PIC10(L)F320/322

6/8 引脚 8 位闪存单片机

高性能 RISC CPU：

- 只需要学习 35 条指令：
  - 除了跳转指令以外，所有的指令都是单周期指令
- 工作速度：
  - DC – 16 MHz 时钟输入
  - DC – 250 ns 指令周期
- 最多 512 字的闪存程序存储器
- 64 字节数据存储器
- 8 级深硬件堆栈
- 中断功能
- 处理器可对程序存储器进行自写 / 读访问
- 引脚布局与其他 6 引脚 PIC10FXXX 单片机兼容

单片机的特性：

- 低功耗 16 MHz 内部振荡器：
  - 软件可选频率范围是 31 kHz 至 16 MHz
  - 出厂时校准为 ± 1% （典型值）
- 宽工作电压范围：
  - 1.8V 至 3.6V （PIC10LF320/322）
  - 2.3V 至 5.5V （PIC10F320/322）
- 上电复位 （Power-On Reset， POR）
- 上电延时定时器 （Power-up Timer， PWRT）
- 欠压复位 （Brown-Out Reset， BOR）
- 超低功耗休眠稳压器
- 扩展型看门狗定时器 （Watchdog Timer， WDT）
- 可编程代码保护
- 节能的休眠模式
- 可选振荡器选项 （EC 模式或内部振荡器）
- 通过两个引脚进行在线串行编程 （In-Circuit Serial Programming™， ICSP™）
- 在线调试器支持
- 固定参考电压（Fixed Voltage Reference，FVR），具有 1.024V、2.048V 和 4.096V（仅限 F 型器件）输出电平
- 集成温度指示器
- 闪存数据的保持时间为 40 年

低功耗特性 （PIC10LF320/322）：

- 待机电流：
  - 1.8V 时为 20 nA （典型值）
- 工作电流：
  - 1 MHz、 1.8V 时为 25 μA （典型值）
- 看门狗定时器电流：
  - 1.8V 时为 500 nA （典型值）

外设特性：

- 4 个 I/O 引脚：
  - 1 个仅用作输入的引脚
  - 高拉 / 灌电流可直接驱动 LED
  - 独立的可选弱上拉
  - 电平变化中断
- Timer0：带 8 位可编程预分频器的 8 位定时器 / 计数器
- Timer2：带 8 位周期寄存器、预分频器和后分频器的 8 位定时器 / 计数器
- 两个 PWM 模块：
  - 10 位 PWM，最高频率为 16 kHz
  - 结合为两相单路输出
- A/D 转换器：
  - 8 位分辨率和 3 路通道
- 可配置逻辑单元（Configurable Logic Cell，CLC）：
  - 8 个可选输入信号源
  - 每个模块有 2 个输入
  - 软件可选择的逻辑功能包括：AND/OR/XOR/D 触发器 /D 锁存器 /SR/JK
  - 外部或内部输入 / 输出
  - 可在休眠期间工作
- 数控振荡器 （Numerically Controlled Oscillator，NCO）：
  - 20 位累加器
  - 16 位递增值
  - 线性频率控制
  - 高速时钟输入
  - 可选输出模式
    - 固定占空比 （Fixed Duty Cycle， FDC）
    - 脉冲频率 （Pulse Frequency ， PF）模式
- 互补波形发生器 （Complementary Waveform Generator， CWG）：
  - 可选下降沿和上升沿死区控制
  - 极性控制
  - 2 个自动关闭源
  - 多个输入源：PWM、 CLC 和 NCO

© 2011-2014 Microchip Technology Inc.

PIC10(L)F320/322 系列型号

器件	数据手册索引	闪存程序存储器（字）	数据 SRAM（字节）	I/O(2)	8 位 ADC（通道）	定时器（8 位）	PWM	互补波形发生器（CWG）	可配置逻辑单元（CLC）	固定参考电压（FVR）	数控振荡器（NCO）	调试(1)	XLP
PIC10(L)F320	(1)	256	64	4	3	2	2	1	1	1	1	H	无
PIC10(L)F322	(1)	512	64	4	3	2	2	1	1	1	1	H	无

注 **1：** I —— 在芯片上集成调试功能； H —— 需要调试头进行调试； E —— 需要仿真头进行仿真。
2： 其中一个引脚仅用作输入引脚。

数据手册索引：

1： DS40001585B_CN PIC10(L)F320/322 数据手册， 6/8 引脚高性能闪存单片机。

注： 关于其他可用的小封装和标识信息，请访问**http://www.microchip.com/packaging**或联系当地的销售办事处。

© 2011-2014 Microchip Technology Inc.

图 1： PIC10(L)F320/322 的 6 引脚图

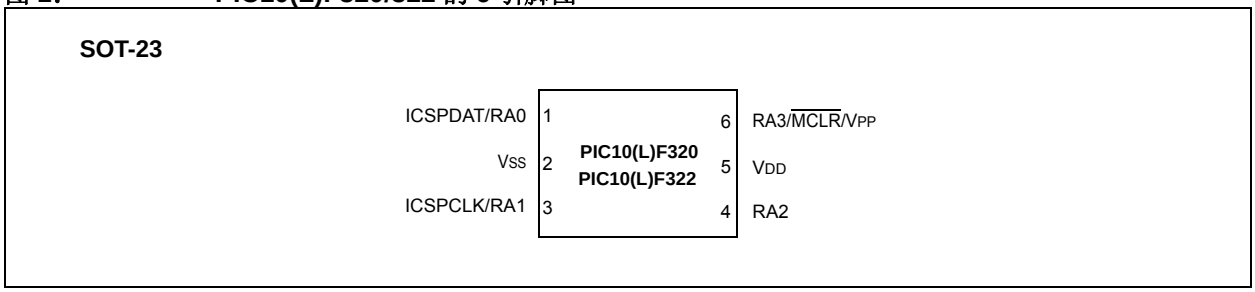


图 2： PIC10(L)F320/322 的 8 引脚图

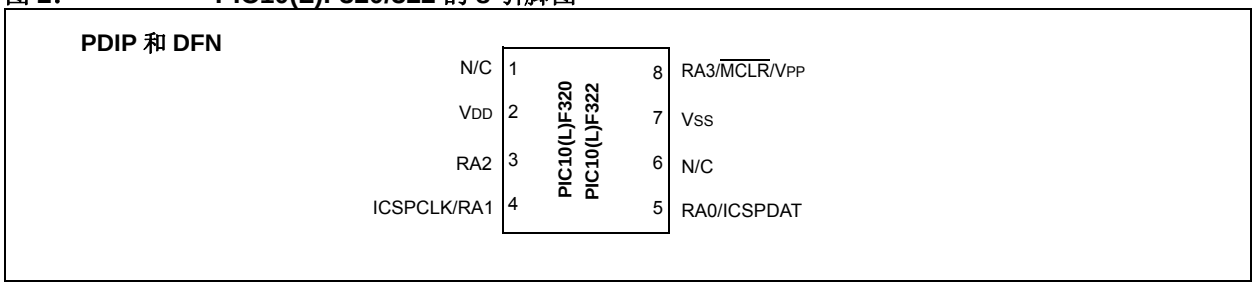


表 1： PIC10(L)F320/322 的 6 引脚和 8 引脚分配表

I/O	6 引脚	8 引脚	模拟	定时器	PWM	中断	上拉	CWG	NCO	CLC	基本功能	ICSP
RA0	1	5	AN0	—	PWM1	IOC0	有	CWG1A	—	CLC1IN0	—	ICSPDAT
RA1	3	4	AN1	—	PWM2	IOC1	有	CWG1B	NCO1CLK	CLC1	CLKIN	ICSPCLK
RA2	4	3	AN2	T0CKI	—	INT/IOC2	有	$\overline{\text{CWG1FLT}}$	NCO1	CLC1IN1	CLKR	
RA3	6	8	—	—	—	IOC3	有	—	—	—	$\overline{\text{MCLR}}$	VPP
N/C	—	1	—	—	—	—	—	—	—	—	—	—
N/C	—	6	—	—	—	—	—	—	—	—	—	—
VDD	5	2	—	—	—	—	—	—	—	—	VDD	—
VSS	2	7	—	—	—	—	—	—	—	—	VSS	—

目录

1.0 器件概述 6
2.0 存储器构成 9
3.0 器件配置 19
4.0 振荡器模块 24
5.0 复位 28
6.0 中断 35
7.0 掉电模式 （休眠） 44
8.0 看门狗定时器 （WDT） 46
9.0 闪存程序存储器控制 50
10.0 I/O 端口 67
11.0 电平变化中断 73
12.0 固定参考电压 （FVR） 77
13.0 内部稳压器 （IVR） 79
14.0 温度指示器模块 81
15.0 模数转换器 （ADC）模块 83
16.0 Timer0 模块 93
17.0 Timer2 模块 96
18.0 脉宽调制 （PWM）模块 98
19.0 可配置逻辑单元 （CLC） 104
20.0 数控振荡器 （NCO）模块 119
21.0 互补波形发生器 （CWG）模块 129
22.0 在线串行编程 （ICSP™） 144
23.0 指令集汇总 147
24.0 电气规范 156
25.0 直流和交流特性图表 176
26.0 开发支持 177
27.0 封装信息 181
附录 A： 数据手册版本历史 188
Microchip 网站 189
变更通知客户服务 189
客户支持 189
产品标识体系 190

© 2011-2014 Microchip Technology Inc.

致 客 户

我们旨在提供最佳文档供客户正确使用 Microchip 产品。为此，我们将不断改进出版物的内容和质量，使之更好地满足您的需求。出版物的质量将随新文档及更新版本的推出而得到提升。

如果您对本出版物有任何问题和建议，请通过电子邮件联系我公司 TRC 经理，电子邮件地址为 **CTRC@microchip.com**。我们期待您的反馈。

最新数据手册

欲获得本数据手册的最新版本，请访问我公司网站：

http://www.microchip.com

查看数据手册中任意一页下边角处的文献编号即可确定其版本。文献编号中紧跟数字串后的字母是版本号，例如：DS300000000A_CN 是文档的 A 版本。

勘误表

现有器件可能带有一份勘误表，描述了实际运行与数据手册中记载内容之间存在的细微差异以及建议的变通方法。一旦我们了解到器件 / 文档存在某些差异时，就会发布勘误表。勘误表上将注明其所适用的硅片版本和文件版本。

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

- Microchip 网站 **http://www.microchip.com**
- 当地 Microchip 销售办事处 （见最后一页）

在联络销售办事处时，请说明您所使用的器件型号、硅片版本和数据手册版本 （包括文献编号）。

客户通知系统

欲及时获知 Microchip 产品的最新信息，请到我公司网站 **http://www.microchip.com** 上注册。

1.0 器件概述

本数据手册描述了 PIC10(L)F320/322 器件。该器件采用了 6/8 引脚封装形式。图 1-1 给出了 PIC10(L)F320/322 器件的框图，而表 1-2 给出了引脚说明。

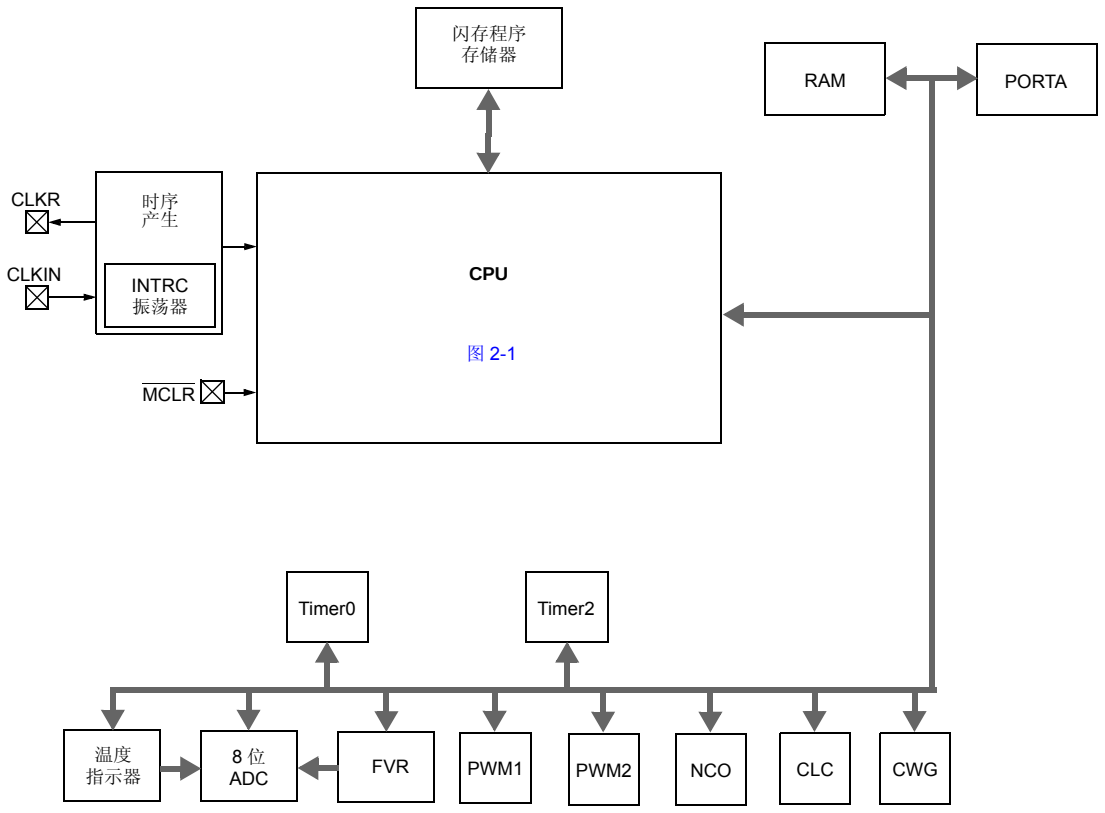
每个器件的可用外设，请参见表 1-1。

表 1-1： 器件外设汇总

外设		PIC10(L)F320	PIC10(L)F322
模数转换器 （ADC）		●	●
可配置逻辑单元 （CLC）		●	●
互补波形生成器 （CWG）		●	●
固定参考电压 （FVR）		●	●
数控振荡器 （NCO）		●	●
温度指示器		●	●
PWM 模块			
	PWM1	●	●
	PWM2	●	●
定时器			
	Timer0	●	●
	Timer2	●	●

© 2011-2014 Microchip Technology Inc.

图 1-1： PIC10(L)F320/322 框图



注 1： 关于外设的更多信息，请参见适用的章节。

表 1-2： PIC10(L)F320/322 引脚说明

名称	功能	输入类型	输出类型	说明
RA0/PWM1/CLC1IN0/CWG1A/AN0/ICSPDAT	RA0	TTL	CMOS	带 IOC 和 WPU 的通用 I/O。
	PWM1	—	CMOS	PWM 输出。
	CLC1IN0	ST	—	CLC 输入。
	CWG1A	—	CMOS	CWG 主输出。
	AN0	AN	—	A/D 输入通道。
	ICSPDAT	ST	CMOS	ICSP™ 数据 I/O。
RA1/PWM2/CLC1/CWG1B/AN1/CLKIN/ICSPCLK/NCO1CLK	RA1	TTL	CMOS	带 IOC 和 WPU 的通用 I/O。
	PWM2	—	CMOS	PWM 输出。
	CLC1	—	CMOS	CLC 输出。
	CWG1B	—	CMOS	CWG 互补输出。
	AN1	AN	—	A/D 输入通道。
	CLKIN	ST	—	外部时钟输入 （EC 模式）。
	ICSPCLK	ST	—	ICSP 编程时钟。
	NCO1CLK	ST	—	数控振荡器外部时钟输入。
RA2/INT/T0CKI/NCO1/CLC1IN1/CLKR/AN2/CWG1FLT	RA2	TTL	CMOS	带 IOC 和 WPU 的通用 I/O。
	INT	ST	—	外部中断。
	T0CKI	ST	—	Timer0 时钟输入。
	NCO1	—	CMOS	数控振荡器输出。
	CLC1IN1	ST	—	CLC 输入。
	CLKR	—	CMOS	参考时钟输出。
	AN2	AN	—	A/D 输入通道。
	$\overline{\text{CWG1FLT}}$	ST	—	互补波形发生器故障源 1 输入。
RA3/$\overline{\text{MCLR}}$/VPP	RA3	TTL	—	通用输入。
	$\overline{\text{MCLR}}$	ST	—	带内部上拉的主复位。
	VPP	HV	—	编程电压。
VDD	VDD	电源	—	正电源电压。
VSS	VSS	电源	—	参考地。

图注： AN = 模拟输入或输出 CMOS = CMOS 兼容输入或输出
TTL = 带 TTL 电平的 CMOS 输入 ST = 带施密特触发器电平的 CMOS输入
HV = 高电压

© 2011-2014 Microchip Technology Inc.

2.0 存储器构成

这些器件包含以下类型的存储器：

- 程序存储器
  - 配置字
  - 器件 ID
  - 用户 ID
  - 闪存程序存储器
- 数据存储器
  - 内核寄存器
  - 特殊功能寄存器
  - 通用 RAM
  - 公共 RAM

以下功能与程序存储器和数据存储器的访问和控制有关：

- PCL 和 PCLATH
- 堆栈
- 间接寻址

2.1 程序存储器构成

中档内核具有一个 13 位的程序计数器，可寻址 8K x 14 的程序存储空间。本器件系列仅实现了 8K 程序存储空间的最多 512 字。表 2-1 列出了 PIC10(L)F320/322 系列已实现的存储器大小。访问超出地址边界的单元将导致操作返回到已实现的存储空间内。复位向量地址为 0000h，中断向量地址为 0004h（见图 2-1 和图 2-2）。

表 2-1： 器件大小和地址

器件	程序存储器空间 （字）	最后的程序存储器地址	高耐用性闪存地址范围 (1)
PIC10(L)F320	256	00FFh	0F80h-00FFh
PIC10(L)F322	512	01FFh	1F80h-01FFh

注 1： 高耐用性闪存适用于范围内每个地址的低字节。

图 2-1： PIC10(L)F320 的程序存储器映射和堆栈

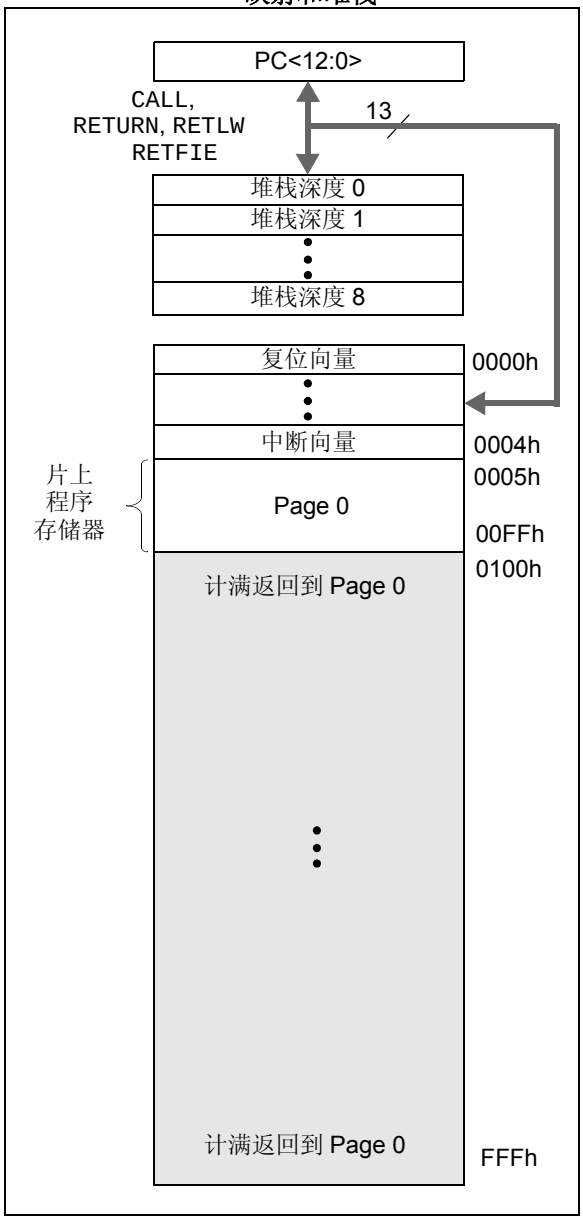
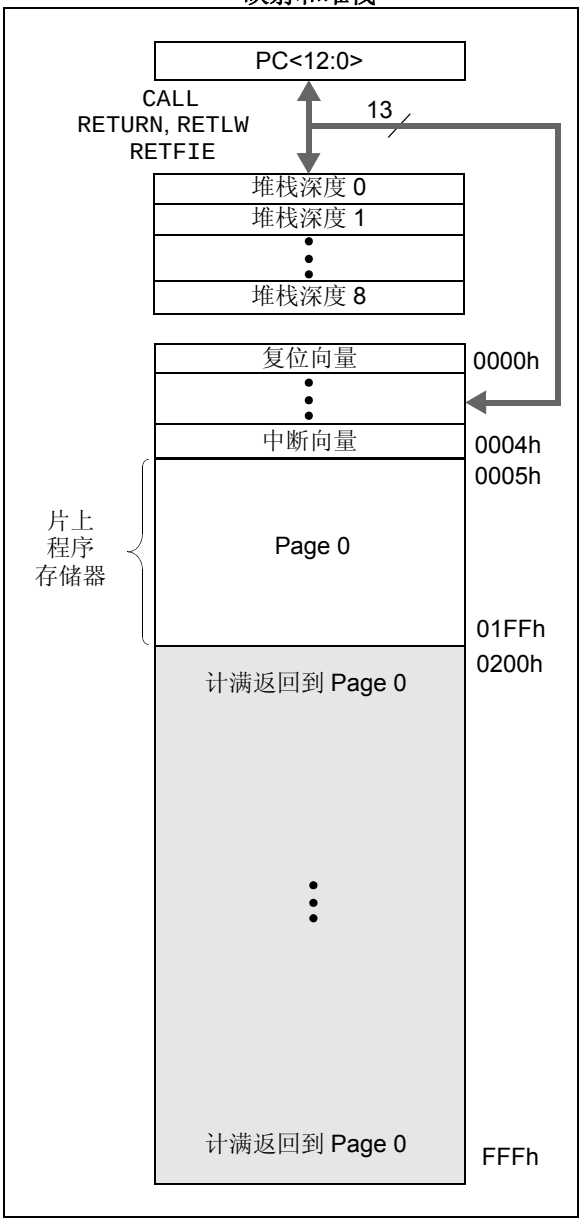


图 2-2： PIC10(L)F322 的程序存储器映射和堆栈



© 2011-2014 Microchip Technology Inc.

2.2 数据存储器构成

数据存储器位于一个存储区中，其中包含通用寄存器（General Purpose Register，GPR）和特殊功能寄存器（Special Function Register，SFR）。STATUS 寄存器的 RP<1:0> 位是存储区选择位。

RP1	RP0	
0	0	→ 选择 Bank 0

该存储区最多可以扩展到 7Fh （128 字节）。存储区的低地址单元保留为特殊功能寄存器。通用寄存器位于特殊功能寄存器的上方，实现为静态 RAM。

2.2.1 通用寄存器文件

PIC10(L)F320/322 的寄存器文件按 64 x 8 的形式构成。通过文件选择寄存器 （File Select Register， FSR）可以直接或间接访问每个寄存器 （见**第 2.4 节 “间接寻址、 INDF 和 FSR 寄存器”**）。

2.2.2 特殊功能寄存器

特殊功能寄存器是由 CPU 和外设功能对所需的器件操作进行控制的寄存器（见表 2-3）。这些寄存器是静态 RAM。

特殊寄存器可以分为两类：内核和外设。本章介绍有关“内核”的特殊功能寄存器。外设功能章节将介绍有关外设功能操作的特殊功能寄存器。

2.2.2.1 STATUS 寄存器

如寄存器 2-1 所示， STATUS 寄存器包含：

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

与任何其他寄存器一样，STATUS 寄存器可以作为任何指令的目标寄存器。如果 STATUS 寄存器是影响 Z、DC 或 C 位的指令的目标寄存器，那么将禁止对这 3 位进行写操作。根据器件逻辑，这些位会被置 1 或清零。此外，$\overline{TO}$ 和 $\overline{PD}$ 位均是不可写位。因此，当执行一条将 STATUS 寄存器作为目标寄存器的指令时，结果可能和预想的不一样。

例如， `CLRF STATUS` 会清零高 3 位，将 Z 位置 1，从而使得 STATUS 寄存器的内容变为 `000u u1uu`（其中 u = 不变）。

因此，建议仅使用 `BCF`、 `BSF`、 `SWAPF` 和 `MOVWF` 指令来改变 STATUS 寄存器的内容，因为这些指令不影响任何状态位。关于其他不影响状态位的指令，请参见**第 23.0 节 “指令集汇总”**。

注	1：	在 PIC10(L)F320 中， STATUS 寄存器的 IRP 和 RP1 位未使用，应保持清零。建议不要使用这些位，因为这可能影响与未来产品的向上兼容性。
	2：	在减法运算中， C 和 DC 位分别作为$\overline{借位}$位和半借位位。

© 2011-2014 Microchip Technology Inc.

寄存器 2-1： STATUS：状态寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R-1/q	R-1/q	R/W-x/u	R/W-x/u	R/W-x/u
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	q = 值取决于具体条件

bit 7 **IRP**：保留 (2)

bit 6-5 **RP<1:0>**：保留 (2)

bit 4 **$\overline{TO}$**：超时位
1 = 在上电、执行 CLRWDT 指令或 SLEEP 指令后
0 = 发生了 WDT 超时

bit 3 **$\overline{PD}$**：掉电位
1 = 在上电或执行 CLRWDT 指令后
0 = 执行了 SLEEP 指令

bit 2 **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

bit 1 **DC**：半进位 / $\overline{\text{半借位}}$位 （ADDWF、 ADDLW、 SUBLW 和 SUBWF 指令） (1)
1 = 结果的第 4 个低位发生了进位
0 = 结果的第 4 个低位未发生进位

bit 0 **C**：进位 / $\overline{\text{借位}}$位 （ADDWF、 ADDLW、 SUBLW 和 SUBWF 指令） (1)
1 = 结果的最高有效位发生了进位
0 = 结果的最高有效位未发生进位

注 **1**：对于$\overline{\text{借位}}$，极性是相反的。减法指令是通过加上第二个操作数的二进制补码来执行的。对于移位指令 （RRF 和 RLF），该位值来自源寄存器的高位或低位。

2：保持 0。

2.2.3 器件存储器映射

PIC10(L)F320/322 的存储器映射如表 2-2 所示。

表 2-2： PIC10(L)F320/322 存储器映射 （BANK 0）

INDF(*)	00h	PMADRL	20h	通用寄存器 32 字节	40h	通用寄存器 32 字节	60h
TMR0	01h	PMADRH	21h				
PCL	02h	PMDATL	22h				
STATUS	03h	PMDATH	23h				
FSR	04h	PMCON1	24h				
PORTA	05h	PMCON2	25h				
TRISA	06h	CLKRCON	26h				
LATA	07h	NCO1ACCL	27h				
ANSELA	08h	NCO1ACCH	28h				
WPUA	09h	NCO1ACCU	29h				
PCLATH	0Ah	NCO1INCL	2Ah				
INTCON	0Bh	NCO1INCH	2Bh				
PIR1	0Ch	保留	2Ch				
PIE1	0Dh	NCO1CON	2Dh				
OPTION_REG	0Eh	NCO1CLK	2Eh				
PCON	0Fh	保留	2Fh				
OSCCON	10h	WDTCON	30h				
TMR2	11h	CLC1CON	31h				
PR2	12h	CLC1SEL1	32h				
T2CON	13h	CLC1SEL2	33h				
PWM1DCL	14h	CLC1POL	34h				
PWM1DCH	15h	CLC1GLS0	35h				
PWM1CON	16h	CLC1GLS1	36h				
PWM2DCL	17h	CLC1GLS2	37h				
PWM2DCH	18h	CLC1GLS3	38h				
PWM2CON	19h	CWG1CON0	39h				
IOCAP	1Ah	CWG1CON1	3Ah				
IOCAN	1Bh	CWG1CON2	3Bh				
IOCAF	1Ch	CWG1DBR	3Ch				
FVRCON	1Dh	CWG1DBF	3Dh				
ADRES	1Eh	VREGCON	3Eh				
ADCON	1Fh	BORCON	3Fh		5Fh		7Fh

图注： = 未实现的数据存储单元，读为 0。

* = 非物理寄存器。

© 2011-2014 Microchip Technology Inc.

表 2-3： 特殊功能寄存器汇总 （BANK 0）

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
Bank 0											
00h	INDF	通过使用 FSR 的内容寻址此存储单元来寻址数据存储器 （非物理寄存器）								xxxx xxxx	xxxx xxxx
01h	TMR0	Timer0 模块寄存器								xxxx xxxx	uuuu uuuu
02h	PCL	程序计数器 （PC）的最低有效字节								0000 0000	0000 0000
03h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h	FSR	间接数据存储器地址指针								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	—	—	RA3	RA2	RA1	RA0	---- xxxx	---- uuuu
06h	TRISA	—	—	—	—	—(1)	TRISA2	TRISA1	TRISA0	---- 1111	---- 1111
07h	LATA	—	—	—	—	—	LATA2	LATA1	LATA0	---- -xxx	---- -uuu
08h	ANSELA	—	—	—	—	—	ANSA2	ANSA1	ANSA0	---- -111	---- -111
09h	WPUA	—	—	—	—	WPUA3	WPUA2	WPUA1	WPUA0	---- 1111	---- 1111
0Ah	PCLATH	—	—	—	—	—	—	—	PCLH0	---- ---0	---- ---0
0Bh	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 000u
0Ch	PIR1	—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—	-0-0 0-0-	-0-0 0-0-
0Dh	PIE1	—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—	-0-0 0-0-	-0-0 0-0-
0Eh	OPTION_REG	$\overline{WPUEN}$	INTEDG	T0CS	T0SE	PSA	PS<2:0>			1111 1111	uuuu uuuu
0Fh	PCON	—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$	---- --qq	---- --uu
10h	OSCCON	—	IRCF<2:0>			HFIOFR	—	LFIOFR	HFIOFS	-110 0-00	-110 0-00
11h	TMR2	Timer2 模块寄存器								0000 0000	0000 0000
12h	PR2	Timer2 周期寄存器								1111 1111	1111 1111
13h	T2CON	—	TOUTPS<3:0>				TMR2ON	T2CKPS<1:0>		-000 0000	-000 0000
14h	PWM1DCL	PWM1DCL<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
15h	PWM1DCH	PWM1DCH<7:0>								xxxx xxxx	uuuu uuuu
16h	PWM1CON	PWM1EN	PWM1OE	PWM1OUT	PWM1POL	—	—	—	—	0000 ----	0000 ----
17h	PWM2DCL	PWM2DCL<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
18h	PWM2DCH	PWM2DCH<7:0>								xxxx xxxx	uuuu uuuu
19h	PWM2CON	PWM2EN	PWM2OE	PWM2OUT	PWM2POL	—	—	—	—	0000 ----	0000 ----
1Ah	IOCAP	—	—	—	—	IOCAP3	IOCAP2	IOCAP1	IOCAP0	---- 0000	---- 0000
1Bh	IOCAN	—	—	—	—	IOCAN3	IOCAN2	IOCAN1	IOCAN0	---- 0000	---- 0000
1Ch	IOCAF	—	—	—	—	IOCAF3	IOCAF2	IOCAF1	IOCAF0	---- 0000	---- 0000
1Dh	FVRCON	FVREN	FVRRDY	TSEN	TSRNG	—	—	ADFVR<1:0>		0x00 --00	0x00 --00
1Eh	ADRES	A/D 结果寄存器								xxxx xxxx	uuuu uuuu
1Fh	ADCON	ADCS<2:0>			CHS<2:0>			GO/$\overline{DONE}$	ADON	0000 0000	0000 0000

图注： x = 未知； u = 不变； q = 值取决于具体条件； - = 未实现，读为 0； r = 保留。
阴影单元未实现，读为 0。

注 1： 未实现，读为 1。

表 2-3： 特殊功能寄存器汇总 （BANK 0） （续）

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
Bank 0 （续）											
20h	PMADRL	PMADR<7:0>								0000 0000	0000 0000
21h	PMADRH	—	—	—	—	—	—	—	PMADR8	---- ---0	---- ---0
22h	PMDATL	PMDAT<7:0>								xxxx xxxx	uuuu uuuu
23h	PMDATH	—	—	PMDAT<13:8>						--xx xxxx	--uu uuuu
24h	PMCON1	—	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	1000 0000	1000 q000
25h	PMCON2	程序存储器控制寄存器 2（非物理寄存器）								0000 0000	0000 0000
26h	CLKRCON	—	CLKROE	—	—	—	—	—	—	-0-- ----	-0-- ----
27h	NCO1ACCL	NCO1 累加器 <7:0>								0000 0000	0000 0000
28h	NCO1ACCH	NCO1 累加器 <15:8>								0000 0000	0000 0000
29h	NCO1ACCU	—				NCO1 累加器 <19..16>				---- 0000	---- 0000
2Ah	NCO1INCL	NCO1 递增值 <7:0>								0000 0001	0000 0001
2Bh	NCO1INCH	NCO1 递增值 <15:8>								0000 0000	0000 0000
2Ch	—	未实现								—	—
2Dh	NCO1CON	N1EN	N1OE	N1OUT	N1POL	—	—	—	N1PFM	0000 ---0	00x0 ---0
2Eh	NCO1CLK	N1PWS<2:0>			—	—	—	N1CKS<1:0>		000- --00	000- --00
2Fh	保留	保留								xxxx xxxx	uuuu uuuu
30h	WDTCON	—	—	WDTPS<4:0>					SWDTEN	--01 0110	--01 0110
31h	CLC1CON	LC1EN	LC1OE	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>			00x0 -000	00x0 -000
32h	CLC1SEL0	—	LC1D2S<2:0>			—	LC1D1S<2:0>			-xxx -xxx	-uuu -uuu
33h	CLC1SEL1	—	LC1D4S<2:0>			—	LC1D3S<2:0>			-xxx -xxx	-uuu -uuu
34h	CLC1POL	LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	0--- xxxx	0--- uuuu
35h	CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	xxxx xxxx	uuuu uuuu
36h	CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	xxxx xxxx	uuuu uuuu
37h	CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	xxxx xxxx	uuuu uuuu
38h	CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	xxxx xxxx	uuuu uuuu
39h	CWG1CON0	G1EN	G1OEB	G1OEA	G1POLB	G1POLA	—	—	G1CS0	0000 0--0	0000 0--0
3Ah	CWG1CON1	G1ASDLB<1:0>		G1ASDLA<1:0>		—	—	G1IS<1:0>		xxxx --xx	uuuu --uu
3Bh	CWG1CON2	G1ASE	G1ARSEN	—	—	—	—	G1ASDCLC1	G1ASDFLT	xx-- --xx	uu-- --uu
3Ch	CWG1DBR	—	—	CWG1DBR<5:0>						--xx xxxx	--uu uuuu
3Dh	CWG1DBF	—	—	CWG1DBF<5:0>						--xx xxxx	--uu uuuu
3Eh	VREGCON	—	—	—	—	—	—	VREGPM1	保留	---- --01	---- --01
3Fh	BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	10-- ---q	uu-- ---u

图注： x = 未知； u = 不变； q = 值取决于具体条件； - = 未实现，读为 0； r = 保留。
阴影单元未实现，读为 0。

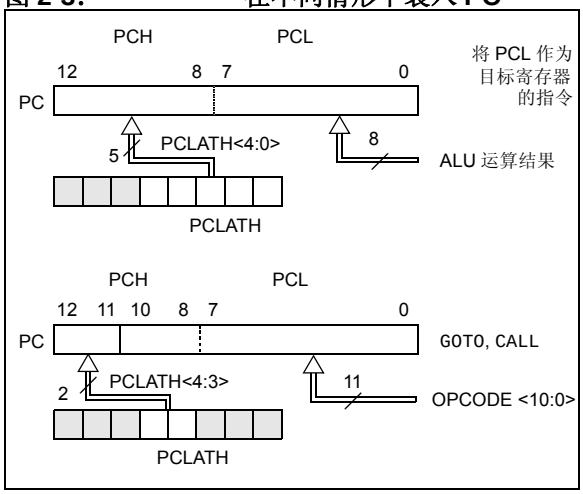
注 1： 未实现，读为 1。

© 2011-2014 Microchip Technology Inc.

2.3 PCL 和 PCLATH

程序计数器（Program Counter，PC）为 13 位宽。它的低字节来自可读写的 PCL 寄存器。高字节（PC<12:8>）来自 PCLATH，不可直接读写。任何复位都将清零 PC。图 2-3 给出了装入 PC 的两种情形。图 2-3 中上方的示例说明了如何在执行写操作时将 PC 装入到 PCL（PCLATH<4:0> → PCH）。图 2-3 中下方的示例说明了如何在 CALL 或 GOTO 指令执行期间装入 PC（PCLATH<4:3> → PCH）。

图 2-3：在不同情形下装入 PC



2.3.1 修改 PCL

在执行任何将 PCL 寄存器作为目标寄存器的指令的同时，程序计数器PC<12:8>位（PCH）将替换为PCLATH寄存器的内容。通过将所需的高 5 位写入 PCLATH 寄存器，可以更改程序计数器的整个内容。当将低 8 位写入 PCL 寄存器后，程序计数器的所有 13 位值将更改为 PCLATH 寄存器中的值以及那些写入到 PCL 寄存器的值。

计算 GOTO 指令是通过向程序计数器添加偏移量来实现的（ADDWF PCL）。通过修改 PCL 寄存器跳转到查询表或程序跳转表（计算 GOTO）时应谨慎。假设 PCLATH 设置为表起始地址，如果表长度大于 255 个指令或存储器地址的低 8 位在表中间从 0xFF 计满返回 0x00，则对于在表开头和表内目标位置之间发生的每次地址计满返回，PCLATH 都必须递增 1。

更多信息，请参考应用笔记 AN556 *“Implementing a Table Read”*（DS00556）。

2.3.2 堆栈

所有器件都具有一个 8 级深 x 13 位宽的硬件堆栈（见图 2-1）。堆栈空间不属于程序空间或数据空间的一部分，且堆栈指针不可读写。当执行 CALL 指令或中断导致跳转时，PC 值会被压入堆栈。当执行 RETURN、RETLW 或 RETFIE 指令时，PC 值从堆栈弹出。PCLATH 的值不受压栈或出栈操作的影响。

堆栈充当循环缓冲区。这就是说在压栈 8 次后，第 9 次压栈操作会覆盖第 1 次压栈操作存储的值，而第 10 次压栈操作会覆盖第 2 次压栈操作存储的值（以此类推）。

> **注 1：** 不存在指示堆栈上溢或堆栈下溢情况的状态位。
>
> **2：** 不存在称为 PUSH 或 POP 的指令 / 助记符。这些操作都是在执行 CALL、RETURN、RETLW 和 RETFIE 指令或跳转到中断地址向量时发生。

2.4 间接寻址、INDF 和 FSR 寄存器

INDF 寄存器不是物理寄存器。寻址 INDF 寄存器将导致间接寻址。

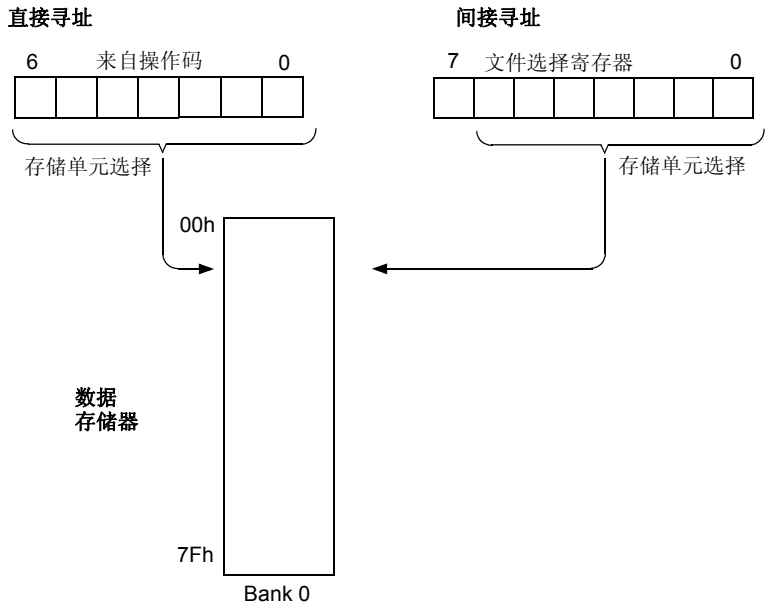
可以使用 INDF 寄存器进行间接寻址。任何使用 INDF 寄存器的指令实际上都将访问文件选择寄存器（FSR）指向的数据。读 INDF 本身将间接产生 00h。写入 INDF 寄存器不会间接产生任何操作（尽管可能影响状态位）。通过组合 8 位 FSR 和 STATUS 寄存器的 IRP 位，可以获得有效的 9 位地址，如图 2-4 所示。

例 2-1 给出了使用间接寻址清零 RAM 单元 40h-7Fh 的简单程序。

例 2-1：间接寻址

```
        MOVLW   0x40    ;initialize pointer
        MOVWF   FSR     ;to RAM
NEXT    CLRF    INDF    ;clear INDF register
        INCF    FSR     ;inc pointer
        BTFSS   FSR,7   ;all done?
        GOTO    NEXT    ;no clear next
CONTINUE                ;yes continue
```

图 2-4：　直接 / 间接寻址 PIC10(L)F320/322



注：　关于存储器映射的详细信息，请参见图 2-2。

© 2011-2014 Microchip Technology Inc.

3.0 器件配置

器件配置由配置字和器件 ID 组成。

3.1 配置字

有几个配置字位可用于选择不同的振荡器和存储器保护选项，这些位实现为位于 2007h 处的配置字。

3.2 寄存器定义：配置字

寄存器 3-1： CONFIG：配置字

U-1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1
—	WRT<1:0>		BORV	LPBOR	LVP
bit 13					bit 8

R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1
$\overline{CP}$	MCLRE	$\overline{PWRTE}$	WDTE<1:0>		BOREN<1:0>		FOSC
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	P = 可编程位

bit 13 **未实现：**读为 1

bit 12-11 **WRT<1:0>：**闪存自写保护位

256 W 闪存 （PIC10(L)F320）：
- 11 = 写保护关闭
- 10 = 000h 至 03Fh 受写保护， 040h 至 0FFh 可由 PMCON 控制修改
- 01 = 000h 至 07Fh 受写保护， 080h 至 0FFh 可由 PMCON 控制修改
- 00 = 000h 至 0FFh 受写保护，没有地址可由 PMCON 控制修改

512 W 闪存 （PIC10(L)F322）：
- 11 = 写保护关闭
- 10 = 000h 至 07Fh 受写保护， 080h 至 1FFh 可由 PMCON 控制修改
- 01 = 000h 至 0FFh 受写保护， 100h 至 1FFh 可由 PMCON 控制修改
- 00 = 000h 至 1FFh 受写保护，没有地址可由 PMCON 控制修改

bit 10 **BORV：**欠压复位电压选择位
- 1 = 欠压复位电压 （V_{BOR}），选择低跳变点。
- 0 = 欠压复位电压 （V_{BOR}），选择高跳变点。

bit 9 **LPBOR：**低功耗欠压复位使能位
- 1 = 使能低功耗欠压复位
- 0 = 禁止低功耗欠压复位

bit 8 **LVP：**低电压编程使能位
- 1 = 使能低电压编程。$\overline{MCLR}$/VPP 引脚功能为 $\overline{MCLR}$。
- 0 = 必须使用 $\overline{MCLR}$/VPP 引脚上的高电压进行编程

bit 7 **$\overline{CP}$：**代码保护位 (2)
- 1 = 禁止程序存储器代码保护
- 0 = 使能程序存储器代码保护

bit 6 **MCLRE：**$\overline{MCLR}$/VPP 引脚功能选择位

如果 LVP 位 = 1：
- 忽略该位。

如果 LVP 位 = 0：
- 1 = $\overline{MCLR}$/VPP 引脚功能为 $\overline{MCLR}$ ；使能弱上拉功能。
- 0 = $\overline{MCLR}$/VPP 引脚功能为数字输入；内部禁止 $\overline{MCLR}$ ；弱上拉功能由 WPUA3 位控制。

注 **1：** 使能欠压复位并不能自动使能上电延时定时器。
2： 一旦使能代码保护，则只能通过批量擦除器件来禁止代码保护。
3： 请参见 V_{BOR} 参数了解特定的跳变点电压。

© 2011-2014 Microchip Technology Inc.

寄存器 3-1： CONFIG：配置字 （续）

bit 5 **$\overline{\text{PWRTE}}$**：上电延时定时器使能位 (1)
1 = 禁止 PWRT
0 = 使能 PWRT

bit 4-3 **WDTE<1:0>**：看门狗定时器使能位
11 = 使能 WDT
10 = 运行时使能 WDT，休眠时禁止 WDT
01 = WDT 由 WDTCON 寄存器中的 SWDTEN 位控制
00 = 禁止 WDT

bit 2-1 **BOREN<1:0>**：欠压复位使能位
11 = 使能欠压复位；忽略 SBOREN 位
10 = 运行时使能欠压复位；休眠时禁止欠压复位；忽略 SBOREN 位
01 = 欠压复位由 BORCON 寄存器中的 SBOREN 位控制
00 = 禁止欠压复位；忽略 SBOREN 位

bit 0 **FOSC**：振荡器选择位
1 = CLKIN 引脚上的 EC
0 = CLKIN 引脚为 INTOSC 振荡器 I/O 功能

注 **1**：使能欠压复位并不能自动使能上电延时定时器。
2：一旦使能代码保护，则只能通过批量擦除器件来禁止代码保护。
3：请参见 VBOR 参数了解特定的跳变点电压。

3.3 代码保护

代码保护可以防止对器件的未授权访问。程序存储器保护和数据存储器保护都是独立控制的。任何代码保护设置不会影响对程序存储器和数据存储器的内部访问。

3.3.1 程序存储器保护

配置字中的 $\overline{CP}$ 位可以禁止对整个程序存储空间的外部读写操作。当 $\overline{CP}$ = 0 时，禁止对程序存储器的外部读写操作，读操作将返回全 0。无论保护位的设置如何，CPU 可以继续读程序存储器。写程序存储器取决于写保护设置。更多信息，请参见**第 3.4 节 “写保护”**。

3.4 写保护

写保护可以防止对器件意外的自写访问。在允许修改程序存储器其他区域的同时，可以保护应用程序 （例如，引导加载程序软件）。

配置字中的WRT<1:0>位定义了受保护的程序存储块的大小。

3.5 用户 ID

有 4 个存储单元 （2000h-2003h）被指定为 ID 地址单元，供用户存储校验和或其他代码标识号。在正常执行过程中可以读写这些单元。关于访问这些存储单元的更多信息，请参见**第 3.6 节 “器件 ID 和版本 ID”**。关于校验和计算的更多信息，请参见 “PIC10(L)F320/322 Flash Memory Programming Specification” （DS41572）。

© 2011-2014 Microchip Technology Inc.

3.6 器件 ID 和版本 ID

存储单元 2006h 用于存储器件 ID 和版本 ID。高 9 位保存器件 ID。低 5 位保存版本 ID。关于访问这些存储单元的更多信息，请参见第 9.4 节 “用户 ID、器件 ID 和配置字访问”。

开发工具（例如，器件编程器和调试器）可以用来读取器件 ID 和版本 ID。

3.7 寄存器定义：器件和版本

寄存器 3-2： DEVID：器件 ID 寄存器 (1)

R	R	R	R	R	R
DEV<8:3>					
bit 13					bit 8

R	R	R	R	R	R	R	R
DEV<2:0>			REV<4:0>				
bit 7							bit 0

图注：	
R = 可读位	
1 = 置 1	0 = 清零

bit 13-5 **DEV<8:0>：**器件 ID 位

器件	DEVID<13:0> 值	
	DEV<8:0>	REV<4:0>
PIC10F320	10 1001 101	x xxxx
PIC10LF320	10 1001 111	x xxxx
PIC10F322	10 1001 100	x xxxx
PIC10LF322	10 1001 110	x xxxx

bit 4-0 **REV<4:0>：**版本 ID 位

这些位用于标识版本。

注 1： 该单元不可写。

4.0 振荡器模块

4.1 概述

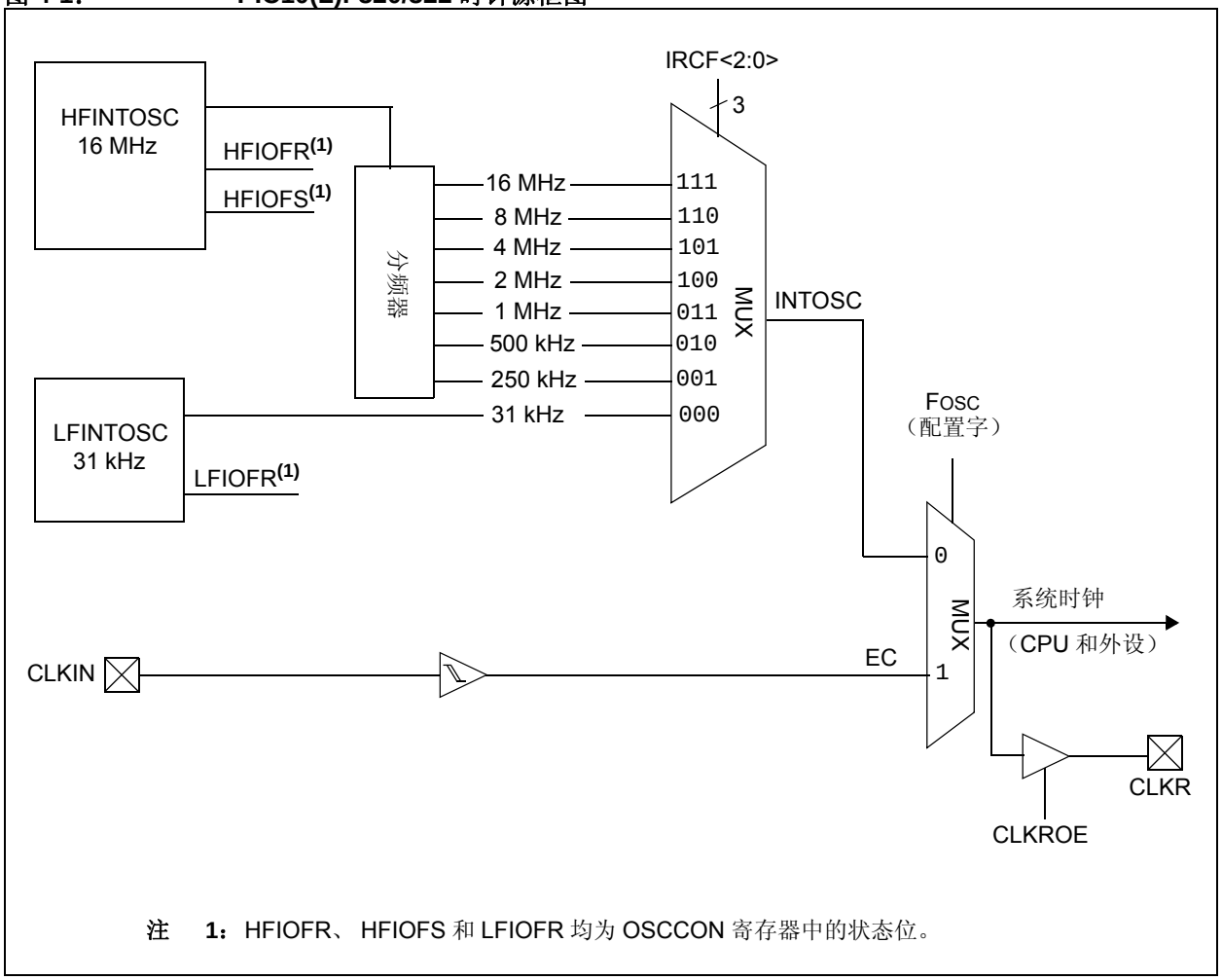
该振荡器模块有多种时钟源和选择功能，从而使其应用非常广泛，并可最大程度地提高性能和降低功耗。图 4-1 给出了振荡器模块的框图。

系统可配置为使用内部校准高频振荡器作为时钟源，并可通过软件选择速度。

时钟源模式由配置字中的 FOSC 位（CONFIG）设置。

1. 来自 CLKIN 的 EC 振荡器。
2. INTOSC 振荡器，未使能 CLKIN。

图 4-1： PIC10(L)F320/322 时钟源框图



注 1：HFIOFR、 HFIOFS 和 LFIOFR 均为 OSCCON 寄存器中的状态位。

© 2011-2014 Microchip Technology Inc.

4.2 时钟源模式

时钟源模式可以分为外部或内部两类。

- 内部时钟源 （INTOSC）包含在振荡器模块中，具有 8 个可选输出频率，最高内部频率为 16 MHz。
- 外部时钟（External Clock， EC）模式依靠外部信号作为时钟源。

可以通过配置字的FOSC位选择外部或内部时钟源作为系统时钟。

4.3 内部时钟模式

内部时钟源内置在振荡器模块中。内部振荡器模块具有两个内部振荡器，用于生成所有内部系统时钟源：16 MHz 高频内部振荡器 （HFINTOSC）和 31 kHz 低频内部振荡器（LFINTOSC）。

HFINTOSC 由一个主时钟和一个辅助时钟组成。辅助时钟先以快速启动时间启动，但精度较低。辅助时钟就绪信号通过 OSCCON 寄存器的 HFIOFR 位指示。接着，主时钟以较慢的启动时间启动，但具有更高的精度。当 OSCCON 寄存器的 HFIOFS 位为高电平时，主时钟稳定。

4.3.1 INTOSC 模式

当配置字的 FOSC 位清零时，即选择了 INTOSC 模式。选择 INTOSC 后， CLKIN 引脚可用于通用 I/O。更多信息，请参见**第 3.0 节 “器件配置”**。

4.3.2 频率选择位 （IRCF）

16 MHz HFINTOSC 的输出连接到分频器和多路开关（见图 4-1）。 OSCCON 寄存器的内部振荡器频率选择位 （IRCF）用于选择内部振荡器的输出频率：

- HFINTOSC
  - 16 MHz
  - 8 MHz （复位后的默认频率）
  - 4 MHz
  - 2 MHz
  - 1 MHz
  - 500 kHz
  - 250 kHz
- LFINTOSC
  - 31 kHz

注：	发生任何复位后， OSCCON 寄存器的 IRCF<2:0> 位被设置为 110 且频率选择设置为 8 MHz。用户可以修改 IRCF 位来选择不同的频率。

使用 IRCF 位在 HFINTOSC 频率间进行切换时没有延时。这是因为切换仅涉及对频率输出分频器的更改。

关于起振延时规范，请参见**第 24.0 节 “电气规范”**。

4.4 寄存器定义：参考时钟控制

寄存器 4-1： CLKRCON—— 参考时钟控制寄存器

U-0	R/W-0/0	U-0	U-0	U-0	U-0	U-0	U-0
—	CLKROE	—	—	—	—	—	—
bit 7							bit 0

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
q = 值取决于具体条件			

bit 7 未实现：读为 0

bit 6 **CLKROE**：参考时钟输出使能位
1 = 使能参考时钟输出 （CLKR），无论 TRIS 设置如何
0 = 禁止参考时钟输出

bit 5-0 未实现：读为 0

4.5 寄存器定义：振荡器控制

寄存器 4-2： OSCCON：振荡器控制寄存器

U-0	R/W-1/1	R/W-1/1	R/W-0/0	R-0/0	U-0	R-0/0	R-0/0
—	IRCF<2:0>			HFIOFR	—	LFIOFR	HFIOFS
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	q = 值取决于具体条件

bit 7 未实现：读为 0

bit 6-4 **IRCF<2:0>**：INTOSC （Fosc）频率选择位
111 = 16 MHz
110 = 8 MHz （默认值）
101 = 4 MHz
100 = 2 MHz
011 = 1 MHz
010 = 500 kHz
001 = 250 kHz
000 = 31 kHz （LFINTOSC）

bit 3 **HFIOFR**：高频内部振荡器就绪位
1 = 16 MHz 内部振荡器 （HFINTOSC）已就绪
0 = 16 MHz 内部振荡器 （HFINTOSC）未就绪

bit 2 未实现：读为 0

bit 1 **LFIOFR**：低频内部振荡器就绪位
1 = 31 kHz 内部振荡器 （LFINTOSC）已就绪
0 = 31 kHz 内部振荡器 （LFINTOSC）未就绪

bit 0 **HFIOFS**：高频内部振荡器稳定位
1 = 16 MHz 内部振荡器 （HFINTOSC）已稳定
0 = 16 MHz 内部振荡器 （HFINTOSC）未稳定

© 2011-2014 Microchip Technology Inc.

4.6 外部时钟模式

4.6.1 EC 模式

外部时钟（EC）模式将外部产生的逻辑电平信号作为系统时钟源。在该模式下工作时，外部时钟源应连接到 CLKIN 输入引脚。

表 4-1： 与时钟源相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
CLKRCON	—	CLKROE	—	—	—	—	—	—	26
OSCCON	—	IRCF<2:0>			HFIOFR	—	LFIOFR	HFIOFS	26

图注： x = 未知， u = 不变， – = 未实现单元，读为 0。 ECWG 不使用阴影单元。

表 4-2： 与时钟源相关的配置字汇总

名称	Bit	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG	13:8	—	—	—	WRT<1:0>		BORV	LPBOR	LVP	20
	7:0	$\overline{CP}$	MCLRE	$\overline{PWRTE}$	WDTE<1:0>		BOREN<1:0>		FOSC	

图注： — = 未实现单元，读为 0。时钟源不使用阴影单元。

5.0 复位

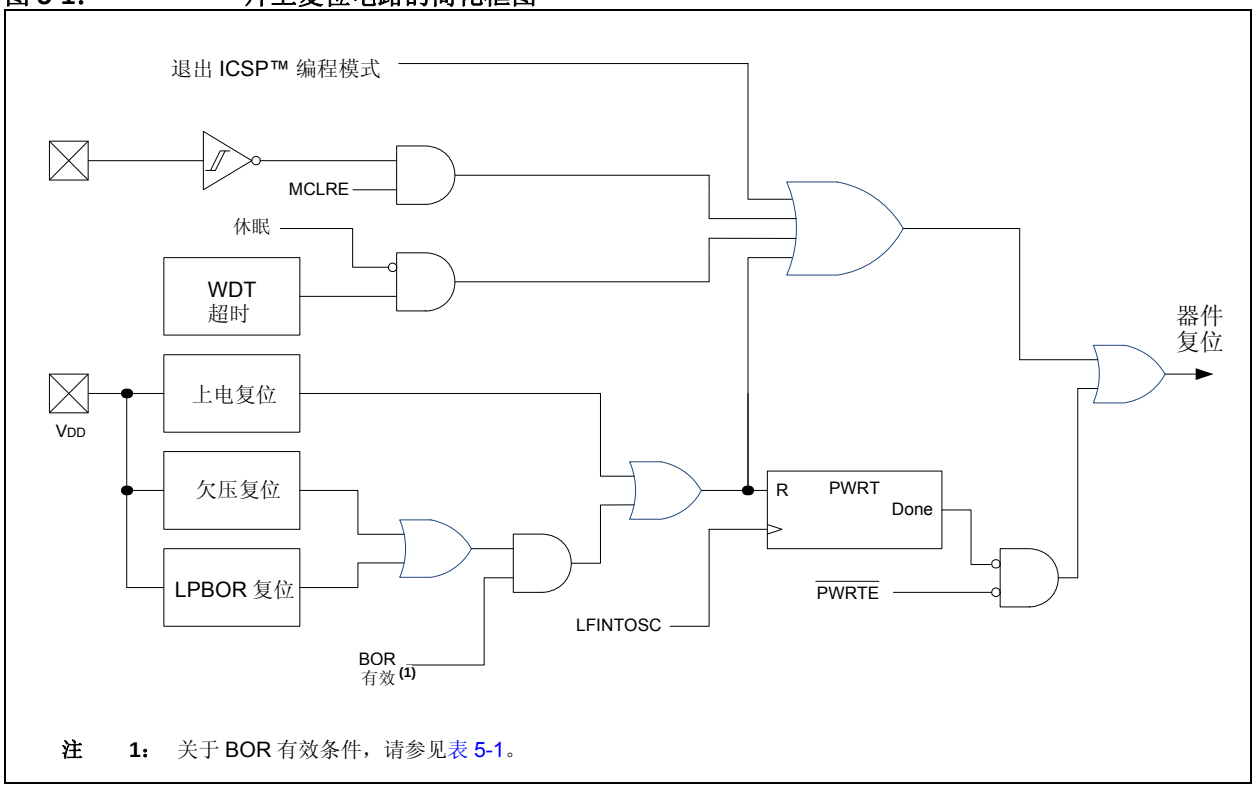
有多种方式可以复位器件：

- 上电复位 （POR）
- 欠压复位 （BOR）
- 低功耗欠压复位 （Low-Power Brown-out Reset，LPBOR）
- $\overline{\text{MCLR}}$ 复位
- WDT 复位
- 退出编程模式

要使 VDD 稳定，可以使能可选的上电延时定时器，以在 BOR 或 POR 事件后延长复位时间。

图 5-1 给出了片上复位电路的简化框图。

图 5-1： 片上复位电路的简化框图



注 1： 关于 BOR 有效条件，请参见表 5-1。

© 2011-2014 Microchip Technology Inc.

5.1 上电复位 （POR）

在 VDD 达到足以使器件正常工作的最低电平之前，POR 电路将器件保持在复位状态。 VDD 上升缓慢、高速运行或模拟性能可能要求工作电压大于最小 VDD。在满足所有器件工作条件之前，可以使用 PWRT、BOR 或 MCLR 功能延长启动周期。

5.1.1 上电延时定时器 （PWRT）

在 POR 或欠压复位时，上电延时定时器提供一个标称值为 64 ms 的延时。

只要 PWRT 处于工作状态，器件就保持在复位状态。PWRT 延时为 VDD 上升到所需的电平提供额外的时间。通过清零配置字中的 PWRTE 位，可以使能上电延时定时器。

在释放 POR 和 BOR 之后，启动上电延时定时器。

更多信息，请参见应用笔记 AN607 “*Power-up Trouble Shooting*”（DS00607）。

5.2 欠压复位 （BOR）

当 VDD 达到可选的最低电平时， BOR 电路将器件保持在复位状态。在 BOR 和 POR 之间的整个电压范围内，可实现执行保护功能。

欠压复位模块有4种工作模式，由配置字的BOREN<1:0>位控制这些模式。这 4 种工作模式是：

- BOR 始终开启
- BOR 在休眠时关闭
- BOR 由软件控制
- BOR 始终关闭

更多信息，请参见表 5-1。

通过配置寄存器 3-1 中的 BORV 位，可以选择欠压复位电压级别。

VDD 噪声抑制滤波器可以防止在发生小事件时触发 BOR。如果 VDD 下降到 VBOR 以下，且持续时间大于参数 TBORDC，器件将复位。更多信息，请参见图 5-2。

表 5-1： BOR 工作模式

BOREN<1:0>	SBOREN	器件模式	BOR 模式	器件工作的情况： 释放 POR 或从休眠中唤醒
11	X	X	有效	等待 BOR 就绪 (1) （BORRDY = 1）
10	X	唤醒	有效	等待 BOR 就绪 （BORRDY = 1）
		休眠	禁止	
01	1	X	有效	等待 BOR 就绪 (1) （BORRDY = 1）
	0	X	禁止	立即开始 （BORRDY = x）
00	X	X	禁止	

注 1： 在这两种特殊情况 （释放 POR 和从休眠中唤醒）下，不会延时启动。由于 BOR 电路由 BOREN<1:0> 位强制启动，因此在 CPU 准备执行指令之前，必须将 BOR 就绪标志 （BORRDY = 1）置 1。

5.2.1 BOR 始终开启

当配置字的 BOREN 位编程为 11 时， BOR 始终开启。在 BOR 准备就绪且 VDD 高于 BOR 阈值之前，器件启动将延时。

BOR 保护功能在休眠期间有效。 BOR 不会延时从休眠中唤醒。

5.2.2 BOR 在休眠时关闭

当配置字的 BOREN 位编程为 10 时，除了 BOR 在休眠时关闭之外，BOR 始终开启。在 BOR 准备就绪且 VDD 高于 BOR 阈值之前，器件启动将延时。

BOR保护功能在休眠期间无效。在BOR准备就绪之前，器件唤醒将延时。

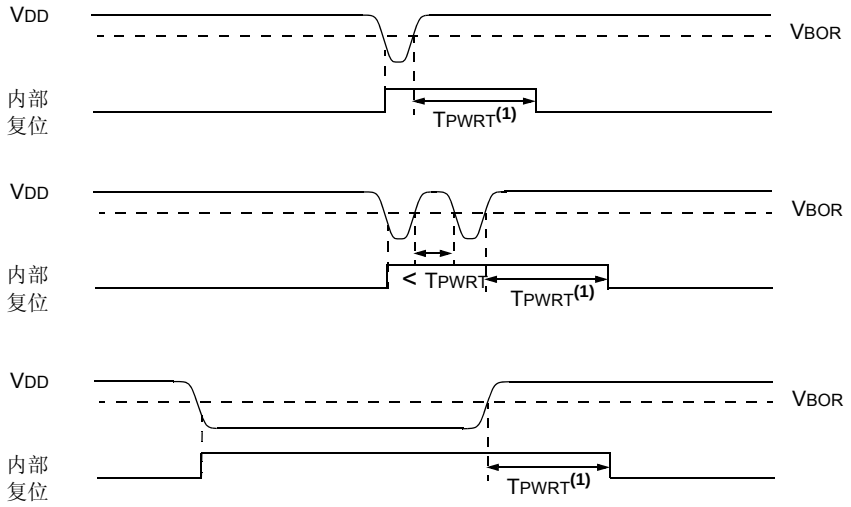
5.2.3 BOR 由软件控制

当配置字的 BOREN 位编程为 01 时，BOR 由 BORCON 寄存器的 SBOREN 位控制。BOR 就绪条件或 VDD 电平不会延时器件启动。

只要 BOR 电路准备就绪，则开始 BOR 保护。BORCON 寄存器的 BORRDY 位会反映 BOR 电路的状态。

BOR 保护功能不受休眠影响。

图 5-2： 欠压情形



注 1： 只有在 $\overline{\text{PWRTE}}$ 位编程为 0 时，才会有 TPWRT 延时。

5.3 寄存器定义：BOR 控制

寄存器 5-1： **BORCON：欠压复位控制寄存器**

R/W-1/u	R/W-0/u	U-0	U-0	U-0	U-0	U-0	R-q/u
SBOREN	BORFS(1)	—	—	—	—	—	BORRDY
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	q = 值取决于具体条件

bit 7 **SBOREN：**软件欠压复位使能位
如果配置字中的 BOREN <1:0> ≠ 01：
SBOREN 位是可读写位，但是不会影响 BOR。
如果配置字中的 BOREN <1:0> = 01：
1 = 使能 BOR
0 = 禁止 BOR

bit 6 **BORFS：**欠压复位快速启动位 (1)
如果 BOREN<1:0> = 11 （始终开启）或 BOREN<1:0> = 00 （始终关闭）：
BORFS 位是可读写位，但是不产生任何影响。
如果 BOREN <1:0> = 10 （在休眠模式下禁止）或 BOREN<1:0> = 01 （由软件控制）：
1 = 带隙始终被强制开启 （包括休眠 / 唤醒 / 工作情形）
0 = 带隙正常工作，并且可以关闭

bit 5-1 **未实现：**读为 0

bit 0 **BORRDY：**欠压复位电路就绪状态位
1 = 欠压复位电路有效
0 = 欠压复位电路无效

注 1： BOREN<1:0> 位位于配置字中。

© 2011-2014 Microchip Technology Inc.

5.4 低功耗欠压复位 （LPBOR）

低功耗欠压复位 （LPBOR）是复位子系统的一个重要组成部分。欲了解 BOR 如何与其他模块交互，请参见图 5-1。

LPBOR用于监视外部VDD引脚。当检测到电压太低时，器件将保持在复位状态。当此情形发生时，寄存器位（$\overline{\text{BOR}}$）将发生变化以指示已发生BOR复位。对于BOR和 LPBOR，所设置的是同一位。请参见寄存器 5-2。

5.4.1 使能 LPBOR

通过配置字的 LPBOR 位控制 LPBOR。当擦除器件时，LPBOR 模块默认为禁止状态。

5.4.1.1 LPBOR 模块输出

LPBOR 模块的输出是一个用于指示是否要将复位置为有效的信号。该信号与 BOR 模块的复位信号进行或运算以提供通用 $\overline{\text{BOR}}$ 信号，并送至 PCON 寄存器和电源控制模块。

5.5 $\overline{\text{MCLR}}$

$\overline{\text{MCLR}}$ 是用于复位器件的可选外部输入。$\overline{\text{MCLR}}$ 功能由配置字的 MCLRE 和 LVP 位控制 （表 5-2）。

表 5-2： $\overline{\text{MCLR}}$ 配置

MCLRE	LVP	$\overline{\text{MCLR}}$
0	0	禁止
1	0	使能
x	1	使能

5.5.1 $\overline{\text{MCLR}}$ 使能

当 $\overline{\text{MCLR}}$ 使能且引脚保持为低电平时，器件保持在复位状态。 $\overline{\text{MCLR}}$ 引脚通过内部弱上拉连接到 VDD。

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

注：	复位不会驱动 $\overline{\text{MCLR}}$ 引脚为低电平。

5.5.2 $\overline{\text{MCLR}}$ 禁止

当禁止 $\overline{\text{MCLR}}$ 时，引脚用作通用输入，内部弱上拉由软件控制。

5.6 看门狗定时器 （WDT）复位

如果固件在超时周期内未发出 CLRWDT 指令，看门狗定时器将产生复位。 STATUS 寄存器中的 $\overline{\text{TO}}$ 和 $\overline{\text{PD}}$ 位都会更改，以指示 WDT 复位。更多信息，请参见**第 8.0 节 “看门狗定时器”**。

5.7 退出 ICSP 编程模式

退出编程模式时，器件执行的操作与发生 POR 时一样。

5.8 上电延时定时器

在 BOR 或 POR 事件之后，上电延时定时器可以延迟器件执行。在器件开始运行之前，该定时器通常用于使 VDD 稳定下来。

上电延时定时器由配置字中的 $\overline{\text{PWRTE}}$ 位控制。

5.9 启动序列

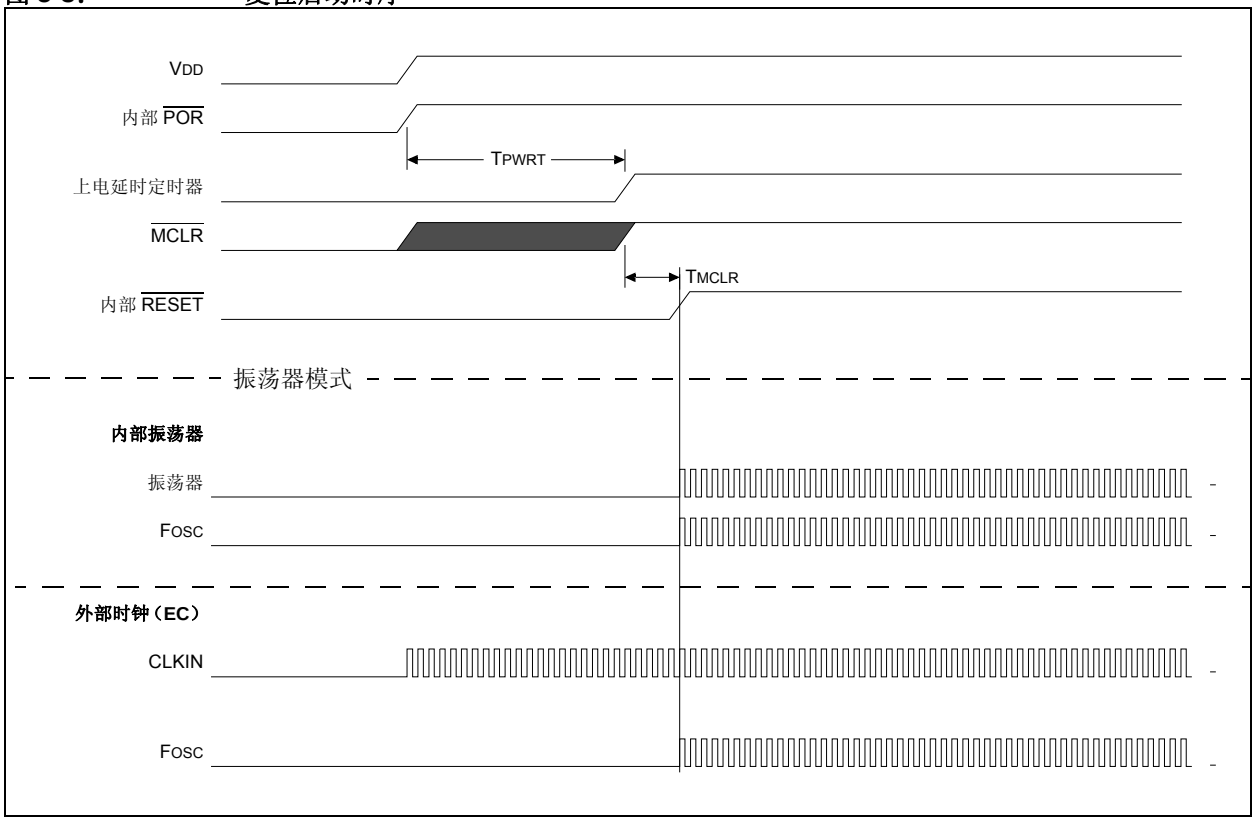
当释放 POR 或 BOR 之后，在器件开始执行之前必须发生以下事件：

1. 上电延时定时器运行结束 （如果使能）。
2. 必须释放 $\overline{\text{MCLR}}$ （如果使能）。

总超时时间将根据振荡器配置和上电延时定时器配置而变化。更多信息，请参见**第 4.0 节 “振荡器模块”**。

上电延时定时器独立于 $\overline{\text{MCLR}}$ 复位运行。如果 $\overline{\text{MCLR}}$ 保持足够长时间的低电平，上电延时定时器将超时。当 $\overline{\text{MCLR}}$ 拉为高电平之后，器件将在 10 个 FOSC 周期后开始执行 （见图 5-3）。这对于测试或同步多个并行工作的器件来说非常有用。

图 5-3： 复位启动时序



© 2011-2014 Microchip Technology Inc.

5.10 确定复位原因

发生任何复位时，STATUS 和 PCON 寄存器中的多个位都会更新，以指示复位原因。表 5-3 和表 5-4 给出了这些寄存器的复位条件。

表 5-3：复位状态位及其含义

$\overline{POR}$	$\overline{BOR}$	$\overline{TO}$	$\overline{PD}$	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 从休眠模式中唤醒
u	u	u	u	正常工作期间的 $\overline{MCLR}$ 复位
u	u	1	0	休眠期间的 $\overline{MCLR}$ 复位

表 5-4：特殊寄存器的复位条件

条件	程序计数器	STATUS 寄存器	PCON 寄存器
上电复位	0000h	0001 1000	---- --0x
正常工作期间的 $\overline{MCLR}$ 复位	0000h	000u uuuu	---- --uu
休眠期间的 $\overline{MCLR}$ 复位	0000h	0001 0uuu	---- --uu
WDT 复位	0000h	0000 uuuu	---- --uu
WDT 从休眠模式中唤醒	PC + 1	0000 0uuu	---- --uu
欠压复位	0000h	0001 1uuu	---- --u0
通过中断从休眠模式中唤醒	PC + 1(1)	0001 0uuu	---- --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）置 1 时，将返回地址压入堆栈，并且在执行 PC + 1 之后将中断向量 （0004h）装入 PC。

5.11 电源控制 （PCON）寄存器

电源控制（PCON）寄存器包含用于区分以下复位的标志位：

- 上电复位 （$\overline{\text{POR}}$）
- 欠压复位 （$\overline{\text{BOR}}$）

PCON 寄存器位如寄存器 5-2 所示。

5.12 寄存器定义：电源控制

寄存器 5-2： PCON：电源控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W/HC-q/u	R/W/HC-q/u
—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

图注：		
HC = 由硬件清零		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	q = 值取决于具体条件

bit 7-2 **未实现：**读为 0

bit 1 **$\overline{\text{POR}}$：**上电复位状态位
1 = 未发生上电复位
0 = 发生了上电复位 （发生上电复位后，必须用软件置 1）

bit 0 **$\overline{\text{BOR}}$：**欠压复位状态位
1 = 未发生欠压复位
0 = 发生了欠压复位 （发生上电复位或欠压复位后，必须用软件置 1）

表 5-5： 与复位相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	30
PCON	—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	34
STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	13
WDTCON	—	—	WDTPS<4:0>					SWDTEN	48

图注： — = 未实现单元，读为 0。复位不使用阴影单元。

表 5-6： 与复位相关的配置字汇总

名称	Bit	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG	13:8	—	—	—	WRT<1:0>		BORV	LPBOR	LVP	20
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		BOREN<1:0>		FOSC	

图注： — = 未实现单元，读为 0。复位不使用阴影单元。

© 2011-2014 Microchip Technology Inc.

6.0 中断

中断功能允许某些事件的优先级高于正常的程序流程。固件用于判断中断源并采取相应的操作。一些中断还可配置为将 MCU 从休眠模式中唤醒。

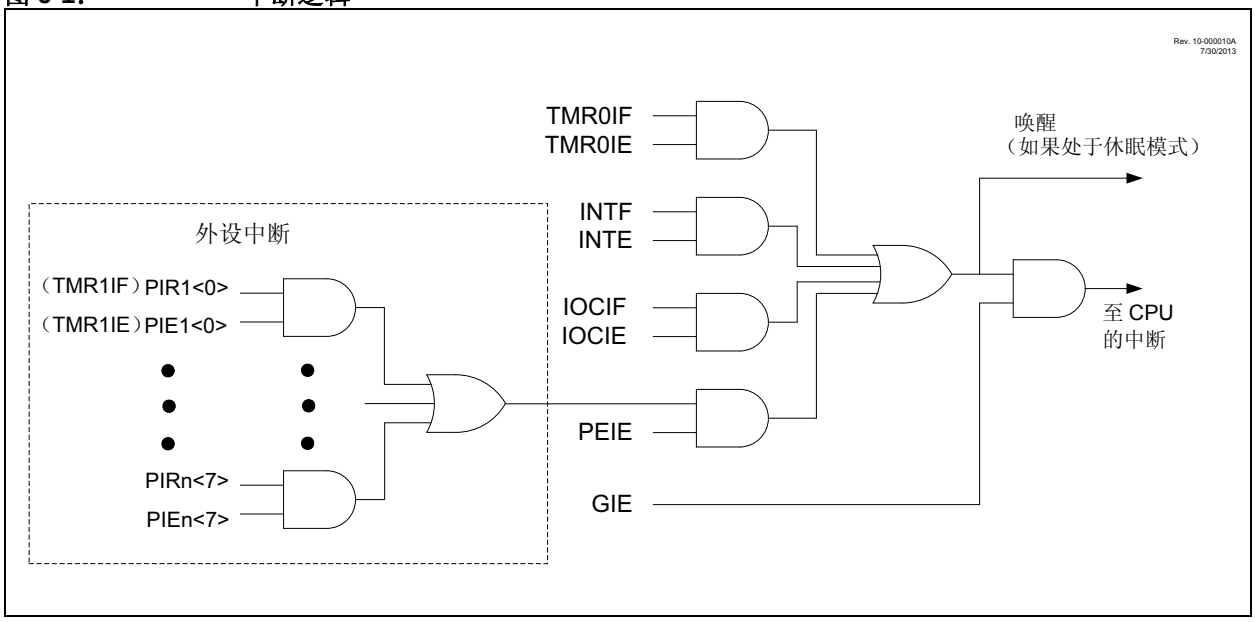
本章包含以下中断信息：

- 工作原理
- 中断响应延时
- 休眠期间的中断
- INT 引脚
- 中断期间的现场保护

许多外设都可以产生中断。更多详细信息，请参见相应的章节。

图 6-1 给出了中断逻辑的框图。

图 6-1：中断逻辑



6.1 工作原理

任何器件复位都将禁止中断。可通过将以下位置 1 来允许中断：

- INTCON 寄存器的 GIE 位
- 特殊中断事件的中断允许位
- INTCON 寄存器的 PEIE 位 （如果 PIE1 寄存器包含中断事件的中断允许位）

INTCON 和 PIR1 寄存器通过中断标志位来记录各个中断。无论 GIE、PEIE 和各个中断允许位的状态如何，中断标志位都会在中断发生时置 1。

当 GIE 位置 1 时，中断事件的发生会引发以下事件：

- 清除当前预取的指令
- GIE 位清零
- 当前程序计数器 （PC）值被压入堆栈
- 将中断向量 0004h 装入 PC

中断服务程序（Interrupt Service Routine， ISR）中的固件应该通过查询中断标志位来确定中断源。在退出 ISR 之前必须将中断标志位清零，以避免重复的中断。由于 GIE 位被清零，所以执行 ISR 期间发生的任何中断将会通过其中断标志进行记录，但是不会使处理器重定向到中断向量。

`RETFIE` 指令通过将先前地址从堆栈中弹出并将 GIE 位置 1 来退出 ISR。

关于特殊中断操作的更多信息，请参见其相应的外设章节。

注	1：	无论任何其他中断允许位的状态如何，各个中断标志位都会在中断发生时置 1。
	2：	当 GIE 位清零时，忽略所有中断。GIE 位清零时发生的任何中断在 GIE 位再次置 1 时得到处理。

6.2 中断延时

中断延时定义为从发生中断事件到开始执行中断向量处的代码所经过的时间。同步中断的延时为 3 或 4 个指令周期。对于异步中断，延时为 3 至 5 个指令周期，具体取决于中断发生的时间。更多详细信息，请参见图 6-2 和**第 6.3 节 “休眠期间的中断”**。

© 2011-2014 Microchip Technology Inc.

图 6-2： 中断响应延时

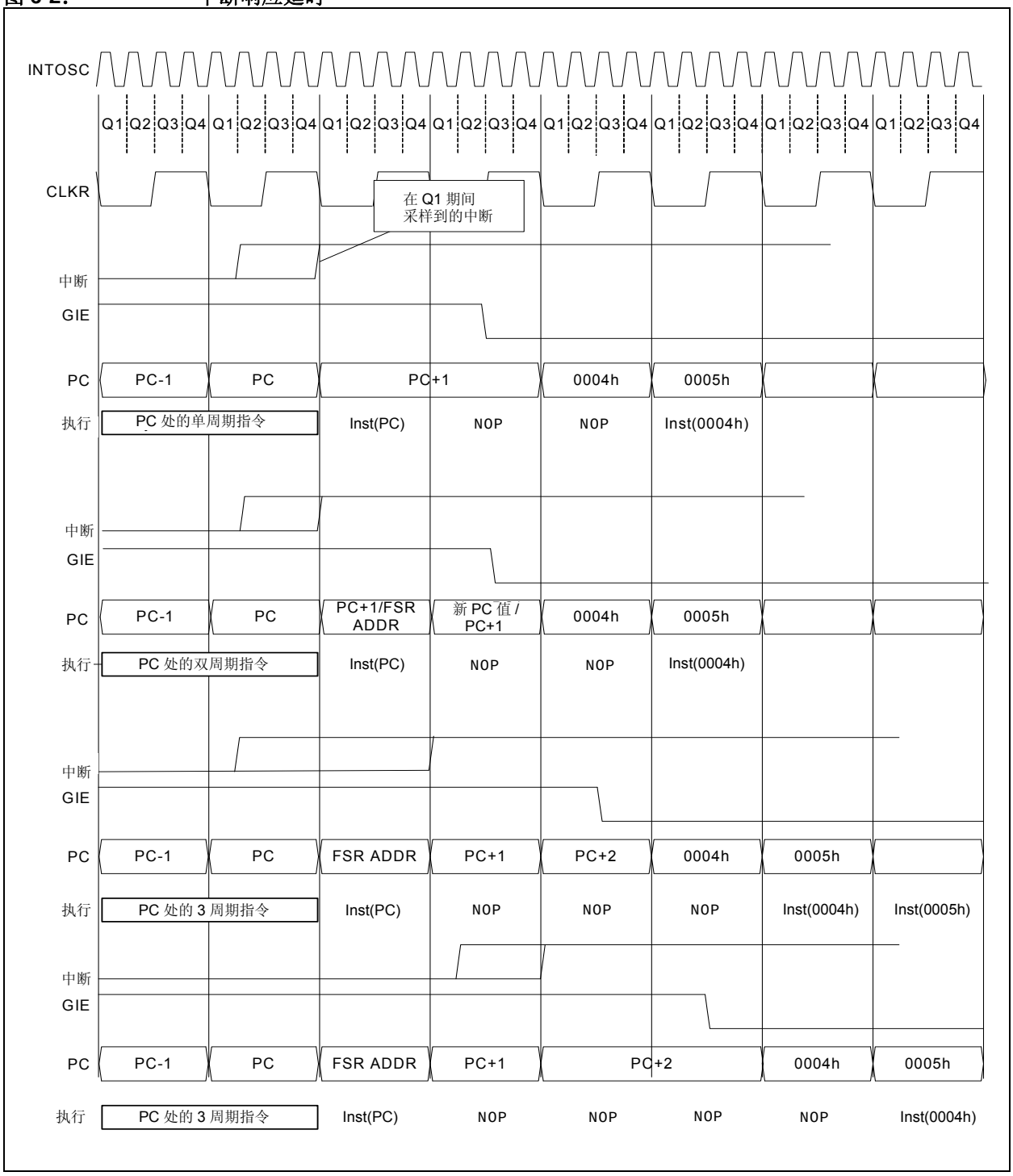
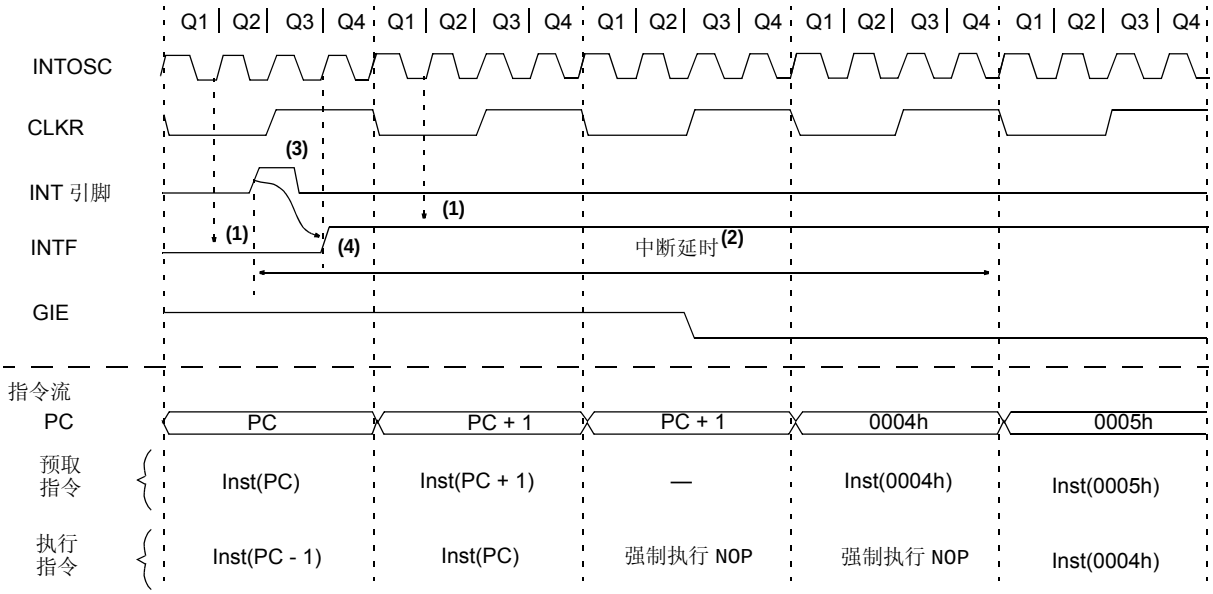

INTOSC
Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
CLKR
在 Q1 期间
采样到的中断
中断
GIE
PC
PC-1
PC
PC+1
0004h
0005h
执行
PC 处的单周期指令
Inst(PC)
NOP
NOP
Inst(0004h)
中断
GIE
PC
PC-1
PC
PC+1/FSR ADDR
新 PC 值 / PC+1
0004h
0005h
执行
PC 处的双周期指令
Inst(PC)
NOP
NOP
Inst(0004h)
中断
GIE
PC
PC-1
PC
FSR ADDR
PC+1
PC+2
0004h
0005h
执行
PC 处的 3 周期指令
Inst(PC)
NOP
NOP
NOP
Inst(0004h)
Inst(0005h)
中断
GIE
PC
PC-1
PC
FSR ADDR
PC+1
PC+2
0004h
0005h
执行
PC 处的 3 周期指令
Inst(PC)
NOP
NOP
NOP
NOP
Inst(0004h)

图 6-3： INT 引脚中断时序



注 1： 在此采样 INTF 标志 （每个 Q1 周期）。

2： 异步中断延时 = 3-5 TCY。同步中断延时 = 3-4 TCY，其中 TCY = 指令周期时间。不管 Inst(PC) 是单周期还是双周期指令，中断延时都是一样的。

3： 关于 INT 脉冲的最小宽度，请参见第 24.0 节 “电气规范”中的交流规范。

4： 允许在 Q1-Q4 周期内的任何时间将 INTF 置 1。

© 2011-2014 Microchip Technology Inc.

6.3 休眠期间的中断

有些中断可用于将器件从休眠模式中唤醒。要将器件从休眠模式中唤醒，相应外设必须能够在无系统时钟的条件下工作。进入休眠模式之前，必须将相应中断源的中断允许位置 1。

当处理器从休眠模式中唤醒时，如果 GIE 位也被置 1，则处理器跳转到中断向量。否则，处理器将继续执行 SLEEP 指令后的指令。在跳转到 ISR 之前，始终会执行紧接着 SLEEP 指令的指令。更多详细信息，请参见**第 7.0 节 “掉电模式 （休眠）”**。

6.4 INT 引脚

INT 引脚用于产生异步边沿触发中断。将 INTCON 寄存器的 INTE 位置 1 可允许这种中断。OPTION_REG 寄存器的 INTEDG 位可确定在哪个边沿发生中断。当 INTEDG 位置 1 时，上升沿触发中断。当 INTEDG 位清零时，下降沿触发中断。当 INT 引脚上出现有效边沿时，INTCON 寄存器的 INTF 位置 1。如果 GIE 和 INTE 位也置 1，则处理器将程序执行重定向到中断向量。

6.5 中断期间的现场保护

中断期间，仅将返回的 PC 值保存在堆栈中。通常，用户可能想要在中断期间保存主要寄存器（例如，W 寄存器和 STATUS 寄存器）。这必须由软件实现。

临时保存寄存器 W_TEMP 和 STATUS_TEMP 应位于 GPR 的最后 16 字节 （见表 1-2）。这使得现场保护和恢复操作更简单。例 6-1 给出的代码可用于：

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

注：	这些器件不需要保存 PCLATH。但是，如果 ISR 和主代码均使用了计算 GOTO 指令，则必须在 ISR 中保存和恢复 PCLATH。

例 6-1： 将 STATUS 寄存器和 W 寄存器的内容保存在 RAM 中

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

6.6 中断控制寄存器

寄存器 6-1： INTCON：中断控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0
GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF(1)
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 **GIE**：全局中断允许位
1 = 允许所有有效中断
0 = 禁止所有中断

bit 6 **PEIE**：外设中断允许位
1 = 允许所有有效外设中断
0 = 禁止所有外设中断

bit 5 **TMR0IE**：Timer0 溢出中断允许位
1 = 允许 Timer0 中断
0 = 禁止 Timer0 中断

bit 4 **INTE**：INT 外部中断允许位
1 = 允许 INT 外部中断
0 = 禁止 INT 外部中断

bit 3 **IOCIE**：电平变化中断允许位
1 = 允许电平变化中断
0 = 禁止电平变化中断

bit 2 **TMR0IF**：Timer0 溢出中断标志位
1 = TMR0 寄存器已溢出
0 = TMR0 寄存器未溢出

bit 1 **INTF**：INT 外部中断标志位
1 = 发生了 INT 外部中断
0 = 未发生 INT 外部中断

bit 0 **IOCIF**：电平变化中断的中断标志位 (1)
1 = 至少一个电平变化中断引脚的状态发生改变
0 = 没有一个电平变化中断引脚的状态发生改变

注 1： IOCIF 标志位是只读位，在 IOCBF 寄存器中的所有电平变化中断标志都由软件清零后，该位被清零。

> **注：** 当发生中断条件时，无论相应的中断允许位或全局中断允许位（INTCON 寄存器的 GIE）的状态如何，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。

© 2011-2014 Microchip Technology Inc.

寄存器 6-2： PIE1：外设中断允许寄存器 1

U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0
—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 **未实现：**读为 0

bit 6 **ADIE：**A/D 转换器中断允许位
1 = 允许 A/D 转换器中断
0 = 禁止 A/D 转换器中断

bit 5 **未实现：**读为 0

bit 4 **NCO1IE：**数控振荡器中断允许位
1 = 允许 NCO 上溢中断
0 = 禁止 NCO 上溢中断

bit 3 **CLC1IE：**可配置逻辑块中断允许位
1 = 允许 CLC 中断
0 = 禁止 CLC 中断

bit 2 **未实现：**读为 0

bit 1 **TMR2IE：**TMR2 与 PR2 匹配中断允许位
1 = 允许 TMR2 与 PR2 匹配中断
0 = 禁止 TMR2 与 PR2 匹配中断

bit 0 **未实现：**读为 0

注： 要允许任何一个外设中断，都必须将 INTCON 寄存器中的 PEIE 位置 1。

寄存器 6-3： PIR1：外设中断请求寄存器 1

U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0
—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 **未实现：**读为 0

bit 6 **ADIF：**A/D 转换器中断标志位
1 = 完成了 A/D 转换
0 = 未完成 A/D 转换

bit 5 **未实现：**读为 0

bit 4 **NCO1IF：**数控振荡器中断标志位
1 = 发生了 NCO1 上溢 （必须由软件清零）
0 = 未发生 NCO1 上溢

bit 3 **CLC1IF：**可配置逻辑块上升沿中断标志位
1 = 发生了 CLC 中断 （必须由软件清零）
0 = 未发生 CLC 中断

bit 2 **未实现：**读为 0

bit 1 **TMR2IF：**TMR2 与 PR2 匹配中断标志位
1 = 发生了 TMR2 与 PR2 匹配 （必须由软件清零）
0 = 未发生 TMR2 与 PR2 匹配
注：发生匹配的次数必须是 TMR2 后分频器 （寄存器 17-1）指定的次数。

bit 0 **未实现：**读为 0

> **注：** 当发生中断条件时，无论相应的中断允许位或全局中断允许位 （INTCON 寄存器的 GIE）的状态如何，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。

© 2011-2014 Microchip Technology Inc.

表 6-1： 与中断相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
IOCAF	—	—	—	—	IOCAF3	IOCAF2	IOCAF1	IOCAF0	76
IOCAN	—	—	—	—	IOCAN3	IOCAN2	IOCAN1	IOCAN0	75
IOCAP	—	—	—	—	IOCAP3	IOCAP2	IOCAP1	IOCAP0	75
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	T0CS	T0SE	PSA	PS<2:0>			95
PIE1	—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—	41
PIR1	—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—	42

图注： — = 未实现单元，读为 0。中断不使用阴影单元。

7.0 掉电模式 （休眠）

器件通过执行 SLEEP 指令进入掉电模式。

进入休眠模式时，存在以下条件：

1. WDT 将清零，但是保持运行 （如果使能了在休眠期间工作）。
2. STATUS 寄存器的 $\overline{PD}$ 位清零。
3. STATUS 寄存器的 $\overline{TO}$ 位置 1。
4. 禁止 CPU 时钟。
5. 31 kHz LFINTOSC 不受影响，并且由 31 kHz LFINTOSC提供时钟的外设可以在休眠模式下继续工作。
6. ADC 不受影响 （如果选择了专用 FRC 时钟）。
7. I/O 端口保持执行 SLEEP 指令之前的状态（驱动为高电平、低电平或高阻态）。
8. WDT 之外的复位不受休眠模式影响。

关于外设在休眠期间工作的更多详细信息，请参见各个章节。

要最大程度地降低电流消耗，应考虑以下条件：

- I/O 引脚不应悬空
- 外部电路从 I/O 引脚的灌电流
- 内部电路从 I/O 引脚的拉电流
- 电流汲取自带有内部弱上拉的引脚
- 模块使用 31 kHz LFINTOSC
- CWG 和 NCO 模块使用 HFINTOSC

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 VDD 或 VSS。

可以拉电流的内部电路示例包括FVR模块。关于这些模块的更多信息，请参见**第 12.0 节 “固定参考电压（FVR）”**。

7.1 从休眠模式中唤醒

可以通过以下任一事件将器件从休眠状态唤醒：

1. $\overline{MCLR}$ 引脚上的外部复位输入 （如果使能）
2. BOR 复位 （如果使能）
3. POR 复位
4. 看门狗定时器 （如果使能）
5. 任何外部中断
6. 能够在休眠期间运行的外设产生的中断（更多信息，请参见各个外设）。

前 3 个事件会使器件复位。后 3 个事件认为是程序执行的延续。要确定是发生器件复位还是唤醒事件，请参见**第 5.10 节 “确定复位原因”**。

当执行 SLEEP 指令时，下一条指令 （PC + 1）被预先取出。如果希望通过中断事件唤醒器件，则必须允许相应的中断允许位。唤醒与 GIE 位的状态无关。如果禁止 GIE 位，则器件将继续执行 SLEEP 指令后的指令。如果使能 GIE 位，则器件先执行 SLEEP 指令后的指令，然后调用中断服务程序。如果不想执行SLEEP指令后的指令，用户应该在 SLEEP 指令后面放置一条 NOP 指令。

当器件从休眠模式中唤醒时，无论唤醒的原因如何，WDT 将清零。

互补波形发生器 （CWG）和数控振荡器 （NCO）模块可使用 HFINTOSC 振荡器作为各自的时钟源。在某些情况下，选择 HFINTOSC 用于 CWG 或 NCO 模块时，HFINTOSC 将在休眠期间保持有效状态。这将直接影响休眠模式下的电流。更多信息，请参见**第 21.0 节 “互补波形发生器 （CWG）模块”**和**第 20.0 节 “数控振荡器 （NCO）模块”**。

© 2011-2014 Microchip Technology Inc.

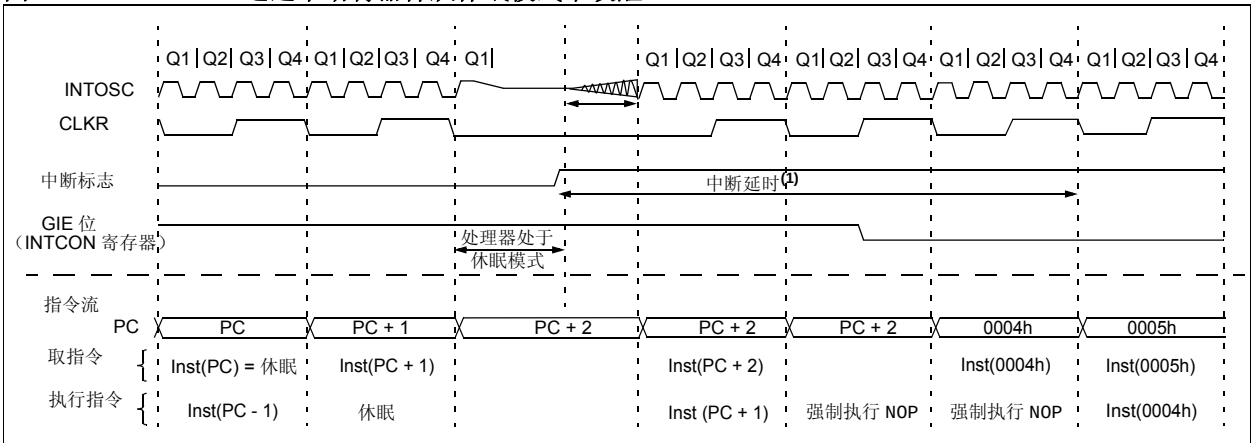
7.1.1 使用中断唤醒

当禁止全局中断 （GIE 清零），并且任一中断源的中断允许位和中断标志位均置 1 时，将发生以下事件之一：

- 如果在执行 SLEEP 指令**之前**发生中断
  - SLEEP 指令将作为 NOP 执行
  - WDT 和 WDT 预分频器不会清零
  - STATUS 寄存器的 $\overline{TO}$ 位不会置 1
  - STATUS 寄存器的 $\overline{PD}$ 位不会清零
- 如果在执行 SLEEP 指令**期间或之后**发生中断
  - SLEEP 指令将完全执行
  - 器件将立即从休眠模式中唤醒
  - WDT 和 WDT 预分频器将清零
  - STATUS 寄存器的 $\overline{TO}$ 位将置 1
  - STATUS 寄存器的 $\overline{PD}$ 位将清零

即使在执行 SLEEP 指令之前检查到标志位为 0，但是在执行完 SLEEP 指令之前标志位可能置 1。要确定是否执行了 SLEEP 指令，可以测试 $\overline{PD}$ 位。如果 $\overline{PD}$ 位置 1，则说明 SLEEP 指令被作为一条 NOP 指令执行了。

图 7-1： 通过中断将器件从休眠模式中唤醒



注 1： 假定 GIE = 1。这种情况下，处理器被唤醒后，将调用 0004h 处的 ISR。如果 GIE = 0，程序继续执行。

表 7-1： 与掉电模式相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	13
WDTCON	—	—	WDTPS<4:0>					SWDTEN	48

图注： — = 未实现单元，读为 0。掉电模式不使用阴影单元。

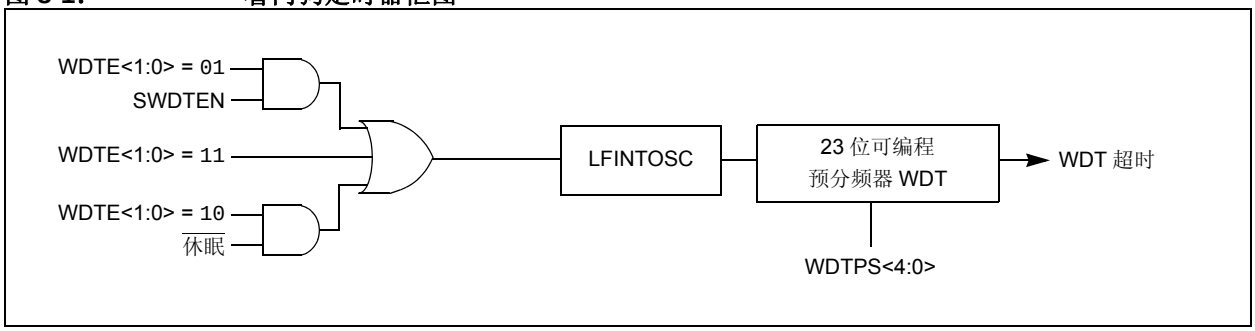
8.0 看门狗定时器

看门狗定时器是系统定时器，如果固件在超时周期内未发出 CLRWDT 指令，看门狗定时器将产生复位。看门狗定时器通常用于将系统从意外事件中恢复。

WDT 具有如下特性：

- 独立的时钟源
- 多种工作模式
  - WDT 始终开启
  - WDT 在休眠时关闭
  - WDT 由软件控制
  - WDT 始终关闭
- 可配置的典型超时周期为 1 ms 至 256s （典型值）
- 多个复位条件
- 休眠期间的工作原理

图 8-1： 看门狗定时器框图



© 2011-2014 Microchip Technology Inc.

8.1 独立时钟源

WDT 以 31 kHz LFINTOSC 的内部振荡器作为其时基。本章中的时间间隔基于 1 ms 的标称时间间隔。关于 LFINTOSC 的容差，请参见第 24.0 节 “电气规范”。

8.2 WDT 工作模式

看门狗定时器模块有 4 种工作模式，由配置字的 WDTE<1:0> 位控制。请参见表 8-1。

8.2.1 WDT 始终开启

当配置字的 WDTE 位设置为 11 时， WDT 始终开启。

WDT 保护功能在休眠期间有效。

8.2.2 WDT 在休眠时关闭

当配置字的 WDTE 位设置为 10 时，除了 WDT 在休眠时关闭之外， WDT 一直开启。

WDT 保护功能在休眠期间无效。

8.2.3 WDT 由软件控制

当配置字的 WDTE 位设置为 01 时，WDT 由 WDTCON 寄存器的 SWDTEN 位控制。

WDT 保护功能不受休眠影响。更多详细信息，请参见表 8-1。

表 8-1： WDT 工作模式

WDTE<1:0>	SWDTEN	器件模式	WDT 模式
11	X	X	有效
10	X	唤醒	有效
		休眠	禁止
01	1	X	有效
	0		禁止
00	X	X	禁止

8.3 超时周期

WDTCON 寄存器的 WDTPS 位可以设置从 1 ms 至 256s（标称值）的超时周期。复位后，默认超时周期为 2s。

8.4 清零 WDT

发生以下任何条件时， WDT 清零：

- 任何复位
- 执行 CLRWDT 指令
- 器件进入休眠模式
- 器件从休眠模式中唤醒
- 振荡器故障
- WDT 禁止

更多信息，请参见表 8-2。

8.5 休眠期间的工作原理

器件进入休眠模式时， WDT 清零。如果在休眠期间使能 WDT，则 WDT 将恢复计数。

器件退出休眠模式时， WDT 再次清零。

如果在器件处于休眠模式时发生 WDT 超时，不会产生复位，而是将器件唤醒并继续工作。STATUS 寄存器中的 $\overline{TO}$ 和 $\overline{PD}$ 位会更改，以指示此事件。更多信息，请参见第 2.0 节 “存储器构成” 和寄存器 2-1。

表 8-2： WDT 清零条件

条件	WDT
WDTE<1:0> = 00	清零
WDTE<1:0> = 01 且 SWDTEN = 0	
WDTE<1:0> = 10 且进入休眠模式	
CLRWDT 命令	
退出休眠	
更改 INTOSC 分频比 （IRCF 位）	不受影响

8.6 看门狗控制寄存器

寄存器 8-1： WDTCON：看门狗定时器控制寄存器

U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-1/1	R/W-0/0
—	—	WDTPS<4:0>					SWDTEN
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-6 **未实现：**读为 0

bit 5-1 **WDTPS<4:0>：**看门狗定时器周期选择位 (1)

位值 = 预分频比

11111 = 保留。产生最小间隔 （1:32）

•
•
•

10011 = 保留。产生最小间隔 （1:32）

10010 = 1:8388608 （2^{23}）（标称间隔 256s）
10001 = 1:4194304 （2^{22}）（标称间隔 128s）
10000 = 1:2097152 （2^{21}）（标称间隔 64s）
01111 = 1:1048576 （2^{20}）（标称间隔 32s）
01110 = 1:524288 （2^{19}）（标称间隔 16s）
01101 = 1:262144 （2^{18}）（标称间隔 8s）
01100 = 1:131072 （2^{17}）（标称间隔 4s）
01011 = 1:65536 （标称间隔 2s）（复位值）
01010 = 1:32768 （标称间隔 1s）
01001 = 1:16384 （标称间隔 512 ms）
01000 = 1:8192 （标称间隔 256 ms）
00111 = 1:4096 （标称间隔 128 ms）
00110 = 1:2048 （标称间隔 64 ms）
00101 = 1:1024 （标称间隔 32 ms）
00100 = 1:512 （标称间隔 16 ms）
00011 = 1:256 （标称间隔 8 ms）
00010 = 1:128 （标称间隔 4 ms）
00001 = 1:64 （标称间隔 2 ms）
00000 = 1:32 （标称间隔 1 ms）

bit 0 **SWDTEN：**看门狗定时器的软件使能 / 禁止位

如果 WDTE<1:0> = 00：
忽略该位。
如果 WDTE<1:0> = 01：
1 = WDT 开启
0 = WDT 关闭
如果 WDTE<1:0> = 1x：
忽略该位。

注 1： 时间均为近似值。 WDT 时间基于 31 kHz LFINTOSC。

© 2011-2014 Microchip Technology Inc.

表 8-3： 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	寄存器所在页
OSCCON	—	IRCF<2:0>			HFIOFR	—	LFIOFR	HFIOFS	26
STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	13
WDTCON	—	—	WDTPS<4:0>					SWDTEN	48

图注： x = 未知， u = 不变， – = 未实现单元，读为 0。看门狗定时器不使用阴影单元。

表 8-4： 与看门狗定时器相关的配置字汇总

名称	Bit	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG	13:8	—	—	—	WRT<1:0>		BORV	LPBOR	LVP	20
	7:0	$\overline{CP}$	MCLRE	$\overline{PWRTE}$	WDTE<1:0>		BOREN<1:0>		FOSC	

图注： — = 未实现单元，读为 0。看门狗定时器不使用阴影单元。

9.0 闪存程序存储器控制

在整个 VDD 范围内的正常工作期间，闪存程序存储器是可读写的。程序存储器通过特殊功能寄存器（SFR）进行间接寻址。用于访问程序存储器的 SFR 包括：

- PMCON1
- PMCON2
- PMDATL
- PMDATH
- PMADRL
- PMADRH

当访问程序存储器时，PMDATH:PMDATL 寄存器对组成双字节字，保存要读 / 写的 14 位数据，而 PMADRH:PMADRL 寄存器组成双字节字，保存被读取的程序存储单元的 9 位地址。

写入时间由片上定时器控制。写入和擦除电压由片上电荷泵产生，该电荷泵可以在器件工作电压范围内正常工作。

可通过两种方式保护闪存程序存储器：代码保护（配置字中的 $\overline{CP}$ 位）和写保护（配置字中的 WRT<1:0> 位）。

代码保护（$\overline{CP}$ = 0）(1) 禁止通过外部器件编程器对闪存程序存储器进行访问和读写。代码保护不会影响自写和擦除功能。仅可通过器件编程器对该器件执行块擦除，并清零整个闪存程序存储器、配置位以及用户 ID 来复位代码保护。

写保护禁止对部分或整个闪存程序存储器（由 WRT<1:0> 位定义）的自写和擦除操作。写保护不会影响器件编程器读写或擦除器件的功能。

> 注 1：通过将配置字的 $\overline{CP}$ 位清零，可以使能对整个闪存程序存储器阵列的代码保护。

9.1 PMADRL 和 PMADRH 寄存器

PMADRH:PMADRL 寄存器对可以寻址最大 512 字的程序存储器。当选择程序地址值时，地址的 MSB 写入 PMADRH 寄存器，而 LSB 写入 PMADRL 寄存器。

9.1.1 PMCON1 和 PMCON2 寄存器

PMCON1 是访问闪存程序存储器的控制寄存器。

控制位 RD 和 WR 分别启动读和写操作。用软件只能将这些位置 1 而无法清零。在完成读或写操作后，由硬件将它们清零。由于无法用软件将 WR 位清零，可避免因意外而过早地终止写操作。

当 WREN 位置 1 时，允许执行写操作。上电时，WREN 位被清零。在正常运行期间，当写操作被复位中断时，WRERR 位置 1。在这些情况下，复位后用户可以检查 WRERR 位并执行相应的错误处理程序。

PMCON2 寄存器是只写寄存器。试图读 PMCON2 寄存器将返回全 0。

要使能对程序存储器的写操作，必须将特定模式（解锁序列）写入 PMCON2 寄存器。所需的解锁序列可防止意外写入程序存储器写锁存器和闪存程序存储器。

9.2 闪存程序存储器概述

要进行擦除和编程操作，了解闪存程序存储器结构是非常重要的。闪存程序存储器按行排列。每行包括固定数量的 14 位程序存储器字。行是用户软件可擦除的最小大小。

擦除某行后，用户可以对该行的部分或全部内容进行重新编程。写入程序存储器行的数据将被写入 14 位宽的数据写锁存器中。用户不能直接访问这些写锁存器，但是可以通过对 PMDATH:PMDATL 寄存器对的连续写入来装入写锁存器的内容。

> 注：如果用户只希望修改先前已编程行的部分内容，则必须在擦除之前先读取整行内容，并保存到 RAM 中。然后，可以将新数据和已保存数据写入写锁存器，以对闪存程序存储器行进行再编程。但是，对于任何未经过编程的单元，则无需先擦除行即可写入。这种情况下，不需要保存并重新写入其他先前已编程的单元。

关于闪存程序存储器的擦除行大小和写锁存器数，请参见表 9-1。

© 2011-2014 Microchip Technology Inc.

表 9-1： 闪存存储器构成 （按器件）

器件	行擦除（字）	写锁存器（字）
PIC10(L)F320	16	16
PIC10(L)F322		

9.2.1 读闪存程序存储器

要读程序存储单元，用户必须：

1. 将所需的地址写入PMADRH:PMADRL寄存器对。
2. 将 PMCON1 寄存器的 CFGS 位清零。
3. 然后，将 PMCON1 寄存器的控制位 RD 置 1。

一旦将读控制位置1，闪存程序存储器控制器将使用第二个指令周期读取数据。这会导致紧跟 `BSF PMCON1,RD` 指令后的第二条指令被忽略。在紧接着的下一个周期，PMDATH:PMDATL寄存器对中的数据可供使用，因此可在随后的指令中读取为两个字节。

PMDATH:PMDATL 寄存器对将保存该值，直到开始下一次读操作或用户向该单元写入数据。

> **注：** 程序存储器读操作后的两条指令必须为 `NOP`。这可以防止用户在 RD 位置 1 后的下一条指令执行双周期指令。

图 9-1： 闪存程序存储器读操作流程图

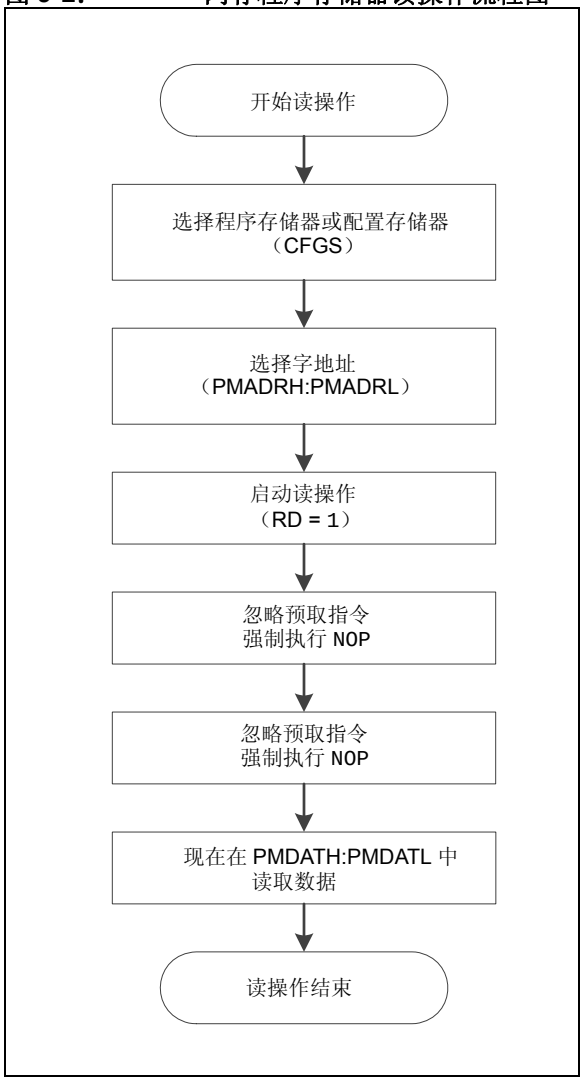
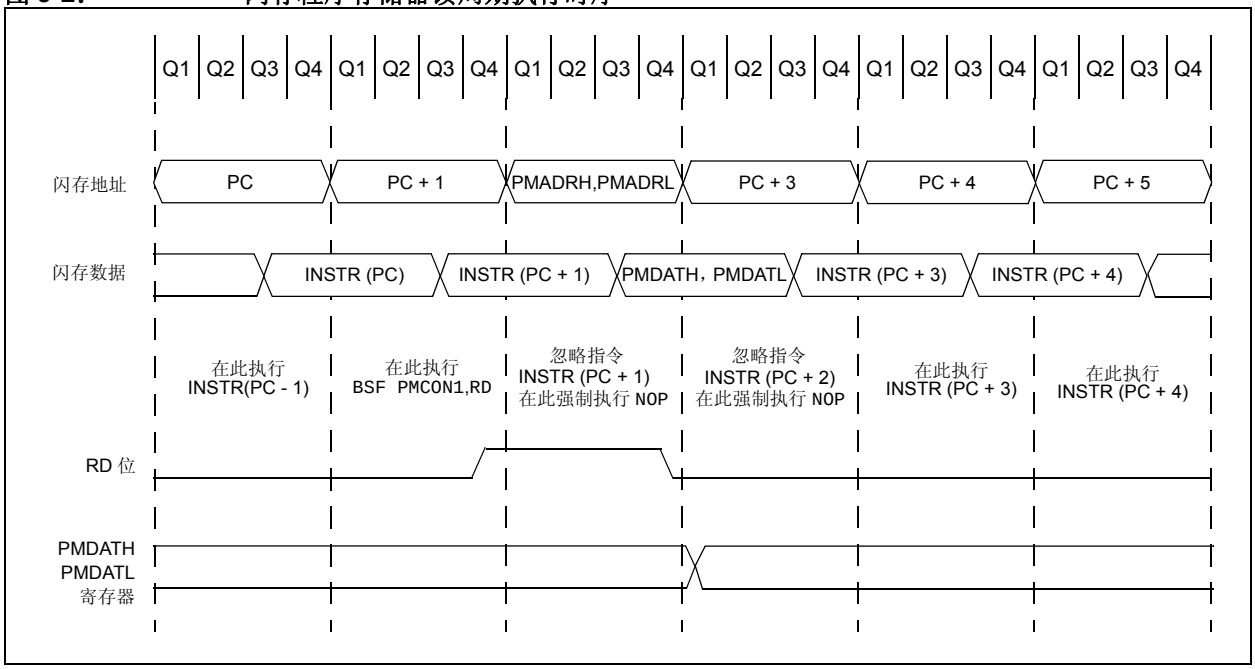


图 9-2： 闪存程序存储器读周期执行时序



例 9-1： 读闪存程序存储器

```
* This code block will read 1 word of program
* memory at the memory address:
    PROG_ADDR_HI: PROG_ADDR_LO
*   data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

    BANKSEL  PMADRL          ; not required on devices with 1 Bank of SFRs
    MOVLW    PROG_ADDR_LO    ;
    MOVWF    PMADRL          ; Store LSB of address
    MOVLW    PROG_ADDR_HI    ;
    MOVWF    PMADRH          ; Store MSB of address

    BCF      PMCON1,CFGS     ; Do not select Configuration Space
    BSF      PMCON1,RD       ; Initiate read
    NOP                      ; Ignored (Figure 9-2)
    NOP                      ; Ignored (Figure 9-2)

    MOVF     PMDATL,W        ; Get LSB of word
    MOVWF    PROG_DATA_LO    ; Store in user location
    MOVF     PMDATH,W        ; Get MSB of word
    MOVWF    PROG_DATA_HI    ; Store in user location
```


© 2011-2014 Microchip Technology Inc.

9.2.2 闪存存储器解锁序列

注：	在执行闪存解锁序列之前、上电复位（POR）之后，需要至少 100 μs 延时。

解锁序列是一种防止意外自写编程或擦除闪存程序存储器的保护机制。该序列必须在无中断的情况下执行并完成，以成功完成以下任一操作：

- 行擦除
- 装入程序存储器写锁存器
- 程序存储器写锁存器到程序存储器的写操作
- 程序存储器写锁存器到用户 ID 的写操作

解锁序列的步骤如下所示：

1. 将 55h 写入 PMCON2
2. 将 AAh 写入 PMCON2
3. 将 PMCON1 中的 WR 位置 1
4. NOP 指令
5. NOP 指令

一旦 WR 位置 1，处理器通常强制执行两条 NOP 指令。当执行擦除行或编程行操作时，处理器将暂停内部操作（典型值为 2 ms）直至该操作完成，然后继续执行下一条指令。当该操作正在装入程序存储器写锁存器时，处理器通常强制执行两条 NOP 指令并继续执行下一条指令。

由于解锁序列不能中断，因此在执行解锁序列之前应该禁止全局中断，并在完成解锁序列之后重新允许全局中断。

图 9-3：闪存程序存储器解锁序列流程图

9.2.3 擦除闪存程序存储器

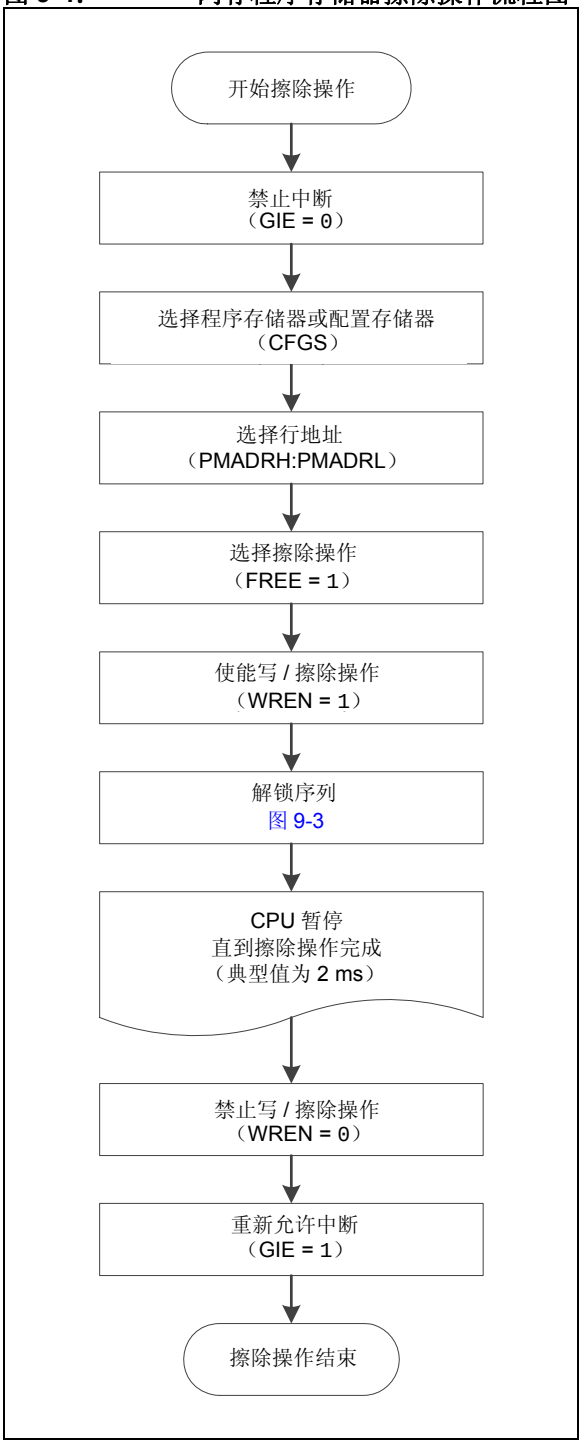
当执行代码时，程序存储器只能按行擦除。要擦除某行，请执行以下操作：

1. 将要擦除的行中的任何地址装入 PMADRH:PMADRL 寄存器对。
2. 将 PMCON1 寄存器的 CFGS 位清零。
3. 将 PMCON1 寄存器的 FREE 和 WREN 位置 1。
4. 依次将 55h 和 AAh 写入 PMCON2（闪存编程解锁序列）。
5. 将 PMCON1 寄存器的控制位 WR 置 1，以开始擦除操作。

请参见例 9-2。

在 `BSF PMCON1,WR` 指令后，处理器需要两个周期来设置擦除操作。用户必须在 WR 位置 1 后，执行两条 `NOP` 指令。处理器将暂停内部操作，通常为 2 ms 擦除时间。这不是休眠模式，因为时钟和外设将继续运行。擦除周期后，处理器从 PMCON1 写指令后的第三条指令继续执行。

图 9-4：闪存程序存储器擦除操作流程图



© 2011-2014 Microchip Technology Inc.

例 9-2： 擦除程序存储器的一行

```
; This row erase routine assumes the following:
; 1. A valid address within the erase row is loaded in ADDRH:ADDRL
; 2. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)


        BCF         INTCON,GIE     ; Disable ints so required sequences will execute properly
        BANKSEL     PMADRL         ; not required on devices with 1 Bank of SFRs
        MOVF        ADDRL,W        ; Load lower 8 bits of erase address boundary
        MOVWF       PMADRL
        MOVF        ADDRH,W        ; Load upper 6 bits of erase address boundary
        MOVWF       PMADRH
        BCF         PMCON1,CFGS    ; Not configuration space
        BSF         PMCON1,FREE    ; Specify an erase operation
        BSF         PMCON1,WREN    ; Enable writes

必需的序列  MOVLW       55h            ; Start of required sequence to initiate erase
        MOVWF       PMCON2         ; Write 55h
        MOVLW       0AAh           ;
        MOVWF       PMCON2         ; Write AAh
        BSF         PMCON1,WR      ; Set WR bit to begin erase
        NOP                        ; NOP instructions are forced as processor starts
        NOP                        ; row erase of program memory.
                                   ;
                                   ; The processor stalls until the erase process is complete
                                   ; after erase processor continues with 3rd instruction

        BCF         PMCON1,WREN    ; Disable writes
        BSF         INTCON,GIE     ; Enable interrupts
```

9.2.4 写闪存程序存储器

使用以下步骤编程程序存储器：

1. 将要编程的行地址装入 PMADRH:PMADRL。
2. 将数据装入每个写锁存器。
3. 启动编程操作。
4. 重复步骤 1 至 3，直到写入所有数据。

写入程序存储器之前，要写入的字必须已擦除或者以前未写入。程序存储器每次只能擦除一行。启动写操作时不会自动擦除。

程序存储器可以每次写入一个或多个字。每次写入的最多字数等于写锁存器数。更多详细信息，请参见图 9-5（使用 16 个写锁存器对程序存储器进行行写操作）。

写锁存器对齐到由 PMADRH:PMADRL 的高 10 位（PMADRH<6:0>:PMADRL<7:5>）定义的闪存行地址边界，而 PMADRL 的低 5 位（PMADRL<4:0>）确定装入的写锁存器。写操作不会跨越这些边界。当程序存储器写操作完成时，写锁存器中的数据复位为包含 0x3FFF。

要装入写锁存器和编程程序存储行，必须完成以下步骤。这些步骤可分为两个部分。首先，使用解锁序列（LWLO = 1）将 PMDATH:PMDATL 中的数据装入每个写锁存器。当要装入写锁存器的最后一个字准备就绪时，清零 LWLO 位并执行解锁序列。这将启动编程操作，将所有锁存器写入闪存程序存储器。

> **注：** 必须执行特殊的解锁序列，才能将数据装入写锁存器或启动闪存编程操作。如果解锁序列中断，将不会启动对锁存器或程序存储器的写操作。

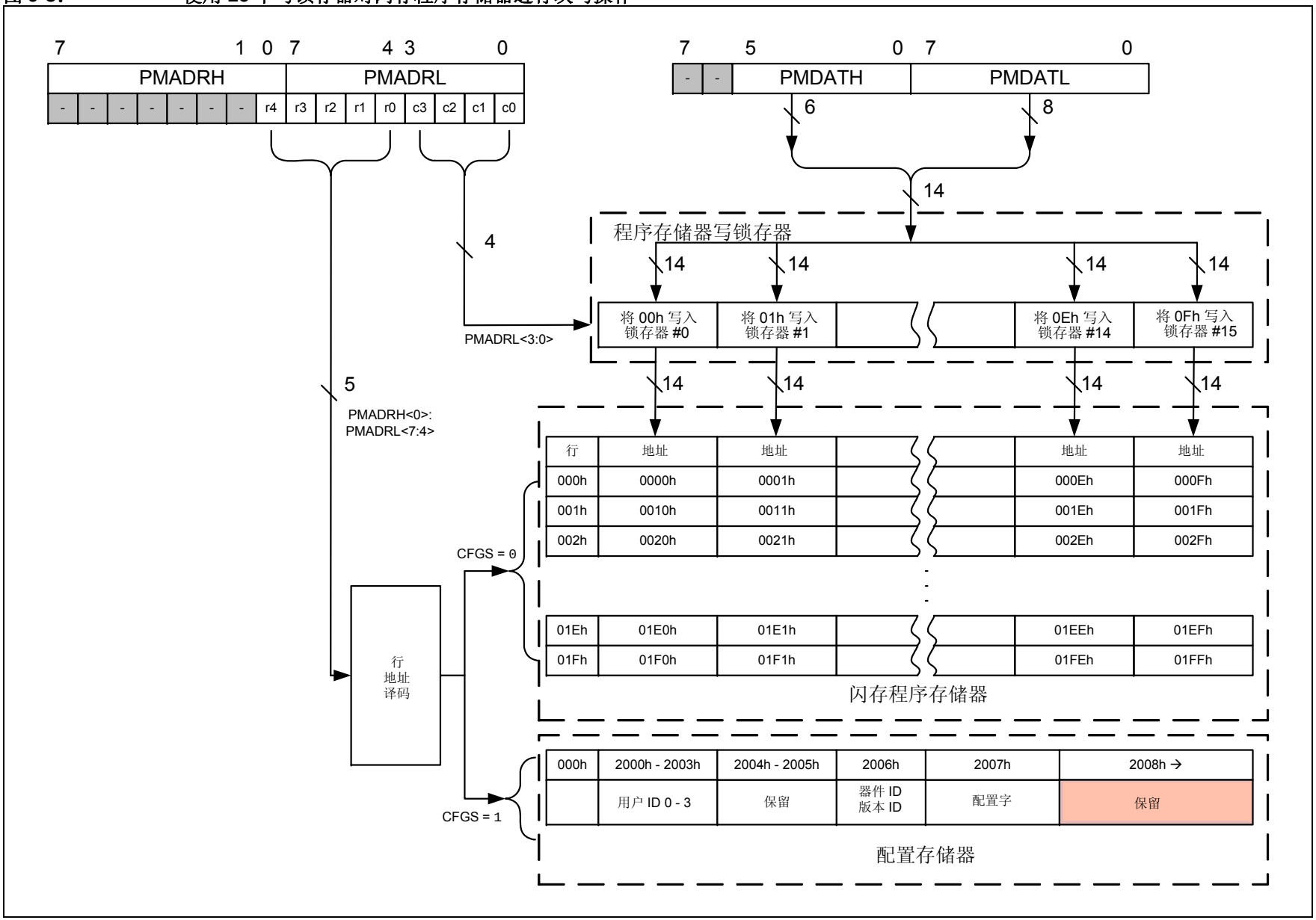
1. 将 PMCON1 寄存器的 WREN 位置 1。
2. 将 PMCON1 寄存器的 CFGS 位清零。
3. 将PMCON1寄存器的LWLO位置1。当PMCON1寄存器的 LWLO 位为 1 时，写序列只能装入写锁存器，不会启动对闪存程序存储器的写操作。
4. 将要写入的单元地址装入PMADRH:PMADRL寄存器对。
5. 将要写入的程序存储器数据装入 PMDATH:PMDATL 寄存器对。
6. 执行解锁序列（**第 9.2.2 “闪存存储器解锁序列”**）。现在装入写锁存器。
7. 将 PMADRH:PMADRL 寄存器对的内容加 1，以指向下一个单元。
8. 重复步骤 5 至 7，直到除最后一个写锁存器以外的所有写锁存器都已装入完毕为止。
9. 将 PMCON1 寄存器的 LWLO 位清零。当 PMCON1 寄存器的 LWLO 位为 0 时，写序列将启动对闪存程序存储器的写操作。
10. 将要写入的程序存储器数据装入 PMDATH:PMDATL 寄存器对。
11. 执行解锁序列（**第 9.2.2 “闪存存储器解锁序列”**）。现在可将整个程序存储器锁存器的内容写入闪存程序存储器。

> **注：** 当每次完成写或擦除操作时，程序存储器写锁存器复位为空白状态（0x3FFF）。因此，不需要装入全部程序存储器写锁存器。未装入的锁存器将处于空白状态。

例 9-3 给出了完整的写序列示例。将初始地址装入 PMADRH:PMADRL寄存器对。数据通过间接寻址装入。

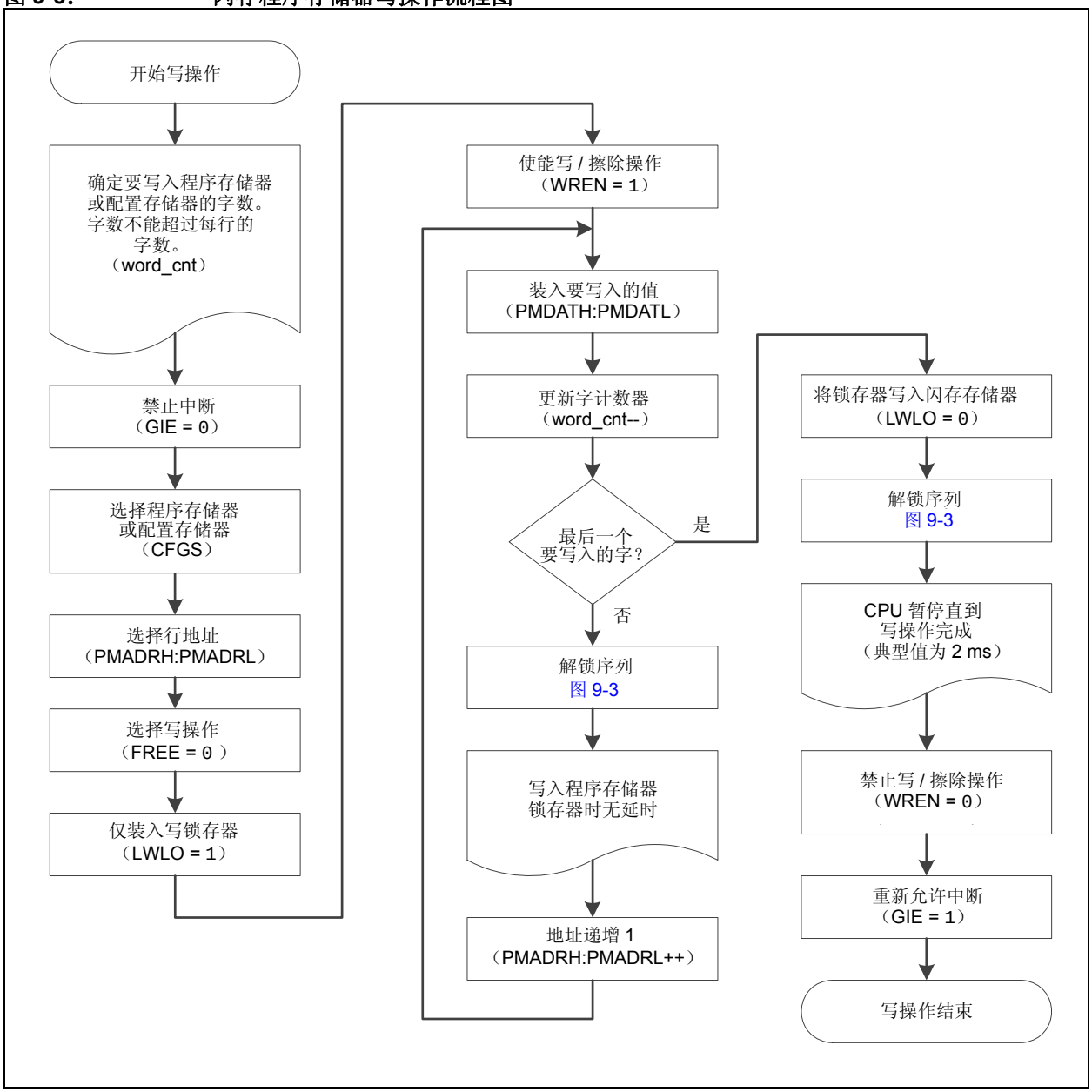
© 2011-2014 Microchip Technology Inc.

图 9-5： 使用 16 个写锁存器对闪存程序存储器进行块写操作



© 2011-2014 Microchip Technology Inc.

图 9-6： 闪存程序存储器写操作流程图



© 2011-2014 Microchip Technology Inc.

例 **9-3：** 写入闪存程序存储器

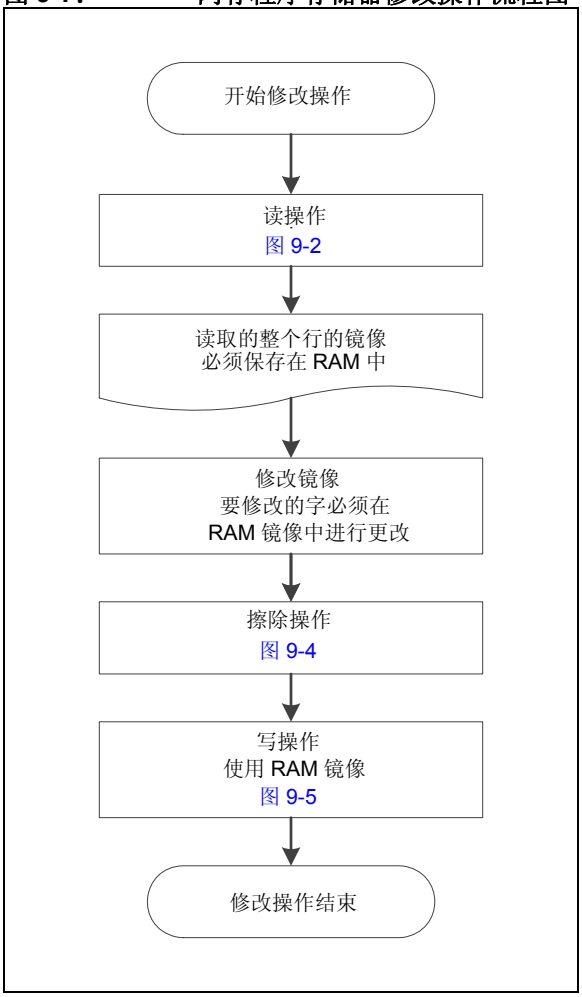
```
    ;;;
```

9.3 修改闪存程序存储器

当修改程序存储器行中的已有数据，并且必须保存该行中的数据时，首先必须读取它并将其保存在 RAM 镜像中。使用以下步骤修改程序存储器：

1. 装入要修改的行的起始地址。
2. 从行中读取已有数据并将其保存到RAM镜像中。
3. 修改 RAM 镜像，以包含要写入到程序存储器的新数据。
4. 装入要重新写入的行的起始地址。
5. 擦除程序存储器行。
6. 将来自 RAM 镜像的数据装入写锁存器。
7. 启动编程操作。

图 9-7： 闪存程序存储器修改操作流程图



© 2011-2014 Microchip Technology Inc.

9.4 用户 ID、器件 ID 和配置字访问

当 PMCON1 寄存器中的 CFGS = 1 时，不是访问程序存储器，而是访问用户 ID、器件 ID/ 版本 ID 和配置字。这是 PC<13> = 1 时指向的区域，但不是所有的地址都可访问。对于读和写操作存在不同的访问权限。请参见表 9-2。

在对表 9-2 列出的参数以外的地址进行读访问时，PMDATH:PMDATL 寄存器对将被清零，读回 0。

表 9-2： 用户 ID、器件 ID 和配置字访问 （CFGS = 1）

地址	功能	读访问	写访问
2000h-2003h	用户 ID	允许	允许
2006h	器件 ID/ 版本 ID	允许	禁止
2007h	配置字	允许	禁止

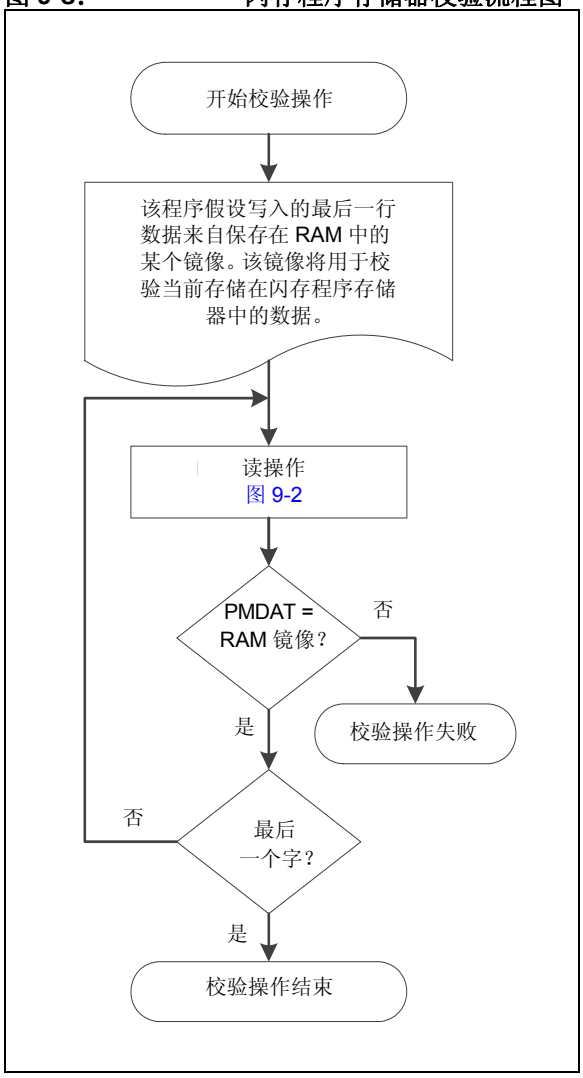
例 9-4： 配置字和器件 ID 访问

```
* This code block will read 1 word of program memory at the memory address:
*   PROG_ADDR_LO (must be 00h-08h) data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

    BANKSEL  PMADRL          ; not required on devices with 1 Bank of SFRs
    MOVLW    PROG_ADDR_LO    ;
    MOVWF    PMADRL          ; Store LSB of address
    CLRF     PMADRH          ; Clear MSB of address

    BSF      PMCON1,CFGS     ; Select Configuration Space
    BCF      INTCON,GIE      ; Disable interrupts
    BSF      PMCON1,RD       ; Initiate read
    NOP                      ; Executed (See Figure 9-2)
    NOP                      ; Ignored (See Figure 9-2)
    BSF      INTCON,GIE      ; Restore interrupts

    MOVF     PMDATL,W        ; Get LSB of word
    MOVWF    PROG_DATA_LO    ; Store in user location
    MOVF     PMDATH,W        ; Get MSB of word
    MOVWF    PROG_DATA_HI    ; Store in user location
```

9.5 写校验

将写入程序存储器的值对照期望写入的值进行校验是一个良好的编程习惯。由于程序存储器按整页存储，因此可在完成最后一个写操作之后将存储的程序存储器内容与存储在 RAM 中的期望数据做比较。

图 9-8：闪存程序存储器校验流程图



© 2011-2014 Microchip Technology Inc.

9.6 闪存程序存储器控制寄存器

寄存器 9-1： PMDATL：程序存储器数据低字节寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PMDAT<7:0>							
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0 **PMDAT<7:0>**：在程序存储器读命令之后， PMADRH 和 PMADRL 指向的程序存储器字的值。

寄存器 9-2： PMDATH：程序存储器数据高字节寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	PMDAT<13:8>					
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-6 **未实现**：读为 0

bit 5-0 **PMDAT<13:8>**：在程序存储器读命令之后， PMADRH 和 PMADRL 指向的程序存储器字的值。

寄存器 9-3： PMADRL：程序存储器地址低字节寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PMADR<7:0>							
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0 **PMADR<7:0>：**程序存储器读地址低位

寄存器 9-4： PMADRH：程序存储器地址高字节寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	PMADR8
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-1 未实现：读为 0

bit 0 **PMADR8：**程序存储器读地址高位

© 2011-2014 Microchip Technology Inc.

寄存器 9-5： PMCON1：程序存储器控制寄存器 1

U-1[(1)]	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HC-0/q[(2)]	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
—	CFGS	LWLO	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
S = 只可置 1 位	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	HC = 由硬件清零

bit 7 **未实现：**读为 1

bit 6 **CFGS：**配置寄存器选择位
1 = 访问配置寄存器、用户 ID 寄存器和器件 ID 寄存器
0 = 访问闪存程序存储器

bit 5 **LWLO：**仅装入写锁存器位 [(3)]
1 = 在下一条 WR 指令时仅装入 / 更新寻址的程序存储器写锁存器
0 = 装入 / 更新寻址的程序存储器写锁存器，且在下一条 WR 指令将启动对所有程序存储器写锁存器的写操作

bit 4 **FREE：**程序闪存擦除使能位
1 = 在下一条 WR 命令执行擦除操作 （擦除完成后由硬件清零）
0 = 在下一条 WR 命令执行写操作

bit 3 **WRERR：**编程 / 擦除错误标志位
1 = 该状态表示试图进行不当的编程或试图擦除序列或意外终止（试图将 WR 位置 1（写入 1）时，该位自动置 1）。
0 = 编程或擦除操作正常完成。

bit 2 **WREN：**编程 / 擦除使能位
1 = 允许执行编程 / 擦除操作
0 = 禁止编程 / 擦除闪存程序存储器

bit 1 **WR：**写控制位
1 = 启动闪存程序存储器编程 / 擦除操作。
该操作是自定时的，且该位在操作完成时由硬件清零。
用软件只能将 WR 位置 1，但不能清零。
0 = 对闪存的编程 / 擦除操作已完成，当前不在进行中。

bit 0 **RD：**读控制位
1 = 启动对闪存程序存储器的读操作。读操作只占用一个周期。RD 由硬件清零。用软件只能将 RD 位置 1，但不能清零。
0 = 不启动闪存程序存储器读操作。

注 **1：** 未实现位，读为 1。
2： 当启动程序存储器写或擦除操作 （WR = 1）时， WRERR 位由硬件自动置 1。
3： 在执行程序存储器擦除操作 （FREE = 1）时，忽略 LWLO 位。

寄存器 9-6： PMCON2：程序存储器控制寄存器 2

W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0
程序存储器控制寄存器 2							
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
S = 只可置 1 位	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0 闪存解锁模式位

要解锁写操作，在将 PMCON1 寄存器的 WR 位置 1 前，必须先写 55h，随后写入 AAh。写入该寄存器的值用于解锁写操作。对这些写操作有特殊的时序要求。

表 9-3： 与闪存程序存储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	寄存器所在页
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
PMCON1	—	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	65
PMCON2	程序存储器控制寄存器 2								66
PMADRL	PMADR<7:0>								64
PMADRH	—	—	—	—	—	—	—	PMADR8	64
PMDATL	PMDAT<7:0>								63
PMDATH	—	—	PMDAT<13:8>						63

图注： — = 未实现单元，读为 0。闪存程序存储器模块不使用阴影单元。

表 9-4： 与闪存程序存储器相关的配置字汇总

名称	Bit	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在页
CONFIG	13:8	—	—	—	WRT<1:0>		BORV	LPBOR	LVP	20
	7:0	$\overline{CP}$	$\overline{MCLR}$	$\overline{PWRTE}$	WDTE<1:0>		BOREN<1:0>		FOSC	

图注： — = 未实现单元，读为 0。闪存程序存储器不使用阴影单元。

© 2011-2014 Microchip Technology Inc.

10.0 I/O 端口

根据所使能的外设，部分或所有引脚可能不能用作通用 I/O。通常情况下，当某个端口引脚上的外设使能时，该引脚不能用作通用输出引脚，但可读取该引脚。

PORTA 有 3 个标准寄存器供其操作使用。这些寄存器包括：

- TRISA 寄存器 （数据方向寄存器）
- PORTA 寄存器 （用于读器件引脚上的电平）
- LATA 寄存器 （输出锁存器）

某些端口可能有以下一个或几个额外寄存器。这些寄存器包括：

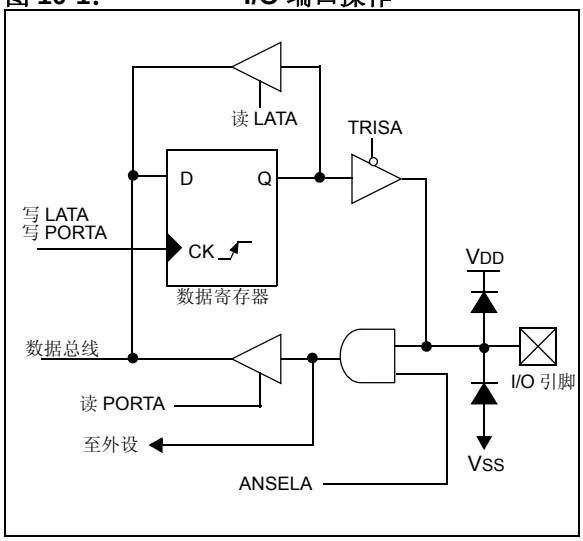
- ANSELA （模拟选择寄存器）
- WPUA （弱上拉寄存器）

数据锁存器 （LATA 寄存器）用于对 I/O 引脚所驱动的值进行读 - 修改 - 写操作。

对 LATA 寄存器的写操作与对相应 PORTA 寄存器的写操作具有相同的效果。读 LATA 寄存器将读取保存在 I/O 端口锁存器中的值，而读 PORTA 寄存器将读取实际的 I/O 引脚值。

支持模拟输入的端口具有相关的 ANSELA 寄存器。当 ANSEL 位置 1 时，禁止与该位相关的数字输入缓冲器。当禁止输入缓冲器时，可防止介于逻辑高电平和逻辑低电平之间的引脚上的模拟信号电平导致逻辑输入电路产生过电流。图 10-1 给出了一个通用 I/O 端口的简化模型，图中未显示与其他外设的接口。

图 10-1： I/O 端口操作



例 10-1： 初始化 PORTA

```
; This code example illustrates
; initializing the PORTA register. The
; other ports are initialized in the same
; manner.

BANKSEL    PORTA          ;not required on devices with 1 Bank of SFRs
CLRF       PORTA          ;Init PORTA
BANKSEL    LATA           ;not required on devices with 1 Bank of SFRs
CLRF       LATA           ;
BANKSEL    ANSELA         ;not required on devices with 1 Bank of SFRs
CLRF       ANSELA         ;digital I/O
BANKSEL    TRISA          ;not required on devices with 1 Bank of SFRs
MOVLW      B'00000011'    ;Set RA<1:0> as inputs
MOVWF      TRISA          ;and set RA<2:3> as
                          ;outputs
```

10.1 PORTA 寄存器

PORTA 是 8 位宽的双向端口。其对应的数据方向寄存器是 TRISA （寄存器 10-2）。当 TRISA 某位置 1 （= 1）时，可以将相应的 PORTA 引脚作为输入引脚（即，禁止输出驱动器）。当 TRISA 某位清零 （= 0）时，可以让相应的 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传送至选定的引脚上） 。例 10-1 说明了如何初始化 PORTA。

读 PORTA 寄存器 （寄存器 10-1）将读出引脚的状态，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 （LATA）。

即使 PORTA 引脚用作模拟输入引脚， TRISA 寄存器（寄存器 10-2）仍然控制 PORTA 引脚的输出驱动器。当用作模拟输入引脚时，用户必须确保 TRISA 寄存器中的位保持为置 1 状态。配置为模拟输入引脚的 I/O 引脚总是读为 0。

10.1.1 弱上拉

每个 PORTA 引脚均具有可独立配置的内部弱上拉。控制位 WPUA<3:0> 可使能或禁止每个上拉 （见寄存器 10-5）。当端口引脚配置为输出时，其弱上拉电路会自动切断。上电复位时，通过 OPTION_REG 寄存器的 WPUEN 位禁止所有上拉。

10.1.2 ANSELA 寄存器

ANSELA 寄存器 （寄存器 10-4）用于将 I/O 引脚的输入模式配置为模拟模式。当相应的 ANSELA 位设置为高电平时，可以使引脚上的所有数字读操作都读为 0，并允许引脚上的模拟功能正确工作。

ANSELA 位的状态不会影响数字输出功能。 TRIS 清零且 ANSEL 置 1 的引脚将仍作为数字输出引脚工作，但输入模式将变为模拟输入模式。当在受影响的端口上执行读 - 修改 - 写指令时，得到的结果可能与预期不符。

> **注：** 复位后， ANSELA 位默认为模拟模式。要将任一引脚用作数字通用或者外设输入，相应的ANSEL位必须由用户软件初始化为 0。

10.1.3 PORTA 功能和输出优先级

每个 PORTA 引脚均复用多个功能。表 10-1 给出了引脚及其组合功能和输出优先级。

使能多个输出时，引脚实际上由具有最高优先级的外设控制。

当引脚处于模拟模式并具有表 10-1 所示的优先级时，数字输出功能可以控制该引脚。

表 10-1： PORTA 输出优先级

引脚名称	功能优先级 (1)
RA0	ICSPDAT CWG1A PWM1 RA0
RA1	CWG1B PWM2 CLC1 RA1
RA2	NCO1 CLKR RA2
RA3	无

注 1： 按优先级从高至低排列。

© 2011-2014 Microchip Technology Inc.

10.2 寄存器定义：PORTA

寄存器 10-1： PORTA：PORTA 寄存器

U-0	U-0	U-0	U-0	R-x/x	R/W-x/x	R/W-x/x	R/W-x/x
—	—	—	—	RA3	RA2	RA1	RA0
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-4 **未实现：**读为 0

bit 3-0 **RA<3:0>：**PORTA I/O 值位 （RA3 是只读位）

注 1： 写入 PORTx 时，实际上写入相应的 LATx 寄存器。读取 PORTx 寄存器时，将返回实际的 I/O 引脚值。

寄存器 10-2： TRISA：PORTA 三态寄存器

U-0	U-0	U-0	U-0	U-1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	—(1)	TRISA2	TRISA1	TRISA0
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-4 **未实现：**读为 0

bit 3 **未实现：**读为 1

bit 2-0 **TRISA<2:0>：**RA<2:0> 端口 I/O 三态控制位
1 = 禁止端口输出驱动器
0 = 使能端口输出驱动器

注 1： 未实现，读为 1。

寄存器 10-3： LATA：PORTA 数据锁存寄存器

U-0	U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	—	LATA2	LATA1	LATA0
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-3 **未实现：**读为 0

bit 2-0 **LATA<2:0>：**RA<2:0> 输出锁存器值位

注 1： 写入 PORTx 时，实际上写入相应的 LATx 寄存器。读取 LATx 寄存器时，将返回寄存器值，而不是 I/O 引脚值。

寄存器 10-4： ANSELA：PORTA 模拟选择寄存器

U-0	U-0	U-0	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	—	ANSA2	ANSA1	ANSA0
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-3 **未实现：**读为 0

bit 2-0 **ANSA<2:0>：**将 RA<2:0> 引脚选择为模拟或数字功能
1 = 模拟输入。引脚设置为模拟输入引脚 (1)。禁止数字输入缓冲器。
0 = 数字 I/O。引脚设置为端口或是数字特殊功能。

注 1： 当引脚设置为模拟输入时，可以自动禁止数字输入电路。弱上拉 （如果有）不受影响。用户必须将相应的 TRIS 位设置为输入模式，以允许外部控制引脚电压。

© 2011-2014 Microchip Technology Inc.

寄存器 10-5： WPUA：弱上拉 PORTA 寄存器

U-0	U-0	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	WPUA3	WPUA2	WPUA1	WPUA0
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-4 **未实现：**读为 0

bit 3-0 **WPUA<3:0>：**弱上拉 PORTA 控制位
1 = 使能弱上拉 (1)
0 = 禁止弱上拉

注 1： 要使能弱上拉，还需要将 OPTION_REG 寄存器的 $\overline{\text{WPUEN}}$ 位清零 （寄存器 16-1）。

表 10-2： 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	寄存器所在页
ANSELA	—	—	—	—	—	ANSA2	ANSA1	ANSA0	70
IOCAF	—	—	—	—	IOCAF3	IOCAF2	IOCAF1	IOCAF0	76
IOCAN	—	—	—	—	IOCAN3	IOCAN2	IOCAN1	IOCAN0	75
IOCAP	—	—	—	—	IOCAP3	IOCAP2	IOCAP1	IOCAP0	75
LATA	—	—	—	—	—	LATA2	LATA1	LATA0	70
PORTA	—	—	—	—	RA3	RA2	RA1	RA0	69
TRISA	—	—	—	—	—(1)	TRISA2	TRISA1	TRISA0	69
WPUA	—	—	—	—	WPUA3	WPUA2	WPUA1	WPUA0	71

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

注 1： 未实现，读为 1。

© 2011-2014 Microchip Technology Inc.

11.0 电平变化中断

PORTA 引脚可以配置为电平变化中断 （Interrupt-On-Change， IOC）引脚。当检测到具有上升沿或下降沿的信号时，产生中断。任何独立的 PORTA 引脚或 PORTA 引脚组合都可以配置为产生中断。电平变化中断模块具有以下特性：

- 允许电平变化中断 （主开关）
- 独立的引脚配置
- 上升沿和下降沿检测
- 独立的引脚中断标志

图 11-1 给出了 IOC 模块的框图。

11.1 使能模块

要允许独立的 PORTA 引脚产生中断，必须将 INTCON 寄存器的 IOCIE 位置 1。如果禁止 IOCIE 位，则仍然会对引脚上的信号进行边沿检测，但是不产生中断。

11.2 独立的引脚配置

每个 PORTA 引脚都带有一个上升沿检测器和一个下降沿检测器。要使能某个引脚检测上升沿，必须将 IOCAP 寄存器中相应的 IOCAPx 位置 1。要使能某个引脚检测下降沿，必须将IOCAN寄存器中相应的IOCANx位置1。

通过分别将IOCAP寄存器中相应的IOCAPx位和IOCAN寄存器中相应的 IOCANx 位置 1，可将某个引脚配置为同时检测上升沿和下降沿。

11.3 中断标志

IOCAF 寄存器中的 IOCAFx 位是对应于 PORTA 的电平变化中断引脚的状态标志位。如果在相应的已使能引脚上检测到预期的边沿，则该引脚的状态标志位将置 1，且如果 IOCIE 位置 1 的话，还会产生中断。INTCON 寄存器的 IOCIF 位会反映所有 IOCAFx 位的状态。

11.4 清零中断标志

各个状态标志 （IOCAFx 位）可以通过将其复位为零的方式进行清零。如果在清零操作期间检测到另一个边沿，则无论实际写入的值如何，相关的状态标志位都会在序列结束时置 1。

为了确保清零标志时不丢失所检测到的边沿，只需执行用于屏蔽已知更改位的 “逻辑与”操作。以下举例说明了应执行的序列。

例 11-1：

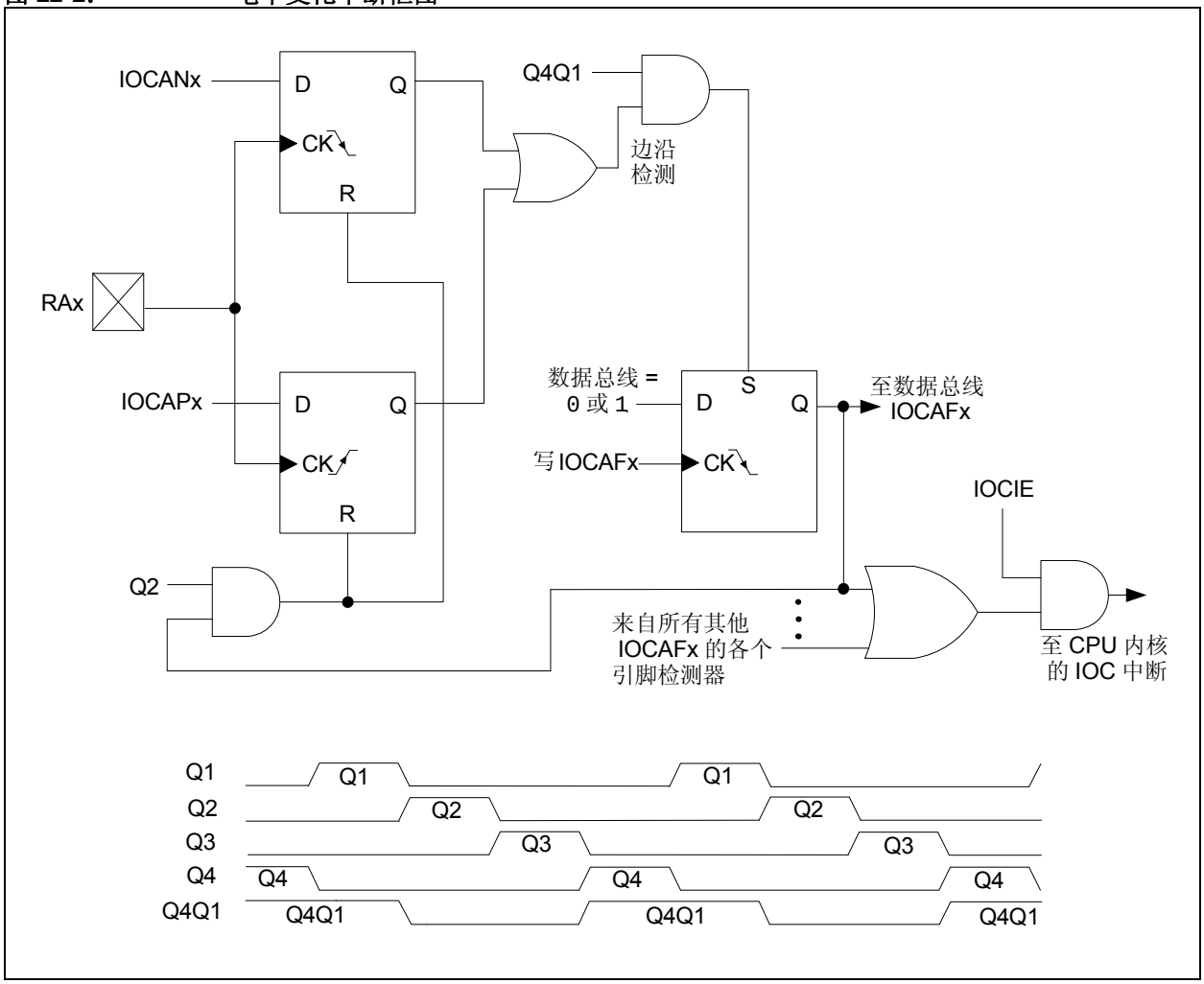
```
MOVLW   0xff
XORWF   IOCAF, W
ANDWF   IOCAF, F
```

11.5 休眠期间的工作原理

如果 IOCIE 位置 1 的话，电平变化中断序列将器件从休眠模式中唤醒。

如果在休眠模式下检测到边沿，则在退出休眠模式执行第一条指令之前，会更新 IOCAF 寄存器。

图 11-1： 电平变化中断框图



© 2011-2014 Microchip Technology Inc.

11.6 电平变化中断寄存器

寄存器 11-1： IOCAP：电平变化中断 PORTA 正边沿寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	IOCAP3	IOCAP2	IOCAP1	IOCAP0
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-4 **未实现：**读为 0。

bit 3-0 **IOCAP<3:0>：**电平变化中断 PORTA 正边沿允许位
1 = 允许引脚上的正边沿电平变化中断。检测到边沿时，将相关的状态位和中断标志位置 1。(1)
0 = 禁止相关引脚的电平变化中断。

注 1： 电平变化中断还需要将 INTCON 寄存器的 IOCIE 位置 1 （寄存器 6-1）。

寄存器 11-2： IOCAN：电平变化中断 PORTA 负边沿寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	IOCAN3	IOCAN2	IOCAN1	IOCAN0
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-4 **未实现：**读为 0。

bit 3-0 **IOCAN<3:0>：**电平变化中断 PORTA 负边沿允许位
1 = 允许引脚上的负边沿电平变化中断。检测到边沿时，将相关的状态位和中断标志位置 1。(1)
0 = 禁止相关引脚的电平变化中断。

注 1： 电平变化中断还需要将 INTCON 寄存器的 IOCIE 位置 1 （寄存器 6-1）。

寄存器 11-3： IOCAF：电平变化中断 PORTA 标志寄存器

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	IOCAF3	IOCAF2	IOCAF1	IOCAF0
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	HS = 由硬件清零

bit 7-4 **未实现：**读为 0。

bit 3-0 **IOCAF<3:0>：**电平变化中断 PORTA 标志位

1 = 检测到相关引脚上发生允许的电平变化。
当 IOCAPx = 1 且检测到 RAx 引脚上出现上升沿时，或当 IOCANx = 1 且检测到 RAx 引脚上出现下降沿时，该标志位置 1。(1)

0 = 未检测到电平变化，或用户清除检测到的电平变化。

注 1： 电平变化中断还需要将 INTCON 寄存器的 IOCIE 位置 1（寄存器 6-1）。

表 11-1： 与电平变化中断相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
IOCAF	—	—	—	—	IOCAF3	IOCAF2	IOCAF1	IOCAF0	76
IOCAN	—	—	—	—	IOCAN3	IOCAN2	IOCAN1	IOCAN0	75
IOCAP	—	—	—	—	IOCAP3	IOCAP2	IOCAP1	IOCAP0	75
TRISA	—	—	—	—	—(1)	TRISA2	TRISA1	TRISA0	69

图注： — = 未实现单元，读为 0。电平变化中断不使用阴影单元。

注 1： 未实现，读为 1。

© 2011-2014 Microchip Technology Inc.

12.0 固定参考电压 （FVR）

固定参考电压（FVR）是独立于 VDD 的稳定参考电压，可选的输出电平具有 1.024V、2.048V 或 4.096V。可配置 FVR 的输出为以下项提供参考电压：

- ADC 输入通道

通过将 FVRCON 寄存器的 FVREN 位置 1 可使能 FVR。

12.1 独立的增益放大器

FVR 的输出可通过 1 个独立的可编程增益放大器连接到 ADC。放大器可配置为将参考电压放大 1 倍、 2 倍或 4 倍，从而提供三种可能的电压级别。

FVRCON 寄存器的 ADFVR<1:0> 位可以使能和配置增益放大器设置，用于给 ADC 模块提供参考电压。更多信息，请参见第 15.0 “模数转换器 （ADC）模块”。

要最大程度地降低 FVR 禁止时的电流消耗，应该通过清零 ADFVR<1:0> 位来关闭 FVR 缓冲器。

12.2 FVR 稳定时间

当使能固定参考电压模块时，参考电路和放大器电路需要一定的时间才能稳定。一旦电路达到稳定且就绪时，FVRCON 寄存器的 FVRRDY 位将置 1。关于最小延时的要求，请参见第 24.0 “电气规范”。

图 12-1： 参考电压框图

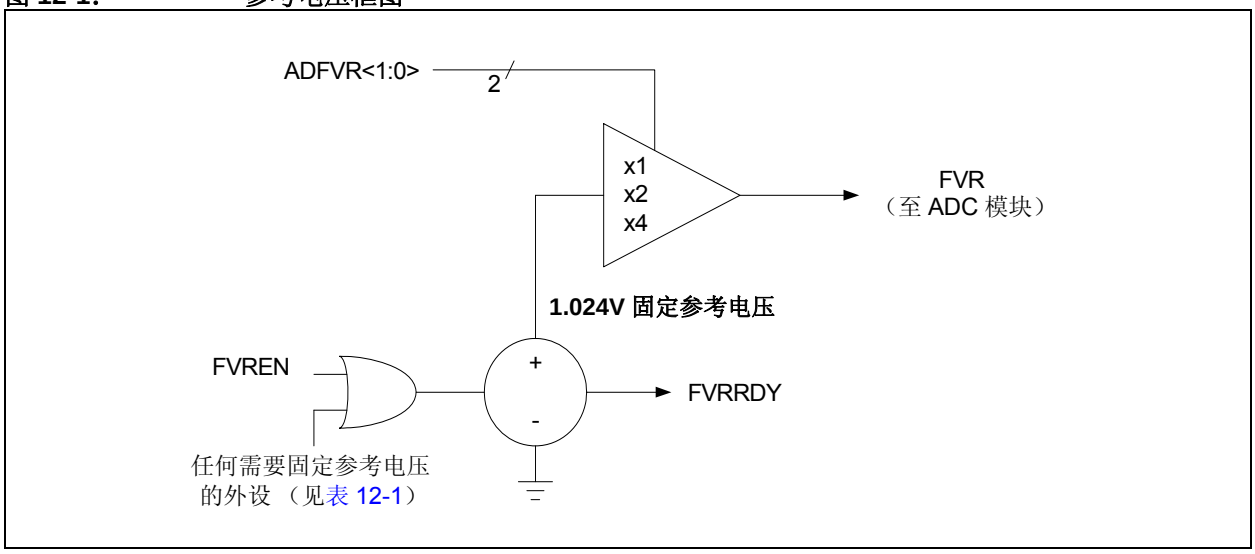


表 12-1： 需要固定参考电压 （FVR）的外设

外设	条件	说明
HFINTOSC	FOSC = 1	CLKIN 引脚上的 EC。
BOR	BOREN<1:0> = 11	始终使能 BOR。
	BOREN<1:0> = 10 且 BORFS = 1	在休眠模式下禁止 BOR，使能 BOR 快速启动。
	BOREN<1:0> = 01 且 BORFS = 1	BOR 由软件控制，使能 BOR 快速启动。
IVR	所有 PIC10F320/322 器件，VREGPM1 = 1 且不处于休眠模式	当处于休眠模式时，器件将使用节能模式稳压器。

12.3 FVR 控制寄存器

寄存器 12-1： FVRCON：固定参考电压控制寄存器

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
FVREN	FVRRDY(1)	TSEN(3)	TSRNG(3)	—	—	ADFVR<1:0>	
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	q = 值取决于具体条件

bit 7 **FVREN**：固定参考电压使能位
1 = 使能固定参考电压
0 = 禁止固定参考电压

bit 6 **FVRRDY**：固定参考电压就绪标志位 (1)
1 = 固定参考电压输出已就绪
0 = 固定参考电压输出未就绪或未使能

bit 5 **TSEN**：温度指示器使能位 (3)
1 = 使能温度指示器
0 = 禁止温度指示器

bit 4 **TSRNG**：温度指示器范围选择位 (3)
1 = $V_{OUT} = V_{DD} - 4V_T$ （高电压范围）
0 = $V_{OUT} = V_{DD} - 2V_T$ （低电压范围）

bit 3-2 未实现：读为 0

bit 1-0 **ADFVR<1:0>**：ADC 固定参考电压选择位
11 = ADC 固定参考电压外设输出为 4x (4.096V)(2)
10 = ADC 固定参考电压外设输出为 2x (2.048V)(2)
01 = ADC 固定参考电压外设输出为 1x (1.024V)
00 = ADC 固定参考电压外设输出关闭。

注 1： FVRRDY 表示 FVR 的真实状态。
2： 固定参考电压输出不能超过 V_{DD}。
3： 更多信息，请参见第 14.0 “温度指示器模块”。

表 12-2： 与固定参考电压相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	—	—	ADFVR<1:0>		78

图注： 固定参考电压不使用阴影单元。

© 2011-2014 Microchip Technology Inc.

13.0 内部稳压器 （IVR）

以下器件提供工作电压高于3.6V的内部稳压器（Internal Voltage Regulator， IVR）：

- PIC10F320
- PIC10F322

该电路可为内部器件逻辑提供稳压，同时允许 VDD 和 I/O 引脚工作在较高的电压下。VDD 接近稳定电压时，IVR 输出将自动跟踪输入电压。

根据用户配置和外设选择，IVR 可在三种功耗模式之一下工作。工作功耗模式包括：

- 高功耗模式
- 低功耗模式
- 节能休眠模式

功耗模式根据器件工作情况自动选择，如表 13-1 所示。当 VDD 降至低于内核的安全工作电压时，自动选择跟踪模式。

> **注：** IVR 在跟踪模式下禁止，但是还会耗电。更多信息，请参见**第 24.0 节 “电气规范”**。

表 13-1： IVR 功耗模式 —— 稳压

VREGPM1 位	休眠模式	存储器偏置功耗模式	IVR 功耗模式
x	否	EC 模式或 INTOSC = 16 MHz （高功耗偏置）	高
		INTOSC = 1 MHz 至 8 MHz （中等功耗偏置）	
		INTOSC = 31 kHz 至 500 kHz （低功耗偏置）	低
0	是	无关位	低
1	是	无 HFINTOSC 无外设	节能 (1)

注 1： 在以下任意条件下强制为低功耗模式：
- 使能 BOR
- HFINTOSC 是有效外设源
- 自写有效
- ADC 正在进行有效转换

寄存器 13-1： VREGCON：稳压器控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-1/1
—	—	—	—	—	—	VREGPM1	保留
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-2 **未实现：**读为 0

bit 1 **VREGPM1：**稳压器功耗模式选择位
1 = 休眠时使能节能休眠模式。休眠时消耗的电流最少，唤醒较慢。
0 = 休眠时使能低功耗模式。休眠时消耗的电流较多，唤醒较快。

bit 0 **保留：**保持该位置 1。

© 2011-2014 Microchip Technology Inc.

14.0 温度指示器模块

本器件系列配置了一个温度电路，该电路旨在测量硅片的工作温度。电路的工作温度范围在 -40°C 至 +85°C 之间。输出电压与器件温度成正比。温度指示器的输出内部连接到器件 ADC。

该电路可用作温度阈值检测器或更精确的温度指示器，具体取决于执行的校准级别。单点校准使电路指示该点附近的温度。两点校准使电路能更精确地测量整个温度范围。关于校准过程的更多信息，请参见应用笔记《内部温度指示器的使用与校准》（DS01333A_CN）。

14.1 电路工作原理

图 14-1 给出了该温度电路的简化框图。与温度成正比的电压输出通过测量多个硅片接点上的正向压降实现。

公式 14-1 说明了温度指示器的输出特性。

公式 14-1： VOUT 范围

高电压范围：$V_{OUT} = V_{DD} - 4V_T$

低电压范围：$V_{OUT} = V_{DD} - 2V_T$

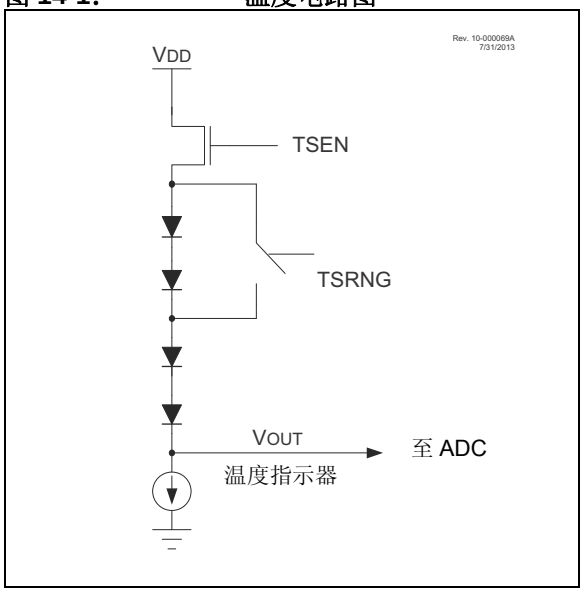
温度检测电路集成了固定参考电压 （FVR）模块。更多信息，请参见**第 12.0 节 “固定参考电压 （FVR）”**。

将 FVRCON 寄存器的 TSEN 位置 1 可使能该电路。禁止时，该电路不消耗电流。

该电路可在高或低电压范围下工作。高电压范围通过将 FVRCON 寄存器的 TSRNG 位置 1 来选择，可提供更宽的输出电压。这为整个温度范围提供了更高的分辨率，但是各部件之间一致性较差。该范围需要更高的偏置电压才能工作，因此，需要更高的 VDD。

低电压范围通过将 FVRCON 寄存器的 TSRNG 位清零来选择。低电压范围会产生较低的压降，因此，该电路需要较低的偏置电压才能工作。低电压范围针对低电压工作而提供。

图 14-1： 温度电路图



14.2 最小工作电压 VDD 与最低检测温度

当温度电路在低电压范围下工作时，器件可以在规范范围内的任何工作电压下工作。

当温度电路在高电压范围下工作时，器件工作电压 VDD 必须足够高，以确保温度电路正确偏置。

表 14-1 给出了建议的最小 VDD 与范围设置。

表 14-1： 建议的 VDD 与范围

最小 VDD， TSRNG = 1	最小 VDD， TSRNG = 0
3.6V	1.8V

14.3 温度输出

使用内部模数转换器测量电路的输出。预留了一个通道用于温度电路输出。更多详细信息，请参见**第 15.0 节 “模数转换器 （ADC）模块”**。

14.4 ADC 采集时间

要确保精确的温度测量，在 ADC 输入多路开关连接到温度指示器输出之后，用户必须至少等待 200 μs 才能执行转换。此外，在连续转换温度指示器输出之间，用户必须等待 200 μs。

表 14-2： 与温度指示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	寄存器所在页
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	—	—	ADFVR<1:0>		78
ADCON	ADCS<2:0>			CHS<2:0>			GO/DONE	ADON	88
ADRES	A/D 结果寄存器								89

图注： — = 未实现单元，读为 0。温度指示器模块不使用阴影单元。

© 2011-2014 Microchip Technology Inc.

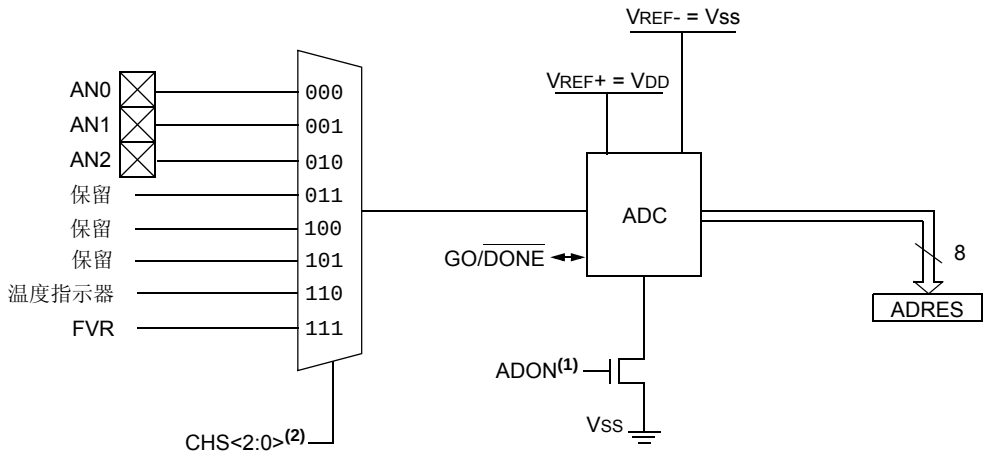
15.0 模数转换器 （ADC）模块

模数转换器（Analog-to-Digital Converter，ADC）可以将模拟输入信号转换为该信号的 8 位二进制表示。该器件使用三路模拟输入通道，这些输入通道通过多路开关连接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。该转换器通过逐次逼近法生成 8 位二进制数并将转换结果存储到ADC结果寄存器（ADRES）。图 15-1 给出了 ADC 框图。

可通过软件选择内部产生的电压作为 ADC 参考电压。

ADC 可在转换完成时产生中断。此中断可用于将器件从休眠状态中唤醒。

图 15-1： ADC 简化框图



注 1： 当 ADON = 0 时，断开所有多路开关输入。

2： 关于每个器件的模拟通道选择的详细信息，请参见 ADCON 寄存器 （寄存器 15-1）。

15.1 ADC 配置

当配置和使用 ADC 时，必须考虑以下功能：

- 端口配置
- 通道选择
- ADC 转换时钟源
- 中断控制

15.1.1 端口配置

ADC 可用于将模拟信号转化为数字信号。当转换模拟信号时，应通过设置相关 TRIS 和 ANSEL 位将 I/O 引脚配置为模拟引脚。更多信息，请参见**第 10.0 节 “I/O 端口”**。

注：	在任何定义为数字输入的引脚上施加模拟电压可能导致输入缓冲器消耗的电流过大。

15.1.2 通道选择

最多 5 路通道可供选择：

- AN<2:0> 引脚
- 温度指示器
- FVR （固定参考电压）输出

更多关于这些通道选择的信息，请参见**第 12.0 节 “固定参考电压（FVR）”**和**第 14.0 节“温度指示器模块”**。

ADCON 寄存器的 CHS 位控制将哪一路通道与采样保持电路相连。

当更改通道时，需要一段时间的延时才能启动下一次转换。更多信息，请参见**第 15.2 节 “ADC 工作原理”**。

15.1.3 ADC 参考电压

没有外部参考电压连接到 ADC。仅 VDD 可用作参考电压源。 FVR 仅可用作输入通道，而不是连接到 ADC 的 VREF+ 输入。

15.1.4 转换时钟

可通过软件设置 ADCON 寄存器的 ADCS 位来选择转换时钟源（寄存器 15-1）。有以下 7 种时钟频率可供选择：

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC （专用内部 RC 振荡器）

转换 1 位所需的时间定义为 TAD。一次完整的 8 位转换需要 9.5 个 TAD 周期，如图 15-2 所示。

要实现正确的转换，必须满足适当的 TAD 规范。更多信息，请参见**第 24.0 节 “电气规范”**中的 A/D 转换要求。表 15-1 给出了一个适当的 ADC 时钟选择的示例。

注 1：	除非使用 FRC，否则系统时钟频率的任何变化都会改变 ADC 时钟频率，这会影响 ADC 结果。

© 2011-2014 Microchip Technology Inc.

表 15-1： ADC 时钟周期 （TAD）与器件工作频率的关系

ADC 时钟周期 （TAD）		器件频率 （FOSC）			
ADC 时钟源	ADCS<2:0>	16 MHz	8 MHz	4 MHz	1 MHz
FOSC/2	000	125 ns(1)	250 ns(1)	500 ns(1)	2.0 μs
FOSC/4	100	250 ns(1)	500 ns(1)	1.0 μs	4.0 μs
FOSC/8	001	0.5 μs(1)	1.0 μs	2.0 μs	8.0 μs(2)
FOSC/16	101	1.0 μs	2.0 μs	4.0 μs	16.0 μs(2)
FOSC/32	010	2.0 μs	4.0 μs	8.0 μs(2)	32.0 μs(2)
FOSC/64	110	4.0 μs	8.0 μs(2)	16.0 μs(2)	64.0 μs(2)
FRC	x11	1.0-6.0 μs(1,3)	1.0-6.0 μs(1,3)	1.0-6.0 μs(1,3)	1.0-6.0 μs(1,3)

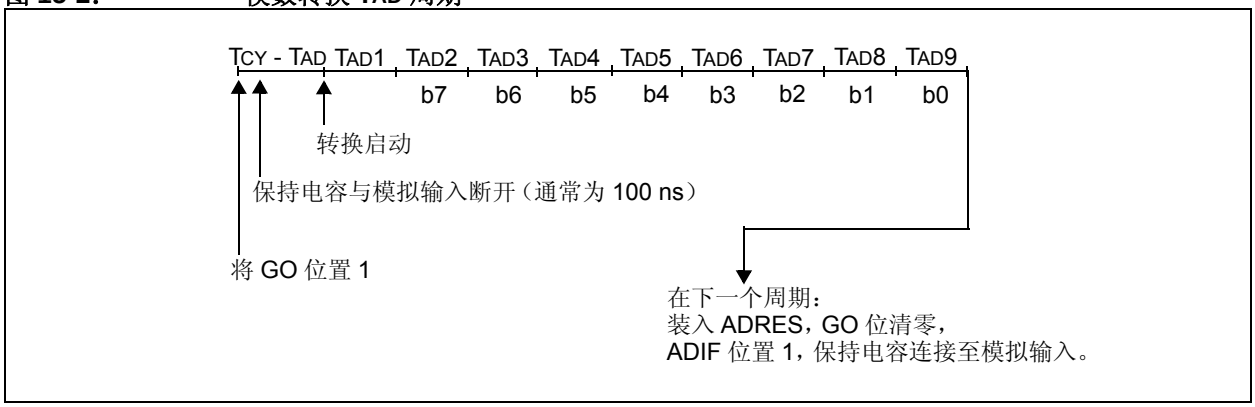
图注： 阴影单元表示超出了建议值的范围。

注 1： 这些值均违反了最小 TAD 时间要求。

2： 为了加快转换速度，建议选用其他时钟源。

3： 当 ADC 时钟由系统时钟 FOSC 提供时，可以最大程度地减少 ADC 时钟周期 （TAD）和 ADC 总转换时间。但是，如果当器件处于休眠模式时执行转换，则必须使用 FRC 时钟源。

图 15-2： 模数转换 TAD 周期



15.1.5 中断

ADC 模块可以在模数转换完成时产生中断。 ADC 中断标志位是 PIR1 寄存器中的 ADIF 位。 ADC 中断允许位是 PIE1 寄存器中的 ADIE 位。必须用软件将 ADIF 位清零。

注：	无论是否允许 ADC 中断，每次转换完成时都会将 ADIF 位置 1。

器件在工作期间或者休眠模式下都可以产生该中断。如果器件处于休眠模式，则该中断可唤醒器件。从休眠模式中唤醒时，总是执行 SLEEP 指令后的下一条指令。如果用户试图将器件从休眠模式中唤醒并继续执行主程序代码，则必须禁止 INTCON 寄存器的 GIE 和 PEIE 位。如果使能了 INTCON 寄存器的 GIE 和 PEIE 位，则转为执行中断服务程序。

15.2 ADC 工作原理

15.2.1 启动转换

要使能 ADC 模块，必须将 ADCON 寄存器的 ADON 位设置为 1。通过将 ADCON 寄存器的 GO/$\overline{\text{DONE}}$ 位设置为 1，可以启动模数转换。

注：	不应该使用启动 ADC 的同一指令将 GO/$\overline{\text{DONE}}$ 位置 1。请参见**第 15.2.5 节 “A/D 转换步骤”**。

15.2.2 完成转换

当转换完成时， ADC 模块将：

- 清零 GO/$\overline{\text{DONE}}$ 位
- 置 1ADIF 中断标志位
- 使用新的转换结果更新 ADRES 寄存器

15.2.3 终止转换

如果必须在转换完成前终止转换，则可用软件清零 GO/$\overline{\text{DONE}}$ 位。 ADRES 寄存器将使用部分完成的模数转换结果进行更新。未完成位将用最后转换的一位填充。

注：	器件复位会强制所有寄存器进入复位状态。因此关闭 ADC 模块并终止任何待处理的转换。

15.2.4 休眠期间的 ADC 工作原理

ADC 模块可以在休眠模式下工作。这需要将 ADC 时钟源设置为 FRC 选项。当选择 FRC 时钟源时， ADC 需要等待一个额外的指令周期才能开始转换。这允许执行一条 SLEEP 指令，从而降低转换期间的系统噪声。如果允许 ADC 中断，则转换完成时器件将从休眠模式中唤醒。如果禁止 ADC 中断，即使 ADON 位保持置 1，但是在转换完成后 ADC 模块会关闭。

当 ADC 时钟源不是 FRC，而是其他时钟源时，即使 ADON 位保持置 1，但是 SLEEP 指令会导致当前转换中止且 ADC 模块关闭。

© 2011-2014 Microchip Technology Inc.

15.2.5 A/D 转换步骤

以下是使用 ADC 执行模数转换的示例步骤：

1. 配置端口：
   - 禁用引脚输出驱动器 （见 TRIS 寄存器）
   - 将引脚配置为模拟引脚 （见 ANSEL 寄存器）
2. 配置 ADC 模块：
   - 选择 ADC 转换时钟
   - 选择 ADC 输入通道
   - 启动 ADC 模块
3. 配置 ADC 中断 （可选）：
   - 清零 ADC 中断标志
   - 允许 ADC 中断
   - 允许外设中断
   - 允许全局中断 (1)
4. 等待所需的采集时间 (2)。
5. 通过将 GO/$\overline{\text{DONE}}$ 位置 1 来启动转换。
6. 通过以下任一方式等待 ADC 转换完成：
   - 查询 GO/$\overline{\text{DONE}}$ 位
   - 等待 ADC 中断 （允许中断）
7. 读 ADC 结果。
8. 清零 ADC 中断标志 （如果允许中断，则需要进行此操作）。

注	**1：** 如果用户尝试将器件从休眠模式中唤醒并继续执行主程序代码，则应禁止全局中断。
	2： 请参见**第 15.4 节 “A/D 采集要求”**。

15.3 ADC 寄存器定义

以下寄存器用于控制 ADC 的工作。

寄存器 15-1： ADCON：A/D 控制寄存器 0

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ADCS<2:0>			CHS<2:0>			GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-5 **ADCS<2:0>：**A/D 转换时钟选择位
111 = FRC
110 = FOSC/64
101 = FOSC/16
100 = FOSC/4
011 = FRC
010 = FOSC/32
001 = FOSC/8
000 = FOSC/2

bit 4-2 **CHS<2:0>：**模拟通道选择位
111 = FVR （固定参考电压）缓冲器输出 (2)
110 = 温度指示器 (1)
101 = 保留。未连接通道。
100 = 保留。未连接通道。
011 = 保留。未连接通道。
010 = AN2
001 = AN1
000 = AN0

bit 1 **GO/$\overline{\text{DONE}}$：**A/D 转换状态位
如果 ADON = 1：
1 = A/D 转换正在进行 （将该位置 1 可启动 A/D 转换）
0 = A/D 转换未进行 （当 A/D 转换完成时，该位由硬件自动清零。）
如果该位在转换正在进行时清零，则转换将停止，此时的转换结果将传送到结果寄存器，但是 ADIF 中断标志位不会置 1。
如果 ADON = 0：
0 = A/D 转换未进行

bit 0 **ADON：**ADC 使能位
1 = 使能 ADC
0 = 禁止 ADC 且不消耗工作电流

注 1： 更多信息，请参见**第 14.0 节 “温度指示器模块”**。
2： 更多信息，请参见**第 12.0 节 “固定参考电压 （FVR）”**。

© 2011-2014 Microchip Technology Inc.

寄存器 15-2： ADRES：ADC 结果寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<7:0>							
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0 **ADRES<7:0>：**ADC 结果寄存器位
8 位结果

15.4 A/D 采集要求

为了使 ADC 达到规定精度，必须使充电保持电容（CHOLD）完全充电至输入通道的电平。模拟输入模型如图 15-3 所示。信号源阻抗（RS）和内部采样开关（RSS）阻抗直接影响电容 CHOLD 的充电时间。采样开关（RSS）阻抗随着器件电压（VDD）的不同而不同，请参见图15-3。**模拟信号源的最大阻抗推荐值为10 kΩ。**

采集时间随着信号源阻抗的减少而缩短。选择（或更改）模拟输入通道后，必须在转换开始之前完成 A/D 采集。采用公式 15-1 计算最小采集时间。该公式假定误差为 1/2 LSb（ADC 转换需要 511 步）。1/2 LSb 的误差是 ADC 达到规定分辨率所允许的最大误差。

公式 15-1：采集时间示例

假设： *温度 = 50°C，外部阻抗为 10 kΩ，VDD 为 5.0V*

$$T_{ACQ} = \text{放大器稳定时间} + \text{保持电容充电时间} + \text{温度系数}$$
$$= T_{AMP} + T_C + T_{COFF}$$
$$= 2\ \mu s + T_C + [\ (\text{温度} - 25°C)(0.05\ \mu s/°C)\ ]$$

采用以下公式计算 TC 的近似值：

$$V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) = V_{CHOLD}$$ ；[1] 充电到 VCHOLD（1/2 LSb 误差范围）

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{CHOLD}$$ ；[2] 响应 VAPPLIED 充电到 VCHOLD

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

注：其中 n = ADC 的位数。

求解 TC：

$$T_C = -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ ln(1/511)$$
$$= -10\ pF(1\ k\Omega + 7\ k\Omega + 10\ k\Omega)\ 1n(0.001957)$$
$$= 1.12\ \mu s$$

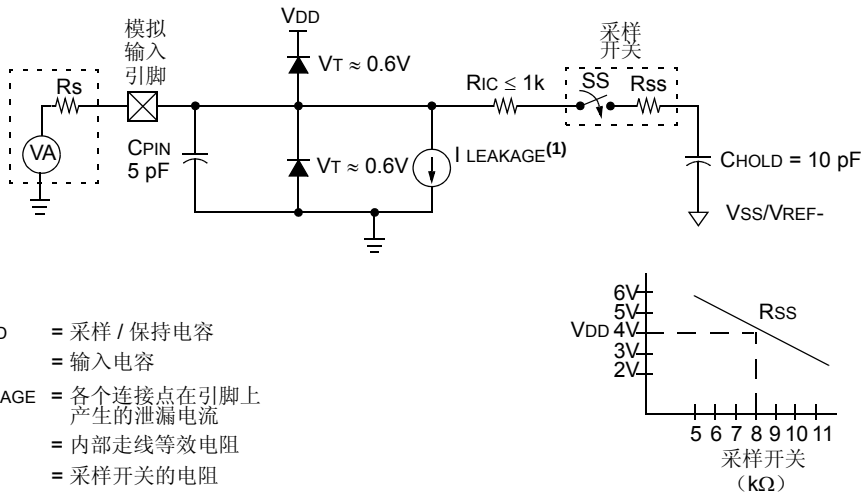
因此：

$$T_{ACQ} = 2\ \mu s + 1.12\ \mu s + [(50°C - 25°C)(0.05\ \mu s/°C)]$$
$$= 4.37\ \mu s$$

注 1： 参考电压（VREF）自行抵消，因此它对该公式没有影响。
2： 每次转换后，充电保持电容（CHOLD）不会放电。
3： 模拟信号源的最大推荐阻抗为 10 kΩ。此要求是为了符合引脚泄漏电流规范。

© 2011-2014 Microchip Technology Inc.

图 15-3： 模拟输入模型



图注： CHOLD = 采样 / 保持电容
CPIN = 输入电容
I LEAKAGE = 各个连接点在引脚上产生的泄漏电流
RIC = 内部走线等效电阻
RSS = 采样开关的电阻
SS = 采样开关
VT = 阈值电压

注 1： 请参见第 24.0 节 “电气规范”。

图 15-4： ADC 传递函数

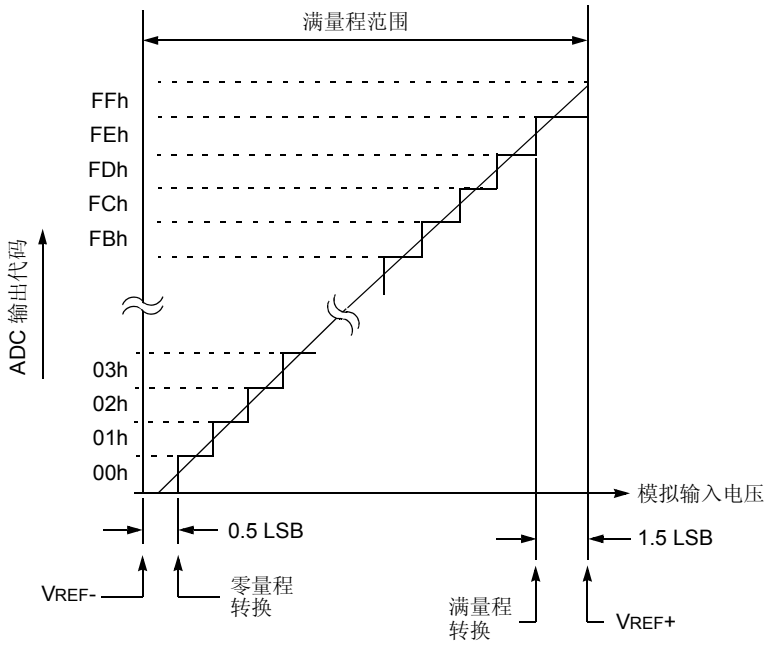


表 15-2： 与 ADC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ADCON	ADCS<2:0>			CHS<2:0>			GO/$\overline{\text{DONE}}$	ADON	88
ADRES	ADRES<7:0>								89
ANSELA	—	—	—	—	—	ANSA2	ANSA1	ANSA0	70
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	—	—	ADFVR<1:0>		78
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
PIE1	—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—	41
PIR1	—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—	42
TRISA	—	—	—	—	—	TRISA2	TRISA1	TRISA0	69

图注： x = 未知； u = 不变； — = 未实现，读为 0； q = 值取决于具体条件。 ADC 模块不使用阴影单元。

© 2011-2014 Microchip Technology Inc.

16.0 TIMER0 模块

Timer0 模块是具有以下功能的 8 位定时器 / 计数器：

- 8 位定时器 / 计数器寄存器 （TMR0）
- 8 位预分频器 （独立于看门狗定时器）
- 可编程的内部或外部时钟源
- 可编程的外部时钟边沿选择
- 溢出时中断

图 16-1 是 Timer0 模块的框图。

16.1 Timer0 工作原理

Timer0 模块可以用作 8 位定时器或 8 位计数器。

16.1.1 8 位定时器模式

如果不使用预分频器，则 Timer0 模块在每个指令周期递增 1。通过将 OPTION_REG 寄存器的 T0CS 位清零，可以选择 8 位定时器模式。

当写入 TMR0 时，在接下来的两个指令周期内 TMR0 都不会递增 1。

注：	考虑到写入 TMR0 后两个指令周期的延时，可以调整写入 TMR0 寄存器的值。

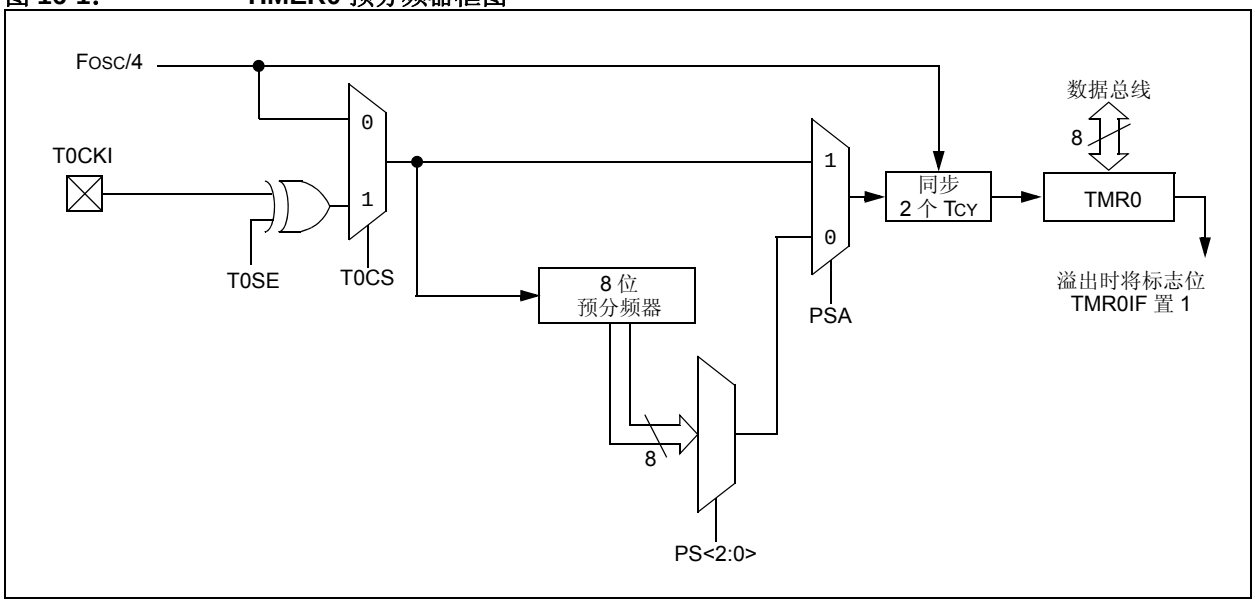
16.1.2 8 位计数器模式

在 8 位计数器模式下，Timer0 模块将在 T0CKI 引脚信号的每个上升沿或下降沿进行递增计数。

通过将 OPTION_REG 寄存器的 T0CS 位设置为 1，可以选择使用 T0CKI 引脚的 8 位计数器模式。

外部输入源的递增沿是上升沿还是下降沿由 OPTION_REG 寄存器中的 T0SE 位确定。

图 16-1： TIMER0 预分频器框图



16.1.3 软件可编程预分频器

软件可编程预分频器可用于 Timer0。OPTION_REG 寄存器的PSA位控制预分频器分配。要将预分频器分配给 Timer0， PSA 位必须清零。

Timer0 模块具有 8 种预分频比选择，范围为 1:2 至 1:256。可通过 OPTION_REG 寄存器中的 PS<2:0> 位选择预分频值。

该预分频器不可读写。当分配给 Timer0 模块时，写入 TMR0 寄存器的所有指令将清零预分频器。

16.1.4 TIMER0 中断

当 TMR0 寄存器从 FFh 溢出至 00h 时，Timer0 会产生中断。每次 TMR0 寄存器溢出时，不管是否允许 Timer0 中断，INTCON 寄存器的 TMR0IF 中断标志位都将置 1。只能用软件将 TMR0IF 位清零。 Timer0 中断允许位是 INTCON 寄存器的 TMR0IE 位。

注：	由于在休眠状态下定时器是关闭的，因此 Timer0 中断无法唤醒处理器。

16.1.5 8 位同步计数器模式

在 8 位计数器模式下，必须使 T0CKI 引脚信号的递增沿与指令时钟同步。通过在指令时钟的 Q2 和 Q4 周期采样预分频器输出，可以实现同步。外部时钟源的高电平和低电平周期必须符合**第 24.0 节 “电气规范”**中的时序要求。

© 2011-2014 Microchip Technology Inc.

寄存器 16-1： OPTION_REG：选项寄存器

R/W-1/u	R/W-1/u	R/W-1/u	R/W-1/u	R/W-1/u	R/W-1/u	R/W-1/u	R/W-1/u
$\overline{\text{WPUEN}}$(1)	INTEDG	T0CS	T0SE	PSA	PS<2:0>		
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 **$\overline{\text{WPUEN}}$：**弱上拉使能位 (1)
1 = 禁止弱上拉
0 = 根据各端口锁存值使能弱上拉

bit 6 **INTEDG：**中断边沿选择位
1 = INT 引脚信号的上升沿触发中断
0 = INT 引脚信号的下降沿触发中断

bit 5 **T0CS：**TMR0 时钟源选择位
1 = T0CKI 引脚上信号的跳变
0 = 内部指令周期时钟 （FOSC/4）

bit 4 **T0SE：**TMR0 时钟源边沿选择位
1 = 在 T0CKI 引脚电平发生由高到低的跳变时递增
0 = 在 T0CKI 引脚电平发生由低到高的跳变时递增

bit 3 **PSA：**预分频器分配位
1 = 预分频器无效且对 Timer0 模块无影响
0 = 将预分频器分配给 Timer0 模块

bit 2-0 **PS<2:0>：**预分频器分频比选择位

位值	TMR0 分频比
000	1 : 2
001	1 : 4
010	1 : 8
011	1 : 16
100	1 : 32
101	1 : 64
110	1 : 128
111	1 : 256

注 1： 当 $\overline{\text{MCLR}}$ = 1 时， $\overline{\text{WPUEN}}$ 不会禁止 $\overline{\text{MCLR}}$ 输入的上拉。

表 16-1： 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	寄存器所在页
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	T0CS	T0SE	PSA	PS<2:0>			95
TMR0	Timer0 模块寄存器								46
TRISA	—	—	—	—	—	TRISA2	TRISA1	TRISA0	69

图注： — = 未实现单元，读为 0； u = 未变； x = 未知。 Timer0 模块不使用阴影单元。

17.0 TIMER2 模块

Timer2 模块是具有以下功能的 8 位定时器：

- 8 位定时器寄存器 （TMR2）
- 8 位周期寄存器 （PR2）
- TMR2 与 PR2 匹配时发生中断
- 可软件编程的预分频器 （预分频比为 1:1、 1:4、 1:16 和 1:64）
- 可软件编程的后分频器 （后分频比为 1:1 至 1:16）

请参见图 17-1 中的 Timer2 框图。

17.1 Timer2 工作原理

Timer2 模块的时钟输入是系统指令时钟（Fosc/4）。时钟输出传送至 Timer2 预分频器，该预分频器具有 1:1、1:4 和 1:64 预分频比选项。然后，预分频器的输出用于递增 TMR2 寄存器。

不断比较 TMR2 和 PR2 的值以确定它们何时匹配。TMR2 将从 00h 递增直至它与 PR2 中的值匹配。当匹配发生时，会发生以下两种情况：

- TMR2 在下一个递增周期复位为 00h。
- Timer2 后分频器递增。

然后，Timer2/PR2 比较器的匹配输出传送至 Timer2 后分频器。后分频器具有 1:1 至 1:16（包括 1:1 和 1:16）的后分频比选项。 Timer2 后分频器的输出用于将 PIR1 寄存器中的 TMR2IF 中断标志位置 1。

TMR2 和 PR2 寄存器都是可完全读写的。发生任何复位时， TMR2 寄存器设置为 00h，而 PR2 寄存器设置为 FFh。

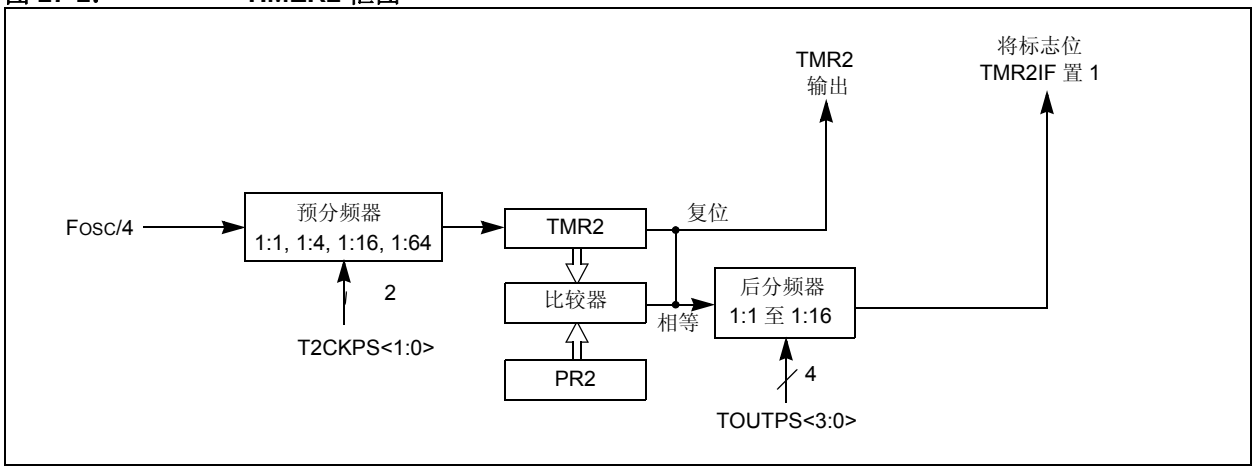
通过将 T2CON 寄存器中的 TMR2ON 位设置为 1，可以使能Timer2。通过清零TMR2ON位，可以禁止Timer2。

Timer2 预分频器由 T2CON 寄存器中的 T2CKPS 位控制。 Timer2 后分频器由 T2CON 寄存器中的 TOUTPS 位控制。发生以下事件时，预分频计数器和后分频计数器都将清零：

- 对 TMR2 进行写操作。
- 对 T2CON 进行写操作。
- 发生任一器件复位 （上电复位、 $\overline{\text{MCLR}}$ 复位、看门狗定时器复位或欠压复位）。

注：	写 T2CON 时 TMR2 不会清零。

图 17-1： TIMER2 框图



© 2011-2014 Microchip Technology Inc.

寄存器 17-1： T2CON：TIMER2 控制寄存器

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	TOUTPS<3:0>				TMR2ON	T2CKPS<1:0>	
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 未实现：读为 0

bit 6-3 **TOUTPS<3:0>**：Timer2 输出后分频比选择位
1111 = 1:16 后分频比
1110 = 1:15 后分频比
1101 = 1:14 后分频比
1100 = 1:13 后分频比
1011 = 1:12 后分频比
1010 = 1:11 后分频比
1001 = 1:10 后分频比
1000 = 1:9 后分频比
0111 = 1:8 后分频比
0110 = 1:7 后分频比
0101 = 1:6 后分频比
0100 = 1:5 后分频比
0011 = 1:4 后分频比
0010 = 1:3 后分频比
0001 = 1:2 后分频比
0000 = 1:1 后分频比

bit 2 **TMR2ON**：Timer2 使能位
1 = 使能 Timer2
0 = 禁止 Timer2

bit 1-0 **T2CKPS<1:0>**：Timer2 时钟预分频值选择位
11 = 预分频值为 64
10 = 预分频值为 16
01 = 预分频值为 4
00 = 预分频值为 1

表 17-1： 与 TIMER2 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
PIE1	—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—	41
PIR1	—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—	42
PR2	Timer2 模块周期寄存器								96
TMR2	Timer2 模块寄存器								96
T2CON	—	TOUTPS<3:0>				TMR2ON	T2CKPS<1:0>		97

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

18.0 脉宽调制 （PWM）模块

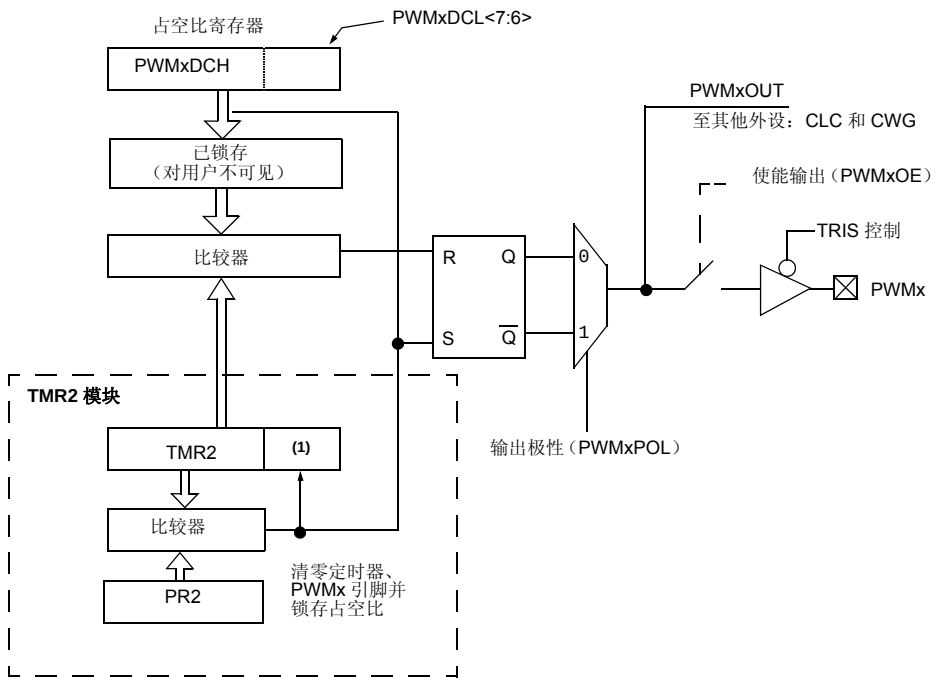
PWM 模块产生脉宽调制信号，该信号由通过以下寄存器配置的占空比、周期和分辨率确定：

- PR2
- T2CON
- PWMxDCH
- PWMxDCL
- PWMxCON

图 18-1 给出了 PWM 操作的简化框图。

图 18-2 给出了 PWM 信号的典型波形。

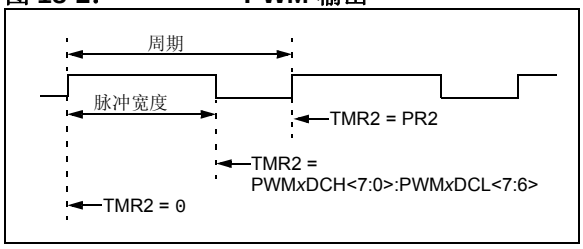
图 18-1： PWM 简化框图



注 1： 8 位定时器与通过 Timer2 预分频器调整的 1/FOSC 的低 2 位一起构成 10 位时基。

关于如何设置该模块进行 PWM 操作的详细步骤，请参见**第 18.1.9 节 “使用 PWMx 引脚设置 PWM 工作模式”**。

图 18-2： PWM 输出



© 2011-2014 Microchip Technology Inc.

18.1 PWMx 引脚配置

所有 PWM 输出均与端口数据锁存器复用。用户必须通过清零相关的 TRIS 位将引脚配置为输出。

> **注：** 清零 PWMxOE 位将放弃对 PWMx 引脚的控制。

18.1.1 基本工作原理

PWM 模块产生 10 位分辨率的输出。Timer2 和 PR2 设置 PWM 的周期。 PWMxDCL 和 PWMxDCH 寄存器配置占空比。周期对所有 PWM 模块都是通用的，而占空比是独立控制的。

> **注：** Timer2 后分频器不用于确定 PWM 频率。使用该后分频器时，其伺服更新速率不同于 PWM 输出频率。

当清零 TMR2 时，所有与 Timer2 相关的 PWM 输出均置 1。当 TMR2 等于相应的 PWMxDCH （高 8 位）寄存器和 PWMxDCL<7:6> （低 2 位）寄存器中指定的值时，清零所有 PWMx。当值大于或等于 PR2 时，永不清零 PWM 输出 （100% 占空比）。

> **注：** PWMxDCH 和 PWMxDCL 寄存器均为双缓冲。当 Timer2 与 PR2 匹配时，将更新这两个缓冲区。在定时器匹配发生前应小心更新这两个寄存器。

18.1.2 PWM 输出极性

通过将 PWMxCON 寄存器的 PWMxPOL 位置 1 来翻转输出极性。

18.1.3 PWM 周期

PWM 周期由 Timer2 的 PR2 寄存器指定。采用公式 18-1 计算 PWM 周期。

公式 18-1： PWM 周期

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

注： TOSC = 1/FOSC

当 TMR2 等于 PR2 时，在下一个递增计数周期中将发生以下三个事件：

- 清零 TMR2
- PWM 输出有效。（例外情况：当 PWM 占空比为 0% 时， PWM 输出将保持无效状态。）
- PWMxDCH 和 PWMxDCL 寄存器值锁存到缓冲器中。

> **注：** Timer2 后分频器对 PWM 操作无任何影响。

18.1.4 PWM 占空比

通过将 10 位值写入 PWMxDCH 和 PWMxDCL 寄存器对指定 PWM 占空比。PWMxDCH 寄存器包含高 8 位，而 PWMxDCL<7:6> 包含低 2 位。可以在任何时候写入 PWMxDCH 和 PWMxDCL 寄存器。

公式 18-2 用于计算 PWM 脉冲宽度。

公式 18-3 用于计算 PWM 占空比。

公式 18-2： 脉冲宽度

$$脉冲宽度 = (PWMxDCH{:}PWMxDCL<7{:}6>) \bullet T_{OSC} \bullet (TMR2\ 预分频值)$$

注： TOSC = 1/FOSC

公式 18-3： 占空比

$$占空比 = \frac{(PWMxDCH{:}PWMxDCL<7{:}6>)}{4(PR2 + 1)}$$

8 位定时器 TMR2 寄存器与通过 Timer2 预分频器调整的 1/FOSC 的低 2 位一起构成 10 位时基。如果将 Timer2 预分频比设置为 1:1，则使用系统时钟。

18.1.5 PWM 分辨率

分辨率决定在给定周期内的可用占空比数。例如，10 位分辨率将产生 1024 个离散的占空比，而 8 位分辨率将产生 256 个离散的占空比。

PR2 为 255 时， PWM 的最高分辨率为 10 位。分辨率是 PR2 寄存器值的函数，如公式 18-4 所示。

公式 18-4： PWM 分辨率

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

> **注：** 如果脉宽值大于周期值，则指定的 PWM 引脚电平将保持不变。

表 18-1： PWM 频率与分辨率示例 （FOSC = 20 MHz）

PWM 频率	0.31 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
定时器预分频值（1、4 或 64）	64	4	1	1	1	1
PR2 值	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最高分辨率 （位）	10	10	10	8	7	6.6

表 18-2： PWM 频率与分辨率示例 （FOSC = 8 MHz）

PWM 频率	0.31 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
定时器预分频值（1、4 或 64）	64	4	1	1	1	1
PR2 值	0x65	0x65	0x65	0x19	0x0C	0x09
最高分辨率 （位）	8	8	8	6	5	5

18.1.6 休眠模式下的工作原理

在休眠模式下， TMR2 寄存器将不会递增，并且模块的状态将保持不变。如果 PWMx 引脚有输出，它将继续保持该输出值不变。当器件被唤醒时， TMR2 将从先前的状态继续工作。

18.1.7 系统时钟频率的改变

PWM 频率是由系统时钟频率（FOSC）产生的。系统时钟频率发生任何变化都会使 PWM 频率发生变化。更多信息，请参见**第 4.0 节 “振荡器模块”**。

18.1.8 复位的影响

任何复位都会将所有端口强制为输入模式，并强制 PWM 寄存器进入复位状态。

© 2011-2014 Microchip Technology Inc.

18.1.9 使用 PWMx 引脚设置 PWM 工作模式

当使用 PWMx 引脚配置该模块使之工作于 PWM 模式时，应遵循以下步骤：

1. 通过将相应的 TRIS 位置 1 禁止 PWMx 引脚输出驱动器。
2. 清零 PWMxCON 寄存器。
3. 将 PWM 周期值装入 PR2 寄存器。
4. 清零 PWMxDCH 寄存器和 PWMxDCL 寄存器的 bit<7:6>。
5. 配置并启动 Timer2：
   - 清零 PIR1 寄存器的 TMR2IF 中断标志位。请参见下面的 “注”。
   - 使用 Timer2 预分频值配置 T2CON 寄存器的 T2CKPS 位。
   - 通过将 T2CON 寄存器的 TMR2ON 位置 1 来使能 Timer2。
6. 使能 PWM 输出引脚并等待直到 Timer2 溢出，此时 PIR1 寄存器的 TMR2IF 位置 1。请参见下面的 “注”。
7. 通过将相应的 TRIS 位清零并将 PWMxCON 寄存器的 PWMxOE 位置 1，可以使能 PWMx 引脚输出驱动器。
8. 通过将合适的值装入 PWMxCON 寄存器来配置 PWM 模块。

注 1： 为了在第一个 PWM 输出中发送完整的占空比和周期，必须按照给定顺序执行上述步骤。如果以完整的 PWM 信号开始不是很重要，那么可以用步骤 8 替换步骤 4。

2： 为了仅使用其他外设进行工作，请禁止 PWMx 引脚输出。

18.2 PWM 寄存器定义

寄存器 18-1： PWMxCON：PWM 控制寄存器

R/W-0/0	R/W-0/0	R-0/0	R/W-0/0	U-0	U-0	U-0	U-0
PWMxEN	PWMxOE	PWMxOUT	PWMxPOL	—	—	—	—
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 **PWMxEN**：PWM 模块使能位
1 = 使能 PWM 模块
0 = 禁止 PWM 模块

bit 6 **PWMxOE**：PWM 模块输出使能位
1 = 使能 PWMx 引脚输出
0 = 禁止 PWMx 引脚输出

bit 5 **PWMxOUT**：PWM 模块输出值位

bit 4 **PWMxPOL**：PWMx 输出极性选择位
1 = PWM 输出为低电平有效
0 = PWM 输出为高电平有效

bit 3-0 未实现：读为 0

© 2011-2014 Microchip Technology Inc.

寄存器 18-2： PWMxDCH：PWM 占空比高位

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PWMxDCH<7:0>							
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0 **PWMxDCH<7:0>：**PWM 占空比高位
这些位是 PWM 占空比的高位。低 2 位在 PWMxDCL 寄存器中。

寄存器 18-3： PWMxDCL：PWM 占空比低位

R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0	U-0	U-0
PWMxDCL<7:6>		—	—	—	—	—	—
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-6 **PWMxDCL<7:6>：**PWM 占空比低位
这些位是 PWM 占空比的低位。高位在 PWMxDCH 寄存器中。

bit 5-0 未实现：读为 0

表 18-3： 与 PWM 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在页
ANSELA	—	—	—	—	—	ANSA2	ANSA1	ANSA0	70
LATA	—	—	—	—	—	LATA2	LATA1	LATA0	70
PORTA	—	—	—	—	RA3	RA2	RA1	RA0	69
PR2	Timer2 模块周期寄存器								96
PWM1CON	PWM1EN	PWM1OE	PWM1OUT	PWM1POL	—	—	—	—	102
PWM1DCH	PWM1DCH<7:0>								103
PWM1DCL	PWM1DCL<7:6>		—	—	—	—	—	—	103
PWM2CON	PWM2EN	PWM2OE	PWM2OUT	PWM2POL	—	—	—	—	102
PWM2DCH	PWM2DCH<7:0>								103
PWM2DCL	PWM2DCL<7:6>		—	—	—	—	—	—	103
T2CON	—	TOUTPS<3:0>				TMR2ON	T2CKPS<1:0>		97
TMR2	Timer2 模块寄存器								96
TRISA	—	—	—	—	—	TRISA2	TRISA1	TRISA0	69

图注： - = 未实现单元，读为 0； u = 不变， x = 未知。 PWM 不使用阴影单元。

19.0 可配置逻辑单元（CLC）

可配置逻辑单元（CLCx）提供可在软件执行速度限制之外工作的可编程逻辑。此逻辑单元可选择八个输入信号的任意组合，并通过使用可配置的门将可选输入减少至四根逻辑线，这些逻辑线可驱动八个可选单输出逻辑功能中的一个。

输入源为以下信号的组合：

- 两个 I/O 引脚
- 内部时钟
- 外设
- 寄存器位

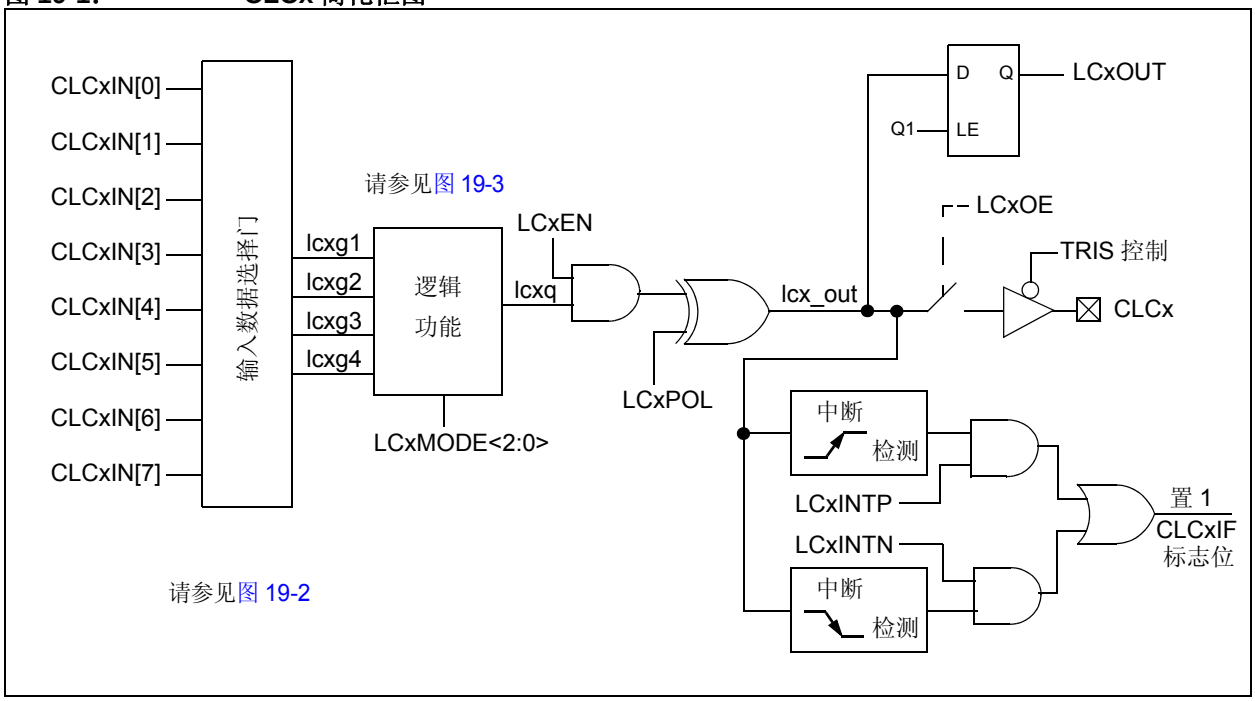
可将输出内部连接到外设和一个输出引脚。

请参见图19-1了解显示通过CLCx的信号流的简化框图。

可行的配置包括：

- 组合逻辑
  - AND
  - NAND
  - AND-OR
  - AND-OR-INVERT
  - OR-XOR
  - OR-XNOR
- 锁存器
  - S-R
  - 带置 1 和复位功能的时钟控制 D 型锁存器
  - 带置 1 和复位功能的透明 D 型锁存器
  - 带复位功能的时钟控制 J-K 型锁存器

图 19-1： CLCx 简化框图



© 2011-2014 Microchip Technology Inc.

19.1 CLCx 设置

通过配置逻辑信号流中的四个阶段，可以对 CLCx 模块进行编程。这四个阶段为：

- 数据选择
- 数据门控
- 逻辑功能选择
- 输出极性

每个阶段都可在运行时通过写入相应的 CLCx 特殊功能寄存器进行设置。这允许在编程期间随时重新配置逻辑，从而提供了额外的好处。

19.1.1 数据选择

有八个信号可作为可配置逻辑的输入。4 个 8 输入的多路开关用于选择传递到下一阶段的输入。

数据输入可以使用 CLCxSEL0 和 CLCxSEL1 寄存器（分别为寄存器 19-3 和寄存器 19-4）来选择。

如图 19-2 的左侧所示，通过四个多路开关进行数据选择。图中的数据输入由通用编码的输入名称进行标记。

表 19-1 给出了每个 CLC 模块中的通用输入名称与实际信号之间的关联。标为 lcxd1 至 lcxd4 的列表示所选数据输入的 MUX 输出。D1S 至 D4S 是 MUX 选择输入编码的缩写：分别为 LCxD1S<2:0> 至 LCxD4S<2:0>。选择了某列中的一个数据输入即排除了该列中的其他输入。

注：	数据选择在上电时未定义。

表 19-1： CLCx 数据输入选择

数据输入	lcxd1 D1S	lcxd2 D2S	lcxd3 D3S	lcxd4 D4S	CLC 1
CLCxIN[0]	000	000	000	000	CLCx
CLCxIN[1]	001	001	001	001	CLCxIN1
CLCxIN[2]	010	010	010	010	CLCxIN2
CLCxIN[3]	011	011	011	011	PWM1
CLCxIN[4]	100	100	100	100	PWM2
CLCxIN[5]	101	101	101	101	NCOx
CLCxIN[6]	110	110	110	110	FOSC
CLCxIN[7]	111	111	111	111	LFINTOSC

19.1.2 数据门控

输入多路开关的输出通过数据门控阶段与所需的逻辑功能输入相连。每个数据门控都可以控制四个选定输入的任意组合。

注：	数据门控在上电时未定义。

门阶段不仅仅是信号方向。可将门配置为将每个输入信号指定为将数据反相或同相。在每个门中，将连接的信号进行与操作。在传送到逻辑功能阶段之前，可将每个门的输出反相。

门控本质上就是一个1至4输入的AND/NAND/OR/NOR门。当将每个输入和输出反相时，该门的作用为对所有使能数据输入进行或操作。当将输入和输出同相时，门的作用是将所有使能输入进行与操作。

表 19-2 总结了通过使用门逻辑选择位在门 1 中可获得的基本逻辑。该表给出了四个输入变量的逻辑，但是可将每个门配置为使用少于四个的变量。如果不选择任何输入，根据门输出极性位的不同，输出为 0 或 1。

表 19-2： 数据门控逻辑

CLCxGLS0	LCxGyPOL	门逻辑
0x55	1	AND
0x55	0	NAND
0xAA	1	NOR
0xAA	0	OR
0x00	0	逻辑 0
0x00	1	逻辑 1

可以（但不推荐）选择同一个输入的正负值。如果是这样的话，不管其它输入值如何，门输出总是为 0，但是可能引发逻辑故障（瞬态电流引起的脉冲）。如果通道的输出必须为 0 或 1，推荐的方法是将所有门位设置为 0，并使用门信号极性位设置所需的电平。

使用以下逻辑门选择寄存器配置数据门控：

- 门 1：CLCxGLS0（寄存器 19-5）
- 门 2：CLCxGLS1（寄存器 19-6）
- 门 3：CLCxGLS2（寄存器 19-7）
- 门 4：CLCxGLS3（寄存器 19-8）

寄存器编号后缀与门编号不同，这是因为该模块的其他形式在同一寄存器中拥有多种门选择。

数据门控如图 19-2 的右侧所示。其中仅详细说明了一个门。剩下的 3 个门配置相同，只是数据使能信号对应相应的门使能信号。

19.1.3 逻辑功能

八个可用的逻辑功能包括：

- AND-OR
- OR-XOR
- AND
- S-R 锁存器
- 带置 1 和复位功能的 D 型触发器
- 带复位功能的 D 型触发器
- 带复位功能的 J-K 型触发器
- 带置 1 和复位功能的透明锁存器

逻辑功能如图 19-3 所示。每个逻辑功能都有四个输入和一个输出。四个输入为上一阶段四个数据门的输出。输出传送到反相阶段，接着到其他外设、输出引脚，最后返回 CLCx。

19.1.4 输出极性

可配置逻辑单元的最后一个阶段是输出极性。通过将 CLCxCON 寄存器的 LCxPOL 位置 1，可以将逻辑阶段的输出信号反相。允许中断时更改极性会导致中断结果输出转换。

19.1.5 CLCx 设置步骤

当设置 CLCx 时，应遵循以下步骤：

- 通过清零 LCxEN 位禁止 CLCx。
- 使用 CLCxSEL0 和 CLCxSEL1 寄存器选择所需的输入（见表 19-1）。
- 将所有相关的 ANSEL 位清零。
- 将所有与输入相关的 TRIS 位置 1。
- 将所有与输出相关的 TRIS 位清零。
- 使用 CLCxGLS0、 CLCxGLS1、 CLCxGLS2 和 CLCxGLS3 寄存器通过四个门来使能选中的输入。
- 使用CLCxPOL寄存器的LCxPOLy位选择门输出极性。
- 使用 CLCxCON 寄存器的 LCxMODE<2:0> 位选择所需的逻辑功能。
- 使用 CLCxPOL 寄存器的 LCxPOL 位选择所需的逻辑输出极性。（该步骤可与先前的门输出极性步骤相结合）。
- 如果驱动 CLCx 引脚，将 CLCxCON 寄存器的 LCxOE 位置1，并将与该输出对应的TRIS位清零。
- 如果需要中断，需配置以下位：
  - 上升事件时，将 CLCxCON 寄存器中的 LCxINTP 位置 1
  - 下降事件时，将 CLCxCON 寄存器中的 LCxINTN 位置 1
  - 将相关的 PIE 寄存器的 CLCxIE 位置 1。
  - 将 INTCON 寄存器的 GIE 和 PEIE 位置 1。
- 通过将 CLCxCON 寄存器的 LCxEN 位置 1 来使能 CLCx。

© 2011-2014 Microchip Technology Inc.

19.2 CLCx 中断

当相应的中断允许位置 1 时，在改变 CLCx 输出值时将产生中断。因此，每个 CLC 都有一个上升沿检测器和下降沿检测器。

当触发任意一个边沿检测器且其相应使能位置 1 时，相关 PIR 寄存器的 CLCxIF 位将置 1。LCxINTP 位允许上升沿中断，LCxINTN 位允许下降沿中断。这两位都位于 CLCxCON 寄存器中。

要完全允许中断，必须将以下各位置 1：

- CLCxCON 寄存器的 LCxON 位
- 相关 PIE 寄存器的 CLCxIE 位
- CLCxCON 寄存器的 LCxINTP 位（用于上升沿检测）
- CLCxCON 寄存器的 LCxINTN 位（用于下降沿检测）
- INTCON 寄存器的 PEIE 和 GIE 位

作为中断服务的一部分，必须由软件清零相关 PIR 寄存器的 CLCxIF 位。如果在该标志位清零期间检测到另一个边沿，则在该序列结束时仍将该标志位置 1。

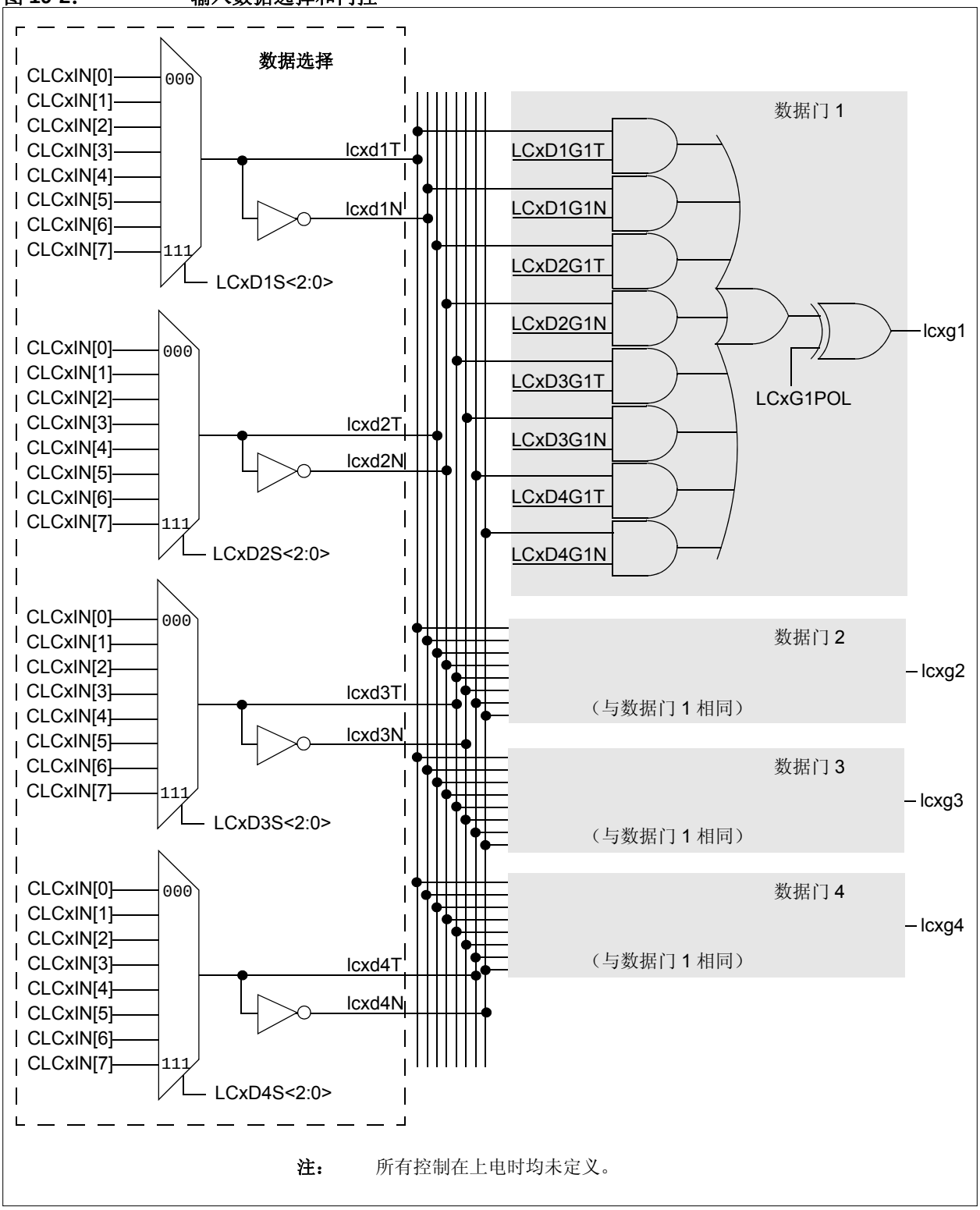
19.3 复位的影响

复位将导致 CLCxCON 寄存器清零。所有其他选择和门控值将保持不变。

19.4 休眠期间的工作原理

选择、门控以及逻辑功能不受休眠的影响。如果源信号也不受休眠的影响，则将继续执行操作。

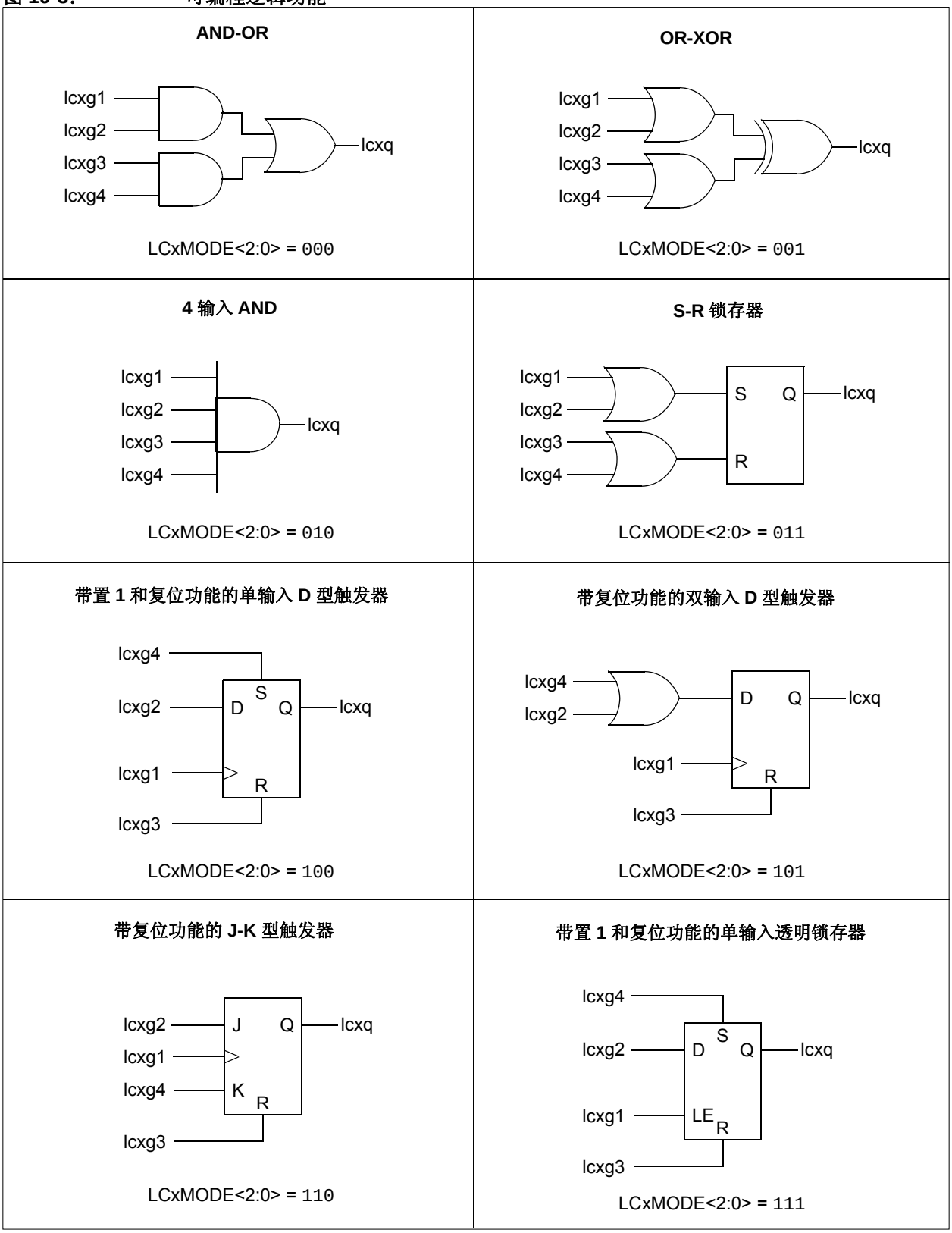
图 19-2： 输入数据选择和门控



注： 所有控制在上电时均未定义。

© 2011-2014 Microchip Technology Inc.

图 **19-3：** 可编程逻辑功能



19.5 CLC 控制寄存器

寄存器 19-1： CLCxCON：可配置逻辑单元控制寄存器

R/W-0/0	R/W-0/0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
LCxEN	LCxOE	LCxOUT	LCxINTP	LCxINTN	LCxMODE<2:0>		
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 **LCxEN：**可配置逻辑单元使能位
1 = 使能可配置逻辑单元并混合输入信号
0 = 禁止可配置逻辑单元，输出逻辑零

bit 6 **LCxOE：**可配置逻辑单元输出使能位
1 = 使能可配置逻辑单元端口引脚输出
0 = 禁止可配置逻辑单元端口引脚输出

bit 5 **LCxOUT：**可配置逻辑单元数据输出位
只读：经 LCxPOL 后逻辑单元输出数据；从 lcx_out 线采样数据。

bit 4 **LCxINTP：**可配置逻辑单元上升沿中断允许位
1 = 当 lcx_out 出现上升沿时， CLCxIF 将置 1
0 = CLCxIF 不会置 1

bit 3 **LCxINTN：**可配置逻辑单元下降沿中断允许位
1 = 当 lcx_out 出现下降沿时， CLCxIF 将置 1
0 = CLCxIF 不会置 1

bit 2-0 **LCxMODE<2:0>：**可配置逻辑单元功能模式位
111 = 单元为带置 1 和复位功能的单输入透明锁存器
110 = 单元为带复位功能的 J-K 型触发器
101 = 单元为带复位功能的双输入 D 型触发器
100 = 单元为带置 1 和复位功能的单输入 D 型触发器
011 = 单元为 S-R 锁存器
010 = 单元为 4 输入 AND 门
001 = 单元为 OR-XOR 门
000 = 单元为 AND-OR 门

© 2011-2014 Microchip Technology Inc.

寄存器 19-2： CLCxPOL：信号极性控制寄存器

R/W-x/u	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LCxPOL	—	—	—	LCxG4POL	LCxG3POL	LCxG2POL	LCxG1POL
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 **LCxPOL**：LCOUT 极性控制位
1 = 逻辑单元的输出反相
0 = 逻辑单元的输出同相

bit 6-4 未实现：读为 0

bit 3 **LCxG4POL**：门 4 输出极性控制位
1 = 当应用于逻辑单元时，门 4 的输出反相
0 = 门 4 的输出同相

bit 2 **LCxG3POL**：门 3 输出极性控制位
1 = 当应用于逻辑单元时，门 3 的输出反相
0 = 门 3 的输出同相

bit 1 **LCxG2POL**：门 2 输出极性控制位
1 = 当应用于逻辑单元时，门 2 的输出反相
0 = 门 2 的输出同相

bit 0 **LCxG1POL**：门 1 输出极性控制位
1 = 当应用于逻辑单元时，门 1 的输出反相
0 = 门 1 的输出同相

寄存器 19-3： CLCxSEL0：多路开关数据 1 和 2 选择寄存器

U-0	R/W-x/u	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	LCxD2S<2:0>(1)			—	LCxD1S<2:0>(1)		
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 **未实现：**读为 0

bit 6-4 **LCxD2S<2:0>：**输入数据 2 选择控制位 (1)

111 = 为 lcxd2 选择 CLCxIN[7]
110 = 为 lcxd2 选择 CLCxIN[6]
101 = 为 lcxd2 选择 CLCxIN[5]
100 = 为 lcxd2 选择 CLCxIN[4]
011 = 为 lcxd2 选择 CLCxIN[3]
010 = 为 lcxd2 选择 CLCxIN[2]
001 = 为 lcxd2 选择 CLCxIN[1]
000 = 为 lcxd2 选择 CLCxIN[0]

bit 3 **未实现：**读为 0

bit 2-0 **LCxD1S<2:0>：**输入数据 1 选择控制位 (1)

111 = 为 lcxd1 选择 CLCxIN[7]
110 = 为 lcxd1 选择 CLCxIN[6]
101 = 为 lcxd1 选择 CLCxIN[5]
100 = 为 lcxd1 选择 CLCxIN[4]
011 = 为 lcxd1 选择 CLCxIN[3]
010 = 为 lcxd1 选择 CLCxIN[2]
001 = 为 lcxd1 选择 CLCxIN[1]
000 = 为 lcxd1 选择 CLCxIN[0]

注 1： 关于与输入相关的信号名称，请参见表 19-1。

© 2011-2014 Microchip Technology Inc.

寄存器 19-4： CLCxSEL1：多路开关数据 3 和 4 选择寄存器

U-0	R/W-x/u	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	LCxD4S<2:0>(1)			—	LCxD3S<2:0>(1)		
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 **未实现：**读为 0

bit 6-4 **LCxD4S<2:0>：**输入数据 4 选择控制位 (1)

111 = 为 lcxd4 选择 CLCxIN[7]
110 = 为 lcxd4 选择 CLCxIN[6]
101 = 为 lcxd4 选择 CLCxIN[5]
100 = 为 lcxd4 选择 CLCxIN[4]
011 = 为 lcxd4 选择 CLCxIN[3]
010 = 为 lcxd4 选择 CLCxIN[2]
001 = 为 lcxd4 选择 CLCxIN[1]
000 = 为 lcxd4 选择 CLCxIN[0]

bit 3 **未实现：**读为 0

bit 2-0 **LCxD3S<2:0>：**输入数据 3 选择控制位 (1)

111 = 为 lcxd3 选择 CLCxIN[7]
110 = 为 lcxd3 选择 CLCxIN[6]
101 = 为 lcxd3 选择 CLCxIN[5]
100 = 为 lcxd3 选择 CLCxIN[4]
011 = 为 lcxd3 选择 CLCxIN[3]
010 = 为 lcxd3 选择 CLCxIN[2]
001 = 为 lcxd3 选择 CLCxIN[1]
000 = 为 lcxd3 选择 CLCxIN[0]

注 1： 关于与输入相关的信号名称，请参见表 19-1。

寄存器 19-5： CLCxGLS0：门 1 逻辑选择寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LCxG1D4T	LCxG1D4N	LCxG1D3T	LCxG1D3N	LCxG1D2T	LCxG1D2N	LCxG1D1T	LCxG1D1N
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 **LCxG1D4T：**门 1 数据 4 （同相）位
1 = 将 lcxd4T 通过门输入到 lcxg1
0 = lcxd4T 未通过门输入到 lcxg1

bit 6 **LCxG1D4N：**门 1 数据 4 取反 （反相）位
1 = 将 lcxd4N 通过门输入到 lcxg1
0 = lcxd4N 未通过门输入到 lcxg1

bit 5 **LCxG1D3T：**门 1 数据 3 （同相）位
1 = 将 lcxd3T 通过门输入到 lcxg1
0 = lcxd3T 未通过门输入到 lcxg1

bit 4 **LCxG1D3N：**门 1 数据 3 取反 （反相）位
1 = 将 lcxd3N 通过门输入到 lcxg1
0 = lcxd3N 未过门输入到 lcxg1

bit 3 **LCxG1D2T：**门 1 数据 2 （同相）位
1 = 将 lcxd2T 通过门输入到 lcxg1
0 = lcxd2T 未通过门输入到 lcxg1

bit 2 **LCxG1D2N：**门 1 数据 2 取反 （反相）位
1 = 将 lcxd2N 通过门输入到 lcxg1
0 = lcxd2N 未通过门输入到 lcxg1

bit 1 **LCxG1D1T：**门 1 数据 1 （同相）位
1 = 将 lcxd1T 通过门输入到 lcxg1
0 = lcxd1T 未通过门输入到 lcxg1

bit 0 **LCxG1D1N：**门 1 数据 1 取反 （反相）位
1 = 将 lcxd1N 通过门输入到 lcxg1
0 = lcxd1N 未通过门输入到 lcxg1

© 2011-2014 Microchip Technology Inc.

寄存器 19-6： CLCxGLS1：门 2 逻辑选择寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LCxG2D4T	LCxG2D4N	LCxG2D3T	LCxG2D3N	LCxG2D2T	LCxG2D2N	LCxG2D1T	LCxG2D1N
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 **LCxG2D4T：**门 2 数据 4 （同相）位
1 = 将 lcxd4T 通过门输入到 lcxg2
0 = lcxd4T 未通过门输入到 lcxg2

bit 6 **LCxG2D4N：**门 2 数据 4 取反 （反相）位
1 = 将 lcxd4N 通过门输入到 lcxg2
0 = lcxd4N 未通过门输入到 lcxg2

bit 5 **LCxG2D3T：**门 2 数据 3 （同相）位
1 = 将 lcxd3T 通过门输入到 lcxg2
0 = lcxd3T 未通过门输入到 lcxg2

bit 4 **LCxG2D3N：**门 2 数据 3 取反 （反相）位
1 = 将 lcxd3N 通过门输入到 lcxg2
0 = lcxd3N 未通过门输入到 lcxg2

bit 3 **LCxG2D2T：**门 2 数据 2 （同相）位
1 = 将 lcxd2T 通过门输入到 lcxg2
0 = lcxd2T 未通过门输入到 lcxg2

bit 2 **LCxG2D2N：**门 2 数据 2 取反 （反相）位
1 = 将 lcxd2N 通过门输入到 lcxg2
0 = lcxd2N 未通过门输入到 lcxg2

bit 1 **LCxG2D1T：**门 2 数据 1 （同相）位
1 = 将 lcxd1T 通过门输入到 lcxg2
0 = lcxd1T 未通过门输入到 lcxg2

bit 0 **LCxG2D1N：**门 2 数据 1 取反 （反相）位
1 = 将 lcxd1N 通过门输入到 lcxg2
0 = lcxd1N 未通过门输入到 lcxg2

寄存器 19-7： CLCxGLS2：门 3 逻辑选择寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LCxG3D4T	LCxG3D4N	LCxG3D3T	LCxG3D3N	LCxG3D2T	LCxG3D2N	LCxG3D1T	LCxG3D1N
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 **LCxG3D4T：**门 3 数据 4 （同相）位
1 = 将 lcxd4T 通过门输入到 lcxg3
0 = lcxd4T 未通过门输入到 lcxg3

bit 6 **LCxG3D4N：**门 3 数据 4 取反 （反相）位
1 = 将 lcxd4N 通过门输入到 lcxg3
0 = lcxd4N 未通过门输入到 lcxg3

bit 5 **LCxG3D3T：**门 3 数据 3 （同相）位
1 = 将 lcxd3T 通过门输入到 lcxg3
0 = lcxd3T 未通过门输入到 lcxg3

bit 4 **LCxG3D3N：**门 3 数据 3 取反 （反相）位
1 = 将 lcxd3N 通过门输入到 lcxg3
0 = lcxd3N 未通过门输入到 lcxg3

bit 3 **LCxG3D2T：**门 3 数据 2 （同相）位
1 = 将 lcxd2T 通过门输入到 lcxg3
0 = lcxd2T 未通过门输入到 lcxg3

bit 2 **LCxG3D2N：**门 3 数据 2 取反 （反相）位
1 = 将 lcxd2N 通过门输入到 lcxg3
0 = lcxd2N 未通过门输入到 lcxg3

bit 1 **LCxG3D1T：**门 3 数据 1 （同相）位
1 = 将 lcxd1T 通过门输入到 lcxg3
0 = lcxd1T 未通过门输入到 lcxg3

bit 0 **LCxG3D1N：**门 3 数据 1 取反 （反相）位
1 = 将 lcxd1N 通过门输入到 lcxg3
0 = lcxd1N 未通过门输入到 lcxg3

© 2011-2014 Microchip Technology Inc.

寄存器 19-8： CLCxGLS3：门 4 逻辑选择寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LCxG4D4T	LCxG4D4N	LCxG4D3T	LCxG4D3N	LCxG4D2T	LCxG4D2N	LCxG4D1T	LCxG4D1N
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 **LCxG4D4T：**门 4 数据 4 （同相）位
1 = 将 lcxd4T 通过门输入到 lcxg4
0 = lcxd4T 未通过门输入到 lcxg4

bit 6 **LCxG4D4N：**门 4 数据 4 取反 （反相）位
1 = 将 lcxd4N 通过门输入到 lcxg4
0 = lcxd4N 未通过门输入到 lcxg4

bit 5 **LCxG4D3T：**门 4 数据 3 （同相）位
1 = 将 lcxd3T 通过门输入到 lcxg4
0 = lcxd3T 未通过门输入到 lcxg4

bit 4 **LCxG4D3N：**门 4 数据 3 取反 （反相）位
1 = 将 lcxd3N 通过门输入到 lcxg4
0 = lcxd3N 未通过门输入到 lcxg4

bit 3 **LCxG4D2T：**门 4 数据 2 （同相）位
1 = 将 lcxd2T 通过门输入到 lcxg4
0 = lcxd2T 未通过门输入到 lcxg4

bit 2 **LCxG4D2N：**门 4 数据 2 取反 （反相）位
1 = 将 lcxd2N 通过门输入到 lcxg4
0 = lcxd2N 未通过门输入到 lcxg4

bit 1 **LCxG4D1T：**门 4 数据 1 （同相）位
1 = 将 lcxd1T 通过门输入到 lcxg4
0 = lcxd1T 未通过门输入到 lcxg4

bit 0 **LCxG4D1N：**门 4 数据 1 取反 （反相）位
1 = 将 lcxd1N 通过门输入到 lcxg4
0 = lcxd1N 未通过门输入到 lcxg4

表 19-3： 与 CLCx 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	寄存器所在页
CLC1CON	LC1EN	LC1OE	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>			110
CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	114
CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	115
CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	116
CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	117
CLC1POL	LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	111
CLC1SEL0	—	LC1D2S<2:0>			—	LC1D1S<2:0>			112
CLC1SEL1	—	LC1D4S<2:0>			—	LC1D3S<2:0>			113
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
PIE1	—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—	41
PIR1	—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—	42
TRISA	—	—	—	—	—	TRISA2	TRISA1	TRISA0	69

图注： — = 未实现位，读为 0。 CLC 模块不使用阴影单元。

© 2011-2014 Microchip Technology Inc.

20.0 数控振荡器 （NCO）模块

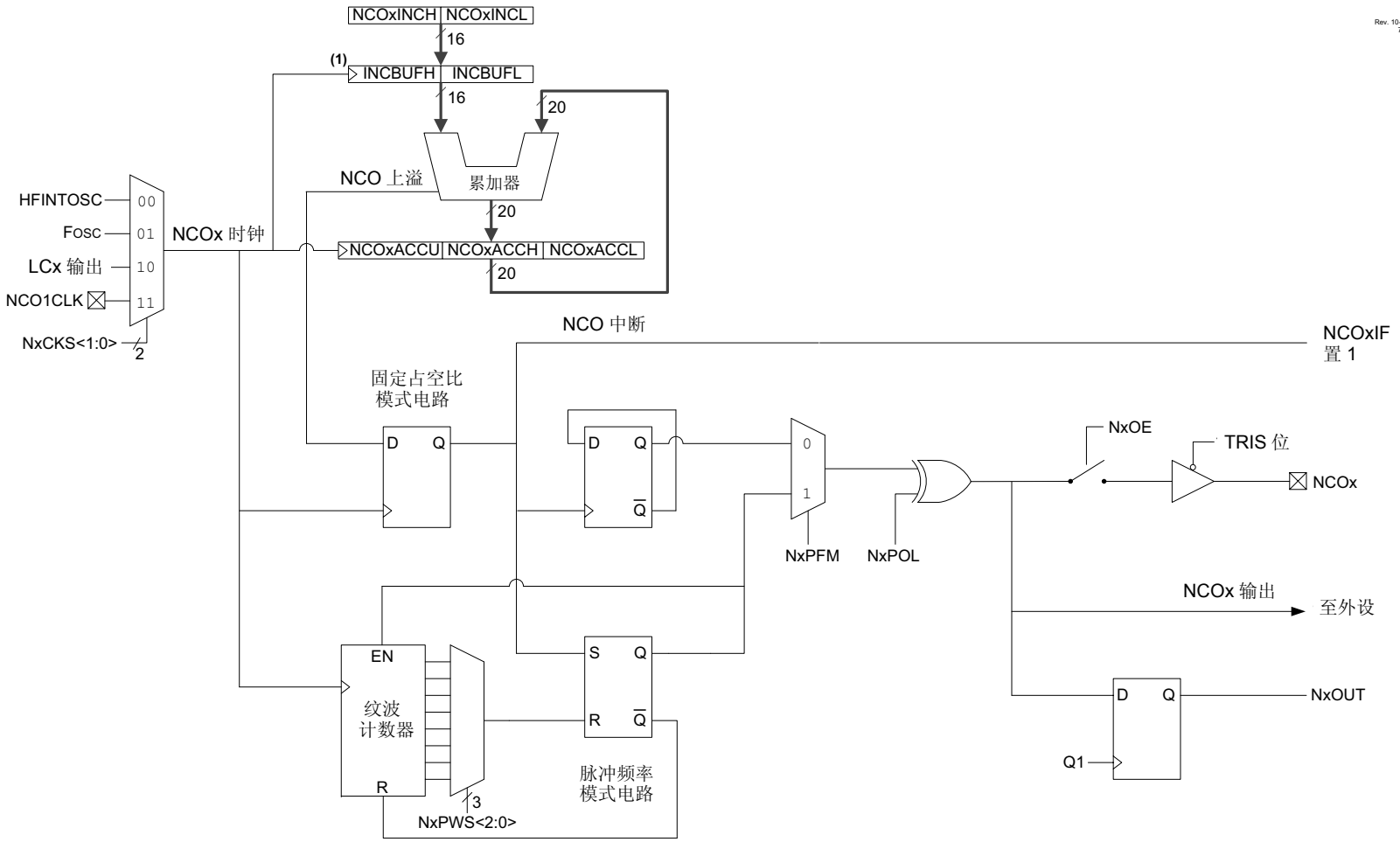
数控振荡器（NCOx）模块是一个定时器，该定时器使用由于增加递增值而引起的上溢来对输入频率进行分频。加法运算优于简单计数器驱动的定时器的优势是：分频分辨率不会随着分频器值的变化而变化。NCOx对于要求在固定占空比条件下确保频率精度和精细分辨率的应用最为有用。

NCOx 的特性包括：

- 16 位递增功能
- 固定占空比 （FDC）模式
- 脉冲频率 （PF）模式
- 输出脉宽控制
- 多种时钟输入源
- 输出极性控制
- 中断功能

图 20-1 给出了 NCOx 模块的简化框图。

图 20-1：　　数控振荡器（NCOx）模块简化框图



注 1： 递增寄存器进行了双重缓冲，以允许在没有先禁止 NCO 模块的情况下更改值。在紧跟写入 NCOxINCL 寄存器之后出现的 NCOx 时钟信号的第二个上升沿，将完整的递增值装入缓冲寄存器。用户不能访问缓冲器，在此仅供参考。

© 2011-2014 Microchip Technology Inc.

20.1 NCOx 工作原理

NCOx 通过重复向累加器增加一个固定的值来工作。达到输入时钟频率时会进行加法运算。累加器会定期进行进位上溢，该进位为原始 NCOx 的输出。这可通过增加值与最大累加器值的比率来有效地降低输入时钟。请参见公式 20-1。

NCOx 输出可以通过延长脉冲或翻转触发器进一步修改。修改后的 NCOx 输出随后会内部分配到其他外设，也可选择地输出到引脚。累加器上溢也会产生中断。

NCOx 输出会产生瞬时频率，从而导致偏差。此输出依靠接收电路（即， CWG 或外部谐振转换器电路）平均瞬时频率的能力来降低偏差。

20.1.1 NCOx 时钟源

可用于 NCOx 的时钟源包括：

- HFINTOSC
- FOSC
- LC1OUT
- NCO1CLK 引脚

通过配置 NCOxCLK 寄存器中的 NxCKS<1:0> 位，可以选择 NCOx 时钟源。

20.1.2 累加器

累加器是个 20 位寄存器。累加器的读写访问通过三个寄存器进行：

- NCOxACCL
- NCOxACCH
- NCOxACCU

20.1.3 加法器

NCOx 加法器是一个全加器，与所选时钟源异步工作。先前结果与递增值的加法运算结果将在每个输入时钟上升沿替换累加器值。

20.1.4 递增寄存器

递增值存储在两个 8 位寄存器中，以构成一个 16 位递增值。寄存器按从低字节到高字节的顺序排列如下：

- NCOxINCL
- NCOxINCH

这两个寄存器都是可读写的。递增寄存器进行了双重缓冲，允许在没有先禁止 NCOx 模块的情况下更改值。

禁止模块后，会立即装入缓冲器。先写入 MS 寄存器是非常必要的，因为在对低位递增寄存器执行写操作后，缓冲器的装入与 NCOx 工作同步。

注：	用户不能访问递增缓冲寄存器。

公式 20-1：

$$F_{OVERFLOW} = \frac{NCO\text{时钟频率} \times \text{递增值}}{2^n}$$

$$n = \text{累加器宽度（单位为位）}$$

20.2 固定占空比 （FDC）模式

在固定占空比 （FDC）模式下，每次累加器上溢时，输出都会翻转。在递增值不变的情况下，这会提供 50% 的占空比。更多信息，请参见图 20-2。

可以通过清零 NCOxCON 寄存器中的 NxPFM 位来选择 FDC 模式。

20.3 脉冲频率 （PF）模式

在脉冲频率 （PF）模式下，每次累加器上溢时，输出都会在一个或多个时钟周期内有效。更多信息，请参见**第 20.3.1 节 “输出脉宽控制”**。时钟周期结束后，输出返回到无效状态。这就产生了一个脉冲控制的输出。

输出在紧跟上溢事件后的时钟上升沿时变为有效。更多信息，请参见图 20-2。

有效状态和无效状态的值取决于 NCOxCON 寄存器中的极性位 NxPOL。

通过将 NCOxCON 寄存器中的 NxPFM 位置 1，可以选择 PF 模式。

20.3.1 输出脉宽控制

在 PF 模式下工作时，输出有效状态的脉宽会随多个时钟周期而变化。使用 NCOxCLK 寄存器中的 NxPWS<2:0> 位来选择各种脉宽。

选定脉宽大于累加器上溢时间帧时，未定义NCOx操作。

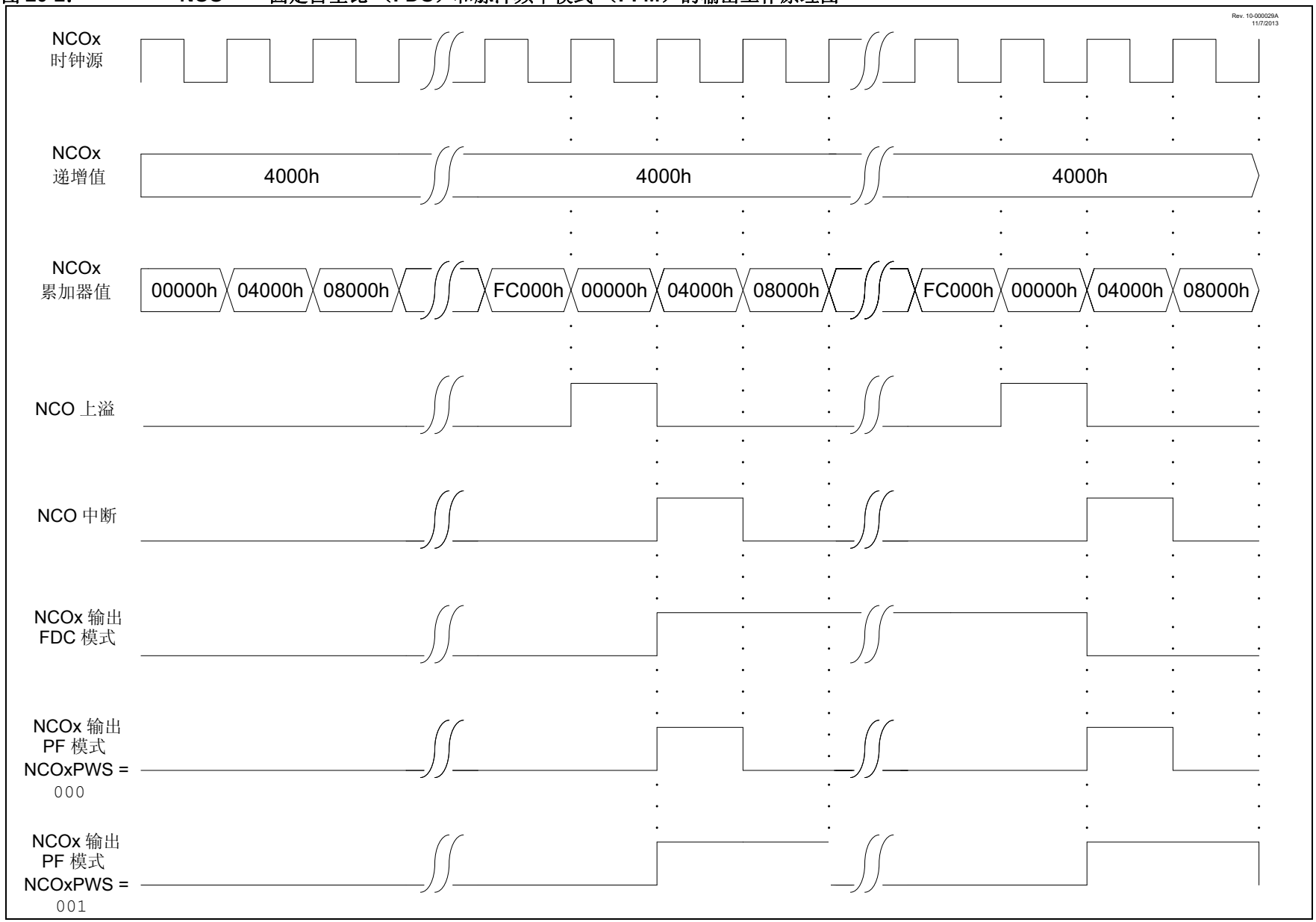
20.4 输出极性控制

NCOx 模块中的最后一个阶段是输出极性。NCOxCON 寄存器中的 NxPOL 位用于选择输出极性。允许中断时更改极性会导致中断结果输出转换。

NCOx 输出可以由源代码或其他外设内部使用。这可以通过读 NCOxCON 寄存器的 NxOUT（只读）位来完成。

© 2011-2014 Microchip Technology Inc.

图 20-2： NCO—— 固定占空比 （FDC）和脉冲频率模式 （PFM）的输出工作原理图



© 2011-2014 Microchip Technology Inc.

20.5 中断

累加器上溢时， PIR1 寄存器的 NCOx 中断标志位 NCOxIF置1。要允许此中断事件，必须将以下各位置1：

- NCOxCON 寄存器的 NxEN 位
- PIE1 寄存器的 NCOxIE 位
- INTCON 寄存器的 PEIE 位
- INTCON 寄存器的 GIE 位

中断必须由软件通过在中断服务程序中清零 NCOxIF 位来清除。

20.6 复位的影响

复位后，所有 NCOx 寄存器都会清零。

20.7 休眠期间的工作原理

NCO模块独立于系统时钟工作，在所选时钟源仍然有效的情况下，此模块在休眠期间将继续运行。

使能 NCO 模块且选择 HFINTOSC 作为时钟源时，HFINTOSC 在休眠期间仍有效，与选择的系统时钟源无关。

也就是说，如果 HFINTOSC 同时被选为系统时钟和 NCO 时钟源，则在使能 NCO 的情况下， CPU 在休眠期间下将处于空闲，但是 NCO 将继续工作，且 HFINTOSC 将保持有效。

这将对休眠模式下的电流产生直接影响。

© 2011-2014 Microchip Technology Inc.

20.8 NCOx 控制寄存器

寄存器 20-1： NCOxCON：NCOx 控制寄存器

R/W-0/0	R/W-0/0	R-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
NxEN	NxOE	NxOUT	NxPOL	—	—	—	NxPFM
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7 **NxEN**：NCOx 使能位
1 = 使能 NCOx 模块
0 = 禁止 NCOx 模块

bit 6 **NxOE**：NCOx 输出使能位
1 = 使能 NCOx 输出引脚
0 = 禁止 NCOx 输出引脚

bit 5 **NxOUT**：NCOx 输出位
1 = NCOx 输出为高电平
0 = NCOx 输出为低电平

bit 4 **NxPOL**：NCOx 极性位
1 = NCOx 输出信号为低电平有效 （反相）
0 = NCOx 输出信号为高电平有效 （同相）

bit 3-1 未实现：读为 0。

bit 0 **NxPFM**：NCOx 脉冲频率模式位
1 = NCOx 工作在脉冲频率模式下
0 = NCOx 工作在固定占空比模式下

寄存器 20-2： NCOxCLK：NCOx 输入时钟控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
NxPWS<2:0>(1, 2)			—	—	—	NxCKS<1:0>	
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-5 **NxPWS<2:0>**：NCOx 输出脉宽选择位 (1, 2)
111 = 128 个 NCOx 时钟周期
110 = 64 个 NCOx 时钟周期
101 = 32 个 NCOx 时钟周期
100 = 16 个 NCOx 时钟周期
011 = 8 个 NCOx 时钟周期
010 = 4 个 NCOx 时钟周期
001 = 2 个 NCOx 时钟周期
000 = 1 个 NCOx 时钟周期

bit 4-2 未实现：读为 0

bit 1-0 **NxCKS<1:0>**：NCOx 时钟源选择位
11 = LC1OUT
10 = HFINTOSC （16 MHz）
01 = FOSC
00 = NCO1CLK 引脚

注 1： 仅在脉冲频率模式下工作时 NxPWS 才适用。
2： 如果 NCOx 脉宽大于 NCOx 上溢周期，则未定义操作。

寄存器 20-3： NCOxACCL：NCOx 累加器寄存器 —— 低字节

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NCOxACC<7:0>							
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0 **NCOxACC<7:0>**：NCOx 累加器低字节

注 **1**：仅在脉冲频率模式下工作时 NxPWS 才适用。
2：如果 NCOx 脉宽大于 NCOx 上溢周期，则未定义操作。

寄存器 20-4： NCOxACCH：NCOx 累加器寄存器 —— 高字节

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NCOxACC<15:8>							
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0 **NCOxACC<15:8>**：NCOx 累加器高字节

寄存器 20-5： NCOxACCU：NCOx 累加器寄存器 —— 最高字节

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	NCOxACC<19:16>			
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-4 未实现：读为 0
bit 3-0 **NCOxACC<19:16>**：NCOx 累加器最高字节

© 2011-2014 Microchip Technology Inc.

寄存器 20-6： NCOxINCL：NCOx 递增寄存器 —— 低字节

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-1/1
NCOxINC<7:0>							
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0 **NCOxINC<7:0>：**NCOx 递增寄存器低字节

寄存器 20-7： NCOxINCH：NCOx 递增寄存器 —— 高字节

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NCOxINC<15:8>							
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0 **NCOxINC<15:8>：**NCOx 递增寄存器高字节

表 20-1： 与 NCOx 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	寄存器所在页
CLC1SEL0	—	LC1D2S2	LC1D2S1	LC1D2S0	—	LC1D1S2	LC1D1S1	LC1D1S0	112
CLC1SEL1	—	LC1D4S2	LC1D4S1	LC1D4S0	—	LC1D3S2	LC1D3S1	LC1D3S0	113
CWG1CON1	G1ASDLB<1:0>		G1ASDLA<1:0>		—	—	G1IS<1:0>		140
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
NCO1ACCH	NCO1ACCH<15:8>								126
NCO1ACCL	NCO1ACCL<7:0>								126
NCO1ACCU	—				NCO1ACCU<19:16				126
NCO1CLK	N1PWS<2:0>			—	—	—	N1CKS<1:0>		125
NCO1CON	N1EN	N1OE	N1OUT	N1POL	—	—	—	N1PFM	125
NCO1INCH	NCO1INCH<15:8>								127
NCO1INCL	NCO1INCL<7:0>								127
PIE1	—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—	41
PIR1	—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—	42
TRISA	—	—	—	—	—	TRISA2	TRISA1	TRISA0	69

图注： x = 未知； u = 不变； — = 未实现，读为 0； q = 值取决于具体条件。 NCO 模块不使用阴影单元。

© 2011-2014 Microchip Technology Inc.

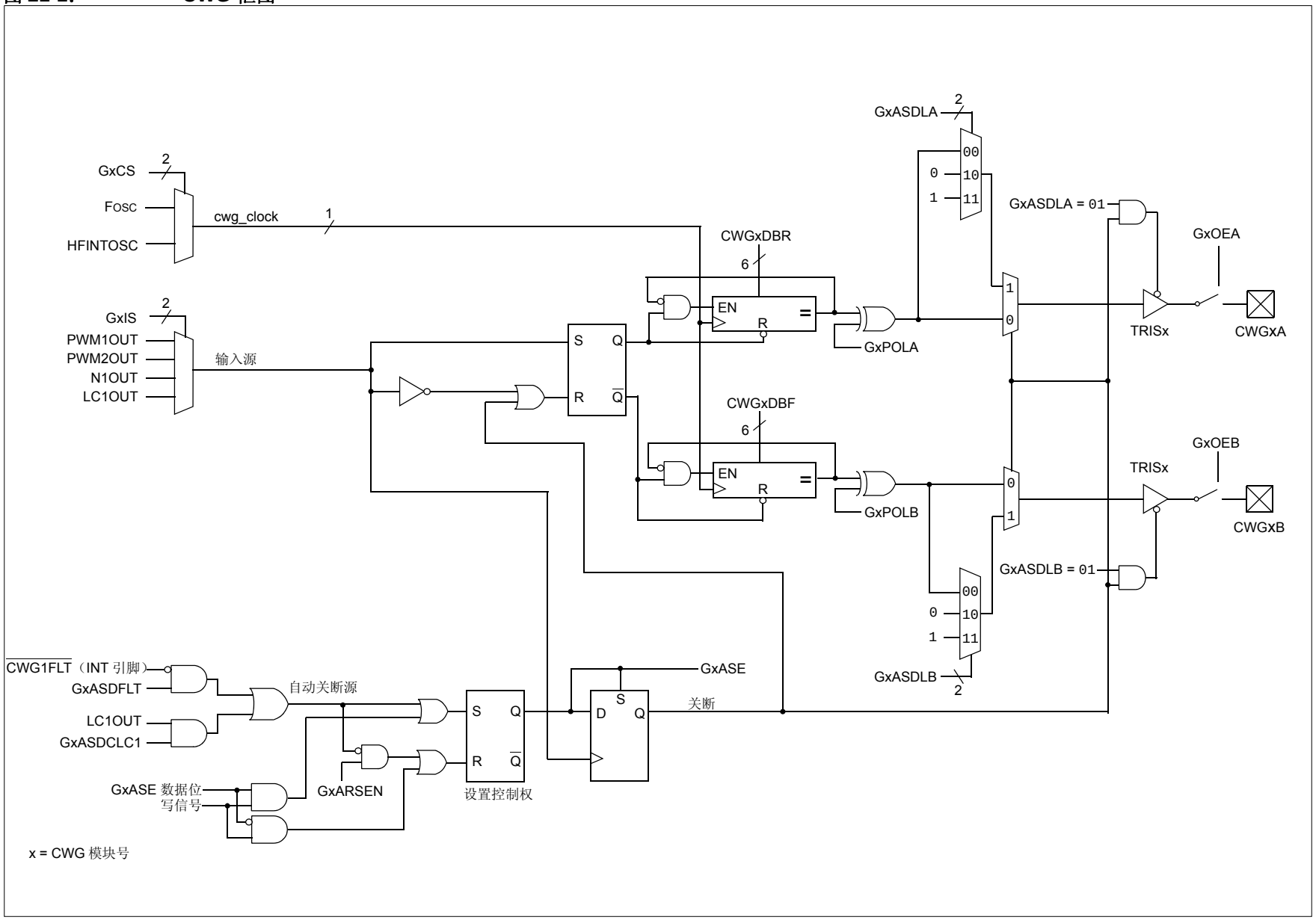
21.0 互补波形发生器 （CWG）模块

互补波形发生器 （CWG）从选择的输入源生成一个带死区延时的互补波形。

CWG 模块具有如下特性：

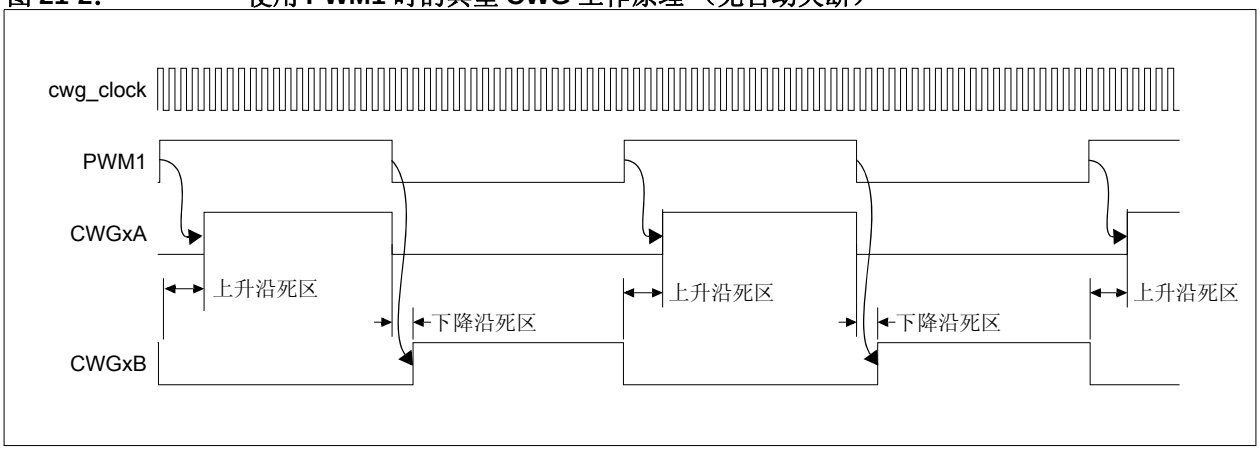
- 可选死区时钟源控制
- 可选输入源
- 输出使能控制
- 输出极性控制
- 使用独立的 6 位上升沿和下降沿死区计数器控制死区
- 使用以下方法控制自动关断：
  - 可选的关断源
  - 使能自动重启
  - 自动关断引脚改写控制

图 21-1： CWG 框图



© 2011-2014 Microchip Technology Inc.

图 **21-2：** 使用 **PWM1** 时的典型 **CWG** 工作原理 （无自动关断）



21.1 基本工作原理

CWG 通过四个可选输入源之一产生一个两路输出互补波形。

每路输出由关到开的转变可能会延时其他输出的由开到关的转变，因而在没有驱动任何输出时会立即产生延时。这被称为死区，在第 21.5 节 “死区控制”中进行介绍。图 21-2 给出了由单个输入信号产生的带有死区的典型工作波形。

可能需要防止可能发生的电路故障或反馈事件到达太晚或根本不到达。这种情况下，在故障条件导致损坏之前必须终止有效的驱动。这被称为自动关断，在第 21.9 节 “自动关断控制”中进行介绍。

21.2 时钟源

CWG 模块可以选择以下时钟源：

- Fosc （系统时钟）
- HFINTOSC （仅 16 MHz）

使用 CWGxCON0 寄存器的 G1CS0 位选择时钟源（寄存器 21-1）。

21.3 可选输入源

CWG 使用以下输入源产生互补波形：

- PWM1
- PWM2
- N1OUT
- LC1OUT

使用 CWGxCON1 寄存器的 GxIS<1:0> 位选择输入源（寄存器 21-2）。

21.4 输出控制

使能 CWG 模块之后，立即配置互补驱动，并清零 CWGxA 和 CWGxB 驱动。

21.4.1 输出使能

每个 CWG 输出引脚都有各自的输出使能控制。使用 CWGxCON0 寄存器的 GxOEA 和 GxOEB 位选择输出使能。当清零输出使能控制时，该模块对引脚无控制权。当输出使能置 1 时，每次选择端口极性，都将对该引脚施加改写值或有效的 PWM 波形。输出引脚使能取决于模块使能位 GxEN。当清零 GxEN 时， CWG 输出使能且 CWG 驱动电平不起作用。

21.4.2 极性控制

每个 CWG 输出的极性可单独选择。当输出极性位置 1 时，相应的输出为高电平有效。清零输出极性位将相应的输出配置为低电平有效。但是，极性不会影响改写电平。使用 CWGxCON0 寄存器的 GxPOLA 和 GxPOLB 位选择输出极性。

21.5 死区控制

死区控制提供无重叠的输出信号以避免在功率开关中出现直通电流。 CWG 包含两个 6 位死区计数器。一个死区计数器将用于输入源控制的上升沿。另一个将用于输入源控制的下降沿。

通过对 CWG 时钟周期从 0 到上升沿或下降沿死区计数器寄存器中的值进行计数，可以实现死区计时。请参见 CWGxDBR 和 CWGxDBF 寄存器（分别为寄存器 21-4 和寄存器 21-5）。

21.6 上升沿死区

上升沿死区将在禁止 CWGxB 输出时延时使能 CWGxA 输出。当输入源信号出现上升沿时，上升沿死区计时开始。此时， CWGxB 输出立即禁止且上升沿死区延时计时开始。当到达上升沿死区延时时，使能 CWGxA 输出。

CWGxDBR 寄存器设置输入源信号上升沿的死区时间间隔的持续时间。持续时间为死区的 0 至 64 次计数。

死区总是在输入源信号的上升沿停止计数。计数为 0 表示无死区。

如果提供输入源信号出现的时间不足以使计数完成，那么相应的输出上将不会产生输出。

© 2011-2014 Microchip Technology Inc.

21.7 下降沿死区

下降沿死区将在禁止 CWGxA 输出时延时使能 CWGxB 输出。当输入源出现下降沿时，下降沿死区计时开始。此时， CWGxA 输出立即禁止且下降沿死区延时计时开始。当到达下降沿死区延时时，使能 CWGxB 输出。

CWGxDBF 寄存器设置输入源信号下降沿的死区时间间隔的持续时间。持续时间为死区的 0 至 64 次计数。

死区总是在输入源信号的下降沿停止计数。计数为 0 表示无死区。

如果提供输入源信号出现的时间不足以使计数完成，那么相应的输出上将不会产生输出。

请参见图 21-3 和图 21-4 的示例。

21.8 死区偏差

当输入源的上升沿和下降沿触发死区计数器时，输入可能是异步的。这将导致死区延时存在一定的偏差。最长偏差时间等于一个 CWG 时钟周期。更多详细信息，请参见公式 21-1。

图 21-3： 死区工作原理 （**CWGxDBR = 01H， CWGxDBF = 02H**）

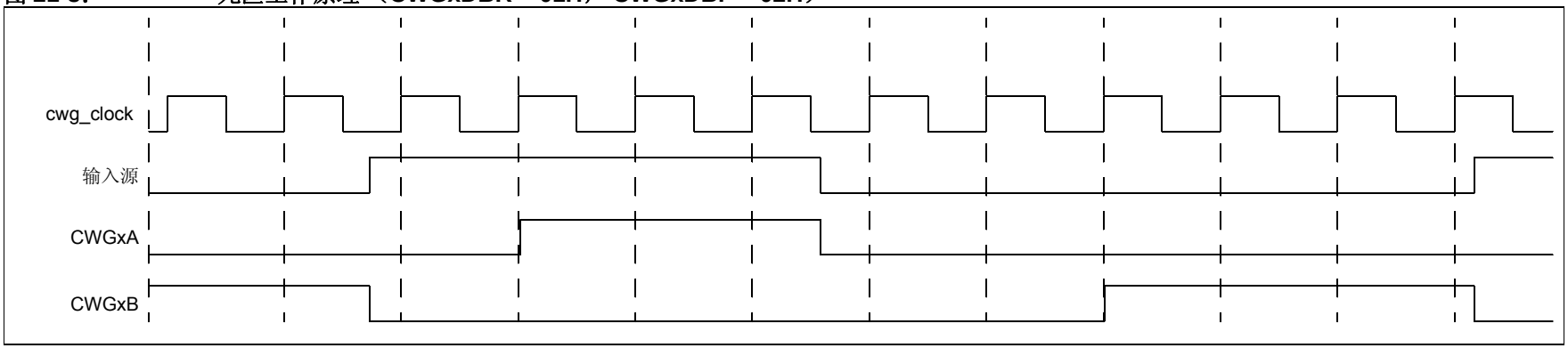
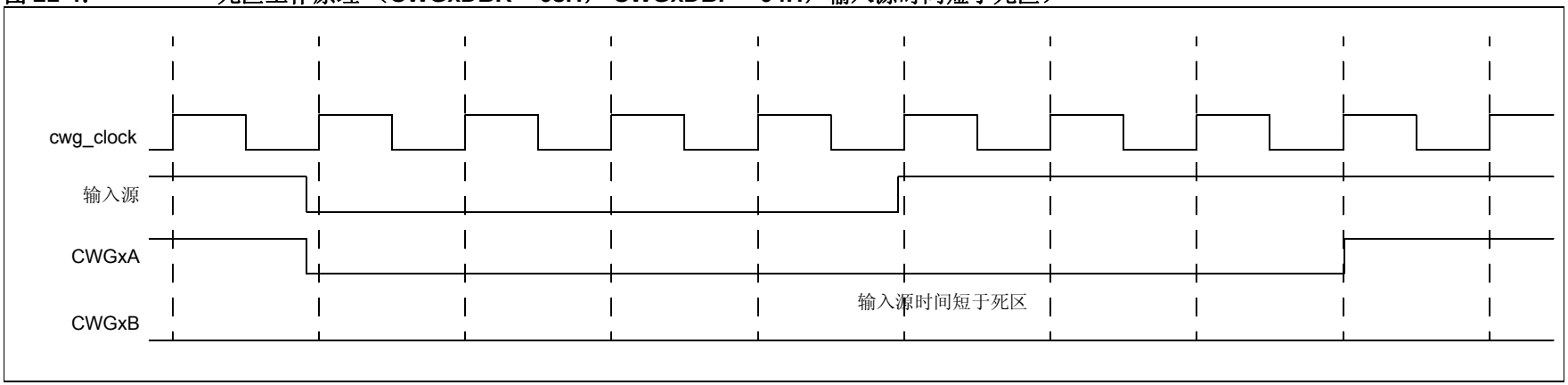


图 21-4： 死区工作原理 （**CWGxDBR = 03H， CWGxDBF = 04H，输入源时间短于死区**）



© 2011-2014 Microchip Technology Inc.

公式 **21-1：** 死区延时偏差

$$T_{DEADBAND_UNCERTAINTY} = \frac{1}{Fcwg_clock}$$

例 **21-1：** 死区延时偏差

$$Fcwg_clock = 16\ MHz$$

因此：

$$T_{DEADBAND_UNCERTAINTY} = \frac{1}{Fcwg_clock} = \frac{1}{16\ MHz} = 625\ ns$$

21.9 自动关断控制

自动关断是一种使用特定改写信号立即改写 CWG 输出电平，从而安全关断电路的方法。关断状态可以自动清除或者保持到由软件清除为止。

21.9.1 关断

关断状态可以由以下两种方法之一实现：

- 由软件产生
- 外部输入

21.9.1.1 由软件产生的关断

通过将 CWGxCON2 寄存器的 GxASE 位置 1，可以强制 CWG 进入关断状态。

当禁止自动重启时，只要 GxASE 位置 1，关断状态就一直保持。

当使能自动重启时， GxASE 位将自动清零并在下一个上升沿事件时重新恢复工作。请参见图 21-6。

21.9.1.2 外部输入源

当发生故障条件时，外部关断输入可以最快地安全中止 CWG 工作。当选中的关断输入中的任何一个变为高电平时， CWG 输出将立即变为选择的改写电平，而没有软件延时。选择两个输入源的任何组合都将产生关断条件。这两个输入源为：

- LC1OUT
- $\overline{\text{CWG1FLT}}$

通过使用CWGxCON2寄存器的GxASDS0和GxASDS1位，可以选择关断输入 （见寄存器 21-3）。

注：	关断输入对电平敏感，而对边沿不敏感。只要关断输入电平不变，就不能清除关断状态，除非禁止自动关断。

21.10 休眠期间的工作原理

CWG 模块独立于系统时钟工作，如果选择的时钟和输入源仍有效，那么此模块在休眠期间将继续运行。

如果使能 CWG 模块，输入源有效并且选择 HFINTOSC 作为时钟源，那么 HFINTOSC 在休眠期间仍有效，与选择的系统时钟源无关。

换句话说，如果 HFINTOSC 同时被选为系统时钟和 CWG 时钟源，则在使能 CWG 且输入源有效时， CPU 在休眠期间将处于空闲，但是 CWG 将继续运行，且 HFINTOSC 将继续有效。

这将直接影响休眠模式下的电流。

© 2011-2014 Microchip Technology Inc.

21.11 配置 CWG

以下步骤阐述了如何正确配置CWG才能确保同步启动：

1. 确保与 CWGxA 和 CWGxB 对应的 TRIS 控制位置 1，以便将它们配置为输入。
2. 清零 GxEN 位 （如果尚未清零）。
3. 使用 CWGxDBR 和 CWGxDBF 寄存器设置所需的死区时间。
4. 在CWGxCON2自动关断寄存器中设置以下控制：
   - 选择所需的关断源。
   - 将两个输出改写电平选为所需的电平 （即使不使用自动关断时这也是必需的，因为启动将从关断状态开始）。
   - 将 GxASE 位置 1，将 GxARSEN 位清零。
5. 使用 CWGxCON1 寄存器选择所需的输入源。
6. 在 CWGxCON0 寄存器中配置如下控制：
   - 选择所需的时钟源。
   - 选择所需的输出极性。
   - 对要使用的输出设置输出使能。
7. 将 GxEN 位置 1。
8. 将与要使用的 CWGxA 和 CWGxB 对应的 TRIS 控制位清零，以便将这 2 个引脚配置为输出。
9. 如果使用自动重启，将 GxARSEN 位置 1，GxASE 位将自动清零。否则，清零 GxASE 位启动 CWG。

21.11.1 引脚改写电平

当关断输入为真时，驱动输出引脚的电平由 CWGxCON1 寄存器 （寄存器 21-2）的 GxASDLA 和 GxASDLB 位控制。GxASDLA 控制 CWG1A 改写电平，而 GxASDLB 控制 CWG1B 改写电平。控制位逻辑电平与关断状态下的输出逻辑驱动电平相对应。极性控制不适用于改写电平。

21.11.2 自动关断重启

发生自动关断事件后，有两种办法恢复工作：

- 由软件控制
- 自动重启

使用 CWGxCON2 寄存器的 GxARSEN 位选择重启方法。由软件控制和自动重启的波形如图 21-5 和图 21-6 所示。

21.11.2.1 由软件控制重启

当清零 CWGxCON2 寄存器的 GxARSEN 位时，必须在自动关断事件之后由软件重启 CWG。

CWG将在清零GxASE位之后的第一个上升沿事件时恢复工作。清除关断状态需要所有选择的关断输入为低电平，否则 GxASE 位仍保持置 1。

21.11.2.2 自动重启

当 CWGxCON2 寄存器的 GxARSEN 位置 1 时，CWG 将从自动关断状态自动重启。

清除关断事件之后，GxASE 位将自动清零，CWG 将在第一个上升沿事件时恢复工作。

图 21-5： 禁止自动重启时的关断功能 （GxARSEN = 0）

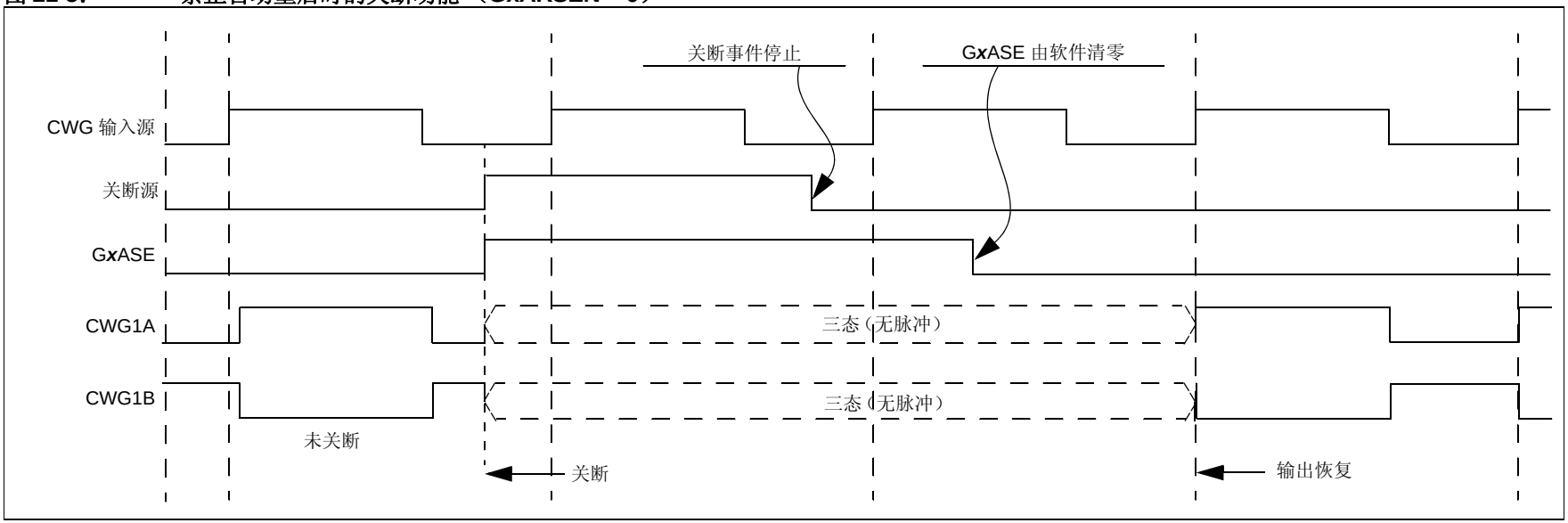
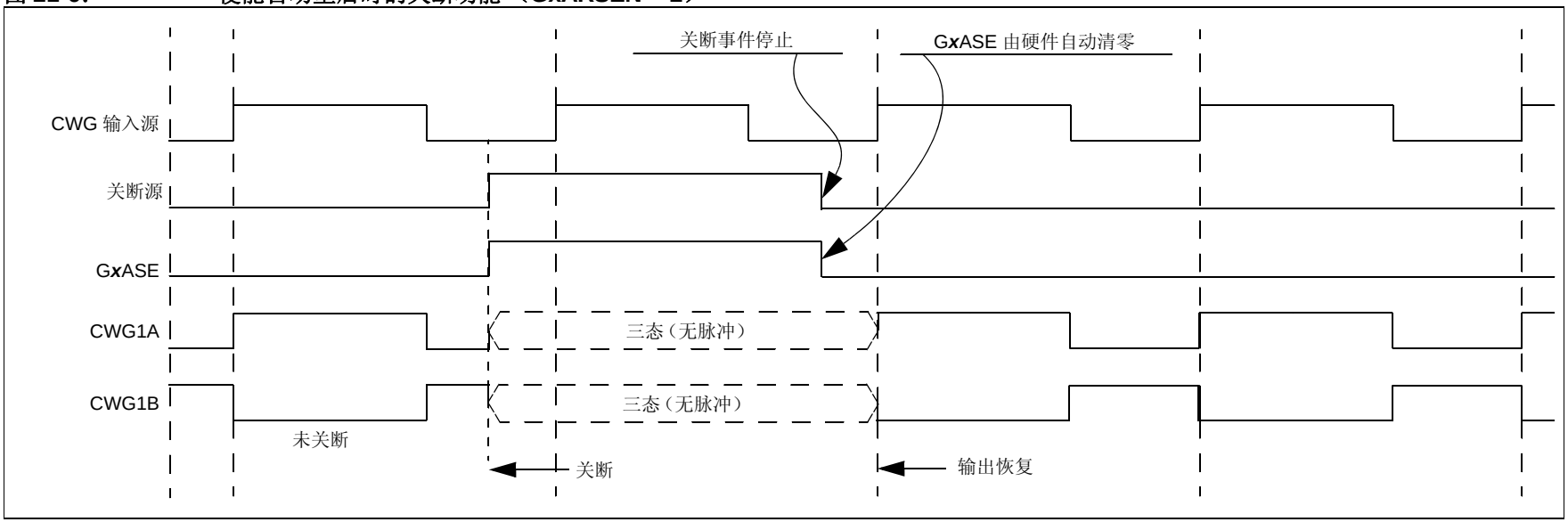


图 21-6： 使能自动重启时的关断功能 （GxARSEN = 1）



© 2011-2014 Microchip Technology Inc.

21.12 CWG 控制寄存器

寄存器 21-1： CWGxCON0：CWG 控制寄存器 0

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0
GxEN	GxOEB	GxOEA	GxPOLB	GxPOLA	—	—	GxCS0
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	q = 值取决于具体条件

bit 7 **GxEN**：CWGx 使能位
1 = 使能模块
0 = 禁止模块

bit 6 **GxOEB**：CWGxB 输出使能位
1 = CWGxB 在相应的 I/O 引脚上可用
0 = CWGxB 在相应的 I/O 引脚上不可用

bit 5 **GxOEA**：CWGxA 输出使能位
1 = CWGxA 在相应的 I/O 引脚上可用
0 = CWGxA 在相应的 I/O 引脚上不可用

bit 4 **GxPOLB**：CWGxB 输出极性位
1 = 输出极性翻转
0 = 输出极性不翻转

bit 3 **GxPOLA**：CWGxA 输出极性位
1 = 输出极性翻转
0 = 输出极性不翻转

bit 2-1 未实现：读为 0

bit 0 **GxCS0**：CWGx 时钟源选择位
1 = HFINTOSC
0 = FOSC

寄存器 21-2： CWGxCON1：CWG 控制寄存器 1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	R/W-0/0	R/W-0/0
GxASDLB<1:0>		GxASDLA<1:0>		—	—	GxIS<1:0>	
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	q = 值取决于具体条件

bit 7-6 **GxASDLB<1:0>**：CWGxB 的 CWGx 关断状态
发生自动关断事件 （GxASE = 1）时：
11 = CWGxB 引脚驱动为 1，与 GxPOLB 位的设置无关。
10 = CWGxB 引脚驱动为 0，与 GxPOLB 位的设置无关。
01 = CWGxB 引脚为三态
00 = CWGxB 在选择的死区时间间隔之后驱动为无效状态。 GxPOLB 仍将控制输出的极性。

bit 5-4 **GxASDLA<1:0>**：CWGxA 的 CWGx 关断状态
发生自动关断事件 （GxASE = 1）时：
00 = CWGxA 在选择的死区时间间隔之后驱动为无效状态。 GxPOLA 仍将控制输出的极性。
01 = CWGxA 引脚为三态
10 = CWGxA 引脚驱动为 0，与 GxPOLA 位的设置无关。
11 = CWGxA 引脚驱动为 1，与 GxPOLA 位的设置无关。

bit 3-2 未实现：读为 0

bit 1-0 **GxIS<1:0>**：CWGx 死区源选择位
11 = LC1OUT
10 = N1OUT
01 = PWM2OUT
00 = PWM1OUT

© 2011-2014 Microchip Technology Inc.

寄存器 21-3： CWGxCON2：CWG 控制寄存器 2

R/W/HC/HS-0/0	R/W-0/0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
GxASE	GxARSEN	—	—	—	—	GxASDCLC1	GxASDFLT
bit 7							bit 0

图注：		
HC = 由硬件清零		HS = 由硬件置 1
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	q = 值取决于具体条件

bit 7 **GxASE：**自动关断事件状态位
1 = 发生了自动关断事件。改写 GxOEB/GxOEA 输出控制，禁止输出。
0 = 未发生自动关断事件，或者发生了自动重启。使能 GxOEB/GxOEA 输出控制。

bit 6 **GxARSEN：**自动重启使能位
1 = 使能自动重启
0 = 禁止自动重启

bit 5-2 **未实现：**读为 0

bit 1 **GxASDCLC1：**CWG 自动关断源使能位 1
1 = 当 LC1OUT 为高电平时关断
0 = LC1OUT 对关断没有影响

bit 0 **GxASDFLT：**CWG 自动关断源使能位 0
1 = 当 $\overline{\text{CWG1FLT}}$ 输入为低电平时关断
0 = $\overline{\text{CWG1FLT}}$ 输入对关断没有影响

寄存器 21-4： CWGxDBR：互补波形发生器 （CWGx）上升沿的死区计数寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	CWG**x**DBR<5:0>					
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	q = 值取决于具体条件

bit 7-6 未实现：读为 0

bit 5-0 **CWGxDBR<5:0>**：互补波形发生器 （CWGx）上升沿计数位

11 1111 = 死区的 63-64 次计数
11 1110 = 死区的 62-63 次计数

•
•
•

00 0010 = 死区的 2-3 次计数
00 0001 = 死区的 1-2 次计数
00 0000 = 死区的 0 次计数

寄存器 21-5： CWGxDBF：互补波形发生器 （CWGx）下降沿的死区计数寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	CWGxDBF<5:0>					
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	q = 值取决于具体条件

bit 7-6 未实现：读为 0

bit 5-0 **CWGxDBF<5:0>**：互补波形发生器 （CWGx）下降沿计数位

11 1111 = 死区的 63-64 次计数
11 1110 = 死区的 62-63 次计数

•
•
•

00 0010 = 死区的 2-3 次计数
00 0001 = 死区的 1-2 次计数
00 0000 = 死区的 0 次计数。死区被旁路。

© 2011-2014 Microchip Technology Inc.

表 21-1： 与 CWG 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	寄存器所在页
ANSELA	—	—	—	—	—	ANSA2	ANSA1	ANSA0	70
CWG1CON0	G1EN	G1OEB	G1OEA	G1POLB	G1POLA	—	—	G1CS0	139
CWG1CON1	G1ASDLB<1:0>		G1ASDLA<1:0>		—	—	G1IS<1:0>		140
CWG1CON2	G1ASE	G1ARSEN	—	—	—	—	G1ASDCLC1	G1ASDFLT	141
CWG1DBF	—	—	CWG1DBF<5:0>						142
CWG1DBR	—	—	CWG1DBR<5:0>						142
LATA	—	—	—	—	—	LATA2	LATA1	LATA0	70
TRISA	—	—	—	—	—	TRISA2	TRISA1	TRISA0	69

图注： x = 未知， u = 不变， – = 未实现单元，读为 0。 CWG 不使用阴影单元。

22.0 在线串行编程（ICSP™）

ICSP™ 编程允许客户在生产电路板时使用未编程器件。编程可以在装配流程之后完成，从而可以使用最新版本的固件或者定制固件对器件编程。ICSP™ 编程需要 5 个引脚：

- ICSPCLK
- ICSPDAT
- $\overline{\text{MCLR}}$/VPP
- VDD
- VSS

在编程 / 校验模式下，通过串行通信对程序存储器、用户 ID 和配置字进行编程。ICSPDAT 引脚是用于传输串行数据的双向 I/O，ICSPCLK 引脚是时钟输入引脚。关于 ICSP™ 的更多信息，请参见 *“PIC10(L)F320/322 Flash Memory Programming Specification”*（DS41572）。

22.1 高电压编程进入模式

通过将 ICSPCLK 和 ICSPDAT 引脚保持为低电平然后将 $\overline{\text{MCLR}}$/VPP 引脚上的电压升至 VIHH，可将器件置于高电压编程进入模式。

22.2 低电压编程进入模式

通过使用低电压编程进入模式，只需使用 VDD 即可对 PIC®闪存MCU进行编程，而无需使用高电压。当配置字的 LVP 位设置为 1 时，将使能低电压 ICSP 编程进入模式。要禁止低电压 ICSP 模式，必须将 LVP 位编程为 0。

要使用低电压编程进入模式，需要执行以下步骤：

1. 将 $\overline{\text{MCLR}}$ 拉至 VIL。
2. 在 ICSPDAT 引脚上输出 32 位密钥序列，而在 ICSPCLK 引脚上输出时钟。

在密钥序列完成后，只要保持在编程 / 校验模式下，$\overline{\text{MCLR}}$ 就必须保持在 VIL 电平。

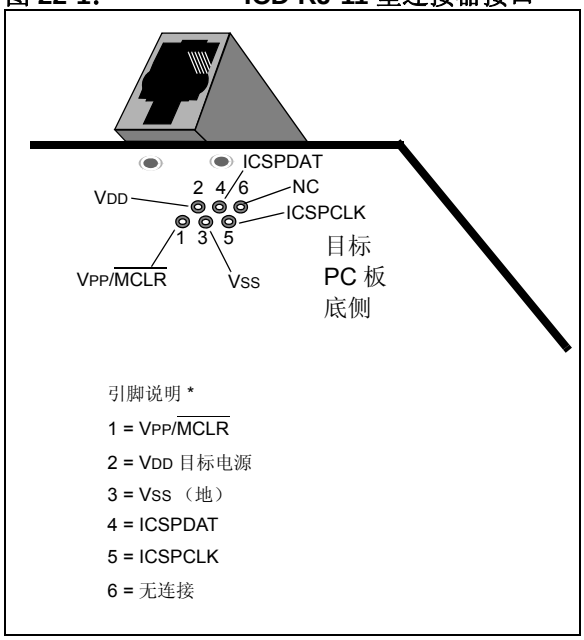
如果使能了低电压编程（LVP = 1），将自动使能 $\overline{\text{MCLR}}$ 复位功能，且无法禁止。更多信息，请参见**第 5.4 节 “低功耗欠压复位（LPBOR）”**。

只能通过使用高电压编程模式将 LVP 位重新编程为 0。

22.3 通用编程接口

通常，与目标器件的连接通过一个ICSP™连接头完成。开发工具上常见的连接器是采用 6P6C（6 个引脚，6 个连接器）配置的 RJ-11。请参见图 22-1。

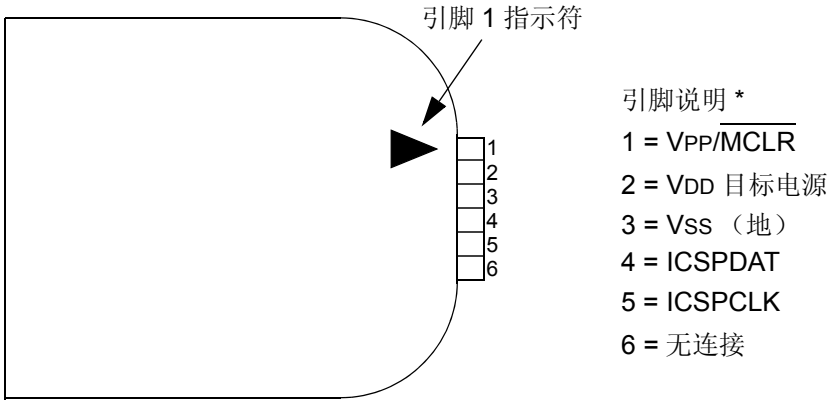
图 22-1： ICD RJ-11 型连接器接口



另一种常用于 PICkit™ 编程器的连接器是间距为 0.1 英寸的标准 6 引脚插头。请参见图 22-2。

© 2011-2014 Microchip Technology Inc.

图 22-2： PICkit™ 型连接器接口

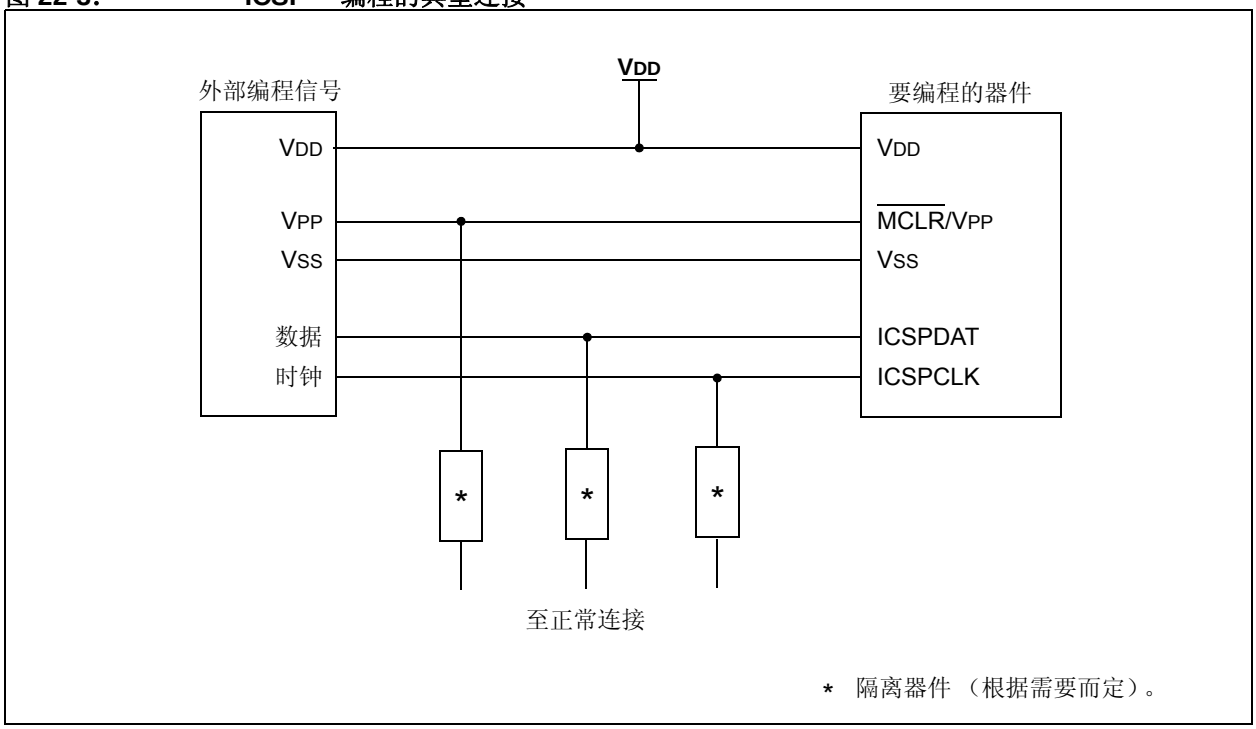


* 6 引脚连接头 （0.100" 间距）可连接 0.025" 的方形引脚。

关于其他推荐接口，在进行 PCB 设计之前，请参见具体的器件编程器手册。

建议使用隔离器件将编程引脚与其他电路隔离。隔离类型主要取决于具体应用，可能包括电阻、二极管甚至跳线之类的器件。更多信息，请参见图 22-3。

图 22-3： ICSP™ 编程的典型连接



© 2011-2014 Microchip Technology Inc.

23.0 指令集汇总

PIC10(L)F320/322 指令集具有高度正交性，分为以下 3 个基本类型：

- **面向字节**操作类指令
- **面向位**操作类指令
- **立即数和控制**操作类指令

每条 PIC16 指令都是一个 14 位字，由**操作码**（指定指令类型）和一个或多个**操作数**（指定指令操作）组成。图 23-1 给出了每个指令类别的格式，而表 23-1 汇总了各种操作码字段。

表 23-2 列出了可被 MPASM™ 汇编器识别的指令。

对于**面向字节**的指令，f 表示文件寄存器指示符，d 表示目标寄存器指示符。文件寄存器指示符指定指令所使用的文件寄存器。

目标寄存器指示符指定操作结果存放的位置。如果 d 为 0，结果存放到 W 寄存器。如果 d 为 1，结果存放到指令中指定的文件寄存器。

对于**面向位**的指令， b 表示位字段指示符，它将选择操作所影响的位，而 f 表示位所在的文件寄存器地址。

对于**立即数和控制**操作类指令，k 表示 8 位或 11 位常数或者立即数值。

一个指令周期由 4 个振荡周期组成；如果振荡器频率为 4 MHz，则可提供 1 μs 的标称指令执行时间。所有指令均可在单指令周期内执行完毕，条件测试为真或程序计数器更改为指令的结果时除外。此时，指令执行将需要两个指令周期，第二个周期执行一条 NOP 指令。

所有指令示例均使用 0xhh 格式表示一个十六进制数，其中 h 表示一位十六进制数字。

23.1 读 - 修改 - 写操作

所有需要使用文件寄存器的指令都会执行读 - 修改 - 写（RMW）操作。根据指令或目标寄存器指示符 d 读寄存器、修改数据和保存结果。即使存在写寄存器的指令 ，也将先对该寄存器进行读操作。

例如， CLRF PORTA 指令将读 PORTA，清零所有数据位，然后将结果写回 PORTA。该示例可能造成意外地清除将 IOCIF 标志置 1 的条件。

表 23-1： 操作码字段说明

字段	说明
f	文件寄存器地址 （0x00 至 0x7F）
W	工作寄存器 （累加器）
b	8 位文件寄存器内的位地址
k	立即数字段、常数或标号
x	忽略 （= 0 或 1）。 汇编器将生成 x = 0 的代码。为了与所有的 Microchip 软件工具兼容，建议使用这种格式。
d	目标寄存器选择； d = 0：结果保存至 W ； d = 1：结果保存至文件寄存器 f。 默认值为 d = 1。
PC	程序计数器
$\overline{TO}$	超时位
C	进位位
DC	半进位位
Z	全零标志位
$\overline{PD}$	掉电位

图 23-1： 指令的通用格式

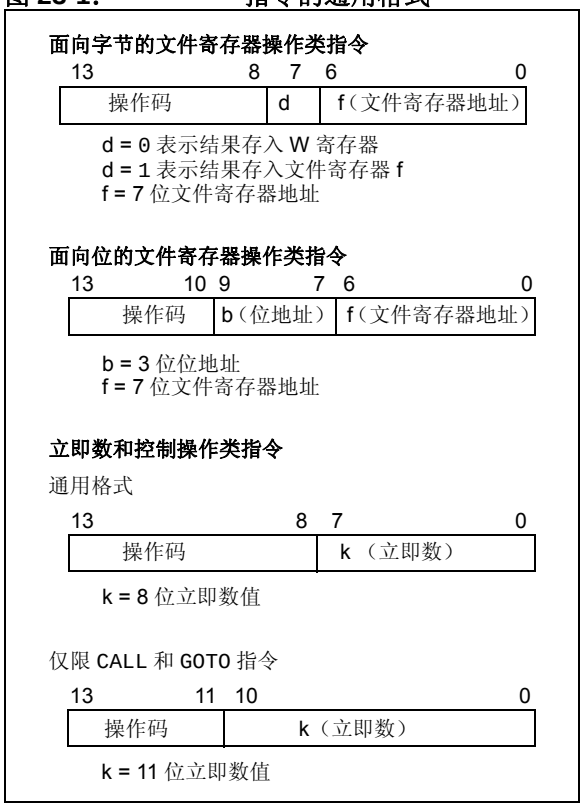


表 23-2： 指令集

助记符，操作数		说明	周期数	14 位操作码 MSb			LSb	受影响的状态位	注
面向字节的文件寄存器操作									
ADDWF	f, d	W 和 f 相加	1	00	0111	dfff	ffff	C、DC 和 Z	**1, 2**
ANDWF	f, d	W 和 f 作逻辑与运算	1	00	0101	dfff	ffff	Z	**1, 2**
CLRF	f	将 f 清零	1	00	0001	lfff	ffff	Z	**2**
CLRW	–	将 W 寄存器清零	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	f 取反	1	00	1001	dfff	ffff	Z	**1, 2**
DECF	f, d	f 递减 1	1	00	0011	dfff	ffff	Z	**1, 2**
DECFSZ	f, d	f 递减 1，为 0 则跳过	1 **(2)**	00	1011	dfff	ffff		**1, 2, 3**
INCF	f, d	f 递增 1	1	00	1010	dfff	ffff	Z	**1, 2**
INCFSZ	f, d	f 递增 1，为 0 则跳过	1 **(2)**	00	1111	dfff	ffff		**1, 2, 3**
IORWF	f, d	W 与 f 作逻辑或运算	1	00	0100	dfff	ffff	Z	**1, 2**
MOVF	f, d	传送 f	1	00	1000	dfff	ffff	Z	**1, 2**
MOVWF	f	将 W 的内容送到 f	1	00	0000	lfff	ffff		
NOP	–	空操作	1	00	0000	0xx0	0000		
RLF	f, d	对 f 执行带进位的循环左移	1	00	1101	dfff	ffff	C	**1, 2**
RRF	f, d	对 f 执行带进位的循环右移	1	00	1100	dfff	ffff	C	**1, 2**
SUBWF	f, d	f 减去 W	1	00	0010	dfff	ffff	C、DC 和 Z	**1, 2**
SWAPF	f, d	将 f 中的两个半字节进行交换	1	00	1110	dfff	ffff		**1, 2**
XORWF	f, d	W 与 f 作逻辑异或运算	1	00	0110	dfff	ffff	Z	**1, 2**
面向位的文件寄存器操作									
BCF	f, b	将 f 中的某位清零	1	01	00bb	bfff	ffff		**1, 2**
BSF	f, b	将 f 中的某位置 1	1	01	01bb	bfff	ffff		**1, 2**
BTFSC	f, b	检测 f 中的某位，为 0 则跳过	1 **(2)**	01	10bb	bfff	ffff		**3**
BTFSS	f, b	检测 f 中的某位，为 1 则跳过	1 **(2)**	01	11bb	bfff	ffff		**3**
立即数和控制操作									
ADDLW	k	立即数和 W 相加	1	11	111x	kkkk	kkkk	C、DC 和 Z	
ANDLW	k	立即数与 W 作逻辑与运算	1	11	1001	kkkk	kkkk	Z	
CALL	k	调用子程序	2	10	0kkk	kkkk	kkkk		
CLRWDT	–	清零看门狗定时器	1	00	0000	0110	0100	$\overline{TO}$ 和 $\overline{PD}$	
GOTO	k	跳转到地址	2	10	1kkk	kkkk	kkkk		
IORLW	k	立即数与 W 作逻辑或运算	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	将立即数传送到 W	1	11	00xx	kkkk	kkkk		
RETFIE	–	从中断返回	2	00	0000	0000	1001		
RETLW	k	返回并将立即数传送到 W	2	11	01xx	kkkk	kkkk		
RETURN	–	从子程序返回	2	00	0000	0000	1000		
SLEEP	–	进入待机模式	1	00	0000	0110	0011	$\overline{TO}$ 和 $\overline{PD}$	
SUBLW	k	从立即数中减去 W 的内容	1	11	110x	kkkk	kkkk	C、DC 和 Z	
XORLW	k	立即数与 W 作逻辑异或运算	1	11	1010	kkkk	kkkk	Z	

注 **1：** 当 I/O 寄存器作为其自身的一个函数（例如，MOVF PORTA, 1）被修改时，所使用的值将为该引脚上的当前值。例如，当配置为输入的引脚上的数据锁存器为 1 并被外部器件驱动为低电平时，数据将写回 0。

2： 如果该指令在 TMR0 寄存器（适用时，d = 1）上执行，分配给 Timer0 模块的预分频器将清零。

3： 如果程序计数器（PC）被修改或条件测试为真，则执行该指令需要两个周期。第二个周期执行一条 NOP 指令。

© 2011-2014 Microchip Technology Inc.

23.2 指令说明

ADDLW	立即数和 W 相加
语法：	[*标号*] ADDLW k
操作数：	0 ≤ k ≤ 255
操作：	(W) + k → (W)
受影响的状态位：	C、 DC 和 Z
说明：	将 W 寄存器的内容与 8 位立即数 k 相加，结果存放到 W 寄存器。

ADDWF	W 和 f 相加
语法：	[*标号*] ADDWF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(W) + (f) → (目标寄存器)
受影响的状态位：	C、 DC 和 Z
说明：	将 W 寄存器的内容与 f 寄存器的内容相加。如果 d 为 0，结果存放到 W 寄存器。如果 d 为 1，结果存回寄存器 f。

ANDLW	立即数与 W 作逻辑与运算
语法：	[*标号*] ANDLW k
操作数：	0 ≤ k ≤ 255
操作：	(W) .AND. (k) → (W)
受影响的状态位：	Z
说明：	将 W 寄存器的内容与 8 位立即数 k 作逻辑与运算。结果存放到 W 寄存器。

ANDWF	W 和 f 作逻辑与运算
语法：	[*标号*] ANDWF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(W) .AND. (f) → (目标寄存器)
受影响的状态位：	Z
说明：	W寄存器与f寄存器作逻辑与运算。如果 d 为 0，结果存放到 W 寄存器。如果d为1，结果存回寄存器f。

BCF	将 f 寄存器中的某位清零
语法：	[*标号*] BCF f,b
操作数：	0 ≤ f ≤ 127 0 ≤ b ≤ 7
操作：	0 → (f<b>)
受影响的状态位：	无
说明：	将寄存器 f 中的位 b 清零。

BSF	将 f 中的某位置 1
语法：	[*标号*] BSF f,b
操作数：	0 ≤ f ≤ 127 0 ≤ b ≤ 7
操作：	1 → (f<b>)
受影响的状态位：	无
说明：	将寄存器 f 中的位 b 置 1。

BTFSC	检测 f 中的某位，为 0 则跳过
语法：	[*标号*] BTFSC f,b
操作数：	0 ≤ f ≤ 127 0 ≤ b ≤ 7
操作：	如果 (f<b>) = 0 则跳过
受影响的状态位：	无
说明：	如果 f 寄存器中的位 b 为 1，则执行下一条指令。 如果 f 寄存器中的位 b 为 0，则丢弃下一条指令，转而执行一条 NOP 指令，从而使该指令成为双周期指令。

BTFSS	**检测 f 中的某位，为 1 则跳过**
语法：	[*标号*] BTFSS f,b
操作数：	0 ≤ f ≤ 127 0 ≤ b < 7
操作：	如果 (f<b>) = 1 则跳过
受影响的状态位：	无
说明：	如果 f 寄存器中的位 b 为 0，则执行下一条指令。 如果位 b 为 1，那么下一条指令将丢弃，转而执行一条 NOP 指令，从而使该指令成为双周期指令。

CALL	**调用子程序**
语法：	[*标号*] CALL k
操作数：	0 ≤ k ≤ 2047
操作：	(PC)+ 1→ TOS， k → PC<10:0>， (PCLATH<4:3>) → PC<12:11>
受影响的状态位：	无
说明：	调用子程序。首先，将返回地址（PC+ 1）压入堆栈。将 11 位立即数地址装入 PC 位 <10:0>。将 PCLATH 的内容装入 PC 的高位。CALL 是双周期指令。

CLRF	**将 f 清零**
语法：	[*标号*] CLRF f
操作数：	0 ≤ f ≤ 127
操作：	00h → (f) 1 → Z
受影响的状态位：	Z
说明：	将 f 寄存器的内容清零，Z 位置 1。

CLRW	**将 W 寄存器清零**
语法：	[*标号*] CLRW
操作数：	无
操作：	00h → (W) 1 → Z
受影响的状态位：	Z
说明：	将W寄存器清零。全零标志位（Z）置 1。

CLRWDT	**清零看门狗定时器**
语法：	[*标号*] CLRWDT
操作数：	无
操作：	00h → WDT 0 → WDT 预分频器， 1 → $\overline{TO}$ 1 → $\overline{PD}$
受影响的状态位：	$\overline{TO}$ 和 $\overline{PD}$
说明：	CLRWDT 指令复位看门狗定时器。还将复位 WDT 的预分频器。 状态位 $\overline{TO}$ 和 $\overline{PD}$ 置 1。

COMF	**f 取反**
语法：	[*标号*] COMF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	($\bar{f}$) → (目标寄存器)
受影响的状态位：	Z
说明：	将 f 寄存器的内容取反。如果 d 为 0，结果存放到 W 寄存器。如果 d 为 1，结果存回寄存器 f。

DECF	**f 递减 1**
语法：	[*标号*] DECF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(f) - 1 → (目标寄存器)
受影响的状态位：	Z
说明：	将 f 寄存器的内容减 1。如果 d 为 0，结果存放在W寄存器。如果d为 1，结果存回寄存器 f。

© 2011-2014 Microchip Technology Inc.

DECFSZ	f 递减 1，为 0 则跳过
语法：	[*标号*] DECFSZ f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(f) - 1 → (目标寄存器)； 结果 = 0 则跳过
受影响的状态位：	无
说明：	将 f 寄存器的内容减 1。如果 d 为 0，结果存放到W寄存器。如果d为 1，结果存放到寄存器 f。 如果结果为 1，则执行下一条指令。如果结果为 0，则转而执行一条 NOP 指令，从而使该指令成为双周期指令。

INCFSZ	f 递增 1，为 0 则跳过
语法：	[*标号*] INCFSZ f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(f) + 1 → (目标寄存器)， 结果 = 0 则跳过
受影响的状态位：	无
说明：	将 f 寄存器的内容加 1。如果 d 为 0，结果存放到W寄存器。如果d为 1，结果存放到寄存器 f。 如果结果为 1，则执行下一条指令。如果结果为0，则转而执行NOP指令，从而使该指令成为双周期指令。

GOTO	无条件跳转
语法：	[*标号*] GOTO k
操作数：	0 ≤ k ≤ 2047
操作：	k → PC<10:0> PCLATH<4:3> → PC<12:11>
受影响的状态位：	无
说明：	GOTO 是无条件跳转指令。将 11 位立即数值装入 PC 位 <10:0>。将 PCLATH<4:3> 的内容装入 PC 的高位。 GOTO 是双周期指令。

IORLW	立即数与 W 作逻辑或运算
语法：	[*标号*] IORLW k
操作数：	0 ≤ k ≤ 255
操作：	(W) .OR. k → (W)
受影响的状态位：	Z
说明：	将 W 寄存器的内容与 8 位立即数 k 作逻辑或运算。结果存放在 W 寄存器。

INCF	f 递增 1
语法：	[*标号*] INCF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(f) + 1 → (目标寄存器)
受影响的状态位：	Z
说明：	将 f 寄存器的内容递增 1。如果 d 为 0，结果存放到 W 寄存器。如果 d 为 1，结果存回寄存器 f。

IORWF	W 与 f 作逻辑或运算
语法：	[*标号*] IORWF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(W) .OR. (f) → (目标寄存器)
受影响的状态位：	Z
说明：	W寄存器与f寄存器作逻辑或运算。如果 d 为 0，结果存放到 W 寄存器。如果d为1，结果存回寄存器f。

MOVF	**传送 f**
语法：	[*标号*] MOVF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(f) → (目标寄存器)
受影响的状态位：	Z
说明：	根据 d 的状态，寄存器 f 的内容传送到目标寄存器。如果 d = 0，目标寄存器为 W 寄存器。如果 d = 1，则目标寄存器为文件寄存器 f。由于状态标志位 Z 会受到指令结果的影响，可用 d = 1 检测文件寄存器。
指令字数：	1
指令周期数：	1
示例：	`MOVF FSR, 0` 执行指令后 W = FSR 寄存器中的值 Z = 1

MOVWF	**将 W 的内容传送到 f**
语法：	[*标号*] MOVWF f
操作数：	0 ≤ f ≤ 127
操作：	(W) → (f)
受影响的状态位：	无
说明：	将数据从 W 寄存器传送到寄存器 f。
指令字数：	1
指令周期数：	1
示例：	`MOVWF OPTION_REG` 执行指令前 OPTION_REG = 0xFF W = 0x4F 执行指令后 OPTION_REG = 0x4F W = 0x4F

MOVLW	**将立即数传送到 W**
语法：	[*标号*] MOVLW k
操作数：	0 ≤ k ≤ 255
操作：	k → (W)
受影响的状态位：	无
说明：	将 8 位立即数 k 装入 W 寄存器。“无关位”被汇编为 0。
指令字数：	1
指令周期数：	1
示例：	`MOVLW 0x5A` 执行指令后 W = 0x5A

NOP	**空操作**
语法：	[*标号*] NOP
操作数：	无
操作：	空操作
受影响的状态位：	无
说明：	空操作。
指令字数：	1
指令周期数：	1
示例：	`NOP`

© 2011-2014 Microchip Technology Inc.

RETFIE	从中断返回
语法：	[*标号*] RETFIE
操作数：	无
操作：	TOS → PC， 1 → GIE
受影响的状态位：	无
说明：	从中断返回。执行出栈操作，将栈顶（TOS）的内容装入 PC。通过将全局中断允许位 GIE（INTCON<7>）置 1 允许中断。这是一条双周期指令。
指令字数：	1
指令周期数：	2
示例：	`RETFIE` 中断后 PC = TOS GIE = 1

RETLW	返回并将立即数传送到 **W**
语法：	[*标号*] RETLW k
操作数：	0 ≤ k ≤ 255
操作：	k → (W)； TOS → PC
受影响的状态位：	无
说明：	将 8 位立即数 k 装入 W 寄存器。将栈顶内容（返回地址）装入程序计数器。这是一条双周期指令。
指令字数：	1
指令周期数：	2

示例：	`CALL TABLE;W contains`
	`;table offset`
	`;value`
	`GOTO DONE`
	`TABLE`
	•
	•
	`ADDWF PC ;W = offset`
	`RETLW k1 ;Begin table`
	`RETLW k2 ;`
	•
	•
	•
	`RETLW kn ;End of table`
	`DONE`
	执行指令前
	W = 0x07
	执行指令后
	W = k8 的值

RETURN	从子程序返回
语法：	[*标号*] RETURN
操作数：	无
操作：	TOS → PC
受影响的状态位：	无
说明：	从子程序返回。执行出栈操作，将栈顶（TOS）内容装入程序计数器。这是一条双周期指令。

RLF 对 f 执行带进位的循环左移

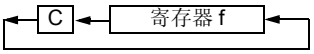
语法： [*标号*] RLF f,d

操作数： $0 \le f \le 127$
$d \in [0,1]$

操作： 参见下面的说明

受影响的状态位： C

说明： 将寄存器 f 的内容连同进位标志位一起循环左移 1 位。如果 d 为 0，结果存放到 W 寄存器。如果 d 为 1，结果存回寄存器 f。



指令字数： 1

指令周期数： 1

示例：

```
RLF     REG1,0
```

执行指令前

```
REG1    =   1110 0110
C       =   0
```

执行指令后

```
REG1    =   1110 0110
W       =   1100 1100
C       =   1
```

RRF 对 f 执行带进位的循环右移

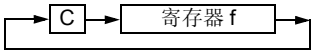
语法： [*标号*] RRF f,d

操作数： $0 \le f \le 127$
$d \in [0,1]$

操作： 参见下面的说明

受影响的状态位： C

说明： 将寄存器 f 的内容连同进位标志位一起循环右移 1 位。如果 d 为 0，结果存放到 W 寄存器。如果 d 为 1，结果存回寄存器 f。



SLEEP 进入休眠模式

语法： [*标号*] SLEEP

操作数： 无

操作： 00h → WDT，
0 → WDT 预分频器，
1 → $\overline{TO}$，
0 → $\overline{PD}$

受影响的状态位： $\overline{TO}$ 和 $\overline{PD}$

说明： 掉电状态位 $\overline{PD}$ 清零。超时状态位 $\overline{TO}$ 置 1。看门狗定时器及其预分频器清零。
振荡器停振，处理器进入休眠模式。

SUBLW 从立即数中减去 W 的内容

语法： [*标号*] SUBLW k

操作数： $0 \le k \le 255$

操作： k - (W) → (W)

受影响的状态位： C、 DC 和 Z

说明： 从 8 位立即数 k 中减去 W 寄存器的内容（采用二进制补码方法进行运算）。结果存放到 W 寄存器。

结果	条件
C = 0	W > k
C = 1	W ≤ k
DC = 0	W<3:0> > k<3:0>
DC = 1	W<3:0> ≤ k<3:0>

© 2011-2014 Microchip Technology Inc.

SUBWF **f 减去 W**

语法： [*标号*] SUBWF f,d

操作数： $0 \le f \le 127$
$d \in [0,1]$

操作： (f) - (W) → (目标寄存器)

受影响的状态位： C、 DC 和 Z

说明： 从寄存器 f 中减去 W 寄存器的内容（采用二进制补码方法进行运算）。如果 d 为 0，结果存放在 W 寄存器。如果 d 为 1，结果存回寄存器 f。

C = 0	W > f
C = 1	W ≤ f
DC = 0	W<3:0> > f<3:0>
DC = 1	W<3:0> ≤ f<3:0>

SWAPF **将 f 中的两个半字节进行交换**

语法： [*标号*] SWAPF f,d

操作数： $0 \le f \le 127$
$d \in [0,1]$

操作： (f<3:0>) → (目标寄存器 <7:4>)，
(f<7:4>) → (目标寄存器 <3:0>)

受影响的状态位： 无

说明： 将寄存器 f 的高半字节和低半字节交换。如果 d 为 0，结果存放到 W 寄存器。如果 d 为 1，结果存放到寄存器 f。

XORLW **立即数与 W 作逻辑异或运算**

语法： [*标号*] XORLW k

操作数： $0 \le k \le 255$

操作： (W) .XOR. k → (W)

受影响的状态位： Z

说明： 将 W 寄存器的内容与 8 位立即数 k 作逻辑异或运算。结果存放到 W 寄存器。

XORWF **W 与 f 作逻辑异或运算**

语法： [*标号*] XORWF f,d

操作数： $0 \le f \le 127$
$d \in [0,1]$

操作： (W) .XOR. (f) → (目标寄存器)

受影响的状态位： Z

说明： 将 W 寄存器的内容与 f 寄存器的内容作逻辑异或运算。如果 d 为 0，结果存放到 W 寄存器。如果 d 为 1，结果存回寄存器 f。

24.0 电气规范

24.1 绝对最大值 (†)

偏置电压下的环境温度 ..... -40°C 至 +125°C
储存温度 ..... -65°C 至 +150°C
引脚相对于 VSS 的电压
VDD 引脚
PIC10F320/322 ..... -0.3V 至 +6.5V
PIC10LF320/322 ..... -0.3V 至 +4.0V
$\overline{\text{MCLR}}$ 引脚 ..... -0.3V 至 +9.0V
所有其他引脚 ..... -0.3V 至 （VDD + 0.3V）
最大电流
VSS 引脚 (1)
-40°C ≤ TA ≤ +85°C ..... 150 mA
-40°C ≤ TA ≤ +125°C ..... 100 mA
VDD 引脚 (1)
-40°C ≤ TA ≤ +85°C ..... 150 mA
-40°C ≤ TA ≤ +125°C ..... 100 mA
任一 I/O 引脚 ..... ± 25 mA
钳位电流 IK （VPIN < 0 或 VPIN > VDD） ..... ± 20 mA

注 1： 最大电流要求 I/O 引脚上具有均匀的负载分布。最大电流可能受限于器件封装功率耗散特性，请参见第 24.4 节 “散热考虑” 来计算器件规范值。

† 注： 如果器件工作条件超过上述 “绝对最大值”，则可能会对器件造成永久性损坏。上述值仅为运行条件极大值，我们建议不要使器件在该规范规定的范围以外运行。器件长时间工作在最大值条件下，其稳定性会受到影响。

© 2011-2014 Microchip Technology Inc.

24.2 标准工作条件

所有器件的标准工作条件定义如下：

工作电压： $V_{DDMIN} \le V_{DD} \le V_{DDMAX}$
工作温度： $T_{A_MIN} \le T_A \le T_{A_MAX}$

V_{DD} —— 供电电压工作范围 (1)

PIC10LF320/322

V_{DDMIN} （Fosc ≤ 16 MHz） ..... +1.8V
V_{DDMIN} （16 MHz < Fosc ≤ 20 MHz） ..... +2.5V
V_{DDMAX} ..... +3.6V

PIC10F320/322

V_{DDMIN} （Fosc ≤ 16 MHz） ..... +2.3V
V_{DDMIN} （16 MHz < Fosc ≤ 20 MHz） ..... +2.5V
V_{DDMAX} ..... +5.5V

T_A —— 工作环境温度范围

工业级温度

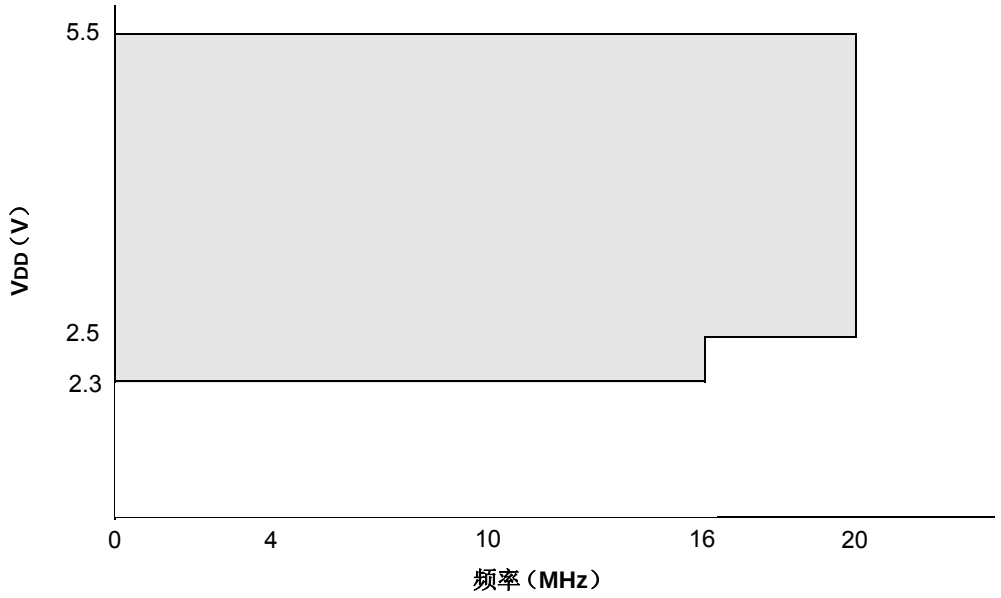
T_{A_MIN} ..... -40°C
T_{A_MAX} ..... +85°C

扩展级温度

T_{A_MIN} ..... -40°C
T_{A_MAX} ..... +125°C

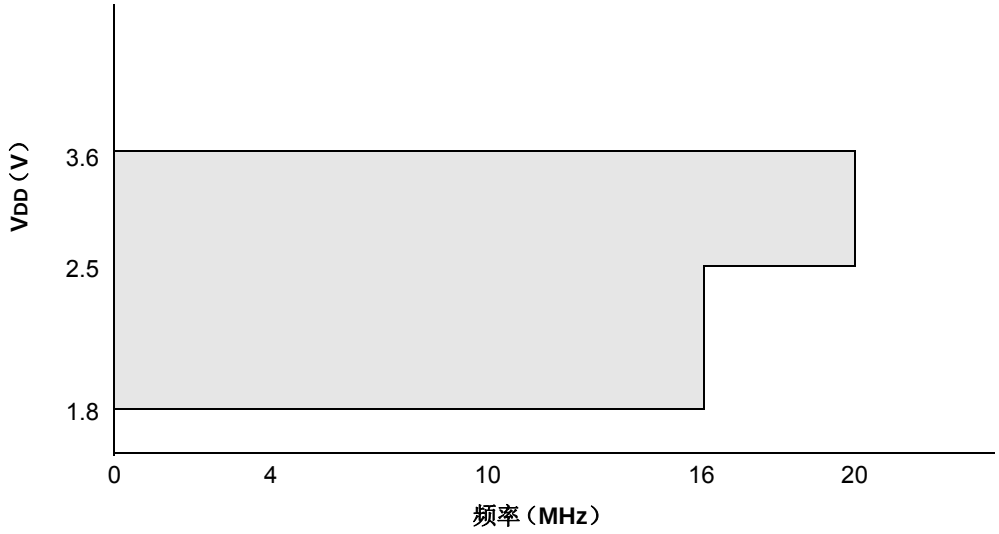
注 1： 请参见参数 D001，直流特性：供电电压。

图 24-1： PIC10F320/322 电压 — 频率关系图， -40°C ≤ TA ≤ +125°C



注 1： 阴影区域表示允许的电压和频率的组合。
2： 关于每种振荡器模式所支持的频率，请参见表 24-6。

图 24-2： PIC10LF320/322 电压 — 频率关系图， -40°C ≤ TA ≤ +125°C



注 1： 阴影区域表示允许的电压和频率的组合。
2： 关于每种振荡器模式所支持的频率，请参见表 24-6。

© 2011-2014 Microchip Technology Inc.

24.3 直流特性

表 24-1： 供电电压

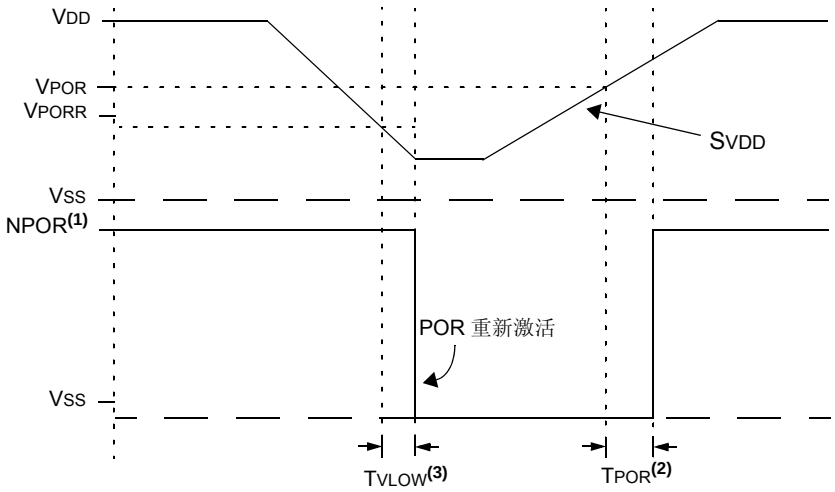
PIC10LF320/322			标准工作条件 （除非另外声明）				
PIC10F320/322							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
D001	VDD	供电电压					
			1.8 2.5	— —	3.6 3.6	V V	FOSC ≤ 16 MHz FOSC ≤ 20 MHz
D001			2.3 2.5	— —	5.5 5.5	V V	FOSC ≤ 16 MHz FOSC ≤ 20 MHz
D002*	VDR	RAM 数据保持电压 (1)					
			1.5	—	—	V	器件处于休眠模式
D002*			1.7	—	—	V	器件处于休眠模式
	VPOR*	上电复位释放电压	—	1.6	—	V	
	VPORR*	上电复位重新激活电压	—	0.8	—	V	器件处于休眠模式
			—	1.7	—	V	器件处于休眠模式
D004*	SVDD	VDD 上升速率确保能够产生内部上电复位信号	0.05	—	—	V/ms	详情请参见第 5.1 节 “上电复位（POR）”。

* 这些参数仅为特征值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 3.3V、 25°C 条件下的值。这些参数仅作为设计参考，未经测试。

注 1： 该电压是休眠模式下保证不丢失 RAM 数据的最小 VDD。

图 24-3： VDD 缓慢上升时的 POR 和 POR 重新激活



注 1： 当 NPOR 为低电平时，器件保持在复位状态。
2： TPOR 的典型值为 1 μs。
3： TVLOW 的典型值为 2.7 μs。

© 2011-2014 Microchip Technology Inc.

表 24-2： 供电电流 （IDD） (1,2)

PIC10LF320/322			标准工作条件 （除非另外声明）				
PIC10F320/322							
参数编号	器件特性	最小值	典型值 †	最大值	单位	条件	
						VDD	注
D013		—	34	—	μA	1.8	FOSC = 500 kHz EC 模式
		—	60	—	μA	3.0	
D013		—	76	—	μA	2.3	FOSC = 500 kHz EC 模式
		—	110	—	μA	3.0	
		—	153	—	μA	5.0	
D014		—	200	396	μA	1.8	FOSC = 8 MHz EC 模式
		—	380	650	μA	3.0	
D014		—	302	430	μA	2.3	FOSC = 8 MHz EC 模式
		—	420	655	μA	3.0	
		—	502	775	μA	5.0	
D015		—	0.8	1.3	mA	3.0	FOSC = 20 MHz EC 模式
		—	1.1	1.8	mA	3.6	
D015		—	0.8	1.4	mA	3.0	FOSC = 20 MHz EC 模式
		—	1.1	1.8	mA	5.0	
D016		—	2.4	—	μA	1.8	FOSC = 32 kHz LFINTOSC 模式，85°C
		—	7	—	μA	3.0	
D016		—	31	—	μA	2.3	FOSC = 32 kHz LFINTOSC 模式，85°C
		—	40	—	μA	3.0	
		—	71	—	μA	5.0	
D016A		—	2.4	—	μA	1.8	FOSC = 32 kHz LFINTOSC 模式， 125°C
		—	7	—	μA	3.0	
D016A		—	31	—	μA	2.3	FOSC = 32 kHz LFINTOSC 模式， 125°C
		—	40	—	μA	3.0	
		—	71	—	μA	5.0	

注 1： 有效工作模式下，所有 IDD 测试的测试条件为：CLKIN = 外部方波，轨到轨满幅；所有 I/O 引脚均为三态，上拉至 VDD；$\overline{MCLR}$ = VDD；禁止 WDT。

2： 供电电流主要受工作电压和频率的影响。其他因素如 I/O 引脚负载和开关速率、振荡器类型、内部代码执行模式以及温度也对电流消耗有影响。

表 24-2： 供电电流（IDD）(1,2)（续）

PIC10LF320/322			标准工作条件（除非另外声明）				
PIC10F320/322							
参数编号	器件特性	最小值	典型值 †	最大值	单位	条件	
						VDD	注
D017		—	293	—	μA	1.8	FOSC = 500 kHz HFINTOSC 模式
		—	286	—	μA	3.0	
D017		—	272	—	μA	2.3	FOSC = 500 kHz HFINTOSC 模式
		—	310	—	μA	3.0	
		—	372	—	μA	5.0	
D018		—	350	1.0	mA	1.8	FOSC = 8 MHz HFINTOSC 模式
		—	530	1.1	mA	3.0	
D018		—	0.45	1.0	mA	2.3	FOSC = 8 MHz HFINTOSC 模式
		—	0.56	1.1	mA	3.0	
		—	0.64	1.2	mA	5.0	
D019		—	0.46	1.1	mA	1.8	FOSC = 16 MHz HFINTOSC 模式
		—	0.73	1.2	mA	3.0	
D019		—	0.60	1.1	mA	2.3	FOSC = 16 MHz HFINTOSC 模式
		—	0.76	1.2	mA	3.0	
		—	0.85	1.3	mA	5.0	

注 1： 有效工作模式下，所有 IDD 测试的测试条件为：CLKIN = 外部方波，轨到轨满幅；所有 I/O 引脚均为三态，上拉至 VDD；MCLR = VDD；禁止 WDT。

2： 供电电流主要受工作电压和频率的影响。其他因素如 I/O 引脚负载和开关速率、振荡器类型、内部代码执行模式以及温度也对电流消耗有影响。

© 2011-2014 Microchip Technology Inc.

表 24-3： 掉电电流 （IPD）(1,2)

PIC10LF320/322			标准工作条件 （除非另外声明）					
PIC10F320/322								
参数编号	器件特性	最小值	典型值 †	最大值 +85°C	最大值 +125°C	单位	条件 VDD	条件 注
D023		—	0.06	1.1	2	μA	1.8	禁止 WDT、 BOR 和 FVR，所有外设不工作
		—	0.08	1.3	2	μA	3.0	
D023		—	0.20	1.1	2	μA	2.3	禁止 WDT、 BOR 和 FVR，所有外设不工作
		—	0.30	1.4	2	μA	3.0	
		—	0.40	2.4	2.4	μA	5.0	
D024		—	0.5	9	11	μA	1.8	WDT 电流 （注 1）
		—	0.8	11	13	μA	3.0	
D024		—	4.0	10	12	μA	2.3	WDT 电流 （注 1）
		—	4.2	12	14	μA	3.0	
		—	4.3	14	16	μA	5.0	
D025		—	30	96	120	μA	1.8	FVR 电流
		—	39	106	123	μA	3.0	
D025		—	32	96	120	μA	2.3	FVR 电流
		—	39	106	133	μA	3.0	
		—	70	136	170	μA	5.0	
D026		—	7.5	16	18	μA	3.0	BOR 电流 （注 1）
D026		—	8	18	20	μA	3.0	BOR 电流 （注 1）
		—	9	20	20.2	μA	5.0	
D026A		—	2.7	10	15	μA	3.0	LPBOR 电流
D026A		—	3.0	10	15	μA	3.0	LPBOR 电流
		—	3.2	15	20	μA	5.0	
D028		—	0.1	4	5	μA	1.8	A/D 电流 （注 1，注 3），未进行转换
		—	0.1	5	6	μA	3.0	
D028		—	3.4	6	7	μA	2.3	A/D 电流 （注 1，注 3），未进行转换
		—	3.6	7	8	μA	3.0	
		—	3.8	8	9	μA	5.0	
D029		—	250	—	—	μA	1.8	A/D 电流 （注 1，注 3），正在进行转换
		—	250	—	—	μA	3.0	
D029		—	280	—	—	μA	2.3	A/D 电流 （注 1，注 3），正在进行转换
		—	280	—	—	μA	3.0	
		—	280	—	—	μA	5.0	

* 这些参数仅为特征值，未经测试。

† 除非另外声明，否则 “典型值”栏中的数据均为 3.0V、 25°C 条件下的值。这些参数仅作为设计参考，未经测试。

注 1： 外设电流是基本 IDD 或 IPD 电流以及该外设使能时额外消耗的电流之和。外设 Δ 电流可以从该外设电流中减去基本 IDD 或 IPD 电流而得出。在计算总电流消耗时应使用最大值。

2： 休眠模式下的掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式时，所有 I/O 引脚处于高阻态并连接到 VDD 时测得的。

3： A/D 振荡器的时钟源为 FRC。

表 24-4： I/O 端口

标准工作条件（除非另外声明）							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
	V_{IL}	输入低电压					
		I/O 端口：					
D032		带 TTL 缓冲器	—	—	0.8	V	$4.5V \le V_{DD} \le 5.5V$
D032A			—	—	0.15 V_{DD}	V	$1.8V \le V_{DD} \le 4.5V$
D033		带施密特触发缓冲器	—	—	0.2 V_{DD}	V	$2.0V \le V_{DD} \le 5.5V$
D034		$\overline{MCLR}$	—	—	0.2 V_{DD}	V	
	V_{IH}	输入高电压					
		I/O 端口：					
D040		带 TTL 缓冲器	2.0	—	—	V	$4.5V \le V_{DD} \le 5.5V$
D040A			0.25 V_{DD} + 0.8	—	—	V	$1.8V \le V_{DD} \le 4.5V$
D041		带施密特触发缓冲器	0.8 V_{DD}	—	—	V	$2.0V \le V_{DD} \le 5.5V$
D042		$\overline{MCLR}$	0.8 V_{DD}	—	—	V	
	I_{IL}	输入泄漏电流 (2)					
D060		I/O 端口	—	± 5 ± 5	± 125 ± 1000	nA nA	$V_{SS} \le V_{PIN} \le V_{DD}$，引脚处于高阻态（85°C 时） 125°C
D061		$\overline{MCLR}$	—	± 50	± 200	nA	$V_{SS} \le V_{PIN} \le V_{DD}$（85°C 时）
	I_{PUR}	弱上拉电流					
D070*			25 25	100 140	200 300	μA	V_{DD} = 3.3V， V_{PIN} = V_{SS} V_{DD} = 5.0V， V_{PIN} = V_{SS}
	V_{OL}	输出低电压					
D080		I/O 端口	—	—	0.6	V	I_{OL} = 8 mA， V_{DD} = 5V I_{OL} = 6 mA， V_{DD} = 3.3V I_{OL} = 1.8 mA， V_{DD} = 1.8V
	V_{OH}	输出高电压					
D090		I/O 端口	V_{DD} - 0.7	—	—	V	I_{OH} = 3.5 mA， V_{DD} = 5V I_{OH} = 3 mA， V_{DD} = 3.3V I_{OH} = 1 mA， V_{DD} = 1.8V

* 这些参数仅为特征值，未经测试。

† 除非另外声明，否则“典型值”栏中的数据均为 3.0V、 25°C 条件下的值。这些参数仅作为设计参考，未经测试。

注 1： 负电流定义为引脚的拉电流。

2： MCLR 引脚上的泄漏电流主要取决于所施加的电压。规定电压为正常工作条件下的电压。在不同的输入电压下可能测得更高的泄漏电流。

© 2011-2014 Microchip Technology Inc.

表 24-5： 存储器编程要求

标准工作条件 （除非另外声明）							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
		程序存储器编程规范					
D110	VIHH	$\overline{MCLR}$/VPP 引脚上的电压	8.0	—	9.0	V	（注 2）
D111	IDDP	编程期间的供电电流	—	—	10	mA	
D112		用于批量擦除的 VDD	2.7	—	VDD 最大值	V	
D113	VPEW	用于写或行擦除的 VDD	VDD 最小值	—	VDD 最大值	V	
D114	IPPPGM	擦 / 写期间 $\overline{MCLR}$/VPP 上的电流	—	—	1.0	mA	
D115	IDDPGM	擦 / 写期间 VDD 上的电流	—		5.0	mA	
		闪存程序存储器					
D121	EP	电池耐用性	10K	—	–	E/W	-40°C 至 +85°C （注 1）
D122	VPR	用于读操作的 VDD	VDD 最小值	—	VDD 最大值	V	
D123	TIW	自定时的写周期	—	2	2.5	ms	
D124	TRETD	特性保持时间	40	—	—	年	假设没有违反其他规范

† 除非另外声明，否则 “典型值”栏中的数据均为 3.0V、 25°C 条件下的值。这些参数仅作为设计参考，未经测试。

注 1： 自写和块擦除。

2： 仅在禁止单电源编程的情况下需要。

24.4 散热考虑

标准工作条件 （除非另外声明）					
参数编号	符号	特性	典型值	单位	条件
TH01	θ_{JA}	结点到环境的热阻	60	°C/W	6 引脚 SOT-23 封装
			80	°C/W	8 引脚 PDIP 封装
			90	°C/W	8 引脚 DFN 封装
TH02	θ_{JC}	结点到管壳的热阻	31.4	°C/W	6 引脚 SOT-23 封装
			24	°C/W	8 引脚 PDIP 封装
			24	°C/W	8 引脚 DFN 封装
TH03	T_{JMAX}	最高结温	150	°C	
TH04	PD	功耗	—	W	$P_D = P_{INTERNAL} + P_{I/O}$
TH05	$P_{INTERNAL}$	内部功耗	—	W	$P_{INTERNAL} = I_{DD} \times V_{DD}$(1)
TH06	$P_{I/O}$	I/O 功耗	—	W	$P_{I/O} = \Sigma (I_{OL} * V_{OL}) + \Sigma (I_{OH} * (V_{DD} - V_{OH}))$
TH07	P_{DER}	降额功耗	—	W	$P_{DER} = P_{DMAX} (T_J - T_A)/\theta_{JA}$(2)

注 1：I_{DD} 为单独运行芯片而不驱动输出引脚上的任何负载时的电流。
2：T_A = 环境温度。
3：T_J = 结温。

© 2011-2014 Microchip Technology Inc.

24.5 交流特性

可根据以下任一格式来创建时序参数符号：

1. TppS2ppS
2. TppS

T			
F	频率	T	时间

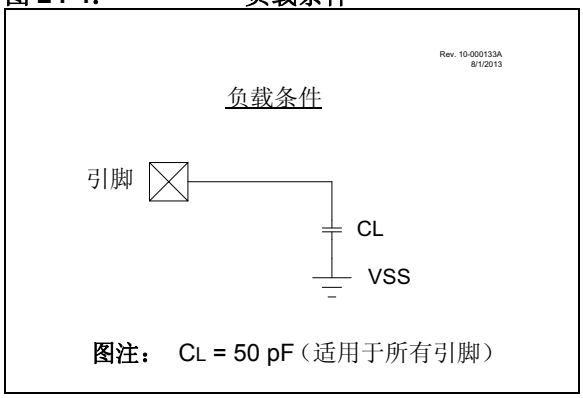
小写字母（pp）及其含义：

pp			
cc	CCP1	osc	CLKIN
ck	CLKR	rd	$\overline{\text{RD}}$
cs	$\overline{\text{CS}}$	rw	$\overline{\text{RD}}$ 或 $\overline{\text{WR}}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{\text{SS}}$
dt	数据输入	t0	T0CKI
io	I/O 端口	t1	T1CKI
mc	$\overline{\text{MCLR}}$	wr	$\overline{\text{WR}}$

大写字母及其含义：

S			
F	下降	P	周期
H	高	R	上升
I	无效（高阻态）	V	有效
L	低	Z	高阻态

图 24-4：　　负载条件



图注： C_L = 50 pF（适用于所有引脚）

图 24-5： 时钟时序

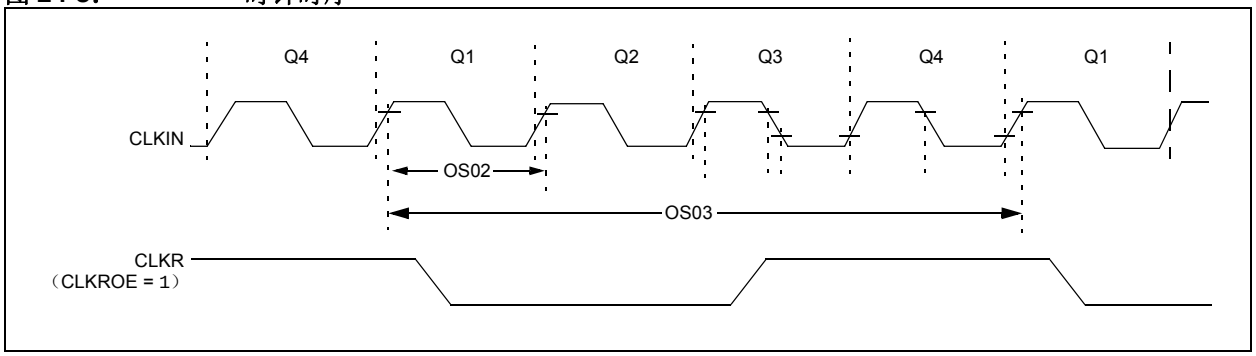


表 24-6： 时钟振荡器时序要求

标准工作条件 （除非另外声明）							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
OS01	FOSC	外部 CLKIN 频率 (1)	DC	—	20	MHz	EC 模式
OS02	TOSC	外部 CLKIN 周期 (1)	31.25	—	∞	ns	EC 振荡器模式
OS03	TCY	指令周期 (1)	200	TCY	DC	ns	TCY = 4/FOSC

* 这些参数仅为特征值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 3.0V、 25°C 条件下的值。这些参数仅作为设计参考，未经测试。

注 1： 指令周期（TCY）等于输入振荡器时基周期的 4 倍。所有规范值均基于器件在标准工作条件下执行代码时对应特定振荡器类型的特征数据。超出这些规范值可能导致振荡器运行不稳定和 / 或电流消耗超出预期值。所有器件在测试 “最小值” 时，都在 CLKIN 引脚上连接了外部时钟。当使用了外部时钟输入时，所有器件的 “最大值” 周期时限为 “DC”（没有时钟）。

表 24-7： 振荡器参数

标准工作条件 （除非另外声明）								
参数编号	符号	特性	频率容差	最小值	典型值 †	最大值	单位	条件
OS08	HFOSC	经过校准的内部 HFINTOSC 频率 (1)	±3%	—	16.0	—	MHz	0°C ≤ TA ≤ +85°C， VDD ≥ 2.3V
			-8% 至 +4%	—	16.0	—	MHz	-40°C ≤ TA ≤ +125°C
OS09	LFOSC	内部 LFINTOSC频率	±25%	—	31	—	kHz	
OS10*	TIOSCST	HFINTOSC 从休眠模式唤醒的起振时间	—	—	5	8	µs	

* 这些参数仅为特征值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 3.0V、 25°C 条件下的值。这些参数仅作为设计参考，未经测试。

注 1： 为了确保振荡器的频率容差，必须尽量靠近器件在 VDD 和 VSS 之间连接去耦电容。建议并联 0.1 µF 和 0.01 µF 的电容。

© 2011-2014 Microchip Technology Inc.

图 **24-6：** 器件 **VDD** 和温度范围内的 **HFINTOSC** 频率精度

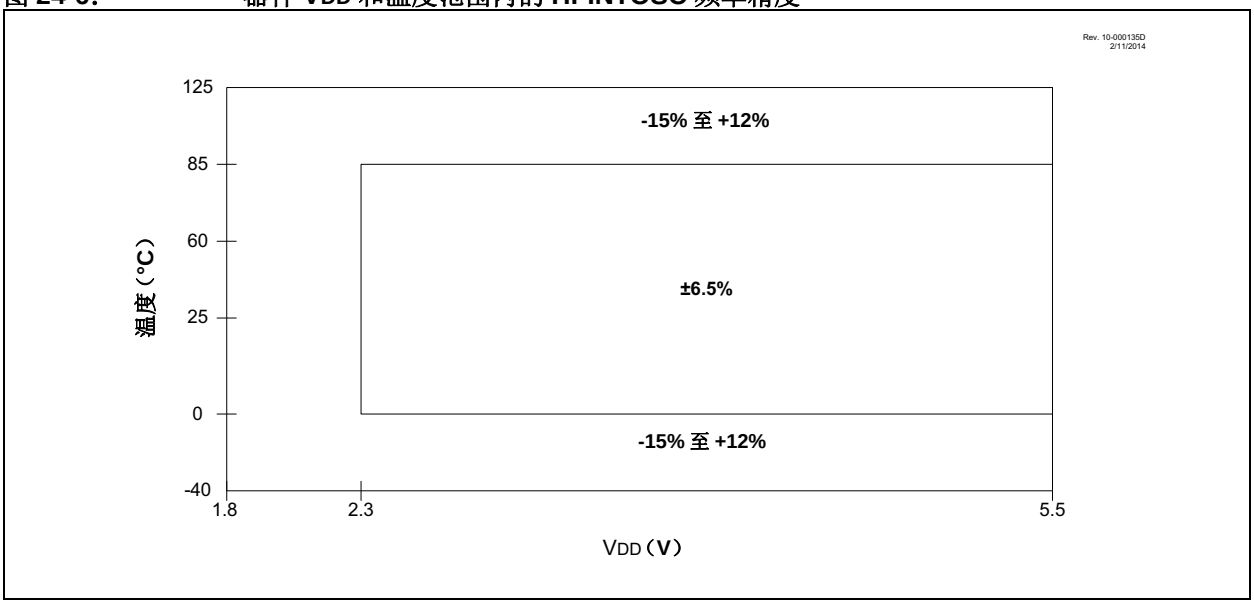


图 24-7： CLKR 和 I/O 时序

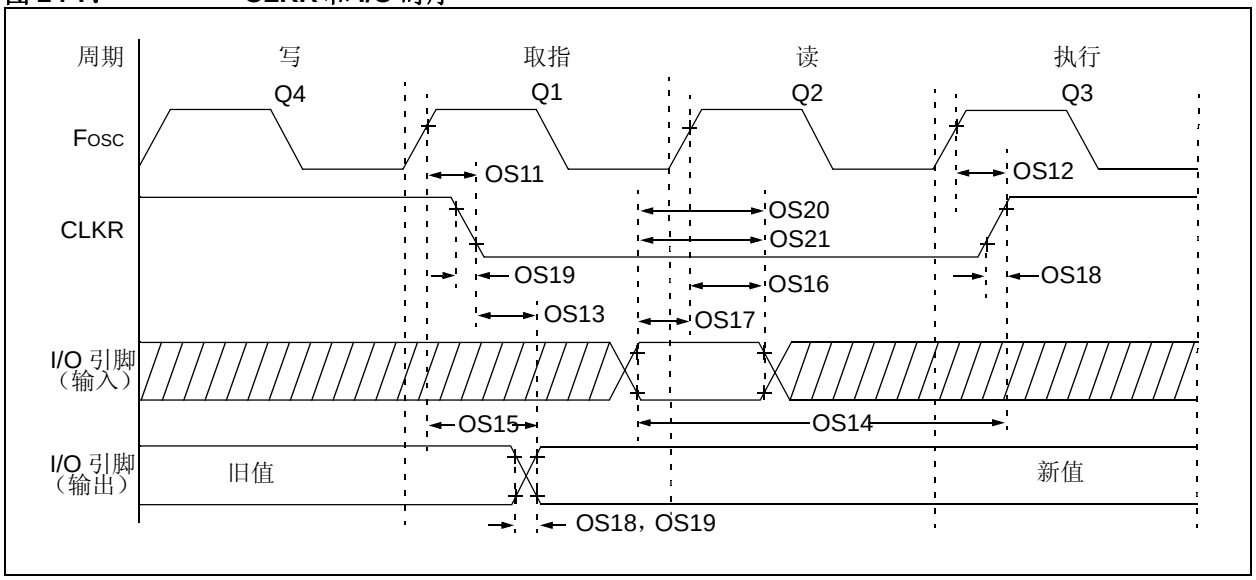


表 24-8： CLKR 和 I/O 时序参数

标准工作条件 （除非另外声明）							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
OS11	TosH2ckL	FOSC↑ 至 CLKOUT↓ 的时间 (1)	—	—	70	ns	3.3V ≤ VDD ≤ 5.0V
OS12	TosH2ckH	FOSC↑ 至 CLKOUT↑ 的时间 (1)	—	—	72	ns	3.3V ≤ VDD ≤ 5.0V
OS13	TckL2ioV	CLKOUT↓ 至端口输出有效的时间 (1)	—	—	20	ns	
OS14	TioV2ckH	CLKOUT↑ 之前端口输入有效的时间 (1)	TOSC + 200 ns	—	—	ns	
OS15	TosH2ioV	FOSC↑（Q1 周期）至端口输出有效的时间	—	50	70*	ns	3.3V ≤ VDD ≤ 5.0V
OS16	TosH2ioI	FOSC↑（Q2 周期）至端口输入无效的时间（I/O 输入保持时间）	50	—	—	ns	3.3V ≤ VDD ≤ 5.0V
OS17	TioV2osH	端口输入有效至 FOSC↑（Q2 周期）的时间（I/O 输入建立时间）	20	—	—	ns	
OS18*	TioR	端口输出上升时间	— —	40 15	72 32	ns	VDD = 1.8V 3.3V ≤ VDD ≤ 5.0V
OS19*	TioF	端口输出下降时间	— —	28 15	55 30	ns	VDD = 1.8V 3.3V ≤ VDD ≤ 5.0V
OS20*	Tinp	INT 引脚输入高电平或低电平的时间	25	—	—	ns	
OS21*	Tioc	电平变化中断信号的新输入电平时间	25	—	—	ns	

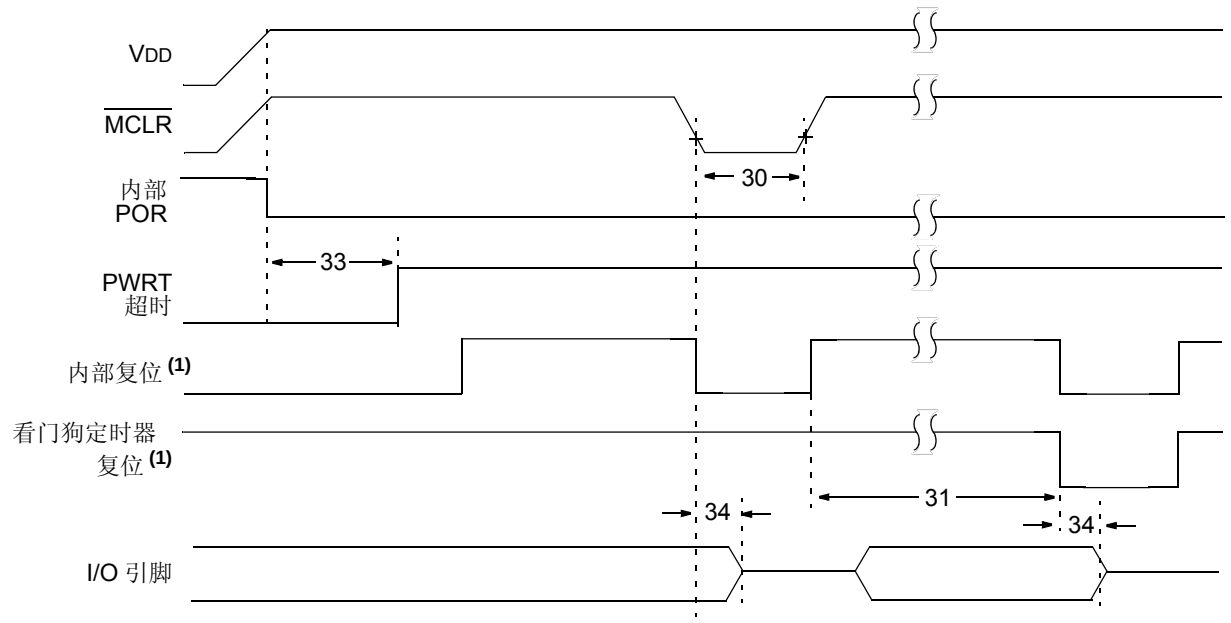
* 这些参数仅为特征值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 3.0V、 25°C 条件下的值。

注 1： 测量是在 EXTRC 模式下进行的，其中 CLKOUT 输出为 4 x TOSC。

© 2011-2014 Microchip Technology Inc.

图 24-8： 复位、看门狗定时器和上电延时定时器时序



注 1：低电平有效。

图 24-9： 欠压复位时序和特性

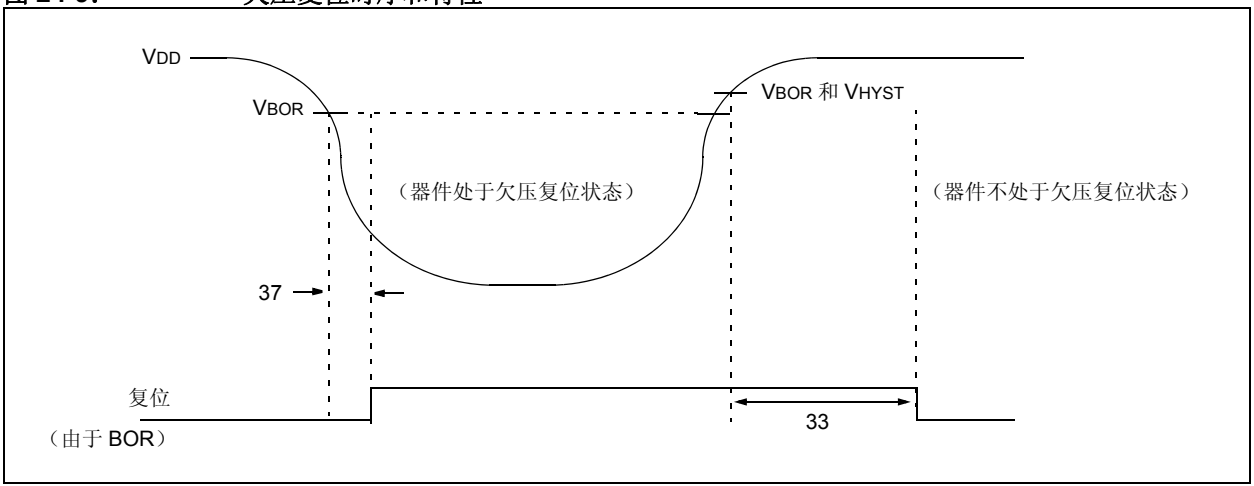


表 24-9： 复位、看门狗定时器、上电延时定时器和欠压复位参数

标准工作条件（除非另外声明）							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
30	TMCL	$\overline{\text{MCLR}}$ 脉宽（低电平）	2 5	— —	— —	μs μs	VDD = 3.3-5V，-40°C 至 +85°C VDD = 3.3-5V
31	TWDTLP	低功耗看门狗定时器超时周期	10	16	27	ms	VDD = 3.3V-5V 使用 1:16 预分频比
33*	TPWRT	上电延时定时器周期，$\overline{\text{PWRTE}}$ = 0	40	64	140	ms	
34*	TIOZ	自 $\overline{\text{MCLR}}$ 低电平或看门狗定时器复位起 I/O 处于高阻态的时间	—	—	2.0	μs	
35	VBOR	欠压复位电压 (1)	2.55 2.30 1.80	2.70 2.40 1.90	2.85 2.55 2.05	V V V	BORV = 0 BORV = 1（PIC10F320/322） BORV = 1（PIC10LF320/322）
36*	VHYST	欠压复位迟滞	0	25	50	mV	-40°C 至 +85°C
37*	TBORDC	欠压复位直流响应时间	1	3	5	μs	VDD ≤ VBOR

* 这些参数仅为特征值，未经测试。

† 除非另外声明，否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考，未经测试。

注 1：为了确保这些电压容差，必须尽量靠近器件在 VDD 和 VSS 之间连接去耦电容。建议并联 0.1 μF 和 0.01 μF 的电容。

图 24-10： TIMER0 和 TIMER1 外部时钟时序

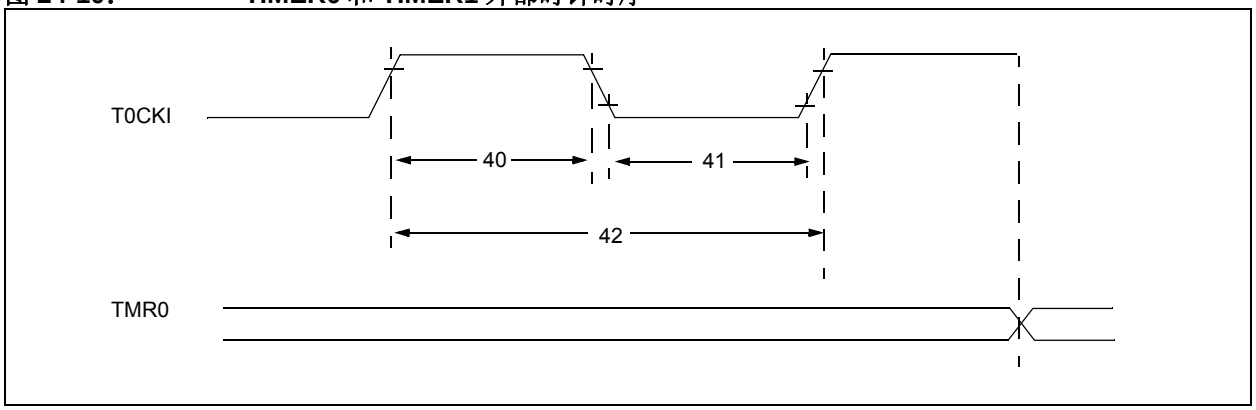


表 24-10： TIMER0 外部时钟要求

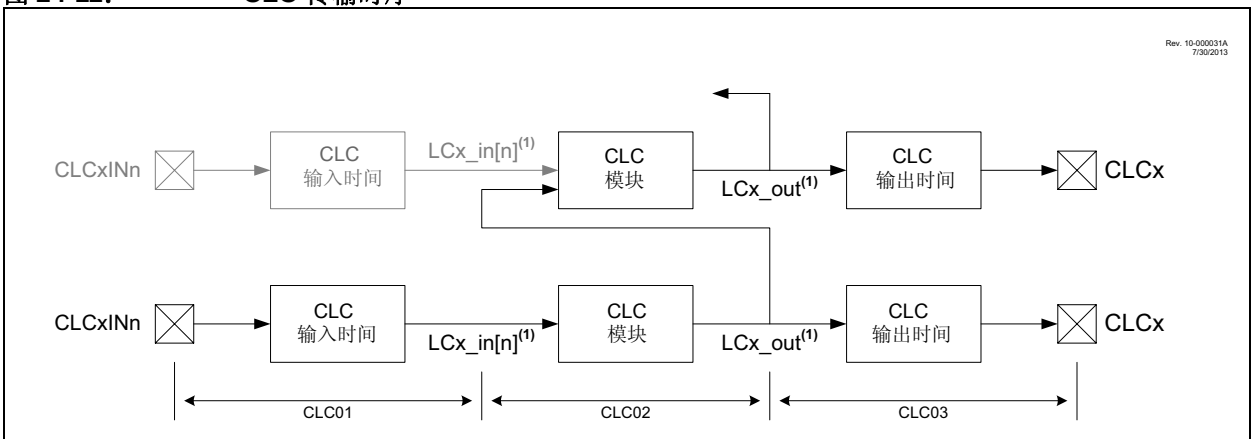
标准工作条件（除非另外声明）								
参数编号	符号	特性		最小值	典型值 †	最大值	单位	条件
40*	TT0H	T0CKI 高电平脉冲宽度	无预分频器	0.5 TCY + 20	—	—	ns	
			有预分频器	10	—	—	ns	
41*	TT0L	T0CKI 低电平脉冲宽度	无预分频器	0.5 TCY + 20	—	—	ns	
			有预分频器	10	—	—	ns	
42*	TT0P	T0CKI 周期		取以下两者中的较大值： 20 或 $\frac{T_{CY} + 40}{N}$	—	—	ns	N = 预分频值（2，4，…，256）

* 这些参数仅为特征值，未经测试。

† 除非另外声明，否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考，未经测试。

© 2011-2014 Microchip Technology Inc.

图 24-11： CLC 传输时序



注 1：请参见图 19-1 来确定特定的 CLC 信号。

表 24-11： 可配置逻辑单元 （CLC）特性

标准工作条件 （除非另外声明）							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
CLC01*	TCLCIN	CLC 输入时间	—	7	—	ns	
CLC02*	TCLC	CLC 模块输入至输出传输的时间	— —	24 12	— —	ns ns	VDD = 1.8V VDD > 3.6V
CLC03*	TCLCOUT	CLC 输出时间 上升时间	—	OS18	—	—	（注 1）
		下降时间	—	OS19	—	—	（注 1）
CLC04*	FCLCMAX	CLC 最高开关频率	—	45	—	MHz	

* 这些参数仅为特征值，未经测试。

† 除非另外声明，否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考，未经测试。

注 1：请参见表 24-8 了解 OS18 的上升时间和和 OS19 的下降时间。

表 24-12： A/D 转换器 （ADC）特性

标准工作条件 （除非另外声明）							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
AD01	NR	分辨率	—	—	8	位	
AD02	EIL	积分误差	—	—	±1.7	LSb	VREF = 3.0V
AD03	EDL	微分误差	—	—	±1	LSb	不丢失编码 VREF = 3.0V
AD04	EOFF	失调误差	—	—	±2.5	LSb	VREF = 3.0V
AD05	EGN	增益误差	—	—	±2.0	LSb	VREF = 3.0V
AD06	VREF	参考电压 (3)	1.8	—	VDD	V	VREF = (VREF+ - VREF-) （注 5）
AD07	VAIN	满量程范围	VSS	—	VREF	V	
AD08	ZAIN	模拟电压源推荐阻抗	—	—	10	kΩ	如果输入引脚上有外部 0.01μF 电容，则该值可以更高。

* 这些参数仅为特征值，未经测试。

† 除非另外声明，否则 “典型值”栏中的数据均为 3.0V、 25°C 条件下的值。这些参数仅作为设计参考，未经测试。

注 1： 总绝对误差包括积分误差、微分误差、失调误差和增益误差。

2： A/D 转换结果不会因输入电压的增加而减小，并且不会丢失编码。

3： ADC VREF 来自选为参考输入引脚的外部 VREF、 VDD 引脚或 FVR。

4： 当 ADC 关闭时，它除了消耗泄漏电流外，不消耗任何其他电流。掉电电流规范包括 ADC 模块消耗的任何泄漏电流。

5： 选择的 FVR 电压必须是 2.048V 或 4.096V。

表 24-13： A/D 转换要求

标准工作条件 （除非另外声明）							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
AD130*	TAD	A/D 时钟周期 A/D 内部 FRC 振荡器周期	1.0 1.0	— 1.6	6.0 6.0	μs μs	基于 TOSC ADCS<1:0> = 11 （ADRC 模式）
AD131	TCNV	转换时间 （不包括采集时间） (1)	—	9.5	—	TAD	将 GO/$\overline{\text{DONE}}$ 位置 1 以完成转换
AD132*	TACQ	采集时间	—	5.0	—	μs	

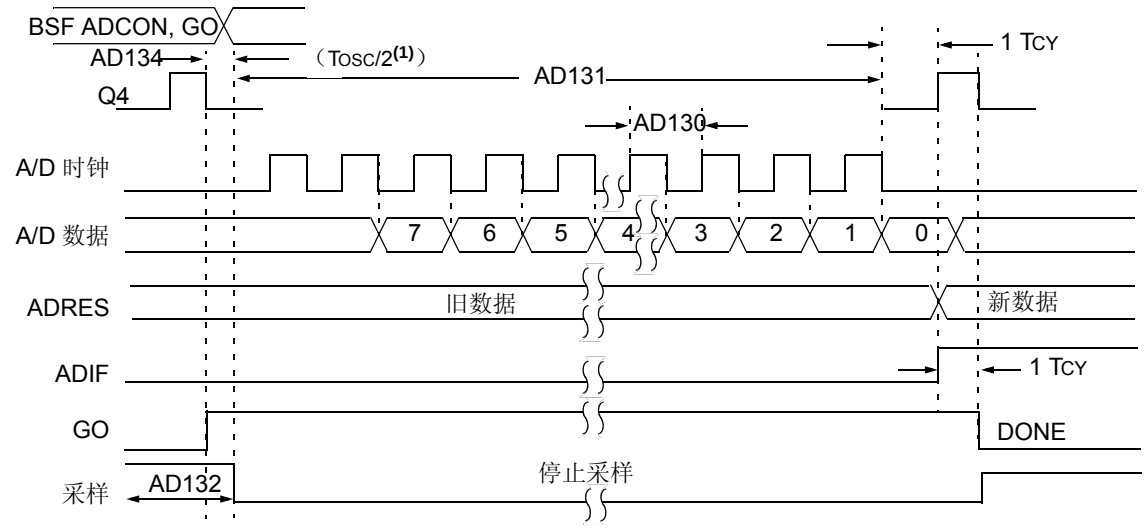
* 这些参数仅为特征值，未经测试。

† 除非另外声明，否则 “典型值”栏中的数据均为 3.0V、 25°C 条件下的值。这些参数仅作为设计参考，未经测试。

注 1： ADRES 寄存器可在下一个 TCY 周期被读取。

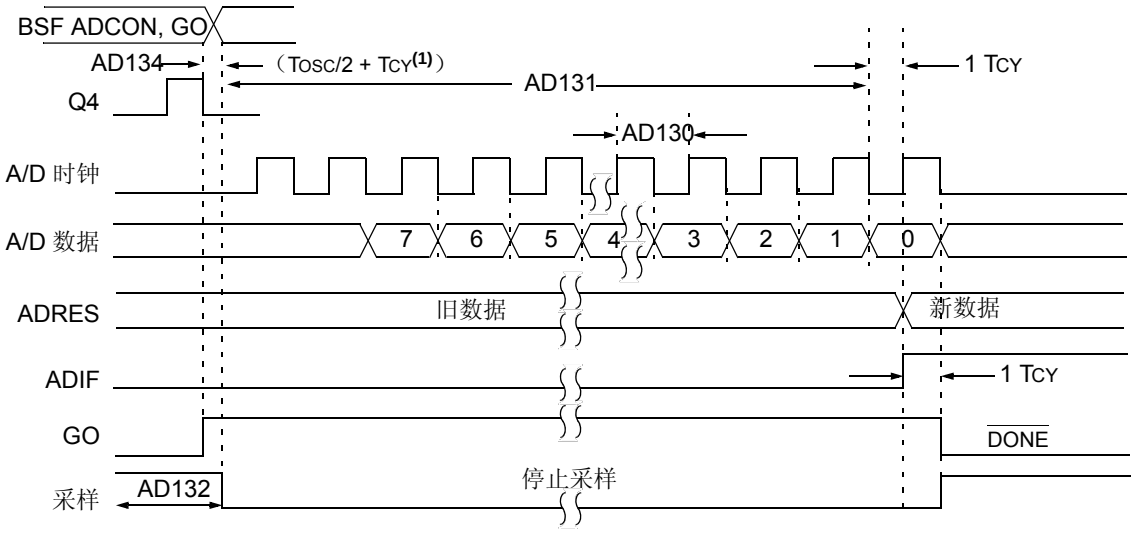
© 2011-2014 Microchip Technology Inc.

图 24-12： A/D 转换时序 （正常模式）



注 1： 如果选择 FRC 作为 A/D 时钟源，则在 A/D 时钟开始前要加上一个 TCY 时间，用于执行 SLEEP 指令。

图 24-13： A/D 转换时序 （休眠模式）



注 1： 如果选择 FRC 作为 A/D 时钟源，则在 A/D 时钟开始前要加上一个 TCY 时间，用于执行 SLEEP 指令。

25.0 直流和交流特性图表

当前没有可用图表。

© 2011-2014 Microchip Technology Inc.

26.0 开发支持

一系列软件及硬件开发工具对 PIC® 单片机（MCU）和 dsPIC® 数字信号控制器 （DSC）提供支持：

- 集成开发环境
  - MPLAB® X IDE 软件
- 编译器 / 汇编器 / 链接器
  - MPLAB XC 编译器
  - MPASM™ 汇编器
  - MPLINK™ 目标链接器 / MPLIB™ 目标库管理器
  - 适用于各种器件系列的 MPLAB 汇编器 / 链接器 / 库管理器
- 模拟器
  - MPLAB X SIM 软件模拟器
- 仿真器
  - MPLAB REAL ICE™ 在线仿真器
- 在线调试器 / 编程器
  - MPLAB ICD 3
  - PICkit™ 3
- 器件编程器
  - MPLAB PM3 器件编程器
- 低成本演示 / 开发板、评估工具包及入门工具包
- 第三方开发工具

26.1 MPLAB X 集成开发环境软件

MPLAB X IDE 是适用于 Microchip 和第三方软硬件开发工具统一的通用图形用户界面，可以在 Windows®、Linux 和 Mac OS® X 上运行。MPLAB X IDE 是一款全新的 IDE，它基于 NetBeans IDE，包含许多免费的软件组件和插件，适用于高性能的应用程序开发和调试。通过这一无缝交互的用户界面，在不同工具之间的迁移以及从软件模拟器到硬件调试和编程工具的升级都变得极为简便。

MPLAB X IDE 具有完善的项目管理、可视化的调用图、可配置的观察窗口以及包含代码补全功能和上下文菜单的功能丰富编辑器，因此对于新用户来说非常灵活和友好。 MPLAB X IDE 支持对多个项目使用多个工具和同时调试，因此也完全可以满足经验丰富用户的需求。

功能丰富的编辑器：

- 彩色高亮显示语法
- 智能代码补全功能，在输入代码时提供建议和提示
- 基于用户定义规则，代码自动格式化
- 即时解析

用户友好的可定制界面：

- 完全可定制界面：工具栏、工具栏图标、窗口和窗口放置等
- 调用图窗口

基于项目的工作空间：

- 多个项目
- 多个工具
- 多种配置
- 同时调试会话

文件历史和错误跟踪：

- 本地文件历史功能
- 内建对 Bugzilla 缺陷跟踪系统的支持

26.2 MPLAB XC 编译器

MPLAB XC编译器是适用于Microchip所有8位、16位和32位MCU以及DSC器件的完全ANSI C编译器。这些编译器提供强大的集成功能以及出色的代码优化功能，且易于使用。MPLAB XC 编译器可在 Windows、Linux 或 Mac OS X 上运行。

为方便进行源代码级调试，编译器提供了已针对 MPLAB X IDE 优化的调试信息。

MPLAB XC 编译器的免费版支持所有器件和命令，没有时间或存储容量限制，且为大多数应用程序提供了充分的代码优化。

MPLAB XC 编译器包含汇编器、链接器和实用程序。汇编器生成可重定位目标文件，然后通过链接器将生成的可重定位目标文件与其他可重定位目标文件或归档文件归档或链接在一起，进而生成可执行文件。MPLAB XC 编译器使用汇编器来生成目标文件。汇编器具有如下突出特性：

- 支持全部器件指令集
- 支持定点和浮点数据
- 命令行接口
- 丰富的伪指令集
- 灵活的宏语言
- 与 MPLAB X IDE 兼容

26.3 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于 PIC10/12/16/18 MCU。

MPASM 汇编器可生成用于 MPLINK 目标链接器的可重定位目标文件、Intel® 标准 HEX 文件、详细描述存储器使用状况和符号参考的 MAP 文件、包含源代码行及生成机器码的绝对 LST 文件以及用于调试的 COFF 文件。

MPASM 汇编器具有如下特性：

- 集成在 MPLAB X IDE 项目中
- 用户定义的宏可简化汇编代码
- 对多用途源文件进行条件汇编
- 允许完全控制汇编过程的指令

26.4 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK 目标链接器组合由 MPASM 汇编器生成的可重定位目标文件。通过使用链接器脚本中的伪指令，它还可链接预编译库中的可重定位目标文件。

MPLIB目标库管理器管理预编译代码库文件的创建和修改。当从源文件调用库中的一段子程序时，只有包含此子程序的模块被链接到应用程序。这样可使大型库在许多不同应用中被高效地利用。

目标链接器 / 库管理器具有如下特性：

- 高效地连接单个的库而不是许多小文件
- 通过将相关的模块组合在一起来增强代码的可维护性
- 只要列出、替换、删除和抽取模块，便可灵活地创建库

26.5 适用于各种器件系列的 MPLAB 汇编器、链接器和库管理器

MPLAB汇编器为PIC24和PIC32 MCU以及dsPIC DSC器件从符号汇编语言生成可重定位机器码。MPLAB XC 编译器使用该汇编器生成目标文件。汇编器产生可重定位目标文件之后，可将这些目标文件存档，或与其他可重定位目标文件和存档链接以生成可执行文件。该汇编器有如下显著特性：

- 支持整个器件指令集
- 支持定点和浮点数据
- 命令行接口
- 丰富的指令集
- 与 MPLAB X IDE 兼容

© 2011-2014 Microchip Technology Inc.

26.6 MPLAB X SIM 软件模拟器

MPLAB X SIM 软件模拟器通过在指令级对 PIC MCU 和 dsPIC DSC 进行模拟，可在 PC 主机环境下进行代码开发。对于任何给定的指令，都可以对数据区进行检查或修改，并通过一个全面的激励控制器来施加激励。可以将各寄存器记录在文件中，以便进行进一步的运行时分析。跟踪缓冲区和逻辑分析器的显示使软件模拟器还能记录和跟踪程序的执行、 I/O 的动作、大部分的外设及内部寄存器。

MPLAB X SIM 软件模拟器完全支持使用 MPLAB XC 编译器以及 MPASM 和 MPLAB 汇编器的符号调试。该软件模拟器可用于在硬件实验室环境外灵活地开发和调试代码，是一款完美且经济的软件开发工具。

26.7 MPLAB REAL ICE 在线仿真器系统

MPLAB REAL ICE 在线仿真器系统是 Microchip 针对其闪存 DSC 和 MCU 器件推出的新一代高速仿真器。结合 MPLAB X IDE 易于使用且功能强大的图形用户界面，该仿真器可对所有 8 位、16 位和 32 位 MCU 及 DSC 器件进行调试和编程。

该仿真器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与在线调试器系统兼容的连接器（RJ-11）或新型抗噪声、高速低压差分信号 （LVDS）互连电缆（CAT5）与目标板相连。

可通过 MPLAB X IDE 下载将来版本的固件，对该仿真器进行现场升级。在同类仿真器中，MPLAB REAL ICE 的优势十分明显：全速仿真、运行时变量观察、跟踪分析、复杂断点、逻辑探针、耐用的探针接口及较长 （长达 3 米）的互连电缆。

26.8 MPLAB ICD 3 在线调试器系统

MPLAB ICD 3 在线调试器系统是 Microchip 成本效益最高的高速硬件调试器 / 编程器，适用于 Microchip 的闪存 DSC 和 MCU 器件。结合 MPLAB X IDE 功能强大但易于使用的图形用户界面，该调试器可对 PIC 闪存单片机和 dsPIC DSC 进行调试和编程。

MPLAB ICD 3 在线调试器通过高速 USB 2.0 接口与设计工程师的PC相连，并利用与MPLAB ICD 2或MPLAB REAL ICE 系统兼容的连接器（RJ-11）与目标板相连。MPLAB ICD 3 支持所有 MPLAB ICD 2 连接器。

26.9 PICkit 3 在线调试器 / 编程器

结合 MPLAB X IDE 功能强大的图形用户界面，MPLAB PICkit 3 可对 PIC 闪存单片机和 dsPIC 数字信号控制器进行调试和编程，且价位较低。MPLAB PICkit 3 通过全速USB接口与设计工程师的PC相连，并利用Microchip 调试连接器 （RJ-11）（与 MPLAB ICD 3 和 MPLAB REAL ICE 兼容）与目标板相连。连接器使用两个器件 I/O 引脚和复位线来实现在线调试和在线串行编程 （In-Circuit Serial Programming™， ICSP™）。

PICkit 3 Debug Express 包括 PICkit 3、演示板和单片机、连接电缆和光盘 （内含用户指南、课程、教程、编译器和 MPLAB IDE 软件）。

26.10 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款符合 CE 规范的通用器件编程器，在 V_{DDMIN} 和 V_{DDMAX} 点对其可编程电压进行校验以确保可靠性最高。它有一个用来显示菜单和错误消息的大 LCD 显示器（128 x 64），以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSP 电缆。在单机模式下， MPLAB PM3 器件编程器不必与 PC 相连即可对 PIC MCU 和 dsPIC DSC 器件进行读取、校验和编程。在该模式下它还可设置代码保护。MPLAB PM3 通过 RS-232 或 USB 电缆连接到 PC 主机上。 MPLAB PM3 具备高速通信能力以及优化算法，可对具有大存储器的器件进行快速编程。它还包含了 MMC 卡，用于文件存储及数据应用。

26.11 演示 / 开发板、评估工具包及入门工具包

有许多演示、开发和评估板可用于各种 PIC MCU 和 dsPIC DSC，实现对全功能系统的快速应用开发。大多数的演示、开发和评估板都有实验布线区，供用户添加定制电路；还有应用固件和源代码，用于检查和修改。

这些板支持多种功能部件，包括 LED、温度传感器、开关、扬声器、 RS-232 接口、 LCD 显示器、电位计和附加 EEPROM 存储器。

演示和开发板可用于教学环境，在实验布线区设计定制电路，从而掌握各种单片机应用。

除了 PICDEM™ 和 dsPICDEM™ 演示 / 开发板系列电路外，Microchip 还有一系列评估工具包和演示软件，适用于模拟滤波器设计、KEELOQ® 数据安全产品 IC、CAN、IrDA®、 PowerSmart 电池管理、 SEEVAL® 评估系统、Σ–Δ ADC、流速传感器，等等。

同时还提供入门工具包，其中包含体验指定器件功能所需的所有软硬件。通常提供单个应用以及调试功能，都包含在一块电路板上。

关于演示、开发和评估工具包的完整列表，请访问 Microchip 网站 （www.microchip.com）。

26.12 第三方开发工具

Microchip 还提供一些来自第三方供应商的优秀开发工具。这些工具均经过精心挑选，功能独特，物有所值。

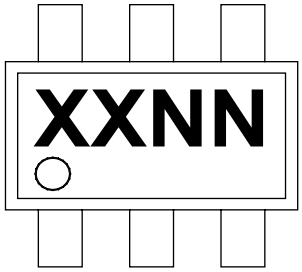
- SoftLog 和 CCS 等公司提供的器件编程器和量产编程器
- Gimpel 和 Trace Systems 等公司提供的软件工具
- Saleae 和 Total Phase 等公司提供的协议分析器
- MikroElektronika、Digilent® 和 Olimex 等公司提供的演示板
- EZ Web Lynx、WIZnet 和 IPLogika® 等公司提供的嵌入式以太网解决方案

© 2011-2014 Microchip Technology Inc.

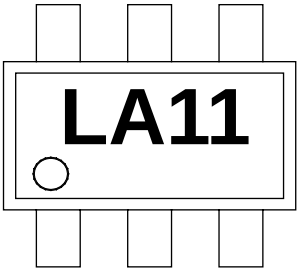
27.0 封装信息

27.1 封装标识信息

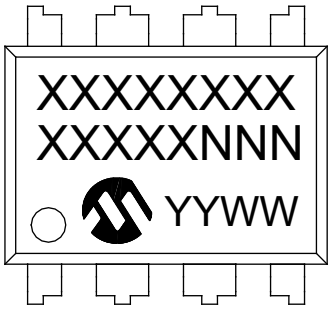
6 引脚 SOT-23



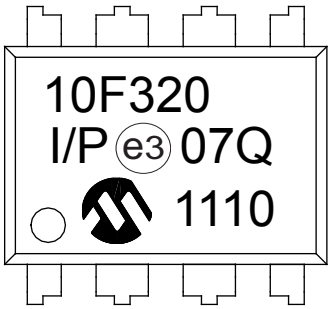
示例



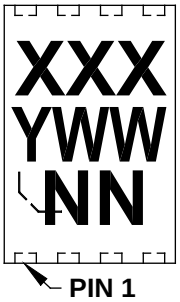
8 引脚 PDIP（300 mil）



示例



8 引脚 DFN（2x3x0.9 mm）



示例

图注：	XX...X	客户指定信息
	Y	年份代码 （日历年的最后一位数字）
	YY	年份代码 （日历年的最后两位数字）
	WW	星期代码 （一月一日的星期代码为 “01”）
	NNN	以字母数字排序的追踪代码
	e3	雾锡 （Matte Tin， Sn）的 JEDEC 无铅标志
	*	表示无铅封装。JEDEC 无铅标志（e3）标示于此种封装的外包装上。

注： Microchip 器件编号如果无法在同一行内完整标注，将换行标出，因此会限制表示客户信息的字符数。

表 **27-1：** **8** 引脚 **2x3 DFN** （**MC**）顶部标识

器件编号	标识
PIC10F322(T)-I/MC	BAA
PIC10F322(T)-E/MC	BAB
PIC10F320(T)-I/MC	BAC
PIC10F320(T)-E/MC	BAD
PIC10LF322(T)-I/MC	BAF
PIC10LF322(T)-E/MC	BAG
PIC10LF320(T)-I/MC	BAH
PIC10LF320(T)-E/MC	BAJ

表 **27-2：** **6** 引脚 **SOT-23** （**OT**）封装顶部标识

器件编号	标识
PIC10F322(T)-I/OT	LA/LJ
PIC10F322(T)-E/OT	LB/LK
PIC10F320(T)-I/OT	LC
PIC10F320(T)-E/OT	LD
PIC10LF322(T)-I/OT	LE
PIC10LF322(T)-E/OT	LF
PIC10LF320(T)-I/OT	LG
PIC10LF320(T)-E/OT	LH

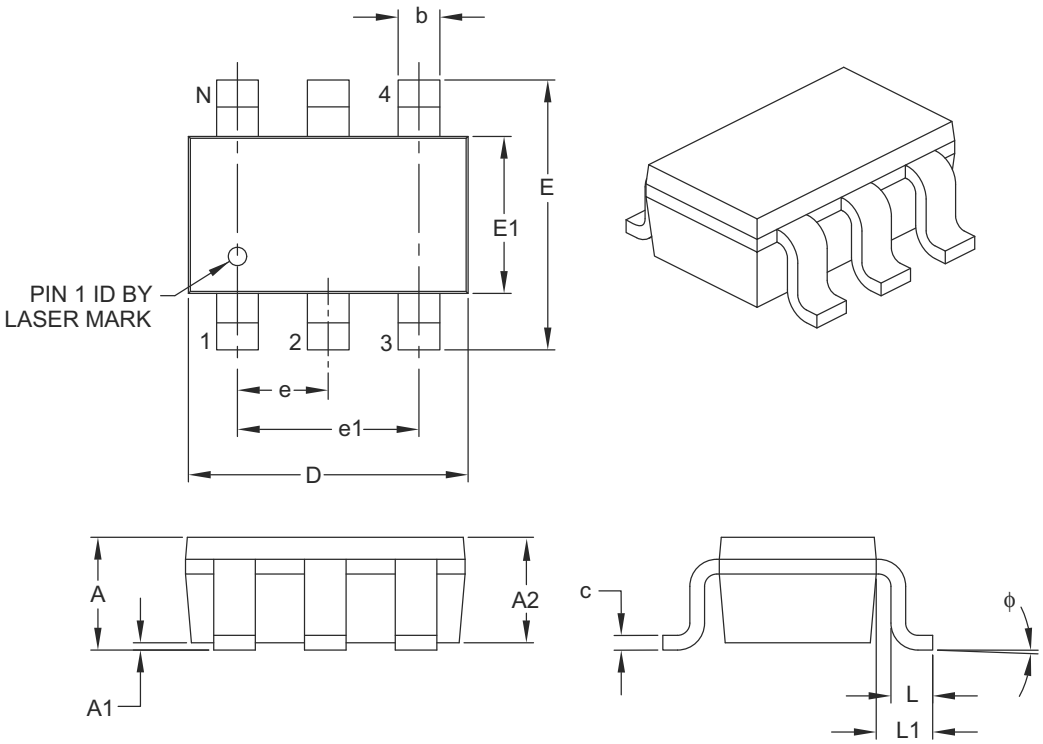
© 2011-2014 Microchip Technology Inc.

27.2 封装详细信息

以下部分将介绍各种封装的技术细节。

6 引脚塑封小外形晶体管封装 （OT）[SOT-23]

注：	最新封装图请至 http://www.microchip.com/packaging 查看 Microchip 封装规范。



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	6		
Pitch	e	0.95 BSC		
Outside Lead Pitch	e1	1.90 BSC		
Overall Height	A	0.90	–	1.45
Molded Package Thickness	A2	0.89	–	1.30
Standoff	A1	0.00	–	0.15
Overall Width	E	2.20	–	3.20
Molded Package Width	E1	1.30	–	1.80
Overall Length	D	2.70	–	3.10
Foot Length	L	0.10	–	0.60
Footprint	L1	0.35	–	0.80
Foot Angle	ϕ	0°	–	30°
Lead Thickness	c	0.08	–	0.26
Lead Width	b	0.20	–	0.51

Notes:

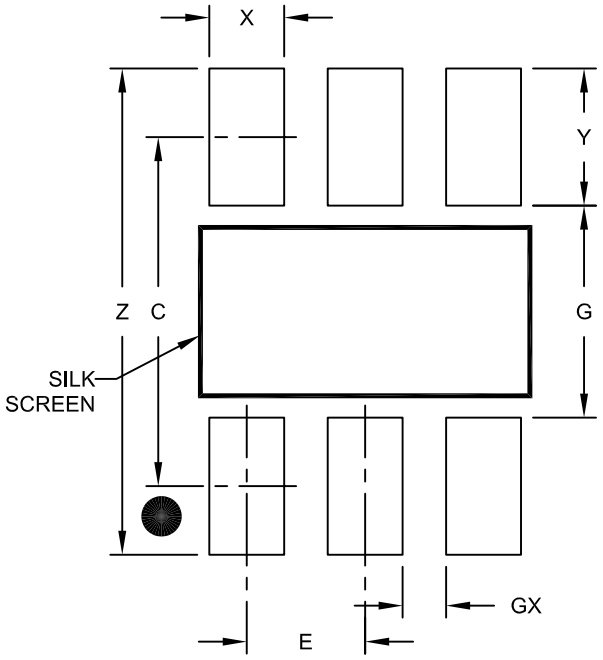
1. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.127 mm per side.
2. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-028B

6 引脚塑封小外形晶体管封装 （OT）[SOT-23]

注： 最新封装图请至 http://www.microchip.com/packaging 查看 Microchip 封装规范。



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.95 BSC		
Contact Pad Spacing	C		2.80	
Contact Pad Width (X6)	X			0.60
Contact Pad Length (X6)	Y			1.10
Distance Between Pads	G	1.70		
Distance Between Pads	GX	0.35		
Overall Width	Z			3.90

Notes:

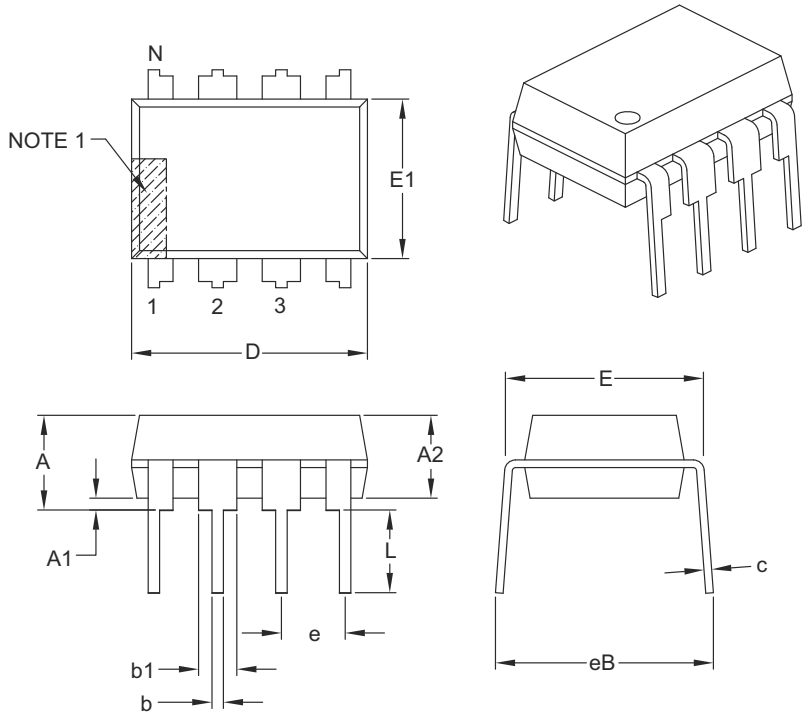
1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2028A

© 2011-2014 Microchip Technology Inc.

8 引脚塑封双列直插式封装 （P） —— 主体 300 mil [PDIP]

注： 最新封装图请至 http://www.microchip.com/packaging 查看 Microchip 封装规范。



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

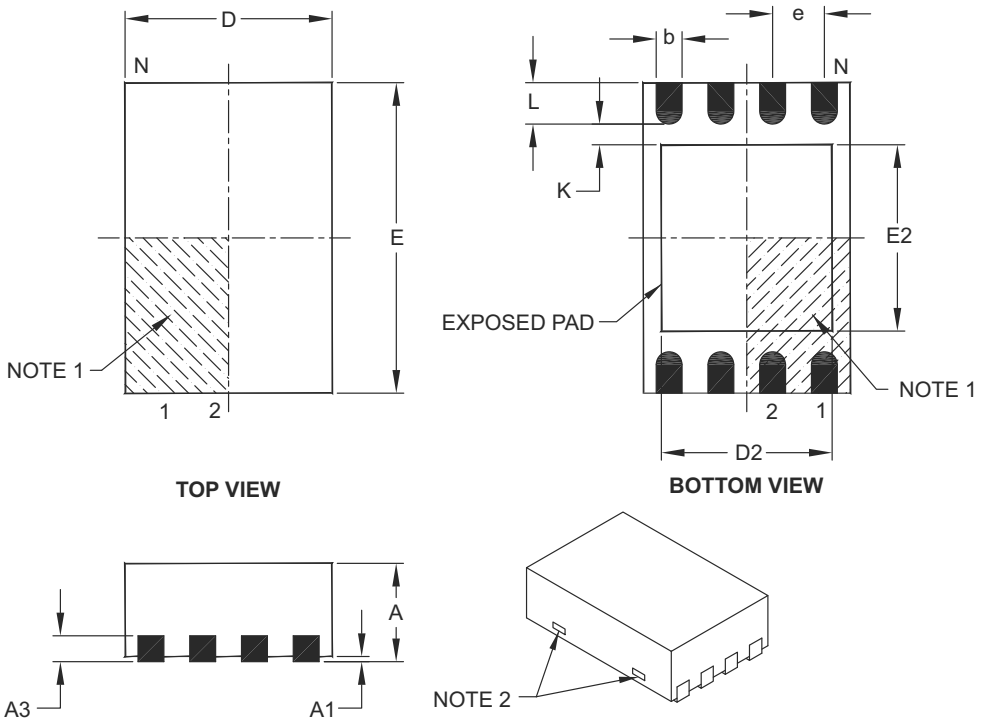
Notes:

1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-018B

8 引脚塑封双列扁平无脚封装 （MC） —— 主体 2x3x0.9 mm [DFN]

注： 最新封装图请至 http://www.microchip.com/packaging 查看 Microchip 封装规范。



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	8		
Pitch	e	0.50 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Length	D	2.00 BSC		
Overall Width	E	3.00 BSC		
Exposed Pad Length	D2	1.30	–	1.55
Exposed Pad Width	E2	1.50	–	1.75
Contact Width	b	0.20	0.25	0.30
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	–	–

Notes:

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

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

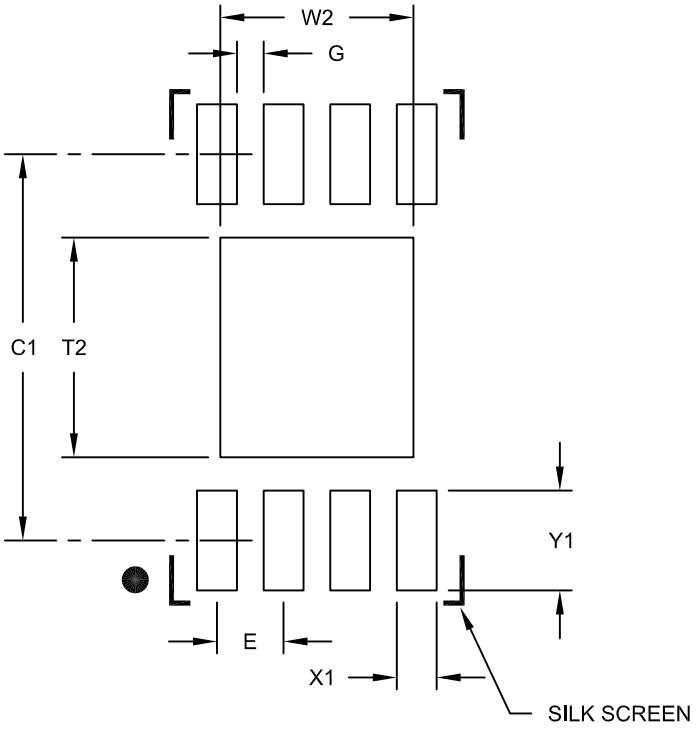
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-123C

© 2011-2014 Microchip Technology Inc.

8 引脚塑封双列扁平无脚封装 （MC） —— 主体 2x3x0.9 mm [DFN]

注： 最新封装图请至 http://www.microchip.com/packaging 查看 Microchip 封装规范。



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Optional Center Pad Width	W2			1.45
Optional Center Pad Length	T2			1.75
Contact Pad Spacing	C1		2.90	
Contact Pad Width (X8)	X1			0.30
Contact Pad Length (X8)	Y1			0.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2123B

附录 A： 数据手册版本历史

版本 A （2011 年 7 月）

本文档的初始版本。

版本 B （2014 年 2 月）

更新了电气规范，新增了表格，并且进行了一些细微的修改。

© 2011-2014 Microchip Technology Inc.

MICROCHIP 网站

Microchip 网站 （www.microchip.com）为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的互联浏览器即可访问。网站提供以下信息：

- **产品支持** —— 数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持** —— 常见问题解答（FAQ）、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务** —— 产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请登录 Microchip 网站 www.microchip.com。在“支持”（Support）下，点击“变更通知客户”（Customer Change Notification）服务后按照注册说明完成注册。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 （FAE）
- 技术支持

客户应联系其代理商、代表或应用工程师 （FAE）寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 http://microchip.com/support 获得网上技术支持。

产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

部件编号	[X](1)	-	X	/XX	XXX
器件	卷带式选项		温度范围	封装	定制编号

器件：	PIC10F320、 PIC10LF320、 PIC10F322 和 PIC10LF322	
卷带式选项：	空白	= 标准封装 （料管或托盘）
	T	= 卷带式 (1)
温度范围：	I	= -40°C 至 +85°C（工业级）
	E	= -40°C 至 +125°C（扩展级）
封装：	OT	= SOT-23
	P	= PDIP
	MC	= DFN
定制编号：	QTP、 SQTP、代码或特殊要求 （其它情况均为空白）	

示例：

a) PIC10LF320T - I/OT
卷带式，
工业级温度，
SOT-23 封装

b) PIC10F322 - I/P
工业级温度，
PDIP 封装

c) PIC10F322 - E/MC
扩展级温度，
DFN 封装

注 1： 卷带式标识符仅出现在目录元器件编号说明中，此标识符用于订购用途，不会印刷于器件封装上。请与当地的销售办事处联系，确认是否提供卷带式包装。

© 2011-2014 Microchip Technology Inc.

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下， Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是 “牢不可破”的。

代码保护功能处于持续发展中。 Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了 《数字器件千年版权法案 （Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。**Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。** Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和 / 或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。在 Microchip 知识产权保护下，不得暗中或以其他方式转让任何许可证。

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

商标

Microchip 的名称和徽标组合、 Microchip 徽标、 dsPIC、 FlashFlex、 KEELOQ、 KEELOQ 徽标、 MPLAB、 PIC、 PICmicro、 PICSTART、 PIC32 徽标、 rfPIC、 SST、 SST 徽标、 SuperFlash 和 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

FilterLab、 Hampshire、 HI-TECH C、 Linear Active Thermistor、 MTP、 SEEVAL 和 The Embedded Control Solutions Company 均为 Microchip Technology Inc. 在美国的注册商标。

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

Analog-for-the-Digital Age、 Application Maestro、 BodyCom、 chipKIT、 chipKIT 徽标、 CodeGuard、 dsPICDEM、 dsPICDEM.net、 dsPICworks、 dsSPEAK、 ECAN、 ECONOMONITOR、 FanSense、 HI-TIDE、 In-Circuit Serial Programming、 ICSP、 Mindi、 MiWi、 MPASM、 MPF、 MPLAB Certified 徽标、 MPLIB、 MPLINK、 mTouch、 Omniscient Code Generation、 PICC、 PICC-18、 PICDEM、 PICDEM.net、 PICkit、 PICtail、 REAL ICE、 rfLAB、 Select Mode、 SQI、 Serial Quad I/O、 Total Endurance、 TSHARC、 UniWinDriver、 WiperLock、 ZENA 和 Z-Scale 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Inc. 在美国的服务标记。

GestIC 和 ULPP 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2014, Microchip Technology Inc. 版权所有。

ISBN：978-1-63276-086-9

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。 Microchip 的 PIC® MCU 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外， Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

© 2011-2014 Microchip Technology Inc.

全球销售及服务网点

美洲

公司总部 Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277
技术支持：
http://www.microchip.com/support
网址：www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

奥斯汀 Austin, TX
Tel: 1-512-257-3370

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

克里夫兰 Cleveland
Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Novi, MI
Tel: 1-248-848-4000

休斯敦 Houston, TX
Tel: 1-281-894-5983

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

纽约 New York, NY
Tel: 1-631-435-6000

圣何塞 San Jose, CA
Tel: 1-408-735-9110

加拿大多伦多 Toronto
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 北京
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

中国 - 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重庆
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 杭州
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

中国 - 香港特别行政区
Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

中国 - 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦门
Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 - 珠海
Tel: 86-756-321-0040
Fax: 86-756-321-0049

亚太地区

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

台湾地区 - 新竹
Tel: 886-3-5778-366
Fax: 886-3-5770-955

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

印度 India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 India - Pune
Tel: 91-20-3019-1500

日本 Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

日本 Japan - Tokyo
Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

韩国 Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

韩国 Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 Denmark-Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Dusseldorf
Tel: 49-2129-3766400

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

德国 Germany - Pforzheim
Tel: 49-7231-424750

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

意大利 Italy - Venice
Tel: 39-049-7625286

荷兰 Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

波兰 Poland - Warsaw
Tel: 48-22-3325737

西班牙 Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

瑞典 Sweden - Stockholm
Tel: 46-8-5090-4654

英国 UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

03/25/14

© 2011-2014 Microchip Technology Inc.