

# MDT10F685

## 数据手册（版本 V1.1）

### 8 位 FLASH CMOS 单片机

## 1.0 器件概述

### ◆ 存储器:

FLASH 空间: 4K\*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: 256 字节。

### ◆ 8 级深硬件堆栈

### ◆ I/O 引脚配置

具有独立方向控制的 22 个 I/O 引脚: PA 口 6 个、PB 口 8 个、PC 口 8 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 2 位预分频器的 16 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 62.5ns。
- 16 位比较模块, 最大分辨率为 62.5ns。
- 带一个 1、2 或 4 路输出通道, 可编程“死区时间”的 12 位 PWM 模块, 输出信号的最大频率为 8MHz。
- 带 3 个同周期输出可功能引脚转移 PWM 模块

### ◆ 12 通道 12 位 ADC

12 个外部 ADC 输入。  
可选择 ADC 启动方式。  
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。

### ◆ 两个模拟比较器

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: 16MHz RC (Fcpu 仅支持 16MHz、8MHz、4MHz、2MHz、1MHz、500KHz、250KHz)。  
内部低速时钟: RC 振荡器 32 KHz。

### ◆ 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 250kHz 到 16MHz。
- 软件可选的 32 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)。
- LVR 低电压复位 (检测电压有 1.7V、2.1V、3.8V 可选)。
- 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)。
- 带上拉的复位脚, 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 3.3V 时, 典型值为 600uA。  
- 频率为 31kHz、电压为 2.0V 时, 典型值为 20uA。  
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	12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)							
MDT10F685	4096	256	256	22	12	3	4	2	14	SOP14/DIP14 SOP18/DIP18 SOP20/DIP20 SOP24

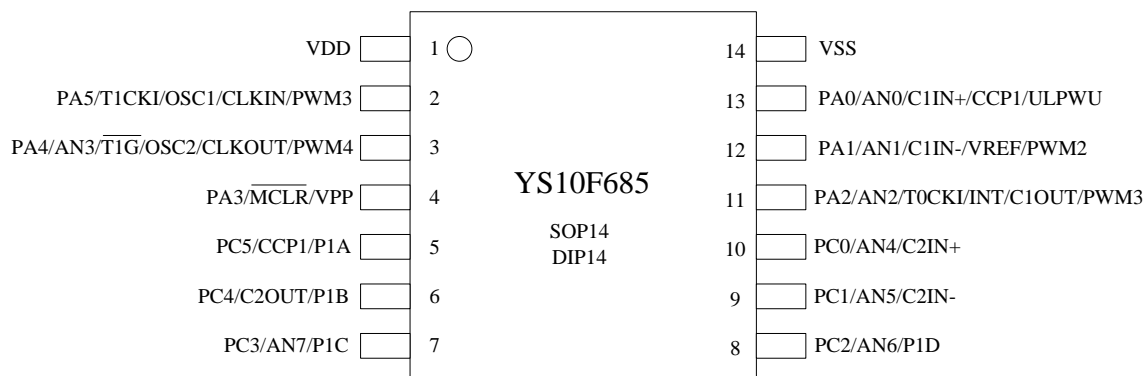
<b>1.0 器件概述</b> .....	<b>1</b>
1.2 封装脚位图.....	5
1.3 引脚说明.....	6
<b>2.0 存储器构成</b> .....	<b>8</b>
2.1 程序存储器构成.....	8
2.2 数据存储器构成.....	8
2.3 数据 EEPROM 存储器.....	18
2.4 PCL 和 PCHLAT.....	22
2.5 间接寻址、IAR 和 MSR 寄存器.....	23
2.6 看门狗定时器 (WDT).....	24
<b>3.0 复位</b> .....	<b>27</b>
3.1 概述.....	27
3.2 上电复位.....	28
3.3 MCLR.....	28
3.4 上电延时定时器 (PWRT).....	28
3.5 欠压复位.....	29
3.6 延时时序.....	30
3.7 电源控制 (PSTA) 寄存器.....	31
<b>4.0 系统时钟</b> .....	<b>32</b>
4.1 概述.....	32
4.2 振荡器控制.....	32
4.3 时钟源模式.....	33
4.4 外部时钟模式.....	34
4.5 内部时钟模式.....	37
4.6 时钟切换.....	39
4.7 双速时钟启动模式.....	40
4.8 掉电模式 (休眠).....	41
4.9 故障保护时钟监控器.....	43
<b>5.0 中断</b> .....	<b>44</b>
5.1 中断特殊功能寄存器.....	46
5.2 PA2/INT 中断.....	50
5.3 TIMER0 中断.....	51
5.4 PORTA、PORTB 电平变化中断.....	51
5.5 EEPROM 写数据中断.....	51
5.6 A/D 中断.....	51
5.7 中断的现场保护.....	51
<b>6.0 I/O 端口</b> .....	<b>52</b>

6.1	PORTA 和 CPIOA 寄存器.....	52
6.2	PORTB 和 CPIOB 寄存器.....	53
6.3	PORTC 和 CPIOC 寄存器.....	54
6.4	其它引脚功能.....	55
6.5	PORTA 引脚说明和引脚图.....	59
6.6	PORTB 引脚说明和引脚图.....	66
6.7	PORTC 引脚说明和引脚图.....	74
<b>7.0</b>	<b>定时器.....</b>	<b>78</b>
7.1	TIMER0 模块.....	78
7.2	带门控的 TIMER1 模块.....	81
7.3	TIMER2 模块.....	87
<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>96</b>
8.1	表：ECCP 模式——所需的定时器资源.....	96
8.2	寄存器：CCP1CON：增强型 CCP1 控制寄存器.....	96
8.3	捕捉模式.....	97
8.4	比较模式.....	99
8.5	PWM 模式.....	101
8.6	PWM（增强模式）.....	103
<b>9.0</b>	<b>模拟数字转换器（ADC）模块.....</b>	<b>114</b>
9.1	SAR ADC 模块设计指标.....	114
9.2	转换启动方式.....	114
9.3	寄存器说明.....	116
9.4	ADC 寄存器定义.....	117
<b>10.0</b>	<b>比较器模块.....</b>	<b>121</b>
10.1	比较器概述.....	122
10.2	比较器的配置（图）.....	125
10.3	比较器的控制.....	126
10.4	比较器响应时间.....	126
10.5	比较器中断的工作原理.....	127
10.6	休眠时的工作原理.....	128
10.7	复位的影响.....	128
10.8	比较器 C2 门控 TIMER1.....	129
10.9	比较器 C2 输出与 TIMER1 同步.....	130
10.10	比较器参考电压.....	130
<b>11.0</b>	<b>配置寄存器.....</b>	<b>132</b>
11.1	CONFIG OPTION.....	132
<b>12.0</b>	<b>指令表.....</b>	<b>134</b>
<b>13.0</b>	<b>电气特性.....</b>	<b>135</b>
13.1	绝对极限参数.....	135
13.2	直流电器特性.....	137
13.3	交流电气特性.....	138

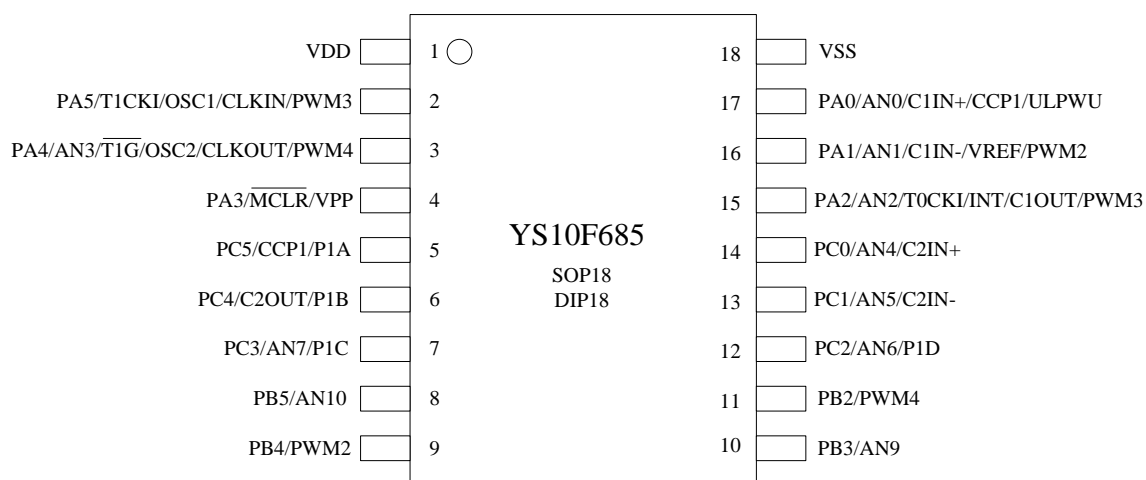
<b>14.0 开发支持</b> .....	<b>139</b>
14.1 仿真信息.....	139
14.2 烧录信息.....	140
<b>15.0 封装信息</b> .....	<b>142</b>
15.1 DIP 14 PIN.....	142
15.2 SOP 14 PIN.....	143
15.3 DIP 18 PIN.....	144
15.4 SOP 18 PIN.....	145
15.5 DIP 20 PIN.....	146
15.6 SOP 20 PIN.....	147
15.7 SOP 24 PIN.....	148
<b>16.0 汇春知识产权政策</b> .....	<b>149</b>
16.1 专利权.....	149
16.2 著作权.....	149

## 1.2 封装脚位图

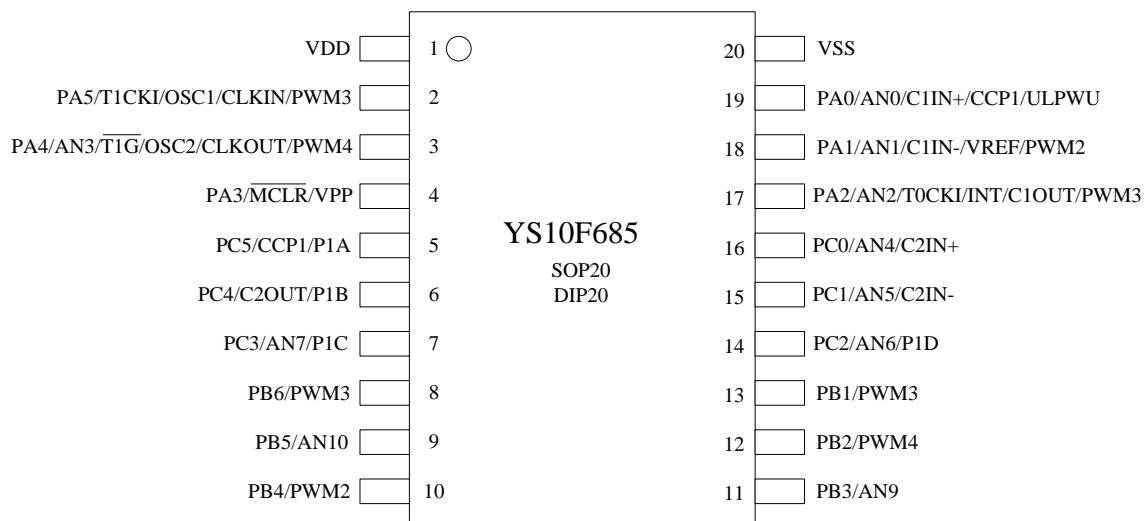
### 1.2.1 14 引脚图 (DIP、SOP)



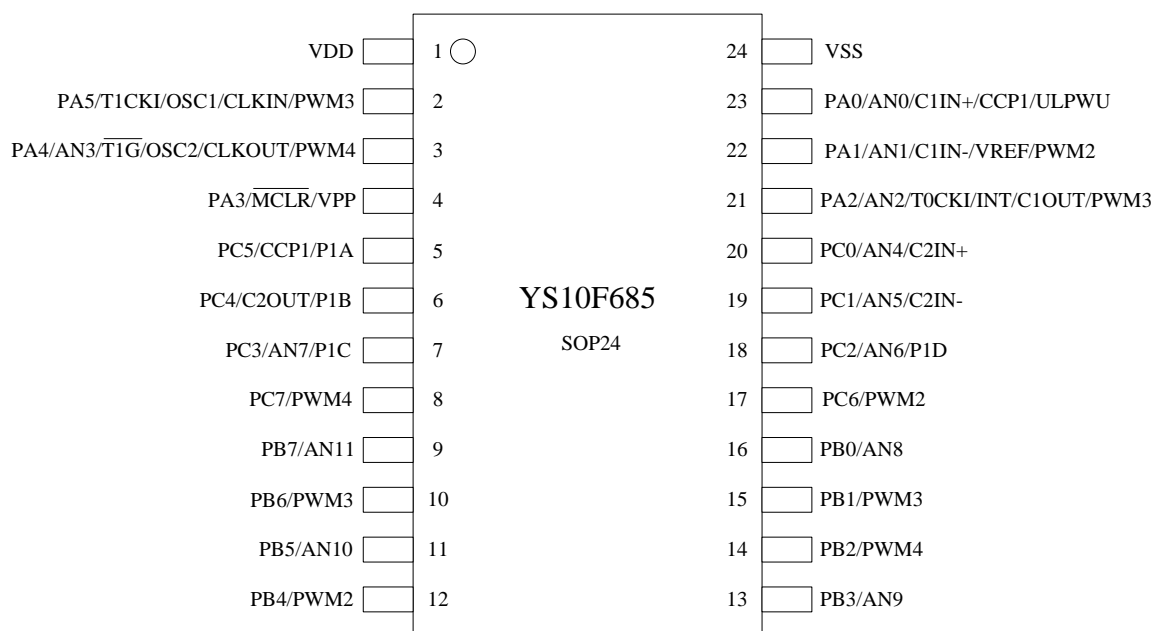
### 1.2.2 18 引脚图 (DIP、SOP)



### 1.2.3 20 引脚图 (DIP、SOP)



## 1.2.4 24 引脚图 (SOP)



## 1.3 引脚说明

名称	功能	输入类型	输出类型	说明
PA0/AN0/C1INP/IC_SCK/ULPWU/CCP1	PA0	TTL	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	AN0	AN	—	A/D通道0输入
	C1IN+	AN	—	比较器1的正相输入
	IC_SCK	ST	—	串行编程和调试时钟输入
	ULPWU	AN	—	超低功耗唤醒输入
	CCP1	ST	CMOS	捕捉模式输入/比较模式输出
PA1/AN1/C1INM/VREF/IC_SDA/PWM2	PA1	TTL	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	AN1	AN	—	A/D通道1输入
	C1IN-	AN	—	比较器1的反相输入
	VREF	AN	—	A/D外部参考电压
	IC_SDA	ST/TTL	CMOS	串行编程和调试数据输入输出
	PWM2	—	CMOS	PWM2输出
PA2/AN2/T0CKI/INT/C1OUT/PWM3	PA2	ST	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	AN2	AN	—	A/D通道2输入
	T0CKI	ST	—	Timer0时钟输入
	INT	ST	—	外部中断
	C1OUT	—	CMOS	比较器1输出
	PWM3	—	CMOS	PWM3输出
PA3/MCLR/VPP	PA3	TTL	—	带电平变化中断的PORTA输入
	MCLR	ST	—	带有内部上拉的主复位
	VPP	HV	—	编程电压
PA4/AN3/T1G/OSC2/CLKOUT/PWM4	PA4	TTL	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	AN3	AN	—	A/D通道3输入
	T1G	ST	—	Timer1门控(计数使能)
	OSC2	—	XTAL	晶振/谐振器
	CLKOUT	—	CMOS	Fosc/4输出
	PWM4	—	CMOS	PWM4输出
PA5/T1CKI/OSC1/CLKIN/PWM3	PA5	TTL	CMOS	具有可编程上拉和电平变化中断的PORTA I/O
	T1CKI	ST	—	Timer1时钟
	OSC1	XTAL	—	晶振/谐振器
	CLKIN	ST	—	外部时钟输入/RC振荡器连接

	PWM3	—	CMOS	PWM3输出
PB0/AN8	PB0	TTL	CMOS	具有可编程独立上拉和电平变化中断的GPIO
	AN8	AN	—	ADC输入通道8
PB1/PWM3	PB1	TTL	CMOS	具有可编程独立上拉和电平变化中断的GPIO
	PWM3	—	CMOS	PWM3输出
PB2/PWM4	PB2	TTL	CMOS	具有可编程独立上拉和电平变化中断的GPIO
	PWM4	—	CMOS	PWM4输出
PB3/AN9	PB3	TTL	CMOS	具有可编程独立上拉和电平变化中断的GPIO
	AN9	AN	—	ADC输入通道9
PB4/PWM2	PB2	TTL	CMOS	具有可编程独立上拉和电平变化中断的GPIO
	PWM2	—	CMOS	PWM2输出
PB5/AN10	PB5	TTL	CMOS	具有独立上拉和电平变化中断的GPIO
	AN10	AN	—	ADC输入通道10
PB6/PWM3	PB6	TTL	CMOS	具有独立上拉和电平变化中断的GPIO
	PWM3	—	CMOS	PWM3输出
PB7/AN11	PB7	TTL	CMOS	具有独立上拉和电平变化中断的GPIO
	AN11	AN	—	ADC输入通道10
PC0/AN4/C2INP	PC0	TTL	CMOS	具有可编程上拉的GPIO
	AN4	AN	—	A/D通道4输入
	C2IN+	AN	—	比较器2的正相输入
PC1/AN5/C2INM	PC1	TTL	CMOS	具有可编程上拉的GPIO
	AN5	AN	—	A/D通道5输入
	C2IN-	AN	—	比较器2的反相输入
PC2/AN6/P1D	PC2	TTL	CMOS	具有可编程上拉的GPIO
	AN6	AN	—	A/D通道6输入
	P1D	—	CMOS	PWM输出
PC3/AN7/P1C	PC3	TTL	CMOS	具有可编程上拉的GPIO
	AN7	AN	—	A/D通道7输入
	P1C	—	CMOS	PWM输出
PC4/C2OUT/P1B	PC4	TTL	CMOS	具有可编程上拉的GPIO
	C2OUT	—	CMOS	比较器2输出
	P1B	—	CMOS	PWM输出
PC5/CCP1/P1A	PC5	TTL	CMOS	具有可编程上拉的GPIO
	CCP1	ST	CMOS	捕捉输入/比较输出
	P1A	—	CMOS	PWM输出
PC6/PWM2	PC6	TTL	CMOS	具有可编程上拉的GPIO
	PWM2	—	CMOS	PWM2输出
PC7/PWM4	PC7	TTL	CMOS	具有可编程上拉的GPIO
	PWM4	—	CMOS	PWM4输出
VDD	VDD	电源	—	正电源端
VSS	VSS	电源	—	接地参考端

图注: AN = 模拟输入或输出

CMOS = CMOS 兼容输入或输出

HV = 高压

ST = 带 CMOS 电平的施密特触发器输入

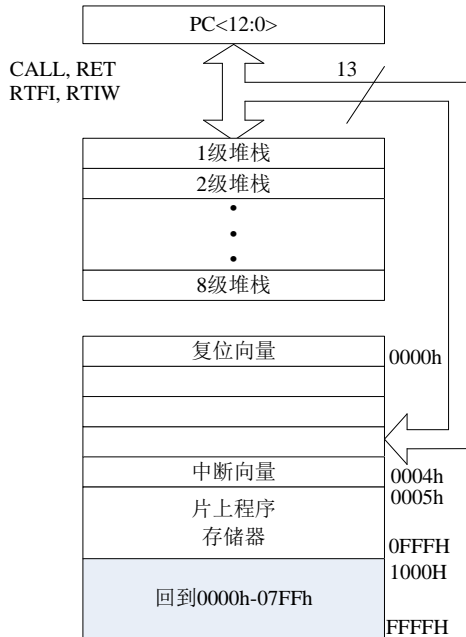
TTL = TTL 兼容输出

XTAL = 晶振

## 2.0 存储器构成

### 2.1 程序存储器构成

MDT10F685 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 MDT10F685 仅在物理上实现了第一个 4Kx14 (0000h-0FFFh) 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 4Kx14 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 数据存储器构成

数据存储器被分为 4 个存储区 (bank)，其中包含通用寄存器 (GeneralPurposeRegister, GPR) 和特殊功能寄存器 (SpecialFunctionRegister, SFR) 特殊功能寄存器位于每个存储区开头的 32 个单元。通用寄存器位于 Bank0 中的 20h-7Fh、Bank1 中的 A0h-EFh 和 Bank2 的 120h-16Fh 寄存器单元中。它们以静态 RAM 的方式实现。Bank1、Bank2 和 Bank3 中的寄存器单元 F0h-FFh 指向 Bank0 中的地址单元 70h-7Fh。所有其他的 RAM 均未实现，读取它们时将返回 0。



STATUS 寄存器的 RP1、RP0 位为存储区选择位。

00 一> 选定 Bank0

01 一> 选定 Bank1

10 一> 选定 Bank2

11 一> 选定 Bank3

## 2.2.1 通用文件寄存器

在 MDT10F685 中通用寄存器是按 256x8 的形式实现的。可直接访问每个寄存器或通过文件选择寄存器 (FileSelectRegister, FSR) 间接访问每个存储器 (见第 2.5 节“[间接寻址、IAR 和 MSR 寄存器](#)”。)

## 2.2.2 特殊功能寄存器

特殊功能寄存器是 CPU 和外设模块用来控制所需的器件操作的寄存器，这些寄存器都实现为静态 RAM。

特殊功能寄存器可分为两类：内核与外设。本章仅讲述与“内核”有关的特殊功能寄存器。那些与外设功能部件的操作有关的特殊功能寄存器将在相应的外设功能部件章节中讲述。

图 2-2: 数据存储映射

寄存器地址		寄存器地址	
IAR	00h	IAR	80h
RTCC	01h	TMR	81h
PCL	02h	PCL	82h
STATUS	03h	STATUS	83h
MSR	04h	MSR	84h
PORTA	05h	CPIOA	85h
PORTB	06h	CPIOB	86h
PORTC	07h	CPIOC	87h
	08h		88h
	09h		89h
PCHLAT	0Ah	PCHLAT	8Ah
INTS	0Bh	INTS	8Bh
PIFB1	0Ch	PIEB1	8Ch
	0Dh		8Dh
TMR1L	0Eh	PSTA	8Eh
TMR1H	0Fh	OSCCON	8Fh
T1STA	10h		90h
ADINS0	11h	ADINS1	91h
T2CON	12h		92h
CCPR1L	13h		93h
CCPR1H	14h		94h
CCP1CON	15h	PAPHR	95h
PWM1CON	16h	PAINTR	96h
ECCPAS	17h		97h
WDTCON	18h		98h
CMSTA	19h	VRSTA	99h
CMCON	1Ah	EEDATA	9Ah
ECCPR	1Bh	EEADR	9Bh
	1Ch	EECON1	9Ch
	1Dh	EECON2	9Dh
ADC0H	1Eh	ADC0L	9Eh
	1Fh		9Fh
	20h		A0h
		通用寄存器 80字节	
通用寄存器 96字节	6Fh		EFh
	70h		F0h
	7Fh	访问70h~7Fh	FFh

**BANK 0** **BANK 1**

■ 未实现的数据存储单元，读为0。

寄存器地址		寄存器地址	
IAR	100	IAR	180
	101		181
PCL	102	PCL	182
STATUS	103	STATUS	183
MSR	104	MSR	184
	105		185
PBPU	106	PBINTR	186
PBCPUEN	107		187
	108		188
	109		189
PCHLAT	10A	PCHLAT	18A
INTS	10B	INTS	18B
PIFB2	10C	PIEB2	18C
	10D		18D
T1CCPL	10E	T1CCPH	18E
	10F		18F
	110		190
TMR2L	111	TMR2H	191
PR2L	112	PR2H	192
PR3L	113	PR3H	193
PR4L	114	PR4H	194
	115		195
PWM2CON	116		196
T2PRL	117	T2PRLH	197
	118		198
EPR2	119	EPR3	199
EPR4	11A		19A
	11B		19B
	11C		19C
ADC0MX	11D	REF0CN	19D
ADC0CN	11E	ADC0CF	19E
	11F	ADC0PWR	19F
	120		1A0
		禁止访问	
通用寄存器 80字节	16F		1EF
访问70h~7Fh	170		1F0
	17F	访问70h~7Fh	1FF

**BANK 2** **BANK 3**

■ 未实现的数据存储单元，读为0。

## 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										
00H	IAR	使用 MSR 的内容对数据存储寄存器进行寻址来寻址此单元 (不是物理寄存器)								xxxx xxxx
01H	<a href="#">RTCC</a>	Timer0 模块寄存器								xxxx xxxx
02H	<a href="#">PCL</a>	程序计数器 (PC) 的低字节								0000 0000
03H	<a href="#">STATUS</a>	IRP	RP1	RP0	$\overline{T F}$	$\overline{P F}$	Z	HC	C	0001 1xxx
04H	MSR	间接数据存储寄存器地址指针								xxxx xxxx
05H	<a href="#">PORTA</a>	—	—	PA5	PA4	PA3	PA2	PA1	PA0	00xx xxxx
06H	<a href="#">PORTB</a>	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	xxxx xxxx
07H	<a href="#">PORTC</a>	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	xxxx xxxx
08H	—	未实现								—
09H	—	未实现								—
0AH	<a href="#">PCHLAT</a>	—	—	—	程序计数器高 5 位的写缓冲器					—0 0000
0BH	<a href="#">INTS</a>	GIE	PE1E	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000
0CH	<a href="#">PIFB1</a>	EEIF	ADIF	CCP1IF	C2IF	C1IF	OSFIF	TMR2IF	TMRIF	0000 0000
0DH	—	未实现								—
0EH	<a href="#">TMR1L</a>	16 位 TMR1 低字节的保持寄存器								xxxx xxxx
0FH	<a href="#">TMR1H</a>	16 位 TMR1 高字节的保持寄存器								xxxx xxxx
10H	<a href="#">T1STA</a>	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	0000 0000
11H	<a href="#">ADINS0</a>	AN11	NA	AN10	NA	AN9	NA	NA	AN8	1010 1001
12H	<a href="#">T2CON</a>	T2MOD	TMR2CS	P4EN	P3EN	P2EN	TMR2ON	T2CKPS1	T2CKPS0	0000 0000
13H	<a href="#">CCPR1L</a>	捕捉/比较/PWM 寄存器 1 的低字节								0000 0000
14H	<a href="#">CCPR1H</a>	捕捉/比较/PWM 寄存器 1 的高字节								0000 0000
15H	<a href="#">CCP1CON</a>	P1M1	P1M0	TICS	ECCP_CS	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000
16H	<a href="#">PWM1CON</a>	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000
17H	<a href="#">ECCPAS</a>	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000
18H	<a href="#">WDTCN</a>	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	—0 1000
19H	<a href="#">CMSTA</a>	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000
1AH	<a href="#">CMCON</a>	—	—	—	—	—	TIMER0_ON	T1GSS	C2SYNC	----010
1BH	<a href="#">ECCPR</a>	TMR1 做 PWM 时占空比高 4 位				TMR1 做 PWM 时占空比重载值高 4 位				0000 0000
1CH	—	未实现								—
1DH	—	未实现								—
1EH	<a href="#">ADCOH</a>	ADC 结果寄存器高字节								0000 0000
1FH	—	未实现								—

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

## 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										
80H	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 xxxx
81H	TMR	PAPH	IES	TCS	TCE	PSC	PS2	PS1	PS0	1111 1111
82H	PCL	程序计数器 (PC) 的低字节								0000 0000
83H	STATUS	IRP	RP1	RP0	$\overline{TF}$	$\overline{PF}$	Z	HC	C	0001 1xxx
84H	MSR	间接数据存储器地址指针								xxxx xxxx
85H	CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	—11 1111
86H	CPIOB	CPIOB7	CPIOB6	CPIOB5	CPIOB4	CPIOB3	CPIOB2	CPIOB1	CPIOB0	1111 1111
87H	CPIOC	CPIOC7	CPIOC6	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	1111 1111
88H	—	未实现								—
89H	—	未实现								—
8AH	PCHLAT	—	—	—	程序计数器高 5 位的写缓冲器					—0 0000
8BH	INTS	GIE	PE1E	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000
8CH	PIEB1	EEIE	ADIE	CCP1IE	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000
8DH	—	未实现								—
8EH	PSTA	—	—	ULPWUE	SBOREN	MCR	IER	POR	BOR	0001 100x
8FH	OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS	0101 1000
90H	—	未实现								—
91H	ADINS1	AN7	AN6	AN5	AN3	AN4	AN2	AN1	AN0	1111 1111
92H	—	未实现								—
93H	—	未实现								—
94H	—	未实现								—
95H	PAPHR	—	—	PHA5	PHA4	—	PHA2	PHA1	PHA0	—11 -111
96H	PAINTR	—	—	PAINTR5	PAINTR4	PAINTR3	PAINTR2	PAINTR1	PAINTR0	—00 0000
97H	—	未实现								—
98H	—	未实现								—
99H	VRSTA	VREN	—	VRR	—	VR3	VR2	VR1	VR0	0-0- 0000
9AH	EEDATA	EEDATA7	EEDATA6	EEDATA5	EEDATA4	EEDATA3	EEDATA2	EEDATA1	EEDATA0	0000 0000
9BH	EEADR	EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	0000 0000
9CH	EECON1	—	—	—	—	WRERR	WREN	WR	RD	---- x000
9DH	EECON2	EEPROM 控制寄存器 2 (非物理寄存器)								---- ----
9EH	ADC0L	ADC 结果寄存器低字节								0000 0000
9FH	—	未实现								—

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

## 2.2.2.3 特殊功能寄存器位定义 (BANK2)

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR 和 BOR 的值
Bank 2										
100H	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 xxxx
101H	TMR	PAPH	IES	TCS	TCE	PSC	PS2	PS1	PS0	1111 1111
102H	PCL	程序计数器 (PC) 的低字节								0000 0000
103H	STATUS	IRP	RP1	RP0	$\overline{TF}$	$\overline{PF}$	Z	HC	C	00011xxx
104H	MSR	间接数据存储器地址指针								xxxx xxxx
105H		未实现								
106H	PBPUR	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0	1111 1111
107H	PBCPUEN	PBPUEN	PCPUEN							1100 0000
108H	—	未实现								—
109H	—	未实现								—
10AH	PCHLAT	—	—	—	程序计数器高 5 位的写缓冲器				—	00000
10BH	INTS	GIE	PE1E	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000
10CH	PIFB2								PBIF	0000 000x
10DH	—	未实现								—
10EH	T1CCPL	Timer1 在 PWM 时周期的低 8 位								0000 0000
10FH		未实现								
110H		未实现								
111H	TMR2L	16 位 Timer2 定时/计数低字节寄存器								xxxx xxxx
112H	PR2L	PWM2 占空比重载寄存器低字节								0000 0000
113H	PR3L	PWM3 占空比重载寄存器低字节								
114H	PR4L	PWM4 占空比重载寄存器低字节								
115H		未实现								
116H	PWM2CON	P4CS1	P4CS0	P3CS1	P3CS0	P2CS1	P2CS0	DUMMY	DUMMY	0000 0010
117H	T2PRLL	PWM2/3/4 周期寄存器的低字节								0000 0000
118H	—	未实现								—
119H	EPR2	PWM2 占空比重载寄存器高 4 位 (从动)				PWM2 占空比重载寄存器高 4 位				0000 0000
11AH	EPR4	PWM4 占空比重载寄存器高 4 位 (从动)				PWM4 占空比重载寄存器高 4 位				0000 0000
11BH		未实现								
11CH		未实现								
11DH	ADC0MUX	NA	NA	NA	NA	MUX3	MUX2	MUX1	MUX0	0000 0000
11EH	ADC0CN	ADC0EN	AD12BEN	AD0CLK	AD0BUSY	AD0GN	NA	AD0CM1	AD0CM0	0000 0000
11FH		未实现								

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

## 2.2.2.4 特殊功能寄存器位定义 (BANK3)

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR 和 BOR 的值
Bank 3										
180H	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 xxxx
181H	TMR	PAPH	IES	TCS	TCE	PSC	PS2	PS1	PS0	1111 1111
182H	PCL	程序计数器 (PC) 的低字节								0000 0000
183H	STATUS	IRP	RP1	RP0	$\overline{TF}$	$\overline{PF}$	Z	HC	C	0001 1xxx
184H	MSR	间接数据存储器地址指针								xxxx xxxx
185H		未实现								
186H	PBINTR	PBINTR7	PBINTR6	PBINTR5	PBINTR4	PBINTR3	PBINTR2	PBINTR1	PBINTR0	0000 0000
187H		未实现								
188H	—	未实现								—
189H	—	未实现								—
18AH	PCHLAT	—	—	—	程序计数器高 5 位的写缓冲器				—0 0000	
18BH	INTS	GIE	PE1E	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000
18CH	PIEB2								PBIE	0000 0000
18DH	—	未实现								—
18EH	T1CCPH	NA	NA	NA	NA	Timer1 在做 PWM 时周期的高 4 位			0000 0000	
18FH		未实现								
190H	—	未实现								
191H	TMR2H	16 位 Timer2 定时/计数高字节寄存器								xxxx xxxx
192H	PR2H	PWM2 占空比寄存器低字节 (从动)								0000 0000
193H	PR3H	PWM3 占空比寄存器低字节 (从动)								0000 0000
194H	PR4H	PWM4 占空比寄存器低字节 (从动)								0000 0000
195H		未实现								
196H		未实现								
197H	T2PRLH	PWM2/3/4 周期寄存器的高字节								0000 0000
198H	—	未实现								—
199H	EPR3	PWM3 占空比寄存器重载寄存器 (从动)				PWM3 占空比寄存器重载寄存器				0000 0000
19AH		未实现								
19BH		未实现								
19CH		未实现								
19DH	REF0CN	NA	NA	NA	DUMMY	REFSL1	REFSL0	IREFSL1	IREFSL0	0000 0100
19EH	ADC0CF	NA	NA	NA	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0001 1110
19FH	ADCPWR	SJST	LPM	PWR5	PWR4	PWR3	PWR2	PWR1	PWR0	0001 1111

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

### 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
- 数据存储器（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。欲知其他不会影响任何状态位的指令，请参见第 12.0 节“指令集汇总”。

#### 寄存器 03H/83H/183H：状态寄存器（STATUS）

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
<b>IRP</b>	<b>RP1</b>	<b>RP0</b>	<b>TF</b>	<b><math>\overline{\text{PF}}</math></b>	<b>Z</b>	<b>HC</b>	<b>C</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	<b>IRP</b> ：间接寻址最高位： 1 = 间接寻址第2、3页 0 = 间接寻址第0、1页
bit6 bit5	<b>RP1RP0</b> ：寄存器存储区选择位（用于直接寻址） 00 = 直接寻址第0页 01 = 直接寻址第1页 10 = 直接寻址第2页 11 = 直接寻址第3页
bit4	<b>TF</b> ：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	<b><math>\overline{\text{PF}}</math></b> ：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	<b>Z</b> ：零标志位 1 = 算术运算或逻辑运算的结果为零 0 = 算术运算或逻辑运算的结果不为零
bit1	<b>HC</b> ：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 (减法时, 没有发生借位时为1)  
 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

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
—	—	<b>ULPWUE</b>	<b>SBOREN</b>	<b>MCR</b>	<b>IER</b>	<b>POR</b>	<b>BOR</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

**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

bit1

**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<1:0> = 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。MDT10F685 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

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

### 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 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 得到的是不确定值。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

bit7~4 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           ;将要读取的地址送地址寄存器
BSR    EECON1,RD       ;发送读命令
LDR    EEDATA,W        ;读取结果送 W 寄存器返回

```

### 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           ;写地址送地址寄存器
BCR    STATUS,PAGE     ;bank 0
LDR    data_buff,W     ;
BSR    STATUS,PAGE     ;BANK1
STWR   EEDATA         ;将要写的的数据送数据寄存器
BSR    EECON1,WREN     ;使能写操作
BCR    INTS,GIE       ;关全局中断
BTSC   INTS,GIE       ;
LJUMP  $-2            ;
LDWI   0X55           ;下面四行为固定写法，必须这样写
STWR   EECON2         ;
LDWI   0XAA           ;
STWR   EECON2         ;
BSR    EECON1,WR      ;写操作
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      ;使能全局中断

```

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，固定写法的四行语句必须禁止中断，否则可能带来不可预测结果。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	;上次读取的数据保存到 W
BSR	EECON1,RD	;读 EEPROM
XORWR	EEDATA,W	;比较两次读结果
BCR	STATUS,PAGE	
BTSS	STATUS,Z	;是否相等
LJUMP	WRITE_ERR	;不相等, 转出错处理
:		;相等, 继续

### 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 CPDB 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								---- ----	---- ----

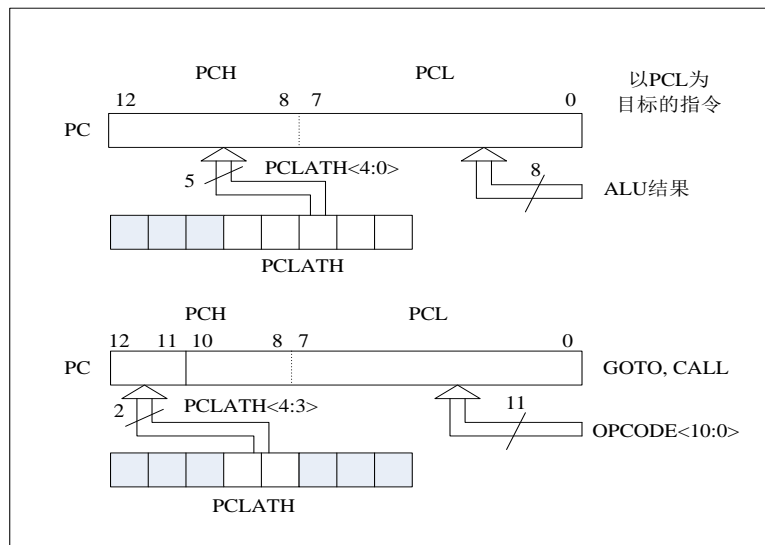
图注: x = 未知, u = 不变, — = 未实现 (读为 0), q = 取值视情况而定。  
数据 EEPROM 模块不使用阴影单元。

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

MDT10F685 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/182H: 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/18AH: 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。

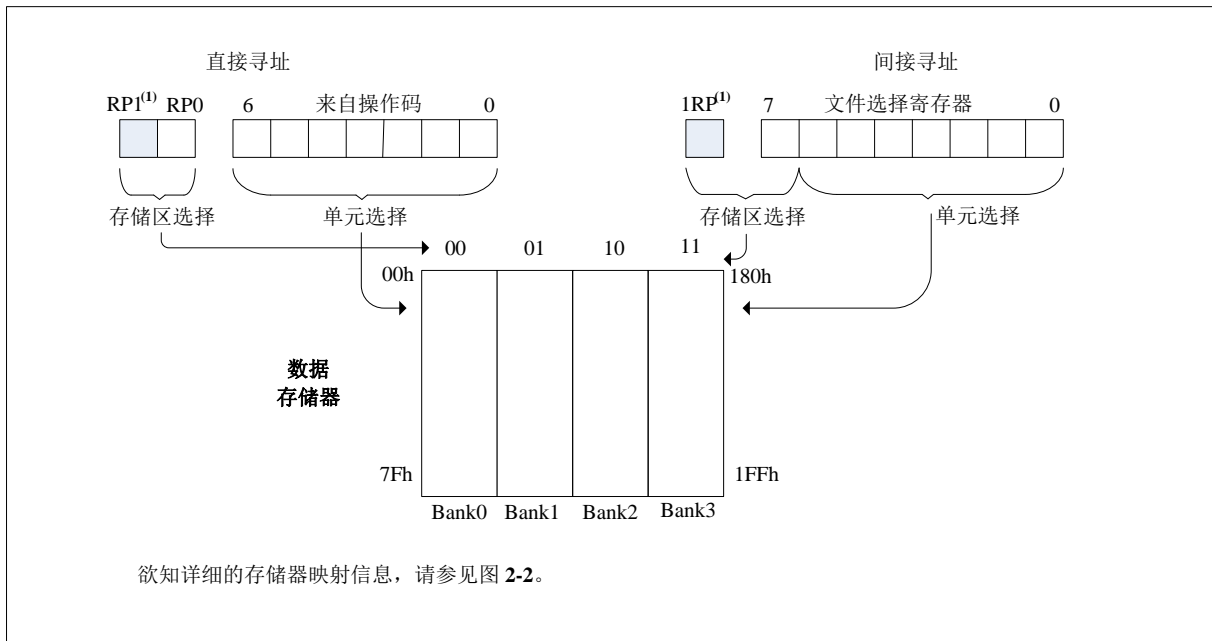
例 2-5: 间接寻址

```

;功能: 地址 20 开始的连续 16 个字节的 RAM 清 0
LDWI    0X20    ;
STWR    MSR     ;地址指针赋初值
NEXT:   CLRR    IAR    ;清除 MSR 值对应的寄存器值
        INCR    MSR,R  ;递增地址指针
        BTSS    MSR,4  ;完成没有?
        LJUMP   NEXT   ;没完成, 继续
CONTINUE:                ;完成了

```

图 2-5: MDT10F685 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 (32kHz)
- 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 32kHz 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）。

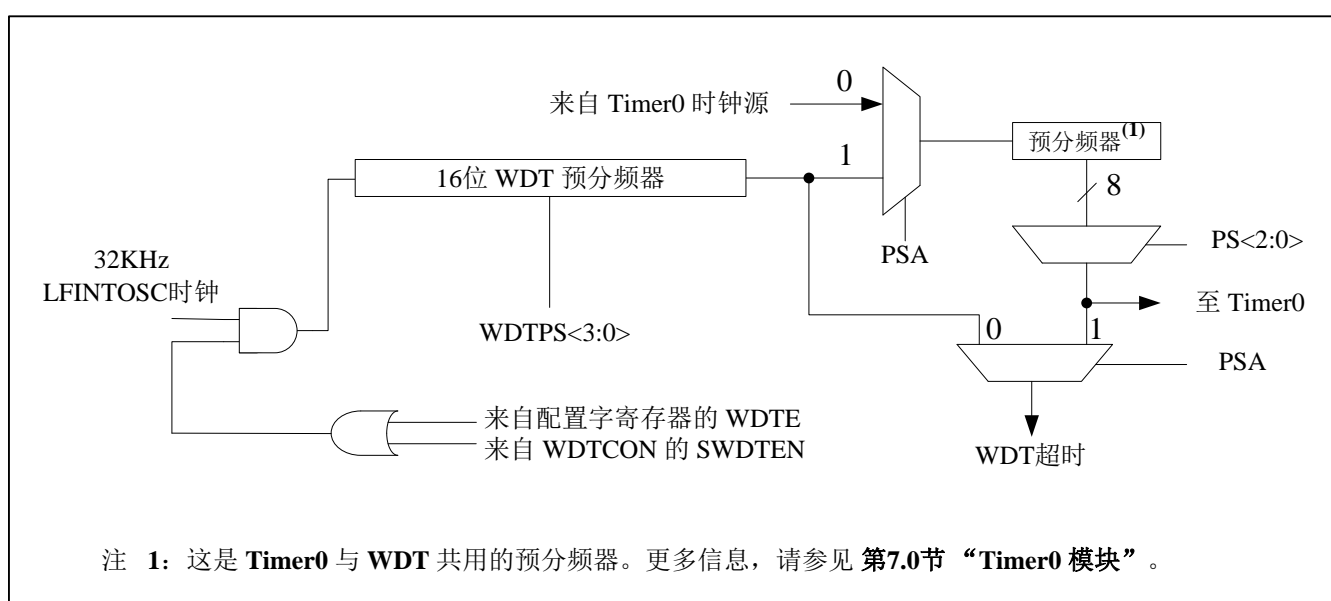
## 2.6.2 WDT 控制

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 MDT10F685 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

MDT10F685 有以下几种不同类型的复位：

- 上电复位（POR）
- 正常工作期间的 WDT 复位
- 休眠期间的 WDT 复位
- 休眠期间的  $\overline{\text{MCLR}}$  复位
- 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
- $\overline{\text{MCLR}}$  复位
- 休眠期间的  $\overline{\text{MCLR}}$  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。

图 3-1 给出了片上复位电路的简化框图。 $\overline{\text{MCLR}}$  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

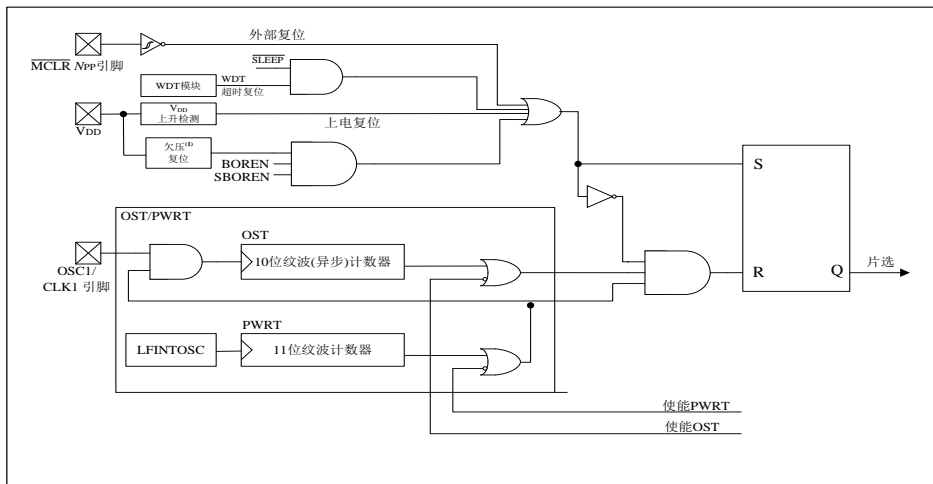


表 3-1：STATUS/PSTA 位及其含义

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 $\overline{\text{MCLR}}$ 复位
U	u	1	0	休眠期间的 $\overline{\text{MCLR}}$ 复位

图注： u = 不变， x = 未知

## 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

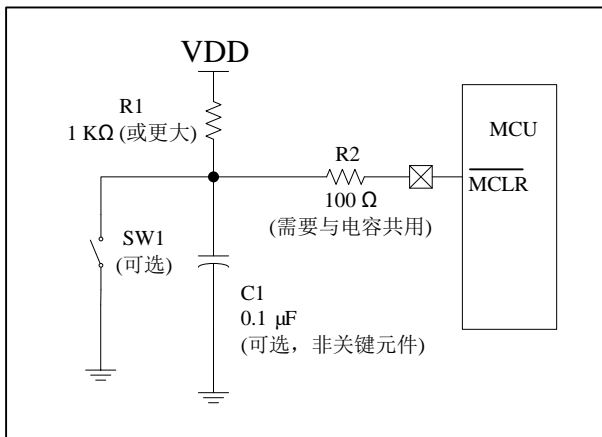
MDT10F685 在  $\overline{\text{MCLR}}$  复位路径中有一个噪声滤波器。该滤波器检测并滤除小脉冲。

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 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 32kHz。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。

### 3.5 欠压复位

配置字寄存器中的  $\text{BOREN0}$  和  $\text{BOREN1}$  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  $\text{PSTA}$  寄存器的  $\text{SBOREN}$  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， $\text{SBOREN}$  位被禁止。关于配置字的定义，请参见寄存器  $\text{PSTA}$ 。

如果 VDD 下降到  $\text{VBOR}$  以下，且持续时间超过参数值（ $\text{TBOR}$ ）（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  $\text{VBOR}$  的时间少于参数值（ $\text{TBOR}$ ），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  $\text{VBOR}$  以上（见图 3-5）。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  $\text{VBOR}$  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  $\text{VBOR}$  以上，上电延时定时器将执行一段 64ms 的复位。

图 3-5: 欠压情形

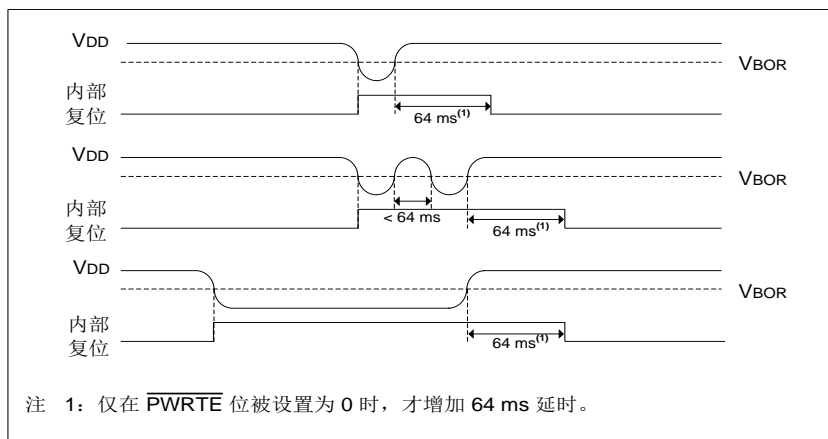


表 3-5: 与欠压有关的寄存器汇总

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	MCR	IER	POR	BOR	0001 100x	0001 100x
STATUS	IRP	PR1	PAGE	$\overline{T F}$	$\overline{P F}$	Z	HC	C	0001 1xxx	0001 1xxx

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。

### 3.6 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  $\overline{PWRT E}$  位的状态。例如, 在 EC 模式且  $\overline{PWRT E}$  位被擦除 (PWRT 禁止) 的情况下, 根本不会出现延时。图 3-6-1、3-6-2 和 3-6-3 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 3-6-2)。这对于测试或同步多个并行工作的 MDT10F685 器件来说是非常有用的。

表 3-6-1 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 3-6-1: 上电时的超时时序 (MCLR 延时): 情形 1

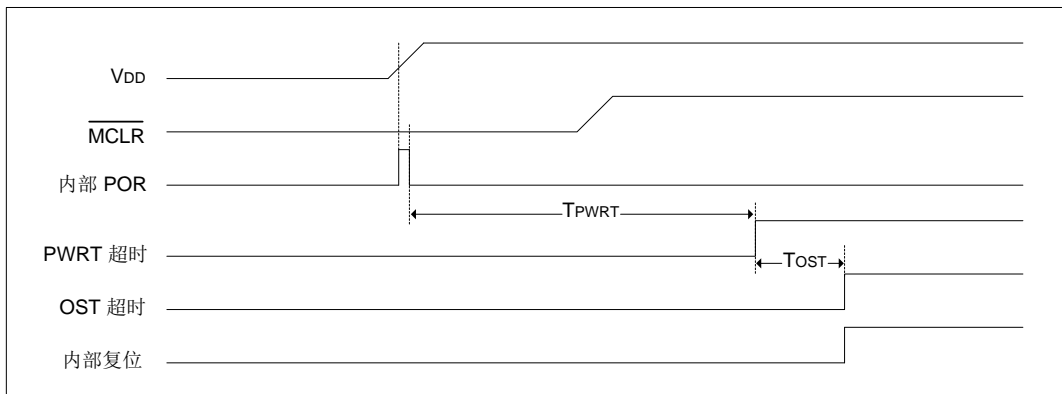


图 3-6-2: 上电时的超时时序 (MCLR 延时): 情形 2

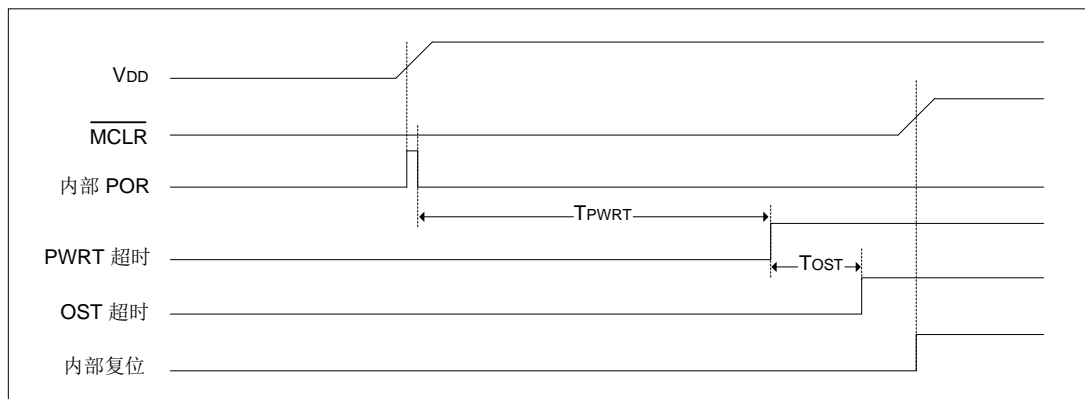


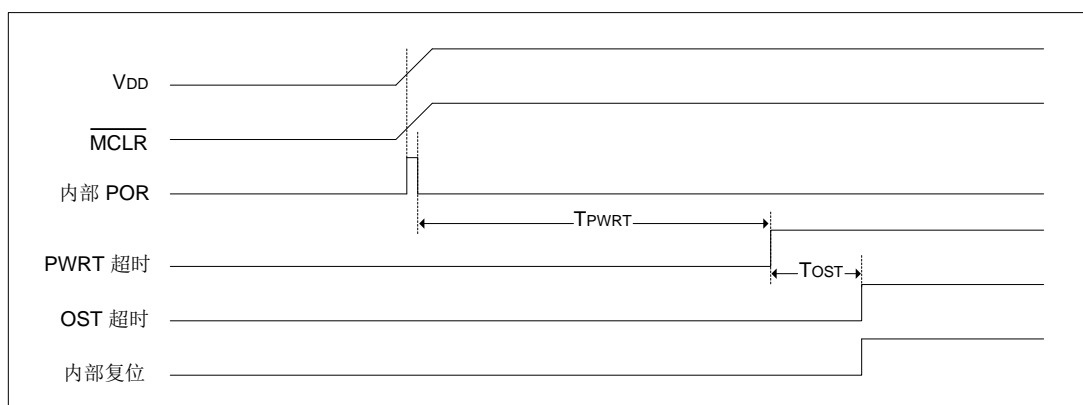
图 3-6-3: 上电时的超时时序 ( $\overline{\text{MCLR}}$  及 VDD)

表 3-6-1: 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 $\overline{\text{MCLR}}$ 复位	0000h	000u uuuu	--0u --uu
休眠期间的 $\overline{\text{MCLR}}$ 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 3-6-2: 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
	$\overline{\text{PWRTE}} = 0$	$\overline{\text{PWRTE}} = 1$	$\overline{\text{PWRTE}} = 0$	$\overline{\text{PWRTE}} = 1$	
XT, HS, LP	$T_{\text{PWRT}} + 1024 * T_{\text{OSC}}$	$1024 * T_{\text{OSC}}$	$T_{\text{PWRT}} + 1024 * T_{\text{OSC}}$	$1024 * T_{\text{OSC}}$	$1024 * T_{\text{OSC}}$
RC, EC, INTOSC	$T_{\text{PWRT}}$	—	$T_{\text{PWRT}}$	—	—

### 3.7 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, 并在随后的复位发生时检查 BOR 是否为 0, 如果是, 则表示已发生欠压复位。当禁止欠压复位电路 (配置字寄存器中的  $\text{BOREN}<1:0> = 00$ ) 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)。

更多信息, 请参见[第 6.4.4 节“超低功耗唤醒”](#)和[第 3.5 节“欠压复位 \(BOR\)”](#)。

## 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：

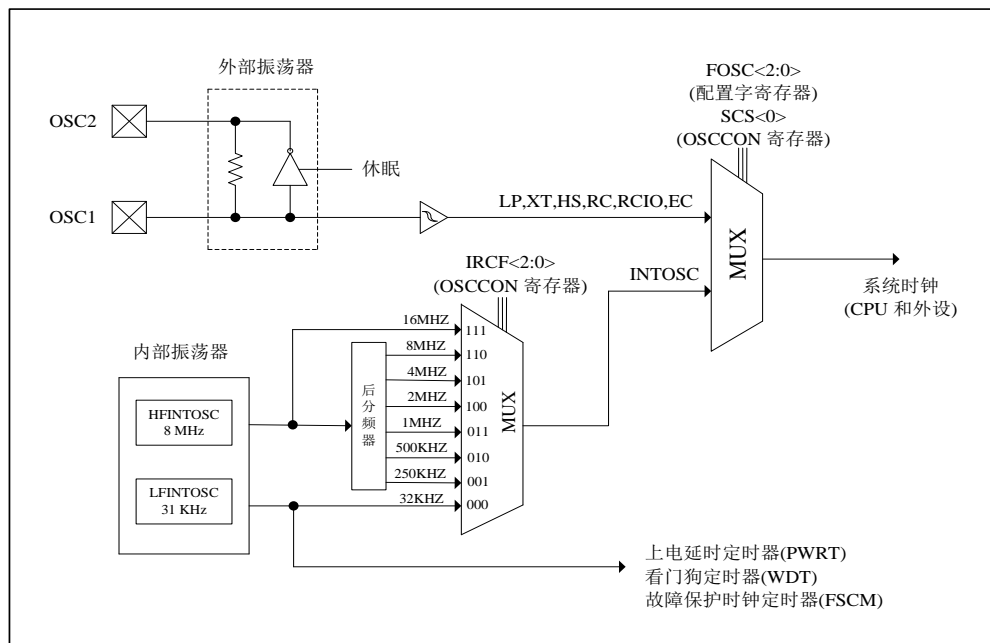
- 通过软件选择外部或内部系统时钟源。
- 双速启动模式，使外部振荡器从启动到代码执行间的延时达到最小。
- 故障保护时钟监视器（FSCM）旨在检测外部时钟源的故障（LP、XT、HS、EC 或 RC 模式）并自动切换到内部振荡器。

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 PA4 上，时钟从 PA5 输入。
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. RC+IO——外部阻容，I/O 在 PA4 上。
7. IRC——内部振荡器，PA4 输出 1/4 系统时钟频率，PA5 为 I/O PIN。
8. IRC+IO——内部振荡器，PA4、PA5 作为 I/O PIN。

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-0	R/W-1	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 = 32kHz

001 = 250kHz

010 = 500kHz

011 = 1MHz

100 = 2MHz

101 = 4MHz (缺省值)

110 = 8MHz

111 = 16MHz

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——16MHz到250kHz) 状态位

1 = HFINTOSC稳定

0 = HFINTOSC不稳定

bit1 **LTS**: LFINTOSC (低频——32kHz) 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: 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 (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 16MHz 高频内部振荡器

(HFINTOSC)，另一个是 32kHz 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

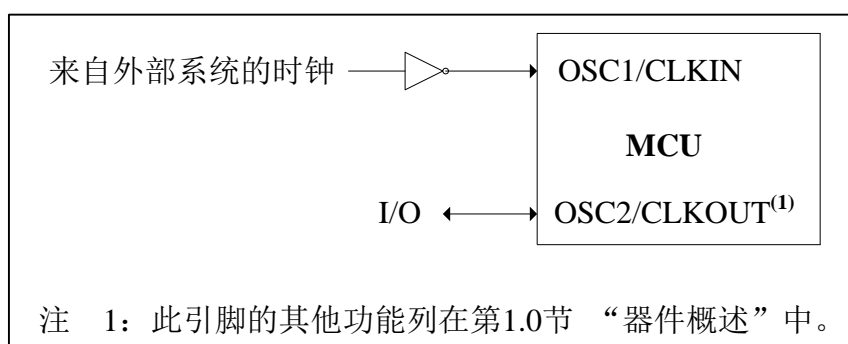
切换自	切换到	频率	振荡器延时
休眠/POR	LFINTOSC HFINTOSC	32kHz 250kHz到16MHz	振荡器预热延时 ( $T_{WARM}$ )
休眠/POR	EC, RC	DC—20MHz	双周期
LFINTOSC (32kHz)	EC, RC	DC—20MHz	每次一周
休眠/POR	LP, XT, HS	32kHz到20MHz	1024个时钟周期 (OST)
LFINTOSC (32kHz)	HFINTOSC	250kHz到16MHz	1us (近似值)

### 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。
  - 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）

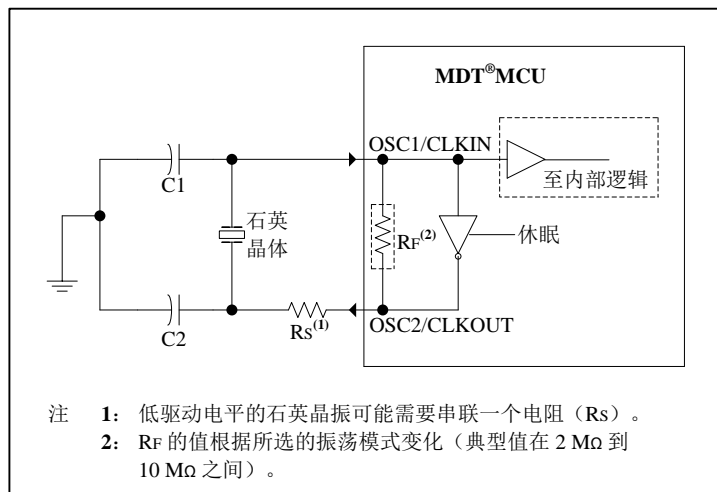
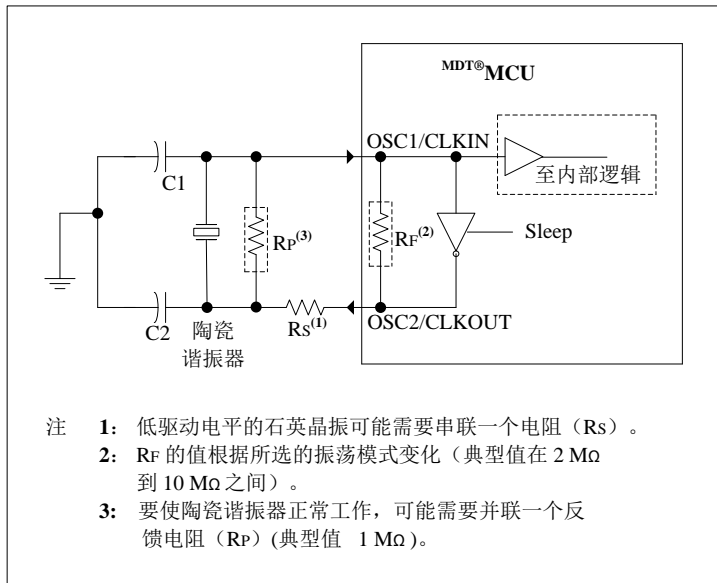


图 4-4-3-2: 陶瓷谐振器的工作原理（XT 或 HS 模式）

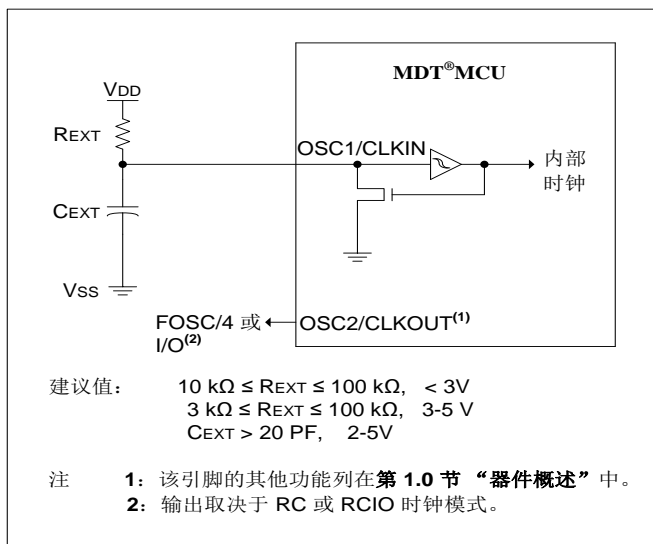


#### 4.4.4 外部 RC 模式

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。该信号可用来为外部电路、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 16MHz，精度范围为±1%。
2. LFINTOSC（低频内部振荡器）未经校准，工作频率为 32kHz。

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 16MHz 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≠000 选择 16MHz 到 250kHz 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 32kHz 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 32kHz。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 32kHz（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）

16MHz HFINTOSC 和 32kHz 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：

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

注：任何复位后，OSCCON寄存器的IRCF<2:0>位将被置为101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：

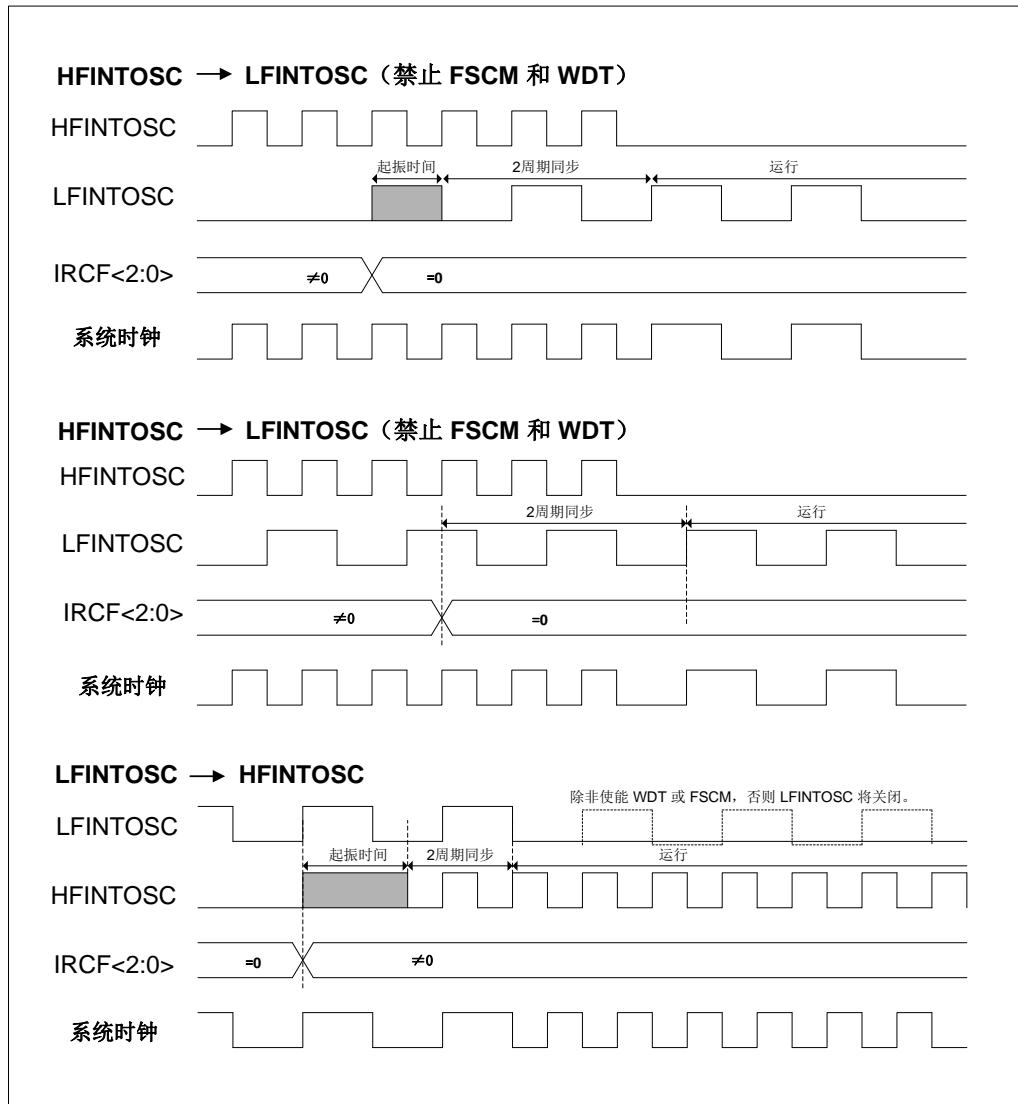
1. OSCCON 寄存器的 IRCF<2:0>位被修改。
2. 如果新时钟是关闭的，开始一个时钟启动延时。
3. 时钟切换电路等待当前时钟下降沿的到来。
4. CLKOUT 保持为低，时钟切换电路等待新时钟上升沿的到来。
5. 现在 CLKOUT 连接到新时钟。OSCCON 寄存器的 HTS 和 LTS 位按要求被更新。
6. 时钟切换完成。

更多信息请参见[图 4-1](#)。

如果选取的内部振荡器速度在 16MHz 到 250kHz 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

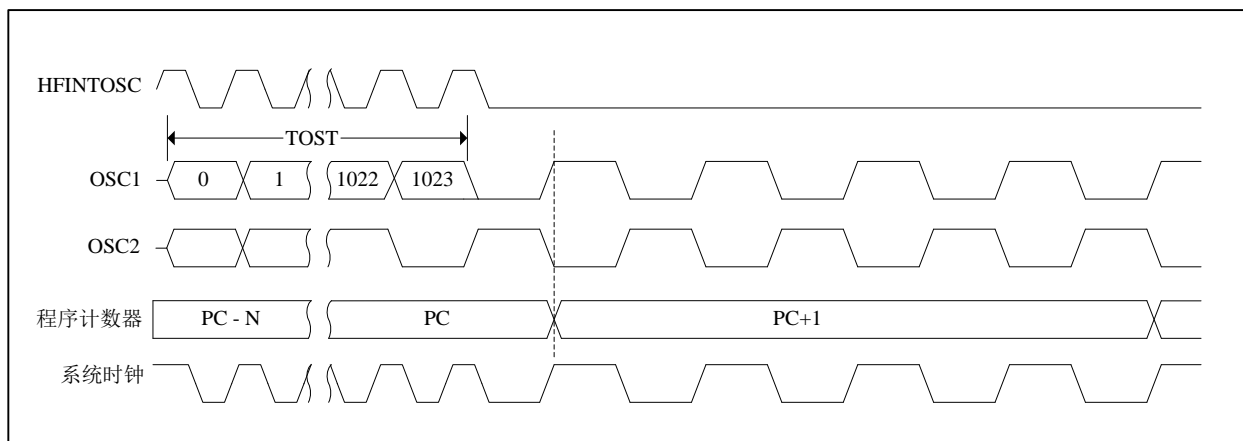
1. 从上电复位或休眠中唤醒。
2. 使用内部振荡器以 OSCCON 寄存器的 IRCF<2:0>位设置的频率开始执行指令。
3. OST 使能，计数 1024 个时钟周期。
4. OST 超时，等待内部振荡器下降沿的到来。
5. OSTS 置 1。
6. 系统时钟保持为低，直到新时钟下一个下降沿的到来（LP、XT 或 HS 模式）。
7. 系统时钟切换到外部时钟源。



### 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 从休眠状态唤醒

由于在休眠期间没有片上时钟处于工作状态，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. PORTA、PORTB 电平变化中断
7. 来自 INT 引脚的外部中断

## 8. 看门狗定时器唤醒（如果 WDT 使能）

MCLR 引脚上的有效信号会导致器件复位。其他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。

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

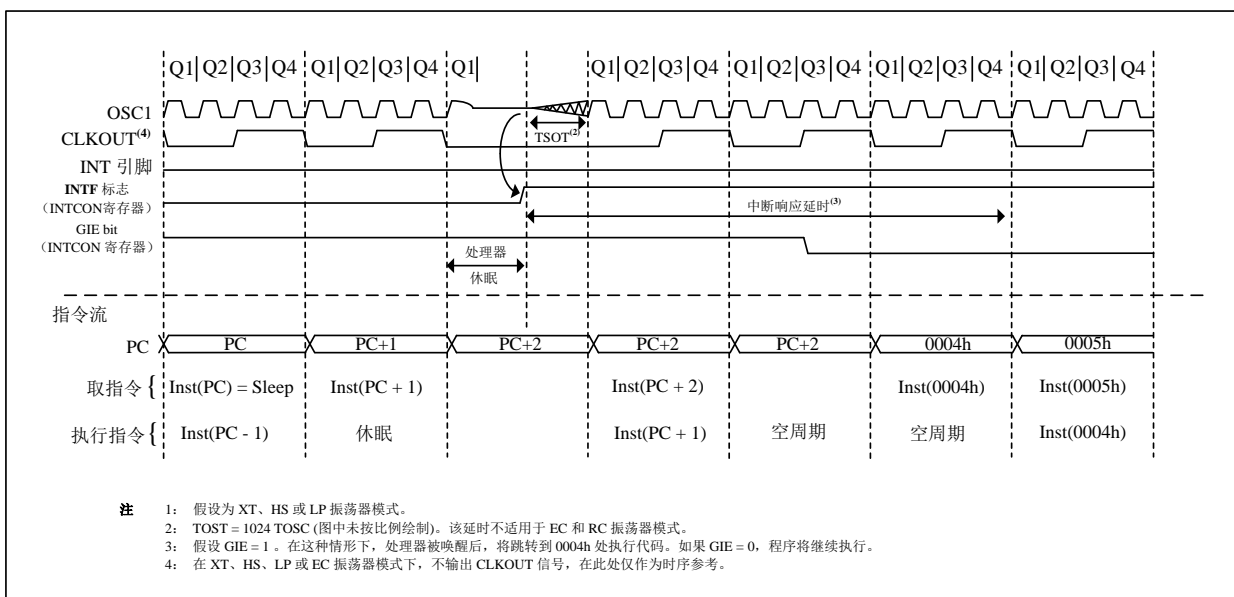
当禁止全局中断（GIE 被清零）时，并且有任一中断源将其中断允许位和中断标志位置 1，将会发生下列事件之一：

- 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。

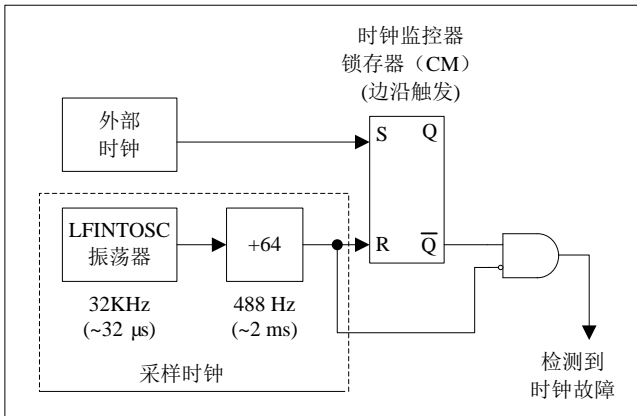
图 4-8：通过中断将器件从休眠状态唤醒



## 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

MDT10F685 有以下多种中断源：

- 外部中断 PA2/INT
- Timer0 溢出中断
- PORTA、PORTB 电平变化中断
- 两个比较器中断
- A/D 中断
- Timer1 溢出中断
- Timer2 溢出中断
- 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（PIFB1、PIFB2）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 和 PIFB2 寄存器中。相应的中断允许位在 PIEB1 和 PIFB2 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

PIFB2 寄存器只包含 PORTB 电平变化中断标志位。

对于外部中断事件，如 INT 引脚或 PORTA、PORTB 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

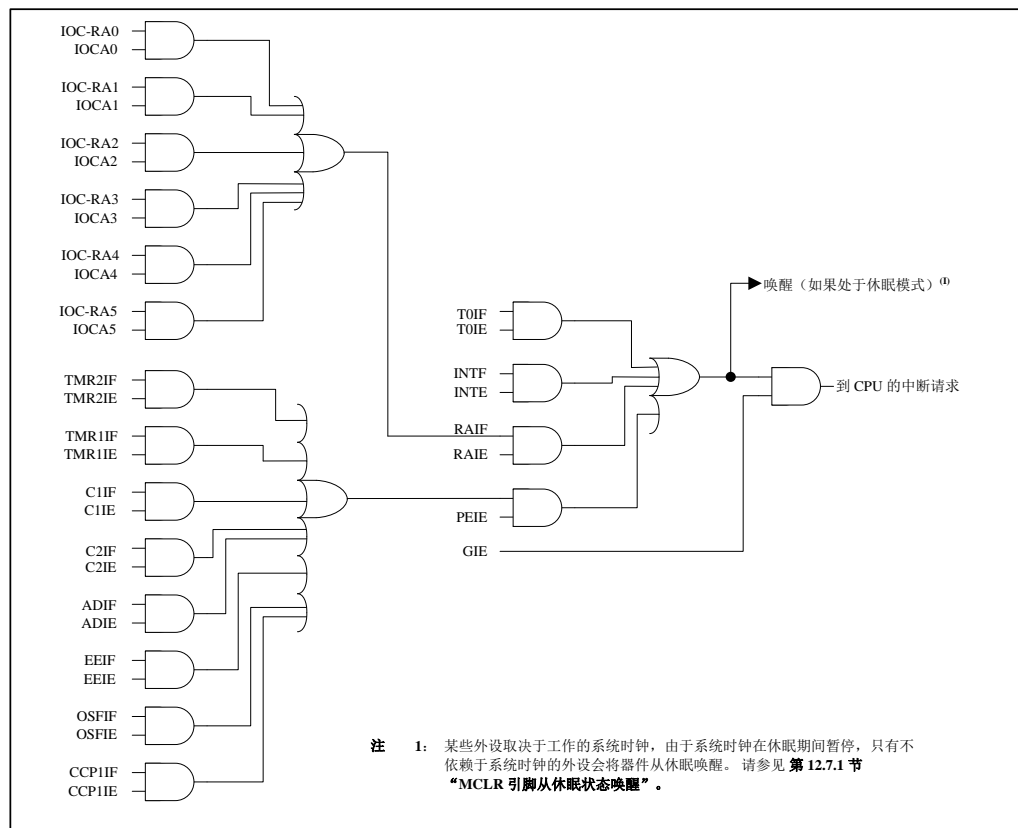
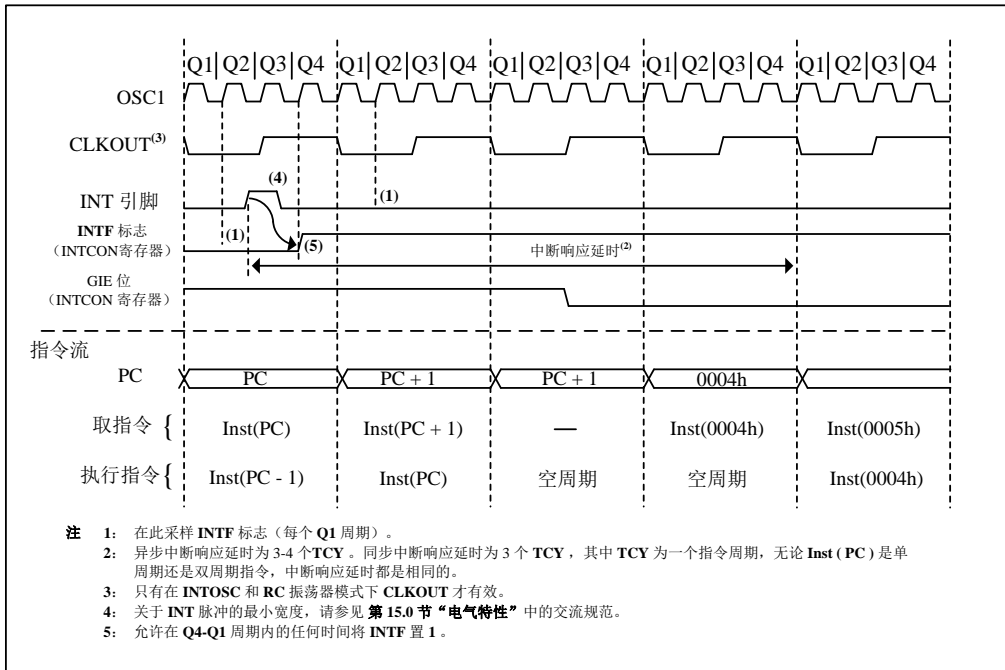


图 5-1-2: INT 引脚中断时序



## 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/10BH/18BH: 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
<b>EEIE</b>	<b>ADIE</b>	<b>CCP1IE</b>	<b>C2IE</b>	<b>C1IE</b>	<b>OSFIE</b>	<b>TMR2IE</b>	<b>TMR1IE</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 **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	<b>C1IE</b> : 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	<b>OSFIE</b> : 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	<b>TMR2IE</b> : Timer2溢出中断允许位 1 = 允许Timer2溢出中断 0 = 禁止Timer2溢出中断
bit0	<b>TMR1IE</b> : 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 PIEB2 寄存器

PIEB2 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 18CH: 外设中断允许寄存器1 (PIEB2)

-	-	-	-	-	-	-	R/W-0
							<b>PBIE</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	保留位
bit6	保留位
bit5	保留位
bit4	保留位
bit3	保留位
bit2	保留位
bit1	保留位
bit0	<b>PBIE</b> : PORTB电平变化中断允许位 1 = 允许PORTB端口电平变化产生中断 0 = 禁止PORTB端口电平变化产生中断



## 5.1.4 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	<b>EEIF</b> : EEPROM写操作中中断标志位 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	<b>ADIF</b> : 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	<b>CCP1IF</b> : 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	<b>C2IF</b> : 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	<b>C1IF</b> : 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	<b>OSFIF</b> : 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	<b>TMR2IF</b> : Timer2溢出中断标志位 1 = 发生了Timer2溢出 (必须用软件清零) 0 = 未发生Timer2溢出
bit0	<b>TMR1IF</b> : 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

## 5.1.5 PIFB2 寄存器

PIFB2 寄存器包含外设中断允许位。

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 10CH: 外设中断允许寄存器1 (PIEB2)

-	-	-	-	-	-	-	-	R/W-0
								<b>PBIE</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 保留位

bit6 保留位

bit5 保留位

bit4 保留位

bit3 保留位

bit2 保留位

bit1 保留位

bit0 **PBIE**: PORTB电平变化中断标志位

1 = PORTB端口电平改变

0 = PORTB端口电平没有改变

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	TMRIF	0000 0000	0000 0000
PIFB2								PBIF		
PIEB1	EEIE	ADIE	CCP1IE	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIEB2								PBIE	0000 0000	0000 0000
PBINTR	PBINTR7	PBINTR6	PBINTR5	PBINTR4	PBINTR3	PBINTR2	PBINTR1	PBINTR0	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、PORTB 电平变化中断

PORTA、PORTB 输入电平的变化会使的 PAIF 与 PBIF 位置 1。可以通过置 1/清零的 PAIE、PBIE 位来使能/禁止该中断。此外，可通过 PAINTR、PBINTR 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，PIFB1 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： MDT10F685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           ; 保存 W 寄存器到 W_TEMP
SWAPR  STATUS, W        ; 将 STATUS 寄存器高低四位交换后保存到 W（目的：不改变标志）
STWR   STATUS_TEMP      ; 保存 STATUS 寄存器内容到 STATUS_TEMP
;
;
;(ISR) ; 中断服务程序
;
SWAPR  STATUS_TEMP, W   ; 交换 STATUS_TEMP 高低四个字节
STWR   STATUS            ; 恢复标志状态寄存器
SWAPR  W_TEMP, R        ; 恢复 W 寄存器值（用交换指令的目的：不改变状态寄存器值）
SWAPR  W_TEMP, W

```

## 6.0 I/O 端口

此系列器件共有 22 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ADINS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	;清除端口 A 输出缓冲器
LDWI	07H	;
STWR	CMSTA	;关闭比较器
BSR	STATUS,PAGE	;Bank1
CLRR	ANSEL	;所有模拟/数字复用端口设置为数字端口
LDWI	0CH	;设置端口 A 的<3,2>位为输入，其他输出
STWR	CPIOA	;
BCR	STATUS,PAGE	;Bank0

**寄存器 05H: PORTA寄存器 (PORTA)**

U-0	U-0	R/W-x	R/W-x	R-x	R/W-x	R/W-x	R/W-x
—	—	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>**: PORTA I/O引脚位1 = PORTA引脚电平>V<sub>IH</sub>0 = PORTA引脚电平<V<sub>IL</sub>**寄存器 85H: PORTA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
—	—	CPIOA5	CPIOA4	—	CPIOA2	CPIOA1	CPIOA0
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>**: PORTA三态控制位

1 = PORTA引脚配置为输入 (三态)

0 = PORTA引脚配置为输出

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

**6.2 PORTB 和 CPIOB 寄存器****寄存器 06H: PORTB寄存器 (PORTB)**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
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

**PB<7:0>**: PORTB I/O引脚位1 = PORTB引脚电平 $>V_{IH}$ 0 = PORTB引脚电平 $<V_{IL}$ 

寄存器 86H: PORTB三态寄存器 (CPIOB)

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>CPIOB7</b>	<b>CPIOB6</b>	<b>CPIOB5</b>	<b>CPIOB4</b>	<b>CPIOB3</b>	<b>CPIOB2</b>	<b>CPIOB1</b>	<b>CPIOB0</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

**PB<7:0>**: PORTB 三态控制位

1 = PORTB引脚配置为输入 (三态)

0 = PORTB引脚配置为输出

## 6.3 PORTC 和 CPIOC 寄存器

PORTC 是由 8 个双向引脚组成的通用 I/O。引脚可以配置为 I/O 或作为 A/D 转换器 (ADC) 或比较器的模拟输入。要了解各个功能的特定信息 (如 CCP 或 ADC), 请参见本数据手册的相应章节。

注: 必须对ADINSL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 6-4: 初始化 PORTC

BCR	STATUS,PAGE	;Bank 0
CLRR	PORTC	;清输出缓冲区
LDWI	07h	
STWR	CMSTA	;关比较器
BSR	STATUS,PAGE	;Bank 1
CLRR	ANSEL	;设置端口为数字端口
LDWI	0Ch	;设置 PC1, PC2 为输入, 其他 PC 口输出
STWR	CPIOC	
BCR	STATUS.PAGE	;Bank 0

寄存器 07H: PORTC寄存器 (PORTC)

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>PC7</b>	<b>PC6</b>	<b>PC5</b>	<b>PC4</b>	<b>PC3</b>	<b>PC2</b>	<b>PC1</b>	<b>PC0</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

1 = PORTC引脚电平&gt;VIH

0 = PORTC引脚电平&lt;VIL

寄存器 87H: PORTC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

## 6.4 其它引脚功能

MDT10F685 的 PORTA、PORTB 端口上的每一个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。

### 6.4.1 ADINS1 寄存器

ADINS 寄存器用于将 I/O 引脚的输入模式配置为模拟。将相应的 ADINS 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。

ADINS 位的状态对数字输出功能没有影响。CPIO 清零且 ADINS 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。

寄存器 91H: 模拟选择寄存器 (ADINS1)

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>AN7</b>	<b>AN6</b>	<b>AN5</b>	<b>AN3</b>	<b>AN4</b>	<b>AN2</b>	<b>AN1</b>	<b>AN0</b>
<b>PC3</b>	<b>PC2</b>	<b>PC1</b>	<b>PA4</b>	<b>PC0</b>	<b>PA2</b>	<b>PA1</b>	<b>PA0</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 **AN<7:0>**: 模拟选择位

在AN&lt;7:0&gt;引脚上分别进行模拟或数字功能的模拟选择。

1 = 模拟输入。引脚被分配为模拟输入<sup>(1)</sup>。

0 = 数字I/O。引脚被分配给端口或特殊功能。

注 1: PA0、PA1若想启用数字I/O，则需变更比较器模式。将某引脚设置为模拟输入将自动禁止数字输入电路、弱上拉以及电平变化中断（如果有的话）。相应CPIO位必须设置为输入模式以允许对该引脚的电压进行外部控制。

#### 寄存器 11H: 模拟选择寄存器 (ADINS0)

R/W-1	R-0	R/W-1	R-0	R/W-1	R-0	R-0	R/W-1
<b>AN11</b>		<b>AN10</b>		<b>AN9</b>			<b>AN8</b>
<b>PB7</b>		<b>PB5</b>		<b>PB3</b>			<b>PB0</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 **AN11**: PORTB端口bit7模拟选择位  
1 = 模拟输入。引脚被分配为模拟输入  
0 = 数字I/O

bit6 保留位，读为0

bit5 **AN10**: PORTB端口bit5模拟选择位  
1 = 模拟输入。引脚被分配为模拟输入  
0 = 数字I/O

bit4 保留位，读为0

bit3 **AN9**: PORTB端口bit3模拟选择位  
1 = 模拟输入。引脚被分配为模拟输入  
0 = 数字I/O

bit2 保留位，读为0

bit1 保留位，读为0

bit0 **AN8**: PORTB端口bit0模拟选择位  
1 = 模拟输入。引脚被分配为模拟输入  
0 = 数字I/O

## 6.4.2 弱上拉

每一个 PORTA、PORTB 引脚（除 PA3）具有各自的可配置内部弱上拉。控制位 PAPU<sub>x</sub>、PBPU<sub>x</sub> 使能或禁止每一个弱上拉。请参见 [寄存器 95H](#)、[寄存器 106H](#) 和 [寄存器 107H](#)。PORTC 端口上只有上拉总使能控制位，每个端口引脚再没有独立的上拉控制使能位。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  $\overline{\text{MCLR}}$  并禁止时，自动启动 PA3 的弱上拉功能。 $\overline{\text{MCLR}}$  上拉不受软件控制。



## 寄存器 95H: 弱上拉PORTA寄存器 (PAPHR)

R-0	R-0	R/W-1	R/W-1	R-0	R/W-1	R/W-1	R/W-1
—	—	PAPU5	PAPU4	—	PAPU2	PAPU1	PAPU0
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 **PAPU<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 **PAPU<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 ( $\text{CPIO} = 0$ ), 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 PAPU&lt;5:4&gt; 始终读为 1。

## 寄存器 106H: 弱上拉PORTB寄存器 (PBPUR)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
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 **PBPU<7:0>**: 弱上拉控制位

1 = 使能上拉

0 = 禁止上拉

## 寄存器 107H: 弱上拉PORTB、PORTC总控制寄存器 (PBCPUEN)

R/W-1	R/W-1	R-0	R-0	R-0	R-0	R-0	R-0
PBPUEN	PCPUEN						
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 **PBPUEN**: PORTB端口上拉总使能控制位

0 = 使能上拉

1 = 禁止上拉

Bit6 PCPUEN: PORTC端口上拉总使能控制位  
0 = 使能上拉  
1 = 禁止上拉

### 6.4.3 电平变化中断

每一个 PORTA、PORTB 引脚均可分别配置为电平变化中断引脚。控制位 PAINTR\_x、PBINTR\_x 使能或禁止每个引脚的中断功能，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、PORTB 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、PORTB 电平变化中断标志位（PAIF、PBIF）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：

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

电平不匹配条件会继续将 PAIF、PBIF 标志位置 1。而读 PORTA、PORTB 将结束不匹配条件并允许将 PAIF、PBIF 标志位清零。锁存器将保持最后一次读取的值不受 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、PBIF 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	PAINTRO
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<5:4> 始终读为 1。

#### 寄存器 186H: 电平变化中断 PORTB 寄存器 (PBINTR)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PBINTR7	PBINTR6	PBINTR5	PBINTR4	PBINTR3	PBINTR2	PBINTR1	PBINTRO
bit7							bit0

图注：                              W = 可写位                              U = 未实现位，读为0                              x = 未知

R = 可读位	1 = 置1	0 = 清零
-n = POR时的值		

Bit7-0      **PBINTR<7:0>**: 电平变化中断PORTB控制位  
 1 = 允许电平变化中断  
 0 = 禁止电平变化中断

## 6.4.4 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      ;清 0 端口 A 的 PA0
LDWI  07H
STWR  CMSTA        ;关闭比较器
BSF   STATUS,PAGE   ;Bank 1
BCR   ANSEL,0      ;端口 A 的 PA0 设置为数字口
BCR   CPIOA,0      ;端口 A 的 PA0 设置为输出
LCALL CapDelay     ; 延时
BSR   PSTA,ULPWUE  ;使能超低功耗唤醒功能
BSR   PAINTR,0     ;设置 PA0 口电平变化唤醒功能
BSR   CPIOA,0      ;设置 PA0 口输入
LDWI  B'10001000'  ;使能电平变化中断、全局中断
STWR  INTS         ;
SLEEP                                     ;睡眠，等待唤醒
NOP
NOP
  
```

## 6.5 PORTA 引脚说明和引脚图

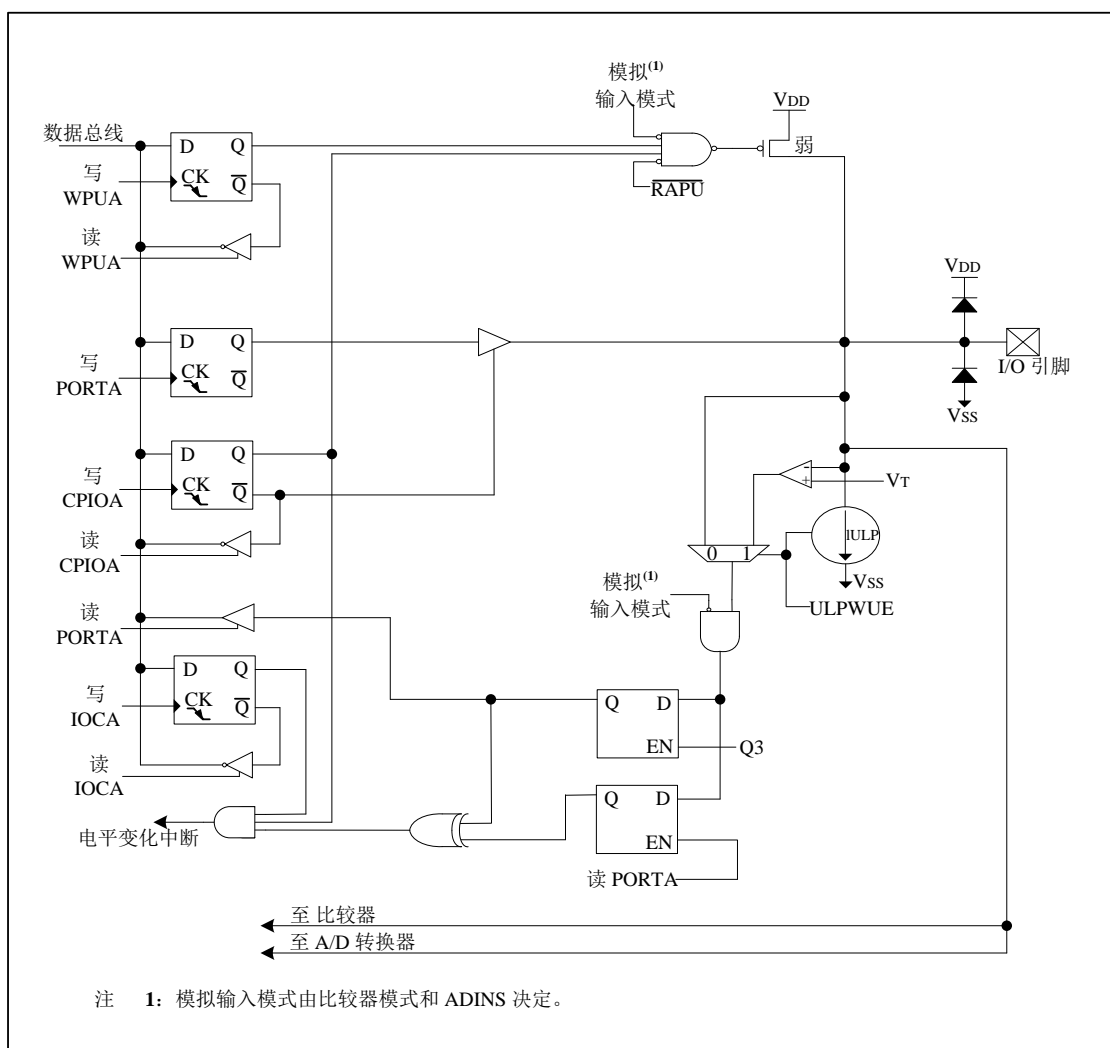
每个 PORTA 引脚都与其他功能复用。这里将简要说明引脚及其复合功能。各功能如比较器或 A/D 转换器 (A/D Converter, ADC) 的具体信息，请参见本数据手册中的相关章节。

## 6.5.1 PA0/AN0/C1IN+/SCK/ULPWU/CCP1

图 6-5-1 给出了此引脚的引脚图。PA0/AN0/C1IN+/SCK/ULPWU/CCP1 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
- 增强型 CCP 数字输入/输出

图 6-5-1: PA0 框图



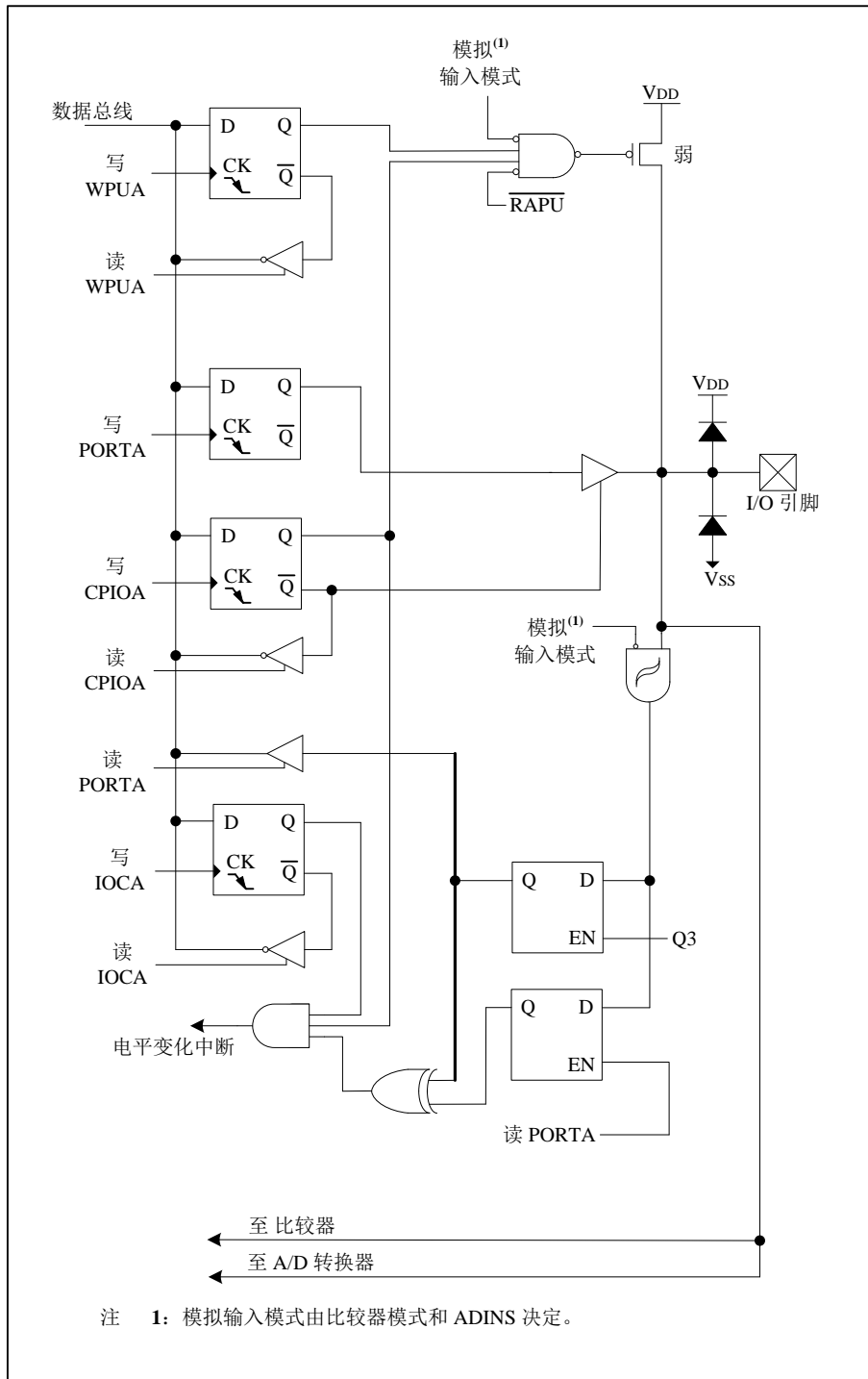
## 6.5.2 PA1/AN1/C1IN-/VREF/SDA/PWM2

图 6-5-2 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
- 连接至 ADC 的模拟输入
- 可软件配置的 PWM2 输出引脚

图 6-5-2: PA1 框图



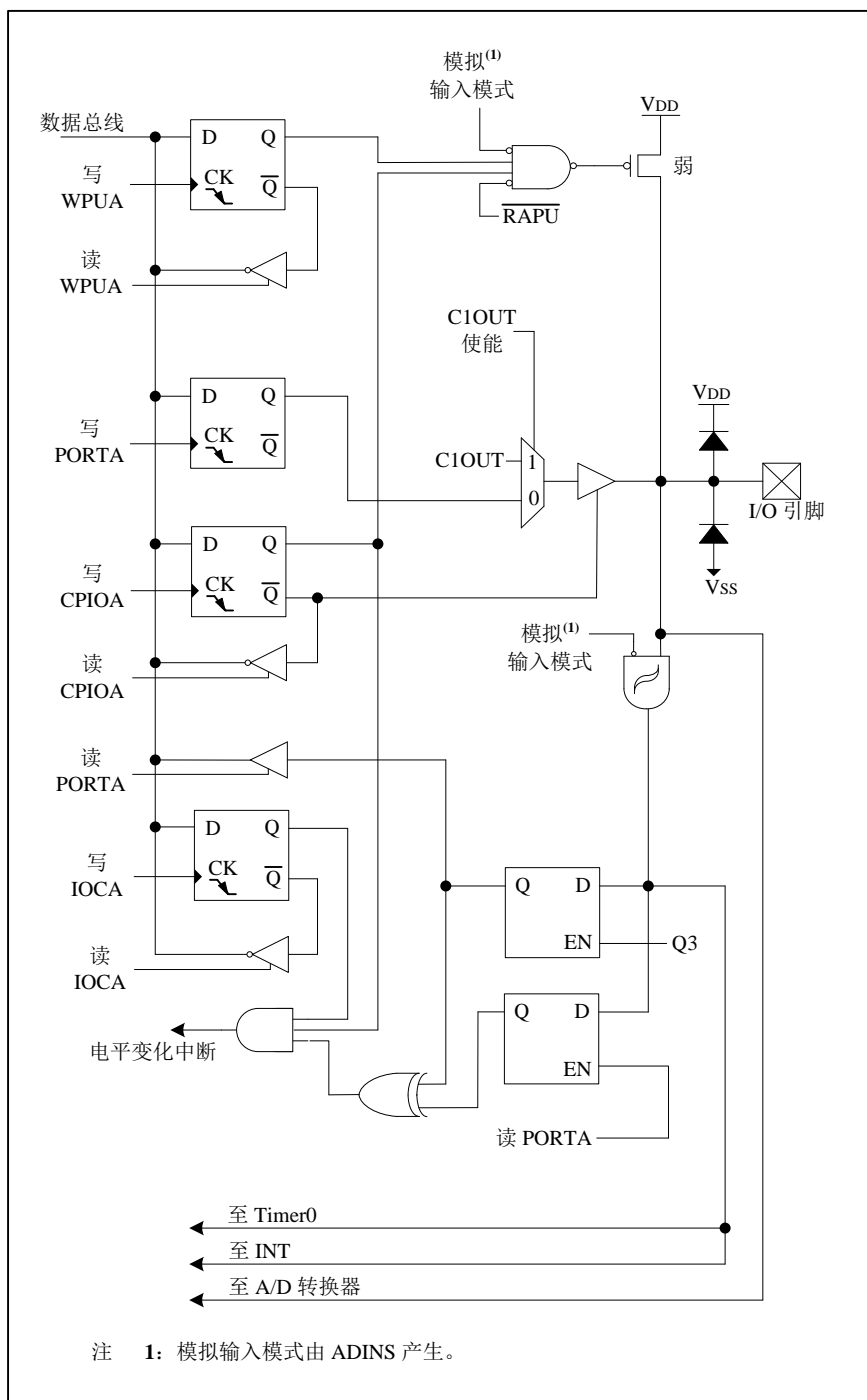
### 6.5.3 PA2/AN2/T0CKI/INT/C1OUT/PWM3

图 6-5-3 给出了此引脚的引脚图。PA2/AN2/T0CKI/INT/C1OUT/PWM3 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
- 可软件配置的 PWM3 输出引脚

图 6-5-3: PA2 框图

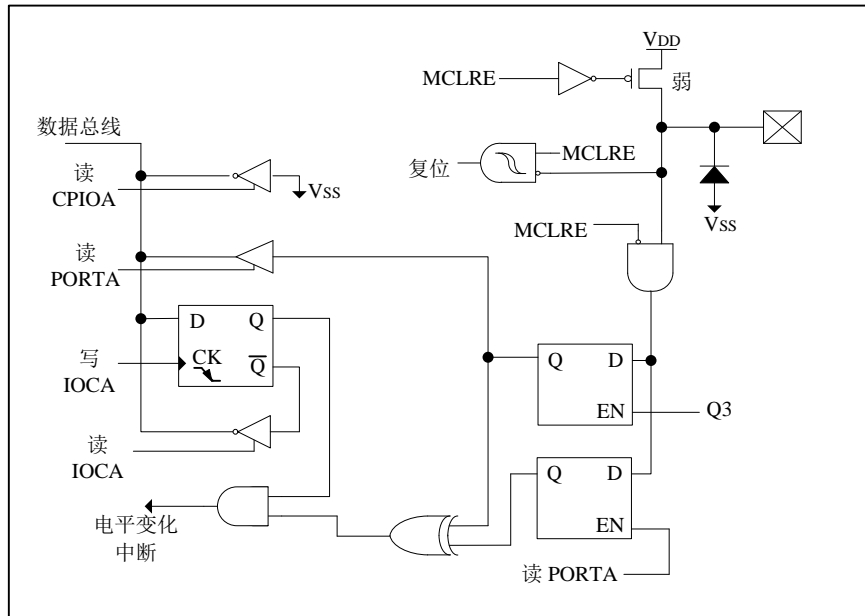


## 6.5.4 PA3/MCLR/VPP

图 6-5-4 给出了此引脚的引脚图。PA3/MCLR/VPP 引脚可配置为下列功能之一：

- 通用输入
- 带弱上拉的主清零复位

图 6-5-4: PA3 框图

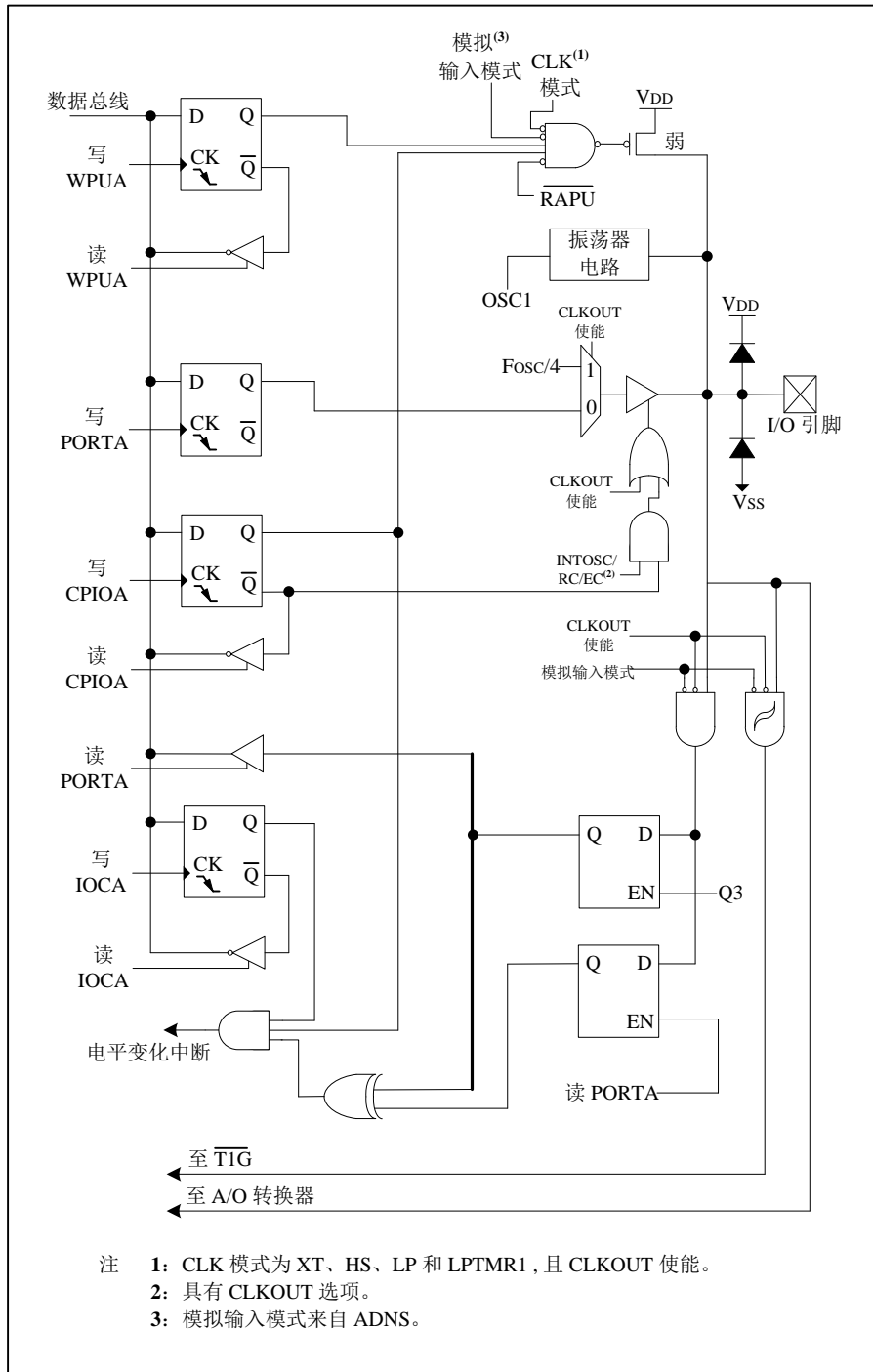


## 6.5.5 PA4/AN3/T1G/OSC2/CLKOUT/PWM4

图 6-5-5 给出了此引脚的引脚图。PA4/AN3/T1G/OSC2/CLKOUT/PWM4 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
- 可软件配置的 PWM4 输出引脚

图 6-5-5: PA4 框图



### 6.5.6 PA5/T1CKI/OSC1/CLKIN/PWM3

图 6-5-6 给出了此引脚的引脚图。PA5/T1CKI/OSC1/CLKIN/PWM3 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
- 可软件配置的 PWM3 输出引脚

图 6-5-6: PA5 框图



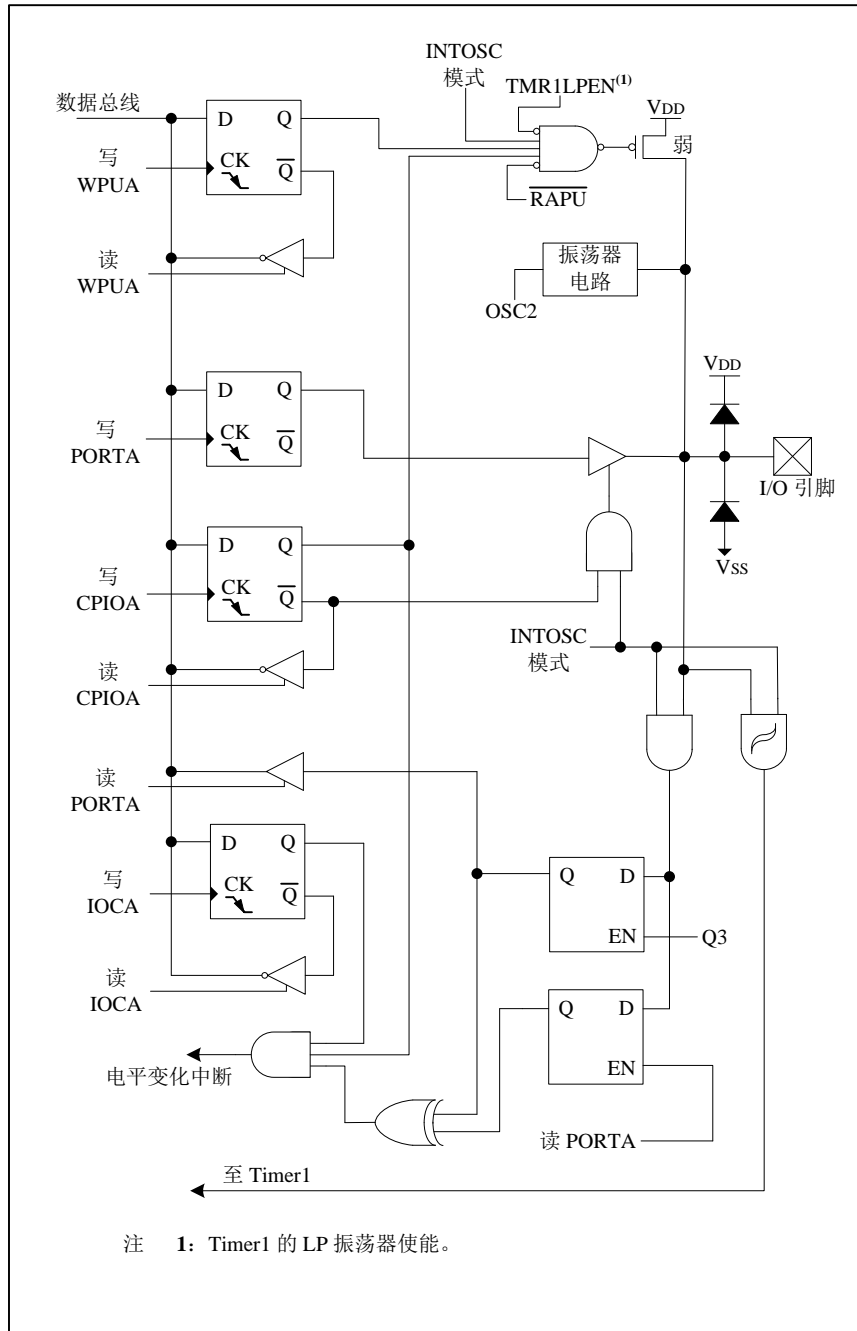


表 6-5: 与 PORT A 相关的寄存器汇总

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

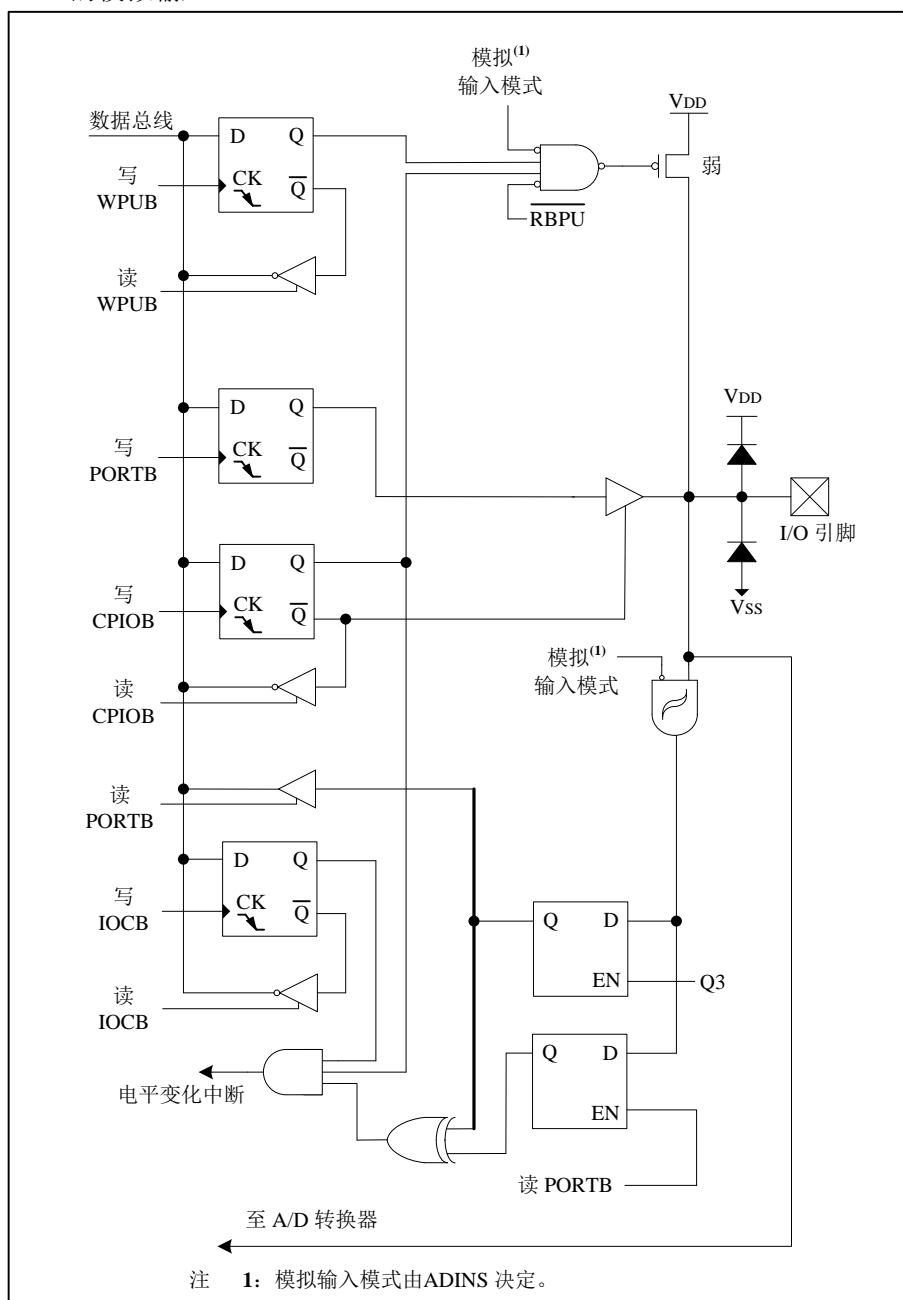
图注: x = 未知, u = 不变, — = 未实现 (读为 0)。PORT A 不使用阴影单元。

## 6.6 PORTB 引脚说明和引脚图

### 6.6.1 PB0/AN8

PB0 引脚可以配置为下列功能之一：

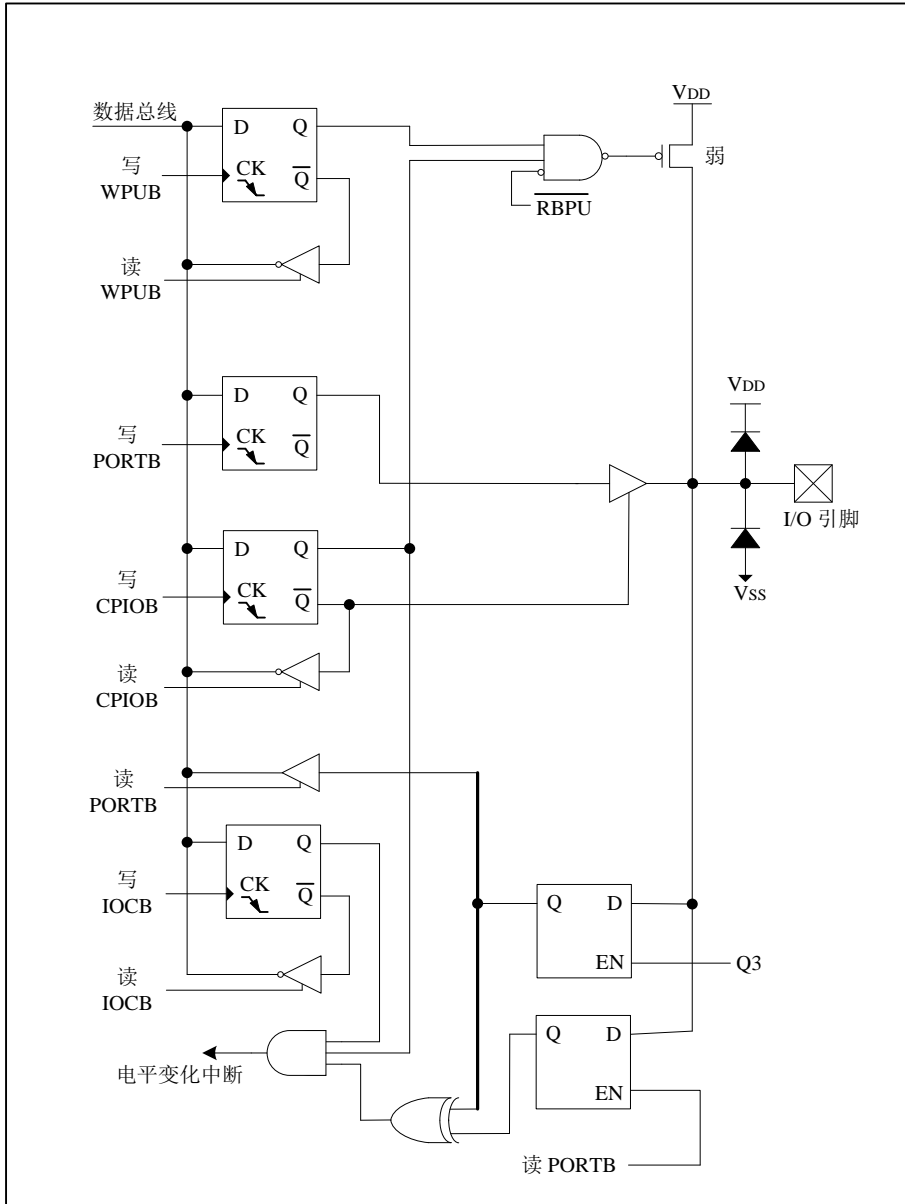
- 通用 I/O
- ADC 的模拟输入



## 6.6.2 PB1/PWM3

PB0 引脚可以配置为下列功能之一：

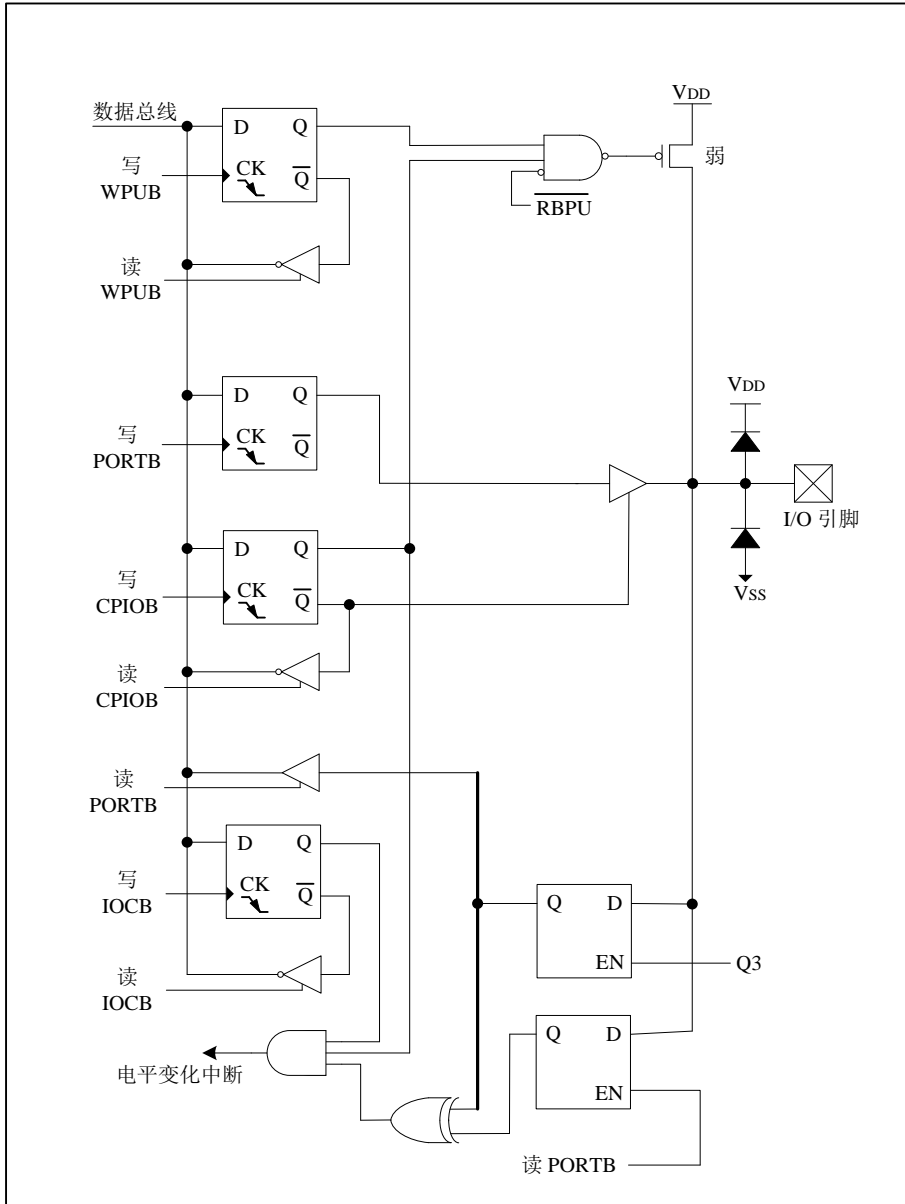
- 通用 I/O
- 可软件配置的 PWM3 输出引脚



### 6.6.3 PB2/PWM4

PB0 引脚可以配置为下列功能之一：

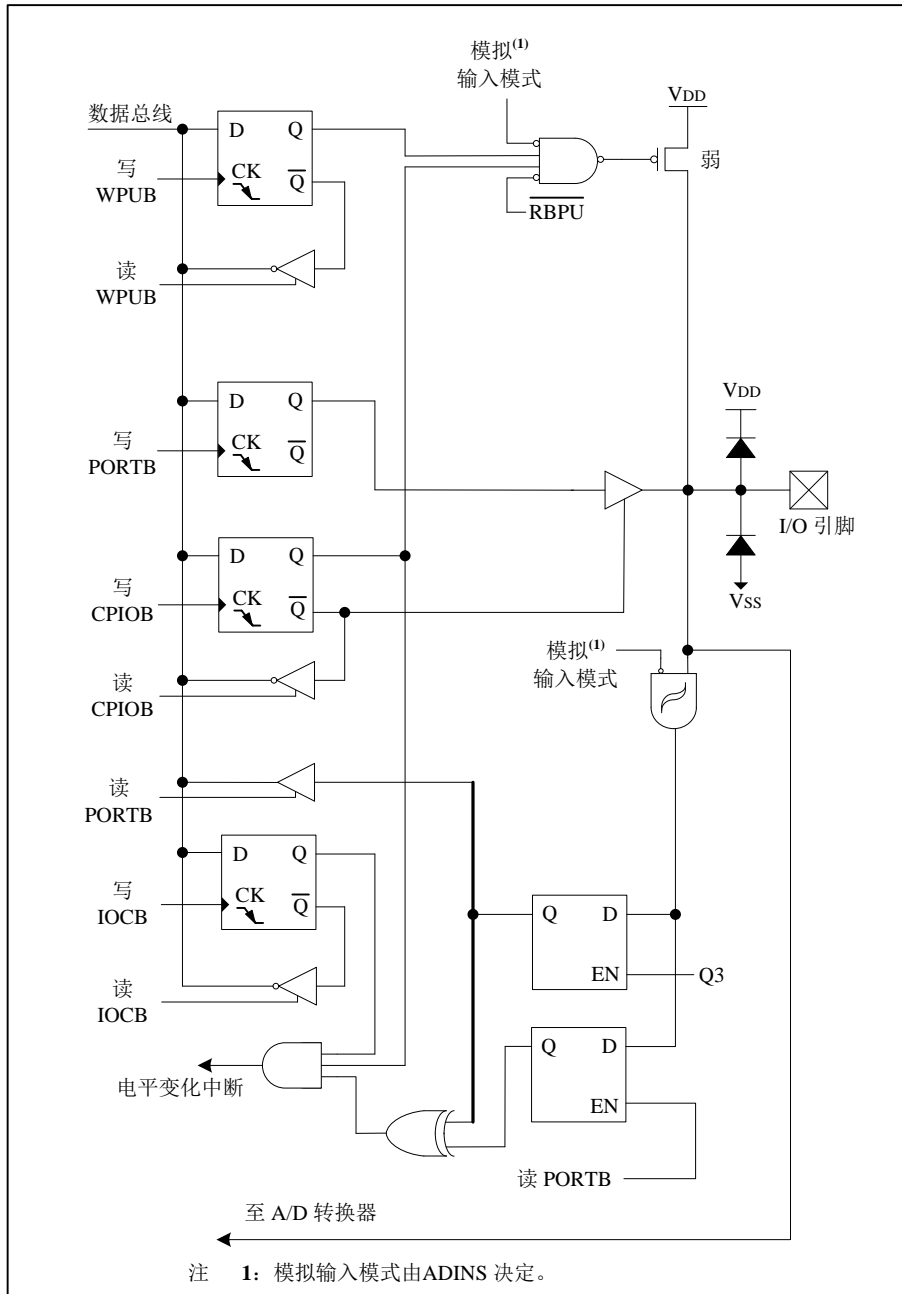
- 通用 I/O
- 可软件配置的 PWM4 输出引脚



### 6.6.4 PB3/AN9

PB0 引脚可以配置为下列功能之一：

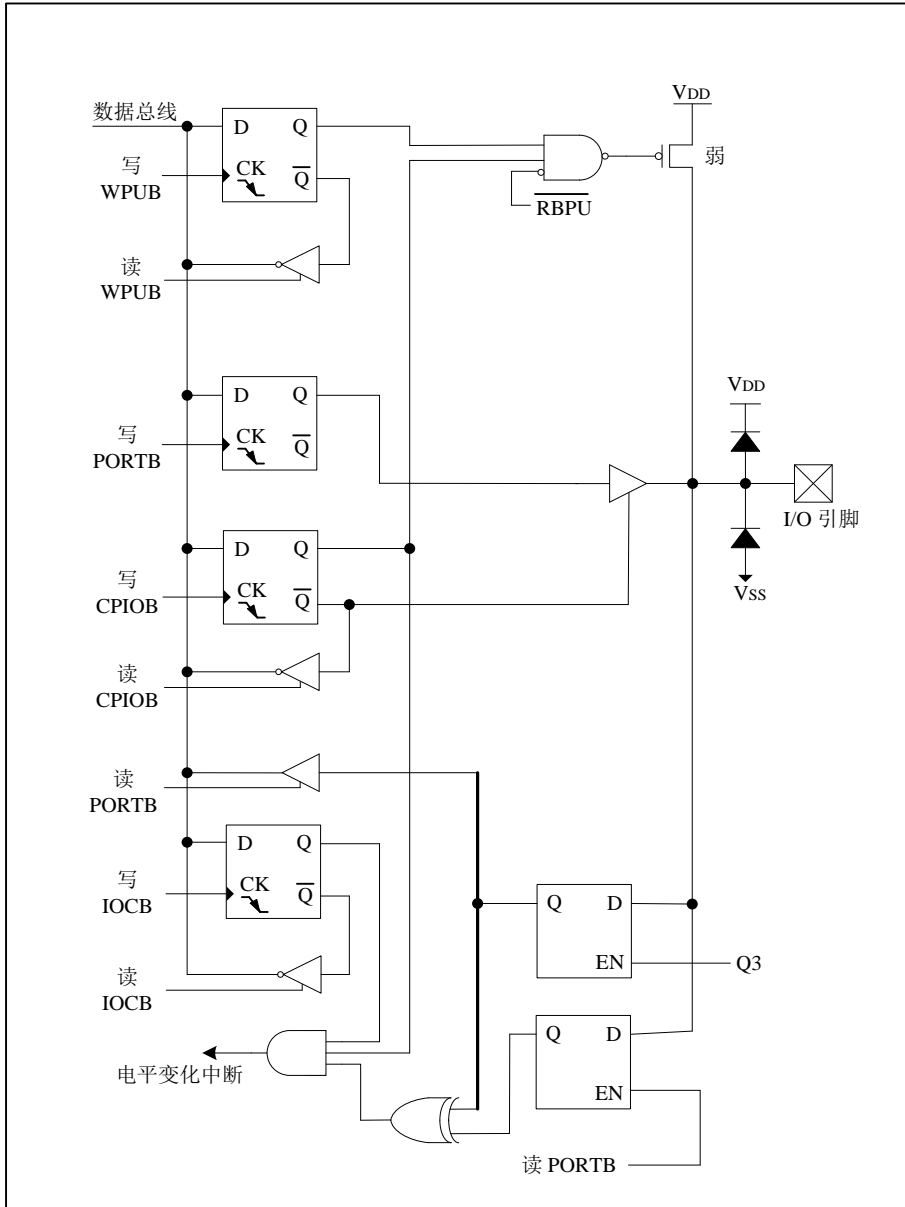
- 通用 I/O
- ADC 的模拟输入



## 6.6.5 PB4/PWM2

PB0 引脚可以配置为下列功能之一：

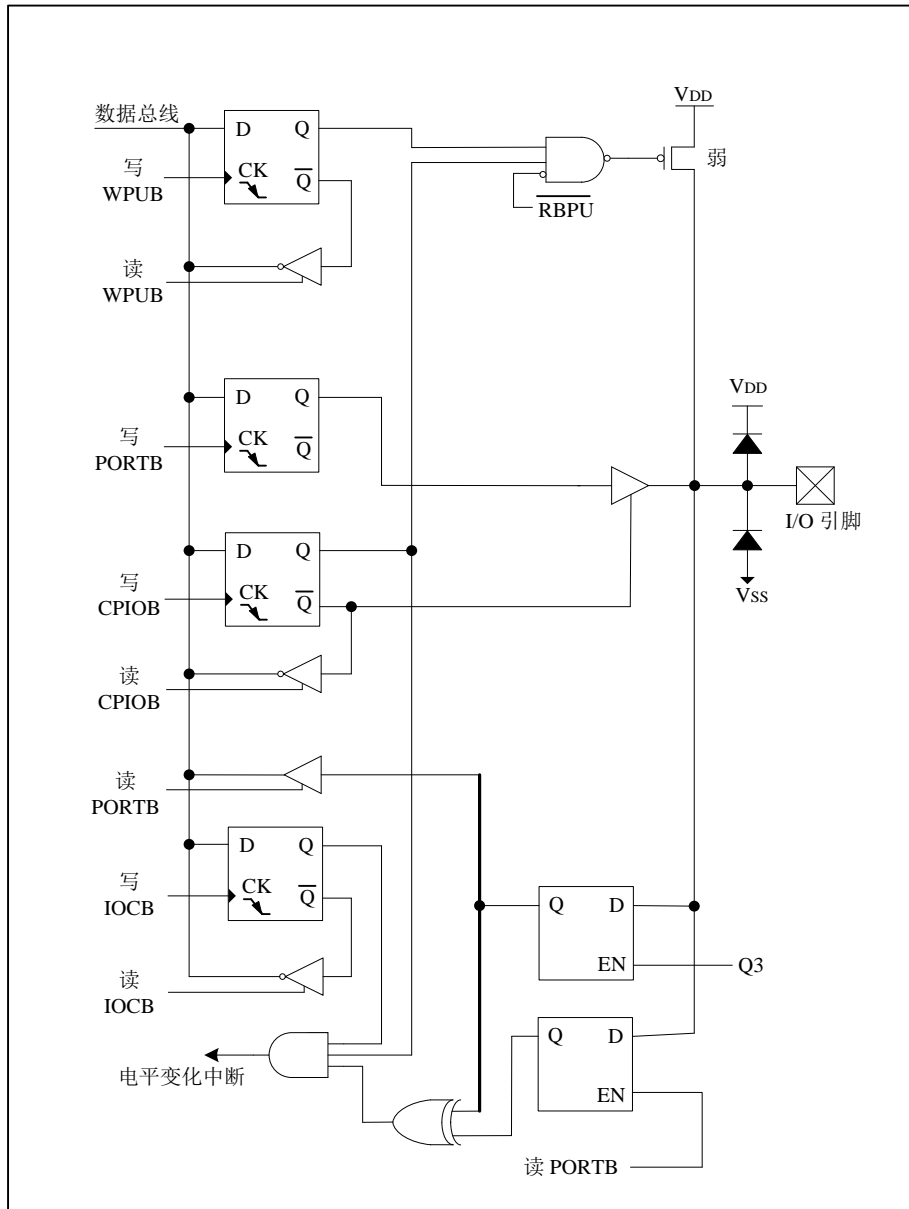
- 通用 I/O
- 可软件配置的 PWM2 输出引脚



## 6.6.4 PB5/AN10

PB0 引脚可以配置为下列功能之一：

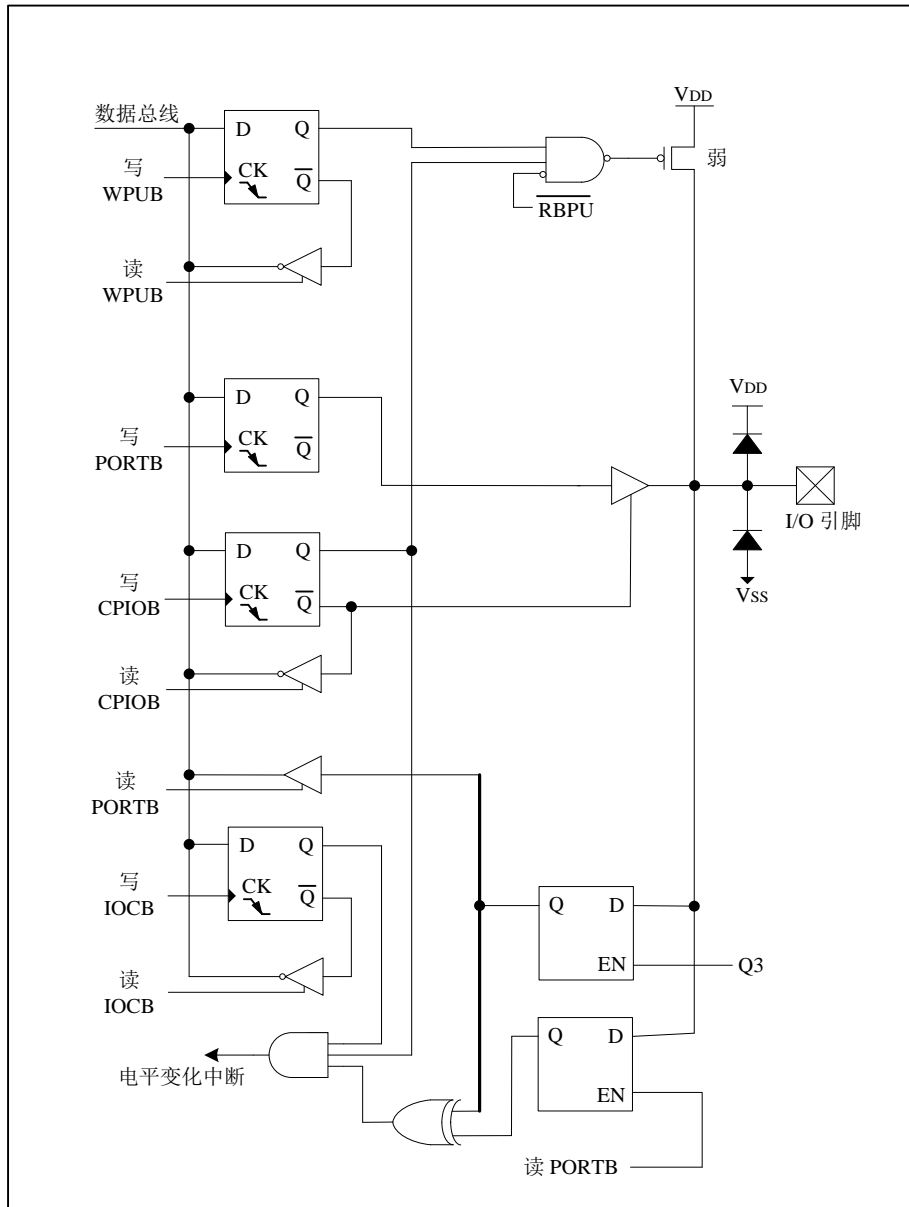
- 通用 I/O
- ADC 的模拟输入



### 6.6.5 PB6/PWM3

PB0 引脚可以配置为下列功能之一：

- 通用 I/O
- 可软件配置的 PWM3 输出引脚



## 6.6.5 PB7/AN11

PB0 引脚可以配置为下列功能之一：

- 通用 I/O
- ADC 的模拟输入



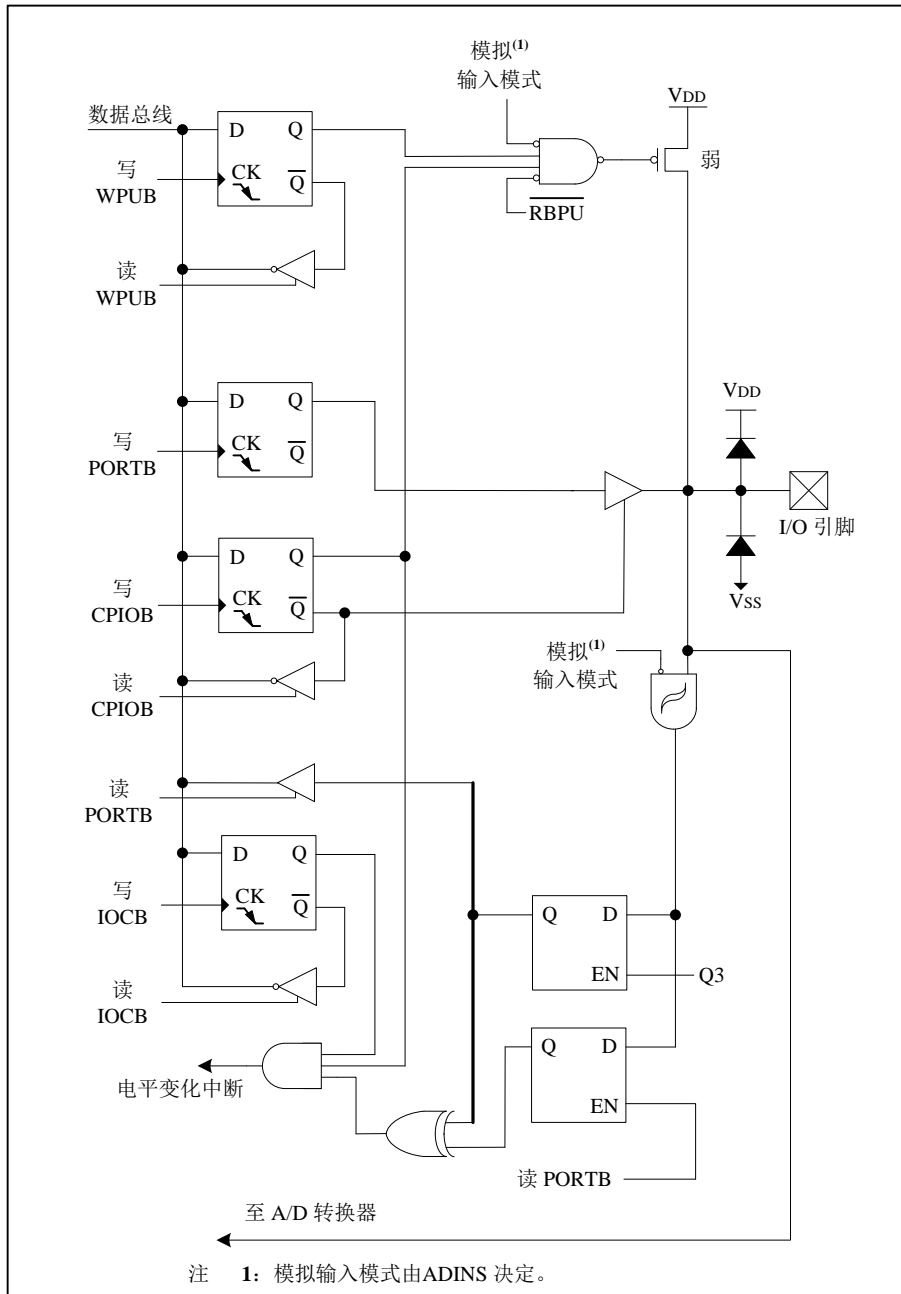


表 6-6: 与 PORTB 相关的寄存器汇总

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
CMSTA	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	xxxx 0000	uuuu uu00
CPIOB	CPIOB7	CPIOB6	CPIOB5	CPIOB4	CPIOB3	CPIOB2	CPIOB1	CPIOB0	1111 1111	1111 1111
ADINS1	AN7	AN6	AN5	AN3	AN4	AN2	AN1	AN0	1111 1111	1111 1111
ADINS0	AN11		AN10		AN9			AN8	1010 1001	1010 1001
PBPUR	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0	1111 1111	1111 1111
PBINTR	PNINTR7	PBINTR6	PBINTR5	PBINTR4	PBINTR3	PBINTR2	PBINTR1	PBINTR0	1111 1111	1111 1111
PBCPUEN	PBPUEN	PCPUEN							1100 0000	1100 0000

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

## 6.7 PORTC 引脚说明和引脚图

### 6.7.1 PC0/AN4/C2IN+

PC0 引脚可以配置为下列功能之一：

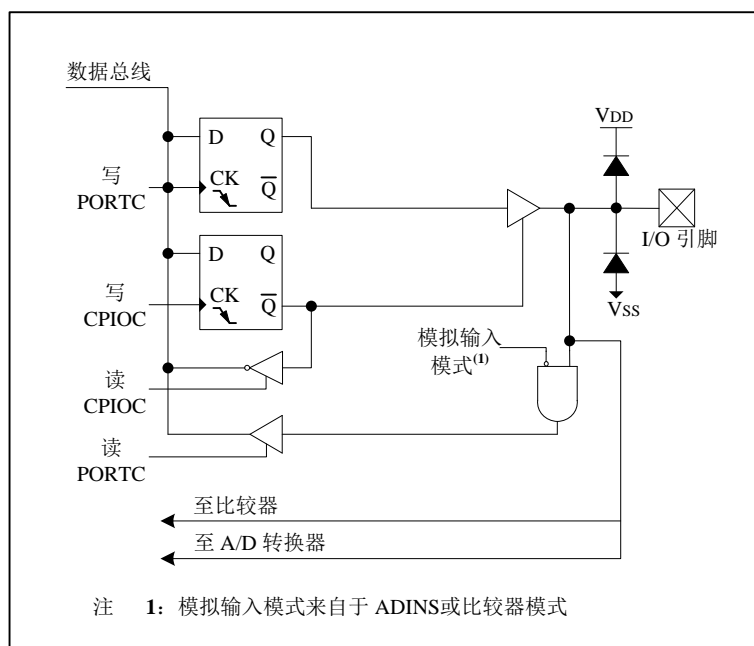
- 通用 I/O
- ADC 的模拟输入
- 比较器的模拟输入

### 6.7.2 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 6-7-2: PC0 和 PC1 框图



### 6.7.3 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

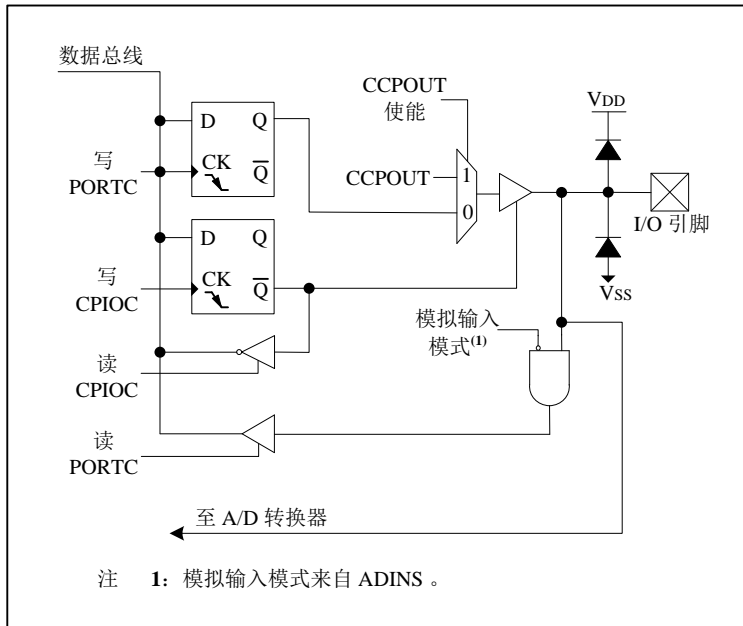
### 6.7.4 PC3/AN7/P1C

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 6-7-4: PC2 和 PC3 框图



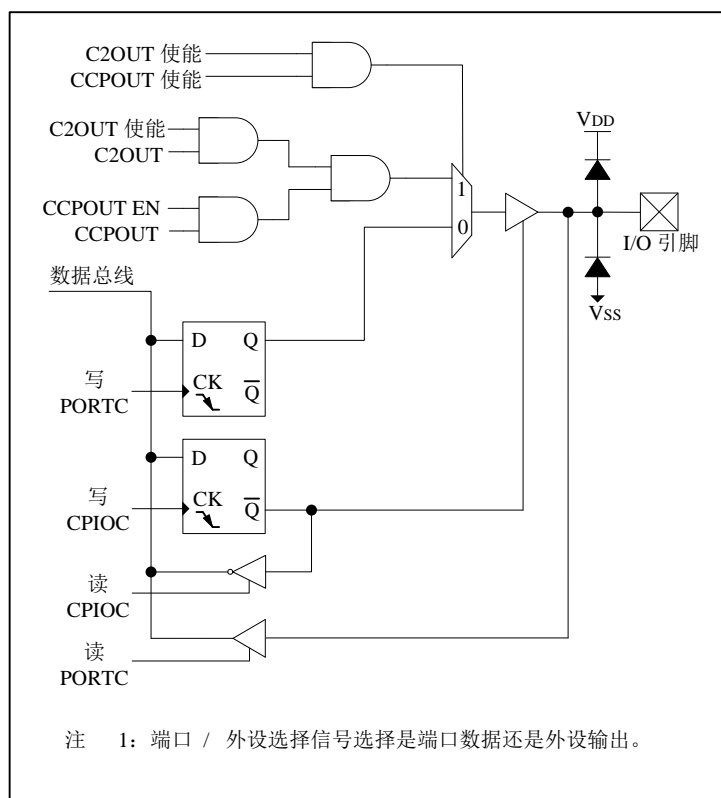
### 6.7.5 PC4/C2OUT/P1B

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 6-7-5: PC4 框图

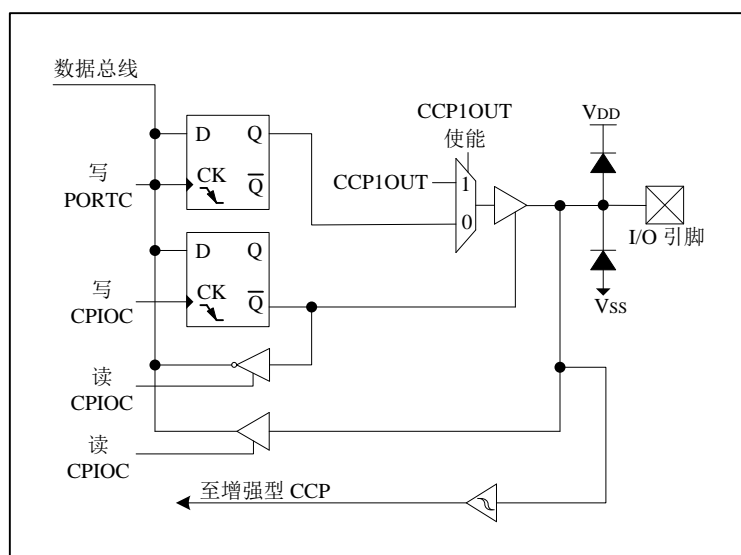


### 6.7.6 PC5/CCP1/P1A

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 6-7-6: PC5 框图



## 6.7.7 PC6/PWM2

PC6 引脚可以配置为下列功能之一：

- 通用 I/O
- 可软件配置的 PWM2 输出引脚

## 6.7.8 PC7/PWM4

PC7 引脚可以配置为下列功能之一：

- 通用 I/O
- 可软件配置的 PWM4 输出引脚

表 6-7: 与 PORTC 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
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
ADINS1	AN7	AN6	AN5	AN3	AN4	AN2	AN1	AN0	1111 1111	1111 1111
PBCPUEN	PBPUEN	PCPUEN							1100 0000	1100 0000

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

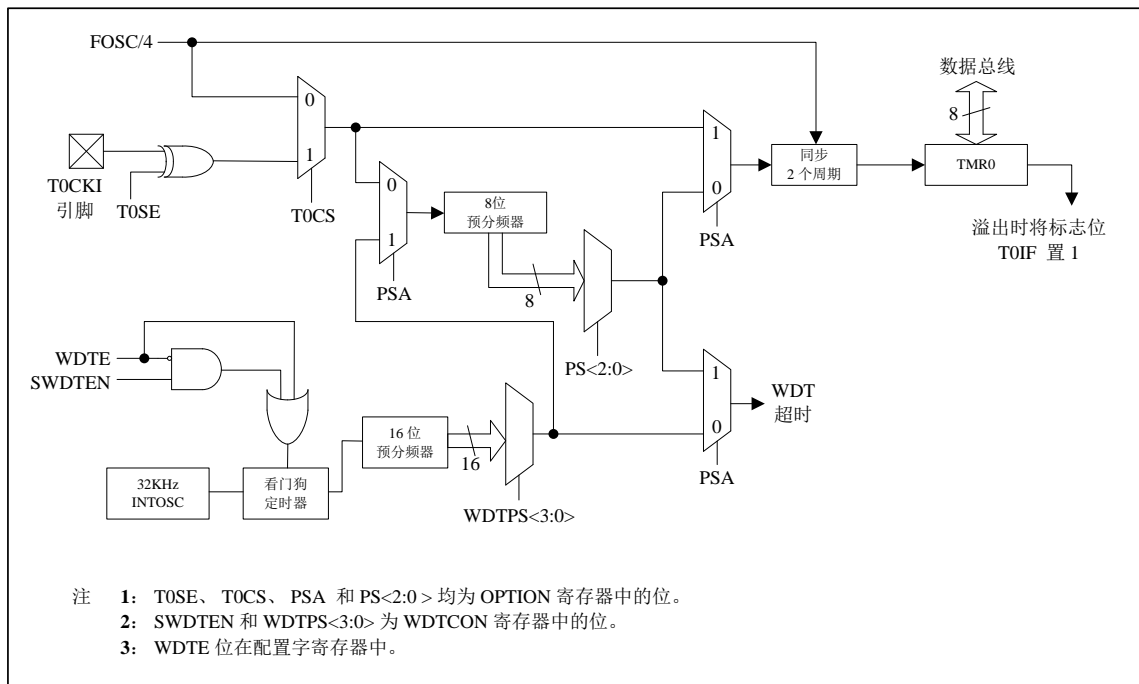
## 7.0 定时器

### 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
- 定时器/计数器重载功能

图 7-1: Timer0 模块的框图



#### 7.1.1 Timer0 的工作原理

Timer0 是一个 8 位并具有重载功能的定时器/计数器，读 RTCC，读的是计数器的值，写 RTCC，写的是重载寄存器里的值。

##### 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 &lt;7:0&gt;: 8位定时器/计数器寄存器

## 7.1.2 8 位定时器模式

Timer0 的开关使能位 `TIMER0_ON` 见寄存器 `CMCON` (寄存器 `1AH`)，只有打开使能位，Timer0 才能工作。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                      ;清看门狗定时器
CLRR    TMR0          ;清 TMR0
BSR    STATUS,PAGE    ;Bank 1
LDWI    b'10001111'   ;禁止上拉，RTCC 分频给看门狗，分频比 1:128
TMODE                      ;
CLRWT                      ;清看门狗
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                      ;清看门狗
BSR    STATUS,PAGE    ;Bank 1
LDWI    b'xxxx0xxx'   ;将分频器分配给 TMR0
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
CMCON						TIMER0_ON	T1GSS	C2SYNC	0000 0010	0000 0010
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。

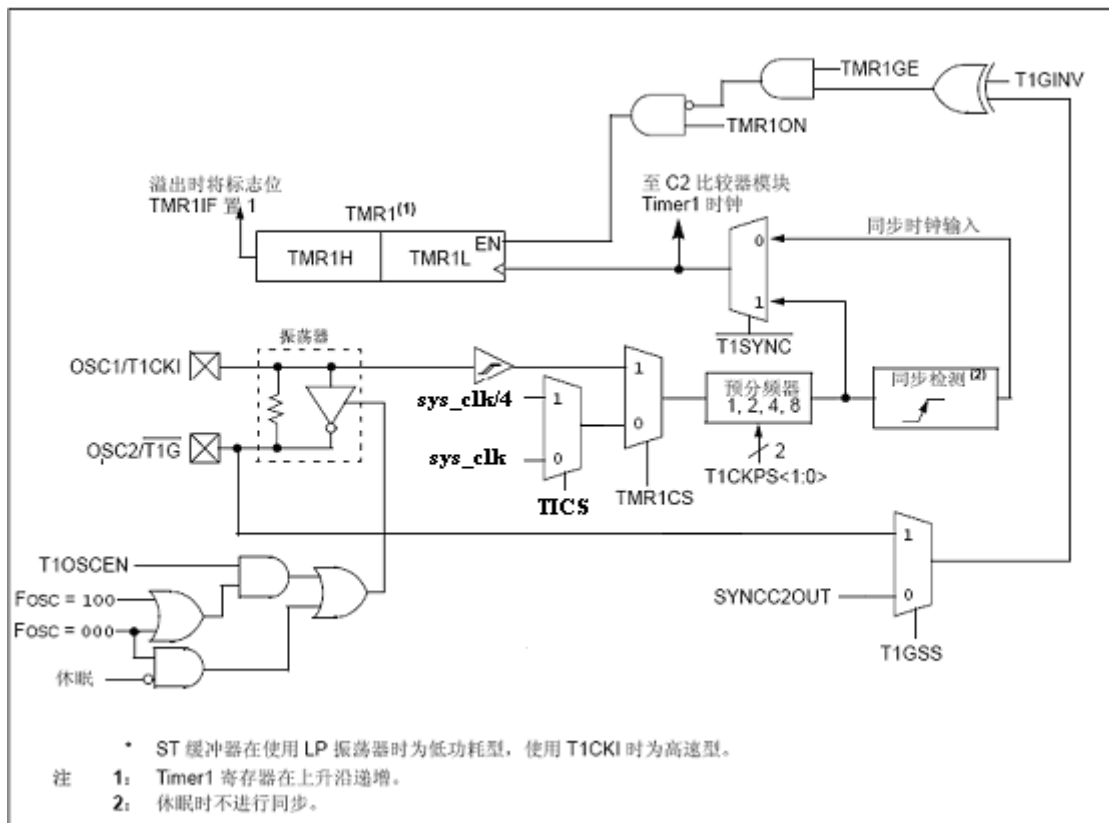


## 7.2 带门控的 Timer1 模块

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）
- 溢出中断
- 溢出时唤醒（仅限外部时钟且异步模式时）
- 捕捉/比较功能的时基
- 特殊事件触发（带 ECCP）
- 比较器输出与 Timer1 时钟同步

图 7-2: Timer1 模块的框图



### 7.2.1 Timer1 的工作原理

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

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

如寄存器 10H 所示，Timer1 控制寄存器 T1STA 用于控制 Timer1 并选择 Timer1 模块的各种功能。

#### 寄存器 10H: TIMER1 控制寄存器 (T1STA)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

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      **T1GINV**: Timer1门控翻转位<sup>(1)</sup>  
 1 = Timer1门控为高电平有效 (门控为高电平时Timer1计数)  
 0 = Timer1门控为低电平有效 (门控为低电平时Timer1计数)

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  
 10 = 1:4  
 01 = 1:2  
 00 = 1:1

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

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 <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。

当 TMR1CS = 0 时, 由寄存器 CCP1CON ([寄存器 15H](#)) Bit[5] Timer1 内部时钟选择位选择时钟源

TICS = 0 时内部时钟使用内部时钟 sys\_clk/4;

TICS = 1 时内部时钟使用系统时钟 sys\_clk。

### 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。

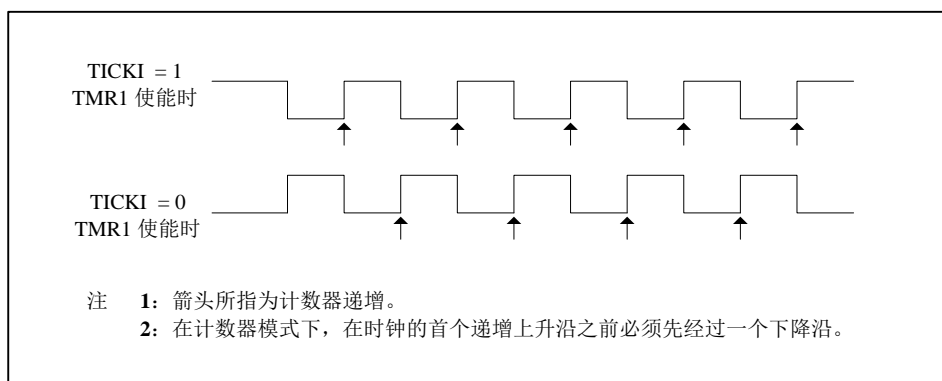
若需外部时钟振荡器(且单片机使用不带 CLKOUT 的 INTOSC), 则 Timer1 可使用 LP 振荡器作为时钟源。

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 1AH](#)）。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 1AH](#)）。

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:TTMR1L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	—	—	—	—	—	TIMER0_ON	T1GSS	C2SYNC	0000 0010	0000 0010
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
CCP1CON	PIM1	PIM0	TICS	ECCP_CS	CCP1M3	CCP1M2	CCP1M1	CCP1M0	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。

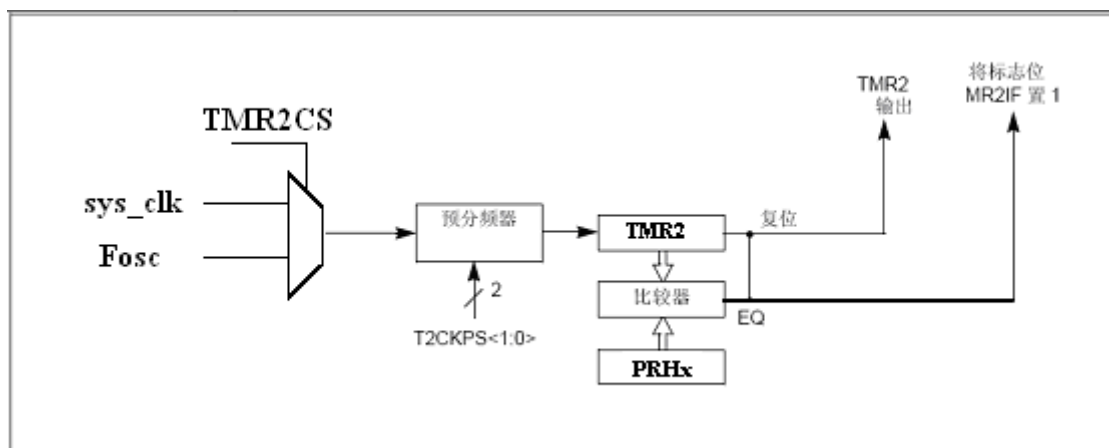
## 7.3 Timer2 模块

Timer2 模块是一个 16 位定时器，具备以下功能：

- 16 位定时器寄存器（TMR2H:TMR2L）
- 16位周期寄存器
- TMR2 计数溢出中断
- 软件可编程预分频比
- 三路同周期可独立配置占空比的PWM

Timer2 的框图请参见图 7-3。

图 7-3: TIMER2 框图

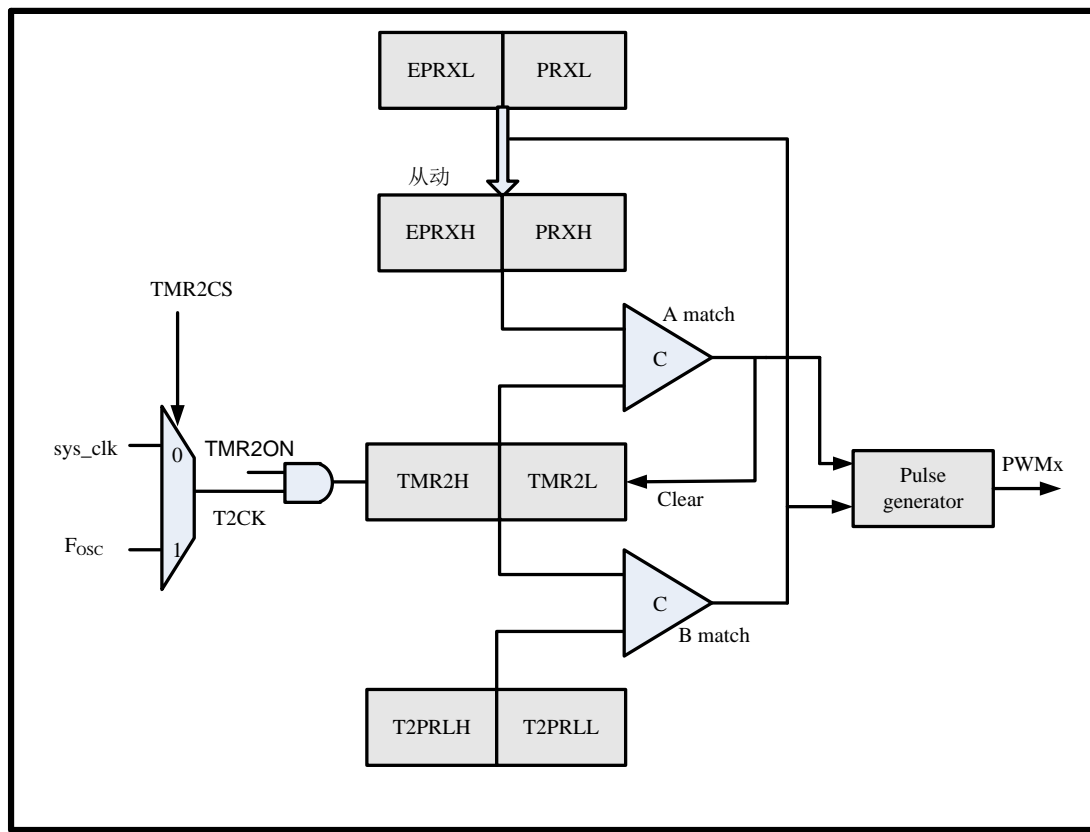


### 7.3.1 Timer2 时钟源

Timer2模块是16位递增计数器，通过一对寄存器TMR2H:TMR2L访问。对TMR2H或TMR2L的写操作将直接更新计数器。 Timer2模块的时钟输入是系统指令时钟（ $F_{Osc}/4$ ）。该时钟送入Timer2预分频器，其预分频比有1:1、1:2、1:4或1:8 四种选择。随后预分频器的输出被用于递增TMR2寄存器。

### 7.3.2 Timer2 的 PWM

简化的PWM功能见下图



Timer2的PWM框图

### 7.3.3 PWM 的输出

PWM输出见图7-3-3，由时基（周期）和一段输出保持为高电平（占空比）的时间组成。

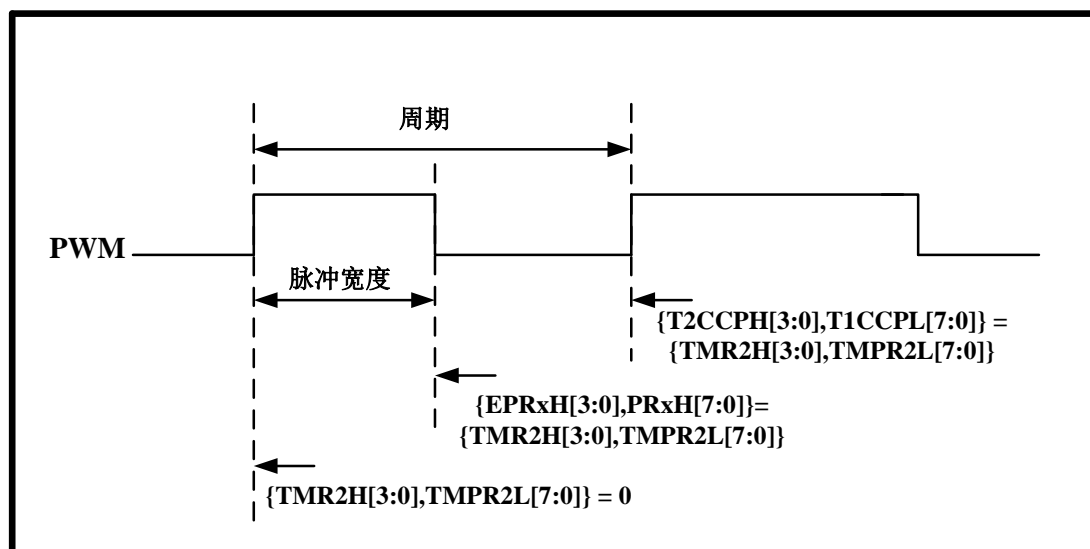
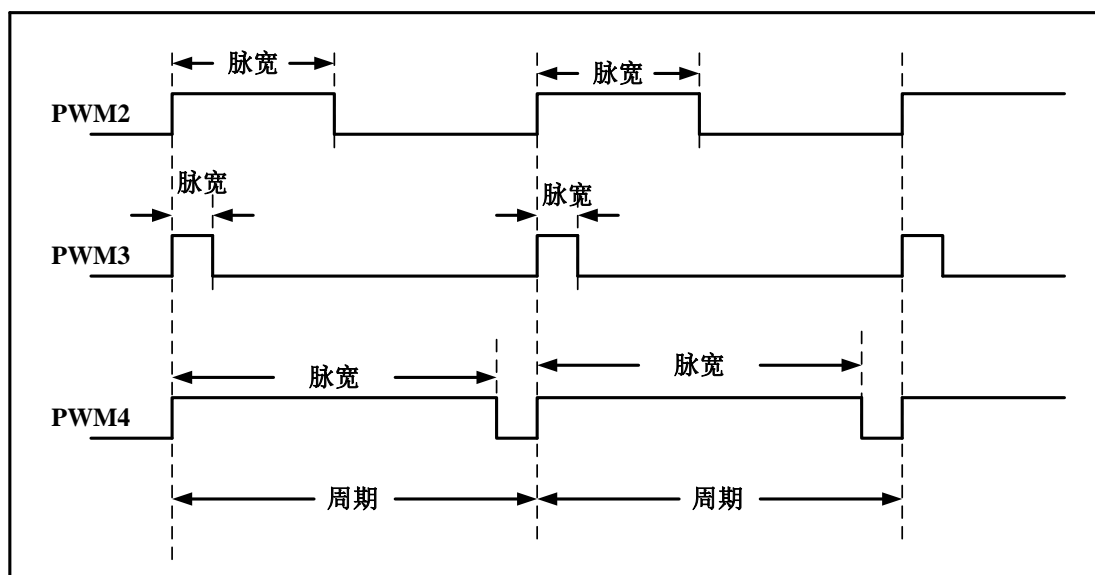


图7-3-3





三路PWM输出

### 7.3.4 PWM 的周期

PWM 周期由 Timer2 的  $\{T2PRLH, T2PRLL\}$  寄存器指定。用可计算 PWM 周期。

$$\text{PWM 周期} = (T2PRL + 1) \times T2CK \times (\text{TMR2 预分频值})$$

当 TL2, TH2 等于  $\{T2PRLH, T2PRLL\}$  时，下一次递增周期将发生以下事件：。

1. TMR2L, TMR2H 被清零。
2. PWM 输出引脚被置 1（例外：若 PWM 占空比 = 0%，引脚不被置 1）。
3. PWM 占空比从  $EPR_xL$  锁存到  $EPR_xH$ 。

### 7.3.5 PWM 的占空比

通过对以下几个寄存器写入 12 位值可指定 PWM 占空比：T2PRLH 寄存器和 T2PRLL 寄存器。

$$\text{PWM 脉冲宽度} = (EPR_xL : PR_xL) \times T2CK \times (\text{TMR2 预分频值})$$

$$\text{PWM 占空比} = \frac{(EPR_xL : PR_xL)}{T2PRL + 1}$$

PRH 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。

12 位定时器 TMR2H<3:0>寄存器与 TMR2L<7:0>寄存器，组成 12 位时基。如果 Timer2 预分频寄存器设置为 1:1 则使用系统时钟。

当 12 位时基与 PRH 及 2 位的锁存器匹配时，CCP2 引脚被清零。

### 7.3.6 PWM 的分辨率

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。

PWM分辨率

$$\text{分辨率} = \log_2(T2PRL + 1) \text{ 位}$$

### 7.3.6 PWM 的工作设置

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。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。

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 引脚的输出驱动器。
2. 装载 T2PRLH:T2PRLL 寄存器以设置 PWM 周期。
3. 装载 Timer2 的计数寄存器 TMR2H:TMR2L, 值与 T2PRLH:T2PRLL 相等。
4. 装载 EPRxL: PRxL 寄存器和 T2CON 寄存器的 T2CKPS 设置 PWM 占空比。
5. 用适当的值装载 T2CON 寄存器将 T2MOD 配置为 PWM 模式，并且使能 Timer2。

MCU工作在调试模式下时，PWM会维持输出。

寄存器 12H: TIMER2控制寄存器 (T2CON)

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>T2MOD</b>	<b>TMR2CS</b>	<b>P4EN</b>	<b>P3EN</b>	<b>P2EN</b>	<b>TMR2ON</b>	<b>T2CKPS1</b>	<b>T2CKPS0</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      **T2MOD:** Timer2模式的选择。  
0 = Timer2 工作在定时/计数模式；  
1 = Timer2工作在PWM模式
- bit6      **TMR2CS:** Timer2时钟源选择位  
0 = sys\_clk系统时钟  
1 = F<sub>osc</sub>内部快时钟
- bit5      **P4EN:** PWM4输出使能位  
0 = PWM4 禁止输出  
1 = PWM4 允许输出
- bit4      **P3EN:** PWM3输出使能位  
0 = PWM3 禁止输出  
1 = PWM3 允许输出

- bit3      **P3EN**: PWM2输出使能位  
0 = PWM2 禁止输出。  
1 = PWM2 允许输出。
- 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2  
10 = 预分频比为4  
11 = 预分频比为8

#### 寄存器 116H: PWM控制寄存器 (PWM2CON)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0
<b>P4CS1</b>	<b>P4CS0</b>	<b>P3CS1</b>	<b>P3CS0</b>	<b>P2CS1</b>	<b>P2CS0</b>	<b>DUMMY_2</b>	<b>DUMMY_1</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    **P4CS<1:0>**: PWM4输出I/O选择位  
00 = PWM4 在 PC7 输出  
01 = PWM4 在 PB2 输出  
1x = PWM4 在 PA4 输出
- bit5-4    **P3CS<1:0>**: PWM3输出I/O选择位。  
00 = PWM3 在 PB1 输出。  
01 = PWM3 在 PB6 输出。  
10 = PWM3 在 PA2 输出。  
11 = PWM3 在 PA5 输出。
- bit3-2    **P2CS<1:0>**: PWM2输出I/O选择位  
00 = PWM2 在 PC6 输出。  
01 = PWM2 在 PB4 输出。  
1x = PWM2 在 PA1 输出。
- bit1      DUMMY\_2, Dummy Cell, 复位值为 1
- bit0      DUMMY\_1, Dummy Cell, 复位值为0

#### 寄存器 111H: Timer2寄存器 (TMR2L)

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>TMR2L&lt;7&gt;</b>	<b>TMR2L&lt;6&gt;</b>	<b>TMR2L&lt;5&gt;</b>	<b>TMR2L&lt;4&gt;</b>	<b>TMR2L&lt;3&gt;</b>	<b>TMR2L&lt;2&gt;</b>	<b>TMR2L&lt;1&gt;</b>	<b>TMR2L&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            **TMR2L<7:0>**: Timer2低8位计数寄存器

**寄存器 191H: Timer2寄存器 (TMR2H)**

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>TMR2H&lt;7</b>	<b>TMR2H&lt;6</b>	<b>TMR2H&lt;5</b>	<b>TMR2H&lt;4</b>	<b>TMR2H&lt;3</b>	<b>TMR2H&lt;2</b>	<b>TMR2H&lt;1</b>	<b>TMR2H&lt;0</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            **TMR2H<7:0>**: Timer2高8位计数寄存器

**寄存器 112H: PWM2占空比重载寄存器 (PR2L)**

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>PR2L &lt;7&gt;</b>	<b>PR2L &lt;6&gt;</b>	<b>PR2L &lt;5&gt;</b>	<b>PR2L &lt;4&gt;</b>	<b>PR2L &lt;3&gt;</b>	<b>PR2L &lt;2&gt;</b>	<b>PR2L &lt;1&gt;</b>	<b>PR2L &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            **PR2L <7:0>**: PWM2占空比寄存器低字节

**寄存器 192H: PWM2占空比重载寄存器 (PR2H) (从动)**

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>PR2H &lt;7&gt;</b>	<b>PR2H &lt;6&gt;</b>	<b>PR2H &lt;5&gt;</b>	<b>PR2H &lt;4&gt;</b>	<b>PR2H &lt;3&gt;</b>	<b>PR2H &lt;2&gt;</b>	<b>PR2H &lt;1&gt;</b>	<b>PR2H &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            **PR2L <7:0>**: PWM2占空比寄存器低字节 (从动)

**寄存器 119H: PWM2占空比重载寄存器 (EPR2)**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

EPR2H3	EPR2H2	EPR2H1	EPR2H0	EPR2L3	EPR2L2	EPR2L1	EPR2L0
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

bit7-4 EPR2H, PWM2 占空比重载寄存器高 4 位 (从动)

bit3-0 EPR2L, PWM2 占空比重载寄存器高 4 位

## 寄存器 113H: PWM3占空比重载寄存器 (PR3L)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PR3L <7>	PR3L <6>	PR3L <5>	PR3L <4>	PR3L <3>	PR3L <2>	PR3L <1>	PR3L <0>
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 PR3L &lt;7:0&gt;: PWM3占空比寄存器低字节

## 寄存器 193H: PWM3占空比重载寄存器 (PR3H) (从动)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PR3H <7>	PR3H <6>	PR3H <5>	PR3H <4>	PR3H <3>	PR3H <2>	PR3H <1>	PR3H <0>
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 PR3H &lt;7:0&gt;: PWM3占空比寄存器低字节 (从动)

## 寄存器 199H: PWM3占空比重载寄存器 (EPR3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EPR3H3	EPR3H2	EPR3H1	EPR3H0	EPR3L3	EPR3L2	EPR3L1	EPR3L0

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

bit7-4      EPR3H, PWM3 占空比重载寄存器高 4 位 (从动)

bit3-0      EPR3L, PWM3 占空比重载寄存器高 4 位

**寄存器 114H: PWM4占空比重载寄存器 (PR4L)**

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>PR4L &lt;7&gt;</b>	<b>PR4L &lt;6&gt;</b>	<b>PR4L &lt;5&gt;</b>	<b>PR4L &lt;4&gt;</b>	<b>PR4L &lt;3&gt;</b>	<b>PR4L &lt;2&gt;</b>	<b>PR4L &lt;1&gt;</b>	<b>PR4L &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      PR4L &lt;7:0&gt;: PWM4占空比寄存器低字节

**寄存器 194H: PWM4占空比重载寄存器 (PR4H) (从动)**

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>PR4H &lt;7&gt;</b>	<b>PR4H &lt;6&gt;</b>	<b>PR4H &lt;5&gt;</b>	<b>PR4H &lt;4&gt;</b>	<b>PR4H &lt;3&gt;</b>	<b>PR4H &lt;2&gt;</b>	<b>PR4H &lt;1&gt;</b>	<b>PR4H &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      PR4H &lt;7:0&gt;: PWM4占空比寄存器低字节 (从动)

## 寄存器 11AH: PWM4占空比重载寄存器 (EPR4)

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>EPR4H3</b>	<b>EPR4H2</b>	<b>EPR4H1</b>	<b>EPR4H0</b>	<b>EPR4L3</b>	<b>EPR4L2</b>	<b>EPR4L1</b>	<b>EPR4L0</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

bit7-4 EPR4H, PWM4 占空比重载寄存器高 4 位 (从动)

bit3-0 EPR4L, PWM4 占空比重载寄存器高 4 位

## 寄存器 117H: PWM2/3/4周期寄存器低字节 (T2PRL)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
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 T2PRL &lt;7:0&gt;: PWM2/3/4周期寄存器低字节

## 寄存器 197H: PWM2/3/4周期寄存器高字节 (T2PRLH)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
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 T2PRLH &lt;7:0&gt;: PWM2/3/4周期寄存器高字节

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
T2CON	T2MOD	TMR2CS	P4EN	P3EN	P2EN	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
TMR2L	Timer2 数据寄存器低字节									0000 0000
TMR2H	Timer2 数据寄存器高字节									0000 0000
PR2L	PWM2 占空比寄存器低字节									0000 0000
PR2H	PWM2 占空比寄存器低字节 (从动)									0000 0000
EPR2	PWM2 占空比寄存器重载寄存器									0000 0000
PR3L	PWM3 占空比寄存器低字节									0000 0000
PR3H	PWM3 占空比寄存器低字节 (从动)									0000 0000
EPR3	PWM3 占空比寄存器重载寄存器									0000 0000
PR4L	PWM4 占空比寄存器低字节									0000 0000
PR4H	PWM4 占空比寄存器低字节 (从动)									0000 0000
EPR4	PWM4 占空比寄存器重载寄存器									0000 0000
T2PRLL	PWM2/3/4 周期寄存器的低字节									0000 0000
T2PRLH	PWM2/3/4 周期寄存器的高字节									0000 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	Timer1

### 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	TICS	ECCP_CS	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	<p><b>P1M&lt;1:0&gt;</b>: PWM输出配置位</p> <p>如果CCP1M&lt;3:2&gt; = 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</p> <p>如果CCP1M&lt;3:2&gt; = 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</p>
bit5	<p><b>TICS</b>: Timer1内部时钟选择位</p> <p>0 = 使用<math>F_{osc}/4</math>内部时钟  1 = 使用<math>F_{osc}</math>系统时钟</p>
bit4	<p><b>ECCP_CS</b>: Timer1PWM/比较/捕获输出IO口选择</p> <p>1 = PWM在PA0口输出  0 = PWM在PC5口输出</p>
bit3-0	<p><b>CCP1M&lt;3:0&gt;</b>: ECCP模式选择位</p> <p>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</p>

## 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-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CCPR1<7	CCPR1<6	CCPR1<5	CCPR1<4	CCPR1<3	CCPR1<2	CCPR1<1	CCPR1<0
>	>	>	>	>	>	>	>
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<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-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CCPR1<15	CCPR1<14	CCPR1<13	CCPR1<12	CCPR1<11	CCPR1<10	CCPR1<9	CCPR1<8
>	>	>	>	>	>	>	>
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

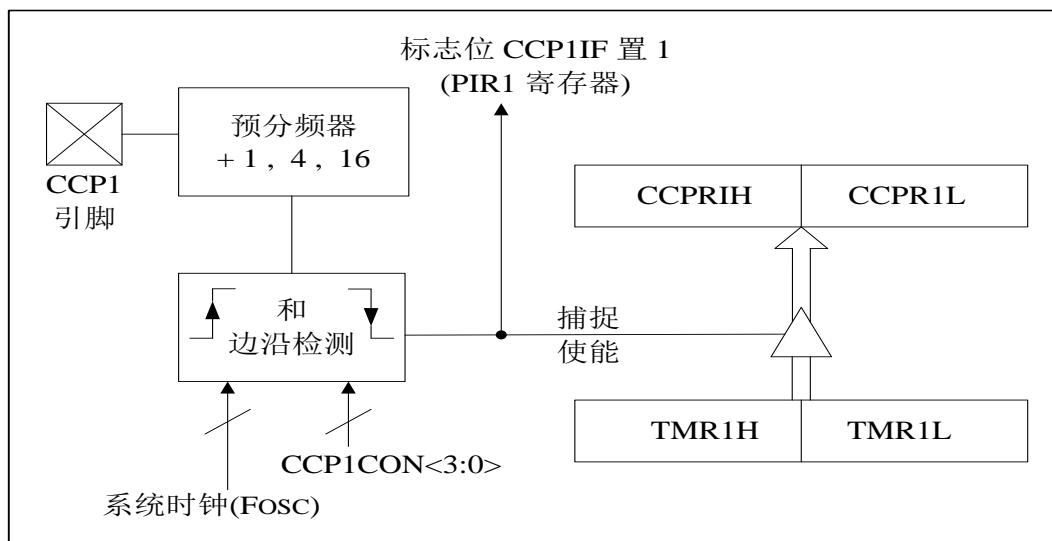
bit7-0      **CCPR1<15:8>**: 捕捉/比较/PWM寄存器1的高字节

### 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      ;复位 CCP 模块
LDWI  NEW_CAPT_PS ;
STWR  CCP1CON      ;设置新的 CCP 参数
```

## 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：

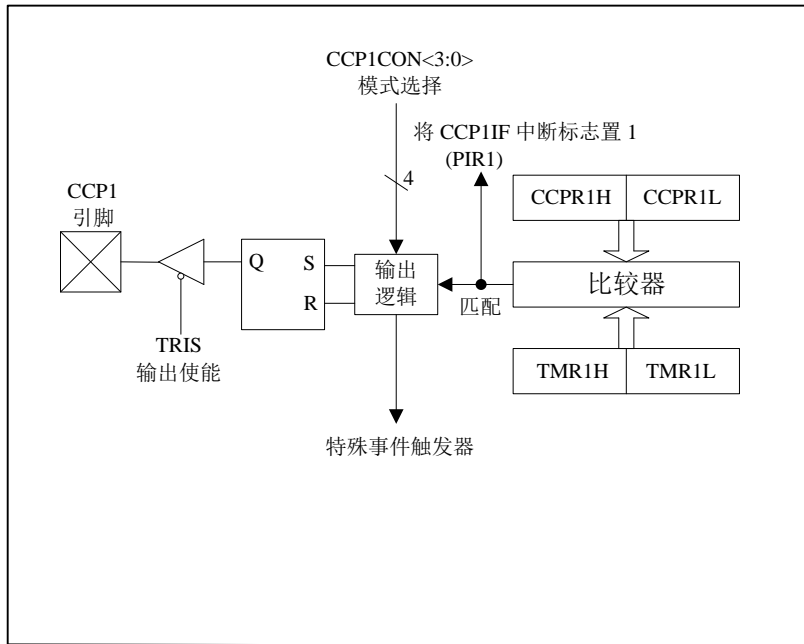
- 翻转 CCP1 输出
- 将 CCP1 输出置 1
- 将 CCP1 输出清零
- 触发特殊事件
- 产生软件中断

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

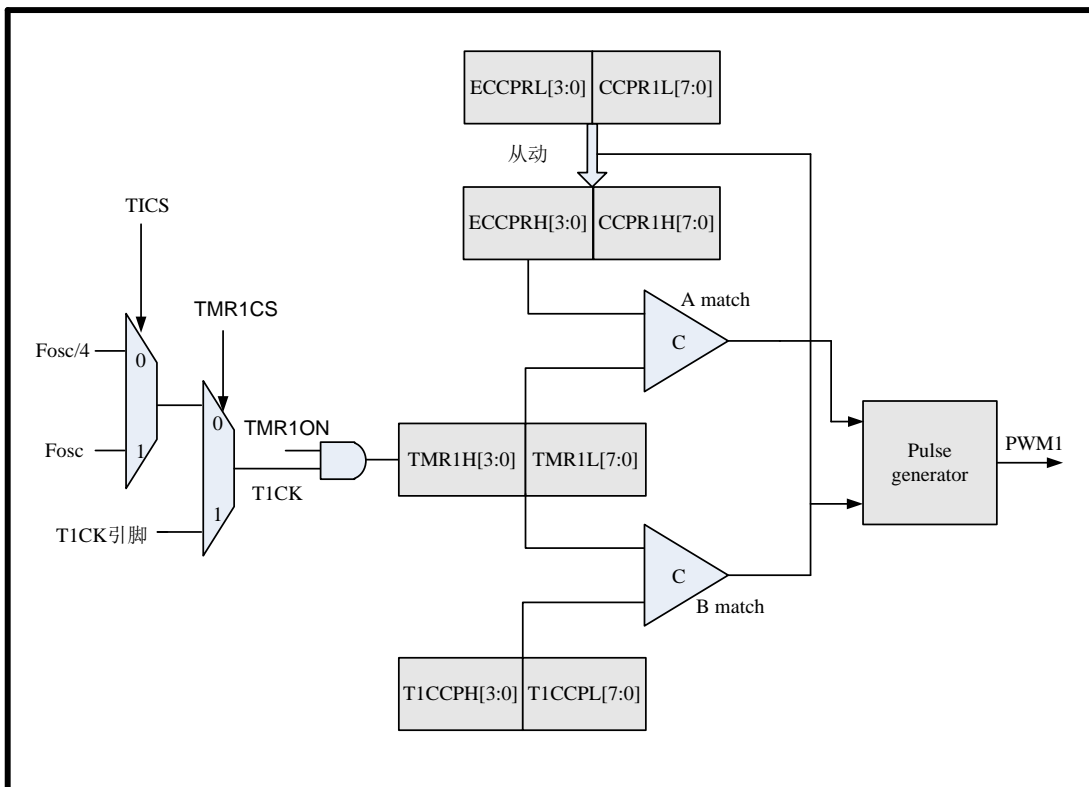
PWM 模式将在 CCP1 引脚产生脉宽调制信号。其占空比，周期和分辨率由以下寄存器决定：

- T1CCPL、T1CCPH
- ECCPR
- T1CON
- 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 12 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。

### 8.5.1 图：简化的 PWM 框图

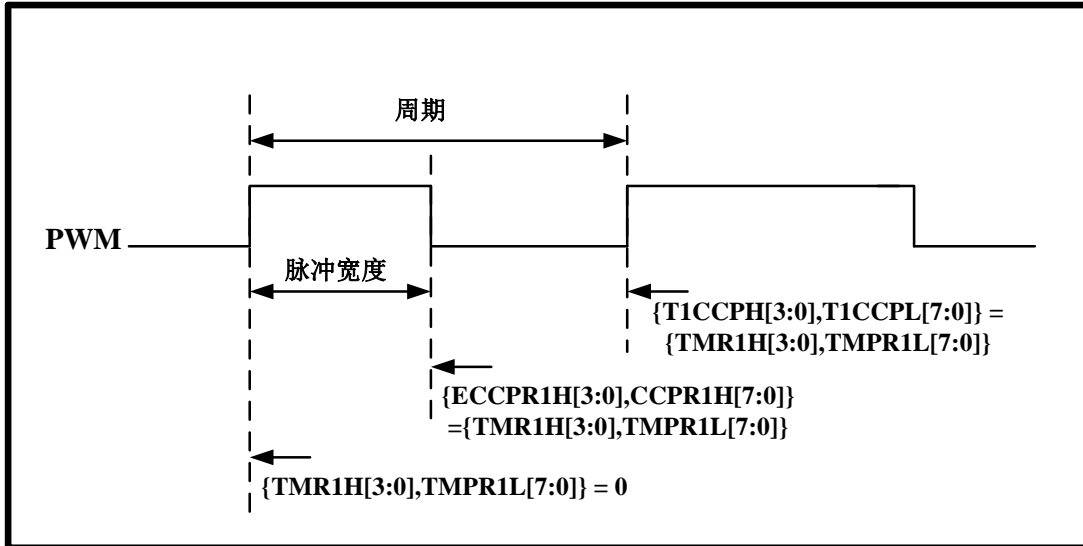
图 8-5-1: 简化的 PWM 框图



## 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 Timer1 的 T1CCPH:T1CCPL 寄存器指定。用可计算 PWM 周期。

$$\text{PWM 周期} = (T1CCP + 1) \times T1CK \times (\text{TMR1 预分频值})$$

当 { T1CCPH[3:0], T1CCPL[7:0] } 等于 { TMR1H[3:0], TMR1L[7:0] } 时，下一次递增周期将发生以下事件：

4. {TMR1H[7:0], TMR1L[7:0]} 被清零。
5. CCP1 引脚被置 1（例外：若 PWM 占空比 = 0%，引脚不被置 1）。
6. PWM 占空比从 ECCPR1L:CCPR1L 锁存到 ECCPR1H:CCPR1H。

## 8.5.4 PWM 占空比

通过对以下寄存器 CCPR1L 寄存器和 CCPR1H[3:0] 写入 12 位值可指定 PWM 占空比。CCPR1L 包含八个 LSB，CCPR1H[3:0] 包含四个 MSB，可在任何时候被写入。占空比直到周期完成时（即 T1CCPH:T1CCPL 和 TMR1H:TMR1L 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。

$$\text{PWM 脉冲宽度} = (\{ECCPRL : CCPRL\}) \times T1CK \times (\text{TMR1 预分频值})$$

$$\text{PWM 占空比} = \frac{\{ECCPRL : CCPRL\}}{T1CCP + 1}$$

ECCPR1H:CCPR1H 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。

当 12 位时基与 ECCPR1H:CCPR1H 锁存器匹配时，PWM1 输出引脚被清零。

### 8.5.5 PWM 分辨率

分辨率决定某个周期的有效占空比。例如，12 位分辨率有 4096 个分立的占空比，而 8 位分辨率则有 256 个分立的占空比。

$T1CCP = \{T1CCPH:T1CCPL\}$  为 FFFH 时产生 12 位的最大 PWM 分辨率。分辨率是 CCPR1L: CCPR1H 寄存器值的函数。

PWM 分辨率：

$$\text{分辨率} = \log_2(T1CCP + 1) \text{ 位}$$

### 8.5.6 休眠模式下的工作

在休眠模式下，可使用外部晶振或时钟源信号使计数器递增，Timer1 才能在休眠模式下工作。

### 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。

### 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。

## 8.6 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 12 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 PA0/PC5、PC4、PC3 和 PC2。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。

增强型 PWM 各种模式所对应的 IO 分配表格。

ECCP 模式	P1M[1:0]	I/O 口
---------	----------	-------

		PA0	PC5	PC4	PC3	PC2
单 PWM	00	是, P1CS 选择		是	是	是
全桥正向	01	是, P1CS 选择		是	否	否
半桥 PWM	10	是, P1CS 选择		是	是	是
全桥反向	11	是, P1CS 选择		是	是	是

图 8-6-1: 增强型 PWM 模式的简化框图示例

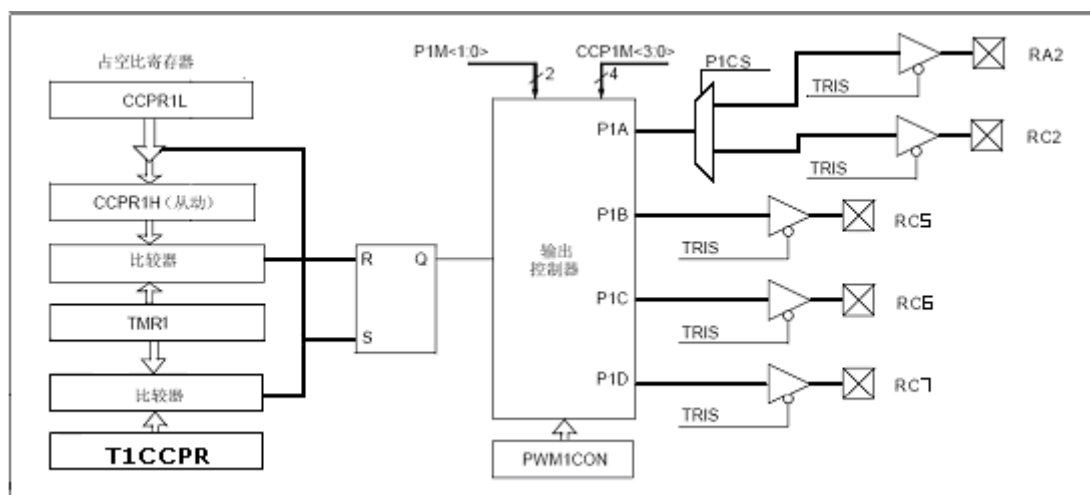


图 8-6-2: PWM (增强模式) 输出关系示例 (高电平有效状态)



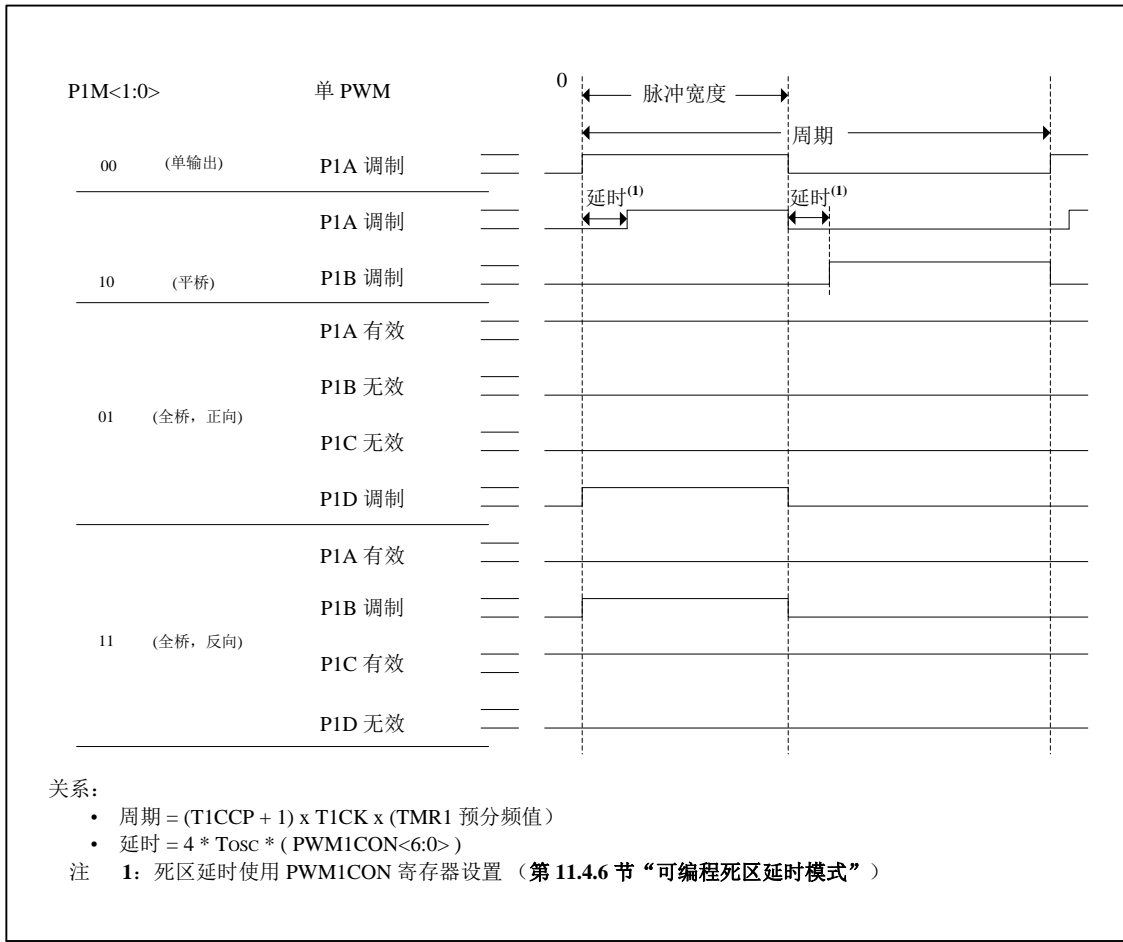


图 8-6-3: PWM (增强模式) 输出关系示例 (低电平有效状态)

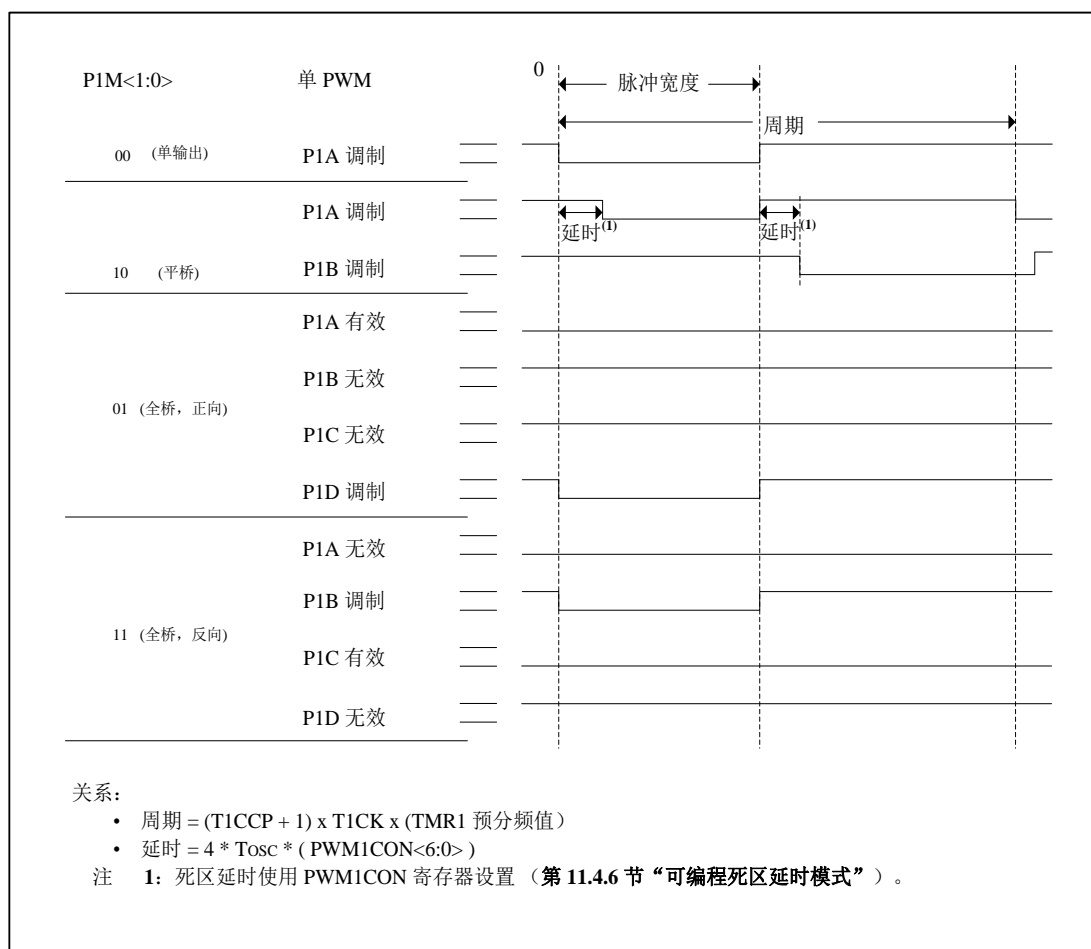


表 8-6-1: 不同 PWM 增强模式的引脚分配示例

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 PA0 或 PC5 引脚 (寄存

器选择)，而互补 PWM 输出信号被输出到 P1B 引脚（见图 8-6-1-2）。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。

在半桥模式下，如果占空比配置为 0，死区时间 PDC[6:0]≠0，则 P1B 输出状态不确定；如果占空比配置为 0，死区时间 PDC[6:0]=0，P1B 输出为高，符合设计要求。因此，在半桥模式下，如果占空比为 0，则死区时间 PDC[6:0]必须配置为 0。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

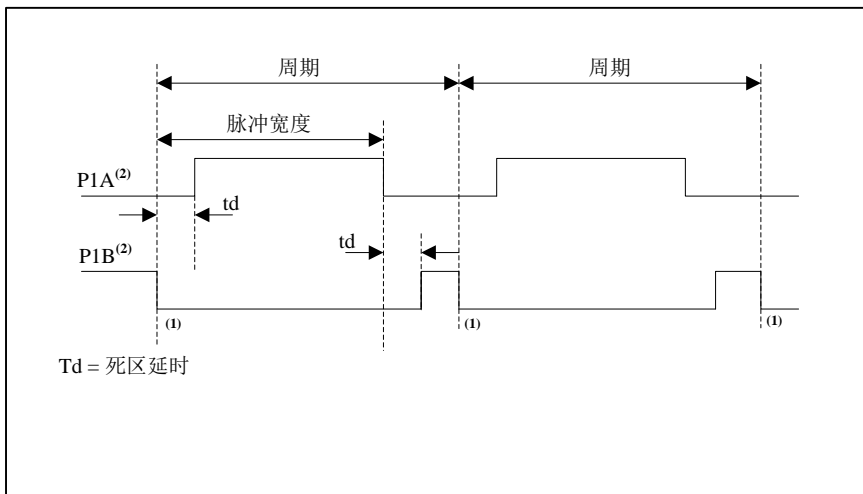
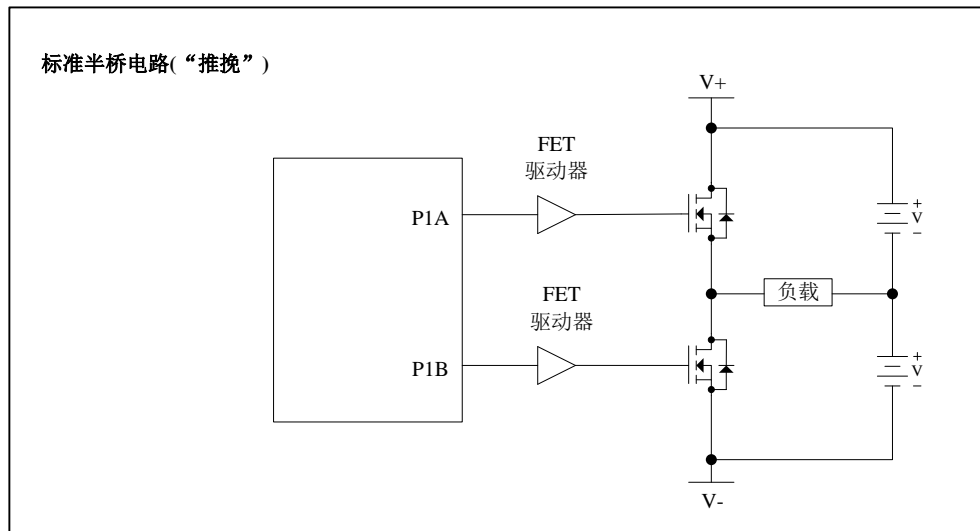


图 8-6-1-2: 半桥应用示例



## 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

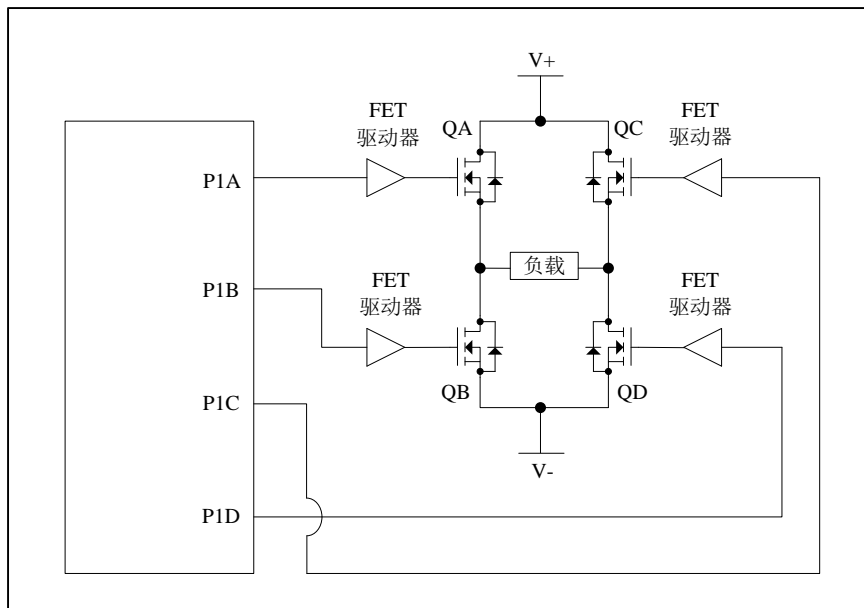
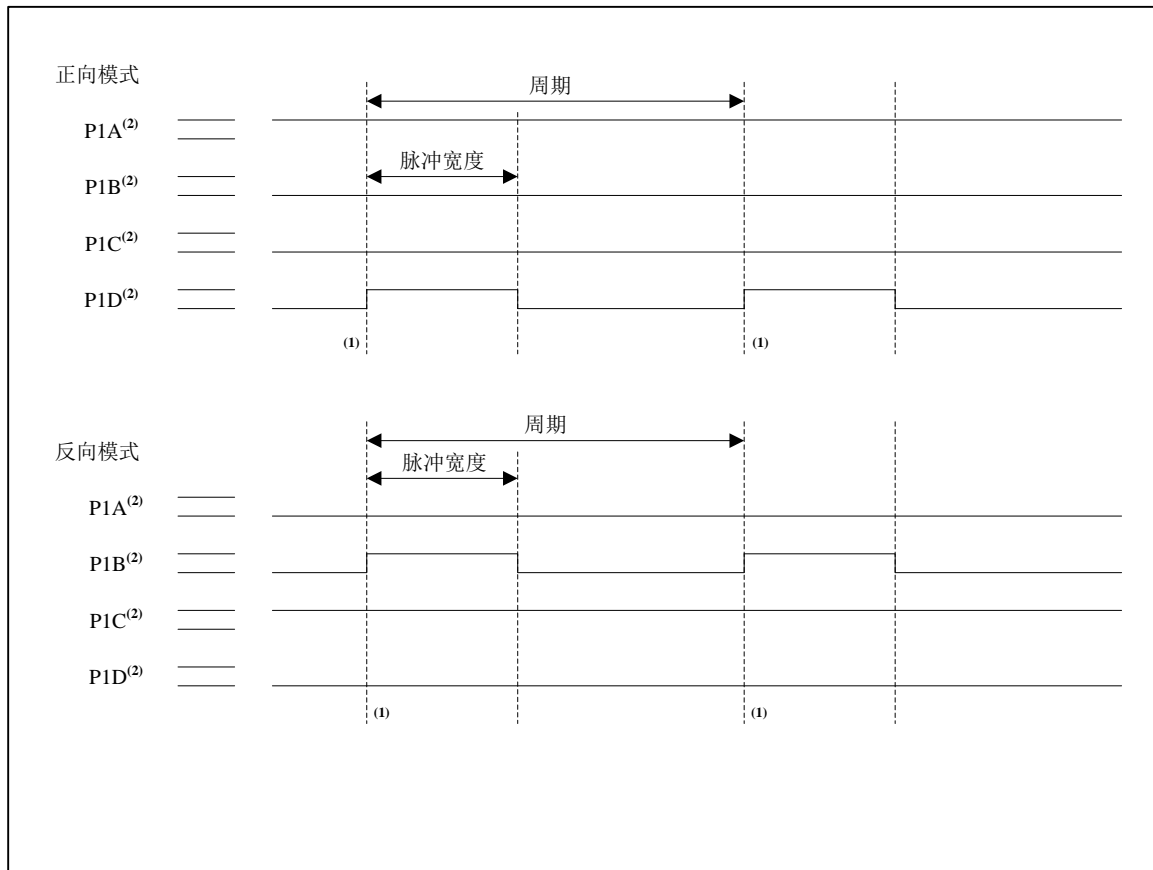


图 8-6-2-2: 全桥 PWM 输出示例



### 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 PIFB1 寄存器的 TMR2IF 位在第二个 PWM 周期开始时是否置 1 查看。

## 8.6.4 E 增强型 PWM 自动关闭模式

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

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

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

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

发生关闭事件时，将出现两个状况：

ECCPASE 位置 1。ECCPASE 保持置 1 状态直到被固件清零或发生了自动重启（见第 8.6.5 节“自动重启模式”）。

使能的 PWM 引脚被异步置于其关闭状态。PWM 输出引脚被分为两对[P1A/P1C]和[P1B/P1D]。两对引脚的状态由 ECCPAS 寄存器的 PSSAC 和 PSSBD 位决定。每对引脚均可置于以下三种状态之一：

- 驱动为逻辑 1
- 驱动为逻辑 0
- 三态（高阻态）

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>ECCPASE</b>	<b>ECCPAS2</b>	<b>ECCPAS1</b>	<b>ECCPAS0</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

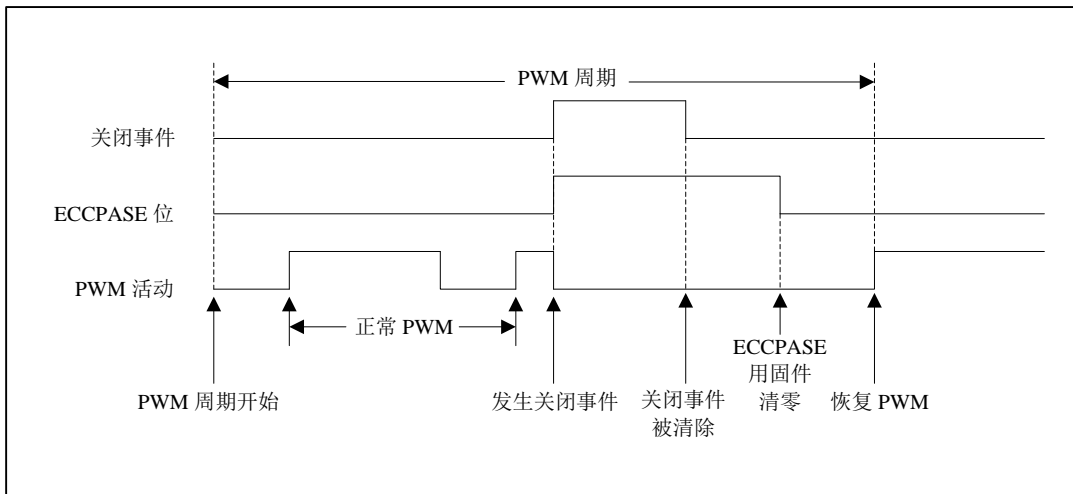
bit6-4 **ECCPAS<2:0>**: ECCP自动关闭源选择位  
000 = 禁止自动关闭  
001 = 比较器1输出变化  
010 = 比较器2输出变化  
011 = 比较器1或2之一变化  
100 = INT引脚电压为VIL  
101 = INT引脚电压为VIL或比较器1变化  
110 = INT引脚电压为VIL或比较器2变化  
111 = INT引脚电压为VIL或比较器1/2之一变化

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

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

- 注:
- 1: 自动关闭条件是基于电平的信号，而非基于边沿的信号。只要电平不变，自动关闭就不变。
  - 2: 自动关闭条件下禁止写入ECCPASE位。
  - 3: 一旦自动关闭条件被清除并且PWM重启（通过固件或自动重启），PWM信号将始终在下一个PWM周期开始时重启。

图 8-6-4: 固件重启时的 PWM 自动关闭 (PRSEN=0)

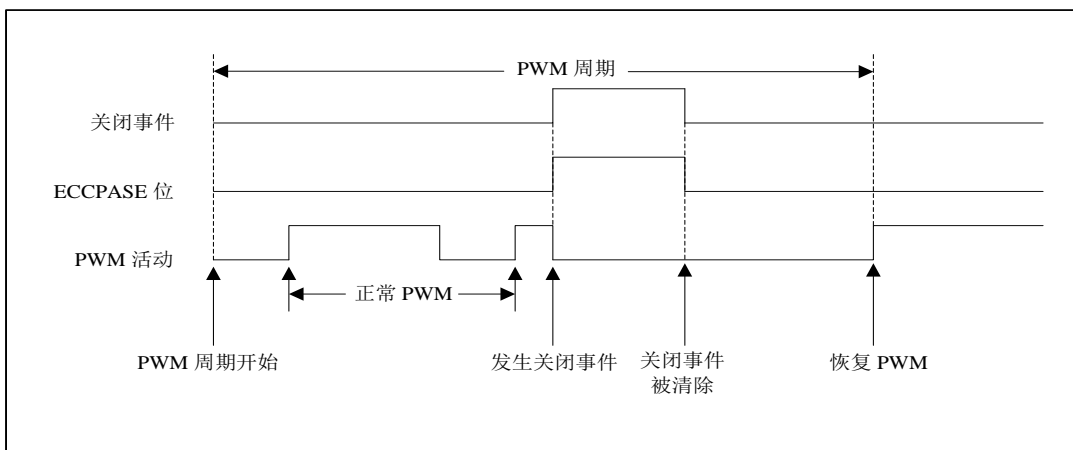


### 8.6.5 自动重启模式

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。相关 PWM1CON 寄存器（寄存器 16H）的低七位以单片机的指令周期（TCY，即 4TOSC）为单位设置延时期限。

### 寄存器 16H：增强型PWM控制寄存器（PWM1CON）

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延時計數位

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

注

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 10EH：增强型PWM周期的低8位寄存器（T1CCPL）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
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

Timer1在做PWM时周期的低八位



## 寄存器 18EH: 增强型PWM周期的高8位寄存器 (T1CCPH)

R-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
NA	NA	NA	NA	PDC3	PDC2	PDC1	PDC0
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

bit7-4 NA 读为0

Bit4-0 Timer1在做PWM时周期的高四位。

## 寄存器 1BH: 增强型PWM占空比寄存器 (ECCPR)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCPRH3	ECCPRH2	ECCPRH1	ECCPRH0	ECCPRL3	ECCPRL2	ECCPRL1	ECCPRL0
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

bit7-4 **ECCPRH<3: 0>**: Timer1在做PWM时占空比的高四位Bit4-0 **ECCPRL<3: 0>**: Timer1在做PWM时占空比重载值的高四位

表 8-6-6: 捕捉/比较/PWM 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
CCPR1L	捕捉/比较/PWM 寄存器 1 的低字节								xxxx xxxx	uuuu uuuu
CCPR1H	捕捉/比较/PWM 寄存器 1 的高字节								xxxx xxxx	uuuu uuuu
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
CMSTA	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
CMCON1	—	—	—	—	—	—	T1GSS	C2SYNC	---- --10	---- --10
ECCPAS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000	0000 0000
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	TMR1IF	0000 0000	0000 0000
ECCPR	Timer1 在做 PWM 时占空比								1111 1111	1111 1111
PWM1CON	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000	0000 0000
T1STA	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	uuuu uuuu

TMR1L	16 位 TMR1 低字节的保持寄存器								xxxx xxxx	uuuu uuuu
TMR1H	16 位 TMR1 高字节的保持寄存器								xxxx xxxx	uuuu 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)。捕捉/比较/PWM 不使用阴影单元。

## 9.0 模拟数字转换器 (ADC) 模块

SAR (Successive approximation register) ADC 为 YS10F685 系统中的逐次逼近型模数转换器, 其转换速度为 300ksps (kilo samples per second) (10bit) 或 75ksps (12bit), ADC 集成了跟踪保持电路、可编程窗口检测器。

### 9.1 SAR ADC 模块设计指标

一个支持 10bit/12bit 的 SAR ADC, 数字设计指标如下表所示 (模拟参数不列出)。

表1 SAR ADC模块设计指标列表

一级规格点	子规格点	SAR ADC 需要支持
转换启动源	写 1 到 ADC0CN 的 AD0BUSY 位	支持
	Timer1 比较模式特殊事件启动	支持
	Timer 1 溢出	支持
	Timer 2 溢出	支持
中断	转换结束中断	支持
寄存器接口	SFR 总线	支持
时钟	sys_clk, osc_16m	支持
ADC 工作时钟	0~31 分频 (时钟源: sys_clk 或 osc_16m)	支持

### 9.2 转换启动方式

有 4 种 A/D 转换启动方式, 由 ADC0CN 中的 ADC0 转换启动方式位 AD0CM[1:0]的状态决定采用哪一种方式。转换的触发源有:

- ADC0BUSY 写 1
- Timer1 比较模式特殊事件
- Timer1 溢出
- Timer2 溢出

注: ADC 转换进行中, 即使新的转换启动条件有效, ADC 并不中断当前转换, 硬件忽略新的转换启动。

以 Timer1 溢出或者 Timer2 溢出启动 ADC 的时候, 注意 ADC 转换时钟分频系数与 Timer 分频系数, 否则可能在一次溢出中引起多次 ADC 转换。

Timer1 和 Timer2 初始化配置按以下:

```
INTS = 0xc0; //打开总中断使能, 打开外设中断使能
PIEB1 |= 0x01; //打开 Timer1 中断使能
```

```
//PIEB1 |= 0x02; //打开 Timer2 中断使能
TMR1H = 0xff;
TMR1L = 0x00;
```

Timer1 和 Timer2 的中断程序配置如下:

```
PIFB1 &= 0xfe; //清 Timer1 中断标志位
//PIEB1 &= 0xfd; //清 Timer2 中断标志位
TMR1H = 0xff;
TMR1L = 0x00
```

## 9.2.1 跟踪方式

每次 ADC0 转换之前都必须有一个最小的跟踪时间，以保证转换结果准确。在启动条件有效后，convert\_start 立即有效，需插入 3 个 adc\_clk 时钟（用于跟踪）后才开始转换。对于 10bitADC 来说，完成一次 ADC 转化需要 14 个 adc\_clk；12bitADC 完成一次转换需要 16 个 adc\_clk。

## 9.2.2 数据对齐方式

ADC 转换结果可以设置右对齐或者左对齐，寄存器 ADC0PWR 的 AD0\_SJST 位来决定。

右对齐，12bit

ADC0H	0	0	0	0	BIT 11	BIT 10	BIT9	BIT8
-------	---	---	---	---	-----------	-----------	------	------

ADC0L	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-------	------	------	------	------	------	------	------	------

右对齐，10bit

ADC0H	0	0	0	0	0	0	BIT9	BIT8
-------	---	---	---	---	---	---	------	------

ADC0L	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-------	------	------	------	------	------	------	------	------

左对齐，12bit

ADC0H	BIT 11	BIT 10	BIT9	BIT8	BIT7	BIT6	BIT5	BIT4
-------	-----------	-----------	------	------	------	------	------	------

ADC0L	BIT3	BIT2	BIT1	BIT0	0	0	0	0
-------	------	------	------	------	---	---	---	---

左对齐，10bit

ADC0H	BIT9	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2
-------	------	------	------	------	------	------	------	------

ADC0L	BIT1	BIT0	0	0	0	0	0	0
-------	------	------	---	---	---	---	---	---

### 9.2.3 增益设置

ADC0 增益可设置为 1x 和 0.5x。在 1x 模式下，从 ADC0 读取的满量程由 VREF 决定。在 0.5x 模式下，从 ADC0 读取的满量程由 VREF\*2 决定，这允许 VREF 更低或者输入电压达到 VREF~VDD。

### 9.2.4 10bit/12bit 模式

ADC 有两种工作模式，一种是 10Bit 模式，使能 10Bit 模式，只需要配置寄存器 AD0\_EN。除了 10Bit 模式之外，ADC 还可以工作在 12Bit 模式下，使能 12Bit 模式，除了配置寄存器 AD0\_EN 之外还需要配置寄存器 AD0\_12BEN 寄存器。

### 9.2.5 低功耗模式（Low Power Mode）

当 ADC 工作在低速 SAR 时钟时，SAR 转换提供了低功耗模式减少了电流消耗。配置 AD0LPM=1，使能低功耗模式。一般来说，当 SAR 转换时钟频率低于 4MHz 时，推荐使用低功耗模式。

系统进入 sleep 状态时，如果 ADC 正在执行转换，ADC 的时钟将不会立刻关闭，等待 ADC 转换完成之后，唤醒 MCU 的时钟。

### 9.2.6 ADC0 模拟多路选择器

AMUX0 选择 ADC 的输入通道。可参考寄存器 ADC0MX。

## 9.3 寄存器说明

寄存器名	地址	有效位宽	复位值	说明
ADC0CN	0X11E	8	0X00	ADC0 控制寄存器
ADC0CF	0X19E	8	0X1E	ADC0 配置寄存器
ADC0PWR	0X19F	8	0X1F	ADC0 上电时间延迟寄存器
ADC0H	0X1E	8	0X00	ADC0 数据字高字节寄存器
ADC0L	0X9E	8	0X00	ADC0 数据字低字节寄存器
ADC0MX	0X11D	4	0X00	ADC0 输入通道选择寄存器
REF0CN	0X19D	8	0X04	电压参考控制寄存器

## 9.4 ADC 寄存器定义

以下寄存器用于控制 ADC 的工作。

### 9.4.1 ADC0CN

寄存器 11EH: A/D控制寄存器

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>AD0_EN</b>	<b>AD0_12BEN</b>	<b>AD0CLK</b>	<b>AD0_BUSY</b>	<b>AD0_GN</b>	<b>NA</b>	<b>AD0_CM1</b>	<b>AD0_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      **AD0\_EN**: ADC0使能位  
0 = ADC0 不使能 (低功耗关断)。  
1 = ADC0 使能 (使能且准备开始数据转换)
- bit6      **AD0\_12BEN**: ADC0 12Bit模式使能位。  
0 = ADC 12Bit模式不使能  
1 = ADC 12Bit模式使能  
注: 必须同时满足以下条件12bit模式才真正使能:  
1、ADC0使能;  
2、12bit模式使能;
- bit5      **AD0CLK**: ADC0时钟选择位  
0 = ADC选择系统时钟。  
1 = ADC选择内部快时钟, 即16MHz的时钟
- bit4      **AD0\_BUSY**: ADC0启动转换位  
0 = ADC不动作  
1 = 并且有AD0\_CM[1:0] == 00时, ADC启动转换
- bit3      **AD0\_BUSY**: 增益控制位  
0 = 片上PGA增益为1  
1 = 片上PGA增益为0.5
- bit2      **NA**: 可读可写
- bit1-0    **ADC0\_CM<1:0>**: ADC0启动转换模式选择, 指定ADC启动转换源  
00 = ADC0BUSY 写1。  
01 = 比较模式触发特殊事件。  
10 = Timer1 溢出  
11 = Timer2 溢出

### 9.4.2 ADC0CF

寄存器 19EH: A/D控制寄存器

R-0	R-0	R-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-0
<b>NA</b>	<b>NA</b>	<b>NA</b>	<b>ADCS4</b>	<b>ADCS3</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

bit7-5 **NA:** 读为0

Bit4-0 ADC SAR转换时钟分频系数ADCLK是ADC工作时钟。  
 CLKSAR = ADCLK/AD0SC (AD0SC=1~31), 默认为 31 分频。  
 CLKSAR = ADCLK/32 (AD0SC=0)

### 9.4.3 ADC0PWR

寄存器 19FH: A/D控制寄存器

R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
AD0_SJST	ADC0_LPM	ADCPWR5	ADCPWR4	ADCPWR3	ADCPWR2	ADCPWR1	ADCPWR0
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 **AD0\_SJST:** ADC数据对齐方式

0 = ADC 数据右对齐

1 = ADC 数据左对齐

bit6 **AD0\_LPM:** ADC0 低功耗模式使能位

0 = 低功耗模式不使能

1 = 低功耗模式使能

bit5-0 ADC0 上电时间延时时间, 设置 ADC0 从启动到开始转换的延时时间。

延时的计算公式如下:

$$T_{startup} = (AD0PWR) * sys\_clk$$

### 9.4.4 ADC0H: ADC 结果高位寄存器

寄存器 1EH: ADC结果高位寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
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

ADC结果高字节寄存器位,

注: 写数据动作使得ADCOH寄存器清零

## 9.4.5 ADC0L: ADC 结果低位寄存器

寄存器 9EH: ADC结果寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
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

ADC结果低字节寄存器位

注: 写数据动作使得ADC0L寄存器清零

## 9.4.6 ADC0MUX

寄存器 11DH: A/D控制寄存器

R-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
NA	NA	NA	NA	MUX3	MUX2	MUX1	MUX0
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

bit7-4

NA: 读为0

Bit3-0

MUX&lt;3: 0&gt;: ADC0 转换通道选择位

0000 = ADC 输入通道关断

0001 = PA0 ADC 输入通道 0

0010 = PA1 ADC 输入通道 1

0011 = PA2 ADC 输入通道 2

0100 = PA4 ADC 输入通道 3

0101 = PC0 ADC 输入通道 4

0110 = PC1 ADC 输入通道 5

0111 = PC2 ADC 输入通道 6

1000 = PC3 ADC 输入通道 7

1001 = PB0 ADC 输出通道 8

1010 = PB3 ADC 输入通道 9

1011 = PB5 ADC 输入通道 10

1100 = PB7 ADC 输入通道 11

1101 = PB7 ADC 输入通道 11

1110 = PB7 ADC 输入通道 11

1111 = PB7 ADC 输入通道 11

注：因 ADC 输入通道 0~ADC 输入通道 11 与其他模拟功能复用 PAD，当其他模拟功能打开时，对应的 ADC 输入通道屏蔽。

## 9.4.7 REF0CN

寄存器 19DH: A/D控制寄存器

R-0	R-0	R-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0
NA	NA	NA	DUMMY	REFSL1	REFSLO	IREFSL1	IREFSL0
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      **NA:** 读为0

bit4        **DUMMY**

bit3-2      **REFSL<1:0>:** 电压参考选择  
 00 = ADC0电压参考是PA1/VREF pin  
 01 = ADC0电压参考是VDD pin  
 1X = ADC0电压参考是内部电压参考

bit1-0      **REFSL<1:0>:** 当 REFSL=2'b1x 时, 内部电压参考选择 (2V/3V/4V)  
 00 = 3V  
 01 = 4V  
 1x = 2V

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
ADC0CN	ADC0EN	ADC012BEN	ADC0CLK	ADC0BUSY	ADC0GN	NA	ADC0CM1	ADC0CM0	0000 0000	0000 0000
ADC0CF	NA	NA	NA	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0001 1110	0001 1110
ADINS1	AN7	AN6	AN5	AN3	AN4	AN2	AN1	AN0	1111 1111	1111 1111
ADINS0	AN11	NA	AN10	NA	AN9	NA	NA	AN8	1010 1001	1010 1001
ADCPWR	SJST	LPM	PWR5	PWR4	PWR3	PWR2	PWR1	PWR0	0001 1111	0001 1111
ADCMUX	NA	NA	NA	NA	MUX3	MUX2	MUX1	MUX0	0000 0000	0000 0000
REFCON	NA	NA	NA	DUMMY	REFSL1	REFSL0	IREFSL1	IREFSL0	0000 0100	0000 0100
ADC0H	A/D 结果寄存器高字节								xxxx xxxx	uuuu uuuu
ADC0L	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
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	xxxx xxxx	xxxx xxxx
PORTA	—	—	PA5	PA4	PA3	PA2	PA1	PA0	00xx xxxx	00xx xxxx
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	xxxx xxxx	xxxx xxxx
CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11 1111	--11 1111
CPIOB	CPIOB7	CPIOB6	CPIOB5	CPIOB4	CPIOB3	CPIOB2	CPIOB1	CPIOB0	1111 1111	1111 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

寄存器 1AH: 比较器配置寄存器 (CMCON)							
R-0	R-0	R-0	R-0	R-0	R/W-0	R/W-1	R/W-0
—	—	—	—	—	TIMER0_ON	T1GSS	C2SYNC
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

bit7-3 未实现: 读为 0

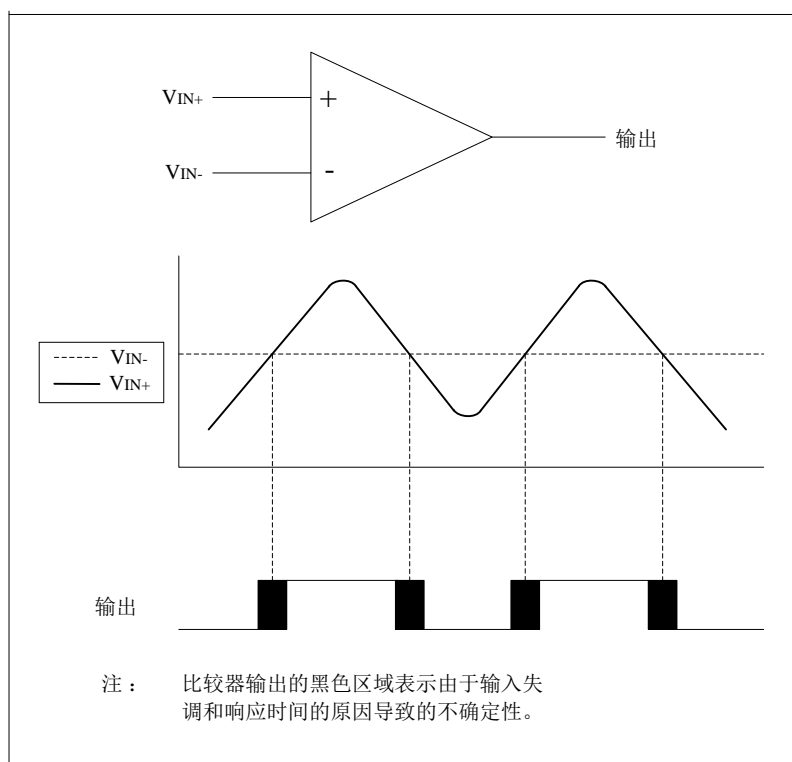
- bit2      **TIMER0\_ON**: Timer0 开关使能。  
1 = Timer0 打开;  
0 = Timer0 关闭。
- bit1      **T1GSS**: Timer1门控源选择位(1)  
1 = Timer1门控源为T1G引脚（引脚应配置为数字输入）  
0 = Timer1门控源为比较器C2的输出
- bit0      **C2SYNC**: 比较器C2输出同步位(2)  
1 = 输出与Timer1时钟的下降沿同步  
0 = 异步输出

注： 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  $V_{IN+}$  的模拟电压小于  $V_{IN-}$  的模拟电压时，比较器输出数字低电平。当  $V_{IN+}$  的模拟电压大于  $V_{IN-}$  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

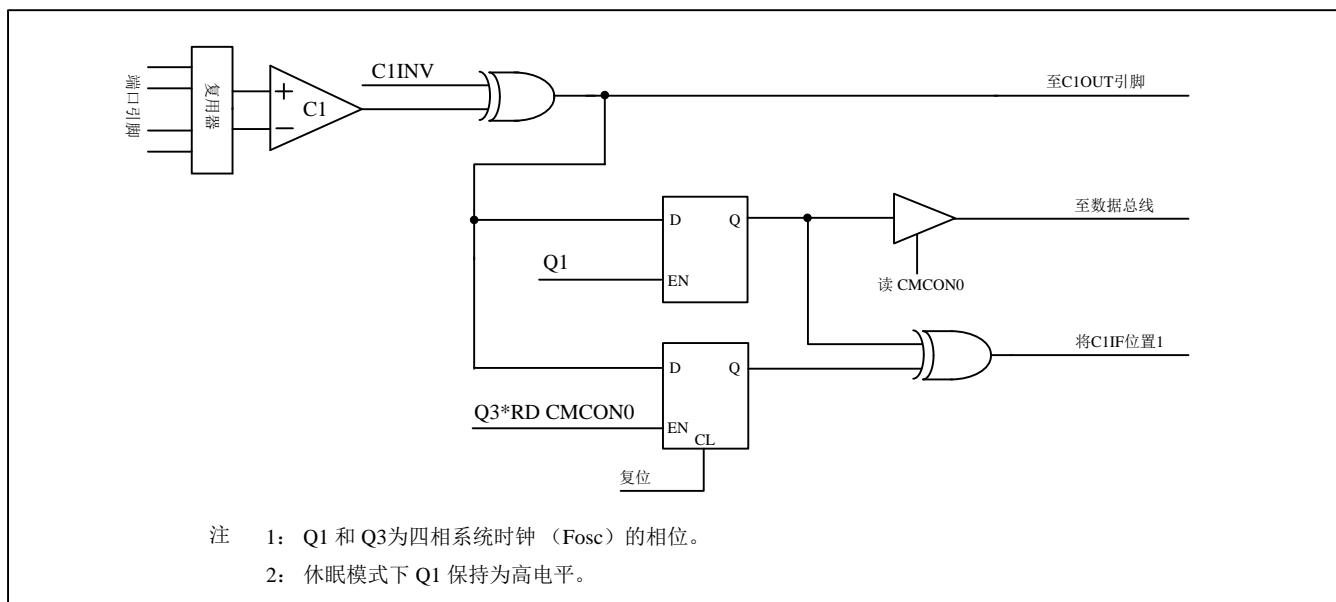
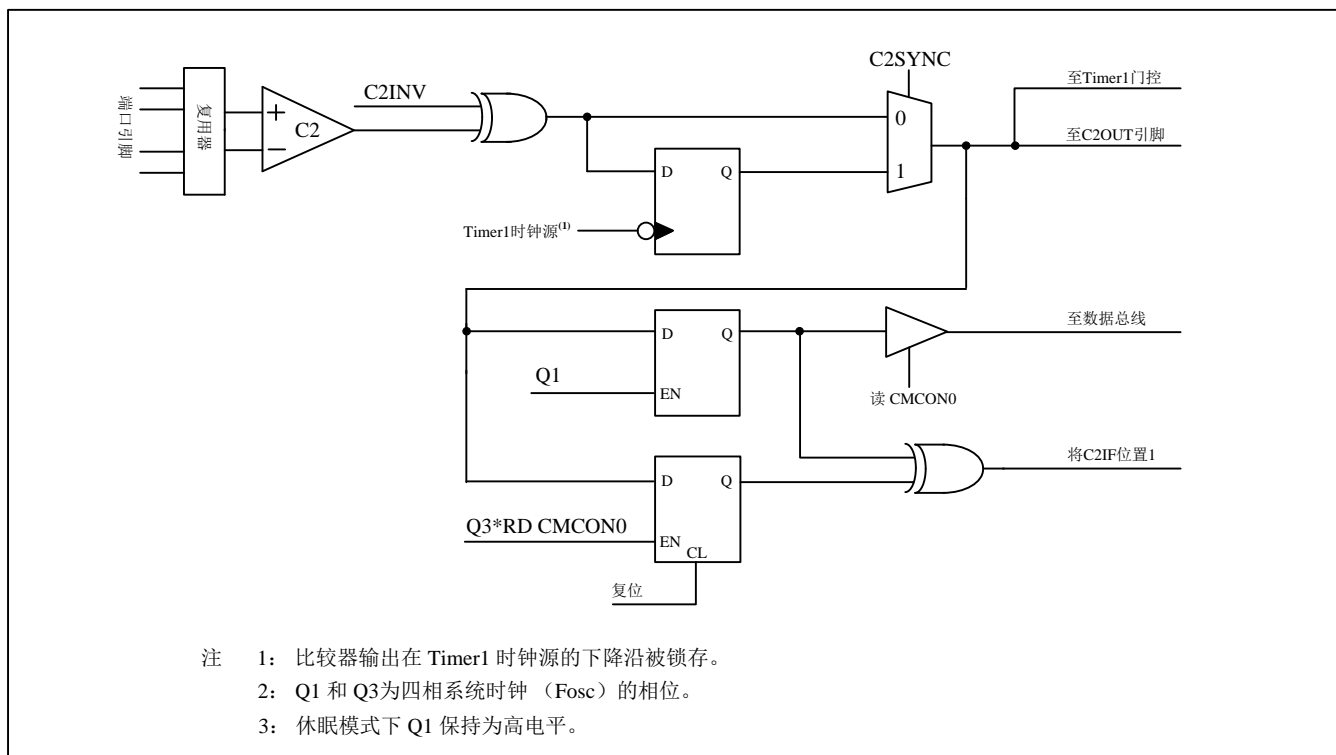


图 10-1-3: 比较器 C2 输出框图



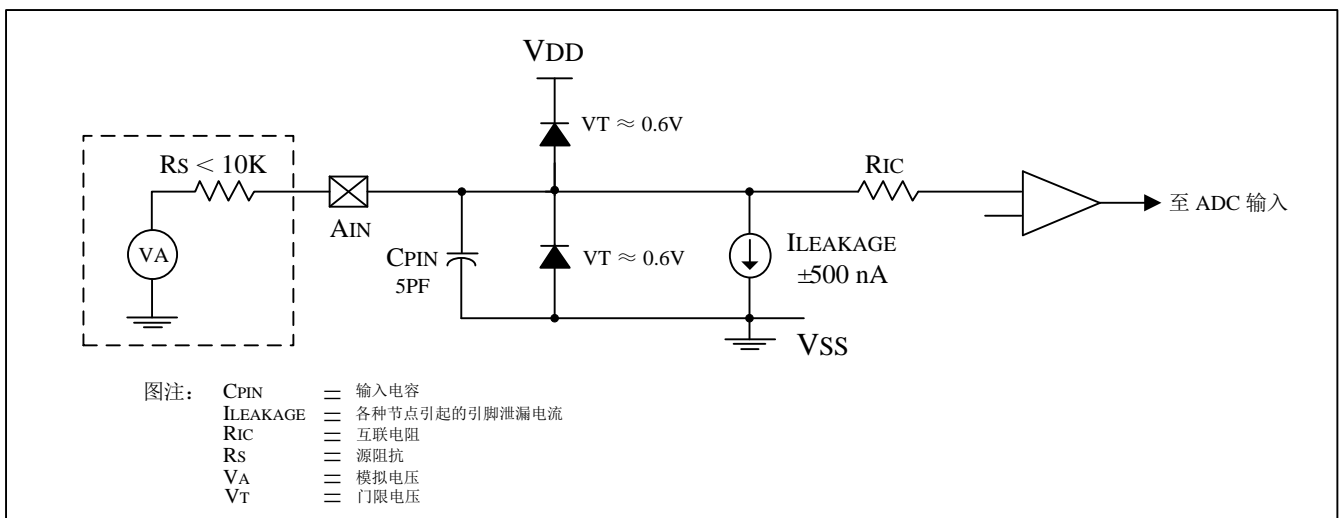
### 10.1.1 模拟输入连接考虑事项

图 10-1-1-1 所示为模拟输入的简化电路。由于模拟输入引脚与数字输入共用连接，因此它们具有连接到 VDD 和 VSS 的反向偏置 ESD 保护二极管。因此模拟输入电压必须介于 VSS 和 VDD 之间。如果输入电压与该范围的绝对值偏差超过 0.6V，其中一个二极管将正向偏置，从而可能发生闭锁。

建议为模拟输入源连接一个最大为 10kΩ 的源阻抗。另外，连接到模拟输入引脚的外部元件，如电容或稳压二极管，应具有极小的泄漏电流以将引入误差减到最小。

- 注： 1： 读取PORT寄存器时，所有配置为模拟输入的引脚均将读为0。根据输入规范，配置为数字输入的引脚将转变成模拟输入。
- 2： 在任何定义为数字输入的引脚上若有模拟电平，将导致输入缓冲器消耗超出规范的电流。

图 10-1-1-1： 模拟输入模型



## 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 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。

PIFB1 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，PIFB1 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：

- 对 CMSTA 的任何读或写操作。这将结束不匹配条件。
- 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

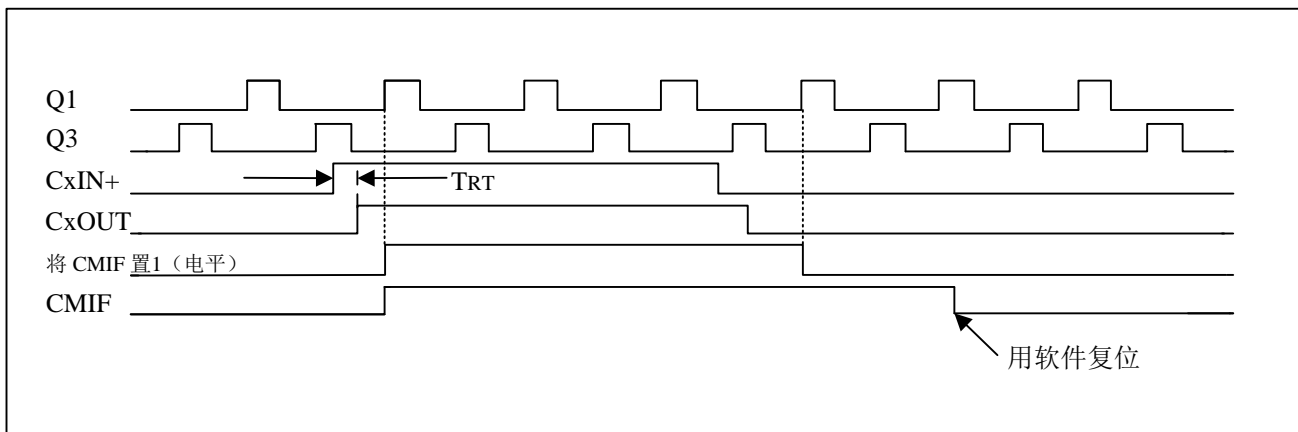
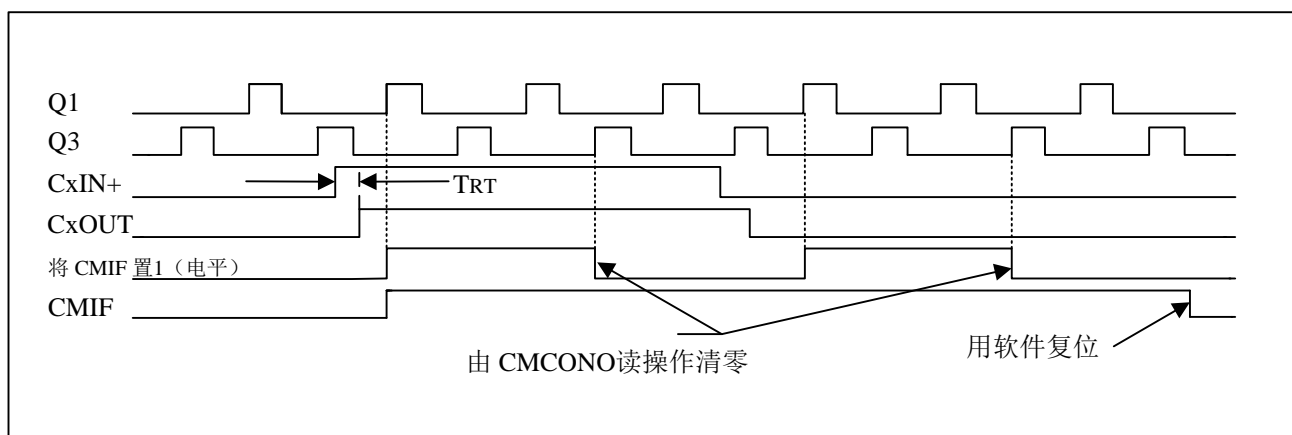


图 10-5-2：读取 CMSTA 时的比较器中断时序



- 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PIFB1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  $CM\langle 2:0 \rangle = 000$  或  $CM\langle 2:0 \rangle = 111$  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 ( $CM\langle 2:0 \rangle = 000$ )。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	<p><b>C1OUT</b>: 比较器1输出位</p> <p>C1INV = 0时:</p> <p>1 = C1VIN+&gt;C1VIN-</p> <p>0 = C1VIN+&lt;C1VIN-</p> <p>C1INV = 1时:</p> <p>1 = C1VIN+&lt;C1VIN-</p> <p>0 = C1VIN+&gt;C1VIN-</p>
bit5	<p><b>C2INV</b>: 比较器2输出翻转位</p> <p>1 = C2输出翻转</p> <p>0 = C2输出不翻转</p>
bit4	<p><b>C1INV</b>: 比较器1输出翻转位</p> <p>1 = C1输出翻转</p> <p>0 = C1输出不翻转</p>
bit3	<p><b>CIS</b>: 比较器输入开关位</p> <p>CM&lt;2:0&gt; = 010时:</p> <p>1 = C1IN+连接至C1VIN- C2IN+连接至C2VIN-</p> <p>0 = C1IN-连接至C1VIN- C2IN-连接至C2VIN-</p> <p>CM&lt;2:0&gt; = 001时:</p> <p>1 = C1IN+连接至C1VIN-</p> <p>0 = C1IN-连接至C1VIN-</p>
bit2-0	<p><b>CM&lt;2:0&gt;</b>: 比较器模式选择位 (见图8-5)</p> <p>000 = 比较器关闭。CxIN引脚配置为模拟</p> <p>001 = 三个输入复用到两个比较器</p> <p>010 = 四个输入复用到两个比较器</p> <p>011 = 两个公共参考比较器</p> <p>100 = 两个独立比较器</p> <p>101 = 一个独立比较器</p> <p>110 = 两个带输出的公共参考比较器</p> <p>111 = 比较器关闭。CxIN引脚配置为数字I/O</p>

## 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](#)）。

## 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
<b>VREN</b>	—	<b>VRR</b>	—	<b>VR3</b>	<b>VR2</b>	<b>VR1</b>	<b>VR0</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      **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

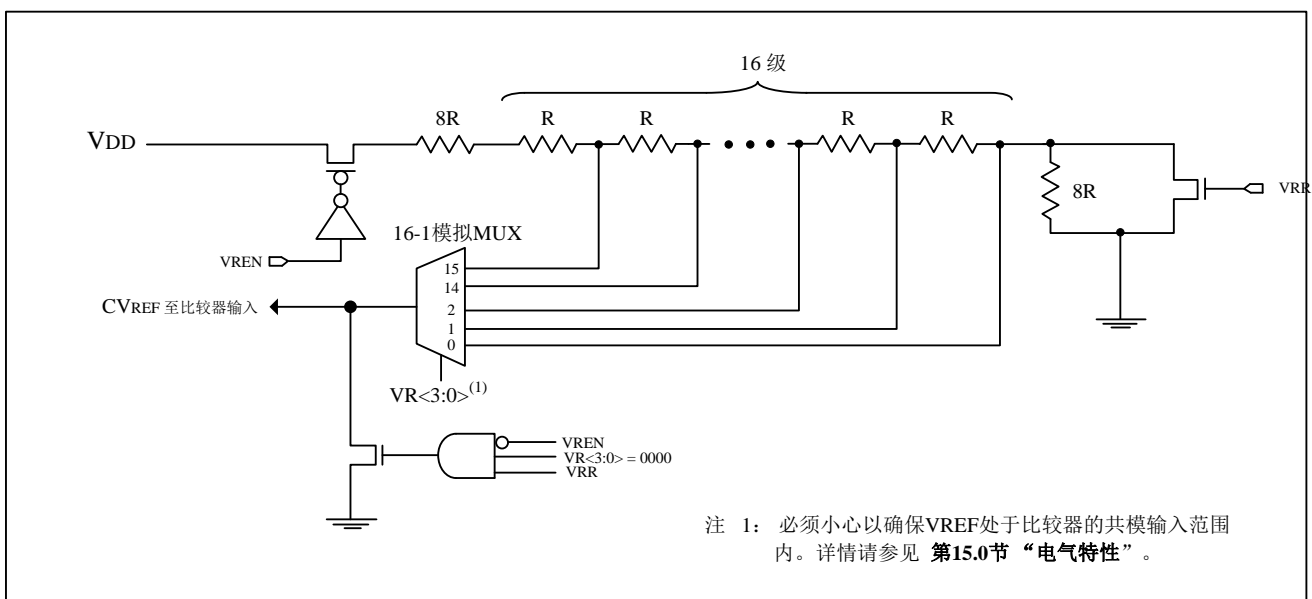


表 10-10-4: 与比较器参考电压模块相关的寄存器汇总

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
ADINS1	AN7	AN6	AN5	AN3	AN4	ANS2	ANS1	ANS0	1111 1111	1111 1111
CMSTA	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
CMCON	—	—	—	—	—	—	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 Config Option

Config Option 用于给客户提供配置选项。

#### 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 **LV DEN <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/ $\overline{\text{MCLR}}$ 脚执行MCLR功能, 是复位脚  
 0: PA3/ $\overline{\text{MCLR}}$ 脚执行PA3功能, 是数字输入引脚
- bit4 **PWRTE**: 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: INTOSCIO Oscillator模式, PA4为IO引脚, PA5为IO引脚  
 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: RC Oscillator模式, PA4输出时钟CLKOUT, PA5是时钟输入脚CLKIN  
 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

指令	功能	操作	影响状态位
NOP	空操作	无	无
CLRWT	清看门狗	$0 \rightarrow WT$	/TF, /PF
SLEEP	进入睡眠模式	$0 \rightarrow WT$ , stop OSC	/TF, /PF
TMODE	将 W 寄存器值装入选项寄存器 (81H)	$W \rightarrow 81H$	无
CPIO R	设置端口方向寄存器 (1 输入, 0 输出)	$W \rightarrow CPIO\ r$	无
STWR R	将 W 寄存器内容送 R 寄存器	$W \rightarrow R$	无
LDR R, t	读寄存器 R, 结果保存在 R(t=1) 或者 W(t=0)	$R \rightarrow t$	Z
LDWI I	立即数送 W 寄存器	$I \rightarrow W$	无
SWAPR R, t	交换寄存器 R 的高低四位, 结果保存在 R(t=1) 或者 W(t=0) 中	$[R(0 \sim 3)$ $R(4 \sim 7)] \rightarrow t$	无
INCR R, t	递增寄存器 R, 结果保存在 R(t=1) 或者 W(t=0) 中	$R + 1 \rightarrow t$	Z
INCRSZ R, t	递增寄存器 R, 结果保存在 R(t=1) 或者 W(t=0) 中; 如果结果等于 0 则跳过该指令接下来的指令	$R + 1 \rightarrow t$	无
ADDWR R, t	W 寄存器与 R 寄存器相加, 结果保存在 R(t=1) 或者 W(t=0) 中	$W + R \rightarrow t$	C, HC, Z
SUBWR R, t	R 寄存器减去 W 寄存器, 结果保存在 R(t=1) 或者 W(t=0) 中	$RAW \rightarrow t$ $(R+/W+1 \rightarrow t)$	C, HC, Z
DECR R, t	递减寄存器 R, 结果保存在 R(t=1) 或者 W(t=0) 中	$R A1 \rightarrow t$	Z
DECRSZ R, t	递减寄存器 R, 结果保存在 R(t=1) 或者 W(t=0) 中; 如果结果等于 0 则跳过该指令接下来的指令	$R A1 \rightarrow t$	无
ANDWR R, t	R 寄存器与 W 寄存器做“与”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$R \cap W \rightarrow t$	Z
ANDWI I	W 寄存器与立即数 I 做“与”操作, 结果保存到 W 寄存器中	$I \cap W \rightarrow W$	Z
IORWR R, t	R 寄存器与 W 寄存器做“或”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$R \cup W \rightarrow t$	Z
IORWI I	W 寄存器与立即数 I 做“或”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$I \cup W \rightarrow W$	Z
XORWR R, t	R 寄存器与 W 寄存器做“异或”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$R \oplus W \rightarrow t$	Z
XORWI I	W 寄存器与立即数 I 做“异或”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$I \oplus W \rightarrow W$	Z
COMR R, t	R 寄存器“取反”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$/R \rightarrow t$	Z
RRR R, t	R 寄存器循环“右移”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$R(n) \rightarrow R(n-1)$ , $C \rightarrow R(7)$ , $R(0) \rightarrow C$	C
RLR R, t	R 寄存器循环“左移”操作, 结果保存在 R(t=1) 或者 W(t=0) 中	$R(n) \rightarrow r(n+1)$ , $C \rightarrow R(0)$ , $R(7) \rightarrow C$	C
CLRW	W 寄存器清 0	$0 \rightarrow W$	Z
CLRR R	R 寄存器清 0	$0 \rightarrow R$	Z
BCR R, b	R 寄存器的第 b 位清 0	$0 \rightarrow R(b)$	无
BSR R, b	R 寄存器的第 b 位置 1	$1 \rightarrow R(b)$	无
BTSC R, b	如果 R 寄存器的第 b 位为 0, 则跳过该指令接下来的指令	Skip if R(b)=0	无
BTSS R, b	如果 R 寄存器的第 b 位为 1, 则跳过该指令接下来的指令	Skip if R(b)=1	无
LCALL N	在整个 2K 区域内的调用指令	$N \rightarrow PC$ , $PC+1 \rightarrow Stack$	无
LJUMP N	在整个 2K 区域内的跳转指令	$N \rightarrow PC$	无
RTIW I	带立即数从子程序返回	$Stack \rightarrow PC$ , $I \rightarrow W$	无
ADDWI I	W 寄存器与立即数 I 相加, 结果保存到 W 中	$PC+1 \rightarrow PC$ , $W+I \rightarrow W$	C, HC, Z

SUBWI I	立即数 I 减去 W 寄存器，结果保存到 W 寄存器中	I-W→W	C, HC, Z
RTFI	中断返回	Stack→PC, I→GIS	无
RET	从子程序返回	Stack→PC	无

## 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°C 至+85°C
储存温度.....	-65°C 至+150°C
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 <sub>IK</sub> (V <sub>I</sub> <0 或 V <sub>I</sub> >VDD).....	±20mA
输出钳位电流, I <sub>OK</sub> (V <sub>O</sub> <0 或 V <sub>O</sub> >VDD).....	±20mA
任一 I/O 引脚的最大输出灌电流.....	40mA
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。

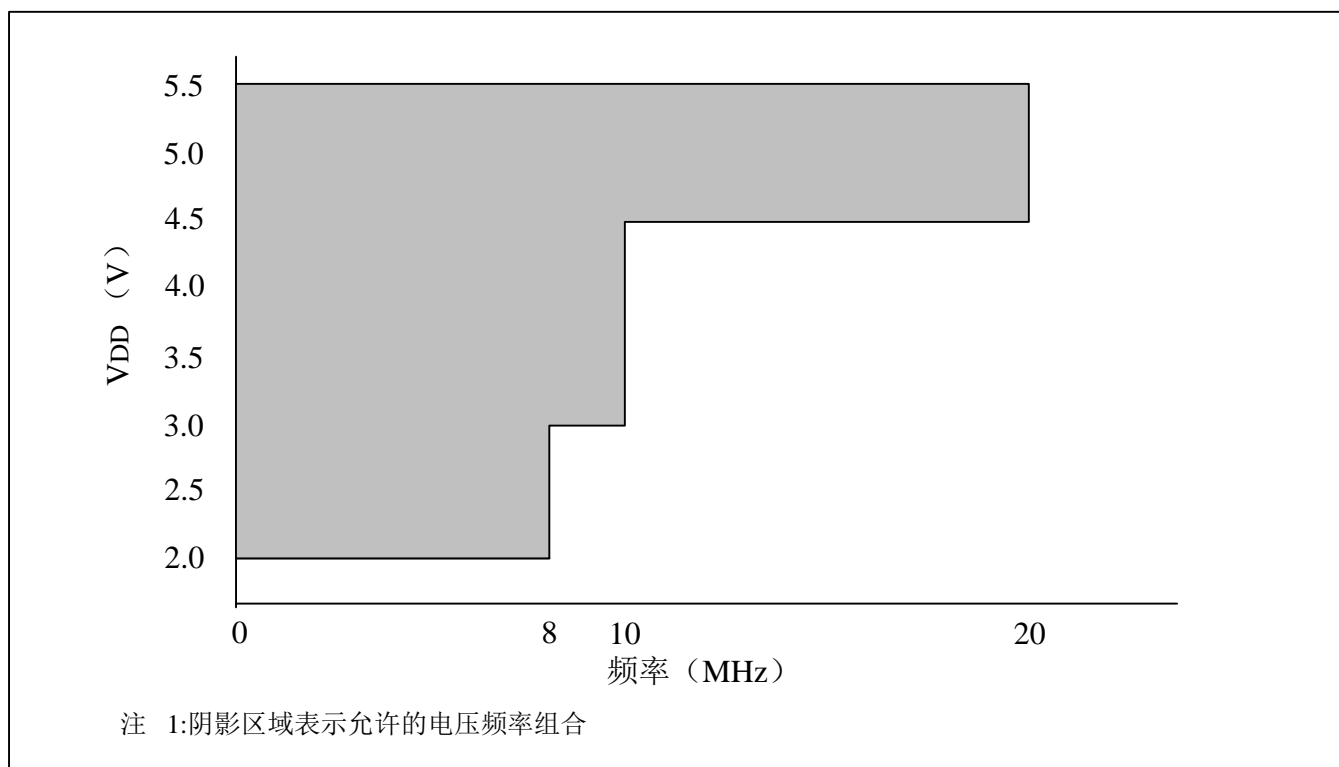
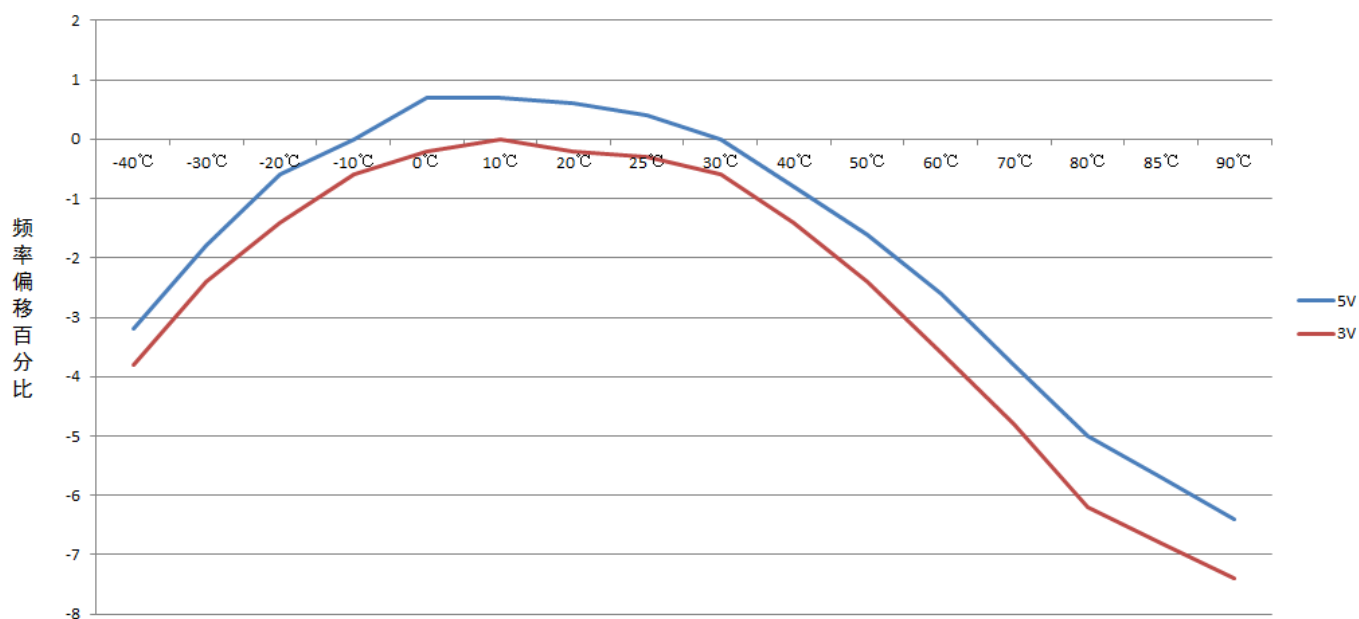
图 13-1-1: MDT10F685 电压-频率图,  $-40\text{ }^{\circ}\text{C} \leq T_A \leq +85\text{ }^{\circ}\text{C}$ 

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 $-40^{\circ}\text{C} \leq \text{Ta} \leq 85^{\circ}\text{C}$						
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	器件处于掉电状态		
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>	—	400	—	uA	3.3V	1M	WDT En PED Dis IRC
			580				4M	
			800				8M	
			1400				16M	
			550			5V	1M	
			900				4M	
			1300				8M	
			2400				16M	
IPD	掉电流 <sup>(4)</sup>	—	1	—	uA	WDT Disable VDD=2.5V		
$\Delta\text{IWDT}$	WDT 电流 <sup>(4)</sup>	—	1	—	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	—	18	—	K	3V		
		—	12	—		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。

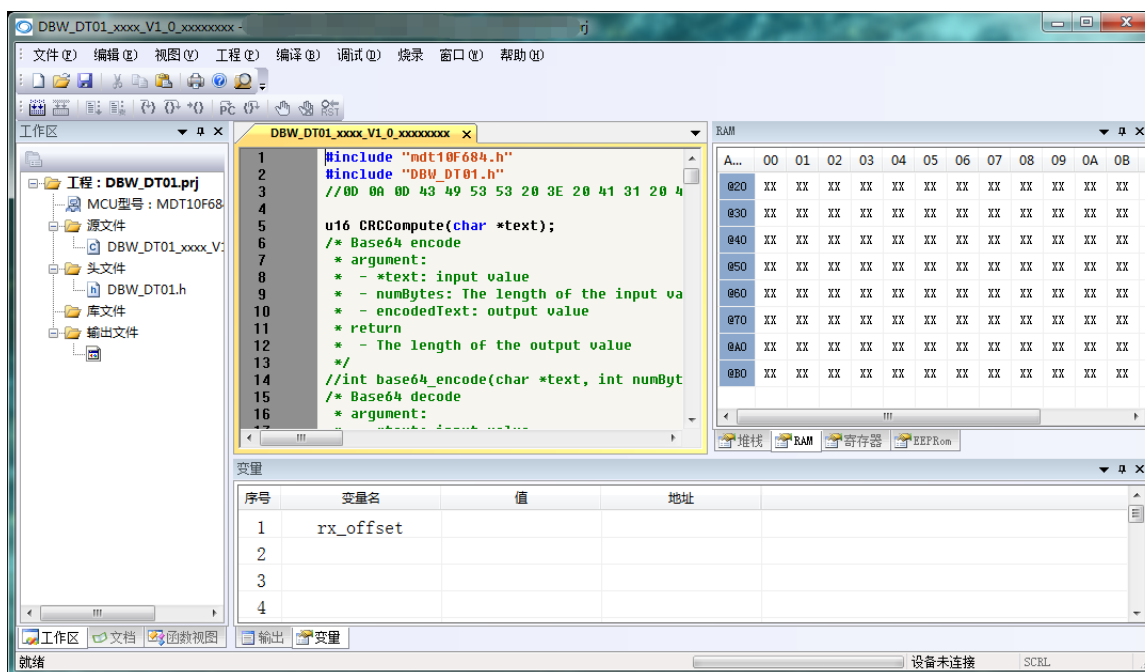
## 13.3 交流电气特性

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	16M	Hz	5V
F <sub>LIRC</sub>	内部低速时钟	—	32K	—	Hz	5V
T <sub>EERD</sub>	EEPROM 读时间	—	1	—	Tins	指令周期
T <sub>EEWR</sub>	EEPROM 写时间	—	1.5ms	2ms		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
	系统启动时间(由 SLEEP 模式唤醒,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

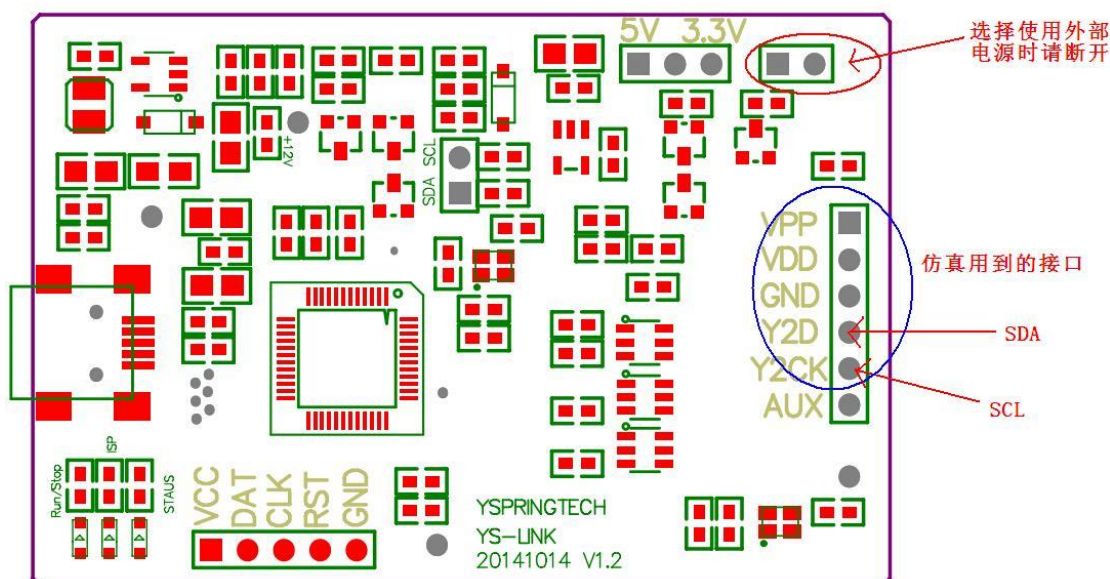
## 14.0 开发支持

### 14.1 仿真信息

#### 14.1.1 软件：MDT-IDE



#### 14.1.2 硬件：YS-Link



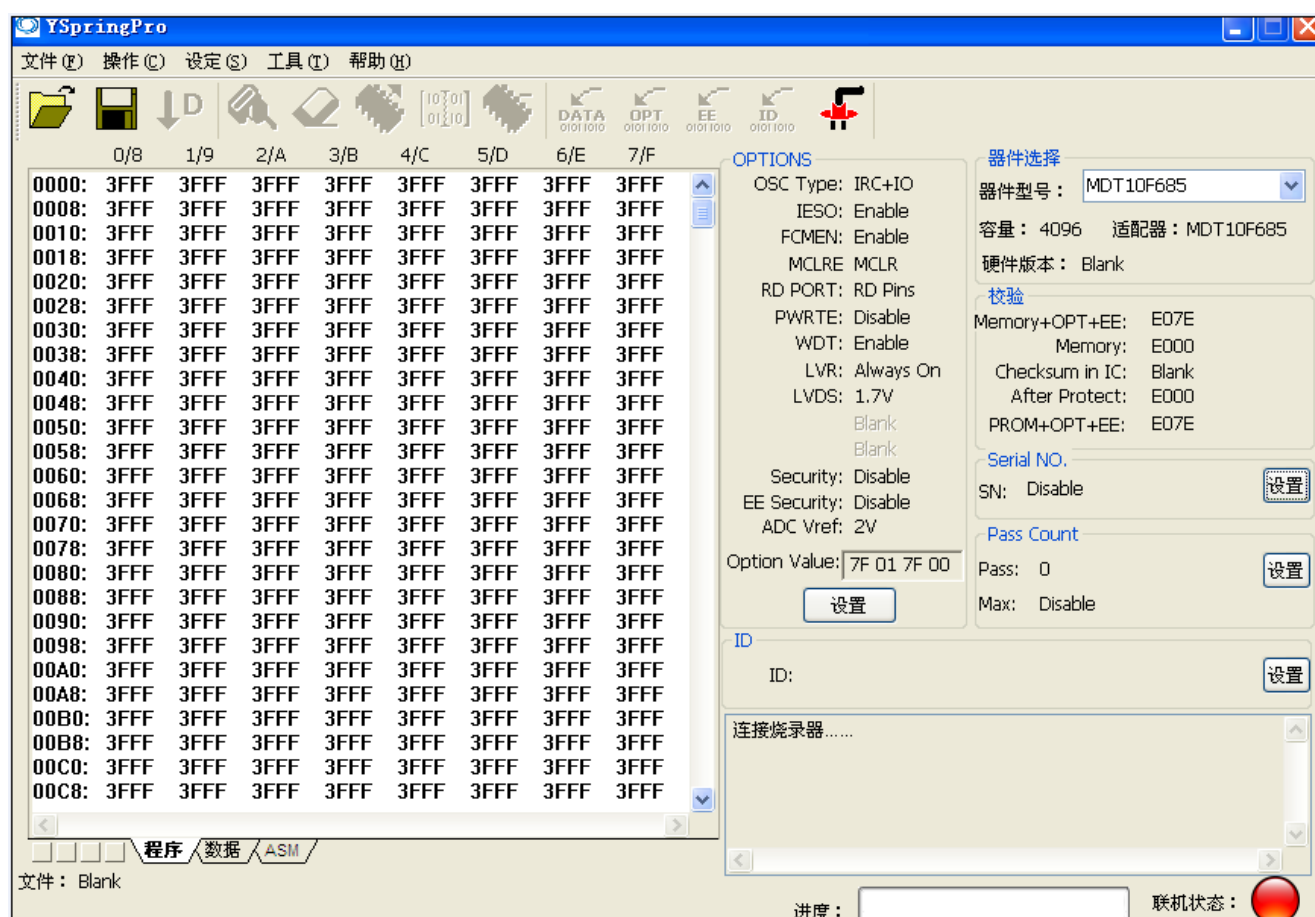
### 14.1.3 接口:

MDT10F685 具备片内仿真功能，可通过串口模式仿真调试程序，串口接口如下：

MDT10F685	YS-Link
VDD	VDD
GND	GND
PA3	VPP
PA0	SCK
PA1	SDA

## 14.2 烧录信息

### 14.2.1 烧录软件：YSpringPro

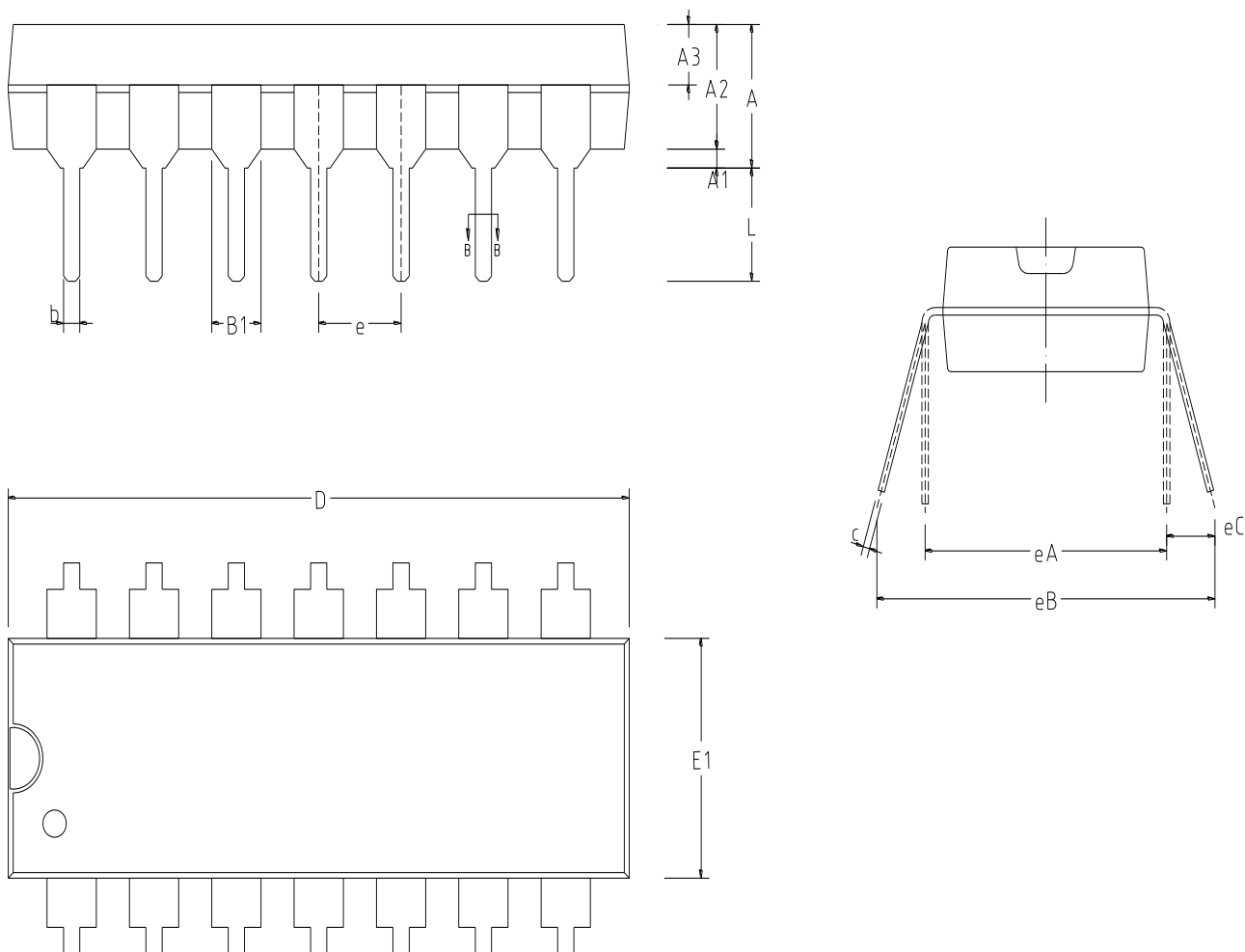


## 14.2.2 烧录器：YS-Writer



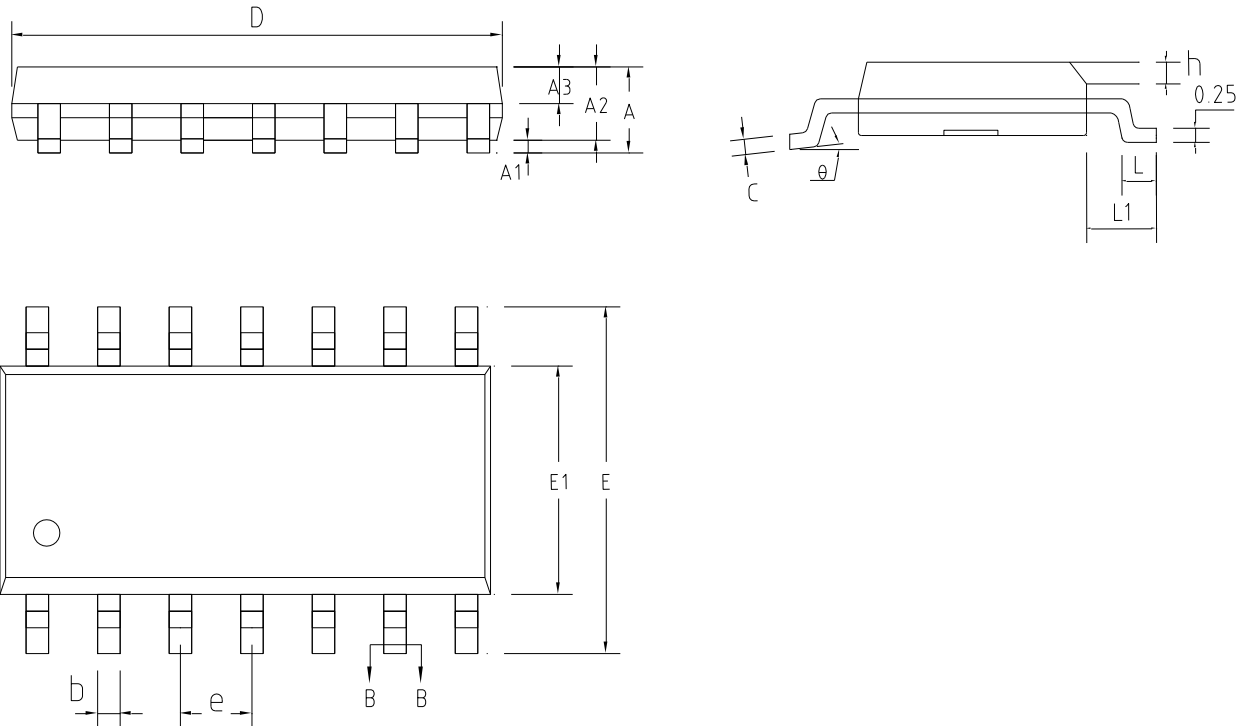
## 15.0 封装信息

### 15.1 DIP 14 PIN



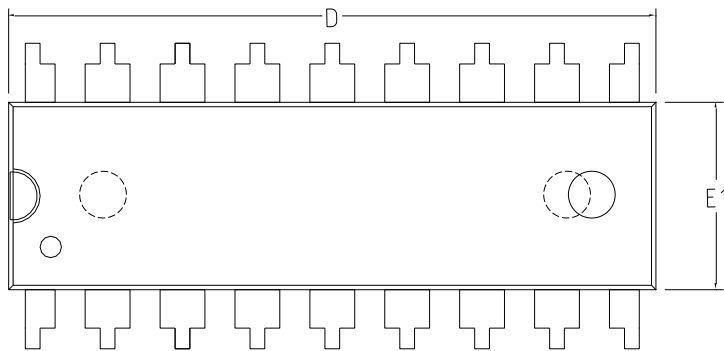
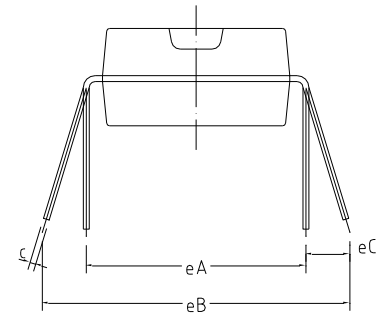
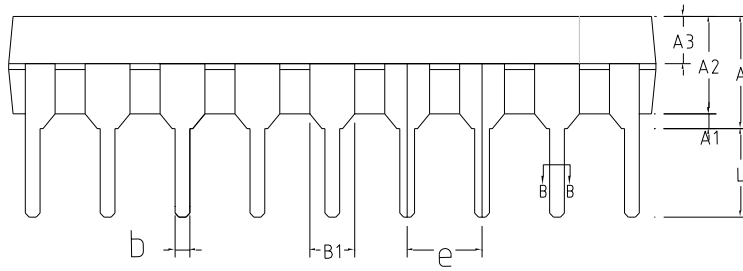
SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	3.60	3.80	4.00	E1	6.25	6.35	6.45
A1	0.51	—	—	e	2.54BSC		
A2	3.20	3.30	3.40	eA	7.62REF		
A3	1.47	1.52	1.57	eB	7.62	—	9.30
b	0.44	—	0.52				
B1	1.52BSC			eC	0	—	0.84
D	19.00	19.10	19.20	L	3.00	—	—

## 15.2 SOP 14 PIN



SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	—	—	1.75	E1	3.80	3.90	4.00
A1	0.05	—	0.225	e	1.27BSC		
A2	1.30	1.40	1.50	h	0.25	—	0.50
A3	0.60	0.65	0.70	L	0.50	—	0.80
D	8.55	8.65	8.75	L1	1.05REF		
E	5.80	6.00	6.20	θ	0	—	8°
L/F 载体 尺寸(mil)	70*70 90*110	98*150	100.4*210	b	0.39	—	0.47

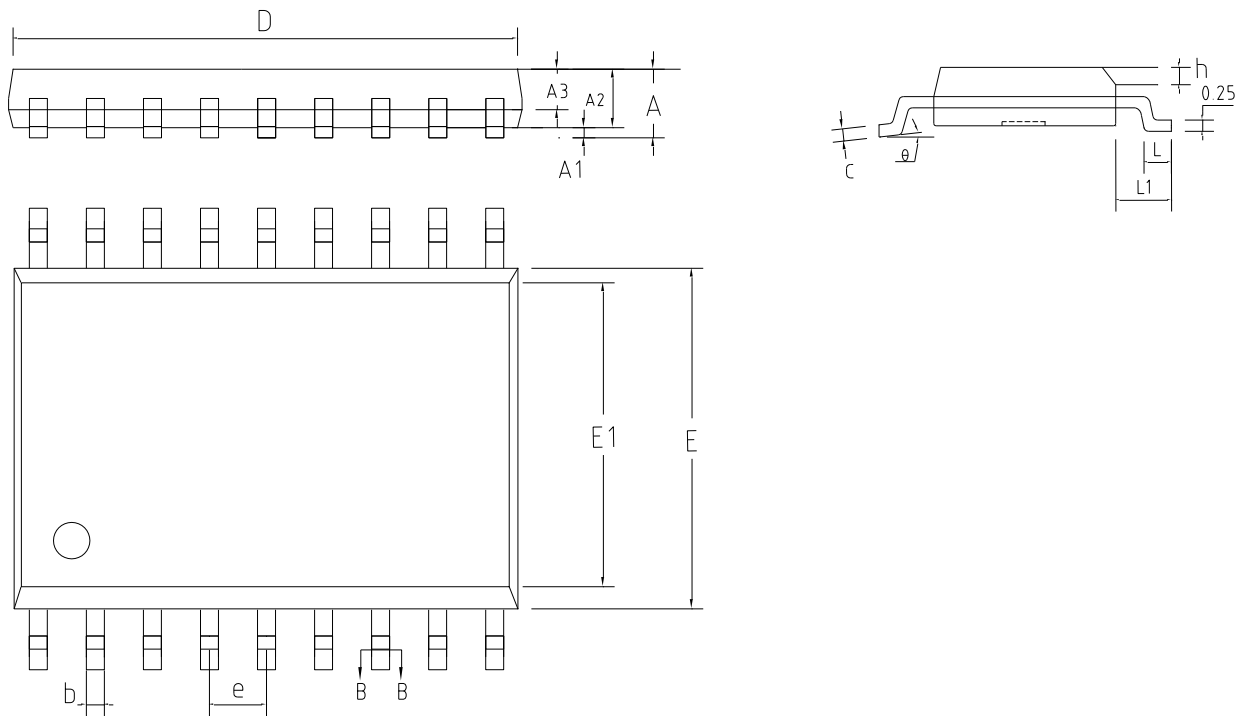
## 15.3 DIP 18 PIN



SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	3.60	3.80	4.00	E1	6.35	6.55	6.75
A1	0.51	—	—	e	2.54BSC		
A2	3.20	3.30	3.40	eA	7.62BSC		
A3	1.47	1.52	1.57	eB	7.62	—	9.30
b	0.44	—	0.53				
B1	1.52BSC			eC	0	—	0.84
D	22.70	22.90	23.10	L	3.00	—	—

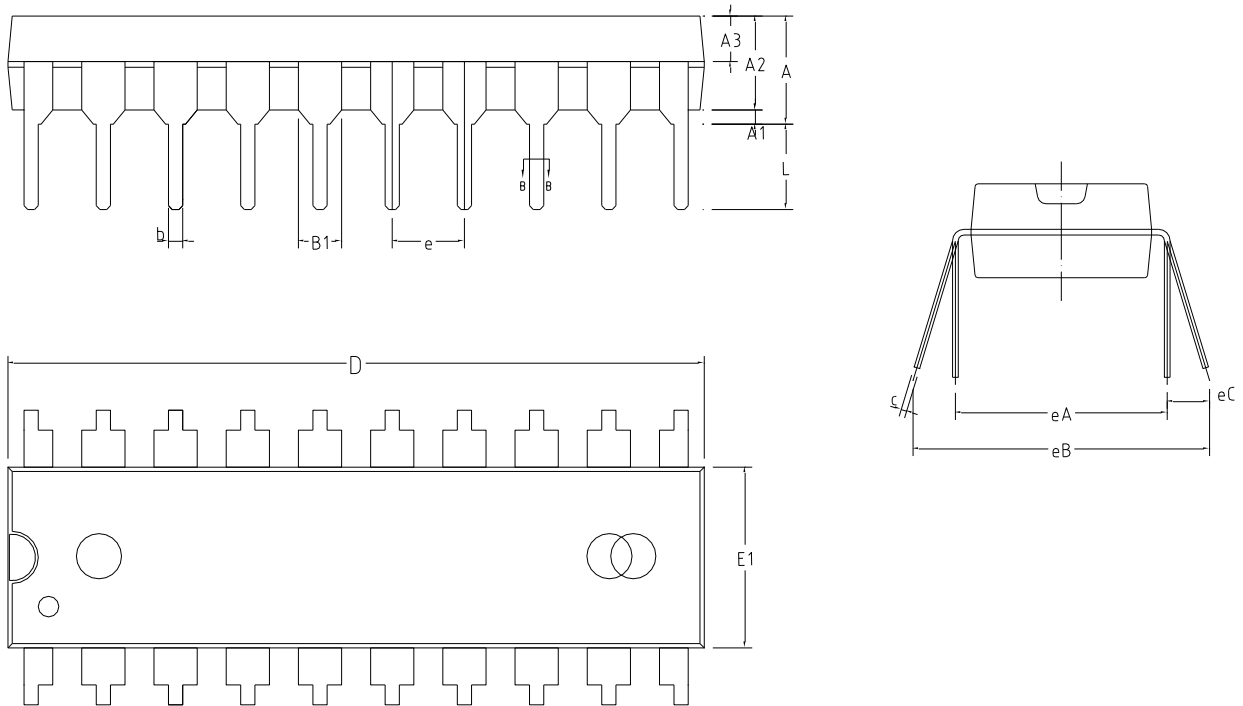


## 15.4 SOP 18 PIN



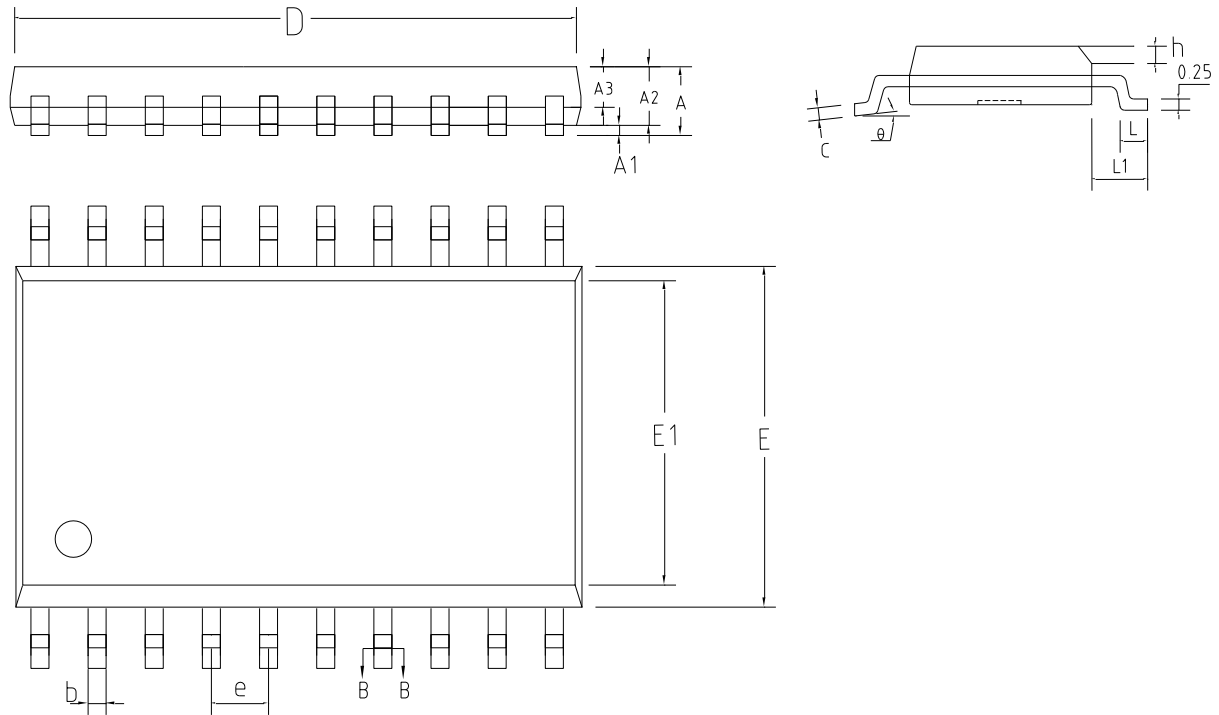
SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	—	—	2.65	E1	7.40	7.50	7.60
A1	0.10	—	0.30	e	1.27BSC		
A2	2.20	2.30	2.40	h	0.25	—	0.75
A3	0.97	1.02	0.07	L	0.70	—	1.00
D	11.35	11.45	11.55	L1	1.40REF		
E	10.10	10.30	10.50	theta	0	—	8°
L/F 载体 尺寸(mil)	70*70 90*110	98*150	100.4*210	b	0.35	—	0.43

## 15.5 DIP 20 PIN



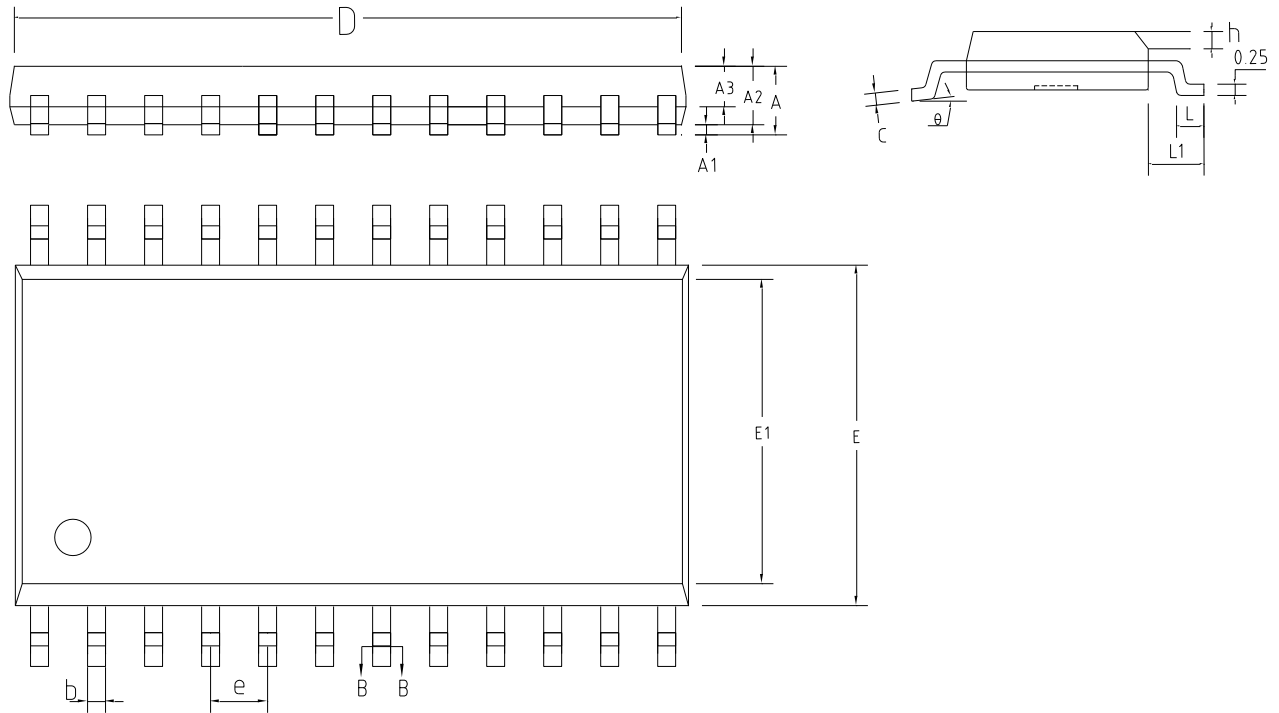
SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	3.60	3.80	4.00	E1	6.45	6.55	6.65
A1	0.51	—	—	e	2.54BSC		
A2	3.20	3.30	3.40	eA	7.62REF		
A3	1.47	1.52	1.57	eB	7.62	—	9.30
b	0.44	—	0.52				
B1	1.52REF			eC	0	—	0.84
D	25.80	25.90	26.00	L	3.00	—	—

## 15.6 SOP 20 PIN



SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	—	—	2.65	E1	7.40	7.50	7.60
A1	0.10	—	0.30	e	1.27BSC		
A2	2.25	2.30	2.35	h	0.25	—	0.75
A3	0.97	1.02	1.07	L	0.70	—	1.00
D	12.70	12.80	12.90	L1	1.40REF		
E	10.10	10.30	10.50	theta	0	—	8°
L/F 载体 尺寸(mil)	70*70 90*110	98*150	100.4*210	b	0.35	—	0.43

## 15.7 SOP 24 PIN



SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	2.36	2.54	2.64	E1	7.40	7.50	7.60
A1	0.10	0.20	0.30	e	1.27BSC		
A2	2.26	2.30	2.35	h	0.25	—	0.75
A3	0.97	1.02	1.07	L	0.70	—	1.00
D	15.30	15.40	15.50	L1	1.40REF		
E	10.10	10.30	10.50	theta	0	—	8°
L/F 载体 尺寸(mil)	70*70 90*110	98*150	100.4*210	b	0.39	—	047

## 16.0 汇春知识产权政策

### 16.1 专利权

汇春公司在全球各地区已核准和申请中之专利权至少有 20 件以上,享有绝对之合法权益。与汇春公司 MCU 或其它产品有关的专利权并未被同意授权使用,任何经由不当手段侵害汇春公司专利权之公司、组织或个人,汇春将采取一切可能的法律行动,遏止侵权者不当的侵权行为,并追讨汇春公司因侵权行为所受之损失、或侵权者所得之不法利益。

### 16.2 著作权

Copyright 2015 by INC.

规格书中所出现的信息在出版当时相信是正确的,然而汇春对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明,汇春不保证或不表示这些应用没有更深入的修改就能适用,也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。汇春产品不授权使用于救生、维生器件或系统中做为关键器件。汇春拥有不事先通知而修改产品的权利,对于最新的信息,请参考我们的网址

<http://www.yspringtech.com>;