

## 特性

- 精简指令集架构
- 8 层硬件堆栈 x12bit
- 2T 或 4T 指令周期
- 4Kx14b 程序存储空间
  - i. 程序存储空间的 checksum 自动校验
  - ii. 可配置, User Option
- 256x8b 数据 EEPROM
  - i. 数据 EEPROM 在应用编程
  - ii. 可配置, Factory Option
- 高耐用性 EEPROM
  - i. 程序和数据 EEPROM 可经受 100 万次写操作
  - ii. EEPROM 保存时间>40 年
- 256x8b SRAM
- 1 x 带 8 位预分频的定时器 0
- 1 x 带 3 位预分频的 16 位定时器 1
- 1 x 带 8 位预分频的定时器 2
  - i. 慢时钟周期测量模式
- 增强性捕捉、比较和可编程“死区”时间的 PWM 模块
  - i. 时钟源可选: 系统时钟或者是内部 32MHz 时钟
  - ii. 单次脉冲模式
  - iii. 最多 3 对带“死区”的 PWM 输出
- 4x12bit Timer, 4x12bit PWM, 支持 BUZZER 模式
  - i. 支持每组 PWM 的互补脉冲输出
  - ii. 时钟最快 32M
  - iii. 外设时钟输出
- 1x9bit 可编程脉冲发生器 (PPG)
  - i. 两个重载寄存器
  - ii. 脉冲极性可选择
  - iii. 支持脉宽限制
  - iv. 支持不可重复触发模式
  - v. 手动触发方式和比较器结果触发方式
- 带 7 位预分频的 WDT, 溢出频率范围为 16ms-256s
- 上电延迟计数器 PWRT
- 低功耗模式 SLEEP
- 多个唤醒源, 外部中断 INT、端口变化中断、WDT 和数据 EEPROM 写完成, 等等
  - i. 可配置硬件去抖的外部中断 INT
- 内置高速 16M RC 振荡器
- 内置低速 32K RC 振荡器
- 支持外部晶振 16M 或 32K, 以及外部时钟模式
  - i. 时钟缺失检测
  - ii. 双速启动模式
- 内置 10 位的 ADC, 支持 8 个通道 (7 个外部通道 + 1 个内部 1/4VDD 通道)
  - i. 参考电压可选: 外部 Vref, VDD, 内部 2V/3V
  - ii. 可配置, Factory Option
- 内置 6 个高速高精度比较器
  - i. 可编程的参考电压
  - ii. 比较结果可直接输出
  - iii. 比较结果可配置去抖
  - iv. 输入失调电压可校准
  - v. 迟滞控制 (比较器 0/1/2/3)
- 1x 运算放大器, 支持软件校准输入失调电压
- 低电压复位 LVR:
  - i. 2.0/2.2/2.5/2.8/3.1/3.6/4.1V
- 低电压检测 LVD:
  - i. 2.0/2.4/2.8/3.0/3.6/4.0V
- 3 对 USB 充电端口:
  - i. 专用充电端口 DC+/DC-
  - ii. 充电下行端口 DA+/DA-, DB+/DB-
  - iii. 支持 QC2.0 和 QC3.0 以及苹果、三星设备
  - iv. 最多同时对两个设备充电
  - v. 集成 D2I 模块
- 最多 18 个通用 IO, 20 个芯片管脚

- i. 所有 IO 带独立上拉控制
- ii. 4 个 IO 带独立下拉控制
  - 下降沿中断, RA0-RA7
  - 支持在系统编程 ICSP
  - 支持在线调试, 3 个硬件断点
  - 程序空间保护
  - 工作电压范围: 2.0V- 5.5V
  - 最大时钟工作频率: 16MHz
- i.  $F_{SYS}=8\text{MHz}$ (2T mode):  
2.0V-5.5V
- ii.  $F_{SYS}=16\text{MHz}$ (2T mode):  
2.7V-5.5V
- 封装类型: SOP14, TSSOP14,  
SOP16, SOIC20, SSOP20, PDIP20,  
QFN20

## 目录

特性.....	- 1 -
芯片版本历史.....	12
1 系统功能框图和程序存储器、脚位图.....	13
1.1 系统功能框图.....	13
1.2 程序存储器.....	13
1.3 脚位图.....	15
2 数据存储器.....	21
2.1 数据存储区构成.....	21
2.2 寄存器地址映射.....	21
2.2.1 SFR, BANK0.....	21
2.2.2 SFR, BANK1.....	22
2.2.3 SFR, BANK2.....	23
2.2.4 SFR, BANK3.....	24
2.2.5 STATUS 寄存器, 地址 0x03, 0x83, 0x103, 0x183.....	26
2.2.6 配置寄存器 UCFGx.....	27
2.2.7 PCL 和 PCLATH.....	31
2.2.8 INDF 和 FSR 寄存器.....	31
2.2.9 关于寄存器保留位.....	32
3 系统时钟源.....	33
3.1 时钟源模式.....	33
3.2 外部时钟模式.....	34
3.2.1 振荡器起振定时器 (OST).....	34
3.2.2 EC 模式.....	34
3.2.3 LP 和 XT 模式.....	34
3.3 内部时钟模式.....	34
3.3.1 频率选择位 (IRCF).....	35
3.3.2 HFINTOSC 和 LFINTOSC 时钟切换时序.....	35
3.4 时钟切换.....	36
3.4.1 系统时钟选择 (SCS) 位.....	36
3.4.2 振荡器起振超时状态 (OSTS) 位.....	36
3.5 双速时钟启动模式.....	36
3.5.1 双速启动模式配置.....	37
3.5.2 双速启动顺序.....	37
3.6 故障保护时钟监控器.....	37
3.6.1 故障保护检测.....	38
3.6.2 故障保护操作.....	38
3.6.3 故障保护条件清除.....	38
3.6.4 复位或从休眠中唤醒.....	38
3.7 与时钟源相关寄存器汇总.....	39
3.7.1 OSCCON 寄存器, 地址 0x8F.....	39
4 复位时序.....	41

4.1	POR 上电复位 .....	42
4.2	外部复位 MCLR .....	42
4.3	PWRT (上电计时器) .....	42
4.4	BOR 低电压复位.....	43
4.4.1	频率与最小工作电压的关系.....	43
4.5	看门狗复位.....	44
4.6	错误指令复位.....	44
4.7	LVD 低电压侦测.....	44
4.8	超时动作.....	44
4.9	上电配置过程.....	45
4.10	上电校验过程.....	45
4.11	PCON 寄存器, 地址 0x8E.....	47
5	看门狗定时器.....	49
5.1	看门狗.....	49
5.2	与看门狗相关寄存器汇总.....	49
5.2.1	WDTCON 寄存器, 地址 0x18 .....	50
6	定时器 0 .....	51
6.1	Timer 0.....	51
6.2	Timer 0 定时器模式 .....	51
6.3	Timer 0 计数器模式 .....	51
6.3.1	软件可配置预分频电路.....	52
6.3.2	定时器 0 中断.....	53
6.3.3	用外部时钟驱动定时器 0.....	53
6.4	与 TIMER0 相关寄存器汇总 .....	53
6.4.1	OPTION 寄存器, 地址 0x81 .....	53
6.4.2	TMR0, 地址 0x01 .....	54
7	带门控的 TIMER1 模块 .....	55
7.1	Timer1 工作原理 .....	55
7.2	选择时钟源.....	56
7.2.1	内部时钟源.....	56
7.2.2	外部时钟源.....	56
7.3	Timer1 预分频器 .....	57
7.4	Timer1 振荡器 .....	57
7.5	Timer1 工作于异步计数器模式下 .....	57
7.5.1	在异步计数器模式下读写 TIMER1 .....	57
7.6	TIMER1 门控 .....	58
7.7	Timer1 中断 .....	58
7.8	Timer1 在休眠模式下的工作原理 .....	58
7.9	ECCP 捕捉/比较时基.....	58
7.10	ECCP 特殊事件触发器.....	59
7.11	与 TIMER1 相关寄存器汇总 .....	59
7.11.1	T1CON 寄存器, 地址 0x10.....	59
7.11.2	TMR1L 寄存器, 地址 0x0E.....	60

7.11.3	TMR1H 寄存器, 地址 0x0F .....	61
8	定时器 2 .....	62
8.1	Timer2 工作原理 .....	62
8.2	PPG 禁止重复触发的定时 .....	63
8.3	与 TIMER2 相关寄存器汇总 .....	63
8.3.1	PR2 寄存器, 地址 0x92 .....	64
8.3.2	TMR2 寄存器, 地址 0x11 .....	64
8.3.3	T2CON 寄存器, 地址 0x12 .....	64
9	增强型捕捉/比较/PWM 模块 .....	66
9.1	捕捉模式 .....	66
9.1.1	CCP1 引脚配置 .....	67
9.1.2	TIMER1 模式选择 .....	67
9.1.3	软件中断 .....	67
9.1.4	CCP 预分频器 .....	67
9.2	比较模式 .....	67
9.2.1	CCP1 引脚的配置 .....	68
9.2.2	TIMER1 模式选择 .....	68
9.2.3	软件中断模式 .....	68
9.2.4	特殊事件触发器 .....	68
9.3	PWM 模式 .....	69
9.3.1	PWM 周期 .....	70
9.3.2	PWM 占空比 .....	70
9.3.3	PWM 分辨率 .....	71
9.3.4	休眠模式下的工作 .....	71
9.3.5	系统时钟频率的改变 .....	71
9.3.6	复位的影响 .....	72
9.3.7	PWM 工作的设置 .....	72
9.4	PWM (增强模式) .....	72
9.4.1	半桥模式 .....	74
9.4.2	全桥模式 .....	77
9.4.3	启动考虑事项 .....	81
9.4.4	增强型 PWM 自动关闭模式 .....	81
9.4.5	自动重启模式 .....	82
9.4.6	可编程死区延时模式 .....	83
9.5	PWM 的辅助功能 .....	83
9.5.1	一次性脉冲模式 .....	84
9.5.2	3 对 PWM 信号输出 .....	85
9.5.3	PWM 辅助功能的使用 .....	86
9.6	与 ECCP 相关寄存器汇总 .....	86
9.6.1	CCPR1L 寄存器, 地址 0x13 .....	87
9.6.2	CCPR1H 寄存器, 地址 0x14 .....	87
9.6.3	CCP1CON 寄存器, 地址 0x15 .....	87
9.6.4	PWM1CON 寄存器, 地址 0x16 .....	88

9.6.5	ECCPAS 寄存器, 地址 0x17 .....	89
9.6.6	PWM1AUX 寄存器, 地址 0x90 .....	89
10	比较器模块 .....	91
10.1	比较器 0/1/2/3 .....	91
10.1.1	比较器概述 .....	91
10.1.2	比较器的配置及结构框图 .....	92
10.1.3	模拟输入开关 .....	94
10.1.4	比较器中断 .....	95
10.1.5	比较器的参考电压 .....	95
10.1.6	比较器输出的去抖 .....	96
10.2	比较器 4/5 .....	96
10.2.1	比较器配置 .....	97
10.2.2	模拟输入开关 .....	98
10.2.3	比较器 5 门控 Timer1 .....	99
10.2.4	比较器 5 输出与 Timer1 同步 .....	99
10.2.5	比较器中断 .....	99
10.2.6	比较器的参考电压 .....	100
10.3	模拟输入端的连接 .....	102
10.4	比较器控制 .....	102
10.5	模拟输出状态 .....	103
10.6	模拟输出极性 .....	103
10.7	比较器反应时间 .....	103
10.8	比较器在睡眠状态下的工作 .....	103
10.9	比较器在复位状态下的工作 .....	103
10.10	校准输入失调电压 .....	104
10.11	与比较器相关寄存器汇总 .....	104
10.11.1	CMCON0 寄存器, 地址 0x19 .....	105
10.11.2	CMCON2/3 寄存器, 地址 0x101, 0x107 .....	106
10.11.3	CMCON1 寄存器, 地址 0x1A .....	107
10.11.4	CMCON4 寄存器, 地址 0x11F .....	107
10.11.5	CMPxCR 寄存器, 地址 0x190~0x193 .....	108
10.11.6	VRCON 寄存器, 地址 0x99 .....	109
11	数据 EEPROM .....	110
11.1	编程数据 EEPROM 步骤 .....	110
11.2	关于编程周期 .....	111
11.3	读数据 EEPROM .....	112
11.4	与数据 EEPROM 相关寄存器汇总 .....	112
11.4.1	EEDAT 寄存器, 地址 0x9A .....	112
11.4.2	EEADR 寄存器, 地址 0x9B .....	113
11.4.3	EECON1 寄存器, 地址 0x9C .....	113
11.4.4	EECON2 寄存器, 地址 0x9D .....	113
12	模数转换器 (ADC) 模块 .....	115
12.1	ADC 的配置 .....	115

12.1.1	端口配置.....	115
12.1.2	通道选择.....	116
12.1.3	ADC 参考电压 .....	116
12.1.4	转换时钟.....	116
12.1.5	中断.....	117
12.1.6	转换结果的格式.....	117
12.2	ADC 的工作原理 .....	118
12.2.1	启动转换.....	118
12.2.2	转换完成.....	118
12.2.3	终止转换.....	118
12.2.4	休眠模式下 ADC 的工作 .....	119
12.2.5	特殊事件触发器.....	119
12.2.6	A/D 转换步骤.....	119
12.3	A/D 采集时间要求.....	121
12.4	与 ADC 相关寄存器汇总 .....	121
12.4.1	ADCON0 寄存器, 地址 0x1F .....	122
12.4.2	ADCON1 寄存器, 地址 0x9F .....	123
12.4.3	ADRESH 寄存器, 地址 0x1E .....	123
12.4.4	ADRESL 寄存器, 地址 0x9E .....	124
12.4.5	ANSEL 寄存器, 地址 0x91 .....	125
13	慢时钟测量.....	126
13.1	使用方法.....	126
13.2	与慢时钟测量相关寄存器汇总 .....	127
13.2.1	MSCKCON 寄存器, 地址 0x1B .....	128
13.2.2	SOSCPR 寄存器, 地址 0x1C, 1D.....	128
14	中断模式.....	130
14.1	INT 中断.....	130
14.1.1	硬件去抖.....	131
14.2	PORTA 下降沿中断 .....	132
14.3	中断响应.....	132
14.4	中断过程中的现场保存.....	133
14.5	与中断相关寄存器汇总.....	133
14.5.1	INTCON 寄存器, 地址 0x0B .....	134
14.5.2	PIR1 寄存器, 地址 0x0C.....	135
14.5.3	PIR2 寄存器, 地址 0x0D.....	135
14.5.4	PIE1 寄存器, 地址 0x8C.....	136
14.5.5	PIE2 寄存器, 地址 0x8D.....	137
14.5.6	IOFA 寄存器, 地址 0x96 .....	138
14.5.7	DEBCR 寄存器, 地址 0x181 .....	138
15	睡眠省电模式.....	139
15.1	唤醒模式.....	139
15.2	看门狗唤醒.....	139
15.3	睡眠状态下运行系统时钟.....	139

16	I/O 端口 .....	140
16.1	PORTA 端口和 TRISA 寄存器 .....	140
16.2	端口的其他功能 .....	140
16.2.1	弱上拉 .....	140
16.2.2	弱下拉 .....	140
16.2.3	ANSEL 寄存器 .....	140
16.2.4	下降沿中断 .....	141
16.3	数字输出在管脚上的复用 .....	141
16.4	模拟功能管脚的使用 .....	143
16.5	端口描述 .....	144
16.5.1	PORTA[0] .....	144
16.5.2	PORTA[1] .....	145
16.5.3	PORTA[2] .....	146
16.5.4	PORTA[3] .....	147
16.5.5	PORTA[4] .....	148
16.5.6	PORTA[5] .....	149
16.5.7	PORTA[6] .....	150
16.5.8	PORTA[7] .....	151
16.5.9	PORTC[0] .....	152
16.5.10	PORTC[1] .....	152
16.5.11	PORTC[2] .....	153
16.5.12	PORTC[3] .....	155
16.5.13	PORTC[4] .....	156
16.5.14	PORTC[5] .....	157
16.5.15	PORTB[0] .....	157
16.5.16	PORTB[1] .....	158
16.5.17	PORTB[2] .....	159
16.5.18	PORTB[3] .....	160
16.6	与 GPIO 相关寄存器汇总 .....	162
16.6.1	PORTA 寄存器, 地址 0x05 .....	162
16.6.2	PORTB 寄存器, 地址 0x06 .....	163
16.6.3	PORTC 寄存器, 地址 0x07 .....	163
16.6.4	TRISA 寄存器, 地址 0x85 .....	163
16.6.5	TRISB 寄存器, 地址 0x86 .....	163
16.6.6	TRISC 寄存器, 地址 0x87 .....	164
16.6.7	WPUA 寄存器, 地址 0x95 .....	164
16.6.8	WPUC 寄存器, 地址 0x88 .....	165
16.6.9	WPD 寄存器, 地址 0x89 .....	165
16.6.10	WPUB 寄存器, 地址 0x106 .....	166
17	PWMx 和 TIMERx .....	167
17.1.1	TIMERx 工作原理 .....	167
17.1.2	时钟源选择 .....	168
17.1.3	TIMERx 时钟分频 .....	169



17.1.4	TIMERx 中断 .....	169
17.1.5	TIMERx 预分频 .....	169
17.1.6	TIMERx 周期 .....	169
17.1.7	TIMERx 在休眠模式下工作 .....	170
17.1.8	读写 TMRxH/L 寄存器 .....	170
17.2	BUZZER 工作模式 .....	170
17.2.1	BUZZER 周期 .....	170
17.2.2	BUZZER 在休眠模式下工作 .....	171
17.3	PWM 工作模式 .....	171
17.3.1	PWM 周期 .....	171
17.3.2	PWM 占空比 .....	172
17.3.3	PWM 工作设置 .....	172
17.3.4	PWM 工作在休眠模式 .....	172
17.3.5	PWM 脉冲的互补输出 .....	173
17.4	PWMx 相关寄存器 .....	173
17.4.1	EPWMCR0 寄存器, 地址 0x11E .....	173
17.4.2	EPWMCR1 寄存器, 地址 0x105 .....	174
17.4.3	TMRxL 寄存器 .....	174
17.4.4	TMRxH 寄存器 .....	175
17.4.5	PRxL 寄存器 .....	175
17.4.6	PWMxCR0 寄存器 .....	175
17.4.7	PWMxCR1 寄存器 .....	176
17.4.8	TxCKDIV 寄存器 .....	177
18	可编程脉冲生成器 (PPG) .....	178
18.1	基本工作原理 .....	178
18.2	PPG 管脚的状态 .....	179
18.3	禁止重复触发 .....	179
18.4	脉宽限制 .....	180
18.5	PPG 脉冲的同步 .....	180
18.6	有效触发事件 .....	181
18.7	停止 PPG 事件 .....	182
18.8	PPGT 的重载 .....	182
18.9	延迟触发 .....	182
18.9.1	基本工作原理 .....	182
18.10	PPG 工作设置步骤 .....	184
18.11	与 PPG 相关寄存器汇总 .....	184
18.11.1	PPGCON1 寄存器, 地址 0x188 .....	184
18.11.2	PPRH 寄存器, 地址 0x189 .....	185
18.11.3	PPRA 寄存器, 地址 0x18C .....	186
18.11.4	PPRB 寄存器, 地址 0x18D .....	186
18.11.5	PWLTPRL 寄存器, 地址 0x18E .....	186
18.11.6	PPGCON2 寄存器, 地址 0x18F .....	186
18.11.7	PPGCON3 寄存器, 地址 0x19E .....	187

19	运算放大器.....	188
19.1	运放 0.....	188
19.1.1	校准输入失调电压.....	188
19.1.2	运放 0 典型应用.....	189
19.2	复位的影响.....	190
19.3	休眠模式下的影响.....	190
19.4	与运放 0 相关寄存器汇总.....	190
19.4.1	OP0CR0 寄存器, 地址 0x194.....	191
19.4.2	UCFG4 配置寄存器.....	191
20	USB 充电口.....	192
20.1	USBC 端口.....	192
20.1.1	充电检测.....	192
20.1.2	应用框图.....	193
20.2	USBA 和 USBB 端口.....	194
20.2.1	基本工作原理.....	195
20.2.2	USBA 应用框图.....	196
20.2.3	USBB 应用框图.....	198
20.3	DAC 模块.....	198
20.4	与 USB 口相关寄存器汇总.....	199
20.4.1	DACOE 寄存器, 地址 0x93.....	199
20.4.2	DACCON0 寄存器, 地址 0x94.....	200
20.4.3	VCON2 寄存器, 地址 0x109.....	201
20.4.4	D2ICON 寄存器, 地址 0x97.....	201
21	芯片的电气特性.....	202
21.1	绝对极限参数.....	202
21.2	内置高频振荡器(HFINTOSC).....	202
21.3	内置低频振荡器(LFINTOSC).....	202
21.4	低电压复位电路(LVR).....	203
21.5	低电压侦测电路(LVD).....	203
21.6	上电复位电路(POR).....	203
21.7	I/O PAD 电路.....	203
21.8	Comparator0~3 比较器电路.....	204
21.9	Comparator4~5 比较器电路.....	204
21.10	Comparator4~5 比较器参考电压设置.....	205
21.11	运算放大器 0.....	205
21.12	10bit ADC 电路.....	205
21.13	稳压器输出电路.....	206
21.14	6bit D2I 电路.....	207
21.15	总体工作电流( $I_{VDD}$ ).....	207
21.16	AC 电气参数.....	208
21.17	时序图.....	208
21.18	直流和交流特性图表.....	209
22	指令集列表.....	213

23	芯片封装信息.....	215
附录 1,	文档更改历史 .....	222
附录 2,	从其他器件移植 .....	224
附录 3,	应用电路 .....	226

FMD Confidential

## 芯片版本历史

版本	描述
A	初版
B	修正以下问题： 当 PA1 配置为 MCLRB 管脚时，PA1 没有复位功能，见 4.2“外部复位 MCLR” ADC 内置参考电压稳定时间由 15us 变为 100us
C	D2I 的输出电压由 2V 改为 1.5V，片外电阻由 100K 改为 75K

FMD Confidential

# 1 系统功能框图和程序存储器、脚位图

## 1.1 系统功能框图

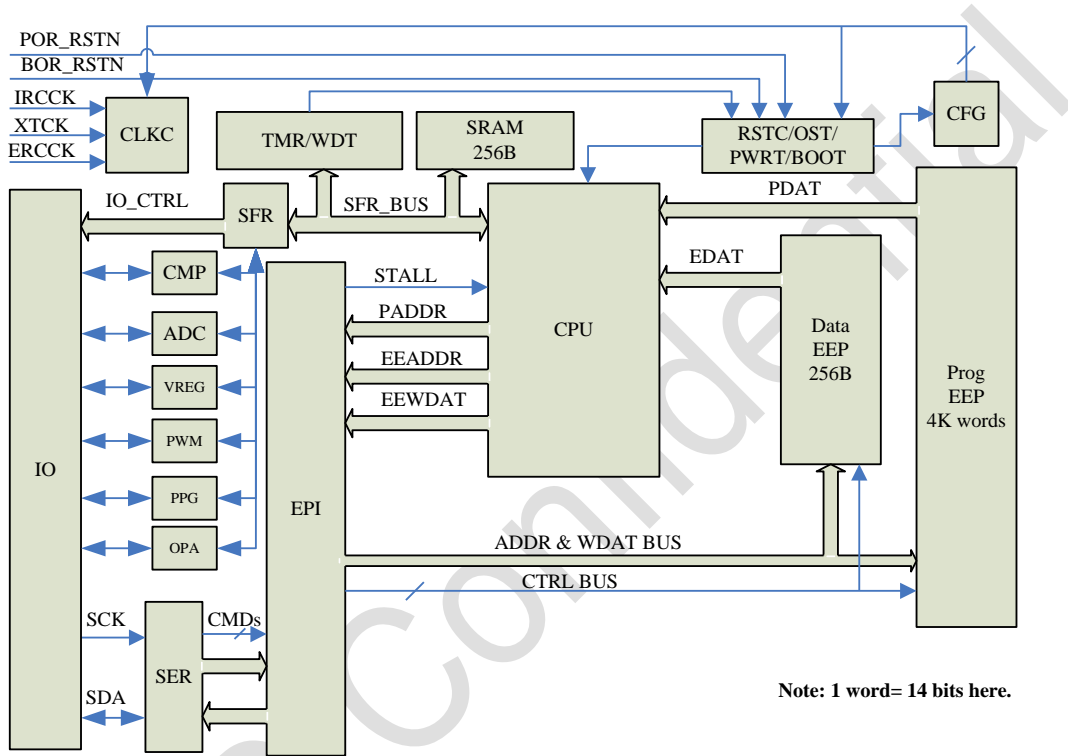


图 1.1 系统功能框图

## 1.2 程序存储器

程序地址计数器 PC 为 13 位(0x0000 ~ 0x1FFF)，最多支持 8K 地址空间。程序存储器一共有 4K 个字 (0x0000 ~ 0x0FFF)，加上 6 个额外的用户配置、工厂配置区，总共有 4K+192 个字，它们由 EEPROM 构成。其中一个 PAGE 是 32 个字，整个 PROM 一共有 134 个 PAGE。

注意这里所提到的字，其字宽为 14bit。

其中 0~0x0FFF 对主程序区访问，未实现部分 0x1000~0x1FFF 保留。用户和工厂配置信息区从 0x2000 开始，到 0x20BF 结束。

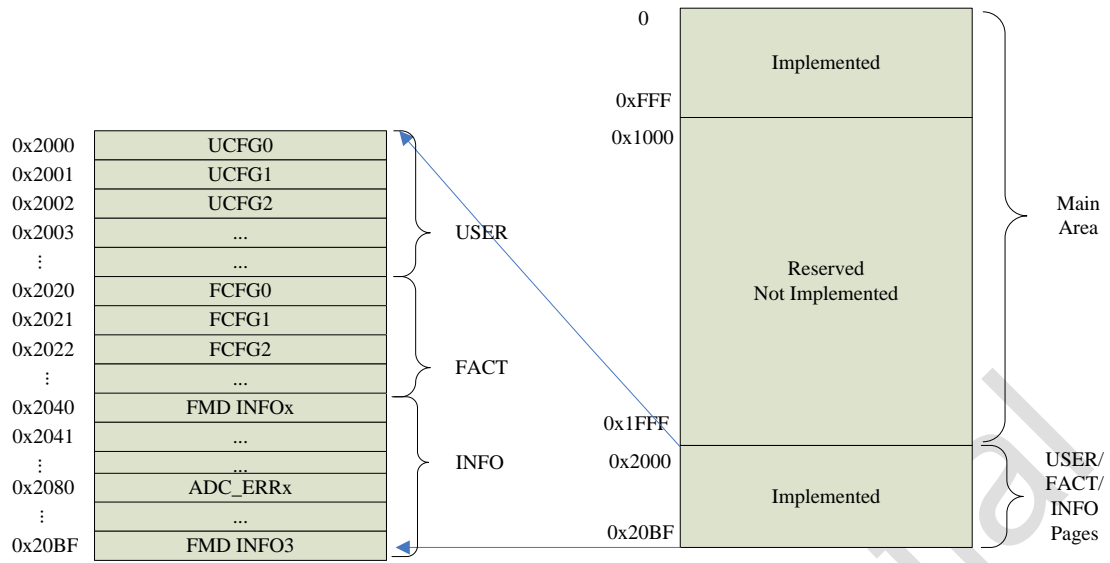


图 1.2 程序空间地址映射

### 1.3 脚位图

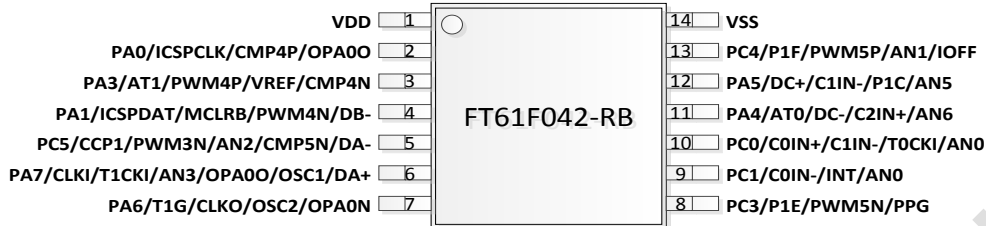


图 1.3 SOP14 脚位图

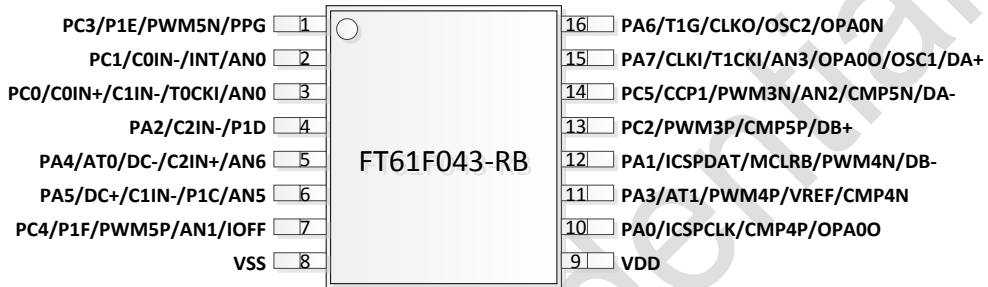


图 1.4 SOP16 脚位图

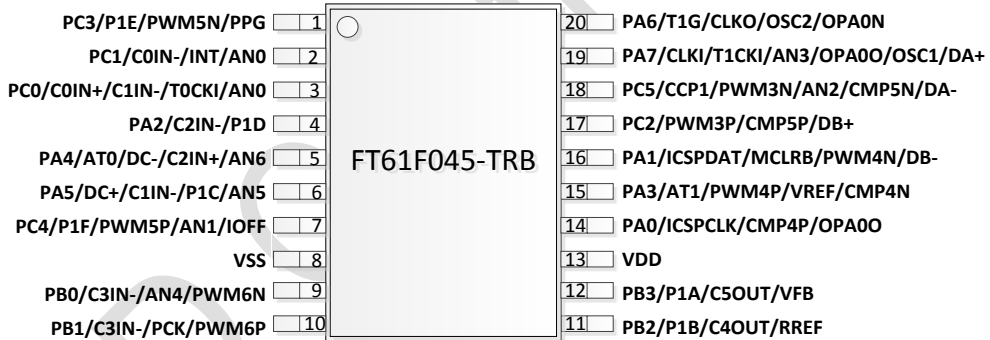


图 1.5 TSSOP20 脚位图

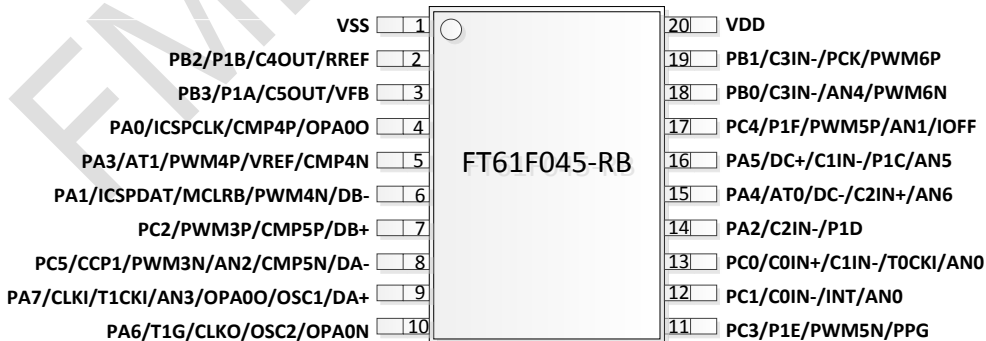


图 1.6 SOP20 脚位图

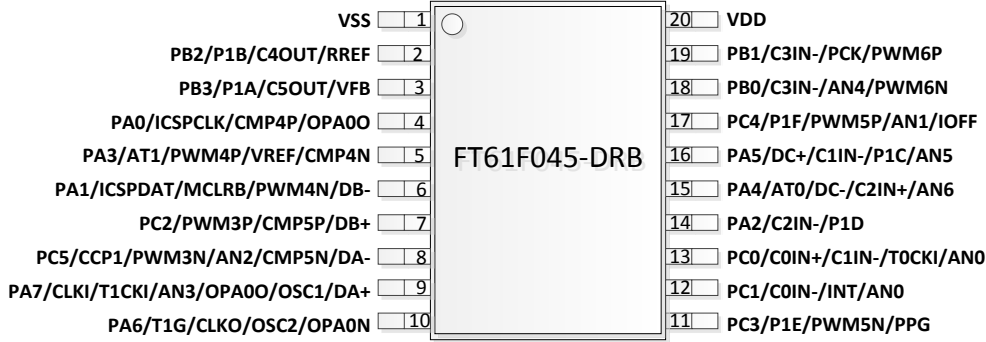


图 1.7 PDIP20 脚位图

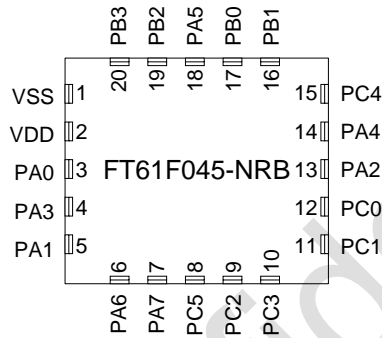


图 1.8 QFN20 脚位图

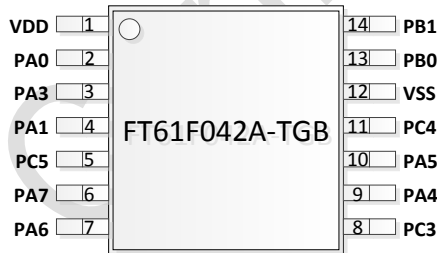


图 1.9 TSSOP14 脚位图

以下为芯片管脚的详细描述:

管脚名	功能名	输入信号类型	输出信号类型	具体描述	上下拉
PC3/P1E/PWM5N/PPG	PC3	ST	CMOS	GPIO wilth WPU	可配置上拉/下拉
	P1E	—	CMOS	增强型 PWM 输出	
	PWM5N	—	CMOS	PWM5 反相输出	
	PPG	—	CMOS	PPG 脉冲输出	
PC1/C0IN-/INT/AN0	PC1	ST	CMOS	GPIO with WPU	可配置上拉/下拉
	C0IN-	AN	—	比较器 0 反相输	



				入	
	INT	ST	—	外部中断输入	
	AN0	—	AN	A/D 通道 0 输入	
PC0/C0IN+/C1IN-/T0CKI/AN0	PC0	ST	CMOS	GPIO with WPU	可配置上拉
	C0IN+	AN	—	比较器 0 同相输入	
	C1IN-	AN	—	比较器 1 反相输入	C1NSEL
	T0CKI	ST	—	Timer 0 外部时钟输入 (Fmax=4MHz)	
	AN0	—	AN	A/D 通道 0 输入	
PA2/C2IN-/P1D	PA2	ST	CMOS	GPIO with IOF and WPU	可配置上拉
	C2IN-	AN	—	比较器 2 反相输入	
	P1D	—	CMOS	增强性 PWM 输出	
PA4/AT0/DC-/C2IN+/AN6	PA4	ST	CMOS	GPIO with IOF and WPU	可配置上拉/下拉
	AT0	AN	AN	模拟测试管脚	内部测试用
	DC-	—	AN	USB DC-输出	
	C2IN+	AN	—	比较器 2 同相输入	
	AN6	AN	—	A/D 通道 6 输入	
PA5/DC+/C1IN-/P1C/AN5	PA5	ST	CMOS	GPIO with IOF and WPU	可配置上拉
	DC+	—	AN	USB DC+输出	
	C1IN-	AN	—	比较器 1 反相输入	C1NSEL
	P1C	—	CMOS	增强性 PWM 输出	
	AN5	AN	—	A/D 通道 5 输入	
PC4/P1F/PWM5P/AN1/IOFF	PC4	ST	CMOS	GPIO with WPU	
	P1F	—	CMOS	增强性 PWM 输出	
	PWM5P	—	CMOS	PWM5 同相输出	
	AN1	AN	—	A/D 通道 1 输入	
	IOFF	—	CMOS	PPG 有效指示信号输出	

VSS	VSS	电源	—	接地参考端	
PB0/C3IN-/AN4/PWM6N	PB0	ST	CMOS	GPIO with WPU	可配置上拉
	C3IN-	AN	—	比较器 3 反相输入	
	AN4	AN	—	A/D 通道 4 输入	
	PWM6N	—	CMOS	PWM6 反相输出	
PB1/C3IN-/PCK/PWM6P	PB1	ST	CMOS	GPIO with WPU	可配置上拉
	C3IN-	AN	—	比较器 3 反相输入	
	PCK	—	CMOS	外设时钟输出	
	PWM6P	—	CMOS	PWM6 同相输出	
PB2/P1B/C4OUT/R <sub>REF</sub>	PB2	ST	CMOS	GPIO with WPU	可配置上拉
	P1B	—	CMOS	增强性 PWM 输出	
	C4OUT	—	CMOS	比较器 4 输出	
	R <sub>REF</sub>	—	AN	D2I 的参考电压输出, 外接电阻	
PB3/P1A/C5OUT/V <sub>FB</sub>	PB3	ST	CMOS	GPIO with WPU	可配置上拉
	P1A	—	CMOS	增强性 PWM 输出	
	C5OUT	—	CMOS	比较器 5 输出	
	V <sub>FB</sub>	AN	AN	D2I 的源电流/沉电流	
VDD	VDD	电源	—	正电源端	
PA0/ICSPCLK/CMP4P/OPA00	PA0	ST	CMOS	GPIO with IOF and WPU	可配置上拉
	ICSPCLK	ST	—	Debug/烧录模式串口 clock 信号 (F <sub>max</sub> =6MHz)	
	CMP4P	AN	—	比较器 4 同相输入	
	OPA00	—	AN	运放 0 输出	OP0OSEL
PA3/AT1/PWM4P/VREF/CMP4N	PA3	ST	CMOS	GPIO with IOF and WPU	可配置上拉
	AT1	AN	AN	模拟测试管脚	内部测试用

	PWM4P	—	CMOS	PWM4 同相输出	
	VREF	AN	—	A/D 外部参考电压输入	
	CMP4N	AN	—	比较器 4 反相输入	
PA1/ICSPDAT/MCLR/DB-	PA1	ST	CMOS	GPIO with IOF and WPU	可配置上拉
	ICSPDAT	ST	CMOS	Debug/烧录模式串口 data 信号 (Fmax=6MHz)	
	MCLR	ST	—	外部复位输入	
	PWM4N	—	CMOS	PWM4 反相输出	
	DB-	—	AN	USB DB-输出	
PC2/PWM3P/CMP5P/DB+	PC2	ST	CMOS	GPIO with WPU	可配置上拉/下拉
	PWM3P	—	CMOS	PWM3 同相输出	
	CMP5P	AN	—	比较器 5 同相输入	
	DB+	—	AN	USB DB+输出	
PC5/CCP1/PWM3N/AN2/CMP5N/DA-	PC5	ST	CMOS	GPIO with WPU	
	CCP1	ST	CMOS	捕捉输入/比较输出	
	PWM3N	—	CMOS	PWM3 反相输出	
	AN2	AN	—	A/D 通道 2 输入	
	CMP5N	AN	—	比较器 5 反相输入	
	DA-	—	AN	USB DA-输出	可配置下拉
PA7/CLKI/T1CKI/AN3/OPA00/OSC1/DA+	PA7	ST	CMOS	GPIO with IOF and WPU	可配置上拉
	CLKI	ST	—	External clock input/RC oscillator connection	
	T1CKI	ST	—	Timer1 外部时钟	
	AN3	AN	—	A/D 通道 3 输入	
	OPA00	—	AN	运放 0 输出	OP00SEL
	OSC1	XTAL	—	Crystal/Resonator	

	DA+	—	AN	USB DA+输出	可配置下拉
PA6/T1G/CLKO/OSC2/OPA0N	PA6	ST	CMOS	GPIO with IOF and WPU	可配置上拉
	T1G	ST	—	Timer1 门控输入	
	CLKO	—	CMOS	测试时钟输出 (Fmax=10MHz)	CLKO
	OSC2	—	XTAL	Crystal/Resonator	
	OPA0N	AN	—	运放 0 反相输入	

注意:

1. **IOF**: Interrupt on falling edge, 通用 IO
2. **WPU**: Weak pullup
3. **ST**: 带 CMOS 电平的施密特触发器输入
4. **AN**: 模拟输入或输出

FMD Confidential

## 2 数据存储区

### 2.1 数据存储区构成

数据存储区分为四个存储区（Bank），这四个存储区中包含通用寄存器 GPR 和特殊功能寄存器 SFR。特殊功能寄存器位于每个存储区的前 32 个单元中。以静态 RAM 形式实现的通用寄存器位于每个存储区的后 96 个单元中。每个存储区的最后 16 个字节为共用 RAM（如 0x70~7F, 0xF0~FF, 0x170~17F 和 0x1F0~0x1FF），这样就保证了无论当前处于哪一个区，程序总是可以访问到它们。

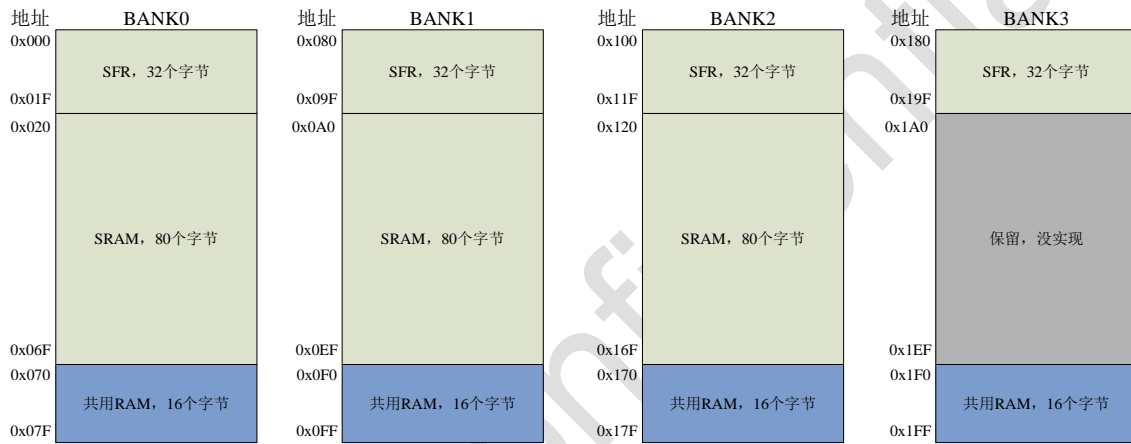


图 2.1.1 数据存储区构成示意图

### 2.2 寄存器地址映射

#### 2.2.1 SFR, BANK0

ADDR	Name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	POR reset
BANK0										
0	INDF	使用 FSR 的内容对数据存储区进行访问（非物理寄存器）								xxxx xxxx
1	TMR0	TIMER0 计数器								xxxx xxxx
2	PCL	程序计数器低 8 位								0000 0000
3	STATUS	FSR_B8	PAGE[1:0]		/TF	/PF	Z	HC	C	0001 1xxx
4	FSR	间接寻址指针寄存器								xxxx xxxx
5	PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	xxxx xxxx
6	PORTB	-	-	-	-	PB3	PB2	PB1	PB0	---- xxxx
7	PORTC	-	-	PC5	PC4	PC3	PC2	PC1	PC0	--xx xxxx
8	-									---- ----
9										

A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000	
B	INTCON	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000	
C	PIR1	EEIF	CKMIEIF	-	C5IF	C4IF	OSFIF	TMR2IF	TMR1IF	00-0 0000	
D	PIR2	-	-	C3IF	C2IF	C1IF	C0IF	ADIF	CCP1IF	--00 0000	
E	TMR1L	16 位 TIMER1 低字节保持寄存器								xxxx xxxx	
F	TMR1H	16 位 TIMER1 高字节保持寄存器								xxxx xxxx	
10	T1CON	T1GINV	TMR1GE	T1CKPS 1	T1CKPS 0	T1OSCE N	T1SYNC	TMR1CS	TMR1ON	0000 0000	
11	TMR2	TIMER2[7:0] timer2 module register								0000 0000	
12	T2CON	NRPM	TOUTPS[3:0]				TMR2ON	T2CKPS[1:0]		0000 0000	
13	CCPR1L	捕捉、比较、PWM 寄存器 1 的低字节								xxxx xxxx	
14	CCPR1H	捕捉、比较、PWM 寄存器 1 的高字节								xxxx xxxx	
15	CCP1CON	P1M[1:0]		DC1B[1:0]		CCP1M[3:0]				0000 0000	
16	PWM1CON	PRSEN	PDC[6:0]								0000 0000
17	ECCPAS	ECCPASE	ECCPAS[2:0]			PSSAC[1:0]		PSSBD[1:0]		0000 0000	
18	WDTCON	-	-	-	WDTPS[3:0]				SWDTEN	---0 1000	
19	CMCON0	C5OUT	C4OUT	C5INV	C4INV	CIS	CM[2:0]			xx00 0000	
1A	CMCON1	HYSEN[3:0]				-	-	T1GSS	C5SYNC	0000 --10	
1B	MSCKCON			T2CKSRC	SLVREN		CKMAVG	CKCNTI		0000 000-	
1C	SOSCPRL	SOSCPRL [7:0]								1111 1111	
1D	SOSCPRH	-	-	-	-	SOSCPRH [11:8]				---- 1111	
1E	ADRESH	左对齐格式下 AD 结果的高 8 位或者右对齐格式下的高 2 位								xxxx xxxx	
1F	ADCON0	ADFM	VCFG1	VCFG0	CHS[2:0]			GO/DONE	ADON	0000 0000	
20-6F	SRAM BANK0, (96Bytes) Physical address 0x00-0x4F									xxxx xxxx	
70-7F	SRAM BANK0, (16Bytes) Physical address 0XF0-0xFF									xxxx xxxx	

## 2.2.2 SFR, BANK1

ADDR	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR reset	
BANK1											
80	INDF	使用 SFR 的内容对数据存储器进行访问（非物理寄存器）									xxxx xxxx
81	OPTION	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	
82	PCL	程序计数器低 8 位								0000 0000	
83	STATUS	FSR_B8	PAGE[1:0]		/TF	/PF	Z	HC	C	0001 1xxx	
84	FSR	间接寻址指针寄存器								xxxx xxxx	
85	TRISA	TRISA[7:6]		TRISA[5]	TRISA[4:0]					1111 1111	
86	TRISB	-	-	-	-	TRISB[3:0]				---- 1111	
87	TRISC	-	-	TRISC[5:0]						--11 1111	
88	WPUC	-	-	WPUC[5:0]						--00 0000	
89	WPD	-	-	-	WPDA4	WPDC1	WPDC2	WPDC3	-	---0 000-	

8A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000
8B	INTCON	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
8C	PIE1	EEIE	CKMEAI E	-	C5IE	C4IE	OSFIE	TMR2IE	TMR1IE	00-0 0000
8D	PIE2	-	-	C3IE	C2IE	C1IE	C0IE	ADIE	CCP1IE	--00 0000
8E	PCON	SYSON	LVDL[2:0]			LVDEN	LVDW	/POR	/BOR	0000 00qq
8F	OSCCON	LFMOD	IRCF[2:0]			OSTS	HTS	LTS	SCS	0100 x000
90	PWM1AUX	AUX1EN	P1OS	P1FOE	P1EOE	P1DOE	P1COE	P1BOE	P1AOE	0--0 0000
91	ANSEL	ANSEL[7:0]								1111 1111
92	PR2	PR2[7:0] timer2 period register								1111 1111
93	DACOE	-	DAC2OE[2:0]			-	DAC1OE[2:0]			-000 -000
94	DACCON0	-	-	-	DBSH	-	DANPD	DAPPD	DASH	---0 -000
95	WPUA	WPUA[7:6]		WPUA[5]	WPUA[4:0]					1111 1111
96	IOFA	IOFA[7:0]								0000 0000
97	D2ICON	D2IEN	D2ISRC	D2IVAL[5:0]						0000 0000
98	-	-								0000 0000
99	VRCON	VREN	-	VRR	-	VR[3:0]				0-0- 0000
9A	EEDAT	EEDAT[7:0]								0000 0000
9B	EEADR	EEADR[7:0]								0000 0000
9C	EECON1	-	-	WREN3	WREN2	WRERR	WREN1	-	RD	--00 x0-0
9D	EECON2	-	-	-	-	-	-	-	WR	---- ---0
9E	ADRESL	左对齐格式下 AD 结果的低 2 位或者右对齐下结果的低 8 位								xxxx xxxx
9F	ADCON1	DIVS	ADCS[2:0]			-	-	-	AN0SEL	0000 ---0
A0-EF	SRAM BANK1 (80Bytes), Physical address 0x50~0x9F									xxxx xxxx
F0-FF	SRAM, 访问 BANK0 的 0x70~0x7F									xxxx xxxx

## 2.2.3 SFR, BANK2

ADDR	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR reset
BANK2										
100	INDF	使用 SFR 的内容对数据存储器进行访问（非物理寄存器）								xxxx xxxx
101	CMCON2	C1POL	C0POL	VR2C2:0]			VR1C[2:0]			0000 0000
102	PCL	程序计数器低 8 位								0000 0000
103	STATUS	FSR_B8	PAGE[1:0]		/TF	/PF	Z	HC	C	0001 1xxx
104	FSR	间接寻址指针寄存器								xxxx xxxx
105	EPWMCR1	-	-	-	-	PWMFCKS[3:0]				---- 0000
106	WPUB	-	-	-	-	WPUB[3:0]				---- 0000
107	CMCON3	C3POL	C2POL	VR4C[2:0]			VR3C[2:0]			0000 0000
108	VCON1	-	VREGM[1:0]		VREGHB[4:0]					-000 0000
109	VCON2	-	DAC1ON	DAC2ON	VREGLB[4:0]					-000 0000

10A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000
10B	INTCON	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
10C	TMR3L	12 位定时器 3 低 8 位								xxxx xxxx
10D	TMR3H	12 位定时器 3 高 4 位				PWM3 匹配寄存器 PR3 高 4 位				xxxx 1111
10E	PR3L	PWM3 匹配寄存器 PR3 低 8 位								1111 1111
10F	PWM3CR0	P3INTS	P3PER[2:0]			P3CKSRC[2:0]		P3BZR		0000 0000
110	PWM3CR1	P3EN	P3POL	TMR3PS[2:0]			TMR3ON	TMR3IE	TMR3IF	0000 0000
111	T3CKDIV	定时器 3 时钟分频寄存器								0000 0000
112	TMR4L	12 位定时器 4 低 8 位								xxxx xxxx
113	TMR4H	12 位定时器 4 高 4 位				PWM4 匹配寄存器 PR4 高 4 位				xxxx 1111
114	PR4L	PWM4 匹配寄存器 PR4 低 8 位								1111 1111
115	PWM4CR0	P4INTS	P4PER[2:0]			P4CKSRC[2:0]		P4BZR		0000 0000
116	PWM4CR1	P4EN	P4POL	TMR4PS[2:0]			TMR4ON	TMR4IE	TMR4IF	0000 0000
117	T4CKDIV	定时器 4 时钟分频寄存器								0000 0000
118	TMR5L	12 位定时器 5 低 8 位								xxxx xxxx
119	TMR5H	12 位定时器 5 高 4 位				PWM5 匹配寄存器 PR5 高 4 位				xxxx 1111
11A	PR5L	PWM5 匹配寄存器 PR5 低 8 位								1111 1111
11B	PWM5CR0	P5INTS	P5PER[2:0]			P5CKSRC[2:0]		P5BZR		0000 0000
11C	PWM5CR1	P5EN	P5POL	TMR5PS[2:0]			TMR5ON	TMR5IE	TMR5IF	0000 0000
11D	T5CKDIV	定时器 5 时钟分频寄存器								0000 0000
11E	EPWMCR0	PCKOE	-	-	-	P6IOE	P5IOE	P4IOE	P3IOE	0--- 0000
11F	CMCON4	C3DEB	C2DEB	C1DEB	-	CMPON[3:0]				111- 0000
120~16F	SRAM Bank2, (80 Bytes) Physical address 0xA0~0xEF									---- ----
170~17F	SRAM, 访问 BANK0 的 0x70~0x7F									xxxx xxxx

## 2.2.4 SFR, BANK3

ADDR	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR reset	
BANK3											
180	INDF	使用 SFR 的内容对数据存储器进行访问（非物理寄存器）								xxxx xxxx	
181	DEBCR	OP0RS	TRIGSR	DEBDLY[5:0]							0000 0000
182	PCL	程序计数器低 8 位								0000 0000	
183	STATUS	FSR_B8	PAGE[1:0]		/TF	/PF	Z	HC	C	0001 1xxx	
184	FSR	间接寻址指针寄存器								xxxx xxxx	
185	-									---- ----	
186	-									---- ----	
187	-									---- ----	
188	PPGCON1	PTRIGM	-	PWLTEN	PRLSEL	PRSTEN	PSTPEN	-	PPGST	0-00 00-0	
189	PPRH	-	-	-	PPRB[8]	-	-	-	PPRA[8]	---x ---x	
18A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000	



18B	INTCON	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
18C	PPRA	PPRA[7:0], PPG 重载寄存器 A 低 8 位								xxxx xxxx
18D	PPRB	PPRB[7:0], PPG 重载寄存器 B 低 8 位								xxxx xxxx
18E	PWLT PRL	PPG 脉宽限制比较寄存器低 8 位								1111 1111
18F	PPGCON2	PPGON	RELDB	PPGDLY[5:0]						0000 0000
190	CMP0CR	C0OUT	C0TM	C0CSEL	C0COF[4:0]					x001 0000
191	CMP1CR	C1OUT	C1TM	C1CSEL	C1COF[4:0]					x001 0000
192	CMP2CR	C2OUT	C2TM	C2CSEL	C2COF[4:0]					x001 0000
193	CMP3CR	C3OUT	C3TM	C3CSEL	C3COF[4:0]					x001 0000
194	OP0CR0	OP0OUT	OP0TM	-	OP0COF[4:0]					x00- 0000
195	-	-								---- ----
196	-	-								---- ----
197	-	-								---- ----
198	TMR6L	12 位定时器 6 低 8 位								xxxx xxxx
199	TMR6H	12 位定时器 6 高 4 位				PWM6 匹配寄存器 PR6 高 4 位				xxxx 1111
19A	PR6L	PWM6 匹配寄存器 PR6 低 8 位								1111 1111
19B	PWM6CR0	P6INTS	P6PER[2:0]			P6CKSRC[2:0]			P6BZR	0000 0000
19C	PWM6CR1	P6EN	P6POL	TMR6PS[2:0]			TMR6ON	TMR6IE	TMR6IF	0000 0000
19D	T6CKDIV	定时器 6 时钟分频寄存器								0000 0000
19E	PPGCON3	-	-	-	-	PSYNC	PPOL	PPGOEB	IOEB	---- 0011
19F	-	-								---- ----
1A0~16 F	SRAM Bank3, not implemented									---- ----
1F0~1F F	SRAM, 访问 BANK0 的 0x70~0x7F									xxxx xxxx

**注意:**

1. **INDF** 不是物理寄存器;
2. 灰色部分表示没有实现;
3. 未实现的寄存器位不要写 1, 以后芯片升级可能会用到;
4. **ANSEL** 复位值为 0xFF, 即相关 IO 为模拟管脚, 故任何复位之后去读端口寄存器 **PORTx**, 跟 **ADC** 通道复用的 IO 返回值为 0, 因为此时它们为模拟管脚, 这与它们的数据寄存器内容无关。
5. 绿色高亮部分寄存器 **WPUC**、**WPD**、**WPUA**、**WPUB**、**CMCON4** 以及 **PPGCON3** 的写有别于其它 **SFR**, 这 6 个寄存器跟 **UCFGx** 一样, 在上电配置中初始化。配置过程结束后受 **UCFG1.7** 位控制: 当 **UCFG1.7** 为 1 时, 用户程序可以对这些寄存器进行写操作, 任何复位都可使它们置为默认值 (**POR RESET VALUE**); 而当 **UCFG1.7** 为 0 时, 用户程序对它们只能读不能写, 只有上电复位、低电压复位、看门狗溢出复位以及错误指令复位能使它们置为默认值。

## 2.2.5 STATUS 寄存器，地址 0x03, 0x83, 0x103, 0x183

Bit	7	6	5	4	3	2	1	0
Name	FSR_B8	PAGE[1:0]		/TF	/PF	Z	HC	C
Reset	0	00		1	1	x	x	x
Type	RW	RW		RO	RO	RW	RW	RW

Bit	Name	Function
7	FSR_B8	FSR 寄存器第 8 位，与 FSR 组成一个 9 位的寄存器，在间接寻址时使用 详见 <a href="#">INDF 和 FSR 寄存器</a> 一节
6:5	PAGE[1:0]	<b>PAGE:</b> 寄存器存储区选择位（用于直接寻址） 00 = Bank0 (0x00~ 0x7F) 01 = Bank 1 (0x80~ 0xFF) 10= Bank2 (0x100~ 0x17F) 11 = Bank3 (0x180~0x1FF)
4	/TF	<b>/TF:</b> 超时状态位 1 = 上电后，执行了CLRWDT 指令或SLEEP 指令 0 = 发生 WDT 超时溢出
3	/PF	<b>/PF:</b> 掉电标志位 1 = 上电复位后或执行了CLRWDT 指令 0 = 执行了SLEEP 指令
2	Z	<b>Z:</b> 零标志位 1 = 算术运算或逻辑运算的结果为零 0 = 算术运算或逻辑运算的结果不为零
1	HC	<b>HC:</b> 半进位/ 借位位（ADDWF、ADDWI、SUBWI 和SUBWF 指令）。对于借位，极性是相反的。 1 = 结果的第4低位向高位发生了进位 0 = 结果的第4低位未向高位发生进位
0	C	<b>C:</b> 进位/ 借位位（ADDWF、ADDWI、SUBWI 和SUBWF 指令） 1 = 结果的最高位发生了进位 0 = 结果的最高位未发生进位

/TF	/PF	条件
1	1	上电或者低电压复位
0	U	WDT 复位
0	0	WDT 唤醒
U	U	正常运行下发生 MCLR 复位
1	0	睡眠状态下发生 MCLR 复位

### 注意:

1. 和其它寄存器一样，状态寄存器也可以作为任何指令的目标寄存器。如果一条指令影响 Z、HC 或 C 位的指令以状态寄存器作为目标寄存器，将禁止对这三位的写操作，它们只受逻辑结

果影响，被置 1 或清 0。因此，当执行一条把状态寄存器作为目标寄存器的指令后，STATUS 内容可能和预想的不一致；

2. 建议只使用 BCR、BSR、SWAPR 和 STR 指令来改变状态寄存器。

## 2.2.6 配置寄存器 UCFGx

软件不能访问 UCFG0、UCFG1、UCFG2、UCFG3 及 UCFG4，它们只在上电过程由硬件写入。

### 2.2.6.1 UCFG0, PROM 地址 0x2000

Bit	7	6	5	4	3	2	1	0
Name	-	CPB	MCLRE	PWRTEB	WDTE	FOSC[2:0]		

位	名称	描述
Bit[7]	NA	保留位，读 0
Bit[6]	CPB	1: Flash 内容不保护 0: 启动 Flash 内容保护，MCU 能读，串口不能读 注意： 此位只能由 1 改写为 0，而不能由 0 改写为 1。由 0 改写成 1 的唯一方法是进行一次包括 USER_OPT 在内的片擦操作，并且重新上电后 CPB 才变为 1
Bit[5]	MCLRE	1: PA1/MCLR 脚执行 MCLR 功能，是复位脚 0: PA1/MCLR 脚执行 GPIO 功能
Bit[4]	PWRTEB	1: PWRT 禁止 0: PWRT 使能
Bit[3]	WDTE	1: WDT 使能，程序不能禁止 0: WDT 禁止，但程序可通过设置 WDTCON 的 SWDTEN 位将 WDT 使能
Bit[2:0]	FOSC[2:0]	000: LP 模式，PA6/PA7 接低速晶体（32KHz） 001: XT 模式，PA6/PA7 接高速晶体（4~20MHz） 010: EC 模式，PA6 为 IO 功能，PA7 接时钟输入 011: INTOSC 模式，PA6 输出系统时钟的 2 分频，PA7 为 IO 引脚 1xx: INTOSCIO 模式，PA6 为 IO 引脚，PA7 为 IO 引脚

### 2.2.6.2 UCFG1, PROM 地址 0x2001

Bit	7	6	5	4	3	2	1	0
Name	USFRSEL	CSUMENB	TSEL	FCMEN	IESO	RD_CTRL	LVREN1	LVREN0

位	名称	描述
Bit[7]	USFRSEL	寄存器 WPUC/WPD/WPUA/WPUB/CMCON4/PPGCON3 的写控制

		1: 以上寄存器除了由上电配置之外, 也能由软件进行写操作 0: 以上寄存器只能由上电配置, 软件不能进行写操作, 读不受影响
Bit[6]	CSUMENB	Checksum enable 1: 禁止 checksum 功能 0: 使能 checksum 功能, boot 完成后, 硬件自动把 4K PROM 空间所有单元内容相加, 结果与 0x2007 单元内容比较, 如果相等说明校验成功; 校验失败时 CPU 将不执行程序
Bit[5]	TSEL	指令周期选择位 1: 指令周期为 2T 0: 指令周期为 4T
Bit[4]	FCMEN	时钟故障监视使能 1: 使能时钟故障监视 0: 禁止时钟故障监视
Bit[3]	IESO	双速时钟使能 1: 使能双速时钟模式 0: 禁止双速时钟模式
Bit[2]	RD_CTRL	输出模式时读端口控制 1: 读数据端口返回的 PAD 上的值 0: 读数据端口返回的 Latch 上的值
BIT[1:0]	LVREN[1:0]	低电压复位选择 00: 使能低电压复位 01: LVR 由 MSCKCON 的 SLVREN 决定 10: MCU 正常模式时开启 LVR, 睡眠模式时关闭 LVR, 跟 SLVREN 位无关 11: 禁止低电压复位

### 2.2.6.3 UCFG2, PROM 地址 0x2002

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	LVRS[3:0]			

位	名称	描述																
Bit[7:4]	-	保留位																
Bit[3:0]	LVRS[3:0]	低电压复位阈值选择 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>数值</th> <th>电压</th> </tr> </thead> <tbody> <tr> <td>0011</td> <td>2.0V</td> </tr> <tr> <td>0100</td> <td>2.2V</td> </tr> <tr> <td>0101</td> <td>2.5V</td> </tr> <tr> <td>0110</td> <td>2.8V</td> </tr> <tr> <td>0111</td> <td>3.1V</td> </tr> <tr> <td>1000</td> <td>3.6V</td> </tr> <tr> <td>1001</td> <td>4.1V</td> </tr> </tbody> </table>	数值	电压	0011	2.0V	0100	2.2V	0101	2.5V	0110	2.8V	0111	3.1V	1000	3.6V	1001	4.1V
数值	电压																	
0011	2.0V																	
0100	2.2V																	
0101	2.5V																	
0110	2.8V																	
0111	3.1V																	
1000	3.6V																	
1001	4.1V																	

		其它值	保留
--	--	-----	----

### 2.2.6.4 UCFG3, PROM 地址 0x2003

Bit	7	6	5	4	3	2	1	0
Name	C3NSEL		C2PSEL	C2NSEL	-	C1NSEL	-	-

位	名称	描述	
Bit[7:6]	C3NSEL	比较器 3 的反相输入选择	
		值	比较器 3 反相输入
		11	运放 0 输出
		10	PB0 管脚
		01, 00	PB1 管脚
Bit[5]	C2PSEL	比较器 2 的同相输入选择	
		值	比较器 2 同相输入
		1	PA4 管脚
		0	内部参考 VR2
Bit[4]	C2NSEL	比较器 2 的反相输入选择	
		值	比较器 2 反相输入
		1	PA2 管脚
		0	内部参考 VR3
Bit[3]	-		
Bit[2]	C1NSEL	比较器 1 的反相输入选择	
		值	比较器 1 反相输入
		1	PA5 管脚
		0	PC0 管脚
Bit[1:0]	-	保留位, 未实现	

### 2.2.6.5 UCFG4, PROM 地址 0x2004

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	OP0ON	OP0OSEL	OP0PSEL	OP0NFB	OP0NSEL

位	名称	描述
Bit[7:5]	-	保留位, 未实现
Bit[4]	OP0ON	运放 0 使能控制
		1 = 使能运放
		0 = 关闭运放
Bit[3]	OP0OSEL	运放输出选择位

		<p>当 OP0ON 为 1 时</p> <p>1 = 运放输出串 10K (OP0R4) 电阻后连到引脚 PA0</p> <p>0 = 运放输出直接连到引脚 PA7/AN3</p> <p>当 OP0ON 为 0 时, 此位无意义, PA7 和 PA0 皆为普通 IO</p>
Bit[2]	OP0PSEL	<p>运放的同相输入端选择</p> <p>1 = 同相端串接 1K (OP0R1) 电阻到地</p> <p>0 = 同相端直接连到地</p>
Bit[1]	OP0NFB	<p>运放负反馈选择</p> <p>1 = 运放输出串接 60K 电阻 (OP0R3) 连接到反相端</p> <p>0 = 运放没有负反馈</p>
Bit[0]	OP0NSEL	<p>运放的反相输入端选择</p> <p>1 = 引脚 PA6/OPA0N 串接 1K 电阻连到反相端</p> <p>0 = 引脚 PA6/OPA0N 直接连到反相端</p>

### 2.2.6.6 UCFG8(WPUC), PROM 地址 0x2008

参考 [WPUC 寄存器](#) 描述。

### 2.2.6.7 UCFG9(WPD), PROM 地址 0x2009

参考 [WPD 寄存器](#) 描述。

### 2.2.6.8 UCFG10(WPUA), PROM 地址 0x200A

参考 [WPUA 寄存器](#) 描述。

### 2.2.6.9 UCFG11(WPUB), PROM 地址 0x200B

参考 [WPUB 寄存器](#) 描述。

### 2.2.6.10 UCFG12(CMCON4), PROM 地址 0x200C

参考 CMCON4 寄存器描述。

### 2.2.6.11 UCFG13(PPGCON3), PROM 地址 0x200D

参考 [PPGCON3 寄存器](#) 描述。

## 2.2.7 PCL 和 PCLATH

程序计数器（PC）为 12 位宽。其低 8 位来自可读写的 PCL 寄存器，高 4 位（PC<11:8>）来自 PCLATH，不能直接读写。只要发生复位，PC 就将被清 0。下图显示了装载 PC 值的两种情形。注意图右边的 LCALL 和 LJUMP 指令，由于指令中的操作码为 11 位，而芯片的 PC 是 12 位，所以此时只用到 PCLATH 的第 3 位，和操作码的 11 位组成 12 位地址。

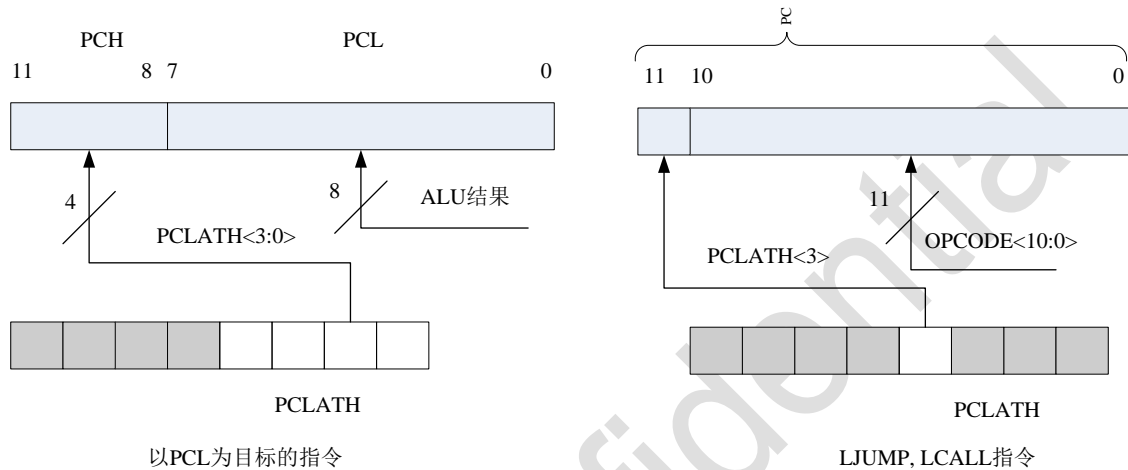


图 2.1.1 不同情况下 PC 的加载

### 修改 PCL

执行任何以 PCL 寄存器为目标寄存器的指令将同时使程序计数器 PC<11:8>位被 PCLATH 内容取代。这样可通过将所需的高 4 位写入 PCLATH 寄存器来改变程序计数器的所有内容。

计算 LJUMP 指令是通过向程序计数器加入偏移量（ADDWR PCL）来实现的。通过修改 PCL 寄存器跳转到查找表或程序分支表（计算 LJUMP）时应特别谨慎。假定 PCLATH 设置为表的起始地址，如果表长度大于 255 条指令，或如果存储器地址的低 8 位在表的中间从 0xFF 计满返回到 0x00，那么在每次表起始地址与表内的目标地址之间发生计满返回时，PCLATH 必须递增。

## 2.2.8 INDF 和 FSR 寄存器

INDF 不是物理存在的寄存器，对 INDF 进行寻址将产生间接寻址，可寻址范围为 0~511。

任何使用 INDF 寄存器的指令，实际上是对文件选择寄存器{FSR\_B8,FSR}所指向的单元进行存取。间接对 INDF 进行读操作将返回 0。间接对 INDF 进行写将导致空操作（可能会影响状态标志位）。

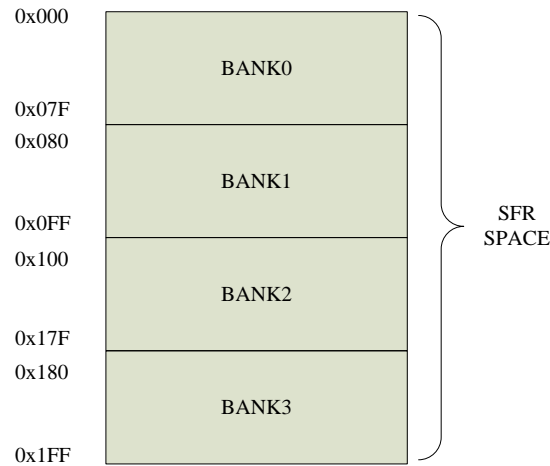


图 2.1.2 间接寻址

## 2.2.9 关于寄存器保留位

如 2.2.1/2.2.2/2.2.3/2.2.4 章节的表格所示，SFR 空间有部分寄存器或者寄存器位没有实现，未实现的寄存器位是**保留位**，软件读返回的是 0，写无效。

不建议程序对这些保留位写 1，这有可能会给以后程序的移植带来问题，因为后续芯片产品可能会用到这些位。



### 3 系统时钟源

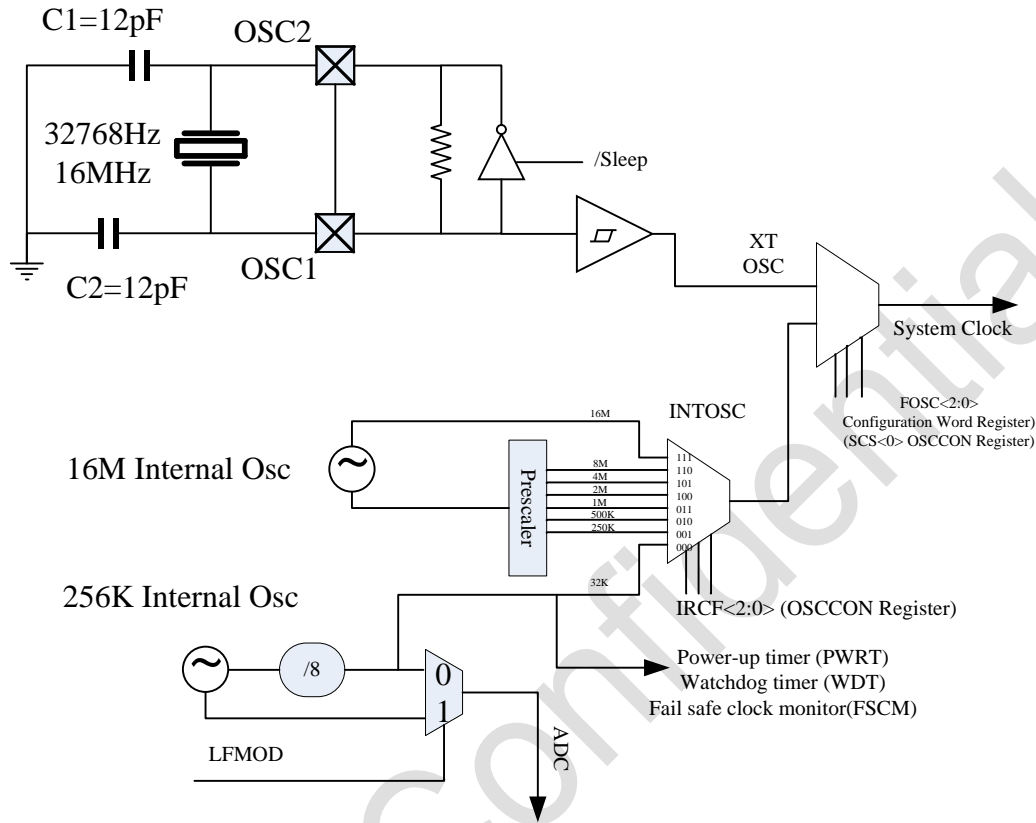


图 3.1 系统时钟源框图

本芯片包含 4 个时钟源：2 个内置振荡器，1 个外部晶体振荡器，1 个外部时钟灌入源。内置振荡器包括 1 个内部 16M 高速精准振荡器(HFINTOSC)，1 个内部 32K/256K(LFINTOSC)低速低功耗振荡器。这些时钟或振荡器结合预分频器可以给系统提供各种频率的时钟源。

系统时钟源的预分频器比例由 OPTION 寄存器里的 IRCF<2:0>位控制。

**注意：**

内部慢时钟的 256KHz 模式只供 ADC(当 ADCS 为 x11 时)使用。看门狗、系统时钟源(IRCFCF=000)以及 PWRT 统一使用 8 分频之后的输出，即 32KHz，而不管 LFMOD 为何值。

#### 3.1 时钟源模式

时钟源模式分为外部和内部模式。

- 外部时钟模式依靠外部电路提供时钟源，比如外部时钟 EC 模式，晶体谐振器 XT、LP 模式。
- 内部时钟模式内置于振荡器模块中，振荡器模块有 16MHz 高频振荡器和 32KHz 低频振荡器。

可通过 OSCCON 寄存器的系统时钟选择位（SCS）来选择内部或者外部时钟源。

## 3.2 外部时钟模式

### 3.2.1 振荡器起振定时器（OST）

如果振荡器模块配置为 LP、XT 模式，振荡器起振定时器（OST）将对来自 OSC1 的振荡计数 1024（在 LP 晶体模式下为 32768）次。这发生在上电复位（POR）之后以及上电延时定时器（PWRT）延时结束（如果被使能）时，或从休眠中唤醒后。在此期间，程序计数器不递增，程序执行暂停。OST 确保使用石英晶体谐振器或陶瓷谐振器的振荡器电路已经启动并向振荡器模块提供稳定的系统时钟信号。当在时钟源之间切换时，需要一定的延时以使新时钟稳定。

**注意：OST 复用了 WDT 定时器，故在 OST 对晶体时钟计数时，WDT 功能被屏蔽，待 OST 发生溢出后，WDT 功能才恢复（如果此前 WDT 被使能的话）。**

### 3.2.2 EC 模式

外部时钟模式允许外部产生的逻辑电平作为系统时钟源。工作在此模式下时，外部时钟源连接到 OSC1 输入，OSC2 引脚可用作通用 I/O。

当选取 EC 模式时，振荡器起振定时器（OST）被禁止。因此，上电复位（POR）后或者从休眠中唤醒后的操作不存在延时。MCU 被唤醒后再次启动外部时钟，器件恢复工作，就好像没有停止过一样。

### 3.2.3 LP 和 XT 模式

LP 和 XT 模式支持连接到 OSC1 和 OSC2 的石英晶体谐振器或陶瓷谐振器的使用。

模式选择内部反相放大器的低或高增益设定，以支持各种谐振器类型及速度。

LP 振荡器模式选择内部反相放大器的最低增益设定。

LP 模式的电流消耗在两种模式中最小。该模式设计仅用于驱动 32.768 kHz 音叉式晶振（钟表晶振）。

XT 振荡器模式选择内部反相放大器的高增益设定。

## 3.3 内部时钟模式

振荡器模块有两个独立的内部振荡器，可配置或选取为系统时钟源。

1. HFINTOSC（高频内部振荡器）出厂时已校准，工作频率为 16MHz。
2. LFINTOSC（低频内部振荡器）未经校准，工作频率为 32 kHz。软件对 OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0>进行操作，可选择系统时钟速度。

可通过 OSCCON 寄存器的系统时钟选择（SCS）位，在外部或内部时钟源之间选择系统时钟。

**注意：OSCCON 寄存器的 LFMOD 可以选择 LFINTOSC 是 32KHz 或者 256KHz，但看门狗固定**

使用 32KHz，不管 LFMOD 为何值。

### 3.3.1 频率选择位 (IRCF)

16MHz HFINTOSC 和 32KHz LFINTOSC 的输出连接到预分频器和多路复用器（见图 3.1）。OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0>用于选择内部振荡器的频率输出。可通过软件选择以下 8 个频率之一：

- 16MHz
- 8MHz
- 4MHz
- 2MHz（复位后的缺省值）
- 1MHz
- 500 kHz
- 250 kHz
- 32 kHz

### 3.3.2 HFINTOSC 和 LFINTOSC 时钟切换时序

当在 LFINTOSC 和 HFINTOSC 之间切换时，新的振荡器可能为了省电已经关闭（见图 3.2 和图 3.3）。在这种情况下，OSCCON 寄存器的 IRCF 位被修改之后、频率选择生效之前，存在一个延时。OSCCON 寄存器的 LTS 和 HTS 位将反映 LFINTOSC 和 HFINTOSC 振荡器的当前活动状态。频率选择时序如下：

1. OSCCON 寄存器的 IRCF<2:0>位被修改。
2. 如果新时钟是关闭的，开始一个时钟启动延时。
3. 时钟切换电路等待当前时钟下降沿的到来。
4. CLKOUT 保持为低，时钟切换电路等待两个新时钟下降沿的到来。
5. 现在 CLKOUT 连接到新时钟。OSCCON 寄存器的 HTS 和 LTS 位按要求被更新。
6. 时钟切换完成。

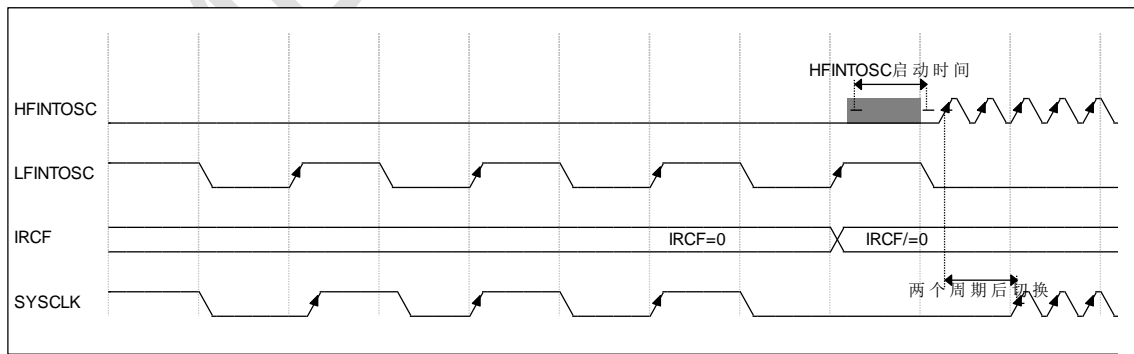


图 3.2 由慢时钟切换到快时钟

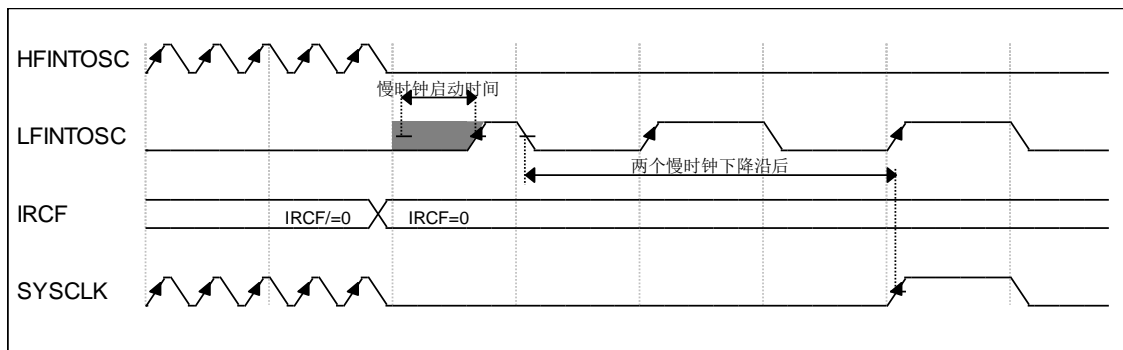


图 3.3 由快时钟切换到慢时钟

### 3.4 时钟切换

通过软件对 OSCCON 寄存器的系统时钟选择 (SCS) 位进行操作, 可将系统时钟源在外部和内部时钟源之间切换。

#### 3.4.1 系统时钟选择 (SCS) 位

OSCCON 寄存器的系统时钟选择 (SCS) 位选择用于 CPU 和外设的系统时钟源。

- OSCCON 寄存器的位 SCS = 0 时, 系统时钟源由配置字寄存器 (UCFG0) 中 FOSC<2:0>位的配置决定。
- OSCCON 寄存器的位 SCS = 1 时, 根据 OSCCON 寄存器的 IRCF<2:0>位所选的内部振荡器频率选取系统时钟源。复位后, OSCCON 寄存器的 SCS 总是被清零。

注: 任何由硬件引起的时钟切换 (可能产生自双速启动或故障保护时钟监控器) 都不会更新 OSCCON 寄存器的 SCS 位。用户应该监控 OSCCON 寄存器的 OSTS 位以确定当前的系统时钟源。

#### 3.4.2 振荡器起振超时状态 (OSTS) 位

OSCCON 寄存器的振荡器起振超时状态 (OSTS) 位用于指示系统时钟是来自外部时钟源, 还是来自内部时钟源。外部时钟源由配置字寄存器 (UCFG0) 的 FOSC<2:0>定义。OSTS 还特别指明在 LP 或 XT 模式下, 振荡器起振定时器 (OST) 是否已超时。

### 3.5 双速时钟启动模式

双速启动模式通过最大限度地缩短外部振荡器起振与代码执行之间的延时, 进一步节省了功耗。对于频繁使用休眠模式的应用, 双速启动模式将在器件唤醒后除去外部振荡器的起振时间, 从而可降低器件的总体功耗。该模式使得应用能够从休眠中唤醒, 将 INTOSC 用作时钟源执行数条指令, 然后再返回休眠状态而无需等待主振荡器的稳定。

注：执行 SLEEP 指令将中止振荡器起振时间，并使 OSCCON 寄存器的 OSTS 位保持清零。

当振荡器模块配置为 LP 或 XT 模式时，振荡器起振定时器（OST）使能（见第 3.2.1 节“振荡器起振定时器”）。OST 将暂停程序执行，直到完成 1024（在 LP 晶体模式下为 32768）次振荡计数。双速启动模式在 OST 计数时使用内部振荡器进行工作，使代码执行的延时最大限度地缩短。当 OST 计数到 1024（在 LP 晶体模式下为 32768）且 OSCCON 寄存器的 OSTS 位置 1 时，程序执行切换至外部振荡器。

### 3.5.1 双速启动模式配置

通过以下设定来配置双速启动模式：

- 配置字寄存器（UCFG1）中的位 IESO = 1；内部/外部切换位（使能双速启动模式）。
- OSCCON 寄存器的位 SCS = 0。
- 配置字寄存器（CONFIG）中的 FOSC<2:0>配置为 LP 或 XT 模式。

在下列操作之后，进入双速启动模式：

- 上电复位（POR）且上电延时定时器（PWRT）。
- 延时结束（使能时）后，或者从休眠状态唤醒。

如果外部时钟振荡器配置为除 LP 或 XT 模式以外的任一模式，那么双速启动将被禁止。这是因为 POR 后或从休眠中退出时，外部时钟振荡器不需要稳定时间。

### 3.5.2 双速启动顺序

1. 从上电复位或休眠中唤醒。
2. 使用内部振荡器以 OSCCON 寄存器的 IRCF<2:0>位设置的频率开始执行指令。
3. OST 使能，计数 1024（在 LP 晶体模式下为 32768）个时钟周期。
4. OST 超时，等待内部振荡器下降沿的到来。
5. OSTS 置 1。
6. 系统时钟保持为低，直到新时钟下一个下降沿的到来（LP 或 XT 模式）。
7. 系统时钟切换到外部时钟源。

## 3.6 故障保护时钟监控器

故障保护时钟监控器（FSCM）使得器件在出现外部振荡器故障时仍能继续工作。FSCM 能在振荡器起振延时定时器（OST）到期后的任一时刻检测振荡器故障。FSCM 通过将配置字寄存器（UCFG1）中的 FCMEN 位置 1 来使能。FSCM 可用于所有外部振荡模式（LP、XT 和 EC）。

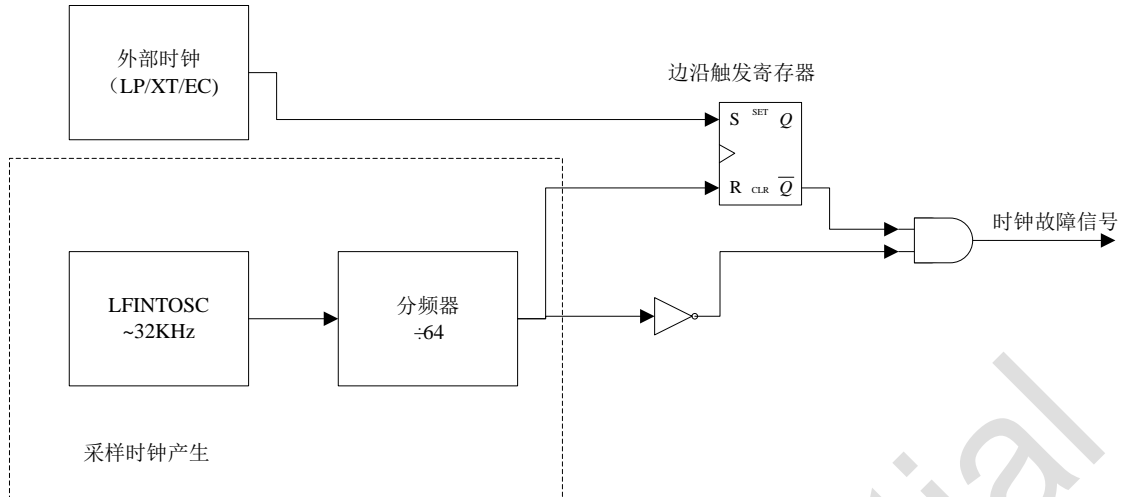


图 3.4 FSCM 原理框图

### 3.6.1 故障保护检测

FSCM 模块通过将外部振荡器与 FSCM 采样时钟比较来检测振荡器故障。LFINTOSC 除以 64，就产生了采样时钟。请参见图 3.4。故障检测器内部有一个锁存器。在外部时钟的每个下降沿，锁存器被置 1。在采样时钟的每个上升沿，锁存器被清零。如果采样时钟的整个半周期流逝而主时钟依然未进入低电平，就检测到故障。

### 3.6.2 故障保护操作

当外部时钟出现故障时，FSCM 将器件时钟切换到内部时钟源，并将 PIR1 寄存器的 OSFIF 标志位置 1。如果在 PIR1 寄存器的 OSFIE 位置 1 的同时将该标志位置 1，将产生中断。器件固件随后会采取措施减轻可能由故障时钟所产生的问题。系统时钟将继续来自内部时钟源，直到器件固件成功重启外部振荡器并切换回外部操作。

FSCM 所选的内部时钟源由 OSCCON 寄存器的 IRCF<2:0>位决定。这使内部振荡器可以在故障发生前就得以配置。

### 3.6.3 故障保护条件清除

复位、执行 SLEEP 指令或翻转 OSCCON 寄存器的 SCS 位后，故障保护条件被清除。OSCCON 寄存器的 SCS 位被修改后，OST 将重新启动。OST 运行时，器件继续从 OSCCON 中选定的 INTOSC 进行操作。OST 超时后，故障保护条件被清除，器件将从外部时钟源进行操作。必须先清除故障保护条件，才能清零 OSFIF 标志位。

### 3.6.4 复位或从休眠中唤醒

FSCM 设计为能在振荡器起振延时定时器 (OST) 到期后的任一时刻检测振荡器故障。OST 的使

用场合为从休眠状态唤醒后以及任何类型的复位后。OST 不能在 EC 时钟模式下使用，所以一旦复位或唤醒完成，FSCM 就处于激活状态。当 FSCM 被使能时，双速启动也被使能。因此，当 OST 运行时，器件总是处于代码执行阶段。

注：由于振荡器起振时间的范围变化较大，在振荡器起振期间（从复位或休眠中退出时），故障保护电路不处于激活状态。经过一段适当的时间后，用户应检查 OSCCON 寄存器的 OSTS 位，以验证振荡器是否已成功起振以及系统时钟是否切换成功。

### 3.7 与时钟源相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
UCFG0	0x2000	-	CPB	MCLRE	PWRTEB	WDTE	FOSC2	FOSC1	FOSC0	---- ----
OSCCON	0x8F	LFMOD	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS	0100 x000

#### 3.7.1 OSCCON 寄存器，地址 0x8F

Bit	7	6	5	4	3	2	1	0
Name	LFMOD	IRCF[2:0]			OSTS	HTS	LTS	SCS
Reset	0	3'b100			1	0	0	0
TYPE	RW	RW			RO	RO	RO	RW

Bit	Name	Function																											
7	LFMOD	低频内振模式： 1 = 256K 振荡频率模式 0 = 32K 振荡频率模式																											
6:4	IRCF[2:0]	内部振荡器（系统时钟）频率选择： <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>值</th> <th>2T 模式</th> <th>4T 模式</th> </tr> </thead> <tbody> <tr> <td>111</td> <td>16MHz</td> <td>8MHz</td> </tr> <tr> <td>110</td> <td>8MHz</td> <td>4MHz</td> </tr> <tr> <td>10</td> <td>4MHz</td> <td>2MHz</td> </tr> <tr> <td>100</td> <td>2MHz(default)</td> <td>1MHz(default)</td> </tr> <tr> <td>011</td> <td>1MHz</td> <td>500kHz</td> </tr> <tr> <td>010</td> <td>500kHz</td> <td>250kHz</td> </tr> <tr> <td>001</td> <td>250kHz</td> <td>125kHz</td> </tr> <tr> <td>000</td> <td>32kHz(LFINTOSC)</td> <td>16kHz(LFINTOSC/2)</td> </tr> </tbody> </table>	值	2T 模式	4T 模式	111	16MHz	8MHz	110	8MHz	4MHz	10	4MHz	2MHz	100	2MHz(default)	1MHz(default)	011	1MHz	500kHz	010	500kHz	250kHz	001	250kHz	125kHz	000	32kHz(LFINTOSC)	16kHz(LFINTOSC/2)
值	2T 模式	4T 模式																											
111	16MHz	8MHz																											
110	8MHz	4MHz																											
10	4MHz	2MHz																											
100	2MHz(default)	1MHz(default)																											
011	1MHz	500kHz																											
010	500kHz	250kHz																											
001	250kHz	125kHz																											
000	32kHz(LFINTOSC)	16kHz(LFINTOSC/2)																											
3	OSTS	振荡器起振超时状态位 1 = 器件运行在 FOSC<2:0>指定的外部时钟之下 0 = 器件运行在内部振荡器之下																											
2	HTS	高速内部时钟状态 1 = HFINTOSC is ready																											

		0 = HFINTOSC is not ready
1	LTS	低速内部时钟状态 1 = LFINTOSC is ready 0 = LFINTOSC is not ready
0	SCS	系统时钟选择位 1 = 系统时钟选择为内部振荡器 0 = 时钟源由 FOSC<2:0>决定

FMD Confidential



## 4 复位时序

FT61F04X 有以下几种不同的复位：

- A) 上电复位 POR
- B) WDT(看门狗)复位 – 在常规运行期间
- C) WDT(看门狗)唤醒 – 在睡眠期间
- D) /MCLR 管脚复位 – 在常规运行期间
- E) /MCLR 管脚复位 – 在睡眠期间
- F) 低电压 (BOR/LVR) 复位
- G) 错误指令复位

有些寄存器是不被任何复位影响的；这些寄存器的状态在上电复位时是未知的，也不受复位事件影响。大多数其它寄存器都会在以下复位事件时恢复到其“复位状态”：

- 上电复位 POR
- WDT(看门狗)复位 – 在常规运行期间
- WDT(看门狗)复位 – 在睡眠期间
- /MCLR 管脚复位 – 在常规运行期间
- 低电压 (BOR) 复位
- 错误指令复位

WDT(看门狗)睡眠唤醒不会造成和在常规运行状态下 WDT(看门狗)超时所造成的复位。因为睡眠唤醒本身就是一种继续的意思，而不是复位/TF 和/PF 位的设置和清零在不同复位条件下的动作是不同的。具体可参考表 4.1 和 4.2。

/MCLR 管脚背后的电路带有防抖功能，能够滤除一些干扰造成的尖细脉冲信号。

下图为复位电路的总体概述框图。

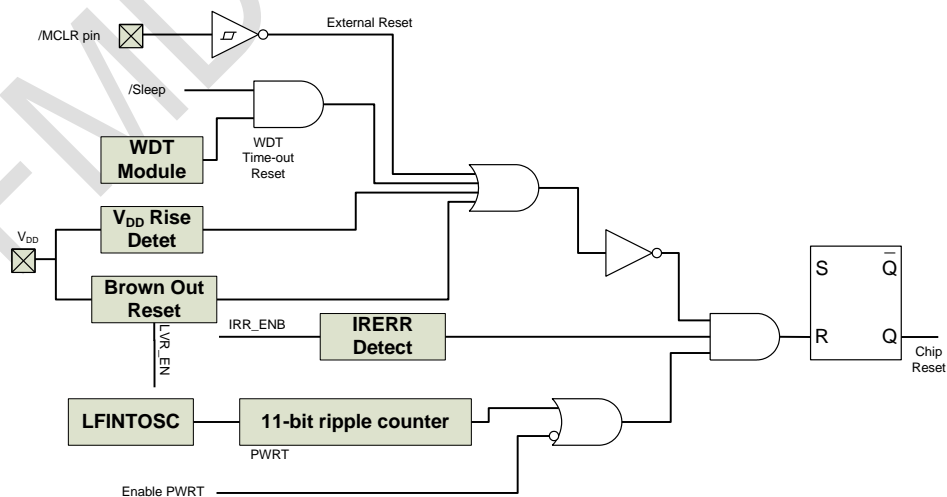


图 4.1 复位功能框图

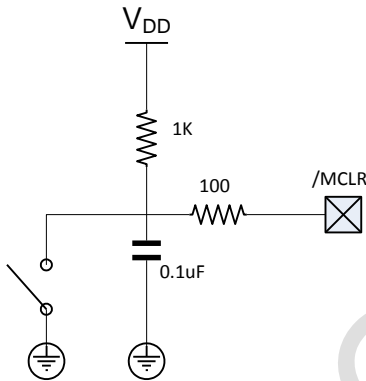
## 4.1 POR 上电复位

片上的 POR 电路会将芯片保持在复位状态直到 VDD 电源电压达到足够高，为充分利用片内复位电路的功能，用户可以简单地直接在 VDD 和/MCLR 之间接一个电阻。这样外部就无需任何 RC 复位电路。不过这要求 VDD 电压上升时间为最大。

上电完成后，系统复位不会立即释放，还要等一个约 4ms 的延时，期间数字电路保持在复位状态。

## 4.2 外部复位 MCLR

需要注意的是，WDT 复位不会把/MCLR 管脚拉低。在/MCLR 管脚上施加超过指标的电压（例如 ESD 事件）会造成/MCLR 复位，而且在管脚上产生超标的大电流，因此我们推荐用户不再直接用一个电阻将/MCLR 和 VDD 连接起来，而是采用以下电路。



在芯片的 CONFIG OPTION 寄存器（UCFG0）中有一个 MCLRE 使能位，将此位清零会使得复位信号由芯片内部产生。当此位为 1 时，芯片的 PA1/MCLR 脚成为外部复位脚。在这个模式下，/MCLR 脚上有个对 VDD 的弱上拉。

### 注意：

在 A 版本芯片中，配置选项把 PA1 设置为 MCLRB 功能时，PA1 没有复位功能，换句话说，A 版本芯片没有外部复位功能。

在 B 版本之后（包括 B 版本），该问题已经修复。

## 4.3 PWRT（上电计时器）

PWRT 为上电复位，低电压复位提供一个固定的 64ms（正常情况下）的定时。这个定时器由内部慢时钟驱动。芯片在定时器超时之前都是被保持在复位状态。这段时间能保证 VDD 上升到足够高的电压使得系统能正常工作。PWRT 也可以通过系统 CONFIG 寄存器（UCFG0）来使能。在开启低电压复位功能时，用户应该也打开 PWRT。PWRT 定时是由 VDD 电压超过 VBOR 门限事件启动的。另外需要注意的是，由于由内部慢时钟驱动，定时的实际时间长度是随温度，电压等条件变

化而变化的。这个时间不是一个精准参数。

## 4.4 BOR 低电压复位

低电压复位由  $UCFG1<1:0>$  位和  $MSCKCON$  的  $SLVREN$  位来控制。低电压复位就是指当电源电压低于  $VBOR$  门限电压时所产生的复位。不过当  $VDD$  电压低于  $VBOR$  不超过  $TBOR$  时间时，低电压复位可能不会发生。

$VBOR$  电压在芯片出货之前需要校准，校准可通过串口写入内部校准寄存器来完成。如果 BOR（低电压复位）是使能（ $UCFG1<1:0>=00$ ）的，那么最大  $VDD$  电压上升时间的要求就不存在。BOR 电路会将芯片控制在复位状态，一直到  $VDD$  电压达到  $VBOR$  门限电压以上。需要注意的是，当  $VDD$  低于系统能正常工作的门限时，POR 电路并不会产生复位信号。

当  $UCFG1<1:0>=10$  时，BOR 电路关闭将由 CPU 的运行状态决定：CPU 正常工作时 BOR 电路工作，CPU 处于睡眠模式时 BOR 电路关闭，这样可以方便的使系统功耗降至更低水平。

低电压复位结束后，内部复位不会立即释放，而是有一个  $4ms$  左右的复位延时时间。

### 4.4.1 频率与最小工作电压的关系

系统时钟频率越高，MCU 的对最小工作电压（安全工作电压）的要求就越高，例如，2T 工作模式下如果要跑 16M， $VDD$  应至少在 2.7V 以上。

FT61F04x 系列芯片的上电复位电压  $V_{POR}$  在 1.6V 左右，即当上电过程  $VDD$  超过  $V_{POR}$  之后复位释放，再经过约  $4ms$  延时之后完成配置，然后开始执行程序指令。对于一些慢上电且要跑 16M/2T 的应用，如果  $VDD$  从  $V_{POR}$  到 2.7V 的  $VDD_{min}$  的时间过长，而恰好在这段“死区时间”内程序切换到最高的系统时钟 16M 的话，MCU 将有可能跑飞。

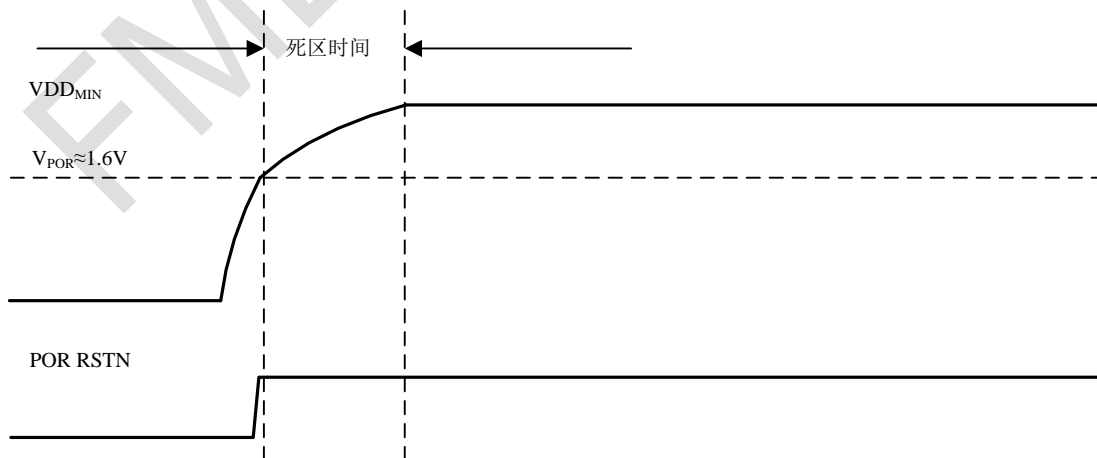


图 4.2 上电过程的“死区时间”

对于慢上电的应用，有以下几种方法可避免这种情况发生：

1. 烧录选项 **LVR** 必须使能且设置合适的值，比如 **16M/2T** 应该设置 **2.8V** 的复位电压；
2. 上电复位后，软件可延时足够长的时间让 **VDD** 升到安全工作电压后再切换到 **16M** 的系统时钟，即延时一段时间再做时钟的初始化；
3. 使能 **PWRT** 选项，**PWRT** 时间大约为 **64ms**，这段额外的复位时间有利于让 **VDD** 爬升到的最小工作电压；

以上 3 点中，强烈建议采用第 1 种方式，因为它不仅能解决上电速度过慢的问题，还能监测在正常工作时的 **VDD** 意外跌落。

## 4.5 看门狗复位

程序在正常执行（非 **SLEEP** 模式）过程中发生了看门狗溢出事件，将产生一次 **WDT** 复位，跟上电复位和低电压复位一样，看门狗复位也有一个 **4ms** 的复位延时时间。

## 4.6 错误指令复位

当 CPU 的指令寄存器取指到非法指令时，系统将进行复位，利用此功能可增加系统的抗干扰能力。

## 4.7 LVD 低电压侦测

除了低电压复位功能外，芯片还内置有低电压侦测功能。当电源电压低于设置的电压档位（由 **PCON** 的 **LVDL<2:0>** 选择）超过 **TBOR**（3 到 4 个慢时钟周期）以上时，标志位 **LVDW** 将会被置 1。软件可以利用此位来监控电源电压。

## 4.8 超时动作

在上电过程中，芯片内部的超时动作顺序按以下流程执行：

**POR** 结束后启动 **PWRT** 计时

由于计时是由 **POR** 脉冲结束启动的，如果 **/MCLR** 在低电平状态下保持足够长的时间，超时事件就会发生。那么将 **/MCLR** 拉高会让 CPU 立即开始执行。这在测试或者需要多个 MCU 同步的情况下会很有用。

**Power Control Register (PCON)**

**PCON** 寄存器里有 2 位指示哪一种复位发生了。**Bit0** 是 **/BOR** 指示位，其在上电复位是未知态，软件必须将其置 1，然后检查其是否为 0。**Bit1** 是 **/POR** 指示位，其在上电复位后为 0，软件必须将其置 1。

## 4.9 上电配置过程

发生上电复位、低电压复位、看门狗溢出复位以及错误指令复位后，除了固有的 4ms 复位延时外，还有一个初始化配置寄存器 UCFGx 的动作。该动作从 PROM 的保留地址读取内容写到 UCFGx，待所有配置地址读取完成后，才可以释放系统复位，如图 4.2 和图 4.3 的所示，该过程大概需要 17us。

## 4.10 上电校验过程

如果 UCFG1.6(CSUMENB)为 0，则上电配置过程结束后，CPU 不会立即执行程序，而是会启动 PROM 的内部校验过程。硬件会把 PROM 主程序区内容读出来自加，4K 字全部做完累加后和存储在 0x2007 的值相比较，如果相等说明校验成功，CPU 会从 0 地址执行程序，如图 4.5 所示；如果不相等说明校验失败，CPU 将不会执行程序。校验过程大概需要耗时 1ms。

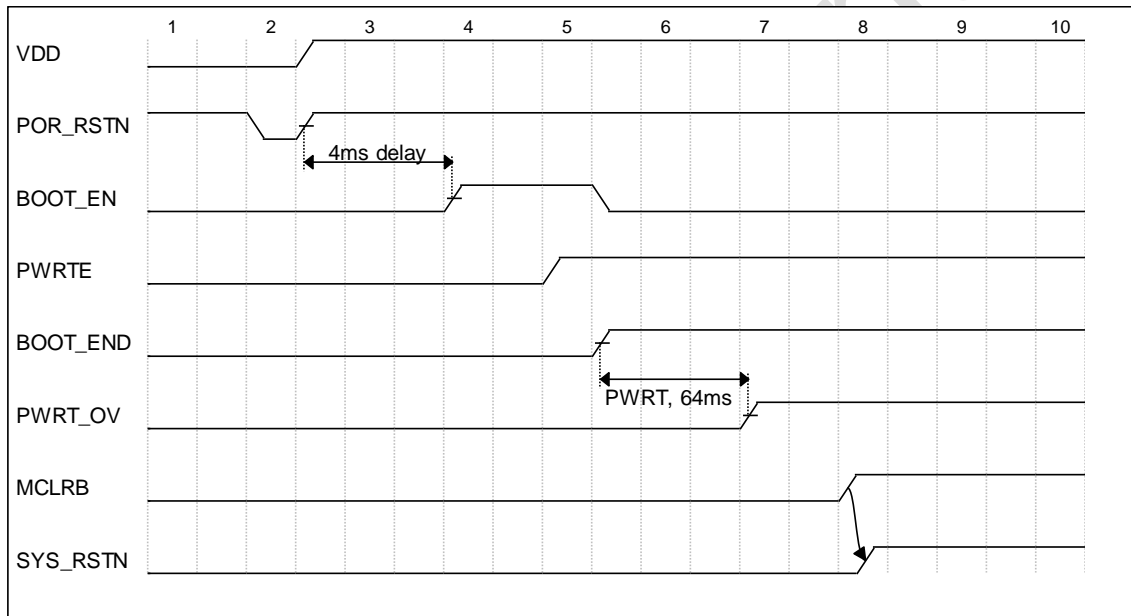


图 4.3 上电复位，使用了 MCLR

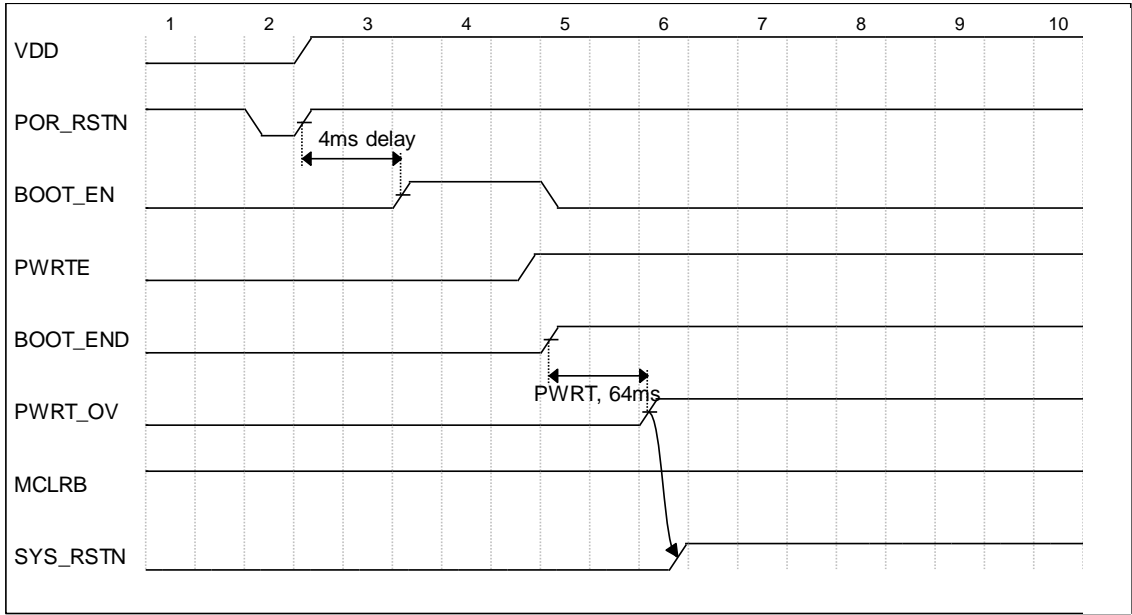


图 4.4 上电复位，没使用 MCLR

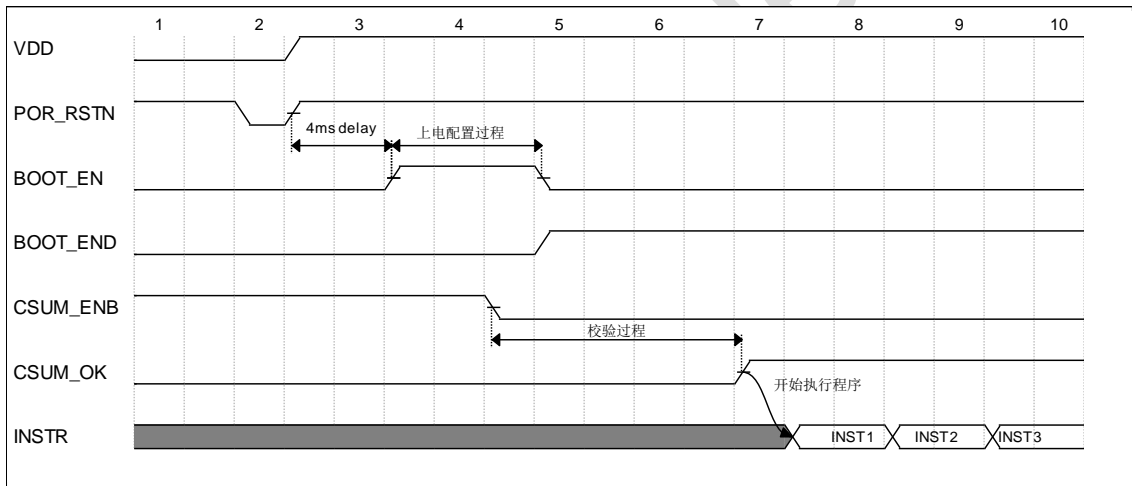


图 4.5 校验过程

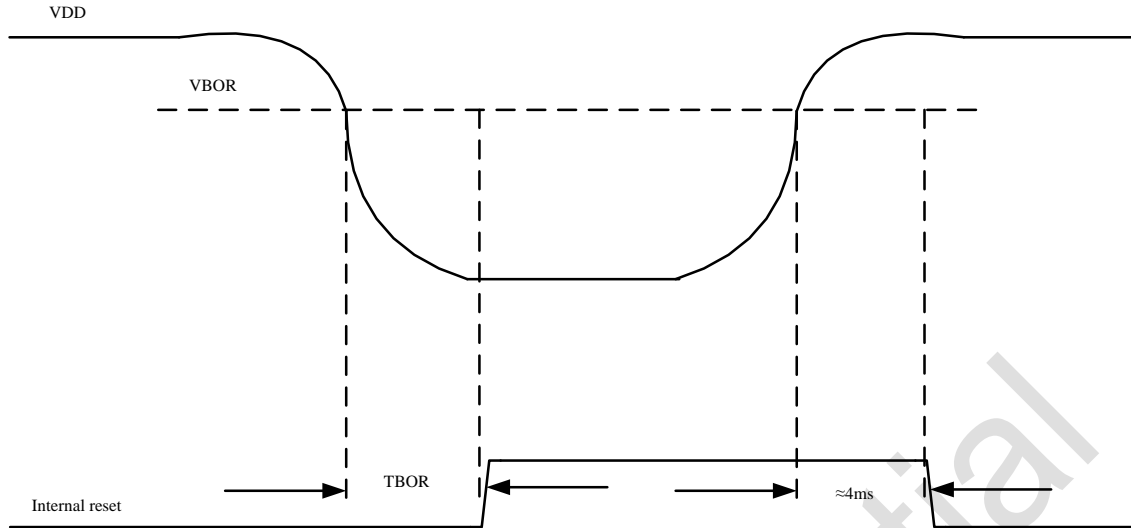


图 4.6 BOR 复位

注意:

1. 上电复位或低电压复位后，并且在 **PWRTEB (UCFG0.4)** 为低时，**PWRT** 有效。它是 **2048** 个内部慢时钟周期，约 **64ms**；
2. **TBOR** 时间约为 **122~152us**；
3. 电压恢复正常之后，内部复位不会立即释放，而是要等约为 **4ms** 的时间。

振荡器配置	上电复位		低电压复位		睡眠醒来
	/PWRTEB=0	/PWRTEB=1	/PWRTEB=0	/PWRTEB=1	
INTOSC	TPWRT	-	TPWRT	-	-

表 4.1 各种情况下的超时

/POR	/BOR	/TF	/PF	条件
0	x	1	1	POR
u	0	1	1	BOR
u	u	0	u	WDT 复位
u	u	0	0	WDT 唤醒
u	u	u	u	常规运行下/MCLR 复位
u	u	1	0	睡眠下/MCLR 复位

表 4.2 STATUS/PCON 位及其意义 (u-没变化 x-未知)

### 4.11 PCON 寄存器，地址 0x8E

Bit	7	6	5	4	3	2	1	0
Name	SYSON	LVDL[2:0]			LVDEN	LVDW	/POR	/BOR
Reset	0	0	0	0	0	-	q	q
Type	RW	RW	RW	RW	RW	RO	RW	RW

Bit	Name	Function																		
7	SYSON	睡眠时系统时钟的控制 1 = 睡眠时系统时钟保持运行 0 = 睡眠时系统时钟关闭																		
6:4	LVDL	低电压侦测选择位， <b>这些位不受低电压复位的影响</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>值</th> <th>检测电压</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>保留</td> </tr> <tr> <td>001</td> <td>保留</td> </tr> <tr> <td>010</td> <td>2.0V</td> </tr> <tr> <td>011</td> <td>2.4V</td> </tr> <tr> <td>100</td> <td>2.8V</td> </tr> <tr> <td>101</td> <td>3.0V</td> </tr> <tr> <td>110</td> <td>3.6V</td> </tr> <tr> <td>111</td> <td>4.2V</td> </tr> </tbody> </table>	值	检测电压	000	保留	001	保留	010	2.0V	011	2.4V	100	2.8V	101	3.0V	110	3.6V	111	4.2V
值	检测电压																			
000	保留																			
001	保留																			
010	2.0V																			
011	2.4V																			
100	2.8V																			
101	3.0V																			
110	3.6V																			
111	4.2V																			
3	LV DEN	低电压侦测使能， <b>该位不受低电压复位的影响</b> 1: 开启 LVD 侦测功能 0: 关闭 LVD 侦测功能																		
2	LV DW	低电压标志位，只读 1: VDD 掉到了 LV DL[2:0]所设置的电压超过 100us 以上 0: VDD 正常，高于 LV DL[2:0]所设置的电压																		
1	/POR	上电复位标志，低有效 0: 发生了上电复位 1: 没发生上电复位或者由软件置 1 /POR 在上电复位后值为 0，此后软件应该将其置 1																		
0	/BOR	低电压复位标志，低有效 0: 发生了低电压复位 1: 没发生低电压复位或者由软件置 1 /BOR 在上电复位后其值不确定，必须由软件置 1。发生后续复位后，通过查询此位来确定是否低电压复位																		



## 5 看门狗定时器

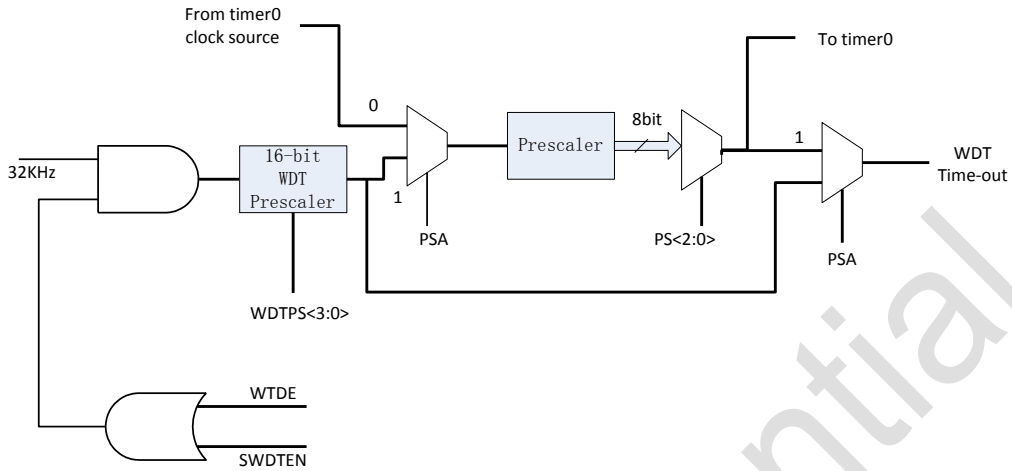


图 5.1 看门狗和定时器 0 框图

### 5.1 看门狗

看门狗的时钟源为内部慢时钟（32KHz/256K），它是一个 16 位的计数器，和定时器 0 共用一个 8 位的预分频器，使能位位于配置寄存器 UCFG0 的第 3 位，WDTEN。为 1 时表示使能看门狗，为 0 时禁止，由上电启动过程 BOOT 决定，或者可通过外部串口写入。

清看门狗指令 CLRWDT、SLEEP 会清除看门狗计数器。

在使能了看门狗的情况下，MCU 睡眠时看门狗溢出事件可以作为一个唤醒源，而 MCU 正常工作时作为一个复位源。

条件	看门狗状态
WDTEN 和 SWDTEN 同时为 0	清零
CLRWDT 指令	
进入 SLEEP、退出 SLEEP 时刻	

注意：

1. 如果内部慢时钟从 32K 切换到 256K 模式(或反之从 256K 切换到 32K 模式)，都不影响看门狗计时，因为 WDT 固定使用 32K 时钟源，见 3.1 小节的时钟框图。

### 5.2 与看门狗相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
WDTCN	0x18	-	-	-	WDTPS[3:0]				SWDTEN	---0 1000

UCFG0	0x2000	-	CPB	MCLRE	PWRTEB	WDTE	FOSC2	FOSC1	FOSC0	---- ----
OPTION	0x81	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111

### 5.2.1 WDTCON 寄存器，地址 0x18

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN
Reset	-	-	-	0	1	0	0	0
Type	-	-	-	RW	RW	RW	RW	RW

Bit	Name	Function
7:5	N/A	-
4:1	WDTPS[3:0]	<p><b>WDTPS&lt;3:0&gt;</b>: 看门狗定时器周期选择:</p> <p>Bit Value = 看门狗定时器驱动时钟之预分频比值</p> <p>0000 = 1:32</p> <p>0001 = 1:64</p> <p>0010 = 1:128</p> <p>0011 = 1:256</p> <p>0100 = 1:512 (复位值)</p> <p>0101 = 1:1024</p> <p>0110 = 1:2048</p> <p>0111 = 1:4096</p> <p>1000 = 1:8192</p> <p>1001 = 1:16384</p> <p>1010 = 1:32768</p> <p>1011 = 1:65536</p> <p>1100 = 1:65536</p> <p>1101 = 1:65536</p> <p>1110 = 1:65536</p> <p>1111 = 1:65536</p>
0	SWDTEN	<p>软件打开/关闭看门狗定时器:</p> <p>1 = 打开</p> <p>0 = 关闭</p>

## 6 定时器 0

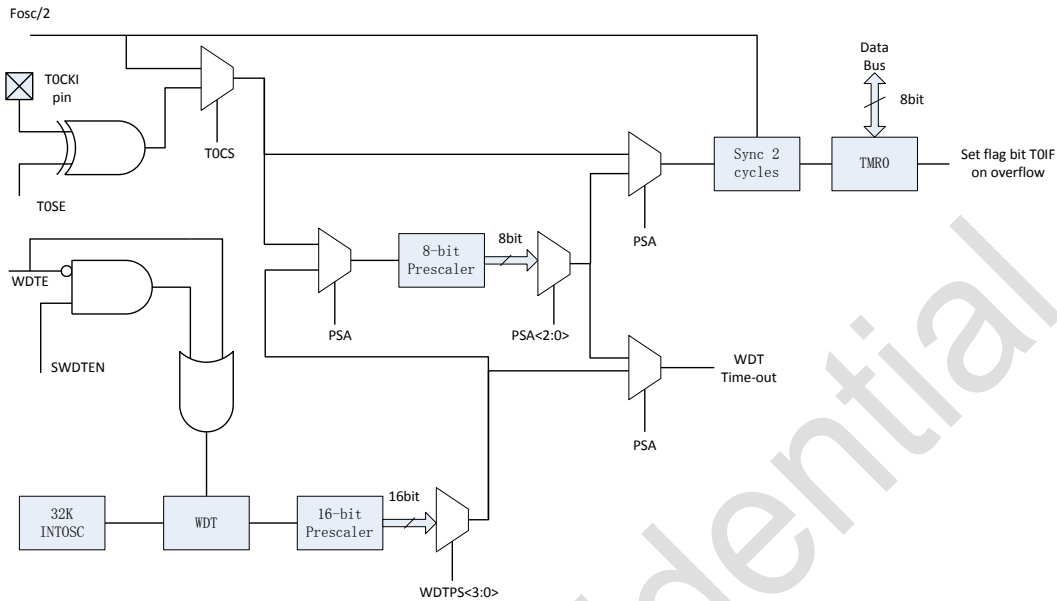


图 6.1 看门狗和定时器 0 框图

### 6.1 Timer 0

定时器 0 为 8 位，可配置为计数器或定时器使用，当作为外部事件（PC0/T0CKI 管脚）计数器时，可以配置为上升沿或者下降沿计数。作为定时器时，其计数时钟为系统时钟的 2 分频，即每一指令周期递增一次。

有一个与 WDT 共用的 8 位预分频器，PSA 为 0 时该预分频器分配给定时器 0 使用。

注意：

1. 当改变 PSA 的值时，硬件会自动把预分频器清 0。

### 6.2 Timer 0 定时器模式

该模式下，定时器 0 在每个指令周期加 1(不带预分频)。软件可以清零 OPTION 寄存器里的 TOCS 位以进入定时器模式。当软件对 TMR0 进行写操作时，定时器在写后面 2 个周期内不会递增。

### 6.3 Timer 0 计数器模式

该模式下，定时器 0 由每个 T0CKI 管脚的上升沿或下降沿触发加 1(不带预分频)。具体哪一钟沿触发由 OPTION 寄存器里的 TOSE 位决定。软件可以将 OPTION 寄存器里的 TOCS 位置 1 以进入计

计数器模式。

### 6.3.1 软件可配置预分频电路

芯片在定时器 0 和 watchdog 定时器前面有一个分频电路，可以分配给 Timer0 或者 watchdog 定时器用，但二者不能同时使用这个预分频电路。具体分配给 Timer0 还是 watchdog 由 OPTION 寄存器里的 PSA 位决定，PSA 为 0 时，预分频分配给 Timer0 使用。在 Timer0 预分频模式下，总共有 8 个预分频比（1:2 到 1:256）可以通过 OPTION 寄存器里的 PS[2:0]位设置。预分频电路既不可读也不可写。任何对 TMR0 寄存器的写动作会清零预分频电路。

当预分频电路分配给 watchdog 时，1 条 CLRWDT 指令会清零预分频电路。

#### 6.3.1.1 在定时器和 watchdog 之间切换预分频电路

由于分频电路可以分配给 Timer0 或者 watchdog 定时器用，在二者之间切换预分频器是有可能导致误复位。

在将预分频电路从分配给 TMR0 切换到分配给 watchdog 时，请务必执行以下指令顺序：

```
BANKSEL TMR0
CLRWDT ;Clear WDT
CLRR TMR0 ;Clear TMR0 and prescaler
BANKSEL OPTION_REG
BSR OPTION_REG,PSA ;Select WDT
CLRWDT

LDWI b'11111000' ;Mask prescaler bits
ANDWR OPTION_REG,W
IORWI b'00000101' ;Set WDT prescaler bits to 1:32
LDWI OPTION_REG
```

在将预分频电路从分配给 watchdog 切换到分配给 TMR0 时，请务必执行以下指令顺序：

```
CLRWDT ;Clear WDT and prescaler
BANKSEL OPTION_REG
LDWI b'11110000' ;Mask TMR0 select and prescaler bits
ANDWR OPTION_REG,W
IORWI b'00000011' ;Set prescale to 1:16
STR OPTION_REG
```

### 6.3.2 定时器 0 中断

芯片在定时器 0 从 0xFF 溢出到 0x00 时会置起 T0IF 标志，并产生中断（如果使能了的话）。注意，timer0 中断无法唤醒 CPU 因为在睡眠状态下，定时器是被冻结的。

### 6.3.3 用外部时钟驱动定时器 0

在计数其模式下，T0CKI 管脚输入和 Timer0 寄存器之间的同步是由在 Q1, Q2 内部时钟相位采样实现的，所以外部时钟源周期的高电平时间和低电平时间必须满足相关时序要求。

## 6.4 与 TIMER0 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
TMR0	0x01	TIMER0 计数值寄存器								xxxx xxxx
INTCON	0x0B/8B	GIE	PEIE	TOIE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
OPTION	0x81	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
TRISA	0x85	TRISA[7:0], PORTA 方向控制								1111 1111

### 6.4.1 OPTION 寄存器，地址 0x81

Bit	7	6	5	4	3	2	1	0
Name	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	/PAPU	/PAPU: PORTA 上拉使能位 1 = 禁止 PORTA 上拉 0 = PORTA 上拉由 WPUA 决定
6	INTEDG	INTEDG:中断沿选择位 1 = 在 INT 引脚的上升沿中断 0 = 在 INT 引脚的下降沿中断
5	T0CS	T0CS: Timer0 时钟源选择位 1 = PC0/T0CKI 引脚的跳变 0 = 内部指令周期 (FOSC/2)
4	T0SE	T0SE: Timer0 时钟源边沿选择位 1 = 在 PC0/T0CKI 引脚的下降沿递增 0 = 在 PC0/T0CKI 引脚的上升沿递增
3	PSA	PSA:预分频器分配位

		1 = 预分频器分配给 WDT 0 = 预分频器分配给 Timer0 模块
2	PS2	PS<2:0>:预分频比选择位 位值      TIMER0 分频比      WDT 分频比 000          1 : 2                      1 : 1
1	PS1	001          1 : 4                      1 : 2 010          1 : 8                      1 : 4 011          1 : 16                     1 : 8
0	PS0	100          1 : 32                     1 : 16 101          1 : 64                     1 : 32 110          1 : 128                    1 : 64 111          1 : 256                    1 : 128

### 6.4.2 TMR0, 地址 0x01

Bit	7	6	5	4	3	2	1	0
Name	TMR0[7:0]							
Reset	xxxx xxxx							
Type	RW							

Bit	Name	Function
7:0	TMR0[7:0]	Timer 0 计数结果寄存器

## 7 带门控的 TIMER1 模块

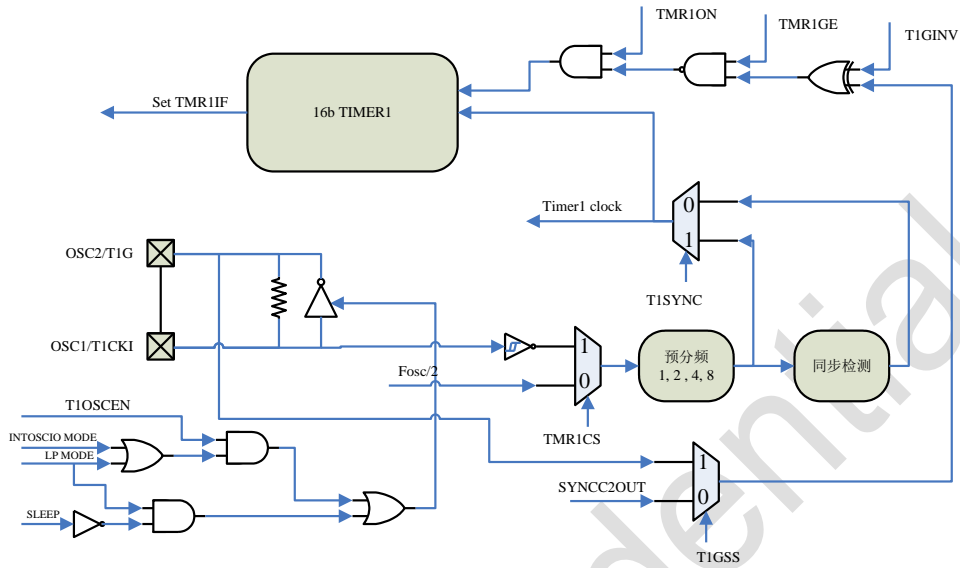


图 7.1 TIMER1 原理框图

TIMER1 是一个 16 位的定时器、计数器，有以下特性：

- 一对 16 位定时器/计数器寄存器 (TMR1H:TMR1L)
- 可编程内部或外部时钟源
- 3 位预分频器
- 可选 LP 振荡器
- 同步或异步操作
- 通过比较器或 T1G 引脚的 Timer1 门控 (计数使能)
- 溢出中断
- 溢出时唤醒 (仅限外部时钟且异步模式时)
- 捕捉/比较功能的时基
- 特殊事件触发 (带 ECCP)
- 比较器输出与 Timer1 时钟同步

### 7.1 Timer1 工作原理

Timer1 模块是 16 位递增计数器，通过一对寄存器 TMR1H:TMR1L 访问。对 TMR1H 或 TMR1L 的写操作将直接更新计数器。

与内部时钟源配合使用时，该模块为定时器。与外部时钟源配合使用时，该模块可用作定时器或计数器。

## 7.2 选择时钟源

T1CON 寄存器的 TMR1CS 位用于选择时钟源。当 TMR1CS=0 时，时钟源为 Fosc/2，即指令时钟（2T 模式下）。当 TMR1CS=1 时，时钟源由外部提供（T1CKI 管脚）。

### 7.2.1 内部时钟源

选择内部时钟源时，TMR1H:TMR1L 这对寄存器将在 FOSC 的某个倍数递增，倍数由 Timer1 预分频器决定。

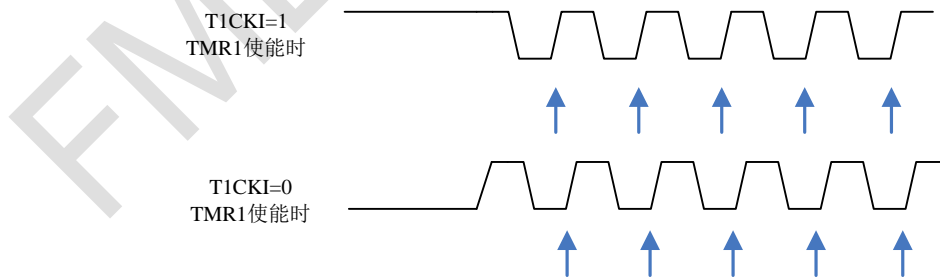
### 7.2.2 外部时钟源

选择外部时钟源时，Timer1 模块可作为定时器，也可作为计数器工作。计数时，Timer1 在外部时钟输入 T1CKI 的上升沿递增。此外，计数器模式时钟可同步到单片机系统时钟，也可异步运行。

若需外部时钟振荡器（且单片机使用不带 CLKOUT 的 INTOSC），则 Timer1 可使用 LP 振荡器作为时钟源。

**注意**，在计数器模式下，发生以下任何一个或多个情况后，计数器在首个上升沿递增前，必须先经过一个下降沿：

- 在计数器模式下，发生以下任何一个或多个情况后，计数器在首个上升沿递增前，必须先经过一个下降沿：
- POR 复位后使能 Timer1
- 写入 TMR1H 或 TMR1L
- Timer1 被禁止
- T1CKI 为高电平时 Timer1 被禁止（TMR1ON=0），然后在 T1CKI 为低电平时 Timer1 被使能（TMR1ON=1）。



注意：

1. 箭头所指边沿为计数器递增；
2. 计数器模式下，计数器递增之前必须先经过一个下降沿。

图 7.2 TIMER1 边沿递增示意



## 7.3 Timer1 预分频器

Timer1 有四种预分频选项，可对时钟输入进行 1、2、4 或 8 分频。T1CON 寄存器的 T1CKPS 位控制预分频计数器。预分频计数器不可直接读写；但当发生对 TMR1H 或 TMR1L 的写操作时，或者 TIMER1 被关闭时（TMR1ON 为 0），预分频计数器被清零。

## 7.4 Timer1 振荡器

OSC1（输入）引脚与 OSC2（放大器输出）引脚之间内置有一个低功耗 32.768kHz 晶振电路。将 T1CON 寄存器的 T1OSCEN 控制位置 1 使能振荡器。休眠时振荡器继续工作。

Timer1 振荡器与系统 LP 振荡器共用。这样，Timer1 就只能在主系统时钟来自内部振荡器或振荡器处于 LP 模式时使用此模式。用户必须提供软件延时以确保振荡器正常起振。

注意：

1. 振荡器在使用前需要一段起振和稳定时间。所以，应将 T1OSCEN 置 1 并在使能 Timer1 之前应有一段适当的延时；
2. 配置为振荡器模式时，T1G 固定输出 1，固不能用它来门控 TIMER1。

## 7.5 Timer1 工作于异步计数器模式下

若 T1CON 寄存器的控制位 T1SYNC 置 1，则外部时钟输入不同步。定时器与内部相位时钟异步递增。若选择了外部时钟源，则定时器将在休眠时继续运行，并可在溢出时产生中断，唤醒处理器。然而，读写定时器时应特别谨慎（见第 7.5.1 节“在异步计数器模式下读写 Timer1”）。

注意：

1. 从同步操作切换为异步操作时，有可能错过一次递增；从异步操作切换为同步操作时，有可能多产生一次递增。

### 7.5.1 在异步计数器模式下读写 TIMER1

定时器运行于外部异步时钟时，读取 TMR1H 或 TMR1L 将确保读操作有效（由硬件负责）。然而，应该注意的是，用两个 8 位值来读取 16 位定时器本身就会产生某些问题，这是因为定时器可能在两次读操作之间产生溢出。

对于写操作，建议用户直接停止计数器，然后写入所期望的值。如果寄存器正进行递增计数，对定时器寄存器进行写操作，可能会导致写入竞争，从而可能在 TMR1H:TMR1L 这对寄存器中产生不可预测的值。

## 7.6 TIMER1 门控

Timer1 门控源可软件配置为 T1G 引脚或比较器 C2 的输出。这使器件可直接使用 T1G 为外部事件定时，或使用比较器 C2 对模拟事件定时。Timer1 门控源的选择请参见 CMCON1 寄存器。这个特性可以简化  $\Delta$ - $\Sigma$  A/D 转换器和许多其他应用的程序。

使用 T1CON 寄存器的 T1GINV 位可翻转 Timer1 门控，不论其来自 T1G 引脚还是比较器 C2 的输出。这将配置 Timer1 以确保事件之间存在低电平有效或高电平有效的的时间。

## 7.7 Timer1 中断

Timer1 的一对寄存器 (TMR1H:TMR1L) 递增至 FFFFh 后返回 0000h。Timer1 计满返回时，PIR1 寄存器的 Timer1 中断标志位被置 1。为确保计满返回时产生中断，必须将以下位置 1：

- T1CON 寄存器的 TMR1ON 位
- PIE1 寄存器的 TMR1IE 位
- INTCON 寄存器的 PEIE 位
- INTCON 寄存器的 GIE 位

在中断服务程序中将 TMR1IF 位清零将清除中断。

## 7.8 Timer1 在休眠模式下的工作原理

只有在设定异步计数器模式时，Timer1 才能在休眠模式下工作。在该模式下，可使用外部晶振或时钟源信号使计数器递增。要做以下设置定时器以唤醒器件：

- 必须将 T1CON 寄存器的 TMR1ON 位置 1
- 必须将 PIE1 寄存器的 TMR1IE 位置 1
- 必须将 INTCON 寄存器的 PEIE 位置 1
- 必须将 T1CON 寄存器的 T1SYNC 位置 1
- 必须将 T1CON 寄存器的 TMR1CS 位置 1
- 可将 T1CON 寄存器的 T1OSCEN 位置 1

溢出时器件将被唤醒并执行下一条指令。若 INTCON 寄存器的 GIE 位置 1，器件将调用中断服务程序 (0004h)。

## 7.9 ECCP 捕捉/比较时基

工作于捕捉或比较模式时，ECCP 模块使用一对 TMR1H:TMR1L 寄存器作为时基。在捕捉模式下，TMR1H:TMR1L 这对寄存器的值在发生某个配置好的事件时被复制到 CCPR1H:CCPR1L 这对寄存器中。

在比较模式下，当 CCPR1H:CCPR1L 这对寄存器的值与 TMR1H:TMR1L 的值匹配时，将触发一个事件。该事件可以是特殊事件触发。

更多信息请参见第 9 节“增强型捕捉/比较/PWM+（带自动关闭和死区）模块”。

## 7.10 ECCP 特殊事件触发器

当 ECCP 配置为触发特殊事件时，触发器会将 TMR1H:TMR1L 这对寄存器清零。该特殊事件不会产生 Timer1 中断。ECCP 模块仍可配置为产生 ECCP 中断。在此工作模式下，CCPR1H:CCPR1L 这对寄存器成为了 Timer1 的周期寄存器。

- Timer1 应同步为 FOSC 以充分利用特殊事件触发器
- Timer1 异步工作可导致错过特殊事件触发器
- 当对 TMR1H 或 TMR1L 的写操作与一个 ECCP 特殊事件触发器同时发生时，写操作具有优先权

更多信息请参见第 9.2.4 节“特殊事件触发器”。

## 7.11 与 TIMER1 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
TMR1L	0x0E	16 位 TIMER1 计数值低 8 位								xxxx xxxx
TMR1H	0x0F	16 位 TIMER1 计数值高 8 位								xxxx xxxx
INTCON	0x0B/8B	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
PIE1	0x8C	EEIE	CKMEAIE	-	C5IE	C4IE	OSFIE	TMR2IE	TMR1IE	00-0 0000
PIR1	0x0C	EEIF	CKMEAIF	-	C5IF	C4IF	OSFIF	TMR2IF	TMR1IF	00-0 0000
T1CON	0x10	T1GINV	TMR1GE	T1CKPS[1:0]		T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000
CMCON1	0x1A	HYSEN[3:0]				-	-	T1GSS	C5SYNC	0000 --10

### 7.11.1 T1CON 寄存器，地址 0x10

Bit	7	6	5	4	3	2	1	0
Name	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	T1GINV	TIMER1 门控翻转位 1 = Timer1 门控为高电平有效（门控为高电平时 Timer1 计数） 0 = Timer1 门控为低电平有效（门控为低电平时 Timer1 计数）
6	TMR1GE	Timer1 门控使能位(2)

		<p>如果 TMR1ON = 0: 此位被忽略</p> <p>如果 TMR1ON = 1: 1 = Timer1 在 Timer1 门控不活动时打开 0 = Timer1 打开</p>
5:4	T1CKPS[1:0]	<p>Timer1 输入时钟预分频比选择位</p> <p>11 = 1:8 预分频比 10 = 1:4 预分频比 01 = 1:2 预分频比 00 = 1:1 预分频比</p>
3	T1OSCEN	<p>LP 振荡器使能控制位</p> <p>如果无 CLKOUT 振荡器的 INTOSC 处于激活状态: 1 = LP 振荡器被使能用于 Timer1 时钟 0 = LP 振荡器关闭</p> <p>否则: 此位被忽略</p>
2	T1SYNC	<p>Timer1 外部时钟输入同步控制位</p> <p>TMR1CS = 1: 1 = 不同步外部时钟输入 0 = 同步外部时钟输入</p> <p>TMR1CS = 0: 此位被忽略。Timer1 使用内部时钟</p>
1	TMR1CS	<p>Timer1 时钟源选择位</p> <p>1 = 来自 T1CK1 引脚 (上升沿) 的外部时钟 0 = 内部时钟 (指令时钟)</p>
0	TMR1ON	<p>Timer1 使能位</p> <p>1 = 使能 Timer1 0 = 停止 Timer1</p>

### 7.11.2 TMR1L 寄存器, 地址 0x0E

Bit	7	6	5	4	3	2	1	0
Name	TMR1L[7:0]							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TMR1L[7:0]	16 位 TIMER1 定时器、计数器计数值的低 8 位

### 7.11.3 TMR1H 寄存器，地址 0x0F

Bit	7	6	5	4	3	2	1	0
Name	TMR1H[7:0]							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TMR1H[7:0]	16 位 TIMER1 定时器、计数器计数值的高 8 位

FMD Confidential

## 8 定时器 2

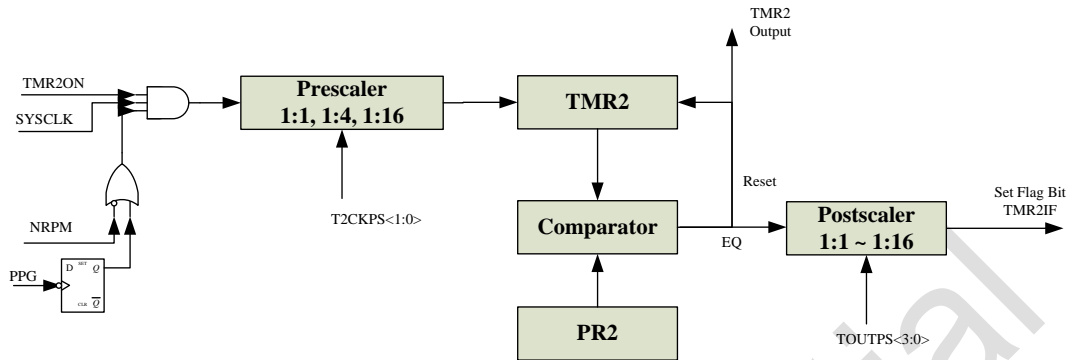


图 8.1 定时器 2 框图

定时器 2 为 8 位定时器包含以下功能：

- 8 位计数寄存器
- 8 位周期寄存器
- TMR2 值等同 PR2 时产生中断
- 1:1, 1:4, 1:16 预分频比
- 1:1~1:16 后分频比
- 时钟源可选系统时钟或者内部 32MHz 时钟（由 HFINTOSC 的 2 倍频得到）
- 对 PPG 禁止重复触发定时

图 8.1 为 Timer2 的整体框图。

### 8.1 Timer2 工作原理

Timer2 模块的时钟输入是系统指令时钟。该时钟送入 Timer2 预分频器，其预分频比有 1:1、1:4 或 1:16 三种选择。随后预分频器的输出被用于递增 TMR2 寄存器。

TMR2 和 PR2 的值被不断比较以确定何时匹配。TMR2 将从 00h 开始递增直到与 PR2 的值相同。匹配时将发生以下两种情况：

- TMR2 在下一递增周期复位为 00h
- Timer2 后分频比递增

Timer2/PR2 比较器的匹配输出送入 Timer2 后分频器。后分频器的选项范围为 1:1 至 1:16。Timer2 后分频器的输出用于将 PIR1 寄存器的 TMR2IF 中断标志置 1。

TMR2 和 PR2 都是可读写寄存器。在复位时，他们的值分别是 0 和 0xFF。

将 T2CON 寄存器中的 TMR2ON 位置 1 可打开 Timer2，反之将 TMR2ON 位清零关闭 Timer2。

Timer2 预分频器由 T2CON 寄存器的 T2CKPS 位控制。

Timer2 后分频器由 T2CON 寄存器的 TOUTPS 位控制。

预分频和后分频计数器会在写以下寄存器时清零：

- 写 TMR2
- 写 T2CON
- 任何 reset 动作

注：

1. 写 T2CON 并不会清零 TMR2 寄存器；
2. TIMER2 的时钟源由 MSCKCON.5 控制，当 T2CKSRC 为 1 时表示选择内部 32MHz 时钟，与当前运行的系统时钟无关。32MHz 时钟是由内部 HFINTOSC 倍频得到，所以一旦 TIMER2 选择 32MHz 时钟源并且 TMR2ON=1 时，即使系统时钟选择的是内部慢时钟或者外部晶体时钟，HFINTOSC 是不会关闭的，除非进入了睡眠模式。

## 8.2 PPG 禁止重复触发的定时

当 T2CON 的 NRPM 为 1 时，Timer2 可以为 PPG 的禁止重复触发提供定时。处于这种模式时，Timer2 定时器在 PPG 有效时清 0 并停止计数，一直等到 PPG 变为无效状态。假设 PPG 设置为高电平有效，则只有发生 PPG 的下降沿时 timer2 才开始计数，直到计数值和 PR2 发生匹配，这时计数器重新清 0，等待下一次 PPG 下降沿的到来，在这段时间内任何 PPG 的触发事件都会被忽略。

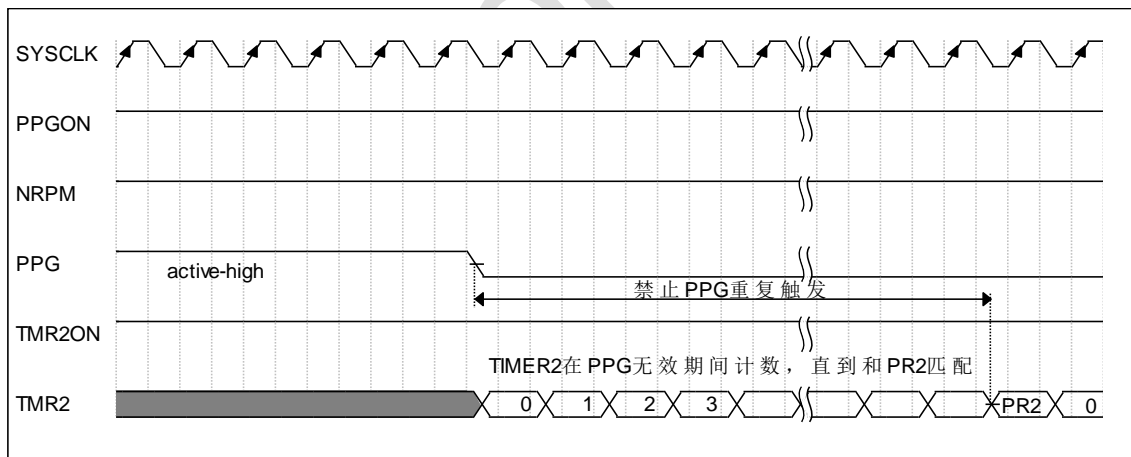


图 8.2 定时器 2 和 PPG 配合工作

## 8.3 与 TIMER2 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
TMR2	0x11	TIMER2 计数值寄存器								xxxx xxxx
INTCON	0x0B/8B	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
PIE1	0x8C	EEIE	CKMEAIE	-	C5IE	C4IE	OSFIE	TMR2IE	TMR1IE	00-0 0000

PIR1	0x0C	EEIF	CKMEAIF	-	C5IF	C4IF	OSFIF	TMR2IF	TMR1IF	00-0 0000
MSCKCON	0x1B	-	-	T2CKSRC	SLVREN	-	CKMAVG	CKCNTI	-	0000 000-
PR2	0x92	TIMER2 周期寄存器								1111 1111
T2CON	0x12	NRPM	TOUTPS[3:0]			TMR2ON	T2CKPS[1:0]		0000 0000	

### 8.3.1 PR2 寄存器，地址 0x92

Bit	7	6	5	4	3	2	1	0
Name	PR2[7:0]							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PR2[7:0]	Timer 2 周期(比较)寄存器 (详见 Timer2 描述章节)

### 8.3.2 TMR2 寄存器，地址 0x11

Bit	7	6	5	4	3	2	1	0
Name	TMR2[7:0]							
Reset	0000 0000							
Type	RW							

Bit	Name	Function
7:0	TMR2[7:0]	Timer 2 计数结果寄存器

### 8.3.3 T2CON 寄存器，地址 0x12

Bit	7	6	5	4	3	2	1	0
Name	NRPM	TOUTPS[3:0]			TMR2ON	T2CKPS[1:0]		
Reset	0	0000			0	00		
Type	RW	RW			RW	RW		

Bit	Name	Function
7	NRPM	PPG 的非重复触发模式 1 = 使能 PPG 非重复触发模式，Timer2 将配合 PPG 模块工作 0 = 关闭 PPG 非重复触发模式
6:3	TOUTPS[3:0]	<b>TOUTPS&lt;3:0&gt;</b> : Timer2 Output Postscaler Select bits 定时器2输出后分频比选择 0000 = 1:1 后分频比 0001 = 1:2 后分频比



		<p>0010 = 1:3 后分频比</p> <p>0011 = 1:4 后分频比</p> <p>0100 = 1:5 后分频比</p> <p>0101 = 1:6 后分频比</p> <p>0110 = 1:7 后分频比</p> <p>0111 = 1:8 后分频比</p> <p>1000 = 1:9 后分频比</p> <p>1001 = 1:10 后分频比</p> <p>1010 = 1:11 后分频比</p> <p>1011 = 1:12 后分频比</p> <p>1100 = 1:13 后分频比</p> <p>1101 = 1:14 后分频比</p> <p>1110 = 1:15 后分频比</p> <p>1111 = 1:16 后分频比</p>
2	TMR2ON	<p><b>TMR2ON:</b> Timer2 On bit 打开定时器2</p> <p>1 = Timer2打开</p> <p>0 = Timer2 关闭</p>
1:0	T2CKPS[1:0]	<p><b>T2CKPS&lt;1:0&gt;:</b> Timer2 Clock Prescale Select bits 定时器2驱动时钟预分频比选择</p> <p>00 = Prescaler is 1</p> <p>01 = Prescaler is 4</p> <p>1x = Prescaler is 16</p>

## 9 增强型捕捉/比较/PWM 模块

增强型捕捉/比较/PWM 模块（ECCP）是一种用户可用来对不同事件进行定时和控制的外设。在捕捉模式下，此外设可对事件的持续时间定时。比较模式使用户可在一段预定时长后触发外部事件。PWM 模式可生成频率可变的脉宽调制信号和占空比。

ECCP	定时器资源
捕捉	Timer1
比较	Timer1
PWM	Timer2

表 9.1 ECCP 各模式所需的定时器资源

### 9.1 捕捉模式

在捕捉模式下，当在 CCP1 引脚上发生某一事件时，CCPR1H:CCPR1L 捕捉 TMR1 寄存器中的 16 位值。事件定义为以下之一，并由 CCP1CON 寄存器的 CCP1M<3:0>位进行配置：

- 每个下降沿
- 每个上升沿
- 每 4 个上升沿
- 每 16 个上升沿

进行捕捉后，PIR2 寄存器中的中断请求标志位 CCP1IF 被置 1，该位必须用软件清零。如果在 CCPR1H 和 CCPR1L 这对寄存器中的值被读出之前又发生另一次捕捉，那么原来的捕捉值会被新捕捉值覆盖（见图 9.1）。

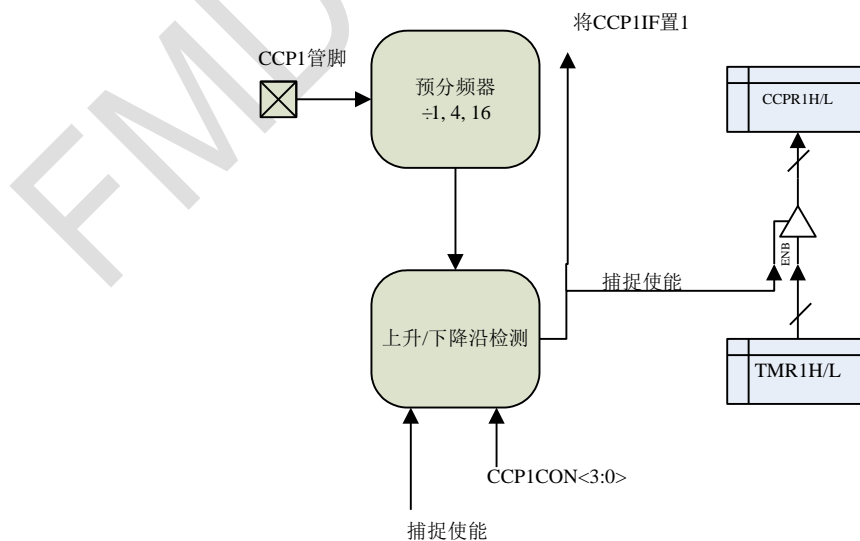


图 9.1 捕捉模式原理框图

### 9.1.1 CCP1 引脚配置

在捕捉模式下，应该通过将相关的 TRISC 控制位置 1 将 CCP1 引脚配置为输入。

**注意：**如果 CCP1 引脚配置为输出，则写端口将产生一次捕捉条件。

### 9.1.2 TIMER1 模式选择

为使 CCP 模块使用捕捉特性，Timer1 必须运行在定时器模式或同步计数器模式。在异步计数器模式下，捕捉操作可能无法进行。

### 9.1.3 软件中断

当捕捉模式改变时，可能会产生一次误捕捉中断。用户应该保持 PIE1 寄存器的 CCP1IE 位清零以避免误中断。此外，用户还应在任何这种工作模式改变之后清零 PIR2 寄存器的中断标志位 CCP1IF。

### 9.1.4 CCP 预分频器

CCP1CON 寄存器的 CCP1M<3:0>位指定了 4 种不同的预分频比。每当关闭 CCP 模块，或 CCP 模块不在捕捉模式时，预分频计数器都将被清零。任何复位都会将预分频计数器清零。

从一个捕捉预分频比切换到另一个不会清零预分频器并可能产生一次中断。要避免意外操作，在改变预分频比前应将 CCP1CON 寄存器清零，以关闭 CCP 模块（见例 9.1）。

```
BANKSEL CCP1CON      ;Set Bank bits to point to CCP1CON
CLRR CCP1CON         ;Turn CCP module off
LDWI NEW_CAPT_PS     ;Load the W reg with the new prescaler
STR CCP1CON          ;Load CCP1CON with this value
```

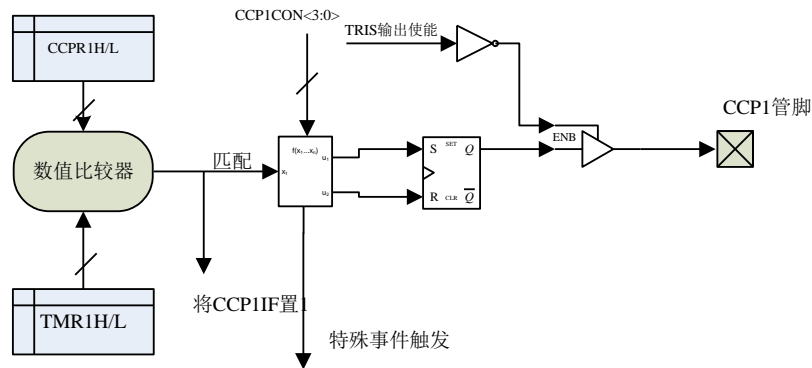
例 9.1 切换预分频比

## 9.2 比较模式

在比较模式下，16 位 CCPR1 寄存器值被不断与 TMR1 的一对寄存器的值相比较。发生匹配时，CCP 模块可能：

- 翻转 CCP1 输出
- 将 CCP1 输出置 1
- 将 CCP1 输出清零
- 触发特殊事件
- 产生软件中断

引脚上的动作取决于 CCP1CON 寄存器的 CCP1M<3:0>控制位的值。所有比较模式均可产生中断。



特殊事件触发包括:

1. 清零TMR1H和TMR1L寄存器;
2. 不会将PIR1寄存器相关标志位TMR1IF置1;
3. 将GO/DONE位置1启动ADC转换;
4. 发生系统复位, CCP1将被清0.

图 9.2 比较模式原理框图

## 9.2.1 CCP1 引脚的配置

用户必须通过清零相关的 TRIS 位将 CCP1 引脚配置为输出。

## 9.2.2 TIMER1 模式选择

在比较模式下, Timer1 必须运行在定时器模式下或同步计数器模式下。比较操作在异步计数器模式下可能不能正常工作。

## 9.2.3 软件中断模式

当选择产生软件中断模式 (CCP1M<3:0> = 1010) 时, CCP 模块并不得到 CCP1 引脚的控制权 (见 CCP1CON 寄存器)。

## 9.2.4 特殊事件触发器

当选定了特殊事件触发模式 (CCP1M<3:0> = 1011) 时, CCP 模式将会:

- 复位 Timer1
- 若 ADC 使能, 则启动一次 ADC 转换

在此模式下, CCP 模块不得到 CCP1 引脚的控制权 (见 CCP1CON 寄存器)。

一旦 TMR1H:TMR1L 这对寄存器和 CCPR1H:CPR1L 这对寄存器之间发生匹配，便会发生 CCP 的特殊事件触发输出。TMR1H:TMR1L 寄存器在 Timer1 时钟的下一个上升沿到来之前不会复位。这使 CCPR1H:CCPR1L 寄存器可作为 Timer1 的 16 位可编程周期寄存器。

注意：

1. CCP 模块的特殊事件触发不会将 PIR1 寄存器的中断标志位 TMR1IF 置 1；
2. 在生成特殊事件触发的时钟边沿和复位 Timer1 的时钟边沿之间，通过更改 CCPR1H 和 CCPR1L 这对寄存器的内容来移除匹配条件，可以预防复位的发生。

### 9.3 PWM 模式

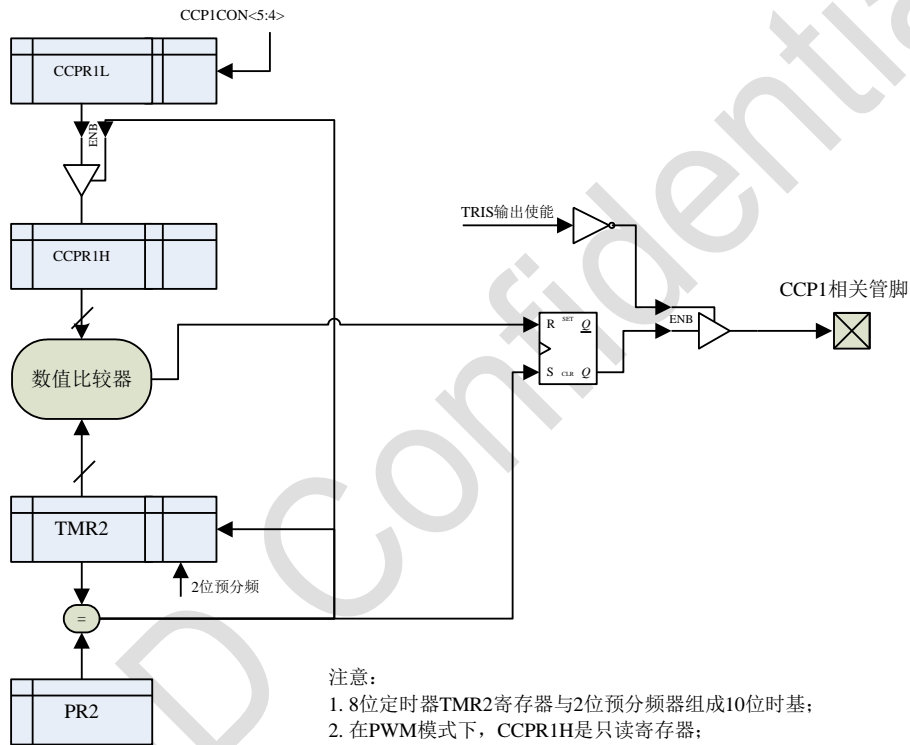


图 9.3 PWM 原理框图

PWM 模式将在 CCP1 引脚产生脉宽调制信号。其占空比，周期和分辨率由以下寄存器决定：

- PR2
- T2CON
- CCPR1L
- CCP1CON

在脉宽调制（Pulse-Width-Modulation, PWM）模式下，CCP 模块在 CCP1 引脚上产生高达 10 位分辨率的 PWM 输出。由于 CCP1 引脚与 PORT 数据锁存器复用，必须将该引脚的 TRIS 清零以使用 CCP1 引脚输出驱动器。

注意：

1. 将 CCP1CON 寄存器清零将放弃对 CCP1 引脚的控制；

2. 由于慢时钟测量也占用 **TIMER2**，所以想使用 **PWM** 功能时不能启动慢时钟测量。

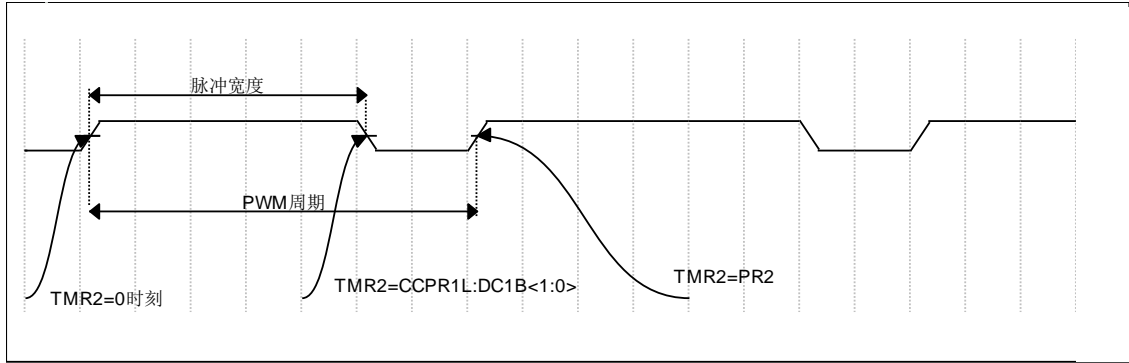


图 9.4 PWM 输出波型

### 9.3.1 PWM 周期

PWM 周期由 Timer2 的 PR2 寄存器指定。用公式 11.1 可计算 PWM 周期。

$$PWM \text{ 周期} = (PR2 + 1) * 4 * T_{sys} * (TMR2 \text{ 预分频值}) \quad \text{公式 9.1}$$

当 TMR2 等于 PR2 时，下一次递增周期将发生以下三个事件：

- TMR2 被清零
- CCP1 引脚被置 1（例外：若 PWM 占空比= 0%，引脚不被置 1）
- PWM 占空比从 CCPR1L 锁存到 CCPR1H

注意：

1. PWM 模式下，Timer2 的时钟源为系统时钟 **Fosc**，而不是指令时钟；
2. 确定 PWM 频率时不使用 Timer2 的后分频器。见第 8 节的“定时器 2”相关工作原理描述；
3. 当 **MSCKCON.5** 为 1 时，Timer2 的时钟源为 32MHz。

### 9.3.2 PWM 占空比

通过对以下几个寄存器写入 10 位值可指定 PWM 占空比：CCPR1L 寄存器和 CCP1CON 寄存器的 DC1B<1:0>位。CCPR1L 包含八个 MSB，CCP1CON 寄存器的 DC1B<1:0>位包含两个 LSB。CCP1CON 寄存器的 DC1B<1:0>和 CCPR1L 可在任何时候被写入。占空比直到周期完成时（即 PR2 和 TMR2 寄存器发生匹配时）才被锁存到 CCPR1H 中。使用 PWM 时，CCPR1H 寄存器是只读的。

公式 9.2 用于计算 PWM 脉宽。

公式 9.3 用于计算 PWM 占空比。

$$脉冲宽度 = (CCPR1L:CCP1CON<5:4>) * T_{sys} * (TMR2 \text{ 预分频值}) \quad \text{公式 9.2}$$

$$\text{占空比} = (\text{CCPR1L}:\text{CCP1CON}\langle 5:4 \rangle) \div (4 * (\text{PR2} + 1)) \quad \text{公式 9.3}$$

CCPR1H 寄存器和 2 位的内部锁存器用于为 PWM 占空比提供双缓冲。双缓冲对 PWM 的无毛刺工作起着非常重要的作用。

8 位定时器 TMR2 寄存器与 2 位的内部系统时钟 (FOSC) 或 2 位的预分频器连接, 组成 10 位时基。如果 Timer2 预分频器置为 1:1, 则使用系统时钟。

当 10 位时基与 CCPR1H 及 2 位的锁存器匹配时, CCP1 引脚被清零 (见图 9.3)。

### 9.3.3 PWM 分辨率

分辨率决定某个周期的有效占空比。例如, 10 位分辨率有 1024 个分立的占空比, 而 8 位分辨率则有 256 个分立的占空比。

PR2 为 255 时产生 10 位的最大 PWM 分辨率。分辨率是 PR2 寄存器值的函数, 如公式 9.4 所示。

$$\text{分辨率} = \log_2[4(\text{PR2} + 1)] - \log_2(2) \text{ 位} \quad \text{公式 9.4}$$

**注意:** 如果脉冲宽度大于周期, 那么分配的 PWM 引脚将保持不变。

PWM 频率	1.22kHz	4.88kHz	19.53 kHz	78.12 kHz	156.3kHz	263.1kHz
Timer2 预分频比	16	4	1	1	1	1
PR2	204	204	204	50	25	18
最大分辨率	9.7	9.7	9.7	7.7	6.7	6.3

表 9.1 PWM 频率和分辨率示例 (Fosc=20MHz)

### 9.3.4 休眠模式下的工作

在休眠模式下, TMR2 寄存器不递增, 模块的状态不变。如果 CCP1 引脚正在驱动一个值, 它将继续驱动该值。器件唤醒时, TMR2 将继续先前的状态。

### 9.3.5 系统时钟频率的改变

PWM 频率来自系统时钟频率, 系统时钟频率的任何改变将导致 PWM 频率的改变。

**注意:**

1. 当 TIMER2 的时钟源选择内部 32MHz 时, 系统时钟频率的改变不会影响 PWM 周期。

### 9.3.6 复位的影响

任何复位均将强制所有端口为输入模式，并强制 CCP 寄存器为其复位状态。

### 9.3.7 PWM 工作的设置

应按照以下步骤将 CCP 模块配置为 PWM 工作：

1. 将相关的 TRIS 位置 1 禁止 PWM 引脚（CCP1）的输出驱动器；
2. 装载 PR2 寄存器以设置 PWM 周期；
3. 用适当的值装载 CCP1CON 寄存器将 CCP 模块配置为 PWM 模式；
4. 装载 CCPR1L 寄存器和 CCP1CON 寄存器的 DC1B<1:0>设置 PWM 占空比；
5. 配置并启动 Timer2：
  - 将 PIR1 寄存器的 TMR2IF 中断标志位清零
  - 装载 T2CON 寄存器的 T2CKPS 位设置 Timer2 预分频比
  - 将 T2CON 寄存器的 TMR2ON 位置 1 使能 Timer2
  - 如果要设置高速模式，则需要把 MSCKCON.5 置 1
6. 重新开始一个 PWM 周期后，使能 PWM 输出：
  - 等待 Timer2 溢出（PIR1 寄存器的 TMR2IF 位置 1）
  - 将相关的 TRIS 位清零使能 CCP1 引脚的输出驱动器

### 9.4 PWM（增强模式）

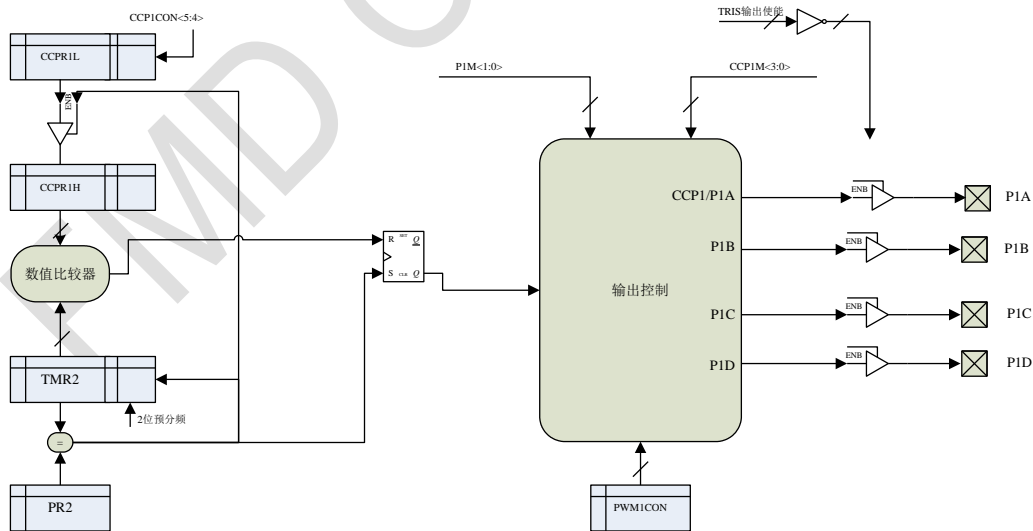


图 9.5 增强型 PWM 模式原理框图

增强型 PWM 模式可在最多四个输出引脚上产生高达 10 位分辨率的 PWM 信号。它通过四种 PWM 输出模式做到：



- 单 PWM
- 半桥 PWM
- 全桥 PWM，正向模式
- 全桥 PWM，反向模式

要选择增强型 PWM 模式，CCP1CON 寄存器的 P1M 位必须被正确设置。

PWM 输出与 I/O 引脚复用，并被指定为 P1A、P1B、P1C 和 P1D。PWM 引脚的极性可配置，可通过将 CCP1CON 寄存器中的 CCP1M 位进行适当置 1 选择极性。

注意：

1. 必须正确配置每个 PWM 输出的 TRIS 寄存器值；
2. 清零 CCP1CON 寄存器将放弃所有 PWM 输出引脚的 ECCP 控制权；
3. 增强型 PWM 模式所不使用的任何引脚均可用于其他引脚功能。

ECCP	P1M<1:0>	P1A	P1B	P1C	P1D
单 PWM	00	是	否	否	否
单桥 PWM	10	是	是	否	否
全桥，正向	01	是	是	是	是
全桥，反向	11	是	是	是	是

表 9.2 不同 PWM 增强模式的引脚分配示例

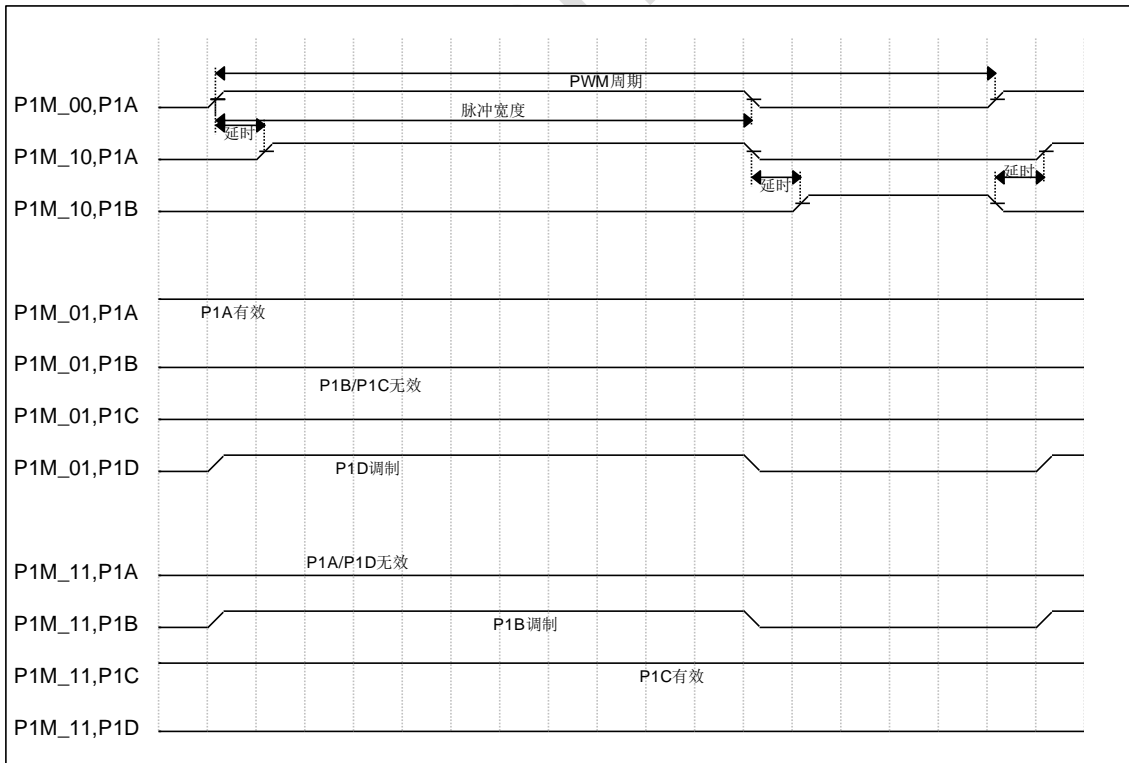


图 9.6 PWM 输出关系示意图（高电平有效）

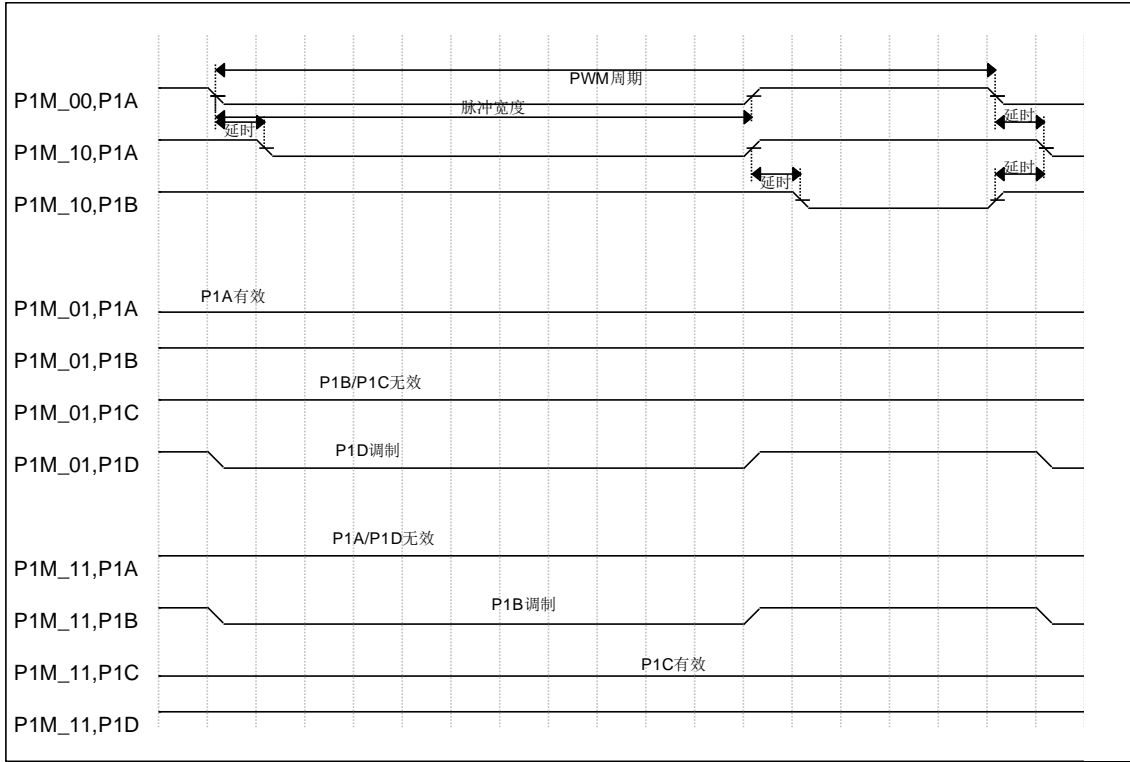


图 9.7 PWM 输出关系示意图（低电平有效）

### 9.4.1 半桥模式

在半桥模式下，有两个引脚用作输出以驱动推挽负载。PWM 输出信号被输出到 P1A 引脚，而互补 PWM 输出信号被输出到 P1B 引脚（见图 9.6）。此模式可用于半桥应用，或用于全桥应用，此时用两个 PWM 信号调制四个功率开关。

在半桥模式下，可使用编程死区延时防止半桥功率器件中出现穿通电流。PWM1CON 寄存器的 PDC<6:0>位用于设置将输出驱动为有效前的指令数。如果该值大于占空比，则相应的输出在整个周期中将保持无效状态。死区延时操作的详情请参见第 9.4.6 节“可编程死区延时模式”。

由于 P1A 和 P1B 输出与 PORT 数据锁存器复用，必须清零相关的 TRIS 位以将 P1A 和 P1B 配置为输出。

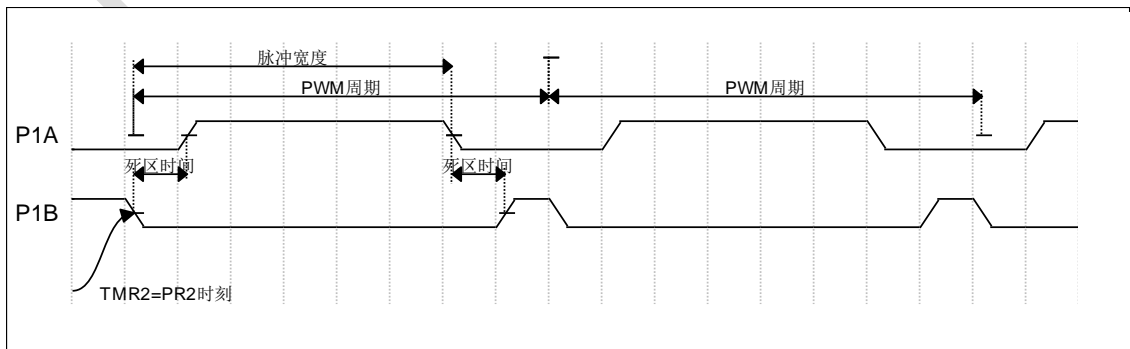


图 9.8 半桥 PWM 输出

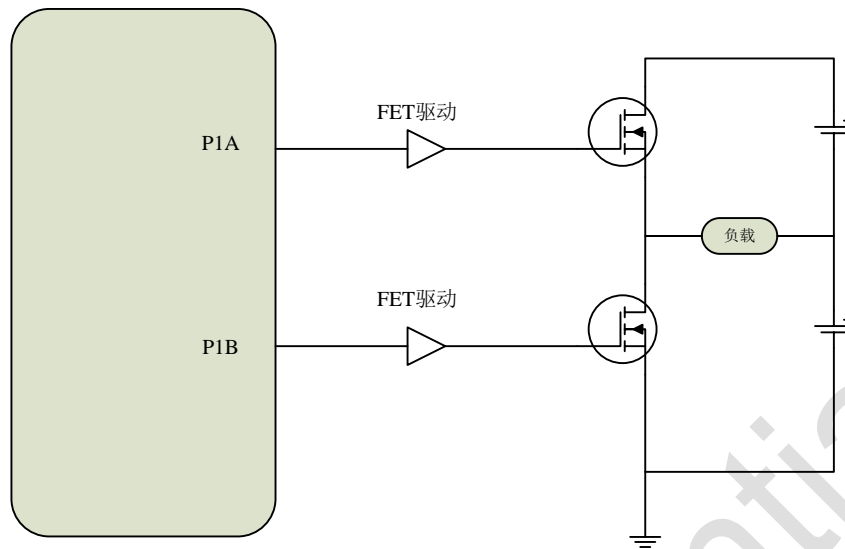


图 9.9a 标准半桥电路（推挽）

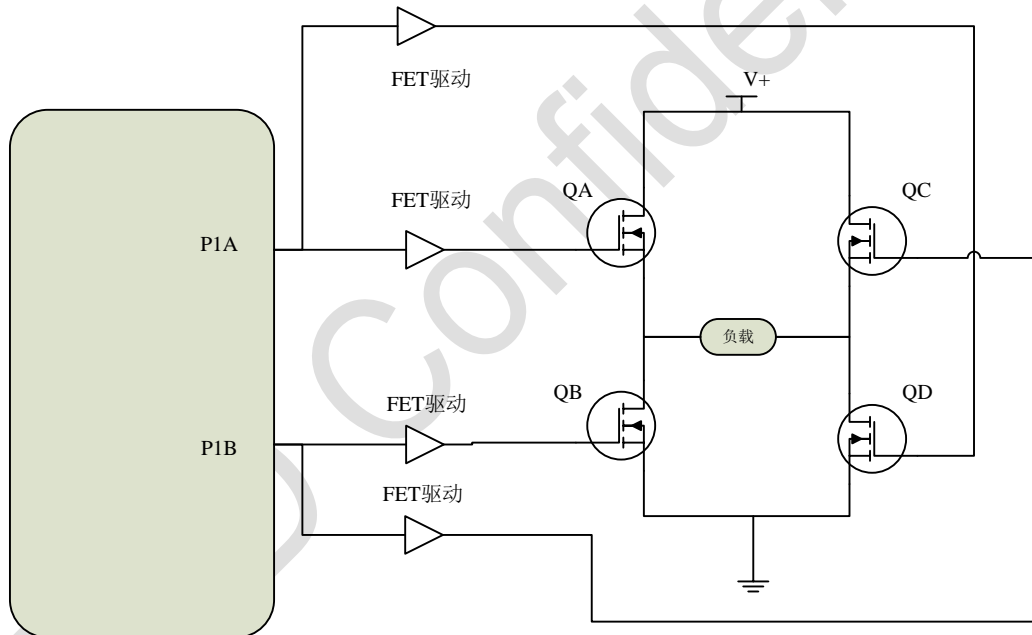


图 9.9b 半桥输出驱动全桥电路（4NMOS）

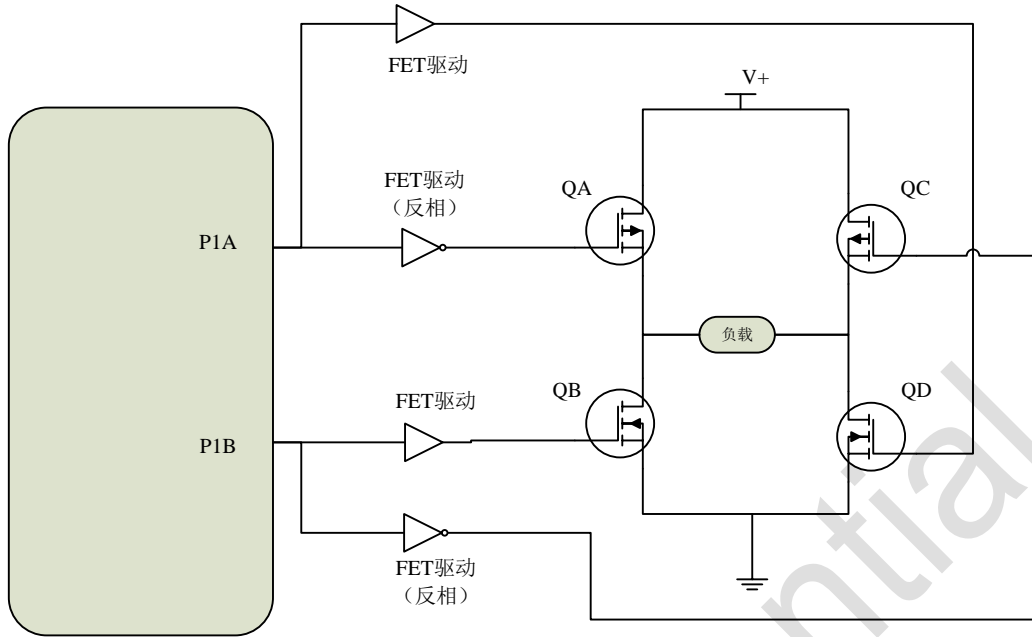


图 9.9c 半桥输出驱动全桥电路 (2PMOS+2NMOS)

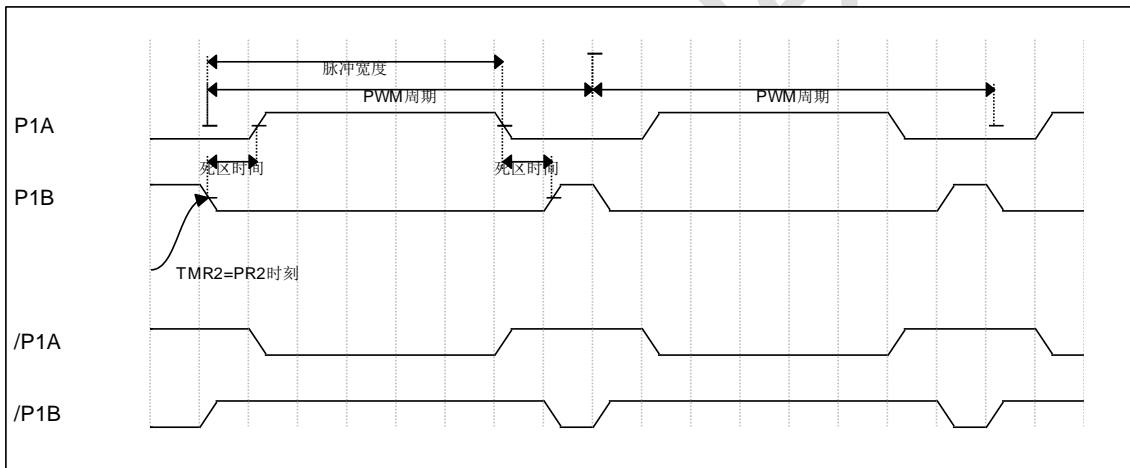


图 9.9d 半桥 PWM 输出 (需要外接反相器)

### 9.4.1.1 关闭 PWM 输出

如果要停止 PWM 输出，建议通过对写 ECCPAS 寄存器相关值，或者把相关 IO 的输出驱动关闭 (TRISC.x 置 1)，通过外部的上下拉电阻 IO 处于确定状态，而不是简单把 CCP1CON 的 CCP1M 清 0，因为此时 IO 还处于输出状态，其输出值不确定。

下面是关闭和重启 PWM 的程序示例：

```

;关闭 PWM 输出
BANKSEL ECCPAS
LDWI 0F0H
IORWR ECCPAS, F
;重开 PWM 输出
    
```

BANKSEL ECCPAS  
LDWI 00FH  
ANDWR ECCPAS, F

## 9.4.2 全桥模式

在全桥模式下，所有四个引脚均用作输出。

图 9.10a 所示为一个全桥应用示例，使用 4 个 NMOS，这时 CCP1CON 的 CCP1M 需要设置为 1100，即 P1A~P1D 都是高有效。

在正向模式下，P1A 引脚驱动为有效状态，P1D 引脚为调制输出，而 P1B 和 P1C 则被驱动为无效状态，如图 9.11a 所示。

在反向模式下，P1C 驱动为有效状态，P1B 引脚为调制输出，而 P1A 和 P1D 则被驱动为无效状态，如图 9.11b 所示。

图 9.10b 是另外的一个全桥应用，使用 2 个 PMOS 和 2 个 NMOS，这时 CCP1CON 寄存器的 CCP1M 需要设置为 1110，即 P1A 和 P1C 为低电平有效，P1B 和 P1D 为高电平有效。图 9.11c 是该应用下的 PWM 输出波形。

P1A、P1B、P1C 和 P1D 输出与 PORT 数据锁存器复用。必须清零相关 TRIS 位以将 P1A、P1B、P1C 和 P1D 引脚配置为输出。

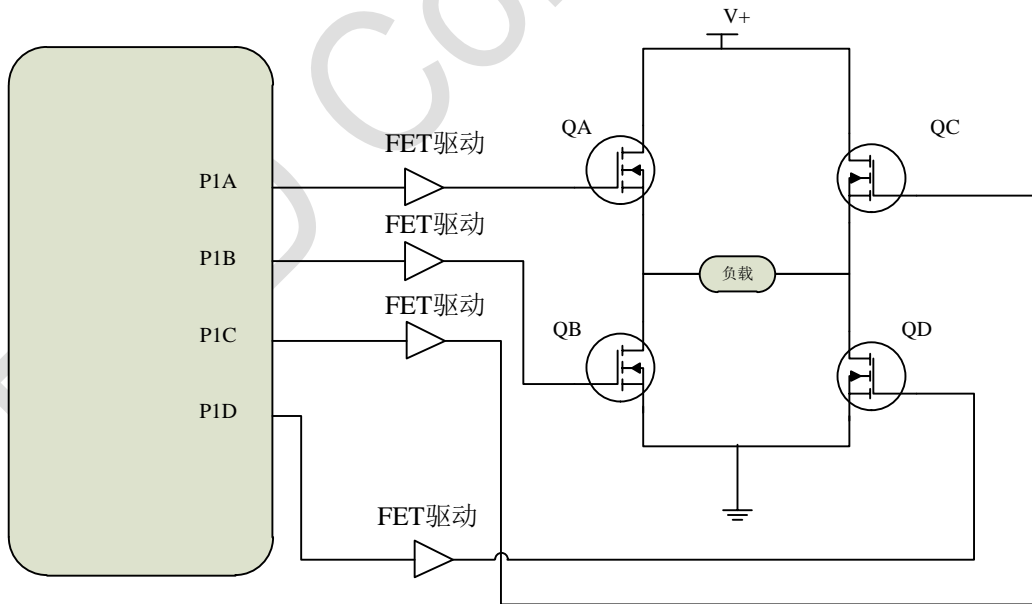


图 9.10a 全桥应用示例 a

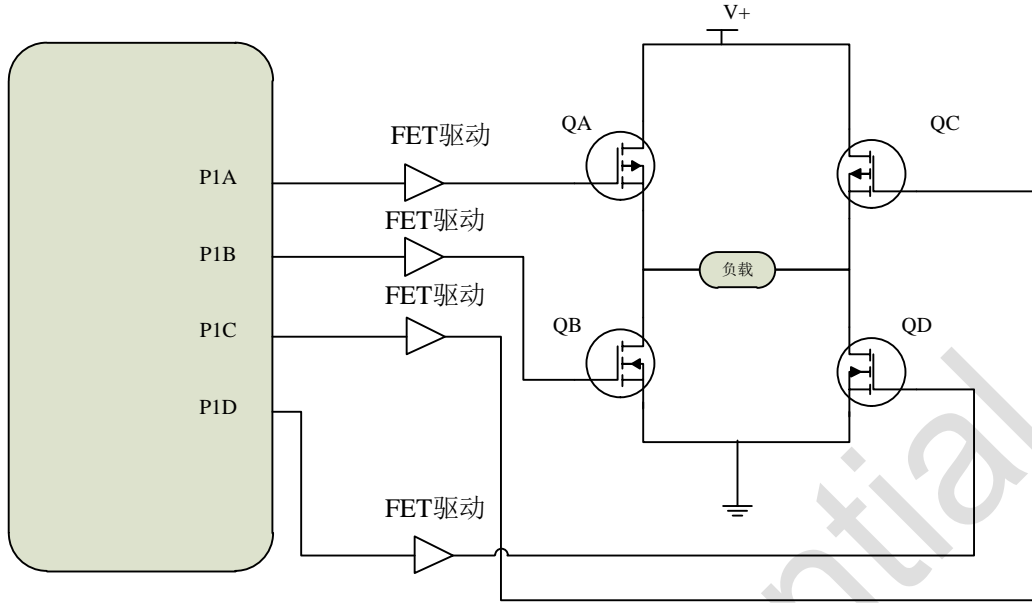


图 9.10b 全桥应用示例 b

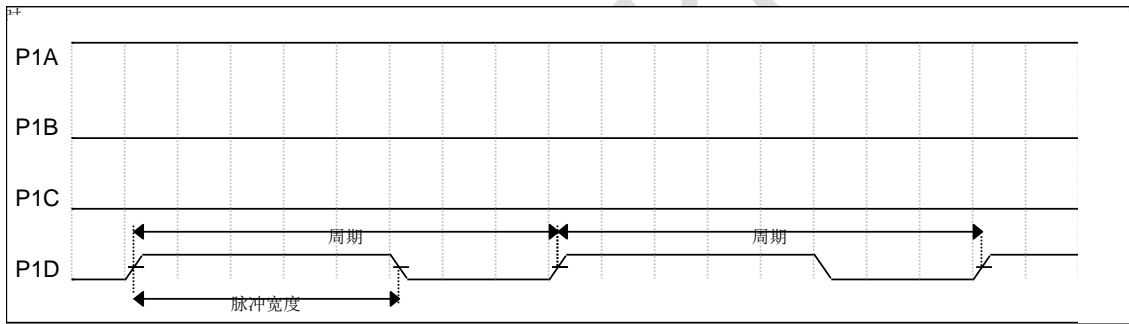


图 9.11a 全桥 PWM 输出示例 a

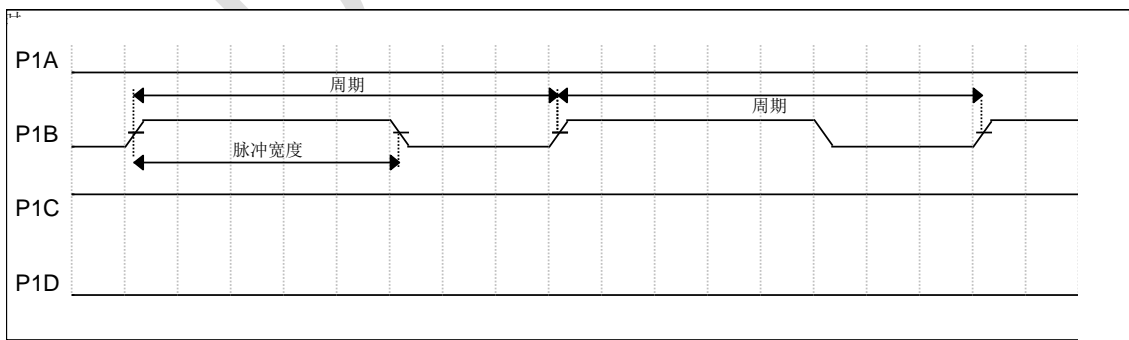


图 9.11b 全桥 PWM 输出示例 b

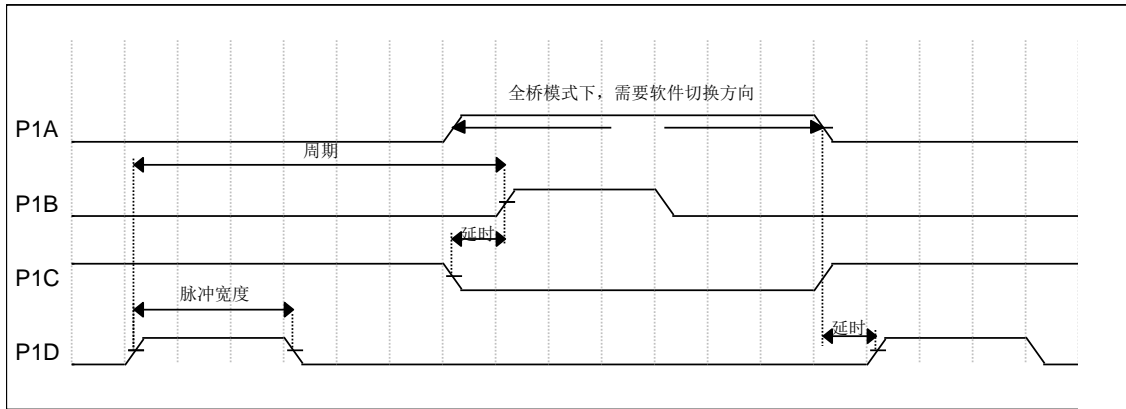


图 9.11c 全桥 PWM 输出示例 c

### 9.4.2.1 关闭 PWM 输出

如果要停止 PWM 输出，建议通过对写 ECCPAS 寄存器相关值，或者把相关 IO 的输出驱动关闭（TRISC.x 置 1），通过外部的上下拉电阻 IO 处于确定状态，而不是简单把 CCP1CON 的 CCP1M 清 0，因为此时 IO 还处于输出状态，其输出值不确定。

下面是关闭和重启 PWM 的程序示例：

```

;关闭 PWM 输出
BANKSEL ECCPAS
LDWI 0F0H
IORWR ECCPAS, F
;重开 PWM 输出
BANKSEL ECCPAS
LDWI 00FH
ANDWR ECCPAS, F
    
```

### 9.4.2.2 在全桥模式下改变方向

在全桥模式下，CCP1CON 寄存器的 P1M1 位可供用户控制正/反方向。当应用固件改变方向控制位时，模块将在下一个 PWM 周期改变方向。

用软件改变 CCP1CON 寄存器的 P1M1 位可启动方向改变。在当前 PWM 周期前的四个 Timer2 周期，发生以下时序：

- 调制输出（P1B 和 P1D）被置于无效状态
- 相关的未调制输出（P1A 和 P1C）切换为相反方向驱动
- 在下一个周期恢复 PWM 调制

此时序的图示请参见图 9.12。

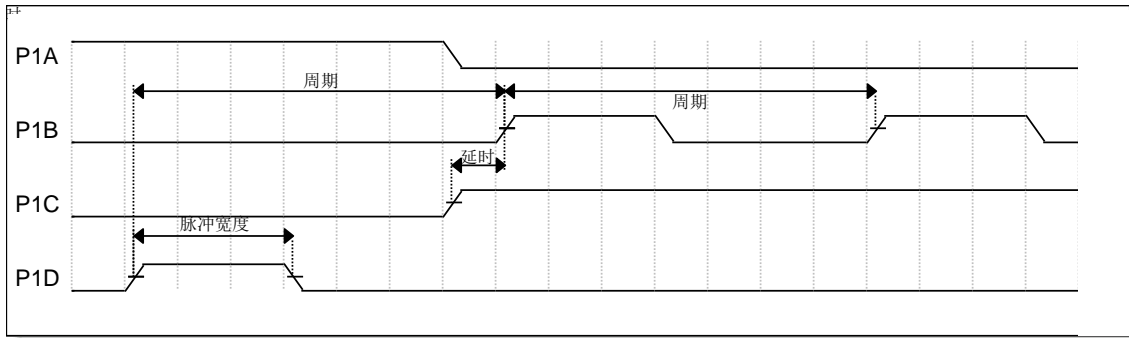


图 9.12 PWM 方向改变示例

注意：

1. **CCP1CON** 寄存器的方向位 **P1M1** 可在 **PWM** 周期的任何时刻被写入；
2. 改变方向时，**P1A** 和 **P1C** 信号在当前 **PWM** 周期结束前切换。此时 **P1B** 和 **P1D** 调制信号无效。此时长为 4 次 **Timer2** 计数。

全桥模式不提供死区延时。在调制一个输出时，一般不需要死区延时。但有一种情况需要死区延时，当以下两个条件同时成立时即发生需要死区延时的情况：

1. 输出占空比达到或接近 100%时 **PWM** 输出方向改变；
2. 功率开关（包括功率器件和驱动器电路）的关断时间大于导通时间。

图 9.13 所示为点空比接近 100%时，**PWM** 方向从正向变为反向的示例。此示例中，在时间  $t_1$  处，**P1A** 和 **P1D** 输出变为无效，而 **P1C** 输出变为有效。由于功率器件的关断时间大于导通时间，穿通电流将流过功率器件 **QC** 和 **QD**（见图 9.10）并持续时间“ $T$ ”。当 **PWM** 方向由反向变为正向时，同样的情况将发生在功率器件 **QA** 和 **QB** 上。

如果某个应用要求在占空比很高时改变 **PWM** 方向，以下提供了两种消除穿通电流的方法：

1. 改变方向前将减小 **PWM** 占空比；
2. 使用能使开关的关断时间快于导通时间的开关驱动器。



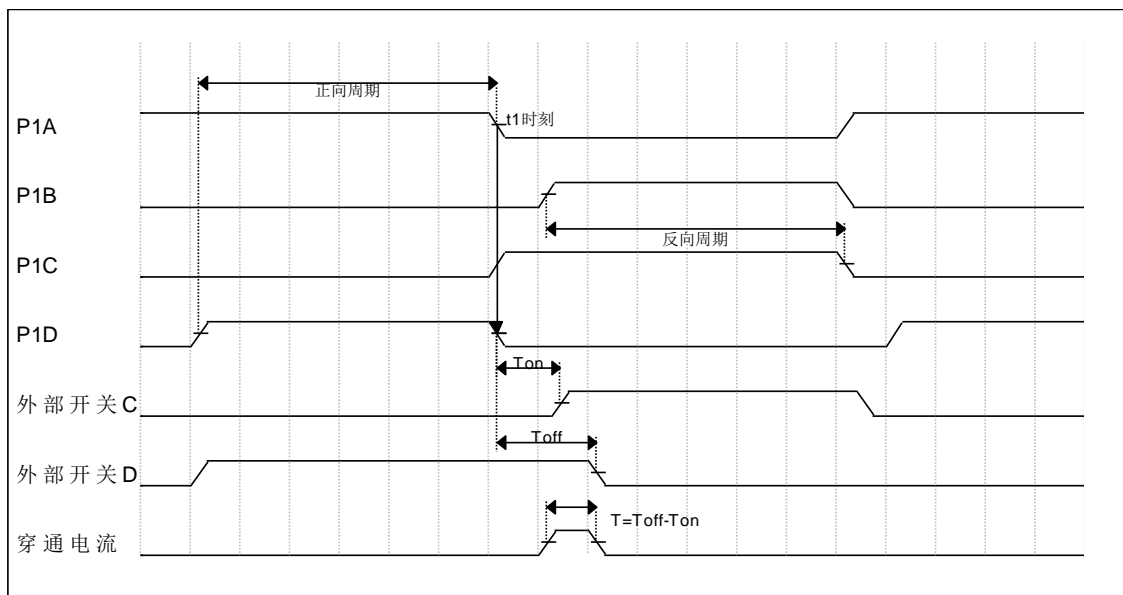


图 9.13 占空比接近 100%时 PWM 方向变化示例

注意:

1. 上图所有信号均为高电平有效;
2.  $T_{on}$  为功率开关 QC 及其驱动器的导通延时;
3.  $T_{off}$  为功率开关 QD 及其驱动器的判断延时。

### 9.4.3 启动考虑事项

使用任何 PWM 模式时，应用硬件中必须在 PWM 输出引脚上使用适当的外部上拉/下拉电阻。

注意：单片机从复位退出时，所有 I/O 引脚均为高阻态。在单片机以正确的信号电平驱动 I/O 引脚或激活 PWM 输出前，外部电路必须使功率开关置于关断状态。

CCP1CON 寄存器的 CCP1M<1:0>位可供用户选择每对 PWM 输出引脚（P1A/P1C 和 P1B/P1D）的输出信号为高电平有效还是低电平有效。PWM 输出极性必须在使能 PWM 引脚的输出驱动器前选定。不建议在 PWM 引脚的输出驱动器使能时改变极性配置，因为这可能会损坏应用电路。

在 PWM 模块初始化时，P1A、P1B、P1C 和 P1D 输出锁存器可能不在正确的状态下。将 PWM 引脚的输出驱动器与增强型 PWM 模式同时使能可能导致应用电路的损坏。增强型 PWM 模式必须在正确的输出模式下使能、并且在 PWM 引脚的输出驱动器被使能前完成一整个 PWM 周期。一整个 PWM 周期是否完成可通过查看 PIR1 寄存器的 TMR2IF 位在第二个 PWM 周期开始时是否置 1 查看。

### 9.4.4 增强型 PWM 自动关闭模式

PWM 模块支持自动关闭模式，它会在发生外部关闭事件时禁止 PWM 输出。自动关闭模式会将 PWM 输出引脚置于预定状态。该模块用于防止 PWM 损坏应用。

使用 ECCPAS 寄存器的 ECCPASx 位可选择自动关闭源。关闭事件可由以下产生：

- INT 引脚出现逻辑 0
- 比较器 C1
- 比较器 C2
- 在固件中将 ECCPASE 位置 1

关闭状态由 ECCPAS 寄存器的 ECCPASE（自动关闭事件状态）位指示。如果该位为 0，PWM 引脚正常工作。如果该位为 1，PWM 输出处于关闭状态。

发生关闭事件时，将出现两个状况：

1. ECCPASE 位置 1。ECCPASE 保持置 1 状态直到被固件清零或发生了自动重启（见第 9.4.5 节“自动重启模式”）；
2. 使能的 PWM 引脚被异步置于其关闭状态。PWM 输出引脚被分为两对[P1A/P1C]和[P1B/P1D]。两对引脚的状态由 ECCPAS 寄存器的 PSSAC 和 PSSBD 位决定。每对引脚均可置于以下三种状态之一：
  - 驱动为逻辑 1
  - 驱动为逻辑 0
  - 三态（高阻态）

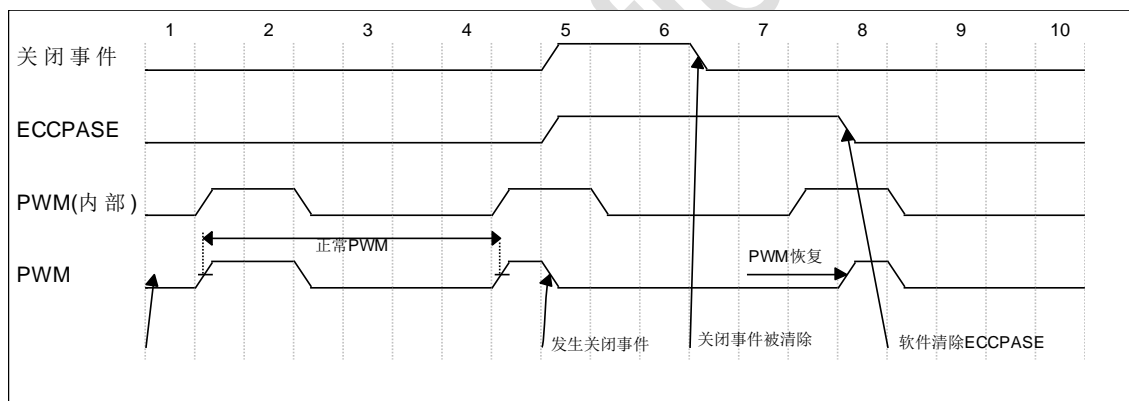


图 9.14 非自动重启时的 PWM 自动关闭

注意：

1. 自动关闭条件是基于电平的信号，而非基于边沿的信号。只要电平不变，自动关闭就不变；
2. 自动关闭条件下禁止写入 ECCPASE 位；
3. 自动关闭条件有效时，PWM 停止输出但 PWM 计数器还在继续运行，所以当自动关闭条件清除时（通过固件或自动重启），PWM 将立即恢复输出。

## 9.4.5 自动重启模式

增强型 PWM 可配置为在自动关闭条件被清除时自动重启 PWM 信号。通过将 PWM1CON 寄存器中的 PRSEN 位置 1 可使能自动重启。

使能自动重启时，只要自动关闭条件有效，ECCPASE 位就保持置 1。当自动关闭条件被清除时，

ECCPASE 位将被硬件清零，恢复正常工作。

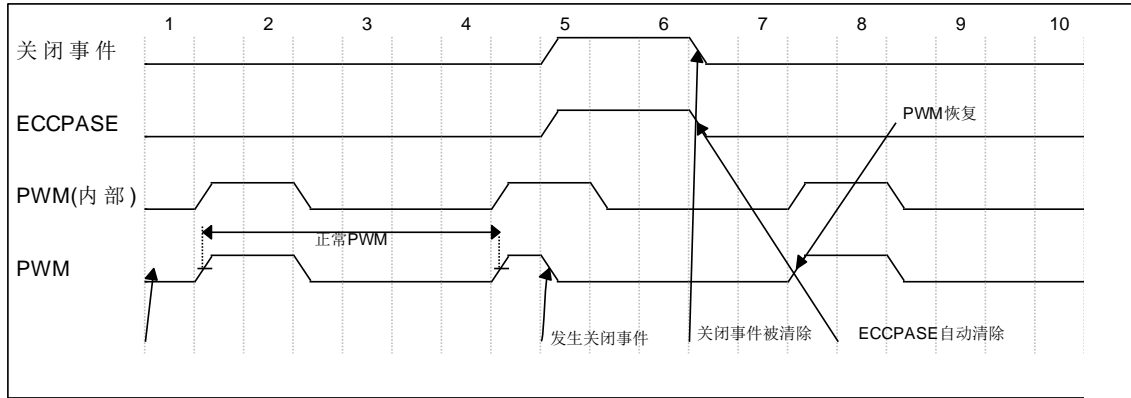


图 9.15 自动重启时的 PWM 自动关闭

### 9.4.6 可编程死区延时模式

在所有功率开关均调制为 PWM 频率的半桥应用中，功率开关从关断到导通通常需要较长的时间。如果上下两个功率开关同时动作（一个导通另一个关断），在一个开关完全关断前，两个开关可能在一个很短的时间内同时导通。在这段很短的时间内，在两个功率开关中会流过极高的电流（穿通（shoot-through）电流），使桥路的电源短路。为避免在开关时出现这种极具破坏力的穿通电流，通常使任一功率开关的导通时间延后，以使另一个开关有时间完全关断。

在半桥模式下，使用数字可编程死区延时来避免穿通电流破坏桥路的功率开关。信号从无效状态变为有效状态时发生延时，如图 9.16 中的图解说明。相关 PWM1CON 寄存器的低 7 位以单片机的指令周期（TCY，即 2 个系统时钟周期）为单位设置延时期限。

注意：

1. 由于死区计数器是以指令时钟作为时钟源，而睡眠状态下指令时钟被关闭，所以即使 PCON 寄存器的 SYSON 位为 1，使能了死区延时的 ECCP 模块的 PWM 输出在 SLEEP 状态下将不能工作。

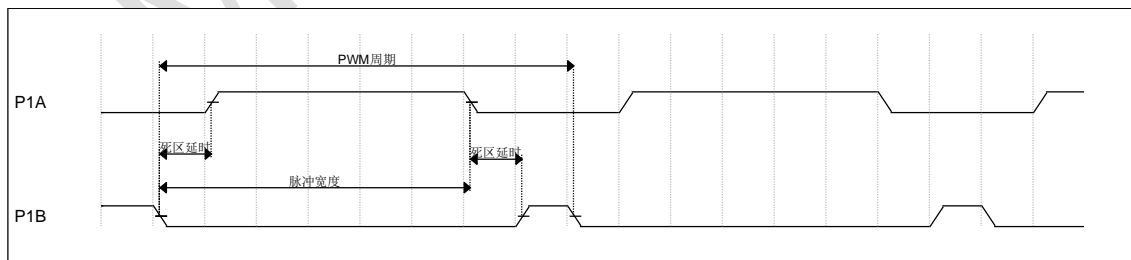


图 9.16 半桥 PWM 输出示例

## 9.5 PWM 的辅助功能

通过适当设置寄存器 PWM1AUX，工作在半桥模式下的增强型 PWM 可以做到：

- 输出一次 PWM 信号后自动关闭 PWM 输出
- 最多有 3 对 6 路 PWM 信号同时输出（当 P1xOE 全部为 1 时，这里 x 是 A~F）
- 输出极性可配置（通过 CCP1M[3:0]）

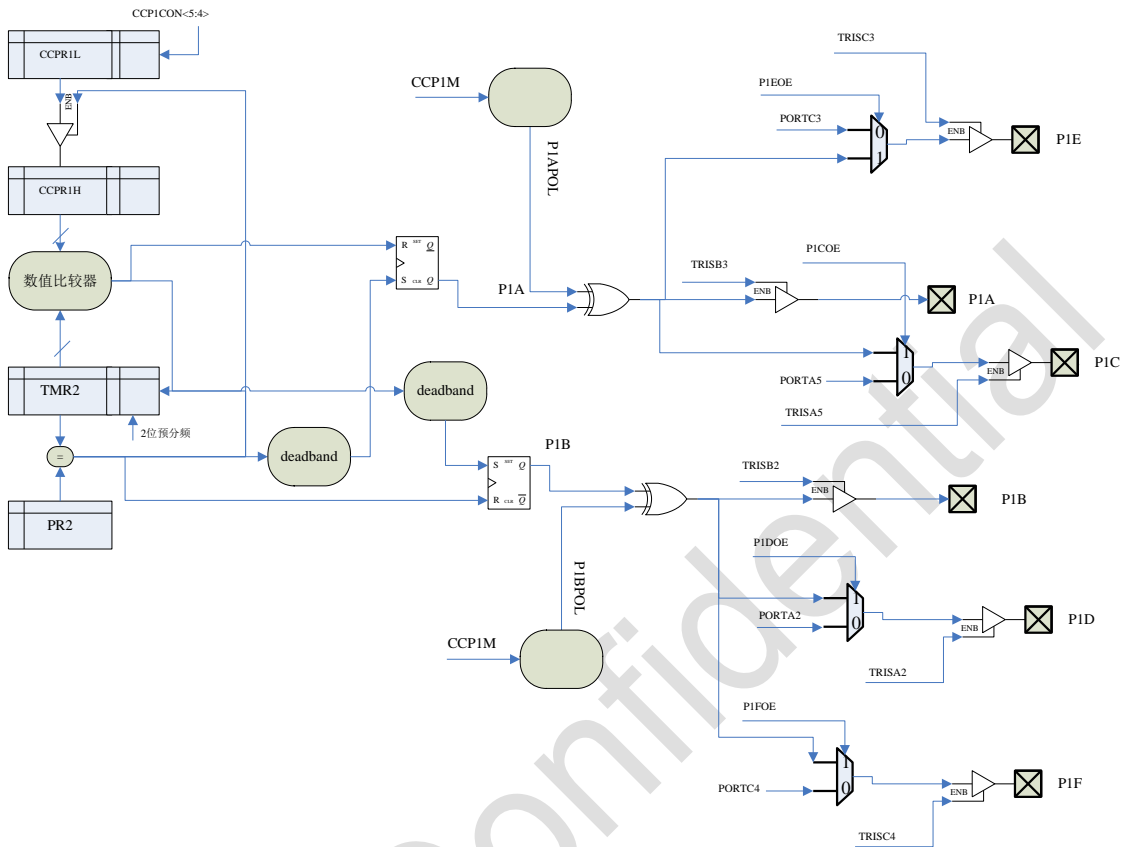


图 9.17 PWM 的辅助功能原理框图

注意：

1. PWM 的辅助功能只对半桥模式起作用，其它的单输出或全桥模式不起作用，即  $P1M<1:0>=10$ 。

### 9.5.1 一次性脉冲模式

配置 CCP1CON 使 ECCP 处于 PWM 半桥模式，同时把 PWM1AUX 的 AUX1EN 置 1 和 P1OS 置 1，此时 PWM 为一次性脉冲模式。

当下一个周期 PWM 到来时（TIMER2 等于 PR2+1），PWM 输出由硬件自动关闭，P1A~P1F 变为通用 IO。

需要注意的是，在该模式下，PWM 输出一次脉冲波形后只是把 P1xOE 清 0，里面的 PWM 计数器将保持计数，如果软件再次把 P1xOE（x 可以是 A~F）置 1，则在下一个 PWM 周期管脚 P1x（x 可以是 A~F）会输出一个 PWM 波形，如下图所示。

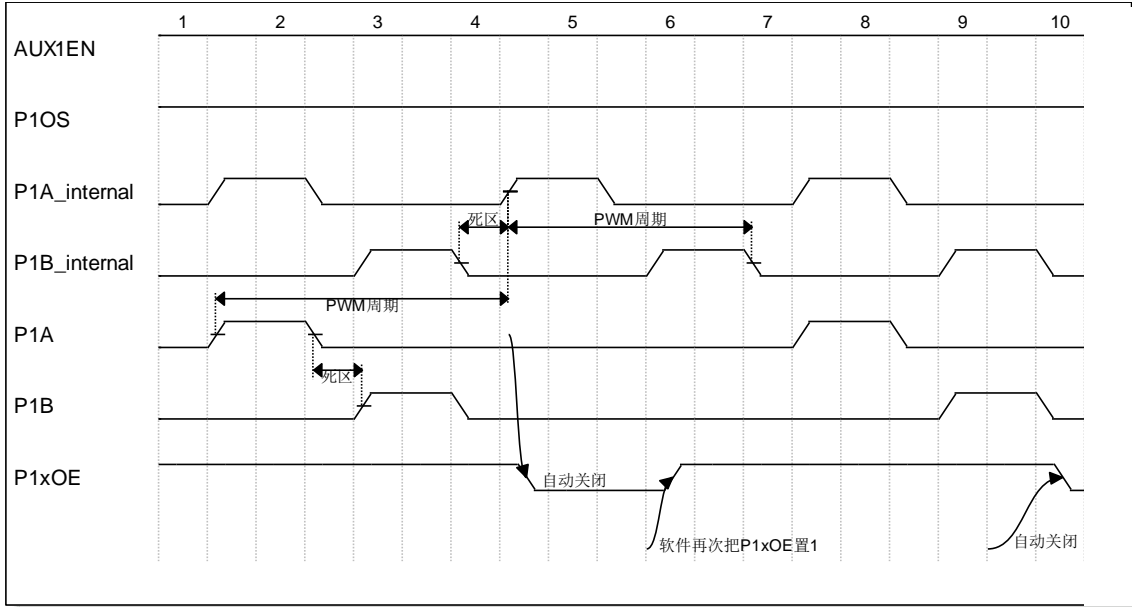


图 9.18 PWM 的辅助功能原理框图

### 9.5.2 3 对 PWM 信号输出

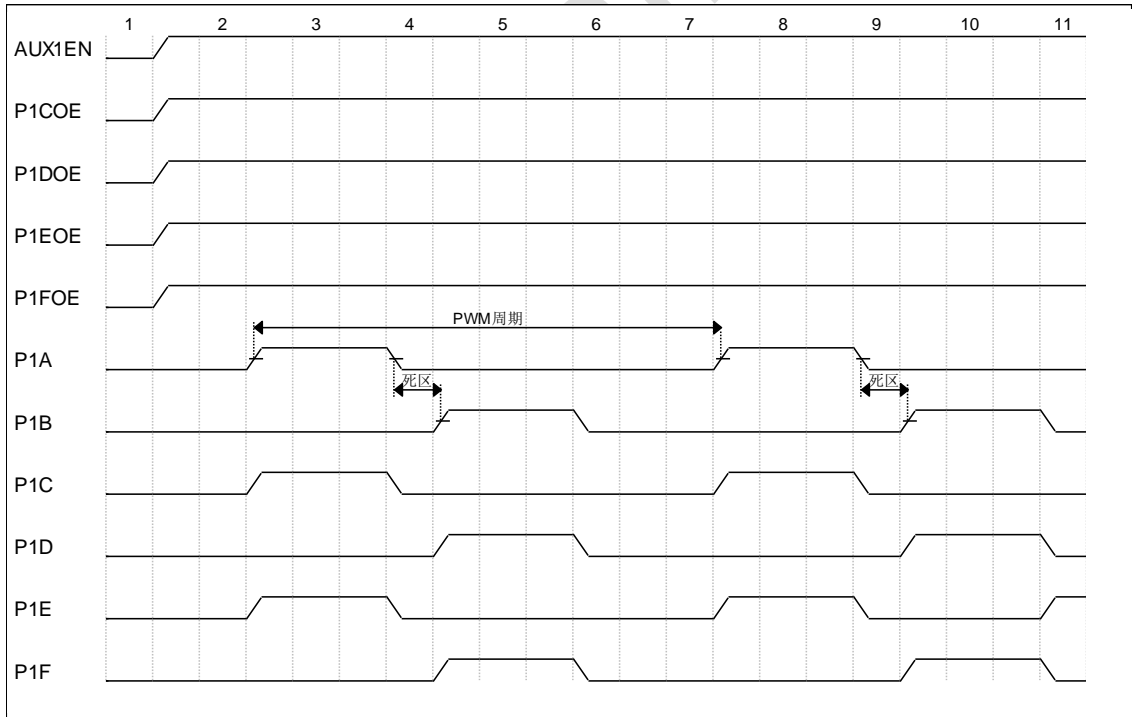


图 9.18 PWM 的辅助功能原理框图

注意：如上图所示，P1A 和 P1B 是第一组带死区时间的半桥 PWM 输出，P1C 和 P1D，P1E 和 P1F 是第二和第三组，它们的波形和第一组是一样的。

### 9.5.3 PWM 辅助功能的使用

应按照以下步骤将 CCP 模块配置为 PWM 工作：

1. 将相关的 TRIS 位置 1 禁止 PWM 引脚（CCP1）的输出驱动器；
2. 装载 PR2 寄存器以设置 PWM 周期；
3. 用适当的值装载 CCP1CON 寄存器将 CCP 模块配置为 PWM 半桥模式；
4. 装载 CCPR1L 寄存器和 CCP1CON 寄存器的 DC1B<1:0>设置 PWM 占空比；
5. 配置并启动 Timer2:
  - 将 PIR1 寄存器的 TMR2IF 中断标志位清零
  - 装载 T2CON 寄存器的 T2CKPS 位设置 Timer2 预分频比
  - 将 T2CON 寄存器的 TMR2ON 位置 1 使能 Timer2
  - 如果要设置高速模式，则需要把 MSCKCON.5 置 1
6. 重新开始一个 PWM 周期后，使能 PWM 输出：
  - 等待 Timer2 溢出（PIR1 寄存器的 TMR2IF 位置 1）
  - 设置 PWM1AUX，把 AUX1EN 位置 1，其它各位根据应用需要设置
  - 将相关的 TRIS 位清零使能 CCP1 引脚的输出驱动器

### 9.6 与 ECCP 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值	
TMR2	0x11	TIMER2 计数值寄存器								xxxx xxxx	
PR2	0x92	TIMER2 周期寄存器								1111 1111	
INTCON	0x0B/8B	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000	
PIE1	0x8C	EEIE	CKMEAIE	-	C5IE	C4IE	OSFIE	TMR2IE	TMR1IE	00-0 0000	
PIR1	0x0C	EEIF	CKMEAIF	-	C5IF	C4IF	OSFIF	TMR2IF	TMR1IF	00-0 0000	
PIE2	0x8D	-	-	C3IE	C2IE	C1IE	C0IE	ADIE	CCP1IE	--00 0000	
PIR2	0x0D	-	-	C3IF	C2IF	C1IF	C0IF	ADIF	CCP1IF	--00 0000	
PWM1AUX	0x90	AUX1EN	P1OS	P1FOE	P1EOE	P1DOE	P1COE	P1BOE	P1AOE	0000 0000	
T1CON	0x10	T1GINV	TMR1GE	T1CKPS[1:0]		T1OSCN	T1SYNC	TMR1CS	TMR1ON	0000 0000	
T2CON	0x12	NRPM	TOUTPS[3:0]			TMR2ON	T2CKPS[1:0]			0000 0000	
TMR1L	0x0E	16 位 TIMER1 计数值低 8 位								xxxx xxxx	
TMR1H	0x0F	16 位 TIMER1 计数值高 8 位								xxxx xxxx	
TRISA	0x85	TRISA[7:0], PORTA 方向控制								1111 1111	
TRISC	0x87	-	-	TRISC[5:0], PORTC 方向控制						--11 1111	
CCPR1L	0x13	ECCP1 寄存低 8 位								xxxx xxxx	
CCPR1H	0x14	ECCP1 寄存高 8 位								xxxx xxxx	
CCP1CON	0x15	P1M[1:0]		DC1B[1:0]		CCP1M[3:0]				0000 0000	
PWM1CON	0x16	PRSEN	PDC[6:0]								0000 0000
ECCPAS	0x17	ECCPASE	ECCPAS[2:0]			PSSAC[1:0]		PSSBD[1:0]		0000 0000	

### 9.6.1 CCPR1L 寄存器，地址 0x13

Bit	7	6	5	4	3	2	1	0
Name	CCPR1L							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	CCPR1L[7:0]	捕捉、比较、PWM 寄存器 1 的低字节

### 9.6.2 CCPR1H 寄存器，地址 0x14

Bit	7	6	5	4	3	2	1	0
Name	CCPR1H							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	CCPR1H[7:0]	捕捉、比较、PWM 寄存器 1 的高字节

### 9.6.3 CCP1CON 寄存器，地址 0x15

Bit	7	6	5	4	3	2	1	0
Name	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	P1M[1:0]	PWM 输出配置位 如果 CCP1M<3:2> = 00、01 或 10: xx =P1A 配置为捕捉/比较输入；P1B、P1C 和 P1D 配置为端口引脚 如果 CCP1M<3:2> = 11: 00 = 单输出；P1A 调制；P1B、P1C 和 P1D 配置为端口引脚 01 = 全桥正向输出；P1D 调制；P1A 有效；P1B 和 P1C 无效 10 = 半桥输出；P1A 和 P1B 调制，带有死区控制；P1C 和 P1D 配置为端口引脚 11 = 全桥反向输出；P1B 调制；P1C 有效；P1A 和 P1D 无效
5:4	DC1B[1:0]	PWM 占空比最低有效位 捕捉模式：

		未使用 比较模式： 未使用 PWM 模式： 这些位是 PWM 占空比的低 2 位。高 8 位在 CCPR1L 中
3:0	CCP1M[3:0]	<p>ECCP 模式选择位</p> <p>0000 = 捕捉/比较/PWM 关闭 (复位 ECCP 模块)</p> <p>0001 = 未使用 (保留)</p> <p>0010 = 比较模式, 匹配时翻转输出 (CCP1IF 位置 1)</p> <p>0011 = 未使用 (保留)</p> <p>0100 = 捕捉模式, 每个下降沿 (CCP1IF 位置 1)</p> <p>0101 = 捕捉模式, 每个上升沿 (CCP1IF 位置 1)</p> <p>0110 = 捕捉模式, 每 4 个上升沿 (CCP1IF 位置 1)</p> <p>0111 = 捕捉模式, 每 16 个上升沿 (CCP1IF 位置 1)</p> <p>1000 = 比较模式, 匹配时输出置 1 (CCP1IF 位置 1)</p> <p>1001 = 比较模式, 匹配时输出清零 (CCP1IF 位置 1)</p> <p>1010 = 比较模式, 匹配时产生软件中断 (CCP1IF 位置 1, CCP1 引脚不受影响)</p> <p>1011 = 比较模式, 触发特殊事件 (CCP1IF 位置 1, CCP1 复位 TMR1, 且如果 A/D 模块被使能, 启动一次 A/D 转换)</p> <p>1100 = PWM 模式; P1A 和 P1C 高电平有效; P1B 和 P1D 高电平有效</p> <p>1101 = PWM 模式; P1A 和 P1C 高电平有效; P1B 和 P1D 低电平有效</p> <p>1110 = PWM 模式; P1A 和 P1C 低电平有效; P1B 和 P1D 高电平有效</p> <p>1111 = PWM 模式; P1A 和 P1C 低电平有效; P1B 和 P1D 低电平有效</p>

### 9.6.4 PWM1CON 寄存器, 地址 0x16

Bit	7	6	5	4	3	2	1	0
Name	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	PRSEN	<p>PWM 重启使能位</p> <p>1 = 自动关闭时, ECCPASE 位在退出关闭事件时自动清零, PWM 自动重启</p> <p>0 = 自动关闭时, 必须用软件将 ECCPASE 清零以重启 PWM</p>
6:0	PDC[6:0]	<p>PWM 延时计数位</p> <p>PDCn = 预定 PWM 信号应转变为有效与 PWM 信号实际转为有效之间的指令周期数</p>



### 9.6.5 ECCPAS 寄存器，地址 0x17

Bit	7	6	5	4	3	2	1	0
Name	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	ECCPASE	ECCP 自动关闭事件状态位 1 = 发生了自动关闭事件；ECCP 输出处于关闭状态 0 = ECCP 输出正常工作
6:4	ECCPAS[2:0]	ECCP 自动关闭源选择位 000 = 禁止自动关闭 001 = 比较器 4 输出 C4OUT 变高 010 = 比较器 5 输出 C5OUT 变高 011 = 比较器 4 或 5 之一输出变高 100 = INT 引脚电压为 VIL 101 = INT 引脚电压为 VIL 或比较器 4 输出变高 110 = INT 引脚电压为 VIL 或比较器 5 输出变高 111 = INT 引脚电压为 VIL 或比较器 1/2 之一输出变高
3:2	PSSAC[1:0]	P1A 和 P1C 引脚关闭状态控制位 00 = 驱动引脚 P1A 和 P1C 为 0 01 = 驱动引脚 P1A 和 P1C 为 1 1x = P1A 和 P1C 引脚为三态
1:0	PSSBD[1:0]	P1B 和 P1D 引脚关闭状态控制位 00 = 驱动引脚 P1B 和 P1D 为 0 01 = 驱动引脚 P1B 和 P1D 为 1 1x = P1B 和 P1D 引脚为三态

### 9.6.6 PWM1AUX 寄存器，地址 0x90

Bit	7	6	5	4	3	2	1	0
Name	AUX1EN	P1OS	P1FOE	P1EOE	P1DOE	P1COE	P1BOE	P1AOE
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	AUX1EN	PWM1 辅助功能使能位 1 = 使能 PWM1 辅助功能 0 = 禁止 PWM1 辅助功能

6	P1OS	当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时， 1 = PWM 输出一次脉冲后自动停止，P1A/P1B/P1C/P1D/P1E/P1F 变为普通 IO 0 = PWM 连续输出 当 AUX1EN=0 时，该位无意义
5	P1FOE	当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时， 1 = 引脚 P1F 变为 PWM 输出 0 = 引脚 P1F 变为 IO 当 AUX1EN=0 时，该位无意义
4	P1EOE	当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时， 1 = 引脚 P1E 变为 PWM 输出 0 = 引脚 P1E 变为 IO 当 AUX1EN=0 时，该位无意义
3	P1DOE	当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时， 1 = 引脚 P1D 变为 PWM 输出 0 = 引脚 P1D 变为 IO 当 AUX1EN=0 时，该位无意义
2	P1COE	当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时， 1 = 引脚 P1C 变为 PWM 输出 0 = 引脚 P1C 变为 IO 当 AUX1EN=0 时，该位无意义
1	P1BOE	当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时， 1 = 引脚 P1B 变为 PWM 输出 0 = 引脚 P1B 变为 IO 当 AUX1EN=0 时，该位无意义
0	P1AOE	当 AUX1EN=1 且 ECCP 处于半桥 PWM 模式时， 1 = 引脚 P1A 变为 PWM 输出 0 = 引脚 P1A 变为 IO 当 AUX1EN=0 时，该位无意义

**注意：**

1. 如果 PWM 工作在半桥模式且 AUX1EN 和 P1OS 同时为 1 时，P1AOE~P1FOE 这 6 位将在下一个 PWM 周期到来后自动清 0；
2. 当使用 PWM 的辅助功能，即 AUX1EN 为 1 且 ECCP 处于半桥模式时，P1A~P1F 是否输出 PWM 波形还要取决于 P1AOE~P1FOE 的值，只有当它们为 1 时才作为 PWM 功能管脚，否则是 GPIO。

## 10 比较器模块

片内集成 6 个模拟比较器，每个比较器可以用来比较 2 个模拟电压值而产生数字结果反映模拟电压值的高低关系。其中比较器 0/1/2/3 跟 PPG 紧耦合，而比较器 4/5 独立于 PPG。

比较器具有以下特性：

- 比较结果可输出到片外（CMP4 和 CMP5）
- 输出极性可编程
- 输出可作为中断源
- 输出可作为唤醒源
- 输出带去抖动功能（CMP0/1/2/3）
- 输入可挂内部参考电压
- 多种比较器输入输出连接设定模式
- 可编程参考电压
- 输入失调电压可校准（CMP0/1/2/3）

### 10.1 比较器 0/1/2/3

#### 10.1.1 比较器概述

模拟比较器的输入输出信号关系以及其在电路图中的标志如图 10.1 所示。当比较器的模拟输入电压  $V_{in+}$  小于其模拟输入电压  $V_{in-}$  时，输出为逻辑低电平，反之则输出逻辑高电平。

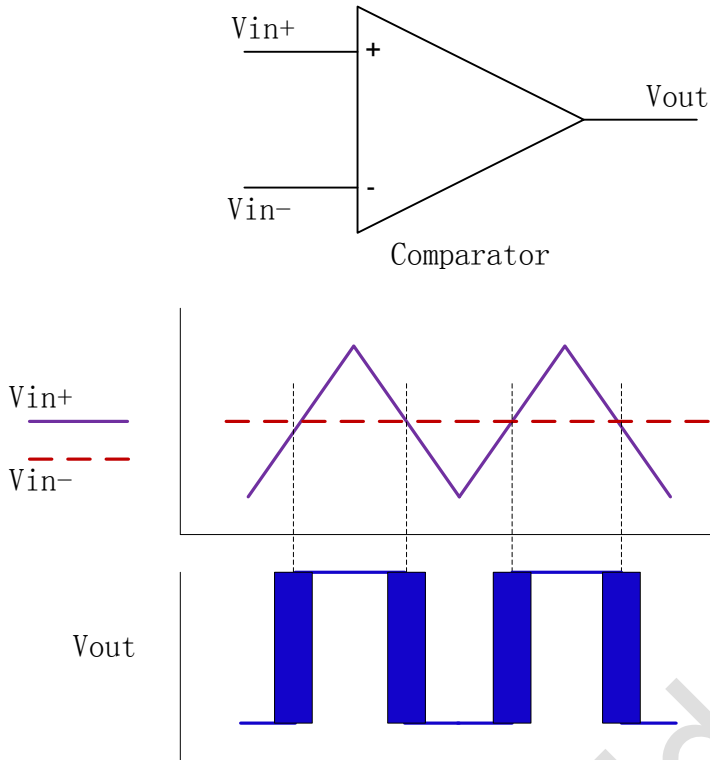


图 10.1 比较器的输入，输出信号特性

当比较器的输入和输出接近相等时，其输出为不定态，因此，为了避免这种不定态，或在某个输入端有微弱噪声存在，造成比较输入出现不稳定态，一般比较器的设计会保护一个所谓迟滞电压，是比较器的输出态变化会延后输入变化一定的距离，这样基本比较电平稍有波动也不会造成杂散输出信号。

### 10.1.2 比较器的配置及结构框图

模拟比较器 0/1/2/3 的使能控制由 SFR 寄存器 CMCON4 低 4 位 COMPON[3:0] 决定，上电复位后这几位为 0，比较器处于关闭状态，使用比较器之前需要把相关位置 1。

由于和数字 IO 复用管脚的关系，使用比较器之前还需要正确配置 IO 的数模、输入输出。下面是各个比较器的结构框图，可以看出，不是每个比较器的输入端都接到了外部管脚上，如比较器 3 的同相输入端固定接内部参考源 VR4，而反相输入端可以连运放输出、PB0 或者 PB1，更详细的管脚配置见 10.1.3 小节。

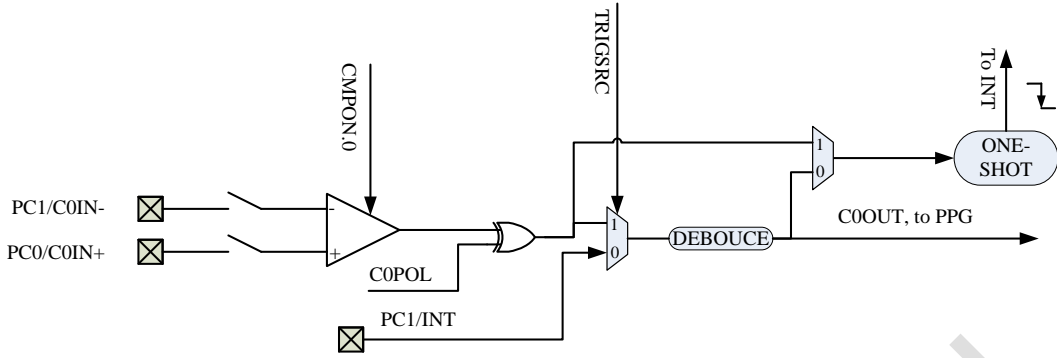


图 10.2 比较器 0 结构框图

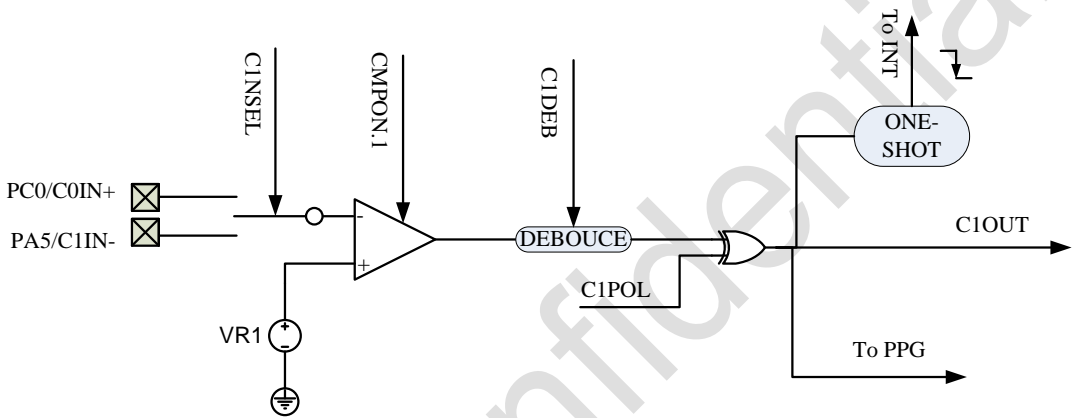


图 10.2 比较器 1 结构框图

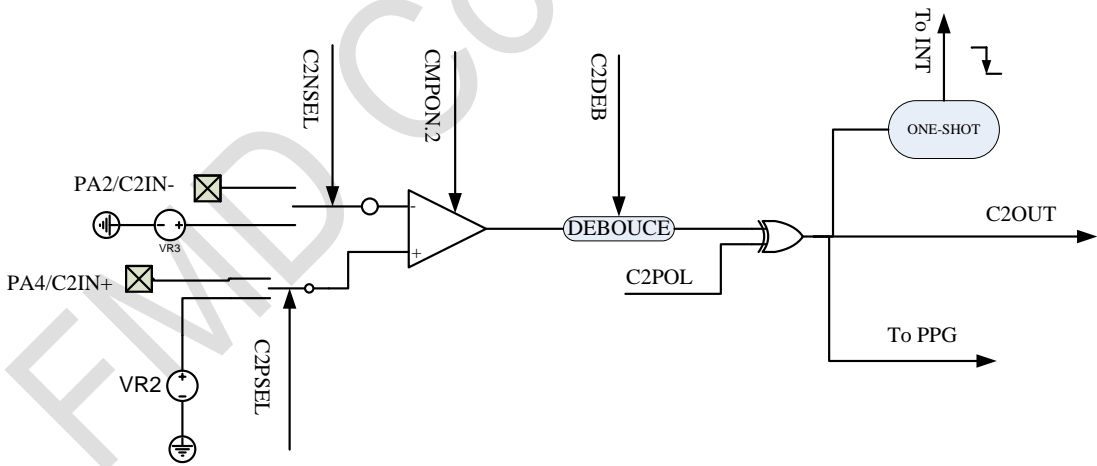


图 10.3 比较器 2 结构框图

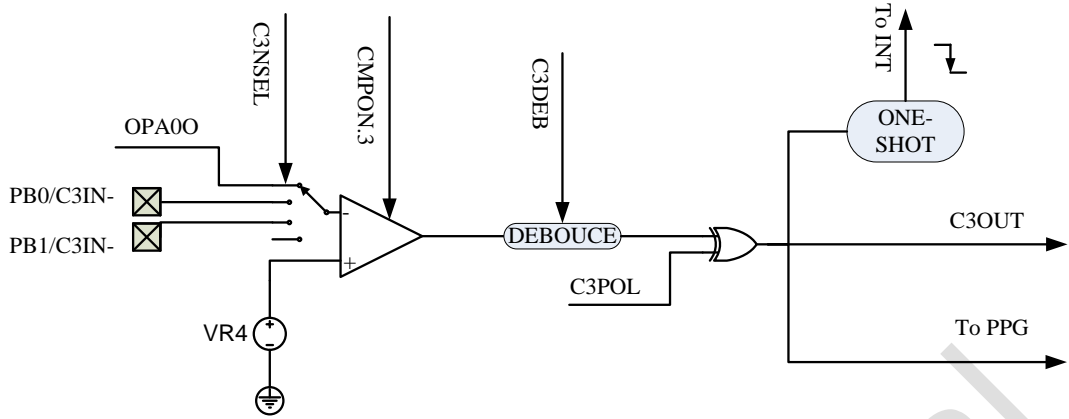


图 10.3 比较器 3 结构框图

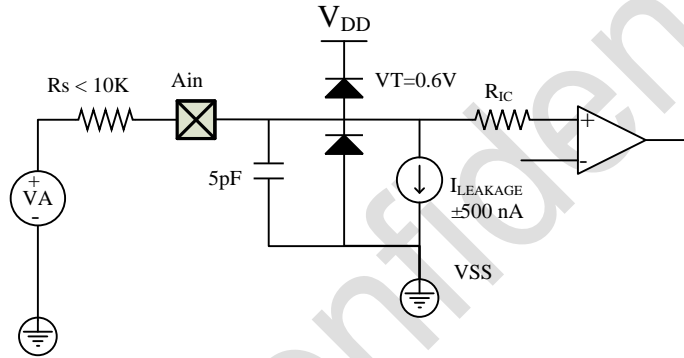


图 10.4 比较器模拟输入模型

### 10.1.3 模拟输入开关

并不是 4 个比较器的输入端都可以接到芯片管脚上, 如比较器 1 和比较器 3 的同相输入就固定接内部参考源。下表是各比较器输入端连接关系说明:

比较器 \ 输入端	CxIN+同相输入	CxIN-反相输入
CMP0	PC0	PC1
CMP1	VR1	PA5 或者 PC0, 由 C1NSEL 选择
CMP2	PA4 或 VR2, 由 C2PSEL 选择	PA2 或 VR3, 由 C2NSEL 选择
CMP3	VR4	PB0 或 PB1 或 OPA00, 由 C3NSEL 选择

表中的 VR1~VR4 是片内参考。

需要注意的是, 比较器输入是模拟电压, 在使用比较器时, 相关引脚要配置为输入以关闭数字驱动, 即 TRIS 位要置 1。

### 10.1.4 比较器中断

比较器一共有 6 个独立中断源，对应的中断标志位分别是 C0IF/C1IF/C2IF/C3IF/C4IF/C5IF。比较器 0/1/2/3 是下降沿触发中断，即当反相输入电压比同相输入电压高时产生中断事件把 CxIF 置 1。但由于比较器输出极性可以设置，故当 CxPOL 为 1 时，同相端电压比反相端电压高的时候产生中断。

比较器 4/5 是状态变化中断，见 10.2.5 小节。

另外，比较器电路被启动时，其偏置电路需要 1uS 的时间稳定，在此期间，比较器的输出是无效的，软件应该在此期间关闭中断以免产生不必要的误触发。

### 10.1.5 比较器的参考电压

模拟比较器模块一共有 4 个独立的内部参考源，VR1~VR4。比较器 2 的两个输入可以设置为参考源 VR2 和 VR3，比较器 1/3 的同相输入端分别接 VR1 和 VR4，比较器 0 没有连到内部参考。参考电压源有以下特性：

- 8 级电压范围
- 输出电压与 VDD 成比例

参考电压的控制由 CMCON2/CMCON3 寄存器的 VRxC[2:0] 控制，如果比较器关闭则参考源也会被关闭，如图 10.5。

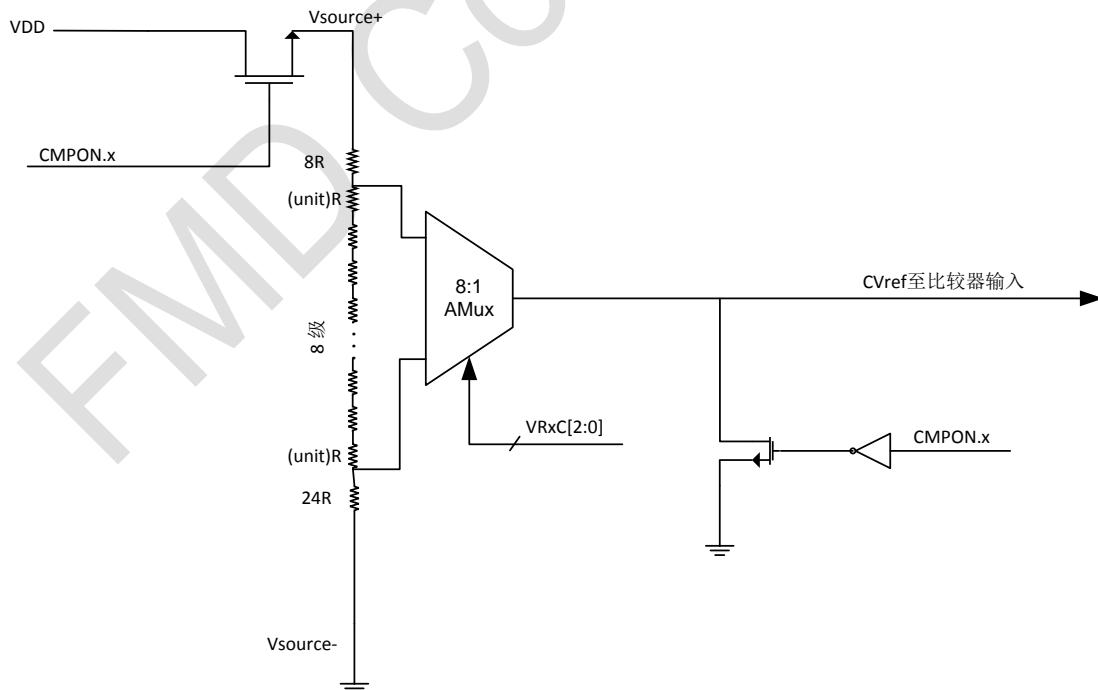


图 10.5 比较器参考电压框图

4 个参考源的计算公式分别如下：

$$V_{VR1}=(0.6 + VR1C/40)V_{DD};$$

$$V_{VR2}=(0.6 + VR2C/40)V_{DD};$$

$$V_{VR3}=(0.075 + VR3C/40)V_{DD};$$

$$V_{VR4}=(0.6 + VR4C/40)V_{DD};$$

### 10.1.6 比较器输出的去抖

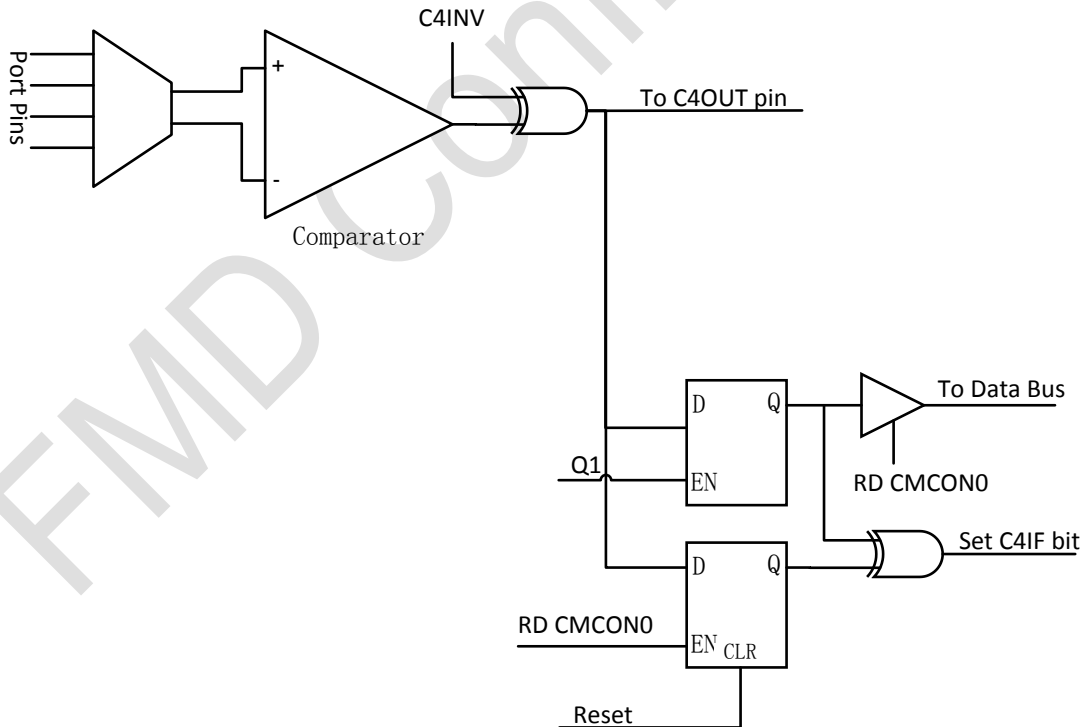
比较器 0/1/2/3 都带有硬件去抖功能，其中比较器 0 的去抖和外部中断管脚 PC1 复用，比较器 1~3 各带独立的去抖模块，通过 CMCON1 的 CxDEB 启用。去抖模块对比较器输出过滤 8 个系统时钟。

比较器 0 的去抖可参考图 14.1。

在使用去抖功能时需要注意去抖时间是跟系统时钟相关的，在 16M 的系统时钟频率下，去抖时间为 0.5us，而在 250K 频率下，去抖时间为 32us。

### 10.2 比较器 4/5

除了比较器 0/1/2/3 之外，芯片还包含了如下图所示的 2 个比较器，但这 2 个比较器的配置不是独立分开的，它们都通过 CMCON0 的 CM[2:0]控制。

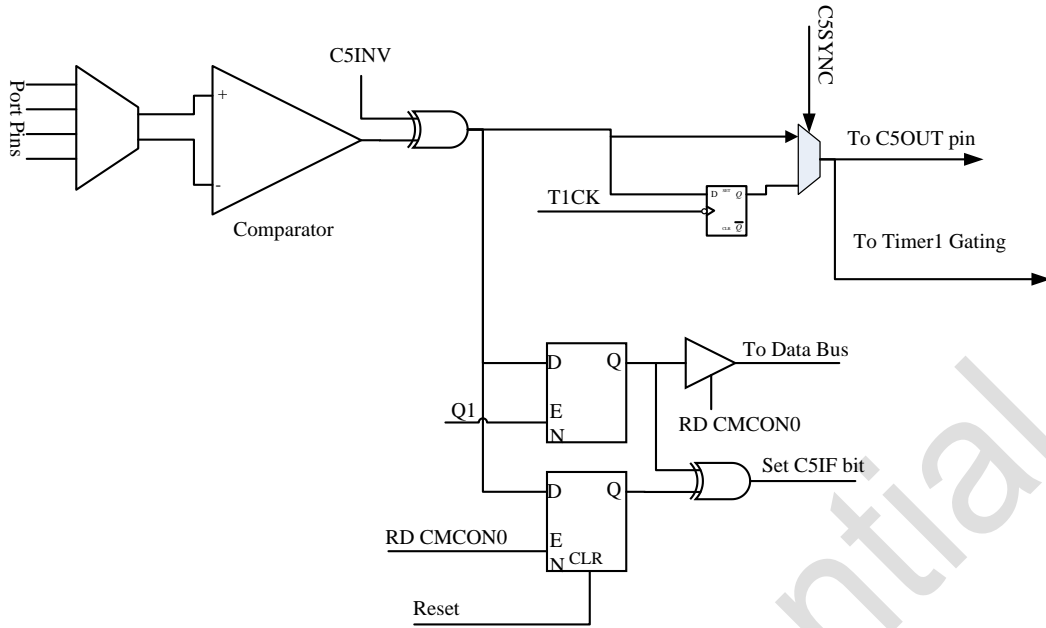


Q1为系统时钟相位。

软件在改变比较器的输入，输出控制寄存器时最好关闭比较器，避免比较器结果错误。

图 10.5 比较器 4 的周边电路





Q1为系统时钟相位。

软件在改变比较器的输入，输出控制寄存器时最好关闭比较器，避免比较器结果错误。

图 10.6 比较器 5 的周边电路

## 10.2.1 比较器配置

模拟比较器 4/5 总共有 8 种配置模式。由 CMCON0 寄存器的 CM<2:0>这 3 位来选择。图 10.7 详细描述了这 8 种选择的具体内容。I/O 信号线的功能也随着这 8 中不同的配置而改变，具体如下：

- 模拟功能 (A)：数字输入缓存被屏蔽
- 数字功能 (D)：比较器数字输出会覆盖管脚上的其他功能
- 正常端口功能 (I/O)：独立于比较器

当端口上标注“A”字样，读的时候无论当前管脚上的状态或 I/O 控制寄存器 TRIS 位的状态都会返回 0 值。用户应该把与被用作模拟输入的管脚相对应的 TRIS 位置为 1 来关闭其数字输出驱动电路。

当端口上标注“D”字样，用户应该将其相应的 TRIS 位置 0 来打开数字输出驱动电路。

另外，比较器配置切换应该屏蔽比较器中断以避免不必要的误触发事件。

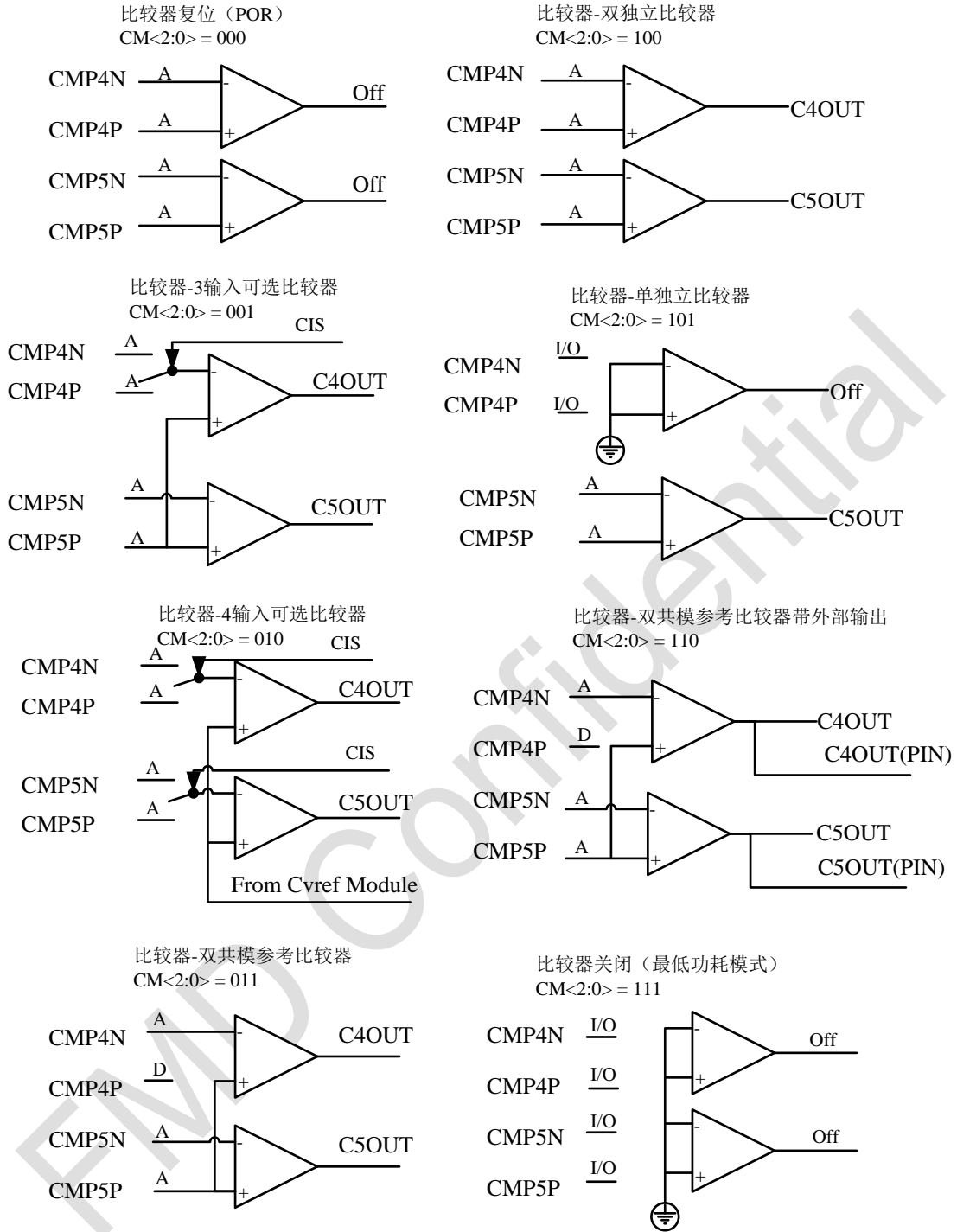


图 10.7 比较器输入，输出配置模式

### 10.2.2 模拟输入开关

比较器的模拟输入负端在以下模式可被切换到 2 个模拟管脚。

- $CM\langle 2:0 \rangle = 001$  (仅比较器 4)
- $CM\langle 2:0 \rangle = 010$  (比较器 4 和 5)

在以上模式下，无论哪根管脚被选做输入，这 2 个管脚都处在模拟模式。CMCON0 寄存器里的 CIS

位控制比较器的输入切换开关。

### 10.2.3 比较器 5 门控 Timer1

可使用此功能为模拟事件的持续时间或间隔定时。将 CMCON1 寄存器的 T1GSS 位清零将使 Timer1 根据比较器 5 的输出递增。这要求 Timer1 打开且门控使能。

详情请参见第 7 节“[带门控的 Timer1 模块](#)”。

当比较器用作 Timer1 门控源时，建议通过将 C5SYNC 位置 1 将比较器 5 与 Timer1 同步。这将确保 Timer1 在递增时若比较器变化，Timer1 不会错过递增。

### 10.2.4 比较器 5 输出与 Timer1 同步

通过将 CMCON1 寄存器的 C5SYNC 位置 1 可将比较器 5 的输出与 Timer1 同步。使能时，比较器输出在 Timer1 时钟源的下降沿被锁存。Timer1 使用预分频器时，比较器输出在预分频后被锁存。为了防止竞争情况，比较器输出在 Timer1 时钟源下降沿被锁存，而 Timer1 在时钟源的上升沿递增。更多信息，请参见比较器框图（图 10.6）以及 Timer1 框图（图 7.1）。

### 10.2.5 比较器中断

一旦使能，模拟比较器 4/5 输出状态的变化就会触发中断。输出状态变化的捕捉用两个锁存器和一个异或门完成。（见图 10.8, 10.9）。当读取 CMCON0 寄存器时，一个锁存器由比较器输出更新。该锁存器保持着状态直到 CMCON0 下一次被读或复位。另一个锁存器在系统时钟的 Q1 期更新。状态失配状态会被一直保存，并使 PIR1 寄存器的 CxIF 位为 1，直到 CMCON0 被读或比较器输出返回到以前的状态。（注，对 CMCON0 寄存器的写动作也会清除失配条件，应为所有的写动作实际上在写之前都包含着一个读动作）。

软件需要自身保存比较器的输出态来决定实际上变化是否有发生。PIR1 寄存器里的 CxIF 位是比较器的中断指示位。这 1 位必须由软件来清零。由于软件可以将这 1 位写成 1，因此软件可以做模拟比较器变化的动作。

比较器的中断发生还需要 INTCON 寄存器中的 PEIE 位和 GIE 位联合搭配工作。如果这其中任何 1 位没有置 1，哪怕 CxIF 位变成 1 了，中断都不会发生。用户可以通过以下动作清除中断：

1. 对 CMCON0 进行读或写操作，这会结束失配条件
2. 清除 CxIF 位

持续的失配条件会屏蔽住 CxIF 标志位被清，因此有必要先读一下 CMCON0 寄存器先清除失配条件。

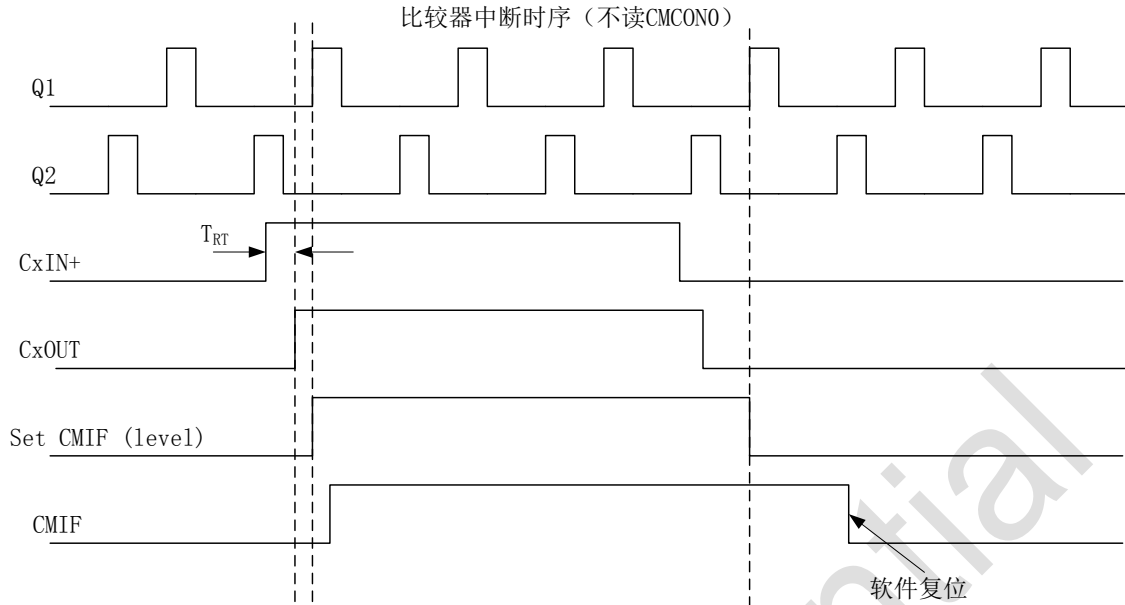


图 10.8 比较器中断时序 1

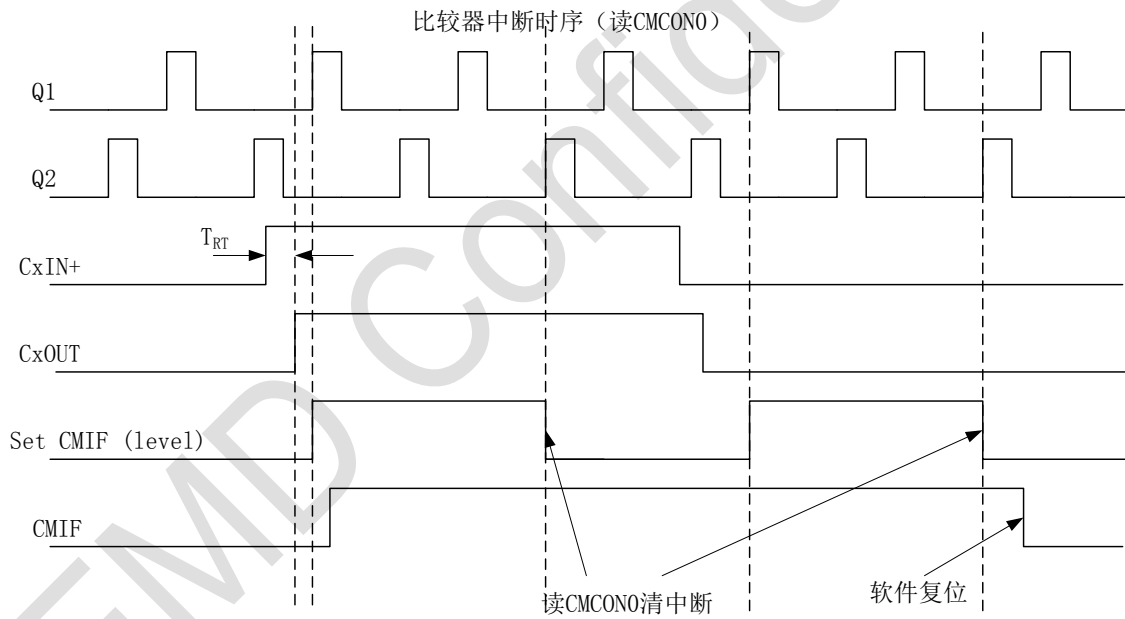


图 10.9 比较器中断时序 2

需要注意的是如果在读 **CMCON0** 的时候(在系统时钟的 **Q2** 期), **CxOUT** 刚好发生变化, 那么 **CxIF** 标志位有可能不会被置高。

另外, 比较器电路被启动时, 其偏置电路需要 **1uS** 的时间稳定, 在此期间, 比较器的输出是无效的, 软件应该在此期间关闭中断以免产生不必要的误触发。

## 10.2.6 比较器的参考电压

模拟比较器 **4/5** 的输入之一可以被设置成为内部参考电压, 该参考电压都有以下特性:

- 独立于比较器的运行
- 两个 16 级电压范围
- 输出低钳位到 VSS
- 输出电压与 VDD 成比例

参考电压的控制由 VRCON 寄存器来控制，如图 10.10。

### 10.2.6.1 独立工作

比较器参考电压与比较器配置不相关。将 VRCON 寄存器的 VREN 位将使能参考电压。

### 10.2.6.2 输出电压选择

CVREF 参考电压有两种范围，每种均为 16 个电平。范围的选择由 VRCON 寄存器的 VRR 位控制。这 16 个电平由 VRCON 寄存器的 VR<3:0>位设置。

CVREF 输出电压由以下公式确定：

**VRR = 1 (低电压范围):**  

$$CVREF = (VR<3:0>/24) \times VDD$$
**VRR = 0 (高电压范围):**  

$$CVREF = (VDD/4) + (VR<3:0> \times VDD/32)$$

如图 10.10 所示，由于模块的构造所限，无法实现 VSS 至 VDD 的满量程。

### 10.2.6.3 输出钳位至 VSS

通过将 VRCON 按如下配置可将 CVREF 输出电压设置为 VSS，从而不消耗功率：

- VREN = 0
- VRR = 1
- VR<3:0> = 0000

这使比较器可进行过零检测而不消耗额外的 CVREF 模块电流。

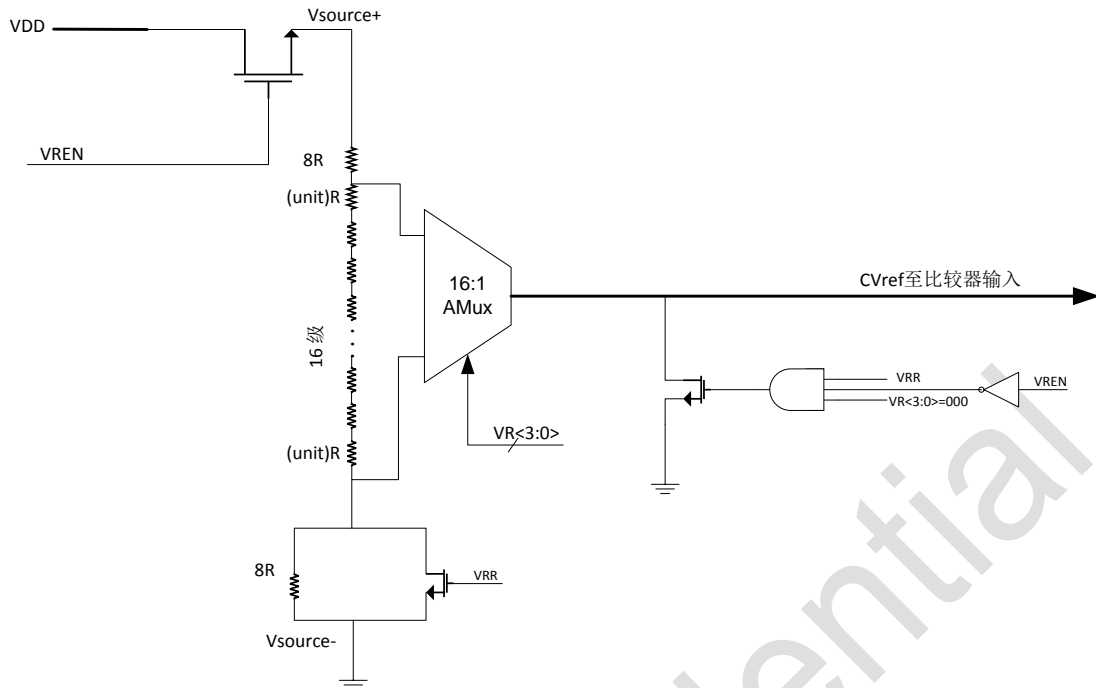


图 10.10 比较器参考电压框图

### 10.3 模拟输入端的连接

比较器的模拟输入端和同样连接该端口的数字输入共享管脚，需要注意的是在这些管脚上都有对 VDD 和 VSS 的反向保护二极管。如果输入电压偏离这个保护范围以外 0.6V，二极管就会导通到时间锁效应。我们推荐的最大输入源信号的源阻抗不超过 10K 欧姆。另外，如果管脚上外挂电容或齐纳二极管之类的元件，其不应该产生漏电流，不然可能会造成结果不精确。

需要注意的是，当读一个端口寄存器时，如果改管脚被配置为模拟信号管脚，软件会读出 0 值。当管脚被设置为数字输入管脚时，比较器仍然会以为该管脚输入一个模拟信号，并输出相应结果。如果一个管脚被设置为数字输入，而这个管脚上面的实际电压又是一个模拟电平，这可能造成输入缓存电路消耗比说明书上标的更大的电流。

### 10.4 比较器控制

CMCONx, CMPCRx 寄存器提供了以下比较器特性的控制功能：

- 打开关闭
- 迟滞控制
- 输出状态
- 输出极性
- 输入开关

## 10.5 模拟输出状态

每个比较器的状态都能通过 CMPxCR 寄存器（或者 CMCON0）的 CxOUT 位从内部读出，如果使能了去抖功能的话，CPU 读到的是经过去抖之后的输出。

比较器 4 和比较器 5 的结果可直接输出到管脚上，此时需要把 CMCON0 的 CM<2:0>设置为 110，同时 C4OUT 和 C5OUT 相应的 TRIS 位必须被清零来使能输出驱动电路。

## 10.6 模拟输出极性

将比较器输出反向就等效于将比较器的输入端交换。比较器输出极性可由 CMCON2/3 寄存器里的 CxPOL 位（比较器 4/5 是 CxINV 位）来控制。将 CxPOL 位清零代表比较器输出未被反向。完整的输出，输入条件和极性如下表所列：

Input Condition	CxPOL (或 CxINV)	CxOUT
Vin- > Vin+	0	0
Vin- < Vin+	0	1
Vin- > Vin+	1	1
Vin- < Vin+	1	0

CxOUT 包括寄存器位和实际输出管脚。

## 10.7 比较器反应时间

模拟比较器输出在输入变化或输入新的参考电压后若干时间后才会变化，这个时间叫做比较器反应时间。反应时间和参考电压的稳定时间还不是同一个概念。这两者加起来才是比较器的总体反应时间。具体参数，请参考本文档后面的电气参数部分。

## 10.8 比较器在睡眠状态下的工作

如果用户在使芯片进入睡眠之前使能比较器（CMPON.x 为 1），那么比较器会在睡眠状态下继续工作。但此时芯片的功耗会相应增加。如果用户并不使用比较器输出来唤醒芯片，用户可以在睡眠前关闭比较器。关闭比较器需要把 CMPON.x 清 0。

如上所述，比较器的输出可以用来唤醒芯片。如果需要用比较器输出来唤醒芯片，需要在 PIE1 寄存器中将 CxIE 位置 1，以及 INTCON 寄存器下的 PEIE 位置 1。如果 INTCON 寄存器的 GIE 位被置 1，器件就会执行中断服务程序。

## 10.9 比较器在复位状态下的工作

芯片复位会强制把 CMCON4 寄存器进入复位状态，也就是 CMPON[3:0]=0000 模式，比较器 0/1/2/3 被关闭以节省功耗。

而比较器 4/5 的情况稍微不同，复位时  $CM[2:0]=000$ ，若想得到最低功耗，需要软件把  $CM[2:0]$  置为 111。

## 10.10 校准输入失调电压

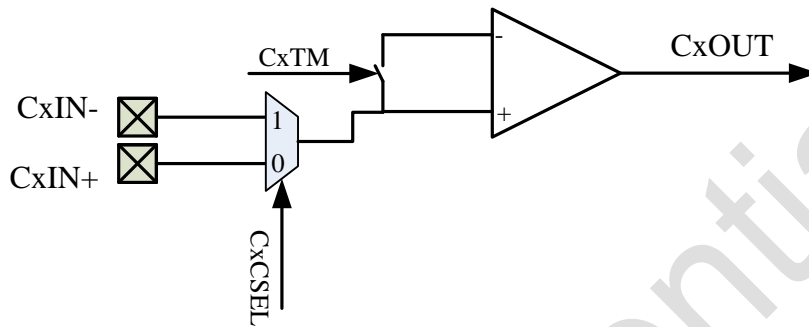


图 10.5 输入失调电压校准

比较器 0/1/2/3 的输入失调电压均可在应用中校准，它是通过调整共模输入来实现的。操作步骤如下：

1. 设置  $CMPON.x$  为 1，打开比较器  $x$ ；
2. 设置  $CxTM$  为 1，进入校准测试模式（比较器两个输入端短接）；
3. 设置  $CxCSEL$ ，选择同相或者反相端作为参考电压输入；
4. 设置相关  $TRIS$  为 1 把数字输出驱动关闭；
5. 调整  $CxCOF[4:0]$  直到  $CxOUT$  输出状态改变；
6. 把  $CxTM$  清 0，恢复比较器为正常模式；

注意：比较器 1 和比较器 3 的同相输入端没有连接到管脚，所以做输入失调校准时只能选择反相端作为参考电压输入脚。

## 10.11 与比较器相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值	
ANSEL	0x91	ANSEL[7:0], 数模管脚选择								1111 1111	
INTCON	0x0B/8B	GIE	PEIE	T0IE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000	
CMCON4	0x11F	C3DEB	C2DEB	C1DEB	-	CMPON[3:0]				111- 0000	
VRCON	0x99	VREN	-	VRR	-	VR[3:0]				0-0- 0000	
CMCON0	0x19	C5OUT	C4OUT	C5INV	C4INV	CIS	CM[2:0]			xx00 0000	
CMCON1	0x1A	HYSEN[3:0]				-	-	T1GSS	C5SYNC		0000 --10
CMCON2	0x101	C1POL	C0POL	VR2C[2:0]			VR1C[2:0]			0000 0000	
CMCON3	0x107	C3POL	C2POL	VR4C[2:0]			VR3C[2:0]			0000 0000	
CMP0CR	0x190	C0OUT	C0TM	C0CSEL	C0COF[4:0]					x001 0000	



CMP1CR	0x191	C1OUT	C1TM	C1CSEL	C1COF[4:0]					x001 0000
CMP2CR	0x192	C2OUT	C2TM	C2CSEL	C2COF[4:0]					x001 0000
CMP3CR	0x193	C3OUT	C3TM	C3CSEL	C3COF[4:0]					x001 0000
PIE1	0x8C	EEIE	CKMEAIE	-	C5IE	C4IE	OSFIE	TMR2IE	TMR1IE	00-0 0000
PIE2	0x8D	-	-	C3IE	C2IE	C1IE	C0IE	ADIE	CCP1IE	--00 0000
PIR1	0x0C	EEIF	CKMEAIF	-	C5IF	C4IF	OSFIF	TMR2IF	TMR1IF	00-0 0000
PIR2	0x0D	-	-	C3IF	C2IF	C1IF	C0IF	ADIF	CCP1IF	--00 0000
TRISA	0x85	TRISA[7:0], PORTA 方向控制								1111 1111
PORTA	0x05	PORTA[7:0], PORTA 数据寄存器								xxxx xxxx
TRISC	0x87	-	-	TRISC[5:0], PORTC 方向控制					--11 1111	
PORTC	0x07	-	-	PORTC[5:0], PORTC 数据寄存器					--xx xxxx	

### 10.11.1 CMCON0 寄存器，地址 0x19

Bit	7	6	5	4	3	2	1	0
Name	C5OUT	C4OUT	C5INV	C4INV	CIS	CM2	CM1	CM0
Reset	X	X	0	0	1	0	0	0
Type	RO	RO	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	C5OUT	比较器 5 输出 bit 当 C5INV=0 1: C5 V <sub>IN+</sub> > C5 V <sub>IN-</sub> 0: C5 V <sub>IN+</sub> < C5 V <sub>IN-</sub> 当 C5INV=1 1: C5 V <sub>IN+</sub> < C5 V <sub>IN-</sub> 0: C5 V <sub>IN+</sub> > C5 V <sub>IN-</sub>
6	C4OUT	比较器 4 输出 bit 当 C4INV=0 1: C4 V <sub>IN+</sub> > C4 V <sub>IN-</sub> 0: C4 V <sub>IN+</sub> < C4 V <sub>IN-</sub> 当 C4INV=1 1: C4 V <sub>IN+</sub> < C4 V <sub>IN-</sub> 0: C4 V <sub>IN+</sub> > C4 V <sub>IN-</sub>
5	C5INV	比较器 5 输出反向 bit 0 = 不反向 1 = 反向
4	C4INV	比较器 4 输出反向 bit 0 = 不反向 1 = 反向
3	CIS	比较器输入切换

		<p>当CM[2:0] = 010时， 1 = CMP4P连接到C4 V<sub>IN</sub>。 CMP5P连接到C5 V<sub>IN</sub>。 0 = CMP4N连接到C4 V<sub>IN</sub>。 CMP5N连接到C5 V<sub>IN</sub>。</p> <p>当CM[2:0] = 001时， 1 = CMP4P连接到C4 V<sub>IN</sub>。 0 = CMP4N连接到C4 V<sub>IN</sub>。</p>
2-0	CM[2:0]	<p>比较器模式位:</p> <p>000 = 比较器关闭, CxIN管脚为模拟IO管脚</p> <p>001 = 3个输入共用到2个比较器上</p> <p>010 = 4个输入共用到2个比较器上</p> <p>011 = 2个共参考比较器</p> <p>100 = 2个独立比较器</p> <p>101 = 1个独立比较器</p> <p>110 = 2个带输出共参考比较器</p> <p>111 = 比较器关闭, CxIN管脚为数字IO管脚</p>

### 10.11.2 CMCON2/3 寄存器, 地址 0x101, 0x107

#### CMCON2 寄存器

Bit	7	6	5	4	3	2	1	0
Name	C1POL	C0POL	VR2C[2:0]			VR1C[2:0]		
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	C1POL	比较器 1 输出极性选择 1 = 取反比较器 1 输出 0 = 不取反比较器 1 输出
6	C0POL	比较器 0 输出极性选择 1 = 取反比较器 0 输出 0 = 不取反比较器 0 输出
5:3	VR2C[2:0]	比较器参考源 VR2 设置 VR2电压= (0.6+VR2C/40)VDD
2:0	VR1C[2:0]	比较器参考源 VR1 设置 VR1电压= (0.6+VR1C/40)VDD

#### CMCON3 寄存器

Bit	7	6	5	4	3	2	1	0
Name	C3POL	C2POL	VR4C[2:0]			VR3C[2:0]		

Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	C3POL	比较器 3 输出极性选择 1 = 取反比较器 3 输出 0 = 不取反比较器 3 输出
6	C2POL	比较器 2 输出极性选择 1 = 取反比较器 2 输出 0 = 不取反比较器 2 输出
5:3	VR4C[2:0]	比较器参考源 VR4 设置 VR4电压= (0.6+VR4C/40)VDD
2:0	VR3C[2:0]	比较器参考源 VR3 设置 VR3电压= (0.075+VR3C/40)VDD

### 10.11.3 CMCON1 寄存器，地址 0x1A

Bit	7	6	5	4	3	2	1	0
Name	HYSEN[3:0]				-	-	T1GSS	C5SYNC
Reset	0	0	0	0	-	-	1	0
Type	RW	RW	RW	RW	-	-	RW	RW

Bit	Name	Function
7:4	HYSEN[3:0]	比较器迟滞使能位，HYSEN[3:0]位分别控制比较器 3/2/1/0 1 = 迟滞使能 0 = 禁止迟滞
3:2	-	保留位，未实现
1	T1GSS	Timer1 门控源选择位 1 = Timer1 门控源为 T1G 引脚（引脚应配置为数字输入） 0 = Timer1 门控源为比较器 5 的输出
0	C5SYNC	比较器 5 输出同步位 1 = 输出与 Timer1 时钟的下降沿同步 0 = 异步输出

### 10.11.4 CMCON4 寄存器，地址 0x11F

Bit	7	6	5	4	3	2	1	0
Name	C3DEB	C2DEB	C1DEB	-	CMPON[3:0]			
Reset	1	1	1	-	0	0	0	0

Type	Q-RW	Q-RW	Q-RW	R-0	Q-RW	Q-RW	Q-RW	Q-RW
------	------	------	------	-----	------	------	------	------

**Q-RW:** 当 UCFG1.7 为 1 时，这些位可以由软件读写，否则不能由软件改写，其值只在上电配置过程确定。

Bit	Name	Function
7	C3DEB	比较器 3 输出去抖使能 1 = 输出带去抖，时间为 8 个系统时钟周期 0 = 输出不带去抖
6	C2DEB	比较器 2 输出去抖使能 1 = 输出带去抖，时间为 8 个系统时钟周期 0 = 输出不带去抖
5	C1DEB	比较器 1 输出去抖使能 1 = 输出带去抖，时间为 8 个系统时钟周期 0 = 输出不带去抖
4	-	保留位，未实现
3:0	CMPON[3:0]	比较器使能位，CMPON[3:0]分别控制比较器 3/2/1/0 1 = 开启比较器 0 = 关闭比较器

### 10.11.5 CMPxCR 寄存器，地址 0x190~0x193

CMPxCR 寄存器，x 的范围是 0~3

Bit	7	6	5	4	3	2	1	0
Name	CxOUT	CxTM	CxCSEL	CxCOF[4:0]				
Reset	x	0	0	1	0	0	0	0
Type	RO	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	CxOUT	比较器 x 的输出 当 CxPOL=0 时： 1 = CxIN+ > CxIN- 0 = CxIN+ < CxIN- 当 CxPOL=1 时： 1 = CxIN+ < CxIN- 0 = CxIN+ > CxIN-
6	CxTM	比较器输入失调校准模式 1 = 进入失调校准模式 0 = 正常比较模式
5	CxCSEL	输入失调校准模式下比较器的输入端选择 1 = 选择 CxIN-作为参考输入脚 0 = 选择 CxIN+作为参考输入脚

4:0	CxCOF[4:0]	比较器输入失调校准位
-----	------------	------------

### 10.11.6 VRCON 寄存器，地址 0x99

比较器 4、5 的参考电压控制。

Bit	7	6	5	4	3	2	1	0
Name	VREN	-	VRR	-	VR3	VR2	VR1	VR0
Reset	0	-	0	-	0	0	0	0
Type	RW	-	RW	-	RW	RW	RW	RW

Bit	Name	Function
7	VREN	CVref 使能位 1 = CVref 电路通电 0 = CVref 电路断电，无泄漏电流
6	-	未实现，读 0
5	VRR	CVref 范围选择位 1 = 低电平范围 0 = 高电平范围
4	-	未实现，读 0
3:0	VR<3:0>	CVref 值选择位 VRR=1 时: $CVref = (VR<3:0>/24)*VDD$ VRR=0 时: $CVref = VDD/4 + (VR<3:0>/32)*VDD$

## 11 数据 EEPROM

片内集成有 256 个字节的 EEPROM, 通过 EEADR 进行寻址访问。软件可通过 EECON1 和 EECON2 对 EEPROM 进行编程操作, 硬件实现了擦除和编程的自定时功能, 无需软件查询, 节省有限的代码空间, 同时利用此特性, 启动编程周期之后可以进入睡眠模式, 以降低功耗。

编程 EEPROM 需要遵循一定的步骤, 这种机制可以防止程序跑飞或者程序丢失引起的误写操作。

### 11.1 编程数据 EEPROM 步骤

- A. 把 INTCON 的 GIE 位清 0;
- B. 判断 GIE 是否为 1, 是则重复 A 步骤, 否则可以进行下一步;
- C. 往 EEADR 写入目标地址;
- D. 往 EEDAT 写入目标数据;
- E. 把位 WREN3/WREN2/WREN1 全部置 1;
- F. 把位 WR 置 1 (EECON2.0, 此后 WR 会维持高);
- G. 写过程不能改变 WREN3/2/1 的值, 否则编程终止;
- H. 等大概 2ms 之后编程自动完成, WR 自动清 0, WREN3、WREN2、WREN1 自动清 0;
- I. 如果想再次编程, 重复步骤 C~H 即可;

例子 1:

```
BCR INTCON, GIE
BTSC INTCON, GIE
LJUMP $-2
BANKSEL EEADR
LDWI 55H
STR EEADR _____;地址为 0x55
STR EEDAT _____;数据为 0x55
LDWI 34H
STR EECON1 _____;WREN3/2/1 同时置 1
BSR EECON2, 0 _____;启动写
BSR INTCON, GIE _____;把 GIE 置 1
```

例子 2:

```
BCR INTCON, GIE
BTSC INTCON, GIE
LJUMP $-2
BANKSEL EEADR
LDWI 55H
STR EEADR _____;地址为 0x55
STR EEDAT _____;数据为 0x55
```

LDWI 34H

STR EECON1 ;WREN3/2/1 同时置 1

NOP ;这里 NOP 可以换成其他指令

BSR EECON2, 0 ;启动写，实际上硬件不会启动编程 EEPROM 操作

BCR EECON1, WREN1 ;先清 WREN1，使得 WREN3/2/1 不同时为 1

BSR EECON1, WREN1 ;重新置位 WREN1，令 WREN3/2/1 同时为 1

BSR EECON2, 0 ;启动写，这次硬件将对 EEPROM 编程

BSR INTCON, GIE

注意：

1. 以上步骤的 E、F 两步必须是连续的两条指令周期完成，不能错开（如例子 2），否则编程操作不会启动，其中 WREN3、WREN2 和 WREN1 可以不是同一条指令置 1，比如可以用 BSR 指令分开对各位置 1；
2. 如果 E、F 两步被错开执行，要想启动下一次编程操作，必须在 E、F 之前加入一步，把 WREN3、WREN2 或者 WREN1 任意一位清 0，如例子 2；
3. 编程过程中读操作无效；
4. 启动编程（EECON2.0 置 1）后如果要睡眠，至少要等一条指令，例如 NOP。

```
BSR EECON2, 0 ;
NOP ;可以是别的非 SLEEP 指令
SLEEP
```

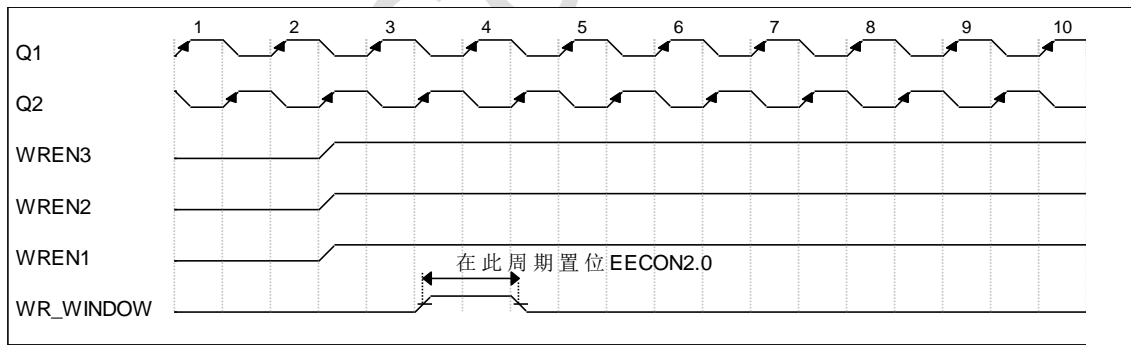


图 11.1 软件编程数据 EEPROM 时序

## 11.2 关于编程周期

启动数据 EEPROM 的编程操作后，2ms 的编程计时开始，在这段时间内，CPU 并不会暂停，而是继续执行程序。

## 11.3 读数据 EEPROM

要读取数据存储单元，用户必须将地址写入 **EEADR** 寄存器，然后将 **EECON1** 寄存器的控制位 **RD** 置 1。在紧接着的下一周期，**EEDAT** 寄存器就被 **EEPROM** 数据写入。因此该数据可由下一条指令读取。**EEDAT** 将保持这个值直到用户下一次从该单元读取或向该单元写入数据时（在写操作过程中）。

下面是读取 **EEPROM** 的一段示例程序：

**BANKSEL EEADR**

**LDWI dest\_addr**

**STR EEADR**

**BSR EECON1, RD**

**LDR EEDAT, W**

## 11.4 与数据 EEPROM 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
INTCON	0x0B/8B	GIE	PEIE	T0IE	INTE	PAIE	TOIF	INTF	PAIF	0 0 0 0 0 0 0 0
PIE1	0x8C	EEIE	CKMEAIE	-	C5IE	C4IE	OSFIE	TMR2IE	TMR1IE	0 0 - 0 0 0 0 0
PIR1	0x0C	EEIF	CKMEaIF	-	C5IF	C4IF	OSFIF	TMR2IF	TMR1IF	0 0 - 0 0 0 0 0
EEDAT	0x9A	EEDAT[7:0]								0 0 0 0 0 0 0 0
EEADR	0x9B	EEADR[7:0]								0 0 0 0 0 0 0 0
EECON1	0x9C	-	-	WREN3	WREN2	WRERR	WREN1	-	RD	--00 x0-0
EECON2	0x9D	-	-	-	-	-	-	-	WR	---- ---0

### 11.4.1 EEDAT 寄存器，地址 0x9A

Bit	7	6	5	4	3	2	1	0
Name	EEDAT[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW



### 11.4.2 EEADR 寄存器，地址 0x9B

Bit	7	6	5	4	3	2	1	0
Name	EEADR[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

### 11.4.3 EECON1 寄存器，地址 0x9C

Bit	7	6	5	4	3	2	1	0
Name	-	-	WREN3	WREN2	WRERR	WREN1	-	RD
Reset	-	-	0	0	x	0	-	0
Type	-	-	RW	RW	RW	RW	-	WO

Bit	Name	Function
7:6	-	保留位，读 0
5	WREN3	数据 EEPROM 写使能 3 和 WREN2、WREN1 结合使用
4	WREN2	数据 EEPROM 写使能 2 和 WREN3、WREN1 结合使用
3	WRERR	数据 EEPROM 写错误标志位 1: 在 EEPROM 编程周期发生了看门狗或者外部复位，中止 0: 在 EEPROM 编程周期正常完成
2	WREN1	数据 EEPROM 写使能 1 WREN3-1=111: 允许软件对 EEPROM 编程，编程完成后各位自动清 0 WREN3-1=其他值: 禁止软件对 EEPROM 编程
1	-	保留位，读 0
0	RD	数据 EEPROM 读控制位 写 1: 启动一次数据 EEPROM 读周期 写 0: 不启动读

### 11.4.4 EECON2 寄存器，地址 0x9D

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	WR
Reset	-	-	-	-	-	-	-	0
Type	-	-	-	-	-	-	-	RW

Bit	Name	Function
7:1	-	保留位，读 0
0	WR	数据 EEPROM 写控制位 读操作： 1: 数据 EEPROM 编程周期进行中 0: 数据 EEPROM 不处于编程周期 写操作： 1: 启动一次数据 EEPROM 编程周期 0: 无意义

FMD Confidential

## 12 模数转换器（ADC）模块

模数转换器（Analog-to-digital Converter, ADC）可将模拟输入信号转换为相应的 10 位二进制表征值。该系列器件采用多个模拟输入复用到一个采样保持电路。采样保持电路的输出与转换器的输入相连接。转换器通过逐次逼近法产生 10 位二进制值，并将转换结果保存在 ADC 结果寄存器（ADRESL:ADRESH）中。ADC 参考电压可用软件选择为 VDD 或施加在外部参考引脚上的电压。ADC 可在转换完成时产生中断。该中断可用于将器件从休眠唤醒。

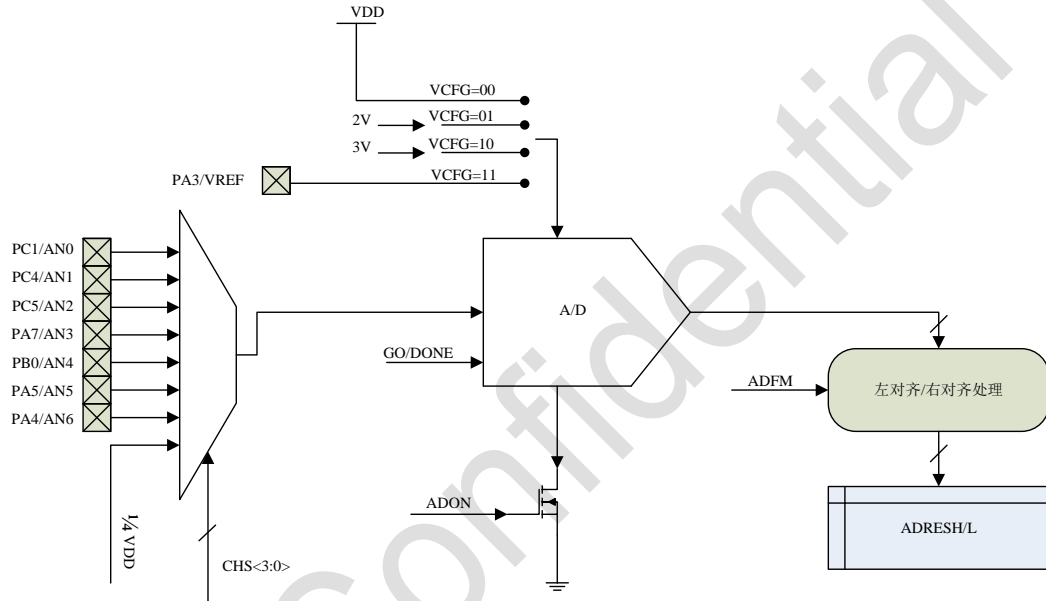


图 12.1 ADC 原理框图

### 12.1 ADC 的配置

配置和使用 ADC 时，必须考虑以下功能：

- 端口配置
- 通道选择
- ADC 参考电压的选择
- ADC 转换时钟源
- 中断控制
- 转换结果的格式

#### 12.1.1 端口配置

ADC 可用于转换模拟和数字信号。转换模拟信号时，应将相关的 TRIS 和 ANSEL 位置 1 将 I/O 引脚应配置为模拟功能。更多信息请参见相应的端口章节。

注意：如果定义为数字输入的引脚上存在模拟电压，可导致输入缓冲器传导过大的电流。

### 12.1.2 通道选择

ADCON0 寄存器的 CHS 位决定将哪个通道连接到采样保持电路。改变通道时，开始下一次转换前需要一个延时。更多信息请参见第 12.2 节“ADC 的工作原理”。

### 12.1.3 ADC 参考电压

ADCON0 寄存器的 VCFG 位提供对正参考电压的控制。正参考电压可以是 VDD，内部参考 2V 或者 3V，也可以是外部电压源。负参考电压始终连接到参考地。

### 12.1.4 转换时钟

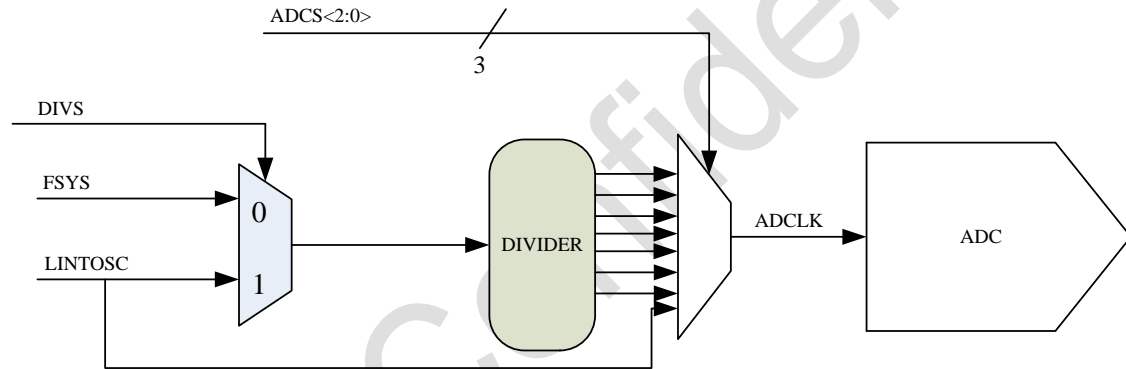


图 12.2 ADC 的时钟配置原理

转换时钟源可通过 ADCON1 寄存器的 ADCS 位用软件选择。有以下 13 种时钟选项：

- SYSCLK/2 或者 LFINTOSC/2
- SYSCLK/4 或者 LFINTOSC/4
- SYSCLK/8 或者 LFINTOSC/8
- SYSCLK/16 或者 LFINTOSC/16
- SYSCLK/32 或者 LFINTOSC/32
- SYSCLK/64 或者 LFINTOSC/64
- FRC（内部慢时钟振荡器）

完成一位 (bit) 的转换时间定义为 TAD。完成 10 位转换需要 11.5 个 TAD 周期（不包括采样时间），如图 12.3 和 12.5 所示。

进行正确的转换必须满足相应的 TAD 规范。更多信息请参见第 18 节“电气特性”中的 A/D 转换要求。表 12.1 所示为正确选择 ADC 时钟的示例。

注意:

1. 除非使用的是 **FRC**，否则任何系统时钟频率的变化均会改变 **ADC** 时钟频率，这将对 **ADC** 结果产生负面影响；
2. **FRC** 可以是 **256KHz** 或者是 **32KHz**，取决于 **LFMOD** 为何值。

ADC 时钟周期 ( $T_{AD}$ )		系统时钟频率 ( $F_{SYS}$ )			
ADC 时钟源	ADCS<2:0>	16MHz	8MHz	4MHz	1MHz
$F_{SYS} / 2$	000	125ns	250ns	500ns	2.0us
$F_{SYS} / 4$	100	250ns	500ns	1.0us	4.0us
$F_{SYS} / 8$	001	500ns	1.0us	2.0us	8.0us
$F_{SYS} / 16$	101	1.0us	2.0us	4.0us	16.0us
$F_{SYS} / 32$	010	2.0us	4.0us	8.0us	32.0us
$F_{SYS} / 64$	110	4.0us	8.0us	16.0us	64.0us
$F_{RC}$	x11	2~6us	2~6us	2~6us	2~6us

表 12.1 ADC 时钟周期和器件工作频率

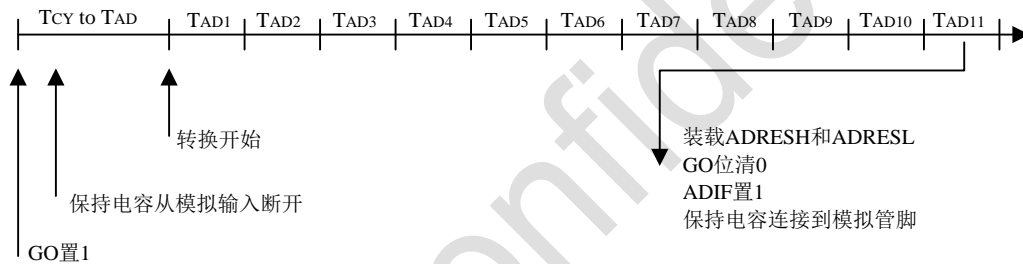


图 12.3 模数转换 TAD 周期

### 12.1.5 中断

ADC 模块可使中断在模数转换完成时产生。ADC 中断标志为 PIR2 寄存器中的 ADIF 位。ADC 中断使能为 PIE2 寄存器中的 ADIE 位。ADIF 位必须用软件清零。

注意：无论 ADC 中断是否被打开，ADIF 位在每次转换完成时均置 1。

器件工作或处于休眠状态时均可产生中断。如果器件处于休眠状态，中断将唤醒器件。从休眠唤醒时，始终执行 SLEEP 指令后的那条指令。如果用户试图唤醒器件并恢复顺序执行代码，必须禁止全局中断。如果允许全局中断，代码执行将转至中断服务程序。

### 12.1.6 转换结果的格式

10 位 A/D 转换结果有两种格式，即左对齐和右对齐。ADCON0 寄存器的 ADFM 位控制输出格式。

图 12.4 所示为两种输出格式。

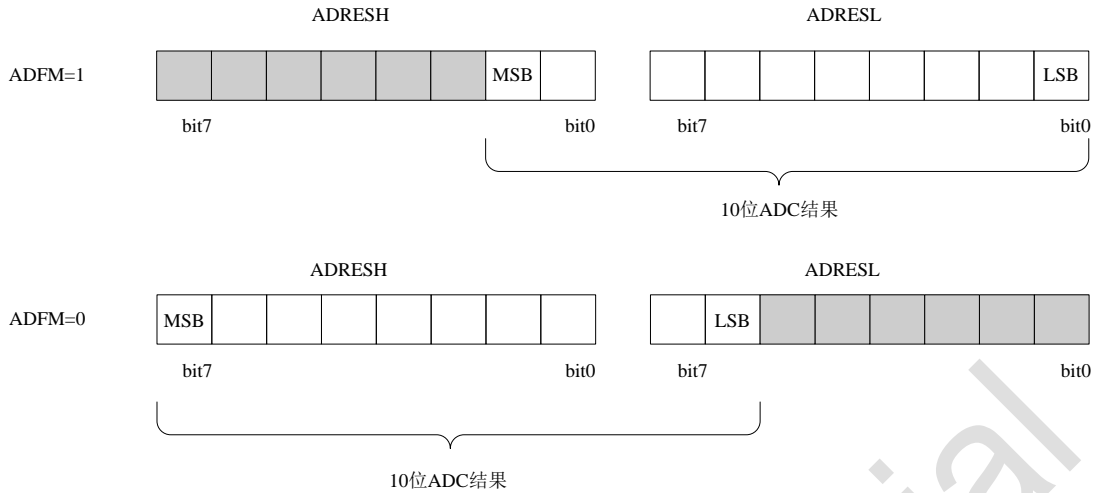


图 12.4 ADC 转换结果格式示意

## 12.2 ADC 的工作原理

### 12.2.1 启动转换

要启用 ADC 模块，必须将 ADCON0 寄存器的 ADON 位置 1。将 ADCON0 寄存器的 GO/DONE 位置 1 将启动模数转换。

**注意：**应在打开 ADC 的那条指令中将 GO/DONE 位置 1。请参见第 12.2.6 节“A/D 转换步骤”。

### 12.2.2 转换完成

转换完成时，ADC 模块将：

- 将 GO/DONE 位清零
- 将 ADIF 标志位置 1
- 用新的转换结果更新 ADRESH:ADRESL 寄存器

### 12.2.3 终止转换

如果转换必须在完成前被终止，可用软件将 GO/DONE 清零。ADRESH:ADRESL 寄存器不会被未完成的模数转换采样更新。相反，ADRESH:ADRESL 这对寄存器将保持先前转换的值。此外，启动下一次采集前，需等待 TACQ（采样时间，图 12.4）的延时。延时结束后，所选通道的输入采集将自动启动。

**注意：**器件复位将强制所有寄存器回到其复位状态。这样，ADC 模块就被关闭，并且任何待处理的转换均被终止。

## 12.2.4 休眠模式下 ADC 的工作

ADC 模块可在休眠期间工作，这要求将 ADC 时钟源置于 FRC 选项。选定 FRC 时钟源后，软件可以置 MCU 于 SLEEP 模式，从而降低 ADC 转换期间的系统噪声。如果允许 ADC 中断，转换完成后器件将从休眠唤醒。如果禁止 ADC 中断，ADC 模块在转换完成后关闭，尽管 ADON 位保持置 1 状态。

如果 ADC 时钟源不是 FRC，执行一条 SLEEP 指令将使当前转换中止，并且 ADC 模块被关闭，尽管 ADON 位保持置 1 状态。

## 12.2.5 特殊事件触发器

ECCP 特殊事件触发器可在软件不干预的情况下周期性地进行 ADC 测量。发生触发事件时，GO/DONE 位由硬件置 1，Timer1 计数器复位为零。特殊事件触发器的使用并不确保正常 ADC 定时，用户必须确保满足 ADC 定时要求。

更多信息请参见第 9 节“增强型捕捉/比较/PWM+（带自动关闭和死区）模块”。

## 12.2.6 A/D 转换步骤

以下是使用 ADC 进行模数转换的步骤示例：

1. 配置端口：
  - 禁止引脚输出驱动器（见 TRIS 寄存器）
  - 将引脚配置为模拟
2. 配置 ADC 模块：
  - 选择 ADC 转换时钟
  - 配置参考电压
  - 选择 ADC 输入通道
  - 选择转换结果的格式
  - 打开 ADC 模块
3. 配置 ADC 中断（可选）：
  - 将 ADC 中断标志清零
  - 允许 ADC 中断
  - 允许外设中断
  - 允许全局中断
4. 等待所需稳定时间  $T_{ST}^{(1)}$ ；
5. 等待所需的采集时间  $T_{ACQ}^{(2)}$ ；
6. 将 GO/DONE 置 1 启动转换；
7. 通过以下情况之一等待 ADC 转换完成：
  - 查询 GO/DONE 位
  - 等待 ADC 中断（允许中断时）

8. 读取 ADC 结果;
9. 将 ADC 中断标志清零 (在允许了中断的情况下这一步是必需的)。

以下是一段示例代码:

```

BANKSEL ADCON1 ;
LDWI B'01110000' ;ADC Frc clock
STR ADCON1 ;
BANKSEL TRISC ;
BSR TRISC,4 ;Set PC4 to input
BANKSEL ANSEL ;
BSR ANSEL,1 ;Set PC4 to analog
BANKSEL ADCON0 ;
LDWI B'10000101' ;Right justify,
STR ADCON0 ; Vdd Vref, AN1, On
LCALL StableTime ; ADC stable time
LCALL SampleTime ;Acquisiton delay
BSR ADCON0,GO ;Start conversion
BTSC ADCON0,GO ;Is conversion done?
LJUMP $-1 ;No, test again
BANKSEL ADRESH ;
LDR ADRESH,W ;Read upper 2 bits
STR RESULTHI ;store in GPR space
BANKSEL ADRESL ;
LDR ADRESL,W ;Read lower 8 bits
STR RESULTLO ;Store in GPR space

```

注意:

1.  $T_{ST}$  时间是 ADC 的稳定时间, 当使用内部参考时, ADC 首次启动还需要考虑参考电压的稳定时间  $T_{VRINT}$ , 等待时间应取两者的较大者, 即  $\max(T_{VRINT}, T_{ST})$ ;
2. 见图 12.5, ADC 转换时序;
3. 切换通道后必须等待足够长的  $T_{ACQ}$  时间, 即上面示例中的 **SampleTime** 必须满足表格 21.12 的  $T_{ACQ}$  时间要求, 否则 ADC 精度、线性度将不能保证;



### 12.3 A/D 采集时间要求

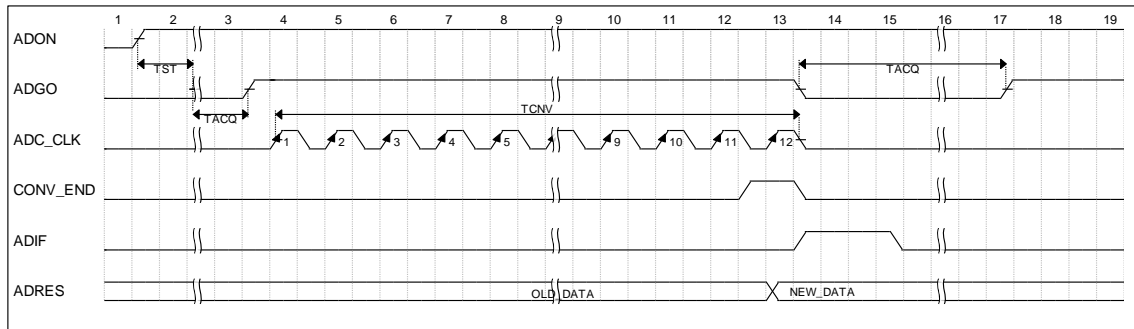


图 12.5 ADC 转换时序图

为了使 ADC 达到规定的精度，必须使充电保持电容 (CHOLD) 充满至输入通道的电平。模拟输入模型请参见图 12.6。源阻抗 (RS) 和内部采样开关 (RSS) 阻抗直接影响电容 CHOLD 的充电时间。采样开关 (RSS) 阻抗随器件电压 (VDD) 的变化而变化，参见图 12.6。建议模拟信号源的最大阻抗为 10kΩ。采集时间随着源阻抗的降低而缩短。在选择 (或改变) 模拟输入通道后，必须在开始转换前完成采集。

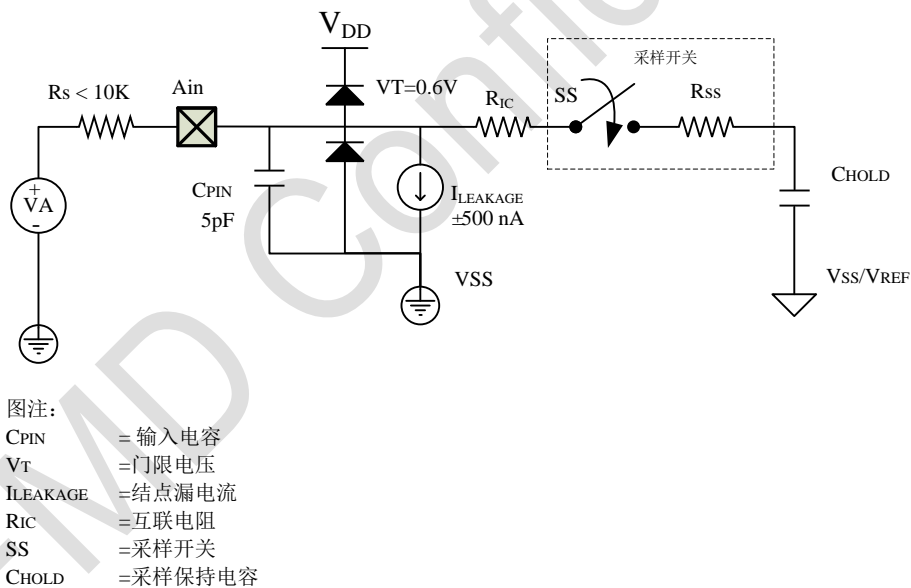


图 12.6 模拟输入模型

### 12.4 与 ADC 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
ADCON0	0x1F	ADFM	VCFG1	VCFG0	CHS2	CHS1	CHS0	GO/DONE	ADON	0000 0000
ADCON1	0x9F	DIVS	ADCS2	ADCS1	ADCS0	-	-	-	AN0SEL	0000 ---0

ADRESH	0x1E	ADC 结果高 8 位								xxxx xxxx
ADRESL	0x9E	ADC 结果低 8 位								xxxx xxxx
ANSEL	0x91	ANSEL[7:0], 数模管脚选择								1111 1111
INTCON	0x0B/8B	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
PIE1	0x8C	EEIE	CKMEAIE	-	C5IE	C4IE	OSFIE	TMR2IE	TMR1IE	00-0 0000
PIR1	0x0C	EEIF	CKMEaIF	-	C5IF	C4IF	OSFIF	TMR2IF	TMR1IF	00-0 0000
TRISA	0x85	TRISA[7:0], PORTA 方向控制								1111 1111
PORTA	0x05	PORTA[7:0], PORTA 数据寄存器								xxxx xxxx
TRISC	0x87	-	-	TRISC[5:0], PORTC 方向控制						--11 1111
PORTC	0x07	-	-	PORTC[5:0], PORTC 数据寄存器						--xx xxxx

### 12.4.1 ADCON0 寄存器，地址 0x1F

Bit	7	6	5	4	3	2	1	0
Name	ADFM	VCFG1	VCFG0	CHS2	CHS1	CHS0	GO/DONE	ADON
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	ADFM	A/D 转换结果格式选择位 1 = 右对齐 0 = 左对齐
6:5	VCFG[1:0]	参考电压选择位 11 = PA3/VREF 引脚 10 = 内部 3V 电压 01 = 内部 2V 电压 00 = VDD
4:2	CHS[2:0]	模拟通道选择位 000 = AN0 001 = AN1 010 = AN2 011 = AN3 100 = AN4 101 = AN5 110 = AN6 111 = 内部 1/4 VDD
1	GO/DONE	A/D 转换状态位 1 = A/D 转换正在进行。 将本位置 1 启动一次 A/D 转换。A/D 转换完成时此位由硬件自动清零 0 = A/D 转换完成/不在进行中
0	ADON	ADC 使能位 1 = 使能 ADC

		0 = 禁止 ADC，不消耗工作电流
--	--	--------------------

### 12.4.2 ADCON1 寄存器，地址 0x9F

Bit	7	6	5	4	3	2	1	0
Name	DIVS	ADCS2	ADCS1	ADCS0	-	-	-	AN0SEL
Reset	0	0	0	0	-	-	-	0
Type	RW	RW	RW	RW	-	-	-	RW

Bit	Name	Function
7	DIVS	A/D 分频时钟源选择位 1 = 分频时钟选择慢时钟 0 = 分频时钟选择 FOSC
6:4	ADCS[2:0]	A/D 转换时钟选择位 <b>当 DIVS 为 0 时:</b> 000 = $F_{SYS}/2$ 001 = $F_{SYS}/8$ 010 = $F_{SYS}/32$ x11 = FRC (时钟来自内部振荡器, 32KHz 或者 256KHz) 100 = $F_{SYS}/4$ 101 = $F_{SYS}/16$ 110 = $F_{SYS}/64$ <b>当 DIVS 为 1 时:</b> 000 = LFINTOSC /2 001 = LFINTOSC /8 010 = LFINTOSC /32 x11 = FRC (时钟来自内部振荡器, 32KHz 或者 256KHz) 100 = LFINTOSC /4 101 = LFINTOSC /16 110 = LFINTOSC /64
3:1	-	未实现，读 0 软件不要向这些位写 1
0	AN0SEL	ADC 通道 0 选择 0 = 通道 0 选择 PC0 1 = 通道 0 选择 PC1

### 12.4.3 ADRESH 寄存器，地址 0x1E

当 ADFM = 0 时:

Bit	7	6	5	4	3	2	1	0
Name	ADRESH							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	ADRES[9:2]	ADC 结果寄存器位 10 位转换结果的高 8 位

当 ADFM = 1 时:

Bit	7	6	5	4	3	2	1	0
Name	-						ADRESH	
Reset	-						x	x
Type	RO-0						RW	RW

Bit	Name	Function
7:2	-	未实现, 读 0
1:0	ADRES[9:8]	ADC 结果寄存器位 10 位转换结果的高 2 位

#### 12.4.4 ADRESL 寄存器, 地址 0x9E

当 ADFM = 0 时:

Bit	7	6	5	4	3	2	1	0
Name	ADRESL		-					
Reset	x	x	-					
Type	RW	RW	RO-0					

Bit	Name	Function
7:6	ADRES[1:0]	ADC 结果寄存器位 10 位转换结果的低 2 位
5:0	-	未实现, 读 0

当 ADFM = 1 时:

Bit	7	6	5	4	3	2	1	0
Name	ADRESL							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
-----	------	----------

7:0	ADRES[7:0]	ADC 结果寄存器位 10 位转换结果的低 8 位
-----	------------	------------------------------

### 12.4.5 ANSEL 寄存器，地址 0x91

Bit	7	6	5	4	3	2	1	0
Name	ANSEL							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	ANSEL[7:0]	模拟选择位 在 AN<7:0>引脚上分别进行模拟或数字功能的模拟选择 1 = 模拟输入，引脚被分配为模拟输入 0 = 数字 IO，引脚被分配给端口或者特殊功能 <b>注意：当 ADC 配置为采样内部 1/4 VDD 通道时，ANSEL&lt;7&gt;要设置为 1</b>

**注意：**将某引脚设置为模拟输入将自动禁止数字输入电路、弱上拉以及电平变化中断(如果有的话)。相应 TRIS 位必须设置为输入模式以允许对该引脚的电压进行外部控制。

FMD Confidential

## 13 慢时钟测量

芯片集成了两个内部 RC 振荡器，一个是经过出厂校准的高速高精度的 16M 快时钟 HFINTOSC，一个是低速低功耗的 32K 时钟 LFINTOSC，利用慢时钟测量功能可以把 LFINTOSC 的周期用系统时钟计算出来。此功能可以比较精准的测量内部慢时钟周期。

### 13.1 使用方法

在此模式下，TIMER2 的预分频、后分频配置自动变为 1:1，组成一个 12 位的定时器，TIMER2 的计数时钟为系统时钟  $F_{osc}$ ，而不是普通模式下的指令时钟  $F_{osc}/2$ 。计数结束后结果自动存到 SOSCPR 寄存器，其单位是系统时钟  $F_{osc}$  的个数。

#### 操作步骤：

1. 为提高计量精度，建议设置 IRCF 为 111，SCS=1，选择 16M 的系统时钟；
2. 把 T2CON.2 置 1，使能 TIMER2；
3. 如果选择 4 次平均，则把 MSCKCON.2 置 1，否则把它清 0；
4. 置位 MSCKCON.1，开始测量；
5. 测量结束后 MSCKCON.1 自动清 0，中断标志置 1；
6. 可以用查询或中断的方式等待结束；
7. 当查询到中断标志为 1 时读取得到的 SOSCPR 即为最终结果。

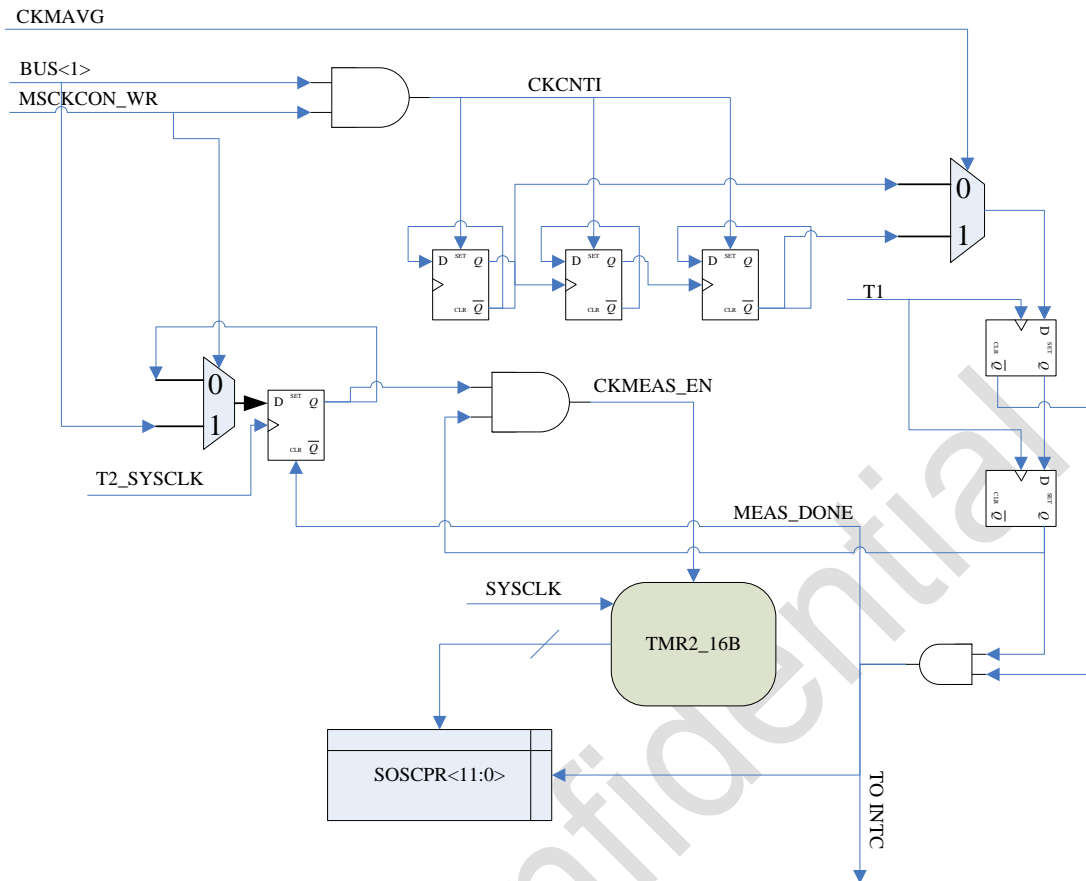


图 13.4 慢时钟测量模式原理框图

注意：

1. 在慢时钟测量过程中软件不要写 **SOSCPRH/L**;
2. 不要在单步调试下做慢时钟测量，因为暂停模式下 **TIMER2** 被停止，这样会导致测量结果不正确；

## 13.2 与慢时钟测量相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
MSCKCON	0x1B	-	-	T2CKSRC	SLVREN	-	CKMAVG	CKCNTI	-	0000 000-
SOSCPRL	0x1C	SOSCPR[7:0]								1111 1111
SOSCPRH	0x1D	-				SOSCPR[11:8]				---- 1111
ANSEL	0x91	ANSEL[7:0], 数模管脚选择								1111 1111
INTCON	0x0B/8B	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
PIE1	0x8C	EEIE	CKMEAIE	-	C5IE	C4IE	OSFIE	TMR2IE	TMR1IE	0000 0000
PIR1	0x0C	EEIF	CKMEAIF	-	C5IF	C4IF	OSFIF	TMR2IF	TMR1IF	0000 0000

### 13.2.1 MSCKCON 寄存器，地址 0x1B

Bit	7	6	5	4	3	2	1	0
Name	-	-	T2CKSRC	SLVREN	-	CKMAVG	CKCNTI	-
Reset	0	0	0	0	0	0	0	-
Type	RW	RW	RW	RW	RW	RW	RW	-

Bit	Name	Function
7:6, 3	-	保留位，不要写 1
5	T2CKSRC	TIMER2 时钟源选择 0 = TIMER2 时钟源为系统时钟 1 = TIMER2 时钟源为内部 32MHz
4	SLVREN	软件控制 LVR 使能位，当 UCFG1<1:0>为 01 时： 1 = 打开 LVR 0 = 禁止 LVR 当 UCFG1<1:0>不为 01 时，此位无实际意义 <b>注意：发生欠压复位时，该位不会清 0。其它任何复位都可将其清 0</b>
2	CKMAVG	快时钟测量慢时钟周期的测量平均模式 1 = 打开平均模式（自动测量并累加 4 次） 0 = 关闭平均模式
1	CKCNTI	Clock Count Init -使能快时钟测量慢时钟周期 1 = 使能快时钟测量慢时钟周期 0 = 关闭快时钟测量慢时钟周期 注：这一位在测量完毕后会自动归零
0	-	保留位，不能写 1

### 13.2.2 SOSCPRL 寄存器，地址 0x1C, 1D

SOSCPRL, 地址 0x1C

Bit	7	6	5	4	3	2	1	0
Name	SOSCPRL[7:0]							
Reset	8'hff							
Type	RW							

SOSCPRH, 地址 0x1D

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	SOSCPRH[11:8]			
Reset	-	-	-	-	4'hf			
Type	-	-	-	-	RW			



Bit	Name	Function
0x1C: 7:0	SOSCPR[11:0]	低速振荡器周期（单位：快时钟周期数）
0x1D: 3:0		用于慢时钟测量功能

FMD Confidential

## 14 中断模式

FT61F04X 有以下中断源：

- PC1/INT 管脚进来的外部中断
- Timer0 溢出中断
- Timer1 溢出中断
- PORTA 电平变化中断
- Timer2 比对相等中断
- EEPROM 数据写中断
- 故障保护时钟监控器中断
- 比较器 0/1/2/3 中断
- TIMER3/4/5/6 中断
- ECCP 相关中断

中断控制寄存器 (INTCON) 和外围中断请求寄存器 (PIR1) 记录了中端标志位。INTCON 同时也包含全局中断使能位 GIE。

当中断被服务后，以下动作自动发生：

- GIE 被清零，从而关闭中断
- 返回地址被推上堆栈
- 程序指针加载 0004h 地址

中断返回指令，RETI 将退出中断函数时同时设置 GIE 位，重新使能未屏蔽的中断。需要注意的是，执行中断返回 RETI 之前应该把相关的中断标志位清 0，以免重复进入中断处理程序。

INTCON 寄存器包含以下中断标志位：

- INT 管脚中断
- PORTA 变化中断
- Timer0 溢出中断

PIR1/2 中包含着外围中断标志位，PIE1/2 中包含着其对应的中断使能位，具体请参照寄存器各位的描述。

### 14.1 INT 中断

INT 引脚中断是边沿触发的：当 OPTION 寄存器的 INTEDG 位被置 1 时在上升沿触发，而当 INTEDG 位被清零时在下沿触发。当 INT 引脚输出上出现有效边沿时，INTCON 寄存器的 INTF 位置 1。可以通过将 INTCON 寄存器的 INTE 控制位清零来禁止该中断。在重新允许该中断前，必须在中断服务程序中先用软件将 INTF 位清零。如果 INTE 位在进入休眠状态前被置 1，则 INT 中断能将处理器从休眠状态唤醒。

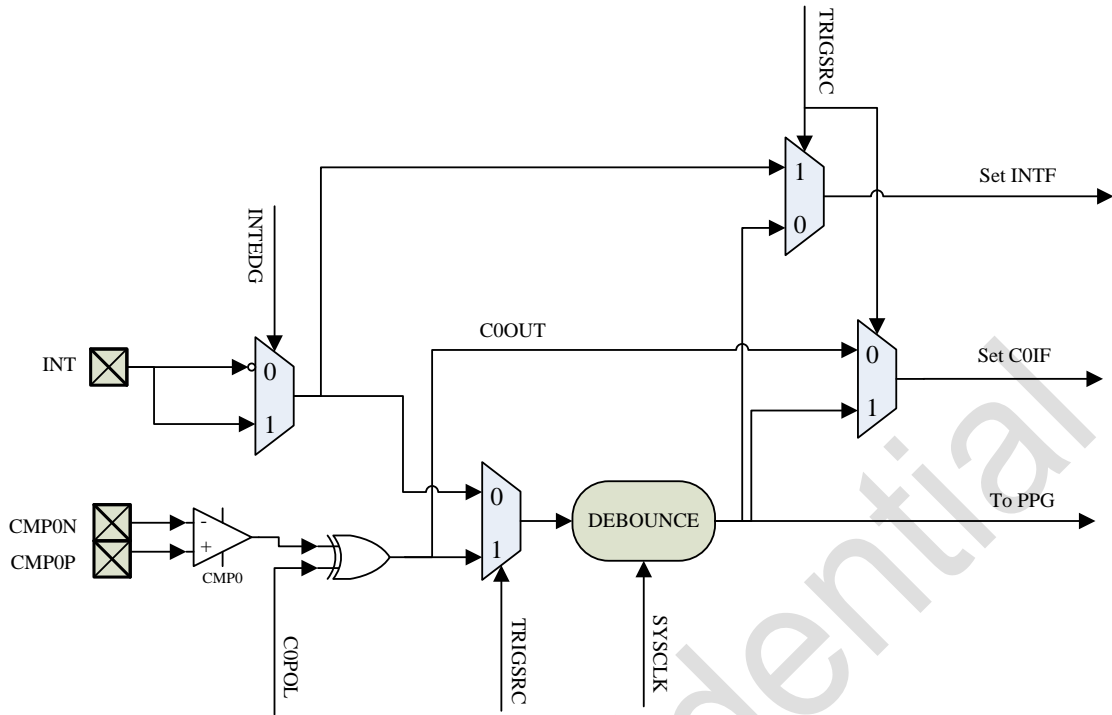


图 14.1 INT 管脚/比较器 0 输出中断

### 14.1.1 硬件去抖

INT 引脚/比较器 0 输出具有硬件去抖功能，最多可过滤 63 个系统时钟周期的毛刺。去抖模块是 INT 和 CMP0 共用的，寄存器 DEBCR 的 TRIGSRC 位决定了去抖对谁有效，当 TRIGSRC=0 时，去抖模块对 INT 引脚作用。修改 DEBDLY=0 可以把去抖功能屏蔽。睡眠时由于系统时钟被停止，所以去抖功能在睡眠模式下也将被屏蔽，但当 PCON 的寄存器位 SYSON 为 1 时，去抖功能在睡眠模式下仍然有效。

**注意：**硬件去抖功能不受 PPGON 的影响，即使 PPGON=0，去抖模块仍然可以工作。

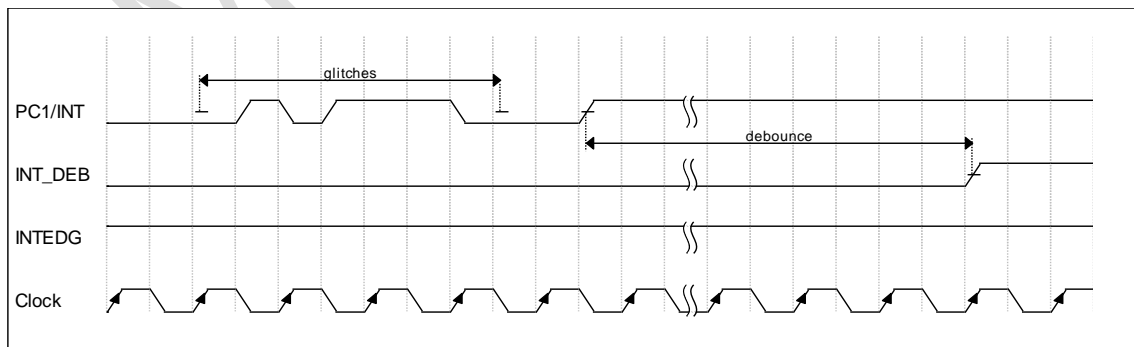


图 14.2 INT 管脚去抖时序

**注意：**

1. 使用 INT 中断时，必须对 ANSEL 和 CMCON0 寄存器进行初始化，以将模拟通道配置为数字输入。否则配置为模拟输入的引脚总是读为 0，INT 的沿中断功能将不能使用。

## 14.2 PORTA 下降沿中断

PORTA 的电平高到低变化可以使 INTCON 寄存器的 PAIF 位置 1。可以通过置 1/清零 INTCON 寄存器的 PAIE 位来使能/禁止该中断。此外，可通过 IOFA 寄存器对该端口的各个引脚进行配置。

注意：

1. 使用 PORTA 下降沿中断时，必须对 ANSEL 和 CMCON0 寄存器进行初始化，以将模拟通道配置为数字输入。配置为模拟输入的引脚总是读为 0；
2. 初始化 PORTA 口的下降沿中断时，应先配置为数字输入 IO (TRISA.x=1)，然后把相应的 IOFA 置 1；

## 14.3 中断响应

外部中断包括 INT 管脚进来的、PORTA 变化中断或者比较器中断的延时一般为 1 到 2 个指令周期。具体延迟取决于中断发生时刻跟当前 CPU 的执行指令（单周期/双周期，Q1/Q2 时刻）。

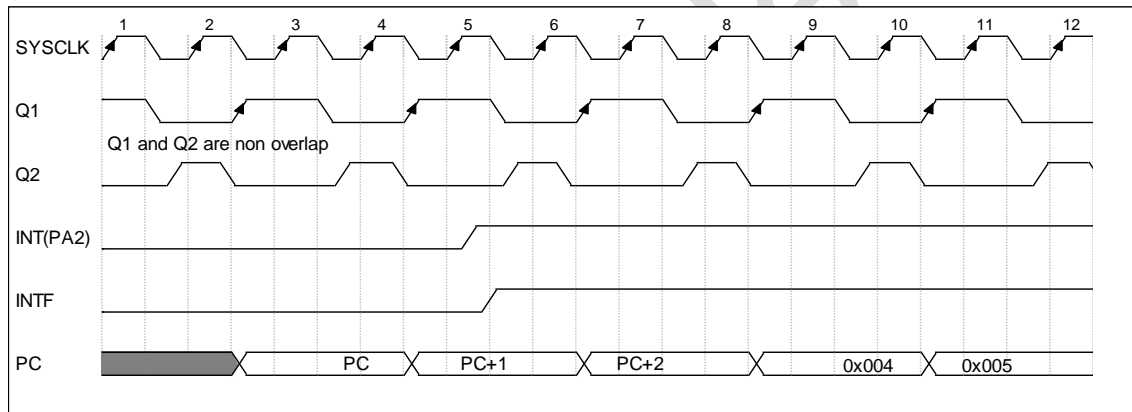


图 14.3 中断响应时序图

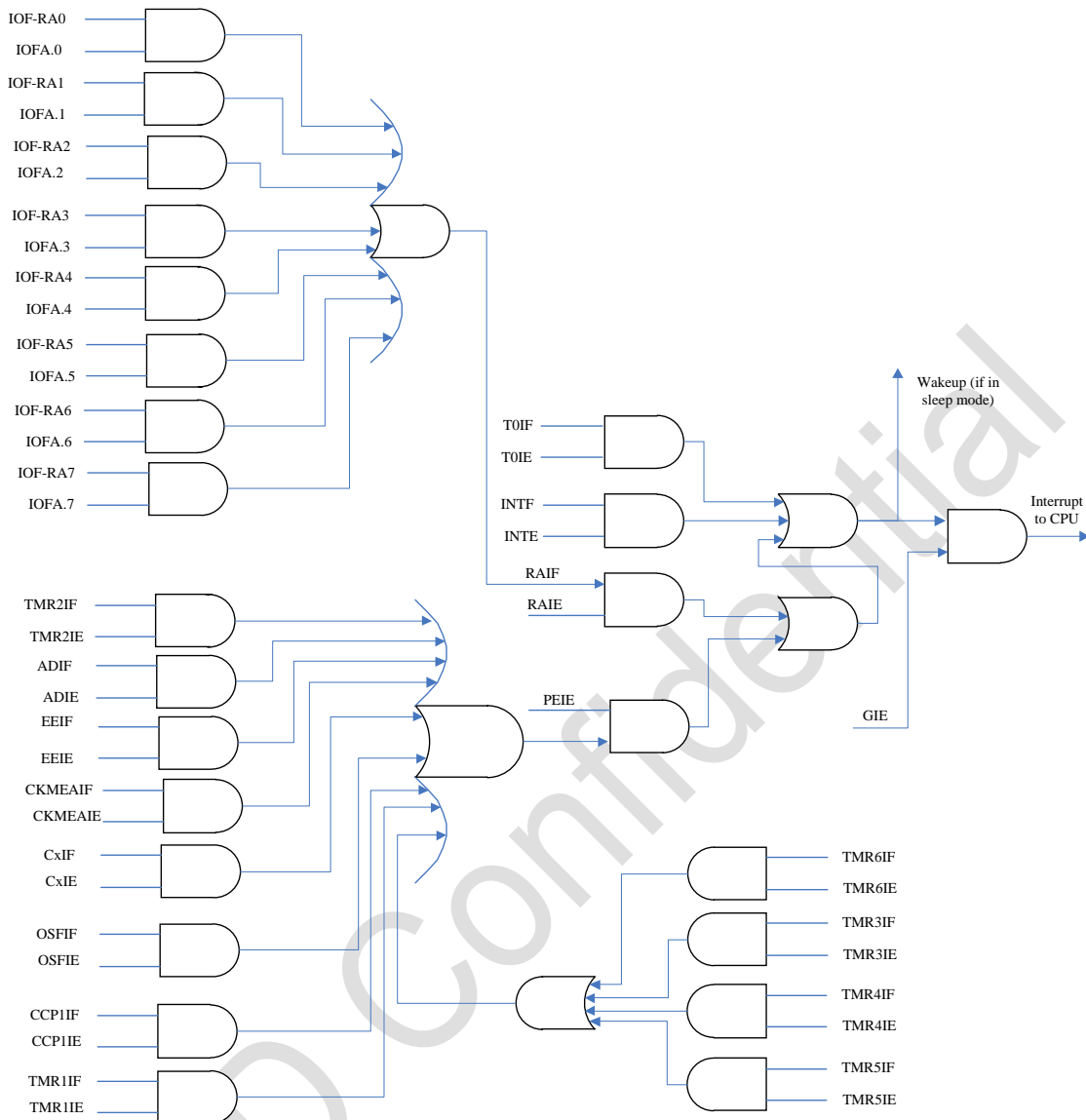


图 14.4 中断产生电路架构框图

## 14.4 中断过程中的现场保存

在中断过程中，只有返回 PC 被自动保存在堆栈上。一般来说，用户可能需要保存重要的寄存器值在堆栈上，例如 W，STATUS 寄存器等。这些必须由软件来完成。临时寄存器 W\_TEMP 和 STATUS\_TEMP 应该被放置在 GPR 的最后 16byte 里。这 16 个 GPR 落在每个数据存储区间 Bank0~Bank3，因此可以稍微节省代码。

## 14.5 与中断相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
INTCON	0x0B/8B	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000

PIE1	0x8C	EEIE	CKMEAIE	-	C5IE	C4IE	OSFIE	TMR2IE	TMR1IE	00-0 0000
PIR1	0x0C	EEIF	CKMEAIF	-	C5IF	C4IF	OSFIF	TMR2IF	TMR1IF	00-0 0000
DEBCR	0x181	OPORS	TRIGSRC	DEBDLY[5:0]						0000 0000
PIE2	0x8D	-	-	C3IE	C2IE	C1IE	C0IE	ADIE	CCP1IE	--00 0000
PIR2	0x0D	-	-	C3IF	C2IF	C1IF	C0IF	ADIF	CCP1IF	--00 0000
TRISA	0x85	TRISA[7:0], PORTA 方向控制								1111 1111
ANSEL	0x91	ANSEL[7:0], 数模管脚选择								1111 1111
IOFA	0x96	IOFA[7:0], 端口下降沿中断允许位								0000 0000

### 14.5.1 INTCON 寄存器，地址 0x0B

Bit	7	6	5	4	3	2	1	0
Name	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	GIE	<b>GIE:</b> 全局中断使能 1 = 允许所有未屏蔽的中断 0 = 禁止所有中断
6	PEIE	<b>PEIE:</b> 外设中断使能 1 = 允许所有未屏蔽的外设中断 0 = 禁止所有外设中断
5	TOIE	<b>TOIE:</b> 定时器0溢出中断使能 1 = 允许Timer0中断 0 = 禁止Timer0中断
4	INTE	<b>INTE:</b> 外部中断使能 1 = 允许PC1/INT外部中断 0 = 禁止PC1/INT外部中断
3	PAIE	<b>PAIE:</b> PORTA电平中断使能位 1 = 允许PORTA电平变化中断 0 = 禁止PORTA电平变化中断
2	TOIF	<b>TOIF:</b> 定时器0溢出中断标志位 1 = Timer0寄存器已经溢出 (必须用软件清零) 0 = Timer0寄存器没有溢出
1	INTF	<b>INTF:</b> PC1/INT外部中断标志位 1 = 发生了PC1/INT外部中断 (必须用软件清零) 0 = 未发生PC1/INT外部中断
0	PAIF	<b>PAIF:</b> PORTA 电平变化中断标志位 1 = 至少一个PORTA<7:0>引脚的电平状态发生了改变 (必须用软件清零) 0 = 没有一个PORTA<7:0>引脚的电平状态发生改变

### 14.5.2 PIR1 寄存器，地址 0x0C

Bit	7	6	5	4	3	2	1	0
Name	EEIF	CKMEAIF	-	C5IF	C4IF	OSFIF	TMR2IF	TMR1IF
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	-	RW	RW	RW	RW	RW

Bit	Name	Function
7	EEIF	<b>EEIF:</b> EE写中断标志位 1 = EE 写操作完成 (必须软件清零) 0 = EE 写操作未完成
6	CKMEAIF	<b>CKMEAIF:</b> 快时钟测量慢时钟操作完成中断标志位 1 = 快时钟测量慢时钟操作完成 (必须软件清零) 0 = 快时钟测量慢时钟未完成
5	-	保留位，不要写1
4	C5IF	比较器 5 中断标志位 1 = 比较器 5 输出发生了变化 0 = 比较器 5 输出未发生改变
3	C4IF	比较器 4 中断标志位 1 = 比较器 4 输出发生了变化 0 = 比较器 4 输出未发生改变
2	OSFIF	振荡器故障中断标志位 1 = 系统振荡器发生故障，时钟输入切换为 INTOSC (必须用软件清零) 0 = 系统时钟运行正常
1	TMR2IF	<b>TMR2IF:</b> Timer2与PR2比较相等中断标志位 1 = timer2的值等于PR2 (必须软件清零) 0 = timer2 的值不等于 PR2
0	TMR1IF	TIMER1 溢出中断标志位 1 = timer1 发生了溢出 0 = timer1 未发生溢出

### 14.5.3 PIR2 寄存器，地址 0x0D

Bit	7	6	5	4	3	2	1	0
Name	-	-	C3IF	C2IF	C1IF	C0IF	ADIF	CCP1IF
Reset	-	-	0	0	0	0	0	0
Type	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	-	未实现，读 0

		软件不要向这些位写 1
5	C3IF	比较器 3 中断标志位 1 = 比较器 3 输出产生了下降沿 0 = 比较器 3 输出没有下降沿
4	C2IF	比较器 2 中断标志位 1 = 比较器 2 输出产生了下降沿 0 = 比较器 2 输出没有下降沿
3	C1IF	比较器 1 中断标志位 1 = 比较器 1 输出产生了下降沿 0 = 比较器 1 输出没有下降沿
2	C0IF	比较器 0 中断标志位 1 = 比较器 0 输出产生了下降沿 0 = 比较器 0 输出没有下降沿
1	ADIF	AD中断标志位 1 = AD转换完成 0 = AD 转换未完成或者没启动
0	CCP1IF	CCP1 中断标志位 捕捉模式： 1 = 发生了 TMR1 寄存器捕捉（必须用软件清 0） 0 = 未发生 TMR1 寄存器捕捉 比较模式： 1 = 发生了 TMR1 寄存器比较匹配（必须用软件清 0） 0 = 未发生 TMR1 寄存器比较匹配 PWM 模式： 此模式下该位未使用

#### 14.5.4 PIE1 寄存器，地址 0x8C

Bit	7	6	5	4	3	2	1	0
Name	EEIE	CKMEAIE	-	C5IE	C4IE	OSFIE	TMR2IE	TMR1IE
Reset	0	0	0	0	0	0	0	0
TYPE	RW	RW	-	RW	RW	RW	RW	RW

Bit	Name	Function
7	EEIE	EE写中断使能位 1 = 使能EE 写操作完成中断 0 = 关闭 EE 写操作完成中断
6	CKMEAIE	快时钟测量慢时钟操作完成中断使能位 1 = 使能快时钟测量慢时钟操作完成中断 0 = 关闭快时钟测量慢时钟操作完成中断
5	-	保留位，不要写1



4	C5IE	比较器 5 中断允许位 1 = 允许比较器 5 中断 0 = 禁止比较器 5 中断
3	C4IE	比较器 4 中断允许位 1 = 允许比较器 4 中断 0 = 禁止比较器 4 中断
2	OSFIE	振荡器故障中断允许位 1 = 允许振荡器故障中断 0 = 禁止振荡器故障中断
1	TMR2IE	Timer2 与PR2比较相等中断使能位 1 = 使能 timer2的值等于PR2 中断 0 = 关闭使能 timer2 的值等于 PR2 中断
0	TMR1IE	Timer1 溢出中断使能位 1 = 允许 Timer1 溢出中断 0 = 禁止 Timer1 溢出中断

### 14.5.5 PIE2 寄存器，地址 0x8D

Bit	7	6	5	4	3	2	1	0
Name	-	-	C3IE	C2IE	C1IE	C0IE	ADIE	CCP1IE
Reset	-	-	0	0	0	0	0	0
TYPE	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	-	未实现，读 0 软件不要向这些位写 1
5	C3IE	比较器 3 中断使能位 1 = 允许比较器 3 中断 0 = 禁止比较器 3 中断
4	C2IE	比较器 2 中断使能位 1 = 允许比较器 2 中断 0 = 禁止比较器 2 中断
3	C1IE	比较器 1 中断使能位 1 = 允许比较器 1 中断 0 = 禁止比较器 1 中断
2	C0IE	比较器 0 中断使能位 1 = 允许比较器 0 中断 0 = 禁止比较器 0 中断
1	ADIE	A/D转换器中断使能位 1 = 允许A/D转换器中断 0 = 禁止 A/D 转换器中断

0	CCP1IE	CCP1 中断使能位 1 = 允许 CCP1 溢出中断 0 = 禁止 CCP1 溢出中断
---	--------	--

### 14.5.6 IOFA 寄存器，地址 0x96

Bit	7	6	5	4	3	2	1	0
Name	IOFA[7:0]							
Reset	8'h00							
Type	RW							

Bit	Name	Function
7:0	IOFA[7:0]	端口下降沿中断设置 1 = 使能端口下降沿中断 0 = 关闭端口下降沿中断

### 14.5.7 DEBCR 寄存器，地址 0x181

Bit	7	6	5	4	3	2	1	0
Name	OP0RS	TRIGSRC	DEBDLY[5:0]					
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function								
7	OP0RS	运放 0 输入失调电压选择位 0 = 选择反相端作为参考输入脚 1 = 选择同相端作为参考输入脚								
6	TRIGSRC	PPG 触发源选择 1 = 选择比较器 0 输出为 PPG 触发源 0 = 选择 PC1/INT 管脚为 PPG 触发源								
5:0	DEBDLY[5:0]	INT 引脚/比较器 0 输出去抖时间选择 0 = 禁止硬件去抖功能 其它值 = 去抖时间为 (DEBDLY-1~DEBDLY) 个系统时钟周期 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>DEBDLY 值</th> <th>去抖时间</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0~1 个 <math>T_{SYS}</math></td> </tr> <tr> <td>2</td> <td>1~2 个 <math>T_{SYS}</math></td> </tr> <tr> <td>63</td> <td>62~63 个 <math>T_{SYS}</math></td> </tr> </tbody> </table>	DEBDLY 值	去抖时间	1	0~1 个 $T_{SYS}$	2	1~2 个 $T_{SYS}$	63	62~63 个 $T_{SYS}$
DEBDLY 值	去抖时间									
1	0~1 个 $T_{SYS}$									
2	1~2 个 $T_{SYS}$									
63	62~63 个 $T_{SYS}$									

## 15 睡眠省电模式

芯片在执行完 SLEEP 指令后进入睡眠状态，CPU 停止取指。

为了达到最低睡眠功耗，软件应该将所有 IO 置高或低，而且没有外部电路从 IO 耗电。I/O 作为输入的，外部电路应将其拉高或拉低，避免翻转耗电。/MCLR 应该在高电平。

### 15.1 唤醒模式

以下事件可以唤醒芯片：

- /MCLR 管脚上有外部复位
- WDT 超时
- PC1/INT 管脚上有中断，PORTA 下降沿中断
- 比较器中断
- ADC 转换完成
- DROM 写完成

清看门狗指令 CLRWDI、SLEEP（进入睡眠模式）或者从睡眠模式唤醒，都将清除看门狗计数器。

### 15.2 看门狗唤醒

看门狗工作在内部慢时钟（32KHz），它是一个 16 位的计数器，和定时器 0 共用一个 8 位的预分频器，使能位位于配置寄存器 UCFG0 的第 3 位，WDTEN，为 1 时表示使能看门狗，为 0 时将由 SWDTEN 位决定使能与否，SWDTEN 位于 WDTCN 寄存器。

清看门狗指令 CLRWDI、SLEEP 会清除看门狗计数器。

在使能了看门狗的情况下，MCU 睡眠时看门狗溢出事件可以作为一个唤醒源，而 MCU 正常工作时作为一个复位源。

### 15.3 睡眠状态下运行系统时钟

当 PCON 寄存器的 SYSON 为 1 时，即使 MCU 进入睡眠模式，系统时钟还可以保持运行，这种特性对于那些需要外设工作（如 PWM 或 PPG）但又不用 CPU 执行指令的应用十分有用。

工作状态	指令时钟	系统时钟
正常模式	运行	运行
睡眠模式，SYSON=0	关闭	关闭
睡眠模式，SYSON=1	关闭	运行

表 15.1 MCU 的各种工作状态

## 16 I/O 端口

本芯片共包含 16 个 GPIO。这些 IO 除了作为普通输入/输出端口以外还通常具备一些与内核周边电路通讯的功能，具体见下。

### 16.1 PORTA 端口和 TRISA 寄存器

PORTA 是一个 8 位双向端口。与其相应的进出方向寄存器就是 TRISA 寄存器。反之，将某一位设置为“0”会将该对应 PORTA 端口设置为输出端口。在置为输出端口时，输出驱动电路会被打开，输出寄存器里的数据会被放置到输出端口。当 IO 处于输入状态时（TRISA=1），在 PORTA 上进行读动作时，PORTA 内容会是反映输入端口的状态。在 PORTA 上进行写动作时，PORTA 内容会被写入输出寄存器。所有的写操作都是“读-更改-写”这样一个微流程，即数据被读，然后更改，再写入输出寄存器的过程。

当 MCLRE 为 1 时，PORTA[1]读的值为 0，此时它是作为外部复位管脚。

### 16.2 端口的其他功能

芯片在 PORTA 的每个端口都有一个状态变化中断选项和弱上拉选项。

#### 16.2.1 弱上拉

PORTA 和 PORTC 的每个端口都有一个可以单独设置的内部弱上拉功能。控制 WPUAx 寄存器里的位就可使能或关断这些弱上拉电路。当 GPIO 被设置为输出时，这些弱上拉电路会被自动关断。弱上拉电路在上电复位期间被置为关断。PORTA[1]内部弱上拉有点不一样，它同时受配置寄存器的 MCLRE 位和 WPUA[1]位控制，当配置为外部复位脚 MCLR B 时，上拉自动使能，而当 PORTA[1]被设置为 GPIO 时，该弱上拉电路由 WPUA[1]控制。

#### 16.2.2 弱下拉

PORTA[4]和 PORTC[3:1]这 4 个 IO 配置为数字输入管脚时具有内部弱下拉功能，由寄存器 WPD 控制。当弱上拉和弱下拉同时在这些 IO 打开时，弱下拉将被禁止，弱上拉起作用。

#### 16.2.3 ANSEL 寄存器

ANSEL 寄存器用于控制 IO 的数模输入，当 ANSEL.x 为 1 时，对应的 IO 口为模拟引脚，IO 的输入上拉自动禁止，软件读该 IO 返回的是 0。

ANSEL 位对数字输出驱动没有影响，换言之，TRIS 位的优先级更高，即当 TRIS.x 为 0 时，不管

ANSEL.x 是 0 还是 1，对应的 IO 为数字输出 IO。要想配置真正的模拟管脚，TRIS.x 要置 1，把数字输出驱动关闭。

### 16.2.4 下降沿中断

PORTA 的每个端口都可以被单独设置成一个中断源，每个中断源都由下降沿触发。控制 IOFA 寄存器里的位就可使能或关断这些端口的中断功能。PORTA 的下降沿中断的功能在上电复位时是无效的，这是因为 IOFA 复位值为 0。

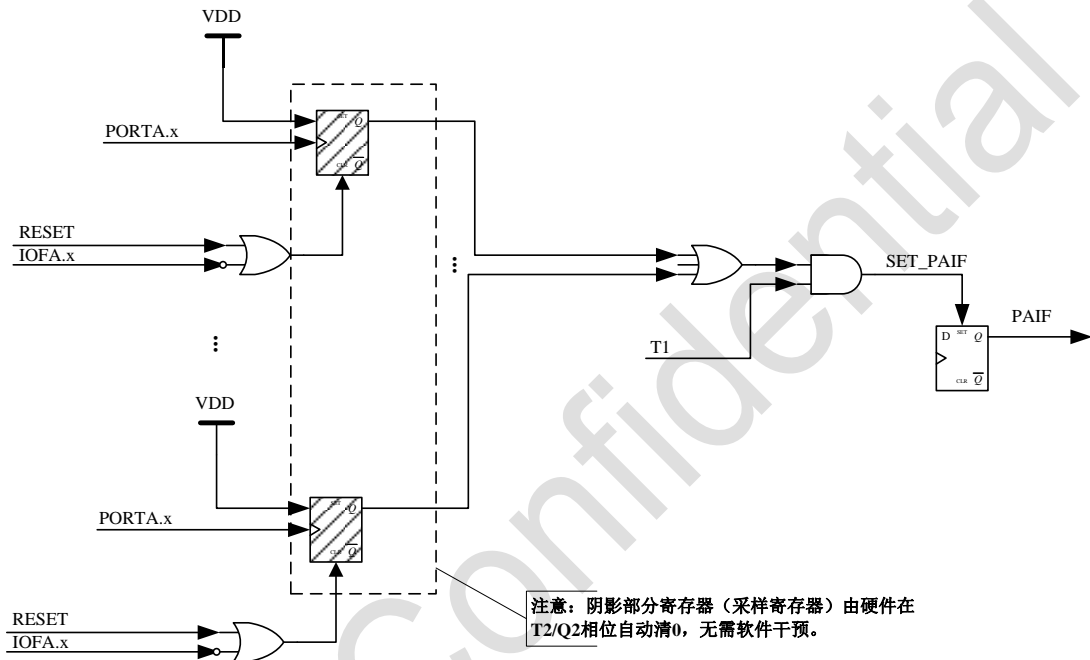


图 16.2.1 PORTA 下降沿中断原理框图

由上图可以知道，所有下降沿中断都或在一起形成 INTCON 寄存器中的 PAIF 标志位。当 PAIF 为 1 时，该中断可以将芯片从睡眠状态中唤醒。该标志位需要软件清除，以免 CPU 反复执行中断。

### 16.3 数字输出在管脚上的复用

由脚位图（图 1.3）可以知道，所有管脚除了可以作通用 IO 功能外，还复用了第 2、第 3 功能甚至更多功能，各管脚的复用情况不尽相同。

作为数字输入功能时，一个管脚的可以为多个功能模块提供信号输入路径，如管脚 2，可以同时作为 PC1 输入和外部中断，又如管脚 3，可以同时作为 PC0 输入和定时器 0 外部时钟；但一个管脚不能在同一时刻为多个功能模块提供输出，下面的表格将对各管脚的数字输出复用作出说明，各寄存器位可以在本文档的其它章节找到更详细的描述。**需要注意的是，当管脚配置为模拟口时，数字输入将被钳位到地。**另外，数字功能输出的条件为 TRISx=0。

注意：

PC3 的 PPG 输出和 PC4 的 IOFF 输出拥有比其他复用功能更高的优先级，即配置选项的 PPGOEB=0 和 IOEB=0 时，这两个 PAD 就输出 PPG 和 IOFF 信号。

管脚功能	数字功能输出	寄存器设置
PC3/P1E/PWM5N/PPG	PC3	PPGOEB=1 & P1EOE=0 & P5IOE=0
	PWM5N	PPGOEB=1 & P5IOE=1 & P5EN=1 & (P1EOE=0 或 ECCP 非半桥模式)
	P1E	PPGOEB=1 & P1EOE=1 & ECCP 半桥模式
	PPG	PPGOEB=0
PA2/P1D	PA2	ECCP 不处于全桥模式
	P1D	ECCP 处于全桥模式
PA5/P1C	PA5	ECCP 不处于全桥模式，且 DAC2OE.2=0
	P1C	ECCP 处于全桥模式，且 DAC2OE.2=0
PC4/P1F/PWM5P/IOFF	PC4	IOEB=1 & P5EN≠1 & (P1FOE=0 或 ECCP 非半桥模式)
	PWM5P	IOEB=1 & P5EN=1 & (P1FOE=0 或 ECCP 非半桥模式)
	P1F	IOEB=1 & P1FOE=1 & ECCP 半桥模式
	IOFF	IOEB=0
PB0/PWM6N	PB0	P6IOE=0   P6EN=0
	PWM6N	P6IOE=1 & P6EN=1
PB1/PCK/PWM6P	PB1	PCKOE≠1   P6EN≠1
	PCK	PCKOE=1
	PWM6P	PCKOE≠1, P6EN=1
PB2/P1B/C4OUT	PB2	CM[2:0]≠110, ECCP 不处于 PWM 半桥或者全桥模式
	P1B	CM[2:0]≠110, ECCP 处于 PWM 半桥或者全桥模式
	C4OUT	CM[2:0]=110, ECCP 不处于 PWM 半桥或者全桥模式
PB3/P1A/C5OUT	PB3	CM[2:0]≠110, ECCP 不处于 PWM 模式
	P1A	CM[2:0]≠110, ECCP 处于 PWM 模式
	C5OUT	CM[2:0]=110, ECCP 不处于 PWM 模式
PA3/PWM4P	PA3	P4EN≠1
	PWM4P	P4EN=1
PA1/PWM4N	PA1	P4EN=0   P4IOE=0, 且 DAC2OE.1=0
	PWM4N	P4EN=1 & P4IOE=1, 且 DAC2OE.1=0
PC2/PWM3P	PC2	P3EN≠1, 且 DAC1OE.1=0
	PWM3P	P3EN=1, 且 DAC1OE.1=0
PC5/PWM3N	PC5	P3EN=0   P3IOE=0, 且 DAC1OE.0=0
	PWM3N	P3EN=1 & P3IOE=1, 且 DAC1OE.0=0

## 16.4 模拟功能管脚的使用

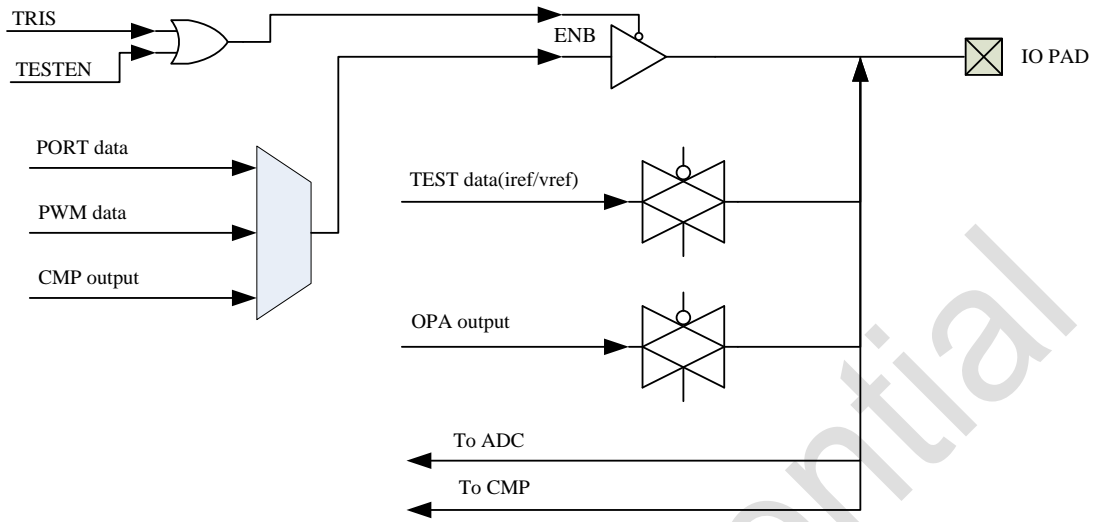


图 16.4.1 模拟与数字功能的复用示意图

大部分管脚兼有模拟 IO 和数字 IO 功能（如 PB0、PB1 等），由上图可以知道，要使管脚成为模拟功能管脚，除了模拟模块的相关模式寄存器及其使能位正确设置之外，还需要软件把该管脚的 TRIS 位写 1 以关闭数字输出驱动。

需要提出的是，管脚如果配置成**模拟测试脚或者模拟输出（DAC 输出或者运放）管脚**，则硬件自动把数字输出关闭，不管三态控制 TRIS 位为何值。

## 16.5 端口描述

GPIO 的每个端口都包含着不同的复用功能，其具体功能和控制在这一节里描述。

### 16.5.1 PORTA[0]

图 16.5.1 描述了此端口的内部电路结构。PA[0]可以被配置为以下功能端口：

- GPIO
- 调试串口时钟
- 比较器 4 同相输入
- 运放 0 输出

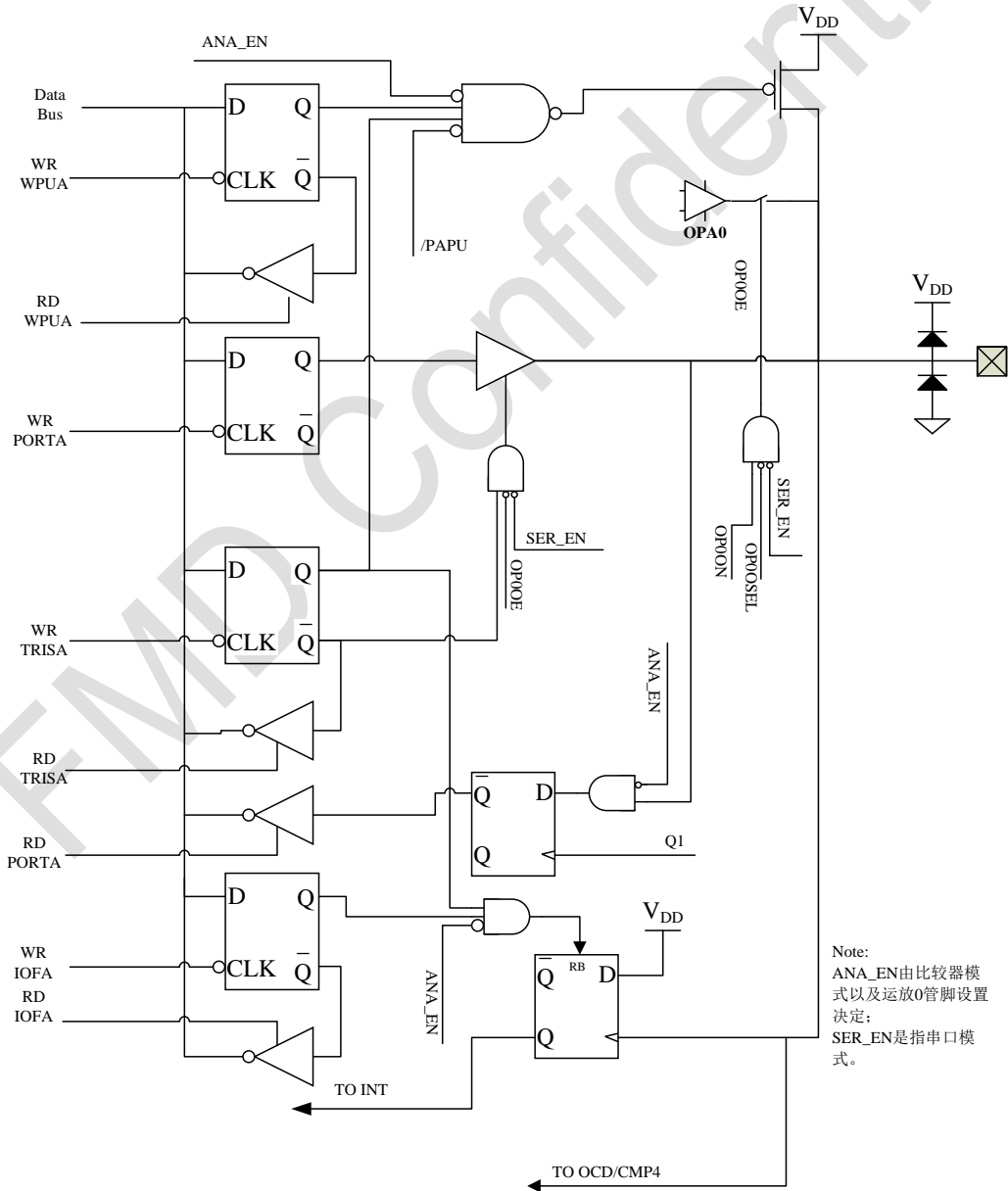




图 16.5.1 PA0 架构框图

## 16.5.2 PORTA[1]

图 16.5.2 描述了此端口的内部电路结构。PA1 可以被配置为以下功能端口：

- GPIO
- 调试串口数据
- 外部复位输入
- PWM4 互补输出
- USB 口的 D-

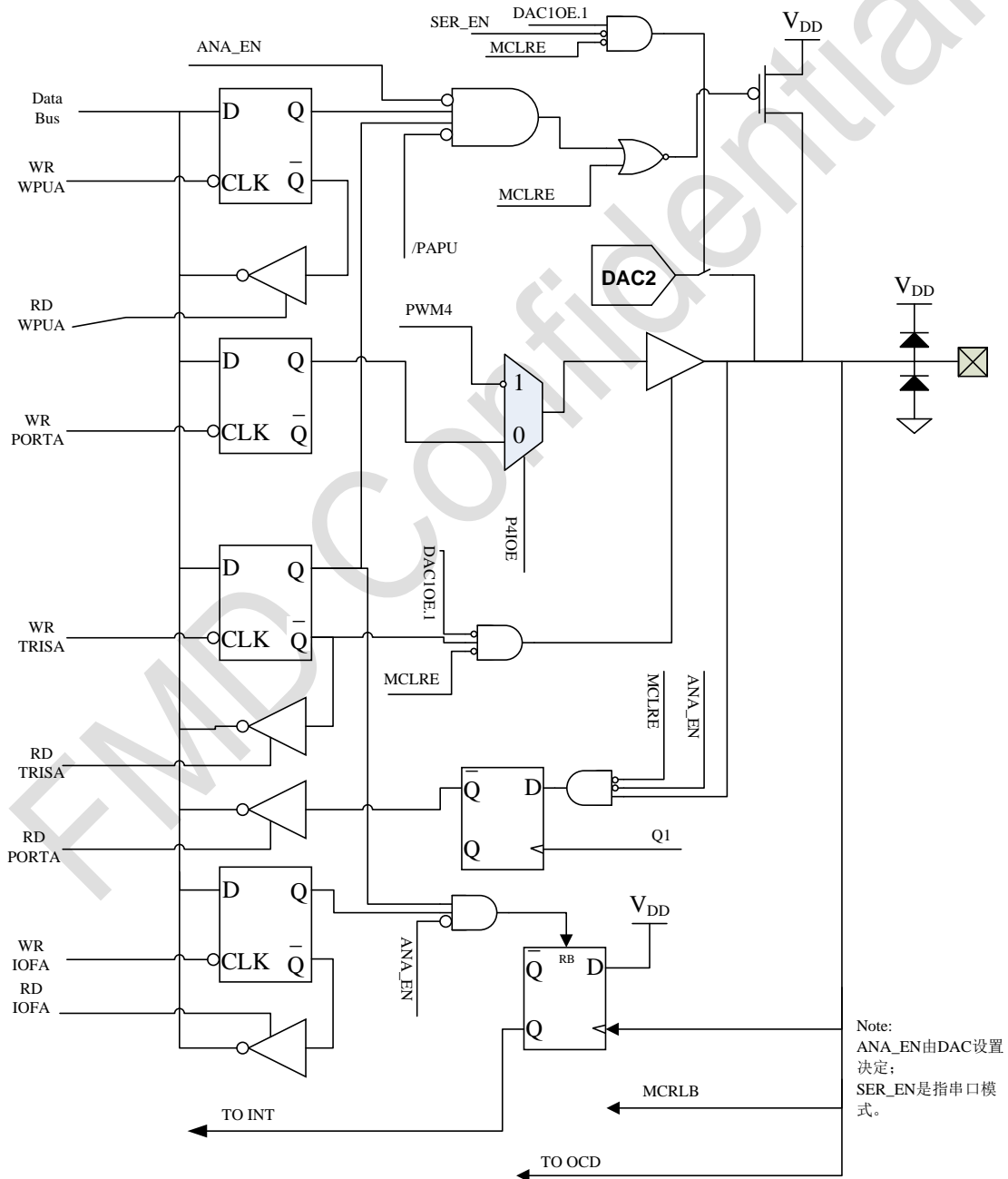


图 16.5.2 PA1 架构框图

### 16.5.3 PORTA[2]

图 16.5.3 描述了此端口的内部电路结构。PA[2]可以被配置为以下功能端口：

- GPIO
- 比较器 2 反相输入
- 增强型 PWM 输出

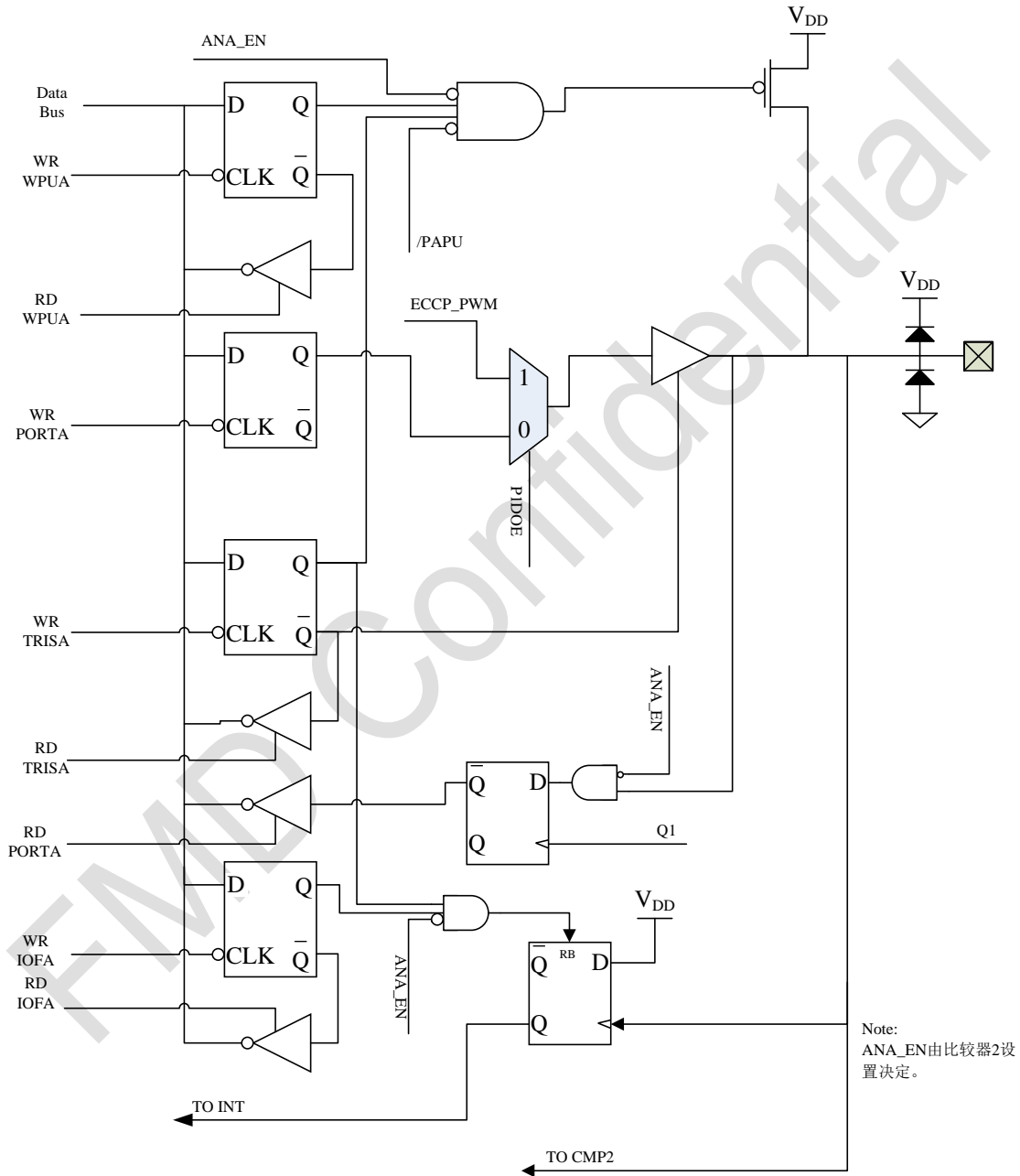


图 16.5.3 PA2 架构框图

### 16.5.4 PORTA[3]

图 16.5.4 描述了此端口的内部电路结构。PA[3]可以被配置为以下功能端口：

- GPIO
- 模拟测试 IO
- PWM4 输出
- ADC 外部参考输入
- 比较器 4 反相输入

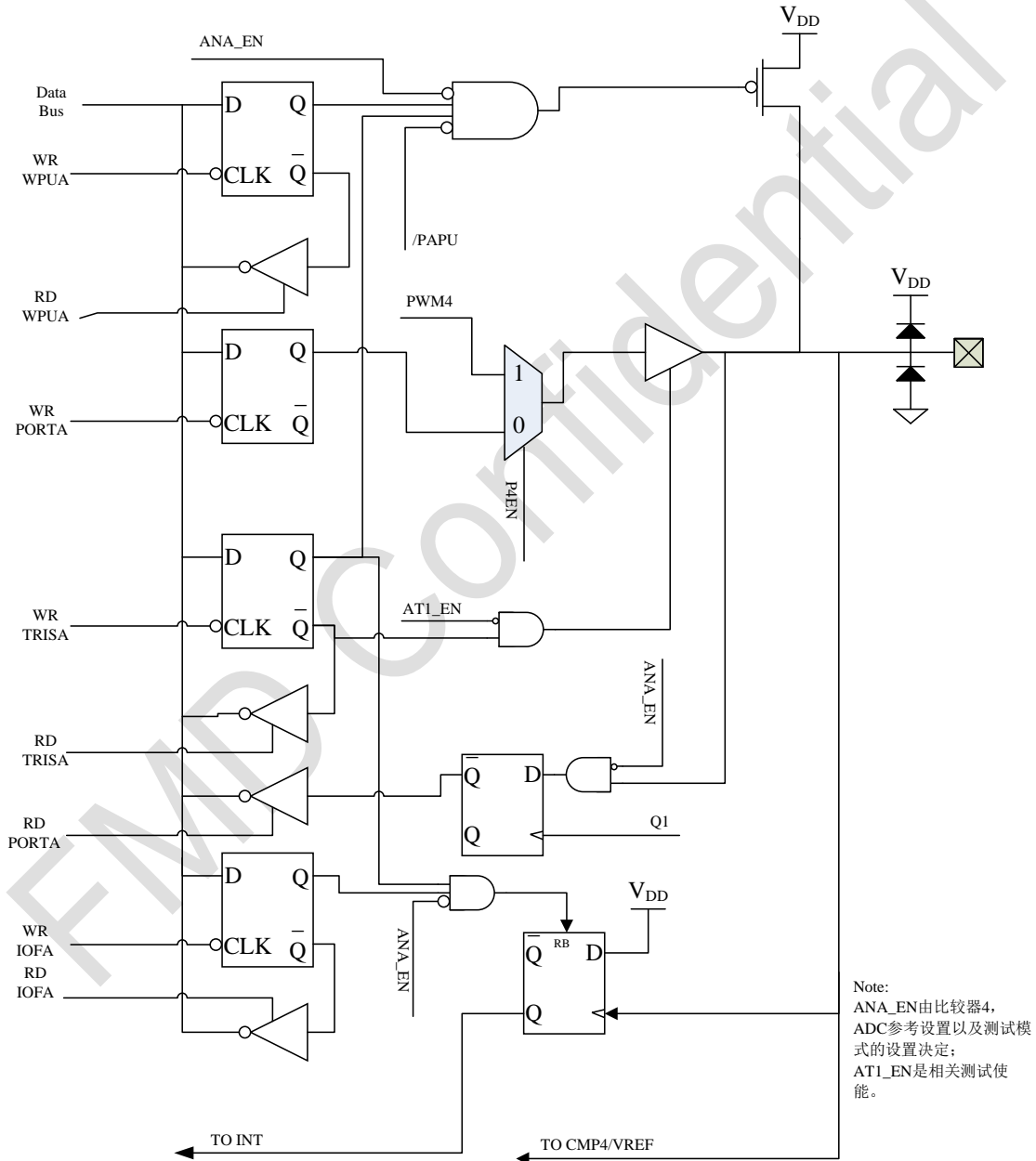


图 16.5.4 PA3 架构框图

## 16.5.5 PORTA[4]

图 16.5.5 描述了此端口的内部电路结构。PA[4]可以被配置为以下功能端口：

- GPIO
- 模拟测试 IO
- USBC 口的 D-
- 比较器 2 同相输入
- ADC 通道 6 输入

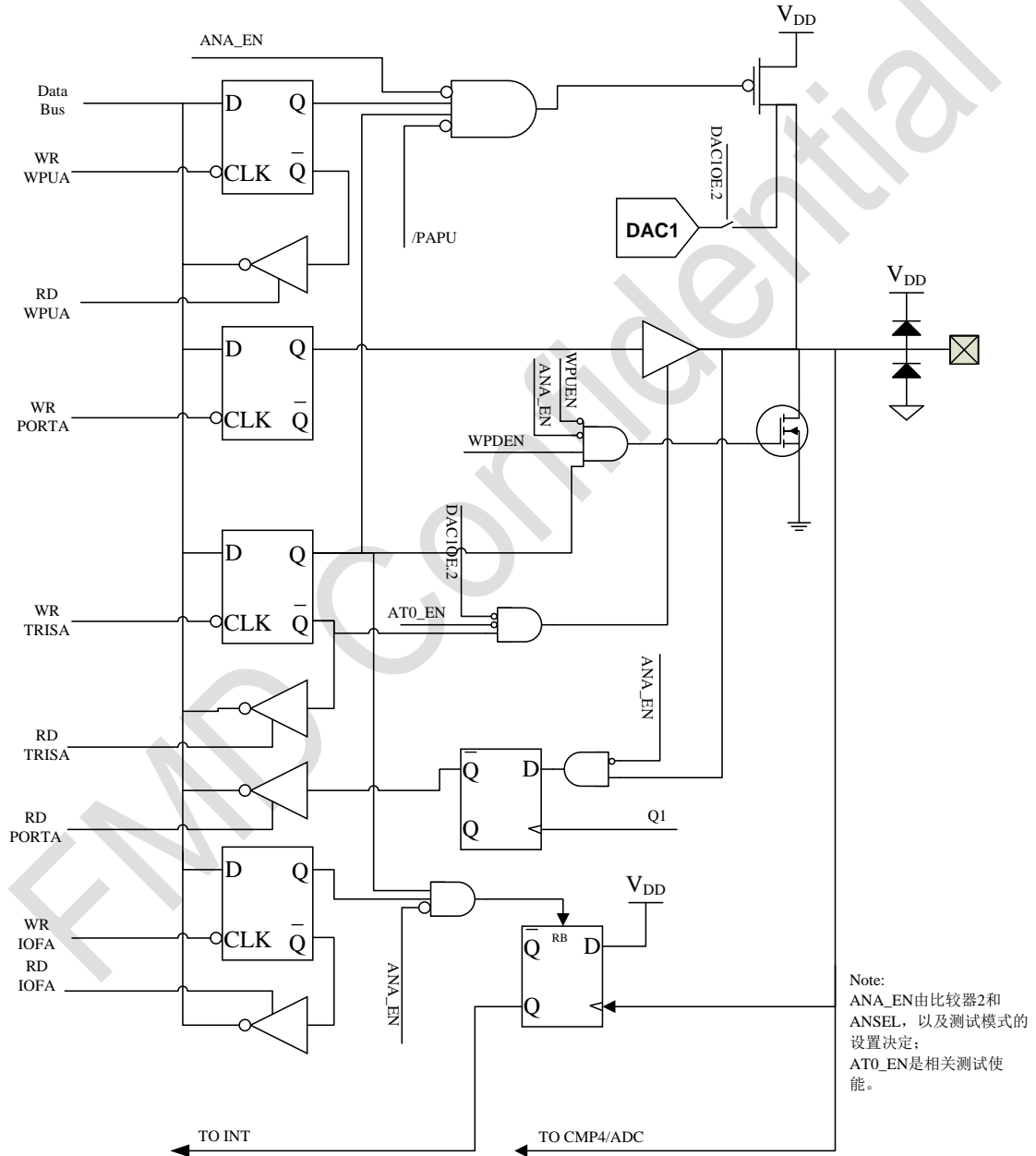


图 16.5.5 PA4 架构框图

### 16.5.6 PORTA[5]

图 16.5.6 描述了此端口的内部电路结构。PA[5]可以被配置为以下功能端口：

- GPIO
- USBC 口的 D+
- 比较器 1 反相输入
- 增强型 PWM 输出
- ADC 通道 5 输入

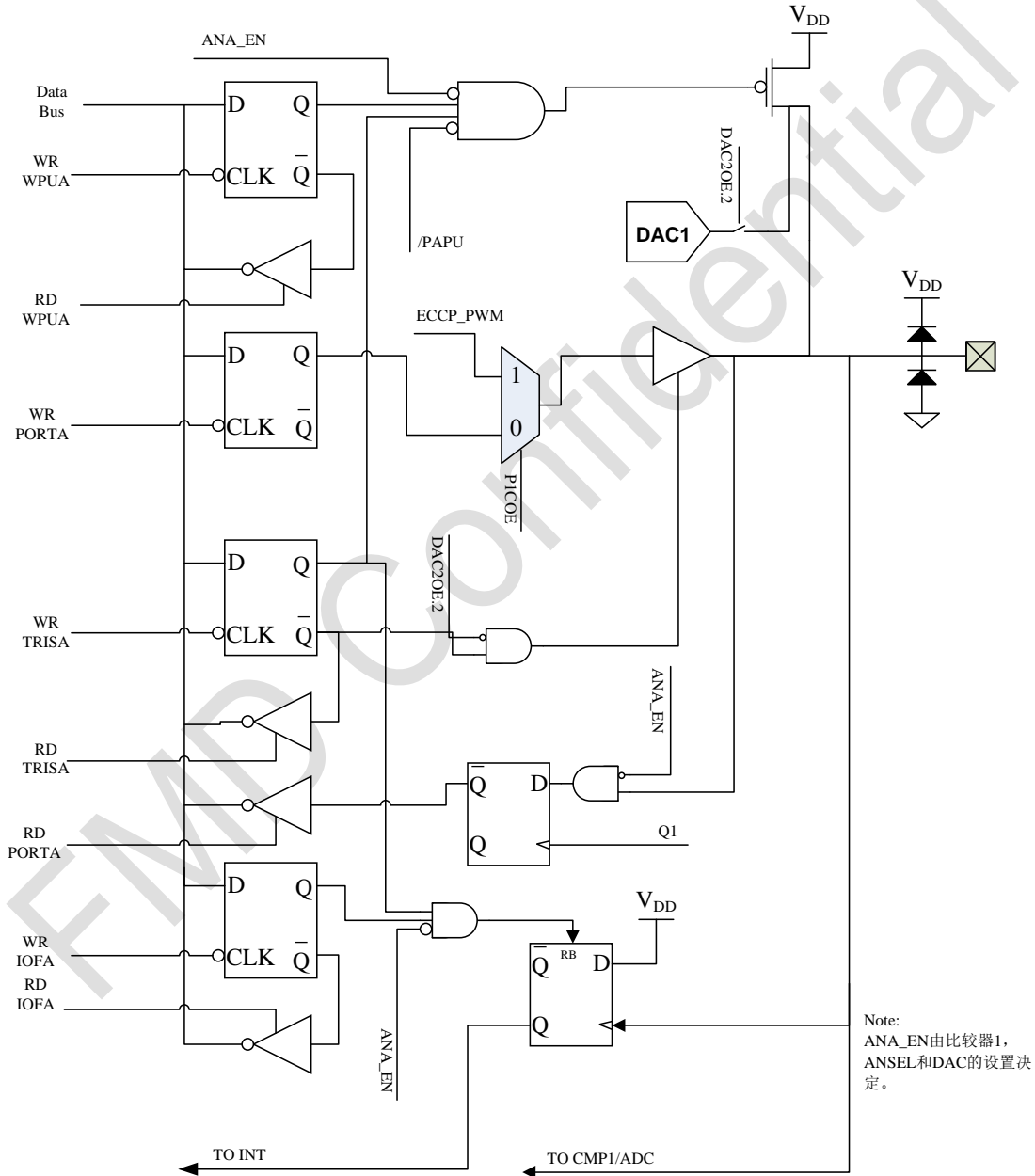


图 16.5.6 PA5 架构框图

## 16.5.7 PORTA[6]

图 16.5.7 描述了此端口内部电路结构。PA[6]可以配置为以下功能端口：

- GPIO
- TIMER1 门控输入
- 时钟输出
- 晶振、谐振器连接
- 运放 0 反相输入

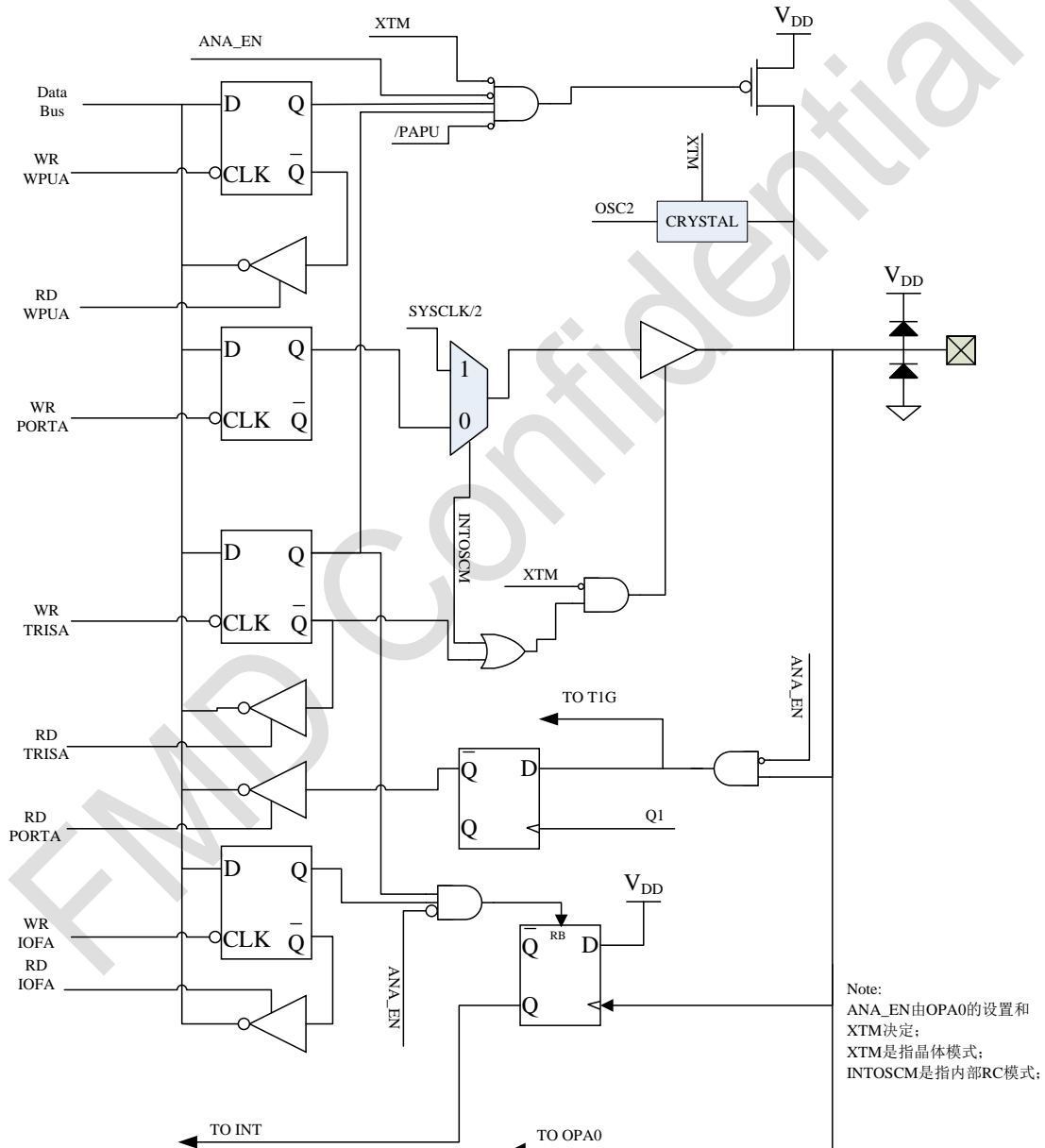


图 16.5.7 PA[6]架构框图

### 16.5.8 PORTA[7]

图 16.5.8 描述了此端口内部电路结构。PA[7]可以配置为以下功能端口：

- GPIO
- 外部时钟输入
- TIMER1 外部时钟源输入
- ADC 通道 3 输入
- 运放 0 输出
- 晶振、谐振器连接
- USB 口的 D+

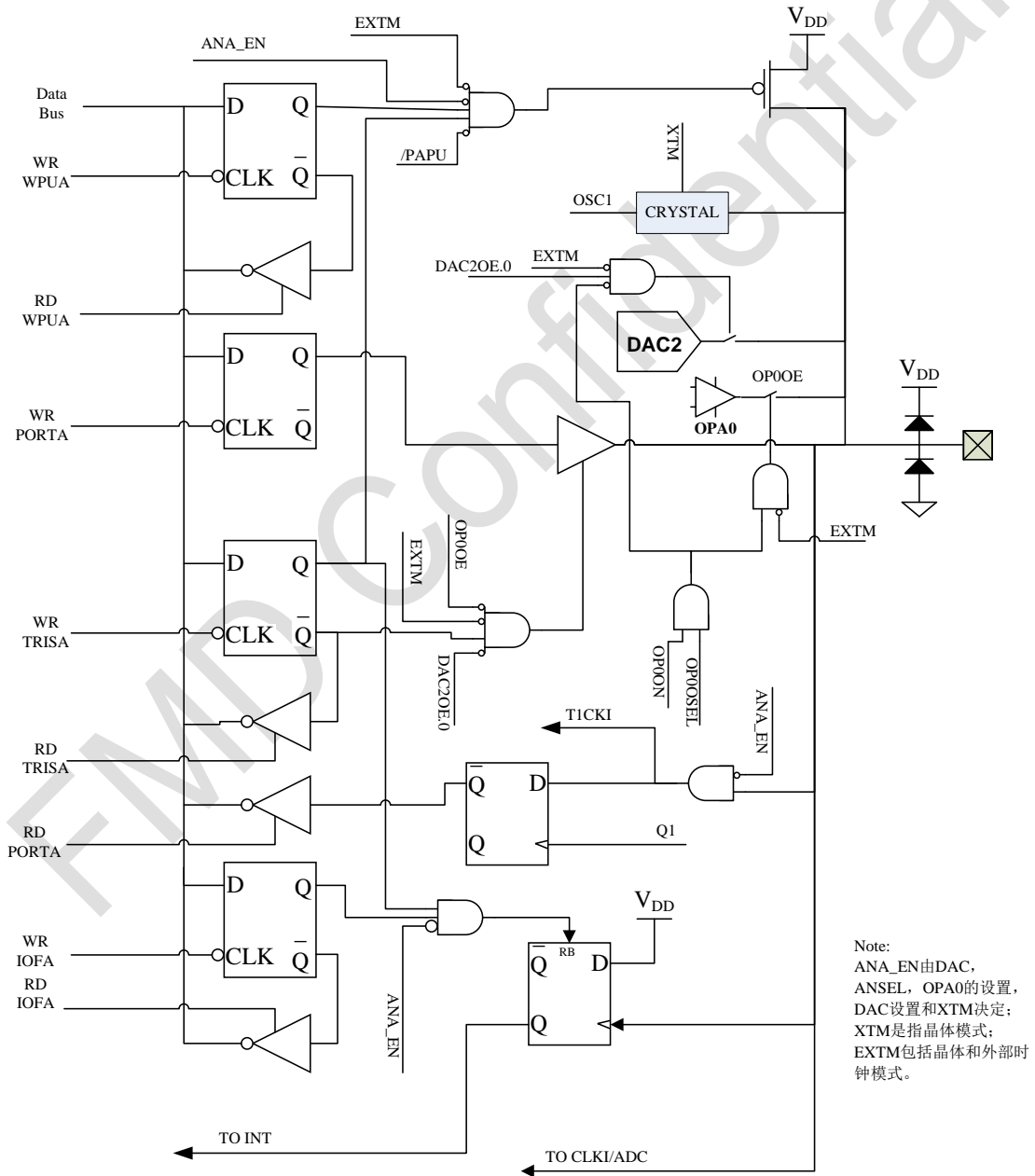


图 16.5.8 PA[7]架构框图

### 16.5.9 PORTC[0]

图 16.5.9 描述了此端口的内部电路结构。PORTC[0]可以被配置为以下功能端口：

- GPIO
- 比较器 0 同相输入
- TIMER0 外部时钟输入
- ADC 通道 0 输入

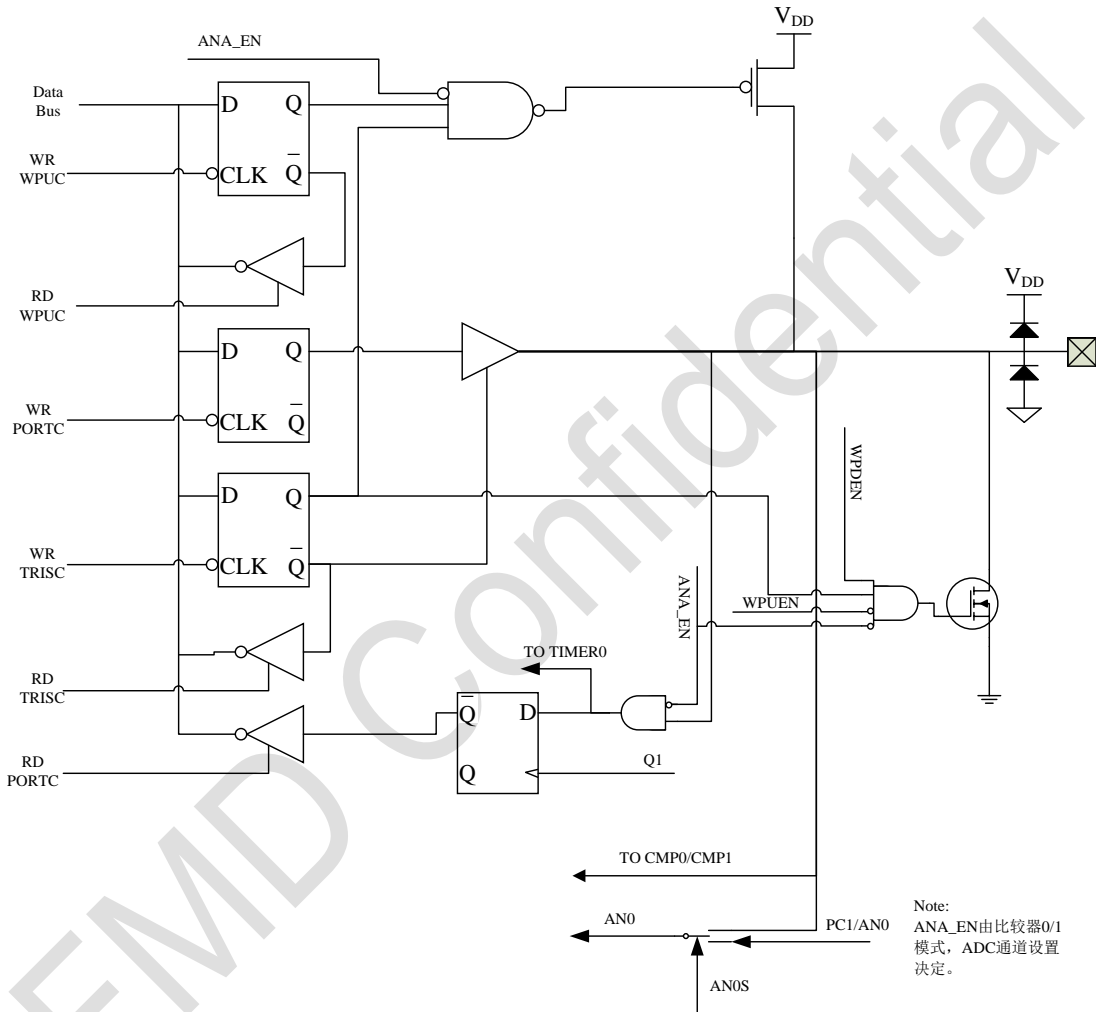


图 16.5.9 PC0 架构框图

### 16.5.10 PORTC[1]

图 16.5.10 描述了此端口的内部电路结构。PORTC[1]可以被配置为以下功能端口：

- GPIO
- 比较器 0 反相输入
- 外部中断输入
- ADC 通道 0 输入



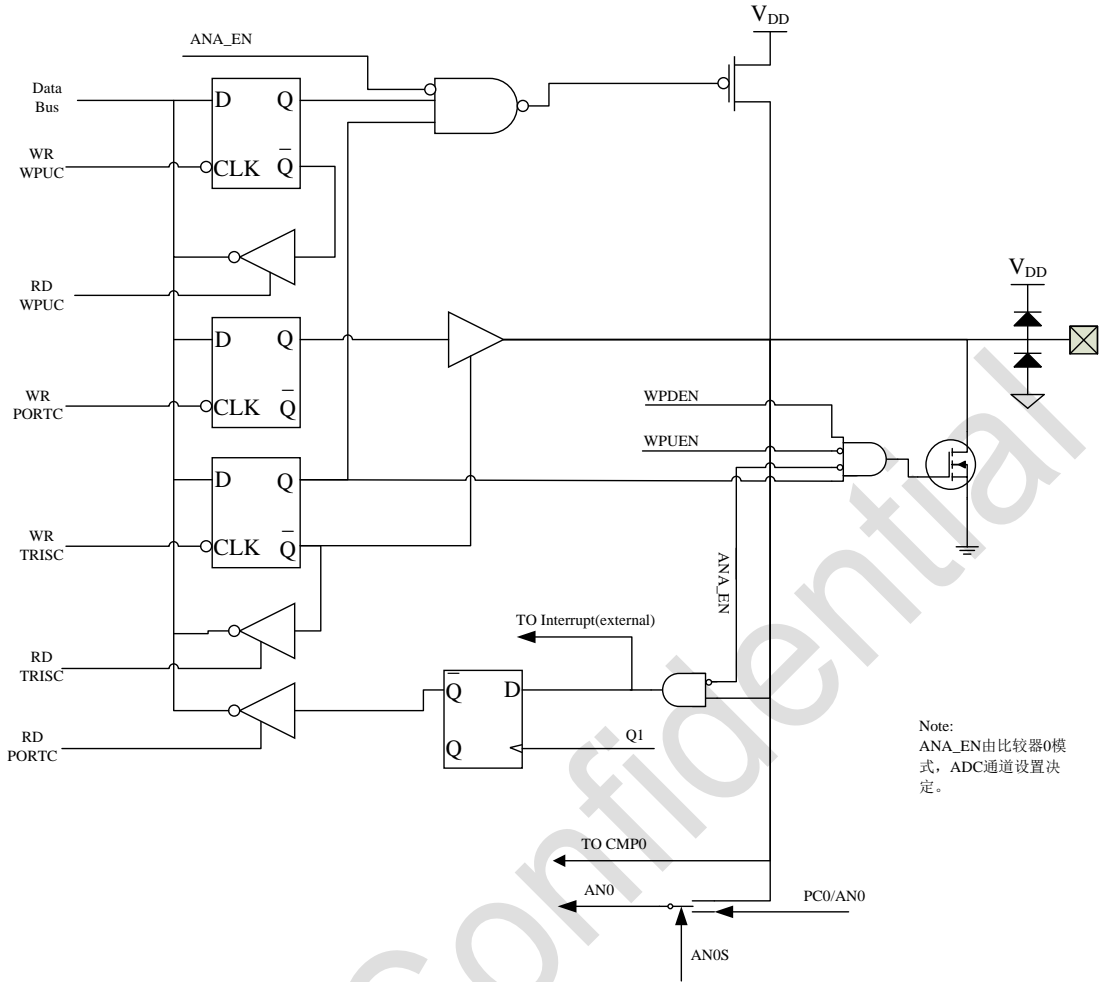


图 16.5.10 PC1 架构框图

### 16.5.11 PORTC[2]

图 16.5.11 描述了此端口的内部电路结构。PORTC[2]可以被配置为以下功能端口：

- GPIO
- PWM3 输出
- 比较器 5 同相输入
- USB 口的 D+

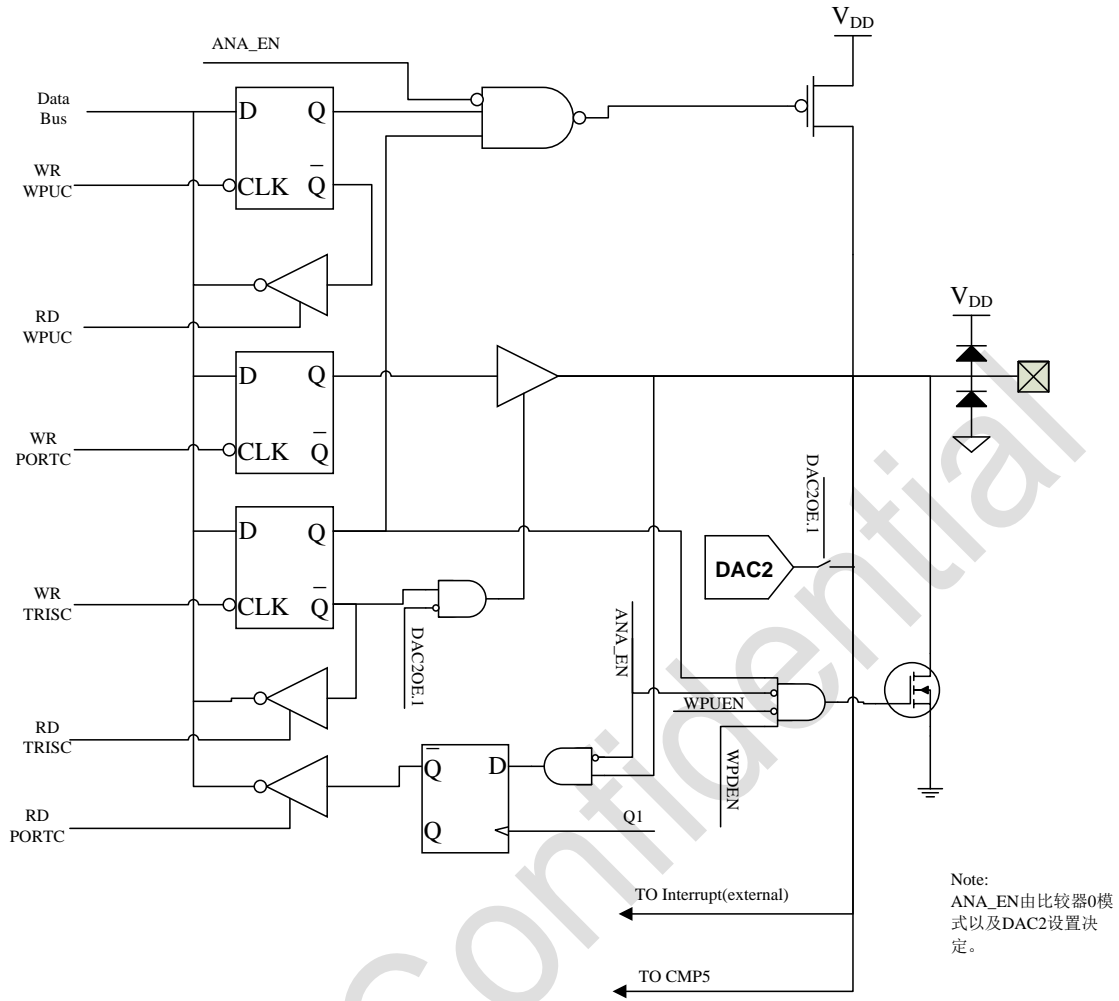


图 16.5.11 PC2 架构框图

### 16.5.12 PORTC[3]

图 16.5.12 描述了此端口的内部电路结构。PORTC[3]可以被配置为以下功能端口：

- GPIO
- 增强型 PWM 输出
- PWM5 互补输出
- PPG 输出

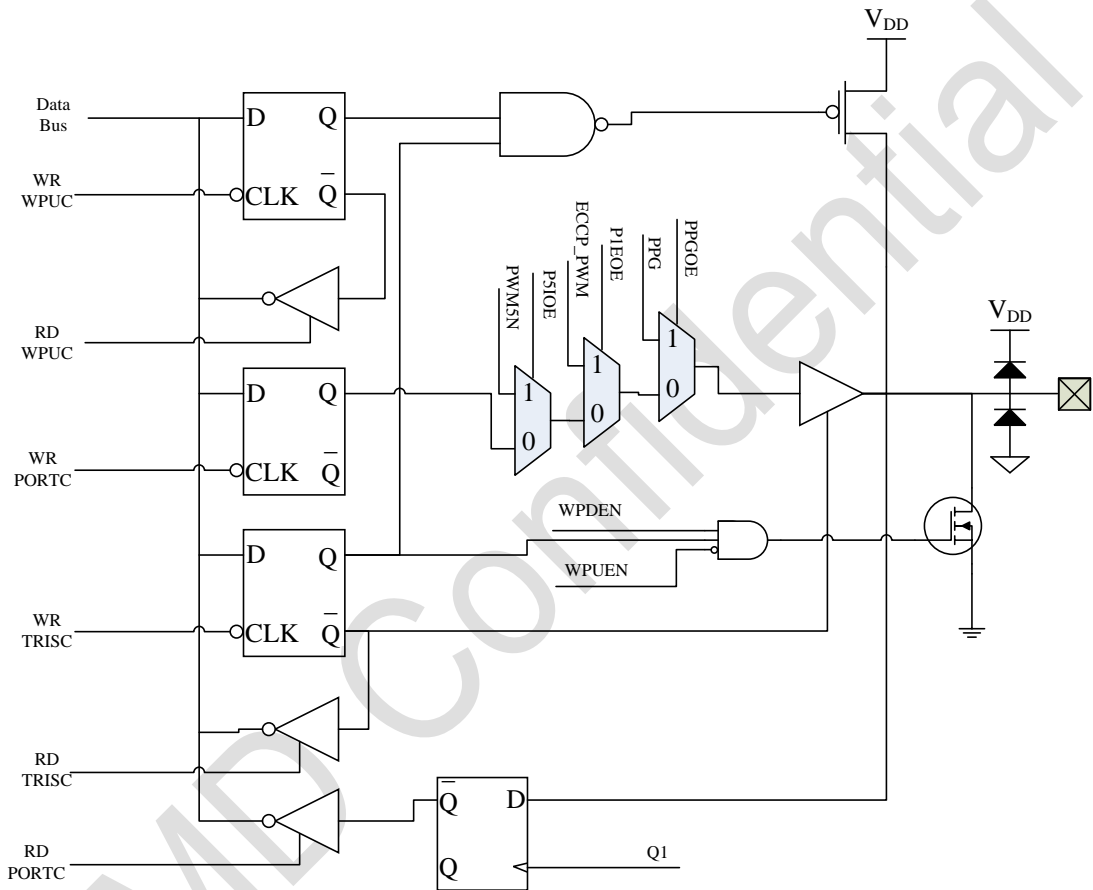


图 16.5.12 PC3 架构框图

注意：图中的 PPGOE 是 PPGOEB 的反相信号。

### 16.5.13 PORTC[4]

图 16.5.13 描述了此端口的内部电路结构。PORTC[4]可以被配置为以下功能端口：

- GPIO
- 增强型 PWM 输出
- PWM5 输出
- ADC 通道 1 输入
- IOFF 输出（PPG 有效信号）

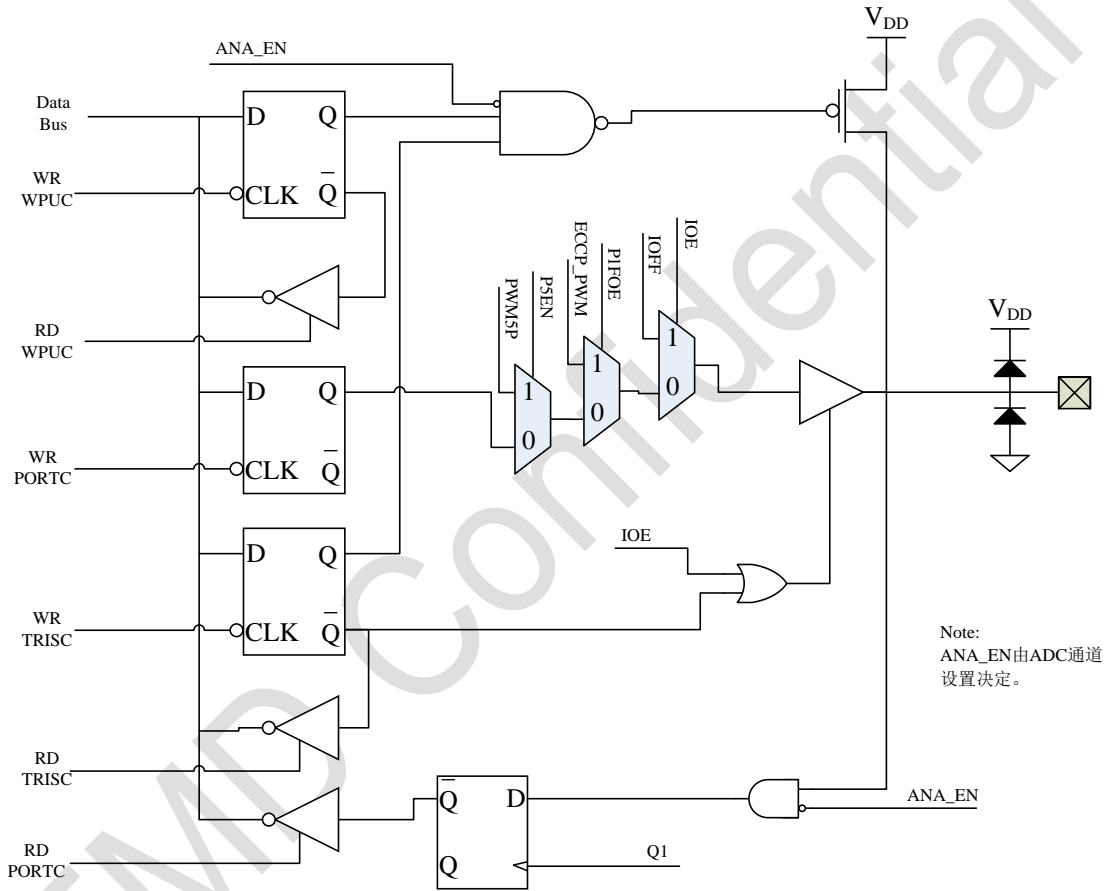


图 16.5.13 PC4 架构框图

注意：图中的 IOE 是 IOEB 的反相信号。

### 16.5.14 PORTC[5]

图 16.5.14 描述了此端口的内部电路结构。PORTC[5]可以被配置为以下功能端口：

- GPIO
- ECCP 的捕捉输入
- PWM3 互补输出
- ADC 通道 2 输入
- 比较器 5 反相输入
- USBA 口的 D-

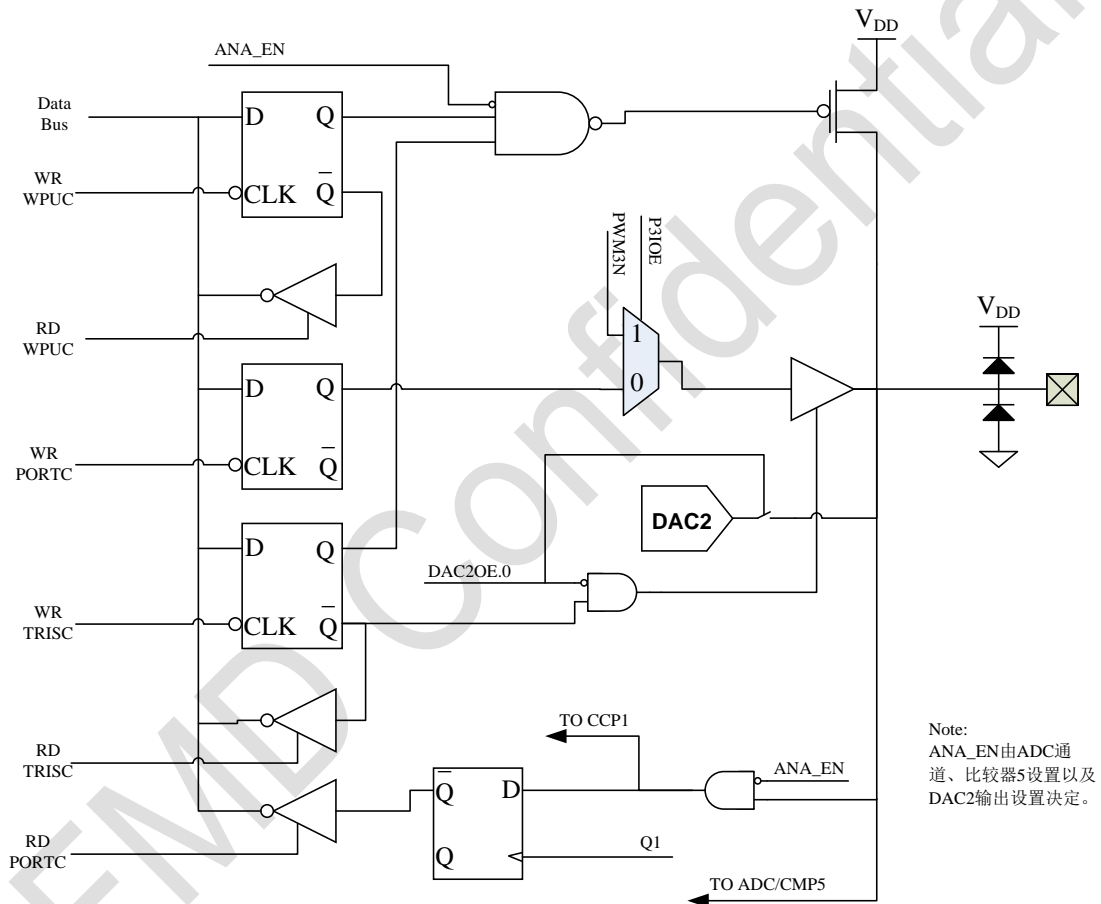


图 16.5.14 PC5 架构框图

### 16.5.15 PORTB[0]

图 16.5.15 描述了此端口的内部电路结构。PORTB[0]可以被配置为以下功能端口：

- GPIO
- 比较器 3 反相输入
- ADC 通道 4 输入
- PWM6 互补输出

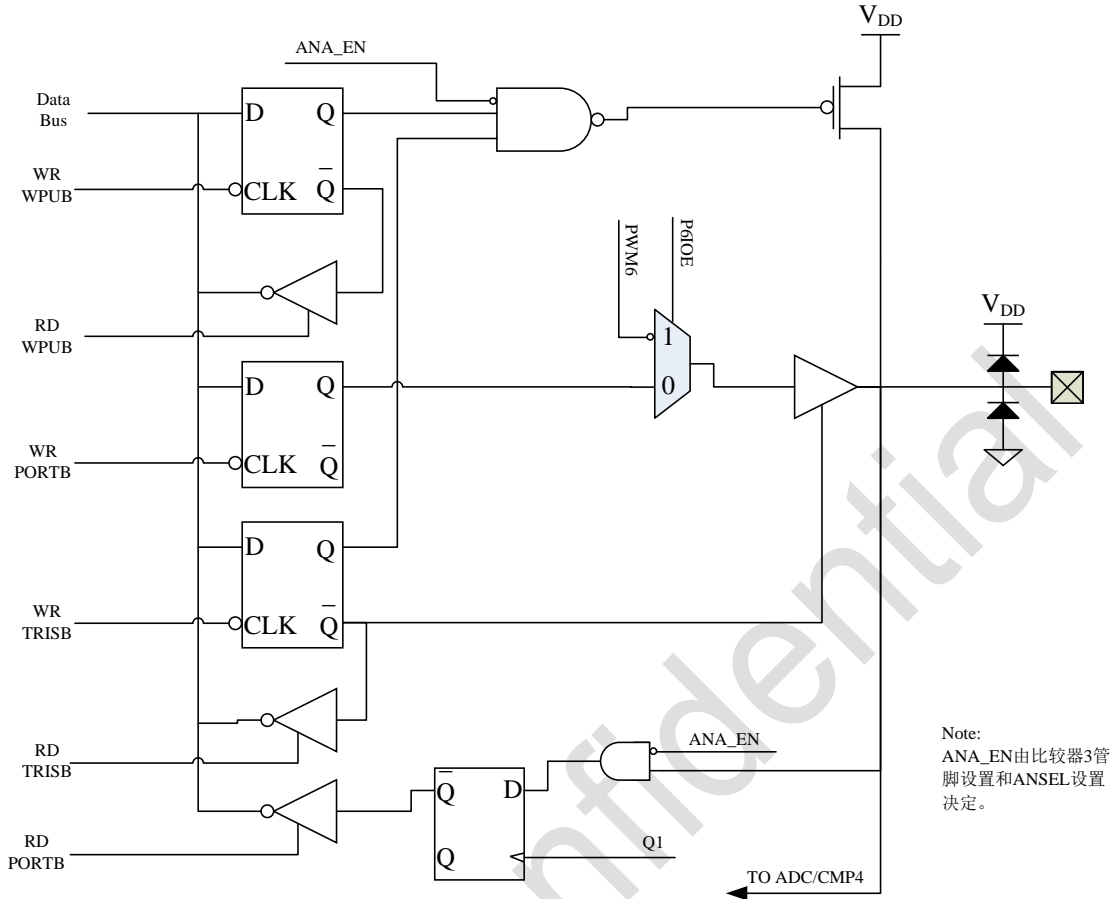


图 16.5.15 PB0 架构框图

### 16.5.16 PORTB[1]

图 16.5.16 描述了此端口的内部电路结构。PORTB[1]可以被配置为以下功能端口：

- GPIO
- 比较器 3 反相输入
- 外设时钟输出
- PWM6 输出

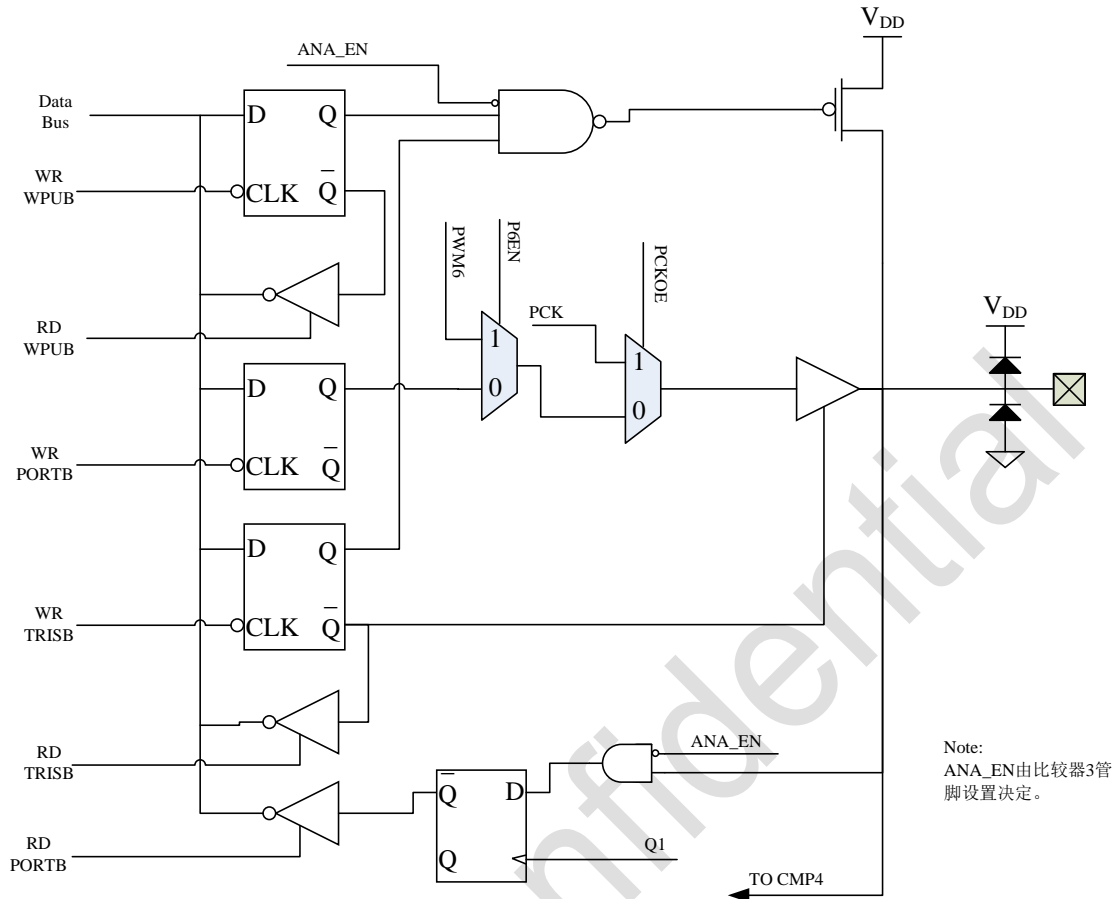


图 16.5.16 PB1 架构框图

### 16.5.17 PORTB[2]

图 16.5.17 描述了此端口的内部电路结构。PORTB[2]可以被配置为以下功能端口：

- GPIO
- 增强型 PWM 输出
- 比较器 4 输出
- D2I 模块的外部电阻

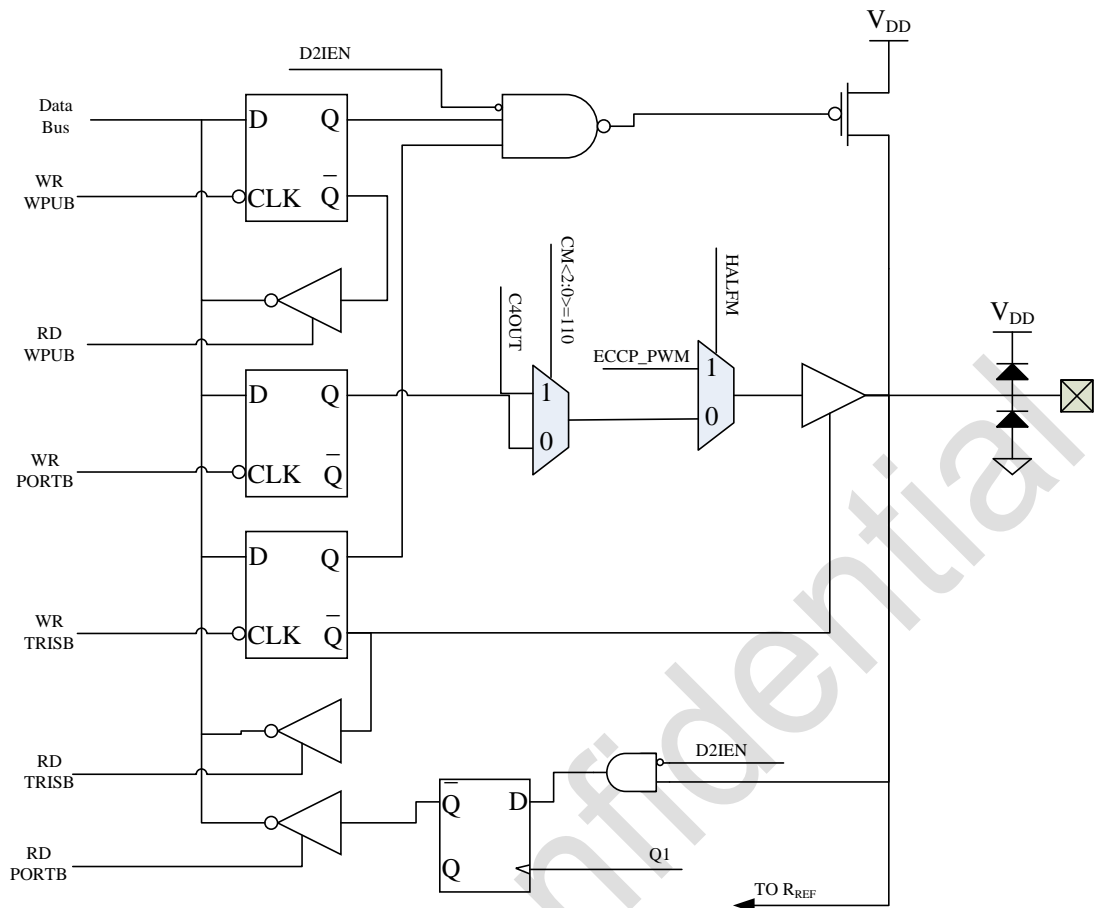


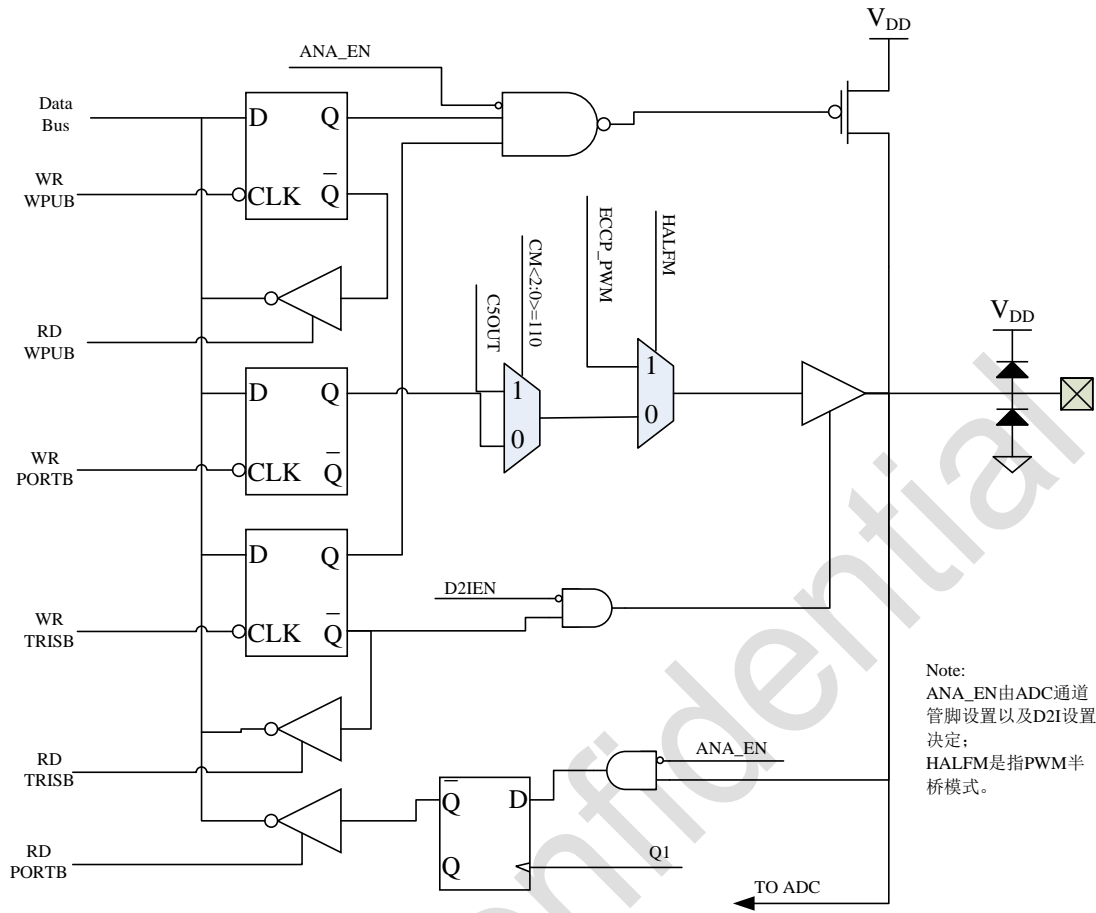
图 16.5.17 PB2 架构框图

### 16.5.18 PORTB[3]

图 16.5.18 描述了此端口的内部电路结构。PORTB[3]可以被配置为以下功能端口：

- GPIO
- ADC 通道 5 输入
- 增强型 PWM 输出
- 比较器 5 输出
- D2I 的电流输入输出





Note:  
ANA\_EN由ADC通道管脚设置以及D2I设置决定;  
HALFM是指PWM半桥模式。

图 16.5.18 PB3 架构框图

## 16.6 与 GPIO 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
WPUA	0x95	PORTA 上拉控制位								1111 1111
TRISA	0x85	TRISA[7:0], PORTA 方向控制								1111 1111
PORTA	0x05	PORTA[7:0], PORTA 数据寄存器								xxxx xxxx
TRISC	0x87	-	-	TRISC[5:0], PORTC 方向控制						--11 1111
PORTC	0x07	-	-	PORTC[5:0], PORTC 数据寄存器						--xx xxxx
WPUC	0x88	-	-	PORTC 弱上拉控制位						--00 0000
IOFA	0x96	IOFA[7:0], 端口下降沿中断允许位								0 0 0 0 0 0 0 0
ANSEL	0x91	ANSEL[7:0], 数模管脚选择								1111 1111
OPTION	0x81	/PAPU	INTEDG	TOCS	T0SE	PSA	PS2	PS1	PS0	1111 1111
WPD	0x89	-	-	-	WPDA4	WPDC1	WPDC2	WPDC3	-	---0 000-
WPUB	0x106	-	-	-	-	PORTB 弱上拉控制位				---- 0000
TRISB	0x86	-	-	-	-	PORTB 方向控制				---- 1111
PORTB	0x06	-	-	-	-	PORTB 数据寄存器				---- xxxx

### 16.6.1 PORTA 寄存器，地址 0x05

Bit	7	6	5	4	3	2	1	0
Name	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	PA[7]	PORTA7 数据
6	PA[6]	PORTA6 数据
5	PA[5]	PORTA5 数据
4	PA[4]	PORTA4 数据
3	PA[3]	PORTA3 数据
2	PA[2]	PORTA2 数据
1	PA[1]	当 MCLRE 为 1 时, PA1 为外部复位管脚, 否则是通用 IO PORTA1 数据
0	PA[0]	PORTA0 数据

### 16.6.2 PORTB 寄存器，地址 0x06

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	PB3	PB2	PB1	PB0
Reset	-	-	-	-	x	x	x	x
Type	R-0	R-0	R-0	R-0	RW	RW	RW	RW

Bit	Name	Function
7:4	-	保留位，读 0
3:0	PORTB [3:0]	PORTB 数据寄存器

### 16.6.3 PORTC 寄存器，地址 0x07

Bit	7	6	5	4	3	2	1	0
Name	-	-	PC5	PC4	PC3	PC2	PC1	PC0
Reset	-	-	x	x	x	x	x	x
Type	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	-	保留位
5:0	PORTC[5:0]	PORTC 数据寄存器

### 16.6.4 TRISA 寄存器，地址 0x85

Bit	7	6	5	4	3	2	1	0
Name	TRISA[7]	TRISA[6]	TRISA[5]	TRISA[4]	TRISA[3]	TRISA[2]	TRISA[1]	TRISA[0]
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TRISA[7:0]	PORTA I/O 三态控制位 1 = PORTA配置为输入 0 = PORTA 配置为输出

### 16.6.5 TRISB 寄存器，地址 0x86

Bit	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---

Name	-	-	-	-	TRISB			
Reset	-	-	-	-	1	1	1	1
Type	R-0	R-0	R-0	R-0	RW	RW	RW	RW

Bit	Name	Function
7:4	-	保留位
3:0	TRISB[3:0]	PORTB I/O 三态控制位 1 = PORTB配置为输入 0 = PORTB配置为输出

### 16.6.6 TRISC 寄存器，地址 0x87

Bit	7	6	5	4	3	2	1	0
Name	-	-	TRISC					
Reset	-	-	1	1	1	1	1	1
Type	-	-	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	-	保留位
5:0	TRISC[5:0]	PORTC I/O三态控制位 1 = PORTC配置为输入 0 = PORTC配置为输出

### 16.6.7 WPUA 寄存器，地址 0x95

Bit	7	6	5	4	3	2	1	0
Name	WPUA7	WPUA6	WPUA5	WPUA4	WPUA	WPUA2	WPUA1	WPUA0
Reset	1	1	1	1	1	1	1	1
Type	Q-RW	Q-RW	Q-RW	Q-RW	Q-RW	Q-RW	Q-RW	Q-RW

**Q-RW:** 当 UCFG1.7 为 1 时，这些位可以由软件读写，否则不能由软件改写，其值只在上电配置过程确定。

Bit	Name	Function
7:2,0	WPUAx	Port A 弱上拉使能 1 = 使能 PORT A 端口弱上拉 0 = 断开 PORT A 端口弱上拉
1	WPUA1	MCLRE = 0 时，PORTA[1]的弱上拉功能 WPUA1 决定： WPUA1 = 1 时，使能 PA1 弱上拉 WPUA1 = 0 时，断开 PA1 弱上拉

		而当 MCLRE = 1 时，PA1 弱上拉一直打开
--	--	----------------------------

### 16.6.8 WPUC 寄存器，地址 0x88

Bit	7	6	5	4	3	2	1	0
Name	-	-	WPUC[5:0]					
Reset	-	-	0	0	0	0	0	0
Type	R-0	R-0	Q-RW	Q-RW	Q-RW	Q-RW	Q-RW	Q-RW

**Q-RW:** 当 UCFG1.7 为 1 时，这些位可以由软件读写，否则不能由软件改写，其值只在上电配置过程确定。

Bit	Name	Function
7:6	-	保留位，读 0
5:0	WPUC[5:0]	Port C 弱上拉使能 1 = 使能 PORTC.x 端口弱上拉 0 = 断开 PORTC.x 端口弱上拉

### 16.6.9 WPD 寄存器，地址 0x89

当 UCFG1.7 为 0 时，此寄存器不能由软件改写，其值只在上电配置过程确定。

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	WPDA4	WPDC1	WPDC2	WPDC3	-
Reset	-	-	-	0	0	0	0	-
Type	R-0	R-0	R-0	Q-RW	Q-RW	Q-RW	Q-RW	R-0

**Q-RW:** 当 UCFG1.7 为 1 时，这些位可以由软件读写，否则不能由软件改写，其值只在上电配置过程确定。

Bit	Name	Function
7:5, 0	-	保留位，读 0
4	WPDA4	PORTA[4]弱下拉使能 1 = 使能 PORTA[4]端口弱下拉 0 = 断开 PORTA[4]端口弱下拉
3	WPDC[1]	PORTC[1]弱下拉使能 1 = 使能 PORTC[3]端口弱下拉 0 = 断开 PORTC[3]端口弱下拉
2	WPDC[2]	PORTC[2]弱下拉使能 1 = 使能 PORTC[2]端口弱下拉 0 = 断开 PORTC[2]端口弱下拉
1	WPDC[3]	PORTC[3]弱下拉使能 1 = 使能 PORTC[1]端口弱下拉 0 = 断开 PORTC[1]端口弱下拉

### 16.6.10 WPUB 寄存器，地址 0x106

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	WPUB[3:0]			
Reset	-	-	-	-	0	0	0	0
Type	R-0	R-0	R-0	R-0	Q-RW	Q-RW	Q-RW	Q-RW

**Q-RW:** 当 UCFG1.7 为 1 时，这些位可以由软件读写，否则不能由软件改写，其值只在上电配置过程确定。

Bit	Name	Function
7:4	-	保留位，读 0
3:0	WPUB[3:0]	Port B[3:0] 弱上拉使能 1 = 使能 PORTB.x 端口弱上拉 0 = 断开 PORTB.x 端口弱上拉

FMD Confidential

## 17 PWMx 和 TIMERx

除了 ECCP 模块所提供的 1 路增强型 PWM 之外，片内还集成了 4 路相互独立的 12bit PWM，它们都具备以下特性：

- BUZZER 或 PWM 输出可选
- 递增计数器
- PWM/BUZZER 输出极性可选
- 8 种 PWM 周期分辨率
- 4 种时钟源可选，最快内部 32M
- 1~256 分频可选
- 7 位预分频
- 溢出中断或者匹配中断
- 时钟调制输出
- 互补脉冲输出
- 外设时钟输出

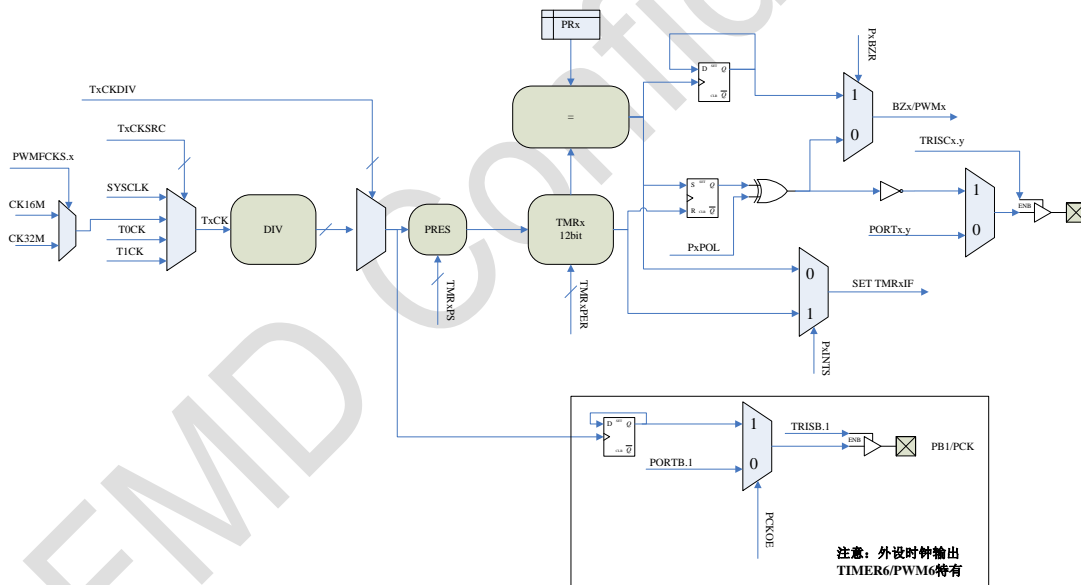


图 17.1 PWM3/4/5/6 原理框图

### 17.1.1 TIMERx 工作原理

当 PxEN (PWMxCR1.7) 为 0 时，工作在定时器模式。

TIMER3/4/5/6 都是 12 位的递增计数器，可以通过寄存器 TMRxH:TMRxL (x 可以是 3/4/5/6) 对计数值进行访问，软件对 TMRxH:TMRxL 的写操作将直接更新计数值。

如果选择内部时钟作为计数源，它们表现为定时器，如果选择外部时钟作为计数源，则表现为计数器。

## 17.1.2 时钟源选择

PWMxCR0 的 PxCKSRC[2:0]用于选择时钟源。配置值和时钟源的关系如下表：

PxCKSRC[2:0]值	PxCK 时钟
000	系统时钟 SYSCLK/(TxCKDIV+1)
001	内部 RC 快时钟/(TxCKDIV+1), 16M 或 32M, 取决于 PWMFCKS 位
010	T0CK/(TxCKDIV+1)
011	T1CK/(TxCKDIV+1)
100	内部 RC 快时钟/(TxCKDIV+1), 16M 或 32M, 取决于 PWMFCKS 位 同时 PWMx 输出低电平
101	内部 RC 快时钟/(TxCKDIV+1), 16M 或 32M, 取决于 PWMFCKS 位 同时 PWMx 输出高电平
110	内部 RC 快时钟/(TxCKDIV+1), 16M 或 32M, 取决于 PWMFCKS 位 同时 PWMx 根据高脉冲调制 PxCK
111	内部 RC 快时钟/(TxCKDIV+1), 16M 或 32M, 取决于 PWMFCKS 位 同时 PWMx 根据低脉冲调制 PxCK

注意：

1. 当 PxCKSRC[2:0]=001 或者 PxCKSRC[2]=1, 即选择为内部 RC 快时钟时, 不管系统时钟选择慢时钟或者外部时钟、晶体时钟, 内部 HFINTOSC 都会打开, 除非进入睡眠模式。

### 17.1.2.1 选择低功耗晶体时钟 (LPOSC)

当 T1CON 的 T1OSCN 为 1 且 UCFG0 的 FOSC[2:0]配置为 32K 晶体或者 INTOSCIO 模式时, TIMERx 的时钟源可以选择 32.768K 晶体振荡, 把 PxCKSRC[2:0]设置为 011 即可。处于这样的配置下, 即使 CPU 进入睡眠模式, TIMERx 还可以自由计数。

系统时钟和定时器时钟互相独立的特性使得对定时精度和 CPU 执行指令速度有要求的应用可以实现。

### 17.1.2.2 外设时钟输出 (PCK)

TIMER6/PWM6 具有外设时钟输出能力, TMR6ON=1 且当 PCKOE(EPWMCR0.7)为 1 和 TRISB.1 为 0 时, PB1 输出由 T6CKSRC 和 T6CKDIV 决定的时钟方波。这种特性可以为外部硬件提供与



MCU 同步的时钟信号。

PCK 的频率为  $F_{T6CK}/((T6CKDIV+1)*2)$ ，即 T6CK 的 2~512 的偶数分频，占空比为 50%。

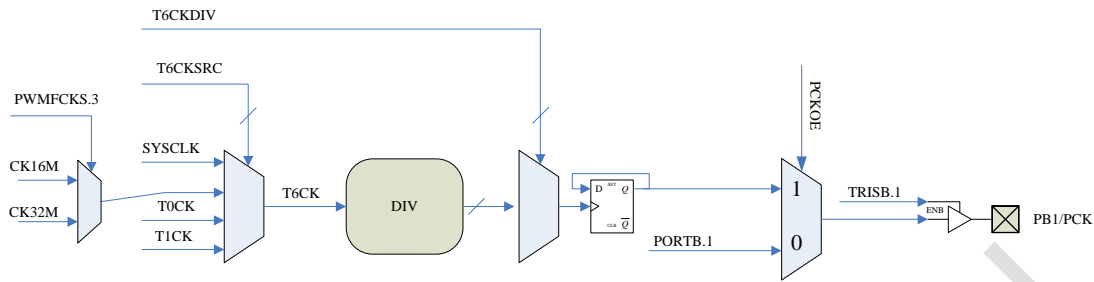


图 17.2 外设时钟输出原理框图

### 17.1.3 TIMERx 时钟分频

PWMx/TIMERx 可以对时钟源进 1~256 分频，其除数由 TxCKDIV 指定。

注意：当发生对 TxCKDIV 寄存器的写操作时，除数分频器将自动清 0。

### 17.1.4 TIMERx 中断

当 TIMERx 计数值 TMRxH:TMRxL 递增到由 PxPER 指定的最大计数值后，再过一个计数时钟便返回 0x000，此时 TIMERx 发生溢出，相关中断标志位被置 1。如果以下条件满足，则 CPU 进入中断处理程序：

- PWMxCR1 寄存器的 TMRxIE 为 1
- INTCON 寄存器的 PEIE 为 1
- INTCON 寄存器的 GIE 为 1

注意：

1. 应该在中断服务退出前把标志位 TMRxIF 清 0，以免循环进入中断。

### 17.1.5 TIMERx 预分频

TIMERx 有 8 种预分频选项，分别可以对 PxCK 进行 1、2、4、8、16、32、64、128 分频。预分频计数器不能直接读写，但当发生对 TMRxH 或者 TMRxL 的写操作时，预分频计数器会自动清 0。

### 17.1.6 TIMERx 周期

TIMERx 最大位数为 12bit，通过对 PxPER[2:0]的配置可以选择不同的溢出周期。

注意：

1. 在 BUZZER 模式下，当计数值和 PR 寄存器相等时，再来一个计数时钟 TIMERx 便会自动清 0；
2. 在 TIMERx 已经开启的情况下改写 PxPER 的值可能会把 TMRxIF 置 1，所以建议先配置好 PxPER 后才开启 TMRxON。

### 17.1.7 TIMERx 在睡眠模式下工作

如果被配置使用外部时钟 T0CK/T1CK (TxCKSRC=010 或者 011)，则在睡眠模式下 TIMERx 可以作为一个定时唤醒源。软件上需要做以下配置：

- PxEN=0，工作在 TIMER 模式
- TxCKSRC=010 或者 011
- PWMxCR1 的 TMRxON 为 1
- PWMxCR1 寄存器的 TMRxIE 为 1
- INTCON 寄存器的 PEIE 为 1

当 TIMERx 定时溢出时，CPU 将被唤醒。如果此时 GIE=1，则执行 SLEEP 的下一条指令后进入中断处理，否则是顺序执行。

### 17.1.8 读写 TMRxH/L 寄存器

由图 17.1 可知，TIMERx 运行时钟相对于 2T 或者 4T 的 CPU 时钟来说是异步时钟，所以在 TIMERx 处于运行状态下软件读取一个 12bit 的计数值时会产生一些问题：有可能在读取了一个低 8bit 之后，TIMERx 产生了溢出导致计数值复位，即再读高 4bit 时已经被清 0。

而对于写操作，建议先把 TIMERx 停止 (TMRxON=0)，再把目标值写入 TMRxH/TMRxL。

## 17.2 BUZZER 工作模式

当 PxEN=1 且 PxBZR=1 时，TIMERx 就处于蜂鸣器 (BUZZER) 模式。

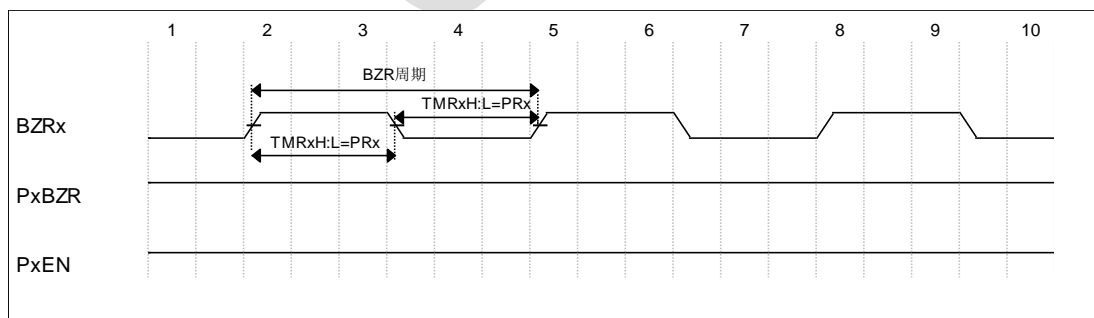


图 17.3 50%占空比的 BUZZER 方波

### 17.2.1 BUZZER 周期

$$T_{BUZ} = 2 * 2^{TMRxPS} * PRx * T_{PxCK}$$

注意：

1. 工作在 BUZZER 模式下时，TIMERx 自动工作在 12bit 模式，而不管 PxPER 为何值；
2. 当 TMRxH:L 等于 PRx 时，TMRxH:L 自动清 0；
3. 处于 BUZZER 模式下，如果 PRx=0x000，则 BZR<sub>x</sub> 管脚固定输出 0，12bit 的 TIMERx 溢出时 TMRxIF 才会置 1。

## 17.2.2 BUZZER 在休眠模式下工作

同 TIMER 模式一样，通过适当的配置，BUZZER 也可以工作在睡眠模式：

- PxEN=1，工作在 PWM/BUZZER 模式
- PxBZR=1，选择 BUZZER 功能
- TxCKSRC=010 或者 011
- PWMxCR1 的 TMRxON 为 1
- PWMxCR1 寄存器的 TMRxIE 为 1
- INTCON 寄存器的 PEIE 为 1

## 17.3 PWM 工作模式

当 PxEN=1 且 PxBZR=0 时，TIMERx 工作于 PWM 模式。

下图为 PWM 输出的一个示例，它由周期和一段输出保持为高（占空比）的时间组成。

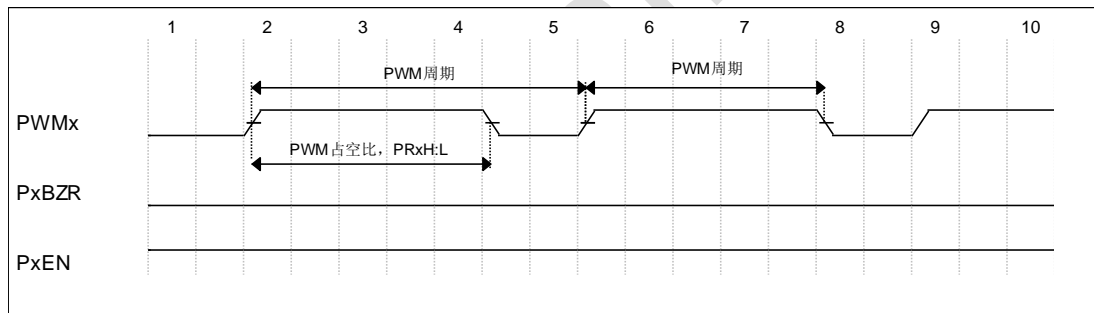


图 17.4 PWM 工作模式，正向输出

PWM3/4/5/6 的时基和占空比相互独立，能最大限度满足各种应用的需求。

注意：

如果想同时使用 ECCP 的 PWM（P1A 管脚输出 PWM 调制）和 PWM3/4/5，则寄存器 CCP1CON 的 P1M<1:0>要配置为 00，即单输出模式，P1A 调制，P1B/P1C/P1D 为端口配置。换句话说，ECCP 优先级比 PWM3/PWM4/PWM5 要高。

### 17.3.1 PWM 周期

PWMx 周期由 TIMERx 的预分频比设置寄存器 TMRxPS，PxPER 决定。公式如下：

$$PWM \text{ 周期} = 2^{TMRxPS} * 2^{Nbit} * T_{PxCK}$$

式中，Nbit 为 PxPER 设定的定时器位数。

### 17.3.2 PWM 占空比

PWM 的占空比由寄存器 PRxH:L 决定。PRxH:L 一共 12bit，软件需要分开两次写。由于没有映像寄存器，软件对 PRxH:L 的写是立刻生效，即 PRxH:L 值的改变对当前 PWM 占空比有影响，应用必须考虑这一点。

下式是计算占空比（脉冲宽度）的公式：

$$T_{pwm} = 2^{TMRxPS} * (PRx) * T_{PxCK}$$

**注意：**

由于 PRx 没有对应的映像寄存器，所以 PRx 的写操作将直接影响当前占空比，建议在没启动 TIMERx 之前把目标值写入 PRx。

### 17.3.3 PWM 工作设置

建议按以下步骤设置 PWM3/4/5 工作：

1. 将相关的 TRIS 位置 1 禁止 PWMx 引脚的输出驱动器；
2. 装载 PWMxCR0 寄存器以设置 PWMx 周期，中断产生方式以及选择时钟源；
3. 用适当的值装载 PWMxCR1 寄存器配置为 PWM 模式，合适的预分频比；
4. 装载 PRx 寄存器设置 PWM 占空比；
5. 配置并启动 TIMERx：
  - 将 PWMxCR1 寄存器的 TMRXIF 中断标志位清零
  - 将 PWMxCR1 寄存器的 TMRxON 位置 1 启动 TIMERx
6. 重新开始一个 PWM 周期后，使能 PWM 输出：
  - 等待 TIMERx 溢出（PWMxCR1 寄存器的 TMRxIF 位置 1）
  - 将相关的 TRIS 位清零使能 PWMx 引脚的输出驱动器

**注意：**

- 1 如果 PWM 时钟设置为系统时钟（TxCKSRC=000），则系统时钟频率的任何改变将导致 PWM 频率的改变。

### 17.3.4 PWM 工作在休眠模式

同 TIMER 模式一样，通过适当的配置，PWM 也可以工作在睡眠模式：

- PxEN=1，工作在 PWM/BUZZER 模式
- PxBZR=0，选择 PWM 功能
- TxCKSRC=010 或者 011
- PWMxCR1 的 TMRxON 为 1

- PWMxCR1 寄存器的 TMRxIE 为 1
- INTCON 寄存器的 PEIE 为 1

### 17.3.5 PWM 脉冲的互补输出

通过对寄存器 EPWMCR0 的低 4 位置 1, 4 路 PWM/BZR 均可输出其互补脉冲, 如图 17.5 的 PWMxN 所示。

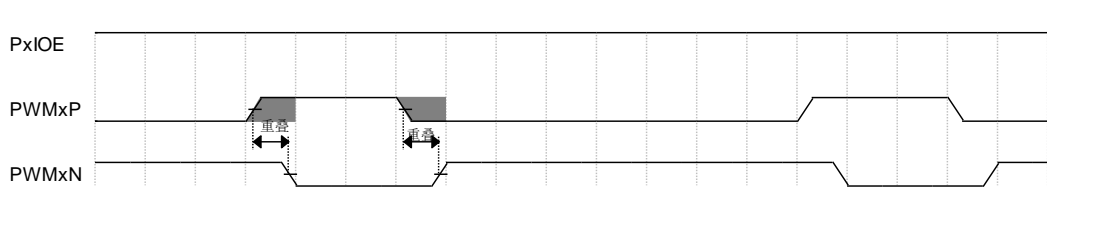


图 17.5 PWM 互补输出时序

注意:

1. 要想输出 PWMxN 脉冲, 相关 TRISCx 的要清 0;
2. PWMxP 和 PWMxN 这一对互补 PWM 脉冲并不具有“死区”控制, 在脉冲翻转时可能有一小段时间处于交叠状态, 在应用上应当引起重视。

## 17.4 PWMx 相关寄存器

### 17.4.1 EPWMCR0 寄存器, 地址 0x11E

Bit	7	6	5	4	3	2	1	0
Name	PCKOE	-	-	-	P6IOE	P5IOE	P4IOE	P3IOE
Reset	0	-	-	-	0	0	0	0
Type	RW	-	-	-	RW	RW	RW	RW

Bit	Name	Function
7	PCKOE	外设时钟输出使能 1 = 管脚 PB1 可以输出 TIMER6 的分频时钟, 频率为 P6CK/2 0 = 管脚 PB1 作为普通 IO
6:4	-	保留位, 未实现
3	P6IOE	PWM6 互补输出使能 1 = 引脚 PWM6N 可输出 PWM6 互补脉冲 0 = 引脚 PWM6N 作为普通 IO
2	P5IOE	PWM5 互补输出使能 1 = 引脚 PWM5N 可输出 PWM5 互补脉冲 0 = 引脚 PWM5N 作为普通 IO
1	P4IOE	PWM4 互补输出使能

		1 = 引脚 PWM4N 可输出 PWM4 互补脉冲 0 = 引脚 PWM4N 作为普通 IO
0	P3IOE	PWM3 互补输出使能 1 = 引脚 PWM3N 可输出 PWM3 互补脉冲 0 = 引脚 PWM3N 作为普通 IO

### 17.4.2 EPWMCR1 寄存器，地址 0x105

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	PWMFCKS[3:0]			
Reset	-	-	-	-	0	0	0	0
Type	-	-	-	-	RW	RW	RW	RW

Bit	Name	Function
7:4	-	保留位，未实现
3	PWMFCKS[3]	PWM6 内部快时钟选择位 1 = 选择 32M 内部快时钟 0 = 选择 16M 内部快时钟
2	PWMFCKS[2]	PWM5 内部快时钟选择位 1 = 选择 32M 内部快时钟 0 = 选择 16M 内部快时钟
1	PWMFCKS[1]	PWM4 内部快时钟选择位 1 = 选择 32M 内部快时钟 0 = 选择 16M 内部快时钟
0	PWMFCKS[0]	PWM3 内部快时钟选择位 1 = 选择 32M 内部快时钟 0 = 选择 16M 内部快时钟

### 17.4.3 TMRxL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	TMRxL							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TMRxL[7:0]	TIMERx 计数结果低 8 位，x 为 3/4/5/6，地址分别是 0x10C/112/118/198

### 17.4.4 TMRxH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	TMRxH[3:0]				PRxH[3:0]			
Reset	x	x	x	x	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:4	TMRxH[7:0]	TIMERx 计数结果高 4 位
3:0	PRxH[3:0]	PWMx 匹配寄存器 PRx 高 4 位

注意：x 为 3/4/5/6，地址分别是 0x10D/113/119/199。

### 17.4.5 PRxL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PRxL							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PRxL[7:0]	PWMx 匹配寄存器 PRx 低 8 位，x 为 3/4/5/6，地址分别是 0x10E/114/11A/19A

### 17.4.6 PWMxCRO 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PxINTS	PxPER[2:0]			PxCKSRC[2:0]			PxBZR
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function										
7	PxINTS	PWMx 中断选择位 1 = TMRx 计数值和 PRx 匹配时产生中断 0 = TMRx 计数值溢出时产生中断										
6:4	PxPER[2:0]	PWMx 周期选择位 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>PxPER[2:0]值</th> <th>PWMx 周期位数</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>4 位</td> </tr> <tr> <td>001</td> <td>5 位</td> </tr> <tr> <td>010</td> <td>6 位</td> </tr> <tr> <td>011</td> <td>8 位</td> </tr> </tbody> </table>	PxPER[2:0]值	PWMx 周期位数	000	4 位	001	5 位	010	6 位	011	8 位
PxPER[2:0]值	PWMx 周期位数											
000	4 位											
001	5 位											
010	6 位											
011	8 位											

		100	9 位
		101	10 位
		110	11 位
		111	12 位
3:1	PxCKSRC[2:0]	TIMERx/PWMx 时钟选择位	
		PxCKSRC[2:0]值	PxCK 时钟
		000	系统时钟 SYSCLK/(TxCKDIV+1)
		001	内部 RC 快时钟/(TxCKDIV+1)
		010	T0CK/(TxCKDIV+1)
		011	T1CK/(TxCKDIV+1)
		100	内部 RC 快时钟/(TxCKDIV+1) 同时 PWMx 输出低电平
		101	内部 RC 快时钟/(TxCKDIV+1) 同时 PWMx 输出高电平
		110	内部 RC 快时钟/(TxCKDIV+1) 同时 PWMx 根据高脉冲调制 PxCK
		111	内部 RC 快时钟/(TxCKDIV+1) 同时 PWMx 根据低脉冲调制 PxCK
		<b>注意：内部 RC 快时钟的频率取决于 PWMFCKS 位的值，当 PWMFCKS.x=1 时，快时钟为 32M；当 PWMFCKS.x=0 时，快时钟为 16M</b>	
0	PxBZR	PWM/BUZZER 选择 1 = 选择 BUZZER 输出 0 = 选择 PWM 输出	

注意：x 为 3/4/5/6，地址分别是 0x10F/115/11B/19B。

### 17.4.7 PWMxCR1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PxEN	PxPOL	TMRxPS[2:0]			TMRxON	TMRxIE	TMRxIF
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	PxEN	TMRxPWMx 工作模式 1 = TMRx 为 PWM/BUZZER 模式 0 = TMRx 为定时器模式
6	PxPOL	PWMx 输出极性选择 1 = PWMx 为低电平有效 0 = PWMx 为高电平有效
5:3	TMRxPS[2:0]	PWMx 预分频比设置位



		TMRxPS[2:0]值	PWMx 分频比
		000	1:1
		001	1:2
		010	1:4
		011	1:8
		100	1:16
		101	1:32
		110	1:64
		111	1:128
2	TMRxON	定时器 x 使能位 1 = 打开定时器 x 0 = 关闭定时器 x	
1	TMRxIE	定时器 x 中断使能位 1 = 允许 TMRx 产生中断 0 = 禁止 TMRx 产生中断	
0	TMRxIF	定时器 x 中断标志位 1: PxINTS =1 时, TMRx 发生了匹配 PxINTS =0 时, TMRx 发生了溢出 0: 没有发生溢出或匹配	

注意: x 为 3/4/5/6, 地址分别是 0x110/116/11C/19C。

### 17.4.8 TxCKDIV 寄存器

Bit	7	6	5	4	3	2	1	0
Name	TMRx 时钟分频寄存器							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TxCKDIV[7:0]	TMRx 时钟频率为 $F_{TxCK}/(TxCKDIV+1)$

注意: x 为 3/4/5/6, 地址分别是 0x111/117/11D/19D。

## 18 可编程脉冲生成器 (PPG)

芯片内置的 PPG 模式有以下特性：

- 独立其它 TIMER 的 9 位计数时基
- 软件或者硬件的触发方式
- 触发延迟可设置，提供最多 64 个系统时钟的延迟
- 不可重复触发死区时间
- 8 位脉宽限制计数器
- 2 个重载寄存器
- 脉冲输出极性可选

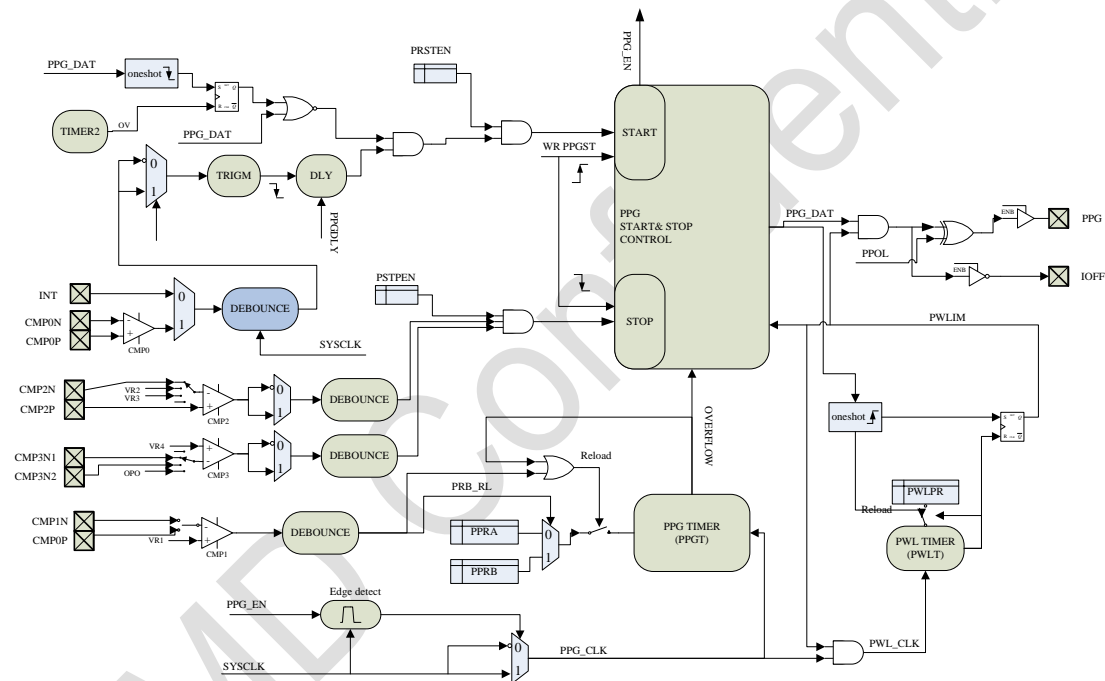


图 18.1 PPG 原理框图

### 18.1 基本工作原理

PPG 模块主要由 PPG 模式控制器，1 个 PPG 定时器，1 个延迟触发计数器，2 个重载寄存器，1 个 PPG 脉宽限制定时器，4 个模拟比较器，以及相关控制逻辑组成。其中，2 个定时器皆为递增计数器。

当发生了一次有效触发事件时，PPG 定时器从预载寄存器（PPRA 或 PPRB）的值开始计数，PPG 将输出有效脉宽，直到定时器溢出，即计数值由 0x1FF 翻转到 0x000，或者停止触发事件发生，如比较器 2、3 输出翻转，这时定时器 PPGT 停止计数，从 PPRA 或者 PPRB 重新加载预置值，同时输出 PPG 也变为无效，直到下一次有效触发事件的发生。

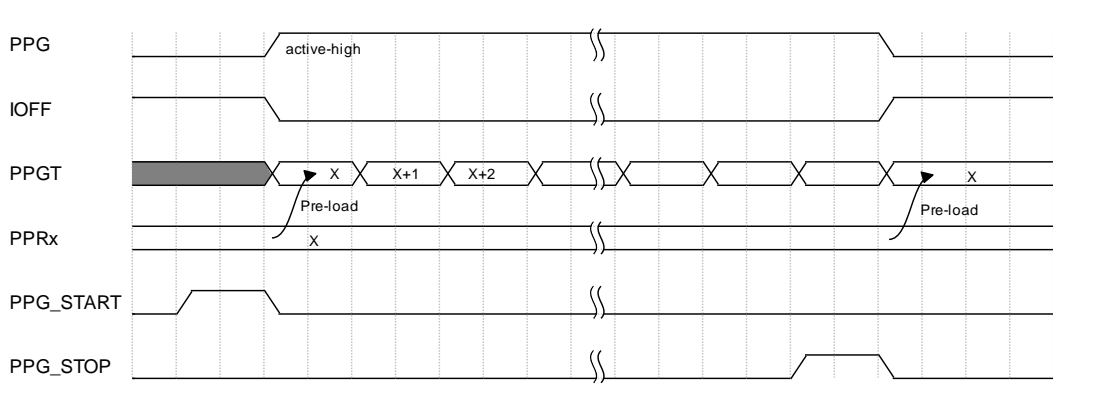


图 18.2 PPG 脉冲基本时序

发生系统复位时，PPG 管脚为数字输入，IOFF 为模拟 IO，在应用中为避免漏电，可以在这两个管脚接上上拉或者下拉电阻。

需要注意的是，PPG 管脚和 IOFF 管脚都与其他 IO 功能复用，它们的管脚配置位于 UCFG13 寄存器，当 PC3 和 PC4 分别被设置为 PPG 和 IOFF 功能时，该管脚的其它功能将自动被禁止，而不管 TRISC 或其它 SFR 如何设置。

## 18.2 PPG 管脚的状态

PPG 和 IOFF 在上电复位后处于悬空状态，上电复位结束后，IOFF 输出高电平，但 PPG 处于无效状态，PPG 无效时管脚是悬空的，在使用时外部可根据实际情况接上拉或者下拉。

	IOFF 状态	IOFF 输出
PPG 有效		低电平
PPG 无效		高电平

表 18.1 IOFF 和 PPG 的关系

## 18.3 禁止重复触发

禁止重复触发功能需要使用到 TIMER2。在禁止重复触发期间，即使发生了有效的触发事件，PPG 不会再次触发。

TIMER2 的禁止重复触发在 [8.2 小节](#)描述。

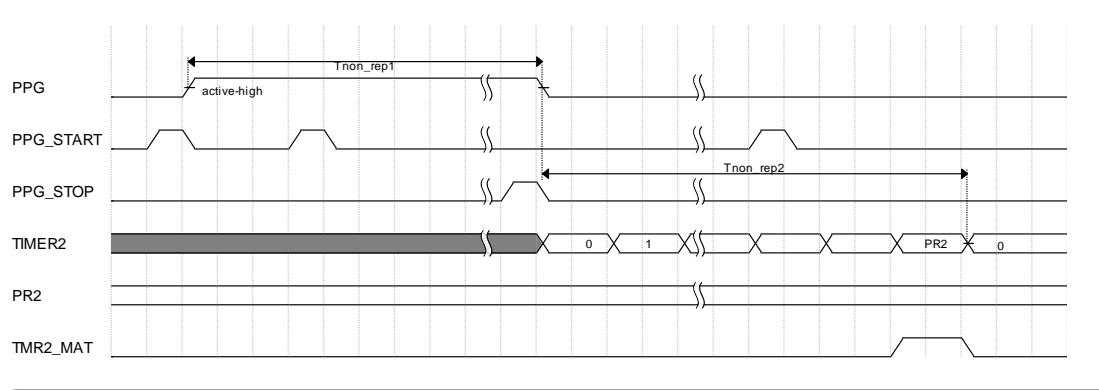


图 18.3 禁止重复触发期

禁止重复触发期定义为以下条件之一：

- 当 PPG 正在输出有效脉冲时（图中的 Tnon\_rep1）；
- 寄存器位 NRPM=1（位于 T2CON.7），从 PPG 输出变为无效的时刻开始直到 TIMER2 和 PR2 发生匹配的这段时间（图中的 Tnon\_rep2）；

## 18.4 脉宽限制

PPG 模块具有脉宽限制功能，模块内嵌了一个 8 位的定时器 PWLT，其计数时钟为系统时钟的 2 分频，它是一个向上递增的定时器。在 PPG 脉冲开始有效的时刻，预置值被加载到 PWLT，当 PWLT 计数溢出时，PPG 脉冲将变为无效，PWLT 重新载入预置值 PWLPR 并停止计数。

有几种情况 PWLT 会停止计数：

- PWLTEN 为 0
- PPG 无效期间
- PWLT 计数溢出后

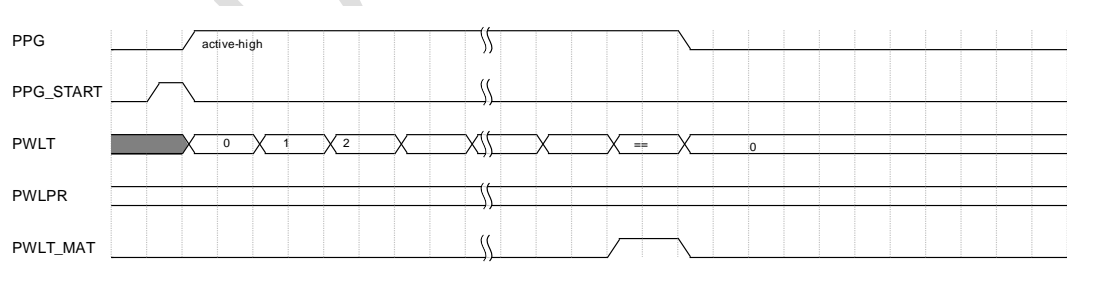


图 18.4 PPG 的脉宽限制

## 18.5 PPG 脉冲的同步

当 PSYNC=1 时，PPG 输出跟系统时钟同步。根据触发事件和系统时钟的相位关系，PPGT 定时器可以在系统时钟的下降沿或者上升沿递增计数，这种特性使得 PPG 的启动延时不大于半个系统时钟周期。

触发事件到来时，PPG 脉冲输出使处于待命状态，当遇到第一个系统时钟沿后，PPG 脉冲将输出到管脚上；同时 PPGT 定时器在跟第一个时钟相同的边沿作用下开始计数。换句话说，如果触发事件到来后遇到的系统时钟沿为上升沿，那么 PPGT 的计数时钟边沿为上升沿；反之如果触发事件到来后首次到达的系统时钟沿为下降沿，则 PPGT 的计数时钟边沿为下降沿，一直保持到停止事件的产生。下图解释了所述的两种情况：

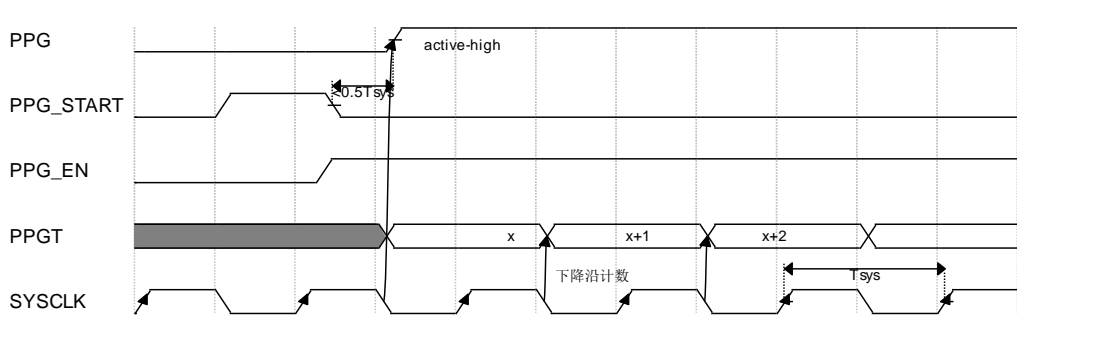


图 18.5 PPG 脉冲在时钟下降沿同步

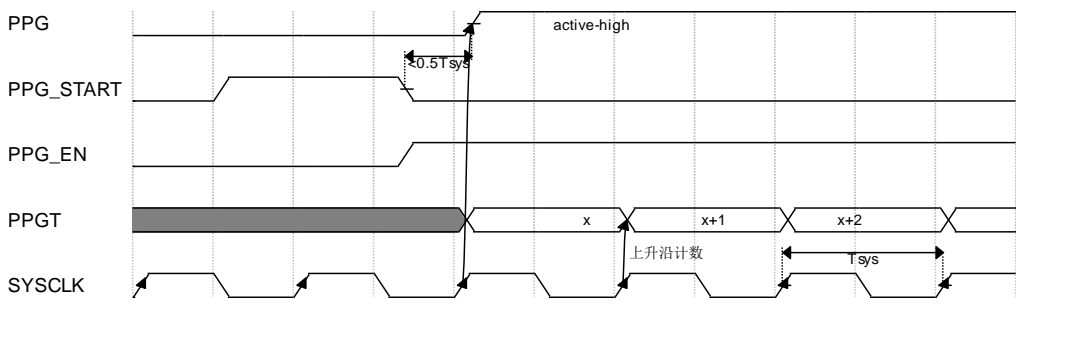


图 18.6 PPG 脉冲在时钟上升沿同步

## 18.6 有效触发事件

有效的触发事件有以下几种：

- 软件向 PPGST 位写 1
- PRSTEN=1, TRIGSRC=0 时：
  1. INTEDG=1 时，INT 管脚由低变高；
  2. INTEDG=0 时，INT 管脚由高变低；
- PRSTEN=1, TRIGSRC=1 时：
  1. COPOL=1 时，比较器由低变高；
  2. COPOL=0 时，比较器由高变低；

注意：

1. 无论 PPG 是否处于有效周期，当 PSTPEN=1 时，比较器 2 或 3 的下降沿总是会清 PRSTEN，这样可阻止 PPG 再次由外部事件比较器 0 或者 INT 管脚触发，而只能由软件写 PPGST 位重新触发，或者软件重新置位 PRSTEN 以允许外部事件触发。

## 18.7 停止 PPG 事件

停止 PPG 事件定义为以下几种：

- 软件向 PPGST 位写 0
- PPGT 定时器计数溢出
- PWLTEN=1 且 PWLT 定时器和 PWLPR 值匹配
- PSTPEN=1 且比较器 2 或者比较器 3 输出由高变低

发生停止 PPG 事件后，PPG 模块将发生以下情况：

- PPG 输出无效，IOFF 输出高电平
- PPGT 停止计数
- PPGT 重载 PPRA 或者 PPRB 的值
- PPGST 位被清 0

## 18.8 PPGT 的重载

PPGT 定时器的计数值不能被软件直接访问，它只在 PPG 有效触发事件发生或者停止事件发生时被初始化，当 PRLSEL 为 0 时加载 PPRA 寄存器的值，当 PRLSEL=1 时，加载 PPRB 寄存器的值。

寄存器位 RELDB 设置为 1 时，比较器 1 输出的下降沿将自动把 PRLSEL 置 1，从而控制 PPGT 的重载源。PRLSEL 位一旦被置 1 就一直保持，直到软件把它清除。

## 18.9 延迟触发

通过对 PPGCON2 可以实现 PPG 的延迟触发，延迟时间可选，为 2~64 个系统时钟周期。延迟时间是由一个 6 位的 PPG 延迟计数器 PDLYT 和寄存器位 PPGDLY[5:0]共同确定的，计数值和 PPGDLY 发生匹配时才把触发事件送到 PPG 相关启动电路。

### 18.9.1 基本工作原理

当发生有效触发事件后，且 PPGDLY≠0 时，PPG 波形输出不会立即有效。这时延迟触发计数器 PDLYT 清 0 并自动使能，在系统时钟上升沿的作用下递增计数，当 PDLYT 计数值与 PPGDLY 相等时，再来一个系统时钟上升沿才把 PPG 触发，PPG 输出有效波形。

下图是设置为 INT 管脚触发源的 PPG 延迟触发时序。

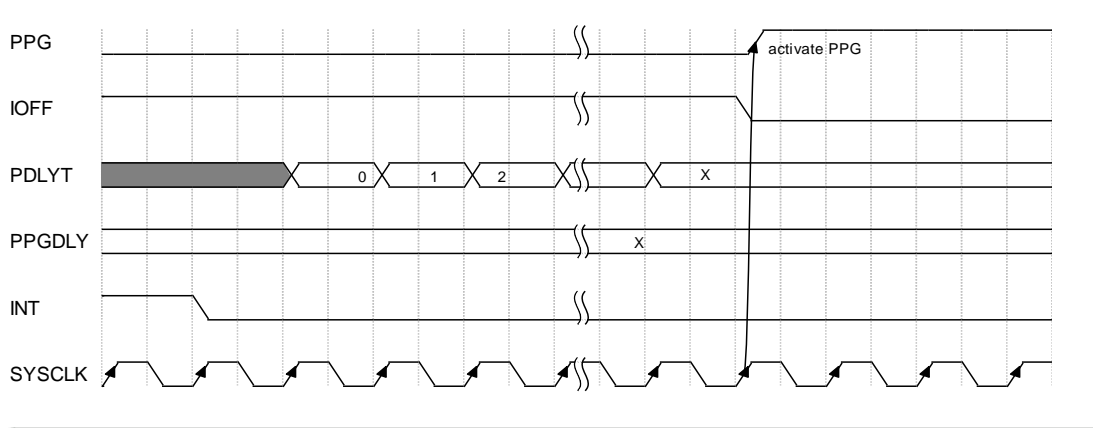


图 18.7 PPG 的延迟触发时序

注意:

1. PDLYT 计数值对软件不可见;
2. 延迟触发计数会将忽略在 PDLYT 计数期间再次发生的有效触发事件;

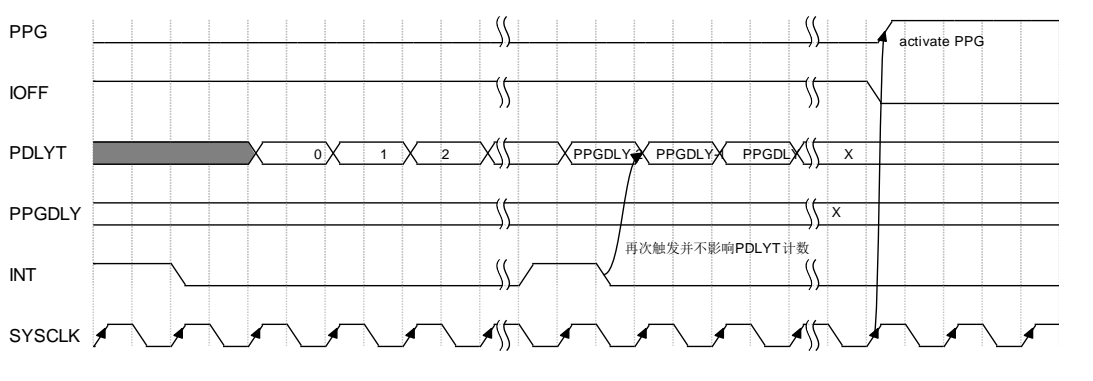


图 18.8 PDLYT 不受后面触发沿的影响（单沿触发）

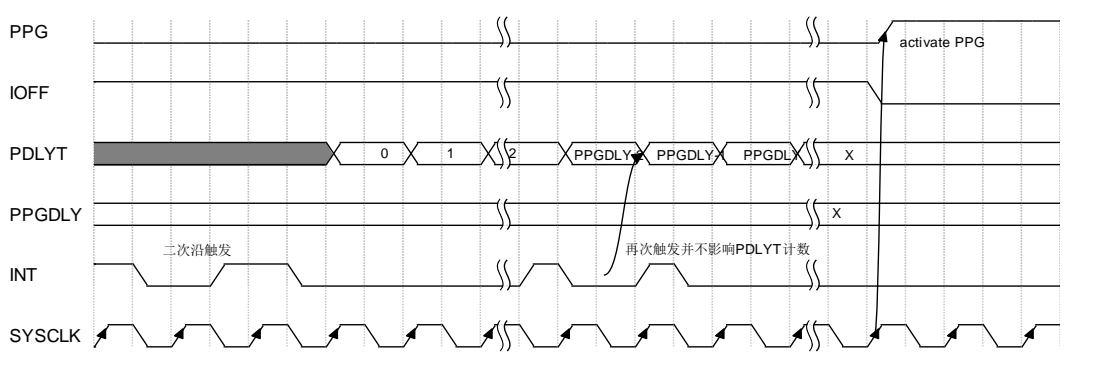


图 18.9 PDLYT 不受后面触发沿的影响（双沿触发）

3. 在禁止重复触发期间发生触发事件，PPG 模块都将忽略，即延时触发计数器也不会被触发;
4. 延迟触发计数器只对 INT 或者比较器 0 输出触发有效，对软件写 PPGST 位不起作用；换言之，PPGST 被软件置 1 后，PPG 输出立即有效，而不管 PPGDLY 为何值。

5. 在延迟触发期间，如果发生了停止事件（PPGST 由 1 变化到 0，或者 PSTPEN=1 时比较器 2/3 有下降沿产生），则延迟触发计数器强制停止，PPG 不会被触发输出。

## 18.10 PPG 工作设置步骤

建议使用以下步骤启动 PPG：

- 配置选项把 PC3 管脚设置为 PPG 功能；
- 设置 PPGCON1 寄存器，选择好 PPG 输出极性，选择触发事件和停止事件，重载源，是否需要同步和使用脉宽限制以及触发模式（单次/双次沿）；
  - 如果 PRSTEN 为 1，则需要设置寄存器 DEBCR，选择 INT 管脚或者比较器 0 输出触发，及其去抖时间；
  - 如果选择 INT 管脚触发需要设置寄存器 OPTION 的 INTEDG 位，PC1 设置为数字输入；
  - 如果选择比较器 0 输出触发需要设置 CMCON1/2/4，PC0/PC1 设置为数字输入；
  - 如果选择比较器 2/3 输出作为停止事件，则需要设置 CMCON1/2/4；
- 设置 PPGT 重载源寄存器 PPR A/PPR B；
- 如果使用到脉宽限制功能，则设置 PWLTPR，以及置位 PPGCON1 的 PWLTEN；
- 设置 PPGCON2，选择是否需要比较器 1 的下降沿置位 PRLSEL，是否需要延迟触发；
  - 如果 RELDB 为 1，则设置 CMCON1/2/4，配置好比较器 1，相关 IO 设置数字输入；
- 最后置位 PPGON 使能 PPG 模块；

## 18.11 与 PPG 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
PPGCON1	0x188	PTRIGM	-	PWLTEN	PRLSEL	PRSTEN	PSTPEN	-	PPGST	0-00 00-0
PPRH	0x189	-	-	-	PPRB[8]	-	-	-	PPRA[8]	---x ---x
PPRA	0x18C	PPRA[7:0], PPG 重载寄存器 A 低 8 位								xxxx xxxx
PPRB	0x18D	PPRB[7:0], PPG 重载寄存器 B 低 8 位								xxxx xxxx
PWLTPRL	0x18E	PPG 脉宽限制比较寄存器低 8 位								1111 1111
PPGCON2	0x18F	PPGON	RELDB	PPGDLY[5:0]						0000 0000
DEBCR	0x181	-	TRIGSRC	DEBDLY[5:0]						-000 0000
PPGCON3	0x19E	-	-	-	-	PSYNC	PPOL	PPGOEB	IOEB	---- 0011

### 18.11.1 PPGCON1 寄存器，地址 0x188

Bit	7	6	5	4	3	2	1	0
Name	PTRIGM	-	PLWTEN	PRLSEL	PRSTEN	PSTPEN	-	PPGST
Reset	0	-	0	0	0	0	-	0
Type	RW	R-0	RW	RW	RW	RW	R-0	RW

Bit	Name	Function
7	PTRIGM	PPG 触发模式选择（只针对 INT 管脚或者比较器 0 输出）



		1 = 发生两次有效边沿后触发 PPG 0 = 发生一次有效边沿后触发 PPG
6	-	保留位，读 0
5	PLWTEN	PPG 脉宽限制使能 1 = 使用脉宽限制 0 = 不使用脉宽限制
4	PRLSEL	PPG 重载寄存器选择 1 = 选择 PPRB 作为重载寄存器 0 = 选择 PPRA 作为重载寄存器
3	PRSTEN	PPG 触发选择 1 = PPG 可由软件，INT 或者比较器 0 输出触发，见寄存器 TRIGSRC 0 = PPG 只能由软件触发（PPGST 位写 1） <b>当 PSTPEN=1，比较器 2 或者 3 发生下降沿跳变事件时，该位自动被清 0</b>
2	PSTPEN	PPG 停止条件选择 1 = PPG 可由软件，比较器 2 或者比较器 3 输出停止 0 = PPG 只能由软件停止（PPGST 位写 0）
1	-	保留位，读 0
0	PPGST	PPG 的软件启动位 软件写操作 1 = 触发 PPG 脉冲输出 0 = 停止 PPG 脉冲输出 软件读操作： 1 = PPG 正有效输出 0 = PPG 停止 <b>注意：PRSTEN=1 且发生 INT 或者比较器 0 有效触发事件时，该位会自动置 1</b>

### 18.11.2 PPRH 寄存器，地址 0x189

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	PPRB[8]	-	-	-	PPRA[8]
Reset	-	-	-	x	-	-	-	x
Type	-	-	-	RW	-	-	-	RW

Bit	Name	Function
7:5	-	保留位，未实现
4	PPRB[8]	PPG 重载寄存器 B 第 8 位
3:1	-	保留位，未实现
0	PPRA[8]	PPG 重载寄存器 A 第 8 位

### 18.11.3 PPRA 寄存器，地址 0x18C

Bit	7	6	5	4	3	2	1	0
Name	PPRA[7:0]							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PPRA	PPG 重载寄存器 A 低 8 位

### 18.11.4 PPRB 寄存器，地址 0x18D

Bit	7	6	5	4	3	2	1	0
Name	PPRB[7:0]							
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PPRB	PPG 重载寄存器 B 低 8 位

### 18.11.5 PWLTPRL 寄存器，地址 0x18E

Bit	7	6	5	4	3	2	1	0
Name	PPG 脉宽限制比较寄存器低 8 位							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PWLTPRL	PPG 脉宽限制比较寄存器低 8 位

### 18.11.6 PPGCON2 寄存器，地址 0x18F

Bit	7	6	5	4	3	2	1	0
Name	PPGON	RELDB	PPGDLY[5:0], 延迟触发选择寄存器					
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	PPGON	PPG 使能位

		1 = 启用 PPG 模块, 引脚 PC3 变成 PPG 引脚 0 = 关闭 PPG 模块
6	RELDDB	比较器 1 下降沿“置位 PRLSEL”使能位 1 = 比较器 1 下降沿置位 PRLSEL, 使 PPGT 选择重载寄存器 PPRB 0 = 比较器 1 下降沿对 PRLSEL 无影响
5:0	PPGDLY[5:0]	PPG 延迟触发选择寄存器 0 = 旁路延迟触发功能 其它值 = 延迟触发时间为 (PPGDLY+1) 个系统时钟周期

### 18.11.7 PPGCON3 寄存器, 地址 0x19E

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	PSYNC	PPOL	PPGOEB	IOEB
Reset	-	-	-	-	0	0	1	1
Type	R-0	R-0	R-0	R-0	Q-RW	Q-RW	Q-RW	Q-RW

**Q-RW:** 当 UCFG1.7 为 1 时, 这些位可以由软件读写, 否则不能由软件改写, 其值只在上电配置过程确定。

Bit	Name	Function
7:4	-	保留位, 读 0
3	PSYNC	PPG 脉冲被触发后的同步选择 1 = 输出脉冲与系统时钟同步 0 = 输出脉冲不与系统时钟同步
2	PPOL	PPG 输出有效电平选择 1 = PPG 输出为低电平有效 0 = PPG 输出为高电平有效
1	PPGOEB	PC3 管脚功能选择 0 = PC3 用作 PPG 功能 1 = PC3 用作普通 IO
0	IOEB	IOFF 输出使能 0 = PC4 用作 IOFF 功能 1 = PC4 用作普通 IO

## 19 运算放大器

FT61F04X 内部集成了 1 个运算放大器，用于对模拟信号进行处理。

### 19.1 运放 0

运放 0 具有以下特点：

- 输入失调电压可校准
- 3MHz 的单位增益带宽
- 外部直接或者串 1kΩ 电阻到反相端
- 输出可经 60kΩ 电阻反馈到反相端
- 输出可连接外部管脚

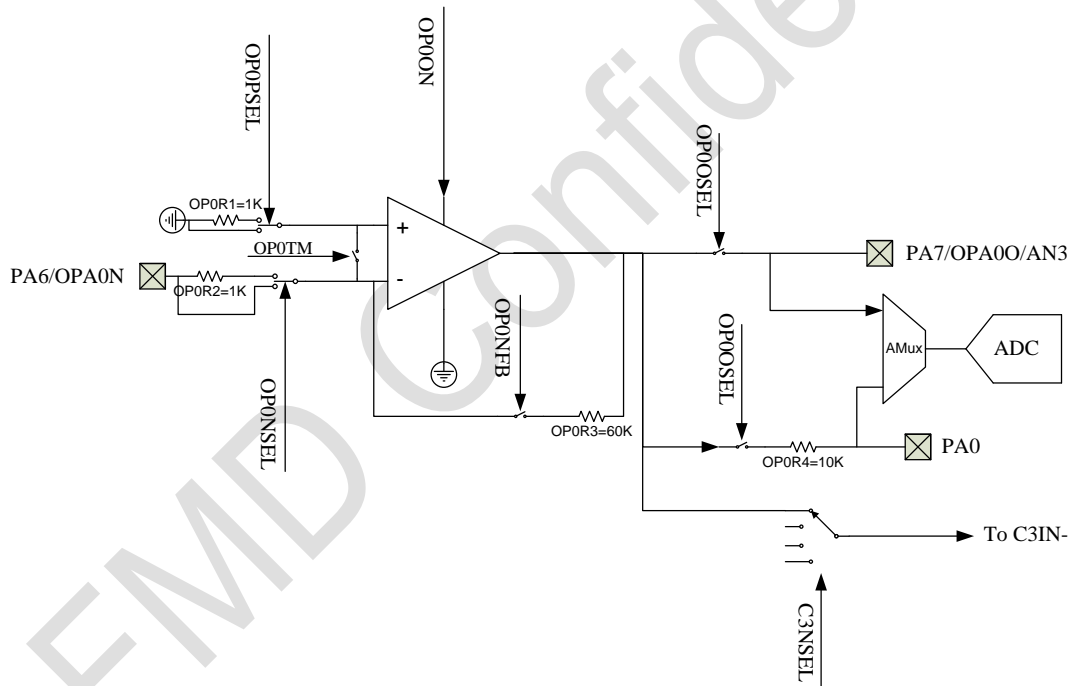


图 19.1 运放结构框图

#### 19.1.1 校准输入失调电压

运放的输入失调电压可以用软件校准，校准的步骤如下：

1. 把 OP0ON 置 1，打开运放；
2. TRISA.6 置 1，关闭 PA6 的数字输出驱动；
3. 把 OP0TM 置 1，让运放进入校准模式，运放两个端将短接；

4. 设置 OP0RS，选择同相或者反相端；
5. 配置 OP0NSEL 选择反相端，直接与 PA6 相连，或者与内部 1k 电阻串联后与 PA6 相连（此位为配置选项）；
6. 改变 OP0COF[4:0]直到 OP0OUT 发生改变
7. 把 OP0TM 清 0，运放恢复正常模式；

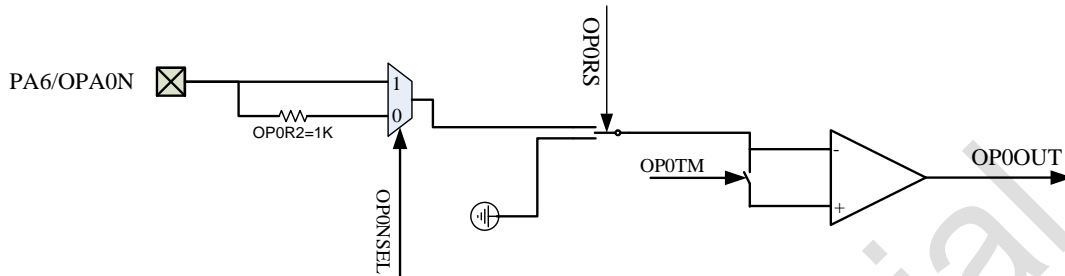


图 19.2 运放校准模式下的连接

## 19.1.2 运放 0 典型应用

### 19.1.2.1 使用内置负反馈

此时配置成 OP0NSEL=1，OP0OSEL=1；

OP0PSEL=0 或 1 都可以，区别是当 OP0PSEL=1 时，多串一个 1k 电阻接到运放正端。

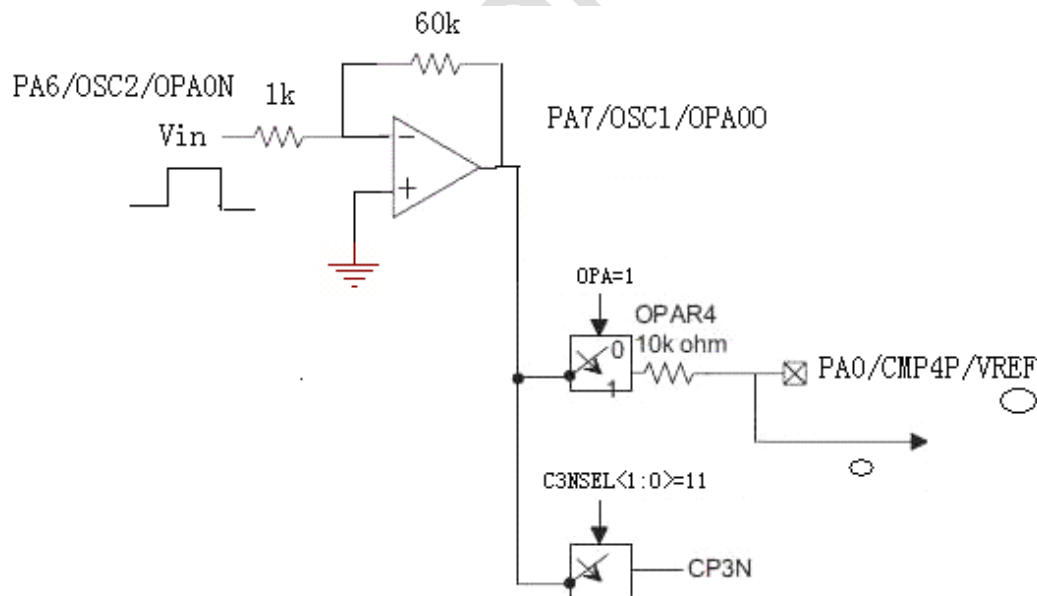


图 19.3 使用内部负反馈，对信号负相放大 60 倍

### 19.1.2.2 不使用内置负反馈

用户外部接 1k、500k 可变电阻、100n、100n 电容时。

此时配置成 OP0NSEL=0, OP0OSEL=0;

OP0PSEL=0 或 1 都可以, 区别是当 OP0PSEL=1 时, 多串一个 1k 电阻接到运放正端。

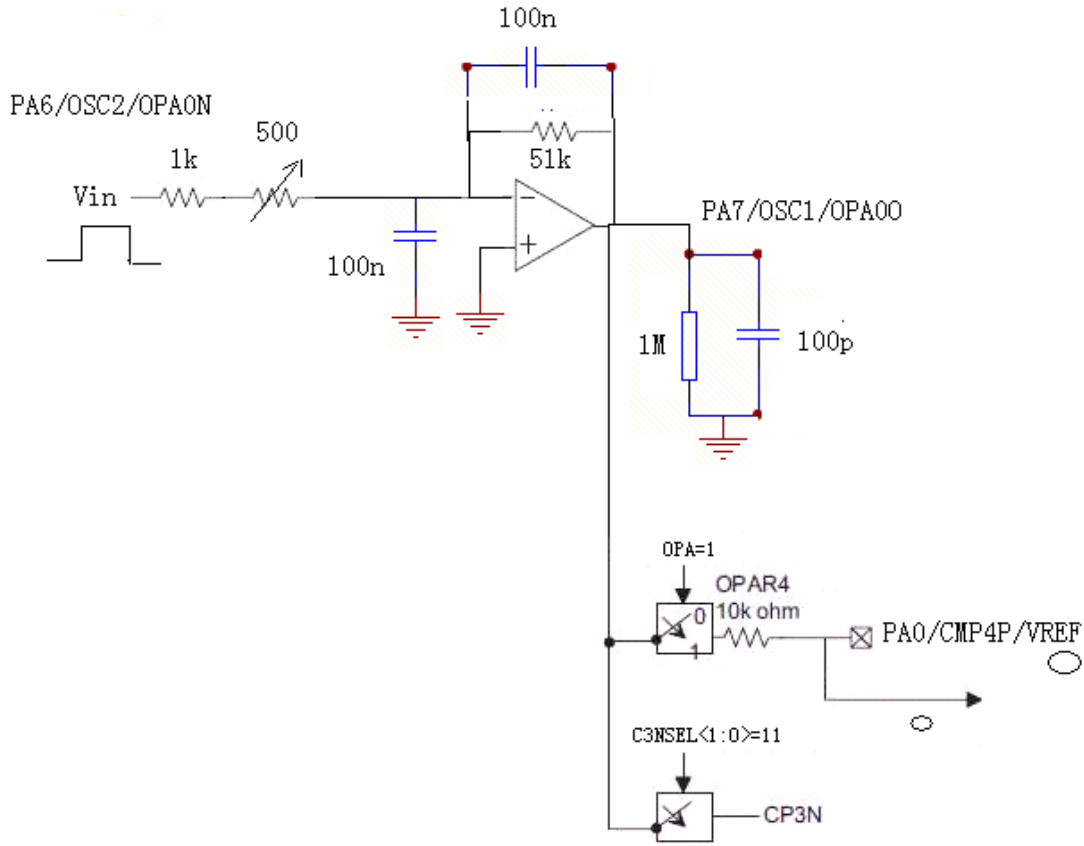


图 19.4 不使用内部负反馈, 对信号负相放大 34 倍到 51 倍

## 19.2 复位的影响

发生任何系统复位将使运放的使能位 OP0ON 清 0, 所以复位后运放是关闭的。

## 19.3 休眠模式下的影响

休眠不会影响运放的工作状态, 如果休眠前运放是打开的 (OP0ON 为 1), 则休眠后运放将继续工作并消耗一定的电流。

## 19.4 与运放 0 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
UCFG4	-	-	-	-	-	OP0OS	OP0PS	OP0NF	OP0NS	---- ----
						EL	EL	B	EL	
OP0CR0	0x19 4	OP0O UT	OP0TM	-	OP0COF[4:0]				x00- 0000	

### 19.4.1 OP0CR0 寄存器，地址 0x194

Bit	7	6	5	4	3	2	1	0
Name	OP0OUT	OP0TM	-	OP0COF[4:0]				
Reset	x	0	-	1	0	0	0	0
Type	RO	RW	R-0	RW	RW	RW	RW	RW

Bit	Name	Function
7	OP0OUT	运放 0 输出
6	OP0TM	运放 0 输入失调校准模式 1 = 进入失调校准模式 0 = 正常运放模式
5	-	保留位，读 0
4:0	OP0COF[4:0]	运放 0 输入失调校准位

### 19.4.2 UCFG4 配置寄存器

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	OP0OSEL	OP0PSEL	OP0NFB	OP0NSEL

详见 [2.2.7.5](#) 小节。

## 20 USB 充电口

片内集成了 3 对 USB 端口，USBA，USBB 以及 USBC，支持自适应快速充电 QC2.0 以及 QC3.0（USBA 口）协议，用于具有快充功能的移动电源或者其它便携式充电器。结合 2 个 5bit 的 DAC 和 10bit ADC，用户可以监测 USB 接口端的电压与电流，以识别连接到 USB 端口的是一个专用的充电器、便携设备、一般的 USB 接口或是带 USB 接口的充电装置。

USB 口具备以下特性：

- USBA 口内建下拉电阻
- USBA 和 USBB 口内建短路电阻
- 2 个 5 位 DAC，32 级电压输出
- DA+/DA-/DC-复用 ADC 通道

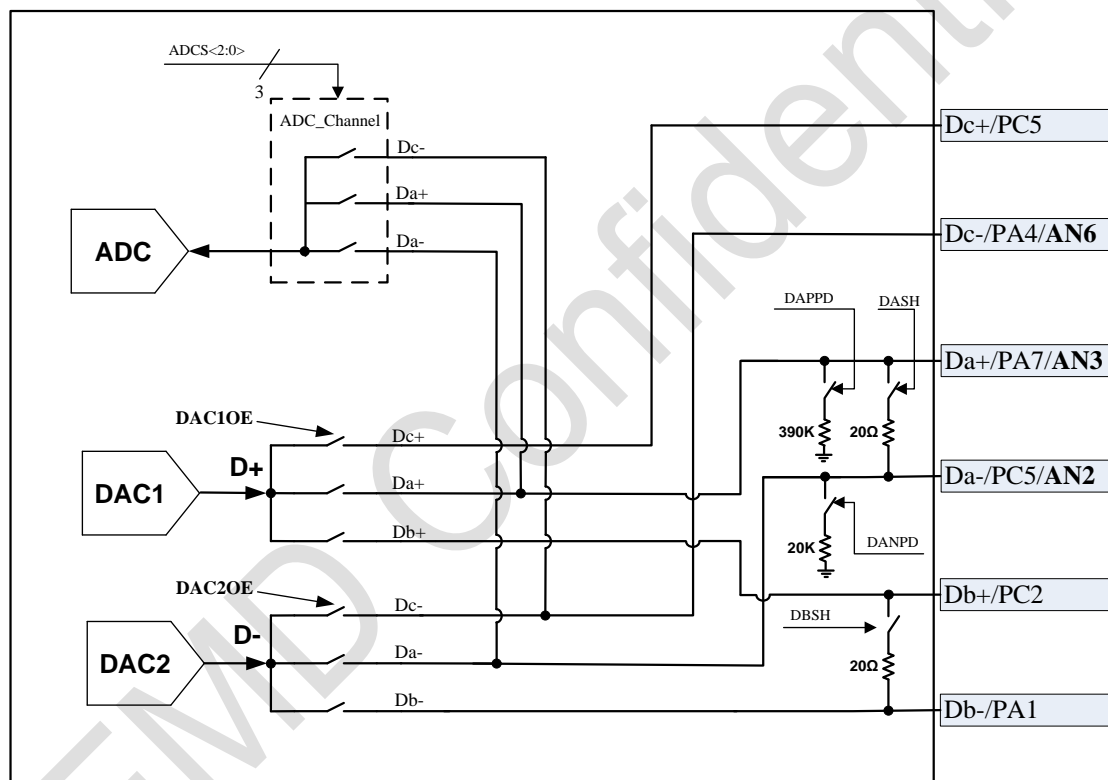


图 20.1 USB 原理框图

### 20.1 USBC 端口

USBC 端口可以提供特定的电压和电流，用于专用的充电器端口对移动电源进行充电。

#### 20.1.1 充电检测

充电器通过数据线连到手机时，DC+和 DC-默认是短接一起的。当系统检测到有充电器接入时，用



户程序通过正确设置 VCON1 寄存器的 VREGM 位以及 VREGHB 位, 让 DC+ 端口输出 DAC1 的 0.6V 电压, 同时 DC- 设置为 ADC 转换通道。MCU 在一段时间内 (时间长短由协议定义) 持续检查 DC- 电压。如果充电器支持 QC 充电协议, 则程序会看到 DC- 端有 0.6V 到 GND 的变化, 根据 DC- 的结果和移动电源自身充电电压要求, MCU 在 DC+、DC- 输出一定的电压组合, 以通知充电器自身所需电压。

下表给出了 QC2.0 协议的电压组合与充电电压关系。

Dc+ out	Dc- out	所需充电电压
3.6V	3.6V	20V
0.6V	0.6V	12V
3.6V	0.6V	9V
0.6V	GND	5V

表 20.1 QC2.0 D+/D- 电压与充电电压的关系

注意:

1. DAC 输出优先级比数字输出要高, 所以一旦设置 DAC 电压输出到管脚, 复用在相关管脚的其它功能将被自动禁止。

## 20.1.2 应用框图

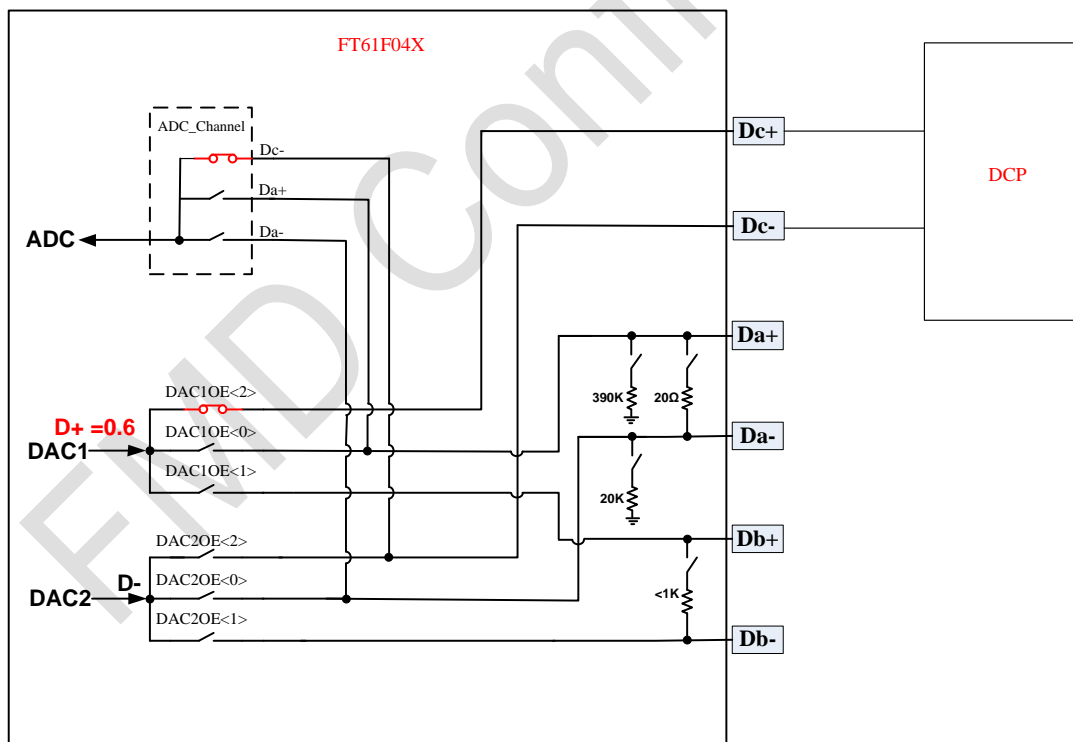


Fig 1 DAC1OE<2> 闭合, DAC2OE<2> 断开, DAC1 上给 D+=0.6V, 持续时间大于 1.25s, 同时, DC- 送往 ADC 通道的开关也闭合, 通过 ADC 检测 DC- 电压是否大于 0.325V。1.25s 后, 检测 DC- 是否小于 0.325V, 持续时间大于 1ms, 如果是, 表明 DCP 端支持快充。

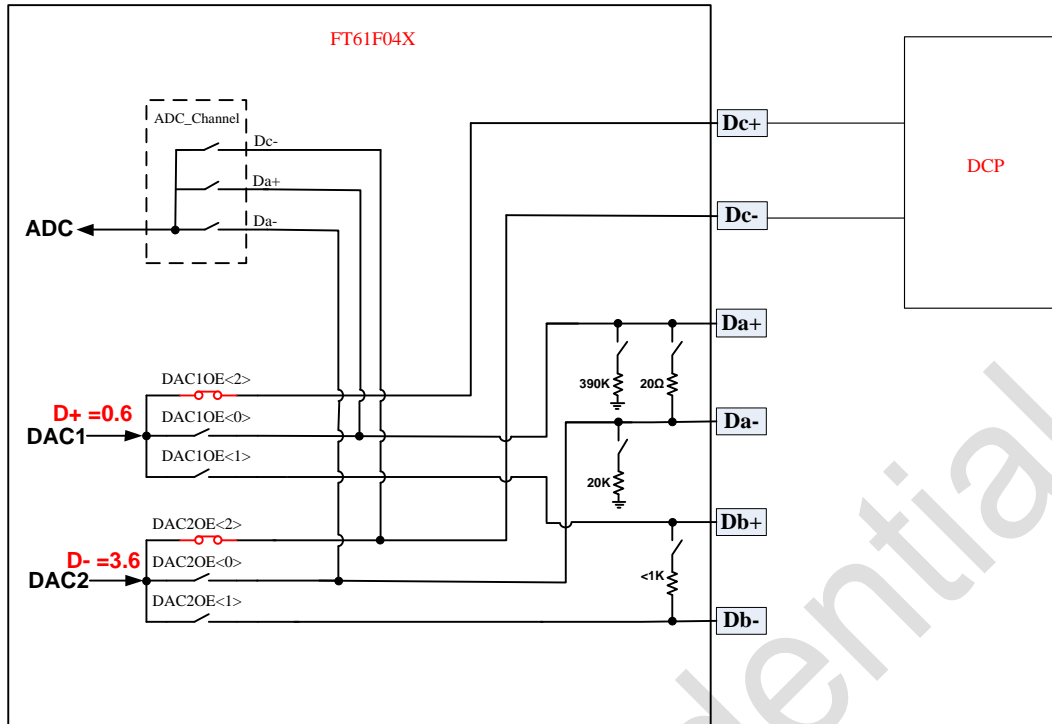


Fig. 2 检测通过以后，断开 DC-送往 ADC 的开关，闭合 DAC1OE<2>以及 DAC2OE<2>，在 DAC1 和 DAC2 上根据需要，分别输出电平组合或者输出脉冲（QC3.0），PD 端进入 QC2.0 和 QC3.0 充电模式。

## 20.2 USBA 和 USBB 端口

USBA, USBB 端口可提供特定的电压和电流，用于便携设备的充电。其中 USBA 口支持包括 QC3.0/2.0，三星和 APPLE 的设备，USB B 口支持三星和 APPLE 的设备。

### 20.2.1 基本工作原理

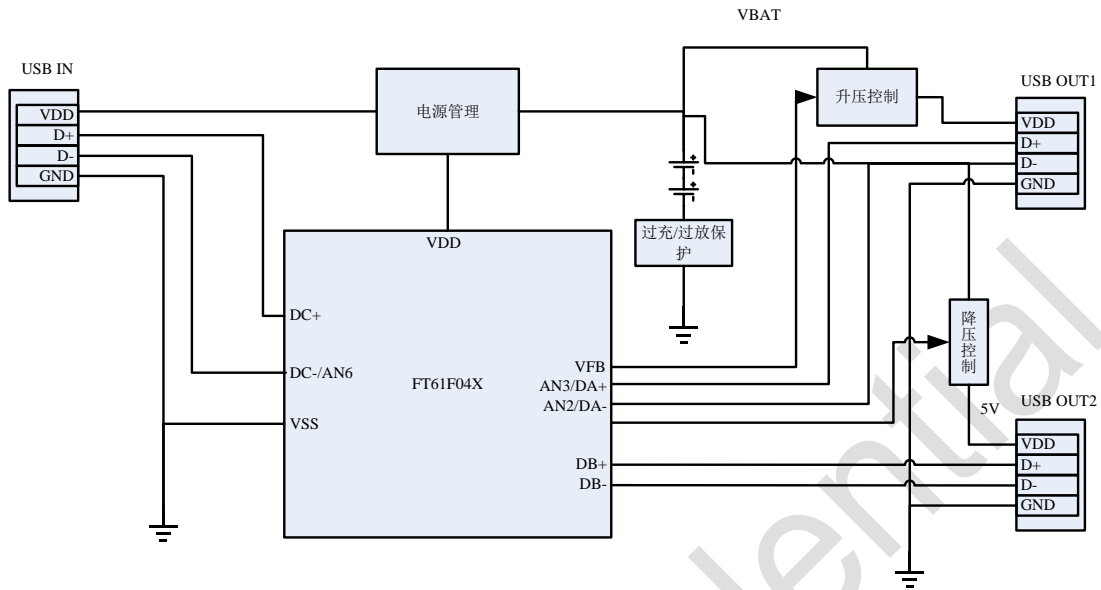


图 20.2 移动电源方案框图

上图是一种移动电源方案，下面给出 USB A 口检测充电设备握手过程流程图：

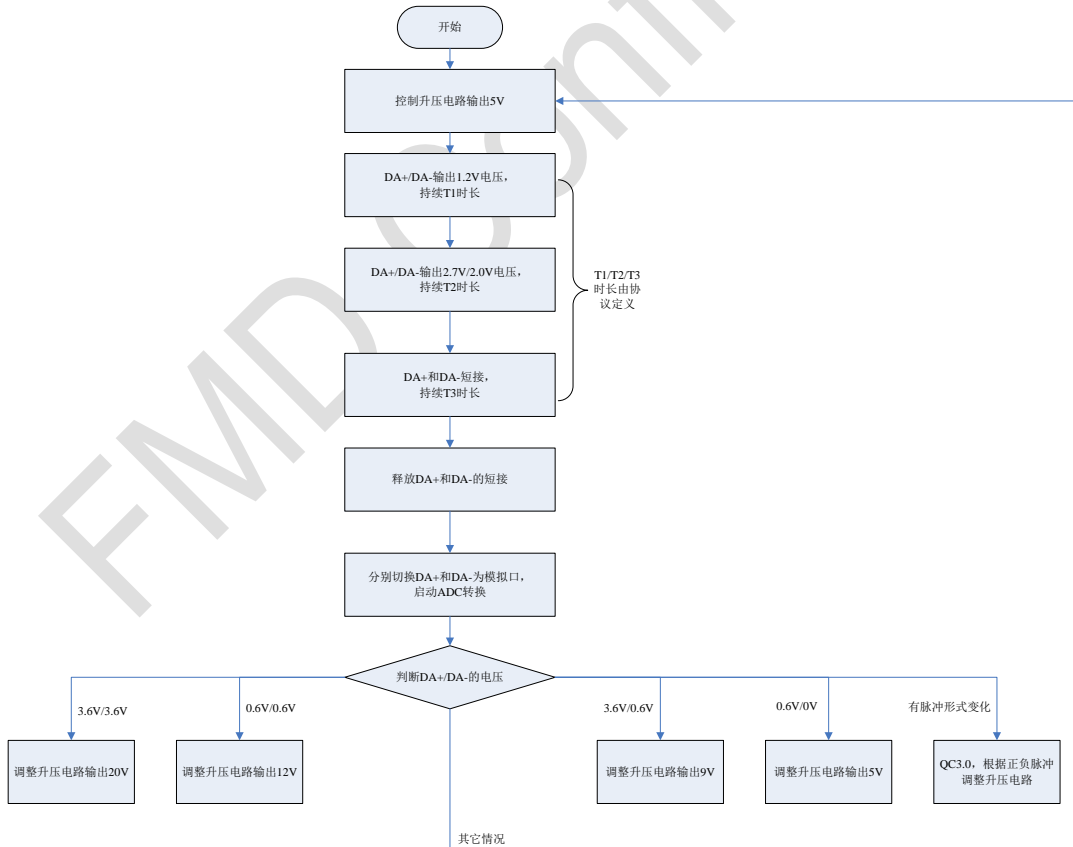


图 20.3 USB A 口检测充电设备流程图

### 20.2.2 USBA 应用框图

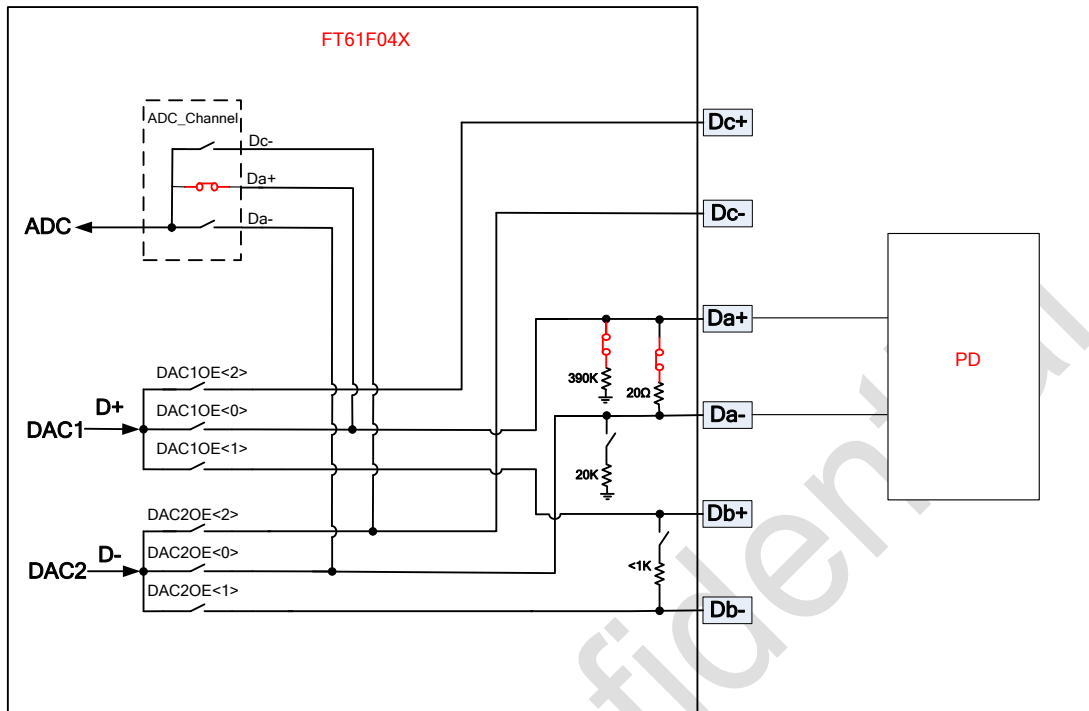


Fig. 3 DAC1OE<2:0>以及 DAC2OE<2:0>断开，默认 DA+/DA-的 20ohm 以及 390Kohm 开关闭合，同时闭合 DA+到 ADC 的开关。DCP 端通过 ADC 检测 DA+电压高于 0.325V 持续时间大于 1.25S，知道 PD 支持快充。

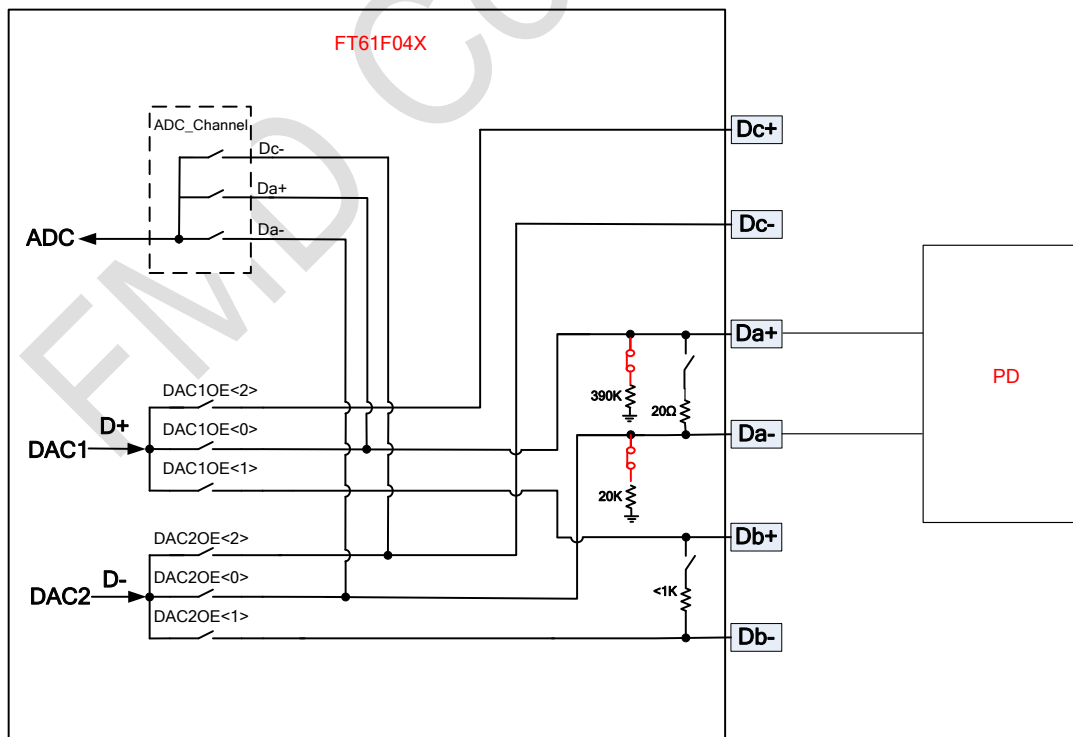


Fig. 4 DCP 端作为响应，断开 20ohm 电阻，同时把 DA-通过 20Kohm 电阻下拉到地，持续时间大于

1ms, 然后释放。PD 端在 1.25s 后检测 DA-端, 当检测到 DA-端电压 $<0.325V$ , 持续时间大于 1ms, 知道 DCP 端支持快充。QC2.0 识别成功。

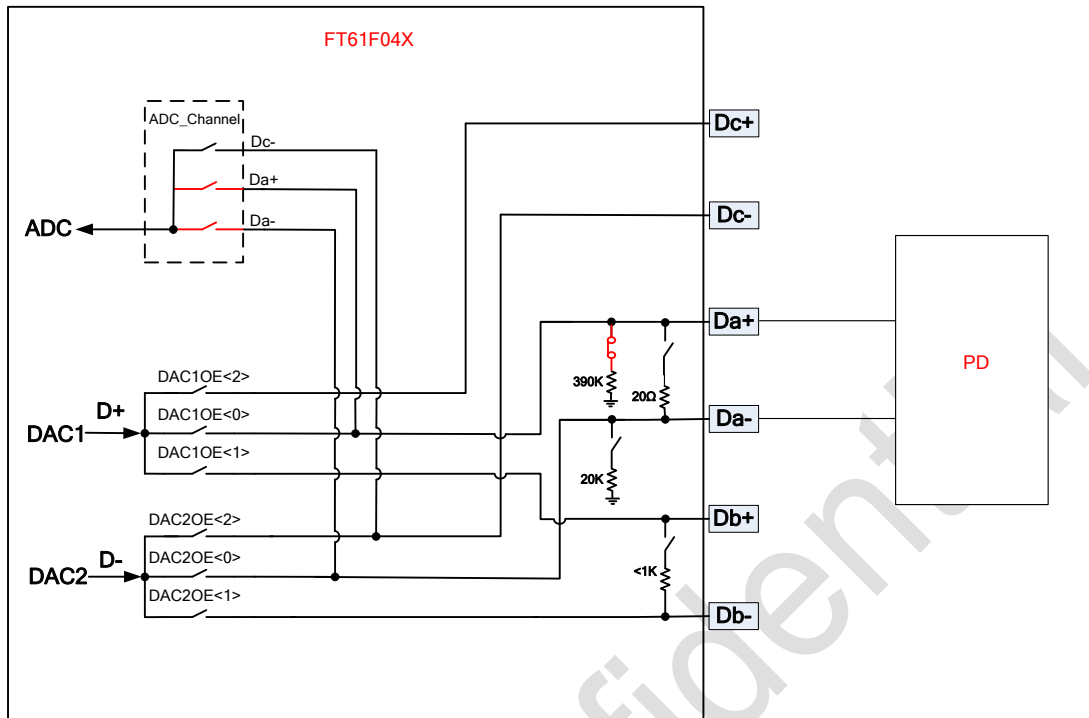


Fig. 5 QC2.0 识别成功后, PD 进入快充模式, 在 D+/D-上加载电压; 然后 DCP 端通过 ADC 分别检测 D+/D-上的电压并根据电压组合配合其它 IC 调整输出充电电压值。

### 20.2.3 USBB 应用框图

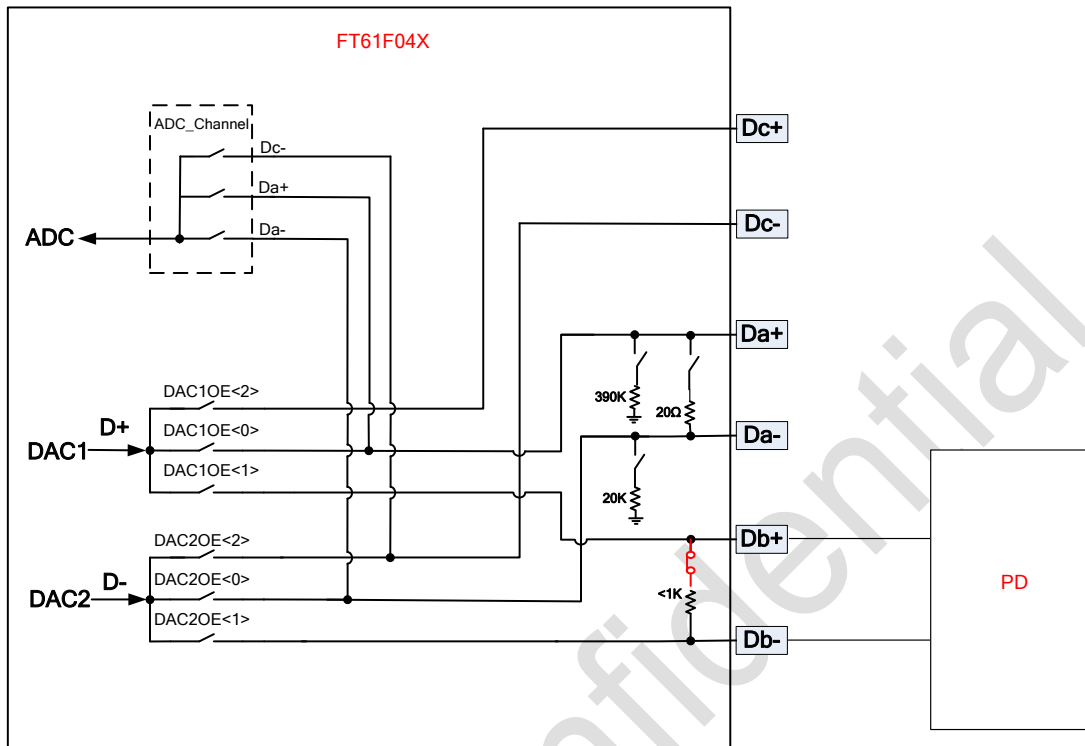


Fig. 6 DCP 端闭合短接电阻，PD 端发送 0.6V 到 DB+，40ms 后 DCP 端的 DAC1 DAC2 分别输出苹果、三星快充电压，PD 端检测到以后进入快充。其中 DCP 端短接的电阻不需要断开。

### 20.3 DAC 模块

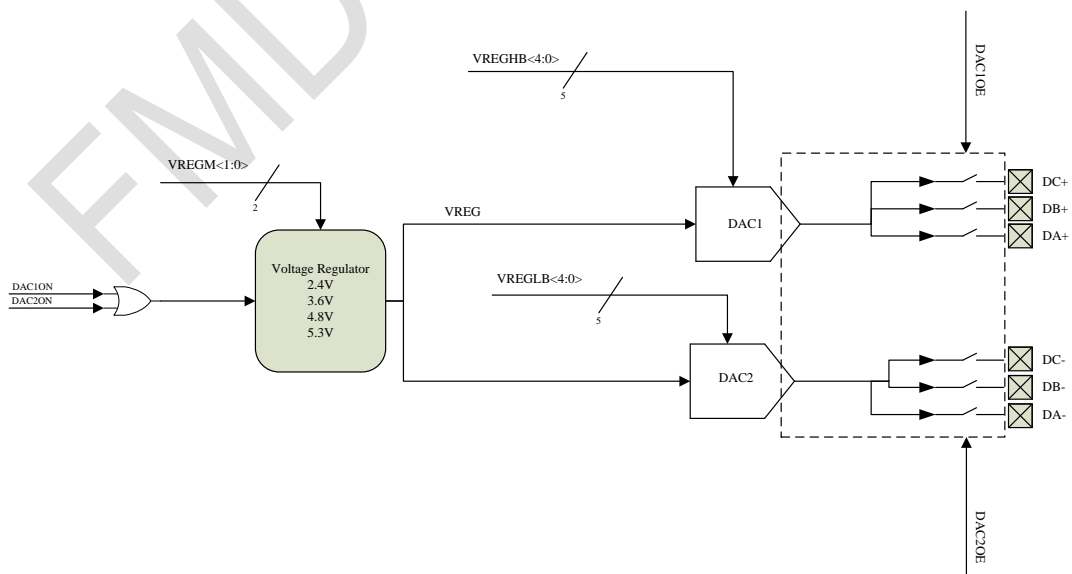


图 20.4 DAC 原理框图

输出电压由以下公式决定:

$$V_{OUT1} = V_{REG} * (VREGHB + 1) / 32 \quad (\text{公式 20.1})$$

$$V_{OUT2} = V_{REG} * (VREGLB + 1) / 32 \quad (\text{公式 20.2})$$

其中公式 20.1 为管脚 D+ 的电压输出, 公式 20.2 为管脚 D- 的电压输出, 参考电压  $V_{REG}$  可以是 2.4V、3.6V、4.8V、5.3V 中的一种, 由寄存器位 VREGM[1:0] 选择。

## 20.4 与 USB 口相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
DACOE	0x93	-	DAC2OE[2:0]			-	DAC1OE[2:0]			-000 -000
DACCON0	0x94	-	-	-	DBSH	-	DANPD	DAPPD	DASH	---0 -000
VCON1	0x108	-	VREGM[1:0]		VREGHB[4:0]					-000 0000
VCON2	0x109	-	DAC1ON	DAC2ON	VREGLB[4:0]					-000 0000
D2ICON	0x97	D2IEN	D2ISRC	D2IVAL[5:0]						0000 0000

### 20.4.1 DACOE 寄存器, 地址 0x93

Bit	7	6	5	4	3	2	1	0
Name	-	DAC2OE[2:0]			-	DAC1OE[2:0]		
Reset	-	0	0	0	-	0	0	0
Type	-	RW	RW	RW	-	RW	RW	RW

Bit	Name	Function
7	-	保留位, 不要写 1 读 0
6:4	DAC2OE[2:0]	DAC2 输出使能位, 高有效 DAC2OE[2], 控制 USBC 的 D- DAC2OE[1], 控制 USBB 的 D- DAC2OE[0], 控制 USBA 的 D-
3	-	保留位, 不要写 1 读 0
2:0	DAC1OE[2:0]	DAC1 输出使能位, 高有效 DAC1OE[2], 控制 USBC 的 D+ DAC1OE[1], 控制 USBB 的 D+ DAC1OE[0], 控制 USBA 的 D+
注意: DAC 的模拟输出比数字输出优先级高, 即如果 USB 的 D+/D- 设置了 DAC 输出, 硬件会自动把该 IO 的数字输出驱动以及上、下拉关闭。		

## 20.4.2 DACCON0 寄存器，地址 0x94

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	DBSH	-	DANPD	DAPPD	DASH
Reset	-	-	-	0	-	0	0	0
Type	-	-	-	RW	-	RW	RW	RW

Bit	Name	Function
7:5	-	保留位，不要写 1 读 0
4	DBSH	USBB 口的 D+/D-短接选择 1 = D+和 D-通过内部 200Ω 电阻短接 0 = D+和 D-不短接
3	-	保留位，不要写 1 读 0
2	DANPD	USBA 口的 D-下拉使能 1 = D-接内部 20KΩ 下拉电阻 0 = D-不接内部下拉
1	DAPPD	USBA 口的 D+下拉使能 1 = D+接内部 390KΩ 下拉电阻 0 = D+不接内部下拉
0	DASH	USBA 口的 D+/D-短接选择 1 = D+和 D-通过内部 20Ω 电阻短接 0 = D+和 D-不短接

### 20.4.2.1 VCON1 寄存器，地址 0x108

Bit	7	6	5	4	3	2	1	0
Name	-	VREGM[1:0]		VREGHB[4:0]				
Reset	-	2'b0		5'b0				
Type	-	RW		RW				

Bit	Name	Function								
7	-	保留位，不要写 1 读 0								
6:5	VREGM[1:0]	DAC 参考电压选择位 <table border="1" style="margin-left: 20px;"> <tr> <th>值</th> <th>DAC 参考电压 V<sub>REG</sub></th> </tr> <tr> <td>00</td> <td>2.4V</td> </tr> <tr> <td>01</td> <td>3.6V</td> </tr> <tr> <td>10</td> <td>4.8V</td> </tr> </table>	值	DAC 参考电压 V <sub>REG</sub>	00	2.4V	01	3.6V	10	4.8V
值	DAC 参考电压 V <sub>REG</sub>									
00	2.4V									
01	3.6V									
10	4.8V									



		11	5.3V
4:0	VREGHB[4:0]	当 VCON2 的 DAC1ON 位为 1 时, DAC1 输出电压为 $V_{OUT} = V_{REG} * (VREGHB + 1) / 32$	

### 20.4.3 VCON2 寄存器, 地址 0x109

Bit	7	6	5	4	3	2	1	0
Name	-	DAC1ON	DAC2ON	VREGLB[4:0]				
Reset	-	0	0	5'b0				
Type	-	RW	RW	RW				

Bit	Name	Function
7	-	保留位, 不要写 1 读 0
6	DAC1ON	DAC1 使能 1 = 使能 DAC1 0 = 关闭 DAC1
5	DAC2ON	DAC2 使能 1 = 使能 DAC2 0 = 禁止 DAC2
4:0	VREGLB[4:0]	当 VCON2 的 DAC2ON 位为 1 时, DAC2 输出电压为 $V_{OUT} = V_{REG} * (VREGLB + 1) / 32$

### 20.4.4 D2ICON 寄存器, 地址 0x97

Bit	7	6	5	4	3	2	1	0
Name	D2IEN	D2ISRC	D2IVAL[5:0]					
Reset	0	0	6'h0					

Bit	Name	Function
7	D2IEN	D2I 模块使能 1 = 使能 D2I, PB2 和 PB3 变为 R <sub>REF</sub> 和 V <sub>FB</sub> 功能管脚 0 = 关闭 D2I
6	D2ISRC	当 D2IEN 为 1 时, D2I 的源电流或者源电流选择 1 = 沉电流 0 = 源电流
5:0	D2IVAL[5:0]	D2I 的源电流、沉电流大小设置

## 21 芯片的电气特性

### 21.1 绝对极限参数

工作温度 .....	-40~+85℃
存储温度 .....	-40~+125℃
电源电压 .....	$V_{SS}-0.3V \sim V_{SS}+6.0V$
端口输入电压 .....	$V_{SS}-0.3V \sim V_{DD}+0.3V$

**注意：**如果运行条件超过了上述“绝对极限参数值”，即可能对器件造成永久性损坏。上述值仅为运行条件的极大值，我们不建议器件运行在该规范范围以外。器件长时间工作在绝对极限参数条件下，其稳定性可能受到影响。

### 21.2 内置高频振荡器(HFINTOSC)

电气参数	最小值 <sup>(1)</sup>	典型	最大值 <sup>(1)</sup>	单位	条件/备注
Ivdd 工作电流	—	30	—	uA	3V, typ
随温度变化范围	-5.4%	—	4.6%	—	-40~85℃, 3V
随电源电压变化范围	-0.5%	—	0.5%	—	2.0~ 5.5V, 25℃
启动时间	—	2.5	—	us	3V, 25℃
漏电流	—	0.8	2	nA	禁止快时钟和带隙基准
(校准) 频率	15.76	16	16.24	MHz	校准偏差

(1)数据基于特性值，并未生产测试

### 21.3 内置低频振荡器(LFINTOSC)

此低频振荡器有双模模式，一种模式下振动频率为 32KHz，另一种模式下振动频率为 256KHz。振荡频率模式由 OSCCON 寄存器中的 LFMOD 位控制，0 为 32KHz 模式，1 为 256KHz 模式（看门狗固定使用 32KHz，因此 LFMOD 没实际意义）。

电气参数	最小值 <sup>(1)</sup>	典型	最大值 <sup>(1)</sup>	单位	条件/备注
Ivdd 工作电流	—	1.1	—	uA	3V, 25℃
随温度变化范围	-2.1%	—	1.9%	—	-40 ~ 85℃ ,3V
随电源电压变化范围	-2%	8%	2%	—	2.0~ 5.5V ,25℃
振荡频率 (LFMOD=0)	—	32	—	KHz	3V, 25℃
启动时间	—	4.6	—	us	3V, 25℃
漏电流	—	0.15	1	nA	

(1)数据基于特性值，并未生产测试

## 21.4 低电压复位电路(LVR)

电气参数		最小值	典型	最大值 <sup>(1)</sup>	单位	条件/备注
LVR Tolerance	$V_{LVR}=2.0V$	-2	—	2	%	TT, 25°C
	$V_{LVR}=2.2V$	-2	—	2		
	$V_{LVR}=2.5V$	-2	—	2		
	$V_{LVR}=2.8V$	-2	—	2		
	$V_{LVR}=3.1V$	-2	—	2		
	$V_{LVR}=3.6V$	-2	—	2		
	$V_{LVR}=4.1V$	-2	—	2		
LVR delay		—	122	157	us	2.0V~5.5V, 25°C
Ivdd 工作电流		—	16.19	—	uA	3.3V, 25°C

(1) 数据基于特性值，并未生产测试

## 21.5 低电压侦测电路(LVD)

电气参数		最小值	典型	最大值 <sup>(1)</sup>	单位	条件/备注
LVD Tolerance	$V_{LVD}=2.0V$	-3	—	3	%	TT, 25°C
	$V_{LVD}=2.4V$	-3	—	3		
	$V_{LVD}=2.8V$	-3	—	3		
	$V_{LVD}=3.0V$	-3	—	3		
	$V_{LVD}=3.6V$	-3	—	3		
	$V_{LVD}=4.0V$	-3	—	3		
LVD delay		—	122	157	us	2.0V~5.5V, 25°C
Ivdd 工作电流		—	19.1	—	uA	3.3V

(1) 数据基于特性值，并未生产测试

## 21.6 上电复位电路(POR)

电气参数	最小值	典型	最大值	单位	条件/备注
Ivdd 工作电流	—	50	—	nA	3.3V
POR 电压	—	1.55	—	V	25°C, TT

## 21.7 I/O PAD 电路

电气参数	最小值	典型	最大值 <sup>(1)</sup>	单位	条件/备注
VIL	0	—	$0.3V_{DD}$	—	2.0V~5.5V, 25°C
VIH	$0.7 V_{DD}$	—	$V_{DD}$	—	2.0V~5.5V, 25°C
漏电流	-1	—	1	uA	5V

源电流	—	25.50	25.85	mA	5V, $V_{OH}=4.5V$ , 25°C
沉电流	—	22.95	23.06	mA	5V, $V_{OL}=0.5V$ , 25°C
上拉电阻	—	25	—	kΩ	5V
	—	43	—		3.3V
下拉电阻	—	93	—	kΩ	5V
	—	150	—		3.3V

(1) 数据基于特性值，并未生产测试

## 21.8 Comparator0~3 比较器电路

电气参数	最小值	典型	最大值	单位	条件/备注
工作电压	3	—	5.5	V	—
I <sub>vdd</sub> 工作电流	—	37	—	uA	5V, 25°C, 单个比较器电流
输入共模电压	0	—	$V_{DD} - 1.4$	V	—
输入偏移电压(校准前)	-15	—	+15	mV	$V_{DD} = 5V$
输入偏移电压(校准后)	-2*	—	+2*	mV	$V_{DD} = 5V$
迟滞(Hysteresis)	20	40*	60	mV	同上
响应时间(Response Time)	—	200	1000	nS	正常模式:输出低->高
	—	150	600	nS	正常模式:输出高->低
比较器 1 参考源 VR1	—	0.6~0.775 0.025 $V_{DD}$ / 级	—	—	5V, -20°C~85°C
比较器 2 参考源 VR2	—	0.6~0.775 0.025 $V_{DD}$ / 级	—	—	5V, -20°C~85°C
比较器 2 参考源 VR3	—	0.075~0.25 0.025 $V_{DD}$ / 级	—	—	5V, -20°C~85°C
比较器 3 参考源 VR4	—	0.6~0.775 0.025 $V_{DD}$ / 级	—	—	5V, -20°C~85°C

\* 为设计参考值

**注意:**

1. 输入失调电压的测试方法为同相端接 0.6V<sub>DD</sub>，反相端从 0V 扫描到 V<sub>DD</sub>;

## 21.9 Comparator4~5 比较器电路

电气参数	最小值 <sup>(1)</sup>	典型 <sup>(1)</sup>	最大值	单位	条件/备注
工作电压	2	—	5.5	V	

lvdd 工作电流	—	70	—	uA	3V, 25°C
输入共模电压	0	—	$V_{DD} - 1.5$	V	2.0V~5.5V, -40°C~85°C
输入偏移电压(Offset)	—	±5	±10	mV	同上
共模抑制比(CMRR)	55	—	—	dB	同上
迟滞 (Hysteresis)	—	0	—	mV	
响应时间(Response Time)	—	200	—	ns	正常模式:输出低->高
	—	150	—	ns	正常模式:输出高->低

(1) 数据基于特性值，并未生产测试

## 21.10 Comparator4~5 比较器参考电压设置

电气参数	最小值	典型	最大值	单位	条件/备注
相对精度	—	$V_{DD} / 16$	—	V	2.0V~5.5V, -20°C~85°C
绝对精度	—	—	1/2*	LSB	同上
单位电阻(unit resistor)	—	5000*	—	—	同上
稳定时间(Settle Time)	—	—	10	us	00000->11111

\* 为设计参考值

## 21.11 运算放大器 0

电气参数	最小值	典型	最大值 <sup>(1)</sup>	单位	条件/备注
lvdd 工作电流	—	61.713	63.537	uA	3.3V, 正常模式
工作电压	3.0	—	5.5	V	—
输入失调电压 1	-15	—	15	mV	未校准
输入失调电压 2	-2*	—	2*	mV	校准后
共模输入范围	$V_{SS}$	—	$V_{DD} - 1.4$	V	$V_{DD} = 5V$
增益带宽	—	2.2*	—	MHz	$R_L = 1M, C_L = 100pF$
电源电压抑制比	—	80*	—	dB	$V_{DD} = 5V$
共模抑制比	—	80*	—	dB	$V_{CM} = V_{DD}/2, V_{DD} = 5V$
SR 转换速率	—	1.8*	—	V/us	空载, 5V
OP0R1	0.75	1	1.25	kΩ	$T = 25C, V_{DD} = 5V$
OP0R3	0.75	1	1.25	kΩ	$T = 25C, V_{DD} = 5V$
OP0R3	45	60	75	kΩ	$T = 25C, V_{DD} = 5V$
OP0R4	—	10	—	kΩ	$T = 25C, V_{DD} = 5V$

\* 为设计参考值

(1) 数据基于特性值，并未生产测试

## 21.12 10bit ADC 电路

ADC 特性参数<sup>(1)</sup>

电气参数	最小值	典型	最大值	单位	条件/备注
I <sub>VDD</sub> 工作电流	—	60	150	uA	VDD=3, T=25℃, 系统时钟 16MHz, 转换时钟 Fosc/2, 参考电压 VDD
模拟输入电压	VSS	—	VREF	V	
分辨率	—	—	10	位	
外置参考电压	2	—	5.5	V	
单调性	—	保证	—		V <sub>SS</sub> ≤VAIN≤VREF+
总绝对误差 E <sub>ABS</sub>	—	—	±1	LSB	VREF=5.0V
积分误差 E <sub>IL</sub>	—	—	±1	LSB	VREF=5.0V
微分误差 E <sub>DL</sub>	—	—	±1	LSB	不丢码至 10 位, VREF=5.0V
满量程范围 E <sub>FS</sub>	2.2	—	5.5	V	
偏移误差 E <sub>OFF</sub>	—	—	±1	LSB	VREF=5.0V
增益误差 E <sub>GN</sub>	—	—	±1	LSB	VREF=5.0V
参考电压 VREF	2.2	—	—	V	
	2.5	—	V <sub>DD</sub> +0.3	V	确保 10 位精度的绝对最小值
转换时钟周期 TAD	—	2	—	us	VREF>3.0V
转换时钟数	—	11.5	—	TAD	
稳定时间(T <sub>ST</sub> )	—	200	—	us	VDD=5V, T=25℃
采样时间(T <sub>ACQ</sub> )	—	5	—	us	VREF=VDD=5V
	—	10	—	us	VREF=VDD=3V
建议的模拟电压源阻抗(ZAI)	—	—	10	kΩ	—

(1) 数据基于特性结果, 未经生产测试。

除非另外说明, 否则“典型”值一栏的数据都是在 5.0V, 25℃ 的条件下给出。

#### ADC<sub>Vref</sub> 特性参数

电气参数	最小值	典型 <sup>(1)</sup>	最大值	单位	条件/备注
内置参考电压 ADC <sub>Vref</sub>	1.99	2	2.01	V	VDD=2.5~5.5V, T=25℃
	2.99	3	3.01	V	VDD=3.5~5.5V, T=25℃
内置参考电压稳定时间 T <sub>V<sub>RINT</sub></sub>	—	100	—	us	VDD=5V, T=25℃

(1) 数据基于特性结果, 未经生产测试。

## 21.13 稳压器输出电路

电气参数	最小值	典型	最大值 <sup>(1)</sup>	单位	条件/备注
输出电压 V <sub>out</sub>	—	2.4*	—	V	VREGM=00, V <sub>DD</sub> =3.3~5.5V
	—	3.6*	—	V	VREGM=01, V <sub>DD</sub> =3.8~5.5V
	—	4.8*	—	V	VREGM=10, V <sub>DD</sub> =5~5.5V
	—	5.24*	—	V	VREGM=11,

					$V_{DD}=5.5V$
输出电流	—	200	—	uA	$V_{DD}=3.3\sim 5.5V$
工作电流	—	76.6	78.976	uA	$V_{DD}=3.3V$
相对精度	—	$V_{out}/32$	—	V	

\* 为设计参考值

(1) 数据基于特性值，并未生产测试

## 21.14 6bit D2I 电路

电气参数	符号	条件	最小值	典型	最大值	单位
Supply Voltage Range	$V_{DD}$	—	3.4*	—	5.5	V
Reference Pin Voltage	VR	Rref=75K		1.5		V
Toggle Up Current Source Step	$\Delta I_T(UP)$	Rref=75K	—	2	—	uA
Toggle Down Current Source Step	$\Delta I_T(DO)$	Rref=75K	—	2	—	uA
Output Current Accuracy (配置输出最大电流)	—	Rref=75K	68.2*	70	71.8*	uA
Operating Current	Icc	—	—	200	—	uA
Standby Current	Istandby	—	—	1	—	uA

\* 为设计参考值，非测试值

## 21.15 总体工作电流( $I_{VDD}$ )

电气参数	最小值	典型	最大值 <sup>(1)</sup>	单位	条件/备注
正常模式	—	1.515	1.548	mA	3.0V, 16MHz
	—	1.100	1.127	mA	3.0V, 8MHz
	—	0.666	0.680	mA	3.0V, 4MHz
	—	0.286	0.293	mA	3.0V, 1MHz
	—	0.045	0.048	mA	3.0V, 32KHz
休眠模式 (Sleep, WDT OFF, LVR OFF)	—	0.3	3	uA	3.3V
休眠模式 (Sleep, WDT ON, LVR OFF)	—	2.363	3.275	uA	3.3V
休眠模式 (Sleep, WDT OFF, LVR ON)	—	16.49	19.4	uA	3.3V
休眠模式 (Sleep, WDT ON, LVR ON)	—	18.553	19.675	uA	3.3V
休眠模式 (Sleep, WDT OFF, LVR OFF, LVD ON)	—	19.4	21.80	uA	3.3V

(1) 数据基于特性值，并未生产测试

注意：

1. 在环境温度 25°C 下测试；
2. 工作电流的测试条件为所有 IO 输出 0；
3. 睡眠电流的测试条件为 IO 处于输入模式，IO 外部下拉；
4. 使用 32KHz 内部 RC。

## 21.16 AC 电气参数

电气参数	最小值	典型	最大值	单位	条件/备注
指令周期 (Tins)	250	—	—	ns	系统时钟 HFINTOSC
	125	—	—	us	系统时钟 LFINTOSC
TOCKI 输入周期	(Tins+40)/N 和 20 中较大者	—	—	ns	N = 预分频值 (2, 4, ..., 256)
上电复位保持时间 (Tdrh)	—	4.2	—	ms	T=25°C, PWRT disable
外部复位脉冲宽度 (Trst)	2000	—	—	ns	T=25°C
WDT 周期 (Twdt)	—	1	—	ms	无预分频, WDTPS<3:0>=0000

注：除特殊说明，特性测试条件为：T=-40~85°C，VDD=2.0~5.5V，VSS=0V。

## 21.17 时序图

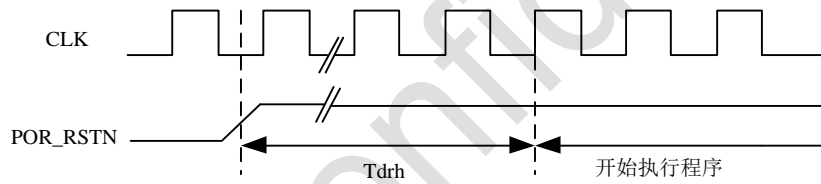


图 21.1 上电复位时序图

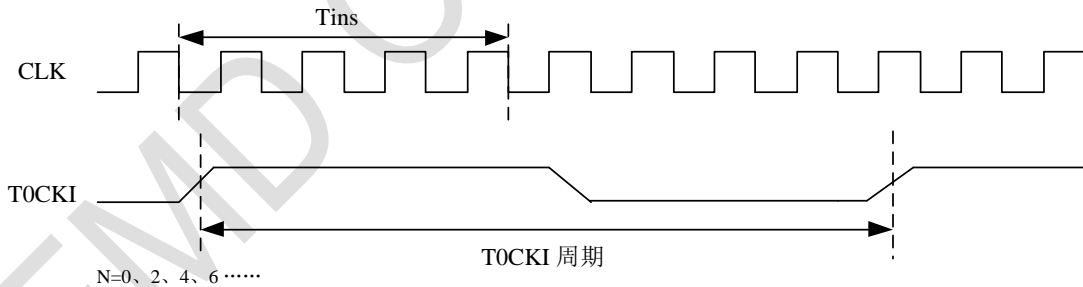


图 21.2 TOCKI 时序图 (4T 指令周期)



### 21.18 直流和交流特性图表

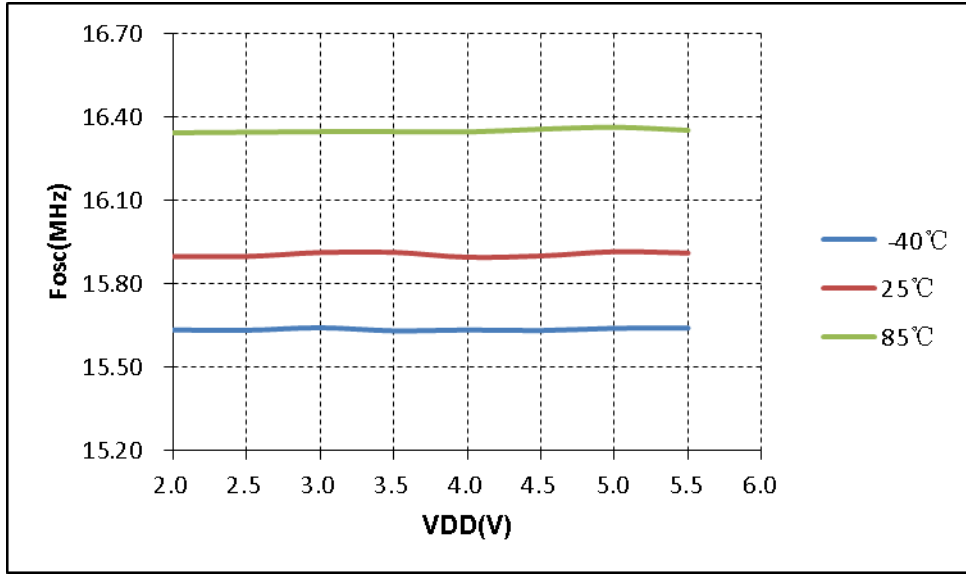


图 21.3 不同温度下 Fosc 与 VDD 曲线

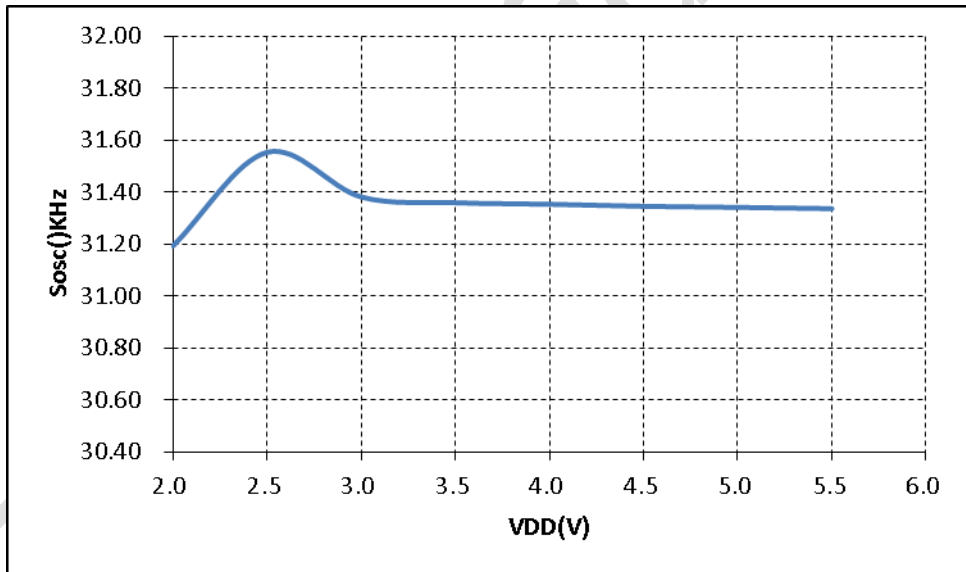


图 21.4 Sosc 与 VDD 关系曲线

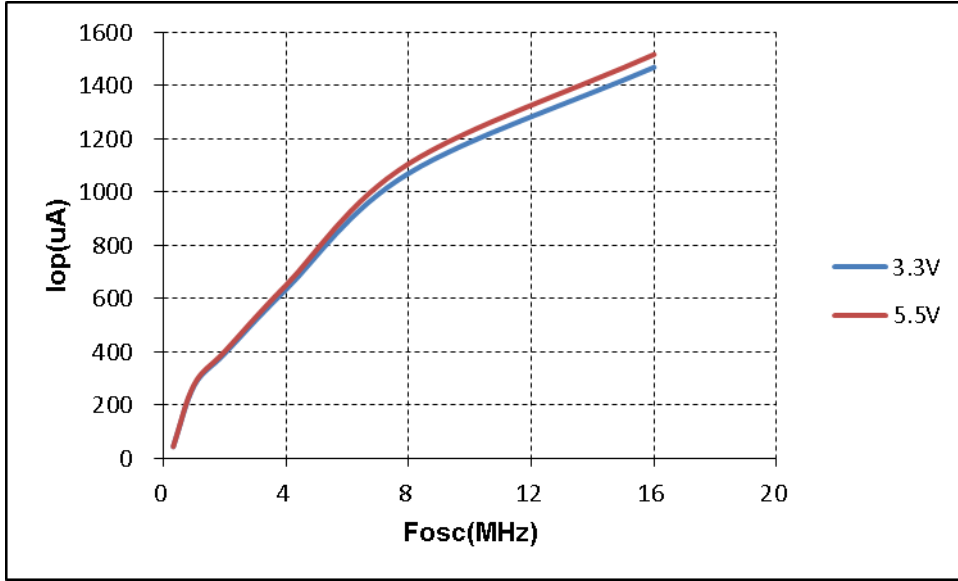


图 21.5 不同 VDD 下 Fosc 与工作电流曲线 (T=25°C)

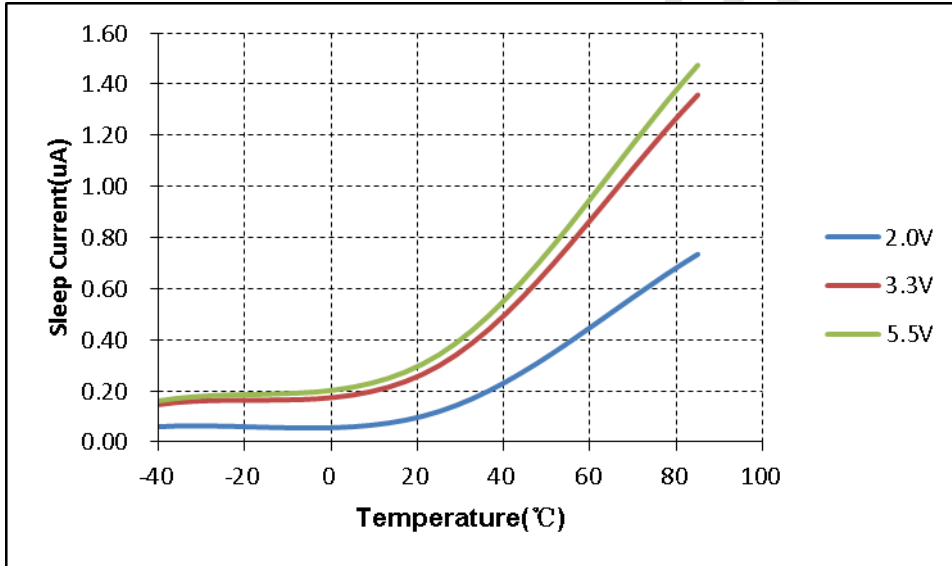


图 21.6 不同电压下睡眠电流与温度曲线 (WDT、LVD 和 LVR 禁止)

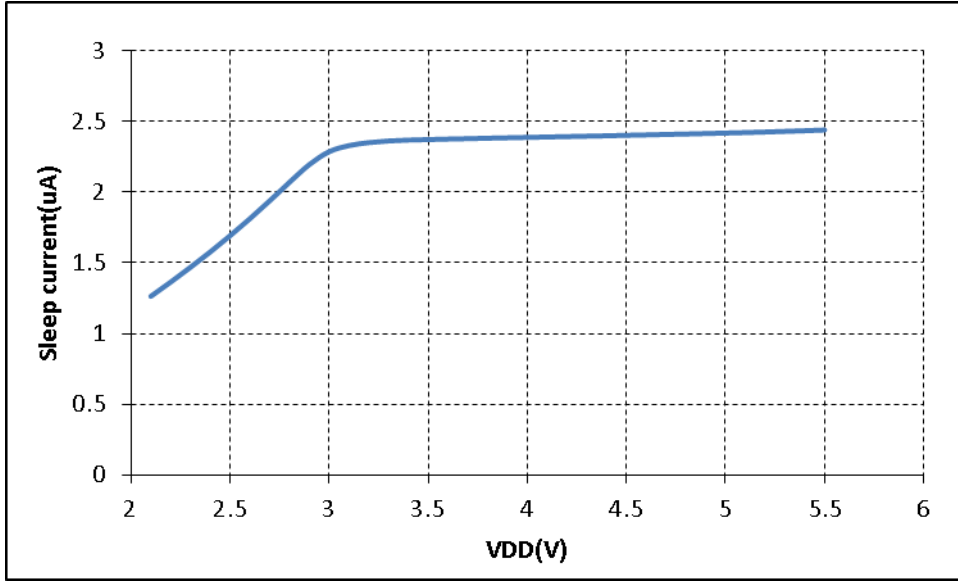


图 21.7 睡眠电流与 VDD 曲线 (WDT 使能, LVD 和 LVR 禁止)

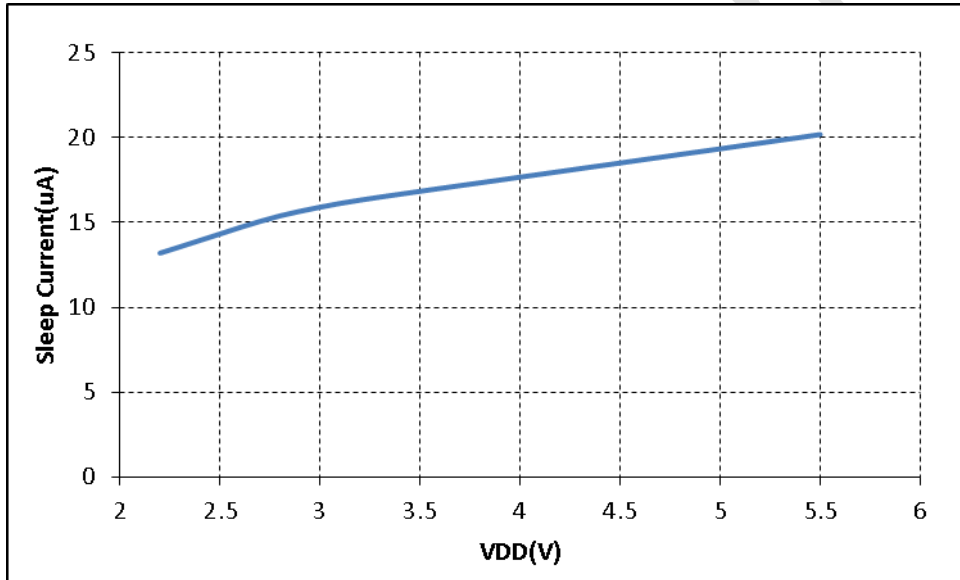


图 21.8 睡眠电流与 VDD 曲线 (LVR 使能, WDT 和 LVD 禁止)

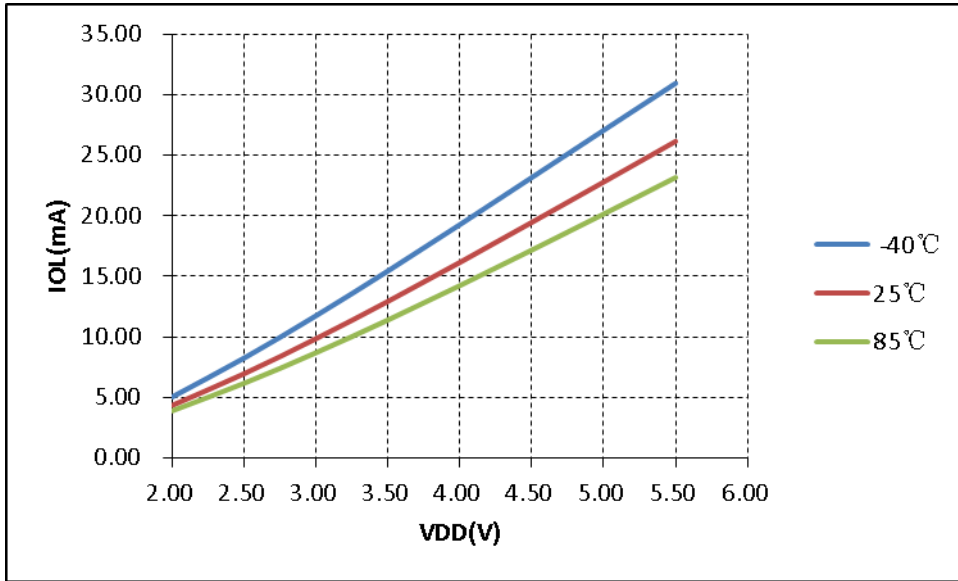


图 21.9 不同温度下 IOL 与 VDD 曲线 (VOL=0.5V)

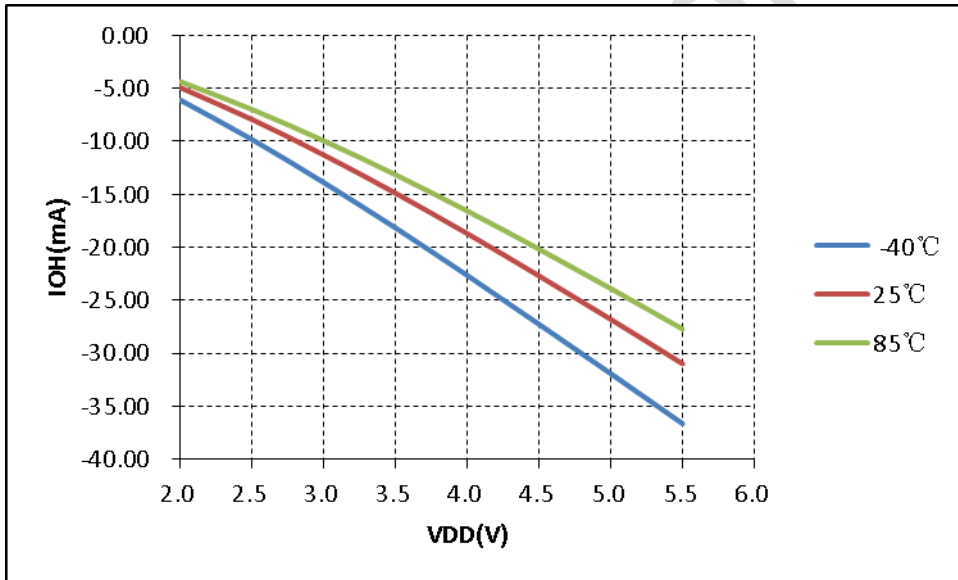


图 21.10 不同温度下 IOH 与 VDD 曲线 (VOH=4.5V)

## 22 指令集列表

本芯片采用精简指令架构，一共 37 条指令，以下是各指令的描述。

汇编语法	功能	运算	状态位
BCR R, b	Bit clear	0-> R(b)	NONE
BSR R, b	Bit set	1-> R(b)	NONE
BTSC R, b	Bit test, skip if 0	Skip if R(b)=0	NONE
BTSS R, b	Bit test, skip if 1	Skip if R(b)=1	NONE
NOP	No operation	None	NONE
CLRWDT	Clear WDT	0-> WDT	/PF, /TF
SLEEP	ENTER SLEEP MODE	0-> WDT, STOP OSC	/PF, /TF
STTMD	Store W TO TMODE	W-> TMODE	NONE
CTLIO R	Control IO direction reg	W-> IODIRr	NONE
STR R(MOVWF)	Store W to reg	W-> R	NONE
LDR R, d(MOVF)	Load reg to d	R-> d	Z
SWAPR R, d	Swap halves reg	[R(0-3)R(4-7)]-> d	NONE
INCR R, d	Increment reg	R+ 1-> d	Z
INCRSZ R, d	Increment reg, skip if 0	R+ 1-> d	NONE
ADDWR R, d	Add W and reg	W+ R-> d	C, HC, Z
SUBWR R, d	Sub W from reg	R- W-> d R+ /W+ 1-> d	C, HC, Z
DECR R, d	Decrement reg	R- 1-> d	Z
DECRSZ R, d	Decrement reg, skip if 0	R- 1-> d	NONE
ANDWR R, d	AND W and reg	R& W-> d	Z
IORWR R, d	Inclu.OR W and reg	W  R-> d	Z
XORWR R, d	Exclu.OR W and reg	W^ R-> d	Z
COMR R, d	Complement reg	/R-> d	Z
RRR R, d	Rotate right reg	R(n)-> R(n-1), C-> R(7), R(0)-> C	C
RLR R, d	Rotate left reg	R(n)-> R(n+1), C-> R(0), R(7)-> C	C
CLRW	Clear working reg	0-> W	Z
CLRR R	Clear reg	0-> R	Z
RETI	Return from interrupt	Stack-> PC, 1-> GIE	NONE
RET	Return from subroutine	Stack-> PC	NONE
LCALL N	Long CALL subroutine	N-> PC,	NONE

		PC+1-> Stack	
LJUMP N	Long JUMP address	N-> PC	NONE
LDWI I(MOVLW)	Load immediate to W	I-> W	NONE
ANDWI I	AND W and imm	W& I-> W	Z
IORWI I	Inclu.OR W and imm	W  I-> W	Z
XORWI I	Exclu.OR W and imm	W^ I-> W	Z
RETW I	Return, place imm to W	Stack-> PC, I-> W	NONE
ADDWI I	Add imm to W	W+I-> W	C, HC, Z
SUBWI I	Subtract W from imm	I-W-> W	C, HC, Z

## 操作码字段说明

Field	Description
R(F)	SFR/GPR address
W	Working register
b	Bit address within the 8-bit register/ram
I/Imm(k)	Immediate data
X	Don't care, may be 0 or 1
d	Destination select 0: store result in W 1: store result in register/ram
N	Immediate program address
PC	Program counter
TMODE	SFR TMODE <sup>1</sup>
IODIRr	SFR IODIR, r can be A, B, C
C	Carry bit
HC	Half carry
Z	Zero flag
/PF	Power down flag
/TF	Time out flag

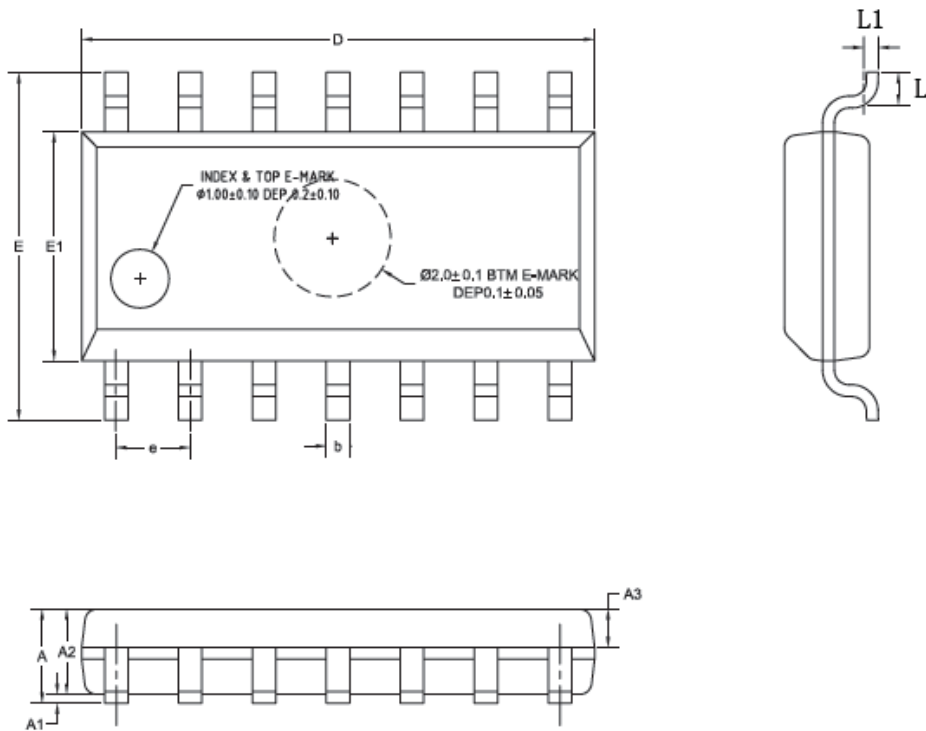
## 注意:

- 在 FT61F04X 系列芯片里, TMODE 寄存器是指 OPTION, 即 STTMD 指令的操作是把 W 存到 OPTION;

## 23 芯片封装信息

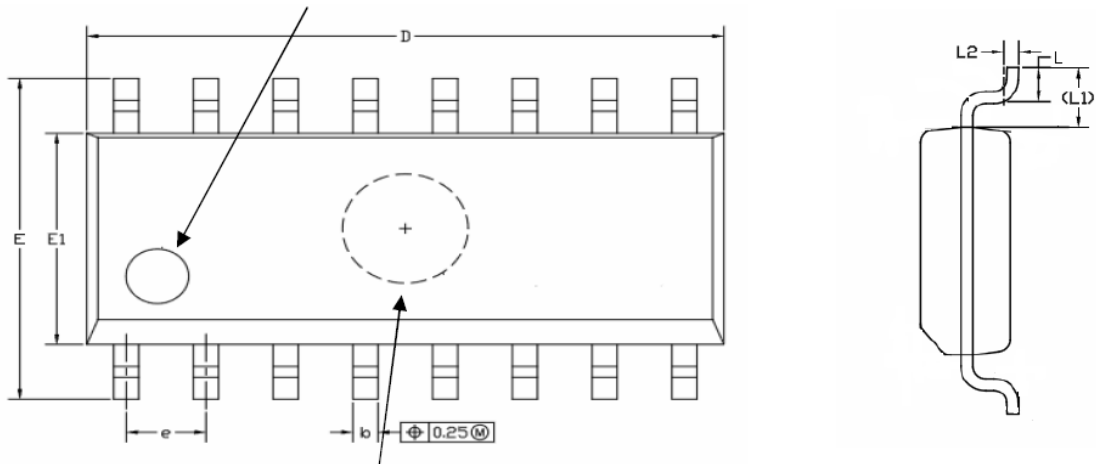
本芯片采用 SOP14、SOP16、TSSOP20、SOP20、PDIP20、QFN20 和 TSSOP14 封装方式，具体封装尺寸信息如下。

SOP14 封装尺寸如下：

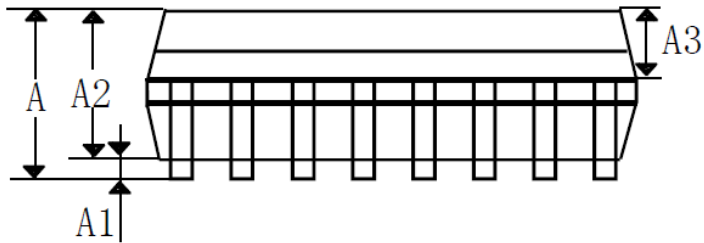


SYMBOL	MIN	NOM	MAX
A	—	—	1.70
A1	0.10	0.15	0.20
A2	1.40	1.45	1.50
A3	0.62	0.65	0.68
b	0.37	0.40	0.42
D	8.71	8.81	8.91
E	5.90	6.00	6.10
E1	3.80	—	3.95
e	1.27BSC		
L	0.50	0.60	0.70
L1	0.25BSC		

SOP-16 封装尺寸如下



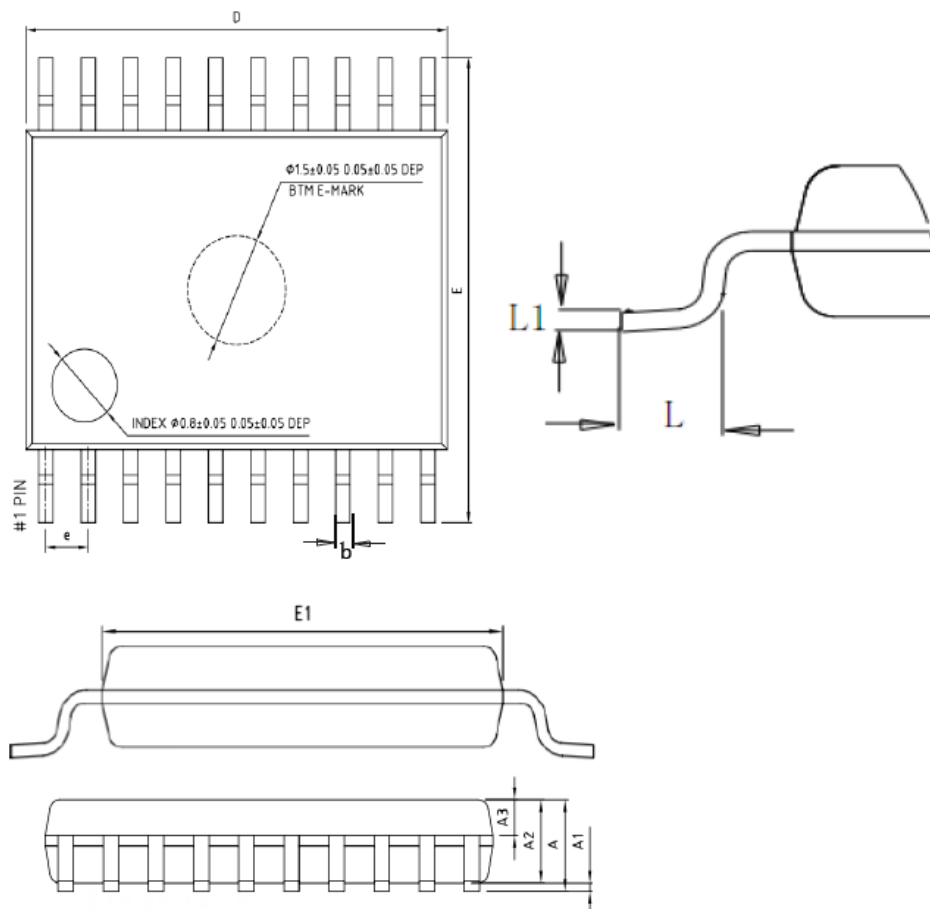
$\Phi 2.0 \pm 0.05$  DEP  $0.1 + 0.03 / -0.05$



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	1.700	-	0.066
A1	0.100	0.200	0.039	0.008
A2	1.420	1.480	0.554	0.058
A3	0.620	0.680	0.242	0.027
D	9.960	10.160	3.884	0.396
E	5.900	6.100	2.301	0.238
E1	3.870	3.930	1.509	0.153
b	0.370	0.430	0.144	0.017
e	1.240	1.300	0.484	0.051
L	0.500	0.700	0.195	0.027
L1	1.050(REF)		0.041(REF)	
L2	0.250(BSC)		0.010(BSC)	

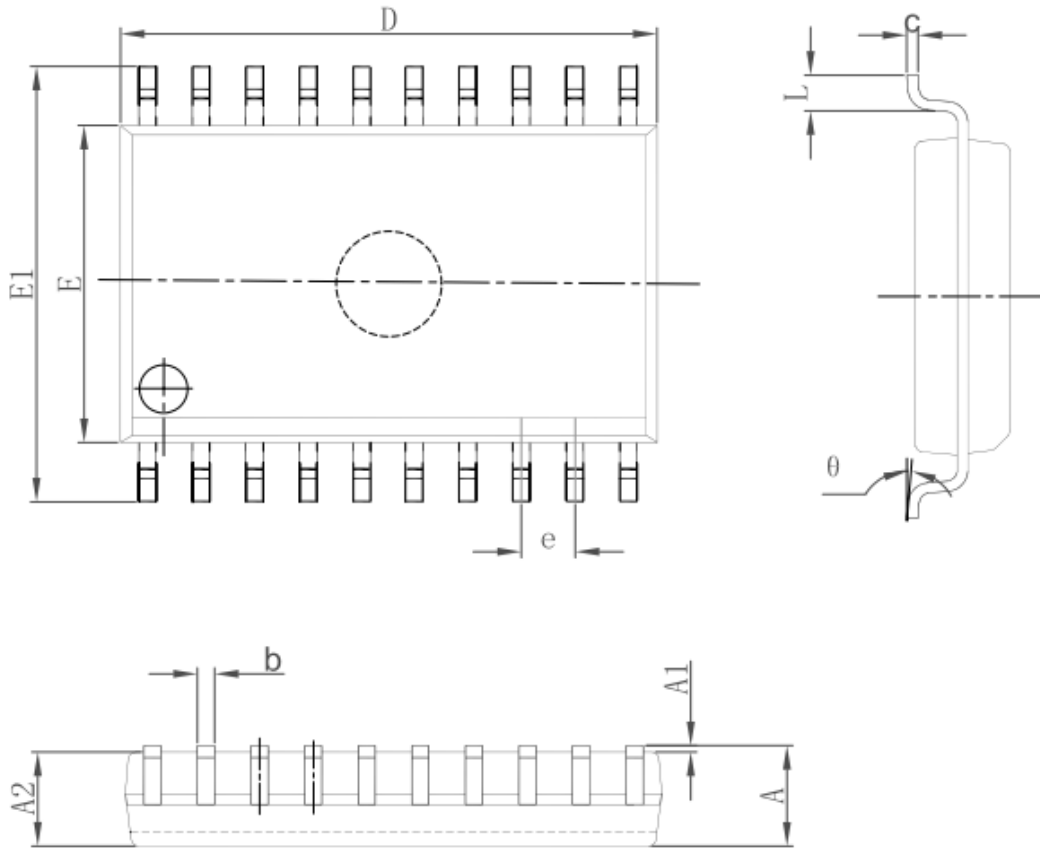


TSSOP-20 封装尺寸如下:



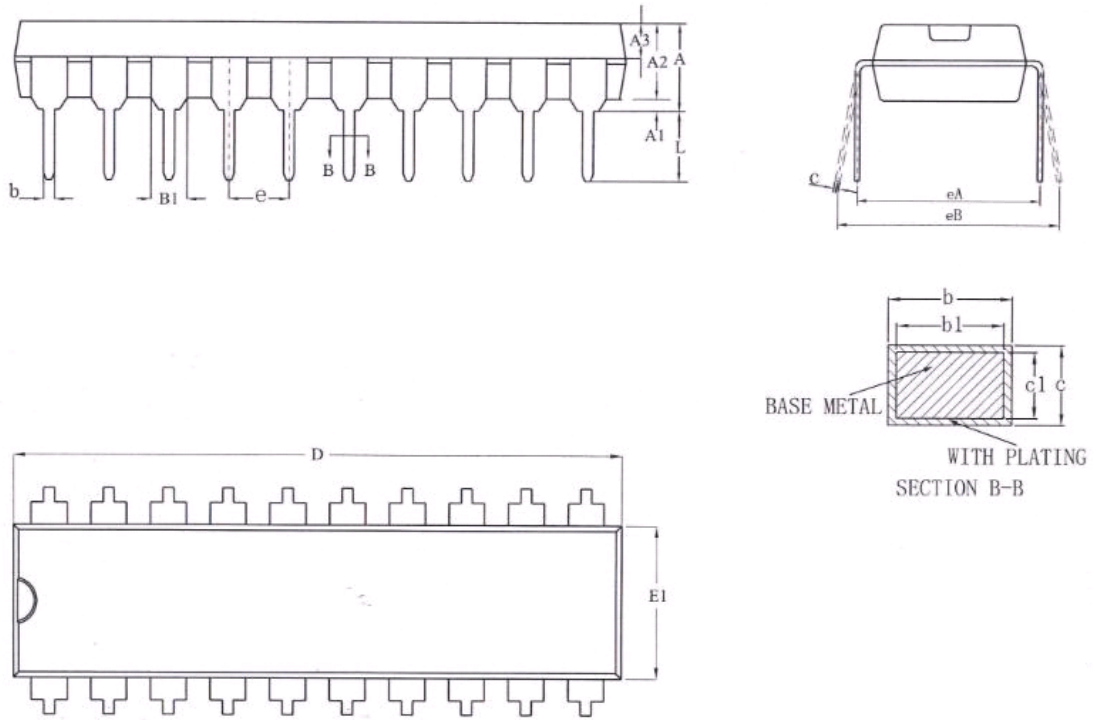
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	1.070	1.230	0.042	0.048
A1	0.050	0.150	0.002	0.006
A2	0.950	1.050	0.037	0.041
A3	0.411	0.461	0.016	0.018
b	0.200	0.280	0.008	0.011
D	6.400	6.600	0.250	0.257
E	6.300	6.500	0.246	0.254
E1	4.300	4.500	0.168	0.176
e	0.625	0.675	0.024	0.026
L	0.550	0.650	0.021	0.025
L1	0.25(BSC)		0.010(BSC)	

SOP-20 封装尺寸如下:



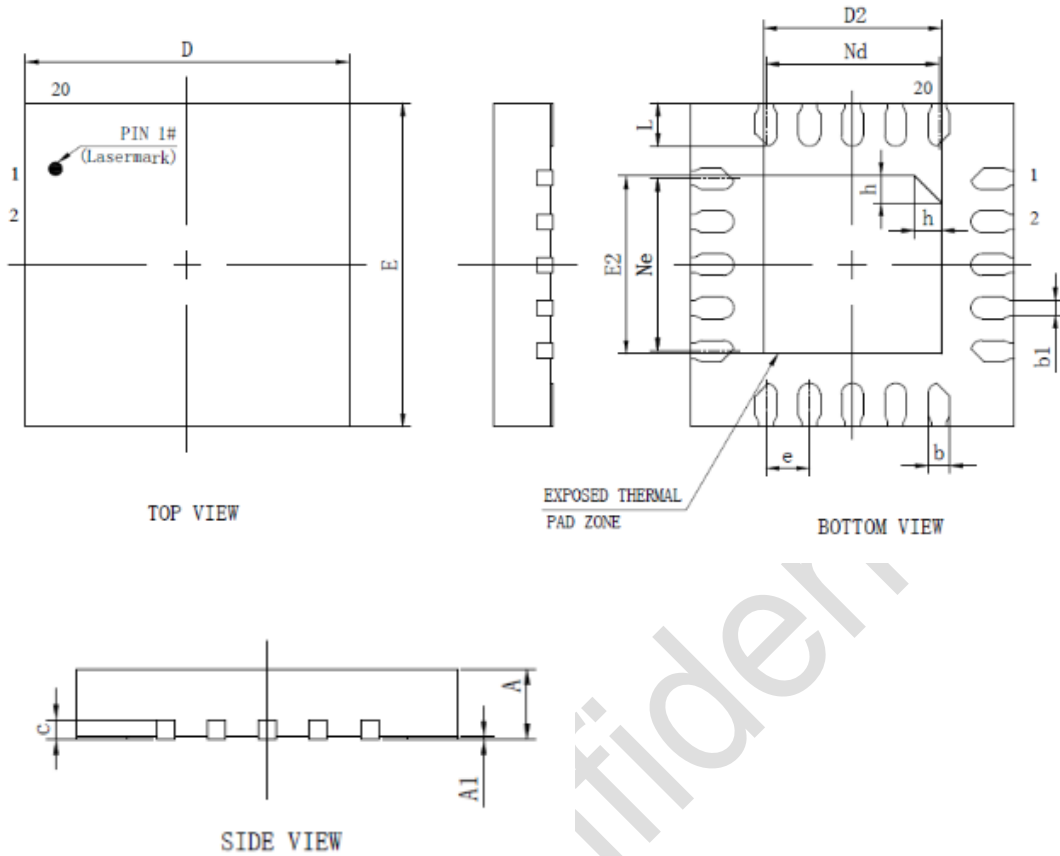
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	2.350	2.650	0.093	0.104
A1	0.100	0.300	0.004	0.012
A2	2.100	2.500	0.083	0.098
b	0.330	0.510	0.013	0.020
c	0.204	0.330	0.008	0.013
D	12.520	13.000	0.493	0.512
E	7.400	7.600	0.291	0.299
E1	10.210	10.610	0.402	0.418
e	1.270 (BSC)		0.050 (BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°

PDIP20 封装尺寸如下:



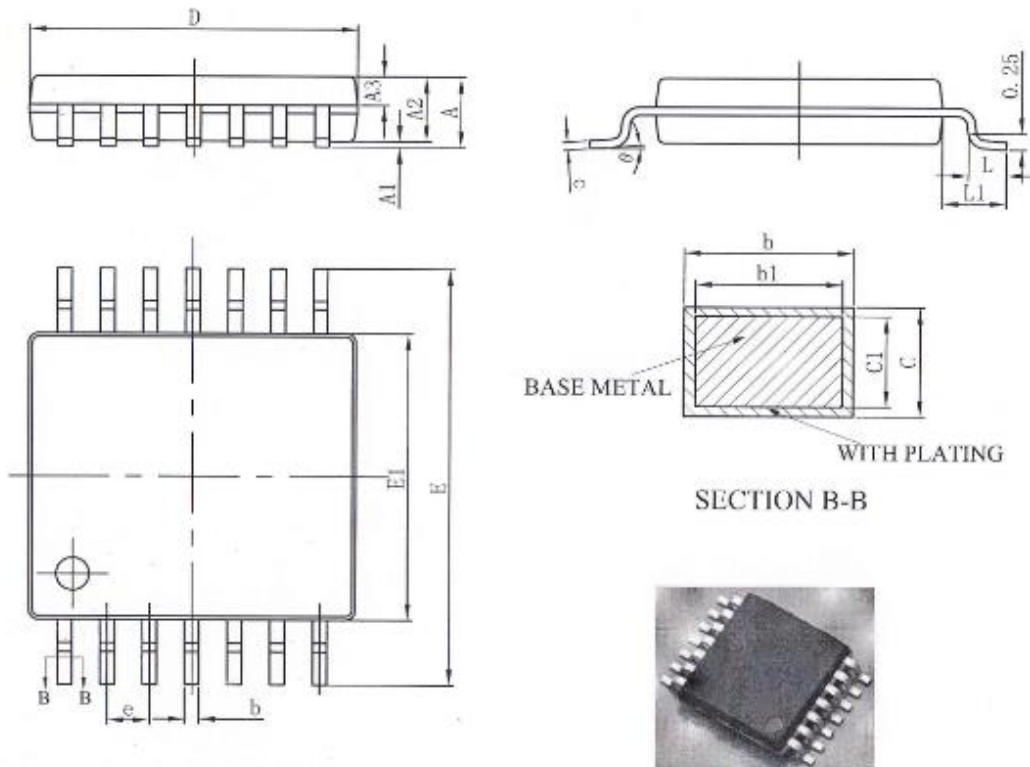
Symbol	Millimeters		
	Min	Nom	Max
A	3.60	3.80	4.00
A1	0.51	—	—
A2	3.20	3.30	3.40
A3	1.47	1.52	1.57
b	0.44	—	0.52
b1	0.43	0.46	0.49
B1	1.52REF		
c	0.25	—	0.29
c1	0.24	0.25	0.26
D	25.80	25.90	26.00
E1	6.45	6.55	6.65
e	2.54BSC		
eA	7.62REF		
eB	7.62	—	9.30
eC	0	—	0.84
L	3.00	—	—

QFN20 封装尺寸如下:



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	0.500	0.600	0.020	0.024
A1	-	0.050	-	0.002
b	0.150	0.250	0.006	0.010
b1	0.140 (REF)		0.006 (REF)	
c	0.100	0.200	0.004	0.008
D	2.900	3.100	0.114	0.122
D2	1.550	1.750	0.061	0.069
e	0.400 (BSC)		0.016 (BSC)	
Ne	1.600 (BSC)		0.063 (BSC)	
Nd	1.600 (BSC)		0.063 (BSC)	
E	2.900	3.100	0.114	0.122
E2	1.550	1.750	0.061	0.069
L	0.350	0.450	0.014	0.018
h	0.200	0.300	0.008	0.012

TSSOP14 封装尺寸如下:



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	1.200	-	0.047
A1	0.050	0.150	0.002	0.006
A2	0.900	1.050	0.035	0.041
A3	0.390	0.490	0.015	0.019
b	0.200	0.280	0.008	0.011
b1	0.190	0.250	0.007	0.010
c	0.130	0.170	0.005	0.007
c1	0.120	0.140	0.005	0.006
D	4.900	5.100	0.193	0.201
E1	4.300	4.500	0.169	0.177
E	6.200	6.600	0.244	0.260
e	0.650 (BSC)		0.026 (BSC)	
L	0.450	0.750	0.018	0.030
L1	1.000 (BSC)		0.039 (BSC)	
θ	0°	8°	0°	8°

## 附录 1，文档更改历史

日期	版本	内容
2016-11-24	1.01	初始版本
2016-11-29	1.02	更新表格的一些笔误
2016-12-08	1.03	添加 SOP16、SOP20 封装 删除表 21.2 的 IVDD，PPGCON2 复位值改为全 0
2017-5-27	1.04	更新各电路特性表格 更新第 9 章的半桥、全桥相关描述 工作电压范围改为 2.0V~5.5V 特性列表加入： 程序和数据 EEPROM 可经受 100 万次写操作 EEPROM 保存时间>40 年
2017-7-24	1.05	P1A 和 CCP1 管脚是分开的，更正第 9 章相关内容 PA1、PC2 和 USBB 口输出复用的笔误改正 添加 FT61F042 脚位图 添加注意事项第 3 点到 12.2.6“A/D 转换步骤”，明确 ADC 的采样时间要求 T <sub>ACQ</sub> 加入“芯片版本历史” 添加注意事项到 4.2 小节“外部复位 MCLR”
2017-8-5	1.06	改正笔误“FT60F03X” 在相关章节注明 OST 在 LP 晶体模式下的溢出周期为 32768 删除 21.14 节 D2I 电路电气参数 Reference Pin Voltage 的最大值和最小值，修改典型值为 1.5 修改 21.14 节 D2I 电路电气参数的 Ref 值为 75K
2017-8-24	1.07	修改了一个笔误，IRCF default 值为 100
2017-9-18	1.08	修改“芯片版本历史”的 C 版本描述
2017-10-20	1.09	“极限参数”改为“绝对极限参数”，并加入注意说明
2017-11-15	1.10	更新了内置低频振荡器的电气参数，随电源电压变化范围更新为典型值 8%，条件为 1.8~5.5V，T=27°C
2017-12-13	1.11	添加了 21.16 小节“AC 电气参数” 添加了 21.17 小节“时序图” 添加了 21.18 小节“直流和交流特性图表” 修改了 12.2.6 小节实例程序
2018-02-01	1.12	添加了 21.12 小节内置 ADC 参考电压的稳定时间参数 修改 2.2.6.1 小节中 16M 晶振改为高速晶振，最大 20MHz 更新 12.2.6 小节注意事项 更新芯片版本历史
2018-03-20	1.13	修改 2.2.6.1 小节中 FOSC[2:0]的相关描述，将 32K 晶振模式改为 LP 模式，并且标注 32KHz；高速晶振模式改为 XT 模式，范围为 4~20MHz；外部时钟模式改为 EC 模式

2018-03-24	1.14	增加了 PDIP20 封装管脚图
2018-04-11	1.15	增加了 21.4 小节 LVR 电压的最大值和最小值 增加了 21.5 小节 LVD 电压的最大值和最小值
2018-04-27	1.16	修改了 21.12 小节内置 ADC 参考电压的稳定时间参数
2018-09-27	1.17	增加了 11.3 小节“关于编程周期小节” 修改了脚位图名称 1.2 小节管脚描述表中输入信号类型 TTL 全部改为 ST
2018-11-22	1.18	更新 LVR 档位描述
2018-12-7	1.19	更正 FT61F045-TRB 和 FT61F045-RB 脚位
2018-12-12	1.20	添加 FT61F042A-TGB 脚位和封装信息
2019-01-08	1.21	修改 1.3 小节脚位图的说明 修改 4.7 小节中 4 到 5 个慢时钟周期改为 3 到 4 个慢时钟周期 4.10 小节中 2K 改为 4K 21 小节芯片的电气特性中 27°C 全部改为 25°C 21 小节芯片的电气特性中 typ 全部改为 25°C 21.3 小节振荡频率注明为 LFMOD=0 条件 更新了 23 小节 SOP20 封装图 将 STATUS、OPTION、INTCON、TRISA、TRISB 和 TRISC 寄存器英文描述改为中文描述 文档中所有 /TO 和 /PD 全部改为 /TF 和 /PF 删掉了页眉中的 Preliminary
2019-9-19	1.22	更正 PAPU 寄存器位描述
2019-12-03	1.23	修改高频振荡器的校准偏差为±1.5%

## 附录 2，从其他器件移植

FT61F04X 相对于 FT61F02X 有以下变化：

### 1. SFR 空间比较

PN	FT61F02X	FT61F04X
地址		
0x06	-	PORTB
0x09	PR0	IOTTM 模块删除
0x93	IOTTMRESL	DACOE
0x94	IOTTMRESH	DACCON0
0x96	-	TRISB
0x97	IOTTMCFG	D2ICON
0x101	-	CMCON2
0x106	-	WPUB
0x11F	-	CMCON4
0x180~0x1FF	-	新增的一个 SFR bank

### 2. Status 寄存器变化

PN	FT61F02X	FT61F04X
Status		
bit6:5	PAGE[1:0], Register Bank Select bit 00 = Bank0 (0x00~ 0x7F) 01 = Bank 1 (0x80~ 0xFF) <b>1x = Bank2 (0x100~ 0x17F)</b>	PAGE[1:0], Register Bank Select bit 00 = Bank0 (0x00~ 0x7F) 01 = Bank 1 (0x80~ 0xFF) <b>10 = Bank2 (0x100~ 0x17F)</b> <b>11 = Bank3 (0x180~ 0x1FF)</b>

### 3. 寄存器位变化

PN	FT61F02X	FT61F04X
T2CON		
bit7	-	NRPM PPG 的非重复触发模式 1 = 使能 PPG 非重复触发模式，Timer2 将配合 PPG 模块工作 0 = 关闭 PPG 非重复触发模式

PN	FT61F02X	FT61F04X
CMCON0		
bit7	C2OUT	<b>C5OUT</b>



bit6	C1OUT	<b>C4OUT</b>
bit5	C2INV	<b>C5INV</b>
bit4	C1INV	<b>C4INV</b>
bit3	CIS	<b>CIS</b>
bit2	CM[2:0]	<b>CM[2:0]</b>
bit1		
bit0		

	PN	FT61F02X	FT61F04X
CMCON1			
bit7		-	<b>HYSEN[3]</b>
bit6		-	<b>HYSEN[2]</b>
bit5		-	<b>HYSEN[1]</b>
bit4		-	<b>HYSEN[0]</b>
bit3		-	-
bit2		-	-
bit1		T1GSS	<b>T1GSS</b>
bit0		C2SYNC	<b>C5SYNC</b>

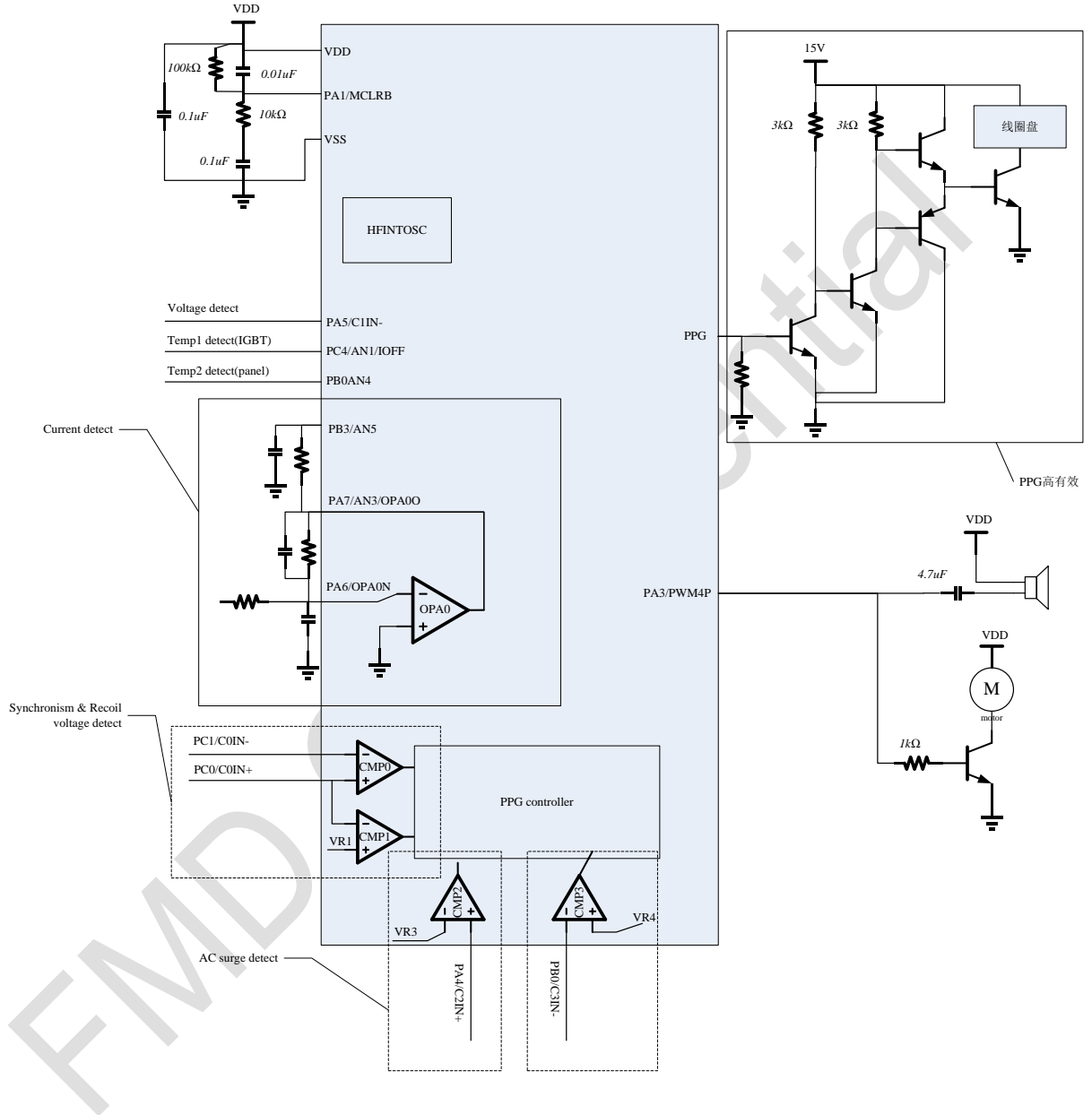
	PN	FT61F02X	FT61F04X
VCON2			
bit6		-	<b>DAC1ON</b>
bit5		-	<b>DAC2ON</b>

	PN	FT61F02X	FT61F04X
ADCON1			
bit0		-	<b>AN0SEL</b>

	PN	FT61F02X	FT61F04X
OSCCON			
bit[6:4], value	default	101	<b>100</b>

### 附录 3, 应用电路

#### 电磁炉



## Fremont Micro Devices (SZ) Limited

#5-8, 10/F, Changhong Building, Ke-Ji Nan 12 Road, Nanshan District, Shenzhen, Guangdong  
518057

Tel: (86 755) 86117811

Fax: (86 755) 86117810

## Fremont Micro Devices (Hong Kong) Limited

#16, 16/F, Blk B, Veristrong Industrial Centre, 34-36 Au Pui Wan Street, Fotan, Shatin, Hong  
Kong

Tel: (852) 27811186

Fax: (852) 27811144

Web Site: <http://www.fremontmicro.com/>

\* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices, Incorporated (BVI) assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices, Incorporated (BVI). Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices, Incorporated (BVI) products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices, Incorporated (BVI). The FMD logo is a registered trademark of Fremont Micro Devices, Incorporated (BVI). All other names are the property of their respective own.