PIC16F610/16HV610
PIC16F616/16HV616
数据手册

14 引脚 8 位
CMOS 闪存单片机

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。

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip 位于美国亚利桑那州 Chandler 和 Tempe、位于俄勒冈州 Gresham 及位于加利福尼亚州 Mountain View 的全球总部、设计中心和晶圆生产厂均通过了 ISO/TS-16949:2002 认证。公司在 PIC® 单片机与 dsPIC® 数字信号控制器、 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方面的质量体系流程均符合 ISO/TS-16949:2002。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。

PIC16F610/616/16HV610/616

14 引脚 8 位 CMOS 闪存单片机

高性能 RISC CPU：

- 仅需学习 35 条指令：
  - 除了跳转指令以外，所有指令都是单周期的
- 工作速度：
  - DC － 20 MHz 振荡器 / 时钟输入
  - DC － 200 ns 指令周期
- 中断能力
- 8 层深硬件堆栈
- 直接、间接和相对寻址模式

单片机特性：

- 精确的内部振荡器：
  - 出厂时精度已校准到 ±1%，典型值
  - 用户可选择频率范围：4 MHz 或 8 MHz
- 省电休眠模式
- 电压范围：
  - PIC16F610/616：2.0V 到 5.5V
  - PIC16HV610/616：2.0V 到用户定义的最大值 （见 "注"）
- 工业级和扩展级温度范围
- 上电复位 （Power-on Reset， POR）
- 上电延时定时器 （Power-up Timer， PWRT）和振荡器起振定时器 （Oscillator Start-up Timer，OST）
- 欠压复位 （Brown-out Reset， BOR）
- 带独立振荡器的看门狗定时器 （Watchdog Timer， WDT）以保证可靠运行
- 带弱上拉的主复位引脚，可复用为输入引脚
- 可编程代码保护
- 高耐用性闪存：
  - 闪存耐写次数达 100,000 次
  - 闪存数据保存期：> 40 年

低功耗特性：

- 待机电流：
  - 2.0V 时典型值为 50 nA
- 工作电流：
  - 32 kHz、 2.0V 时典型值为 20 μA
  - 4 MHz、 2.0V 时典型值为 220 μA
- 看门狗定时器电流：
  - 2.0V 时典型值为 1 μA

外设特性：

- 并联稳压器 （仅限 PIC16HV610/616）：
  - 5 伏稳压
  - 4 mA 到 50 mA 并联电流范围
- 11 个 I/O 引脚和 1 个只能用作输入的引脚
  - 高灌 / 拉电流能力，可直接驱动 LED
  - 电平变化中断引脚
  - 可单独编程的弱上拉
- 模拟比较器模块，具有：
  - 2 个模拟比较器
  - 片上可编程参考电压 （CVREF）模块 （VDD 的 %）
  - 固定参考电压
  - 可从外部访问比较器输入和输出
  - SR 锁存器
  - 内建滞后 （用户可选择）
- Timer0：带有 8 位可编程预分频器的 8 位定时器 / 计数器
- 增强型 Timer1：
  - 带有预分频器的 16 位定时器 / 计数器
  - 外部 Timer1 门控 （计数使能）
  - 如果选用 INTOSC 模式，可选择 OSC1 和 OSC2 之间的振荡器 （工作在 LP 模式下）作为 Timer1 的振荡器
  - Timer1 振荡器
- 通过两个引脚进行在线串行编程 （In-Circuit Serial Programming™， ICSP™）

仅限 PIC16F616/16HV616：

- A/D 转换器：
  - 10 位精度
  - 8 路外部输入通道
  - 2 路内部参考通道
- Timer2：带有 8 位周期寄存器、预分频器和后分频器的 8 位定时器 / 计数器
- 增强型捕捉、比较和 PWM 模块：
  - 16 位捕捉，最大精度 12.5 ns
  - 16 位比较，最大精度 200 ns
  - 10 位 PWM，带有 1、 2 或 4 路输出通道及可编程 "死区"，最高频率 20 kHz

注： 通过内部并联稳压器的电压不能超过 5V。

<table>
<tr><th rowspan="2">器件</th><th>程序存储器</th><th>数据存储器</th><th rowspan="2">I/O</th><th rowspan="2">10 位 A/D
（通道）</th><th rowspan="2">比较器</th><th rowspan="2">8/16 位
定时器</th><th rowspan="2">电压范围</th></tr>
<tr><th>闪存 （字）</th><th>SRAM （字节）</th></tr>
<tr><td>PIC16F610</td><td>1024</td><td>64</td><td>11</td><td>—</td><td>2</td><td>1/1</td><td>2.0-5.5V</td></tr>
<tr><td>PIC16HV610</td><td>1024</td><td>64</td><td>11</td><td>—</td><td>2</td><td>1/1</td><td>2.0- 用户定义</td></tr>
<tr><td>PIC16F616</td><td>2048</td><td>128</td><td>11</td><td>8</td><td>2</td><td>2/1</td><td>2.0-5.5V</td></tr>
<tr><td>PIC16HV616</td><td>2048</td><td>128</td><td>11</td><td>8</td><td>2</td><td>2/1</td><td>2.0- 用户定义</td></tr>
</table>

PIC16F610/16HV610 14 引脚示意图 （PDIP、 SOIC 和 TSSOP）

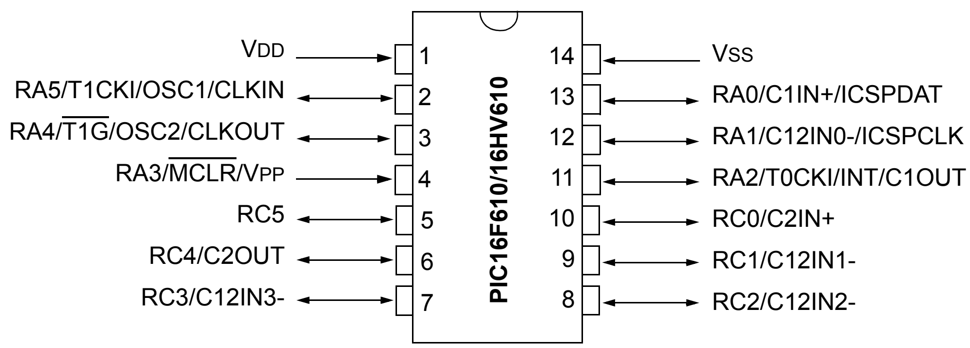


表 1： PIC16F610/16HV610 14 引脚汇总

I/O	引脚	比较器	定时器	中断	上拉	基本
RA0	13	C1IN+	—	IOC	有	ICSPDAT
RA1	12	C12IN0-	—	IOC	有	ICSPCLK
RA2	11	C1OUT	T0CKI	INT/IOC	有	—
RA3(1)	4	—	—	IOC	有 (2)	$\overline{\text{MCLR}}$/VPP
RA4	3	—	$\overline{\text{T1G}}$	IOC	有	OSC2/CLKOUT
RA5	2	—	T1CKI	IOC	有	OSC1/CLKIN
RC0	10	C2IN+	—	—	—	—
RC1	9	C12IN1-	—	—	—	—
RC2	8	C12IN2-	—	—	—	—
RC3	7	C12IN3-	—	—	—	—
RC4	6	C2OUT	—	—	—	—
RC5	5	—	—	—	—	—
—	1	—	—	—	—	VDD
—	14	—	—	—	—	VSS

注 1： 仅用作输入。
2： 仅当引脚被配置为外部 $\overline{\text{MCLR}}$ 时。

PIC16F616/16HV616 14 引脚示意图 （PDIP、 SOIC 和 TSSOP）

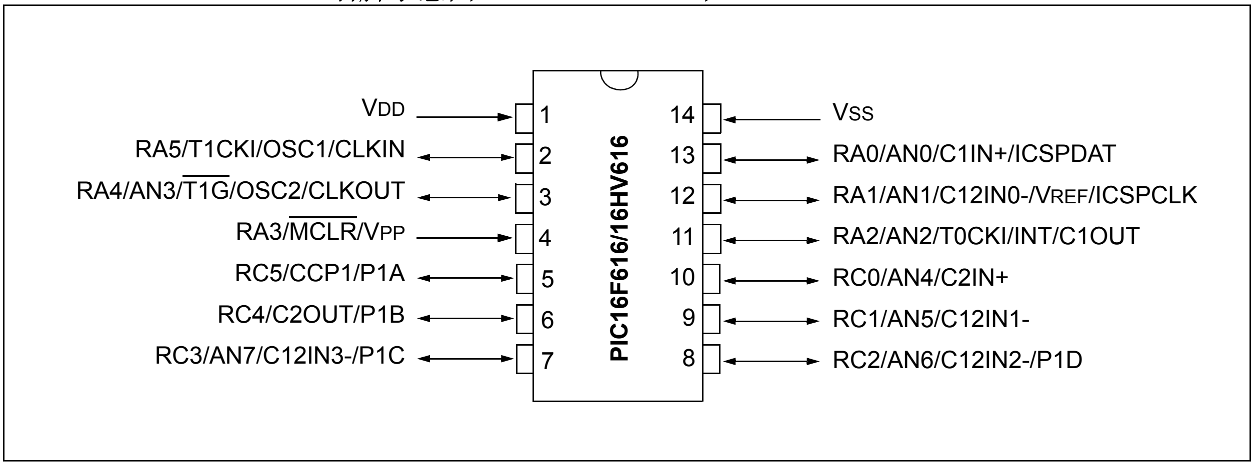


表 2： PIC16F616/16HV616 14 引脚汇总

I/O	引脚	模拟	比较器	定时器	CCP	中断	上拉	基本
RA0	13	AN0	C1IN+	—	—	IOC	有	ICSPDAT
RA1	12	AN1/VREF	C12IN0-	—	—	IOC	有	ICSPCLK
RA2	11	AN2	C1OUT	T0CKI	—	INT/IOC	有	—
RA3[(1)]	4	—	—	—	—	IOC	有 [(2)]	$\overline{\text{MCLR}}$/VPP
RA4	3	AN3	—	$\overline{\text{T1G}}$	—	IOC	有	OSC2/CLKOUT
RA5	2	—	—	T1CKI	—	IOC	有	OSC1/CLKIN
RC0	10	AN4	C2IN+	—	—	—	—	—
RC1	9	AN5	C12IN1-	—	—	—	—	—
RC2	8	AN6	C12IN2-	—	P1D	—	—	—
RC3	7	AN7	C12IN3-	—	P1C	—	—	—
RC4	6	—	C2OUT	—	P1B	—	—	—
RC5	5	—	—	—	CCP1/P1A	—	—	—
—	1	—	—	—	—	—	—	VDD
—	14	—	—	—	—	—	—	VSS

注 **1：** 仅用作输入。
2： 仅当引脚被配置为外部 $\overline{\text{MCLR}}$ 时。

PIC16F610/16HV610 16 引脚示意图 （QFN）

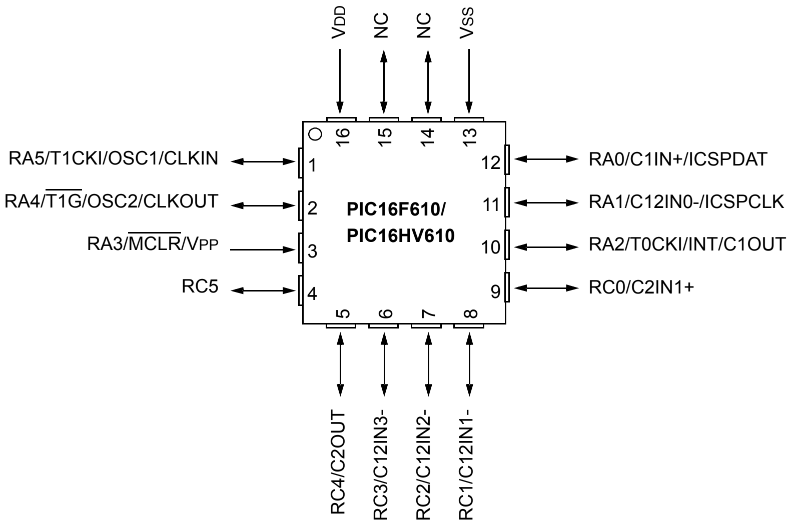


表 3： PIC16F610/16HV610 16 引脚汇总

I/O	引脚	比较器	定时器	中断	上拉	基本
RA0	12	C1IN+	—	IOC	有	ICSPDAT
RA1	11	C12IN0-	—	IOC	有	ICSPCLK
RA2	10	C1OUT	T0CKI	INT/IOC	有	—
RA3(1)	3	—	—	IOC	有 (2)	$\overline{\text{MCLR}}$/VPP
RA4	2	—	$\overline{\text{T1G}}$	IOC	有	OSC2/CLKOUT
RA5	1	—	T1CKI	IOC	有	OSC1/CLKIN
RC0	9	C2IN+	—	—	—	—
RC1	8	C12IN1-	—	—	—	—
RC2	7	C12IN2-	—	—	—	—
RC3	6	C12IN3-	—	—	—	—
RC4	5	C2OUT	—	—	—	—
RC5	4	—	—	—	—	—
—	16	—	—	—	—	VDD
—	13	—	—	—	—	VSS

注 1： 仅用作输入。

2： 仅当引脚被配置为外部 $\overline{\text{MCLR}}$ 时。

PIC16F616/16HV616 16 引脚示意图 （QFN）

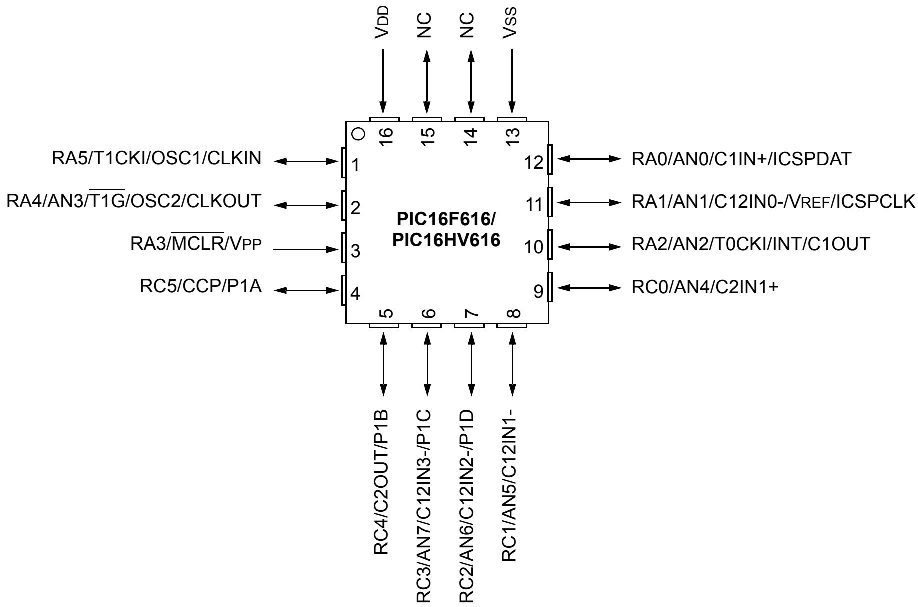


表 4： PIC16F616/16HV616 16 引脚汇总

I/O	引脚	模拟	比较器	定时器	CCP	中断	上拉	基本
RA0	12	AN0	C1IN+	—	—	IOC	有	ICSPDAT
RA1	11	AN1/VREF	C12IN0-	—	—	IOC	有	ICSPCLK
RA2	10	AN2	C1OUT	T0CKI	—	INT/IOC	有	—
RA3(1)	3	—	—	—	—	IOC	有 (2)	MCLR/VPP
RA4	2	AN3	—	T1G	—	IOC	有	OSC2/CLKOUT
RA5	1	—	—	T1CKI	—	IOC	有	OSC1/CLKIN
RC0	9	AN4	C2IN+	—	—	—	—	—
RC1	8	AN5	C12IN1-	—	—	—	—	—
RC2	7	AN6	C12IN2-	—	P1D	—	—	—
RC3	6	AN7	C12IN3-	—	P1C	—	—	—
RC4	5	—	C2OUT	—	P1B	—	—	—
RC5	4	—	—	—	CCP1/P1A	—	—	—
—	16	—	—	—	—	—	—	VDD
—	13	—	—	—	—	—	—	VSS

注 1： 仅用作输入。
2： 仅当引脚被配置为外部 MCLR 时。

目录

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要求。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 **docerrors@microchip.com**，或将本数据手册后附的 **《读者反馈表》**传真到 86-21-5407 5066。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，请查询我公司的网站：

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。文献编号中数字串后的字母是版本号，例如：DS30000A 是 DS30000 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 **www.microchip.com** 上注册。

1.0 器件概述

本数据手册涵盖 PIC16F610/616/16HV610/616 器件。这些器件采用 14 引脚 PDIP、SOIC、TSSOP 和 16 引脚 QFN 封装。

这些器件的框图和引脚说明如下所示：

- PIC16F610/16HV610 （图 1-1，表 1-1）
- PIC16F616/16HV616 （图 1-2，表 1-2）

图 1-1： PIC16F610/16HV610 框图

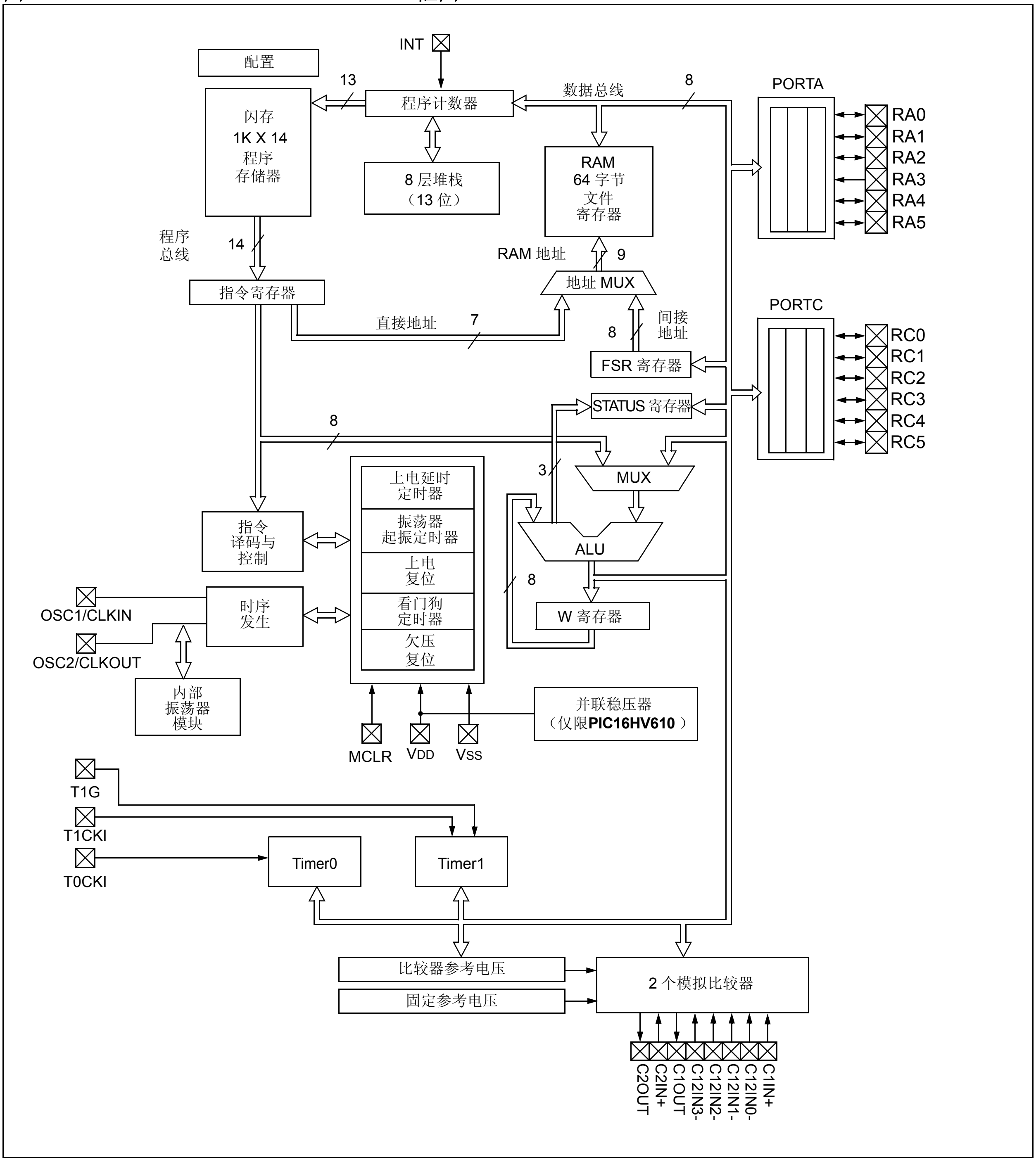
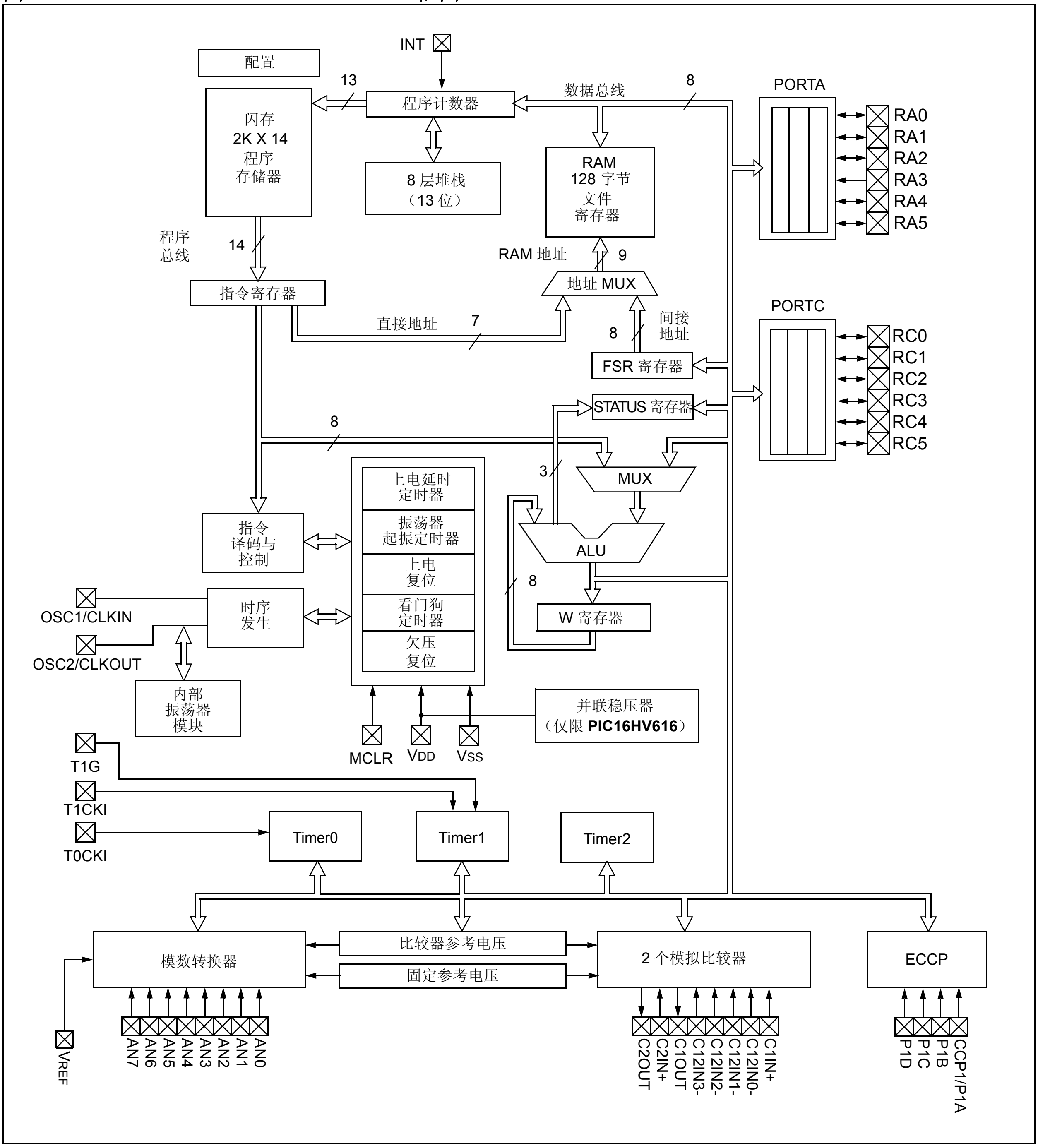


图 1-2： PIC16F616/16HV616 框图



 初稿

表 1-1： PIC16F610/16HV610 引脚说明

名称	功能	输入类型	输出类型	说明
RA0/C1IN+/ICSPDAT	RA0	TTL	CMOS	具有可编程上拉和电平变化中断功能的 PORTA I/O
	C1IN+	AN	—	比较器 C1 的同相输入
	ICSPDAT	ST	CMOS	串行编程数据 I/O
RA1/C12IN0-/ICSPCLK	RA1	TTL	CMOS	具有可编程上拉和电平变化中断功能的 PORTA I/O
	C12IN0-	AN	—	比较器 C1 和 C2 的反相输入
	ICSPCLK	ST	—	串行编程时钟
RA2/T0CKI/INT/C1OUT	RA2	ST	CMOS	具有可编程上拉和电平变化中断功能的 PORTA I/O
	T0CKI	ST	—	Timer0 的时钟输入
	INT	ST	—	外部中断
	C1OUT	—	CMOS	比较器 C1 的输出
RA3/$\overline{\text{MCLR}}$/VPP	RA3	TTL	—	具有电平变化中断功能的 PORTA 输入
	$\overline{\text{MCLR}}$	ST	—	带内部上拉的主复位
	VPP	HV	—	编程电压
RA4/$\overline{\text{T1G}}$/OSC2/CLKOUT	RA4	TTL	CMOS	具有可编程上拉和电平变化中断功能的 PORTA I/O
	$\overline{\text{T1G}}$	ST	—	Timer1 门控 （计数使能）
	OSC2	—	XTAL	晶振 / 谐振器
	CLKOUT	—	CMOS	FOSC/4 输出
RA5/T1CKI/OSC1/CLKIN	RA5	TTL	CMOS	具有可编程上拉和电平变化中断功能的 PORTA I/O
	T1CKI	ST	—	Timer1 的时钟输入
	OSC1	XTAL	—	晶振 / 谐振器
	CLKIN	ST	—	外部时钟输入 /RC 振荡器连接
RC0/C2IN+	RC0	TTL	CMOS	PORTC I/O
	C2IN+	AN	—	比较器 C2 的同相输入
RC1/C12IN1-	RC1	TTL	CMOS	PORTC I/O
	C12IN1-	AN	—	比较器 C1 和 C2 的反相输入
RC2/C12IN2-	RC2	TTL	CMOS	PORTC I/O
	C12IN2-	AN	—	比较器 C1 和 C2 的反相输入
RC3/C12IN3-	RC3	TTL	CMOS	PORTC I/O
	C12IN3-	AN	—	比较器 C1 和 C2 的反相输入
RC4/C2OUT	RC4	TTL	CMOS	PORTC I/O
	C2OUT	—	CMOS	比较器 C2 的输出
RC5	RC5	TTL	CMOS	PORTC I/O
VDD	VDD	电源	—	正电源
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　HV = 高电压
ST = CMOS 电平的施密特触发器输入　TTL = TTL 兼容输入　XTAL = 晶振

表 1-2： PIC16F616/16HV616 引脚说明

名称	功能	输入类型	输出类型	说明
RA0/AN0/C1IN+/ICSPDAT	RA0	TTL	CMOS	具有可编程上拉和电平变化中断功能的 PORTA I/O
	AN0	AN	—	A/D 通道 0 的输入
	C1IN+	AN	—	比较器 C1 的同相输入
	ICSPDAT	ST	CMOS	串行编程数据 I/O
RA1/AN1/C12IN0-/VREF/ICSPCLK	RA1	TTL	CMOS	具有可编程上拉和电平变化中断功能的 PORTA I/O
	AN1	AN	—	A/D 通道 1 的输入
	C12IN0-	AN	—	比较器 C1 和 C2 的反相输入
	VREF	AN	—	A/D 的外部参考电压
	ICSPCLK	ST	—	串行编程时钟
RA2/AN2/T0CKI/INT/C1OUT	RA2	ST	CMOS	具有可编程上拉和电平变化中断功能的 PORTA I/O
	AN2	AN	—	A/D 通道 2 的输入
	T0CKI	ST	—	Timer0 的时钟输入
	INT	ST	—	外部中断
	C1OUT	—	CMOS	比较器 C1 的输出
RA3/$\overline{\text{MCLR}}$/VPP	RA3	TTL	—	具有电平变化中断功能的 PORTA 输入
	$\overline{\text{MCLR}}$	ST	—	带内部上拉的主复位
	VPP	HV	—	编程电压
RA4/AN3/$\overline{\text{T1G}}$/OSC2/CLKOUT	RA4	TTL	CMOS	具有可编程上拉和电平变化中断功能的 PORTA I/O
	AN3	AN	—	A/D 通道 3 的输入
	$\overline{\text{T1G}}$	ST	—	Timer1 门控 （计数使能）
	OSC2	—	XTAL	晶振 / 谐振器
	CLKOUT	—	CMOS	FOSC/4 输出
RA5/T1CKI/OSC1/CLKIN	RA5	TTL	CMOS	具有可编程上拉和电平变化中断功能的 PORTA I/O
	T1CKI	ST	—	Timer1 的时钟输入
	OSC1	XTAL	—	晶振 / 谐振器
	CLKIN	ST	—	外部时钟输入 /RC 振荡器连接
RC0/AN4/C2IN+	RC0	TTL	CMOS	PORTC I/O
	AN4	AN	—	A/D 通道 4 的输入
	C2IN+	AN	—	比较器 C2 的同相输入
RC1/AN5/C12IN1-	RC1	TTL	CMOS	PORTC I/O
	AN5	AN	—	A/D 通道 5 的输入
	C12IN1-	AN	—	比较器 C1 和 C2 的反相输入
RC2/AN6/C12IN2-/P1D	RC2	TTL	CMOS	PORTC I/O
	AN6	AN	—	A/D 通道 6 的输入
	C12IN2-	AN	—	比较器 C1 和 C2 的反相输入
	P1D	—	CMOS	PWM 输出
RC3/AN7/C12IN3-/P1C	RC3	TTL	CMOS	PORTC I/O
	AN7	AN	—	A/D 通道 7 的输入
	C12IN3-	AN	—	比较器 C1 和 C2 的反相输入
	P1C	—	CMOS	PWM 输出
RC4/C2OUT/P1B	RC4	TTL	CMOS	PORTC I/O
	C2OUT	—	CMOS	比较器 C2 的输出
	P1B	—	CMOS	PWM 输出
RC5/CCP1/P1A	RC5	TTL	CMOS	PORTC I/O
	CCP1	ST	CMOS	捕捉输入 / 比较输出
	P1A	—	CMOS	PWM 输出
VDD	VDD	电源	—	正电源
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　　HV = 高电压
ST = CMOS 电平的施密特触发器输入　　TTL = TTL 兼容输入　　XTAL = 晶振

2.0 存储器构成

2.1 程序存储器构成

PIC16F610/616/16HV610/616 器件具有一个 13 位程序计数器，能够寻址 8k x 14 的程序存储空间。只有 PIC16F610/16HV610 中的前 1k x 14 （0000h-3FF）空间和 PIC16F616/16HV616 中的前 2k x 14 （0000h-07FFh）空间是物理实现的。访问超出上述范围的存储单元，将回到前 1k x 14 空间内（PIC16F610/16HV610）和前 2K x 14 空间内 （PIC16F616/16HV616）。复位向量位于 0000h，而中断向量位于 0004h （见图 2-1）。

图 2-1： PIC16F610/16HV610 的程序存储器映射和堆栈

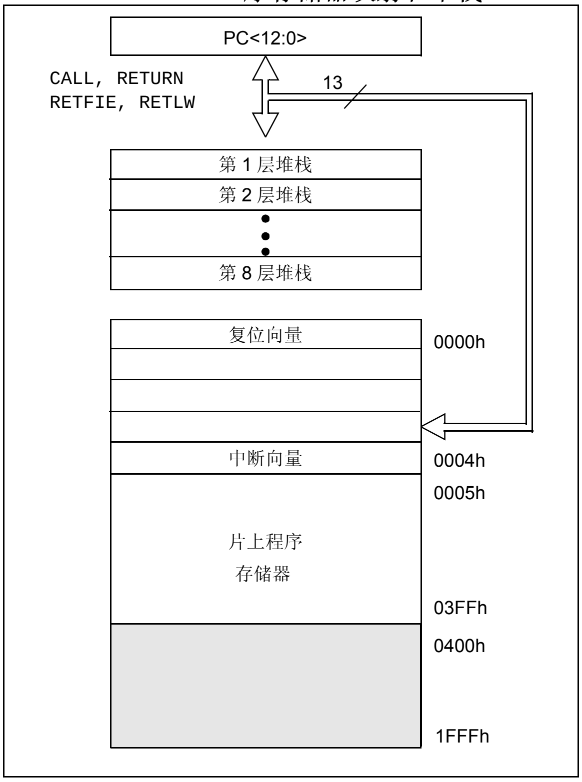
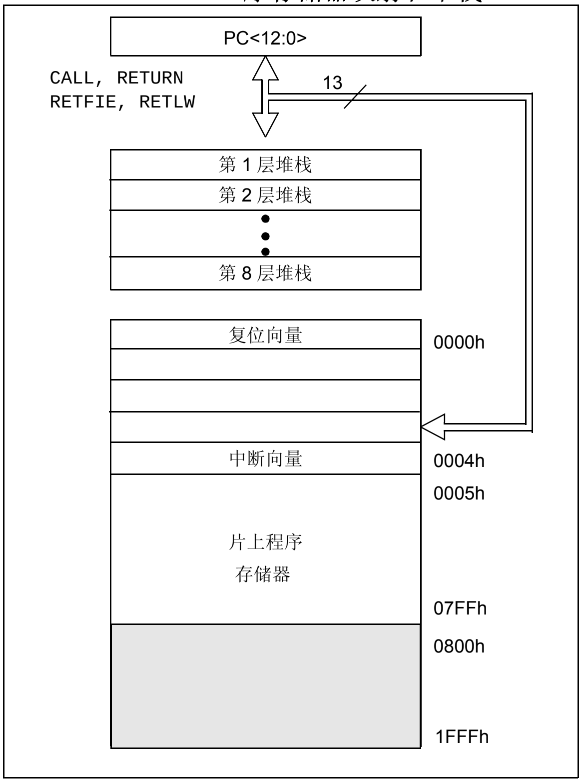


图 2-2： PIC16F616/16HV616 的程序存储器映射和堆栈



2.2 数据存储器构成

数据存储器 （见图 2-4）被划分为两个存储区，每个存储区由通用寄存器（General Purpose Register，GPR）和特殊功能寄存器（Special Function Register，SFR）组成。特殊功能寄存器位于每个存储区的前 32 个单元中。 PIC16F610/16HV610 Bank 0 中的寄存器单元 40h-7Fh 是通用寄存器，以静态 RAM 的形式实现。PIC16F616/16HV616 Bank 0 中的 20h-7Fh 和 Bank 1 中的 A0h-BFh 寄存器单元是通用寄存器，以静态 RAM 的形式实现。 Bank 1 中的寄存器单元 F0h-FFh 指向 Bank 0 中的地址 70h-7Fh。所有其他 RAM 均未实现，读取时返回 0。STATUS 寄存器的 RP0 位是存储区选择位。

RP0

0 → 选择 Bank 0

1 → 选择 Bank 1

注：	STATUS 寄存器的 IRP 和 RP1 位保留，且应始终保持为 0。

2.2.1 通用寄存器文件

PIC16F610/16HV610 的寄存器文件组织为 64 x 8，而 PIC16F616/16HV616 的寄存器文件则组织为 128 x 8。通过文件选择寄存器 （File Select Register， FSR），可以直接或间接地访问每个寄存器（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

特殊功能寄存器为 CPU 和外设模块用来对器件所需操作进行控制的寄存器 （见表 2-1）。这些寄存器皆为静态 RAM。

特殊功能寄存器可分为两类，即：内核和外设。本节将介绍与 “内核” 相关的特殊功能寄存器。与外设模块相关的特殊功能寄存器将在相应的外设功能模块章节中介绍。

 初稿

图 2-3： **PIC16F610/16HV610 的数据存储器映射**

Bank 0	文件地址	Bank 1	文件地址
间接地址 (1)	00h	间接地址 (1)	80h
TMR0	01h	OPTION_REG	81h
PCL	02h	PCL	82h
STATUS	03h	STATUS	83h
FSR	04h	FSR	84h
PORTA	05h	TRISA	85h
	06h		86h
PORTC	07h	TRISC	87h
	08h		88h
	09h		89h
PCLATH	0Ah	PCLATH	8Ah
INTCON	0Bh	INTCON	8Bh
PIR1	0Ch	PIE1	8Ch
	0Dh		8Dh
TMR1L	0Eh	PCON	8Eh
TMR1H	0Fh		8Fh
T1CON	10h	OSCTUNE	90h
	11h	ANSEL	91h
	12h		92h
	13h		93h
	14h		94h
	15h	WPUA	95h
	16h	IOCA	96h
	17h		97h
	18h		98h
VRCON	19h	SRCON0	99h
CM1CON0	1Ah	SRCON1	9Ah
CM2CON0	1Bh		9Bh
CM2CON1	1Ch		9Ch
	1Dh		9Dh
	1Eh		9Eh
	1Fh		9Fh
	20h–3Fh		A0h–EFh
通用寄存器 64 字节	40h–7Fh	访问 70h-7Fh	F0h–FFh

未实现的数据存储单元，读为 0。

注 1： 不是实际存在的寄存器。

图 2-4： **PIC16F616/16HV616 的数据存储器映射**

Bank 0	文件地址	Bank 1	文件地址
间接地址 (1)	00h	间接地址 (1)	80h
TMR0	01h	OPTION_REG	81h
PCL	02h	PCL	82h
STATUS	03h	STATUS	83h
FSR	04h	FSR	84h
PORTA	05h	TRISA	85h
	06h		86h
PORTC	07h	TRISC	87h
	08h		88h
	09h		89h
PCLATH	0Ah	PCLATH	8Ah
INTCON	0Bh	INTCON	8Bh
PIR1	0Ch	PIE1	8Ch
	0Dh		8Dh
TMR1L	0Eh	PCON	8Eh
TMR1H	0Fh		8Fh
T1CON	10h	OSCTUNE	90h
TMR2	11h	ANSEL	91h
T2CON	12h	PR2	92h
CCPR1L	13h		93h
CCPR1H	14h		94h
CCP1CON	15h	WPUA	95h
PWM1CON	16h	IOCA	96h
ECCPAS	17h		97h
	18h		98h
VRCON	19h	SRCON0	99h
CM1CON0	1Ah	SRCON1	9Ah
CM2CON0	1Bh		9Bh
CM2CON1	1Ch		9Ch
	1Dh		9Dh
ADRESH	1Eh	ADRESL	9Eh
ADCON0	1Fh	ADCON1	9Fh
通用寄存器 96 字节	20h–6Fh	通用寄存器 32 字节	A0h–BFh
			C0h–EFh
	70h–7Fh	访问 70h-7Fh	F0h–FFh

未实现的数据存储单元，读为 0。

注 1： 不是实际存在的寄存器。

表 **2-1**： **PIC16F610/616/16HV610/616 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/BOR 时的值	所在页
Bank 0											
00h	INDF	通过用 FSR 的内容寻址这个存储单元来寻址数据存储器 （不是实际存在的寄存器）								xxxx xxxx	22, 114
01h	TMR0	Timer0 模块寄存器								xxxx xxxx	43, 114
02h	PCL	程序计数器 （PC）的低字节								0000 0000	22, 114
03h	STATUS	IRP[1]	RP1[1]	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	16, 114
04h	FSR	间接数据存储器地址指针								xxxx xxxx	22, 114
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--x0 x000	31, 114
06h	—	未实现								—	—
07h	PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx 00xx	40, 114
08h	—	未实现								—	—
09h	—	未实现								—	—
0Ah	PCLATH	—	—	—	程序计数器高 5 位的写缓冲器					---0 0000	22, 114
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	18, 114
0Ch	PIR1	—	ADIF[2]	CCP1IF[2]	C2IF	C1IF	—	TMR2IF[2]	TMR1IF	-000 0-00	20, 114
0Dh	—	未实现								—	—
0Eh	TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								xxxx xxxx	47, 114
0Fh	TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								xxxx xxxx	47, 114
10h	T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	50, 114
11h	TMR2[2]	Timer2 模块寄存器								0000 0000	53, 114
12h	T2CON[2]	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	54, 114
13h	CCPR1L[2]	捕捉 / 比较 /PWM 寄存器 1 的低字节								XXXX XXXX	84, 114
14h	CCPR1H[2]	捕捉 / 比较 /PWM 寄存器 1 的高字节								XXXX XXXX	84, 114
15h	CCP1CON[2]	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	83, 114
16h	PWM1CON[2]	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000	83, 114
17h	ECCPAS[2]	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000	100, 114
18h	—	未实现								—	—
19h	VRCON	C1VREN	C2VREN	VRR	FVREN	VR3	VR2	VR1	VR0	0000 0000	70, 114
1Ah	CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0	0000 -000	60, 114
1Bh	CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0	0000 -000	61, 114
1Ch	CM2CON1	MC1OUT	MC2OUT	—	T1ACS	C1HYS	C2HYS	T1GSS	C2SYNC	00-0 0010	63, 114
1Dh	—	未实现								—	—
1Eh	ADRESH[2]	左对齐 A/D 结果的高 8 位或右对齐 A/D 结果的高 2 位								xxxx xxxx	78, 114
1Fh	ADCON0[2]	ADFM	VCFG	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	0000 0000	76, 114

图注： – = 未实现位 （读为 0）， u = 不变， x = 未知， q = 值取决于具体条件，阴影 = 未实现

注 1： IRP 和 RP1 位保留，始终保持这两位清零。

2： 仅限 PIC16F616/16HV616。

表 2-2： PIC16F610/616/16HV610/616 特殊功能寄存器汇总， BANK 1

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所在页
Bank 1											
80h	INDF	通过用 FSR 的内容寻址这个存储单元来寻址数据存储器 （不是实际存在的寄存器）								xxxx xxxx	22, 114
81h	OPTION_REG	$\overline{RAPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	17, 114
82h	PCL	程序计数器 （PC）的低字节								0000 0000	22, 114
83h	STATUS	IRP[(1)]	RP1[(1)]	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	16, 114
84h	FSR	间接数据存储器地址指针								xxxx xxxx	22, 114
85h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	31, 114
86h	—	未实现								—	—
87h	TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	40, 114
88h	—	未实现								—	—
89h	—	未实现								—	—
8Ah	PCLATH	—	—	—	程序计数器高 5 位的写缓冲器					---0 0000	22, 114
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	18, 114
8Ch	PIE1	—	ADIE[(3)]	CCP1IE[(3)]	C2IE	C1IE	—	TMR2IE[(3)]	TMR1IE	-000 0-00	19, 114
8Dh	—	未实现								—	—
8Eh	PCON	—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$	---- --qq	21, 114
8Fh	—	未实现								—	—
90h	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	29, 114
91h	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3[(3)]	ANS2[(3)]	ANS1	ANS0	1111 1111	32, 114
92h	PR2[(3)]	Timer2 模块周期寄存器								1111 1111	53, 114
93h	—	未实现								—	—
94h	—	未实现								—	—
95h	WPUA	—	—	WPUA5	WPUA4	—	WPUA2	WPUA1	WPUA0	--11 -111	33, 114
96h	IOCA	—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	--00 0000	33, 115
97h	—	未实现								—	—
98h	—	未实现								—	—
99h	SRCON0	SR1	SR0	C1SEN	C2REN	PULSS	PULSR	—	SRCLKEN	0000 00-0	67, 115
9Ah	SRCON1	SRCS1	SRCS0	—	—	—	—	—	—	00-- ----	67, 115
9Bh	—	未实现								—	—
9Ch	—	未实现								—	—
9Dh	—	未实现								—	—
9Eh	ADRESL[(3)]	左对齐结果的低 2 位或右对齐结果的低 8 位								xxxx xxxx	78, 115
9Fh	ADCON1[(3)]	—	ADCS2	ADCS1	ADCS0	—	—	—	—	-000 ----	77, 115

图注： – = 未实现位 （读为 0）， u = 不变， x = 未知， q = 值取决于具体条件，阴影 = 未实现

注 1： IRP 和 RP1 位保留，始终保持这两位清零。

2： 当在配置字寄存器中 MCLRE 为 1 时， RA3 上拉使能。

3： 仅限 PIC16F616/16HV616。

2.2.2.1 STATUS 寄存器

STATUS 寄存器如寄存器 2-1 所示，包括：

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

STATUS 寄存器与任何其他寄存器一样，可作为任何指令的目标寄存器。如果一条影响 Z、 DC 或 C 位的指令以 STATUS 寄存器作为目标寄存器，那么对这三个位的写操作将被禁止。这些位根据器件逻辑被置 1 或清零。而且， $\overline{TO}$ 和 $\overline{PD}$ 位均为不可写位。因此，当执行一条将 STATUS 寄存器作为目标寄存器的指令时，运行结果可能会与预想的不同。

例如，指令 `CLRF STATUS` 将会清除 STATUS 寄存器中的高三位，并将 Z 位置 1。这将使 STATUS 寄存器中的值成为 `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 寄存器的值，因为这些指令不会影响任何状态位。关于其他不会影响任何状态位的指令，请参见**第 13.0 节 "指令集汇总"**。

> **注 1：** PIC16F610/616/16HV610/616 未使用 STATUS 寄存器的 IRP 和 RP1 位，它们应保持清零。不建议使用这些位，因为这可能会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。请参见 `SUBLW` 和 `SUBWF` 指令中的示例。

寄存器 2-1： STATUS：状态寄存器

保留	保留	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
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	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **IRP：**该位保留且应保持为 0。

bit 6 **RP1：**该位保留且应保持为 0。

bit 5 **RP0：**寄存器存储区选择位 （用于直接寻址）
1 = Bank 1 （80h – FFh）
0 = Bank 0 （00h – 7Fh）

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

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

bit 2 **Z：**全零标志位
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 / $\overline{借位}$位 （`ADDWF`、 `ADDLW`、 `SUBLW` 和 `SUBWF` 指令），对于$\overline{借位}$，极性是相反的。
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 / $\overline{借位}$位 [(1)] （`ADDWF`、 `ADDLW`、 `SUBLW` 和 `SUBWF` 指令）
1 = 结果的最高位发生了进位
0 = 结果的最高位未发生进位

注 1： 对于$\overline{借位}$，极性是相反的。减法是通过加上第二个操作数的二进制补码来执行的。对于移位指令 （`RRF` 和 `RLF`），此位装入源寄存器的最高位或最低位。

2.2.2.2 OPTION 寄存器

OPTION 寄存器是可读写寄存器，有各种控制位，用来配置以下各项：

- Timer0/WDT 预分频器
- 外部 RA2/INT 中断
- Timer0
- PORTA 的弱上拉

> **注：** 要使 Timer0 获得 1:1 的预分频比，可将 OPTION 寄存器的 PSA 位置为 1，以将预分频器分配给 WDT。请参见**第 5.1.3 节“可软件编程的预分频器”**。

寄存器 2-2： OPTION_REG：OPTION 寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
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

bit 7 **$\overline{\text{RAPU}}$**：PORTA 上拉使能位
1 = 禁止 PORTA 上拉
0 = 通过各端口锁存值使能 PORTA 上拉

bit 6 **INTEDG**：中断触发边沿选择位
1 = RA2/INT 引脚的上升沿触发中断
0 = RA2/INT 引脚的下降沿触发中断

bit 5 **T0CS**：Timer0 时钟源选择位
1 = RA2/T0CKI 引脚上的电平跳变
0 = 内部指令周期时钟 （FOSC/4）

bit 4 **T0SE**：Timer0 时钟源边沿选择位
1 = 在 RA2/T0CKI 引脚信号从高至低跳变时，递增计数
0 = 在 RA2/T0CKI 引脚信号从低至高跳变时，递增计数

bit 3 **PSA**：预分频器分配控制位
1 = 预分频器分配给 WDT
0 = 预分频器分配给 Timer0 模块

bit 2-0 **PS<2:0>**：预分频比选择位

位值	TIMER0 预分频比	WDT 预分频比
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

2.2.2.3 INTCON 寄存器

INTCON 寄存器是可读写的寄存器，它包含 TMR0 寄存器溢出、PORTA 电平变化和外部 RA2/INT 引脚中断等各种使能控制位和标志位。

> **注：** 当中断条件发生时，无论相应中断允许位或全局中断允许位 GIE （在 INTCON 寄存器中）的状态如何，中断标志位都将被置 1。用户软件应在允许一个中断前，先将相应的中断标志位清零。

寄存器 2-3： INTCON：中断控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF
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

bit 7 **GIE：**全局中断允许位
1 = 允许所有未被屏蔽的中断
0 = 禁止所有中断

bit 6 **PEIE：**外设中断允许位
1 = 允许所有未被屏蔽的外设中断
0 = 禁止所有外设中断

bit 5 **T0IE：**Timer0 溢出中断允许位
1 = 允许 Timer0 中断
0 = 禁止 Timer0 中断

bit 4 **INTE：**RA2/INT 外部中断允许位
1 = 允许 RA2/INT 外部中断
0 = 禁止 RA2/INT 外部中断

bit 3 **RAIE：**PORTA 电平变化中断允许位 (1)
1 = 允许 PORTA 电平变化中断
0 = 禁止 PORTA 电平变化中断

bit 2 **T0IF：**Timer0 溢出中断标志位 (2)
1 = Timer0 寄存器溢出 （必须用软件清零）
0 = Timer0 寄存器未溢出

bit 1 **INTF：**RA2/INT 外部中断标志
1 = RA2/INT 外部中断发生 （必须用软件清零）
0 = RA2/INT 外部中断没有发生

bit 0 **RAIF：**PORTA 电平变化中断标志位
1 = 至少有一个 PORTA <5:0> 引脚状态发生变化时 （必须用软件清零）
0 = PORTA <5:0> 引脚状态均未发生变化

注 1： IOCA 寄存器也必须被使能。
2： 当 TMR0 计数出现计满返回时， T0IF 位将被置 1。 TMR0 计数值在复位时不变，而且应在清除 T0IF 位之前对其进行初始化。

2.2.2.4 PIE1 寄存器

PIE1 寄存器包含外设中断允许位，如寄存器 2-4 所示。

注：	必须将 INTCON 寄存器的 PEIE 位置 1，以允许外设中断。

寄存器 2-4： PIE1：外设中断允许寄存器 1

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
—	ADIE(1)	CCP1IE(1)	C2IE	C1IE	—	TMR2IE(1)	TMR1IE
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

bit 7 **未实现**：读为 0

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

bit 5 **CCP1IE**：CCP1 中断允许位 (1)
1 = 允许 CCP1 中断
0 = 禁止 CCP1 中断

bit 4 **C2IE**：比较器 C2 中断允许位
1 = 允许比较器 C2 中断
0 = 禁止比较器 C2 中断

bit 3 **C1IE**：比较器 C1 中断允许位
1 = 允许比较器 C1 中断
0 = 禁止比较器 C1 中断

bit 2 **未实现**：读为 0

bit 1 **TMR2IE**：Timer2 与 PR2 匹配中断允许位 (1)
1 = 允许 Timer2 与 PR2 匹配中断
0 = 禁止 Timer2 与 PR2 匹配中断

bit 0 **TMR1IE**：Timer1 溢出中断允许位
1 = 允许 Timer1 溢出中断
0 = 禁止 Timer1 溢出中断

注 **1**： 仅限 PIC16F616/16HV616。 PIC16F610/16HV610 未实现，读为 0。

2.2.2.5 PIR1 寄存器

PIR1 寄存器包含外设中断标志位，如寄存器 2-5 所示。

注：	当中断条件发生时，无论相应中断允许位或全局中断允许位 GIE （在 INTCON 寄存器中）的状态如何，中断标志位都将被置 1。用户软件应在允许一个中断前，先将相应的中断标志位清零。

寄存器 2-5： PIR1：外设中断请求寄存器 1

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
—	ADIF(1)	CCP1IF(1)	C2IF	C1IF	—	TMR2IF(1)	TMR1IF
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

bit 7 **未实现**：读为 0

bit 6 **ADIF**：A/D 中断标志位 (1)
1 = A/D 转换完成
0 = A/D 转换未完成或尚未开始

bit 5 **CCP1IF**：CCP1 中断标志位 (1)
<u>捕捉模式：</u>
1= 发生了 TMR1 寄存器捕捉 （必须用软件清零）
0= 未发生 TMR1 寄存器捕捉
<u>比较模式：</u>
1= 发生了 TMR1 寄存器的比较匹配 （必须用软件清零）
0= 未发生 TMR1 寄存器的比较匹配
<u>PWM 模式：</u>
在此模式下不使用

bit 4 **C2IF**：比较器 C2 中断标志位
1 = 比较器 C2 输出已改变 （必须用软件清零）
0 = 比较器 C2 输出未改变

bit 3 **C1IF**：比较器 C1 中断标志位
1 = 比较器 C1 输出已改变 （必须用软件清零）
0 = 比较器 C1 输出未改变

bit 2 **未实现**：读为 0

bit 1 **TMR2IF**：Timer2 与 PR2 匹配中断标志位 (1)
1 = Timer2 与 PR2 发生匹配 （必须用软件清零）
0 = Timer2 与 PR2 未发生匹配

bit 0 **TMR1IF**：Timer1 溢出中断标志位
1 = Timer1 寄存器溢出 （必须用软件清零）
0 = Timer1 寄存器未溢出

注 **1**： 仅限 PIC16F616/16HV616。 PIC16F610/16HV610 未实现，读为 0。

2.2.2.6 PCON 寄存器

电源控制 （PCON）寄存器 （见表 12-2）包含区分以下各种复位的标志位：

- 上电复位 （$\overline{\text{POR}}$）
- 欠压复位 （$\overline{\text{BOR}}$）
- 看门狗定时器复位 （WDT）
- 外部 $\overline{\text{MCLR}}$ 复位

PCON 寄存器还控制 $\overline{\text{BOR}}$ 的软件使能。

PCON 寄存器位如寄存器 2-6 所示。

寄存器 2-6： PCON：电源控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0(1)
—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
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

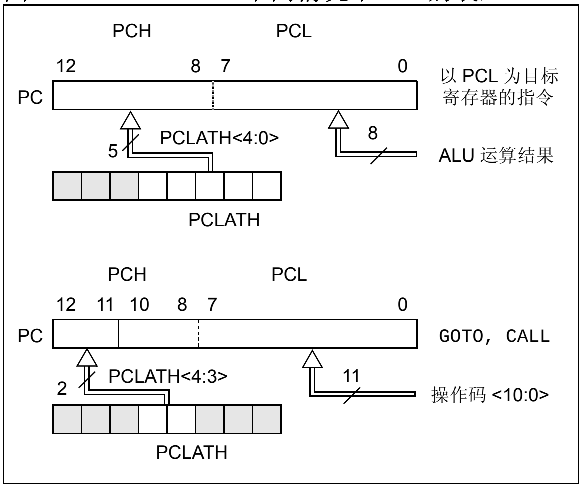
bit 7-2	**未实现**：读为 0
bit 1	**$\overline{\text{POR}}$**：上电复位状态位 1 = 未发生上电复位 0 = 发生上电复位 （上电复位发生后，必须用软件置 1）
bit 0	**$\overline{\text{BOR}}$**：欠压复位状态位 1 = 未发生欠压复位 0 = 发生欠压复位 （欠压复位发生后，必须用软件置 1）

注 **1**： 如果禁止欠压复位，则读为 0。

2.3 PCL 和 PCLATH

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

图 2-5： 不同情况下 **PC** 的装入



2.3.1 修改 PCL

在执行以 PCL 寄存器作为目标寄存器的任何指令的同时，也会使程序计数器的 PC<12:8> 位（PCH）被 PCLATH 寄存器的内容所代替。这使得可以通过将所需的高 5 位写入 PCLATH 寄存器来改变程序计数器的整个内容。当将低 8 位写入 PCL 寄存器时，程序计数器的所有 13 位都将变为 PCLATH 寄存器中和那些被写入 PCL 寄存器的值。

计算 GOTO 指令是通过向程序计数器加一个偏移量（ADDWF PCL）来实现的。当通过修改 PCL 寄存器跳转到查找表或程序转移表（计算 GOTO）时，应特别注意。假设 PCLATH 被设置为表起始地址，如果表长度大于 255 条指令，或如果在表中间存储器地址的低 8 位从 0xFF 计满返回到 0x00，那么 PCLATH 必须为在表起始和表中的目标存储单元之间发生计满返回的每个地址进行递增。

更多信息，请参见应用笔记 AN556，"*Implementing a Table Read*"（DS00556）。

2.3.2 堆栈

PIC16F610/616/16HV610/616 系列具有 8 层 x 13 位宽的硬件堆栈（见图 2-1）。堆栈空间既不占用程序存储空间，也不占用数据存储空间，而且堆栈指针是不可读写的。当执行 CALL 指令或当中断导致程序跳转时，PC 值将被压入堆栈。而在执行 RETURN、RETLW 或 RETFIE 指令时，将从堆栈中弹出 PC 值。PCLATH 不受压栈或出栈操作的影响。

堆栈的工作原理犹如循环缓冲区。这意味着当压栈 8 次后，第 9 次压入堆栈的值将会覆盖第一次压栈时所保存的值，而第 10 次压入堆栈的值将覆盖第二次压栈时保存的值，依此类推。

> **注 1：** 不存在指明堆栈是否上溢或下溢的状态标志位。
>
> **2：** 不存在被称为 PUSH 或 POP 的指令 / 助记符。堆栈的压入或弹出是源于执行了 CALL、RETURN、RETLW 和 RETFIE 指令，或源于跳转到中断向量地址。

2.4 间接寻址、INDF 和 FSR 寄存器

INDF 寄存器不是实际存在的寄存器，对 INDF 寄存器进行寻址将产生间接寻址。

使用 INDF 寄存器可进行间接寻址。任何使用 INDF 寄存器的指令，实际上是对文件选择寄存器（FSR）所指向的数据进行存取。间接对 INDF 进行读操作将返回 00h。间接对 INDF 寄存器进行写操作将导致空操作（尽管可能会影响状态位）。通过将 8 位的 FSR 寄存器与 STATUS 寄存器的 IRP 位进行组合可得到一个有效的 9 位地址，如图 2-7 所示。

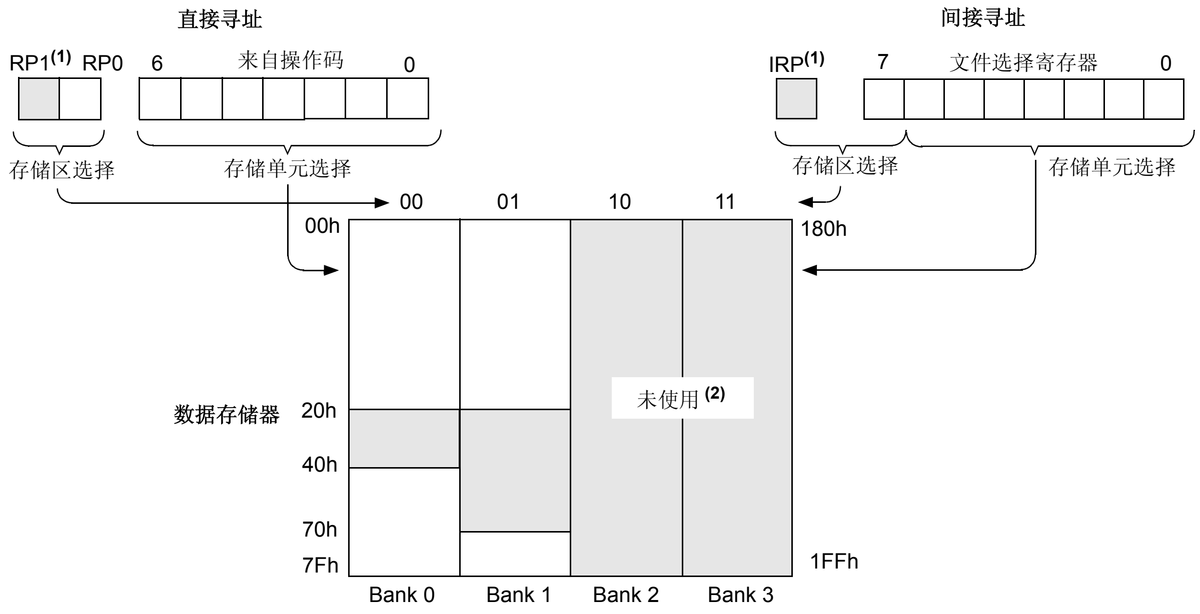
例 2-1 给出了一个使用间接寻址将 RAM 地址单元 40h-4Fh 清零的简单程序。

例 2-1： 间接寻址

```
        MOVLW   0x40    ;initialize pointer
        MOVWF   FSR     ;to RAM
NEXT    CLRF    INDF    ;clear INDF register
        INCF    FSR, F  ;inc pointer
        BTFSS   FSR,4   ;all done?
        GOTO    NEXT    ;no clear next
CONTINUE                ;yes continue
```

 初稿

图 2-6： PIC16F610/16HV610 的直接 / 间接寻址



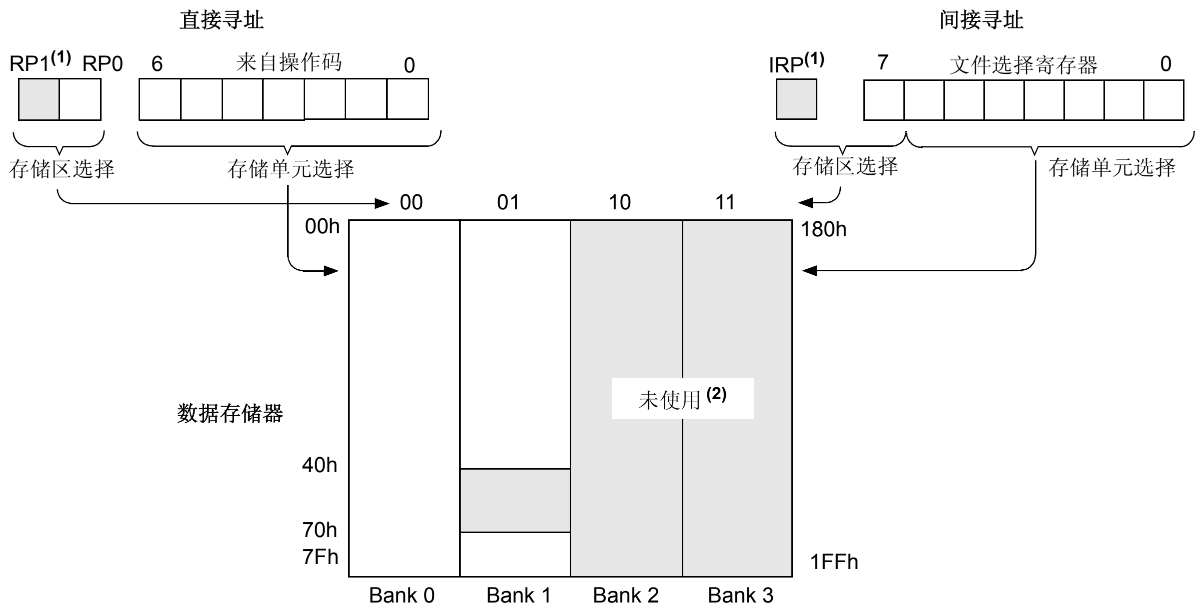
关于存储器映射的详细信息，请参见图 2-3。

未实现的数据存储单元，读为 0。

注 1： RP1 和 IRP 位保留；始终保持这两位清零。

2： 对 Bank 2 和 Bank 3 的访问会分别映射到 Bank 0 和 Bank 1。

图 2-7： PIC16F616/16HV616 的直接 / 间接寻址



关于存储器映射的详细信息，请参见图 2-4。

未实现的数据存储单元，读为 0。

注 1： RP1 和 IRP 位保留；始终保持这两位清零。

2： 对 Bank 2 和 Bank 3 的访问会分别映射到 Bank 0 和 Bank 1。

注：

3.0 振荡器模块

3.1 概述

振荡器模块具有多种时钟源和选择特性，从而使之可广泛使用于各种应用中，同时最大限度地发挥应用性能并降低功耗。图 3-1 为振荡器模块的框图。

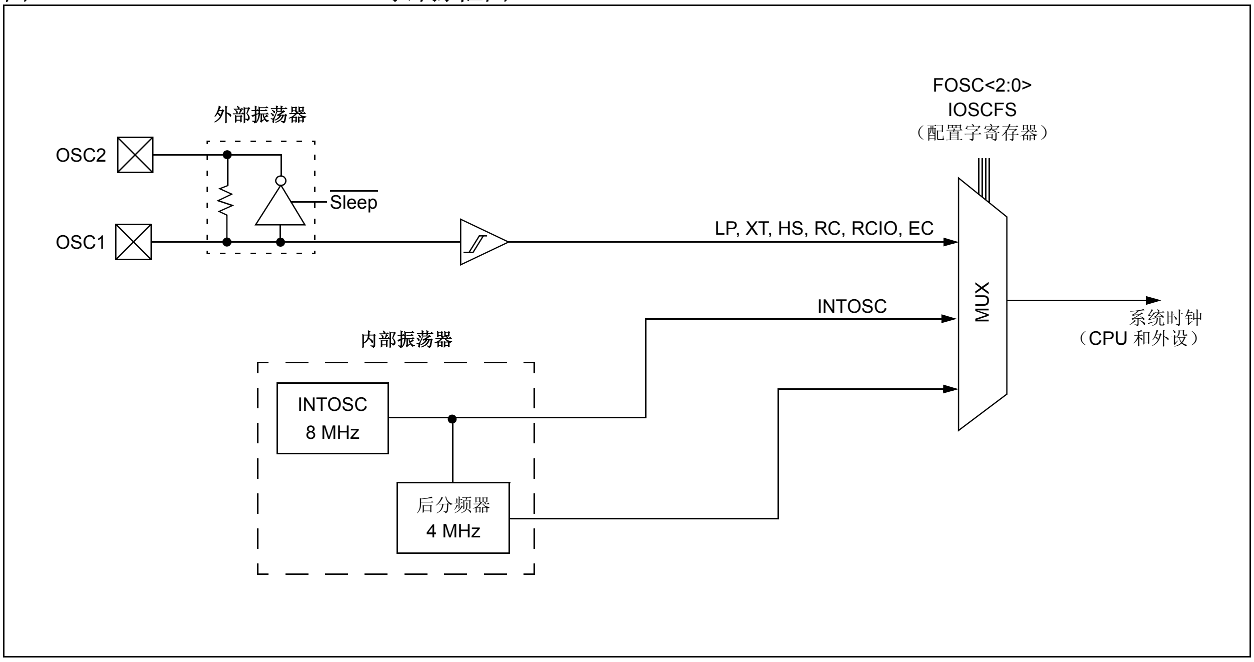
时钟源可配置为来自外部振荡器、石英晶体谐振器、陶瓷谐振器以及阻容（Resistor-Capacitor，RC）电路。此外，系统时钟源还可配置为两种可选速度之一：内部或外部系统时钟源。

振荡器模块可配置为以下 8 种时钟模式之一。

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

通过配置字寄存器 （CONFIG）中的 FOSC<2:0> 位来配置时钟源模式。内部振荡器模块提供一种可选的系统时钟模式，4 MHz（后分频器）或 8 MHz（INTOSC）。

图 3-1： PIC® MCU 时钟源框图



3.2 时钟源模式

时钟源模式可分为外部和内部模式。

- 外部时钟模式依靠外部电路提供时钟源。例如：振荡器模块（EC 模式）、石英晶体谐振器或陶瓷谐振器（LP、XT 和 HS 模式）以及阻容（RC）模式电路。
- 内部时钟源内置于振荡器模块中。振荡器模块有两种时钟频率供选择：4 MHz 和 8 MHz

通过配置字寄存器的 FOSC<2:0> 位在外部或内部时钟源之间选择系统时钟。

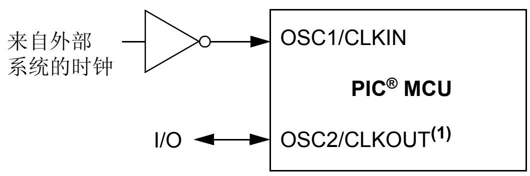
3.3 外部时钟模式

3.3.1 EC 模式

外部时钟（EC）模式允许外部产生的逻辑电平作为系统时钟源。工作在此模式下时，外部时钟源连接到 OSC1 输入，OSC2 引脚可用作通用 I/O。图 3-2 给出了 EC 模式的引脚连接。

当选取 EC 模式时，振荡器起振定时器（OST）被禁止。因此，上电复位（POR）后或者从休眠中唤醒后的操作不存在延时。因为 PIC® MCU 的设计是完全静态的，停止外部时钟输入将使器件暂停工作并保持所有数据完整。当再次启动外部时钟时，器件恢复工作，就好像没有停止过一样。

图 3-2：　外部时钟（EC）模式的工作原理



注　1：　**第 1.0 节“器件概述”**中列出了备用引脚功能。

3.3.2 振荡器起振定时器（OST）

如果振荡器模块被配置为 LP、XT 或 HS 模式，则振荡器起振定时器（OST）对来自 OSC1 的振荡计数 1024 次。这发生在上电复位（POR）和上电延时定时器（PWRT）超时（如果配置了），或从休眠中唤醒后。在此期间，程序计数器不递增，程序执行暂停。OST 确保使用石英晶体谐振器或陶瓷谐振器的振荡器电路已经起振并为PIC16F610/16HV610和PIC16F616/16HV616器件提供稳定的系统时钟。当在时钟源之间切换时，需要一定的延时以使新时钟稳定。表 3-1 给出了振荡器延时的例子。

表 3-1：　振荡器延时示例

切换自	切换到	频率	振荡器延时
休眠 /POR	INTOSC	4 MHz 到 8 MHz	振荡器预热延时（TWARM）
休眠 /POR	EC 或 RC	DC – 20 MHz	2 个指令周期
休眠 /POR	LP、XT 或 HS	32 kHz 到 20 MHz	1024 个时钟周期（OST）

3.3.3 LP、 XT 和 HS 模式

LP、XT 和 HS 模式支持使用连接到 OSC1 和 OSC2 引脚的石英晶体谐振器或陶瓷谐振器（图 3-3）。模式选择内部反相放大器的低、中或高增益设定，以支持各种谐振器类型及速度。

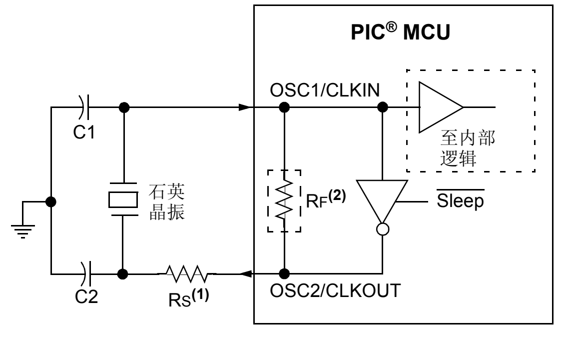
LP 振荡器模式选择内部反相放大器的最低增益设定。LP模式的电流消耗在三种模式中最小。该模式设计用来驱动仅 32.768 kHz 的音叉（Tuning Fork）型晶振（钟表晶振）。

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

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

图 3-3 和图 3-4 分别给出了石英晶体谐振器和陶瓷谐振器的典型电路。

图 3-3： 石英晶振的工作原理 （**LP、XT 或 HS 模式**）

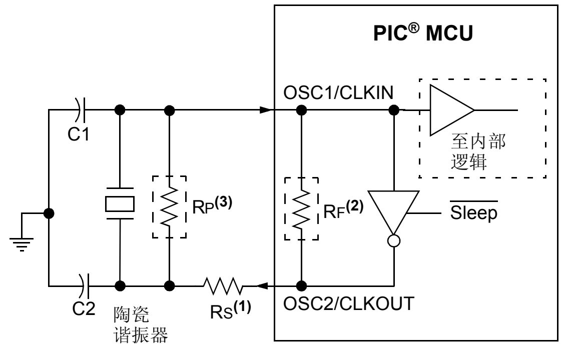


注 1：对于低驱动级别的石英晶振，可能需要串联一个电阻 （RS）。

2：RF 的值随选取的振荡模式而变化 （典型值在 2 MΩ 到 10 MΩ 之间）。

> 注 1：石英晶振的特性随类型、封装和制造商而变化。要了解规格说明和推荐应用，应查阅制造商提供的数据手册。
>
> 2：应始终验证振荡器在应用要求的 VDD 和温度范围内的性能。
>
> 3：如需获取振荡器设计帮助，请参见以下 Microchip 应用笔记：
> - AN826，“*Crystal Oscillator Basics and Crystal Selection for rfPIC® and PIC® Devices*”（DS00826）
> - AN849，“*Basic PIC® Oscillator Design*”（DS00849）
> - AN943，“*Practical PIC® Oscillator Analysis and Design*”（DS00943）
> - AN949，“*Making Your Oscillator Work*”（DS00949）

图 3-4： 陶瓷谐振器的工作原理 （**XT 或 HS 模式**）



注 1：对于低驱动级别的陶瓷谐振器，可能需要串联一个电阻 （RS）。

2：RF 的值随选取的振荡模式而变化 （典型值在 2 MΩ 到 10 MΩ 之间）。

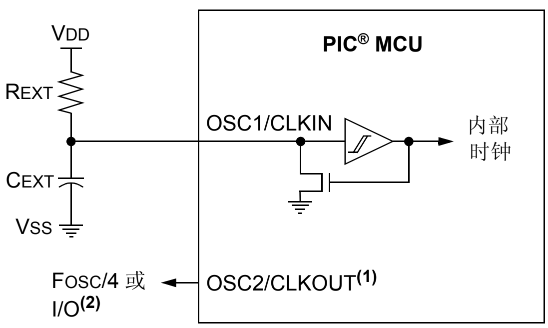
3：为使陶瓷谐振器正常工作，可能需要并联一个反馈电阻 （RP）。

3.3.4 外部 RC 模式

外部阻容（RC）模式支持使用外部 RC 电路。对时钟精度要求不高时，这使设计人员有了很大的频率选择空间，且保持成本最低。有以下两种模式：RC 和 RCIO。

在 RC 模式下，RC 电路连接到 OSC1 引脚。OSC2/CLKOUT 引脚输出为 RC 振荡器频率的 4 分频。该信号可用来为外部电路、同步、校准、测试或其他应用需求提供时钟。图 3-5 给出了外部 RC 模式的连接图。

图 3-5： 外部 RC 模式



推荐值： 10 kΩ ≤ REXT ≤ 100 kΩ， <3V
3 kΩ ≤ REXT ≤ 100 kΩ， 3-5V
CEXT > 20 pF， 2-5V

注 1： **第 1.0 节 “器件概述”**中列出了备用引脚功能。
2： 输出取决于 RC 或 RCIO 时钟模式。

在 RCIO 模式下，RC 电路连接到 OSC1。OSC2 成为额外的通用 I/O 引脚。

RC 振荡器频率与供电电压、电阻（REXT）和电容（CEXT）值以及工作温度有关。影响振荡器频率的其他因素有：

- 电压门限值变化
- 元件容差
- 不同封装的电容

用户还应考虑因所使用的外部 RC 元件的容差而导致的差异。

3.4 内部时钟模式

振荡器模块提供一个可选的系统时钟源，4 MHz 或 8 MHz。通过配置字的 IOSCFS 位来配置可选频率。

内部振荡器的频率可使用 OSCTUNE 寄存器通过软件由用户来调节。

3.4.1 INTOSC 和 INTOSCIO 模式

当在配置字寄存器（CONFIG）中使用振荡器选择或 FOSC<2:0> 位对器件进行编程时，在 INTOSC 和 INTOSCIO 模式下将内部振荡器配置为系统时钟源。更多信息，请参见**第 12.0 节 “CPU 的特殊功能”**。

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

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

 初稿

3.4.1.1 OSCTUNE 寄存器

该振荡器在出厂时已校准，但可通过在软件中写入 OSCTUNE 寄存器 （寄存器 3-1）来进行调节。

OSCTUNE 寄存器的默认值为 0。该值是一个 5 位的二进制补码。

当 OSCTUNE 寄存器被修改时，该频率将开始转变为新频率。转变期间，代码将继续执行。是否已发生频率转变并无明确的指示。

寄存器 3-1： OSCTUNE：振荡器调节寄存器

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0
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

bit 7-5 未实现：读为 0

bit 4-0 **TUN<4:0>**：频率调节位
01111 = 最高频率
01110 =
•
•
•
00001 =
00000 = 振荡器模块运行在出厂校准的频率上
11111 =
•
•
•
10000 = 最低频率

表 3-2： 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	POR/BOR 时的值	所有其他复位时的值 (1)
CONFIG(2)	IOSCFS	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0	—	—
OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	---u uuuu

图注： x = 未知， u = 不变， – = 未实现位 （读为 0）。振荡器不使用阴影单元。
注 1： 其他 （非上电）复位包括在正常操作期间的 MCLR 复位和看门狗定时器复位。
2： 请参见配置字寄存器 （寄存器 12-1）了解所有寄存器位的操作。

注：

4.0 I/O 端口

器件具有多达 11 个通用 I/O 引脚和 1 个输入引脚。根据外设的使能情况，部分甚至全部引脚可能不能用于通用 I/O。通常而言，当某个外设使能时，其相关引脚可能不能用作通用 I/O 引脚。

4.1 PORTA 和 TRISA 寄存器

PORTA 是一个 6 位宽的双向端口，对应的数据方向寄存器是 TRISA （寄存器 4-2）。将 TRISA 某位置 1 （= 1）时，会将 PORTA 的相应引脚设为输入 （即，禁止输出驱动器）。将 TRISA 某位清零 （= 0）时，会将 PORTA 的相应引脚设为输出 （即，使能输出驱动器并将输出锁存器中的内容输出到选中引脚）。 RA3 是个例外，仅可作为输入引脚，其 TRIS 位总是读为 1。例 4-1 显示了如何初始化 PORTA。

读 PORTA 寄存器 （寄存器 4-1）将读出相应引脚的状态，而对其进行写操作则是将数据写入端口锁存器。所有写操作都是 "读－修改－写" 操作。因此，对端口的写操作意味着总是先读端口引脚电平状态，然后修改这个值，最后再写入该端口的数据锁存器。当 MCLRE = 1 时， RA3 读为 0。

TRISA 寄存器控制着 PORTA 引脚的方向，即使它们用作模拟输入引脚。当引脚用于模拟输入时，用户必须确保 TRISA 寄存器中的各位保持置 1。配置为模拟输入的 I/O 引脚总是读为 0。

> **注：** 必须对 ANSEL 寄存器进行初始化，以将模拟通道配置为数字输入。配置为模拟输入的引脚将读为 0，且不能产生中断。

例 4-1： 初始化 PORTA

```
BCF    STATUS,RP0     ;Bank 0
CLRF   PORTA          ;Init PORTA
BSF    STATUS,RP0     ;Bank 1
CLRF   ANSEL          ;digital I/O
MOVLW  0Ch            ;Set RA<3:2> as inputs
MOVWF  TRISA          ;and set RA<5:4,1:0>
                      ;as outputs
BCF    STATUS,RP0     ;Bank 0
```

寄存器 4-1： PORTA：PORTA 寄存器

U-0	U-0	R/W-x	R/W-0	R-x	R/W-0	R/W-0	R/W-0
—	—	RA5	RA4	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	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **未实现：** 读为 0

bit 5-0 **RA<5:0>：** PORTA I/O 引脚位
1 = PORTA 引脚 > V_{IH}
0 = PORTA 引脚 < V_{IL}

寄存器 4-2： TRISA：PORTA 三态寄存器

U-0	U-0	R/W-1	R/W-1	R-1	R/W-1	R/W-1	R/W-1
—	—	TRISA5	TRISA4	TRISA3	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	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **未实现：** 读为 0

bit 5-0 **TRISA<5:0>：** PORTA 三态控制位
1 = PORTA 引脚配置为输入 （三态）
0 = PORTA 引脚配置为输出

注 **1：** TRISA<3> 总是读为 1。
2： 在 XT、 HS 和 LP 振荡器模式下， TRISA<5:4> 总是读为 1。

 初稿

4.2 引脚的其他功能

PIC16F610/616/16HV610/616 中每个 PORTA 引脚都具有电平变化中断功能和弱上拉功能。以下三节将对这些功能进行介绍。

4.2.1 ANSEL 寄存器

ANSEL 寄存器用于将 I/O 引脚的输入模式配置为模拟。将相应的 ANSEL 位设置为高电平将使引脚上的所有数字读数都读为 0，并允许引脚上的模拟功能正确操作。

ANSEL 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工作，但输入模式将变为模拟。当在受影响的端口上执行读－修改－写指令时，这会引起意外操作。

4.2.2 弱上拉

除 RA3 外的每个 PORTA 引脚都具有可单独配置的内部弱上拉功能。控制位 WPUAx 用于使能或禁止每个上拉功能。请参见寄存器 4-4。当端口引脚被配置为输出时，这些弱上拉会自动关闭。上电复位时上拉功能被 OPTION 寄存器的 RAPU 位禁止。当配置为 MCLR 时，RA3 上的弱上拉自动使能；当 RA3 配置为 I/O 时则被禁止。没有对 MCLR 上拉的软件控制。

4.2.3 电平变化中断

每个 PORTA 引脚都可单独配置为电平变化中断引脚。控制位 IOCAx 用于使能或禁止各引脚的中断功能。请参见寄存器 4-5。电平变化中断功能在上电复位时被禁止。

对于允许了电平变化中断功能的引脚，其值将与上次读取的 PORTA 的旧锁存值相比较。所有与上次读取值不匹配的输出进行或运算，运算结果用来设置 INTCON 寄存器 （寄存器 2-3）中的 PORTA 电平变化中断标志位（RAIF）。

该中断可将器件从休眠状态唤醒。用户在中断服务程序中可通过以下方式清除该中断：

a） 对 PORTA 的任何读或写操作。这将结束不匹配条件，然后：

b） 清零标志位 RAIF。

不匹配条件将继续将标志位 RAIF 置 1。读 PORTA 将结束不匹配条件并将标志位 RAIF 清零。保持上一次读取值的锁存器不受 MCLR 或 BOR 复位的影响。在这些复位之后，如果存在不匹配情况， RAIF 标志还将继续被置 1。

注：	当正在执行任何 PORTA 操作时发生了 I/O 引脚电平变化，则 RAIF 中断标志位可能不会被置 1。

寄存器 4-3： ANSEL：模拟选择寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
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

bit 7-0 **ANS<7:0>**：模拟选择位
将 AN<7:0> 引脚的功能分别选为模拟或数字。
1 = 模拟输入。引脚被配置为模拟输入 (1)。
0 = 数字 I/O。分配引脚为端口或特殊功能。

注 1： 将引脚设置为模拟输入将自动禁止数字输入电路、弱上拉和电平变化中断 （如果具有这些功能的话）。必须将相应的 TRIS 位设置为输入模式，以允许从外部控制引脚电压。

寄存器 4-4： WPUA：弱上拉 PORTA 寄存器

U-0	U-0	R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1
—	—	WPUA5	WPUA4	—	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	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **未实现：**读为 0

bit 5-4 **WPUA<5:4>：**弱上拉控制位
1 = 上拉使能
0 = 上拉禁止

bit 3 **未实现：**读为 0

bit 2-0 **WPUA<2:0>：**弱上拉控制位
1 = 上拉使能
0 = 上拉禁止

注 **1：** 必须使能全局 $\overline{\text{RAPU}}$ 位才能使能单独的上拉功能。
2： 如果引脚处于输出模式，则自动禁止弱上拉功能 （TRISA = 0）。
3： 当在配置字中将 RA3 配置为 $\overline{\text{MCLR}}$ 时， RA3 上拉被使能，当 RA3 被配置为输入时上拉功能被禁止。
4： 在 XT、 HS 和 LP 振荡器模式下， WPUA<5:4> 总是读为 1。

寄存器 4-5： IOCA：电平变化中断 PORTA 寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0
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

bit 7-6 **未实现：**读为 0

bit 5-0 **IOCA<5:0>：**电平变化中断 PORTA 控制位
1 = 允许电平变化中断
0 = 禁止电平变化中断

注 **1：** 要使各中断能够被识别，必须使能全局中断允许控制位 （GIE）。
2： 在 XT、 HS 和 LP 振荡器模式下， IOCA<5:4> 总是读为 1。

4.2.4 引脚说明和引脚原理图

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

4.2.4.1 RA0/AN0(1)/C1IN+/ICSPDAT

图 4-1 给出了此引脚的原理图。RA0 引脚可配置为下列功能之一：

- 通用 I/O
- ADC(1) 的模拟输入
- 比较器的模拟同相输入
- 在线串行编程数据

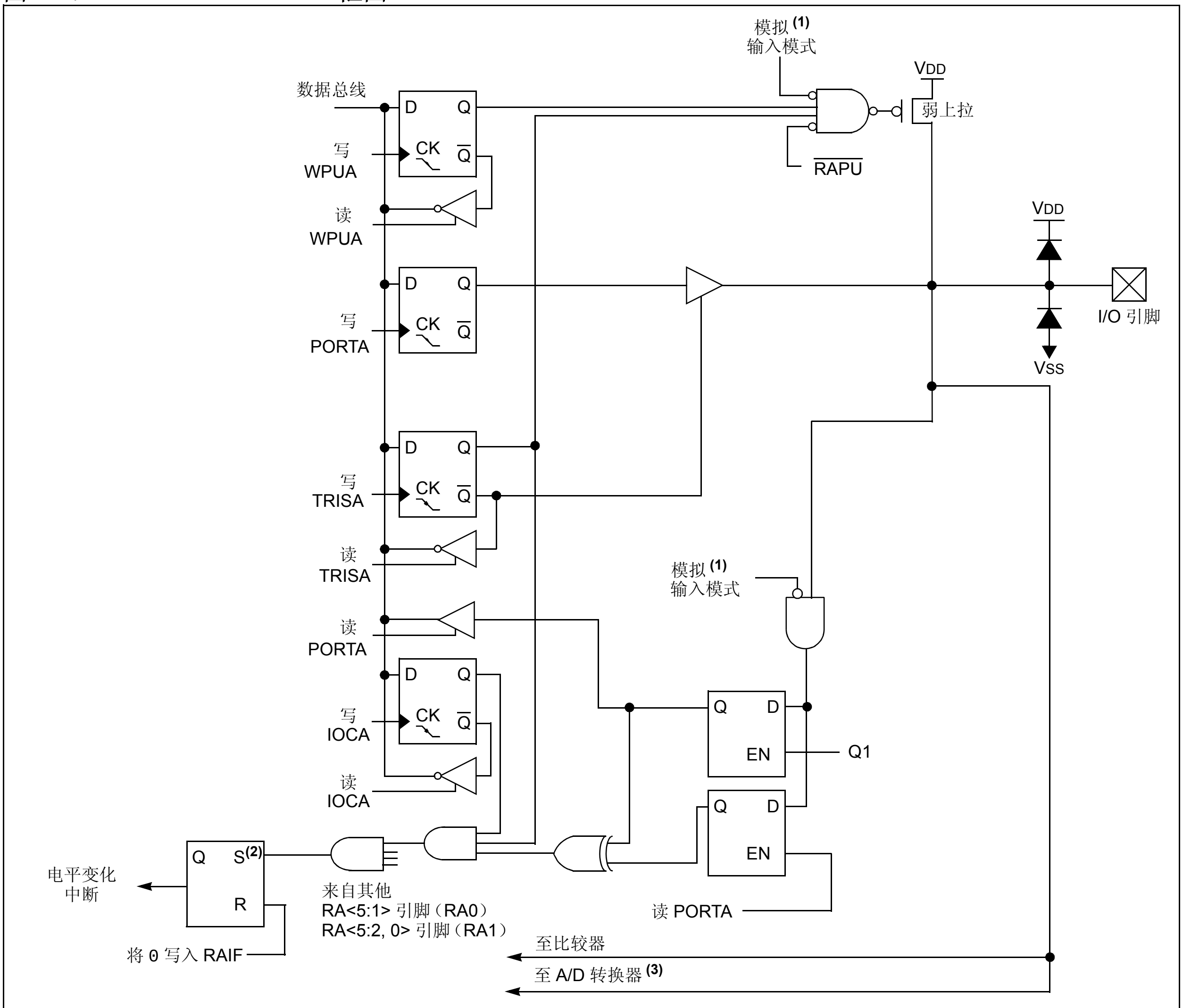
4.2.4.2 RA1/AN1(1)/C12IN0-/VREF(1)/ICSPCLK

图 4-1 给出了此引脚的原理图。RA1 引脚可配置为下列功能之一：

- 通用 I/O
- ADC(1) 的模拟输入
- 比较器的模拟反相输入
- ADC(1) 的参考电压输入
- 在线串行编程时钟

注 1：仅限 PIC16F616/16HV616。

图 4-1： RA<1:0> 框图



注 1：比较器模式和 ANSEL 决定模拟输入模式。
2：置 1 优先于复位。
3：仅限 PIC16F616/16HV616。

 初稿

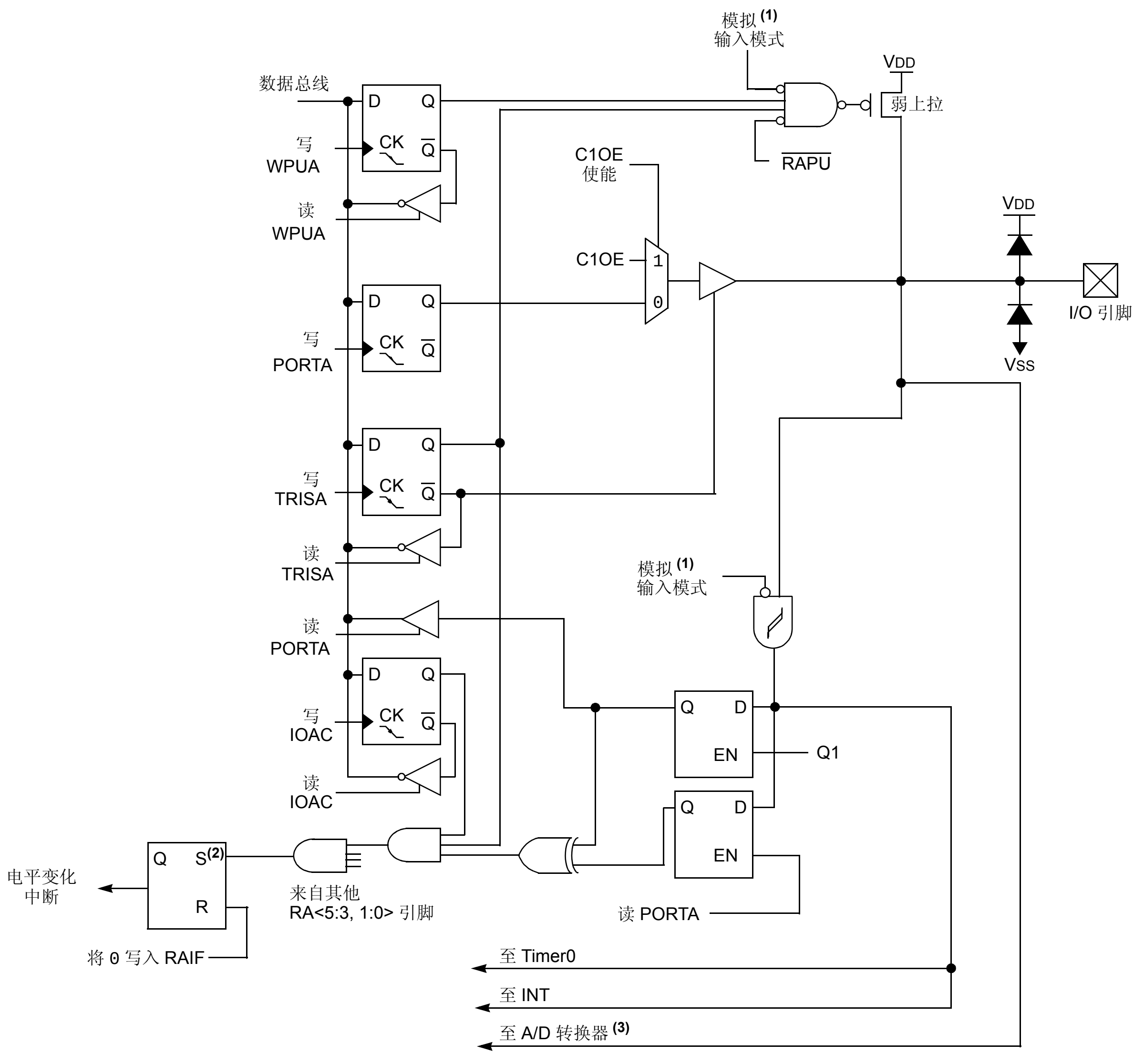
4.2.4.3 RA2/AN2[1]/T0CKI/INT/C1OUT

图 4-2 给出了此引脚的原理图。RA2 引脚可配置为下列功能之一：

- 通用 I/O
- ADC[1] 的模拟输入
- TMR0 的时钟输入
- 外部边沿触发中断
- 比较器 C1 的数字输出

注	1： 仅限 PIC16F616/16HV616。

图 4-2： RA2 框图



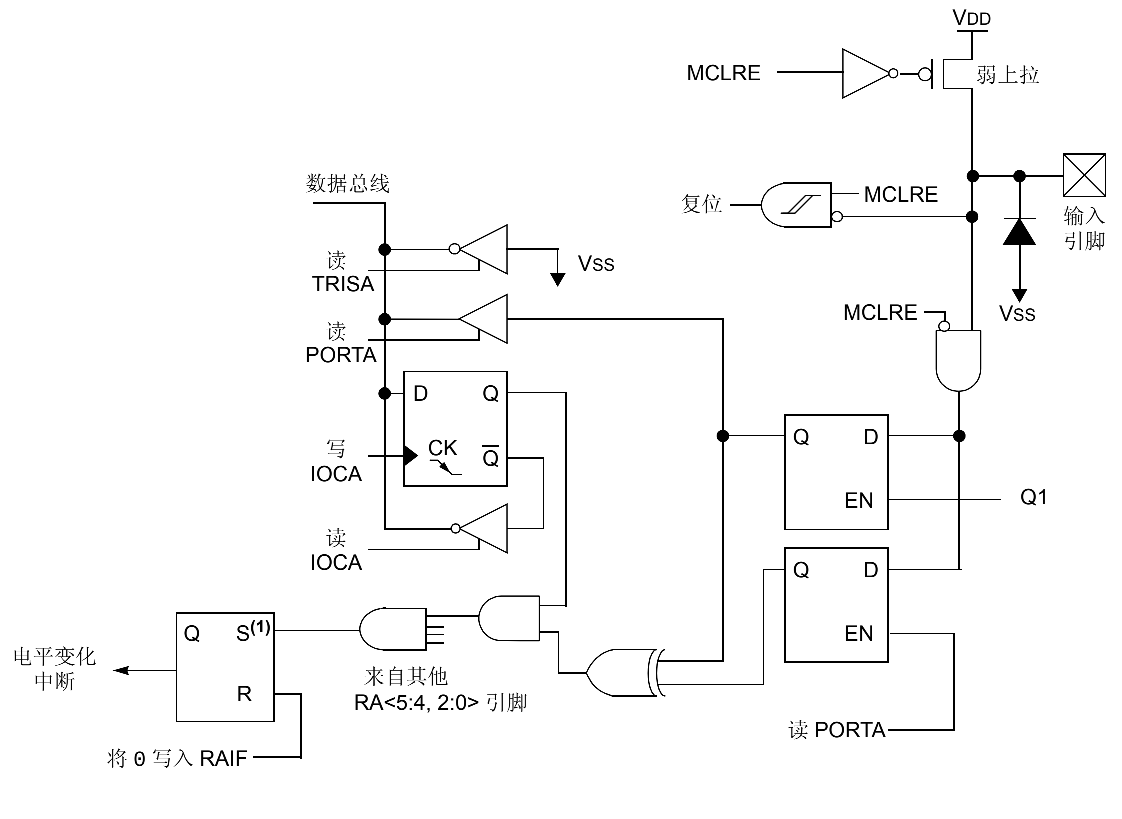
注 1： 比较器模式和 ANSEL 决定模拟输入模式。
2： 置 1 优先于复位。
3： 仅限 PIC16F616/16HV616。

4.2.4.4 RA3/$\overline{MCLR}$/VPP

图 4-3 给出了此引脚的原理图。RA3 引脚可配置为下列功能之一：

- 通用输入
- 带弱上拉的主复位

图 4-3： RA3 框图



注 1： 置 1 优先于复位。

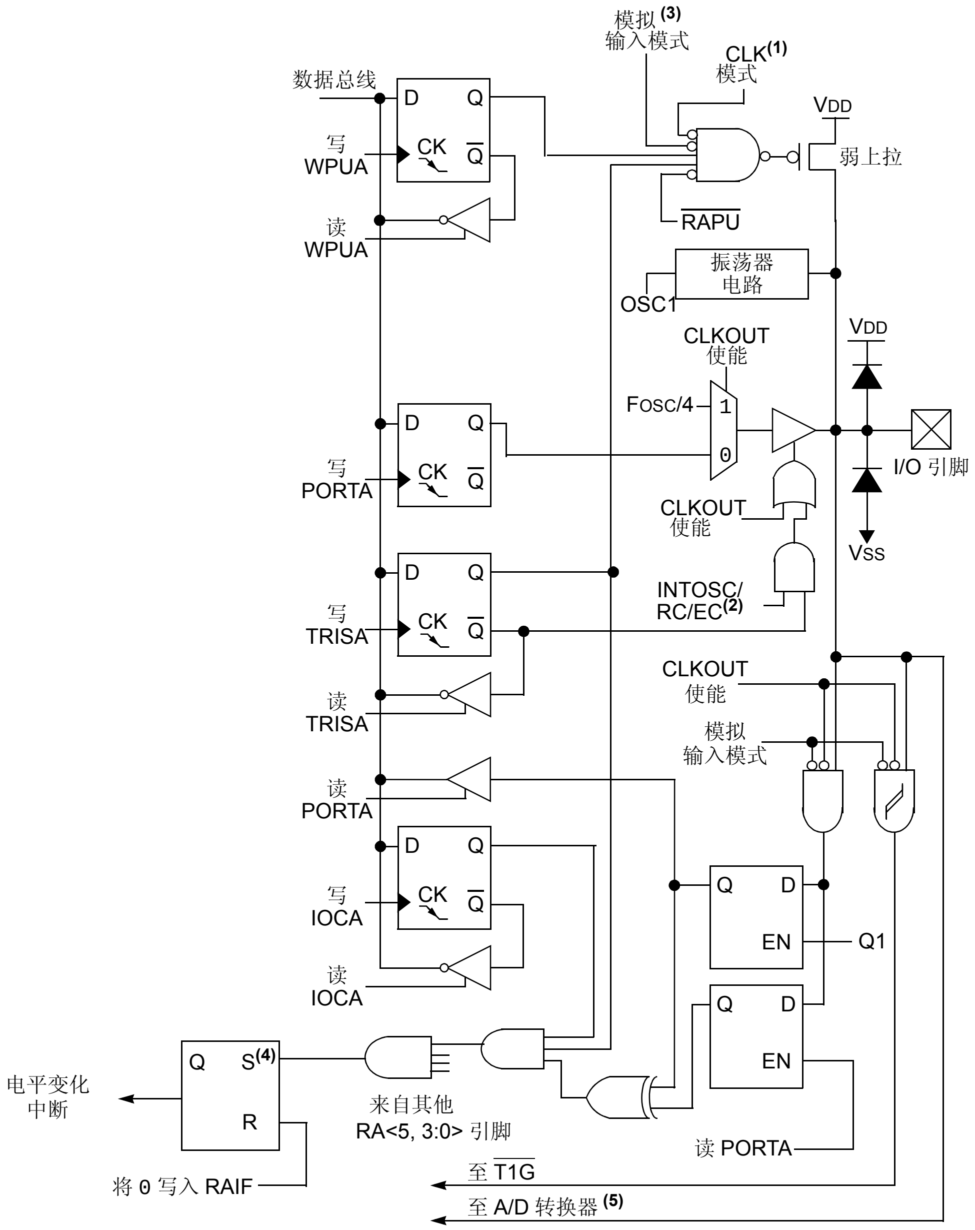
4.2.4.5 RA4/AN3[1]/$\overline{\text{T1G}}$/OSC2/CLKOUT

图 4-4 给出了此引脚的原理图。RA4 引脚可配置为下列功能之一：

- 通用 I/O
- ADC[1] 的模拟输入
- Timer1 门控 （计数使能）
- 晶振 / 谐振器连接
- 时钟输出

注	1：仅限 PIC16F616/16HV616。

图 4-4： RA4 框图



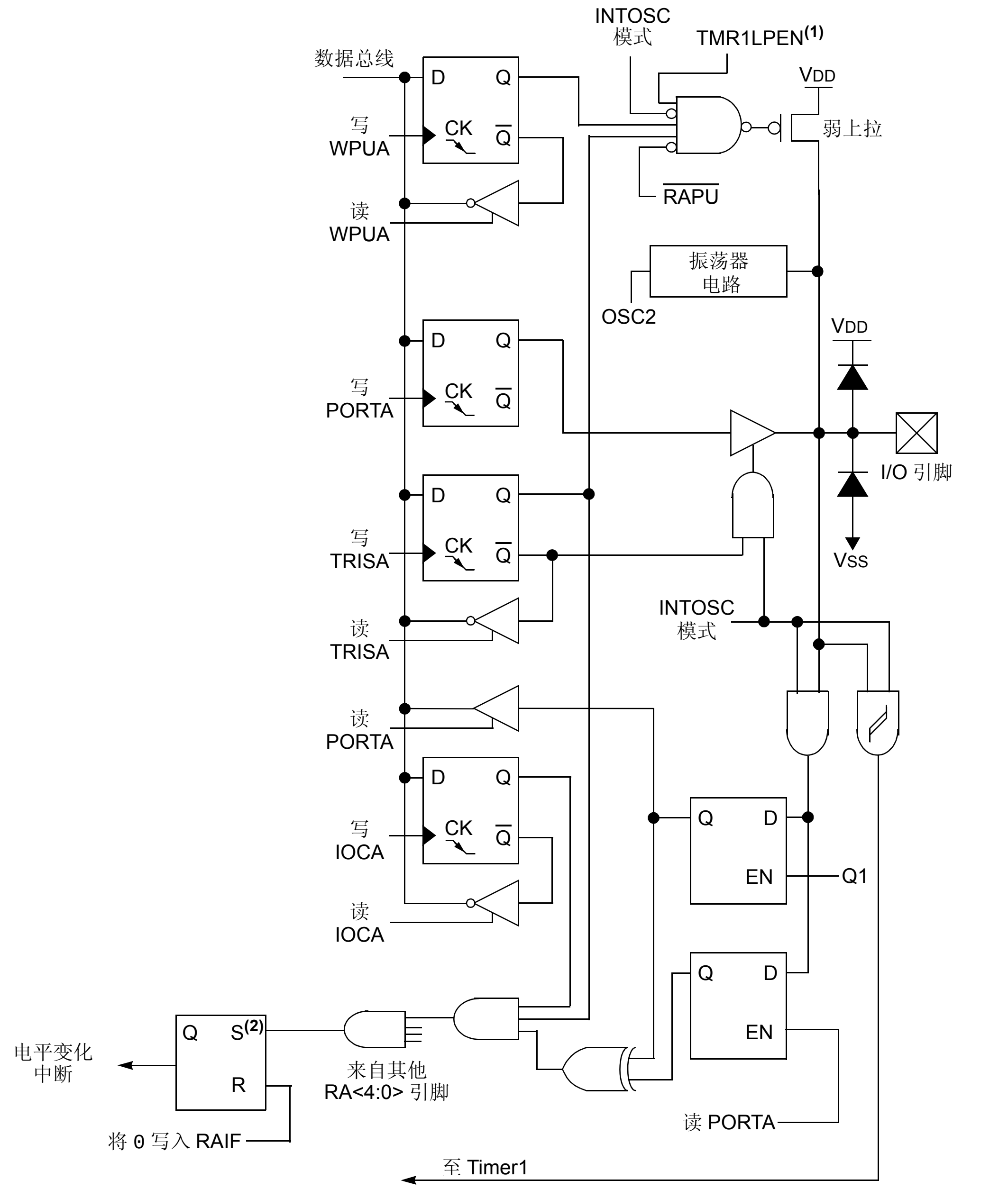
注 1：CLK 模式为 XT、 HS、 LP、 TMR1 LP 以及 CLKOUT 使能。
2：具有 CLKOUT 选项。
3：ANSEL 决定模拟输入模式。
4：置 1 优先于复位。
5：仅限 PIC16F616/16HV616。

4.2.4.6 RA5/T1CKI/OSC1/CLKIN

图 4-5 给出了此引脚的原理图。RA5 引脚可配置为下列功能之一：

- 通用 I/O
- Timer1 的时钟输入
- 晶振 / 谐振器连接
- 时钟输入

图 4-5： RA5 框图



注 1： Timer1 LP 振荡器使能。
2： 置 1 优先于复位。

表 **4-1**： 与 **PORTA** 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0	0000 -000	0000 -000
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0	0000 -000	0000 -000
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
IOCA	—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	--00 0000	--00 0000
OPTION_REG	$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--x0 x000	--u0 u000
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
WPUA	—	—	WPUA5	WPUA4	—	WPUA2	WPUA1	WPUA0	--11 -111	--11 -111

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

4.3 PORTC 和 TRISC 寄存器

PORTC 是通用 I/O 端口，由 6 个双向引脚组成。这些引脚可以配置为数字 I/O 或 A/D 转换器（ADC）/ 比较器的模拟输入。各功能模块（如增强型 CCP 或 ADC）的具体信息，请参见本数据手册中的相关章节。

注：	必须对 ANSEL 寄存器进行初始化，以将模拟通道配置为数字输入。配置为模拟输入的引脚将读为 0，且不能产生中断。

例 4-2： 初始化 PORTC

```
BCF     STATUS,RP0      ;Bank 0
CLRF    PORTC           ;Init PORTC
BSF     STATUS,RP0      ;Bank 1
CLRF    ANSEL           ;digital I/O
MOVLW   0Ch             ;Set RC<3:2> as inputs
MOVWF   TRISC           ;and set RC<5:4,1:0>
                        ;as outputs
BCF     STATUS,RP0      ;Bank 0
```

寄存器 4-6： PORTC：PORTC 寄存器

U-0	U-0	R/W-x	R/W-x	R/W-0	R/W-0	R/W-x	R/W-x
—	—	RC5	RC4	RC3	RC2	RC1	RC0
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

bit 7-6 未实现：读为 0

bit 5-0 **RC<5:0>**：PORTC I/O 引脚位
1 = PORTC 引脚 > V_{IH}
0 = PORTC 引脚 < V_{IL}

寄存器 4-7： TRISC：PORTC 三态寄存器

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
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

bit 7-6 未实现：读为 0

bit 5-0 **TRISC<5:0>**：PORTC 三态控制位
1 = PORTC 引脚配置为输入（三态）
0 = PORTC 引脚配置为输出

4.3.1 RC0/AN4(1)/C2IN+

RC0 引脚可配置为下列功能之一：

- 通用 I/O
- ADC(1) 的模拟输入
- 比较器 C2 的模拟同相输入

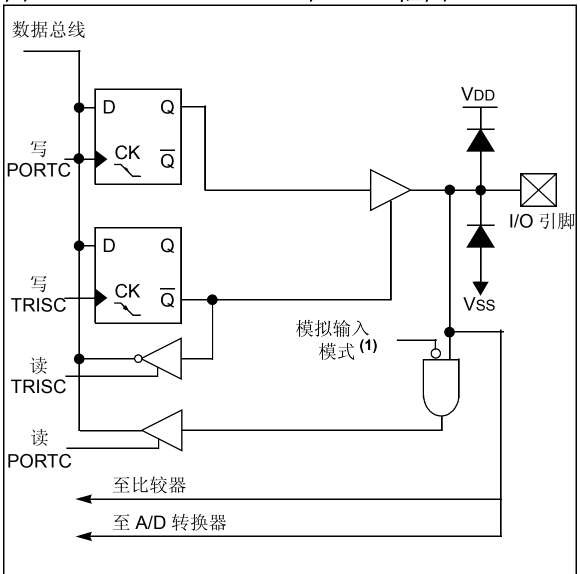
4.3.2 RC1/AN5(1)/C12IN1-

RC1 引脚可配置为下列功能之一：

- 通用 I/O
- ADC(1) 的模拟输入
- 比较器的模拟反相输入

注	1：仅限 PIC16F616/16HV616。

图 4-6： RC0 和 RC1 框图



注 1： ANSEL 或比较器模式决定模拟输入模式。

4.3.3 RC2/AN6(1)/C12IN2-/P1D(1)

RC2 引脚可配置为下列功能之一：

- 通用 I/O
- ADC(1) 的模拟输入
- 比较器 C1 和 C2 的模拟输入
- 增强型 CCP(1) 的数字输出

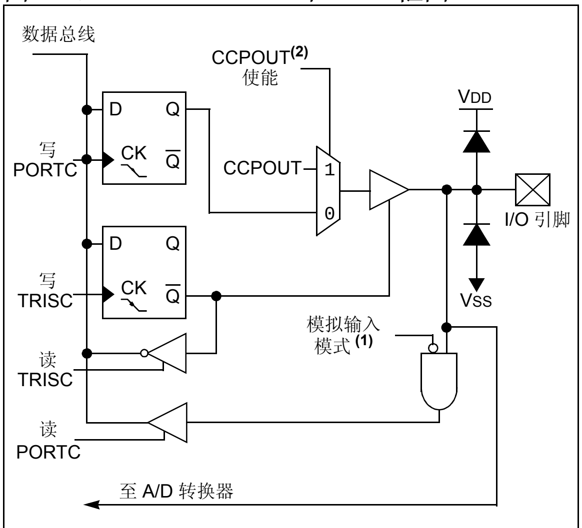
4.3.4 RC3/AN7(1)/C12IN3-/P1C(1)

RC3 引脚可配置为下列功能之一：

- 通用 I/O
- ADC(1) 的模拟输入
- 比较器 C1 和 C2 的模拟反相输入
- 增强型 CCP(1) 的数字输出

注	1：仅限 PIC16F616/16HV616。

图 4-7： RC2 和 RC3 框图



注 1： ANSEL 决定模拟输入模式。
2： 仅限 PIC16F616/16HV616。

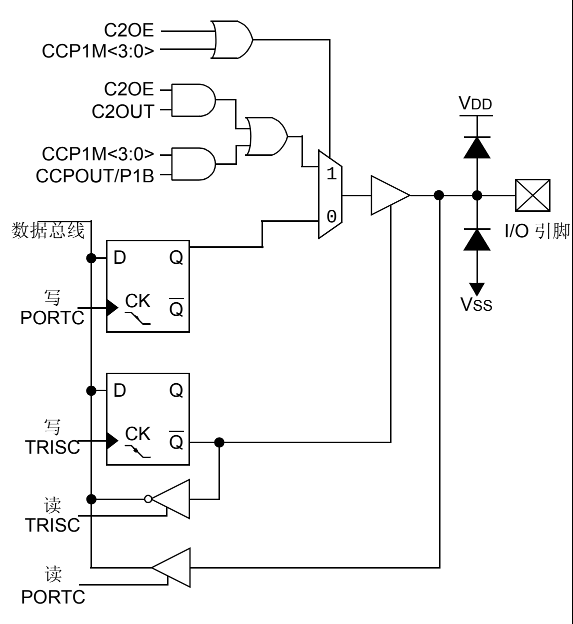
4.3.5 RC4/C2OUT/P1B[1]

RC4 引脚可配置为下列功能之一：

- 通用 I/O
- 比较器 C2 的数字输出
- 增强型 CCP[1] 的数字输出

> 注 1：仅限 PIC16F616/16HV616。
>
> 2：同时使能 C2OUT 和 P1B 将在 RC4 上引起冲突并产生不可预知的结果。因此，如果 C2OUT 被使能，则 ECCP 不能用于半桥或全桥模式下，反之亦然。

图 4-8： RC4 框图



注 1：端口 / 外设选择信号在端口数据和外设输出之间进行选择。

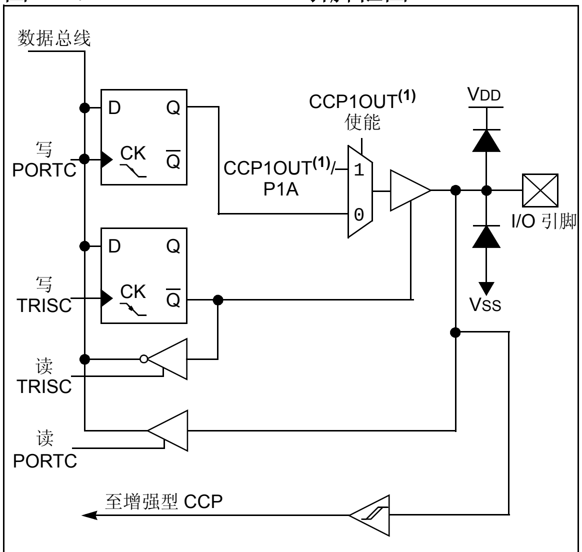
4.3.6 RC5/CCP1[1]/P1A[1]

RC5 引脚可配置为下列功能之一：

- 通用 I/O
- 增强型 CCP[1] 的数字输入 / 输出

> 注 1：仅限 PIC16F616/16HV616。

图 4-9： RC5 引脚框图



注 1：仅限 PIC16F616/16HV616。

表 4-2： 与 PORTC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0	0000 -000	0000 -000
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0	0000 -000	0000 -000
PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx 00xx	--uu 00uu
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111

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

5.0 TIMER0 模块

Timer0 模块是 8 位定时器 / 计数器，具有以下特性：

- 8 位定时器 / 计数器寄存器 （TMR0）
- 8 位预分频器 （与看门狗定时器共用）
- 可编程内部或外部时钟源
- 可编程外部时钟边沿选择
- 溢出时产生中断

图 5-1 给出了 Timer0 模块的框图。

5.1 Timer0 工作原理

当用作定时器时，Timer0 模块可被用作 8 位定时器或 8 位计数器。

5.1.1 8 位定时器模式

当用作定时器时，Timer0 模块在每个指令周期递增（不带预分频器）。通过清零 OPTION 寄存器的 T0CS 位选择定时器模式。

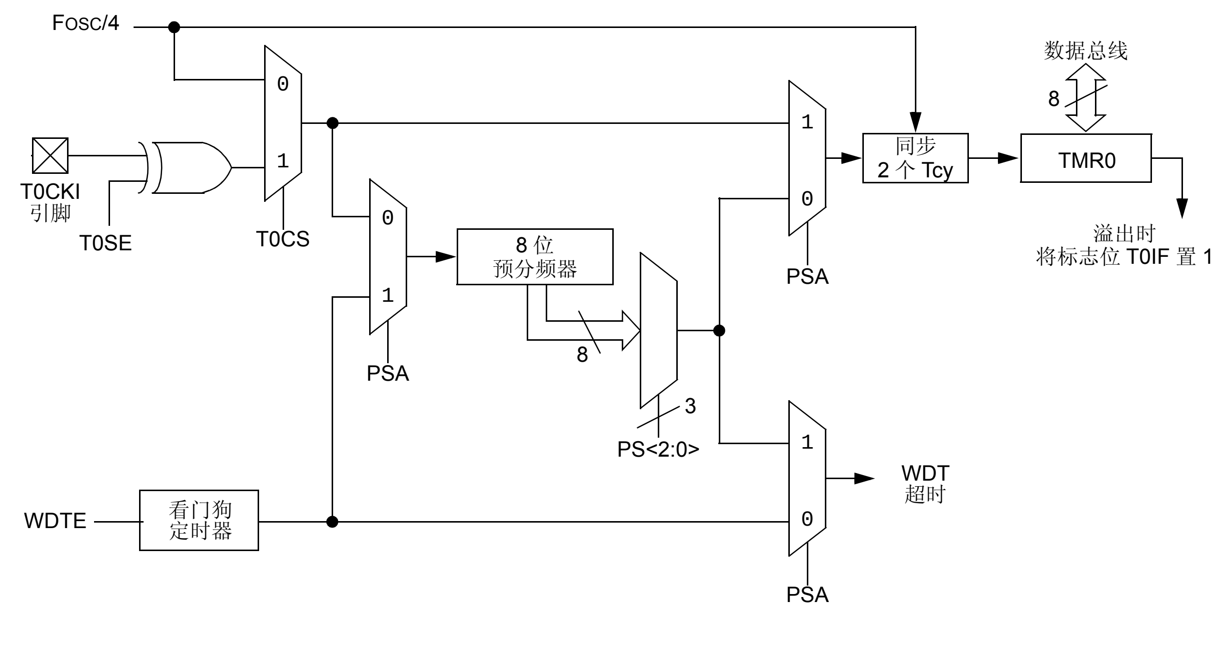
如果对 TMR0 执行写操作，则紧跟写操作之后的两个指令周期内 TMR0 禁止递增。

注：	当写 TMR0 时，考虑到存在两个指令周期的延时，可以调整写入TMR0寄存器的值。

5.1.2 8 位计数器模式

当用作计数器时，Timer0 模块将在 T0CKI 引脚的每个上升或下降沿递增。递增边沿由 OPTION 寄存器的 T0SE 位决定。通过将 OPTION 寄存器的 T0CS 位设置为 1 选择计数器模式。

图 5-1： TIMER0/WDT 预分频器的框图



注 1：T0SE、 T0CS、 PSA 和 PS<2:0> 是 OPTION 寄存器中的位。

2：WDTE 位是配置字寄存器中的位。

5.1.3 可软件编程的预分频器

有一个可软件编程的预分频器可供 Timer0 或看门狗定时器（WDT）使用，但不能同时使用。预分频器的分配由 OPTION 寄存器的 PSA 位控制。要将预分频器分配给 Timer0，必须将 PSA 位清零。

Timer0 模块有 8 个预分频比选项，范围从 1:2 到 1:256。预分频值可通过 OPTION 寄存器的 PS<2:0> 位进行选择。为了对 Timer0 模块使用 1:1 预分频值，必须将预分频器分配给 WDT 模块。

预分频器是不可读写的。将其分配给 Timer0 模块时，所有写入 TMR0 寄存器的指令都会将预分频器清零。

将预分频器分配给 WDT 时， `CLRWDT` 指令会同时将预分频器和 WDT 清零。

5.1.3.1 在 Timer0 和 WDT 模块之间切换预分频器

由于将预分频器分配给 Timer0 或 WDT，在切换预分频值时，可能会产生意外的器件复位。当把预分频器从 Timer0 分配给 WDT 模块时，必须执行如例 5-1 中所示的指令序列。

例 5-1： 改变预分频器的分配（**TIMER0 → WDT**）

```
BANKSEL  TMR0           ;
CLRWDT                  ;Clear WDT
CLRF     TMR0           ;Clear TMR0 and
                        ;prescaler
BANKSEL  OPTION_REG     ;
BSF      OPTION_REG,PSA ;Select WDT
CLRWDT          ;
                ;
MOVLW   b'11111000'     ;Mask prescaler
ANDWF   OPTION_REG,W    ;bits
IORLW   b'00000101'     ;Set WDT prescaler
MOVWF   OPTION_REG      ;to 1:32
```

将预分频器从 WDT 分配给 Timer0 模块时，必须执行以下指令序列（例 5-2）。

例 5-2： 改变预分频器的分配（**WDT → TIMER0**）

```
CLRWDT                  ;Clear WDT and
                        ;prescaler
BANKSEL OPTION_REG      ;
MOVLW   b'11110000'     ;Mask TMR0 select and
ANDWF   OPTION_REG,W    ;prescaler bits
IORLW   b'00000011'     ;Set prescale to 1:16
MOVWF   OPTION_REG      ;
```

5.1.4 TIMER0 中断

TMR0 寄存器从 FFh 溢出到 00h 时，将产生 Timer0 中断。每次 TMR0 寄存器溢出时都会将 INTCON 寄存器的 T0IF 中断标志位置 1，与是否允许了 Timer0 中断无关。 T0IF 位必须用软件清零。 Timer0 中断允许位是 INTCON 寄存器的 T0IE 位。

注：	由于定时器在休眠状态下是冻结的，所以 Timer0 中断无法将处理器从休眠状态唤醒。

5.1.5 TIMER0 与外部时钟配合使用

当 Timer0 处于计数器模式下时，通过对内部相位时钟 Q2 和 Q4 周期的预分频输出进行采样，可实现 T0CKI 输入与 Timer0 寄存器的同步。因此，外部时钟源的高低电平周期必须满足**第 15.0 节 “电气规范”**中所示的时序要求。

寄存器 5-1： OPTION_REG：OPTION 寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
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

bit 7 **$\overline{\text{RAPU}}$**：PORTA 上拉使能位
1 = 禁止 PORTA 上拉
0 = 通过各端口锁存值使能 PORTA 上拉

bit 6 **INTEDG**：中断触发边沿选择位
1 = INT 引脚的上升沿触发中断
0 = INT 引脚的下降沿触发中断

bit 5 **T0CS**：TMR0 时钟源选择位
1 = T0CKI 引脚上的电平跳变
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 引脚信号从高至低跳变时，递增计数
0 = 在 T0CKI 引脚信号从低至高跳变时，递增计数

bit 3 **PSA**：预分频器分配控制位
1 = 预分频器分配给 WDT
0 = 预分频器分配给 Timer0 模块

bit 2-0 **PS<2:0>**：预分频比选择位

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

表 5-1： 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	POR/BOR 时的值	所有其他复位时的值
TMR0	Timer0 模块寄存器								xxxx xxxx	uuuu uuuu
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
OPTION_REG	$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111

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

注：

6.0 带门控的 TIMER1 模块

Timer1 模块是 16 位定时器 / 计数器，具有以下特性：

- 16 位定时器 / 计数器定时器对 （TMR1H:TMR1L）
- 可编程内部或外部时钟源
- 3 位预分频器
- 可选的 LP 振荡器
- 同步或异步工作
- 门控源来自比较器或 $\overline{\text{T1G}}$ 引脚的 Timer1 门控（计数使能）
- 溢出时产生中断
- 溢出触发唤醒 （仅限外部时钟，异步模式）
- 捕捉 / 比较功能的时基
- 特殊事件触发器 （带有 ECCP）
- 比较器输出与 Timer1 时钟同步

图 6-1 给出了 Timer1 模块的框图。

6.1 Timer1 工作原理

Timer1模块是16位递增计数器，可通过TMR1H:TMR1L 寄存器对访问。写 TMR1H 或 TMR1L 会直接更新计数器。

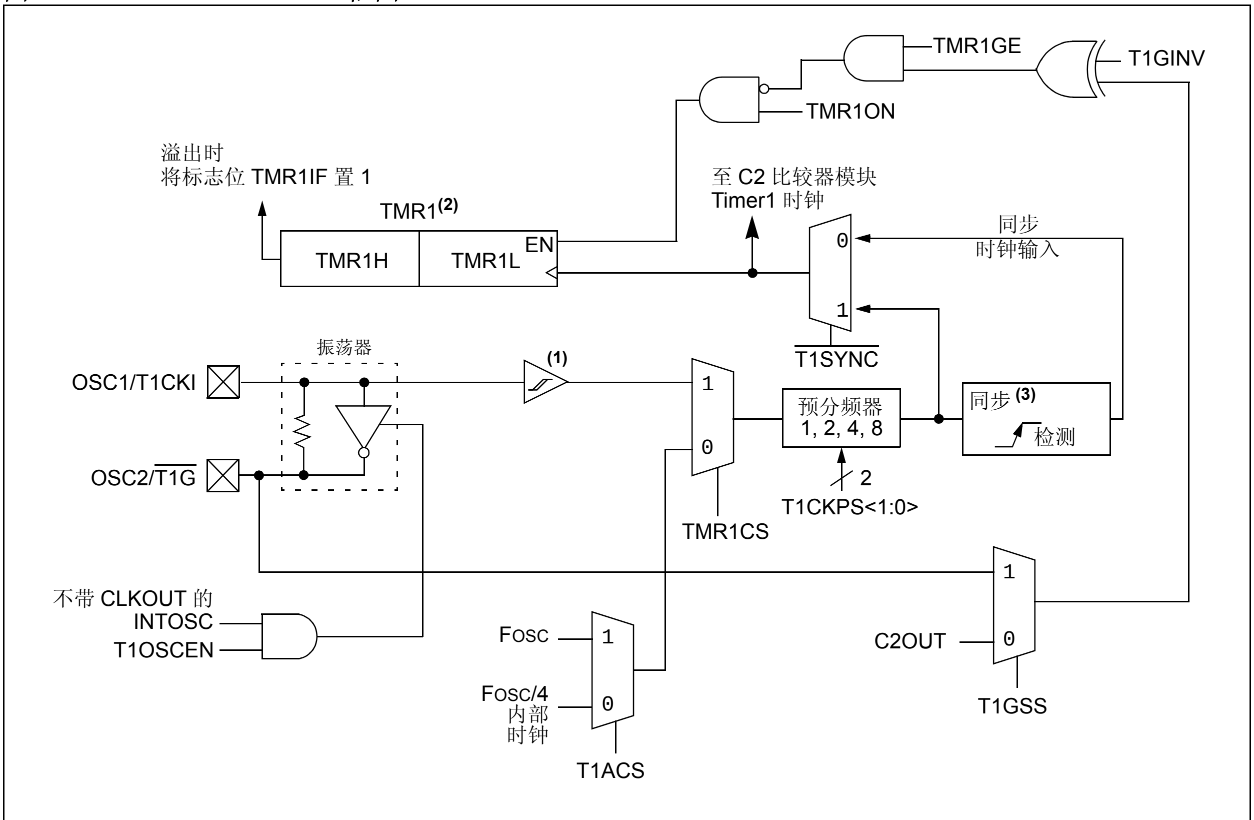
当使用内部时钟源时，模块为定时器；当使用外部时钟源时，模块既可用作定时器，也可用作计数器。

6.2 时钟源选择

T1CON 寄存器的 TMR1CS 位用于选择时钟源。当 TMR1CS = 0 时，时钟源为 FOSC/4。当 TMR1CS = 1 时，时钟源由外部提供。

时钟源	TMR1CS	T1ACS
FOSC/4	0	0
FOSC	0	1
T1CKI 引脚	1	x

图 6-1： TIMER1 框图



注 1： 在使用 LP 振荡器时，ST 缓冲器是低功耗型；在使用 T1CKI 时，ST 缓冲器是高速型。
2： Timer1 寄存器在上升沿递增。
3： 处于休眠模式时同步不工作。

6.2.1 内部时钟源

当选择内部时钟源时， TMR1H:TMR1L 寄存器对将在 TCY 的整数倍 （取决于 Timer1 预分频器）处递增。

6.2.2 外部时钟源

当选择外部时钟源时，Timer1 模块可以作为定时器或计数器工作。

计数时，Timer1 在外部时钟输入 T1CKI 的上升沿递增。此外，计数器模式时钟可以与单片机系统时钟同步，也可以异步工作。

如果需要使用外部时钟振荡器（并且单片机正在使用不带 CLKOUT 的 INTOSC），Timer1 可以将 LP 振荡器用作时钟源。

注：	在计数器模式下，在第一个要递增计数的上升沿之前，计数器应先记录一个下降沿。

6.3 Timer1 预分频器

Timer1 有 4 种预分频比选择，允许对时钟输入进行 1、2、4 或 8 分频。T1CON 寄存器的 T1CKPS 位控制预分频计数器。对预分频计数器不能直接进行读写操作；但是，通过写入TMR1H或TMR1L可将预分频计数器清零。

6.4 Timer1 振荡器

在引脚 OSC1（输入）和 OSC2（输出）之间接有一个内置低功耗 32.768 kHz 晶体振荡器。通过将 T1CON 寄存器的 T1OSCEN 控制位置 1 可使能该振荡器。在休眠期间，振荡器将继续工作。

Timer1 振荡器与系统 LP 振荡器共用。因此，只有当主系统时钟来自于内部振荡器或振荡器在 LP 振荡器模式下时，Timer1 才能采用该模式。用户必须提供软件延时以确保振荡器的正常起振。

当 Timer1 振荡器被使能时，TRISA5 和 TRISA4 位被置 1。RA5和RA4位读为0，而TRISA5和TRISA4位读为1。

注：	振荡器在使用之前需要一定的起振和稳定时间。因此，T1OSCEN 应置 1，且在使能 Timer1 之前确保有适当的延时。

6.5 Timer1 在异步计数器模式下的工作原理

如果 T1CON 寄存器的控制位 $\overline{\text{T1SYNC}}$ 置 1，外部时钟输入将不同步。定时器继续异步于内部相位时钟进行递增计数。在休眠期间定时器将继续运行，并在溢出时产生中断，溢出中断将唤醒处理器。但是，用软件对定时器进行读 / 写操作时，要特别当心 （见第 **6.5.1** 节 **“在异步计数器模式下读写 Timer1”**）。

注：	当从同步切换到异步操作时，可能会跳过一次递增。当从异步切换到同步操作时，可能会产生一次额外递增。

6.5.1 在异步计数器模式下读写 TIMER1

当定时器采用外部异步时钟工作时，对 TMR1H 或 TMR1L 的读操作将确保有效（由硬件实现）。但是，应该注意的是，用两个 8 位值来读取 16 位定时器本身就会产生某些问题，这是因为定时器可能在两次读操作之间产生溢出。

对于写操作，建议用户直接停止计数器，然后写入需要的值。如果定时器寄存器正进行递增计数，对定时器寄存器进行写操作，可能会导致写入竞争，从而可能在 TMR1H:TMR1L 寄存器对中产生不可预测的值。

6.6 Timer1 门控

Timer1 门控源可用软件配置为 $\overline{\text{T1G}}$ 引脚或比较器 C2 的输出。这使得器件能够使用 $\overline{\text{T1G}}$ 直接对外部事件进行计时，或者使用比较器 C2 直接对模拟事件进行计时。Timer1 门控源的选择，请参见 CM2CON1 寄存器 （寄存器 8-3）。这个特性可以简化 Δ–Σ A/D 转换器和许多其他应用的程序。 Δ–Σ A/D 转换器的更多信息，请参见 Microchip 网站 （www.microchip.com）。

注：	要将 $\overline{\text{T1G}}$ 或 C2OUT 用作 Timer1 门控源，必须将 T1CON 寄存器的 TMR1GE 位置 1。选择 Timer1 门控源的更多信息，请参见 CM2CON1 寄存器 （寄存器 8-3）。

可使用 T1CON 寄存器的 T1GINV 位将 Timer1 门控反相，无论门控源来自 $\overline{\text{T1G}}$ 引脚还是比较器 C2 的输出。这样可将 Timer1 配置为测量事件之间的有效高电平或有效低电平时间。

初稿

6.7 Timer1 中断

Timer1 寄存器对（TMR1H:TMR1L）递增到 FFFFh，然后返回到 0000h。当 Timer1 计满返回时，PIR1 寄存器的 Timer1 中断标志位将置 1。为允许计满返回时的中断，必须将以下位置 1：

- PIE1 寄存器的 TMR1IE 位
- INTCON 寄存器的 PEIE 位
- INTCON 寄存器的 GIE 位
- T1CON 寄存器的 $\overline{\text{T1SYNC}}$ 位
- T1CON 寄存器的 TMR1CS 位
- T1CON 寄存器的 T1OSCEN 位（可以被置 1）

在中断服务程序中将 TMR1IF 位清零将清除中断。

注：	在重新允许中断前，应将 TMR1H:TTMR1L 寄存器对以及 TMR1IF 位清零。

6.8 Timer1 在休眠模式下的工作原理

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

- 必须将 T1CON 寄存器的 TMR1ON 位置 1
- 必须将 PIE1 寄存器的 TMR1IE 位置 1
- 必须将 INTCON 寄存器的 PEIE 位置 1

器件将在溢出时被唤醒并执行下一条指令。如果将 INTCON 寄存器的 GIE 位置 1，器件将调用中断服务程序（0004h）。

6.9 ECCP 捕捉 / 比较时基（仅限 PIC16F616/16HV616）

当工作在捕捉或比较模式下时，ECCP 模块使用 TMR1H:TMR1L 寄存器对作为时基。

在捕捉模式下，当发生配置的事件时，TMR1H:TMR1L 寄存器对中的值被复制到 CCPR1H:CCPR1L 寄存器对中。

在比较模式下，当 CCPR1H:CCPR1L 寄存器对中的值与 TMR1H:TMR1L 寄存器对中的值相匹配时触发事件。该事件可以是特殊事件触发信号。

更多信息，请参见**第 10.0 节“增强型捕捉 / 比较 /PWM（带自动关闭和死区）模块（仅限 PIC16F616/16HV616）”**。

6.10 ECCP 特殊事件触发器（仅限 PIC16F616/16HV616）

当将 ECCP 配置为触发特殊事件时，触发器将清零 TMR1H:TMR1L 寄存器对。该特殊事件不会引起 Timer1 中断。ECCP 模块仍然可以配置为产生 ECCP 中断。

在该工作模式下，CCPR1H:CCPR1L 寄存器对有效地成为 Timer1 的周期寄存器。

要利用特殊事件触发信号，应使 Timer1 与 FOSC 同步。Timer1 的异步操作会导致错过特殊事件触发信号。

如果对 TMR1H 或 TMR1L 的写操作和来自 ECCP 的特殊事件触发信号同时发生，则写操作优先。

更多信息，请参见**第 10.2.4 节“特殊事件触发器”**。

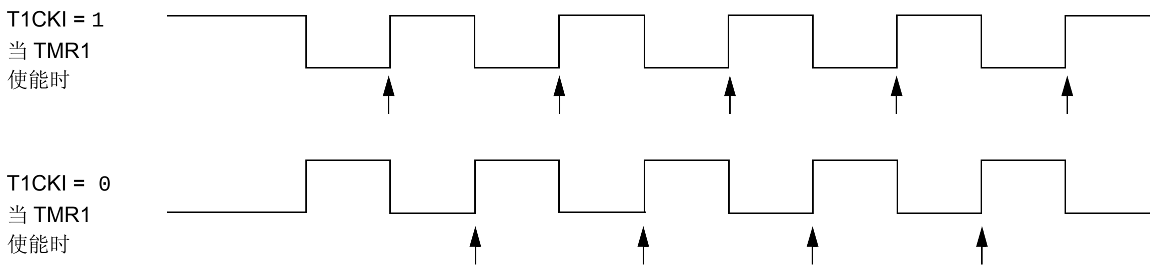
6.11 比较器同步

用于递增 Timer1 的时钟也可以用于同步比较器输出。比较器模块中使能了此功能。

当使用比较器作为 Timer1 门控时，应将比较器输出与 Timer1 保持同步。这可以确保 Timer1 在比较器发生变化时不会错过递增。

更多信息，请参见**第 8.8.2 节“使比较器 C2 的输出与 Timer1 同步”**。

图 6-2： TIMER1 递增边沿



注 1： 箭头表明计数器递增。
2： 在计数器模式下，在时钟要进行递增的第一个上升沿之前，计数器应先记录一个下降沿。

6.12 Timer1 控制寄存器

Timer1 控制寄存器（T1CON）用于控制 Timer1 以及选择 Timer1 模块的各种特性，如寄存器 6-1 中所示。

寄存器 6-1： T1CON：TIMER1 控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T1GINV(1)	TMR1GE(2)	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
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

bit 7 **T1GINV**：Timer1 门控反相位 (1)
1 = Timer1 门控高电平有效（当门控为高电平时 Timer1 计数）
0 = Timer1 门控低电平有效（当门控为低电平时 Timer1 计数）

bit 6 **TMR1GE**：Timer1 门控使能位 (2)
如果 TMR1ON = 0：
该位为无关位
如果 TMR1ON = 1：
1 = Timer1 计数由 Timer1 门控功能控制
0 = Timer1 始终保持计数

bit 5-4 **T1CKPS<1:0>**：Timer1 输入时钟预分频比选择位
11 = 1:8 预分频比
10 = 1:4 预分频比
01 = 1:2 预分频比
00 = 1:1 预分频比

bit 3 **T1OSCEN**：LP 振荡器使能控制位
如果不带 CLKOUT 的 INTOSC 处于有效状态：
1 = LP 振荡器使能作为 Timer1 的时钟
0 = LP 振荡器关闭
否则：
该位为无关位

bit 2 **$\overline{\text{T1SYNC}}$**：Timer1 外部时钟输入同步控制位
TMR1CS = 1：
1 = 不同步外部时钟输入
0 = 同步外部时钟输入
TMR1CS = 0：
该位为无关位。 Timer1 使用内部时钟

bit 1 **TMR1CS**：Timer1 时钟源选择位
1 = 使用来自 T1CKI 引脚（在上升沿）的外部时钟
0 = 内部时钟
如果 TMR1ACS = 0：
FOSC/4
如果 TMR1ACS = 1：
FOSC

bit 0 **TMR1ON**：Timer1 使能位
1 = 使能 Timer1
0 = 停止 Timer1

注 1： T1GINV 位将 Timer1 门控逻辑反相，无论其来源为何。
2： 要将 $\overline{\text{T1G}}$ 引脚或 C2OUT（通过 CM2CON1 寄存器的 T1GSS 位在二者之间进行选择）用作 Timer1 门控源，必须将 TMR1GE 位置 1。

表 **6-1**： 与 **TIMER1** 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0	0000 -000	0000 -000
CM2CON1	MC1OUT	MC2OUT	—	T1ACS	C1HYS	C2HYS	T1GSS	C2SYNC	00-0 0010	00-0 0010
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
PIE1	—	ADIE(1)	CCP1IE(1)	C2IE	C1IE	—	TMR2IE(1)	TMR1IE	-000 0-00	-000 0-00
PIR1	—	ADIF(1)	CCP1IF(1)	C2IF	C1IF	—	TMR2IF(1)	TMR1IF	-000 0-00	-000 0-00
TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								xxxx xxxx	uuuu uuuu
TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								xxxx xxxx	uuuu uuuu
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	uuuu uuuu

图注： x = 未知， u = 不变， — = 未实现 （读为 0）。 Timer1 模块不使用阴影单元。
注 1： 仅限 PIC16F616/16HV616。

注：

7.0 TIMER2 模块（仅限 PIC16F616/16HV616）

Timer2 模块是 8 位定时器，具有以下特性：

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

Timer2 框图请参见图 7-1。

7.1 Timer2 工作原理

Timer2 模块的时钟输入是系统指令时钟（Fosc/4）。时钟送入 Timer2 预分频器， Timer2 预分频器提供 1:1、1:4 或 1:16 三种预分频比选择。然后预分频器的输出用于递增 TMR2 寄存器。

TMR2 的值与 PR2 的值不断进行比较以确定何时匹配。TMR2 将从 00h 开始递增，直到与 PR2 中的值匹配。当匹配发生时，会发生以下两件事：

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

Timer2/PR2 比较器的匹配输出被送入 Timer2 后分频器。后分频器具有 1:1 到 1:16 的后分频比选择。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 寄存器均可完全读写。在任何复位时，TMR2 寄存器被设置为 00h，PR2 寄存器被设置为 FFh。

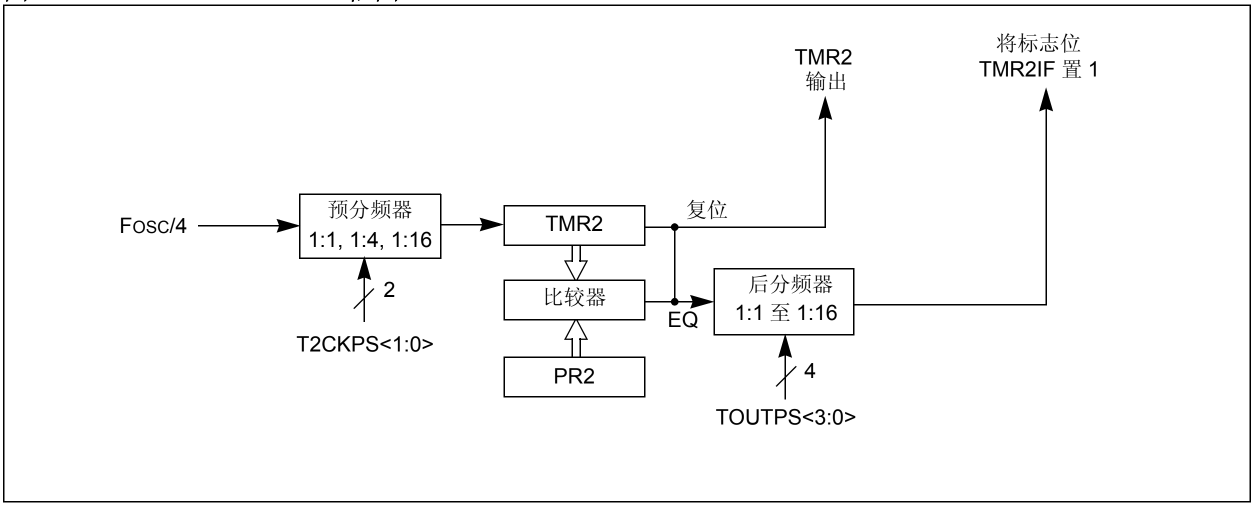
通过将 T2CON 寄存器中 TMR2ON 位设置为 1 可使能 Timer2。通过将 TMR2ON 位设置为 0 可关闭 Timer2。

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。预分频和后分频计数器均会在发生以下事件时清零：

- 对 TMR2 进行写操作。
- 对 T2CON 进行写操作。
- 发生任何器件复位 （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 7-1： TIMER2 框图



寄存器 7-1： T2CON：TIMER2 控制寄存器

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

bit 7　未实现：读为 0

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

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

bit 1-0　**T2CKPS<1:0>**：Timer2 时钟预分频比选择位
00 = 预分频比为 1
01 = 预分频比为 4
1x = 预分频比为 16

表 7-1： 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	POR/BOR 时的值	所有其他复位时的值
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
PIE1	—	ADIE(1)	CCP1IE(1)	C2IE	C1IE	—	TMR2IE(1)	TMR1IE	-000 0-00	-000 0-00
PIR1	—	ADIF(1)	CCP1IF(1)	C2IF	C1IF	—	TMR2IF(1)	TMR1IF	-000 0-00	-000 0-00
PR2	Timer2 模块周期寄存器								1111 1111	1111 1111
TMR2	8 位 TMR2 寄存器的保持寄存器								0000 0000	0000 0000
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000

图注： x = 未知， u = 不变， — = 未实现 （读为 0）。 Timer2 模块不使用阴影单元。
注 1： 仅限 PIC16F616/16HV616。

8.0 比较器模块

比较器模块通过比较两个模拟电压并提供其相对幅值的数字表示，用于建立模拟电路与数字电路的接口。比较器是非常有用的混合信号模块，因为它提供了与器件相独立的模拟功能。模拟比较器模块包括以下特性：

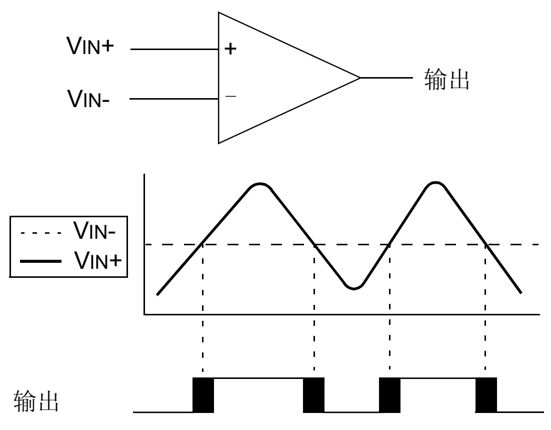
- 独立的比较器控制
- 可编程输入选择
- 比较器输出内部 / 外部有效
- 可编程输出极性
- 电平变化中断
- 从休眠状态唤醒
- PWM 关闭
- Timer1 门控 （计数使能）
- 输出与 Timer1 时钟输入同步
- SR 锁存器
- 可编程和固定参考电压
- 用户可使能的比较器滞后

注：	只有比较器 C2 可连接到 Timer1。

8.1 比较器概述

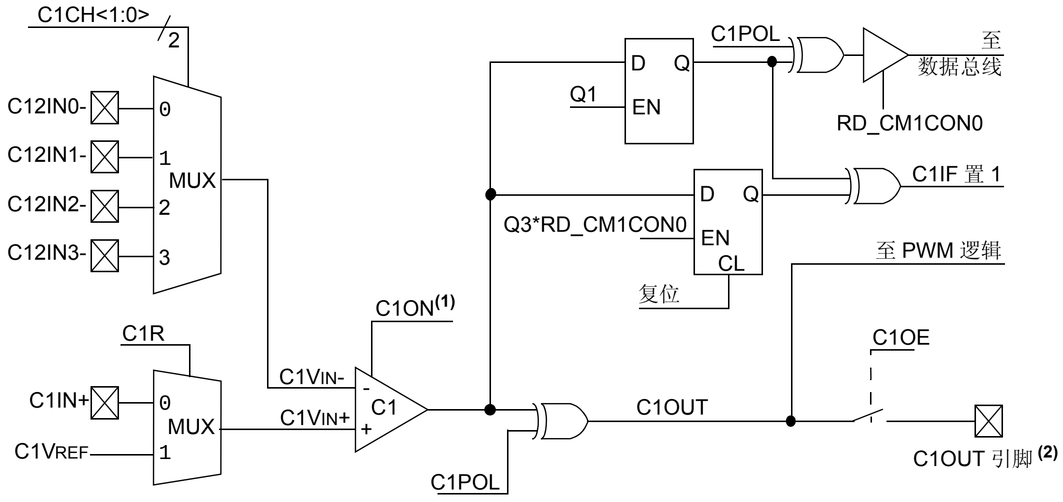
图 8-1 所示为单比较器以及模拟输入电平与数字输出之间的关系。当 VIN+ 上的模拟电压小于 VIN- 上的模拟电压值时，比较器输出为数字低电平。当 VIN+ 上的模拟电压大于 VIN- 上的模拟电压值时，比较器输出为数字高电平。

图 8-1： 单比较器



注： 比较器输出的黑色区域表示因输入失调电压和响应时间所造成的输出不确定区域。

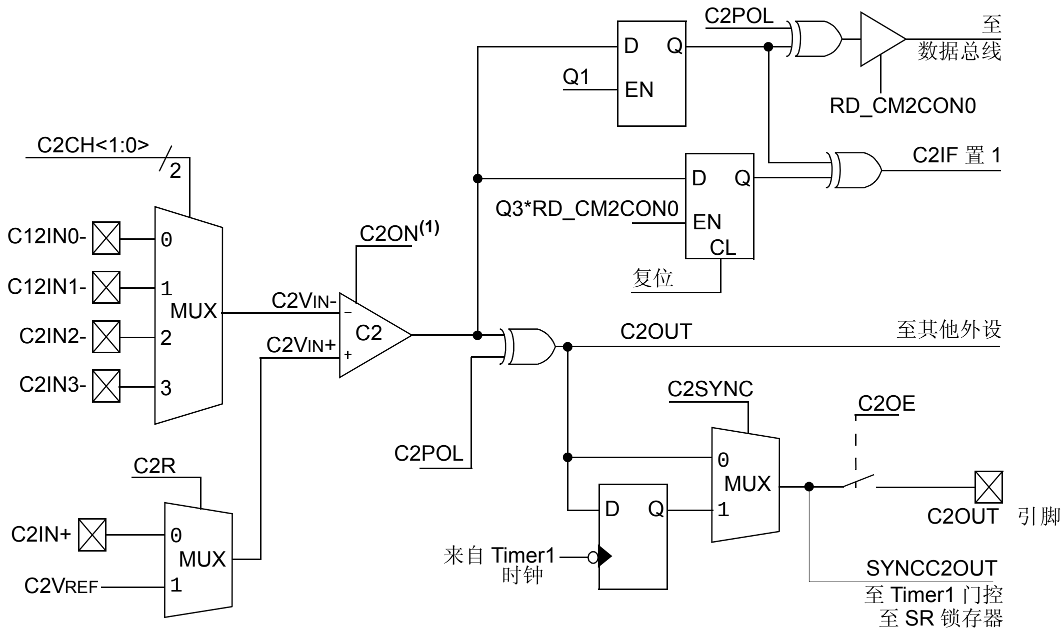
图 8-2： 比较器 C1 的简化框图



注 1： 当 C1ON = 0 时， C1 比较器将输出 0 到异或门。

2： 图中的输出仅供参考。更多详细信息，请参见 I/O 端口引脚框图。

图 8-3： 比较器 C2 的简化框图



注 1： 当 C2ON = 0 时， C2 比较器将输出 0 到异或门。

2： 图中的输出仅供参考。更多详细信息，请参见 I/O 端口引脚框图。

 初稿

8.2 比较器控制

每个比较器都有独立的控制和配置寄存器：比较器 C1 为 CM1CON0，比较器 C2 为 CM2CON0。此外，比较器 C2 还有另外一个控制寄存器 CM2CON1，用于控制与 Timer1 的交互和两个比较器输出的同时读取。

CM1CON0 和 CM2CON0 寄存器（分别见寄存器 8-1 和 8-2）包含以下控制和状态位：

- 使能
- 输入选择
- 参考电压选择
- 输出选择
- 输出极性

8.2.1 比较器使能

将 CMxCON0 寄存器的 CxON 位置 1 可以使能比较器操作。清零 CxON 位可以禁止比较器，以使电流消耗降至最低。

8.2.2 比较器输入选择

CMxCON0 寄存器的 CxCH<1:0> 位指示将 4 个模拟输入引脚之一连接到比较器的反相输入端。

注：	要将 CxIN+ 和 CxIN- 引脚用作模拟输入，必须将 ANSEL 寄存器中相应的位置 1，同时也必须置 1 相应的 TRIS 位来禁止输出驱动器。

8.2.3 比较器参考电压选择

通过设置 CMxCON0 寄存器的 CxR 位，将内部参考电压或模拟输入引脚连接到比较器的同相输入。关于内部参考电压模块的更多信息，请参见**第 8.11 节“比较器参考电压”**。

8.2.4 比较器输出选择

可以通过读 CMxCON0 寄存器的 CxOUT 位或 CM2CON1 寄存器的 MCxOUT 位监视比较器的输出。为了使输出可用于外部连接，必须满足以下条件：

- 必须将 CMxCON0 寄存器的 CxOE 位置 1
- 必须清零相应的 TRIS 位
- 必须将 CMxCON0 寄存器的 CxON 位置 1

注	
1：	CxOE 位改写端口数据锁存器。将 CxON 置 1 对端口改写没有影响。
2：	比较器的内部输出在每个指令周期被锁存。除非另外指定，否则不锁存外部输出。

8.2.5 比较器输出极性

反相比较器的输出在功能上等同于交换比较器输入。可以通过将 CMxCON0 寄存器的 CxPOL 位置 1 来使比较器输出的极性反相。清零 CxPOL 位导致非反相输出。

表 8-1 给出了输出状态与输入条件的关系（包括极性控制）。

表 8-1： 比较器输出状态与输入条件

输入条件	CxPOL	CxOUT
CxVIN- > CxVIN+	0	0
CxVIN- < CxVIN+	0	1
CxVIN- > CxVIN+	1	1
CxVIN- < CxVIN+	1	0

8.3 比较器响应时间

在改变输入源或选择新的参考电压后，比较器输出要经过一段不确定状态的时间。这段时间被称为响应时间。比较器的响应时间不同于参考电压的稳定时间。因此，在确定比较器输入改变的总响应时间时，必须考虑这两个时间。更多详细信息，请参见**第 15.0 节“电气规范”**中的比较器和参考电压规范。

8.4 比较器中断操作

只要比较器的输出值发生变化，相应的比较器中断标志位就会置 1。比较器输出值的变化由不匹配电路识别，该电路由两个锁存器和一个异或门组成 （见图 8-2 和图 8-3）。当读 CMxCON0 寄存器时，一个锁存器更新为比较器的输出电平。该锁存器将保持该值，直到下次读 CMxCON0 寄存器，或者发生复位。不匹配电路的另一个锁存器在每个 Q1 系统时钟更新。当比较器的输出变化在 Q1 时钟周期输入到第二个锁存器时，即产生不匹配条件。这时两个不匹配锁存器具有相反的输出电平，由异或门检测并被送到中断电路。不匹配条件持续直到读CMxCON0寄存器或者比较器输出返回先前的状态。

注	**1：** 对 CMxCON0 寄存器的写操作也将清除不匹配条件，因为所有写操作的写周期开始都包含一个读操作。 **2：** 比较器中断的正常工作与 CxOE 的状态无关。

比较器中断是根据不匹配边沿而不是不匹配电平置 1 的。这意味着不需要额外的读取或写入 CMxCON0 寄存器来将不匹配寄存器清零这一步骤，即可将中断标志复位。当不匹配寄存器清零时，比较器输出返回前一个状态将发生中断，否则不会发生中断。

需要用软件保存比较器输出状态的信息（从 CMxCON0 寄存器或 CM2CON1 寄存器读取），以确定实际发生的变化。

PIR1 寄存器的 CxIF 位是比较器中断标志。必须用软件将该位清零以将其复位。由于可以对该寄存器写入 1，因此可产生中断。

必须将 PIE1 寄存器的 CxIE 位以及 INTCON 寄存器的 PEIE 和 GIE 位都置 1 以允许比较器中断。如果上述位中有任何一位被清零，则无法允许中断，尽管中断条件发生时仍会将 PIR1 寄存器的 CxIF 位置 1。

图 8-4： 比较器中断时序 （不带 **CMxCON0** 读取）

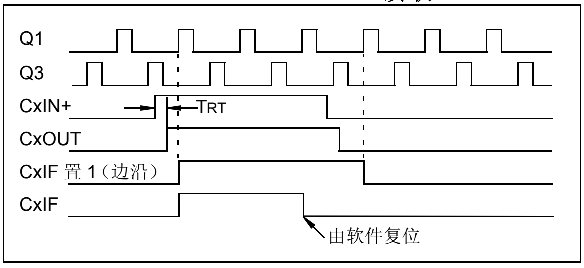
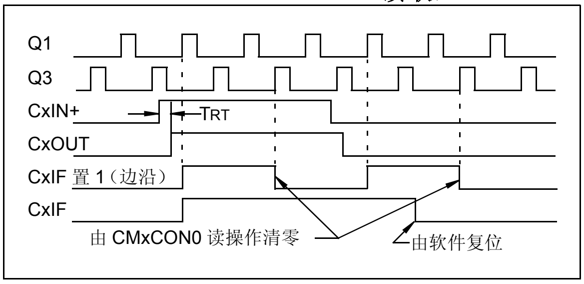


图 8-5： 比较器中断时序 （带 **CMxCON0** 读取）



注	**1：** 在读操作执行过程中 （Q2 周期的起始时刻），如果 CMxCON0 寄存器（CxOUT）发生变化，那么 PIR1 寄存器的 CxIF 中断标志位可能不会被置 1。 **2：** 当两个比较器之一先使能时，比较器模块中的偏置电路在稳定前可能产生比较器的无效输出。应允许偏置电路有 1 μs 的稳定时间，然后在允许比较器中断前将不匹配条件和中断标志清零。

8.5 休眠模式下的工作原理

如果在进入休眠模式之前比较器使能，则它在休眠期间仍将处于运行状态。**第 15.0 节“电气规范”**中单独给出了比较器消耗的额外电流。如果不使用比较器来唤醒器件，则在休眠模式下可通过关闭比较器使功耗降至最低。可以通过清零 CMxCON0 寄存器的 CxON 位来关闭每个比较器。

比较器输出变化可以将器件从休眠状态唤醒。要使比较器能将器件从休眠状态唤醒，必须将 PIE1 寄存器的 CxIE 位和 INTCON 寄存器的 PEIE 位置 1。紧跟 `Sleep` 指令后的指令总是在从休眠状态唤醒之后执行。如果也将 INTCON 寄存器的 GIE 位置 1，则器件将执行中断服务程序。

8.6 复位的影响

器件复位强制 CMxCON0 和 CM2CON1 寄存器为复位状态。这使比较器和参考电压被强制为 “关闭”状态。

寄存器 8-1： CM1CON0：比较器 1 的控制寄存器 0

R/W-0	R-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0
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

bit 7 **C1ON**：比较器 C1 使能位
1 = 比较器 C1 使能
0 = 比较器 C1 禁止

bit 6 **C1OUT**：比较器 C1 输出位
如果 C1POL = 1 （极性反相）：
当 $C1V_{IN}+ > C1V_{IN}-$ 时， C1OUT = 0
当 $C1V_{IN}+ < C1V_{IN}-$ 时， C1OUT = 1
如果 C1POL = 0 （极性非反相）：
当 $C1V_{IN}+ > C1V_{IN}-$ 时， C1OUT = 1
当 $C1V_{IN}+ < C1V_{IN}-$ 时， C1OUT = 0

bit 5 **C1OE**：比较器 C1 输出使能位
1 = C1OUT 出现在 C1OUT 引脚上 (1)
0 = C1OUT 仅在内部有效

bit 4 **C1POL**：比较器 C1 输出极性选择位
1 = C1OUT 逻辑反相
0 = C1OUT 逻辑不反相

bit 3 **未实现**：读为 0

bit 2 **C1R**：比较器 C1 参考电压选择位 （同相输入）
1 = $C1V_{IN}+$ 连接至 $C1V_{REF}$ 输出
0 = $C1V_{IN}+$ 连接至 C1IN+ 引脚

bit 1-0 **C1CH<1:0>**：比较器 C1 通道选择位
00 = C1 的 C12IN0- 引脚连接至 $C1V_{IN}-$
01 = C1 的 C12IN1- 引脚连接至 $C1V_{IN}-$
10 = C1 的 C12IN2- 引脚连接至 $C1V_{IN}-$
11 = C1 的 C12IN3- 引脚连接至 $C1V_{IN}-$

注 1： 比较器输出需要以下 3 个条件：C1OE = 1， C1ON = 1 且相应的端口 TRIS 位 = 0。

寄存器 8-2： CM2CON0：比较器 2 的控制寄存器 0

R/W-0	R-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0
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

bit 7 **C2ON**：比较器 C2 使能位
1 = 比较器 C2 使能
0 = 比较器 C2 禁止

bit 6 **C2OUT**：比较器 C2 输出位
如果 C2POL = 1 （极性反相）：
当 C2VIN+ > C2VIN- 时， C2OUT = 0
当 C2VIN+ < C2VIN- 时， C2OUT = 1
如果 C2POL = 0 （极性非反相）：
当 C2VIN+ > C2VIN- 时， C2OUT = 1
当 C2VIN+ < C2VIN- 时， C2OUT = 0

bit 5 **C2OE**：比较器 C2 输出使能位
1 = C2OUT 出现在 C2OUT 引脚上 (1)
0 = C2OUT 仅在内部有效

bit 4 **C2POL**：比较器 C2 输出极性选择位
1 = C2OUT 逻辑反相
0 = C2OUT 逻辑不反相

bit 3 **未实现**：读为 0

bit 2 **C2R**：比较器 C2 参考电压选择位 （同相输入）
1 = C2VIN+ 连接至 C2VREF
0 = C2VIN+ 连接至 C2IN+ 引脚

bit 1-0 **C2CH<1:0>**：比较器 C2 通道选择位
00 = C2 的 C2VIN- 引脚连接至 C12IN0-
01 = C2 的 C2VIN- 引脚连接至 C12IN1-
10 = C2 的 C2VIN- 引脚连接至 C12IN2-
11 = C2 的 C2VIN- 引脚连接至 C12IN3-

注 **1**： 比较器输出需要以下 3 个条件：C2OE = 1， C2ON = 1 且相应的端口 TRIS 位 = 0。

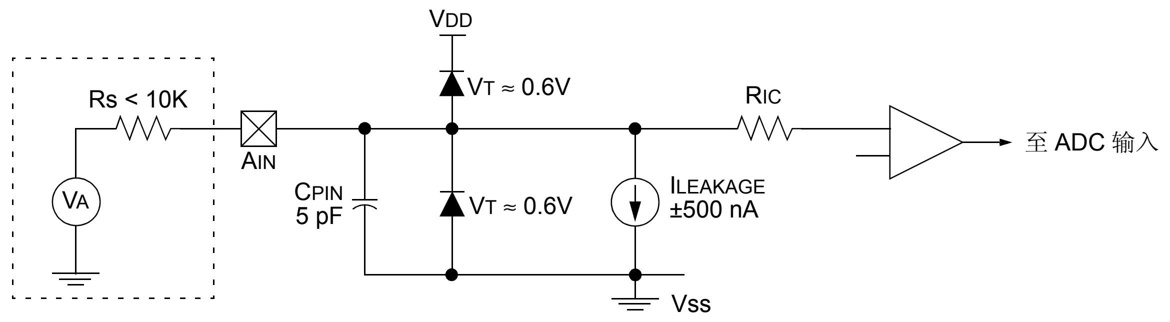
8.7 比较器模拟输入连接注意事项

模拟输入的简化电路如图 8-6 所示。由于模拟输入引脚与数字输入共用连接，它们在 VDD 和 VSS 之间连有反向偏置的 ESD 保护二极管。因此，模拟输入必须在 VSS 和 VDD 之间。如果输入电压与这一范围偏离的绝对值超过 0.6V，就可能发生一个二极管正向导通，从而可能导致锁死发生。

模拟信号源的最大阻抗推荐值为 10 kΩ。任何连接到模拟输入引脚的外部元件 （如电容或齐纳二极管），应保证其泄漏电流极小以使引入的误差降至最低。

注	
1：	读端口寄存器时，所有配置为模拟输入的引脚均读为 0。配置为数字输入的引脚将根据输入规范转换为模拟输入。
2：	定义为数字输入引脚上的模拟电平可能会使输入缓冲器的电流消耗超过规定值。

图 8-6： 模拟输入模型



图注：
- CPIN = 输入电容
- ILEAKAGE = 各结点在引脚上产生的泄漏电流
- RIC = 片内走线等效电阻
- RS = 信号源阻抗
- VA = 模拟电压
- VT = 门限电压

 初稿

8.8 其他比较器特性

共有 3 个额外的比较器特性：

- Timer1 计数使能 （门控）
- 使输出与 Timer1 同步
- 比较器输出的同时读取

8.8.1 由比较器 C2 提供 TIMER1 的门控源

这个特性可用来对模拟事件的持续时间或时间间隔计时。清零 CM2CON1 寄存器的 T1GSS 位将使 Timer1 根据比较器 C2 的输出递增。这需要已开启 Timer1 且已使能门控。详见**第 6.0 节 "带门控的 Timer1 模块"**。

当比较器用作 Timer1 门控源时，建议通过将 C2SYNC 位置 1 使比较器与 Timer1 同步。这可以确保当比较器输出在递增期间改变时， Timer1 不会错过递增。

8.8.2 使比较器 C2 的输出与 TIMER1 同步

通过将 CM2CON1 寄存器的 C2SYNC 位置 1，可以使比较器 C2 的输出与 Timer1 保持同步。使能 C2 的输出时， C2 的输出在 Timer1 时钟源的下降沿被锁存。如果 Timer1 使用了预分频器，则比较器的输出在经过预分频后被锁存。为了防止发生竞争，比较器的输出在 Timer1 时钟源的下降沿被锁存，而 Timer1 在其时钟源的上升沿递增。更多信息，请参见比较器框图（图 8-3）和 Timer1 框图 （图 6-1）。

8.8.3 比较器输出的同时读取

CM2CON1 寄存器的 MC1OUT 和 MC2OUT 位是两个比较器输出的镜像副本。从同一个寄存器中同时读取两个输出的功能，可以消除从不同寄存器读取时存在的时序差。

注	1：	通过读 CM2CON1 获取 C1OUT 或 C2OUT 的状态并不影响比较器中断不匹配寄存器。

寄存器 8-3： CM2CON1：比较器 2 的控制寄存器 1

R-0	R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0
MC1OUT	MC2OUT	—	T1ACS	C1HYS	C2HYS	T1GSS	C2SYNC
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

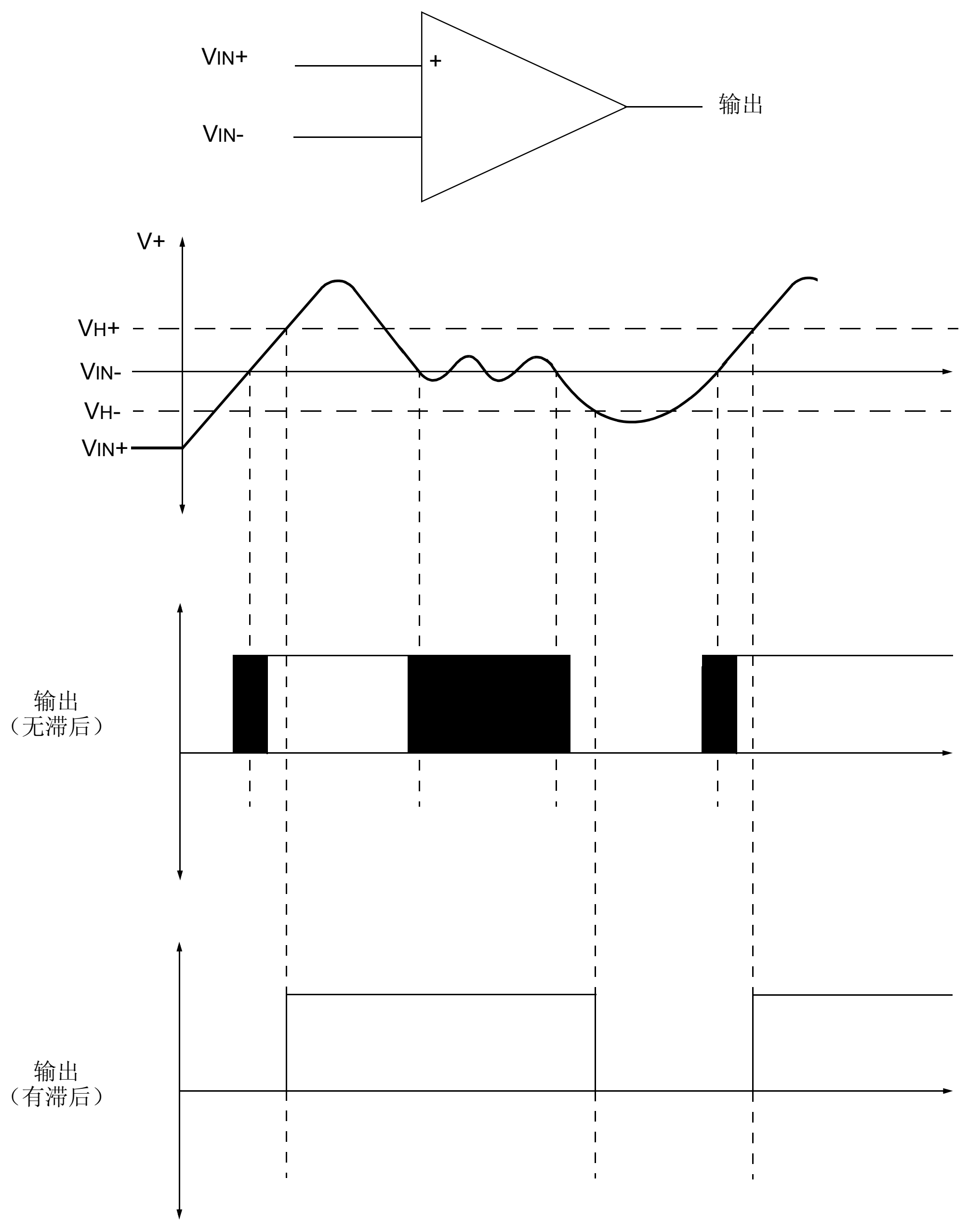
bit 7	**MC1OUT**：C1OUT 位的镜像副本
bit 6	**MC2OUT**：C2OUT 位的镜像副本
bit 5	**未实现**：读为 0
bit 4	**T1ACS**：Timer1 备用时钟选择位 1 = Timer1 时钟源为系统时钟 （Fosc） 0 = Timer1 时钟源为内部时钟 （Fosc/4）
bit 3	**C1HYS**：比较器 C1 滞后使能位 1 = 比较器 C1 滞后使能 0 = 比较器 C1 滞后禁止
bit 2	**C2HYS**：比较器 C2 滞后使能位 1 = 比较器 C2 滞后使能 0 = 比较器 C2 滞后禁止
bit 1	**T1GSS**：Timer1 门控源选择位 1 = Timer1 门控源为 $\overline{\text{T1G}}$ 0 = Timer1 门控源为 SYNCC2OUT
bit 0	**C2SYNC**：比较器 C2 输出同步位 1 = C2 输出与 Timer1 时钟的下降沿同步 0 = C2 输出异步

8.9 比较器滞后

每个比较器都具有内置的滞后功能，它可以由用户通过将 CM2CON1 寄存器的 C1HYS 或 C2HYS 位置 1 来使能。该滞后功能有助于滤除噪声，并在比较器输出改变状态时减少多个比较器输出跳变。

图 8-9 显示了使用滞后和不使用滞后情况下模拟输入电平和比较器数字输出之间的关系。只有当 VIN+ 上的模拟电压上升到滞后电路上限阈值（VH+）以上时，比较器输出才从低电平状态变化到高电平状态。只有当 VIN+ 上的模拟电压下降到滞后电路下限阈值（VH-）以下时，比较器输出才从高电平状态变化到低电平状态。

图 8-7： 比较器滞后



注： 比较器输出的黑色区域表示因输入失调电压和响应时间所造成的输出不确定区域。

初稿

表 **8-2**： 与比较器和参考电压模块相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
CM1CON0	C1ON	C1OUT	C1OE	C1POL	C1SP	C1R	C1CH1	C1CH0	0000 0000	0000 0000
CM2CON0	C2ON	C2OUT	C2OE	C2POL	C2SP	C2R	C2CH1	C2CH0	0000 0000	0000 0000
CM2CON1	MC1OUT	MC2OUT	—	T1ACS	C1HYS	C2HYS	T1GSS	C2SYNC	00-0 0010	00-0 0010
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 000x	0000 000x
PIE1	—	ADIE(1)	CCP1IE(1)	C2IE	C1IE	—	TMR2IE(1)	TMR1IE	-000 0-00	-000 0-00
PIR1	—	ADIF(1)	CCP1IF(1)	C2IF	C1IF	—	TMR2IF(1)	TMR1IF	-000 0-00	-000 0-00
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--x0 x000	--x0 x000
PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx 00xx	--uu 00uu
SRCON0	SR1	SR0	C1SEN	C2REN	PULSS	PULSR	—	SRCLKEN	0000 00-0	0000 00-0
SRCON1	SRCS1	SRCS0	—	—	—	—	—	—	00-- ----	00-- ----
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
VRCON	C1VREN	C2VREN	VRR	FVREN	VR3	VR2	VR1	VR0	0000 0000	0000 0000

图注： x = 未知， u = 不变， — = 未实现 （读为 0）。比较器不使用阴影单元。
注 1： 仅限 PIC16F616/16HV616。

8.10 比较器 SR 锁存器

SR 锁存器模块为比较器输出提供了其他控制方法。该模块由一个 SR 锁存器和输出多路开关组成。SR 锁存器可以由比较器输出置 1、复位或翻转。SR 锁存器也可以独立于比较器输出，通过 SRCON0 控制寄存器中的控制位置 1 或复位。SR 锁存器输出多路开关选择是将锁存器输出还是比较器输出送到 I/O 端口逻辑作为引脚的最终输出。

SR 锁存器也使用可变时钟，它连接到锁存器的置 1 输入端。SRCON0 的 SRCLKEN 位使能 SR 锁存器置 1 时钟。该时钟将定期为锁存器的置 1 输入提供脉冲。 SR 锁存器置 1 时钟的频率可以通过 SRCON1 寄存器的 SRCS<1:0> 位控制。

8.10.1 锁存器操作

锁存器是不依赖于时钟源的置 1- 复位锁存器。每个置 1 和复位输入均为高电平有效。每个锁存器输入都被连接到比较器输出和软件控制的脉冲发生器。可以通过 C1OUT 或 SRCON0 寄存器的 PULSS 位将锁存器置 1。可以通过 C2OUT 或 SRCON0 寄存器的 PULSR 位将锁存器复位。该锁存器是复位优先型的，因此，如果置 1 和复位输入同时为高电平，则锁存器将进入复位状态。 PULSS 和 PULSR 位都是自复位的，也就是说，对两个位中的任一个位执行一次写操作是完成锁存器置 1 或复位操作的必要条件。

8.10.2 锁存器输出

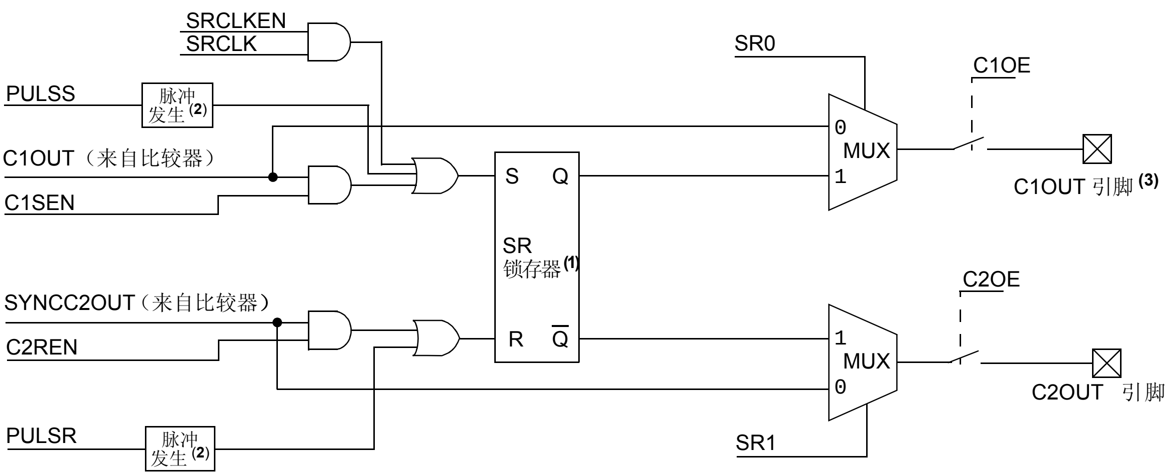
SRCON0 寄存器的 SR<1:0> 位控制锁存器输出多路开关，并确定 4 种可能的输出配置。在这 4 种配置中，CxOUT I/O 端口逻辑连接到：

- C1OUT 和 C2OUT
- C1OUT 和 SR 锁存器 $\overline{Q}$
- C2OUT 和 SR 锁存器 Q
- SR 锁存器 Q 和 $\overline{Q}$

发生任何复位之后，默认的输出配置是未锁存的 C1OUT 和 C2OUT 模式。这维持了与不具有 SR 锁存器功能的器件的兼容性。

必须清零相应端口对应的 TRIS 位以使能端口引脚输出驱动器。此外，要使比较器或锁存器的输出出现在输出引脚上，必须将 CMxCON0 寄存器的 CxOE 比较器输出使能位置 1。锁存器配置使能状态与比较器的使能状态完全无关。

图 8-8： SR 锁存器的简化框图



注 1： 如果 R = 1 且 S = 1，则 Q = 0，$\overline{Q}$ = 1。
2： 脉冲发生器产生 1 个 Tosc 脉冲宽度。
3： 图中的输出仅供参考。更多详细信息，请参见 I/O 端口引脚框图。

 初稿

寄存器 8-4： SRCON0：SR 锁存器的控制寄存器 0

R/W-0	R/W-0	R/W-0	R/W-0	R/S-0	R/S-0	U-0	R/W-0
SR1[(2)]	SR0[(2)]	C1SEN	C2REN	PULSS	PULSR	—	SRCLKEN
bit 7							bit 0

图注：		S = 只可置 1 位	
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

bit 7 **SR1**：SR 锁存器配置位 [(2)]
1 = C2OUT 引脚为锁存器 $\overline{Q}$ 输出
0 = C2OUT 引脚为 C2 比较器输出

bit 6 **SR0**：SR 锁存器配置位 [(2)]
1 = C1OUT 引脚为锁存器 Q 输出
0 = C1OUT 引脚为 C1 比较器输出

bit 5 **C1SEN**：C1 置 1 使能位
1 = C1 比较器输出将 SR 锁存器置 1
0 = C1 比较器输出对 SR 锁存器没有影响

bit 4 **C2REN**：C2 复位使能位
1 = C2 比较器输出将 SR 锁存器复位
0 = C2 比较器输出对 SR 锁存器没有影响

bit 3 **PULSS**：为 SR 锁存器提供置 1 输入脉冲位
1 = 触发脉冲发生器将 SR 锁存器置 1。该位立即由硬件复位。
0 = 不触发脉冲发生器。

bit 2 **PULSR**：为 SR 锁存器提供复位输入脉冲位
1 = 触发脉冲发生器将 SR 锁存器复位。该位立即由硬件复位。
0 = 不触发脉冲发生器。

bit 1 **未实现**：读为 0

bit 0 **SRCLKEN**：SR 锁存器置 1 时钟使能位
1 = SRCLK 为 SR 锁存器的置 1 输入提供脉冲
0 = SRCLK 不为 SR 锁存器的置 1 输入提供脉冲

注 **1：** 不管 SR 锁存器的操作如何， CMxCON0 寄存器中的 C1OUT 和 C2OUT 位总是反映实际的比较器输出 （不是引脚上的电平）。
2： 要使 SR 锁存器输出出现在引脚上，相应的 CxOE 和 TRIS 位必须正确配置。

寄存器 8-5： SRCON1：SR 锁存器的控制寄存器 1

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
SRCS1	SRCS0	—	—	—	—	—	—
bit 7							bit 0

图注：		S = 只可置 1 位	
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

bit 7-6 **SRCS<1:0>**：SR 锁存器预分频比位
00 = Fosc/16
01 = Fosc/32
10 = Fosc/64
11 = Fosc/128

bit 5-0 **未实现**：读为 0

8.11 比较器参考电压

比较器参考电压模块为比较器提供了内部产生的参考电压。它具有以下特性：

- 与比较器操作独立
- 两个 16 级电压范围
- 输出箝位到 V_{SS}
- 与 V_{DD} 成比例
- 固定参考电压 （0.6V）

VRCON 寄存器 （寄存器 8-6）控制参考电压模块，如图 8-9 所示。

8.11.1 独立操作

比较器参考电压独立于比较器配置。将 VRCON 寄存器的 FVREN 位置 1 将使能参考电压。

8.11.2 输出电压选择

CV_{REF} 参考电压有 2 个范围，每个范围有 16 个电平。范围的选择由 VRCON 寄存器的 VRR 位控制。16 个电平通过 VRCON 寄存器的 VR<3:0> 位进行设置。

CV_{REF} 输出电压由下面的公式确定：

公式 8-1： CV_{REF} 输出电压

V_{RR} = 1（低电平范围）：

$$CV_{REF} = (VR\langle 3:0\rangle/24) \times V_{DD}$$

V_{RR} = 0（高电平范围）：

$$CV_{REF} = (V_{DD}/4) + (VR\langle 3:0\rangle \times V_{DD}/32)$$

此模块的结构可以实现 V_{SS} 到 V_{DD} 的满量程。请参见图 8-9。

8.11.3 输出箝位到 V_{SS}

可通过清零 VRCON 寄存器的 FVREN 位（FVREN = 0），将固定参考电压输出电压设为 V_{SS} 且不产生功耗。这使得比较器可以检测到过零点，且不额外消耗模块电流。

8.11.4 输出与 VDD 成比例

比较器参考电压来自 V_{DD}，因此，CV_{REF} 输出会随着 V_{DD} 的变化而变化。经过测试的比较器参考电压的绝对精度请参见**第 15.0 节 “电气规范”**。

初稿

8.11.5 固定参考电压

固定参考电压独立于 VDD，输出电压标称值为 0.6V。可通过将 VRCON 寄存器的 FVREN 位设为 1 使能该参考电压。当 HFINTOSC 振荡器有效时，该参考电压始终是使能的。

8.11.6 固定参考电压稳定周期

当固定参考电压模块使能时，需要一段时间使参考电压及其放大器电路达到稳定。用户程序必须包含允许模块稳定的延时子程序。关于最小延时要求，请参见电气规范章节。

8.11.7 参考电压选择

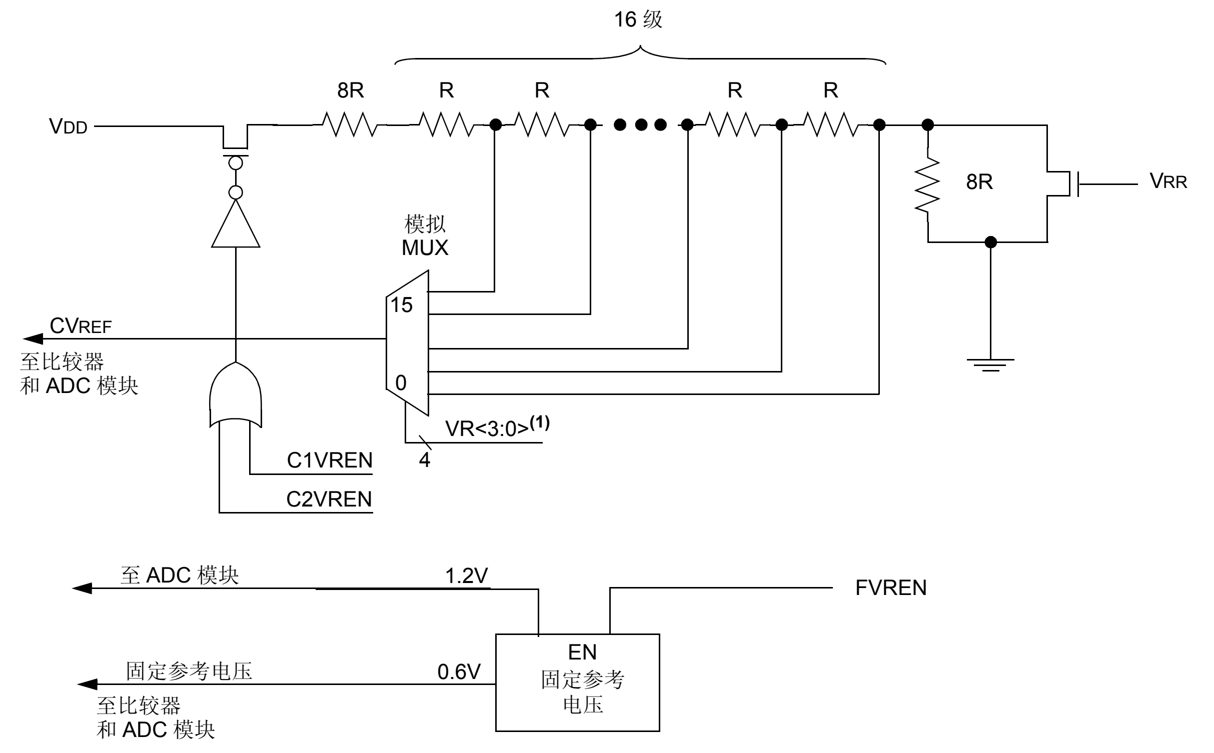
参考电压模块输出上的多路开关使比较器可以选择使用 CVREF 或固定参考电压。

将 VRCON 寄存器的 C1VREN 位置 1 使电流流入 CVREF 分压器，并使 C1 选择使用 CVREF 电压。清零 C1VREN 位使 C1 选择使用固定电压。

将 VRCON 寄存器的 C2VREN 位置 1 使电流流入 CVREF 分压器，并使 C2 选择使用 CVREF 电压。清零 C2VREN 位使 C2 选择使用固定电压。

当同时清零 C1VREN 和 C2VREN 位时，禁止电流流入 CVREF 分压器，从而将参考电压外设的功耗降至最低。

图 8-9： 比较器参考电压框图



注 1： 应注意确保 VREF 维持在比较器共模输入范围之内。更多详细信息，请参见**第15.0节“电气规范”**。

寄存器 8-6： VRCON：参考电压控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
C1VREN	C2VREN	VRR	FVREN	VR3	VR2	VR1	VR0
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

bit 7 **C1VREN**：比较器 1 参考电压使能位
1 = CVREF 电路上电并送到比较器 C1 的 C1VREF 输入
0 = 0.6 伏固定参考电压送到比较器 C1 的 C1VREF 输入

bit 6 **C2VREN**：比较器 2 参考电压使能位
1 = CVREF 电路上电并送到比较器 C2 的 C2VREF 输入
0 = 0.6 伏固定参考电压送到比较器 C2 的 C2VREF 输入

bit 5 **VRR**：CVREF 电压范围选择位
1 = 低电平范围
0 = 高电平范围

bit 4 **FVREN**：固定参考电压 （0.6V）使能位
1 = 使能
0 = 禁止

bit 3-0 **VR<3:0>**：比较器参考电压 CVREF 值选择位 （0 ≤ VR<3:0> ≤ 15）
当 VRR = 1 时：CVREF = (VR<3:0>/24) * VDD
当 VRR = 0 时：CVREF = VDD/4 + (VR<3:0>/32) * VDD

9.0 模数转换器 （ADC）模块 （仅限 PIC16F616/16HV616）

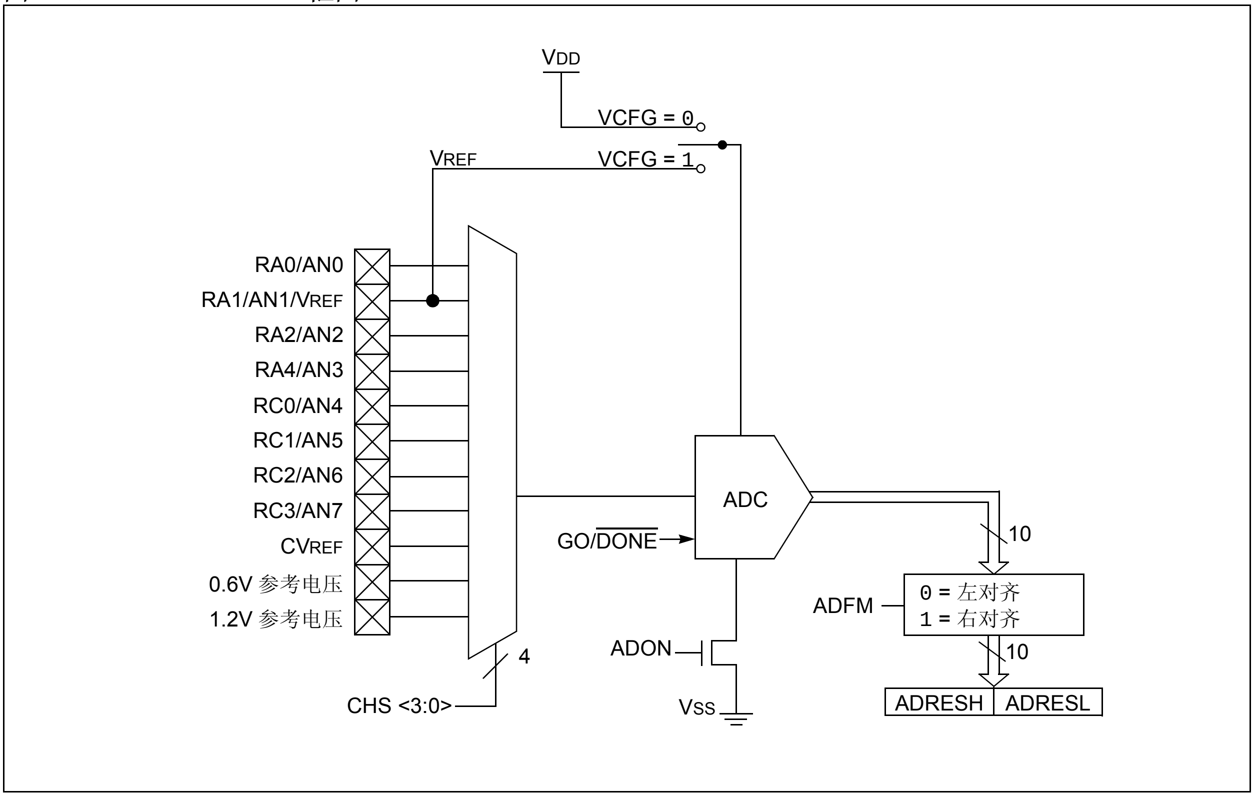
模数转换器（Analog-to-digital Converter，ADC）可将模拟输入信号转换为信号的 10 位二进制表示。该器件使用模拟输入，这些输入被多路转换到同一个采样和保持电路。采样保持电路的输出与转换器的输入相连接。转换器通过逐次逼近法产生 10 位二进制结果，并将转换结果存储在ADC结果寄存器（ADRESL和ADRESH）中。

可通过软件方式选择 VDD 或施加在外部参考引脚上的电压作为 ADC 参考电压。

ADC 可在转换完成时产生中断。该中断可用于将器件从休眠状态唤醒。

图 9-1 给出了 ADC 的框图。

图 9-1： ADC 框图



9.1 ADC 配置

配置和使用 ADC 时必须考虑以下功能：

- 端口配置
- 通道选择
- ADC 参考电压选择
- ADC 转换时钟源
- 中断控制
- 结果格式

9.1.1 端口配置

ADC 可用于转换模拟和数字信号。转换模拟信号时，I/O 引脚应通过将相关的 TRIS 和 ANSEL 位置 1 配置为模拟。更多信息，请参见有关相应端口的章节。

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增加。

9.1.2 通道选择

ADCON0 寄存器的 CHS 位决定与采样和保持电路相连接的通道。

改变通道时，在开始下一次转换前需要一段延时。更多信息，请参见**第 9.2 节 “ADC 工作原理”**。

9.1.3 ADC 参考电压

ADCON0 寄存器的 VCFG 位用于控制正参考电压。正参考电压可以是 VDD 或外部电压源。负参考电压总是连接到参考地。

9.1.4 转换时钟

可通过软件方式设置 ADCON1 寄存器的 ADCS 位来选择转换时钟源。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 （专用内部振荡器）

完成一个位转换所需的时间定义为 TAD。一次完整的 10 位转换需要 11 个 TAD 周期，如图 9-3 中所示。

为正确转换，必须满足适当的 TAD 规范。更多信息，请参见**第 15.0 节 “电气规范”**中的 A/D 转换要求。表 9-1 给出了适当的 ADC 时钟选择的示例。

注：	除非使用 FRC，否则系统时钟频率的任何改变都会改变 ADC 时钟频率，从而对 ADC 结果产生不利影响。

表 9-1： ADC 时钟周期 （TAD）与器件工作频率 （VDD ≥ 3.0V）关系表

ADC 时钟周期 （TAD）		器件频率 （FOSC）			
ADC 时钟源	**ADCS<2:0>**	**20 MHz**	**8 MHz**	**4 MHz**	**1 MHz**
FOSC/2	000	100 ns(2)	250 ns(2)	500 ns(2)	2.0 μs
FOSC/4	100	200 ns(2)	500 ns(2)	1.0 μs(2)	4.0 μs
FOSC/8	001	400 ns(2)	1.0 μs(2)	2.0 μs	8.0 μs(3)
FOSC/16	101	800 ns(2)	2.0 μs	4.0 μs	16.0 μs(3)
FOSC/32	010	1.6 μs	4.0 μs	8.0 μs(3)	32.0 μs(3)
FOSC/64	110	3.2 μs	8.0 μs(3)	16.0 μs(3)	64.0 μs(3)
FRC	x11	2-6 μs(1,4)	2-6 μs(1,4)	2-6 μs(1,4)	2-6 μs(1,4)

图注： 阴影单元表示超出了建议范围。

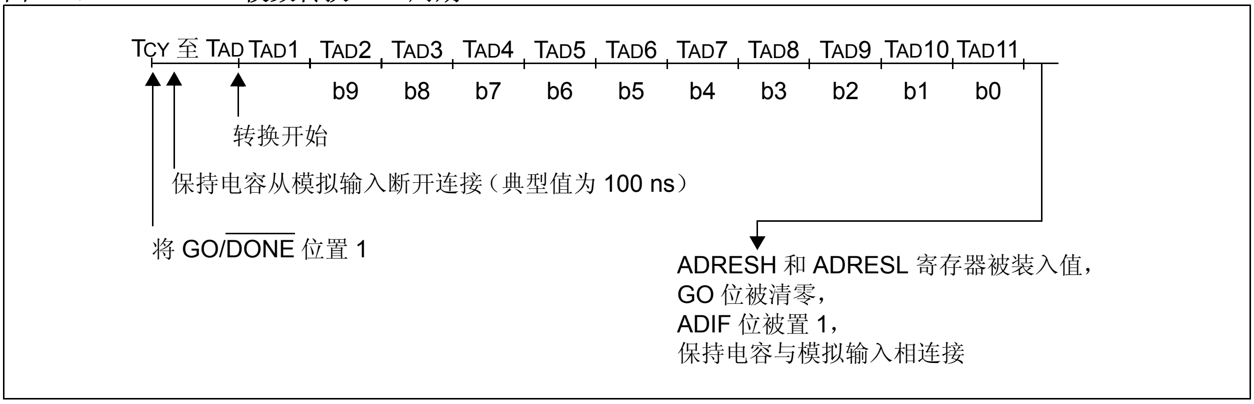
注 **1：** VDD > 3.0V 时， FRC 时钟源具有 4 μs 的典型 TAD 时间。

2： 这些值均违反了所需的最小 TAD 时间。

3： 为了加快转换速度，建议选用其他时钟源。

4： 当器件频率高于 1 MHz 时，仅当在休眠状态下进行转换时才推荐使用 FRC 时钟源。

图 9-2： 模数转换 TAD 周期



9.1.5 中断

ADC 模块可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。 ADIF 位必须用软件清零。

注：	ADIF 位在每次转换完成时置 1，与是否允许了 ADC 中断无关。

器件运行或休眠时都可产生该中断。如果器件处于休眠状态，该中断会唤醒器件。从休眠状态唤醒时，总是执行 SLEEP 指令后紧跟的下一条指令。如果用户试图从休眠状态唤醒器件并恢复主代码执行，必须禁止全局中断。如果允许了全局中断，执行将切换到中断服务程序。

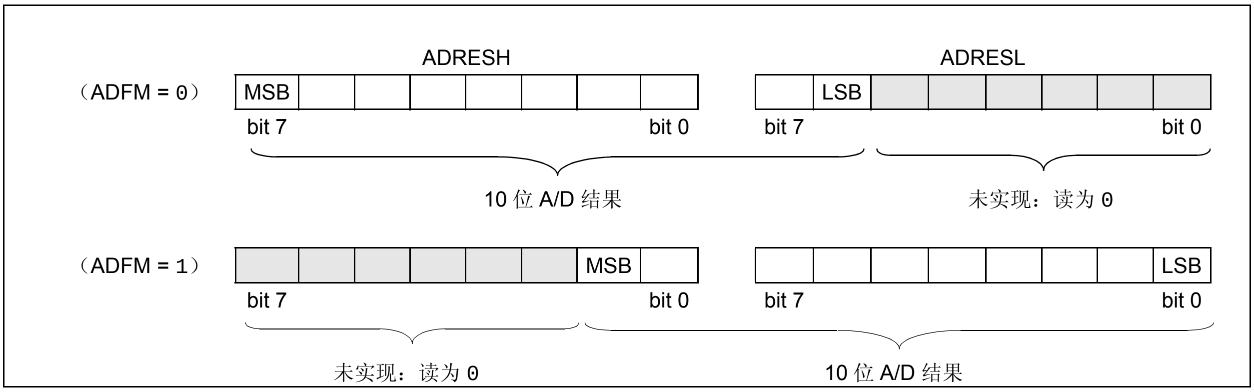
更多信息，请参见**第 9.1.5 节 “中断”**。

9.1.6 结果格式

10 位 A/D 转换结果可以两种格式提供：左对齐或右对齐。 ADCON0 寄存器的 ADFM 位控制输出格式。

图 9-4 给出了两种输出格式。

图 9-3： **10 位 A/D 转换结果格式**



9.2 ADC 工作原理

9.2.1 启动转换

要使能 ADC 模块，ADCON0 寄存器的 ADON 位必须设置为 1。将 ADCON0 寄存器的 GO/$\overline{\text{DONE}}$ 位设置为 1 将启动模数转换。

> 注： 不应在启动 ADC 的同一条指令中将 GO/$\overline{\text{DONE}}$ 位置 1。请参见**第 9.2.6 节“A/D 转换步骤”**。

9.2.2 转换完成

转换完成时， ADC 模块将：

- 清零 GO/$\overline{\text{DONE}}$ 位
- 将 ADIF 标志位置 1
- 用新的转换结果更新 ADRESH:ADRESL 寄存器

9.2.3 终止转换

如果必须在转换完成前终止转换，可用软件将 GO/$\overline{\text{DONE}}$ 位清零。不会用部分完成的模数转换结果更新 ADRESH 和 ADRESL 寄存器， ADRESH:ADRESL 寄存器对仍然保留前一次转换的值。此外，必须经过2 TAD 的延时后才能开始下一次采集。延时结束后，将自动开始对选定通道的输入进行采集。

> 注： 器件复位将强制所有寄存器为复位状态。因此， ADC 模块被关闭，任何进行中的转换操作被终止。

9.2.4 休眠期间的 ADC 操作

ADC 模块可以在休眠模式下工作。这需要把 ADC 转换时钟设置为 FRC 选项。选择 FRC 时钟源后， ADC 需等待一个指令周期后才能启动转换操作。这使得可以执行 SLEEP 指令，以降低转换期间的系统噪声。如果允许了 ADC 中断，转换完成时器件将从休眠状态唤醒。如果禁止了 ADC 中断，尽管 ADON 位仍保持置 1，转换完成后 ADC 模块将关闭。

ADC 时钟源不是 FRC 时，尽管 ADON 位仍保持置 1，SLEEP 指令会导致当前转换中止， ADC 模块关闭。

9.2.5 特殊事件触发器

ECCP 特殊事件触发器允许定期测试 ADC 而无需软件干预。当出现触发信号后， GO/$\overline{\text{DONE}}$ 位由硬件置 1，Timer1 计数器复位为零。

使用特殊事件触发器不能确保正确的 ADC 时序。用户需负责确保 ADC 时序要求得到满足。

更多信息，请参见**第 10.0 节“增强型捕捉 / 比较 /PWM（带自动关闭和死区）模块 （仅限 PIC16F616/16HV616）”**。

9.2.6 A/D 转换步骤

以下是用 ADC 执行模数转换的示例步骤：

1. 配置端口：
   - 禁止引脚输出驱动器 （见 TRIS 寄存器）
   - 将引脚配置为模拟
2. 配置 ADC 模块：
   - 选择 ADC 转换时钟
   - 配置参考电压
   - 选择 ADC 输入通道
   - 选择结果格式
   - 开启 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. 等待所需采集时间 (2)。
5. 通过将 GO/$\overline{\text{DONE}}$ 位置 1 启动转换。
6. 通过以下方式之一等待 ADC 转换完成：
   - 查询 GO/$\overline{\text{DONE}}$ 位
   - 等待 ADC 中断 （已允许中断）
7. 读取 ADC 结果
8. 清零 ADC 中断标志（如果已允许中断则需要）。

> 注 **1**： 如果用户试图从休眠状态唤醒器件并恢复主代码执行，必须禁止全局中断。
> **2**： 请参见**第 9.3 节 “A/D 采集要求”**。

例 9-1： A/D 转换

```
;This code block configures the ADC
;for polling, Vdd reference, Frc clock
;and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL   ADCON1       ;
MOVLW     B'01110000'  ;ADC Frc clock
MOVWF     ADCON1       ;
BANKSEL   TRISA        ;
BSF       TRISA,0      ;Set RA0 to input
BANKSEL   ANSEL        ;
BSF       ANSEL,0      ;Set RA0 to analog
BANKSEL   ADCON0       ;
MOVLW     B'10000001'  ;Right justify,
MOVWF     ADCON0       ;Vdd Vref, AN0, On
CALL      SampleTime   ;Acquisiton delay
BSF       ADCON0,GO    ;Start conversion
BTFSC     ADCON0,GO    ;Is conversion done?
GOTO      $-1          ;No, test again
BANKSEL   ADRESH       ;
MOVF      ADRESH,W     ;Read upper 2 bits
MOVWF     RESULTHI     ;store in GPR space
BANKSEL   ADRESL       ;
MOVF      ADRESL,W     ;Read lower 8 bits
MOVWF     RESULTLO     ;Store in GPR space
```

9.2.7 ADC 寄存器定义

以下寄存器用于控制 ADC 的操作。

寄存器 9-1： ADCON0：A/D 控制寄存器 0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	VCFG	CHS3	CHS2	CHS1	CHS0	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	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **ADFM**：A/D 转换结果格式选择位
1 = 右对齐
0 = 左对齐

bit 6 **VCFG**：参考电压位
1 = V_{REF} 引脚
0 = V_{DD}

bit 5-2 **CHS<3:0>**：模拟通道选择位
0000 = 通道 00 （AN0）
0001 = 通道 01 （AN1）
0010 = 通道 02 （AN2）
0011 = 通道 03 （AN3）
0100 = 通道 04 （AN4）
0101 = 通道 05 （AN5）
0110 = 通道 06 （AN6）
0111 = 通道 07 （AN7）
1000 = 保留——未使用
1001 = 保留——未使用
1010 = 保留——未使用
1011 = 保留——未使用
1100 = CV_{REF}
1101 = 0.6V 固定参考电压 (1)
1110 = 1.2V 固定参考电压 (1)
1111 = 保留——未使用

bit 1 **GO/$\overline{\text{DONE}}$**：A/D 转换状态位
1 = A/D 转换正在进行。将该位置 1 可启动 A/D 转换周期。
A/D 转换完成后，该位由硬件自动清零。
0 = A/D 转换已完成 / 未进行

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

注 1： CHS<3:0> 位改为选择 1.2V 或 0.6V 固定参考电压时，参考输出电压将有一个瞬变。如果比较器模块使用该 VP6 参考电压，比较器输出可能由于该瞬变而暂时改变状态。

 初稿

寄存器 9-2： ADCON1：A/D 控制寄存器 1

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	ADCS2	ADCS1	ADCS0	—	—	—	—
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

bit 7 **未实现：**读为 0

bit 6-4 **ADCS<2:0>：**A/D 转换时钟选择位
000 = FOSC/2
001 = FOSC/8
010 = FOSC/32
x11 = FRC（由专用内部振荡器产生的时钟，其频率的最大值为 500 kHz）
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64

bit 3-0 **未实现：**读为 0

寄存器 9-3： ADRESH：ADC 结果寄存器高字节 （ADRESH） ADFM = 0 （只读）

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
ADRES9	ADRES8	ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2
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

bit 7-0 **ADRES<9:2>**：ADC 结果寄存器位
10 位转换结果的高 8 位

寄存器 9-4： ADRESL：ADC 结果寄存器低字节 （ADRESL） ADFM = 0 （只读）

R-x	R-x	U-0	U-0	U-0	U-0	U-0	U-0
ADRES1	ADRES0	—	—	—	—	—	—
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

bit 7-6 **ADRES<1:0>**：ADC 结果寄存器位
10 位转换结果的低 2 位

bit 5-0 **保留**：不要使用。

寄存器 9-5： ADRESH：ADC 结果寄存器高字节 （ADRESH） ADFM = 1 （只读）

U-0	U-0	U-0	U-0	U-0	U-0	R-x	R-x
—	—	—	—	—	—	ADRES9	ADRES8
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

bit 7-2 **保留**：不要使用。

bit 1-0 **ADRES<9:8>**：ADC 结果寄存器位
10 位转换结果的高 2 位

寄存器 9-6： ADRESL：ADC 结果寄存器低字节 （ADRESL） ADFM = 1 （只读）

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
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

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

9.3 A/D 采集要求

为了使 ADC 达到规定的精度，必须使充电保持电容（C_{HOLD}）充满至输入通道的电平。模拟输入模型见图 9-4。模拟信号源阻抗（R_S）和内部采样开关阻抗（R_{SS}）直接影响电容 C_{HOLD} 的充电时间。采样开关阻抗（R_{SS}）随器件电压（V_{DD}）的变化而变化，参见图 9-4。**建议模拟信号源的最大阻抗为 10 kΩ**。采集时间随着源阻抗的降低而缩短。在选择（或改变）模拟输入通道后，必须在开始转换前完成 A/D 采集。可以使用公式 9-1 来计算最小采集时间。该公式假设误差为 1/2 LSb（ADC 转换需要 1024 步）。1/2 LSb 误差是 ADC 达到规定分辨率所允许的最大误差。

公式 **9-1**： 采集时间示例

假设： 温度 = 50°C，外部阻抗 10KΩ，5.0V V_{DD}

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

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

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

T_C 值可以用以下公式近似计算：

$$V_{APPLIED}\left(1 - \frac{1}{2047}\right) = V_{CHOLD} \qquad ;[1]\ V_{CHOLD}\ \text{充电到}\ 1/2\ LSb\ \text{误差范围}$$

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

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

求解 T_C：

$$T_C = -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ \ln(1/2047)$$

$$= -10pF(1k\Omega + 7k\Omega + 10k\Omega)\ \ln(0.0004885)$$

$$= 1.37\mu s$$

因此：

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

$$= 7.67\mu s$$

注	
	1： 因为参考电压（V_{REF}）自行抵消，因此它对该公式没有影响。
	2： 充电保持电容（C_{HOLD}）在每次转换后不会放电。
	3： 建议模拟信号源的最大阻抗为 10 kΩ。此要求是为了符合引脚泄漏电流规范。

图 9-4： 模拟输入模型

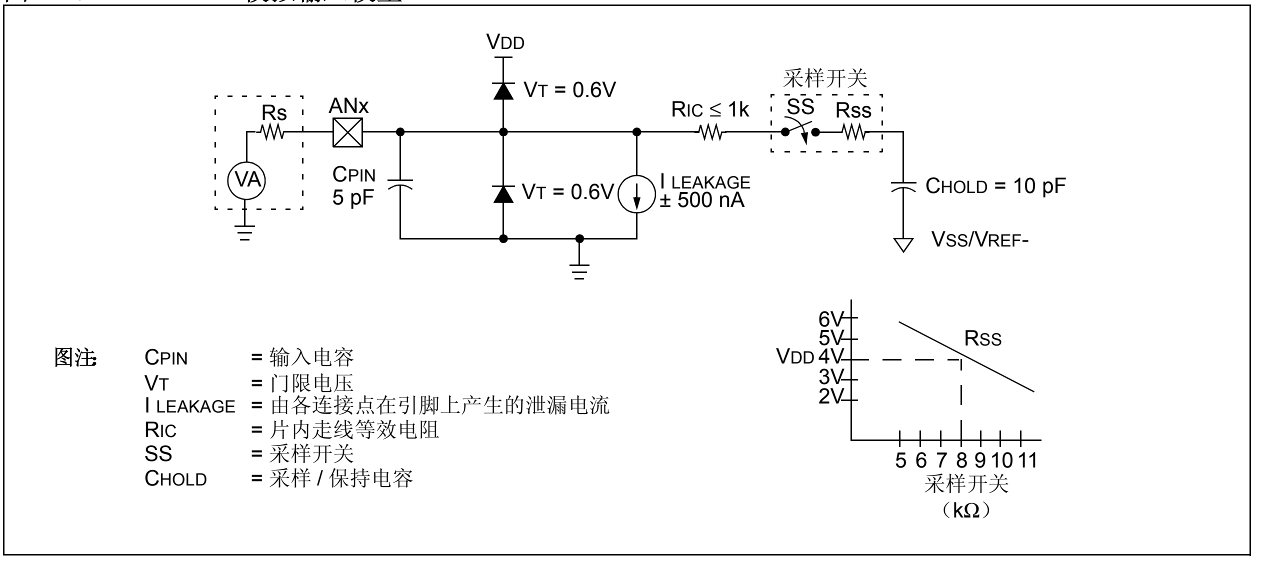
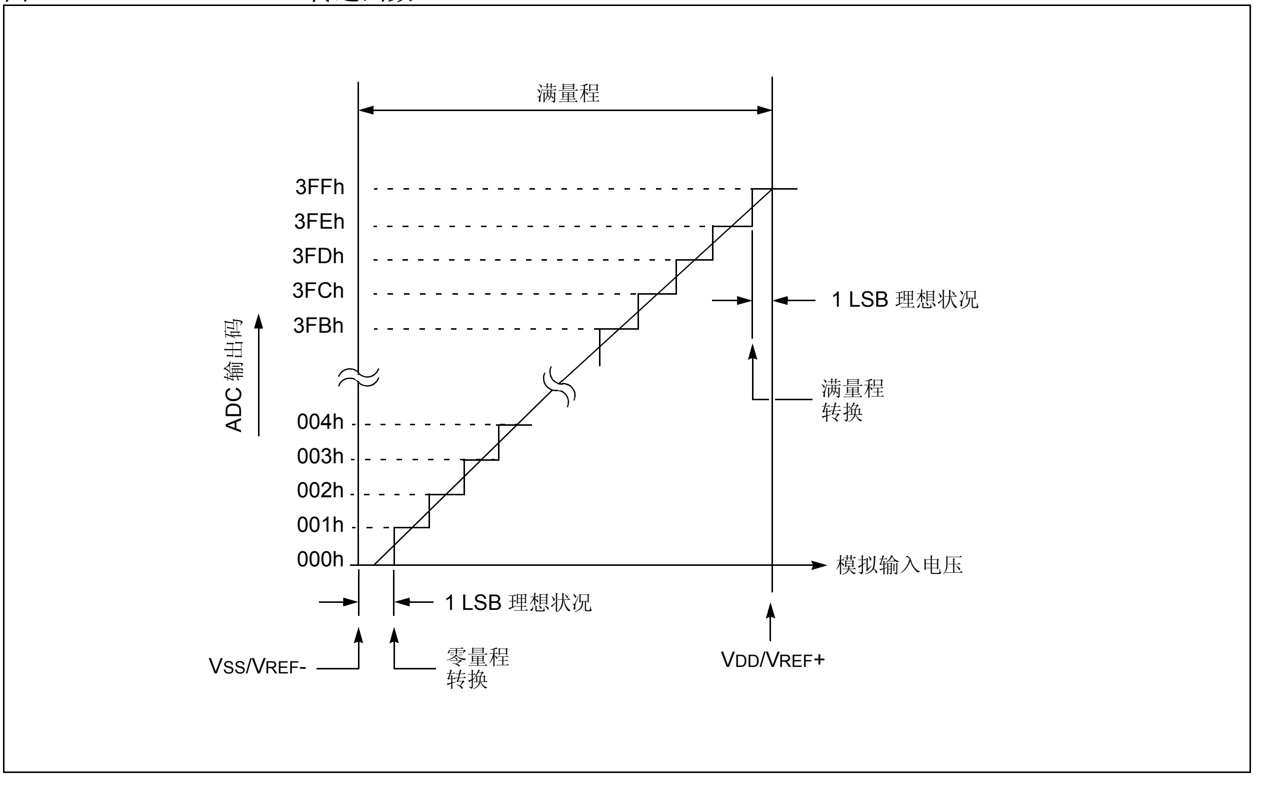


图 9-5： ADC 传递函数



 初稿

表 9-2： 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	POR/BOR 时的值	所有其他复位时的值
ADCON0[(1)]	ADFM	VCFG	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	0000 0000	0000 0000
ADCON1[(1)]	—	ADCS2	ADCS1	ADCS0	—	—	—	—	-000 ----	-000 ----
ANSEL	ANS	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
ADRESH	A/D 结果寄存器的高字节								xxxx xxxx	uuuu uuuu
ADRESL	A/D 结果寄存器的低字节								xxxx xxxx	uuuu uuuu
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
PIE1	—	ADIE[(1)]	CCP1IE[(1)]	C2IE	C1IE	—	TMR2IE[(1)]	TMR1IE	-000 0-00	-000 0-00
PIR1	—	ADIF[(1)]	CCP1IF[(1)]	C2IF	C1IF	—	TMR2IF[(1)]	TMR1IF	-000 0-00	-000 0-00
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--x0 x000	--u0 u000
PORTC	—	—	RC5	RC4	RC3	RC2	RC1	RC0	--xx 00xx	--uu 00uu
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111

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

注 1： 仅限 PIC16F616/16HV616。

注：

10.0 增强型捕捉 / 比较 /PWM （带自动关闭和死区）模块（仅限 PIC16F616/16HV616）

增强型捕捉 / 比较 /PWM 模块是允许用户计时和控制不同事件的外设。在捕捉模式下，外设允许对事件的持续时间进行计时。当预先确定的时间超时后，比较模式允许用户触发一个外部事件。 PWM 模式可以产生不同频率和占空比的脉宽调制（Pulse-Width Modulate）信号。

表 10-1 给出了 ECCP 模块所需的定时器资源。

表 10-1： ECCP 模式——所需的定时器资源

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

寄存器 10-1： CCP1CON：增强型 CCP1 控制寄存器

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

bit 7-6 **P1M<1:0>**：PWM 输出配置位
如果 CCP1M<3:2> = 00、 01 和 10：
xx = P1A 配置为捕捉 / 比较输入； P1B、 P1C 和 P1D 配置为端口引脚
如果 CCP1M<3:2> = 11：
00 = 单输出； P1A 被调制； P1B、 P1C 和 P1D 配置为端口引脚
01 = 全桥正向输出； P1D 被调制； P1A 有效； P1B 和 P1C 无效
10 = 半桥输出； P1A 和 P1B 被调制，带有死区控制； P1C 和 P1D 配置为端口引脚
11 = 全桥反向输出； P1B 被调制； P1C 有效； P1A 和 P1D 无效

bit 5-4 **DC1B<1:0>**：PWM 占空比最低有效位
捕捉模式：
未使用。
比较模式：
未使用。
PWM 模式：
这些位是 PWM 占空比的低 2 位。高 8 位在 CCPR1L 中。

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

10.1 捕捉模式

在捕捉模式下，当在引脚 CCP1 上发生下列事件之一时， CCPR1H:CCPR1L 捕捉 TMR1 寄存器中的 16 位值。这些事件如下所示，可由 CCP1CON 寄存器的 CCP1M<3:0> 位进行配置：

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

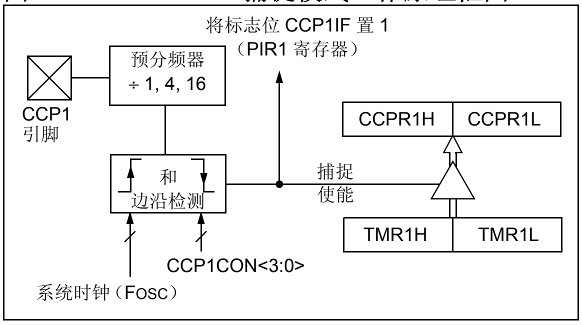
进行捕捉后，PIR1 寄存器的中断请求标志位 CCP1IF 被置 1。该中断标志位必须用软件清零。如果在 CCPR1H 和CCPR1L寄存器对中的值被读出之前又发生另一次捕捉，那么原来的捕捉值会被新捕捉值覆盖（见图10-1）。

10.1.1 CCP1 引脚配置

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

注：	如果 CCP1 引脚被配置为输出，则写端口将产生一次捕捉条件。

图 10-1： 捕捉模式工作原理框图



10.1.2 TIMER1 模式选择

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

10.1.3 软件中断

当捕捉模式改变时，可能会产生错误的捕捉中断。用户应该保持 PIE1 寄存器的 CCP1IE 中断允许位清零以避免错误中断。此外，用户应在工作模式的任何改变之后清零 PIR1 寄存器的 CCP1IF 中断标志位。

10.1.4 CCP 预分频器

通过对CCP1CON寄存器的CCP1M<3:0>位进行设置，可以指定 4 种不同的预分频比。每当关闭 CCP 模块，或者 CCP 模块不在捕捉模式下时，预分频器计数器就会被清零。任何复位都会将预分频器计数器清零。

从一个捕捉预分频比切换到另一个不会清零预分频器，而且可能产生一次错误中断。为避免此意外操作，可在改变预分频比前通过清零 CCP1CON 寄存器来关闭模块（见例 10-1）。

例 10-1： 改变捕捉预分频比

```
BANKSEL CCP1CON     ;Set Bank bits to point
                    ;to CCP1CON
CLRF    CCP1CON     ;Turn CCP module off
MOVLW   NEW_CAPT_PS ;Load the W reg with
                    ; the new prescaler
                    ; move value and CCP ON
MOVWF   CCP1CON     ;Load CCP1CON with this
                    ; value
```

 初稿

表 10-2： 与捕捉相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
CCP1CON[(1)]	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
CCPR1L[(1)]	捕捉 / 比较 /PWM 寄存器 1 的低字节								xxxx xxxx	uuuu uuuu
CCPR1H[(1)]	捕捉 / 比较 /PWM 寄存器 1 的高字节								xxxx xxxx	uuuu uuuu
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
PIE1	—	ADIE[(1)]	CCP1IE[(1)]	C2IE	C1IE	—	TMR2IE[(1)]	TMR1IE	-000 0-00	0000 0-00
PIR1	—	ADIF[(1)]	CCP1IF[(1)]	C2IF	C1IF	—	TMR2IF[(1)]	TMR1IF	-000 0-00	0000 0-00
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	uuuu uuuu
TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								xxxx xxxx	uuuu uuuu
TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								xxxx xxxx	uuuu uuuu
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111

图注： – = 未实现位 （读为 0）， u = 不变， x = 未知。捕捉、比较和 PWM 不使用阴影单元。
注 1： 仅限 PIC16F616/16HV616。

10.2 比较模式

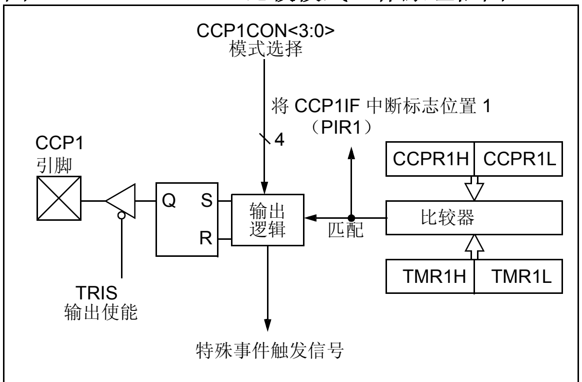
在比较模式下，16位CCPR1寄存器的值不断地与TMR1寄存器对的值作比较。如果二者匹配， CCP1 模块可能会：

- 翻转 CCP1 输出
- 将 CCP1 输出置 1
- 将 CCP1 输出清零
- 产生特殊事件触发信号
- 产生软件中断

引脚的动作由 CCP1CON 寄存器的 CCP1M<3:0> 控制位的值决定。

所有比较模式都能产生中断。

图 10-2： 比较模式工作原理框图



10.2.1 CCP1 引脚配置

用户必须通过把相关的 TRIS 位清零将 CCP1 引脚配置为输出。

注：	清零 CCP1CON 寄存器将把 CCP1 比较输出锁存器强制设为默认的低电平。这不是端口 I/O 数据锁存器。

10.2.2 TIMER1 模式选择

在比较模式下，Timer1 必须运行在定时器模式或同步计数器模式。在异步计数器模式下，比较操作可能无法进行。

10.2.3 软件中断模式

当选择“产生软件中断”模式（CCP1M<3:0> = `1010`）时， CCP1 模块不会对 CCP1 引脚进行控制（见 CCP1CON 寄存器）。

10.2.4 特殊事件触发器

当选择“特殊事件触发器”模式（CCP1M<3:0> = `1011`）时，CCP1 模块将进行以下操作：

- 复位 Timer1
- 如果 ADC 被使能，则启动 ADC 转换

在此模式下， CCP1 模块不会对 CCP1 引脚进行控制（见 CCP1CON 寄存器）。

一旦TMR1H和TMR1L寄存器对与CCPR1H和CCPR1L寄存器对之间发生匹配，便会发生 CCP 的特殊事件触发输出。TMR1H 和 TMR1L 寄存器对在 Timer1 时钟的下一个上升沿到来之前不会复位。这使 CCPR1H 和 CCPR1L 寄存器对可作为 Timer1 的 16 位可编程周期寄存器。

注	1：	CCP 模块的特殊事件触发信号不会将 PIR1 寄存器的中断标志位 TMR1IF 置 1。
	2：	通过在生成特殊事件触发信号的时钟边沿和使 Timer1 复位的时钟边沿之间更改 CCPR1H 和 CCPR1L 寄存器对的内容来移除匹配条件，可以避免复位发生。

表 **10-3**： 与比较相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
CCP1CON[(1)]	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
CCPR1L[(1)]	捕捉 / 比较 /PWM 寄存器 1 的低字节								xxxx xxxx	uuuu uuuu
CCPR1H[(1)]	捕捉 / 比较 /PWM 寄存器 1 的高字节								xxxx xxxx	uuuu uuuu
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
PIE1	—	ADIE[(1)]	CCP1IE[(1)]	C2IE	C1IE	—	TMR2IE[(1)]	TMR1IE	-000 0-00	0000 0-00
PIR1	—	ADIF[(1)]	CCP1IF[(1)]	C2IF	C1IF	—	TMR2IF[(1)]	TMR1IF	-000 0-00	0000 0-00
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	uuuu uuuu
TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								xxxx xxxx	uuuu uuuu
TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								xxxx xxxx	uuuu uuuu
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111

图注： – = 未实现位 （读为 0）， u = 不变， x = 未知。捕捉、比较和 PWM 不使用阴影单元。
注 1： 仅限 PIC16F616/16HV616。

10.3 PWM 模式

PWM 模式在 CCP1 引脚上产生脉宽调制信号。占空比、周期和分辨率由以下寄存器决定：

- PR2
- T2CON
- CCPR1L
- CCP1CON

在脉宽调制（PWM）模式下，CCP 模块会在 CCP1 引脚上产生最大 10 位分辨率的 PWM 输出信号。由于 CCP1 引脚与端口数据锁存器复用，该引脚的 TRIS 必须被清零以将 CCP1 引脚设置为输出。

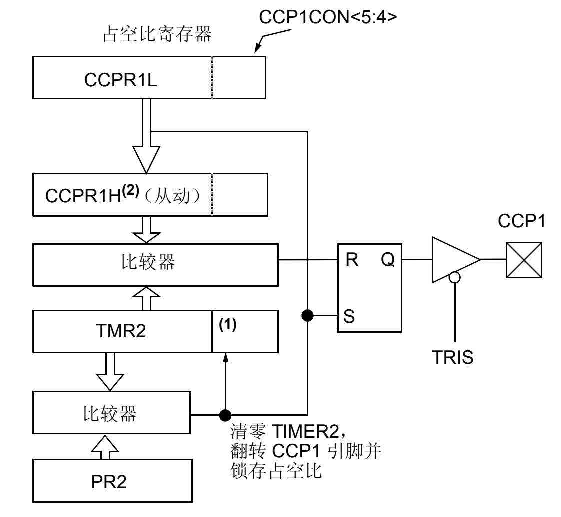
注：	清零 CCP1CON 寄存器将放弃对 CCP1 引脚的 CCP1 控制。

图 10-3 给出了 PWM 工作原理的简化框图。

图 10-4 给出了 PWM 信号的典型波形。

关于如何设置 CCP 模块使之工作于 PWM 模式的详细步骤，请参见第 **10.3.7 节 “设置 PWM 操作”**。

图 10-3： 简化的 PWM 框图

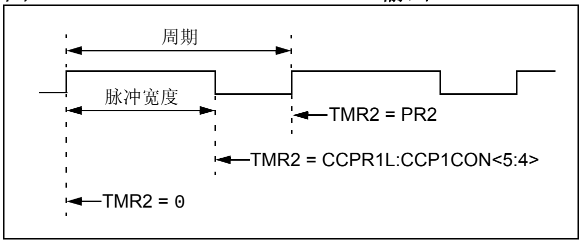


注 1： 8 位定时器 TMR2 寄存器与 2 位内部系统时钟（FOSC）或预分频器的 2 位一起构成 10 位时基。

2： 在 PWM 模式下， CCPR1H 是只读寄存器。

PWM 输出 （图 10-4）有一个时基 （周期）和一段输出保持为高电平的时间 （占空比）。

图 10-4： CCP PWM 输出



 初稿

10.3.1 PWM 周期

PWM周期可通过写Timer2的PR2寄存器来指定。PWM 周期可由公式 10-1 计算。

公式 10-1： PWM 周期

$$PWM周期 = [(PR2) + 1] \bullet 4 \bullet Tosc \bullet (TMR2预分频值)$$

当 TMR2 中的值与 PR2 中的值相等时，在下一个递增周期将发生以下 3 个事件：

- TMR2 被清零
- CCP1 引脚被置 1。（例外情况：如果 PWM 占空比 = 0%，引脚将不会被置 1。）
- PWM 占空比从 CCPR1L 锁存到 CCPR1H。

注：	在确定 PWM 频率时不会用到 Timer2 后分频比（见第 **7.1 节“Timer2 工作原理”**）。

10.3.2 PWM 占空比

通过将 10 位值写入多个寄存器来指定 PWM 占空比：CCPR1L 寄存器和 CCP1CON 寄存器的 CCP1<1:0> 位。CCPR1L 包含高 8 位而 CCP1CON 寄存器的 CCP1<1:0> 位包含低 2 位。可以在任何时候写入 CCPR1L 和 CCP1CON 寄存器的 CCP1<1:0> 位。在周期结束（即 PR2 和 TMR2 寄存器发生匹配）前占空比值不会被锁存到 CCPR1H 中。当使用 PWM 时，CCPR1H 寄存器是只读的。

公式 10-2 用于计算 PWM 脉冲宽度。

公式 10-3 用于计算 PWM 占空比。

公式 10-2： 脉冲宽度

$$脉冲宽度 = (CCPR1L:CCP1CON<5:4>) \bullet Tosc \bullet (TMR2预分频值)$$

公式 10-3： 占空比

$$占空比 = \frac{(CCPR1L:CCP1CON<5:4>)}{4(PR2 + 1)}$$

CCPR1H 寄存器和一个 2 位的内部锁存器用于给 PWM 占空比提供双重缓冲。这种双重缓冲结构非常重要，它可以避免在 PWM 操作中产生毛刺。

8 位定时器 TMR2 寄存器与 2 位内部系统时钟（FOSC）或预分频器的 2 位一起构成 10 位时基。如果 Timer2 预分频比设置为 1:1，则使用系统时钟。

当 10 位时基与 CCPR1H 和 2 位锁存值匹配时，CCP1 引脚被清零 （见图 10-3）。

10.3.3 PWM 分辨率

分辨率决定给定周期的可用占空比数。例如，10 位分辨率将可得到 1024 个不连续的占空比，而 8 位分辨率将可得到 256 个不连续的占空比。

当 PR2 为 255 时，PWM 最大分辨率为 10 位。分辨率是 PR2 寄存器值的函数，如公式 10-4 所示。

公式 10-4： PWM 分辨率

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

注：	如果脉冲宽度值比周期长，则指定的 PWM 引脚将保持不变。

表 10-4： PWM 频率和分辨率示例 （FOSC = 20 MHz）

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

表 10-5： PWM 频率和分辨率示例 （FOSC = 8 MHz）

PWM 频率	1.22 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 和 16）	16	4	1	1	1	1
PR2 值	0x65	0x65	0x65	0x19	0x0C	0x09
最大分辨率 （位）	8	8	8	6	5	5

10.3.4 休眠模式下的操作

在休眠模式下，TMR2 寄存器将不会递增，模块状态也不会改变。如果 CCP1 引脚正在驱动一个值，则会继续驱动该值。当器件被唤醒时，TMR2将从先前状态继续。

10.3.5 改变系统时钟频率

PWM 频率来自于系统时钟频率。系统时钟频率的任何改变将导致 PWM 频率的改变。更多详细信息，请参见**第 3.0 节 "振荡器模块"**。

10.3.6 复位的影响

任何复位都将强制所有端口为输入模式，并强制CCP寄存器为其复位状态。

10.3.7 设置 PWM 操作

当配置 CCP 模块的 PWM 操作时，可采用以下步骤：

1. 通过将相关的 TRIS 位置 1，配置 PWM 引脚（CCP1）为输入。
2. 通过装入 PR2 寄存器设置 PWM 周期。
3. 通过装入恰当的值到 CCP1CON 寄存器将 CCP 模块配置为 PWM 模式。
4. 通过装入 CCPR1L 寄存器和 CCP1CON 寄存器的 CCP1 位设置 PWM 占空比。
5. 配置和启动 Timer2：
   - 清零 PIR1 寄存器的 TMR2IF 中断标志位。
   - 通过装入 T2CON 寄存器的 T2CKPS 位设置 Timer2 预分频值。
   - 通过将 T2CON 寄存器的 TMR2ON 位置 1 使能 Timer2。
6. 在新的 PWM 周期开始后使能 PWM 输出：
   - 等待直到 Timer2 溢出（PIR1 寄存器的 TMR2IF 位置 1）。
   - 通过清零相关的 TRIS 位使能 CCP1 引脚输出。

10.4 PWM（增强型模式）

增强型PWM模式可以在最多4个输出引脚上产生PWM信号，最高可达10位分辨率。可通过4种PWM输出模式实现：

- 单PWM
- 半桥PWM
- 全桥PWM，正向模式
- 全桥PWM，反向模式

要选择增强型PWM模式，CCP1CON寄存器的P1M位必须适当置1。

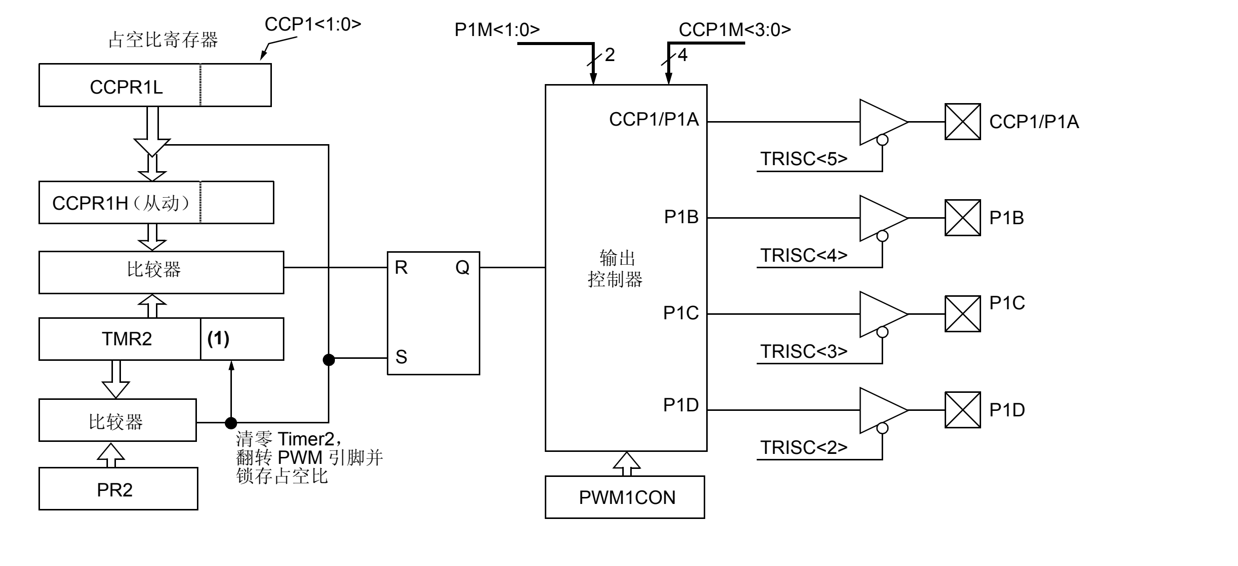
PWM输出与I/O引脚复用，指定为P1A、P1B、P1C和P1D。PWM引脚的极性是可配置的，通过将CCP1CON寄存器中的CCP1M位适当置1来选择。

表10-6给出了每种增强型PWM模式的引脚分配。

图10-5给出了增强型PWM模块的简化框图的示例。

> **注：** 为防止在第一次使能PWM时产生不完整的波形，ECCP模块在产生PWM信号前会等待直到新的PWM周期开始。

图 10-5： 增强型 PWM 模式的简化框图示例



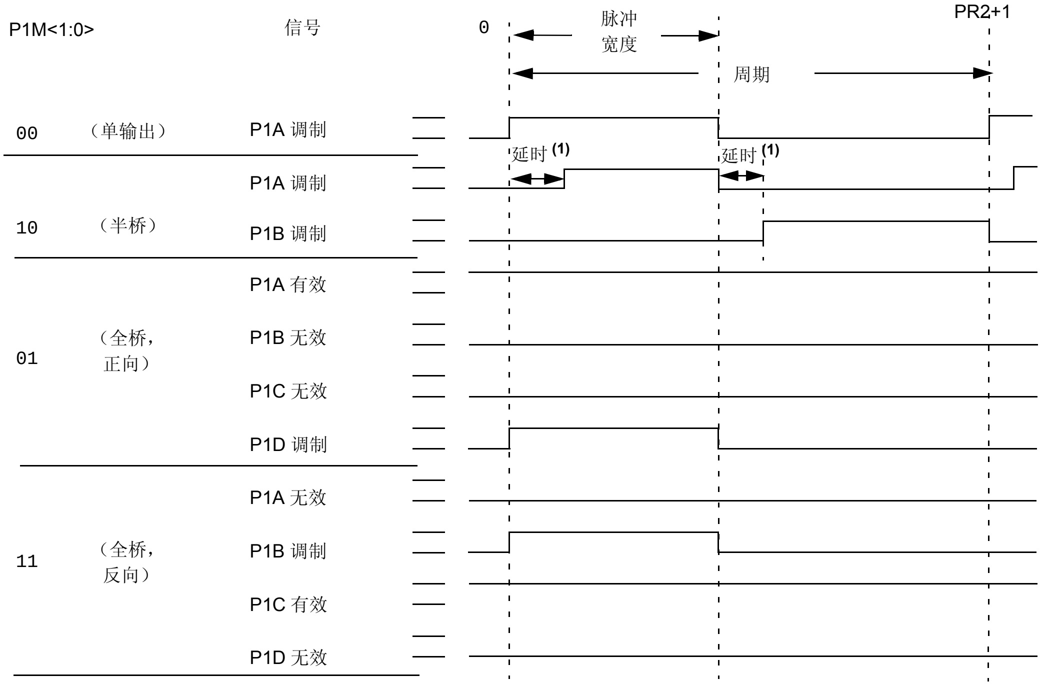
注 1：8位定时器TMR2寄存器与2位内部Q时钟或预分频器中的2位一起构成10位时基。

> **注** 1：每个PWM输出的TRIS寄存器值必须进行适当配置。
> 2：清零CCP1CON寄存器将放弃对所有PWM输出引脚的ECCP控制。
> 3：增强型PWM模式没有使用的任何引脚均可用于备用引脚功能。

表 10-6： 各种 PWM 增强型模式的引脚分配示例

ECCP 模式	P1M	CCP1/P1A	P1B	P1C	P1D
单	00	有	无	无	无
半桥	10	有	有	无	无
全桥，正向	01	有	有	有	有
全桥，反向	11	有	有	有	有

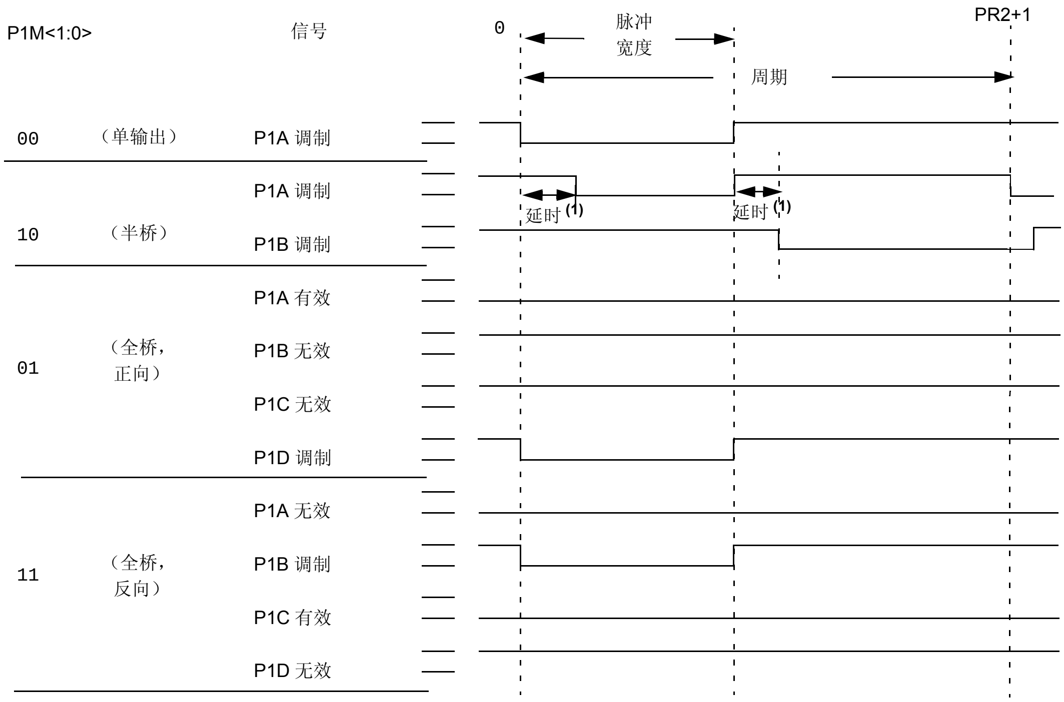
图 **10-6：** **PWM** （增强型模式）输出关系示例 （高电平有效状态）



关系：
- 周期 = 4 * Tosc * (PR2 + 1) * (TMR2 预分频值)
- 脉冲宽度 = Tosc * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 预分频值)
- 延时 = 4 * Tosc * (PWM1CON<6:0>)

注 **1：** 使用 PWM1CON 寄存器对死区延时编程 （**第 10.4.6 节 “可编程死区延时模式”**）。

图 10-7： 增强型 PWM 输出关系示例 （低电平有效状态）



关系：
- 周期 = 4 * Tosc * (PR2 + 1) * (TMR2 预分频值)
- 脉冲宽度 = Tosc * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 预分频值)
- 延时 = 4 * Tosc * (PWM1CON<6:0>)

注 1： 使用 PWM1CON 寄存器对死区延时编程 （**第 10.4.6 节 "可编程死区延时模式"**）。

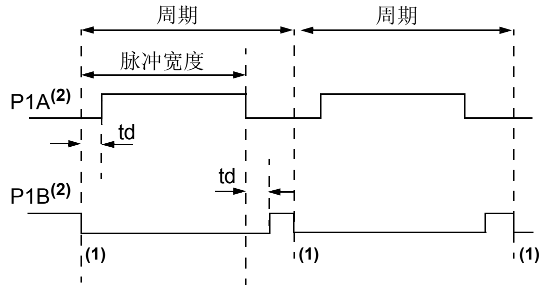
10.4.1 半桥模式

在半桥模式下，有两个引脚用作输出驱动推挽式负载。CCP1/P1A 引脚输出 PWM 输出信号，P1B 引脚输出互补的 PWM 输出信号 （见图 10-8）。这种模式可用于半桥应用（如图 10-9 所示），或者用于全桥应用——这种情况下使用两个 PWM 信号调制 4 个功率开关。

在半桥模式下，可编程死区延时可用来防止半桥功率器件中流过直通（Shoot-through）电流。PWM1CON 寄存器的 PDC<6:0> 位的值设置在输出被驱动为有效之前的指令周期数。如果这个值比占空比大，则在整个周期中相应的输出保持为无效。关于死区延时操作的详细信息，请参见第 **10.4.6** 节 **"可编程死区延时模式"**。

由于 P1A 和 P1B 输出与端口数据锁存器是复用的，相关的TRIS位必须清零，从而将P1A和P1B配置为输出。

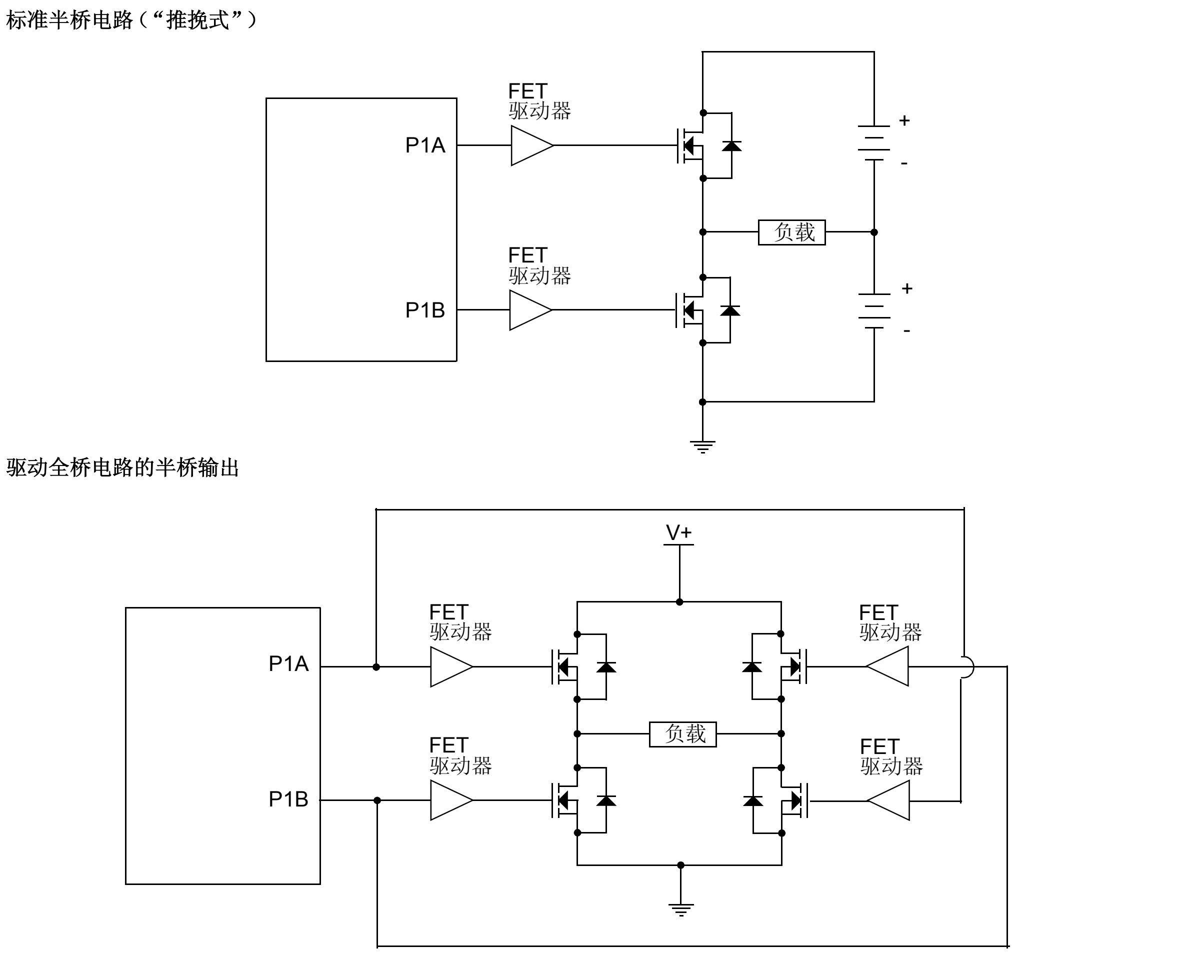
图 10-8： 半桥 **PWM** 输出示例



td = 死区延时

注 **1**：此时 TMR2 寄存器和 PR2 寄存器的值相等。
2：图中输出信号为高电平有效。

图 10-9： 半桥应用示例



 初稿

10.4.2 全桥模式

在全桥模式下， 4 个引脚都用作输出。全桥应用的示例如图 10-10 所示。

在正向模式下，引脚 CCP1/P1A 被驱动为有效状态，引脚 P1D 被调制，而 P1B 和 P1C 将被驱动为无效状态，如图 10-11 中所示。

在反向模式下，P1C 被驱动为有效状态，引脚 P1B 被调制，而 P1A 和 P1D 将被驱动为无效状态，如图 10-11 中所示。

P1A、P1B、P1C 和 P1D 输出与端口数据锁存器复用。相关的 TRIS 位必须清零，从而将 P1A、P1B、P1C 和 P1D 引脚配置为输出。

图 10-10： 全桥应用示例

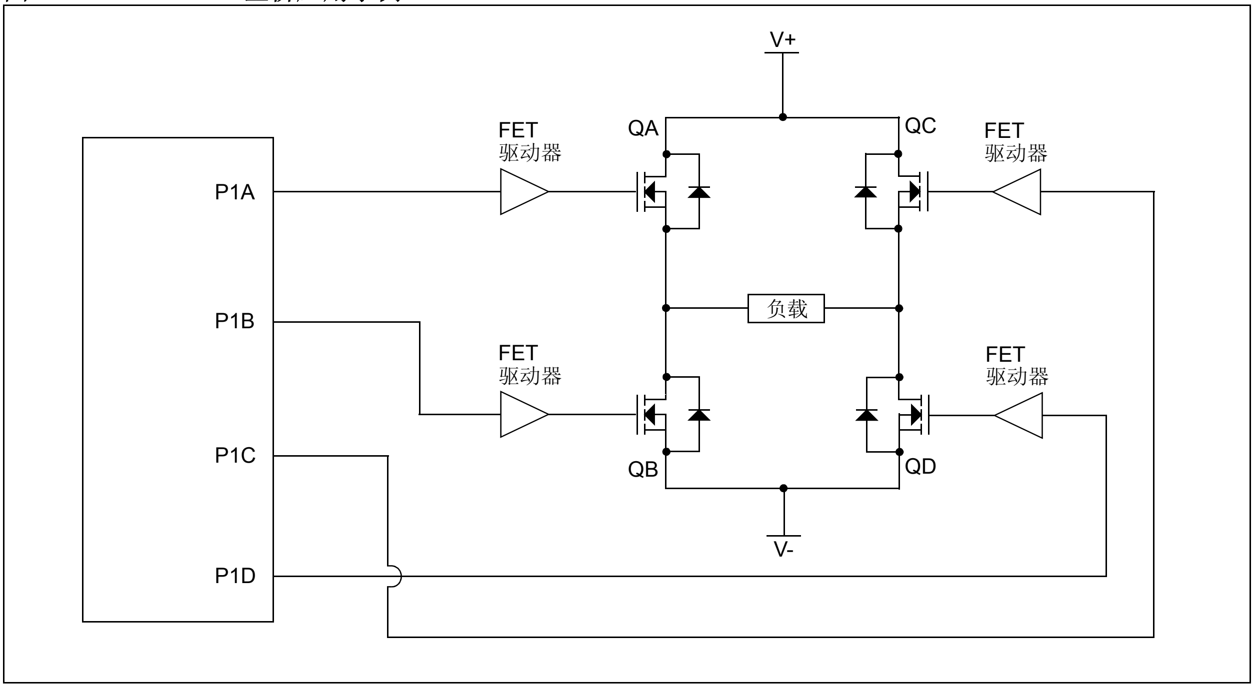
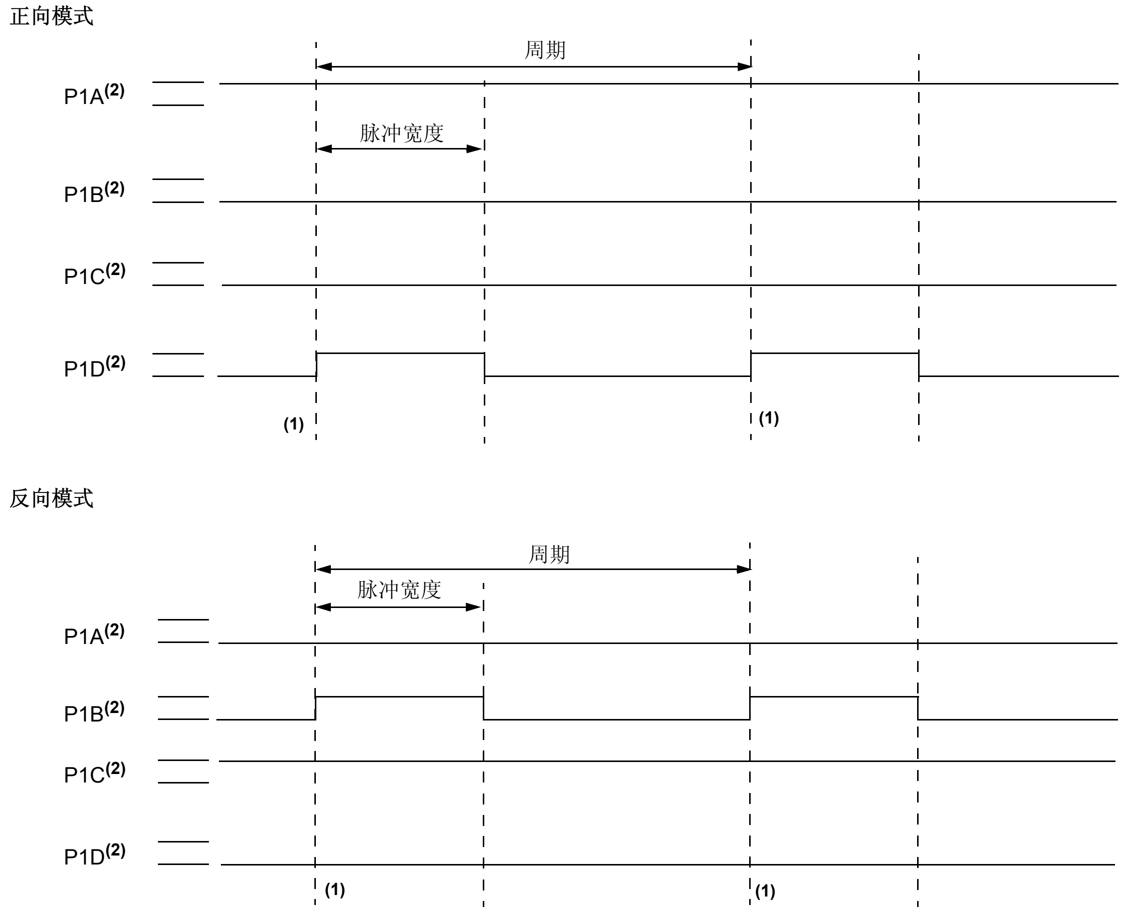


图 10-11： 全桥 **PWM** 输出示例



注 **1：** 此时 TMR2 寄存器和 PR2 寄存器的值相等。
2： 图中输出信号为高电平有效。

10.4.2.1 全桥模式中的方向改变

在全桥模式下，CCP1CON 寄存器中的 P1M1 位允许用户控制正 / 反方向。当应用软件改变这个方向控制位时，模块将在下一个 PWM 周期改用新的方向。

通过改变 CCP1CON 寄存器的 P1M1 位，可以用软件启动方向改变。以下序列在当前 PWM 周期结束前发生 4 个 Timer2 周期：

- 调制输出 （P1B 和 P1D）进入无效状态。
- 相关的未调制输出 （P1A 和 P1C）被切换到以相反的方向驱动。
- PWM 调制在下一个周期开始继续。

关于该序列的说明，请参见图 10-12。

全桥模式不提供死区延时。因为始终只有一个输出被调制，所以通常不需要死区延时。有一种情况需要死区延时。这一情况发生在以下两个条件同时满足时：

1. 当输出的占空比达到或者接近 100%，PWM 输出方向改变。
2. 功率开关（包括功率器件和驱动电路）的关断时间比导通时间要长。

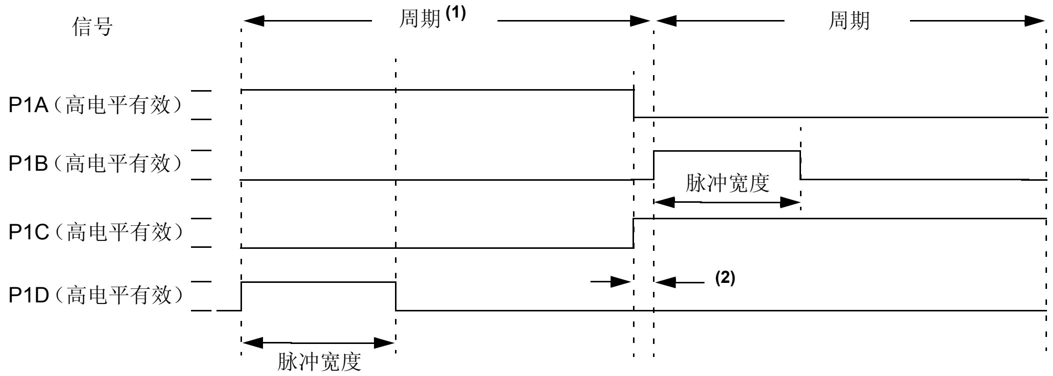
在图 10-13 所示的例子中，在占空比接近 100% 时，PWM 方向从正向改变到反向。在这个示例中，在时间 t1，输出 P1A 和 P1D 变为无效，而输出 P1C 变为有效。因为功率器件的关断时间比导通时间要长，在 “t” 时间内，功率器件 QC 和 QD 中可能流过直通电流 （见图 10-10）。当 PWM 方向从反向改变到正向时，功率器件 QA 和 QB 也将出现相同的现象。

如果应用中需要在高占空比时改变 PWM 方向，避免直通电流的两种可能方法是：

1. 在改变方向之前的一个 PWM 周期降低 PWM 占空比。
2. 使用开关驱动电路，使开关管的关断时间比导通时间短。

也可能存在其他避免直通电流的方案。

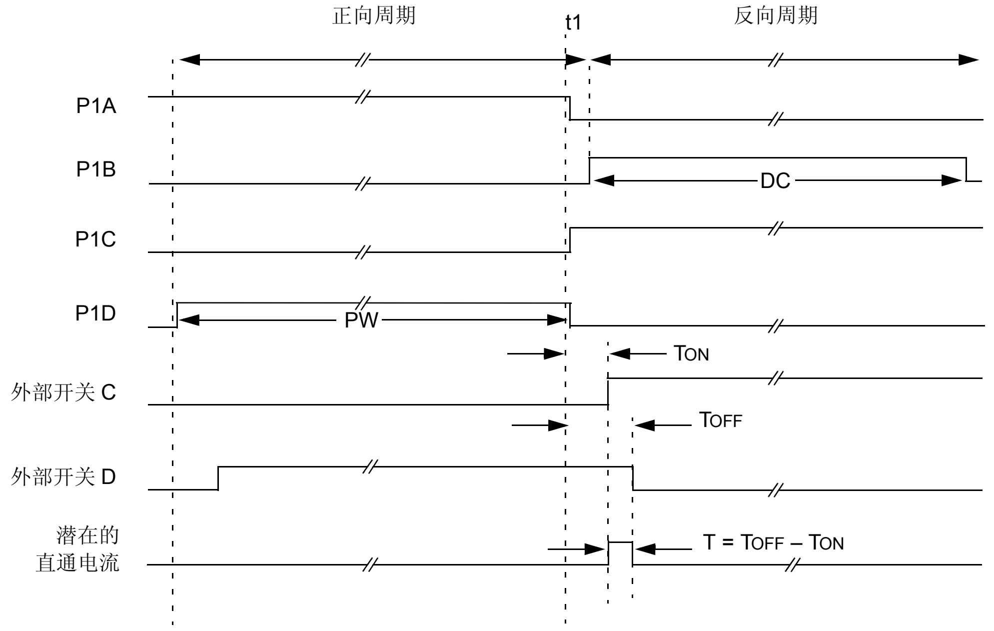
图 10-12： PWM 方向改变的示例



注 1： 可在 PWM 周期的任意时刻，写 CCP1CON 寄存器的方向位 P1M1。

2： 当改变方向时， P1A 和 P1C 信号在当前 PWM 周期结束之前切换。此时被调制的 P1B 和 P1D 信号是无效的。该时间长度为 4 个 Timer2 计数。

图 **10-13：** 在占空比接近 **100%** 时改变 **PWM** 方向的示例



注 **1：** 图中所有信号均为高电平有效。

2： TON 是功率开关 QC 及其驱动电路的导通延时。

3： TOFF 是功率开关 QD 及其驱动电路的关断延时。

10.4.3 启动注意事项

当使用 PWM 模式时，应用硬件必须在 PWM 输出引脚上外接适当的上拉和 / 或下拉电阻。

> **注：** 当单片机退出复位状态时，所有 I/O 引脚呈高阻态。外部电路必须保持功率开关器件处于关断状态，直到单片机将 I/O 引脚驱动为适当的信号电平，或者激活 PWM 输出为止。

CCP1CON 寄存器的 CCP1M<1:0> 位允许用户为每一对 PWM 输出引脚（P1A/P1C 和 P1B/P1D）选择 PWM 输出信号为高电平有效还是低电平有效。PWM 输出极性必须在 PWM 引脚配置为输出之前选择。建议不要在 PWM 引脚被配置为输出时改变极性配置，因为这可能导致应用电路损坏。

当 PWM 模块初始化时，P1A、P1B、P1C 和 P1D 输出锁存器可能不在正确的状态。这样在配置为增强型 PWM 模式的同时使能 PWM 引脚为输出，可能损坏应用电路。应首先将增强型 PWM 模式配置为正确的输出模式并经过一个完整的 PWM 周期之后，再将 PWM 引脚配置为输出。当第二个 PWM 周期开始时，PIR1 寄存器的 TMR2IF 位置 1 表明一个完整的 PWM 周期结束了。

10.4.4 增强型 PWM 自动关闭模式

PWM模式支持自动关闭模式，当外部关闭事件发生时将禁止 PWM 输出。自动关闭模式将 PWM 输出引脚置于预先确定的状态。该模式用于防止 PWM 损坏应用。

通过使用ECCPAS寄存器的ECCPASx位来选择自动关闭源。关闭事件由以下条件产生：

- INT 引脚上的逻辑 0
- 比较器 C1
- 比较器 C2
- 用固件将 ECCPASE 位置 1

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

当关闭事件发生时，会发生以下两件事：

ECCPASE 位被设置为 1。 ECCPASE 将保持置 1 直到由固件清零或发生自动重启 （见第 **10.4.5 节 “自动重启模式”**）。

使能的 PWM 引脚被异步置为其关闭状态。PWM 输出引脚被分组为 [P1A/P1C] 和 [P1B/P1D] 对。每对引脚的状态由 ECCPAS 寄存器的 PSSAC 和 PSSBD 位决定。每对引脚可以设置为以下 3 种状态之一：

- 驱动逻辑 1
- 驱动逻辑 0
- 三态 （高阻态）

寄存器 10-2： ECCPAS：增强型捕捉 / 比较 /PWM 自动关闭控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0
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

bit 7 **ECCPASE**：ECCP 自动关闭事件状态位
1 = 发生了关闭事件； ECCP 输出为关闭状态
0 = ECCP 输出正常工作

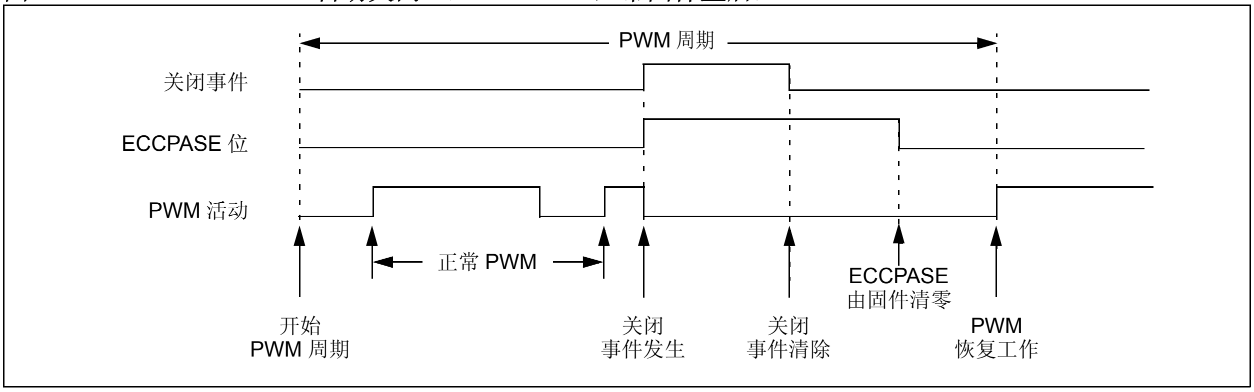
bit 6-4 **ECCPAS<2:0>**：ECCP 自动关闭源选择位
000 = 禁止自动关闭
001 = 比较器 C1 输出高电平
010 = 比较器 C2 输出高电平 (1)
011 = 任一比较器输出为高电平
100 = INT 引脚上的 V_{IL}
101 = INT 引脚上的 V_{IL} 或比较器 C1 输出高电平
110 = INT 引脚上的 V_{IL} 或比较器 C2 输出高电平
111 = INT 引脚上的 V_{IL} 或任一比较器输出为高电平

bit 3-2 **PSSACn**：引脚 P1A 和 P1C 关闭状态控制位
00 = 驱动引脚 P1A 和 P1C 为 0
01 = 驱动引脚 P1A 和 P1C 为 1
1x = 引脚 P1A 和 P1C 为三态

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

注	
	1： 自动关闭条件是基于电平的信号，而不是基于边沿的信号。只要电平存在，自动关闭就将持续。
	2： 当自动关闭条件持续时，禁止写 ECCPASE 位。
	3： 一旦自动关闭条件被移除并且 PWM 重启（通过固件或自动重启），PWM 信号将总是在下一个 PWM 周期重启。

图 10-14： PWM 自动关闭 （PRSEN = 0，带固件重启）

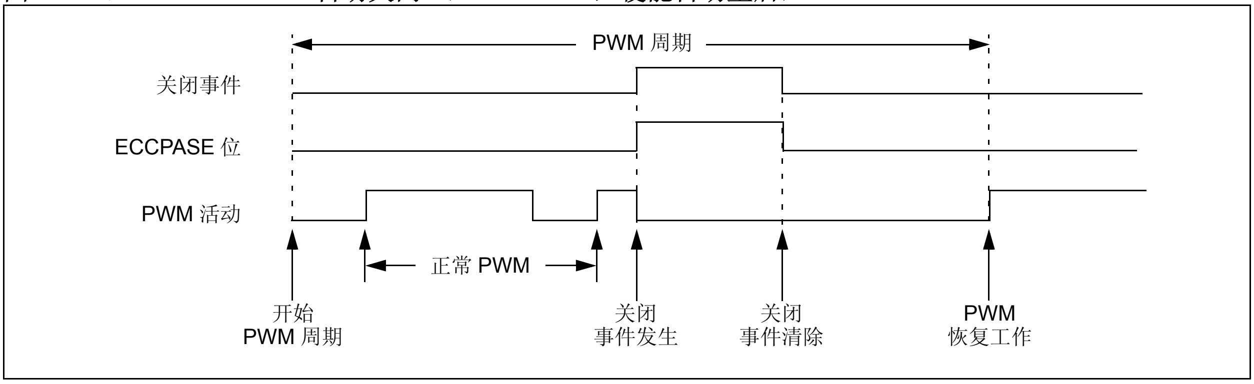


10.4.5 自动重启模式

一旦自动关闭条件被移除，增强型 PWM 可被配置为自动重启 PWM 信号。通过将 PWM1CON 寄存器中的 PRSEN 位置 1 使能自动重启。

如果使能了自动重启，只要自动关闭条件有效，ECCPASE 位将保持置 1。当自动关闭条件被移除时，ECCPASE 位将由硬件清零并继续正常工作。

图 10-15： PWM 自动关闭 （PRSEN = 1，使能自动重启）

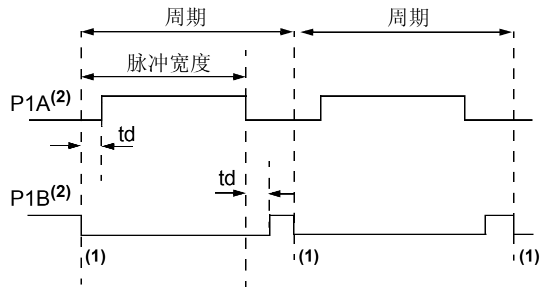


10.4.6 可编程死区延时模式

在所有功率开关管都以 PWM 频率调制的半桥应用中，功率开关管关断通常比导通需要更多的时间。如果上下两个功率开关管在同一时间开关（一个导通，另一个关断），那么在一段很短的时间里，两个开关管可能同时导通，直到一个开关管完全关断为止。在这短暂的时间中，两个功率开关管中可能流过较高的电流（*直通电流*），将逆变桥的电源与地短路。为避免开关过程中可能会出现的破坏性直通电流，通常需要延迟功率开关管的导通，保证在另一个开关管完全关断之后，再导通相应的功率开关管。

在半桥模式下，可采用数字可编程死区延时来避免出现损坏逆变桥功率开关管的直通电流。在信号从无效状态切换到有效状态时增加延时。请参见图 10-16。PWM1CON 寄存器（寄存器 10-3）的低 7 位以单片机指令周期（TCY 或 4 TOSC）为单位设置延时。

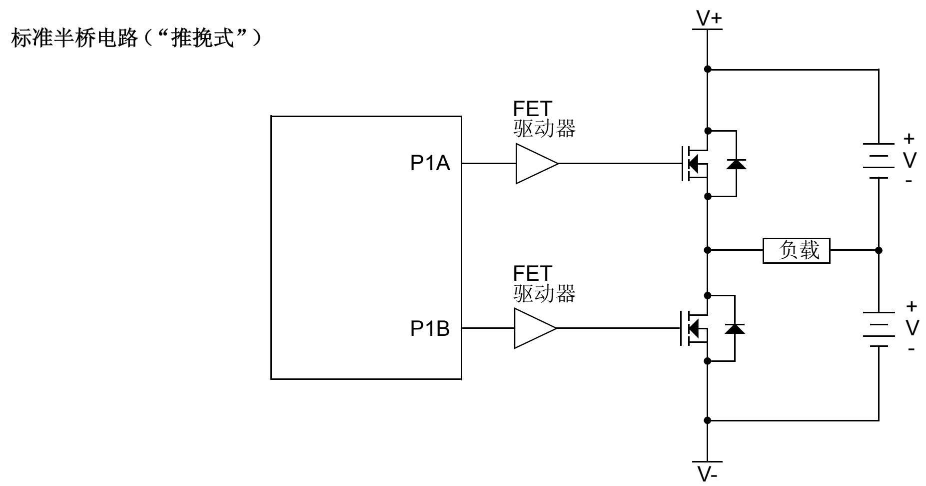
图 10-16： 半桥 **PWM** 输出示例



td = 死区延时

注 **1：** 此时 TMR2 寄存器和 PR2 寄存器的值相等。
2： 图中输出信号为高电平有效。

图 10-17： 半桥应用示例



寄存器 10-3： PWM1CON：增强型 PWM 控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
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

bit 7 **PRSEN**：PWM 重启使能位
1 = 自动关闭时，一旦关闭事件消失， ECCPASE 位自动清零； PWM 自动重启
0 = 自动关闭时， ECCPASE 必须用软件清零以重启 PWM

bit 6-0 **PDC<6:0>**：PWM 延时计数位
PDCn = 在 PWM 信号应该转换为有效的预定时间和转换为有效的实际时间之间的 Fosc/4 (4 * Tosc) 周期数。

表 10-7： 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	POR/BOR 时的值	所有其他复位时的值
CCP1CON(1)	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
CCPR1L(1)	捕捉 / 比较 /PWM 寄存器 1 的低字节								xxxx xxxx	uuuu uuuu
CCPR1H(1)	捕捉 / 比较 /PWM 寄存器 1 的高字节								xxxx xxxx	uuuu uuuu
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0	0000 -000	0000 -000
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0	0000 -000	0000 -000
CM2CON1	MC1OUT	MC2OUT	—	T1ACS	C1HYS	C2HYS	T1GSS	C2SYNC	00-0 0010	00-0 0010
ECCPAS(1)	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000	0000 0000
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
PIE1	—	ADIE(1)	CCP1IE(1)	C2IE	C1IE	—	TMR2IE(1)	TMR1IE	-000 0-00	0000 0-00
PIR1	—	ADIF(1)	CCP1IF(1)	C2IF	C1IF	—	TMR2IF(1)	TMR1IF	-000 0-00	0000 0-00
PWM1CON(1)	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000	0000 0000
T2CON(1)	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
TMR2(1)	Timer2 模块寄存器								0000 0000	0000 0000
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
TRISC	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111

图注： — = 未实现位 （读为 0）， u = 不变， x = 未知。捕捉、比较和 PWM 不使用阴影单元。
注 1： 仅限 PIC16F616/16HV616。

注：

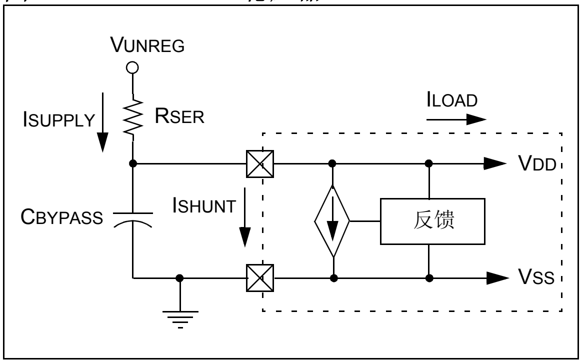
11.0 稳压器

PIC16HV616 包括一个永久的内部 5 伏（标称值）并联稳压器，与 VDD 引脚并联。从而使由未稳压电源供电的系统无需使用外部稳压器。所有直接连接到 VDD 引脚的外部器件将共用经过稳压的电源电压且其电流消耗是总的 VDD 供电电流 （ILOAD）的一部分。

11.1 稳压器工作原理

通过电阻 RSER 上产生的压降，并联稳压器可产生特定的供电电压。单片机 VDD 引脚上的电压被监控并与外部参考电压进行比较。随后根据比较结果调整流经电阻的电流，产生的压降等于单片机供电电压 VUNREG 和 VDD 之间的差值。关于稳压器原理图，请参见图 11-1。

图 11-1： 稳压器



位于未经稳压的供电电压 VUNREG 和 VDD 引脚之间的外部限流电阻 RSER，可降低 VUNREG 和 VDD 之间的电压差值。 RSER 必须在 RMAX 和 RMIN 之间，如公式 11-1 所定义。

公式 11-1： RSER 限流电阻

$$R_{MAX} = \frac{(VU_{MIN} - 5V)}{1.05 \bullet (4\ MA + I_{LOAD})}$$

$$R_{MIN} = \frac{(VU_{MAX} - 5V)}{0.95 \bullet (50\ MA)}$$

其中：

RMAX = RSER 的最大值 （欧姆）

RMIN = RSER 的最小值 （欧姆）

VUMIN = VUNREG 的最小值

VUMAX = VUNREG 的最大值

VDD = 稳压电压 （5V 标称值）

ILOAD = 最大期望负载电流（mA），包括 I/O 引脚电流和连接到 VDD 的外部电路电流。

1.05 = RSER 的 +5% 容差的补偿

0.95 = RSER 的 -5% 容差的补偿

11.2 稳压器注意事项

供电电压 VUNREG 和负载电流不是恒定的。因此，稳压器的电流范围是有限的。为 RSER 选择值时必须考虑这三个因素。

由于稳压器将带隙电压用作经过稳压的参考电压，该参考电压在 PIC16F610/16HV610 器件中将永久使能。

注：

12.0 CPU 的特殊功能

PIC16F610/616/16HV610/616 有许多功能，旨在最大限度地提高系统可靠性，通过减少外部元件的使用将成本降至最低，并提供省电工作模式和代码保护功能。

这些特殊功能包括：

- 复位
  - 上电复位 （POR）
  - 上电延时定时器 （PWRT）
  - 振荡器起振定时器 （OST）
  - 欠压复位 （BOR）
- 中断
- 看门狗定时器 （WDT）
- 振荡器选择
- 休眠
- 代码保护
- ID 存储单元
- 在线串行编程

PIC16F610/616/16HV610/616 有两个定时器提供必要的上电延时。一个是振荡器起振定时器 （OST），旨在确保芯片在晶体振荡器达到稳定之前始终处于复位状态。另一个是上电延时定时器 （PWRT），仅在上电时提供 64 ms（标称值）的固定延时，用来确保器件在供电电压稳定之前处于复位状态。还有当器件发生欠压时使器件复位的电路，该电路可使用上电延时定时器，提供至少 64 ms 的复位延时。有了这三种片上功能，绝大多数应用就无需再外接复位电路了。

休眠模式的设计是为了提供电流极低的掉电模式。用户可通过以下方法将器件从休眠模式唤醒：

- 外部复位
- 看门狗定时器唤醒
- 中断

有几种振荡器模式可供选择，以使器件适应各种应用。选择 INTOSC 可节约系统成本，而选择 LP 晶振可以节能。通过配置位的设定可选择不同选项 （见寄存器 12-1）。

12.1 配置位

可以通过对配置位编程 （读为 `0`）或不编程 （读为 `1`）来选择不同的器件配置，如寄存器 12-1 所示。这些位映射到程序存储单元 2007h 中。

注：	地址2007h超出了用户程序存储空间范围。它属于特殊配置存储空间（2000h-3FFFh），只能在编程时对其进行访问。更多信息，请参见“*PIC12F60X/12F61X/16F61X Memory Programming Specification*”（DS41284）。

寄存器 12-1： CONFIG：配置字寄存器

—	—	—	—	—	—	BOREN1(1)	BOREN0(1)
bit 15							bit 8

IOSCFS	$\overline{\text{CP}}$(2)	MCLRE(3)	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0
bit 7							bit 0

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

bit 15-10 **未实现**：读为 1

bit 9-8 **BOREN<1:0>**：欠压复位选择位 (1)
11 = 使能 BOR
10 = 正常工作时使能 BOR，休眠时禁止 BOR
0x = 禁止 BOR

bit 7 **IOSCFS**：内部振荡器频率选择位
1 = 8 MHz
0 = 4 MHz

bit 6 **$\overline{\text{CP}}$**：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 5 **MCLRE**：$\overline{\text{MCLR}}$ 引脚功能选择位 (3)
1 = $\overline{\text{MCLR}}$ 引脚功能为 MCLR
0 = $\overline{\text{MCLR}}$ 引脚功能为数字输入，$\overline{\text{MCLR}}$ 内部连接到 VDD

bit 4 **$\overline{\text{PWRTE}}$**：上电延时定时器使能位
1 = 禁止 PWRT
0 = 使能 PWRT

bit 3 **WDTE**：看门狗定时器使能位
1 = 使能 WDT
0 = 禁止 WDT

bit 2-0 **FOSC<2:0>**：振荡器选择位
111 = RC 振荡器：RA4/OSC2/CLKOUT 引脚为 CLKOUT 功能， RA5/OSC1/CLKIN 引脚上连接 RC
110 = RCIO 振荡器：RA4/OSC2/CLKOUT 引脚为 I/O 功能， RA5/OSC1/CLKIN 引脚上连接 RC
101 = INTOSC 振荡器：RA4/OSC2/CLKOUT 引脚为 CLKOUT 功能，RA5/OSC1/CLKIN 引脚为 I/O 功能
100 = INTOSCIO 振荡器：RA4/OSC2/CLKOUT 引脚为 I/O 功能，RA5/OSC1/CLKIN 引脚为 I/O 功能
011 = EC：RA4/OSC2/CLKOUT 引脚为 I/O 功能， RA5/OSC1/CLKIN 引脚上连接 CLKIN
010 = HS 振荡器：RA4/OSC2/CLKOUT 和 RA5/OSC1/CLKIN 引脚上连接高速晶振 / 谐振器
001 = XT 振荡器：RA4/OSC2/CLKOUT 和 RA5/OSC1/CLKIN 引脚上连接晶振 / 谐振器
000 = LP 振荡器：RA4/OSC2/CLKOUT 和 RA5/OSC1/CLKIN 引脚上连接低功耗晶振

注 1：使能欠压复位并不能自动使能上电延时定时器。
2：当关闭代码保护时，将擦除整个程序存储器的内容。
3：当 MCLR 在 INTOSC 或 RC 模式下变为有效时，将禁止内部时钟振荡器。

12.2 校准位

8 MHz 内部振荡器是出厂时校准的。这些校准值存储在校准字（2009h）的熔丝中。使用“*PIC12F60X/12F61X/16F61X Memory Programming Specification*”（DS41284）中指定的批量擦除序列时，校准字不会被擦除，因此也不需要再编程。

12.3 复位

PIC16F610/616/16HV610/616 有以下几种不同类型的复位：

a） 上电复位 （POR）
b） 正常工作期间的 WDT 复位
c） 休眠期间的 WDT 复位
d） 正常工作期间的 $\overline{\text{MCLR}}$ 复位
e） 休眠期间的 $\overline{\text{MCLR}}$ 复位
f） 欠压复位 （BOR）

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

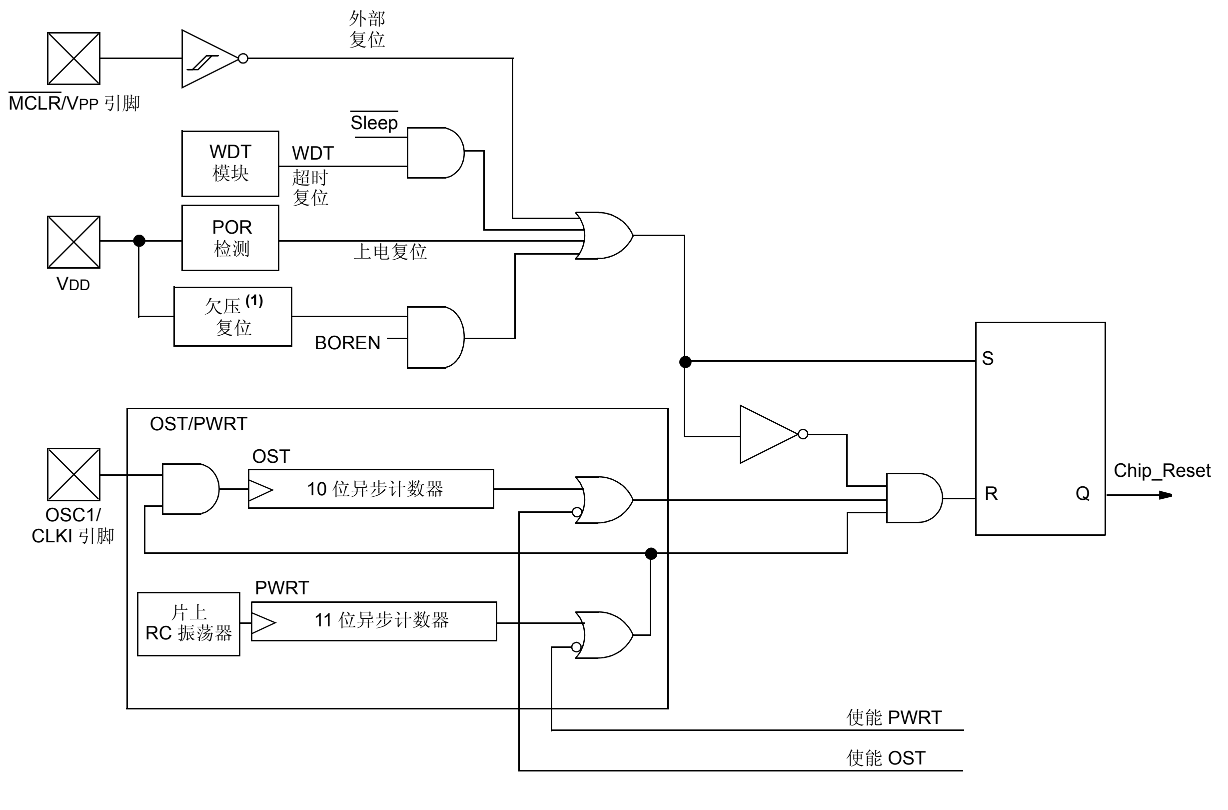
- 上电复位
- $\overline{\text{MCLR}}$ 复位
- 休眠期间的 $\overline{\text{MCLR}}$ 复位
- WDT 复位
- 欠压复位 （BOR）

WDT 唤醒不会引起类似于 WDT 复位方式下的寄存器复位，因为唤醒被视为恢复正常工作。如表 12-2 所示，$\overline{\text{TO}}$ 和 $\overline{\text{PD}}$ 位在不同的复位情形下会分别被置 1 或清零。软件可以使用这些位来判断复位的性质。关于所有寄存器的复位状态的完整说明，请参见表 12-4。

图 12-1 给出了片上复位电路的简化框图。

$\overline{\text{MCLR}}$ 复位路径上有一个噪声滤波器，用来检测并滤除小脉冲。关于脉冲宽度规范，请参见**第 15.0 节 “电气规范”**。

图 12-1： 片上复位电路的简化框图



注 1： 请参见配置字寄存器 （寄存器 12-1）。

12.3.1 上电复位 （POR）

在 VDD 达到足以使器件正常工作的电平之前，片上上电复位电路将使器件保持在复位状态。要有效利用 POR，用户只需在 MCLR 引脚和 VDD 之间串联一个电阻。这样就可无需使用通常产生上电复位所需的外接 RC 元件。需要一个最大上升时间才能达到 VDD。详见**第 15.0 节 “电气规范”**。如果使能了欠压复位，那么该最大上升时间规范将不再适用。 BOR 电路将使器件保持在复位状态，直到 VDD 达到 VBOR （见**第 12.3.4 节 “欠压复位 （BOR）”**）。

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

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

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

12.3.2 $\overline{\text{MCLR}}$

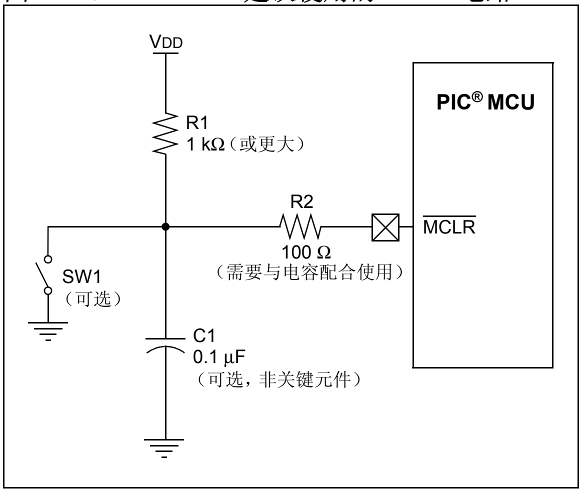
PIC16F610/616/16HV610/616 在 $\overline{\text{MCLR}}$ 复位路径中有一个噪声滤波器。该滤波器检测并滤除小脉冲。

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

施加在 $\overline{\text{MCLR}}$ 引脚上的电压超过规范值将导致 $\overline{\text{MCLR}}$ 复位，并且在 ESD 事件中产生的电流也将超过器件的规范值。因此， Microchip 建议不要把 MCLR 引脚直接连接到 VDD。建议使用图 12-2 给出的 RC 网络。

通过清零配置字寄存器中的 MCLRE 位，可使能内部 MCLR 选项。当 MCLRE = 0 时，在内部产生芯片的复位信号。当 MCLRE = 1 时，RA3/$\overline{\text{MCLR}}$ 引脚变成外部复位输入。在这种模式下，RA3/MCLR 引脚具有到 VDD 的弱上拉。

图 **12-2**： 建议使用的 **$\overline{\text{MCLR}}$** 电路



12.3.3 上电延时定时器 （PWRT）

上电延时定时器仅在上电时（上电复位或欠压复位）提供一个 64 ms（标称值）的固定延时。上电延时定时器采用内部 RC 振荡器作为时钟源。更多信息，请参见**第 3.4 节 “内部时钟模式”**。只要 PWRT 处于活动状态，芯片就保持在复位状态。PWRT 延时使 VDD 有足够的时间上升到所需的电平。配置位 $\overline{\text{PWRTE}}$ 可以禁止（如果置 1）或使能 （如果清零或被编程）上电延时定时器。虽然不是必需的，但是在使能欠压复位时也应使能上电延时定时器。

由于以下原因不同芯片的上电延时定时器的延时也各不相同：

- VDD 差异
- 温度差异
- 制造工艺差异

详见直流参数 （**第 15.0 节 “电气规范”**）。

注：	$\overline{\text{MCLR}}$引脚上若出现低于Vss的尖峰电压，感应电流超过 80 mA，可能导致锁死。因此，在 MCLR 引脚上施加低电平时，应使用一个 50-100Ω 的串联电阻，而不是将该引脚直接与 Vss 连接。

12.3.4 欠压复位 （BOR）

配置字寄存器中的 BOREN0 和 BOREN1 位用于选择 3 种 BOR 模式中的一种。选择 BOREN<1:0> = `10`，可使 BOR在休眠时被自动禁止，从而节约功耗；而在唤醒后被重新使能。关于配置字的定义，请参见寄存器 12-1。

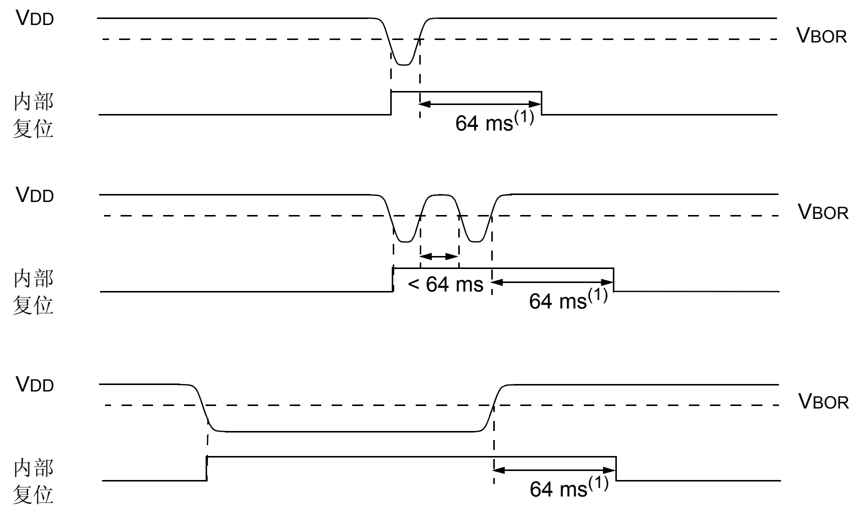
如果VDD下降到VBOR以下且持续时间超过参数值TBOR（见第 **15.0** 节 **"电气规范"**），则会发生欠压。欠压条件将使器件复位。不管 VDD 的变化速率如何，上述情况都会发生。如果 VDD 下降到 VBOR 以下且持续时间少于参数值 TBOR，则不一定会发生复位。

任何复位（上电复位、欠压复位或看门狗定时器复位等）都会使芯片保持复位状态，直到 VDD 上升到 VBOR 以上（见图 12-3）。如果使能了上电延时定时器，此时它将启动，并且会使芯片保持复位状态的时间延长 64 ms。

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

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

图 12-3： 欠压情形



注 1： 仅在 $\overline{\text{PWRTE}}$ 位被编程为 `0` 时，才增加 64 ms 延时。

12.3.5 延时时序

上电延时时序如下：

- 在 POR 延时结束后，施加一段 PWRT 延时。
- 在 PWRT 延时结束后，振荡器起振。

总延时时间取决于振荡器配置和 $\overline{\mathrm{PWRTE}}$ 位的状态。例如，在 EC 模式且 $\overline{\mathrm{PWRTE}}$ 位被擦除（PWRT 禁止）的情况下，根本不会出现延时。图 12-4、图 12-5 和图 12-6 分别描绘了各种情形下的延时时序。

由于延时是由 POR 脉冲触发的，因此如果 $\overline{\mathrm{MCLR}}$ 保持足够长时间的低电平，所有延时都将结束。将 MCLR 电平拉高后，器件将立即开始执行代码（见图 12-5）。这对于测试或同步多个并行工作的 PIC16F610/616/16HV610/616 器件来说是非常有用的。

表 12-5 给出了一些特殊寄存器的复位条件，而表 12-4 给出了所有寄存器的复位条件。

12.3.6 电源控制（PCON）寄存器

电源控制寄存器 PCON（地址 8Eh）有两个状态位，用于指示上次发生的复位的类型。

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

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

更多信息，请参见**第 12.3.4 节“欠压复位（BOR）”**。

表 **12-1**： 各种情形下的延时

振荡器配置	上电		欠压复位		从休眠状态唤醒
	$\overline{\mathrm{PWRTE}}$ = 0	**$\overline{\mathrm{PWRTE}}$ = 1**	**$\overline{\mathrm{PWRTE}}$ = 0**	**$\overline{\mathrm{PWRTE}}$ = 1**	
XT、 HS 和 LP	TPWRT + 1024 • TOSC	1024 TOSC	TPWRT + 1024 • TOSC	1024 TOSC	1024 TOSC
RC、 EC 和 INTOSC	TPWRT		TPWRT		

表 **12-2**： **STATUS/PCON** 位及其含义

$\overline{\mathrm{POR}}$	**$\overline{\mathrm{BOR}}$**	**$\overline{\mathrm{TO}}$**	**$\overline{\mathrm{PD}}$**	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 唤醒
u	u	u	u	正常工作期间的 $\overline{\mathrm{MCLR}}$ 复位
u	u	1	0	休眠期间的 $\overline{\mathrm{MCLR}}$ 复位

图注： u = 不变， x = 未知

表 **12-3**： 与欠压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	POR/BOR 时的值	所有其他复位时的值 [1]
PCON	—	—	—	—	—	—	$\overline{\mathrm{POR}}$	$\overline{\mathrm{BOR}}$	`---- --qq`	`---- --uu`
STATUS	IRP	RP1	RP0	$\overline{\mathrm{TO}}$	$\overline{\mathrm{PD}}$	Z	DC	C	`0001 1xxx`	`000q quuu`

图注： u = 不变， x = 未知， – = 未实现位（读为 `0`）， q = 值取决于具体条件。 BOR 不使用阴影单元。

注 1： 其他（非上电）复位包括在正常操作期间的 $\overline{\mathrm{MCLR}}$ 复位和看门狗定时器复位。

图 12-4：　上电时的延时时序 （$\overline{\text{MCLR}}$ 延时）：情形 1

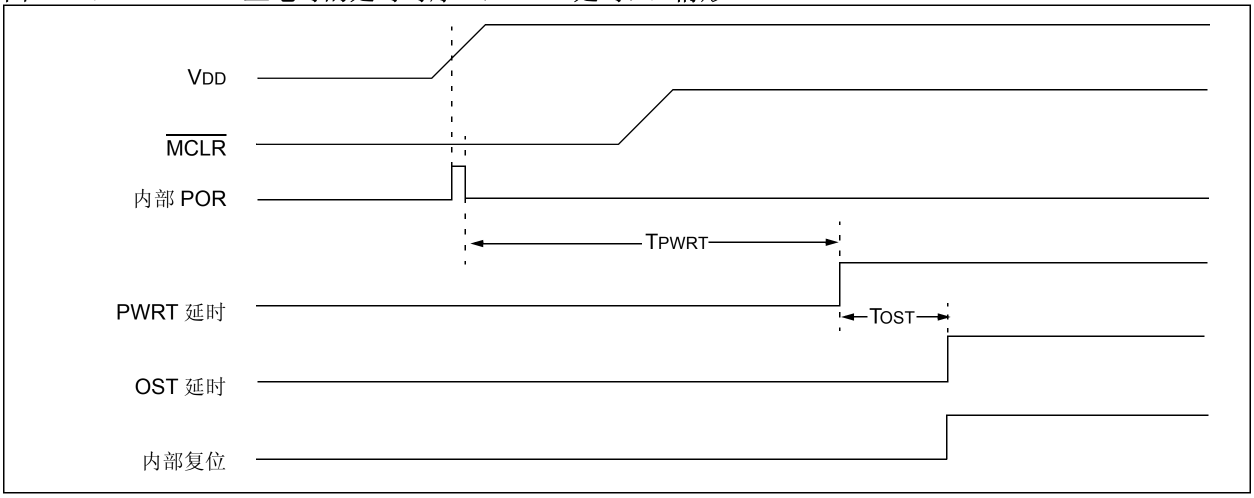


图 12-5：　上电时的延时时序 （$\overline{\text{MCLR}}$ 延时）：情形 2

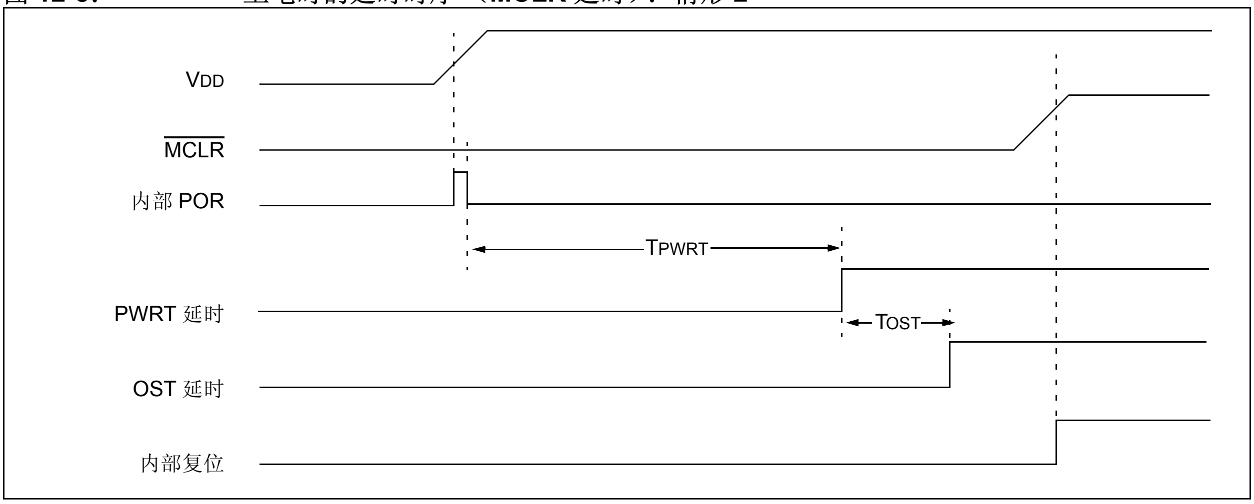


图 12-6：　上电时的延时时序 （带有 VDD 的 $\overline{\text{MCLR}}$）

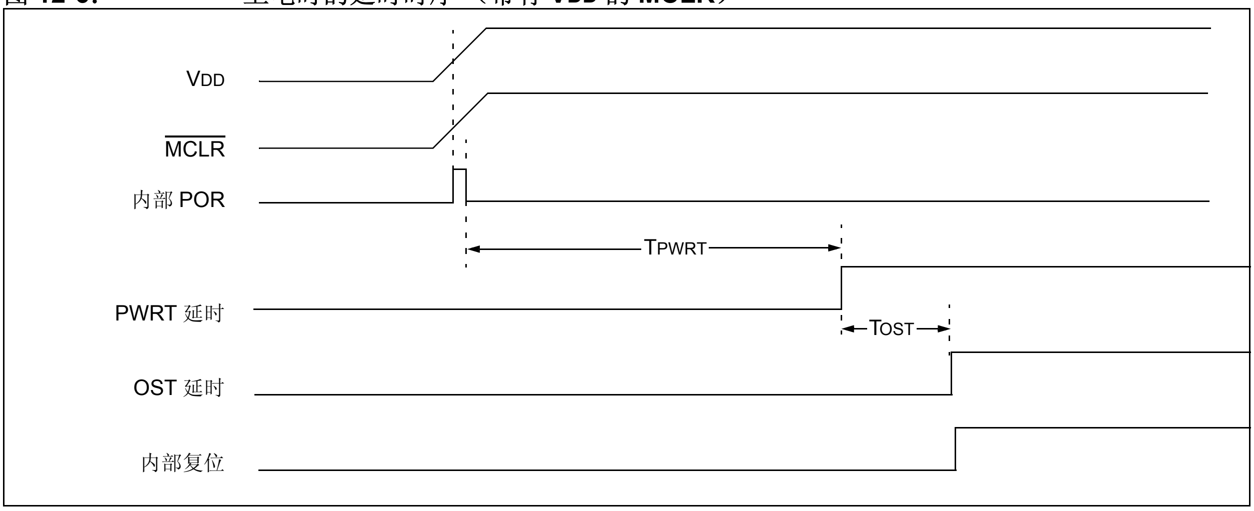


表 12-4： 寄存器的初始化条件

寄存器	地址	上电复位	MCLR 复位 WDT 复位 欠压复位 (1)	通过中断从休眠状态唤醒 通过 WDT 延时从休眠状态唤醒
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h/80h	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h/82h	0000 0000	0000 0000	PC + 1(3)
STATUS	03h/83h	0001 1xxx	000q quuu(4)	uuuq quuu(4)
FSR	04h/84h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	--x0 x000	--u0 u000	--uu uuuu
PORTC	07h	--xx xx00	--uu 00uu	--uu uuuu
PCLATH	0Ah/8Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh/8Bh	0000 0000	0000 0000	uuuu uuuu(2)
PIR1	0Ch	-000 0-00	-000 0-00	-uuu u-uu(2)
TMR1L	0Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	0Fh	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	10h	0000 0000	uuuu uuuu	-uuu uuuu
TMR2(6)	11h	0000 0000	0000 0000	uuuu uuuu
T2CON(6)	12h	-000 0000	-000 0000	-uuu uuuu
CCPR1L(6)	13h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H(6)	14h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON(6)	15h	0000 0000	0000 0000	uuuu uuuu
PWM1CON(6)	16h	0000 0000	0000 0000	uuuu uuuu
ECCPAS(6)	17h	0000 0000	0000 0000	uuuu uuuu
VRCON	19h	0000 0000	0000 0000	uuuu uuuu
CM1CON0	1Ah	0000 -000	0000 -000	uuuu -uuu
CM2CON0	1Bh	0000 -000	0000 -000	uuuu -uuu
CM2CON1	1Ch	00-0 0000	00-0 0000	uu-u uuuu
ADRESH(6)	1Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0(6)	1Fh	0000 0000	0000 0000	uuuu uuuu
OPTION_REG	81h	1111 1111	1111 1111	uuuu uuuu
TRISA	85h	--11 1111	--11 1111	--uu uuuu
TRISC	87h	--11 1111	--11 1111	--uu uuuu
PIE1	8Ch	-000 0-00	-000 0-00	-uuu u-uu
PCON	8Eh	---- --0x	---- --uu(1, 5)	---- --uu
OSCTUNE	90h	---0 0000	---u uuuu	---u uuuu
ANSEL	91h	1111 1111	1111 1111	uuuu uuuu
PR2(6)	92h	1111 1111	1111 1111	1111 1111
WPUA	95h	--11 -111	--11 -111	--uu -uuu

图注： u = 不变， x = 未知， – = 未实现位 （读为 0）， q = 值取决于具体条件。

注 1： 如果 VDD 过低，将激活上电复位，寄存器将受到不同的影响。
2： INTCON 和 / 或 PIR1 寄存器中的 1 位或多位会受到影响 （引起唤醒）。
3： 当器件被中断唤醒且 GIE 位置 1 时， PC 装入中断向量 （0004h）。
4： 关于特定条件下的复位值，请参见表 12-5。
5： 如果复位是由于欠压引起的，则 bit 0 = 0。所有其他复位将导致 bit 0 = u。
6： 仅限 PIC16F616/16HV616。

表 12-4： 寄存器的初始化条件 （续）

寄存器	地址	上电复位	$\overline{MCLR}$ 复位 WDT 复位 （续） 欠压复位 (1)	通过中断从休眠状态唤醒 通过 WDT 延时从休眠状态唤醒 （续）
IOCA	96h	--00 0000	--00 0000	--uu uuuu
SRCON0	99h	0000 00-0	0000 00-0	uuuu uu-u
SRCON1	9Ah	00-- ----	00-- ----	uu-- ----
ADRESL(6)	9Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1(6)	9Fh	-000 ----	-000 ----	-uuu ----

图注： u = 不变， x = 未知， – = 未实现位 （读为 0）， q = 值取决于具体条件。

注 1： 如果 VDD 过低，将激活上电复位，寄存器将受到不同的影响。
2： INTCON 和 / 或 PIR1 寄存器中的 1 位或多位会受到影响 （引起唤醒）。
3： 当器件被中断唤醒且 GIE 位置 1 时， PC 装入中断向量 （0004h）。
4： 关于特定条件下的复位值，请参见表 12-5。
5： 如果复位是由于欠压引起的，则 bit 0 = 0。所有其他复位将导致 bit 0 = u。
6： 仅限 PIC16F616/16HV616。

表 12-5： 特殊寄存器的初始化条件

条件	程序计数器	状态寄存器	PCON 寄存器
上电复位	000h	0001 1xxx	---- --0x
正常工作期间的 $\overline{MCLR}$ 复位	000h	000u uuuu	---- --uu
休眠期间的 $\overline{MCLR}$ 复位	000h	0001 0uuu	---- --uu
WDT 复位	000h	0000 uuuu	---- --uu
WDT 唤醒	PC + 1	uuu0 0uuu	---- --uu
欠压复位	000h	0001 1uuu	---- --10
通过中断从休眠状态唤醒	PC + 1(1)	uuu1 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 位置 1 时，执行 PC + 1 后， PC 装入中断向量 （0004h）。

12.4 中断

PIC16F610/616/16HV610/616 有以下多种中断源：

- 外部中断 RA2/INT
- Timer0 溢出中断
- PORTA 电平变化中断
- 两个比较器中断
- A/D 中断 （仅限 PIC16F616/16HV616）
- Timer1 溢出中断
- Timer2 匹配中断 （仅限 PIC16F616/16HV616）
- 增强型 CCP 中断 （仅限 PIC16F616/16HV616）

中断控制寄存器 （INTCON）和外设中断请求寄存器 1 （PIR1）在各自的标志位中记录各种中断请求。INTCON 寄存器还包括各个中断允许位和全局中断允许位。

INTCON 寄存器的全局中断允许位 GIE 允许 （如果置 1）所有未屏蔽的中断，而禁止（如果清零）所有中断。可以通过 INTCON 寄存器和 PIE1 寄存器中相应的允许位来禁止各个中断。复位时 GIE 被清零。

当响应一个中断时，将自动发生以下操作：

- 将 GIE 位清零以禁止其他中断。
- 将返回地址压入堆栈。
- PC 中装入 0004h。

执行 “从中断返回” 指令 `RETFIE` 退出中断服务程序并将 GIE 位置 1，从而重新允许未屏蔽的中断。

INTCON 寄存器包含以下中断标志位：

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

外设中断标志位在特殊寄存器 PIR1 中。相应的中断允许位在特殊寄存器 PIE1 中。

PIR1 寄存器包含以下中断标志位：

- A/D 中断
- 两个比较器中断
- Timer1 溢出中断
- Timer2 匹配中断
- 增强型 CCP 中断

对于外部中断事件，例如 INT 引脚或 PORTA 电平变化中断，中断响应延时将会是 3 到 4 个指令周期。确切的延时时间取决于发生中断事件的时间（见图 12-8）。对于单周期或双周期指令，中断响应延时完全相同。进入中断服务程序之后，就可以通过查询中断标志位来确定中断源。在重新允许中断前，必须用软件将中断标志位清零，以避免重复响应该中断。

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

关于 Timer1、Timer2、比较器、ADC、增强型 CCP 模块的更多信息，请参见相应的外设章节。

12.4.1 RA2/INT 中断

RA2/INT 引脚上的外部中断是边沿触发的；当 OPTION 寄存器的 INTEDG 位被置 1 时在上升沿触发，而当 INTEDG 位被清零时在下降沿触发。当 RA2/INT 引脚上出现有效边沿时， INTCON 寄存器的 INTF 位置 1。可以通过将 INTCON 寄存器的 INTE 控制位清零来禁止该中断。在重新允许该中断前，必须在中断服务程序中先用软件将 INTF 位清零。如果 INTE 位在进入休眠状态前被置 1，则 RA2/INT 中断能将处理器从休眠状态唤醒。关于休眠的详细信息，请参见**第 12.7 节 “掉电模式（休眠）”**；关于 RA2/INT 中断将处理器从休眠状态唤醒的时序，请参见图 12-9。

注：	必须对 ANSEL 寄存器进行初始化，以将模拟通道配置为数字输入。配置为模拟输入的引脚将读为 `0`，且不能产生中断。

12.4.2 TIMER0 中断

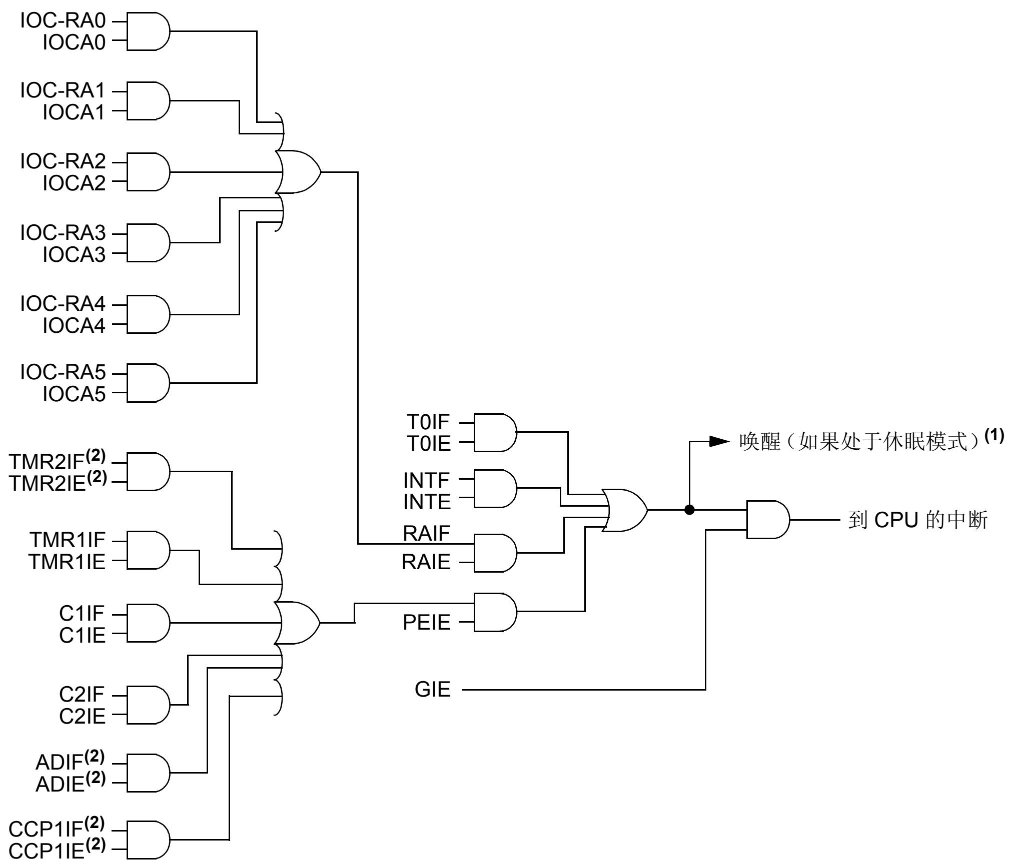
TMR0 寄存器溢出（FFh → 00h）会将 INTCON 寄存器的 T0IF 位置 1。可以通过置 1/ 清零 INTCON 寄存器的 T0IE 位来允许 / 禁止该中断。关于 Timer0 模块的操作，请参见**第 5.0 节 “Timer0 模块”**。

12.4.3 PORTA 电平变化中断

PORTA 输入电平的变化会使 INTCON 寄存器的 RAIF 位置 1。可以通过置 1/ 清零 INTCON 寄存器的 RAIE 位来允许 / 禁止该中断。此外，可通过 IOCA 寄存器对该端口的各个引脚进行配置。

> **注：** 当正在执行任何 PORTA 操作时发生了 I/O 引脚电平变化，则 RAIF 中断标志位可能不会被置 1。

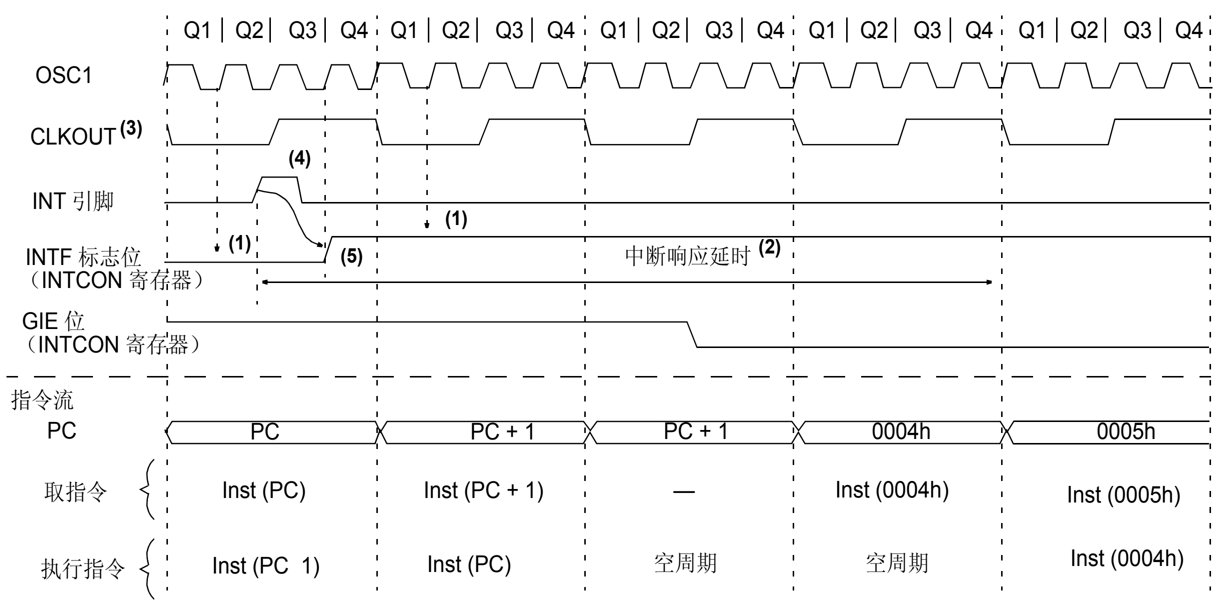
图 12-7： 中断逻辑



注 1： 某些外设取决于工作的系统时钟。由于系统时钟在休眠期间暂停，所以只有那些不依靠系统时钟进行工作的外设会将器件从休眠状态唤醒。请参见**第 12.7.1 节 “从休眠状态唤醒”**。

2： 仅限 PIC16F616/16HV616。

图 12-8： INT 引脚中断时序



注 1：在此时采样 INTF 标志 （每个 Q1 周期）。

2：异步中断响应延时为 3-4 个 TCY。同步中断响应延时为 3 个 TCY，其中 TCY 为一个指令周期。不管 Inst (PC) 是单周期还是双周期指令，中断响应延时都是一样的。

3：只有在 INTOSC 和 RC 振荡器模式下 CLKOUT 才有效。

4：关于 INT 脉冲的最小宽度，请参见**第 15.0 节 “电气规范”**中的交流规范。

5：允许在 Q4-Q1 周期内的任何时间将 INTF 置 1。

表 12-6： 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	POR/BOR 时的值	所有其他复位时的值
INTCON	GIE	PEIE	T0IE	INTE	RAIE	T0IF	INTF	RAIF	0000 0000	0000 0000
IOCA	—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	--00 0000	--00 0000
PIR1	—	ADIF(1)	CCP1IF(1)	C2IF	C1IF	—	TMR2IF(1)	TMR1IF	-000 0-00	-000 0-00
PIE1	—	ADIE(1)	CCP1IE(1)	C2IE	C1IE	—	TMR2IE(1)	TMR1IE	-000 0-00	-000 0-00

图注： x = 未知， u = 不变， - = 未实现 （读为 0）， q = 值取决于具体条件。中断模块不使用阴影单元。

注 1： 仅限 PIC16F616/16HV616。

 初稿

12.5 中断的现场保护

在中断期间，仅将返回 PC 值压入堆栈。通常情况下，用户可能希望在中断期间保存关键寄存器（例如，W 和 STATUS 寄存器）。这必须用软件实现。

临时保存寄存器 W_TEMP 和 STATUS_TEMP 都应该放在 GPR 的后 16 个字节中 （见图 2-4）。这 16 个存储单元对所有存储区都是公用的，而且不需要分区，因此便于现场保护和恢复。例 12-1 中所示的代码可被用于：

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

注：	PIC16F610/616/16HV610/616 通常不需要保存 PCLATH。但是，如果要在 ISR 和主代码中同时使用计算 `GOTO`，就必须在 ISR 中保存和恢复 PCLATH。

例 12-1： 在 RAM 中保存 STATUS 寄存器和 W 寄存器

```
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
```

12.6 看门狗定时器 （WDT）

看门狗定时器是自由运行的片上 RC 振荡器，不需要外部元件。该 RC 振荡器独立于 CLKIN 引脚的外部 RC 振荡器和 INTOSC。这意味着即使器件 OSC1 和 OSC2 引脚上的时钟已经停止， WDT 也会运行 （例如，通过执行一条 SLEEP 指令）。在正常操作期间， WDT 超时会产生器件复位。如果器件处于休眠模式， WDT 超时会引起器件唤醒并继续正常操作。可通过将配置位 WDTE 清零来永久禁止 WDT （第 **12.1** 节 **“配置位”**）。

12.6.1 WDT 周期

WDT 超时溢出周期的标称值为 18 ms（无预分频器）。超时溢出周期随温度、VDD 和不同器件制造工艺的差异而不同 （见表 15-4 中的参数 31）。如果需要更长的超时溢出周期，可通过写入 OPTION 寄存器将分频比最高可达 1:128 的预分频器分配给用软件控制的 WDT。因此，可实现最长 2.3 秒的超时溢出周期。

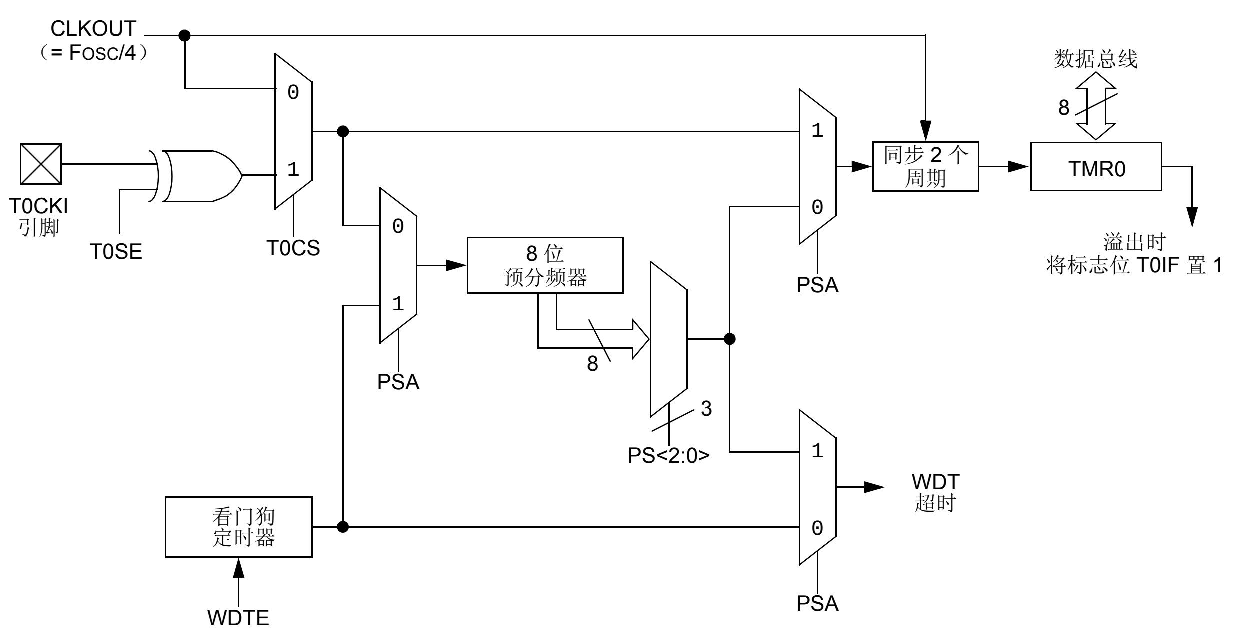
如果将预分频器分配给 WDT， CLRWDT 和 SLEEP 指令会清零 WDT 和预分频器，并防止其超时以及产生器件复位。

一旦看门狗定时器超时，STATUS 寄存器中的 $\overline{\text{TO}}$ 位就会被清零。

12.6.2 WDT 编程注意事项

还应考虑最坏情况下的结果（即，VDD = 最小值，温度 = 最大值，最大 WDT 预分频比），在 WDT 超时发生前可能需要几秒钟。

图 12-2： 看门狗定时器框图



注 1：T0SE、 T0CS、 PSA 和 PS<2:0> 是 OPTION 寄存器中的位。

表 12-7： WDT 状态

条件	WDT
WDTE = 0	清零
CLRWDT 指令	
退出休眠 + 系统时钟 = EXTRC、 INTRC 或 EC	
退出休眠 + 系统时钟 = XT、 HS 或 LP	清零直到 OST 结束

表 12-8： 与看门狗定时器相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
OPTION_REG	$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
CONFIG[(1)]	IOSCFS	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0	—	—

图注： 看门狗定时器不使用阴影单元。
注 1： 关于配置字寄存器中所有位的操作，请参见寄存器 12-1。

12.7 掉电模式（休眠）

通过执行 SLEEP 指令可进入掉电模式。

如果使能看门狗定时器：

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

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

$\overline{MCLR}$ 引脚必须处于逻辑高电平。

注：	应该注意到WDT超时导致的复位并不会将 $\overline{MCLR}$ 引脚驱动为低电平。

12.7.1 从休眠状态唤醒

可以通过以下任一事件将器件从休眠状态唤醒：

1. $\overline{MCLR}$ 引脚上的外部复位输入。
2. 看门狗定时器唤醒（如果 WDT 使能）。
3. RA2/INT 引脚中断、 PORTA 电平变化中断或外设中断。

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

下列外设中断可以将器件从休眠状态唤醒：

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

由于在休眠期间没有片上时钟处于工作状态，因此其他外设不能产生中断。

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

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

器件从休眠状态唤醒时， WDT 都将被清零，而与唤醒原因无关。

12.7.2 使用中断唤醒

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

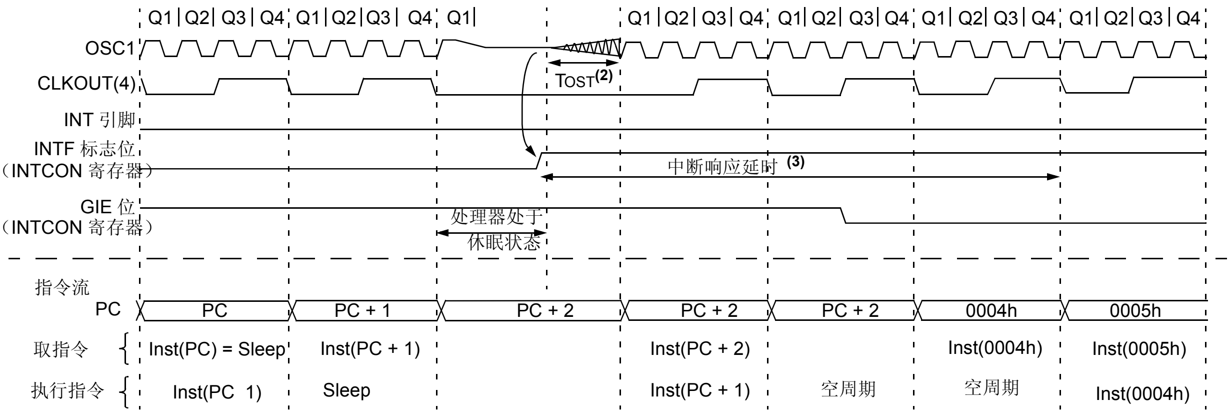
- 如果在执行 SLEEP 指令**之前**产生了中断，那么 SLEEP 指令将被作为一条 NOP 指令执行。因此，WDT 及其预分频器和后分频器（如果使能）将不会被清零，并且 $\overline{TO}$ 位将不会被置 1，同时 $\overline{PD}$ 位也不会被清零。
- 如果在执行SLEEP指令**期间或之后**产生了中断，那么器件将被立即从休眠状态唤醒。执行了 SLEEP 指令，因此， WDT 及其预分频器和后分频器（如果使能）将被清零，并且 $\overline{TO}$ 位将被置 1，同时 $\overline{PD}$ 位也将被清零。

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

在执行SLEEP指令之前，必须先执行一条CLRWDT指令，来确保将 WDT 清零。更多详细信息，请参见图 12-9。

 初稿

图 12-9： 通过中断将器件从休眠状态唤醒



注 1： 假设为 XT、 HS 或 LP 振荡器模式。
2： TOST = 1024 TOSC （图中未按比例绘制）。该延时不适用于 EC、 INTOSC 和 RC 振荡器模式。
3： 假设 GIE = 1。在这种情形下，处理器被唤醒后，将跳转到 0004h 处执行代码。如果 GIE = 0，程序将继续执行。
4： 在 XT、 HS、 LP 或 EC 振荡器模式下，不输出 CLKOUT 信号，在此处仅作为时序参考。

12.8 代码保护

如果代码保护位未编程，可以通过使用ICSP™将片上程序存储器读出来进行校验。

注：	当关闭代码保护功能时，将擦除整个闪存程序存储器的内容。更多信息，请参见 *“PIC12F60X/12F61X/16F61X Memory Programming Specification”* （DS41284）。

12.9 ID 存储单元

有 4 个存储器单元 （2000h-2003h）被指定为 ID 存储单元，供用户存储校验和或其他编码标识号。在正常执行过程中不能访问这些存储单元，但可在编程 / 校验模式下对它们进行读写。只使用了 ID 存储单元的低 7 位。

12.10 在线串行编程

可在最终的应用电路中对 PIC16F610/616/16HV610/616 单片机进行串行编程。只需要以下 5 根线即可实现这一操作：

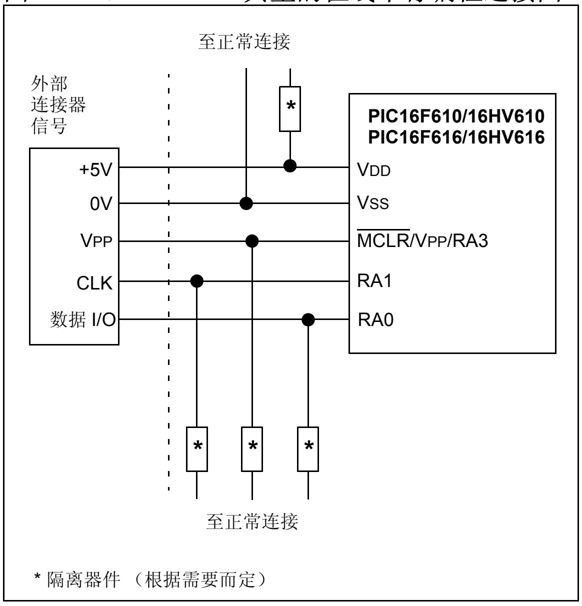
- 时钟线
- 数据线
- 电源线
- 接地线
- 编程电压线

这允许用户在生产电路板时使用未编程器件，仅在产品交付之前才对单片机进行编程，从而可以使用最新版本的固件或者定制固件进行编程。

通过将RA0和RA1引脚保持为低电平，并同时将 $\overline{\text{MCLR}}$（VPP）引脚电平从 VIL 上升到 VIHH，可将器件置于编程/校验模式。更多信息，请参见"*PIC12F60X/12F61X/16F61X Memory Programming Specification*"（DS41284）。RA0 变成编程数据引脚，而 RA1 变成编程时钟引脚。在编程 / 校验模式下，RA0 和 RA1 均为施密特触发器输入引脚。

图 12-10 给出了典型的在线串行编程连接图。

图 12-10： 典型的在线串行编程连接图



注：	要进行擦除，器件 VDD 必须高于"*PIC12F615/12HV615/16F616/16HV616 Memory Programming Specification*"*（DS41284）*中规定的批量擦除的最小 VDD。

12.11 在线调试器

由于在线调试需要访问三个引脚，用 MPLAB® ICD 2 调试 14 引脚器件是不实际的。一款特殊的 28 引脚 PIC16F610/616/16HV610/616 ICD 器件与 MPLAB ICD 2 配合使用，可以提供独立的时钟线、数据线和 $\overline{\text{MCLR}}$ 引脚，因此释放了所有通常可用的引脚供用户使用。

有一个特殊的调试适配器使得 ICD 器件可以代替 **PIC16F610/616/16HV610/616** 器件使用。该调试适配器是 ICD 器件的唯一来源。

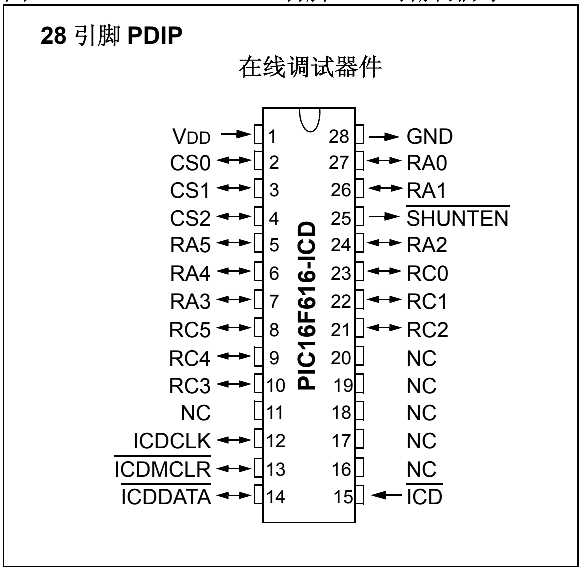
当 PIC16F610/616/16HV610/616 ICD 器件上的 $\overline{\text{ICD}}$ 引脚保持低电平时，在线调试功能使能。这一功能在与 MPLAB ICD 2 配合使用时，即可进行一些简单的调试。当单片机的这项功能使能时，某些资源就不能用于一般用途了。表 12-9 给出了后台调试器所占用的资源。

表 12-9： 调试器资源

资源	说明
I/O 引脚	ICDCLK 和 ICDDATA
堆栈	1 级深度
程序存储器	地址 0h 必须为 NOP 700h-7FFh

更多信息，请参见 *《MPLAB® ICD 2 在线调试器用户指南》*（DS51331A_CN），该文档可在 Microchip 网站（www.microchip.com）上找到。

图 12-11： 28 引脚 ICD 引脚排列



13.0 指令集汇总

PIC16F610/616/16HV610/616 指令集具有高度正交性，由以下三种基本类型的指令组成：

- **字节**操作类指令
- **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**（进一步指定指令操作）组成。图 13-1 给出了上述各种类型的指令格式，表 13-1 给出了不同操作码字段的说明。

表 13-2 列出了 MPASM™ 汇编器可识别的指令。

对于**字节**操作类指令，“f”表示文件寄存器标识符，“d”表示目标寄存器标识符。文件寄存器标识符指定了指令将会使用哪一个文件寄存器。

目标标识符指定了操作结果的存放位置。如果“d”为 0，操作结果存入 W 寄存器。如果“d”为 1，操作结果存入指令指定的文件寄存器中。

对于**位**操作类指令，“b”表示位域标识符，它可选择受到操作影响的位，而“f”表示该位所处的文件寄存器地址。

对于**立即数和控制**操作类指令，“k”表示一个 8 位或 11 位的常数或立即数。

每个指令周期由 4 个振荡器周期组成。因此，如果振荡器频率为 4 MHz，正常的指令执行时间为 1 μs。除非条件测试结果为真或指令执行改变了程序计数器的值，否则执行所有的指令都只需要一个指令周期。对于上述两种特殊情况，指令执行需要两个指令周期，在第二个指令周期中执行一条 `NOP` 指令。

所有指令示例均使用格式“`0xhh`”来表示一个十六进制数，其中“h”表示一个十六进制数字。

13.1 读－修改－写操作

任何一条指定文件寄存器作为指令一部分的指令都进行读－修改－写（RMW）操作。读寄存器、修改数据并根据指令或目标标识符“d”存储结果。即使指令写入该寄存器，还是会发生对寄存器的读操作。

例如，`CLRF PORTA` 指令将读取 PORTA，清除所有数据位，然后把结果写回到 PORTA。这个示例会得到一个意外结果，即清除将 RAIF 标志位置 1 的条件。

表 13-1： 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	无关位（= `0` 或 `1`）。 汇编器将产生 x = `0` 的代码。为了与所有的 Microchip 软件工具兼容，建议使用这种形式。
d	目标寄存器选择；d = `0`：结果存入 W， d = `1`：结果存入文件寄存器 f。 缺省值 d = `1`。
PC	程序计数器
$\overline{TO}$	超时位
C	进位位
DC	半进位位
Z	全零位
$\overline{PD}$	掉电位

图 13-1： 指令的通用格式

表 13-2： PIC16F610/616/16HV610/616 指令集

助记符，操作数		说明	周期	14 位操作码 MSb			LSb	受影响的状态位	注
针对字节的文件寄存器操作									
ADDWF	f, d	W 与 f 相加	1	00	0111	dfff	ffff	C, DC, 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 清零	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	对 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 和 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 内容传送到 f	1	00	0000	lfff	ffff		
NOP		空操作	1	00	0000	0xx0	0000		
RLF	f, d	f 带进位循环左移	1	00	1101	dfff	ffff	C	**1, 2**
RRF	f, d	f 带进位循环右移	1	00	1100	dfff	ffff	C	**1, 2**
SUBWF	f, d	f 减去 W	1	00	0010	dfff	ffff	C, DC, 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 和 f 作逻辑异或运算	1	00	0110	dfff	ffff	Z	**1, 2**
针对位的文件寄存器操作									
BCF	f, b	将 f 寄存器中的某位清零	1	01	00bb	bfff	ffff		**1, 2**
BSF	f, b	将 f 寄存器中的某位置 1	1	01	01bb	bfff	ffff		**1, 2**
BTFSC	f, b	测试 f 中的某位，为 0 则跳过	1 **(2)**	01	10bb	bfff	ffff		**3**
BTFSS	f, b	测试 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	立即数与 W 相加	1	11	111x	kkkk	kkkk	C, DC, Z	
ANDLW	k	立即数和 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		将看门狗定时器清零	1	00	0000	0110	0100	$\overline{TO}$, $\overline{PD}$	
GOTO	k	跳转到地址	2	10	1kkk	kkkk	kkkk		
IORLW	k	立即数和 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	返回并将立即数送入 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	$\overline{TO}$, $\overline{PD}$	
SUBLW	k	立即数减去 W	1	11	110x	kkkk	kkkk	C, DC, Z	
XORLW	k	立即数和 W 作逻辑异或运算	1	11	1010	kkkk	kkkk	Z	

注 **1：** 当 I/O 寄存器修改自身时（例如，MOVF PORTA, 1），所使用的值是引脚上的当前值。例如，如果将一引脚配置为输入，且对应数据锁存器中的值为 1，但此时若有外部器件将该引脚驱动为低电平，则被写回数据锁存器的数据值将是 0。

2： 当对 TMR0 寄存器执行该指令（并且 d = 1）时，如果已为 Timer0 模块分配了预分频器，则将该预分频器清零。

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

13.2 指令说明

ADDLW	**立即数与 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，结果存储在 W 寄存器中。

ADDWF	**W 与 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 寄存器的内容与寄存器 f 的内容相加。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

ANDLW	**立即数和 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 寄存器中的内容与 8 位立即数 k 进行逻辑与运算。结果存入 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：	将 W 寄存器的内容与寄存器 f 的内容进行逻辑与运算。如果 d 为 0，结果存入 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 寄存器中的某位置 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 的位 b 置 1。

BTFSC	**测试 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：	如果寄存器 f 的位 b 为 1，则执行下一条指令。 如果寄存器 f 的位 b 为 0，则丢弃下一条指令转而执行一条 NOP 指令，使之成为一条双周期指令。

BTFSS	测试 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：	如果寄存器 f 的位 b 为 0，则执行下一条指令。 如果位 b 为 1，则丢弃下一条指令转而执行一条 NOP 指令，使之成为一条双周期指令。

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。 11 位直接地址值被装入 PC 的 <10:0> 位。PC 的高位值从 PCLATH 装入。 CALL 是一条双周期指令。

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：	寄存器 f 的内容被清零，并且 Z 位被置 1。

CLRW	将 W 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：	W 寄存器被清零。全零位 （Z）被置 1。

CLRWDT	将看门狗定时器清零
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及其预分频器。 状态位 $\overline{TO}$ 和 $\overline{PD}$ 均被置 1。

COMF	对 f 取反
语法：	[*标号*] COMF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	$(\overline{f})$ → （目标寄存器）
受影响的状态位：	Z
说明：	将寄存器 f 的内容取反。如果 d 为 0，结果存入 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：	将寄存器 f 的内容递减 1。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

初稿

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）， 如果结果 = 0 则跳过
受影响的状态位：	无
说明：	将寄存器 f 的内容递减 1。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。 如果结果为 1，则执行下一条指令。如果结果为 0，代之执行一条 NOP 指令，使之成为一条双周期指令。

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 是一条无条件转移指令。 11 位立即数被装入 PC 的 <10:0> 位。 PC 的高位从 PCLATH<4:3> 装入。 GOTO 是一条双周期指令。

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：	将寄存器 f 的内容递增 1。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

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）， 如果结果 = 0 则跳过
受影响的状态位：	无
说明：	将寄存器 f 的内容递增 1。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。 如果结果为 1，则执行下一条指令。如果结果为 0，代之执行一条 NOP 指令，使之成为一条双周期指令。

IORLW	立即数和 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 进行逻辑或运算。结果存入 W 寄存器。

IORWF	W 和 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：	将 W 寄存器的内容与寄存器 f 的内容进行逻辑或运算。如果 d 为 0，结果存入 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，将寄存器 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，目标寄存器为文件寄存器 f 本身。由于状态标志位 Z 要受影响，可用 d = 1 对文件寄存器进行检测。
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 W 寄存器的数据传送到寄存器 f。
指令字数：	1
指令周期数：	1

示例：	`MOVW OPTION`
	`F`
	执行指令前
	OPTION = 0xFF
	W = 0x4F
	执行指令后
	OPTION = 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。其余无关位均汇编为 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：	不执行任何操作。
指令字数：	1
指令周期数：	1
示例：	`NOP`

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（Top-of-Stack， TOS）的内容装入 PC。通过将全局中断允许位 GIE （INTCON<7>）置 1，来允许中断。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 返回并将立即数送入 **W**

语法：	[*标号*] RETLW k
操作数：	$0 \leq k \leq 255$
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 **f 带进位循环左移**

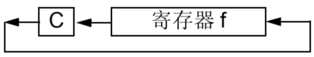
语法： [*标号*] RLF f,d

操作数： 0 ≤ f ≤ 127
d ∈ [0,1]

操作： 参见如下说明

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，结果存入 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

<u>示例：</u>
```
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
```

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 → $\overline{\text{TO}}$，
0 → $\overline{\text{PD}}$

受影响的状态位： $\overline{\text{TO}}$ 和 $\overline{\text{PD}}$

说明： 掉电状态位 $\overline{\text{PD}}$ 被清零。超时状态位 $\overline{\text{TO}}$ 被置 1。看门狗定时器及其预分频器被清零。
振荡器停振，处理器进入休眠模式。

RRF **f 带进位循环右移**

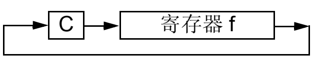
语法： [*标号*] RRF f,d

操作数： 0 ≤ f ≤ 127
d ∈ [0,1]

操作： 参见如下说明

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，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。



SUBLW **立即数减去 W**

语法： [*标号*] SUBLW k

操作数： 0 ≤ k ≤ 255

操作： k - (W) → (W)

受影响的状态位： C、 DC 和 Z

说明： 用 8 位立即数 k 减去 W 寄存器的内容 （通过二进制补码方式进行运算）。结果存入 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>

 初稿

SUBWF	f 减去 W
语法：	[*标号*] SUBWF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(f) - (W) → （目标寄存器）
受影响的状态位：	C、 DC 和 Z
说明：	用寄存器 f 的内容减去 W 寄存器的内容 （通过二进制补码方式进行运算）。如果 d 为 0，结果存入 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 的高半字节和低半字节交换
语法：	[*标号*] SWAPF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [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：	寄存器 f 的高半字节和低半字节相互交换。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

XORLW	立即数和 W 作逻辑异或运算
语法：	[*标号*] XORLW k
操作数：	0 ≤ k ≤ 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 进行逻辑异或运算。结果存入 W 寄存器。

XORWF	W 和 f 作逻辑异或运算
语法：	[*标号*] XORWF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(W) .XOR.(f) → （目标寄存器）
受影响的状态位：	Z
说明：	将 W 寄存器的内容与寄存器 f 的内容进行逻辑异或运算。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

注：

14.0 开发支持

一系列硬件及软件开发工具对 PIC® 单片机提供支持：

- 集成开发环境
  - MPLAB® IDE 软件
- 汇编器 / 编译器 / 链接器
  - MPASM™ 汇编器
  - MPLAB C18 和 MPLAB C30 C 编译器
  - MPLINK™ 目标链接器 / MPLIB™ 目标库管理器
  - MPLAB ASM30 汇编器 / 链接器 / 库
- 模拟器
  - MPLAB SIM 软件模拟器
- 仿真器
  - MPLAB ICE 2000 在线仿真器
  - MPLAB REAL ICE™ 在线仿真器
- 在线调试器
  - MPLAB ICD 2
- 器件编程器
  - PICSTART® Plus 开发编程器
  - MPLAB PM3 器件编程器
  - PICkit™ 2 开发编程器
- 低成本演示和开发板及评估工具包

14.1 MPLAB 集成开发环境软件

MPLAB IDE 软件为 8/16 位单片机市场提供了前所未有的易于使用的软件开发平台。MPLAB IDE 是基于 Windows® 操作系统的应用软件，包括：

- 一个包含所有调试工具的图形界面
  - 模拟器
  - 编程器 （单独销售）
  - 仿真器 （单独销售）
  - 在线调试器 （单独销售）
- 具有彩色上下文代码显示的全功能编辑器
- 多项目管理器
- 内容可直接编辑的可定制式数据窗口
- 高级源代码调试
- 可视化器件初始化程序，便于进行寄存器的初始化
- 鼠标停留在变量上进行查看的功能
- 通过拖放把变量从源代码窗口拉到观察窗口
- 丰富的在线帮助
- 集成了可选的第三方工具，如 HI-TECH 软件 C 编译器和 IAR C 编译器

MPLAB IDE 可以让您：

- 编辑源文件 （汇编语言或 C 语言）
- 点击一次即可完成汇编 （或编译）并将代码下载到 PIC MCU 仿真器和模拟器工具中 （自动更新所有项目信息）
- 可使用如下各项进行调试：
  - 源文件 （汇编语言或 C 语言）
  - 混合汇编语言和 C 语言
  - 机器码

MPLAB IDE 在单个开发范例中支持使用多种调试工具，包括从成本效益高的模拟器到低成本的在线调试器，再到全功能的仿真器。这样缩短了用户升级到更加灵活而功能更强大的工具时的学习时间。

14.2 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于所有的 PIC 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 汇编器具有如下特征：

- 集成在 MPLAB 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

14.3 MPLAB C18 和 MPLAB C30 C 编译器

MPLAB C18 和 MPLAB C30 代码开发系统是完全的 ANSI C 编译器，分别适用于 Microchip 的 PIC18 和 PIC24 系列单片机及 dsPIC30F 和 dsPIC33 系列数字信号控制器。这些编译器可提供其他编译器并不具备的强大的集成功能和出众的代码优化能力，且使用方便。

为便于源代码调试，编译器提供了针对 MPLAB IDE 调试器的优化符号信息。

14.4 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK 目标链接器包含了由 MPASM 汇编器、MPLAB C18 C 编译器产生的可重定位目标。通过使用链接器脚本中的指令，它还可链接预编译库中的可重定位目标。

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中。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征：

- 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

14.5 MPLAB ASM30 汇编器、链接器和库管理器

MPLAB ASM30 汇编器为 dsPIC30F 器件提供转换自符号汇编语言的可重定位机器码。 MPLAB C30 C 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征：

- 支持整个 dsPIC30F 指令集
- 支持定点数据和浮点数据
- 命令行界面
- 丰富的指令集
- 灵活的宏语言
- MPLAB IDE 兼容性

14.6 MPLAB SIM 软件模拟器

MPLAB SIM软件模拟器在指令级对PIC MCU和dsPIC® DSC 进行模拟，使得用户可以在 PC 主机的环境下进行代码开发。对于任何给定的指令，用户均可对数据区进行检查或修改，并通过各种触发机制来产生激励。可以将各寄存器的情况记录在文件中，以便进行进一步地运行时分析。跟踪缓冲器和逻辑分析器的显示使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的状况。

MPLAB SIM 软件模拟器完全支持使用 MPLAB C18 和 MPLAB C30 C 编译器以及 MPASM 和 MAPLAB ASM30 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。

14.7 MPLAB ICE 2000 高性能在线仿真器

MPLAB ICE 2000 在线仿真器旨在为产品开发工程师提供一整套用于 PIC 单片机的设计工具。 MPLAB ICE 2000在线仿真器的软件控制由MPLAB集成开发环境平台提供，它允许在单一环境下进行编辑、编译、下载以及源代码调试。

MPLAB ICE 2000 是全功能仿真器系统，它具有增强的跟踪、触发和数据监控功能。处理器模块可插拔，使系统可轻松进行重新配置以适应各种不同处理器的仿真需要。MPLAB ICE 2000 在线仿真器的架构允许对其进行扩展以支持新的 PIC 单片机。

MPLAB ICE 2000 在线仿真器系统设计为一款实时仿真系统，该仿真系统具备通常只有昂贵的开发工具中才有的高级功能。选择 PC 平台和 Microsoft® Windows® 32 位操作系统可使这些功能在一个简单而统一的应用中得到很好的利用。

14.8 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 器件而推出的新一代高速仿真器。结合 MPLAB 集成开发环境（IDE）所具有的易于使用且功能强大的图形用户界面，该仿真器可对 PIC® 闪存 MCU 和 dsPIC® DSC进行调试和编程。IDE是随每个工具包一起提供的。

MPLAB REAL ICE 探针通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与常用 MPLAB ICD 2 系统兼容的连接器 （RJ11）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 IDE 下载将来版本的固件，对 MPLAB REAL ICE 进行现场升级。在即将推出的 MPLAB IDE 版本中，会支持许多新器件，还将增加一些新特性，如软件断点和汇编代码跟踪等。在同类仿真器中， MPLAB REAL ICE 的优势十分明显：低成本、高速仿真、实时变量监视、跟踪分析、复杂断点、耐用的探针接口及较长（长达 3 米）的互连电缆。

14.9 MPLAB ICD 2 在线调试器

Microchip 的在线调试器 MPLAB ICD 2 是一款功能强大而成本低廉的运行时开发工具，通过 RS-232 或高速 USB 接口与 PC 主机相连。该工具基于闪存 PIC MCU，可用于开发本系列及其他 PIC MCU 和 dsPIC DSC。MPLAB ICD 2使用了闪存器件中内建的在线调试功能。该功能结合Microchip的在线串行编程（In-Circuit Serial Programming™， ICSP™）协议，可在 MPLAB 集成开发环境的图形用户界面上提供成本效益很高的在线闪存调试。这使设计人员可通过设置断点、单步运行以及对变量、CPU状态以及外设寄存器进行监视的方法实现源代码的开发和调试。其全速运行特性可对硬件和应用进行实时测试。 MPLAB ICD 2 还可用作某些 PIC 器件的开发编程器。

14.10 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款通用的、符合 CE 规范的器件编程器，其可编程电压设置在 V_{DDMIN} 和 V_{DDMAX} 之间时可靠性最高。它有一个用来显示菜单和错误信息的大 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 器件进行读取、验证和编程。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，可对存储器很大的器件进行快速编程，它还采用 SD/MMC 卡用作文件存储及数据安全应用。

14.11 PICSTART Plus 开发编程器

PICSTART Plus开发编程器是一款易于使用而成本低廉的原型编程器。它通过 COM （RS-232）端口与 PC 相连。 MPLAB 集成开发环境软件使得该编程器的使用简便、高效。PICSTART Plus 开发编程器支持采用 DIP 封装的大部分 PIC 器件，其引脚数最多可达 40 个。引脚数更多的器件，如 PIC16C92X 和 PIC17C76X，可通过连接一个转接插槽来获得支持。PICSTART Plus 开发编程器符合 CE 规范。

14.12 PICkit 2 开发编程器

PICkit™ 2 开发编程器是一个低成本编程器；对于某些选定闪存器件，它也是一个调试器，通过其易于使用的接口可对众多 Microchip 的低档、中档和 PIC18F 系列闪存单片机进行编程。 PICkit 2 入门工具包中包含一个有实验布线区的开发板、十二堂系列课程、软件和 HI-TECH 的 PICC™ Lite C 编译器，有助于用户快速掌握 PIC® 单片机的使用。这一工具包为使用 Microchip 功能强大的中档闪存系列单片机进行编程、评估和应用开发，提供了所需的一切。

14.13 演示、开发和评估板

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，用于测试和修改。

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。

有关演示、开发和评估工具包的完整列表，请查阅 Microchip 公司网页 （www.microchip.com）。

 初稿

15.0 电气规范

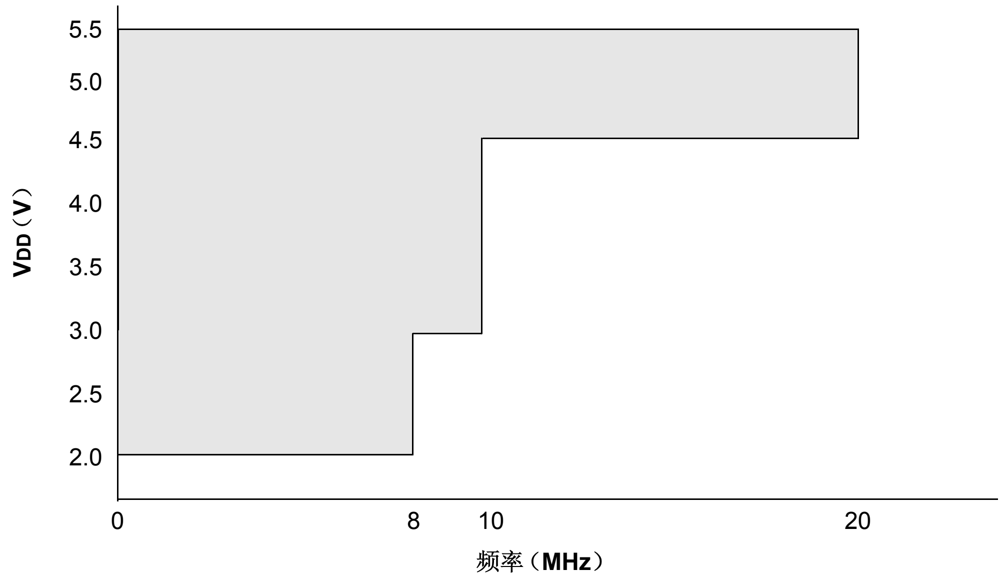
绝对最大额定值 (†)

环境温度 .... -40°C 至 +125°C
储存温度 .... -65°C 至 +150°C
V_{DD} 引脚相对于 V_{SS} 的电压 .... -0.3V 至 +6.5V
$\overline{MCLR}$ 引脚相对于 V_{SS} 的电压 .... -0.3V 至 +13.5V
所有其他引脚相对于 V_{SS} 的电压 .... -0.3V 至 （V_{DD} + 0.3V）
总功耗 (1) .... 800 mW
流出 V_{SS} 引脚的最大电流 .... 95 mA
流入 V_{DD} 引脚的最大电流 .... 95 mA
输入箝位电流， I_{IK} （$V_I < 0$ 或 $V_I > V_{DD}$） .... ± 20 mA
输出箝位电流， I_{OK} （$V_O < 0$ 或 $V_O > V_{DD}$） .... ± 20 mA
任一 I/O 引脚的最大输出灌电流 .... 25 mA
任一 I/O 引脚的最大输出拉电流 .... 25 mA
PORTA 和 PORTC （联合）最大灌电流 .... 90 mA
PORTA 和 PORTC （联合）最大拉电流 .... 90 mA

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

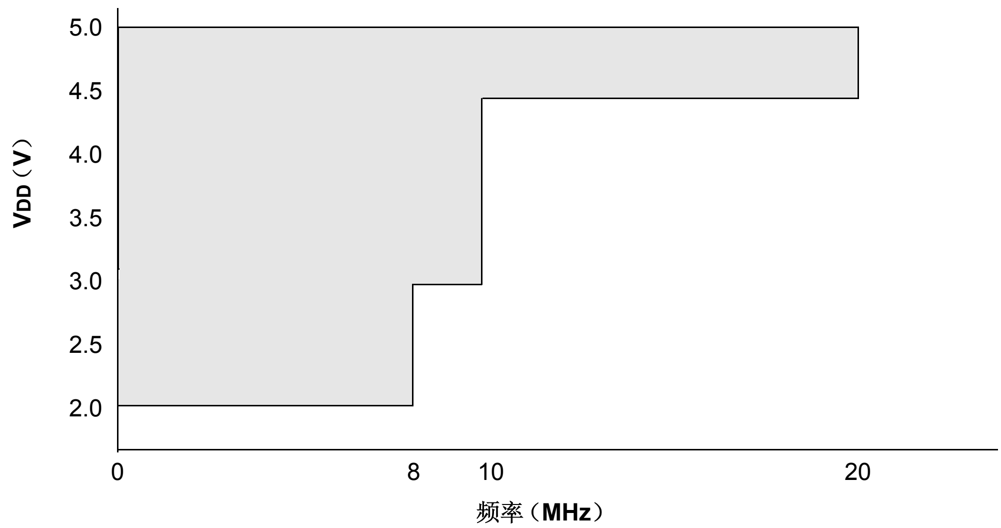
† 注：如果运行条件超过了上述 “绝对最大额定值”，即可能对器件造成永久性损坏。上述值仅为运行条件的极大值，我们不建议器件运行在该规范范围以外。器件长时间工作在高于绝对最大额定值条件下，其稳定性可能受到影响。

图 15-1： PIC16F610/616 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频率组合。

图 15-2： PIC16HV610/616 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频率组合。

 初稿

图 **15-3**： **PIC16F610/616** 电压－频率关系图， **-40°C ≤ TA ≤ +125°C**

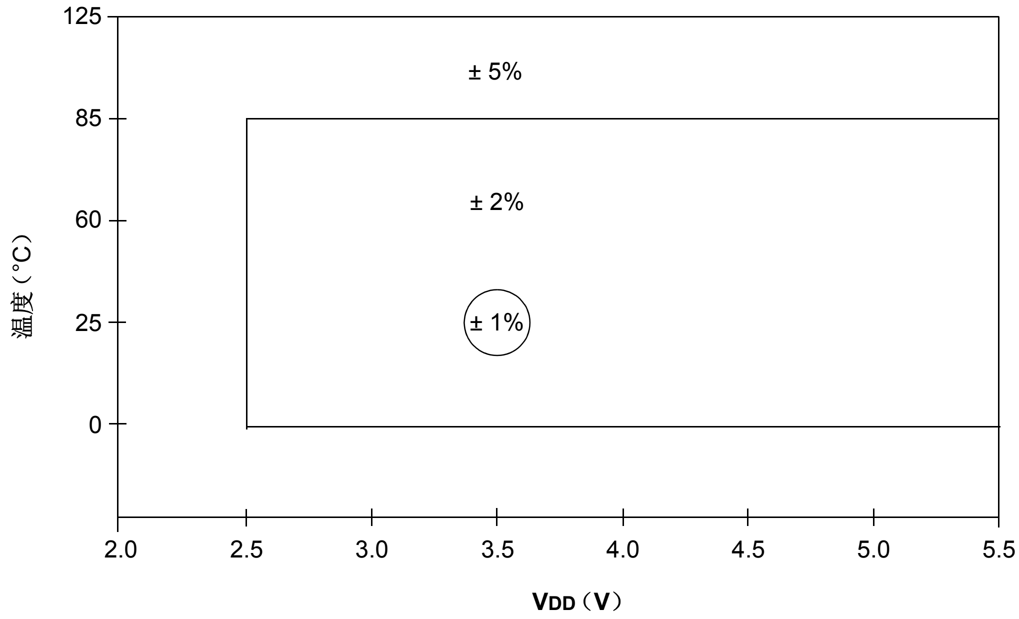
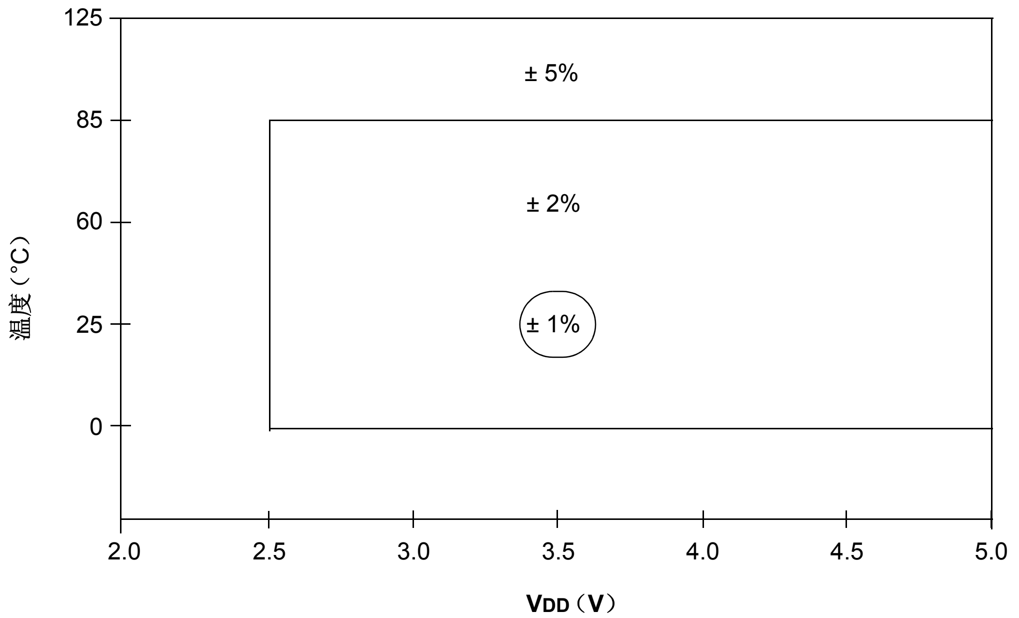


图 **15-4**： **PIC16HV610/616** 电压－频率关系图， **-40°C ≤ TA ≤ +125°C**



15.1 直流特性：PIC16F610/616/16HV610/616-I （工业级） PIC16F610/616/16HV610/616-E （扩展级）

直流特性			标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级） -40°C ≤ TA ≤ +125°C （扩展级）				
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
	VDD	供电电压					
D001		PIC16F610/616	2.0	—	5.5	V	FOSC < = 4 MHz
D001		PIC16HV610/616	2.0	—	5.0	V	FOSC < = 4 MHz
D001B		PIC16F610/616	2.0	—	5.5	V	FOSC < = 8 MHz
D001B		PIC16HV610/616	2.0	—	5.0	V	FOSC < = 8 MHz
D001C		PIC16F610/616	3.0	—	5.5	V	FOSC < = 10 MHz
D001C		PIC16HV610/616	3.0	—	5.0	V	FOSC < = 10 MHz
D001D		PIC16F610/616	4.5	—	5.5	V	FOSC < = 20 MHz
D001D		PIC16HV610/616	4.5	—	5.0	V	FOSC < = 20 MHz
D002*	VDR	**RAM 数据保持电压** (1)	1.5	—	—	V	器件处于休眠模式
D003	VPOR	**VDD 启动电压** （确保内部上电复位信号）	—	VSS	—	V	详情请参见**第 12.3.1 节 "上电复位（POR）"**。
D004*	SVDD	**VDD 上升率** （确保内部上电复位信号）	0.05	—	—	V/ms	详情请参见**第 12.3.1 节 "上电复位（POR）"**。

* 这些参数为特性值，未经测试。

† 除非另外声明，否则"典型值"栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 这是在不丢失 RAM 数据的前提下，休眠模式下 VDD 所能降到的最小电压值。

15.2 直流特性：PIC16F610/616/16HV610/616-I （工业级） PIC16F610/616/16HV610/616-E （扩展级）

直流特性		标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级） -40°C ≤ TA ≤ +125°C （扩展级）					
参数编号	器件特性	最小值	典型值 †	最大值	单位	条件	
						VDD	注
D010	供电电流 （IDD）(1, 2)	—	11	16	μA	2.0	Fosc = 32 kHz LP 振荡器模式
		—	18	28	μA	3.0	
		—	35	54	μA	5.0	
D011*		—	140	240	μA	2.0	FOSC = 1 MHz XT 振荡器模式
		—	220	380	μA	3.0	
		—	380	550	μA	5.0	
D012		—	260	360	μA	2.0	FOSC = 4 MHz XT 振荡器模式
		—	420	650	μA	3.0	
		—	0.8	1.1	mA	5.0	
D013*		—	130	220	μA	2.0	FOSC = 1 MHz EC 振荡器模式
		—	215	360	μA	3.0	
		—	360	520	μA	5.0	
D014		—	220	340	μA	2.0	FOSC = 4 MHz EC 振荡器模式
		—	375	550	μA	3.0	
		—	0.65	1.0	mA	5.0	
D016*		—	340	450	μA	2.0	FOSC = 4 MHz INTOSC 模式
		—	500	700	μA	3.0	
		—	0.8	1.2	mA	5.0	
D017		—	410	650	μA	2.0	FOSC = 8 MHz INTOSC 模式
		—	700	950	μA	3.0	
		—	1.30	1.65	mA	5.0	
D018		—	230	400	μA	2.0	FOSC = 4 MHz EXTRC 模式 (3)
		—	400	680	μA	3.0	
		—	0.63	1.1	mA	5.0	
D019		—	2.6	3.25	mA	4.5	FOSC = 20 MHz HS 振荡器模式
		—	2.8	3.35	mA	5.0	

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 有效工作模式下，所有 IDD 测量值的测试条件为：OSC1 = 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 ； WDT 禁止。

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、内部代码执行模式和温度等，也会对电流消耗产生影响。

3： 对于 RC 振荡器配置，该电流不包括流经 REXT 的电流。流经该电阻的电流可以由公式 IR = VDD/2REXT （mA）来估算，其中 REXT 的单位是 kΩ。

15.3 直流特性：PIC16F616/16HV616-I （工业级）

直流特性		标准工作条件 （除非另外声明） 工作温度 -40°C ≤ T_A ≤ +85°C （工业级）					
参数编号	器件特性	最小值	典型值 †	最大值	单位	条件 V_{DD}	条件 注
D020	掉电基本电流 （I_{PD}） (2)	—	0.05	1.2	µA	2.0	WDT、 BOR、比较器、 V_{REF} 和 T1OSC 均被禁止
	PIC16F610/616	—	0.15	1.5	µA	3.0	
		—	0.35	1.8	µA	5.0	
		—	150	500	nA	3.0	-40°C ≤ T_A ≤ +25°C
	PIC16HV610/616	—	350	—	µA	2.0	
		—	350	—	µA	3.0	
		4	—	50	mA	5.0	注 3
D021		—	1	TBD	µA	2.0	WDT 电流 (1)
		—	2	TBD	µA	3.0	
		—	8	TBD	µA	5.0	
D022		—	3	TBD	µA	3.0	BOR 电流 (1)
		—	4	TBD	µA	5.0	
D023		—	32	TBD	µA	2.0	比较器电流 (1)，两个比较器均被使能
		—	60	TBD	µA	3.0	
		—	120	TBD	µA	5.0	
D024		—	30	30	µA	2.0	CV_{REF} 电流 (1) （高电压范围）
		—	45	55	µA	3.0	
		—	75	95	µA	5.0	
D025*		—	39	47	µA	2.0	CV_{REF} 电流 (1) （低电压范围）
		—	59	72	µA	3.0	
		—	98	124	µA	5.0	
D026		—	45	7.0	µA	2.0	T1OSC 电流 (1)， 32.768 kHz
		—	5.0	8.0	µA	3.0	
		—	6.0	12	µA	5.0	
D027		—	0.30	1.6	µA	3.0	A/D 电流 (1)，无转换进行
		—	0.36	1.9	µA	5.0	

图注： TBD = 待定

* 这些参数为特性值，未经测试。

† 除非另外声明，否则“典型值”栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 外设电流为基本 I_{DD} 或 I_{PD} 与该外设使能时所额外消耗的电流之和。可通过从该极限值中减去基本 I_{DD} 或 I_{PD} 电流，以确定外设 Δ 电流。在计算总电流消耗时应使用最大值。

2： 在休眠模式下，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、所有 I/O 引脚处于高阻态并且连接到 V_{DD} 时测得的。

3： 并联稳压器始终使能并始终消耗工作电流。

15.4 直流特性：PIC16F610/616/16HV610/616-E （扩展级）

直流特性		标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C （扩展级）					
参数编号	器件特性	最小值	典型值 †	最大值	单位	条件 VDD	条件 注
D020E	掉电基本电流 （IPD）(2)	—	0.05	9	μA	2.0	WDT、 BOR、比较器、 VREF 和 T1OSC 均被禁止
		—	0.15	11	μA	3.0	
	PIC16F610/616	—	0.35	15	μA	5.0	
	PIC16HV610/616	—	350	—	μA	2.0	注 3
		—	350	—	μA	3.0	
		4	—	200	nA	5.0	
D021E		—	1	TBD	μA	2.0	WDT 电流 (1)
		—	2	TBD	μA	3.0	
		—	8	TBD	μA	5.0	
D022E		—	3	TBD	μA	3.0	BOR 电流 (1)
		—	4	TBD	μA	5.0	
D023E		—	32	TBD	μA	2.0	比较器电流 (1)，两个比较器均被使能
		—	60	TBD	μA	3.0	
		—	120	TBD	μA	5.0	
D024E		—	30	70	μA	2.0	CVREF 电流 (1) （高电压范围）
		—	45	90	μA	3.0	
		—	75	120	μA	5.0	
D025E*		—	39	91	μA	2.0	CVREF 电流 (1) （低电压范围）
		—	59	117	μA	3.0	
		—	98	156	μA	5.0	
D026E		—	4.5	25	μA	2.0	T1OSC 电流 (1)， 32.768 kHz
		—	5	30	μA	3.0	
		—	6	40	μA	5.0	
D027E		—	0.30	12	μA	3.0	A/D 电流 (1)，无转换进行
		—	0.36	16	μA	5.0	

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 外设电流为基本 IDD 或 IPD 与该外设使能时所额外消耗的电流之和。可通过从该极限值中减去基本 IDD 或 IPD 电流，以确定外设 Δ 电流。在计算总电流消耗时应使用最大值。

2： 在休眠模式下，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、所有 I/O 引脚处于高阻态并且连接到 VDD 时测得的。

3： 并联稳压器始终使能并始终消耗工作电流。

15.5 直流特性： PIC16F610/616/16HV610/616-I （工业级） PIC16F610/616/16HV610/616-E （扩展级）

直流特性			标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级） -40°C ≤ TA ≤ +125°C （扩展级）				
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
	VIL	输入低电压					
		I/O 端口：					
D030		带 TTL 缓冲器	Vss	—	0.8	V	4.5V ≤ VDD ≤ 5.5V
D030A			Vss	—	0.15 VDD	V	2.0V ≤ VDD ≤ 4.5V
D031		带施密特触发缓冲器	Vss	—	0.2 VDD	V	2.0V ≤ VDD ≤ 5.5V
D032		MCLR 和 OSC1 （RC 模式）(1)	VSS	—	0.2 VDD	V	
D033		OSC1 （XT 和 LP 模式）	VSS	—	0.3	V	
D033A		OSC1 （HS 模式）	VSS	—	0.3 VDD	V	
	VIH	输入高电压					
		I/O 端口：					
D040		带 TTL 缓冲器	2.0	—	VDD	V	4.5V ≤ VDD ≤ 5.5V
D040A			0.25 VDD + 0.8	—	VDD	V	2.0V ≤ VDD ≤ 4.5V
D041		带施密特触发缓冲器	0.8 VDD	—	VDD	V	2.0V ≤ VDD ≤ 5.5V
D042		MCLR	0.8 VDD	—	VDD	V	
D043		OSC1 （XT 和 LP 模式）	1.6	—	VDD	V	
D043A		OSC1 （HS 模式）	0.7 VDD	—	VDD	V	
D043B		OSC1 （RC 模式）	0.9 VDD	—	VDD	V	（注 1）
	IIL	输入泄漏电流 (2)					
D060		I/O 端口	—	± 0.1	± 1	μA	VSS ≤ VPIN ≤ VDD，引脚处于高阻态
D061		MCLR(3)	—	± 0.1	± 5	μA	VSS ≤ VPIN ≤ VDD
D063		OSC1	—	± 0.1	± 5	μA	VSS ≤ VPIN ≤ VDD，XT、HS 和 LP 振荡器配置
D070*	IPUR	PORTA 弱上拉电流	50	250	400	μA	VDD = 5.0V， VPIN = VSS
	VOL	输出低电压 (4)					
D080		I/O 端口	—	—	0.6	V	IOL = 8.5 mA， VDD = 4.5V （工业级）
	VOH	输出高电压 (4)					
D090		I/O 端口	VDD – 0.7	—	—	V	IOH = -3.0 mA，VDD = 4.5V （工业级）

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值”栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 在 RC 振荡器配置中， OSC1/CLKIN 引脚被配置为施密特触发器输入。在 RC 模式下，建议不要使用外部时钟。

2： 负电流定义为引脚的拉电流。

3： MCLR 引脚上的泄漏电流主要取决于所施加电压。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测得更高的泄漏电流。

4： 在 CLKOUT 模式下包括 OSC2。

 初稿

15.5 直流特性： PIC16F610/616/16HV610/616-I （工业级） PIC16F610/616/16HV610/616-E （扩展级） （续）

直流特性			标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级） -40°C ≤ TA ≤ +125°C （扩展级）				
参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
		输出引脚上的容性负载规范					
D101*	COSC2	OSC2 引脚	—	—	15	pF	当外部时钟用于驱动 OSC1 时处于 XT、 HS 和 LP 模式下
D101A*	CIO	所有 I/O 引脚	—	—	50	pF	
		程序闪存					
D130	EP	单元耐擦写能力	10K	100K	—	E/W	-40°C ≤ TA ≤ +85°C
D130A	ED	单元耐擦写能力	1K	10K	—	E/W	+85°C ≤ TA ≤ +125°C
D131	VPR	用于读取的 VDD	VMIN	—	5.5	V	VMIN = 最小工作电压
D132	VPEW	用于擦除 / 写的 VDD	4.5	—	5.5	V	
D133	TPEW	擦除 / 写周期时间	—	2	2.5	ms	
D134	TRETD	特性保持时间	40	—	—	年	假设没有违反其他规范

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值”栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 **1**： 在 RC 振荡器配置中， OSC1/CLKIN 引脚被配置为施密特触发器输入。在 RC 模式下，建议不要使用外部时钟。

2： 负电流定义为引脚的拉电流。

3： $\overline{\text{MCLR}}$ 引脚上的泄漏电流主要取决于所施加电压。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测得更高的泄漏电流。

4： 在 CLKOUT 模式下包括 OSC2。

15.6 热阻注意事项

标准工作条件（除非另外声明） 工作温度 -40°C ≤ T_A ≤ +125°C					
参数编号	符号	特性	典型值	单位	条件
TH01	θ_{JA}	热阻 （结到环境）	70	C/W	14 引脚 PDIP 封装
			85.0	C/W	14 引脚 SOIC 封装
			100	C/W	14 引脚 TSSOP 封装
			46.3	C/W	16 引脚 QFN 4x4mm 封装
TH02	θ_{JC}	热阻 （结到管壳）	32.5	C/W	14 引脚 PDIP 封装
			31.0	C/W	14 引脚 SOIC 封装
			31.7	C/W	14 引脚 TSSOP 封装
			2.6	C/W	16 引脚 QFN 4x4mm 封装
TH03	T_{DIE}	极限温度	150	C	
TH04	PD	功耗	—	W	PD = $P_{INTERNAL}$ + $P_{I/O}$
TH05	$P_{INTERNAL}$	内部功耗	—	W	$P_{INTERNAL}$ = I_{DD} x V_{DD} （注 **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}	降功耗	—	W	$P_{DER} = PD_{MAX} (T_{DIE} - T_A)/\theta_{JA}$ （注 **2**）

注 **1：** I_{DD} 是独立运行芯片而无需驱动输出引脚上任何负载时的电流。

2： T_A = 环境温度。

15.7 时序参数符号体系

时序参数符号采用以下格式之一进行创建：

1. TppS2ppS
2. TppS

T			
F	频率	T	时间

小写字母（pp）及其含义：

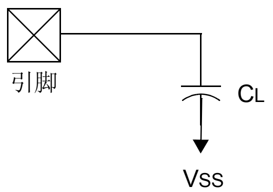
pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{\text{RD}}$
cs	$\overline{\text{CS}}$	rw	$\overline{\text{RD}}$ 或 $\overline{\text{WR}}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{\text{SS}}$
dt	数据输入	t0	T0CKI
io	I/O 端口	t1	T1CKI
mc	$\overline{\text{MCLR}}$	wr	$\overline{\text{WR}}$

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）	V	有效
L	低	Z	高阻

图 15-5： 负载条件

负载条件



图注： C_L = 50 pF （对于所有引脚）
15 pF （对于 OSC2 输出）

15.8 交流特性：PIC16F610/616/16HV610/616 （工业级，扩展级）

图 15-6： 时钟时序

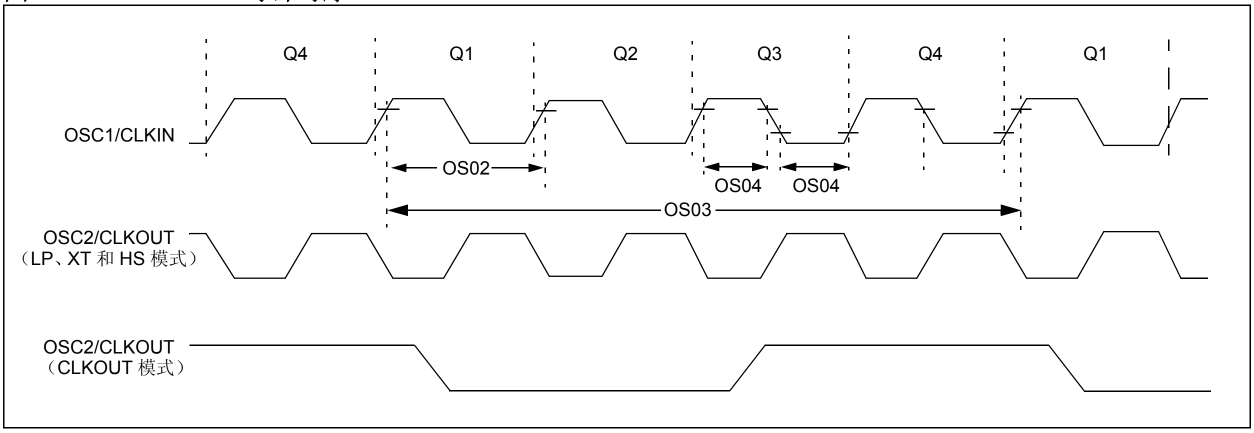


表 15-1： 时钟振荡器时序要求

标准工作条件（除非另外声明）
工作温度 -40°C ≤ T_A ≤ +125°C

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	—	37	kHz	LP 振荡器模式
			DC	—	4	MHz	XT 振荡器模式
			DC	—	20	MHz	HS 振荡器模式
			DC	—	20	MHz	EC 振荡器模式
		振荡器频率 (1)	—	32.768	—	kHz	LP 振荡器模式
			0.1	—	4	MHz	XT 振荡器模式
			1	—	20	MHz	HS 振荡器模式
			DC	—	4	MHz	RC 振荡器模式
OS02	TOSC	外部 CLKIN 周期 (1)	27	—	∞	µs	LP 振荡器模式
			250	—	∞	ns	XT 振荡器模式
			50	—	∞	ns	HS 振荡器模式
			50	—	∞	ns	EC 振荡器模式
		振荡器周期 (1)	—	30.5	—	µs	LP 振荡器模式
			250	—	10,000	ns	XT 振荡器模式
			50	—	1,000	ns	HS 振荡器模式
			250	—	—	ns	RC 振荡器模式
OS03	TCY	指令周期时间 (1)	200	TCY	DC	ns	TCY = 4/FOSC
OS04*	TOSH，TOSL	外部 CLKIN 高电平时间，外部 CLKIN 低电平时间	2	—	—	µs	LP 振荡器
			100	—	—	ns	XT 振荡器
			20	—	—	ns	HS 振荡器
OS05*	TOSR，TOSF	外部 CLKIN 上升时间，外部 CLKIN 下降时间	0	—	∞	ns	LP 振荡器
			0	—	∞	ns	XT 振荡器
			0	—	∞	ns	HS 振荡器

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值”栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 指令周期时间 （TCY）等于输入振荡器时基周期的四倍。所有规定值均为基于针对特定振荡器类型，器件在标准工作条件下执行代码时的特性数据。超出这些规定的极限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 OSC1 引脚连接了外部时钟。当使用了外部时钟输入时，所有器件的 “最大”周期时间限制为 “DC”（无时钟）。

 初稿

表 15-2： 振荡器参数

标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C								
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
OS06	TWARM	运行时的内部振荡器开关 (3)	—	—	—	2	TOSC	最慢时钟
OS08	INTOSC	内部已校准 INTOSC 频率 (2)	±1%	7.92	8.0	8.08	MHz	VDD = 3.5V， 25°C
			±2%	7.84	8.0	8.16	MHz	2.5V ≤ VDD ≤ 5.5V， 0°C ≤ TA ≤ +85°C
			±5%	7.60	8.0	8.40	MHz	2.0V ≤ VDD ≤ 5.5V， -40°C ≤ TA ≤ +85°C （工业级）， -40°C ≤ TA ≤ +125°C （扩展级）
OS10*	TIOSC ST	INTOSC 振荡器从休眠模式唤醒的起振时间	—	5.5	12	24	μs	VDD = 2.0V， -40°C 至 +85°C
			—	3.5	7	14	μs	VDD = 3.0V， -40°C 至 +85°C
			—	3	6	11	μs	VDD = 5.0V， -40°C 至 +85°C

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值”栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 指令周期时间 （TCY）等于输入振荡器时基周期的四倍。所有规定值均为基于针对特定振荡器类型，器件在标准工作条件下执行代码时的特性数据。超出这些规定的极限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 OSC1 引脚连接了外部时钟。当使用了外部时钟输入时，所有器件的 “最大”周期时间限制为 “DC”（无时钟）。

2： 为了确保振荡器频率容差，必须尽可能靠近器件，在 VDD 和 VSS 之间接去耦电容。建议并联 0.1 μF 和 0.01 μF 的电容。

3： 根据设计。

图 15-7： CLKOUT 和 I/O 时序

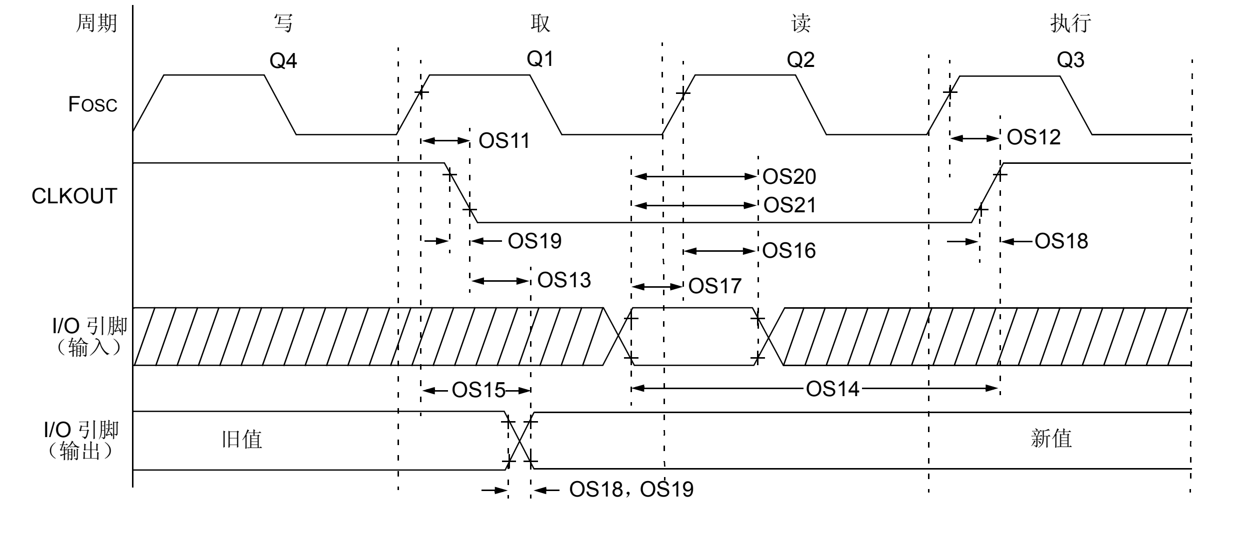


表 15-3： CLKOUT 和 I/O 时序参数

标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
OS11	TOSH2CKL	FOSC↑ 到 CLKOUT↓ 的时间 (1)	—	—	70	ns	VDD = 5.0V
OS12	TOSH2CKH	FOSC↑ 到 CLKOUT↑ 的时间 (1)	—	—	72	ns	VDD = 5.0V
OS13	TCKL2IOV	CLKOUT↓ 到端口输出有效的时间 (1)	—	—	20	ns	
OS14	TIOV2CKH	在 CLKOUT↑ 之前端口输入有效的时间 (1)	TOSC + 200 ns	—	—	ns	
OS15	TOSH2IOV	FOSC↑（Q1 周期）到端口输出有效的时间	—	50	70*	ns	VDD = 5.0V
OS16	TOSH2IOI	FOSC↑（Q2 周期）到端口输入无效的时间（I/O 输入保持时间）	50	—	—	ns	VDD = 5.0V
OS17	TIOV2OSH	端口输入有效到 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 (2)	— —	15 40	72 32	ns	VDD = 2.0V VDD = 5.0V
OS19	TIOF	端口输出下降时间 (2)	— —	28 15	55 30	ns	VDD = 2.0V VDD = 5.0V
OS20*	TINP	INT 引脚输入高电平或低电平时间	25	—	—	ns	
OS21*	TRAP	PORTA 电平变化中断新输入电平时间	TCY	—	—	ns	

* 这些参数为特性值，未经测试。

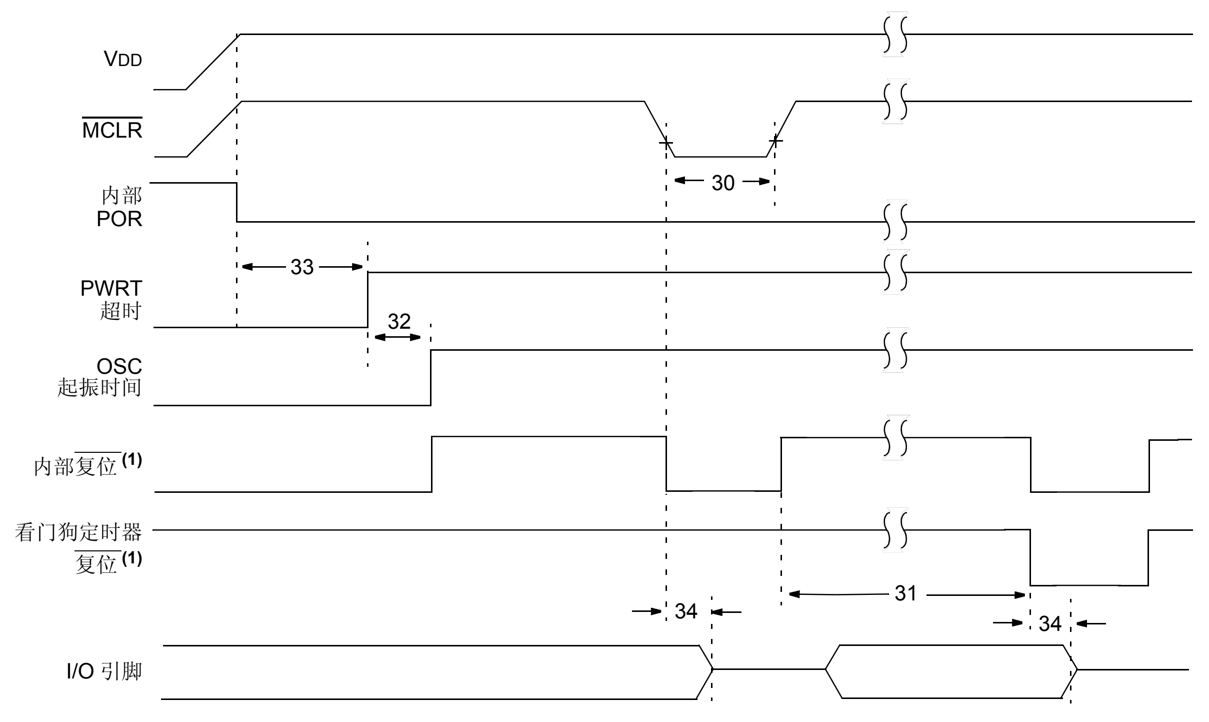
† 除非另外声明，否则 “典型值” 栏中的数据均为 5.0V 和 25°C 条件下的值。

注 1： 测量是在 RC 模式下进行的，其中 CLKOUT 输出为 4 x TOSC。

2： 在 CLKOUT 模式下包括 OSC2。

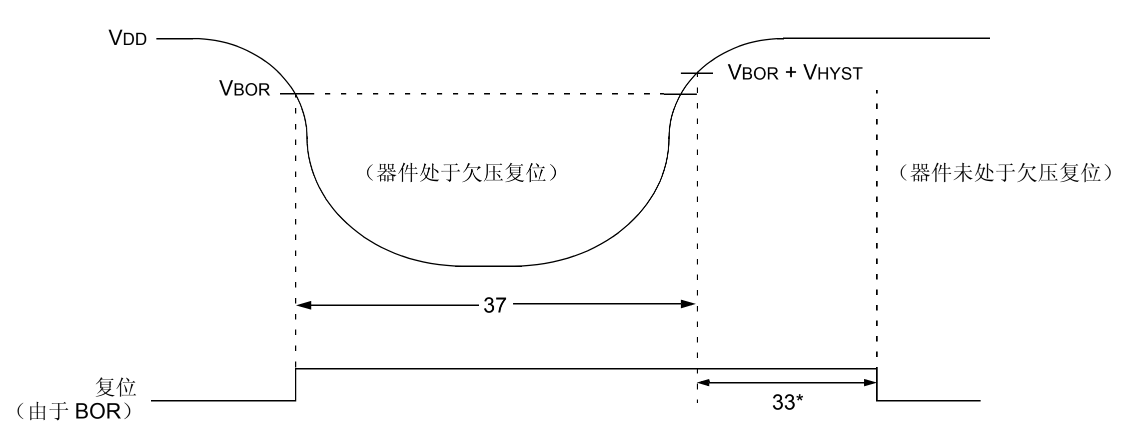
 初稿

图 15-8：　　　　复位、看门狗定时器、振荡器起振定时器和上电延时定时器时序



注　1：　拉低。

图 15-9：　　　　欠压复位时序和特性



* 仅在配置字寄存器中 PWRTE 位编程为 0 时才有 64 ms 延时。

表 15-4： 复位、看门狗定时器、振荡器起振定时器、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）
工作温度 -40°C ≤ TA ≤ +125°C

参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
30	TMCL	$\overline{\text{MCLR}}$ 脉冲宽度 （低电平）	2 5	— —	— —	μs μs	VDD = 5V， -40°C 至 +85°C VDD = 5V， +85°C 至 +125°C
31	TWDT	看门狗定时器超时周期 （无预分频器）	7 TBD	18 18	33 TBD	ms ms	VDD = 5V， -40°C 至 +85°C VDD = 5V， +85°C 至 +125°C
32	TOST	振荡器起振定时器周期 (1, 2)	—	1024	—	TOSC	（注 3）
33*	TPWRT	上电延时定时器周期	40	65	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	TBD	2.1	TBD	V	（注 4）
36*	VHYST	欠压复位滞后	—	50	—	mV	
37*	TBOR	欠压复位最小检测周期	100	—	—	μs	VDD ≤ VBOR

图注： TBD = 待定

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 指令周期时间 （TCY）等于输入振荡器时基周期的四倍。所有规定值均为基于针对特定振荡器类型，器件在标准工作条件下执行代码时的特性数据。超出这些规定的极限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 OSC1 引脚连接了外部时钟。当使用了外部时钟输入时，所有器件的 “最大” 周期时间限制为 “DC”（无时钟）。

2： 根据设计。

3： 较慢时钟的周期。

4： 为了确保这些电压容差，必须尽可能靠近器件，在 VDD 和 VSS 之间接去耦电容。建议并联 0.1 μF 和 0.01 μF 的电容。

图 15-10： TIMER0 和 TIMER1 外部时钟时序

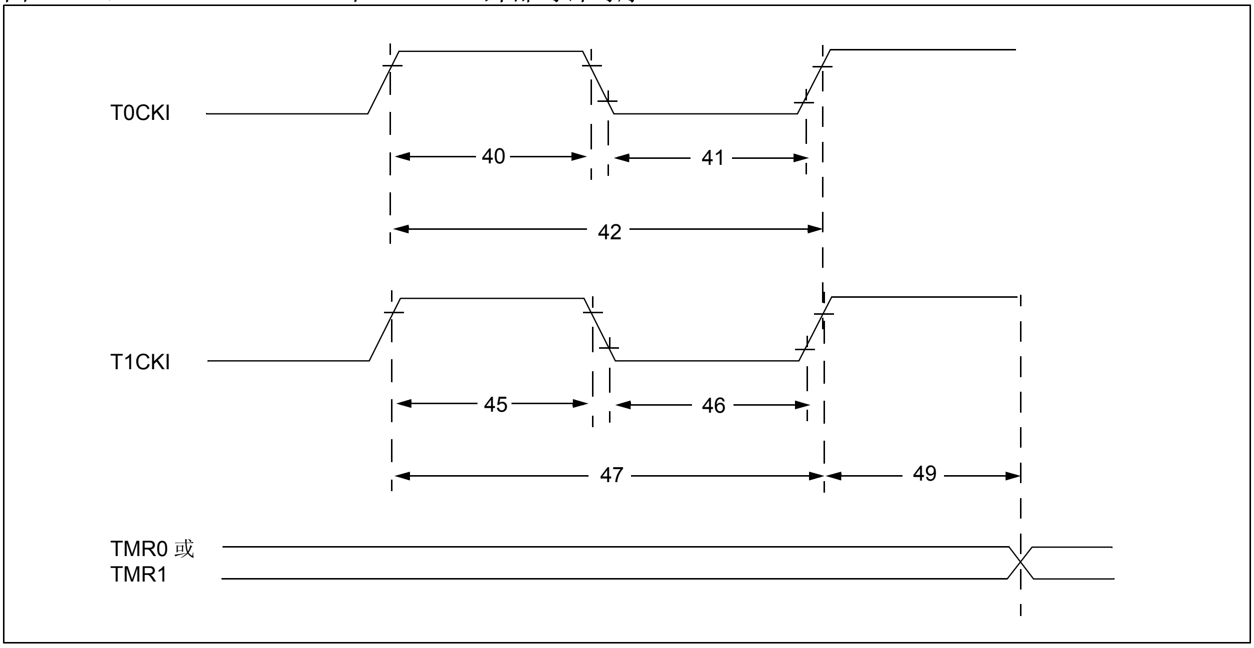


表 15-5： TIMER0 和 TIMER1 外部时钟要求

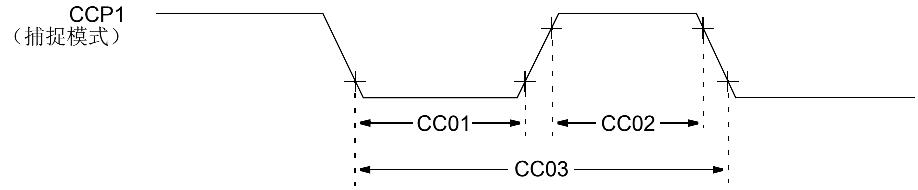
标准工作条件 （除非另外声明）
工作温度 -40°C ≤ TA ≤ +125°C

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		取如下二者中较大值：20 或 $\frac{TCY + 40}{N}$	—	—	ns	N = 预分频值 （2, 4, ..., 256）
45*	TT1H	T1CKI 高电平时间	同步，无预分频器	0.5 TCY + 20	—	—	ns	
			同步，有预分频器	15	—	—	ns	
			异步	30	—	—	ns	
46*	TT1L	T1CKI 低电平时间	同步，无预分频器	0.5 TCY + 20	—	—	ns	
			同步，有预分频器	15	—	—	ns	
			异步	30	—	—	ns	
47*	TT1P	T1CKI 输入周期	同步	取如下二者中较大值：30 或 $\frac{TCY + 40}{N}$	—	—	ns	N = 预分频值 （1, 2, 4, 8）
			异步	60	—	—	ns	
48	FT1	Timer1 振荡器输入频率范围（通过将 T1OSCEN 位置 1，使能振荡器）		—	32.768	—	kHz	
49*	TCKEZTMR1	从外部时钟边沿到定时器递增的延时		2 TOSC	—	7 TOSC	—	同步模式下的定时器

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

图 15-11： 捕捉 / 比较 /PWM 时序 （ECCP）



注： 负载条件请参见图 15-5。

表 15-6： 捕捉 / 比较 /PWM 要求 （ECCP）

标准工作条件 （除非另外声明）
工作温度 -40°C ≤ T_A ≤ +125°C

参数编号	符号	特性		最小值	典型值 †	最大值	单位	条件
CC01*	TccL	CCP1 输入低电平时间	无预分频器	0.5TCY + 20	—	—	ns	
			有预分频器	20	—	—	ns	
CC02*	TccH	CCP1 输入高电平时间	无预分频器	0.5TCY + 20	—	—	ns	
			有预分频器	20	—	—	ns	
CC03*	TccP	CCP1 输入周期		$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = 预分频值（1、 4 或 16）

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值”栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

表 15-7： 比较器规范

标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C								
参数编号	符号	特性		最小值	典型值 †	最大值	单位	备注
CM01	VOS	输入失调电压		—	± 5.0	± 10	mV	(VDD - 1.5)/2
CM02	VCM	输入共模电压		0	—	VDD – 1.5	V	
CM03*	CMRR	共模抑制比		+55	—	—	dB	
CM04*	TRT	响应时间	下降	—	150	600	ns	（注 1）
			上升	—	200	1000	ns	
CM05*	TMC2COV	比较器模式变为输出有效的时间		—	—	10	µs	
CM06*	VHYS	输入滞后电压		—	45	—	mV	

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 响应时间是在比较器的一个输入端电压从 (VDD - 1.5)/2 - 100 mV 增加到 (VDD - 1.5)/2 + 20 mV 时测得的。

表 15-8： 比较器参考电压（CVREF）规范

标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	备注
CV01	CLSB	步长 (2)	— —	VDD/24 VDD/32	— —	V V	低电平范围（VRR = 1） 高电平范围（VRR = 0）
CV02	CACC	绝对精度	— —	— —	± 1/2 ± 1/2	LSb LSb	低电平范围（VRR = 1） 高电平范围（VRR = 0）
CV03	CR	单位电阻值（R）	—	2k	—	Ω	
CV04	CST	稳定时间 (1)	—	—	10	µs	

† 除非另外声明，否则“典型值”栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 稳定时间是在 VRR = 1 并且 VR<3:0> 从 0000 跳变到 1111 时测得的。

2： 更多信息，请参见第 **8.11** 节 **“比较器参考电压”**。

表 15-9： 参考电压规范

VR 参考电压规范			标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C				
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
VR01	VP6OUT	VP6 电压输出	0.55	0.6	0.65	V	
VR02	V1P2OUT	V1P2 电压输出	1.1	1.200	1.3	V	
VR03	TSTABLE	稳定时间	—	10	—	µs	

* 这些参数为特性值，未经测试。

表 15-10： 并联稳压器规范 （仅限 PIC16HV610/616）

并联稳压器特性			标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C				
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
SR01	VSHUNT	并联电压	4.75	5	5.25	V	
SR02	ISHUNT	并联电流	4	—	50	mA	
SR03*	TSETTLE	稳定时间	—	—	150	ns	至最终值的 1%
SR04	CLOAD	负载电容	0.01	—	10	μF	VDD 引脚上的旁路电容
SR05	ΔISNT	稳压器工作电流	—	—	180	μA	包括带隙参考电流

* 这些参数为特性值，未经测试。

表 15-11： PIC16F616/16HV616 A/D 转换器 （ADC）特性：

标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C							
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	—	—	10 位	位	
AD02	EIL	积分线性误差	—	—	±1	LSb	VREF = 5.12V
AD03	EDL	微分线性误差	—	—	±1	LSb	10 位不会丢失编码 VREF = 5.12V
AD04	EOFF	失调误差	—	1.5		LSb	VREF = 5.12V
AD07	EGN	增益误差	—	—	±1	LSb	VREF = 5.12V
AD06 AD06A	VREF	参考电压 (3)	2.2 2.5	—	— VDD	V	绝对确保最小 1 LSb 精度
AD07	VAIN	满量程	VSS	—	VREF	V	
AD08	ZAIN	模拟信号源阻抗的推荐值	—	—	10	kΩ	
AD09*	IREF	VREF 输入电流 (3)	10	—	1000	μA	在采集 VAIN 期间。 基于 VHOLD 对 VAIN 之差。
			—	—	50	μA	在 A/D 转换周期期间。

* 这些参数为特性值，未经测试。

† 除非另外声明，否则“典型值”栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 总的绝对误差包括积分、微分、失调和增益误差。

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 来自选择作为参考输入的外部 VREF 或 VDD 引脚。

4： 当 ADC 关断时，它除了消耗很少的泄漏电流外，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。

表 15-12： PIC16F616/16HV616 A/D 转换要求

参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C							
AD130*	TAD	A/D 时钟周期	1.6	—	9.0	μs	基于 TOSC， VREF ≥ 3.0V
			3.0	—	9.0	μs	基于 TOSC， VREF 满量程
		A/D 内部 RC 振荡器周期					ADCS<1:0> = 11 （ADRC 模式）
			3.0	6.0	9.0	μs	VDD = 2.5V
			1.6	4.0	6.0	μs	VDD = 5.0V
AD131	TCNV	转换时间 （不包括采集时间）(1)	—	11	—	TAD	将 GO/$\overline{\text{DONE}}$ 位设定为 A/D 结果寄存器中的新数据
AD132*	TACQ	采集时间		11.5	—	μs	
AD133*	TAMP	放大器稳定时间	—	—	5	μs	
AD134	TGO	Q4 到 A/D 时钟启动	—	TOSC/2	—	—	
			—	TOSC/2 + TCY	—	—	如果将 A/D 时钟源选择为 RC，在 A/D 时钟启动前要加上一个 TCY 时间，用以执行 SLEEP 指令。

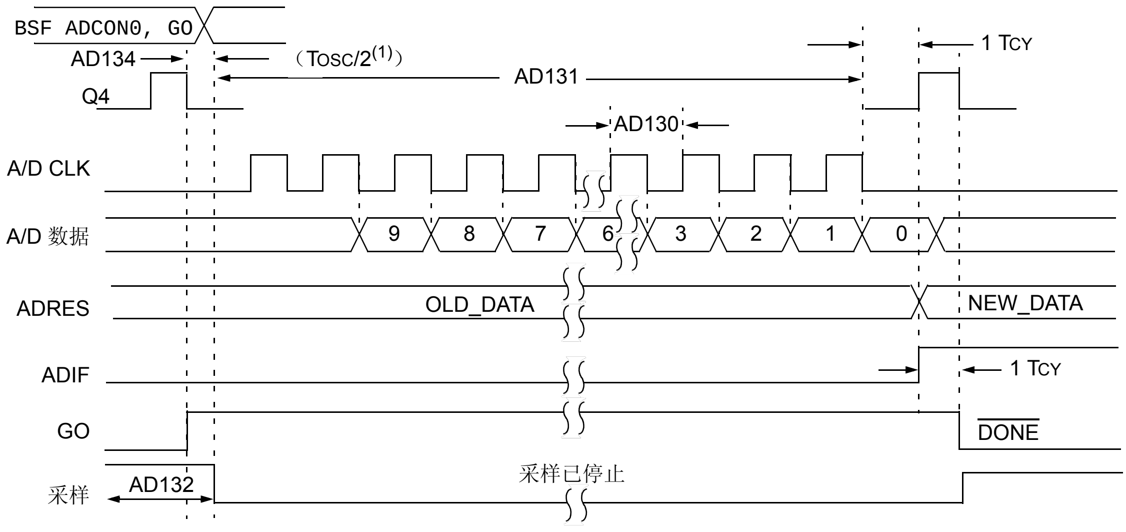
* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 **1：** 可在后续 TCY 周期内读 ADRESH 和 ADRESL 寄存器。

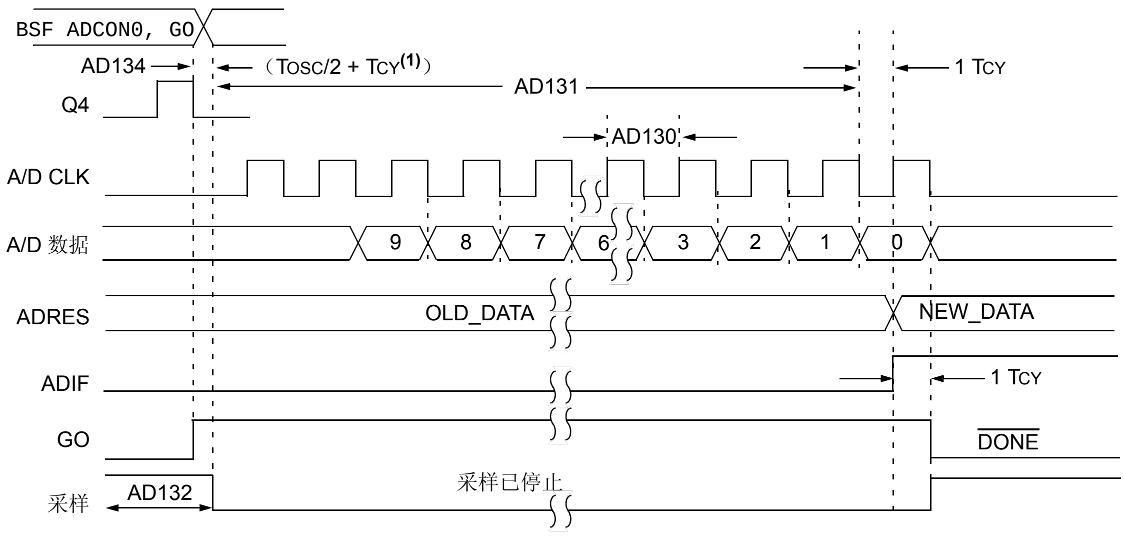
2： 最小条件请参见**第 9.3 节 “A/D 采集要求”**。

图 15-12： PIC16F616/16HV616 A/D 转换时序 （正常模式）



注 1： 如果将 A/D 时钟源选择为 RC，在 A/D 时钟启动前要加上一个 TCY 时间，用以执行 SLEEP 指令。

图 15-13： PIC16F616/16HV616 A/D 转换时序 （休眠模式）



注 1： 如果将 A/D 时钟源选择为 RC，在 A/D 时钟启动前要加上一个 TCY 时间，用以执行 SLEEP 指令。

16.0 直流和交流特性图表

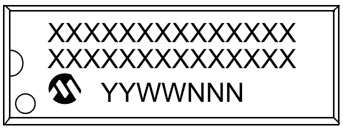
当前没有可用图表。

注：

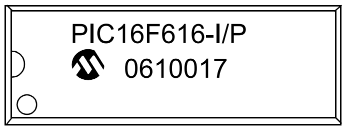
17.0 封装信息

17.1 封装标识信息

14 引脚 PDIP



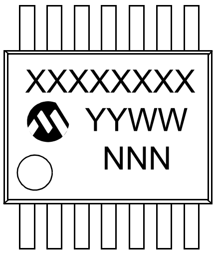
示例



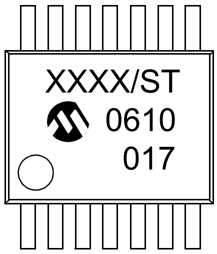
14 引脚 SOIC（0.150 英寸）

示例

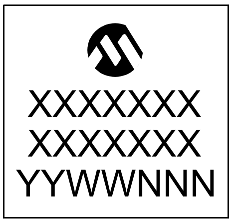
14 引脚 TSSOP



示例



16 引脚 QFN



示例

图注：		
	XX...X	客户信息
	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 （1 月 1 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 元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。

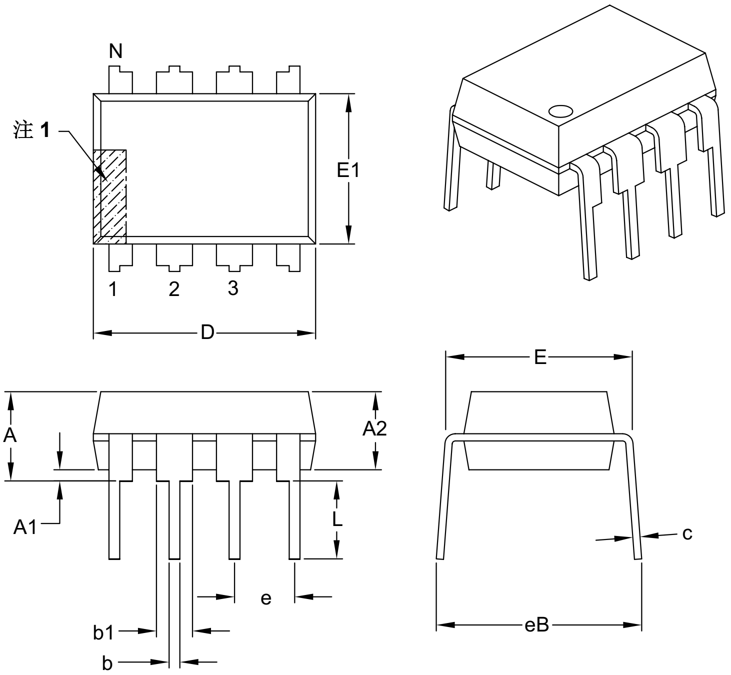
* 标准 PIC® 器件标识由 Microchip 元器件编号、年份代码、星期代码和追踪代码组成。若 PIC® 器件标识超出上述内容，需支付一定的附加费用。请向当地的 Microchip 销售办事处了解确认。对于 QTP 器件，任何特殊标记的费用都已包含在 QTP 价格中。

17.2 封装详细信息

以下部分将介绍各种封装的技术细节。

8 引脚塑封双列直插式 （P）——300 mil 主体 [PDIP]

注：	最新封装图请至 http://www.microchip.com/packaging 查看 Microchip 封装规范。



单位		英寸		
尺寸范围		最小	正常	最大
引脚数	N	8		
引脚间距	e	.100 BSC		
顶端到固定面高度	A	–	–	.210
塑模封装厚度	A2	.115	.130	.195
塑模底面到固定面高度	A1	.015	–	–
肩到肩宽度	E	.290	.310	.325
塑模封装宽度	E1	.240	.250	.280
总长度	D	.348	.365	.400
引脚尖到固定面高度	L	.115	.130	.150
引脚厚度	c	.008	.010	.015
引脚上部宽度	b1	.040	.060	.070
引脚下部宽度	b	.014	.018	.022
总排列间距 §	eB	–	–	.430

注：

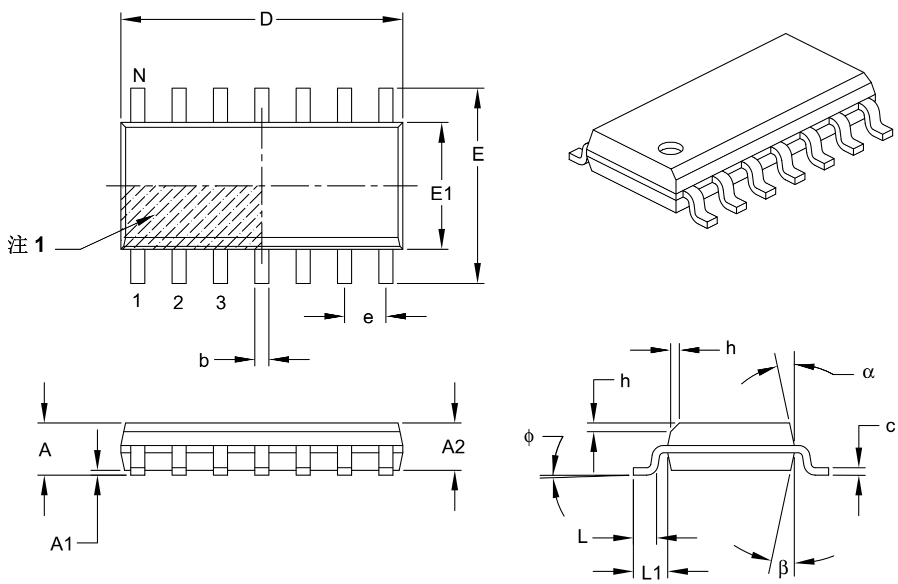
1. 引脚 1 的可见定位功能可能不同，但必须在阴影区域内。
2. § 重要特性。
3. 尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.010 英寸。
4. 尺寸和公差请参见 ASME Y14.5M。

BSC：基本尺寸。显示的是没有公差的理论精确值。

Microchip Technology 图号 C04-018B

14 引脚塑封小外形封装 （SL）——窄条， 3.90 mm 主体 [SOIC]

注：	最新封装图请至 http://www.microchip.com/packaging 查看 Microchip 封装规范。



单位		毫米		
尺寸范围		最小	正常	最大
引脚数	N	14		
引脚间距	e	1.27 BSC		
总高度	A	–	–	1.75
塑模封装厚度	A2	1.25	–	–
悬空间隙 §	A1	0.10	–	0.25
总宽度	E	6.00 BSC		
塑模封装宽度	E1	3.90 BSC		
总长度	D	8.65 BSC		
斜面 （可选）	h	0.25	–	0.50
底脚长度	L	0.40	–	1.27
引脚投影长度	L1	1.04 REF		
底脚倾斜角	φ	0	–	8
引脚厚度	c	0.17	–	0.25
引脚宽度	b	0.31	–	0.51
塑模顶部锥度	α	5	–	15
塑模底部锥度	β	5	–	15

注：

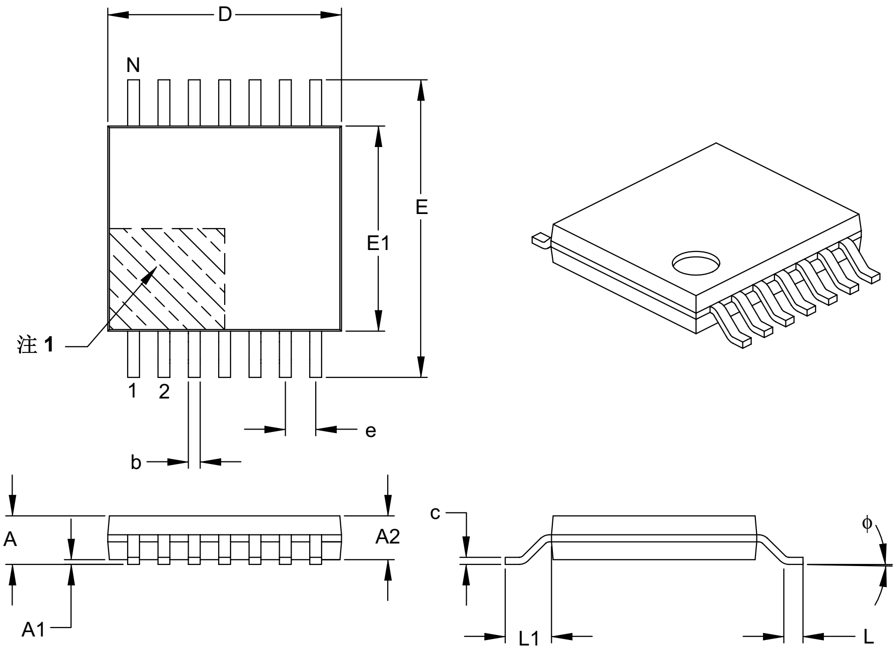
1. 引脚 1 的可见定位功能可能不同，但必须在阴影区域内。
2. § 重要特性。
3. 尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.15 毫米。
4. 尺寸和公差请参见 ASME Y14.5M。

   BSC：基本尺寸。显示的是没有公差的理论精确值。
   REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology 图号 C04-065B

14 引脚塑封超薄缩小外形封装 （ST）——4.4 mm 主体 [TSSOP]

注：	最新封装图请至 http://www.microchip.com/packaging 查看 Microchip 封装规范。



		单位	毫米	
	尺寸范围	最小	正常	最大
引脚数	N		14	
引脚间距	e		0.65 BSC	
总高度	A	–	–	1.20
塑模封装厚度	A2	0.80	1.00	1.05
悬空间隙	A1	0.05	–	0.15
总宽度	E		6.40 BSC	
塑模封装宽度	E1	4.30	4.40	4.50
塑模封装长度	D	4.90	5.00	5.10
底脚长度	L	0.45	0.60	0.75
引脚投影长度	L1		1.00 REF	
底脚倾斜角	φ	0	–	8
引脚厚度	c	0.09	–	0.20
引脚宽度	b	0.19	–	0.30

注：

1. 引脚 1 的可见定位功能可能不同，但必须在阴影区域内。
2. 尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.15 毫米。
3. 尺寸和公差请参见 ASME Y14.5M。

   BSC：基本尺寸。显示的是没有公差的理论精确值。

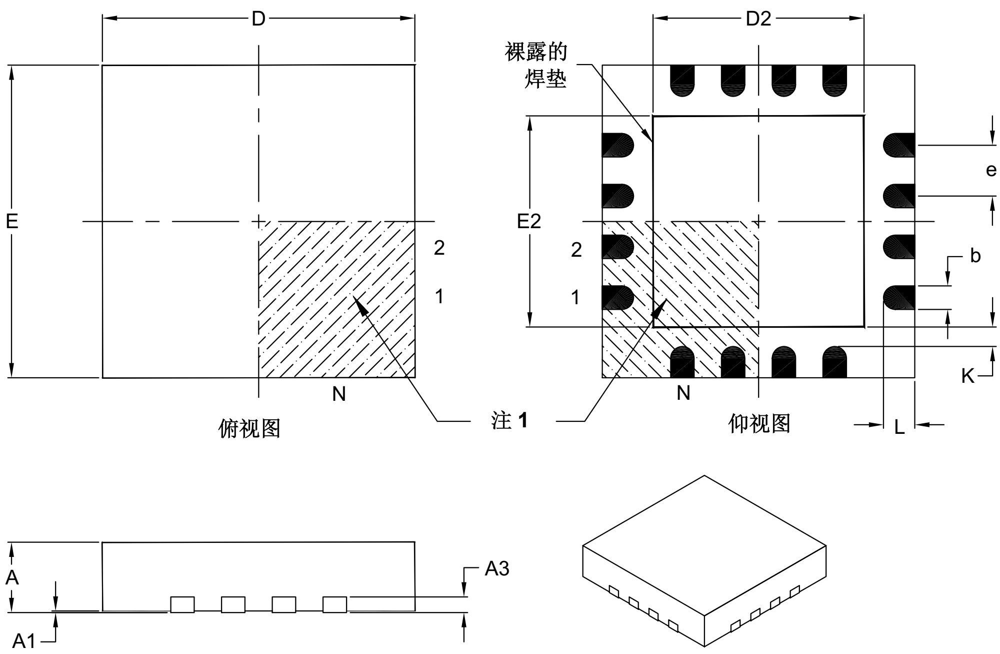
   REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology 图号 C04-087B

 初稿

16 引脚塑封正方扁平无脚封装 （ML）——4x4x0.9 mm 主体 [QFN]

注： 最新封装图请至 http://www.microchip.com/packaging 查看 Microchip 封装规范。



单位		毫米		
尺寸范围		最小	正常	最大
引脚数	N		16	
引脚间距	e		0.65 BSC	
总高度	A	0.80	0.90	1.00
悬空间隙	A1	0.00	0.02	0.05
触点厚度	A3		0.20 REF	
总宽度	E		4.00 BSC	
裸露金属焊垫宽度	E2	2.50	2.65	2.80
总长度	D		4.00 BSC	
裸露金属焊垫长度	D2	2.50	2.65	2.80
触点宽度	b	0.25	0.30	0.35
触点长度	L	0.30	0.40	0.50
触点到裸露金属焊垫的距离	K	0.20	–	–

注：

1. 引脚 1 的可见定位功能可能不同，但必须在阴影区域内。
2. 封装为切割分离。
3. 尺寸和公差请参见 ASME Y14.5M。

   BSC：基本尺寸。显示的是没有公差的理论精确值。
   REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology 图号 C04-127B

注：

附录 A： 数据手册版本历史

版本 A

这是新的数据手册。

版本 B （12/06）

添加了 PIC16F610/16HV610 器件。
更换了封装图。

附录 B： 从其他 PIC® 器件移植

本节将讨论从其他 PIC® 器件移植到 PIC16F6XX 系列器件的一些问题。

B.1 从 PIC16F676 移植到 PIC16F610/616/16HV610/616

表 B-1： 功能比较

功能	PIC16F676	PIC16F610/16HV610	PIC16F616/16HV616
最大工作速度	20 MHz	20 MHz	20 MHz
最大程序存储器 （字）	1024	1024	2048
SRAM （字节）	64	64	128
A/D 分辨率	10 位	无	10 位
定时器 （8/16 位）	1/1	1/1	2/1
振荡器模式	8	8	8
欠压复位	有	有	有
内部上拉	RA0/1/2/4/5	RA0/1/2/4/5， $\overline{\text{MCLR}}$	RA0/1/2/4/5， $\overline{\text{MCLR}}$
电平变化中断	RA0/1/2/3/4/5	RA0/1/2/3/4/5	RA0/1/2/3/4/5
比较器	1	2	2
ECCP	无	无	有
INTOSC 频率	4 MHz	8 MHz	8 MHz
内部并联稳压器	无	有 （PIC16HV610）	有 （PIC16HV616）

注：	该器件设计为按照数据手册上的参数来工作。它已通过电气规范测试，该规范用于确定器件是否符合这些参数。由于存在器件生产工艺上的差异，其性能特性可能与其早期版本的产品有所不同。这些不同可能导致该器件与其早期版本在应用中的性能差异。

注：

索引

 初稿

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。只要使用常用的因特网浏览器即可访问。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存档软件
- **一般技术支持**——常见问题 （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、 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 的客户通知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，点击“变更通知客户（Customer Change Notification）”服务并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://support.microchip.com获得网上技术支持。

读者反馈表

我们努力为您提供最佳文档，以确保您能够成功使用 Microchip 产品。如果您对文档的组织、条理性、主题及其他有助于提高文档质量的方面有任何意见或建议，请填写本反馈表并传真给我公司 TRC 经理，传真号码为 86-21-5407-5066。

请填写以下信息，并从下面各方面提出您对本文档的意见。

致：　TRC 经理　　　　　　　　　　　　　　　总页数 ________

关于：　读者反馈

发自：　姓名 ____________________

　　　　公司 ____________________

　　　　地址 ____________________

　　　　国家 / 省份 / 城市 / 邮编 ____________________

　　　　电话：(_______) _________ - _________　　　　传真：(______) _________ - _________

应用（选填）：

您希望收到回复吗？是 ____ 否____

器件：　PIC16F610/616/16HV610/616　文献编号：　DS41288B_CN

问题：

1. 本文档中哪些部分最有特色？

2. 本文档是否满足了您的软硬件开发要求？如何满足的？

3. 您认为本文档的组织结构便于理解吗？如果不便于理解，那么问题何在？

4. 您认为本文档应该添加哪些内容以改善其结构和主题？

5. 您认为本文档中可以删减哪些内容，而又不会影响整体使用效果？

6. 本文档中是否存在错误或误导信息？如果存在，请指出是什么信息及其具体页数。

7. 您认为本文档还有哪些方面有待改进？

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。

器件编号	X	/XX	XXX
器件	温度范围	封装	模式

器件：	PIC16F610/616/16HV610/616， PIC16F610/616/16HV610/616T(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：	ML = 正方扁平无引脚 （QFN） P = 塑封 DIP SL = 14 引脚小外形 （150 mil） ST = 超薄缩小外形 （4.4 mm）
模式：	QTP、SQTP 或 ROM 代码；特殊要求 （其他情况空白）

示例：

a） PIC16F610/616/16HV610/616-E/P 301 = 扩展级温度，PDIP 封装，20 MHz，QTP 模式 #301

b） PIC16F610/616/16HV610/616-I/SO = 工业级温度， SOIC 封装， 20 MHz

注 1： T = 卷带式，仅限 TSSOP 和 SOIC 封装。