

# MDT10F273

## 数据手册（版本 V1.2）

### 8 位 MTP CMOS 单片机

## 1 器件概述

### ◆ 存储器:

MTP 空间: 2K\*14 位。可经受 1000 次写操作。  
SRAM 空间: 128 字节。

### ◆ 8 级深硬件堆栈

### ◆ I/O 引脚配置

具有独立方向控制的 14 个 I/O 引脚: PA 口 5 个、PC 口 8 个。

PA3 做输出时, 仅能开漏输出。

高灌电流可直接驱动 LED (PA3 除外)。

PA 端口引脚电平变化中断。

PA、PC 端口独立的可编程弱上拉/下拉(PA3 除外)。

### ◆ 定时器

Timer0: 带 8 位可编程预分频器的 8 位定时器/计数器。

Timer1/2: 16bit 定时/计数器

- 带有预分频器的 16 位定时器/计数器。

- 外部 Timer1/2 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 Timer1/2 的振荡器。

### ◆ 4 路 12bit 分辨率的 PWM 模块

4 路 12bit 高分辨率 PWM 输出。

### ◆ 13+1 通道 12 位 ADC

13 个外部 ADC 输入。

内部 1/4VDD 通道输入。

多种 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 1.024V,

2.048V, 4.096V 或者 VDD 作为参考电压。

### ◆ 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 31KHz。

### ◆ 高性能的 RISC CPU

仅需学习 35 条指令。

除跳转指令外的所有指令都是单周期的。

直接、间接和相对寻址模式。

### ◆ 特殊特性

- 高精度内部振荡器, 出厂时精度校准为 +/-1%。  
偏差小于 +/-2%

- 可用软件选择的频率范围为 250kHz 到 16MHz。

- 软件可选的 31kHz 内部振荡器。

- 节能的休眠模式。

- 宽工作电压范围 (2.2V 到 5.5V)。

- 工业级温度范围 ( $-40^{\circ}\text{C} \leq T_{\text{as}} \leq +85^{\circ}\text{C}$ )。

- 上电复位 (Power-onReset, POR)。

- 上电延时定时器 (Power-upTimer, PWRT) 和振荡器起振定时器 (OscillatorStart-upTimer, OST)。

- 带软件控制选择的 LVR 低电压侦测选择(侦测电压有 2.0V、2.5V、3.0V、4.0V 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)。

- 带上拉的主复位, 可复用为输入引脚。

- 可编程代码保护。

### ◆ 低功耗特性

待机电流:

- 电压为 2.2V 时, 典型值 500nA。

工作电流:

- 频率为 8MHz、电压为 2.2V 时, 典型值为 500uA。

看门狗定时器电流:

- 电压为 2.2V 时, 典型值为 1uA。

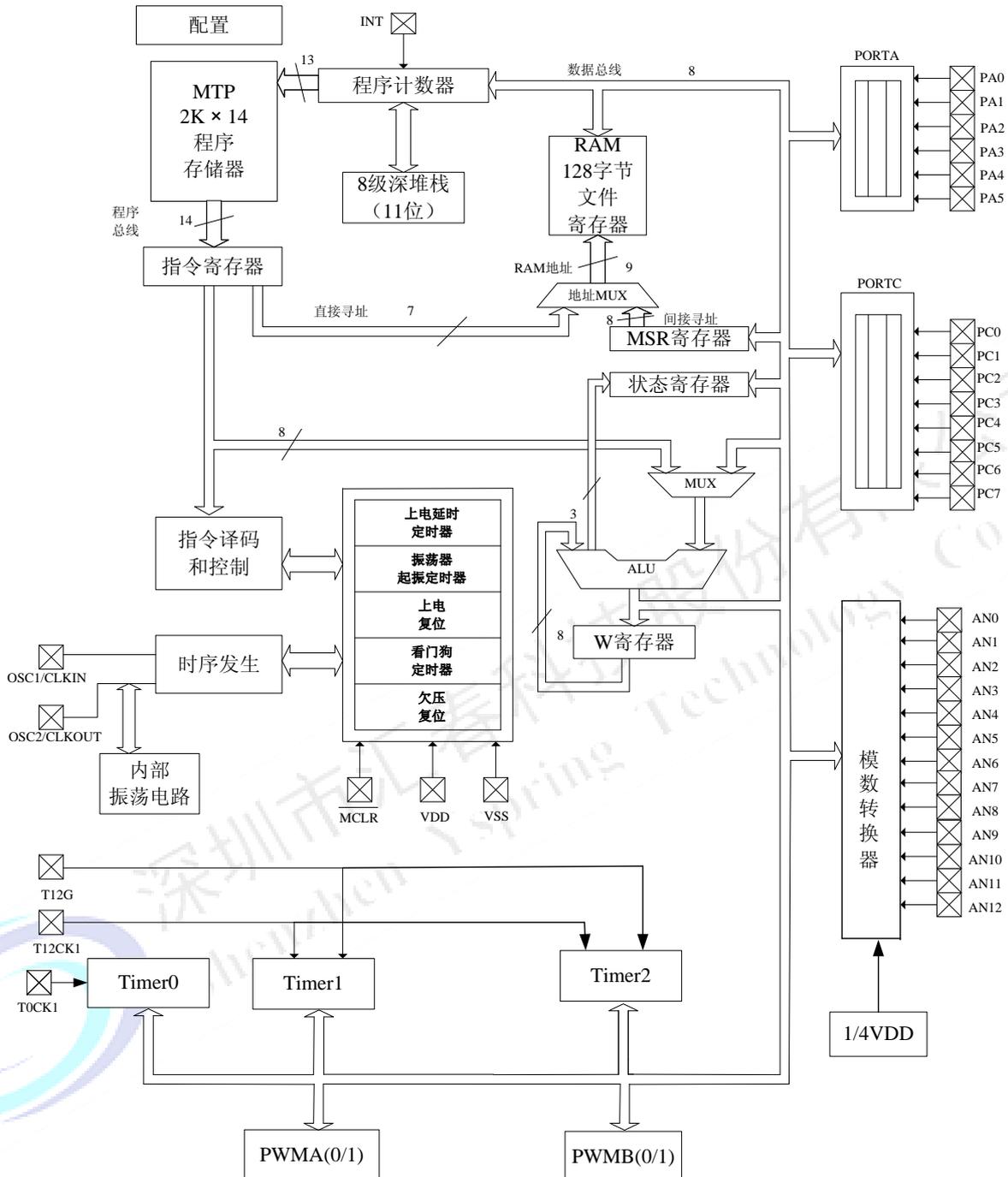
### ◆ 通过两个引脚(PA0, PA1)实现在线串行编程和调试。

器件	ROM	RAM	I/O	12位A/D转换器 (通道数)	Timer (8/16bit)	PWM 输出	唤醒功 能 引脚数 目	封装
MDT10F273	2048	128	14	13+1	1/2	4	8	SOT23-6、DFN8、SOP8、MSOP10、TSSOP14、SOP14、SOP16。

<b>1 器件概述</b> .....	<b>1</b>
1.1 系统结构图.....	4
1.2 封装脚位图.....	5
1.3 引脚说明.....	7
<b>2 存储器构成</b> .....	<b>9</b>
2.1 程序存储器构成.....	9
2.2 数据存储器构成.....	9
2.3 PCL 和 PCLATH.....	19
2.4 间接寻址、IAR 和 MSR 寄存器.....	20
2.5 看门狗定时器 (WDT).....	21
<b>3 复位</b> .....	<b>24</b>
3.1 概述.....	24
3.2 上电复位.....	25
3.3 MCLR.....	25
3.4 上电延时定时器 (PWRT).....	25
3.5 欠压复位.....	26
3.6 延时时序.....	27
3.7 电源控制 (PSTA) 寄存器.....	28
<b>4 系统时钟</b> .....	<b>29</b>
4.1 概述.....	29
4.2 振荡器控制.....	30
4.3 时钟源模式.....	30
4.4 外部时钟模式.....	31
4.5 内部时钟模式.....	33
4.6 时钟切换.....	36
4.7 双速时钟启动模式.....	36
4.8 掉电模式 (休眠).....	37
4.9 故障保护时钟监控器.....	39
4.10 内部快时钟调频功能.....	40
<b>5 中断</b> .....	<b>41</b>
5.1 中断特殊功能寄存器.....	44
5.2 外部中断.....	46
5.3 TIMER0 中断.....	47
5.4 TIMER1 中断.....	47
5.5 TIMER2 中断.....	47
5.6 PWM 中断.....	47
5.7 PORTA 电平变化中断.....	47
5.8 A/D 中断.....	47
5.6 中断的现场保护.....	47
<b>6 I/O 端口</b> .....	<b>49</b>
6.1 PORTA 和 CPIOA 寄存器.....	49
6.2 PORTC 和 CPIOC 寄存器.....	50

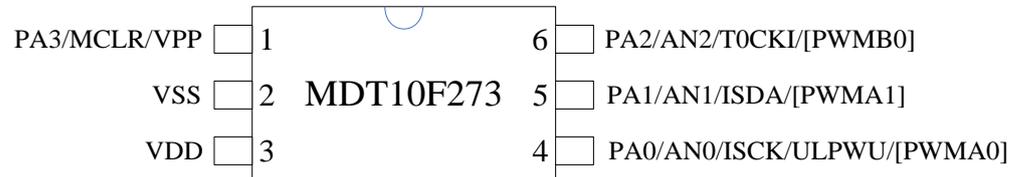
6.3 其它引脚功能 .....	51
6.4 PORTA 引脚说明和引脚图 .....	54
6.5 PORTC 引脚说明和引脚图 .....	62
<b>7 定时器模块/PWM 模块 .....</b>	<b>71</b>
7.1 TIMER0 模块 .....	71
7.2 带门控的 TIMER1/TIMER2 模块 .....	73
7.3 PWM 模块 .....	80
7.4 PWM 模块寄存器说明 .....	82
<b>8 模拟数字转换器 (ADC) 模块 .....</b>	<b>88</b>
8.1 ADC 框图 .....	89
8.2 转换启动方式 .....	89
8.3 ADC 寄存器定义 .....	90
<b>9 配置寄存器 .....</b>	<b>92</b>
9.1 CONFIG OPTION .....	93
<b>10 指令表 .....</b>	<b>94</b>
<b>11 电气特性 .....</b>	<b>96</b>
11.1 绝对极限参数 .....	96
11.2 直流电器特性 .....	99
11.3 交流电气特性 .....	100
<b>12 开发支持 .....</b>	<b>101</b>
12.1 仿真信息 .....	101
12.2 烧录信息 .....	102
<b>13 封装信息 .....</b>	<b>104</b>
13.1 SOP 8 PIN .....	104
13.2 DFN 8 PIN .....	106
13.3 MSOP 10 PIN .....	107
13.4 SOP 14 PIN .....	108
13.5 TSSOP 14 PIN .....	109
13.6 SOP16 PIN .....	110
<b>14 汇春知识产权政策 .....</b>	<b>111</b>
14.1 专利权 .....	111
14.2 著作权 .....	111

## 1.1 系统结构图

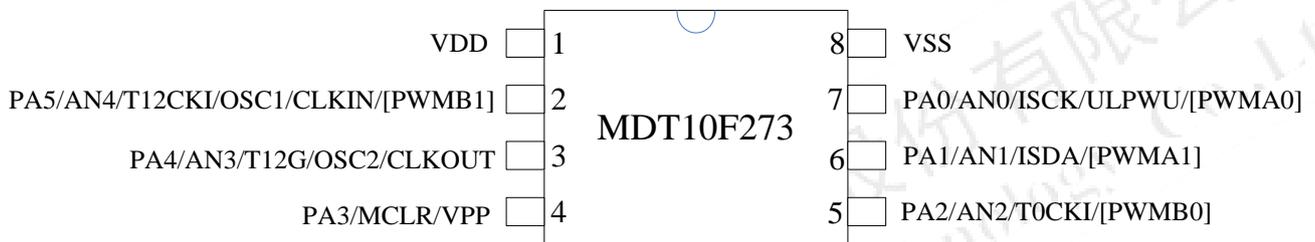


## 1.2 封装脚位图

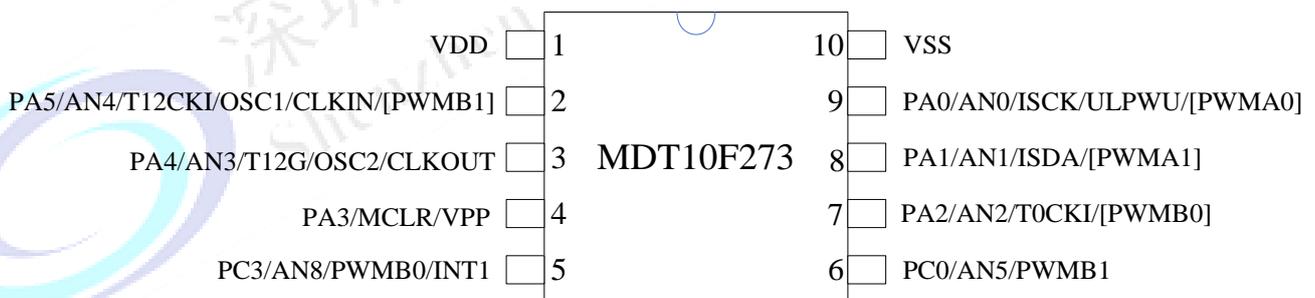
### 1.2.1 6 引脚图 (SOT23-6)



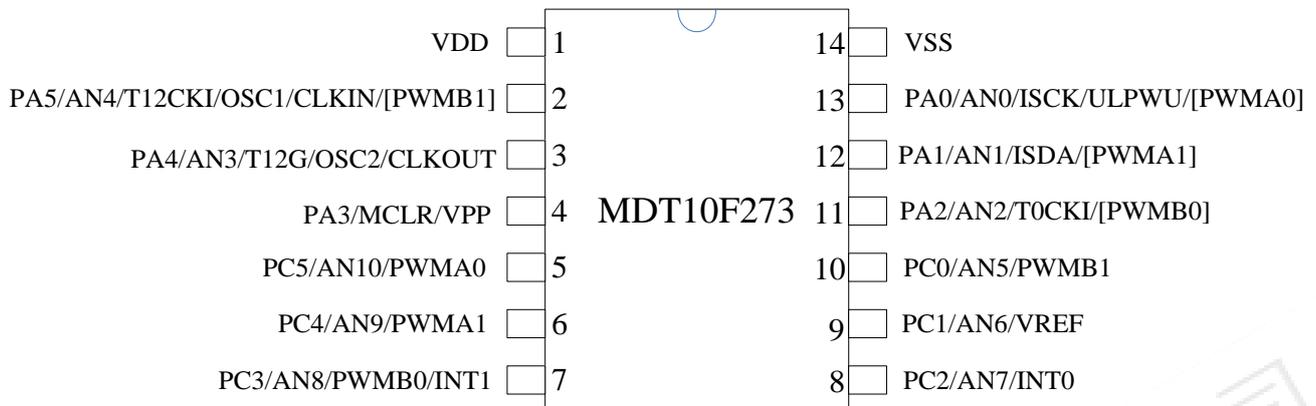
### 1.2.2 8 引脚图 (DFN、SOP)



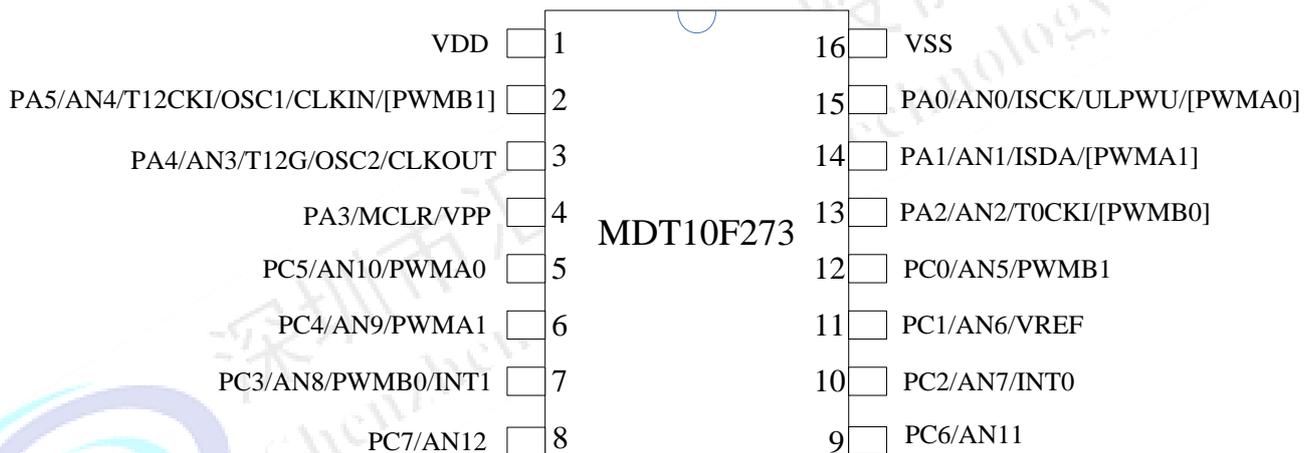
### 1.2.3 10 引脚图 (MSOP)



### 1.2.4 14 引脚图 (TSSOP、SOP)



### 1.2.5 16 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/ISCK/ULPWU/[PWMA0]	PA0	ST	CMOS	具有可编程上拉/下拉和电平变化中断的PORTA I/O
	AN0	AN	—	A/D通道0输入
	ISCK	ST	—	串行编程和调试时钟输入
	ULPWU	AN	—	超低功耗唤醒输入
	[PWMA0]	—	CMOS	功能转移之后PWMA0信号输出
PA1/AN1/ISDA/[PWMA1]	PA1	ST	CMOS	具有可编程上拉/下拉和电平变化中断的PORTA I/O
	AN1	AN	—	A/D通道1输入
	[PWMA1]	—	CMOS	功能转移之后PWMA1信号输出
	ISDA	ST	CMOS	串行编程和调试数据输入输出
PA2/AN2/T0CKI/[PWMB0]	PA2	ST	CMOS	具有可编程上拉/下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
	[PWMB0]	—	CMOS	功能转移之后PWMB0信号输出
PA3/MCLR/VPP	PA3	ST	—	做输出时, 仅能开漏输出、具有可编程上拉和带电平变化中断的PORTA I/O
	MCLR	ST	—	带有内部上拉的主复位
	VPP	HV	—	编程电压(8.9V)
PA4/AN3/T12G/OSC2/CLKOUT	PA4	ST	CMOS	具有可编程上拉/下拉和电平变化中断的PORTA I/O
	AN3	AN	—	A/D通道3输入
	T12G	ST	—	Timer1/2门控(计数使能)
	OSC2	—	XTAL	晶振/谐振器
	CLKOUT	—	CMOS	Fosc/4输出
PA5/AN4/T12CKI/OSC1/CLKIN/[PWMB1]	PA5	ST	CMOS	具有可编程上拉/下拉和电平变化中断的PORTA I/O
	T12CKI	ST	—	Timer1/2外部时钟输入
	AN4	AN	—	A/D通道4输入
	[PWMB1]	—	CMOS	功能转移之后PWMB1信号输出
	OSC1	XTAL	—	晶振/谐振器
	CLKIN	ST	—	外部时钟输入/RC振荡器连接
PC0/AN5/PWMB1	PC0	ST	CMOS	具有可编程上拉/下拉的PORTC I/O
	PWMB1	—	CMOS	功能转移之前PWMB1信号输出
	AN5	AN	—	A/D通道5输入
PC1/AN6/VREF	PC1	ST	CMOS	具有可编程上拉/下拉的PORTC I/O
	AN6	AN	—	A/D通道6输入
	VREF	AN	—	A/D外部参考电压
PC2/AN7/INT0	PC2	ST	CMOS	具有可编程上拉/下拉的PORTC I/O
	AN7	AN	—	A/D通道7输入
	INT0	ST	—	外部中断
PC3/AN8/PWMB0/INT1	PC3	ST	CMOS	P具有可编程上拉/下拉的PORTC I/O
	AN8	AN	—	A/D通道8输入
	PWMB0	—	CMOS	功能转移之前PWMB0信号输出
	INT1	ST	—	外部中断
PC4/AN9/PWMA1	PC4	ST	CMOS	具有可编程上拉/下拉的PORTC I/O
	AN9	AN	—	A/D通道9输入
	PWMA1	—	CMOS	功能转移之前PWMA1信号输出
PC5/AN10/PWMA0	PC5	ST	CMOS	具有可编程上拉/下拉的PORTC I/O
	AN10	AN	—	A/D通道10输入
	PWMA0	—	CMOS	功能转移之前PWMA0信号输出
PC6/AN11	PC6	ST	CMOS	具有可编程上拉/下拉的PORTC I/O
	AN11	AN	—	A/D通道11输入
PC7/AN12	PC7	ST	CMOS	具有可编程上拉/下拉的PORTC I/O
	AN12	AN	—	A/D通道12输入
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

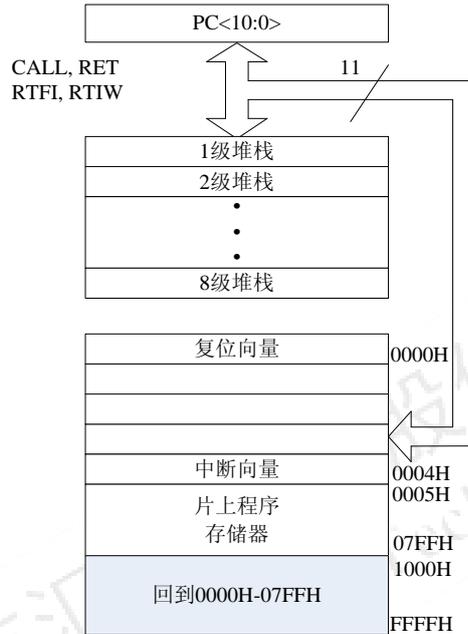
XTAL = 晶振

深圳市汇春科技股份有限公司  
Shenzhen Yspring Technology Co., Ltd.

## 2 存储器构成

### 2.1 程序存储器构成

MDT10F273 具有一个 11 位程序计数器，可以对  $2K \times 14$  的程序存储空间进行寻址。而 MDT10F273 仅在物理上实现了第一个  $2K \times 14$  (0000H-07FFH) 的存储空间。访问该边界以外的单元将导致实际访问存储器的第一个  $2K \times 14$  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；
- 欠压复位；

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 和 Bank1 中的 A0H-BFH 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 → 选定 Bank0

01 → 选定 Bank1

10 → 选定 Bank2

11 → 选定 Bank3

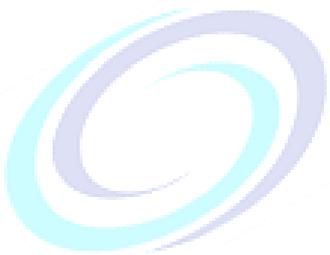
## 2.2.1 通用文件寄存器

在 MDT10F273 中通用寄存器是按 128x8 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, MSR) 间接访问每个存储器 (见第 2.4 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。



深圳市汇春科技股份有限公司  
Shenzhen Yspring Technology Co., Ltd.

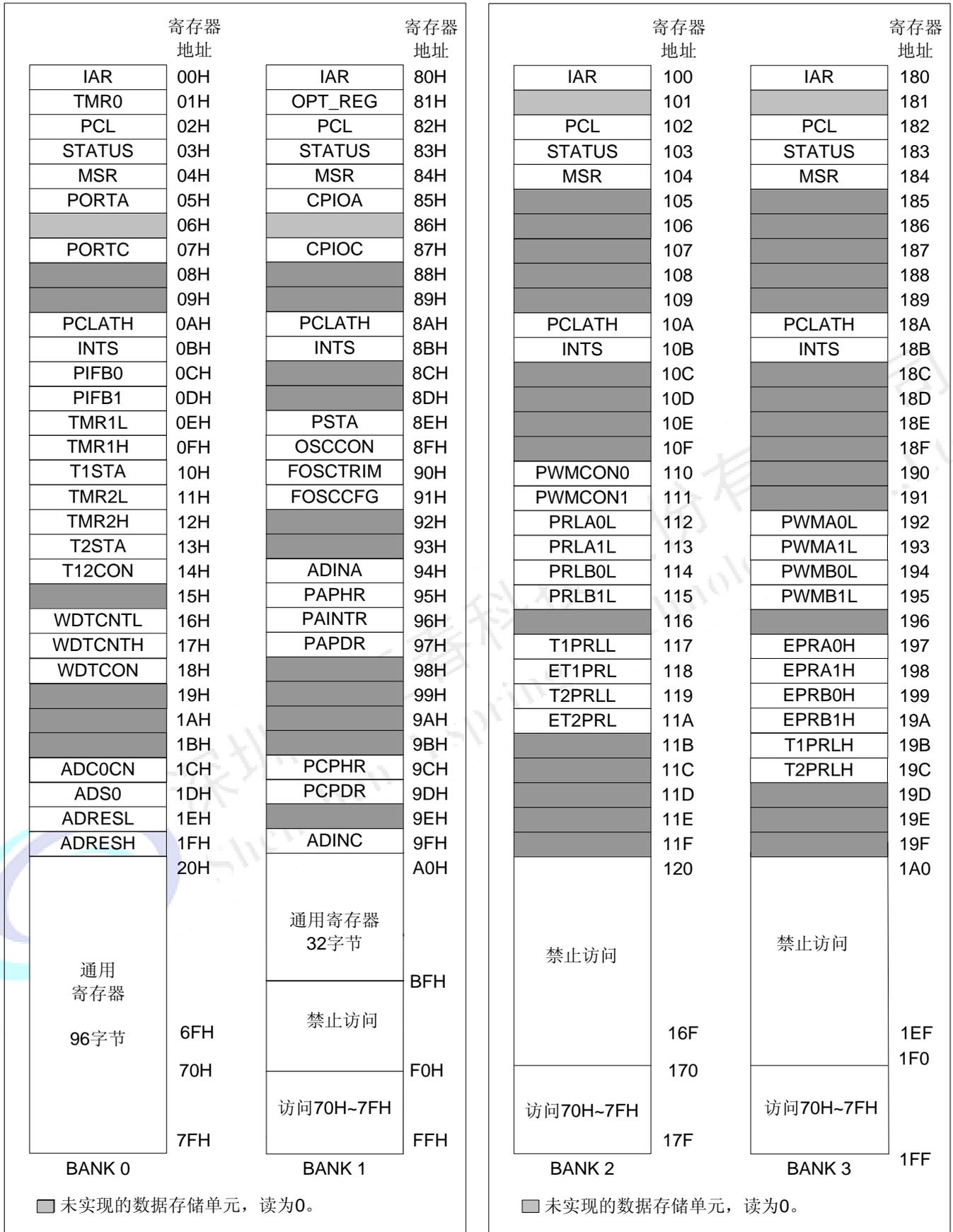


图 2-2: 数据存储映射

## 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	默认值
BANK0										
00H	<a href="#">IAR</a>	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="#">TMR0</a>	TIMER0 模块寄存器								xxxx xxxx
02H	<a href="#">PCL</a>	程序计数器（PC）低字节								0000_0000
03H	<a href="#">STATUS</a>	IRP	RP1	RP0	TF	PF	Z	HC	C	0001_1xxx
04H	<a href="#">MSR</a>	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	--x0_x000
06H	—	未实现								—
07H	<a href="#">PORTC</a>	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	--x0_0000
08H	—	未实现								—
09H	—	未实现								—
0AH	<a href="#">PCLATH</a>	—	—	—	—	—	程序计数器高 3 位写缓冲器			----_000
0BH	<a href="#">INTS</a>	GIE	PEIE	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000_0000
0CH	<a href="#">PIFB0</a>	INT1E	ADIE	OSFIE	TMR1IE	INT1F	ADIF	OSFIF	TMR1IF	0000_0000
0DH	<a href="#">PIFB1</a>	INT1SEL	PWMBIE	PWMAIE	TMR2IE	INT0SEL	PWMBF	PWMAF	TMR2IF	0000_0000
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	T1CKPS		T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000_0000
11H	<a href="#">TMR2L</a>	16 位 TMR1 低字节的保持寄存器								0000_0000
12H	<a href="#">TMR2H</a>	16 位 TMR1 高字节的保持寄存器								0000_0000
13H	<a href="#">T2STA</a>	T2GINV	TMR2GE	T2CKPS		T2OSCEN	T2SYNC	TMR2CS	TMR2ON	0000_0000
14H	<a href="#">T12CON</a>	DIE_EN	DIE_NUM<2: 0>			T2SELEN	T2FOSC	T1SELEN	T1FOSC	0000_0000
15H	—	未实现								—
16H	<a href="#">WDTCNTRL</a>	WDT 计数器低字节								1111_1111
17H	<a href="#">WDTCNTH</a>	WDT 计数器高字节								1111_1111
18H	<a href="#">WDTCON</a>	PWMB0CP	PWMB1CP	—	WDTPS				SWDTEN	---0_1000
19H	—	未实现								—
1AH	—	未实现								—
1BH	—	未实现								—
1CH	<a href="#">ADC0CN</a>	—	ADFM	SC		ADC_EN	GO/GON $\bar{E}$	CM		0000_0000
1DH	<a href="#">ADS0</a>	CHNSEL				REFSEL		VREFSEL		0000_0011
1EH	<a href="#">ADRESL</a>	AD 结果寄存器低字节								0000_0000
1FH	<a href="#">ADRESH</a>	AD 结果寄存器高字节								0000_0000

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

## 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	默认值
BANK1										
80H	<a href="#">IAR</a>	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	<a href="#">OPT_REG</a>	—	TMR0EN	TCS	TCE	PSC	PS			-111_1111
82H	<a href="#">PCL</a>	程序计数器（PC）低字节								0000_0000
83H	<a href="#">STATUS</a>	IRP	RP1	RP0	TF	PF	Z	HC	C	0001_1xxx
84H	<a href="#">MSR</a>	间接数据存储器地址指针								xxxx-xxxx
85H	<a href="#">CPIOA</a>	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11_1111
86H	—	未实现								—
87H	<a href="#">CPIOC</a>	CPIOC7	CPIOC6	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	1111_1111
88H	—	未实现								—
89H	—	未实现								—
8AH	<a href="#">PCLATH</a>	—	—	—	—	—	程序计数器高 3 位写缓冲器			----_000
8BH	<a href="#">INTS</a>	GIE	PEIE	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000_0000
8CH	—	未实现								—
8DH	—	未实现								—
8EH	<a href="#">PSTA</a>	—	—	ULPWUE	SLVREN	MCR	IER	POR	LVR	--01_100x
8FH	<a href="#">OSCCON</a>	FOSCEN	IRCF			OSTS	HTS	LTS	SCS	0101_x001
90H	<a href="#">FOSCTRM</a>	快时钟软件调解位								0000_0000
91H	<a href="#">FOSCCFG</a>	快时钟出厂校正位								xxxx-xxxx
92H	—	未实现								—
93H	—	未实现								—
94H	<a href="#">ADINA</a>	—	—	AN4	AN3	—	AN2	AN1	AN0	--11_-111
95H	<a href="#">PAPHR</a>	—	—	PAPHR5	PAPHR4	PAPHR3	PAPHR2	PAPHR1	PAPHR0	--00_0000
96H	<a href="#">PAINTR</a>	—	—	PAINTR5	PAINTR4	PAINTR3	PAINTR2	PAINTR1	PAINTR0	--00_0000
97H	<a href="#">PAPDR</a>	—	—	PAPDR5	PAPDR 4	—	PAPDR 2	PAPDR 1	PAPDR 0	--00_0000
98H	—	未实现								—
99H	—	未实现								—
9AH	—	未实现								—
9BH	—	未实现								—
9CH	<a href="#">PCPHR</a>	PCPHR7	PCPHR6	PCPHR5	PCPHR4	PCPH3	PCPHR2	PCPHR1	PCPHR0	0000_0000
9DH	<a href="#">PCPDR</a>	PCPDR7	PCPDR6	PCPDR5	PCPDR4	PCPDR3	PCPDR2	PCPDR1	PCPDR0	0000_0000
9EH	—	未实现								—
9FH	<a href="#">ADINC</a>	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	1111_1111

## 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	默认值
BANK2										
100H	<a href="#">IAR</a>	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	—	未实现								—
102H	<a href="#">PCL</a>	程序计数器（PC）低字节								0000_0000
103H	<a href="#">STATUS</a>	IRP	RP1	RP0	$\overline{TF}$	$\overline{PF}$	Z	HC	C	0001_1xxx
104H	<a href="#">MSR</a>	间接数据存储器地址指针								xxxx-xxxx
105H	—	未实现								—
106H	—	未实现								—
107H	—	未实现								—
108H	—	未实现								—
109H	—	未实现								—
10AH	<a href="#">PCLATH</a>	—	—	—	—	—	程序计数器高 3 位写缓冲器			----_000
10BH	<a href="#">INTS</a>	GIE	PEIE	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000_0000
10CH	—	未实现								—
10DH	—	未实现								—
10EH	—	未实现								—
10FH	—	未实现								—
110H	<a href="#">PWMCON0</a>	PWMB1ON	PWMB0ON	PWMA1ON	PWMA0ON	PWMB1EN	PWMB0EN	PWMA1EN	PWMA0EN	0000_0000
111H	<a href="#">PWMCON1</a>	PWMB1TR	PWMB0TR	PWMA1TR	PWMA0TR	PWMB1OL	PWMB0OL	PWMA1OL	PWMA0OL	0000_0000
112H	<a href="#">PRLA0L</a>	PWMA0 占空比重载寄存器低字节								0000_0000
113H	<a href="#">PRLA1L</a>	PWMA1 占空比重载寄存器低字节								0000_0000
114H	<a href="#">PRLB0L</a>	PWMB0 占空比重载寄存器低字节								0000_0000
115H	<a href="#">PRLB1L</a>	PWMB1 占空比重载寄存器低字节								0000_0000
116H	—	未实现								—
117H	<a href="#">T1PRL</a>	PWMA0/1 周期重载寄存器的低字节								0000_0000
118H	<a href="#">ET1PRL</a>	PWMA0/1 周期寄存器的低字节(只读)								0000_0000
119H	<a href="#">T2PRL</a>	PWMB0/1 周期重载寄存器的低字节								0000_0000
11AH	<a href="#">ET2PRL</a>	PWMB0/1 周期寄存器的低字节(只读)								0000_0000
11BH	—	未实现								—
11CH	—	未实现								—
11DH	—	未实现								—
11EH	—	未实现								—
11FH	—	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	默认值
BANK3										
180H	<a href="#">IAR</a>	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	—	未实现								—
182H	<a href="#">PCL</a>	程序计数器（PC）低字节								0000_0000
183H	<a href="#">STATUS</a>	IRP	RP1	RP0	TF	PF	Z	HC	C	0001_1xxx
184H	<a href="#">MSR</a>	间接数据存储器地址指针								xxxx-xxxx
185H	—	未实现								—
186H	—	未实现								—
187H	—	未实现								—
188H	—	未实现								—
189H	—	未实现								—
18AH	<a href="#">PCLATH</a>	—	—	—	—	—	程序计数器高 3 位的缓冲器			----_000
18BH	<a href="#">INTS</a>	GIE	PEIE	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000_0000
18CH	—	未实现								—
18DH	—	未实现								—
18EH	—	未实现								—
18FH	—	未实现								—
190H	—	未实现								—
191H	—	未实现								—
192H	<a href="#">PWMA0L</a>	PWMA0 占空比较寄存器低字节								0000_0000
193H	<a href="#">PWMA1L</a>	PWMA1 占空比较寄存器低字节								0000_0000
194H	<a href="#">PWMB0L</a>	PWMB0 占空比较寄存器低字节								0000_0000
195H	<a href="#">PWMB1L</a>	PWMB1 占空比较寄存器低字节								0000_0000
196H	—	未实现								—
197H	<a href="#">EPRA0H</a>	PWMA0H高4位				PRLA0H 高 4 位				0000_0000
198H	<a href="#">EPRA1H</a>	PWMA1H高4位				PRLA1H 高 4 位				0000_0000
199H	<a href="#">EPRB0H</a>	PWMB0H高4位				PRLB0H 高 4 位				0000_0000
19AH	<a href="#">EPRB1H</a>	PWMB1H高4位				PRLB1H 高 4 位				0000_0000
19BH	<a href="#">T1PRLH</a>	ET1PRH 高 4 位（只读）				T1PRLH 高 4 位				0000_0000
19CH	<a href="#">T2PRLH</a>	ET2PRH 高 4 位（只读）				T2PRLH 高 4 位				0000_0000
19DH	—	未实现								—
19EH	—	未实现								—
19FH	—	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（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 TF 和 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 10.0 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
IRP	RP1	RP0	TF	PF	Z	HC	C
bit7							bit0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit7

**IRP**：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

**RP1RP0**：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

**TF**：超时状态位

1 = 上电后，执行了 CLRWT 指令或 SLEEP 指令

0 = 发生 WDT 超时溢出

bit3

**PF**：掉电标志位

1 = 上电复位后或执行了 CLRWT 指令

0 = 执行了 SLEEP 指令

bit2

**Z**：零标志位

1 = 算术运算或逻辑运算的结果为零

0 = 算术运算或逻辑运算的结果不为零

bit1

**HC**：半进位/借位位（ADDWR、ADDWI、SUBWR 和 SUBWI 指令）。对于借位，极性是相反的。

1 = 结果的第 4 低位向高位发生了进位

0 = 结果的第 4 低位未向高位发生进位

bit0 **C:** 进位/借位<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 OPT\_REG 选项寄存器

选项 (OPT\_REG) 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
- 外部中断
- Timer0

注: 要为Timer0指定1:1的预分频比, 应将OPT\_REG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 (OPT\_REG)

U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	<b>TMR0EN</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 未实现: 读为0

bit6 **TMR0EN:** TMR0使能位

1 = 使能Timer0

0 = 停止Timer0

bit5 **TCS:** TMR0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:** TMR0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	TMR0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（LVR）
- 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LVR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-1	R/W-0	R/W-0	R/W-x
—	—	ULPWUE	SLVREN	MCLR	IER	POR	LVR
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 **SLVREN**：软件欠压检测使能位  
1 = 使能欠压检测  
0 = 禁止欠压检测

bit3 **MCLR**：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 **LVR**：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）

## 2.3 PCL 和 PCLATH

程序计数器 (PC) 为 11 位宽。其低 8 位来自可读写的 PCL 寄存器，高 3 位 (PC<10:8>) 来自 PCLATH，不能直接读写。只要发生复位，PC 就将被清零。图 2-3 显示了装载 PC 值的两种情形。图 2-3 上方的例子说明在写 PCL (PCLATH<2:0>→PCH) 时是如何装载 PC 的。

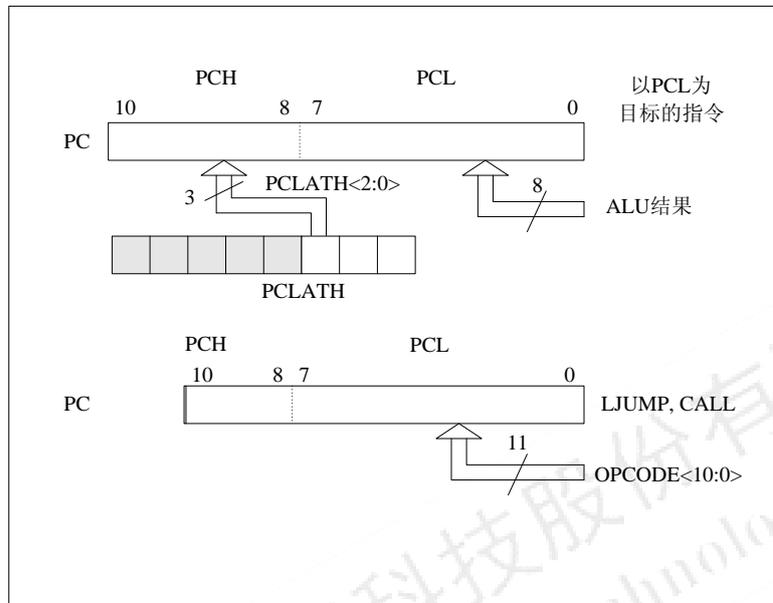


图 2-3: 在不同情况下装载 PC

### 2.3.1 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<10:8>位 (PCH) 被 PCLATH 寄存器的内容所取代。这样可通过将所需的高 3 位写入 PCLATH 寄存器来改变程序计数器的所有内容。当低 8 位写入 PCL 寄存器时，程序计数器的所有 11 位将变为 PCLATH 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 PCLATH 设置为表的起始地址，如果表长度大于 255 条指令，或如果存储器地址的低 8 位在表的中间从 0xFF 计满返回到 0x00，那么在每次表起始地址与表内的目标地址之间发生计满返回时，PCLATH 均必须递增。

### 2.3.2 堆栈

MDT10F273 器件具有 8 级深 x11 位宽的硬件堆栈 (见图 2-2 和图 2-3)。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。PCLATH 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、RTWI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							
<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>**：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（PCLATH）

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	<b>PC10</b>	<b>PC9</b>	<b>PC8</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

bit7-3                未实现：读为0

Bit2-0                **PC<10:8>**：程序计数器高3位。

## 2.4 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。

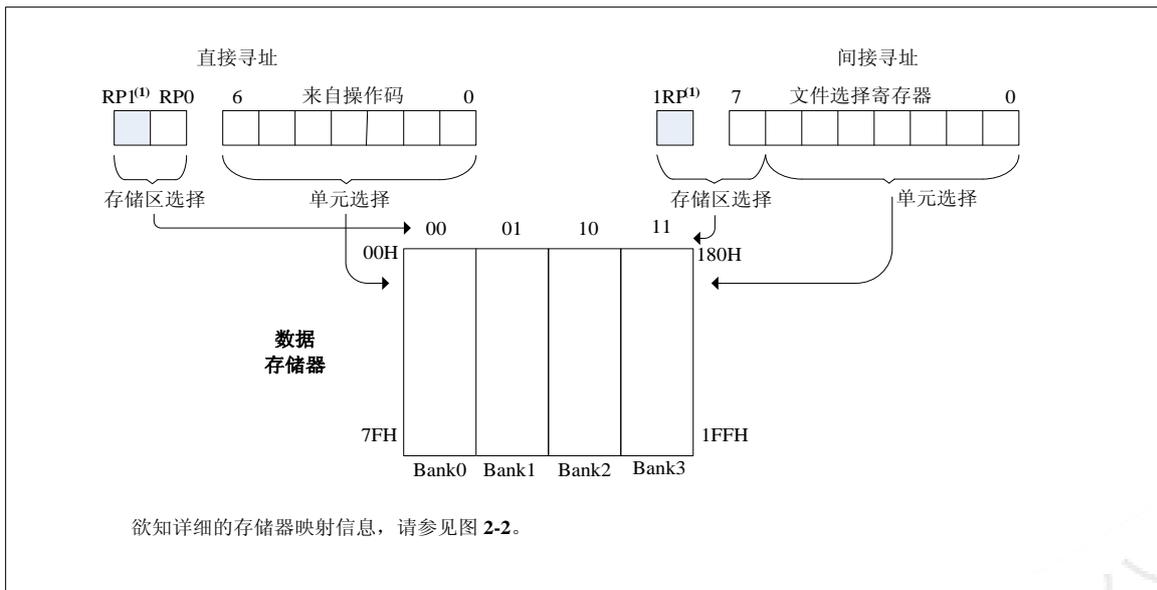


图 2-5: MDT10F273 的直接/间接寻址

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

```

## 2.5 看门狗定时器 (WDT)

WDT 具有以下特性：

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

WDT 在如表 2-6-1 所述的情况下清零。

### 2.5.1 WDT 振荡器

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

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

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

## 2.5.2 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。

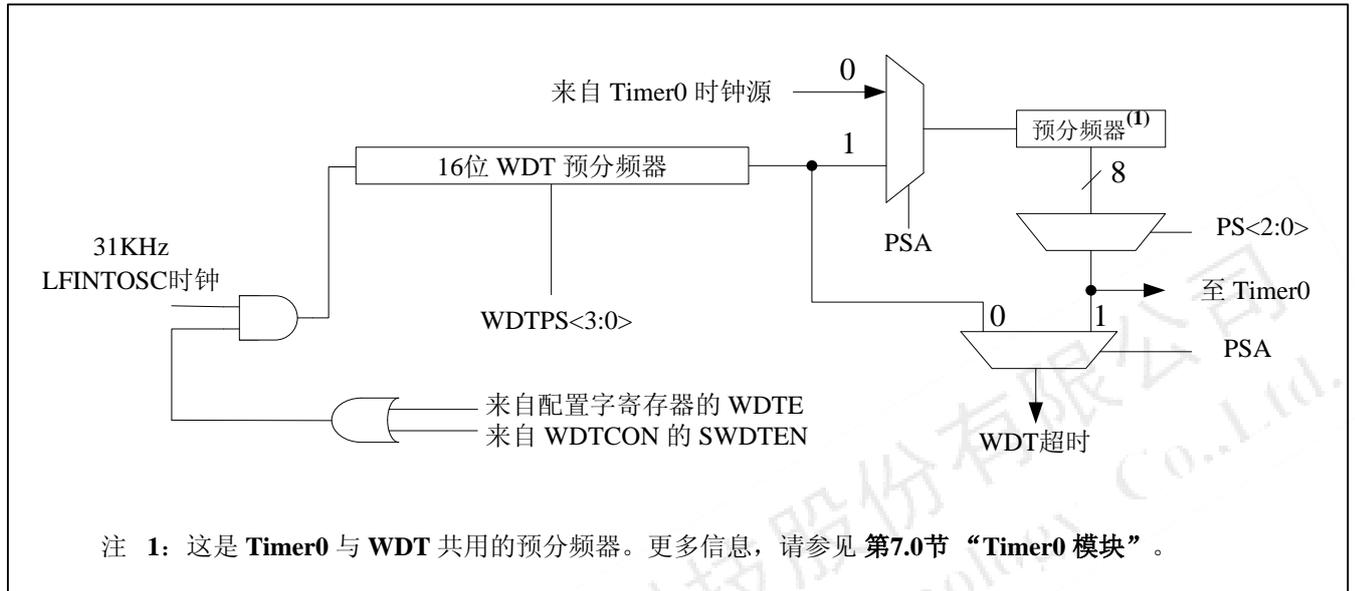


图 2-6: 看门狗定时器框图

寄存器 16H: 16 位看门狗定时器低字节寄存器 (WDTCNTRL)

R-1	R-1	R-1	R-1	R-1	R-1	R-1	R-1
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 16 位看门狗计数器低 8 位: 只读

寄存器 17H: 16 位看门狗定时器高字节寄存器 (WDTCNTH)

R-1	R-1	R-1	R-1	R-1	R-1	R-1	R-1
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 16 位看门狗计数器高 8 位: 只读

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
PWMB0CP	PWMB1CP	—	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 **PWMB0CP** : PWMA0复制到PWMB0输出使能位,“1”有效。  
将PWMA0输出结果复制到PWMB0输出。
- Bit6 **PWMB1CP** : PWMA1复制到PWMB1输出使能位,“1”有效。  
将PWMA1输出结果复制到PWMB1输出
- Bit5 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 LVR 时的值	所有其他复位值
WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0_0000	---0_1000
WDCNTL	WDT 计数器低字节								1111_1111	1111_1111
WDCNTH	WDT 计数器高字节								1111_1111	1111_1111
OPT_REG		TMR0EN	TCS	TCE	PSC	PS2	PS1	PS0	1111_1111	-111_1111
CONFIG0	—	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 800H](#)。

## 3 复位

### 3.1 概述

MDT10F273 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 MCLR 复位
- 欠压复位 (LVR)

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

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

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

图 3-1 给出了片上复位电路的简化框图。MCLR 复位路径上有一个噪声滤波器，用来检测并滤除小脉冲。关于脉冲宽度规范，请参见第 11.0 节“电气特性”。

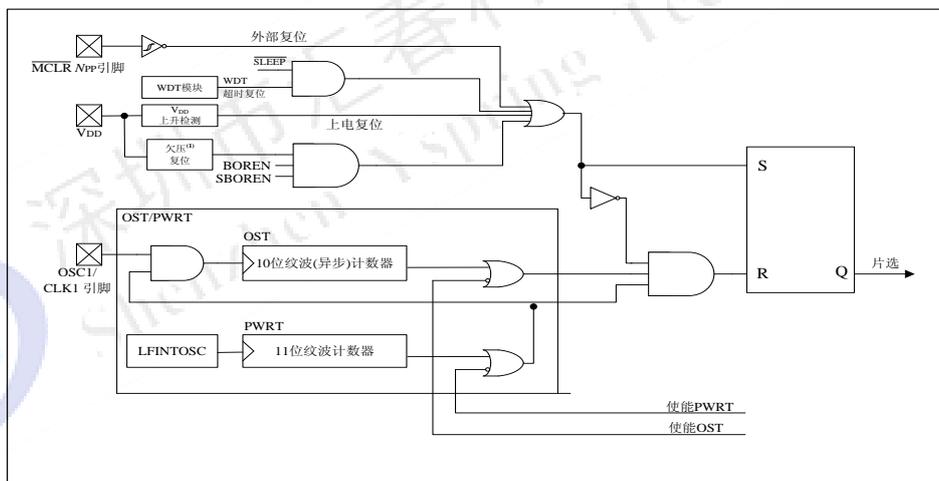


图 3-1: 片上复位电路的简化框图

表 3-1: STATUS/PSTA 位及其含义

POR	LVR	TF	PF	条件
0	x	1	1	上电复位
u	0	1	1	欠压复位
u	u	0	u	WDT 复位
u	u	0	0	WDT 唤醒
u	u	u	u	正常工作期间的 MCLR 复位
U	u	1	0	休眠期间的 MCLR 复位

图注: u = 不变, x = 未知

## 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 11.0 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 VLVR（见第 3.5 节“欠压复位 (LVR)”）。

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

MDT10F273 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，可使能内部  $\overline{\text{MCLR}}$  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。

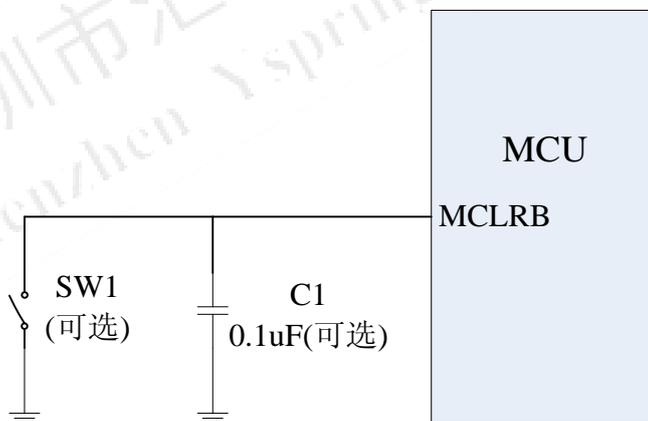


图 3-3: 建议的  $\overline{\text{MCLR}}$  电路

## 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 31kHz。更多信息，请参见第 4.5 节“内部时钟模式”。

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

置位  $\overline{\text{PWRTE}}$  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 11.0 节“电气特性”）。

注：在 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 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 LVREN0 和 LVREN1 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 LVR 的使能进行控制的模式。当 LVREN<1:0> = 01 时，可由 PSTA 寄存器的 SLVREN 位使能/禁止 LVR，从而能用软件对其进行控制。通过选择 LVREN<1:0>，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，SLVREN 位被禁止。关于配置字的定义，请参见寄存器 PSTA。

如果 VDD 下降到 VLVR 以下，且持续时间超过参数值 (TLVR)（见第 11.0 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 VLVR 的时间少于参数值 (TLVR)，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 VLVR 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 PWRTE 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 VLVR 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 VLVR 以上，上电延时定时器将执行一段 64ms 的复位。

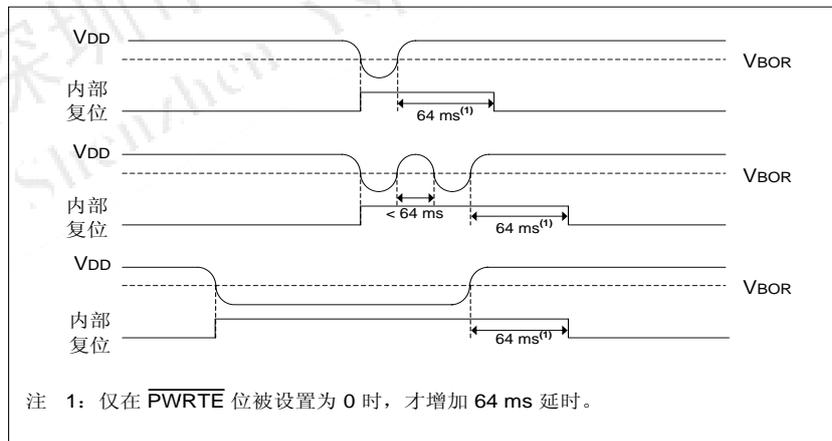


图 3-5: 欠压情形

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 LVR 时的值	所有其他复位值
PSTA	—	—	ULPWUE	SLVREN	MCR	IER	POR	LVR	0001_100x	0001_100x
STATUS	IRP	PR1	PAGE	TF	PF	Z	HC	C	0001_1xxx	0001_1xxx

图注： u = 不变  
q = 取值视具体情况而定

x = 未知  
LVR 不使用阴影单元。

— = 未实现位，读为 0

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PWRTE位的状态。例如，在EC模式且PWRTE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MDT10F273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。

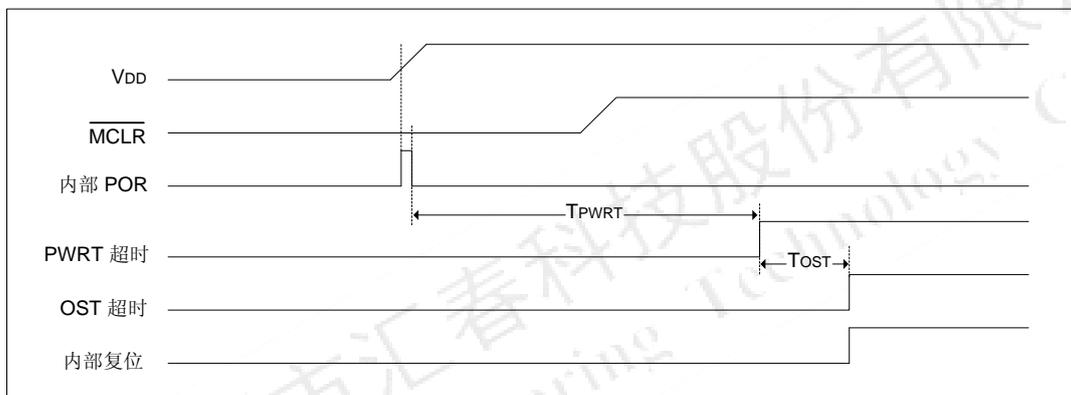


图 3-6-1: 上电时的超时时序（MCLR延时）：情形 1

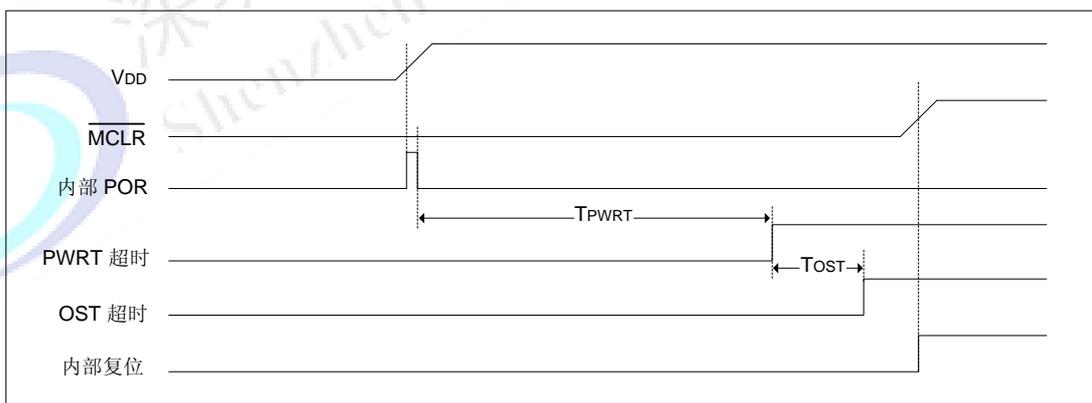


图 3-6-2: 上电时的超时时序（MCLR延时）：情形 2

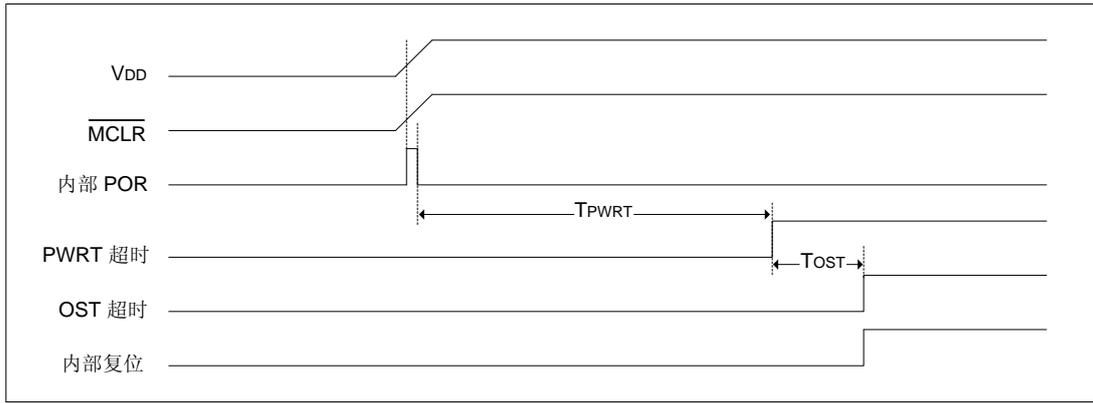


图 3-6-3: 上电时的超时时序 (MCLR 及 VDD)

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MCLR复位	0000H	000u_uuuu	--0u_--uu
休眠期间的MCLR复位	0000H	0001_0uuu	--0u_--uu
WDT 复位	0000H	0000_uuuu	--0u_--uu
WDT 唤醒	PC + 1	uuu0_0uuu	--uu_--uu
欠压复位	0000H	0001_1uuu	--01_--u0
通过中断从休眠唤醒	PC + 1 <sup>(1)</sup>	uuu1_0uuu	--uu_--uu

图注: u = 不变

x = 未知

— = 未实现位, 读为 0

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

表 3-6-2: 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
	PWRTE = 0	PWRTE = 1	PWRTE = 0	PWRTE = 1	
XT, HS, LP	$T_{PWRT} + 1024 * T_{OSC}$	$1024 * T_{OSC}$	$T_{PWRT} + 1024 * T_{OSC}$	$1024 * T_{OSC}$	$1024 * T_{OSC}$
RC, EC, INTOSC	$T_{PWRT}$	—	$T_{PWRT}$	—	—

## 3.7 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 LVR (欠压复位) 标志位。LVR 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 LVR 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 (配置寄存器中的 LVREN<1:0> = 00) 时, LVR 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 6.3.4 节“超低功耗唤醒”和第 3.5 节“欠压复位 (LVR)”。

## 4 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。

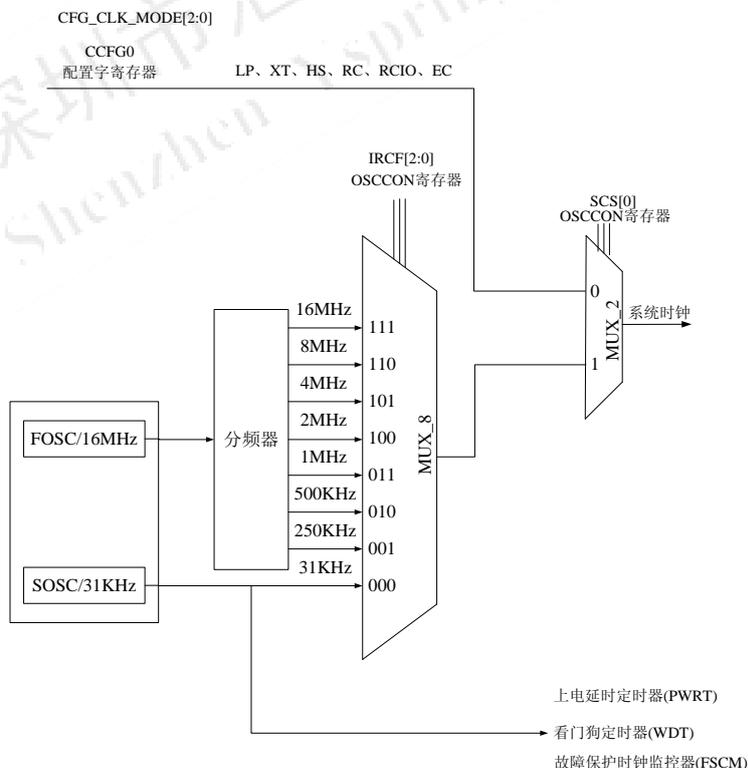


图 4-1: MCU 时钟源框图

## 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）

R/W-0	R/W-1	R/W-0	R/W-1	R-x	R-0	R-0	R/W-1
<b>FOSCEN</b>	<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	<b>FOSCEN:</b> 内部快时钟校正选择位 1 = 选择寄存器FOSCTRIM里的值校正 0 = 选择出厂校正位
bit6-4	<b>IRCF&lt;2:0&gt;:</b> 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	<b>OSTS:</b> 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	<b>HTS:</b> HFINTOSC（高频——16MHz到250kHz）状态位 1 = HFINTOSC稳定 0 = HFINTOSC不稳定
bit1	<b>LTS:</b> LFINTOSC（低频——31kHz）状态位 1 = LFINTOSC稳定 0 = LFINTOSC不稳定
bit0	<b>SCS:</b> 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 31kHz 低频内部振荡器 (LFINTOSC)。

可通过 OSCCON 寄存器的系统时钟选择 (SCS) 位，在外部或内部时钟源之间选择系统时钟。(欲了解更多信息，请参见第 4.6 节“时钟切换”)。

## 4.4 外部时钟模式

### 4.4.1 振荡器起振定时器 (OST)

如果振荡器模块配置为 LP、XT 或 HS 模式，振荡器起振定时器 (OST) 对来自 OSC1 的振荡计数 1024 次。这发生在上电复位 (POR) 之后以及上电延时定时器 (PWRT) 延时结束 (如果配置了) 时，或从休眠中唤醒后。在此期间，程序计数器不递增，程序执行暂停。OST 确保使用石英晶体谐振器或陶瓷谐振器的振荡器电路已经启动并向振荡器模块提供稳定的系统时钟信号。当在时钟源之间切换时，需要一定的延时以使新时钟稳定。表 4-4-1 给出了振荡器延时的例子。

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

表 4-4-1: 振荡器延时示例

切换自	切换到	频率	振荡器延时
休眠/POR	LFINTOSC HFINTOSC	31kHz 250kHz到16MHz	振荡器预热延时( $T_{WARM}$ )
休眠/POR	EC, RC	DC—20MHz	双周期
LFINTOSC (31kHz)	EC, RC	DC—20MHz	每次一周期
休眠/POR	LP, XT, HS	32kHz到20MHz	1024个时钟周期 (OST)
LFINTOSC (31kHz)	HFINTOSC	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 MDT 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。

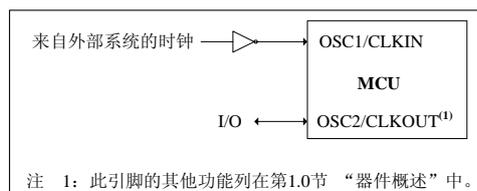


图 4-4-2: 外部时钟 (EC) 模式的工作原理

### 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。

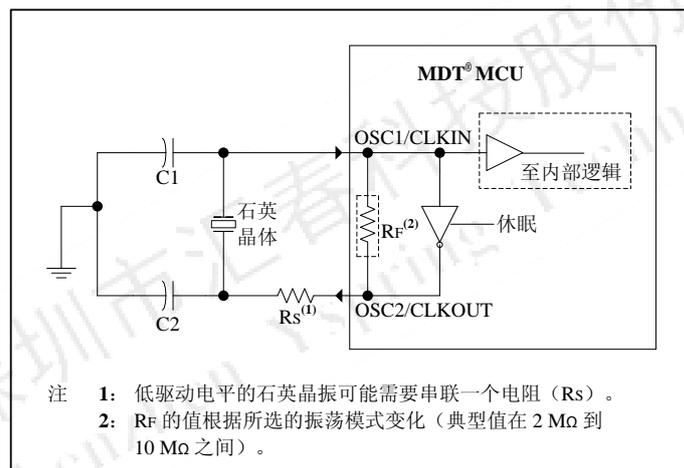


图 4-4-3-1: 石英晶体的工作原理 (LP、XT 或 HS 模式)

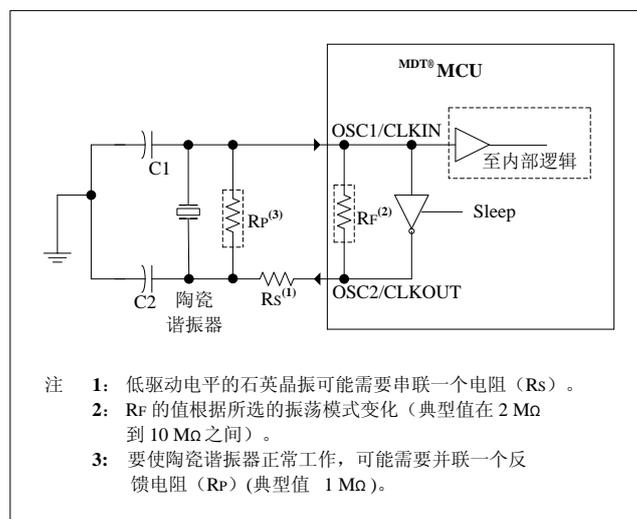


图 4-4-3-2: 陶瓷谐振器的工作原理 (XT 或 HS 模式)

#### 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。

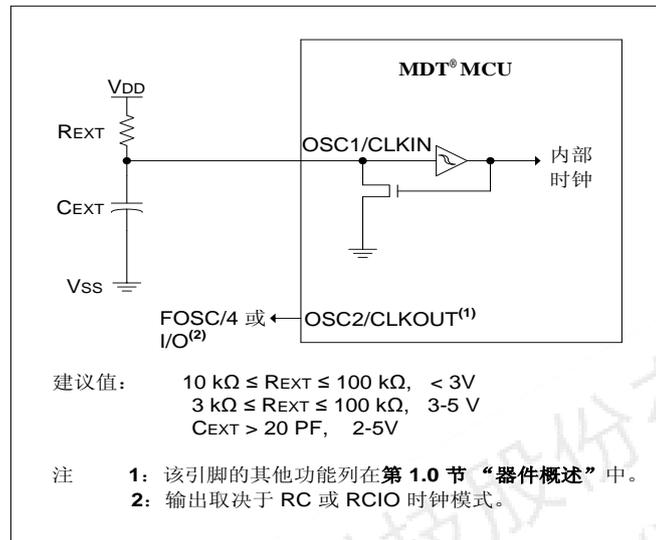


图 4-4-4: 外部 RC 模式

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  $\pm 1\%$ 。

2. LFINTOSC (低频内部振荡器) 未经校准, 工作频率为 31kHz。

通过软件对 OSCCON 寄存器的内部振荡器频率选择位  $IRCF\langle 2:0 \rangle$  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  $\neq 000$  选择 16MHz 到 250kHz 之间的任一频率，可启用 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 31kHz 内部时钟源。

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

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

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

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

## 4.5.4 频率选择位（IRCF）

16MHz HFINTOSC 和 31kHz LFINTOSC 的输出连接到后分频器和多路复用器（见图 4-1）。OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0>用于选择内部振荡器的频率输出。可通过软件选择以下 8 个频率之一：

- 16MHz
- 8MHz
- 4MHz （复位后的缺省值）
- 2MHz
- 1MHz
- 500kHz
- 250kHz
- 31kHz

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 11.0 节“电气特性”中与振荡器相关的表格中。

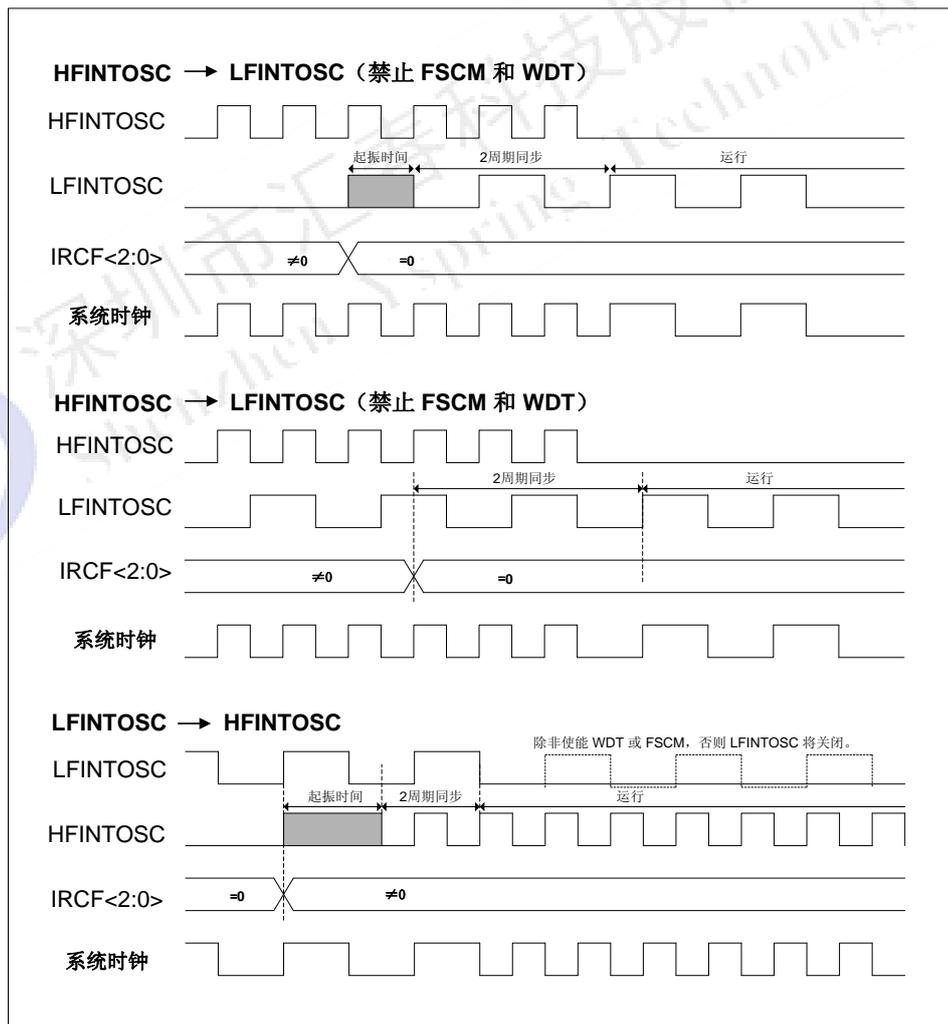


图 4-5-5: 内部振荡器切换时序

## 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 总是被置 1。

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 OST 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 OST 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 OST 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. OST<sub>S</sub> 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 OST<sub>S</sub> 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。

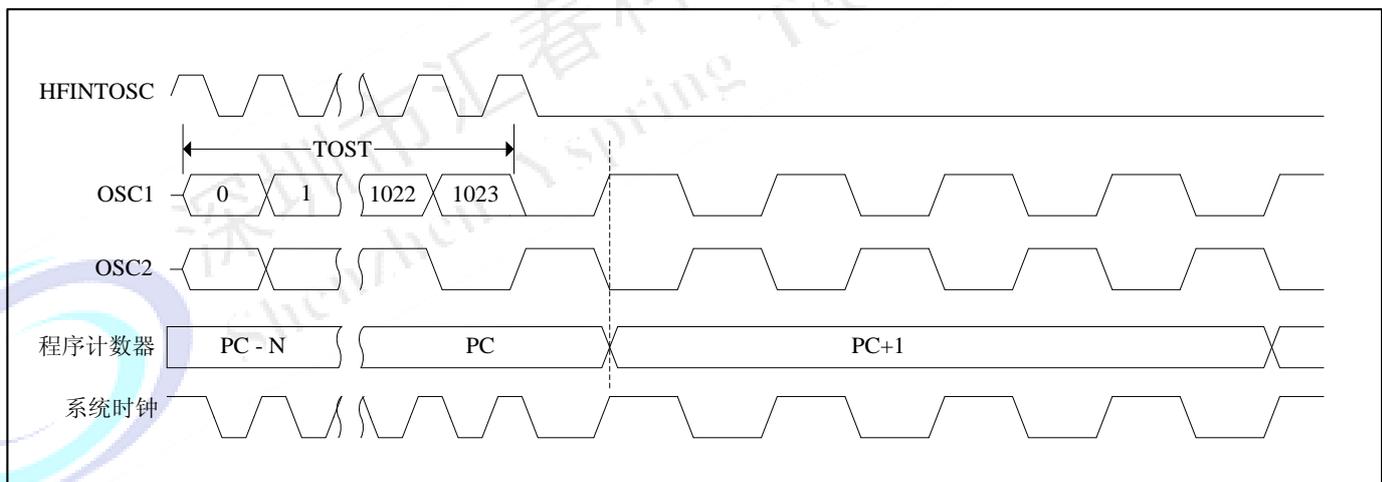


图 4-7-3: 双速启动

## 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。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： 1. 由于 WDT 超时产生的复位不会将  $\overline{\text{MCLR}}$  引脚驱动为低电平。

2. 睡眠前把 ADC 参考电压设为内部参考，否则会漏电。

## 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. PORTA 电平变化中断
3. 来自 INT0 与 INT1 引脚的外部中断
4. 看门狗定时器唤醒（如果 WDT 使能）

$\overline{\text{MCLR}}$  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 使用中断唤醒

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

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

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

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

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

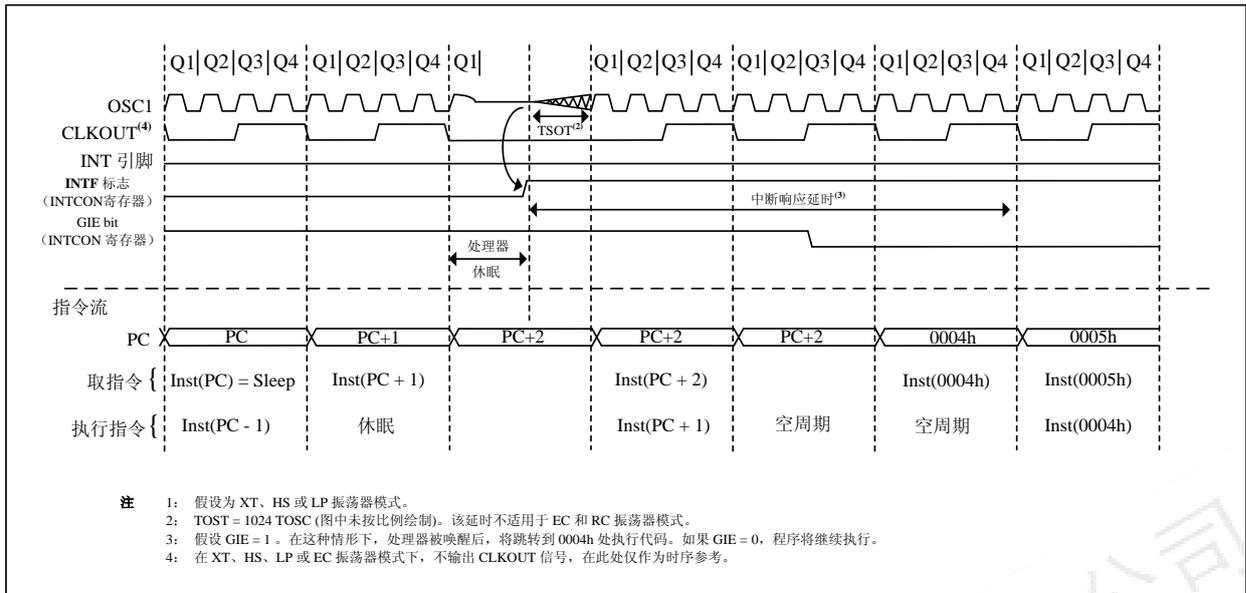


图 4-8: 通过中断将器件从休眠状态唤醒

## 4.9 故障保护时钟监控器

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

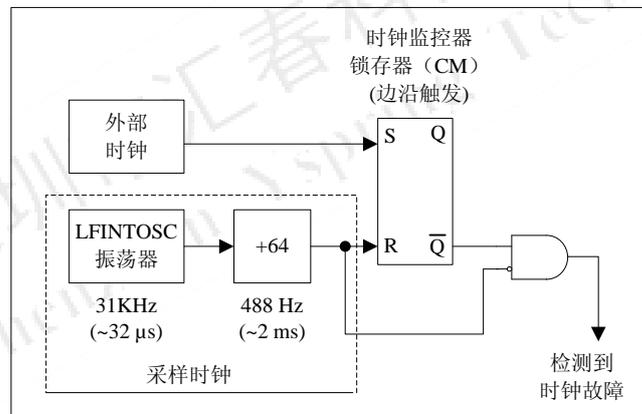


图 4-9: FSCM 框图

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

## 4.10 内部快时钟调频功能

当系统时钟为内部快时钟（IRC）且需要对频率进行调节时，可将 OSCCON 寄存器的 bit7(FOSCEN)置一来使能软件调解。

IRCTRIM 值变化对应频率变化的规律如下：

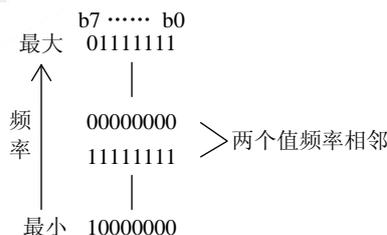


图 4-10 IRCTRIM 值变化对应频率变化规律图

- 注： 1、IRCTEN 为 0 时，对 IRC 的调节失效，此时 IRC 频率为出厂校准值；  
2、频率的变化步长并非绝对的；

寄存器90H：快时钟校正寄存器 < FOSCTRIM >

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 **IRCTRIM<7: 0>**: IRC 频率调节位

注: bit7 为方向调节控制位 (变化规律见[图10-4](#))

**寄存器91H: 内部快时钟出厂校正寄存器 < FOSCCFG >**

R-X	R-X	R-X	R-X	R-X	R-X	R-X	R-X
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 内部快时钟出厂校正寄存器

## 5 中断

MDT10F273 有以下多种中断源:

- 外部中断 INT0(PC2)与 INT1(PC3)

- Timer0 溢出中断
- Timer1 溢出中断
- Timer2 溢出中断
- PWM 中断
- PORTA 电平变化中断
- A/D 中断
- 故障保护时钟监控器中断

中断控制寄存器（INTS）和外设中断请求寄存器（PIFB0、PIFB1）在各自的标志位中记录各种中断请求。INTS 寄存器还包括各个中断允许位和全局中断允许位。

INTS 寄存器中的全局中断允许位 GIE 在置 1 时允许所有未屏蔽的中断，而在清零时禁止所有中断。INTS 寄存器中的外设中断允许位 PEIE 在置 1 时允许所有未屏蔽的外设中断，而在清零时禁止所有外设中断。可以通过 INTS 和 PIFB0、PIFB1 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：

- INT0 引脚中断
- PORTA 电平变化中断
- Timer0 溢出中断

外设中断标志位在 PIFB0、PIFB1 寄存器中。相应的中断允许位在 PIFB0、PIFB1 寄存器中，用到外设中断时，必须使能外设中断使能位 PEIE（INTS）。

PIFB0 寄存器包含以下中断标志位：

- A/D 中断
- Timer1 溢出中断
- INT1 引脚中断
- 故障保护时钟监视器中断

PIFB1 寄存器包含以下中断标志位：

- Timer1/2 的 PWM 中断
- Timer2 溢出中断

对于外部中断事件，如 INT0、INT1 引脚或 PORTA 电平变化中断，中断响应延时将为 3 到 4 个指令周期。确切的延时时间取决于发生中断事件的时间（见图 5-1-1）。对于单周期或双周期指令，中断响应延时完全相同。进入中断服务程序之后，就可以通过查询中断标志位来确定中断源。在重新允许中断前，必须用软件将中断标志位清零，以避免重复响应该中断。

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

关于 Timer1、A/D 模块的更多信息，请参见相应的外设章节。

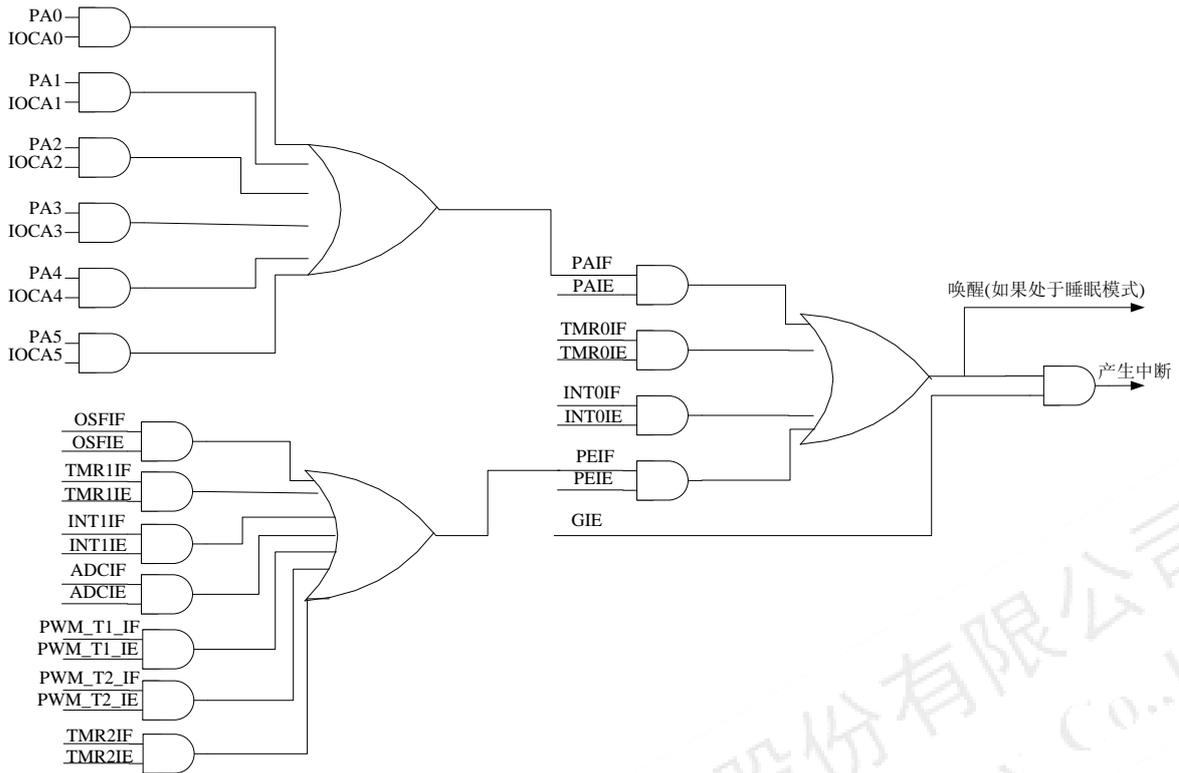


图 5-1-1: 中断逻辑

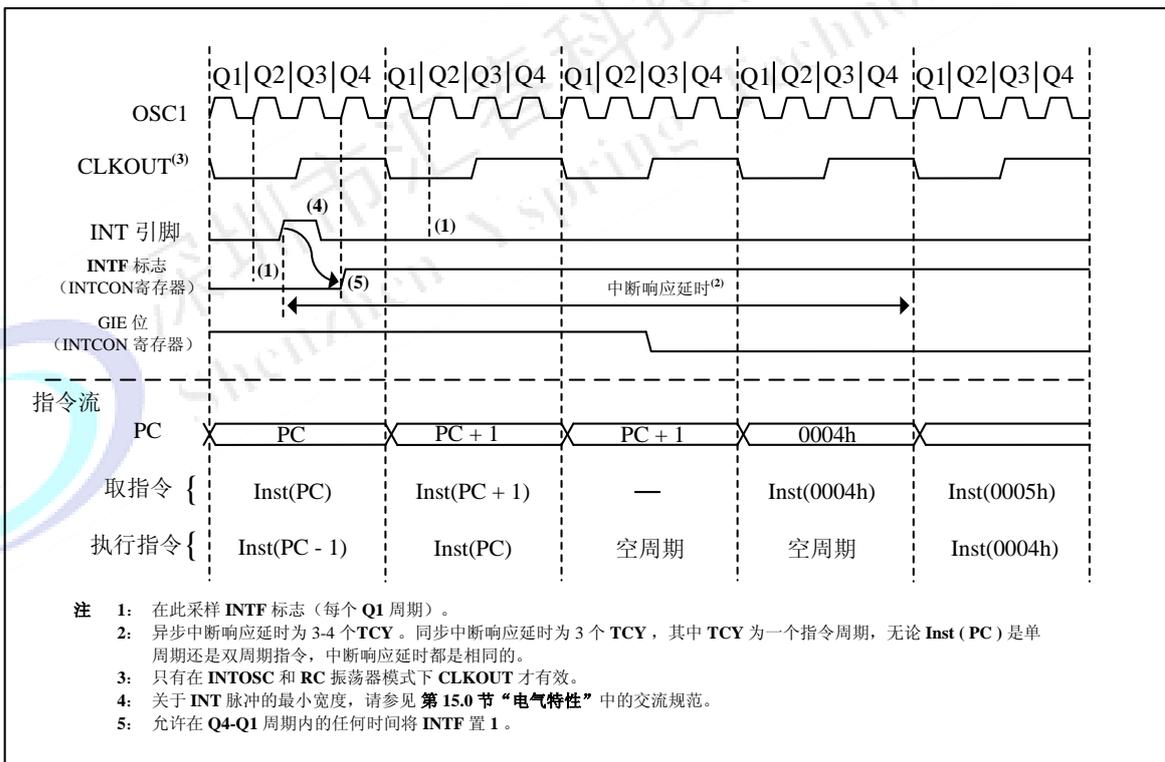


图 5-1-2: INT 引脚中断时序

## 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 TMR0 寄存器溢出、PORTA 电平变化和外部 PC2/INT0 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>INT0E</b>	<b>PAIE</b>	<b>TIF</b>	<b>INT0F</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 **INT0E**: PC2/INT0外部中断允许位  
1 = 允许PC2/INT0外部中断  
0 = 禁止PC2/INT0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 **INT0F**: PC2/INT0外部中断标志位  
1 = 发生了PC2/INT0外部中断 (必须用软件清零)  
0 = 未发生PC2/INT0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TMR0计满回零时, TIF位置1。复位时TMR0的状态不变, 它应该在清零TIF位之前被初始化。

## 5.1.2 PIFB0/PIFB1 寄存器

PIFB0 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0 (PIFB0)

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>INT1E</b>	<b>ADCIE</b>	<b>OSFIE</b>	<b>TMR1IE</b>	<b>INT1F</b>	<b>ADCIF</b>	<b>OSFIF</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 **INT1E**: PC3/INT1外部中断允许位  
1 = 允许PC3/INT1外部中断  
0 = 禁止PC3/INT1外部中断
- bit6 **ADCIE**: ADC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 **OSFIE**: 振荡器故障中断允许位  
1 = 允许振荡器故障中断  
0 = 禁止振荡器故障中断
- bit4 **TMR1IE**: Timer1溢出中断允许位  
1 = 允许Timer1溢出中断  
0 = 禁止Timer1溢出中断
- bit3 **INT1F**: PC3/INT1外部中断标志位  
1 = 发生了PC3/INT1外部中断 (必须用软件清零)  
0 = 未发生PC3/INT1外部中断
- bit2 **ADCIF**: ADC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
- bit1 **OSFIF**: 振荡器故障中断标志位  
1 = 系统振荡器发生故障, 时钟输入切换为INTOSC (必须用软件清零)  
0 = 系统时钟正常运行
- bit0 **TMR1IF**: Timer1溢出中断标志位  
1 = Timer1寄存器已经溢出 (必须用软件清零)  
0 = Timer1寄存器未溢出

## 寄存器 0DH: 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>INT1SEL</b>	<b>PWMBIE</b>	<b>PWMAIE</b>	<b>TMR2IE</b>	<b>INT0SEL</b>	<b>PWMBF</b>	<b>PWMAF</b>	<b>TMR2IF</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>INT1SEL:</b> PC3/INT1 边沿选择位 1 = 由PC3/INT1引脚的上升沿触发中断 0 = 由PC3/INT1引脚的下降沿触发中断
bit6	<b>PWMBIE:</b> Timer2的PWM中断允许位 1 = 允许Timer2的PWM中断 0 = 禁止Timer2的PWM中断
bit5	<b>PWMAIE:</b> Timer1的PWM中断允许位 1 = 允许Timer1的PWM中断 0 = 禁止Timer1的PWM中断
bit4	<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
bit3	<b>INT0SEL:</b> PC2/INT0 边沿选择位 1 = 由PC2/INT0引脚的上升沿触发中断 0 = 由PC2/INT0引脚的下降沿触发中断
bit2	<b>PWMBF:</b> Timer2的PWM中断标志位 1 = Timer2的PWM中断发生 0 = Timer2的PWM中断未发生
bit1	<b>PWMAF:</b> Timer1的PWM中断标志位 1 = Timer1的PWM中断发生 0 = Timer1的PWM中断未发生
bit0	<b>TMR2IF:</b> Timer2溢出中断标志位 1 = Timer2寄存器已经溢出 (必须用软件清零) 0 = Timer2寄存器未溢出

表 5-1: 与中断相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 LVR 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
PIFB0	INT1IE	ADCIE	OSFIE	TMR1IE	INT1IF	ADCIF	OSFIF	TMR1IF	0000_0000	0000_0000
PIFB1	INT1SEL	PWMBIE	PWMAIE	TMR2IE	INT0SEL	PWMBF	PWMAF	TMR2IF	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 外部中断

INT0(PC2)、INT1(PC3)引脚上的外部中断是边沿触发的; 当 PIFB1 寄存器的 INT1SEL 与 INT0SEL 位被置 1 时在上升沿触发, 而当 INT1SEL 与 INT0SEL 位被清零时在下降沿触发。当 INT0(PC2)、INT1(PC3)引脚上出现有效边沿时, INTS 寄存器的 INT0F 位、PIFB0 寄存器的 INT1F 位置 1。可以通过将 INTS 寄存器的 INT0E、PIFB 寄存器的 INT1E 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

INT0F、INT1F 位清零。如果 INT0E 与 INT1F 位在进入休眠状态前被置 1，则 INT0(PC2)、INT1(PC3)引脚上中断能将处理器从休眠状态唤醒。关于 INT0(PC2)、INT1(PC3)引脚上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ADINC和CPIOC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

TMR0 寄存器溢出(FFH→00H)会将 INTS 寄存器的 T0IF 位置 1。可以通过置 1/清零 INTS 寄存器的 T0IE 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 Timer1 中断

TMR1 寄存器溢出(FFFFH→0000H)会将 PIFB0 寄存器的 TMR1IF 位置 1。可以通过置 1/清零 PIFB0 寄存器的 TMR1IE 位来使能/禁止该中断。关于 Timer1 模块的操作，请参见第 7.2 节“Timer1/2 模块”。

## 5.5 Timer2 中断

TMR2 寄存器溢出(FFFFH→0000H)会将 PIFB1 寄存器的 TMR2IF 位置 1。可以通过置 1/清零 PIFB1 寄存器的 TMR2IE 位来使能/禁止该中断。关于 Timer2 模块的操作，请参见第 7.2 节“Timer1/2 模块”。

## 5.6 PWM 中断

当 TMR1/2 计数到 PWM 周期寄存器时产生 PWM 溢出，会将 PIFB1 寄存器的 PWMBF/PWMAF 位置 1。可以通过置 1/清零 PIFB1 寄存器的 PWMBIE/PWMAIE 位来使能/禁止该中断。关于 PWM 模块的操作，请参见第 7.3 节“PWM 模块”。

## 5.7 PORTA 电平变化中断

PORTA 输入电平的变化会使得 PAIF 位置 1。可以通过置 1/清零的 PAIE 位来使能/禁止该中断。此外，可通过 PAINTR 寄存器对该端口的各个引脚进行配置。

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

## 5.8 A/D 中断

ADC 模块在模数转换完成时，PIFB0 寄存器中的 ADCIF 中断标志位均置 1。可以通过置 1/清零 PIFB 寄存器中的 ADCIE 位来使能/禁止该中断。

## 5.6 中断的现场保护

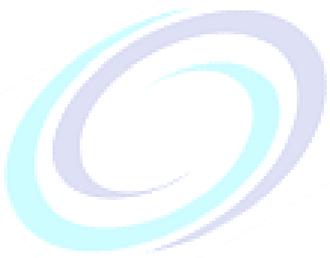
在中断期间，仅将返回的 PC 值压入堆栈。通常情况下，用户可能希望在中断期间保存关键寄存器（例如，W 寄存器和状态寄存器）。这必须用软件实现。应将临时保存寄存器 W\_TEMP 和 STATUS\_TEMP 置于 GPR 的末 16 字节中（见图 2-2）。这 16 个单元是所有存储区共用的，无需分区。这样就简化了现场保护和恢复操作。例 12-1 中所示的代码可用于：

- 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

例 5-6: 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	



深圳市汇春科技股份有限公司  
Shenzhen Yspring Technology Co., Ltd.

## 6 I/O 端口

MDT10F273 共有 14 个通用 I/O 引脚。根据使能的外设不同，有些（或全部）引脚不能用作通用 I/O。通常使能了一个外设后，相关的引脚就不能用作通用 I/O 引脚了。

### 6.1 PORTA 和 CPIOA 寄存器

PORTA 是 6 位宽的双向端口。PORTA 对应的数据方向寄存器是 CPIOA（[寄存器 85H](#)）。将 CPIOA 位置 1（= 1）可以使对应的 PORTA 引脚作为输入引脚（即禁止相应的输出驱动器）。将 CPIOA 位清零（= 0）将使对应的 PORTA 引脚作为输出引脚（即使能输出驱动器并将输出锁存器的内容置于所选的引脚上）。[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 ADINA 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
BSR    STATUS,PAGE    ;Bank1
CLRR   ADINA          ;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 输入、输出控制寄存器 (CPIOA)

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	CPIOA5	CPIOA4	CPIOA3	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输入、输出控制寄存器控制位

1 = PORTA引脚配置为输入

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。

2: PA3做输出时, 仅能开漏输出。

## 6.2 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 PWM 输出。要了解各个功能的特定信息 (如 PWM 或 ADC), 请参见本数据手册的相应章节。

注: 必须对ADINC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
BSR    STATUS,PAGE    ;Bank 1
CLRR   ADINC          ;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      |
|------------|------------|------------|------------|------------|------------|------------|------------|
| <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

**注: PC4/PC5 为大电流驱动 PAD**

寄存器 87H: PORTC 输入、输出控制寄存器 (CPIOC)

| 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输入、输出控制寄存器控制位  
 1 = PORTC引脚配置为输入  
 0 = PORTC引脚配置为输出

## 6.3 其它引脚功能

MDT10F273 的 PORTA 端口上的每一个引脚都具有电平变化中断和弱上拉/下拉功能(PA3 做为复位脚时, 使能上拉。作为 IO 口时, 没有下拉电阻)。PA0 具有超低功耗唤醒功能。下面三个小节将介绍这些功能。

### 6.3.1 ADINA 和 ADINC 寄存器

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

ADINA 和 ADINC 位的状态对数字输出功能没有影响。CPIO 清零且 ADINA 和 ADINC 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94H: 模拟选择寄存器 (ADINA)

U-0	U-0	R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1
—	—	<b>AN4</b>	<b>AN3</b>	—	<b>AN2</b>	<b>AN1</b>	<b>AN0</b>
		<b>PA5</b>	<b>PA4</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<5:4>**: 模拟选择位  
**AN<2:0>**: 模拟选择位  
 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: 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9FH: 模拟选择寄存器 (ADINC)

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>AN12</b>	<b>AN11</b>	<b>AN10</b>	<b>AN9</b>	<b>AN8</b>	<b>AN7</b>	<b>AN6</b>	<b>AN5</b>
<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 **AN<7:0>**: 模拟选择位  
 在AN<7:0>引脚上分别进行模拟或数字功能的模拟选择。  
 1 = 模拟输入。引脚被分配为模拟输入。  
 0 = 数字I/O。引脚被分配给端口或特殊功能。

## 6.3.2 弱上拉/下拉

每一个 PORTA、PORTC 引脚具有各自的可配置内部弱上拉和下拉(PA3 只有内部上拉)。控制位 PAPHRx、PCPHRx 使能或禁止每一个弱上拉, 控制位 PAPDRx、PCPDRx 使能或禁止每一个弱下拉。当将端口引脚配置为输出时, 其弱上拉电路会自动切断。在 PA3 配置为 MCLR 时, 自动启动 PA3 的弱上拉功能。MCLR 上拉不受软件控制。

### 寄存器95H: PORTA弱上拉控制寄存器 (PAPHR)

R-0	R-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	PAPHR5	PAPHR4	PAPHR3	PAPHR2	PAPHR1	PAPHR0
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            **PAPHR<5:0>**: PORTA弱上拉控制位  
 1 = 使能PORTA相应引脚上拉  
 0 = 禁止PORTA相应引脚上拉

注                1: 如果引脚处于**输出模式** (CPIOAx = 0), 则弱上拉器件被自动禁止。  
 2: 在XT、HS和LP振荡模式下PAPHR<5:4>始终读为1。

### 寄存器97H: PORTA弱下拉控制寄存器 (PAPDR)

U-0	U-0	R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1
—	—	PAPDR5	PAPDR 4	—	PAPDR 2	PAPDR 1	PAPDR 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

bit7-6            未实现: 读为0

bit5-0            **PAPDR<5:0>**: PORTA弱下拉控制位  
 1 = 使能PORTA相应引脚下拉  
 0 = 禁止PORTA相应引脚下拉

### 寄存器9CH: PORTC弱上拉控制寄存器 (PCPHR)

R/W-1							
PCPHR7	PCPHR6	PCPHR5	PCPHR4	PCPHR3	PCPHR2	PCPHR1	PCPHR0
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            **PCPHR<7: 0>**: PORTC弱上拉控制位  
 1 = 使能PORTC相应引脚上拉  
 0 = 禁止PORTC相应引脚上拉

寄存器9DH PORTC弱下拉控制寄存器 (PCPDR)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PCPDR7	PCPDR 6	PCPDR 5	PCPDR4	PCPDR3	PCPDR2	PCPDR1	PCPDR0
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            **PCPDR <7:0>**: PORTC弱下拉控制位  
 1 = 使能PORTC相应引脚下拉  
 0 = 禁止PORTC相应引脚下拉

### 6.3.3 电平变化中断

每一个 PORTA 引脚均可分别配置为电平变化中断引脚。控制位 PAINTR\_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 时锁存的值进行比较。将上一次“不匹配”的输出一起作逻辑或运算，以便将 INTS 寄存器(寄存器 0BH)中 PORTA 电平变化中断标志位(PAIF)置 1。

该中断能唤醒休眠下的器件。用户在中断服务程序中通过以下方式清除中断:

- 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 清零。

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

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

寄存器 96H: PORTA电平变化中断控制寄存器 (PAINTR)

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

图注:

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

bit7-6            未实现: 读为0。

bit5-0            **PAINTR<5:0>**: PORTA电平变化中断控制位  
 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。

### 6.3.4 超低功耗唤醒

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

要使用该功能，PA0 引脚应先被配置为输出高电平以对电容充电，允许 PA0 的电平变化中断并且 PA0 要配置为输入引脚。将 ULPWUE 置 1 开始放电，执行 SLEEP 指令。当 PA0 上的电压下降到 VIL 后，器件将被唤醒并执行下一条指令。如果 INTS 寄存器的 GIE 位置 1，器件将调用中断服务程序 (0004H)。更多信息，请参见第 6.3.3 节“电平变化中断”和第 5.4 节“PORTA 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。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
BSF   STATUS,PAGE   ;Bank 1
BCR   ADINA0        ;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   PAINTR0       ;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
```

## 6.4 PORTA 引脚说明和引脚图

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

### 6.4.1 PA0/AN0/ISCK/ULPWU/[PWMA0]

图 6-4-1 给出了此引脚的引脚图。PA0/AN0/ISCK/ULPWU/[PWMA0] 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
- 在线串行编程 (In-Circuit Serial Programming™) 烧录和调试的时钟 ISCK
- 超低功耗唤醒的模拟输入
- 功能转移后 PWMA0 信号输出

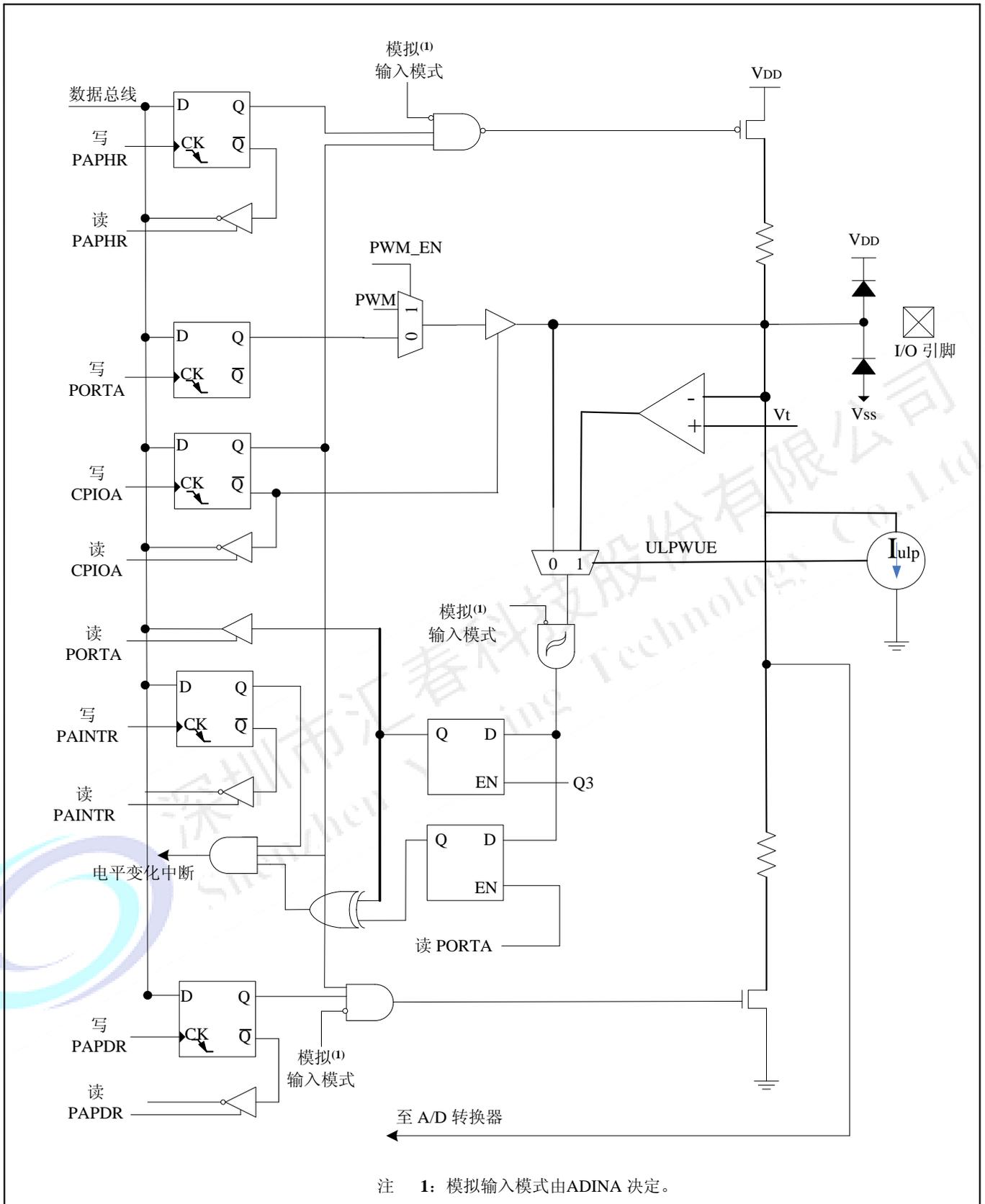


图 6-4-1: PA0 框图

## 6.4.2 PA1/AN1/ISDA/[PWMA1]

图 6-4-2 给出了此引脚的引脚图。PA1/AN1/ISDA/[PWMA1] 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
- 在线串行编程烧录和调试的数据端口 ISDA
- 功能转移后 PWMA1 信号输出

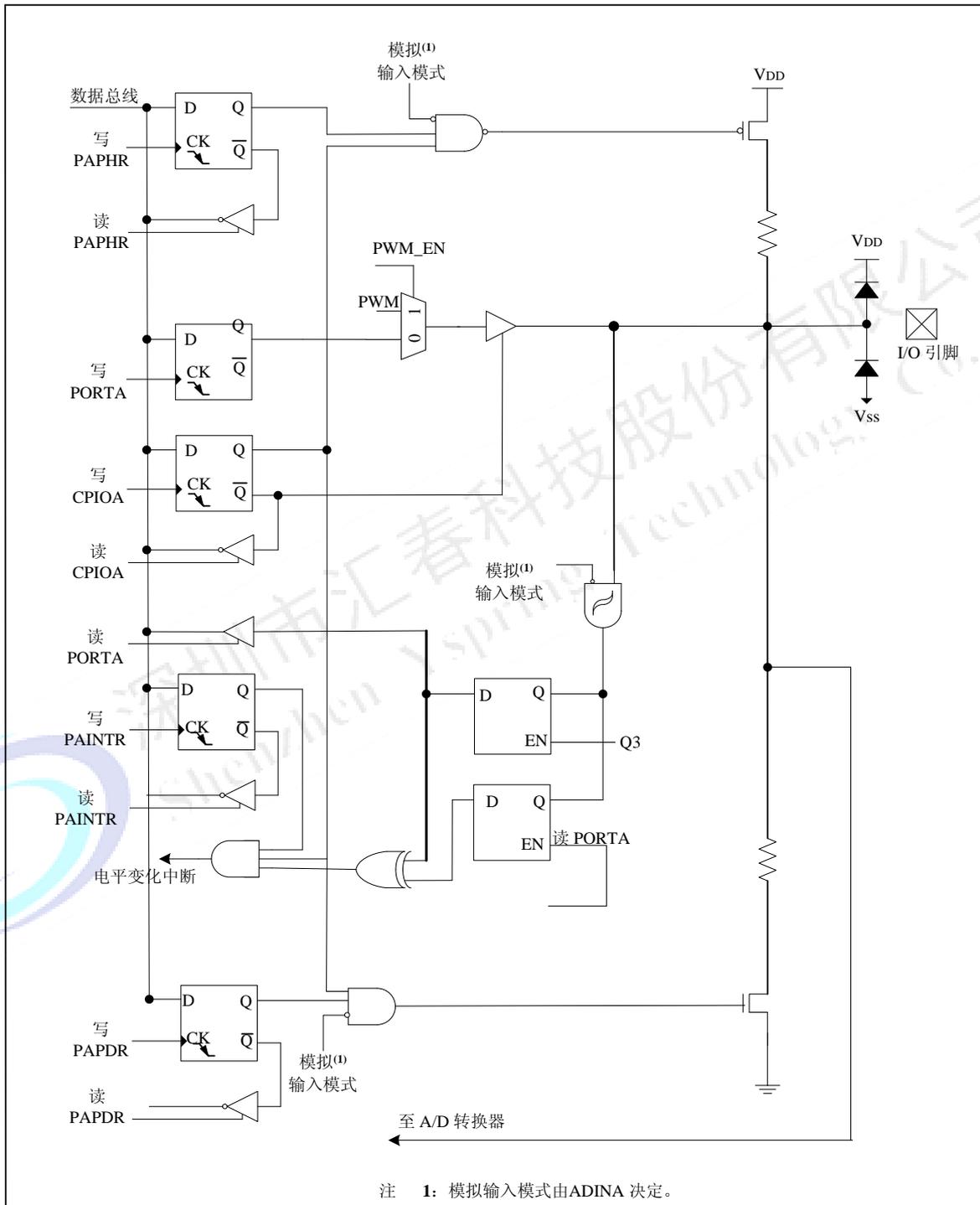


图 6-4-2: PA1 框图

### 6.4.3 PA2/AN2/T0CKI/[PWMB0]

图 6-4-3 给出了此引脚的引脚图。PA2/AN2/T0CKI/[PWMB0] 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
- TIMER0 外部时钟输入
- 功能转移后 PWMB0 信号输出

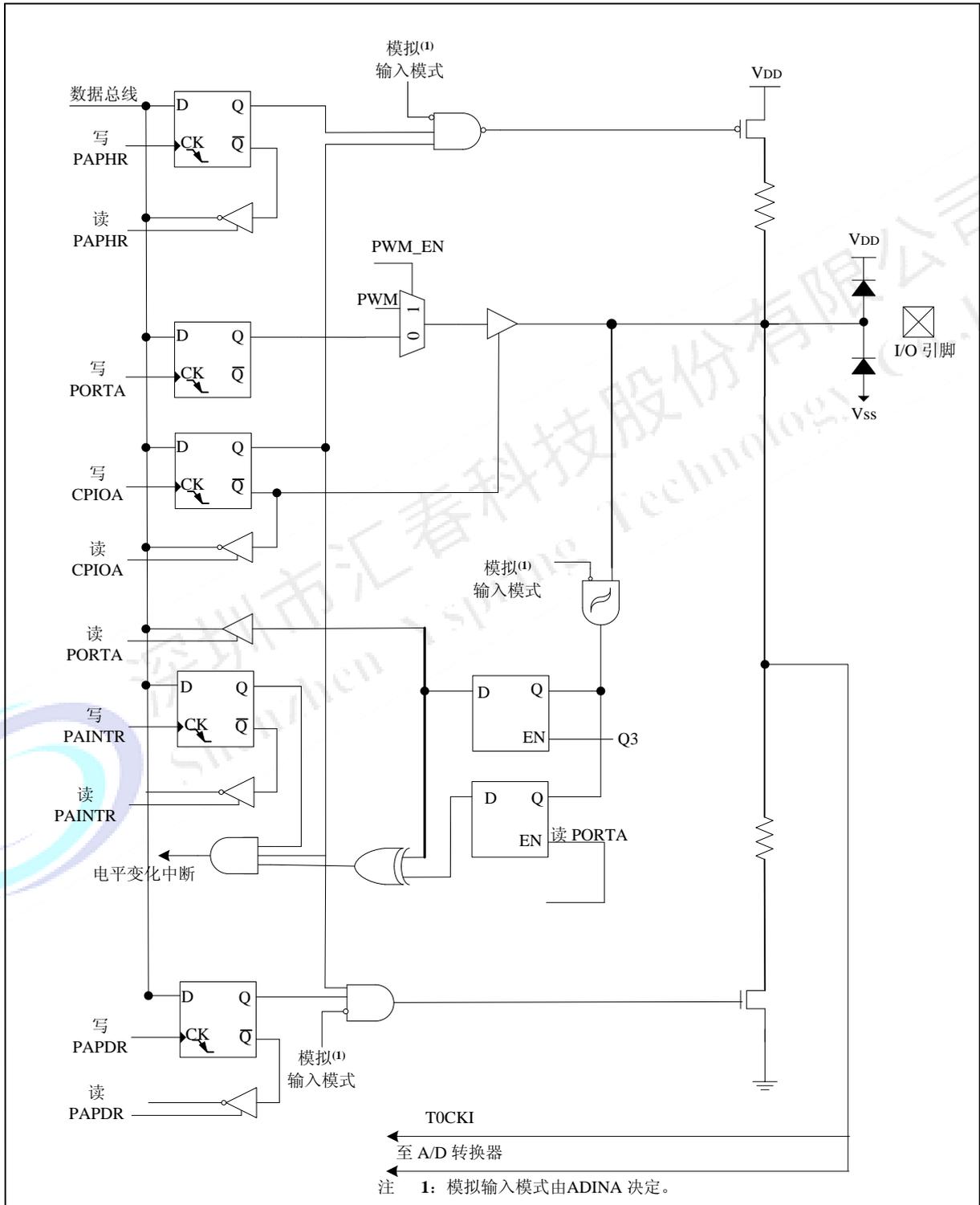


图 6-4-3: PA2 框图

## 6.4.4 PA3/MCLR/VPP

图 6-4-4 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；
- 高压烧写引脚

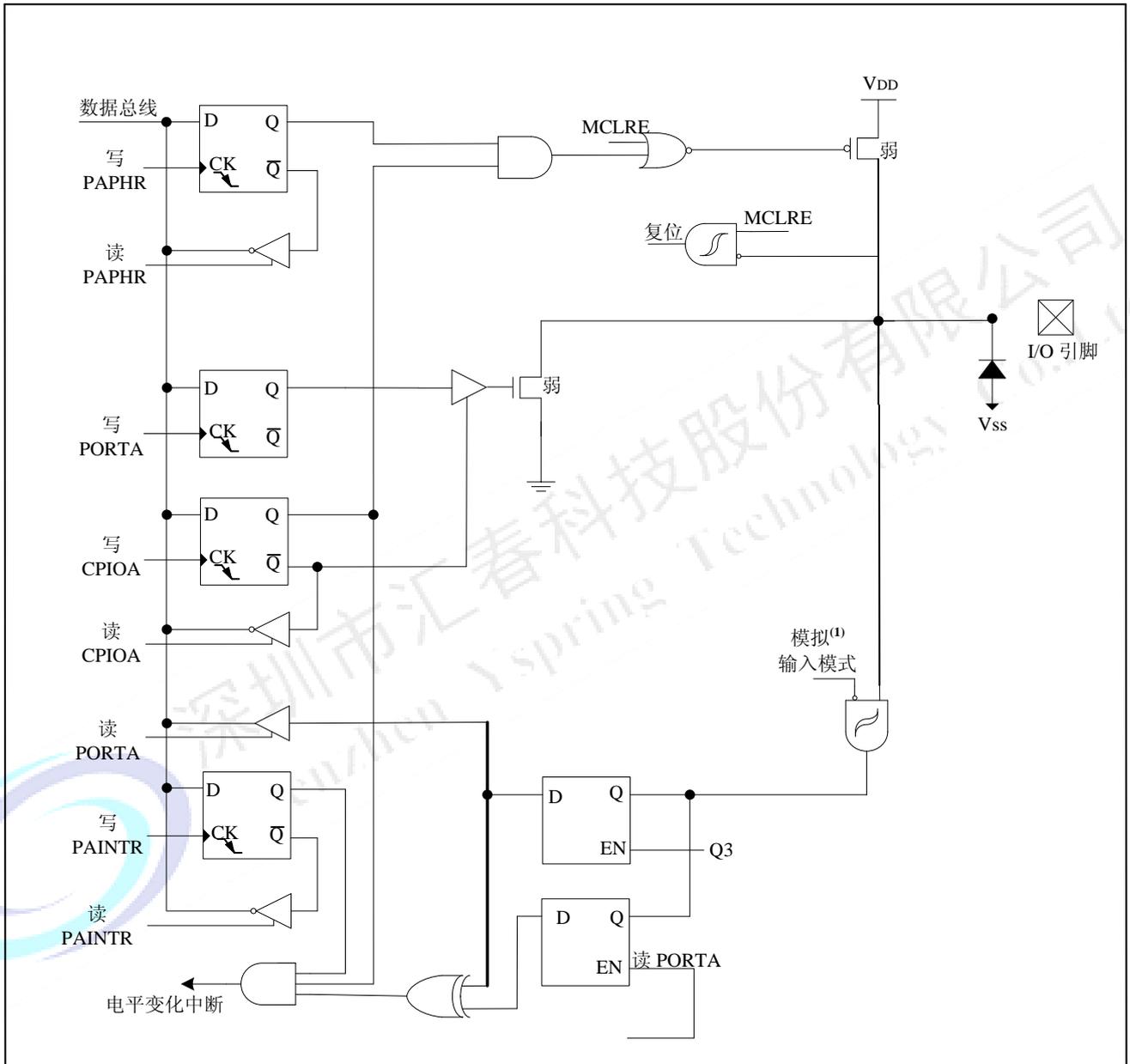


图 6-4-4: PA3 框图

## 6.4.5 PA4/AN3/T12G/OSC2/CLKOUT

图 6-4-5 给出了此引脚的引脚图。PA4/AN3/T12G/OSC2/CLKOUT 引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入
- Timer1/2 门控（计数使能）
- 晶振/谐振器连接
- 时钟输出

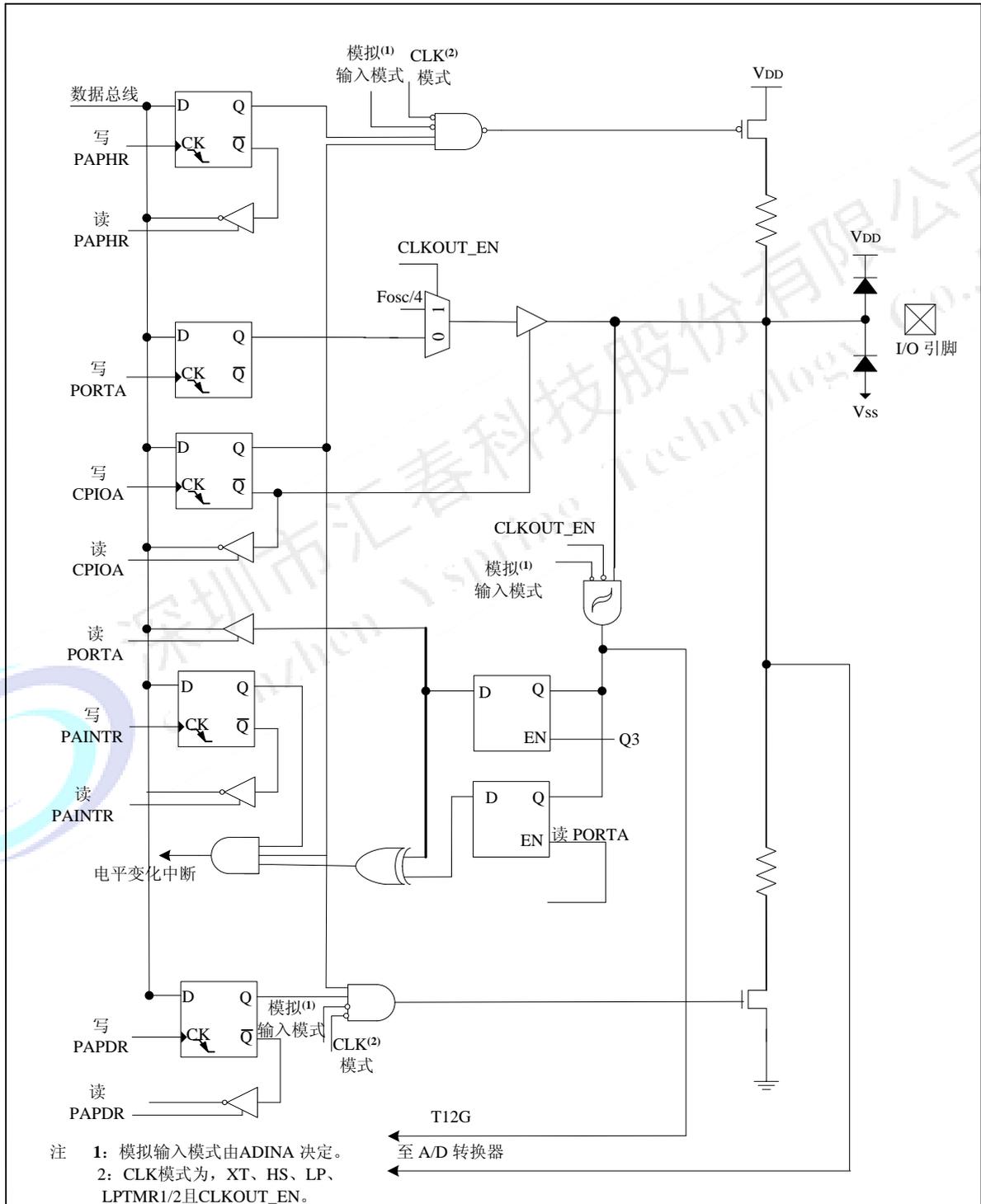


图 6-4-5: PA4 框图

## 6.4.6 PA5/AN4/T12CKI/OSC1/CLKIN/[PWMB1]

图 6-4-6 给出了此引脚的引脚图。PA5/AN4/T12CKI/OSC1/CLKIN/[PWMB1] 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
- TMR1/2 外部时钟输入
- 晶振/谐振器连接
- 时钟输入
- 功能转移后 PWMB1 信号输出

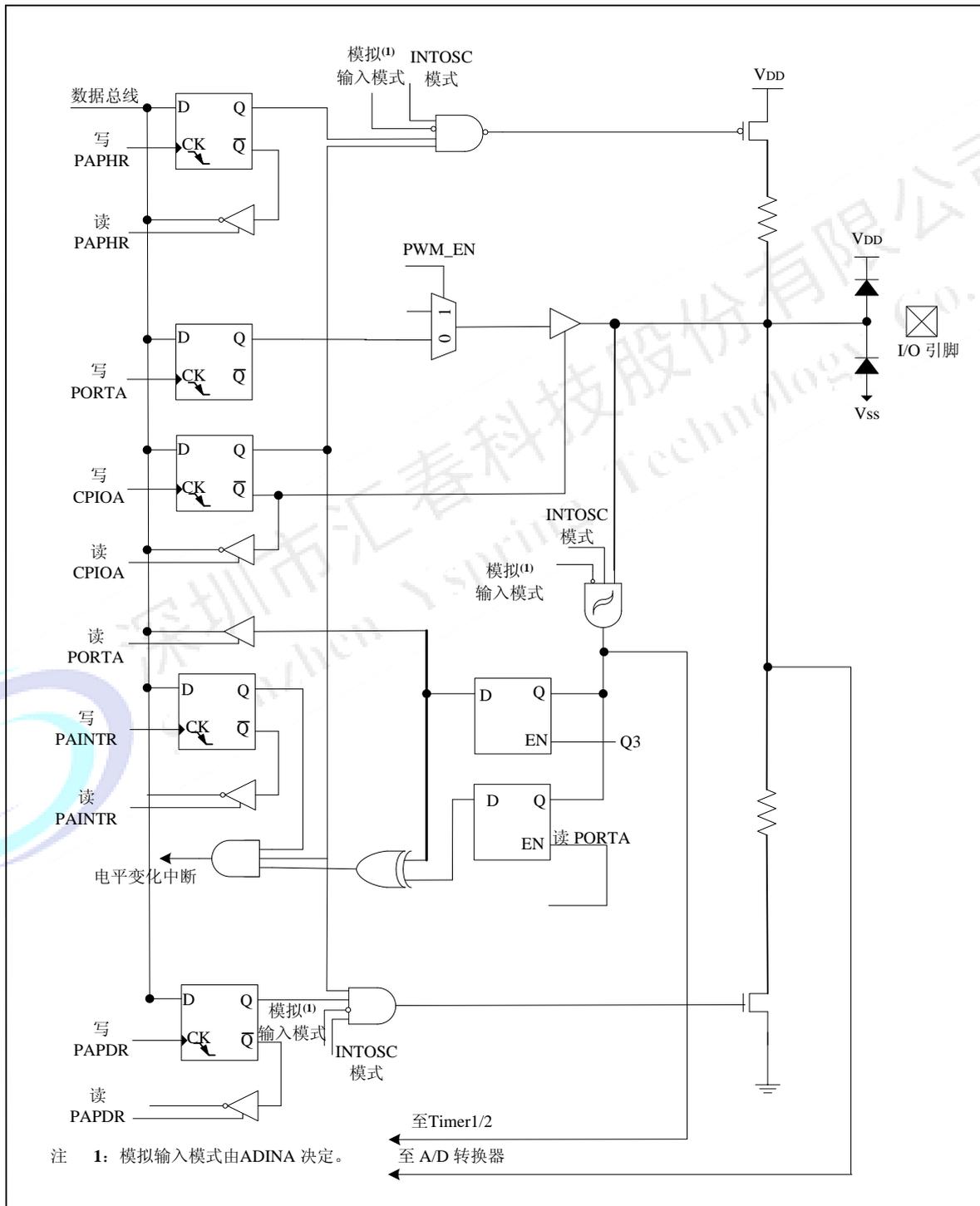


图 6-4-6: PA5 框图

表 6-4: 与 PORTA 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 LVR 时的值	所有其他复位值
ADINA	—	—	AN4	AN3	—	AN2	AN1	AN0	0111_1111	xxxx_xxxx
PSTA	—	—	ULPWUE	SLVREN	MCLR	IER	POR	LVR	--01_--qq	--0u_--uu
INTS	GIE	PE1E	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000_0000	xxxx_xxxx
PAINTR	—	—	PAINTR5	PAINTR4	PAINTR3	PAINTR2	PAINTR1	PAINTR0	--00_0000	xxxx_xxxx
OPT_REG	—	TMR0EN	TCS	TCE	PSC	PS2	PS1	PS0	-111_1111	xxxx_xxxx
PORTA	—	—	PA5	PA4	PA3	PA2	PA1	PA0	--x0_x000	xxxx_xxxx
CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11_1111	xxxx_xxxx
PAPHR	—	—	PAPHR5	PAPHR4	PAPHR3	PAPHR2	PAPHR1	PAPHR0	--00_0000	xxxx_xxxx
PAPDR	—	—	PAPDR 5	PAPDR 4	—	PAPDR 2	PAPDR 1	PAPDR 0	--00_0000	xxxx_xxxx

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

深圳市汇春科技股份有限公司  
Shenzhen Yspring Technology Co., Ltd.

## 6.5 PORTC 引脚说明和引脚图

### 6.5.1 PC0/AN5/PWMB1

PC0 引脚可以配置为下列功能之一：

- 通用 I/O
- ADC 的模拟输入
- 功能转移前 PWMB1 信号输出

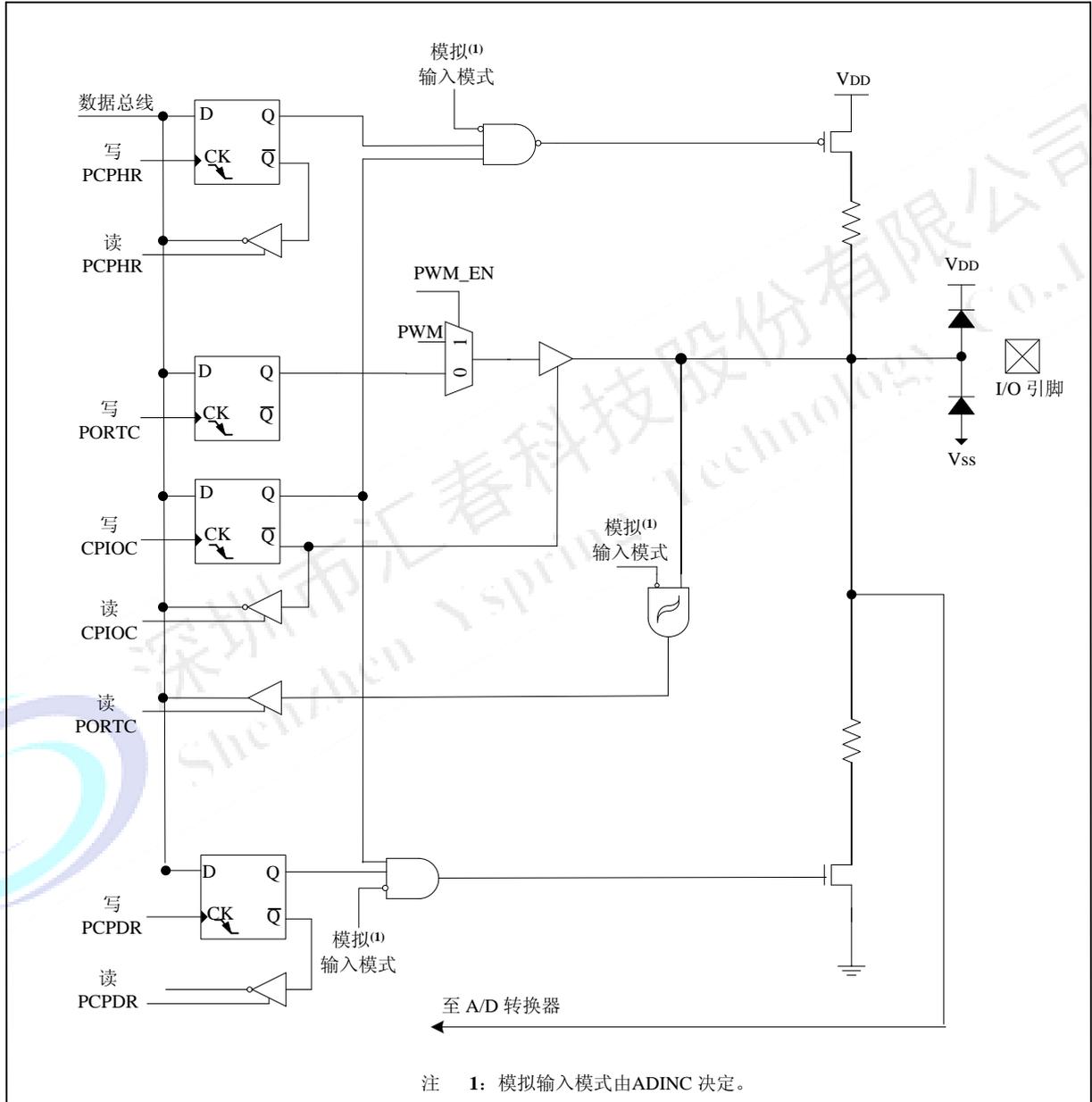


图 6-5-1: PC0 框图

## 6.5.2 PC1/AN6/VREF

PC1 引脚可以配置为下列功能之一：

- 通用 I/O
- ADC 的模拟输入
- ADC 外部参考引脚

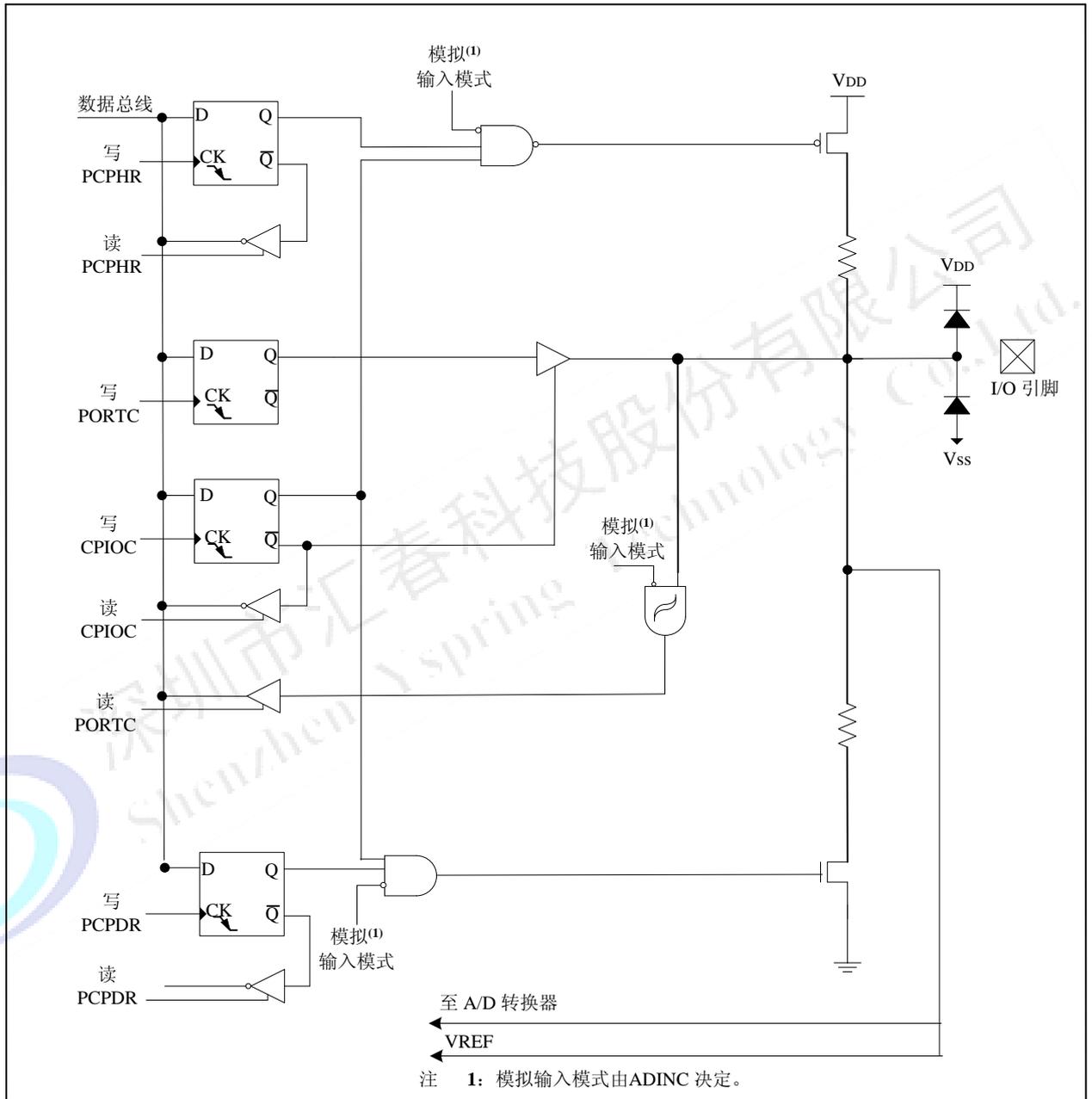


图 6-5-2: PC1 框图

### 6.5.3 PC2 /AN7/INT0

PC2 引脚可以配置为下列功能之一：

- 通用 I/O
- 外部中断 INT0
- ADC 的模拟输入

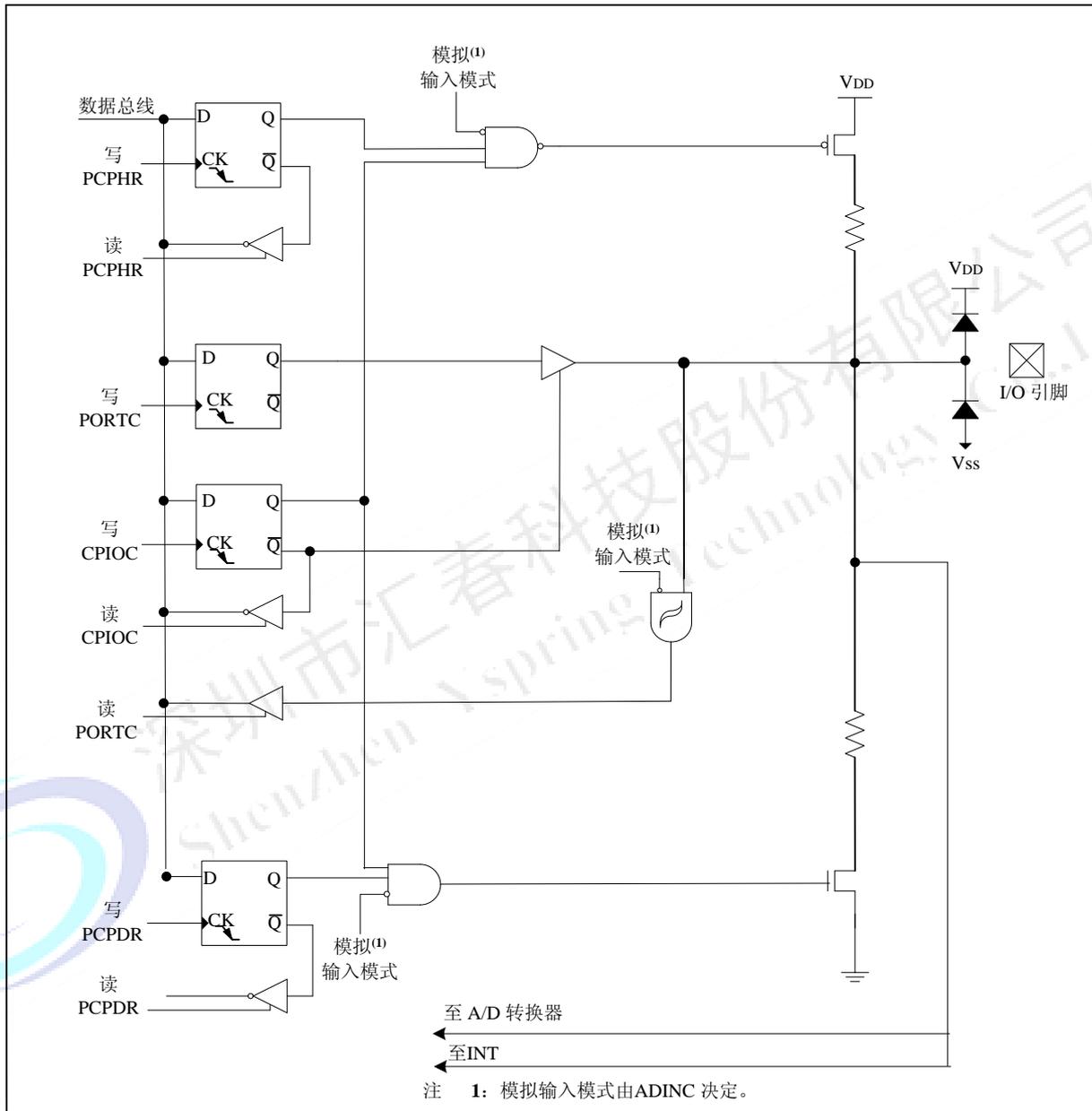


图 6-5-3: PC2 框图

## 6.5.4 PC3/AN8/PWMB0/INT1

PC3 引脚可以配置为下列功能之一：

- 通用 I/O
- 外部中断 INT1
- ADC 的模拟输入
- 功能转移前 PWMB0 信号输出

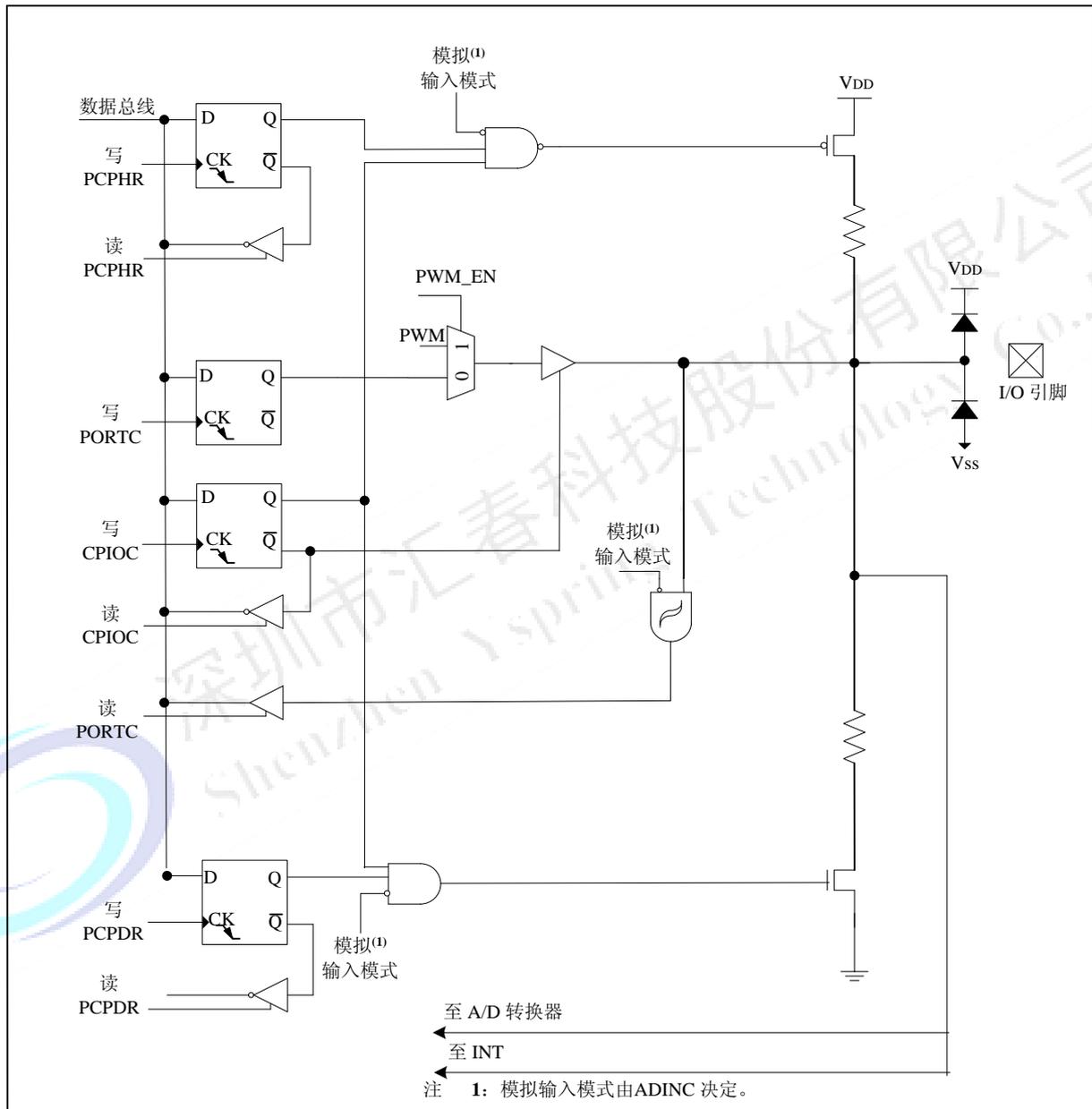


图 6-5-4: PC3 框图



## 6.5.6 PC5/AN10/PWMA0

PC5 引脚可以配置为下列功能之一：

- 通用 I/O
- ADC 的模拟输入
- 功能转移前 PWMA0 信号输出

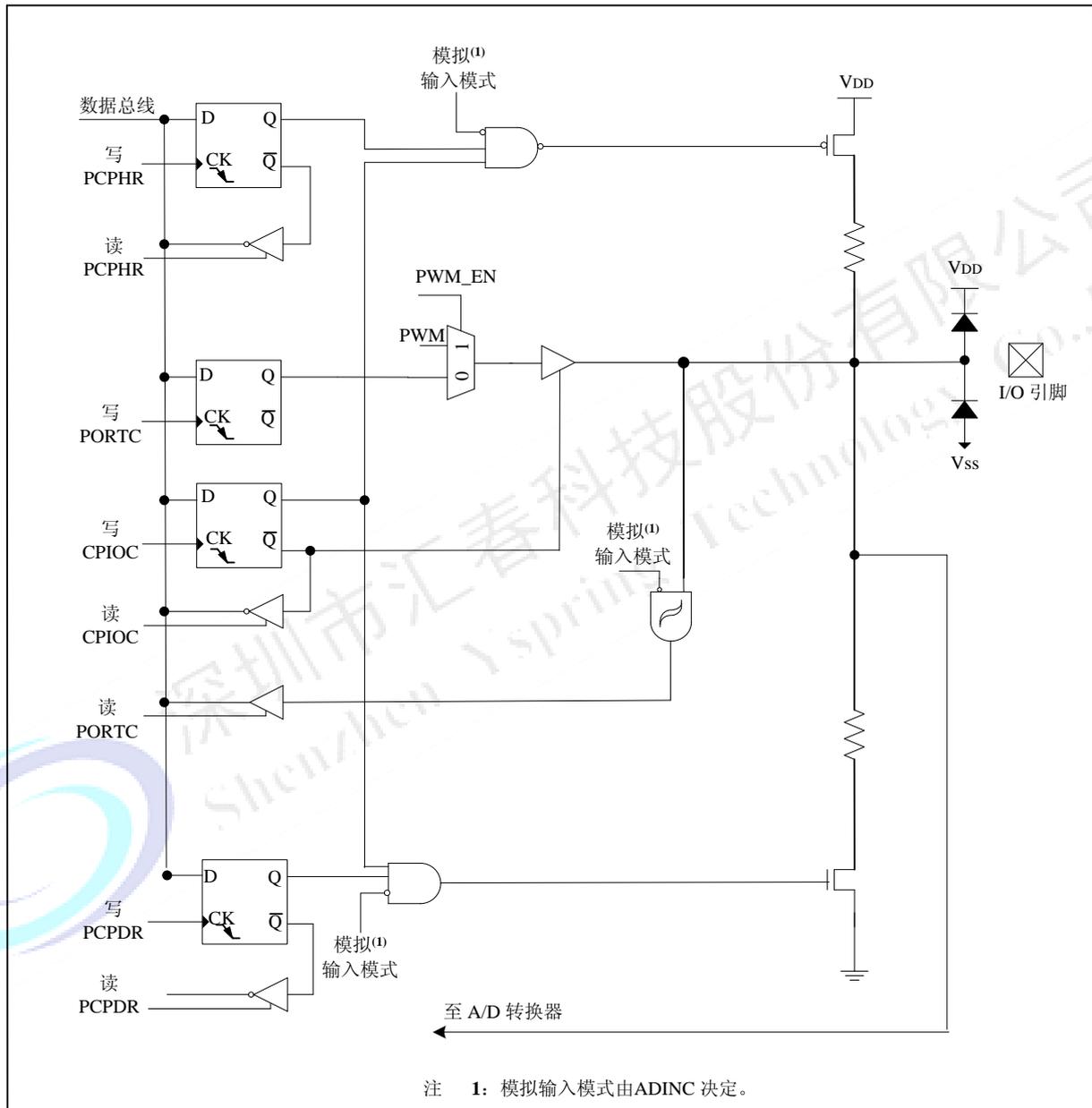


图 6-5-6: PC5 框图

## 6.5.7 PC6/AN11

PC6 引脚可以配置为下列功能之一：

- 通用 I/O
- ADC 的模拟输入

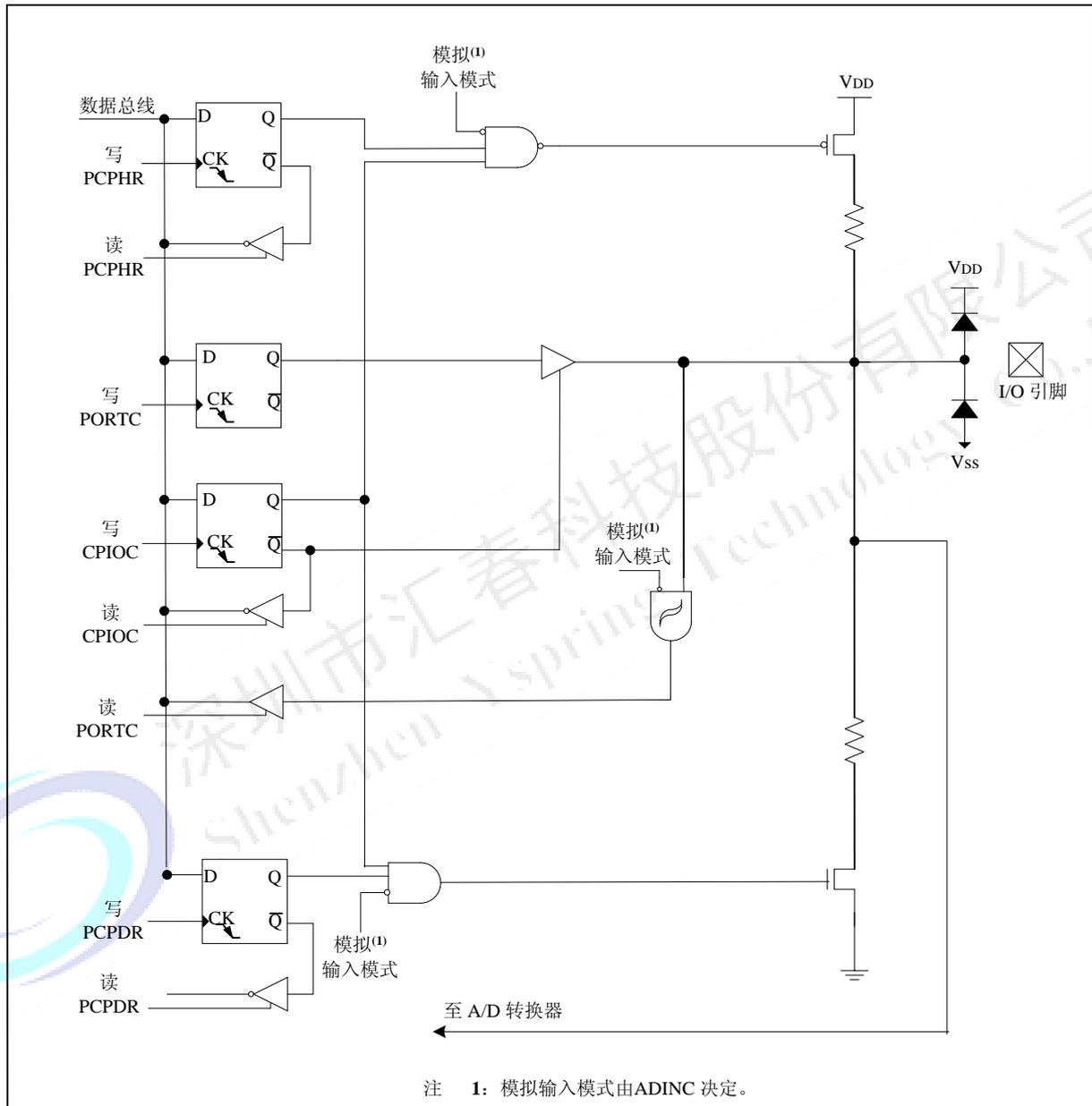


图 6-5-7: PC6 框图

## 6.5.8 PC7/AN12

PC7 引脚可以配置为下列功能之一：

- 通用 I/O
- ADC 的模拟输入

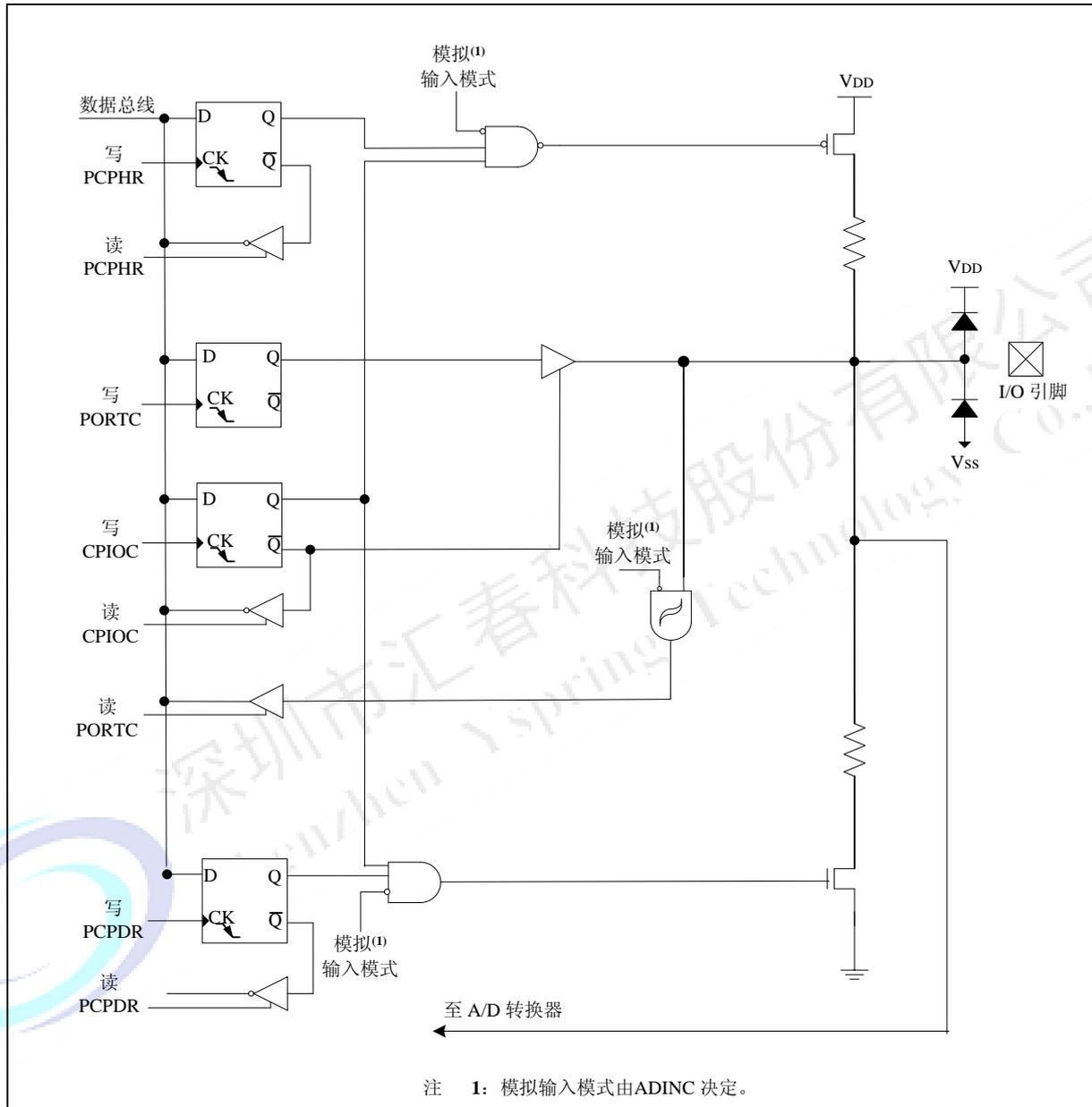


图 6-5-8: PC7 框图

表 6-5: 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 LVR 时的值	所有其他复位值
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	xxxx_xxxx	xxxx_xxxx
CPIOC	CPIOC7	CPIOC6	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	1111_1111	xxxx_xxxx
PCPDR	PCPDR 7	PCPDR6	PCPDR5	PCPDR4	PCPDR3	PCPDR2	PCPDR1	PCPDR0	0000_0000	xxxx_xxxx
PCPHR	PCHR7	PCHR6	PCPHR5	PCPHR4	PCPHR3	PCPHR2	PCPHR1	PCPHR0	0000_0000	xxxx_xxxx
ADINC	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	1111_1111	xxxx_xxxx

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

深圳市汇春科技股份有限公司  
Shenzhen Yspring Technology Co., Ltd.

## 7 定时器模块/PWM 模块

### 7.1 Timer0 模块

Timer0 模块是 8 位可重载定时器/计数器，具备以下特性：

- 8 位定时器/计数器寄存器（TMR0）
- 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

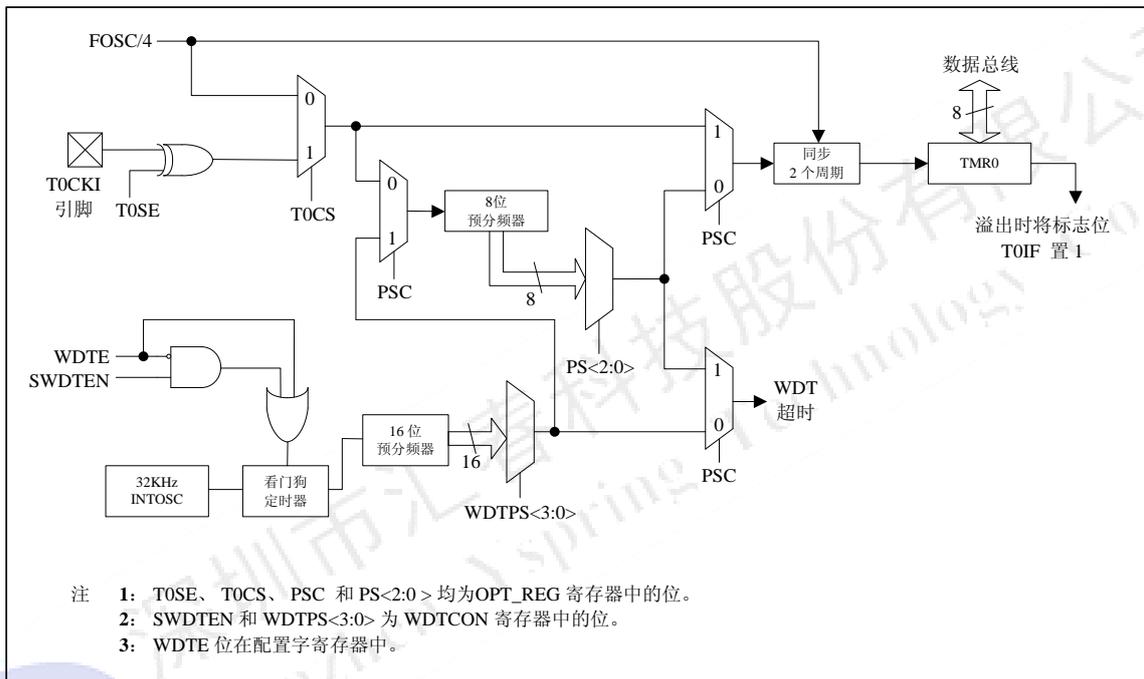


图 7-1: Timer0 模块的框图

#### 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 TMR0，读的是计数器的值，写 TMR0，写的是重载寄存器里的值。

寄存器 01H: Timer0 模块寄存器 (TMR0)

| R/W-x |
|-------|-------|-------|-------|-------|-------|-------|-------|
| TMR07 | TMR06 | TMR05 | TMR04 | TMR03 | TMR02 | TMR01 | TMR00 |
| 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

**TMR0 <7:0>**: 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 TMR0EN 见寄存器 OPT\_REG 寄存器 ([寄存器 81H](#))，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 OPT\_REG 寄存器 ([寄存器 81H](#)) 的 TCS 位清零选择定时器模式。

当 TMR0 被写入时，写入后将立即禁止 2 周期递增。

注：在 TMR0 被写入时，为了计入两个指令周期的延时，可以调整写入 TMR0 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 OPT\_REG 寄存器 ([寄存器 81H](#)) 的 TCE 位决定。将 OPT\_REG 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 OPT\_REG 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 OPT\_REG 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 TMR0 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，TMR0 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 TMR0 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 TIS 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 11.0 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 LVR 时的值	所有其他复位值
TMR0	Timer0 模块寄存器								xxxx_xxxx	uuuu_uuuu
INTS	GIE	PE1E	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000_0000	0000_0000
OPT_REG	—	TMR0EN	TCS	TCE	PSC	PS2	PS1	PS0	1111_1111	1111_1111
CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11_1111	--11_1111

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

## 7.2 带门控的 Timer1/ Timer2 模块

Timer1/2 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（TMRxH:TMRxL）
- 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
- T12G 引脚的 Timer1/2 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）

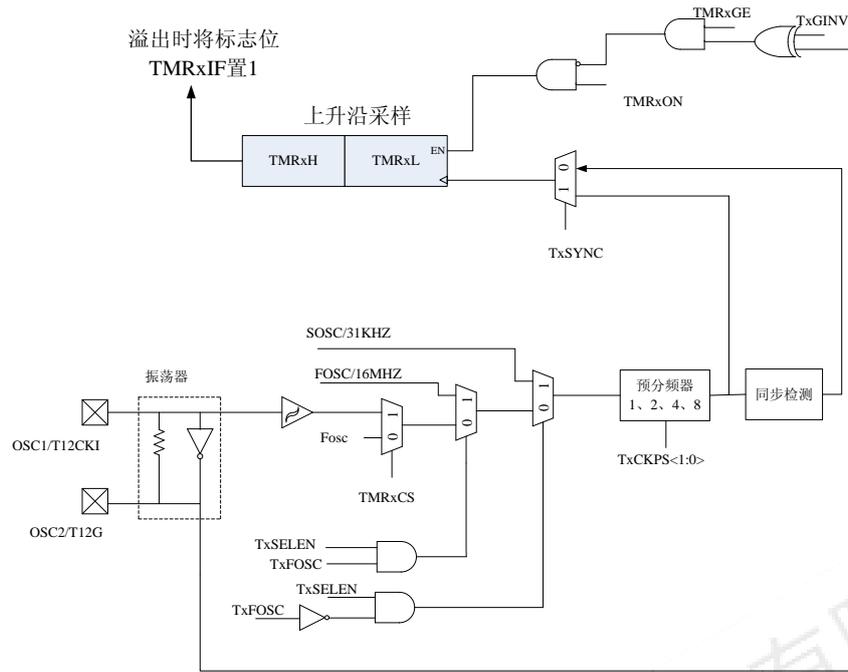


图 7-2: Timer1/2 模块的框图

## 7.2.1 Timer1/2 的工作原理

Timer1/2 模块是 16 位递增计数器，通过一对寄存器 TMRxH:TMRxL 访问。对 TMRxH 或 TMRxL 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 和寄存器 13H 所示，Timer1/2 控制寄存器 TxSTA 用于控制并选择定时器模块的各种功能。

## 7.2.2 Timer1/2 的工作原理

Timer1/2 模块是 16 位递增计数器，通过一对寄存器 TMRxH:TMRxL 访问。对 TMRxH 或 TMRxL 的写操作将直接更新计数器。与内部时钟源配合使用时，该模块为定时器。与外部时钟源配合使用时，该模块可用作定时器或计数器。

## 7.2.3 Timer1/2 时钟源选择

寄存器 T12CON 的 TxSELEN = 0 时，Timer\_x 的时钟源由 TMRxCS 位确定。TxSELEN = 1 时，Timer\_x 的时钟源由 TxFOSC 位确定。下表列出了时钟源的选择方式。

表 7-3-2: 时钟源选择方式

TxSELEN	TMRxCS	TxOSCEN	TxFOSC	时钟源
0	0	x	x	系统时钟
0	1	0	x	T12CKI 引脚外部时钟源
0	1	1	x	OSC1/OSC2 引脚上的 LP 振荡电路
1	x	x	0	内部慢时钟
1	x	x	1	内部快时钟

注: TxSELEN 为 1 时，MCU 将一直处于工作状态，无法进入睡眠。

## 7.2.4 Timer1/2 外部时钟源

选择外部时钟源时，Timer1/2 模块可作为定时器，也可作为计数器工作。

计数时，Timer1/2 在外部时钟输入 T12CKI 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 Timer1/2 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Timer1/2
- 写入TMRxH或TMRxL
- Timer1/2被禁止
- T12CKI为高电平时Timer1/2被禁止
- (TMRxON = 0)，然后在T12CKI为低电平时Timer1/2被使能 (TMRxON = 1)。

请参见图7-2-4

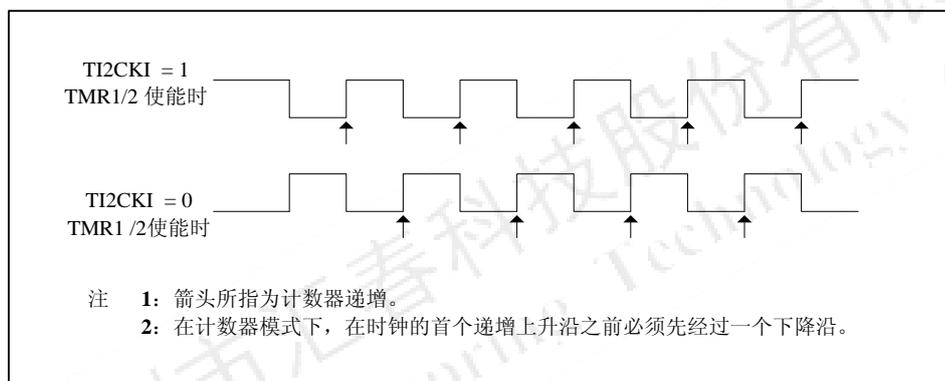


图 7-2-4: TIMER1/2 递增边沿

## 7.2.5 Timer1/2 的预分频器

Timer1/2 有四种预分频选项，可对时钟输入进行 1、2、4 或 8 分频。TxSTA 寄存器的 TxCKPS 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 TMRxH 或 TMRxL 的写操作时，预分频计数器被清零。

## 7.2.6 Timer1/2 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。TxSTA 寄存器将的 TxOSCEN 控制位置 1 使能振荡器。休眠时振荡器继续工作。

Timer1/2 振荡器与系统 LP 振荡器共用。这样，Timer1/2 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。

Timer1/2 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TxOSCEN置1并在使能Timer1前应有一段适当的延时。

## 7.2.6 Timer1/2 工作于异步计数器模式

若 **TxSTA** 寄存器的控制位 **TxSYNC** 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 7.2.6.1 节“在异步计数器模式下读写 Timer1/2”）。

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。

### 7.2.6.1 在异步计数器模式下读写 Timer1/2

定时器运行于外部异步时钟时，读取 **TMRxH** 或 **TMRxL** 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 **TMRxH:TMRxL** 这对寄存器中产生不可预测的值。

## 7.2.7 Timer1/2 门控

Timer1/2 门控源可软件配置为 **T12G** 引脚，这使器件可直接使用 **T12G** 为外部事件定时。使用 **TxSTA** 寄存器的 **TxGINV** 位可翻转 Timer1/2 门控。这将配置 Timer1/2 以确保事件之间存在低电平有效或高电平有效的有效时间。

## 7.2.8 Timer1/2 中断

Timer1/2 的一对寄存器（**TMRxH:TMRxL**）递增至 **FFFFH** 后将产生溢出中断。**PIFB0** 和 **PIFB1** 寄存器的中断标志位被置 1。为确保计满返回时产生中断，您必须将以下位置 1：

- **TxSTA** 寄存器的 **TMRxON** 位
- **PIFB0** 和 **PIFB1** 寄存器的 **TMRxIE** 位
- **INTS** 寄存器的 **PEIE** 位
- **INTS** 寄存器的 **GIE** 位

在中断服务程序中将 **TMRxIF** 位清零将清除中断标志。

注：**TMRxH:TMRxL** 这对寄存器和 **TMRxIF** 位应在允许中断前清零。

## 7.2.9 Timer1/2 在休眠模式下的工作原理

只有在设定异步计数器模式时，Timer1/2 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 **TxSTA** 寄存器的 **TMRxON** 位置 1
- 必须将 **PIFB0** 或 **PIFB1** 寄存器的 **TMRxIE** 位置 1
- 必须将 **INTS** 寄存器的 **PEIE** 位置 1
- 必须将 **TxSTA** 寄存器的 **TxSYNC** 位置 1
- 必须将 **TxSTA** 寄存器的 **TMRxCS** 位置 1
- 可将 **TxSTA** 寄存器的 **TxOSCEN** 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）。

**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
<b>T1GINV</b>	<b>TMR1GE</b>	<b>T1CKPS1</b>	<b>T1CKPS0</b>	<b>T1OSCEN</b>	<b>T1SYNC</b>	<b>TMR1CS</b>	<b>TMR1ON</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 **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T12CKI引脚 (上升沿) 的外部时钟  
0 = 内部时钟 (FOSC)
- 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T12G引脚。

Timer1/2 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                |
|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| <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 &lt;15:8&gt;: 16位Timer1定时器/计数器高字节寄存器

## 寄存器13H: TIMER2控制寄存器 (T2STA)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T2GINV	TMR2GE	T2CKPS1	T2CKPS0	T2OSCEN	T2SYNC	TMR2CS	TMR2ON
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

**T2GINV:** Timer2门控翻转位<sup>(1)</sup>

1 = Timer2门控为高电平有效 (门控为高电平时Timer2计数)

0 = Timer2门控为低电平有效 (门控为低电平时Timer2计数)

bit6

**TMR2GE:** Timer1门控使能位<sup>(2)</sup>

如果TMR2ON = 0:

此位被忽略

如果TMR2ON = 1:

1 = Timer2在Timer1门控不活动时打开

0 = Timer2打开

bit5-4

**T2CKPS<1:0>:** Timer2输入时钟预分频比选择位

11 = 1:8

10 = 1:4

01 = 1:2

00 = 1:1

bit3

**T2OSCEN:** 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Timer2时钟

0 = LP振荡器关闭

否则:

此位被忽略

bit2

**T2SYNC:** Timer2外部时钟输入同步控制位

TMR2CS = 1:

1 = 不同步外部时钟输入

0 = 同步外部时钟输入

TMR2CS = 0:

此位被忽略。Timer2使用内部时钟

bit1

**TMR2CS:** Timer2时钟源选择位

1 = 来自T12CKI引脚 (上升沿) 的外部时钟

0 = 内部时钟 (FOSC)

bit0

**TMR1ON:** Timer2打开位

1 = 使能Timer2

0 = 停止Timer2

注

1: 无论门控源为何, T2GINV位均将翻转Timer2门控逻辑。

2: 必须将TMR2GE位置1以使用T12G引脚。

## 寄存器 11H: 16位Timer2低字节的保持寄存器 (TMR2L)

| R/W-x   |
|---------|---------|---------|---------|---------|---------|---------|---------|
| TMR2<7> | TMR2<6> | TMR2<5> | TMR2<4> | TMR2<3> | TMR2<2> | TMR2<1> | TMR2<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

TMR2 &lt;7:0&gt;: 16位Timer2定时器/计数器低字节寄存器

## 寄存器 12H: 16位Timer2高字节的保持寄存器 (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
TMR2<15>	TMR2<14>	TMR2<13>	TMR2<12>	TMR2<11>	TMR2<10>	TMR2<9>	TMR2<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

TMR2 &lt;15:8&gt;: 16位Timer2定时器/计数器高字节寄存器

## 寄存器 14H: TIMER1/2内部快慢时钟源选择控制寄存器 (T12CON)

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
DIE_EN	DIE_NUM<2: 0>			T2SELEN	T2FOSC	T1SELEN	T1FOSC
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

DIE\_EN: PWM死区延时使能位, 高有效。

0: 无死区时间延时

1: 在TMR1控制的PWMA0、PWMA1互补输出之间加入死区延时

bit6-4

DIE\_NUM&lt;2:0&gt;:死区延时时间计数

注: 该计数器仅当死区延时使能位置1时有效;

PWM死区时间 = (NUM+1) / Fsys

bit3

T2SELEN: Timer2 时钟源选择位

1 = Timer2 时钟源由 T2FOSC 位确定。

0 = Timer2 时钟源由TMR2CS位确定。

bit2

T2FOSC: Timer2 时钟源选择内部快时钟使能位。

当 T2SELEN = 0 时 : 忽略此位。

当 T2SELEN = 1 时 :

1 = Timer2 时钟源选择内部 16MHZ 快时钟。

0 = Timer2 时钟源选择内部慢时钟。

bit1

T1SELEN: Timer1 时钟源选择位

1 = Timer1 时钟源由 T1FOSC 位确定。

0 = Timer1 时钟源由TMR1CS位确定。

bit0

T1FOSC: Timer1 时钟源选择内部快时钟使能位。

当 T1SELEN = 0 时 : 忽略此位。

当 T1SELEN = 1 时 :

1 = Timer1 时钟源选择内部16MHZ快时钟。

0 = Timer1 时钟源选择内部慢时钟。

## 7.2.10 与 Timer1/2 相关的寄存器汇总

表 7-2-10: 与 TIMER1/2 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 LVR 时的值	所有其他复位值
T12CON	—	—	—	—	T2SELEN	T2FOSC	T1SELEN	T1FOSC	---- 0000	xxxx xxxx
INTS	GIE	PEIE	TIS	INT0E	PAIE	TIF	INT0F	PAIF	0000 0000	xxxx xxxx
PIFB0	INT1E	ADCIE	OSFIE	TMR1IE	INT1IF	ADCIF	OSFIF	TMR1IF	0000 0000	xxxx xxxx
PIFB1	INT1SEL	PWMBIE	PWMAIE	TMR2IE	INT0SEL	PWMBF	PWMAF	TMR2IF	0000 0000	xxxx xxxx
TMR1H	16 位 TMR1 高字节的保持寄存器								xxxx xxxx	xxxx xxxx
TMR1L	16 位 TMR1 低字节的保持寄存器								xxxx xxxx	xxxx xxxx
T1STA	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	xxxx xxxx
TMR2H	16 位 TMR2 高字节的保持寄存器								xxxx xxxx	xxxx xxxx
TMR2L	16 位 TMR2 低字节的保持寄存器								xxxx xxxx	xxxx xxxx
T2STA	T2GINV	TMR2GE	T2CKPS1	T2CKPS0	T2OSCEN	T2SYNC	TMR2CS	TMR2ON	0000 0000	xxxx xxxx

图注: x = 未知, u = 不变, — = 未实现 (读为 0)。Timer1/2 不使用阴影单元。

## 7.3 PWM 模块

利用 Timer1/2 的 16 位定时器最多可以生成 4 路 PWM 信号, 最高分辨率可达 12 位。每一个定时器最多生成两路 PWM 信号, Timer1 生成的 PWM 信号在引脚功能转移前由 PC5(PWMA0)、PC4(PWMA1)输出, 引脚功能转移之后由 PA0(PWMA0)、PA1(PWMA1)输出。Timer2 生成的 PWM 信号在引脚功能转移之前由 PC3(PWMB0)、PC0(PWMB1)输出, 引脚功能转移后由 PA2(PWMB0)、PA5(PWMB1) 输出。

### 7.3.1 PWM 的输出

PWM 使能时, 当{ $TMRxH, TMRxL$ } 与{ $TxPRLH, TxPRL$ }相等时, 下一计数周期 Timer1/2 计数器清零。当{ $TMRxH, TMRxL$ } = 0 时, PWM 输出高电平, 当{ $TMRxH, TMRxL$ } 大于等于 { $PWMXXH, PWMXXL$ }时, PWM 输出低电平。其占空比, 周期和分辨率由以下寄存器决定:

- TxPRL
- TxPRLH
- PWMXXL
- PWMXXH

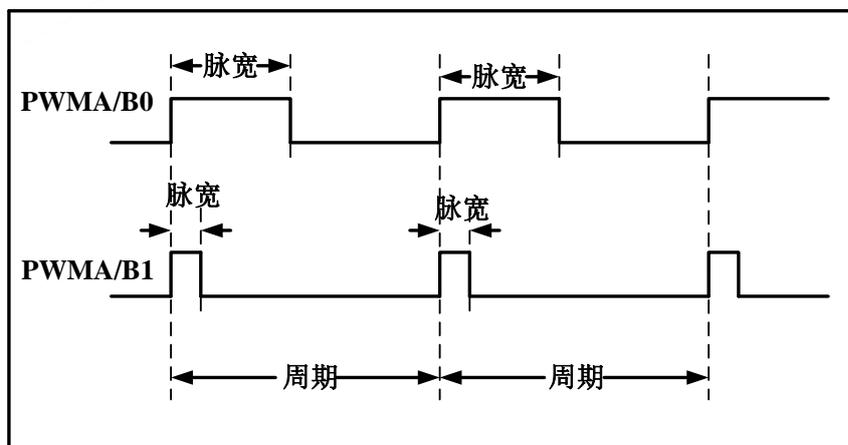


图7-3-3 PWM输出

### 7.3.2 PWM 的周期与占空比

PWM 周期由 Timer1/2 的 TxPRLH、TxPRLL 寄存器来指定。ETxPRL 是周期只读寄存器的低 8 位，高 4 位是 ETxPRH[bit7:4]。PWM 周期计算公式：

PWM 周期 =  $\{[TxPRLH, TxPRLL]+1\} * T * TMRx$  预分频值。

**注：**上式“T”为定时器时钟,其来源可通过(TxSELEN、TMRxCS、TxOSCEN、TxFOOSC)设置,详情见表 7-2-3

当 TMRx 等于{TxPRLH,TxPRLL}时, 下一次递增周期将发生以下三个事件:

- TMRx 被清零;
- PWM 引脚被置 1 (例外: 若 PWM 占空比 = 0%, 引脚不被置 1);
- PWM 占空比重载寄存器值锁存到占空比寄存器。

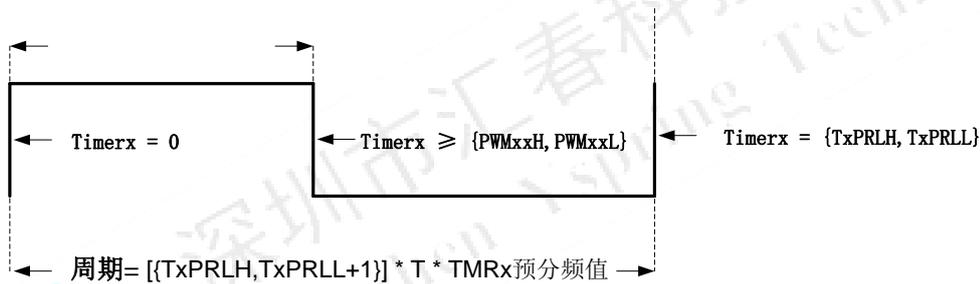
通过对以下几个寄存器写入 12 位值可设定 PWM 占空比: PRLXXH, PRLxxL 为占空比重载寄存器, PWMxxL、PWMxxH 为占空比从动寄存器 (只读)。脉冲宽度直到周期完成时才将占空比重载寄存器的值锁存到占空比从动寄存器, 这样可以防止程序改变占空比时造成脉冲宽度跳变。

脉冲宽度计算公式:

脉冲宽度 =  $\{[PWMxxH, PWMxxL]+1\} * T * TMRx$  预分频值。

占空比 =  $\{[PWMxxH, PWMxxL]+1\} / \{[TxPRLH, TxPRLL]+1\}$ 。

高脉宽 =  $\{[PWMxxH, PWMxxL]+1\} * T * TMRx$  预分频值



### 7.3.3 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。

$\{[T1PRLH, T1PRLL]\}$  为 4096 时产生 12 位的最大 PWM 分辨率。分辨率是 $\{[TxPRLH, TxPRLL]\}$ 的函数。

PWM 分辨率

$$\text{分辨率} = \frac{\text{Log}(\{[TxPRLH, TxPRLL] + 1\})}{\text{Log}[2]} \text{ 位}$$

表7-4-1: PWM频率和分辨率示例 (Fosc = 16MHz)

PWM 频率	0.976kHz	3.906kHz	7.8125kHz	15.625kHz	62.5kHz	40kHz
定时器预分频比 (1、2、4、8)	1	1	1	1	1	1
$\{[T1PRLH, T1PRLL]\}$ 的值	0x0F, 0xFF	0x03, 0xFF	0x01, 0xFF	0x00, 0xFF	0x00, 0x80	0x00, 0x64
最大分辨率 (位)	12	10	9	8	7	6

### 7.3.4 PWM 的中断

PWM可产生定时中断。当TIMER1/TIMER2完成一个PWM周期时，可产生一个PWM中断，PIFB1的PWMxIF中断标志位会置1。此时GIE与PEIE也为1，则会进入中断服务程序。

### 7.3.5 PWM 输出配置

以下是配置PWM输出的步骤示例：

- 1、端口配置：  
配置相应的PWM信号引脚为输出口
- 2、配置周期：  
配置TIMERx控制寄存器（TxSTA），设定TIMERx为定时器模式，并设定预分频比  
配置TxPRLH、TxPRLL（详见7.3.2 PWM的周期）
- 3、配置占空比：  
配置占空比重载寄存器PRLXXH，PRLXXL
- 4、使能PWM：  
设置PWM控制寄存器（PWMCON0），如果只要PWM
- 5、使能TIMERx  
置位TMRxON，此时将会使PRLXXH，PRLXXL载入PWMxxH，PWMxxL

### 7.3.6 PWM 死区输出配置

以下为配置PWM带死区互补输出的步骤示例：

- 1、端口配置：  
配置PWMA0、PWMA1引脚为输出口（PS：仅PWMA0、PWMA1可设置为互补输出，如需要全桥，可复制到PWMB0、PWMB1，再配置PWMB0、PWMB1是否反相输出）；
- 2、配置周期：  
配置TIMERx控制寄存器（TxSTA），设定TIMERx为定时器模式，并设定预分频比  
配置TxPRLH、TxPRLL（详见7.3.2 PWM的周期）
- 3、配置占空比：  
配置占空比重载寄存器PRLXXH，PRLXXL
- 4、使能PWM：  
设置PWM控制寄存器（PWMCON0），如果只要PWM
- 5、使能TIMERx  
置位TMRxON，此时将会使PRLXXH，PRLXXL载入PWMxxH，PWMxxL

## 7.4 PWM 模块寄存器说明

### 寄存器110H：PWM控制寄存器0（PWMCON0）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1
PWMB1ON	PWMB0ON	PWMA1ON	PWMA0ON	PWMB1EN	PWMB0EN	PWMA1EN	PWMA0EN
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      **PWMB1ON**: PWMB1信号输出允许位

- 1 = PWMB1允许输出  
0 = PWMB1禁止输出
- bit6 **PWMB0ON:** PWMB0信号输出允许位  
1 = PWMB0允许输出  
0 = PWMB0禁止输出
- bit5 **PWMA1ON:** PWMA1信号输出允许位  
1 = PWMA1允许输出  
0 = PWMA1禁止输出
- bit4 **PWMA0ON:** PWMA0信号输出允许位  
1 = PWMA0允许输出  
0 = PWMA0禁止输出
- bit3 **PWMB1EN:** PWMB1模块使能位  
1 = PWMB1 模块允许  
0 = PWMB1 模块禁止
- bit2 **PWMB0EN:** PWMB0模块使能位  
1 = PWMB0 模块允许  
0 = PWMB0 模块禁止
- bit1 **PWMA1EN:** PWMA1模块使能位  
1 = PWMA1模块允许  
0 = PWMA1 模块禁止
- bit0 **PWMA0EN:** PWMA0模块使能位  
1 = PWMA0模块允许  
0 = PWMA0 模块禁止

## 寄存器111H: PWM控制寄存器1 (PWMCON1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1
<b>PWMB1TR</b>	<b>PWMB0TR</b>	<b>PWMA1TR</b>	<b>PWMA0TR</b>	<b>PWMB1OL</b>	<b>PWMB0OL</b>	<b>PWMA1OL</b>	<b>PWMA0OL</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 **PWMB1TR:** PWMB1信号输出引脚选择  
1 = PWMB1信号在PA5引脚上输出  
0 = PWMB1信号在PC0引脚上输出
- bit6 **PWMB0TR:** PWMB0信号输出引脚选择  
1 = PWMB0信号在PA2引脚上输出  
0 = PWMB0信号在PC3引脚上输出
- bit5 **PWMA1TR:** PWMA1信号输出引脚选择  
1 = PWMA1信号在PA1引脚上输出  
0 = PWMA1信号在PC4引脚上输出
- bit4 **PWMA0TR:** PWMA0信号输出引脚选择  
1 = PWMA0信号在PA0引脚上输出  
0 = PWMA0信号在PC5引脚上输出
- bit3 **PWMB1OL:** PWMB1信号输出极性选择

1 = PWMB1 信号反相  
0 = PWMB1 信号不反相

bit2 **PWMB0OL:** PWMB0信号输出极性选择  
1 = PWMB0 信号反相  
0 = PWMB0 信号不反相

bit1 **PWMA1OL:** PWMA1信号输出极性选择  
1 = PWMA1 信号反相  
0 = PWMA1 信号不反相

bit0 **PWMA0OL:** PWMA0信号输出极性选择  
1 = PWMA0 信号反相  
0 = PWMA0 信号不反相

#### 寄存器112H: PWMA0占空比重载寄存器低字节 (PRLA0L)

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
<b>PRLA0L</b>							

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 **PRLA0L:** PWMA0占空比重载寄存器的低8位

#### 寄存器113H: PWMA1占空比重载寄存器低字节 (PRLA1L)

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
<b>PRLA1L</b>							

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 **PRLA1L:** PWMA1占空比重载寄存器的低8位

#### 寄存器114H: PWMB0占空比重载寄存器低字节 (PRLB0L)

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
<b>PRLB0L</b>							

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 **PRLB0L:** PWMB0占空比重载寄存器的低8位

#### 寄存器115H: PWMB1占空比重载寄存器低字节 (PRLB1L)

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
<b>PRLB1L</b>							

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            **PRLB1L:**      PWMB1占空比重载寄存器的低8位

**寄存器117H: PWMA0/1周期重载寄存器低字节 (T1PRL)**

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
<b>T1PRL</b>							

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            **T1PRL:**      PWMA0/1周期重载寄存器的低8位

**寄存器118H: PWMA0/1周期寄存器低字节(只读) (ET1PRL)**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
bit7							bit0
<b>ET1PRL</b>							

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            **ET1PRL:**      PWMA0/1周期寄存器的低8位(只读)

**寄存器119H: PWMB0/1周期重载寄存器低字节 (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
<b>T2PRL</b>							

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:**      PWMB0/1周期重载寄存器的低8位

**寄存器11AH: PWMB0/1周期寄存器低字节(只读) (ET2PRL)**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
bit7							bit0
<b>ET2PRL</b>							

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            **ET2PRL:**      PWMB0/1周期寄存器的低8位(只读)

**寄存器192H: PWMA0占空比较寄存器低字节 (PWMA0L)**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
bit7							bit0
<b>PWMA0L</b>							

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            **PWMA0L:**      PWMA0占空比较寄存器的低8位

**寄存器193H: PWMA1占空比较寄存器低字节 (PWMA1L)**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
bit7							bit0
<b>PWMA1L</b>							

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            **PWMA1L:**      PWMA1占空比较寄存器的低8位

**寄存器194H: PWMB0占空比较寄存器低字节 (PWMB0L)**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
bit7							bit0
<b>PWMB0L</b>							

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            **PWMB0L:**      PWMB0占空比较寄存器的低8位

**寄存器195H: PWMB1占空比较寄存器低字节 (PWMB1L)**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
bit7							bit0
<b>PWMB1L</b>							

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            **PWMB1L:**      PWMB1占空比较寄存器的低8位

**寄存器197H: PWMA0占空比较寄存器高4位与占空比重载寄存器高4位 (EPRA0H)**

R-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
bit7							bit0
<b>PWMA0H</b>				<b>PRLA0H</b>			

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            **PWMA0H:**      PWMA0占空比较寄存器的高4位

bit3-0            **EPRA0H:**      PWMA0占空比重载寄存器的高4位

**寄存器198H: PWMA1占空比较寄存器的高4位与占空比重载寄存器高4位 (EPRA1H)**

R-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
bit7							bit0
<b>PWMA1H</b>				<b>PRLA1H</b>			

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 **PWMA1H:** PWMA1占空比较寄存器的高4位

bit3-0 **EPR1H:** PWMA1占空比重载寄存器的高4位

**寄存器199H: PWMB0占空比较寄存器的高4位与占空比重载寄存器高4位 (EPRB0H)**

R-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
bit7							bit0
<b>PWMB0H</b>				<b>PRLB0H</b>			

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 **PWMB0H:** PWMB0占空比较寄存器的高4位

bit3-0 **EPRB0H:** PWMB0占空比重载寄存器的高4位

**寄存器19AH: PWMB1占空比较寄存器的高4位与占空比重载寄存器高4位 (EPRB1H)**

R-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
bit7							bit0
<b>PWMB1H</b>				<b>PRLB1H</b>			

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 **PWMB1H:** PWMB1占空比较寄存器的高4位

bit3-0 **EPRB1H:** PWMB1占空比重载寄存器的高4位

**寄存器19BH: PWMA0/1周期寄存器的高4位 (只读) 与周期重载寄存器高4位 (T1PRLH)**

R-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
bit7							bit0
<b>ET1PRH</b>				<b>T1PRLH</b>			

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 **ET1PRH:** ET1PRH周期寄存器的高4位 (只读)

bit3-0 **T1PRLH:** T1PRLH周期重载寄存器的高4位

**寄存器19CH: PWMB0/1周期寄存器的高4位 (只读) 与周期重载寄存器高4位 (T2PRLH)**

R-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
bit7							bit0
<b>ET2PRH</b>				<b>T2PRLH</b>			

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 **ET2PRH:** ET2PRH周期寄存器的高4位 (只读)

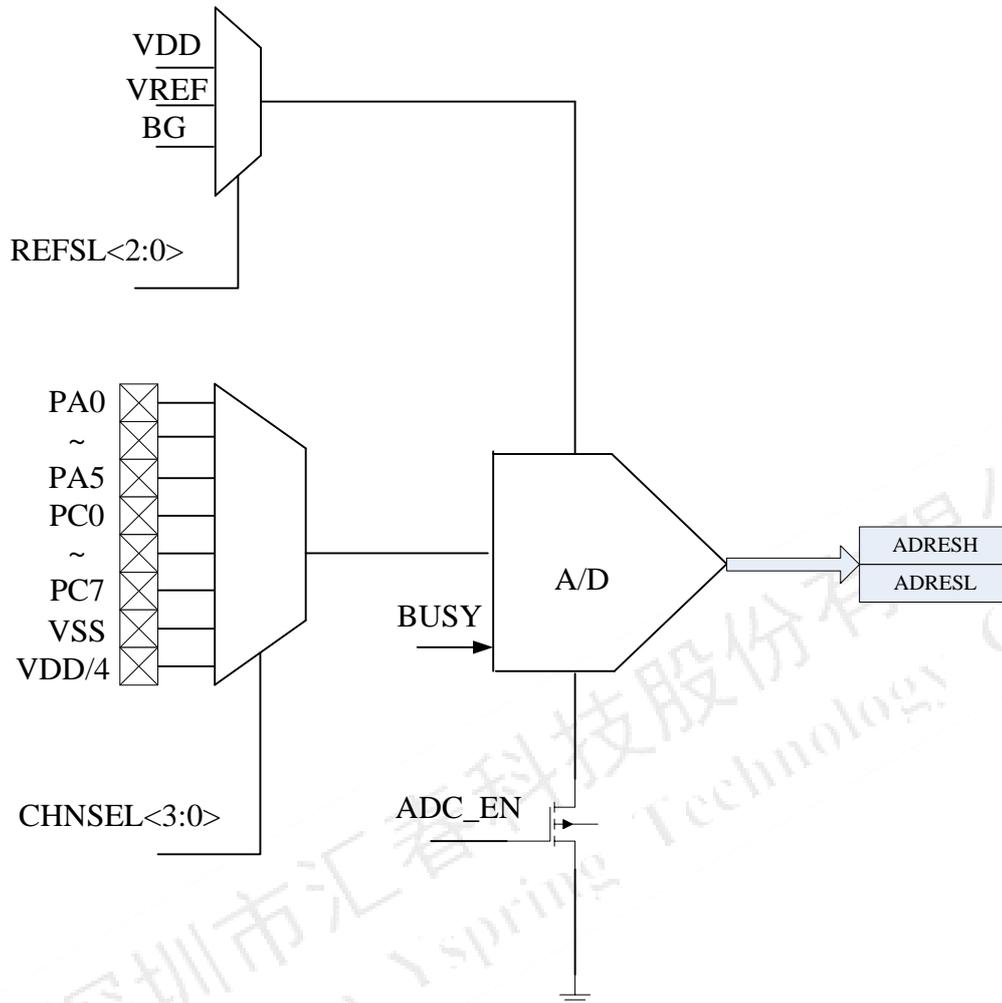
bit3-0 **T2PRLH:** T2PRLH周期重载寄存器的高4位

## 8 模拟数字转换器（ADC）模块

模数转换器（Analog-to-digital Converter, ADC）可将模拟输入信号转换为相应的 12 位二进制表征值。该系列器件采用多个模拟输入复用到一个采样保持电路。采样保持电路的输出与转换器的输入相连接。转换器通过逐次逼近法产生 12 位二进制值，并将转换结果保存在 ADC 结果寄存器（ADRESL:ADRESH）中。

ADC 参考电压可用软件选择为 VDD 或施加在外部参考引脚上的电压。

## 8.1 ADC 框图



## 8.2 转换启动方式

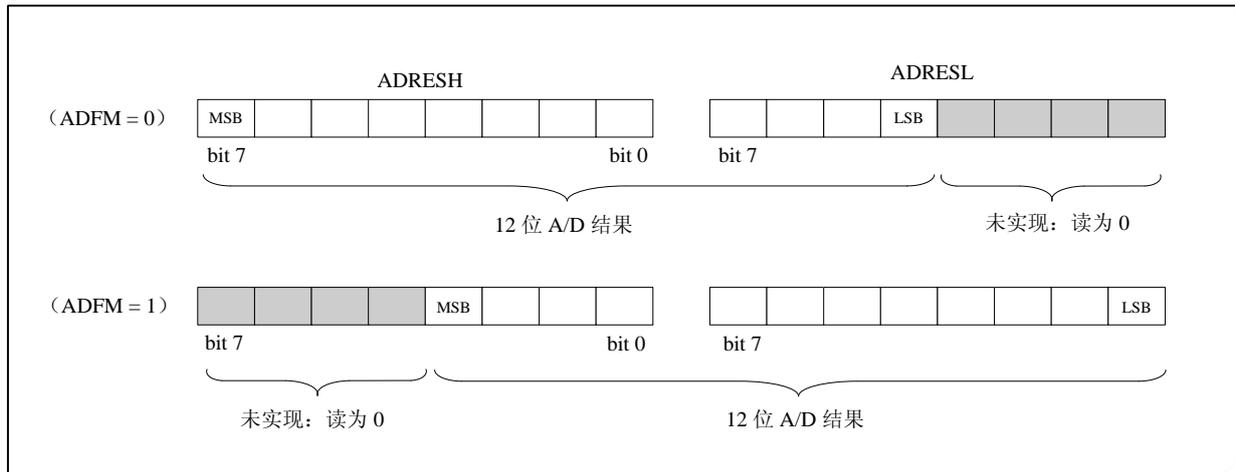
有 3 种 A/D 转换启动方式，由 [ADC0CN](#) 中的 ADC0 转换启动方式位 CM[1:0] 的状态决定采用哪一种方式。转换的触发源有：

- GO/DONE 写 1
- Timer0 溢出
- Timer1 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。

### 8.2.1 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 [ADC0CN](#) 的 ADFM 位来决定。



## 8.2.2 端口配置

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

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

## 8.2.3 ADC 参考选择寄存器

[ADS0 寄存器](#) 的 CHNSEL 位决定将哪个通道连接到采样保持电路。正参考电压可以是：

- 选电压 VDD 做为 ADC 的参考。
- 选内部 BANDGAP 的输出电压做为 ADC 的参考，这个电压可以通过配置寄存器选择 1.024V, 2.048V, 4.096V。

负参考电压始终连接到参考地。

注：选择外部  $V_{REF}$  作为 ADC 参考时，ADC 采样结果误差较大，不建议采用。

## 8.3 ADC 寄存器定义

寄存器1CH: A/D控制寄存器ADC0CN

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	<b>ADFM</b>	<b>SC1</b>	<b>SC0</b>	<b>ADC_EN</b>	<b>GO/DONE</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 未实现：读为0

bit6 **ADFM**: ADC输出对齐标志位。

0 = 左对齐，低位补零

1 = 右对齐，高位补零

- bit5-4 **SC<1:0>**: ADC时钟预分频选择位  
 00 = sys\_clk 8分频  
 01 = sys\_clk 16分频  
 10 = sys\_clk 32分频  
 11 = sys\_clk 64分频
- bit3 **ADC\_EN**: ADC使能位  
 0 = ADC禁止  
 1 = ADC使能
- bit2 **GO/DONE (BUSY)**: ADC启动（忙）标志位  
 0 = 不忙  
 1 = 转换（忙）
- bit1-0 **CM<1:0>**: ADC0启动转换模式选择, 指定ADC启动转换源  
 0x = GO/DONE写1。  
 10 = Timer0溢出  
 11 = Timer1溢出

**寄存器1FH: ADC结果高位寄存器 (ADRESH)**

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ADRESH寄存器清零

**寄存器1EH: ADC结果低位寄存器 (ADRESL)**

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ADRESL寄存器清零

**寄存器1DH: A/D通道选择寄存器ADS0**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1
<b>CHNSEL3</b>	<b>CHNSEL2</b>	<b>CHNSEL1</b>	<b>CHNSEL0</b>	<b>REFSEL1</b>	<b>REFSEL0</b>	<b>VREFSEL</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            **CHNSEL<3:0>**: ADC模拟通道选择位  
 0000 = PA0    ADC 输入通道 0  
 0001 = PA1    ADC 输入通道 1  
 0010 = PA2    ADC 输入通道 2  
 0011 = PA4    ADC 输入通道 3  
 0100 = PA5    ADC 输入通道 4  
 0101 = PC0    ADC 输入通道 5

0110 = PC1 ADC 输入通道 6  
 0111 = PC2 ADC 输入通道 7  
 1000 = PC3 ADC 输入通道 8  
 1001 = PC4 ADC 输入通道 9  
 1010 = PC5 ADC 输入通道 10  
 1011 = PC6 ADC 输入通道 11  
 1100 = PC7 ADC 输入通道 12  
 1101 = ADC 输入电压为 VSS  
 1110 = ADC 输入电压为 VDD/4  
 1111 = ADC 输入通道关闭

bit3-2 **REFSEL<1:0>**: ADC 电压参考选择  
 0X = ADC 电压参考是内部电压参考。(默认值)  
 10 = ADC 电压参考是 VDD pin。  
 11 = ADC 电压参考是 PC1/VREF pin。

(注意: 睡眠前把 ADC 参考电压设为内部参考, 否则会漏电)

bit1-0 **VREFSEL**: 内部电压参考选择 (1.024V/2.048V/4.096)  
 00 = 1.024V  
 01 = 2.048V  
 10 禁止  
 11 = 4.096V

表 8-4: 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 LVR 时的值	所有其他复位值
ADC0CN	—	ADFM	SC1	SC0	ADCEN	GO/ DONE	CM1	CM0	0000 0000	xxxx xxxx
ADS0	CHNSEL3	CHNSEL2	CHNSEL1	CHNSEL0	REFSEL1	REFSEL0	VREFSEL		1111 0111	xxxx xxxx
ADRESH	A/D 结果寄存器高字节								xxxx xxxx	xxxx xxxx
ADRESL	A/D 结果寄存器低字节								xxxx xxxx	xxxx xxxx
INTS	GIE	PEIE	TIS	INS	PAIE	TIF	INTF	PAIF	0000 0000	xxxx xxxx
PIFB0	INT1E	ADCIE	OSFIE	TMR1IE	INT1IF	ADCIF	OSFIF	TMRIF	0000 0000	xxxx xxxx
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	xxxx xxxx	xxxx xxxx
PORTA	—	—	PA5	PA4	PA3	PA2	PA1	PA0	xxxx xxxx	xxxx xxxx
CPIOA	—	—	CPIOA5	CPIOA4	CPIOA3	CPIOA2	CPIOA1	CPIOA0	--11 1111	xxxx xxxx
CPIOC	CPIOC7	CPIOC6	CPIOC5	CPIOC4	CPIOC3	CPIOC2	CPIOC1	CPIOC0	1111 1111	xxxx xxxx
ADINA	—	—	AN4	AN3	—	AN2	AAN1	AN0	--11 1111	xxxx xxxx
ADINC	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	1111 1111	xxxx xxxx

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

## 9 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。

## 9.1 Config Option

Config Option 用于给客户提供配置选项。

### 寄存器 801H: 配置字寄存器高8位 (Config1 Option)

—	RD_CTRL	LVDS1	LVDS0	FCMEN	IESO	LVREN1	LVREN0
bit15							bit8

### 寄存器 800H: 配置字寄存器低8位 (Config0 Option)

—	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

00: 2.0V

01: 2.5V

10: 3.0V

11: 4.0V

bit11 **FCMEN**: 系统时钟失效监测使能位

1: 使能系统时钟失效监测, 仅在系统时钟为外部晶振模式才起作用

0: 禁止系统时钟失效监测

bit10 **IESO**: 内、外时钟切换使能位, 二速启动模式使能,

1: 使能二速启动模式, 仅在系统时钟为外部晶振模式才起作用

0: 禁止二速启动模式

bit9-8 **LVREN <1:0>**: 低电压复位选择<sup>(1)</sup>

11: 低电压复位使能

10: MCU工作时, 低电压复位开启; MCU睡眠时, 低电压复位关闭

01: 低电压复位由PCON之SLVREN位决定

00: 禁止低电压复位

bit7 未实现: 保留

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/MCLR引脚功能选择位<sup>(4)</sup>

1: PA3/MCLR脚执行MCLR功能, 是复位脚

0: PA3/MCLR脚执行PA3功能, 是数字输入、(开漏) 输出引脚

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: 在INTOSC或RC模式下当MCLR有效时, 内部时钟振荡器被禁止。

深圳市汇春科技股份有限公司  
 Shenzhen Yspring Technology Co., Ltd.

## 10 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→WT	/TF, /PF
SLEEP	进入睡眠模式	0→WT, stop OSC	/TF, /PF
TMODE	将 W 寄存器值装入选项寄存器 (81H)	W→81H	无
CPIO R	设置端口方向寄存器 (1 输入, 0 输出)	W→CPIO r	无

STWR R	将 W 寄存器内容送 R 寄存器	W→R	无
LDR R, t	读寄存器 R, 结果保存在 R(t=1)或者 W(t=0)	R→t	Z
LDW I I	立即数送 W 寄存器	I→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(0~3) R(4~7)]→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→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→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→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→t (R+W+1→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→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→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 ∩ W→t	Z
ANDW I 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 ∩ W→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 ∪ W→t	Z
IORW I 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 ∪ W→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 ⊕ W→t	Z
XORW I 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 ⊕ W→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→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)→R(n-1), C→R(7), R(0)→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)→r(n+1), C→R(0), R(7)→C	C
CLRW	W 寄存器清 0	0→W	Z
CLRR R	R 寄存器清 0	0→R	Z
BCR R, b	R 寄存器的第 b 位清 0	0→R(b)	无
BSR R, b	R 寄存器的第 b 位置 1	1→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→PC, PC+1→Stack	无
LJUMP N	在整个 2K 区域内的跳转指令	N→PC	无
RTIW I	带立即数从子程序返回	Stack→PC, I→W	无
ADDW I I	W 寄存器与立即数 I 相加, 结果保存到 W 中	PC+1→PC, W+I→W	C, HC, Z
SUBW I 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,1→GIS	无
RET	从子程序返回	Stack→PC	无

深圳市汇春科技股份有限公司  
Shenzhen Yspring Technology Co., Ltd.

## 11 电气特性

### 11.1 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V <sub>SS</sub> 的电压.....	-0.3V 至+9.5V
所有其他引脚相对于V <sub>SS</sub> 的电压.....	-0.3V 至(V <sub>DD</sub> +0.3V)
总功耗 <sup>(1)</sup> .....	600mW
流出V <sub>SS</sub> 引脚的最大电流.....	95mA
流入V <sub>DD</sub> 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> >V <sub>DD</sub> ).....	±20mA
输出钳位电流, I <sub>OK</sub> (V <sub>O</sub> <0 或 V <sub>O</sub> >V <sub>DD</sub> ).....	±20mA
任一I/O引脚的最大输出灌电流.....	25mA
任一I/O引脚的最大输出拉电流.....	25mA
PORTA 和 PORTC(联合)最大灌电流.....	90mA
PORTA 和 PORTC(联合)最大拉电流.....	90mA

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

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。

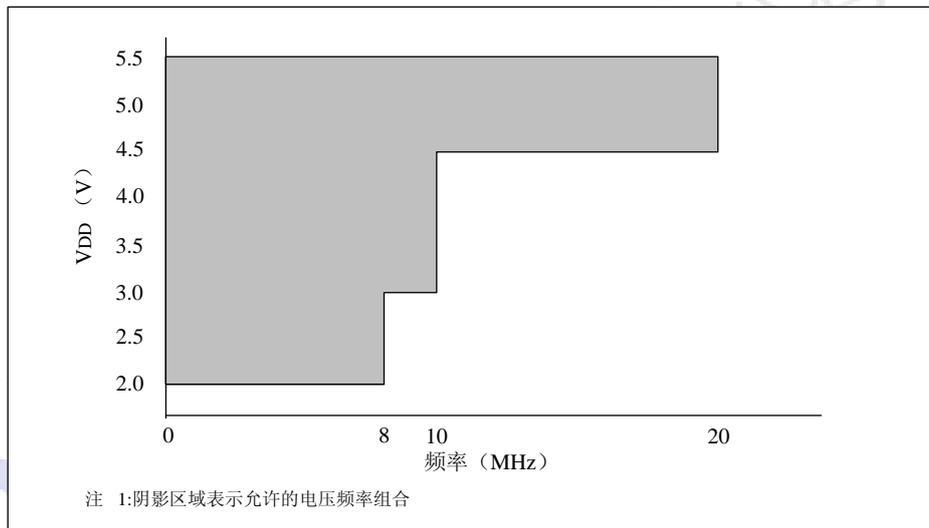


图 11-1-1: MDT10F273 外部振荡电压—频率图,  $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$

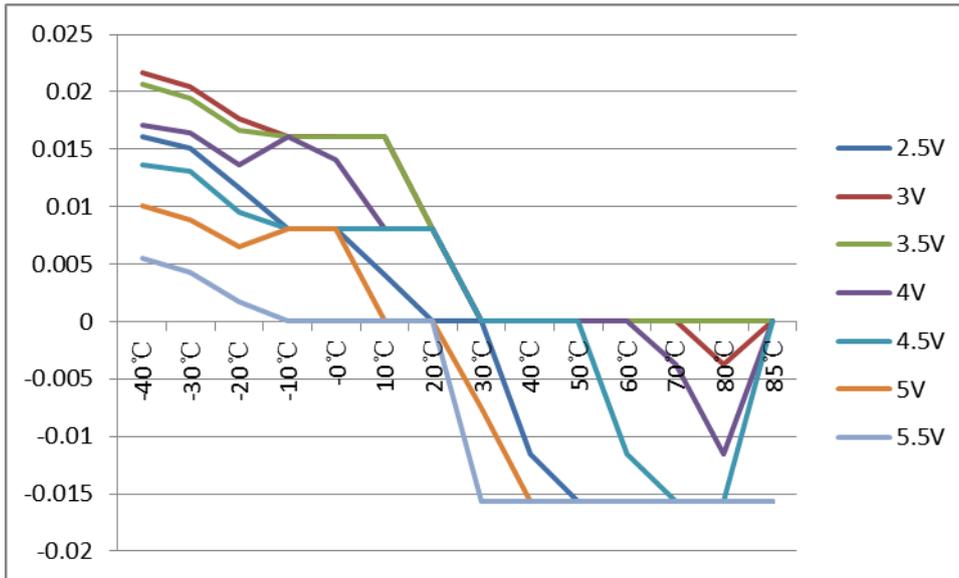


图 11-1-2: 器件 VDD 和温度范围内 HFINTOSC 的频率准确性

深圳市汇春科技股份有限公司  
Shenzhen Yspring Technology Co., Ltd.

## 11.2 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	2.2		5.5	V			
VDR	RAM 数据保持电压 <sup>(2)</sup>	—	0.5*	—	V	器件处于休眠模式		
VPOR	Vdd 起始电压确保能够产生上电复位信号	—	Vss	—	V			
SVDD	Vdd 上升速率确保能够产生上电复位信号	0.05*	—	—	V/ms			
IDD	工作电流 <sup>(3)</sup>	—	400	—	uA	3.3V	1M	WDT En PED Dis IRC
			410				4M	
			420				8M	
			500				1M	
			800				4M	
			1100				8M	
IPD	掉电流 <sup>(4)</sup>	—	1	—	uA	WDT Disable VDD=2.5V		
$\Delta$ IWDT	WDT 电流 <sup>(4)</sup>	—	0.5	—	uA	VDD=5V		
VIL	输入低电压	VSS	—	1.1 (不含 PA3)	V	3V SCHMITT		
		VSS	—	1.1 (PA3)				
		VSS	—	1.5 (不含 PA3)		5V SCHMITT		
		VSS	—	1.7 (PA3)				
VIH	输入高电压	1.6 (不含 PA3)	—	VDD	V	3V SCHMITT		
		1.2 (PA3)	—	VDD				
		3.1 (不含 PA3)	—	VDD		5V SCHMITT		
		1.9 (PA3)	—	VDD				
IOL	输出灌电流	—	15	—	mA	VOL=0.1VDD	3V	
		—		—				
		—	33	—		VOL=0.1VDD	5V	
		—		—				
IOH	输出拉电流	—	7	—	mA	VOH=0.9VDD	3V	
		—		—				
		—	16	—		VOH=0.9VDD	5V	
		—		—				
VPED	低电压复位电压	2.0 -20%	2.0	2.0 +20%	V	—		
		2.5-20%	2.5	2.5 +20%				
		3.0-20%	3.0	3.0 +20%				
		4.0-20%	4.0	4.0+20%				
Rpu	上拉电阻	—	23	—	K	3V		
		—	24	—		5V		
Rd	下拉电阻	—	23	—	K	3V		
		—	23	—		5V		

注：“—”表示没有，或待定。

- (1) 典型栏中数据均为 25℃ 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。

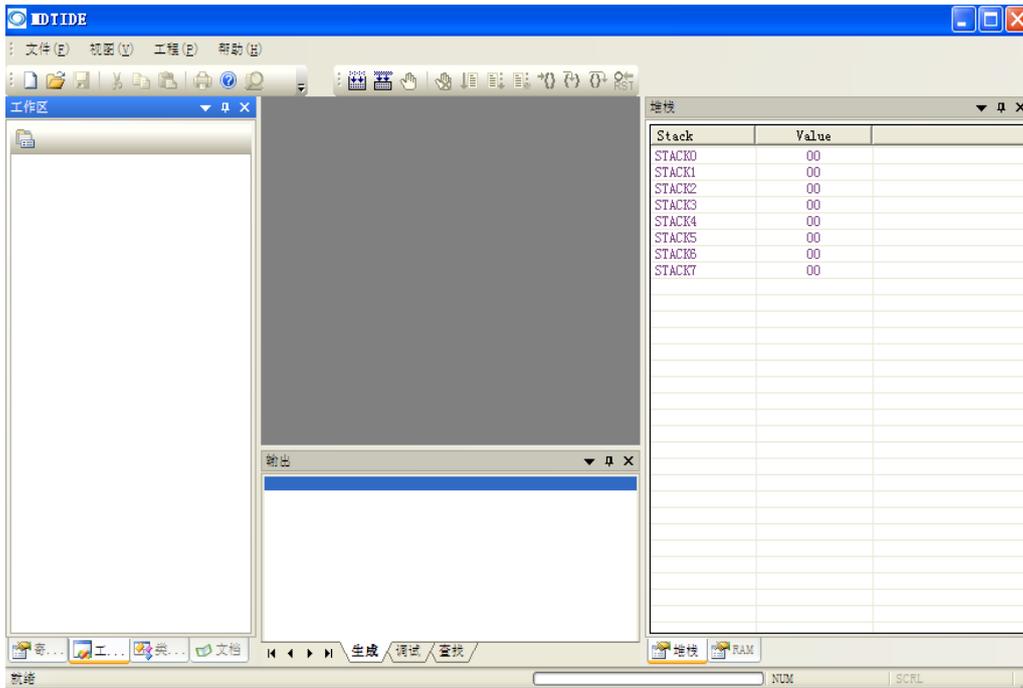
### 11.3 交流电气特性

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
$F_{\text{OSC}}$	系统时钟	0	8M	20M	Hz	5V
$F_{\text{HIRC}}$	内部高速时钟	—	4M	16M	Hz	5V
$F_{\text{LIRC}}$	内部低速时钟	—	31K	—	Hz	5V
$T_{\text{INT}}$	中断脉冲		2	3	Tins	指令周期
$T_{\text{SST}}$	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， $F_{\text{sys}}$ 在 SLEEP 模式下关闭)		1024		$T_{\text{sys}}$	5V
						5V
	系统启动时间 (由 SLEEP 模式唤醒，主要在 SLEEP 模式下开启)		1024		$T_{\text{sys}}$	5V
$T_{\text{RSTD}}$	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

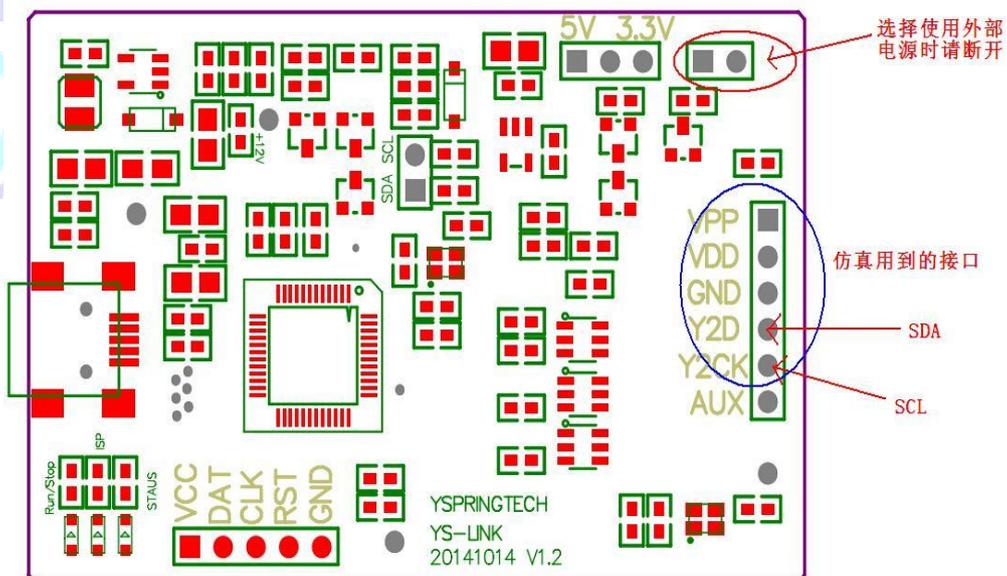
## 12 开发支持

### 12.1 仿真信息

#### 12.1.1 软件：MDT-IDE



#### 12.1.2 硬件：YS-Link



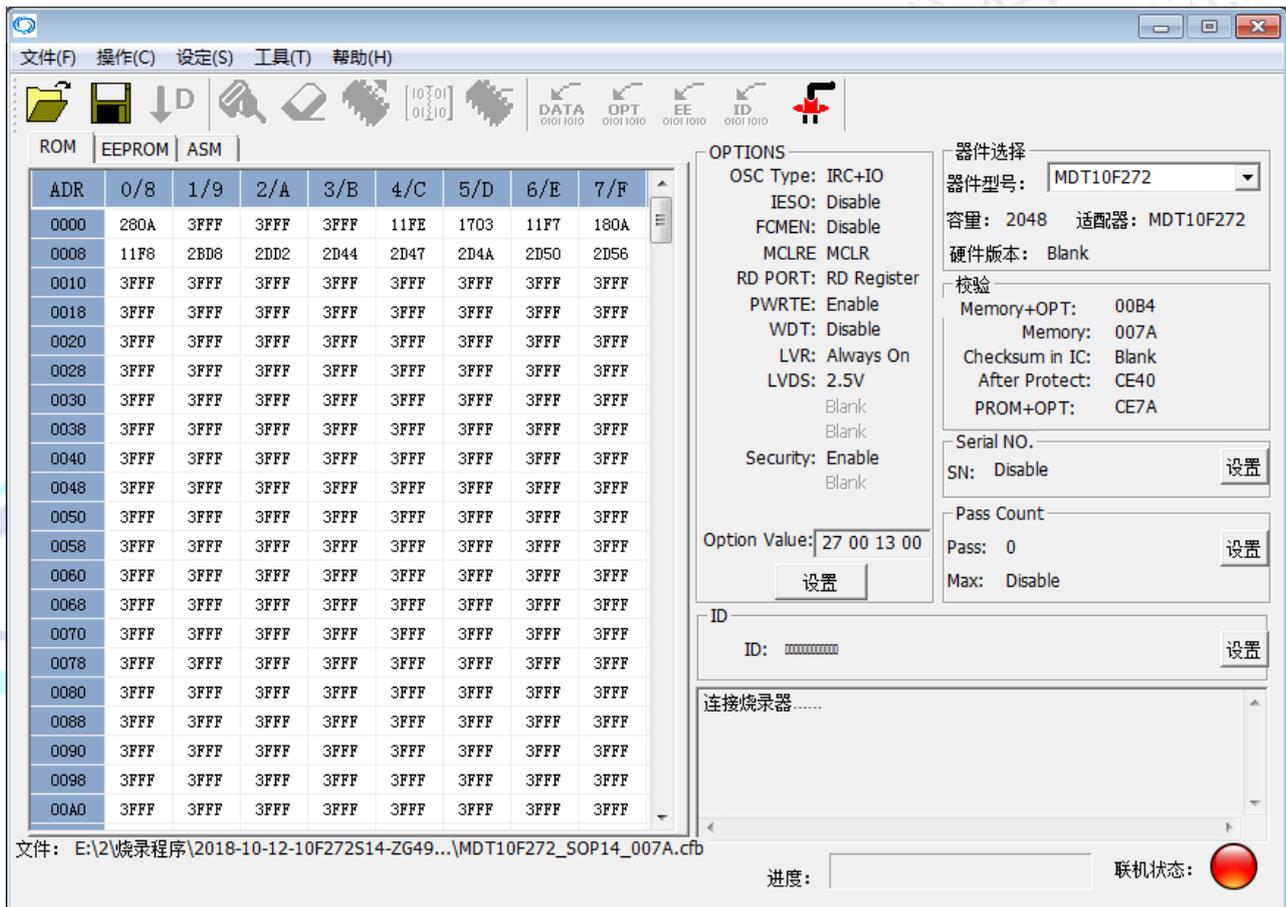
### 12.1.3 接口:

MDT10F273 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：

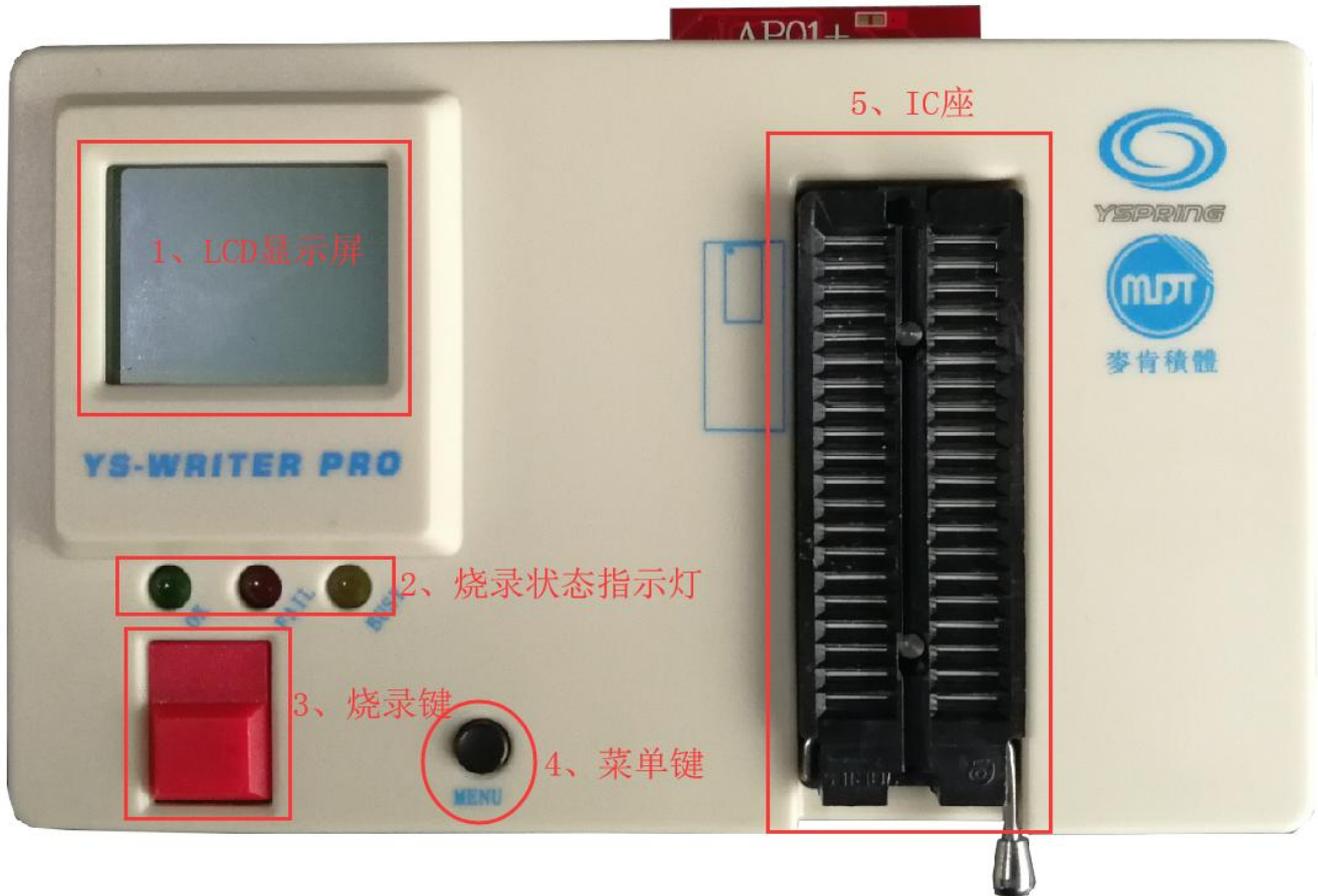
MDT10F273	YS-Link
VDD	VDD
GND	GND
PA3	VPP
PA0	SCK
PA1	SDA

## 12.2 烧录信息

### 12.2.1 烧录软件：YSpringPro



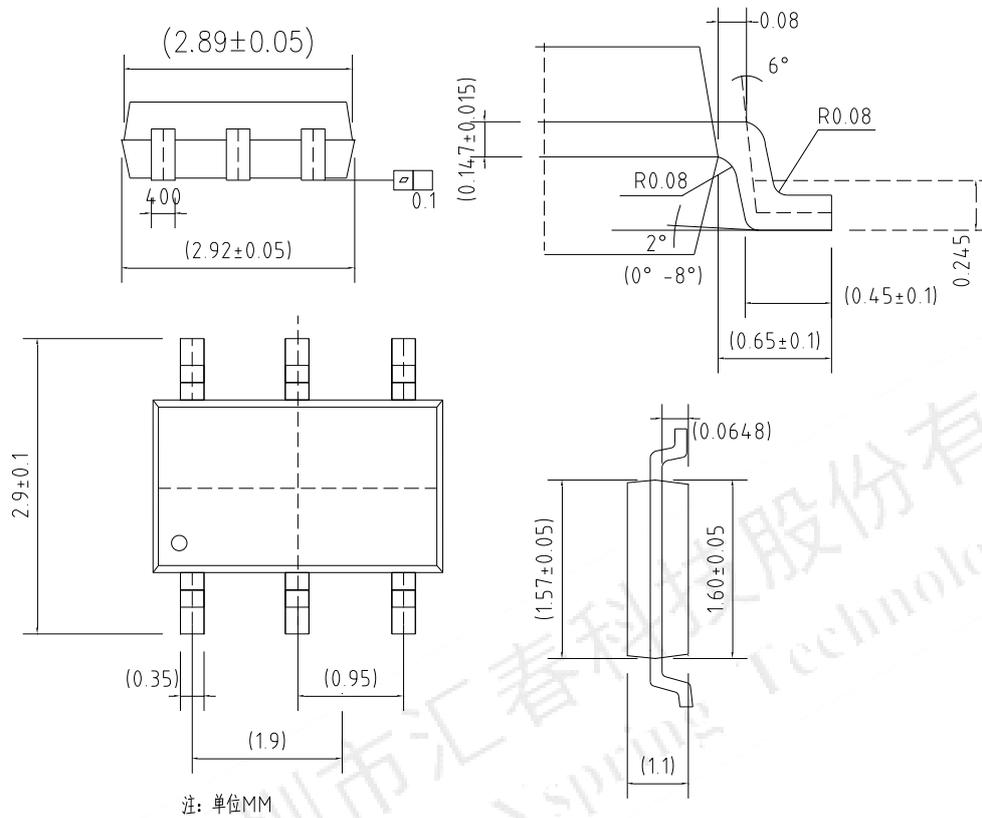
## 12.2.2 烧录器：YS-Writer



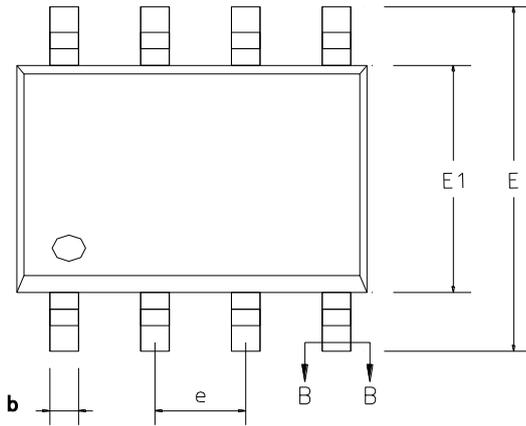
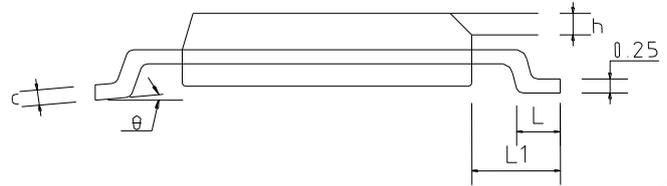
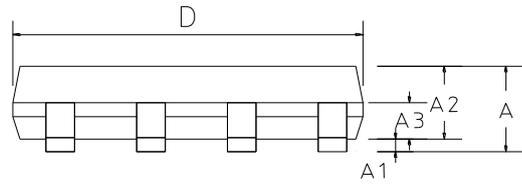
烧录器使用引脚：VDD、VSS、VPP、PA0、PA1、PA2 (Busy 信号)

## 13 封装信息

### 13.1 SOT23-6

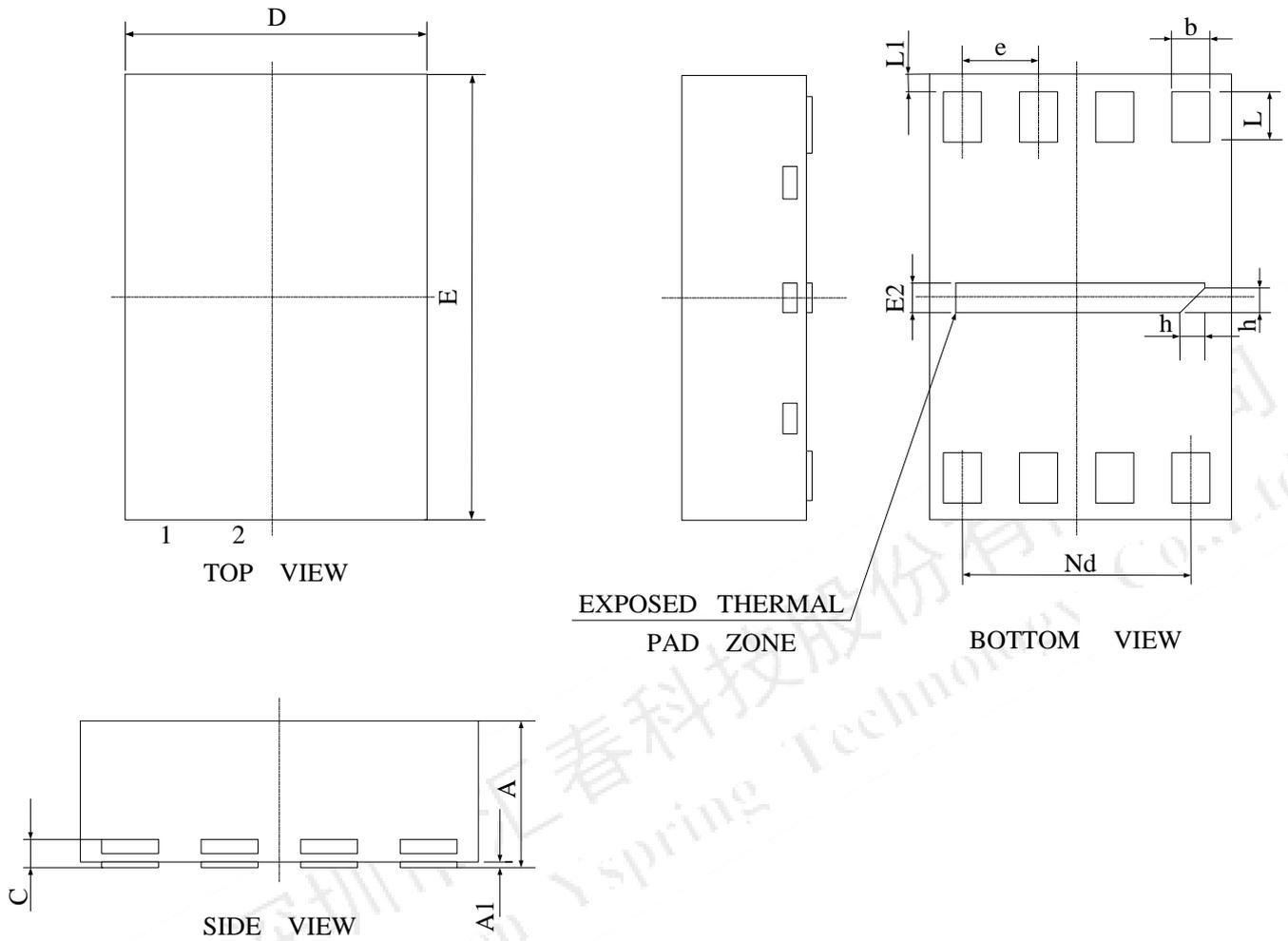


## 13.2 SOP 8 PIN



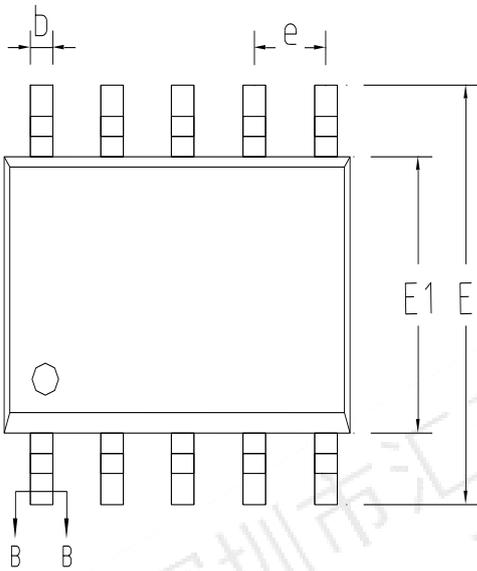
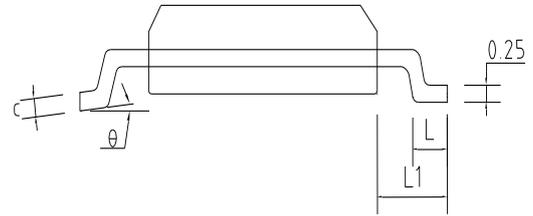
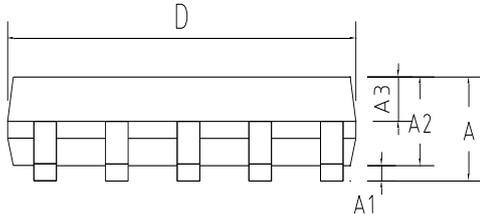
SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	—	—	1.75	E1	3.70	3.90	4.10
A1	0.10	—	0.225	e	1.27BSC		
A2	1.30	1.40	1.50	h	0.25	—	0.50
A3	0.60	0.65	0.70	L	0.50	—	0.80
D	4.70	4.90	5.10	L1	1.05 BSC		
E	5.80	6.00	6.20	$\theta$	0	—	8°
L/F 载体 尺寸(mil)	80*80	90*90	95*130	—			

## 13.2 DFN 8 PIN



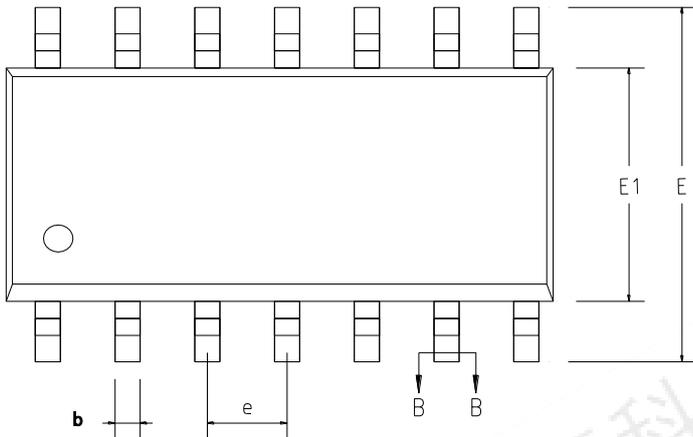
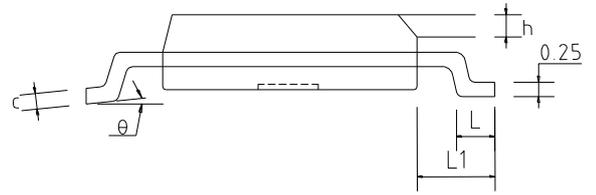
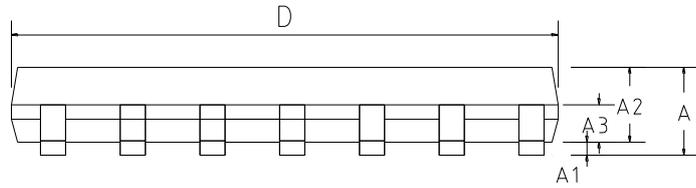
SYMBOLS	MIN	NOM	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
3.70	3.90	4.10	0.60	e	0.50BSC		
A1	0	0.02	0.05	Nd	1.50BSC		
b	0.18	0.25	0.30	E	2.90	3.00	3.10
c	0.10	0.15	0.20	E2	0.10	0.20	0.30
D	1.90	2.00	2.10	L	0.30	0.35	0.40
D2	1.50	1.60	1.70	L1	0.05	0.10	0.15
L/F 载体 尺寸(mil)				H	0.05	0.15	0.25

## 13.3 MSOP 10 PIN



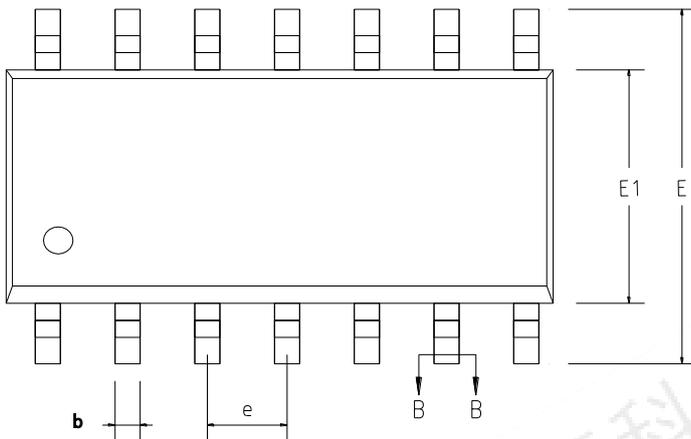
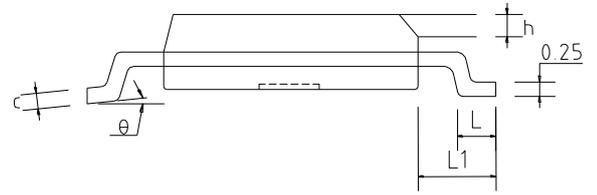
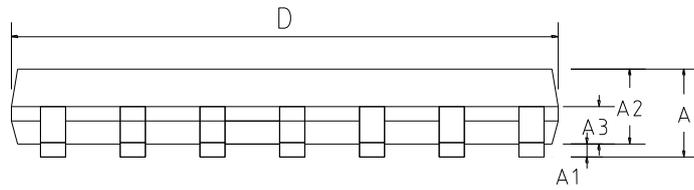
SYMBOLS	MIN	NOM	MAX	SYMBOLS	MIN	NOM	MAX
	(mm)				(mm)		
A	—	—	1.10	E1	2.90	3.00	3.10
A1	0.05	—	0.15	e	0.50BSC		
A2	0.75	0.85	0.95	c	0.15	—	0.19
A3	0.30	0.35	0.40	L	0.40	—	0.70
D	2.90	3.00	3.10	L1	0.95REF		
E	4.70	4.90	5.10	$\theta$	0	—	8°
L/F 载体尺寸 (mil)				b	0.18	—	0.26

## 13.4 SOP 14 PIN



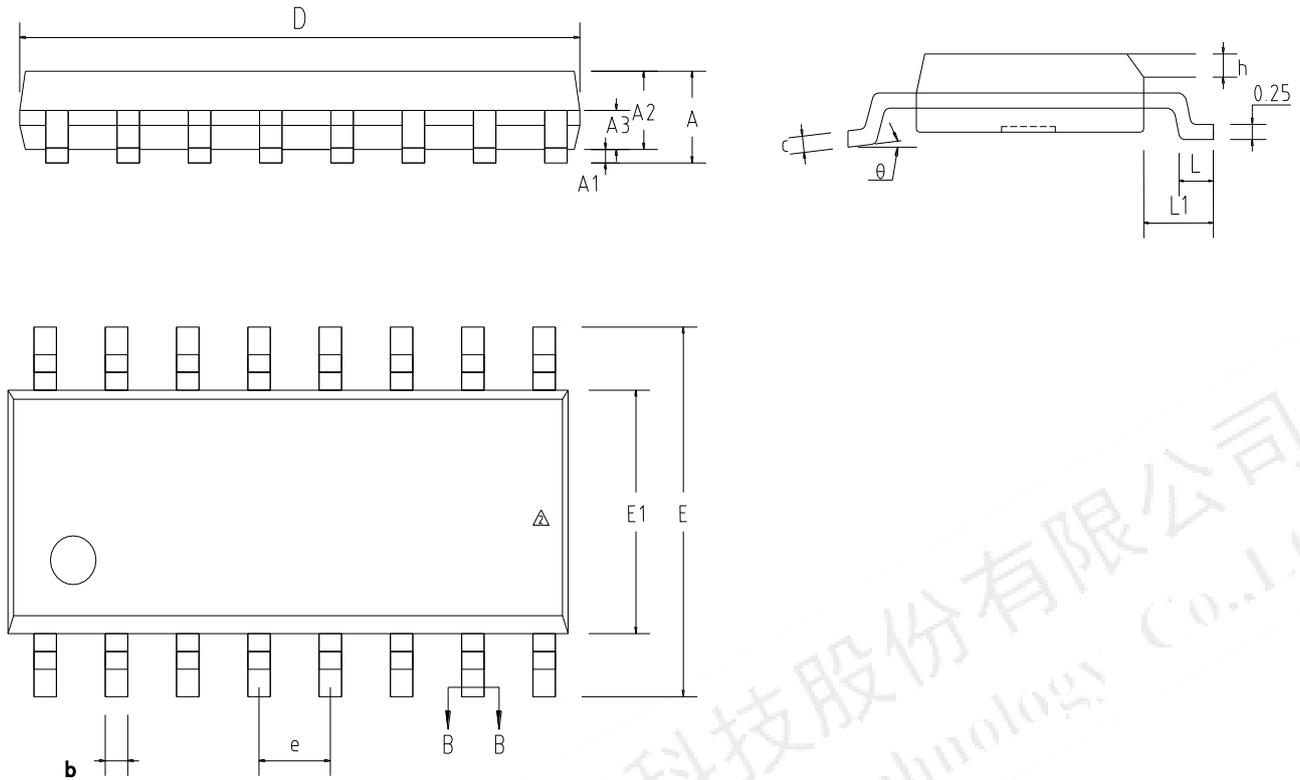
SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	—	—	1.75	E1	3.70	3.90	4.10
A1	0.10	—	0.225	e	1.27BSC		
A2	1.30	1.40	1.50	h	0.25	—	0.50
A3	0.60	0.65	0.70	L	0.50	—	0.80
D	8.45	8.65	8.85	L1	1.05 BSC		
E	5.80	6.00	6.20	θ	0	—	8°
L/F 载体 尺寸(mil)	70*70 90*110	98*150	100.4*210	—			

## 13.5 TSSOP 14 PIN



SYMBOLS	MIN	NOM	MAX	SYMBOLS	MIN	NOM	MAX
	(mm)				(mm)		
A	—	—	1.20	E1	4.30	4.40	4.50
A1	0.05	—	0.15	e	0.65BSC		
A2	0.90	1.00	1.05	h	0.25	—	0.50
A3	0.39	0.44	0.49	L	0.45	0.60	0.75
b	0.20	—	0.28	L1	1.00BSC		
D	4.90	5.00	5.10	θ	0	—	8°
E	6.20	6.40	6.60	L/F 载体 尺寸(mil)	70*70 90*110	98*150	100.4*210

## 13.6 SOP16 PIN



SYMBOLS	MIN	NOR	MAX	SYMBOLS	MIN	NOR	MAX
	(mm)				(mm)		
A	—	—	1.75	E1	3.70	3.90	4.10
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	9.70	9.90	10.10	L1	1.05 BSC		
E	5.80	6.00	6.20	θ	0	—	8°

## 14 汇春知识产权政策

### 14.1 专利权

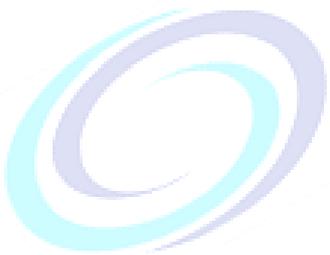
汇春公司在全球各地区已核准和申请中之专利权至少有 16 件以上,享有绝对之合法权益。与汇春公司 MCU 或其它产品有关的专利权并未被同意授权使用,任何经由不当手段侵害汇春公司专利权之公司、组织或个人,汇春将采取一切可能的法律行动,遏止侵权者不当的侵权行为,并追讨汇春公司因侵权行为所受之损失、或侵权者所得之不法利益。

### 14.2 著作权

Copyright 2014 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;>



深圳市汇春科技股份有限公司  
Shenzhen Yspring Technology Co., Ltd.