

## 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）
- 指令周期：4T
  - ✓ 250ns @ 4T, 16MHz, VDD≥2.7V
- 8 层硬件堆栈
- 存储架构
  - ✓ 程序 ROM: 1k x 14bits
  - ✓ 数据 RAM: 64 x 8bits
  - ✓ 数据 EEPROM: 128 x 8bits

## 特殊单片机特性

- 时钟
  - ✓ 两个内部时钟
  - ✓ 16MHz 高速高精度 HIRC
  - ✓ 32kHz 低速低功耗 LIRC
  - ✓ 外部时钟输入, max. 20MHz
  - ✓ 系统时钟源可选
- 带 7 位预分频的 16 位看门狗
- 上电延时计数器
- 低功耗模式 SLEEP
- 宽工作电压范围: **1.9V - 5.5V**
- 工业级温度范围: **-40 - 85°C**
- 最大时钟工作频率: **16MHz**
  - ✓ F<sub>SYS</sub> = 8MHz: 1.9V - 5.5V
  - ✓ F<sub>SYS</sub> = 16MHz: 2.7V - 5.5V
- 低电压复位 LVR:
  - ✓ 2.0/2.2/2.5/2.8/3.1/3.6/4.1V
- 低电压检测 LVD:
  - ✓ 1.2/1.8/2.0/2.4/2.7/3.0/3.3/3.6/4.0V
  - ✓ 可检测外部电压, 当比较器使用
  - ✓ 检测输出可作为 PWM 的刹车源
- 支持 ISP 和在线调试 OCD
  - ✓ 1 个硬件断点
  - ✓ 软复位, 单步, 暂停, 跳跃等
- 程序 ROM 保护功能
- 封装形式: **SOT23-6, SOP8**

## 外设特性

- **GPIO**
  - ✓ 最多 6 个通用 IO: PORTA
  - ✓ 端口变化中断, PA0 – PA5
  - ✓ 6 个带独立上拉功能的管脚
  - ✓ PA4/PA5 可直接驱动 LED, 节省限流电阻
  - ✓ 外部复位管脚: PA5
- **Timer0**
  - ✓ 带 8 位预分频的 8 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
  - ✓ 占空比和周期寄存器双缓冲
  - ✓ 时钟源: HIRC, HIRC 的 2 倍频, 外部时钟的 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	封装
FT60F210-URT	1024	128	64	4	8bit timer0 12bit timer2	-	SOT23-6
FT60F211-RB				6		-	SOP8
FT62F211-RB				5		√	

## 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 .....	8
1.1. 程序存储器 .....	9
1.2. 脚位图 .....	10
1.3. 管脚描述 .....	11
2. 特殊功能寄存器 .....	12
2.1. SFR, BANK0 .....	12
2.2. SFR, BANK1 .....	13
2.2.1. STATUS 寄存器, 地址 0x03, 0x83 .....	14
2.3. 配置寄存器 UCFGx .....	15
2.3.1. UCFG0, PROM 地址 0x400 .....	15
2.3.2. UCFG1, PROM 地址 0x401 .....	15
2.3.3. UCFG2, PROM 地址 0x402 .....	16
2.4. PCL 和 PCLATH .....	17
2.4.1. 修改 PCL .....	17
2.5. INDF 和 FSR 寄存器 .....	18
2.6. 关于寄存器保留位 .....	18
3. 系统时钟源 .....	19
3.1. 时钟源模式 .....	19
3.2. 内部时钟模式 .....	19
3.2.1. 频率选择位 (IRCF) .....	20
3.2.2. HIRC 和 LIRC 时钟切换时序 .....	20
3.2.3. 频率与最小工作电压的关系 .....	21
3.2.4. HIRC 时钟精度调节 .....	22
3.2.5. 复位或从休眠中唤醒 .....	22
3.3. 与时钟源相关寄存器汇总 .....	23
3.3.1. OSCCON 寄存器, 地址 0x8F .....	23
3.3.2. FOSCCAL 寄存器, 地址 0x0D .....	24
4. 复位时序 .....	25
4.1. POR 上电复位 .....	26
4.2. 外部复位 MCLR .....	26
4.3. PWRT (上电计时器) .....	26
4.4. BOR 低电压复位 .....	27
4.5. LVD 低电压侦测 .....	27
4.5.1. 检测外部电压 .....	27
4.6. 错误指令复位 .....	27
4.7. 超时动作 .....	28
4.7.1. PCON 寄存器 .....	28

4.8.	上电配置过程 .....	28
4.9.	上电校验过程 .....	28
4.10.	PCON 寄存器, 地址 0x8E .....	31
4.11.	LVDCON 寄存器, 地址 0x8D .....	32
5.	看门狗定时器 .....	33
5.1.	看门狗 .....	33
5.2.	看门狗时钟源 .....	34
5.3.	与看门狗相关寄存器汇总 .....	34
5.3.1.	WDTCON 寄存器, 地址 0x18 .....	35
5.3.2.	OPTION 寄存器, 地址 0x81 .....	35
6.	定时器 0 .....	36
6.1.	Timer0 定时器模式 .....	36
6.1.1.	Timer0 的时钟源 .....	37
6.1.2.	TMR0 寄存器的读写 .....	37
6.2.	Timer0 计数器模式 .....	37
6.2.1.	软件可配置预分频电路 .....	37
6.2.2.	定时器 0 中断 .....	38
6.2.3.	用外部时钟驱动定时器 0 .....	38
6.2.4.	睡眠模式下的状态 .....	38
6.3.	与 Timer0 相关寄存器汇总 .....	39
6.3.1.	OPTION 寄存器, 地址 0x81 .....	39
6.3.2.	TMR0, 地址 0x01 .....	40
6.3.3.	T0CON0, 地址 0x1F .....	40
7.	定时器 2 .....	41
7.1.	Timer2 工作原理 .....	42
7.2.	关于 PR2 的更新 .....	42
7.3.	Timer2 计数值的读写 .....	43
7.4.	Timer2 时钟源 .....	44
7.5.	与 Timer2 相关寄存器汇总 .....	44
7.5.1.	PR2 寄存器, 地址 0x91, 0x92 .....	44
7.5.2.	TMR2 寄存器, 地址 0x11, 0x13 .....	45
7.5.3.	T2CON0 寄存器, 地址 0x12 .....	46
7.5.4.	T2CON1 寄存器, 地址 0x9E .....	47
8.	慢时钟测量 .....	48
8.1.	测量原理 .....	48
8.2.	操作步骤 .....	49
8.3.	与慢时钟测量相关寄存器汇总 .....	49
8.3.1.	MSCON 寄存器, 地址 0x1B .....	50
8.3.2.	SOSCPR 寄存器, 地址 0x1C, 1D .....	51
9.	PWM1 模块 .....	52
9.1.	周期 .....	52
9.2.	占空比 .....	53
9.3.	时钟源选择 .....	53
9.4.	睡眠下 PWM 状态 .....	53

9.5.	P1A 的死区时间 .....	54
9.6.	故障刹车 .....	54
9.6.1.	刹车状态 .....	54
9.6.2.	故障清除 .....	55
9.6.3.	自动重启 .....	55
9.7.	关于周期和占空比寄存器的更新 .....	56
9.8.	蜂鸣器模式 (Buzzer) .....	57
9.9.	单脉冲输出 .....	57
9.10.	P1D、P1C 输出重映射 .....	57
9.11.	P1D 的第 2 功能输出 .....	58
9.12.	与 PWM1 相关寄存器汇总 .....	59
9.12.1.	P1ADTL 寄存器, 地址 0x0E .....	59
9.12.2.	P1BDTL 寄存器, 地址 0x0F .....	60
9.12.3.	P1CDTL 寄存器, 地址 0x10 .....	60
9.12.4.	TMR2L 寄存器, 地址 0x11 .....	60
9.12.5.	TMR2H 寄存器, 地址 0x13 .....	60
9.12.6.	T2CON0 寄存器, 地址 0x12 .....	61
9.12.7.	P1ADTH 寄存器, 地址 0x14 .....	61
9.12.8.	P1BDTH 寄存器, 地址 0x15 .....	61
9.12.9.	P1CDTH 寄存器, 地址 0x1A .....	61
9.12.10.	P1DDTL 寄存器, 地址 0x8 .....	62
9.12.11.	P1DDTH 寄存器, 地址 0x9 .....	62
9.12.12.	P1CON 寄存器, 地址 0x16 .....	62
9.12.13.	P1BR0 寄存器, 地址 0x17 .....	63
9.12.14.	P1BR1 寄存器, 地址 0x19 .....	64
9.12.15.	P1OE 寄存器, 地址 0x90 .....	65
9.12.16.	PR2L 寄存器, 地址 0x91 .....	65
9.12.17.	PR2H 寄存器, 地址 0x92 .....	66
9.12.18.	P1POL 寄存器, 地址 0x99 .....	66
9.12.19.	P1AUX 寄存器, 地址 0x1E .....	67
10.	中断处理 .....	68
10.1.	INT 外部中断 .....	69
10.2.	端口变化中断 .....	69
10.2.1.	PAIF 标志位的清除 .....	69
10.3.	中断响应 .....	70
10.4.	中断过程中的现场保存 .....	70
10.5.	关于中断标志位 .....	70
10.6.	与中断相关寄存器汇总 .....	70
10.6.1.	INTCON 寄存器, 地址 0x0B/0x8B .....	71
10.6.2.	PIR1 寄存器, 地址 0x0C .....	72
10.6.3.	PIE1 寄存器, 地址 0x8C .....	73
10.6.4.	IOCA 寄存器, 地址 0x96 .....	73
11.	睡眠模式 .....	74
11.1.	唤醒模式 .....	74

11.2.	看门狗唤醒 .....	74
12.	I/O 端口 .....	75
12.1.	PORTx 端口和 TRISx 寄存器 .....	75
12.2.	端口的其他功能 .....	75
12.2.1.	弱上拉 .....	75
12.3.	与 GPIO 相关寄存器汇总 .....	76
12.3.1.	WPUA, 地址 0x95 .....	76
12.3.2.	TRISA, 地址 0x85 .....	76
12.3.3.	PORTA, 地址 0x05 .....	77
12.3.4.	PSRCA, 地址 0x88 .....	77
12.3.5.	PSINKA, 地址 0x97 .....	77
13.	数据 EEPROM .....	78
13.1.	编程数据 EEPROM 步骤 .....	78
13.2.	读数据 EEPROM .....	79
13.3.	关于编程周期 .....	79
13.4.	与数据 EEPROM 相关寄存器汇总 .....	80
13.4.1.	EEDAT 寄存器, 地址 0x9A .....	80
13.4.2.	EEADR 寄存器, 地址 0x9B .....	80
13.4.3.	EECON1 寄存器, 地址 0x9C .....	81
13.4.4.	EECON2 寄存器, 地址 0x9D .....	81
14.	TOUCH 模块 .....	82
14.1.	触摸按键功能 .....	82
14.2.	触摸按键结构 .....	82
14.3.	TKCON 寄存器, 地址 0x86 .....	83
15.	芯片的电气特性 .....	84
15.1.	极限参数 .....	84
15.2.	内置高频振荡器 (HIRC) .....	84
15.3.	内置低频振荡器 (LIRC) .....	84
15.4.	低电压复位电路 (LVR) .....	85
15.5.	低电压侦测电路 (LVD) .....	85
15.6.	上电复位电路 (POR) .....	85
15.7.	I/O PAD 电路 .....	86
15.8.	总体工作电流 ( $I_{DD}$ ) .....	86
15.9.	AC 电气参数 .....	87
15.10.	直流和交流特性曲线图 .....	87
15.10.1.	不同 $V_{DD}$ 下, $I_{DD}$ vs Freq ( $T_A=25^\circ\text{C}$ ) .....	87
15.10.2.	不同 $V_{DD}$ 下, $I_{SB}$ (睡眠电流) 随温度变化曲线 .....	88
15.10.3.	HIRC vs $V_{DD}$ ( $T_A=25^\circ\text{C}$ ) .....	88
15.10.4.	LIRC vs $V_{DD}$ ( $T_A=25^\circ\text{C}$ ) .....	89
15.10.5.	$I_{OH}$ ( level -3mA ) vs $V_{OH}$ @ $V_{DD}=5V$ .....	89
15.10.6.	$I_{OH}$ ( level -6mA ) vs $V_{OH}$ @ $V_{DD}=5V$ .....	90
15.10.7.	$I_{OH}$ ( level -24mA ) vs $V_{OH}$ @ $V_{DD}=5V$ .....	90
15.10.8.	$I_{OH}$ ( level -18mA ) vs $V_{OH}$ @ $V_{DD}=5V$ .....	91
15.10.9.	$I_{OL}$ ( L0 53mA ) vs $V_{OL}$ @ $V_{DD}=5V$ .....	91

15.10.10.	$I_{OL}$ ( L1 67mA ) vs $V_{OL}$ @ $V_{DD}=5V$ .....	92
16.	指令集列表.....	93
17.	芯片封装信息.....	95
附录 1,	文档更改历史.....	97

### 1. 系统功能框图及脚位

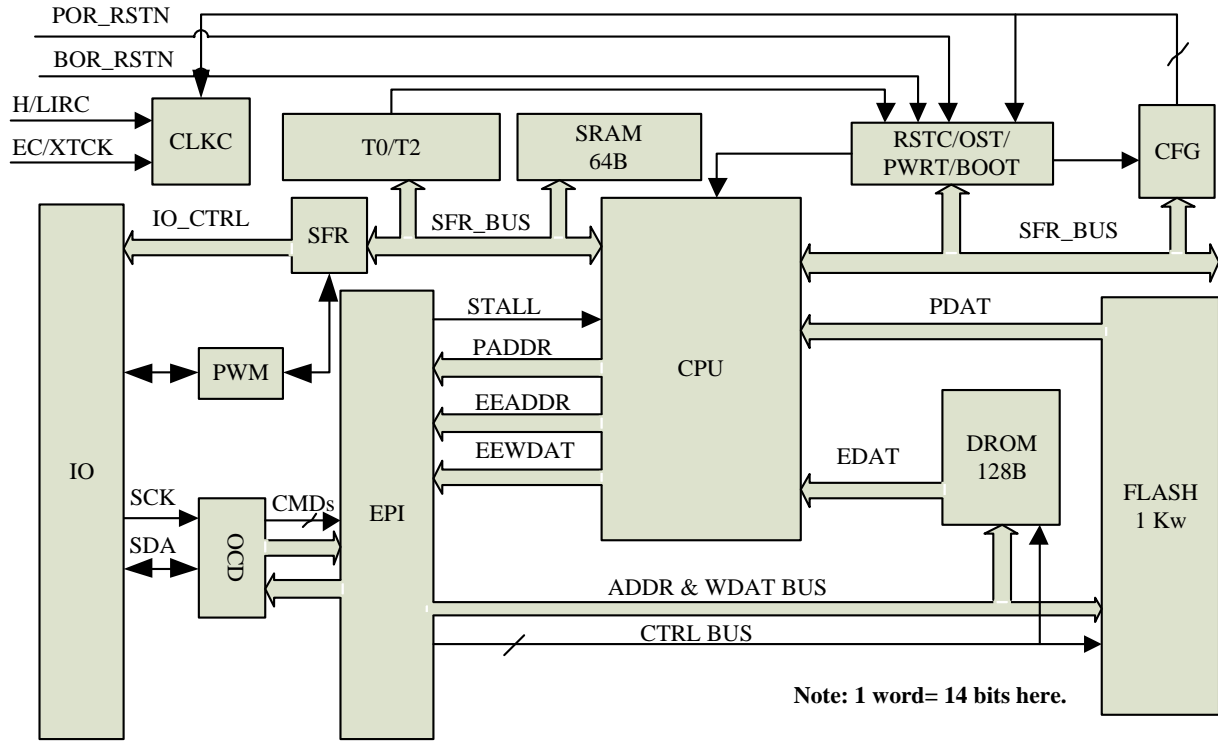


图 1.1 芯片整体功能框图



## 1.1. 程序存储器

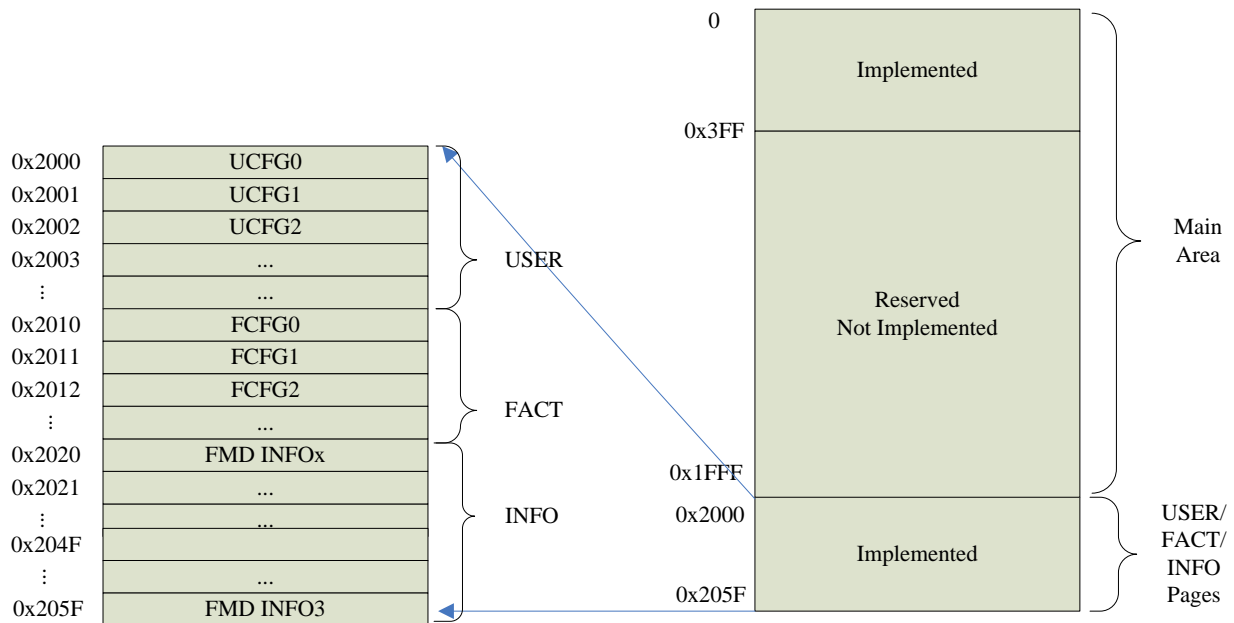


图 1.2 程序空间地址映射

程序存储器的地址为 13 位 (0x0000 – 0x1FFF)，最多支持 8K 地址空间。

程序存储器由页组成，每页大小为 16 字（这里所提的字，位宽为 14bit）。FT60F21X 实现了 1K 的单元 (0x0000 – 0x03FF)，外加上 6 个额外的用户配置、工厂配置区，总共有 1K+64 个字，即整个程序 ROM 一共有 70 页。

地址空间的分配如下：

0–0x03FF，对主程序区访问

0x400–0x1FFF，未实现，保留

0x2000–0x205F，用户和工厂配置信息区

## 1.2. 脚位图



图 1.3 SOT23-6 脚位

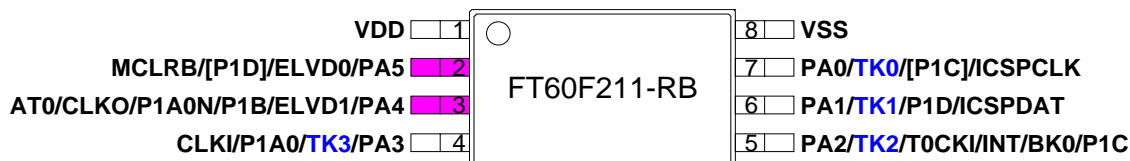
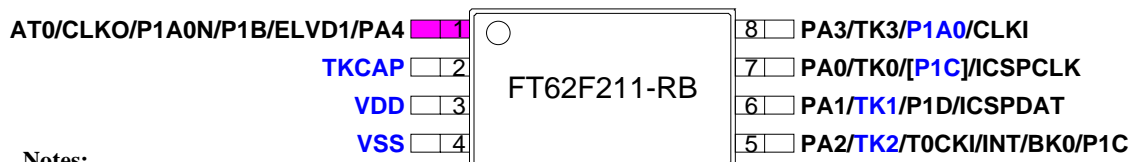


图 1.4 SOP8 脚位 A



**Notes:**

1. 源电流设置:

<span style="border: 1px solid black; display: inline-block; width: 15px; height: 10px;"></span>	18mA
<span style="background-color: pink; border: 1px solid black; display: inline-block; width: 15px; height: 10px;"></span>	3/6/24 mA

图 1.5 SOP8 脚位 B

**注意:**

1. 用[]括起来的是指可重映射的功能管脚;
2. 4路独立 PWM 是 P1A, P1B, P1C, P1D 其中 P1A 带互补及死区输出 (P1A0/0N 是 1 对), 极性可独立设置;

### 1.3. 管脚描述

管脚名	功能名	输入信号类型	输出信号类型	具体描述	上拉
<b>VDD</b>	Supply	—	—	正电源输入	—
<b>PA5/MCLR/ELVD/[P1D]</b>	PA5	CMOS	CMOS	PORTA IO, IOC	√
	MCLR	CMOS	—	外部复位输入	PU
	ELVD	AN	—	外部 LVD 输入	—
	[P1D]	—	CMOS	PWMD 第 2 映射输出	—
<b>PA4/P1B/P1A0N/ELVD1/AT0</b>	PA4	CMOS	CMOS	PORTC IO	√
	AT0	—	AN	TEST 测试功能输出	—
	P1A0N	—	CMOS	PWMA1 输出	—
	P1B	—	CMOS	PWMB 输出	—
	ELVD1	—	—	外部 LVD 输入	—
<b>PA3/TK3/P1A0/CLKI</b>	PA3	CMOS	CMOS	PORTA IO, IOC	√
	TK3	AN	AN	触摸按键 3	—
	P1A0	—	CMOS	PWMA0 输出	—
	CLKI	XTAL	—	外部时钟输入	—
<b>TKCAP</b>	TKCAP	AN	AN	触摸参考电容	—
<b>PA2/TK2/T0CKI/INT/BK0/P1C</b>	PA2	CMOS	CMOS	PORTA IO, IOC	√
	TK2	AN	AN	触摸按键 2	—
	T0CKI	CMOS	—	T0 外部时钟输入	√
	INT	CMOS	—	外部中断管脚输入	√
	BK0	CMOS	—	PWM 刹车输入	√
	P1C	—	CMOS	PWMC 输出	—
<b>PA1/TK1/P1D/LVR/ICSPDAT</b>	PA1	CMOS	CMOS	PORTA IO, IOC	√
	TK1	AN	AN	触摸按键 1	—
	P1D	—	CMOS	PWMD 输出	—
	LVR	—	CMOS	LVR 电平输出	—
	ICSPDAT	CMOS	CMOS	烧录调试数据脚	—
<b>PA0/TK0/[PIC]/ICSPCLK</b>	PA0	CMOS	CMOS	PORTA IO, IOC	√
	TK0	AN	AN	触摸按键 0	—
	ICSPCLK	CMOS	—	烧录调试时钟脚	—
	[PIC]	—	CMOS	PWMC 第 2 映射输出	—
<b>VSS</b>	Ground	—	—	负电源输入	—

管脚类型:

1. **CMOS:** CMOS 逻辑电平输入输出
2. **AN:** 模拟输入或输出

## 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 的内容对数据存储寄存器进行访问（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	—	PAGE	/TF	/PF	Z	HC	C		--01 1xxx	
4	FSR	间接寻址指针寄存器								xxxx xxxx	
5	PORTA	—	PA5	PA4	PA3	PA2	PA1	PA0		--xx xxxx	
6	—	—								---- ----	
7	—	—								---- ----	
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	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000	
C	PIR1	EEIF	CKMIF	LVDIF	—			TMR2IF	—	000- --0-	
D	FOSCCAL	—		FOSCCAL[5:0]						--xx xxxx	
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	LVDP	—	WCKSRC	WDTPS[3:0]			SWDTEN		0-00 1000	
19	P1BR1	P1DSS[1:0]		P1D2SS[1:0]		P1CALT	P1DALT	P1CSS[1:0]		0000 0000	
1A	P1CDTH	P1C 占空比寄存器高 8 位								0000 0000	
1B	MSCON	BGRBOE	LVROE	—		SLVREN	CKMAVG	CKCNTI	T2CKRUN	00-- 0000	
1C	SOSCPRL	SOSCPR [7:0]								1111 1111	
1D	SOSCPRH	—				SOSCPR [11:8]				---- 1111	
1E	P1AUX	—					P1DF2E		P1DF2		---- --00
1F	T0CON0	—			T0ON	T0CKRUN	T0CKSRC[1:0]			---- 1000	
20-3F	—									xxxx xxxx	
40-7F	SRAM BANK0, (64Bytes)									xxxx xxxx	

## 2.2. SFR, BANK1

ADDR	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR reset	
BANK1											
80	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	
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	—	PAGE	/TF	/PF	Z	HC	C		--01 1xxx	
84	FSR	间接寻址指针寄存器								xxxx xxxx	
85	TRISA	—	PORTA 方向控制							--11 1111	
86	TKCON	—	TKCHGS[1:0]	TKEN	TKCHE[3:0]					-000 0000	
87	—	—								---- ----	
88	PSRCA	—				PSRCA5[1:0]	PSRCA4[1:0]				---- 0000
89	—	—								---- ----	
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	—			TMR2IE	—	000- --0-	
8D	LVDCON	—					LVDDEB	LVDM[1:0]			---- -100
8E	PCON	LVDL[3:0]				LVDEN	LVDW	/POR	/BOR	0000 00qq	
8F	OSCCON	LFMOD	IRCF[2:0]			—	HTS	LTS	—	0101 -00-	
90	P1OE	P1COE	P1BOE	P1DOE	—	P1C2SS[1:0]		P1A0NOE	P1A0OE	000- 0000	
91	PR2L	PR2[7:0], Timer2 的周期设置寄存器, 低 8 位								1111 1111	
92	PR2H	PR2[15:8], Timer2 的周期设置寄存器, 高 8 位								1111 1111	
93	—	—								---- ----	
94	—	—								---- ----	
95	WPUA	—			WPUA[5:0]					--11 1111	
96	IOCA	—			IOCA[5:0]					--00 0000	
97	PSINKA	—						PSINKA[1:0]			---- --11
98	—	—								0000 0000	
99	P1POL	P1CP	P1BP	P1DP	—			P1A0NP	P1A0P	000- --00	
9A	EEDAT	EEDAT[7:0]								0000 0000	
9B	EEADR	EEADR[7:0]								0000 0000	
9C	EECON1	—		WREN3	WREN2	WRERR	WREN1	—	RD	--00 x0-0	
9D	EECON2	—							WR		---- --0
9E	T2CON1	—			P1OS	P1BZM	T2CKSRC[2:0]				---0 0000
9F	—	—								---- ----	
A0–EF	—								xxxx xxxx		
F0–FF	SRAM BANK1 (16Bytes), 访问的是 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;

### 2.2.1. STATUS 寄存器，地址 0x03, 0x83

Bit	7	6	5	4	3	2	1	0
Name	—	—	PAGE	/TF	/PF	Z	HC	C
Reset	—	—	0	1	1	x	x	x
Type	RO-0	RO-0	RW	RO	RO	RW	RW	RW

Bit	Name	Function
7:6	N/A	保留位，读 0
5	PAGE	寄存器存储区bank选择位 0 = Bank0 (0x00– 0x7F) 1 = Bank 1 (0x80– 0xFF)
4	/TF	超时标志位 1 = 在上电后，执行了CLRWDWT指令或SLEEP指令 0 = 发生了 WDT 超时
3	/PF	掉电标志位 1 = 上电复位后或执行了CLRWDWT指令 0 = 执行了 SLEEP 指令
2	Z	零状态位 1 = 算术运算或逻辑运算结果为零 0 = 算术运算或逻辑运算结果不为零
1	HC	半进位/ 半借位标志位 (ADDWR、ADDWI、SUBWI 和SUBWR指令) (1) 1 = 结果的第4个低位向高位发生了进位 0 = 结果的第 4 个低位未向高位发生进位
0	C	进位/借位标志位(1) (ADDWR、ADDWI、SUBWI 和SUBWR指令) (1) 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。

## 2.3. 配置寄存器 UCFGx

### 2.3.1. UCFG0, PROM 地址 0x400

Bit	7	6	5	4	3	2	1	0
Name	—	CPB	MCLRE	PWRTEB	WDTE	FOSC[2:0]		
POR val.	—	1	0	1	0	3'b111		

位	名称	描述
7	N/A	保留位，读 0
6	CPB	1: Flash 内容不保护 0: 启动 Flash 内容保护，MCU 能读，串口不能读 注意： 此位只能由 1 改写为 0，而不能由 0 改写为 1。由 0 改写成 1 的唯一方法是进行一次包括 USER_OPT 在内的片擦操作，并且重新上电后 CPB 才变为 1
5	MCLRE	1: PA5/MCLR 脚执行 MCLR 功能，是复位脚 0: PA5/MCLR 脚执行 GPIO
4	PWRTEB	1: PWRT 禁止 0: PWRT 使能
3	WDTE	1: WDT 使能，程序不能禁止 0: WDT 禁止，但程序可通过设置 WDTCON 的 SWDTEN 位将 WDT 使能
2:0	FOSC[2:0]	010: 外部时钟模式，PA3接时钟输入 其他值: INTOSCIO 模式，即使用内部

### 2.3.2. UCFG1, PROM 地址 0x401

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	RDCTRL	LVREN1	LVREN0
POR val.	—	—	—	—	—	0	1	1

位	名称	描述
7:3	N/A	保留位，读 0
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[1:0]	低电压复位选择 00: 使能低电压复位 01: LVR 由 MCON 的 SLVREN 决定 10: MCU 正常模式时开启 LVR，睡眠模式时关闭 LVR，跟 SLVREN 位无关 11: 禁止低电压复位

### 2.3.3. UCFG2, PROM 地址 0x402

Bit	7	6	5	4	3	2	1	0
Name	UDMY3	MBTEB	UDMY1	RBTEN	LVRS[3:0]			
POR val.	1	1	0	0	4'b0000			

位	名称	描述																										
7	UDMY3	冗余位																										
6	MBTEB	MCLR <sub>B</sub> 的 BOOT 控制 0 = 允许外部复位启动 BOOT 1 = 不允许外部复位启动 BOOT																										
5	UDMY1	冗余位																										
4	RBTEN	WDT 复位 BOOT 使能位 0 = 允许 WDT 复位产生 BOOT 1 = 禁止 WDT 复位产生 BOOT																										
3:0	LVRS[3:0]	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 rowspan="10">保留值</td></tr> <tr><td>1011</td></tr> <tr><td>1100</td></tr> <tr><td>1101</td></tr> <tr><td>1110</td></tr> <tr><td>1111</td></tr> <tr><td>0000</td></tr> <tr><td>0001</td></tr> <tr><td>0010</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	保留值	1011	1100	1101	1110	1111	0000	0001	0010	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	保留值																									
		1011																										
		1100																										
		1101																										
		1110																										
		1111																										
		0000																										
		0001																										
		0010																										
		0011		2.0V																								
		0100	2.2V																									
		0101	2.5V																									
		0110	2.8V																									
0111	3.1V																											
1000	3.6V																											
1001	4.1V																											



## 2.4. PCL 和 PCLATH

程序计数器 (PC) 为 11 位宽。其低 8 位来自可读写的 PCL 寄存器，高 3 位 (PC<10:8>) 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。

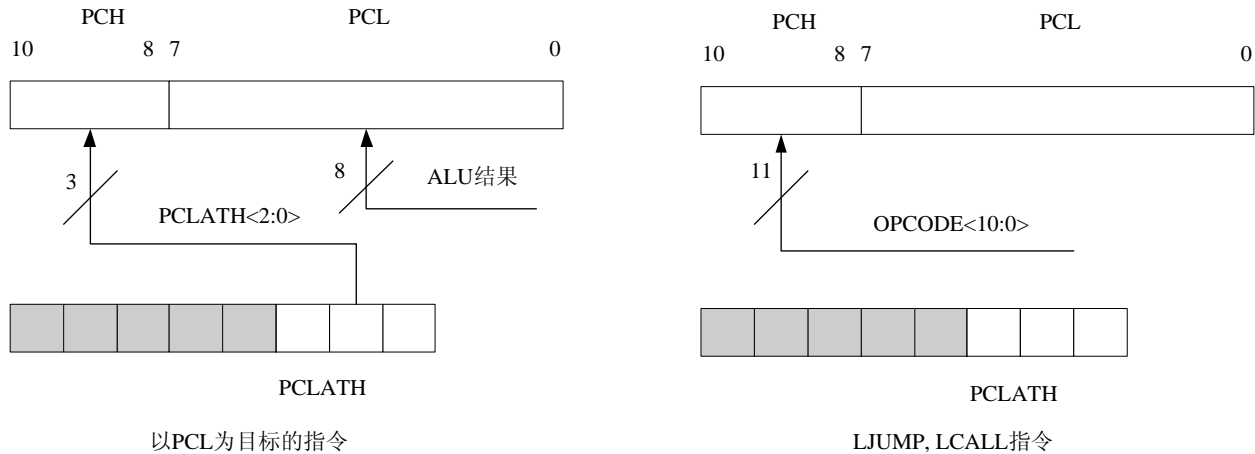


图 2.4.1 不同情况下 PC 的加载

### 2.4.1. 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。

## 2.5. 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）。

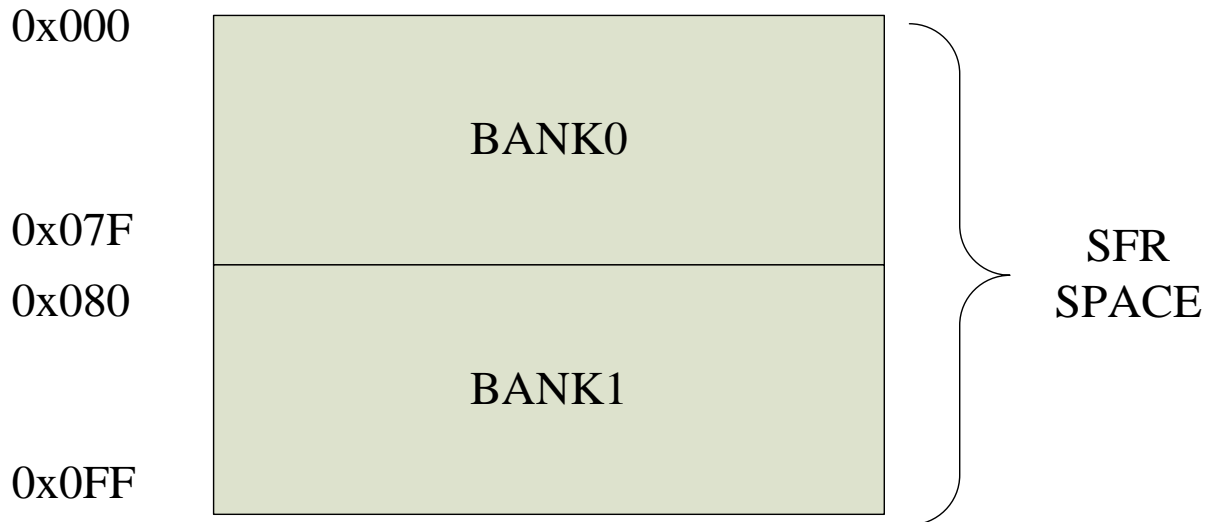


图 2.5.1 间接寻址

## 2.6. 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

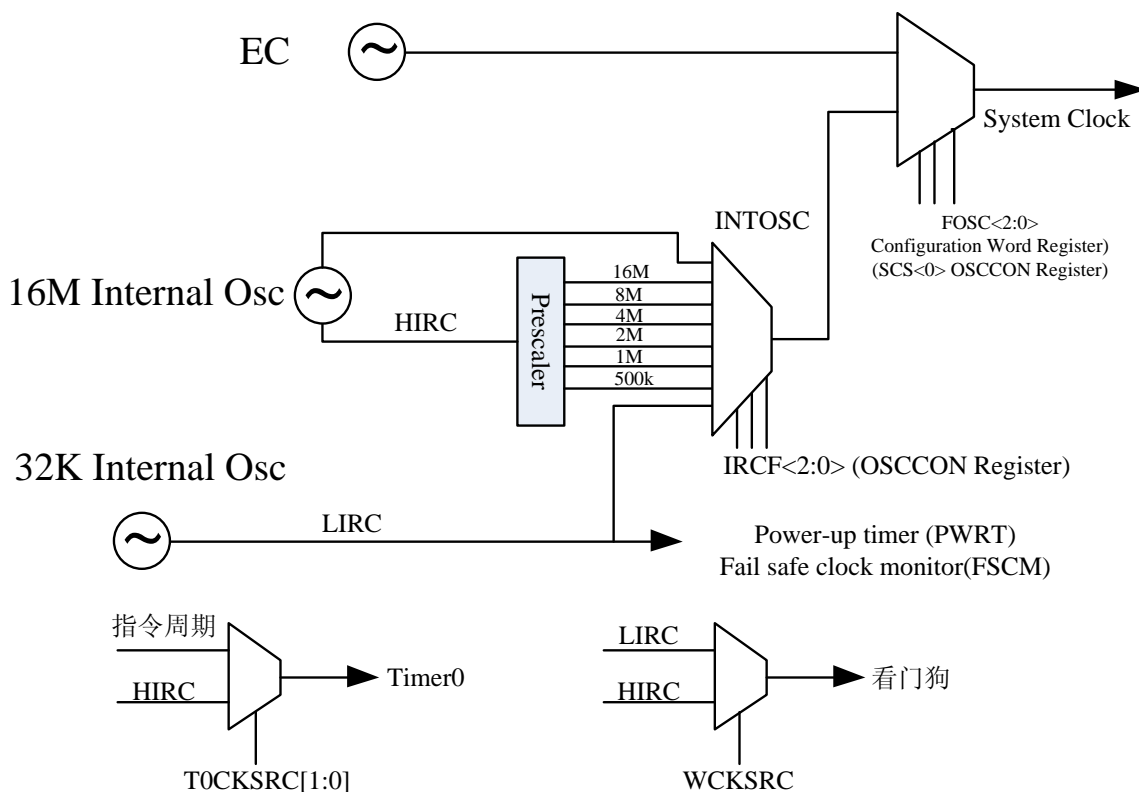


图 3.1 系统时钟源框图

本芯片包含 3 个时钟源：2 个内置 RC 振荡器（高、低速），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

- 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。

#### 3.2. 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。

### 3.2.1. 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

### 3.2.2. 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

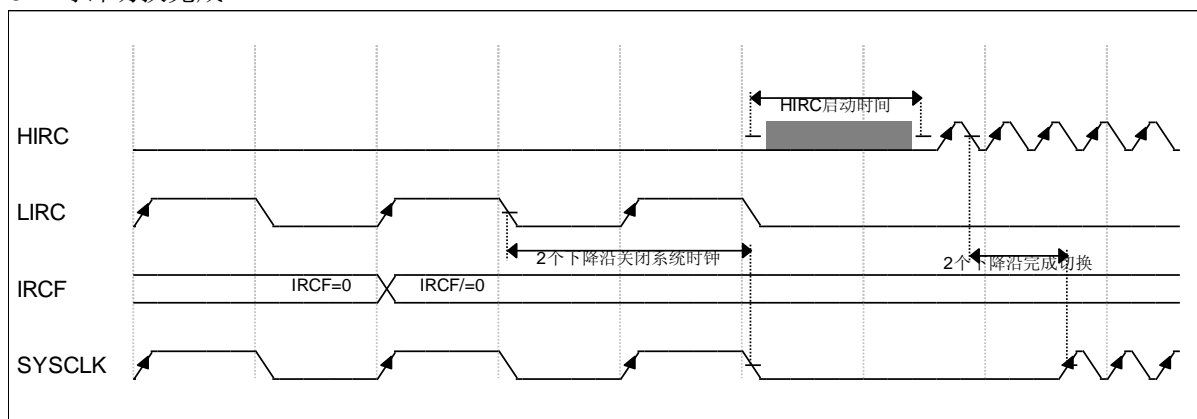


图 3.2 由慢时钟切换到快时钟

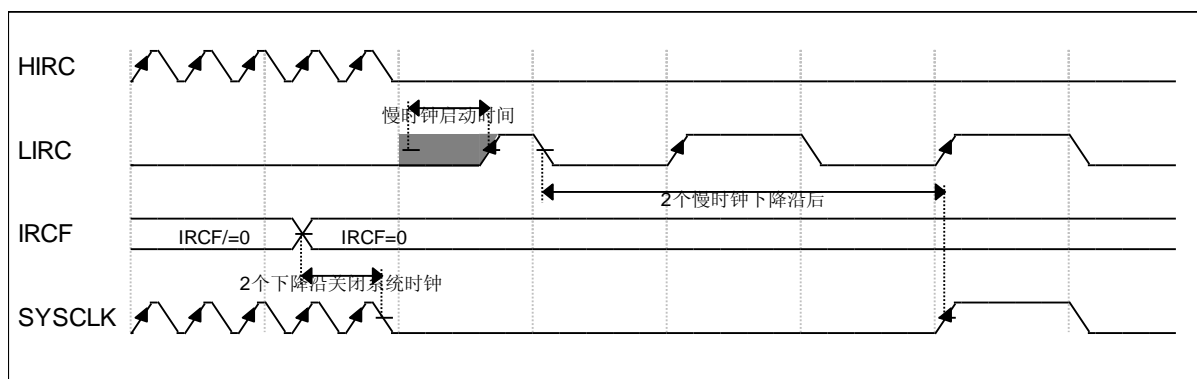


图 3.3 由快时钟切换到慢时钟

### 3.2.3. 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，如果系统时钟要跑 16M，VDD 应至少在 2.7V 以上。

FT62F21X 系列芯片的上电复位电压  $V_{POR}$  在 1.6V 左右，即当上电过程 VDD 超过  $V_{POR}$  之后复位释放，再经过约 4ms 延时之后完成上电配置，然后开始执行程序指令。对于一些慢上电且要跑 16M 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。

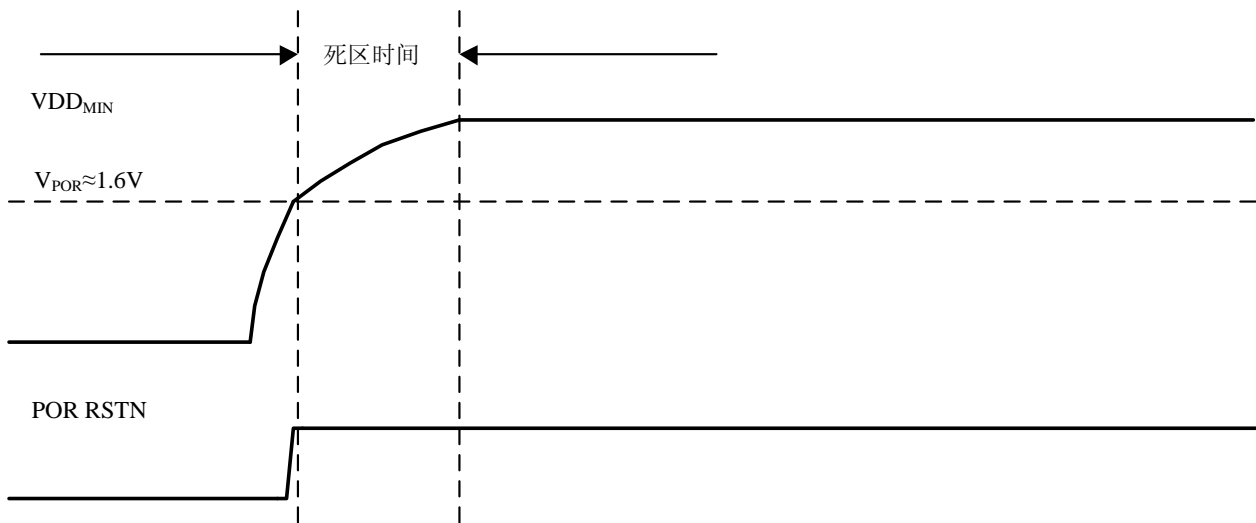


图 3.4 上电过程的“死区时间”

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，比如系统时钟跑 16M 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。

### 3.2.4. HIRC 时钟精度调节

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 130kHz。FOSCCAL[5:0] 和 HIRC 输出的关系如下：

FOSCCAL[5:0]值	HIRC 实际输出频率（以 16M 为例）
FOSCCAL[s]-n	(16000-n*130)
.....	.....
FOSCCAL[s]-2	16000-2*130=15740
FOSCCAL[s]-1	16000-1*130=15870
FOSCCAL[s]	16000
FOSCCAL[s]+1	16000+1*130=16130
FOSCCAL[s]+2	16000+2*130=16260
.....	.....
FOSCCAL[s]+n	(16000+n*130)

### 3.2.5. 复位或从休眠中唤醒

FSCM 设计为能在振荡器起振延时定时器（OST）到期后的任一时刻检测振荡器故障。OST 的使用场合为从休眠状态唤醒后以及任何类型的复位后。OST 不能在 EC 时钟模式下使用，所以一旦复位或唤醒完成，FSCM 就处于激活状态。当 FSCM 被使能时，双速启动也被使能。因此，当 OST 运行时，器件总是处于代码执行阶段。

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

### 3.3. 与时钟源相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
UCFG0	0x2000	—	CPB	MCLRE	PWRTEB	WDTE	FOSC2	FOSC1	FOSC0	-qqq qqqq
OSCCON	0x8F	LFMOD	IRCF2	IRCF1	IRCF0	—	HTS	LTS	—	0101 -00-
FOSCCAL	0x0D	—			HIRC 修调位					--xx xxxx

#### 3.3.1. OSCCON 寄存器，地址 0x8F

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

Bit	Name	Function																		
7	LFMOD	当 Timer2 时钟源选择 LIRC 时 (T2CKSRC=101)，LIRC 的频率选择 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"> <thead> <tr> <th>值</th> <th>4T 模式</th> </tr> </thead> <tbody> <tr> <td>111</td> <td>16MHz</td> </tr> <tr> <td>110</td> <td>8MHz</td> </tr> <tr> <td>101</td> <td>4MHz(default)</td> </tr> <tr> <td>100</td> <td>2MHz</td> </tr> <tr> <td>011</td> <td>1MHz</td> </tr> <tr> <td>010</td> <td>500kHz</td> </tr> <tr> <td>001</td> <td>250kHz</td> </tr> <tr> <td>000</td> <td>32kHz(LIRC)</td> </tr> </tbody> </table>	值	4T 模式	111	16MHz	110	8MHz	101	4MHz(default)	100	2MHz	011	1MHz	010	500kHz	001	250kHz	000	32kHz(LIRC)
值	4T 模式																			
111	16MHz																			
110	8MHz																			
101	4MHz(default)																			
100	2MHz																			
011	1MHz																			
010	500kHz																			
001	250kHz																			
000	32kHz(LIRC)																			
3	—	—																		
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	—	—																		

### 3.3.2. FOSCCAL 寄存器，地址 0x0D

Bit	7	6	5	4	3	2	1	0
Name	—	—	FOSCCAL					
Reset	—	—	0	1	1	1	1	1
TYPE	RO-0	RO-0	RW	RW	RW	RW	RW	RW

Bit	Name	Function																				
7:6	N/A	保留位，读 0																				
5:0	FOSCCAL	片内 HIRC 时钟校准位																				
		<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">FOSCCAL[5:0]值</td> <td style="width: 50%;">HIRC 实际输出频率, kHz</td> </tr> <tr> <td>FOSCCAL[s]-n</td> <td>(16000-n*130)</td> </tr> <tr> <td>.....</td> <td>.....</td> </tr> <tr> <td>FOSCCAL[s]-2</td> <td>1600 -2*130=15740</td> </tr> <tr> <td>FOSCCAL[s]-1</td> <td>16000-1*130=15870</td> </tr> <tr> <td>FOSCCAL[s]</td> <td>16000</td> </tr> <tr> <td>FOSCCAL[s]+</td> <td>16000+1*130=16130</td> </tr> <tr> <td>FOSCCAL[s]+2</td> <td>16000+2*130=16260</td> </tr> <tr> <td>.....</td> <td>.....</td> </tr> <tr> <td>FOSCCAL[s] n</td> <td>(16000+n*130)</td> </tr> </table>	FOSCCAL[5:0]值	HIRC 实际输出频率, kHz	FOSCCAL[s]-n	(16000-n*130)	.....	.....	FOSCCAL[s]-2	1600 -2*130=15740	FOSCCAL[s]-1	16000-1*130=15870	FOSCCAL[s]	16000	FOSCCAL[s]+	16000+1*130=16130	FOSCCAL[s]+2	16000+2*130=16260	.....	.....	FOSCCAL[s] n	(16000+n*130)
		FOSCCAL[5:0]值	HIRC 实际输出频率, kHz																			
		FOSCCAL[s]-n	(16000-n*130)																			
		.....	.....																			
		FOSCCAL[s]-2	1600 -2*130=15740																			
		FOSCCAL[s]-1	16000-1*130=15870																			
		FOSCCAL[s]	16000																			
		FOSCCAL[s]+	16000+1*130=16130																			
		FOSCCAL[s]+2	16000+2*130=16260																			
.....	.....																					
FOSCCAL[s] n	(16000+n*130)																					



## 4. 复位时序

FT62F21x 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 管脚复位 – 在常规运行期间
- D) /MCLR 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。/TO 和/PD 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 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。

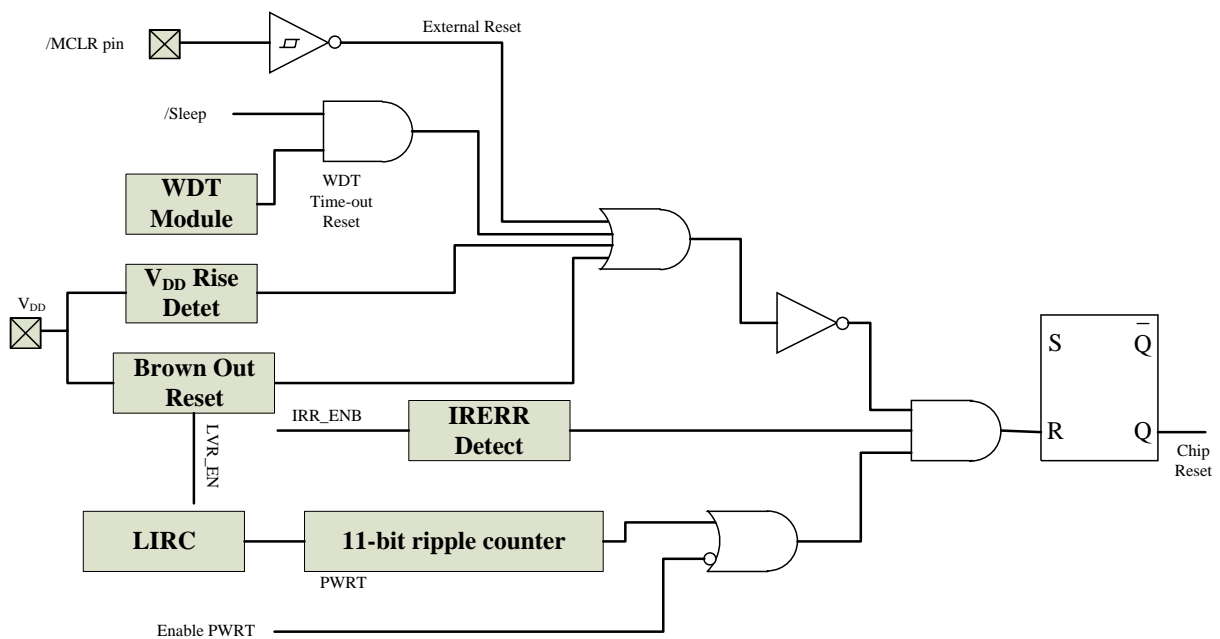


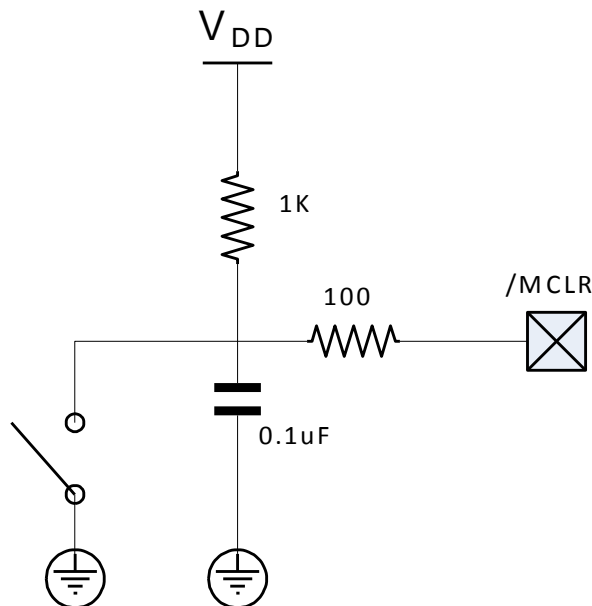
图 4.1 复位功能框图

## 4.1. POR 上电复位

片上的 POR 电路会将芯片保持在复位状态直到 VDD 电源电压达到足够高。上电复位后，系统复位不会立即释放，还要等一个约 4ms 的延时，期间数字电路保持在复位状态。

## 4.2. 外部复位 MCLR

需要注意的是，WDT 复位不会把/MCLR 管脚拉低。在/MCLR 管脚上施加超过指标的电压（例如 ESD 事件）会造成/MCLR 复位，而且在管脚上产生超标的大电流，因此我们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。



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 PA5/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  $MSCON$  的  $SLVREN$  位来控制。低电压复位就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，当它为 00 时表示对外部管脚  $ELVD$  进行监控，此时，该管脚（PA5）的斯密特输入被关闭以防漏电，当它为 01 时表示对外部管脚  $ELVD1$  进行监控，此时，该管脚（PA4）的斯密特输入被关闭以防漏电。

注意：

1. PA5 的外部复位功能优先级高于外部 LVD 功能，换言之，当配置为外部复位管脚时，外部 LVD 的检测是无效的；
2. 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 4ms 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 4ms 复位延时外，还有一个初始化配置寄存器 UCFGx 的动作。该动作从 PROM 的保留地址读取内容写到 UCFGx，待所有配置地址读取完成后，才可以释放系统复位，如图 4.2 和图 4.3 的所示，该过程大概需要 17us。

## 4.9. 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，1K 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 1ms。

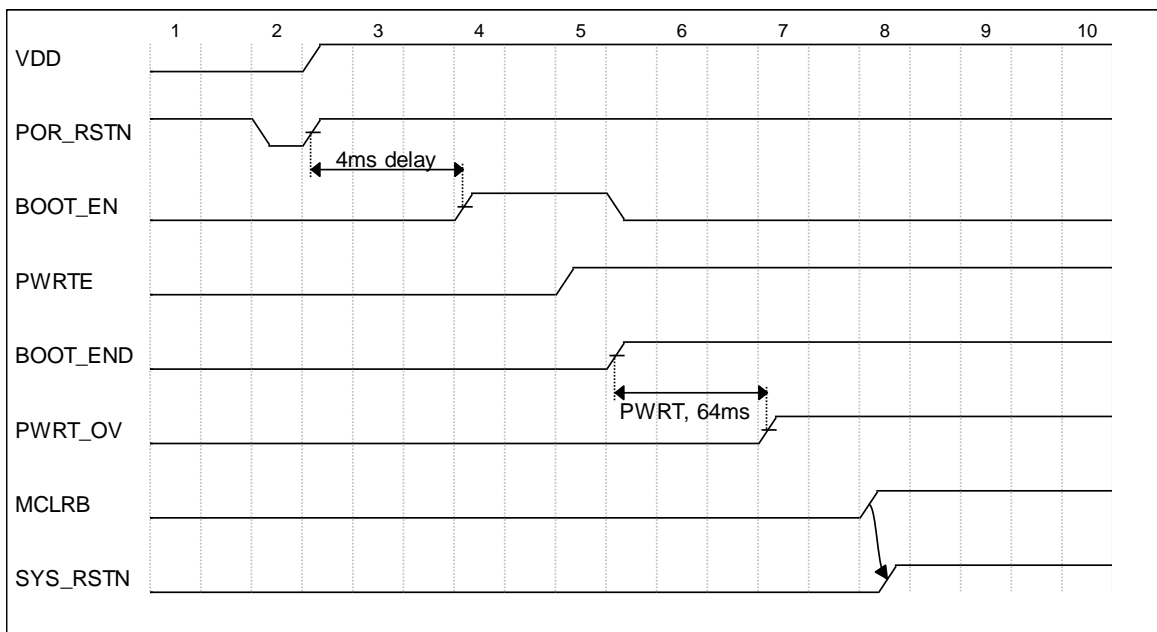


图 4.2 上电复位，使用了 MCLR

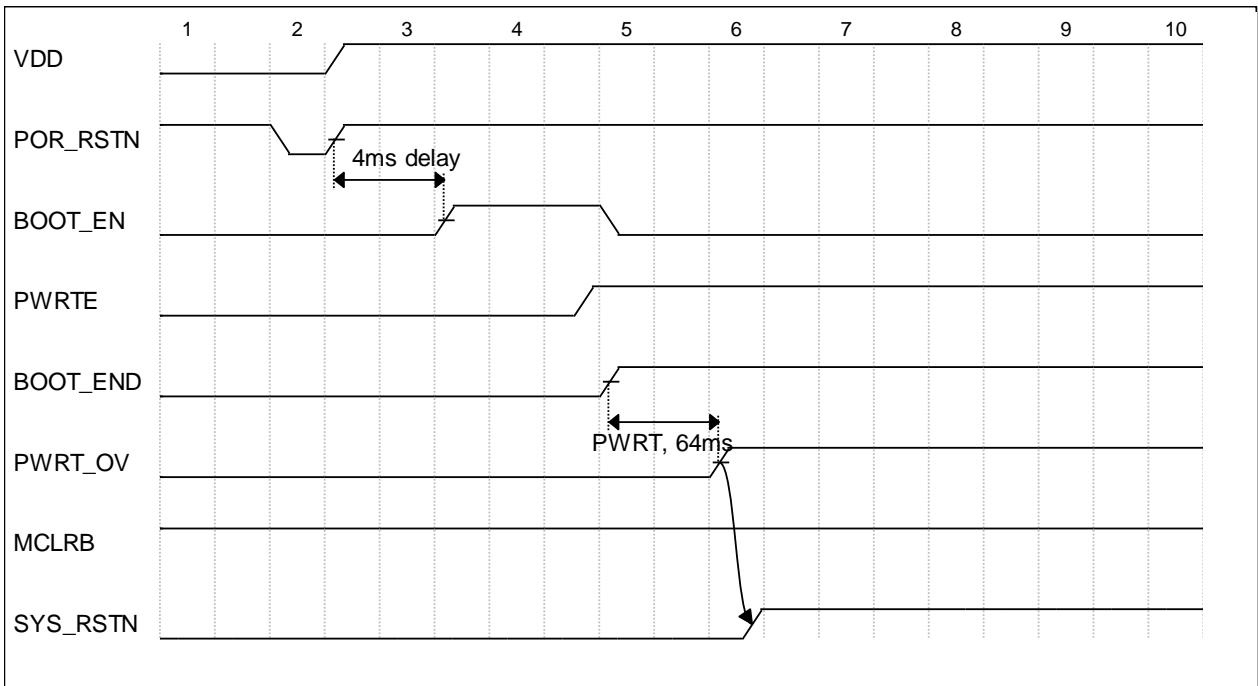


图 4.3 上电复位，没使用 MCLR

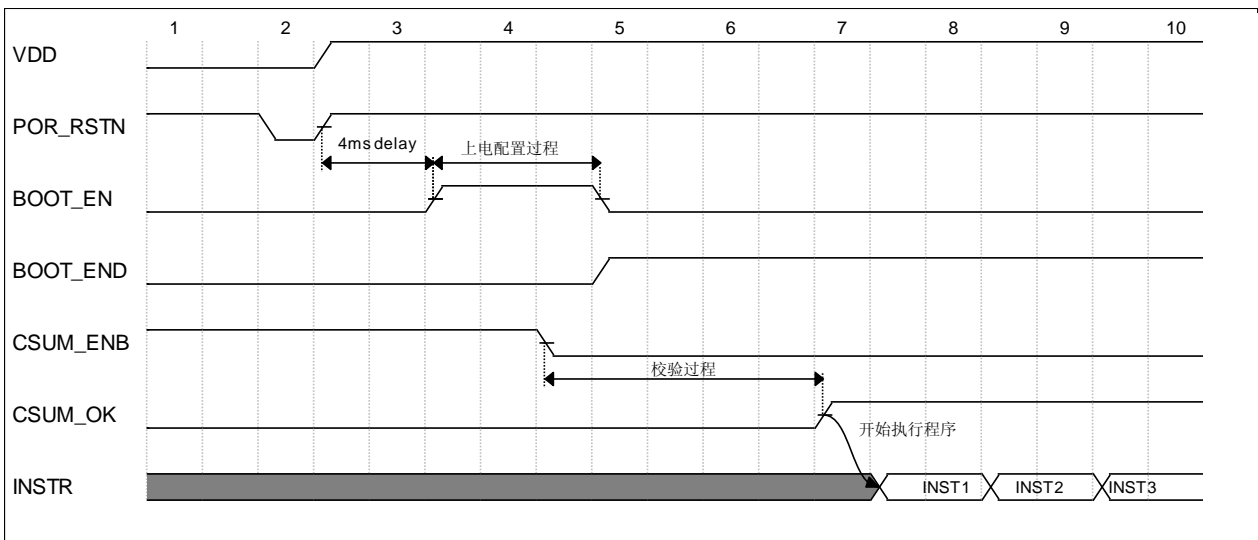


图 4.4 校验过程

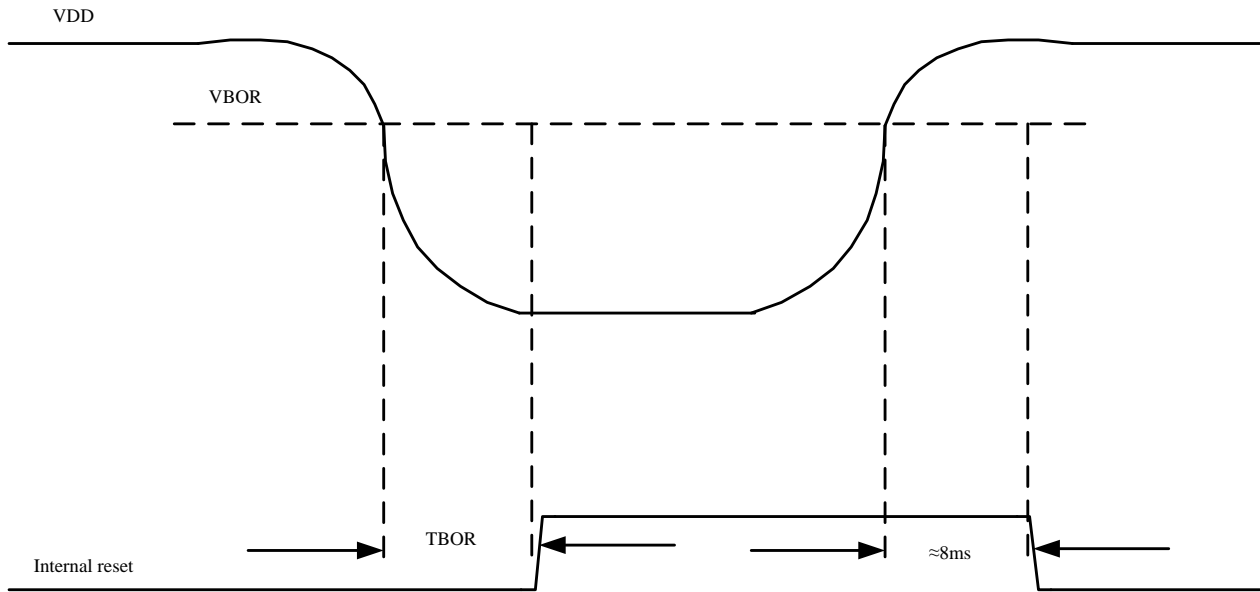


图 4.5 BOR 复位

注意:

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

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

表 4.1 各种情况下的超时

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

表 4.2 STATUS/PCON 位及其意义 (u-没变化 x-未知)

### 4.10. PCON 寄存器，地址 0x8E

Bit	7	6	5	4	3	2	1	0
Name	LVDL[3:0]				LV DEN	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	<p>低电压侦测选择位</p> <table border="1"> <thead> <tr> <th>值</th> <th>检测电</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>1.8V</td> </tr> <tr> <td>0001</td> <td>2.0 V</td> </tr> <tr> <td>0010</td> <td>2.4V</td> </tr> <tr> <td>0011</td> <td>2.7V</td> </tr> <tr> <td>0100</td> <td>3.0V</td> </tr> <tr> <td>0101</td> <td>3.3V</td> </tr> <tr> <td>0110</td> <td>3.6V</td> </tr> <tr> <td>0111</td> <td>4.0V</td> </tr> <tr> <td>1xxx</td> <td>1.2V</td> </tr> </tbody> </table>	值	检测电	0000	1.8V	0001	2.0 V	0010	2.4V	0011	2.7V	0100	3.0V	0101	3.3V	0110	3.6V	0111	4.0V	1xxx	1.2V
值	检测电																					
0000	1.8V																					
0001	2.0 V																					
0010	2.4V																					
0011	2.7V																					
0100	3.0V																					
0101	3.3V																					
0110	3.6V																					
0111	4.0V																					
1xxx	1.2V																					
3	LV DEN	<p>低电压侦测使能</p> <p>1: 开启 LVD 侦测功能</p> <p>0: 关闭 LVD 侦测功能</p>																				
2	LVDW	<p>低电压标志位，只读</p> <p>1: VDD 掉到了 LVDL[2:0]所设置的电压超过 <math>T_{BOR}</math></p> <p>0: VDD 正常，高于 LVDL[2:0]所设置的电压</p>																				
1	/POR	<p>上电复位标志，低有效</p> <p>0: 发生了上电复位</p> <p>1: 没发生上电复位或者由软件置 1</p> <p>/POR 在上电复位后值为 0，此后软件应该将其置 1</p>																				
0	/BOR	<p>低电压复位标志，低有效</p> <p>0: 发生了低电压复位</p> <p>1: 没发生低电压复位或者由软件置 1</p> <p>/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</p>																				

#### 4.11. LVDCON 寄存器，地址 0x8D

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

Bit	Name	Function
7:3	—	保留位，读 0
2	LVDDDEB	LVD 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
1:0	LVDM[1:0]	00: VDD 01: PA5 10: PA4 11: TOUCH



## 5. 看门狗定时器

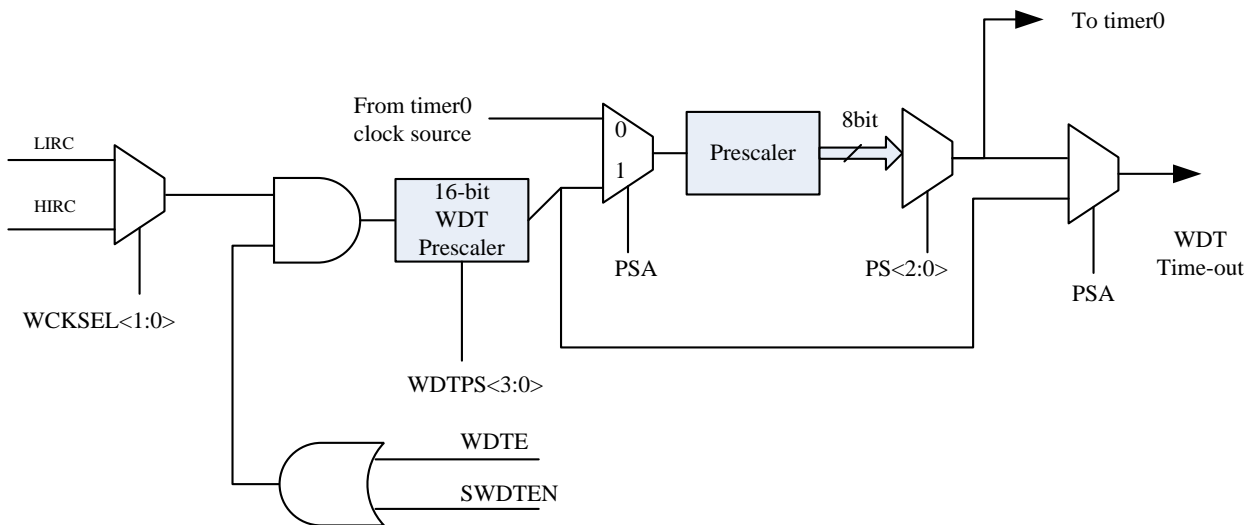


图 5.1 看门狗框图

### 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 **WDTE** 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 **CLRWDT**、**SLEEP** 会清除看门狗计数器。

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

条件	看门狗状态
WDTE 和 SWDTEN 同时为 0	清零
CLRWDT 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

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。

## 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	LVDP	—	WCKSRC	WDTPS[3:0]				SWDTEN	0-00 1000
UCFG0	0x2000	—	CPB	MCLRE	PWRTEB	WDTE	FOSC[2:0]		----	
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	LVDP	—	WCKSRC	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN
Reset	0	0	0	0	1	0	0	0
Type	RW	RO.0	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	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
6	—	保留位，读 0
5	WCKSRC	看门狗时钟源选择 0 = LIRC (可切换成 256K) 1 = HIRC
4:1	WDTPS[3:0]	<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

### 5.3.2. OPTION 寄存器，地址 0x81

见 6.3.1 小节，[OPTION 寄存器，地址 0x81](#)。

## 6. 定时器 0

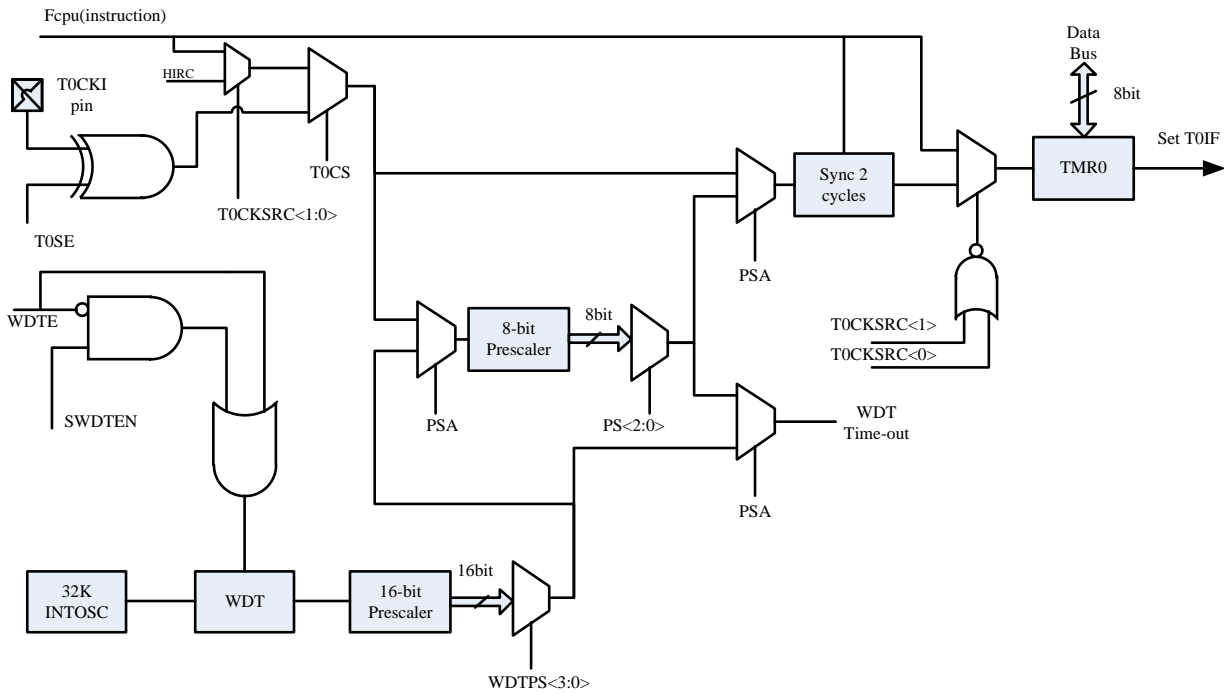


图 6.1 看门狗和定时器 0 框图

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，其计数时钟可由 T0CKSRC 控制，选择时钟源来进行计数。有一个与 WDT 共用的 8 位预分频器，PSA 为 0 时该预分频器分配给定时器 0 使用。

注意：

1. 当改变 PSA 的值时，硬件会自动把预分频器清 0。

### 6.1. Timer0 定时器模式

该模式下，定时器 0 在每个指令周期加 1(不带预分频)。软件可以清零 OPTION 寄存器里的 T0CS 位以进入定时器模式。当软件对 TMR0 进行写操作时，定时器在写后面 2 个周期内不会递增。

### 6.1.1. 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

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；

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。

### 6.1.2. TMR0 寄存器的读写

当 Timer0 时钟源跟 CPU 时钟不一样时，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。

## 6.2. 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。

### 6.2.1. 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。

### 6.2.1.1. 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
LDWI OPTION_REG
```

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

```
CLRWDT ;Clear WDT and prescaler
BANKSEL OPTION_REG
LDWI b'11111000' ;Mask TMR0 select and prescaler bits
ANDWR OPTION_REG,W
IORWI b'00000011' ;Set prescale to 1:16
STR OPTION_REG
```

### 6.2.2. 定时器 0 中断

芯片在定时器 0 从 0xFF 溢出到 0x00 时会置起 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 T0CKRUN 为 1 且其时钟源不是指令时钟时。

### 6.2.3. 用外部时钟驱动定时器 0

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

### 6.2.4. 睡眠模式下的状态

当 T0CKRUN=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 T0CKSRC 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。

### 6.3. 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		---- 1000
TRISA	0x85	TRISA[7:0], PORTA 方向控制								1111 1111

#### 6.3.1. 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>PORTA上拉使能</b> 1 = 禁止上拉 0 = 由 WPUA 决定是否上拉																											
6	INTEDG	<b>中断边沿选择</b> 1 = PA2/INT上升沿产生中断 0 = PA2/INT下降沿产生中断																											
5	T0CS	<b>Timer0时钟源选择</b> 1 = PA2/T0CKI引脚跳变 0 = 时钟由 T0CKSRC 寄存器位决定																											
4	T0SE	<b>Timer0时钟边沿选择</b> 1 = PA2/T0CKI由高到低变化时递增 0 = PA2/T0CKI由低到高变化时递增																											
3	PSA	<b>预分频分配位</b> 1 = 将预分频分配给WDT 0 = 将预分频分配给 Timer0																											
2:0	PS<2:0>	<b>预分频比设置</b> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">值</th> <th style="text-align: left;">Timer0分频比</th> <th style="text-align: left;">WDT分频比</th> </tr> </thead> <tbody> <tr><td>000</td><td>1 : 2</td><td>1 : 1</td></tr> <tr><td>001</td><td>1 : 4</td><td>1 : 2</td></tr> <tr><td>010</td><td>1 : 8</td><td>1 : 4</td></tr> <tr><td>011</td><td>1 : 16</td><td>1 : 8</td></tr> <tr><td>100</td><td>1 : 32</td><td>1 : 16</td></tr> <tr><td>101</td><td>1 : 64</td><td>1 : 32</td></tr> <tr><td>110</td><td>1 : 128</td><td>1 : 64</td></tr> <tr><td>111</td><td>1 : 256</td><td>1 : 128</td></tr> </tbody> </table>	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
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																											

### 6.3.2. TMR0, 地址 0x01

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

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

### 6.3.3. T0CON0, 地址 0x1F

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	T0ON	T0CKRUN	T0CKSRC	
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



## 7. 定时器 2

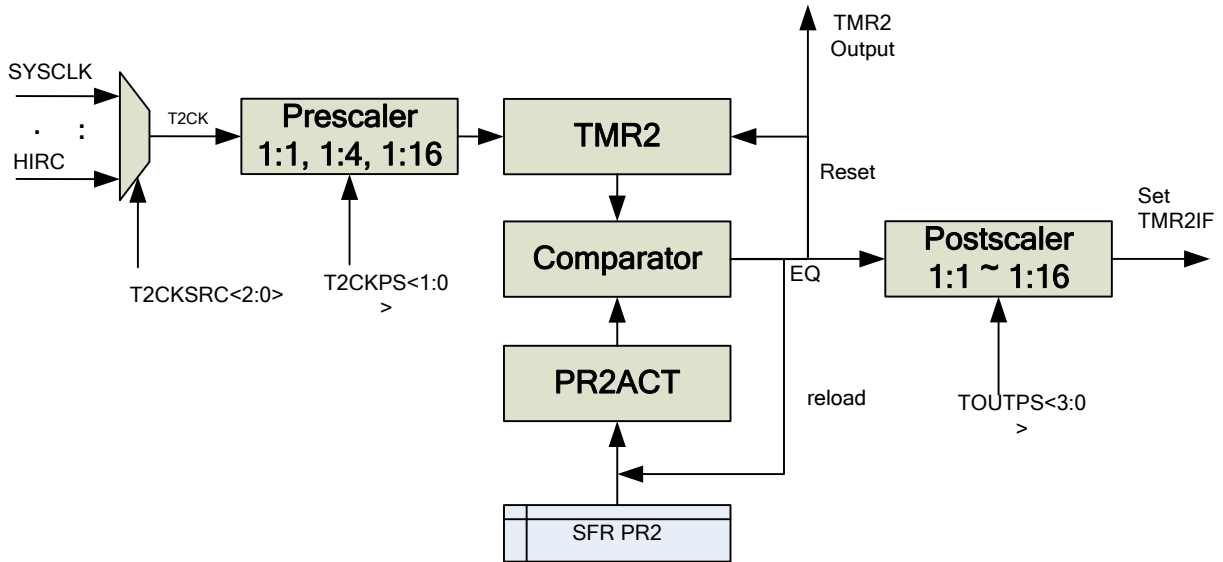


图 7.1 定时器 2 框图

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
- 1:1–1:16 后分频比
- 时钟源可选：系统时钟或者内部 32MHz 时钟或 LIRC

## 7.1. 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（4T:  $F_{SYS}/4$ ）。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。

## 7.2. 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。**

### 7.3. Timer2 计数值的读写

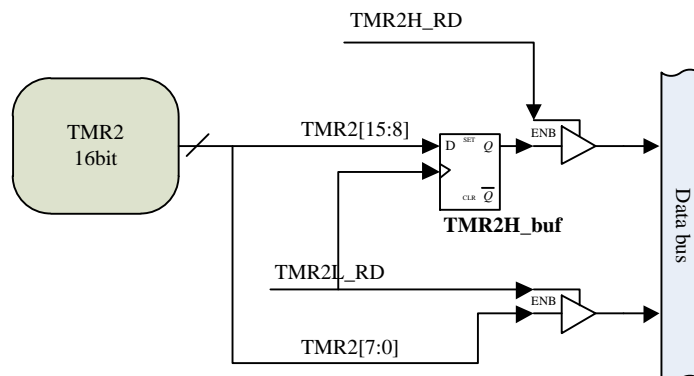


图 7.2 Timer2 计数值的读操作原理框图

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;

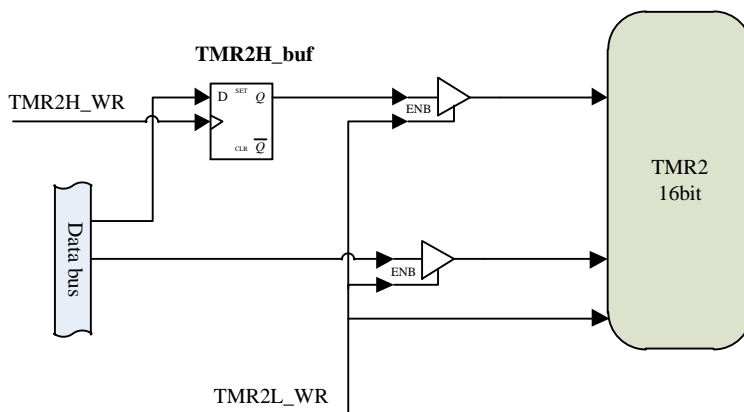


图 7.3 Timer2 计数值的写操作原理框图

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。

## 7.4. 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
- 外部时钟 2 倍频（只有当 FOSC 为 EC 模式时才有效）
- HIRC
- LIRC

## 7.5. 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	—	—	OSFIE	TMR2IE	—	000- -00-
PIR1	0x0C	EEIF	CKMIF	LVDIF	—	—	—	TMR2IF	—	000- -00-
MSCON	0x1B	BGRBOE	LVROE	—	—	SLVREN	CKMAVG	CKCNTI	T2CKRUN	0011 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			---0 0000

### 7.5.1. PR2 寄存器，地址 0x91, 0x92

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

### 7.5.2. TMR2 寄存器，地址 0x11, 0x13

TMR2L, 地址 0x11

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

TMR2H, 地址 0x13

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

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

### 7.5.3. 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	

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[3:0]	<b>TOUTPS&lt;3:0&gt;</b> : Timer2 Output Postscaler Select bits 定时器2输出后分频比选择 0000 = 1:1 后分频比 0001 = 1:2 后分频比 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[1:0]	<b>T2CKPS&lt;1:0&gt;</b> : Timer2 Clock Prescale Select bits 定时器2驱动时钟预分频比选择 00 = Prescaler is 1 01 = Prescaler is 4 1x = Prescaler is 16

### 7.5.4. T2CON1 寄存器，地址 0x9E

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	P1OS	P1BZM	T2CKSRC		
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	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 = 外部时钟 2 倍频（只有当 FOSC 配置为 EC 模式时才有效） 100 = HIRC 101 = LIRC

## 8. 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。

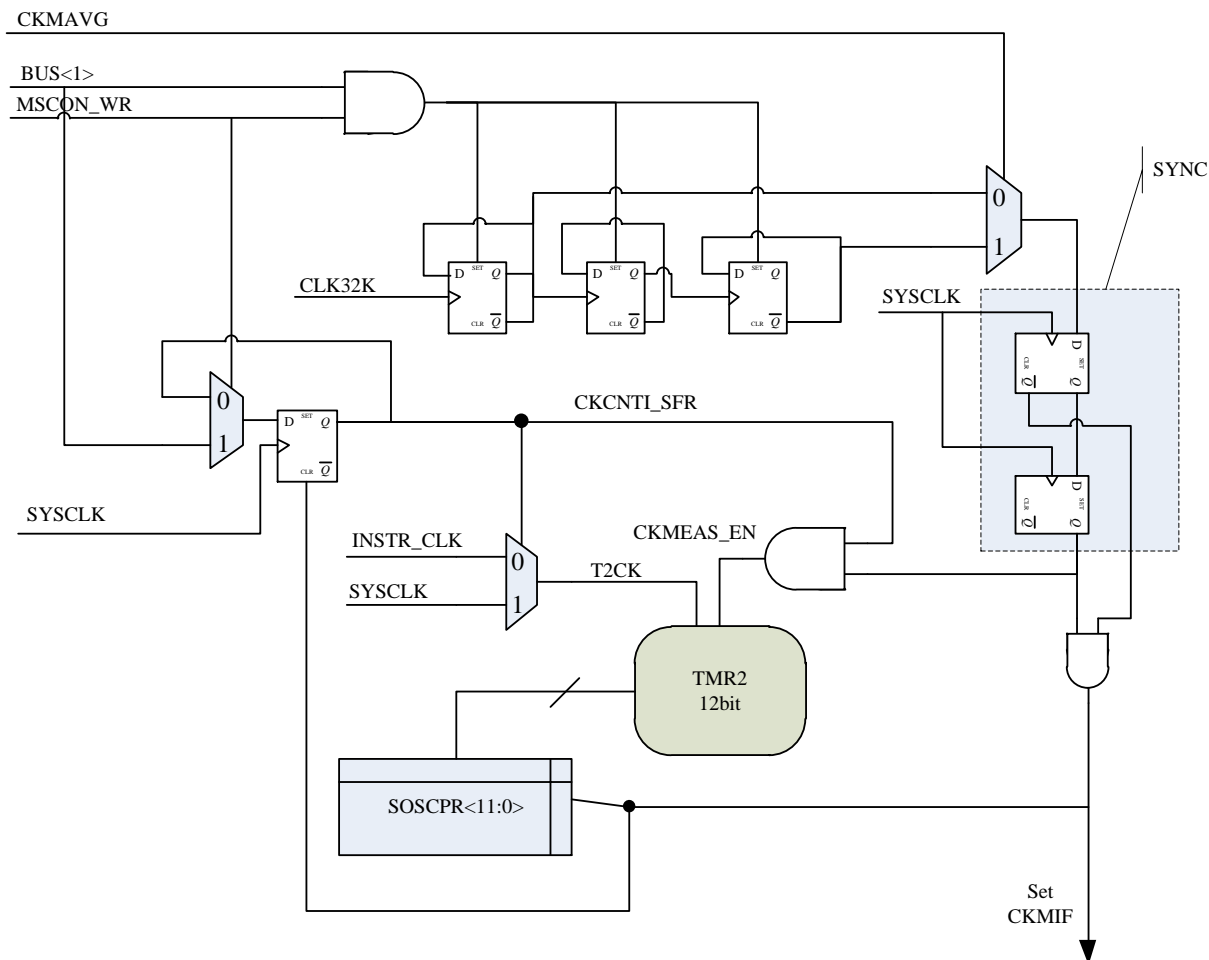


图 8.1 慢时钟测量结构框图

### 8.1. 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。



## 8.2. 操作步骤

1. 为提高计量精度，建议设置 IRCF 为 111，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 MSCON.2 置 1，否则把它清 0；
4. 置位 MSCON.1，开始测量；
5. 测量结束后 MSCON.1 自动清 0，中断标志置 1；
6. 可以用查询或中断的方式等待结束；
7. 当查询到中断标志为 1 时读取得到的 SOSCPR 即为最终结果。

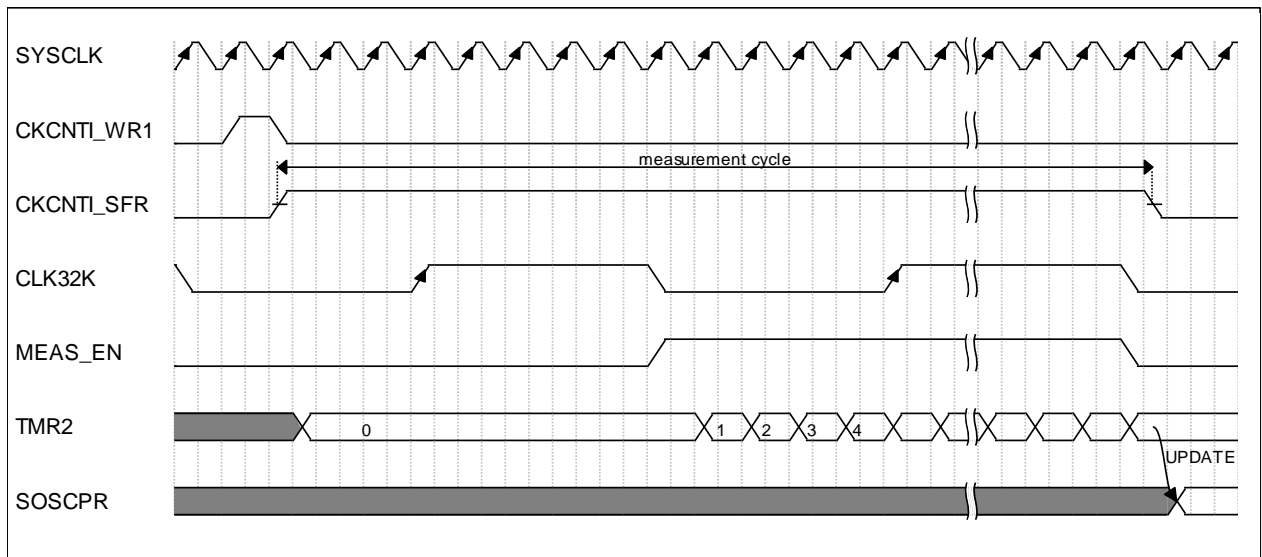


图 8.2 单次测量时序示意图（CLK32K 和 SYSCLK 未按真实比例画）

注意：

1. 在慢时钟测量过程中软件不要写 SOSCPRH/L；
2. 不要在单步调试下做慢时钟测量，因为暂停模式下 Timer2 被停止，这样会导致测量结果不正确；

## 8.3. 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
MSCON	0x1B	BGRBOE	LVROE	—	—	SLVREN	CKMAVG	CKCNTI	T2CKRUN	0011 0000
SOSCPRL	0x1C	SOSCPR[7:0]								1111 1111
SOSCPRH	0x1D	—				SOSCPR[11:8]				---- 1111
INTCON	0x0B	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF	0000 0000
PIE1	0x8C	EEIE	CKMIE	LVDIE	—	—	OSFIE	TMR2IE	—	000- -00-
PIR1	0x0C	EEIF	CKMIF	LVDIF	—	—	OSFIF	TMR2IF	—	000- -00-

### 8.3.1. MSCON 寄存器，地址 0x1B

Bit	7	6	5	4	3	2	1	0
Name	BGRBOE	LVROE	—		SLVREN	CKMAVG	CKCNTI	T2CKRUN
Reset	0	0	—		0	0	0	0
Type	RW	RW	RO.0	RO.0	RW	RW	RW	RW

Bit	Name	Function
7	N/A	保留位，不要写 1
6	N/A	保留位，不要写 1
5	—	—
4	—	—
3	SLVREN	<p>软件控制 LVR 使能位，当 UCFG1&lt;1:0&gt;为 01 时：            1 = 打开 LVR            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）            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            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            0 = T2CK 睡眠时停止工作</p>

### 8.3.2. 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

## 9. PWM1 模块

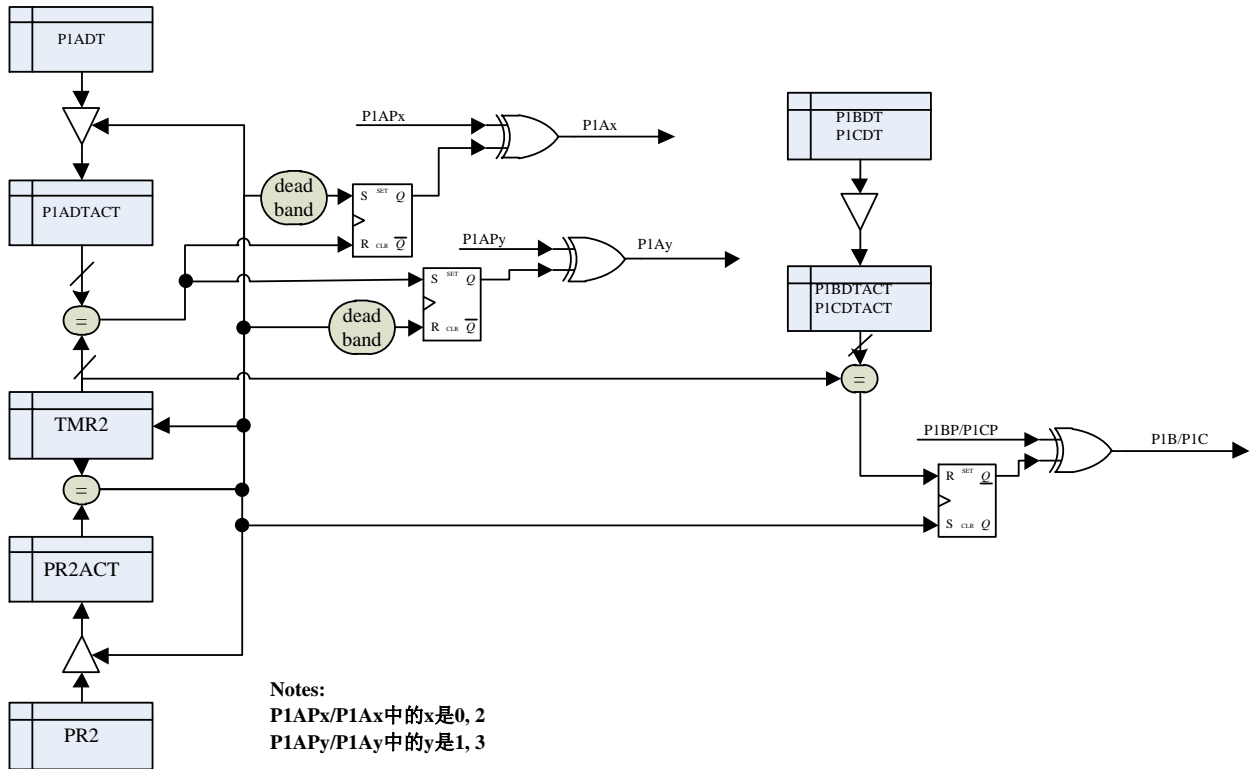


图 9.1 PWM1 结构框图

PWM1 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

### 9.1. 周期

PWM 周期由 Timer2 的 PR2 寄存器指定。用公式 9.1 可计算 PWM 周期。

$$\text{PWM 周期} = (\text{PR2} + 1) * \text{TT2CK} * (\text{TMR2 预分频值})$$

公式 9.1

当 TMR2 等于 PR2 时，下一次递增周期将发生以下三个事件:

- TMR2 被清零
- 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

## 9.2. 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 9.2 用于计算 PWM 脉宽。

公式 9.3 用于计算 PWM 占空比。

脉冲宽度 =  $P1xDT * TT2CK * (TMR2 \text{ 预分频值})$

公式 9.2

占空比 =  $P1xDT \div (PR2 + 1)$

公式 9.3

## 9.3. 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

## 9.4. 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。

## 9.5. P1A 的死区时间

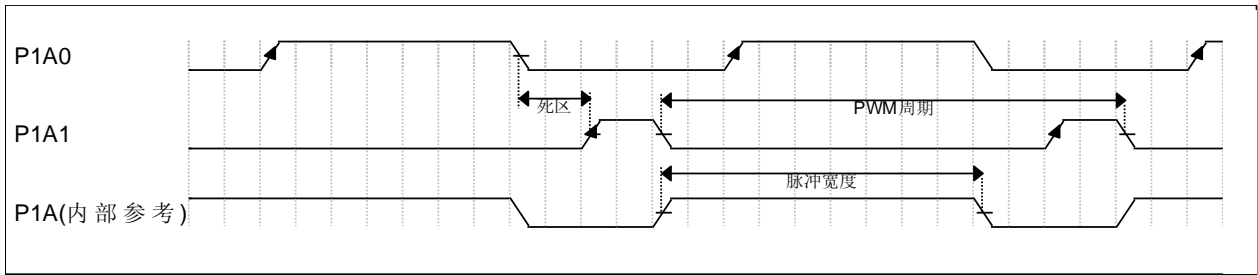


图 9.5.1 死区 PWM 示意图

有 1 路 PWM 带互补输出，P1Ax。其中 P1A0 定义为正输出，[P1A0]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]} 这对互补输出共享同一死区设置。

## 9.6. 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：

- BK0 管脚为低电平
- BK0 管脚为高电平
- LVDW 被 LVD 模块置 1
- LVDW=1 或 BK0=0
- LVDW=1 或 BK0=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。

### 9.6.1. 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 定时器处于复位状态
- TMR2ON 位不受影响

有效电平由 P1POL 各寄存器位决定。

### 9.6.2. 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。

### 9.6.3. 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 位将被硬件清零，TMR2 恢复计数，在下次计数溢出时，PWM1 恢复正常输出。

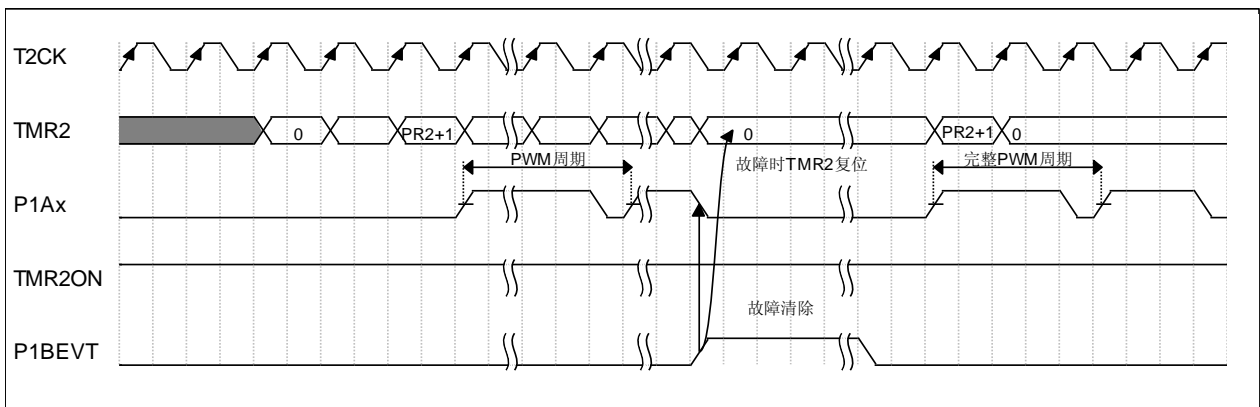


图 9.6.1 PWM 的自动重启时序图

## 9.7. 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。

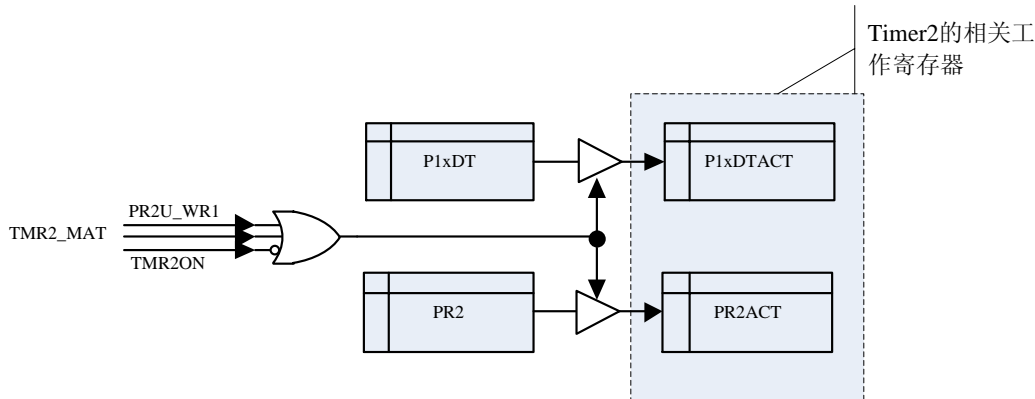


图 9.7.1 T2 工作寄存器的更新

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 9.7。

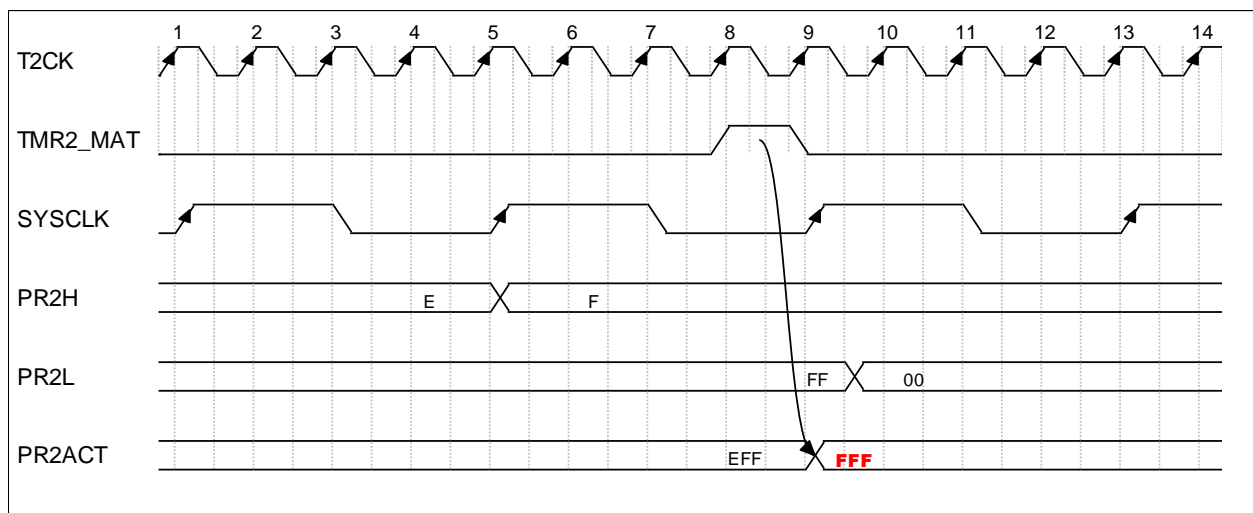


图 9.7 PR2ACT 被更新为意外值 FFF（期望值是 F00）

所以强烈建议更新 PR2 和 xxxDT 只在 TMR2 匹配中断里面做。



## 9.8. 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。

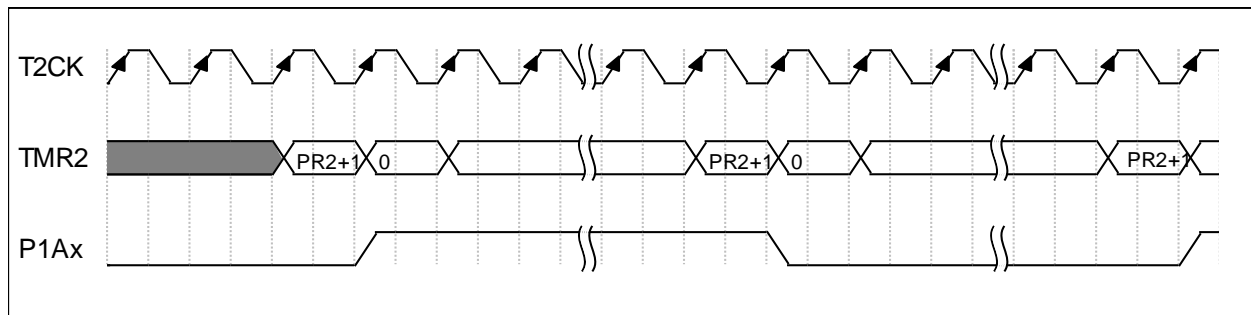


图 9.8 蜂鸣器模式输出

## 9.9. 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。

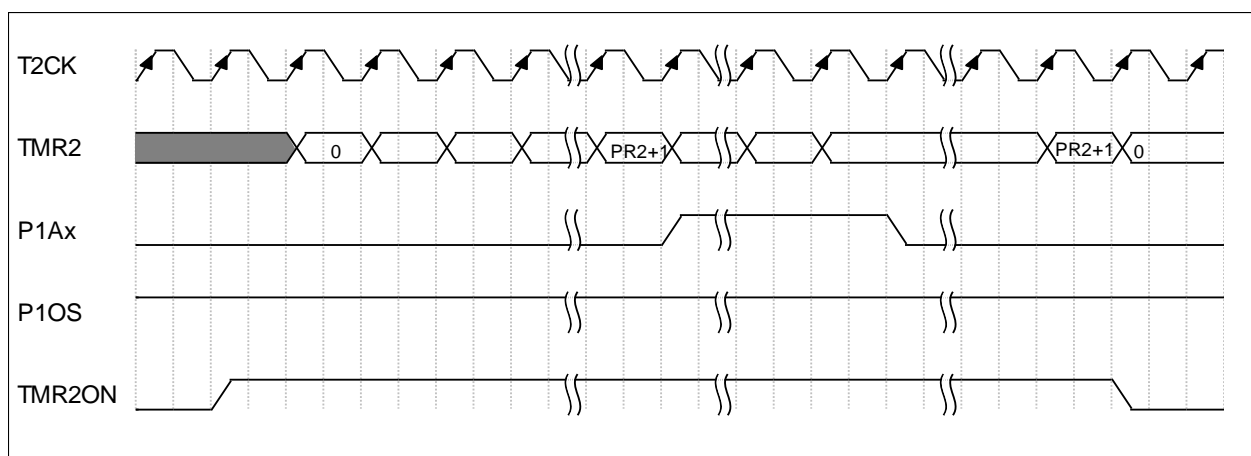


图 9.9 单脉冲模式输出

注意:

1. [P1C]、[P1D]不支持单脉冲输出功能

## 9.10. P1D、P1C 输出重映射

P1D 和 P1C 路 PWM 可以分别映射到 PA5 和 PA0, 由寄存器位 P1DALT 和 P1CALT 控制, 它们的复位值为 0, P1D 默认从 PA1 输出, P1C 默认从 PA2 输出。

### 9.11. P1D 的第 2 功能输出

除了正常的 PWM 波形输出，P1D 的重映射管脚（PA5）可输出 P1B、P1C 之间的同或，异或输出，由寄存器 P1DF2E 控制。

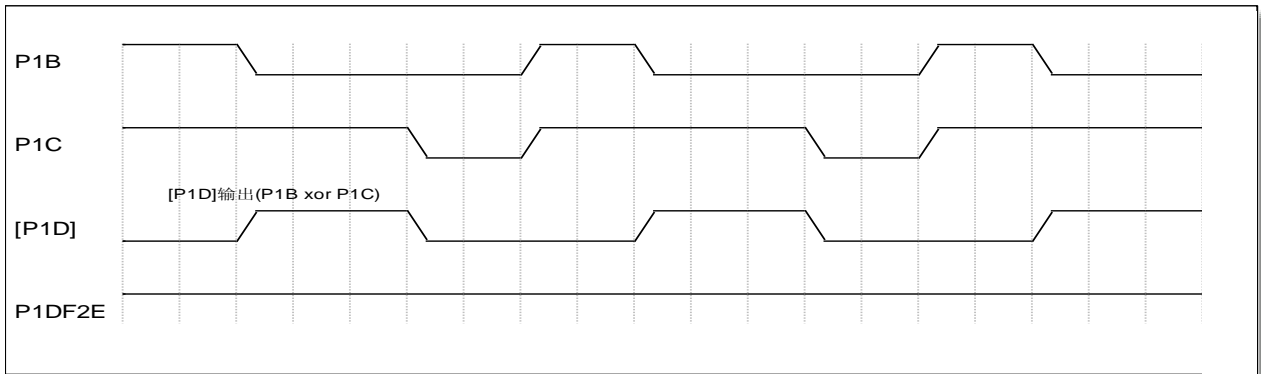


图 9.10 P1D 的第 2 功能时序示例

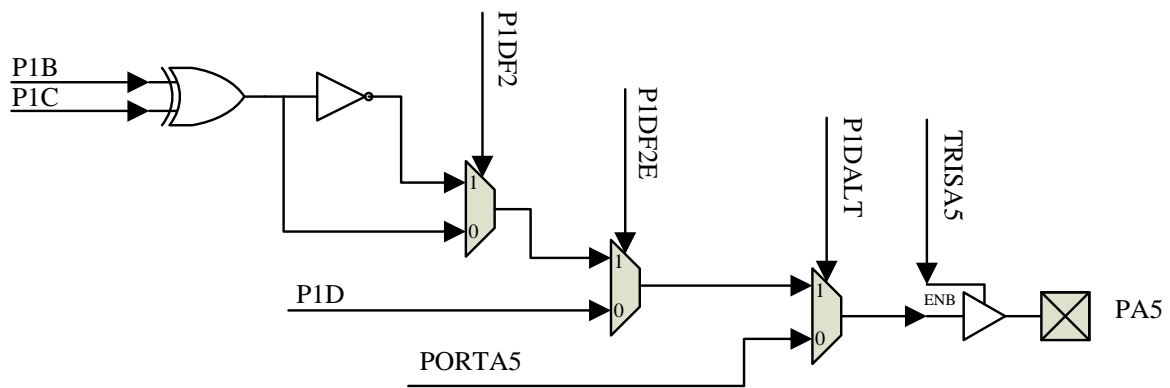


图 9.11 P1D 的第 2 功能输出

## 9.12. 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	P1DSS		P1D2SS		P1CALT	P1DALT	P1CSS		0000 0000	
P1OE	0x90	P1COE	P1BOE	P1DOE	—	P1C2SS[1:0]		P1A0NOE	P1A0OE	000- 0000	
PR2L	0x91	PR2[7:0]								1111 1111	
PR2H	0x92	PR2[11:8]								1111 1111	
P1POL	0x99	P1CP	P1BP	P1DP	—	—	—	P1A0NP	P1A0P	000- --00	
P1AUX	0x1E	—	—	—	—	—	—	P1DF2E	P1DF2	---- --00	

### 9.12.1. P1ADTL 寄存器，地址 0x0E

Bit	7	6	5	4	3	2	1	0
Name	P1ADTL							
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

### 9.12.2. P1BDTL 寄存器，地址 0x0F

Bit	7	6	5	4	3	2	1	0
Name	P1BDTL							
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

### 9.12.3. P1CDTL 寄存器，地址 0x10

Bit	7	6	5	4	3	2	1	0
Name	P1CDTL							
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

### 9.12.4. TMR2L 寄存器，地址 0x11

Bit	7	6	5	4	3	2	1	0
Name	TMR2L							
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

### 9.12.5. TMR2H 寄存器，地址 0x13

Bit	7	6	5	4	3	2	1	0
Name	TMR2H							
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

### 9.12.6. T2CON0 寄存器，地址 0x12

见 [T2CON0 寄存器，地址 0x12](#)。

### 9.12.7. P1ADTH 寄存器，地址 0x14

Bit	7	6	5	4	3	2	1	0
Name	P1ADTH							
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

### 9.12.8. P1BDTH 寄存器，地址 0x15

Bit	7	6	5	4	3	2	1	0
Name	P1BDTH							
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

### 9.12.9. P1CDTH 寄存器，地址 0x1A

Bit	7	6	5	4	3	2	1	0
Name	P1CDTH							
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

### 9.12.10. P1DDTL 寄存器，地址 0x8

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

Bit	Name	Function
3:0	P1DDTH	P1D 占空比寄存器低 8 位

### 9.12.11. P1DDTH 寄存器，地址 0x9

Bit	7	6	5	4	3	2	1	0
Name	P1DDTH							
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

### 9.12.12. P1CON 寄存器，地址 0x16

Bit	7	6	5	4	3	2	1	0
Name	P1AUE	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	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[6:0]	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

**9.12.13. P1BR0 寄存器，地址 0x17**

Bit	7	6	5	4	3	2	1	0
Name	P1BEVT	P1BKS			P1BSS		P1ASS	
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[2:0]	<p>PWM1 故障源选择位</p> <p>000 = 禁止故障刹车功能</p> <p>001 = BK0 为低电平</p> <p>010 = BK0 为高电平</p> <p>011 = LVDW=1</p> <p>100 = BK0 为低电平或 LVDW=1</p> <p>101 = BK0 为高电平或 LVDW=1</p> <p>110 = 保留（禁止故障刹车）</p> <p>111 = 保留（禁止故障刹车）</p>
3:2	P1BSS	<p>故障下，P1B 管脚的状态</p> <p>00 = 高阻</p> <p>01 = 无效电平</p> <p>1x = 有效电平（高电平或低电平，由 P1POL 寄存器指定）</p>
1:0	P1ASS	<p>故障下，P1A 管脚的状态</p> <p>00 = 高阻</p> <p>01 = 无效电平</p> <p>1x = 有效电平（高电平或低电平，由 P1POL 寄存器指定）</p>

**9.12.14. P1BR1 寄存器，地址 0x19**

Bit	7	6	5	4	3	2	1	0
Name	P1DSS		P1D2SS		P1CALT	P1DALT	P1CSS	
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	P1DSS	故障下，P1D 管脚的状态 00 = 高阻 01 = 无效电平 1x = 有效电平（高电平或低电平，由 P1POL 寄存器指定）
5:4	P1D2SS	故障下，[P1D]管脚的状态，只有当 P1DALT 为 1 时才有效 00 = 高阻 01 = 输出 0 1x = 输出 1
3	P1CALT	[P1C]功能映射选择 1 = 输出 P1C 0 = GPIO
2	P1DALT	[P1D]功能映射选择 1 = 输出 P1D，或者 P1B 和 P1C 的逻辑波形 0 = GPIO
1:0	P1CSS	故障下，P1C 管脚的状态 00 = 高阻 01 = 无效电平 1x = 有效电平（高电平或低电平，由 P1POL 寄存器指定）



**9.12.15. P1OE 寄存器，地址 0x90**

Bit	7	6	5	4	3	2	1	0
Name	P1COE	P1BOE	P1DOE	—	P1C2SS		P1A0NOE	P1A0OE
Reset	0	0	0	—	0	0	0	0
Type	RW	RW	RW	RO.0	RW	RW	RW	RW

Bit	Name	Function
7	P1COE	P1C 输出使能，高有效 1 = 允许 P1C 输出到管脚 0 = 禁止 P1C 输出到管脚
6	P1BOE	P1B 输出使能，高有效 1 = 允许 P1B 输出到管脚 0 = 禁止 P1B 输出到管脚
5	P1DOE	P1D 输出使能，高有效 1 = 允许 P1D 输出到管脚 0 = 禁止 P1D 输出到管脚
4	—	—
3:2	P1C2SS	故障下， <b>[P1C]</b> 管脚的状态，只有当 P1CALT 为 1 时才有效 00 = 高阻 01 = 输出 0 1x = 输出 1
1	P1A0NOE	P1A0NOE 输出使能，高有效 1 = 允许 P1A0N 输出到管脚 0 = 禁止 P1A0N 输出到管脚
0	P1A0OE	P1A0OE 输出使能，高有效 1 = 允许 P1A0 输出到管脚 0 = 禁止 P1A0 输出到管脚

**9.12.16. PR2L 寄存器，地址 0x91**

Bit	7	6	5	4	3	2	1	0
Name	PR2L							
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

**9.12.17. PR2H 寄存器，地址 0x92**

Bit	7	6	5	4	3	2	1	0
Name	PR2H							
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

**9.12.18. P1POL 寄存器，地址 0x99**

Bit	7	6	5	4	3	2	1	0
Name	P1CP	P1BP	P1DP	—	—	—	P1A0NP	P1A0P
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RO.0	RO.0	RO.0	RW	RW

Bit	Name	Function
7	P1CP	P1C 输出极性设置 1 = P1C 低电平有效 0 = P1C 高电平有效
6	P1BP	P1B 输出极性设置 1 = P1B 低电平有效 0 = P1B 高电平有效
5	P1DP	P1D 输出极性设置 1 = P1D 低电平有效 0 = P1D 高电平有效
4:2	—	—
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

**9.12.19. P1AUX 寄存器，地址 0x1E**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	P1DF2E	P1DF2
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	P1DF2E	当 P1DALT 为 1 时，[P1D]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	[P1D]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

## 10. 中断处理

FT62F21x 有以下中断源：

- PA2/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
- EEPROM 数据写中断
- 时钟缺失中断
- LVD 中断

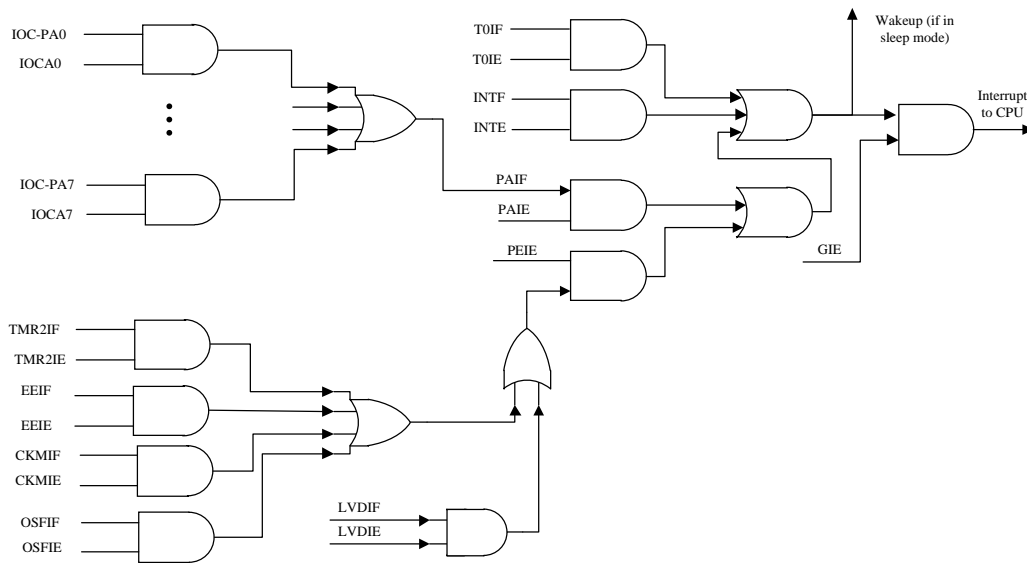


图 10.1 中断产生电路架构框图

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：

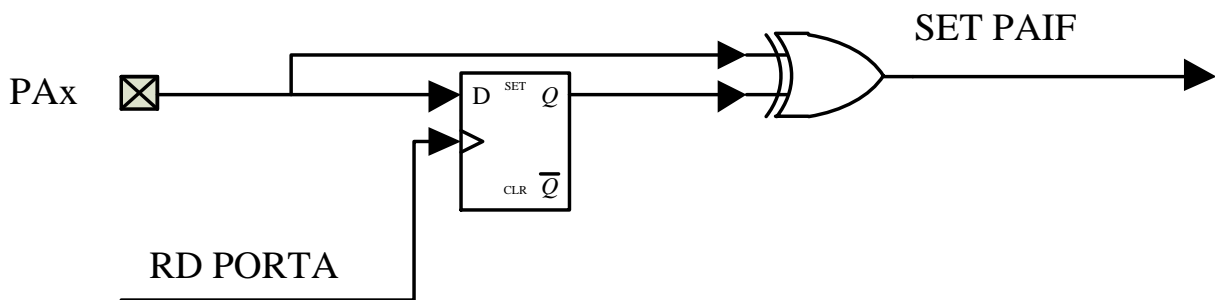
- INT 管脚中断
- PORTA 变化中断
- Timer0 溢出中断

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。

## 10.1. 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。

## 10.2. 端口变化中断



10.2 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：

2. 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；
3. 当 I/O 电平发生变化时，PAIF 被置 1；
4. 清中断标志位之前应该读取一下 PORTA，然后再对 PAIF 清 0；

### 10.2.1. 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

### 10.3. 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。

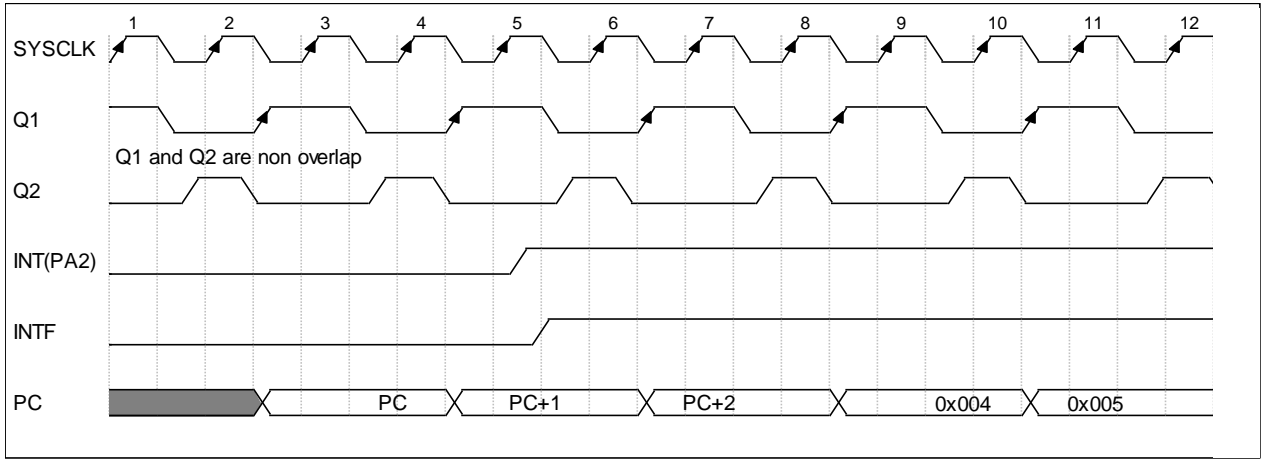


图 10.3 中断响应时序图

### 10.4. 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。

### 10.5. 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。

### 10.6. 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	—	—	—	TMR2IE	—	000- --0-
PIR1	0x0C	EEIF	CKMIF	LVDIF	—	—	—	TMR2IF	—	000- --0-
TRISA	0x85	—	—	TRISA[5:0], PORTA 方向控制						--11 1111
IOCA	0x96	—	—	IOCA[5:0], 端口变化中断允许位						--00 0000

### 10.6.1. INTCON 寄存器，地址 0x0B/0x8B

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

Bit	Name	Function
7	GIE	<b>GIE:</b> 全局中断使能 1 = 打开全局中断 0 = 关闭全局中断
6	PEIE	<b>PEIE:</b> 外设中断使能 1 = 允许外设中断 0 = 禁止外设中断
5	TOIE	<b>TOIE:</b> Timer0溢出中断使能 1 = 允许Timer0中断 0 = 禁止Timer0中断
4	INTE	<b>INTE:</b> PA2/INT外部中断使能 1 = 允许PA2外部中断 0 = 禁止PA2外部中断
3	PAIE	<b>PAIE:</b> PAX端口变化中断使能 1 = 允许PAX端口变化中断 0 = 禁止PAX端口变化中断
2	TOIF	<b>TOIF:</b> Timer0溢出中断标志位 1 = Timer0产生了溢出（必须由软件清0） 0 = Timer0未产生溢出
1	INTF	<b>INTF:</b> PA2外部中断标志位 1 = 产生了PA2外设中断（必须由软件清0） 0 = 未产生PA2外设中断
0	PAIF	<b>PAIF:</b> PAX端口变化中断标志位 1 = 由 IOCAx 使能的 PAX 口产生了变化中断 0 = 由 IOCAx 使能的 PAX 未产生变化中断

### 10.6.2. PIR1 寄存器，地址 0x0C

Bit	7	6	5	4	3	2	1	0
Name	EEIF	CKMIF	LVDIF	—	—	—	TMR2IF	—
Reset	0	0	0	—	—	—	0	—
Type	RW	RW	RW	RO-0	RO-0	RO-0	RW	RO-0

Bit	Name	Function
7	EEIF	<b>EEIF:</b> 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	<b>CKMIF:</b> 快时钟测量慢时钟操作完成中断标志位 1 = 快时钟测量慢时钟操作完成 0 = 快时钟测量慢时钟未完成，或已经由软件清 0
5	LVDIF	<b>LVDIF:</b> 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	N/A	保留位，读 0
3	N/A	保留位，读 0
2	N/A	保留位，读 0
1	TMR2IF	<b>TMR2IF:</b> 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	N/A	保留位，读 0



### 10.6.3. PIE1 寄存器，地址 0x8C

Bit	7	6	5	4	3	2	1	0
Name	EEIE	CKMIE	LVDIE	—	—	—	TMR2IE	—
Reset	0	0	0	—	—	—	0	—
TYPE	RW	RW	RW	RO-0	RO-0	RO-0	RW	RO-0

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	N/A	保留位，读 0
3	N/A	保留位，读 0
2	N/A	保留位，读 0
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	N/A	保留位，读 0

### 10.6.4. IOCA 寄存器，地址 0x96

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

Bit	Name	Function
5:0	IOCA[5:0]	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

## 11. 睡眠模式

芯片在执行完 **SLEEP** 指令后进入睡眠状态。

为了达到最低睡眠功耗，软件应该将所有 I/O 置高或低，而且没有外部电路从 I/O 耗电。I/O 作为输入的，外部电路应将其拉高或拉低，或者打开内部上拉，避免翻转耗电。

### 11.1. 唤醒模式

以下事件可以唤醒芯片：

- /MCLR 管脚上有外部复位
- WDT 超时溢出
- PA2/INT 管脚上有中断，PORTA 变化中断
- TMR2 中断（当 T2CKRUN 为 1 且其时钟源不选择指令时钟时）
- LVD 使能并发生低电压事件

清看门狗指令 **CLRWDWT**、**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。

### 11.2. 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。

## 12. I/O 端口

本芯片共包含 6 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。

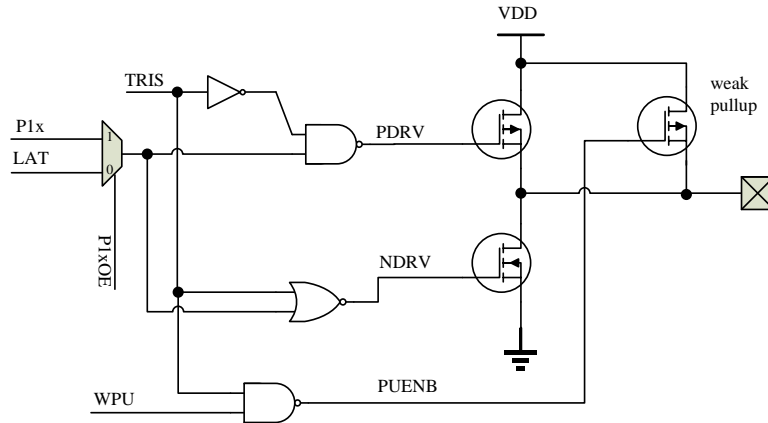


图 12.1 I/O 的一般结构

### 12.1. 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，PORTA[5] 读的值为 0，此时它是作为外部复位管脚。

### 12.2. 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。

#### 12.2.1. 弱上拉

PORTA 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。PORTA[5] 内部也有弱上拉功能，它是在将 PORTA[5] 设置为 /MCLR 功能时自动使能的。当 PORTA[5] 被设置为 GPIO 时，该弱上拉电路由 WPUA5 控制。

## 12.3. 与 GPIO 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
WPUA	0x95	—		PORTA 上拉控制位						1111 1111
TRISA	0x85	—		TRISA[5:0], PORTA 方向控制						--11 1111
PORTA	0x05	—		PORTA[5:0], PORTA 数据寄存器						xxxx xxxx
IOCA	0x96	—		IOCA[5:0], PORTA 端口变化中断允许位						--00 0000
OPTION	0x81	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
PSRCA	0x88	—				PSRCA[3:0]				---- 1111
PSINKA	0x97	—						PSINKA[1:0]		---- --11

### 12.3.1. WPUA, 地址 0x95

Bit	7	6	5	4	3	2	1	0
Name	—	—	WPUA					
Reset	—	—	1	1	1	1	1	1
Type	RO.0	RO.0	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	—	保留位, 读 0
5:0	WPUA	PORTA 弱上拉控制寄存器 1 = 使能弱上拉 0 = 关闭弱上拉

### 12.3.2. TRISA, 地址 0x85

Bit	7	6	5	4	3	2	1	0
Name	—	—	TRISA					
Reset	—	—	1	1	1	1	1	1
Type	RO.0	RO.0	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	—	保留位, 读 0
5:0	TRISA	PORTA 方向控制寄存器 1 = 输入 0 = 输出

### 12.3.3. PORTA, 地址 0x05

Bit	7	6	5	4	3	2	1	0
Name	—	—	PORTA					
Reset	—	—	x	x	x	x	x	x
Type	RO.0	RO.0	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	—	保留位, 读 0
5:0	PORTA	PORTA 数据寄存器

### 12.3.4. PSRCA, 地址 0x88

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

Bit	Name	Function
7:4	—	保留位, 读 0
3:0	PSRCA	控制 PORTA[5:4]源电流能力 PSRCA[3:2] 控制 PORTA[5], PSRCA[1:0]控制 PORTA[4]

PSRCA 值	源电流能力
00	3mA
01/10	6mA
11	24mA

注意: 其他不可设置的 IO (PA0~PA3), 源电流能力为 18mA。

### 12.3.5. PSINKA, 地址 0x97

Bit	7	6	5	4	3	2	1	0
Name	—						PSINKA	
Reset	—						1	1
Type	RO.0	RO.0	RO.0	RO.0	RO.0	RO.0	RW	RW

Bit	Name	Function
7:2	—	保留位, 读 0
1:0	PSINKA	PORTA[5:4]的灌电流能力设置 PSINKA[1]控制 PORTA[5], PSINKA[0]控制 PORTA[4] 0: L0 (最小) 1: L1 (最大)

注意: 其他不可设置的 IO (PA0~PA3), 灌电流能力为 L0 档。

## 13. 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。

### 13.1. 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；

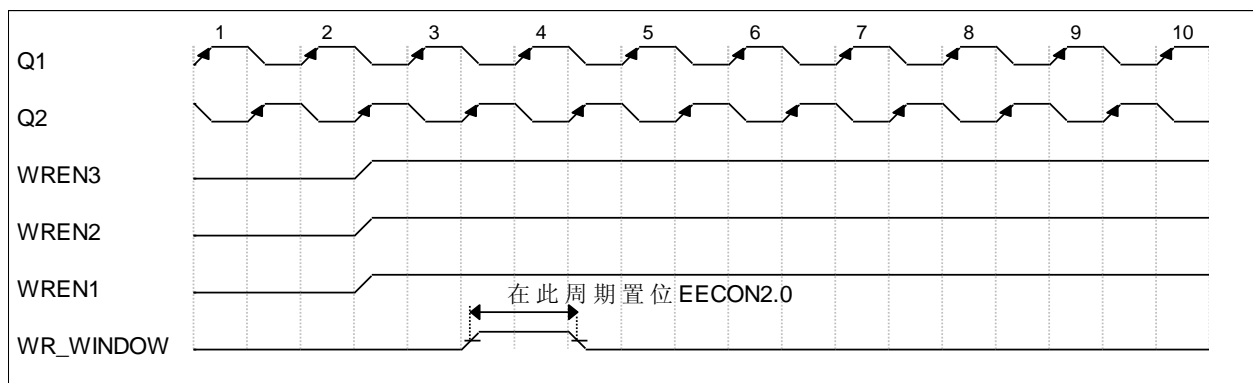


图 13.1 软件编程数据 EEPROM 时序

## 13.2. 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。在紧接着的下一周期，EEDAT 寄存器就被 EEPROM 数据写入。因此该数据可由下一条指令读取。EEDAT 将保持这个值直到用户下一次从该单元读取或向该单元写入数据时（在写操作过程中）。

下面是读取 EEPROM 的一段示例程序：

```
BANKSEL EEADR
LDWI dest_addr
STR EEADR
BSR EECON1, RD
LDR EEDAT, W
```

## 13.3. 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。

### 13.4. 与数据 EEPROM 相关寄存器汇总

名称	地址	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	复位值
INTCON	0x0B/8B	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
PIE1	0x8C	EEIE	CKMIE	LVDIE	—	—	OSFIE	TMR2IE	—	000- 000-
PIR1	0x0C	EEIF	CKMIF	LVDIF	—	—	OSFIF	TMR2IF	—	000- 000-
EEDAT	0x9A	EEDAT[7:0]								0000 0000
EEADR	0x9B	EEADR[7:0]								0000 0000
EECON1	0x9C	—	—	WREN3	WREN2	WRERR	WREN1	—	RD	--00 x0-0
EECON2	0x9D	—	—	—	—	—	—	—	WR	---- ----0

#### 13.4.1. EEDAT 寄存器，地址 0x9A

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

#### 13.4.2. EEADR 寄存器，地址 0x9B

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



### 13.4.3. EECON1 寄存器，地址 0x9C

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

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

### 13.4.4. 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

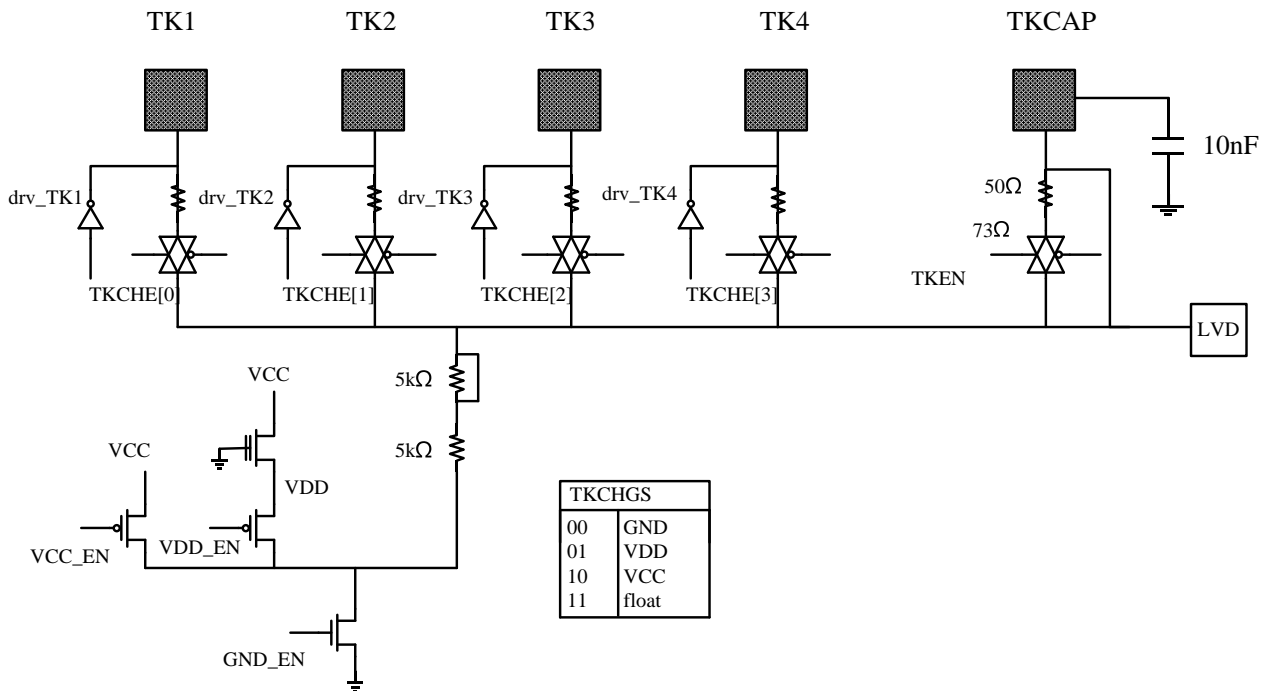
## 14. TOUCH 模块

### 14.1. 触摸按键功能

该系列单片机均提供多触控按键功能，触控按键功能完全内部集成不需外接元件，可通过对内部寄存器的简单操作来实现此功能。

### 14.2. 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。按键被分成四组，每组为一个模块，编号为 TK1~TK4。



14.1 触摸检测原理框图

### 14.3. TKCON 寄存器，地址 0x86

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

Bit	Name	Function
7	—	保留位，读 0
6:5	TKCHGS	Touchkey 预充电选择位 00: GND 01: VDD 10: VCC 11: float
4	TKEN	Touch 模块的使能开关 1: 开启 0: 关闭
3:0	TKCHE	Touchkey 通道选择开关 1: 开启 0: 关闭

## 15. 芯片的电气特性

### 15.1. 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 .....	$V_{SS}-0.3V \sim V_{DD}+0.3V$

### 15.2. 内置高频振荡器（HIRC）

电气参数	最小值 <sup>(1)</sup>	典型值	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%	—	—	-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	—	51	—	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。

### 15.3. 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	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	-3.5%	—	1.0%	—	25°C, $V_{DD} = 1.9\sim 5.5V$
$I_{LIRC}$ 工作电流	—	2.0	—	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。

## 15.4. 低电压复位电路（LVR）

电气参数	最小值 <sup>(1)</sup>	典型值	最大值 <sup>(1)</sup>	单位	条件/备注
I <sub>LVR</sub> 工作电流	—	13.54	—	μA	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		
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。

## 15.5. 低电压侦测电路（LVD）

电气参数	最小值 <sup>(1)</sup>	典型值	最大值 <sup>(1)</sup>	单位	条件/备注
I <sub>LVD</sub> 工作电流	—	21.4	—	μA	V <sub>DD</sub> = 3.3V
V <sub>LVD</sub> , LVD 阈值	1.94	2.0	2.06	V	25°C
	2.33	2.4	2.47		
	2.72	2.8	2.88		
	2.91	3.0	3.09		
	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。

## 15.6. 上电复位电路（POR）

电气参数	最小值	典型值	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

## 15.7. I/O PAD 电路

电气参数		最小值	典型值	最大值	单位	条件/备注
$V_{IL}$		0	—	$0.3 \cdot V_{DD}$	V	
$V_{IH}$		$0.7 \cdot V_{DD}$	—	$V_{DD}$	V	
漏电流		-1	—	1	$\mu A$	5V
源电流(source)	PA4, PA5	L0	—	-3	mA	25°C, $V_{DD} = 5V$ , $V_{OH} = 4.5V$
		L1	—	-6		
		L2	—	-24		
	PA0, PA1, PA2, PA3	—	-18	—		
沉电流(sink)	L0	—	53	—	mA	25°C, $V_{DD} = 5V$ , $V_{OL} = 0.5V$
	L1	—	67	—		
上拉电阻		—	20	—	k $\Omega$	

(1) 数据基于特性值，并未生产测试。源电流(source)

## 15.8. 总体工作电流 ( $I_{DD}$ )

电气参数	Sysclk	典型值@ $V_{DD}^{(1)}$			单位
		2.0V	3.0V	5.0V	
正常模式, $I_{DD}$	16MHz	0.358	0.541	0.554	mA
	8MHz	0.283	0.424	0.432	
	4MHz	0.245	0.364	0.371	
	2MHz	0.226	0.334	0.340	
	1MHz	0.217	0.320	0.326	
	32kHz	0.142	0.211	0.215	
休眠模式 (Sleep, WDT OFF, LVR OFF), $I_{SB}$	—	0.077	0.099	0.112	$\mu A$
休眠模式 (Sleep, WDT ON, LVR OFF)	—	1.163	2.464	2.574	
休眠模式 (Sleep, LVR ON, WDT OFF)	—	12.116	17.318	22.299	
休眠模式 (Sleep, LVR ON, WDT ON)	—	13.094	19.494	24.505	
休眠模式 (Sleep, LVD ON, LVR OFF, WDT OFF)	—	18.454	21.402	24.840	

(1) 数据基于特性值，并未生产测试。

注意：

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

## 15.9. AC 电气参数

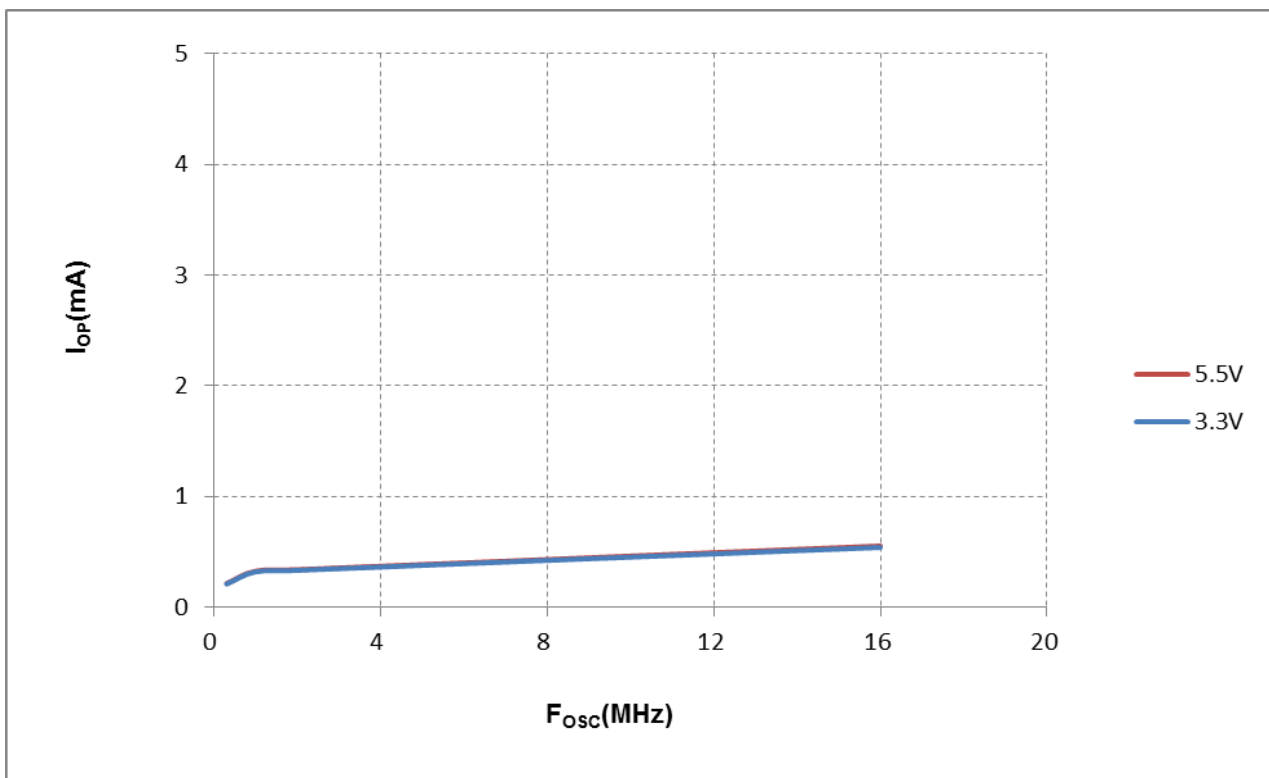
电气参数		最小值	典型值	最大值	单位	条件/备注
指令周期 ( $T_{INS}$ )	4T	250	—	—	ns	系统时钟 HIRC
	4T	122	—	—	$\mu$ s	系统时钟 LIRC
TOCKI 输入周期		$(T_{INS}+40)/N$ 和 20 中较大者	—	—	ns	N = 预分频值 (2, 4, ..., 256)
上电复位保持时间 ( $T_{DRH}$ )		—	4.2	—	ms	25°C, PWRT disable
外部复位脉冲宽度 ( $T_{MCLRB}$ )		2000	—	—	ns	25°C
WDT 周期 ( $T_{WDT}$ )		—	1	—	ms	无预分频, WDTPS<3:0>=0000

注：除特殊说明，特性测试条件为： $T=-40\sim 85^{\circ}\text{C}$ ， $V_{DD}=1.9\sim 5.5\text{V}$ 。

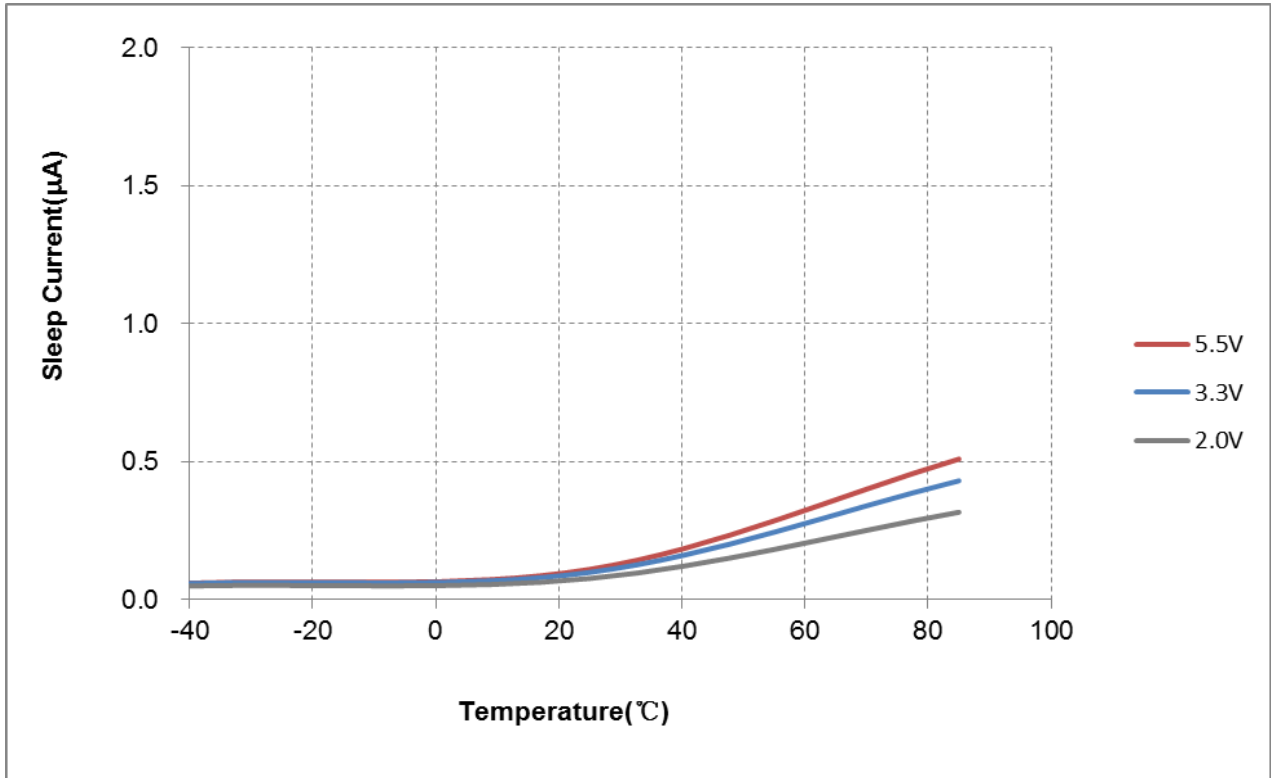
## 15.10. 直流和交流特性曲线图

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。

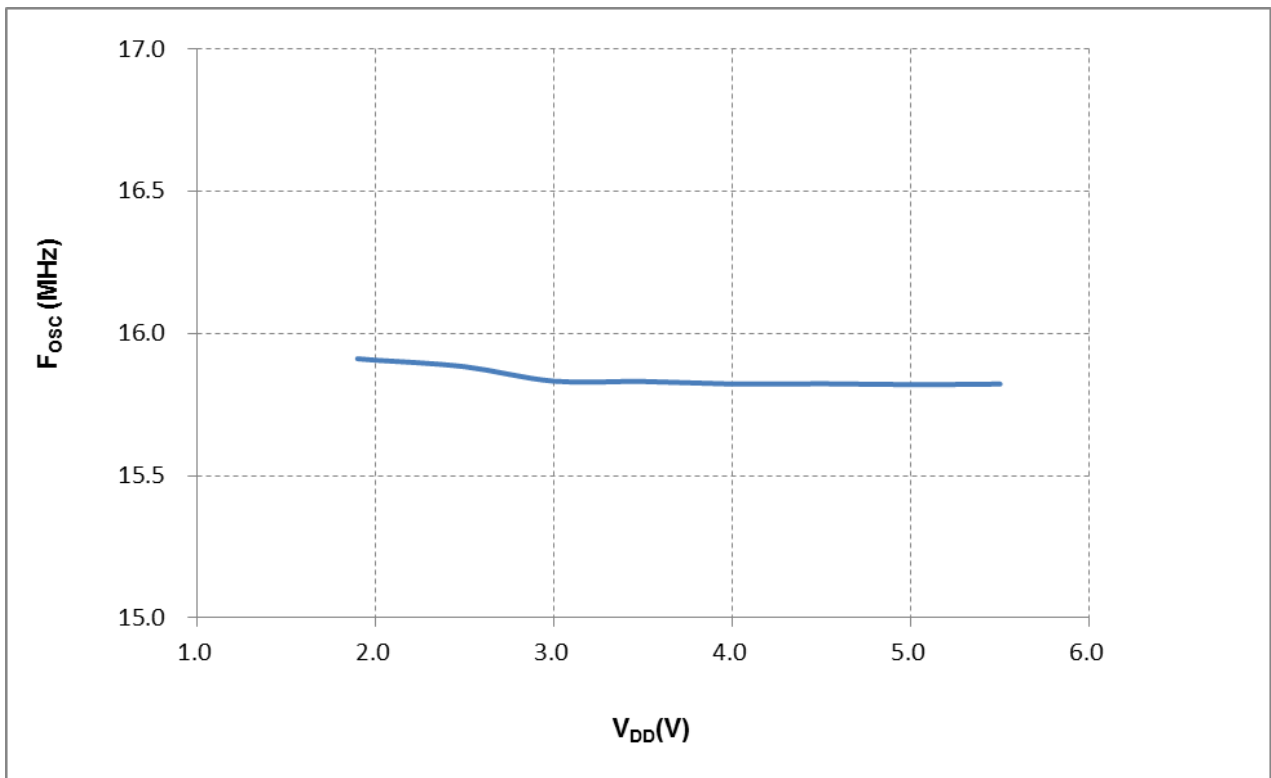
### 15.10.1. 不同 $V_{DD}$ 下， $I_{DD}$ vs Freq ( $T_A=25^{\circ}\text{C}$ )



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

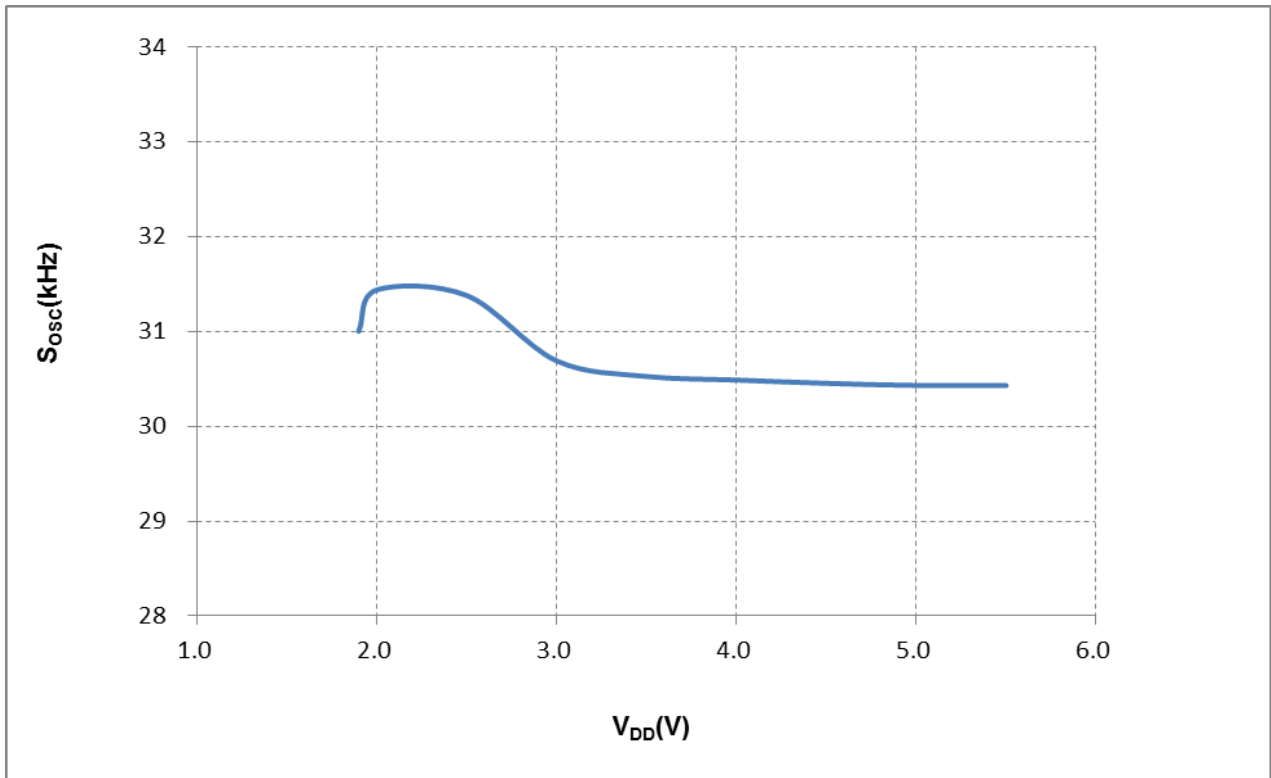


### 15.10.3. HIRC vs $V_{DD}$ ( $T_A=25^\circ\text{C}$ )

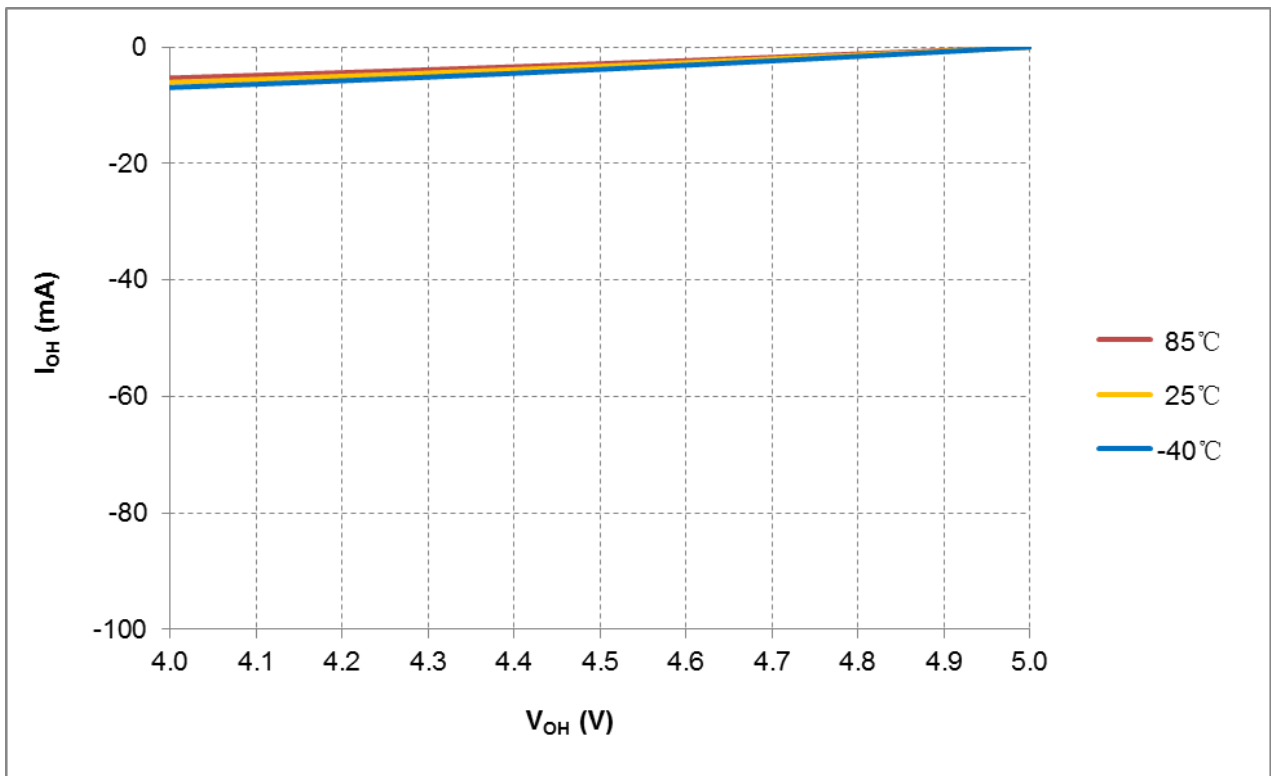




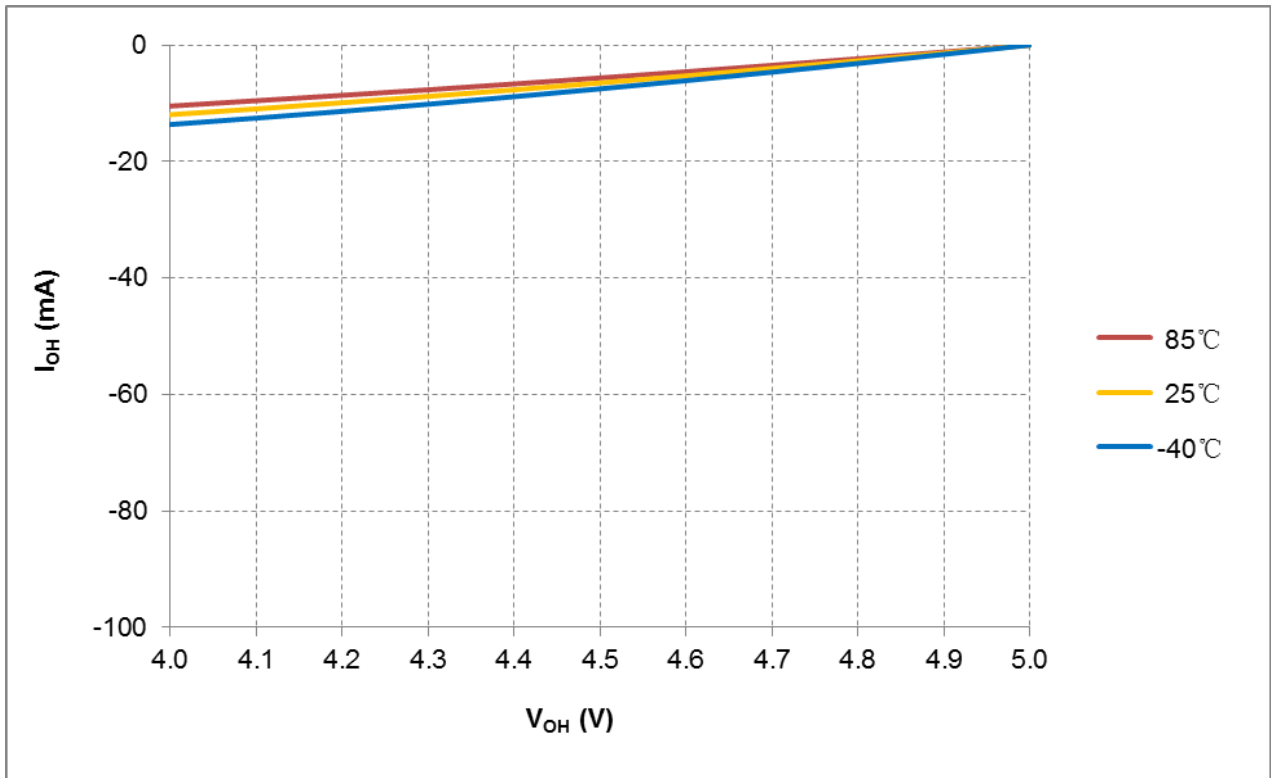
### 15.10.4. LIRC vs $V_{DD}$ ( $T_A=25^\circ\text{C}$ )



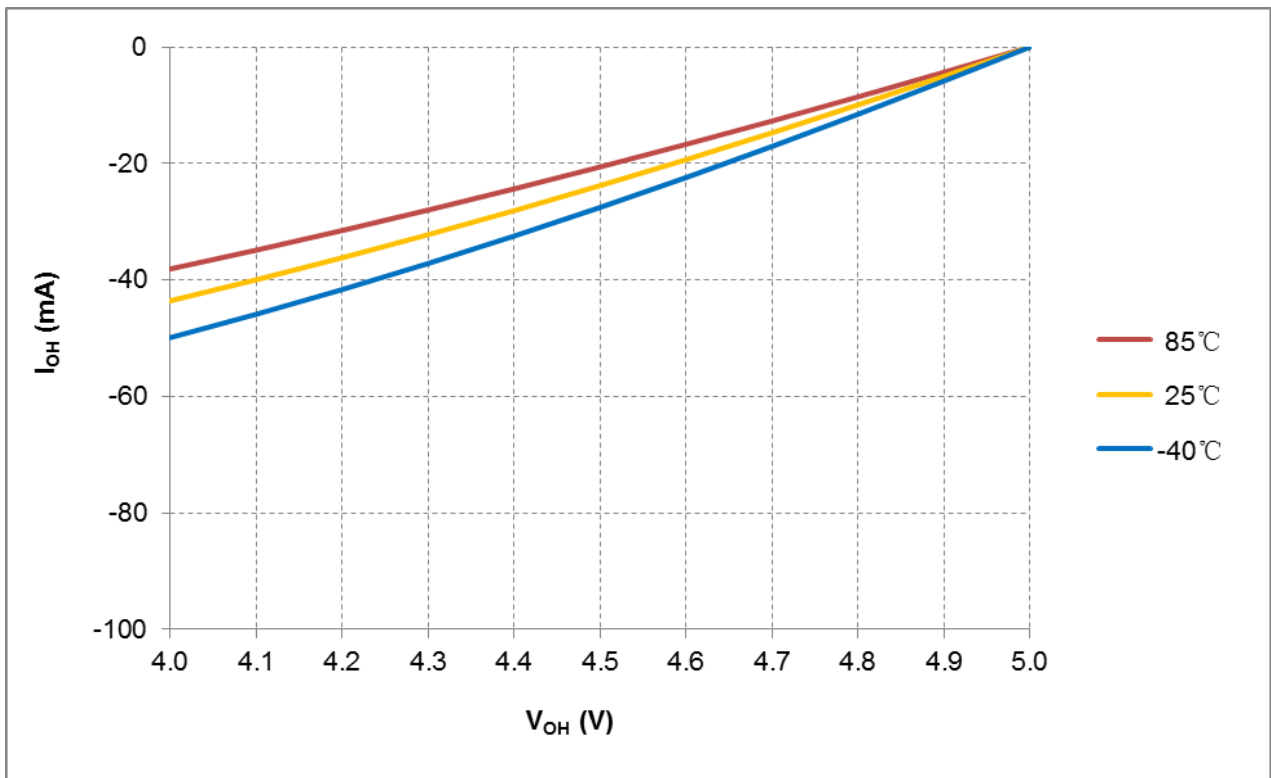
### 15.10.5. $I_{OH}$ ( level -3mA ) vs $V_{OH}$ @ $V_{DD}=5V$



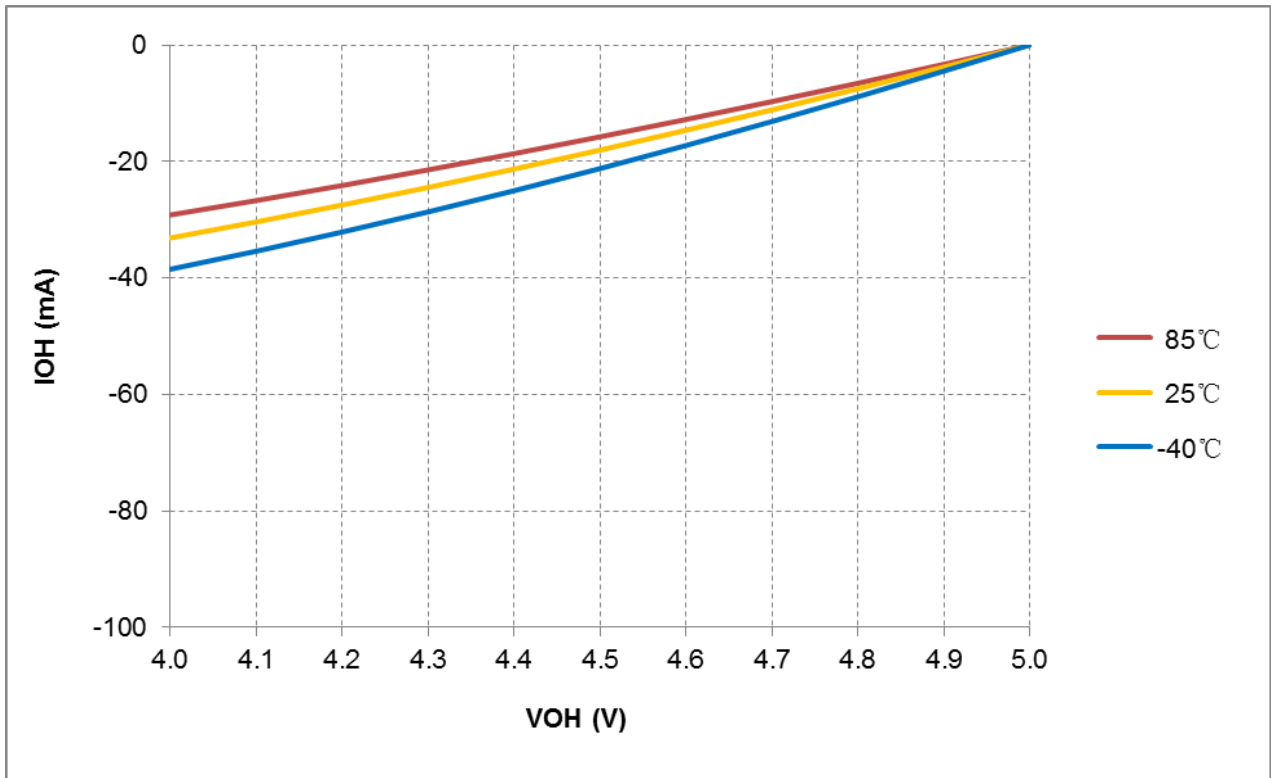
**15.10.6.  $I_{OH}$  ( level -6mA ) vs  $V_{OH}$  @ $V_{DD}=5V$**



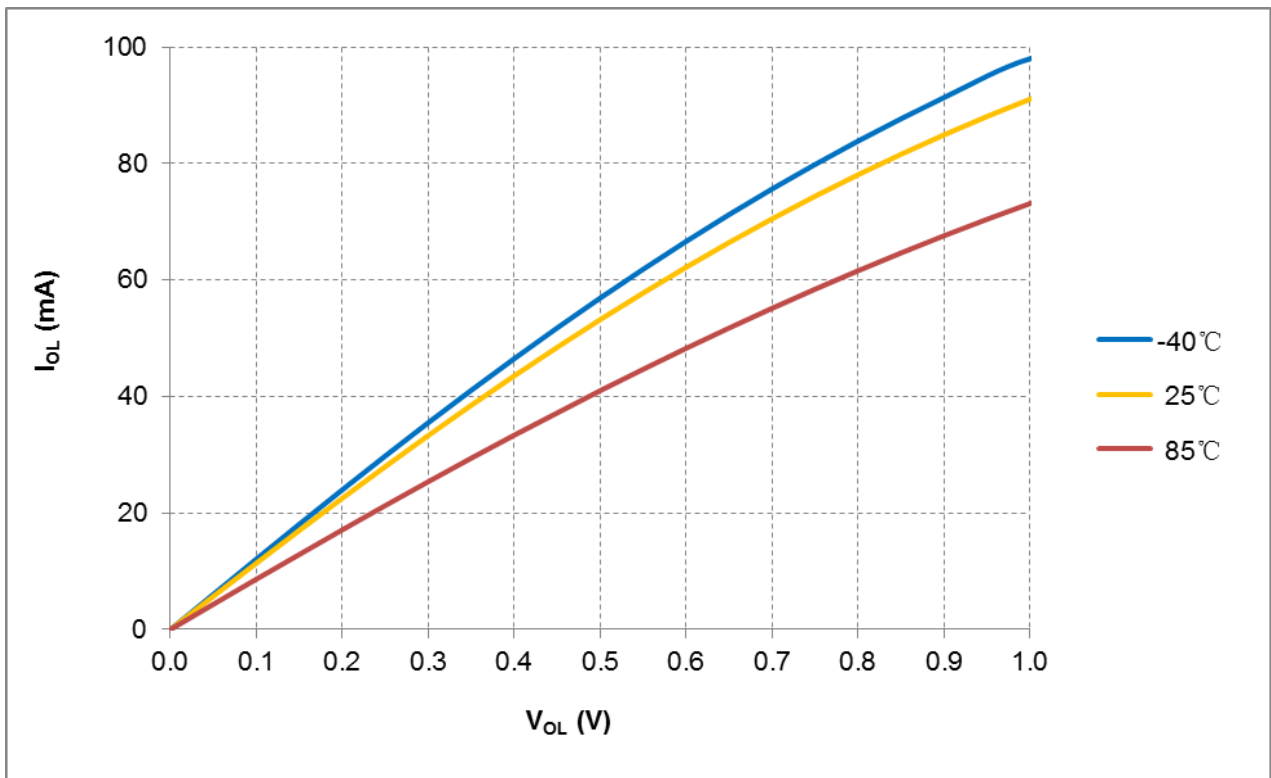
**15.10.7.  $I_{OH}$  ( level -24mA ) vs  $V_{OH}$  @ $V_{DD}=5V$**



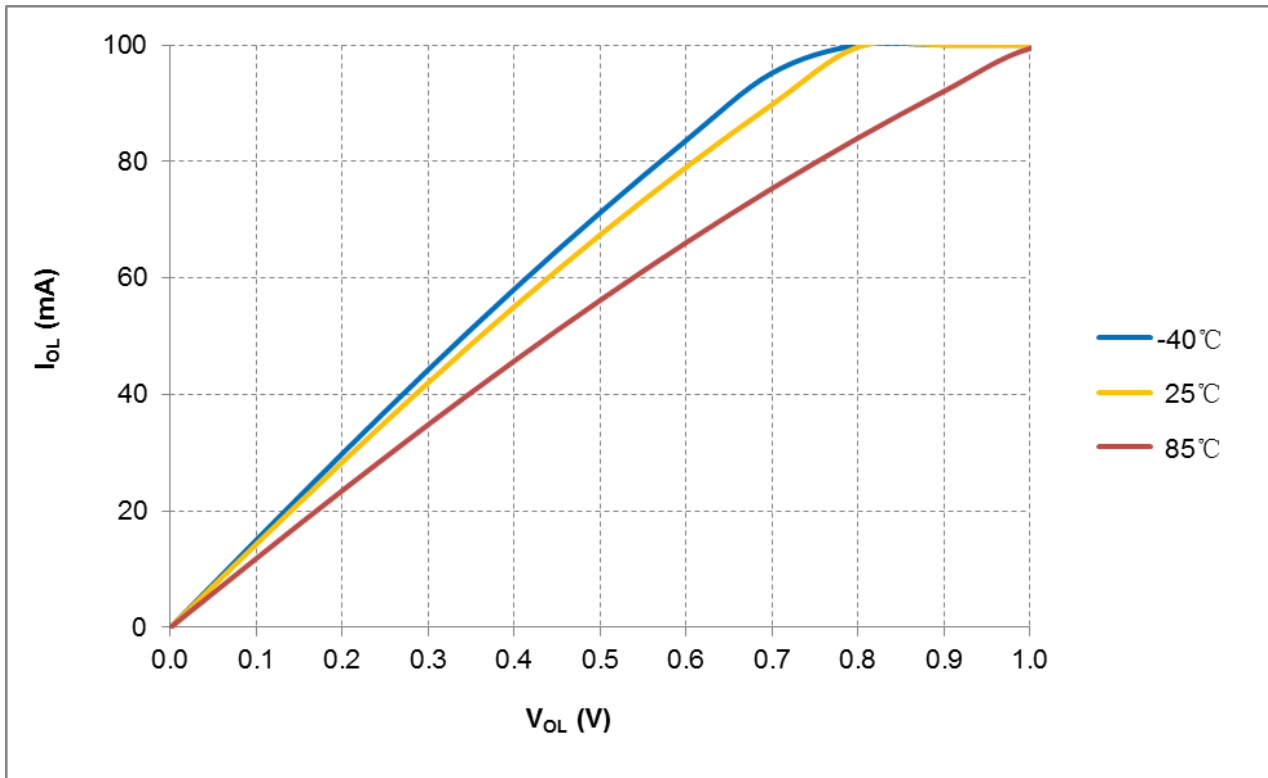
**15.10.8.  $I_{OH}$  ( level -18mA ) vs  $V_{OH}$  @ $V_{DD}=5V$**



**15.10.9.  $I_{OL}$  ( L0 53mA ) vs  $V_{OL}$  @ $V_{DD}=5V$**



### 15.10.10. $I_{OL}$ ( L1 67mA ) vs $V_{OL}$ @ $V_{DD}=5V$



## 16. 指令集列表

本芯片采用精简指令架构，一共 37 条指令，以下是各指令的描述。

汇编语法	功能	运算	状态位
BCR R, b	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

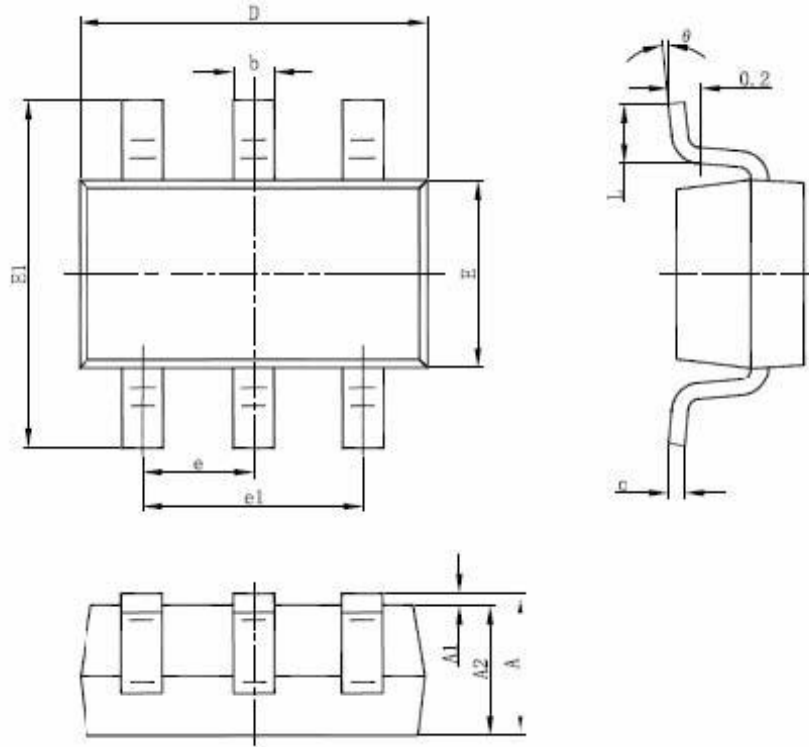
注意:

1. 在 FT60F21x、FT62F21x 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；

## 17. 芯片封装信息

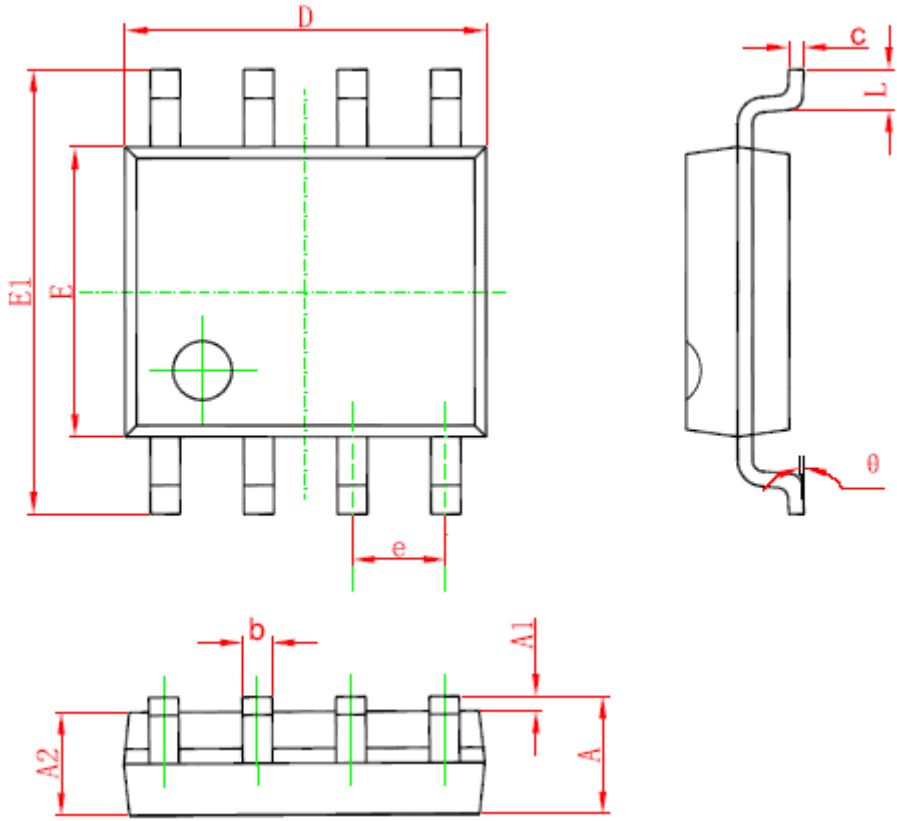
本芯片采用 SOT23-6 及 SOP8 封装方式，具体封装尺寸信息如下：

SOT23-6 封装尺寸：



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	1.050	1.250	0.041	0.049
A1	0.000	0.100	0.000	0.004
A2	1.050	1.150	0.041	0.045
b	0.300	0.500	0.012	0.020
c	0.100	0.200	0.004	0.008
D	2.820	3.020	0.111	0.119
E	1.500	1.700	0.059	0.067
E1	2.650	2.950	0.104	0.116
e	0.950 (BSC)		0.037 (BSC)	
e1	1.800	2.000		0.079
L	0.300	0.600		0.024
θ	0°	8°	0°	8°

SOP8 封装尺寸:



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°



## 附录 1， 文档更改历史

日期	版本	内容
2019-8-29	1.00	初版

\* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices, Incorporated (BVI) assumes no responsibility for the consequences of use of such information or for any infringement of patents of other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices, Incorporated (BVI). Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices, Incorporated (BVI) products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices, Incorporated (BVI). The FMD logo is a registered trademark of Fremont Micro Devices, Incorporated (BVI). All other names are the property of their respective own.