

## 高性能 RISC CPU

- 只需学习 **35 条指令**
- 所有指令均为单周期（除了分支跳转）
- **存储架构**
- 程序 ROM: **3k x 14bits**
- 数据 RAM: **256 x 8bits**
- 数据 EEPROM: **128 x 8bits**
- **8 层硬件堆栈**
- 可选的指令周期: 2T/4T
- **125ns @ 2T, 16MHz, VDD≥2.7V**

## 特殊单片机特性

- **工作温度范围: -40℃~85℃**
- **宽工作电压范围: 1.9V~5.5V**
- **时钟源**
- 两个内部时钟
- **13.5M/16M 高速高精度 HIRC**
  - i. 支持软件微调, **每步 2.5%**
- **32k 低速低功耗 LIRC**
- 晶体振荡器和外部时钟输入
- 晶体时钟缺失检测
- 晶体时钟配置下的双速时钟启动
- 慢时钟周期测量
- **带 7 位预分频的 16 位看门狗, 时钟源可选**
- **上电复位延时计数器**
- **低功耗模式 SLEEP**
- 系统时钟可选择保持运行或关闭
- **低电压复位 LVR:**  
2.0V/2.2V/2.5V/2.8V/3.1/3.6V/4.1V
- **低电压检测 LVD:**
- 1.2V/2.0V/2.4V/2.7V/3.0V/3.3V/3.6V/4.0V
- 或外部输入电压
- **支持 ISP 和在线调试 OCD**
- 3 个硬件断点
- 软复位, 单步, 暂停, 跳跃等
- **程序 ROM 分区保护功能**
- **封装形式:**
- SOP8, MSOP10, SOP14, SOP16, SOP20, TSSOP20, DIP20, QFN20

## 外设特性

- **GPIO**
- 18 个方向独立控制的通用 IO: PORTA, PORTB, PORTC
- 8 个唤醒管脚: PORTA
- 18 个带开漏功能的管脚, 独立控制
- 18 个带上拉功能的管脚, 独立控制
- 18 个带下拉功能的管脚, 独立控制
- ADC 输入通道: AN0-7
- 8 个可编程源电流 PC0-1, PB2-7: 3/6/24mA@5V
- 8 个可编程灌电流 IO: max 55mA@5V
- **支持管脚第二功能的重映射**
- **1 个 12 位的 SAR ADC**
- 8 个外部通道 + 3 个内部参考电压通道
- 内部参考电压: VDD, 0.5V, 2V, 3V
- 外部参考: VREFP, VREFN
- **Timer0**
- 带 8 位预分频的 8 位定时器, 时钟源可选
- **Timer1**
- 12 位定时器, 时钟源可选
- **Timer2**
- 带 4 位预分频和 4 位后分频的 16 位定时器
- 内部慢时钟测量
- 4 路独立极性、独立占空比的 PWM
- 1 对带死区控制的互补 PWM 输出, 最多可映射到 6 个 IO 上
- 占空比和周期寄存器双缓冲
- 时钟源: HIRC, 晶体时钟, HIRC 的 2 倍频, 晶体时钟的 2 倍频, 以及指令时钟, 系统时钟, LIRC
- 睡眠模式下可保持工作
- 刹车输入
- **Buzzer 模式**
- **单脉冲模式**

选型表

型号	PROM(字)	DROM(byte)	SRAM(byte)	I/Os	Timers	Touch keys	封装
FT62F131-RB	3k	128	256	6	3	2	SOP8
FT62F13F-MRB				8		4	MSOP10
FT62F132-RB				12		5	SOP14
FT62F133-RB				14		7	SOP16
FT62F135-RB				18		8	SOP20
FT62F135-TRB				18		8	TSSOP20
FT62F135-DRB				18		8	DIP20
FT62F135-NRT				18		8	QFN20

## 目录

高性能 RISC CPU .....	- 1 -
特殊单片机特性 .....	- 1 -
外设特性 .....	- 1 -
选型表 .....	- 2 -
1. 系统功能框图及脚位 .....	13
1.1. 脚位图 .....	14
1.2. 管脚描述 .....	16
2. 特殊功能寄存器 .....	18
2.1. SFR, BANK0 .....	18
2.2. SFR, BANK1 .....	19
2.3. SFR, BANK2 .....	20
2.4. SFR, BANK3 .....	21
2.4.1. STATUS 寄存器, 地址 0x03, 0x83 .....	22
2.5. PCL 和 PCLATH .....	23
2.5.1. 修改 PCL .....	23
2.6. INDF 和 FSR 寄存器 .....	24
2.7. 关于寄存器保留位 .....	24
3. 系统时钟源 .....	25
3.1. 时钟源模式 .....	25
3.2. 外部时钟模式 .....	26
3.2.1. 振荡器起振定时器 (OST) .....	26
3.2.2. EC 模式 .....	26
3.2.3. LP 和 XT 模式 .....	26
3.3. 内部时钟模式 .....	26
3.3.1. 频率选择位 (IRCF) .....	27
3.3.2. HIRC 和 LIRC 时钟切换时序 .....	27
3.3.3. 频率与最小工作电压的关系 .....	28
3.3.4. HIRC 频率微调 .....	29

3.4.	时钟切换 .....	29
3.4.1.	系统时钟选择 (SCS) 位 .....	29
3.4.2.	振荡器起振超时状态 (OSTS) 位 .....	29
3.5.	双速时钟启动模式 .....	30
3.5.1.	双速启动模式配置 .....	30
3.5.2.	双速启动顺序 .....	30
3.6.	故障保护时钟监控器 .....	31
3.6.1.	故障保护检测 .....	31
3.6.2.	故障保护操作 .....	31
3.6.3.	故障保护条件清除 .....	32
3.6.4.	复位或从休眠中唤醒 .....	32
3.7.	与时钟源相关寄存器汇总 .....	33
3.7.1.	OSCCON 寄存器, 地址 0x8F .....	33
3.7.2.	FOSCCAL 寄存器, 地址 0x0D .....	34
3.7.3.	MSCON1 寄存器, 地址 0x18E .....	34
4.	复位源 .....	35
4.1.	POR 上电复位 .....	36
4.2.	外部复位 MCLR .....	36
4.3.	PWRT (上电计时器) .....	36
4.4.	BOR 低电压复位 .....	37
4.5.	LVD 低电压侦测 .....	37
4.5.1.	检测外部电压 .....	37
4.6.	错误指令复位 .....	37
4.7.	超时动作 .....	37
4.7.1.	PCON 寄存器 .....	38
4.8.	上电配置过程 .....	38
4.9.	上电校验过程 .....	38
4.10.	PCON 寄存器, 地址 0x8E .....	41
4.11.	LVDCON 寄存器, 地址 0x110 .....	42
4.12.	LVDTRIM 寄存器, 地址 0x19F .....	42

4.13.	配置寄存器汇总 .....	43
4.13.1.	UCFG0, PROM 地址 0x2000 .....	43
4.13.2.	UCFG1, PROM 地址 0x2001 .....	44
4.13.3.	UCFG2, PROM 地址 0x2002 .....	45
4.13.4.	UCFG3, PROM 地址 0x2003 .....	46
4.13.5.	MAINCSUM (地址: 0x2007) .....	46
5.	看门狗定时器 .....	47
5.1.	看门狗 .....	47
5.2.	看门狗时钟源 .....	48
5.3.	与看门狗相关寄存器汇总 .....	48
5.3.1.	WDTCON 寄存器, 地址 0x18 .....	49
6.	中断 .....	50
6.1.	INT 外部中断 .....	51
6.2.	端口变化中断 .....	51
6.2.1.	PAIF 标志位的清除 .....	51
6.3.	中断响应 .....	52
6.4.	中断过程中的现场保存 .....	52
6.5.	关于中断标志位 .....	52
6.6.	与中断相关寄存器汇总 .....	53
6.6.1.	INTCON 寄存器, 地址 0x0B/0x8B .....	53
6.6.2.	PIR1 寄存器, 地址 0x0C .....	54
6.6.3.	PIE1 寄存器, 地址 0x8C .....	55
6.6.4.	IOCA 寄存器, 地址 0x96 .....	55
7.	睡眠模式 .....	56
7.1.	唤醒模式 .....	56
7.2.	看门狗唤醒 .....	57
7.3.	中断唤醒 .....	57
7.4.	关于 SLEEP 后的第一条指令 .....	57
8.	数据 EEPROM .....	58
8.1.	编程数据 EEPROM 步骤 .....	58

8.2.	读数据 EEPROM.....	60
8.3.	关于编程周期.....	60
8.4.	EEPROM 的单编程模式.....	60
8.5.	与数据 EEPROM 相关寄存器汇总.....	61
8.5.1.	EEDAT 寄存器, 地址 0x9A.....	61
8.5.2.	EEADR 寄存器, 地址 0x9B.....	61
8.5.3.	EECON1 寄存器, 地址 0x9C.....	62
8.5.4.	EECON2 寄存器, 地址 0x9D.....	62
9.	12bit ADC 模块.....	63
9.1	ADC 的配置.....	63
9.1.1	端口配置.....	64
9.1.2	通道选择.....	64
9.1.3	触发方式选择.....	64
9.1.4	触发源选择.....	64
9.1.5	触发类型选择.....	64
9.1.6	触发延时配置.....	64
9.1.7	ADC 参考电压.....	65
9.1.8	转换时钟.....	65
9.1.9	中断.....	66
9.1.10	转换结果的格式.....	67
9.1.11	阈值比较.....	67
9.2	ADC 的工作原理.....	68
9.2.1	启动转换.....	68
9.2.2	转换完成.....	68
9.2.3	终止转换.....	68
9.2.4	休眠模式下 ADC 的工作.....	68
9.2.5	A/D 转换步骤.....	69
9.2.6	A/D 采集时间要求.....	70
9.3	与 ADC 相关寄存器汇总.....	71
9.3.1	ADRESL, 地址 0x111.....	71

9.3.2	ADRESH, 地址 0x112.....	71
9.3.3	ADCON0, 地址 0x113.....	72
9.3.4	ADCON1, 地址 0x114.....	73
9.3.5	ADCON2, 地址 0x115.....	74
9.3.6	LEBCON 寄存器, 地址 0x185.....	75
9.3.7	ADCON3, 地址 0x186.....	75
9.3.8	ADCMPH, 地址 0x187.....	76
9.3.9	ADDLY/LEBPRL, 地址 0x188 .....	76
9.3.10	VRP5VCAL, 地址 0x97 .....	76
9.3.11	VR2VCAL, 地址 0x108 .....	77
9.3.12	VR3VCAL, 地址 0x11F .....	77
10.	定时器 0.....	78
10.1.	Timer0 定时器模式.....	78
10.1.1.	Timer0 的时钟源.....	79
10.1.2.	TMR0 寄存器的读写.....	79
10.2.	Timer0 计数器模式.....	79
10.2.1.	软件可配置预分频电路 .....	79
10.2.2.	定时器 0 中断 .....	80
10.2.3.	用外部时钟驱动定时器 0 .....	80
10.2.4.	睡眠模式下的状态 .....	80
10.3.	Timer0 的 PWM 控制信号模式.....	80
10.4.	与 Timer0 相关寄存器汇总.....	81
10.4.1.	OPTION 寄存器, 地址 0x81.....	81
10.4.2.	TMR0, 地址 0x01 .....	82
10.4.3.	T0CON0, 地址 0x1F.....	82
10.4.4.	T0DUTY, 地址 0x18C .....	83
10.4.5.	PR0, 地址 0x18D .....	83
11.	定时器 1.....	84
11.1.	Timer1 工作原理.....	84
11.2.	Timer1 计数值的读写.....	85

11.3.	与 Timer1 相关寄存器汇总.....	85
11.3.1.	PR1L 寄存器, 地址 0x116,0x117.....	85
11.3.2.	TMR1 寄存器, 地址 0x118, 0x119.....	86
11.3.3.	T1CON0 寄存器, 地址 0x11A.....	86
12.	定时器 2.....	87
12.1.	Timer2 工作原理.....	88
12.2.	关于 PR2 的更新.....	88
12.3.	Timer2 计数值的读写.....	89
12.4.	Timer2 预分频清零.....	90
12.5.	Timer2 时钟源.....	90
12.6.	睡眠下的工作.....	90
12.7.	与 Timer2 相关寄存器汇总.....	91
12.7.1.	PR2 寄存器, 地址 0x91, 0x92.....	91
12.7.2.	TMR2 寄存器, 地址 0x11, 0x13.....	91
12.7.3.	T2CON0 寄存器, 地址 0x12.....	92
12.7.4.	T2CON1 寄存器, 地址 0x9E.....	93
13.	PWM 模块.....	94
13.1.	周期.....	94
13.2.	占空比.....	95
13.3.	时钟源选择.....	95
13.4.	睡眠下 PWM 状态.....	95
13.5.	P1A 的死区时间.....	96
13.6.	故障刹车.....	96
13.6.1.	刹车状态.....	97
13.6.2.	故障清除.....	97
13.6.3.	自动重启.....	97
13.7.	关于周期和占空比寄存器的更新.....	98
13.8.	蜂鸣器模式 (Buzzer).....	99
13.9.	单脉冲输出.....	99
13.10.	PWM 输出重映射.....	99



13.11.	P1C、P1D 的第 2 功能输出 .....	100
13.12.	与 PWM1 相关寄存器汇总 .....	101
13.12.1.	P1ADTL 寄存器, 地址 0x0E .....	101
13.12.2.	P1BDTL 寄存器, 地址 0x0F .....	102
13.12.3.	P1CDTL 寄存器, 地址 0x10 .....	102
13.12.4.	P1DDTL 寄存器, 地址 0x8 .....	102
13.12.5.	TMR2L 寄存器, 地址 0x11 .....	102
13.12.6.	TMR2H 寄存器, 地址 0x13 .....	103
13.12.7.	T2CON0 寄存器, 地址 0x12 .....	103
13.12.8.	P1ADTH 寄存器, 地址 0x14 .....	103
13.12.9.	P1BDTH 寄存器, 地址 0x15 .....	103
13.12.10.	P1CDTH 寄存器, 地址 0x1A .....	104
13.12.11.	P1DDTH 寄存器, 地址 0x9 .....	104
13.12.12.	P1CON 寄存器, 地址 0x16 .....	104
13.12.13.	P1BR0 寄存器, 地址 0x17 .....	105
13.12.14.	P1BR1 寄存器, 地址 0x19 .....	106
13.12.15.	P1OE2 寄存器, 地址 0x11B .....	107
13.12.16.	P1OE 寄存器, 地址 0x90 .....	108
13.12.17.	PR2L 寄存器, 地址 0x91 .....	108
13.12.18.	PR2H 寄存器, 地址 0x92 .....	109
13.12.19.	P1POL 寄存器, 地址 0x99 .....	109
13.12.20.	P1POL2 寄存器, 地址 0x109 .....	110
13.12.21.	P1AUX 寄存器, 地址 0x1E .....	111
14.	I/O 端口 .....	112
14.1.	PORTx 端口和 TRISx 寄存器 .....	112
14.2.	端口的其他功能 .....	113
14.2.1.	弱上拉 .....	113
14.2.2.	弱下拉 .....	113
14.2.3.	ANSEL 寄存器 .....	113
14.3.	源电流选择 .....	113

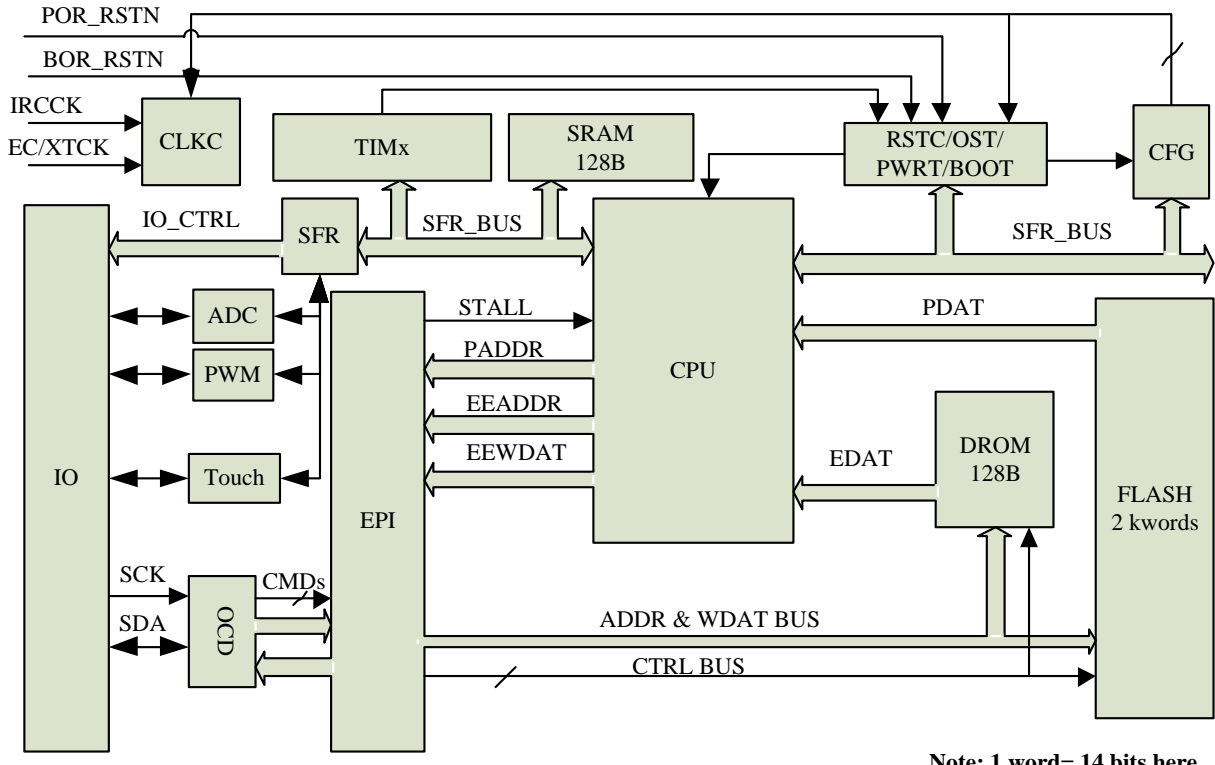
14.4.	灌电流选择 .....	113
14.5.	开漏功能 .....	114
14.6.	与 GPIO 相关寄存器汇总 .....	114
14.6.1.	WPUA, 地址 0x95 .....	114
14.6.2.	WPUB, 地址 0x10D .....	115
14.6.3.	WPUC, 地址 0x93 .....	115
14.6.4.	TRISA, 地址 0x85 .....	115
14.6.5.	TRISB, 地址 0x86 .....	116
14.6.6.	TRISC, 地址 0x87 .....	116
14.6.7.	PORTA, 地址 0x05 .....	116
14.6.8.	PORTB, 地址 0x06 .....	117
14.6.9.	PORTC, 地址 0x7 .....	117
14.6.10.	WPDA, 地址 0x89 .....	117
14.6.11.	WPDB, 地址 0x10E .....	118
14.6.12.	WPDC, 地址 0x8D .....	118
14.6.13.	PSRCB1, 地址 0x88 .....	118
14.6.14.	PSRCB2, 地址 0x10C .....	119
14.6.15.	PSRCC, 地址 0x94 .....	119
14.6.16.	PSINKB, 地址 0x10F .....	120
14.6.17.	PSINKC, 地址 0x9F .....	120
14.6.18.	ODCONA, 地址 0x105 .....	120
14.6.19.	ODCONB, 地址 0x106 .....	121
14.6.20.	ODCONC, 地址 0x107 .....	121
14.6.21.	ANSEL0, 地址 0x11E .....	121
15.	慢时钟测量 .....	122
15.1.	测量原理 .....	122
15.2.	操作步骤 .....	123
15.3.	与慢时钟测量相关寄存器汇总 .....	123
15.3.1.	MSCON0 寄存器, 地址 0x1B .....	124
15.3.2.	SOSCPR 寄存器, 地址 0x1C, 1D .....	125

16.	TOUCH 模块.....	126
16.1.	触摸按键功能.....	126
16.2.	触摸按键结构.....	126
16.3.	触摸按键检测.....	126
16.4.	硬件检测模式.....	129
16.4.1.	与触摸按键相关寄存器汇总.....	130
16.4.2.	TKCON0 寄存器, 地址 0x11C.....	131
16.4.3.	TKCON1 寄存器, 地址 0x11D.....	131
17.	指令集汇总.....	132
18.	芯片的电气特性.....	134
18.1.	极限参数.....	134
18.2.	内置高频振荡器 (HIRC).....	134
18.3.	内置低频振荡器 (LIRC).....	134
18.4.	低电压复位电路 (LVR).....	135
18.5.	低电压侦测电路 (LVD).....	135
18.6.	上电复位电路 (POR).....	135
18.7.	I/O PAD 电路.....	136
18.8.	总体工作电流 ( $I_{DD}$ ).....	137
18.9.	AC 电气参数.....	137
18.10.	12bit ADC 特性.....	138
18.11.	直流和交流特性曲线图.....	139
18.11.1.	HIRC vs $V_{DD}$ ( $T_A=25^\circ\text{C}$ ).....	139
18.11.2.	LIRC vs $V_{DD}$ ( $T_A=25^\circ\text{C}$ ).....	140
18.11.3.	不同 $V_{DD}$ 下, $I_{DD}$ vs Freq (2T, $T_A=25^\circ\text{C}$ ).....	140
18.11.4.	不同 $V_{DD}$ 下, $I_{SB}$ (睡眠电流)随温度变化曲线.....	141
18.11.5.	不同温度下, $I_{OH}$ (L0 -3mA) vs $V_{OH}$ @ $V_{DD}=5V$ .....	141
18.11.6.	不同温度下, $I_{OH}$ (L1 -6mA) vs $V_{OH}$ @ $V_{DD}=5V$ .....	142
18.11.7.	不同温度下, $I_{OH}$ (L2 -18mA) vs $V_{OH}$ @ $V_{DD}=5V$ .....	142
18.11.8.	不同温度下, $I_{OH}$ (L3 -24mA) vs $V_{OH}$ @ $V_{DD}=5V$ .....	143
18.11.9.	不同温度下, $I_{OL}$ (L0 35mA) vs $V_{OL}$ @ $V_{DD}=5V$ .....	143

---

18.11.10.	不同温度下, $I_{OL}$ ( L1 53mA ) vs $V_{OL}$ @ $V_{DD}=5V$ .....	144
18.11.11.	不同温度下, $I_{OL}$ ( L2 55mA ) vs $V_{OL}$ @ $V_{DD}=5V$ .....	144
19.	芯片封装信息 .....	145
附录 1,	文档更改历史 .....	153

### 1. 系统功能框图及脚位



Note: 1 word= 14 bits here.

图 1.1 芯片整体功能框图

## 1.1. 脚位图

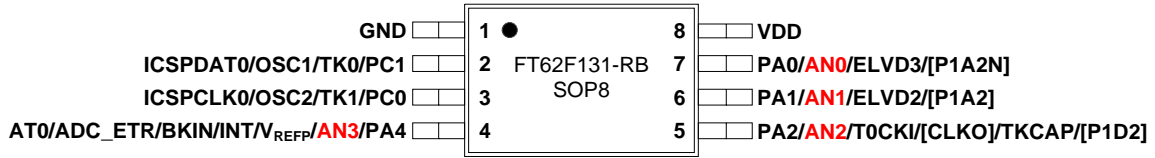


图 1.2 SOP8 脚位

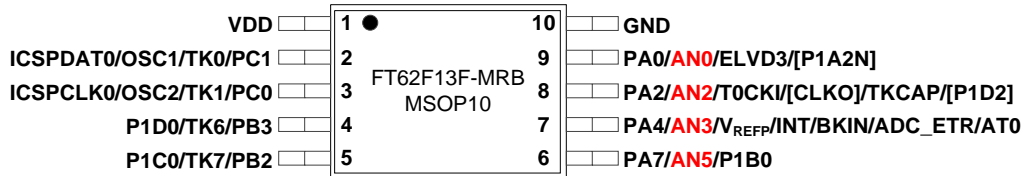


图 1.3 MSOP10 脚位

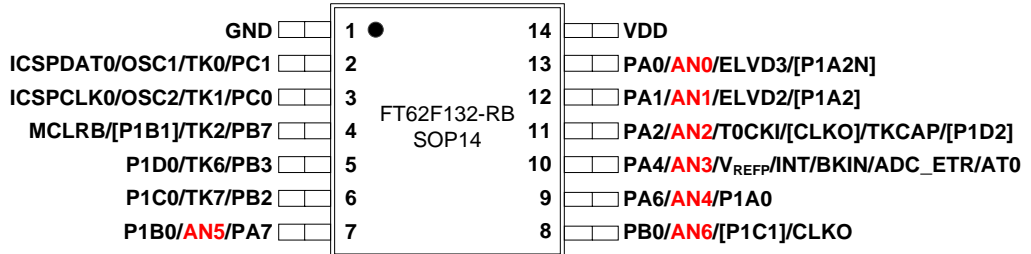


图 1.4 SOP14 脚位

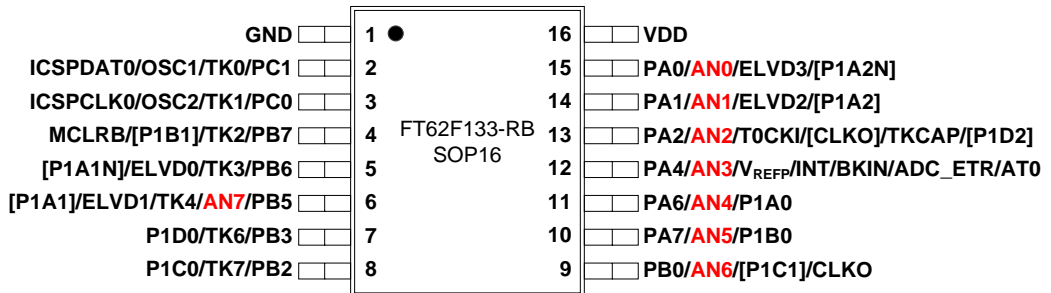


图 1.5 SOP16 脚位

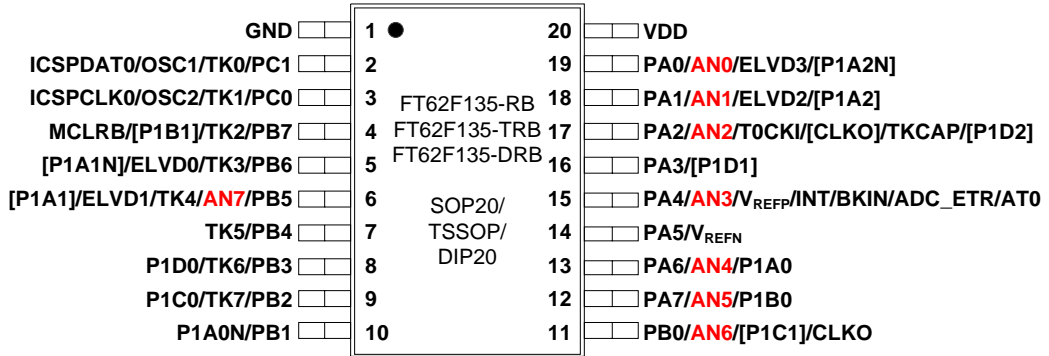


图 1.6 SOP20/TSSOP20/DIP20 脚位

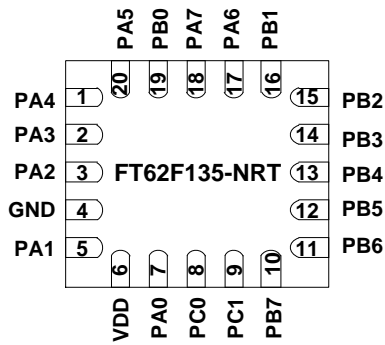


图 1.7 QFN20 脚位

注意：FT62F135-NRT（QFN20）底部的焊垫跟 GND 是连一起的。

## 1.2. 管脚描述

SOP20	Pin name	Type	INT input	Main func.	Pullup/down
1	GND	Ground	—	Ground	
2	PC1/TK0/OSC1/ICSPDAT	IO	—	PC1	√
	TK0, Touch 按键 0				
	OSC1, 晶体管脚 1				
3	ISPDAT, ISP 数据 IO				
	PC0/TK1/OSC2/ICSPCLK	IO	—	PC0	√
	TK1, Touch 按键 1				
4	OSC2, 晶体管脚 2				
	ISPK, ISP 时钟输入				
	PB7/TK2/[P1B1]/MCLR	IO	—	PB7	√
5	TK2, Touch 按键 2				
	[P1B1], P1B 的映射输出				
	MCLR, 外部复位输入				PU
6	PB6/TK3 /ELVD0/[P1A1N]	IO	—	PB6	√
	TK3, Touch 按键 3				
	ELVD0, 外部 LVD 检测输入 0				
7	[P1A1N], P1A0N 的映射输出				
	PB5/ AN7/TK4/ELVD0/[P1A1]	IO	—	PB5	√
	AN7, 模拟输入通道 7				
8	TK4, Touch 按键 4				
	ELVD1, 外部 LVD 检测输入 1				
	[P1A1], P1A0 的映射输出				
9	PB4/TK5/ ATEST0	IO	—	PB4	√
	TK5, Touch 按键 5				
	ATEST0, 内部测试管脚 0				
10	PB3 /TK6/P1D0	IO	—	PB3	√
	TK6, Touch 按键 6				
	P1D0, PWMD 输出				
11	PB2 /TK7/P1C0	IO	—	PB2	√
	TK7, Touch 按键 7				
	P1C0, PWMC 输出				
12	PB1/P1A0N	IO	—	PB1	√
	P1A0N, P1A0N 的 PWM 输出				
	PB0/AN6/[P1C1]/CLKO	IO	—	PB0	√
13	AN6, 模拟输入通道 6				
	[P1C1], P1C 的映射输出				
	CLKO, 指令时钟输出				



SOP20	Pin name	Type	INT input	Main func.	Pullup/down
12	PA7/AN5/P1B0/[ICSPCLK1]	IO	√	PA7	√
	AN5, 模拟输入通道 5				
	P1B0, PWMB 的 PWM 输出				
	[ICSPCLK1], ISP 数据映射 IO				
13	PA6/AN4/P1A0/[ICSPCLK1]	IO	√	PA6	√
	AN4, 模拟输入通道 4				
	P1A0, P1A0 的 PWM 输出				
	[ICSPCLK1], ISP 时钟映射输入				
14	PA5/VREFN	IO	√	PA5	√
	VREFN, ADC 外部负参考输入				
15	PA4/AN3/VREFP /INT/BKIN/ADC_ETR/AT0	IO	√	PA4	√
	AN4, 模拟输入通道 4				
	VREFP, ADC 外部正参考输入				
	INT, 外部中断输入				√
	BKIN,PWM刹车输入				
	ADC_ETR,ADC外部触发信号输入				
	AT0, TEST 测试管脚				
16	PA3/[P1D1]	IO	√	PA3	√
	[P1D1], P1D 的映射输出				
17	PA2/AN2/T0CKI/[CLKO]/TKCAP/[P1D2]	IO	√	PA2	√
	AN2, 模拟输入通道 2				
	T0CKI, Timer0 外部时钟				
	[CLKO],系统时钟映射输出管脚				
	TKCAP, Touch 模块 CAP 管脚				
	[P1D2], P1D 的映射输出				
18	PA1/AN1/ELVD2/[P1A2]	IO	√	PA1	√
	AN1, 模拟输入通道 1				
	ELVD2, 外部 LVD 检测输入 2				
	[P1A2], P1A0 的映射输出				
19	PA0/AN0/ELVD3/[P1A2N]	IO	√	PA0	√
	AN0, 模拟输入通道 0				
	ELVD3, 外部 LVD 检测输入 3				
	[P1A2N], P1A0N 的映射输出				
20	VDD	Power	—		Power

## 2. 特殊功能寄存器

### 2.1. SFR, BANK0

ADDR	Name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	POR reset	
BANK0											
0	INDF	使用 FSR 的内容对数据存储器进行访问（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	FSRB8	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	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	xxxx xxxx	
7	PORTC	—	—	—	—	—	—	PC1	PC0	---- --xx	
8	P1DDTL	P1D 占空比寄存器低 8 位								0000 0000	
9	P1DDTH	P1D 占空比寄存器高 8 位								0000 0000	
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	CKMIF	LVDIF	ACMPIF	TMR1IF	OSFIF	TMR2IF	ADCIF	0000 0000	
D	FOSCCAL	FOSCCAL[7:0]								0110 1000	
E	P1ADTL	P1A 占空比寄存器低 8 位								0000 0000	
F	P1BDTL	P1B 占空比寄存器低 8 位								0000 0000	
10	P1CDTL	P1C 占空比寄存器低 8 位								0000 0000	
11	TMR2L	Timer2 [7:0]								0000 0000	
12	T2CON0	PR2U	TOUTPS[3:0]			TMR2ON	T2CKPS[1:0]			0000 0000	
13	TMR2H	Timer2 [15:8]								0000 0000	
14	P1ADTH	P1A 占空比寄存器高 8 位								0000 0000	
15	P1BDTH	P1B 占空比寄存器高 8 位								0000 0000	
16	P1CON	P1AUE	P1DC[6:0]								0000 0000
17	P1BR0	P1BEVT	P1BKS[2:0]			P1BSS[1:0]		P1ASS[1:0]		0000 0000	
18	WDTCON	—	WCKSRC[1:0]		WDTPS[3:0]			SWDTEN		-000 1000	
19	P1BR1	P1D2SS[1:0]		P1DSS[1:0]		P1C2SS[1:0]		P1CSS[1:0]		0000 0000	
1A	P1CDTH	P1C 占空比寄存器高 8 位								0000 0000	
1B	MSCON0	—	—	ROMLPE	CLKOS	SLVREN	CKMAVG	CKCNT1	T2CKRUN	0001 0000	
1C	SOSCPRL	SOSCPRL [7:0]								1111 1111	
1D	SOSCPRH	—	—	—	—	SOSCPRH [11:8]					---- 1111
1E	P1AUX	—	—	P1B2SS[1:0]		P1CF2E	P1CF2	P1DF2E	P1DF2	--00 0000	
1F	T0CON0	—	—	—	—	T0ON	T0CKRUN	T0CKSRC[1:0]		---- 1000	
20-3F	SRAM BANK0, (32Bytes) Physical address 0x20-0x3F									xxxx xxxx	
40-7F	SRAM BANK0, (64Bytes) Physical address 0x40-0x7F									xxxx xxxx	

## 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	FSRB8	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	PORTA 方向控制								1111 1111
86	TRISB	PORTB 方向控制								1111 1111
87	TRISC	—	—	—	—	—	—	TRISC[1:0]		---- --11
88	PSRCB1	PORTB[5:2]源电源设置								1111 1111
89	WPDA	PORTA 弱下拉控制								0000 0000
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	CKMIE	LVDIE	ACMPIE	TMR1IE	OSFIE	TMR2IE	ADCIE	0000 0000
8D	WPDC	—	—	—	—	—	—	PORTC 弱下拉控制		---- --00
8E	PCON	LVDL[3:0]				LVDEN	LVDW	/POR	/BOR	0000 0xqq
8F	OSCCON	LFMOD	IRCF[2:0]			OSTS	HTS	LTS	SCS	0101 x000
90	P1OE	P1C0OE	P1B0OE	P1A2NOE	P1A2OE	P1A1NOE	P1A1OE	P1A0NOE	P1A0OE	0000 0000
91	PR2L	PR2[7:0] Timer2 period register, low byte								1111 1111
92	PR2H	PR2[15:8], Timer2 period register, high byte								1111 1111
93	WPUC	—	—	—	—	—	—	PORTC 弱上拉控制		---- --00
94	PSRCC	—	—	—	—	PORTC[1:0]源电流设置				---- 1111
95	WPUA	PORTA 弱上拉控制								1111 1111
96	IOCA	IOCA[7:0]								0000 0000
97	VRP5VCAL	VRP5VCAL[7:0]								xxxx xxxx
98	—	保留								0000 0000
99	P1POL	P1C0P	P1B0P	P1A2NP	P1A2P	P1A1NP	P1A1P	P1A0NP	P1A0P	0000 0000
9A	EEDAT	EEDAT[7:0]								0000 0000
9B	EEADR	EEADR[7:0]								0000 0000
9C	EECON1	—	Reserved	WREN3	WREN2	WRERR	WREN1	PONLY	RD	-000 x000
9D	EECON2	—	—	—	—	—	—	—	WR	---- --0
9E	T2CON1	—	—	—	P1OS	P1BZM	T2CKSRC[2:0]		---0 0000	
9F	PSINKC	—	—	—	—	—	—	PSINKC[1:0]		---- --00
A0–BF	SRAM BANK1 (32Bytes), Physical address 0x00–0x1F									xxxx xxxx
C0–EF	SRAM BANK1 (48Bytes), Physical address 0x80–0xAF									xxxx xxxx
F0–FF	SRAM, 访问 BANK0 的 0x70–0x7F									xxxx xxxx

## 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	RXRSM	—	—	RSVH1	RSVH1X	RSAST	RSBST	REB	RSDATA	--00 0010	
102	PCL	程序计数器低 8 位								0000 0000	
103	STATUS	FSRB8	PAGE[1:0]		/TF	/PF	Z	HC	C	0001 1xxx	
104	FSR	间接寻址指针寄存器								xxxx xxxx	
105	ODCONA	ODCONA[7:0]								0000 0000	
106	ODCONB	ODCONB[7:0]								0000 0000	
107	ODCONC	—	—	—	—	—	—	ODCONC[1:0]		---- --00	
108	VR2VCAL	VR2VCAL[7:0]								xxxx xxxx	
109	P1POL2	P1D2P	P1D1P	P1D0P	—	—	P1C1P	P1B1P	—	000- -00-	
10A	PCLATH	—	—	—	程序计数器高 5 位锁存器					---0 0000	
10B	INTCON	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000	
10C	PSRCB2	—	—	—	—	PORTB[7:6] 源电流设置				---- 1111	
10D	WPUB	WPUB[7:0]								0000 0000	
10E	WPDB	PORTB 下拉控制								0000 0000	
10F	PSINKB	PSINKB[7:2]						—		0000 00--	
110	LVDCON	—			LVDP	LVDDEB	LVDM[2:0]			---0 1100	
111	ADRESL	左对齐格式下 AD 结果的低 4 位或者右对齐下结果的低 8 位								xxxx xxxx	
112	ADRESH	左对齐格式下 AD 结果的高 8 位或者右对齐格式下的高 4 位								xxxx xxxx	
113	ADCON0	—	CHS[3:0]				ADEX	GO/DONE	ADON	-000 0000	
114	ADCON1	ADFM	ADCS[2:0]			ADNREF[1:0]		ADPREF[1:0]		0000 0000	
115	ADCON2	ADINTREF[1:0]		ETGTYP[1:0]		ADDLY.8	ETGSEL[2:0]			0000 0000	
116	PR1L	PR1[7:0] Timer1 period register, low byte								1111 1111	
117	PR1H	—				PR1[11:8] Timer1 period register, high byte				---- 1111	
118	TMR1L	Timer1[7:0]								0000 0000	
119	TMR1H	—				Timer1[11:8]				---- 0000	
11A	T1CON0	—			T1CKPSA	T1CKRUN	T1ON	T1CKSRC[1:0]			---0 0000
11B	P1OE2	P1D2OE	P1D1OE	P1D0OE	—	—	P1C1OE	P1B1OE	—	000- -00-	
11C	TKCON0	TKRSEL[1:0]		VREFM	TKCAPM	TKMODE	TKCHGS[1:0]		TKEN	0000 0000	
11D	TKCON1	TKCHE[7:0]								0000 0000	
11E	ANSEL0	ANSEL0[7:0]								0000 0000	
11F	VR3VCAL	VR3VCAL[7:0]								xxxx xxxx	
120-16F	SRAM BANK2 (80Bytes), Physical address 0xB0-0xFF									xxxx xxxx	
170-17F	SRAM, 访问 BANK0 的 0x70-0x7F									xxxx xxxx	

## 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	SECCODE	只读，存放 13.5M HIRC 的校准值								xxxx xxxx
182	PCL	程序计数器低 8 位								0000 0000
183	STATUS	FSRB8	PAGE[1:0]		/TF	/PF	Z	HC	C	0001 1xxx
184	FSR	间接寻址指针寄存器								xxxx xxxx
185	LEBCON	LEBEN	LEBCH[1:0]		—	EDGS	—	—	—	000- 0---
186	ADCON3	ADFBEN	ADCMPOP	ADCMPEN	—	LEBADT	—	—	—	000- 0---
187	ADCMPH	ADCMPH[7:0]								0000 0000
188	ADDLY	ADDLY[7:0] / LEBPRL[7:0]								0000 0000
18A	PCLATH	—	—	—	程序计数器高 5 位锁存器					---0 0000
18B	INTCON	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000
18C	T0DUTY	Timer0 产生的 PWM 波占空比设置								0000 0000
18D	PR0	Timer0 产生的 PWM 波周期设置								1111 1111
18E	MSCON1	—							HIRCM	---- ---0
19F	LVDTRIM	—	LVDADJ[3:0]			LVRADJ[2:0]				-100 0011
1A0-1EF	—								----	
1F0-1FF	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;

### 2.4.1. STATUS 寄存器，地址 0x03, 0x83

Bit	7	6	5	4	3	2	1	0
Name	FSRB8	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	FSRB8	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 INDF 和 FSR 寄存器一节
6:5	PAGE	<b>PAGE:</b> 寄存器存储区选择位（用于直接寻址） 11 = Bank 3 (0x180 – 0x1FF) 10 = Bank 2 (0x100 – 0x17F) 01 = Bank 1 (0x80h – 0xFFh) 00 = Bank 0 (0x00h – 0x7Fh)
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、ADDLI、SUBLI 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。

## 2.5. 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 刚好是 11 位，所以这时 PCLATH 并不需要用到。

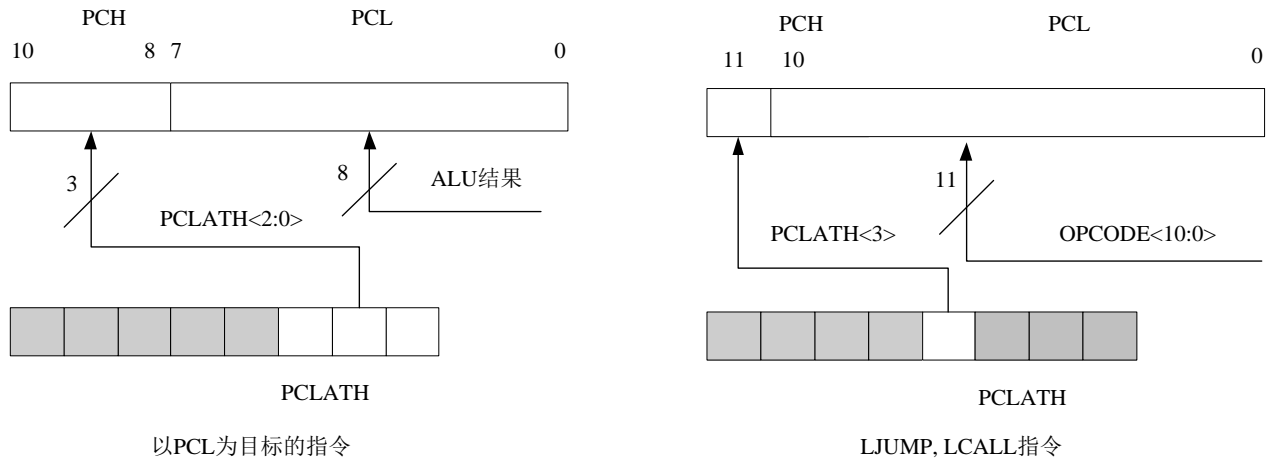


图 2.1 不同情况下 PC 的加载

### 2.5.1. 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<10:8>位被 PCLATH 内容取代。这样可通过将所需的高 3 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。

## 2.6. 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 0–255。任何使用 INDF 寄存器的指令，实际上是对文件选择寄存器 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）。

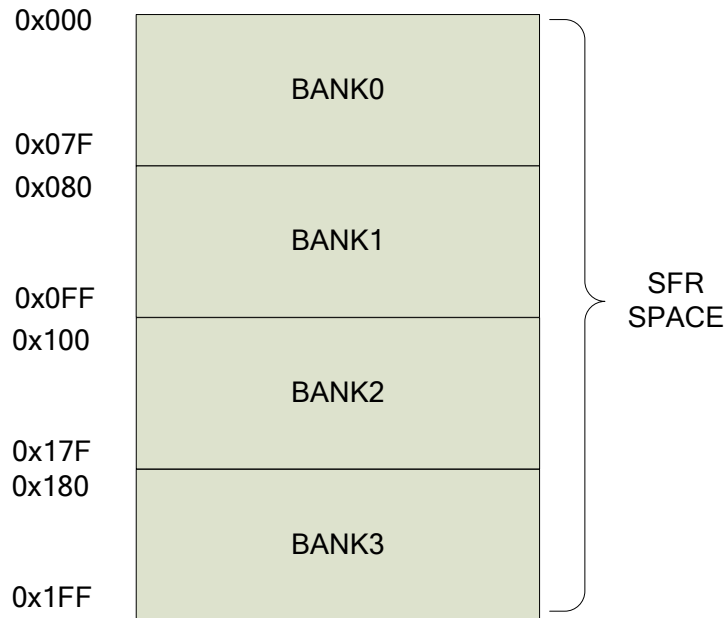


图 2.2 间接寻址

## 2.7. 关于寄存器保留位

如 2.1/1.2 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此位。



### 3. 系统时钟源

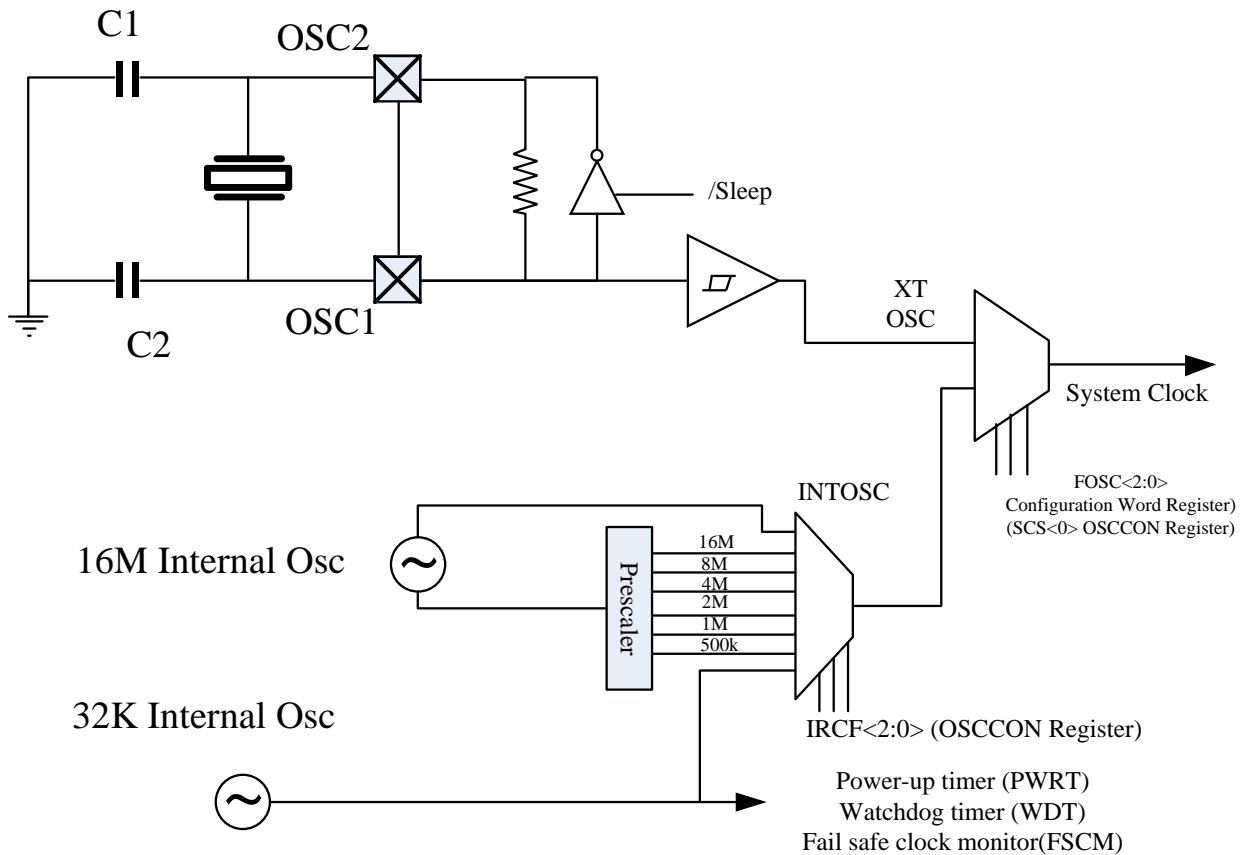


图 3.1 系统时钟源框图

本芯片包含 4 个时钟源：2 个内置 RC 振荡器（高、低速），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(HIRC)，1 个内部 32k/256k(LIRC)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。

#### 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 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：

1. 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）。
2. 由于 OST 和 WDT 共用一个定时器，所以在 OST 未溢出期间，不要写 WDTCN 或 OPTION 寄存器，否则可产生不可预期的行为。

### 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. HIRC（高频内部振荡器）出厂时已校准，工作频率为 16MHz。
2. LIRC（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。

### 3.3.1. 频率选择位 (IRCF)

16MHz HIRC 和 32kHz LIRC 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 (复位后的缺省值)
- 2MHz
- 1MHz
- 500 kHz
- 250 kHz
- 32 kHz

### 3.3.2. HIRC 和 LIRC 时钟切换时序

当在 LIRC 和 HIRC 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 LIRC 和 HIRC 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

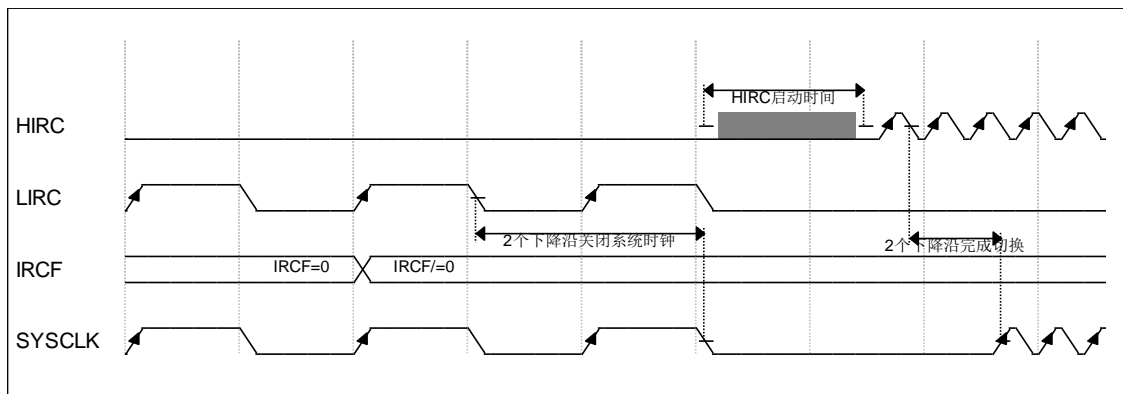


图 3.1 由慢时钟切换到快时钟

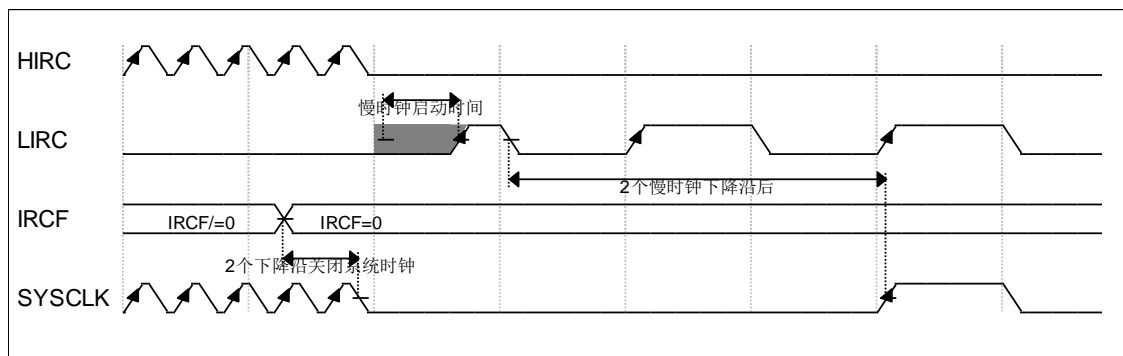


图 3.2 由快时钟切换到慢时钟

### 3.3.3. 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。

FT62F13x 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 8ms 延时之后完成上电配置，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  $V_{POR}$  上升到 2.7V 的  $V_{DDmin}$  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。

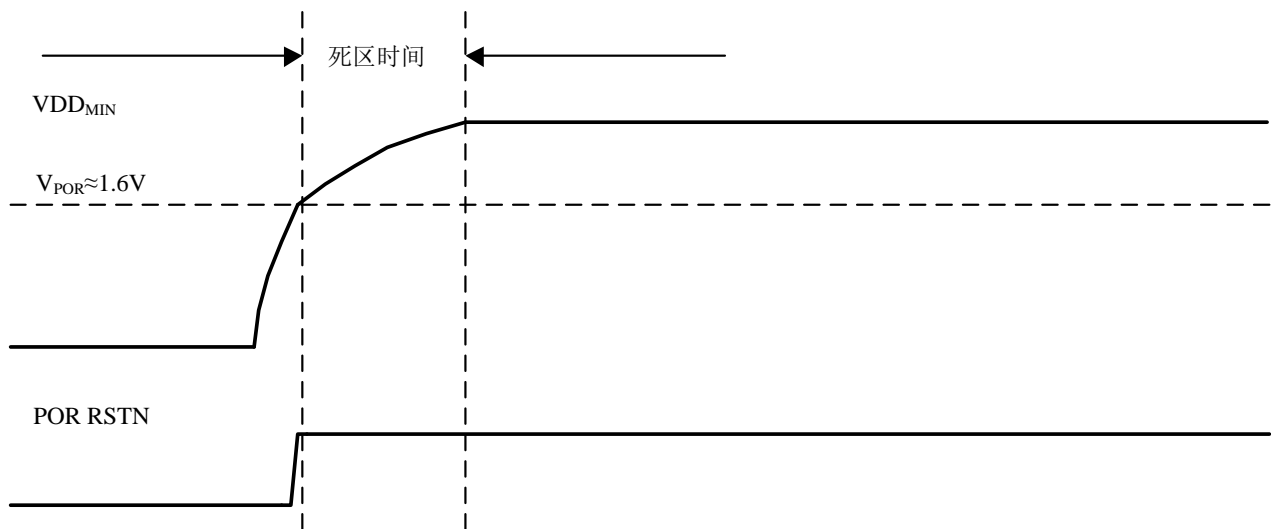


图 3.3 上电过程的“死区时间”

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。

### 3.3.4. HIRC 频率微调

内建的高精度 HIRC 出厂时被校准至 16MHz @ 2.5V/25°C。校准过程是过滤掉制程上的偏差对精度造成的影响，此 HIRC 还会到受工作环境温度和工作电压的影响，其频率会有一些的漂移。

除了出厂校准外，还提供了一种方式供用户对 HIRC 进行微调：通过对 FOSCCAL 寄存器的值进行改写。FOSCCAL 的初始值确保 HIRC 在上电后工作在 16MHz，该值在每颗 IC 上会有差异。设初始值为 FOSCCAL[s]，此时芯片工作在 16M，每改变 1 个 LSB 则 HIRC 频率变化约为 40kHz。FOSCCAL[7:0] 和 HIRC 输出的关系如下：

FOSCCAL[7:0]值	HIRC 实际输出频率（16M 为例）
FOSCCAL[s]-n	(16000-n*40)
.....	.....
FOSCCAL[s]-2	16000-2*40=15920
FOSCCAL[s]-1	16000-1*40=15960
FOSCCAL[s]	16000
FOSCCAL[s]+1	16000+1*40=16040
FOSCCAL[s]+2	16000+2*40=16080
.....	.....
FOSCCAL[s]+n	(16000+n*40)

## 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 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 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 FOOSC<2:0>配置为 LP 或 HS 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 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 HS 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、HS 和 EC）。

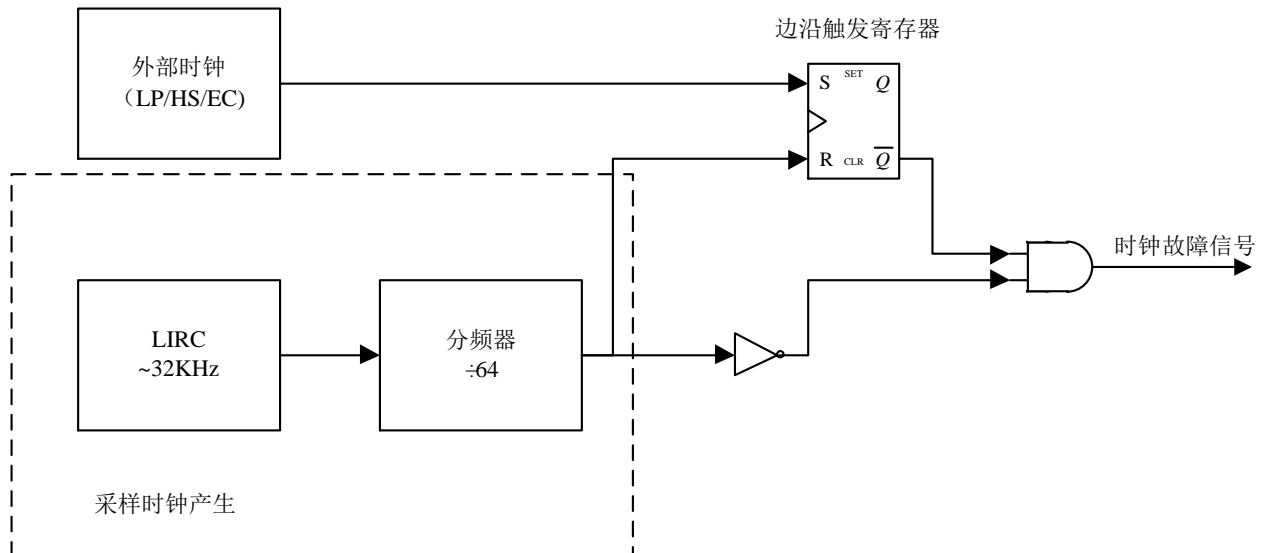


图 3.4 FSCM 原理框图

#### 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。LIRC 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。

**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	CPDB	CPB	MCLRE	PWRTEB	WDTE	FOSC2	FOSC1	FOSC0	1100 0111
OSCCON	0x8F	LFMOD	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS	0101 x000
FOSCCAL	0x0D	HIRC 修调位								0110 1000
MSCON1	0x18E	—							HIRCM	---- ----0

#### 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			x	0	0	0
TYPE	RW	RW			RO	RO	RO	RW

Bit	Name	Function																											
7	LFMOD	LIRC 的频率选择（系统时钟不受其影响） 0 = 32kHz 1 = 256kHz																											
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>101</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(LIRC)</td> <td>16kHz(LIRC/2)</td> </tr> </tbody> </table>	值	2T 模式	4T 模式	111	16MHz	8MHz	110	8MHz	4MHz	101	4MHz	2MHz	100	2MHz(default)	1MHz(default)	011	1MHz	500kHz	010	500kHz	250kHz	001	250kHz	125kHz	000	32kHz(LIRC)	16kHz(LIRC/2)
值	2T 模式	4T 模式																											
111	16MHz	8MHz																											
110	8MHz	4MHz																											
101	4MHz	2MHz																											
100	2MHz(default)	1MHz(default)																											
011	1MHz	500kHz																											
010	500kHz	250kHz																											
001	250kHz	125kHz																											
000	32kHz(LIRC)	16kHz(LIRC/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 = HIRC is ready 0 = HIRC is not ready																											
1	LTS	低速内部时钟状态 1 = LIRC is ready 0 = LIRC 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																											

### 3.7.2. FOSCCAL 寄存器，地址 0x0D

Bit	7	6	5	4	3	2	1	0
Name	FOSCCAL[7:0]							
Reset	0	1	1	0	1	0	0	0
TYPE	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	FOSCCAL	片内 HIRC 时钟校准位

### 3.7.3. MSCON1 寄存器，地址 0x18E

Bit	7	6	5	4	3	2	1	0
Name	—							HIRCM
Reset	—							0
TYPE	RO.0	RO.0	RO.0	RO.0	RO.0	RO.0	RO.0	RW

Bit	Name	Function
7:1	N/A	保留位，读 0
0	HIRCM	<p>HIRC 模式选择：</p> <p>0 = HIRC 中心频率为 16MHz</p> <p>1 = HIRC 中心频率为 13.5MHz</p> <p><b>注意：</b></p> <p>当选择 13.5M 的振荡频率时，需要先把 FOSCCAL 的值保存下来，然后把 SECCODE 的值拷贝到 FOSCCAL，以校准 13.5M 频率；</p> <p>如果要切换到 16M 的中心频率，则需要把之前保存的值写回到 FOSCCAL；</p>

## 4. 复位源

FT62F13X 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) MCLR<sub>B</sub> 管脚复位 – 在常规运行期间
- D) MCLR<sub>B</sub> 管脚复位 – 在睡眠期间
- E) 低电压 (BOR/LVR) 复位
- F) 指令错误复位

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复位事件时恢复到其“复位状态”，见寄存器 SFR 表格。

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 位的置位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<sub>B</sub> 管脚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。

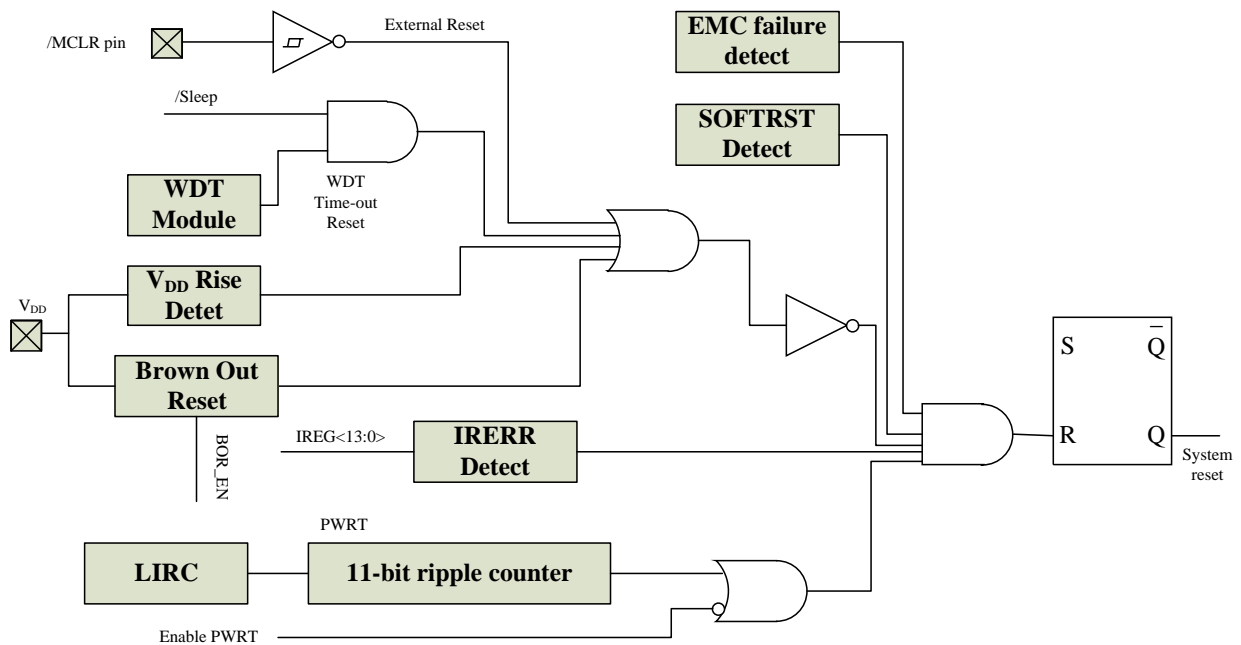


图 4.1 复位框图

## 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。上电复位后，系统复位不会立即释放，还要等一个约 8ms 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）会造成在管脚上产生超标的大电流，因此我们不推荐用户再直接用一个电阻将 MCLR 和 VDD 连接起来，而是采用以下电路。

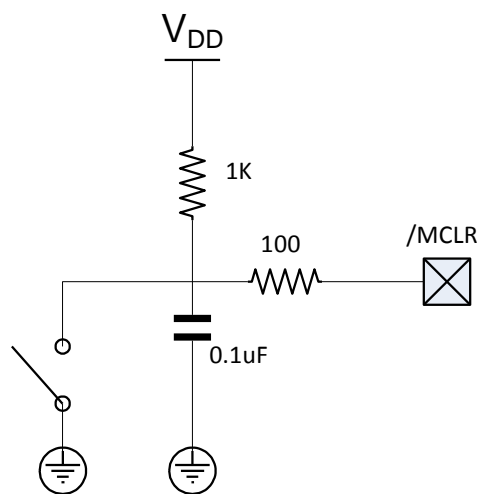


图 4.2 MCLR 复位电路

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 PB7/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。

## 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 MCON0 的 SLVREN 位来控制。低电压复位是指当电源电压低于  $V_{BOR}$  门限电压时所产生的复位。不过当 VDD 电压低于  $V_{BOR}$  不超过  $T_{BOR}$  时间时，低电压复位就不会发生。

$V_{BOR}$  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  $V_{BOR}$  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。

## 4.5. 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<3:0>选择）超过  $T_{BOR}$ （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。如果电源电压大于 LVDL 设置的电压档，该标志位会自动清除，换言之，LVDW 位不具有锁存功能。

### 4.5.1. 检测外部电压

除了可以监控片内 VDD 外，LVD 模块还具备检测外部电压的功能。寄存器位 LVDM 决定了 LVD 作用于 VDD 还是外部电压，当选择对 EVLDx 进行检测时，对应管脚的斯密特输入被关闭以防漏电。

注意：

1. ELVD 管脚的电压不能高于 VDD。

## 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. 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：

- a) POR 结束后，启动 8ms 计时
- b) 启动芯片配置过程（BOOT）
- c) 启动 PWRT 计时（如果使能）

### 4.7.1. PCON 寄存器

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。

## 4.8. 上电配置过程

发生上电复位或者低电压复位后，除了固有的 8ms 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。

## 4.9. 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，3k 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 4.4 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 3ms。

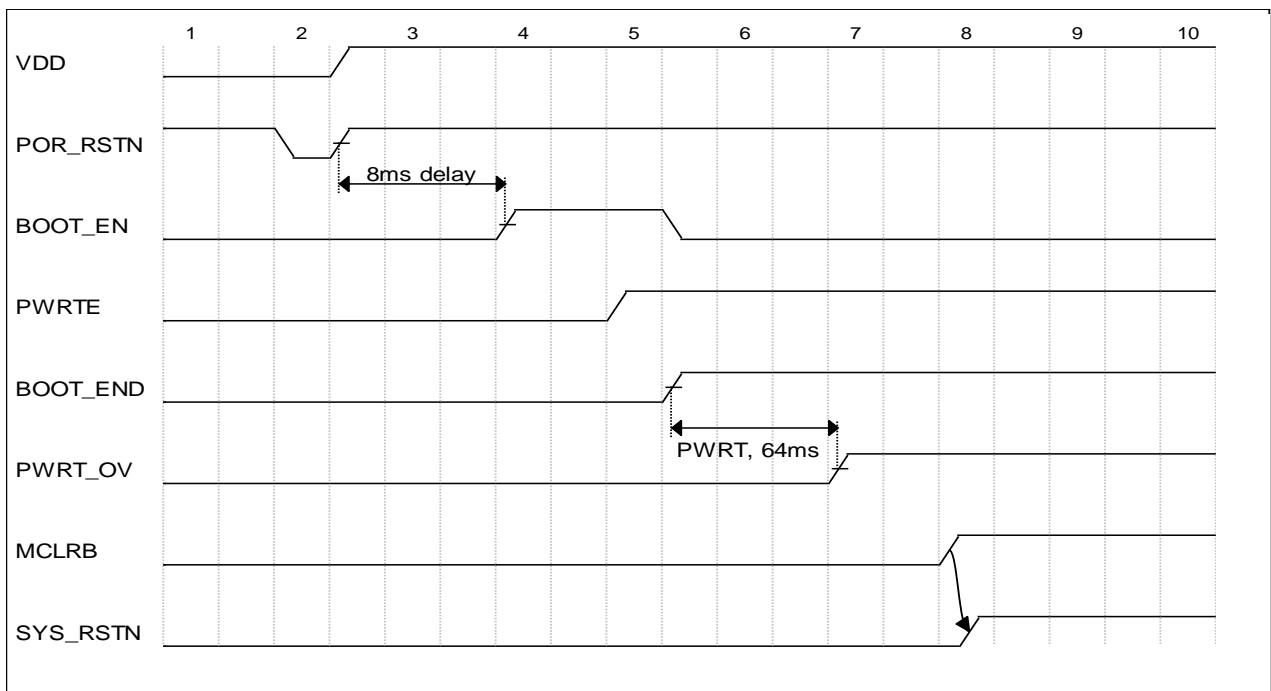


图 4.3 上电复位，使用了 MCLR

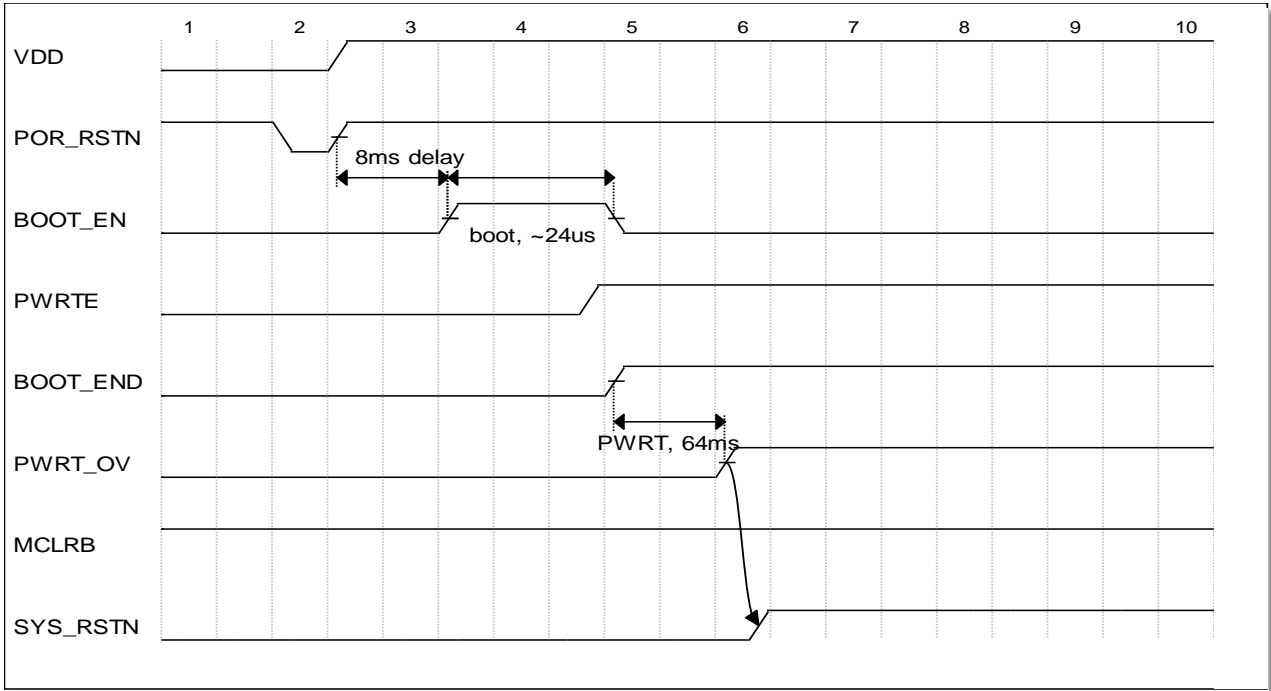


图 4.4 上电复位，没使用 MCLR

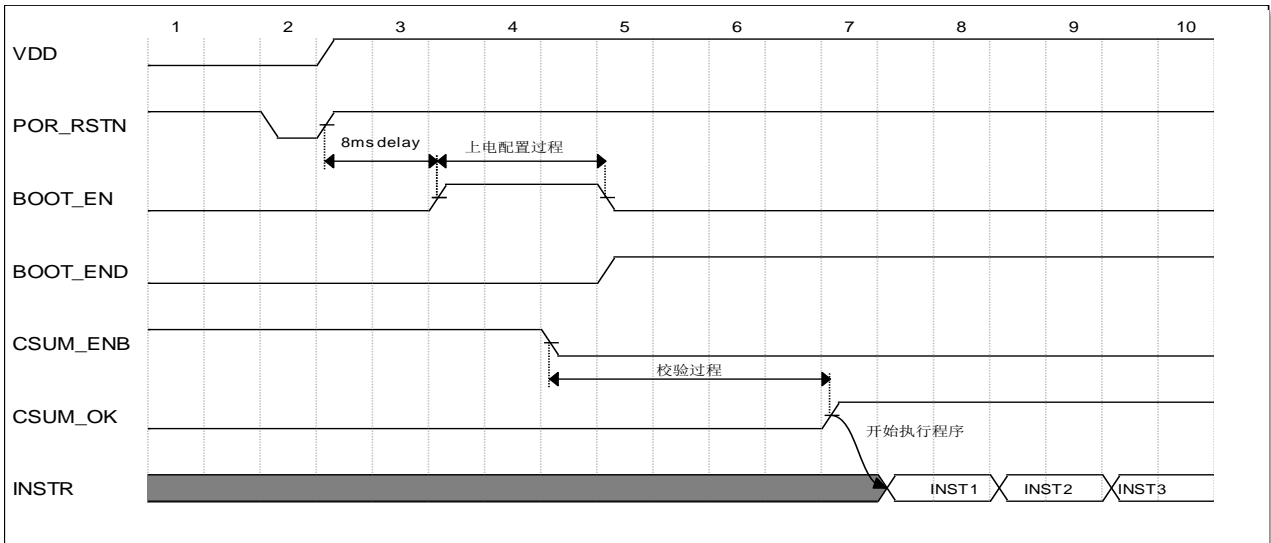


图 4.5 校验过程

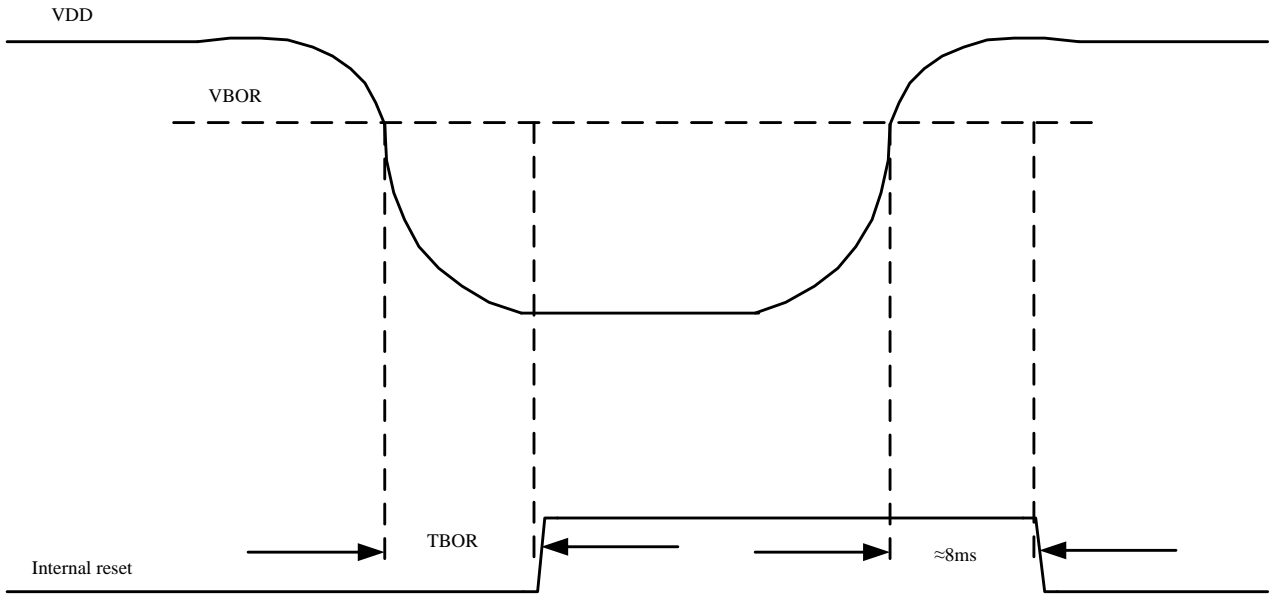


图 4.6 BOR 复位

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 **8ms** 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 <sub>B</sub> 复位
u	u	1	0	睡眠下 MCLR <sub>B</sub> 复位

表 4.2 STATUS/PCON 位及其意义 (u-没变化 x-未知)



### 4.10. PCON 寄存器，地址 0x8E

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

Bit	Name	Function
7:4	LVDL	低电压侦测选择位 值      检测电压 0000: 保留 0001: 2.0V 0010: 2.4V 0011: 2.7V 0100: 3.0V 0101: 3.3V 0110: 3.6V 0111: 4.0V 1xxx: 1.2V
3	LVDEN	低电压侦测使能 1:开启 LVD 侦测功能 0:关闭 LVD 侦测功能
2	LVDW	低电压标志位，只读 当 LVDP=0 时： 1:VDD 掉到了 LVDL[2:0]所设置的电压，时间超过 $T_{BOR}$ 0:VDD 高于 LVDL[2:0]所设置的电压 当 LVDP=1 时： 1:VDD 高于 LVDL[2:0]所设置的电压，时间超过 $T_{BOR}$ 0:VDD 掉到了 LVDL[2:0]所设置的电压 <b>注意：当 LVDP 和 LVDEN 同时为 1 时，LVDW 固定为 1，LVDIF 也被置 1。</b>
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

### 4.11. LVDCON 寄存器，地址 0x110

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	LVDP	LVDDDEB	LVDM[2:0]		
Reset	0	0	0	0	1	1	0	0
Type	RO.0	RO.0	RO.0	RW	RW	RW	RW	RW

Bit	Name	Function
7:5	N/A	保留位，读 0
4	LVDP	LVDP 的极性选择，复位值为 0 1:LVDP 标志位表示 VDD 高于所设阈值 0:LVDP 标志位表示 VDD 低于所设阈值
3	LVDDDEB	LVDD 电平输出是否经过去抖电路 1:经过去抖电路 0:不经过去抖电路
2:0	LVDM	LVDD 检测源选择 000:PB6 001:PB5 010:PA1 011:PA0 100:VDD 101/110/111:TOUCH

### 4.12. LVDTRIM 寄存器，地址 0x19F

Bit	7	6	5	4	3	2	1	0
Name	—	LVDADJ[3:0]				LVRADJ[2:0]		
Reset	0	1	0	0	0	0	1	1
Type	RO	RW	RW	RW	RW	RO	RO	RO

Bit	Name	Function
7	N/A	保留位，读 0
6:3	LVDADJ	LVDD 修调位，上电复位值1000 3% / step
2:0	LVRADJ	只读，LVR精度调节寄存器，上电复位值为011

## 4.13. 配置寄存器汇总

### 4.13.1. UCFG0, PROM 地址 0x2000

Bit	7	6	5	4	3	2	1	0
Name	CPDB	CPB	MCLRE	PWRTEB	WDTE	FOSC[2:0]		
POR val.	1	1	0	1	0	111		

位	名称	描述
7	CPDB	<p>DROM代码保护位</p> <p>1:DROM内容不保护</p> <p>0:启动DROM内容保护, MCU能读, 串口不能读</p> <p>注意:</p> <p>1. 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 CPDB 才变为 1</p> <p>2. 在 DROM 保护的情况下, 全芯片擦除结束后, DROM 将被自动擦除</p>
6	CPB	<p>Flash 全区域 (4K words) 保护设置</p> <p>1:不对 Flash 进行全区域保护</p> <p>0:启用 Flash 全区域保护, 除了 CPU 取指, CPU 或外部串口皆读返回 0</p> <p>注意:</p> <p>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</p>
5	MCLRE	<p>1:PB7/MCLR<sub>B</sub> 为复位脚功能</p> <p>0:PB7/MCLR<sub>B</sub> 脚为 GPIO</p>
4	PWRTEB	<p>1:PWRT 禁止</p> <p>0:PWRT 使能</p>
3	WDTE	<p>1:WDT 使能, 程序不能禁止</p> <p>0:WDT 禁止, 但程序可通过设置 WDTCON 的 SWDTEN 位将 WDT 使能</p>
2:0	FOSC	<p>000:LP晶振模式, PC0/PC1接低速晶体</p> <p>001:HS晶振模式, PC0/PC1接高速晶体</p> <p>010:外部时钟模式, PC0为IO功能, PC1接时钟输入</p> <p>011:INTOSC模式, PB0(PA2)输出系统时钟的2分频, PC1为IO管脚</p> <p>1xx:INTOSCIO 模式, PC0 为 IO 管脚, PC1 为 IO</p>

### 4.13.2. UCFG1, PROM 地址 0x2001

Bit	7	6	5	4	3	2	1	0
Name	—	CSUMEB	TSEL	FCMEN	IESO	RDCTRL	LVREN[1:0]	
POR val.	RO.0	1	1	1	1	0	11	

位	名称	描述
7	N/A	保留位
6	CSUMEB	Checksum enable 1:禁止 checksum 功能 0:使能 checksum 功能, boot 完成后, 硬件自动把 2K 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
5	TSEL	指令周期选择位 1:指令周期为 2T 0:指令周期为 4T
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
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
2	RDCTRL	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
1:0	LVREN	低电压复位选择 00:使能低电压复位 01:LVR 由 MSCON0 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

**4.13.3. UCFG2, PROM 地址 0x2002**

Bit	7	6	5	4	3	2	1	0
Name	—	IRBTE	MRBTE	WDTBTE	LVRS[3:0]			
POR val.	1	0	0	0	4'b0000			

位	名称	描述																																		
7	N/A	保留位																																		
6	IRBTE	IRERR复位BOOT使能位 0 = 禁止错误指令复位产生BOOT 1 = 允许错误指令复位产生 BOOT																																		
5	MRBTE	MCLR复位BOOT使能位 0 = 禁止MCLR复位产生BOOT 1 = 允许 MCLR 复位产生 BOOT																																		
4	WDTBTE	WDT复位BOOT使能位 0 = 禁止WDT复位产生BOOT 1 = 允许 WDT 复位产生 BOOT																																		
3:0	LVDS	低电压复位阈值选择 <table border="1"> <thead> <tr> <th>数值</th> <th>电压</th> </tr> </thead> <tbody> <tr><td>1010:</td><td>保留</td></tr> <tr><td>1011:</td><td>保留</td></tr> <tr><td>1100:</td><td>保留</td></tr> <tr><td>1101:</td><td>保留</td></tr> <tr><td>1110:</td><td>保留</td></tr> <tr><td>1111:</td><td>保留</td></tr> <tr><td>0000:</td><td>保留</td></tr> <tr><td>0001:</td><td>保留</td></tr> <tr><td>0010:</td><td>保留</td></tr> <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	1010:	保留	1011:	保留	1100:	保留	1101:	保留	1110:	保留	1111:	保留	0000:	保留	0001:	保留	0010:	保留	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																																			
1010:	保留																																			
1011:	保留																																			
1100:	保留																																			
1101:	保留																																			
1110:	保留																																			
1111:	保留																																			
0000:	保留																																			
0001:	保留																																			
0010:	保留																																			
0011:	2.0V																																			
0100:	2.2V																																			
0101:	2.5V																																			
0110:	2.8V																																			
0111:	3.1V																																			
1000:	3.6V																																			
1001:	4.1V																																			

#### 4.13.4. UCFG3, PROM 地址 0x2003

Bit	7	6	5	4	3	2	1	0
Name	—					SECPB2	SECPB1	SECPB0
POR val.	RO.0	RO.0	RO.0	RO.0	RO.0	1	1	1

位	名称	描述
7:3	N/A	保留位，需保持为 1
2:0	SECP2/1/0	<p>PROM 分扇区保护控制位，低有效。1 扇区=1kW，每个 bit 控制一个扇区</p> <p>SECPx 值： 0 = 扇区 x 被保护，串口不能读或者编程，只能擦除 1 = 扇区 x 不被保护</p>

#### 4.13.5. MAINCSUM (地址: 0x2007)

Bit	7	6	5	4	3	2	1	0
Name	—				MAINCSUM[3:0]			
POR val.	RO.0	RO.0	RO.0	RO.0	x	x	x	x

位	名称	描述
7:4	N/A	保留位，读0
3:0	MAINCSUM	4位3K主程序区的校验和，由上位机在烧录程序时

在上电配置过程 (BOOT) 完成后，当 Config1 的 CSUMEB 为 0 时，硬件将 3K 的主程序区所有单元进行累加。在做完最后一个字 (地址 0xBFF) 的加法操作后，累加结果与 MAINCSUM 的值做比较，如果两者相等，则校验成功，否则校验失败，复位模块将产生一次 reboot 复位。

## 5. 看门狗定时器

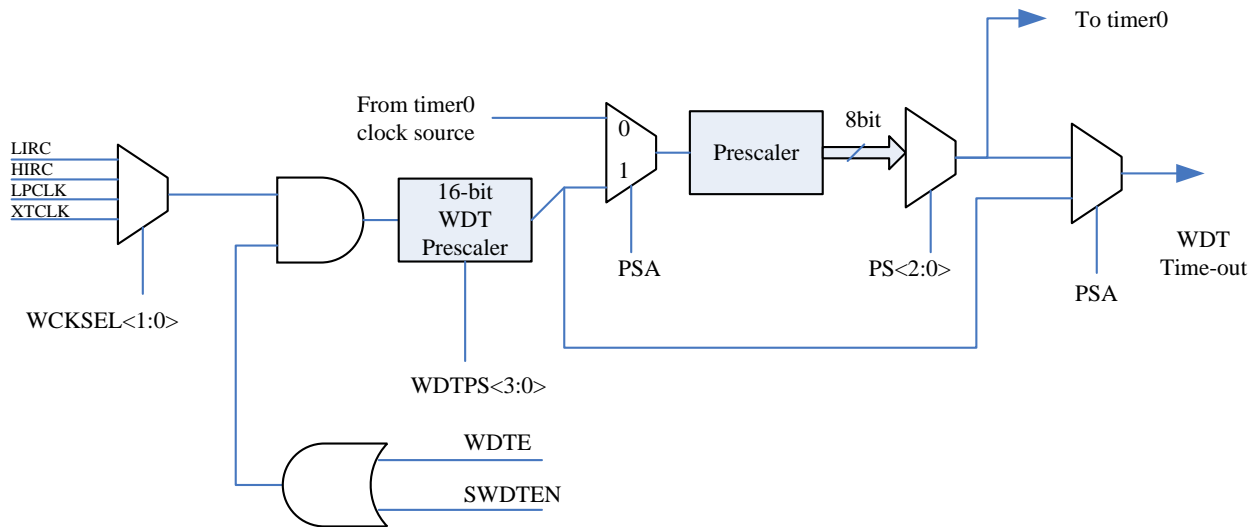


图 5.1 看门狗和定时器 0 框图

### 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，硬件使能位 **WDTEN** 位于配置寄存器 **UCFG0** 的第 3 位，软件使能位 **SWDTEN** 位于 **WDTCON** 寄存器的第 0 位，为 1 时表示使能看门狗，为 0 时禁止。

清看门狗指令 **CLRWDWT**、**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
CLRWDWT 指令	
进入 SLEEP、退出 SLEEP 时刻	
OST 计数溢出时	
写 WDTCON	

## 5.2. 看门狗时钟源

看门狗时钟源有以下几种：

- LIRC
- HIRC
- LP 晶体时钟
- XT 晶体时钟

在看门狗被使能的情况下，被选择的时钟源将自动打开，在 SLEEP 模式下也将保持运行。

注意：

1. 选择 LP 作为 WDT 时钟源时，配置选项 FOSC 必须是 LP 或者 INTOSCIO 模式，否则，LP 晶体电路将不打开；
2. 同理，选择 XT 作为 WDT 时钟源时，配置选项 FOSC 必须是 XT 或者 INTOSCIO 模式，否则，XT 晶体电路将不打开；

## 5.3. 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
WDTCON	0x18	—	WCKSRC[1:0]		WDTPS[3:0]			SWDTEN		-000 1000
UCFG0	0x2000	CPDB	CPB	MCLRE	PWRTEB	WDTE	FOSC[2:0]			1100 0111
OPTION	0x81	/PAPU	INTEDG	T0CS	T0SE	PSA	PS[2:0]			1111 1111



**5.3.1. WDTCON 寄存器，地址 0x18**

Bit	7	6	5	4	3	2	1	0
Name	Reserved	WCKSRC[1:0]		WDTPS[3:0]				SWDTEN
Reset	0	0	0	0	1	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	Reserved	保留位
6:5	WCKSRC	看门狗时钟源选择 00 = LIRC(256k or 32k) 01 = HIRC 10 = LP 晶体时钟，只有当 FOSC 配置为 LP 或 INTOSCIO 模式时才有意义 11 = XT 晶体时钟，只有当 FOSC 配置为 XT 或 INTOSCIO 模式时才有意义
4:1	WDTPS	<b>WDTPS&lt;3:0&gt;</b> : 看门狗定时器周期选择: Bit Value = 看门狗定时器驱动时钟之预分频比值 0000 = 1:32 0001 = 1:64 0010 = 1:128 0011 = 1:256 0100 = 1:512 (复位值) 0101 = 1:1024 0110 = 1:2048 0111 = 1:4096 1000 = 1:8192 1001 = 1:16384 1010 = 1:32768 1011 = 1:65536 1100 = 1:65536 1101 = 1:65536 1110 = 1:65536 1111 = 1:65536
0	SWDTEN	软件打开/关闭看门狗定时器: 1 = 打开 0 = 关闭

## 6. 中断

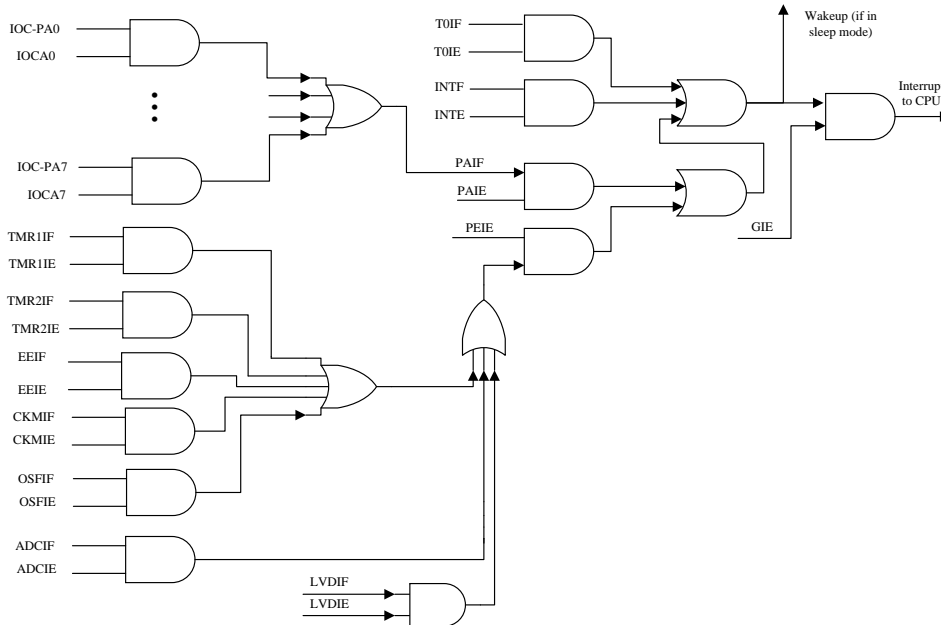


图 6.1 中断逻辑框图

FT62F13x 有以下中断源，部分中断可以把 CPU 从睡眠状态唤醒：

- ADC 中断
- PA4/INT 管脚进来的外部中断
- Timer0 溢出中断
- PORTA 电平变化中断
- Timer2 匹配相等中断
- Timer1 匹配相等中断
- EEPROM 数据写中断
- 时钟缺失中断
- LVD 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：

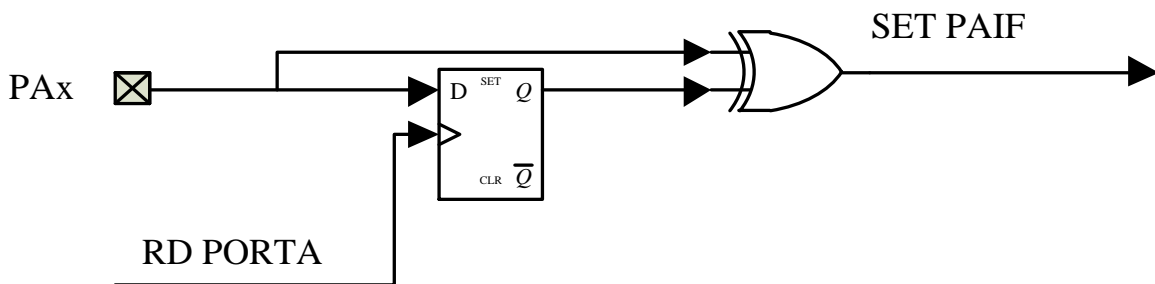
- INT 管脚中断
- PORTA 变化中断
- Timer0 溢出中断

PIR1 中包含着外围中断标志位，PIE1 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。

## 6.1. INT 外部中断

INT 管脚上的外部中断是边沿触发的：当 OPTION 寄存器的 INTEDG 位被置 1 时在上升沿触发，而当 INTEDG 位被清零时在下降沿触发。当 INT 管脚上出现有效边沿时，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 管脚中断能将处理器从休眠状态唤醒。

## 6.2. 端口变化中断



6.2 端口变化中断原理框图

PORTA 输入电平的变化会使 INTCON 寄存器的 PAIF 位置 1。可以通过置 1/清零 INTCON 寄存器的 PAIF 位来使能/禁止该中断。此外，可通过 IOCA 寄存器对该端口的各个管脚进行配置。

注意：

1. 初始化电平变化中断时，应先配置为数字输入 IO，把相应的 IOCA 置 1，然后读取一下该 PORTA；
2. 当 I/O 电平发生变化时，PAIF 被置 1；
3. 清中断标志位之前应该读取一下 PORTA，然后再对 PAIF 清 0；

### 6.2.1. PAIF 标志位的清除

PAIF 寄存位是异步置位的，即，如果端口不匹配事件一直存在，软件是无法完成对其清除操作的。要想把它清 0，有以下两种方法：

方法 1

- a) 等待外部管脚恢复原来的电平
- b) 软件清 PAIF

方法 2

- a) 读取 PORTA，主动清除不匹配事件
- b) 软件清除 PAIF

### 6.3. 中断响应

外部中断包括 INT 管脚进来的或者 PORTA 变化中断的延时一般为 1 到 2 个指令周期。它跟中断发生时刻与正在执行的指令有关系。

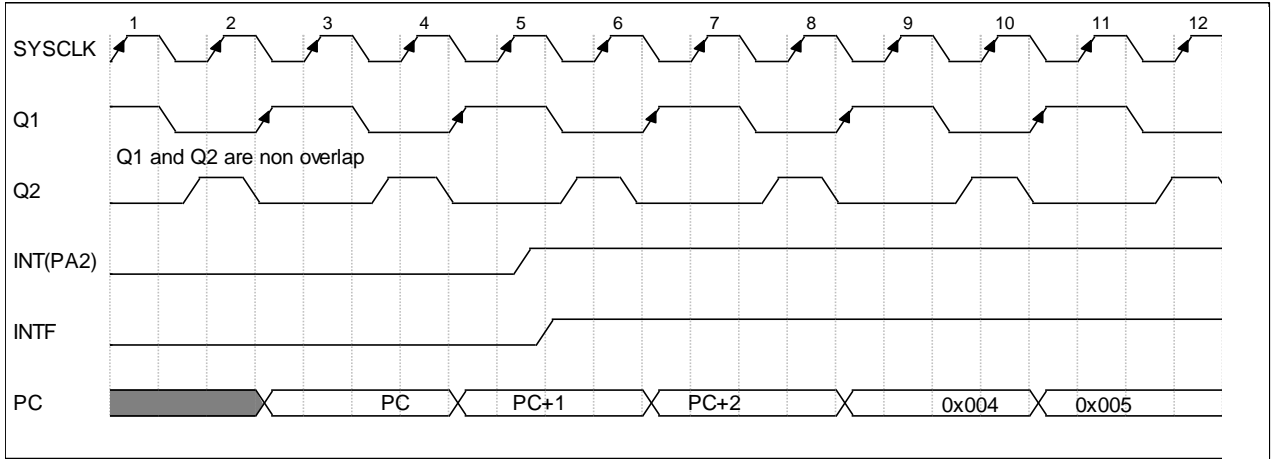


图 6.3 中断响应时序图

### 6.4. 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 落在各个页区间，因此可以稍微节省代码。

### 6.5. 关于中断标志位

所有外设的中断标志位均独立于其中断使能（允许）位，即使各中断使能位为 0，发生了相关中断事件，其标志位仍然会被置 1。

## 6.6. 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	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
PIE1	0x8C	EEIE	CKMIE	LVDIE	ACMPIE	TMR1IE	OSFIE	TMR2IE	ADCIE	0000 0000
PIR1	0x0C	EEIF	CKMIF	LVDIF	ACMPPIF	TMR1IF	OSFIF	TMR2IF	ADCIF	0000 0000
TRISA	0x85	TRISA[7:0], PORTA 方向控制								1111 1111
IOCA	0x96	IOCA[7:0], 端口变化中断允许位								0000 0000

### 6.6.1. INTCON 寄存器，地址 0x0B/0x8B

Bit	7	6	5	4	3	2	1	0
Name	GIE	PEIE	T0IE	INTE	PAIE	T0IF	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	GIE: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	PEIE: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	T0IE	T0IE:定时器0溢出中断使能 1 = 允许Timer0中断 0 = 禁止Timer0中断
4	INTE	INTE: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	PAIE: 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	T0IF	T0IF: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	INTF: PA4/INT外部中断标志位 1 = 发生了PA4/INT外部中断 (必须用软件清零) 0 = 未发生PA4/INT外部中断
0	PAIF	PAIF: 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

### 6.6.2. PIR1 寄存器，地址 0x0C

Bit	7	6	5	4	3	2	1	0
Name	EEIF	CKMIF	LVDIF	ACMPIF	TMR1IF	OSFIF	TMR2IF	ADCIF
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	EEIF	EEIF: EE写中断标志位 1 = EE 写操作完成 0 = EE 写操作未完成，或已经由软件清 0
6	CKMIF	CKMIF: 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 0 = 快时钟测量慢时钟未完成，或已经由软件清 0
5	LVDIF	LVDIF: LVD中断标志位 1 = LVD检测电压低于所设置阈值 0 = LVD检测电压高于所设置阈值，或已经由软件清0
4	ACMPIF	ADC 阈值比较中断标志位 1 = ADC 阈值比较结果高于预设值 0 = ADC 阈值比较结果低于预设值，或已经由软件清 0
3	TMR1IF	TMR1IF: Timer1与PR1比较相等中断标志位 1 = Timer1 的值等于PR1 0 = Timer1 的值不等于 PR1，或已经由软件清 0
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 0 = 系统时钟运行正常，或已经由软件清 0
1	TMR2IF	TMR2IF: Timer2与PR2比较相等中断标志位 1 = Timer2 的值等于PR2 0 = Timer2 的值不等于 PR2，或已经由软件清 0
0	ADCIF	ADC 转换完成中断标志位 1 = ADC 转换完成 0 = ADC 转换未完成，或已由软件清 0

### 6.6.3. PIE1 寄存器，地址 0x8C

Bit	7	6	5	4	3	2	1	0
Name	EEIE	CKMIE	LVDIE	ACMPIE	TMR1IE	OSFIE	TMR2IE	ADCIE
Reset	0	0	0	0	0	0	0	0
TYPE	RW	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	CKMIE	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	LVDIE	按键中断使能位 1 = 使能按键中断 0 = 禁止按键中断
4	ACMPIE	ADC阈值比较结果中断使能位 1 = 使能 ADC阈值比较中断 0 = 关闭 ADC 阈值比较中断
3	TMR1IE	Timer1 与PR1比较相等中断使能位 1 = 使能 Timer1的匹配中断 0 = 关闭 Timer1 的匹配中断
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的匹配中断 0 = 关闭 Timer2 的匹配中断
0	ADCIE	ADC 转换完成中断使能 1 = 允许ADC转换中断 0 = 禁止 ADC 转换中断

### 6.6.4. IOCA 寄存器，地址 0x96

Bit	7	6	5	4	3	2	1	0
Name	IOCA[7:0]							
Reset	0x00							
Type	RW							

Bit	Name	Function
7:0	IOCA	端口变化中断设置 1 = 使能端口变化中断 0 = 关闭端口变化中断

## 7. 睡眠模式

器件通过执行 **SLEEP** 指令进入睡眠模式。

进入休眠模式时，MCU 的状态如下：

1. WDT 将清零但是保持运行（如果使能了在休眠期间工作）
2. STATUS 寄存器的 PD 位清零
3. STATUS 寄存器的 TO 位置 1
4. CPU 时钟停止
5. 32kHz LIRC 不受影响，并且由其提供时钟的外设可以在休眠模式下继续工作
6. LP 晶体振荡器不受影响（当 TIMx 使用它作为工作时钟时）
7. ADC 不受影响（如果选择了专用 FRC 时钟）
8. I/O 端口保持执行 **SLEEP** 指令之前的状态（驱动为高电平、低电平或高阻态）
9. WDT 之外的复位不受休眠模式影响

关于外设的在休眠期间工作的更多详细信息，请参见各个章节。

要最大程度地降低电流消耗，应考虑以下条件：

1. I/O 引脚不应悬空，I/O 作为输入时可打开内部的上拉或下拉
2. 外部电路从 I/O 引脚灌电流
3. 内部电路从 I/O 引脚拉电流
4. 内部弱上拉的引脚
5. 模块使用 31kHz LIRC
6. 模块使用 LP 振荡器

### 7.1. 唤醒模式

以下事件可以唤醒芯片：

- MCLR<sub>B</sub> 管脚上有外部复位
- BOR 复位（如果使能）
- POR 复位
- WDT 超时溢出
- 任何外部中断
- 能够在休眠期间运行的外设产生的中断（更多信息请参见各个外设）

前 3 个事件会使器件复位，后 3 个事件认为是程序执行的延续。

清看门狗指令 **CLRWD<sub>T</sub>**、**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。



## 7.2. 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 WDTCON 寄存器。

清除看门狗指令 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。

## 7.3. 中断唤醒

中断唤醒时，CPU 先进入中断处理程序，退出中断之后，再执行 SLEEP 的下一条指令。

## 7.4. 关于 SLEEP 后的第一条指令

需要把紧跟 SLEEP 后面的指令写为 NOP，因为使用非中断方式唤醒时（如 WDT 唤醒，GIE 未使能时的中断事件唤醒），该指令会被执行两遍。

如下示例程序所示：

```
SLEEP      // 睡眠  
NOP        // 非中断唤醒时，该指令将执行两遍
```

## 8. 数据 EEPROM

片内集成有 128 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。

### 8.1. 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 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. 如编程完成前 **WREN3、WREN2** 或者 **WREN1** 任意一位清 0，在下次编程前需清除 **EEIF** 标志位。

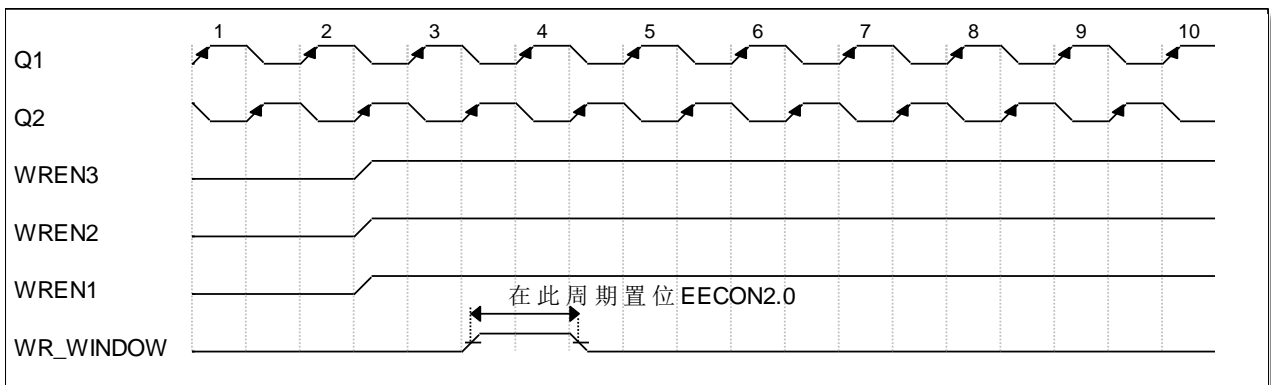


图 8.1 软件编程数据 EEPROM 时序

## 8.2. 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。EEPROM 读周期为 4 个系统时钟，所以在发起读操作之后的第 2 个指令周期，EEDAT 寄存器才被 EEPROM 数据写入，即该数据可由下下一条指令读取。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  
NOP _____ ; 读等待  
LDR EEDAT, W _____ ; 这时，数据可以被软件读取
```

## 8.3. 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。

## 8.4. EEPROM 的单编程模式

EEPROM 除了正常的自擦写模式（编程操作包含了自动擦除），还支持单编程模式，即按 8.1 小节所述步骤启动编程后，所选中地址单元并不做自动擦除，而是仅仅包含编程，通过 PONLY 寄存器位控制。**注意 EEPROM 的单编程模式只能把数据由 1 写成 0，而不能由 0 写为 1，这有点像“按位与”运算。**

举例说明，假如 0 地址存储了数据 0xAA，对其启动正常的自擦写模式（PONLY=0）写 0x55，待操作结束后，0 地址存储的数据变成 0x55；

但如果设置的是单编程模式（PONLY=1），写同样的数据 0x55，最终 0 地址存放的数据将会是 0x00。

### 8.5. 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	TOIE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
PIE1	0x8C	EEIE	CKMIE	LVDIE	ACMPIE	TMR1IE	OSFIE	TMR2IE	ADCIE	0000 0000
PIR1	0x0C	EEIF	CKMIF	LVDIF	ACMPPIF	TMR1IF	OSFIF	TMR2IF	ADCIF	0000 0000
EEDAT	0x9A	EEDAT[7:0]								0000 0000
EEADR	0x9B	EEADR[7:0]								0000 0000
EECON1	0x9C	—	Reserved	WREN3	WREN2	WRERR	WREN1	PONLY	RD	-000 x000
EECON2	0x9D	—	—	—	—	—	—	—	WR	---- ---0

#### 8.5.1. 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

Bit	Name	Function
7:0	EEDAT	数据 EEPROM 数据寄存器

#### 8.5.2. 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

Bit	Name	Function
7:0	EEADR	数据 EEPROM 地址寄存器

### 8.5.3. EECON1 寄存器，地址 0x9C

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

Bit	Name	Function
7	N/A	保留位，读 0
6	Reserved	保留位
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	PONLY	数据 EEPROM 写编程模式 1:只写不擦除 0:先擦除再写
0	RD	数据 EEPROM 读控制位 使能 RD 时，此位高电平只持续两个指令周期，之后读永远返回 0 写 1:启动一次数据 EEPROM 读周期 写 0:不启动读

### 8.5.4. EECON2 寄存器，地址 0x9D

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

Bit	Name	Function
7:1	N/A	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

## 9. 12bit 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 12 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 12 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、外部参考电压或内部产生的参考电压。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。

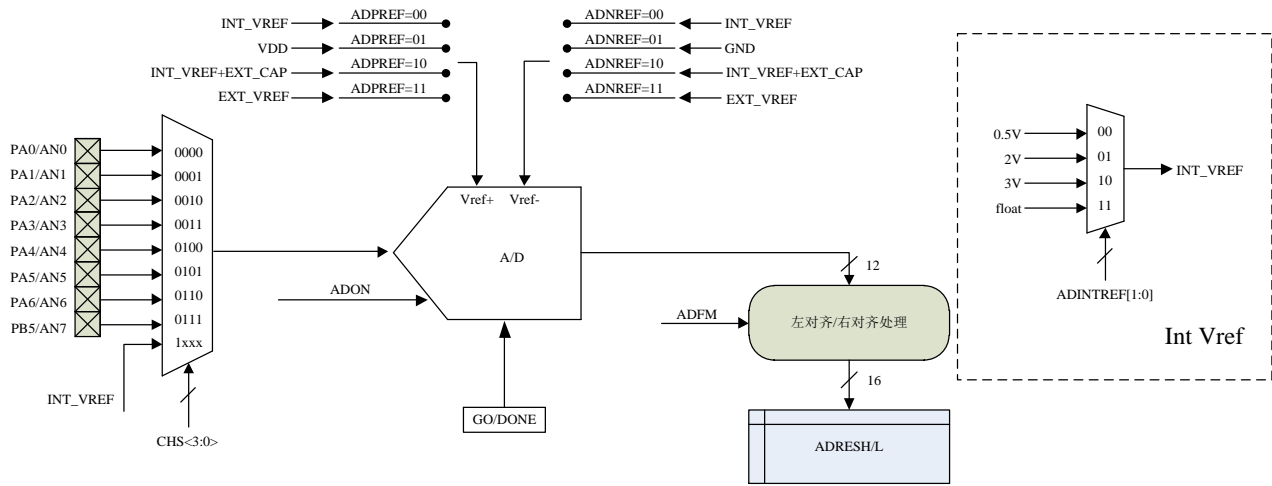


图 9.1 ADC 原理框图

### 9.1 ADC 的配置

配置和使用 ADC 时，必须考虑以下功能：

- 校准 ADC
- 端口配置
- 通道选择
- 触发方式选择
- 触发源选择
- 触发类型选择
- 触发延时配置
- 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
- 阈值比较

**注意：**在进行各项配置更改的时候，需要确保 AD 转换并未正在进行或外部触发功能未开启。建议在 ADON 关闭时进行更改。

### 9.1.1 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，会导致输入缓冲器传导过大的电流。

### 9.1.2 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，根据采样稳定的需要在启动转换前加入一定延时，硬件已固定有  $1.5T_{AD}$  的采样延时。更多信息请参见第 9.2 节“ADC 的工作原理”。

### 9.1.3 触发方式选择

ADCON0 寄存器的 ADEX 位决定是否使用外部触发信号。

若 ADEX=0 时，ADGO 可由程序置位，AD 转换完成自动清零。

若 ADEX=1 时，ADGO 将由外部硬件触发置位，AD 转换完成清零。

**注意：**若选择了前沿消隐触发 ADC，即 LEBADT 设为 1 时，需要先置位 ADEX 和 ADON。

### 9.1.4 触发源选择

在设定 ADEX 后，ADCON2 寄存器的 ETGSEL 位决定使用哪个外部触发信号。其中可选 I/O 引脚触发，需要配置相关寄存器。具体请参见相应的端口章节。

### 9.1.5 触发类型选择

ADCON2 寄存器的 ETGTYP 位决定外部触发信号的触发类型。

### 9.1.6 触发延时配置

ADCON2 寄存器的 ADDLY.8 位和 ADDLY 寄存器组成 9 位延时计数器，共同决定外部触发信号的触发延时时间。由于需要同步异步信号，实际延迟时间为： $(ADDLY+6)/F_{ADC}$ 。

**注意：**若选择了前沿消隐触发功能时，则实际延迟时间为： $(ADDLY+3)/F_{T2CK} + 3/F_{ADC}$ 。T2CK 为 Timer2 时钟，由 T2CKSRC 位选择。



## 9.1.7 ADC 参考电压

ADCON1寄存器的ADPREF位提供对正参考电压的控制，ADNREF位提供对负参考电压的控制。正/负参考电压可以是内部参考电压、VDD/GND、内部参考电压加外部电容、外部参考电压。正/负参考电压可以有各种组合，但不可以同时选择位内部参考电压。若发生则强制负参考电压连接到GND。

ADCON2寄存器的ADINTREF位提供对内部参考电压的控制。内部参考电压可以选择0.5V、2V、3V或者悬空。

## 9.1.8 转换时钟

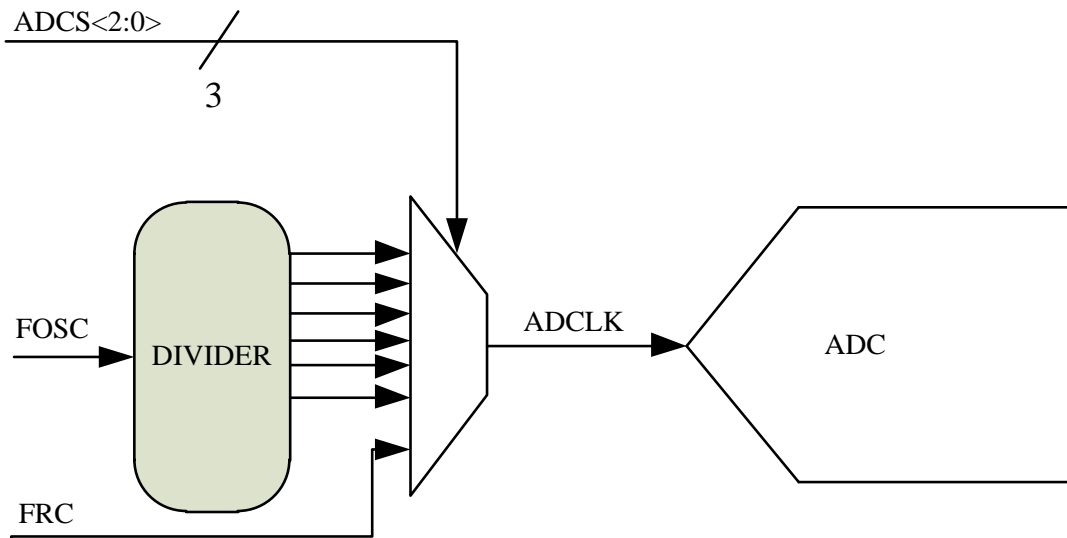


图 9.2 ADC 的时钟配置原理

转换时钟源可通过 ADCON1 寄存器的 ADCS 位用软件选择。有以下 8 种时钟选项：

- $F_{OSC}$
- $F_{OSC}/2$
- $F_{OSC}/4$
- $F_{OSC}/8$
- $F_{OSC}/16$
- $F_{OSC}/32$
- $F_{OSC}/64$
- $F_{RC}$ （内部慢时钟振荡器）

完成一位（bit）的转换时间定义为  $T_{AD}$ 。完成 12 位转换需要 12 个  $T_{AD}$  周期（包括  $1.5T_{AD}$  的采样时间和  $1T_{AD}$  的数据传输处理时间），如图 9.3 和 9.6 所示。

进行正确的转换必须满足相应的  $T_{AD}$  规范。更多信息请参见第 21 节“电气特性”中的 A/D 转换要求。表 9.1 所示为正确选择 ADC 时钟的示例。

注意:

1. 除非使用的是  $F_{RC}$ ，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.  $F_{RC}$  可以是 256kHz 或者是 32kHz，取决于 LFMOD 为何值；
3. 下表灰色单元格表示不支持的频率；

ADC 时钟周期 ( $T_{AD}$ )					
ADC 时钟源	ADCS<2:0>	16MHz	8MHz	4MHz	1MHz
$F_{OSC}$	011	62.5ns	125ns	250ns	1 $\mu$ s
$F_{OSC}/2$	000	125ns	250ns	500ns	2.0 $\mu$ s
$F_{OSC}/4$	100	250ns	500ns	1.0 $\mu$ s	4.0 $\mu$ s
$F_{OSC}/8$	001	0.5 $\mu$ s	1.0 $\mu$ s	2.0 $\mu$ s	8.0 $\mu$ s
$F_{OSC}/16$	101	1.0 $\mu$ s	2.0 $\mu$ s	4.0 $\mu$ s	16.0 $\mu$ s
$F_{OSC}/32$	010	2.0 $\mu$ s	4.0 $\mu$ s	8.0 $\mu$ s	32.0 $\mu$ s
$F_{OSC}/64$	110	4.0 $\mu$ s	8.0 $\mu$ s	16.0 $\mu$ s	64.0 $\mu$ s
$F_{RC}$	x11	4.0 $\mu$ s	4.0 $\mu$ s	4.0 $\mu$ s	4.0 $\mu$ s

表 9.1 ADC 时钟周期和器件工作频率

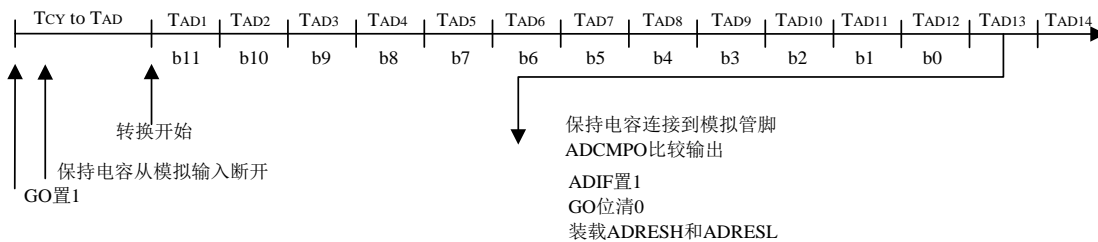


图 9.3 模数转换  $T_{AD}$  周期

### 9.1.9 中断

ADC 模块可使中断在模数转换完成时产生，或者通过转换完成的阈值比较产生。ADC 转换中断标志为 PIR1 寄存器中的 ADIF 位。ADC 中断使能为 PIE1 寄存器中的 ADIE 位。ADIF 位必须用软件置 1 清零。ADC 的阈值比较中断位置为 PIR 1 寄存器中的 ACMPIF。ADC 阈值比较中断使能为 PIE1 寄存器中的 ACMPIE 位。

- 注意:
- 1、无论 ADC 中断是否被打开，ADIF 位在每次正常转换完成时均置 1。
  - 2、软件停止 AD 转换都不会置位 ADIF。
  - 3、仅当在选择了  $F_{RC}$  振荡器，ADC 才能在休眠期间工作。

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，中断可唤醒器件。从休眠唤醒时，先进入中断处理程序，退出中断后，再执行 SLEEP 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。

### 9.1.10 转换结果的格式

12 位 A/D 转换结果有两种格式，即左对齐和右对齐。ADCON1 寄存器的 ADFM 位控制输出格式。AD 自动校准值也受输出格式影响。

图 9.4 所示为两种输出格式。

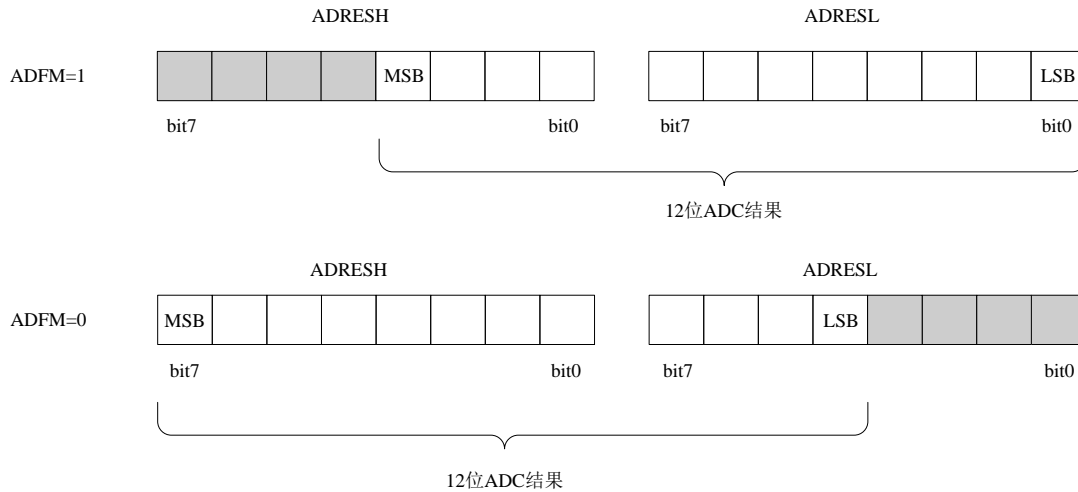


图 9.4 ADC 转换结果格式示意

### 9.1.11 阈值比较

ADCMPOH 寄存器为 ADC 结果比较阈值，ADCON3 寄存器的 ADCMPEN 位控制比较功能使能，ADCMPOP 位控制比较极性，ADCMPO 指示比较结果。

AD 可以在每次转换完成时进行比较。比较结果会一直保持，直到下次转换完成被更新。ADCMPEN 或 ADON 的清零可以关闭比较功能或 AD 模块，同时可以清零 ADCMPO。进入睡眠不会清零 ADCMPO。

在每次比较完成时可以产生故障刹车事件，由 ADCON3 寄存器的 ADFBEN 控制。

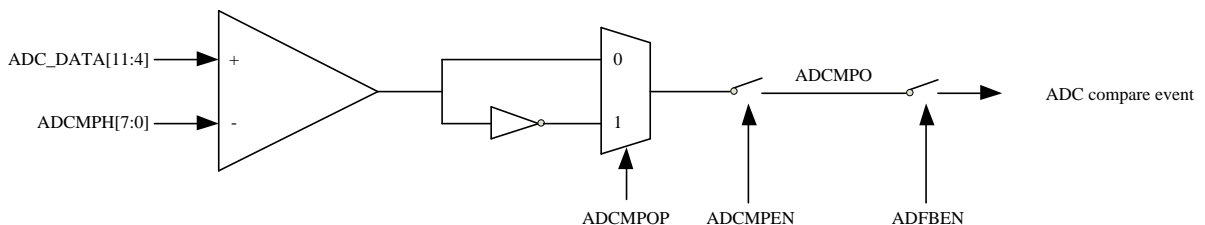


图 9.5 ADC 阈值比较功能框图

注意：

1. ADCMPO 是内部信号，对软件不可见，但软件可以通过 ADCMPIF 位间接判断 ADCMPO 的值。ADC 控制模块在转换结束时刻判断 ADCMPO 的值，当 ADCMPO 为 1 时，ADCMPIF 被置 1。

## 9.2 ADC 的工作原理

### 9.2.1 启动转换

要使能 ADC 模块，必须将 ADCON0 寄存器的 ADON 位置 1。

若 ADEX=0 时，将 ADCON0 寄存器的 GO/DONE 位置 1 将启动 AD 转换。

若 ADEX=1 时，需要外部触发信号才能启动，并且硬件置位 GO/DONE 位，程序置位 ADGO 无效。

注意：

1. 不应在打开 ADC 的那条指令中将 GO/DONE 位置 1。请参见第 9.2.7 节“A/D 转换步骤”
2. 不应在启动 ADC 转换后或等待外部触发时更改 AD 配置。
3. 置位 ADGO 后需要等待一个系统周期才可读回 ADGO 标志。

### 9.2.2 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

### 9.2.3 终止转换

当 ADEX=0，ADC 处于软件触发状态，如果转换必须在完成前被终止，可用软件将 GO/DONE 清零。那么 ADC 将不会更新这次转换数据。当 ADEX=1，ADC 处于硬件触发状态，如果需要终止转换，则需要将 ADON 置 0，关闭 ADC 的使能开关。

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。

### 9.2.4 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 F<sub>RC</sub> 选项。

ADC 需要等待  $4 \cdot T_{AD}$  后才开始转换。这允许软件在设置 ADGO 后，执行一个 SLEEP 指令置 MCU 于 SLEEP 模式，从而降低 ADC 转换期间的系统噪声。通过配置 ADC 时钟为 F<sub>RC</sub>，可进一步降低系统噪声。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 F<sub>RC</sub>，执行一条 SLEEP 指令将使当前转换强制中止，ADC 模块被直接关闭，尽管 ADON 位保持置 1 状态。

## 9.2.5A/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 启动转换或等待硬件触发；
7. 等待一个系统周期才可回读 GO/DONE；
8. 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）
9. 读取 ADC 结果；
10. 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 TRISA           ;
BSR TRISA,0             ;Set RA0 to input
BANKSEL ANSEL           ;
BSR ANSEL,0             ;Set RA0 to analog
BANKSEL ADCON0          ;
LDWI B'10000001'        ;Right justify,
STR ADCON0              ; Vdd Vref, AN0, On
LCALL StableTime      ; ADC stable time
LCALL SampleTime     ;Acquisition 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 9.6，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 18.10 的  $T_{ACQ}$  时间要求，否则 ADC 精度、线性度将不能保证；

### 9.2.6A/D 采集时间要求

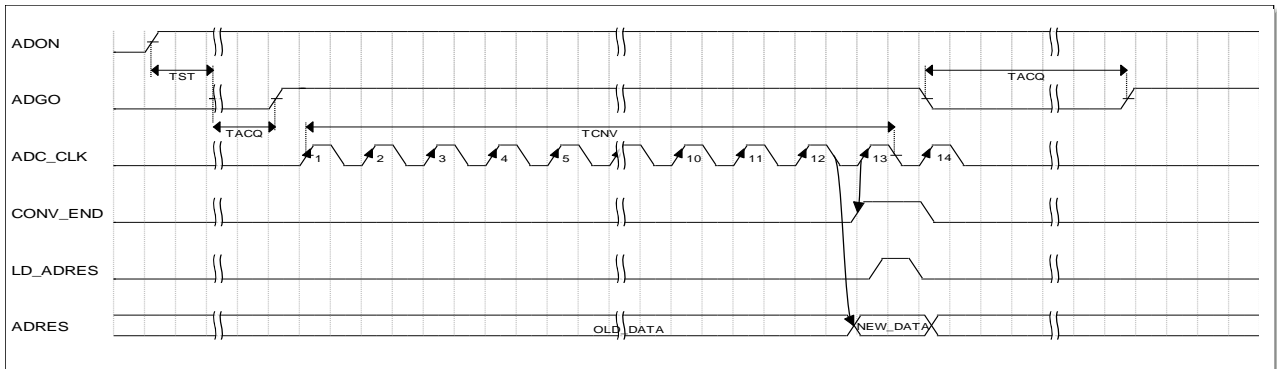
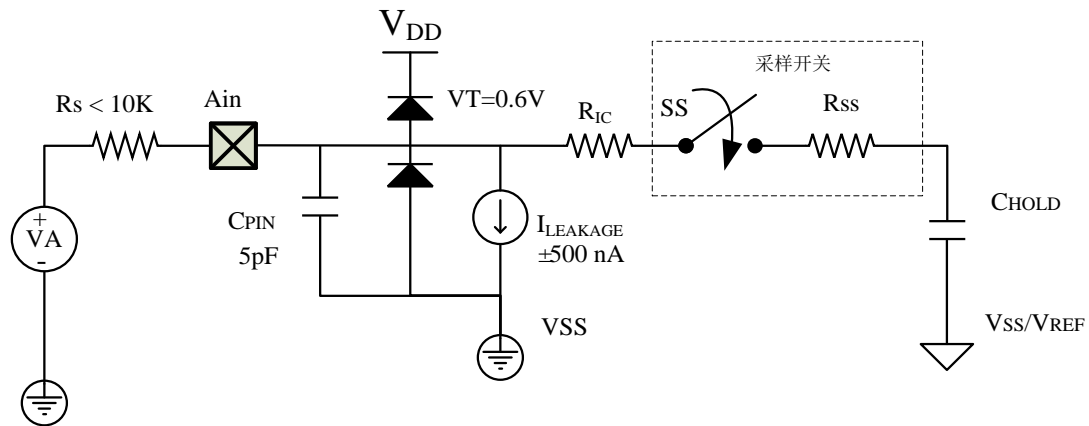


图 9.6 ADC 软件触发转换时序图

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 9.7。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 9.7。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。



- 图注：
- CPIN = 输入电容
  - VT = 门限电压
  - ILEAKAGE = 结点漏电流
  - RIC = 互联电阻
  - SS = 采样开关
  - CHOLD = 采样保持电容

图 9.7 模拟输入模型

### 9.3 与 ADC 相关寄存器汇总

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
ADRESL	0x111	A/D 结果寄存器的低字节								xxxx xxxx
ADRESH	0x112	A/D 结果寄存器的高字节								xxxx xxxx
ADCON0	0x113	—	CHS<3:0>				ADEX	GO/DONE	ADON	-000 0000
ADCON1	0x114	ADFM	ADCS<2:0>			ADNREF<1:0>		ADPREF<1:0>		0000 0000
ADCON2	0x115	ADINTREF<1:0>		ETGTYP<1:0>		ADDLY.8	ETGSEL<2:0>			0000 0000
ADDLY	0x188	ADDLY<7:0> / LEBPRL<7:0>								0000 0000
ADCON3	0x186	ADFBN	ADCMPOP	ADCMPEM	—	LEBADT	—			000- 0---
ADCMPL	0x187	ADCMPL<7:0>								0000 0000
LEBCON	0x185	LEBEN	LEBCH		—	EDGS	—			000- 0---
VRP5VCAL	0x97	VRP5VCAL<7:0>								xxxx xxxx
VR2VCAL	0x108	VR2VCAL<7:0>								xxxx xxxx
VR3VCAL	0x11F	VR3VCAL<7:0>								xxxx xxxx

#### 9.3.1 ADRESL, 地址 0x111

Bit	7	6	5	4	3	2	1	0
Name	ADRESL<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	ADRESL	ADC 结果寄存器的低字节 ADFM=0 时, ADRESL[7:4]为 12 位转换结果的低 4 位, 其余为 0。 ADFM=1 时, ADRESL[7:0]为 12 位转换结果的低 8 位。

#### 9.3.2 ADRESH, 地址 0x112

Bit	7	6	5	4	3	2	1	0
Name	ADRESH<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	ADRESH	ADC 结果寄存器的高字节 ADFM=0 时, ADRESH[7:0]为 12 位转换结果的高 8 位。 ADFM=1 时, ADRESH[3:0]为 12 位转换结果的高 4 位, 其余为 0。

### 9.3.3ADCON0, 地址 0x113

Bit	7	6:3	2	1	0
Name	—	CHS<3:0>	ADEX	GO/DONE	ADON
Reset	—	0	0	0	0
Type	RO.0	RW	RW	RW	RW

Bit	Name	Function
7	N/A	保留:读为 0
6:3	CHS	<p>模拟通道选择位</p> <p>0000 = AN0                      0001 = AN1                      0010 = AN2                      0011 = AN3                      0100 = AN4                      0101 = AN5                      0110 = AN6                      0111 = AN7</p> <p>1xxx = 内部参考电压通道, 具体电压值根据 ADINTREF 进行选择</p>
2	ADEX	<p>ADC 触发信号类型选择</p> <p>该位决定启动 ADC 的触发条件</p> <p>0 = 当软件设定 GO/DONE 位, 启动 AD 转换</p> <p>1 = 需要外部触发信号触发才可启动 AD 转换, 触发事件置位 GO/DONE 位。                      外部触发信号条件由寄存器 ETGSEL&lt;2:0&gt;和 ETGTYP&lt;1:0&gt;决定。</p>
1	GO/DONE	<p>AD 转换状态位 (硬件触发事件直接置位)</p> <p>将该位置 1 可启动 A/D 转换周期。当 A/D 转换完成以后, 该位由硬件自动清零。</p> <p>0 = A/D 转换完成/未进行。</p> <p>1 = A/D 转换正在进行或硬件触发延时正在计数。</p>
0	ADON	<p>ADC 使能位</p> <p>0 = ADC 被禁止且不消耗工作电流</p> <p>1 = ADC 被使能</p>



### 9.3.4ADCON1, 地址 0x114

Bit	7	6	5	4	3	2	1	0
Name	ADFM	ADCS<2:0>			ADNREF<1:0>		ADPREF<1:0>	
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	ADFM	<p>ADC 结果格式选择位</p> <p>1 = 右对齐。装入转换结果时, ADRESH 的高 4 位被设置为 0。</p> <p>0 = 左对齐。装入转换结果时, ADRESL 的低 4 位被设置为 0。</p>
6:4	ADCS	<p>ADC 转换时钟选择位</p> <p>000 = <math>F_{Osc}/2</math></p> <p>001 = <math>F_{Osc}/8</math></p> <p>010 = <math>F_{Osc}/32</math></p> <p>011 = <math>F_{Osc}</math></p> <p>100 = <math>F_{Osc}/4</math></p> <p>101 = <math>F_{Osc}/16</math></p> <p>110 = <math>F_{Osc}/64</math></p> <p>111 = <math>F_{RC}</math> (由专用 RC 振荡器提供时钟)</p>
3:2	ADNREF	<p>ADC 负参考电压配置位 (使用 PA5 连接外部参考电压或外部电容)</p> <p>00 = Int Vref (内部参考电压)</p> <p>01 = GND</p> <p>10 = Int Vref + Ext Cap (内部参考电压 + 外部电容)</p> <p>11 = Ext Vref (外部参考电压)</p>
1:0	ADPREF	<p>ADC 正参考电压配置位 (使用 PA4 连接外部参考电压或外部电容)</p> <p>00 = Int Vref (内部参考电压)</p> <p>01 = <math>V_{DD}</math></p> <p>10 = Int Vref + Ext Cap (内部参考电压 + 外部电容)</p> <p>11 = Ext Vref (外部参考电压)</p>

### 9.3.5ADCON2, 地址 0x115

Bit	7	6	5	4	3	2	1	0
Name	ADINTREF<1:0>		ETGTYP<1:0>		ADDLY.8	ETGSEL<2:0>		
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	ADINTREF	<p>ADC 内部参考电压配置位</p> <p>00 = 0.5V</p> <p>01 = 2V</p> <p>10 = 3V</p> <p>11 = float (悬空)</p>
5:4	ETGTYP	<p>外部触发信号类型选择</p> <p>当 ADEX 置 1, 该位决定响应外部触发的类型</p> <p>00 = PWM 或 ADC_ETR 脚的下降沿</p> <p>01 = PWM 或 ADC_ETR 脚的上升沿</p>
3	ADDLY.8 /LEBPR9	<p>ADC 外部触发延时计数器阈值 第 8 位</p> <p>详见 ADDLY 寄存器描述</p>
2:0	ETGSEL	<p>外部触发源选择</p> <p>当 ADEX 为 1, 该位选择外部触发 ADC 的来源</p> <p>000 = P1A0</p> <p>001 = P1A0N</p> <p>010 = P1B</p> <p>011 = P1C</p> <p>100 = P1D</p> <p>101 = ADC_ETR</p> <p>其他位无效</p>

### 9.3.6 LEBCON 寄存器，地址 0x185

Bit	7	6	5	4	3	2	1	0
Name	LEBEN	LEBCH[1:0]		—	EDGS	—	—	—
Reset	0	0	0	—	0	—	—	—
Type	RW	RW	RW	RO-0	RW	RO-0	RO-0	RO-0

Bit	Name	Function
7	LEBEN	前沿消隐使能位（仅当 <b>ADGO=0</b> 时可进行切换，否则 <b>ADC</b> 工作异常） 1 = 使能 0 = 禁止
6:5	LEBCH	前沿消隐通道选择 00 = P1A0 01 = P1B 10 = P1C 11 = P1D
4	N/A	保留位，读 0
3	EDGS	PWM 消隐沿选择 0 = PWM 上升沿 1 = PWM 下降沿
2:0	N/A	保留位，读 0

### 9.3.7 ADCON3，地址 0x186

Bit	7	6	5	4	3	2	1	0
Name	ADFBEN	ADCMPOP	ADCM PEN	—	LEBADT	—	—	—
Reset	0	0	0	—	0	—	—	—
Type	RW	RW	RW	RO.0	RW	RO.0	RO.0	RO.0

Bit	Name	Function
7	ADFBEN	ADC 比较结果响应故障刹车使能 0 = 禁止 1 = ADC 触发故障刹车功能使能
6	ADCMPOP	ADC 比较器输出极性选择位 0 = 若 ADC 结果的高八位大于或等于 ADCMPH[7:0]，ADCMPO 为 1 1 = 若 ADC 结果的高八位小于 ADCMPH[7:0]，ADCMPO 为 1
5	ADCM PEN	ADC 结果比较使能位 0 = ADC 结果比较功能关闭 1 = ADC 结果比较功能打开
4	N/A	保留位，读 0
3	LEBADT	前沿消隐周期结束后，ADC 触发使能 1 = 触发 ADC 转换 0 = 不触发 ADC 转换
2:0	N/A	保留，读为 0

### 9.3.8 ADCMPH, 地址 0x187

Bit	7	6	5	4	3	2	1	0
Name	ADCMPH<7:0>							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	ADCMPH	ADC 比较阈值 仅 8 位, 用于 ADC 结果高 8 位比较。

### 9.3.9 ADDLY/LEBPR, 地址 0x188

Bit	7	6	5	4	3	2	1	0
Name	ADDLY<7:0>							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	ADDLY	ADC 外部触发启动延时计数器阈值低位 该 8 位寄存器与 ADCON2.7 组成 9 位计数器, 用于在外部触发启动 ADC 之前加入一段延迟。延迟计数器结束再开始 ADC 转换 外部延迟时间 = (ADDLY+6)/F <sub>ADC</sub> 注, 该延时仅当 ADEX 置 1 时有效。如果启用 PWM 输出触发 ADC 功能, 在 PWM 运行过程中不得更改 ADDLY 计数值。同时复用为前沿消隐计数阈值

### 9.3.10 VRP5VCAL, 地址 0x97

Bit	7	6	5	4	3	2	1	0
Name	VRP5VCAL<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	VRP5VCAL	ADC内部0.5V参考电压校准位

### 9.3.11 VR2VCAL, 地址 0x108

Bit	7	6	5	4	3	2	1	0
Name	VR2VCAL<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	VR2VCAL	ADC内部2V参考电压校准位

### 9.3.12 VR3VCAL, 地址 0x11F

Bit	7	6	5	4	3	2	1	0
Name	VR3VCAL<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	VR3VCAL	ADC内部3V参考电压校准位

## 10. 定时器 0

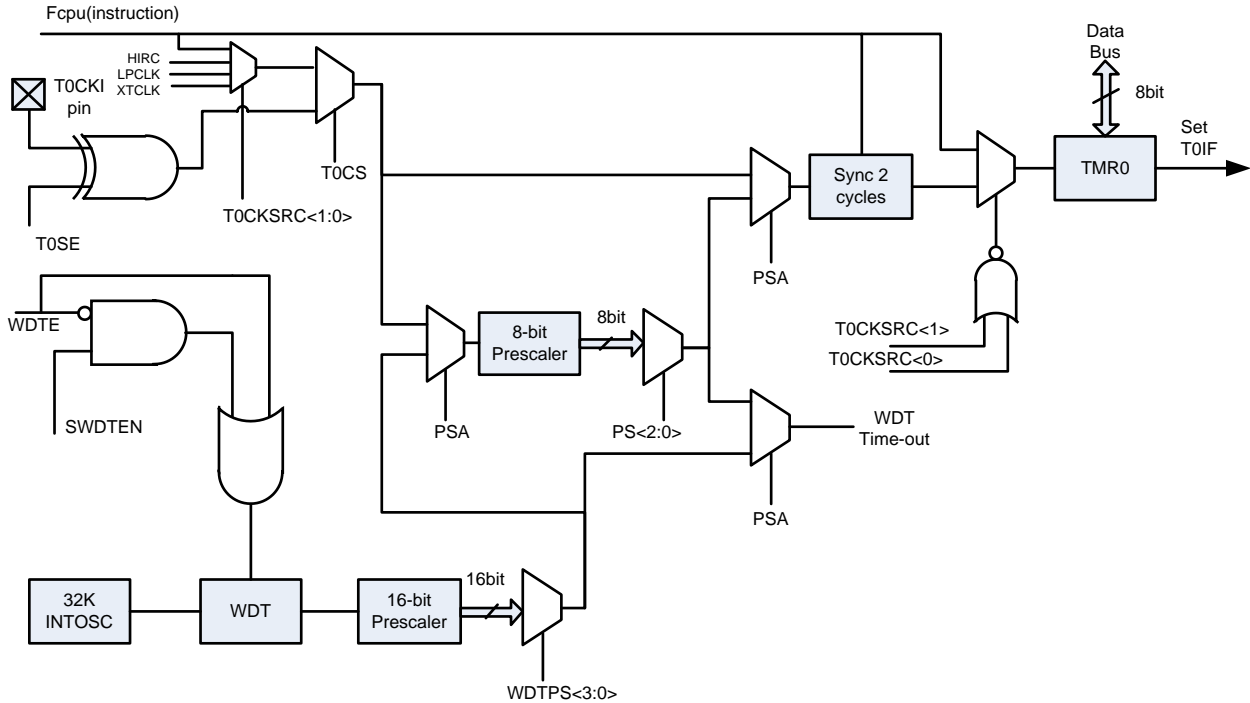


图 10.1 看门狗和定时器 0 框图

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（T0CKI）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，其计数时钟源由 T0CKSRC 可选时钟源，即每一时钟周期递增一次。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。

### 10.1. Timer0 定时器模式

该模式下，定时器 0 在每个时钟周期（时钟源可选）加 1（不带预分频）。软件可以清零 OPTION 寄存器里的 T0CS 位以进入定时器模式。当软件对 TMRO 进行写操作时，定时器在写后面 2 个周期内不会递增。

### 10.1.1. Timer0 的时钟源

Timer0 的时钟源由寄存器位 T0CKSRC 控制，默认选择指令时钟。

- 指令时钟
- HIRC 内部高速时钟
- LP 晶体时钟
- XT 晶体时钟

在改变 Timer0 的时钟配置之前，建议先把 T0ON 清 0，以防止时钟切换过程中产生的毛刺对 Timer0 的影响。

在非 SLEEP 模式下且 Timer0 被使能（T0ON=1）的情况下，被选择的时钟源将自动打开；

**注意：**

1. 选择 LP 作为 T0 时钟源时，配置选项 FOSC 必须是 LP 或者 INTOSCIO 模式，否则，LP 晶体电路将不打开；
2. 同理，选择 XT 作为 T0 时钟源时，配置选项 FOSC 必须是 XT 或者 INTOSCIO 模式，否则，XT 晶体电路将不打开。

在 SLEEP 模式下，如果要 Timer0 继续保持计数，则需要把 T0CKRUN 置 1，且时钟源不能选择指令时钟，因为在 SLEEP 模式下，指令时钟是关闭的。

### 10.1.2. TMR0 寄存器的读写

当 Timer0 时钟源跟 CPU 时钟不一样时（如配置为 LP 或者 XT 模式），建议软件对 TMR0 的启动读写操作时先把 T0ON 清 0，以避免读错或写错。

## 10.2. Timer0 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 T0SE 位决定。软件可以将 OPTION 寄存器里的 T0CS 位置 1 以进入计数器模式。

### 10.2.1. 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。

### 10.2.1.1. 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
STR 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'11111000' ;Mask TMR0 select and prescaler bits
ANDWR OPTION_REG,W
IORWI b'00000011' ;Set prescale to 1:16
STR OPTION_REG
```

### 10.2.2. 定时器 0 中断

当寄存器 TMR0（定时器 0 计数值）与 PR0 匹配时将置起 TOIF 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，除非 TOCKRUN 为 1 且其时钟源不是指令时钟时。

### 10.2.3. 用外部时钟驱动定时器 0

在计数其模式下，TOCKI 管脚输入和 Timer0 寄存器之间的同步，是由 Timer0 的时钟源（时钟源可选）产生的非交叠时钟来采样实现的，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。

### 10.2.4. 睡眠模式下的状态

当 TOCKRUN=1 而且 Timer0 的时钟源不是选择指令时钟时，MCU 进入睡眠后，Timer0 以保持在运行状态，由 TOCKSRC 所选择的时钟源不会关闭。否则，Timer0 将停止计数，维持睡眠前的计数值。

## 10.3. Timer0 的 PWM 控制信号模式

当 TOUCH 模块用于硬件检测模式下，Timer0 将被用于产生 PWM 的控制信号，其中 TODUTY 和 PR0 将用于设定 PWM 波形的占空比和周期。



### 10.4. 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	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
OPTION	0x81	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
T0CON0	0x1F	—	—	—	—	T0ON	T0CKRUN	T0CKSRC[1:0]		---- 1000
TRISA	0x85	TRISA[7:0], PORTA 方向控制								1111 1111
T0DUTY	0x18C	T0DUTY[7:0],								0000 0000
PR0	0x18D	PR0[7:0],								1111 1111

#### 10.4.1. 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	<b>/PAPU:</b> PORTA上拉使能位 1 = 禁止PORTA上拉 0 = PORTA上拉由各端口锁存器值使能
6	INTEDG	<b>INTEDG:</b> 中断沿选择位 1 = 在PC1/INT引脚的上升沿中断 0 = 在PC1/INT引脚的下降沿中断
5	T0CS	<b>T0CS:</b> Timer0时钟源选择位 1 = PA2/T0CKI引脚的跳变 0 = 根据T0CKSRC位确定
4	T0SE	<b>T0SE:</b> Timer0 时钟源边沿选择位 1 = 在PA2/T0CKI引脚的下降沿递增 0 = 在PA2/T0CKI引脚的上升沿递增
3	PSA	<b>PSA:</b> 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	<b>PS&lt;2:0&gt;:</b> 预分频比选择位 位值      TIMER0 分频比      WDT 分频比 000          1 : 2                  1 : 1 001          1 : 4                  1 : 2 010          1 : 8                  1 : 4 011          1 : 16                  1 : 8 100          1 : 32                  1 : 16 101          1 : 64                  1 : 32 110          1 : 128                  1 : 64 111          1 : 256                  1 : 128
1	PS1	
0	PS0	

### 10.4.2. 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	Timer 0 计数结果寄存器

### 10.4.3. T0CON0, 地址 0x1F

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	T0ON	T0CKRUN	T0CKSRC[1:0]	
Reset	—	—	—	—	1	0	0	0
Type	RO-0	RO-0	RO-0	RO-0	RW	RW	RW	RW

Bit	Name	Function
7:4	N/A	保留位, 读 0
3	T0ON	定时器 0 使能位 1 = 使能 (default 值为 1, 保持向前兼容) 0 = 禁止
2	T0CKRUN	当 T0 时钟不是选择指令时钟时, 睡眠状态 T0CK 的运行控制位 1 = T0CK 睡眠时保持工作 0 = T0CK 睡眠时停止工作
1:0	T0CKSRC	T0 时钟源选择 00 = 指令时钟 01 = HIRC 10 = LP 晶体时钟, 只有当 FOSC 配置为 LP 或 INTOSCIO 模式时才有意义 11 = XT 晶体时钟, 只有当 FOSC 配置为 XT 或 INTOSCIO 模式时才有意义

#### 10.4.4. T0DUTY, 地址 0x18C

Bit	7	6	5	4	3	2	1	0
Name	T0DUTY							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	T0DUTY	当 TKMODE = 1,开启 TOUCH 硬件检测时, TIMER0 用于产生 TOUCH 的 PWM 控制信号, T0DUTY 是该 PWM 的占空比寄存器

#### 10.4.5. PR0, 地址 0x18D

Bit	7	6	5	4	3	2	1	0
Name	PR0							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PR0	当 TKMODE = 1,开启 TOUCH 硬件检测时, TIMER0 用于产生 TOUCH 的 PWM 控制信号, PR0 是该 PWM 的周期寄存器

## 11. 定时器 1

定时器 1 为 12 位定时器，包含以下功能：

- 12 位计数寄存器
- 12 位周期寄存器
- TMR1 值等同 PR1 时产生中断
- 1:1, 1:4, 1:16 预分频比(与 Timer2 共用同一分频器)

Timer1 的时钟源由寄存器位 T1CKSRC 控制，默认选择指令时钟。

- 指令时钟
- HIRC 内部高速时钟
- LP 晶体时钟
- XT 晶体时钟

**注意：**当 TKMODE 为 1 时，Timer1 时钟源固定选择为 Timer0 占空比匹配脉冲。

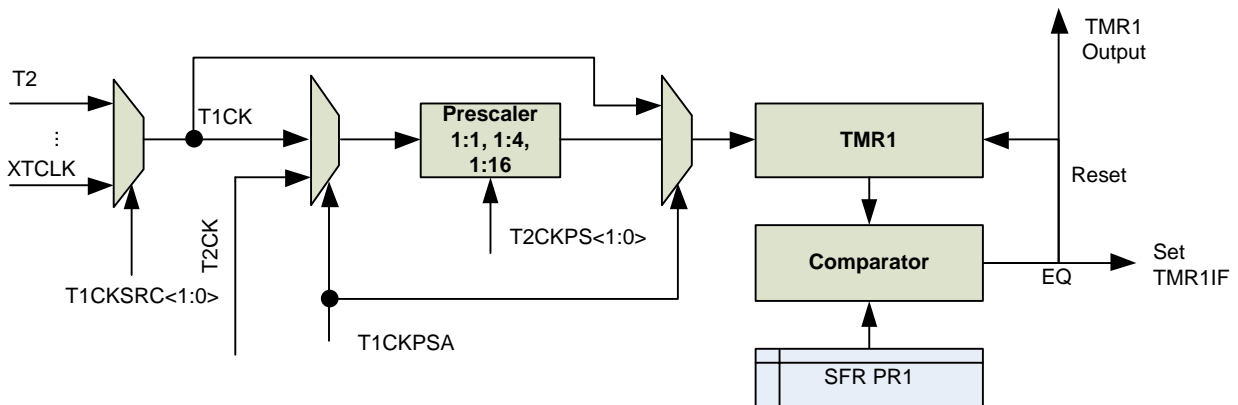


图 11.1 定时器 1 框图

### 11.1. Timer1 工作原理

Timer1 模块的时钟源可选，默认输入是系统指令时钟（FOSC/2）。该时钟被用于递增 TMR1 寄存器。

{TMR1H, TMR1L} 和 {PR1H, PR1L} 的值被不断比较以确定何时匹配。{TMR1H, TMR1L} 将从 00h 开始递增直到与 {PR1H, PR1L} 的值相同。匹配时将发生以下两种情况：

{TMR1H, TMR1L} 在下一递增周期复位为 00h；

Timer1/PR1 比较器的匹配输出用于将 PIR1 寄存器的 TMR1IF 中断标志置 1。

{TMR1H, TMR1L} 和 {PR1H, PR1L} 都是可读写寄存器。在复位时，他们的值分别是 0 和 0xFFFF。将 T1CON0 寄存器中的 T1ON 位置 1 可打开 Timer1，反之将 T1ON 位清零关闭 Timer1。

## 11.2. Timer1 计数值的读写

具体读写操作步骤请参照 12.3 小节的 Timer2 计数值的读写。

注意：Timer1 的写操作会影响 Timer2 预分频（见 12.4 小节），为避免影响 Timer2 计数，针对 Timer1 的清 0，可以通过以下方式规避：

读 TMR1L，读 TMR1H，得到当前计数值 x；

对 PR1H:PR1L 写 x，即 PR1= TMR1；

对 PR1H:PR1L 写其他值，如 Timer1 的目标匹配值，此时，TMR1H:L 将自动清 0；

## 11.3. 与 Timer1 相关寄存器汇总

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
PR1L	0x116	PR1L[7:0]								1111 1111
PR1H	0x117	—	—	—	—	PR1H[3:0]				---- 1111
TMR1L	0x118	TMR1L[7:0]								0000 0000
TMR1H	0x119	—	—	—	—	TMR1H[3:0]				---- 0000
T1CON0	0x11A	—	—	—	T1CKPSA	T1CKRUN	T1ON	T1CKSRC[1:0]		---0 0000

### 11.3.1. PR1L 寄存器，地址 0x116,0x117

PR1L，地址 0x116

Bit	7	6	5	4	3	2	1	0
Name	PR1L[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	PR1L[7:0]	PR1 周期寄存器低 8 位

PR1H，地址 0x117

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PR1H[3:0]			
Reset	—	—	—	—	1	1	1	1
Type	RO.0	RO.0	RO.0	RO.0	RW	RW	RW	RW

Bit	Name	Function
3:0	PR1H[3:0]	PR1 周期寄存器高 4 位

### 11.3.2. TMR1 寄存器，地址 0x118, 0x119

TMR1L, 地址 0x118

Bit	7	6	5	4	3	2	1	0
Name	TMR1L[7:0]							
Reset	0000 0000							
Type	RW							

TMR1H, 地址 0x119

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

Bit	Name	Function
11:0	TMR1	Timer 1 计数结果寄存器

### 11.3.3. T1CON0 寄存器，地址 0x11A

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	T1CKPSA	T1CKRUN	T1ON	T1CKSRC[1:0]	
Reset	—	—	—	0	0	0	0	0
Type	RO-0	RO-0	RO-0	RW	RW	RW	RW	RW

Bit	Name	Function
7:5	N/A	未实现，读 0
4	T1CKPSA	Timer1 预分频器分配位 1 = 预分频器分配给Timer1，此时，不管T2ON为何值，Timer2将启动计数 0 = 预分频器分配给Time2模块
3	T1CKRUN	当 T1 时钟不是选择指令时钟时，睡眠状态 T1CK 的运行控制位 1 = T1CK 睡眠时保持工作 0 = T1CK 睡眠时停止工作
2	T1ON	Timer1使能位 1 = Timer1打开 0 = Timer1 关闭
1:0	T1CKSRC	T1 时钟源选择 00 = 指令时钟 01 = HIRC 10 = LP 晶体时钟，只有当 FOSC 配置为 LP 或 INTOSCIO 模式时才有意义 11 = XT 晶体时钟，只有当 FOSC 配置为 XT 或 INTOSCIO 模式时才有意义

## 12. 定时器 2

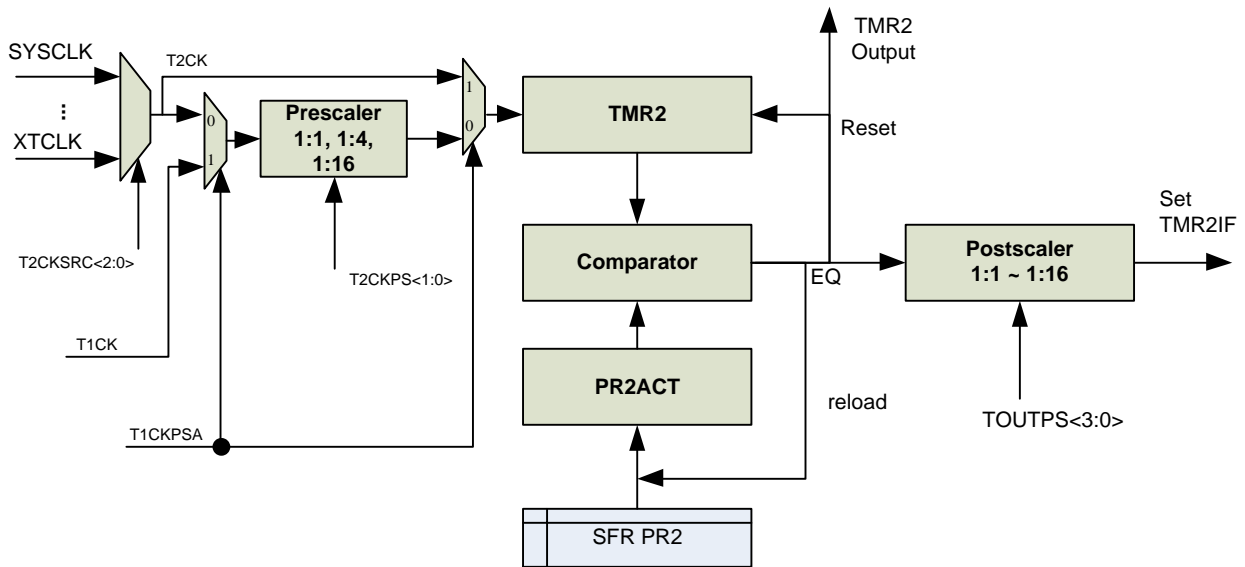


图 12.1 定时器 2 框图

定时器 2 为 16 位定时器，包含以下功能：

- 16 位计数寄存器
- 16 位周期寄存器，双缓冲
- TMR2 值等同 PR2 时产生中断
- 1:1, 1:4, 1:16 预分频比（Timer1 复用同一预分频器）
- 1:1-1:16 后分频比
- 时钟源可选：系统时钟或者内部 32MHz 时钟（由晶体的 2 倍频得到）或 LIRC

## 12.1. Timer2 工作原理

在非 PWM 模式下，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事件：

- 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 0xFFFF。

将 T2CON0 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 T2CON0 寄存器的 T2CKPS 位控制。

Timer2 后分频器由 T2CON0 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 T2CON0
- 任何 reset 动作

注：

1. 写 T2CON0 并不会清零 TMR2 寄存器；
2. Timer2 的时钟源由 T2CKSRC 控制，当 T2CKSRC≠000 时，如果 T2CKRUN 设置为 1，则 Timer2 的时钟在睡眠状态下继续运行。

## 12.2. 关于 PR2 的更新

Timer2 的周期寄存器 PR2 具有双缓冲结构，在模块内部分别是 PR2ACT，PR2。PR2ACT 是活动寄存器，也就是 TMR2 要比较的寄存器。正常情况下，只有在 Timer2 发生匹配事件时 PR2ACT 才会被更新为 PR2 的内容。

软件也可以在写完 PR2 寄存器对 PR2ACT 进行更新，而无需等待匹配事件，方法是向 PR2U 位写 1。

**注意：PR2ACT 对软件不可见。**



### 12.3. Timer2 计数值的读写

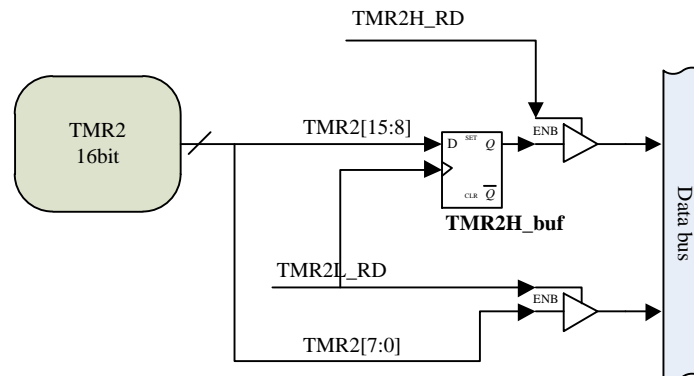


图 12.2 Timer2 计数值的读操作原理框图

Timer2 是 16bit 的定时器，由于内部数据总线是 8bit 的限制，软件读 Timer2 的计数值需要分开两次读，计数值的低 8 位 TMR2L 可直接访问，高 8 位有一个内部缓存 TMR2H\_buf，该缓存在软件读 TMR2L 的时刻被更新。这种机制保证了即使 Timer2 在计数期间，软件总是可以读到一个完整的 16bit 计数值，避免了诸如在两次读间隔内，Timer2 发生了溢出等情况。

综上所述，读操作应按照下面的顺序进行：

- 读 TMR2L;
- 读 TMR2H;

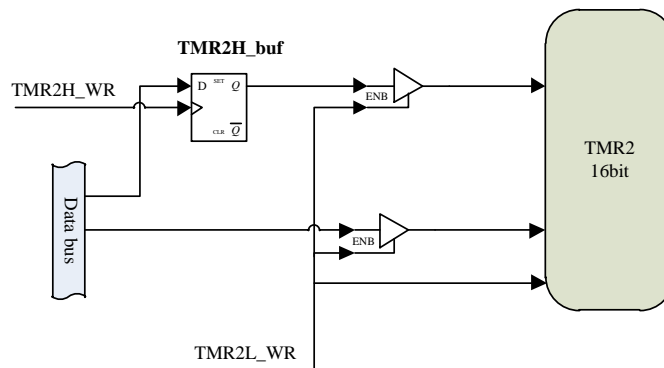


图 12.3 Timer2 计数值的写操作原理框图

和读操作相似，软件对 TMR2H 寄存器的写并不会立即更新内部计数值，而是先写到缓存 TMR2H\_buf 中，当软件写 TMR2L 时，硬件自动把缓存的高 8 位一起更新到计数值当中。

写顺序：

- 写 TMR2H;
- 写 TMR2L;

注意：当 Timer2 工作在异步时钟时，建议先清除 TMR2ON 位以停止计数，然后至少等 1 个计数时钟后再发起对 TMR2 的读。

另外，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。如果寄存器正在进行递增计数，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 TMR2H:TMR2L 这对寄存器中产生不可预测的值。

## 12.4. Timer2 预分频清零

当软件执行以下动作时，预分频自动清 0，且 Timer2 将停止计数一个指令时钟。

- 写 TMR2H;
- 写 TMR2L;
- 写 TMR1H;
- 写 TMR1L;
- 写 T2CON;

## 12.5. Timer2 时钟源

Timer2 支持 6 种不同的时钟源：

- 指令时钟
- 系统时钟
- HIRC 的 2 倍频
- 晶体/外部时钟 2 倍频（只有当 FOSC 配置为 LP/XT 或 EC 模式时才有效）
- HIRC
- LIRC
- LP 晶体时钟（只有当 FOSC 配置为 INTOSCIO，或 LP 模式时才有效）
- XT 晶体时钟（只有当 FOSC 配置为 INTOSCIO，或 XT 模式时才有效）

## 12.6. 睡眠下的工作

以下条件成立时，Timer2 将在 SLEEP 模式下保持计数：

- a) T2ON=1, T2CKRUN=1, T2CKSRC 选择非指令时钟并且合法（见寄存器描述）；或者
- b) T1CKPSA=1 且 T2CKSRC 所选时钟与 T1CKSRC 所选时钟一致，不管 T2ON 为何值，Timer2 都将工作；

## 12.7. 与 Timer2 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
TMR2L	0x11	Timer2 计数值寄存器, 低 8 位								0000 0000
TMR2H	0x13	Timer2 计数值寄存器, 高 8 位								0000 0000
INTCON	0x0B/8B	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000
PIE1	0x8C	EEIE	CKMIE	LVDIE	ACMPIE	TMR1IE	OSFIE	TMR2IE	ADCIE	0000 0000
PIR1	0x0C	EEIF	CKMIF	LVDIF	ACMPIF	TMR1IF	OSFIF	TMR2IF	ADCIF	0000 0000
MSCON0	0x1B	BGRBOE	LVROE	ROMLPE	CLKOS	SLVREN	CKMAVG	CKCNT1	T2CKRUN	0001 0000
PR2L	0x91	Timer2 周期寄存器, 低 8 位								1111 1111
PR2H	0x92	Timer2 周期寄存器, 高 8 位								1111 1111
T2CON0	0x12	PR2U	TOUTPS[3:0]			TMR2ON	T2CKPS[1:0]			0000 0000
T2CON1	0x9E	—			P1OS	P1BZM	T2CKSRC[2:0]			---0 0000

### 12.7.1. PR2 寄存器, 地址 0x91, 0x92

见 [PR2L 寄存器, 地址 0x91](#), [PR2H 寄存器, 地址 0x92](#)。

### 12.7.2. TMR2 寄存器, 地址 0x11, 0x13

TMR2L, 地址 0x11

Bit	7	6	5	4	3	2	1	0
Name	TMR2L[7:0]							
Reset	0000 0000							
Type	RW							

TMR2H, 地址 0x13

Bit	7	6	5	4	3	2	1	0
Name	TMR2H[7:0]							
Reset	0000 0000							
Type	RW							

Bit	Name	Function
15:0	TMR2	Timer 2 计数结果寄存器

### 12.7.3. T2CON0 寄存器，地址 0x12

Bit	7	6	5	4	3	2	1	0
Name	PR2U	TOUTPS[3:0]				TMR2ON	T2CKPS[1:0]	
Reset	0	0000				0	00	
Type	WO-1	RW	RW	RW	RW	RW	RW	

Bit	Name	Function
7	PR2U	PR2、P1xDTy 寄存器的软件更新控制位，只写 写 1:把 PR2/P1xDTy 缓冲值分别更新到 PR2 寄存器和 P1xDTy_ACT 写 0:无意义
6:3	TOUTPS	<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 0010 = 1:3 后分频比 0011 = 1:4 后分频比 0100 = 1:5 后分频比 0101 = 1:6 后分频比 0110 = 1:7 后分频比 0111 = 1:8 后分频比 1000 = 1:9 后分频比 1001 = 1:10 后分频比 1010 = 1:11 后分频比 1011 = 1:12 后分频比 1100 = 1:13 后分频比 1101 = 1:14 后分频比 1110 = 1:15 后分频比 1111 = 1:16 后分频比
2	TMR2ON	<b>TMR2ON</b> : Timer2 On bit 打开定时器2 1 = Timer2打开 0 = Timer2 关闭 PWM1 单脉冲模式下，该位自动清 0
1:0	T2CKPS/T1CKPS	<b>T2CKPS&lt;1:0&gt;/T1CKPS&lt;1:0&gt;</b> : Timer2/Timer1 Clock Prescale Select bits 定时器2/定时器1 驱动时钟预分频比选择 00 = Prescaler is 1 01 = Prescaler is 4 1x = Prescaler is 16

12.7.4. T2CON1 寄存器，地址 0x9E

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	P1OS	P1BZM	T2CKSRC[2:0]		
Reset	—	—	—	0	0	0	0	0
Type	RO-0	RO-0	RO-0	RW	RW	RW	RW	RW

Bit	Name	Function
7:5	N/A	保留位，读 0
4	P1OS	PWM1 单脉冲模式选择 0 = 连续模式 1 = 单脉冲模式
3	P1BZM	PWM1 蜂鸣器模式选择 0 = PWM 模式 1 = buzzer 模式
2:0	T2CKSRC	Timer2 时钟源选择 000 = 指令时钟 001 = 系统时钟 010 = HIRC 的 2 倍频 011 = 晶体/外部时钟 2 倍频（只有当 FOSC 配置为 LP/XT 或 EC 模式时才有效） 100 = HIRC 101 = LIRC 110 = LP 晶体时钟（只有当 FOSC 配置为 INTOSCIO，或 LP 模式时才有效） 111 = XT 晶体时钟（只有当 FOSC 配置为 INTOSCIO，或 XT 模式时才有效）

## 13. PWM 模块

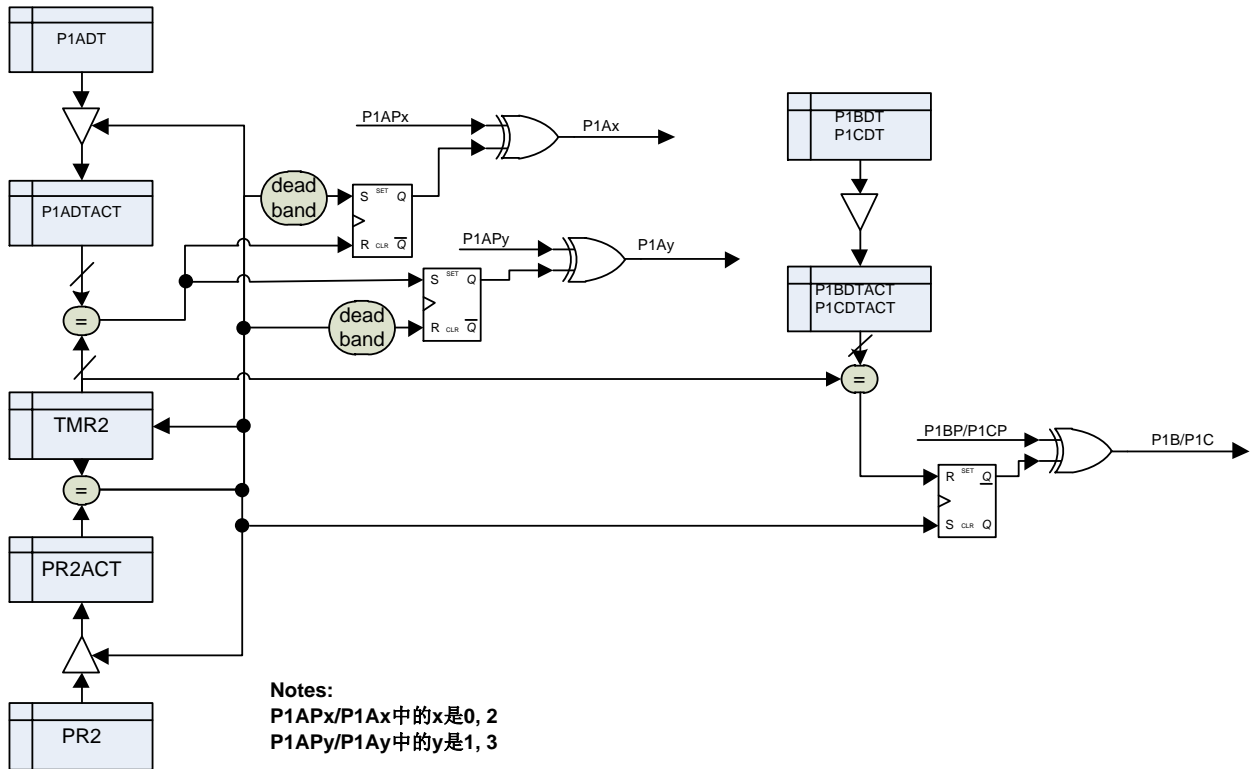


图 13.1 PWM 结构框图

PWM 支持以下特性：

- 16bit 的分辨率
- 周期和占空比匹配双缓冲设计
- 1 路带死区控制的 PWM 输出：P1A
- 4 路独立占空比的 PWM 输出：P1A, P1B, P1C, P1D
- 每路 PWM 输出极性可独立设置
- 故障刹车以及自动重启

### 13.1. 周期

PWM 周期由 Timer2 的 PR2 寄存器指定。用公式 13.1 可计算 PWM 周期。

$$\text{PWM 周期} = (\text{PR2} + 1) * T_{\text{T2CK}} * (\text{TMR2 预分频值})$$

公式 13.1

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
- P1A0, [P1A0], P1B, P1C, P1D 置 1（4 路 PWM 都是高有效的情况下）
- 内部的周期寄存器 PR2ACT 和占空比寄存器 P1xDTACT 被更新

## 13.2. 占空比

通过对以下几个寄存器写入 16 位值可指定 PWM 占空比：

P1xDTL(x= A, B, C, D)

P1xDTH(x= A, B, C, D)

其中，P1xDTH 保存的是 4 路 PWM 占空比寄存器的 8 位 MSb，P1xDTL 则是低 8 位。由于内部的双缓冲设计，占空比寄存器在任何时候被写入，它在软件新占空比时 PWM 的不会产生毛刺起重要作用。

公式 13.2 用于计算 PWM 脉宽。

公式 13.3 用于计算 PWM 占空比。

脉冲宽度 =  $P1xDT * T_{T2CK} * (TMR2 \text{ 预分频值})$

公式 13.2

占空比 =  $P1xDT \div (PR2+1)$

公式 13.3

## 13.3. 时钟源选择

PWM1 使用的时基定时器为 Timer2，Timer2 的时钟源有以下选择：

- 系统时钟
- 指令时钟（即系统时钟的 2 分频或 4 分频）
- HIRC 的 2 倍频
- 外部时钟 2 倍频（只有当 FOSC 配置为 EC 模式时才有效）
- HIRC
- LIRC

## 13.4. 睡眠下 PWM 状态

当 T2CKRUN=1 而且 Timer2 的时钟源不是选择指令时钟时，MCU 进入睡眠后，PWM 可以保持在运行状态，由 T2CKSRC 所选择的时钟源不会关闭。否则，Timer2 将停止计数，而 PWM 各管脚电平保留在执行完 SLEEP 指令后的状态。

### 13.5. P1A 的死区时间

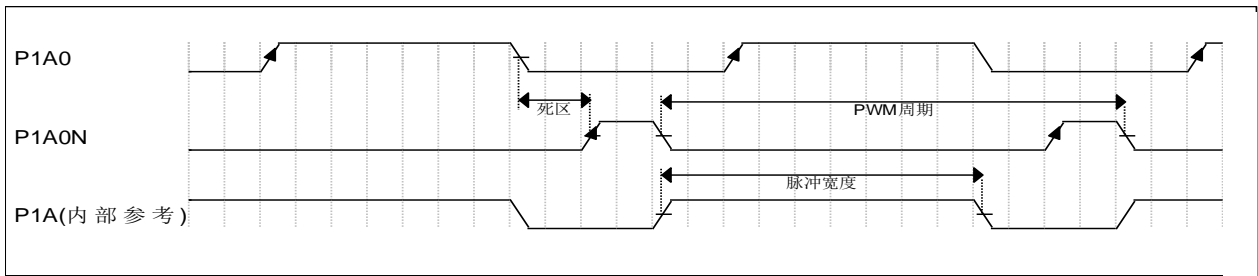


图 13.2 死区 PWM 示意图

有1路PWM带互补输出，P1Ax。其中P1A0/[P1A1]/[P1A2]定义为正输出，[P1A0N]/[P1A1N]/[P1A2N]为互补输出。P1A的PWM带有死区插入功能，其死区时间由P1DC[6:0]控制。死区定时器以Timer2时钟作为计数时钟源。

注意：{P1A0,[ P1A0N]}，{[P1A1], [P1A1N]}，{[P1A2], [P1A2N]}这 3 对互补输出共享同一死区设置。

### 13.6. 故障刹车

PWM1 模块支持故障刹车模式，它会在发生外部刹车事件时禁止 PWM 输出，同时 Timer2 及预分频器处于复位状态。刹车模式会将 PWM 输出引脚置于预定状态，该模块用于防止发生故障条件时 PWM 损坏应用。

使用 P1BR0 寄存器的 P1BKS 位可选择故障源，故障事件可以是以下几种：

- BKIN 管脚为低电平
- BKIN 管脚为高电平
- LVDW 被 LVD 模块置 1
- LVDW=1 或 BKIN=0
- LVDW=1 或 BKIN=1
- ADC 阈值比较为 1

刹车状态由 P1BR0 寄存器的 P1BEVT 位指示。如果该位为 0，PWM1 引脚正常工作。如果该位为 1，PWM1 输出处于关闭状态。

注意：刹车源 LVDW 可以选择去抖，由 LVDDEB 寄存器位决定。



### 13.6.1. 刹车状态

发生故障时，故障下的 PWM1 管脚电平状态由寄存器 P1xSS 位选择，有以下几种：

- PWM1 置于有效电平
- PWM1 置于无效电平
- PWM1 关闭，处于悬空高阻状态
- TMR2 定时器停止计数，预分频计数器复位
- TMR2ON 位不受影响

有效电平由 P1POL 各寄存器位决定。

### 13.6.2. 故障清除

故障刹车条件是基于电平的信号，而非基于边沿的信号。只要故障条件有效，就故障状态一直保持，软件不能清除；只有当相关故障输入或 LVD 事件消除了，P1BEVT 才可能被清 0。

### 13.6.3. 自动重启

PWM1 可配置为在故障条件被清除时自动重启 PWM1 信号。通过将 P1CON 寄存器中的 P1AUE 位置 1 使能自动重启。

自动重启使能时，只要故障条件有效，P1BEVT 位就保持置 1。当故障条件被清除时，P1BEVT 标志位将被硬件清零，TMR2 恢复计数，在下一次计数溢出时，实际控制信号 P1BEVT 被清除，PWM1 恢复正常输出。

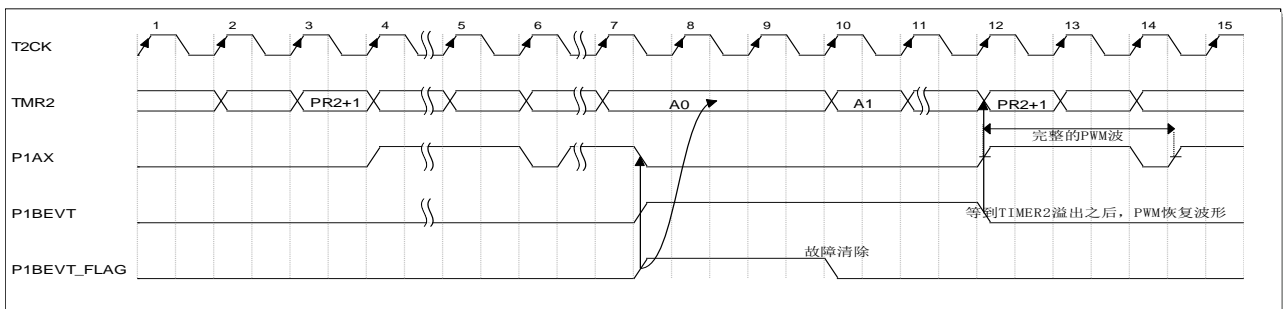


图 13.3 PWM 的自动重启时序图

### 13.7. 关于周期和占空比寄存器的更新

在 Timer2 已经开启的情况下，周期和各占空比寄存器的更新需要 TMR2 和 PR2 的匹配事件，如果用户不想等待，可以通过写 PR2U 位来立即更新。

当 TMR2ON 为 0 时，软件对 PR2, {P1xDTH, P1xDTL}寄存器的写会马上更新到对应的工作寄存器，此时 PWM 输出保持旧值，不会因为 PR2 或 P1xDT 寄存器变化而变化。

注意：工作寄存器 xxxACT 对软件不可见，软件只能读 PR2 和 P1xDTL, P1xDTH。

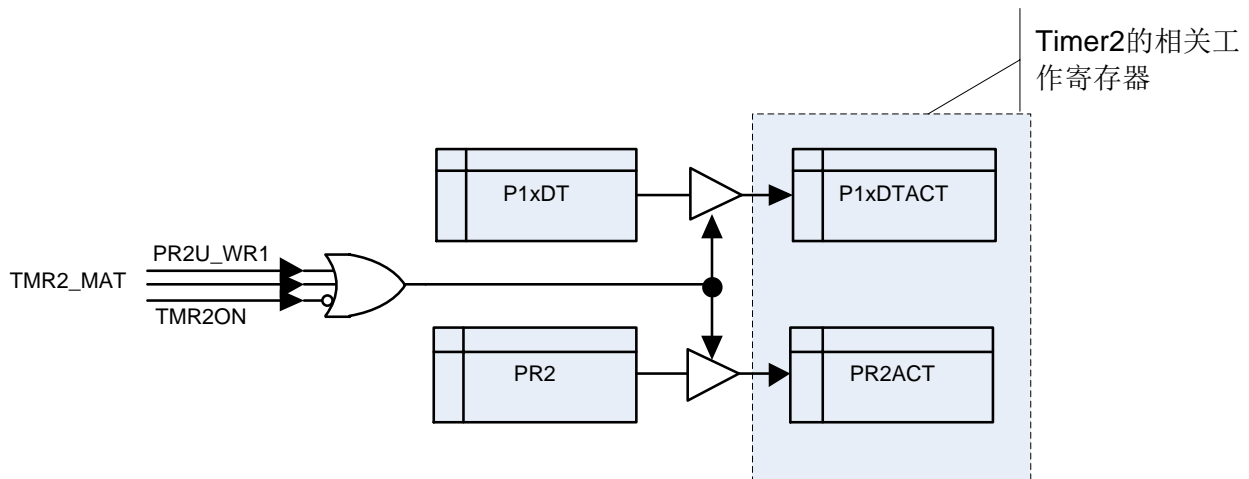


图 13.4 T2 工作寄存器的更新

虽然周期和占空比的双缓冲在很大程度上保证 PWM 输出不会产生毛刺，但如果软件非常靠近 TMR2 匹配时刻去写此寄存器，特别是在 T2 时钟频率比系统时钟频率快的情况下，则有可能出现不可预料的情况，导致工作寄存器组的值不是期望值，见下图 13.5。

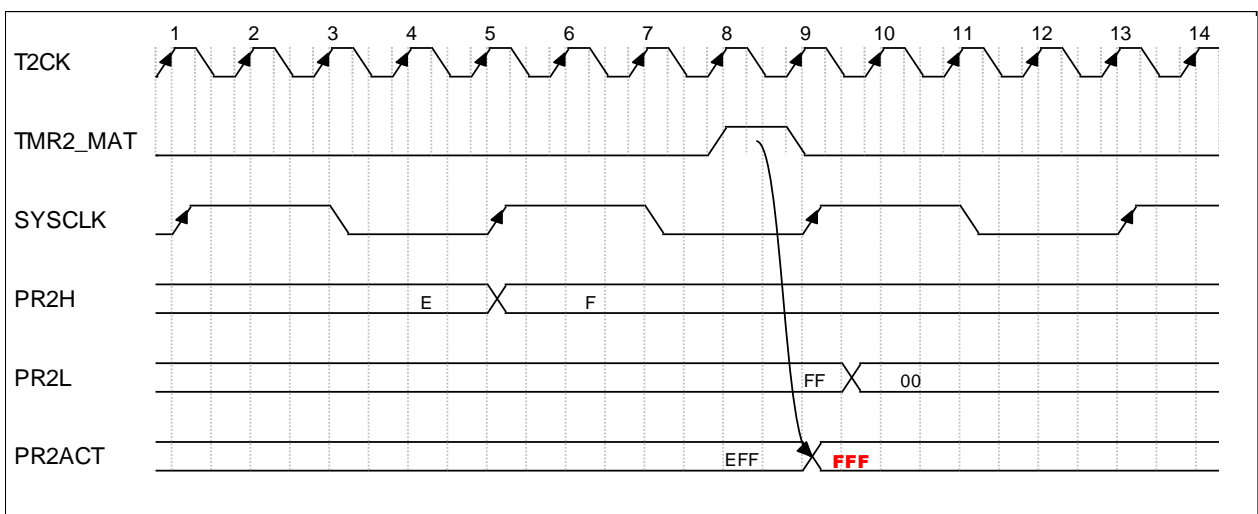


图 13.5 PR2ACT 被更新为意外值 FFF（期望值是 F00）

所以强烈建议更新 PR2 和 xxxDT 只在 TMR2 匹配中断里面做。

### 13.8. 蜂鸣器模式 (Buzzer)

当 T2CON1.P1BZM 设置为 1 时, PWM1 模式将作为蜂鸣器输出, 在这种模式下, 占空比设置寄存器不起作用, P1Ax, P1B, P1C 将输出周期为 $(2*(PR2+1)*T_{T2CK} *TMR2 \text{ 预分频})$ 的方波。

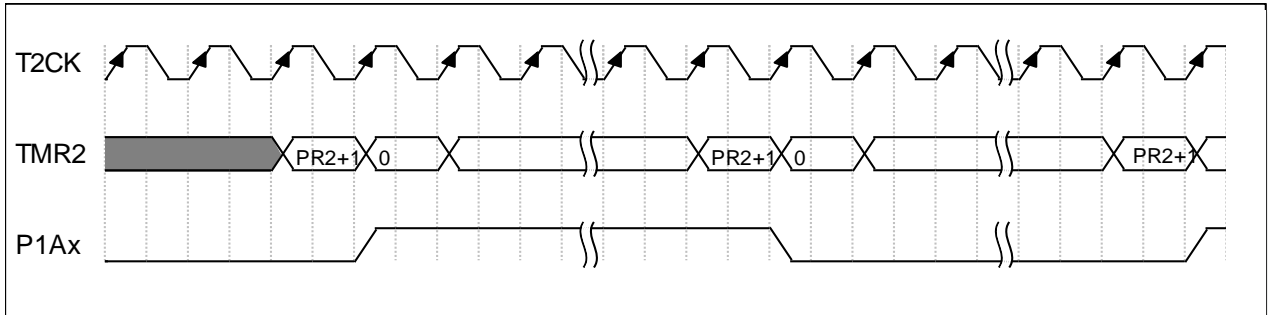


图 13.6 蜂鸣器模式输出

### 13.9. 单脉冲输出

当 P1OS 设置为 1 时, PWM1 就处于单脉冲输出模式。在该模式下, 首次 TMR2 和 (PR2+1) 的匹配将使 P1Ax, P1B, P1C 输出 PWM 脉冲, 在下次匹配时, TMR2ON 被硬件关闭, 且相应管脚的 P1XOE 使能也将被关闭。

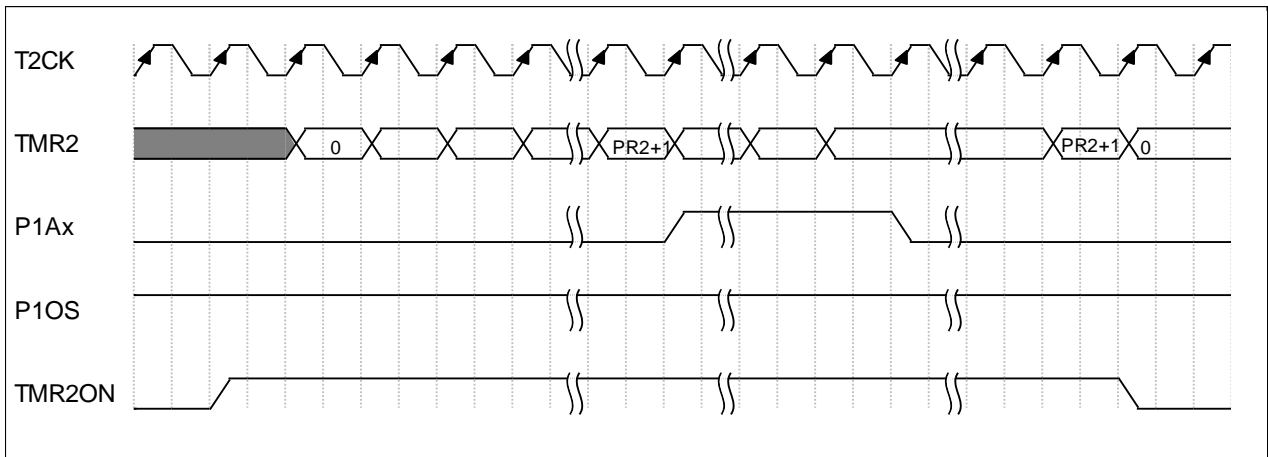


图 13.7 单脉冲模式输出

### 13.10. PWM 输出重映射

P1A、P1B、P1C 和 P1D 这 4 路 PWM 可以分别映射到不同的 I/O, 由寄存器 P1OE0 和 P1OE1 控制, 它们的复位值为 0, 具体映射的 I/O 可以在本文档第 1 章的各脚位图及脚位描述找到。

这个特性可以同时允许在两个 I/O 输出同一路 PWM。

### 13.11. P1C、P1D 的第 2 功能输出

除了正常的 PWM 波形输出，P1C 的重映射管脚（PB0），可以配置成输出 P1C、P1D 之间的同或，异或结果，由寄存器 P1CF2E 和 P1CF2 控制。P1D 的重映射管脚（PA3、PA2）可以配置成输出 P1B、P1C 之间的同或，异或结果，由寄存器 P1DF2E 和 P1DF2 控制。

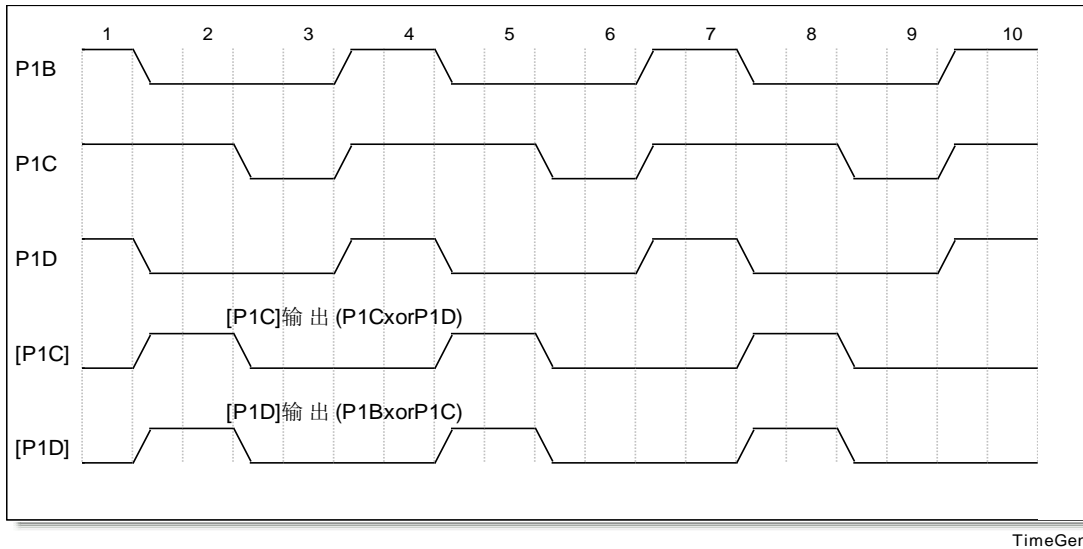


图 13.8 P1B 和 P1C 的第 2 功能时序示例

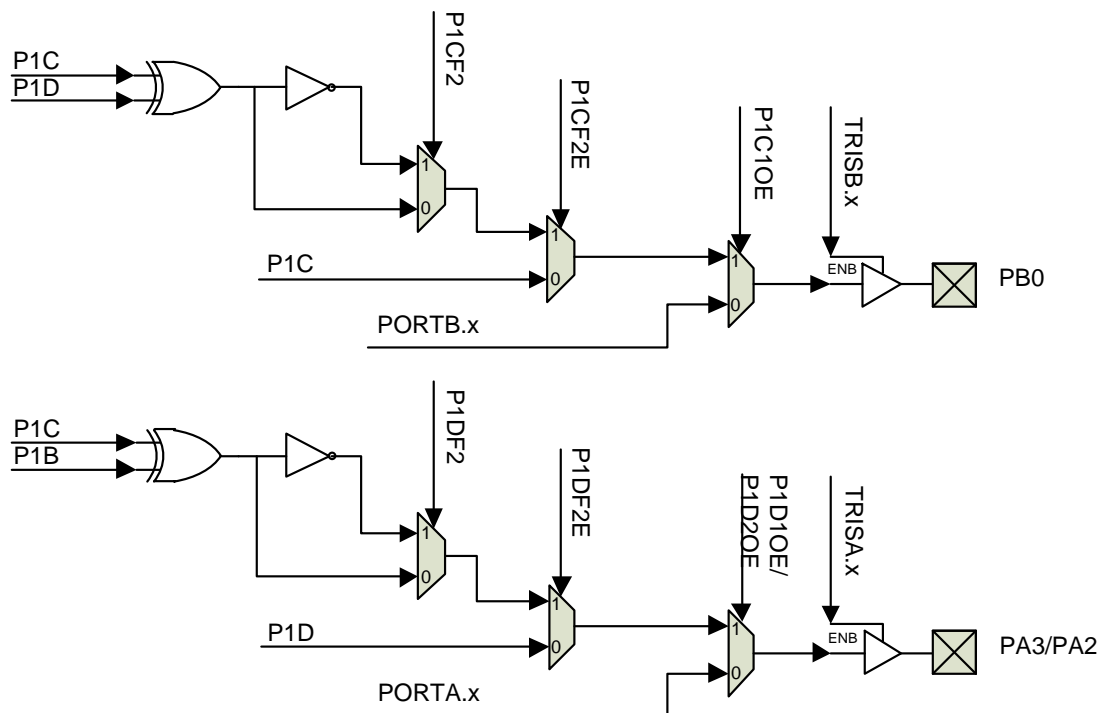


图 13.9 P1C 和 P1D 的第 2 功能输出

### 13.12. 与 PWM1 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值	
P1ADTL	0xE	P1A 占空比低 8 位								0000 0000	
P1BDTL	0xF	P1B 占空比低 8 位								0000 0000	
P1CDTL	0x10	P1C 占空比低 8 位								0000 0000	
P1DDTL	0x08	P1D 占空比低 8 位								0000 0000	
TMR2L	0x11	Timer2 计数器低 8 位								0000 0000	
TMR2H	0x13	Timer2 计数器高 8 位								0000 0000	
T2CON0	0x12	PR2U	TOUTPS				TMR2ON	T2CKPS			0000 0000
T2CON1	0x9E	—			P1OS	P1BZM	T2CKSRC			---0 0000	
P1ADTH	0x14	P1A 占空比高 8 位								0000 0000	
P1BDTH	0x15	P1B 占空比高 8 位								0000 0000	
P1CDTH	0x1A	P1C 占空比高 8 位								0000 0000	
P1DDTH	0x09	P1D 占空比高 8 位								0000 0000	
P1CON	0x16	P1AUE	P1DC								0000 0000
P1BR0	0x17	P1BEVT	P1BKS			P1BSS		P1ASS			0000 0000
P1BR1	0x19	P1D2SS		P1DSS		P1C2SS		P1CSS		0000 0000	
P1OE2	0x11B	P1D2OE	P1D1OE	P1D0OE	—	—	P1C1OE	P1B1OE	—	000- -00-	
P1OE	0x90	P1C0OE	P1B0OE	P1A2NOE	P1A2OE	P1A1NOE	P1A1OE	P1A0NOE	P1A0OE	0000 0000	
PR2L	0x91	PR2[7:0]								1111 1111	
PR2H	0x92	PR2[15:8]								1111 1111	
P1POL	0x99	P1C0P	P1B0P	P1A2NP	P1A2P	P1A1NP	P1A1P	P1A0NP	P1A0P	0000 0000	
P1POL2	0x109	P1D2P	P1D1P	P1D0P	—	—	P1C1P	P1B1P	—	000- -00-	
P1AUX	0x1E	—	—	P1B2SS[1:0]		P1CF2E	P1CF2	P1DF2E	P1DF2	--00 0000	

#### 13.12.1. P1ADTL 寄存器，地址 0x0E

Bit	7	6	5	4	3	2	1	0
Name	P1ADTL[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	P1ADTL	P1A 占空比寄存器低 8 位

### 13.12.2. P1BDTL 寄存器，地址 0x0F

Bit	7	6	5	4	3	2	1	0
Name	P1BDTL[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	P1BDTL	P1B 占空比寄存器低 8 位

### 13.12.3. P1CDTL 寄存器，地址 0x10

Bit	7	6	5	4	3	2	1	0
Name	P1CDTL[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	P1CDTL	P1C 占空比寄存器低 8 位

### 13.12.4. P1DDTL 寄存器，地址 0x8

Bit	7	6	5	4	3	2	1	0
Name	P1DDTL[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
3:0	P1DDTL	P1D 占空比寄存器低 8 位

### 13.12.5. TMR2L 寄存器，地址 0x11

Bit	7	6	5	4	3	2	1	0
Name	TMR2L[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TMR2L	Timer2 计数器低 8 位

**13.12.6. TMR2H 寄存器，地址 0x13**

Bit	7	6	5	4	3	2	1	0
Name	TMR2H[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
3:0	TMR2H	Timer2 计数器高 8 位

**13.12.7. T2CON0 寄存器，地址 0x12**

见 [T2CON0 寄存器，地址 0x12](#)。

**13.12.8. P1ADTH 寄存器，地址 0x14**

Bit	7	6	5	4	3	2	1	0
Name	P1ADTH[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	P1ADTH	P1A 占空比寄存器高 8 位

**13.12.9. P1BDTH 寄存器，地址 0x15**

Bit	7	6	5	4	3	2	1	0
Name	P1BDTH[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	P1BDTH	P1B 占空比寄存器高 8 位

### 13.12.10. P1CDTH 寄存器，地址 0x1A

Bit	7	6	5	4	3	2	1	0
Name	P1CDTH[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	P1CDTH	P1C 占空比寄存器高 8 位

### 13.12.11. P1DDTH 寄存器，地址 0x9

Bit	7	6	5	4	3	2	1	0
Name	P1DDTH[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	P1DDTH	P1D 占空比寄存器高 8 位

### 13.12.12. P1CON 寄存器，地址 0x16

Bit	7	6	5	4	3	2	1	0
Name	P1AUE	PDC[6:0]						
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	P1AUE	PWM1 重启使能位 1 = 故障刹车时，P1BEVT 位在退出关闭事件时自动清零，PWM1 自动重启 0 = 故障刹车时，必须用软件将 P1BEVT 清零以重启 PWM1
6:0	P1DC	PWM1 死区时间设置 P1DCn = 预定 PWM 信号应转变为有效与 PWM 信号实际转为有效之间的 T2CK 周期数



### 13.12.13. P1BR0 寄存器，地址 0x17

Bit	7	6	5	4	3	2	1	0
Name	P1BEVT	P1BKS[2:0]			P1BSS[1:0]		P1ASS[1:0]	
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	P1BEVT	<p>PWM1 故障事件状态位</p> <p>1 = 发生了故障事件</p> <p>0 = 未发生故障事件，PWM1 输出正常工作</p>
6:4	P1BKS	<p>PWM1 故障源选择位</p> <p>000 = 禁止故障刹车功能</p> <p>001 = BKIN 为低电平</p> <p>010 = BKIN 为高电平</p> <p>011 = LVDW=1</p> <p>100 = BKIN 为低电平或 LVDW=1</p> <p>101 = BKIN 为高电平或 LVDW=1</p> <p>110 = ADC 阈值比较</p> <p>111 = 保留（禁止故障刹车）</p>
3:2	P1BSS	<p>故障下，P1B0 管脚的状态（电平极性由 P1POLx 寄存器确定）</p> <p>00 = 高阻</p> <p>01 = 无效电平</p> <p>1x = 有效电平</p>
1:0	P1ASS	<p>故障下，P1A 管脚的状态（电平极性由 P1POLx 寄存器确定）</p> <p>00 = 高阻</p> <p>01 = 无效电平</p> <p>1x = 有效电平</p>

13.12.14. P1BR1 寄存器，地址 0x19

Bit	7	6	5	4	3	2	1	0
Name	P1D2SS[1:0]		P1DSS[1:0]		P1C2SS[1:0]		P1CSS[1:0]	
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	P1D2SS	故障下，[P1D1]/[P1D2]管脚的状态，只有当 P1D1OE/P1D2OE 为 1 时才有效 00 = 高阻 01 = 输出 0 1x = 输出 1
5:4	P1DSS	故障下，P1D0 管脚的状态（电平极性由 P1POLx 寄存器确定） 00 = 高阻 01 = 无效电平 1x = 有效电平
3:2	P1C2SS	故障下，[P1C1]管脚的状态，只有当 P1C1OE 为 1 时才有效 00 = 高阻 01 = 输出 0 1x = 输出 1
1:0	P1CSS	故障下，P1C0 管脚的状态（电平极性由 P1POLx 寄存器确定） 00 = 高阻 01 = 无效电平 1x = 有效电平

**13.12.15. P1OE2 寄存器，地址 0x11B**

Bit	7	6	5	4	3	2	1	0
Name	P1D2OE	P1D1OE	P1D0OE	—	—	P1C1OE	P1B1OE	—
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RO.0	RO.0	RW	RW	RO.0

Bit	Name	Function
7	P1D2OE	P1D2 输出使能，高有效 1 = P1D2 输出到相关管脚 0 = P1D2 不输出到相关管脚
6	P1D1OE	P1D1 输出使能，高有效 1 = P1D1 输出到相关管脚 0 = P1D1 不输出到相关管脚
5	P1D0OE	P1D0 输出使能，高有效 1 = P1D0 输出到相关管脚 0 = P1D0 不输出到相关管脚
4:3	N/A	保留位，读 0
2	P1C1OE	P1C1 输出使能，高有效 1 = P1C1 输出到相关管脚 0 = P1C1 不输出到相关管脚
1	P1B1OE	P1B1 输出使能，高有效 1 = P1B1 输出到相关管脚 0 = P1B1 不输出到相关管脚
0	N/A	保留位，读 0

**13.12.16. P1OE 寄存器，地址 0x90**

Bit	7	6	5	4	3	2	1	0
Name	P1C0OE	P1B0OE	P1A2NOE	P1A2OE	P1A1NOE	P1A1OE	P1A0NOE	P1A0OE
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	P1C0OE	P1C0 输出使能，高有效 1 = P1C0 输出到相关管脚 0 = P1C0 不输出到相关管脚
6	P1B0OE	P1B0 输出使能，高有效 1 = P1B0 输出到相关管脚 0 = P1B0 不输出到相关管脚
5	P1A2NOE	P1A2N 输出使能，高有效 1 = P1A2N 输出到相关管脚 0 = P1A2N 不输出到相关管脚
4	P1A2OE	P1A2 输出使能，高有效 1 = P1A2 输出到相关管脚 0 = P1A2 不输出到相关管脚
3	P1A1NOE	P1A1N 输出使能，高有效 1 = P1A1N 输出到相关管脚 0 = P1A1N 不输出到相关管脚
2	P1A1E	P1A1 输出使能，高有效 1 = P1A1 输出到相关管脚 0 = P1A1 不输出到相关管脚
1	P1A0NOE	P1A0N 输出使能，高有效 1 = P1A0N 输出到相关管脚 0 = P1A0N 不输出到相关管脚
0	P1A0OE	P1A0 输出使能，高有效 1 = P1A0 输出到相关管脚 0 = P1A0 不输出到相关管脚

**13.12.17. PR2L 寄存器，地址 0x91**

Bit	7	6	5	4	3	2	1	0
Name	PR2L[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	PR2L	PR2 周期寄存器低 8 位

**13.12.18. PR2H 寄存器，地址 0x92**

Bit	7	6	5	4	3	2	1	0
Name	PR2H[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	PR2H	PR2 周期寄存器高 8 位

**13.12.19. P1POL 寄存器，地址 0x99**

Bit	7	6	5	4	3	2	1	0
Name	P1C0P	P1B0P	P1A2NP	P1A2P	P1A1NP	P1A1P	P1A0NP	P1A0P
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	P1C0P	P1C0 输出极性设置 1 = P1C0 低电平有效 0 = P1C0 高电平有效
6	P1B0P	P1B0 输出极性设置 1 = P1B0 低电平有效 0 = P1B0 高电平有效
5	P1A2NP	P1A2N 输出极性设置 1 = P1A2N 低电平有效 0 = P1A2N 高电平有效
4	P1A2P	P1A2 输出极性设置 1 = P1A2 低电平有效 0 = P1A2 高电平有效
3	P1A1NP	P1A1N 输出极性设置 1 = P1A1N 低电平有效 0 = P1A1N 高电平有效
2	P1A1P	P1A1 输出极性设置 1 = P1A1 低电平有效 0 = P1A1 高电平有效
1	P1A0NP	P1A0N 输出极性设置 1 = P1A0N 低电平有效 0 = P1A0N 高电平有效
0	P1A0P	P1A0 输出极性设置 1 = P1A0 低电平有效 0 = P1A0 高电平有效

**13.12.20. P1POL2 寄存器，地址 0x109**

Bit	7	6	5	4	3	2	1	0
Name	P1D2P	P1D1P	P1D0P	—	—	P1C1P	P1B1P	—
Reset	0	0	0	—	—	0	0	—
Type	RW	RW	RW	RO.0	RO.0	RW	RW	RO.0

Bit	Name	Function
7	P1D2P	<p>[P1D2]输出极性设置 当输出 P1D 波形 1 = [P1D2]低电平有效 0 = [P1D2]高电平有效</p> <p>当输出第二功能波形 1 = 第二功能反向波形 0 = 第二功能波形</p>
6	P1D1P	<p>[P1D1]输出极性设置 1 = [P1D1]低电平有效 0 = [P1D1]高电平有效</p> <p>当输出 P1D 第二功能波形 1 = 第二功能反向波形 0 = 第二功能波形</p>
5	P1D0P	<p>P1D 输出极性设置 1 = P1D 低电平有效 0 = P1D 高电平有效</p>
4:3	N/A	保留位，读 0
2	P1C1P	<p>[P1C1]输出极性设置 1 = [P1C1]低电平有效 0 = [P1C1]高电平有效</p> <p>当输出 P1C 第二功能波形 1 = 第二功能反向波形 0 = 第二功能波形</p>
1	P1B1P	<p>[P1B1]输出极性设置 1 = [P1B1]低电平有效 0 = [P1B1]高电平有效</p>
0	N/A	保留位，读 0

13.12.21. P1AUX 寄存器，地址 0x1E

Bit	7	6	5	4	3	2	1	0
Name	—	—	P1B2SS[1:0]		P1CF2E	P1CF2	P1DF2E	P1DF2
Reset	—	—	0	0	0	0	0	0
Type	RO-0	RO-0	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	N/A	保留位，读 0
5:4	P1B2SS	故障下，[P1B1]管脚的状态，只有当 P1B1OE 为 1 时才有效 00 = 高阻 01 = 输出 0 1x = 输出 1
3	P1CF2E	当 P1C1OE 为 1 时，[P1C1]管脚第 2 功能使能 1 = 输出 P1C 和 P1D 的同或，或者异或 0 = 输出 P1C
2	P1CF2	[P1C1]管脚第 2 功能选择 1 = 输出 P1C 和 P1D 的同或 0 = 输出 P1C 和 P1D 的异或
1	P1DF2E	当 P1D1OE/P1D2OE 为 1 时，[P1D1]/[P1D2]管脚第 2 功能使能 1 = 输出 P1B 和 P1C 的同或，或者异或 0 = 输出 P1D
0	P1DF2	[P1D1]/[P1D2]管脚第 2 功能选择 1 = 输出 P1B 和 P1C 的同或 0 = 输出 P1B 和 P1C 的异或

## 14. I/O 端口

本芯片共包含 18 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。

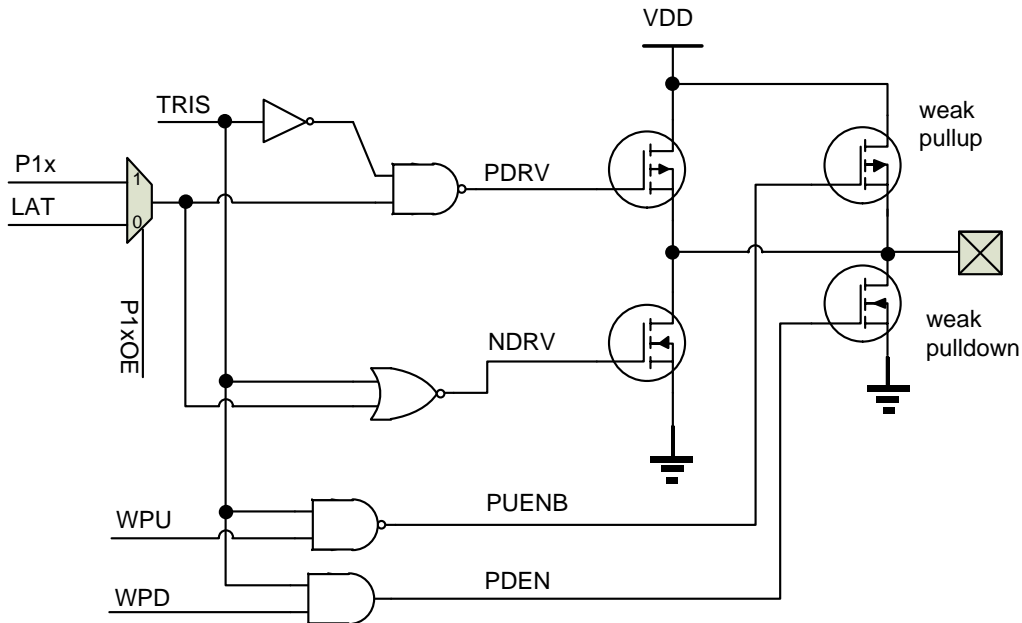


图 14.1 I/O 的一般结构

### 14.1. PORTx 端口和 TRISx 寄存器

片内的所有管脚皆为双向端口，与其相应的进出方向控制寄存器就是 TRISx 寄存器。如果 TRISx 的位为 1，则该管脚做为输入脚，反之将某一位设置为 0 会将对应 PORTx 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 I/O 处于输入状态时 (TRISx=1)，对 PORTx 进行读动作，PORTx 内容会是反映输入端口的状态（也可以通过烧录选项设置为读输出锁存器）。在 PORTx 上进行写动作时，数据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，再写入输出数据寄存器的过程。

当 MCLRE 为 1 时，PORTB[7] 读的值为 0，此时它是作为外部复位管脚。



## 14.2. 端口的其他功能

PORTA 的每个端口都有一个状态变化中断选项和弱上拉选项。

### 14.2.1. 弱上拉

PORTA/B/C 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 WPUx 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，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可以被置为关断。这是由 OPTION 寄存器中的/PAPU 位决定的。PORTB[7]内部也有弱上拉功能，它是在将 PORTB[7]设置为 MCLRB 功能时自动使能的。当 PORTB[7]被设置为 GPIO 时，该弱上拉电路由 WPUB7 控制。

### 14.2.2. 弱下拉

PORTA/B/C 所有管脚作为数字输入管脚时具有内部弱下拉功能，由寄存器 WPDx 控制。

当弱上拉和弱下拉功能是非互斥的，即它们可以同时打开。

### 14.2.3. 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 的输入上拉和下拉自动禁止，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。

## 14.3. 源电流选择

I/O PC0-1 PB2-5 支持不同的源电流驱动能力。通过配置相应的选择寄存器 PSRCx，指定的 I/O 端口可支持 3 种级别的源电流驱动能力。仅当对应的引脚被设为输出时，其源电流选择位才有效。否则，这些选择位无效。用户可参考 I/O 电气特性章节为不同应用选择所需的源电流。

## 14.4. 灌电流选择

I/O PC0-1 PB2-5 支持 2 种不同的灌电流驱动能力，设置寄存器为 PSINKx，当 I/O 设置为输出管脚时，其灌电流设置位才有效。

## 14.5. 开漏功能

每个 I/O 都支持 IO 开漏功能，设置寄存器为 ODCON A/B/C，当相关位为 1 时，该位所对应的管脚即配置为开漏。

## 14.6. 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 弱上拉控制位								1111 1111
WPUB	0x10D	PORTB 弱上拉控制位								0000 0000
WPUC	0x93	—						PORTC 弱上拉控制位		---- --00
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
TRISB	0x86	TRISB[7:0], PORTB 方向控制								1111 1111
PORTB	0x06	PORTB[7:0], PORTB 数据寄存器								xxxx xxxx
TRISC	0x87	—						PORTC 方向控制		---- --11
PORTC	0x07	—						PORTC 数据寄存器		---- --xx
IOCA	0x96	PORTA 端口变化中断允许位								0000 0000
WPDA	0x89	PORTA 弱下拉控制								0000 0000
WPDB	0x10E	PORTB 弱下拉控制								0000 0000
WPDC	0x8D	—						PORTC 弱下拉控制		---- --00
OPTION	0x81	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
PSRCB	0x10C	PORTB 源电流能力设置								1111 1111
PSRCC	0x94	—						PORTC 源电流能力设置		---- 1111
PSINKB	0x10F	PORTB 灌电流能力设置								0000 0000
PSINKC	0x9F	—						PORTC 灌电流设置		---- --00
ODCONA	0x105	PORTA 开漏设置								0000 0000
ODCONB	0x106	PORTB 开漏设置								0000 0000
ODCONC	0x107	—						PORTC 开漏设置		---- --00
ANSEL0	0x11E	ANSEL0[7:0]								0000 0000

### 14.6.1. WPUA, 地址 0x95

Bit	7	6	5	4	3	2	1	0
Name	WPUA[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	WPUA	PORTA 弱上拉控制寄存器 1 = 使能弱上拉 0 = 关闭弱上拉

**14.6.2. WPUB, 地址 0x10D**

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

Bit	Name	Function
7:0	WPUB	PORTB 弱上拉控制寄存器 1 = 使能弱上拉 0 = 关闭弱上拉

**14.6.3. WPUC, 地址 0x93**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	WPUC[1:0]	
Reset	—	—	—	—	—	—	0	0
Type	RO.0	RO.0	RO.0	RO.0	RO.0	RO.0	RW	RW

Bit	Name	Function
7:2	N/A	保留位, 读 0
1:0	WPUC	PORTC 弱上拉控制寄存器 1 = 使能弱上拉 0 = 关闭弱上拉

**14.6.4. TRISA, 地址 0x85**

Bit	7	6	5	4	3	2	1	0
Name	TRISA[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	TRISA	PORTA 方向控制寄存器 1 = 输入 0 = 输出

**14.6.5. TRISB, 地址 0x86**

Bit	7	6	5	4	3	2	1	0
Name	TRISB[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	TRISB	PORTB 方向控制寄存器 1 = 输入 0 = 输出

**14.6.6. TRISC, 地址 0x87**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	TRISC[1:0]	
Reset	—	—	—	—	—	—	1	1
Type	RO.0	RO.0	RO.0	RO.0	RO.0	RO.0	RW	RW

Bit	Name	Function
7:2	N/A	保留位, 读 0
1:0	TRISC	PORTC 方向控制寄存器 1 = 输入 0 = 输出

**14.6.7. PORTA, 地址 0x05**

Bit	7	6	5	4	3	2	1	0
Name	PORTA[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	PORTA	PORTA 数据寄存器

**14.6.8. PORTB, 地址 0x06**

Bit	7	6	5	4	3	2	1	0
Name	PORTB[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	PORTB	PORTB 数据寄存器

**14.6.9. PORTC, 地址 0x7**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	PORTC[1:0]	
Reset	—	—	—	—	—	—	X	x
Type	RO.0	RO.0	RO.0	RO.0	RO.0	RO.0	RW	RW

Bit	Name	Function
7:2	N/A	保留位, 读 0
1:0	PORTC	PORTC 数据寄存器

**14.6.10. WPDA, 地址 0x89**

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

Bit	Name	Function
7:0	WPDA	PORTA 弱下拉控制寄存器 1 = 使能弱下拉 0 = 关闭弱下拉

**14.6.11. WPDB, 地址 0x10E**

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

Bit	Name	Function
7:0	WPDB	PORTB 弱下拉控制寄存器 1 = 使能弱下拉 0 = 关闭弱下拉

**14.6.12. WPDC, 地址 0x8D**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	WPDC[1:0]	
Reset	—	—	—	—	—	—	0	0
Type	RO.0	RO.0	RO.0	RO.0	RO.0	RO.0	RW	RW

Bit	Name	Function
7:2	N/A	保留位, 读 0
1:0	WPDC	PORTC 弱下拉控制寄存器 1 = 使能弱下拉 0 = 关闭弱下拉

**14.6.13. PSRCB1, 地址 0x88**

Bit	7	6	5	4	3	2	1	0
Name	PSRCB1[7:0]							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

PSRCB1 值/[2n+1:2n]	源电流能力
00	L0: 3mA
01/10	L1: 6mA
11	L3: 24mA

Bit	Name	Function
7:6	PSRCB1[7:6]	PB5 的源电流设置位
5:4	PSRCB1[5:4]	PB4 的源电流设置位
3:2	PSRCB1[3:2]	PB3 的源电流设置位
1:0	PSRCB1[1:0]	PB2 的源电流设置位

**14.6.14. PSRCB2, 地址 0x10C**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PSRCB2[3:0]			
Reset	—	—	—	—	1	1	1	1
Type	RO.0	RO.0	RO.0	RO.0	RW	RW	RW	RW

PSRCB2 值/[2n+1:2n]	源电流能力
00	L0: 3mA
01/10	L1: 6mA
11	L3: 24mA

Bit	Name	Function
7:4	N/A	保留位, 读 0
3:2	PSRCB2[3:2]	PB7 的源电流设置位
1:0	PSRCB2[1:0]	PB6 的源电流设置位

**14.6.15. PSRCC, 地址 0x94**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PSRCC[3:0]			
Reset	—	—	—	—	1	1	1	1
Type	RO.0	RO.0	RO.0	RO.0	RW	RW	RW	RW

PSRCC 值/[2n+1:2n]	源电流能力
00	L0: 3mA
01/10	L1: 6mA
11	L3: 24mA

Bit	Name	Function
7:4	N/A	保留位, 读 0
3:2	PSRCC[3:2]	PC1 的源电流设置位
1:0	PSRCC[1:0]	PC0 的源电流设置位

**14.6.16. PSINKB, 地址 0x10F**

Bit	7	6	5	4	3	2	1	0
Name	PSINKB[7:2]						—	—
Reset	0	0	0	0	0	0	—	—
Type	RW	RW	RW	RW	RW	RW	RO.0	RO.0

Bit	Name	Function
7:2	PSINKB	PORTB[7:2]的灌电流能力设置 0: L0, 35mA 1: L2, 55mA
1:0	N/A	保留位, 读 0

**14.6.17. PSINKC, 地址 0x9F**

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

Bit	Name	Function
7:2	N/A	保留位, 读 0
1:0	PSINKC	PORTC[1:0]的灌电流能力设置 0: L0, 35mA 1: L2, 55mA

**14.6.18. ODCONA, 地址 0x105**

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

Bit	Name	Function
7:0	ODCONA	PORTA 开漏输出控制 1 = 使能开漏功能 0 = 关闭开漏功能



**14.6.19. ODCONB, 地址 0x106**

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

Bit	Name	Function
7:0	ODCONB	PORTB 开漏输出控制 1 = 使能开漏功能 0 = 关闭开漏功能

**14.6.20. ODCONC, 地址 0x107**

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

Bit	Name	Function
7:2	N/A	保留位, 读 0
1:0	ODCONC	PORTC 开漏输出控制 1 = 使能开漏功能 0 = 关闭开漏功能

**14.6.21. ANSELO, 地址 0x11E**

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

Bit	Name	Function
7:0	ANSELO	模拟选择位 在AN<7:0>引脚上分别进行模拟或数字功能的选择 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

## 15. 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 HIRC，一个是低速低功耗的 32k 时钟 LIRC，利用慢时钟测量功能可以把 LIRC 的周期用系统时钟计算出来。

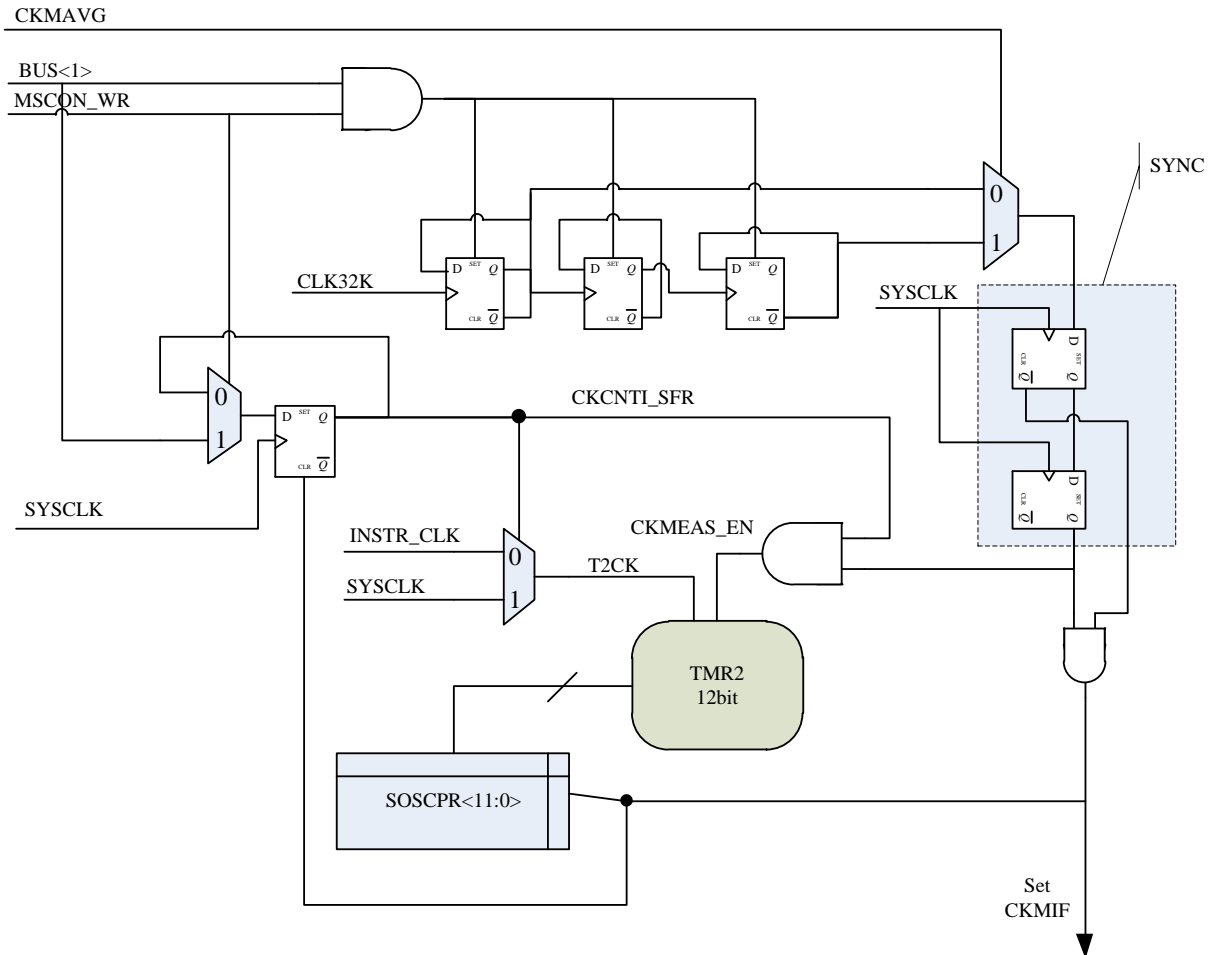


图 15.1 慢时钟测量结构框图

### 15.1. 测量原理

处于慢时钟测量模式下，Timer2 的预分频、后分频配置自动变为 1:1，Timer2 的计数时钟为系统时钟  $F_{SYS}$ ，而不是普通模式下的指令时钟  $F_{SYS}/2$ 。测量结束后结果自动存到 SOSCPR 寄存器，其单位是系统时钟的个数。

## 15.2. 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 T2CON0.2 置 1，使能 Timer2；
3. 如果选择 4 次平均，则把 MSCON0.2 置 1，否则把它清 0；
4. 置位 MSCON0.1，开始测量；
5. 测量结束后 MSCON0.1 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 SOSCPRL 即为最终结果。

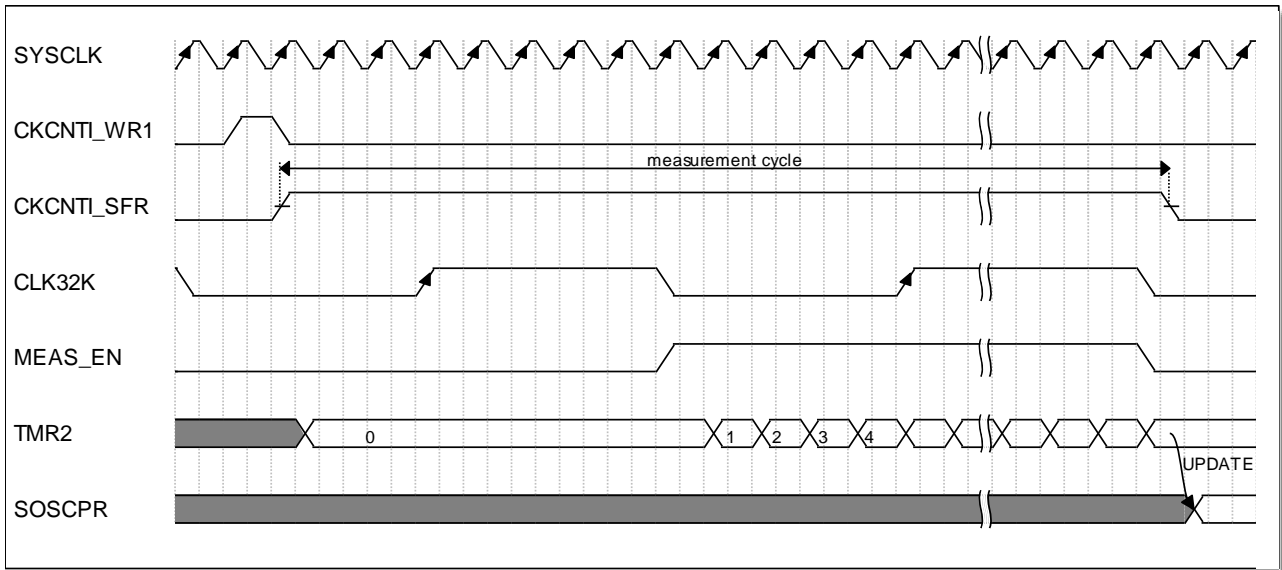


图 15.2 单次测量时序示意图（CLK32K 和 SYSCLK 未按真实比例画）

注意：

1. 在慢时钟测量过程中软件不要写 **SOSCPRL**；
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；

## 15.3. 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
MSCON0	0x1B	BGRBOE	LVROE	ROMLPE	CLKOS	SLVREN	CKMAVG	CKCNTL	T2CKRUN	0001 0000
SOSCPRL	0x1C	SOSCPRL[7:0]								1111 1111
SOSCPRH	0x1D	—				SOSCPRH[11:8]				---- 1111
INTCON	0x0B	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000
PIE1	0x8C	EEIE	CKMIE	LVDIE	ACMPIE	TMR1IE	OSFIE	TMR2IE	ADCIE	0000 0000
PIR1	0x0C	EEIF	CKMIF	LVDIF	ACMPIF	TMR1IF	OSFIF	TMR2IF	ADCIF	0000 0000

### 15.3.1. MCON0 寄存器，地址 0x1B

Bit	7	6	5	4	3	2	1	0
Name	—	—	ROMLPE	CLKOS	SLVREN	CKMAVG	CKCNTI	T2CKRUN
Reset	—	—	0	1	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	Reserved	保留位，请勿写 1
5	ROMLPE	<p>PROM 低功耗模式选择</p> <p>0 = 正常功耗模式</p> <p>1 = 开启低功耗模式</p> <p><b>注意：只能 250kHz 以下的 PROM 读速度才可以开启，即：2T 模式时，系统时钟要小于 500kHz；4T 模式时，系统时钟要小于 1MHz</b></p>
4	CLKOS	<p>CLKO信号映射管脚控制</p> <p>1 = CLKO功能映射到PB0</p> <p>0 = CLKO 功能映射到 PA2</p>
3	SLVREN	<p>软件控制 LVR 使能位，当 UCFG1&lt;1:0&gt;为 01 时：</p> <p>1 = 打开 LVR</p> <p>0 = 禁止 LVR</p> <p>当 UCFG1&lt;1:0&gt;不为 01 时，此位无实际意义</p> <p><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></p>
2	CKMAVG	<p>快时钟测量慢时钟周期的测量平均模式</p> <p>1 = 打开平均模式（自动测量并累加 4 次）</p> <p>0 = 关闭平均模式</p>
1	CKCNTI	<p>Clock Count Init –使能快时钟测量慢时钟周期</p> <p>1 = 使能快时钟测量慢时钟周期</p> <p>0 = 关闭快时钟测量慢时钟周期</p> <p>注：这一位在测量完毕后会自动归零</p>
0	T2CKRUN	<p>当 T2 时钟不是选择指令时钟时，睡眠状态 T2CK 的运行控制位</p> <p>1 = T2CK 睡眠时保持工作</p> <p>0 = T2CK 睡眠时停止工作</p>

## 15.3.2. SOSCPR 寄存器，地址 0x1C, 1D

SOSCPRL, 地址 0x1C

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

SOSCPRH, 地址 0x1D

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	SOSCPR[11:8]			
Reset	—	—	—	—	0xF			
Type	RO-0	RO-0	RO-0	RO-0	RW			

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

## 16. TOUCH 模块

### 16.1. 触摸按键功能

该系列单片机均提供多触控按键功能，该功能完全内部集成，只需一个外部参考电容，即可通过对内部寄存器的简单操作来实现触摸功能。

### 16.2. 触摸按键结构

触控按键与 I/O 引脚共用，通过相应选择寄存器的位来选择此功能。按键编号为 TK0~TK7。

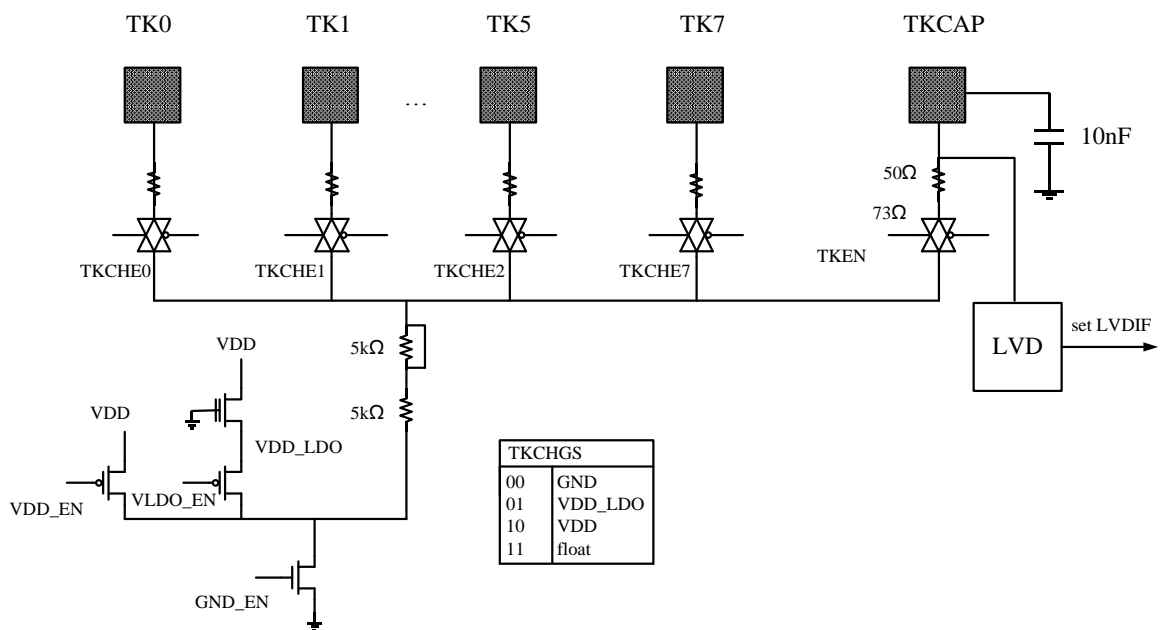


图 16.1 触摸检测原理框图

### 16.3. 触摸按键检测

本系统的按键检测原理是通过 LVD 模块对 TKCAP（参考电容管脚）电平进行检测，当电压低于 LVD 的设定值（或者反向检测，配置为高于 LVD 设定值），将产生 LVD 中断。对开始检测到产生 LVD 中断这段时间进行充放电次数计数，检测期间有按键（KEYx）被触摸时，由于 KEYx 电容增大，导致对 TKCAP 的充电（或放电）时间缩短，LVD 提前产生中断，充（放）电的次数将比无触摸时的要少，软件从而判断到触摸事件的发生。充放电过程可以通过软件实现，也可以通过配置 TKMOD 寄存器位，切换成硬件控制。其中 LDO 电路的充放电功能电位通过 VREFM 位进行控制。

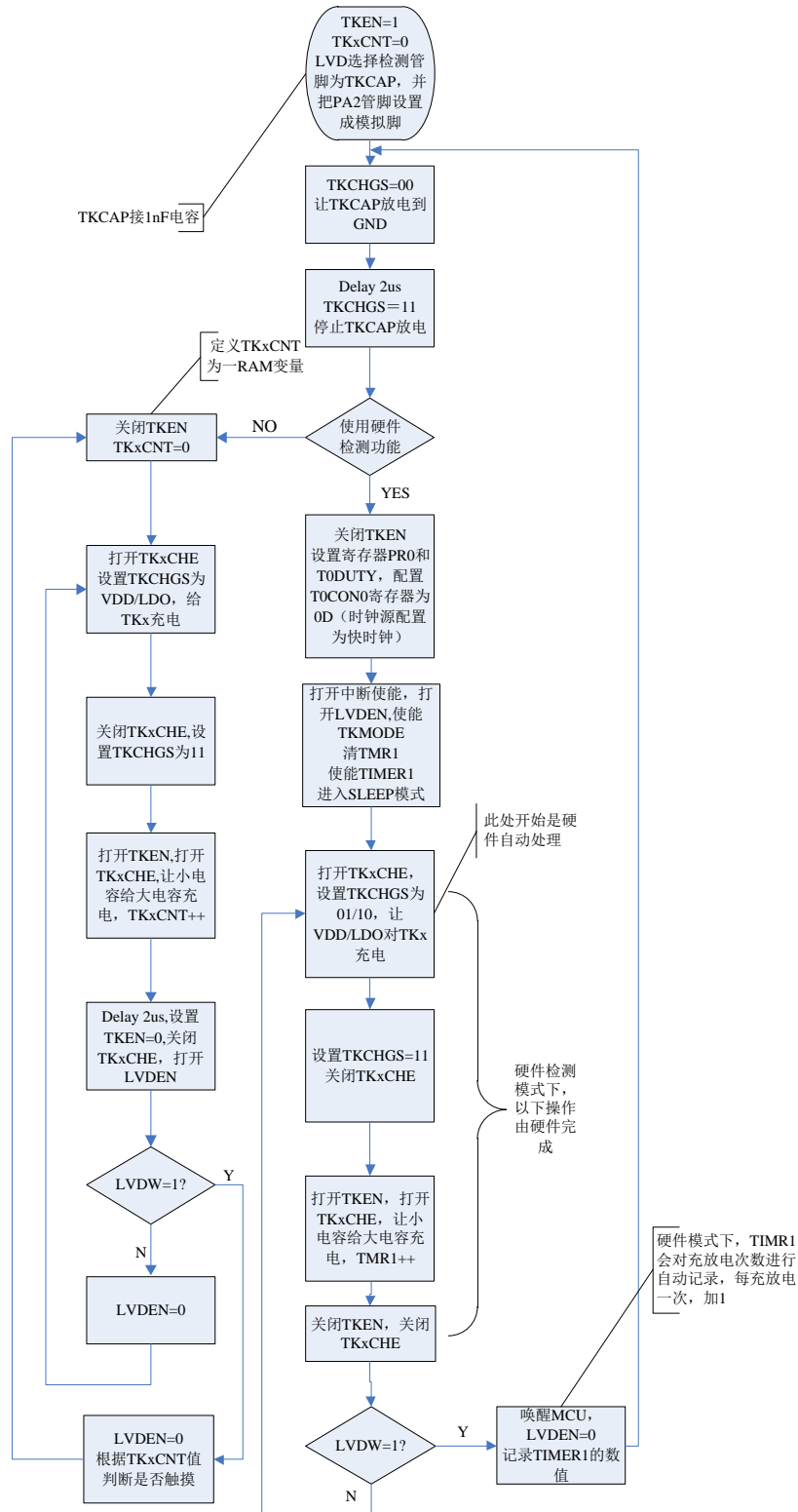


图 16.2 (充电模式下) 按键检测流程

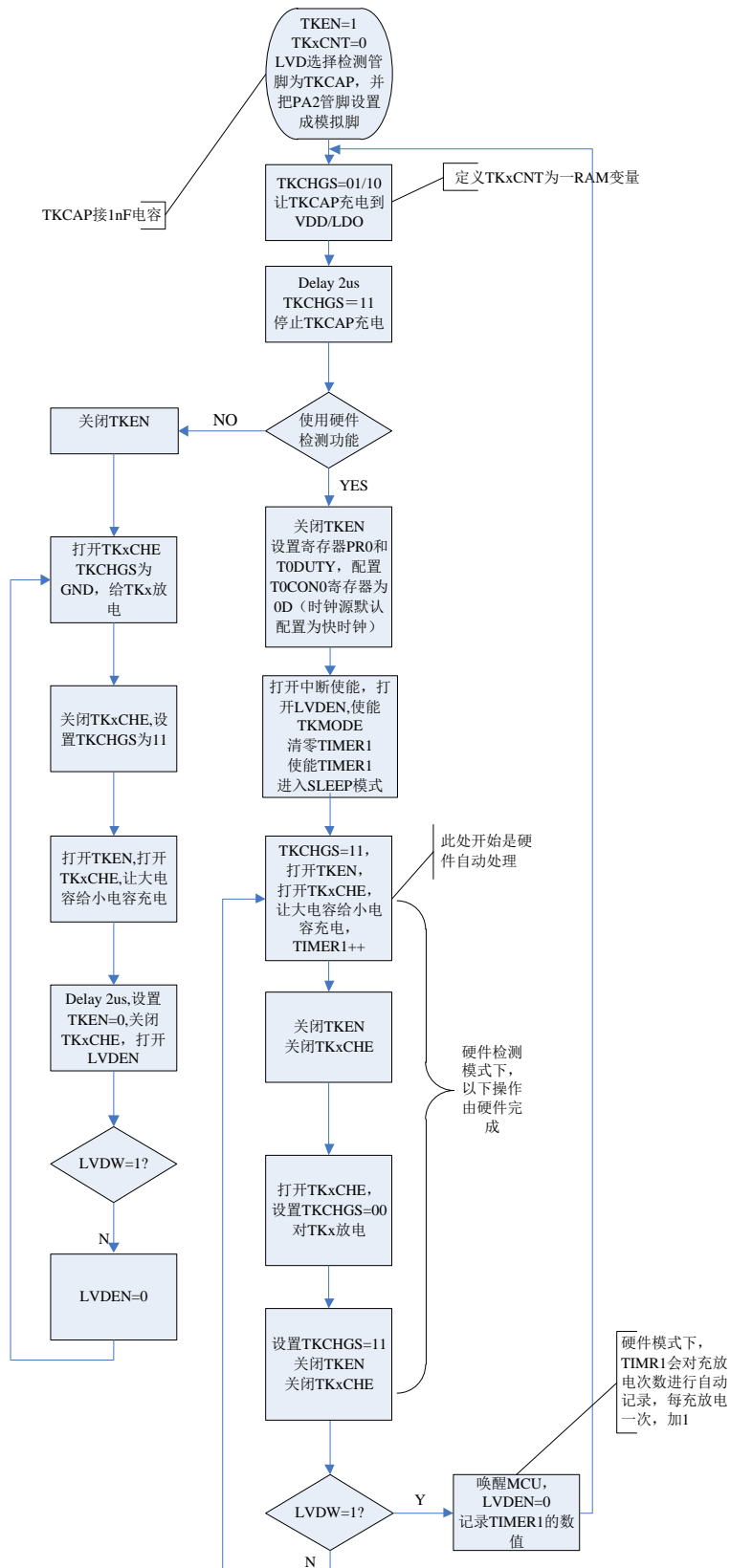


图 16.3 (放电模式下) 按键检测流程



## 16.4. 硬件检测模式

TOUCH 模块配置成硬件检测模式，TIMER0 和 TIMER1 将用于 TOUCH 模块，其中通过配置 TIMER0 的寄存器的 PR0 和 T0DUTY 来设置 PWM 的周期和占空比，控制充放电检测的充放电时间。TIMER1 则强制用于记录充放电的次数。以下是使用硬件检测模式的步骤示例：

- 1、配置端口：
  - 将 TKCAP 管脚设置成模拟管脚
- 2、配置 LVD 和看门狗模块：
  - 根据充电还是放电模式，设置 LVDP 的极性
  - 根据检测频率，设置看门狗分频比
  - **LVD 的阈值需要设置为 1.2V 档位，以免漏电**
- 3、配置 TOUCH 模块的模式和进行初始化工作
  - 设置充电或者放电检测模式
  - 设置电平标准为 VDD 或者 VLDO 模式
  - 根据模式的不同对大电容进行充电或者放电预操作
  - 打开需要检测的通道使能开关
- 4、配置 Timer0 模块功能，并开启相关模块使能，开启硬件检测功能
  - 配置 Timer0 为定时模式
  - 根据需要检测的速度，配置周期和占空比
  - 使能 Timer0
  - 使能 TKMODE 硬件检测模式开关
  - 使能 Timer1 模块
  - 使能看门狗电路
- 5、系统进入睡眠模式，等待一段时间后看门狗唤醒
- 6、读取 Timer1 的计数结果；

**注意：**

1. 在配置完电容电平和充放电模式之后，应预先做好初始化工作之后，再打开 TKMODE 控制开关，该模式可用于睡眠模式下。
2. 当使用 TOUCH 功能时需要将 TKCAP 管脚设置成模拟管脚
3. 见图 16.4、16.5，TOUCH 硬件检测时序图

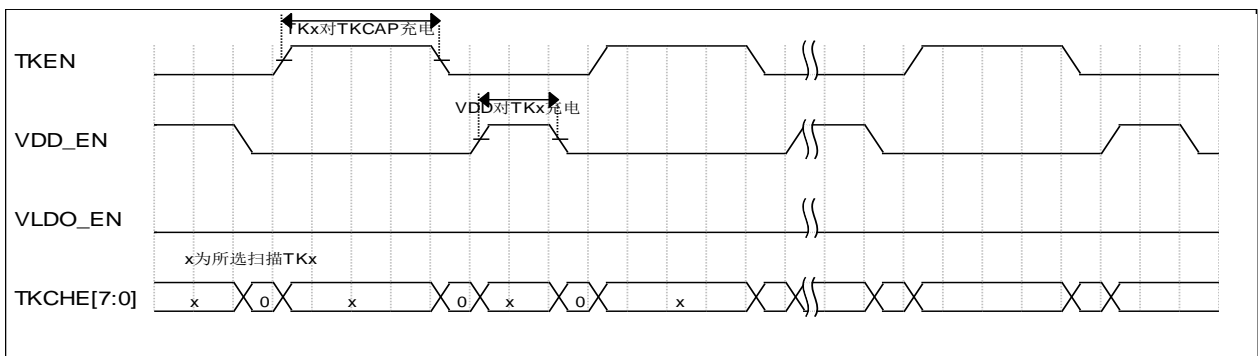


图 16.4 充电模式下 TOUCH 功能时序图

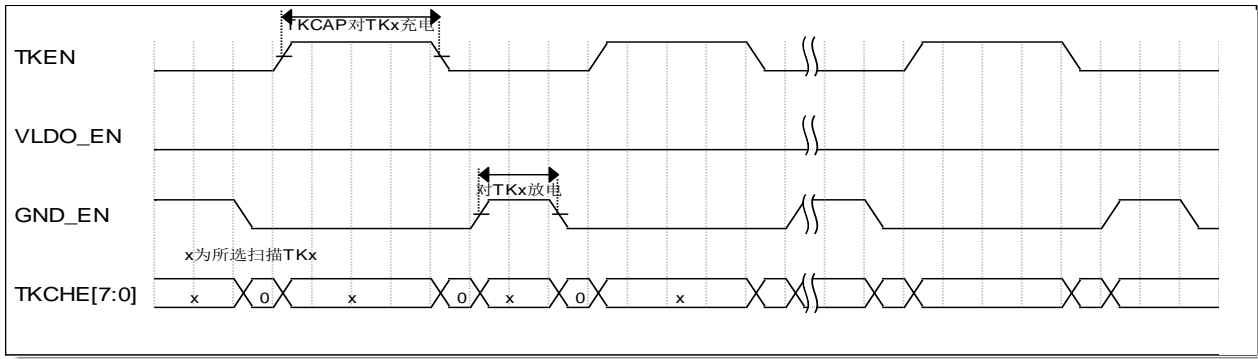


图 16.5 放电模式下 TOUCH 功能时序图

### 16.4.1. 与触摸按键相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
TKCON0	0x11C	TKRSEL[1:0]		VREFM	TKCAPM	TKMODE	TKCHGS[1:0]		TKEN	0000 0000
TKCON1	0x11D	TKCHE[7:0]								0000 0000
T0DUTY	0x18C	Timer0 产生的 PWM 波占空比设置								0000 0000
PR0	0x18D	Timer0 产生的 PWM 波周期设置								1111 1111
T1CON0	0x11A	—			T1CKPSA	T1CKRUN	T1ON	T1CKSRC[1:0]		---0 0000

### 16.4.2. TKCON0 寄存器，地址 0x11C

Bit	7	6	5	4	3	2	1	0
Name	TKRSEL[1:0]		VREFM	TKCAPM	TKMODE	TKCHGS[1:0]		TKEN
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	TKRSEL	Touch cap 下拉电阻档位选择, TKRSEL<1:0> 00: open 01: 32.5 kΩ 10: 65 kΩ 11: 130 kΩ
5	VREFM	硬件检测模式下的预充电选择 1:VDD_LDO 0:VDD
4	TKCAPM	硬件检测充放电模式选择位 1:小电容给大电容充电 0:大电容给小电容充电
3	TKMODE	检测模式选择位 1: 选择硬件检测 0: 选择软件检测
2:1	TKCHGS	Touch key 预充电选择位 00 :GND 01 :VDD_LDO 10 :VDD 11 :float
0	TKEN	touch 模块大电容的使能开关 1:开启 0:关闭

### 16.4.3. TKCON1 寄存器，地址 0x11D

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

Bit	Name	Function
7:0	TKCHE	Touch key 选择开关 1: 开启 0: 关闭  <b>注意:</b> 当系统时钟配置为晶体模式时 (XT 或 LP), 禁止对 TKCHE[1:0]写 1, 否则将影响晶体的振荡, 导致程序跑飞。

## 17. 指令集汇总

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	将寄存器 R 的 b 位清 0	0-> R(b)	
BSR R, b	将寄存器 R 的 b 位置 1	1-> R(b)	
BTSC R, b	位测试，若为 0 则跳过	Skip if R(b)=0	
BTSS R, b	位测试，若为 1 则跳过	Skip if R(b)=1	
NOP	空操作	None	
CLRWDT	清看门狗（喂狗）	0-> WDT	/PF, /TF
SLEEP	进入睡眠模式	0-> WDT, STOP OSC	/PF, /TF
STTMD	把 W 内容存到 TMODE	W-> TMODE <sup>1</sup>	
CTLIO R	设置 TRISr 寄存器	W-> TRISr	
STR R(MOVWF)	将 W 存到 R	W-> R	
LDR R, d(MOVF)	将 R 存到 d	R-> d	Z
SWAPR R, d	R 半字节交换	[R(0-3)R(4-7)]-> d	
INCR R, d	R+1	R+ 1-> d	Z
INCRSZ R, d	R+1, 结果为 0 则跳过	R+ 1-> d	
ADDWR R, d	W 与 R 相加	W+ R-> d	C, HC, Z
SUBWR R, d	R 减 W	R- W-> d	C, HC, Z
		R+ /W+ 1-> d	
DECR R, d	R-1	R- 1-> d	Z
DECRSZ R, d	R-1, 结果为 0 则跳过	R- 1-> d	
ANDWR R, d	W 与 R 相与	R& W-> d	Z
IORWR R, d	W 与 R 相或	W  R-> d	Z
XORWR R, d	W 与 R 异或	W^ R-> d	Z
COMR R, d	求 R 的反码	/R-> d	Z
RRR R, d	R 带进位循环右移	R(n)-> R(n-1),	C
		C-> R(7), R(0)-> C	
RLR R, d	R 带进位循环左移	R(n)-> R(n+1),	C
		C-> R(0), R(7)-> C	
CLRW	把 W 清 0	0-> W	Z
CLRR R	把 R 清 0	0-> R	Z
RETI	从中断返回	Stack-> PC, 1-> GIE	
RET	从子程序返回	Stack-> PC	
LCALL N	调用子程序	N-> PC,	
		PC+1-> Stack	
LJUMP N	无条件跳转	N-> PC	
LDWI I(MOVLW)	立即数存到 W	I-> W	
ANDWI I	W 与立即数 I 相与	W& I-> W	Z
IORWI I	W 与立即数 I 相或	W  I-> W	Z
XORWI I	W 与立即数 I 异或	W^ I-> W	Z
RETW I	带立即数的返回	Stack-> PC, I-> W	
ADDWI I	W 与立即数相加	W+I-> W	C, HC, Z
SUBWI I	立即数减 W	I-W-> W	C, HC, Z

操作码字段说明

字段	描述
R(F)	SFR 地址
W	工作寄存器
b	寄存器的位地址
I/Imm(k)	立即数
X	不关心的值, 可以是 0 或者 1
d	目标寄存器选择
	0: 结果存放到 W
	1: 结果存放到 SFR
N	程序绝对地址
PC	程序计数器
TMODE	TMODE <sup>1</sup> 寄存器
TRISr	TRISr 寄存器, r 可以是 A, B, C
C	进位
HC	半进位
Z	0 标志位
/PF	掉电标志位
/TF	WDT 溢出标志位

注意:

1. 在 FT62F13x 系列芯片中, TMODE 寄存器是指 OPTION, 即 STTMD 指令的操作是把 W 存到 OPTION;

## 18. 芯片的电气特性

### 18.1. 极限参数

工作温度 .....	-40~+85°C
存储温度 .....	-40~+125°C
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：上述值为芯片工作条件的极限参数值，超过极限参数所规定的范围，可能会对芯片造成永久性损坏。若芯片长时间工作在极限参数范围外的条件下，其可靠性可能受到影响。

### 18.2. 内置高频振荡器（HIRC）

电气参数	最小值 <sup>(1)</sup>	典型值 <sup>(1)</sup>	最大值 <sup>(1)</sup>	单位	条件/备注
校准范围	15.84	16	16.16	MHz	25°C, $V_{DD} = 2.5V$
随温度变化范围	—	±4.0%	—	—	-40~85°C, $V_{DD} = 2.5V$
随电源电压变化范围	-1.0%	—	1.0%	—	25°C, $V_{DD} = 1.9\sim 5.5V$
$I_{HIRC}$ 工作电流	—	40	—	μA	25°C, $V_{DD} = 3.0V$
启动时间	—	2.5	—	μs	25°C, $V_{DD} = 3.0V$

(1) 数据基于特性值，并未生产测试。

### 18.3. 内置低频振荡器（LIRC）

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。

电气参数	最小值 <sup>(1)</sup>	典型值 <sup>(1)</sup>	最大值 <sup>(1)</sup>	单位	条件/备注
振荡频率	30.4	32	33.6	kHz	25°C, $V_{DD} = 2.5V$
随温度变化范围	-2.0%	—	2.0%	—	-40 ~ 85°C, $V_{DD} = 2.5V$
随电源电压变化范围	-1.0%	—	1.0%	—	25°C, $V_{DD} = 1.9\sim 5.5V$
$I_{LIRC}$ 工作电流	—	1.3	—	μA	25°C, $V_{DD} = 3.0V$
启动时间	—	4.6	—	μs	25°C, $V_{DD} = 3.0V$

(1) 数据基于特性值，并未生产测试。

## 18.4. 低电压复位电路（LVR）

电气参数	最小值 <sup>(1)</sup>	典型值 <sup>(1)</sup>	最大值 <sup>(1)</sup>	单位	条件/备注
I <sub>LVR</sub> 工作电流	—	16.2	—	μA	25°C, V <sub>DD</sub> = 3.3V
V <sub>LVR</sub> , LVR 阈值	1.94	2.0	2.06	V	25°C
	2.13	2.2	2.27		
	2.42	2.5	2.58		
	2.72	2.8	2.88		
	3.01	3.1	3.19		
	3.49	3.6	3.71		
	3.98	4.1	4.22		
LVR delay	—	125	157	μs	25°C, V <sub>DD</sub> = 1.9~5.5V

(1) 数据基于特性值，并未生产测试。

## 18.5. 低电压侦测电路（LVD）

电气参数	最小值 <sup>(1)</sup>	典型值 <sup>(1)</sup>	最大值 <sup>(1)</sup>	单位	条件/备注
I <sub>LVD</sub> 工作电流	—	21.4	—	μA	25°C, V <sub>DD</sub> = 3.3V
V <sub>LVD</sub> , LVD 阈值	1.16	1.2	1.24	V	25°C
	1.94	2.0	2.06		
	2.33	2.4	2.47		
	2.62	2.7	2.78		
	2.91	3.0	3.09		
	3.20	3.3	3.40		
	3.49	3.6	3.71		
	3.88	4.0	4.12		
LVD delay	—	125	157	μs	25°C, V <sub>DD</sub> = 1.9~5.5V

(1) 数据基于特性值，并未生产测试。

## 18.6. 上电复位电路（POR）

电气参数	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件/备注
I <sub>POR</sub> 工作电流	—	140	—	nA	25°C, V <sub>DD</sub> = 3.3V
V <sub>POR</sub>	—	1.65	—	V	25°C

(1) 数据基于特性值，并未生产测试。

## 18.7. I/O PAD 电路

电气参数		最小值 <sup>(1)</sup>	典型值 <sup>(1)</sup>	最大值 <sup>(1)</sup>	单位	条件/备注
V <sub>IL</sub>		0	—	0.3* V <sub>DD</sub>	V	
V <sub>IH</sub>		0.7* V <sub>DD</sub>	—	V <sub>DD</sub>	V	
漏电流		-1	—	1	μA	V <sub>DD</sub> = 5V
源电流(source)	PB2~7, PC0~1	L0	—	-3	—	mA 25°C, V <sub>DD</sub> = 5V, V <sub>OH</sub> = 4.5V
	PB2~7, PC0~1	L1	—	-6	—	
	PA2, PB2~7, PC0~1	L2	—	-18	—	
	PA0~1, PA3~7, PB0~1	L3	—	-24	—	
沉电流(sink)	PB2~7, PC0~1	L0	—	35	—	mA 25°C, V <sub>DD</sub> = 5V, V <sub>OL</sub> = 0.5V
	PA2, PB2~7, PC0~1	L1	—	53	—	
	PA0~1, PA3~7, PB0~1	L2	—	55	—	
上拉电阻		—	20	—	kΩ	单独使能上拉
下拉电阻		—	20	—	kΩ	单独使能下拉
上拉电阻		—	100	—	kΩ	同时使能上拉和 下拉
下拉电阻		—	100	—	kΩ	

(1) 数据基于特性值，并未生产测试。



## 18.8. 总体工作电流 (I<sub>DD</sub>)

电气参数	Sysclk	典型值@V <sub>DD</sub> <sup>(1)</sup>			单位
		2.0V	3.0V	5.5V	
正常模式 (2T), I <sub>DD</sub>	16MHz	—	1.244	1.320	mA
	8MHz	0.588	0.875	0.924	
	4MHz	0.463	0.687	0.706	
	2MHz	0.349	0.403	0.412	
	1MHz	0.220	0.256	0.260	
	32kHz	0.024	0.032	0.033	
低功耗模式 (2T), I <sub>DD</sub>	32kHz	0.007	0.008	0.009	μA
休眠模式 (Sleep, WDT OFF, LVR OFF), I <sub>SB</sub>	—	0.072	0.092	0.128	
休眠模式 (Sleep, WDT ON, LVR OFF)	—	1.077	1.468	1.582	
休眠模式 (Sleep, WDT OFF, LVR ON)	—	11.475	15.520	20.978	
休眠模式 (Sleep, WDT ON, LVR ON)	—	12.402	16.792	22.286	
休眠模式 (Sleep, WDT OFF, LVR OFF, LVD ON)	—	17.425	20.805	25.274	

(1) 数据基于特性值，并未生产测试。

注：

1. 测试环境温度为 25°C；
2. 睡眠电流的测试条件为 I/O 处于输入模式并外部下拉到 0；

## 18.9. AC 电气参数

电气参数		最小值 <sup>(1)</sup>	典型值 <sup>(1)</sup>	最大值 <sup>(1)</sup>	单位	条件/备注
F <sub>sys</sub> (系统时钟频率)	2T/4T	—	—	8	MHz	-40~85°C, V <sub>DD</sub> = 1.9~5.5V
		—	—	16	MHz	-40~85°C, V <sub>DD</sub> = 2.7~5.5V
指令周期 (T <sub>ins</sub> )	2T	—	125	—	ns	系统时钟 HIRC
	4T	—	250	—	ns	
	2T	—	61	—	μs	系统时钟 LIRC
	4T	—	122	—	μs	
T <sub>0CKI</sub> 输入周期		(T <sub>t0ck</sub> +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 (T <sub>DRH</sub> )		—	4.2	—	ms	25°C, PWRT disable
外部复位脉冲宽度 (T <sub>MCLRb</sub> )		2000	—	—	ns	25°C
WDT 周期 (T <sub>WDT</sub> )		—	1	—	ms	无预分频, WDTPS<3:0>=0000

(1) 数据基于特性值，并未生产测试。

注 1: T<sub>t0ck</sub> 是指由 T<sub>0CKSRC</sub> 所选的时钟周期。

注 2: 除特殊说明，特性测试条件为: T=25°C, V<sub>DD</sub> = 1.9~5.5V。

## 18.10. 12bit ADC 特性

### ADC 特性参数

电气参数	最小值 <sup>(1)</sup>	典型值 <sup>(1)</sup>	最大值 <sup>(1)</sup>	单位	条件/备注
ADC 工作电压 $V_{DD}$	2.7	—	5.5	V	
ADC 工作电流 $I_{VDD}$	—	85	—	$\mu A$	25°C, $V_{REFP} = V_{DD} = 2.7V$ , ADC 转换时钟频率为 250kHz
	—	95	—	$\mu A$	25°C, $V_{REFP} = V_{DD} = 3.0V$ , ADC 转换时钟频率为 250kHz
	—	125	—	$\mu A$	25°C, $V_{REFP} = V_{DD} = 5.5V$ , ADC 转换时钟频率为 250kHz
模拟输入电压 $V_{AIN}$	$V_{REFN}$	—	$V_{REFP}$	V	
外部参考电压 $V_{REF}$	—	—	$V_{DD}$	V	
分辨率	—	—	12	位	
积分误差 $E_{IL}$	—	$\pm 2$	—	LSB	25°C, $V_{REFP} = V_{DD} = 5.0V$ , $V_{REFN} = GND$ , ADC 转换时钟 频率为 250kHz
微分误差 $E_{DL}$	—	$\pm 2$	—	LSB	
偏移误差 $E_{OFF}$	—	$\pm 3$	—	LSB	25°C, $V_{REFP} = V_{DD} = 5.0V$ , $V_{REFN} = GND$
增益误差 $E_{GN}$	—	$\pm 5$	—	LSB	
转换时钟周期 TAD	—	2	—	$\mu s$	$V_{REFP} > 3.0V$ , $V_{DD} > 3.0V$
转换时钟数	—	15	—	TAD	
稳定时间( $T_{ST}$ )	—	15	—	$\mu s$	
采样时间( $T_{ACQ}$ )	—	1.5	—	TAD	
建议的模拟电压源阻抗(ZAI)	—	—	10	k $\Omega$	—

(1) 数据基于特性值，并未生产测试。

### ADC Vref 特性参数

电气参数	最小值 <sup>(1)</sup>	典型值 <sup>(1)</sup>	最大值 <sup>(1)</sup>	单位	条件/备注
内置参考电压 ADCVref	0.492	0.5	0.508	V	25°C, $V_{DD} = 5V$
	1.992	2	2.008	V	25°C, $V_{DD} = 5V$
	2.988	3	3.012	V	25°C, $V_{DD} = 5V$
内置参考电压 0.5V 稳定时间 $T_{VRINT}$	—	400	—	$\mu s$	25°C, $V_{DD} = 5V$
内置参考电压 2.0V 稳定时间 $T_{VRINT}$	—	600	—	$\mu s$	25°C, $V_{DD} = 5V$ , 1 $\mu F$
	—	450	—	$\mu s$	25°C, $V_{DD} = 5V$
内置参考电压 3.0V 稳定时间 $T_{VRINT}$	—	800	—	$\mu s$	25°C, $V_{DD} = 5V$ , 1 $\mu F$
	—	450	—	$\mu s$	25°C, $V_{DD} = 5V$
稳定时间 $T_{VRINT}$	—	1200	—	$\mu s$	25°C, $V_{DD} = 5V$ , 1 $\mu F$

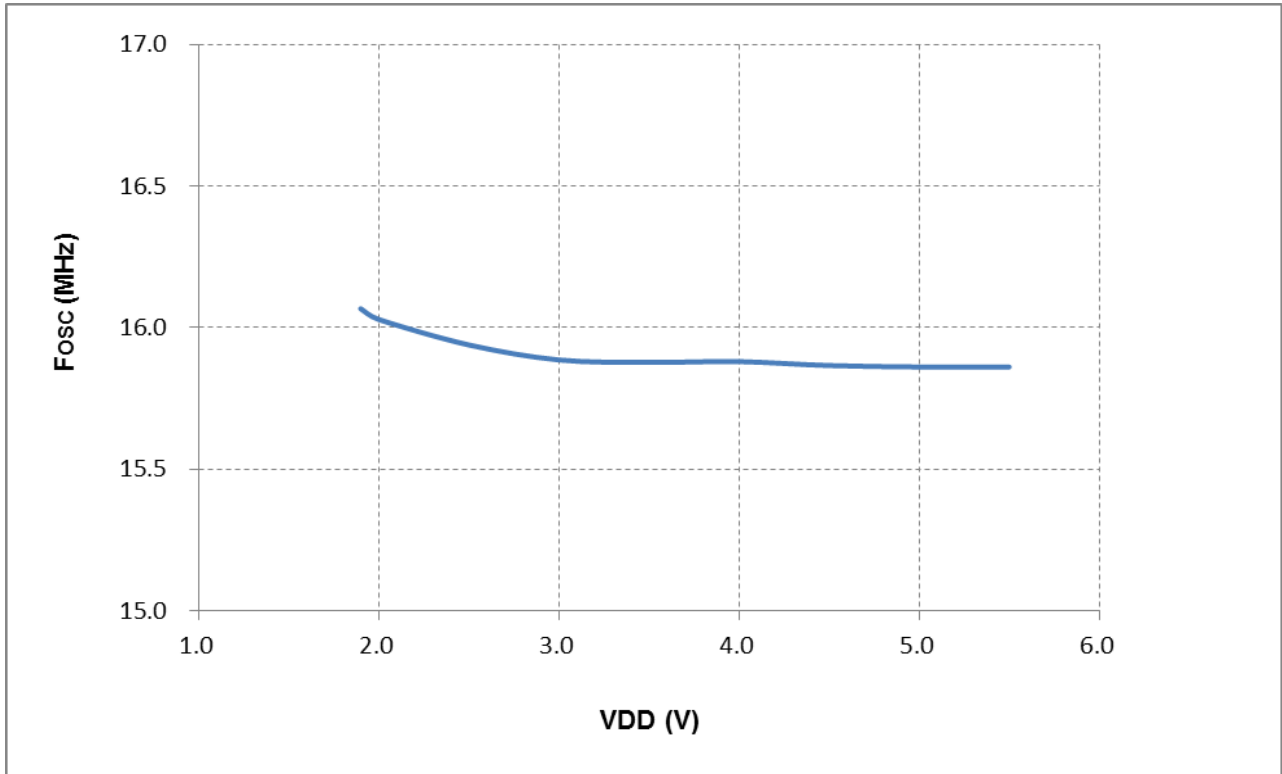
(1) 数据基于特性值，并未生产测试。

注：除非另外说明，否则“典型值”一栏的数据都是在 5.0V, 25°C 的条件下给出。

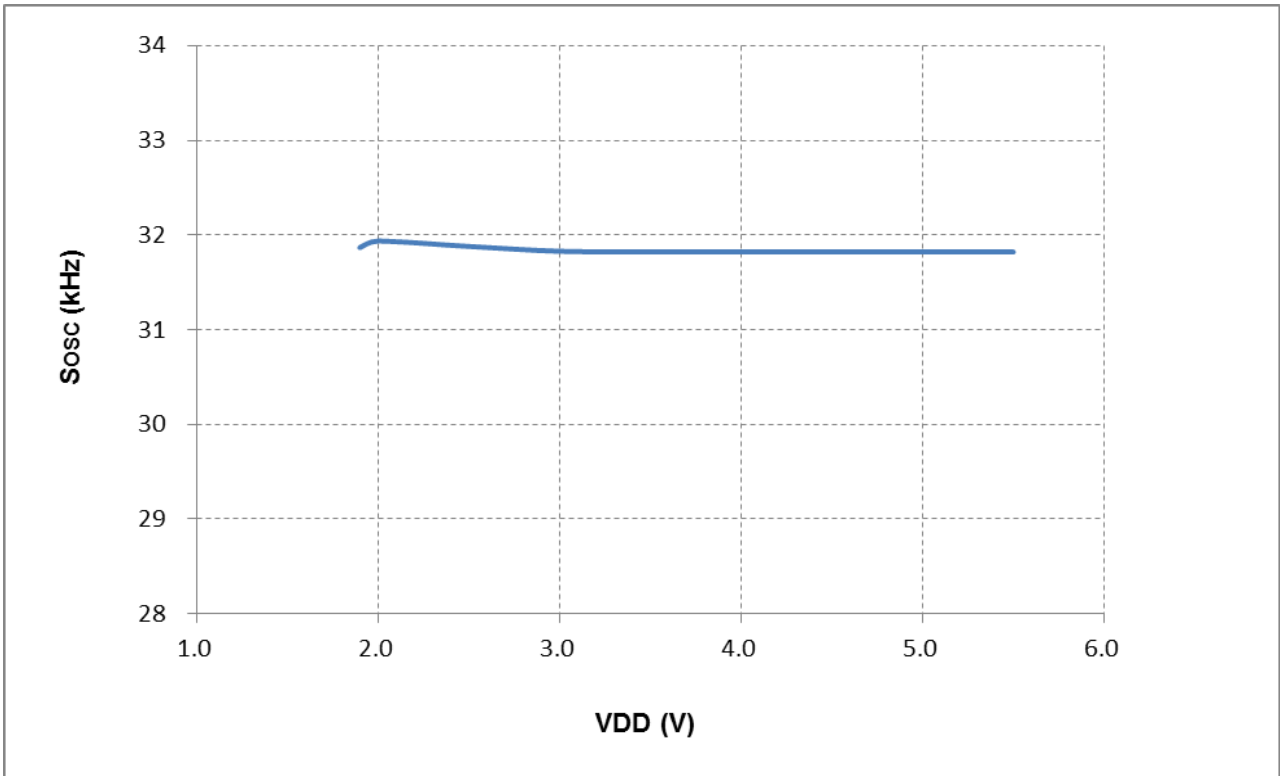
## 18.11. 直流和交流特性曲线图

注意：本节提供的图表基于特性值，仅用作设计参考，未经生产测试。

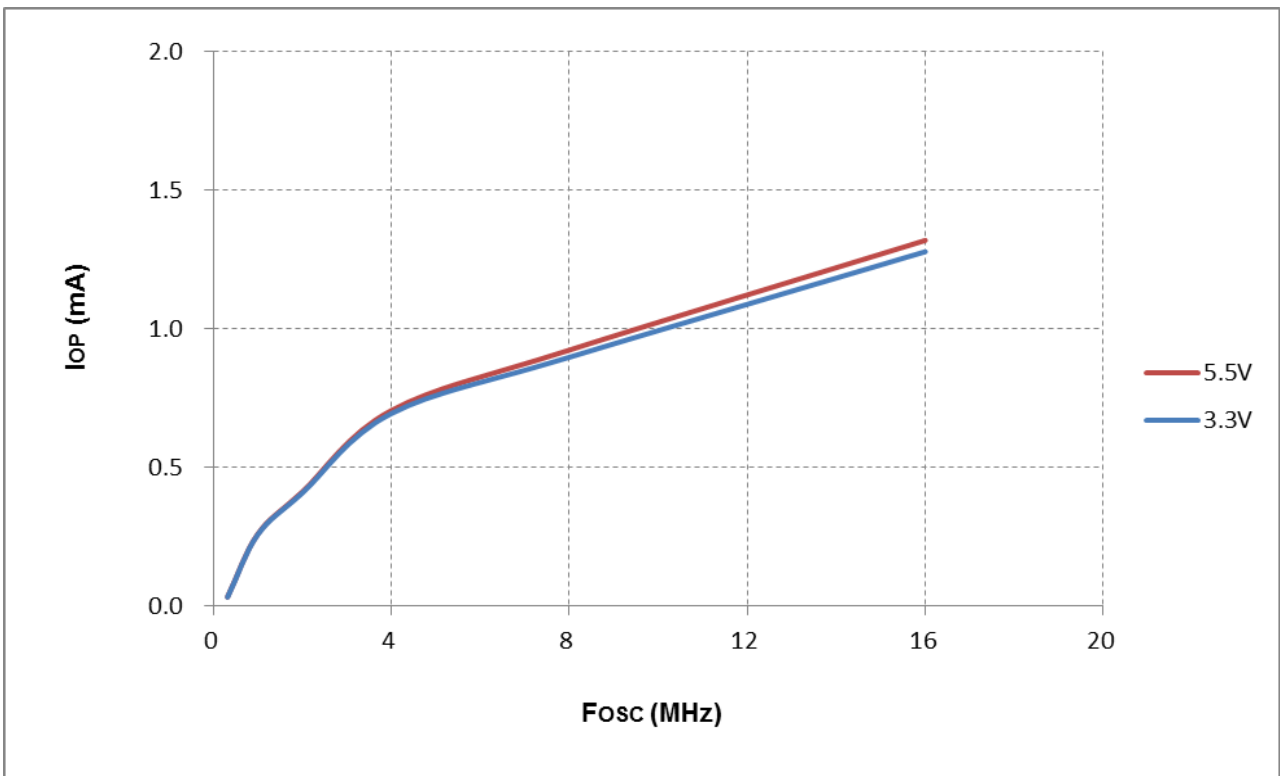
### 18.11.1. HIRC vs $V_{DD}$ ( $T_A=25^\circ\text{C}$ )



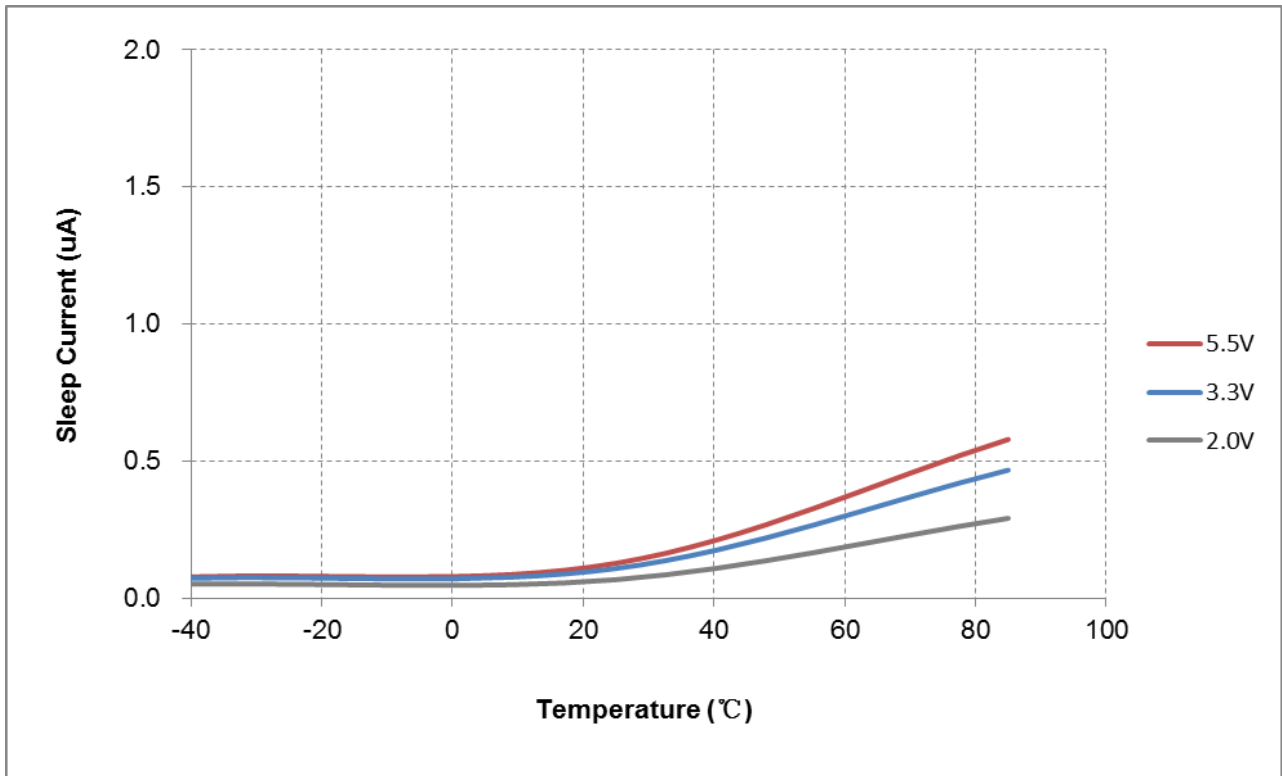
### 18.11.2. LIRC vs $V_{DD}$ ( $T_A=25^\circ\text{C}$ )



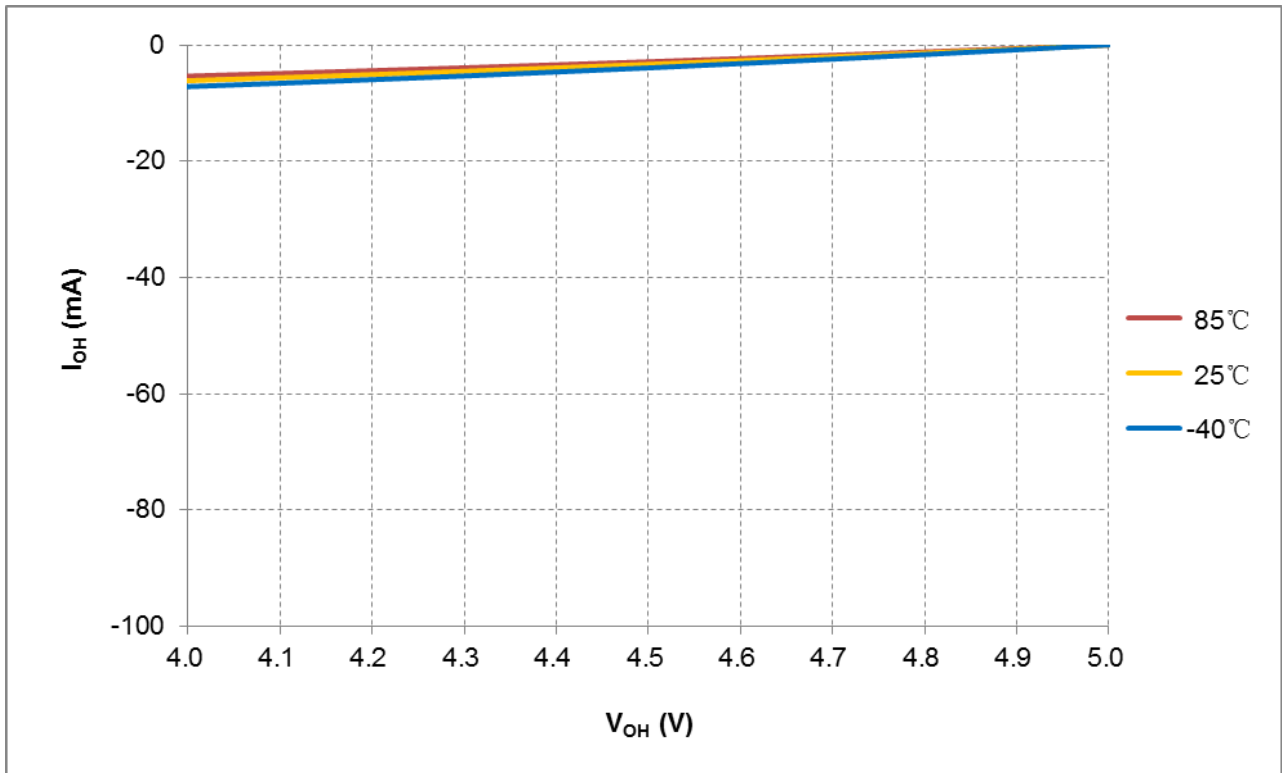
### 18.11.3. 不同 $V_{DD}$ 下, $I_{DD}$ vs Freq ( $2T, T_A=25^\circ\text{C}$ )



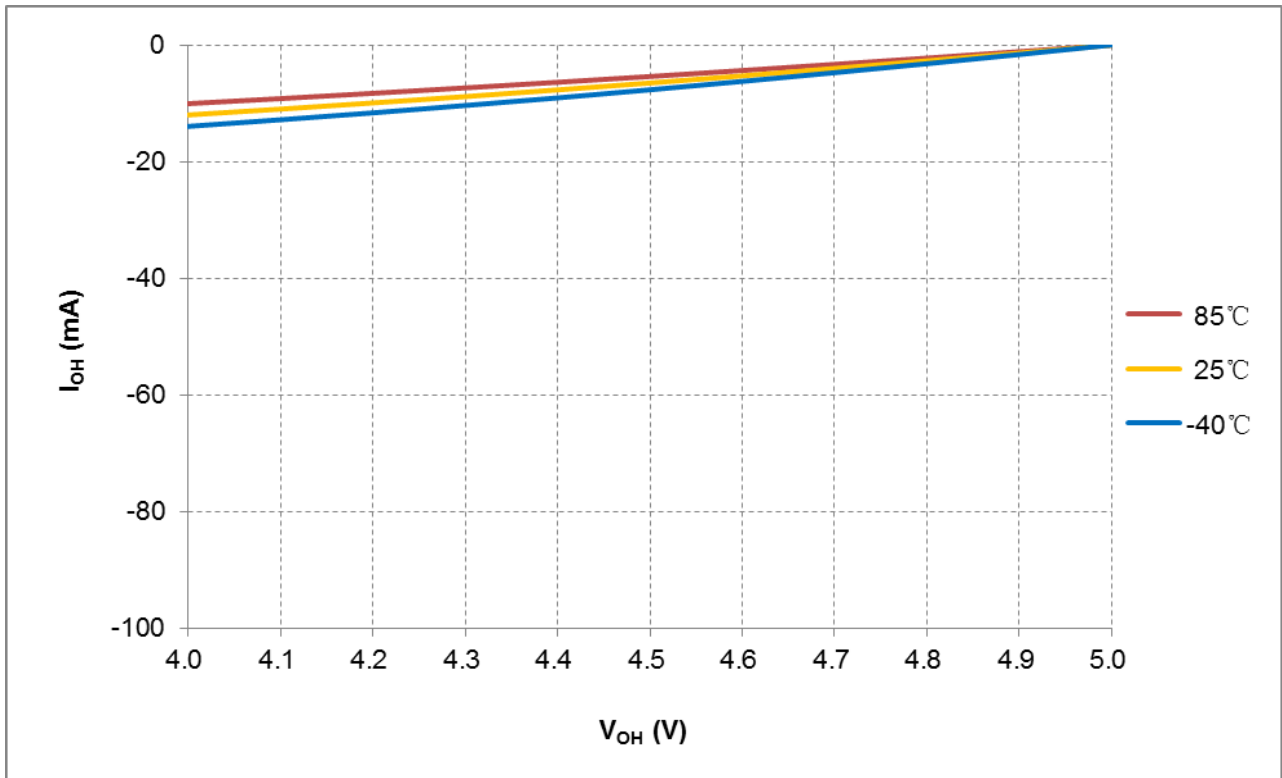
**18.11.4. 不同  $V_{DD}$  下,  $I_{SB}$  (睡眠电流)随温度变化曲线**



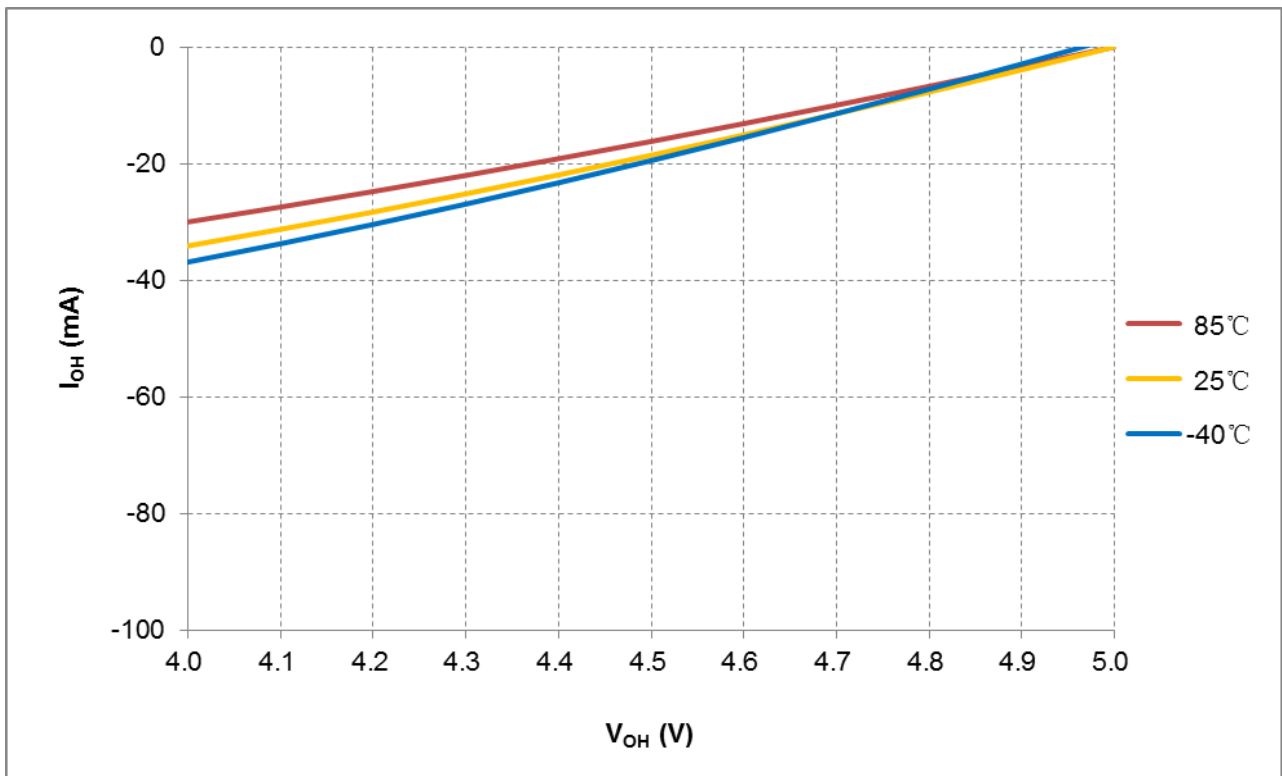
**18.11.5. 不同温度下,  $I_{OH}$  (L0 -3mA) vs  $V_{OH}$  @ $V_{DD}=5V$**



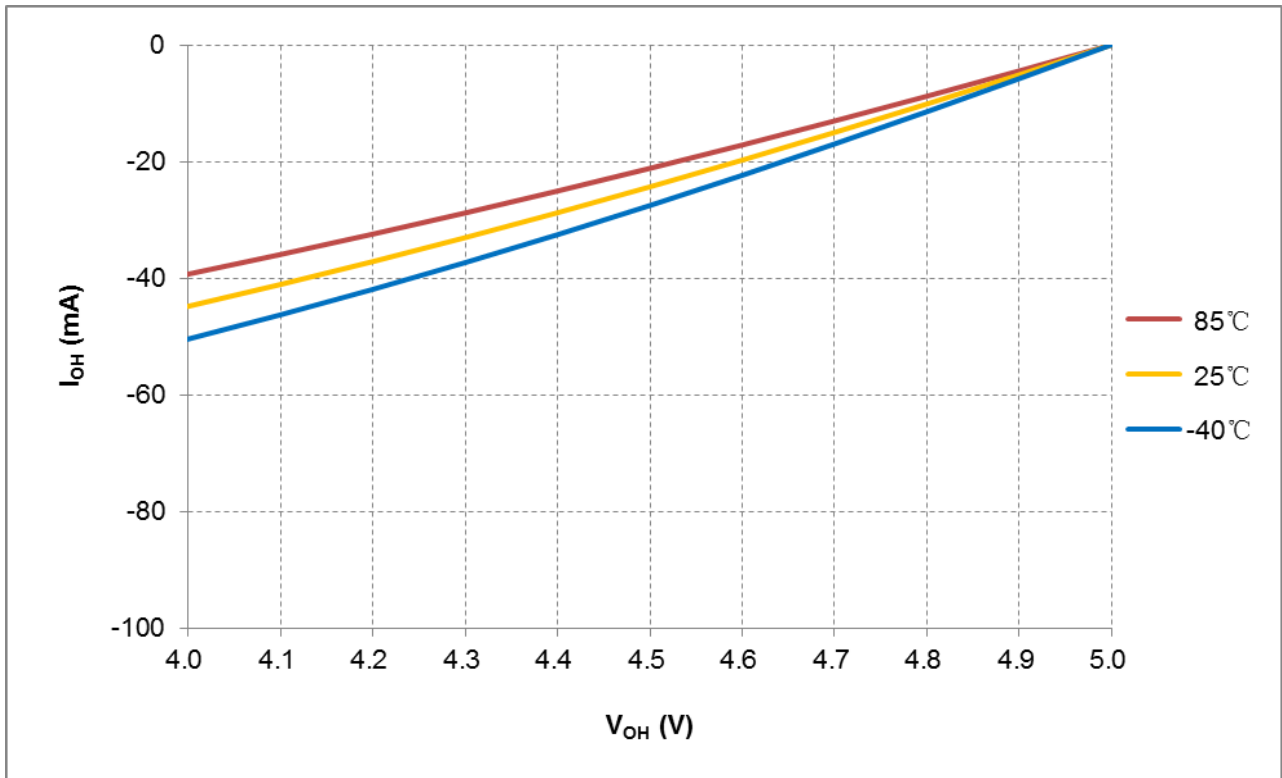
**18.11.6. 不同温度下,  $I_{OH}$  ( L1 -6mA ) vs  $V_{OH}$  @ $V_{DD}=5V$**



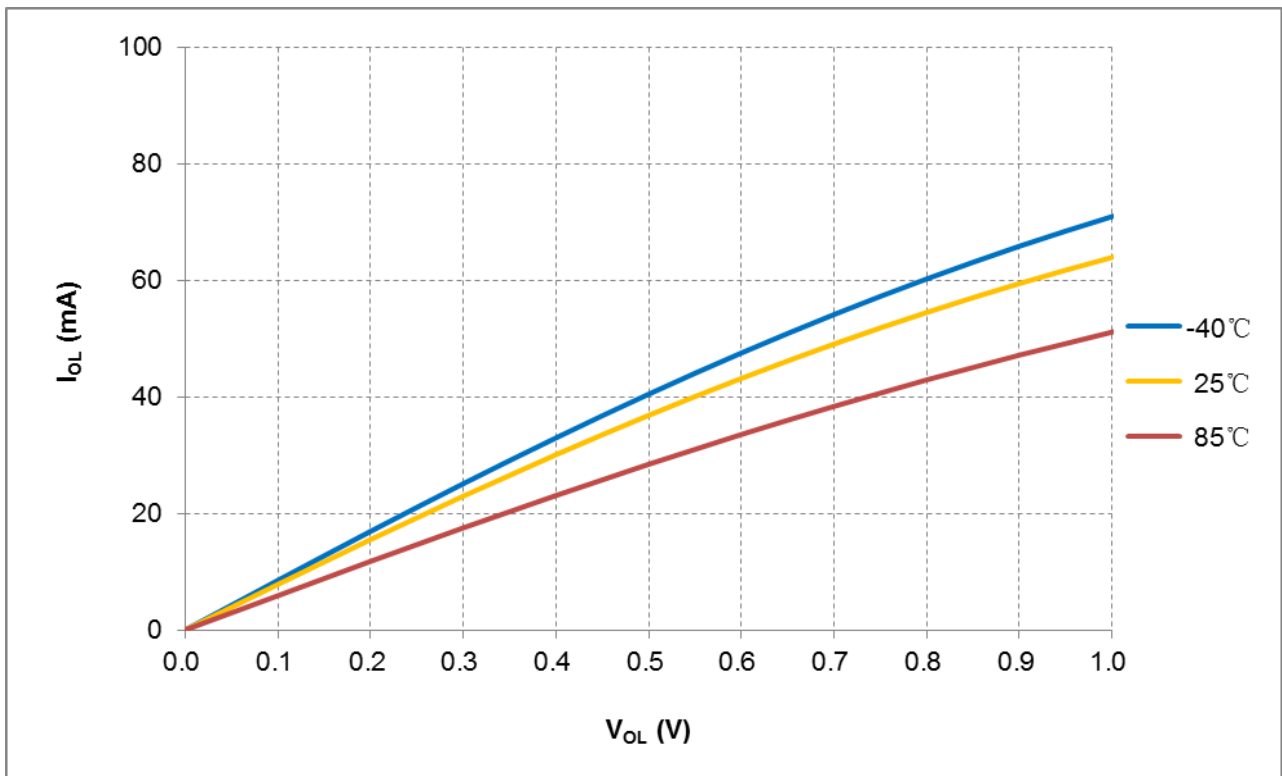
**18.11.7. 不同温度下,  $I_{OH}$  ( L2 -18mA ) vs  $V_{OH}$  @ $V_{DD}=5V$**



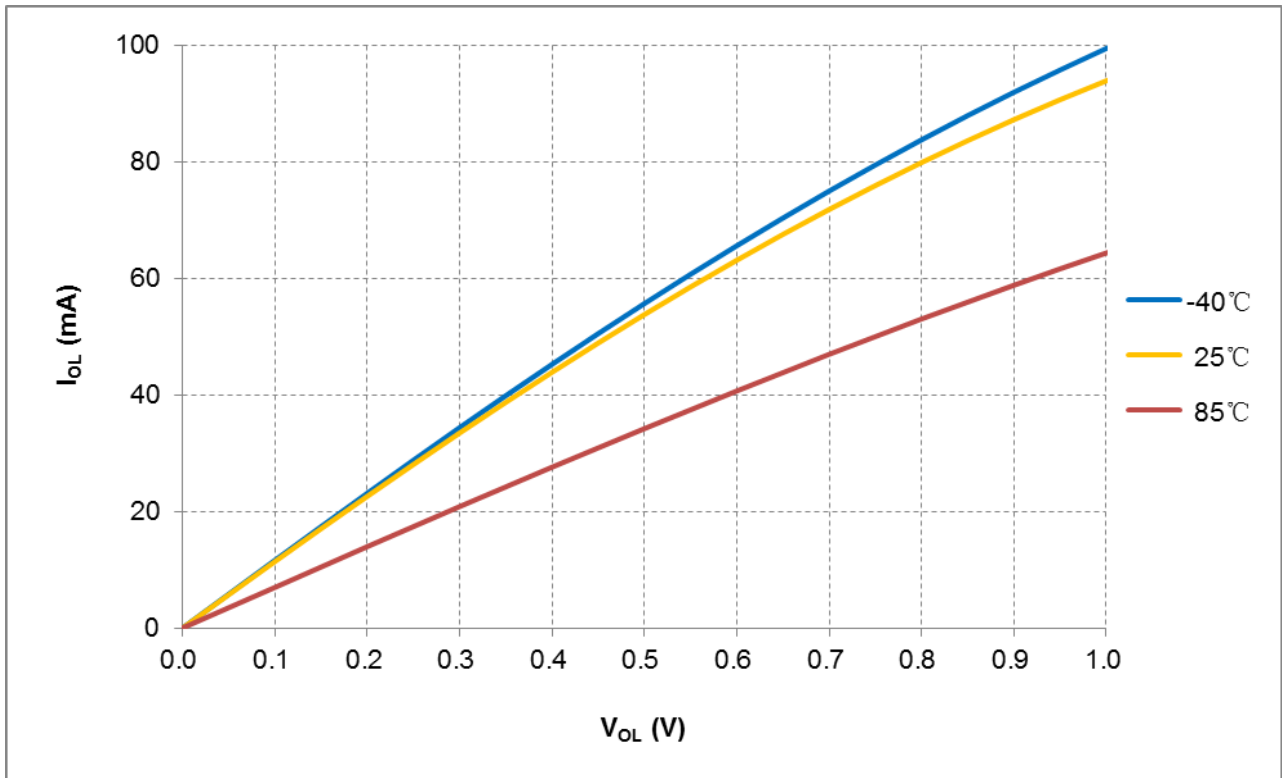
**18.11.8. 不同温度下,  $I_{OH}$  ( L3 -24mA ) vs  $V_{OH}$  @ $V_{DD}=5V$**



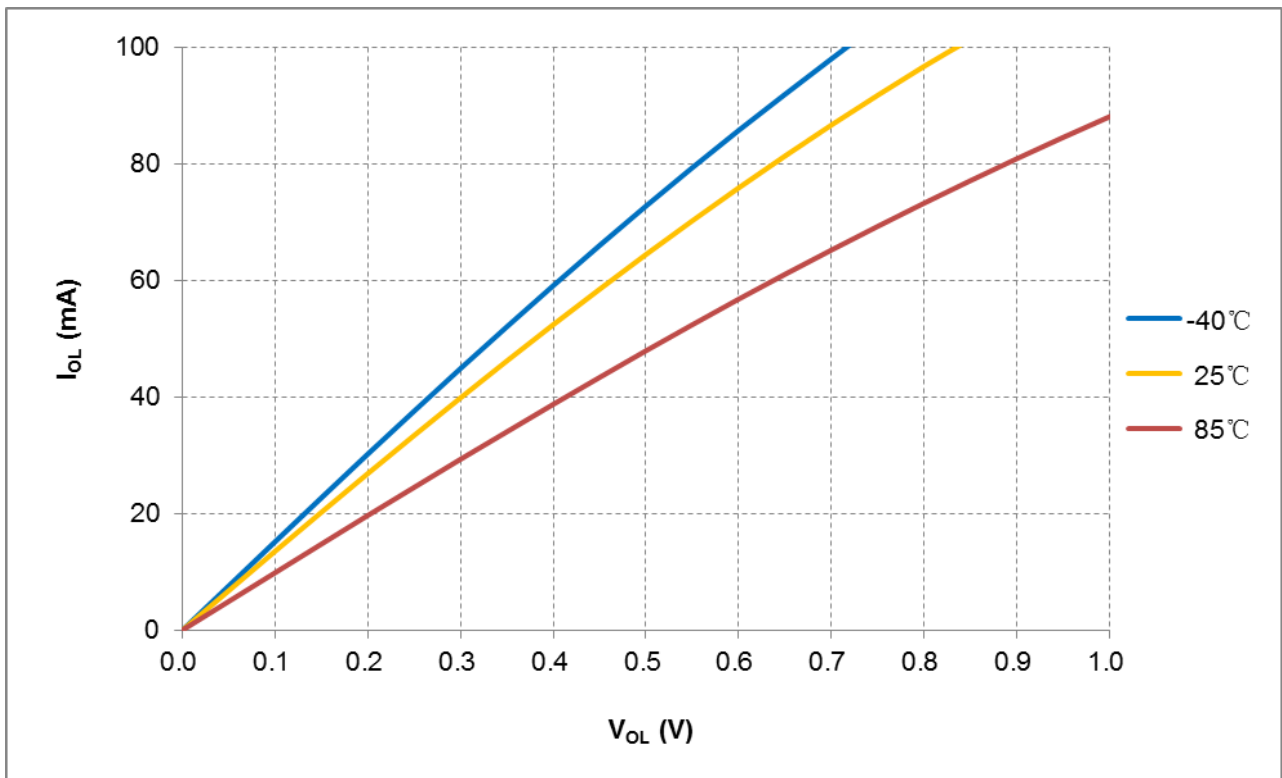
**18.11.9. 不同温度下,  $I_{OL}$  ( L0 35mA ) vs  $V_{OL}$  @ $V_{DD}=5V$**



**18.11.10. 不同温度下,  $I_{OL}$  ( L1 53mA ) vs  $V_{OL}$  @ $V_{DD}=5V$**



**18.11.11. 不同温度下,  $I_{OL}$  ( L2 55mA ) vs  $V_{OL}$  @ $V_{DD}=5V$**

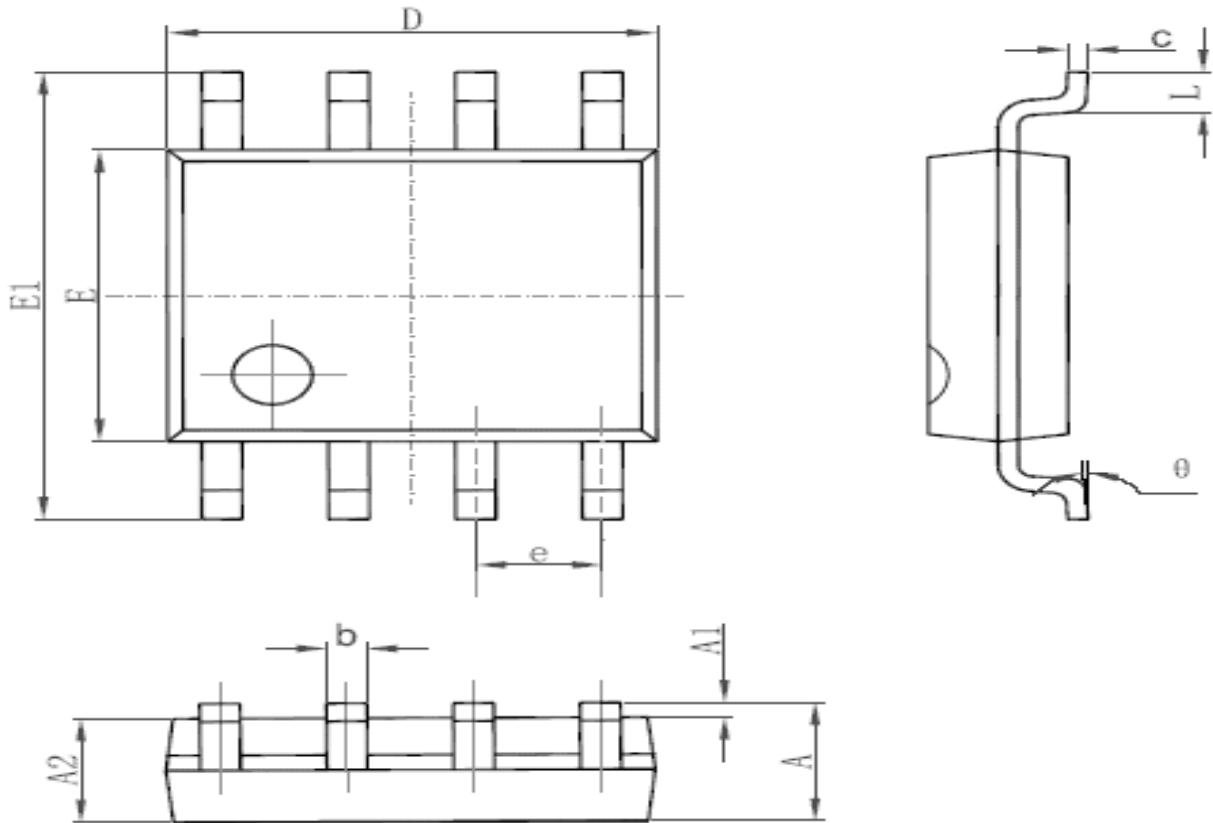




## 19. 芯片封装信息

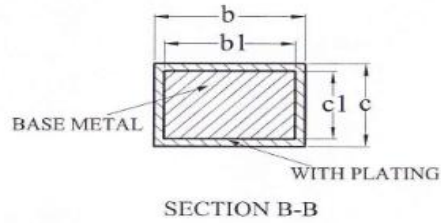
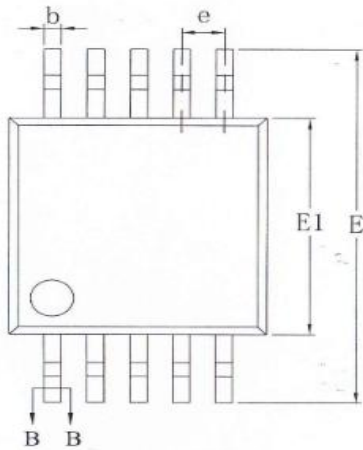
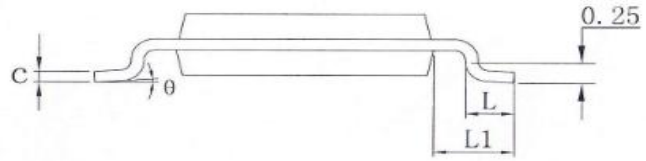
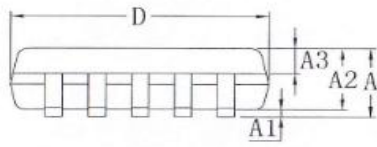
本芯片采用 SOP8、MSOP10、SOP14、SOP16、SOP20、TSSOP20、DIP20 和 QFN20 封装方式，具体封装尺寸信息如下。

SOP8:



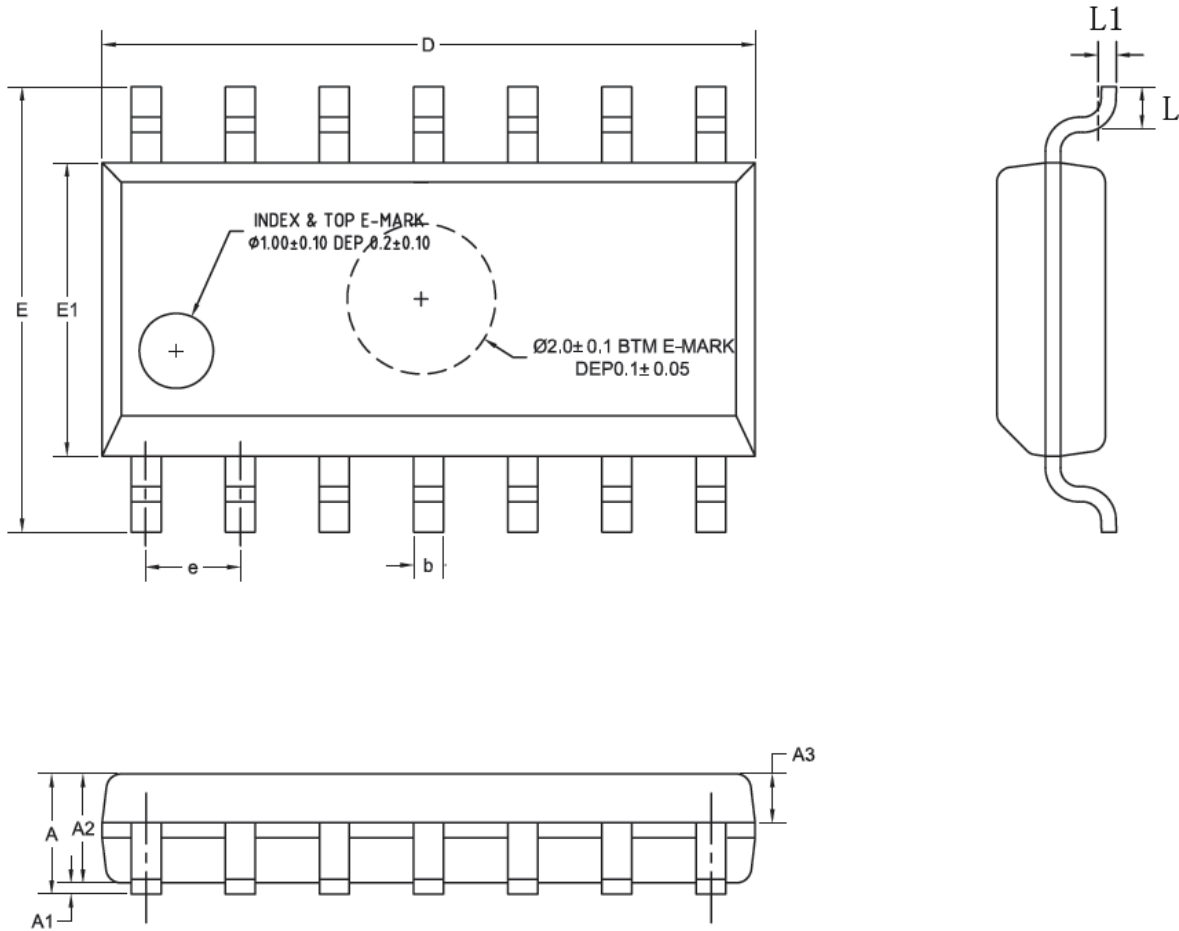
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	1.350	1.750	0.053	0.069
A1	0.100	0.250	0.004	0.010
A2	1.350	1.550	0.053	0.061
b	0.330	0.510	0.013	0.020
c	0.170	0.250	0.006	0.010
D	4.700	5.100	0.185	0.200
E	3.800	4.000	0.150	0.157
E1	5.800	6.200	0.228	0.244
e	1.270 (BSC)		0.050 (BSC)	
L	0.400	1.270	0.016	0.050
theta	0°	8°	0°	8°

MSOP10:



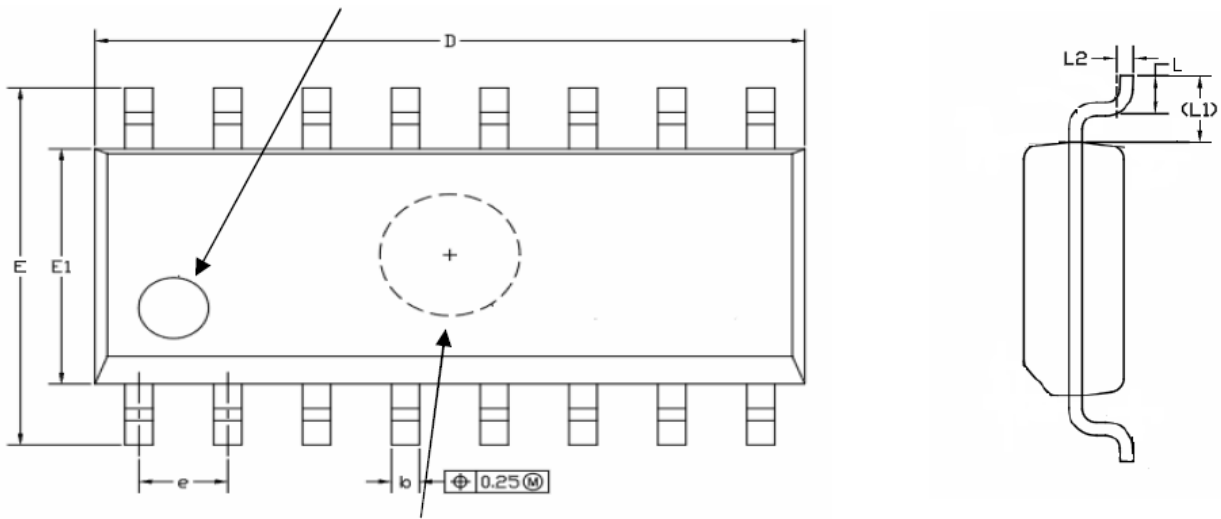
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	1.100	-	0.043
A1	0.050	0.150	0.002	0.006
A2	0.750	0.950	0.030	0.037
A3	0.300	0.400	0.012	0.016
b	0.180	0.260	0.007	0.010
b1	0.170	0.230	0.007	0.009
c	0.150	0.190	0.006	0.007
c1	0.140	0.160	0.006	0.006
D	2.900	3.100	0.114	0.122
E	4.700	5.100	0.185	0.201
E1	2.900	3.100	0.114	0.122
e	0.500(BSC)		0.020(BSC)	
L	0.400	0.700	0.016	0.028
L1	0.950(REF)		0.037(REF)	
theta	0	8°	0	8°

SOP14:

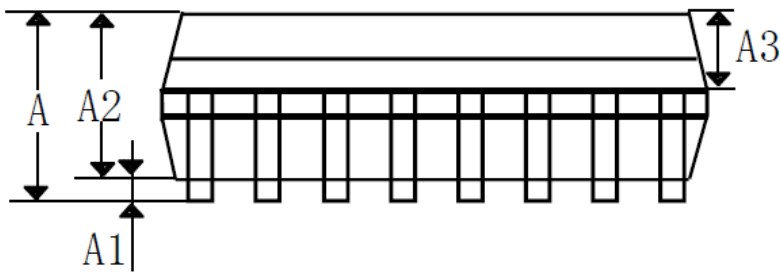


Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	1.700	-	0.066
A1	0.100	0.200	0.004	0.008
A2	1.400	1.500	0.055	0.059
A3	0.620	0.680	0.024	0.027
b	0.370	0.420	0.015	0.016
D	8.710	8.910	0.343	0.347
E	5.900	6.100	0.232	0.238
E1	3.800	3.950	0.150	0.156
e	1.270(BSC)		0.050(BSC)	
L	0.500	0.700	0.020	0.027
L1	0.250(BSC)		0.010(BSC)	

SOP16:

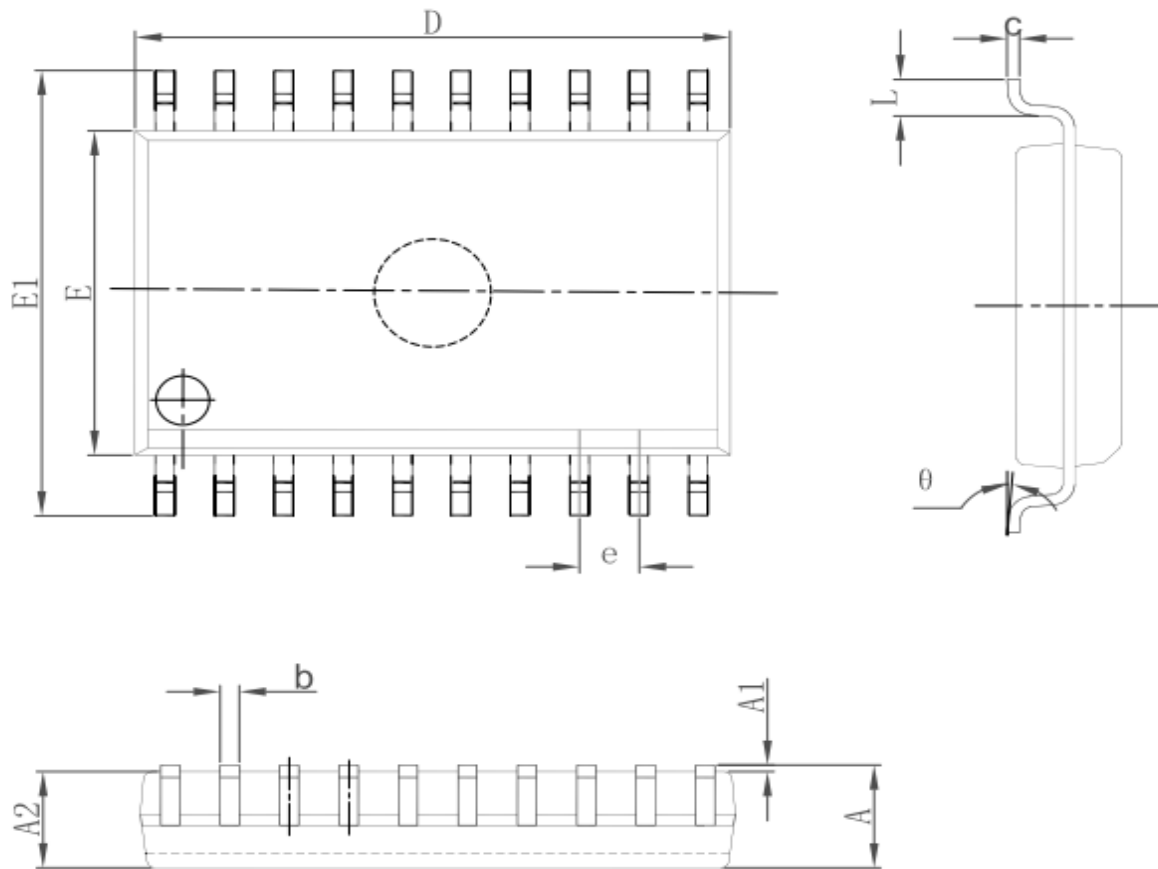


$\Phi 2.0 \pm 0.05$  DEP  $0.1 + 0.03 / -0.05$



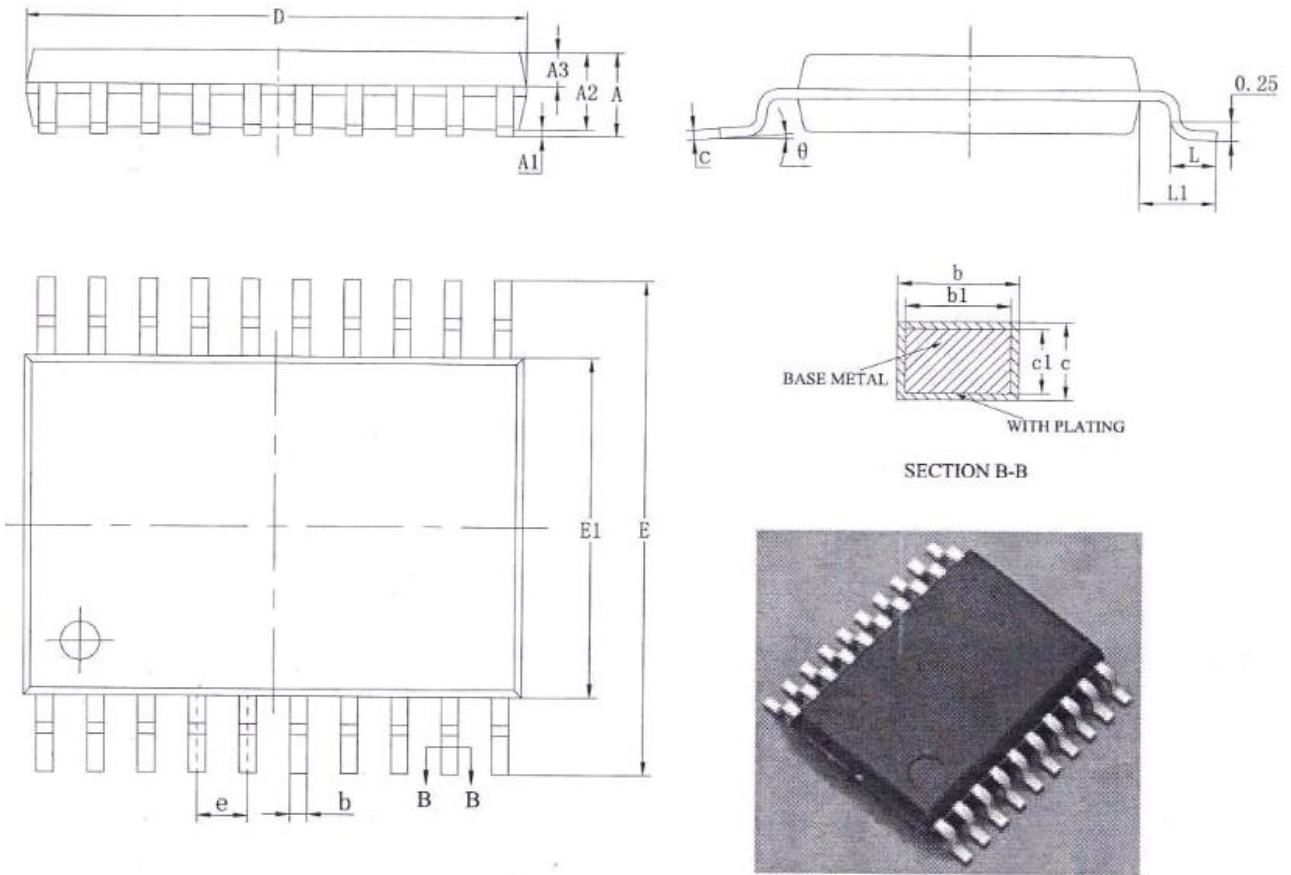
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	1.700	-	0.066
A1	0.100	0.200	0.004	0.008
A2	1.420	1.480	0.056	0.058
A3	0.620	0.680	0.024	0.027
D	9.960	10.160	0.392	0.396
E	5.900	6.100	0.232	0.238
E1	3.870	3.930	0.152	0.153
b	0.370	0.430	0.015	0.017
e	1.240	1.300	0.048	0.051
L	0.500	0.700	0.020	0.027
L1	1.050(REF)		0.041(REF)	
L2	0.250(BSC)		0.010(BSC)	

SOP20:



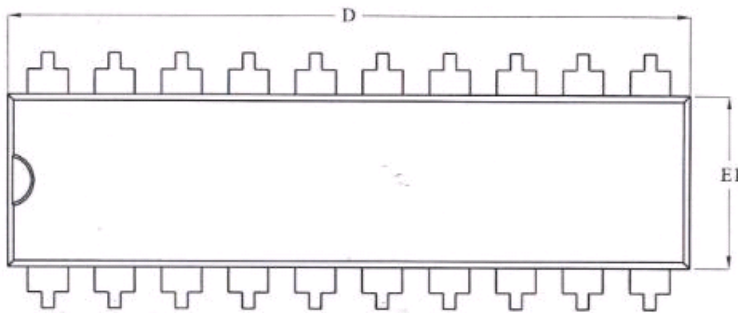
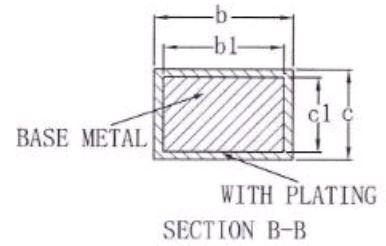
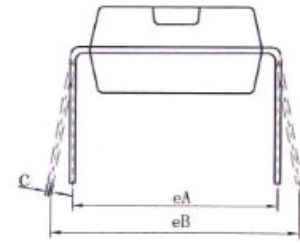
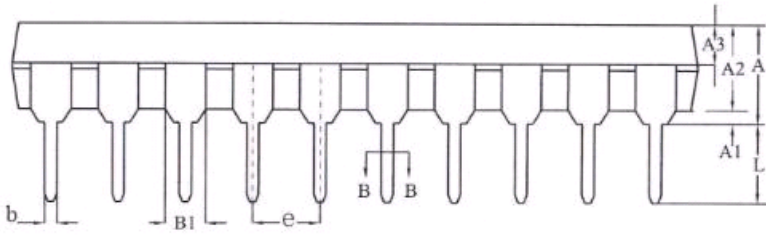
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	2.350	2.650	0.093	0.104
A1	0.100	0.300	0.004	0.012
A2	2.100	2.500	0.083	0.098
b	0.330	0.510	0.013	0.020
c	0.204	0.330	0.008	0.013
D	12.520	13.000	0.493	0.512
E	7.400	7.600	0.291	0.299
E1	10.210	10.610	0.402	0.418
e	1.270 (BSC)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

TSSOP20:



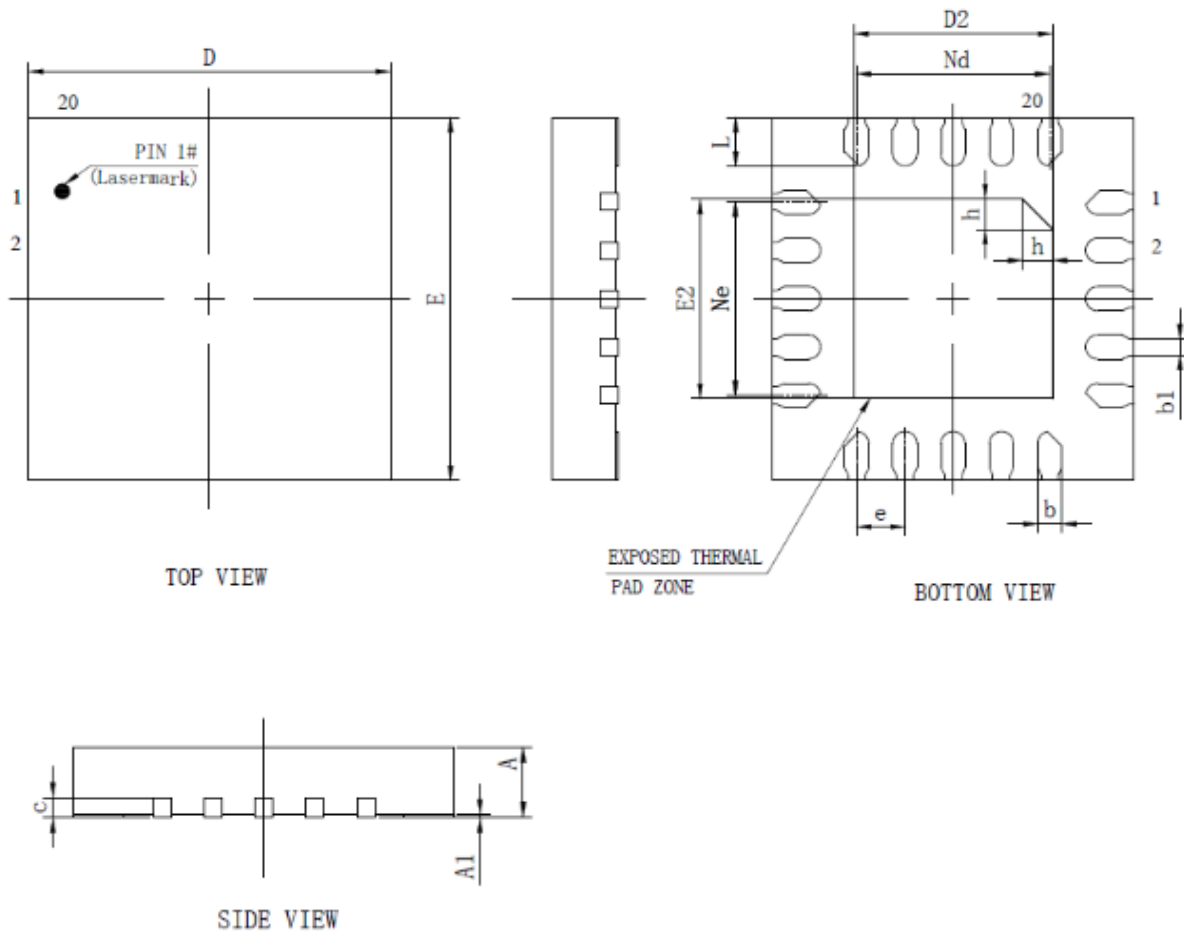
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	1.20	-	0.047
A1	0.05	0.15	0.002	0.006
A2	0.80	1.05	0.031	0.041
A3	0.39	0.49	0.015	0.019
b	0.20	0.28	0.008	0.011
b1	0.19	0.25	0.008	0.010
c	0.13	0.17	0.005	0.007
c1	0.12	0.14	0.005	0.006
D	6.40	6.60	0.252	0.260
E1	4.30	4.50	0.169	0.177
E	6.20	6.60	0.244	0.260
e	0.65(BSC)		0.026(BSC)	
L	0.45	0.75	0.018	0.030
L1	1.00REF		0.040REF	
θ	0	8°	0	8°

DIP20:



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	3.60	4.00	0.142	0.157
A1	0.51	—	0.020	—
A2	3.20	3.40	0.126	0.134
A3	1.47	1.57	0.058	0.062
b	0.44	0.52	0.017	0.020
b1	0.43	0.49	0.017	0.019
B1	1.52REF		0.060REF	
c	0.25	0.29	0.010	0.011
c1	0.24	0.26	0.009	0.010
D	25.80	26.00	1.016	1.024
E1	6.45	6.65	0.253	0.262
e	2.54BSC		0.1BSC	
eA	7.62REF		0.3REF	
eB	7.62	9.30	0.3	0.366
eC	0	0.84	0	0.033
L	3.00	—	0.118	—

QFN20:



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



## 附录 1， 文档更改历史

日期	版本	内容
2020-3-23	1.00	初版

## X-ON Electronics

Largest Supplier of Electrical and Electronic Components

*Click to view similar products for [fremont micro devices](#) manufacturer:*

Other Similar products are found below :

[FT64F0A5-TRB](#) [FT60F123](#) [FT61F023-RB](#) [FT60F121](#) [FT61F141-RB](#) [FT61F0A3-RB](#) [FT24C64A-ENR-T](#) [FT24C32A-ETR-B](#) [FT24C02A-ETR-B](#) [FT825-12S480MA](#) [FT60F021-RB](#) [FT24C16A-ETR-B](#) [FT60F022-RB](#) [FT61F143-RB](#) [FT441AA](#) [FT838R-BL-RT](#) [FT61F132A-RB](#) [FT61F021A-RB](#) [FT24C128A-ESR-T](#) [FT60F122](#) [FT24C128A-ENR-T](#) [FT24C32A-ESR-B](#) [FT61F14F-MRB](#) [FT24C512A-ETR-B](#) [FT24C64A-ETR-B](#) [FT24C128A-ETR-B](#) [FT64F0A3-RB](#) [FT24C256A-ETR-T](#) [FT24C08A-ESR-T](#) [FT24C02A-ETR-T](#) [FT60F112](#) [FT838NB1-RT](#) [FT61F142-RB](#) [FT62F086-TRB](#) [FT24C512A-ESR-B](#) [FT93C46A-USG-T](#) [FT62F132-RB](#) [FT62F13F-MRB](#) [FT24C64A-ESR-B](#) [FT838NB2-RT](#) [FT838D1-BL-RT](#) [FT24C08A-ETR-T](#) [FT24C08A-UPR-T](#) [FT24C02A-ESR-T](#) [FT24C16A-UTR-T](#) [FT24C02A-ESR-B](#) [FT838D2-BL-RT](#) [FT24C16A-ETR-T](#) [FT24C32A-ESR-T](#) [FT24C128A-ESR-B](#)