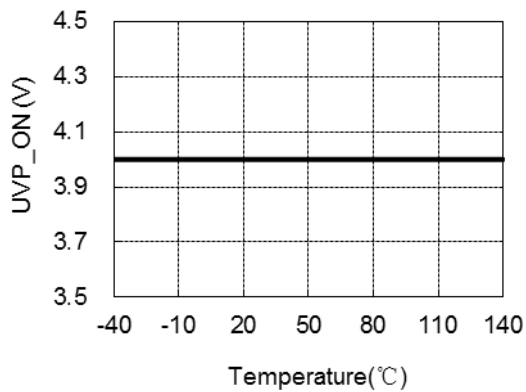
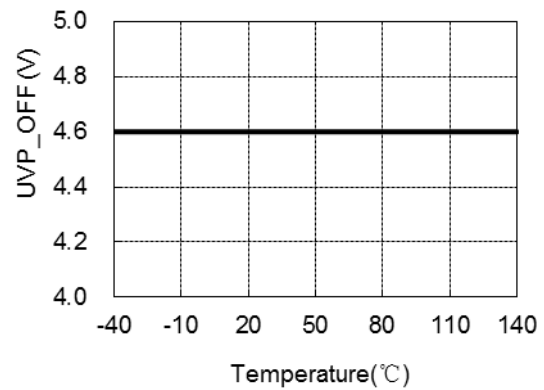
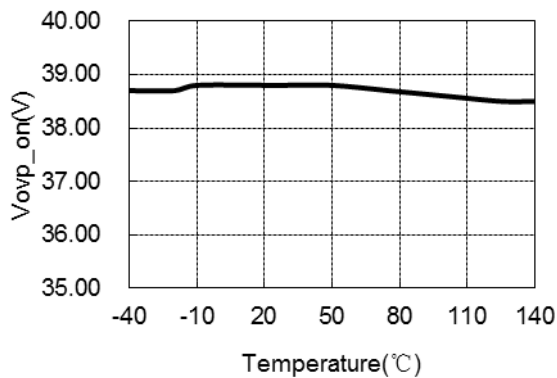
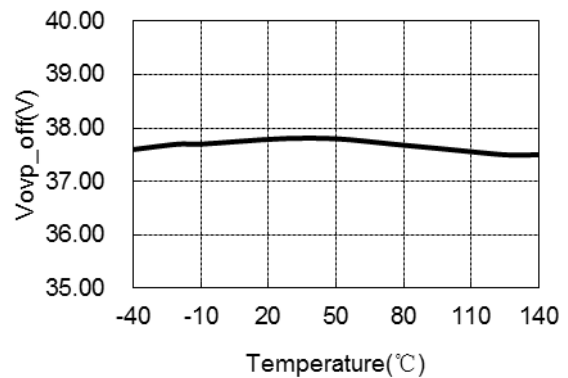
**ICC(mA) vs Temperature**


Figure4.2 Ivcc vs Temperature

**UVP\_ON (V) vs Temperature**

 Figure4.3 V<sub>CC</sub>\_UVP\_ON vs Temperature

**UVP\_OFF (V) vs Temperature**

 Figure4.4 V<sub>CC</sub>\_UVP\_OFF vs Temperature

**Vovp\_on(V) vs Temperature**

 Figure4.5 V<sub>CC</sub>\_OVP\_ON vs Temperature

**Vovp\_off(V) vs Temperature**

 Figure4.6 V<sub>CC</sub>\_OVP\_OFF vs Temperature

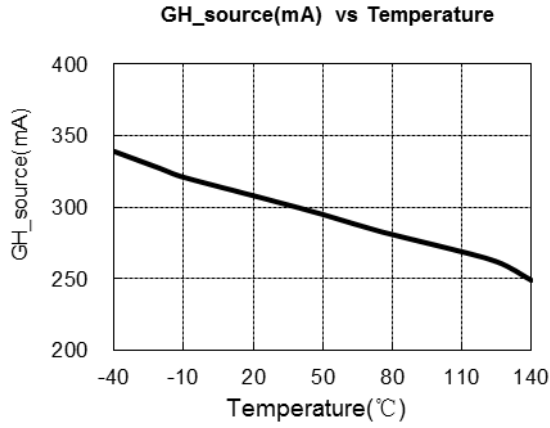


Figure4.7 GH\_source vs Temperature

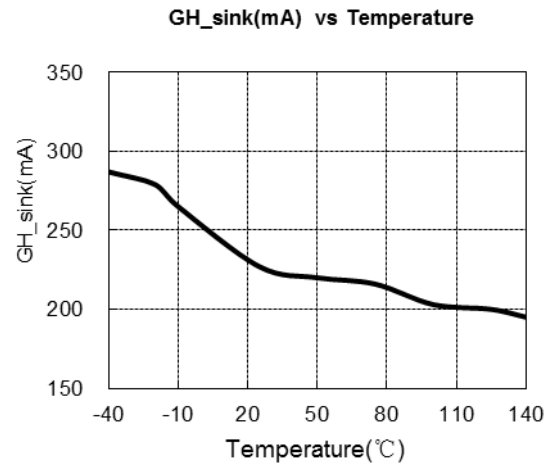


Figure4.8 GH\_sink vs Temperature

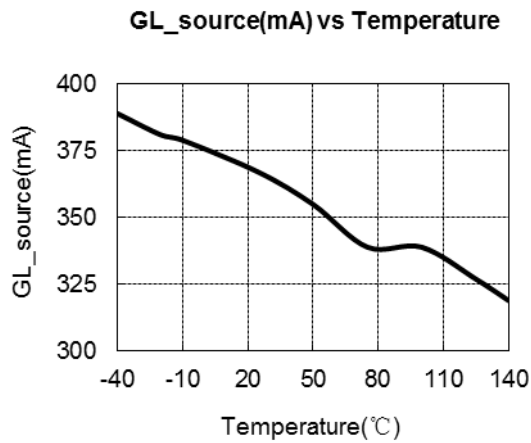


Figure4.9 GL\_source vs Temperature

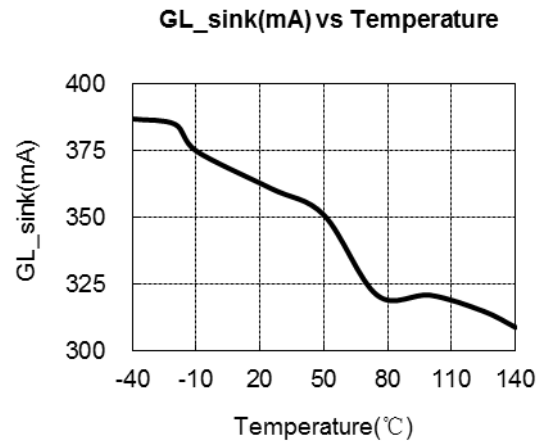


Figure4.10 GL\_sink vs Temperature

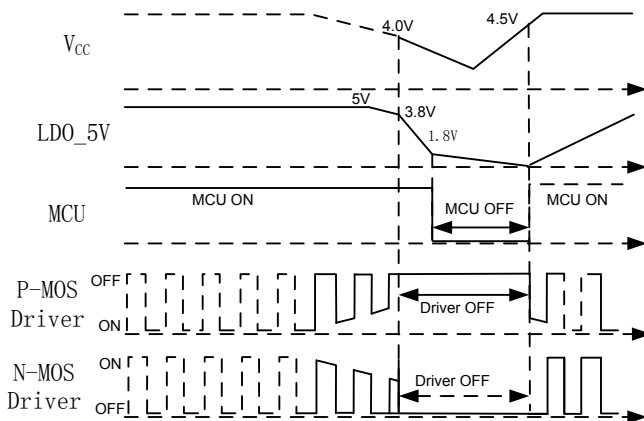


## 5. 工作模式描述

OB6625 是高集成度，高可靠性，低成本的 BLDC/PMSM 控制 IC。

### • Power Supply

当 VCC 电压增加到 4.5V（典型值）以上时，OB6625 开始工作，打开栅极驱动器，设置 LDO 输出。当 VCC 电压降至 4.0V（典型值）以下时，OB6625 将关闭栅极驱动器和 LDO。直到 VCC 电压升高到 4.5V（典型值）以上，OB6625 才会恢复工作。



### • Low-Side Driver

低压侧驱动器设计用于驱动连接参考地增强型 N 沟道 MOSFET。其低的  $R_{ds(on)}$  允许快速打开和关闭外部 MOSFET。

### • High-Side Driver

高压侧驱动器设计用于驱动（P-MOSFET），其源极端子参考直流电源。

### • OTP

当结温升至典型的 150°C 时，将触发内部过热保护，并关闭 6 个驱动器，并使 PWM 输入变为空白。

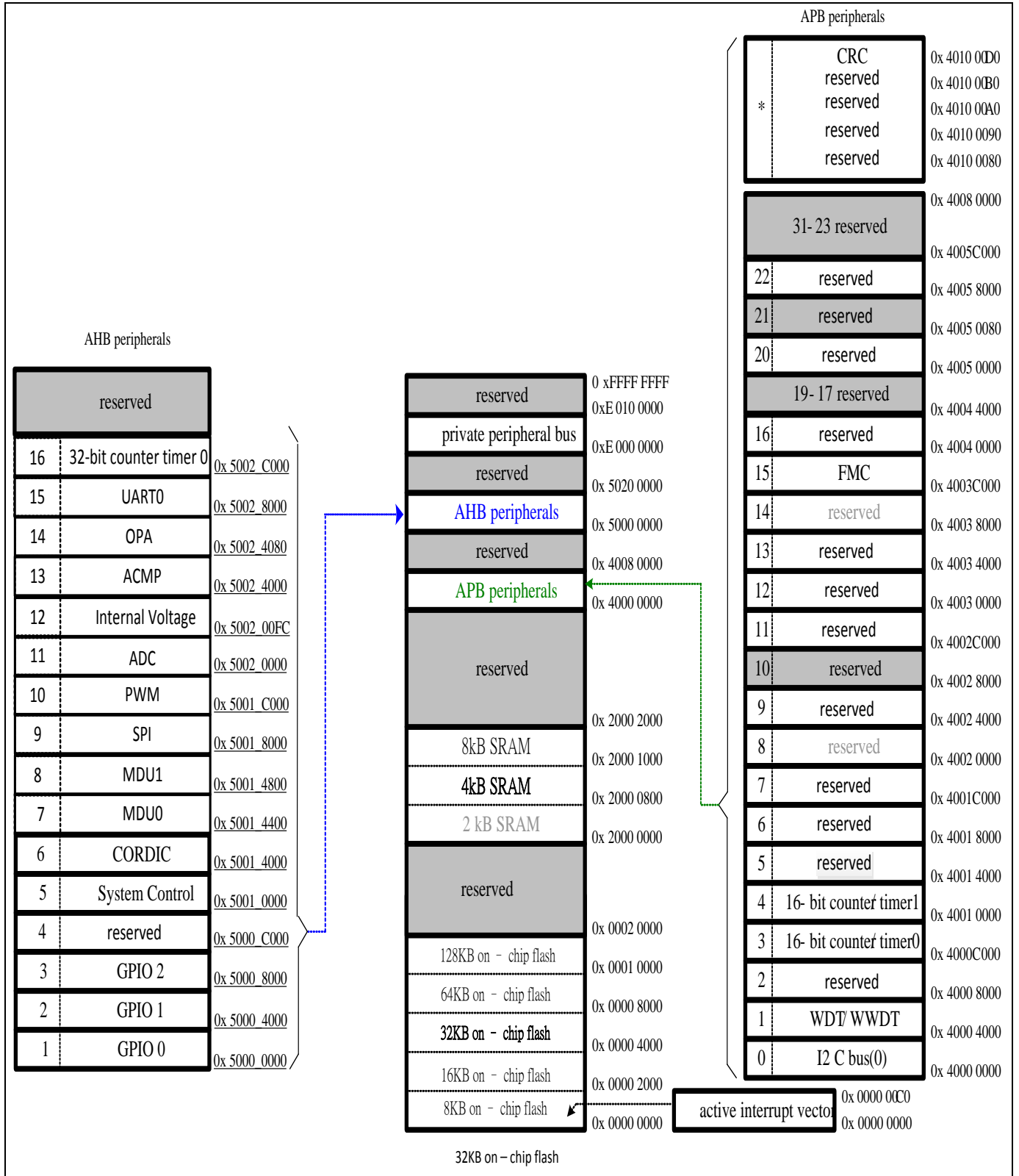
当 OB6625 结温超过 150°C，会发生过温保护，六路驱动器关闭输出。当芯片结温低于 130°C 后，过温保护解除，六路驱动器重新打开

### • OVP

当 VCC 电压增加到高于 39V（典型值）时，OB6625 将关闭 6 个驱动器。当 VCC 电压降至 38V（典型值）以下时，OB6625 释放 OVP 保护，并恢复对 6 个驱动器的控制。

### 6. 系统配置

#### 6.1 存储器映射



## 6.2 寄存器说明

表 6.2.1 寄存器简介: 系统(基址 0x5001 0000)

名称	访问类型	地址偏移	描述	重置值
-	-	0x000 - 0x00F	保留	0
CCLKCFG	R/W	0x010	CPU时钟配置寄存器	0
-	-	0x011 - 0x017	保留	0
PCLKSEL0	R/W	0x018	外设时钟选择寄存器0	0
PCLKSEL1	R/W	0x01C	外设时钟选择寄存器1	0
PCGP	R/W	0x020	电源/时钟门控外设寄存器	0
-	-	0x024 - 0x07C	保留	0
DPLLCON	R/W	0x080	DPLL控制寄存器	0
-	-	0x099 - 0x0AF	保留	0
CPUCHIPCTR0	R/W	0x0B0	CPU / CHIP控制0寄存器	0
-	-	0x0B1 - 0x0BF	保留	0
LVRLVICTRSTA0	R/W	0x0C0	LVR / LVI控制/状态0寄存器	0
USRFLSHCTR0	R/W	0x0D0	USR/Flash控制0寄存器	0
-	-	0x0D1 - 0x0FF	保留	0

## 6.3 重置控制器

### 6.3.1 重置源和框图

系统重置可以由以下列出的事件之一发出。

- 上电重置
- 低电平有效/ 重置按键
- 低电压重置
- 系统重置（软件重置）
- 窗口看门狗超时重置（WWDT 重置）

### 6.3.2 重置注释项目

表 6.3.2.1 重置注释项目说明

名称	注释项目
/RESET PAD	不要重置 1. LVR校准寄存器(FMC)
Low Voltage Reset (LVR)	不要重置 1. LVR校准寄存器(FMC)
System Reset	不要重置 1. LVR校准寄存器(FMC) 2. 引导选择寄存器和ISP进入机制(FMC) 3. 窗口看门狗定时器(WWDT) 4. CPU Clock Select
WWDT Reset	不要重置 1. LVR校准寄存器(FMC) 2. 引导选择寄存器和ISP进入机制(FMC) 3. 窗口看门狗定时器(WWDT)

## 6.4 时钟控制器

### 6.4.1 概述

时钟控制器为整个芯片生成时钟，包括系统时钟和所有外设时钟。时钟控制器还通过单独的时钟 ON / OFF 控制和时钟源选择实现功率控制功能。在 CPU 设置深度睡眠使能位且 Cortex-M0 内核执行 WFI 或 WFE 指令之前，该芯片不会进入深度睡眠模式。之后，芯片进入深度睡眠模式并等待唤醒中断源被触发以退出深度睡眠模式。在深度睡眠模式下，时钟控制器关闭外部 4~24 MHz 晶振和内部 24 MHz 振荡器，以降低整个系统的功耗。

### 6.4.2 时钟发生器

时钟发生器由 3 个时钟源组成，如下所示：

- 一个可编程 PLL 输出（PLL 源由外部 4~24 MHz 晶振和内部 24 MHz 振荡器组成）
- 一个内部 24 MHz 振荡器(HIRC)
- 一个内部 20 KHz 振荡器(LIRC)

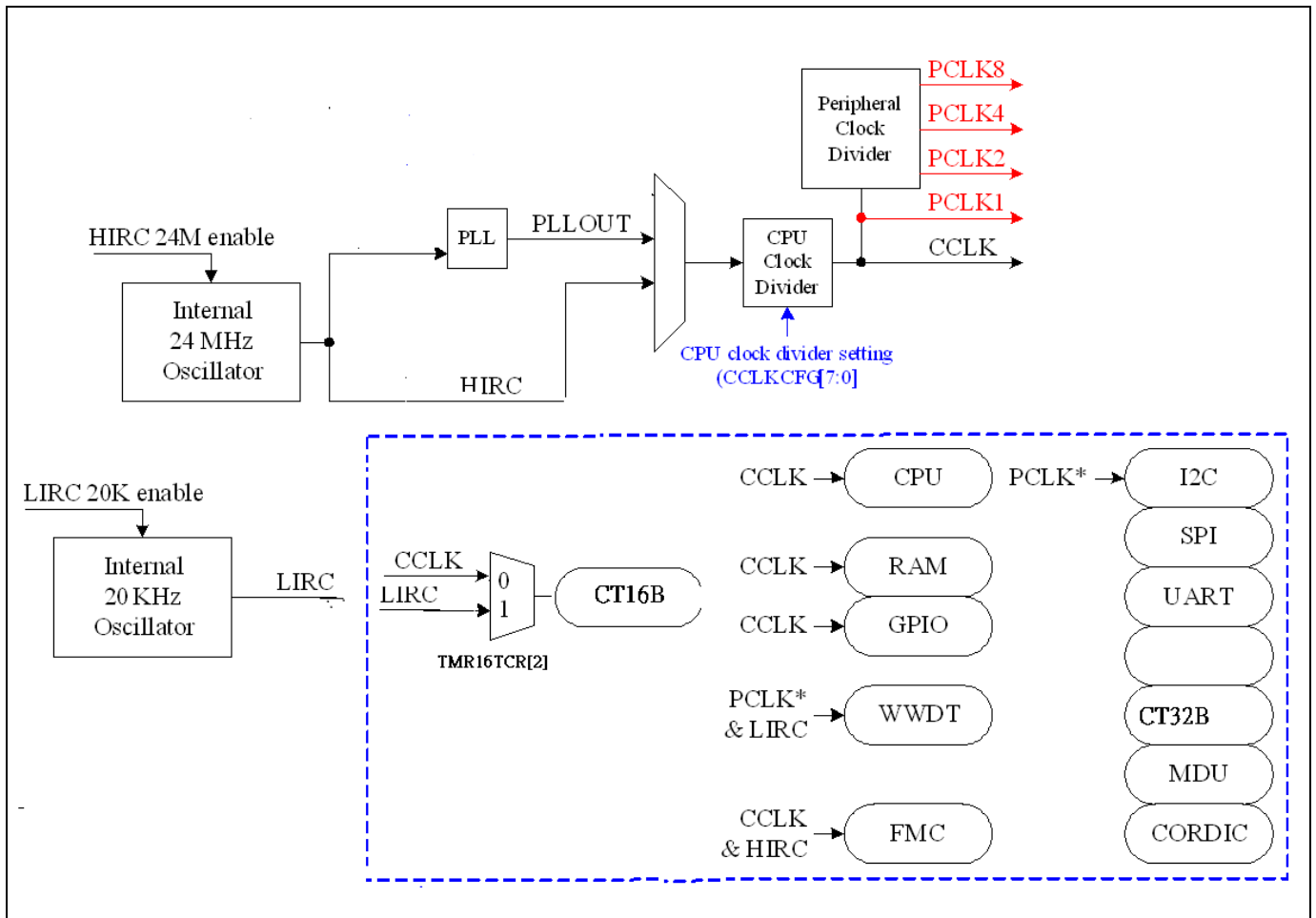


图 6.4.2.1 时钟控制器框图

### 6.4.3 CPU 时钟配置寄存器(CCLKCFG – 0x5001 0010)

CCLKCFG 寄存器控制多路复用器输出 (EXTL / PLLOUT / HIRC) 在被 CPU 使用之前的分频。当 PLL 运行时，必须对输出进行分频，以便将 CPU 时钟频率 (CCLK) 限制在操作范围内。一个 8 位分频器允许一系列选项，包括将 CPU 操作速度减慢至低速率以实现临时节电，而无需关闭 PLL。

表 6.4.3.1 CPU 时钟配置寄存器(CCLKCFG - 0x5001 0010) 位描述

位	符号	值	描述	重置值
7:0	CCLKSEL		从多路复用器输出 (EXTL/PLLOUT/HIRC) 中选择创建 CPU 时钟 (CCLK) 的分频值。	
		0	时钟被 1 分频以产生 CPU 时钟。 注意：速率总是会大于允许的最大 CPU 时钟。	
		1	时钟被 2 分频以产生 CPU 时钟。 注意：速率总是会大于允许的最大 CPU 时钟。	
		2	时钟被 3 分频以产生 CPU 时钟。	
		3	时钟被 4 分频以产生 CPU 时钟。	
		..	..	
		255	时钟被 256 分频以产生 CPU 时钟。	
9:8	MCUSRCSEL		MCU 时钟源选择。	
		0	HIRC 时钟源 (内部时钟)	
		1	PLL 时钟源	
11:10	CPUWAITCNT		CPU 等待计数。 限于闪存数据读取访问时间。如果 CPU 时钟周期小于访问时间，则用户设置等待计数器读取 ROM 核心。	
		0	0 周期(系统时钟介於 0MHz~24MHz)	
		1	1 周期(系统时钟介於 25MHz~48MHz)	
		2	2 周期(系统时钟介於 49MHz~72MHz)	
15:12 -	-	-	保留。用户软件不应向保留位写入 1。未定义从保留位读取的值。	
16	DPLL_ADCE		ADC 时钟源选择	0
		0	系统时钟	
		1	DPLL_ADC (必须与不等于 0 的 ADC0CD 寄存器配合使用)	
30:17 -	-	-	保留。用户软件不应向保留位写入 1。未定义从保留位读取的值。	
31	sysClkProbeE		系统时钟探测使能	0
		0	禁用	
		1	系统时钟探测来自 SWD_DATA 引脚	

CCLK 来自多路复用器输出信号，除以 CCLKSEL + 1。由于 CCLKSEL = 2 导致 CCLK 为多路复用器输出的三分之一，CCLKSEL = 3 导致 CCLK 为多路复用器输出的四分之一。

### 6.4.4 电源/时钟门控外设寄存器(PCGP – 0x5001 0020)

PCGP 寄存器允许关闭选定的外设功能以节省功耗。这是通过关闭指定外设模块的时钟源来完成的。一些外设功能不能关闭 (即看门狗定时器，引脚连接模块和系统控制模块)。

一些外设，特别是那些包含模拟功能的外设，可能会消耗不依赖时钟的功耗。这些外设可能包含一个单独的禁用控制，关闭附加电路以降低功耗。

PCGP 中的每个位控制一个外设，如表 6.4.5.1 所示。

如果外设控制位为 1，则该外设已启用。如果外设控制位为 0，那么该外设的时钟禁用 (门控关闭) 以节省功耗。

重要提示：只有在 PCGP 寄存器中启用了外设时，才能从外设寄存器进行有效读操作，并有效写入外设寄存器。

表 6.4.5.1 电源/时钟门控外设寄存器 (PCGP - 0x5001 0020) 位描述

位	符号	描述	重置值
0	PCSPI	SPI 接口电源/时钟控制位	
1	PC_CORDIC	CORDIC 电源/时钟控制位	
2	-	保留	
3	-	保留	

4	-	保留
5	PC_MDU	MDU 电源/时钟控制位
6	PC_ADC0	A/D 转换器 (ADC0) 电源/时钟控制位
7	-	保留
8	PC_UART_0	UART 0 电源/时钟控制位
9	PC_WWDT	窗口看门狗定时器电源/时钟控制位
10	PC_I2C_0	I2C 接口电源/时钟控制位
11	PC_UART_1	UART 1 电源/时钟控制位
12	PC_PWM	始终设为 1。
13	PC_I2C_1	I2C 接口电源/时钟控制位
14	PC_T16B0	计数/定时器 16 位 0 电源/时钟控制位
15	PC_T16B1	计数/定时器 16 位 1 电源/时钟控制位
16	PC_T32B0	计数/定时器 32 位 0 电源/时钟控制位
31:17	-	保留

#### 6.4.5 功率控制使用说明

每次重置后，PCGP 寄存器都包含使 PCGP 控制的选定接口和外设被启用的值。因此，除了通过外设专用寄存器进行正确配置之外，用户的应用程序可能必须访问 PCGP 才能开始使用某些板载外设。

面向节能的系统只能在 PCGP 寄存器中有 1 个匹配应用中真正使用的外设的位置。声明为“保留”或专用于当前应用中未使用的外设的所有其他位必须清零。

#### 6.4.6 DPLL 控制寄存器(DPLLCON – 0x5001 0080)

表 6.4.7.1 DPLL 控制寄存器(DPLLCON - 0x5001 0080) 位描述

位	符号	值	描述	重置值
0	PLLSRCSEL		DPLL 时钟源选择 用户软件应向此位写入 1。	0
		1	HIRC 时钟源 (内部时钟)	
3:1	-	-	保留, 用户软件不应向保留位写入 1。未定义从保留位读取的值。	0
7:4	CREF_DIV_SEL		DPLL 参考除频时钟	0
		0	DPLL CREF 不要除	
		1	DPLL CREF 除 2 (min)	
		..	..	
		15	DPLL CREF 除 16 (max)	
13:8	DPLL_DIV_SEL		DPLL(DPLL_FREQ_SEL[5:0]) 除频时钟	0
		0	DPLL(DPLL_FREQ_SEL [5:0]) 除 2	
		1	DPLL(DPLL_FREQ_SEL [5:0]) 除 2	
		2	DPLL(DPLL_FREQ_SEL [5:0]) 除 2	
		3	DPLL(DPLL_FREQ_SEL [5:0]) 除 3	
		4	DPLL(DPLL_FREQ_SEL [5:0]) 除 4	
		..	..	
		63	DPLL(DPLL_FREQ_SEL [5:0]) 除 63	
15:14	-	-	保留, 用户软件不应向保留位写入 1。未定义从保留位读取的值。	0
21:16	DPLL_FREQ_SEL		DPLL 频率选择 用户软件应向此位写入 0x02	0
		0x02	DPLL 频率= 144 MHz	
23:22	-	-	保留, 用户软件不应向保留位写入 1。未定义从保留位读取的值。	
27:24	A_IN	0	总是写 0。	
31:28	B_IN	F	总是写 F。	

--



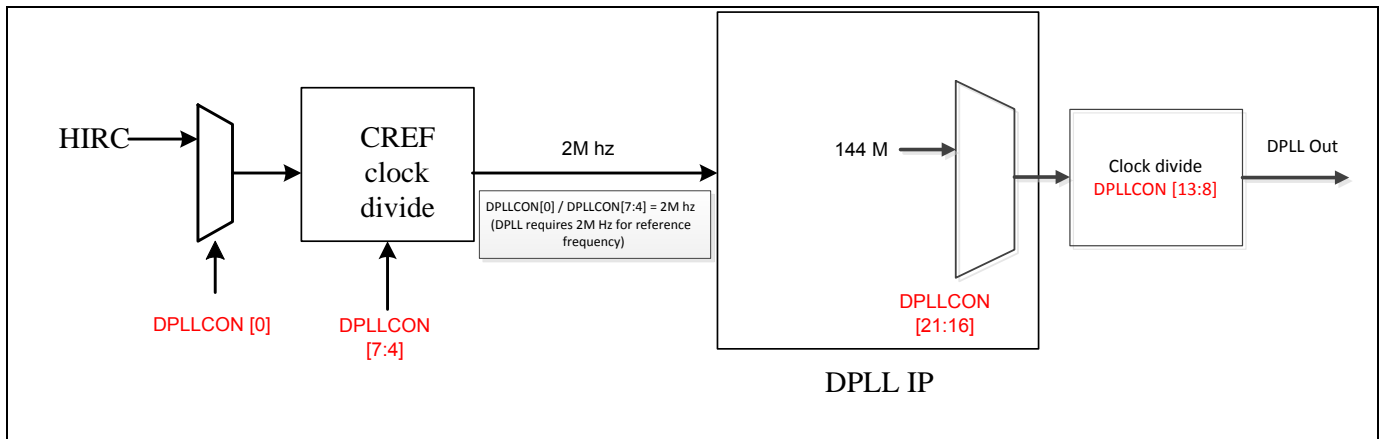


图 6.4.7.1 DPLL 图

### 6.4.7 DPLL2 控制寄存器(DPLL2CON – 0x5001 0084)

表 6.4.8.1 DPLL2 控制寄存器(DPLL2CON - 0x5001 0084) 位描述

位	符号	值	描述	重置值
7:0			保留	0
10:8	ADC_DIV_SEL		DPLL(DPLL_FREQ_SEL[5:0]) 除频时钟	0
		0	DPLL(DPLL_FREQ_SEL [5:0]) 除 2 for ADC	
		1	DPLL(DPLL_FREQ_SEL [5:0]) 除 2 for ADC	
		2	DPLL(DPLL_FREQ_SEL [5:0]) 除 2 for ADC	
		3	DPLL(DPLL_FREQ_SEL [5:0]) 除 3 for ADC	
		4	DPLL(DPLL_FREQ_SEL [5:0]) 除 4 for ADC	
		5	DPLL(DPLL_FREQ_SEL [5:0]) 除 5 for ADC	
		6	DPLL(DPLL_FREQ_SEL [5:0]) 除 6 for ADC	
		7	DPLL(DPLL_FREQ_SEL [5:0]) 除 7 for ADC	
31:11	-	-	保留	0

### 6.4.8 DPLL 调试模式寄存器(DPLLDM – 0x5001 0090)

Table 6.4.7.1 DPLL 调试模式寄存器(DPLLDM - 0x5001 0090) 位描述

位	符号	值	描述	重置值
0	DPLLDDB		DCO_EN_SFR (微调)	0
		0	硬件	
		1	软件	
			(参考 DPLLFT)	
1	DPLLBT		OSC_CAL_EN_SFR (粗调)	0
		0	硬件	
		1	软件	
			(参考 DPLLCT)	
2	CAL_EN_SFR		CAL_EN_SFR	0
		0	(默认)	
		1		
3	INI_EN_SFR		INI_EN_SFR	0
		0	(默认)	
		1		
7:4			保留	0
8	STA		LEAD_LAG	0
15:9			保留	0
21:16	CAL_SFR[5:0]		CAL_SFR[5:0]	0
23:22			保留	0

28:24	INI_SFR[4:0]	INI_SFR [4:0]	0
31:29	-	保留	0

#### 6.4.9 CPU / CHIP 控制 0 寄存器(CPUCHIPCTR0 – 0x5001 00B0)

表 6.4.12.1 CPU/CHIP 控制 0 寄存器 (CPUCHIPCTR0 - 0x5001 00B0) 位描述

位	符号	值	描述	重置值
0	-	-	保留	0
1	PADE_SWD	-	引脚使能 SWD (SWD: 串行线调试) SWD_DATA / SWD_CLK (芯片引脚) 可以设置为两种模式。GPIO 或串行线调试功能。	1
		0	GPIO 引脚功能。(由 GPIO2-6 控制 SWD_DATA, 由 GPIO2-5 控制 SWD_CLK)	
		1	串行线调试功能。(默认)	
2	-	-	保留	0
3	PADE_MXTAL	-	引脚使能 MHz XTAL MXTAL_IN / MXTAL_OUT (芯片引脚) 可以设置为两种模式, 即 GPIO 或 MHz XTAL 功能。	0
		0	GPIO 引脚功能。(通过 GPIO2-9 控制 MXTAL_IN, 通过 GPIO2-10 控制 MXTAL_OUT)	
		1	MHz XTAL 功能。	
4	XTGS	-	XTAL 增益选择。(对于 MXTAL 功能, 默认 1)	1
11:5	-	-	保留。用户软件不应向保留位写入 1。未定义从保留位读取的值。	
15:12	DPLL_SEL	-	DPLL 时钟稳定计数选择 (取决于 LIRC = 20K 时钟)	1
		0x0	LIRC 时钟稳定计数 = 2 <sup>0</sup> 周期	
		0x1	LIRC 时钟稳定计数 = 2 <sup>1</sup> 周期	
		0x2	LIRC 时钟稳定计数 = 2 <sup>2</sup> 周期	
		0x3	LIRC 时钟稳定计数 = 2 <sup>3</sup> 周期	
		0x4	LIRC 时钟稳定计数 = 2 <sup>4</sup> 周期	
		0x5	LIRC 时钟稳定计数 = 2 <sup>5</sup> 周期	
		0x6	LIRC 时钟稳定计数 = 2 <sup>6</sup> 周期	
		0x7	LIRC 时钟稳定计数 = 2 <sup>7</sup> 周期	
		0x8	LIRC 时钟稳定计数 = 2 <sup>8</sup> 周期	
		0x9	LIRC 时钟稳定计数 = 2 <sup>9</sup> 周期	
		0xA	LIRC 时钟稳定计数 = 2 <sup>10</sup> 周期	
		0xF	时刻准备着	
19:16	MXTAL_SEL	-	MXTAL 时钟稳定计数 (取决于 MXTAL 时钟)	4
		0x0	MXTAL 时钟稳定计数 = 2 <sup>7</sup> 周期	
		0x1	MXTAL 时钟稳定计数 = 2 <sup>8</sup> 周期	
		0x2	MXTAL 时钟稳定计数 = 2 <sup>9</sup> 周期	
		0x3	MXTAL 时钟稳定计数 = 2 <sup>10</sup> 周期	
		0x4	MXTAL 时钟稳定计数 = 2 <sup>11</sup> 周期 (默认)	
		0x5	MXTAL 时钟稳定计数 = 2 <sup>12</sup> 周期	
		0x6	MXTAL 时钟稳定计数 = 2 <sup>13</sup> 周期	
		0x7	MXTAL 时钟稳定计数 = 2 <sup>14</sup> 周期	
		0xF	时刻准备着	
20	KXTAL_RDY	-	1: 准备好, 0: 未准备好	
21	MXTAL_RDY	-	1: 准备好, 0: 未准备好	
22	DPLL_RDY	-	1: 准备好, 0: 未准备好	
23	HIRC_RDY	-	1: 准备好, 0: 未准备好	
24	LIRC_RDY	-	1: 准备好, 0: 未准备好	
31:25	-	-	保留	

#### 6.4.10 LVR / LVI 控制/状态 0 寄存器(LVRLVICRSTA0 – 0x5001 00C0)

表 6.4.14.1 LVR/LVI 控制/状态 0 寄存器 (LVRLVICRSTA0 - 0x5001 00C0) 位描述

位	符号	值	描述	重置值
0	PD_LVR		禁用 LVR 功能	0
		0	使能 LVR 功能。(默认)	
		1	禁用 LVR 功能。	
1	PD_LVR_LP		禁用 LVR_LP 功能。(LVR_LP : LVR 低功耗)	0
		0	使能 LVR_LP 功能。(默认)	
		1	禁用 LVR_LP 功能。	
2	PD_LVRLP_INT		禁用 LVRLP_INT 功能。(LVRLP_INT : 内部低功耗)	0
		0	使能 LVRLP_INT 功能。(默认)	
		1	禁用 LVRLP_INT 功能。	
3	PD_LVI		禁用 LVI 功能。	0
		0	使能 LVI 功能。(默认)	
		1	禁用 LVI 功能。	
5:4	LVI_SEL		LVI 选择	0
		0	1.65V(默认)	
		1	2.6V	
		2	3.15V	
6	LVRRSTE		LVR 重置使能	0
		0	LVR 重置功能禁用。	
		1	LVR 重置功能使能。(默认)	
7	LVRLPRSTE		LVRLP 重置使能	0
		0	LVRLP 重置功能禁用。	
		1	LVRLP 重置功能使能。(默认)	
15:8	-	-	保留	0
16	LVR		模拟状态。(高电平动作)	
17	LVR_LP		模拟状态。(高电平动作)	
18	LVRLP_INT		模拟状态。(高电平动作)	
19	LVI		模拟状态。(高电平动作)	
20	LVIINT		LVI 中断	0
		0	软件写 1 清除标志。	
		1	硬件置高。	
21	-	-	保留	0
22	LVRRSTFIAG		LVR 重置标志	0
		0	软件写 1 清除标志。	
		1	硬件置高。	
23	LVRLPRSTFLAG		LVRLP 重置标志	0
		0	软件写 1 清除标志。	
		1	硬件置高。	
31:24	-	-	保留	0

#### 6.4.11 USR/Flash 控制 0 寄存器(USRFLSHCTR0 – 0x5001 00D0)

Table 6.4.15.1 USR/Flash 控制 0 寄存器(USRFLASHCTR0 - 0x5001 00D0) 位描述

位	符号	值	描述	重置值
15:0	-	-	保留。	0
23:16	PADE_RESET		Pad Enable reset. RESET (chip pad) can be set in two modes. GPIO or RESET function. Flash data = 0x5A, or firmware write data = 0x5A. <b>GPIO function</b> enabled. (write bit23 : bit16)	
24	EQADDR		保留	1
25	BRANCH			1
		0	禁用	
		1	使能	
26	PREDICT			1

	0	禁用	
	1	使能	
27	PREFETCH_BUFF	保留	1
28	PREFETCH_MODE	保留	1
29	Filter		0
	0	禁用	
	1	使能	
30:29 -	-	保留	0
31	MCUACC_PBUFEN	1. Reload from flash ( value inverter ) 2. FW write ( no inverter )	0
	0	禁用	
	1	使能	

USR/Flash Control Register reload from User Option (5 word, Address 0x\*\*\*\*\_\*\*\*\*)

## 6.5 功率控制器

OB6625 支持各种功耗控制功能：睡眠模式，深度睡眠模式。通过更改时钟源，重新配置 PLL 值和/或更改 CPU 时钟分频器值，CPU 时钟频率也可以根据需要进行控制。这允许根据应用需求权衡取舍功率与处理速度。此外，外设功耗控制允许关闭单个片上外设的时钟，通过消除应用中不需要的任何外设中的所有动态功耗，可以精细调整功耗。任何降低功耗模式的入口都是从 Cortex-M0 执行 WFI（等待中断）或 WFE（等待异常）指令开始。Cortex-M0 内部支持两种降低功耗模式：睡眠和深度睡眠。这些由 Cortex-M0 系统控制寄存器中的 SLEEPDEEP 位选择。PCON 寄存器包含的标志表明是否进入每个降低功耗模式实际发生。

### 6.5.1 睡眠模式

当进入睡眠模式时，核心时钟停止，并且 PCON 中的 SMFLAG 位被硬件置位为 1。从睡眠模式恢复不需要任何特殊的时序，但可以重新启用 ARM 内核的时钟。

在睡眠模式下，指令的执行暂停，直到发生重置或中断。外设功能在睡眠模式下继续工作，并可能产生中断以使处理器恢复执行。睡眠模式消除了处理器本身，内存系统和相关控制器以及内部总线所使用的动态功耗。

只要有任何使能中断发生，就会从睡眠模式唤醒。

### 6.5.2 深度睡眠模式

当芯片进入深度睡眠模式时，HIRC 关断，几乎所有时钟停止，并且 PCON 中的 DSMFLAG 位被硬件置位为 1。LIRC 保持运行，可配置为驱动看门狗定时器，使看门狗唤醒 CPU。闪光灯处于待机模式，可以快速唤醒。PLL 会自动关闭并断开连接。CPU 时钟分频器自动重置为零。

处理器状态和寄存器，外设寄存器以及内部 SRAM 值在整个深度睡眠模式下保持不变，并且芯片引脚的逻辑电平保持静态。深度睡眠模式可以被终止，正常操作可以通过重置或某些特定的中断来恢复，这些中断可以在没有时钟的情况下工作。由于芯片的所有动态操作都暂停，因此深度睡眠模式可将芯片功耗降至非常低的值。

当相关中断使能时，从深度睡眠模式唤醒可引发 NMI，外部中断 EINT0 至 EINT3，GPIO 中断，看门狗定时器超时。每当有任何使能中断发生时，唤醒都会发生。

### 6.5.3 功率模式控制寄存器(PCON – 0x5001 0024)

PCON 寄存器包含一些降低功耗模式和其他与功耗相关的控制。

表 6.5.3.1 功率模式控制寄存器(PCON - 0x5001 0024) 位描述

位	符号	描述	重置值
3:0	-	保留	0
4	SMFLAG	睡眠模式进入标志。当成功进入睡眠模式时硬件将该位置位为 1。 需通过软件向该位写入 1 由将其清零。	0
5	DSMFLAG	深度睡眠进入标志。当成功进入深度睡眠模式时硬件将该位置位为 1。 需通过软件向该位写入 1 由将其清零。	0
31:6	-	保留	0

### 6.5.4 从低功耗模式中唤醒

任何使能的中断均可将 CPU 从睡眠模式中唤醒。某些特定的中断可将处理器从深度睡眠模式中唤醒。

若特定的中断使能则允许中断将 CPU 从深度睡眠模式中唤醒。唤醒后，将继续执行适当的中断服务程序。这些中断为 NMI、外部中断 EINT0 到 EINT3、GPIO 中断。此外，如果看门狗定时器由 LIRC 振荡器驱动，则看门狗定时器也可将器件从深度睡眠模式中唤醒。为了进行唤醒过程，必须在 NVIC 中启用相应的中断。对于引脚相关的外设功能，相关功能也必须映像到引脚。

这些中断仅用于在处于深度睡眠模式时唤醒 CPU，当外设功能加电但未激活时。通常情况下，如果使用这些中断，则在启用中断并进入所需的降低功耗模式之前应调查它们的标志。这可以通过避免立即唤醒来节省时间和电力。唤醒后，中断服务可以关闭相关的活动中断，执行任何应用程序特定的设置，然后退出以等待正常的外设中断。

## 7. 嵌套向量中断控制器(NVIC)

### 7.1描述

嵌套向量中断控制器（NVIC）是 Cortex-M0 的一个内部器件。与 CPU 紧密结合，降低了中断延时，让新进中断可以得到高效处理。

### 7.2特性

- ARM Cortex-M0内部包含有嵌套向量中断控制器
- 与内核紧密联系的中断控制器，可支持低中断延时
- 可对系统异常和外设中断进行控制
- NVIC支持向量中断
- 4个可编程的中断优先级
- 软件中断功能

### 7.3中断源

表 7.3.1 列出了每一个外设功能所对应的中断源。每一个外围设备可以有一条或几条中断线连接到向量中断控制器。多个中断源也可以共享一条中断线。除过 ARM 内核指定的某些标准外，哪一条中断线连接到哪一个中断源是无关紧要的或没有优先级的。

表 7.3.1 连接到向量中断控制器（VIC）的中断源

例外号码	向量偏移量	功能	标志
0	-	保留	保留
1	-	保留	保留
2	-	LVI interrupt	LVIInt
3	-	ACMP	模拟比较器
4	-	保留	保留
5	-	I2C0	SI (状态改变)
6	-	T16B0	匹配0-2 捕获0
7	-	T16B1	匹配0-1 捕获0
8	-	T32B0	匹配0-3 捕获0
9	-	保留	保留
10	-	SPI/SSP0	TX FIFO半为空 RX FIFO半为满 RX超时 RX溢出
11	-	UART0	Rx线状态(RLS) 发送保持寄存器空(THRE) Rx数据可用(RDA) 字符超时指示器(CTI) 自动波特率结束(ABEO) 自动波特率超时(ABTO)
12	-	保留	保留

13	-	保留	保留
14	-	PWM	电源控制PWM
15	-	WWDT	窗口看门狗
16	-	ADC	ADC控制
17	-	保留	保留
18	-	GPIO0	端口0的GPIO中断状态
19	-	GPIO1	端口1的GPIO中断状态
20	-	GPIO2	端口2的GPIO中断状态
21	-	GPIO3	端口3的GPIO中断状态
22		PWMTM	PWM定时器
31-23	-	保留	保留



## 8. I2C-总线控制

### 8.1 特性

- 标准I2C的总线接口，可配置为主机、从机或者主/从机
- 同时发送的主机之间进行仲裁，避免了串行总线数据的冲突
- 可编程时钟能够实现I2C传输速率控制
- 主、从机之间双向数据传输
- 串行时钟同步允许不同位速率的器件通过同一个串行总线通信
- 串行时钟同步可作为一个握手机制来挂起和恢复串行传输
- 支持快速模式Plus
- 可识别多达2个不同的从机地址
- I2C总线可用作测试和诊断
- I2C 总线包含一个具有两个引脚的标准 I2C 兼容总线接口

### 8.2 应用

与外部 I2C 标准部件连接，如串行 RAM、LCD、音调发生器以及其他微控制器等。

### 8.3 概述

I2C总线的典型配置如图8.3.1所示。根据读写操作位（R/W）状态的不同，I2C总线上存在以下两种类型的数据传输：

- 主发送器向从接收器发送数据。主机发送的第一个字节是从机地址，接下来是数据字节。从机每接收一个字节就返回一个应答位。
- 从发送器向主接收器发送数据。主机发送的第一个字节是从机地址，然后从机返回一个应答位。接下来从机向主机发送数据字节。主机每接收一个字节都会返回一个应答位，最后一个字节除外。接收完最后一个字节后，主机返回一个“非应答位”。主机产生所有串行时钟脉冲、起始条件以及停止条件。每一帧都以一个停止条件或一个重复起始条件来结束。由于重复的起始条件也是下一帧的开始，所以将不会释放I2C总线。

I2C接口面向字节方式，它有四种操作模式：主发送模式、主接收模式、从发送模式和从接收模式。这些I2C接口符合整个I2C规范，可在不影响同一I2C总线上的其它器件的情况下断开与处理器的连接。

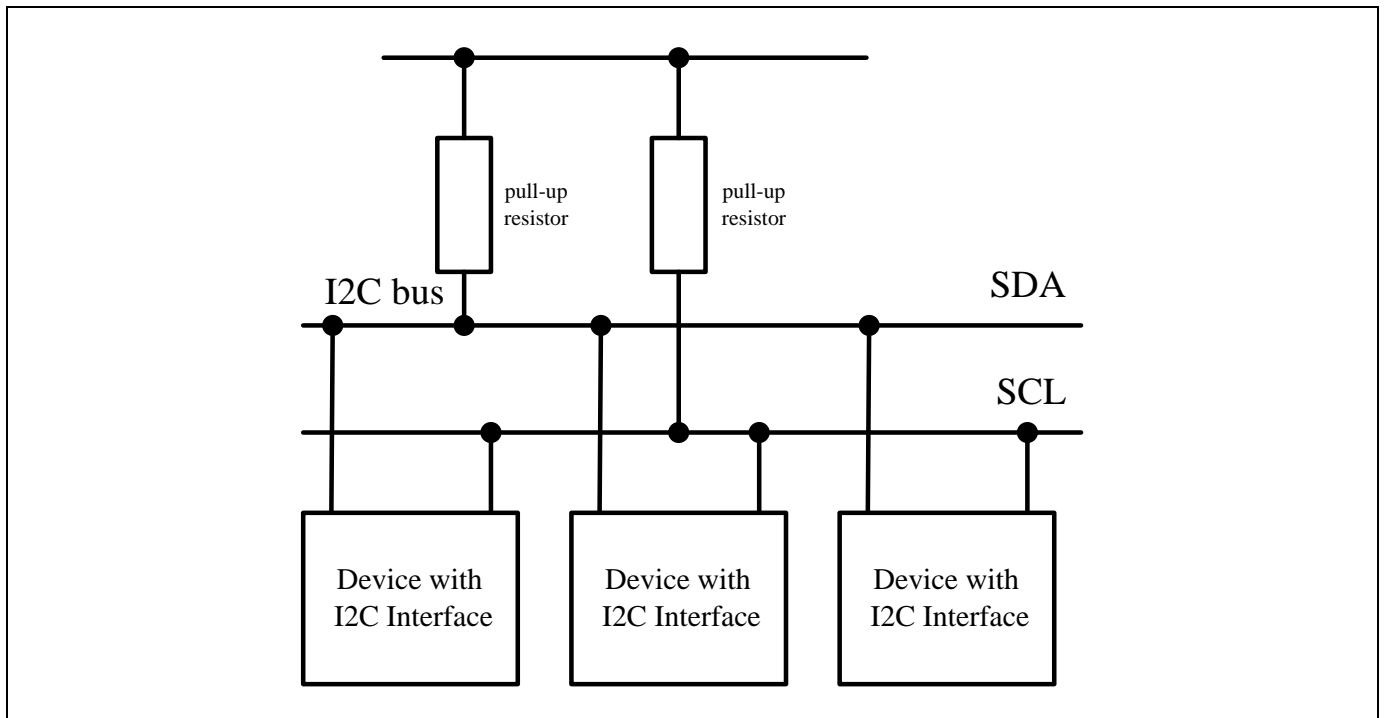


图 8.3.1 I2C 总线配置

快速模式Plus是指以1Mbit/s的速度与I2C设备通信。

#### 8.4 引脚描述

表 8.4.1 I2C 引脚描述

引脚	类型	描述
SDA	输入/输出	I2C串行数据
SCL	输入/输出	I2C串行时钟

#### 8.5 寄存器描述

表 8.5.1 寄存器映射:

I2C0 (基址 0x4000 0000)

名称	访问	地址 偏移	描述	重置值
CONSET	R/W	0x000	<b>控制置位寄存器。</b> 当向该寄存器写入1时，I2C控制寄存器中相应位置位。写0到I2C控制寄存器的相应位没有影响。	0x00
STAT	RO	0x004	<b>状态寄存器。</b> 在I2C操作中，该寄存器提供详细的状态码，使软件确定所需的下一步操作。	0xF8
DAT	R/W	0x008	<b>数据寄存器。</b> 在主/从机发送模式下，要发送的数据被写入该寄存器。在主/从机接收模式下，接收到的数据可从该寄存器中读取。	0x00
ADR0	R/W	0x00C	<b>从地址寄存器0。</b> 包含从机模式下I2C接口操作的7位从地址，不在主机模式下使用。最低位决定从机是否响应通用调用地址。	0x00
SCLH	R/W	0x010	<b>SCH占空比寄存器高半字。</b> 确定I2C时钟的高电平时间。	0x04
SCLL	R/W	0x014	<b>SCL占空比寄存器低半字。</b> 确定I2C时钟的低电平时间。I2nSCLL和I2nSCLH一起确定I2C主机产生的时钟频率和从机模式下使用的特定时间。	0x04

CONCLR	WO	0x018	<b>控制清零寄存器。</b> 当向该寄存器中的位写入1时，I2C控制寄存器中相应位被清零。写0对I2C控制寄存器中的相应位没有影响。	-
-	-	0x01C	保留	-
ADR1	R/W	0x020	<b>从地址寄存器1。</b> 包含从机模式下I2C接口操作的7位从地址，不在主机模式下使用。最低位决定从机是否响应用调用地址。	0x00
-	-	0x024 - 0x028	保留	-
DATABUFFER	RO	0x02C	<b>数据缓冲寄存器。</b> 每次从总线接收到9个位（8个数据位和一个应答位或一个非应答位）之后，移位寄存器DAT中的最高8位就会自动发送到数据缓冲器中。	0x00
MASK0	R/W	0x030	<b>从地址屏蔽寄存器0。</b> 该屏蔽寄存器与地址寄存器I2ADR0相关联，可确定地址匹配。当与通用调用地址（“0000000”）比较时，屏蔽寄存器不起作用。	0x00
MASK1	R/W	0x034	<b>从地址屏蔽寄存器1。</b> 该屏蔽寄存器与地址寄存器I2ADR0相关联，可确定地址匹配。当与通用调用地址（“0000000”）比较时，屏蔽寄存器不起作用。	0x00

### 8.5.1 控制置位寄存器 (CONSET – 0x4000 0000)

CONSET 寄存器控制 CON 寄存器中位的设置，该寄存器中位的设置控制 I2C 接口的操作。写 1 到该寄存器使 I2C 控制寄存器中相应位置位。向这个寄存器中写入 0 无效。

表 8.5.1.1 控制置位寄存器 (CONSET - 0x4000 0000 (I2C0)) 位描述

位	符号	描述	重置值
0	-	保留	0
1	SIPULSE	I2C 中断标志。（功能相同的位 3 (SI)，连接 CPU 中断服务）	
2	AA	应答标志。	
3	SI	I2C 中断标志。	0
4	STO	停止标志。	0
5	STA	起始标志。	0
6	I2EN	I2C 接口使能。	0
31:7	-	保留	-

**I2EN** 接口使能。当I2EN置位时，使能I2C接口。向CONCLR寄存器中的I2ENC位写入1将使I2EN位清零。当I2EN位为0时，I2C接口功能被禁止。

当I2EN为0时，SDA和SCL输入信号被忽略，I2C模块在“不可寻址的”从机状态中，且STO位被强制为“0”。

I2EN不应用于暂时释放I2C总线，当I2EN重置时，I2C总线状态丢失。应使用AA标志代替。

**STA** 起始标志。当STA=1时，I2C接口进入主模式并发送一个起始条件，如果已经处于主模式，则发送一个重复起始条件。

当STA=1并且I2C接口还没进入主模式时，I2C接口进入主模式，检测总线并在总线空闲时产生一个起始条件。如果总线忙，则等待一个停止条件（释放总线）并在延迟半个内部时钟发生器周期后发送一个起始条件。当I2C接口已经处于主模式中并发送或接收了数据时，I2C接口会发送一个重复的起始条件。STA可在任何时候置位，当I2C接口处于可寻址的从模式时，STA也可以置位。

向CONCLR寄存器中的STAC位写入1可使STA位清零。当STA=0时，不会产生起始或重复起始条件。

当 STA 和 STO 都置位时，如果 I2C 接口处于主模式，I2C 接口将向总线发送一个停止条件，然后发送一个起始条件。如果 I2C 接口处于从模式，则产生一个内部停止条件，但不发送到总线上。

**STO** 停止标志。在主模式中，当STO为1时，会使I2C接口发送一个停止条件或在从模式中从错误状态中恢复。当主模式中STO=1时，向I2C总线发送停止条件。当总线检测到停止条件时，STO自动清零。

在从模式中，置位STO位可从错误状态中恢复。这种情况下不向总线发送停止条件。硬件的表现就好像是接收到一

个停止条件并切换到不可寻址的从接收模式。STO标志由硬件自动清零。

**SI** I2C中断标志。当I2C状态改变时SI置位。但是，进入状态F8不会使SI置位，因为在那种情况下中断服务程序不起作用。

当SI置位时，SCL线上的串行时钟低电平周期扩展，且串行传输被中止。当SCL为高时，SI标志的状态不受影响。SI必须通过软件重置，通过向CONCLR寄存器的SIC位写入1来实现。

**AA** 应答标志位。当AA置位时，在SCL线的应答时钟脉冲内，出现下面的任意条件之一将产生一个应答信号（SDA线为低电平）：

1. 接收到从地址寄存器中的地址。
2. 当ADR中的通用调用位（GC）置位时，接收到通用调用地址。
3. 当I2C接口处于主接收模式时，接收到一个数据字节。
4. 当I2C接口处于可寻址的从接收模式时，接收到一个数据字节。

向CONCLR寄存器中的AAC位写入1会使AA位清零。当AA为零时，SCL线的应答时钟脉冲内出现下列情况将返回一个非应答信号（SDA线为高电平）：

1. 当I2C接口处于主接收模式时，接收到一个数据字节。
2. 当I2C接口处于可寻址的从接收模式时，接收到一个数据字节。

### 8.5.2 状态寄存器 (STAT – 0x4000 0004)

每个状态控制器反应对应 I2C 接口的情况。状态寄存器为只读寄存器。

表 8.5.2.1 状态寄存器 (STAT – 0x4000 0004 (I2C0)) 位描述

位	符号	描述	重置值
2:0	-	不使用这些位，它们总是为 0	0
7:3	STATUS	这些位提供 I2C 接口的实际状态	0x1F
31:8	-	保留	-

最低 3 位总是为 0。作为一个整字节时，状态寄存器代表一个状态代码。共有 26 种可能存在的状态代码。当代码为 0xF8 时，无可用的相关信息，SI 位不会置位。所有其他 25 种状态代码都对应一个已定义的 I2C 状态。当进入其中一种状态时，SI 位将置位。

### 8.5.3 数据寄存器(DAT – 0x4000 0008)

该寄存器包含要发送或刚接收的数据。当它没有处理字节的移位时，CPU 可对其进行读写。该寄存器只能在 SI 置位时访问。在 SI 置位期间，DAT 中的数据保持稳定。DAT 中的数据移位总是从右至左进行：第一个发送的位是 MSB（位 7），在接收字节时，第一个接收到的位存放在 DAT 的 MSB。

表 8.5.3.1 数据寄存器(DAT - 0x4000 0008 (I2C0)) 位描述

位	符号	描述	重置值
7:0	DATA	该寄存器保存已接收的数据或要发送的数据	0x00
31:8	-	保留	-

### 8.5.4 从地址寄存器 0 (ADR0 – 0x4000 000C)

这类寄存器可读可写，但只能在I2C设置为从模式时才能使用。在主模式中，该寄存器无效。ADR的LSB为通用调用位。当该位置位时，通用调用地址（0x00）被识别。

其中含有 0x00 的寄存器不能与总线上的地址匹配。重置时，2 个寄存器(ADR0 与 ADR1)都恢复到这个禁止状态。

表 8.5.4.1 从地址寄存器 0 (ADR0 - 0x4000 000C (I2C0)) 位描述

位	符号	描述	重置值
0	GC	通用调用使能位	0
7:1	ADDR	从模式的 I2C 器件地址	0x00
31:8	-	保留	-

### 8.5.5 SCL 高低电平占空比寄存器(SCLH – 0x4000 0010, SCLL – 4000 0014)

**表 8.5.5.1 SCL 高电平占空比寄存器(SCLH - 0x4000 0010 (I2C0)) 位描述**

位	符号	描述	重置值
15:0	SCLH	SCL 高电平周期选择	0x0004
31:16	-	保留	

**表 8.5.5.2 SCL 低电平占空比寄存器(SCLL - 0x4000 0014 (I2C0)) 位描述**

位	符号	描述	重置值
15:0	SCLL	SCL 低电平周期选择	0x0004
31:16	-	保留	-

### 8.5.6 选择合适的 I2C 总线速率和占空比

软件必须通过对 SCLH 和 SCLL 寄存器进行设置来选择合适的总线速率和占空比。SCLH 定义 SCL 高电平所保持的 I2C\_PCLK 周期数，SCLL 定义 SCL 低电平的 I2C\_PCLK 周期数。频率由下面的公式得出（I2C\_PCLK 是外围总线 I2C 的频率）：

$$I2C \text{ frequency} = I2CPCLK / (SCLH + SCLL)$$

选用的 SCLL 和 SCLH 值必须确保得出的总线速率在 I2C 总线速率的范围之内。每个寄存器的值都必须大于或等于 4。表 8.5.6.1 给出了根据 I2C\_PCLK 频率和 SCLL 和 SCLH 值计算出来的 I2C 总线速率的示例。

**表 8.5.6.1 所选 I2C 时钟值的 SCLL + SCLH 值**

I2C mode	I2C bit frequency	I2C_PCLK (MHz)								
		6	8	10	12	16	20	30	40	50
标准模式	100 kHz	60	80	100	120	160	200	300	400	500
快速模式	400 kHz	15	20	25	30	40	50	75	100	125
快速模式 Plus	1 MHz	-	8	10	12	16	20	30	40	50

SCLL 和 SCLH 的值不一定要相同。通过设定这两个寄存器可以得到 SCL 的不同占空比。例如，I2C 总线规范定义在快速模式下和在快速模式 Plus 下的 SCL 低电平时间和高电平时间是不同的。

### 8.5.7 控制清零寄存器(CONCLR – 0x4000 0018)

CONCLR 寄存器控制 CON 寄存器中位的清零并控制 I2C 接口的操作。向该寄存器写入 1 会清零寄存器中对应的。向这个寄存器中写入 0 无效。

**表 8.5.7.1 控制清零寄存器(CONCLR - 0x4000 0018 (I2C0)) 位描述**

位	符号	描述	重置值
1:0	-	保留	0
2	AAC	应答标志清零位	
3	SIC	I2C 中断标志清零位	0
4	-	保留	0
5	STAC	起始标志清零位	0
6	I2ENC	I2C 接口禁止位	0
31:7	-	保留	0

**AAC**：应答标志清零位。向该位写入 1 会清零 CONSET 寄存器中的 AA 位。写入 0 无效。

**SIC**：I2C 中断标志清零位。向该位写入 1 会清零 CONSET 寄存器中的 SI 位。写入 0 无效。

**STAC**：起始标志清零位。向该位写入 1 会清零 CONSET 寄存器中的 STA 位。写入 0 无效。

**I2ENC**：I2C 接口禁止。向该位写入 1 会清零 CONSET 寄存器中的 I2EN 位。写入 0 无效。

### 8.5.8 从地址寄存器 1(ADR1 – 0x4000 0020)

这类寄存器可读可写，但只能在 I2C 设置为从模式时才能使用。在主模式中，该寄存器无效。ADR 的 LSB 为通用调用位。当该位置位时，通用调用地址（0x00）被识别。

其中含有 0x00 的寄存器不能与总线上的地址匹配。重置时，2 个寄存器 (ADR0 与 ADR1) 都恢复到这个禁止状态。



**表8.5.8.1从地址寄存器1(ADR1- 0x4000 0020 (I2C0)) 位描述**

位	符号	描述	重置值
0	GC	通用调用使能位	0
7:1	ADDR	从模式的 I2C 器件地址	0x00
31:8	-	保留.	0

### 8.5.9 数据缓冲器寄存器(DATABUFFER – 0x4000 002C)

在监控模式中，如果 ENA\_SCL 位没置位，I2C 模块就不能延长时钟（使总线迟延）。也就是说，处理器读取总线数据的时间有限。如果处理器以正常速度读取 DAT 移位寄存器，那么在新数据更新接收数据之前，它可能只有 1 位时间来响应中断。

为了给予处理器更多的响应时间，将增加一个新的 8 位只读寄存器 DATA\_BUFFER。每接收到 9 个位（8 个数据位+ 应答位或非应答位）后，寄存器 DAT 的最高 8 位会自动发送到 DATA\_BUFFER。这表示处理器会有 9 个位的传输时间来响应中断，并且在更新接收数据之前读取数据。

处理器仍可直接读取 DAT，DAT 无论如何是不会改变的。

即便 DATA\_BUFFER 主要用于监控模式（ENA\_SCL=0），但在任何一种操作模式下它可以随时读取。

**表 8.5.9.1 数据缓冲器寄存器(DATABUFFER - 0x4000 002C (I2C0)) 位描述**

位	符号	描述	重置值
7:0	DATA	该寄存器保存 DAT 移位寄存器的最高 8 位	0x00
31:8	-	保留	0

### 8.5.10屏蔽寄存器 (MASK0/1 – 0x4000 0030/34)

两个屏蔽寄存器都含有7个有效位（7:1）。这些寄存器中的任意一个位置位都会引起接收地址相应位的自动比较（当它与屏蔽寄存器关联的ADDRn寄存器比较时）。换句话说，就是在确定地址匹配时不会考虑寄存器ADDRn中被屏蔽的位。

重置时所有屏蔽寄存器中的位被清零。

当与通用调用地址（“0000000”）比较时，屏蔽寄存器无效。

屏蔽寄存器中的位（31:8）和（0）不使用也不能写入值。这些为总是为0。

当出现地址匹配中断时，处理器必须读取数据寄存器（DAT）来确定实际引起匹配的接收地址。

**表8.5.10.1屏蔽寄存器 (MASK0/1 –0x4000 0030/34 (I2C0)) 位描述**

位	符号	描述	重置值
0	-	保留	0
7:1	MASK	屏蔽位	0x00
31:8	-	保留	0

### 8.5.11读数据寄存器(RDDAT – 0x4000 0038)

**表 8.5.11.1 读数据寄存器(RDDAT - 0x4000 0038 (I2C0)) 位描述**

位	符号	描述	重置值
7:0	RDATA	该寄存器保存 DAT 移位寄存器的 8 个 MSB 的内容	0x00
31:8	-	保留	0

## 8.6I2C 操作模式

在指定应用中，I2C 模块可用作主机、从机或同时用作主机和从机。在从机模式中，I2C 硬件查找 2 个从机地址之一以及通用调用地址。如果检测到其中一个地址，则产生中断请求。如果处理器想成为总线主机，那么在进入主机模式前硬件必须等到总线空闲，使得从机操作不被中止。如果在主机模式中总线仲裁丢失，则 I2C 模块立刻切换到从机模式并在相同的串行传输中检测自身的从机地址。

### 8.6.1 主发送模式

在该模式中，主机向从机发送数据。在进入主发送模式之前，CONSET 必须按照表 8.6.1.1 进行初始化。必须置位

I2EN 来使能 I2C 功能。如果 AA 位为 0，而另一个器件成为总线的主控制器时，I2C 接口不会对任何地址产生应答，也就是说它无法进入从模式。STA，STO 和 SI 位必须设置为 0。向 CONCLR 寄存器中的 SIC 位写入 1 可清零 SI。STA 应当在写入从地址之后清零。

表 8.6.1.1 用于配置主模式的 CONSET

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	0	-	-

第一个发送的字节包含接收器件的从地址（7位）和读写操作位。在此模式下，读写操作位（R/W）应当为0，表示执行写操作。因此第一个发送的字节为从地址和写操作位。数据的发送每次为8位。每发送完一个字节，都接收到一个应答位。

输出起始和停止条件可用来指示串行传输的起始和结束。

通过软件置位 STA，将进入 I2C 主发送模式。I2C 逻辑在总线空闲后立即发送一个起始条件。当发送完起始条件后，SI 会置位。此时 STAT 中的状态代码为 0x08。

该状态代码用于中断服务程序的处理。该中断程序把从地址和写操作位装入 DAT（数据寄存器），然后清零 SI 位。向 CONCLR 寄存器中的 SIC 位写入 1 可清零 SI。

当从地址和 R/W 位已发送后且接收到应答位，SI 位再次置位，并且对于主模式，可能的状态代码为 0x18，0x20 或 0x38，如果从模式使能（AA=1），可能的状态代码为 0x68，0x78 或 0xB0。The appropriate actions to be taken for each of these status codes are shown in 表 235 to 表 240.

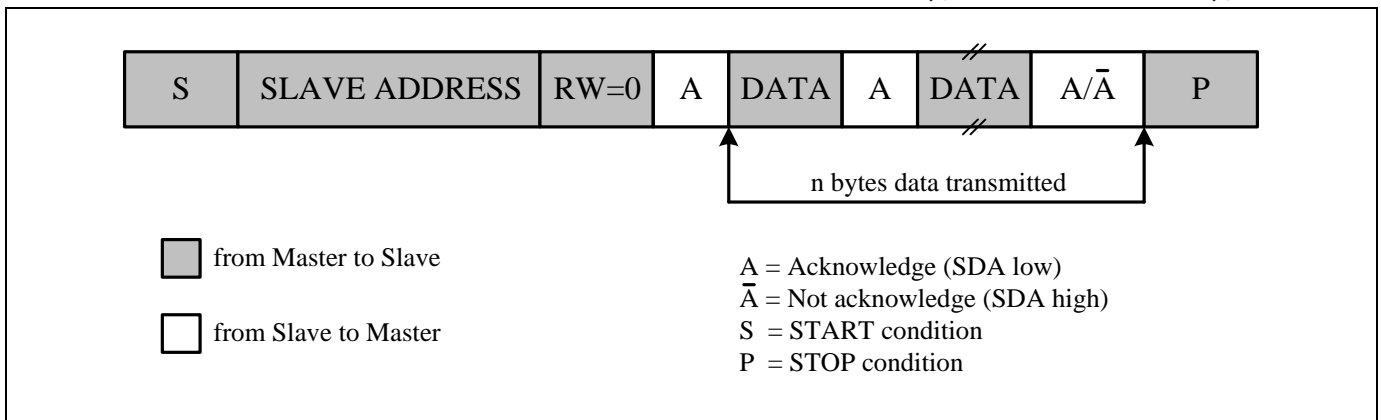


图 8.6.1.1 主发送模式格式

### 8.6.2 主接收模式

在主接收模式中，主机所接收的数据来自从发送器。传输的初始化与主发送模式中的相同。当发送完起始条件后，中断服务程序必须把从地址和读写操作位装入 I2C 数据寄存器（DAT），然后清零 SI 位。在这种情况下，读写操作位（R/W）应该为 1，表示执行读操作。

当从地址和读写操作位已发送且接收到应答位，SI 置位，状态寄存器将显示状态代码。对于主模式，可能的状态代码为 0x40，0x48 或 0x38。对于从模式，可能的状态代码为 0x68，0x78 或 0xB0。For details, refer to 表 236.

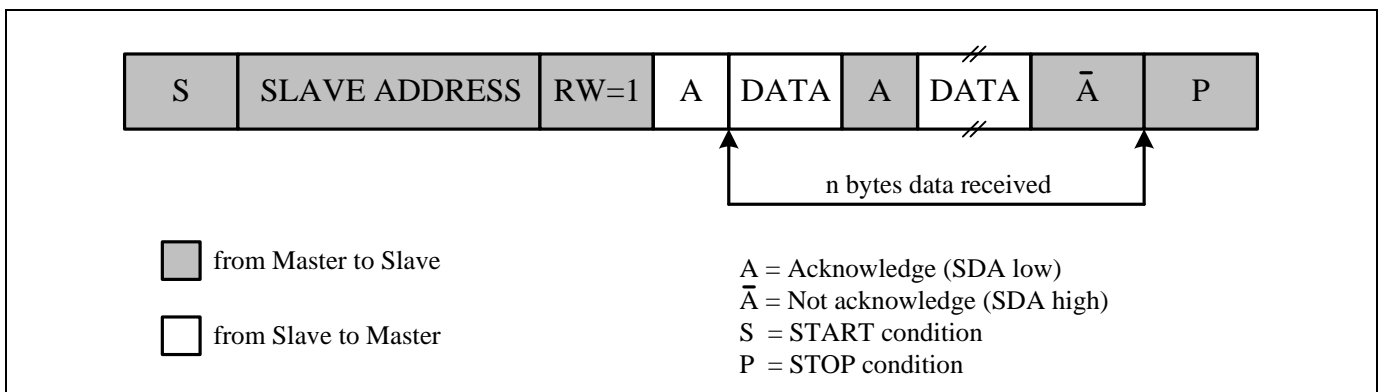


图 8.6.2.1 主接收模式中的格式

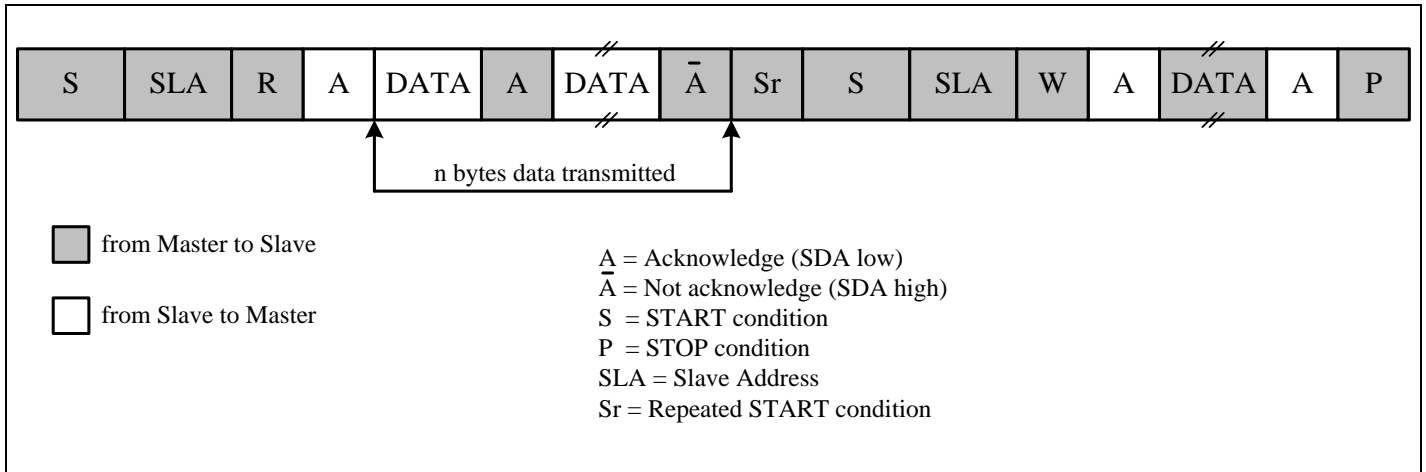


图 8.6.2.2 在发送重复的起始条件后，主接收器切换为主发送器

### 8.6.3 从接收模式

在从接收模式中，接收从主发送器发来的数据字节。要初始化从接收模式，用户必须将从地址写入任意一个从地址寄存器（ADR0~1）并配置 I2C 控制置位寄存器（CONSET），如表 8.6.3.1 所示：

表8.6.3.1用于配置从模式的CONSET

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	1	-	-

必须置位 I2EN 以启用 I2C 功能。AA 位必须置位以使 I2C 应答自身的从地址或通用调用地址。STA，STO 和 SI 设置为 0。

在初始化 ADR 和 CONSET 之后，I2C 接口一直等待直到它被自身的从地址或通用地址（两者后面都紧跟读写操作位）寻址为止。如果读写操作位为 0（W），则 I2C 将进入从接收模式。如果读写操作位为 1（R），则 I2C 将进入从发送模式。在接收到地址和读写操作位后，SI 置位并可从状态寄存器（STAT）中读出有效的状态代码。Refer to 表 239 for the status codes and actions.

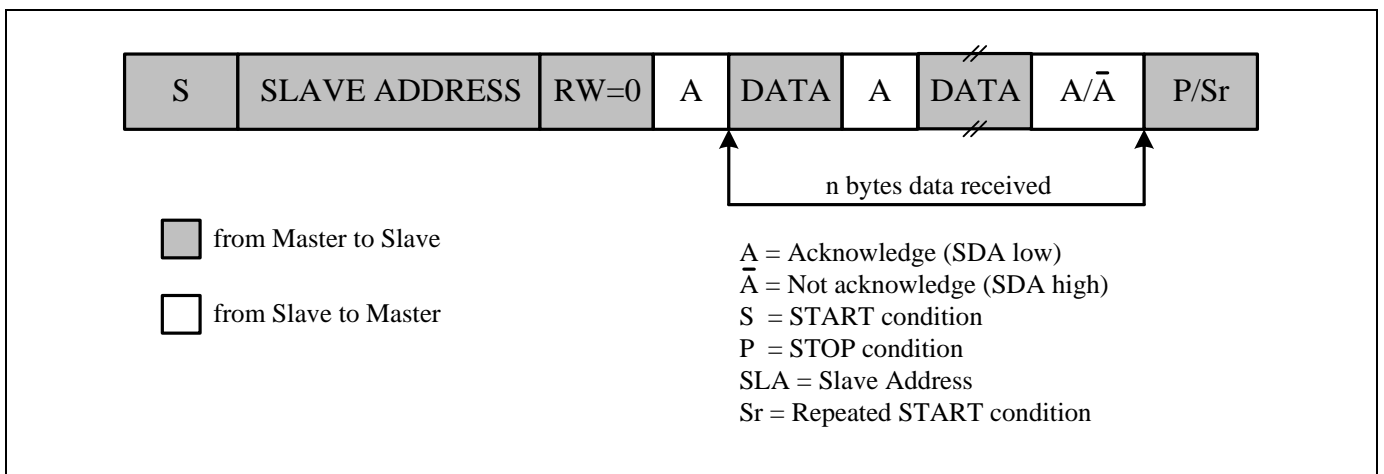


图 8.6.3.1 从接收模式的格式

### 8.6.4 从发送模式

第一个字节的接收和处理与从接收模式中的相同。但在此模式中，读写操作位会为 1，表示执行读操作。串行数据通



过 SDA 发送而串行时钟通过 SCL 输入。起始和停止条件用来指示串行传输的起始和结束。在指定的应用中，I2C 可以为主模式也可以为从模式。在从模式中，I2C 硬件寻找它自身的从地址和通用调用地址。如果检测到其中一个地址，将产生中断请求。当微控制器希望成为总线主机时，硬件在进入主模式前一直等待，直到总线释放，这样就不会中断一个可能的从机动作。如果在主模式中总线仲裁丢失，I2C 接口将立即切换到从模式并可以在同一个串行传输中检测自身的从地址。

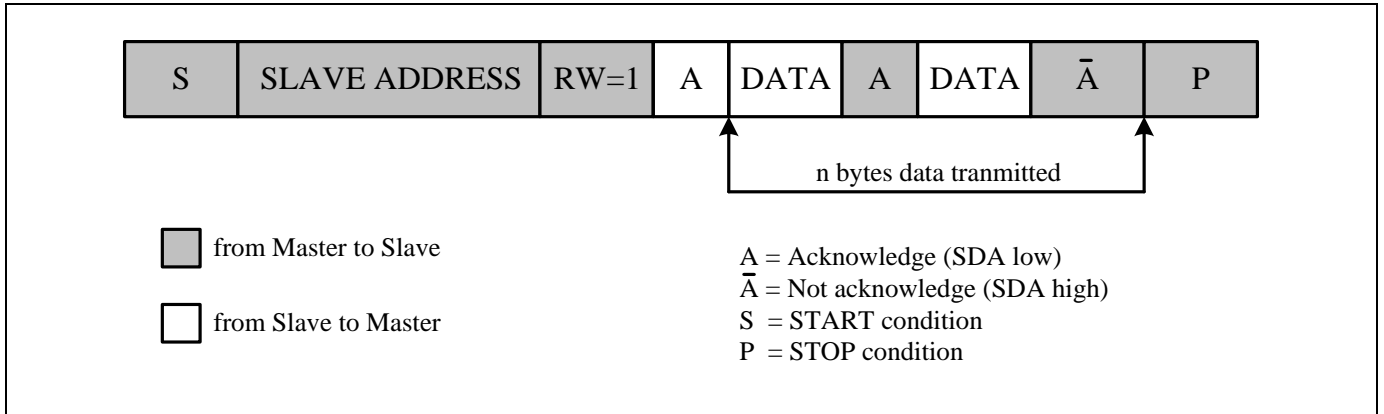


图 8.6.4.1 从发送模式格式

### 8.7 I2C 的实现和操作

图 8.7.1 所示为如何实现片内 I2C 总线接口流程，之后将对图中各模块进行描述。

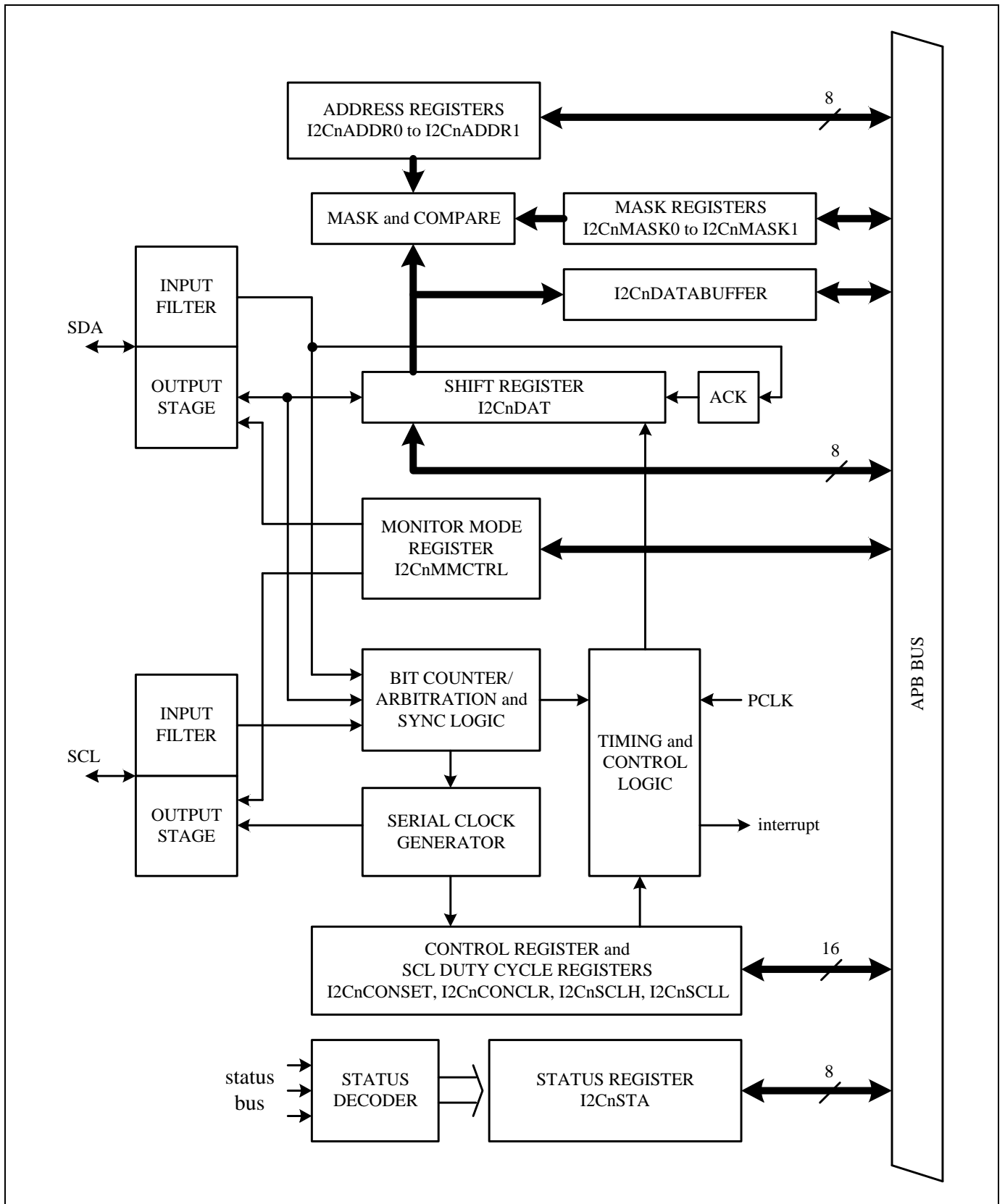


图 8.7.1 I2C 串行接口模块框图

### 8.7.1 输入滤波器和输出级

输入信号与内部时钟同步，不到 3 个时钟的脉冲将被滤除。

I2C 输出是一个特殊的端口，它是为了遵循 I2C 规范而设计的。

### 8.7.2 地址寄存器，ADDR0 与 ADDR1

当器件用作从发送器或接收器时，这类寄存器可装入 I2C 模块即将响应的 7 位从地址（7 个有效位）。LSB（GC）用来使能通用地址（0x00）的识别。当使能多个从机地址时，在接收到器件自身的从地址后，实际接收到的地址可从寄存器 DAT 中读出。

### 8.7.3 地址屏蔽寄存器，ADDR0 与 ADDR1

每个屏蔽寄存器（共2个）都含有7个有效位（7:1）。这些寄存器中的任何一个位被置位都会引起接收地址与屏蔽寄存器相关联的地址寄存器中的相应位做自动比较。换句话说，就是在确定地址匹配时不会考虑寄存器ADDRn被屏蔽的位。

当出现地址匹配中断时，处理器必须读取数据寄存器（DAT）来确定实际引起匹配的地址。

### 8.7.4 比较器

比较器将接收到的 7 位从地址与其自身的从地址（I2ADR 的 7 个最高位）相比较。它还将首次接收到的 8 位字节与通用调用地址（0x00）相比较。如果任何一者相同，相应的状态位就会置位，并产生中断请求。

### 8.7.5 移位寄存器，DAT

该8位寄存器用来存放要发送的一个字节的串行数据或刚接收到的一个字节。DAT的数据总是从右向左移动；最先发送的是最高位（位7），接收完一个字节后，接收到的数据的第一位放置到DAT的最高位。数据移出时，总线上的数据同时移入；DAT通常保存的是总线上的最后一个字节。

因此，在仲裁丢失时，主发送器到从接收器的转变和 DAT 中数据的更新同时进行。

### 8.7.6 仲裁和同步逻辑

在主发送模式中，仲裁逻辑检查每个发送的逻辑 1 是否真正出现在总线上。如果总线的另一个器件撤消了一个逻辑 1 并将 SDA 线拉低，仲裁丢失，I2C 模块立刻由主发送器变为从接收器。I2C 模块将继续输出时钟脉冲（在 SCL 上），直至发送完当前的串行字节。

仲裁也可能在主接收模式中丢失。这种情况只在 I2C 模块正在向总线返回一个“非应答：（逻辑 1）”时出现。

当总线的另一个器件将信号拉低时仲裁丢失。由于它只在串行字节结束时出现，因此 I2C 模块不会再产生时钟脉冲。

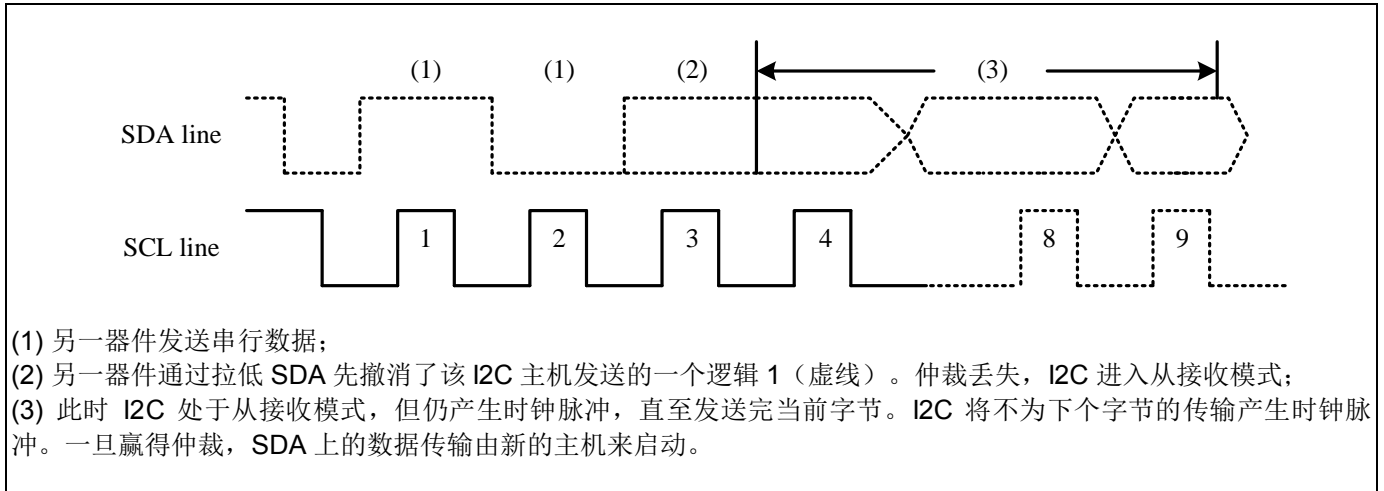


图8.7.6.1仲裁过程

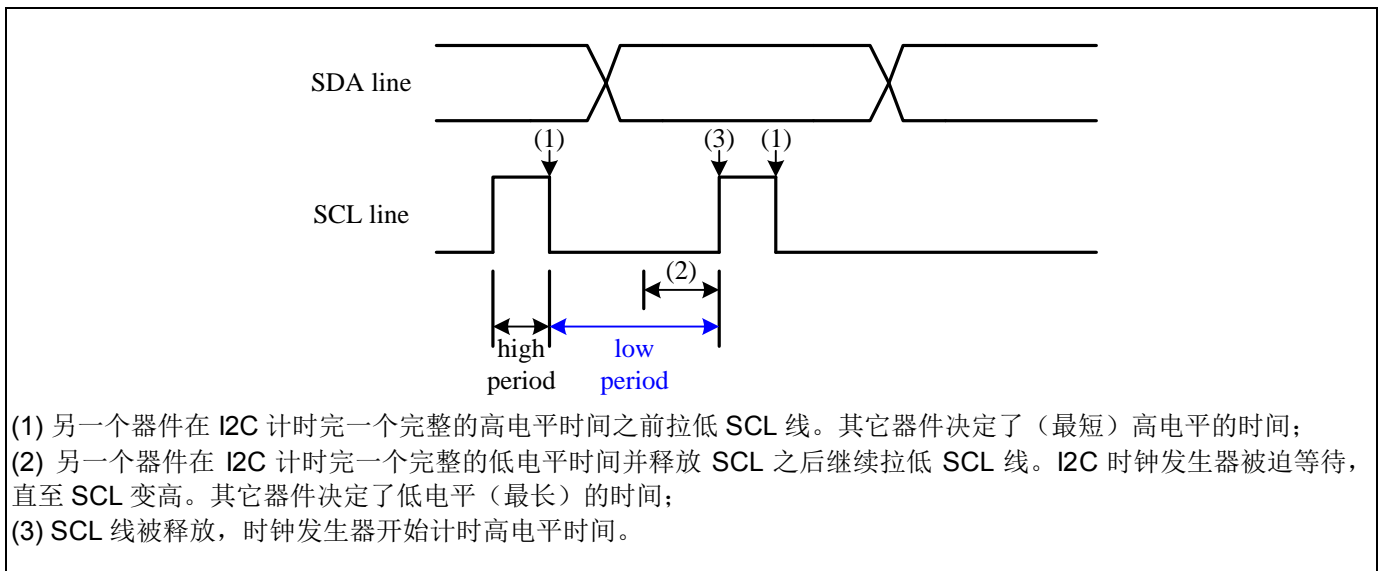


图8.7.6.2串行时钟同步

从机可以延长低电平时间来使总线主机减速。也可通过延长低电平时间来实现握手。延长低电平时间的操作在每位或一个完整字节传输之后执行。I2C 模块将在发送或接收完一个字节且传输完应答位后延长 SCL 低电平时间。设置串行中断标志（SI），继续延长低电平时间，直至串行中断标志清除。

### 8.7.7 串行时钟发生器

当 I2C 模块在主发送器或主接收模式时，这个可编程时钟脉冲发生器可产生 SCL 时钟脉冲。当 I2C 模块处于从机模式时时钟发生器关闭。I2C 输出时钟频率和占空比可通过 I2C 时钟控制寄存器编程。详情请参考 SCLL 和 SCLH 寄存器的描述。除非总线与上面描述的其他 SCL 时钟源同步，否则输出时钟脉冲使用设定好的占空比。

### 8.7.8 时序和控制

时序和控制逻辑会产生处理串行字节的时序和控制信号。该逻辑模块为 DAT 中的数据提供移位脉冲，可启用比较器、产生和检测起始和停止条件以及接收和发送应答位、控制主机和从机模式，该模块还具有中断请求逻辑以及监控 I2C 总线状态等功能。

### 8.7.9 控制寄存器，CONSET 和 CONCLR

I2C控制寄存器包含用于控制以下I2C模块功能的位：串行传输的启动和重启、串行传输的终止、位速率、地址识别和应答。

I2C 控制寄存器的内容可读出。写 CONSET 寄存器会置位 I2C 控制器寄存器中相应的位；相反，写 CONCLR 会清除 I2C 控制寄存器中被写为 1 的位。

### 8.7.10 状态译码器和状态寄存器

状态译码器读取所有内部状态位并将它们压缩成一个 5 位代码。该代码与每个 I2C 总线状态位一一对应。5 位代码可用于产生向量地址，以便快速处理不同的服务程序。每个服务程序处理一个特定的总线状态。如果 I2C 模块的所有四种模式都被使用，则有 26 种可能的总线状态。当串行中断标志置位（通过硬件）并一直保持时，5 位状态代码锁存到状态寄存器的高 5 位，直至中断标志被软件清除。状态寄存器的低 3 位总是为 0。如果状态代码用作服务程序的向量，则程序转移到 8 位地址指向的空间。大多数的服务程序不会超过 8 字节（请参考本节的软件例程）。

### 8.8 I2C 操作模式的详解

有4种操作模式：

- 主发送模式
- 主接收模式
- 从接收模式
- 从发送模式

Data transfers in each mode of operation are shown in 表 8.8.1

表 8.8.1 描述 I2C 操作的缩写

缩写	说明
S	起始条件
SLA	7 位从机地址
R	读数据位 (SDA 为高电平)
W	写数据位 (SDA 为低电平)
A	应答位 (SDA 为低电平)
$\bar{A}$	非应答位 (SDA 为高电平)
Data	8 位数据字节
P	停止条件

In 图 53 to 图 57, circles are used to indicate when the serial interrupt flag is set.

在图8.8.1.1到图8.8.4.1中，圆圈用来指示串行中断标志何时被置位。圆圈中的数字表示STAT寄存器中的状态代码。每当出现这些状态代码时，必须执行服务程序来继续或结束串行传输。若串行传输被挂起，就不要考虑服务程序，直至串行中断标志被软件清除。

当进入串行中断程序时，STAT 的状态代码用来指向跳转到的相应的服务程序。For each status code, the required

#### 8.8.1 主发送模式

在主发送模式中，向从接收器发送数据字节（见图 8.8.1.1）。在进入主发送器模式之前，CON 必须进行如下初始化：

表8.8.1.1 CONSET用于初始化主发送器模式

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	x	-	-

I2C速率也必须在SCLL和SCLH寄存器中配置。必须将I2EN设置为逻辑1来使能I2C模块。如果AA位重置，当另一个器件正变成总线主机时，I2C模块将不会应答其自身的从机地址或通用调用地址。换句话说，如果AA位重置，I2C接口就不能进入从机模式。STA、STO和SI必须重置。

此时，可通过置位STA位进入主发送模式。一旦总线空闲，I2C逻辑会马上测试I2C总线并产生一个起始条件。

当发送起始条件时，串行中断标志（SI）置位，状态寄存器（STAT）中的状态代码为0x08。中断服务程序利用该状态代码进入相应的状态服务程序，将从机地址和数据方向位（SLA+W）装入DAT。CON的SI位必须在串行传输继续之前重置。

当发送完从机地址和方向位且接收到一个应答位时，串行中断标志（SI）再次置位，STAT中可能是一系列不同的状

态代码。主机模式下为0x18, 0x20或0x38, 从机模式(AA=逻辑1)为0x68, 0x78或0xB0。

每个状态代码对应的操作在表8.8.1.2中详细介绍。

在发送完重复起始条件(状态0x10)后, I2C模块通过将SLA+R装入DAT切换到主接收模式。

表 8.8.1.2 主发送模式

状态 代码 STAT	I2C 总线和硬件的状态	应用软件的响应 读/写 DAT	写 CON				I2C 硬件执行的下一个动作
			STA	STO	SI	AA	
0x08	已发送起始条件.	装入 SLA+R; 清除 STA	X	0	0	X	将发送 SLA+R; 接收 ACK 位
0x10	已发送重复的起始条件.	装入 SLA+R	X	0	0	X	同上
		装入 SLA+R; 清除 STA	X	0	0	X	将发送 SLA+W; I2C 将切换为 MST/REC 模式
0x18	已发送 SLA+W; 已接收 ACK	读取数据字节	0	0	0	X	将发送数据字节, 接收 ACK 位
		无 DAT 动作	1	0	0	X	将发送重复的起始条件
		无 DAT 动作	0	1	0	X	将发送停止条件; STO 标志将重置
		无 DAT 动作	1	1	0	X	将发送停止条件, 然后发送起始条件; STO 标志将重置
0x20	已发送 SLA+W; 已接收非 ACK	读取数据字节	0	0	0	X	将发送数据字节, 接收 ACK 位
		无 DAT 动作	1	0	0	X	将发送重复的起始条件
		无 DAT 动作	0	1	0	X	将发送停止条件; STO 标志将重置
		无 DAT 动作	1	1	0	X	将发送停止条件, 然后发送起始条件; STO 标志将重置
0x28	已发送 DAT 中的数据字节; 已接收 ACK	读取数据字节	0	0	0	X	将发送数据字节, 接收 ACK 位
		无 DAT 动作	1	0	0	X	将发送重复的起始条件
		无 DAT 动作	0	1	0	X	将发送停止条件; STO 标志将重置
		无 DAT 动作	1	1	0	X	将发送停止条件, 然后发送起始条件; STO 标志将重置
0x30	已发送 DAT 中的数据字节; 已接收非 ACK.	读取数据字节	0	0	0	X	将发送数据字节, 接收 ACK 位
		无 DAT 动作	1	0	0	X	将发送重复的起始条件
		无 DAT 动作	0	1	0	X	将发送停止条件; STO 标志将重置
		无 DAT 动作	1	1	0	X	将发送停止条件, 然后发送起始条件; STO 标志将重置
0x38	在 SLA+RW 或数据字节 中丢失仲裁	无 DAT 动作	0	0	0	X	I2C 总线将被释放; 进入不可寻址从模式
		无 DAT 动作	1	0	0	X	当总线空闲时发送起始条件

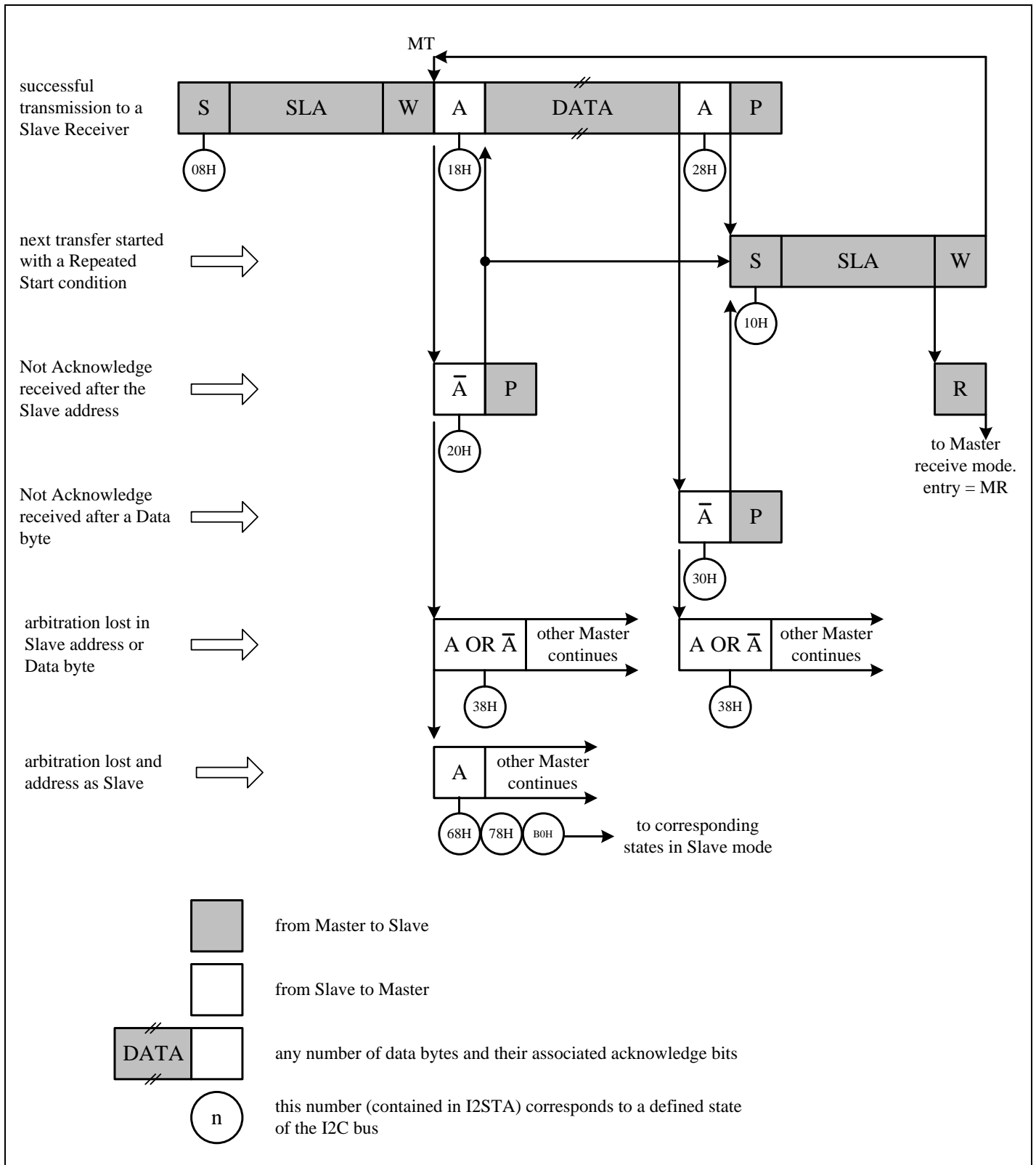


图 8.8.1.1 主发送模式中的格式和状态



### 8.8.2 主接收模式

在主接收模式中，主机所接收的数据字节来自从发送器（见附图8.8.2.1）。按主发送模式中的方法初始化传输。当发送完起始条件后，中断服务程序必须把7位从机地址和数据方向位（SLA+R）装入DAT。必须先清除CON中的SI位，再继续执行串行传输。

当发送完从机地址和数据方向位且接收到一个应答位时，串行中断标志SI再次置位，这时，STAT中可能是一系列不同的状态代码。主机模式下为0x40，0x48或0x38，从机模式（AA= 1）为0x68，0x78或0xB0。每个状态代码对应的操作详见表8.8.2.1。在发送完重复起始条件（状态0x10）后，I2C模块通过将SLA+W装入DAT切换到主发送模式。

表 8.8.2.1 主接收模式

状态 代码 STAT	I2C 总线和硬件的状态	应用软件的响应 读/写 DAT	写 CON				I2C 硬件执行的下一个动作
			STA	STO	SI	AA	
0x08	已发送起始条件.	装入 SLA+R	X	0	0	X	将发送 SLA+R; 接收 ACK 位
0x10	已发送重复的起始条件.	装入 SLA+R	X	0	0	X	同上
		装入 SLA+W	X	0	0	X	SLA+W; I2C 模块将切换为 MST/TRX 模式
0x38	在非 ACK 位中丢失仲裁	无 DAT 动作	0	0	0	X	2C 总线将被释放; 进入从模式
		无 DAT 动作	1	0	0	X	当总线空闲时发送起始条件
0x40	已发送 SLA+R; 已接收 ACK	无 DAT 动作	0	0	0	0	将接收数据字节, 返回非 ACK 位
		无 DAT 动作	0	0	0	1	将接收数据字节, 返回 ACK 位
0x48	已发送 SLA+R; 已接收非 ACK	无 DAT 动作	1	0	0	X	将发送重复的起始条件
		无 DAT 动作	0	1	0	X	将发送停止条件; STO 标志将重置
		无 DAT 动作	1	1	0	X	将发送停止条件, 然后发送起始条件; STO 标志将重置
0x50	已接收数据字节, ACK 已 返回	无 DAT 动作	0	0	0	0	将接收数据字节, 返回非 ACK 位
		无 DAT 动作	0	0	0	1	将接收数据字节, 返回 ACK 位
0x58	已接收数据字节, 非 ACK 已返回	无 DAT 动作	1	0	0	X	将发送重复的起始条件
		无 DAT 动作	0	1	0	X	将发送停止条件; STO 标志将重置
		无 DAT 动作	1	1	0	X	将发送停止条件, 然后发送起始条件; STO 标志将重置



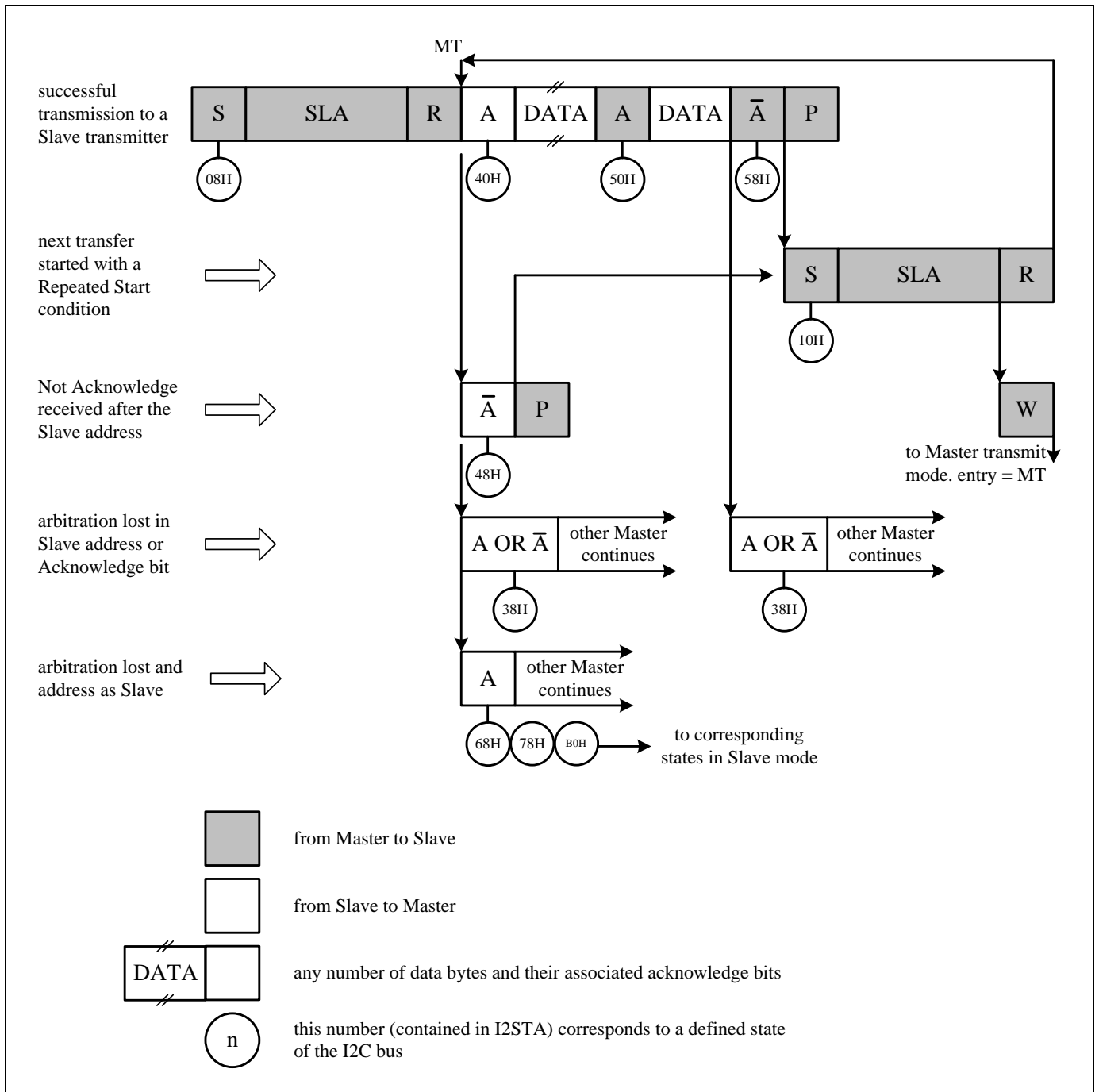


图 8.8.2.1 主接收模式中的格式和状态

### 8.8.3 从接收模式

在从接收模式中，从机接收的数据字节来自自主发送器（见图8.8.3.1）。要初始化从接收模式，必须按照来配置ADR和CON：

表8.8.3.1在从接收模式中使用的ADR

位	7	6	5	4	3	2	1	0
符号	自身 7 位从地址							GC

高7位是主机寻址时I2C模块响应的地址。如果LSB（GC）被置位，I2C模块将响应通用调用地址（0x00）；否则忽略通用调用地址。

表8.8.3.2初始化从接收模式时CONSET的配置

位	7	6	5	4	3	2	1	0
符号	-	I2EN	STA	STO	SI	AA	-	-
值	-	1	0	0	0	1	-	-

I2C总线速率的设置不影响从机模式中的I2C模块。必须置位I2EN来使能I2C模块。AA位必须置位以使能I2C模块来应答其自身从机地址或通用调用地址。STA，STO和SI必须重置。

当ADR和CON完成初始化后，I2C模块一直等待，直至被从机地址寻址，之后是数据方向位寻址，为了工作在从接收模式中，数据方向位必须为“0”（W）。接收完其自身的从机地址和W位后，串行中断标志（SI）置位，可从STAT中读出一个有效的状态代码。该状态代码用作状态服务程序的向量。每个状态代码的对应操作见表8.8.3.3。如果I2C模块在主机模式中仲裁丢失，也可进入从接收模式（请参考状态0x68和0x78的描述）。

如果AA位在传输过程中重置，则在接收完下一个数据字节后I2C模块将向SDA返回一个非应答（逻辑1）。当AA重置时，I2C模块不响应其自身的从机地址或通用调用地址。但是，I2C总线仍被监控，而且，地址识别可随时通过置位AA来恢复。这就意味着AA位可临时将I2C模块从I2C总线上分离出来。

表 8.8.3.3 从接收模式

状态代码 STAT	I2C 总线和硬件的状态	应用软件的响应 读/写 DAT	写 CON				I2C 硬件执行的下一个动作
			STA	STO	SI	AA	
0x60	已接收自身的 SLA+W；已返回 ACK	无 DAT 动作	X	0	0	0	将接收数据字节，返回非 ACK 位
		无 DAT 动作	X	0	0	1	将接收数据字节，返回 ACK 位
0x68	主控制器时在 SLA+R/W 中丢失仲裁；已接收自身 SLA+W，已返回 ACK	无 DAT 动作	X	0	0	0	将接收数据字节，返回非 ACK 位
		无 DAT 动作	X	0	0	1	将接收数据字节，返回 ACK 位
0x70	已接收通用调用地址（0x00）；已返回 ACK	无 DAT 动作	X	0	0	0	将接收数据字节，返回非 ACK 位
		无 DAT 动作	X	0	0	1	将接收数据字节，返回 ACK 位
0x78	主控制器在 SLA+R/W 中丢失仲裁；已接收通用调用地址，已应答	无 DAT 动作	X	0	0	0	将接收数据字节，返回非 ACK 位
		无 DAT 动作	X	0	0	1	将接收数据字节，返回 ACK 位
0x80	前一次寻址使用自身从地址；已接收数据字节；已返回 ACK	读取数据字节	X	0	0	0	将接收数据字节，返回非 ACK 位
		读取数据字节	X	0	0	1	将接收数据字节，返回 ACK 位
0x88	前一次寻址使用自身 SLA；已接收数据字节；已返回非 ACK	读取数据字节	0	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址
		读取数据字节	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 ADR[0]=逻辑 1，将识别通用调用地址
		读取数据字节	1	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		读取数据字节	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 ADR[0]=逻辑 1，将识别通用调用地址；当总线空闲后发送起始条件
0x90	前一次寻址使用通用调用；已接收数据字节；已返回 ACK	读取数据字节	X	0	0	0	将接收数据字节，返回非 ACK 位
		读取数据字节	X	0	0	1	将接收数据字节，返回 ACK 位

0x98	前一次寻址使用通用调用；已接收数据字节；已返回非 ACK	读取数据字节	0	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址
		读取数据字节	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 ADR[0]=逻辑 1，将识别通用调用地址
		读取数据字节	1	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		读取数据字节	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 ADR[0]=逻辑 1，将识别通用调用地址；当总线空闲后发送起始条件
0xA0	当使用 SLV/REC 或 SLV/TRX 静态寻址时，接收到停止条件或重复的起始条件	无 STDAT 动作	0	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址
		无 STDAT 动作	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 ADR[0]=逻辑 1，将识别通用调用地址
		无 STDAT 动作	1	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		无 STDAT 动作	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 ADR[0]=逻辑 1，将识别通用调用地址；当总线空闲后发送起始条件

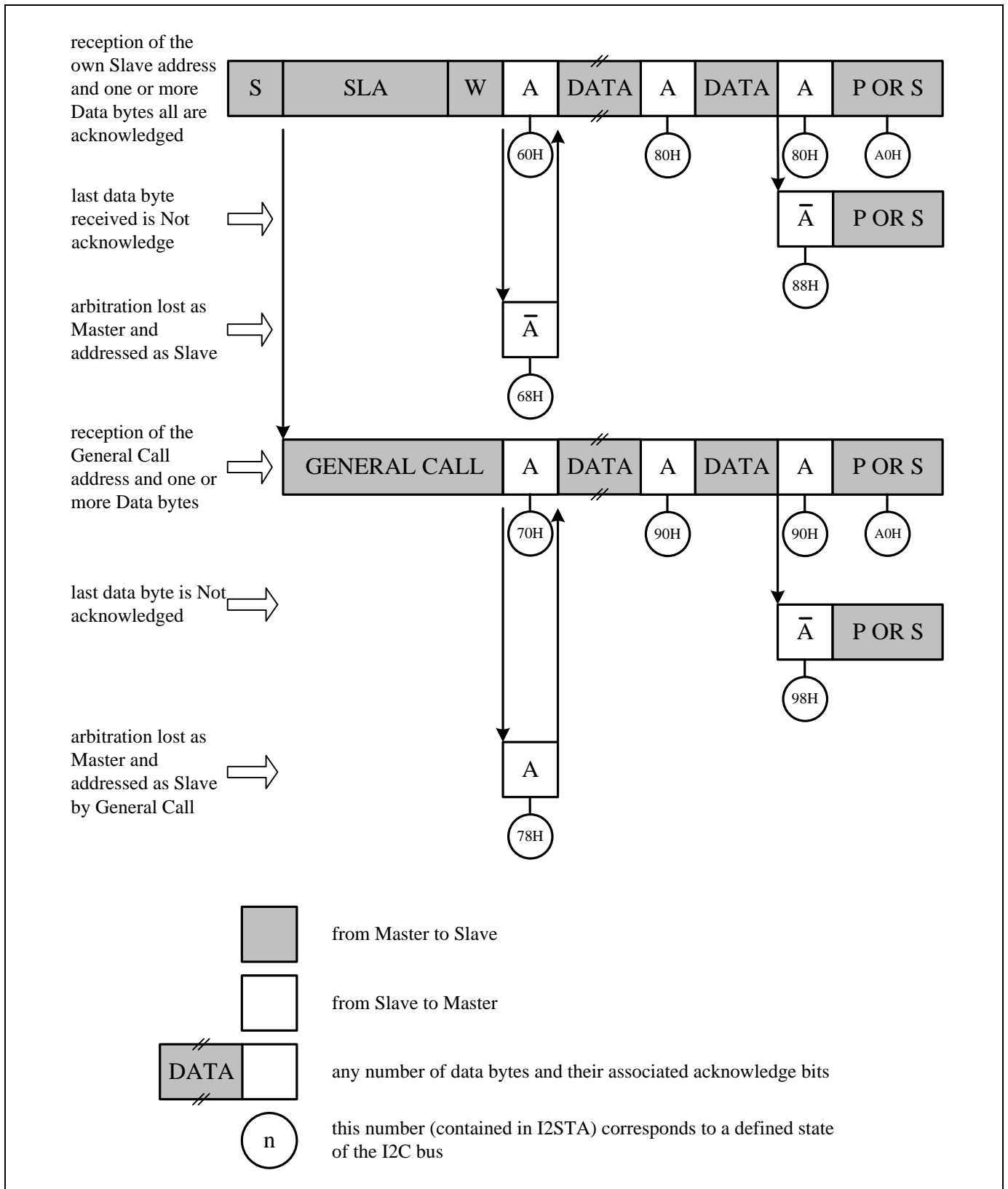


图 8.8.3.1 从接收模式中的格式和状态

### 8.8.4 从发送模式

在从发送模式中，向主接收器发送数据字节（见图 8.8.4.1）。数据传输按照从接收模式中的情况初始化。当初初始化 I2ADR 和 I2CON 后，I2C 模块一直等待，直至被自身的从机地址寻址，之后是数据方向位，该数据方向位必须为“1”（R），以便 I2C 模块工作在从发送模式下。接收完其自身的从机地址和 R 位后，串行中断标志（SI）置位，并且可从 STAT 中读取一个有效的状态代码。

该状态代码用作状态服务程序的向量，每个状态代码的对应操作见表 8.8.4.1。如果 I2C 模块在主机模式下时仲裁丢失，则可进入从发送模式（见状态 0xB0）。

如果 AA 位在传输过程中重置，则 I2C 模块将发送最后一个字节并进入状态 0xC0 或 0xC8。I2C 模块切换到非寻址的从机模式，如果继续传输，它将忽略主接收器。因此主接收器接收所有 1 作为串行数据。当 AA 重置时，I2C 模块不响应其自身的从机地址或通用调用地址。但是，I2C 总线仍被监控，而且，地址识别可随时通过置位 AA 来恢复。这就意味着 AA 位可用来暂时将 I2C 模块从 I2C 总线上分离出来。

表 8.8.4.1 从发送模式

状态 代码 STAT	I2C 总线和硬件的状态	应用软件的响应 读/写 DAT	写 CON				I2C 硬件执行的下一个动作
			STA	STO	SI	AA	
0xA8	已接收自身的 SLA+R；已返回 ACK	读取数据字节	X	0	0	0	最后一个数据字节将被发送，ACK 位将被接收
		读取数据字节	X	0	0	1	数据字节将被发送，ACK 位将被接收
0xB0	主控器时在 SLA+R/W 中丢失仲裁；已返回 ACK	读取数据字节	X	0	0	0	最后一个数据字节将被发送，ACK 位将被接收
		读取数据字节	X	0	0	1	数据字节将被发送，ACK 位将被接收
0xB8	已接收 DAT 中的数据字节；已返回 ACK	读取数据字节	X	0	0	0	最后一个数据字节将被发送，ACK 位将被接收
		读取数据字节	X	0	0	1	数据字节将被发送，ACK 位将被接收
0xC0	已发送 DAT 中的数据字节；已接收非 ACK。	无 DAT 动作	0	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址
		无 DAT 动作	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 ADR[0]=逻辑 1，将识别通用调用地址
		无 DAT 动作	1	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		无 DAT 动作	1	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 ADR[0]=逻辑 1，将识别通用调用地址；当总线空闲后发送起始条件
0xC8	最后一个 DAT 中的数据字节已被发送（AA=0）；已接收 ACK	无 DAT 动作	0	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址
		无 DAT 动作	0	0	0	1	切换到不可寻址 SLV 模式；识别自身 SLA；如果 ADR[0]=逻辑 1，将识别通用调用地址
		无 DAT 动作	1	0	0	0	切换到不可寻址 SLV 模式；不识别自身 SLA 或通用调用地址；当总线空闲后发送起始条件
		无 DAT 动作	1	0	0	01	切换到不可寻址 SLV 模式；识别自身 SLA；如果 ADR[0]=逻辑 1，将识别通用调用地址；当总线空闲后发送起始条件

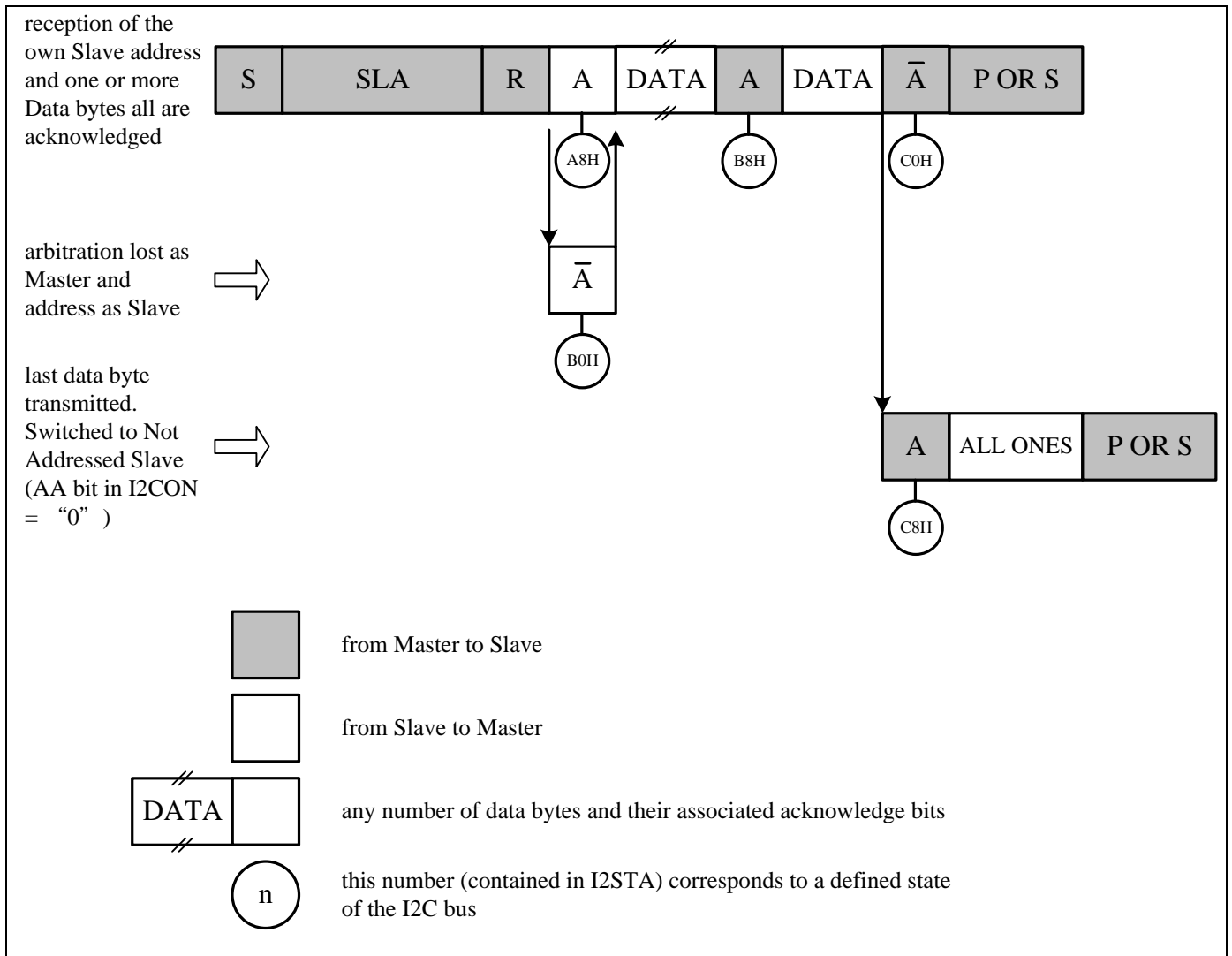


图 8.8.4.1 从发送模式中的格式和状态

### 8.8.5 其他状态

还有 2 种不对应硬件状态的 STAT 代码（见表 8.8.5.1），下面将对这两种代码进行讨论：

#### 8.8.5.1 STAT = 0xF8

这个状态码表示没有任何可用的相关信息，因为串行中断标志 SI 还没有置位。这种情况在其它状态和 I2C 模块还未开始执行串行传输之间出现。

#### 8.8.5.2 STAT = 0x00

该状态代码表示在 I2C 串行传输过程中出现了总线错误。当格式帧的非法位置上出现了起始或停止条件时总线错误产生。这些非法位置是指在串行传输过程中的地址字节、数据字节或应答位。当外部干扰影响到内部 I2C 模块信号时也会产生总线错误。总线错误出现时 SI 置位。要从总线错误中恢复，STO 标志必须置位，SI 必须被清除。这使得 I2C 模块进入“非寻址的”从机模式（已定义的状态）并清除 STO 标志（CON 中的其它位不受影响）。SDA 和 SCL 线被释放（不发送停止条件）。

表 8.8.5.1 I2C 状态-其他状态

状态代码	I2C 总线和硬件的状态	应用软件的响应 读/写 DAT	写 CON	I2C 硬件执行的下一个动作
------	--------------	--------------------	-------	----------------

STAT		STA	STO	SI	AA	
0xF8	无可用的相关状态信息；无 DAT 动作 SI=0					等待或执行当前传输
0x00	由于非法起始或停止条件的出现，在 MST 或选择的从机模式中将出现总线错误。当外部干扰使 I2C 模块进入未定义的状态时也出现 0x00 状态	0	1	0	X	只有 MST 或寻址的 SLV 模式中的内部硬件受影响。一般情况下，总线被释放、I2C 模块切换到非寻址的 SLV 模式。STO 重置

### 8.8.6 一些特殊情况

I2C 硬件可以处理以下几种串行传输过程中出现的特殊情况：

- 两个主机同时启动重复起始条件
- 仲裁丢失后的数据传输
- 强制访问 I2C 总线
- SCL 或 SDA 低电平妨碍 I2C 总线的操作
- 总线错误

#### 8.8.6.1 两个主机同时启动重复起始条件

在主发送模式或主接收模式下可以产生重复起始条件。如果此时另一个主机同时产生重复起始条件，就出现特殊情况（图 8.8.6.1.1）。在出现这种情况之前，任何一个主机都不会丢失仲裁，因为它们发送的数据相同。

如果 I2C 硬件在产生重复起始条件之前在 I2C 总线上检测到重复起始条件，它将释放总线，并且不产生中断请求。如果另一个主机通过产生停止条件来释放总线，则 I2C 模块将发送一个正常的起始条件（状态 0x08），并开始启动重新尝试完整的串行数据传输。



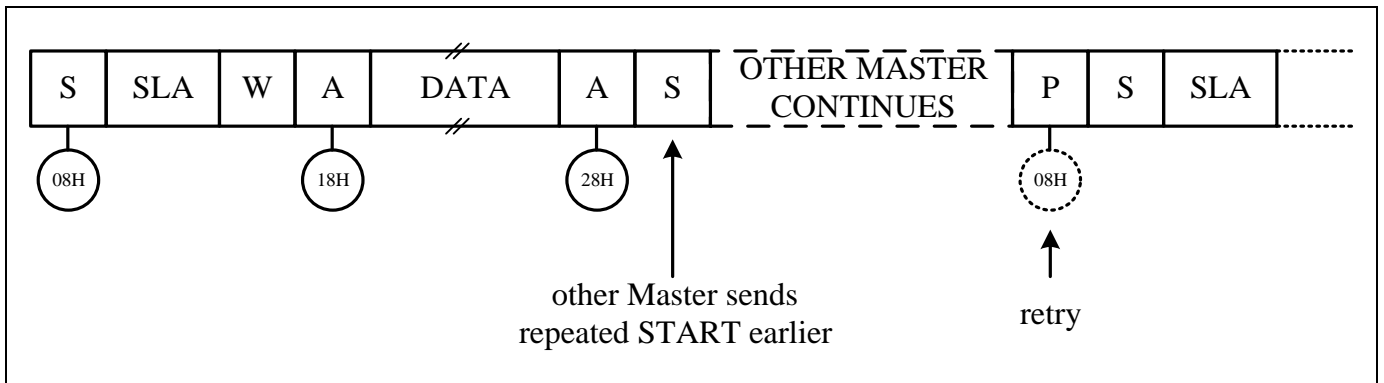


图 8.8.6.1.1 两个主机同时发送重复起始条件

### 8.8.6.2 仲裁丢失后的数据传输

在主发送模式和主接收模式中仲裁可能会丢失。STAT寄存器中的状态代码可表示仲裁丢失，代码有：0x38，0x68，0x78和0xB0。

如果CON中的STA标志被服务这些状态的程序置位，则当总线再次空闲时，会发起一个起始条件（状态0x08），并且不受CPU的影响，开始重新尝试完整的串行数据传输。

### 8.8.6.3 强制访问 I2C 总线

在某些应用中，非控制源可能会造成总线挂起。在这种情况下，干扰、总线的暂时中断或SDA和SCL之间的暂时短路都会导致总线挂起。

如果非控制源产生了一个多余的起始条件或屏蔽了一个停止条件，则I2C总线一直保持忙碌状态。如果STA标志置位且在相应的时间内未访问总线，那么I2C总线有可能会被强制访问。这可通过在STA标志仍被设置时置位STO标志来实现。不发送停止条件。I2C的硬件操作就好像是接收到停止条件一样，可以发送起始条件。STO标志通过硬件清除（见图 8.8.6.3.1）。

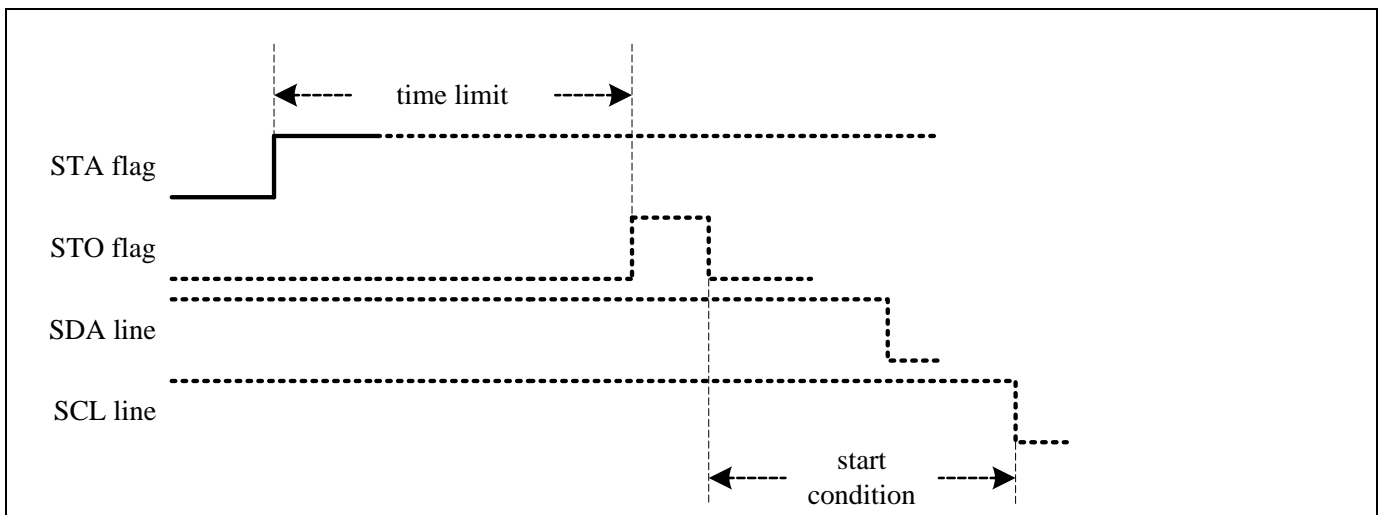


图 8.8.6.3.1 强制访问忙碌的 I2C 总线

### 8.8.6.4 SCL 或 SDA 低电平妨碍 I2C 总线的操作

如果SDA或SCL被总线上任何一个器件拉低，I2C总线就会挂起。如果SCL线被总线上的器件拉低，不能继续串行传输，这时可通过拉低SCL线的器件来处理。

一般来说，SDA线可能会被总线上另一个不当前总线主机同步的器件拉低，不同步的原因可能是丢失了一个时钟周期或是以噪声脉冲作为时钟信号。在这种情况下，可通过向SCL发送另外的时钟脉冲来处理。（见图 8.8.6.4.1）



虽然 I2C 接口没有专门用来检测总线挂起的定时器，但可以用系统的其他定时器来完成。当检测到总线挂起时，软件会强制给 SCL 最多 9 个时钟信号，直至 SDA 被器件释放。此时，从机可能还是不同步，所以还要发送一个起始条件以确保所有 I2C 外设同步。

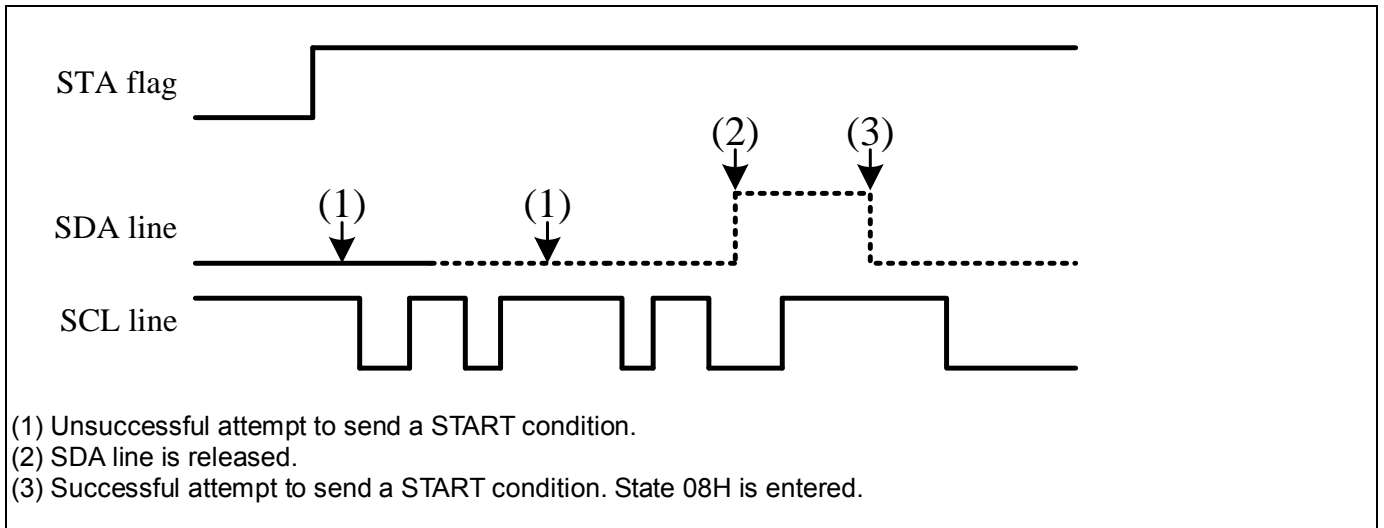


图 8.8.6.4.1 从总线干扰中恢复（由 SDA 低电平引起的）

#### 8.8.6.5 总线错误

当格式帧的非法位置上出现起始或停止条件时总线错误产生。非法位置是指串行传输过程中的地址字节、数据位或应答位。

仅当 I2C 硬件作为主机或被寻址的从机进行串行传输时，它才对总线错误有反应。检测到总线错误时，I2C 模块会立即切换成非寻址的从机模式，并释放 SDA 和 SCL 线，设置中断标志，并将 0x00 装入状态寄存器。该状态代码可用作状态服务程序的向量，尝试再次终止串行传输或从错误状态中恢复。

#### 8.8.7 I2C 状态服务程序

本节将介绍不同 I2C 状态服务程序必须执行的操作，它们包括：

- 重置后 I2C 模块的初始化
- I2C 中断服务
- 支持 4 种 I2C 操作模式的 26 种状态服务程序。

### 8.8.8 初始化

在初始化示例中，I2C模块可在主机模式和从机模式中使能。

对于每种模式，缓冲区可用于发送和接收数据。初始化程序将执行以下操作：

- 向ADR装入器件自身的从机地址和通用调用位（GC）
  - 置位I2C中断使能位和中断优先级位
  - 设置CON寄存器中的I2EN和AA位使能从机模式，向CON装入SCLH和SCLL来定义串行时钟频率（主机模式）。主机程序必须从主程序开始执行。
- 这时，I2C 硬件开始在 I2C 总线上检查自身的从机地址和通用调用位。一旦检测到通用调用或自身从地址，请求中断，相应的状态信息装入 STAT。

### 8.8.9 I2C 中断服务

当进入 I2C 中断时，I2STAT 存入一个状态代码，可识别要执行的 26 个状态服务中的其中一个。

### 8.8.10 状态服务程序

每个状态程序都是I2C中断程序的组成部分，分别用来处理26种状态。

### 8.8.11 实际应用中的状态服务

状态服务示例演示了响应26个I2C状态代码必须执行的典型操作。如果4种I2C操作模式中有一种或几种没被用到，则模式的相关的状态服务可被忽略，只要小心处理，那些状态就不会出现。

在应用中，可能需要在 I2C 操作过程中执行一些超时处理，来限制无效的总线或丢失的服务程序。

## 8.9 软件示例

### 8.9.1 初始化程序

将I2C接口初始化用作从机和/或主机的例子。

1. 将自身的从机地址装入ADR，使能通用调用识别（如果需要的话）
2. 使能I2C中断
3. 向寄存器CONSET写入0x44来置位I2EN和AA位，并使能从机功能。对于主机功能，可向寄存器CONSET写入0x40。

### 8.9.2 启动主机发送功能

通过建立缓冲区、指针和数据计数器然后发启起始条件便可执行主发送操作。

1. 初始化主机数据计数器
2. 建立数据将被发送到的从机地址，并且添加写位
3. 向CONSET写入0x20来置位STA位
4. 在主发送缓冲区内建立要发送的数据
5. 初始化主机数据计数器来匹配正在发送的信息长度
6. 退出

### 8.9.3 启动主机接收功能

通过建立缓冲区、指针和数据计数器然后发启起始条件便可执行主接收操作

1. 初始化主机数据计数器
2. 建立数据将被发送到的从机地址，并且添加读位
3. 向CONSET写入0x20来置位STA位
4. 在主接收缓冲区内建立要发送的数据
5. 初始化主机数据计数器来匹配正在发送的信息长度
6. 退出

#### 8.9.4 I2C 中断程序

确定I2C的状态和处理该状态的状态程序

1. 从STA中读出I2C的状态
2. 使用状态值跳转到 26 个可能状态程序中的一个

#### 8.9.5 无指定模式的状态

##### 8.9.5.1 状态: 0x00

总线错误。进入非寻址的从机模式并释放总线。

1. 向CONSET写入0x14来置位STO和AA位
2. 向CONCLR写入0x08来清除SI标志
3. 退出

##### 8.9.5.2 状态: 0x08

已发送起始条件。即将发送从机地址+R/W位和接收ACK位。

1. 向DAT写入从机地址和R/W位
2. 向CONSET写入0x04来置位AA位
3. 向CONCLR写入0x08来清除SI标志
4. 建立主发送模式数据缓冲区
5. 建立主接收模式数据缓冲区
6. 初始化主机数据计数器
7. 退出

##### 8.9.5.3 状态: 0x10

已发送重复的起始条件。即将发送从机地址+R/W位和接收ACK位。

1. 向DAT写入从机地址和R/W位
2. 向CONSET写入0x04来置位AA位
3. 向CONCLR写入0x08来清除SI标志
4. 建立主发送模式数据缓冲区
5. 建立主接收模式数据缓冲区
6. 初始化主机数据计数器
7. 退出

#### 8.9.6 主发送状态

##### 8.9.6.1 状态: 0x18

之前状态为8或10表示已发送从机地址和写操作位，并接收了应答。即将发送第一个数据字节和接收ACK位。

1. 将主发送缓冲区的第一个数据字节装入DAT
2. 向CONSET写入0x04来置位AA位
3. 向CONCLR写入0x08来清除SI标志
4. 主发送缓冲区指针加1
5. 退出

##### 8.9.6.2 状态: 0x20

已发送从机地址和写操作位并接收了非应答。即将发送停止条件。

1. 向CONSET写入0x14来置位STO和AA位
2. 向CONCLR写入0x08来清除SI标志
3. 退出

#### 8.9.6.3 状态: 0x28

已发送数据并接收了ACK。如果发送的数据是最后一个数据字节则发送一个停止条件，否则发送下一个数据字节。

1. 主机数据计数器减1，如果发送的不是最后一个数据字节就跳至第5步
2. 向CONSET写入0x14来置位STO和AA位
3. 向CONCLR写入0x08来清除SI标志
4. 退出
5. 将主发送缓冲区的下一个数据字节装入DAT
6. 向CONSET写入0x04来置位AA位
7. 向CONCLR写入0x08来清除SI标志
8. 主机发送缓冲区指针加1
9. 退出

#### 8.9.6.4 状态: 0x30

已发送数据并接收到非应答。即将发送停止条件。

1. 向CONSET写入0x14来置位STO和AA位
2. 向CONCLR写入0x08来清除SI标志
3. 退出

#### 8.9.6.5 状态: 0x38

仲裁已在发送从机地址和写操作位或数据的过程中丢失。总线已被释放且进入非寻址的从机模式。当总线再次空闲时将发送一个新的起始条件。

1. 向CONSET写入0x24来置位STA和AA位
2. 向CONCLR写入0x08来清除SI标志
3. 退出

### 8.9.7 主接收状态

#### 8.9.7.1 状态: 0x40

前面的状态是08或10表示已发送从机地址和读操作位，并接收到ACK。将接收数据和返回ACK。

1. 向CONSET写入0x04来置位AA位
2. 向CONCLR写入0x08来清除SI标志
3. 退出

#### 8.9.7.2 状态: 0x48

已发送从机地址和读操作位，并接收到非应答。将发送停止条件。

1. 向CONSET写入0x14来置位STO和AA位
2. 向CONCLR写入0x08来清除SI标志
3. 退出

#### 8.9.7.3 状态: 0x50

已接收到数据，并返回ACK。将从I2DAT读取数据。将接收其它的数据。如果这是最后一个数据字节，则返回非应答，否则返回ACK。

1. 读取I2DAT中的数据字节，存放到主机接收缓冲区
2. 主机数据计数器减1，如果不是最后一个数据字节就跳到第5步
3. 向CONCLR写入0x0C来清除SI标志和AA位
4. 退出
5. 向CONSET写入0x04来置位AA位
6. 向CONCLR写入0x08来清除SI标志
7. 主机接收缓冲区指针加1

## 8. 退出

### 8.9.7.4 状态: 0x58

已接收到数据，已返回非应答。将从I2DAT中读取数据和发送停止条件。

1. 读取DAT中的数据字节，存放到主机接收缓冲区
2. 向CONSET写入0x14来置位STO和AA位
3. 向CONCLR写入0x08来清除SI标志
4. 退出

### 8.9.8 从接收状态

#### 8.9.8.1 状态: 0x60

已接收到自身从机地址和写操作位，已返回ACK。将接收数据和返回ACK。

1. 向CONSET写入0x04来置位AA位
2. 向CONCLR写入0x08来清除SI标志
3. 建立从接收模式数据缓冲区
4. 初始化从机数据计数器
5. 退出

#### 8.9.8.2 状态: 0x68

用作总线主机时仲裁已在传输从机地址和R/W位时丢失。已接收到自身从机地址和写操作位，并已返回ACK。将接收数据和返回ACK。当总线再次空闲后置位STA来重启主机模式。

1. 向CONSET写入0x24来置位STA和AA位
2. 向CONCLR写入0x08来清除SI标志
3. 建立从接收模式数据缓冲区
4. 初始化从机数据计数器
5. 退出

#### 8.9.8.3 状态: 0x70

已接收到通用调用和返回ACK。将接收数据和返回ACK。

1. 向CONSET写入0x04来置位AA位
2. 向CONCLR写入0x08来清除SI标志
3. 建立从接收模式数据缓冲区
4. 初始化从机数据计数器
5. 退出

#### 8.9.8.4 状态: 0x78

用作总线主机时仲裁已在传输从机地址和R/W位时丢失。已接收到通用调用和返回ACK。将接收数据和返回ACK。当总线再次空闲后置位STA来重启主机模式。

1. 向CONSET写入0x24来置位STA和AA位
2. 向CONCLR写入0x08来清除SI标志
3. 建立从接收模式数据缓冲区
4. 初始化从机数据计数器
5. 退出

#### 8.9.8.5 状态: 0x80

之前寻址自身从机地址。已接收到数据并返回ACK。将读取其它数据。

1. 读取DAT的数据字节，存放到从机接收缓冲区。
2. 从机数据计数器减1，如果不是最后一个数据字节就跳到第5步

3. 向CONCLR写入0x0C来清除SI标志和AA位
4. 退出.
5. 向CONSET写入0x04来置位AA位
6. 向CONCLR写入0x08来清除SI标志
7. 从机接收缓冲区指针加1
8. 退出

#### 8.9.8.6 状态: 0x88

之前寻址自身从机地址。已接收到数据并返回非应答。不会保存接收到的数据。进入非寻址的从机模式。

1. 向CONSET写入0x04来置位AA位
2. 向CONCLR写入0x08来清除SI标志
3. 退出

#### 8.9.8.7 状态: 0x90

之前寻址通用调用地址。已接收到数据并返回ACK。

将保存接收到的数据。只接收第一个数据字节并返回ACK。接收其它数据字节后返回非应答。

1. 读取DAT的数据字节，并放入从机接收缓冲区
2. 向CONCLR写入0x0C来清除SI标志和AA位
3. 退出

#### 8.9.8.8 状态: 0x98

之前寻址通用调用地址。已接收到数据并返回非应答。不会保存接收到的数据。进入非寻址的从机模式。

1. 向CONSET写入0x04来置位AA位
2. 向CONCLR写入0x08来清除SI标志
3. 退出

#### 8.9.8.9 状态: 0xA0

已接收停止条件或重复起始条件，但仍作为从机寻址。不保存接收到的数据。进入非寻址的从机模式。

1. 向CONSET写入0x04来置位AA位
2. 向CONCLR写入0x08来清除SI标志
3. 退出

### 8.9.9 从发送状态

#### 8.9.9.1 状态: 0xA8

已接收自身从机地址和读操作位并返回ACK。将发送数据和接收ACK位。

1. 将从机发送缓冲区的第一个数据字节装入DAT
2. 向CONSET写入0x04来置位AA位
3. 向CONCLR写入0x08来清除SI标志
4. 建立从发送模式数据缓冲区
5. 从机发送缓冲区指针加1
6. 退出

#### 8.9.9.2 状态: 0xB0

用作总线主机时，在传输从机地址和RW位时丢失仲裁。已接收自身从机地址和读操作位并返回ACK。将发送数据和接收ACK位。当总线再次空闲后置位STA来重启主机模式。

1. 将从机发送缓冲区的第一个数据字节装入DAT
2. 向CONSET写入0x24来置位STA和AA位
3. 向CONCLR写入0x08来清除SI标志

4. 建立从发送模式数据缓冲区
5. 从机发送缓冲区指针加1
6. 退出

#### 8.9.9.3 状态: 0xB8

已发送数据并接收到ACK。将发送数据和接收ACK位。

1. 将从机发送缓冲区的数据字节装入DAT
2. 向CONCLR写入0x08来清除SI标志
3. 从机发送缓冲区指针加1
4. 退出

#### 8.9.9.4 状态: 0xC0

已发送数据并接收到非应答。进入非寻址的从机模式。

1. 向CONSET写入0x04来置位AA位
2. 向CONCLR写入0x08来清除SI标志
3. 退出。

#### 8.9.9.5 状态: 0xC8

已发送最后一个数据字节并接收到ACK。进入非寻址的从机模式。

1. 向CONSET写入0x04来置位AA位
2. 向CONCLR写入0x08来清除SI标志
3. 退出



## 9. 串行外设接口(SPI)

### 9.1 特性

- 兼容摩托罗拉 SPI、4 线德州仪器 SSI 和国家半导体 Microwire 总线
- 同步串行通信
- 支持主机或从机操作
- 同时适用于发送与接收的 8 帧 FIFO
- 4 位至 16 位帧

### 9.2 简介

SSP/SPI 是同步串行端口 (SSP) 控制器，可控制 SPI、4 线 SSI 或 Microwire 总线的操作。它可以与总线上的多个主机和从机进行交互。

在指定数据传输中，总线上只有一个主机和一个从机进行通信。数据传输原则上为全双工方式，4 位到 16 位数据帧由主机发送到从机或由从机发送到主机。实际上，通常情况下只有一个方向上的数据流包含有意义的数

### 9.3 引脚描述

表 9.3.1 SPI 引脚描述

引脚名称	类型	接口引脚名称 / 功能			引脚说明
		SPI	SSI	Microwire	
SCK0	I/O	SCK	CLK	SK	串行时钟。SCK/CLK/SK 是用于同步数据传输的时钟信号。由主机驱动，从机接收。当使用 SSP/SPI 接口时，可将时钟编程为高电平有效或低电平有效，否则，它一直是高电平有效。SCK 只在数据传输期间跳变。在其它时间，SSP/SPI 接口使其保持非工作状态或不驱动它（使其处于高阻抗状态）。
SSEL0	I/O	SSEL	FS	CS	帧同步 / 从机选择。当 SSP/SPI 接口为总线主机时，它在串行数据发起前将该信号驱动到工作状态，再在发送数据后将信号释放到非工作状态。该信号为高电平有效还是低电平有效取决于所选择的总线和模式。当 SSP/SPI 接口为总线从机时，该信号根据使用的协议限定从主机发出的数据。 当只有一个总线主机和一个总线从机时，来自主机的帧同步或从机选择信号可直接连接到从机的相应输入中。当总线上有多个从机时，通常必需进一步限制其帧选择 / 从机选择输入，以避免多个从机对传输作出响应。□
MISO0	I/O	MISO	DR(M) DX(S)	SI(M) SO(S)	主机输入从机输出。MISO 信号将串行数据由从机传输到主机。当 SSP/SPI 是从机时，从该信号上输出串行数据。当 SSP/SPI 为主机时，它记录从该信号发出的串行数据。当 SSP/SPI 为从机，且未被 FS/SSEL 选择时，它不会驱动该信号（使其处于高阻抗状态）。
MOSI0	I/O	MOSI	DX(M) DR(S)	SO(M) SI(S)	主机输出从机输入。MOSI 信号将串行数据从主机传输到从机。当 SSP/SPI 为主机时，从该信号上输出串行数据。当 SSP/SPI 为从机时，它记录从该信号发出的串行数据。

### 9.4 寄存器描述

SPI 控制器的寄存器地址如表 9.4.1 所示。

重置值仅反映已使用位中所存储的数据。不包含保留位中的内容。

注：寄存器名称使用 SSP 前缀表示 SPI 控制器拥有全部 SSP 功能。

表 9.4.1 寄存器简介: SPI (基址 0x5001 8000)

名称	访问类型	地址偏移	描述	重置值
CR0	R/W	0x000	控制寄存器 0。选择串行时钟速率、总线类型和数据大小。	0
CR1	R/W	0x004	控制寄存器 1。选择主机 / 从机和其他模式。	0
DR	R/W	0x008	数据寄存器。写满发送 FIFO，读空接收 FIFO。	0



SR	RO	0x00C	状态寄存器	0x0000 0003
CPSR	R/W	0x010	时钟预定标寄存器	0
IMSC	R/W	0x014	中断掩码设置和清除寄存器	0
RIS	RO	0x018	原始中断状态寄存器	0x0000 0008
MIS	RO	0x01C	掩码后中断状态寄存器	0
ICR	WO	0x020	ICR 中断清除寄存器	0

#### 9.4.1 控制寄存器 0 (CR0 – 0x5001 8000)

该寄存器控制 SSP/SPI 控制器的基本运行。

表 9.4.1.1 控制寄存器0 (CR0 - 0x5001 8000) 位描述

位	符号	值	描述	重置值
3:0	DSS		数据大小选择。该域控制每帧中传输的位数。不支持且不使用值 0000-0010。	0000
		0x3	4 位传输	
		0x4	5 位传输	
		0x5	6 位传输	
		0x6	7 位传输	
		0x7	8 位传输	
		0x8	9 位传输	
		0x9	10 位传输	
		0xA	11 位传输	
		0xB	12 位传输	
		0xC	13 位传输	
5:4	FRF		帧格式。	00
		0x0	SPI	
		0x1	TI	
		0x2	Microwire	
6	CPOL		时钟输出极性。该位只用于 SPI 模式。	0
		0	在帧与帧之间由 SPI 控制器将总线时钟维持在低电平状态。	
		1	在帧与帧之间由 SPI 控制器将总线时钟维持在高电平状态。	
7	CPHA		时钟输出相位。该位只用于 SPI 模式。	0
		0	SPI 控制器在帧传输的第一次时钟跳变时捕获串行数据，也就是说跳变远离时钟线的帧间状态。	
		1	SPI 控制器在帧传输的第二次时钟跳变时捕获串行数据，也就是说跳变回到时钟线的帧间状态。	
15:8	SCR		串行时钟速率。总线上的每位预分频器输出时钟数目，减去 1。假设 CPSDVSR 为预分频器，APB 时钟 PCLK 计时预分频器，则位频率为 $PCLK/([CPSDVSR+2] \times [SCR+1])$ 。	0x00
31:16	-	-	保留	-

#### 9.4.2 控制寄存器 1 (CR1 – 0x5001 8004)

该寄存器控制 SSP/SPI 控制器运行的某些方面。

表 9.4.2.1 控制寄存器1 (CR1 - 0x5001 8004) 位描述

位	符号	值	描述	重置值
0	-	-	保留	
1	SSE		SPI 使能。	0

	0	SPI 控制器被禁用。	
	1	SPI 控制器将在串行总线上与其他装置进行交互。在设置该位之前，软件应该向其他 SSP/SPI 寄存器和中断控制器寄存器写入合适的控制信息。	
2	MS	主机 / 从机模式。只有在 SSE 位为 0 时，才能对该位执行写入操作。	0
	0	SPI 控制器作为总线上的主机，驱动 SCLK、MOSI 和 SSEL 线路，接收 MISO 线路。	
	1	SPI 控制器作为总线上的从机，驱动 MISO 线路，接收 SCLK、MOSI 和 SSEL 线路。	
7:3	-	保留	
8	SSN	从机选择负（低电平有效）	1
9	SSNP	从机选择负脉冲（参考 9.5.4: SSNP 功能）	0
	0	从机选择来自 SSN（第 8 位），软件控制	
	1	从机选择来脉冲发生，硬件控制	
11:10	SSNCNTL	从机选择负计数低。 SSNP（bit-9）置位，从机选择和 SPI 数据（MSB 或 LSB）之间的时序。	00
	00	1 个 SPI 时钟周期	
	01	2 个 SPI 时钟周期	
	10	3 个 SPI 时钟周期	
	11	4 个 SPI 时钟周期	
13:12	SSNCNTH	从机选择负计数高。 SSNP（bit-9）将从机选择高电平的时序。	00
	00	1 个 SPI 时钟周期	
	01	2 个 SPI 时钟周期	
	10	3 个 SPI 时钟周期	
	11	4 个 SPI 时钟周期	
31:14	-	保留	0

#### 9.4.3 数据寄存器 (DR – 0x5001 8008)

软件可将需要发送的数据写入该寄存器，并从中读取接收到的数据。

表 9.4.3.1 数据寄存器 (DR - 0x5001 8008) 位描述

位	符号	值	描述
15:0	DATA	写入：只要状态寄存器中的 TNF 位为 1，表示 Tx FIFO 不处于满状态，软件就可将即将在未来帧中发送的数据写入该寄存器。如果 Tx FIFO 之前为空且总线上的 SPI 控制器不在忙碌状态，可以立即开始数据发送。否则写入该寄存器的数据将在之前所有数据发送（接收）完毕后立即发送出去。如果数据长度小于 16 位，则软件必须使写入该寄存器的数据向右对齐。 读：只要状态寄存器的 RNE 位为 1，表示 Rx FIFO 不为空，软件就可从该寄存器中读取数据。当软件从该寄存器中读取数据时，SPI 控制器将返回 Rx FIFO 中最近帧中的数据。如果数据长度小于 16 位，则使此字段的数据向右对齐，更高阶位用 0 填充。	0x0000
31:16	-	保留	

#### 9.4.4 状态寄存器 (SR – 0x5001 800C)

该只读寄存器反映 SPI 控制器的当前状态。

表 9.4.4.1 状态寄存器 (SR - 0x5001 800C) 位描述

位	符号	值	描述
0	TFE	发送 FIFO 为空。当发送 FIFO 为空时该位为 1，否则为 0。	1
1	TNF	发送 FIFO 未滿。如果 Tx FIFO 已滿，则该位为 0；反之为 1。	1
2	RNE	接收 FIFO 未空。如果接收 FIFO 为空，则该位为 0；反之为 1。	0
3	RFF	接收 FIFO 满。如果接收 FIFO 已滿，则该位为 1；反之为 0。	0
4	BSY	忙。如果 SPI 控制器空闲，则该位为 0；如果当前正在发送 / 接收一个帧和 / 或 Tx FIFO 未空，则该位为 1。	0
31:5	-	保留	0

#### 9.4.5 时钟预分频寄存器 (CPSR – 0x5001 8010)

该寄存器控制预分频器分频SPI外设时钟SPI\_PCLK以产生预分频器时钟的系数，反过来，再在CR0寄存器中被SCR系数分频，决定位时钟。

表 9.4.5.1 时钟预分频寄存器(CPSR - 0x5001 8010) 位描述

位	符号	值	描述
7:0	CPSDVSR	此偶数值介于 2 至 254 之间， SPI_PCLK 通过该值进行分频以产生预分频器输出时钟。位 0 始终读取为 0。	0
31:8	-	保留	-

注意事项：CPSR 值必须适当初始化，否则 SPI 控制器不能正确发送数据。

在主机模式下， CPSDVSR<sub>min</sub> = 2 或更大的值（只能为偶数）。

#### 9.4.6 中断掩码设置 / 清除寄存器(IMSC – 0x5001 8014)

该寄存器控制是否使能 SPI 控制器中 4 个可能的中断条件。注意：ARM 所使用的术语“masked”与普通计算机术语中的“masked”的意思是相反的，“masked”在普通计算机术语中是“禁用”的意思，而在 ARM 术语中则是“使能”。为了避免混淆，这里不使用“masked”这个词。

表 9.4.6.1 中断掩码设置 / 清除寄存器 (IMSC - 0x5001 8014) 位描述

位	符号	值	描述
0	RORIM	出现接收溢出（即当 Rx FIFO 已满且另一个帧完全接收）时，软件应将此位设置为使能中断。ARM 规范指明，出现这种情况时，前面的帧数据会被新的帧数据覆盖。	0
1	RTIM	出现接收超时条件时，软件应将此位设置为使能中断。当 Rx FIFO 不为空且超过超时周期没有被读取时，将发生接收超时。 对于主机和从机模式来说超时周期是相同的，由 SSP 比特率决定： 在 $PCLK/([CPSDVSR+2] \times [SCR+1])$ 时为 32 位。	0
2	RXIM	当 Rx FIFO 为至少半满状态时，软件应该设置此位以使能中断。	0
3	TXIM	当 Tx FIFO 为至少半空状态时，软件应该设置此位以使能中断。	0
31:4	-	保留	0

#### 9.4.7 原始中断状态寄存器(RIS – 0x5001 8018)

这是一个只读寄存器，当中断条件出现时，寄存器中对应的位置为，这与是否在 IMSC 寄存器中使能无关。

表 9.4.7.1 原始中断寄存器(RIS - 0x5001 8018) 位描述

位	符号	值	描述
0	RORRIS	当 Rx FIFO 满时又接收到另一帧数据时，该位置位。ARM 特别指出，发生接收溢出时新数据帧会将前面的数据帧覆盖。	0
1	RTRIS	如果 Rx FIFO 不为空，且在“超时周期”中没有被读出时，该位置位。对于主机和从机模式来说超时周期是相同的，由 SSP 比特率决定：在 $PCLK/([CPSDVSR+2] \times [SCR+1])$ 时为 32 位。	0
2	RXRIS	当 Rx FIFO 至少有一半为满时，该位置位。	0
3	TXRIS	当 Tx FIFO 至少有一半为空时，该位置位。	1
31:4	-	保留	0

#### 9.4.8 掩码后中断状态寄存器(MIS – 0x5001 801C)

该寄存器是一个只读寄存器，当中断条件出现且相应的中断在 IMSC 寄存器中使能时，该寄存器中对应的位为 1。当出现 SSP/SPI 中断时，中断服务例程会读取此寄存器以确定中断的原因。

表 9.4.8.1 掩码后中断状态寄存器(MIS - 0x5001 801C) 位描述

位	符号	值	描述
0	RORMIS	如果当 Rx FIFO 状态为满时另外一帧被完全接收了，该位为 1 且该中断被使能。	0
1	RTMIS	当 Rx FIFO 不为空且超过超时周期没有被读取时，该位为 1 且该中断被使能。对于主机和从机模式来说超时周期是相同的，由 SSP 比特率决定：在 $PCLK/([CPSDVSR+2] \times [SCR+1])$ 时为 32 位。	0
2	RXMIS	如果 Rx FIFO 为至少半满状态，该位为 1 且该中断被使能。	0

3	TXMIS	如果 Tx FIFO 为至少半空状态, 该位为 1 且该中断被使能。	0
31:4	-	保留	0

#### 9.4.9 中断清除寄存器(ICR – 0x5001 8020)

软件可向该只写寄存器写入 1 个或多个 1 以清除 SPI 控制器中相应的中断条件。注意其他两个中断条件可以通过写入或者读取正确的 FIFO 来清除, 或者通过清除 IMSC 寄存器中对应的位值来禁用。

表 9.4.9.1 中断清除寄存器(ICR - 0x5001 8020) 位描述

位	符号	值	描述
0	RORIC	对该位写入 1 将清除“当 Rx FIFO 为满状态时帧已接收”中断。	0
1	RTIC	对该位写入 1 将清除“Rx FIFO 不为空且超过超时周期没有被读取”中断。对于主机和从机模式来说超时周期是相同的, 由 SSP 比特率决定: 在 $PCLK/([CPSDVS+2] \times [SCR+1])$ 时为 32 位。	0
31:2	-	保留	0

### 9.5 功能说明

#### 9.5.1 德州仪器同步串行帧格式

图 9.5.1.1 显示了 SPI 模块支持的 4 线德州仪器同步串行帧格式。

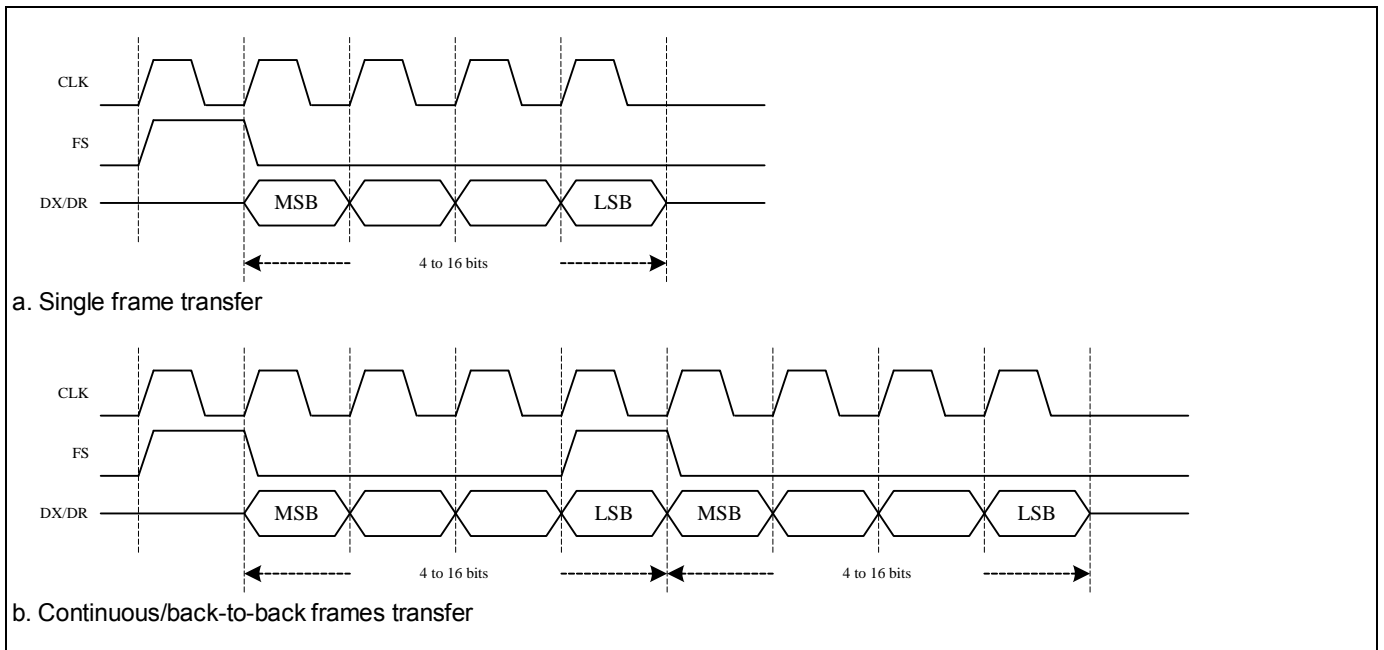


图 9.5.1.1 德州仪器同步串行帧格式: a) 单帧和 b) 连续 / 背靠背 2 帧传输

对于在该模式下配置为主机的设备, CLK 和 FS 被强制为低电平, 且只要 SSP 空闲, 发送数据线 DX 便会处于 3 态模式。一旦发送 FIFO 的底部入口包含数据, FS 将被脉冲触发为一个 CLK 周期的高电平。将要发送的值也被从发送 FIFO 传输到发送逻辑中的串行移位寄存器。在下一个 CLK 的上升沿, 4 至 16 位数据帧的 MSB 被移出至 DX 引脚。同样, 接收数据的 MSB 由片外串行从设备传送到 DR 引脚。

在每个 CLK 的下降沿, SSP 和片外串行从设备将各个数据位放入其串行移位器。LSB 被锁存后, 在 CLK 的第一个上升沿, 接收的数据从串行移位器传输到接收 FIFO。

#### 9.5.2 SPI 帧格式

SPI 接口是 4 线接口，其中 SSEL 信号用作从机选择。SPI 格式的主要特性是 SCK 信号的非工作状态和相位可通过对 CR0 控制寄存器内的 CPOL 和 CPHA 位编程设定。

### 9.5.2.1 时钟极性 (CPOL) 及相位 (CPHA) 控制

当 CPOL 时钟极性控制位为低电平时，它会在 SCK 引脚产生一个稳定状态的低电平值。如果 CPOL 时钟极性控制位为高电平，则在没有传输数据时，它会在 CLK 引脚上产生一个稳态高电平值。

CPHA 控制位选择捕获数据及允许数据更改状态的时钟沿。通过是否在第一个数据捕获沿发生之前允许时钟跳变，该位能最大程度上影响第一个被发送的位。当 CPHA 相位控制位为低电平时，则在第一次出现时钟沿跳变时捕获数据。如果 CPHA 时钟相位控制位为高电平，则在第二次出现时钟沿跳变时捕获数据。

### 9.5.2.2 CPOL=0, CPHA=0 时的 SPI 格式

CPOL = 0 且 CPHA = 0 时 SPI 格式的单帧和连续帧传输信号序列如图 9.5.2.2.1 所示。

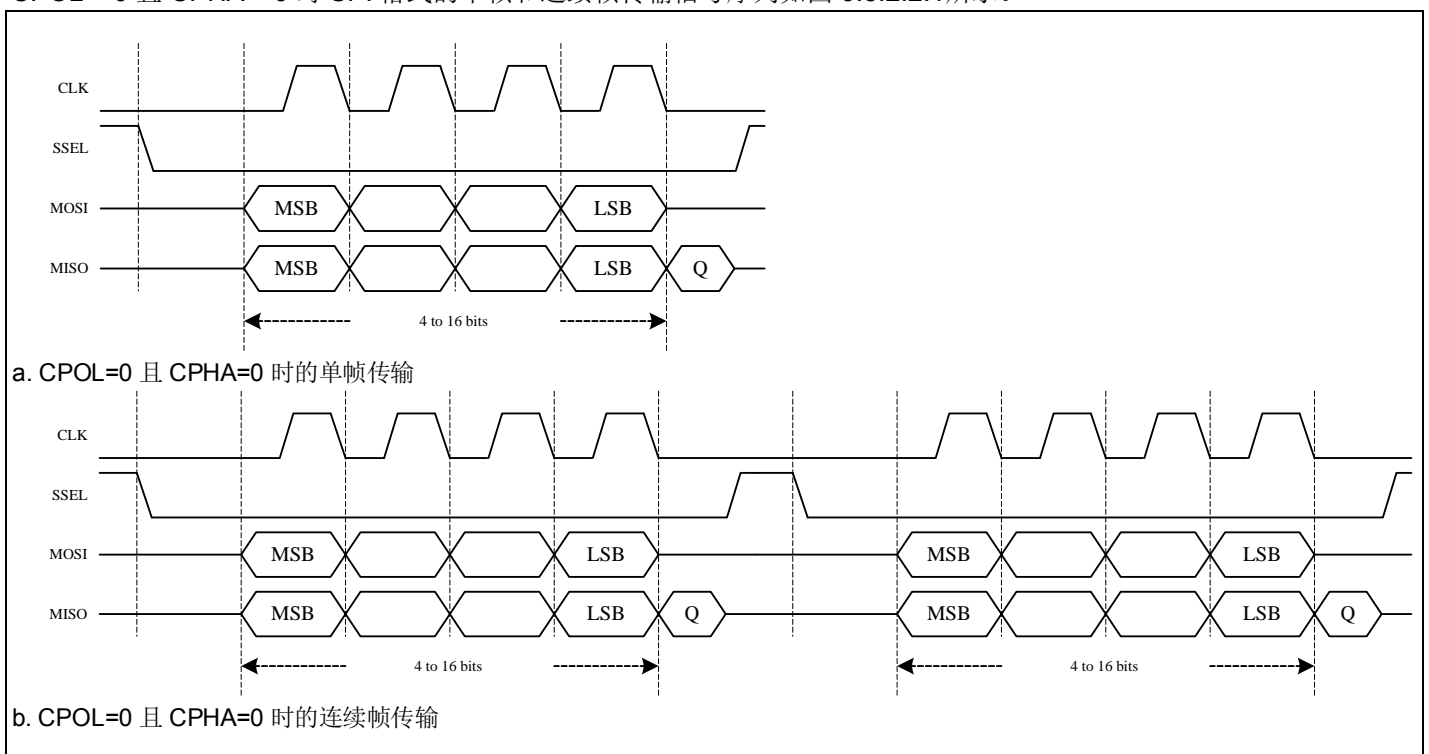


图 9.5.2.2.1 CPOL=0 且 CPHA=0 时的 SPI 帧格式 (a) 单帧和 b) 连续帧传输)

在这种配置下，空闲期间：

- CLK 信号被强制为低电平。
- SSEL 被强制为高电平。
- 发送 MOSI/MISO 垫片处于高阻态。

如果使能 SSP/SPI，且在发送 FIFO 中存在有效数据，则 SSEL 主机信号被驱动为低电平，指示数据传输开始。这将使得从机数据被连接至主机的 MISO 输入线上。使能主机的 MOSI。

1/2 个 SCK 周期后，有效的主机数据将被传输到 MOSI 引脚。由于主机和从机数据均已设定，因此再过 1/2 个 SCK 周期，SCK 主时钟引脚就会变为高电平。

在 SCK 信号的上升沿捕获数据，并在 SCK 信号的下降沿传播数据。

在单字发送的情况下，当数据字的所有位都传输完毕以后，SSEL 线在最后一位被捕获后的一个 SCK 周期后回到空闲高电平状态。

但是，在进行连续的背靠背传输时，传输各数据字之间的 SSEL 信号必须为高电平。这是因为如果 CPHA 位为逻辑值 0，从机选择引脚会冻结其串行外设寄存器中的数据，并且不允许其被更改。因此主机装置必须在每个数据传输之间升高从机装置的 SSEL 引脚上的电平以使能串行外设数据写入。连续传输完成后，SSEL 引脚在捕获到最后一位后的一个 SCK 周期内将返回至空闲状态。

### 9.5.2.3 CPOL=0, CPHA=1 时的 SPI 格式

CPOL = 0 且 CPHA = 1 时 SPI 格式的传输信号序列如图 9.5.2.3 所示，包含单帧和连续帧传输。

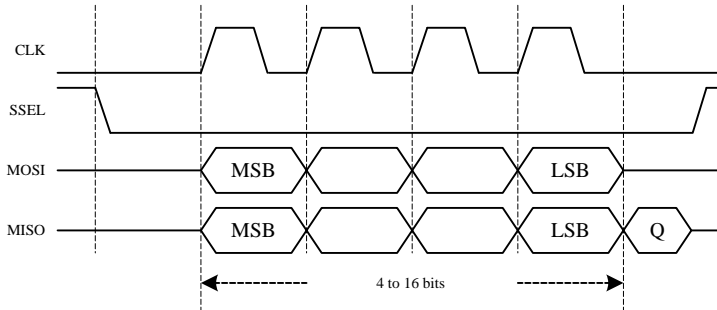


图 9.5.2.3.1 CPOL=0 且 CPHA=1 时的 SPI 帧格式

在这种配置下，空闲期间：

- CLK 信号被强制为低电平。
- SSEL 被强制为高电平。
- 发送 MOSI/MISO 垫片处于高阻态。

如果使能 SSP/SPI，且在发送 FIFO 中存在有效数据，则 SSEL 主机信号被驱动为低电平，指示数据传输开始。使能主机的 MOSI 引脚。在又过了半个 SCK 周期以后，主机和从机的有效数据都被连接至各自的发送线上。同时，SCK 在上升沿跳变时使能。

然后数据将在 SCK 信号下降沿捕获，并且在上升沿传播。

在单字传输的情况下，当所有位都传输完毕以后，SSEL 线在最后一位被捕获后的一个 SCK 周期后回到空闲高电平状态。

对于连续背靠背传输，SSEL 引脚在连续数据字之间保持为低电平，结束过程与单字传输相同。

### 9.5.2.4 CPOL = 1 且 CPHA = 0 时的 SPI 格式

CPOL=1 且 CPHA=0 时，SPI 格式的单帧和连续帧传输信号序列如图 9.5.2.4.1 所示。



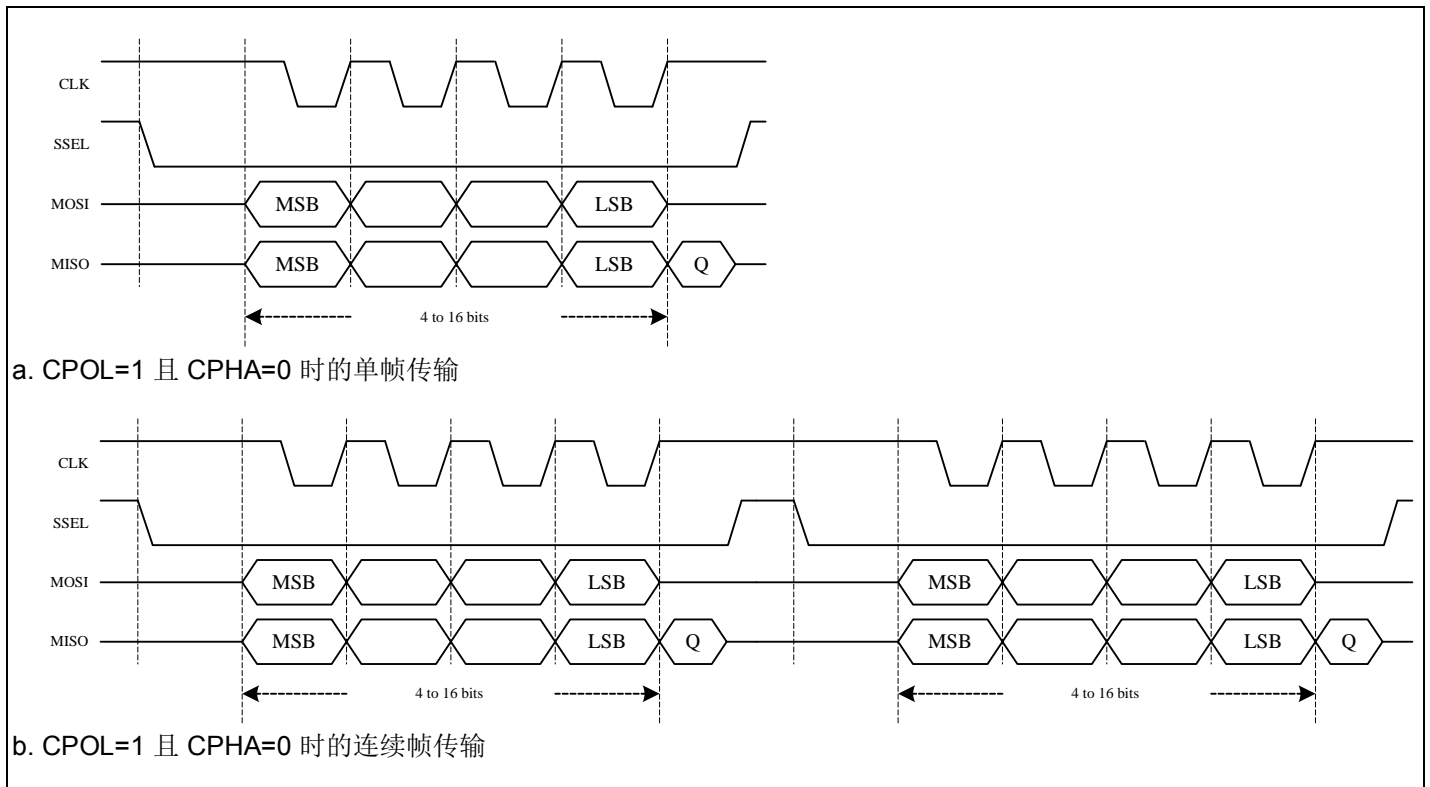


图 9.5.2.4.1 CPOL = 1 且 CPHA = 0 时的 SPI 帧格式 (a) 单帧和 b) 连续帧传输)

在这种配置下，空闲期间：

- CLK 信号被强制为高电平。
- SSEL 被强制为高电平。
- 发送 MOSI/MISO 垫片处于高阻态。

如果使能 SSP/SPI，且在发送 FIFO 中存在有效数据，则 SSEL 主机信号被驱动为低电平，指示数据传输开始，这会使从机数据立即传输到主机MISO线上。使能主机MOSI引脚。

1/2 个SCK 周期后，有效的主机数据将被传输到MOSI 线。现在主机和从机的数据都被设置好了，再过半个SCK 周期SCK 主机时钟引脚转至低电平。这意味着，在SCK 信号的下降沿捕获数据并在SCK 信号的上升沿传播数据。

在单字发送的情况下，当数据字的所有位都传输完毕以后，SSEL 线在最后一位被捕获后的一个SCK 周期后回到空闲高电平状态。

但是，在进行连续的背靠背传输时，传输各数据字之间的SSEL 信号必须为高电平。这是因为如果CPHA 位为逻辑值 0，从机选择引脚会冻结其串行外设寄存器中的数据，并且不允许其被更改。因此主机装置必须在每个数据传输之间升高从机装置的SSEL 引脚上的电平以能使串行外设数据写入。连续传输完成后，SSEL 引脚在捕获到最后一位后的一个SCK周期内将返回至空闲状态。

### 9.5.2.5 CPOL = 1 且 CPHA = 1 时的 SPI 格式

CPOL = 1 且CPHA = 1 时SPI 格式的传输信号序列如图 9.5.2.5.1所示，包含单帧和连续帧传输。



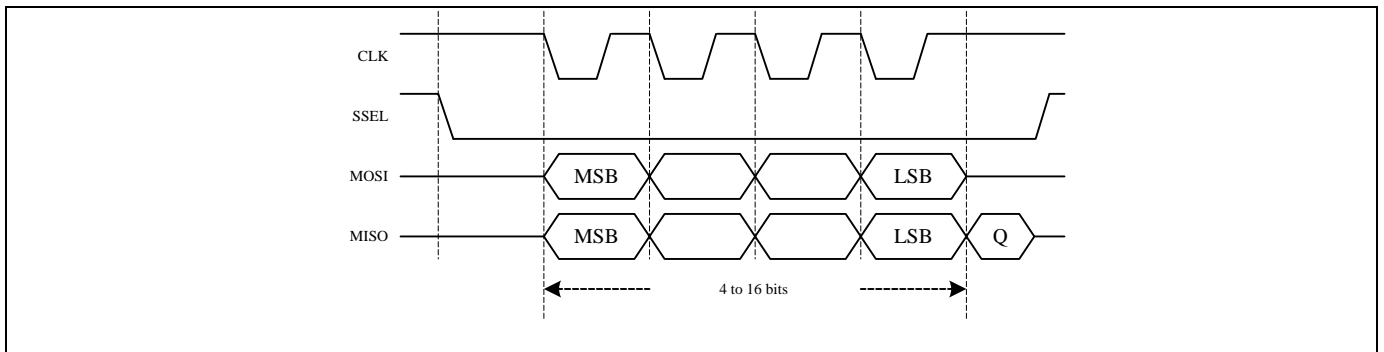


图 9.5.2.5.1 CPOL = 1 且 CPHA = 1 时的 SPI 帧格式

在这种配置下，空闲期间：

- CPOL = 1 且 CPHA = 1 时的 SPI 帧格式。
- SSEL 被强制为高电平。
- 发送 MOSI/MISO 垫片处于高阻态。

如果使能 SSP/SPI，且在发送 FIFO 中存在有效数据，则 SSEL 主机信号被驱动为低电平，指示数据传输开始。使能主机的 MOSI。在又过了半个 SCK 周期以后，主机和从机的数据都被连接至各自的发送线上。同时 SCK 被一个下降沿跳变使能。然后，在 SCK 信号的上升沿捕获数据，并在 SCK 信号的下降沿传播数据。

发送单个字时，在传输完所有位后，在捕获最后一位后的一个 SCK 周期内，SSEL 线返回到其空闲高电平状态。

对于连续背靠背发送，SSEL 引脚保持有效低电平状态，直到最后一个字的最后一位被捕获，然后如上所述回到其空闲状态。通常，当进行连续的背靠背传输时，连续的数据字之间 SSEL 引脚保持为低电平，终止条件与传送单个字时相同。

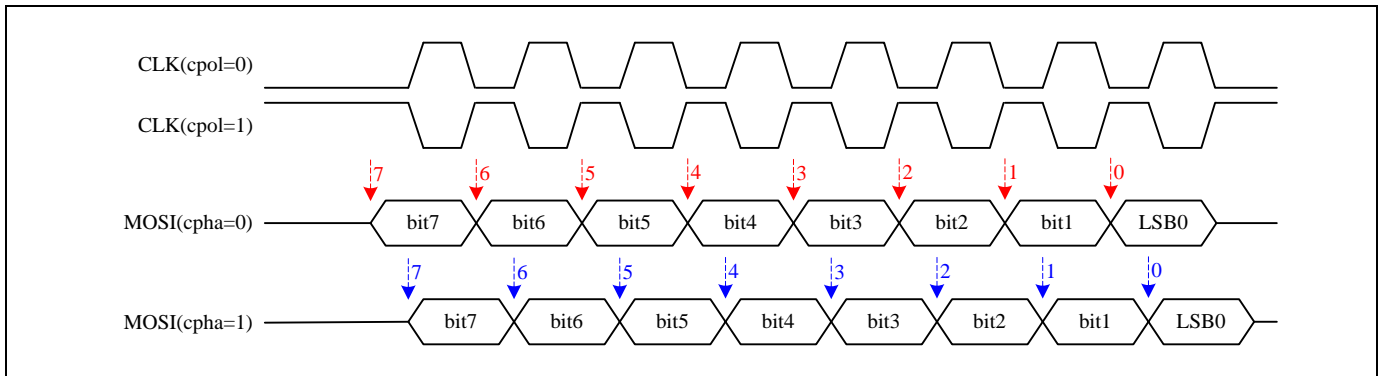


图 9.5.2.5.2 SPI\_MS 发送帧格式

### 9.5.3 半导体 Microwire 帧格式

图 9.5.3.1 显示了单帧的 Microwire 帧格式。图 9.5.3.2 显示了进行背靠背帧发送时的 Microwire 帧格式。

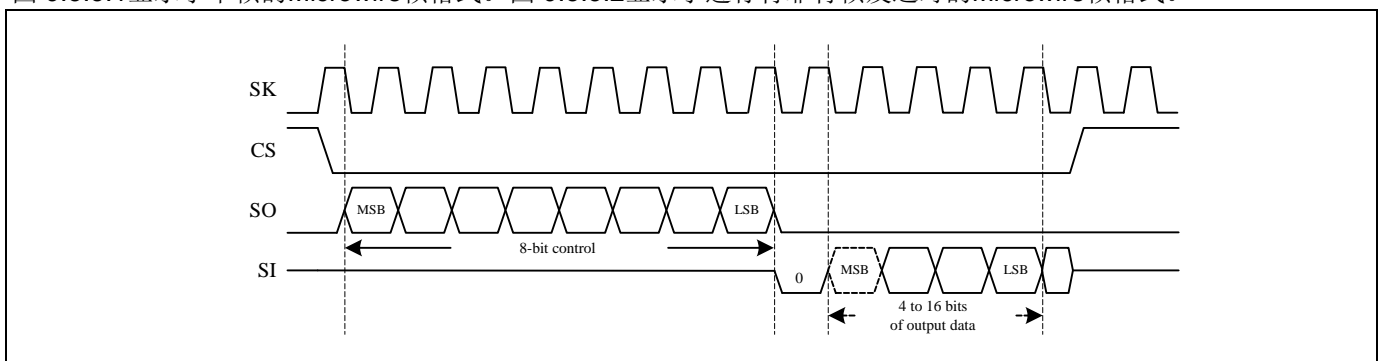


图 9.5.3.1 Microwire 帧格式（单帧传输）



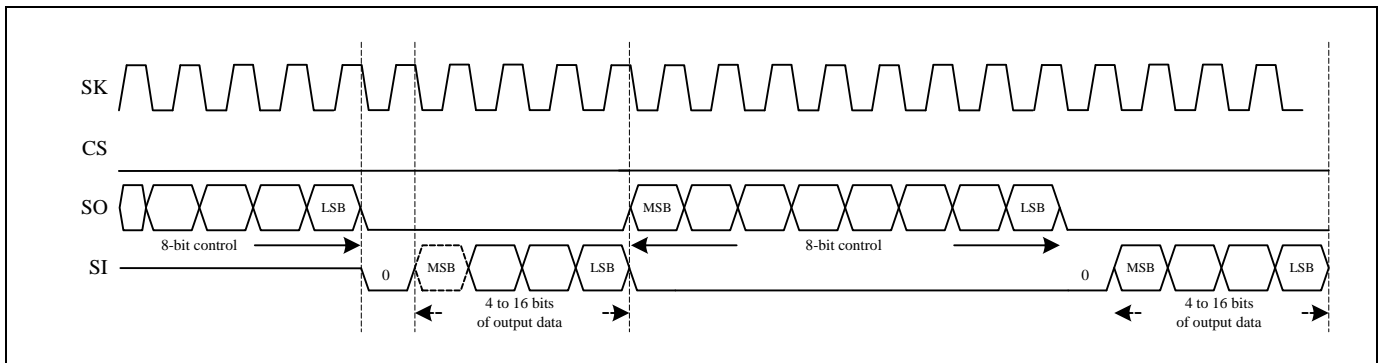


图 9.5.3.2 Microwire 帧格式（连续传送）

Microwire 格式与SPI 格式非常相似，都是使用主-从消息传递技术，但它采用的传输方式是半双工方式而不是全双工方式。每次串行传输均从一个8 位控制字开始，由SSP/SPI 发送到片外从机设备。在此发送过程中，SSP/SPI 不会接收输入的数据。发送完消息后，片外从机对其进行解码，然后在8 位控制消息的最后一位发送结束后再等待一个串行时钟，以所需的数据进行响应。返回的数据长度为4 到16 位，使整个帧长度范围为13 到25 位。

在这种配置下，空闲期间：

- SK 信号被强制为低电平。
- CS 被强制为高电平。
- 发送数据线SO 被任意强制为低电平。

通过向发送FIFO 写入控制字节来启动传送。CS 下降沿会使包含在发送FIFO 底端的值传输到发送逻辑的串行移位寄存器，并使8 位控制帧的MSB 输出到SO 引脚。CS 在帧传输期间保持低电平。SI 引脚在此发送过程中保持三态。

片外串行从设备在每个SK 的上升沿将各控制位锁存到串行移位器中。当从机设备将最后一位锁存后，控制字节会在一个时钟等待状态期间被解码，而从机则通过将数据传回至SSP/SPI 来进行响应。每个位在SK 的下降沿被驱动到SI 线。SSP/SPI 依次在SK 的上升沿锁存每个位。对于单帧传输，在帧的末端，在最后一位已锁存到接收串行移位器的一个时钟周期内，CS 信号会被置为高电平，这会导致数据被传输到接收FIFO。

注：LSB 被接收移位器锁存后或当CS 引脚变为高电平时，片外从机设备的接收线在SK 下降沿呈现三态。

对于连续帧传输，数据传输开始和结束的方式与单帧传输相同。然而，CS 线持续有效（保持低电平），且数据以背靠背方式进行传输。当前数据帧的LSB 被接收后，下一帧的控制字节立即发送。在帧数据的LSB 被锁存到SSP/SPI 后，每个收到的值将在SK 的下降沿从接收移位器传输。

### 9.5.3.1 Microwire 模式下 CS 相对于 SK 的建立和保持时间要求

在Microwire 模式下，CS 变为低电平后，SSP/SPI 从机在SK 上升沿时对接收数据的第一位进行采样。主机驱动SK 自由运行时必须确保CS 信号相对于SK 上升沿有充足的建立和保持时间。

图 9.5.3.1.1描绘了这些建立和保持时间的要求。相对于SK 上升沿（SSP/SPI 从机在该上升沿对接收数据的第一位进行采样），CS 的建立时间必须至少为SK （SSP/SPI 在SK 上运行）周期的2 倍。相对于该边沿之前的SK 上升沿，CS 必须保持至少一个SK 周期。

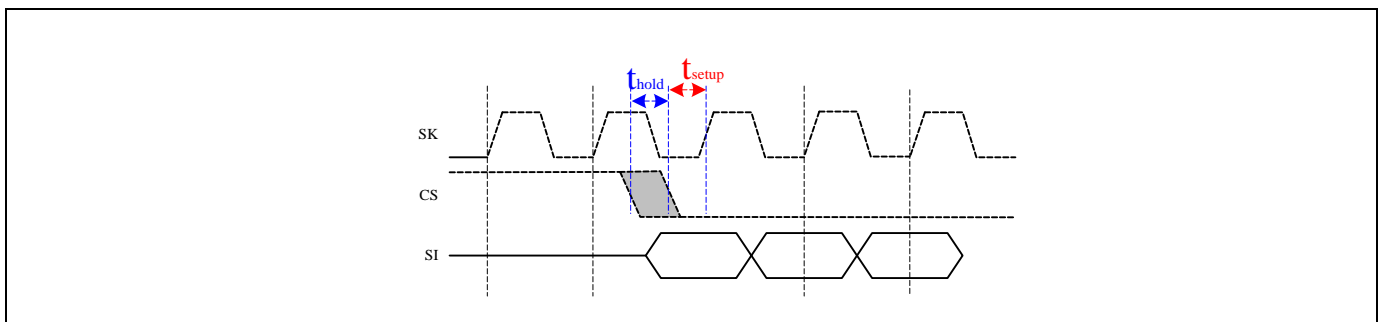


图 9.5.3.1.1 Microwire 帧格式建立及保持时间

9.5.4 SSNP 功能

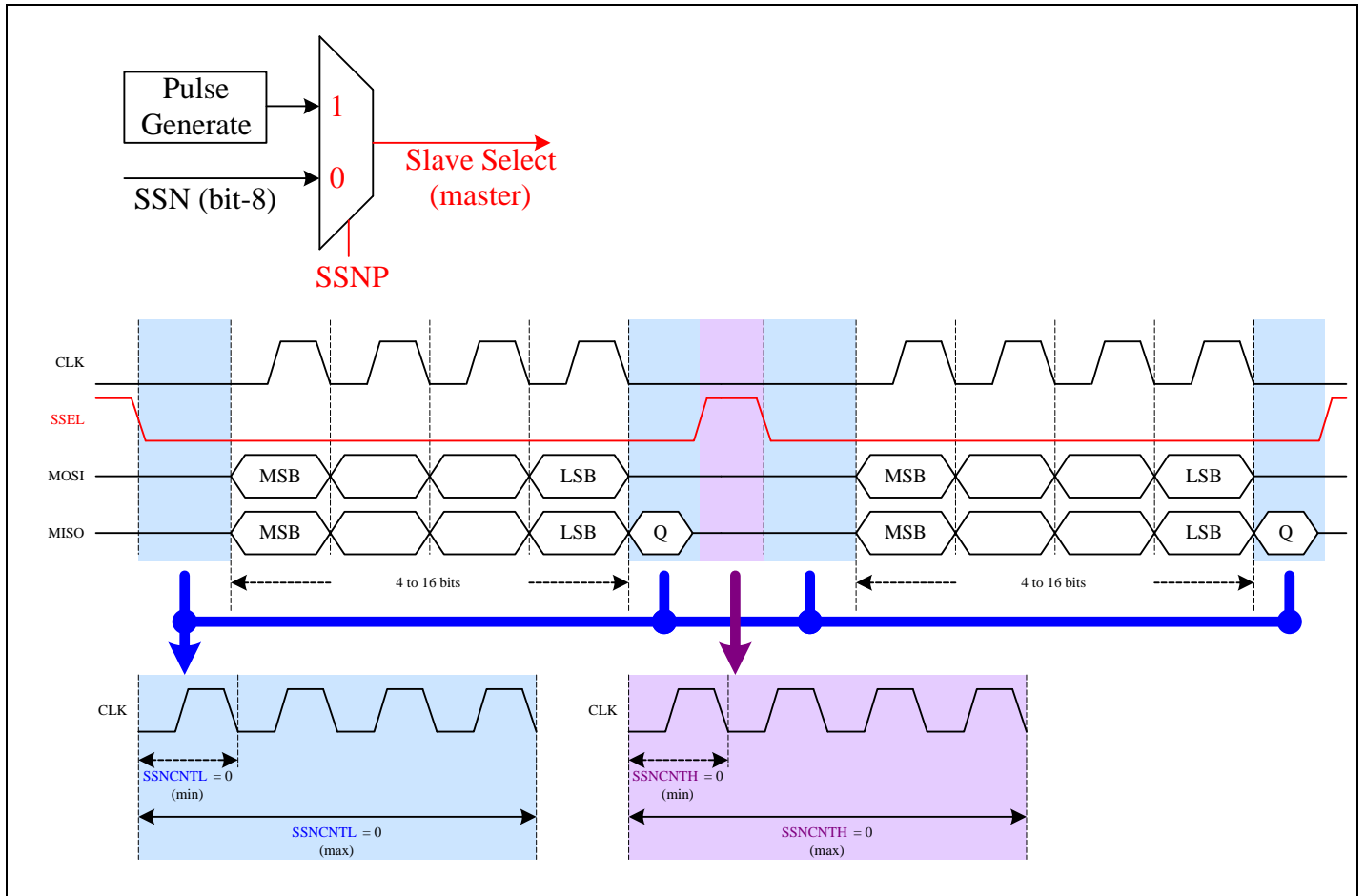


图 9.5.4.1 SPI帧格式，SSNP = 1，CPOL = 0，CPHA = 0连续传输

## 10. UART

### 10.1 特性

- 一个 UART
- 16 字节接收和发送FIFO。
- 接收器FIFO 触发点在1、4、8 和14 字节。
- 内置波特率发生器。
- UART允许实现软件或硬件流量控制。
- RS-485/EIA-485 9 位模式支持输出使能。

### 10.2 引脚描述

表 10.2.1 UART 引脚描述

引脚	类型	描述
RXD	输入	串行输入。串行接收数据。
TXD	输出	串行输出。串行发送数据。

### 10.3 寄存器描述

UART 所包含的寄存器，其结构如表 10.3.1所示。除数锁存器访问位(DLAB) 包含在ULCR 寄存器的位7 中，可实现对除数锁存器的访问。

重置值仅反映存储在使用位中的数据。它不包含保留位的内容。

表 10.3.1 寄存器简介：

UART0 (基址: 0x5002 8000)

名称	访问类型	地址偏移	描述	重置值
URBR	RO	0x00	接收缓冲寄存器。包含下一个待读的已接收字符。	0
UTHR	WO	0x00	发送保持寄存器。下一个待发送字符写入在此。	0
-	-	0x04	保留	0x01
UDL	R/W	0x08	除数锁存器。波特率除数值的最低有效字节。完整的除数用于从分数率分频器生成波特率。	0x00
UIER	R/W	0x0C	中断使能寄存器。包含7个潜在UART中断的各个中断使能位。	0x00
UIIR	RO	0x10	中断ID寄存器，识别要挂起的中断。	0x01
UFCR	WO	0x14	FIFO控制寄存器，控制UART FIFO的使用和模式。	0x00
UFRC	R	0x18	FIFO RX计数器寄存器。	0x00
UFTC	R	0x1C	FIFO TX计数器寄存器。	0x00
ULCR	R/W	0x20	线路控制寄存器。包含对帧格式化和断点产生的控制	0x00
ULSR	RO	0x24	线路状态寄存器，包含发送和接收状态（包括线路错误）的标志。	0x60
UACR	R/W	0x28	自动波特率控制寄存器，包含自动波特率功能的控制。	0x00
-	-	0x2C - 0x48	保留	-
URS485CTRL	R/W	0x4C	RS-485/EIA-485 控制，包含对配置RS-485/EIA-485模式各个方面的控制。	0x00
URS485ADRMATCH	R/W	0x50	RS-485/EIA-485地址匹配，包含RS-485/EIA-485模式的地址匹配值。	0x00

- 0x54 保留 -

### 10.3.1 UART 接收器缓冲寄存器(URBR – 0x5002 8000)

URBR 是UART RX FIFO 的最高字节。RX FIFO 最高字节包含了接收到的最旧的字符，可以通过总线接口读取。LSB（位0）包含最先接收到的数据位。如果接收到的字符少于8位，则未使用的MSB用0填充。

要访问URBR，ULCR 中的除数锁存器访问位(DLAB)必须为零。URBR 始终为“只读”。

由于奇偶错误(PE)、帧错误(FE)和间隔中断(BI)位与RBR FIFO 顶部的字节（即下次从RBR 读取时要读取的字节）相对应，因此，要正确提取有效的接收字节对其状态位，应先读取ULSR 寄存器的内容，然后再读取RBR 中的字节。

表 10.3.1.1 UART 接收器缓冲寄存器 (URBR - 0x5002 8000 (UART0),当DLAB = 0 时, 只读) 位描述

位	符号	值	描述
7:0	RBR		UART 接收缓冲寄存器包含在 UART RX FIFO 中最早收到的字节。
31:8	-		保留

### 10.3.2 UART 发送器保持寄存器(UTHR – 0x5002 8000)

UTHR 是UART TX FIFO 的最高字节。最高字节是TX FIFO 中最新的字符，可通过总线接口写入。LSB 代表第一个要发送的位。

要访问UTHR，ULCR 中的除数锁存器访问位(DLAB)必须为零。UTHR 始终为“只写”。

表 10.3.2.1 UART发送器保持寄存器(UTHR - 0x5002 8000 (UART0),当DLAB = 0 时, 只写) 位描述

位	符号	值	描述
7:0	THR		写入 UART 发送保持寄存器的数据将存储在 UART 发送 FIFO 中。字节将在其变为 FIFO 最旧字节且发送器可用时发出。
31:8	-		保留

### 10.3.3 UART 除数锁存寄存器 (UDL – 0x5002 8008)

UART 除数锁存器是 UART 波特率发生器的一部分，持有所用的值（可选择包含小数分频器），以对 UART\_PCLK 时钟进行分频，产生必须是过采样寄存器所指定的所需波特率倍数（通常为 16X）的波特率时钟。UDL 寄存器是一个 16 位分频器。

0x0000 值被视为 0x0001 值，因为被零除是不允许的。

表 10.3.3.1 UART除数锁存寄存器 (UDL - 0x5002 8008 (UART0),位描述

位	符号	值	描述
15:0	UDL		UART 除数锁存寄存器决定 UART 的波特率。
31:16	-		保留

### 10.3.4 UART 中断使能寄存器(UIER – 0x5002 800C)

UIER 用于使能 3 个 UART 中断源。

表 10.3.4.1 UART中断使能寄存器 (UIER - 0x5002 800C (UART0),位描述

位	符号	值	描述	重置值
0	RBRIE		RBR 中断使能。使能接收数据可用中断。也控制字符接收超时中断。	0
		0	禁用 RDA 中断。	
1	THREIE	1	使能 RDA 中断。	0
		0	禁用 THRE 中断。	
2	RXLIE	1	THRE 中断使能。使能 THRE 中断。此中断的状态可从 ULSR[5]读取。	0
		0	禁用 THRE 中断。	
7:3	-	0	RX 线路中断使能。使能 RX 线路状态中断。此中断的状态可从 ULSR[4:1]读取。	0
		1	禁用 RX 线路状态中断。	
8	ABEOINTEN	1	使能 RX 线路状态中断。	0
		0	保留	
			使能自动波特率结束中断。	0

	0	禁用自动波特率结束中断。	
	1	使能自动波特率结束中断。	
9	ABTOINTEN	使能自动波特率超时中断。	0
	0	禁用自动波特率结束中断。	
	1	使能自动波特率超时中断。	
31:10	-	保留。	0

### 10.3.5 UART 中断识别寄存器 (UIIR – 0x5002 8010)

UIIR提供一个状态代码，表示挂起中断的优先级和源。

这些中断在UIIR访问期间被冻结。如果在UIIR访问期间出现中断，则为下次UIIR访问记录该中断。

表 10.3.5.1 UART中断识别寄存器(UIIR - 0x5002 8010 (UART0),只读) 位描述

位	符号	值	描述	重置值
0	INTSTATUS		中断状态。请注意，UIIR[0]低电平激活。挂起中断可通过评估 UIIR[3:1]来确定。	1
		0	至少一个中断挂起。	
		1	没有中断挂起。	
3:1	INTID		中断识别。UIER[3:1] 识别与 UART Rx FIFO 对应的中断。下面未列出的 UIER[3:1] 的其他值均为保留值。(100,101,111)	0
		0x3	1 - 接收线路状态(RLS).	
		0x2	2a - 接收数据可用(RDA).	
		0x2	2b - 字符超时指示(CTI).	
		0x6	3 - THRE 中断。	
5:4	-	-	保留	0
7:6	FIFOENABLE		这些位相当于 UFCR[0]。	0
8	ABEOINT		自动波特率结束中断。自动波特率成功完成且中断使能时为真。	0
9	ABTOINT		自动波特率超时中断。自动波特率超时且中断使能时为真。	0
31:10	-	-	保留	0

位UIIR[9:8] 由自动波特率函数设置，并发出超时信号或自动波特率条件结束信号。通过设置自动波特率控制寄存器中的相应“清除”位，清除自动波特率中断条件。

如果IntStatus位为1且没有中断挂起，则IntId位将为零。如果IntStatus为0，表示有一个非自动波特率中断被挂起，此时IntId位会确定中断的类型和处理方式，如表 10.3.5.1中所述。如果指定UIIR[3:0] 的状态，中断处理程序就能确定中断原因以及如何清除有效的中断。必须读取UIIR，才能在退出中断服务例程前清除中断。

UART RLS中断(UIIR[3:1] = 011)为最高优先级中断，每当在UART RX输入端发生下面4个错误状况中的任意一个时，都可以进行设置：溢出错误(OE)、奇偶校验错误(PE)、帧处理错误(FE) 和中止中断(BI)。设置中断的UART Rx 错误状况可通过ULSR[4:1]查看。该中断在读取ULSR时被清除。

UART RDA中断(UIIR[3:1] = 010) 与CTI中断(UIIR[3:1] = 010)均为第二优先级。RDA 在UART Rx FIFO达到 UFCR7:6中定义的触发级别时激活，并在UART Rx FIFO降低到触发级别以下时重置。RDA中断激活时，CPU可读取触发级别定义的数据块。

CTI中断(UIIR[3:1] = 010)为第二优先级中断，在UART RX FIFO含有至少一个字符并且在接收到3.5到4.5个字符的时间内没有发生任何UART RX FIFO操作时，可对该中断进行设置。任何UART Rx FIFO操作（读或写UART RSR）都将清除中断。此中断的目的是在收到一条大小不是触发级别大小的倍数的消息后，刷新UART RBR。例如，如果要发送一条105个字符的消息，并且触发级别为10个字符，则CPU将接收10次RDA中断以传输100个字符，还将接收1到5次CTI中断（取决于服务例程）以传输剩余的5个字符。

表 10.3.5.2 UART 中断处理

UIIR[3:0] 优先级	中断类型	中断源	中断重置
0001	无	无	
0110	最高	RX 线路状态/错误	ULSR 读操作
0100	第二	RX 数据可用	URBR 读操作或 UART FIFO 低于触发级别
0100	第二	RX 数据可用或触发级别达到 FIFO (FCR0=1) 用	URBR 读操作
0100	第二	字符超时指示	URBR 读操作
1100	第三	THRE	UIIR 读操作 (如果是中断源) 或 THR 写操作

UART THRE 中断(UIIR[3:1] = 110)为第三优先级中断, 当UART THR FIFO为空, 且满足特定的初始化条件时, 该中断激活。这些初始化条件的目的是为UART UTHR FIFO提供填充数据的机会, 以消除在系统启动时出现的许多THRE中断。当THRE = 1时, 且在上次的THRE = 1事件后, THR中没有出现至少两个字符时, 这些初始化条件就会实现一个字符减去停止位的延时。提供此延迟的目的是让CPU有时间将数据写入UTHR, 而无需THRE中断来进行解码和服务。如果UART UTHR FIFO曾一次持有两个或多个字符, 并且UTHR当前被清空, 则THRE 中断立即设置。当发生UTHR 写操作或UIIR读操作, 且THRE为最高优先级中断(UIIR[3:1] = 110) 时, THRE中断重置。

### 10.3.6 UART FIFO 控制寄存器(UFCR – 0x5002 8014)

UFCR 控制 UART RX 和 TX FIFO 的操作。

表 10.3.6.1 UART FIFO控制寄存器(UFCR - 0x5002 8014 (UART0),只写) 位描述

位	符号	值	描述	重置值
0	FIFOEN		FIFO 使能	0
		0	UART FIFO 禁用, 不得用于应用程序。	
		1	为 UART Rx 和 TX FIFO 以及 UFCR[7:1] 访问使能有效高电平。必须设置此位, UART 才能正确操作。此位上的任何跳变都将自动清除 UART FIFO。	
1	RXFIFORES		RX FIFO 重置	0
		0	对任一 UART FIFO 都没有影响。	
		1	将逻辑 1 写入 UFCR[1]将清除 UART Rx FIFO 中的所有字节, 重置指针逻辑。此位是自清除的。	
2	TXFIFORES		TX FIFO 重置	0
		0	对任一 UART FIFO 都没有影响。	
		1	将逻辑 1 写入 UFCR[2]将清除 USART TX FIFO 中的所有字节, 重置指针逻辑。此位是自清除的。	
5:3	-	-	保留	0
7:6	RXTL		RX 触发级别。这两个位确定在激活中断前, FIFO 必须接收多少 UART FIFO 字符。	0
		0x0	触发级别 0 (1 个字符或 0x01)	
		0x1	触发级别 1 (4 个字符或 0x04)	
		0x2	触发级别 2 (8 个字符或 0x08)	
		0x3	触发级别 3 (14 个字符或 0x0E)	
31:8	-	-	保留	-

### 10.3.7 UART FIFO RX 计数器寄存器(UFRC – 0x5002 8018)

UFRC 是 UART RX FIFO 的编号。

表 10.3.1.1 UART FIFO RX 计数器寄存器(UFCR - 0x5002 8018 (UART0),位描述

位	符号	值	描述
---	----	---	----



3:0	UFRC	UFRC 是 UART RX FIFO 的编号	0
31:4	-	保留	

### 10.3.8 UART FIFO TX 计数器寄存器(UFTC – 0x5002 801C)

UFTC 是 UART TX FIFO 的编号

表 10.3.1.1 UART FIFO TX 计数器寄存器(UFTC - 0x5002 801C (UART0),位描述

位	符号	值	描述	重置值
3:0	UFTC	UFTC 是 UART TX FIFO 的编号		0
31:4	-	保留		

### 10.3.9 UART 线路控制寄存器(ULCR – 0x5002 8020)

ULCR 确定要发送或接收的数据字符的格式。

表 10.3.9.1 UART 线路控制寄存器(ULCR - 0x5002 8020 (UART0), 位描述

位	符号	值	描述	重置值
2:0	WLS		字长度选择	0
		0x0	5 位字符长度。	
		0x1	6 位字符长度。	
		0x2	7 位字符长度。	
		0x3	8 位字符长度。	
3	SBS		停止位选择	0
		0	1 停止位。	
		1	2 停止位（如果 ULCR[1:0]=00，则为 1.5）。	
4	PE		奇偶校验使能	0
		0	禁用奇偶校验生成和检查。	
		1	使能奇偶校验生成和检查。	
6:5	PS		奇偶检验选择	0
		0x0	奇数校验。已发送字符中 1 的数量，附加的奇偶校验位将为奇数。	
		0x1	偶数校验。已发送字符中 1 的数量，附加的奇偶校验位将为偶数。	
		0x2	强制 1 奇偶校验。	
		0x3	强制 0 奇偶校验。	
31:7	-	-	保留	-

### 10.3.10 UART 线路状态寄存器(ULSR – 0x5002 8024)

ULSR 是一个只读寄存器，提供有关 UART TX 和 RX 模块的状态信息。

表 10.3.10.1 UART 线路状态寄存器(ULSR - 0x5002 8024 (UART0),只读) 位描述

位	符号	值	描述	重置值
0	RDR		接收器数据就绪：当 URBR 包含未读字符时，ULSR[0]会被设置；当 UART URBR 0 FIFO 为空时，ULSR[0]会被清零。	0
		0	URBR 被清空。	
		1	URBR 包含有效数据。	
1	OE		超限错误。超限错误条件在其出现时立即设置。ULSR 读取将清除 ULSR[1]。在 UART 0 RSR 汇编了新字符且 UART RBR FIFO 为满时，设置 LSR[1]。在此情况下，UART RBR FIFO 将不会被覆盖，且 UART RSR 中的字符将丢失。	0
		0	超限错误状态未激活。	
		1	超限错误状态激活。	
2	PE		奇偶校验错误。已接收字符的奇偶校验位处于错误状态时，出现奇偶校验错误。ULSR 0 读取将清除 LSR[2]。奇偶校验错误检测的时间取决于 UFRC[0]。 注：奇偶校验错误与 UART RBR FIFO 顶部的字符相关。	0
		0	奇偶校验错误状态未激活。	
		1	奇偶校验错误状态激活。	

3	FE	帧处理错误。已接收字符的停止位为逻辑0时，出现帧处理错误。ULSR读取将清除0	0	帧处理错误状态未激活。
		注：帧处理错误与 UART RBR FIFO 顶部的字符相关。	1	帧处理错误状态激活。
4	BI	中止中断。对于一个完整的字符发送（开始、数据、优先级、停止），RXD保持在间距0	0	中止中断状态未激活。
		状态（全部为零）时，发生断点中断。检测到断点条件后，接收器即进入空闲，直至RXD进入标记状态（全部为1）为止。ULSR读取将清除此状态位。	1	中止中断状态激活。
5	THRE	发送保持寄存器空。THRE 在检测到空 UART UTHR 时立即设置，并在 UTHR 写入时清1	0	UTHR 包含有效数据。
		除。	1	UTHR 为空。
6	TEMT	发送器空。TEMT 在 UTHR 和 UTSR 都为空时设置；TEMT 在 UTSR 或 UTHR 包含有1	0	UTHR 和/或 UTSR 包含有效数据。
		有效数据时被清除。	1	UTHR 和 UTSR 为空。
7	RXFE	RX FIFO 中的错误。在具有 RX 错误（如帧处理错误、奇偶校验错误或断点错误）的字0	0	URBR 不包含 UART RX 错误或 UFCR[0]=0。
		符被加载到 URBR 中时，设置 ULSR[7]。此位在 ULSR 寄存器被读取且 UART FIFO 中	1	UART RBR 包含至少一个 UART RX 错误。
31:8	-	保留	-	-

### 10.3.11 UART 自动波特率控制寄存器(UACR – 0x5002 8028)

UART自动波特率控制寄存器(UACR)控制为波特率生成测量输入时钟/数据速率的过程，并且可由用户决定读取和写入。

表 10.3.11.1 自动波特率控制寄存器(UACR - 0x5002 8028 (UART0)) 位描述

位	符号	值	描述	重置值
0	START		开始位。自动波特率完成后此位自动清除。	0
		0	自动波特率停止（自动波特率不运行）。	
		1	自动波特率起动（自动波特率运行）。自动波特率运行位。自动波特率完成后此位自动清除。	
1	MODE		自动波特率模式选择位。	0
		0	模式 0。	
		1	模式 1。	
7:2	-	-	保留	0
8	ABEOINTCLR		自动波特率中断结束清除位（仅可写访问）	0
		0	写入 0 没有影响。	
		1	写入 1 将清除 UIIR 中的相应中断。	
9	ABTOINTCLR		自动波特率超时中断清除位（仅可写访问）	0
		0	写入 0 没有影响。	
		1	写入 1 将清除 UIIR 中的相应中断。	
31:10	-	-	保留	0

### 10.3.12 自动波特率

UART 自动波特率功能可用于基于AT协议（Hayes 命令）测量输入的波特率。使能后，自动波特率功能将测量接收数据流的位时间，并相应设置除数锁存寄存器UDL。



自动波特率通过设置UACR起始位来起动。可通过清除UACR起始位来停止自动波特率。自动波特率完成后起始位将清除，且读取位的操作将返回自动波特率的状态（挂起/完成）。

有两种自动波特率测量模式可用，可通过UACR模式位选择。在模式0中，在UART Rx引脚的两个后续下降沿上测量波特率（起始位的下降沿和最低有效位的下降沿）。在模式1中，在UART Rx引脚的下降沿和后续上升沿之间测量波特率（起始位的长度）。

出现超时，UACR AutoRestart位可用于自动重启波特率测量（速率测量计数器溢出）。如果设置了此位，则将在UART Rx引脚的下一下降沿重启速率测量。

自动波特率函数可产生两种中断。

- UIIR ABTOInt中断将在中断使能时设置（UIER ABTOIntEn设置，并且自动波特率测量计数器溢出）。
- UIIR ABEOInt中断将在中断使能时设置（UIER ABEOIntEn设置，并且自动波特率成功完成）。

必须通过设置相应的UACR ABTOIntCir 和ABEOIntEn 位来清除自动波特率中断。

小数波特率发生器在自动波特率期间必须禁用(DIVADDVAL = 0)。

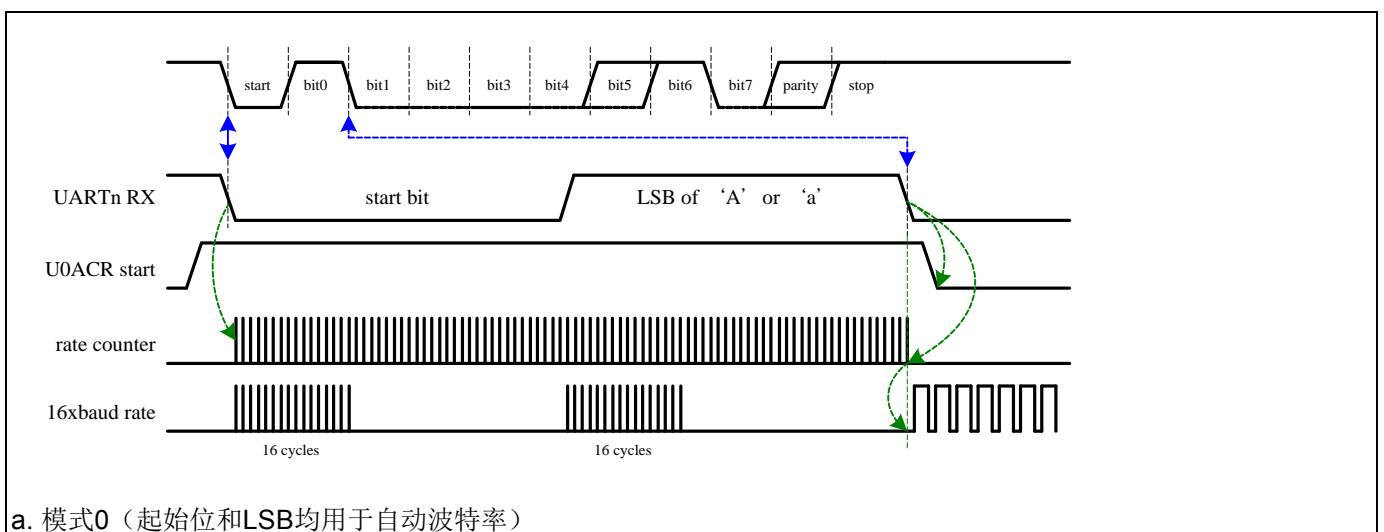
同样，使用自动波特率时，对UDL寄存器的任何写操作都必须在UACR寄存器写操作前完成。UART支持的最小和最大波特率是UART\_PCLK、数据位、停止位和奇偶校验位的数量的函数。

$$ratemin = \frac{2 \times PCLK}{2^{15}} \leq UART_{baudrate} \leq \frac{PCLK}{(2 + databits + paritybit + stopbit)} = ratemax$$

### 10.3.13 自动波特率模式

软件要使用AT命令时，可用预期的字符格式配置UART并设置UACR起始位。无需考虑除数锁存器UDL中的初始值。由于A或者a的ASCII编码（“A” = 0x41，“a” = 0x61），UART Rx引脚检测到起始位，且预期字符的最低有效位由两个下降沿分隔。UACR起始位设置后，自动波特率协议将执行以下几个阶段：

1. UACR起始位设置时，波特率测量计数器被重置，UART RSR也重置。RSR波特率切换到最高速率。
2. UART Rx引脚的下降沿触发起始位的开始。速率测量计数器将开始对UART\_PCLK循环进行计数。
3. 接收起始位期间，RSR波特率输入中产生16个脉冲，频率为UART输入时钟的频率，保证起始位存储在RSR中。
4. 在接收起始位的过程中（以及模式= 0下的字符LSB），速率计数器将随着被预分频的UART输入时钟(UART\_PCLK)递增。
5. 如果是模式= 0，那么速率计数器将在UART Rx引脚的下一个下降沿停止。如果是模式= 1，那么速率计数器将在UART Rx引脚的下一个上升沿停止。
6. 速率计数器被加载到UDL中，并且波特率将切换到正常操作。设置UDL后，自动波特率结束中断UIIR ABEOInt将被设置（如果使能）。RSR现在将继续接收字符的剩余位。



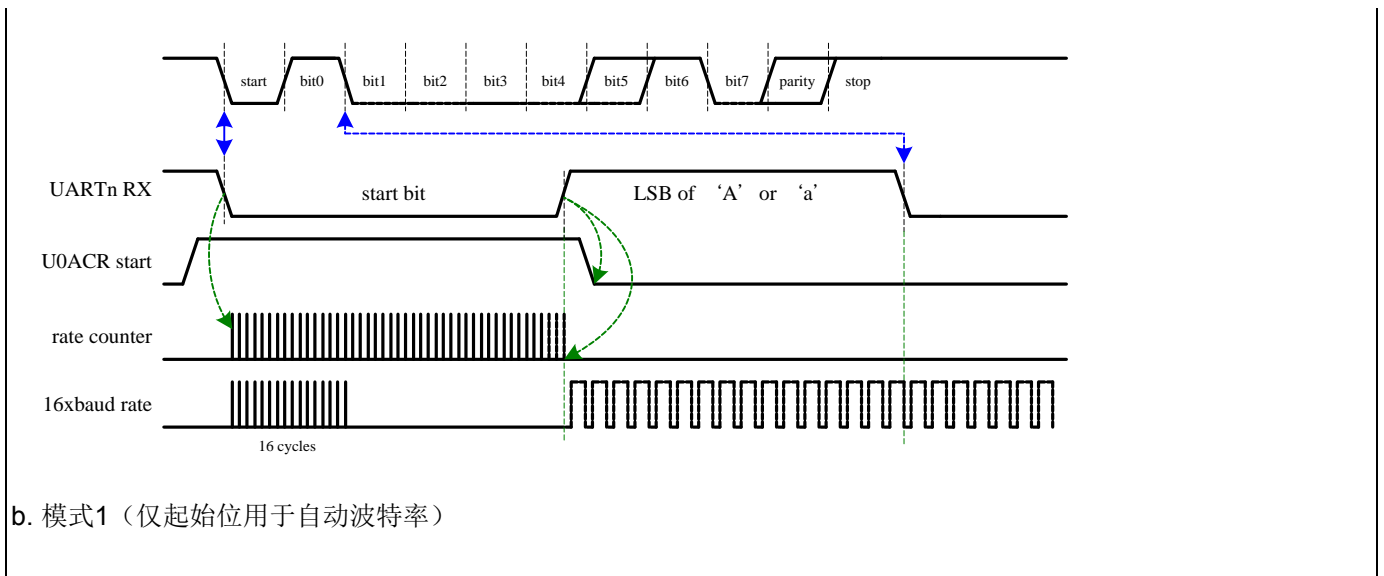


图 10.3.13.1 自动波特率 a) 模式0 和 b) 模式1 的波形图

### 10.3.14 波特率计算

操作UART时可使用也可不使用小数分频器。在实际应用中，使用几个不同的小数分频器设置就可能获得所需的波特率。以下算法展示了一种查找一组UDL值的方法。这组参数产生的波特率相对于所需波特率的误差小于1.1%。

#### 10.3.14.1 示例 1 : UART\_PCLK = 14.7456 MHz, BR = 9600

根据所提供的算法  $UDL = PCLK / BR = 14.7456 \text{ MHz} / 9600 = 1536$ .

#### 10.3.14.2 示例 2 : UART\_PCLK = 12 MHz, BR = 115200

根据所提供的算法  $UDL = PCLK / BR = 12 \text{ MHz} / 115200 = 104$ .

### 10.3.15 UART RS485 控制寄存器(URS485CTRL – 0x5002 804C)

RS485CTRL 寄存器控制 RS-485/EIA-485 模式下 UART 的配置。

表 10.3.15.1 UART RS485控制寄存器(URS485CTRL - 0x5002 804C (UART0)) 位描述

位	符号	值	描述	重置值
0	NMMEN		NMM 使能。	0
		0	RS-485/EIA-485 正常多点模式(NMM) 禁用。	
		1	RS-485/EIA-485 正常多点模式(NMM) 使能。在该模式下，当收到的字节导致 USART 设置奇偶校验错误并生成中断时，检测到地址。	
1	RXDIS		接收器使能。	0
		0	接收器使能。	
		1	接收器禁用。	
2	AADEN		AAD 使能。	0
		0	自动地址检测(AAD) 禁用。	
		1	自动地址检测(AAD) 使能。	
31:3	-	-	保留	0

### 10.3.16 UART RS485 地址匹配寄存器(URS485ADRMATCH – 0x5002 8050)

RS485ADRMATCH 寄存器包含 RS-485/EIA-485 模式的地址匹配值。

表 10.3.16.1 UART RS485地址匹配寄存器 (U0RS485ADRMATCH –0x5002 8050 (UART0)) 位描述

位	符号	值	描述
7:0	ADRMATCH	包含地址匹配值	0x00

### 10.3.17 RS-485/EIA-485 模式的操作

RS-485/EIA-485 功能可将USART 配置为可寻址从机。

可寻址从机是由单个主机控制的多个从机之一。

UART主机发送器将设置奇偶位（第9位）为1以标识一个地址字符。对于数据字符，将奇偶校验位设置为0。

可为每个 UART 从机接收器分配一个唯一的地址。可以对从机进行编程，使其手动或自动拒绝不是其自身地址后的数据。

#### RS-485/EIA-485 正常多点模式 (NMM)

将RS485CTRL位设置为0可使能此模式。在该模式下，当收到的字节导致UART设置奇偶校验错误并生成中断时，检测到地址。

如果接收器被禁用（RS485CTRL位1=1），任何接收到的数据字节都将被忽略，且不会存入RXFIFO。检测到地址字节（奇偶校验位=1）时，会将其放入RXFIFO，并生成一个“Rx 数据就绪中断”。处理器然后即可读取地址字节，并决定是否使能接收器来接收随后的数据。

当接收器被使能（RS485CTRL 位 1=0），所有接收到的字节将会被接受并存入 RXFIFO，不论是数据还是地址。收到地址字符时，将生成一个奇偶校验错误中断，然后处理器会决定是否禁用接收器。

#### RS-485/EIA-485 自动地址检测(AAD) 模式

RS485CTRL寄存器位0（9 位模式使能）和2（AAD模式使能）都设置时，UART 处于自动地址检测模式。

在该模式下，接收器会将任何收到的地址字节（奇偶校验= 1）与编程到RS485ADRMATCH寄存器中的8 位值进行比较。

如果接收器被禁用（RS485CTRL位1=1），任何接收到的与RS485ADRMATCH值不匹配的数据字节或者地址字节都将被丢弃。

检测到匹配的地址字符时，会将其连同奇偶校验位一起推入RXFIFO，然后接收器将自动使能（RS485CTRL位1 将由硬件清除）。接收器也将生成一个“Rx数据就绪中断”。

接收器使能时（RS485CTRL 位 1 = 0），将接受收到的所有字节并存储在 RXFIFO 中，直至收到与RS485ADRMATCH 值不匹配的地址字节。出现这种情况时，接收器将在硬件中自动禁用（RS485CTRL 位 1 将被设置），收到的不匹配地址字符将不会存储在 RXFIFO 中。

#### RS-485/EIA-485 自动方向控制

RS485/EIA-485模式包含允许发送器以方向控制输出信号的方式自动控制DIR引脚状态的选项。

设置RS485CTRL位4=1将使能该功能。

将RS485CTRL位3保持为0，这样，如果已使能方向控制，就会使用RTS引脚。

使能“自动方向控制”后，CPU 将数据写入TXFIFO时，选定的引脚将被断言（驱动低电平）。数据的最后一位发送后，引脚将被取消（驱动高电平）。请参见RS485CTRL寄存器中的位4和5。

除了环回模式之外，RS485CTRL 位 4 优于所有其他控制方向引脚的机制。

#### RS-485/EIA-485 驱动程序延迟时间

驱动程序延迟时间是指最后一个停止位离开TXFIFO到RTS失效这一段时间。此延迟时间可在8位RS485DLY寄存器中编程。

延迟时间以波特率时钟为周期。可以使用 0 到 255 位倍数的任何延迟时间。

#### RS-485/EIA-485 输出反转

引脚 RTS（或 DTR）方向控制信号的极性可通过对寄存器 RS485CTRL 第 5 位编程来反转。此位设置后，发送器有要发送的数据时，方向控制引脚将驱动到逻辑 1。数据的最后一位发送后，方向控制引脚将被驱动到逻辑 0。

## 10.4 架构

UART的架构如以下框图所示。

APB接口提供了CPU或主机与UART之间的通信链接。

UART接收器模块RX监控串行输入线路，RXD用于有效输入。UART RX移位寄存器(RSR) 通过RXD接受有效字符。

有效字符汇编到RSR中后，被传递到UART RX缓冲寄存器FIFO，等待CPU 或主机通过通用主机接口进行访问。

UART发送器模块TX接受CPU或主机写入的数据，并将数据缓冲在UART TX保持寄存器FIFO(THR)中。UART TX移位寄存器(TSR)读取THR中存储的数据，并汇编数据以通过串行输出引脚TXD发送。

UART波特率发生器模块BRG生成由UART TX模块使用的定时。UBRG时钟输入源是UART\_PCLK。主时钟被UDL寄

寄存器中指定的除数分频。此分频后的时钟是16倍过采样时钟，NBAUDOUT。  
 中断接口接收从UTX和URX块使能的多个一时钟宽度。  
 UTX和URX的状态信息存储在ULSR中。UTX和URX的控制信息存储在ULCR中。

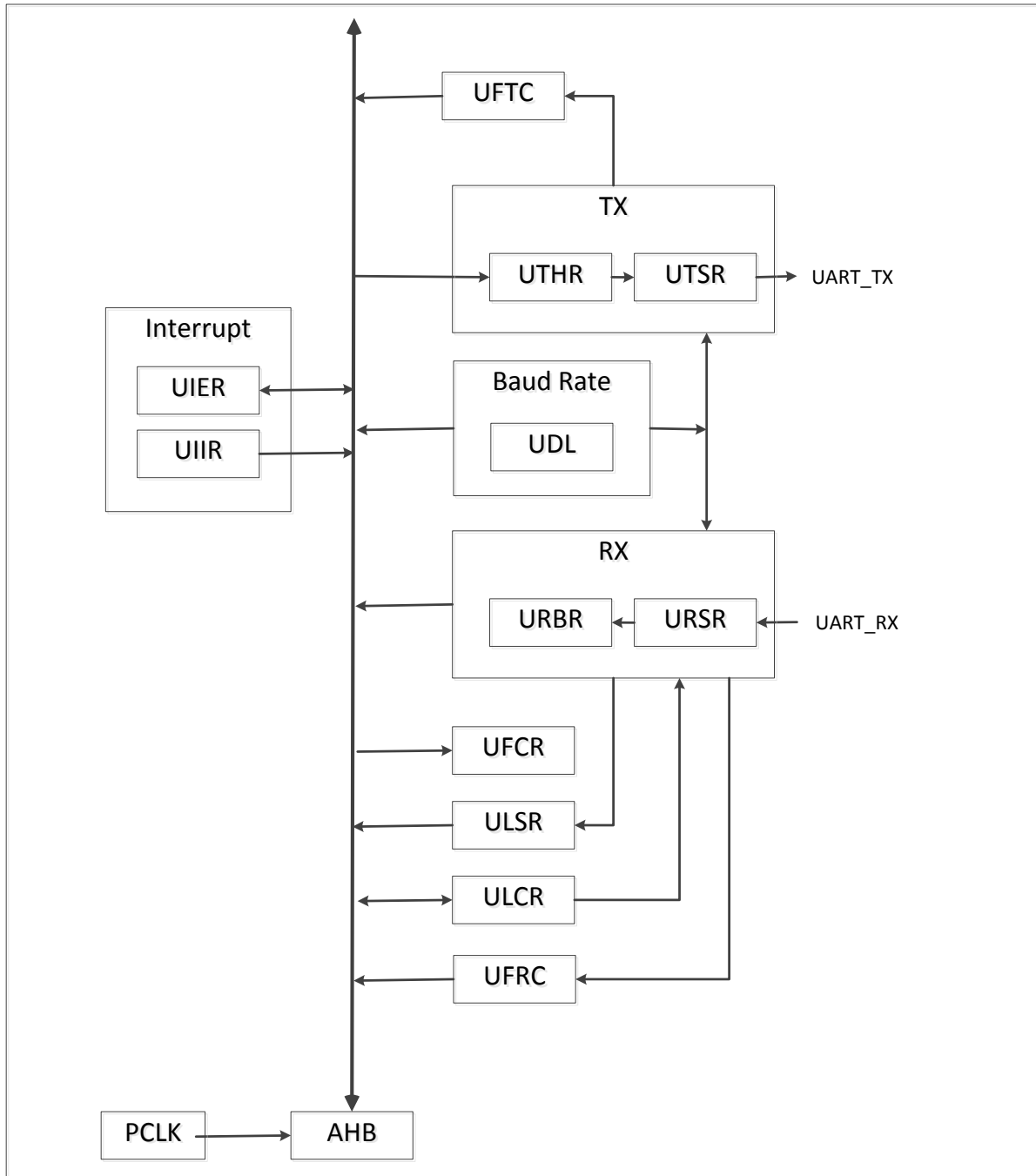


图 10.4.1 UART 框图

## 11. 16位计数器/定时器

### 11.1 特性

- 两个带有可编程16位预分频器的16位计数器/定时器。
- 计数器/定时器操作
- 可配置定时器和预分频器在指定捕获事件清零。此特性通过在输入脉冲前沿清零定时器并捕获定时器在后沿的值，方便进行脉冲宽度测量。
- 两个16位捕获通道，可在输入信号跳变时快速捕获定时器值。捕获事件还可以有选择性地生成中断。
- 四个16位匹配寄存器允许：
  - 连续操作，可选择在匹配时产生中断。
  - 在与可选中断生成相匹配时停止定时器运行。
  - 在与可选中断生成相匹配时进行定时器重置。

### 11.2 应用

- 时间间隔定时器，用于对内部事件进行计数
- 脉冲宽度解调器（经捕获输入）
- 自由运行的定时器

### 11.3 描述

每个计数器/定时器都设计用来计算外设时钟(PCLK)或外部供电时钟的周期，并且可以有选择性地生成中断，或基于四个匹配寄存器以指定定时器值来执行其他操作。每个计数器/定时器还包括1个捕获输入，用来在输入信号跳变时捕获定时器值，同时可根据需要产生一个中断。

注：16位计数器/定时器0(T16B0)和16位计数器/定时器1(T16B1)除外设基址不同外，其他功能相似。

### 11.4 引脚描述

表 11.4.1 简要总结了每个计数器/定时器相关的引脚。

表 11.4.1 计数器/定时器引脚描述

引脚	类型	描述
T16B0_CAP[2:0] T16B1_CAP0[2:0]	输入	捕获信号： 可以配置捕获引脚上的跳变，用计数器/定时器中的值载入捕获寄存器，并且可以有选择性地产生中断。 计数器/定时器模块可选择捕获信号作为时钟源来代替PCLK衍生时钟。

### 11.5 寄存器描述

16位计数器/定时器0包含的寄存器如表 11.5.1 所示，16位计数器/定时器1包含的寄存器如表 279 所示。详细描述如下。

16位计数器/定时器0 T16B0 (基址 0x4000 C000)

16位计数器/定时器1 T16B1 (基址 0x4001 0000)

表 11.5.1 寄存器简介

名称	访问类型	地址偏移	描述	重置值
TMR16IR	R/W	0x000	中断寄存器。可以对IR执行写入操作来清除中断。可以对IR执行读取操作，以确定可能使用哪个中断源（共8个）。	0
TMR16TCR	R/W	0x004	定时器控制寄存器。TCR用于控制定时器计数器功能。通过TCR可以对定时器计数器执行禁用或重置操作。	0
TMR16TC	R/W	0x008	16位TC 每隔PR + 1个PCLK周期递增一次。TC将通过TCR来控制。	0

TMR16PR	R/W	0x00C	预分频寄存器(PR)。当预分频计数器(见下)与该值相等时,下一时钟递增0 TC并清除PC。	
TMR16PC	R/W	0x010	预分频计数器。16位PC是一个计数器,它会增加到与PR中存放的值相等。0当计数到达PR中的值时,TC将递增计数,并且会清除PC值。通过总线接口可以观察和控制PC。	
TMR16MCR	R/W	0x014	匹配控制寄存器。MCR用于控制在发生匹配时是否生成中断,以及是否进行0 TC重置。	
TMR16MR0	R/W	0x018	匹配寄存器0。通过MCR可以使能MR0,以便在每次MR0与TC相匹配时执行0 TC重置、停止TC和PC运行和/或生成中断。	
TMR16MR1	R/W	0x01C	匹配寄存器1。请参见MR0描述。	0
TMR16MR2	R/W	0x020	匹配寄存器2。请参见MR0描述。	0
TMR16MR3	R/W	0x024	匹配寄存器3。请参见MR0描述。	0
TMR16CCR	R/W	0x028	捕获控制寄存器。CCR用于控制将使用哪些捕获输入边缘来加载捕获寄存器,并且在出现捕获时是否生成中断。	
TMR16CR0	RO	0x02C	捕获寄存器0(CR0)。当T16Bn_CAP0输入上产生事件时,CR0载入TC值。	0
TMR16CR1	RO	0x030	捕获寄存器1(CR1)。当T16Bn_CAP1输入上产生事件时,CR1载入TC值。	0
TMR16CR2	RO	0x034	捕获寄存器2(CR2)。当T16Bn_CAP2输入上产生事件时,CR2载入TC值。	0
-	-	0x038 – 0x6C	保留	0
TMR16CDF0	R/W	0x50	CAP0 数字滤波器寄存器	0
TMR16CDF1	R/W	0x54	CAP1 数字滤波器寄存器	0
TMR16CDF2	R/W	0x58	CAP2 数字滤波器寄存器	0
TMR16CTCR	R/W	0x070	计数控制寄存器。CTCR用于选择定时器模式和计数器模式。并且在计数器模式中,会选择要进行计数的信号和边缘。	

### 11.5.1 中断寄存器(TMR16IR – 0x4000 C000, 0x4001 0000)

中断寄存器包含 4 个用于匹配中断的位及 3 个用于捕获中断的位。如果有中断产生, IR 中的相应位为高电平。否则,该位为低电平。将逻辑 1 写入到相应的 IR 位中,将执行中断重置。写入 0 无效。

表 11.5.1.1 中断寄存器(TMR16IR - 0x4000 C000 (T16B0), 0x4001 0000 (T16B1)) 位描述

位	符号	描述	重置值
0	MR0INT	匹配通道 0 的中断标志。	0
1	MR1INT	匹配通道 1 的中断标志。	0
2	MR2INT	匹配通道 2 的中断标志。	0
3	MR3INT	匹配通道 3 的中断标志。	0
4	CR0INT	捕获通道 0 事件的中断标志。	0
5	CR1INT	捕获通道 1 事件的中断标志。	0
6	CR2INT	捕获通道 2 事件的中断标志。	
31:7	-	保留	-

### 11.5.2 定时器控制寄存器 (TMR16TCR – 0x4000 C004, 0x4001 0004)

定时器控制寄存器(TCR)用于控制计数器/定时器的操作。

表 11.5.2.1 定时器控制寄存器 (TMR16TCR - 0x4000 C004 (T16B0), 0x4001 0004 (T16B1)) 位描述

位	符号	描述	重置值
0	CEN	计数器使能。当为 1 时,定时器计数器和预分频计数器使能计数。当为 0 时,计数器被禁用。	0
1	CRST	计数器重置。当为 1 时,定时器计数器和预分频计数器在 PCLK 的下一个上升沿同步重置。计数器保持重置状态,直到 TCR[1]返回到 0。	0
2	CS	定时器源选择 0: 系统时钟	0



	1: 20K(不支持省电模式)	
31:3	- 保留	0

### 11.5.3 定时器计数器寄存器(TMR16TC – 0x4000 C008, 0x4001 0008)

当预分频器计数器达到其终结计数时，16 位定时器计数器会递增计数。如果 TC 在到达计数器上限之前没有重置，它将一直计数到 0x0000 FFFF 然后翻转到 0x0000 0000。该事件不会产生中断，如果需要，可使用匹配寄存器检测溢出。

注意：在 CEN（定时器控制寄存器，TCR 位 0）= 0 条件下写功能。

表 11.5.3.1 定时器计数器寄存器 (TMR16B0TC - 0x4000 C008 (T16B0), 0x4001 0008 (T16B1)) 位描述

位	符号	描述	重置值
15:0	TC	定时器计数器值	0
31:16	-	保留	-

### 11.5.4 预分频寄存器(TMR16PR – 0x4000 C00C, 0x4001 000C)

16 位预分频寄存器指定了预分频计数器的最大计数值。

表 11.5.4.1 预分频寄存器 (TMR16PR - 0x4000 C00C (T16B0), 0x4001 000C (T16B1)) 位描述

位	符号	描述	重置值
15:0	PR	预分频最大值。	0
31:16	-	保留	-

### 11.5.5 预分频计数器寄存器(TMR16PC – 0x4000 C010, 0x4001 0010)

16 位预分频计数器将在 PCLK 应用于定时器计数器之前，使用某一常数对 PCLK 执行分频控制。它所控制的是定时器分辨率与最大时间之间的关系，从而能防止定时器溢流。预分频计数器会在每个 PCLK 时钟上递增计数。当预分频计数器的计数达到预分频寄存器中存储的值时，定时器计数器将递增计数，并且在下一个 PCLK 时钟上进行预分频计数器重置。这将使得 TC 当 PR = 0 时在每个 PCLK 上递增计数，当 PR = 1 时，在每 2 个 PCLK 上递增计数，...

注意：在 CEN（定时器控制寄存器，TCR 位 0）= 0 条件下写功能。

表 11.5.5.1 预分频计数器寄存器(TMR16PC - 0x4001 C010 (T16B0), 0x4000 0010 (T16B1)) 位描述

位	符号	描述	重置值
15:0	PC	预分频计数器值	0
31:16	-	保留	-

### 11.5.6 匹配控制寄存器(TMR16MCR – 0x4000 C014, 0x4001 0014)

匹配控制寄存器用于控制在某个匹配寄存器与定时器计数器相匹配时将执行的操作。匹配控制寄存器各位的功能如表 11.5.6.1 所示。

表 11.5.6.1 匹配控制寄存器(TMR16MCR - 0x4000 C014 (T16B0), 0x4001 0014 (T16B1)) 位描述

位	符号	值	描述	重置值
0	MR0I		MR0 上的中断：当 MR0 与 TC 值匹配时产生中断。	0
		1	使能	
		0	禁用	
1	MR0R		MR0 上的重置：MR0 与 TC 匹配将使 TC 重置。	0
		1	使能	
		0	禁用	
2	MR0S		MR0 上的停止：MR0 与 TC 匹配时将使 TC 和 PC 停止，TCR[0]置 0。	0
		1	使能	
		0	禁用	
3	MR1I		MR1 上的中断：当 MR1 与 TC 值匹配时产生中断。	0
		1	使能	

	0	禁用	
4	MR1R	MR1 上的重置: MR1 与 TC 匹配将使 TC 重置。	0
		1	
		0	禁用
		1	使能
5	MR1S	MR1 上的停止: MR1 与 TC 匹配时将使 TC 和 PC 停止, TCR[0]置 0。	0
		1	
		0	禁用
		1	使能
6	MR2I	MR2 上的中断: 当 MR2 与 TC 值匹配时产生中断。	0
		1	
		0	禁用
		1	使能
7	MR2R	MR2 上的重置: MR2 与 TC 匹配将使 TC 重置。	0
		1	
		0	禁用
		1	使能
8	MR2S	MR2 上的停止: MR2 与 TC 匹配时将使 TC 和 PC 停止, TCR[0]置 0。	0
		1	
		0	禁用
		1	使能
9	MR3I	MR3 上的中断: 当 MR3 与 TC 值匹配时产生中断。	0
		1	
		0	禁用
		1	使能
10	MR3R	MR3 上的重置: MR3 与 TC 匹配将使 TC 重置。	0
		1	
		0	禁用
		1	使能
11	MR3S	MR3 上的停止: MR3 与 TC 匹配时将使 TC 和 PC 停止, TCR[0]置 0。	0
		1	
		0	禁用
		1	使能
31:12 -	-	保留	0

#### 11.5.7 匹配寄存器(TMR16MR0/1/2/3 – 0x4000 C018/1C/20/24, 0x4001 0018/1C/20/24)

匹配寄存器值将不断与定时器计数器值进行比较。直至两个值相等时, 会自动启动各种操作。可能产生的操作有生成中断、定时器计数器重置或停止定时器运行。这些操作将由MCR寄存器中的设置来控制。

表 11.5.7.1 匹配寄存器 (TMR16MR0/1/2/3 - 0x4000 C018/1C/20/24 (T16B0), 0x4001 0018/1C/20/24 (T16B1)) 位描述

位	符号	描述	重置值
15:0	MATCH	定时器计数器匹配值	0
31:16	-	保留	-

#### 11.5.8 捕获控制寄存器(TMR16CCR – 0x4000 C028, 0x4001 0028)

捕获控制寄存器用于控制当捕获事件发生时, 是否将计数器/ 定时器中的值装入捕获寄存器, 以及捕获事件是否产生中断。同时设置上升位和下降位是一种有效的配置, 这会在两个边缘产生捕获事件。在下面描述中, n表示定时器编号, 0或1。

表 11.5.8.1 捕获控制寄存器(TMR16CCR - 0x4000 C028 (T16B0), 0x4001 0028 (T16B1)) 位描述

位	符号	值	描述	重置值
0	CAP0RE		T16Bn_CAP0 上升沿捕获: T16Bn_CAP0 上的从 0 至 1 的序列, 将使 CR0 载入 TC 内 0 容。	0
		1	使能	
		0	禁用	
1	CAP0FE		T16Bn_CAP0 下降沿捕获: T16Bn_CAP0 上的从 1 至 0 的序列, 将使 CR0 载入 TC 内 0 容。	0
		1	使能	
		0	禁用	
2	CAP0I		T16Bn_CAP0 事件中断: T16Bn_CAP0 事件所导致的 CR0 加载将产生一个中断。	0
		1	使能	



	0	禁用	
3	CAP1RE	T16Bn_CAP1 上升沿捕获: T16Bn_CAP1 的从 0 至 1 的序列, 将使 CR1 载入 TC 内 0 容。	0
	1	使能	
	0	禁用	
4	CAP1FE	T16Bn_CAP1 上升沿捕获: T16Bn_CAP1 的从 0 至 1 的序列, 将使 CR1 载入 TC 内 0 容。	0
	1	使能	
	0	禁用	
5	CAP1I	T16Bn_CAP1 事件中断: T16Bn_CAP1 事件所导致的 CR1 加载将产生一个中断。	0
	1	使能	
	0	禁用	
6	CAP2RE	T16Bn_CAP2 上升沿捕获: T16Bn_CAP2 的从 0 至 1 的序列, 将使 CR2 载入 TC 内 0 容。	0
	1	使能	
	0	禁用	
7	CAP2FE	T16Bn_CAP2 上升沿捕获: T16Bn_CAP2 的从 0 至 1 的序列, 将使 CR2 载入 TC 内 0 容。	0
	1	使能	
	0	禁用	
8	CAP2I	T16Bn_CAP2 事件中断: T16Bn_CAP2 事件所导致的 CR2 加载将产生一个中断。	0
	1	使能	
	0	禁用	
27:9	-	保留	
28		选择 T16Bn_CAP0 源	0
	0	请参阅 GPIO 设置(MF0/MF1)	
	1	ACMP0 输出	
29		选择 T16Bn_CAP1 源	0
	0	请参阅 GPIO 设置(MF0/MF1)	
	1	ACMP1 输出	
30		选择 T16Bn_CAP2 源	0
	0	请参阅 GPIO 设置( MF0/MF1)	
	1	ACMP2 输出	
31		保留	

### 11.5.9 捕获寄存器(TMR16CR0/1/2 – 0x4000 C02C/30/34, 0x4001 002C/30/34)

各捕获寄存器与器件引脚相关联, 当引脚发生特定的事件时, 可将计数器/ 定时器的值装入该捕获寄存器。捕获控制寄存器中的设置用于确定是否使能捕获功能, 以及是在相关引脚的上升沿, 还是下降沿或两者之上发生捕获事件。

表 11.5.9.1 捕获寄存器(TMR16CR0/1/2 - 0x4000 C02C/30/34 (T16B0), 0x4001 002C/30/34(T16B1)) 位描述

位	符号	描述	重置值
15:0	CAP	定时器计数器捕获值	0
31:16	-	保留	-

### 11.5.10 CAP0/1/2 数字滤波器寄存器(TMR16CDF0/1/2 – 0x4000 C050/54/58, 0x4001 0050/54/58)

Table 11.5.12.1 CAP0/1/2 数字滤波器寄存器(TMR16CDF0/1/2 - 0x4000 C050/54/58 (T16B0), 0x4001 0050/54/58 (T16B1)) 位描述

位	符号	描述	重置值	位
3:0	CDF	该寄存器过滤了 cap0/1/2 引脚上的噪声		0
		0	禁用	
		1	滤波器宽度 = 系统时钟 / 1	
		2	滤波器宽度 = 系统时钟 / 2	
		3	滤波器宽度 = 系统时钟 / 4	

4	滤波器宽度 = 系统时钟 / 8
5	滤波器宽度 = 系统时钟 / 16
6	滤波器宽度 = 系统时钟 / 32
7	滤波器宽度 = 系统时钟 / 64
8	滤波器宽度 = 系统时钟 / 128
9	滤波器宽度 = 系统时钟 / 256
10	滤波器宽度 = 系统时钟 / 512
11	滤波器宽度 = 系统时钟 / 1024
12	滤波器宽度 = 系统时钟 / 2048
31:4	保留

### 11.5.11 计数控制寄存器(TMR16CTCR – 0x4000 C070, 0x4001 0070)

计数控制寄存器(CTCR)用于选择定时器模式或者计数器模式，并在计数器模式中选择要计数的引脚和边沿。

如果将计数器模式选为操作模式，则在PCLK 时钟的每个上升沿上会对（CTCR位3:2所选的）CAP输入进行采样。在比较该CAP输入的两个连续样本后，将会确认下列四个事件之一：所选CAP输入电平处于上升沿、下降沿，或上升下降沿或无变化。仅当所标识的事件与CTCR寄存器中的位1:0所选的内容相匹配时，定时器计数器寄存器才会递增计数。

计数器的外部提供时钟的有效处理能力具有一定的局限性。

由于PCLK 时钟的两个连续上升沿仅用于标识CAP 所选输入的一个边缘，因此CAP输入的频率不能超过PCLK时钟的一半。因此在这种情况下，相同的CAP 输入上的高电平/低电平的持续时间不能少于 $1/(2 \times \text{PCLK})$ 。

该寄存器的位7:4还用于使能和配置捕获-清除-定时器特性。

该特性允许特定CAP输入的指定边沿将定时器全部清零。使用该机制在输入脉冲前沿清除定时器然后在后沿执行捕获，可以使用单捕获输入来直接测量脉冲宽度，而无需在软件中执行减法操作。

表 11.5.11.1 计数控制寄存器(TMR16CTCR - 0x4000 C070 (T16B0), 0x4001 0070(T16B1)) 位描述

位	符号	值	描述	重置值
1:0	CTM		计数器/定时器模式。该字段用于选择可使用哪个 PCLK 上升沿，使定时器预分频计数器 (PC) 递增计数，或清除 PC 并使定时器计数器 (TC) 递增计数。	00
		0x0	定时器模式：每个 PCLK 的上升沿	
		0x1	计数器模式：将在位 3:2 所选的 CAP 输入的上升沿上执行 TC 递增计数。	
		0x2	计数器模式：将在位 3:2 所选的 CAP 输入的下降沿上执行 TC 递增计数。	
		0x3	计数器模式：将在位 3:2 所选的 CAP 输入的上升沿和下降沿上执行 TC 递增计数。	
3:2	CIS		计数输入选择。在计数器模式下（当该寄存器中位 1:0 不为 00 时），这些位将选择为时钟采样哪个 CAP 引脚或比较器输出。值 0x1 至 0x3 保留。	
		0x0	T16Bn_CAP0	
		0x1	T16Bn_CAP1	
		0x2	T16Bn_CAP2	
4	ENCC		将此位置 1 可在发生位 7:5 指定的捕获-边沿事件时清零定时器和预分频器。	0
7:5	SELCC		当位 4 为 1 时，这些位选择哪个捕获输入边沿将导致定时器和预分频器被清零。当位 4 0 为低电平时，这些位无效。	
		0x0	CAP0 的上升沿清零定时器（如果位 4 被置位）	
		0x1	CAP0 的下降沿清零定时器（如果位 4 被置位）	
		0x2	CAP1 的上升沿清零定时器（如果位 4 被置位）	
		0x3	CAP1 的下降沿清零定时器（如果位 4 被置位）	
		0x4	CAP2 的上升沿清零定时器（如果位 4 被置位）	
		0x5	CAP2 的下降沿清零定时器（如果位 4 被置位）	
31:8	-	-	保留	-

### 11.6 定时器操作示例

如图 11.6.1所示，定时器配置为在匹配时重置计数并产生中断。

将预分频器设置为2，匹配寄存器设置为6。在发生匹配的定时器周期结束时，会进行定时器计数重置。这就为匹配值提供了一个完整周期。当定时器计数到达匹配值后，将在下一个时钟内生成中断以指示出现了匹配。

如图 11.6.2所示，定时器配置为在匹配时停止计数并产生中断。将预分频器再次设置为2，匹配寄存器设置为6。当

定时器计数到达匹配值后，将在下一个时钟内清除TCR中的定时器使能位，并且会生成中断以指示出现了匹配。

如图 11.6.3所示，一个定时器，用于重置计数并在捕获时产生中断。

预分频器设置为2（PR = 2），捕获控制设置为5（CCR = 5），计数控制设置为0x30（CTCR = 0x30）。定时器周期在CAP0的下降沿结束，定时器计数器重置。捕获的数据发生在CAP0的上升沿。

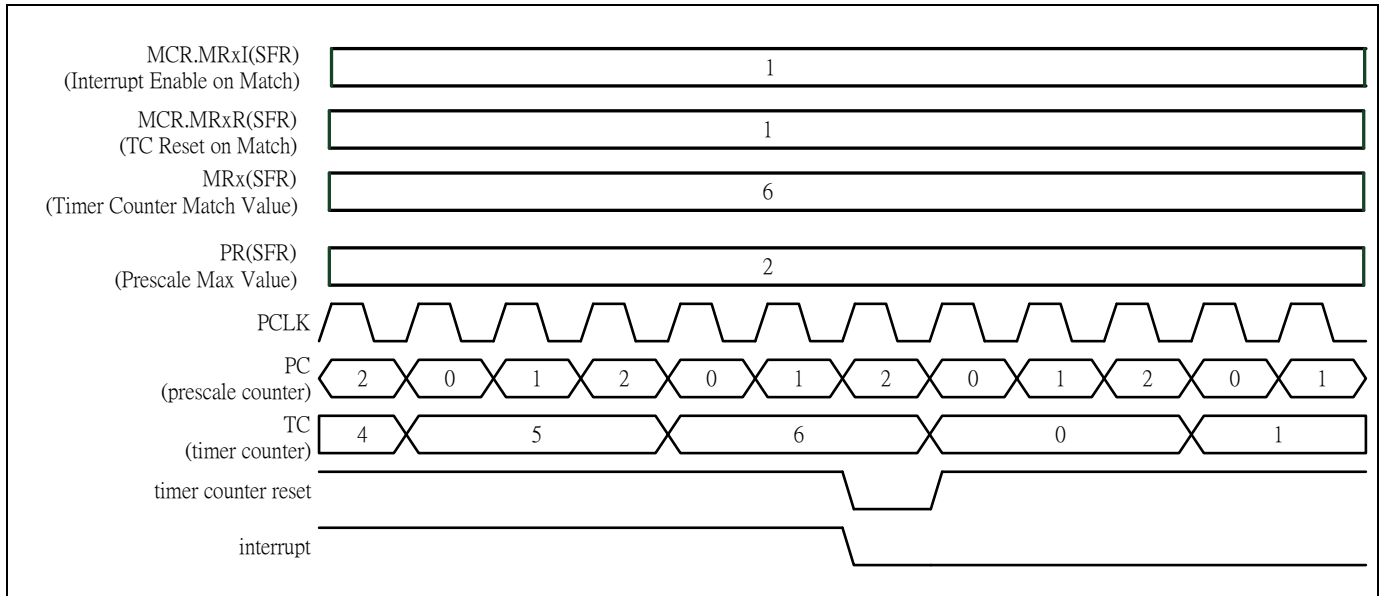


图 11.6.1 定时器周期，其中 PR=2、MRx=6，并且使能了在匹配时同时执行中断和重置操作

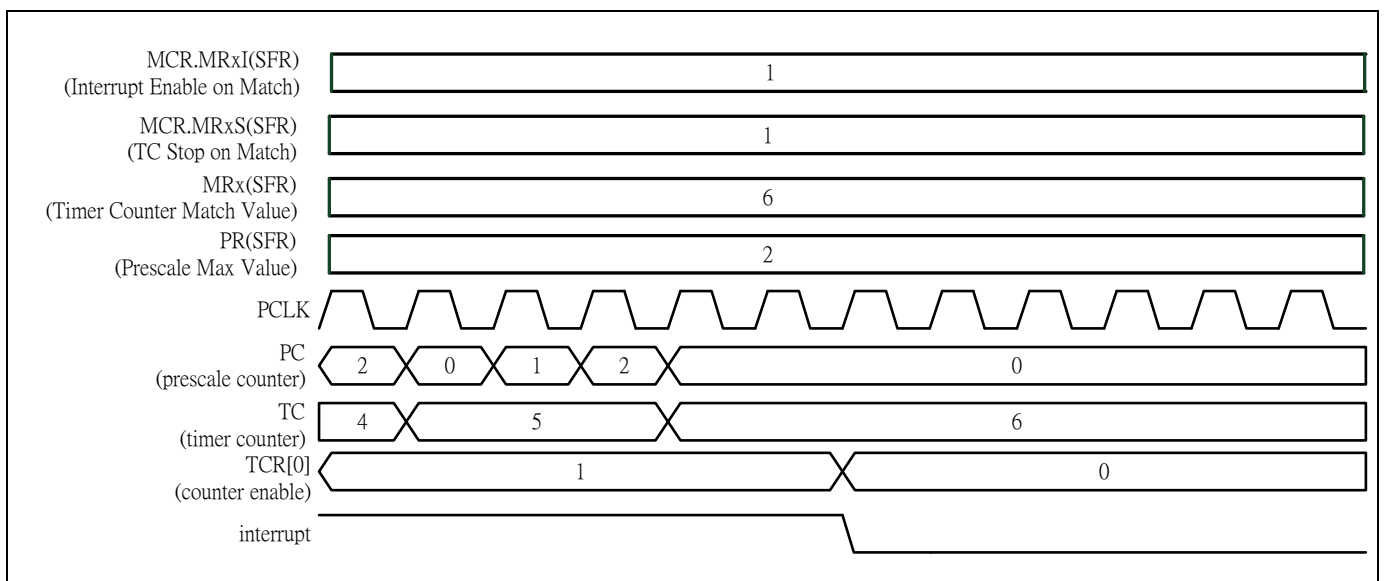


图 11.6.2 定时器周期，其中 PR=2、MRx=6，并且使能了在匹配时同时执行中断和停止操作

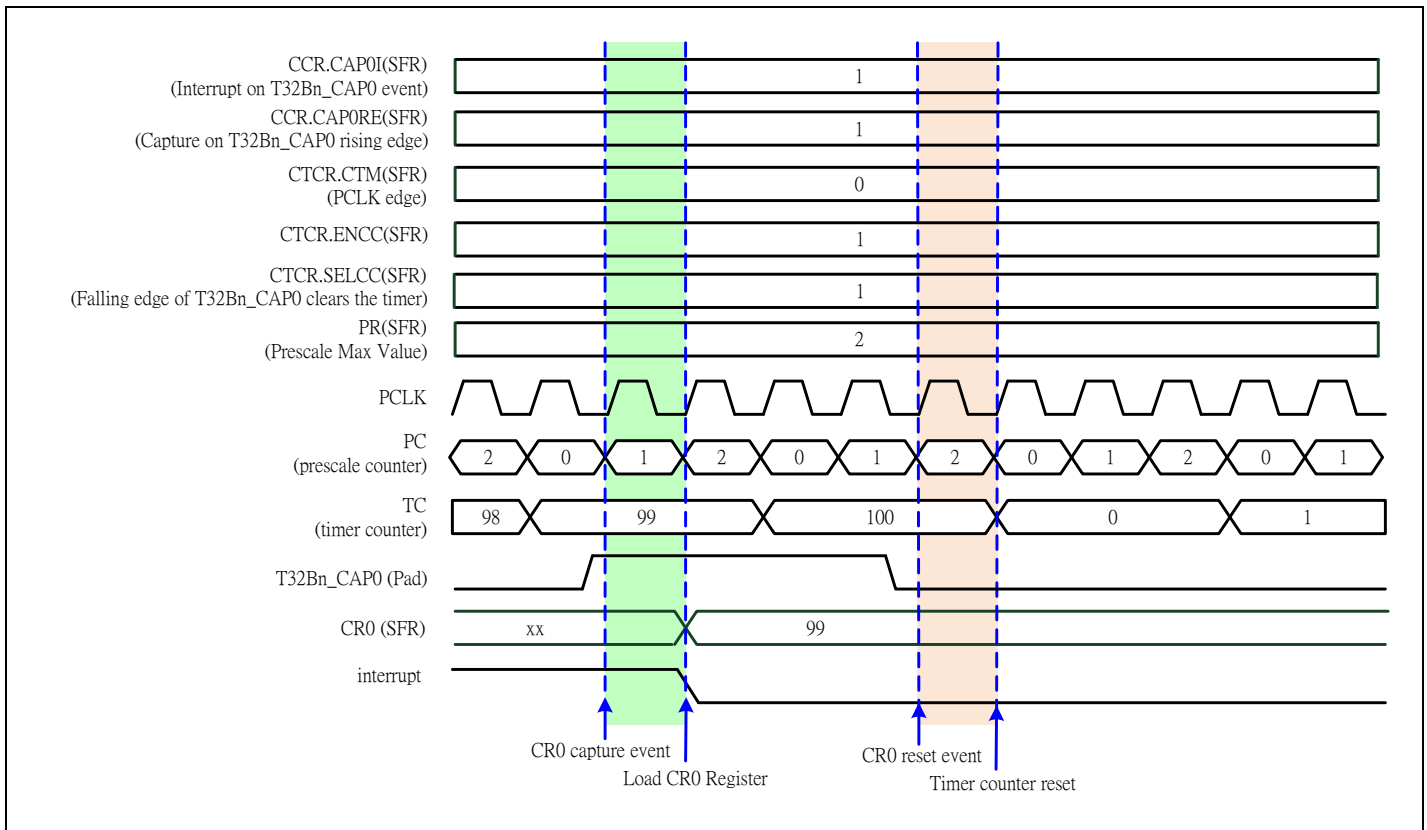


图 11.6.3 一个定时器，用于重置计数并在捕获时产生中断

### 11.7 架构

计数器/定时器 0 和计数器/定时器 1 的功能框图如图 11.7.1 所示。

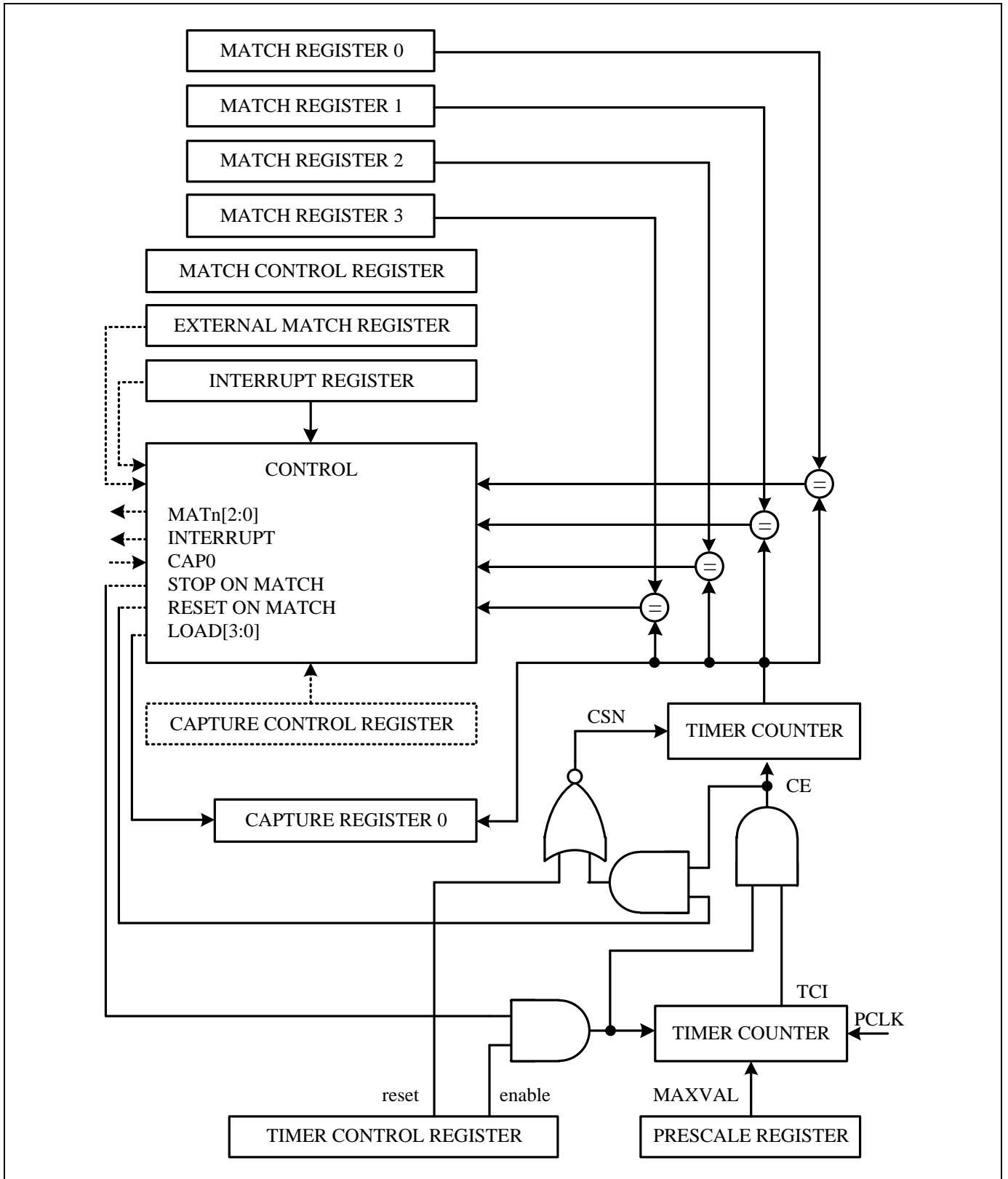


图 11.7.1 16位计数器/定时器功能框图

## 12. 32位计数器/定时器

### 12.1 特性

- 一个带有可编程32位预分频器的32位计数器/定时器。
- 计数器或定时器操作。
- 四个32位匹配寄存器允许：
  - 连续操作，可选择在匹配时产生中断。
  - 在与可选中断生成相匹配时停止定时器运行。
  - 在与可选中断生成相匹配时进行定时器重置。

### 12.2 应用

- 时间间隔定时器，用于对内部事件进行计数
- 自由运行的定时器

### 12.3 简介

每个计数器/定时器都设计用来计算外设时钟(PCLK)或外部供电时钟的周期，并且可以有选择性地生成中断，或基于四个匹配寄存器以指定定时器值来执行其他操作。

## 12.4 寄存器描述

32位计数器/定时器0、1包含的寄存器如表 12.5.1所示。详细描述如下。

表 12.5.1 寄存器简介:

32位计数器/定时器0 T32B0 (基址 0x5002 C000)

名称	访问类型	地址偏移	描述	重置值
TMR32IR	R/W	0x000	中断寄存器。可以对IR 执行写入操作来清除中断。可以对IR执行读取操作，以确定可能使用哪个中断源（共5个）。	0
TMR32TCR	R/W	0x004	定时器控制寄存器。TCR用于控制定时器计数器功能。通过TCR可以对定时器计数器执行禁用或重置操作。	0
TMR32TC	R/W	0x008	定时器计数器。32 位TC每隔PR+1个PCLK周期递增一次。TC将通过TCR来控制。	0
TMR32PR	R/W	0x00C	预分频寄存器(PR)。当预分频计数器（见下）与该值相等时，下一时钟递增TC并清除PC。	0
TMR32PC	R/W	0x010	预分频计数器。32位PC是一个计数器，它会增加到与PR中存放的值相等。当计数到达PR中的值时，TC将递增计数，并且会清除PC值。通过总线接口可以观察和控制PC。	0
TMR32MCR	R/W	0x014	匹配控制寄存器。MCR用于控制在发生匹配时是否生成中断，以及是否进行TC重置。	0
TMR32MR0	R/W	0x018	匹配寄存器0(MR0)。通过MCR可以使能MR0，以便在每次MR0与TC相匹配时执行TC重置、停止TC和PC运行和/或生成中断。	0
TMR32MR1	R/W	0x01C	匹配寄存器1(MR1)。请参见MR0描述。	0
TMR32MR2	R/W	0x020	匹配寄存器2(MR2)。请参见MR0描述。	0
TMR32MR3	R/W	0x024	匹配寄存器3(MR3)。请参见MR0描述。	0
-	-	0x025 – 0x037	保留	0
TMR32TADC0	R/W	0x038	触发ADC0寄存器。	0
-	-	0x040 – 0x06C	保留	0
TMR32CTCR	R/W	0x070	计数控制寄存器。CTCR用于选择定时器模式和计数器模式。并且在计数器模式中，会选择要进行计数的信号和边缘。	0

### 12.4.1 中断寄存器(TMR32IR – 0x5002 C000)

中断寄存器由匹配中断的四个位以及捕获中断的四个位组成。如果有中断产生，IR中的相应位为高电平。否则，该位为低电平。将逻辑1写入到相应的IR位中，将执行中断重置。写入0无效。

表 12.5.1.1 中断寄存器(TMR32IR - 0x5002 C000 (T32B0)) 位描述

位	符号	描述	重置值
0	MR0INT	匹配通道 0 的中断标志。	0
1	MR1INT	匹配通道 1 的中断标志。	0
2	MR2INT	匹配通道 2 的中断标志。	0
3	MR3INT	匹配通道 3 的中断标志。	0
31:4	-	保留	-

### 12.4.2 定时器控制寄存器(TMR32TCR – 0x5002 C004)

定时器控制寄存器(TCR)用于控制计数器/定时器的操作。

表 12.5.2.1 定时器控制寄存器(TMR32TCR - 0x5002 C004 (T32B0)) 位描述

位	符号	描述	重置值
0	CEN	计数器使能。当为 1 时，定时器计数器和预分频计数器使能计数。当为 0 时，计数器被禁用。	0
1	CRST	计数器重置。当为 1 时，定时器计数器和预分频计数器在 PCLK 的下一个上升沿同步重置。计数	0

	器保持重置状态，直到 TCR[1]返回到 0。	
31:2 -	保留	0

#### 12.4.3 定时器计数器寄存器(TMR32TC – 0x5002 C008)

当预分频器计数器达到其终结计数时，32位定时器计数器会递增计数。如果TC 在到达计数器上限之前没有重置，它将一直计数到0xFFFF FFFF然后翻转到0x0000 0000。该事件不会产生中断，如果需要，可使用匹配寄存器检测溢出。

注意：在 CEN（定时器控制寄存器，TCR 位 0）= 0 条件下写功能。

表 12.5.3.1 定时器计数器寄存器(TMR32TC - 0x5002 C008 (T32B0)) 位描述

位	符号	描述	重置值
31:0	TC	定时器计数器值。	0

#### 12.4.4 预分频寄存器(TMR32PR – 0x5002 C00C)

32 位预分频寄存器指定了预分频计数器的最大计数值。

表 11.5.4.1 预分频寄存器(TMR32PR - 0x5002 C00C (T32B0)) 位描述

位	符号	描述	重置值
15:0	PR	预分频最大值。	0
31:16 -		保留	-

#### 12.4.5 预分频计数器寄存器(TMR32PC – 0x5002 C010)

32位预分频计数器将在PCLK应用于定时器计数器之前，使用某一常数值对PCLK执行分频控制。它所控制的是定时器分辨率与最大时间之间的关系，从而能防止定时器溢流。预分频计数器会在每个PCLK时钟上递增计数。当预分频计数器的计数达到预分频寄存器中存储的值时，定时器计数器将递增计数，并且在下一个PCLK时钟上进行预分频计数器重置。

这将使得TC当PR=0时在每个PCLK上递增计数，当PR=1时，在每2个PCLK上递增计数，依次类推。

注意：在 CEN（定时器控制寄存器，TCR 位 0）= 0 条件下写功能。

表 12.5.5.1 预分频寄存器(TMR32PC - 0x5002 C010 (T32B0)) 位描述

位	符号	描述	重置值
31:0	PC	预分频计数器值。	0

#### 12.4.6 匹配控制寄存器(TMR32MCR – 0x5002 C014)

匹配控制寄存器用于控制在某个匹配寄存器与定时器计数器相匹配时将执行的操作。

表 12.5.6.1 匹配控制寄存器(TMR32MCR - 0x5002 C014(T32B0)) 位描述

位	符号	值	描述	重置值
0	MR0I		MR0 上的中断：当 MR0 与 TC 值匹配时产生中断。	0
		1	使能	
		0	禁用	
1	MR0R		MR0 上的重置：MR0 与 TC 匹配将使 TC 重置。	0
		1	使能	
		0	禁用	
2	MR0S		MR0 上的停止：MR0 与 TC 匹配时将使 TC 和 PC 停止，TCR[0]置 0。	0
		1	使能	
		0	禁用	
3	MR1I		MR1 上的中断：当 MR1 与 TC 值匹配时产生中断。	0
		1	使能	
		0	禁用	
4	MR1R		MR1 上的重置：MR1 与 TC 匹配将使 TC 重置。	0
		1	使能	
		0	禁用	



5	MR1S		MR1 上的停止：MR1 与 TC 匹配时将使 TC 和 PC 停止，TCR[0]置 0。	0
		1	使能	
6	MR2I		MR2 上的中断：当 MR2 与 TC 值匹配时产生中断。	0
		1	使能	
7	MR2R		MR2 上的重置：MR2 与 TC 匹配将使 TC 重置。	0
		1	使能	
8	MR2S		MR2 上的停止：MR2 与 TC 匹配时将使 TC 和 PC 停止，TCR[0]置 0。	0
		1	使能	
9	MR3I		MR3 上的中断：当 MR3 与 TC 值匹配时产生中断。	0
		1	使能	
10	MR3R		MR3 上的重置：MR3 与 TC 匹配将使 TC 重置。	0
		1	使能	
11	MR3S		MR3 上的停止：MR3 与 TC 匹配时将使 TC 和 PC 停止，TCR[0]置 0。	0
		1	使能	
31:12	-	-	保留	0

#### 12.4.7 匹配寄存器(TMR32MR0/1/2/3 – 0x5002 C018/1C/20/24)

匹配寄存器值将不断与定时器计数器值进行比较。直至两个值相等时，会自动启动各种操作。可能产生的操作有生成中断、定时器计数器重置或停止定时器运行。这些操作将由MCR寄存器中的设置来控制。

表 12.5.7.1 匹配寄存器(TMR32MR0/1/2/3 - 0x5002 C018/1C/20/24 (T32B0)) 位描述

位	符号	描述	重置值
31:0	MATCH	定时器计数器匹配值	0

#### 12.4.8 Trigger ADC Register (TMR32TADC0 – 0x5002 C038)

Table 12.5.10.1 触发 ADC 寄存器(TMR32TADC0 - 0x5002 C038 (T32B0)) 位描述

位	符号	描述	重置值
30:0	TADC0	满足以下条件时，ADC0 开始转换： <b>TMR32TADC0[30:0] = TMR32TC[30:0]</b>	0
31		TADC0[30:0] 只能在使能此位时使用	
	0	禁用	
	1	使能	

## 12.5 功能描述

### 12.5.1 定时器操作示例

如图 12.6.1.1所示，定时器配置为在匹配时重置计数并产生中断。

将预分频器设置为2，匹配寄存器设置为6。在发生匹配的定时器周期结束时，会进行定时器计数重置。这就为匹配值提供了一个完整周期。当定时器计数到达匹配值后，将在下一个时钟内生成中断以指示出现了匹配。

如图 12.6.1.2所示，定时器配置为在匹配时停止计数并产生中断。将预分频器再次设置为2，匹配寄存器设置为6。当定时器计数到达匹配值后，将在下一个时钟内清除TCR中的定时器使能位，并且会生成中断以指示出现了匹配。

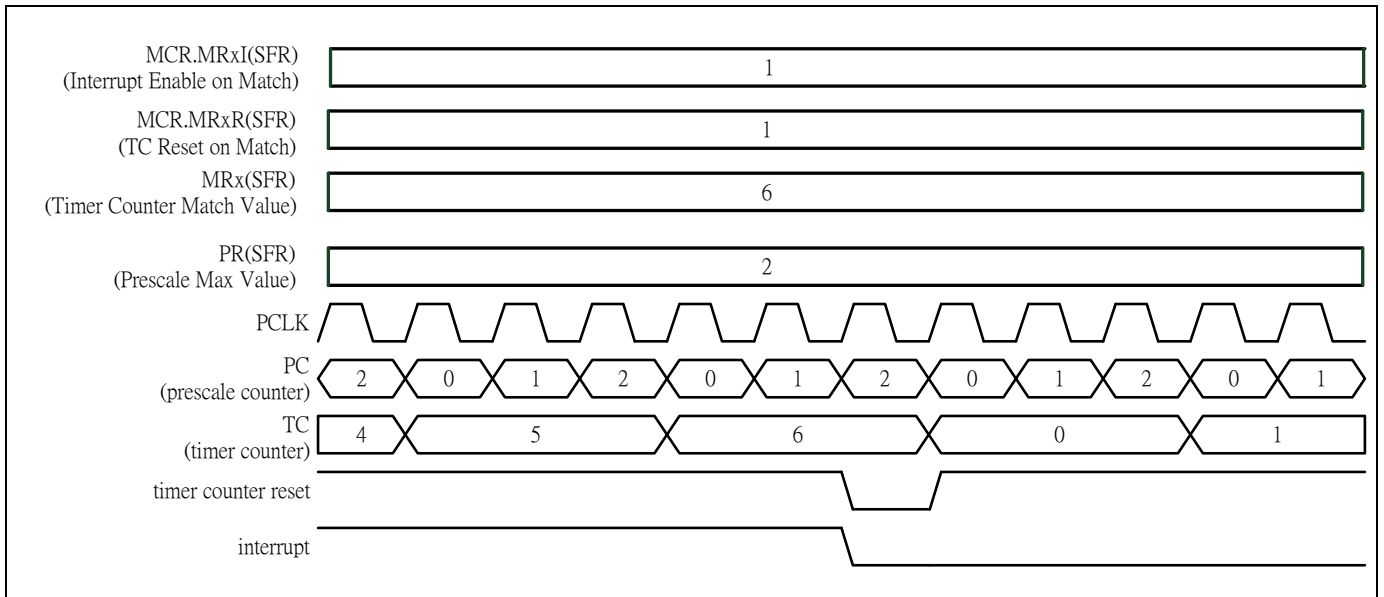


图 12.6.1.1 定时器周期，其中  $PR=2$ 、 $MRx=6$ ，并且使能了在匹配时同时执行中断和重置操作

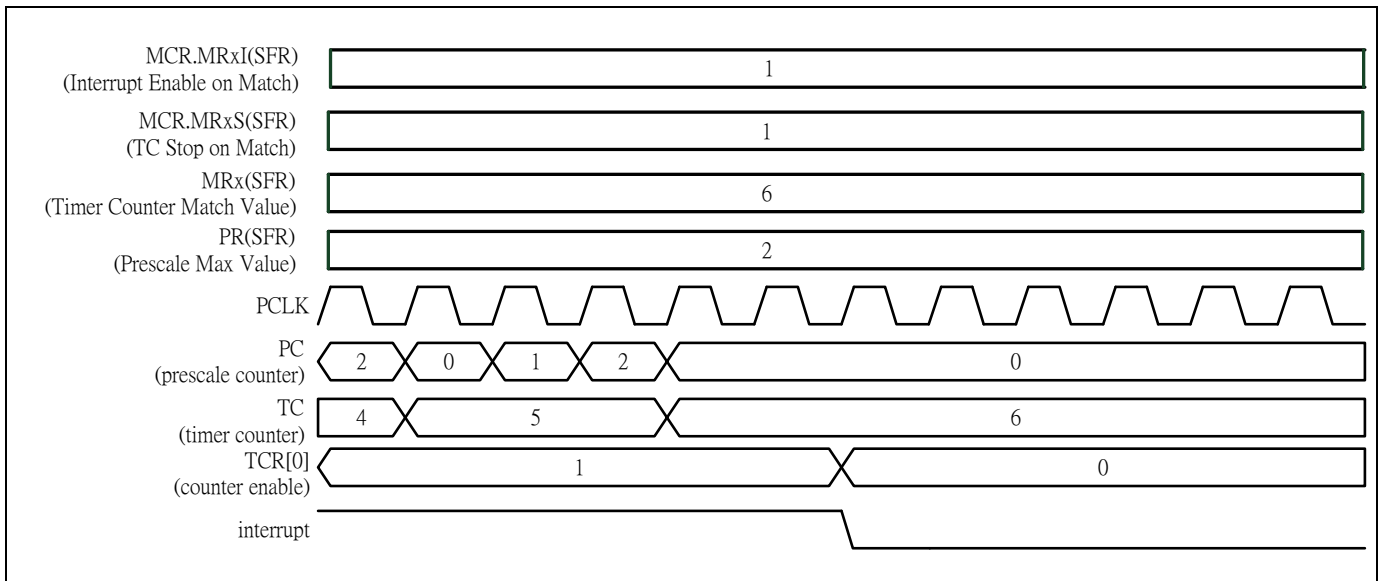


图 12.6.1.2 定时器周期，其中 PR=2、MRx=6，并且使能了在匹配时同时执行中断和停止操作

## 12.6 架构

32位计数器/定时器0和32位计数器/定时器1的功能框图如图 12.7.1所示。

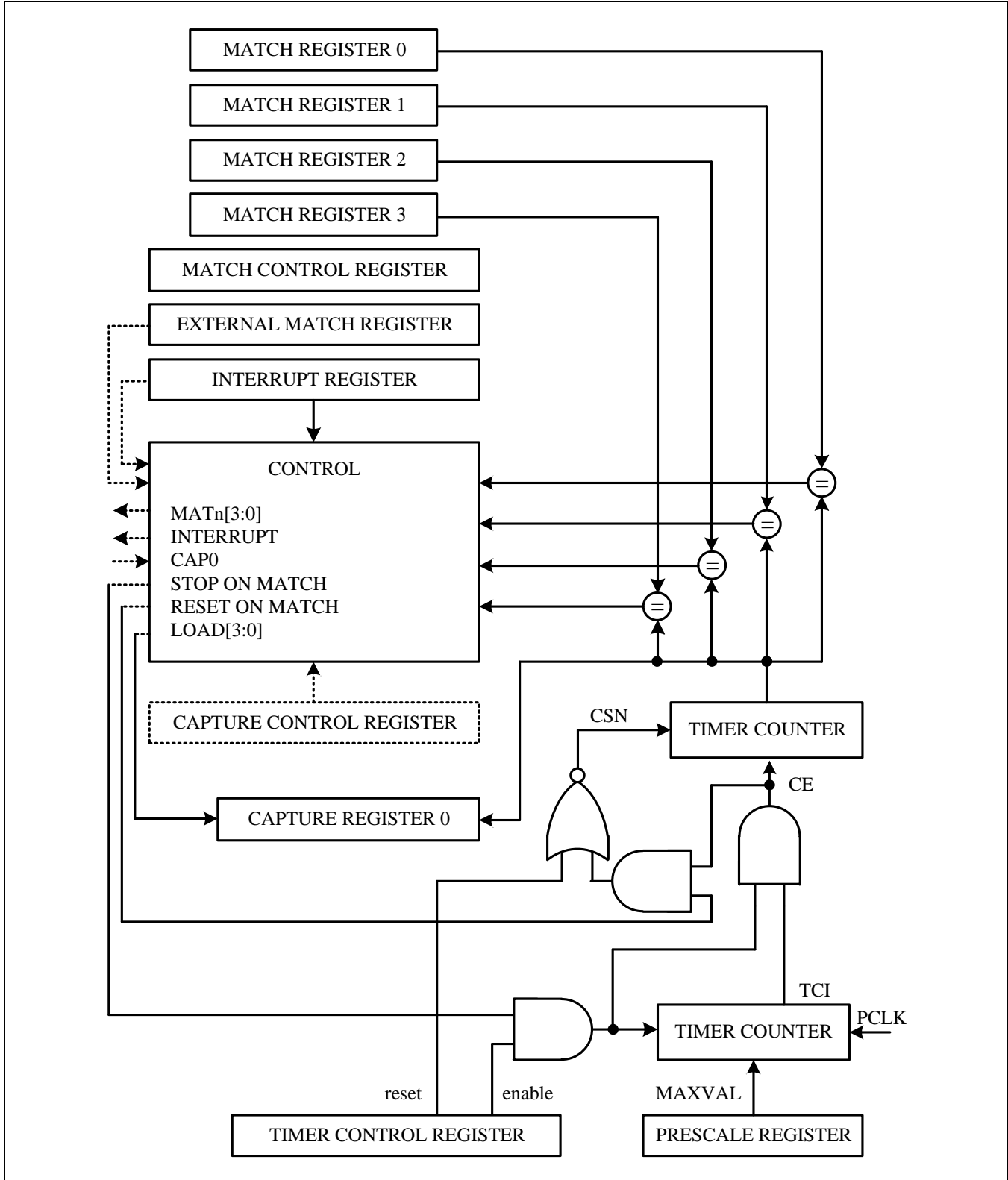


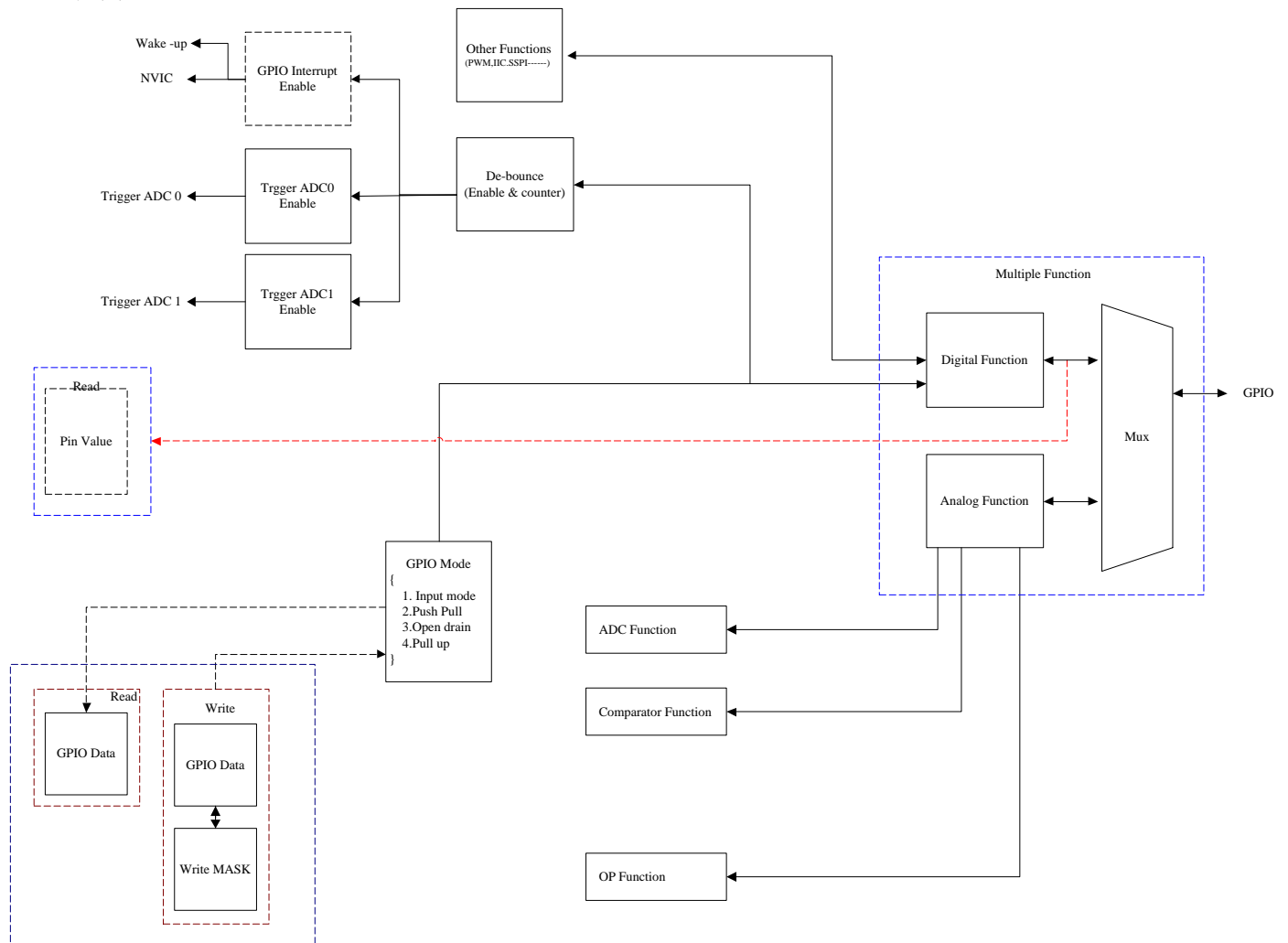
图 12.7.1 32位计数器/定时器功能框图

### 13. IO配置(端口输入/输出)

#### 13.1 特征

- 四种I/O模式:
  - 上拉
  - 推挽
  - 开漏
  - 只输入
- I/O引脚配置为边沿/电平设置的中断源
- 启用引脚中断功能及引脚唤醒功能

#### 13.2 应用



#### 13.3 寄存器说明

表 13.3.1 寄存器概述:

GPIO Port 0 (基址 0x5000 0000)  
 GPIO Port 1 (基址 0x5000 4000)

**GPIO Port 2 (基址 0x5000 8000)**

名称	访问型	地址	描述	重置值
DATA	RW	0x00	GPIO 端口 x 数据寄存器	0x0000FFFF
PIN	RO	0x04	GPIO 端口 x 引脚值	0
MODE	RW	0x08	GPIO 端口 x 模式寄存器	0
WMARK	RW	0x0C	GPIO 端口 x 数据输出写屏蔽寄存器	0
		0x10	保留	0
IEENS0	RW	0x14	GPIO 端口 x 中断使能设置 0 寄存器	0
IEENS1	RW	0x18	GPIO 端口 x 中断使能设置 1 寄存器	0
IDBEN	RW	0x1C	GPIO 端口 x 中断去抖使能寄存器	0
IDBCU	RW	0x20	GPIO 端口 x 中断去抖计数器寄存器	0
IEST	RW	0x24	GPIO 端口 x 中断状态寄存器	0
IESTC	RW	0x28	GPIO 端口 x 中断状态清除寄存器	0
TADC0S0	RW	0x2C	GPIO 端口 x 触发 ADC 0 置 0 寄存器	0
TADC0S1	RW	0x30	GPIO 端口 x 触发 ADC 0 置 1 寄存器	0
	RW	0x34	保留	0
	RW	0x38	保留	0
MF0	RW	0x3C	GPIO 端口 x 多功能 0 寄存器	0
MF1	RW	0x40	GPIO 端口 x 多功能 1 寄存器	0

**13.3.1 GPIO 端口 X 数据寄存器(DATA – 0x5000 0000, 0x5000 4000, 0x5000 8000)**
**表 13.3.1.1 GPIO 端口 X 数据寄存器**

GPIO0 地址 0x5000 0000 : GPIO 端口 0 数据寄存器

GPIO1 地址 0x5000 4000 : GPIO 端口 1 数据寄存器

GPIO2 地址 0x5000 8000 : GPIO 端口 2 数据寄存器

位	符号	内容	描述	重置值
15:0	GPIOXDATA		设定需要输出的电平	FFFF
		0	GPIO 端口将驱动为低电平	
		1	GPIO 端口将驱动高电平	
31:16	-	-	保留	0000

**13.3.2 GPIO 端口 X 引脚值寄存器(PIN – 0x5000 0004, 0x5000 4004, 0x5000 8004)**

读取外部引脚的电平

**表 13.3.2.1 GPIO 端口 X 引脚值寄存器**

GPIO0 地址 0x5000 0004 : GPIO 端口 0 引脚值寄存器

GPIO1 地址 0x5000 4004 : GPIO 端口 1 引脚值寄存器

GPIO2 地址 0x5000 8004 : GPIO 端口 2 引脚值寄存器

位	符号	内容	描述	重置值
0	GPIOXPIN0		引脚 GPIO <sub>n</sub> .0 的输入数据	
1	GPIOXPIN1		引脚 GPIO <sub>n</sub> .1 的输入数据	
2	GPIOXPIN2		引脚 GPIO <sub>n</sub> .2 的输入数据	
3	GPIOXPIN3		引脚 GPIO <sub>n</sub> .3 的输入数据	
4	GPIOXPIN4		引脚 GPIO <sub>n</sub> .4 的输入数据	

5	GPIOXPIN5	引脚 GPIO <sub>n</sub> .5 的输入数据	
6	GPIOXPIN6	引脚 GPIO <sub>n</sub> .6 的输入数据	
7	GPIOXPIN7	引脚 GPIO <sub>n</sub> .7 的输入数据	
8	GPIOXPIN8	引脚 GPIO <sub>n</sub> .8 的输入数据	
9	GPIOXPIN9	引脚 GPIO <sub>n</sub> .9 的输入数据	
10	GPIOXPIN10	引脚 GPIO <sub>n</sub> .10 的输入数据	
11	GPIOXPIN11	引脚 GPIO <sub>n</sub> .11 的输入数据	
12	GPIOXPIN12	引脚 GPIO <sub>n</sub> .12 的输入数据	
13	GPIOXPIN13	引脚 GPIO <sub>n</sub> .13 的输入数据	
14	GPIOXPIN14	引脚 GPIO <sub>n</sub> .14 的输入数据	
15	GPIOXPIN15	引脚 GPIO <sub>n</sub> .15 的输入数据	
31:16	-	保留	0000

### 13.3.3 GPIO 端口 X 模式寄存器(MODE – 0x5000 0008, 0x5000 4008, 0x5000 8008)

上拉模式：支持输出/输入功能,其引脚的状态由内部上拉电阻做控制(源电流较小)

开漏模式：援输出/输入功能,其引脚的高电平由外部上拉电阻做控制,

推挽模式：仅有输出功能,有输出较大电流的能力

只输入模式: 仅有输入功能,引脚为高阻抗, 没有输出能力

表 13.3.3.1 GPIO 端口 X 模式寄存器

GPIO0 地址 0x5000 0008 : GPIO 端口 0 模式寄存器

GPIO1 地址 0x5000 4008 : GPIO 端口 1 模式寄存器

GPIO2 地址 0x5000 8008 : GPIO 端口 2 模式寄存器

位	符号	内容	描述	重置值
2n+1:	GPIOXMODEn		定义 GPIO Port X <sub>n</sub> 的 I/O 模式	0
2n		0	上拉模式	
		1	开漏模式	
		2	推挽模式	
		3	只输入模式	

位	符号	内容	描述	重置值
1:0	GPIOXMODE0		定义 GPIO Port X <sub>0</sub> 的 I/O 模式	0
3:2	GPIOXMODE1		定义 GPIO Port X <sub>1</sub> 的 I/O 模式	0
5:4	GPIOXMODE2		定义 GPIO Port X <sub>2</sub> 的 I/O 模式	0
7:6	GPIOXMODE3		定义 GPIO Port X <sub>3</sub> 的 I/O 模式	0
9:8	GPIOXMODE4		定义 GPIO Port X <sub>4</sub> 的 I/O 模式	0
11:10	GPIOXMODE5		定义 GPIO Port X <sub>5</sub> 的 I/O 模式	0
13:12	GPIOXMODE6		定义 GPIO Port X <sub>6</sub> 的 I/O 模式	0
15:14	GPIOXMODE7		定义 GPIO Port X <sub>7</sub> 的 I/O 模式	0
17:16	GPIOXMODE8		定义 GPIO Port X <sub>8</sub> 的 I/O 模式	0
19:18	GPIOXMODE9		定义 GPIO Port X <sub>9</sub> 的 I/O 模式	0
21:20	GPIOXMODE10		定义 GPIO Port X <sub>10</sub> 的 I/O 模式	0
23:22	GPIOXMODE11		定义 GPIO Port X <sub>11</sub> 的 I/O 模式	0
25:24	GPIOXMODE12		定义 GPIO Port X <sub>12</sub> 的 I/O 模式	0
27:26	GPIOXMODE13		定义 GPIO Port X <sub>13</sub> 的 I/O 模式	0
29:28	GPIOXMODE14		定义 GPIO Port X <sub>14</sub> 的 I/O 模式	0
31:30	GPIOXMODE15		定义 GPIO Port X <sub>15</sub> 的 I/O 模式	0

### 13.3.4 GPIO 端口 X 数据输出写屏蔽寄存器(WMASK – 0x5000 000C, 0x5000 400C, 0x5000 800C)

表 13.3.4.1 GPIO 端口 X 数据输出写屏蔽寄存器



GPIO0 地址 0x5000 000C : GPIO 端口 0 写屏蔽寄存器  
 GPIO1 地址 0x5000 400C : GPIO 端口 1 写屏蔽寄存器  
 GPIO2 地址 0x5000 800C : GPIO 端口 2 写屏蔽寄存器

位	符号	内容	描述	重置值
15:0	GPIOXWMASK		这些位用于保护 DATA 寄存器的相应寄存器位。当设置 1 时，DATA 寄存器的对应位会被保护，不会被更动。	0000
31:16	-	-	保留	0000

位	符号	内容	描述	重置值
0	GPIOXWMASK0		该位用于保护 DATA [0]位。	0
1	GPIOXWMASK1		该位用于保护 DATA [1]位。	0
2	GPIOXWMASK2		该位用于保护 DATA [2]位。	0
3	GPIOXWMASK3		该位用于保护 DATA [3]位。	0
4	GPIOXWMASK4		该位用于保护 DATA [4]位。	0
5	GPIOXWMASK5		该位用于保护 DATA [5]位。	0
6	GPIOXWMASK6		该位用于保护 DATA [6]位。	0
7	GPIOXWMASK7		该位用于保护 DATA [7]位。	0
8	GPIOXWMASK8		该位用于保护 DATA [8]位。	0
9	GPIOXWMASK9		该位用于保护 DATA [9]位。	0
10	GPIOXWMASK10		该位用于保护 DATA [10]位。	0
11	GPIOXWMASK11		该位用于保护 DATA [11]位。	0
12	GPIOXWMASK12		该位用于保护 DATA [12]位。	0
13	GPIOXWMASK13		该位用于保护 DATA [13]位。	0
14	GPIOXWMASK14		该位用于保护 DATA [14]位。	0
15	GPIOXWMASK15		该位用于保护 DATA [15]位。	0

### 13.3.5 GPIO 端口 X 中断使能设置 0 寄存器 (IEENS0 – 0x5000 0014, 0x5000 4014, 0x5000 8014)

表 13.3.5.1 GPIO 端口 X 中断使能置 0 寄存器

GPIO0 地址 0x5000 0014 : GPIO 端口 0 中断使能设置 0 寄存器  
 GPIO1 地址 0x5000 4014 : GPIO 端口 1 中断使能设置 0 寄存器  
 GPIO2 地址 0x5000 8014 : GPIO 端口 2 中断使能设置 0 寄存器

位	符号	内容	描述	重置值
4n+2: 4n	GPIOXIEENSn		该寄存器用于使能 GPIO PortX_n 的中断	
		00	禁用	
		01	下降沿	
		02	上升沿	
		03	两者（下降沿和上升沿）	
		04	低水平	
		05	高水平	
		其他	保留	

位	符号	内容	描述	重置值
2:0	GPIOXIEENS0		该寄存器用于使能 GPIO 端口 X_0 的中断	0
6:4	GPIOXIEENS1		该寄存器用于使能 GPIO 端口 X_1 的中断	0
10:8	GPIOXIEENS2		该寄存器用于使能 GPIO 端口 X_2 的中断	0
14:12	GPIOXIEENS3		该寄存器用于使能 GPIO 端口 X_3 的中断	0
18:16	GPIOXIEENS4		该寄存器用于使能 GPIO 端口 X_4 的中断	0

22:20	GPIOXIEENS5	该寄存器用于使能 GPIO 端口 X_5 的中断	0
26:24	GPIOXIEENS6	该寄存器用于使能 GPIO 端口 X_6 的中断	0
30:28	GPIOXIEENS7	该寄存器用于使能 GPIO 端口 X_7 的中断	0
其他	-	保留	0

### 13.3.6 GPIO 端口 X 中断使能设置 1 寄存器 (IEENS1 – 0x5000 0018, 0x5000 4018, 0x5000 8018)

表 13.3.6.1 GPIO 端口 X 中断使能设置 1 寄存器

GPIO0 地址 0x5000 0018 : GPIO 端口 0 中断使能位 1 寄存器

GPIO1 地址 0x5000 4018 : GPIO 端口 1 中断使能位 1 寄存器

GPIO2 地址 0x5000 8018 : GPIO 端口 2 中断使能位 1 寄存器

位	符号	内容	描述	重置值
4n+2:	GPIOXIEENS <sub>n</sub>		该寄存器用于使能 GPIO PortX_n 的中断 (n >= 8)	
4n		00	禁用	
		01	下降沿	
		02	上升沿	
		03	两者 (下降沿和上升沿)	
		04	低水平	
		05	高水平	
		其他	保留	

位	符号	内容	描述	重置值
2:0	GPIOXIEENS8		该寄存器用于使能 GPIO 端口 X_8 的中断	0
6:4	GPIOXIEENS9		该寄存器用于使能 GPIO 端口 X_9 的中断	0
10:8	GPIOXIEENS10		该寄存器用于使能 GPIO 端口 X_10 的中断	0
14:12	GPIOXIEENS11		该寄存器用于使能 GPIO 端口 X_11 的中断	0
18:16	GPIOXIEENS12		该寄存器用于使能 GPIO 端口 X_12 的中断	0
22:20	GPIOXIEENS13		该寄存器用于使能 GPIO 端口 X_13 的中断	0
26:24	GPIOXIEENS14		该寄存器用于使能 GPIO 端口 X_14 的中断	0
30:28	GPIOXIEENS15		该寄存器用于使能 GPIO 端口 X_15 的中断	0
其他	-	-	保留	0

### 13.3.7 GPIO 端口 X 中断去抖动使能寄存器 (IDBEN – 0x5000 001C, 0x5000 401C, 0x5000 801C)

表 13.3.7.1 GPIO Port X Interrupt De-Bounce Enable Register bit 描述

GPIO0 地址 0x5000 001C : GPIO 端口 0 中断去抖动使能寄存器

GPIO1 地址 0x5000 401C : GPIO 端口 1 中断去抖动使能寄存器

GPIO2 地址 0x5000 801C : GPIO 端口 2 中断去抖动使能寄存器

位	符号	内容	描述	重置值
15:0	GPIOXIDBEN		这用于启用 P0/ P1 /P2 /P3 去抖功能。 如果输入讯号宽度不能大于 IDBCU 寄存器所设定的宽度,则会被视为无效讯号,从而不触发中断	0000
31:16	-	-	保留	0000

### 13.3.8 GPIO 端口 X 中断去抖计数器寄存器 (IDBCU – 0x5000 0020, 0x5000 4020, 0x5000 8020)

表 13.3.8.1 GPIO 端口 X 中断去抖计数器寄存器

GPIO0 地址 0x5000 0020 : GPIO 端口 0 中断去抖计数器寄存器

GPIO1 地址 0x5000 4020 : GPIO 端口 1 中断去抖计数器寄存器

GPIO2 地址 0x5000 8020 : GPIO 端口 2 中断去抖计数器寄存器

位	符号	内容	描述	重置值
15:0	GPIOXIDBCU		去抖计数器(单位为系统频率)	0000
31:16	-	-	保留	0000

### 13.3.9 GPIO 端口 X 中断状态寄存器(IEST – 0x5000 0024, 0x5000 4024, 0x5000 8024)

表 13.3.9.1 GPIO 端口 X 中断状态寄存器

GPIO0 地址 0x5000 0024 : GPIO 端口 0 中断状态寄存器

GPIO1 地址 0x5000 4024 : GPIO 端口 1 中断状态寄存器

GPIO2 地址 0x5000 8024 : GPIO 端口 2 中断状态寄存器

位	符号	内容	描述	重置值
15:0	GPIOXIEST		置 1 位时, 相应的中断被检测到	
31:16	-	-	保留	

### 13.3.10 GPIO 端口 X 中断状态清除寄存器(IESTC – 0x5000 0028, 0x5000 4028, 0x5000 8028)

表 13.3.10.1 GPIO 端口 X 中断状态清除寄存器

GPIO0 地址 0x5000 0028 : GPIO 端口 0 中断状态清除寄存器

GPIO1 地址 0x5000 4028 : GPIO 端口 1 中断状态清除寄存器

GPIO2 地址 0x5000 8028 : GPIO 端口 2 中断状态清除寄存器

位	符号	内容	描述	重置值
15:0	GPIOXIESTC		置 1 位时, 相应的中断被清除(IEST)。(只写) (需软件写 1 清除)	
31:16	-	-	保留	

### 13.3.11 GPIO 端口 X 触发 ADC 0 设置 0 寄存器(TADC0S0 – 0x5000 002C, 0x5000 402C, 0x5000 802C)

利用 GPIO 去执行设定 ADC 模块的 ADCCON[0] 寄存器, 达到外部引脚触发 ADC 工作的功能

表 13.3.11.1 GPIO 端口 X 触发 ADC 0 设置 0 寄存器

GPIO0 地址 0x5000 002C : GPIO 端口 0 触发 ADC 0 设置 0 寄存器

GPIO1 地址 0x5000 402C : GPIO 端口 1 触发 ADC 0 设置 0 寄存器

GPIO2 地址 0x5000 802C : GPIO 端口 2 触发 ADC 0 设置 0 寄存器

位	符号	内容	描述	重置值	
4n+1:	GPIOXTADCSn		该寄存器用于触发 ADC0 的 GPIO PortX_n		
4n		00			禁用
		01			下降沿
		02			上升沿
		03			两者 (下降沿和上升沿)

位	符号	内容	描述	重置值
1:0	GPIOXTADCS0		This register is used to trigger the ADC for GPIO PortX_0	0

5:4	GPIOXTADCS1	This register is used to trigger the ADC for GPIO PortX_1	0
9:8	GPIOXTADCS2	This register is used to trigger the ADC for GPIO PortX_2	0
13:12	GPIOXTADCS3	This register is used to trigger the ADC for GPIO PortX_3	0
17:16	GPIOXTADCS4	This register is used to trigger the ADC for GPIO PortX_4	0
21:20	GPIOXTADCS5	This register is used to trigger the ADC for GPIO PortX_5	0
25:24	GPIOXTADCS6	This register is used to trigger the ADC for GPIO PortX_6	0
29:28	GPIOXTADCS7	This register is used to trigger the ADC for GPIO PortX_7	0
其他	-	保留	0

### 13.3.12 GPIO 端口 X 触发 ADC 0 设置 1 寄存器(TADC0S1 – 0x5000 0030, 0x5000 4030, 0x5000 8030)

利用 GPIO 去执行设定 ADC 模块的 ADCCON[0] 寄存器, 达到外部引脚触发 ADC 工作的功能

表 13.3.12.1 GPIO 端口 X 触发 ADC 0 设置 1 寄存器

GPIO0 地址 0x5000 0030 : GPIO 端口 0 触发 ADC 0 设置 1 寄存器

GPIO1 地址 0x5000 4030 : GPIO 端口 1 触发 ADC 0 设置 1 寄存器

GPIO2 地址 0x5000 8030 : GPIO 端口 2 触发 ADC 0 设置 1 寄存器

位	符号	内容	描述	重置值
4n+1:	GPIOXTADCSn		该寄存器用于触发 ADC1 的 GPIO PortX_n (n >= 8)	
4n		00	禁用	
		01	下降沿	
		02	上升沿	
		03	两者 (下降沿和上升沿)	

位	符号	内容	描述	重置值
1:0	GPIOXTADCS8		该寄存器用于触发 GPIO 端口 X_8 的 ADC	0
5:4	GPIOXTADCS9		该寄存器用于触发 GPIO 端口 X_9 的 ADC	0
9:8	GPIOXTADCS10		该寄存器用于触发 GPIO 端口 X_10 的 ADC	0
13:12	GPIOXTADCS11		该寄存器用于触发 GPIO 端口 X_11 的 ADC	0
17:16	GPIOXTADCS12		该寄存器用于触发 GPIO 端口 X_12 的 ADC	0
21:20	GPIOXTADCS13		该寄存器用于触发 GPIO 端口 X_13 的 ADC	0
25:24	GPIOXTADCS14		该寄存器用于触发 GPIO 端口 X_14 的 ADC	0
29:28	GPIOXTADCS15		该寄存器用于触发 GPIO 端口 X_15 的 ADC	0
其他	-	-	保留	0

### 13.3.13 GPIO 端口 0 多功能 0 寄存器(MF0 – 0x5000 003C)

每个端口上都存在多个功能, 选择到正确的使用功能, 才能正常运作

表 13.3.13.1 GPIO 端口 0 多功能 0 寄存器(MF0 - 0x5000 003C)

位	符号	内容	描述	重置值
3:0	PORTX_0		该寄存器决定了 P0.0 功能	0
		02	ADC_10	
		03	OPA0_N / ACMP0_ON	
		其他	GPIO P0.0	
7:4	PORTX_1		该寄存器决定了 P0.1 功能	0
		03	OPA0_P / ACMP0_0P	
		其他	GPIO P0.1	
11:8	PORTX_2		该寄存器决定了 P0.2 功能	0
		02	ADC_9	
		04	T16B1_CAP2	
		其他	GPIO P0.2	
19:16	PORTX_4		该寄存器决定了 P0.4 功能	0
		02	ADC_7	

	03	ACMP0_1N / ACMP1_1N / ACMP2_1N	
	04	T16B1_CAP0	
	其他	GPIO P0.4	
23:20 PORTX_5		该寄存器决定了 P0.5 功能	0
	02	ADC_6	
	03	ACMP2_1P	
	04	T16B0_CAP2	
	其他	GPIO P0.5	
27:24 PORTX_6		该寄存器决定了 P0.6 功能	0
	02	ADC_5	
	03	ACMP1_1P	
	04	T16B0_CAP1	
	其他	GPIO P0.6	
31:28 PORTX_7		该寄存器决定了 P0.7 功能	0
	02	ADC_4	
	03	ACMP0_1P	
	04	T16B0_CAP0	
	其他	GPIO P0.7	

### 13.3.14 GPIO 端口 1 多功能 0 寄存器(MF0 – 0x5000 403C)

每个端口上都存在多个功能, 选择到正确的使用功能, 才能正常运作

Table 13.3.15.1 GPIO 端口 1 多功能 0 寄存器(MF0 - 0x5000 403C)

位	符号	内容	描述	重置值
3:0	PORTX_0		该寄存器决定了 P1.0 功能	0
			当使用 OPA0 功能时, P1.0 只能设为 OPA0_O 的功能, 否则会影响 OPA0_O 的电压准位	
		03	ACMP0_O	
		04	OPA0_O	
		其他	GPIO P1.0	
7:4	PORTX_1		该寄存器决定了 P1.1 功能	0
		03	OPA1_P / ACMP1_0P	
		其他	GPIO P1.1	

### 13.3.15 GPIO 端口 2 多功能 0 寄存器(MF0 – 0x5000 803C)

每个端口上都存在多个功能, 选择到正确的使用功能, 才能正常运作

表 13.3.16.1 GPIO 端口 2 多功能 0 寄存器(MF0 - 0x5000 803C)

位	符号	内容	描述	重置值
3:0	PORTX_0		该寄存器决定了 P2.0 功能	0
		02	ADC_11	
		03	OPA1_N / ACMP1_0N	
		其他	GPIO P2.0	
7:4	PORTX_1		该寄存器决定了 P2.1 功能	0
			当使用 OPA1 功能时, P2.0 只能设为 OPA1_O 的功能, 否则会影响 OPA1_O 的电压准位	
		01	SSP_MISO	
		03	ACMP1_O	
		04	OPA1_O	
其他	GPIO P2.1			
11:8	PORTX_2		该寄存器决定了 P2.2 功能	0
		01	SSP_SS	
		03	OPA2_P / ACMP2_0P	
其他	GPIO P2.2			
15:12	PORTX_3		该寄存器决定了 P2.3 功能	0
		01	SSP_CLK	
		02	ADC_12	
		03	OPA2_N / ACMP2_0N	
其他	GPIO P2.3			
19:16	PORTX_4		该寄存器决定了 P2.4 功能	0
			当使用 OPA2 功能时, P2.4 只能设为 OPA2_O 的功能, 否则会影响 OPA2_O 的电压准位	
		01	SSP_MOSI	
		03	ACMP2_O	
		04	OPA2_O	
其他	GPIO P2.4			
23:20	PORTX_5		该寄存器决定了 P2.5 功能 ( note : SWD_CLK pad function )	0
		02	ADC_0	
		04	T16B1_CAP2	
		05	UART0_TX	
		06	IIC0_SCL	
		其他	GPIO P2.5	
27:24	PORTX_6		该寄存器决定了 Port2_6 功能 ( note : SWD_DATA pad function )	0
		04	T16B1_CAP1	
		05	UART0_RX	
		06	IIC0_SDA	
		其他	GPIO P2.6	
31:28	PORTX_7		该寄存器决定了 P2.7 功能	0
		01	SSP_SS	
		04	FAULT_B	
		05	UART0_RX	
		06	IIC0_SDA	
		其他	GPIO P2.7	

### 13.3.16 GPIO 端口 2 多功能 1 寄存器(MF1 – 0x5000 8040)

每个端口上都存在多个功能, 选择到正确的使用功能, 才能正常运作

表 13.3.17.1 GPIO 端口 2 多功能 1 寄存器(MF1 - 0x5000 8040)

位	符号	内容	描述	重置值
3:0	PORTX_8		该寄存器决定 P2.8 功能	1

	01	SSP_CLK	
	04	T16B1_CAP0	
	05	UART0_TX	
	06	IIC0_SCL	
	其他	GPIO P2.8	
7:4	PORTX_9	该寄存器决定了 P2.9 功能 ( note : RESET pad function )	1
	01	SSP_MOSI	
	05	UART0_TX	
	06	IIC0_SCL	
	其他	GPIO P2.9	
31:12		保留	



## 14. 闪存控制器(FMC)

### 14.1 概述

OB6625 配备 32K 字节片上嵌入式闪存，用于应用程序存储器（APROM），可通过 ISP 程序进行更新。在线编程（ISP）功能中，用户可以在芯片焊接在 PCB 上时更新程序存储器。OB6625 还为用户提供额外的 NVR（非易失性寄存器）闪存，用于在芯片关闭前存储与应用相关的数据。

### 14.2 特征

- 运行速度高达 72 MHz（连续地址读取访问时为零等待状态）
- 所有嵌入式闪存支持 512 字节页擦除，写入为 1 字节的大小
- 32KB 应用程序存储器（APROM）
- 4 KB 在线编程（ISP）加载程序存储器（LDROM）
- 具有可 512 字节页面擦除的配置数据闪存
- 支持应用程序内编程（IAP）在 APROM 和 LDROM 之间切换代码而无需重置
- 可在线编程（ISP）更新片上 Flash

### 14.3 闪存组织

闪存包含主阵列（32K 字节）和 NVR（非易失性寄存器，6K 字节）。

如果应用程序没有 ISP 功能，主阵列可以分成两块：APROM 和用户选项，

也可分为三块：APROM，LDROM（ISP）和用户选项，用户选项固定为一页大小（512 字节）。LDROM（ISP）和用户选项最大为 4K 字节大小。

NVR 0 ~ 1（非易失性寄存器）。2 页可提供给用户使用。其余不开放

表 14.3.1 闪存组织

Flash定义	没有ISP代码		有ISP代码	
	起始地址	结束地址	起始地址	结束地址
APROM	0x0000_0000	0x0000_7DFF	0x0000_0000	0x0000_6FFF (重置向量)
LDROM	未定义范围		0x0000_7000 (重置向量)	0x0000_7DFF
User Option	0x0000_7E00 ~ 0x0000_7FFF			
NVR 0 ~ 1	0x0010_0200 ~ 0x0010_05FF			

**14.4 框图**

闪存控制器是由 AHB 从机接口，ISP 控制逻辑，写/ ICP 串行接口和闪存宏接口时序控制逻辑组成。闪存控制器的框图如下所示：

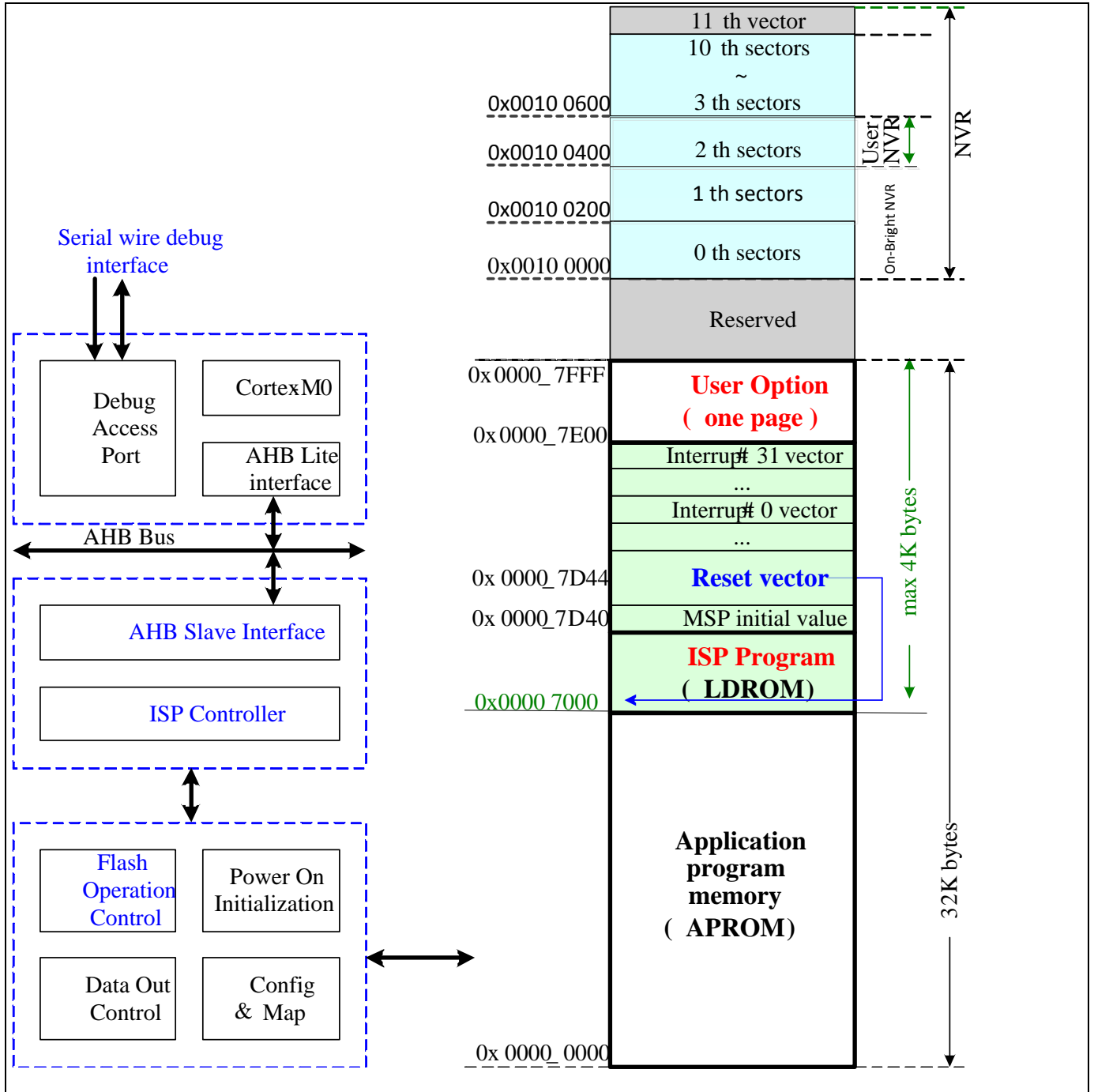


图 14.4.1 32K 闪存控制框图

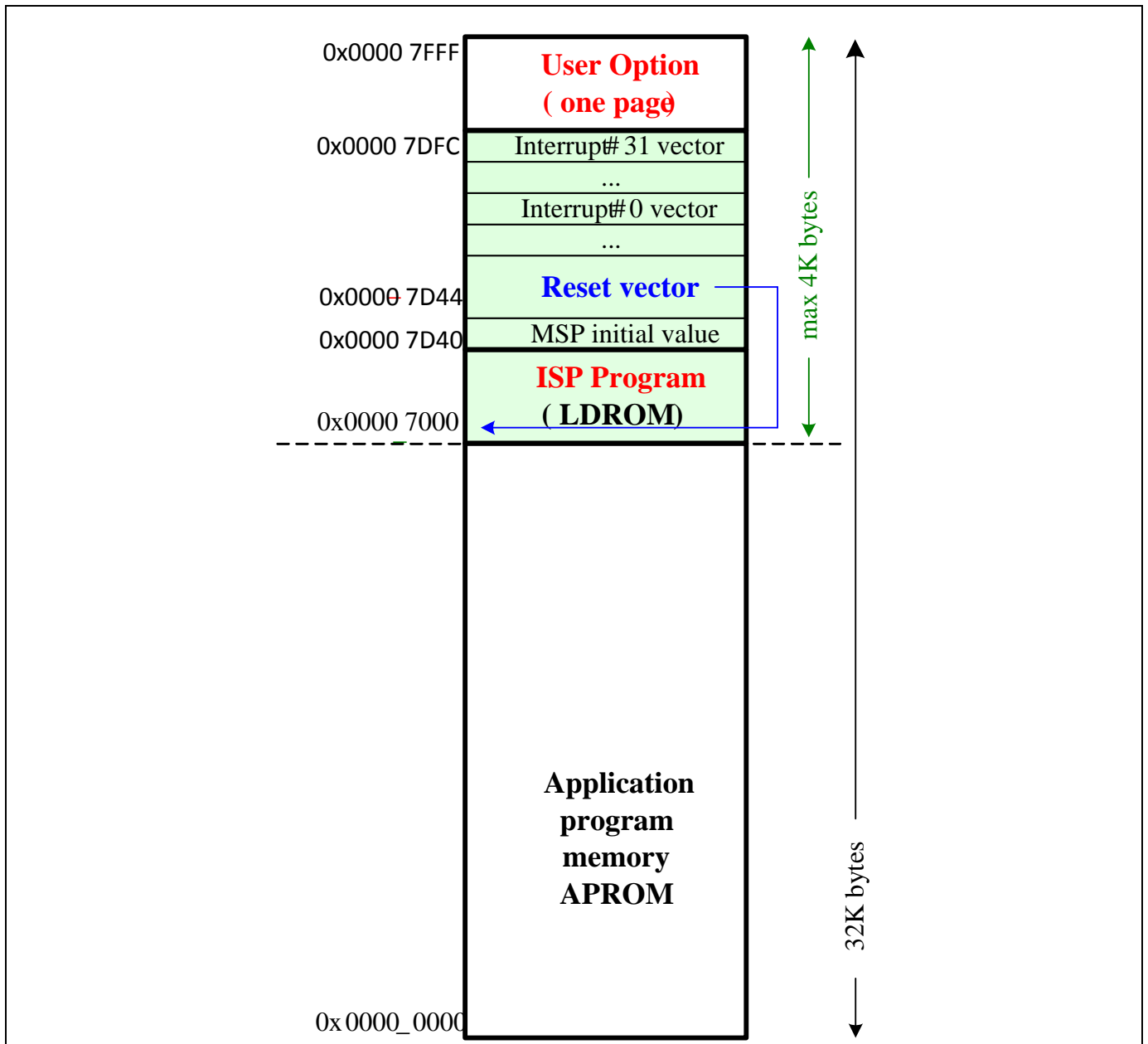


图 14.4.2 32K 闪存框图

## 14.5 闪存控制 (FMC) 功能

### 14.5.1 启动选择功能

OB6625 提供在线编程 (ISP) 功能, 用户可在将芯片安装到 PCB 上时更新程序存储器。专用的 4KB 程序存储器可用于存储 ISP 引导程序。用户可以选择通过引导选择 (BS) 寄存器从用户代码 (APROM) 或 ISP (LDROM 或 RAM) 开始程序提取。

用户选项中的 8 位 ISPET 定义 ISP 启动条件。如果 ISP 启动位全部为零, 则 CPU 地址从用户代码开始, 并且主存储器中没有 ISP 代码。

启动选择 (BS [1: 0]) 默认值为 2'b00, 上电或重置后, 初始过程写入 BS [1] (引导选择 MSB 位) 寄存器的值取决于 ISPET。

表 14.5.2.1 CPU AHB /闪存/ 内存 地址关系

启动选择	CPU AHB 地址 (来源)	Flash 地址 (目的地)
00/01	中断向量 : 0x0000_0000 ~ 0x0000_00BF	中断向量 : 0x0000_0000 ~ 0x0000_00BF
	IAP (没有 ISP) : 0x0000_00C0 ~ 0x0000_7DFF	IAP (没有 ISP) : 0x0000_00C0 ~ 0x0000_7DFF
	IAP (ISP) : 0x0000_00C0 ~ 0x0000_6FFF	IAP (ISP) : 0x0000_00C0 ~ 0x0000_6FFF
10	中断向量 : 0x0000_0000 ~ 0x0000_00BF	中断向量 : 0x0000_7D40 ~ 0x0000_7DFF
	ISP : 0x0000_7000 ~ 0x0000_7D3F	ISP : 0x0000_7000 ~ 0x0000_7D3F

#### 14.5.2 在线更新(ISP)

程序主存储器和 NVR 闪存支持在线更新编程 (ISP)。如果产品刚刚开发或最终产品需要最终用户手中的软件更新, 则 ISP 方法使其变得简单和可能。OB6625 支持 ISP 模式, 允许在软件控制下对器件进行重新编程。

ISP 可在系统板上不移除微控制器的情况下执行。多样接口使 LDROM 能够轻松获得新的程序代码。执行 ISP 最常见的方法是通过 UART。一般来说, PC 通过串口传输新的 APROM 代码。LDROM 接收并通过 ISP 命令重新编程到 APROM 中。OB6625 提供 ISP 引导程序和 PC 应用程序。

#### 14.5.3 ISP 程序

OB6625 支持从启动选择位 (BS) 初始定义的 APROM 或 LDROM (ISP) 启动。如果用户想要更新 APROM 中的应用程序, 用户可以写入 BS = 10 并启动软件重置, 以 LDROM 进行芯片 ISP 引导程序。启动 ISP 功能的第一步是将 ISPEN 位写入 1。

一旦 ISPCCC 寄存器设置正确, 用户可以将 ISPCMD 设置为擦除, 读取或编程。根据闪存的来源, 为目标闪存设置 ISPADR。ISPDAT 可用于设置要编程的数据或用于根据 ISPCMD 返回读取的数据。最后, 设置 ISPTS 控制寄存器的 ISPGO 位来执行相应的 ISP 功能。ISP 控制器的多个寄存器是写保护的, 如 ISPTRIG, ISPCCC。

在软件写入 ISPGO 位后, 如果出现错误情况, ISP 操作没有启动, ISP 失败标志 (ISPPFF) 将被设置 1。ISPPFF 标志须由 S / W 清零。即使 ISPPFF 位保持为 1, 下一个 ISP 程序也可以启动。建议在每次 ISP 操作后检查 ISPPFF 位并清除它。

当 ISPGO 位置 1 时, CPU 将等待 ISP 操作完成, 在此期间; 外设仍照常工作。如果发生任何中断请求, 在 ISP 操作完成之前, CPU 不会对其进行中断请求。当 ISP 操作完成时, ISPGO 位将被硬件自动清零。通过检查该位, 用户可以知道 ISP 操作是否完成。用户应在将 ISPGO 位置 1 的指令旁边添加 ISB 指令, 以确保 ISP 操作后指令的正确执行。

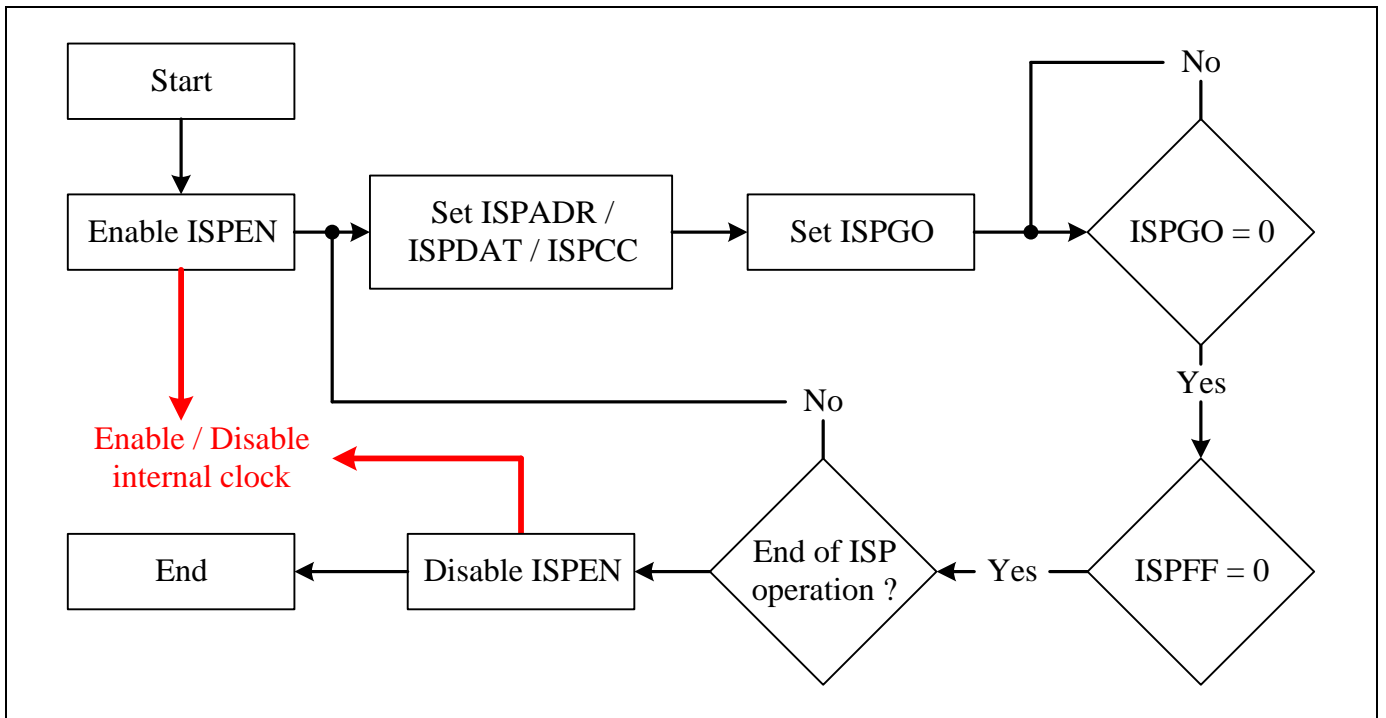


图 14.5.6.1 ISP 程序流程

14.5.4 User Code(IAP) 功能 带/不带 ISP 代码

当 BS [1: 0] = 00/01 时, CPU 运行用户代码。ISP 代码决定是否可以执行芯片擦除功能。在 IAP 开始/结束地址范围 (IAPSA / IAPEA) 的定义中, ISP 允许执行页面擦除和编程功能。

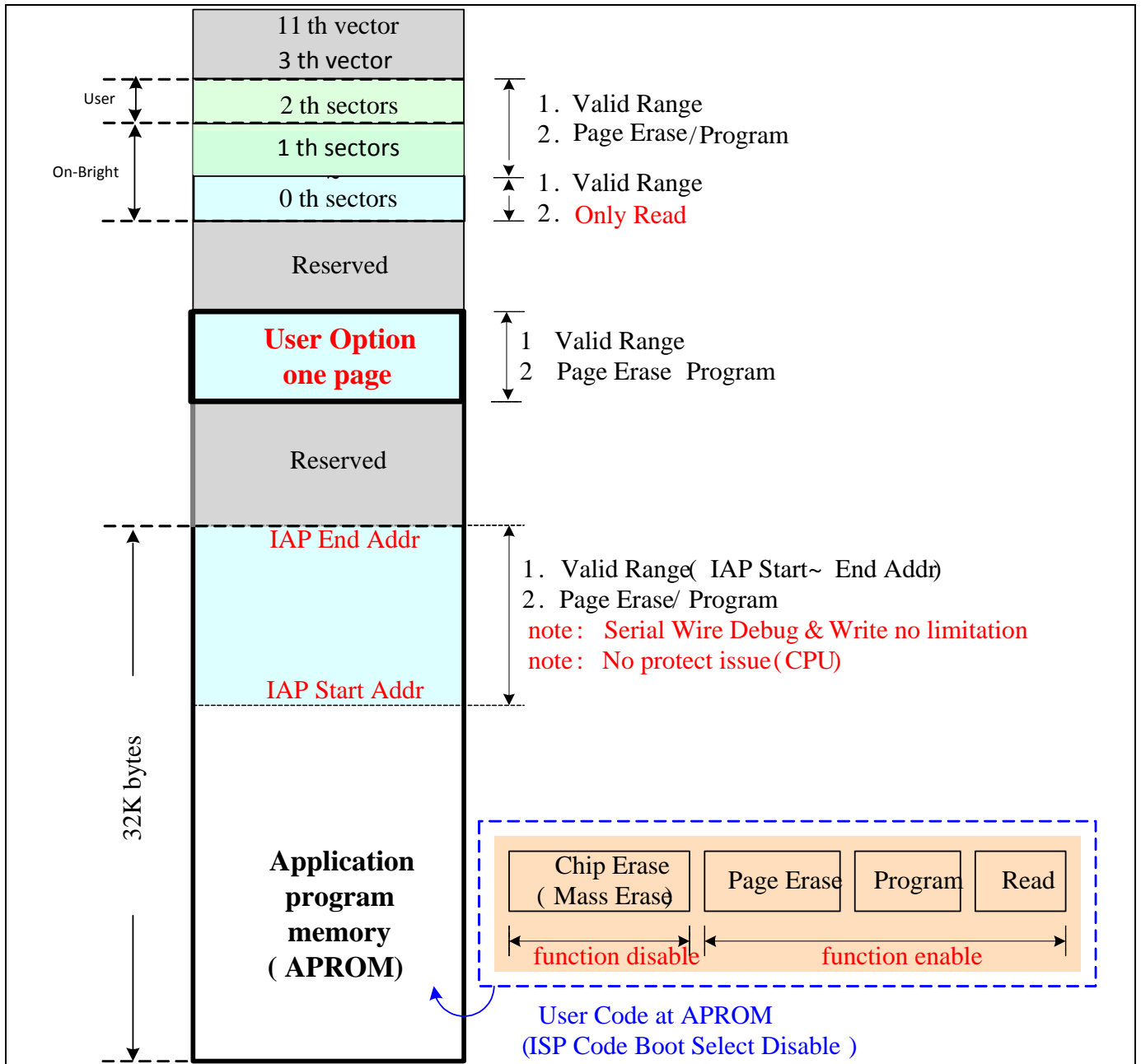


图 14.5.7.1 User Code(IAP) 没有 ISP 代码 功能

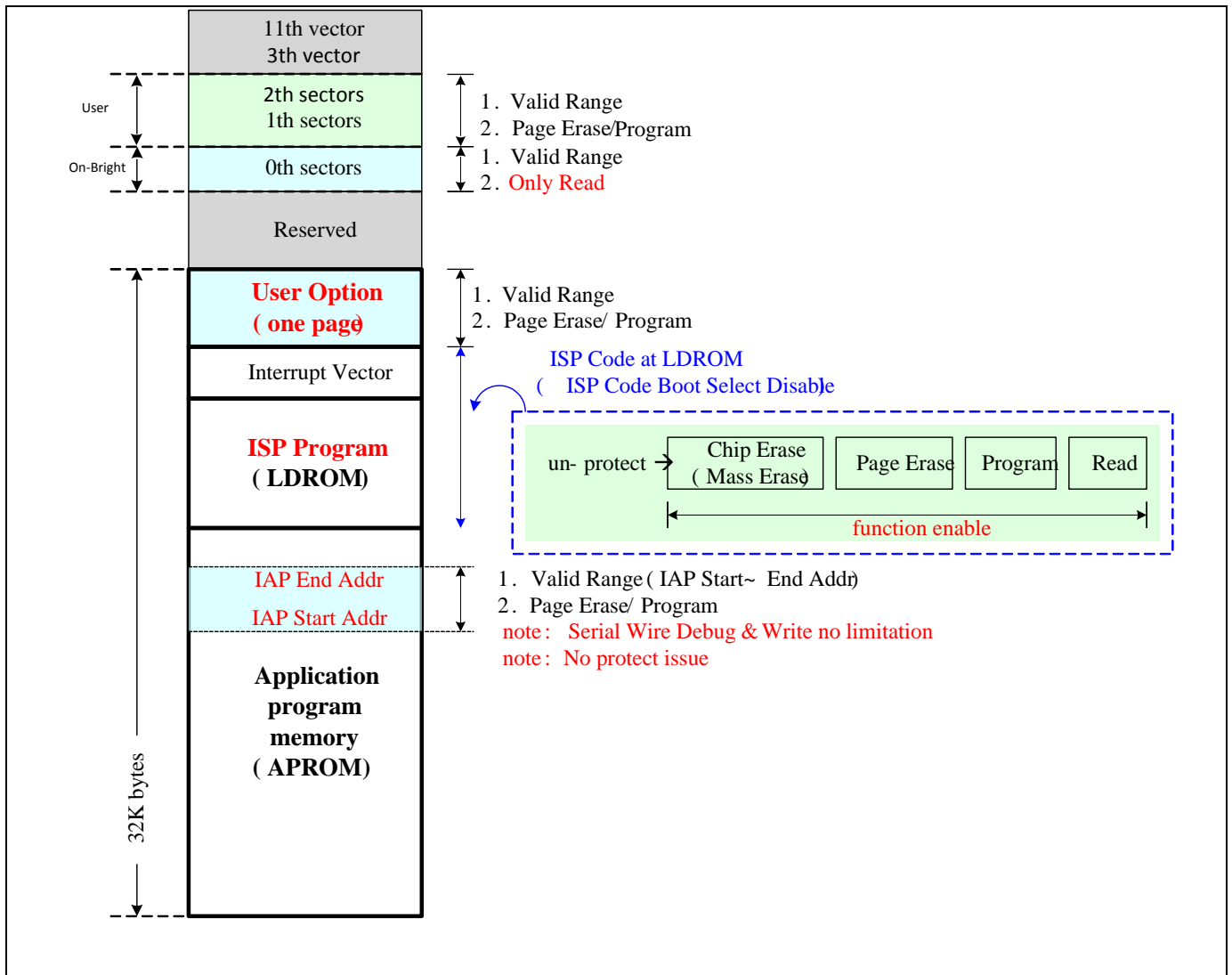


图 14.5.7.2 User Code(IAP) 有 ISP 代码 功能



### 14.5.5 ISP Code 功能

如果 CRPA = 0 (保护使能), ISP 只支持芯片擦除。片擦除后, S / W 必须执行重载功能来改变寄存器状态。芯片擦除/页面擦除/编程不能触及 ISP 范围 (0x0000\_F000~0x0000\_FDFF)。

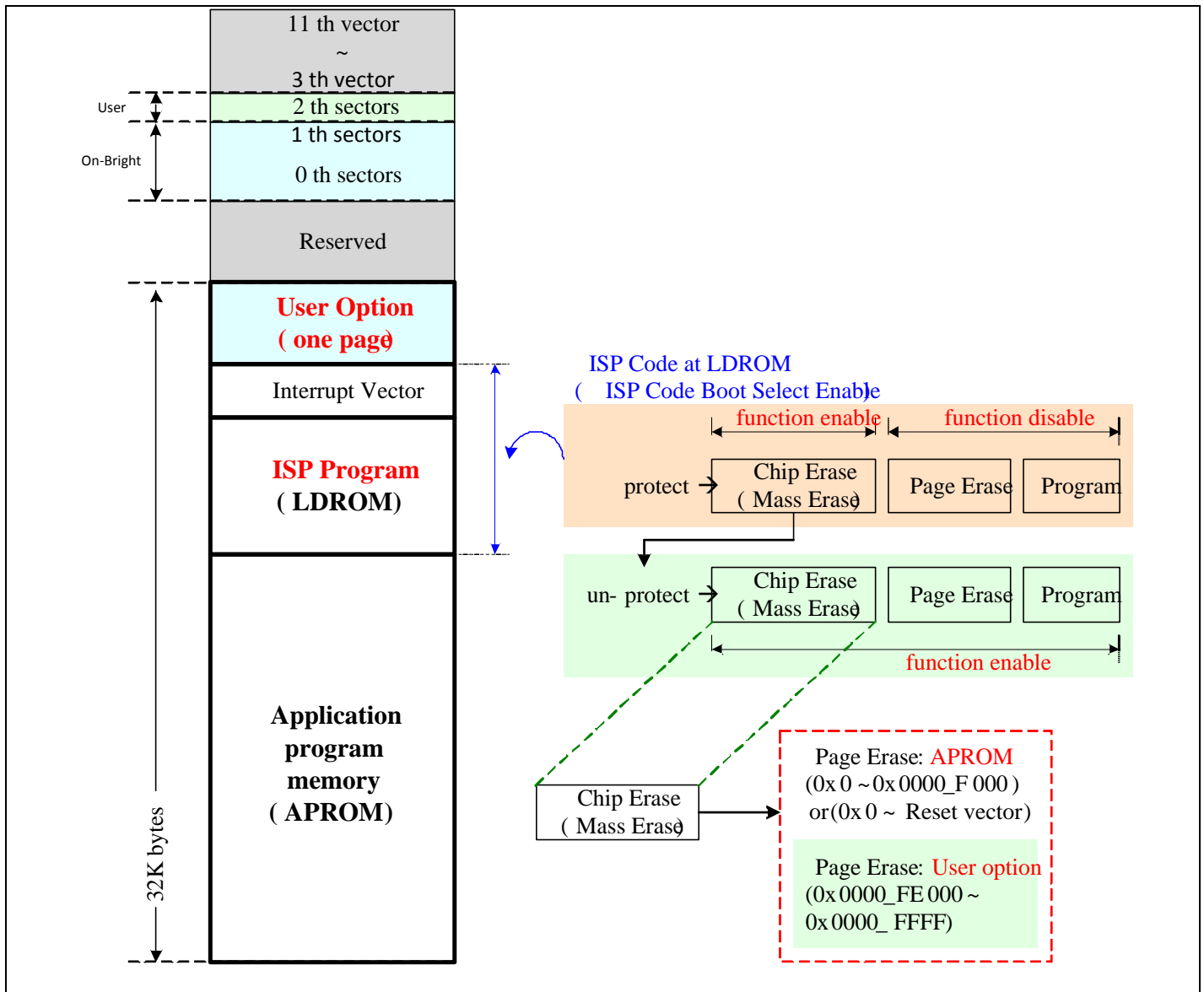


图 14.5.8.1 ISP Code 功能

### 14.6 用户选项 (配置)

表 14.6.1 用户选项概述

名称	地址	描述	重置值
MMCRP	0x0000_7E00	主内存核心读取保护	
ISPET	0x0000_7E04	ISP 启动	

IAPSA	0x0000_7E08	IAP 起始地址
IAPEA	0x0000_7E0C	IAP 结束地址
CTRO	0x0000_7E10	闪存控制 0
	0x0000_7E14	保留
	0x0000_7E18	保留
	0x0000_7E1C	保留
GPIODATA0	0x0000_7E20	GPIO 0 模式
GPIOMODE0	0x0000_7E24	GPIO 0 数据
GPIODATA1	0x0000_7E28	GPIO 1 模式
GPIOMODE1	0x0000_7E2C	GPIO 1 数据
GPIODATA2	0x0000_7E30	GPIO 2 模式
GPIOMODE2	0x0000_7E34	GPIO 2 数据
	0x0000_7E38	保留
	0x0000_7E3C	保留

#### 14.6.1 程序内存读取保护 (MMCRP)

表 14.6.1.1 程序内存读取保护(MMCRPT - 0x0000 7E00)

位	符号	内容	描述	重置值
0	CRPA		核心读取保护所有 32 千字节。	1
		0	保护启用。	
		1	保护禁用。	
31:5	-	-	保留。	

#### 14.6.2 ISP 启动(ISPET)

表 14.6.2.1 ISP 启动(ISPET - 0x0000 7E04)

位	符号	内容	描述	重置值
7:0	ISPET		ISP 启动。如果 ISP 启动位全部为零，则 CPU 地址从用户代码开始，并且主存储器中 0XFF 没有 IPS 代码。	
		00	CPU 地址从用户代码 (APROM) 开始	
		其他	CPU 地址从 ISP 代码开始。	
31:8	-	-	保留。	0

#### 14.6.3 IAP 开始/结束地址(IAPSA/IAPEA)

IAPSA / IAPEA 寄存器默认为 0x0000\_0000 / 0x0000\_7FFF。开机后，从用户选项加载值。当 BS [1: 0] 为 00/01 时，CPU 正在运行应用程序代码。Flash 存储器控制执行页擦除或编程功能，ISP 地址在 IAPST 和 IAPEA (ISP / SWD / 写入无限制) 范围内定义。

范围由用户定义。如果 ISPCE 值是 1，则程序包含 ISP 代码。硬件会自动避开 ISP 范围 (0x0000\_7000 ~ 0x0000\_7DFF)。排除用户选项。如果 IAPSA 和 IAPEA 的值为 0xFFFF\_FFFF，则闪存控制不支持页擦除和编程。

表 14.6.3.1 IAP 起始地址(IAPSA - 0x0000 7E08)

位	符号	内容	描述	重置值
16:0	IAPSA		IAP 起始地址寄存器	0
31:17	-	-	保留	

表 14.6.3.2 IAP 结束地址(IAPEA - 0x0000 7E0C)

位	符号	内容	描述	重置值
16:0	IAPEA		IAP 结束地址寄存器	0
31:17	-	-	保留	

表 14.6.3.3 闪存控制 0 (CTRO - 0x0000 7E10)

位	符号	内容	描述	重置值
15:0			保留	0
23:16	PADE_RESET		重置脚使能。 RESET (芯片焊盘) 可以设置为两种模式。GPIO 或 RESET 功能。 闪存数据 = 0x5A, 或固件写入数据 = 0x5A。GPIO 功能启用。(写入 bit23: bit16)	
24	EQADDR	0	禁用	
		1	使能	
25	BRANCH	0	禁用	
		1	使能	
26	PREDICT	0	禁用	
		1	使能	
27	PREFETCH_BUFF	0	禁用	
		1	使能	
28	PREFETCH_MODE	0	禁用	
		1	使能	
30:29			保留	
31	MCUACC_PBUFEN	0	使能	
		1	禁用	

## 14.7 寄存器说明

表 14.7.1 寄存器概述: 闪存控制器(基址: 0x4003 C000)

名称	访问型	地址	描述	重置值
ISPPC	R/W	0x000	ISP 控制/命令寄存器	0
ISPADR	R/W	0x004	ISP 地址寄存器	0
ISPDAT	R/W	0x008	ISP 数据寄存器	0
ISPTS	R/W	0x00C	ISP 触发/状态寄存器	0

		0x010 - 保留
		0x03C
ICPENTRY	R/W	0x40
		0x044 - 保留
		0x07C

表 14.7.2 寄存器概述: 用户选项 (基址: 0x4003 C000)

名称	访问型	地址	描述	重置值
MMCRPR	RO	0x080	程序内存读取保护寄存器	0xFFFFFFFF
BSCSR	R/W	0x084	引导程序选择控制/状态寄存器	0
		0x088 - 保留		
		0x08C		

#### 14.7.1 ISP 控制/命令寄存器(ISPCC – 0x4003 C000)

表 14.7.1.1 ISP 控制/命令寄存器(ISPCC - 0x4003 C000)

位	符号	内容	描述	重置值
0	ISPEN		ISP 使能 (写保护位) (R/W)	0
		0	ISP 功能使能位。将此位设置为启用 ISP 功能。	
		1	ISP 功能禁用	
3:1	-	-	保留	
7:4	ISPCMD		ISP 命令 (R/W)	0
		0x1	读	
		0x2	编程, ISPADR 必须是有效地址, 一个字节大小; ISPDAT 是编程数据。	
		0x4	页擦除, ISPADR 必须是有效地址, 512 字节页大小。	
		0x8	批量擦除, 擦除整个 APROM。	
		0xF	初始重新加载	
		其他	保留	
23:8	-	-	保留	
31:24	ISPCONP		写保护 ISPCON (W) 当写 ISPCON [31:24] 不等于 0xE8 时, ISPCON 的其余位不能被置位。	

#### 14.7.2 ISP 地址寄存器(ISPADR – 0x4003 C004)

ISPADR 的效果范围为 0x0000\_0000~0x0000\_7FFF (主存储器 32KB) 和 0x0010\_0000~0x0010\_15FF (非易失性寄存器 0~10)。执行页擦除功能后, ISPADR 增加一个扇区地址 (一页 = 128 字 / 512 字节 / 扇区)。执行程序或读取功能后, ISPADR 增加一个字地址。

表 14.7.2.1 ISP 地址寄存器(ISPADR - 0x4003 C004)

位	符号	内容	描述	重置值
31:0	ISPADR		ISP 地址 (R/W) 最大 (32KB) 的嵌入式 Flash, 仅支持字大小。ISP 操作必须保持 ISPADR [1: 0] 为 00b。	

#### 14.7.3 ISP 数据寄存器(ISPDAT – 0x4003 C008)

表 14.7.3.1 ISP 数据寄存器(ISPDAT - 0x4003 C008)

位	符号	内容	描述	重置值
31:0	ISPDAT		ISP 数据 (R/W) 在 ISP 程序操作之前将数据写入该寄存器。ISP 读取操作后从该寄存器读取数据。	0xFFFFFFFF

#### 14.7.4 ISP 触发器/状态寄存器(ISPTS – 0x4003 C00C)

表 14.7.4.1 ISP 触发/状态寄存器(ISPTS - 0x4003 C00C)

位	符号	内容	描述	重置值
0	ISPGO		ISP 启动触发（写保护位）（R/W） 写 1 开始 ISP 操作，当 ISP 操作完成时，该位将被硬件自动清零。	0
		0	ISP 操作完成	
		1	ISP 进展	
3:1	-	-	保留。	
4	ISPPF		ISP 失败标志（写保护位）（R/W）	
23:5	-	-	保留。	
31:24	ISPGOPRT-		写保护 ISPGO（W）	

#### 14.7.5 ICP 启动寄存器(ICPENTRY – 0x4003 C040)

表 14.7.5.1 ICP 启动寄存器 (ICPENTRY - 0x4003 C040)

位	符号	内容	描述	重置值
31:0	ICPENTRY		ICP 启动 当写入 ICPENTRY [31: 0]不等于 0x1234567 时，不能设置 ICPENTRY 寄存器。	0

#### 14.7.6 主存储器代码读保护寄存器(MMCRPR – 0x4003 C080)

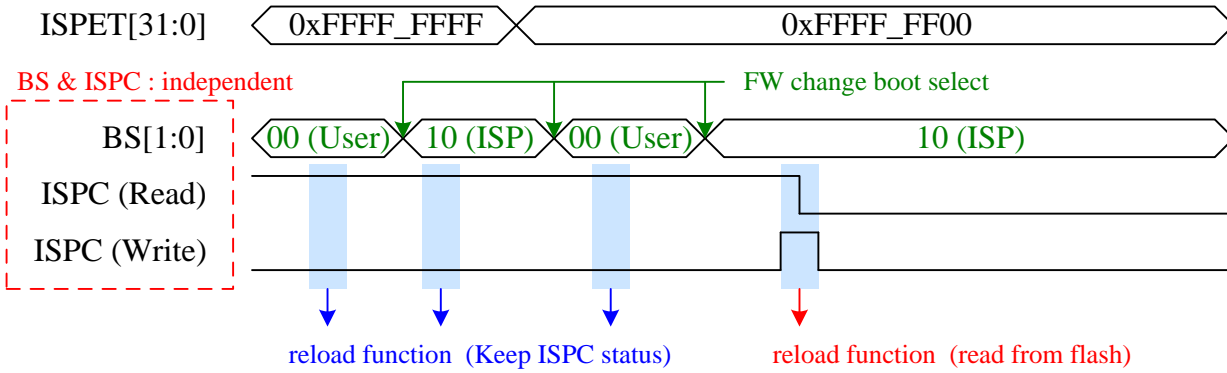
表 14.7.6.1 主内存代码读保护寄存器(MMCRPR - 0x4003 C080)

位	符号	内容	描述	重置值
0	CRPA		代码读取保护所有 32K 字节。（寄存器值）	0
		0	保护使能。	
		1	保护禁用。	
31:1	-	-	保留。	0

#### 14.7.7 引导选择控制/状态寄存器(BSCSR – 0x4003 C084)

表 14.7.7.1 引导选择控制/状态寄存器(BSCSR - 0x4003 C084)

位	符号	内容	描述	重置值
1:0	BS		引导选择（R/W）	
		00/01	IAP 代码启动从闪存上	
		10	ISP 代码启动从闪存上	
		11	ISP 代码启动从内存上。	
2	-	-	保留。	
3	ISPC		ISP 代码 FW 设置位 1，重载功能将读取 ISP 启动（0x0000_FE04）的状态。重载功能后，HW 清除该位（只写） 闪存控制：DFF 状态（只读）	
		0	闪存中没有 ISP 代码	
		1	闪存中有 ISP 代码	
31:4	-	-	保留。	



## 15. 窗口看门狗定时器(WWDT)

### 15.1 概述

看门狗定时器的目的是在指定的周期时间内执行系统重置，以防止软件由于任何不可预知的情况而无法运行到无法控制的状态。

### 15.2 特征

6 位递减计数器 (WWDTVAL [5: 0]) 和 6 位比较值 (WWDTCCR [21:16] - WINCMPvalue) 使周期时间更加灵活。

• 可选的最大 11 位 WWDT 时钟预分频比 (WWDTCCR [11: 8] - PERIODSEL 值) 可使 WWDT 超时间隔变化。

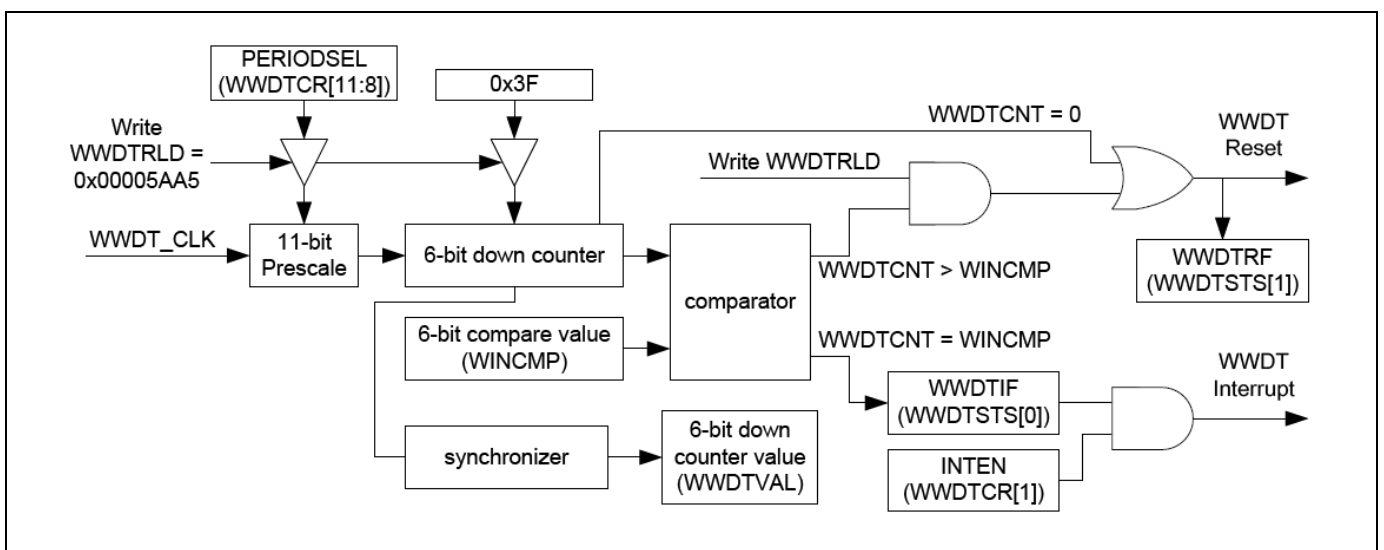


图 15.2.1 看门狗定时器框图

### 15.3 一般描述

看门狗定时器包括一个具有可编程的 6 位递减计数器，用于定义不同的超时间隔。

看门狗定时器的时钟源基于内部 20 kHz 振荡器，具有 11 位预分频值的可编程最大值。此外，可编程 11 位预分频值由 PERIODSEL (WWDTCCR [11: 8] WWDT 预分频周期选择) 控制，PERIODSEL 和预分频值的相关值列于下表。

表 15.3.1 看门狗定时器预分频值选择

PERIODSEL	Prescaler 内容	Time-out Period	Max. Time-out Interval (WWDT_CLK = 20kHz)
0000	1	1 * 64 * TWWDT	3.2 ms
0001	2	2 * 64 * TWWDT	6.4 ms
0010	4	4 * 64 * TWWDT	12.8 ms
0011	8	8 * 64 * TWWDT	25.6 ms
0100	16	16 * 64 * TWWDT	51.2 ms
0101	32	32 * 64 * TWWDT	102.4 ms
0110	64	64 * 64 * TWWDT	204.8 ms
0111	128	128 * 64 * TWWDT	409.6 ms
1000	192	192 * 64 * TWWDT	614.1 ms



1001	256	$256 * 64 * T_{WWDT}$	819.2 ms
1010	384	$384 * 64 * T_{WWDT}$	1.22 s
1011	512	$512 * 64 * T_{WWDT}$	1.63 s
1100	768	$768 * 64 * T_{WWDT}$	2.45 s
1101	1024	$1024 * 64 * T_{WWDT}$	3.27 s
1110	1536	$1536 * 64 * T_{WWDT}$	4.91 s
1111	2048	$2048 * 64 * T_{WWDT}$	6.55 s

芯片上电或重置后，通过软件将 WWDTEN (WWDTCR [0] WWDT 使能) 位设置为 1，只能使能看门狗定时器一次，WWDT 递减计数器将从 0x3F 开始计数，并且可以通过软件停止，除非芯片再次被重置。

在通过 WWDT 计数器进行递减计数时，如果 WWDT 计数器值等于 WINCMP (WWDTCR [21:16] WWDT 比较寄存器) 值，则 WWDTIF (WWDTSR [0] WWDT 比较匹配中断标志) 置 1；如果通过软件将 WWDTIE (WWDTCR [1] WWDT 中断使能) 也设置为 1，则在硬件将 WWDTIF 设置为 1 时也会生成 WWDT 超时中断信号。

当 WWDT 计数器值达到 0 时，会产生 WWDT 超时重置信号。在 WWDT 计数器递减计数到 0 之前，软件可以向 WWDTRLD 寄存器写入 0x00005AA5 以将 WWDT 内部计数器值重新加载到 0x3F，以防止在当前 WWDT 计数器值 (WWDTCVR 值) 等于或小于 WINCMP 值时发生 WWDT 超时重置。

如果当前 WWDT 计数器值 (WWDTCVR 值) 大于 WINCMP 值并且软件将 0x00005AA5 写入 WWDTRLD 寄存器，将产生 WWDT 重置信号，导致芯片重置。

为防止程序运行中误将看门狗定时器计数器计数禁止，控制寄存器 WWDTCR 只能在芯片上电或重置后写入一次。除非芯片重置，否则在软件使能 WWDTEN 位时，软件无法禁止窗口看门狗定时器计数器计数 (WWDTEN)，更改超时预分频周期 (PERIODSEL) 或更改比较值 (WINCMP)。

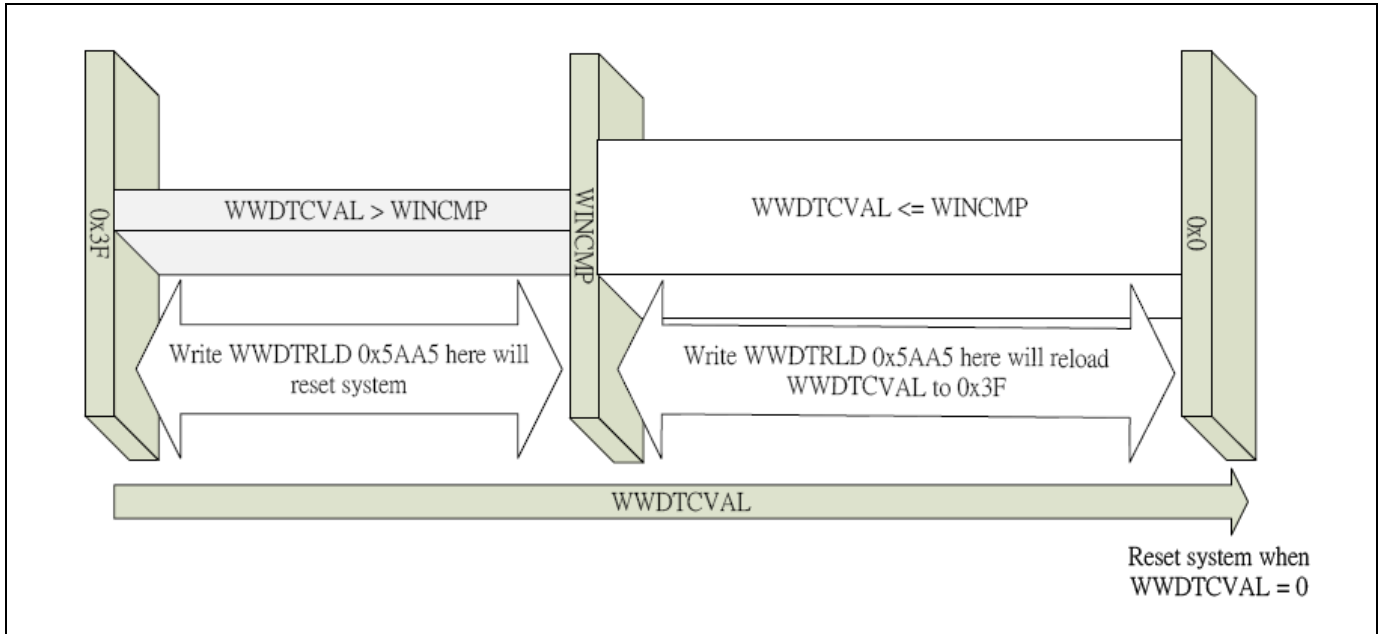


图 15.3.1 看门狗定时器重置和重载行为

当软件向 WWDTRLD 寄存器写入 0x00005AA5 为了将 WWDTCVAL 计数器值重新装载到 0x3F 时，需要 6 个 WWDTC 时钟来同步重装命令以实际执行重装操作。这意味着如果软件将 PERIODSEL WWDTCR [11: 8] WWDTC 预分频周期选择为 0，预分频值应该为 1，并且 WINCMP (WWDTCR [21:16] WWDTC 窗口比较寄存器) 值必须大于 6；否则，由软件编写 WWDTRLD 将 WWDTCVAL 计数器值重新装载到 0x3F 不可用，且 WWDTC 超时重置会一直发生。

表 15.3.2 WINCMP 设置限制 (请参考表 15.3.1 看门狗定时器预分频值选择)

预分频值	有效的 WINCMP 值
1 (PERIODSEL=0000)	0x7 ~ 0x3F
2 (PERIODSEL=0001)	0x4 ~ 0x3F
3 (PERIODSEL=0010)	0x2 ~ 0x3F
4~FF (PERIODSEL=0100~1111)	0x1 ~ 0x3F

## 15.4 寄存器概述

表 15.4.1 寄存器概述：看门狗定时器(基址: 0x4000 4000)

名称	访问型	地址	描述	重置值
WWDTRLD	WO	0x000	看门狗定时器重载计数器寄存器。	0
WWDTCR	R/W	0x004	看门狗定时器控制寄存器。	0
WWDTSR	R/W	0x008	看门狗定时器状态寄存器。	0
WWDTCVR	RO	0x00C	看门狗定时器计数器值寄存器。	0

### 15.4.1 看门狗定时器重载计数器寄存器(WWDTRLD – 0x4000 4000)

表 15.4.1.1 看门狗定时器重载计数器寄存器(WWDTRLD - 0x4000 4000)

位	符号	描述	重置值
31:0	WWDTRLD	将 0X00005AA5 写入该寄存器会将窗口看门狗定时器计数器值重新加载到 0x3F	0000_0000

注意：软件只能写入 WWDTRLD 重新装入 WWDT 计数器值  
 WWDT 计数器值介于 0 和 WINCMP 之间。 如果当 WWDT 计数器值大于 WINCMP 时软件写入 WWDTRLD，则 WWDT 重置信号将立即生成。

#### 15.4.2 看门狗定时器控制寄存器(WWDTCR – 0x4000 4004)

表 15.4.2.1 看门狗定时器控制寄存器(WWDTCR - 0x4000 4004)

位	符号	内容 描述	重置值
0	WWDTEN	使能 WWDT 置位该位以启用看门狗定时器计数器计数。 0 看门狗定时器计数器停止。 1 看门狗定时器计数器正在开始计数。	0
1	WWDTIE	使能 WWDT 中断 0 WWDT 超时中断功能如果 WWDTIF (WWDTSR [0] WWDT 比较匹配中断标志) 为 1，则禁用。 1 WWDT 超时中断功能如果 WWDTIF (WWDTSR [0] WWDT 比较匹配中断标志) 为 1，则启用	
7:2	-	- 保留	
11:8	PERIODSEL	WWDT 预分期期间选择 这 4 位选择 WWDT 计数器周期的预分频周期。	
15:12	-	- 保留	
21:16	WINCMP	设置该寄存器可调整有效的重载窗口。 注意：当前 WWDT 计数器 WWDT 计数器值介于 0 和 WINCMP (请参考表 15.3.2 WINCMP 设置限制) 之间时，软件才能写入 WWDTRLD 以重新加载 WWDT 计数器值。 如果当前 WWDT 计数器值大于 WICMP 时软件写入 WWDTRLD，则 WWDT 复位信号将立即产生。	
30:22	-	- 保留	
31	DBGACK_WWDT	仿真调试模式确认禁用 0 如果系统处于调试模式，则 WWDT 计数器停止 1 即使系统处于调试模式，WWDT 仍然计数	

#### 15.4.3 看门狗定时器状态寄存器(WWDTSR – 0x4000 4008)

表 15.4.3.1 看门狗定时器状态寄存器(WWDTRLD - 0x4000 4008)

位	符号	描述	重置值
0	WWDTIF	WWDT 比较匹配中断标志 当 WWDT 计数器值与 WWCMP 匹配时，该位置 1。通过写入 1 将该位清除为 0。	
1	WWDTRF	WWDT 重置标志 当 WWDT 计数器递减计数到 0 或在当 WWDT 计数器值大于 WINCMP 时写入 WWDTRLD 时，芯片将被重置并将该位设置为 1。通过写入 1 将该位清除为 0。	
31:2	-	- 保留	

#### 15.4.4 看门狗定时器计数器值寄存器(WWDTCVR – 0x4000 400C)

表 15.4.4.1 看门狗定时器计数器值寄存器(WWDTCVR - 0x4000 400C)

位	符号	描述	重置值
5:0	WWDTCVAL	WWDT 计数器值 该寄存器反映当前的 WWDT 计数器值并且是只读的	
31:6	-	- 保留	

## 16. 功率控制PWM 模块

copy from PICROCHIP PIC18F2331/2431/4331/4431

### 16.1 介绍

功率控制 PWM 模块简化了产生多个同步脉宽调制 (PWM) 输出的任务，用于控制电机控制器和功率转换应用。

支持以下电源和运动控制应用:

- 三相和单相交流感应电机
- 开关磁阻电机
- 无刷直流 (BLDC) 电机
- 不间断电源 (UPS)
- 直流有刷电机

### 16.2 特征

支持以下特征:

- 多达 6 个带有 4 个占空比发生器的 PWM I/O 引脚。引脚可以配对以获得完整的半桥控制。
- 高达 16 位分辨率，取决于 PWM 周期。
- “On-the-fly” PWM 频率变化改变。
- 边缘和中心对齐输出模式。
- 单脉冲发生模式。
- 成对 PWM 带可编程死区时间控制。
- 支持不对称更新于中断中央对齐模式。
- 电动换向电机 (ECM) 操作的输出倍率; 例如 BLDC。
- 特殊事件比较器用于安排其它外设事件。
- 处于调试模式时，PWM 输出禁用功能将 PWM 输出设置为非活动状态。

### 16.3 模块功能

PWM 模块支持多种特定电源和电机控制应用的操作模式。

以下针对 PWM 模块与操作模式分开说明:

- PWM 比较
- PWM 中断
- PWM 死区
- PWM 输出
- PWM 时基
- PWM 特殊事件触发器

### 16.4 PWM 时基

PWM 时基由 24 位定时器提供

PWM 时基通过 PWMCON 寄存器进行配置。通过分别设置或清除 PWMCON 寄存器来启用或禁用时基。

PWM 时基可以配置为三种不同的工作模式：

- 自由运行模式 - 自由运行模式产生边沿对齐 PWM 生成。
- 单发模式 - 单次模式允许 PWM 模块支持某些电子换向电机（ECM）的脉冲控制并产生边沿对齐操作。
- 连续递增/递减计数模式 - 递增/递减计数模式产生中心对齐的 PWM 生成。

这三种模式由 PWMCON 寄存器选择。可参考 PWMCON - 0x5001 C000 说明

#### 16.4.1 自由运行模式

在自由运行模式下，PWMCU0 / PWMCU1 寄存器将开始递增计数，直到该值与 MAX0 / MAX1 匹配。PWMCU0 / PWMCU1 寄存器将在下一个输入时钟沿重置，并且 PWMCU0 / PWMCU1 将继续递增计数。

可参考 OUTCON0A0 - 0x5001 C018 ~ OUTCON2B0 - 0x5001 C02C 说明

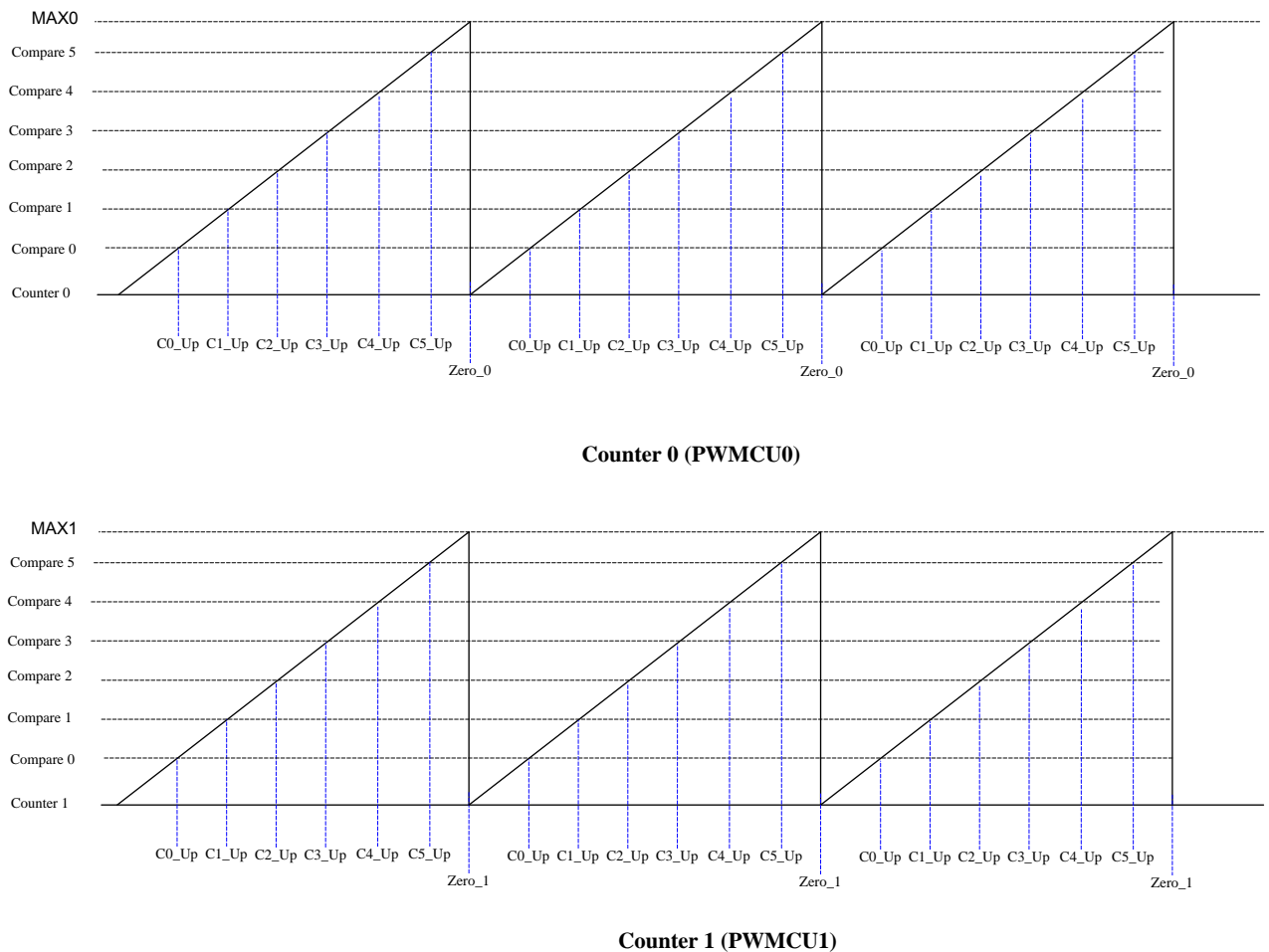


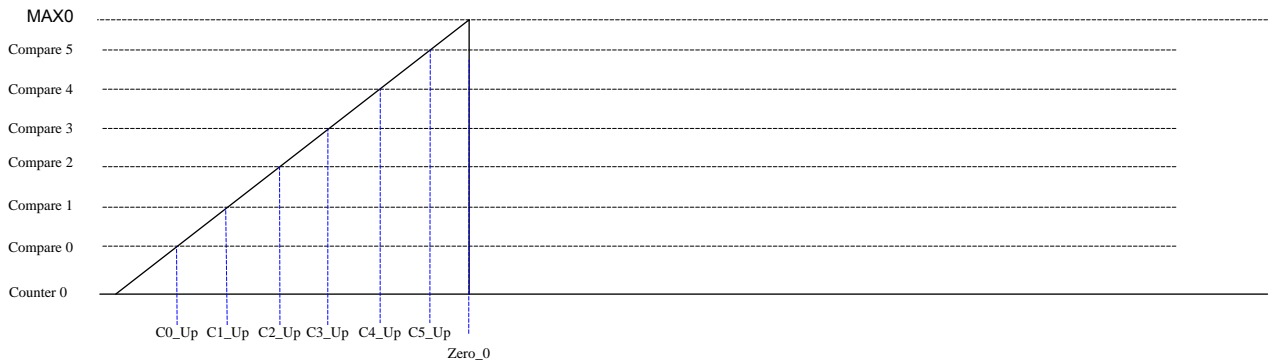
图16.4.1 自由运行模式框图

**16.4.2 单发模式**

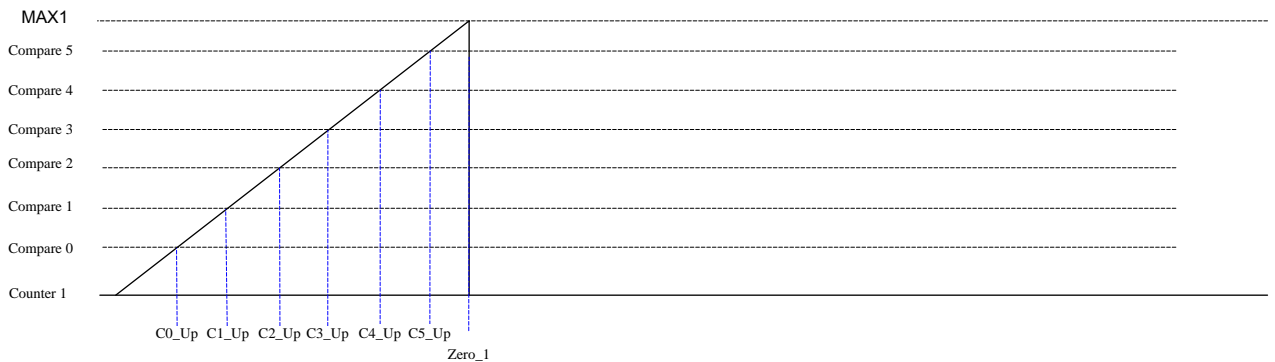
在单发模式下，PWMCU0 / PWMCU1 寄存器将开始递增计数，直到该值与 MAX0 / MAX1 匹配。PWMCU0 / PWMCU1 寄存器将在下一个输入时钟沿重置清零，并且 EN0 及 EN1 不会自动清零，如果需要第二次单发可手动软件清零后再始能。

可参考 OUTCON0A0 - 0x5001 C018 ~ OUTCON2B0 - 0x5001 C02C 说明

可参考 PWMCON - 0x5001 C000 说明



**Counter 0 (PWMCU0)**



**Counter 1 (PWMCU1)**

图16.4.2 单发模式框图

**16.4.3 连续递增/递减计数模式**

在连续递增/递减计数模式下，PWMCU0 / PWMCU1 递增注册，直到该值与 PWMCU0 / PWMCU1 寄存器匹配。在接下来的输入时钟沿，PWMCU0 / PWMCU1 递减注册。PWMSTA 寄存器中的位 0 是只读的并指示计数方向。当定时器递减计数时，位 0 被置成 1。

可参考 OUTCON0A0 - 0x5001 C018 ~ OUTCON2B0 - 0x5001 C02C 说明

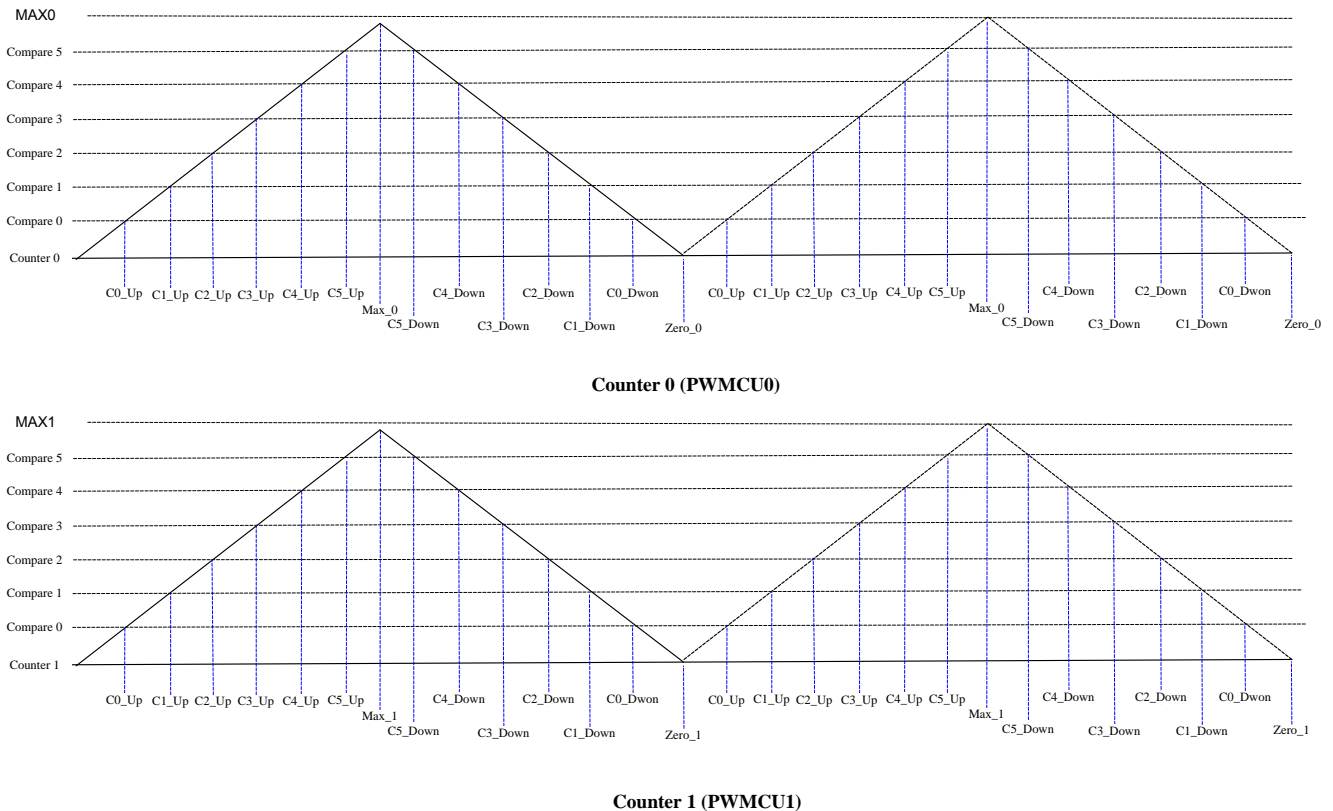


图 16.4.3 连续向上/向下计数模式框图

**16.5 PWM 时钟选择**

PWM 计数器 0/1 的来源为系统频率.故 PWM 模块的所以设定速度为系统频率  
可参考 PWMCS - 0x5001 C00C 说明

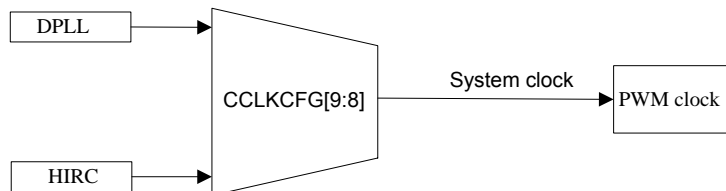


图 16.5.1 PWM 时钟选择框图



**16.6 PWM 计数器选择**

PWM 具有 2 个独立的计数器，可以在不同的时间点对其进行计数。

PWMCON [0] / PWMCON [8] 使能内部 counter0 / counter1。PWMCU0 / PWMCU1 显示内部 counter0 / counter1 的值。

使用 PWMCS[3:0]寄存器选择 PWMCMPx(0~5)比较 PWMCU0 或 PWMCMPx(0~5)比较 PWMCU1，若相等时，则发生相关事件。

可参考 PWMCON - 0x5001 C000,PWMCS - 0x5001 C00C 说明

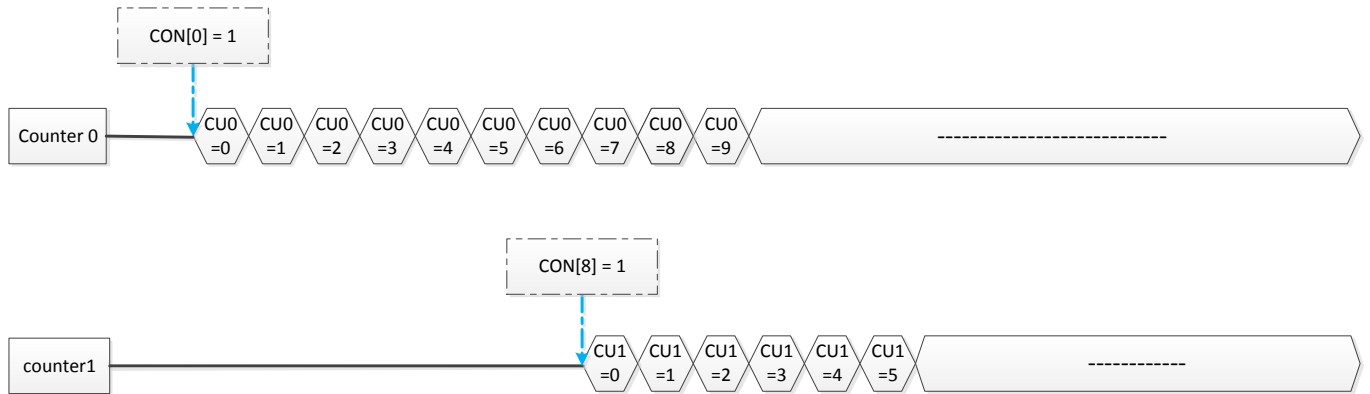


图 16.6.1 PWM 计数器选择框图

### 16.7 PWM 中断

PWM 发生器也采用相同的 12 个计数器事件产生中断。任何事件或一组事件都可以被选为中断源，当发生任何选定的事件时，会产生一个中断。

可参考 IEENS - 0x5001 C0A0 说明

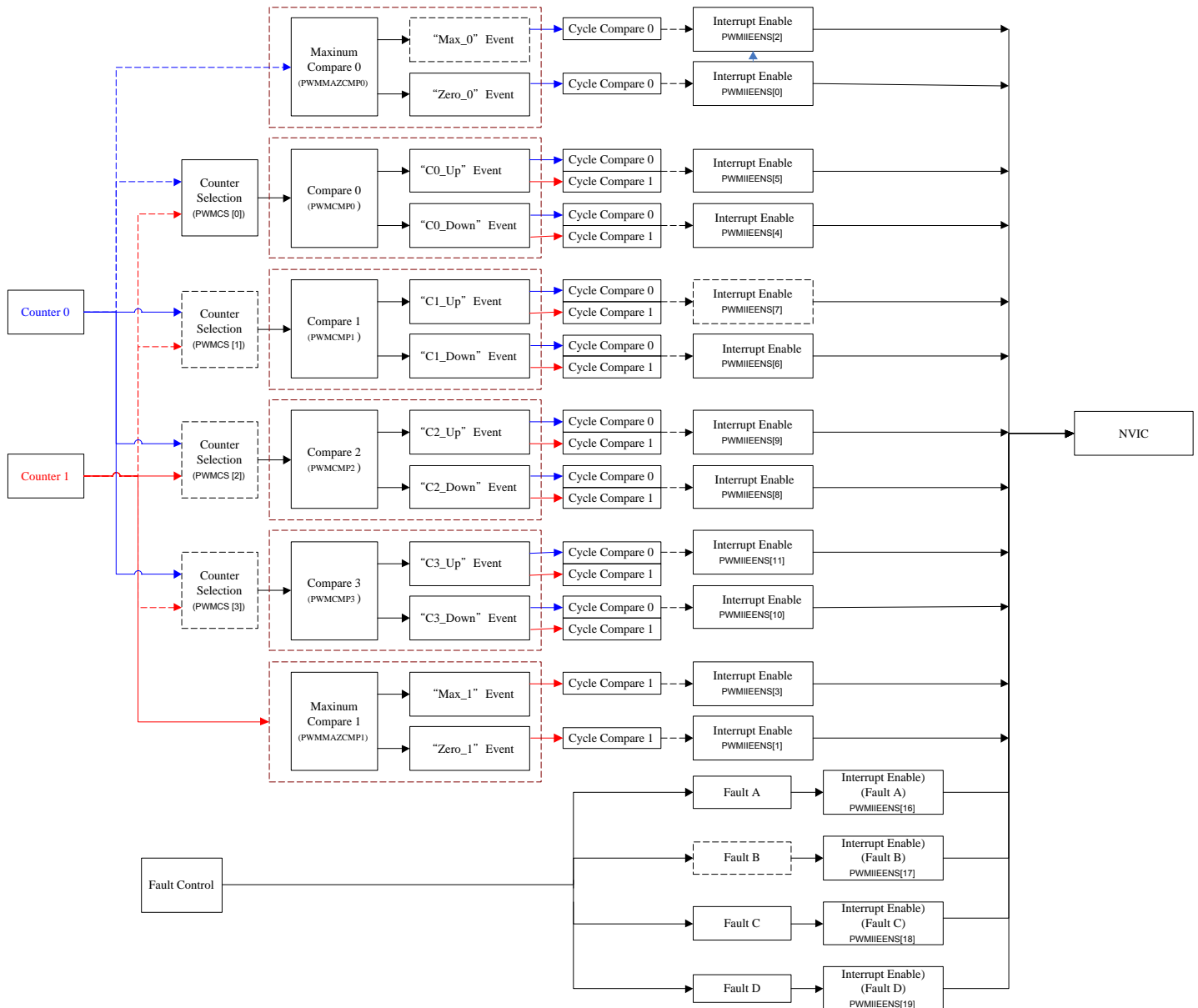


图 16.6.1 PWM 中断框图

### 16.8 PWM 周期比较

该 CYCMP0 / CYCMP1 寄存器可以将每个周期的“中断/更新”时间更改为 n 个周期。

1. CYCMP0 / CYCMP1 寄存器设置为 0 时，则每周期皆会进中断更新
2. CYCMP0 / CYCMP1 寄存器设置为 1 时，则每间隔 1 周期会进中断更新
3. CYCMP0 / CYCMP1 寄存器设置为 2 时，则每间隔 2 周期会进中断更新

以此类推，可参考 CYCMP0/1 - 0x5001 C080/84, CYCU0/1 - 0x5001 C088/8C 说明

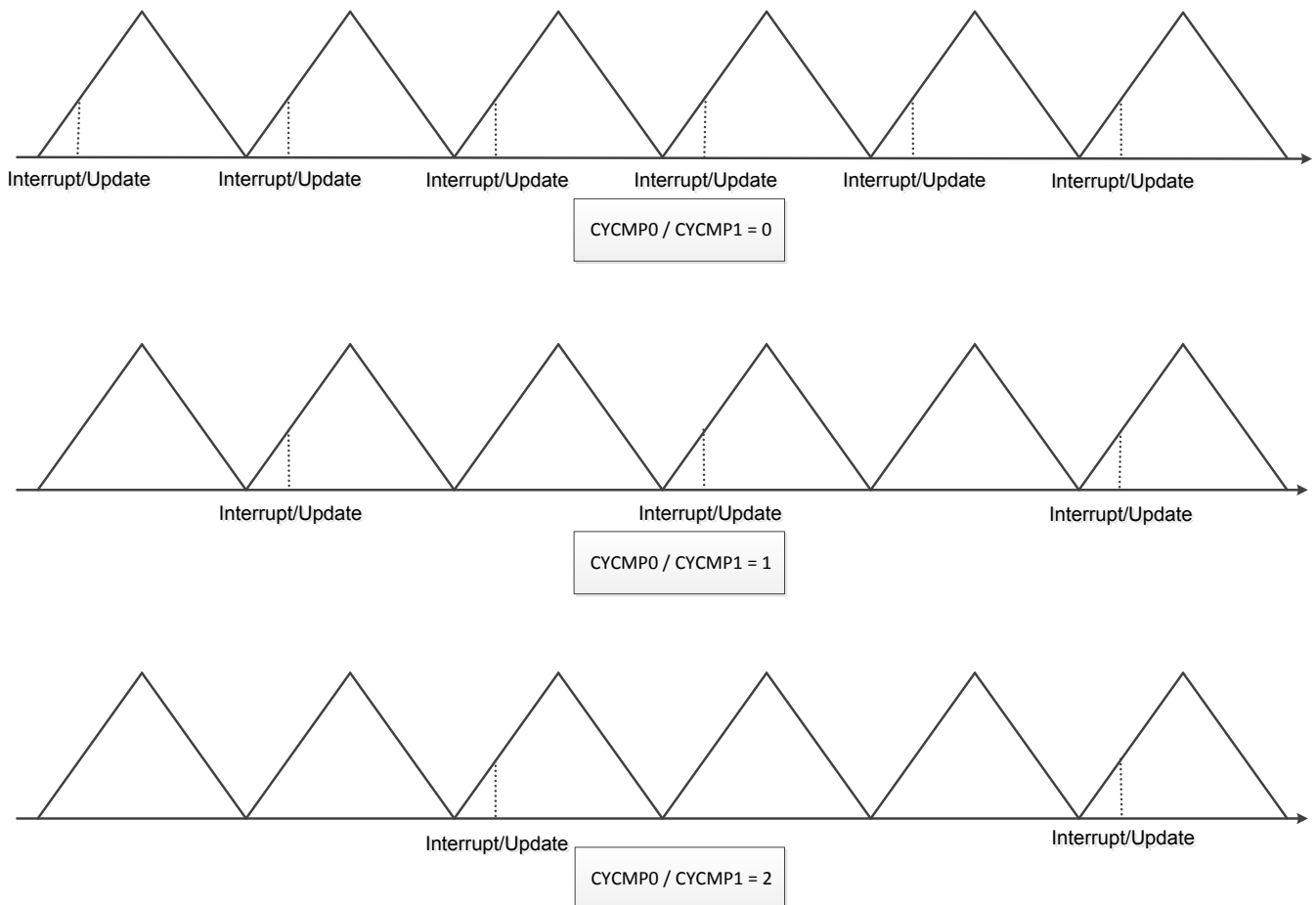


图 16.8.1 PWM 周期比较框图

### 16.9 PWM ADC 触发器

PWM 发生器也采用相同的 16 个计数器事件并使用它们生成 ADC 触发器。

可参考 ADCTDEL0U - 0x5001 C090, ADCTDEL0D - 0x5001 C108, DELCU0 - 0x5001 C098 说明

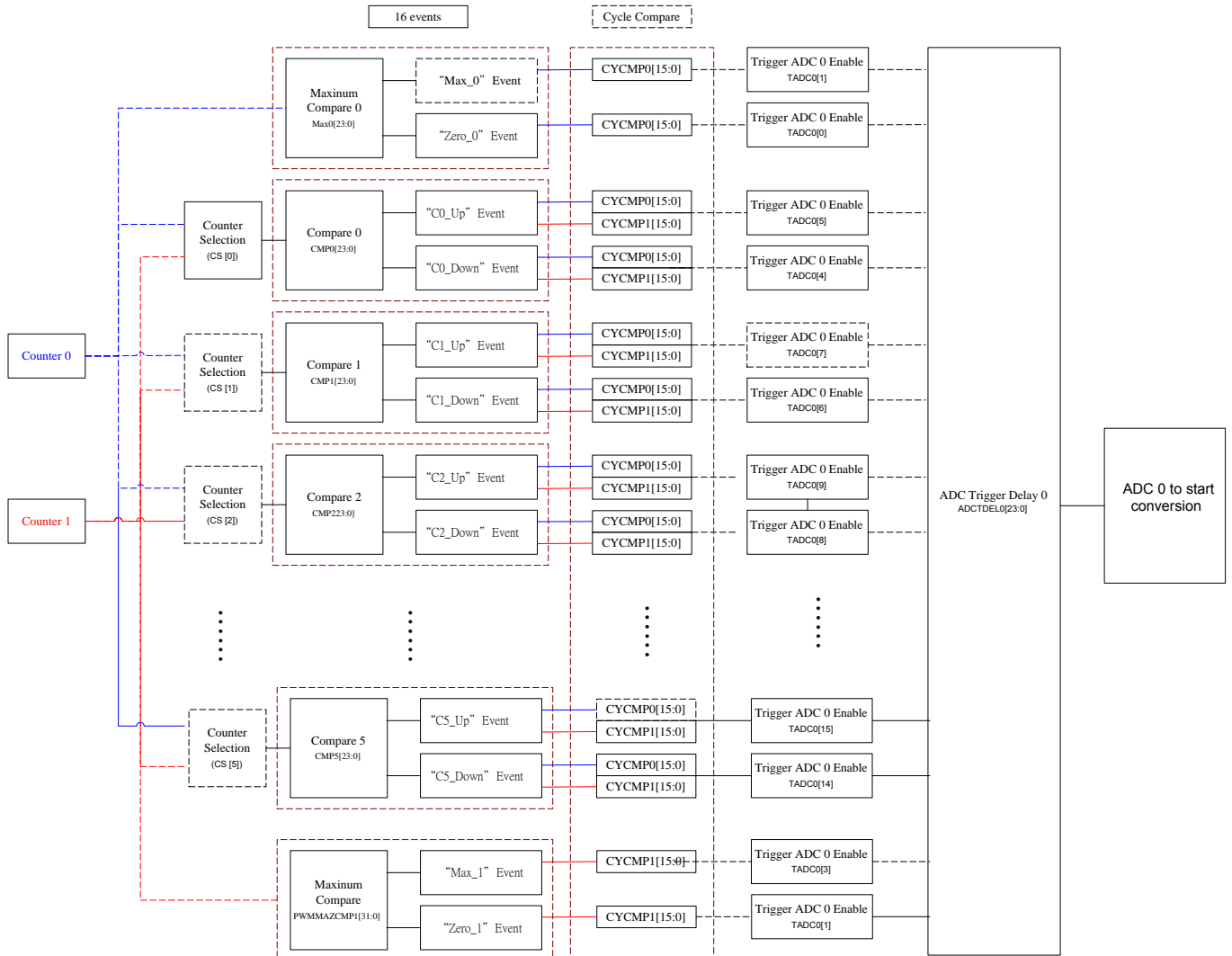


图 16.9.1 PWM ADC0 触发器框图

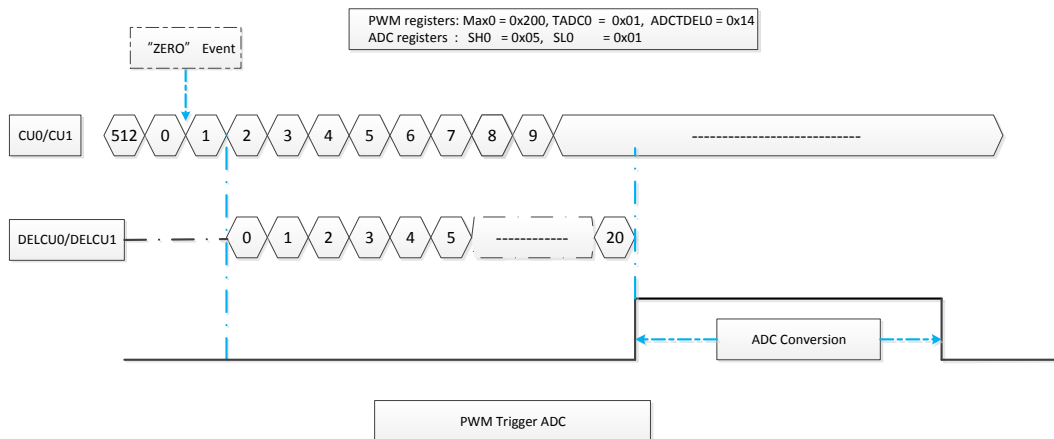


图16.9.2 PWM ADC触发时序框图

当发生任何选定的事件时，

1. 当计数器属于上数时，PWMCU0/1 将开始递增计数，直到值匹配 ADCTDEL0U。该 ADC 触发器将开始。
2. 当计数器属于下数时，PWMCU0/1 将开始递减计数，直到值匹配 ADCTDEL0D。该 ADC 触发器将开始。
3. ADC 的触发延时当前时间 可参考 DELCU0 寄存器

请参考 Table 17.16.46 ADCTDEL0D/ ADCTDEL0U 设置限制 说明

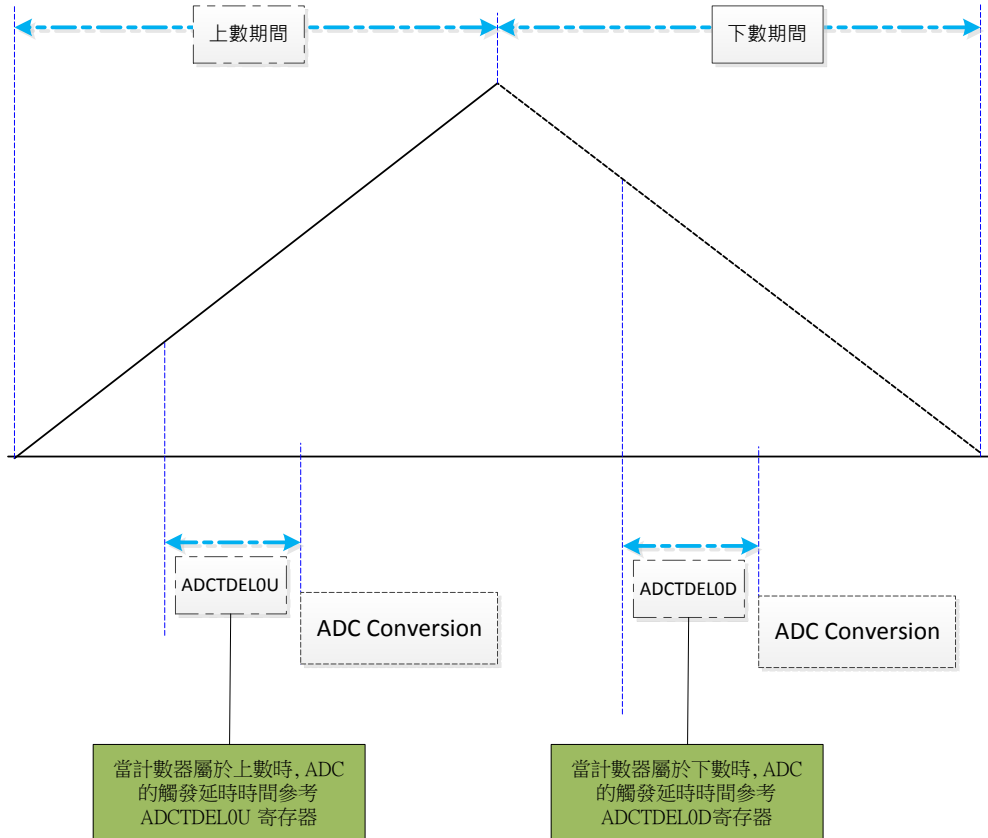
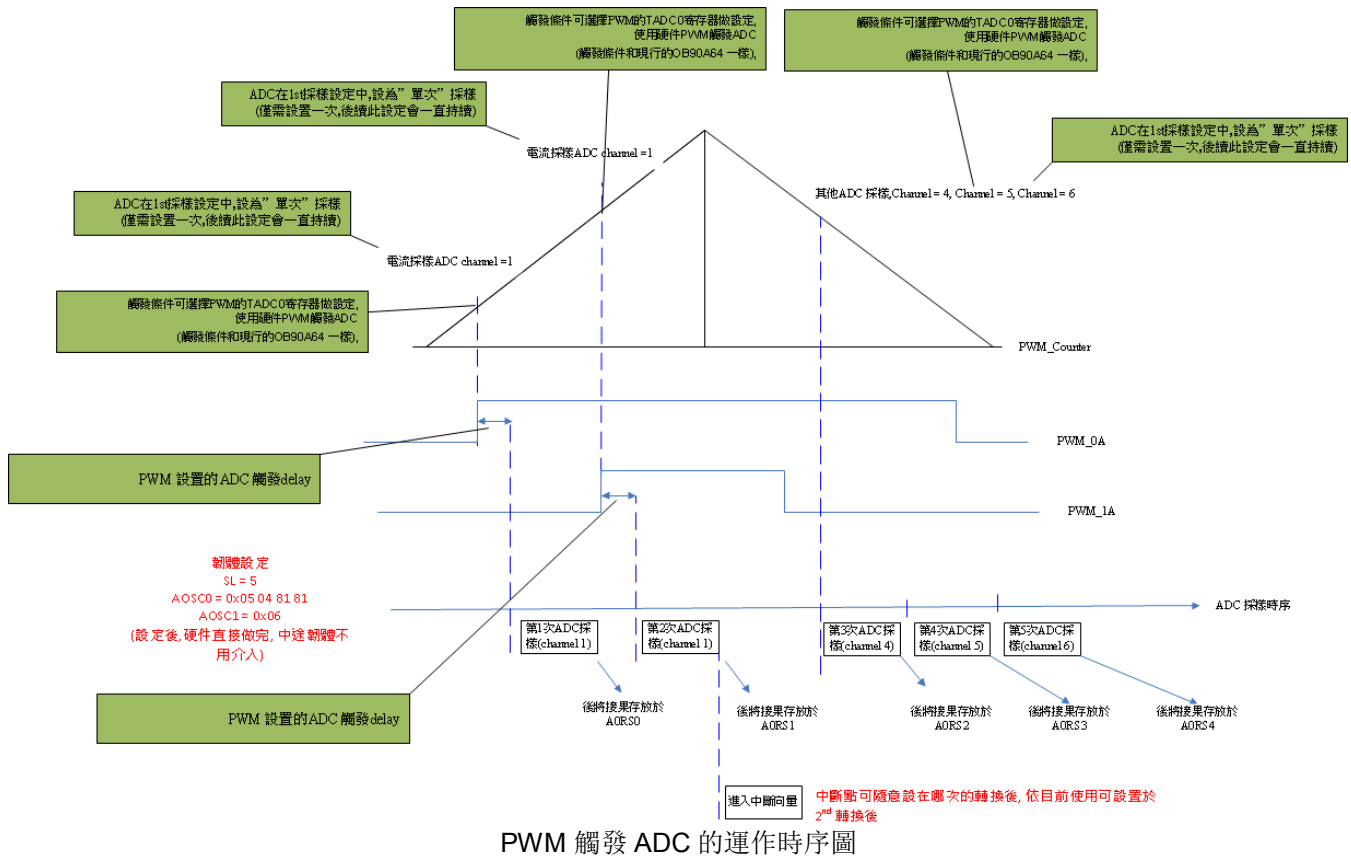
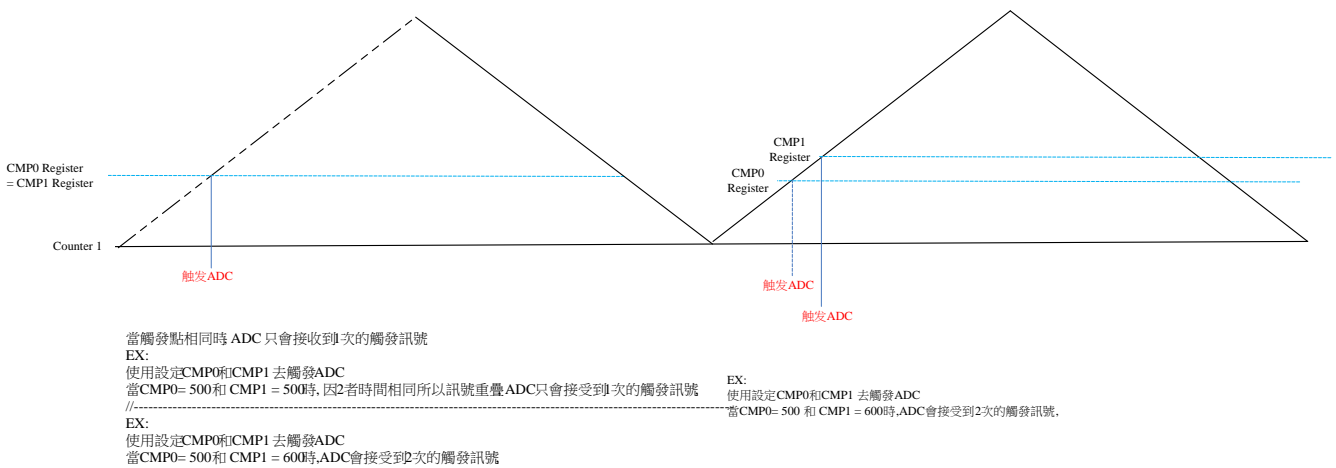


图 16.9.3 PWM 上/下数触发 ADC 方式框图

### 16.10 PWM 触发 ADC 的运作时序



### 16.11 PWM 触发 ADC 注意事项



### 16.12 PWM 死区

由 PWM 发生器产生的 6 个 PWM 信号被传送到死区发生器。如果禁用，则 PWM 信号未修改。如果使能，则第二个 PWM 信号将根据第一个输出 PWM 信号于上升沿延迟可编程量的死区信号。可参考 DB0/1/2 - 0x5001 C048/4C/50 说明

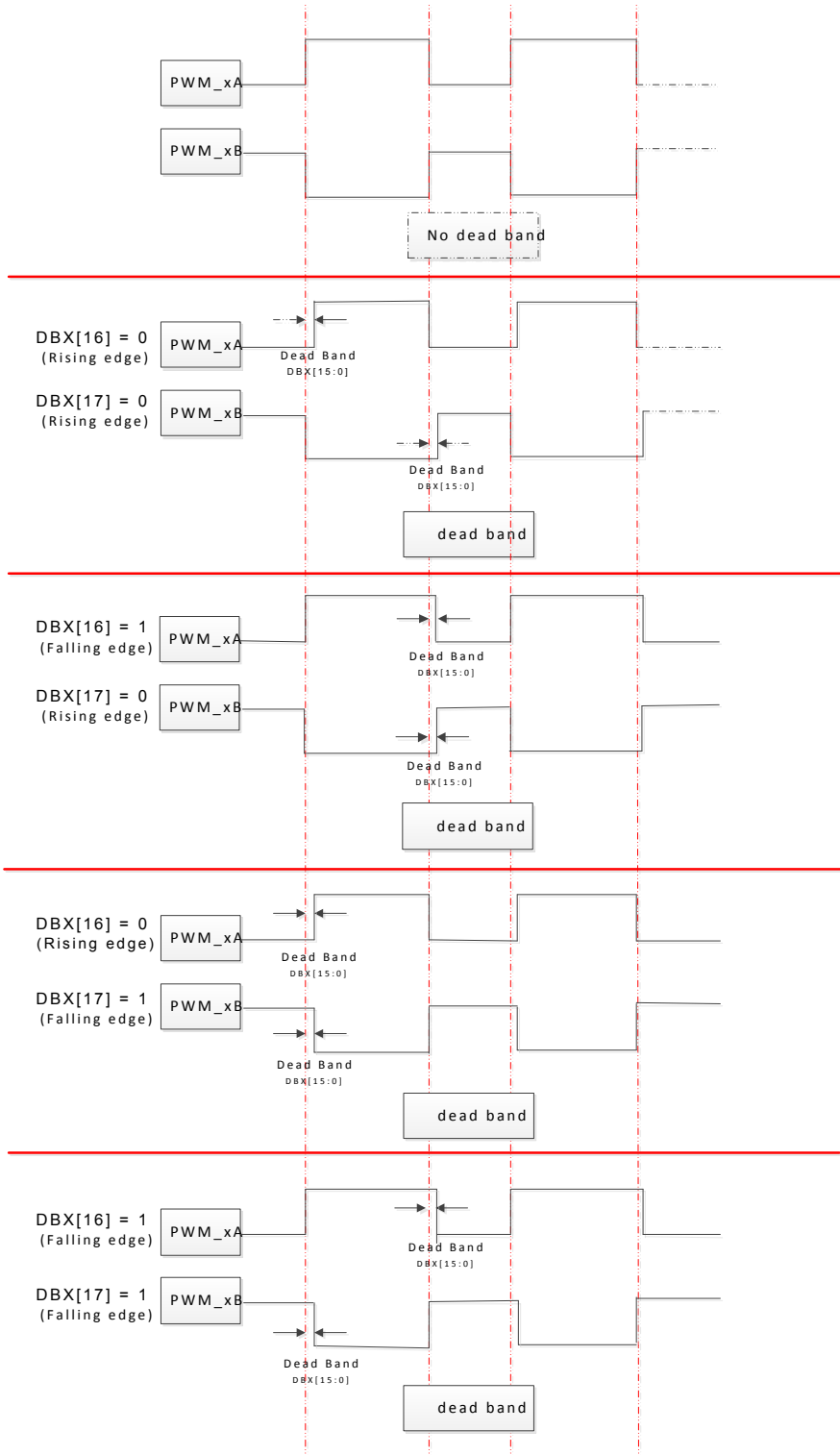
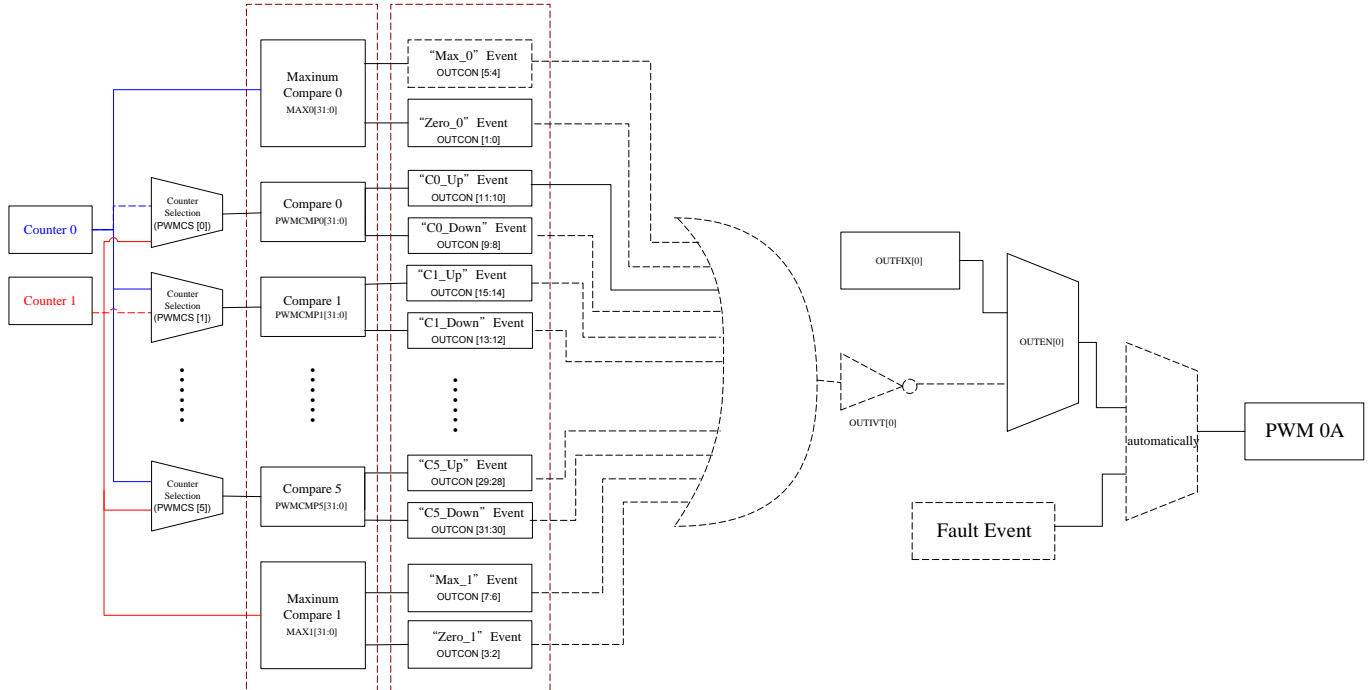


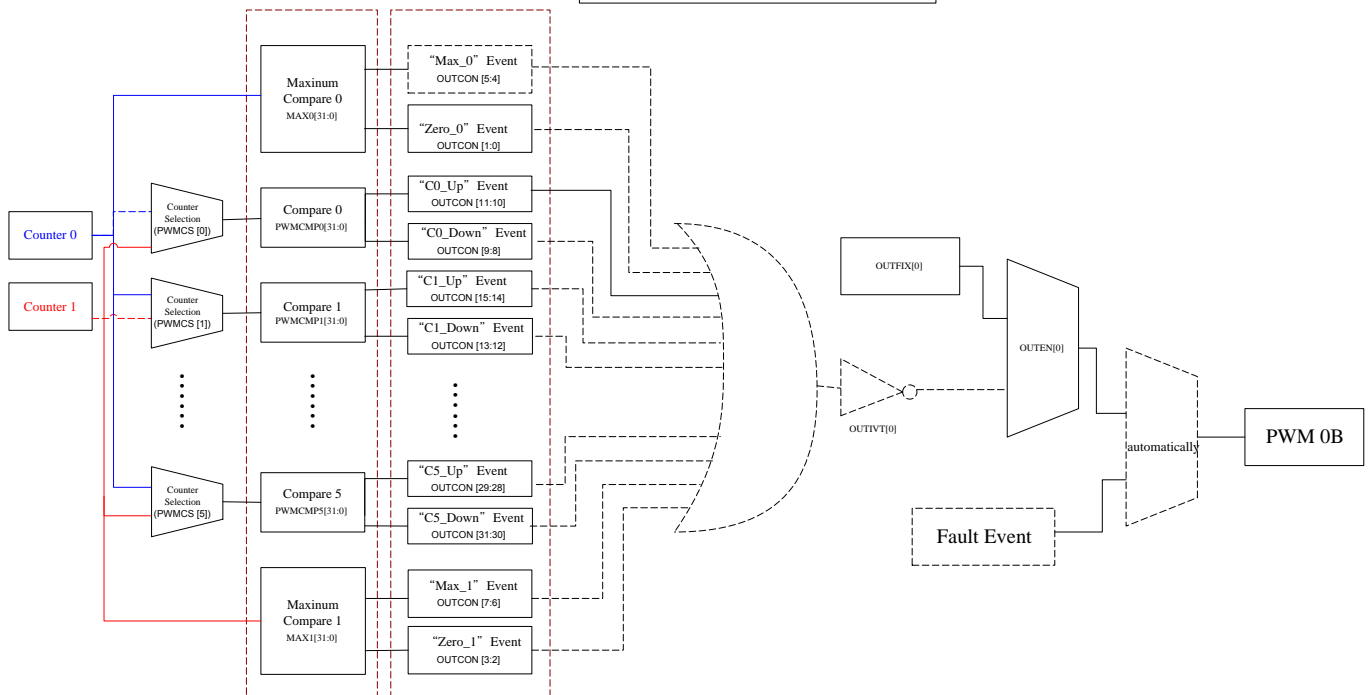
图 16.10.1 PWM 死区框图

### 16.13 PWM 输出

可参考 OUTEN - 0x5001 C01, OUTINI - 0x5001 C014, OUTCON0A - 0x5001 C018 ~ OUTCON2B - 0x5001 C02C, OUTFIX - 0x5001 C100, OUTIVT - 0x5001 C104 说明

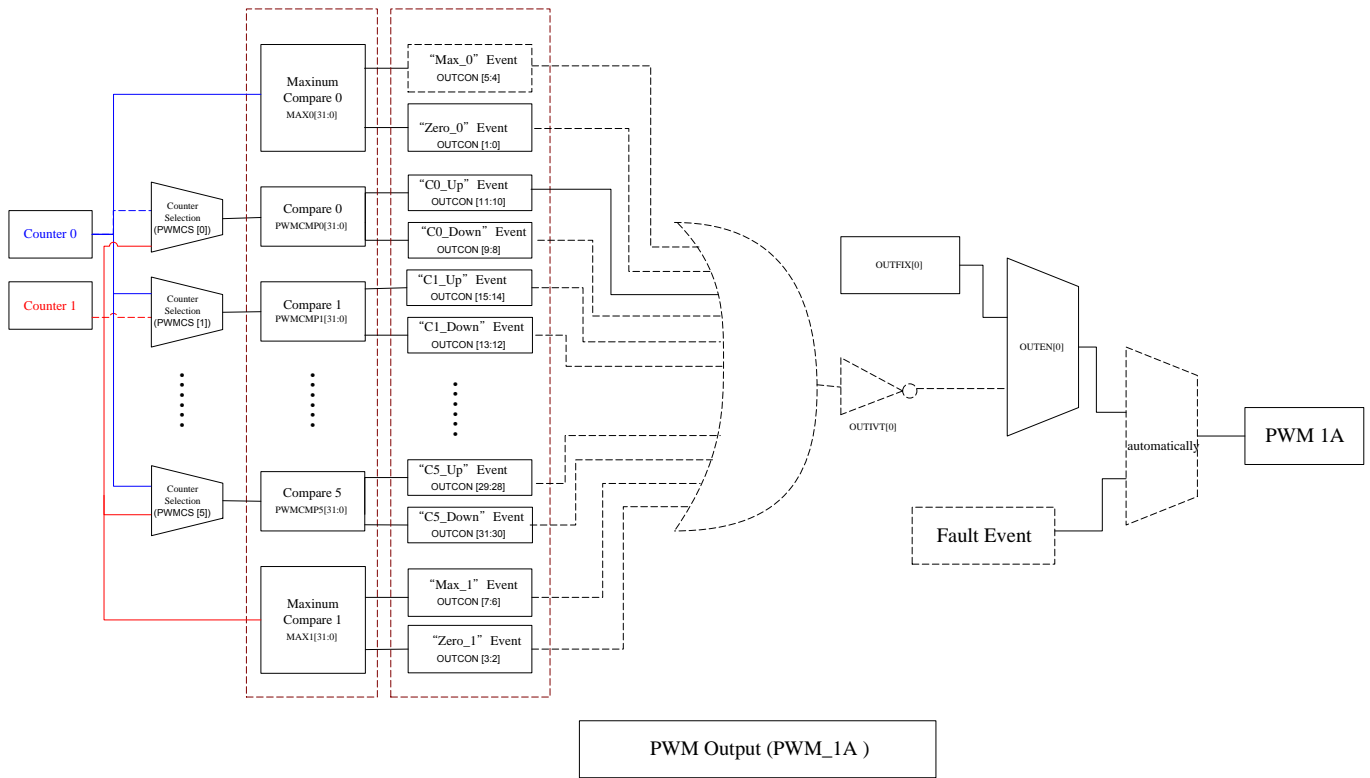


PWM Output (PWM\_0A)

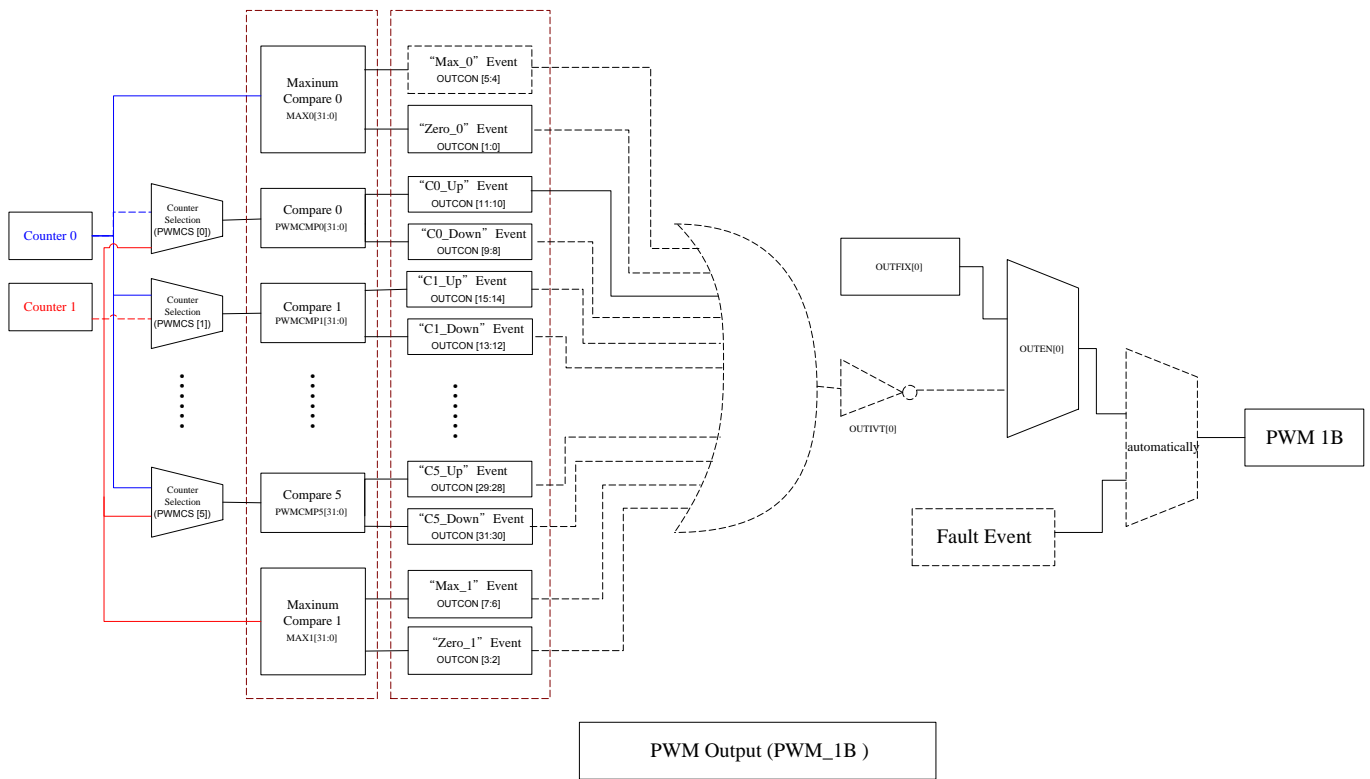


PWM Output (PWM\_0B)





PWM Output (PWM\_1A )



PWM Output (PWM\_1B )

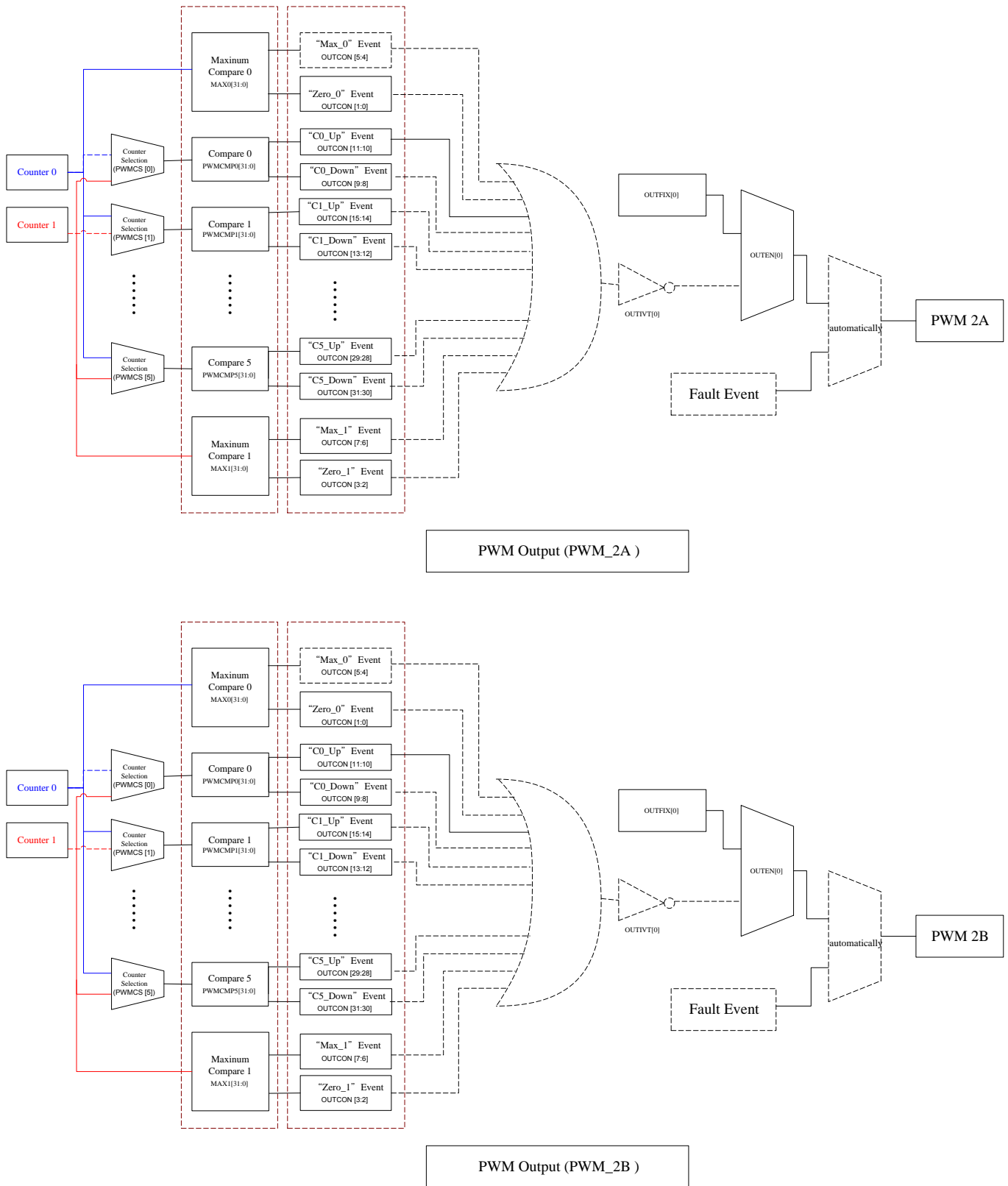


图 16.11.1 PWM 输出框图

### 16.14 PWM 故障输入

有四个 PWM 模块相关故障。故障输入引脚主要目的是禁用 PWM 输出信号并将驱动关闭状态。其直接在硬件中执行，可以快速管理故障输入，使 PWM 输出处于关闭状态。

PWM 故障输入为故障 A /故障 B /故障 C，其来源有自 I / O 引脚、CPU 或其它模块。故障 D 为内部检测

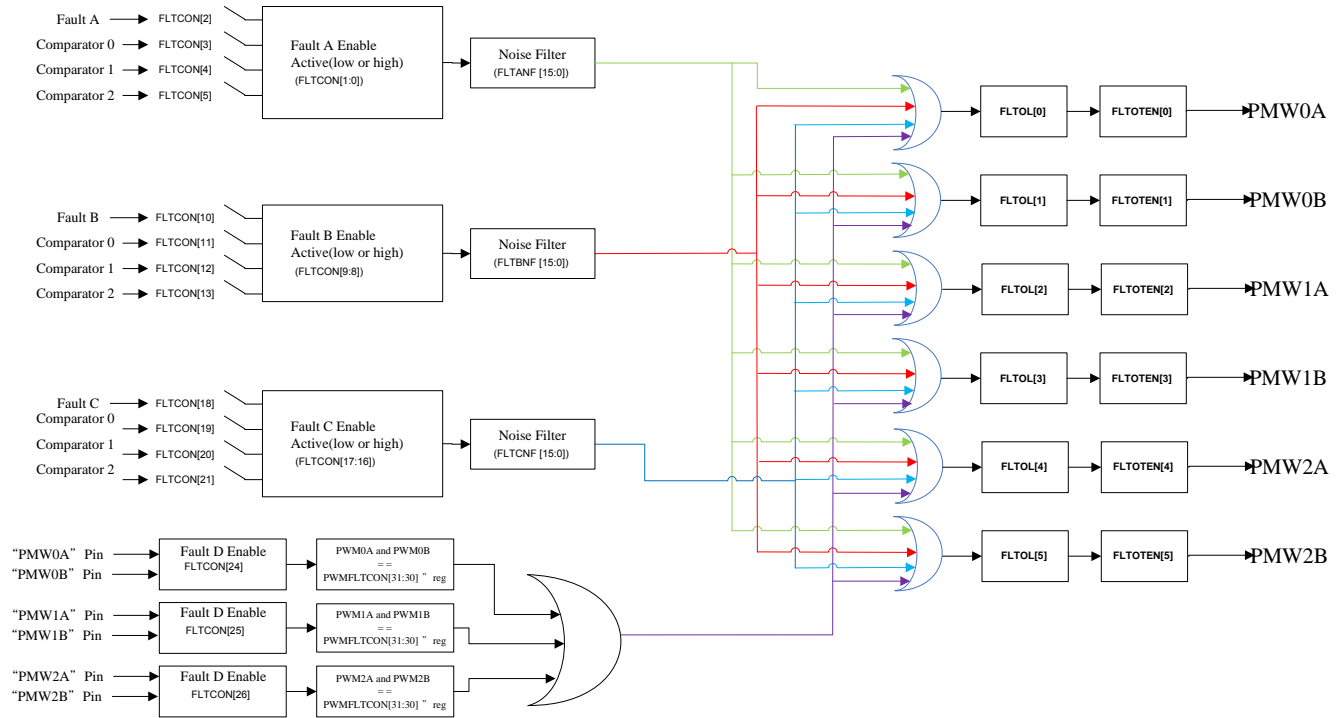


图 16.12.1 PWM 故障输入框图

### 16.12.1 故障引脚使能位

通过设置 FLTCON 寄存器，相应的故障输入被使能。如果两个位均被清零，则故障输入对 PWM 模块没有影响。可参考 FLTCON - 0x5001 C0B0 说明

### 16.12.2 故障状态时的 PWM 输出

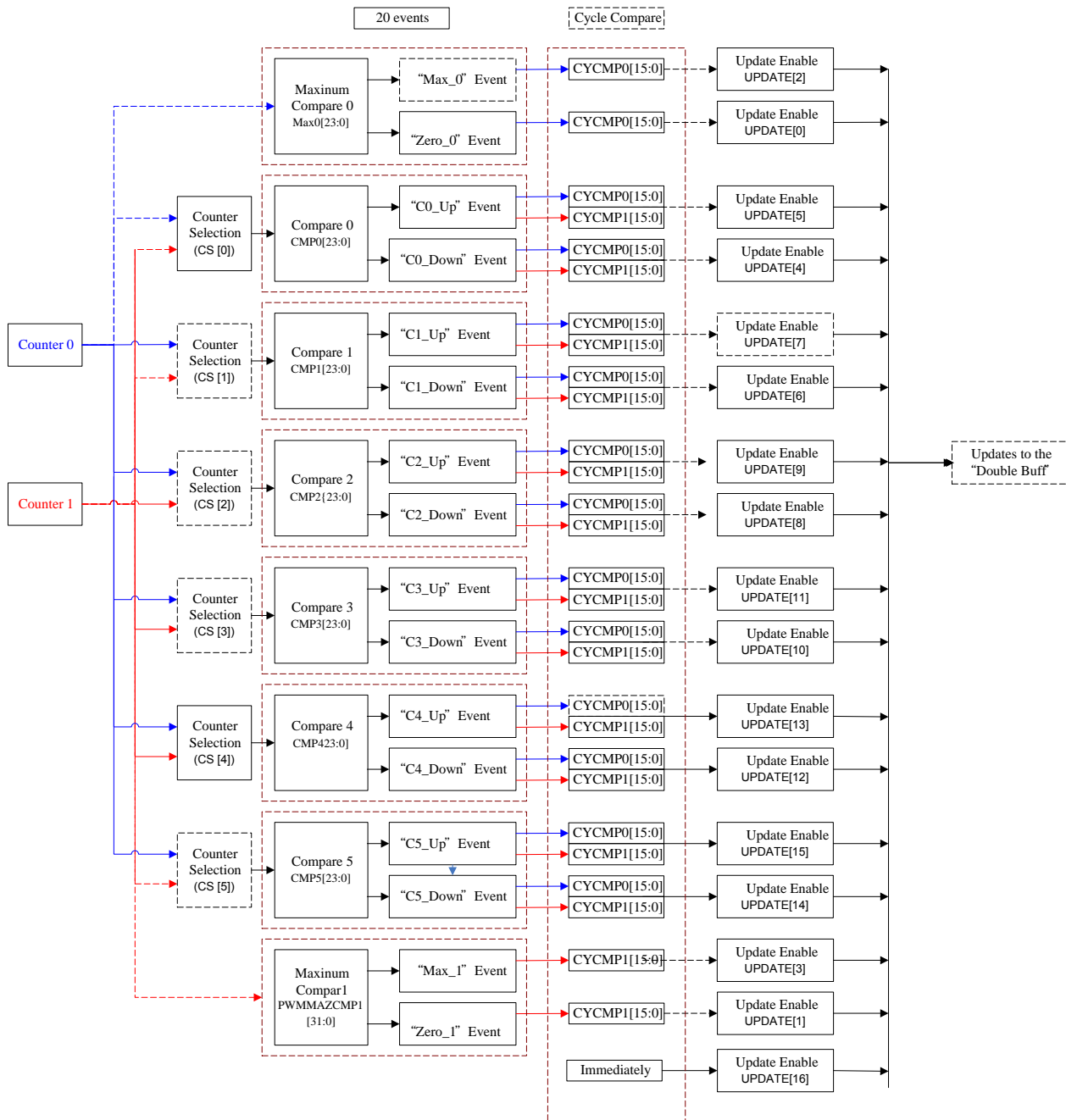
当处于故障状态（故障 A /故障 B /故障 C /故障 D 输入有效）时，PWM 输出信号被驱动到其关闭状态。通过 FLTOT 和 FLTOTEN 寄存器确定哪些 PWM 输出被禁用（处于故障状态时）：

可参考 FLTOTEN - 0x5001 C0B4, FLTOL - 0x5001 C0B8 说明

### 16.15PWM 更新

可以选择启用 PWM 更新功能，以使用户可以指定新值有效的的时间。PWM 更新功能通过设置 UPDATE 寄存器来使能。该位影响所有双缓冲寄存器，执行 PWM 更新功能：

1. 设置 UPDATE 寄存器。
  2. 写入所有双缓冲寄存器。
  3. 事件发生时，缓冲区值将被加载到实际的寄存器中。达到寄存器的同步加载。
- 可参考 UPDATE - 0x5001 C0D4 说明

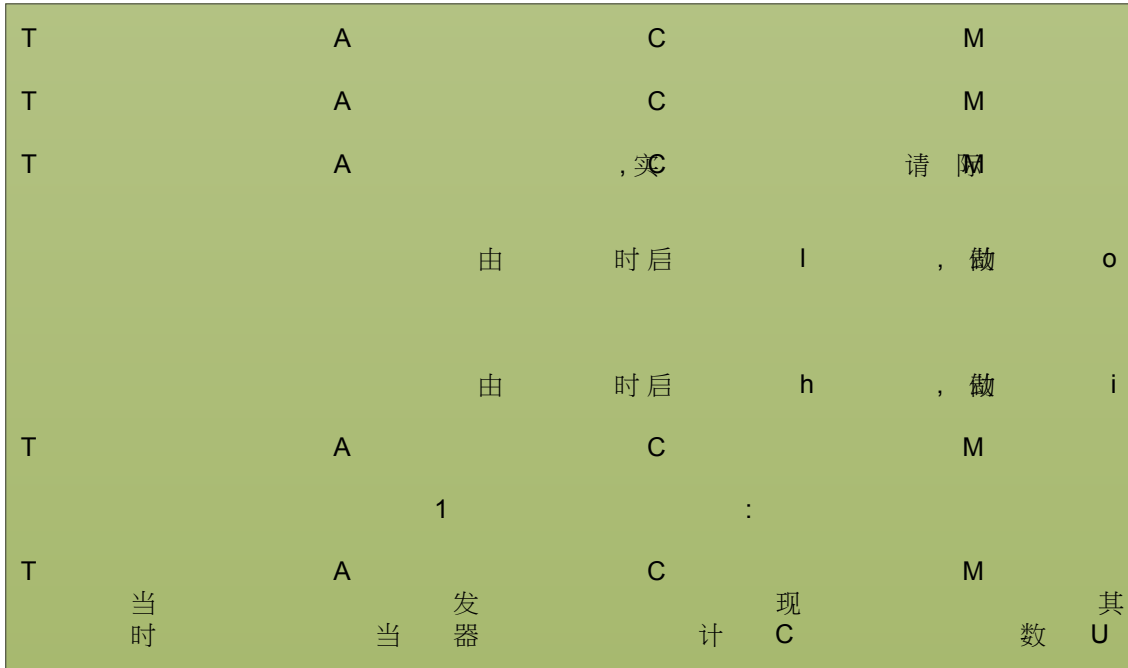


16.13.1 PWM 更新框图

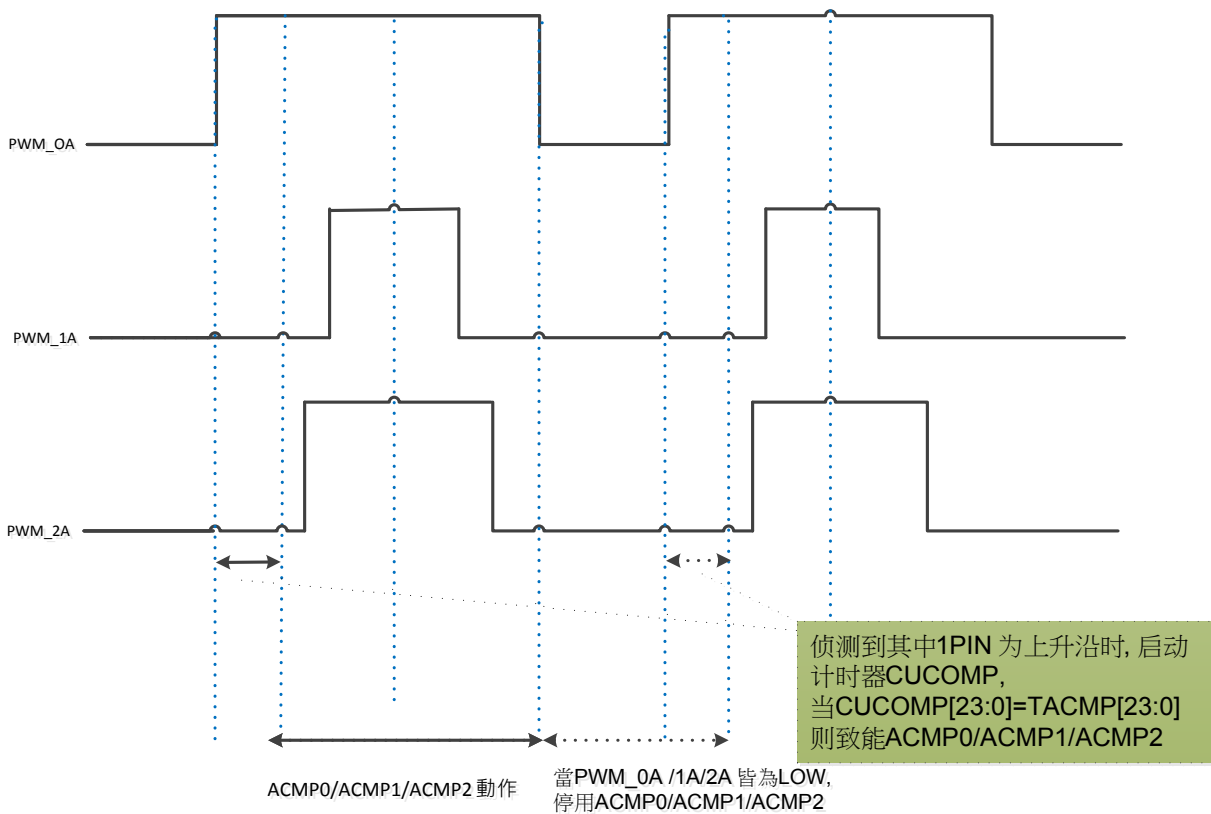
图

**16.16 使用 PWM 触发模拟比较器方式**

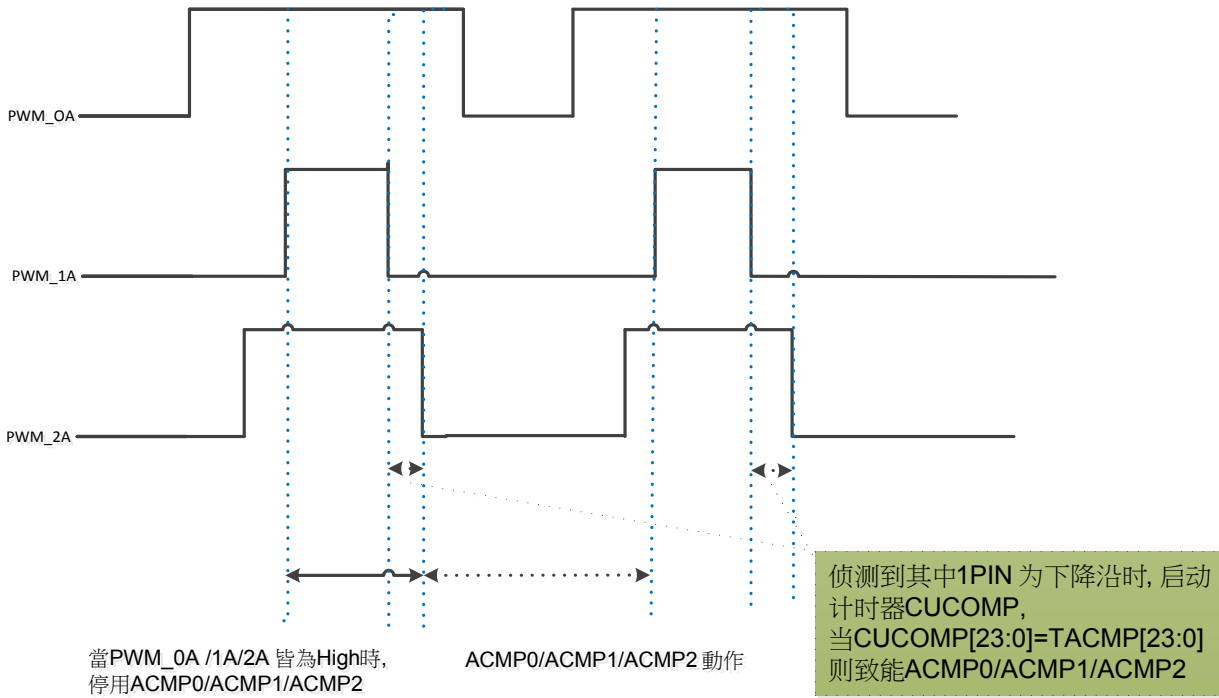
ACMP 的动作方式仍然依照 ACMP 的寄存器设定，此功能仅提供 PWM 触发控制 ACMP 的动作时间



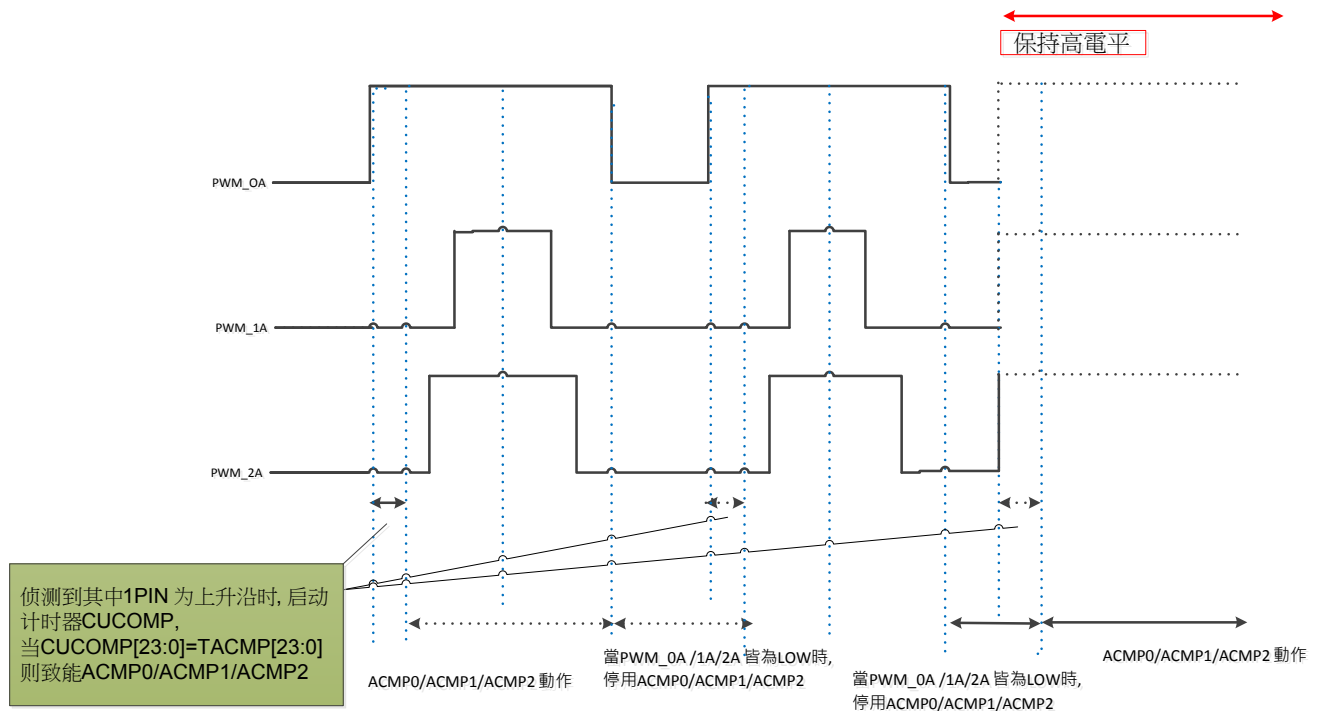
第一种: TACMP[30] = 0 (High Level)



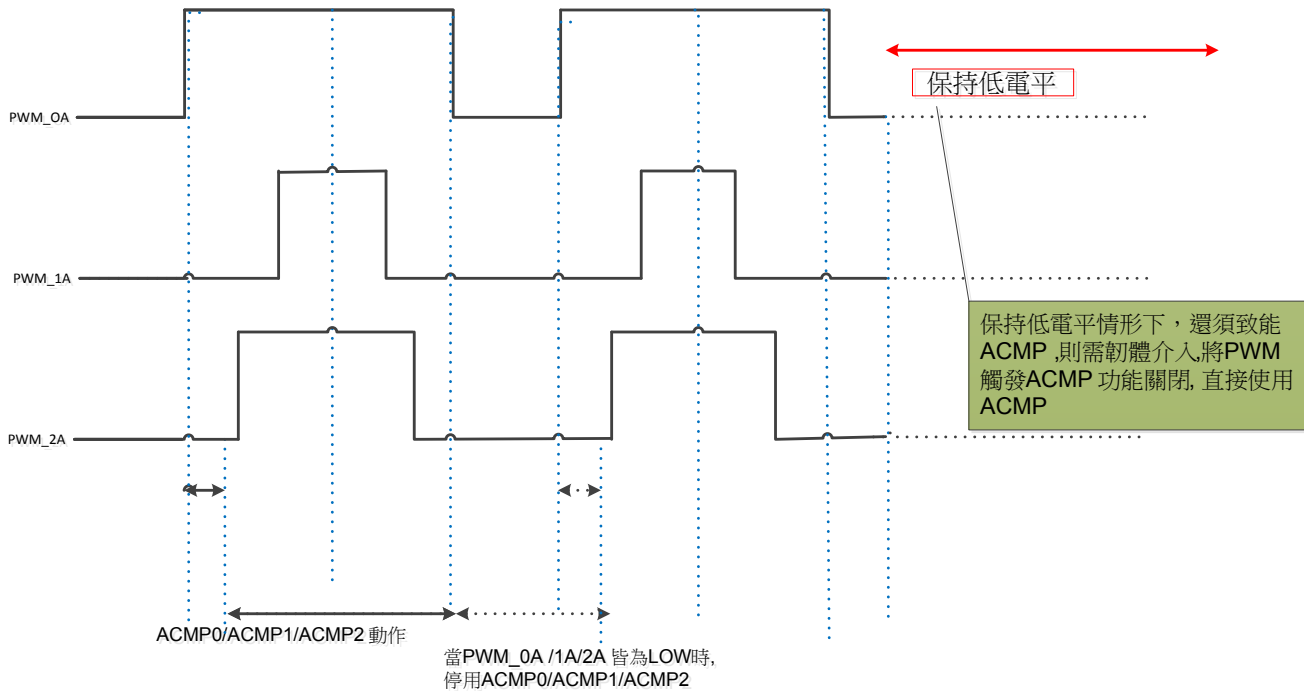
第二种: TACMP[30] = 1 (Low Level)



第三种: TACMP[30] = 0 (High Level) &amp; 下一周期为保持高电平



第四种: TACMP[30] = 1 (Low Level) & 下一周期为保持低电平



### 16.17 寄存器描述

表 1 功率控制 PWM 模块 (基址 0x5001 C000)

名称	访问类型	地址偏移	描述	重置值
PWMCON	R/W	0x00	PWM控制寄存器	0
MAX0	R/W	0x04	PWM最大比较0寄存器	0
MAX1	R/W	0x08	PWM最大比较1寄存器	0
PWMCS	R/W	0x0C	PWM计数器选择	0
OUTEN	R/W	0x10	PWM输出使能寄存器	0
OUTINI	R/W	0x14	PWM输出初始寄存器	0
OUTCON0A	R/W	0x18	PWM 0A输出控制寄存器	0
OUTCON0B	R/W	0x1C	PWM 0B 输出控制寄存器	0
OUTCON1A	R/W	0x20	PWM 1A输出控制寄存器	0
OUTCON1B	R/W	0x24	PWM 1B 输出控制寄存器	0
OUTCON2A	R/W	0x28	PWM 2A输出控制寄存器	0
OUTCON2B	R/W	0x2C	PWM 2B 输出控制寄存器	0
-	-	0x30-0x44	保留	0
DB0	R/W	0x48	死区0寄存器	0
DB1	R/W	0x4C	死区1寄存器	0

DB2	R/W	0x50	死区2寄存器	0
-	-	0x54-0x5C	保留	0
PWMCMP0	R/W	0x60	PWM比较0寄存器	0
PWMCMP1	R/W	0x64	PWM比较1寄存器	0
PWMCMP2	R/W	0x68	PWM比较2寄存器	0
PWMCMP3	R/W	0x6C	PWM比较3寄存器	0
PWMCMP4	R/W	0x70	PWM比较4寄存器	0
PWMCMP5	R/W	0x74	PWM比较5寄存器	0
PWMCU0	RO	0x78	当前计数器0	0
PWMCU1	RO	0x7C	当前计数器1	0
CYCMP0	R/W	0x80	PWM周期比较0寄存器	0
CYCMP1	R/W	0x84	PWM周期比较1寄存器	0
CYCU0	RO	0x88	PWM周期计数器0寄存器	0
CYCU1	RO	0x8C	PWM周期计数器1寄存器	0
ADCTDEL0	R/W	0x90	PWM ADC触发延迟0寄存器	0
-	-	0x94	保留	0
DELCU0	RO	0x98	PWM延迟计数器0寄存器	0
-	-	0x9C	保留	0
IEENS	R/W	0xA0	PWM中断使能设置寄存器	0
IEENC	R/W	0xA4	PWM中断使能清除寄存器	0
EVST	RO	0xA8	PWM事件状态寄存器	0
EVSTC	R/W	0xAC	PWM事件状态清除寄存器	0
FLTCON	R/W	0xB0	故障配置寄存器	0
FLTOTEN	R/W	0xB4	故障输出使能寄存器	0
FLTOL	R/W	0xB8	故障输出电平寄存器	0
FLTNFA	R/W	0xBC	PWM故障A噪声滤波器寄存器	0
FLTNFB	R/W	0xC0	PWM故障B噪声滤波器寄存器	0
FLTNFC	R/W	0xC4	PWM故障C噪声滤波器寄存器	0
STA	RO	0xC8	PWM状态寄存器	0
TADC0	R/W	0xCC	PWM触发ADC 0使能寄存器	0
-	-	0xD0	保留	0
UPDATE	R/W	0xD4	PWM更新模式使能寄存器	0
FLTREL	R/W	0xD8	PWM 故障释放寄存器	0
FLTST	R/W	0xDC	PWM 故障状态寄存器	0
ICE	R/W	0xE0	PWM ICE 模式寄存器	0
CUINIT0	RW	0xE4	PWM 计数器 0 初始寄存器	0
CUINIT1	RW	0xE8	PWM 计数器 1 初始寄存器	0



TACMP	RW	0xEC	PWM 比较器触发控制	0
-	-	0xF0-0xFC	保留	0
OUTFIX	R/W	0x100	PWM 输出 FIX 寄存器	0
OUTIVT	R/W	0x104	PWM 输出反相寄存器	0
ADCTDELD0D	R/W	0x108	PWM ADC 触发延迟 0D 寄存器 (倒数)	0

**Note**
**1: 有双重缓冲区**

MAX0, MAX1, PWMCMP0, PWMCMP1, PWMCMP2, PWMCMP3, PWMCMP4, PWMCMP5,  
 OUTCON0A, OUTCON0B,  
 OUTCON1A, OUTCON1B,  
 OUTCON2A, OUTCON2B,

**2: 读双缓冲寄存器**

MAX0, MAX1, PWMCMP0, PWMCMP1, PWMCMP2, PWMCMP3, PWMCMP4, PWMCMP5,  
 OUTCON0A, OUTCON0B,  
 OUTCON1A, OUTCON1B,  
 OUTCON2A, OUTCON2B,  
 缓冲区的内容值将被读出, 但非寄存器的内容。

**16.15.1 PWM 控制寄存器(PWMCON – 0x5001 C000)**
**表 2 PWM 控制寄存器(PWMCON - 0x5001 C000)**

位	符号	内容	描述	重置值
0	EN0		PWM 计数器 0(PWMCU0)使能位	0
		0	禁用 (计数器 0 停止计数, CUINT0 寄存器的值复制到 PWMCU0 寄存器当初使值)	
		1	致能	
2:1	MODE0		PWM 计数器 0(PWMCU0)有效模式选择位	0
		00	PWM 计数器 0 以自由运行模式运行。	
		01	PWM 计数器 0 在单次模式下工作。	
		10	PWM 计数器 0 以连续递增/递减计数模式运行。	
		11	保留	
3	RSTCU0		PWM 计数器 0(PWMCU0)重置位	0
		0	闲置	
		1	重置计数器 0. CUINT0 寄存器的值复制到 PWMCU0 寄存器当初使值。	
4	RSTCYC0		PWM 周期计数器 0 重置位	0
		0	闲置	
		1	重置循环计数器 0( CYCU0 寄存器).清除后硬件自动清零	
5	RSTDEL0		PWM 延迟计数器 0 重置位	0
		0	闲置	
		1	重置延迟计数器 0. ( DELCU0 寄存器).清除后硬件自动清零	
7:6	-	-	保留	0
8	EN1		PWM 计数器 1(PWMCU1)使能位	0
		0	禁用 (计数器 1 停止计数, CUINT1 寄存器的值复制到 PWMCU1 寄存器当初使值)	
		1	致能	
10:9	MODE1		PWM 计数器 1(PWMCU1)有效的模式选择位	0
		00	PWM 计数器 1 以自由运行模式运行。	
		01	PWM 计数器 1 在单次模式下工作。	
		10	PWM 计数器 1 以连续递增/递减计数模式运行。	

	11	保留	
11	RSTCU1	PWM 计数器 1(PWMCU1)重置位	0
		闲置	
		重置计数器 1。 CUINT1 寄存器的值复制到 PWMCU1 寄存器当初使值	
12	RSTCYC1	PWM 周期计数器 1 重置位	0
		闲置	
		重置循环计数器 1. (CYCU1 寄存器).清除后硬件自动清零	
13	RSTDEL1	PWM 延迟计数器 1 重置位	0
		闲置	
		重置延迟计数器 1。清除延迟计数器 1 后自动清除	
31:14	-	保留	0

### 16.15.2 PWM 最大比较寄存器(MAX0/1 – 0x5001 C004/08)

当计数器 0/1 等于此值时, 会做递减/归零/停止等动作(依据 PWMCON[2:1]/ PWMCON[10:9] 的设定)

表 17.13.2.1 PWM 最大比较寄存器(MAX0/1 - 0x5001 C004/08)

位	符号	内容	描述	重置值
23:0	MAX	-	PWM 最大比较寄存器	FFFFFF
31:24			保留	0

### 16.15.3 PWM 计数器选择寄存器(PWMCS – 0x5001 C00C)

表 17.13.3.1 PWM 计数器选择寄存器(PWMCS - 0x5001 C00C)

位	符号	内容	描述	重置值
0	CMP0	-	选择 PWMCMP0 寄存器的计数来源	0
		0	计数器 0(PWMCU0)	
		1	计数器 1(PWMCU1)	
1	CMP1	-	选择 PWMCMP1 寄存器的计数来源	0
		0	计数器 0(PWMCU0)	
		1	计数器 1(PWMCU1)	
2	CMP2	-	选择 PWMCMP2 寄存器的计数来源	0
		0	计数器 0(PWMCU0)	
		1	计数器 1(PWMCU1)	
3	CMP3	-	选择 PWMCMP3 寄存器的计数来源	0
		0	计数器 0(PWMCU0)	
		1	计数器 1(PWMCU1)	
4	CMP4	-	选择 PWMCMP4 寄存器的计数来源	0
		0	计数器 0(PWMCU0)	
		1	计数器 1(PWMCU1)	
5	CMP5	-	选择 PWMCMP5 寄存器的计数来源	0
		0	计数器 0(PWMCU0)	
		1	计数器 1(PWMCU1)	
31:6	-	-	保留	0

### 16.15.4 PWM 输出使能寄存器(OUTEN – 0x5001 C010)

当设定为"禁用"时, 输出电位请参考 OUTFIX 寄存器

当设定为"致能"时, 初始的输出电位请参考 OUTINI 寄存器, 后续变化请参考 OUTCON0X 寄存器

表 17.13.4.1 PWM 输出使能寄存器(OUTEN - 0x5001 C010)

位	符号	内容	描述	重置值
0	PWM0A		PWM0A 输出使能	0
		0	禁用	

		1	致能	
1	PWM0B		PWM0B 输出使能	0
		0	禁用	
		1	致能	
2	PWM1A		PWM1A 输出使能	0
		0	禁用	
		1	致能	
3	PWM1B		PWM1B 输出使能	0
		0	禁用	
		1	致能	
4	PWM2A		PWM2A 输出使能	0
		0	禁用	
		1	致能	
5	PWM2B		PWM2B 输出使能	0
		0	禁用	
		1	致能	
31:6	-	-	保留	0

#### 16.15.5 PWM 输出初始寄存器(OUTINI – 0x5001 C014)

设定 PWM 引脚在初始时的电平(当 OUTEN 寄存器为 1 时, 此寄存器才有效)

表 17.13.5.1 PWM 输出初始寄存器(OUTINI - 0x5001 C014)

位	符号	内容	描述	重置值
0	PWM0A		PWM0A 输出初始	0
		0	强制 PWM0A 输出低电平	
		1	强制 PWM0A 输出高电平	
1	PWM0B		PWM0B 输出初始	0
		0	强制 PWM0B 输出低电平	
		1	强制 PWM0B 输出高电平	
2	PWM1A		PWM1A 输出初始	0
		0	强制 PWM1A 输出低电平	
		1	强制 PWM1A 输出高电平	
3	PWM1B		PWM1B 输出初始	0
		0	强制 PWM1B 输出低电平	
		1	强制 PWM1B 输出高电平	
4	PWM2A		PWM2A 输出初始	0
		0	强制 PWM2A 输出低电平	
		1	强制 PWM2A 输出高电平	
5	PWM2B		PWM2B 输出初始	0
		0	强制 PWM2B 输出低电平	
		1	强制 PWM2B 输出高电平	
31:6	-	-	保留	0

#### 16.15.6 PWM 0A 输出控制寄存器(OUTCON0A – 0x5001 C018)

设定 PWM 0A 脚位输出的时间点及脚位状态,当 OUTEN[0] = 1 时, 此寄存器才有效

表 17.13.6.1 PWM 0A 输出控制寄存器(OUTCON0A - 0x5001 C018)

位	符号	内容	描述	重置值
1:0	ZERO0		计数器 0 为零时	0
		00	禁用	
		01	强制 PWM0A 输出低电平	
		10	强制 PWM0A 输出高电平	
		11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	

3:2 ZERO1		计数器 1 为零时	0
	00	禁用	
	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
5:4 MAX0		计数器 0 与寄存器 MAX0 匹配时	0
	00	禁用	
	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
7:6 MAX1		当计数器 1 匹配寄存器 MAX1 时	0
	00	禁用	
	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
9:8 DOWNC0		当计数器匹配比较器 0 时倒计时。 (当 PWMCMP0 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
11:10 UPC0		当计数器匹配比较器 0 时加计数。 (当 PWMCMP0 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
13:12 DOWNC1		当计数器与比较器 1 匹配时, 倒计时。 (当 PWMCMP1 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
15:14 UPC1		当计数器匹配比较器 1 时, 计数。 (当 PWMCMP1 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
17:16 DOWNC2		当计数器与比较器 2 匹配时倒计时。 (当 PWMCMP2 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
19:18 UPC2		当计数器匹配比较器 2 时加计数。 (当 PWMCMP2 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
21:20 DOWNC3		当计数器与比较器 3 匹配时倒计时。 (当 PWMCMP3 寄存器=0 时, 不支持此项功能)	0
	00	禁用	

	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
23:22 UPC3		当计数器匹配比较器 3 时加计数。 (当 PWMCMP3 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
25:24 DOWNC4		当计数器与比较器 4 匹配时倒计时。 (当 PWMCMP4 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
27:26 UPC4		当计数器匹配比较器 4 时加计数。 (当 PWMCMP4 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
29:28 DOWNC5		当计数器与比较器 5 匹配时倒计时。 (当 PWMCMP5 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
31:30 UPC5		当计数器匹配比较器 5 时加计数。 (当 PWMCMP5 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM0A 输出低电平	
	10	强制 PWM0A 输出高电平	
	11	切换 PWM0A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	

### 16.15.7 PWM 0B 输出控制寄存器(OUTCON0B – 0x5001 C01C)

设定 PWM 0B 脚位输出的时间点及脚位状态,当 OUTEN[1] = 1 时, 此寄存器才有效

表 17.13.7.1 PWM 0B 输出控制寄存器(OUTCON0B - 0x5001 C01C)

位	符号	内容	描述	重置值
1:0	ZERO0		计数器 0 为零时	0
		00	禁用	
		01	强制 PWM0B 输出低电平	
		10	强制 PWM0B 输出高电平	
		11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平	
3:2	ZERO1		计数器 1 为零时	0
		00	禁用	
		01	强制 PWM0B 输出低电平	
		10	强制 PWM0B 输出高电平	
		11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平	
5:4	MAX0		计数器 0 与寄存器 MAX0 匹配时	0
		00	禁用	
		01	强制 PWM0B 输出低电平	
		10	强制 PWM0B 输出高电平	

7:6 MAX1	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平 当计数器 1 匹配寄存器 MAX1 时	0
	00	禁用	
	01	强制 PWM0B 输出低电平	
	10	强制 PWM0B 输出高电平	
	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平 当计数器匹配比较器 0 时倒计时 (当 PWMCMP0 寄存器=0 时, 不支持此项功能)。	0
9:8 DOWNC0	00	禁用	
	01	强制 PWM0B 输出低电平	
	10	强制 PWM0B 输出高电平	
	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平 当计数器匹配比较器 0 时加计数。 (当 PWMCMP0 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
11:10 UPC0	01	强制 PWM0B 输出低电平	
	10	强制 PWM0B 输出高电平	
	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平 当计数器与比较器 1 匹配时, 倒计时。 (当 PWMCMP1 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM0B 输出低电平	
13:12 DOWNC1	10	强制 PWM0B 输出高电平	
	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平 当计数器匹配比较器 1 时, 计数。 (当 PWMCMP1 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM0B 输出低电平	
	10	强制 PWM0B 输出高电平	
15:14 UPC1	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平 当计数器与比较器 2 匹配时倒计时。 (当 PWMCMP2 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM0B 输出低电平	
	10	强制 PWM0B 输出高电平	
	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平 当计数器匹配比较器 2 时加计数。 (当 PWMCMP2 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
17:16 DOWNC2	00	禁用	
	01	强制 PWM0B 输出低电平	
	10	强制 PWM0B 输出高电平	
	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平 当计数器与比较器 3 匹配时倒计时。 (当 PWMCMP3 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
19:18 UPC2	01	强制 PWM0B 输出低电平	
	10	强制 PWM0B 输出高电平	
	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平 当计数器匹配比较器 3 时加计数。 (当 PWMCMP3 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM0B 输出低电平	
21:20 DOWNC3	10	强制 PWM0B 输出高电平	
	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平 当计数器匹配比较器 3 时加计数。 (当 PWMCMP3 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM0B 输出低电平	
	10	强制 PWM0B 输出高电平	
23:22 UPC3	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平	



25:24	DOWNC4	当计数器与比较器 4 匹配时倒计时。 (当 PWMCMP4 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM0B 输出低电平	
	10	强制 PWM0B 输出高电平	
	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平	
27:26	UPC4	当计数器匹配比较器 4 时加计数。 (当 PWMCMP4 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM0B 输出低电平	
	10	强制 PWM0B 输出高电平	
	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平	
29:28	DOWNC5	当计数器与比较器 5 匹配时倒计时。 (当 PWMCMP5 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM0B 输出低电平	
	10	强制 PWM0B 输出高电平	
	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平	
31:30	UPC5	当计数器匹配比较器 5 时加计数。 (当 PWMCMP5 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM0B 输出低电平	
	10	强制 PWM0B 输出高电平	
	11	切换 PWM0B: 低电平输出信号将被强制为高电平, 并且高电平信号将被强制为低电平	

### 16.15.8 PWM 1A 输出控制寄存器(OUTCON1A – 0x5001 C020)

设定 PWM 1A 脚位输出的时间点及脚位状态,当 OUTEN[2] = 1 时, 此寄存器才有效

表 17.13.8.1 PWM 1A 输出控制寄存器(OUTCON1A - 0x5001 C020)

位	符号	内容	描述	重置值
1:0	ZERO0		计数器 0 为零时	0
	00		禁用	
	01		强制 PWM1A 输出低电平	
	10		强制 PWM1A 输出高电平	
	11		切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
3:2	ZERO1		计数器 1 为零时	0
	00		禁用	
	01		强制 PWM1A 输出低电平	
	10		强制 PWM1A 输出高电平	
	11		切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
5:4	MAX0		计数器 0 与寄存器 MAX0 匹配时	0
	00		禁用	
	01		强制 PWM1A 输出低电平	
	10		强制 PWM1A 输出高电平	
	11		切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
7:6	MAX1		当计数器 1 匹配寄存器 MAX1 时	0
	00		禁用	
	01		强制 PWM1A 输出低电平	
	10		强制 PWM1A 输出高电平	
	11		切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
9:8	DOWNC0		当计数器匹配比较器 0 时倒计时。 (当 PWMCMPO 寄存器=0 时, 不支持此项功能)	0
	00		禁用	
	01		强制 PWM1A 输出低电平	

	10	强制 PWM1A 输出高电平	
	11	切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
11:10 UPC0		当计数器匹配比较器 0 时加计数。 (当 PWMCMP0 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM1A 输出低电平	
	10	强制 PWM1A 输出高电平	
	11	切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
13:12 DOWNC1		当计数器与比较器 1 匹配时, 倒计时。 (当 PWMCMP1 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM1A 输出低电平	
	10	强制 PWM1A 输出高电平	
	11	切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
15:14 UPC1		当计数器匹配比较器 1 时, 计数。 (当 PWMCMP1 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM1A 输出低电平	
	10	强制 PWM1A 输出高电平	
	11	切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
17:16 DOWNC2		当计数器与比较器 2 匹配时倒计时。 (当 PWMCMP2 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM1A 输出低电平	
	10	强制 PWM1A 输出高电平	
	11	切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
19:18 UPC2		当计数器匹配比较器 2 时加计数。 (当 PWMCMP2 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM1A 输出低电平	
	10	强制 PWM1A 输出高电平	
	11	切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
21:20 DOWNC3		当计数器与比较器 3 匹配时倒计时。 (当 PWMCMP3 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM1A 输出低电平	
	10	强制 PWM1A 输出高电平	
	11	切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
23:22 UPC3		当计数器匹配比较器 3 时加计数。 (当 PWMCMP3 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM1A 输出低电平	
	10	强制 PWM1A 输出高电平	
	11	切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
25:24 DOWNC4		当计数器与比较器 4 匹配时倒计时。 (当 PWMCMP4 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM1A 输出低电平	
	10	强制 PWM1A 输出高电平	
	11	切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
27:26 UPC4		当计数器匹配比较器 4 时加计数。 (当 PWMCMP4 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM1A 输出低电平	



	10	强制 PWM1A 输出高电平	
	11	切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
29:28 DOWNC5		当计数器与比较器 5 匹配时倒计时。 (当 PWMCMP5 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM1A 输出低电平	
	10	强制 PWM1A 输出高电平	
	11	切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
31:30 UPC5		当计数器匹配比较器 5 时加计数。 (当 PWMCMP5 寄存器 = MAX0/ MAX1 寄存器时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM1A 输出低电平	
	10	强制 PWM1A 输出高电平	
	11	切换 PWM1A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	

### 16.15.9 PWM 1B 输出控制寄存器(OUTCON1B – 0x5001 C024)

设定 PWM 1B 脚位输出的时间点及脚位状态, 当 OUTEN[3] = 1 时, 此寄存器才有效

表 17.13.9.1 PWM 1B 输出控制寄存器(OUTCON1B - 0x5001 C024)

位	符号	内容	描述	重置值
1:0	ZERO0		计数器 0 为零时	0
		00	禁用	
		01	强制 PWM1B 输出低电平	
		10	强制 PWM1B 输出高电平	
		11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
3:2	ZERO1		计数器 1 为零时	0
		00	禁用	
		01	强制 PWM1B 输出低电平	
		10	强制 PWM1B 输出高电平	
		11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
5:4	MAX0		计数器 0 与寄存器 MAX0 匹配时	0
		00	禁用	
		01	强制 PWM1B 输出低电平	
		10	强制 PWM1B 输出高电平	
		11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
7:6	MAX1		当计数器 1 匹配寄存器 MAX1 时	0
		00	禁用	
		01	强制 PWM1B 输出低电平	
		10	强制 PWM1B 输出高电平	
		11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
9:8	DOWNC0		当计数器匹配比较器 0 时倒计时。 (当 PWMCMP0 寄存器=0 时, 不支持此项功能)	0
		00	禁用	
		01	强制 PWM1B 输出低电平	
		10	强制 PWM1B 输出高电平	
		11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
11:10	UPC0		当计数器匹配比较器 0 时加计数。 (当 PWMCMP0 寄存器 = MAX0/ MAX1 寄存器时, 不支持此项功能)	0
		00	禁用	
		01	强制 PWM1B 输出低电平	
		10	强制 PWM1B 输出高电平	
		11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
13:12	DOWNC1		当计数器与比较器 1 匹配时, 倒计时。 (当 PWMCMP1 寄存器=0 时, 不支持此项功能)	0

	00	禁用	
	01	强制 PWM1B 输出低电平	
	10	强制 PWM1B 输出高电平	
	11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
15:14 UPC1		当计数器匹配比较器 1 时, 计数。 (当 PWMCMP1 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM1B 输出低电平	
	10	强制 PWM1B 输出高电平	
	11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
17:16 DOWNC2		当计数器与比较器 2 匹配时倒计时。 (当 PWMCMP2 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM1B 输出低电平	
	10	强制 PWM1B 输出高电平	
	11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
19:18 UPC2		当计数器匹配比较器 2 时加计数。 (当 PWMCMP2 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM1B 输出低电平	
	10	强制 PWM1B 输出高电平	
	11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
21:20 DOWNC3		当计数器与比较器 3 匹配时倒计时。 (当 PWMCMP3 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM1B 输出低电平	
	10	强制 PWM1B 输出高电平	
	11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
23:22 UPC3		当计数器匹配比较器 3 时加计数。 (当 PWMCMP3 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM1B 输出低电平	
	10	强制 PWM1B 输出高电平	
	11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
25:24 DOWNC4		当计数器与比较器 4 匹配时倒计时。 (当 PWMCMP4 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM1B 输出低电平	
	10	强制 PWM1B 输出高电平	
	11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
27:26 UPC4		当计数器匹配比较器 4 时加计数。 (当 PWMCMP4 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM1B 输出低电平	
	10	强制 PWM1B 输出高电平	
	11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
29:28 DOWNC5		当计数器与比较器 5 匹配时倒计时。 (当 PWMCMP5 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM1B 输出低电平	
	10	强制 PWM1B 输出高电平	
	11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
31:30 UPC5		当计数器匹配比较器 5 时加计数。 (当 PWMCMP5 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0

00	禁用
01	强制 PWM1B 输出低电平
10	强制 PWM1B 输出高电平
11	切换 PWM1B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平

#### 16.15.10 PWM 2A 输出控制寄存器(OUTCON2A – 0x5001 C028)

设定 PWM 2A 脚位输出的时间点及脚位状态,当 OUTEN[4] = 1 时, 此寄存器才有效

表 17.13.10.1 PWM 2A 输出控制寄存器(OUTCON2A - 0x5001 C028)

位	符号	内容	描述	重置值
1:0	ZERO0		计数器 0 为零时	0
		00	禁用	
		01	强制 PWM2A 输出低电平	
		10	强制 PWM2A 输出高电平	
		11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
3:2	ZERO1		计数器 1 为零时	0
		00	禁用	
		01	强制 PWM2A 输出低电平	
		10	强制 PWM2A 输出高电平	
		11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
5:4	MAX0		计数器 0 与寄存器 MAX0 匹配时	0
		00	禁用	
		01	强制 PWM2A 输出低电平	
		10	强制 PWM2A 输出高电平	
		11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
7:6	MAX1		当计数器 1 匹配寄存器 MAX1 时	0
		00	禁用	
		01	强制 PWM2A 输出低电平	
		10	强制 PWM2A 输出高电平	
		11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
9:8	DOWNC0		当计数器匹配比较器 0 时倒计时。 (当 PWMCMP0 寄存器=0 时, 不支持此项功能)	0
		00	禁用	
		01	强制 PWM2A 输出低电平	
		10	强制 PWM2A 输出高电平	
		11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
11:10	UPC0		当计数器匹配比较器 0 时加计数。 (当 PWMCMP0 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
		00	禁用	
		01	强制 PWM2A 输出低电平	
		10	强制 PWM2A 输出高电平	
		11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
13:12	DOWNC1		当计数器与比较器 1 匹配时, 倒计时。 (当 PWMCMP1 寄存器=0 时, 不支持此项功能)	0
		00	禁用	
		01	强制 PWM2A 输出低电平	
		10	强制 PWM2A 输出高电平	
		11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
15:14	UPC1		当计数器匹配比较器 1 时, 计数。 (当 PWMCMP1 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
		00	禁用	
		01	强制 PWM2A 输出低电平	

	10	强制 PWM2A 输出高电平	
	11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
17:16 DOWNC2		当计数器与比较器 2 匹配时倒计时。 (当 PWMCMP2 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM2A 输出低电平	
	10	强制 PWM2A 输出高电平	
	11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
19:18 UPC2		当计数器匹配比较器 2 时加计数。 (当 PWMCMP2 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM2A 输出低电平	
	10	强制 PWM2A 输出高电平	
	11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
21:20 DOWNC3		当计数器与比较器 3 匹配时倒计时。 (当 PWMCMP3 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM2A 输出低电平	
	10	强制 PWM2A 输出高电平	
	11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
23:22 UPC3		当计数器匹配比较器 3 时加计数。 (当 PWMCMP3 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM2A 输出低电平	
	10	强制 PWM2A 输出高电平	
	11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
25:24 DOWNC4		当计数器与比较器 4 匹配时倒计时。 (当 PWMCMP4 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM2A 输出低电平	
	10	强制 PWM2A 输出高电平	
	11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
27:26 UPC4		当计数器匹配比较器 4 时加计数。 (当 PWMCMP4 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM2A 输出低电平	
	10	强制 PWM2A 输出高电平	
	11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
29:28 DOWNC5		当计数器与比较器 5 匹配时倒计时。 (当 PWMCMP5 寄存器=0 时, 不支持此项功能)	0
	00	禁用	
	01	强制 PWM2A 输出低电平	
	10	强制 PWM2A 输出高电平	
	11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
31:30 UPC5		当计数器匹配比较器 5 时加计数。 (当 PWMCMP5 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00	禁用	
	01	强制 PWM2A 输出低电平	
	10	强制 PWM2A 输出高电平	
	11	切换 PWM2A: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	

#### 16.15.11 PWM 2B 输出控制寄存器(OUTCON2B – 0x5001 C02C)

设定 PWM 2B 脚位输出的时间点及脚位状态,当 OUTEN[5] = 1 时, 此寄存器才有效

表 17.13.11.1 PWM 2B 输出控制寄存器(OUTCON2B - 0x5001 C02C)

位	符号	内容	描述	重置值
1:0	ZERO0		计数器 0 为零时	0
		00	禁用	
		01	强制 PWM2B 输出低电平	
		10	强制 PWM2B 输出高电平	
		11	切换 PWM2B: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
3:2	ZERO1		计数器 1 为零时	0
		00	禁用	
		01	强制 PWM2B 输出低电平	
		10	强制 PWM2B 输出高电平	
		11	切换 PWM2B: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
5:4	MAX0		计数器 0 与寄存器 MAX0 匹配时	0
		00	禁用	
		01	强制 PWM2B 输出低电平	
		10	强制 PWM2B 输出高电平	
		11	切换 PWM2B: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
7:6	MAX1		当计数器 1 匹配寄存器 MAX1 时	0
		00	禁用	
		01	强制 PWM2B 输出低电平	
		10	强制 PWM2B 输出高电平	
		11	切换 PWM2B: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
9:8	DOWNC0		当计数器匹配比较器 0 时倒计时。 (当 PWMCMP0 寄存器=0 时, 不支持此项功能)	0
		00	禁用	
		01	强制 PWM2B 输出低电平	
		10	强制 PWM2B 输出高电平	
		11	切换 PWM2B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
11:10	UPC0		当计数器匹配比较器 0 时加计数。 (当 PWMCMP0 寄存器 = MAX0/ MAX1 寄存器时, 不支持此项功能)	0
		00	禁用	
		01	强制 PWM2B 输出低电平	
		10	强制 PWM2B 输出高电平	
		11	切换 PWM2B: 低电平输出信号将被强制为高电平, 高电平信号将被强制为低电平	
13:12	DOWNC1		当计数器与比较器 1 匹配时, 倒计时。 (当 PWMCMP1 寄存器=0 时, 不支持此项功能)	0
		00	禁用	
		01	强制 PWM2B 输出低电平	
		10	强制 PWM2B 输出高电平	
		11	切换 PWM2B: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
15:14	UPC1		当计数器匹配比较器 1 时, 计数。 (当 PWMCMP1 寄存器 = MAX0/ MAX1 寄存器时, 不支持此项功能)	0
		00	禁用	
		01	强制 PWM2B 输出低电平	
		10	强制 PWM2B 输出高电平	
		11	切换 PWM2B: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
17:16	DOWNC2		当计数器与比较器 2 匹配时倒计时。 (当 PWMCMP2 寄存器=0 时, 不支持此项功能)	0
		00	禁用	
		01	强制 PWM2B 输出低电平	
		10	强制 PWM2B 输出高电平	
		11	切换 PWM2B: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
19:18	UPC2		当计数器匹配比较器 2 时加计数。	0

	(当 PWMCMP2 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	
	00 禁用	
	01 强制 PWM2B 输出低电平	
	10 强制 PWM2B 输出高电平	
	11 切换 PWM2B: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
21:20 DOWNC3	当计数器与比较器 3 匹配时倒计时。 (当 PWMCMP3 寄存器=0 时, 不支持此项功能)	0
	00 禁用	
	01 强制 PWM2B 输出低电平	
	10 强制 PWM2B 输出高电平	
	11 切换 PWM2B: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
23:22 UPC3	当计数器匹配比较器 3 时加计数。 (当 PWMCMP3 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00 禁用	
	01 强制 PWM2B 输出低电平	
	10 强制 PWM2B 输出高电平	
	11 切换 PWM2B: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
25:24 DOWNC4	当计数器与比较器 4 匹配时倒计时。 (当 PWMCMP4 寄存器=0 时, 不支持此项功能)	0
	00 禁用	
	01 强制 PWM2B 输出低电平	
	10 强制 PWM2B 输出高电平	
	11 切换 PWM2B: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
27:26 UPC4	当计数器匹配比较器 4 时加计数。 (当 PWMCMP4 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00 禁用	
	01 强制 PWM2B 输出低电平	
	10 强制 PWM2B 输出高电平	
	11 切换 PWM2B: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
29:28 DOWNC5	当计数器与比较器 5 匹配时倒计时。 (当 PWMCMP5 寄存器=0 时, 不支持此项功能)	0
	00 禁用	
	01 强制 PWM2B 输出低电平	
	10 强制 PWM2B 输出高电平	
	11 切换 PWM2B: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	
31:30 UPC5	当计数器匹配比较器 5 时加计数。 (当 PWMCMP5 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	00 禁用	
	01 强制 PWM2B 输出低电平	
	10 强制 PWM2B 输出高电平	
	11 切换 PWM2B: 低输出信号将被强制为高电平, 高电平信号将被强制为低电平	

### 16.15.12 PWM 死区寄存器(DB0/1/2 – 0x5001 C048/4C/50)

死区寄存器的功能在于延迟输出讯号,避免输出讯号同时间开启,

表 17.13.12.1 PWM 死区寄存器( DB0 - 0x5001 C048 (PWM 0A / PWM 0B),  
DB1 - 0x5001 C04C (PWM 1A / PWM 1B),  
DB2 - 0x5001 C050 (PWM 2A / PWM 2B) )

位	符号	内容	描述	重置值
15:0	DBN		DB [15: 0]寄存器包含延迟 PWM_XA 和 PWM_XB 信号的时钟节拍数	0
16	DBAE	-	PWM_XA 边沿模式 DB [15: 0]寄存器在产生 PWM_XA 信号时延迟 PWM_XA 信号的边沿。 当 DBN = 0 时, 该位被忽略。	0
		0	上升沿	



	1	下降沿	
17	DBBE	PWM_XB 边沿模式 DB [15: 0]寄存器在产生 PWM_XB 信号时延迟 PWM_XB 信号的边沿。 当 DBN = 0 时, 该位被忽略。	0
	0	上升沿	
	1	下降沿	
31:18		保留	

### 16.15.13 PWM 比较寄存器(PWMCMP0/1/2/3/4/5 – 0x5001 C060/64/68/6C/70/74)

当 PWMCU0/1 等于 PWMCMP0/1/2/3/4/5 时, 将会产生不同的事件, 再利用事件去执行"中断/输出/更新/触发 ADC"等动作

表 17.13.13.1 PWM 比较寄存器( PWMCMP0 - 0x5001 C060, PWMCMP1 - 0x5001 C064, PWMCMP2 - 0x5001 C068, PWMCMP3 - 0x5001 C06C, PWMCMP4 - 0x5001 C070, PWMCMP5 - 0x5001 C074 )

位	符号	内容	描述	重置值
23:0	CMP	-	PWM 比较寄存器	FFFFFF
31:24	-	-	保留	

### 16.15.14 PWM 当前计数器寄存器(PWMCU0/1 – 0x5001 C078/7C)

计数器 0/1 目前的计数值

表 17.13.14.1 PWM 当前计数器(PWMCU0/1 - 0x5001 C078/7C)

位	符号	内容	描述	重置值
23:0	CU	-	PWM 当前计数器寄存器	0
31:24	-	-	保留	

### 16.15.15 PWM 周期比较寄存器(CYCMP0/1 – 0x5001 C080/84)

(当 CYCU0/1 = CYCMP0/1 时, 会产生事件, 利用事件去执行中断/更新/输出/触发 ADC 等动作).

表 17.13.15.1 PWM 周期比较寄存器(CYCMP0/1 - 0x5001 C080/84)

位	符号	内容	描述	重置值
15:0	CYCMP	-	PWM 周期比较寄存器	0
31:16	-	-	保留	

### 16.15.16 PWM 周期计数器寄存器(CYCU0/1 – 0x5001 C088/8C)

计数器 0/1(PWMCU0/1)为零时, CYCU0/1 循环增加,

表 17.13.16.1 PWM 周期计数器寄存器(CYCU0/1 - 0x5001 C088/8C)

位	符号	内容	描述	重置值
15:0	CYCU	-	PWM 周期计数器寄存器	0
31:16	-	-	保留	0

### 16.15.17 PWM ADC 触发延迟 0U 寄存器(ADCTDELOU – 0x5001 C090)

当 ADCTDEL0U == DELCU0 时(上数计数时), 硬件自动将 ADC 寄存器 (ADCCON[0]) 设置为 1

当 ADC 寄存器 (ADCCON[0] = 1) 时, ADC 开始转换

表 17.13.17.1 PWM ADC 触发延迟 0 寄存器(ADCTDEL0U - 0x5001 C090)

位	符号	内容	描述	重置值
23:0	ADCTDEL0U	-	PWM ADC 触发延迟 0 寄存器 仅在 PWMCU0 / PWMCU1 上数计数时使用 请参考 Table 17.16.46 ADCTDEL0D/ ADCTDEL0U 设置限制 说明	0
31:24	-	-	保留	0

#### 16.15.18 PWM ADC 延时当前计数器 0 寄存器(DEL0CU0 – 0x5001 C098)

当 ADC 触发事件发生时, 显示当前延迟计数器 0, 当 DEL0CU0 = ADCTDEL0U 时, ADC 开始转换工作

表 17.13.18.1 PWM ADC 延迟当前计数器 0 寄存器(DEL0CU0 - 0x5001 C098)

位	符号	内容	描述	重置值
23:0	DEL0CU	-	延迟当前计数器寄存器	0
31:24	-	-	保留	0

#### 16.15.19 PWM 中断使能设置寄存器(IEENS – 0x5001 C0A0)

设定发生中断的"事件"

表 17.13.19.1 PWM 中断使能设置寄存器(IEENS - 0x5001 C0A0)

位	符号	内容	描述	重置值
0	ZERO0	-	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0(PWMCU0)匹配 0。	0
1	ZERO1	-	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 1(PWMCU1)匹配 0。	0
2	MAX0	-	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0(PWMCU0)与 MAX0 寄存器匹配。	0
3	MAX1	-	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 1(PWMCU1)与 MAX1 寄存器匹配。	0
4	DOWMC0	-	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1(PWMCU0/PWMCU1)在倒数时具有匹配的比较器 0。 (当 PWMCMP0 寄存器=0 时, 不支持此项功能)	0
5	UPC0	-	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1(PWMCU0/PWMCU1)在计数时匹配比较器 0。 (当 PWMCMP0 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
6	DOWMC1	-	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1(PWMCU0/PWMCU1)在倒数时具有匹配的比较器 1。 (当 PWMCMP1 寄存器=0 时, 不支持此项功能)	0
7	UPC1	-	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1(PWMCU0/PWMCU1)在计数时匹配比较器 1。 (当 PWMCMP1 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
8	DOWMC2	-	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1(PWMCU0/PWMCU1)在计数时匹配比较器 2。	0



		(当 PWMCMP2 寄存器=0 时, 不支持此项功能)	
9	UPC2	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1(PWMCU0/PWMCU1)在倒计时时具有匹配的比较器 2。 (当 PWMCMP2 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
10	DOWMC3	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1(PWMCU0/PWMCU1)在计数时匹配比较器 3。 (当 PWMCMP3 寄存器=0 时, 不支持此项功能)	0
11	UPC3	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1(PWMCU0/PWMCU1)在倒计时时具有匹配的比较器 3。 (当 PWMCMP3 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
12	DOWMC4	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1(PWMCU0/PWMCU1)在计数时匹配比较器 4。 (当 PWMCMP4 寄存器=0 时, 不支持此项功能)	0
13	UPC4	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1(PWMCU0/PWMCU1)在倒计时时具有匹配的比较器 4。 (当 PWMCMP4 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
14	DOWMC2	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1(PWMCU0/PWMCU1)在计数时匹配比较器 5。 (当 PWMCMP5 寄存器=0 时, 不支持此项功能)	0
15	UPC2	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1(PWMCU0/PWMCU1)在倒计时时具有匹配的比较器 5。 (当 PWMCMP5 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
16	FAULTA	当 FAULT A 输入有效时。	0
17	FAULTB	当 FAULT B 输入有效时。	0
18	FAULTC	当 FAULT C 输入有效时。	0
19	FAULTD	当 FAULT D 输入有效时。	0
31:20	-	保留	0

### 16.15.20 PWM 中断使能清除寄存器(IEENC – 0x5001 C0A4)

表 17.13.20.1 PWM 中断使能清除寄存器(IEENC - 0x5001 C0A4)

位	符号	内容	描述	重置值
0	ZERO0	-	写 1 将清除 IEENS [0]	0
1	ZERO1	-	写 1 将清除 IEENS [1]	0
2	MAX0	-	写 1 将清除 IEENS [2]	0
3	MAX1	-	写 1 将清除 IEENS [3]	0
4	DOWMC0	-	写 1 会清除 IEENS [4]	0
5	UPC0	-	写 1 会清除 IEENS [5]	0
6	DOWMC1	-	写 1 将清除 IEENS [6]	0
7	UPC1	-	写 1 将清除 IEENS [7]	0
8	DOWMC2	-	写 1 将清除 IEENS [8]	0
9	UPC2	-	写 1 将清除 IEENS [9]	0
10	DOWMC3	-	写 1 将清除 IEENS [10]	0
11	UPC3	-	写 1 会清除 IEENS [11]	0
12	DOWMC4	-	写 1 将清除 IEENS [12]	0
13	UPC4	-	写 1 会清除 IEENS [13]	0

14	DOWMC5	写 1 将清除 IEENS [14]	0
15	UPC5	写 1 会清除 IEENS [15]	0
16	FAULTA	写 1 将清除 IEENS [16]	0
17	FAULTB	写 1 将清除 IEENS [17]	0
18	FAULTC	写 1 会清除 IEENS [18]	0
19	FAULTD	写 1 会清除 IEENS [19]	0
31:20	-	保留	0

### 16.15.21 PWM 事件状态寄存器(EVST – 0x5001 C0A8)

显示发生"事件"的状态

表 17.13.21.1 PWM 事件状态寄存器(EVST - 0x5001 C0A8)

位	符号	内容	描述	重置值
0	ZERO0	-	当以下条件成立时： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 匹配 0。	0
1	ZERO1	-	当以下条件成立时： 1.周期计数器具有匹配的周期比较器。 2.计数器 1 匹配 0。	0
2	MAX0	-	当以下条件成立时： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 与 MAX0 寄存器匹配。	0
3	MAX1	-	当以下条件成立时： 1.周期计数器具有匹配的周期比较器。 2.计数器 1 与 MAX1 寄存器匹配。	0
4	DOWMC0	-	当以下条件成立时： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在倒计时时具有匹配的比较器 0。 (当 PWMCMP0 寄存器=0 时, 不支持此项功能)	0
5	UPC0	-	当以下条件成立时： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在计数时匹配比较器 0。 (当 PWMCMP0 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
6	DOWMC1	-	当以下条件成立时： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在倒计时时具有匹配的比较器 1。 (当 PWMCMP1 寄存器=0 时, 不支持此项功能)	0
7	UPC1	-	当以下条件成立时： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在计数时匹配比较器 1。 (当 PWMCMP1 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
8	DOWMC2	-	当以下条件成立时： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在计数时匹配比较器 2。 (当 PWMCMP2 寄存器=0 时, 不支持此项功能)	0
9	UPC2	-	当以下条件成立时： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在倒计时时具有匹配的比较器 2。 (当 PWMCMP2 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
10	DOWMC3	-	当以下条件成立时： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在计数时匹配比较器 3。 (当 PWMCMP3 寄存器=0 时, 不支持此项功能)	0
11	UPC3	-	当以下条件成立时： 1.周期计数器具有匹配的周期比较器。	0

		2.计数器 0 /计数器 1 在倒计时时具有匹配的比较器 3。 (当 PWMCMP3 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	
12	DOWMC4	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在计数时匹配比较器 4。 (当 PWMCMP4 寄存器=0 时, 不支持此项功能)	0
13	UPC4	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在倒计时时具有匹配的比较器 4。 (当 PWMCMP4 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
14	DOWMC5	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在计数时匹配比较器 5。 (当 PWMCMP5 寄存器=0 时, 不支持此项功能)	0
15	UPC5	当以下条件成立时: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在倒计时时具有匹配的比较器 5。 (当 PWMCMP5 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
16	FAULTA	当 FAULT A 输入有效时。	0
17	FAULTB	当 FAULT B 输入有效时。	0
18	FAULTC	当 FAULT C 输入有效时。	0
19	FAULTD	当 FAULT D 输入有效时。	0
31:20	-	保留	0

#### 16.15.22 PWM 事件状态清除寄存器(EVSTC – 0x5001 C0AC)

表 17.13.22.1 PWM 事件状态清除寄存器(EVSTC - 0x5001 C0AC)

位	符号	内容	描述	重置值
0	ZERO0	-	写 1 将清除 EVST [0]	0
1	ZERO1	-	写 1 将清除 EVST [1]	0
2	MAX0	-	写 1 将清除 EVST [2]	0
3	MAX1	-	写 1 将清除 EVST [3]	0
4	DOWMC0	-	写 1 将清除 EVST [4]	0
5	UPC0	-	写 1 将清除 EVST [5]	0
6	DOWMC1	-	写 1 将清除 EVST [6]	0
7	UPC1	-	写 1 将清除 EVST [7]	0
8	DOWMC2	-	写 1 将清除 EVST [8]	0
9	UPC2	-	写 1 会清除 EVST [9]	0
10	DOWMC3	-	写 1 将清除 EVST [10]	0
11	UPC3	-	写 1 会清除 EVST [11]	0
12	DOWMC4	-	写 1 将清除 EVST [12]	0
13	UPC4	-	写 1 会清除 EVST [13]	0
14	DOWMC5	-	写 1 将清除 EVST [14]	0
15	UPC5	-	写 1 会清除 EVST [15]	0
16	FAULTA	-	写 1 将清除 EVST [16]	0
17	FAULTB	-	写 1 会清除 EVST [17]	0
18	FAULTC	-	写 1 会清除 EVST [18]	0
19	FAULTD	-	写 1 会清除 EVST [19]	0
31:20	-	-	保留	0

#### 16.15.23 PWM 故障配置寄存器(FLTCON – 0x5001 C0B0)

设置故障讯号的来源及有效的电平

表 17.13.23.1 PWM 故障配置寄存器(FLTCON - 0x5001 C0B0)

位	符号	内容	描述	重置值
---	----	----	----	-----

0	ENA	故障 A 使能位	0
		0 禁用	
		1 使能	
		事件触发故障 A	0
1	ACTA	0 低电平有效	
		1 高电平有效	
2	SSA0	选择故障 A 引脚作为故障 A 信号	0
		0 禁用	
		1 使能	
		选择比较器 0 作为故障 A 信号	0
3	SSA1	0 禁用	
		1 使能	
4	SSA2	选择比较器 1 作为故障 A 信号	0
		0 禁用	
		1 使能	
		选择比较器 2 作为故障 A 信号	0
5	SSA3	0 禁用	
		1 使能	
7:6	-	保留	0
8	ENB	故障 B 使能位	0
		0 禁用	
		1 使能	
		事件触发故障 B	0
9	ACTB	0 低电平有效	
		1 高电平有效	
10	SSB0	选择故障 B 引脚作为故障 B 信号	0
		0 禁用	
		1 使能	
		选择比较器 0 作为故障 B 信号	0
11	SSB1	0 禁用	
		1 使能	
12	SSB2	选择比较器 1 作为故障 B 信号	0
		0 禁用	
		1 使能	
		选择比较器 2 作为故障 B 信号	0
13	SSB3	0 禁用	
		1 使能	
15:14	-	保留	0
16	ENC	故障 C 使能位	0
		0 禁用	
		1 使能	
		事件触发故障 C	0
17	ACTC	0 低电平有效	
		1 高电平有效	
18	SSC0	选择故障 C 引脚作为故障 C 信号	0
		0 禁用	
		1 使能	
		选择比较器 0 作为故障 C 信号	0
19	SSC1	0 禁用	
		1 使能	
20	SSC2	选择比较器 1 作为故障 C 信号	0
		0 禁用	
		1 使能	
		选择比较器 2 作为故障 C 信号	0
21	SSC3		

	0	禁用	
	1	使能	
23:22 -	-	保留	0
24	END0	故障 D 使能位 (PWM0A 和 PWM0B) PWM0A 和 PWM0B 的电平=FLTCON[31:0]时,产生故障讯号	0
	0	禁用	
	1	使能	
25	END1	故障 D 使能位 (PWM1A 和 PWM1B) PWM1A 和 PWM1B 的电平=FLTCON[31:0]时,产生故障讯号	0
	0	禁用	
	1	使能	
26	END2	故障 D 使能位 (PWM2A 和 PWM2B) PWM2A 和 PWM2B 的电平=FLTCON[31:0]时,产生故障讯号	0
	0	禁用	
	1	使能	
29:27 -	-	保留	
31:30	DLEVEL	当以下条件成立时, 故障 D 发生事件。	0
	0	PWMxA 是低电平 PWMxB 是低电平	
	1	PWMxA 是低电平 PWMxB 是高电平	
	2	PWMxA 是高电平 PWMxB 是低电平	
	3	PWMxB 是高电平 PWMxA 是高电平	

#### 16.15.24 PWM 故障输出使能寄存器(FLTOTEN – 0x5001 C0B4)

当故障讯号产生时, 需要强制输出固定电平的脚步(输出电平请参考 FLTOL 寄存器)

表 17.13.24.1 PWM 故障输出使能寄存器(FLTOTEN - 0x5001 C0B4)

位	符号	内容	描述	重置值
0	PWM0A		PWM0A 输出信号在故障状态	0
		0	禁用	
		1	使能	
1	PWM0B		PWM0B 输出信号处于故障状态	0
		0	禁用	
		1	使能	
2	PWM1A		PWM1A 输出信号在故障条件下	0
		0	禁用	
		1	使能	
3	PWM1B		PWM1B 输出信号处于故障状态	0
		0	禁用	
		1	使能	
4	PWM2A		PWM2A 输出信号处于故障状态	0
		0	禁用	
		1	使能	
5	PWM2B		PWM2B 输出信号处于故障状态	0
		0	禁用	
		1	使能	
31:6	-	-	保留	0

#### 16.15.25 PWM 故障输出电平寄存器(FLTOL – 0x5001 C0B8)

故障讯号产生时,强制输出电平の設定

表 17.13.25.1 故障输出电平寄存器(FLTOL - 0x5001 C0B8)

位	符号	内容	描述	重置值
0	PWM0A		PWM0A 输出信号在故障状态	0
		0	强制 PWM0A 输出低电平	
		1	强制 PWM0A 输出高电平	
1	PWM0B		PWM0B 输出信号处于故障状态	1
		0	强制 PWM0B 输出低电平	
		1	强制 PWM0B 输出高电平	
2	PWM1A		PWM1A 输出信号在故障条件下	0
		0	强制 PWM1A 输出低电平	
		1	强制 PWM1A 输出高电平	
3	PWM1B		PWM1B 输出信号处于故障状态	1
		0	强制 PWM1B 输出低电平	
		1	强制 PWM1B 输出高电平	
4	PWM2A		PWM2A 输出信号处于故障状态	0
		0	强制 PWM2A 输出低电平	
		1	强制 PWM2A 输出高电平	
5	PWM2B		PWM2B 输出信号处于故障状态	1
		0	强制 PWM2B 输出低电平	
		1	强制 PWM2B 输出高电平	
31:6	-	-	保留	

16.15.26 PWM 故障噪声滤波器寄存器(FLTNFA – 0x5001 C0BC)

表 17.13.26.1 PWM 故障噪声滤波器寄存器(FLTNFA - 0x5001 C0BC)

位	符号	内容	描述	重置值
15:0	FLTNFA	-	PWM 故障 A 噪声滤波器 (unit: system clock)	0
31:16	-	-	保留	

16.15.27 PWM 故障 B 噪声滤波器寄存器(FLTNFB – 0x5001 C0C0)

表 17.13.27.1 PWM 故障 B 噪声滤波器寄存器(FLTNFB - 0x5001 C0C0)

位	符号	内容	描述	重置值
15:0	FLTNFB	-	PWM 故障 B 噪声滤波器 (unit: system clock)	0
31:16	-	-	保留	

16.15.28 PWM 故障 C 噪声滤波器寄存器(FLTNFC – 0x5001 C0C4)

表 17.13.28.1 PWM 故障 C 噪声滤波器寄存器(FLTNFC - 0x5001 C0C4)

位	符号	内容	描述	重置值
15:0	FLTNFC	-	PWM 故障 C 噪声滤波器 (unit: system clock)	0
31:16	-	-	保留	

16.15.29 PWM 状态寄存器(STA – 0x5001 C0C8)

表 17.13.29.1 PWM 状态寄存器(STA - 0x5001 C0C8)

位	符号	内容	描述	重置值
0	DIR0		PWM2 当前计数 0 方向状态	0
		0	0 = 递增计数	
		1	1 = 递减计数	
7:1	-	-	保留	0

8	DIR1	-	PWM 电流计数 1 方向状态	0
		0	0 =递增计数	
		1	1 =递减计数	
31:9	-	-	保留	0

### 16.15.30 PWM 触发器 ADC 0 使能寄存器(TADC0 – 0x5001 C0CC)

设定触发 ADC 的"事件"

表 17.13.30.1 PWM 触发器 ADC 0 使能寄存器(TADC0 - 0x5001 C0CC)

位	符号	内容	描述	重置值
0	ZERO0		当以下条件成立时，PWM 触发 ADC x 开始转换： 1.周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3.计数器 0 匹配 0。	0
		0	禁用	
		1	使能	
1	ZERO1		当以下条件成立时，PWM 触发 ADC x 开始转换： 1.周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3.计数器 1 匹配 0。	0
		0	禁用	
		1	使能	
2	MAX0		当以下条件成立时，PWM 触发 ADC x 开始转换： 1.周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3.计数器 0 具有匹配的寄存器 MAX0。	0
		0	禁用	
		1	使能	
3	MAX1		当以下条件成立时，PWM 触发 ADC x 开始转换： 1.周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3.计数器 1 具有匹配的寄存器 MAX1。	0
		0	禁用	
		1	使能	
4	DOWMCO		当以下条件成立时，PWM 触发 ADC x 开始转换： 1.周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3.计数器 0 /计数器 1 在匹配比较器 0 时进行递减计数。 (当 PWMCMPO 寄存器=0 时，不支持此项功能)	0
		0	禁用	
		1	使能	



5	UPC0	0	当以下条件成立时, PWM 触发 ADC x 开始转换: 1. 周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3. 计数器 0 / 计数器 1 具有匹配的比较器 0, 同时加计数。 (当 PWMCMP0 寄存器 = MAX0/ MAX1 寄存器时, 不支持此项功能)	0	
		1			
		0			禁用
6	DOWMC1	0	当以下条件成立时, PWM 触发 ADC x 开始转换: 1. 周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3. 计数器 0 / 计数器 1 在匹配比较器 0 时进行递减计数。 (当 PWMCMP1 寄存器=0 时, 不支持此项功能)	0	
		1			使能
		0			禁用
7	UPC1	0	当以下条件成立时, PWM 触发 ADC x 开始转换: 1. 周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3. 计数器 0 / 计数器 1 具有匹配的比较器 0, 同时加计数。 (当 PWMCMP1 寄存器 = MAX0/ MAX1 寄存器时, 不支持此项功能)	0	
		1			使能
		0			禁用
8	DOWMC2	0	当以下条件成立时, PWM 触发 ADC x 开始转换: 1. 周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3. 计数器 0 / 计数器 1 在匹配比较器 0 时进行递减计数。 (当 PWMCMP2 寄存器=0 时, 不支持此项功能)	0	
		1			使能
		0			禁用
9	UPC2	0	当以下条件成立时, PWM 触发 ADC x 开始转换: 1. 周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3. 计数器 0 / 计数器 1 具有匹配的比较器 0, 同时加计数。 (当 PWMCMP2 寄存器 = MAX0/ MAX1 寄存器时, 不支持此项功能)	0	
		1			使能
		0			禁用
10	DOWMC3	0	当以下条件成立时, PWM 触发 ADC x 开始转换: 1. 周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3. 计数器 0 / 计数器 1 在匹配比较器 0 时进行递减计数。 (当 PWMCMP3 寄存器=0 时, 不支持此项功能)	0	
		1			使能
		0			禁用



11	UPC3	0	当以下条件成立时, PWM 触发 ADC x 开始转换: 1.周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3.计数器 0 /计数器 1 具有匹配的比较器 0, 同时加计数。 (当 PWMCMP3 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
		1		
		0		
		1	使能	
12	DOWMC4	0	当以下条件成立时, PWM 触发 ADC x 开始转换: 1.周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3.计数器 0 /计数器 1 在匹配比较器 0 时进行递减计数。 (当 PWMCMP4 寄存器=0 时, 不支持此项功能)	0
		1		
		0		
		1	使能	
13	UPC4	0	当以下条件成立时, PWM 触发 ADC x 开始转换: 1.周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3.计数器 0 /计数器 1 具有匹配的比较器 0, 同时加计数。 (当 PWMCMP4 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
		1		
		0		
		1	使能	
14	DOWMC5	0	当以下条件成立时, PWM 触发 ADC x 开始转换: 1.周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3.计数器 0 /计数器 1 在匹配比较器 0 时进行递减计数。 (当 PWMCMP5 寄存器=0 时, 不支持此项功能)	0
		1		
		0		
		1	使能	
15	UPC5	0	当以下条件成立时, PWM 触发 ADC x 开始转换: 1.周期计数器具有匹配的周期比较器。 2. ADCTDEL 匹配 DELCU。 3.计数器 0 /计数器 1 具有匹配的比较器 0, 同时加计数。 (当 PWMCMP5 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
		1		
		0		
		1	使能	
31:16	-	-	保留	

### 16.15.31 PWM 更新模式使能寄存器(UPDATE – 0x5001 C0D4)

设定更新寄存器的"事件"

更新的寄存器为( MAX0, MAX1, PWMCMP0, PWMCMP1, PWMCMP2 , PWMCMP3 , PWMCMP4 , PWMCMP5 , OUTCON0A, OUTCON0B, OUTCON1A, OUTCON1B, OUTCON2A, OUTCON2B0)

表 17.13.31.1 PWM 更新模式使能寄存器(UPDATE - 0x5001 C0D4)

位	符号	内容	描述	重置值
0	ZERO0	0	当以下条件成立时, 更新到"Double Buff"寄存器: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 匹配 0。	
		1	禁用	

	1	使能	
1	ZERO1	当以下条件成立时，更新到“Double Buff”寄存器： 1.周期计数器具有匹配的周期比较器。 2.计数器 1 匹配 0。	0
	0	禁用	
	1	使能	
2	MAX0	当以下条件成立时，更新到“Double Buff”寄存器： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 具有匹配的寄存器 MAX0。	0
	0	禁用	
	1	使能	
3	MAX1	当以下条件成立时，更新到“Double Buff”寄存器： 1.周期计数器具有匹配的周期比较器。 2.计数器 1 具有匹配的寄存器 MAX0。	0
	0	禁用	
	1	使能	
4	DOWMC0	当以下条件成立时，更新到“Double Buff”寄存器： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 具有匹配比较器 0 的同时减计数。 (当 PWMCMP0 寄存器=0 时, 不支持此项功能)	0
	0	禁用	
	1	使能	
5	UPC0	当以下条件成立时，更新到“Double Buff”寄存器： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在匹配比较器 0 时进行加计数。 (当 PWMCMP0 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	0	禁用	
	1	使能	
6	DOWMC1	当以下条件成立时，更新到“Double Buff”寄存器： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 具有匹配比较器 1 的同时减计数。 (当 PWMCMP1 寄存器=0 时, 不支持此项功能)	0
	0	禁用	
	1	使能	
7	UPC1	当以下条件成立时，更新到“Double Buff”寄存器： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在匹配比较器 1 时进行加计数。 (当 PWMCMP1 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	0	禁用	
	1	使能	
8	DOWMC2	当以下条件成立时，更新到“Double Buff”寄存器： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 具有匹配比较器 2 的同时减计数。 (当 PWMCMP2 寄存器=0 时, 不支持此项功能)	0
	0	禁用	
	1	使能	
9	UPC2	当以下条件成立时，更新到“Double Buff”寄存器： 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在匹配比较器 2 时进行加计数。 (当 PWMCMP2 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	0	禁用	
	1	使能	
10	DOWMC3	当以下条件成立时，更新到“Double Buff”寄存器： 1.周期计数器具有匹配的周期比较器。	0

		2.计数器 0 /计数器 1 具有匹配比较器 3 的同时减计数。 (当 PWMCMP3 寄存器=0 时, 不支持此项功能)	
	0	禁用	
	1	使能	
11	UPC3	当以下条件成立时, 更新到“Double Buff”寄存器: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在匹配比较器 3 时进行加计数。 (当 PWMCMP3 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	0	禁用	
	1	使能	
12	DOWMC4	当以下条件成立时, 更新到“Double Buff”寄存器: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 具有匹配比较器 4 的同时减计数。 (当 PWMCMP4 寄存器=0 时, 不支持此项功能)	0
	0	禁用	
	1	使能	
13	UPC4	当以下条件成立时, 更新到“Double Buff”寄存器: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在匹配比较器 4 时进行加计数。 (当 PWMCMP4 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	0	禁用	
	1	使能	
14	DOWMC5	当以下条件成立时, 更新到“Double Buff”寄存器: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 具有匹配比较器 2 的同时减计数。 (当 PWMCMP2 寄存器=0 时, 不支持此项功能)	0
	0	禁用	
	1	使能	
15	UPC5	当以下条件成立时, 更新到“Double Buff”寄存器: 1.周期计数器具有匹配的周期比较器。 2.计数器 0 /计数器 1 在匹配比较器 5 时进行加计数。 (当 PWMCMP5 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	0	禁用	
	1	使能	
16	IMMED	立即	1
	0	禁用	
	1	使能	
31:17	-	保留	

### 16.15.32 PWM 故障释放寄存器(FLTREL – 0x5001 C0D8)

设定故障释放的"事件"

表 17.13.32.1 PWM 故障释放使能寄存器(FLTREL - 0x5001 C0D8)

位	符号	内容	描述	重置值
0	ZERO0		当以下条件成立时, 释放故障输出: 1. FLTST 寄存器= 0x00。 2.计数器 0 匹配 0。 (当 FLTREL [31]寄存器等于 1 时, 忽略 item1 和 FLTST 寄存器被清除为 0)	
		0	禁用	
		1	使能	
1	ZERO1		当以下条件成立时, 释放故障输出: 1. FLTST 寄存器= 0x00。 2.计数器 1 匹配 0。	0

		(当 FLTREL [31]寄存器等于 1 时, 忽略 item1 和 FLTST 寄存器被清除为 0)	
	0	禁用	
	1	使能	
2	MAX0	当以下条件成立时, 释放故障输出: 1. FLTST 寄存器= 0x00。 2.计数器 0 匹配 MAX0。 (当 FLTREL [31]寄存器等于 1 时, 忽略 item1 和 FLTST 寄存器被清除为 0)	0
	0	禁用	
	1	使能	
3	MAX1	当以下条件成立时, 释放故障输出: 1. FLTST 寄存器= 0x00。 2.计数器 0 匹配 MAX1。 (当 FLTREL [31]寄存器等于 1 时, 忽略 item1 和 FLTST 寄存器被清除为 0)	0
	0	禁用	
	1	使能	
4	DOWMC0	当以下条件成立时, 释放故障输出: 1. FLTST 寄存器= 0x00。 2.计数器 0 /计数器 1 具有匹配比较器 0 的同时减计数。 (当 FLTREL [31]寄存器等于 1 时, 忽略 item1 和 FLTST 寄存器被清除为 0) (当 PWMCMP0 寄存器=0 时, 不支持此项功能)	0
	0	禁用	
	1	使能	
5	UPC0	当以下条件成立时, 释放故障输出: 1. FLTST 寄存器= 0x00。 2.计数器 0 /计数器 1 在匹配比较器 0 时进行加计数。 (当 FLTREL [31]寄存器等于 1 时, 忽略 item1 和 FLTST 寄存器被清除为 0) (当 PWMCMP0 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	0	禁用	
	1	使能	
6	DOWMC1	当以下条件成立时, 释放故障输出: 1. FLTST 寄存器= 0x00。 2.计数器 0 /计数器 1 具有匹配比较器 1 的同时减计数。 (当 FLTREL [31]寄存器等于 1 时, 忽略 item1 和 FLTST 寄存器被清除为 0) (当 PWMCMP1 寄存器=0 时, 不支持此项功能)	0
	0	禁用	
	1	使能	
7	UPC1	当以下条件成立时, 释放故障输出: 1. FLTST 寄存器= 0x00。 2.计数器 0 /计数器 1 在匹配比较器 1 时进行加计数。 (当 FLTREL [31]寄存器等于 1 时, 忽略 item1 和 FLTST 寄存器被清除为 0) (当 PWMCMP1 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	0	禁用	
	1	使能	
8	DOWMC2	当以下条件成立时, 释放故障输出: 1. FLTST 寄存器= 0x00。 2.计数器 0 /计数器 1 具有匹配比较器 2 的同时减计数。 (当 FLTREL [31]寄存器等于 1 时, 忽略 item1 和 FLTST 寄存器被清除为 0) (当 PWMCMP2 寄存器=0 时, 不支持此项功能)	0
	0	禁用	
	1	使能	
9	UPC2	当以下条件成立时, 释放故障输出: 1. FLTST 寄存器= 0x00。 2.计数器 0 /计数器 1 在匹配比较器 2 时进行加计数。 (当 FLTREL [31]寄存器等于 1 时, 忽略 item1 和 FLTST 寄存器被清除为 0) (当 PWMCMP2 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0

	0	禁用	
	1	使能	
10	DOWMC3	当以下条件成立时，释放故障输出： 1. FLTST 寄存器= 0x00。 2. 计数器 0 /计数器 1 具有匹配比较器 3 的同时减计数。 (当 FLTREL [31]寄存器等于 1 时，忽略 item1 和 FLTST 寄存器被清除为 0) (当 PWMCMP3 寄存器=0 时，不支持此项功能)	0
	0	禁用	
	1	使能	
11	UPC3	当以下条件成立时，释放故障输出： 1. FLTST 寄存器= 0x00。 2. 计数器 0 /计数器 1 在匹配比较器 3 时进行加计数。 (当 FLTREL [31]寄存器等于 1 时，忽略 item1 和 FLTST 寄存器被清除为 0) (当 PWMCMP3 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	0	禁用	
	1	使能	
12	DOWMC4	当以下条件成立时，释放故障输出： 1. FLTST 寄存器= 0x00。 2. 计数器 0 /计数器 1 具有匹配比较器 4 的同时减计数。 (当 FLTREL [31]寄存器等于 1 时，忽略 item1 和 FLTST 寄存器被清除为 0) (当 PWMCMP4 寄存器=0 时，不支持此项功能)	0
	0	禁用	
	1	使能	
13	UPC4	当以下条件成立时，释放故障输出： 1. FLTST 寄存器= 0x00。 2. 计数器 0 /计数器 1 在匹配比较器 4 时进行加计数。 (当 FLTREL [31]寄存器等于 1 时，忽略 item1 和 FLTST 寄存器被清除为 0) (当 PWMCMP4 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	0	禁用	
	1	使能	
14	DOWMC5	当以下条件成立时，释放故障输出： 1. FLTST 寄存器= 0x00。 2. 计数器 0 /计数器 1 具有匹配比较器 5 的同时减计数。 (当 FLTREL [31]寄存器等于 1 时，忽略 item1 和 FLTST 寄存器被清除为 0) (当 PWMCMP5 寄存器=0 时，不支持此项功能)	0
	0	禁用	
	1	使能	
15	UPC5	当以下条件成立时，释放故障输出： 1. FLTST 寄存器= 0x00。 2. 计数器 0 /计数器 1 在匹配比较器 5 时进行加计数。 (当 FLTREL [31]寄存器等于 1 时，忽略 item1 和 FLTST 寄存器被清除为 0) (当 PWMCMP5 寄存器 = MAX0/ MAX1 寄存器时,不支持此项功能)	0
	0	禁用	
	1	使能	
16	IMMED	当以下条件成立时，释放故障输出： 1. FLTST 寄存器= 0x00 (当 FLTREL [31]寄存器等于 0 时)	
30:17 -	-	保留	
31	MODE	故障解除模式	
	0	FLTST 寄存器由用户清除。	
	1	FLTST 寄存器由 HW 清除。	

### 16.15.33 PWM 故障状态寄存器(FLTST – 0x5001 C0DC)

**表 17.13.33.1 PWM 故障状态寄存器(FLTST - 0x5001 C0DC)**

位	符号	内容	描述	重置值
0	FAULTA	-	设置时, 发生故障 A 事件。(硬件配置) 用户可对 FLTST[0]寄存器设置 1 即可清除 FLTST[0]。(用户配置)	0
1	FAULTB	-	设置时, 发生故障 B 事件。(硬件配置) 用户可对 FLTST[1]寄存器设置 1 即可清除 FLTST[1]。(用户配置)	
2	FAULTC	-	设置时, 发生故障 C 事件。(硬件配置) 用户可对 FLTST[2]寄存器设置 1 即可清除 FLTST[2]。(用户配置)	
3	FAULTD	-	设置时, 发生故障 D 事件。(硬件配置) 用户可对 FLTST[3]寄存器设置 1 即可清除 FLTST[3]。(用户配置)	
31:4	-	-	保留	

**16.15.34 PWM 仿真模式寄存器(ICE - 0x5001 C0E0)**

使用仿真器时,当仿真器的"中断/停止"发生,其计数器和输出电平の設定会依照 ICE 寄存器的設定来变化

**表 17.13.34.1 PWM 仿真模式寄存器(ICE - 0x5001 C0E0)**

位	符号	内容	描述	重置值
1:0	BKCU0	-	发生断点时。	0
		0	禁用	
		1	PWMCU0 寄存器将被停止。	
		2	PWMCU0 寄存器将被停止并重置。	
3:2	-	-	保留	0
7:4	BKCU1	-	发生断点时。	0
		0	禁用	
		1	PWMCU1 寄存器将停止。	
		2	PWMCU1 寄存器将被停止并重置。	
8	BK0A	-	在断点条件下的 PWM0A 输出信号 (FLTOL [0])。	0
		0	禁用	
		1	使能	
9	BK0B	-	在断点条件下的 PWM0B 输出信号 (FLTOL [1])。	0
		0	禁用	
		1	使能	
10	BK1A	-	在断点条件下的 PWM1A 输出信号 (FLTOL [2])	0
		0	禁用	
		1	使能	
11	BK1B	-	在断点条件下的 PWM1B 输出信号 (FLTOL [3])	0
		0	禁用	
		1	使能	
12	BK2A	-	在断点条件下的 PWM2A 输出信号 (FLTOL [4])	0
		0	禁用	
		1	使能	
13	BK2B	-	在断点条件下的 PWM2B 输出信号 (FLTOL [5])	0
		0	禁用	
		1	使能	
31:14	-	-	保留	

**16.15.35 PWM 计数器初始寄存器(CUINIT0/1 - 0x5001 C0E4/E8)**

PWMCU0 寄存器的初始来源值

**表 17.13.35.1 PWM 计数器 0/1 初始寄存器(CUINIT0/1 - 0x5001 C0E4/E8)**

位	符号	内容	描述	重置值
23:0	CUINIT	-	计数器(PWMCU0/PWMCU1)的初始值	
31:24	-	-	保留	

**16.15.36 PWM 比较器触发控制寄存器(TACMP- 0x5001 C0EC)**

请参考

表 17.13.36.1 PWM 比较器触发控制寄存器(TACMP - 0x5001 C0EC)

位	符号	内容	描述	重置值
23:0	ACMPC	-	PWM 触发 ACOMP 寄存器	FFFFFF
28:24	-	-	保留	
29	PIN	0	PWM_0A/ PWM_1A/ PWM_2A	
		1	PWM_0B/ PWM_1B/ PWM_2B	
30	LEVEL	0	活动电平选择	
		1	高电平	
31	EN	0	触发 ACMP 功能启用位	
		1	禁用	

**16.15.37 PWM 当前计数器 COMP 寄存器(CUCOMP - 0x5001 C0F0)**

Table 17.16.40.1 PWM 当前计数器 COMP 寄存器(CUCOMP - 0x5001 C0F0)

位	符号	内容	描述	重置值
23:0	CUCOMP	-	TACMP[23:0] 当前计数器	
31:24	-	-	保留	

**16.15.38 PWM 输出固定寄存器(OUTFIX - 0x5001 C100)**

设定当 OUTEN =0 时的引脚电平

表 17.13.38.1 PWM 输出固定 FIX 寄存器(OUTFIX - 0x5001 C100)

位	符号	内容	描述	重置值
0	PWM0A	0	当 OUTEN [0]等于 0 时	0
		1	强制 PWM0A 输出低电平	
1	PWM0B	0	当 OUTEN [1]等于 0 时	0
		1	强制 PWM0B 输出低电平	
2	PWM1A	0	当 OUTEN [2]等于 0 时	0
		1	强制 PWM1A 输出低电平	
3	PWM1B	0	当 OUTEN [3]等于 0 时	0
		1	强制 PWM1B 输出低电平	
4	PWM2A	0	当 OUTEN [4]等于 0 时	0
		1	强制 PWM2A 输出低电平	
5	PWM2B	0	当 OUTEN [5]等于 0 时	0
		1	强制 PWM2B 输出低电平	
31:6	-	-	保留	0

**16.15.39 PWM 输出反相寄存器(OUTIVT - 0x5001 C104)**

将 PWM 的输出引脚电平做反向动作

表 17.13.39.1 PWM 输出反相寄存器(OUTIVT - 0x5001 C104)



位	符号	内容	描述	重置值
0	PWM0A		PWM0A 输出引脚的反相位	0
		0	闲置	
1	PWM0B		PWM0B 输出引脚反相位	0
		0	闲置	
2	PWM1A		PWM1A 输出引脚反转位	0
		0	闲置	
3	PWM1B		PWM1B 输出引脚反相位	0
		0	闲置	
4	PWM2A		PWM2A 输出引脚反相位	0
		0	闲置	
5	PWM2B		PWM2B 输出引脚反相位	0
		0	闲置	
31:6	-	-	保留	0

#### 16.15.40 PWM ADC 触发延迟 0D 寄存器(ADCTDEL0D – 0x5001 C108)

当 ADCTDEL0D = DELCU0 时(下数计数时)，硬件会自动将 ADC 寄存器 (ADCCON[0]) 设置为 1

当 ADC 寄存器 (ADCCON[0] = 1) 时，ADC 开始转换

Table 17.16.45.1 PWM ADC 触发延迟 0D 寄存器(ADCTDEL0D - 0x5001 C108)

位	符号	内容	描述	重置值
23:0	ADCTDELD	-	ADC 触发延迟寄存器 仅在 PWMCU0 / PWMCU1 下数计数时使用 请参考 Table 17.16.46 ADCTDEL0D/ ADCTDEL0U 设置限制 说明	0
31:24			保留	

Table 17.16.46 ADCTDEL0D/ ADCTDEL0U 设置限制

请注意以下限制，当设置 PWM ADC 触发延迟 0U 寄存器为触发 ADC 的延迟定时器时

ADCTDEL0U 内容值	ADCTDEL0D 内容值	ADCTDEL0U 设定
零值	零值	支持
零值	非零值	不支持
非零值	零值	支持
非零值	非零值	支持

说明: 当使用 ADCTDEL0U 为触发 ADC 的延迟定时器时且设置为零值，ADCTDEL0D 设置为非零值无法触发 ADC

请注意以下限制，当设置 PWM ADC 触发延迟 0D 寄存器为触发 ADC 的延迟定时器时

ADCTDEL0U 内容值	ADCTDEL0D 内容值	ADCTDEL0D 设定
零值	零值	支持
零值	非零值	支持
非零值	零值	不支持
非零值	非零值	支持

说明: 当使用 ADCTDEL0D 为触发 ADC 的延迟定时器时且设置为零值，ADCTDEL0U 设置为非零值无法触发 ADC



## 17. 坐标旋转数字计算机(CORDIC)

### 17.1 概况

坐标旋转数字计算机 (CORDIC) 算法具有高效和低成本潜力, 能够实现这一大类的应用程序, 其中包括三角函数, 对数和超越基本函数的生成, 复数乘法, 矩阵求逆, 线性系统和一般科学的解决方案计算。

CORDIC 算法藉由改变一些简单的参数, 它可以作为一个广泛基本超越函数来统一实现单一的算法, 包含对数, 指数和平方。

CORDIC 是非常简单的迭代收敛算法, 可以减少复杂的乘法运算, 大大地简化整体硬件的复杂性。基于 CORDIC 的计算的基本原理, 并针对不同的操作模式和平面坐标系提出其迭代算法。

### 17.2 特性

- 旋转和矢量模式
- 圆形, 线性 and 双曲线坐标系
- 配置  $\sin(\theta)$ ,  $\cos(\theta)$ ,  $a \tan(y/x)$ ,  $\sinh(\theta)$ ,  $\cosh(\theta)$ ,  $a \tanh(y/x)$

### 17.3 CORDIC 方块图

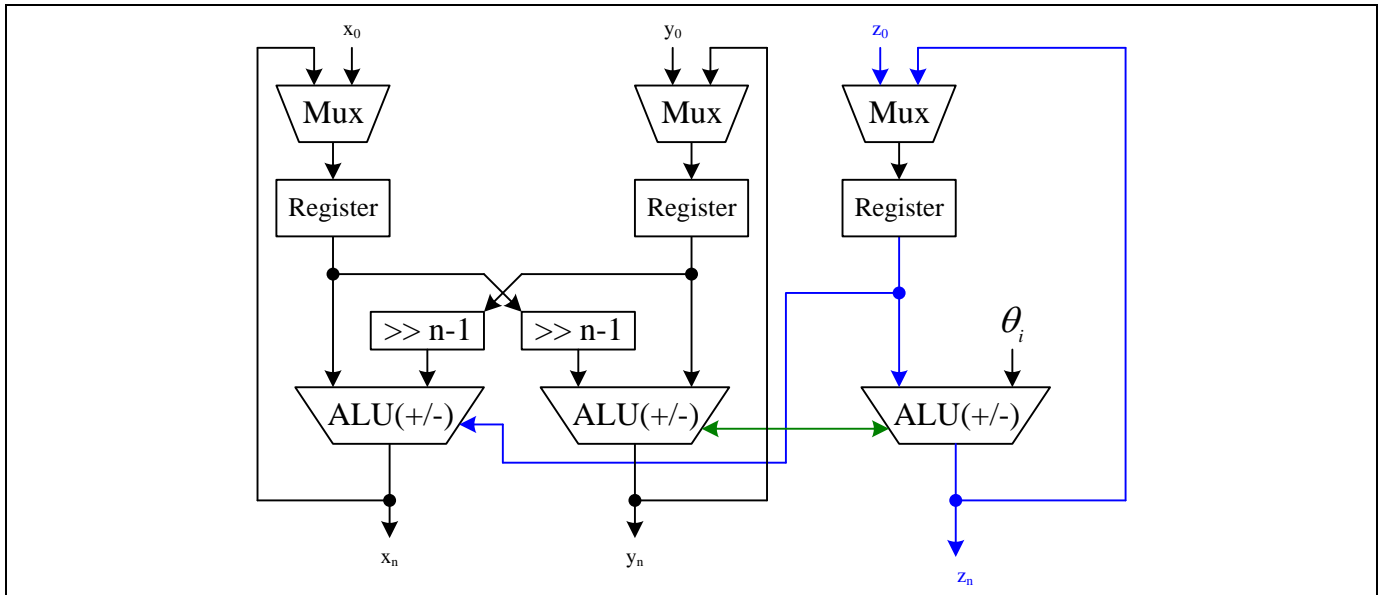


图 18.3.1 CORDIC 方块图.

## 17.4 功能说明

## 17.4.1 圆形坐标系(CORDIC : m=1)

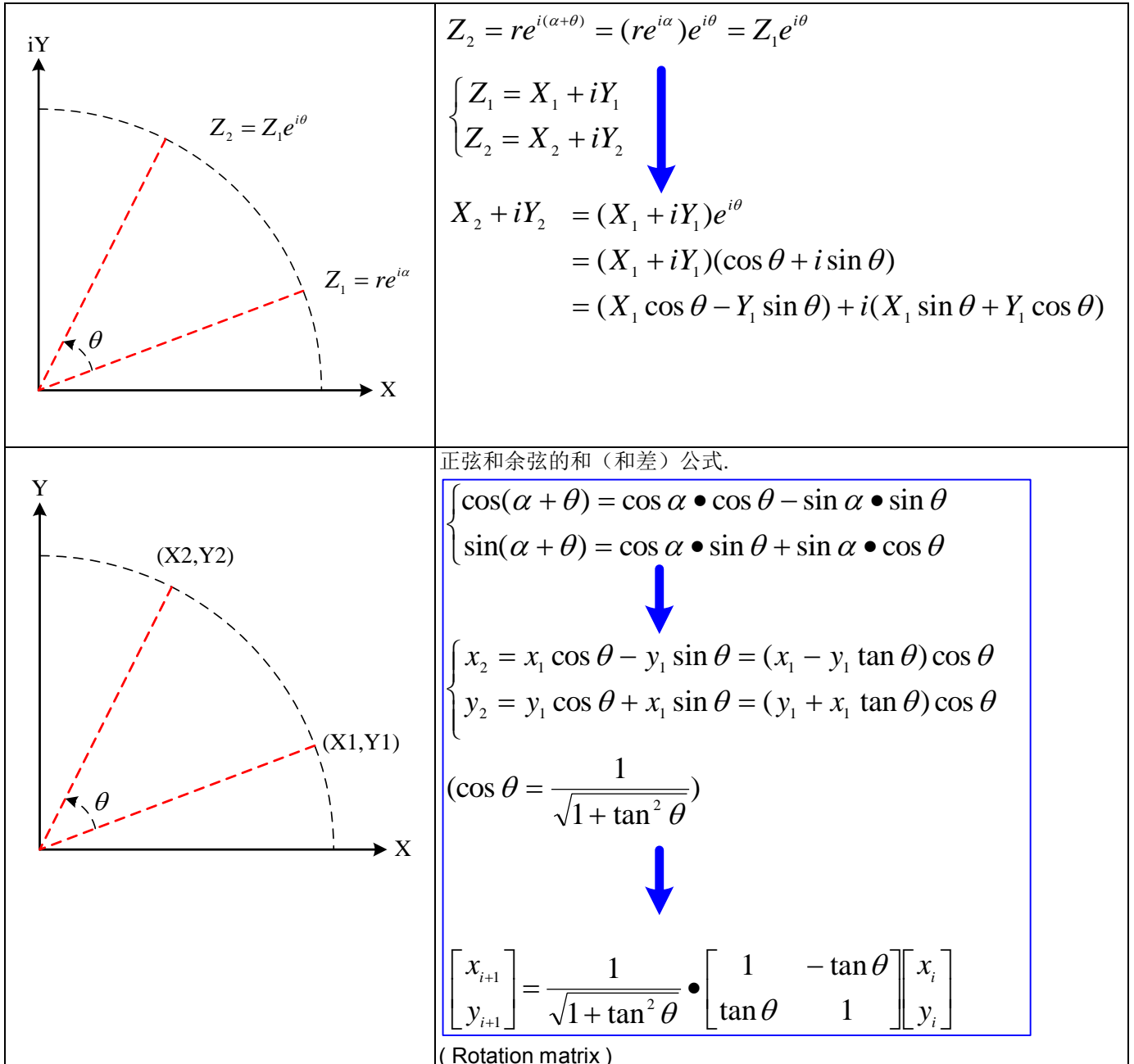
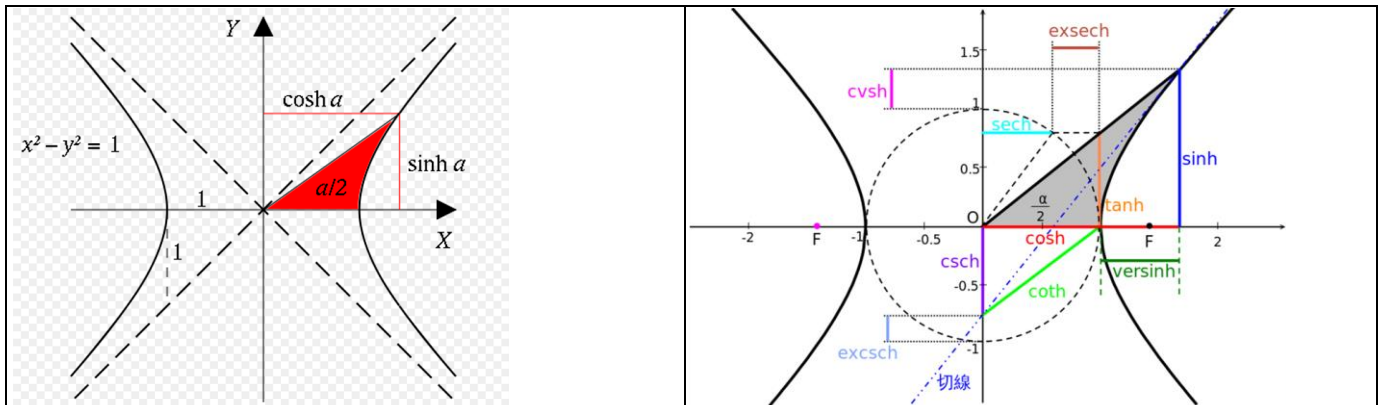
 六个三角函数也可以用单位圆来定义，单位圆是以原点为中心的半径圆。单位圆的方程是:  $X^2 + Y^2 = 1$ 


图 18.4.1.1 圆形坐标系 (CORDIC : m=1)

**17.4.2 双曲型坐标系 (CORDIC : m=-1)**

正弦函数和余弦函数也可以用单位圆来定义( $x^2 + y^2 = 1$ ).在数学中, 双曲函数是普通三角函数或圆函数的类似体透过原点射线截取单位双曲线  $x^2 - y^2 = 1$ , 在这一点上 ( $\cosh \alpha, \sinh \alpha$ ), 射线是双曲线和  $x$  轴之间的面积的两倍。对于  $x$  轴下的双曲线上的点, 该区域被认为是负值。



双曲函数:  $\sinh x = \frac{e^x - e^{-x}}{2}$ ,  $\cosh x = \frac{e^x + e^{-x}}{2}$ ,  $\tanh x = \frac{\sinh x}{\cosh x}$

反函数作为对数: 
$$\begin{cases} \sinh^{-1} x = \ln(x + \sqrt{x^2 + 1}) \\ \cosh^{-1} x = \ln(x + \sqrt{x^2 - 1}) \\ \tanh^{-1} x = \ln \frac{\sqrt{1-x^2} + 1}{1-x} = \frac{1}{2} \ln \frac{1+x}{1-x} \end{cases}$$

总和 (和差异) 公式 “sinh” and “cosh”.

$$\cosh(x + y) = \cosh x \cdot \cosh y + \sinh x \cdot \sinh y$$

$$\sinh(x + y) = \sinh x \cdot \cosh y + \cosh x \cdot \sinh y$$



$$x_2 = x_1 \cosh \theta + y_1 \sinh \theta = (x_1 + y_1 \tanh \theta) \cosh \theta$$

$$y_2 = y_1 \cosh \theta + x_1 \sinh \theta = (y_1 + x_1 \tanh \theta) \cosh \theta$$



$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \cosh \theta \cdot \begin{bmatrix} 1 & \tanh \theta \\ \tanh \theta & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

( Rotation matrix )

图 18.4.2.1 双曲型坐标系(CORDIC: m=-1).

**17.4.3 CORDIC 转换实现**

CORDIC 算法可以被实现作为加/减和移位运算，它们以固定的旋转角度旋转但具有可变的旋转方向。然而，CORDIC 迭代不是一个完美的旋转，它会涉及与切线的乘法运算(sine and cosine)。

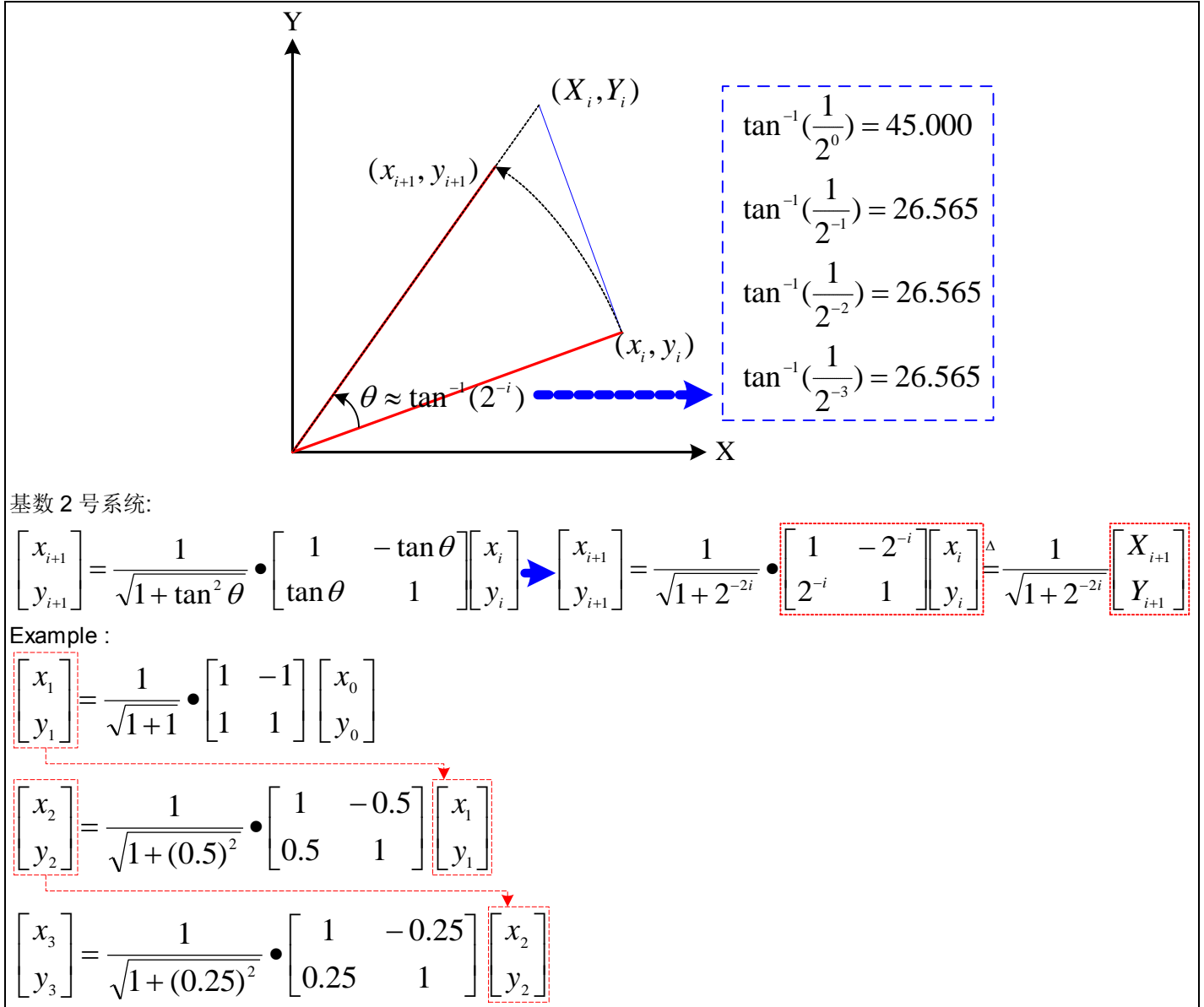


图 18.4.3.1 CORDIC 的改变和增加实现。

**17.4.4 广义 CORDIC 算法**

CORDIC 可以被修改来计算圆形和双曲线函数，并将 CORDIC 算法重新表达为一个广义以及统一的形式和适合在圆形，双曲线和直线坐标系中进行旋转。统一公式包括一个新的变量  $m$ ，为不同的坐标系分配不同的值。广义 CORDIC 如果制定如下：

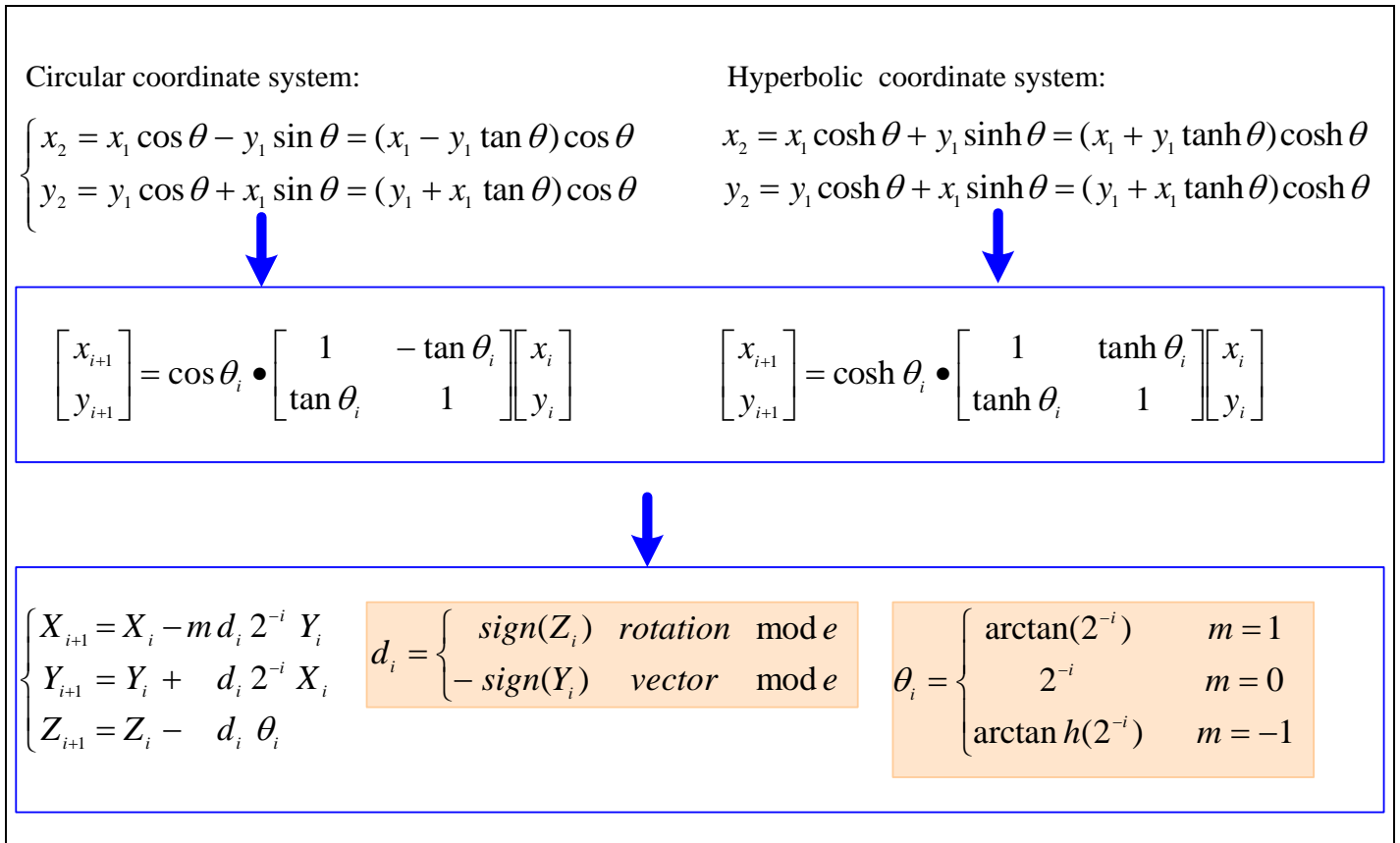


图 18.4.4.1 广义 CORDIC 算法.

**17.4.5 CORDIC 操作模式和功能**

使用 CORDIC 算法和上述的移位序列，可以在旋转模式和矢量模式下计算多种不同的函数，如下所示。

表 18.4.5.1 CORDIC 移位序列

坐标系	移位顺序	比例因子
M	i	$K_m (n \rightarrow \infty)$
1	0,1,2,3,4,...,i,...	~ 1.64676
0	1,2,3,4,...,i,...	1.0
-1	1,2,3,4,4,5,...,12,13,13,14..i,... ( $i \in \{4,13,\dots,3k+1\}$ repeat)	~ 0.82816

表 18.4.5.2 功能藉由 CORDIC 算法来计算

CSYS (m)	MODE	PARASEL	SFR 设置	结果	其它	
01 (circular) (m=1)	0 (rotation)	0 (Disable)	$x_0 = x$ USER define	$x_n = K_1(n) \cdot (x \cos\theta - y \sin\theta)$		
			$y_0 = y$ USER define	$y_n = K_1(n) \cdot (y \cos\theta + x \sin\theta)$		
			$z_0 = \theta$ USER define	$z_n = 0$		
		1 (Enable)	$x_0 = \frac{1}{K_1(n)}$ 0.607 (HW)	$x_n = \cos\theta$		$\tan Z = \frac{\sin Z}{\cos Z}$
			$y_0 = 0$ 0 (HW)	$y_n = \sin\theta$		
			$z_0 = \theta$ -90 < USER < 450	$z_n = 0$		
01 (circular) (m=1)	1 (vector)	0 (Disable)	$x_0 = x$ USER define	$x_n = K_1(n) \cdot \text{sign}(x_0) \cdot \sqrt{x^2 + y^2}$	$\cos^{-1}W = \tan^{-1} \left[ \frac{\sqrt{1-w^2}}{W} \right]$	
			$y_0 = y$ USER define	$y_n = 0$		$\sin^{-1}W = \tan^{-1} \left[ \frac{W}{\sqrt{1-w^2}} \right]$
			$z_0 = \theta$ USER define	$z_n = \theta + \tan^{-1} \left( \frac{y}{x} \right)$		
		1 (Enable)	$x_0 = x$ 1 or -1 (HW)		$y_n = 0$	
			$y_0 = y$ USER < **			
			$z_0 = 0$ 0 (HW)	$z_n = \tan^{-1}(y) \text{ or } \tan^{-1}(-y)$		
00	0 (rotation)		$x_0 = x$	$x_n = x$	For multiplication, set y = 0 note : no implement	
			$y_0 = y$	$y_n = y + x \cdot z$		
			$z_0 = z$	$z_n = 0$		
00	1 (vector)		$x_0 = x$	$x_n = x$	For division, set z = 0 note : no implement	
			$y_0 = y$	$y_n = 0$		
			$z_0 = z$	$z_n = z + \frac{y}{x}$		

表 18.4.5.3 CORDIC 算法的其他功能

CSYS (m)	MODE	PARASEL	SFR 设置	结果	其它
11 (Hyperbolic) (m=-1)	0 (rotation)	0 (Disable)	$x_0 = x$ <b>USER</b> define	$x_n = K_{-1}(n) \cdot (x \cosh\theta - y \sinh\theta)$	$\tanh Z = \frac{\sinh Z}{\cosh Z}$ $e^z = \sinh Z + \cosh Z$ $W^t = e^{t \cdot \ln W}$
			$y_0 = y$ <b>USER</b> define	$y_n = K_{-1}(n) \cdot (y \cosh\theta + x \sinh\theta)$	
			$z_0 = \theta$ <b>USER</b> define	$z_n = 0$	
		1 (Enable)	$x_0 = \frac{1.717}{K_{-1}(n)}$ <b>0</b> (HW)	$x_n = \cosh\theta$	
			$y_0 = 0$ <b>0</b> (HW)	$y_n = \sinh\theta$	
			$z_0 = \theta$ <b>-1.11 &lt;</b> <b>USER</b> <b>&lt; 1.11</b>	$z_n = 0$	
11 (Hyperbolic) (m=-1)	1 (vector)	0 (Disable)	$x_0 = x$ <b>USER</b> define	$x_n = K_{-1}(n) \cdot \text{sign}(x_0) \sqrt{x^2 - y^2}$	$\ln W = 2 \tanh^{-1} \left  \frac{W-1}{W+1} \right $ $\sqrt{W} = \sqrt{(W+1/4)^2 - (W-1/4)^2}$ $\cosh^{-1}(W) = \ln(W + \sqrt{1^2 - W^2})$ $\sinh^{-1}(W) = \ln(W + \sqrt{1^2 + W^2})$
			$y_0 = y$ <b>USER</b> define	$y_n = 0$	
			$z_0 = \theta$ <b>USER</b> define	$z_n = \theta + \tanh^{-1}\left(\frac{y}{x}\right)$	
		1 (Enable)	$x_0 = x$ <b>1</b> (HW)		
			$y_0 = y$ <b>-0.8 &lt;</b> <b>USER</b> <b>&lt; 0.8</b>	$y_n = 0$	
			$z_0 = \theta$ <b>0</b> (HW)	$z_n = \tanh^{-1}(y)$	

表 18.4.5.4 PARASEL = 1, CORDIC\_X / Y / Z 参数和用户设定值

CSYS	MODE	ROTATION			VECTOR		
		X	Y	Z	X	Y	Z
Circular coordinate system (01)		0.607	0	-90 < USER < 450	1 or -1	-*** < USER < ***	0
Hyperbolic coordinate system (11)		1.717	0	-1.11 < USER < 1.11	1	-0.8 < USER < 0.8	0

注意 : 0x011E\_2891 (~ 1.1), 0xFEE1\_D76F (~ -1.1), 0x0000\_0000 ( 0 ), 0x0100\_0000 ( 1 ), 0xFF00\_0000 ( -1 )

表 18.4.5.5 CORDIC 算法的其他功能

模式	0 (rotation)					1 (vector)				
	SFR 设置			结果		SFR 设置			结果	
CSYS (m)	$x_0$	$y_0$	$z_0$	$x_n$	$y_n$ or $z_n$	$x_0$	$y_0$	$z_0$	$x_n$	$y_n$ or $z_n$
01 (m=1) (circular)	1	0	$\theta$	$\cos\theta$	$y_n = \sin\theta$	1	a	$\pi/2$	$\sqrt{a^2 + 1}$	$z_n = \cot^{-1}(a)$
11 (m=-1) (Hyperbolic)	1	0	$\theta$	$\cosh\theta$	$y_n = \sinh\theta$	a	1	0	$\sqrt{a^2 - 1}$	$z_n = \coth^{-1}(a)$
	A	a	$\theta$	$ae^\theta$	$y_n = ae^\theta$	a+1	a-1	0	$\sqrt{a}$	$z_n = 0.5 \cdot \ln(a)$
						$a + \frac{1}{4}$	$a - \frac{1}{4}$	0	$\sqrt{a}$	$z_n = \ln\left(\frac{1}{4}a\right)$
						a+b	a-b	0	$2\sqrt{ab}$	$z_n = 0.5 \cdot \ln\left(\frac{a}{b}\right)$

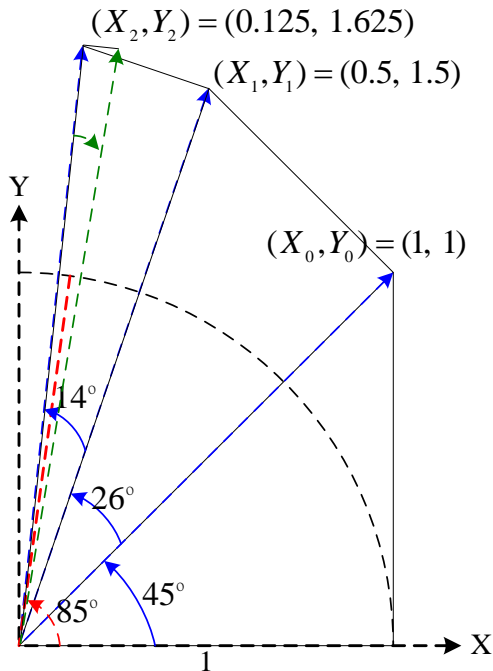


## 17.4.6 例 1：圆形旋转

 例： $\sin 85^\circ$ ,  $\cos 85^\circ$ ,  $\tan 85^\circ$ ?

$$X_{i+1} = \cos \theta [X_i - \tan \theta \cdot Y_i]$$

$$Y_{i+1} = \cos \theta [Y_i + \tan \theta \cdot X_i]$$



$$X_0 = \cos 45.000 \times [X - Y \tan 45.000] = 1.000$$

$$Y_0 = \cos 45.000 \times [Y + X \tan 45.000] = 1.000$$

$$X_1 = \cos 26.565 \times [X_0 - Y_0 \tan 26.565] = 0.500$$

$$Y_1 = \cos 26.565 \times [Y_0 + X_0 \tan 26.565] = 1.500$$

$$X_2 = \cos 14.036 \times [X_1 - Y_1 \tan 14.036] = 0.125$$

$$Y_2 = \cos 14.036 \times [Y_1 + X_1 \tan 14.036] = 1.625$$

$$X_3 = \cos 7.125 \times [X_2 + Y_2 \tan 7.125] = 0.328$$

$$Y_3 = \cos 7.125 \times [Y_2 - X_2 \tan 7.125] = 1.609$$

$$\prod_{i=1}^n \cos(m^{1/2}\theta_i) \sim 1.64676$$

图 18.4.6.1 例 1: 圆形旋转.

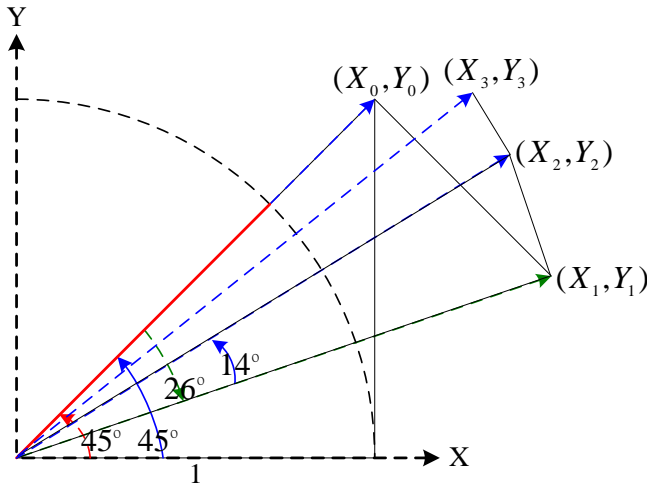
步骤	X 值	Y 值	Z 值
设置	X = 1.000	Y = 0.000	Z = 0.000 < 85
0	$X_0 = X - \tan 45.000 \cdot Y = 1.000$	$Y_0 = Y + \tan 45.000 \cdot X = 1.000$	$Z_0 = Z + 45.000 = 45.000 < 85$
1	$X_1 = X_0 - \tan 26.565 \cdot Y_0 = 0.500$	$Y_1 = Y_0 + \tan 26.565 \cdot X_0 = 1.500$	$Z_1 = Z_0 + 26.565 = 71.565 < 85$
2	$X_2 = X_1 - \tan 14.036 \cdot Y_1 = 0.125$	$Y_2 = Y_1 + \tan 14.036 \cdot X_1 = 1.625$	$Z_2 = Z_1 + 14.036 = 85.601 > 85$
3	$X_3 = X_2 + \tan 7.125 \cdot Y_2 = 0.328$	$Y_3 = Y_2 - \tan 7.125 \cdot X_2 = 1.609$	$Z_3 = Z_2 - 7.125 = 78.476 < 85$
4	$X_4 = X_3 - \tan 3.576 \cdot Y_3 = 0.227$	$Y_4 = Y_3 + \tan 3.576 \cdot X_3 = 1.629$	$Z_4 = Z_3 + 3.576 = 82.052 < 85$
5	$X_5 = X_4 - \tan 1.789 \cdot Y_4 = 0.176$	$Y_5 = Y_4 + \tan 1.789 \cdot X_4 = 1.637$	$Z_5 = Z_4 + 1.789 = 83.842 < 85$
6	$X_6 = X_5 - \tan 0.895 \cdot Y_5 = 0.151$	$Y_6 = Y_5 - \tan 0.895 \cdot X_5 = 1.639$	$Z_6 = Z_5 + 0.895 = 84.737 < 85$

## 17.4.7 例 2：圆形旋转

 Example:  $\sin 45^\circ$ ,  $\cos 45^\circ$ ,  $\tan 45^\circ$ ?

$$X_{i+1} = \cos \theta [X_i - \tan \theta \cdot Y_i]$$

$$Y_{i+1} = \cos \theta [Y_i + \tan \theta \cdot X_i]$$



$$\begin{aligned} X_0 &= \cos 45.000 \times [X - Y \tan 45.000] = 1.000 \\ Y_0 &= \cos 45.000 \times [Y + X \tan 45.000] = 1.000 \\ X_1 &= \cos 26.565 \times [X_0 + Y_0 \tan 26.565] = 1.500 \\ Y_1 &= \cos 26.565 \times [Y_0 - X_0 \tan 26.565] = 0.500 \\ X_2 &= \cos 14.036 \times [X_1 - Y_1 \tan 14.036] = 1.375 \\ Y_2 &= \cos 14.036 \times [Y_1 + X_1 \tan 14.036] = 0.875 \\ X_3 &= \cos 7.125 \times [X_2 - Y_2 \tan 7.125] = 1.265 \\ Y_3 &= \cos 7.125 \times [Y_2 + X_2 \tan 7.125] = 1.046 \end{aligned}$$

$$\prod_{i=1}^n \cos(m^{1/2} \theta_i) \sim 1.64676$$

图 18.4.7.1 例 2: 圆形旋转.

Step	X 值	Y 值	Z 值
Set	X = 1.000	Y = 0.000	Z = 0.000 < 45
0	X0 = X - tan45.000 * Y = 1.000	Y0 = Y + tan45.000 * X = 1.000	Z0 = Z + 45.000 = 45.000 >= 45
1	X1 = X0 + tan26.565 * Y0 = 1.500	Y1 = Y0 - tan26.565 * X0 = 0.500	Z1 = Z0 - 26.565 = 18.439 < 45
2	X2 = X1 - tan14.036 * Y1 = 1.375	Y2 = Y1 + tan14.036 * X1 = 0.875	Z2 = Z1 + 14.036 = 32.471 < 45
3	X3 = X2 - tan7.125 * Y2 = 1.265	Y3 = Y2 + tan7.125 * X2 = 1.046	Z3 = Z2 + 7.125 = 39.596 < 45
4	X4 = X3 - tan3.576 * Y3 = 1.200	Y4 = Y3 + tan3.576 * X3 = 1.126	Z4 = Z3 + 3.576 = 43.172 < 45
5	X5 = X4 - tan1.789 * Y4 = 1.165	Y5 = Y4 + tan1.789 * X4 = 1.163	Z5 = Z4 + 1.789 = 44.962 < 45
6	X6 = X5 - tan0.895 * Y5 = 1.146	Y6 = Y5 + tan0.895 * X4 = 1.181	Z6 = Z5 + 0.895 = 45.857 > 45
...	...	...	...
9	X9 = X8 - tan0.895 * Y8 = 1.1629	Y9 = Y8 + tan0.895 * X8 = 1.1659	Z9 = 45.074

$$\sin(45^\circ) \approx 0.707106 \Leftrightarrow \frac{1.1629}{1.64676} \approx 0.70617$$

## 17.4.8例 1：圆矢量

 Example:  $x = 1, y = 1, \theta?$ 

$$X_{i+1} = \cos \theta [X_i - \tan \theta \cdot Y_i]$$

$$Y_{i+1} = \cos \theta [Y_i + \tan \theta \cdot X_i]$$

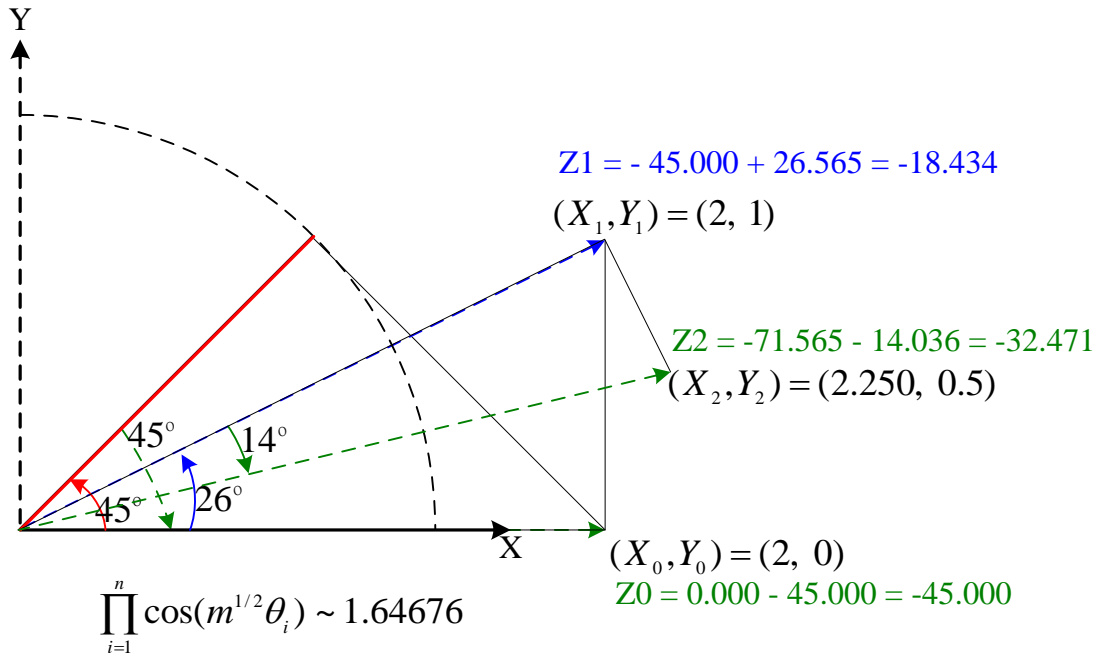


图 18.4.8.1 例 1: 圆矢量

Step	X 值	Y 值	Z 值
Set	$X = 1.000$	$Y = 1.000$	$Z = 0.000$
0	$X_0 = X + \tan 45.000 \cdot Y = 2.000$	$Y_0 = Y - \tan 45.000 \cdot X = 0.000$	$Z_0 = Z - 45.000 = -45.000$
1	$X_1 = X_0 - \tan 26.565 \cdot Y_0 = 2.000$	$Y_1 = Y_0 + \tan 26.565 \cdot X_0 = 1.000$	$Z_1 = Z_0 + 26.565 = -18.434$
2	$X_2 = X_1 + \tan 14.036 \cdot Y_1 = 2.250$	$Y_2 = Y_1 - \tan 14.036 \cdot X_1 = 0.500$	$Z_2 = Z_1 - 14.036 = -32.471$
3	$X_3 = X_2 + \tan 7.125 \cdot Y_2 = 2.312$	$Y_3 = Y_2 - \tan 7.125 \cdot X_2 = 0.218$	$Z_3 = Z_2 - 7.125 = -39.596$
4	$X_4 = X_3 + \tan 3.576 \cdot Y_3 = 2.326$	$Y_4 = Y_3 - \tan 3.576 \cdot X_3 = 0.074$	$Z_4 = Z_3 - 3.576 = -43.172$
5	$X_5 = X_4 + \tan 1.789 \cdot Y_4 = 2.328$	$Y_5 = Y_4 - \tan 1.789 \cdot X_4 = 0.001$	$Z_5 = Z_4 - 1.789 = -44.962$
6	$X_6 = X_5 + \tan 0.895 \cdot Y_5 = 2.328$	$Y_6 = Y_5 - \tan 0.895 \cdot X_5 = -0.034$	$Z_6 = Z_5 - 0.895 = -45.857$
...	...	...	...
9	$X_9 = 2.328$	$Y_9 = -0.003$	$Z_9 = -45.074$

$$\sqrt{x^2 + y^2} \cdot 1.64676 = \sqrt{2} \cdot 1.64676 \approx 2.32887$$

## 17.4.9例 2：圆矢量

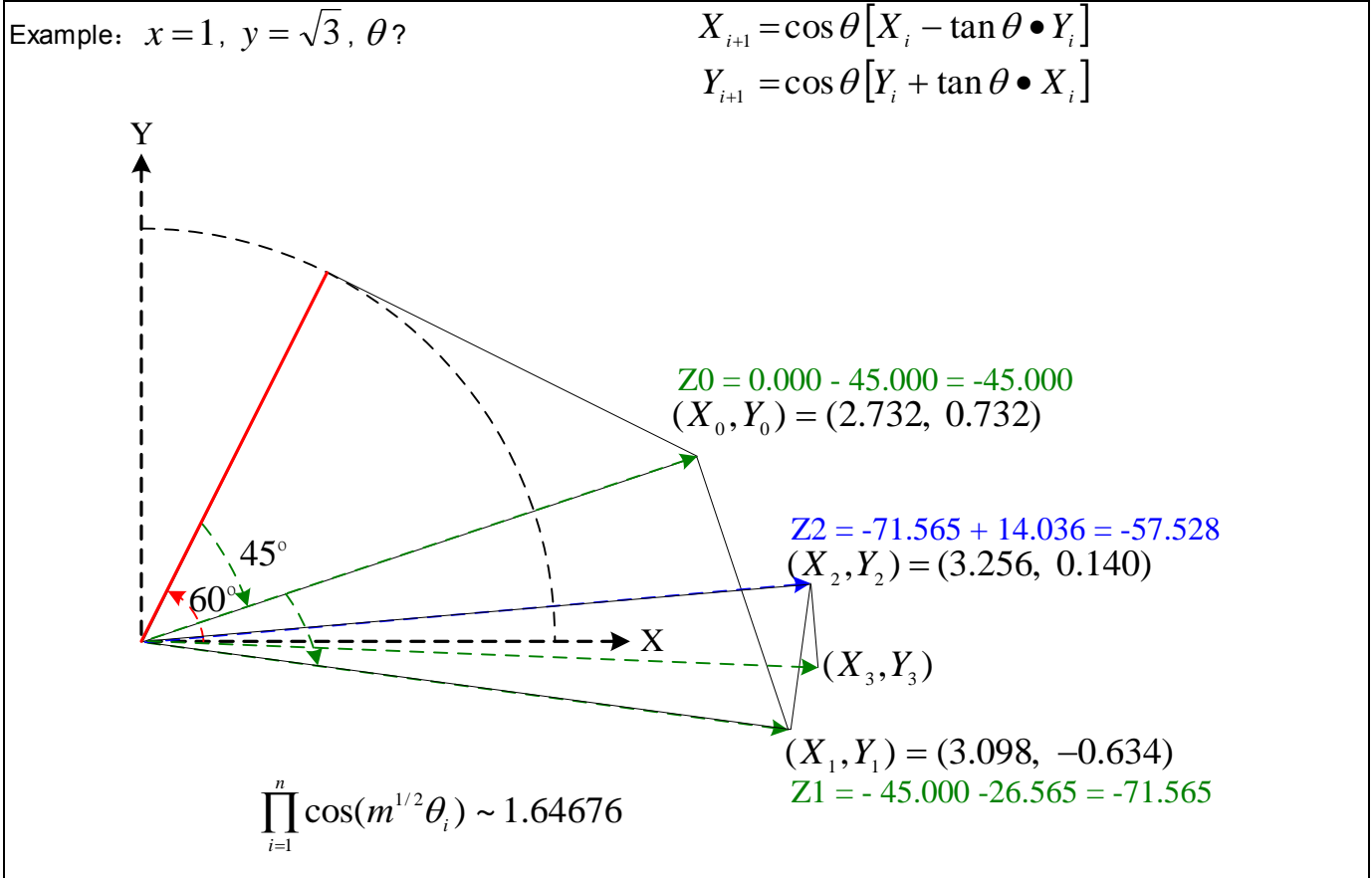


图 18.4.9.1 例 2: Circular Vector.

Step	X 值	Y 值	Z 值
Set	X = 1.000	Y = 1.732	Z = 0.000
0	X0 = X + tan45.000 * Y = 2.732	Y0 = Y - tan45.000 * X = 0.732	Z0 = Z - 45.000 = -45.000
1	X1 = X0 + tan26.565 * Y0 = 3.098	Y1 = Y0 - tan26.565 * X0 = -0.634	Z1 = Z0 - 26.565 = -71.565
2	X2 = X1 - tan14.036 * Y1 = 3.256	Y2 = Y1 + tan14.036 * X1 = 0.140	Z2 = Z1 + 14.036 = -57.528
3	X3 = X2 + tan7.125 * Y2 = 3.274	Y3 = Y2 - tan7.125 * X2 = -0.266	Z3 = Z2 - 7.125 = -64.653
4	X4 = X3 - tan3.576 * Y3 = 3.290	Y4 = Y3 + tan3.576 * X3 = -0.061	Z4 = Z3 + 3.576 = -61.077
5	X5 = X4 + tan1.789 * Y4 = 3.292	Y5 = Y4 + tan1.789 * X4 = 0.040	Z5 = Z4 + 1.789 = -59.287
6	X6 = X5 + tan0.895 * Y5 = 3.293	Y6 = Y5 - tan0.895 * X5 = -0.010	Z6 = Z5 - 0.895 = -60.182

$$\sqrt{x^2 + y^2} \cdot 1.64676 = \sqrt{4} \cdot 1.64676 \approx 3.29352$$

**17.4.10 CORDIC 圆形旋转角度**

CORDIC 圆形旋转角度:

i	$2^{-i}$	$\arctan(2^{-i}) * 360 / 2\pi$	$45 * 2^{-i}$
0	1	45*	
1	0.5	26.56505118*	4.065051177*
2	0.25	14.03624347	0.753717879
3	0.125	7.125016349	0.106894615
4	0.0625	3.576334375	0.013826201
5	0.03125	1.789910608	0.001743421
6	0.015625	0.895173710	0.000218406
7	0.0078125	0.447614171	2.73158E-05
8	0.00390625	0.2238105	3.41494E-06
9	0.001953125	0.111905677	4.26882E-07
10	0.000976563	0.055952892	5.33607E-08
11	0.000488281	0.027976453	6.67010E-09
12	0.000244141	0.013988227	8.33763E-10
13	0.00012207	0.006994114	1.04220E-10
14	6.10352E-05	0.003497057	1.30276E-11
15	3.05176E-05	0.001748528	1.62844E-12
16	1.52588E-05	0.000874264	2.03555E-13
17	7.62939E-06	0.000437132	2.54444E-14
18	3.81470E-06	0.000218566	3.18056E-15
19	1.90735E-06	0.000109283	3.97577E-16
20	9.53674E-07	5.46415E-05	4.96903E-17

\* 4.06505 = 26.565 - 45/2

常用的 CORDIC 的初级角度设置

迭代索引	基本角度	半径值
i = 0	$\tan^{-1}(2^{-0})$ 45	0.785398163
i = 1	$\tan^{-1}(2^{-1})$ 26.56505118	0.463647609
i = 2	$\tan^{-1}(2^{-2})$ 14.03624347	0.244978663
i = 3	$\tan^{-1}(2^{-3})$ 7.125016349	0.124354994
i = 4	$\tan^{-1}(2^{-4})$ 3.576334375	0.062418809
i = 5	$\tan^{-1}(2^{-5})$ 1.789910608	0.031239833
i = 6	$\tan^{-1}(2^{-6})$ 0.895173710	0.015623728
i = 7	$\tan^{-1}(2^{-7})$ 0.447614171	0.007812341
i = 8	$\tan^{-1}(2^{-8})$ 0.2238105	0.003906230
i = 9	$\tan^{-1}(2^{-9})$ 0.111905677	0.001953122
i = 10	$\tan^{-1}(2^{-10})$ 0.055952892	0.000976562
i = 11	$\tan^{-1}(2^{-11})$ 0.027976453	0.000488281
i = 12	$\tan^{-1}(2^{-12})$ 0.013988227	0.000244140
i = 13	$\tan^{-1}(2^{-13})$ 0.006994114	0.000122070
i = 14	$\tan^{-1}(2^{-14})$ 0.003497057	6.10351E-05
i = 15	$\tan^{-1}(2^{-15})$ 0.001748528	3.05175E-05
i = 16	$\tan^{-1}(2^{-16})$ 0.000874264	1.52587E-05
i = 17	$\tan^{-1}(2^{-17})$ 0.000437132	7.62939E-06
i = 18	$\tan^{-1}(2^{-18})$ 0.000218566	3.81469E-06
i = 19	$\tan^{-1}(2^{-19})$ 0.000109283	1.90734E-06

迭代索引		
i = 0	$\tanh^{-1}(2^{-0})$	Inf
i = 1	$\tanh^{-1}(2^{-1})$	0.549306144
i = 2	$\tanh^{-1}(2^{-2})$	0.255412811
i = 3	$\tanh^{-1}(2^{-3})$	0.125657214
i = 4	$\tanh^{-1}(2^{-4})$	0.062581571
i = 5	$\tanh^{-1}(2^{-5})$	0.031260178
i = 6	$\tanh^{-1}(2^{-6})$	0.015626271
i = 7	$\tanh^{-1}(2^{-7})$	0.007812658
i = 8	$\tanh^{-1}(2^{-8})$	0.003906269
i = 9	$\tanh^{-1}(2^{-9})$	0.001953127
i = 10	$\tanh^{-1}(2^{-10})$	0.000976562
i = 11	$\tanh^{-1}(2^{-11})$	0.000488281
i = 12	$\tanh^{-1}(2^{-12})$	0.000244140
i = 13	$\tanh^{-1}(2^{-13})$	0.000122070
i = 14	$\tanh^{-1}(2^{-14})$	6.10351E-05
i = 15	$\tanh^{-1}(2^{-15})$	3.05175E-05
i = 16	$\tanh^{-1}(2^{-16})$	1.52587E-05
i = 17	$\tanh^{-1}(2^{-17})$	7.62939E-06
i = 18	$\tanh^{-1}(2^{-18})$	3.81469E-06
i = 19	$\tanh^{-1}(2^{-19})$	1.90734E-06

### 17.5 寄存器说明

CORDIC包含寄存器显示在表18.5.1 更详细的描述如下.

表 18.5.1 寄存器概述(基址 0x5001 4000)

Name	Access	Address offset	Description	Reset value
CORDIC_CTR	R/W	0x000		0
-	-	0x004 - 0x00C	保留	
CORDIC_X	R/W	0x010	输入/输出格式为Q20, 请参见表18.5.2.1	
CORDIC_Y	R/W	0x014	输入/输出格式为Q20, 请参见表18.5.2.1	
CORDIC_Z	R/W	0x018	输入/输出格式为Q20, 请参见表18.5.2.1	
-	-	0x01C-0x03C	保留	

表 18.5.2 寄存器概述: CORDIC 参数值(基址 0x5001 4000)

Name	Address offset	Description	Reset value
ARCTAN_ANGLE_0	0x040	45	32'h02D0_0000
ARCTAN_ANGLE_1	0x044	26.5650511770780	32'h01A9_0A73
ARCTAN_ANGLE_2	0x048	14.0362434679265	32'h00E0_9474
ARCTAN_ANGLE_3	0x04C	7.12501634890180	32'h0072_0011
ARCTAN_ANGLE_4	0x050	3.57633437499735	32'h0039_38AA
ARCTAN_ANGLE_5	0x054	1.78991060824607	32'h001C_A379
ARCTAN_ANGLE_6	0x058	0.895173710211074	32'h000E_52A2
ARCTAN_ANGLE_7	0x05C	0.447614170860553	32'h0007_296D
ARCTAN_ANGLE_8	0x060	0.223810500368538	32'h0003_94BA
ARCTAN_ANGLE_9	0x064	0.111905677066207	32'h0001_CA5E
ARCTAN_ANGLE_A	0x068	0.0559528918938037	32'h0000_E52F
ARCTAN_ANGLE_B	0x06C	0.0279764526170037	32'h0000_7297
ARCTAN_ANGLE_C	0x070	0.0139882271422650	32'h0000_394C
ARCTAN_ANGLE_D	0x074	0.00699411367535292	32'h0000_1CA6
ARCTAN_ANGLE_E	0x078	0.00349705685070401	32'h0000_0E53
ARCTAN_ANGLE_F	0x07C	0.00174852842698045	32'h0000_0729



表 18.5.3 寄存器概述: CORDIC 参数值 (基址 0x5001 4000)

Name	Address offset	Description	Reset value
ARCTAN_RADIUS_0	0x080	0.785398163397448	32'h000C_90FE
ARCTAN_RADIUS_1	0x084	0.463647609000806	32'h0007_6B1A
ARCTAN_RADIUS_2	0x088	0.244978663126864	32'h0003_EB6F
ARCTAN_RADIUS_3	0x08C	0.124354994546761	32'h0001_FD5C
ARCTAN_RADIUS_4	0x090	0.0624188099959574	32'h0000_FFAB
ARCTAN_RADIUS_5	0x094	0.0312398334302683	32'h0000_7FF5
ARCTAN_RADIUS_6	0x098	0.0156237286204768	32'h0000_3FFF
ARCTAN_RADIUS_7	0x09C	0.00781234106010111	32'h0000_2000
ARCTAN_RADIUS_8	0x0A0	0.00390623013196697	32'h0000_1000
ARCTAN_RADIUS_9	0x0A4	0.00195312251647882	32'h0000_0800
ARCTAN_RADIUS_A	0x0A8	0.000976562189559320	32'h0000_0400
ARCTAN_RADIUS_B	0x0AC	0.000488281211194898	32'h0000_0200
ARCTAN_RADIUS_C	0x0B0	0.000244140620149362	32'h0000_0100
ARCTAN_RADIUS_D	0x0B4	0.000122070311893670	32'h0000_0080
ARCTAN_RADIUS_E	0x0B8	0.000061035156174208	32'h0000_0040
ARCTAN_RADIUS_F	0x0BC	0.000030517578115526	32'h0000_0020
Name	Address offset	Description	Reset value
ARCTANH_0	0x0C0	Infinity	32'h0000_0000
ARCTANH_1	0x0C4	0.549306144334055	32'h0008_C9F5
ARCTANH_2	0x0C8	0.255412811882995	32'h0004_162C
ARCTANH_3	0x0CC	0.125657214140453	32'h0002_02B1
ARCTANH_4	0x0D0	0.0625815714770030	32'h0001_0056
ARCTANH_5	0x0D4	0.0312601784906670	32'h0000_800B
ARCTANH_6	0x0D8	0.0156262717520522	32'h0000_4001
ARCTANH_7	0x0DC	0.00781265895154042	32'h0000_2000
ARCTANH_8	0x0E0	0.00390626986839683	32'h0000_1000
ARCTANH_9	0x0E4	0.00195312748353255	32'h0000_0800
ARCTANH_A	0x0E8	0.000976562810441036	32'h0000_0400
ARCTANH_B	0x0EC	0.000488281288805113	32'h0000_0200
ARCTANH_C	0x0F0	0.000244140629850639	32'h0000_0100
ARCTANH_D	0x0F4	0.000122070313106330	32'h0000_0080
ARCTANH_E	0x0F8	0.000061035156325791	32'h0000_0040
ARCTANH_F	0x0FC	0.000030517578134473	32'h0000_0020
Name	Address offset	Description	Reset value

CIRCULAR_K	0x100	0.607252935009250	32'h0009_B74E
CIRCULAR_K_INV	0x104	1.64676025812007	32'h001A_5921
HYPERBOLIC_K	0x108	1.717329163039773	32'h001B_7A2E
HYPERBOLIC_K_INV	0x10C	0.582299550675505	32'h0009_5119

### 17.5.1 CORDIC 控制寄存器(CORDIC\_CTR – 0x5001 4000)

参数的对应请参考表 18.4.5.2

表 18.5.1.1 CORDIC 控制寄存器(CORDIC\_CTR - 0x5001 4000) bit description

Bit	Symbol	Value	Description	Reset value
0	TRIG		CORDIC 功能触发器	0
		0	CORDIC 功能禁用或准备就绪（硬件清除）	
		1	CORDIC 功能使能（韧带触发）	
1	MODE		CORDIC 算法有两种类型的计算模式矢量旋转和矢量平移。	0
		0	旋转模式	
		1	向量	
3:2	CSYS		CORDIC 算法在圆形，线性和双曲线坐标系中执行旋转。统一公式包括一个新的变量 $m$ ，它对于不同的坐标系分配不同的值	0
		00	保留. ( $m=0$ )	
		01	圆形坐标系. ( $m=1$ )	
		10	保留	
		11	双曲线坐标系. ( $m=-1$ )	
4	PARASEL		参数选择	0
		0	参数选择禁用	
		1	数选择使能	
5	CCPSEL		循环参数选择	
		0	角度	
		1	半径( $\pi=3.1415926$ )	
23:6	-	-	保留。	
27:24	ITCNT		迭代计数，默认值为 8。数值设定越大，表示精准。更改此数值，[31:28] 值须为 0x9。8 ITCNT 寄存器不可读取，只写。	
31:28	ITCNT_KEY		迭代计数键	

### 17.5.2 CORDIC 数据格式寄存器(CORDIC\_X/Y/Z – 0x5001 4010/14/18)

转换前当作"输入参数"寄存器, 转换后当作"输出结果"寄存器

输入/输出的格式为 Q20

表 18.5.2.1 CORDIC 数据格式寄存器 (CORDIC\_X/Y/Z - 0x5001 4010/14/18) bit description

Bit	符号	值	描述	重置值
30:0	CORDIC_*		若有关于设定或求值，MSB 11 bit 定义为整数。	0
31	SIGN_*		符号位尽可能简单. 反转这个位值反转数字的符号。	0
		0	表示一个正数	
		1	表示一个负数	

## 18. 乘法除法单元(MDU)

### 18.1 概观

MDU - 乘法除法单元是一个芯片上算术协处理器，它使能这 CPU 能够执行额外的扩展算术运算。该单元提供 32 位除法，32 位乘法，移位和规范操作。所有操作都是无符号整数操作

算术单元允许目前的执行操作并独立的 CPU 的活动。操作和结果存储在“MD0”...“MD2”寄存器中。该模块由“ARCON”寄存器控制。任何 MDU 的计算都会改写它的操作。

### 18.2 MDU 操作说明

MDU 的操作由三个阶段组成:

#### 18.2.1 装载 MDx 寄存器

表 19.2.1.1 DU 寄存器写序列

运算	16 位/ 16 位	32 位/ 16 位	32 位/ 32 位	32 位 x 32 位	移动标准化
值	MD0 被除数	MD0 被除数	MD0 被除数	MD0 被乘数	MD0
	MD2 除数	MD2 除数	MD2 除数	MD2 乘数	
ARCON[3:0]	0000	0100	0101	0111	1111

根据写入“ARCON [3: 0]”寄存器的值来选择 MDU 必须执行的计算类型。

#### 18.2.2 执行计算

在执行操作期间，MDU 与 CPU 并行工作。

表 19.2.2.1 MDU 操作执行时间

运算	时钟周期数	
除法 16 位/16 位	***时钟周期	
除法 32 位/16 位	***时钟周期	
除法 32 位/32 位	***时钟周期	
乘法	19 时钟周期	
移动	min 3 时钟周期( sc = 01h )	min 18 时钟周期( sc = 1Fh )
正常化	min 4 时钟周期( sc <- 01h )	min 19 时钟周期( sc <- 1Fh )

当状态机转到 16 位/ 16 位的 ST3 或 32 位/ 16 位的 ST4 时，分频计算开始。在完成除法后，FSM 转到 ST10 并存储结果。

#### 18.2.3 从 MDx 寄存器读取结果

表 19.2.3.1 MDU 寄存器写入序列

运算	16 位/ 16 位	32 位/ 16 位	32 位/ 32 位	32 位 x 32 位	移动标准化
值	MD0 商	MD0 商	MD0 商	MD0 积 LSB	MD0
	MD2 余数	MD2 余数	MD2 余数	MD1 积 MSB	

#### 18.2.4 标准化

存储在“MD0”寄存器中的所有 32 位整型变量的前导零将通过左移操作来移除。当“MD0”寄存器的 MSB（最高有效位）包含“1”时，整个操作完成。标准化后，位“ARCON.13”（MSB）...“ARCON.8”（LSB）包含左移操作的数量，这些操作已完成。

#### 18.2.5 移位

在移位操作中，存储在“md0”寄存器中的 32 位整数变量向左或向右移位指定的位数。“slr”位（“ARCON.5”）定义了移位方向，位“ARCON.13”...“ARCON.8”指定了移位计数（不能为 0）。在移位过程中，零点进入“MD0”的左端，用于右移或 MD 0 的右端用于左移

## 18.3 寄存器说明

MDU 包含表 19.3.1 所示的寄存器。更多详细说明如下:

表 19.3.1 寄存器概述: MDU (基地址 0x5001 4400(MDU0), 0x5001 4800(MDU1))

名称	存取	地址 偏移量	描述	重置 值
ARCON	R/W	0x000	算术控制寄存器	
MD0	R/W	0x004	乘法 / 除法寄存器	
MD1	R/W	0x008	乘法 / 除法寄存器	
MD2	R/W	0x00C	乘法 / 除法寄存器	

**18.3.1 算术控制寄存器(ARCON – 0x5001 4400(MDU0),0x5001 4800(MDU1))**

ARCON 寄存器控制 MDU 的运算, 并通知其当前状态

表 19.3.1.1 算术控制寄存器(ARCON - 0x5001 4400(MDU0), 0x5001 4800(MDU1))位描述

位	代号	值	描述	Reset value
3:0	MDUOP		MDU 运算选择枚举类型	
		0000	16 位/ 16 位	
		0100	32 位/ 16 位	
		0101	32 位/ 32 位	
		0111	32 位 x 32 位	
		1111	移位/正常化( only 32 bit )	
4	SIGNE		符号使能	
		0	无符号数字运算	
		1	符号数字运算	
5	SLR		移位方向	
		0	左移操作	
		1	右移操作	
6	MDOV		MDU 溢出标志 MDOV	
		0	在 MDU 运算中没有发生溢出.	
		1	MDU 运算中发生溢出	
7	MDEF		MDU 错误标志 MDEF	
		0	没有错误.	
		1	表示未正确执行的运算(当其中一个算术运算已被重新启动或被新运算中断时).	
13:8	SC		移位计数器 当设置为全 0 时, 选择标准化操作。 标准化后, “sc.0”...“sc.4”包含执行的标准化移位次数。 当这些位中, 至少一个被设置时, 选择高移位操作。 位移执行次数由写入“sc.4”...“sc.0”的编号确定, 其中“sc.4”是 MSB。	
14	-	-	保留	
15	TRIG		MDU 功能触发	0
		0	MDU 功能禁用或准备就绪 (硬件清除)	
		1	MDU 功能使能 (韧带触发)	
31:16	-	-	保留	

**18.3.2 乘法 / 除法寄存器((MD0/1/2 – 0x5001 4404/08/0C)(MDU0), 0x5001 4804/08/0C)(MDU1))**

MD0 ... MD5 是在 MDU 运算中使用的寄存器。

表 19.3.2.1 乘法/除法寄存器(MD0 - 0x5001 4404(MDU0),0x5001 4804(MDU1))位描述

位	代号	值	描述	Reset value
31:0	MD0		表示被乘数、被除数、位移与正规化。	

表 19.3.2.2 乘法/除法寄存器(MD1 - 0x5001 4408(MDU0),0x5001 4808(MDU1))位描述

位	代号	值	描述	Reset value
---	----	---	----	----------------

31:0 MD1                      32 位 x 32 位之积 MSB

表 19.3.2.3 乘法/除法寄存器(MD2 - 0x5001 440C(MDU0), 0x5001 480C(MDU1))位描述

位	代号	值	描述	Reset value
31:0	MD2		表示乘数、除数。	

MDU 功能	符号位的位置(输入)	符号位的位置(输出)
16 位/ 16 位	MD0 (15-位) & MD2 (15-位)	MD0 (15-位商) & MD2 (15-位余)
32 位/ 32 位	MD0 (31-位) & MD2 (31-位)	MD0 (31-位商) & MD2 (31-位余)
32 位/ 16 位	MD0 (31-位) & MD2 (15-位)	MD0 (31-位商) & MD2 (15-位余)
32 位 x 32 位	MD0 (31-位) & MD2 (31-位)	MD0 (31-位积) & MD1

## 19. 模拟数字转换器(ADC)

### 19.1 介绍

模数转换器 (ADC) 是将电压转换为数字的装置。

### 19.2 特征

ADC 模块提供以下功能:

- 9 个模拟输入通道
- 灵活的软硬件触发方式
  - 软件
  - PWM 事件
  - 模拟比较器
  - GPIO
- 对应的转换结果 FIFOs
- 最大 ADC 时钟频率为 72MHz ( Temp.= 25°C, VDD = 4V ~ 5.5V)

### 19.3 功能描述

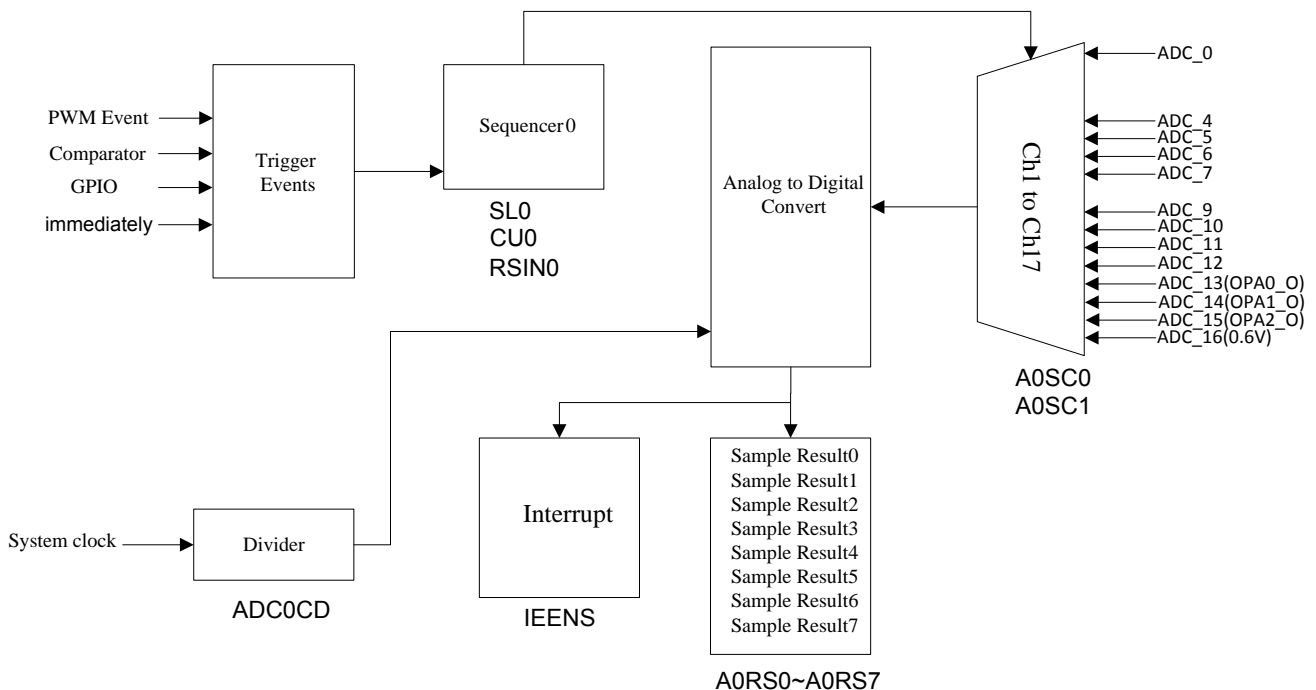


图 19.3.1 ADC 的框图

#### 19.3.1 校准功能

##### 自动校准:

当开启ADC时(ADCCON[1], EN0= 1, ADCCON - 0x5002 0000),硬件会自动做一次校准功能,必须等待

CALCON0[8] = 0 校准完成才可以开始使用ADC。

自动校准软件程序:

1. 设置 ADCCON[1] = 1
2. 等到 CALCON0[8] = 0, 即表示校准完成

可参考 CALCON0 - 0x5002 00A0 说明

### 19.3.2 ADC 时钟

ADC 时钟 = 系统时钟 / ADC0CD

可参考 ADC0CD - 0x5002 0090 说明

### 19.3.3 ADC 通道选择

1. 要转换的通道序列可在 A0SC0/ A0SC1 寄存器中编程设置。

可参考 A0SC0 - 0x5002 0004 / A0SC1 - 0x5002 0008 说明

2. 序列长度可在 SL0 寄存器中编程设置。

可参考 SL0 - 0x5002 000C 说明

3. 设置ADC通道13~15可连结至外部Pin of OPA0\_O~Pin of OPA2\_O方式, 以OPA0\_O为例

1. 设置GPIO 端口1多功能0寄存器(MF0 - 0x5000 403C) 将P1.0 设为OPA0\_O输出

2. 设置GPIO端口1模式寄存器(MODE - 0x5000 4008) 将P1.0 设为只输入模式

3. 设置ADC 0采样序列0寄存器(A0SC0 - 0x5002 0004) 将采样输入选择设为ADC通道13

### 19.3.4 开始转换

当发生软件或硬件触发事件时, ADC 模块会开始执行一系列转换程序

- 软件开始对ADC转换时, 是藉由ADC模块0来设置(ADCCON[0] = 0x0001, START0= 1, ADCCON - 0x5002 0000)
- ADCCON[0], START0位若由硬件清零, 表示ADC已完成。

### 19.3.5 数据管理

A0SC0 与 A0SC1 寄存器可以选择 ADC 转换完成后的动作

当 SCxM 选择“继续”时，ADC 将继续进行转换。

当 SCxM 选择“停止”时，ADC 停止进行转换并等待下一个触发。

可参考 A0SC0 - 0x5002 0004 / A0SC1 - 0x5002 0008 说明

重置模式( RSIN0[31] = 1)

在每次转换结束时，转换数据的结果存储在 12 位宽的内部寄存器中。



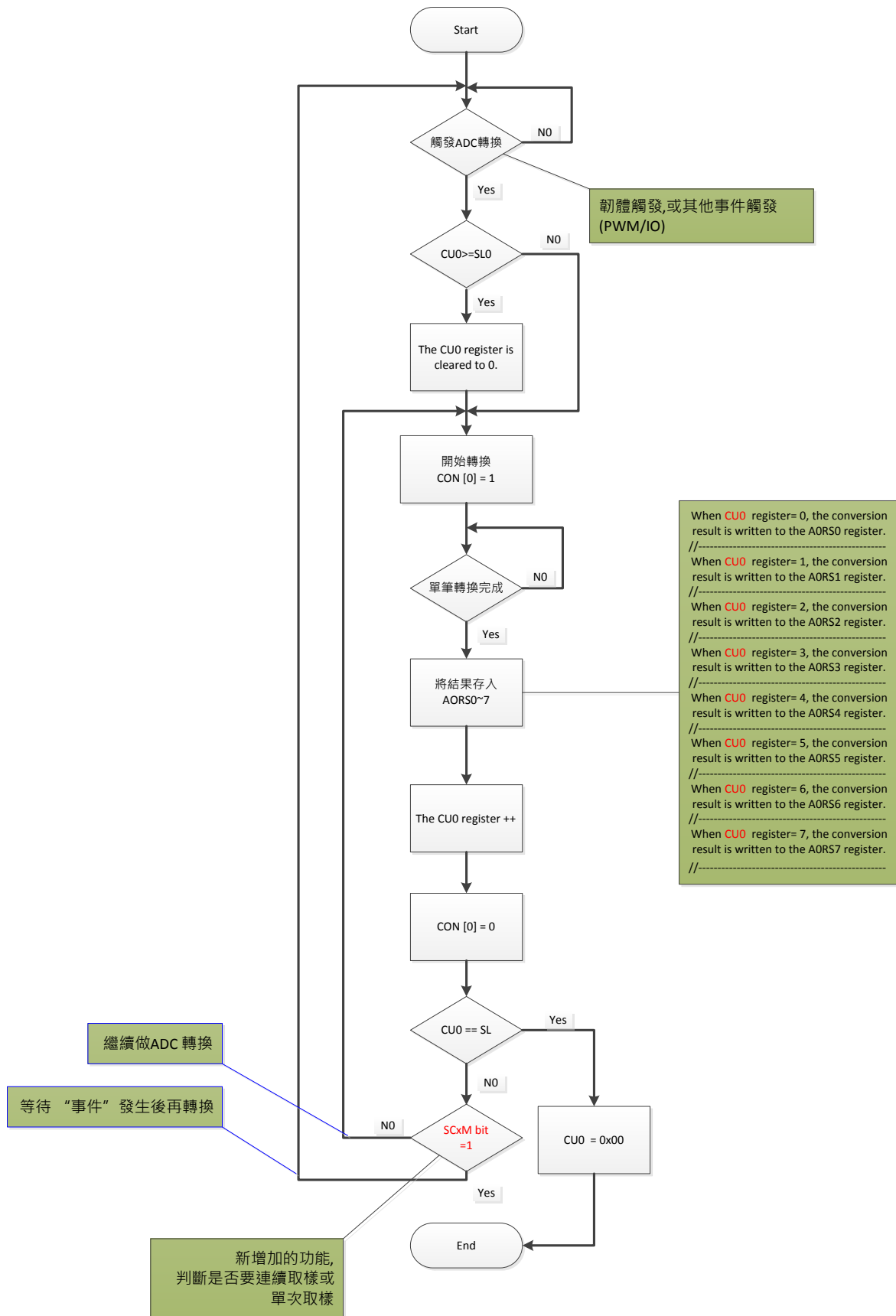
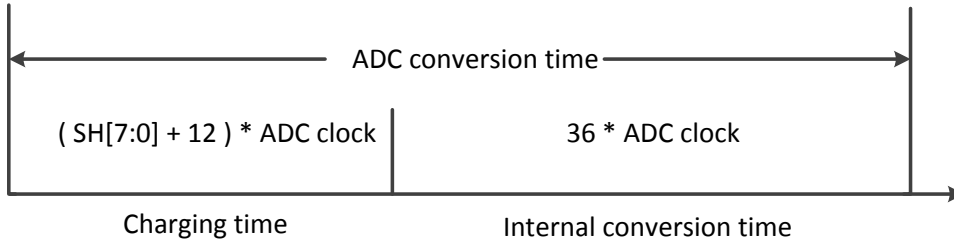


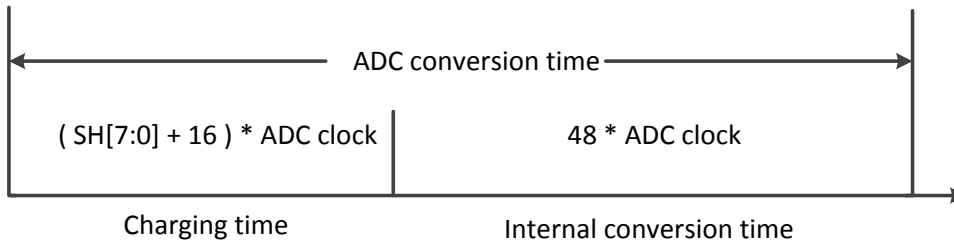
图 19.3.4 ADC 数据管理框图

### 19.3.6 ADC 转换时间

当 CALCON0 [1] = 0 (ADC clock 介于 0MHz~48MHz), ADC转换时间 = 48 \* ADC clock (当SH[7:0]=0)



当 CALCON0 [1] = 1 (ADC clock 介于 48MHz~72MHz), ADC 转换时间 = 64 \* ADC clock(当 SH[7:0]=0)



可参考 CALCON0 - 0x5002 00A0 / SH0 - 0x5002 0034 / ADC0CD - 0x5002 0090 说明

当 ADC 的输入为内置 OPA 输出时, SH[7:0]的建议值如下:

当 CALCON0 [1] = 0,  $SH[7:0] > (157 \text{ ns} * \text{ADC Clock}) - 12$

当 CALCON0 [1] = 1,  $SH[7:0] > (157 \text{ ns} * \text{ADC Clock}) - 16$

若以系统时钟为72MHz, CALCON0[1]及ADC0CD设置0为例:

$$\text{ADC Clock} = 72\text{MHz} / 1, (\text{ADC Clock} = \text{系统时钟} / (\text{ADC0CD} + 1))$$

$$SH[7:0] > (157 \text{ ns} * 72\text{M}) - 12 \Rightarrow \text{设置 } SH[7:0]=0$$

当 ADC 的输入为外部信号时, SH[7:0]的建议值如下:

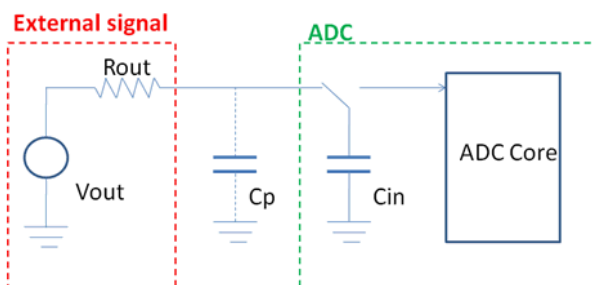
当 CALCON0 [1] = 0

$$(SH[7:0]+12) < (\text{ADC clock} * \text{Rout}) / (3.3 * 10^{-10})$$

当 CALCON0 [1] = 1

$$(SH[7:0]+16) < (\text{ADC clock} * \text{Rout}) / (3.3 * 10^{-10})$$

Rout = 外部信号的等效输出电阻



Cin : Sampling capacitor of ADC = 6.4pF  
 Cp: Packaging and other parasitic = 5pF  
 Rout: equivalent output resistance of external signal

### 19.3.7 ADC 中断进入方式

ADC 有提供以下三种中断进入方式, 可参考 IEENS - 0x5002 0070 说明

- 第一种: 单次中断
- IEENS[0]: 每次 ADC 转换完成都会发生中断
- 第二种: 序列中断
- IEENS[1]: 当 SLO 等于 CU0 时发生中断
- 第三种: 次数中断
- IEENS[8]: ADC0 的第一次转换完成后将产生中断信号
- IEENS[9]: ADC0 的第二次转换完成后将产生中断信号
- IEENS[10]: ADC0 的第三次转换完成后将产生中断信号
- IEENS[11]: ADC0 的第四次转换完成后将产生中断信号
- IEENS[12]: ADC0 的第五次转换完成后将产生中断信号
- IEENS[13]: ADC0 的第六次转换完成后将产生中断信号
- IEENS[14]: ADC0 的第七次转换完成后将产生中断信号
- IEENS[15]: ADC0 的第八次转换完成后将产生中断信号

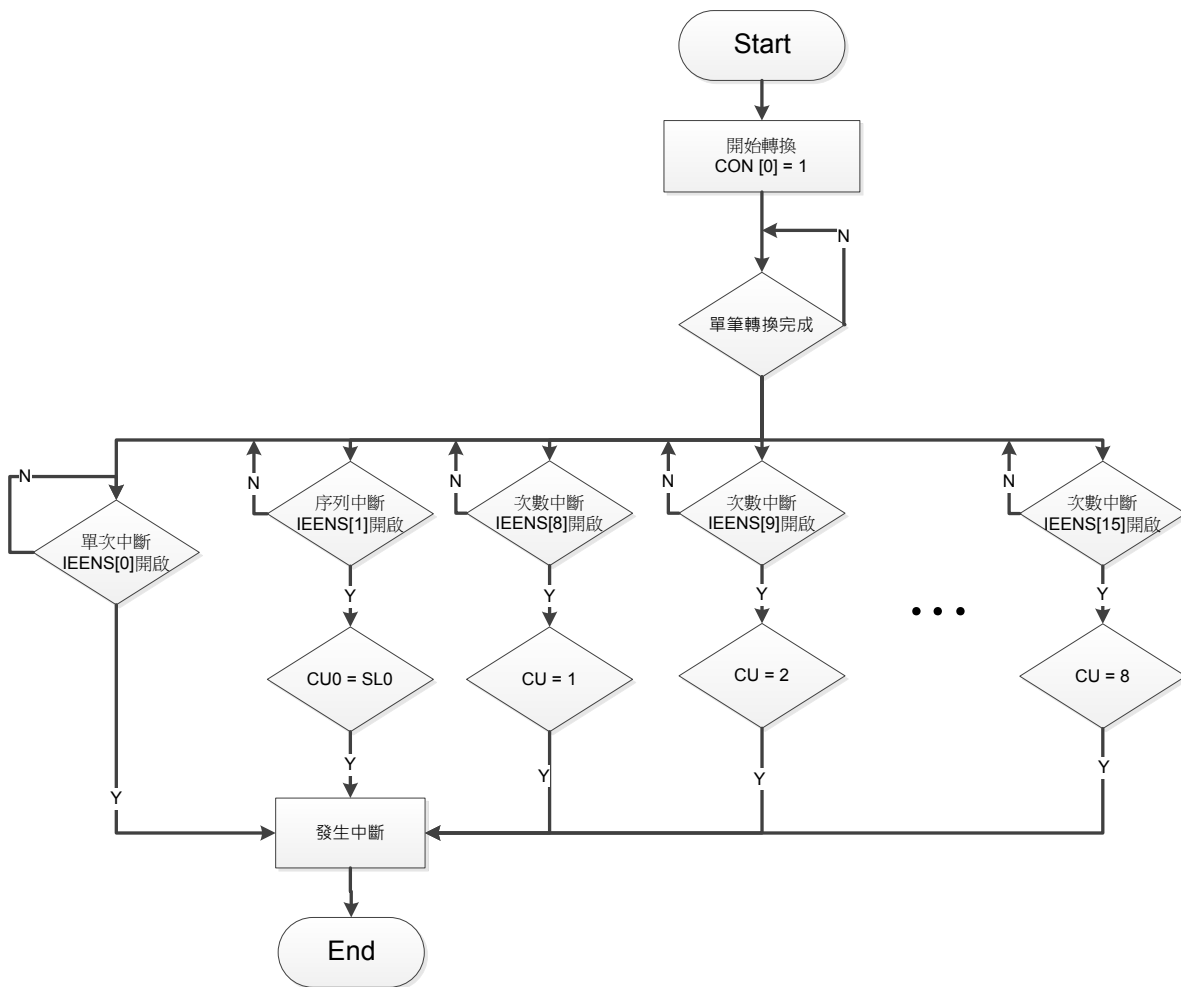


图 19.3.7 ADC 中断进入方式框图

#### 19.4 管脚描述

表 19.4.1 管脚总结

Pin	Type	Description
ADC_0	I	模拟到数字转换输入0
ADC_4	I	模拟到数字转换输入4
ADC_5	I	模拟到数字转换输入5
ADC_6	I	模拟到数字转换输入6
ADC_7	I	模拟到数字转换输入7
ADC_9	I	模拟到数字转换输入9
ADC_10	I	模拟到数字转换输入10
ADC_11	I	模拟到数字转换输入11
ADC_12	I	模拟到数字转换输入12

#### 19.5 寄存器描述

表 19.5.1 ADC 模块寄存器分布图(基地址 0x5002 C000)

名称	存取	地址 偏移	描述	重置 值
ADCCON	R/W	0x00	ADC控制寄存器	0
A0SC0	R/W	0x04	ADC 0采样序列0寄存器	0
A0SC1	R/W	0x08	ADC 0采样序列1寄存器	0
SL0	R/W	0x0C	ADC 0采样序列长度寄存器	0
CU0	RO	0x10	ADC 0采样序列计数器寄存器	0
A0RS0	RO	0x14	ADC 0采样结果0寄存器	0
A0RS1	RO	0x18	ADC 0采样结果1寄存器	0
A0RS2	RO	0x1C	ADC 0采样结果2寄存器	0
A0RS3	RO	0x20	ADC 0采样结果3寄存器	0
A0RS4	RO	0x24	ADC 0采样结果4寄存器	0
A0RS5	RO	0x28	ADC 0采样结果5寄存器	0
A0RS6	RO	0x2C	ADC 0采样结果6寄存器	0
A0RS7	RO	0x30	ADC 0采样结果7寄存器	0
SH0	R/W	0x34	ADC 0采样并且保留寄存器	0
-	-	0x38~ 0x6C	保留	0
IEENS	WO	0x70	ADC 中断使能设置寄存器	0
IEENC	WO	0x74	ADC 中断使能清除寄存器	0
IEST	RO	0x78	ADC 中断状态寄存器	0
IESTC	WO	0x7C	ADC 中断状态清除寄存器	0
-	-	0X84	保留	0

RSIN0	R/W	0x88	ADC 0结果索引寄存器	0
保留	R/W	0x8C	保留	0
ADC0CD	R/W	0x90	ADC 0时钟分频寄存器	0
-	-	0x94- 0x9C	保留	0
CALCON0	R/W	0xA0	ADC 0 校准控制寄存器	0
CALSTA0	R/W	0xA4	ADC 0 校准状态寄存器	0
-	-	0xA8~ 0xDC	保留	0

### 19.5.1 ADC 控制寄存器(ADCCON – 0x5002 0000)

表 19.5.2 ADC 控制寄存器(ADCCON - 0x5002 0000) 位描述

位	代号	值	描述	重置值
0	START0	1	当程序将此位设置为 1 时, ADC 0 模块将开始转换。 (由硬件自动清除为0)	0
1	EN0		ADC 0 模块使能控制位	0
		0	禁用 ADC 0 模块	
		1	使能 ADC 0 模块	
2			保留	0
3			保留	0
4	MODE0		ADC 0 阶数模式选择位	0
		0	12 位(4096 阶)	
		1	10 位(1024 阶)	
5	RSTRSIN0		重置 RSIN 计数器控制位	0
		0	闲置	
		1	重置 RSIN0 的计数器。(由硬件自动清除为 0)	
6	RSTCU0		重置 CU0 的计数器控制位	0
		0	闲置	
		1	重置 CU0 的计数器。(由硬件自动清除为 0)	
31:7	-	-	保留	0

### 19.5.2 ADC 0 采样序列 0 寄存器 (A0SC0 - 0x5002 0004)

选择第 1 次到 4 次的采样通道

表 19.5.3 ADC 0 采样序列 0 寄存器(A0SC0 - 0x5002 0004) 位描述

位	代号	值	描述	重置值
7:0	A0SC0		采样输入选择	0
		0x00	ADC 通道 0	
		0x04	ADC 通道 4	
		0x05	ADC 通道 5	
		0x06	ADC 通道 6	
		0x07	ADC 通道 7	
		0x09	ADC 通道 9	
		0x0A	ADC 通道 10	
		0x0B	ADC 通道 11	
		0x0C	ADC 通道 12	
		0x0D	ADC 通道 13 (仅连接到 OPA0_O)	
		0x0E	ADC 通道 14 (仅连接到 OPA1_O)	
		0x0F	ADC 通道 15 (仅连接到 OPA2_O)	
		0x10	ADC 通道 16 (仅连接到能隙 (0.6v) )	
		其他	不动作	

位	代号	值	描述	重置值
6:0	SC1		1st 采样输入选择 参考表 19.5.3	0
7:7	SC1M		在第一次转换后是否继续 ADC 转换	
		0	继续	
		1	停止	
14:8	SC2		2nd 采样输入选择 参考表 19.5.3	0
15:15	SC2M		在第二次转换后是否继续 ADC 转换	
		0	继续	
		1	停止	
22:16	SC3		3rd 采样输入选择 参考表 19.5.3	0
23:23	SC3M		在第三次转换后是否继续 ADC 转换	
		0	继续	
		1	停止	
30:24	SC4		4th 采样输入选择 参考表 19.5.3	0
31:31	SC4M		在第四次转换后是否继续 ADC 转换	
		0	继续	
		1	停止	

### 19.5.3 ADC 0 采样序列 1 寄存器 (A0SC1 - 0x5002 0008)

选择第 5 次到 8 次的采样通道

表 19.5.4 ADC 0 采样序列 1 寄存器(A0SC1 - 0x5002 0008) 位描述

位	代号	值	描述	重置值
7:0	A0SC1		采样输入选择	0
		0x00	ADC 通道 0	
		0x04	ADC 通道 4	
		0x05	ADC 通道 5	
		0x06	ADC 通道 6	
		0x07	ADC 通道 7	
		0x09	ADC 通道 9	
		0x0A	ADC 通道 10	
		0x0B	ADC 通道 11	
		0x0C	ADC 通道 12	
		0x0D	ADC 通道 13 (仅连接到 OPA0_O)	
		0x0E	ADC 通道 14 (仅连接到 OPA1_O)	
		0x0F	ADC 通道 15 (仅连接到 OPA2_O)	
		0x10	ADC 通道 16 (仅连接到能隙 (0.6v) )	
		其他	不动作	

位	代号	值	描述	重置值
6:0	SC5		5th 采样输入选择	0
7:7	SC5M		在第五次转换后是否继续 ADC 转换	
		0	继续	
		1	停止	
14:8	SC6		6th 采样输入选择 参考表 19.5.4	0
15:15	SC6M		在第六次转换后是否继续 ADC 转换	
		0	继续	

	1	停止	
22:16 SC7		7th 采样输入选择 参考表 19.5.4	0
23:23: SC7M		在第七次转换后是否继续 ADC 转换	
	0	继续	
	1	停止	
30:24 SC8		8th 采样输入选择 参考表 19.5.4	0
31: SC8M		在第八次转换后是否继续 ADC 转换	
	0	继续	
	1	停止	

#### 19.5.4 ADC 0 样本序列长度寄存器(SL0 – 0x5002 000C)

表 19.5.5 ADC 0 样本序列长度寄存器(SL0 - 0x5002 000C) 位描述

位	代号	值	描述	重置值
3:0	SL	-	ADC 0 采样次数选择	0
31:4	-	-	保留	0

#### 19.5.5 ADC 0 采样序列计数器寄存器(CU0 – 0x5002 0010)

表 19.5.6 ADC 0 采样序列计数器寄存器(CU0 - 0x5002 0010) 位描述

位	代号	值	描述	重置值
3:0	CU	-	ADC 0 采样序列计数器寄存器 当转换完成一次后，CU0 数值将会加一。 当 CU0 等于 SL0 时，硬件将自动将 CU0 清零。	0
31:4	-	-	保留	0

**19.5.6 ADC 0 采样结果 0 寄存器(A0RS0 – 0x5002 0014)**

表 19.5.7 ADC 0 采样结果 0 寄存器(A0RS0 - 0x5002 0014) 位描述

位	代号	值	描述	重置值
11:0	RS	-	12bit ADC 0 采样结果 0 寄存器(若为 10bit ADC RS 为[9:0]位) 当第 1 次转换时, 转换结束时转换后的数据的结果被存储在寄存器中	0
31:12	-	-	保留	0

**19.5.7 ADC 0 采样结果 1 寄存器(A0RS1 – 0x5002 0018)**

表 19.5.8 ADC 0 采样结果 1 寄存器(A0RS1 - 0x5002 0018) 位描述

位	代号	值	描述	重置值
11:0	RS	-	12bit ADC 0 采样结果 1 寄存器(若为 10bit ADC RS 为[9:0]位) 当第 2 次转换时, 转换结束时转换后的数据的结果被存储在寄存器中	0
31:12	-	-	保留	0

**19.5.8 ADC 0 采样结果 2 寄存器(A0RS2 – 0x5002 001C)**

表 19.5.9 ADC 0 采样结果 2 寄存器(A0RS2 - 0x5002 001C) 位描述

位	代号	值	描述	重置值
11:0	RS	-	12bit ADC 0 采样结果 2 寄存器(若为 10bit ADC RS 为[9:0]位) 当第 3 次转换时, 转换结束时转换后的数据的结果被存储在寄存器中	0
31:12	-	-	保留	0

**19.5.9 ADC 0 采样结果 3 寄存器(A0RS3 – 0x5002 0020)**

表 19.5.10 ADC 0 采样结果 3 寄存器(A0RS3 - 0x5002 0020) 位描述

位	代号	值	描述	重置值
11:0	RS	-	12bit ADC 0 采样结果 3 寄存器(若为 10bit ADC RS 为[9:0]位) 当第 4 次转换时, 转换结束时转换后的数据的结果被存储在寄存器中	0
31:12	-	-	保留	0

**19.5.10 ADC 0 采样结果 4 寄存器(A0RS4 – 0x5002 0024)**

表 19.5.11 ADC 0 采样结果 4 寄存器(A0RS4 - 0x5002 0024) 位描述

位	代号	值	描述	重置值
11:0	RS	-	12bit ADC 0 采样结果 4 寄存器(若为 10bit ADC RS 为[9:0]位) 当第 5 次转换时, 转换结束时转换后的数据的结果被存储在寄存器中	0
31:12	-	-	保留	0



**19.5.11 ADC 0 采样结果 5 寄存器(A0RS5 – 0x5002 0028)**

表 19.5.12 ADC 0 采样结果 5 寄存器(A0RS5 - 0x5002 0028) 位描述

位	代号	值	描述	重置值
11:0	RS	-	12bit ADC 0 采样结果 5 寄存器(若为 10bit ADC RS 为[9:0]位) 当第 6 次转换时, 转换结束时转换后的数据的结果被存储在寄存器中	0
31:12	-	-	保留	0

**19.5.12 ADC 0 采样结果 6 寄存器(A0RS6 – 0x5002 002C)**

表 19.5.13 ADC 0 采样结果 6 寄存器(A0RS6 - 0x5002 002C) 位描述

位	代号	值	描述	重置值
11:0	RS	-	12bit ADC 0 采样结果 6 寄存器(若为 10bit ADC RS 为[9:0]位) 当第 7 次转换时, 转换结束时转换后的数据的结果被存储在寄存器中	0
31:12	-	-	保留	0

**19.5.13 ADC 0 采样结果 7 寄存器(A0RS7 – 0x5002 0030)**

表 19.5.14 ADC 0 采样结果 7 寄存器(A0RS7 - 0x5002 0030) 位描述

位	代号	值	描述	重置值
11:0	RS	-	12bit ADC 0 采样结果 7 寄存器(若为 10bit ADC RS 为[9:0]位) 当第 8 次转换时, 转换结束时转换后的数据的结果被存储在寄存器中	0
31:12	-	-	保留	0

**19.5.14 ADC 0 采样和保持寄存器(SH0 – 0x5002 0034)**

可参考 19.3.6 ADC 转换时间 章节说明

ADC 引脚须对 ADC 的内部电容做充电的动作, 完成后 ADC 再进行转换, 如果充电不足会影响 ADC 的转换准确度, 当充电电流越小需要充电的时间将会越长

**表 19.5.15 ADC 0 采样和保持寄存器(SH0 - 0x5002 0034) 位描述**

位	代号	值	描述	重置值
7:0	SH	-	ADC 0 采样保留时间寄存器. 设置数值越大其充电保持时间将会越长	5
31:8	-	-	保留	0

**19.5.15 ADC 0 中断使能设置寄存器(IEENS – 0x5002 0070)**
**表19.5.16 ADC中断使能设置寄存器(IEENS - 0x5002 0070) 位描述**

位	代号	值	描述	重置值
0	SIN0	1	设置为 1, 每完成一通道的转换中断置能位。	0
1	SEQ0	1	设置为 1, 每完成一时序的转换中断置能位。 (当 CU0 = SL0)	0
7:2	-	-	保留	0
8	IES1		ADC0 的第 1 次转换完成后, 将产生中断信号。	0
9	IES2		ADC0 的第 2 次转换完成后, 将产生中断信号。	0
10	IES3		ADC0 的第 3 次转换完成后, 将产生中断信号。	0
11	IES4		ADC0 的第 4 次转换完成后, 将产生中断信号。	0
12	IES5		ADC0 的第 5 次转换完成后, 将产生中断信号。	0
13	IES6		ADC0 的第 6 次转换完成后, 将产生中断信号。	0
14	IES7		ADC0 的第 7 次转换完成后, 将产生中断信号。	0
15	IES8		ADC0 的第 8 次转换完成后, 将产生中断信号。	0
31:16	-	-	保留	0

**19.5.16 ADC 0 中断使能清除寄存器(IEENC – 0x5002 0074)**
**表19.5.17 ADC中断使能清除寄存器(IEENC - 0x5002 0074) 位描述**

位	代号	值	描述	重置值
0	SIN0	1	写入 1 将清除 IEENS [0]。	0
1	SEQ0	1	写入 1 将清除 IEENS [1]。	0
7:2	-	-	保留	0
8	IEC1		写入 1 将清除 IEENS [8]。	0
9	IEC2		写入 1 将清除 IEENS [9]。	0
10	IEC3		写入 1 将清除 IEENS [10]。	0
11	IEC4		写入 1 将清除 IEENS [11]。	0
12	IEC5		写入 1 将清除 IEENS [12]。	0
13	IEC6		写入 1 将清除 IEENS [13]。	0
14	IEC7		写入 1 将清除 IEENS [14]。	0
15	IEC8		写入 1 将清除 IEENS [15]。	0
31:16	-	-	保留	0

**19.5.17 ADC 0 中断状态寄存器(IEST – 0x5002 0078)**
**表19.5.18 中断状态寄存器(IEST - 0x5002 0078) 位描述**

位	代号	值	描述	重置值
0	SIN0	1	每完成一通道的转换中断旗标位, 硬件自动设置。(只读)(硬件设置)	0
1	SEQ0	1	每完成一时序的转换中断旗标位, 硬件自动设置。(只读)(硬件设置)	0
7:2	-	-	保留	0

8	IEST1	设置后, 完成 ADC 的第 1 次转换。(只读)(硬件设置)	0
9	IEST2	设置后, 完成 ADC 的第 2 次转换。(只读)(硬件设置)	0
10	IEST3	设置后, 完成 ADC 的第 3 次转换。(只读)(硬件设置)	0
11	IEST4	设置后, 完成 ADC 的第 4 次转换。(只读)(硬件设置)	0
12	IEST5	设置后, 完成 ADC 的第 5 次转换。(只读)(硬件设置)	0
13	IEST6	设置后, 完成 ADC 的第 6 次转换。(只读)(硬件设置)	0
14	IEST7	设置后, 完成 ADC 的第 7 次转换。(只读)(硬件设置)	0
15	IEST8	设置后, 完成 ADC 的第 8 次转换。(只读)(硬件设置)	0
31:16	-	保留	0

### 19.5.18 ADC 0 中断状态清除寄存器(IESTC – 0x5002 007C)

表 19.5.19 ADC 中断状态清除寄存器(IESTC - 0x5002 007C) 位描述

位	代号	值	描述	重置值
0	SIN0	1	写入 1 将清除 IESTC [0]。(需软件写 1 清除)	0
1	SEQ0	1	写入 1 将清除 IESTC [1]。(需软件写 1 清除)	0
7:2	-	-	保留	0
8	IESTC1	1	写入 1 将清除 IESTC [8]。(需软件写 1 清除)	0
9	IESTC2	1	写入 1 将清除 IESTC [9]。(需软件写 1 清除)	0
10	IESTC3	1	写入 1 将清除 IESTC [10]。(需软件写 1 清除)	0
11	IESTC4	1	写入 1 将清除 IESTC [11]。(需软件写 1 清除)	0
12	IESTC5	1	写入 1 将清除 IESTC [12]。(需软件写 1 清除)	0
13	IESTC6	1	写入 1 将清除 IESTC [13]。(需软件写 1 清除)	0
14	IESTC7	1	写入 1 将清除 IESTC [14]。(需软件写 1 清除)	0
15	IESTC8	1	写入 1 将清除 IESTC [15]。(需软件写 1 清除)	0
31:16	-	-	保留	0

### 19.5.19 ADC 0 结果索引寄存器(RSIN0 – 0x5002 0088)

表 19.5.21 ADC 0 结果索引寄存器(RSIN0 - 0x5002 0088) 位描述

位	代号	值	描述	重置值
30:0	-	-	保留	0
31	MODE	1	索引模式	1
1 始终重置(只能设置为 1)				

### 19.5.20 ADC 0 时钟分频寄存器(ADC0CD – 0x5002 0090)

表 19.5.22 ADC 0 时钟分频寄存器(ADC0CD - 0x5002 0090) 位描述

位	代号	值	描述	重置值
2:0	CD0	000 ~ 011: ADC0CD = 0 ~ 3 100 ~ 111: ADC0CD = 4 ~ 7	ADC 0 时钟分频寄存器。 ADC 0 时钟 = 系统时钟 / (ADC0CD + 1)	0
31:3	-	-	保留	0

### 19.5.21 ADC 0 校准控制寄存器(CALCON0 – 0x5002 00A0)

表 19.5.23 ADC 0 校准控制寄存器(CALCON0 - 0x5002 00A0)位描述

位	代号	值	描述	重置值
0	EN	0	ADC 校准使能位	0
		1	禁用 ADC 0 校准	
		1	使能 ADC 0 校准	

1	BC		BC 选择(ADC 转换所需 clock 数)	0
		0	48 个 ADC clock	
		1	64 个 ADC clock	
7:2	-	-	保留	0
8	START		ADC 校准启动位	0
		0	校准完成 (硬件将自动清除此位为 0)	
		1	校准启动	
31:9	-	-	保留	0

### 19.5.22 ADC 0 校准状态寄存器(CALSTA0 – 0x5002 00A4)

表 19.5.24 ADC 0 校准状态寄存器(CALSTA0 - 0x5002 00A4)位描述

位	代号	值	描述	重置值
0	STA		校准状态标志	0
		0	校准通过	
		1	校准失败	
31:1	-	-	保留	0

### 19.5.23 ADC 0 锁存计数器寄存器(LATCON – 0x5002 00E8)

表 19.5.25 ADC 0 锁存计数器寄存器(LATCON - 0x5002 00E8)位描述

位	代号	值	描述	重置值
7:0	LATCON		当 ADC 尚未完成转换时，用来纪录外部来的触发讯号次数(此功能为硬件处理) 当转换完毕后，若此寄存器非为 0，则硬件继续 ADC 转换并将此寄存器数值减 1 此功能可防止因 ADC 转换期间，所忽略掉的触发讯号，最大纪录深度为 8	0
31:8	-	-	保留	0

## 20. 模拟比较器(ACMP)

### 20.1 简介

模拟比较器是比较两个模拟电压的装置。比较器可以将其输出提供给设备引脚。作为板上模拟比较器的替代品，或者它可以通过中断或触发器向ADC发送信号，使其开始捕获采样序列。

### 20.2 特征

比较器模块提供以下功能：

- 3 个集成模拟比较器。
- 输出可配置为驱动输出引脚，产生中断或 ADC 采样序列。

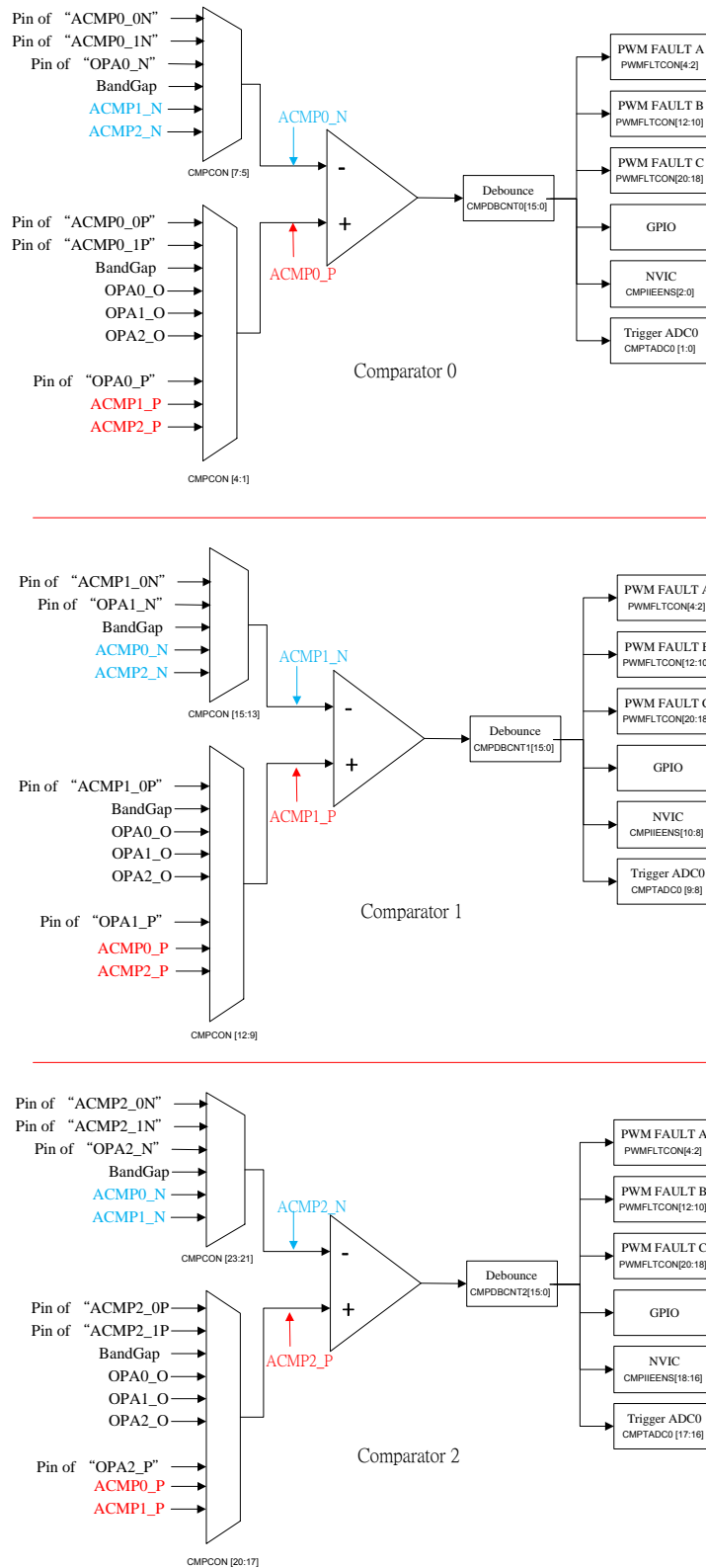
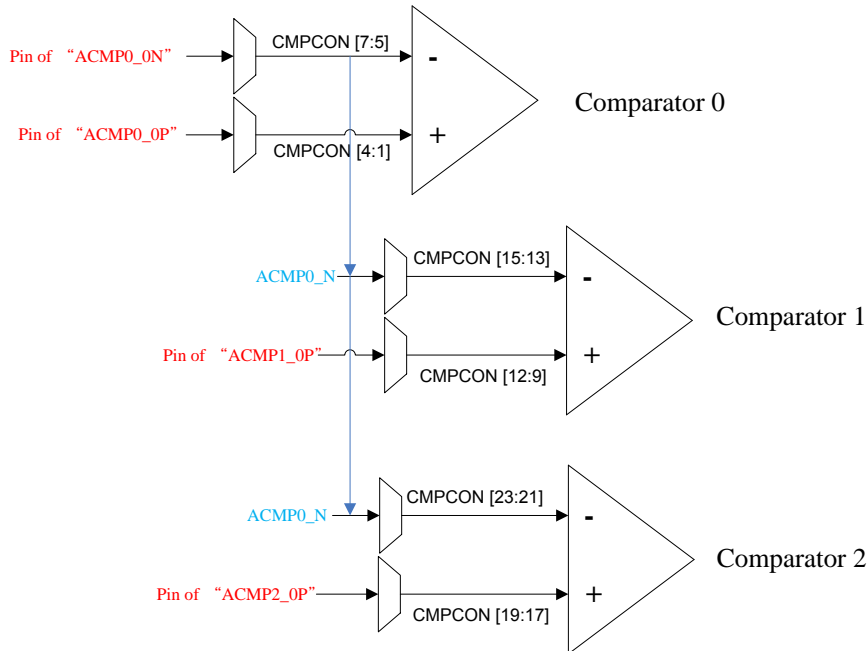
**20.3 功能描述**


图 20.3 显示比较器框图

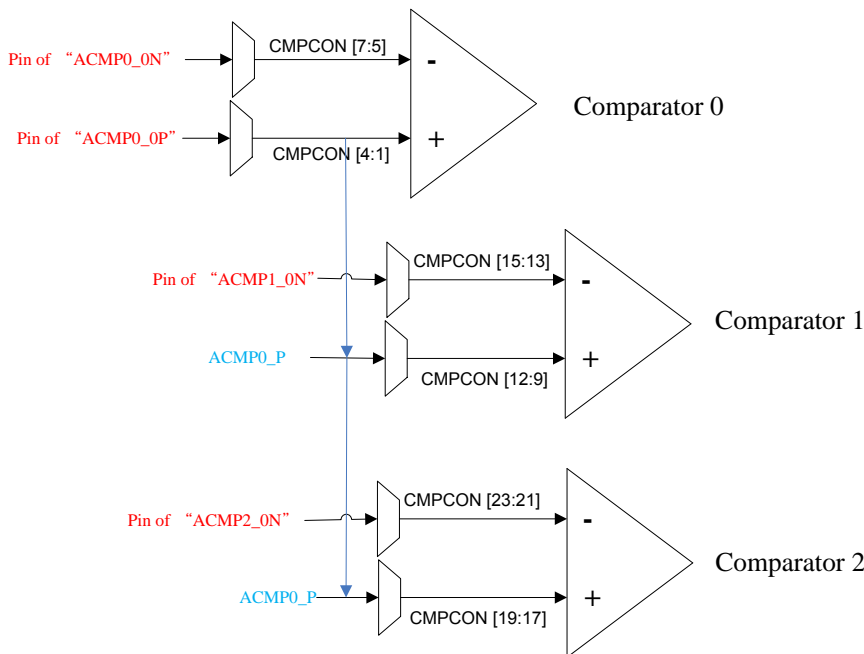
使用 ACMP 内部短接方式，达到减少外部接线的功能示例:



ACMP0\_N/ACMP1\_N/ACMP2\_N 主要用来做内部短接, 达到减少外部接线的功能

当需要将负端共接时使用以下方式(以下范例如 "比较器0"/"比较器1"/"比较器2"皆可使用此方式)

- 1.使用"比较器0"的ACMP0\_ON 当作负端的输入脚
- 2.将"比较器1"的负端, 利用内部接线方式连到比较器0的负端(连接至 ACMP0\_N)
- 3.将"比较器2"的负端, 利用内部接线方式连到比较器0的负端(连接至 ACMP0\_N)
- 4."比较器0"/"比较器1"/"比较器2" 的正端由 ACMP0\_OP/ACMP1\_OP/ACMP2\_OP 作为输入脚



ACMP0\_P/ACMP1\_P/ACMP2\_P 主要用来做内部短接, 达到减少外部接线的功能

当需要将正端共接时使用以下方式(以下范例如 "比较器0"/"比较器1"/"比较器2"皆可使用此方式)

- 1.使用"比较器0"的ACMP0\_OP 当作正端的输入脚
- 2.将"比较器1"的正端, 利用内部接线方式连到比较器0的正端(连接至 ACMP0\_P)
- 3.将"比较器2"的正端, 利用内部接线方式连到比较器0的正端(连接至 ACMP0\_P)
- 4."比较器0"/"比较器1"/"比较器2" 的负端由 ACMP0\_ON/ACMP1\_ON/ACMP2\_ON 作为输入脚

### 20.3.1 COMP 引脚和内部信号

每个比较器都有正输入用于灵活的电压选择:

- pin of CMP\_P
- Band Gap

### 20.3.2 中断和事件

比较器输出在内部连接到中断和事件控制器。每个比较器都可以产生中断或事件（触发 ADC0，触发故障）

### 20.3.3 迟滞功能

模拟比较器提供迟滞功能，使比较器输出转换更加稳定。如果比较器输出为0，则直到正输入电压超过负输入电压的正迟滞电压(Positive hysteresis voltage)时才会变为1。

类似地，如果比较器输出为1，则直到正输入电压下降到负迟滞电压(Negative hysteresis voltage)以下时才会变为0。

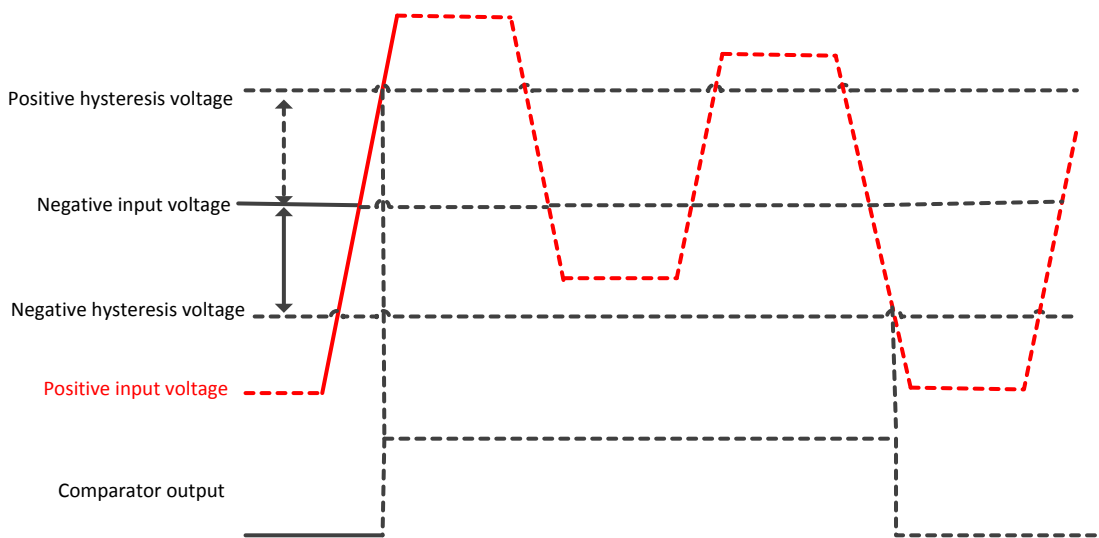


图 20.3.3 显示比较器框图

## 20.4 引脚描述

表 20.4.1 引脚摘要

位	形式	描述
ACMP0_P	I	比较器0正输入
ACMP0_N	I	比较器0负输入
ACMP0_O	O	比较器0输出
ACMP1_P	I	比较器1正输入
ACMP1_N	I	比较器1负输入
ACMP1_O	O	比较器1输出
ACMP2_P	I	比较器2正输入
ACMP2_N	I	比较器2负输入
ACMP2_O	O	比较器2输出

## 20.5 寄存器描述

表 20.5.1 比较器模块寄存器分布图(基地址 0x5002 4000)



Name	存取	地址 偏移	描述	重置值
ACMPCON	R/W	0x00	比较器控制寄存器	
OUT	R/W	0x04	比较器输出寄存器	
HYS	R/W	0x08	较器迟滞寄存器	
OUTSTA	RO	0x0C	比较器输出状态寄存器	
TADC0	R/W	0x10	比较器触发ADC 0寄存器	
-	-	0x14	保留	
IEENS	WO	0x18	比较器中断使能设置寄存器	
IEENC	WO	0x1C	比较器中断使能清除寄存器	
IEST	RO	0x20	比较器中断状态寄存器	
IESTC	WO	0x24	比较器中断状态清除寄存器	
-	-	0x28 - 0x2C	保留	-
DBCNT0	R/W	0x30	比较器去反弹计数器0寄存器	
DBCNT1	R/W	0x34	比较器去反弹计数器1寄存器	
DBCNT2	R/W	0x38	比较器去反弹计数器2寄存器	

### 20.5.1 比较器控制寄存器(ACMPCON – 0x5002 4000)

表 20.5.1.1 比较器控制寄存器(ACMPCON - 0x5002 4000) 位描述

位	代号	值	描述	重置值
0	EN0		比较器 0 功能使能位	0
		0	禁用	
		1	使能	
4:1	PSEL0		选择比较器 0 正输入源 见表 20.5.1.1.1	0
7:5	NSEL0		选择比较器 0 负输入源 见表 20.5.1.1.2	0
8	EN1		比较器 1 功能使能位	0
		0	禁用	
		1	使能	
12:9	PSEL1		选择比较器 1 正输入源 见表 20.5.1.1.1	0
15:13	NSEL1		选择比较器 1 负输入源 见表 20.5.1.1.2	0
16	EN2		比较器 2 功能使能位	0
		0	禁用	
		1	使能	
20:17	PSEL2		选择比较器 2 正输入源 见表 20.5.1.1.1	0
23:21	NSEL2		选择比较器 2 负输入源 见表 20.5.1.1.2	0
31:24			保留	

表 20.5.1.1.1

comparator comparator select (PSEL)	positive source	ACMP0	ACMP1	ACMP2
0		Pin of ACMP0_0P	Pin of ACMP1_0P	Pin of ACMP2_0P
1		Pin of ACMP0_1P	Pin of ACMP1_1P	Pin of ACMP2_1P
2		Pin of OPA0_P	Pin of OPA1_P	Pin of OPA2_P
3		Band Gap(0.6v)	Band Gap(0.6v)	Band Gap(0.6v)
4		Reserved	Pin of ACMP0_P	Pin of ACMP0_P
5		Pin of ACMP1_P	Reserved	Pin of ACMP1_P
6		Pin of ACMP2_P	Pin of ACMP2_P	Reserved
7		Reserved	Reserved	Reserved
8		Pin of OPA0_O	Pin of OPA0_O	Pin of OPA0_O
9		Pin of OPA1_O	Pin of OPA1_O	Pin of OPA1_O
10		Pin of OPA2_O	Pin of OPA2_O	Pin of OPA2_O
其他		Pin of ACMP0_0P	Pin of ACMP1_0P	Pin of ACMP2_0P

表 20.5.1.1.2

comparator comparator select (NSEL)	positive source	ACMP0	ACMP1	ACMP2
0		Pin of ACMP0_0N	Pin of ACMP1_0N	Pin of ACMP2_0N
1		Pin of ACMP0_1N	Pin of ACMP1_1N	Pin of ACMP2_1N
2		Pin of OPA0_N	Pin of OPA1_N	Pin of OPA2_N
3		Band Gap(0.6v)	Band Gap(0.6v)	Band Gap(0.6v)
4		Reserved	Pin of ACMP0_N	Pin of ACMP0_N
5		Pin of ACMP1_N	Reserved	Pin of ACMP1_N
6		Pin of ACMP2_N	Pin of ACMP2_N	Reserved
其他		Pin of ACMP0_0N	Pin of ACMP1_0N	Pin of ACMP2_0N

### 20.5.2 比较器迟滞寄存器(HYS – 0x5002 4008)

表 20.5.2.1 比较器迟滞寄存器(HYS - 0x5002 4008) 位描述

位	代号	值	描述	重置值
2:0	HYS0		比较器 0 具有内部迟滞	0
7:3	-	-	保留	
10:8	HYS1		比较器 1 具有内部迟滞	
15:11	-	-	保留	
18:16	HYS2		比较器 2 具有内部迟滞	
31:19	-	-	保留	

HYS	000	001	010	011	100	101	110	111
迟滞范围(mV)	0	50	100	150	200	250	300	350

### 20.5.3 比较器输出寄存器(OUT – 0x5002 4004)

表 20.5.3.1 比较器输出寄存器(OUT - 0x5002 4004) 位描述

位	代号	值	描述	重置值
0	LATCHSEL0	-	比较器 0 数据锁存选择	0
		0	原始数据	
		1	PWM 触发锁存比较器 0	
1	INV0		比较器 0 输出引脚反相位	
7:2	-	-	保留	

8	LATCHSEL1	比较器 1 数据锁存选择	0
		0 原始数据	
		1 PWM 触发锁存比较器 1	
9	INV1	比较器 1 输出引脚反相位	
15:10	-	保留	
16	LATCHSEL2	比较器 2 数据锁存选择	0
		0 原始数据	
		1 PWM 触发锁存比较器 2	
17	INV2	比较器 2 输出引脚反相位	0
31:18	-	保留	

表 20.5.3.2

Function output invert bit (INV)	CMP <sub>x</sub> _P > CMP <sub>x</sub> _N (x= 0/1/2)	CMP <sub>x</sub> _P < CMP <sub>x</sub> _N (x= 0/1/2)
0	Output pin = high	Output pin = low
1	Output pin = low	Output pin = high

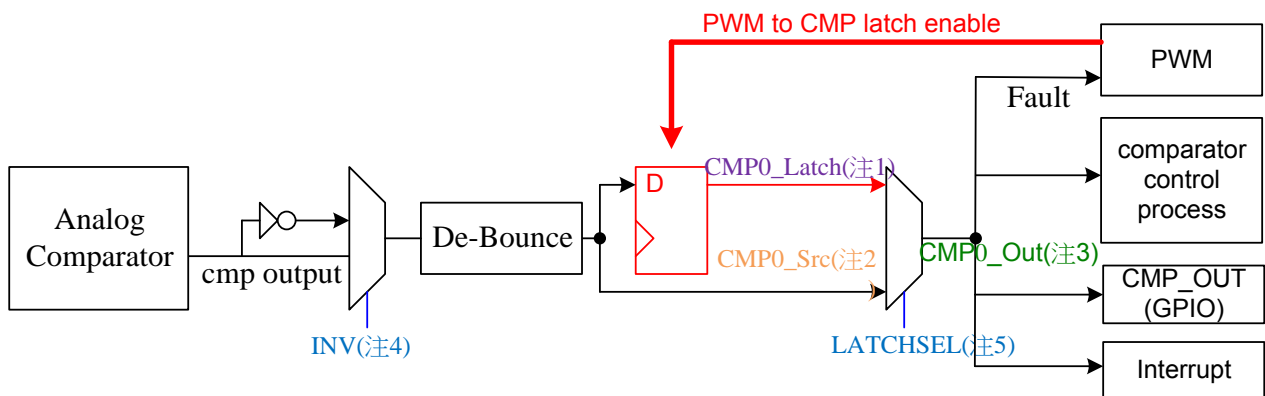


图 20.5.3 比较器输出状态框图(CMP0 为例)

请参考SFR OUT - 0x5002 4004

INV(注4) => OUT[1], OUT[9], OUT[17]

LATCHSEL(注5) 请参考SFR OUT[0], OUT[8], OUT[16],

请参考SFR OUTSTA - 0x5002 400C

CMP0\_Out(注3) 请参考 OUTSTA[2:0]

CMP0\_Src(注 2) 请参考 OUTSTA[6:4]

CMP0\_Latch(注 1) 请参考 OUTSTA[10:8]

**20.5.4 比较器输出状态寄存器(OUTSTA – 0x4005 000C)**
**表 20.5.4.1 比较器输出状态寄存器(OUTSTA - 0x5002 400C) 位描述**

位	代号	值	描述	重置值
0	OUT0(注 3)		LATCHSEL 输出信号(比较器 0)	
1	OUT1(注 3)		LATCHSEL 输出信号(比较器 1)	
2	OUT2(注 3)		LATCHSEL 输出信号(比较器 2)	
3	-	-	保留	
4	SRC0(注 2)		比较器输出值 (原始数据) (比较器 0)	
5	SRC1(注 2)		比较器输出值 (原始数据) (比较器 1)	
6	SRC2(注 2)		比较器输出值 (原始数据) (比较器 2)	
7	-	-	保留	
8	LATCH(注 1)		PWM 锁存比较器数据(比较器 0)	
9	LATCH(注 1)		PWM 锁存比较器数据(比较器 1)	
10	LATCH(注 1)		PWM 锁存比较器数据(比较器 2)	
11	-	-	保留	
31:12	-	-	保留	

**20.5.5 比较器触发 ADC 0 寄存器(TADC0 – 0x5002 4010)**
**表 20.5.5.1 比较器触发 ADC x 寄存器(TADC0/1 - 0x5002 4010) 位描述**

位	代号	值	描述	重置值
0	FEN0		比较器 0 输出的下降沿触发 ADC	0
		0	禁用	
		1	使能	
1	REN0		比较器 0 输出的上升沿触发 ADC	0
		0	禁用	
		1	使能	
7:2	-	-	保留	
8	FEN1		比较器 1 输出的下降沿触发 ADC	0
		0	禁用	
		1	使能	
9	REN1		比较器 1 输出的上升沿触发 ADC	0
		0	禁用	
		1	使能	
15:10	-	-	保留	
16	FEN2		比较器 2 输出的下降沿触发 ADC	0
		0	禁用	
		1	使能	
17	REN2		比较器 2 输出的上升沿触发 ADC	0
		0	禁用	
		1	使能	
31:18	-	-	保留	

**20.5.6 比较器中断使能设置寄存器(IEENS – 0x5002 4018)**
**表 20.5.6.1 比较器中断使能设置寄存器(IEENS - 0x5002 4018) 位描述**

位	代号	值	描述	重置值
0	FIE0		写 1 设置中断使能 (用于比较器 0 的下降沿)	0
1	RIE0		写 1 设置中断使能 (用于比较器 0 的上升沿)	0
2	LIE0		写 1 设置中断使能 (用于 PWM 触发比较器 0 锁存数据)	0
7:3	-	-	保留	
8	FIE1		写 1 设置中断使能 (用于比较器 1 的下降沿)	0

9	RIE1		写 1 设置中断使能 (用于比较器 1 的上升沿)	0
10	LIE1		写 1 设置中断使能 (用于 PWM 触发比较器 1 锁存数据)	0
15:11	-	-	保留	
16	FIE2		写 1 设置中断使能 (用于比较器 2 的下降沿)	0
17	RIE2		写 1 设置中断使能 (用于比较器 2 的上升沿)	0
18	LIE2		写 1 设置中断使能 (用于 PWM 触发比较器 2 锁存数据)	0
31:19	-	-	保留	0

### 20.5.7 比较器中断使能清除寄存器 (IEENC – 0x5002 401C)

表 20.5.7.1 比较器中断使能清除寄存器 (IEENC - 0x5002 401C) 位描述

位	代号	值	描述	重置值
0	FIE0		写入 1 将清除 IEENS[0]	0
1	RIE0		写入 1 将清除 IEENS[1]	0
2	LIE0		写入 1 将清除 IEENS[2]	0
7:3	-	-	保留	
8	FIE1		写入 1 将清除 IEENS[8]	0
9	RIE1		写入 1 将清除 IEENS[9]	0
10	LIE1		写入 1 将清除 IEENS[10]	0
15:11	-	-	保留	
16	FIE2		写入 1 将清除 IEENS[16]	0
17	RIE2		写入 1 将清除 IEENS[17]	0
18	LIE2		写入 1 将清除 IEENS[18]	0
31:19	-	-	保留	0

### 20.5.8 比较器中断状态寄存器 (IEST – 0x5002 4020)

表 20.5.8.1 比较器中断状态寄存器 (IEST - 0x5002 4020) 位描述

位	代号	值	描述	重置值
0	FIE0		比较器 0 输出的下降沿产生一个中断	0
1	RIE0		比较器 0 输出的上升沿产生一个中断	0
2	LIE0		PWM 触发比较器 0 锁存数据输出以产生中断 在 PWM 触发比较器的区间, LIE0 会为 1	
7:3	-	-	保留	
8	FIE1		比较器 1 输出的下降沿产生一个中断	0
9	RIE1		比较器 1 输出的上升沿产生一个中断	0
10	LIE1		PWM 触发比较器 1 锁存数据输出以产生中断 在 PWM 触发比较器的区间, LIE1 会为 1	
15:11	-	-	保留	
16	FIE2		比较器 2 输出的下降沿产生一个中断	0
17	RIE2		比较器 2 输出的上升沿产生一个中断	0
18	LIE2		PWM 触发比较器 2 锁存数据输出以产生中断 在 PWM 触发比较器的区间, LIE2 会为 1	
31:19	-	-	保留	0

### 20.5.9 比较器中断状态清除寄存器 (IESTC – 0x5002 4024)

表 20.5.9.1 比较器中断状态清除寄存器 (IESTC - 0x5002 4024) 位描述

位	代号	值	描述	重置值
0	FIE0		写 1 清除中断状态 (IEST[0])。(需软件写 1 清除)	0
1	RIE0		写 1 清除中断状态 (IEST[1])。(需软件写 1 清除)	0
2	LIE0		写 1 清除中断状态 (IEST[2])。(需软件写 1 清除)	
7:3	-	-	保留	
8	FIE1		写 1 清除中断状态 (IEST[8])。(需软件写 1 清除)	0

---

9	RIE1	写 1 清除中断状态 (IEST[9])。(需软件写 1 清除)	0
10	LIE1	写 1 清除中断状态 (IEST[10])。(需软件写 1 清除)	
15:11	-	保留	
16	FIE2	写 1 清除中断状态 (IEST[16])。(需软件写 1 清除)	0
17	RIE2	写 1 清除中断状态 (IEST[17])。(需软件写 1 清除)	0
18	LIE2	写 1 清除中断状态 (IEST[18])。(需软件写 1 清除)	
31:19	-	保留	0

---

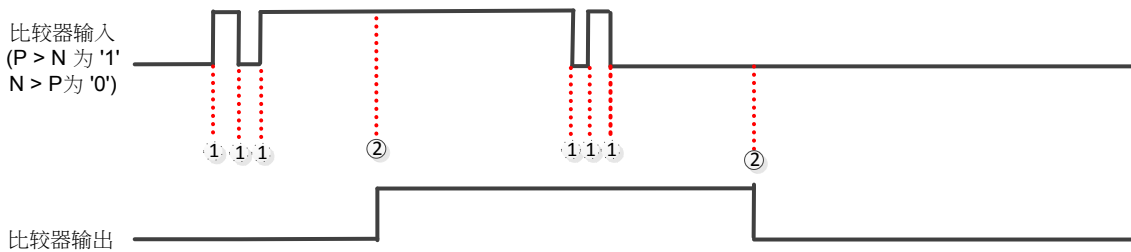
## 20.5.10 比较器去反弹计数器寄存器(DBCNT0/1/2/3 – 0x5002 4030/34/38)

表 20.5.10.1 比较器去反弹计数器寄存器(DBCNT0/1/2/3 - 0x5002 4030/34/38)位描述

位	代号	值	描述	重置值
15:0	DBCNT		比较器 x 去反弹计数器 (注意: value = 0, 没有去反弹功能) 滤波方式:使用Filter Clock 作取样的时间点, 连续2次皆一样才算是合法的输入讯号, 若输入讯号未超过两个Filter Clock取样时间则会被过滤 Filter Clock = system clock / ( DBCNTx )	0
31:16	-	-	保留	0

去反弹功能可避免由于输入噪声而导致输出变化。当输入长度小于 DBCNTx 的长度时, 信号将被忽略

当讯号由Low -> High, 或由High -> Low时,滤波计数器归0重数, 当滤波计数器 = DBCNT \* system clock 时,更新输出讯号



- ① 表示计数器重新归0
- ② 表示计数器 = DBCNT \* system clock时更新输出

## 21. 运算放大器(OPA)

OB6625器件嵌入1个运算放大器。

### 21.1 引脚描述

表 24.1.1 引脚摘要

引脚	形式	描述
OPA0_P	I	OPAMP 0 正的输入
OPA0_N	I	OPAMP 0 负的输入
OPA1_P	I	OPAMP 1 正的输入
OPA1_N	I	OPAMP 1 负的输入
OPA2_P	I	OPAMP 2 正的输入
OPA2_N	I	OPAMP 2 负的输入

### 21.2 寄存器描述

表 24.2.1 OPA 模块寄存器分布图(基地址 0x5002 4080)

名称	存取	地址 偏移	描述	重置值
OPACON	R/W	0x00	OPA 控制寄存器	

#### 21.2.1 OPAMP 控制寄存器(OPACON – 0x5002 4080)

表 24.2.1.1 OPA 控制寄存器(OPACON - 0x5002 4080) 位描述

位	代号	值	描述	重置值
0	EN0		OPA0 功能使能位	0
		0	禁用	
		1	使能 (OPACON[31]需设置为 1, IPD=1, OPA 在非掉电模式)	
2:1	OPGS0		OPA0 增益选择(R3:R2)	0
		0	16X	
		1	8X	
		2	4X	
		3	2X	
3	IPASS0		PGA 模式	
		0	PGA	
		1	bypass PGA	
7:4			保留	
8	EN1		OPA1 功能使能位	0
		0	禁用	
		1	使能	
10:9	OPGS1		OPA1 增益选择(R3:R2)	0
		0	16X	
		1	8X	
		2	4X	
		3	2X	
11	IPASS1		PGA 模式	
		0	PGA	
		1	bypass PGA	
15:12			保留	
16	EN2		OPA2 功能使能位	0
		0	禁用	
		1	使能	



18:17	OPGS2	OPA2 增益选择(R3:R2)	
	0	16X	
	1	8X	
	2	4X	
	3	2X	
19	IPASS2	PGA 模式	
	0	PGA	
	1	bypass PGA	
23:20		保留	
25:24	OPBC	偏置电流(OPA0/OPA1/OPA2 )	0
	0	0	
	1	0	
	2	16uA	
	3	32uA	
26	MODE	电机正弦控制和方形控制模式(OPA0/OPA1/OPA2 )	0
	0	正弦控制(输入电压向上偏移 2V)	
	1	方形控制(输入电压向上偏移 0.4V)	
30:27		保留	
31:31	IPD	OPA 省电模式	
	0	使能 OPA 省电(OPA 在睡眠状态)	
	1	禁用 OPA 省电(OPA 在工作状态)	

## 22. 循环冗余校验码(CRC)

### 22.1 介绍

循环冗余校验 (CRC) 发生器可以使用可编程多项式设置执行 CRC 计算。操作多项式包括 CRCCCITT, CRC-8, CRC-16 和 CRC-32; 软件可以通过设置 CTL 寄存器中的 MODE 字段来选择操作多项式模式。RC 结构支持 CPU PIO 模式。以下序列是一个程序序列示例。

当操作在 CPU PIO 模式下的步骤:

通过设置 CTL 寄存器的 EN 位来使能 CRC 结构。

初始设置, 设置数据格式 (WD, CSR, WDC 和 CSC), 初始化种子值 (SEED) 并通过设置 CTL [CRWDLEN] 寄存器选择数据长度。通过设置 CTL 寄存器中的 RST 位, 将 CRC 重置设置为将初始种子值加载到 CRC 电路。将数据写入 CRC\_WDATA 以执行 CRC 计算。通过读取 CS 寄存器获取 CRC 校验和结果。

### 22.2 特征

- 支持四个通用多项式 CRC-CCITT, CRC-8, CRC-16, and CRC-32
- CRC-CCITT:  $X^{16} + X^{12} + X^5 + 1$
- CRC-8:  $X^8 + X^2 + X + 1$
- CRC-16:  $X^{16} + X^{15} + X^2 + 1$
- CRC-32:  $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 可编程的种子值.
- 支持输入数据和 CRC 校验码的可编程顺序反向设置.
- 支持输入数据和 CRC 校验码的可编程 1 的补码设置.
- 支持 CPU PIO 模式.
- 在 CPU PIO 模式下支持 8/16/32 位数据宽度.
- 8 位写入模式: 1-AHB 时钟周期操作.
- 16 位写入模式: 2-AHB 时钟周期操作.
- 32 位 写 入 模 式 : 4-AHB 时 钟 周 期 操 作 .

## 22.3 寄存器描述

表 25.3.1 较器模块寄存器分布图(基地址 0x4010 00D0)

名称	存取	地址 偏移	描述	重置值
CTL	R/W	0x00	CRC 控制寄存器	
OCR	R/W	0x04	CRC 偏移周期寄存器	
TCR	R/W	0x08	CRC 转移周期寄存器	
DMAIER	R/W	0x0C	CRC 中断使能寄存器	
DMAISR	R/W	0x10	CRC 中断状态寄存器	
WD	R/W	0x14	CRC 写数据	
SEED	R/W	0x18	CRC 种子寄存器	
CS	R/W	0x1C	CRC 检验码寄存器	

## 22.3.1 CRC 控制寄存器(CTL – 0x4010 00D0)

表 25.3.1.1 CRC 控制寄存器(CTL - 0x4010 00D0) 位描述

位	代号	值	描述	重置值
0	EN		CRC 使能 注意1: 在CRC模式下运行时, EN将启动CRC结构进行计算	0
		0	禁用	
		1	使能	
1	RST		CRC 结构重置	0
		0	什么事情都不用做	
		1	重置 注意 1: 在 CPU PIO 模式下运行时, 设置该位将重新加载初始种子值	
3:2	-	-	保留	00
4	PREBREV		位反转输入之前的 CRC 计算结果	0
		0	禁用	
		1	使能	
19:5	-	-	保留	0
21:20	-	-	保留	00
22	ENGINEFSB		CRC 结构的第一个串行位选择	
		0	MSB	
		1	LSB	
23	MODESEL		CRC 模式选择	0
		0	CPU PIO 模式	
24	WD		写入数据顺序反向 注意: 如果写入数据为0xAABBCCDD, 则写入数据的位顺序为反向 0x55DD33BB	0
		0	CRC 写入数据时无位反转.	
		1	CRC 写入数据的位顺序反向.(per byte)	
25	CSR		校验码反转 注意: 如果校验码数据为0XDD7B0F2E, 则CRC校验码的位顺序相反 为 0x74F0DEBB	0

	0	CRC 校验码的无位顺序反向	
	1	CRC 校验码的位顺序反向	
26	WDC	写数据补码	0
	0	CRC 写入数据的无位顺序反向.	
	1	用于 CRC 写入数据的 1 的补码。	
27	CSC	校验码补码	0
	0	CRC 校验码的无位顺序反向	
	1	CRC 校验码的 1 的补码	
29:28	WDLEN	CPU 写入数据长度	0
	00	数据长度是 32 位模式	
	01	数据长度是 16 位模式	
	02	数据长度 8 位模式	
	03	保留	
31:30	MODE	CRC 多项式模式	0
	00	CRC-CCITT 多项式模式	
	01	CRC-8 多项式模式	
	02	CRC-16 多项式模式	
	03	CRC-32 多项式模式	

**22.3.2 CRC 偏移周期寄存器(OCR – 0x4010 00D4)**
**表 25.3.2.1 CRC 偏移周期寄存器(OCR - 0x4010 00D4) 位描述**

位	代号	值	描述	重置值
31:0	OCR		CRC 偏移周期寄存器	0

**22.3.3 CRC 传输周期计数寄存器(TCR – 0x4010 00D8)**
**表 25.3.3.1 CRC 传输周期计数寄存器(TCR - 0x4010 00D8) 位描述**

位	代号	值	描述	重置值
31:0	TCR		CRC 传输周期计数寄存器	0

**22.3.4 CRC 写数据寄存器(WD – 0x4010 00E4)**
**表 25.3.6.1 CRC 写数据寄存器(WD - 0x4010 00E4) 位描述**

位	代号	值	描述	重置值
31:0	WD		在CPU PIO模式下运行时，软件可以将数据写入该字段以执行CRC 注意2: CTL [WDC]和CTL [WD]位设置将影响此字段; 例如，如果CTL [WD] = 1，则CTL [WD]寄存器中的写入数据为0xAABBCCDD。	

**22.3.5 CRC 种子寄存器(SEED – 0x4010 00E8)**
**表 25.3.7.1 CRC 种子寄存器(SEED - 0x4010 00E8) 位描述**

位	代号	值	描述	重置值
31:0	SEED		CRC 种子寄存器	

**22.3.6 CRC 校验码寄存器(CS – 0x4010 00EC)**
**表 25.3.8.1 CRC 校验码寄存器(CS - 0x4010 00EC) 位描述**

位	代号	值	描述	重置值
31:0	CS		CRC 校验码寄存器	

## 23. 内部电压(IV)

### 23.1 特征

输出 1/2 VDD 电压

### 23.2 功能说明

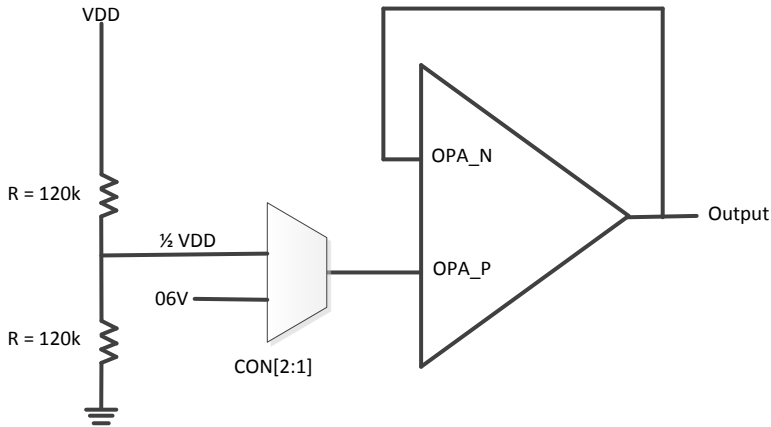


图 25.3 显示比较器框图

### 23.3 引脚说明

Table 25.4.1 引脚摘要

引脚	类型	描述
IV_O	O	内部电压输出

### 23.4 寄存器描述

Table 25.5.1 比较器模块寄存器映射(基址 0x5002 0800)

名称	存取	地址 偏移	描述	重置值
IVCON	R/W	0x00	内部电压控制寄存器	

#### 23.4.1 内部电压控制寄存器(IVCON – 0x5002 00FC)

Table 25.5.1.1 TS 控制寄存器(IVCON - 0x5002 0800) 位描述

位	代号	值	描述	重置值
0	EN0		内部电压	0
		0	禁用	
		1	使能	
2:1	VSEL		内部电压选择	0
		0	1/2 VDD	

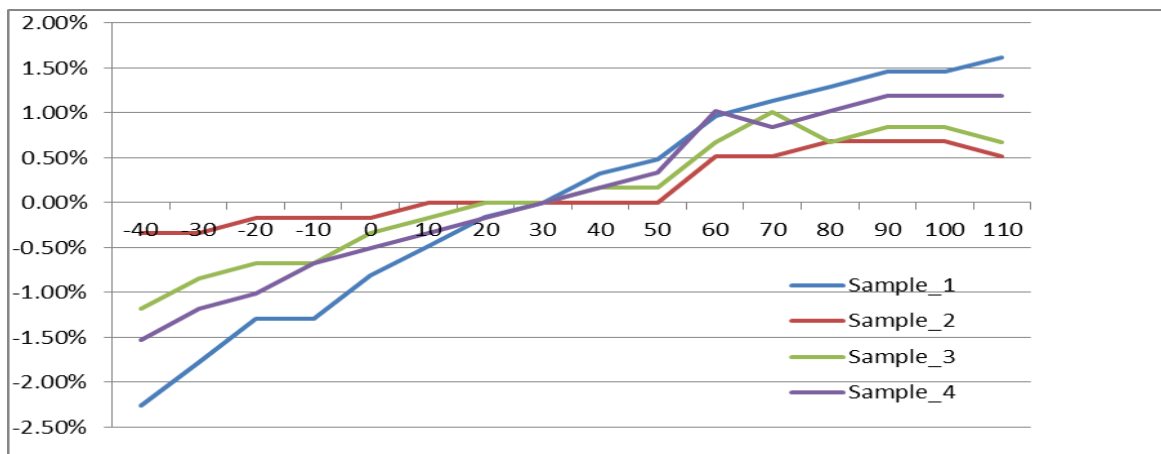
## 24. 电气特性

### 24.1 绝对最大额定值

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
VDD - VSS	DC Power Supply		-0.3		7	V
Vin	Input Voltage		VSS-0.3		VDD+0.3	V
TA	Operating Temperature		-40		+105	°C
TJ	Junction Temperature		-40		+125	°C
TST	Storage Temperature		-55		+150	°C

### 24.2 运行条件

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
VDD	Supply Voltage	VDD = 1.8V ~5.5V, up to 72 MHz	1.8		5.5	V
VSS	Power Ground		-0.3			V
VBG	Band gap Voltage	VDD = 1.8V ~5.5V, Temp.= -40°C ~ +105°C		0.6		V
		TA = 25°C	-1		+1	%
		TA = -40°C ~ +105°C	-2		+2	%



VBG 精度与温度曲线图

## 24.3 DC 电气特性

GPIO 0/1/2						
Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
VIL	Input Low Voltage	VDD = 5.0V	-0.3		1/2VDD-0.3	V
		VDD = 3.0V	-0.3		1/2VDD-0.3	V
		VDD = 5.0V (SWD Pin)	-0.3		1/2VDD-0.7	V
		VDD = 3.0V (SWD Pin)	-0.3		1/2VDD-0.7	V
VIH	input High Voltage	VDD = 5.0V	1/2VDD		VDD+0.3	V
		VDD = 3.0V	1/2VDD		VDD+0.3	V
		VDD = 5.0V (SWD Pin)	1/2VDD		VDD+0.7	V
		VDD = 3.0V (SWD Pin)	1/2VDD		VDD+0.7	V
IOH	Source Current (Pull – up Mode)	VDD = 5.0V, Vo = 4.6V		-75		uA
		VDD = 3.0V, Vo = 2.6V		-45		uA
IOH1	Source Current (Push – pull Mode)	VDD = 5.0V, Vo = 4.6V	-10			mA
		VDD = 3.0V, Vo = 2.6V	-7			mA
		VDD = 5.0V, Vo = 4.6V (PWM Pin)	-14			mA
		VDD = 3.0V, Vo = 2.6V (PWM Pin)	-9			mA
IOL	Sink Current	VDD = 5.0V, Vo = 0.4V	30			mA
		VDD = 3.0V, Vo = 0.4V	20			mA
		VDD = 5.0V, Vo = 0.4V (PWM Pin)	45			mA
		VDD = 3.0V, Vo = 0.4V (PWM Pin)	30			mA
ILI	Input Leakage Current	Open-drain or input only mode	-1		1	uA

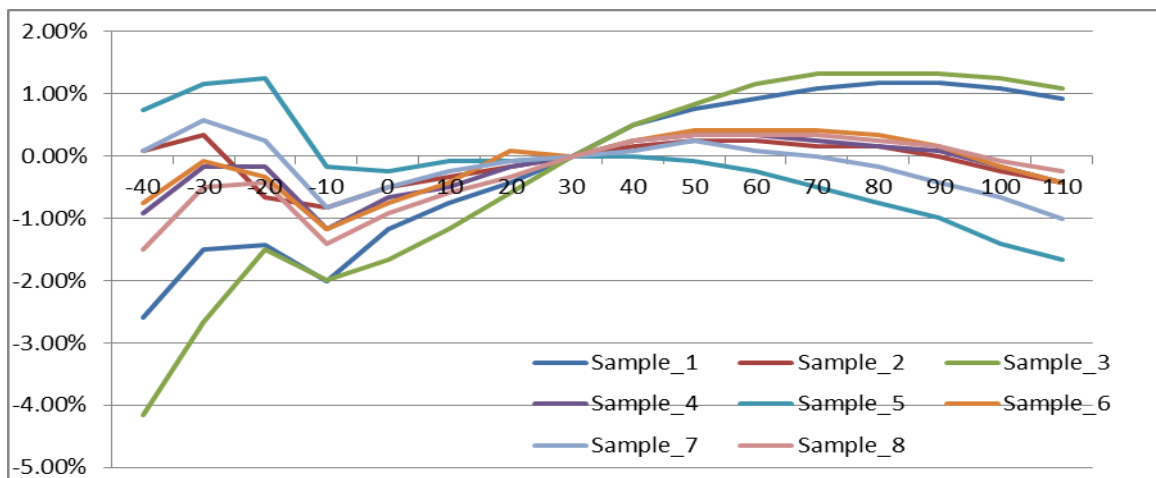


Supply pins						
Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
loc	Operating Current Normal Run Mode While(1);	DPLL = 72M Hz, VDD=5.0V 关闭 LVI/LVR		12		mA
		DPLL = 72M Hz, VDD=3.0V 关闭 LVI/LVR		12		mA
		DPLL = 48M Hz, VDD=5.0V 关闭 LVI/LVR		10.5		mA
		DPLL = 48M Hz, VDD=3.0V 关闭 LVI/LVR		10.5		mA
		HIRC = 24M Hz, VDD=5.0V 关闭 LVI/LVR		4.5		mA
		HIRC = 24M Hz, VDD=3.0V 关闭 LVI/LVR		4.5		mA
IIDLE	Standby Current Power-down Mode (Sleep Mode)_	HIRC = 24M Hz, VDD=5.0V 关闭 LVI/LVR		3.9		mA
		HIRC = 24M Hz, VDD=3.0V 关闭 LVI/LVR		3.9		mA
icc	Standby Current Power-down Mode (Deep Sleep Mode)_	VDD = 5.0		2		uA
		VDD = 3.0		2		uA

## 24.4 AC 电气特性

### 24.4.1 Internal High Speed RC Oscillator (HIRC)

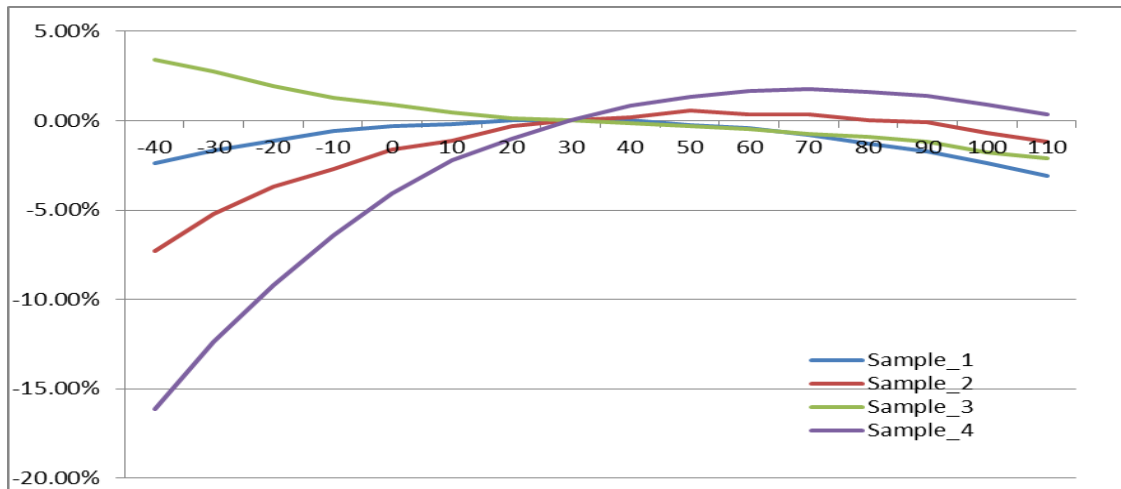
Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
FHIRC	Center Frequency			24		MHz
	Calibrated Internal Oscillator Frequency	TA = 25°C	-1		+1	%
		TA = -40°C ~ +105°C	-3		+3	%



HIRC 精度与温度曲线图

### 24.4.2 20 KHz Internal RC Oscillator (LIRC1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
FLIRC1	Center Frequency			20		KHz
	Calibrated Internal Oscillator Frequency	TA = 25°C	-15		+15	%
		TA = -40°C ~ +105°C	-30		+30	%



LIRC 精度与温度曲线图

### 24.4.3 SAR ADC

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
	Resolution				10 12	Bit
	Differential Nonlinearity Error			+/- 1	+/- 1.5	LSB
	Integral Nonlinearity Error			+/- 1	+/- 2	LSB
	Offset Error	T A = 25°C , AV DD = 5V @ 12bit resolution		3		LSB
	Gain Error (Transfer Gain)	T A = 25°C , AV DD = 5V @ 12bit resolution		1		LSB
	ADC Clock Frequency	AVDD = 3.0~5.5V		1		MHz
	Supply Voltage		3.0		5.5	V
	Supply Current (Avg.)	AVDD = 5.0V @1M Hz conversion rate		2.6		mA
	Analog Input Voltage		0		AVDD	V
	Input Capacitance			6.4		pF
	Input Load			6		kΩ

## 24.4.4 Low Voltage Reset

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
	Threshold Voltage	TA = 25 °C	1.5		1.7	V

## 24.4.5 Low Voltage Interrupt

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
	Threshold Voltage	LVI_SEL [1:0]=11	3.8		4.3	V
		LVI_SEL [1:0]=10	3.0		3.5	V
		LVI_SEL [1:0]=01	2.4		2.9	V
		LVI_SEL [1:0]=00	1.5		2.0	V

## 24.4.6 Comparator

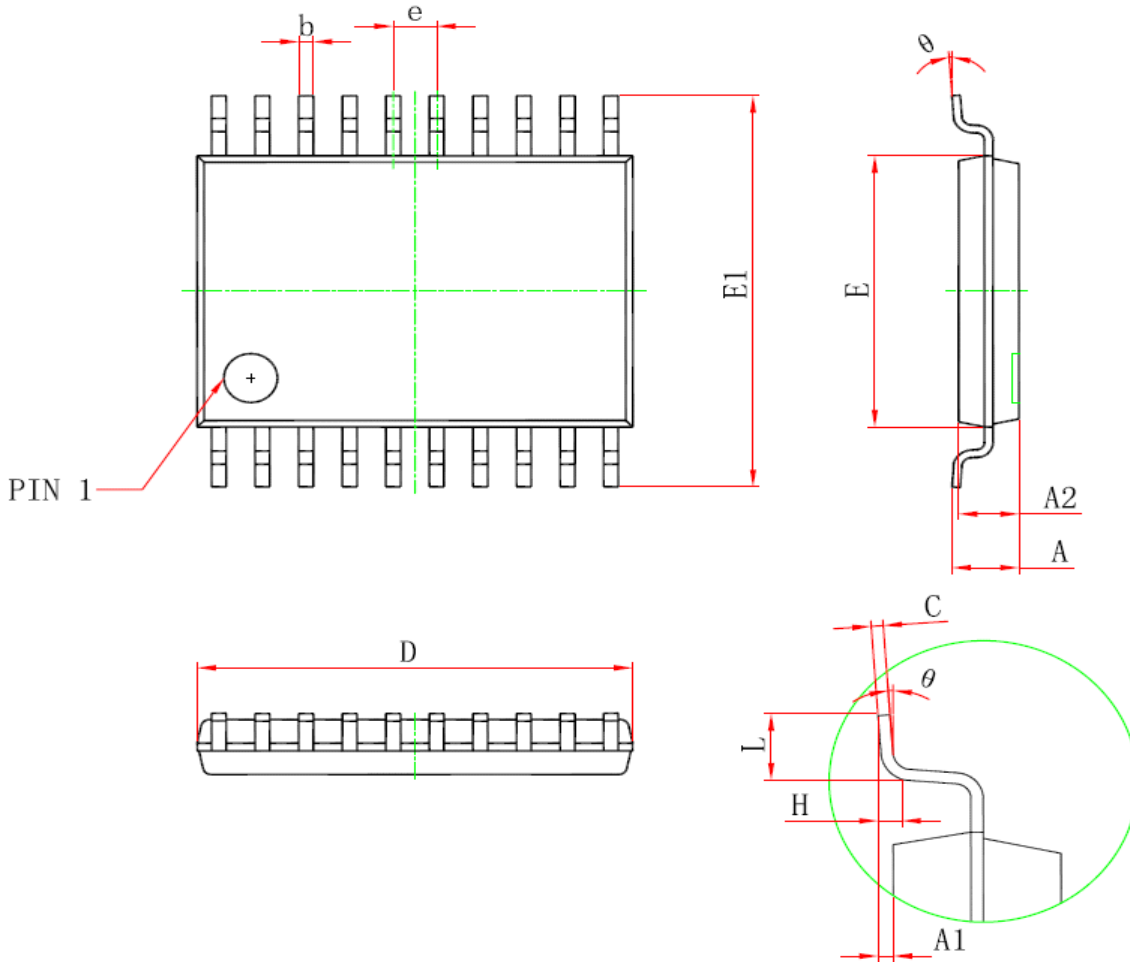
Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
	Operation Current	AVDD = 5V		110		uA
		AVDD = 3.0V		100		uA
	Input Offset Voltage	Hysteresis = 0		10		mV
	Input Common Mode Range		0.1		VDD-0.1	V
	DC Gain			60		db
	Propagation Delay				500	ns
	Hysteresis		0		400	mV

## 24.4.7 Internal Voltage

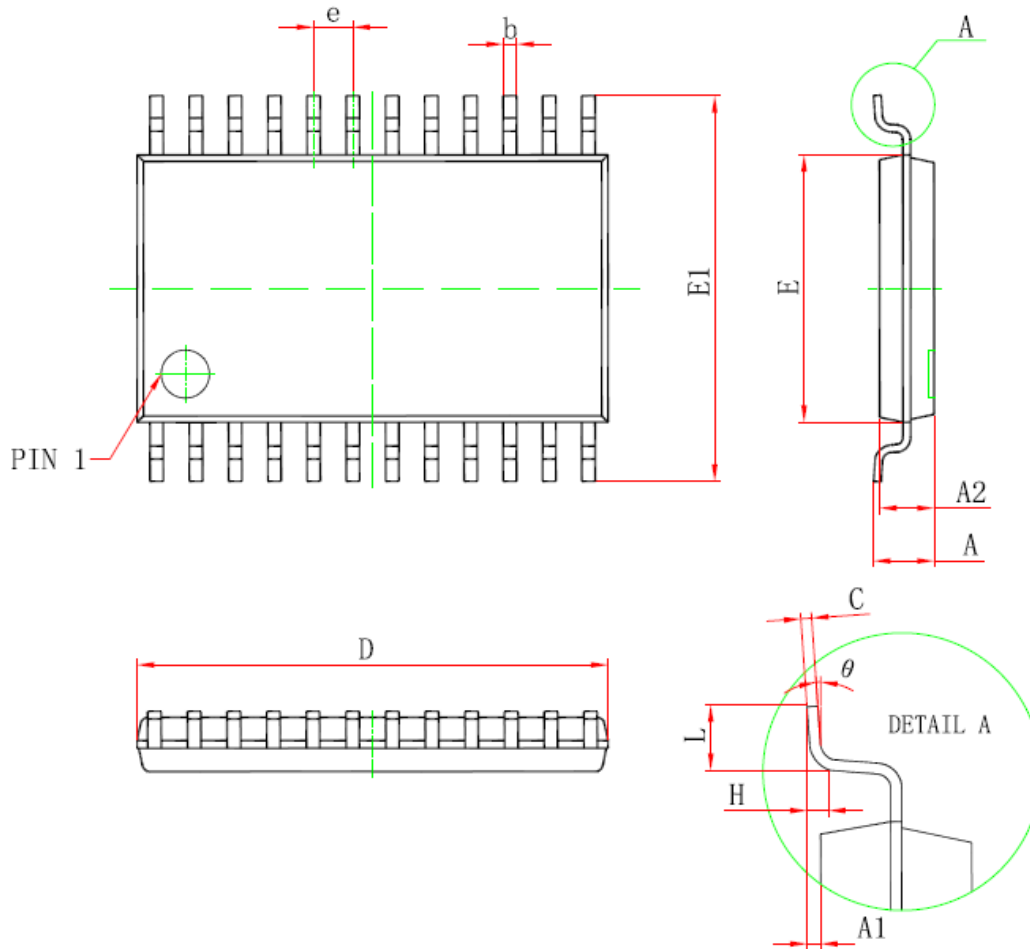
Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V <sub>BG</sub>	Band gap Voltage	TA = 25°C	-1%	0.6	1%	V
		TA = -40°C ~ +85°C,	-3%	0.6	3%	V
	Internal Voltage	TA = 25°C	-1%	1/2 V <sub>DD</sub>	1%	V

## 24.4.8 OPAMP

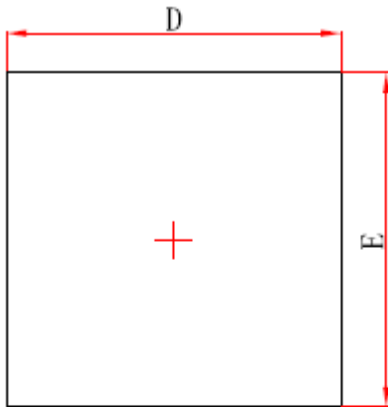
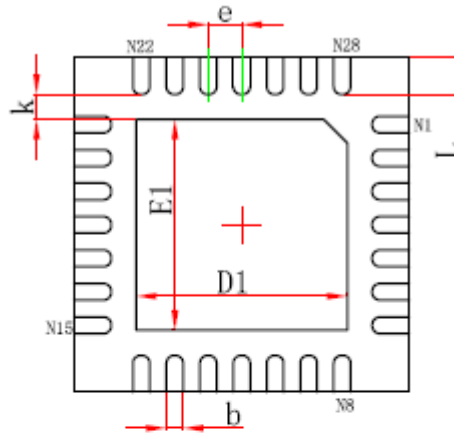
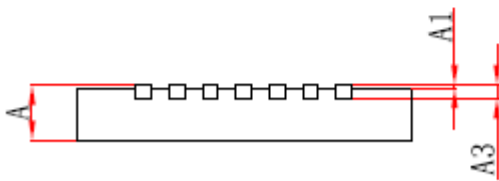
Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
VDD	OPAmP operating voltage		2.7		5.5	V
V <sub>IO</sub>	Input offset voltage		-3		3	mV
V <sub>ICM</sub>	Common mode Input voltage		V <sub>SS</sub> + 0.2		V <sub>DD</sub> - 1.1	V
V <sub>OSW</sub>	Output swing voltage		V <sub>SS</sub> + 0.1		V <sub>DD</sub> - 0.1	V
PSRR	Power supply rejection ratio			65		dB
CMRR	Common mode noise rejection ratio			60		dB
SR	Slew rate	C <sub>L</sub> =10pf	10			V/us
GBW	Gain bandwidth product	C <sub>L</sub> =10pf	6			Mhz

**25. 包装尺寸**  
**20-Pin Plastic TSSOP (TSSOP20)**


Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
D	6.400	6.600	0.252	0.259
E	4.300	4.500	0.169	0.177
b	0.190	0.300	0.007	0.012
c	0.090	0.200	0.004	0.008
E1	6.250	6.550	0.246	0.258
A	0.950	1.200	0.037	0.047
A2	0.800	1.050	0.031	0.041
A1	0.020	0.150	0.001	0.006
e	0.650(BSC)		0.026(BSC)	
L	0.450	0.750	0.018	0.030
H	0.250(TYP)		0.01(TYP)	
θ	0°	8°	0°	8°

**24-Pin Plastic TSSOP (TSSOP24)**


Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
D	7.700	7.900	0.303	0.311
E	4.300	4.500	0.169	0.177
b	0.190	0.300	0.007	0.012
c	0.090	0.200	0.004	0.008
E1	6.250	6.550	0.246	0.258
A	-	1.200	-	0.047
A2	0.800	1.000	0.031	0.039
A1	0.020	0.150	0.001	0.006
e	0.65(BSC)		0.026(BSC)	
L	0.450	0.750	0.018	0.030
H	0.25(TYP)		0.01(TYP)	
$\theta$	0°	8°	0°	8°

**28-Pin Plastic QFN (QFN28)**

**Top View**

**Bottom View**

**Side View**

Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	0.700	0.800	0.028	0.031
A1	-	0.050	-	0.002
A3	0.203(BSC)		0.008(BSC)	
b	0.180	0.300	0.007	0.012
D	4.900	5.100	0.193	0.201
D1	3.050	3.650	0.120	0.144
E	4.900	5.100	0.193	0.201
E1	3.050	3.650	0.120	0.144
e	0.500(BSC)		0.020(BSC)	
L	0.300	0.626	0.012	0.025
k	0.200	-	0.008	-



## **IMPORTANT NOTICE**

### **RIGHT TO MAKE CHANGES**

On-Bright Electronics Corp. reserves the right to make corrections, modifications, enhancements, improvements and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

### **WARRANTY INFORMATION**

On-Bright Electronics Corp. warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with its standard warranty. Testing and other quality control techniques are used to the extent it deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

On-Bright Electronics Corp. assumes no liability for application assistance or customer product design. Customers are responsible for their products and applications using On-Bright's components, data sheet and application notes. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

### **LIFE SUPPORT**

On-Bright Electronics Corp.'s products are not designed to be used as components in devices intended to support or sustain human life. On-bright Electronics Corp. will not be held liable for any damages or claims resulting from the use of its products in medical applications.

### **MILITARY**

On-Bright Electronics Corp.'s products are not designed for use in military applications. On-Bright Electronics Corp. will not be held liable for any damages or claims resulting from the use of its products in military applications.