

# DATA SHEET

**ENROO** 英锐恩

**EN8F683**

用户参考手册

Version1.5

June 12, 2015

---

The Information provided by ENROO Electronics is believed to be accurate and reliable. However, ENROO Electronics assumes no responsibility, for possible errors or omissions, or for any consequences resulting from the use of the information contained herein. ENROO Electronics reserves the right to change this documentation without prior notice.

# 1.0 器件概述

## 8位CMOS 单片机

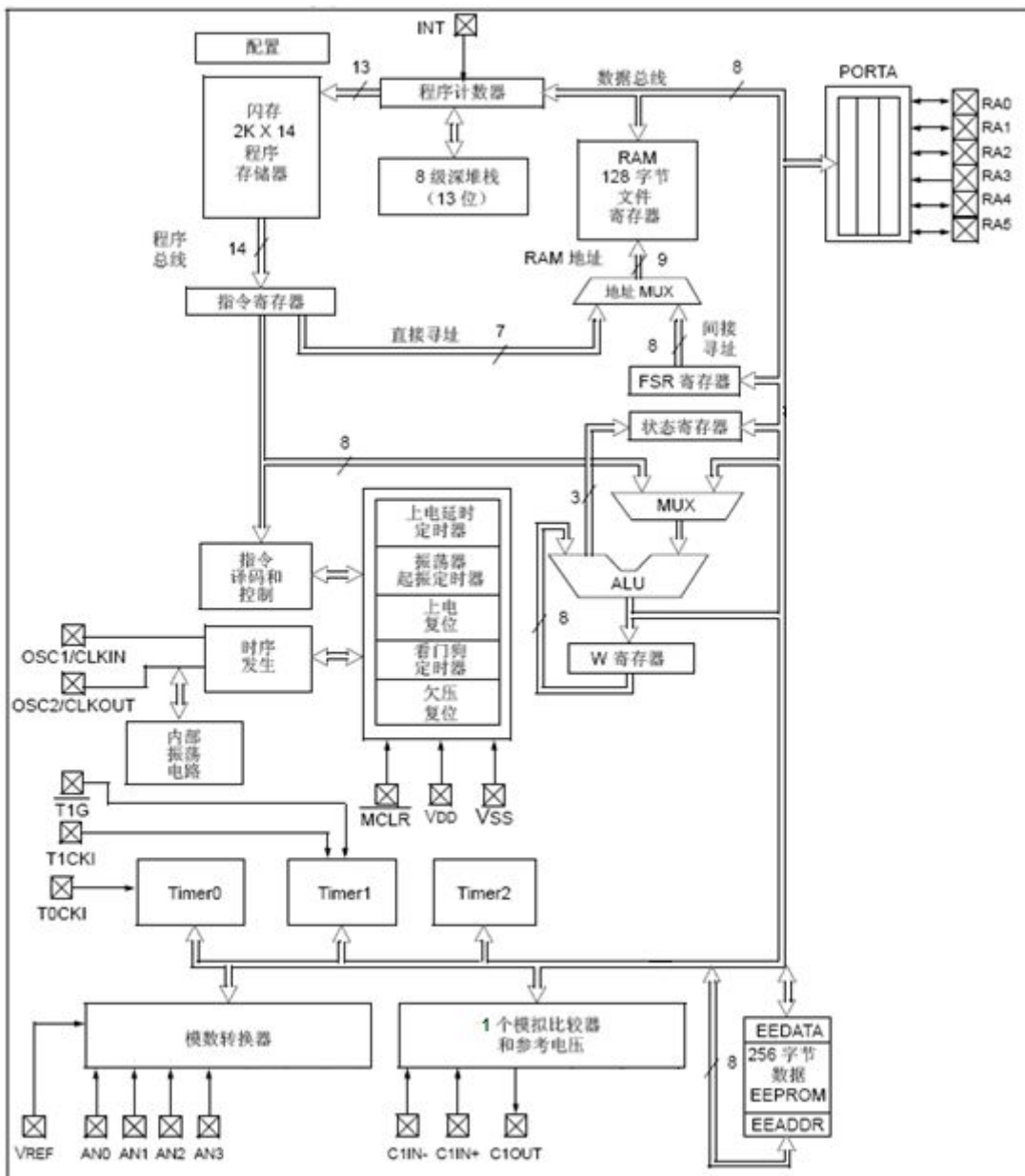
- ◆ 存储器:
  - FLASH 空间: 2K\*14 位。可经受 10 万次写操作。
  - EEPROM 空间: 256 字节。可经受 100 万次写操作。
  - SRAM 空间: 128 字节。
- ◆ 8 级深硬件堆栈
- ◆ I/O 引脚配置
  - 具有独立方向控制的 6 个 I/O 引脚: PA 口 6 个。
  - 高灌/拉电流可直接驱动 LED。
  - PA 端口引脚电平变化中断。
  - PA 端口独立的可编程弱上拉。
- ◆ 定时器
  - Timer0: 带 8 位可编程预分频器的 8 位定时器/计数器。
  - 增强型 Timer1:
    - 带有预分频器的 16 位定时器/计数器。
    - 外部 Timer1 门控 (计数使能)。
    - 如果选择了 INTOSC 模式, 或者在 LP 模式下可选择使用 OSC1 和 OSC2 作为 Timer1 的振荡器。
  - Timer2: 带 8 位周期寄存器、预分频器和后分频器的 8 位定时器/计数器。
- ◆ 增强型捕捉、比较和 PWM 模块
  - 16 位捕捉模块, 最大分辨率为 12.5ns。
  - 16 位比较模块, 最大分辨率为 200ns。
  - 带有 1、2 或 4 路输出通道, 可编程“死区时间”的 10 位 PWM 模块, 输出信号的最大频率为 20kHz。
- ◆ 4 通道 10 位 ADC
  - 4 个外部 ADC 输入。
  - ADC 参考电压可用软件选择为内部或外部参考。当选择内部参考时, 可通过配置寄存器选择片内 2V, 3V, 4V 或者 VDD 作为参考电压。
- ◆ 一个模拟比较器 可在片上编程的比较器的参考电压 (CVREF) 模块 (参考电压一定是小于 VDD, 最低可以是 0V)。比较器输入和输出可直接访问。
- ◆ 双时钟系统 外部高速时钟: 高达 20MHz。
  - 内部高速时钟: 8MHz RC (Fcpu 仅支持 8MHz、4MHz、2MHz、500KHz、250KHz、125KHz)。
  - 内部低速时钟: RC 振荡器 31KHz。
- ◆ 高性能的 RISC CPU 仅需学习 35 条指令。除跳转指令外的所有指令都是单周期的。直接、间接和相对寻址模式。
- ◆ 特殊特性
  - 高精度内部振荡器, 出厂时精度校准为 +/-1%。
  - 可用软件选择的频率范围为 125kHz 到 8MHz。
  - 软件可选的 31kHz 内部振荡器。
  - 节能的休眠模式。
  - 宽工作电压范围 (2.0V 到 5.5V)。
  - 工业级温度范围。
  - 上电复位 (Power-onReset, POR)。
  - 上电延时定时器 (Power-upTimer, PWRT) 和振荡器起振定时器 (OscillatorStart-upTimer, OST)。
  - 带软件控制选择的 PED 低电压侦测选择 (侦测电压有 1.7V、2.1V、3.8V 可选, 而且侦测电压不够准)。
  - 带片上振荡器 (振荡器频率可由软件选择, 当预分频比最大时其标称值为 268 秒) 并且可软件使能的增强型低电流看门狗定时器 (WatchdogTimer, WDT)。
  - 带上拉的主复位, 可复用为输入引脚。
  - 可编程代码保护。
- ◆ 低功耗特性
  - 待机电流:
    - 电压为 2.0V 时, 典型值 50nA。
  - 工作电流:
    - 频率为 4MHz、电压为 2.0V 时, 典型值为 220uA。
    - 频率为 32kHz、电压为 2.0V 时, 典型值为 11uA。
  - 看门狗定时器电流:
    - 电压为 2.0V 时, 典型值为 1uA。
- ◆ 通过两个引脚 (PA0, PA1) 实现在线串行编程和调试

器件	ROM	RAM		I/O	10位A/D转换器 (通道数)	Timer	PWM	比较器	唤醒功能 引脚数目	封装
	FLASH (word)	SRAM (word)	EEPROM (word)							
EN8F683	2048	128	256	6	4	3	4	1	6	SOP8、DIP8、DFN8

<b>1.0 器件概述</b> .....	<b>1</b>
1.1 系统结构图.....	4
1.2 封装脚位图.....	5
1.3 引脚说明.....	6
<b>2.0 存储器构成</b> .....	<b>7</b>
2.1 程序存储器构成.....	7
2.2 数据存储器构成.....	8
2.3 数据 EEPROM 存储器.....	14
2.4 PCL 和 PCHLAT.....	18
2.5 间接寻址、IAR 和 MSR 寄存器.....	20
2.6 看门狗定时器 (WDT).....	20
<b>3.0 复位</b> .....	<b>22</b>
3.1 概述.....	22
3.2 上电复位.....	25
3.3 MCLR.....	25
3.4 看门狗复位.....	25
3.5 上电延时定时器 (PWRT).....	25
3.6 欠压复位.....	26
3.7 延时时序.....	27
3.8 电源控制 (PSTA) 寄存器.....	28
<b>4.0 系统时钟</b> .....	<b>29</b>
4.1 概述.....	29
4.2 振荡器控制.....	30
4.3 时钟源模式.....	30
4.4 外部时钟模式.....	31
4.5 内部时钟模式.....	33
4.6 时钟切换.....	36
4.7 双速时钟启动模式.....	37
4.8 掉电模式 (休眠).....	38
4.9 故障保护时钟监控器.....	40
<b>5.0 中断</b> .....	<b>42</b>
5.1 中断特殊功能寄存器.....	44
5.2 PA2/INT 中断.....	47
5.3 TIMER0 中断.....	47
5.4 PORTA 电平变化中断.....	47
5.5 EEPROM 写数据中断.....	48
5.6 A/D 中断.....	48
5.7 中断的现场保护.....	48
<b>6.0 I/O 端口</b> .....	<b>49</b>
6.1 PORTA 和 CPIOA 寄存器.....	49
6.2 PORTC 和 CPIOC 寄存器.....	50
6.3 其它引脚功能.....	51
6.4 PORTA 引脚说明和引脚图.....	54
6.5 PORT C 引脚说明和引脚图.....	60
<b>7.0 定时器</b> .....	<b>64</b>
7.1 TIMER0 模块.....	64
7.2 带门控的 TIMER1 模块.....	66

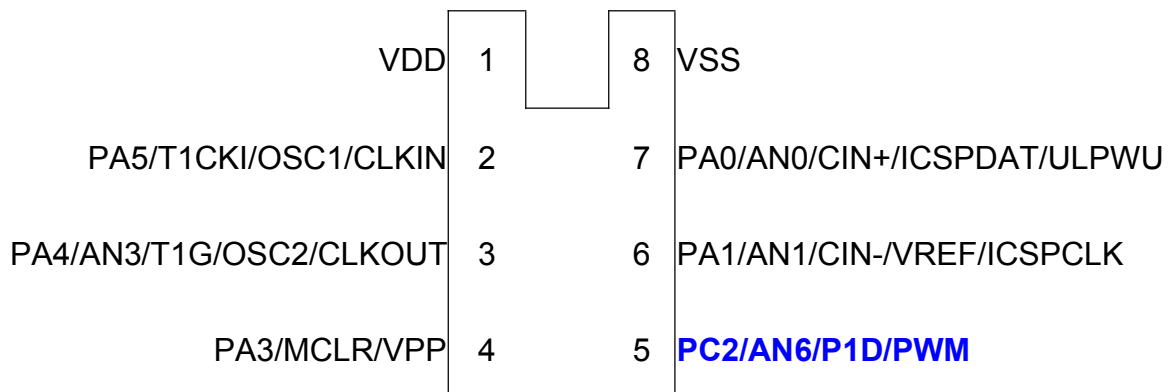
7.3	TIMER2 模块	72
<b>8.0</b>	<b>增强型捕捉/比较/PWM+（带自动关闭和死区）模块</b>	<b>75</b>
8.1	表：ECCP 模式——所需的定时器资源	75
8.2	寄存器：CCP1CON：增强型 CCP1 控制寄存器	75
8.3	捕捉模式	76
8.4	比较模式	78
8.5	PWM 模式	80
8.6	PWM（增强模式）	83
<b>9.0</b>	<b>模拟数字转换器（ADC）模块</b>	<b>92</b>
9.1	图：ADC 框图	92
9.2	ADC 的配置	92
9.3	ADC 的工作原理	95
9.4	ADC 寄存器定义	97
9.5	A/D 采集时间要求	100
<b>10.0</b>	<b>比较器模块</b>	<b>102</b>
10.1	比较器概述	102
10.2	比较器的配置（图）	105
10.3	比较器的控制	106
10.4	比较器响应时间	107
10.5	比较器中断的工作原理	107
10.6	休眠时的工作原理	108
10.7	复位的影响	109
10.8	比较器 C2 门控 TIMER1	110
10.9	比较器 C2 输出与 TIMER1 同步	110
10.10	比较器参考电压	111
<b>11.0</b>	<b>配置寄存器</b>	<b>114</b>
11.1	OPTION ROM	114
11.2	CONFIG OPTION	114
<b>12.0</b>	<b>指令表</b>	<b>116</b>
<b>13.0</b>	<b>电气特性</b>	<b>117</b>
13.1	绝对极限参数	117
13.2	直流电器特性	119
13.3	交流电气特性	120
<b>14.0</b>	<b>开发支持</b>	<b>120</b>
14.1	仿真信息	120
14.2	烧录信息	120
<b>15.0</b>	<b>封装信息</b>	<b>121</b>
15.1	P-DIP 14 PIN	121
15.2	SOP 14 PIN	122
15.3	SOP 16 PIN	123
15.4	QFN 16 PIN	124
<b>16.0</b>	<b>开发说明</b>	<b>128</b>
<b>17.0</b>	<b>联系方式</b>	<b>132</b>

1.1 系统结构图

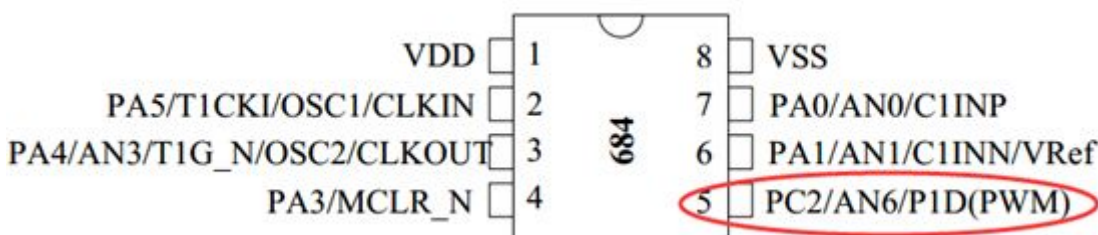
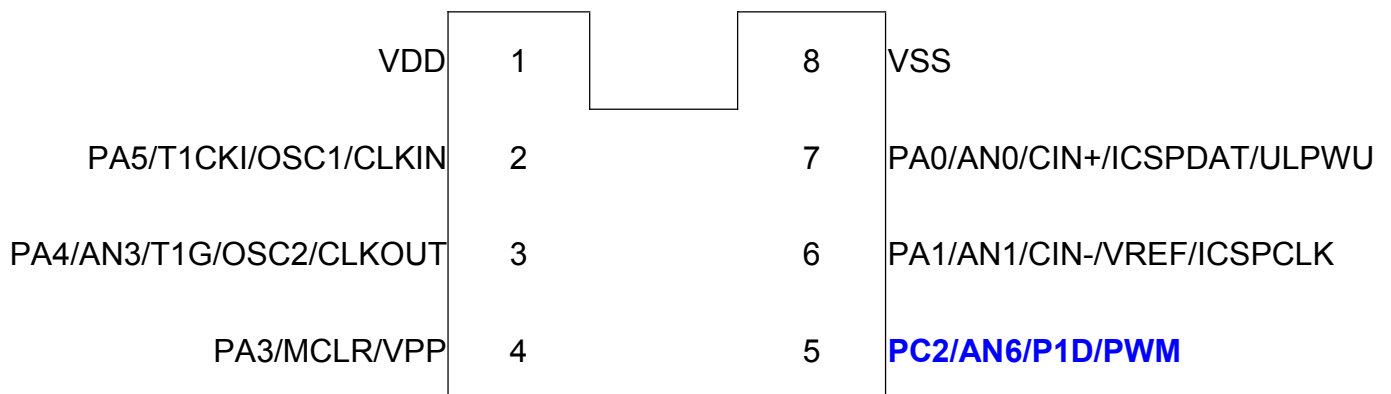


## 1.2 封装脚位图

### 1.2.1 8 引脚图（PDIP、SOIC）



### 1.2.2 8 引脚图（QFN）



### 1.3 引脚说明

I/O	PIN	ANALOG	COMPARATORS	TIMER	CCP	INTERRUPTS	PULL-UPS	BASIC
PA0	7	AN0	CIN+	-	-	IOC	Y	ICSPDAT/ULPWU
PA1	6	AN1/VEREF	CIN-	-	-	IOC	Y	ICSPCLK
PC2	5	AN6			P1D		Y	-
PA3(1)	4	-	-	-	-	IOC	Y(2)	MCLR/VPP
PA4	3	AN3	-	T1G	-	IOC	Y	OSC2/CLKOUT
PA5	2	-	-	T1CKI	-	IOC	Y	OSC1/CLKIN
-	1	-	-	-	-	-	-	VDD
-	8	-	-	-	-	-	-	VSS

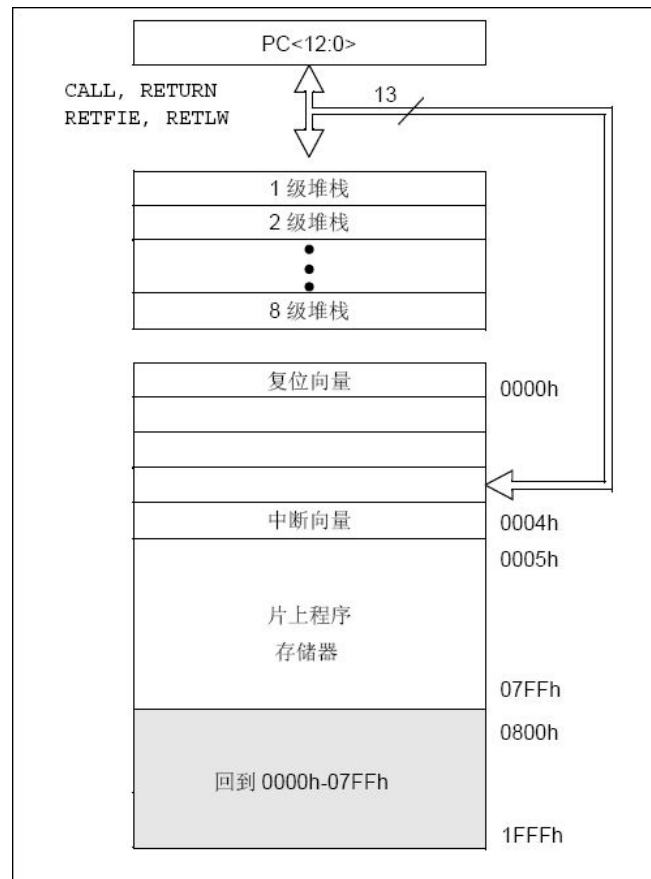
Note 1: Input only.

2: Only when pin is configured for external MCLR.

## 2.0 存储器构成

### 2.1 程序存储器构成

EN8F683 具有一个 13 位程序计数器，可以对 8Kx14 的程序存储空间进行寻址。而 EN8F683 仅在物理上实现了第一个 2Kx14 (0000h-07FFh) 的存储空间。访问该边界以外的单元将导致实际访问存储器的第一个 2Kx14 存储空间。复位向量地址为 0000h，中断向量地址为 0004h。



### 2.1.1 复位向量（0000H）

具有一个字长的系统复位向量（0000H）。

- 上电复位；
- 看门狗复位；
- 外部复位；

发生上述任一种复位后，程序将从 0000H 处重新开始执行，系统寄存器也都将恢复为默认值。根据 STATUS 寄存器中的 TO 和 PD 标志位的内容可以判断系统复位方式。

### 2.1.2 中断向量（0004H）

中断向量的地址为 0004H。一旦有中断响应，程序计数器 PC 的当前值就会存入堆栈缓存器并跳转到 0004H 开始执行中断服务程序。





## 2.2.2.1 特殊功能寄存器位定义 (BANK0)

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR 和 BOR 的值	
Bank 0											
00	IAR	使用 MSR 的内容对数据存储器进行寻址来寻址此单元 (不是物理寄存器)								xxxx	
01	RTCC	Timer0 模块寄存器								xxxx	
02	PCL	程序计数器 (PC) 的低字节								0000	
03	STATUS	IRP <sup>(1)</sup>	PR1 <sup>(1)</sup>	PAGE	—	—	Z	H	C	0001	
04	MSR	间接数据存储器地址指针								xxxx	
05	PORTA <sup>(2)</sup>	—	—	PA5	PA4	PA3	—	PA1	PA0	--x0	
06	—	未实现								—	
07	PORTC <sup>(2)</sup>	—					PC2	—			xxxx
08	—	未实现								—	
09	—	未实现								—	
0A	PCHLAT	—	—	—	程序计数器高 5 位的写缓冲器					---0	
0B	INTS	G	PE1E	T	I	PAIE	T	INTF	PAIF	0000	
0C	PIFB1	EEIF	ADIF	CCP1IF	C2IF	C1IF	OSFIF	TMR2I	TMRIF	0000	
0D	—	未实现								—	
0E	TMR1L	16 位 TMR1 低字节的保持寄存器								xxxx	
0F	TMR1H	16 位 TMR1 高字节的保持寄存器								xxxx	
10	T1STA	T1GIN	TMR1G	T1CKPS	T1CKPS	T1OSCE	T1SYN	TMR1C	TMR1O	0000	
11	TMR2	Timer2 模块寄存器								0000	
12	T2CON	—	TOUTPS	TOUTPS	TOUTPS	TOUTPS	TMR2O	T2CKPS	T2CKPS	-000	
13	CCPR1L	捕捉/比较/PWM 寄存器 1 的低字节								xxxx	
14	CCPR1H	捕捉/比较/PWM 寄存器 1 的高字节								xxxx	
15	CCP1CO	P1M1	P1M0	DC1B1	DC1B0	CCP1M	CCP1M	CCP1M	CCP1M	0000	
16	PWM1CO	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000	
17	ECCPAS	ECCPAS	ECCPAS	ECCPAS	ECCPAS	PSSAC	PSSAC	PSSBD	PSSBD	0000	
18	WDTCON	—	—	—	WDTPS	WDTPS	WDTPS	WDTPS	SWDTE	---0	
19	CMSTA	C2OUT	C1OUT	C2INV	C1INV	C	CM2	CM1	CM0	0000	
1A	CMCON1	—	—	—	—	—	—	T1GSS	C2SYN	---- --	
1B	—	未实现								—	
1C	—	未实现								—	
1D	—	未实现								—	
1E	ADRESH	左对齐格式下 A/D 结果的高 8 位或右对齐格式下 A/D 结果的高 2 位								xxxx	
1F	ADSO	ADFM	ADVRS	ADVRS	CHS2	CHS1	CHS0	GO/DON	ADON	0000	

图注: - = 未实现单元读为 0, u = 不变, x = 未知, q = 取值视情况而定, 阴影 = 未实现

注 1: IRP 和 RP1 位被保留, 应始终保持这两个位清零。

2: 由 ADINS 寄存器控制的具有模拟功能的端口引脚在复位后立即读为 0, 尽管数据锁存器未定义 (POR) 或不变 (其他复位) 也是如此。

## 2.2.2.2 特殊功能寄存器位定义 (BANK1)

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR 和 BOR 的值
Bank 1										
80	IAR	使用 MSR 的内容对数据存储器进行寻址来寻址此单元 (不是物理寄存器)								xxxx
81	<a href="#">TMR</a>	PAPH	I	TCS	TCE	PSC	P	P	P	1111 1111
82	<a href="#">PCL</a>	程序计数器 (PC) 的低字节								0000
83	<a href="#">STATUS</a>	IRP <sup>(1)</sup>	PR1 <sup>(1)</sup>	PAGE	—	—	Z	H	C	0001
84	MSR	间接数据存储器地址指针								xxxx
85	<a href="#">CPIOA</a>	—	—	CPIOA5	CPIOA4	CPIOA3	—	CPIOA1	CPIOA0	--11
86	—	未实现								—
87	<a href="#">CPIOC</a>	—	—	—	—	—	CPIOC2	—	—	1111 1111
88	—	未实现								—
89	—	未实现								—
8A	<a href="#">PCHLA</a>	—	—	—	程序计数器高 5 位的写缓冲器					---0
8B	<a href="#">INTS</a>	G	PE1E	T	I	PAIE	T	INTF	PAIF	0000
8C	<a href="#">PIEB1</a>	EEIE	ADIE	CCP1I	C2IE	C1IE	OSFIE	TMR2I	TMR1I	0000
8D	—	未实现								—
8E	<a href="#">PSTA</a>	—	—	ULPWU	SBORE	—	—	POR	BOR	--01 --
8F	<a href="#">OSCCO</a>	—	IRCF2	IRCF1	IRCF0	OSTS <sup>(3)</sup>	HTS	L	SCS	-110
90	—	未实现								—
91	<a href="#">ADINS</a>	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111
92	<a href="#">PR2</a>	Timer2 模块周期寄存器								1111 1111
93	—	未实现								—
94	—	未实现								—
95	<a href="#">PAPH<sup>(4)</sup></a>	—	—	PHA5	PHA4	—	PHA2	PHA1	PHA0	--11 -
96	<a href="#">PAINTR</a>	—	—	PAINTR	PAINTR	PAINTR	PAINTR	PAINTR	PAINTR	--00
97	—	未实现								—
98	—	未实现								—
99	<a href="#">VRSTA</a>	VREN	—	VRR	—	VR3	VR2	VR1	VR0	0-0-
9A	<a href="#">EEDATA</a>	EEDATA	EEDATA	EEDATA	EEDATA	EEDATA	EEDATA	EEDATA	EEDATA	0000
9B	<a href="#">EEADR</a>	EEADR	EEADR	EEADR	EEADR	EEADR	EEADR	EEADR	EEADR	0000
9C	<a href="#">EECON</a>	—	—	—	—	WRER	WRE	W	R	----
9D	EECON	EEPROM 控制寄存器 2 (非物理寄存器)								---- --
9E	<a href="#">ADRES</a>	左对齐格式下 A/D 结果的低 2 位或右对齐格式下 A/D 结果的低 8 位								xxxx
9F	<a href="#">ADS1</a>	ADRV	ADCS	ADCS	ADCS	—	—	—	—	0000 --

图注: - = 未实现单元读为 0, u = 不变, x = 未知, q = 取值视情况而定, 阴影 = 未实现

注

- 1: IRP 和 RP1 位被保留, 应始终保持这两个位清零。
- 2: OSCCON 寄存器的 OSTS 位复位为 0, 带双速启动且 LP、HS 或 XT 被选定为振荡器。
- 3: 配置寄存器中的 MCLRE 为 1 时 RA3 上拉使能。

### 2.2.2.3 STATUS 状态寄存器

状态 (STATUS) 寄存器包含:

- ALU 的算术运算状态
- 复位状态
- 数据存储 (SRAM) 的存储区选择位

和其他寄存器一样, 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, 将禁止写这三位。根据器件逻辑, 这些位会被置 1 或清零。此外, 也不能写  $\overline{\text{TF}}$  和  $\overline{\text{PF}}$  位。因此, 当执行一条把状态寄存器作为目标寄存器的指令后, 状态寄存器的结果可能和预想的不一樣。

例如, 执行 `CLRR STATUS` 指令会清零该寄存器的高 3 位并将 Z 位置 1。从而使状态寄存器的值为 “000uu1uu” (其中 u 表示不变)。

因此, 建议仅使用 BCR、BSR、SWAPR 和 STWR 指令来改变状态寄存器, 因为这些指令不影响任何状态位。

注: 1: EN8F683 不使用 STATUS 寄存器的 IRP 和 RP1 位, 并且应该保持这两位清零。建议不要使用这些位, 因为这可能会影响未来产品的向上兼容性。

2: 在减法运算中, C 和 HC 位分别作为借位和半借位标志位。具体示例请参见 SUBWR 和 SUBWI 指令。

寄存器 03H/83H: 状态寄存器 (STATUS)

保留	保留	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	PAGE	$\overline{\text{TF}}$	$\overline{\text{PF}}$	Z	HC	C
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit7 **IRP**: 此位保留, 并且应保持为 0

bit6 **RP1**: 此位保留, 并且应保持为 0

bit5 **PAGE**: 寄存器存储区选择位 (用于直接寻址)

1 = Bank1 (80h-FFh)

0 = Bank0 (00h-7Fh)

bit4  **$\overline{\text{TF}}$** : 超时状态位

1 = 上电后, 执行了 CLRWT 指令或 SLEEP 指令

0 = 发生 WDT 超时溢出

bit3  **$\overline{\text{PF}}$** : 掉电标志位

1 = 上电复位后或执行了 CLRWT 指令

0 = 执行了 SLEEP 指令

bit2 **Z**: 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

bit1

**HC:** 半进位/借位位 (ADDWR、ADDWI、SUBWR和SUBWI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

bit0

**C:** 进位/借位位<sup>(1)</sup> (ADDWR、ADDWI、SUBWR和SUBWI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

注 1: 借位的极性是相反的。减法是通过加上第二个操作数的二进制补码 (Two's Complement) 来实现的。对于移位指令 (RRR和RLR)，此位的值来自源寄存器的最高位或最低位。

### 2.2.2.4 TMR 选项寄存器

选项 (TMR) 寄存器是可读写的寄存器，包含可对以下各项进行配置的各种配置位：

- Timer0/WDT 预分频器分配位
- 外部 PA2/INT 中断
- Timer0
- PORTA 上的弱上拉

注：要为Timer0指定1:1的预分频比，应将TMR寄存器的PSC位置1，以将预分频器分配给WDT。请参见第7.1.4节“软件可编程预分频器”。

寄存器 81H: 选项寄存器 (TMR)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
<b>PAPH</b>	<b>IES</b>	<b>TCS</b>	<b>TCE</b>	<b>PSC</b>	<b>PS2</b>	<b>PS1</b>	<b>PS0</b>
bit7							bit0

图注:

R = 可读位                      W = 可写位                      U = 未实现位，读为0  
 -n = POR时的值                1 = 置1                              0 = 清零                              x = 未知

bit7

**PAPH:** PORTA上拉使能位  
 1 = 禁止PORTA上拉  
 0 = 由各个端口锁存值使能PORTA上拉

bit6

**IES:** 中断边沿选择位  
 1 = 由PA2/INT引脚的上升沿触发中断  
 0 = 由PA2/INT引脚的下降沿触发中断

bit5

**TCS:** RTCC时钟源选择位  
 1 = PA2/T0CKI引脚上信号的跳变  
 0 = 内部指令周期时钟 (FOSC/4)

bit4

**TCE:** RTCC时钟源边沿选择位  
 1 = 在PA2/T0CKI引脚电平发生由高到低的跳变时递增  
 0 = 在PA2/T0CKI引脚电平发生由低到高的跳变时递增

bit3      **PSC:** 预分频器分配位  
1 = 将预分频器分配给WDT  
0 = 将预分频器分配给Timer0模块

bit2-0      **PS<2:0>:** 预分频比选择位

位值	RTCC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

### 2.2.2.5 PSTA 寄存器

bit1

电源控制（PSTA）寄存器包含区分以下复位的标志位：

- 上电复位（POR）
- 欠压复位（BOR）
- 看门狗定时器复位（WDT）
- 外部MCLR复位

PSTA 寄存器也用于控制超低功耗唤醒和BOR的软件使能。

寄存器 8EH：电源控制寄存器1（PSTA）

U-0	U-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-x
—	—	ULPWUE	SBOREN	MCR	IER	POR	BOR
bit7							bit0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-6

未实现：读为0

bit5

**ULPWUE**：超低功耗唤醒使能位

1 = 使能超低功耗唤醒

0 = 禁止超低功耗唤醒

bit4

**SBOREN**：软件欠压检测使能位<sup>(1)</sup>

1 = 使能欠压检测

0 = 禁止欠压检测

bit3

**MCR**：外部复位状态位

1 = 上电复位值

0 = 发生了外部复位

bit2

**IER**：指令错误复位状态位

1 = 发生了指令错误复位

0 = 上电复位值

**POR**：上电复位状态位

1 = 未发生上电复位

0 = 发生了上电复位（必须在上电复位发生后用软件置1）

bit0

**BOR**：欠压复位状态位

1 = 未发生欠压复位

0 = 已发生欠压复位（必须在发生欠压复位后用软件置1）

注

1：当配置字寄存器中的LVREN&lt;1:0&gt; = 01时允许使用该位对BOR进行控制。

## 2.3 数据 EEPROM 存储器

数据 EEPROM 在正常工作期间（整个 VDD 范围）是可读写的。该存储器并不直接映射到文件寄存器空间。而是通过特殊功能寄存器来间接寻址。有以下 4 个 SFR 用于读写该存储器：

- EECON1
- EECON2（可写不可读）
- EEDATA
- EEADR

EEDATA 寄存器存放 8 位要读写的数据，而 EEADR 寄存器存放要被访问的 EEPROM 单元的地址。EN8F683 具有 256 字节的数据 EEPROM，地址范围从 00h 到 FFh。

EEPROM 数据存储器允许以字节为单位进行读写。字节写操作将自动擦除目标存储单元并写入新数据（即先擦后写）。EEPROM 数据存储器可以反复擦写很多次。写入时间由片上定时器控制。写入时间会因为电压、温度及芯片的不同而发生变化。请参见第 13.0 节“电气特性”里的交流规范来了解准确的限制。

如果数据存储器有代码保护，CPU 仍可继续对数据 EEPROM 存储器进行读写操作。器件编程器无法再访问数据 EEPROM 的数据，EEPROM 单元读为零。

寄存器 9AH: EEPROM 数据寄存器 (EEDATA)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>EEDATA7</b>	<b>EEDATA6</b>	<b>EEDATA5</b>	<b>EEDATA4</b>	<b>EEDATA3</b>	<b>EEDATA2</b>	<b>EEDATA1</b>	<b>EEDATA0</b>
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7~0      **EEDATAn**，要从数据EEPROM读取或向数据EEPROM写入的字节值

寄存器 9BH: EEPROM地址寄存器 (EEADR)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>EEADR7</b>	<b>EEADR6</b>	<b>EEADR5</b>	<b>EEADR4</b>	<b>EEADR3</b>	<b>EEADR2</b>	<b>EEADR1</b>	<b>EEADR0</b>
bit7							bit0

bit7~0      **EEADR**: 指定在256个单元中要进行EEPROM读/写操作的一个单元

图注:

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知



### 2.3.1 EECON1 和 EECON2 寄存器

EECON1 是控制寄存器，它的低 4 位是物理实现的。高 4 位未实现，读为 0。

控制位 RD 和 WR 分别启动读操作和写操作。用软件只能将这些位置 1 而无法清零。当读操作或写操作完成后，由硬件将其清零。由于不能用软件将 WR 位清零，可有效防止写操作意外或提前终止。

将 WREN 位置 1 将允许一次写操作。上电时将清零 WREN 位。当正常的写操作被  $\overline{\text{MCLR}}$  复位或 WDT 复位中断时，WRERR 将被置 1。在这些情况下，用户可以在复位后检查 WRERR 位，将其清零并重写相应的单元。数据和地址将被清空。因此，需要重新对 EEDATA 和 EEADR 寄存器进行初始化。

写操作结束时，PIFB1 寄存器的中断标志位 EEIF 将被置 1。此标志位必须用软件清零。

EECON2 不是物理寄存器。读 EECON2 得到的是全 0。EECON2 寄存器仅在数据 EEPROM 写入过程中使用。

注： 当对数据EEPROM进行写操作（WR = 1）时，不应修改EECON1、EEDATA和EEADR寄存器。

寄存器 9CH: EEPROM控制寄存器 (EECON1)

U-0	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
—	—	—	—	WRERR	WREN	WR	RD
bit7							bit0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7<sup>~</sup>4

未实现：读为0

bit3

**WRERR**: EEPROM错误标志位

1 = 写操作提前终止（正常工作期间发生的任何 $\overline{\text{MCLR}}$ 复位、WDT复位或欠压检测）

0 = 写操作完成

bit2

**WREN**: EEPROM写使能位

1 = 允许写周期

0 = 禁止对数据EEPROM进行写操作

bit1

**WR**: 写控制位

1 = 启动写周期（一旦完成写操作，硬件会将该位清零。软件只能将WR位置1而不能清零。）

0 = 写入数据EEPROM的周期完成

bit0

**RD**: 读控制位

1 = 启动EEPROM读操作（读取需要一个周期。RD位由硬件清零。软件只能将RD位置1而不能清零。）

0 = 不启动EEPROM读操作

## 2.3.2 读数据 EEPROM 存储器

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，如例 2-3-2 所示。在紧接着的下一周期，EEDATA 寄存器就有数据了。因此该数据可由下一条指令读取。EEDATA 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）。

例 2-3-2: 读数据 EEPROM

```
LDR  data_addr,W      ;
BSR  STATUS,PAGE     ;Bank1
STWR EEADR           ;Address to read
BSR  EECON1,RD       ;EE Read
LDR  EEDATA,W        ;Move data to W
```

## 2.3.3 写数据 EEPROM 存储器

要写入 EEPROM 数据存储单元，用户必须先把该单元的地址写入 EEADR 寄存器并且把数据写入 EEDATA 寄存器。然后用户必须按特定顺序开始写入每个字节。

例 2-3-3: 写数据 EEPROM

```
LDR  data_addr,W      ;
BSR  STATUS,PAGE     ;Bank 1
STWR EEADR           ;Address to write
BCR  STATUS,PAGE     ;bank 0
LDR  data_buff,W     ;
BSR  STATUS,PAGE     ;BANK1
STWR EEDATA         ;
BSR  EECON1,WREN     ;Enable write
BCR  INTS,GIE        ;Disable INTs
BTSC INTS,GIE        ;
LJUMP $-2           ;
LDWI 0X55           ;Unlock write
STWR EECON2         ;
LDWI 0XAA           ;
STWR EECON2         ;
BSR  EECON1,WR       ;Start the write
BTSC EECON1,WR      ;等待 EEPROM 操作完成
LJUMP $-1           ;
BCR  EECON1,WREN     ;关掉写操作，防止误写入
BSR  INTS,GIE        ;Enable INTS
```

如果没有完全按照以上顺序（将 55h 写入 EECON2，将 AAh 写入 EECON2，然后将 WR 位置 1）逐字节写入，写操作将不会开始。在这个代码段执行过程中，强烈建议禁止中断。在必需序列执行过程中会进行周期计数。当计数值与执行必需序列所需的周期不等时，数据无法写入 EEPROM。

此外，必须将 EECON1 中的 WREN 位置 1 以使能写操作。这种机制可防止由于代码执行错误（异常）（即程序失控）导致误写数据 EEPROM。除非更新 EEPROM，用户应始终保持 WREN 位清零。WREN 位无法由硬件清零。

## 2.3.4 写校验

根据应用情况，将写入数据 EEPROM 的实际值与要写入的值进行核对（见例 2-3-4）是一种很好的编程习惯。

例 2-3-4：写校验

BSR	STATUS,PAGE	;BANK1
LDR	EEDATA,W	;EEDATA not changed from previous write
BSR	EECON1,RD	;Yes, Read the value written
XORWR	EEDATA,W	;
BTSS	STATUS,Z	;Is data the same
LJUMP	WRITE_ERR	;No, handle error
:		;Yes, continue

### 2.3.4.1 使用数据 EEPROM

数据 EEPROM 是高耐久性的并可字节寻址的阵列，已对其进行优化，可存储频繁更改的信息（例如程序变量或其他经常更新的数据）。当一个扇区内的变量频繁变动，而另一个扇区内的变量不变，那么就有可能超出对 EEPROM 的总写入次数（规范 D124）却不超出单个字节的总写入次数（规范 D120 和 D120A）。如果出现这种情况，就必须对阵列进行刷新。出于这个原因，应将不经常变动的变量（如常数、ID 和校准值等）保存在闪存存储器中。

## 2.3.5 避免误写操作

有些情况下，用户可能不希望向数据 EEPROM 存储器写入数据。EEPROM 存储器有各种机制以防对 EEPROM 误写。上电时，WREN 被清零。而且，上电延时定时器（延时 64ms）也可以防止误写 EEPROM。

写操作启动序列和 WREN 位可以共同预防在以下情况下发生误写：

- 欠压
- 电源毛刺
- 软件故障

## 2.3.6 代码保护时数据 EEPROM 的操作

数据存储器的代码保护功能可通过将配置字寄存器（寄存器 CONFIG）中的 CPD 位清零来实现。启用数

据存储器的代码保护功能后，CPU 能够将数据从数据 EEPROM 中读出或向其中写入数据。如果启用了数据存储器的代码保护功能，建议同时启用程序存储器的代码保护功能。这将通过将未使用的程序存储单元编程为 0（这将作为 NOP 执行），避免有人通过编写额外的可以输出数据存储器内容的程序来访问数据存储器。将程序存储器中未使用的单元编程为 0 同样有助于避免数据存储器的代码保护被破坏。

## 2.3.7 与数据 EEPROM 相关的寄存器/位

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
INTS	GIE	PEIE	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000	0000 0000
PIFB1	EEIF	ADIF	CCP1IF	C2IF	C1IF	OSFIF	TMR2IF	TMRIF	0000 0000	0000 0000
PIEB1	EEIE	ADIE	CCP1IE	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000	0000 0000
EEDATA	EEDATA7	EEDATA6	EEDATA5	EEDATA4	EEDATA3	EEDATA2	EEDATA1	EEDATA0	0000 0000	0000 0000
EEADR	EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR3	EEADR1	EEADR0	0000 0000	0000 0000
EECON1	—	—	—	—	WRERR	WREN	WR	RD	---- x000	---- q000
EECON2	EEPROM 控制寄存器 2								---- ----	---- ----

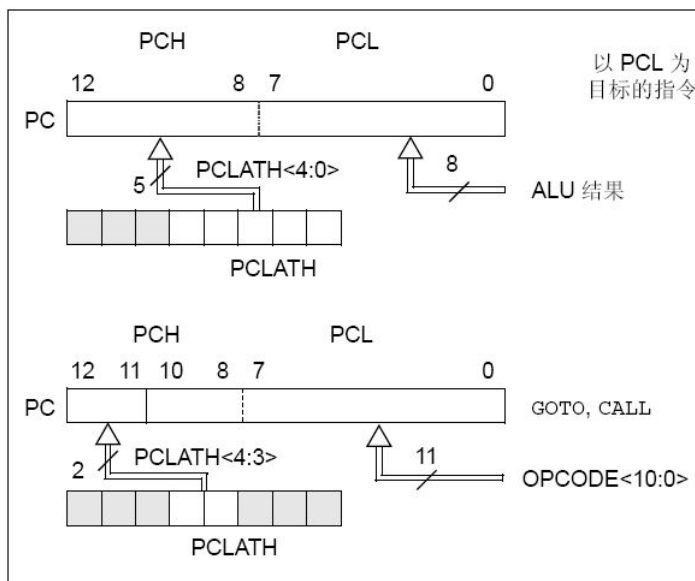
图注： x = 未知， u = 不变， — = 未实现（读为 0）， q = 取值视情况而定。  
数据 EEPROM 模块不使用阴影单元。

注 1: EECON2 不是物理寄存器。

## 2.4 PCL 和 PCHLAT

程序计数器（PC）为 13 位宽。其低 8 位来自可读写的 PCL 寄存器，高 5 位（PC<12:8>）来自 PCHLAT，不能直接读写。只要发生复位，PC 就将被清零。图 2-4 显示了装载 PC 值的两种情形。图 2-4 上方的例子说明在写 PCL（PCHLAT<4:0>→PCH）时是如何装载 PC 的。图 2-4 下方的例子说明了在执行 LCALL 或 LJUMP 指令期间（PCHLAT<4:3>→PCH），是如何装载 PC 的。

图 2-4：在不同情况下装载 PC



### 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的 PC<12:8>位（PCH）被 PCHLAT 寄存器的内容所取代。这样可通过将所需的高 5 位写入 PCHLAT 寄存器来改变程序计数器的所有内容。当低 8 位写入 PCL 寄存器时，程序计数器的所有 13 位将变为 PCHLAT 寄存器中所包含的值以及写入 PCL 寄存器中的值。

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 PCHLAT 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, PCHLAT 必须均必须递增。

## 2.4.2 堆栈

EN8F683 器件具有 8 级深 x13 位宽的硬件堆栈 (见 [图 2-2](#) 和 [图 2-4](#))。堆栈空间既不占用程序存储区空间, 也不占用数据存储区空间, 而且堆栈指针是不可读写的。当执行 CALL 指令或当中断导致程序跳转时, 值 PC 将被压入 (PUSH) 堆栈。而在执行 RET、RTIW 或 RTFI 指令时, 堆栈中的断点地址将从堆栈中弹出 (POP) 到 PC 中。PCHLAT 不受 PUSH 或 POP 操作的影响。

堆栈的工作原理犹如循环缓冲区。这意味着当堆栈压栈 8 次后, 第 9 次压栈的数值将会覆盖第一次压栈时所保存的数值, 而第十次压栈数值将覆盖第二次压栈时保存的数值, 以后依次类推。

注: 1: 不存在指明堆栈是否上溢或下溢的状态标志位。  
2: 不存在被称为 PUSH 或 POP 的指令/助记符。堆栈的压入或弹出是源于执行了 CALL、RET、RTIW 和 RTFI 指令, 或源于指向中断向量地址。

寄存器 02H/82H: 程序计数器低字节控制寄存器 (PCL)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
bit7							Bit0

图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为0  
-n = POR 时的值              1 = 置1                      0 = 清零                      x = 未知

bit7-0              PC<7:0>: 程序计数器低8位。

寄存器 0AH/8AH: 程序计数器高字节控制寄存器 (PCHLAT)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	PC12	PC11	PC10	PC9	PC8
bit7							bit0

图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为0  
-n = POR 时的值              1 = 置1                      0 = 清零                      x = 未知

bit7-5              未实现: 读为0

bit4-0              PC<12:8>: 程序计数器高5位。

## 2.5 间接寻址、IAR 和 MSR 寄存器

IAR 寄存器不是实际存在的寄存器，使用 IAR 寄存器寻址将产生间接寻址。

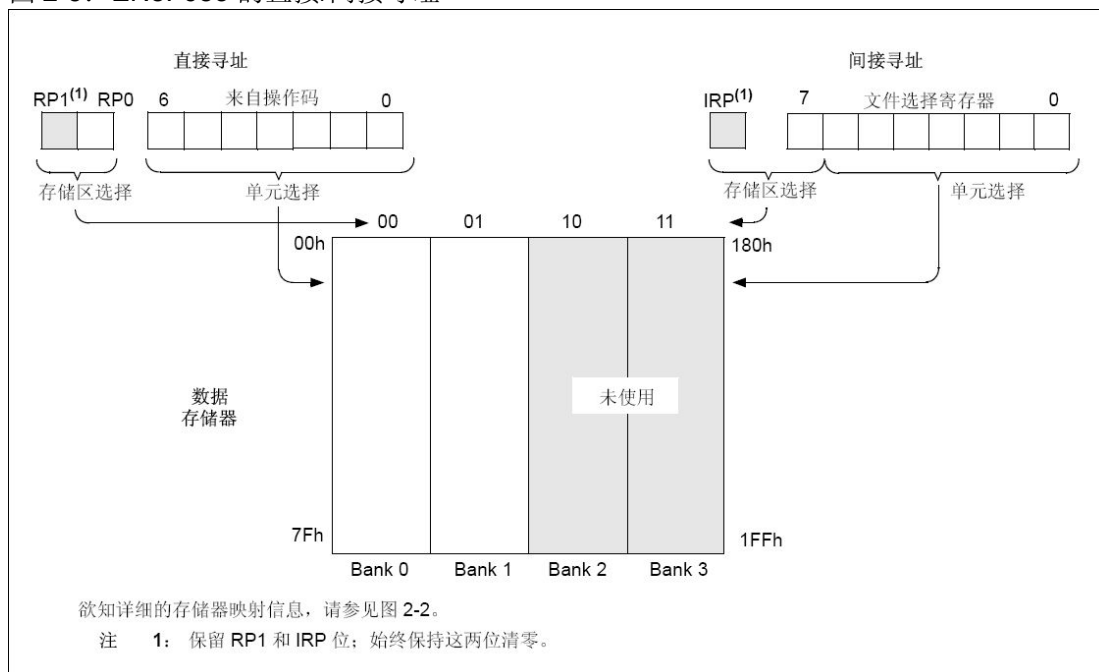
使用 IAR 寄存器可进行间接寻址。任何使用 IAR 寄存器的指令，实际上是对文件选择寄存器（MSR）所指向的数据进行存取。间接对 IAR 进行读操作将返回 00h。间接对 IAR 寄存器进行写操作将导致空操作（尽管可能会影响状态标志位）。通过将 8 位的 MSR 寄存器与 STATUS 寄存器的 IRP 位进行组合可得到一个有效的 9 位地址，如图 2-5 所示。

例 2-5 给出了一个使用间接寻址将 RAM 地址单元 20h-2Fh 清零的简单程序。

例 2-5: 间接寻址

	LDWI	0X20	;initialize pointer
	STWR	MSR	;to RAM
NEXT:	CLRR	IAR	;clear INDF register
	INCR	MSR	;inc pointer
	BTSS	MSR,4	;all done?
	LJUMP	NEXT	;no clear next
CONTINUE:			;yes continue

图 2-5: EN8F683 的直接/间接寻址



## 2.6 看门狗定时器 (WDT)

WDT 具有以下特性:

- 工作于 LFINTOSC (31kHz)
- 包含 16 位预分频器
- 与 Timer0 共用 8 位预分频器
- 超时周期从 1ms 至 268 秒
- 配置位和使用软件控制

WDT 在如表 2-6-1 所述的情况下清零。

## 2.6.1 WDT 振荡器

WDT 的时基来源于 31kHz LFINTOSC。OSCCON 寄存器的 LTS 位不反映 LFINTOSC 是否被使能。

在所有复位时 WDTCON 的值为“---01000”。这样标称时基为 17ms。

注： 当执行振荡器起振定时器（OST）时，WDT保持复位状态，因为OST使用WDT纹波计数器来执行振荡器延时计数。OST计数到期后，WDT将开始计数（如果使能）。

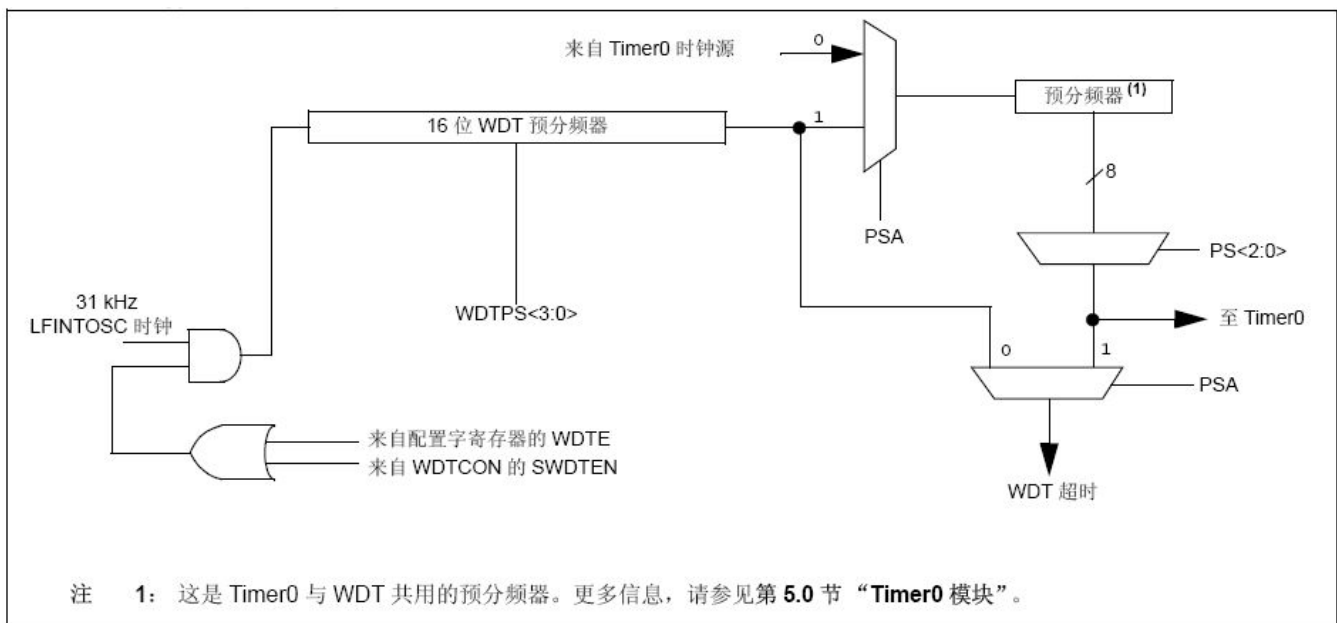
## 2.6.2 WDT 控制

WDTE 位在配置字寄存器中。该位置 1 时，WDT 连续运行。

配置字寄存器中的 WDTE 位置 1 时，WDTCON 寄存器中的 SWDTEN 位不起作用。如果 WDTE 清零，那么 SWDTEN 位可用于使能和禁止 WDT。该位置 1 使能 WDT，该位清零禁止 WDT。

TMR 寄存器的 PSC 和 PS<2:0>位功能与 EN8F683 系列单片机的早期版本相同。更多信息请参见第 7.1 节“Timer0 模块”。

图 2-6：看门狗定时器框图



寄存器 18H：看门狗定时器控制寄存器（WDTCON）

U-0	U-0	U-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0
—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN
bit7							bit0

图注：

R = 可读位	W = 可写位	U = 未实现位，读为0
-n = POR时的值	1 = 置1	0 = 清零
		x = 未知

bit7-5 未实现：读为0



Bit4-1 **WDTPS<3:0>**: 看门狗定时器周期选择位

位值 = 预分频  
 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 = 保留  
 1101 = 保留  
 1110 = 保留  
 1111 = 保留

Bit0 **SWDTEN**: 软件使能或禁止看门狗定时器位<sup>(1)</sup>

1 = WDT 开启  
 0 = WDT 关闭 (复位值)

注 1: 如果配置字寄存器 (CONFIG) 的WDTE配置位=1, 则WDT始终被使能, 而与该控制位的状态无关。如果配置字寄存器 (CONFIG) 的WDTE配置位=0, 则可以使用该控制位开启/关闭WDT。

表 2-6-1: WDT 状态

条件	WDT
CLRWT 命令	清零
振荡器失效检测	
退出休眠 + 系统时钟 = T1OSC、EXTRC、NTRC 或 EXTCLK	
退出休眠 + 系统时钟 = XT、HS 或 LP	清零直到 OST 结束

表 2-6-2: 与看门狗定时器相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
WDTCN	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 0000	---0 0000
TMR	PAPH	IES	TCS	TCE	PSC	PS2	PS1	PS0	1111 1111	1111 1111
CONFIG0	CPDB	CPB	MCLRE	PWRTE	WDTE	FOSC2	FOSC1	FOSC0	—	—

图注: 看门狗定时器不使用阴影单元。

注 1: 关于配置字寄存器中所有位的操作, 请参见[寄存器 1200H](#)。



## 3.0 复位

### 3.1 概述

EN8F683 有以下几种不同类型的复位：

- a) 上电复位 (POR)
- b) 正常工作期间的 WDT 复位
- c) 休眠期间的 WDT 复位
- d) 正常工作期间的 MCLR 复位
- e) 休眠期间的 MCLR 复位
- f) 欠压复位 (BOR)

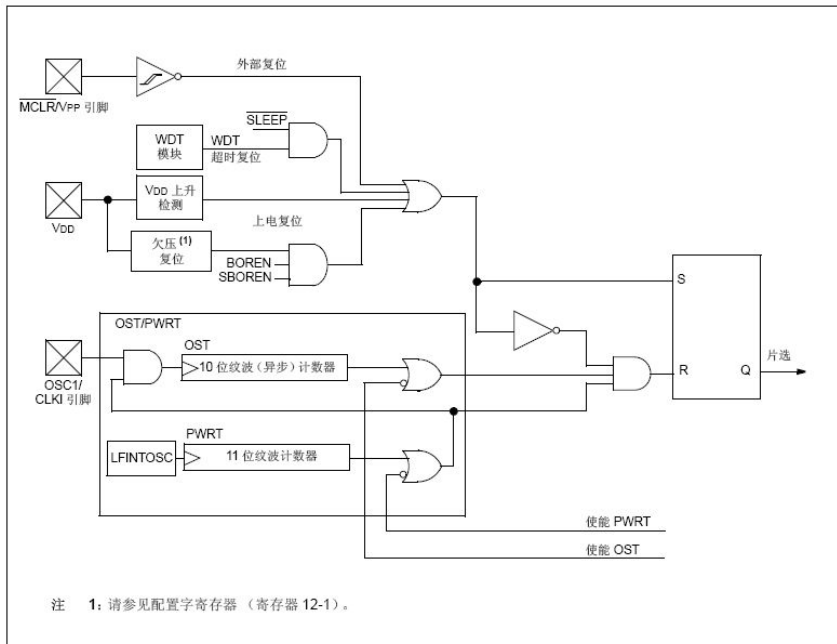
有些寄存器不受任何复位的影响；在上电复位时它们的状态未知，而在其他复位时状态不变。大多数寄存器在以下复位时会复位到各自的“复位状态”：

- 上电复位
- MCLR 复位
- 休眠期间的 MCLR 复位
- WDT 复位

WDT 唤醒不会导致寄存器像 WDT 复位那样复位，这是因为唤醒被视为恢复正常工作。TO 和 PD 位在不同的复位情形下会分别被置 1 或清零，如表 3-1 所示。软件可使用这些位判断复位的性质。所有寄存器详细复位状态如表 3-2 所示。

图 3-1 给出了片上复位电路的简化框图。MCLR 复位路径上有一个噪声滤波器，用来检测并滤除小脉冲。关于脉冲宽度规范，请参见第 13.0 节“电气特性”。

图 3-1：片上复位电路的简化框图



POR	BOR	TF	PF	条件
0	x	1	1	上电复位
u	0	1	1	欠压复位

u	u	0	u	WDT 复位
u	u	0	0	WDT 唤醒
u	u	u	u	正常工作期间的MCLR复位
U	u	1	0	休眠期间的MCLR复位

图注: u = 不变, x = 未知

表 3-2: 寄存器的初始状态

寄存器	地址	上电复位	MCLR 复位 WDT 复位 欠压复位 <sup>(1)</sup>	通过中断将器件从休眠状态唤醒 通过 WDT 延时从休眠状态唤醒
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
IAR	00h/80h	xxxx xxxx	xxxx xxxx	uuuu uuuu
RTCC	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h/82h	0000 0000	0000 0000	PC + 1 <sup>(3)</sup>
STATUS	03h/83h	0001 1xxx	000q quuu <sup>(4)</sup>	uuuq quuu <sup>(4)</sup>
MSR	04h/84h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA <sup>(2)</sup>	05h	--x0 x000	--u0 u000	--uu uuuu
PCHLAT	0Ah/8Ah	---0 0000	---0 0000	---u uuuu
INTS	0Bh/8Bh	0000 0000	0000 0000	uuuu uuuu <sup>(2)</sup>
PIFB1	0Ch	0000 0000	0000 0000	uuuu uuuu <sup>(2)</sup>
TMR1L	0Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	0Fh	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1STA	10h	0000 0000	uuuu uuuu	-uuu uuuu
TMR2	11h	0000 0000	0000 0000	uuuu uuuu
T2CON	12h	-000 0000	-000 0000	-uuu uuuu
CCPR1L	13h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	14h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	15h	0000 0000	0000 0000	uuuu uuuu
WDTCON	18h	---0 1000	---0 1000	---u uuuu
CMSTA	19h	0000 0000	0000 0000	uuuu uuuu
CMCON1	1Ah	---- --10	---- --10	---- --uu
ADRESH	1Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADSO	1Fh	00-0 0000	00-0 0000	uu-u uuuu
TMR	81h	1111 1111	1111 1111	uuuu uuuu
CPIOA	85h	--11 1111	--11 1111	--uu uuuu
PIEB1	8Ch	0000 0000	0000 0000	uuuu uuuu
PSTA	8Eh	--01 100x	--0u 00uq <sup>(1,5)</sup>	--uu uuuu
OSCCON	8Fh	-110 x000	-110 q000	-uuu uuuu
PR2	92h	1111 1111	1111 1111	1111 1111
PAPHR <sup>(3)</sup>	95h	--11 -111	--11 -111	uuuu uuuu
PAINTR	96h	--00 0000	--00 0000	--uu uuuu
VRSTA	99h	0-0- 0000	0-0- 0000	u-u- uuuu
EEDATA	9Ah	0000 0000	0000 0000	uuuu uuuu
EEADR	9Bh	0000 0000	0000 0000	uuuu uuuu
EECON1	9Ch	---- x000	---- q000	---- uuuu
EECON2	9Dh	---- ----	---- ----	---- ----
ADRESL	9Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADS1	9Fh	0000 ----	0000 ----	uuuu ----

图注: u = 不变, x = 未知, - = 未实现位, 读为 0, q = 取值视具体条件而定。

注 1: 如果 VDD 过低, 将激活上电复位, 寄存器将受到不同的影响。

2: INTS 和/或 PIFB1 寄存器中的 1 位或多位会受到影响 (引起唤醒)。

3: 当器件被中断唤醒且 GIE 位置 1 时, PC 装入中断向量 (0004h)。

- 4: 关于特定条件下的复位值，请参见表 3-7-1。
- 5: 如果复位是由于欠压引起的，则 bit0=0。所有其他复位将导致 bit0=u。
- 6: 由 ANINS 寄存器控制的带有模拟功能的端口引脚在复位后立即读为 0，即使数据锁存器未定义（POR）或不变（其他复位时）。

## 3.2 上电复位

在 VDD 达到足以使器件正常工作的电平之前，片上上电复位电路将使器件保持在复位状态。要有效利用 POR，只要将  $\overline{\text{MCLR}}$  引脚通过一个电阻连接到 VDD 即可。需要一个最大上升时间才能达到 VDD。详见第 13.0 节“电气特性”。如果使能了欠压复位，那么该最大上升时间规范将不再适用。欠压复位电路将使器件保持在复位状态，直到 VDD 达到 VBOR（见第 3.6 节“欠压复位（BOR）”）。

注：当 VDD 降低时，上电复位电路不会产生内部复位。要重新使能上电复位，VDD 必须至少保持 100us 的 Vss 电压。

当器件开始正常工作（退出复位状态）时，器件的工作参数（即电压、频率和温度等）必须得到满足，以确保其正常工作。如果不满足这些条件，那么器件必须保持在复位状态，直到满足工作条件为止。

## 3.3 MCLR

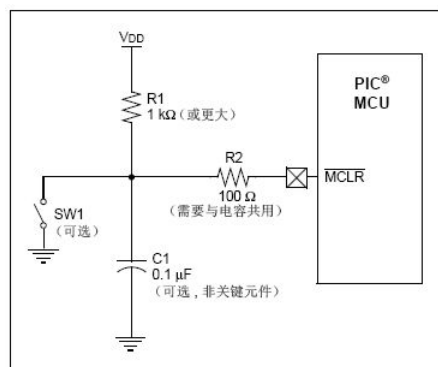
EN8F683 在  $\overline{\text{MCLR}}$  复位路径中有一个噪声滤波器。该滤波器检测并滤除小脉冲。

应注意，WDT 复位不会将  $\overline{\text{MCLR}}$  引脚驱动为低电平。

若施加在  $\overline{\text{MCLR}}$  引脚上的电压超出规范值，则在 ESD 事件发生期间可导致  $\overline{\text{MCLR}}$  复位且器件中流过超过规范值的过电流。因此，建议不要把  $\overline{\text{MCLR}}$  引脚直接连接到 VDD。建议使用图 3-3 给出的 RC 网络。

通过清零配置字寄存器中的  $\overline{\text{MCLR}}$  位，可启用内部  $\overline{\text{MCLR}}$  选项。当  $\text{MCLRE} = 0$  时，在内部产生芯片的复位信号。当  $\text{MCLRE} = 1$  时，PA3/ $\overline{\text{MCLR}}$  引脚变成外部复位输入。在这种模式下，PA3/ $\overline{\text{MCLR}}$  引脚具有到 VDD 的弱上拉功能。

图 3-3: 建议的  $\overline{\text{MCLR}}$  电路



### 3.4 看门狗复位

### 3.5 上电延时定时器（PWRT）

上电延时定时器仅在上电时（上电复位或欠压复位）提供一个 **55ms**（标称值）的固定延时。上电延时定

时器采用 LFINTOSC 振荡器作为时钟源，工作频率为 31kHz。更多信息，请参见第 4.5 节“内部时钟模式”。

只要 PWRT 处于活动状态，芯片就保持在复位状态。配 PWRT 延时使 VDD 有足够的时间上升到所需的电平。

置位  $\overline{\text{PWRT}}E$  可以禁止（如果置 1）或使能（如果清零或被编程）上电延时定时器。虽然不是必需的，但是在使能欠压复位时也应使能上电延时定时器。

由于以下原因不同芯片的上电延时定时器的延时也各不相同：

- VDD 差异
- 温度差异
- 制造工艺差异

详见直流参数（第 13.0 节“电气特性”）。

注：在  $\overline{\text{MCLR}}$  引脚的低于 VSS 的电压尖峰，包括大于 80mA 的电流，可导致闭锁。因此，在  $\overline{\text{MCLR}}$  引脚上施加“低”电平时，应使用阻值在 50-100Ω 的串联电阻，而不是将该引脚直接拉到 VSS。

## 3.6 欠压复位

配置字寄存器中的 BOREN0 和 BOREN1 位用于选择 4 种欠压复位模式中的一种。其中添加了两种允许使用软件或硬件对 BOR 的使能进行控制的模式。当  $\text{BOREN}\langle 1:0 \rangle = 01$  时，可由 PSTA 寄存器的 SBOREN 位使能/禁止 BOR，从而能用软件对其进行控制。通过选择  $\text{BOREN}\langle 1:0 \rangle$ ，可使欠压复位在休眠时被自动禁止，从而节约功耗；而在唤醒后被重新使能。在此模式下，SBOREN 位被禁止。关于配置字的定义，请参见寄存器 PSTA。

如果 VDD 下降到 VBOR 以下，且持续时间超过参数值（TBOR）（见第 13.0 节“电气特性”，欠压状况将使）器件复位。不管 VDD 的变化速率如何，上述情况都会发生。如果 VDD 低于 VBOR 的时间少于参数值（TBOR），则不一定会发生复位。

任何复位（上电复位、欠压复位或看门狗定时器复位等）都会使芯片保持复位状态，直到 VDD 上升到 VBOR 以上（见图 3-6）。如果使能了上电延时定时器，此时它将启动，并且会使器件保持复位状态的时间延长 64ms。

注：配置字寄存器中的  $\overline{\text{PWRT}}E$  位用于使能上电延时定时器。

如果在上电延时定时器运行过程中，VDD 降低到 VBOR 以下，芯片将重新回到欠压复位状态并且上电延时定时器会恢复为初始状态。一旦 VDD 上升到 VBOR 以上，上电延时定时器将执行一段 64ms 的复位。

图 3-6: 欠压情形

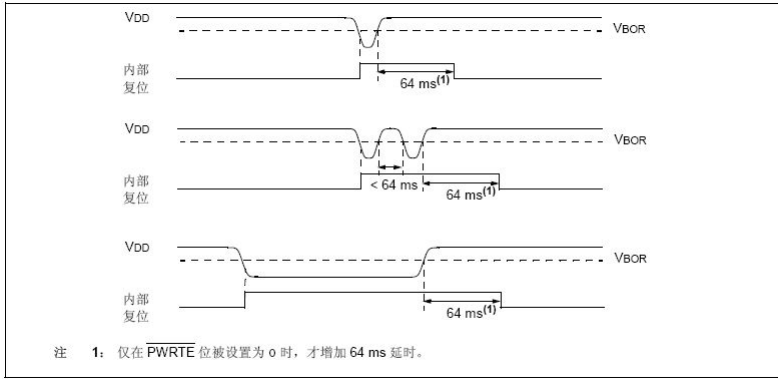


表 3-6: 与欠压有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
PSTA	—	—	ULPWUE	SBOREN	—	—	POR	BOR	—0 0000	—0 0000
STATUS	IRP	PR1	PAGE	$\overline{\text{T}}\text{F}$	$\overline{\text{P}}\text{F}$	Z	HC	C	1111 1111	1111 1111

图注: u = 不变, x = 未知, — = 未实现位, 读为 0, q = 取值视具体情况而定。BOR 不使用阴影单元。

注 1: 其他 (非上电) 复位包括正常工作时的 MCLR 复位和看门狗定时器复位。

### 3.7 延时时序

上电时的延时时序如下:

- 在 POR 延时结束后, 施加一段 PWRT 延时。
- PWRT 超时后激活 OST。

总延时时间取决于振荡器配置和  $\overline{\text{PWRT}}\text{E}$  位的状态。例如, 在 EC 模式且  $\overline{\text{PWRT}}\text{E}$  位被擦除 (PWRT 禁止) 的情况下, 根本不会出现延时。图 3-7-1、3-7-2 和 3-7-3 分别给出了各种情形下的延时时序。当振荡器起振后, 通过使能双速启动或故障保护监控器, 器件将以 INTOSC 作为时钟源来执行代码 (见第 4.7.2 节“双速启动顺序”和第 4.9 节“故障保护时钟监控器”)。

由于延时是由上电复位脉冲触发的, 因此如果 MCLR 保持足够长时间的低电平, 所有延时都将结束。将 MCLR 电平拉高后, 器件将立即开始执行代码 (见图 3-7-2)。这对于测试或同步多个并行工作的 EN8F683 器件来说是非常有用的。

表 3-7-1 给出了一些特殊寄存器的复位条件, 而表 3-2 给出了所有寄存器的复位条件。

图 3-7-1: 上电时的超时时序 (MCLR 延时): 情形 1

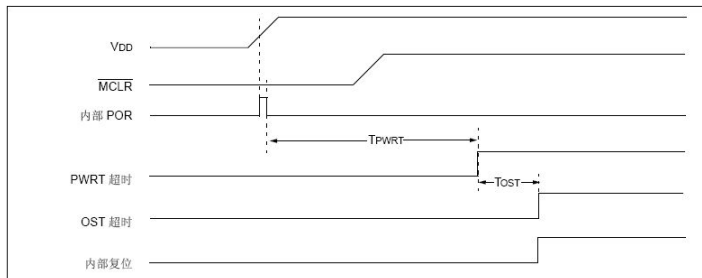


图 3-7-2: 上电时的超时时序 (MCLR 延时): 情形 2

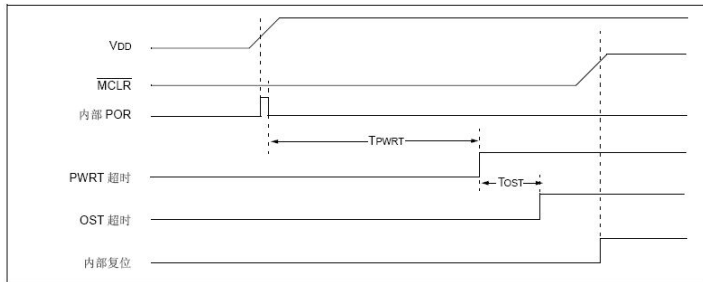


图 3-7-3: 上电时的超时时序 (MCLR 及 VDD)

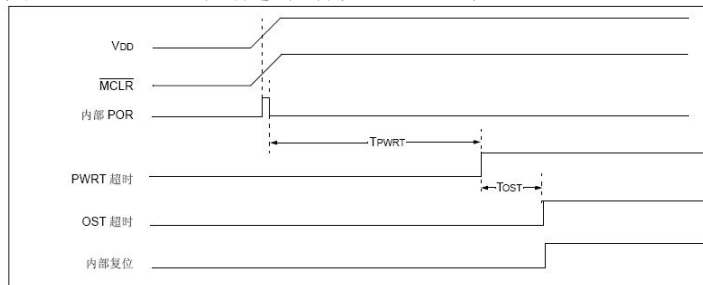


表 3-7-1: 特殊寄存器的初始状态

条件	程序计数器	状态寄存器	PSTA 寄存器
上电复位	0000h	0001 1xxx	--01 --0x
正常工作期间的MCLR复位	0000h	000u uuuu	--0u --uu
休眠期间的MCLR复位	0000h	0001 0uuu	--0u --uu
WDT 复位	0000h	0000 uuuu	--0u --uu
WDT 唤醒	PC + 1	uuu0 0uuu	--uu --uu
欠压复位	0000h	0001 1uuu	--01 --u0
通过中断从休眠唤醒	PC + 1 <sup>(1)</sup>	uuu1 0uuu	--uu --uu

图注: u = 不变, x = 未知, -- = 未实现位, 读为 0。

注 1: 当器件被中断唤醒且全局中断允许位 GIE 位置 1 时, 执行 PC+1 后, PC 装入中断向是 (0004h)。

表 3-7-2: 各种情形下的超时

振荡器配置	上电		欠压复位		从休眠状态唤醒
	PWRTE = 0	PWRTE = 1	PWRTE = 0	PWRTE = 1	
XT, HS, LP	$T_{PWRT} + 1024 * T_{OSC}$	$1024 * T_{OSC}$	$T_{PWRT} + 1024 * T_{OSC}$	$1024 * T_{OSC}$	$1024 * T_{OSC}$
RC, EC, INTOSC	$T_{PWRT}$	—	$T_{PWRT}$	—	—

### 3.8 电源控制 (PSTA) 寄存器

电源控制寄存器 PSTA ([寄存器 8EH](#)) 有两个状态位, 用于指示上次发生的复位的类型。

bit0 是 BOR (欠压复位) 标志位。BOR 在上电复位时未知。然后, 用户必须将该位置 1, 并在随后的复位发如果是, 则表示已发生欠压生时检查中 BOR 是否为 0, 复位。当禁止欠压复位电路 (配置字寄存器中的 BOREN<1:0> = 00) 时, BOR 状态位是“无关位”并且不一定预测得到。

bit1 是 POR (上电复位) 标志位, 在上电复位时值为 0, 其他情况下不受影响。上电复位后, 用户必须对该位写 1。发生后续复位后, 如果 POR 为 0, 则表示发生了上电复位 (即 VDD 可能已经变为了低电平)



## 4.0 系统时钟

### 4.1 概述

振荡器有多种时钟源和选择功能，从而使其应用非常广泛，并可最大限度地提高性能和降低功耗。图 4-1 给出了振荡器模块的框图。

时钟源可以配置为由外部振荡器、石英晶体谐振器、陶瓷谐振器以及阻容（RC）电路提供。此外，系统时钟源可以配置为由两个内部振荡器中的一个提供，并可以通过软件选择速度。其他时钟功能包括：

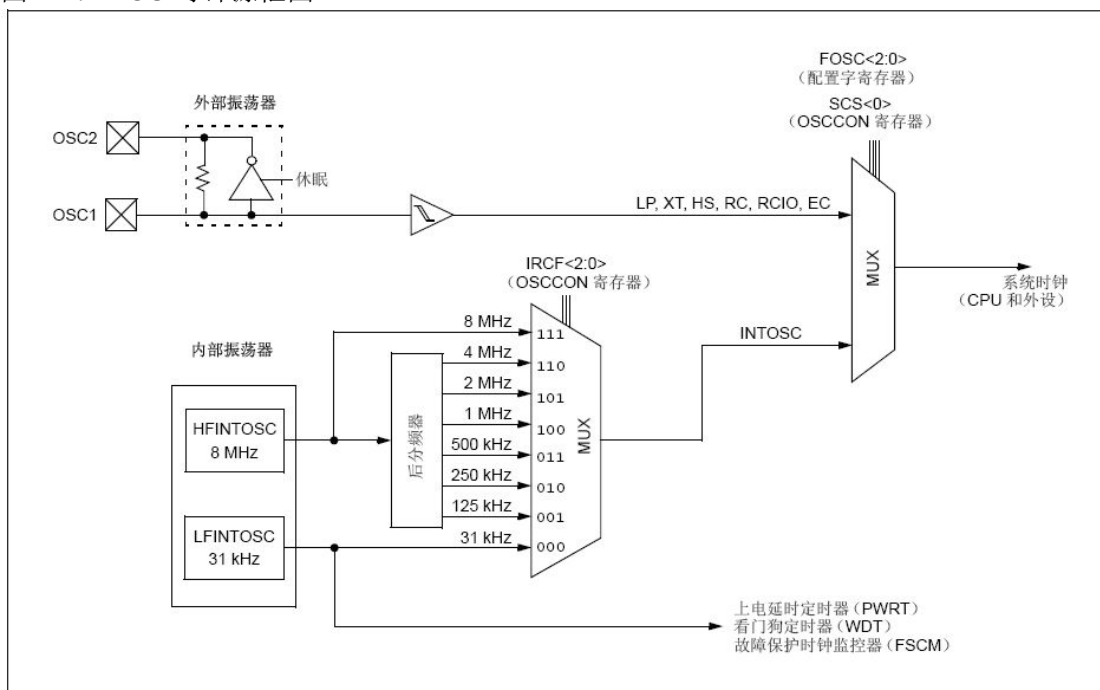
- 通过软件选择外部或内部系统时钟源。
- 双速启动模式，使外部振荡器从启动到代码执行间的延时达到最小。
- 故障保护时钟监视器（FSCM）旨在检测外部时钟源的故障（LP、XT、HS、EC 或 RC 模式）并自动切换到内部振荡器。

振荡器模块可配置为以下 8 种时钟模式之一。

1. EC——外部时钟，I/O 在 OSC2/CLKOUT 上。
2. LP——32kHz 低功耗晶振模式。
3. XT——中等增益晶振或陶瓷谐振振荡器模式。
4. HS——高增益晶振或陶瓷谐振器模式。
5. RC——外部阻容（RC），FOSC/4 输出到 OSC2/CLKOUT。
6. RCIO——外部阻容，I/O 在 OSC2/CLKOUT 上。
7. INTOSC——内部振荡器，FOSC/4 输出到 OSC2 且 I/O 在 OSC1/CLKIN 上。
8. INTOSCIO——内部振荡器，I/O 在 OSC2/CLKOUT 和 OSC1/CLKIN 上。

通过配置字寄存器（CONFIG）的 FOSC<2:0> 位来配置时钟源模式。内部时钟可用两个内部振荡器产生。HFINTOSC 是经过校准的高频振荡器。LFINTOSC 是未经校准的低频振荡器。

图 4-1: MCU 时钟源框图



## 4.2 振荡器控制

振荡器控制（OSCCON）寄存器（图 4-1）控制系统时钟和频率选择等选项。OSCCON 寄存器包含以下位：

- 频率选择位（IRCF）
- 频率状态位（HTS 和 LTS）
- 系统时钟控制位（OSTS 和 SCS）

寄存器 8FH: 振荡器控制寄存器（OSCCON）

U-0	R/W-1	R/W-1	R/W-0	R-1	R-0	R-0	R/W-0
—	<b>IRCF2</b>	<b>IRCF1</b>	<b>IRCF0</b>	<b>OSTS<sup>(1)</sup></b>	<b>HTS</b>	<b>LTS</b>	<b>SCS</b>
bit7							bit0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7 未实现：读为0

bit6-4 **IRCF<2:0>**：内部振荡器频率选择位

000 = 31kHz

001 = 125kHz

010 = 250kHz

011 = 500kHz

100 = 1MHz

101 = 2MHz

110 = 4MHz（缺省值）

111 = 8MHz

bit3 **OSTS**：振荡器起振超时状态位<sup>(1)</sup>

1 = 器件运行在FOSC<2:0>定义的外部时钟之下

0 = 器件运行在内部振荡器之下（HFINTOSC或LFINTOSC）

bit2 **HTS**：HFINTOSC（高频——8MHz到125kHz）状态位

1 = HFINTOSC稳定

0 = HFINTOSC不稳定

bit1 **LTS**：LFINTOSC（低频——31kHz）状态位

1 = LFINTOSC稳定

0 = LFINTOSC不稳定

bit0 **SCS**：系统时钟选择位

1 = 内部振荡器用于系统时钟

0 = 时钟源由FOSC<2:0>决定

注 1：双速启动且选取LP、XT或HS为振荡器模式时，或者故障保护模式使能时，该位将复位为0。

## 4.3 时钟源模式

时钟源模式可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。例子有：振荡器模块（EC 模式）、石英晶体谐振器或陶瓷谐振器

振器（LP、XT 和 HS 模式）以及阻容（RC）模式电路。

- 内部时钟源内置于振荡器模块中。振荡器模块有两个内部振荡器，一个是 8MHz 高频内部振荡器（HFINTOSC），另一个是 31kHz 低频内部振荡器（LFINTOSC）。

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。（欲了解更多信息，请参见[第 4.6 节“时钟切换”](#)）。

## 4.4 外部时钟模式

### 4.4.1 振荡器起振定时器（OST）

如果振荡器模块配置为 LP、XT 或 HS 模式，振荡器起振定时器（OST）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（如果配置了）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。[表 4-4-1](#) 给出了振荡器延时的例子。

为了使外部振荡器起振和代码执行之间的延时最小，可选择双速时钟启动模式（见[第 4.7 节“双速时钟启动模式”](#)）。

表 4-4-1: 振荡器延时示例

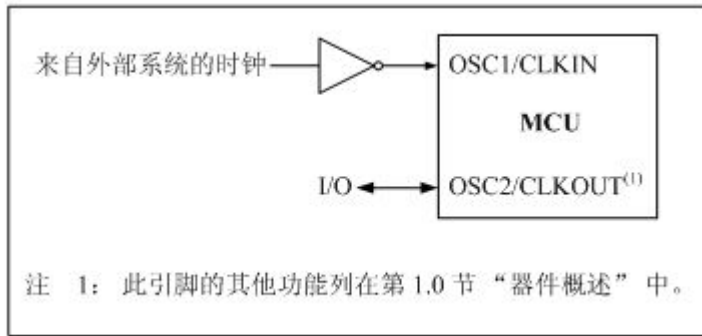
切换自	切换到	频率	振荡器延时
休眠/POR	LFINTOSC HFINTOSC	31kHz 125kHz到8MHz	振荡器预热延时( $T_{WARM}$ )
休眠/POR	EC, RC	DC—20MHz	双周期
LFINTOSC (31kHz)	EC, RC	DC—20MHz	每次一周期
休眠/POR	LP, XT, HS	32kHz到20MHz	1024个时钟周期 (OST)
LFINTOSC (31kHz)	HFINTOSC	125kHz到8MHz	1us (近似值)

### 4.4.2 EC 模式

外部时钟（EC）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。[图 4-4-2](#) 给出了 EC 模式的引脚连接。

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。因为 YSPRING MCU 的设计是完全静态的，停止外部时钟输入将使器件暂停工作并保持所有数据完整。当再次启动外部时钟时，器件恢复工作，就好像没有停止过一样。

图 4-4-2: 外部时钟（EC）模式的工作原理



### 4.4.3 LP、XT 和 HS 模式

LP、XT 和 HS 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（图 4-4-3-1）。模式选择内部反相放大器的低、中或高增益设定，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 模式的电流消耗在三种模式中最小。该模式设计仅用于驱动 32.768kHz 音叉（Tuning Fork）式晶振（钟表晶振）。

XT 振荡器模式选择内部反相放大器的中等增益设定。XT 模式的电流消耗在三种模式中居中。该模式最适用于驱动具备中等驱动电平规格要求的谐振器。

HS 振荡器模式选择内部反相放大器的最高增益设定。HS 模式的电流消耗在三种模式中最大。该模式最适用于驱动需要高驱动设定的谐振器。

图 4-4-3-1 和图 4-4-3-2 分别给出了石英晶体谐振器和陶瓷谐振器的典型电路。

- 注 1: 石英晶振的特性随类型、封装和制造商而变化。要了解规格说明和推荐应用，应查阅制造商提供的数据手册。
- 注 2: 应始终验证振荡器在应用预期的VDD和温度范围内的性能。

图 4-4-3-1: 石英晶体的工作原理（LP、XT 或 HS 模式）

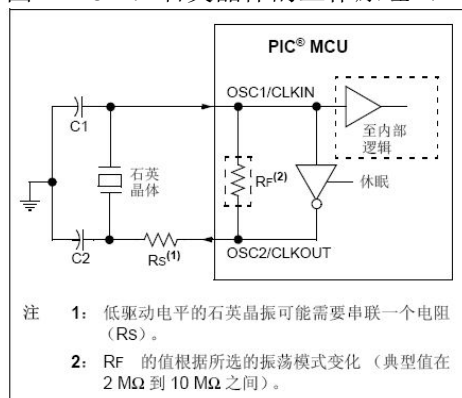
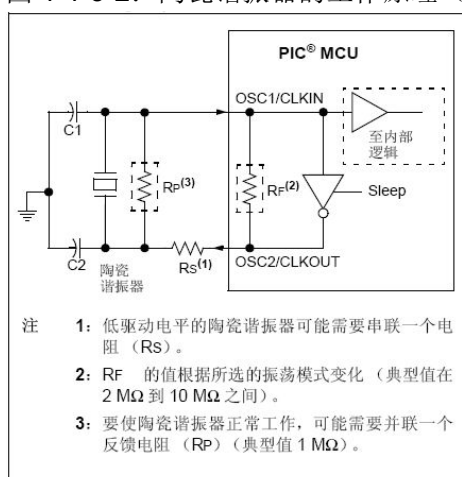


图 4-4-3-2: 陶瓷谐振器的工作原理（XT 或 HS 模式）

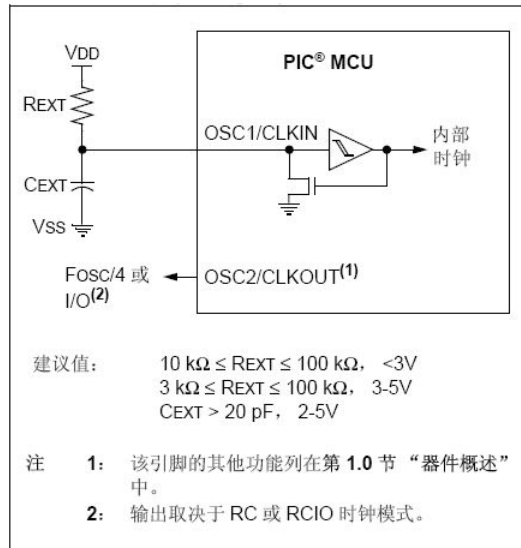


#### 4.4.4 外部 RC 模式

外部阻容 (RC) 模式支持使用外部 RC 电路。对时钟精度要求不高时, 这使设计人员有了很大的频率选择空间, 且保持成本最低。有 RC 和 RCIO 两种模式。

在 RC 模式下, 电路连接到 OSC1。RCOSC2/CLKOUT 输出 RC 振荡频率的 4 分频。该信号可用于为外部电路、同步、校准、测试或其他应用需求提供时钟。[图 4-4-4](#) 给出了外部 RC 模式的连接图。

图 4-4-4: 外部 RC 模式



在 RCIO 模式下, RC 电路连接到 OSC1。OSC2 成为额外的通用 I/O 引脚。I/O 引脚成为 PORTA 的 bit4 (PA4)。[图 4-5-5](#) 给出了 RCIO 模式的连接图。

RC 振荡器频率是供电电压、电阻 ( $R_{EXT}$ ) 和电容 ( $C_{EXT}$ ) 值以及工作温度的函数。影响振荡器频率的其他因素有:

- 电压门限值变化
- 元件容差
- 不同封装的电容

用户还应考虑因所使用的外部 RC 元件的容差而导致的差异。

#### 4.5 内部时钟模式

振荡器模块有两个独立的内部振荡器, 可配置或选取为系统时钟源。

1. HFINTOSC (高频内部振荡器) 出厂时已校准, 工作频率为 8MHz, 精度范围为  $\pm 1\%$ 。
2. LFINTOSC (低频内部振荡器) 未经校准, 工作频率为 31kHz。

通过软件对 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 (见 [第 4.6 节“时钟切换”](#))。

### 4.5.1 INTOSC 和 INTOSCIO 模式

当在配置字寄存器（CONFIG）中使用振荡器选择位 FOSC<2:0>设置器件时，在 INTOSC 和 INTOSCIO 模式下将内部振荡器配置为系统时钟源。

在 INTOSC 模式下，OSC1/CLKIN 可用作通用 I/O。OSC2/CLKOUT 输出所选内部振荡器频率的 4 分频。CLKOUT 信号可用来为外部电路、同步、校准、测试或其他应用需求提供时钟。

在 INTOSCIO 模式下，OSC1/CLKIN 和 OSC2/CLKOUT 引脚可用作通用 I/O。

### 4.5.2 HFINTOSC

高频内部振荡器（HFINTOSC）是出厂时已校准的 8MHz 内部时钟源。

HFINTOSC 的输出连接到后分频器和多路复用器（见图 4-1）。使用 OSCCON 寄存器的 IRCF<2:0>位，可通过软件选择七个频率之一。更多信息，请参见第 4.5.4 节“频率选择位（IRCF）”。

将 OSCCON 寄存器的 IRCF<2:0>位设置为  $\neq 000$  选择 8MHz 到 125kHz 之间的任一频率，可使能 HFINTOSC。然后将 OSCCON 寄存器的系统时钟源（SCS）位置 1，或通过将配置寄存器（CONFIG）中的 IESO 置 1 使能双速启动。

OSCCON 寄存器的 HF 内部振荡器（HTS）位用于显示 HFINTOSC 是否稳定。

### 4.5.3 LFINTOSC

低频内部振荡器（LFINTOSC）是未经校准的 31kHz 内部时钟源。

LFINTOSC 的输出连接到后分频器和多路复用器（见图 4-1）。通过软件对 OSCCON 寄存器的 IRCF<2:0>位进行操作，选取 31kHz。更多信息，请参见第 4.5.4 节“频率选择位（IRCF）”。LFINTOSC 还是上电延时定时器（PWRT）、看门狗定时器（WDT）以及故障保护时钟监控器（FSCM）的时钟源。

选取 31kHz（将 OSCCON 寄存器的 IRCF<2:0>位设置为 000）为系统时钟源（OSCCON 寄存器的位 SCS = 1），或者使能以下任一项时，LFINTOSC 将被使能：

- 双速启动（配置字寄存器的位 IESO = 1 且 OSCCON 寄存器的位 IRCF = 000）
- 上电延时定时器（PWRT）
- 看门狗定时器（WDT）
- 故障保护时钟监控器（FSCM）

OSCCON 寄存器的 LF 内部振荡器（LTS）位用于指示 LFINTOSC 是否稳定。



#### 4.5.4 频率选择位（IRCF）

8MHz HFINTOSC 和 31kHz LFINTOSC 的输出连接到后分频器和多路复用器（见[图 4-1](#)）。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：

- 8MHz
- 4MHz（复位后的缺省值）
- 2MHz
- 1MHz
- 500kHz
- 250kHz
- 125kHz
- 31kHz

注：任何复位后，OSCCON寄存器的IRCF<2:0>位将被置为110且频率选择置为4MHz。用户可修改IRCF位来选择其他频率。

#### 4.5.5 HFINTOSC 和 LFINTOSC 时钟切换时序

当在 LFINTOSC 和 HFINTOSC 之间切换时，新的振荡器可能为了省电已经关闭（见[图 4-5-5](#)）。在这种情况下，OSCCON 寄存器的 IRCF 位被修改之后、频率选择生效之前，存在一个延时。OSCCON 寄存器的 LTS 和 HTS 位将反映 LFINTOSC 和 HFINTOSC 振荡器的当前活动状态。频率选择时序如下：

1. OSCCON 寄存器的 IRCF<2:0>位被修改。
2. 如果新时钟是关闭的，开始一个时钟启动延时。
3. 时钟切换电路等待当前时钟下降沿的到来。
4. CLKOUT 保持为低，时钟切换电路等待新时钟上升沿的到来。
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。

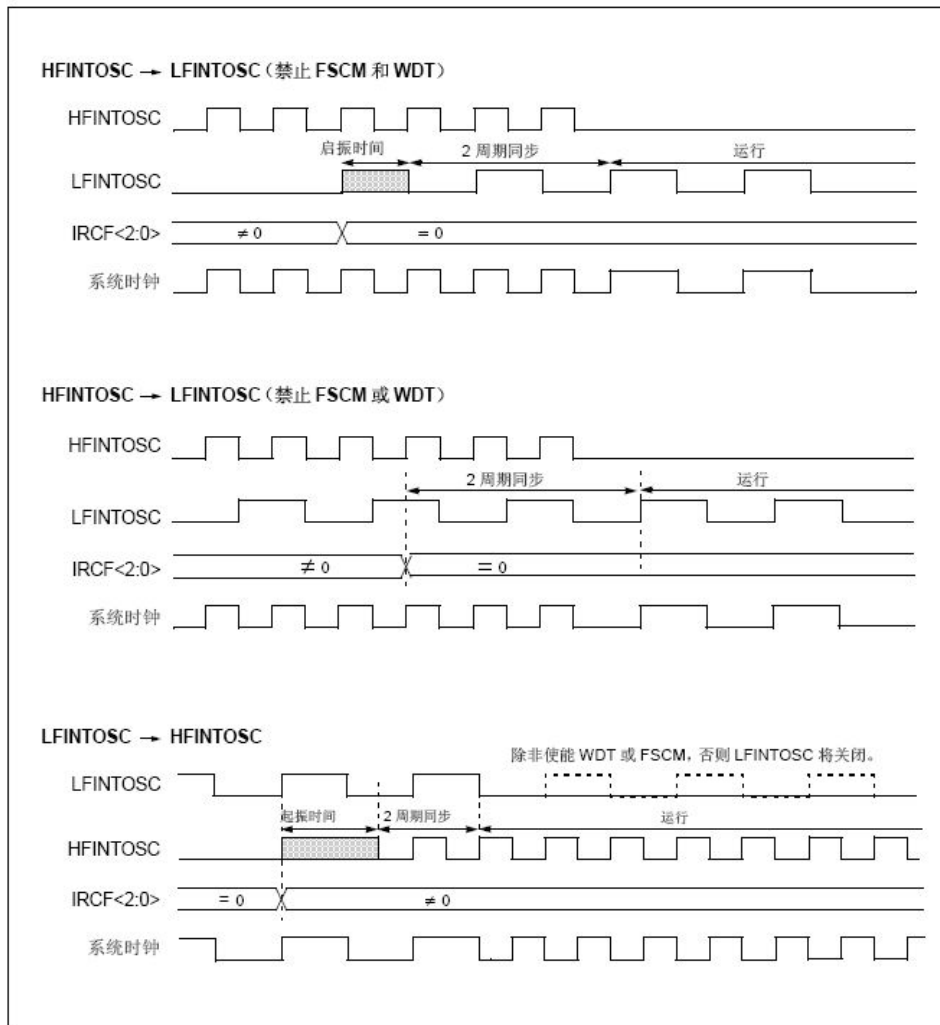
更多信息请参见[图 4-1](#)。

如果选取的内部振荡器速度在 8MHz 到 125kHz 之间，选取新频率不存在启动延时。这是因为新旧频率都来自经过后分频器和多路复用器的 HFINTOSC。

启动延时规范在[第 13.0 节“电气特性”](#)中与振荡器相关的表格中。



图 4-5-5: 内部振荡器切换时序



## 4.6 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。

### 4.6.1 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 (CONFIG) 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: 任何自动时钟切换 (可能产生自双速启动或故障保护时钟监控器) 都不更新 OSCCON 寄存器的 SCS 位。用户可监控 OSCCON 寄存器的 OSTST 位以确定当前的系统时钟源。

## 4.6.2 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（CONFIG）的 FOSC<2:0>定义。OSTS 还特别指明在 LP、XT 或 HS 模式下，振荡器起振定时器（OST）是否已超时。

## 4.7 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，INTOSC 用作时将钟源执行数条指令，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、XT 或 HS 模式时，振荡器起振定时器（OST）使能（见第 4.4.1 节“[振荡器起振定时器（OST）](#)”）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。

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

### 4.7.2 双速启动顺序

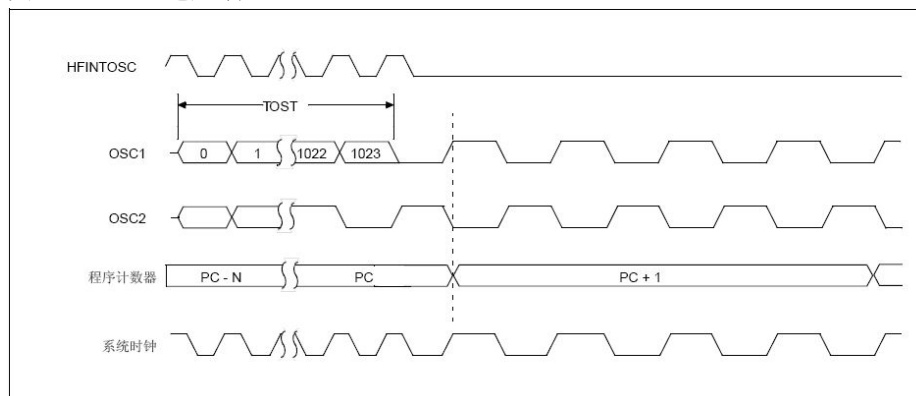
1. 从上电复位或休眠中唤醒。
2. 使用内部振荡器以 OSCCON 寄存器的 IRCF<2:0>位设置的频率开始执行指令。
3. OST 使能，计数 1024 个时钟周期。
4. OST 超时，等待内部振荡器下降沿的到来。
5. OSTS 置 1。

6. 系统时钟保持为低，直到新时钟下一个下降沿的到来（LP、XT 或 HS 模式）。
7. 系统时钟切换到外部时钟源。

### 4.7.3 检查双速时钟状态

通过检查 OSCCON 寄存器的 OSTS 位的状态，可以确定单片机是否如配置字寄存器（CONFIG）中 FOSC<2:0>位定义的那样运行于外部时钟源，抑或是运行于内部振荡器。

图 4-7-3: 双速启动



## 4.8 掉电模式（休眠）

通过执行 SLEEP 指令可进入掉电模式。

如果使能看门狗定时器：

- WDT 将被清零并保持运行。
- 状态寄存器中的 PD 位被清零。
- TO 位被置 1。
- 关闭振荡器驱动器。
- I/O 端口保持执行 SLEEP 指令之前的状态（驱动为高电平、低电平或高阻状态）。

为使这种模式下的电流消耗降至最低，所有 I/O 引脚都应保持为 VDD 或 VSS，以确保没有外部电路从 I/O 引脚消耗电流，同时应禁止比较器和 CVREF。为了避免输入引脚悬空而引入开关电流，应在外部将高阻输入的 I/O 引脚拉为高电平或低电平。为使电流消耗降至最低，T0CKI 输入也应保持为 VDD 或 VSS。还应考虑 PORTA 片上上拉的影响。

$\overline{\text{MCLR}}$  引脚必须为逻辑高电平。

注： 请注意，由于 WDT 超时产生的复位不会将  $\overline{\text{MCLR}}$  引脚驱动为低电平。

### 4.8.1 MCLR 引脚从休眠状态唤醒

可以通过以下任一事件将器件从休眠状态唤醒：

- $\overline{\text{MCLR}}$  引脚上的外部复位输入。

- 看门狗定时器唤醒（如果 WDT 使能）。
- PA2/INT 引脚中断、PORTA 电平变化中断或外设中断。

第一种事件会导致器件复位。后两种事件被认为是程序执行的继续。状态寄存器中的 TO 和 PD 位用于确定器而在执行 SLEEP 件复位的原因。位在上电时被置 1，PD 指令时被清零。TO 位在发生 WDT 唤醒时被清零。

下列外设中断可以将器件从休眠状态唤醒：

1. TMR1 中断。Timer1 必须用作异步计数器。
2. ECCP 捕捉模式中断。
3. A/D 转换（当 A/D 时钟源为  $F_{SM}$  时）。
4. EEPROM 写操作完成。
5. 比较器输出状态变化。
6. 电平变化中断。
7. 来自 INT 引脚的外部中断。

由于在休眠期间没有片上时钟处于工作状态，因此其他外设不能产生中断。

当执行 SLEEP 指令时，下一条指令（PC+1）将预先取出。如果希望通过中断事件唤醒器件，则必须将相应的中断允许位置 1（使能）。发生唤醒与 GIE 位的状态无关。如果 GIE 位被清零（禁止），器件将继续执行 SLEEP 指令之后的指令。如果 GIE 位被置 1（使能）器，件执行 SLEEP 指令之后的指令，然后跳转到中断地址（0004h）处执行代码。如果不希望执行 SLEEP 指令之后的指令，用户应该在 SLEEP 指令后面放置一条 NOP 指令。

注：如果禁止了全局中断（GIE被清零），但有任一中断源将其中断允许位以及相应的中断标志位置1，器件将立即从休眠状态唤醒。

器件从休眠状态唤醒时，WDT 都将被清零，而与唤醒原因无关。

## 4.8.2 使用中断唤醒

当禁止全局中断（GIE 被清零）时，并且有任一中断源将其中断允许位和中断标志位置 1，将会发生下列事件之一：

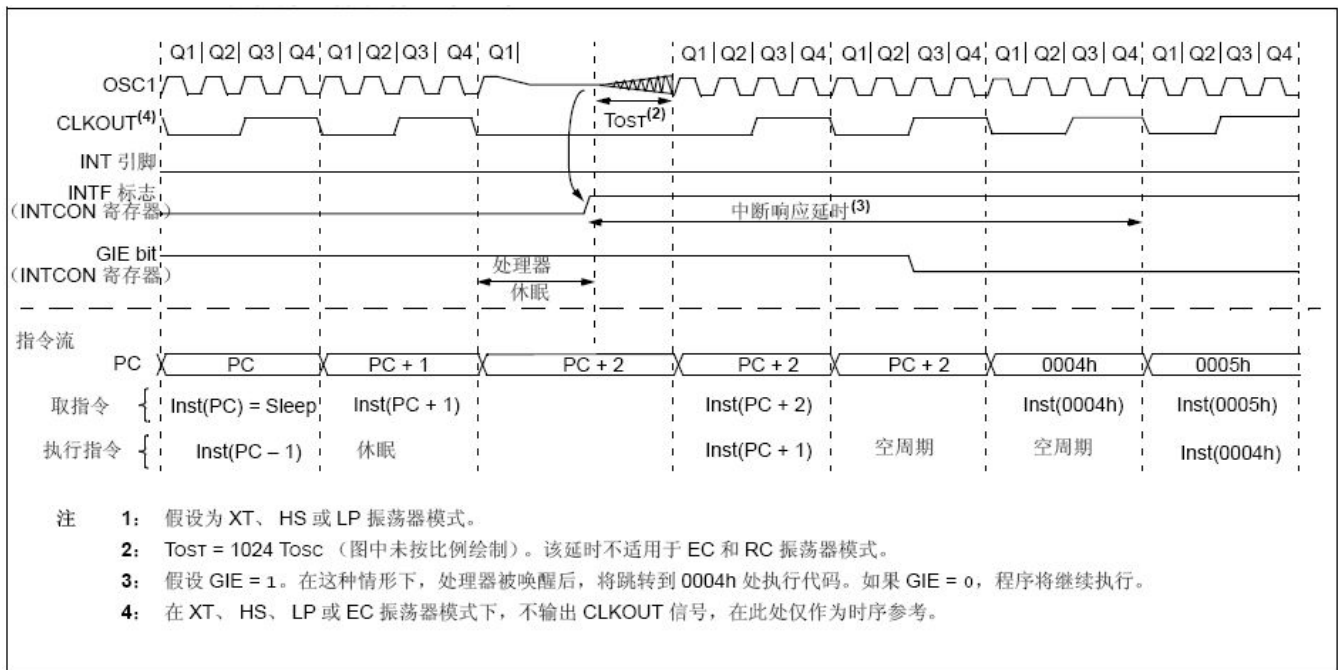
- 如果在执行 SLEEP 之前产生了中断，那么 SLEEP 指令将被作为一条 NOP 指令执行。因此，WDT 及其预分频器和后分频器（如果使能）将不会被清零，并且 TO 位将不会被置 1，同时 PD 位也不会被清零。

- 如果在执行 SLEEP 指令期间或之后产生了中断，那么器件将被立即从休眠状态唤醒。SLEEP 指令将在唤醒之前执行完毕。因此，WDT 及其预分频器和后分频器（如果使能）将被清零，并且 TO 位将被置 1，同时 PD 位也将被清零。

即使在执行 SLEEP 指令之前，检查到标志位为 0，它也可能在 SLEEP 指令执行完毕之前被置 1。要确定是否执行了 SLEEP 指令，可测试 PD 位。如果 PD 位置 1，则说明 SLEEP 指令被当作一条 NOP 指令执行了。

在执行 SLEEP 指令之前，必须先执行一条 CLRWT 指令，来确保将 WDT 清零。详情请参见图 4-8。

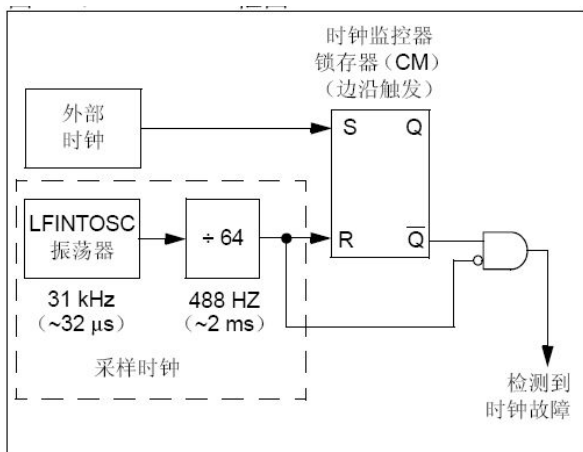
图 4-8：通过中断将器件从休眠状态唤醒



## 4.9 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 (CONFIG) 中的 FCMEN 位置 1 来使能。FSCM 可用于所有外部振荡模式 (LP、XT、HS、EC、RC 和 RCIO)。

图 4-9：FSCM 框图



### 4.9.1 故障保护检测

FSCM 模块通过将外部振荡器与 FSCM 采样时钟比较来检测振荡器故障。LFINTOSC 除以 64，就产生了采样时钟。请参见图 4-9。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。

## 4.9.2 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 PIR2 寄存器的 OSFIF 标志位置 1。如果在 PIR2 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。

## 4.9.3 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。

## 4.9.4 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 或 RC 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。FSCM 被使能时，当双速启动也被使能。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（即，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。



## 5.0 中断

EN8F683 有以下多种中断源：

- 外部中断 PA2/INT
- Timer0 溢出中断
- PORTA 电平变化中断
- 两个比较器中断
- A/D 中断
- Timer1 溢出中断
- Timer2 匹配中断
- EEPROM 数据写中断
- 故障保护时钟监控器中断
- 增强型 CCP 中断

中断控制寄存器 (INTS) 和外设中断请求寄存器 1 (PIFB1) 在各自的标志位中记录各种中断请求。INTS 寄存器还包括各个中断允许位和全局中断允许位。

INTS 寄存器中的全局中断允许位 GIE 在置 1 时允许所有未屏蔽的中断，而在清零时禁止所有中断。可以通过 INTS 和 PIEB1 寄存器中相应的允许位来禁止各个中断。复位时 GIE 被清零。

响应中断时，自动发生以下动作：

- GIE 被清零以禁止任何其他中断。
- 返回地址被压入堆栈。
- 在 PC 中装入 0004h。

执行“从中断返回”指令 RTFI 退出中断程序并将 GIE 位置 1，从而重新使能未屏蔽的中断。

INTS 寄存器包含以下中断标志位：

- INT 引脚中断
- PORTA 电平变化中断
- Timer0 溢出中断

外设中断标志位在 PIFB1 寄存器中。相应的中断允许位在 PIEB1 寄存器中。

PIFB1 寄存器包含以下中断标志位：

- EEPROM 数据写中断
- A/D 中断
- 2 个比较器中断
- Timer1 溢出中断
- Timer2 匹配中断
- 故障保护时钟监视器中断
- 增强型 CCP 中断

对于外部中断事件，如 INT 引脚或 PORTA 电平变化中断，中断响应延时将为 3 到 4 个指令周期。确切的延时时间取决于发生中断事件的时间（见[图 5-1-1](#)）。对于单周期或双周期指令，中断响应延时完全相同。进入中断服务程序之后，就可以通过查询中断标志位来确定中断源。在重新允许中断前，必须用软件将中断标志位清零，以避免重复响应该中断。

注： 1：各中断标志位的置 1 不受相应的中断屏蔽位或 GIE 位状态的影响。  
2：当执行一条清零 GIE 位的指令后，任何等待在下一周期执行的中断都将被忽略。当 GIE 位被再次置 1 后，被忽略的中断仍会继续等待处理。

关于 Timer1、Timer2、比较器、A/D、数据 EEPROM、EUSART、SSP 或增强型 CCP 模块的更多信息，请参见相应的外设章节。

图 5-1-1: 中断逻辑

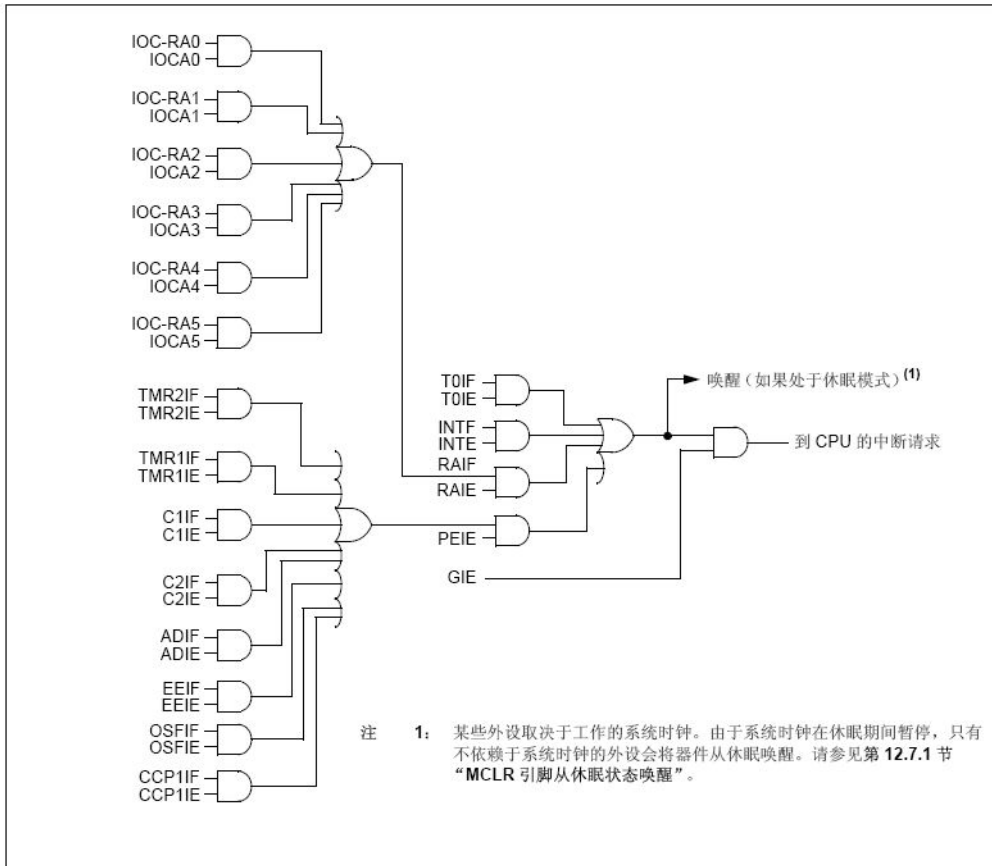
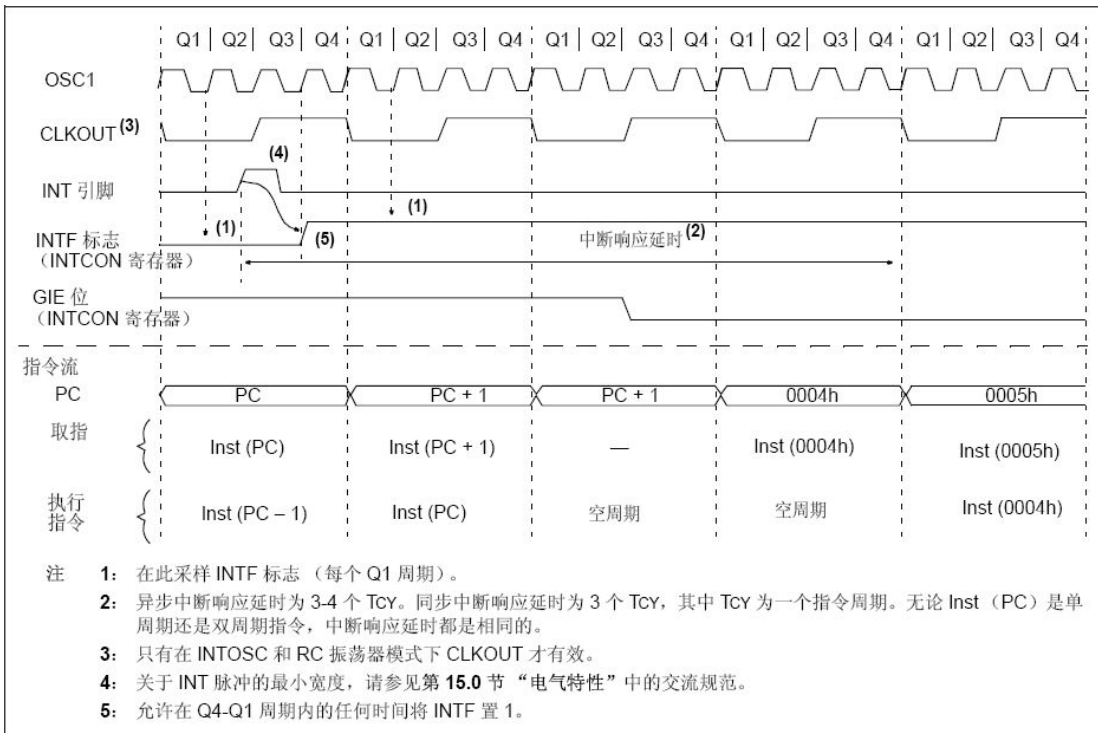


图 5-1-2: INT 引脚中断时序





## 5.1 中断特殊功能寄存器

### 5.1.1 INTS 寄存器

INTS 寄存器是可读写的寄存器，包含 RTCC 寄存器溢出、PORTA 电平变化和外部 PA2/INT 引脚中断的各种允许和标志位。

注： 当有中断条件产生时，不管相应的中断允许位或INTS寄存器的全局允许位GIE状态如何，中断标志位都将置1。用户软件应该在允许中断之前确保将相应的中断标志位清零。

寄存器 0BH/8BH: 中断控制寄存器 (INTS)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>GIE</b>	<b>PEIE</b>	<b>TIS</b>	<b>INS</b>	<b>PAIE</b>	<b>TIF</b>	<b>INTF</b>	<b>PAIF</b>
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7 **GIE**: 全局中断允许位  
1 = 允许所有未屏蔽的中断  
0 = 禁止所有中断

bit6 **PEIE**: 外设中断允许位  
1 = 允许所有未屏蔽的外设中断  
0 = 禁止所有外设中断

bit5 **TIS**: Timer0溢出中断允许位  
1 = 允许Timer0中断  
0 = 禁止Timer0中断

bit4 **INTS**: PA2/INT外部中断允许位  
1 = 允许PA2/INT外部中断  
0 = 禁止PA2/INT外部中断

bit3 **PAIE**: PORTA电平变化中断允许位<sup>(1)</sup>  
1 = 允许PORTA电平变化中断  
0 = 禁止PORTA电平变化中断

bit2 **TIF**: Timer0溢出中断标志位<sup>(2)</sup>  
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

bit1 **INTF**: PA2/INT外部中断标志位  
1 = 发生了PA2/INT外部中断 (必须用软件清零)  
0 = 未发生PA2/INT外部中断

bit0 **PAIF**: PORTA电平变化中断标志位<sup>(1)</sup>  
1 = 至少一个PORTA<5:0>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<5:0>引脚的电平状态发生改变

- 注
- 1: 必须同时使能PAINTR寄存器。
  - 2: 当RTCC计满回零时, TIF位置1。复位时RTCC的状态不变, 它应该在清零TIF位之前被初始化。

## 5.1.2 PIEB1 寄存器

PIEB1 寄存器包含外设中断允许位。

注: 要允许任何一个外设中断, 必须将INTS寄存器的PEIE位置1。

寄存器 8CH: 外设中断允许寄存器1 (PIEB1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EEIE	ADIE	CCP1IE	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

- bit7      **EEIE:** EEPROM写完成中断允许位  
1 = 允许EE写完成中断  
0 = 禁止EE写完成中断
- bit6      **ADIE:** A/D转换器中断允许位  
1 = 允许A/D转换器中断  
0 = 禁止A/D转换器中断
- bit5      **CCP1IE:** CCP1中断允许位  
1 = 允许CCP1中断  
0 = 禁止CCP1中断
- bit4      **C2IE:** 比较器2中断允许位  
1 = 允许比较器2中断  
0 = 禁止比较器2中断
- bit3      **C1IE:** 比较器1中断允许位  
1 = 允许比较器1中断  
0 = 禁止比较器1中断
- bit2      **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
- bit1      **TMR2IE:** Timer2与PR2匹配中断允许位  
1 = 允许Timer2与PR2匹配中断  
0 = 禁止Timer2与PR2匹配中断
- bit0      **TMR1IE:** Timer1溢出中断允许位  
1 = 允许Timer1溢出中断

0 = 禁止Timer1溢出中断

### 5.1.3 PIFB1 寄存器

PIFB1 寄存器包含外设中断标志位。

注： 当有中断条件产生时，不管相应的中断允许位或INTS寄存器的全局允许位GIE状态如何，中断标志位都将置1。用户软件应该在允许中断之前确保将相应的中断标志位清零。

寄存器 0CH: 外设中断请求寄存器1 (PIFB1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>EEIF</b>	<b>ADIF</b>	<b>CCP1IF</b>	<b>C2IF</b>	<b>C1IF</b>	<b>OSFIF</b>	<b>TMR2IF</b>	<b>TMR1IF</b>
bit7						bit0	

图注:

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7 **EEIF**: EEPROM写操作中断标志位  
1 = 写操作完成 (必须用软件清零)  
0 = 写操作尚未完成或尚未启动

bit6 **ADIF**: A/D中断标志位  
1 = A/D转换完成  
0 = A/D转换尚未完成或尚未启动

bit5 **CCP1IF**: CCP1中断标志位  
捕捉模式:  
1 = 发生了TMR1寄存器捕捉 (必须用软件清零)  
0 = 未发生TMR1寄存器捕捉  
比较模式:  
1 = 发生了TMR1寄存器比较匹配 (必须用软件清零)  
0 = 未发生TMR1寄存器比较匹配  
PWM模式: 在此  
模式下未使用

bit4 **C2IF**: 比较器2中断标志位  
1 = 比较器2输出发生了改变 (必须用软件清零)  
0 = 比较器2输出未发生改变

bit3 **C1IF**: 比较器1中断标志位  
1 = 比较器1输出发生了改变 (必须用软件清零)  
0 = 比较器1输出未发生改变

bit2 **OSFIF**: 振荡器故障中断标志位  
1 = 系统振荡器发生故障, 时钟输入切换为INTOSC (必须用软件清零)  
0 = 系统时钟正常运行

- bit1      **TMR2IF**: Timer2与PR2匹配中断标志位  
 1 = 发生了Timer2和PR2的比较匹配（必须用软件清零）  
 0 = 未发生Timer2与PR2的比较匹配
- bit0      **TMR1IF**: Timer1溢出中断标志位  
 1 = Timer1寄存器已经溢出（必须用软件清零）  
 0 = Timer1寄存器未溢出

表 5-1: 与中断相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
INTS	GIE	PEIE	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000	0000 0000
PAINTR	—	—	PAINTR5	PAINTR4	PAINTR3	PAINTR2	PAINTR1	PAINTR0	—00 0000	0000 0000
PIFB1	EEIF	ADIF	CCP1IF	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIEB1	EEIE	ADIE	CCP1IE	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000	0000 0000

图注: x = 未知, u = 不变, — = 未实现 (读为 0), q = 取值视情况而定。中断模块不使用阴影单元。

## 5.2 PA2/INT 中断

PA2/INT 引脚上的外部中断是边沿触发的；当 TMR 寄存器的 IES 位被置 1 时在上升沿触发，而当 IES 位被清零时在下降沿触发。当 PA2/INT 引脚上出现有效边沿时，INTS 寄存器的 INTF 位置 1。可以通过将 INTS 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 PA2/INT 中断能将处理器从休眠状态唤醒。关于 PA2/INT 中断将处理器从休眠状态唤醒的时序，请参见图 4-8。

注： 必须对ANSEL和CMSTA0寄存器进行初始化，以将模拟通道配置为数字输入。配置为模拟输入的引脚总是读为0。

## 5.3 Timer0 中断

RTCC 寄存器溢出 (FFh→00h) 会将 INTS 寄存器的 TOIF 位置 1。可以通过置 1/清零 INTS 寄存器的 TOIE 位来使能/禁止该中断。关于 Timer0 模块的操作，请参见第 7.1 节“Timer0 模块”。

## 5.4 PORTA 电平变化中断

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

注： 当读操作正在执行时发生了I/O引脚电平变化，则PAIF中断标志位可能不会被置1。

## 5.5 EEPROM 写数据中断

为确保 EEPROM 写操作结束时产生中断，您必须将以下位置 1：

- PIEB1 寄存器的 EEPROM 写完成中断允许位 EEIE
- INTS 寄存器的 PEIE 位
- INTS 寄存器的 GIE 位

在中断服务程序中将 PIFB1 寄存器的 EEIF 位清零将清除中断标志。

## 5.6 A/D 中断

ADC 模块在模数转换完成时，PIR1 寄存器中的 ADIF 中断标志位均置 1。可以通过置 1/清零 PIEB1 寄存器中的 ADIE 位来使能/禁止该中断。关于 ADC 模块的中断操作，请参见第 9.2.5 节“ADC 模块中断”。

## 5.7 中断的现场保护

在中断期间，仅将返回的 PC 值压入堆栈。通常情况下，用户可能希望在中断期间保存关键寄存器（例如，W 寄存器和状态寄存器）。这必须用软件实现。应将临时保存寄存器 W\_TEMP 和 STATUS\_TEMP 置于 GPR 的末 16 字节中（见图 2-2）。这 16 个单元是所有存储区共用的，无需分区。这样就简化了现场保护和恢复操作。例 12-1 中所示的代码可用于：

- 保存 W 寄存器
- 保存状态寄存器
- 执行 ISR 代码
- 恢复状态寄存器（和存储区选择位寄存器）
- 恢复 W 寄存器

注：EN8F683 通常不需要保存 PCHLAT。但是，如果要在 ISR 和主程序中使用计算 GOTO，就必须在 ISR 中保存和恢复 PCHLAT。

例 5-7：将状态寄存器和 W 寄存器保存在 RAM 中

```

STWR    W_TEMP           ; Copy W to TEMP register
SWAPR   STATUS, W        ; Swap status to be saved into W
; Swaps are used because they do not affect the status bits
STWR    STATUS_TEMP     ; Save status to bank zero STATUS_TEMP register
;
; (ISR)                  ; Insert user code here
;
SWAPR   STATUS_TEMP, W   ; Swap STATUS_TEMP register into W
; (sets bank to original state)
STWR    STATUS           ; Move W into STATUS register
SWAPR   W_TEMP, R       ; Swap W_TEMP
SWAPR   W_TEMP, W       ; Swap W_TEMP into W

```

## 6.0 I/O 端口

此系列器件共有 14 个通用 I/O 引脚。根据使能的外设不同，有些（或全部）引脚不能用作通用 I/O。通常使能了一个外设后，相关的引脚就不能用作通用 I/O 引脚了。

### 6.1 PORTA 和 CPIOA 寄存器

PORTA 是 6 位宽的双向端口。PORTA 对应的数据方向寄存器是 CPIOA（[寄存器 85H](#)）。将 CPIOA 位置 1（= 1）可以使对应的 PORTA 引脚作为输入引脚（即禁止相应的输出驱动器）。将 CPIOA 位清零（= 0）将使对应的 PORTA 引脚作为输出引脚（即使能输出驱动器并将输出锁存器的内容置于所选的引脚上）。PA3 是个例外，它只能作为输入引脚，CPIO 位始终读为 1。[例 6-1](#) 给出了初始化 PORTA 的方法。

读 PORTA 寄存器（[寄存器 05H](#)）将读取引脚的状态而写该寄存器将会写入端口锁存器。所有写操作都是读—修改—写操作。因此，写一个端口就意味着读该端口的引脚电平，修改读到的值，然后再将改好的值写入端口数据锁存器。当 MCLRE = 1 时，PA3 读为 0。

即使在 PORTA 引脚被用作模拟输入的时候，CPIOA 寄存器仍然控制 PORTA 引脚的方向。在将它们用作模拟输入时，用户必须确保 CPIOA 寄存器中的位保持为置 1 状态。配置为模拟输入的 I/O 引脚始终读为 0。

注： 必须对 ANSEL 和 CMSTA 寄存器进行初始化以将模拟通道配置为数字输入通道。配置为模拟输入的引脚读为 0。

例 6-1：初始化 PORTA

```
BCR    STATUS,PAGE    ;Bank0
CLRR   PORTA          ;Init PORTA
LDWI   05H            ;Set PA<2:0> to
STWR   CMSTA0         ;digital I/O
BSR    STATUS,PAGE    ;Bank1
CLRR   ANSEL          ; digital I/O
LDWI   0CH            ;Set PA<2:0> as inputs
STWR   CPIOA          ;and set PA<5:4 , 1:0>
                          ; as outputs
BCR    STATUS,PAGE    ;Bank0
```

寄存器 05H：PORT A 寄存器（PORTA）

U-0	U-0	R/W-x	R/W-0	R-x	R/W-0	R/W-0	R/W-0
—	—	PA5	PA4	PA3	PA2	PA1	PA0
bit7							bit0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit7-6 未实现：读为 0

bit5-0 **PA<5:0>**：PORT A I/O 引脚位

1 = PORT A 引脚电平 > V<sub>IH</sub>

0 = PORT A 引脚电平 < V<sub>IL</sub>

寄存器 **85H: PORT A**三态寄存器 (**CPIOA**)

U-0	U-0	R/W-1	R/W-1	R-1	R/W-1	R/W-1	R/W-1
—	—	<b>CPIOA5</b>	<b>CPIOA4</b>	<b>CPIOA3</b>	<b>CPIOA2</b>	<b>CPIOA1</b>	<b>CPIOA0</b>
bit7							bit0

图注:			
R = 可读位	W = 可写位	U = 未实现位, 读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

- bit7-6 未实现：读为0
- bit5-0 **CPIOA<5:0>**: PORT\_A三态控制位  
 1 = PORT A引脚配置为输入（三态）  
 0 = PORT A引脚配置为输出
- 注 1: CPIOA<3>始终读为1。  
 2: 在XT、HS和LP振荡模式下，CPIOA<5:4>始终读为1。

## 6.3 其它引脚功能

EN8F683上的每一个 PORTA 引脚都具有电平变化中断和弱上拉功能(PA3 做为复位脚时，使能上拉。作为 IO 口时，没有上拉电阻)。PA0 具有超低功耗唤醒功能。下面三个小节将介绍这些功能。

### 6.3.1 ANSEL 寄存器

ANSEL 寄存器用于将 I/O 引脚的输入模式配置为模拟。将相应的 ANSEL 位置为高电平将使对该引脚的所有读操作结果为 0，并使该引脚的模拟功能正常进行。

ANSEL 位的状态对数字输出功能没有影响。CPIO 清零且 ANSEL 置 1 的引脚仍将作为数字输出工作，但其输出模式将为模拟。在对受影响端口执行读 - 修改 - 写指令时，这将导致意外的操作。

### 6.3.2 弱上拉

每一个 PORTA 引脚（除 PA3）具有各自的可配置内部弱上拉。控制位 WPUAx 使能或禁止每一个弱上拉。请参见 [寄存器 95H](#)。当将端口引脚配置为输出时，其弱上拉电路会自动切断。在上电复位时，由  $\overline{\text{PAPU}}$  位（TMR<7>）禁止上拉功能。在 PA3 为 I/O 且配置为 MCLR 并禁止时，自动启动 PA3 的弱上拉功能。MCLR 上拉不受软件控制。

寄存器 95H: 弱上拉 PORTA 寄存器 (PAPHR)

U-1	U-1	R/W-1	R/W-1	U-1	R/W-1	R/W-1	R/W-1
—	—	<b>PHA5</b>	<b>PHA4</b>	—	<b>PHA2</b>	<b>PHA1</b>	<b>PHA0</b>
bit7							bit0

图注:

R = 可读位	W = 可写位	U = 未实现位，读为0
-n = POR时的值	1 = 置1	0 = 清零
		x = 未知

- bit7-6 未实现：读为0
- bit5-4 **PHA<5:4>**: 弱上拉控制位  
 1 = 使能上拉  
 0 = 禁止上拉
- bit3 未实现：读为0
- bit2-0 **PHA<2:0>**: 弱上拉控制位  
 1 = 使能上拉



0 = 禁止上拉

注

- 1: 必须使能TMR寄存器的全局  $\overline{\text{PAPH}}$  位以使能各个上拉。
- 2: 如果引脚处于输出模式 ( $\text{CPIOC} = 0$ )，则弱上拉器件被自动禁止。
- 3: 在配置字寄存器中配置为  $\overline{\text{MCLR}}$  时PA3上拉被使能，配置为I/O时上拉被禁止。
- 4: 在XT、HS和LP振荡模式下  $\text{PHA}\langle 5:4 \rangle$  始终读为1。

### 6.3.3 电平变化中断

每一个 PORTA 引脚均可分别配置为电平变化中断引脚。控制位  $\text{PAINTR}_x$  使能或禁止每个引脚的中断功能。请参见[寄存器 85H](#)。在上电复位时禁止电平变化中断。

对于已允许电平变化中断的引脚，则将该引脚上的值同上一次读 PORTA 时锁存的值进行比较。将上一次“不匹配”的输出一起作逻辑或运算，以便将 INTS 寄存器([寄存器 0BH](#))中 PORTA 电平变化中断标志位(PAIF)置 1。

该中断能唤醒休眠下的器件。用户在中断服务程序中通过以下方式清除中断：

- a) 对 PORTA 进行读或写操作。这将结束引脚电平不匹配条件。
- b) 将标志位 PAIF 清零。

电平不匹配条件会继续将 PAIF 标志位置 1。而读 PORTA 将结束不匹配条件并允许将 PAIF 标志位清零。锁存器将保持最后一次读取的值不受  $\overline{\text{MCLR}}$  和欠压复位的影响。在这些复位之后，如果出现电平不匹配，PAIF 标志位将继续被置 1。

注： 在执行任何PORTA操作时如果I/O引脚的电平发生变化，则PAIF中断标志位可能不会被置1。

寄存器 96H: 电平变化中断PORTA寄存器 (PAINTR)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	PAINTR5	PAINTR4	PAINTR3	PAINTR2	PAINTR1	PAINTR0
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-6 未实现: 读为0。

bit5-0 **PAINTR<5:0>**: 电平变化中断PORT A控制位

1 = 允许电平变化中断

0 = 禁止电平变化中断

注

1: 必须允许全局中断允许 (GIE) 以使各中断被识别。

2: 在XT、HS和LP振荡模式下PAINTR&lt;5:4&gt;始终读为1。

### 6.3.4 超低功耗唤醒

PA0 上的超低功耗唤醒允许电压缓慢跌落, 从而可在不消耗额外电流的情况下, 产生 PA0 电平变化中断。通过将 ULPWUE 位 (PSTA<5>) 置 1 选择该模式。这将产生一个很小的灌电流, 可用于将 PA0 上的电容放电。

要使用该功能, PA0 引脚应先被配置为输出高电平以对电容充电, 允许 PA0 的电平变化中断并且 PA0 要配置为输入引脚。将 ULPWUE 置 1 开始放电, 执行 SLEEP 指令。当 PA0 上的电压下降到 VIL 后, 器件将被唤醒并执行下一条指令。如果 INTS 寄存器的 GIE 位置 1, 器件将调用中断服务程序 (0004h)。更多信息, 请参见第 6.3.3 节“电平变化中断”和第 5.4 节“PORT A 电平变化中断”。

该功能提供了低功耗技术, 可周期性地唤醒休眠下的器件。延时取决于 PA0 上 RC 电路的放电时间。要了解如何初始化超低功耗唤醒模块, 请参见例 6-3-4。

串联电阻提供了 PA0 引脚的过电流保护功能, 允许在软件中对延时进行校准 (见图 6-3-1)。可用定时器测量电容的充放电时间。然后调整充电时间以提供所需的中断延时。该技术可补偿温度、电压和元件精度所带来的影响。超低功耗唤醒外设还可以配置为简单可编程低压检测设备或温度传感器。

例 6-3-4: 超低功耗唤醒的初始化

BCR	STATUS,PAGE	;Bank 0
BSR	PORTA,0	;Set PA0 data latch
LDWI	H'7'	;Turn off
STWR	CMSTA	;comparators
BSF	STATUS,PAGE	;Bank 1
BCR	ANSEL,0	;PA0 to digital I/O
BCR	CPIOA,0	;Output high to
LCALL	CapDelay	; charge capacitor
BSR	PSTA,ULPWUE	;Enable ULP Wake-up
BSR	PAINTR,0	;Select PA0 IOC
BSR	CPIOA,0	;PA0 to input
LDWI	B'10001000'	;Enable interrupt
STWR	INTS	; and clear flag
SLEEP		;Wait for IOC

## 6.4 PORTA 引脚说明和引脚图

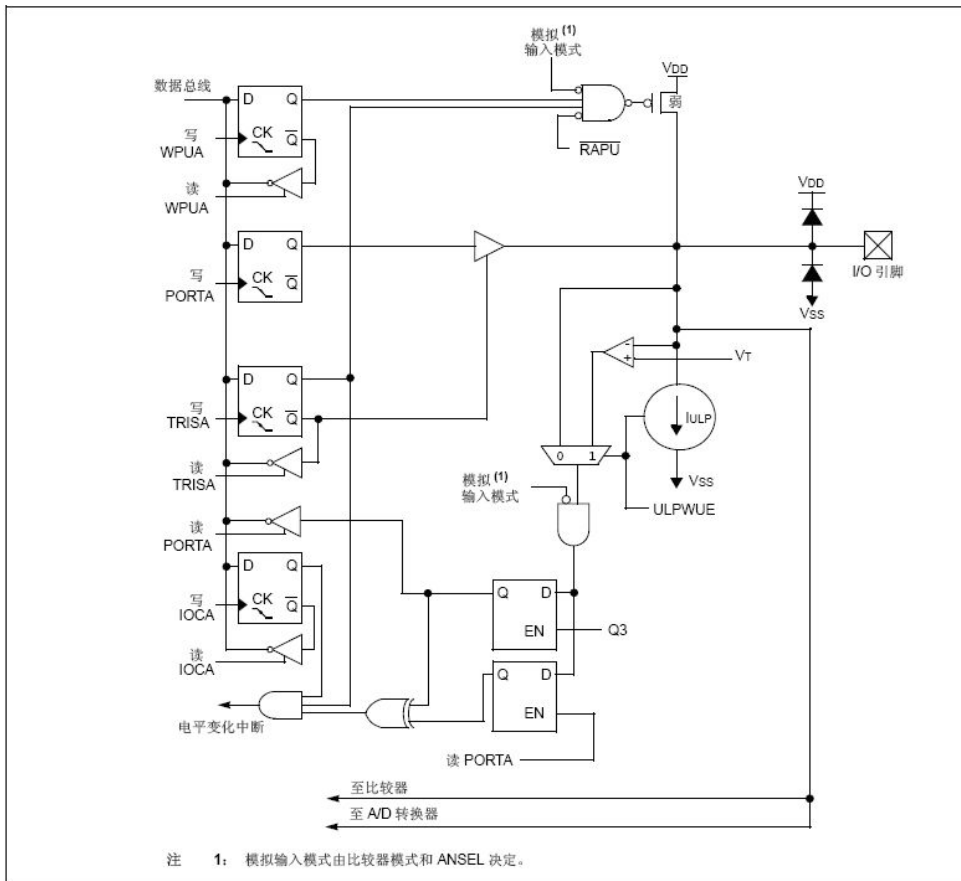
每个 PORTA 引脚都与其他功能复用。这里将简要说明引脚及其复合功能。各功能如比较器或 A/D 转换器 (A/D Converter, ADC) 的具体信息, 请参见本数据手册中的相关章节。

### 6.4.1 PA0/AN0/C1IN+/SCK/ULPWU

图 6-4-1 给出了此引脚的引脚图。PA0/AN0/C1IN+/SCK/ULPWU 引脚可配置为下列功能之一:

- 通用 I/O
- 连接至 ADC 的模拟输入
- 连接至比较器的模拟输入
- 在线串行编程 (In-Circuit Serial Programming™) 烧录和调试的时钟 SCK
- 超低功耗唤醒的模拟输入

图 6-4-1: PA0 框图

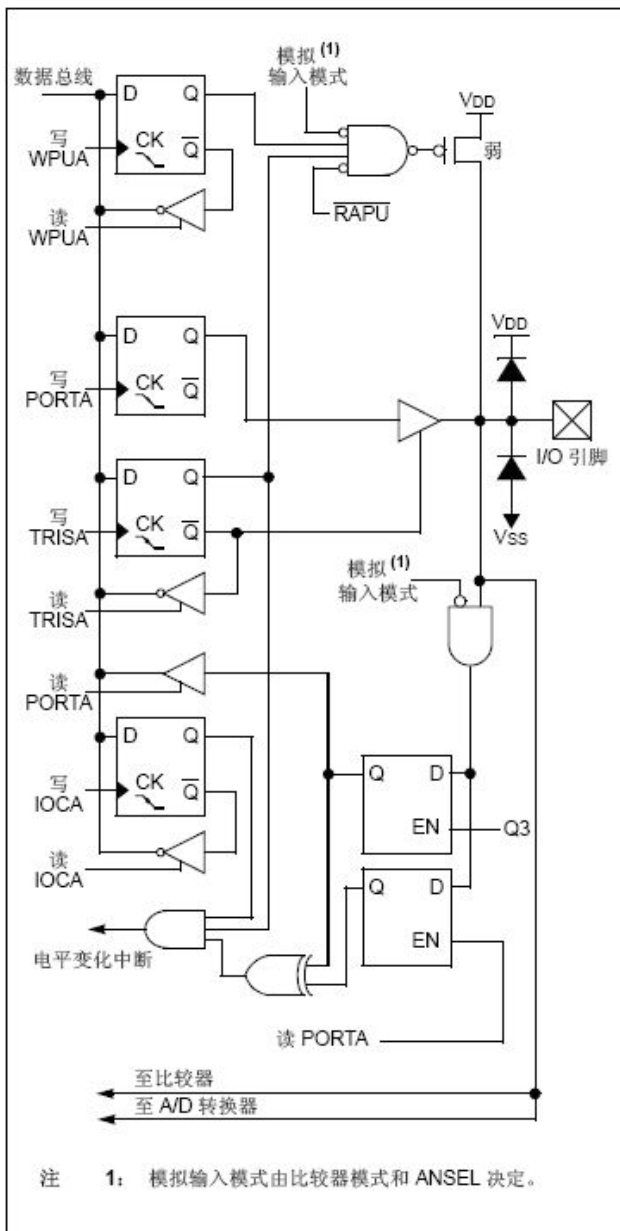


### 6.4.2 PA1/AN1/C1IN-/VREF/SDA

图 6-4-2 给出了此引脚的引脚图。PA1/AN1/C1IN-/VREF/SDA 引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入
- 连接至比较器的模拟输入
- ADC 的参考电压输入
- 在线串行编程烧录和调试的数据端口 SDA

图 6-4-2: PA1 框图

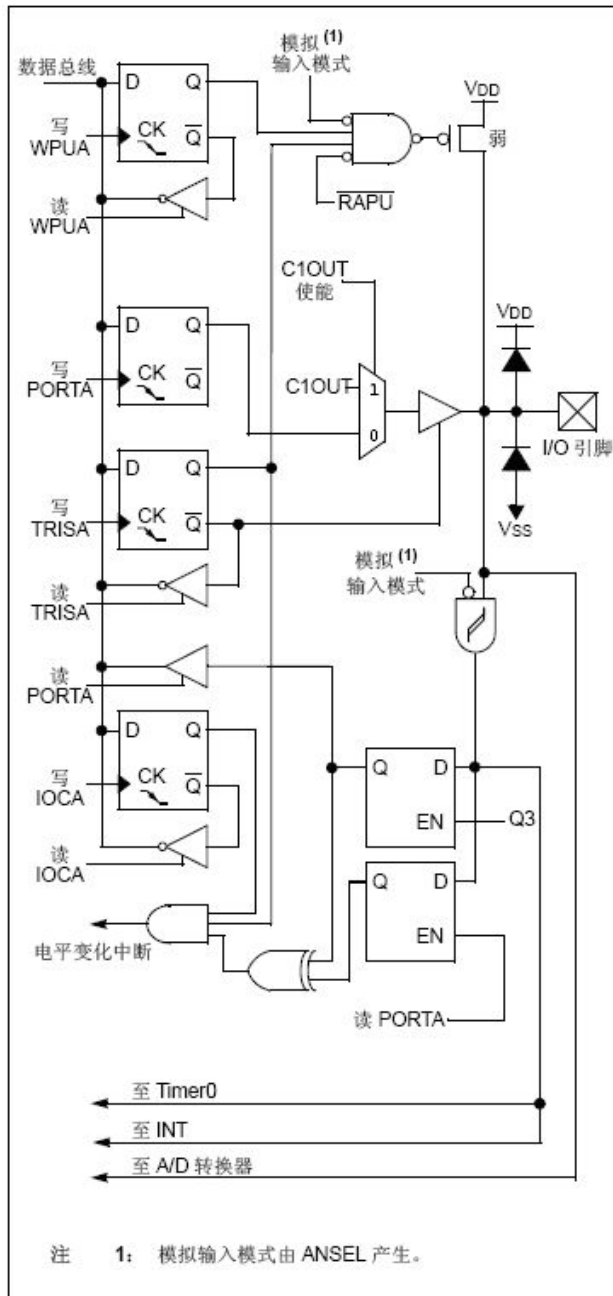


### 6.4.3 PA2/AN2/T0CKI/INT/C1OUT/EE\_EW\_CLK

图 6-4-3 给出了此引脚的引脚图。PA2/AN2/T0CKI/INT/C1OUT/EE\_EW\_CLK 引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入
- RTCC 的时钟输入
- 外部边沿触发的中断
- 来自比较器 1 的数字输出
- DATA EEPROM Erase and Write Self\_timer Clock

图 6-4-3: PA2 框图



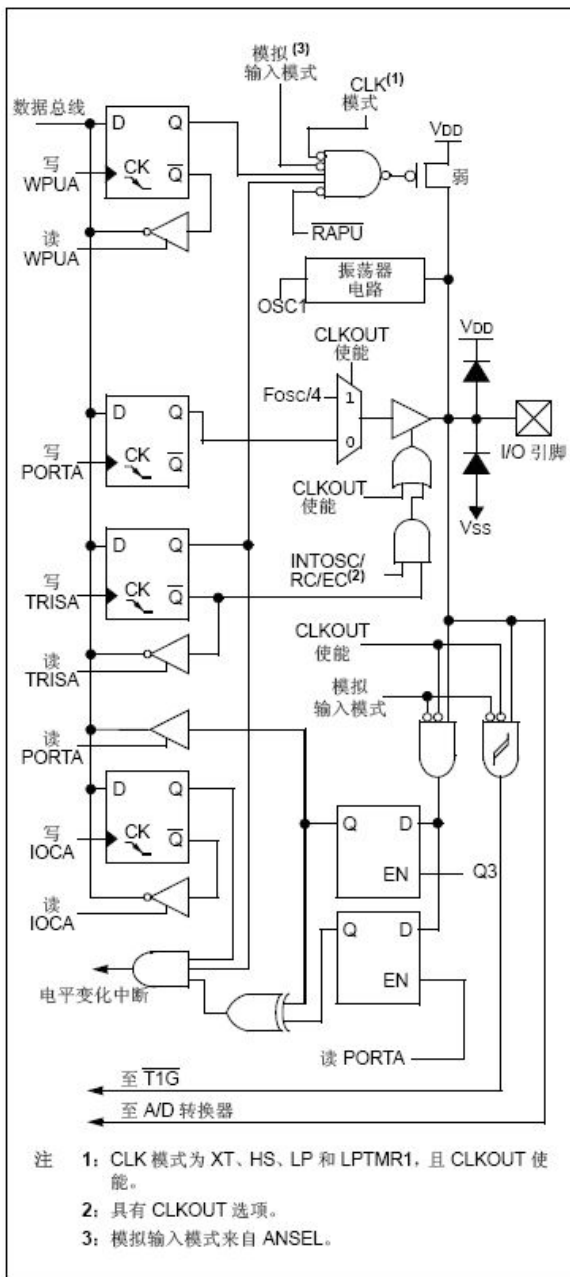


### 6.4.5 PA4/AN3/T1G/OSC2/CLKOUT

图 6-4-5 给出了此引脚的引脚图。PA4/AN3/ T1G /OSC2/CLKOUT 引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入
- Timer1 门控（计数使能）
- 晶振/谐振器连接
- 时钟输出

图 6-4-5: PA4 框图



### 6.4.6 PA5/T1CKI/OSC1/CLKIN

图 6-4-6 给出了此引脚的引脚图。PA5/T1CKI/OSC1/CLKIN 引脚可配置为下列功能之一：

- 通用 I/O
- TMR1 时钟输入
- 晶振/谐振器连接
- 时钟输入

图 6-4-6: PA5 框图

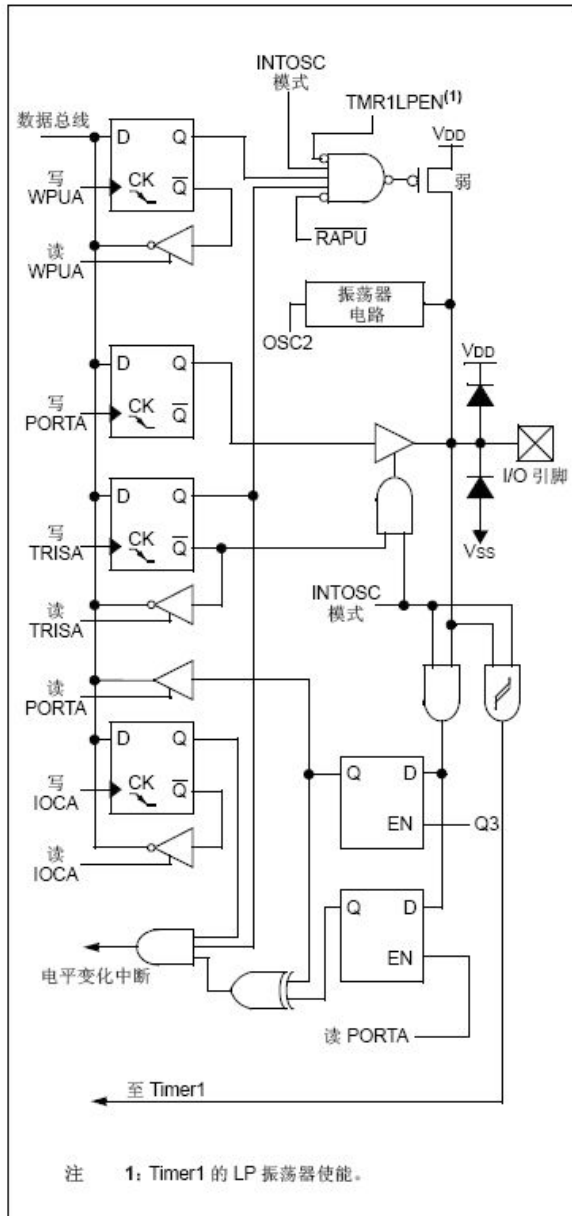




表 6-4: 与 PORT A 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
ADINS	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
CMSTA	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
PSTA	—	—	ULPWUE	SBOREN	—	—	POR	BOR	--01 --qq	--0u --uu
INTS	GIE	PE1E	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000	0000 0000
PAINTR	—	—	PAINTR5	PAINTR4	PAINTR3	PAINTR2	PAINTR1	PAINTR0	--00 0000	--00 0000
TMR	PAPH	IES	TCS	TCE	PSC	PS2	PS1	PS0	1111 1111	1111 1111
PORTA	—	—	PA5	PA4	PA3	PA2	PA1	PA0	--x0 x000	--uu uu00
CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11 1111	--11 1111
PAPHR	—	—	PHA5	PHA4	—	PHA2	PHA1	PHA0	--11 -111	--11 -111

图注: x = 未知, u = 不变, — = 未实现 (读为 0)。PORT A 不使用阴影单元。

## 6.5 PORT C 引脚说明和引脚图

### 6.5.1 PC0/AN4/C2IN+

PC0 引脚可以配置为下列功能之一:

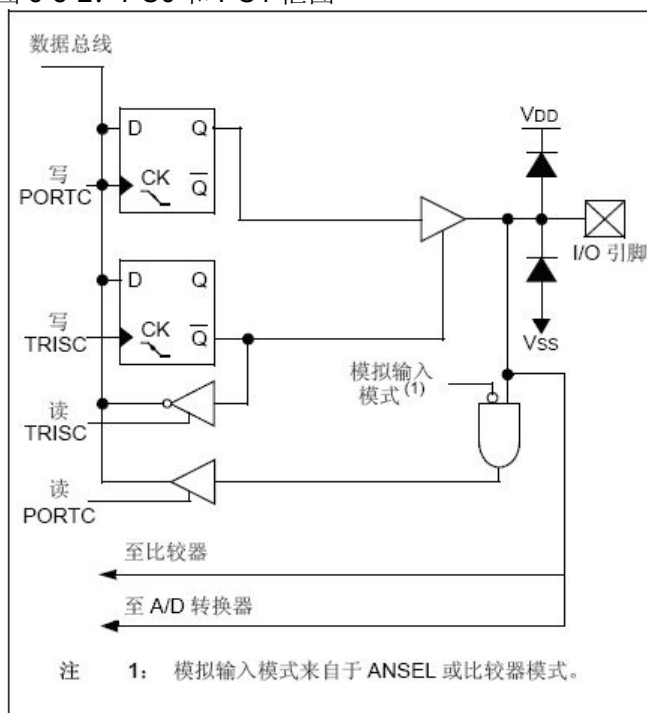
- 通用 I/O
- ADC 的模拟输入
- 比较器的模拟输入

### 6.5.2 PC1/AN5/C2IN-

PC1 引脚可以配置为下列功能之一:

- 通用 I/O
- ADC 的模拟输入
- 比较器的模拟输入

图 6-5-2: PC0 和 PC1 框图



### 6.5.3 PC2/AN6/P1D

PC2 引脚可以配置为下列功能之一：

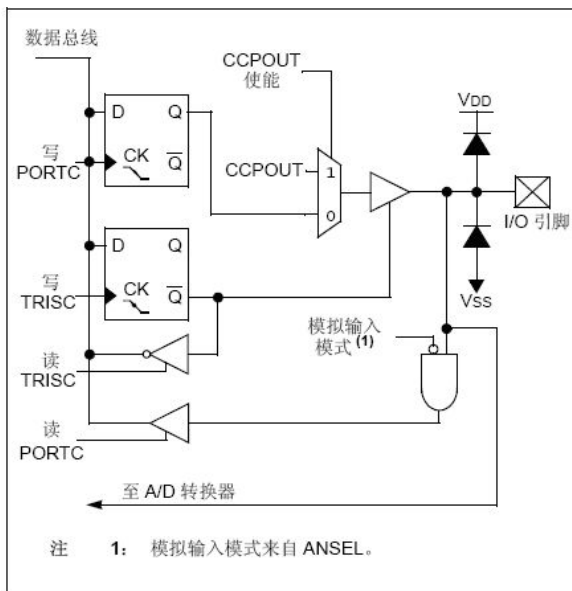
- 通用 I/O
- ADC 的模拟输入
- 来自增强型 CCP 的数字输出

### 6.5.4 PC3/AN7/P1C

PC3 引脚可以配置为下列功能之一：

- 通用 I/O
- ADC 的模拟输入
- 来自增强型 CCP 的数字输出

图 6-5-4：PC2 和 PC3 框图



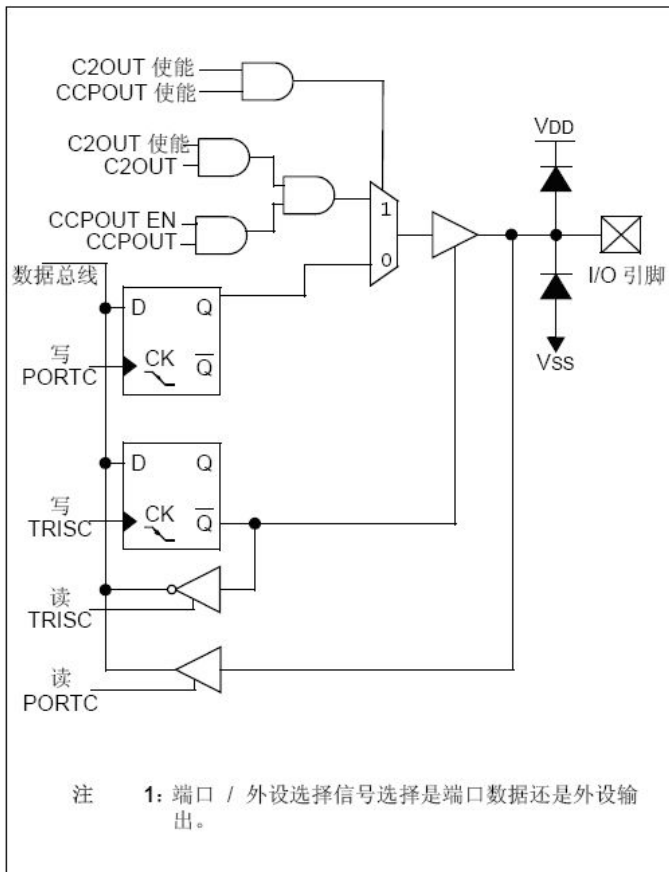
### 6.5.5 PC4/C2OUT/P1B

PC4 引脚可以配置为下列功能之一：

- 通用 I/O
- 来自比较器的数字输出
- 来自增强型 CCP 的数字输出

注：使能 C2OUT 和 P1B 将在 PC4 上引发冲突，造成不可预期的结果。所以，如果使能了 C2OUT，则不能将 ECCP 用于半桥或全桥模式，反之亦然。

图 6-5-5: PC4 框图



### 6.5.6 PC5/CCP1/P1A

PC5 引脚可以配置为下列功能之一:

- 通用 I/O
- 来自增强型 CCP 的数字输入/输出

图 6-5-6: PC5 框图



## 6.5.7 PC6

PC6 引脚可以配置为下列功能之一：

- 通用 I/O

## 6.5.8 PC7

PC7 引脚可以配置为下列功能之一：

- 通用 I/O

表 6-5: 与 PORT C 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
ADINS	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
CMSTA	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	xxxx 0000	uuuu uu00
CPIOC	CPIOC7	CPIOC6	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	1111 1111	1111 1111

图注： x = 未知， u = 不变， — = 未实现（读为 0）。PORT C 不使用阴影单元。

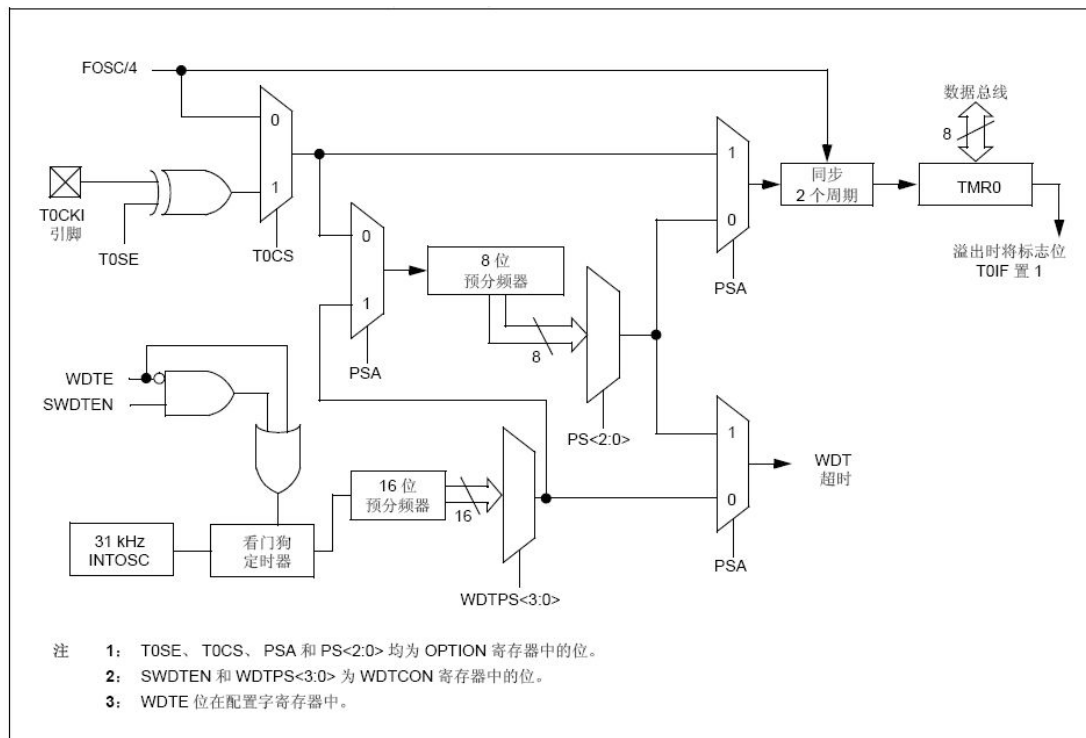
## 7.0 定时器

### 7.1 Timer0 模块

Timer0 模块是 8 位定时器/计数器，具备以下特性：

- 8 位定时器/计数器寄存器（RTCC）
- 8 位预分频器（与看门狗定时器共用）
- 可编程内部或外部时钟源
- 可编程外部时钟边沿选择
- 溢出中断

图 7-1：Timer0 模块的框图



#### 7.1.1 Timer0 的工作原理

作为定时器使用时，Timer0 模块可用作 8 位定时器或 8 位计数器。

寄存器 01H: Timer0 模块寄存器（RTCC）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RTCC7	RTCC6	RTCC5	RTCC4	RTCC3	RTCC2	RTCC1	RTCC0
bit7							bit0

图注：

R = 可读位	W = 可写位	U = 未实现位，读为0
-n = POR时的值	1 = 置1	0 = 清零
		x = 未知

bit7-0 RTCC <7:0>: 8位定时器/计数器寄存器

## 7.1.2 8 位定时器模式

作为定时器使用时，Timer0 模块将在每个指令周期递增（无预分频器）。将 TMR 寄存器（[寄存器 81H](#)）的 TCS 位清零选择定时器模式。

当 RTCC 被写入时，写入后将立即禁止 2 周期递增。

注：在 RTCC 被写入时，为了计入两个指令周期的延时，可以调整写入 RTCC 寄存器的值。

## 7.1.3 8 位计数器模式

作为计数器使用时，Timer0 模块将在 T0CKI 引脚的每个上升沿递增。递增边沿由 TMR 寄存器（[寄存器 81H](#)）的 TCE 位决定。将 TMR 寄存器的 TCS 位置 1 选择计数器模式。

## 7.1.4 软件可编程预分频器

Timer0 或看门狗定时器（WDT）之一可使用一个软件可编程预分频器，但两者不能同时使用。预分频器的分配由 TMR 寄存器的 PSC 位控制。要将预分频器分配给 Timer0，必须将 PSC 位清零。

Timer0 模块的预分频比有 8 种选项，从 1:2 至 1:256。预分频比可通过 TMR 寄存器（[寄存器 81H](#)）的 PS<2:0>位进行选择。要使 Timer0 模块得到 1:1 的预分频比，必须将预分频器分配给 WDT 模块。

预分频器不可读写。当预分频器被分配给 Timer0 模块时，所有写入 RTCC 寄存器的指令均会将预分频器清零。当预分频器被分配给 WDT 时，一条 CLRWT 指令将同时清零预分频器和 WDT。

### 7.1.4.1 在 Timer0 和 WDT 模块间切换预分频器

由于预分频器可分配给 Timer0 或 WDT，因此在切换预分频比时可能产生不想要的器件复位。当把预分频器从 Timer0 切换到 WDT 模块时，必须执行[例 7-1-4-1-1](#)所示的指令序列。

例 7-1-4-1-1: 更改预分频器 (TIMER0→WDT)

BCR	STATUS,PAGE	;Bank 0
CLRWT		;Clear WDT
CLRR	TMR0	;Clear TMR0 and
		; prescaler
BSR	STATUS,PAGE	;Bank 1
LDWI	b'00101111'	;Required if desired
TMODE		; PS2:PS0 is
CLRWT		; 000 or 001
		;
LDWI	b'00101xxx'	;Set postscaler to
TMODE		; desired WDT rate
BCR	STATUS,PAGE	;Bank 0

当将预分频器从 WDT 切换到 Timer0 模块时，必须执行以下指令序列（见例 7-1-4-1-2）。

例 7-1-4-1-2: 更改预分频器 (WDT→TIMER0)

```

CLRWT                ;Clear WDT and
                    ;prescaler
BSR  STATUS,PAGE    ;Bank 1
LDWI  b'xxxx0xxx'  ;Select TMR0,
                    ;prescale, and
                    ;clock source
TMODE                ;
BCR  STATUS,PAGE    ;Bank 0

```

## 7.1.5 Timer0 中断

TMR0 寄存器从 FFh 溢出到 00h 时，Timer0 将产生中断。INTS 寄存器的 TIF 中断标志位将在每次 RTCC 寄存器溢出时被置 1，无论是否允许了 Timer0 中断。必须用软件将 TIF 位清零。Timer0 中断允许位是 INTS 寄存器的 TIE 位。

注： 由于定时器在休眠时被冻结，因此Timer0中断无法将处理器从休眠中唤醒。

## 7.1.6 Timer0 与外部时钟配合使用

Timer0 处理计数器模式时，T0CKI 输入和 Timer0 寄存器的同步是通过采样内部相位时钟的 Q2 和 Q4 周期实现的。因此，外部时钟源的高低周期必须满足第 13.0 节“电气特性”中所列的时序要求。

表 7-1-6: 与 TIMER0 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
RTCC	Timer0 模块寄存器								xxxx xxxx	uuuu uuuu
INTS	GIE	PE1E	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000	0000 0000
TMR	PAPH	IES	TCS	TCE	PSC	PS2	PS1	PS0	1111 1111	1111 1111
CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11 1111	--11 1111

图注： x = 未知， u = 不变， — = 未实现（读为 0）。Timer0 不使用阴影单元。

## 7.2 带门控的 Timer1 模块

Timer1 模块是 16 位定时器/计数器，具备以下特性：

- 16 位定时器/计数器寄存器（TMR1H:TMR1L）
- 可编程内部或外部时钟源
- 3 位预分频器
- 可选 LP 振荡器
- 同步或异步操作
- 通过比较器或 T1G 引脚的 Timer1 门控（计数使能）





- bit6 **TMR1GE**: Timer1门控使能位<sup>(2)</sup>  
 如果TMR1ON = 0:  
 此位被忽略 如果  
 TMR1ON = 1:  
 1 = Timer1在Timer1门控不活动时打开  
 0 = Timer1打开
- bit5-4 **T1CKPS<1:0>**: Timer1输入时钟预分频比选择位  
 11 = 1:8预分频比  
 10 = 1:4预分频比  
 01 = 1:2预分频比  
 00 = 1:1预分频比
- bit3 **T1OSCEN**: LP振荡器使能控制位 如果无  
 CLKOUT振荡器的INTOSC处于激活状态:  
 1 = LP振荡器被使能用于Timer1时钟  
 0 = LP振荡器关闭  
 否则:  
 此位被忽略
- bit2 **T1SYNC**: Timer1外部时钟输入同步控制位  
 TMR1CS = 1:  
 1 = 不同步外部时钟输入  
 0 = 同步外部时钟输入 TMR1CS  
 = 0: 此位被忽略。Timer1使用内  
 部时钟
- bit1 **TMR1CS**: Timer1时钟源选择位  
 1 = 来自T1CK1引脚（上升沿）的外部时钟  
 0 = 内部时钟（FOSC/4）
- bit0 **TMR1ON**: Timer1打开位  
 1 = 使能Timer1  
 0 = 停止Timer1
- 注 1: 无论门控源为何, T1GINV位均将翻转Timer1门控逻辑。  
 2: 必须将TMR1GE位置1以使用T1G引脚或C2OUT, 使用哪个引脚作为Timer1的门控源由  
 CM2CON1寄存器的T1GSS位选择。

Timer1 模块是 16 位定时器/计数器, TMR1H:TMR1L。

寄存器 **0EH**: 16位Timer1低字节的保持寄存器 (TMR1L)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
<b>TMR1&lt;7&gt;</b>	<b>TMR1&lt;6&gt;</b>	<b>TMR1&lt;5&gt;</b>	<b>TMR1&lt;4&gt;</b>	<b>TMR1&lt;3&gt;</b>	<b>TMR1&lt;2&gt;</b>	<b>TMR1&lt;1&gt;</b>	<b>TMR1&lt;0&gt;</b>
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-0 **TMR1 <7:0>**: 16位Timer1定时器/计数器低字节寄存器

寄存器 **0FH**: 16位Timer1高字节的保持寄存器 (TMR1H)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
TMR1<15>	TMR1<14>	TMR1<13>	TMR1<12>	TMR1<11>	TMR1<10>	TMR1<9>	TMR1<8>
bit7							bit0

图注:

R = 可读位

-n = POR时的值

W = 可写位

1 = 置1

U = 未实现位, 读为0

0 = 清零

x = 未知

bit7-0      **TMR1 <15:8>**: 16位Timer1定时器/计数器高字节寄存器

## 7.2.2 时钟源选择

T1STA 寄存器的 TMR1CS 位用于选择时钟源。当 TMR1CS = 0 时, 时钟源为 FOSC/4。TMR1CS = 1 时, 时钟源由外部提供。

时钟源	TMR1CS
FOSC/4	0
T1CKI引脚	1

### 7.2.2.1 内部时钟源

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

### 7.2.2.2 外部时钟源

选择外部时钟源时, Timer1 模块可作为定时器, 也可作为计数器工作。

计数时, Timer1 在外部时钟输入 T1CKI 的上升沿递增。此外, 计数器模式时钟可同步到单片机系统时钟, 也可异步运行。

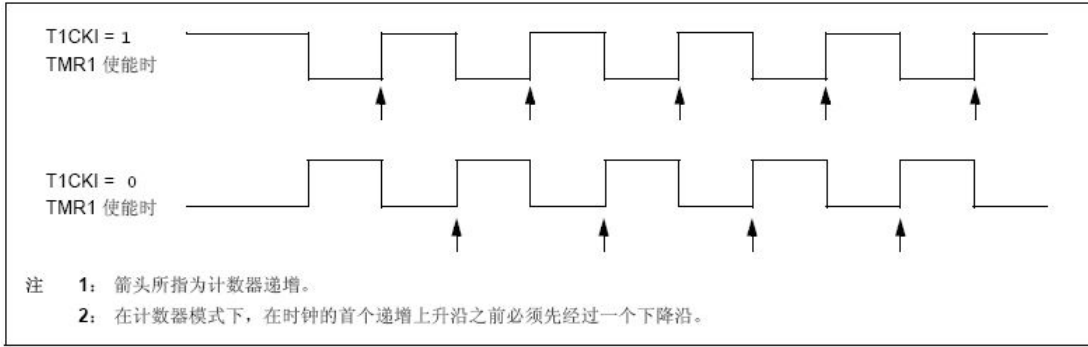
若需外部时钟振荡器(且单片机使用不带 CLKOUT 的 INTOSC), 则 Timer1 可使用 LP 振荡器作为时钟源。

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

- POR复位后使能Timer1
- 写入TMR1H或TMR1L
- Timer1被禁止
- T1CKI为高电平时Timer1被禁止
- (TMR1ON = 0), 然后在T1CKI为低电平时Timer1被使能 (TMR1ON = 1)。

请参见[图7-2-2](#)

图 7-2-2: TIMER1 递增边沿



## 7.2.3 Timer1 的预分频器

Timer1 有四种预分频选项，可对时钟输入进行 1、2、4 或 8 分频。T1STA 寄存器的 T1CKPS 位控制预分频计数器。预分频计数器不可直接读写；但当发生对 TMR1H 或 TMR1L 的写操作时，预分频计数器被清零。

## 7.2.4 Timer1 振荡器

OSC1（输入）引脚与 OSC2（放大器输出）引脚之间内置有一个低功耗 32.768kHz 晶振。T1STA 寄存器将的 T1OSCEN 控制位置 1 使能振荡器。休眠时振荡器继续工作。

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

Timer1 振荡器被使能时，CPIOA5 和 CPIOA4 位置 1。

PA5 和 PA4 位读为 0 且 CPIOA5 和 CPIOA4 位读为 1。

注： 振荡器在使用前需要一段起振和稳定时间。这样，应将 T1OSCEN 置 1 并在使能 Timer1 前应有一段适当的延时。

## 7.2.5 Timer1 工作于异步计数器模式

若 T1STA 寄存器的控制位 T1SYNC 置 1，则外部时钟输入不同步。定时器与内部相位时钟异步递增。若选择了外部时钟源，则定时器将在休眠时继续运行，并可在溢出时产生中断，唤醒处理器。然而，读写定时器时应特别谨慎（见第 7.2.5.1 节“在异步计数器模式下读写 Timer1”）。

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

### 7.2.5.1 在异步计数器模式下读写 Timer1

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

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

## 7.2.6 Timer1 门控

Timer1 门控源可软件配置为  $\overline{T1G}$  引脚或比较器 C2 的输出。这使器件可直接使用  $\overline{T1G}$  为外部事件定时，或使用比较器 C2 对模拟事件定时。Timer1 门控源的选择请参见 CMCON1 寄存器（[寄存器 99H](#)）。这个特性可以简化  $\Delta$ - $\Sigma$  A/D 转换器和许多其他应用的程序。

注： T1STA 寄存器的 TMR1GE 位必须置 1 以将  $\overline{T1G}$  或 C2OUT 用作 Timer1 门控源。更多有关选择 Timer1 门控源的信息，请参见 CMCON1 寄存器（[寄存器 99H](#)）。

使用 T1STA 寄存器的 T1GINV 位可翻转 Timer1 门控，不论其来自  $\overline{T1G}$  引脚还是比较器 C2 的输出。这将配置 Timer1 以确保事件之间存在低电平有效或高电平有效的的时间。

## 7.2.7 Timer1 中断

Timer1 的一对寄存器（TMR1H:TMR1L）递增至 FFFFh 后返回 0000h。Timer1 计满返回时，PIFB1 寄存器的 Timer1 中断标志位被置 1。为确保计满返回时产生中断，您必须将以下位置 1：

- T1STA 寄存器的 TMR1ON 位
- PIEB1 寄存器的 TMR1IE 位
- INTS 寄存器的 PEIE 位
- INTS 寄存器的 GIE 位

在中断服务程序中将 TMR1IF 位清零将清除中断标志。

注： TMR1H:TMR1L 这对寄存器和 TMR1IF 位应在允许中断前清零。

## 7.2.8 Timer1 在休眠模式下的工作原理

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

- 必须将 T1STA 寄存器的 TMR1ON 位置 1
- 必须将 PIEB1 寄存器的 TMR1IE 位置 1
- 必须将 INTS 寄存器的 PEIE 位置 1
- 必须将 T1STA 寄存器的 T1SYNC 位置 1
- 必须将 T1STA 寄存器的 TMR1CS 位置 1
- 可将 T1STA 寄存器的 T1OSCEN 位置 1

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

## 7.2.9 ECCP 捕捉/比较时基

工作于捕捉或比较模式时，ECCP TMR1H:TMR1L 寄存器作为时基。

在捕捉模式下，TMR1H:TMR1L 这对寄存器的值在发生某个配置好的事件时被复制到 CCPR1H:CCPR1L 这对寄存器中。

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

更多信息请参见第 8.0 节“增强型捕捉/比较/PWM+（带自动关闭和死区）模块”。

## 7.2.10 ECCP 特殊事件触发器

当 ECCP 配置为触发特殊事件时，触发器会将 TMR1H:TMR1L 这对寄存器清零。该特殊事件不会产生 Timer1 中断。ECCP 模块仍可配置为产生 ECCP 中断。

在此工作模式下，CCPR1H:CCPR1L 这对寄存器成为了 Timer1 的周期寄存器。

Timer1 应同步为 FOSC 以充分利用特殊事件触发器。Timer1 异步工作可导致错过特殊事件触发器。

当对 TMR1H 或 TMR1L 的写操作与一个 ECCP 特殊事件触发器同时发生时，写操作具有优先权。

更多信息请参见第 8.4.5 节“特殊事件触发器”。

## 7.2.11 比较器同步

用于使 Timer1 递增的时钟还可用于同步比较器输出。这功能可在比较器模块中使能。

将比较器用于 Timer1 门控时，比较器输出应同步为 Timer1。这将在比较器发生变化时确保 Timer1 不错过递增。更多信息请参见第 10.9 节“比较器 C2 输出与 Timer1 同步”。

## 7.2.12 与 Timer1 相关的寄存器汇总

表 7-2-12: 与 TIMER1 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	---- --10
INTS	GIE	PEIE	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000	0000 0000
PIEB1	EEIE	ADIE	CCP1IE	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIFB1	EEIF	ADIF	CCP1IF	C2IF	C1IF	OSFIF	TMR2IF	TMRIF	0000 0000	0000 0000
TMR1H	16 位 TMR1 高字节的保持寄存器								xxxx xxxx	uuuu uuuu
TMR1L	16 位 TMR1 低字节的保持寄存器								xxxx xxxx	uuuu uuuu
T1STA	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	1111 1111

图注: x = 未知, u = 不变, — = 未实现 (读为 0)。Timer1 模块不使用阴影单元。

## 7.3 Timer2 模块

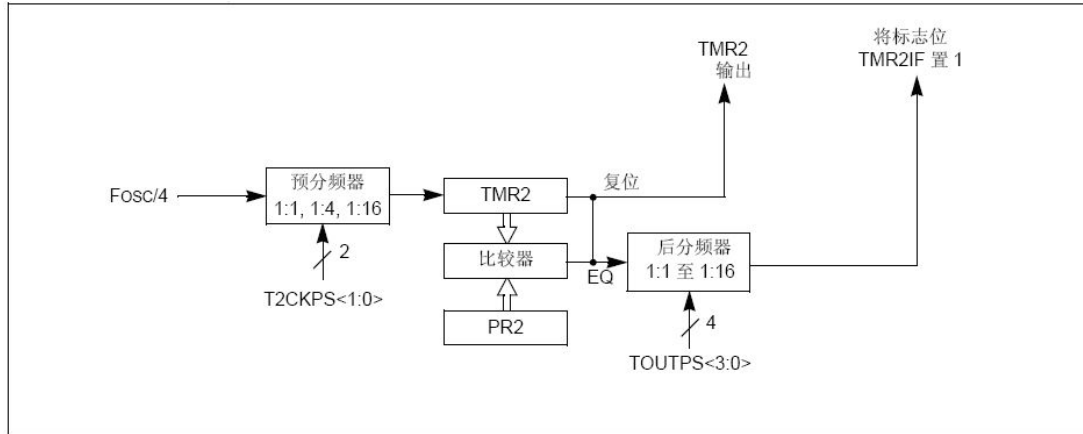
Timer2 模块是一个 8 位定时器，具备以下功能：

- 8 位定时器寄存器 (TMR2)
- 8 位周期寄存器 (PR2)
- TMR2 与 PR2 匹配时中断

- 软件可编程预分频比（1:1、1:4 或 1:16）
- 软件可编程后分频比（1:1 至 1:16）

Timer2 的框图请参见图 7-3。图

7-3: TIMER2 框图



### 7.3.1 Timer2 的工作原理

Timer2 模块的时钟输入是系统指令时钟（FOSC/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情况：

- TMR2 在下一递增周期复位为 00h。
- Timer2 后分频比递增

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

TMR2 和 PR2 寄存器均完全可读写。任何复位时，TMR2 寄存器置为 00h 且 PR2 寄存器置为 FFh。将 T2CON 寄存器中的 TMR2ON 位置 1 可打开 Timer2。将 TMR2ON 位清零可关闭 Timer2。

Timer2 预分频器由 T2CON 寄存器的 T2CKPS 位控制。Timer2 后分频器由 T2CON 寄存器的 TOUTPS 位控制。预分频器和后分频器计数器在以下情况下被清零：

- 发生了向 TMR2 的写入操作。
- 发生了向 T2CON 的写入操作。
- 发生了任一器件复位（上电复位，MCLR 复位，看门狗定时器复位或欠压复位）。

注：T2CON 被写入时 TMR2 不清零。

#### 寄存器 12H: TIMER2 控制寄存器 (T2CON)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit7							bit0



图注:

R = 可读位

-n = POR时的值

W = 可写位

1 = 置1

U = 未实现位, 读为0

0 = 清零

x = 未知

- bit7 未实现：读为0
- bit6-3 **TOUTPS<3:0>**: Timer2输出后分频比选择位  
 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
- bit2 **TMR2ON**: Timer2打开位  
 1 = Timer2打开  
 0 = Timer2关闭
- bit1-0 **T2CKPS<1:0>**: Timer2时钟预分频比选择位  
 00 = 预分频比为1  
 01 = 预分频比为4  
 1x = 预分频比为16

## 寄存器 11H: Timer2模块寄存器 (TMR2)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
<b>TMR2&lt;7&gt;</b>	<b>TMR2&lt;6&gt;</b>	<b>TMR2&lt;5&gt;</b>	<b>TMR2&lt;4&gt;</b>	<b>TMR2&lt;3&gt;</b>	<b>TMR2&lt;2&gt;</b>	<b>TMR2&lt;1&gt;</b>	<b>TMR2&lt;0&gt;</b>
bit7							bit0

图注:

R = 可读位                      W = 可写位                      U = 未实现位，读为0  
 -n = POR时的值                1 = 置1                              0 = 清零                              x = 未知

bit7-0 **TMR2<7:0>**: Timer2定时器寄存器

## 寄存器 92H: Timer2模块周期寄存器 (PR2)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
<b>PR2&lt;7&gt;</b>	<b>PR2&lt;6&gt;</b>	<b>PR2&lt;5&gt;</b>	<b>PR2&lt;4&gt;</b>	<b>PR2&lt;3&gt;</b>	<b>PR2&lt;2&gt;</b>	<b>PR2&lt;1&gt;</b>	<b>PR2&lt;0&gt;</b>
bit7							bit0

图注:

R = 可读位                      W = 可写位                      U = 未实现位，读为0  
 -n = POR时的值                1 = 置1                              0 = 清零                              x = 未知

bit7-0 **PR2<7:0>**: Timer2模块周期寄存器

表 7-3-4: 与 Timer2 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
INTS	GIE	PEIE	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000	0000 0000
PIEB1	EEIE	ADIE	CCP1IE	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIFB1	EEIF	ADIF	CCP1IF	C2IF	C1IF	OSFIF	TMR2IF	TMRIF	0000 0000	0000 0000
PR2	Timer2 模块周期寄存器								1111 1111	1111 1111
TMR2	Timer2 模块寄存器								0000 0000	0000 0000
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000

图注: x = 未知, u = 不变, — = 未实现 (读为 0)。Timer2 模块不使用阴影单元。

## 8.0 增强型捕捉/比较/PWM+ (带自动关闭和死区) 模块

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

### 8.1 表: ECCP 模式——所需的定时器资源

ECCP模式	定时器资源
捕捉	Timer1
比较	Timer1
PWM	Timer2

### 8.2 寄存器: CCP1CON: 增强型 CCP1 控制寄存器

寄存器 15H: 增强型 CCP1 控制寄存器 (CCP1CON)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit7-6

**P1M<1:0>**: PWM 输出配置位 如果

CCP1M<3:2> = 00、01 或 10:

xx = P1A 配置为捕捉/比较输入; P1B、P1C 和 P1D 配置为端口引脚

如果 CCP1M<3:2> = 11:

00 = 单输出; P1A 调制; P1B、P1C 和 P1D 配置为端口引脚

01 = 全桥正向输出; P1D 调制; P1A 有效; P1B 和 P1C 无效

10 = 半桥输出; P1A 和 P1B 调制, 带有死区控制; P1C 和 P1D 配置为端口引脚

11 = 全桥反向输出；P1B调制；P1C有效；P1A和P1D无效

bit5-4 **DC1B<1:0>**: PWM占空比最低有效位 捕捉模式：  
未使用。  
比较模式：未使用。 PWM模式：  
这些位是PWM占空比的低2位。高8位在CCPR1L中。

bit3-0 **CCP1M<3:0>**: ECCP模式选择位  
0000 = 捕捉/比较/PWM关闭（复位ECCP模块）  
0001 = 未使用（保留）  
0010 = 比较模式，匹配时翻转输出（CCP1IF位置1）  
0011 = 未使用（保留）  
0100 = 捕捉模式，每个下降沿  
0101 = 捕捉模式，每个上升沿  
0110 = 捕捉模式，每4个上升沿  
0111 = 捕捉模式，每16个上升沿  
1000 = 比较模式，匹配时输出置1（CCP1IF位置1）  
1001 = 比较模式，匹配时输出清零（CCP1IF位置1）  
1010 = 比较模式，匹配时产生软件中断（CCP1IF位置1，CCP1引脚不受影响）  
1011 = 比较模式，触发特殊事件（CCP1IF位置1，CCP1复位TMR1，且如果A/D模块被使能，启动一次A/D转换）  
1100 = PWM模式；P1A和P1C高电平有效；P1B和P1D高电平有效  
1101 = PWM模式；P1A和P1C高电平有效；P1B和P1D低电平有效  
1110 = PWM模式；P1A和P1C低电平有效；P1B和P1D高电平有效  
1111 = PWM模式；P1A和P1C低电平有效；P1B和P1D低电平有效

### 8.3 捕捉模式

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

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

进行捕捉后，PIFB1 寄存器中的中断请求标志位 CCP1IF 被置 1。该位必须用软件清零。如果在 CCPR1H 和 CCPR1L 这对寄存器中的值被读出之前又发生另一次捕捉，那么原来的捕捉值会被新捕捉值覆盖（见 [图 8-3-1](#)）。

寄存器 13H: 捕捉/比较/PWM寄存器1的低字节（CCPR1L）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CCPR1<7	CCPR1<6	CCPR1<5	CCPR1<4	CCPR1<3	CCPR1<2	CCPR1<1	CCPR1<0
>	>	>	>	>	>	>	>
bit7							bit0

图注:

R = 可读位

-n = POR时的值

W = 可写位

1 = 置1

U = 未实现位, 读为0

0 = 清零

x = 未知

bit7-0 **CCPR1<7:0>**: 捕捉/比较/PWM寄存器1的低字节

寄存器 14H: 捕捉/比较/PWM寄存器1的高字节 (**CCPR1H**)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
<b>CCPR1&lt;15</b>	<b>CCPR1&lt;14</b>	<b>CCPR1&lt;13</b>	<b>CCPR1&lt;12</b>	<b>CCPR1&lt;11</b>	<b>CCPR1&lt;10</b>	<b>CCPR1&lt;9</b>	<b>CCPR1&lt;8</b>
>	>	>	>	>	>	>	>
bit7							bit0

图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为0  
 -n = POR时的值                1 = 置1                              0 = 清零                              x = 未知

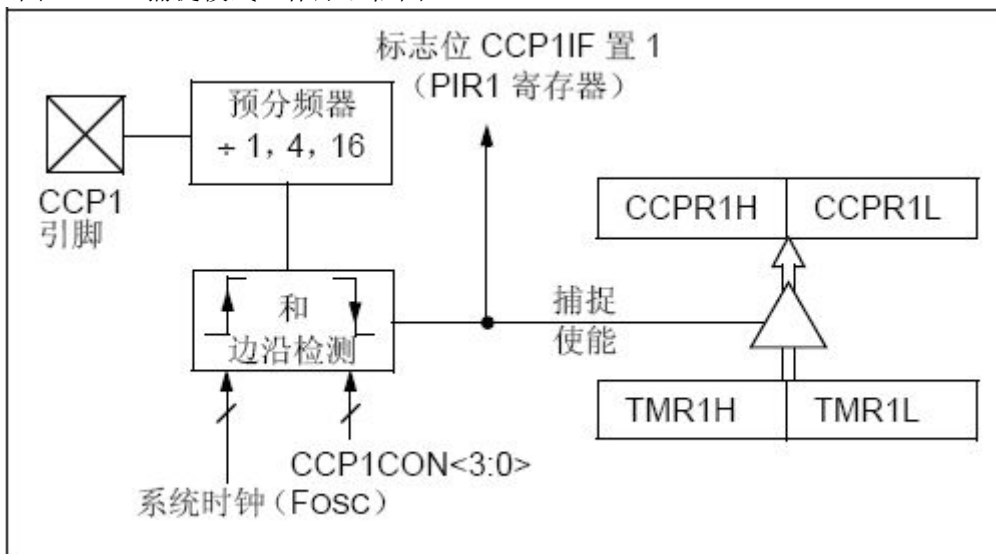
bit7-0 **CCPR1<15:8>**: 捕捉/比较/PWM寄存器1的高字节

### 8.3.1 CCP1 引脚配置

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

注: 如果CCP1引脚配置为输出, 则写端口将产生一次捕捉条件。

图 8-3-1: 捕捉模式工作原理框图



### 8.3.2 Timer1 模式选择

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

### 8.3.3 软件中断

当捕捉模式改变时，可能会产生一次误捕捉中断。用户应该保持 PIEB1 寄存器的 CCP1IE 位清零以避免误中断。此外，用户还应在任何这种工作模式改变之后清零 PIFB1 寄存器的中断标志位 CCP1IF。

### 8.3.4 CCP 预分频器

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

从一个捕捉预分频比切换到另一个不会清零预分频器并可能产生一次中断。要避免意外操作，在改变预分频比前应将 CCP1CON 寄存器清零，以关闭 CCP 模块（见例 8-3-4）。

例 8-3-4：在捕捉预分频比间切换

```

CLRR  CCP1CON      ;Turn ECCP module off
LDWI  NEW_CAPT_PS  ;Load the W reg with
                          ;the new prescaler
                          ;move value and ECCP ON
STWR  CCP1CON      ;Load CCP1CON with this
                          ;value

```

## 8.4 比较模式

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

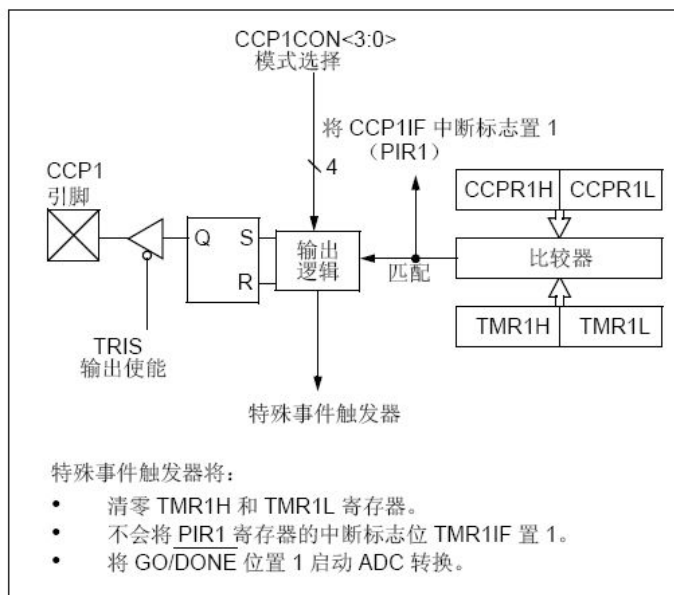
- 翻转 CCP1 输出
- 将 CCP1 输出置 1
- 将 CCP1 输出清零
- 触发特殊事件
- 产生软件中断

引脚上的动作取决于 CCP1CON 寄存器的 CCP1M<3:0>控制位的值。

所有比较模式均可产生中断。

### 8.4.1 比较模式的工作框图

图 8-4-1：比较模式的工作框图



## 8.4.2 CCP1 引脚的配置

用户必须通过清零相关的 CPIO 位将 CCP1 引脚配置为输出。

注：清零 CCP1CON 寄存器将强制 CCP1 比较输出锁存为缺省低电平。这不是 PORTC 的 I/O 数据锁存。

## 8.4.3 Timer1 模式选择

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

## 8.4.4 软件中断模式

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

## 8.4.5 特殊事件触发器

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

- 复位 Timer1
- 若 ADC 使能，则启动一次 ADC 转换

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

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



CCPR1H:CCPR1L 寄存器可作为 Timer1 的 16 位可编程周期寄存器。

- 注：
- 1: CCP模块的特殊事件触发不会将PIFB1寄存器的中断标志位TMR1IF置1。
  - 2: 在生成特殊事件触发的时钟边沿和复位Timer1的时钟边沿之间，通过更改CCPR1H和CCPR1L这对寄存器的内容来移除匹配条件，可以预防复位的发生。

## 8.5 PWM 模式

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

- PR2
- T2CON
- CCPR1L
- CCP1CON

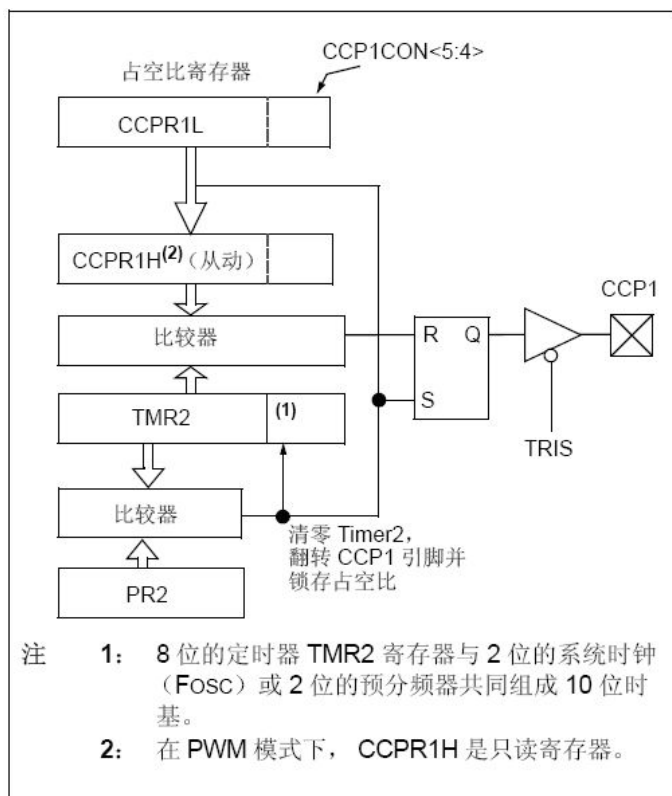
在脉宽调制（Pulse-Width Modulation, PWM）模式下，CCP 模块在 CCP1 引脚上产生高达 10 位分辨率的 PWM 输出。由于 CCP1 引脚与 PORT 数据锁存器复用，必须将该引脚的 CPIO 清零以使能 CCP1 引脚输出驱动器。

注： 将CCP1CON寄存器清零将放弃对CCP1引脚的控制。

### 8.5.1 图：简化的 PWM 框图

欲了解将 CCP 模块设置为 PWM 操作的详细步骤，请参见[第 8.5.9 节“PWM 工作的设置”](#)。

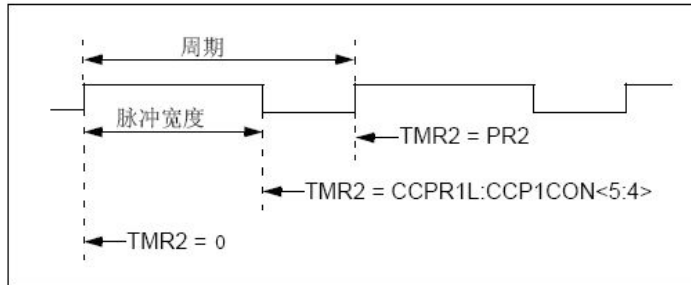
图 8-5-1: 简化的 PWM 框图



## 8.5.2 CCP PWM 输出

PWM 输出（图 8-5-2）由时基（周期）和一段输出保持为高电平（占空比）的时间组成。

图 8-5-2: CCP PWM 输出



## 8.5.3 PWM 周期

PWM 周期由 Timer2 的 PR2 寄存器指定。用公式 8-5-3 可计算 PWM 周期。

公式 8-5-3: PWM 周期

$$PWM \text{ 周期} = [(PR2) + 1] \cdot 4 \cdot T_{osc} \cdot (TMR2 \text{ 预分频值})$$

当 TMR2 等于 PR2 时，下一次递增周期将发生以下三个事件：

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

注：确定 PWM 频率时不使用 Timer2 寄存器（见第 7.3.1 节“Timer2 的工作原理”）。

## 8.5.4 PWM 占空比

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

公式 8-5-4-1 用于计算 PWM 脉宽。

公式 8-5-4-2 用于计算 PWM 占空比。

公式 8-5-4-1: 脉冲宽度

$$脉冲宽度 = (CCPR1L:CCP1CON<5:4>) \cdot T_{osc} \cdot (TMR2 \text{ 预分频值})$$

公式 8-5-4-2: 占空比

$$\text{占空比} = \frac{(\text{CCPR1L}:\text{CCP1CON}\langle 5:4 \rangle)}{4(\text{PR2} + 1)}$$

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

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

当 10 位时基与 CCPR1H 及 2 位的锁存器匹配时, CCP1 引脚被清零 (见图 8-5-1)。

## 8.5.5 PWM 分辨率

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

PR2 为 255 时产生 10 位的最大 PWM 分辨率。分辨率是 PR2 寄存器值的函数, 如公式 8-5-5 所示。

公式 8-5-5: PWM 分辨率

$$\text{分辨率} = \frac{\log[4(\text{PR2} + 1)]}{\log(2)} \text{ 位}$$

注: 如果脉宽大于周期, 那么分配的 PWM 引脚将保持不变。

表 8-5-5-1: PWM 频率和分辨率示例 (Fosc = 20MHz)

PWM 频率	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
定时器预分频比 (1、4 或 16)	16	4	1	1	1	1
PR2 的值	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最大分辨率 (位)	10	10	10	8	7	6.6

表 8-5-5-2: PWM 频率和分辨率示例 (Fosc = 8MHz)

PWM 频率	1.22 kHz	4.9 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
定时器预分频比 (1、4 或 16)	16	4	1	1	1	1
PR2 的值	0x65	0x65	0x65	0x19	0x0C	0x09
最大分辨率 (位)	8	8	8	6	5	5

## 8.5.6 休眠模式下的工作

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

## 8.5.7 系统时钟频率的改变

PWM 频率来自系统时钟频率。系统时钟频率的任何改变将导致 PWM 频率的改变。详情请参见[第 1.1 节“系统结构图”](#)。

## 8.5.8 复位的影响

任何复位均将强制所有端口为输入模式，并强制 CCP 寄存器为其复位状态。

## 8.5.9 PWM 工作的设置

应按照以下步骤将 CCP 模块配置为 PWM 工作：

1. 将相关的 CPIO 位置 1 禁止 PWM 引脚（CCP1）的输出驱动器。
2. 装载 PR2 寄存器以设置 PWM 周期。
3. 用适当的值装载 CCP1CON 寄存器将 CCP 模块配置为 PWM 模式。
4. 装载 CCPR1L 寄存器和 CCP1CON 寄存器的 DC1B<1:0>设置 PWM 占空比。
5. 配置并启动 Timer2:
  - 将 PIFB1 寄存器的 TMR2IF 中断标志位清零。
  - 装载 T2CON 寄存器的 T2CKPS 位设置 Timer2 预分频比。
  - 将 T2CON 寄存器的 TMR2ON 位置 1 使能 Timer2。
6. 重新开始一个 PWM 周期后，使能 PWM 输出：
  - 等待 Timer2 溢出（PIR1 寄存器的 TMR2IF 位置 1）。
  - 将相关的 CPIO 位清零使能 CCP1 引脚的输出驱动器。

## 8.6 PWM（增强模式）

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

- 单 PWM
- 半桥 PWM
- 全桥 PWM，正向模式

- 全桥 PWM，反向模式

要选择增强型 PWM 模式，CCP1CON 寄存器的 P1M 位必须被正确置 1。

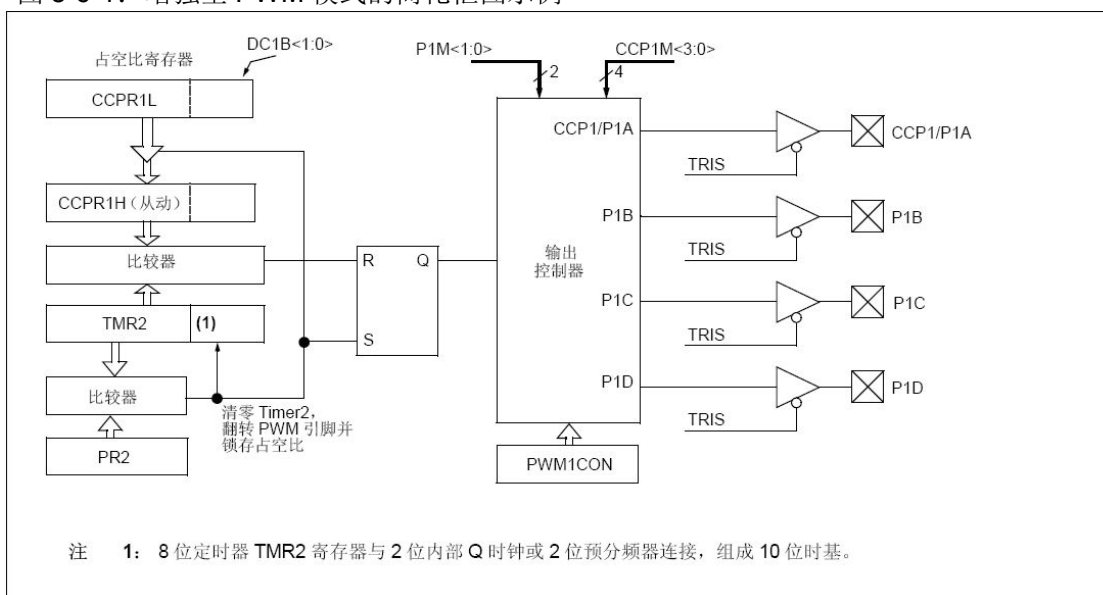
PWM 输出与 I/O 引脚复用，并被指定为 P1A、P1B、P1C 和 P1D。PWM 引脚的极性可配置，可通过将 CCP1CON 寄存器中的 CCP1M 位进行适当置 1 选择极性。

表 8-6-1 所示为每个增强型 PWM 模式的引脚分配。

图 8-6-1 所示为增强型 PWM 模块的简化框图。

注：一开始使能 PWM 时，要防止产生不完整的波形，ECCP 模块在产生 PWM 信号前应先等待新的 PWM 周期开始。

图 8-6-1：增强型 PWM 模式的简化框图示例



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

图 8-6-2：PWM（增强模式）输出关系示例（高电平有效状态）

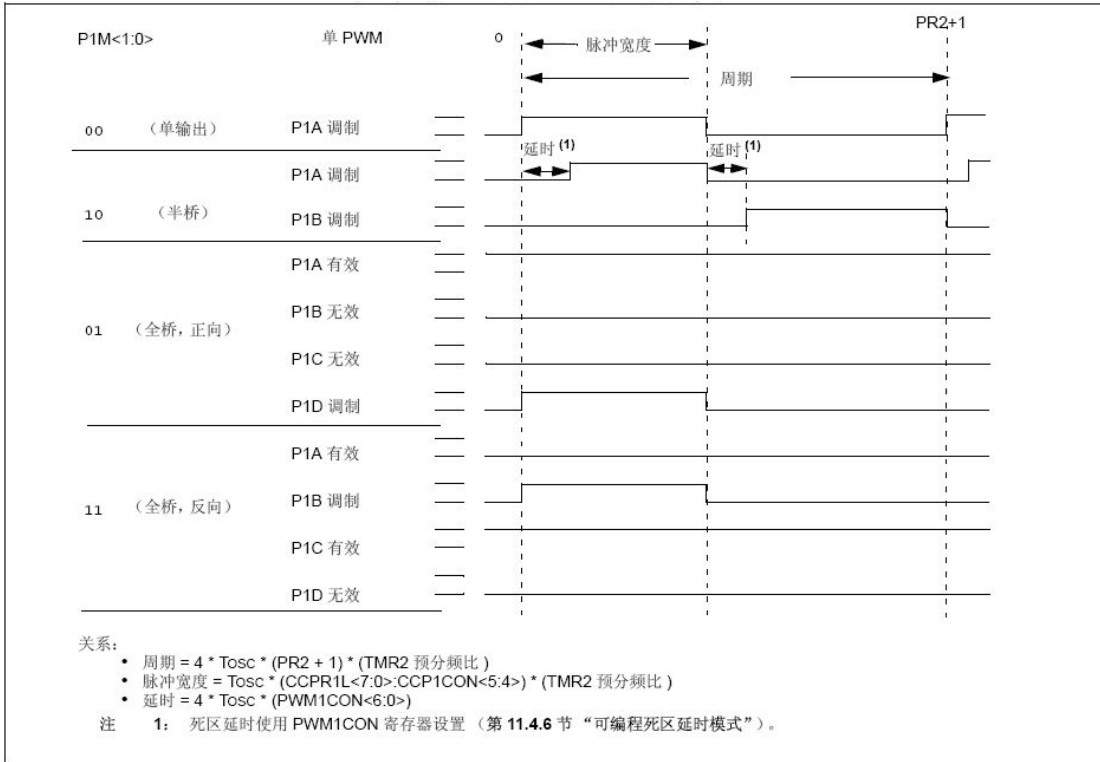


图 8-6-3: PWM (增强模式) 输出关系示例 (低电平有效状态)

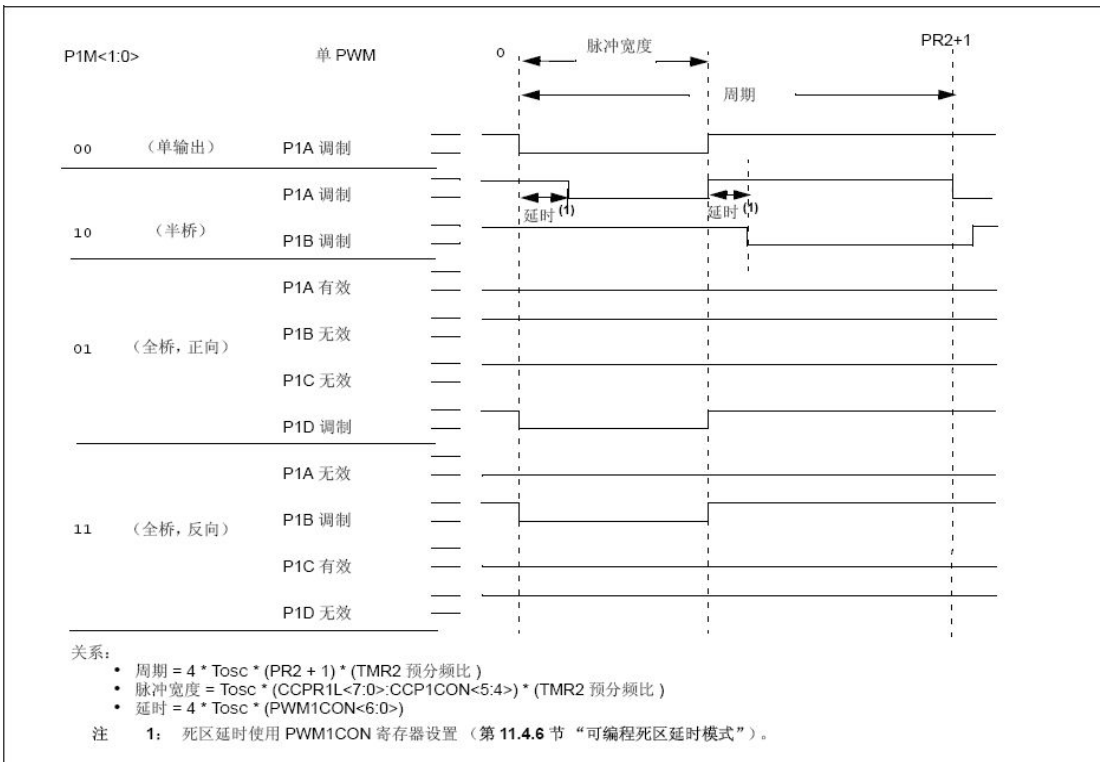


表 8-6-1: 不同 PWM 增强模式的引脚分配示例

ECCP 模式	P1M<1:0>	CCP1/P1A	P1B	P1C	P1D
单 PWM	00	是	是	是	是
半桥 PWM	10	是	是	否	否
全桥, 正向	01	是	是	是	是
全桥, 反向	11	是	是	是	是

## 8.6.1 半桥模式

在半桥模式下，有两个引脚用作输出以驱动推挽负载。PWM 输出信号被输出到 CCP1/P1A 引脚，而互补 PWM 输出信号被输出到 P1B 引脚（见图 8-6-2）。此模式可用于半桥应用，如图 8-6-1-2 所示，或用于全桥应用，此时用两个 PWM 信号调制四个功率开关。

在半桥模式下，可使用编程死区延时防止半桥功率器件中出现穿通电流。PWM1CON 寄存器的 PDC<6:0> 位用于设置将输出驱动为有效前的指令数。如果该值大于占空比，则相应的输出在整个周期中将保持无效状态。死区延时操作的详情请参见第 8.6.6 节“可编程死区延时模式”。

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

图 8-6-1-1: 半桥 PWM 输出示例

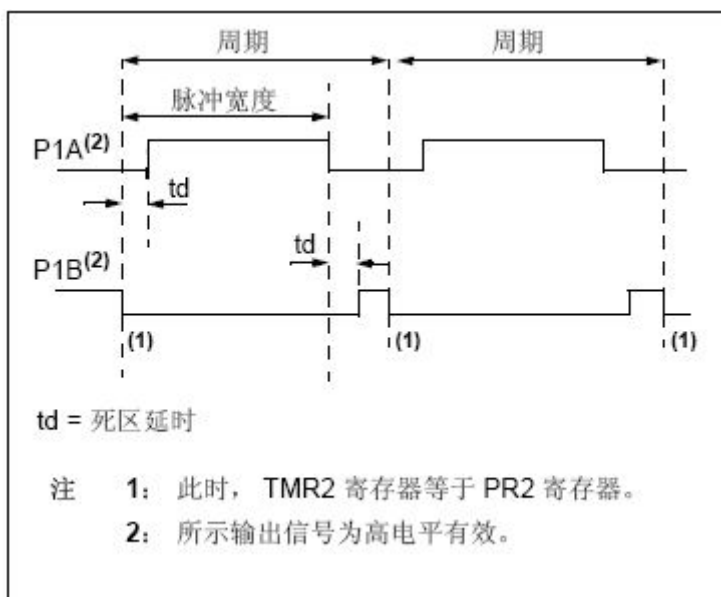
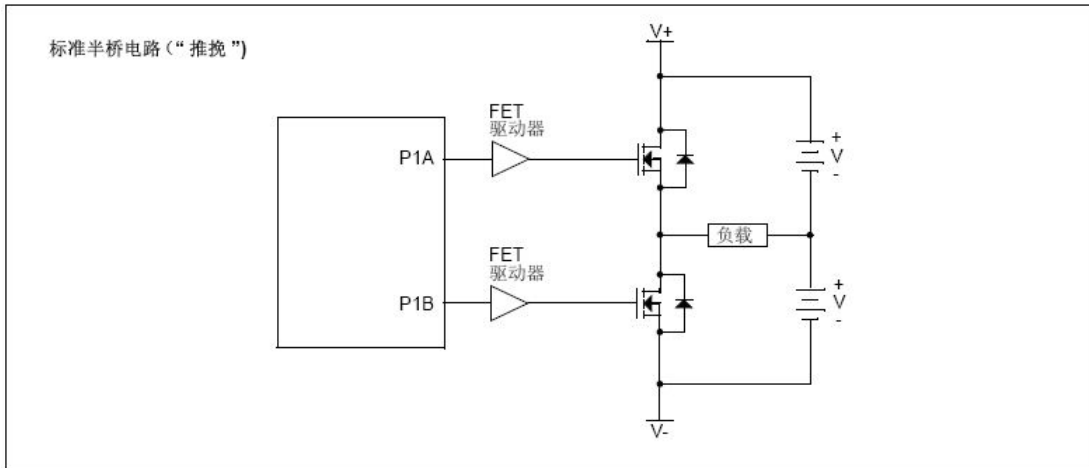


图 8-6-1-2: 半桥应用示例





## 8.6.2 全桥模式

在全桥模式下，所有四个引脚均用作输出。图 8-6-2-1 所示为一个全桥应用示例。在正向模式下，

CCP1/P1A 引脚驱动为有效状态，P1D 引脚为调制输出，而 P1B 和 P1C 则被驱动为无效状态，如图 8-6-2-2 所示。

在反向模式下，P1C 驱动为有效状态，P1B 引脚为调制输出，而 P1A 和 P1D 则被驱动为无效状态，如图 8-6-2-2 所示。

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

图 8-6-2-1: 全桥应用示例

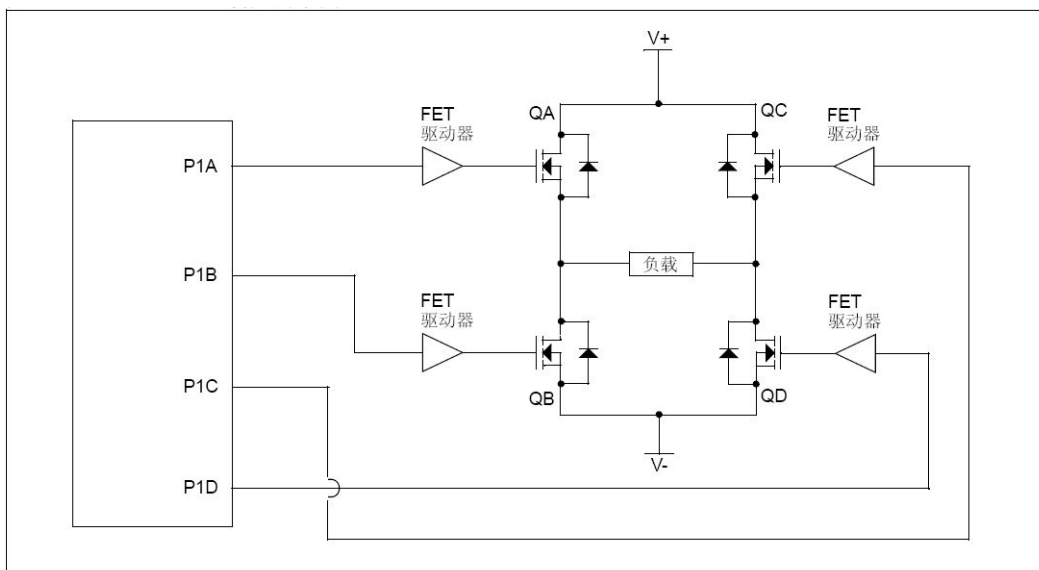
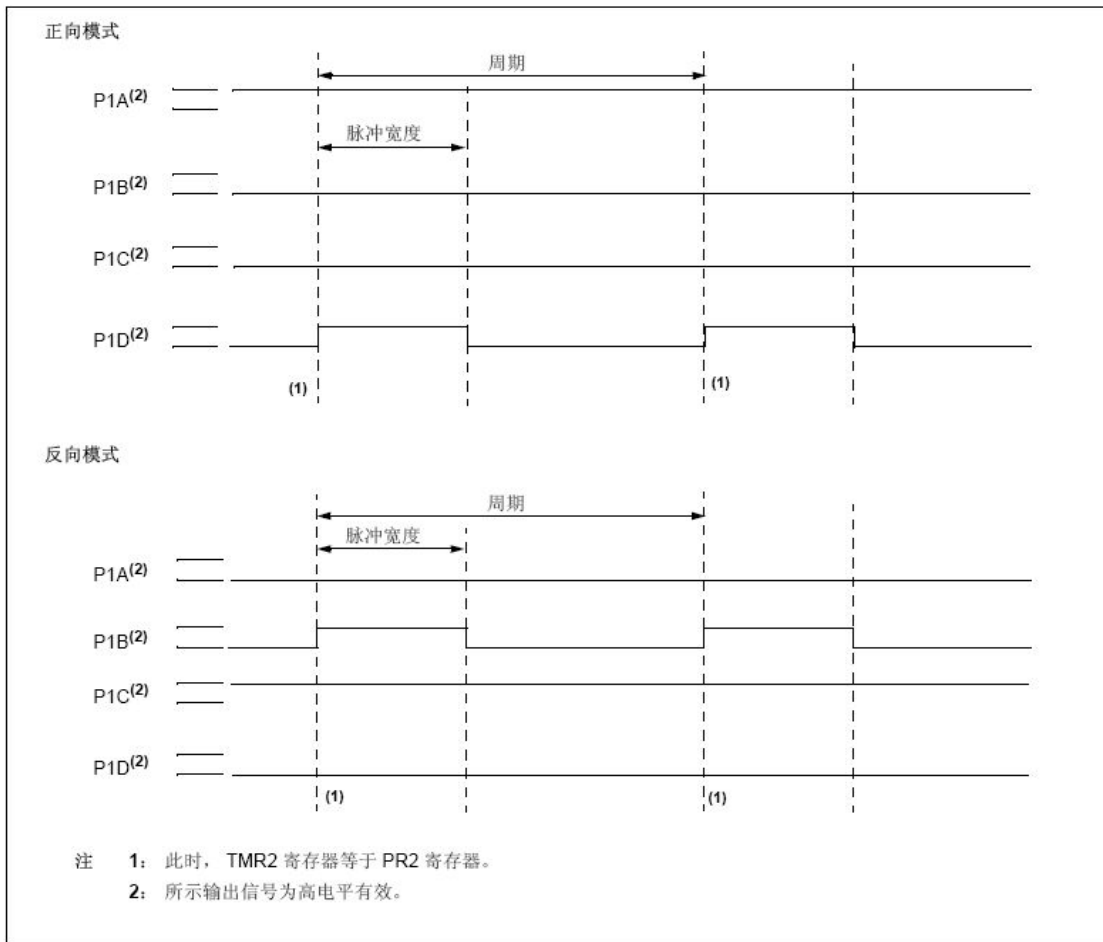


图 8-6-2-2: 全桥 PWM 输出示例



### 8.6.3 启动考虑事项

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

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

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

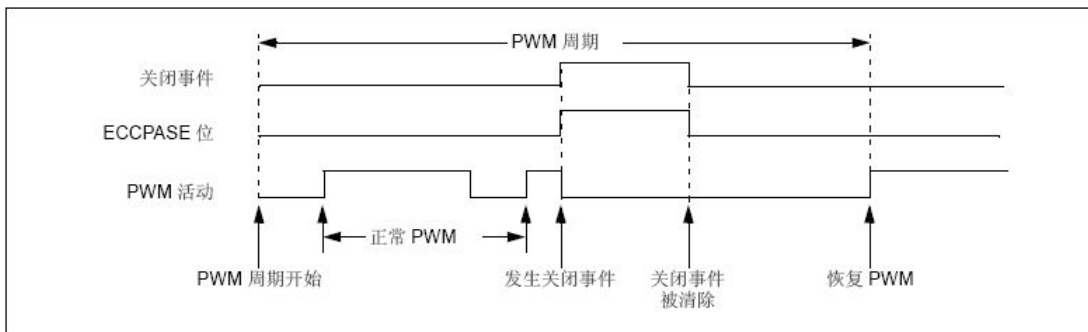
在 PWM 模块初始化时, P1A、P1B、P1C 和 P1D 输出锁存器可能不在正确的状态下。将 PWM 引脚的输出驱动器与增强型 PWM 模式同时使能可能导致应用电路的损坏。增强型 PWM 模式必须在正确的输出模式下使能、并且在 PWM 引脚的输出驱动器被使能前完成一整个 PWM 周期。一整个 PWM 周期是否完成可通过查看 PIR1 寄存器的 TMR2IF 位在第二个 PWM 周期开始时是否置 1 查看。

### 8.6.5 自动重启模式

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

使能自动重启时，只要自动关闭条件有效，ECCPASE 位就保持置 1。当自动关闭条件被清除时，ECCPASE 位将被硬件清零，恢复正常工作。

图 8-6-5: 自动重启使能时的 PWM 自动关闭 (PRSEN=1)



### 8.6.6 可编程死区延时模式

在所有功率开关均调制为 PWM 频率的半桥应用中，功率开关从关断到导通通常需要较长的时间。如果上下两个功率开关同时动作（一个导通另一个关断），在一个开关完全关断前，两个开关可能在一个很短的时间内

同时导通。在这段很短的时间内，在两个功率开关中会流过极高的电流（穿通（shoot-through）电流），使桥路的电源短路。为避免在开关时出现这种极具破坏力的穿通电流，通常使任一功率开关的导通时间延后，以使另一个开关有时间完全关断。

在半桥模式下，使用数字可编程死区延时来避免穿通电流破坏桥路的功率开关。信号从无效状态变为有效状态时发生延时。请参见图 8-6-1-2 中的图解说明。

## 9.0 模拟数字转换器（ADC）模块

模数转换器（Analog-to-digital Converter, ADC）可将模拟输入信号转换为相应的 10 位二进制表征值。该系列器件采用多个模拟输入复用到一个采样保持电路。采样保持电路的输出与转换器的输入相连接。转换器通过逐次逼近法产生 10 位二进制值，并将转换结果保存在 ADC 结果寄存器（ADRESL:ADRESH）中。

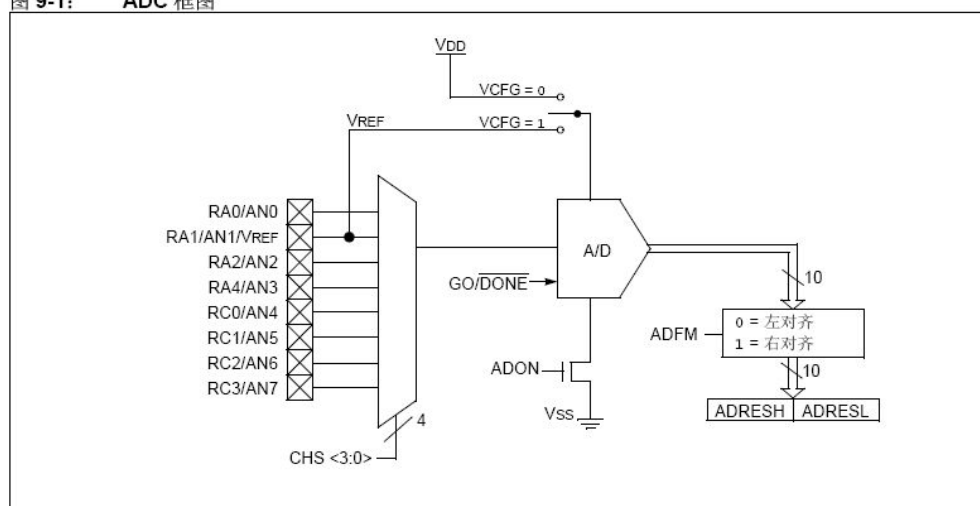
ADC 参考电压可用软件选择为 VDD 或施加在外部参考引脚上的电压。

ADC 可在转换完成时产生中断。该中断可用于将器件从休眠唤醒。

### 9.1 图：ADC 框图

图 9-1：ADC 框图

图 9-1：ADC 框图



### 9.2 ADC 的配置

配置和使用 ADC 时，必须考虑以下功能：

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

- 转换结果的格式

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

## 9.2.2 通道选择

ADS0 寄存器的 CHS 位决定将哪个通道连接到采样保持电路。 改变通道时，开始下一次转换前需要一个延时。更多信息请参见[第 9.3 节“ADC 的工作原理”](#)。

## 9.2.3 ADC 参考电压

ADS0 寄存器的 ADVRS<1:0>位提供对正参考电压的控制。

正参考电压可以是：

- 选电压电压 VDD 做为 ADC 的参考。
- 选外部（来自 PA1）做为 ADC 的参考。
- 选内部 BANDGAP 的输出电压做为 ADC 的参考，这个电压可以通过配置寄存器选择 2V，3V，4V。这个电压也可以通过 ADS1 中的最高位置 1 把参考电压送到 PA1 上，PA1 需要设成模拟输入(ANS1 置 1)。

负参考电压始终连接到参考地。

## 9.2.4 转换时钟

转换时钟源可通过 ADS1 寄存器的 ADCS 位用软件选择。有以下七种时钟选项：

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- F<sub>SM</sub>（专用内部振荡器）

完成一位（bit）的转换时间定义为 T<sub>AD</sub>。完成 10 位转换需要 11 个 T<sub>AD</sub> 周期，如[图 9-2-4](#) 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 13.0 节“电气特性”中的 A/D 转换要求。表 9-2-4 所示为正确选择 ADC 时钟的示例。

注：除非使用的是  $F_{SM}$ ，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。

图 9-2-4：模数转换  $T_{AD}$  周期

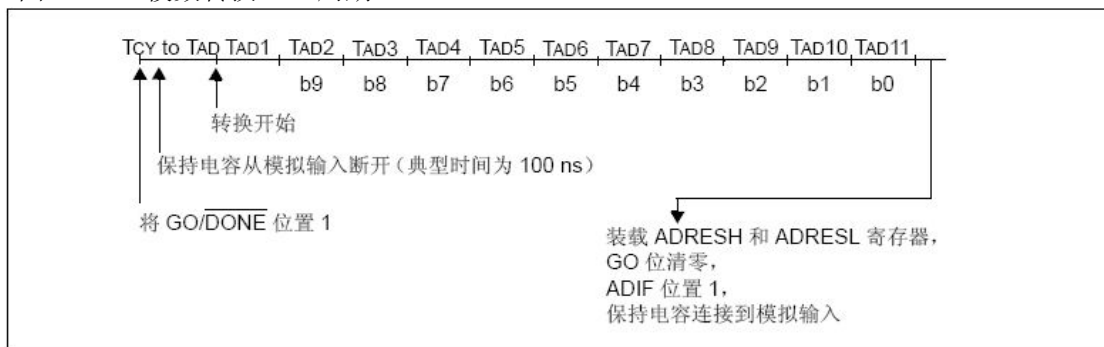


表 9-2-4：ADC 时钟周期 ( $T_{AD}$ ) — 器件工作频率 ( $V_{DD} \geq 3.0V$ )

ADC 时钟周期 ( $T_{AD}$ )		器件频率 ( $F_{Osc}$ )			
ADC 时钟源	ADCS<2:0>	20 MHz	8 MHz	4 MHz	1 MHz
$F_{Osc}/2$	000	100 ns <sup>(2)</sup>	250 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	2.0 us
$F_{Osc}/4$	100	200 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	1.0 us <sup>(2)</sup>	4.0 us
$F_{Osc}/8$	001	400 ns <sup>(2)</sup>	1.0 us <sup>(2)</sup>	2.0 us	8.0 us <sup>(3)</sup>
$F_{Osc}/16$	101	800 ns <sup>(2)</sup>	2.0 us	4.0 us	16.0 us <sup>(3)</sup>
$F_{Osc}/32$	010	1.6 us	4.0 us	8.0 us <sup>(3)</sup>	32.0 us <sup>(3)</sup>
$F_{Osc}/64$	110	3.2 us	8.0 us <sup>(3)</sup>	16.0 us <sup>(3)</sup>	64.0 us <sup>(3)</sup>
$F_{SM}$	x11	2-6 us <sup>(1,4)</sup>	2-6 us <sup>(1,4)</sup>	2-6 us <sup>(1,4)</sup>	2-6 us <sup>(1,4)</sup>

图注：阴影单元超出了建议范围。

- 注
- 1: 在  $V_{DD} > 3.0V$  的情况下， $F_{SM}$  时钟源的典型  $T_{AD}$  时间为 4 us。
  - 2: 这些值违反了最小  $T_{AD}$  时间要求。
  - 3: 要加快转换，建议选择其他时钟源。
  - 4: 当器件频率大于 1 MHz 时，只有在休眠中进行转换才建议使用  $F_{SM}$  时钟源。

## 9.2.5 ADC 模块中断

ADC 模块可使中断在模数转换完成时产生。ADC 中断标志为 PIR1 寄存器中的 ADIF 位。ADC 中断使能为 PIEB1 寄存器中的 ADIE 位。ADIF 位必须用软件清零。

注：无论 ADC 中断是否被允许，ADIF 位在每次转换完成时均置 1。

ADC 无  $F_{SM}$  模式，需要使用 ADC 唤醒时，系统时钟必须是内部快/慢时钟之一。

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

## 9.2.6 A/D 转换结果格式

10 位 A/D 转换结果有两种格式，即左对齐和右对齐。ADS0 寄存器的 ADFM 位控制输出格式。

图 9-2-6-1 所示为两种输出格式。

图 9-2-6-1: 模数转换 T<sub>AD</sub> 周期

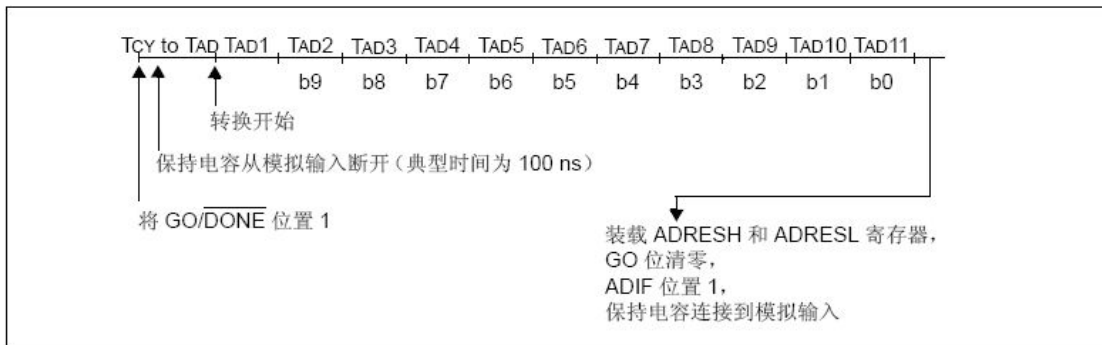
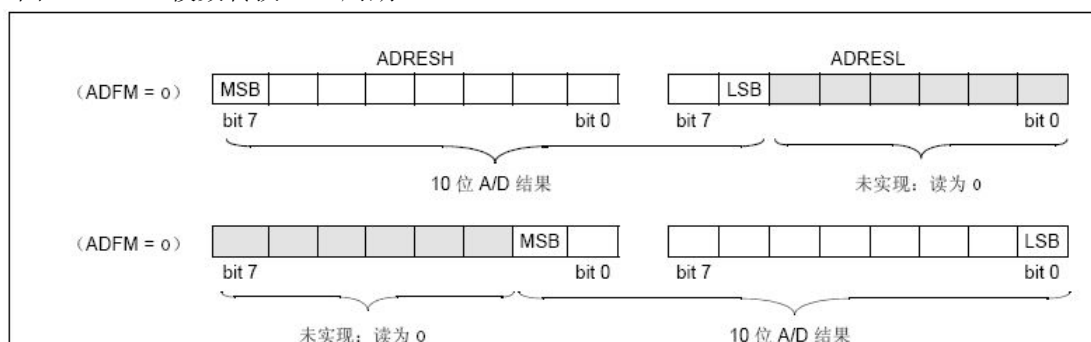


图 9-2-6-2: 模数转换 T<sub>AD</sub> 周期



## 9.3 ADC 的工作原理

### 9.3.1 启动转换

要启用 ADC 模块，必须将 ADS0 寄存器的 ADON 位置 1。将 ADS0 寄存器的 GO/DONE 位置 1 将启动模数转换。

注：应在打开 ADC 的那条指令中将 GO/DONE 位置 1。请参见第 9.3.5 节“A/D 转换步骤”。

### 9.3.2 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



### 9.3.3 终止转换

如果转换必须在完成前被终止，可用软件将  $\overline{GO/DONE}$  清零。ADRESH:ADRESL 寄存器不会被未完成的模数转换采样更新。相反，ADRESH:ADRESL 这对寄存器将保持先前转换的值。此外，启动另一次采集前，需等待  $2 T_{AD}$  的延时。延时后，所选通道的输入采集将自动启动。

注： 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。

### 9.3.4 特殊事件触发器

ECCP 特殊事件触发器可在软件不干预的情况下周期性地进行 ADC 测量。发生触发事件时， $\overline{GO/DONE}$  位由硬件置 1，Timer1 计数器复位为零。

特殊事件触发器的使用并不确保正常 ADC 定时。确保满足 ADC 定时要求是用户的责任。

更多信息请参见[第 8.0 节“增强型捕捉/比较/PWM+（带自动关闭和死区）模块”](#)。

### 9.3.5 A/D 转换步骤

以下是使用 ADC 进行模数转换的步骤示例：

1. 配置端口：
  - 禁止引脚输出驱动器（见 CPIO 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<sup>(1)</sup>
4. 等待所需的采集时间<sup>(2)</sup>
5. 将  $\overline{GO/DONE}$  置 1 启动转换
6. 通过以下情况之一等待 ADC 转换完成：
  - 查询  $\overline{GO/DONE}$  位
  - 等待 ADC 中断（允许中断时）
7. 读取 ADC 结果
8. 将 ADC 中断标志清零（必允许中断）。

注： 1：如果用户试图将器件从休眠中唤醒并顺序执行代码，则可禁止全局中断。

2：请参见[第9.5节“A/D采集时间要求”](#)。

例 9-3-5: A/D 转换

```

This code block configures the A/D
;for polling, Vdd reference, R/C clock
;and RA0 input.
;
;Conversion start & wait for complete
;polling code included.
;
BSR    STATUS,PAGE    ;Bank 1
LDWI   B'01110000'    ;A/D RC clock
STWR   ADS1
BSR    CPIOA,0        ;Set PA0 to input
BSR    ANSEL,0        ;Set PA0 to analog
BCR    STATUS,PAGE    ;Bank 0
LDWI   B'10000001'    ;Right, Vdd Vref, AN0
STWR   ADS0
LCALL  SampleTime     ;Wait min sample time
BSR    ADS0,GO        ;Start conversion
BTSC   ADS0,GO        ;Is conversion done?
LJUMP  $-1            ;No, test again
LDR    ADRESH,W       ;Read upper 2 bits
STWR   RESULTHI
BSR    STATUS,PAGE    ;Bank 1
LDR    ADRESL,W       ;Read lower 8 bits
STWR   RESULTLO

```

## 9.4 ADC 寄存器定义

以下寄存器用于控制 ADC 的工作。

### 9.4.1 ADS0: A/D 控制寄存器 0

寄存器 1FH: A/D控制寄存器0 (ADS0)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>ADFM</b>	<b>ADVRS1</b>	<b>ADVRS0</b>	<b>CHS2</b>	<b>CHS1</b>	<b>CHS0</b>	<b>GO/DONE</b>	<b>ADON</b>
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7 **ADFM**: A/D转换结果格式选择位

1 = 右对齐

0 = 左对齐

bit6-5 **ADVRS<6:5>**: 选择ADC的参考电压

00 = 选电压VDD做为ADC的参考

10 = 选外部(来自PA1)做为ADC的参考

X1 = 选内部BANDGAP的输出电压做为ADC的参考, 这个电压可以通过配置寄存器选择2V, 3V, 4V. 这个电压也可以通过ADS1中的最高位ADRVO置1, PA1需要设成模拟输入(ANS1置1), 把参考电压送到PA1上。

- bit4-2     **CHS<2:0>**: 模拟通道选择位  
 000 = AN0  
 001 = AN1  
 010 = AN2  
 011 = AN3  
 100 = AN4  
 101 = AN5  
 110 = AN6  
 111 = AN7
- bit1     **GO/DONE**: A/D转换状态位  
 1 = A/D转换正在进行。将本位置1启动一次A/D转换。  
       A/D转换完成时此位由硬件自动清零。  
 0 = A/D转换完成/不在进行中
- bit0     **ADON**: ADC使能位  
 1 = 使能ADC  
 0 = 禁止ADC, 不消耗工作电流

## 9.4.2 ADS1: A/D 控制寄存器 1

寄存器 9FH: A/D控制寄存器1 (ADS1)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
<b>ADRVO</b>	<b>ADCS2</b>	<b>ADCS1</b>	<b>ADCS0</b>	—	—	—	—
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

- bit7     **ADRVO**: 控制ADC的内部BANDGAP输出电压作为参考电压是否送到PA1端口上  
 1 = 送到PA1端口上  
 0 = 不送到PA1端口上。
- bit6-4   **ADCS<2:0>**: A/D转换时钟选择位  
 000 = FOSC/2  
 001 = FOSC/8  
 010 = FOSC/32  
 x11 = F<sub>SM</sub>智能分频, ADC的转换时钟会根据振荡模式来选择分频比:  
       当系统时钟选择LP或者RC或者RCIO模式时, ADC的转换时钟就是系统时钟的2分频。  
       当系统时钟选择XT模式时, ADC的转换时钟就是系统时钟的8分频。  
       当系统时钟选择HF模式时, ADC的转换时钟就是系统时钟的32分频。

100 = FOSC/4  
 101 = FOSC/16  
 110 = FOSC/64

bit3-0 未实现：读为0

### 9.4.3 ADRESH: ADC 结果高位寄存器

寄存器 1EH: ADC结果高位寄存器 (ADRESH), ADFM = 0

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES9	ADRES8	ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2
bit7						bit0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-0 **ADRES<9:2>**: ADC结果寄存器位  
 10位转换结果的高8位

寄存器 1EH: ADC结果高位寄存器 (ADRESH), ADFM = 1

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	—	—	—	ADRES9	ADRES8
bit7						bit0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-2 保留：不要使用

bit1-0 **ADRES<9:8>**: ADC结果寄存器位  
 10位转换结果的高2位

### 9.4.4 ADRESL: ADC 结果低位寄存器

寄存器 9EH: ADC结果低位寄存器 (ADRESL), ADFM = 0

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES1	ADRES0	—	—	—	—	—	—
bit7						bit0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-6 **ADRES<1:0>**: ADC结果寄存器位  
 10位转换结果的低2位

bit5-0 保留：不要使用

寄存器 9EH: ADC结果低位寄存器 (ADRESL), ADFM = 1

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-0      **ADRES<7:0>**: ADC结果寄存器位  
10位转换结果的低8位

### 9.5 A/D 采集时间要求

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 9-5-1。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 9-5-1。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。可以使用公式 9-5 来计算最小采集时间。该公式假设误差为 1/2 LSb (ADC 转换需要 1024 步)。1/2LSb 误差是 ADC 达到规定精度所允许的最大误差。

- 注:
- 1: 由于参考电压 (VREF) 将自身消去, 因此对公式没有影响。
  - 2: 充电保持电容 (CHOLD) 在每次转换后并不放电。
  - 3: 模拟输入的最大建议源阻抗为10kΩ。这是为了满足引脚泄漏规范。

图 9-5-1: 模拟输入模型

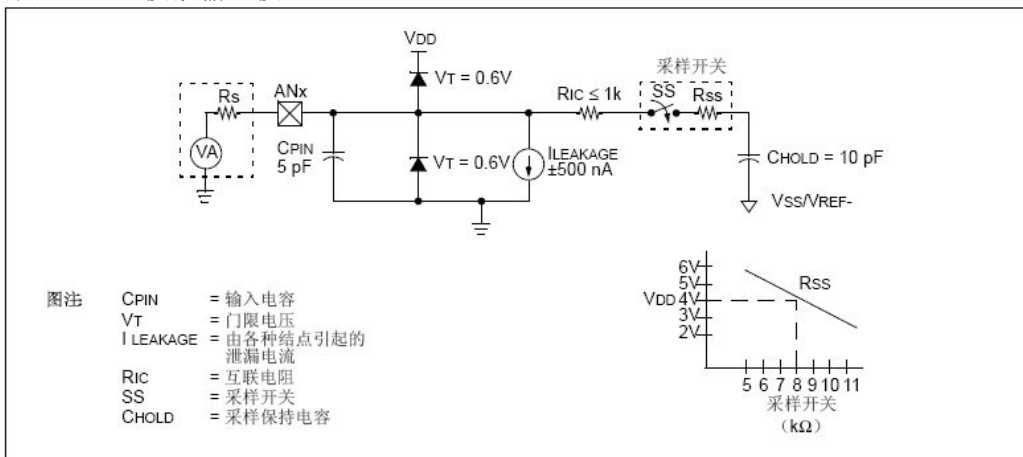
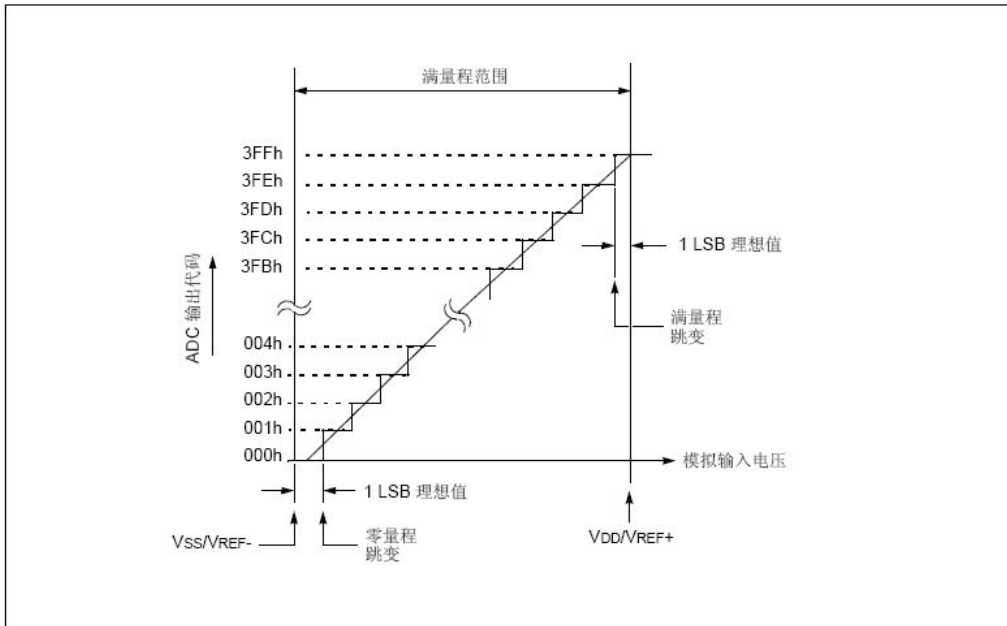


图 9-5-2: ADC 传递函数



公式 9-5: 采样时间示例

假设 温度 = 50°C, 外部电阻 10 kΩ, 5.0V V<sub>DD</sub>

$$T_{ACQ} = \text{放大器稳定时间} + \text{保持电容充电时间} + \text{温度系数}$$

$$= T_{AMP} + T_C + T_{COFF}$$

$$= 5 \mu\text{s} + T_C + [(50^\circ\text{C} - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})]$$

T<sub>C</sub> 值可以用以下公式近似计算:

$$V_{APPLIED} \left(1 - \frac{1}{2047}\right) = V_{CHOLD} \quad ; [1] \text{ 在 } 1/2 \text{ lsb 误差范围内对 } V_{CHOLD} \text{ 充电}$$

$$V_{APPLIED} \left(1 - e^{-\frac{T_C}{RC}}\right) = V_{CHOLD} \quad ; [2] \text{ 依照 } V_{APPLIED} \text{ 对 } V_{CHOLD} \text{ 充电}$$

$$V_{APPLIED} \left(1 - e^{-\frac{T_C}{RC}}\right) = V_{APPLIED} \left(1 - \frac{1}{2047}\right) \quad ; \text{ 结合 } [1] \text{ 和 } [2]$$

求解 T<sub>C</sub>:

$$T_C = -CHOLD(RIC + RSS + RS) \ln(1/2047)$$

$$= -10 \text{ pF}(1 \text{ k}\Omega + 7 \text{ k}\Omega + 10 \text{ k}\Omega) \ln 0.004885$$

$$= 1.37 \mu\text{s}$$

因此

$$T_{ACQ} = 5 \mu\text{s} + 1.37 \mu\text{s} + [(50^\circ\text{C} - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})]$$

$$= 7.67 \mu\text{s}$$

表 9-5: 与 ADC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
ADS0	ADFM	ADVRS1	ADVRS0	CHS2	CHS1	CHS0	GO/DONE	ADON	0000 0000	0000 0000
ADS1	ADRVO	ADCS2	ADCS1	ADCS0	—	—	—	—	0000 ----	0000 ----
ANINS	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
ADRESH	A/D 结果寄存器高字节								xxxx xxxx	uuuu uuuu
ADRESL	A/D 结果寄存器低字节								xxxx xxxx	uuuu uuuu
INTS	GIE	PEIE	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000	0000 0000

PIEB1	EEIE	ADIE	CCP1IE	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIFB1	EEIF	ADIF	CCP1IF	C2IF	C1IF	OSFIF	TMR2IF	TMRIF	0000 0000	0000 0000
PORTA	—	—	PA5	PA4	PA3	PA2	PA1	PA0	--x0 x000	--uu uuuu
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	--xx 0000	--uu uuuu
CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11 1111	--11 1111
CPIOC	CPIOC7	CPIOC6	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	1111 1111	1111 1111

图注： x = 未知， u = 不变， — = 未实现（读为 0）。ADC 模块不使用阴影单元。

## 10.0 比较器模块

比较器用于接口模拟电路与数字电路，它将两个模拟电压相比较并提供一个表示两者相对幅度数字值。比较器由于提供了独立于程序执行以外的模拟功能，因而对于混合信号构建模块是相当有用的。模拟比较器模块包含以下特性：

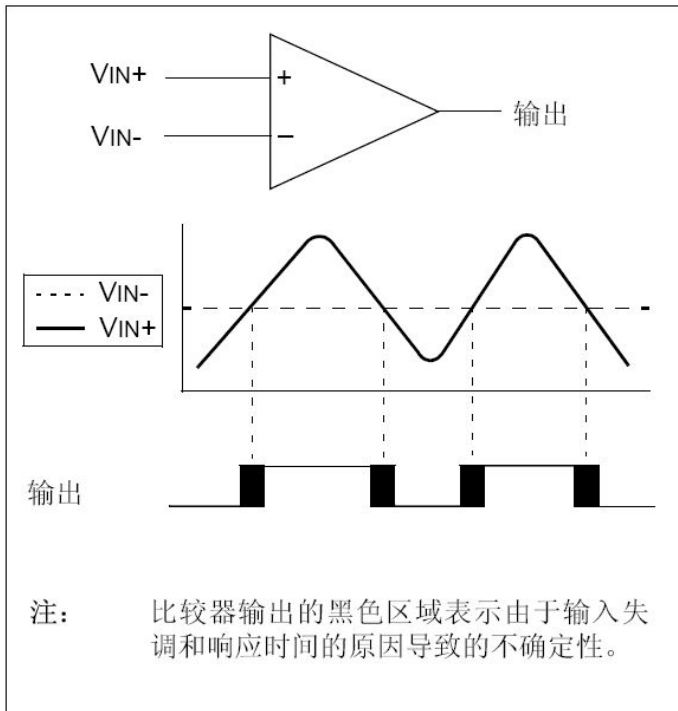
- 独立比较器控制
- 可编程输入选择
- 有内部/外部比较器输出
- 可编程输出极性
- 电平变化中断
- 从休眠中唤醒
- PWM 关闭
- Timer1 门控（计数使能）
- 输出与 Timer1 时钟输入同步
- SR 锁存器
- 可编程和固定参考电压

注： 只有比较器C2可连接到Timer1。

### 10.1 比较器概述

图 10-1-1 所示为单比较器及模拟输入电平和数字输出之间的关系。当 VIN+ 的模拟电压小于 VIN- 的模拟电压时，比较器输出数字低电平。当 VIN+ 的模拟电压大于 VIN- 的模拟电压时，比较器输出数字高电平。

图 10-1-1：单比较器



此器件包含两个比较器，如图 8-2 和图 8-3 所示。两个比较器均无法单独配置。

图 10-1-2: 单比较器

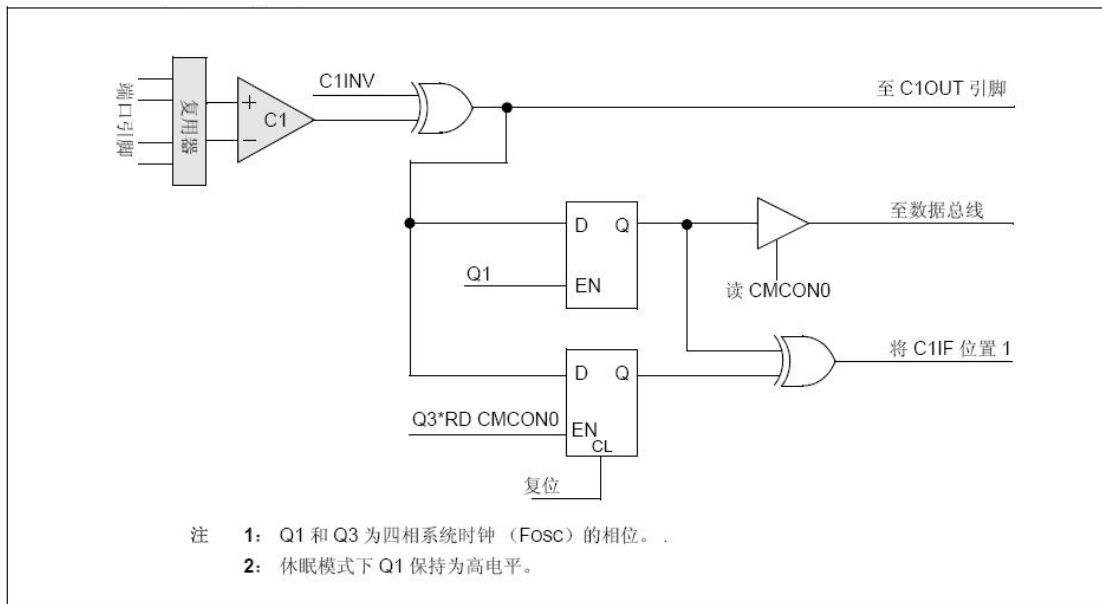
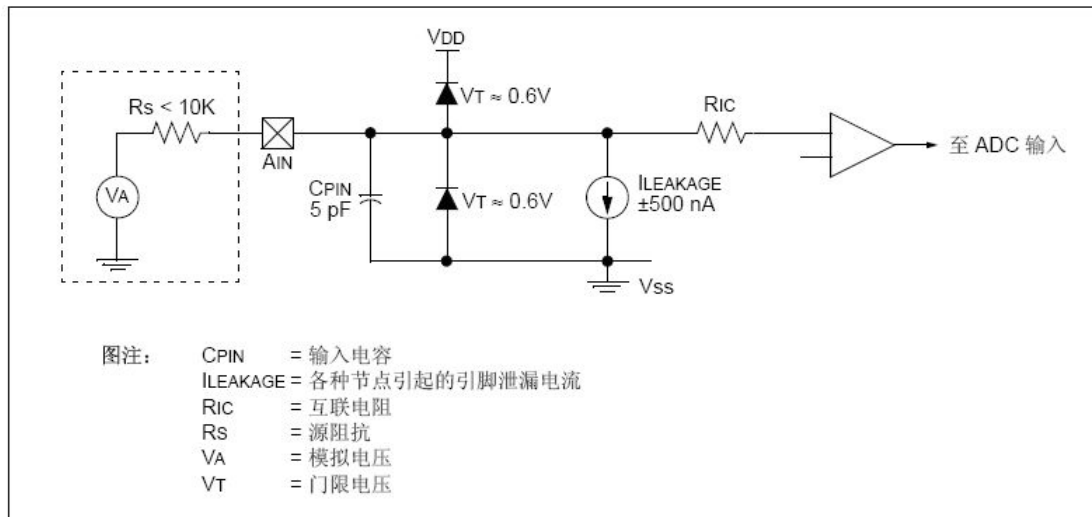


图 10-1-3: 比较器 C2 输出框图







## 10.2 比较器的配置（图）

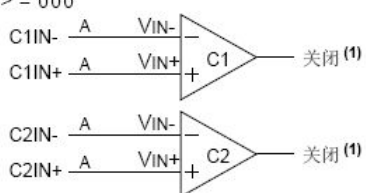
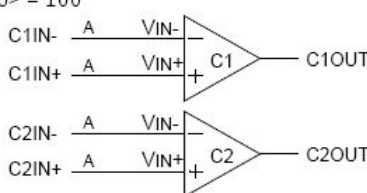
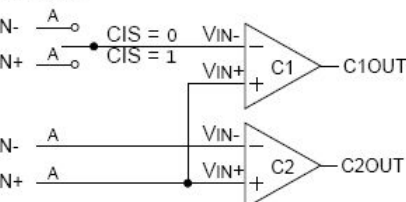
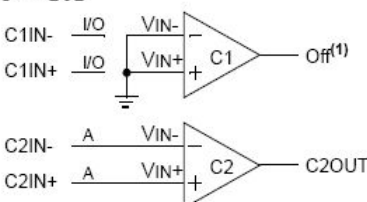
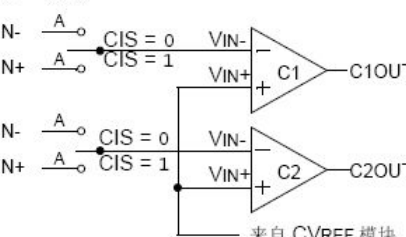
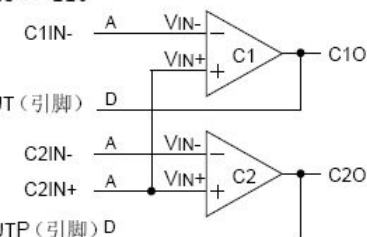
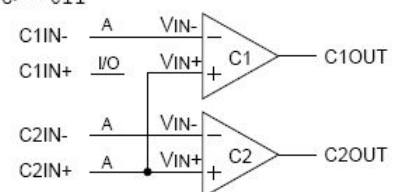
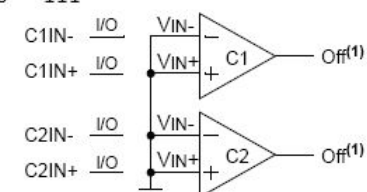
比较器有八种工作模式。CMSTA 寄存器的 CM<2:0>位用于选择这些模式，如[图 10-2](#)所示。I/O 线随着模式的变化而变化并指定为如下功能：

- 模拟功能（A）：数字输入缓冲器被禁止
- 数字功能（D）：比较器数字输出，覆盖端口功能
- 正常端口功能（I/O）：与比较器无关

表示为“A”的端口引脚将读为 0，无论 I/O 引脚或 I/O 控制 CPIO 位的状态如何。用作模拟输入的引脚还应将其对应的 CPIO 位置 1 以禁止数字输出驱动器。表示为“D”的引脚应将其对应的 CPIO 位清零以使能数字输出驱动器。

注： 在比较器模式改变期间应禁止比较器中断以防止意外中断。

图 10-2：比较器 I/O 的工作模式

<p>比较器复位 (POR 默认值) CM&lt;2:0&gt; = 000</p> 	<p>两个独立比较器 CM&lt;2:0&gt; = 100</p> 
<p>三个输入复用到两个比较器 CM&lt;2:0&gt; = 001</p> 	<p>一个独立比较器 CM&lt;2:0&gt; = 101</p> 
<p>四个输入复用到两个比较器 CM&lt;2:0&gt; = 010</p> 	<p>两个带输出的公共参考比较器 CM&lt;2:0&gt; = 110</p> 
<p>两个公共参考比较器 CM&lt;2:0&gt; = 011</p> 	<p>比较器关闭 (功耗最低) CM&lt;2:0&gt; = 111</p> 
<p>图注: A = 模拟输入, 端口始终读为 0 I/O = 正常端口 I/O</p> <p>注 1: 读为 0, 除非 CxINV = 1.</p>	<p>CIS = 比较器输入开关 (CMCON0&lt;3&gt;) D = 比较器数字输出</p>

## 10.3 比较器的控制

CMSTA 寄存器 (寄存器 19H) 提供了对比较器以下功能的控制:

- 模式选择
- 输出状态
- 输出极性
- 输入开关

### 10.3.1 比较器输出状态

每个比较器的状态均可通过 CMSTA 寄存器中相关的 CxOUT 位由内部读取。CM<2:0> = 110 时, 比较器输出到 CxOUT 引脚。选定该模式后, 相关 CxOUT 引脚的 CPIO 位必须清零以使能输出驱动器。

### 10.3.2 比较器输出极性（表）

翻转比较器输出与交换比较器输入的效果相同。比较器的输出极性可通过将 CMxCON0 寄存器的 CxPOL 位置 1 翻转。CxPOL 位清零将产生同相输出。表 10-3-2 所示将为包括极性控制在内的输出状态与输入条件的比较。

表 10-3-2: 比较器输出状态—输入条件

输入条件	CxINV	CxOUT
$V_{IN-} > V_{IN+}$	0	0
$V_{IN-} < V_{IN+}$	0	1
$V_{IN-} > V_{IN+}$	1	1
$V_{IN-} < V_{IN+}$	1	0

注：CxOUT 指寄存器位和输出引脚。

### 10.3.3 比较器输入开关

在以下模式下，比较器的翻转输入可以在模拟引脚之间切换：

- CM<2:0> = 001（仅限比较器 C1）
- CM<2:0> = 010（比较器 C1 和 C2）

在以上模式下，两个引脚均处于模拟模式下，而无论哪个引脚被选为输入。CMSTA 寄存器的 CIS 位控制比较器的输入开关。

## 10.4 比较器响应时间

在比较器的一个输入源发生改变或选择了另一个参考电压后，比较器的输出在一段时间内无法确定。这段时间称为响应时间。比较器的响应时间随参考电压的稳定时间不同而不同。因此，在确定比较器输入改变时的总响应时间时，必须将这两个时间考虑在内。详情请参见第 13.0 节“电气特性”中比较器和参考电压规范。

## 10.5 比较器中断的工作原理

只要比较器的输出值有所变化，比较器的中断标志就可以被置 1。变化的识别是通过由两个锁存器和一个异或门组成的失配电路（见图 10-1-2 和图 10-1-3）来实现的。当读取 CMxSTA0 寄存器时，一个锁存器被比较器输出电平更新。该锁存器保存该值直到下次读取 CMxSTA0 寄存器或发生复位。失配电路的另一个锁存器被系统时钟的每个 Q1 更新。在 Q1 时钟周期，当比较器输出的变化送达第二个锁存器时，将产生失配条件。此时，失配的两个锁存器中有着相反的输出电平，这两个电平被异或门检测到并输送给中断电路。失配条件将持续到 CMxSTA0 寄存器被读取或比较器输出返回到先前的状态为止。

注：对 CMSTA 寄存器的写操作也将清除不匹配条件，这是因为所有写操作均在写周期的开始包含一个读操作。

软件需保存比较器输出的状态信息以确定所发生的实际变化。

PIR1 寄存器的 CxIF 位是比较器中断标志。该位必须用软件清零以复位。由于可以向该寄存器写入 1，有可能产生一个软件中断。

必须将 PIEB1 寄存器的 CxIE 位和 INTS 寄存器的 PEIE 和 GIE 位全都置 1 以允许比较器中断。如果以上任一位清零，则中断未被允许，但在中断条件发生时，PIR1 寄存器的 CxIF 位仍将置 1。

用户可在中断服务程序中通过以下方法清除中断：

- a) 对 CMSTA 的任何读或写操作。这将结束不匹配条件。
- b) 将中断标志 CxIF 清零。

如果不匹配条件持续下去，CxIF 中断标志不会清零。则读 CMSTA 将结束不匹配条件并允许 CxIF 位被清零。

图 10-5-1：不读 CMSTA 时的比较器中断时序

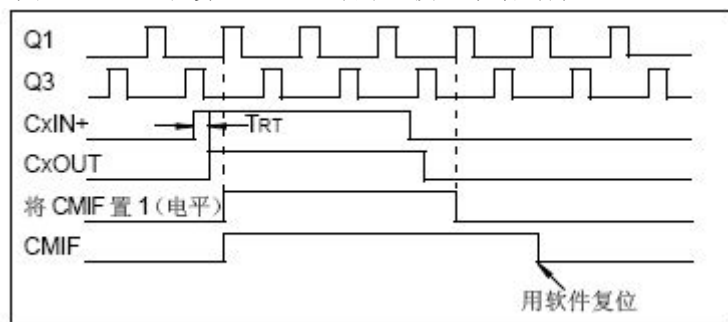
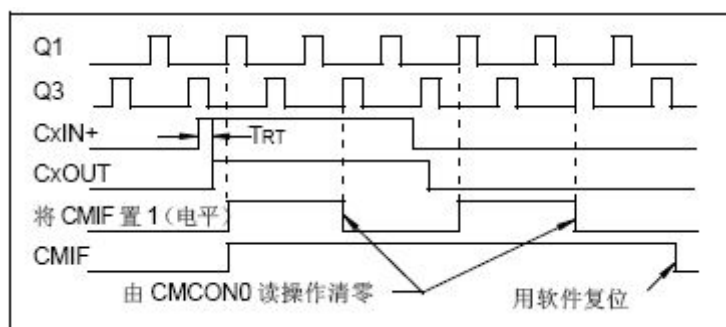


图 10-5-2：读取 CMSTA 时的比较器中断时序



- 注：
- 1: 如果在读操作正在执行时不得不改变 CMxSTA0 寄存器 (CxOUT) (Q2 周期的起始)，则 PIR1 寄存器的 CxIF 中断标志不一定会被置 1。
  - 2: 当任一比较器先被使能时，比较器模块中的偏置电路可产生无效的比较器输出，直到偏置电路稳定为止。应留有 1us 偏置电路稳定时间，然后在使能比较器中断前清除失配条件，并将中断标志清零。

## 10.6 休眠时的工作原理

如果在比较器进入休眠模式前将其使能，它将在休眠期间保持激活状态。比较器消耗的额外电流在 [第 13.0 节“电气特性”](#) 中单独列出。如果不使用比较器唤醒器件，可将比较器关闭，使其在休眠期间的功耗降至最低。通过选择 CMSTA 寄存器的 CM<2:0> = 000 或 CM<2:0> = 111 模式，可将比较器关闭。

比较器输出变化可将器件从休眠模式唤醒。要使能比较器唤醒休眠的器件的功能，必须将 PIEB1 寄存器的 CxIE 位和 INTS 寄存器的 PEIE 位置 1。器件从休眠唤醒后，将始终执行紧随休眠指令之后的一条指令。如果 INTS 寄存器的 GIE 位也被置 1，器件将执行中断服务程序。

## 10.7 复位的影响

器件复位将强制 CMSTA 和 CMCON1 寄存器为其复位状态。这迫使比较器模块处于比较器复位模式 (CM<2:0> = 000)。这样, 所有比较器输入均为模拟输入, 且比较器被禁止, 消耗最少的电流。

寄存器 19H: 比较器配置寄存器 (CMSTA)

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>C2OUT</b>	<b>C1OUT</b>	<b>C2INV</b>	<b>C1INV</b>	<b>CIS</b>	<b>CM2</b>	<b>CM1</b>	<b>CM0</b>
Bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

- bit7      **C2OUT**: 比较器2输出位  
           C2INV = 0时:  
           1 = C2VIN+>C2VIN-  
           0 = C2VIN+<C2VIN-  
           C2INV = 1时:  
           1 = C2VIN+<C2VIN-  
           0 = C2VIN+>C2VIN-
- bit6      **C1OUT**: 比较器1输出位  
           C1INV = 0时:  
           1 = C1VIN+>C1VIN-  
           0 = C1VIN+<C1VIN-  
           C1INV = 1时:  
           1 = C1VIN+<C1VIN-  
           0 = C1VIN+>C1VIN-
- bit5      **C2INV**: 比较器2输出翻转位  
           1 = C2输出翻转  
           0 = C2输出不翻转
- bit4      **C1INV**: 比较器1输出翻转位  
           1 = C1输出翻转  
           0 = C1输出不翻转
- bit3      **CIS**: 比较器输入开关位  
           CM<2:0> = 010时:  
           1 = C1IN+连接至C1VIN-  
               C2IN+连接至C2VIN-  
           0 = C1IN-连接至C1VIN-  
               C2IN-连接至C2VIN-  
           CM<2:0> = 001时:  
           1 = C1IN+连接至C1VIN-  
           0 = C1IN-连接至C1VIN-

bit2-0	<b>CM&lt;2:0&gt;</b> : 比较器模式选择位（见图8-5）
	000 = 比较器关闭。CxIN引脚配置为模拟
	001 = 三个输入复用到两个比较器
	010 = 四个输入复用到两个比较器
	011 = 两个公共参考比较器
	100 = 两个独立比较器
	101 = 一个独立比较器
	110 = 两个带输出的公共参考比较器
	111 = 比较器关闭。CxIN引脚配置为数字I/O

## 10.8 比较器 C2 门控 Timer1

可使用此功能为模拟事件的持续时间或间隔定时。将 CMCON1 寄存器的 T1GSS 位清零将使 Timer1 根据比较器 C2 的输出递增。这要求 Timer1 打开且门控使能。详情请参见第 7.2 节“带门控的 Timer1 模块”。

当比较器用作 Timer1 门控源时，建议通过将 C2SYNC 位置 1 将比较器 C2 与 Timer1 同步。这将确保 Timer1 在递增时若比较器变化，Timer1 不会错过递增。

## 10.9 比较器 C2 输出与 Timer1 同步

通过将 CMCON1 寄存器的 C2SYNC 位置 1 可将比较器 C2 的输出与 Timer1 同步。使能时，比较器输出在 Timer1 时钟源的下降沿被锁存。Timer1 使用预分频器时，比较器输出在预分频后被锁存。为了防止竞争情况，比较器输出在 Timer1 时钟源下降沿被锁存，而 Timer1 在时钟源的上升沿递增。更多信息，请参见比较器框图（图 10-1-2 和图 10-1-3）以及 Timer1 框图（图 7-2）。

寄存器 1AH: 比较器配置寄存器 (CMCON1)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0
—	—	—	—	—	—	<b>T1GSS</b>	<b>C2SYNC</b>
bit7						bit0	

图注:

R = 可读位

W = 可写位

U = 未实现位，读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7-2 未实现：读为0

bit1 **T1GSS**: Timer1门控源选择位(1)  
 1 = Timer1门控源为T1G引脚（引脚应配置为数字输入）  
 0 = Timer1门控源为比较器C2的输出

bit0 **C2SYNC**: 比较器C2输出同步位(2)  
 1 = 输出与Timer1时钟的下降沿同步  
 0 = 异步输出

注 1: 请参见第 7.2.6 节“Timer1 门控”。  
 2: 请参见图 10-1-3。



## 10.10 比较器参考电压

比较器参考电压模块为比较器提供了一个内部产生的参考电压。它有以下特性：

- 独立于比较器操作以外
- 两个 16 级电压范围
- 输出钳位到 VSS
- 与 VDD 成比例
- 固定参考电压 (0.6V)

VRSTA 寄存器 ([寄存器 99H](#)) 控制参考电压模块，如 [图 10-10-4](#) 所示。

### 10.10.1 独立工作

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

### 10.10.2 输出电压选择

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

CVREF 输出电压由以下公式确定：

公式 10-10-2: Cvref 输出电压

$VRR = 1 \text{ (低电压范围):}$ $CVREF = (VR<3:0>/24) \times VDD$ $VRR = 0 \text{ (高电压范围):}$ $CVREF = (VDD/4) + (VR<3:0> \times VDD/32)$
---

由于模块的构造所限，无法实现 VSS 至 VDD 的满量程。请参见 [图 10-10-4](#)。

### 10.10.3 输出钳位到 Vss

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

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

这使比较器可进行过零检测而不消耗额外的 CVREF 模块电流。

### 10.10.4 输出与 Vdd 成比例

比较器的参考电压由 VDD 而来，因此 CVREF 输出随 VDD 的波动而变化。比较器参考电压的绝对准确度在 [第 13.0 节“电气特性”](#) 中列出。



寄存器 99H: 参考电压控制寄存器 (VRSTA)

R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VREN	—	VRR	—	VR3	VR2	VR1	VR0
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

- bit7      **VREN**: CVREF使能位  
1 = CVREF电路通电  
0 = CVREF电路断电, 无泄漏电流IDD且CVREF = VSS。
- bit6      未实现: 读为0
- bit5      **VRR**: CVREF范围选择位  
1 = 低电平范围  
0 = 高电平范围
- bit4      未实现: 读为0
- bit3-0    **VR<3:0>**: CVREF值选择位 ( $0 \leq VR<3:0> \leq 15$ )  
VRR = 1时:  $CVREF = (VR<3:0>/24) * VDD$   
VRR = 0时:  $CVREF = VDD/4 + (VR<3:0>/32) * VDD$

图 10-10-4: 比较参考电压框图

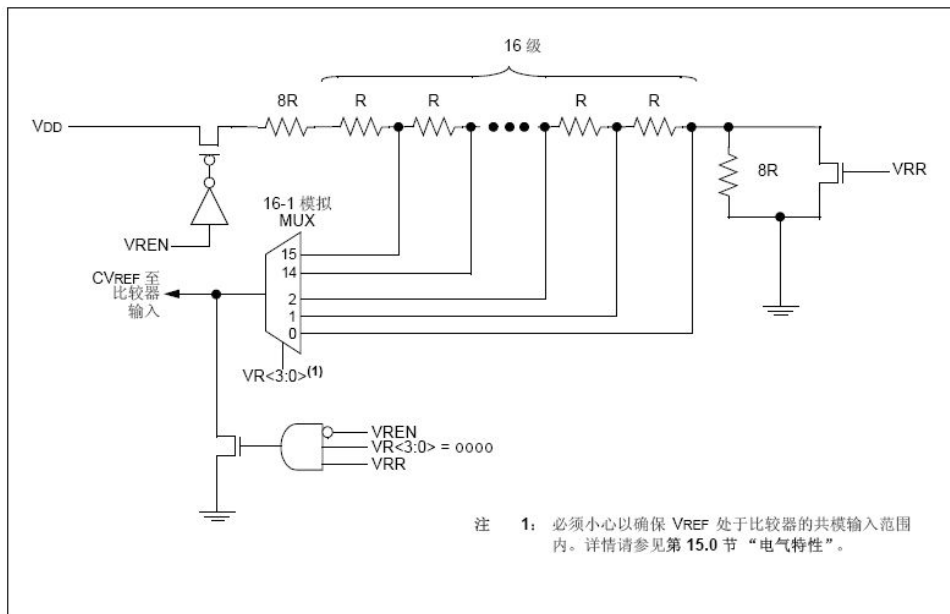


表 10-10-4: 与比较器参考电压模块相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位值
ANINS	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
CMSTA	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	---- --10
INTS	GIE	PEIE	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000	0000 0000
PIEB1	EEIE	ADIE	CCP1IE	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIFB1	EEIF	ADIF	CCP1IF	C2IF	C1IF	OSFIF	TMR2IF	TMRIF	0000 0000	0000 0000
PORTA	—	—	PA5	PA4	PA3	PA2	PA1	PA0	--x0 x000	--uu uu00
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	--xx 0000	--uu uu00
CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11 1111	--11 1111
CPIOC	CPIOC7	CPIOC6	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	1111 1111	1111 1111
VRSTA	VREN	—	VRR	—	VR3	VR2	VR1	VR0	0-0- 0000	0-0- 0000

图注: x = 未知, u = 不变, — = 未实现 (读为 0)。比较器不使用阴影单元。

## 11.0 配置寄存器

配置寄存器是用来配置 MCU 的设置，其原始值存在 Flash 的某些位置，在 MCU 启动时从 Flash Boot 到配置寄存器中，配置寄存器位宽为 8 位，由于 Flash 的位宽为 14 位，Boot 时，只 Boot 低 8 位至配置寄存器。

### 11.1 Option ROM

Option ROM 又分为 Body Option ROM 和 Customer Option ROM，Body Option ROM 用于校准芯片的参数以及公司内部的一些设置，Customer Option ROM 用于给客户提供配置选项。Flash 的数据位宽为 14 位，地址 1200H~120FH 为 16 Word Customer Option ROM，用于存放客户的配置信息；地址 1210H~121FH 为 16 Word Body Option ROM，用于存放厂家的配置信息。

Option ROM 位宽为 14 位，但只使用低 8 位，高 6 位无作用。

### 11.2 Config Option

寄存器 1201H: 配置字寄存器高8位 (Config1 Option)

—	RD_CTRL	LVDS1	LVDS0	FCMEN	IESO	LVREN1	LVREN0
bit15							bit8

寄存器 1200H: 配置字寄存器低8位 (Config0 Option)

CPDB	CPB	MCLRE	PWRTE	WDTE	FOSC2	FOSC1	FOSC0
bit7							bit0

bit15 未实现：保留

bit14 **RD\_CTRL**: 输出模式时读端口控制  
1: 读数据端口返回的PIN上的值  
0: 读数据端口返回的锁存器上的值

bit13-12 **LVDS <1:0>**: 低电压复位电压设定  
11: 1.7V  
10: 1.7V  
01: 2.1V  
00: 3.8V

bit11 **FCMEN**: 系统时钟失效监测使能位  
1: 使能系统时钟失效监测，仅在系统时钟为外部晶振模式才起作用  
0: 禁止系统时钟失效监测

bit10 **IESO**: 内、外时钟切换使能位，二速启动模式使能，  
1: 使能二速启动模式，仅在系统时钟为外部晶振模式才起作用  
0: 禁止二速启动模式

bit9-8 **LVREN <1:0>**: 低电压复位选择<sup>(1)</sup>  
11: 低电压复位使能  
10: MCU工作时，低电压复位开启；MCU睡眠时，低电压复位关闭  
01: 低电压复位由PCON之SBOREN位决定  
00: 禁止低电压复位

- bit7      **CPDB**: 数据代码保护位<sup>(2)</sup>  
 1: Data EEPROM内容不保护  
 0: 启动Data EEPROM内容保护, MCU能读, 串口不能读
- bit6      **CPB**: 代码保护位<sup>(3)</sup>  
 1: Flash 内容不保护  
 0: 启动Flash内容保护, MCU能读, 串口不能读
- bit5      **MCLRE**: PA3/ $\overline{\text{MCLR}}$ 引脚功能选择位<sup>(4)</sup>  
 1: PA3/MCLR脚执行MCLR功能, 是复位脚  
 0: PA3/MCLR脚执行PA3功能, 是数字输入引脚
- bit4      **PWRT**: 上电延时定时器使能位  
 1: PWRT禁止  
 0: PWRT使能
- bit3      **WDTE**: 看门狗定时器使能位  
 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
- bit2-0    **FOSC <2:0>**: 振荡器选择位  
 111: RC Oscillator模式, PA4输出时钟CLKOUT, PA5是时钟输入脚CLKIN  
 110: RCIO Oscillator模式, PA4为IO引脚, PA5时钟输入脚CLKIN  
 101: INTOSC Oscillator模式, PA4输出时钟CLKOUT, PA5为IO引脚  
 100: INTOSCIO Oscillator模式, PA4为IO引脚, PA5为IO引脚  
 011: EC模式, PA4为IO引脚, PA5时钟输入脚CLKIN  
 010: HS Oscillator模式, PA4、PA5接高频晶振  
 001: XT Oscillator模式, PA4、PA5接晶振  
 000: LP Oscillator模式, PA4、PA5接低频晶振
- 注        1: 使能欠压复位不会自动使能上电延时定时器。  
 2: 关闭代码保护时, 整个数据EEPROM都将被擦除。  
 3: 关闭代码保护时, 整个程序EEPROM都将被擦除。  
 4: 在INTOSC或RC模式下当 $\overline{\text{MCLR}}$ 有效时, 内部时钟振荡器被禁止。

## 12.0 指令表

Instruction Code	Mnemonic Operands	Function	Operation	Status
010000 00000000	NOP	No operation	None	None
010000 00000001	CLRWT	Clear Watchdog timer	0→WT	/TF, /PF
010000 00000010	SLEEP	Sleep mode	0→WT, stop OSC	/TF, /PF
010000 00000011	TMODE	Load W to TMODE register	W→TMODE	None
010000 00000rrr	CPIO R	Control I/O port register	W→CPIO r	None
010001 1rrrrrrr	STWR R	Store W to register	W→R	None
011000 trrrrrr	LDR R, t	Load register	R→t	Z
111010 iiiiii	LDWI I	Load immediate to W	I→W	None
010111 trrrrrr	SWAPR R, t	Swap halves register	[R(0~3) R(4~7)]→t	None
011001 trrrrrr	INCR R, t	Increment register	R + 1→t	Z
011010 trrrrrr	INCRSZ R, t	Increment register, skip if zero	R + 1→t	None
011011 trrrrrr	ADDWR R, t	Add W and register	W + R→t	C, HC, Z
011100 trrrrrr	SUBWR R, t	Subtract W from register	R AW→t (R+W+1→t)	C, HC, Z
011101 trrrrrr	DECR R, t	Decrement register	R A1→t	Z
011110 trrrrrr	DECRSZ R, t	Decrement register, skip if zero	R A1→t	None
010010 trrrrrr	ANDWR R, t	AND W and register	R ∩ W→t	Z
110100 iiiiii	ANDWI I	AND W and immediate	I ∩ W→W	Z
010011 trrrrrr	IORWR R, t	Inclu. OR W and register	R ∪ W→t	Z
110101 iiiiii	IORWI I	Inclu. OR W and immediate	I ∪ W→W	Z
010100 trrrrrr	XORWR R, t	Exclu. OR W and register	R ⊕ W→t	Z
110110 iiiiii	XORWI I	Exclu. OR W and immediate	I ⊕ W→W	Z
011111 trrrrrr	COMR R, t	Complement register	/R→t	Z
010110 trrrrrr	RRR R, t	Rotate right register	R(n) →R(n-1), C→R(7), R(0)→C	C
010101 trrrrrr	RLR R, t	Rotate left register	R(n)→r(n+1), C→R(0), R(7)→C	C
010000 1xxxxxxx	CLRW	Clear working register	0→W	Z
010001 0rrrrrrr	CLRR R	Clear register	0→R	Z
0000bb brrrrrrr	BCR R, b	Bit clear	0→R(b)	None
0010bb brrrrrrr	BSR R, b	Bit set	1→R(b)	None
0001bb brrrrrrr	BTSC R, b	Bit Test, skip if clear	Skip if R(b)=0	None
0011bb brrrrrrr	BTSS R, b	Bit Test, skip if set	Skip if R(b)=1	None
100nnn nnnnnnnn	LCALL N	Long CALL subroutine	N→PC, PC+1→Stack	None
101nnn nnnnnnnn	LJUMP N	Long JUMP to address	N→PC	None
110001 iiiiii	RTIW I	Return, place immediate to W	Stack→PC, I→W	None
110111 iiiiii	ADDWI I	Add immediate to W	PC+1→PC, W+I→W	C, HC, Z
111000 iiiiii	SUBWI I	Subtract W from immediate	I-W→W	C, HC, Z
010000 00001001	RTFI	Return from interrupt	Stack→PC, 1→GIS	None
010000 00000100	RET	Return from subroutine	Stack→PC	None

W	:	Working register	b	:	Bit position
WT	:	Watchdog timer	t	:	Target
TMODE	:	TMODE mode register	0	:	Working register
CPIO	:	Control I/O port register	1	:	General register
/TF	:	Timer overflow flag	R	:	General register address
/PF	:	Power loss flag	C	:	Carry flag
PC	:	Program Counter	HC	:	Half carry
OSC	:	Oscillator	Z	:	Zero flag
Inclu.	:	Inclusive 'U'	/	:	Complement
Exclu.	:	Exclusive '⊕'	x	:	Don't care
AND	:	Logic AND '∩'	I	:	Immediate data ( 8 bits )
			N	:	Immediate address

## 13.0 电气特性

### 13.1 绝对极限参数

偏置电压下的环境温度.....	-40℃ 至+85℃
存储温度 .....	-65℃ 至 +150℃
VDD 引脚相对于 VSS 的电压.....	-0.3V 至+6.5V
MCLR 引脚相对于 Vss 的电压.....	-0.3V 至+13.5V
所有其他引脚相对于 VSS 的电压.....	-0.3V 至(VDD+0.3V)
总功耗 <sup>(1)</sup> .....	800mW
流出 VSS 引脚的最大电流.....	95mA
流入 VDD 引脚的最大电流.....	95mA
输入钳位电流, $I_{IK}$ ( $V_I < 0$ 或 $V_I > VDD$ ) .....	$\pm 20$ mA
输出钳位电流, $I_{OK}$ ( $V_O < 0$ 或 $V_O > VDD$ ) .....	$\pm 20$ mA
任一 I/O 引脚的最大输出灌电流.....	25mA
任一 I/O 引脚的最大输出拉电流.....	25mA
PORTA 和 PORTC(联合)最大灌电流.....	90mA
PORTA 和 PORTC(联合)最大拉电流.....	90mA

注 1: 功耗计算公式为:  $P_{DIS} = VDD \times \{ I_{DD} - \sum I_{OH} \} + \sum \{ (V_{DD} - V_{OH}) \times I_{OH} \} + \sum (V_{OL} \times I_{OL})$

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

图 13-1-1: EN8F683 电压-频率图,  $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$

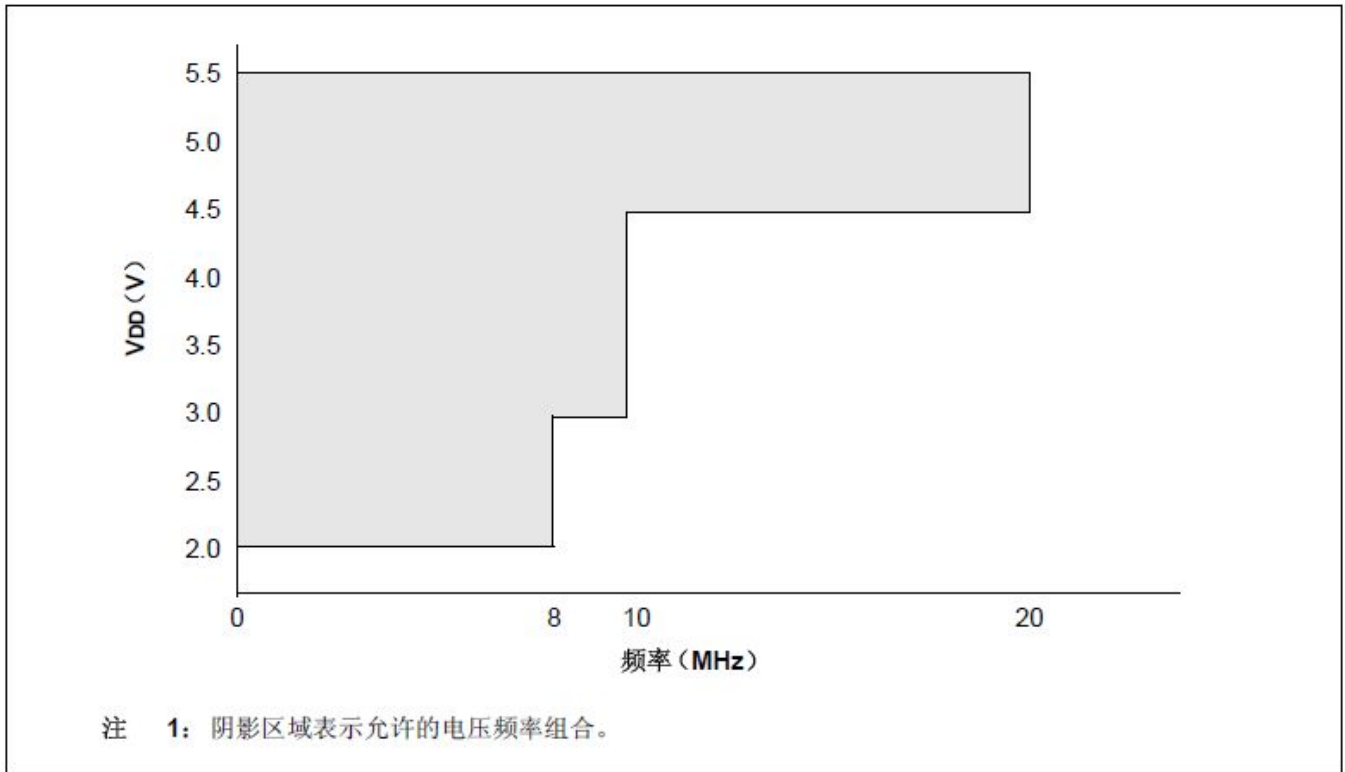
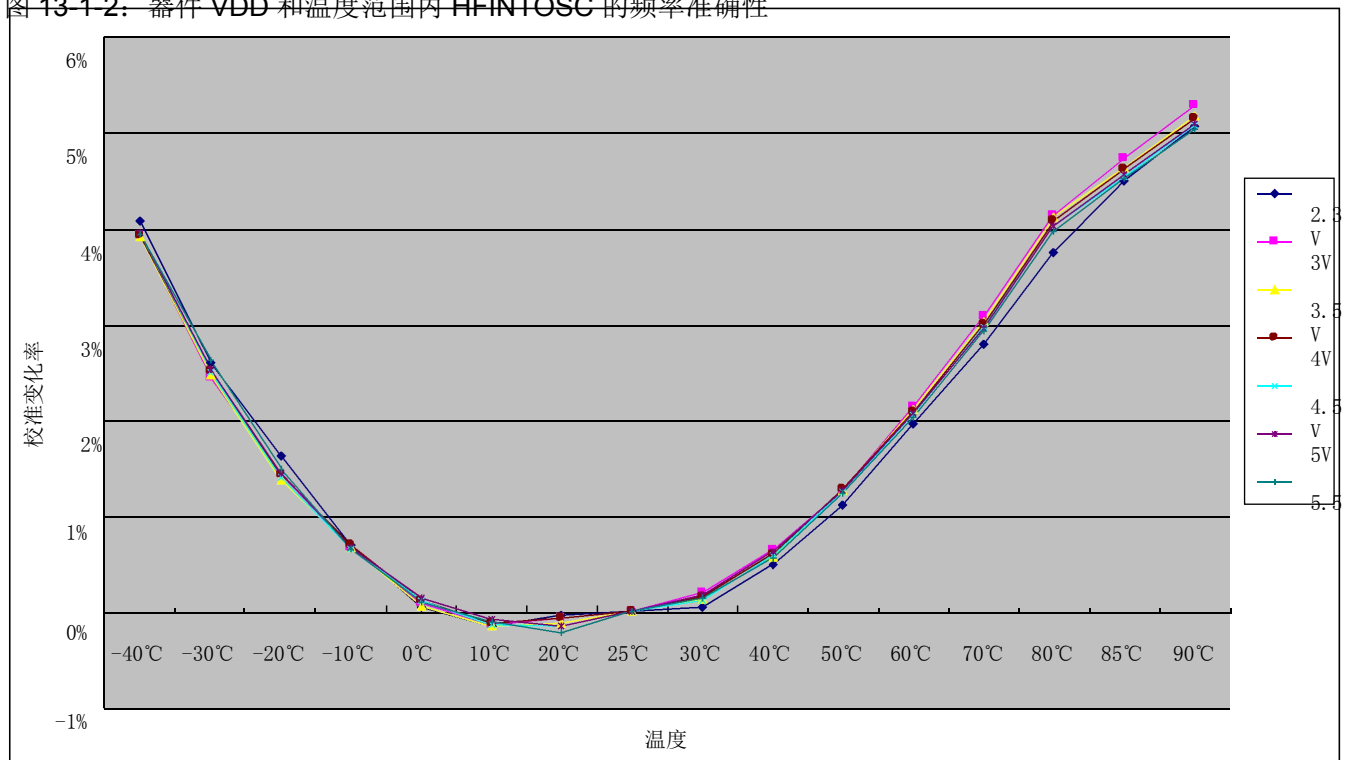


图 13-1-2: 器件 VDD 和温度范围内 HFINTOSC 的频率准确性



## 13.2 直流电器特性

直流特性		标准工作条件 工作温度 $-40^{\circ}\text{C} \leq T_a \leq +85^{\circ}\text{C}$						
符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件		
VDD	电源电压	2		5.5	V			
VDR	RAM 数据保持电压 <sup>(2)</sup>	—	0.5*	—	V	器件处于休眠模式		
VPOR	Vdd 起始电压确保能够产生上电复位信号	—	Vss	—	V			
SVDD	Vdd 上升速率确保能够产生上电复位信号	0.05*	—	—	V/ms			
IDD	工作电流 <sup>(3)</sup>	—	230	—	uA	3.3V	1M	WDT En PED Dis IRC
			370				4M	
			570			5V	8M	
			320				1M	
			580				4M	
			880				8M	
IPD	掉电流 <sup>(4)</sup>	—	1	—	uA	WDT Disable VDD=2.5V		
$\Delta I_{WDT}$	WDT 电流 <sup>(4)</sup>	—	0.5	—	uA	VDD=5V		
VIL	输入低电压	VSS	—	1.1	V	3V TLL		
		VSS	—	1.1		3V SCHMITT		
		VSS	—	1.6		5V TLL		
		VSS	—	1.6		5V SCHMITT		
VIH	输入高电压	1.1	—	VDD	V	3V TLL		
		1.9	—	VDD		3V SCHMITT		
		1.6	—	VDD		5V TLL		
		3.5	—	VDD		5V SCHMITT		
IOL	输出灌电流	—	24	—	mA	VOL=0.7V	3V	
		—	28	—		VOL=0.9V		
		—	35	—		VOL=0.7V	5V	
		—	42	—		VOL=0.9V		
IOH	输出拉电流	—	13	—	mA	VOH=1.7V	3V	
		—	10	—		VOH=2.1V		
		—	22	—		VOH=3.6V	5V	
		—	12	—		VOH=4.2V		
VPED	低电压复位电压	1.7 -20%	1.7	1.7 +20%	V	LOW		
		2.1 -20%	2.1	2.1 +20%		MIDDLE		
		3.8 -20%	3.8	3.8 +20%		HIGH		
Rpu	上拉电阻	—	32	—	K	3V		
		—	18	—		5V		

注：“—”表示没有，或待定。

(1) 典型栏中数据均为 25°C 条件下值，此部分数据仅供参考。

(2) 该电压是保证不丢失 RAM 数据的最小 VDD。

(3) 工作电流主要随工作电压和频率而变化。其它因素，如总线负载、总线速率、内部代码执行模式和温度也会影响电流消耗。

(4) 掉电电流是在器件休眠时，所有 I/O 引脚都处于高阻态并且连接到 Vdd 或 Vss 时测得。



### 13.3 交流电气特性

交流特性		标准工作条件 工作温度 $-40^{\circ}\text{C} \leq T_a \leq +85^{\circ}\text{C}$				
符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件
F <sub>OSC</sub>	系统时钟	0	8M	20M	Hz	5V
F <sub>HIRC</sub>	内部高速时钟	—	4M	8M	Hz	5V
F <sub>LIRC</sub>	内部低速时钟	—	31K	—	Hz	5V
T <sub>EERD</sub>	EEPROM 读时间	—	1	—	Tins	指令周期
T <sub>EEWR</sub>	EEPROM 写时间	—	400us	1ms	s	5V
T <sub>INT</sub>	中断脉冲		2	3	Tins	指令周期
T <sub>SST</sub>	系统启动时间 (上电复位)		13	20	ms	5V
	系统启动时间 (由 SLEEP 模式唤醒, F <sub>sys</sub> 在 SLEEP 模式下关闭)		1024		Tsys	5V
						5V
	系统启动时间(由 SLEEP 模式唤醒, 老板主要在 SLEEP 模式下开启)		1024		Tsys	5V
T <sub>RSTD</sub>	系统复位延迟时间 (上电复位)		13	20	ms	5V
	系统复位时间 (WDT 正常复位)		77	84	ms	5V

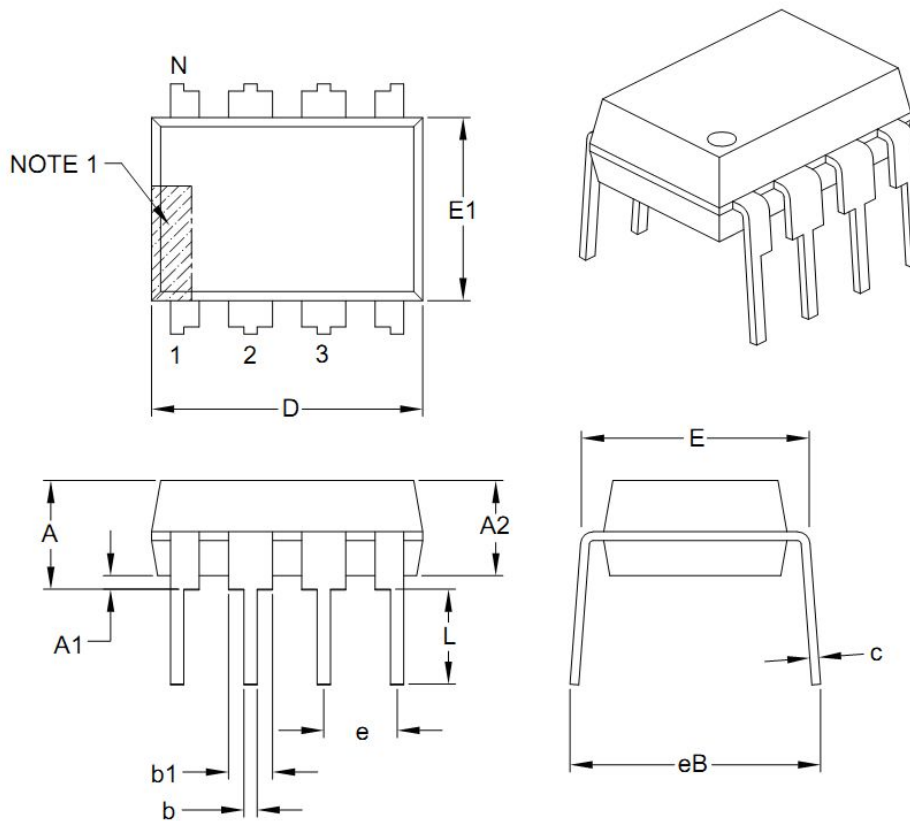
## 14.0 开发支持

### 14.1 仿真信息

### 14.2 烧录信息

## 15.0 封装信息

### 15.1 P-DIP 8 PIN:8-Lead Plastic Dual In-Line (P or PA) – 300 mil Body [PDIP]



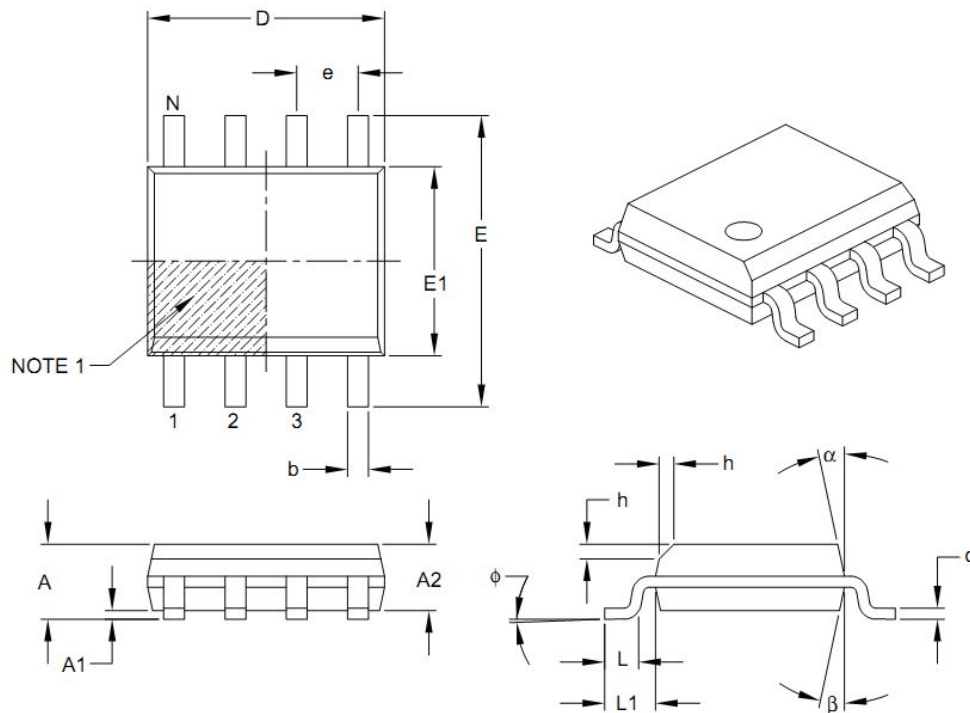
Dimension Limits	Units	INCHES		
		MIN	NOM	MAX
Number of Pins	N	8		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.348	.365	.400
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

**Notes:**

- Pin 1 visual index feature may vary, but must be located with the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

## 15.2 8-Lead Plastic Small Outline (SN or OA) - Narrow, 3.90 mm Body [SOIC]



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	8		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	1.75
Molded Package Thickness	A2	1.25	-	-
Standoff §	A1	0.10	-	0.25
Overall Width	E	6.00 BSC		
Molded Package Width	E1	3.90 BSC		
Overall Length	D	4.90 BSC		
Chamfer (optional)	h	0.25	-	0.50
Foot Length	L	0.40	-	1.27
Footprint	L1	1.04 REF		
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.17	-	0.25
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

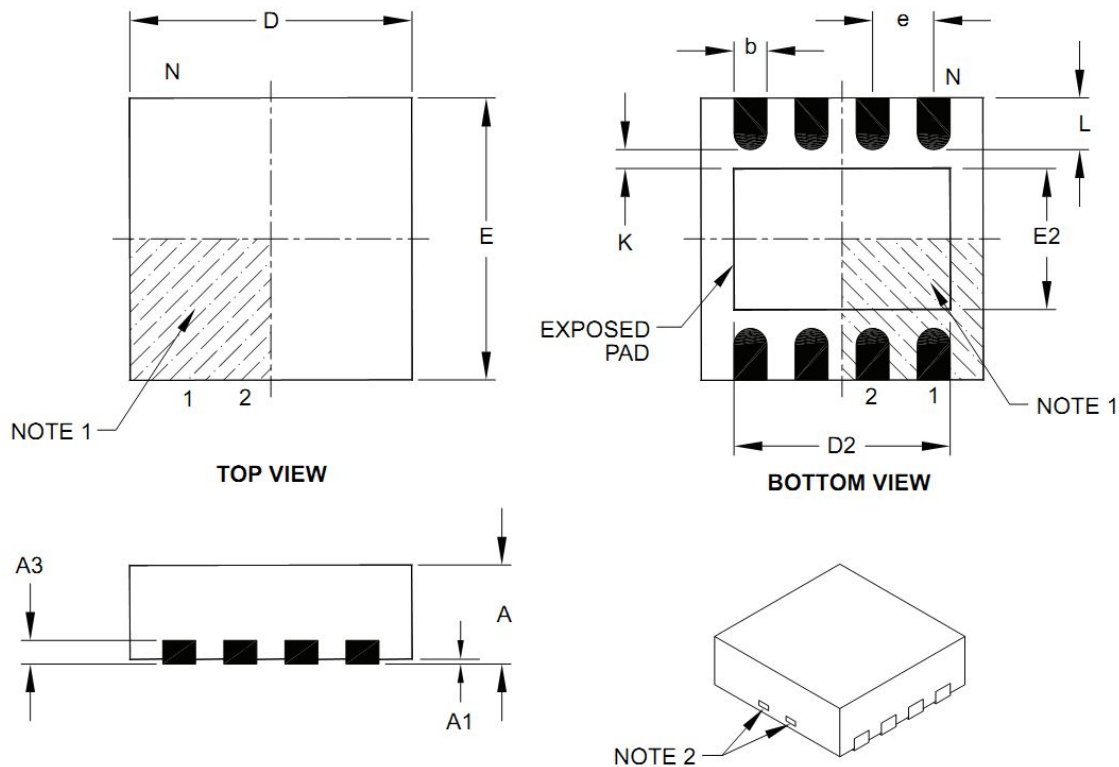
**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

### 15.3 8-Lead Plastic Dual Flat, No Lead Package (MD) - 4x4x0.9 mm Body [DFN]



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	8		
Pitch	e	0.80 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Length	D	4.00 BSC		
Exposed Pad Width	E2	0.00	2.20	2.80
Overall Width	E	4.00 BSC		
Exposed Pad Length	D2	0.00	3.00	3.60
Contact Width	b	0.25	0.30	0.35
Contact Length	L	0.30	0.55	0.65
Contact-to-Exposed Pad	K	0.20	-	-

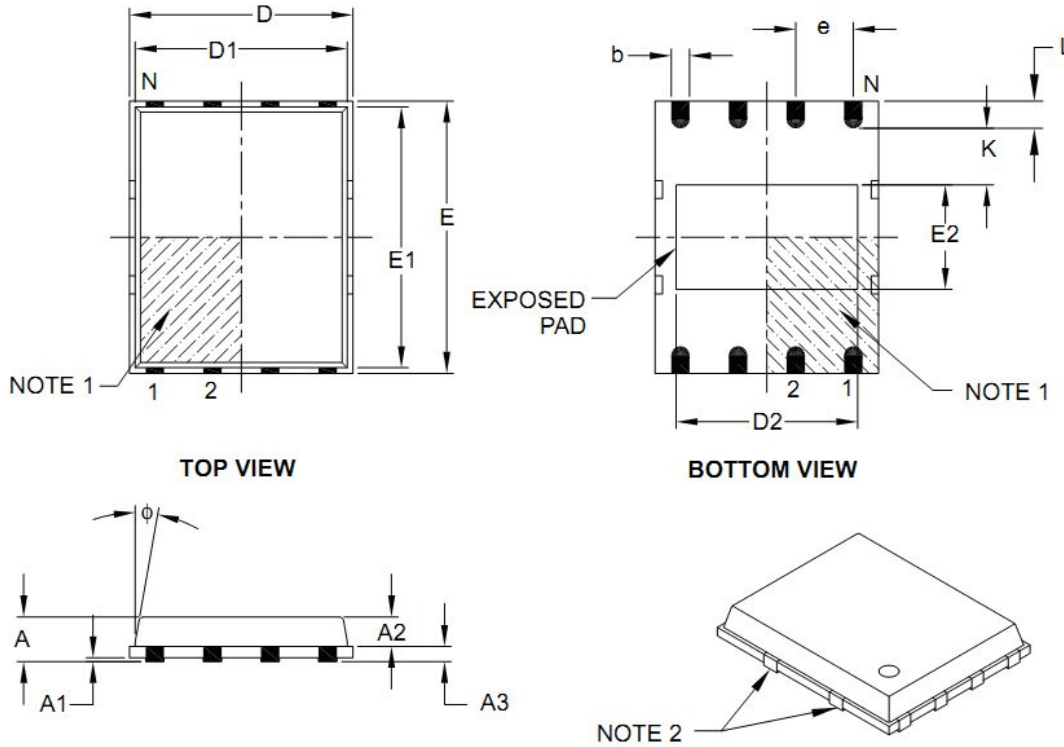
**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package may have one or more exposed tie bars at ends.
3. Package is saw singulated.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

15.4 8-Lead Plastic Dual Flat, No Lead Package (MF) - 6x5 mm Body [DFN-S]



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	8		
Pitch	e	1.27 BSC		
Overall Height	A	-	0.85	1.00
Molded Package Thickness	A2	-	0.65	0.80
Standoff	A1	0.00	0.01	0.05
Base Thickness	A3	0.20 REF		
Overall Length	D	4.92 BSC		
Molded Package Length	D1	4.67 BSC		
Exposed Pad Length	D2	3.85	4.00	4.15
Overall Width	E	5.99 BSC		
Molded Package Width	E1	5.74 BSC		
Exposed Pad Width	E2	2.16	2.31	2.46
Contact Width	b	0.35	0.40	0.47
Contact Length	L	0.50	0.60	0.75
Contact-to-Exposed Pad	K	0.20	-	-
Model Draft Angle Top	φ	-	-	12°

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package may have one or more exposed tie bars at ends.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.  
 REF: Reference Dimension, usually without tolerance, for information purposes only.

## 16. 开发说明

### • 整体说明

EN8F683 与 PIC12F683 完全兼容，开发时可考虑用 PIC12F683 开发，开发完后可将 HEX 文件直接导入到 EN 烧录器进行烧录。

以下为 EN8F683 与 PIC12F683 之间的差异：

差异点见下表：

	PIC12F683	EN8F683
PC 口	没有	有
PWM	不具备全桥、半桥模式	具备全桥、半桥模式（并带死区）
ADC	4 路	4 路
PIN5 对应引脚	PA2	PC2

针对差异，程序从 PIC12F683 移植到 EN8F683（对应 EN8F684 的 8PIN 封装）时注意事项：

1、对 PC 口进行初始化（因 EN8F683 实质为 EN8F684 的 8 pin 封装，为保证功耗，未封装出的脚需要输出为低电平）；

2、对比较器 2 初始化；

3、重新根据 EN8F684 的 PWM 相关寄存器设置 PWM 输出，另外我司针对 PWM 对引脚作了调整（见封装图 PC2 位置），取消了原来 PA2 口（及 PA2 口相应的功能），设置 PWM 输出时要特别注意（应根据 PWM 模式输出正确的端口，PC2 对应 P1D）；

4、ADC 模拟数据口设置从 PIC12F683 的 9FH 改到了 EN8F683 的 91H 寄存器，但 ADC 的转换时钟还是由 9FH 寄存器设置，使用 ADC 时要注意；





1、PORTC 部分

EN8F683 寄存器 87H: PORTC 寄存器 (PORTC)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-0	R/W-0	R/W-0	R/W-0
<b>P</b>	<b>P</b>	<b>P</b>	<b>P</b>	<b>P</b>	<b>P</b>	<b>P</b>	<b>P</b>
bit7							bit0

图注:  
 R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
 -n = POR 时的值                1 = 置 1                              0 = 清零                              x = 未知

bit7-0        **PC<7:0>**: PORTC I/O 引脚位

EN8F683 寄存器 87H: PORT C 三态寄存器 (CPIOC)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
<b>CPIOC7</b>	<b>CPIOC6</b>	<b>CPIOC5</b>	<b>CPIOC4</b>	<b>CPIOC3</b>	<b>CPIOC2</b>	<b>CPIOC1</b>	<b>CPIOC0</b>
bit7							bit0

图注:  
 R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
 -n = POR 时的值                1 = 置 1                              0 = 清零                              x = 未知

bit7-0        **CPIOC<7:0>**: PORTC 三态控制位  
 1 = PORTC 引脚配置为输入 (三态)  
 0 = PORTC 引脚配置为输出

2、PWM 部分

寄存器 16H: 增强型 PWM 控制寄存器 PWMICON

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>PRSEN</b>	<b>PDC6</b>	<b>PDC5</b>	<b>PDC4</b>	<b>PDC3</b>	<b>PDC2</b>	<b>PDC1</b>	<b>PDC0</b>
bit7							bit0

图注:							
R = 可读位	W = 可写位		U = 未实现位, 读为 0				
-n = POR 时的值	1 = 置 1		0 = 清零			x = 未知	

bit7        **PRSEN**: PWM 重启使能位  
 1 = 自动关闭时, ECCPASE 位在退出关闭事件时自动清零) PWM 自动重启  
 0 = 自动关闭时, 必须用软件将 ECCPASE 清零以重启 PWM

bit6-0        **PDC<6:0>**: PWM 延时计数位  
 PDCn = 预定 PWM 信号应转变为有效与 PWM 信号实际转为有效之间的 FOSC/4 (4\*TOSC 周期数  
 1: 速启动并选择了 LP、XT 或 HS 时, 或使能了故障保护模式时, 该位复位为 0。



## 寄存器 17H: 增强型捕捉/比较/PWM自动关闭控制寄存器 (ECCPAS)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>ECCPAS</b>	<b>ECCPAS</b>	<b>ECCPAS</b>	<b>ECCPAS</b>	<b>PSSAC1</b>	<b>PSSAC0</b>	<b>PSSBD1</b>	<b>PSSBD0</b>
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit7 **ECCPASE**: ECCP 自动关闭事件状态位  
 1 = 发生了自动关闭事件; ECCP 输出处于关闭状态  
 0 = ECCP 输出正常工作

bit6-4 **ECCPAS<2:0>**: ECCP 自动关闭源选择位  
 000 = 禁止自动关闭  
 001 = 比较器 1 输出变化  
 010 = 比较器 2 输出变化  
 011 = 比较器 1 或 2 之一变化  
 100 = INT 引脚电压为 VIL  
 101 = INT 引脚电压为 VIL 或比较器 1 变化  
 110 = INT 引脚电压为 VIL 或比较器 2 变化  
 111 = INT 引脚电压为 VIL 或比较器 1/2 之一变化

bit3-2 **PSSACn**: P1A 和 P1C 引脚关闭状态控位  
 00 = 驱动引脚 P1A 和 P1C 为 0  
 01 = 驱动引脚 P1A 和 P1C 为 1  
 1x = P1A 和 P1C 引脚为三态

bit1-0 **PSSBDn**: P1B 和 P1D 引脚关闭状态控制位  
 00 = 驱动引脚 P1B 和 P1D 为 0  
 01 = 驱动引脚 P1B 和 P1D 为 1  
 1x = P1B 和 P1D 引脚为三态

## 4、PWM 输出引脚部分

PIC16F683输出PWM引脚为PA2/P1A, EN8F683 输出PWM引脚为PC2/P1D,要注意CCP1寄存器输出PWM的引脚设置:

## 寄存器 15H: 增强型CCP1控制寄存器 (CCP1CON)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>P1M1</b>	<b>P1M0</b>	<b>DC1B1</b>	<b>DC1B0</b>	<b>CCP1M3</b>	<b>CCP1M2</b>	<b>CCP1M1</b>	<b>CCP1M0</b>
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit7-6 **P1M<1:0>**: PWM 输出配置位 如果  
 CCP1M<3:2> = 00、01 或 10:  
 xx = P1A 配置为捕捉/比较输入; P1B、P1C 和 P1D 配置为端口引  
 脚 如果 CCP1M<3:2> = 11:  
 00 = 单输出; P1A 调制; P1B、P1C 和 P1D 配置为端口引脚  
 01 = 全桥正向输出; P1D 调制; P1A 有效; P1B 和 P1C 无效  
 10 = 半桥输出; P1A 和 P1B 调制, 带有死区控制; P1C 和 P1D 配置为端口引脚

11 = 全桥反向输出；P1B 调制；P1C 有效；P1A 和 P1D 无效

bit5-4 **DC1B<1:0>**: PWM 占空比最低有效位  
 捕捉模式：  
 未使用。  
 比较模式：  
 未使用。  
 PWM 模式：  
 这些位是 PWM 占空比的低 2 位。高 8 位在 CCP1L 中。

bit3-0 **CCP1M<3:0>**: ECCP 模式选择位  
 0000 = 捕捉/比较/PWM 关闭（复位 ECCP 模块）  
 0001 = 未使用（保留）  
 0010 = 比较模式，匹配时翻转输出（CCP1IF 位置 1）  
 0011 = 未使用（保留）  
 0100 = 捕捉模式，每个下降沿  
 0101 = 捕捉模式，每个上升沿  
 0110 = 捕捉模式，每 4 个上升沿  
 0111 = 捕捉模式，每 16 个上升沿  
 1000 = 比较模式，匹配时输出置 1（CCP1IF 位置 1）  
 1001 = 比较模式，匹配时输出清零（CCP1IF 位置 1）  
 1010 = 比较模式，匹配时产生软件中断（CCP1IF 位置 1，CCP1 引脚不受影响）  
 1011 = 比较模式，触发特殊事件（CCP1IF 位置 1，CCP1 复位 TMR1，且如果 A/D 模块被使能，启动一次 A/D 转换）  
 1100 = PWM 模式；P1A 和 P1C 高电平有效；P1B 和 P1D 高电平有效  
 1101 = PWM 模式；P1A 和 P1C 高电平有效；P1B 和 P1D 低电平有效  
 1110 = PWM 模式；P1A 和 P1C 低电平有效；P1B 和 P1D 高电平有效  
 1111 = PWM 模式；P1A 和 P1C 低电平有效；P1B 和 P1D 低电平有效

5、ADCON0 部分

ADCON0 - A/D CONTROL REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>ADFM</b>	<b>VCFG</b>	—	—	<b>CHS1</b>	<b>CHS0</b>	<b>GO/DONE</b>	<b>ADON</b>
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bit 7 ADFM: A/D Conversion Result Format Select bit

- 1 = Right justified
- 0 = Left justified

bit 6 VCFG: Voltage Reference bit

- 1 = VREF pin
- 0 = VDD

bit 5-4 Unimplemented: Read as '0'

bit 3-2 CHS<1:0>: Analog Channel Select bits

- 00 = AN0
- 01 = AN1
- 10 = AN2

11 = AN3

bit 1 GO/DONE: A/D Conversion Status bit

1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.

This bit is automatically cleared by hardware when the A/D conversion has completed.

0 = A/D conversion completed/not in progress

bit 0 ADON: ADC Enable bit

1 = ADC is enabled

0 = ADC is disabled and consumes no operating current

### EN8F683 A/D控制寄存器0 (ADS0)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>ADFM</b>	<b>ADVRS1</b>	<b>ADVRS0</b>	<b>CHS2</b>	<b>CHS1</b>	<b>CHS0</b>	<b>GO/DONE</b>	<b>ADON</b>
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7 **ADFM**: A/D转换结果格式选择位

1 = 右对齐

0 = 左对齐

bit6-5 **ADVRS<6:5>**: 选择ADC的参考电压

00 = 选电压VDD做为ADC的参考

10 = 选外部(来自PA1)做为ADC的参考

X1 = 选内部BANDGAP的输出电压做为ADC的参考, 这个电压可以通过配置寄存器选择2V, 3V, 4V。这个电压也可以通过ADS1中的最高位ADRVO置1, PA1需要设成模拟输入(ANS1置1), 把参考电压送到PA1上。

bit4-2 **CHS<2:0>**: 模拟通道选择位

000 = AN0

001 = AN1

010 = AN2

011 = AN3

100 = AN4

101 = AN5

110 = AN6

111 = AN7

bit1 **GO/DONE**: A/D转换状态位

1 = A/D转换正在进行。将本位置1启动一次A/D转换。

A/D转换完成时此位由硬件自动清零。

0 = A/D转换完成/不在进行中

bit0 **ADON**: ADC使能位

1 = 使能ADC

0 = 禁止ADC, 不消耗工作电流

## 5、ANSEL 部分

## PIC12F683 ANSEL: ANALOG SELECT REGISTER (ADDRESS: 9Fh)

U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1
—	<b>ADCS2</b>	<b>ADCS1</b>	<b>ADCS0</b>	<b>ANS3</b>	<b>ANS2</b>	<b>ANS1</b>	<b>ANS0</b>
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

bit 7 Unimplemented: Read as '0'

bit 6-4 ADCS<2:0>: A/D Conversion Clock Select bits

000 = FOSC/2

001 = FOSC/8

010 = FOSC/32

x11 = FRC (clock derived from a dedicated internal oscillator = 500 kHz max)

100 = FOSC/4

101 = FOSC/16

110 = FOSC/64

bit 3-0 ANS<3:0>: Analog Select bits

Analog select between analog or digital function on pins AN<3:0>, respectively.

1 = Analog input. Pin is assigned as analog input(1).

0 = Digital I/O. Pin is assigned to port or special function.

Note 1: Setting a pin to an analog input automatically disables the digital input circuitry, weak pull-ups and interrupt-on-change,

if available. The corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

## EN8F683 A/D控制寄存器1 (ADS1)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
<b>ADRVO</b>	<b>ADCS2</b>	<b>ADCS1</b>	<b>ADCS0</b>	—	—	—	—
bit7							bit0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit7 **ADRVO**: 控制ADC的内部BANDGAP输出电压作为参考电压是否送到PA1端口上

1 = 送到PA1端口上

0 = 不送到PA1端口上。

bit6-4 **ADCS<2:0>**: A/D转换时钟选择位

000 = FOSC/2

001 = FOSC/8

010 = FOSC/32

x11 = F<sub>SM</sub>智能分频, ADC的转换时钟会根据振荡模式来选择分频比:

当系统时钟选择LP或者RC或者RCIO模式时, ADC的转换时钟就是系统时钟的2分频。

当系统时钟选择XT模式时, ADC的转换时钟就是系统时钟的8分频。

当系统时钟选择HF模式时, ADC的转换时钟就是系统时钟的32分频。

100 = FOSC/4

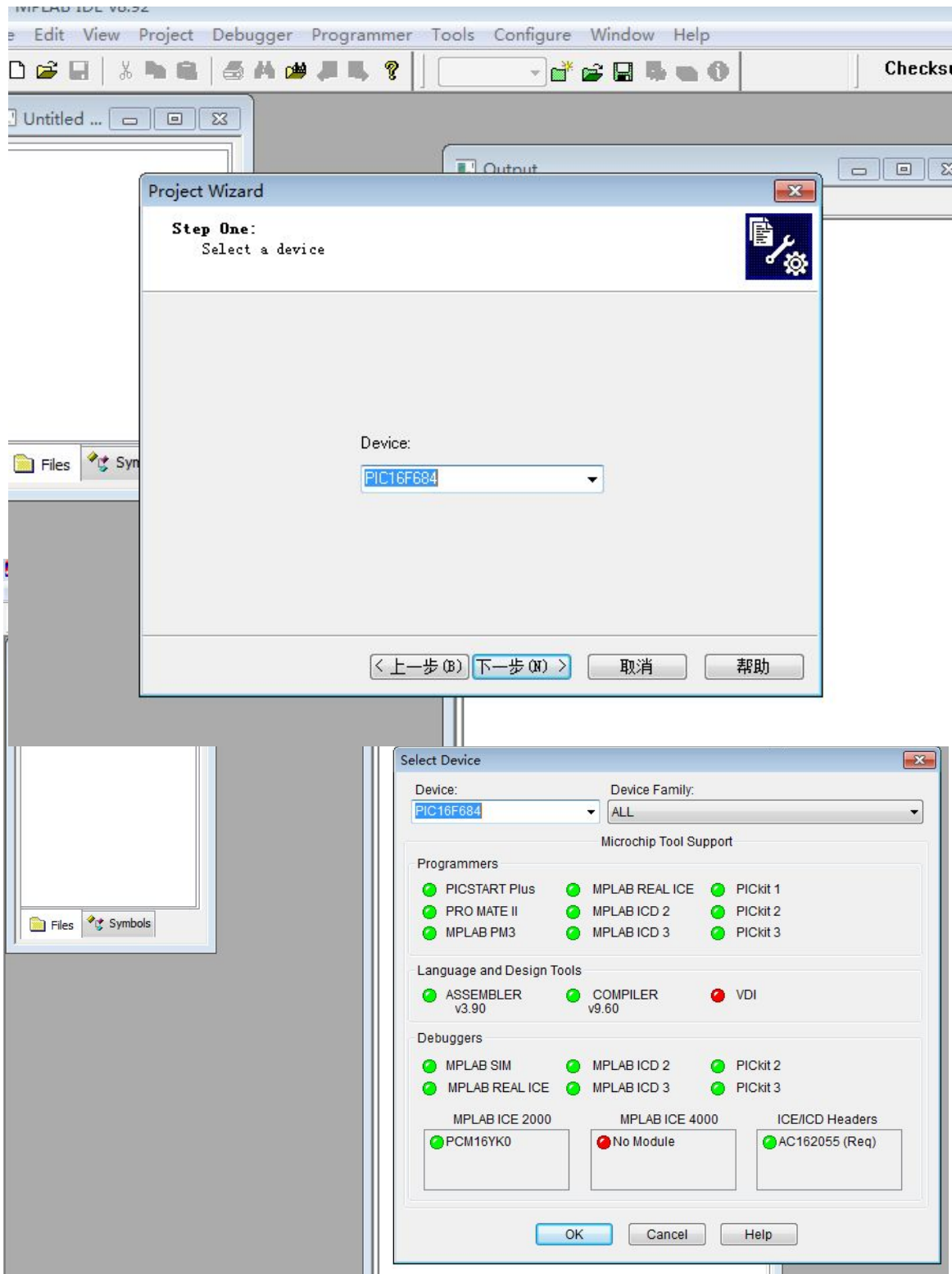
101 = FOSC/16

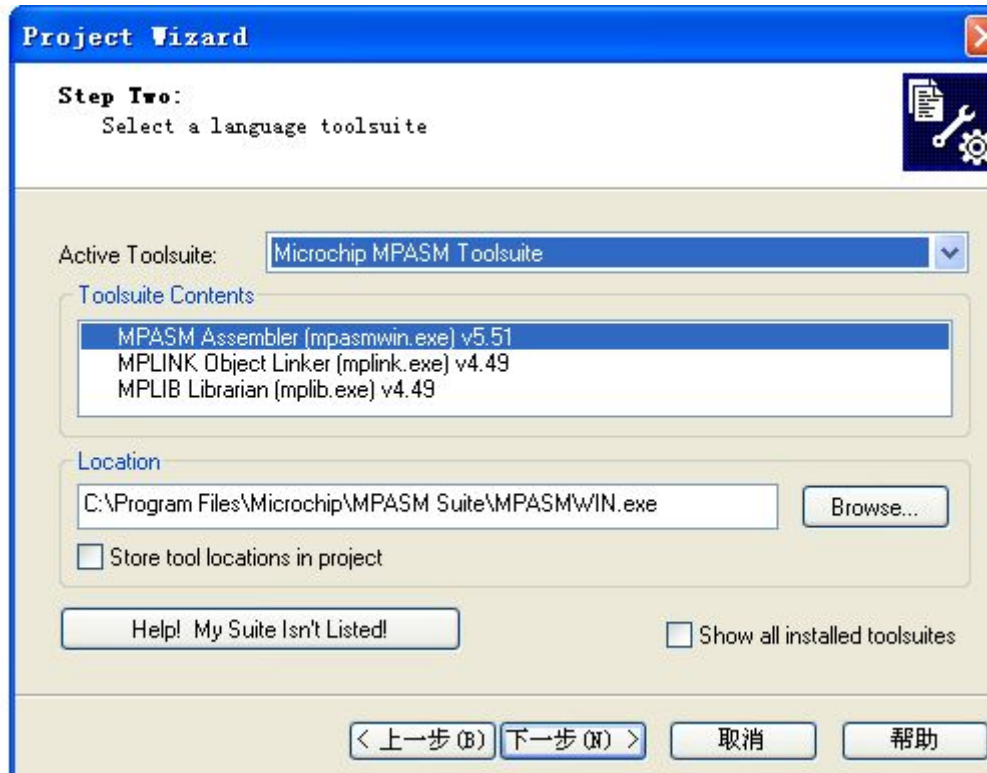
110 = FOSC/64

bit3-0 未实现: 读为0

## •开发流程

1、 打开 MPLAB-IDE，选择 PIC12F683，新建工程文件，使用编译器 MPASM 用汇编进开发

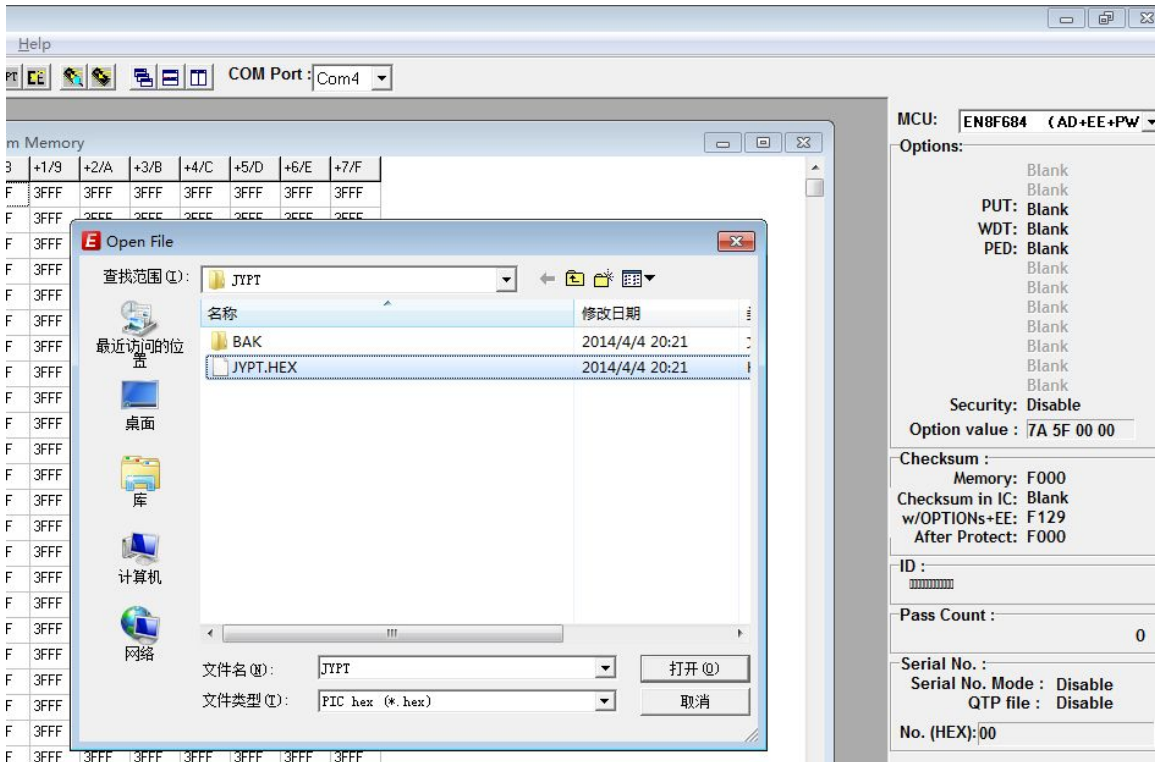




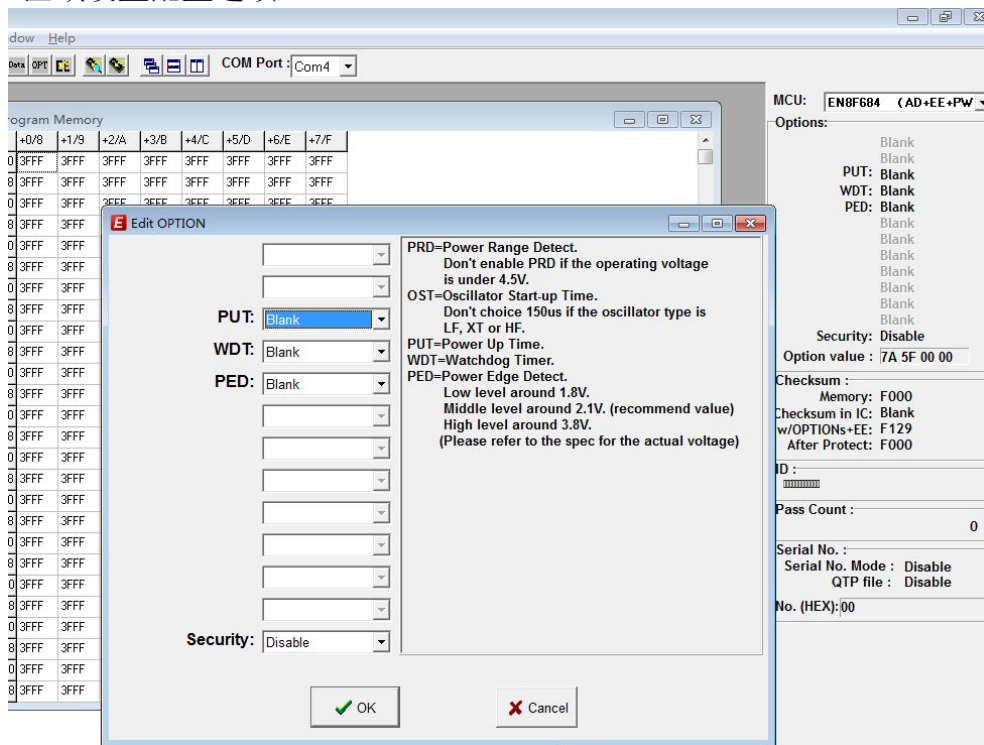
2、编译完成后生成 HEX 文件

3. 连接上 Enwriter, 打开 Enwriter 烧录软件 。

4. 导入程序并设置配置选项

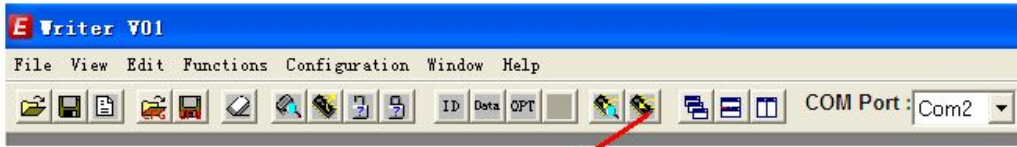


5. 双击 OPTION 区域设置配置选项

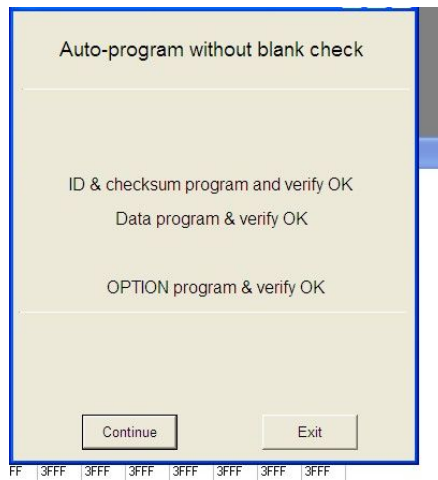




6. 烧录后上机测试



烧录按钮





**ENROO** 英锐恩

单片机集成方案全方位解决服务商

优质智能电子产品“芯”方案解决商

**联系信息：**

深圳市英锐恩科技有限公司  
ENROO-TECH (SHENZHEN) CO.,LTD  
中国·深圳市福田区福华路嘉汇新城汇商中心27楼2701室

Enroo-Tech Technologies CO., Limited  
香港荃灣青山道 388 號中國染廠大廈 17 樓 A 室

联系电话：86-755-82543411,83167411,83283911,88845951  
联系传真：86-755-82543511  
全国热线：4007-888-234  
联系邮件：[enroo@enroo.com](mailto:enroo@enroo.com)  
公司网站：<http://www.enroo.com> <http://www.enroo-tech.com>

