



# **PIC16F627A/628A/648A**

## 数据手册

采用纳瓦技术的  
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 safest 的产品之一。
- 目前, 仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知, 所有这些行为都不是以 **Microchip** 数据手册中规定的操作规范来使用 **Microchip** 产品的。这样做的人极可能侵犯了知识产权。
- **Microchip** 愿与那些注重代码完整性的客户合作。
- **Microchip** 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。 **Microchip** 承诺将不断改进产品的代码保护功能。任何试图破坏 **Microchip** 代码保护功能的行为均可视为违反了《数字器件千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下, 能访问您的软件或其他受版权保护的成果, 您有权依据该法案提起诉讼, 从而制止这种行为。

---

提供本文档的中文版本仅为了便于理解。 **Microchip Technology Inc.** 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 **Microchip Technology Inc.** 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利, 它们可能由更新之信息所替代。确保应用符合技术规范, 是您自身应负的责任。 **Microchip** 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保, 包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。 **Microchip** 对因这些信息及使用这些信息而引起的后果不承担任何责任。未经 **Microchip** 书面批准, 不得将 **Microchip** 的产品用作生命维持系统中的关键组件。在 **Microchip** 知识产权保护下, 不得暗或以其他方式转让任何许可证。

#### 商标

**Microchip** 的名称和徽标组合、 **Microchip** 徽标、 **Accuron**、 **dsPIC**、 **KEELOQ**、 **microID**、 **MPLAB**、 **PIC**、 **PICmicro**、 **PICSTART**、 **PRO MATE**、 **PowerSmart**、 **rfPIC** 和 **SmartShunt** 均为 **Microchip Technology Inc.** 在美国和其他国家或地区的注册商标。

**AmpLab**、 **FilterLab**、 **Migratable Memory**、 **MXDEV**、 **MXLAB**、 **PICMASTER**、 **SEEVAL**、 **SmartSensor** 和 **The Embedded Control Solutions Company** 均为 **Microchip Technology Inc.** 在美国的注册商标。

**Analog-for-the-Digital Age**、 **Application Maestro**、 **dsPICDEM**、 **dsPICDEM.net**、 **dsPICworks**、 **ECAN**、 **ECONOMONITOR**、 **FanSense**、 **FlexROM**、 **fuzzyLAB**、 **In-Circuit Serial Programming**、 **ICSP**、 **ICEPIC**、 **Linear Active Thermistor**、 **MPASM**、 **MPLIB**、 **MPLINK**、 **MPSIM**、 **PICKit**、 **PICDEM**、 **PICDEM.net**、 **PICLAB**、 **PICtail**、 **PowerCal**、 **PowerInfo**、 **PowerMate**、 **PowerTool**、 **rfLAB**、 **rfPICDEM**、 **Select Mode**、 **Smart Serial**、 **SmartTel**、 **Total Endurance** 和 **WiperLock** 均为 **Microchip Technology Inc.** 在美国和其他国家或地区的商标。

**SQTP** 是 **Microchip Technology Inc.** 在美国的服务标记。

在此提及的所有其他商标均为各持有公司所有。

© 2005, **Microchip Technology Inc.** 版权所有。

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

**Microchip** 位于美国亚利桑那州 **Chandler** 和 **Tempe** 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于 2003 年 10 月通过了 **ISO/TS-16949:2002** 质量体系认证。公司在 **PICmicro® 8** 位单片机、 **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。

## 采用纳瓦技术的 18 引脚 8 位 CMOS 闪存单片机

### 高性能 RISC CPU:

- 工作速度可从 DC 到 20 MHz
- 中断能力
- 8 级深度硬件堆栈
- 直接、间接和相对寻址模式
- 35 条单字指令
  - 除了转移指令以外，所有指令均为单周期指令

### 单片机的特殊功能:

- 内部和外部振荡器选择:
  - 高精度的内部 4 MHz 振荡器，出厂时精度校准为 ± 1%
  - 低功耗内部 48kHz 振荡器
  - 可使用晶振和谐振器作为外部振荡器。
- 节能的休眠模式
- PORTB 上有可编程的弱上拉功能
- 主复位 / 输入引脚复用
- 看门狗定时器带有独立的振荡器，能保证可靠的运行
- 低电压编程
- 在线串行编程 (In-Circuit Serial Programming™) (通过两个引脚进行)
- 可编程代码保护
- 欠压复位
- 上电复位
- 上电延时定时器和振荡器起振定时器
- 宽工作电压范围 (2.0V 到 5.5V)
- 工业级和扩展级温度范围
- 高耐用性闪存 /EEPROM 单元
  - 闪存可经受 10 万次写操作
  - EEPROM 可经受 100 万次写操作
  - 数据保持期为 40 年

### 低功耗功能:

- 待机电流:
  - 当电压为 2.0V 时，典型值为 100 nA
- 工作电流:
  - 当频率为 32 kHz，电压为 2.0V 时，典型值为 12 μA
  - 当频率为 1 MHz，电压为 2.0V 时，典型值为 120 μA
- 看门狗定时器电流
  - 当电压为 2.0V 时，典型值为 1 μA
- Timer1 振荡器电流:
  - 当频率为 32 kHz，电压为 2.0V 时，典型值为 1.2 μA
- 双速内部振荡器:
  - 有 4 MHz 和 48kHz 两种频率可供选择
  - 从休眠状态唤醒 4 μs，3.0V，典型值

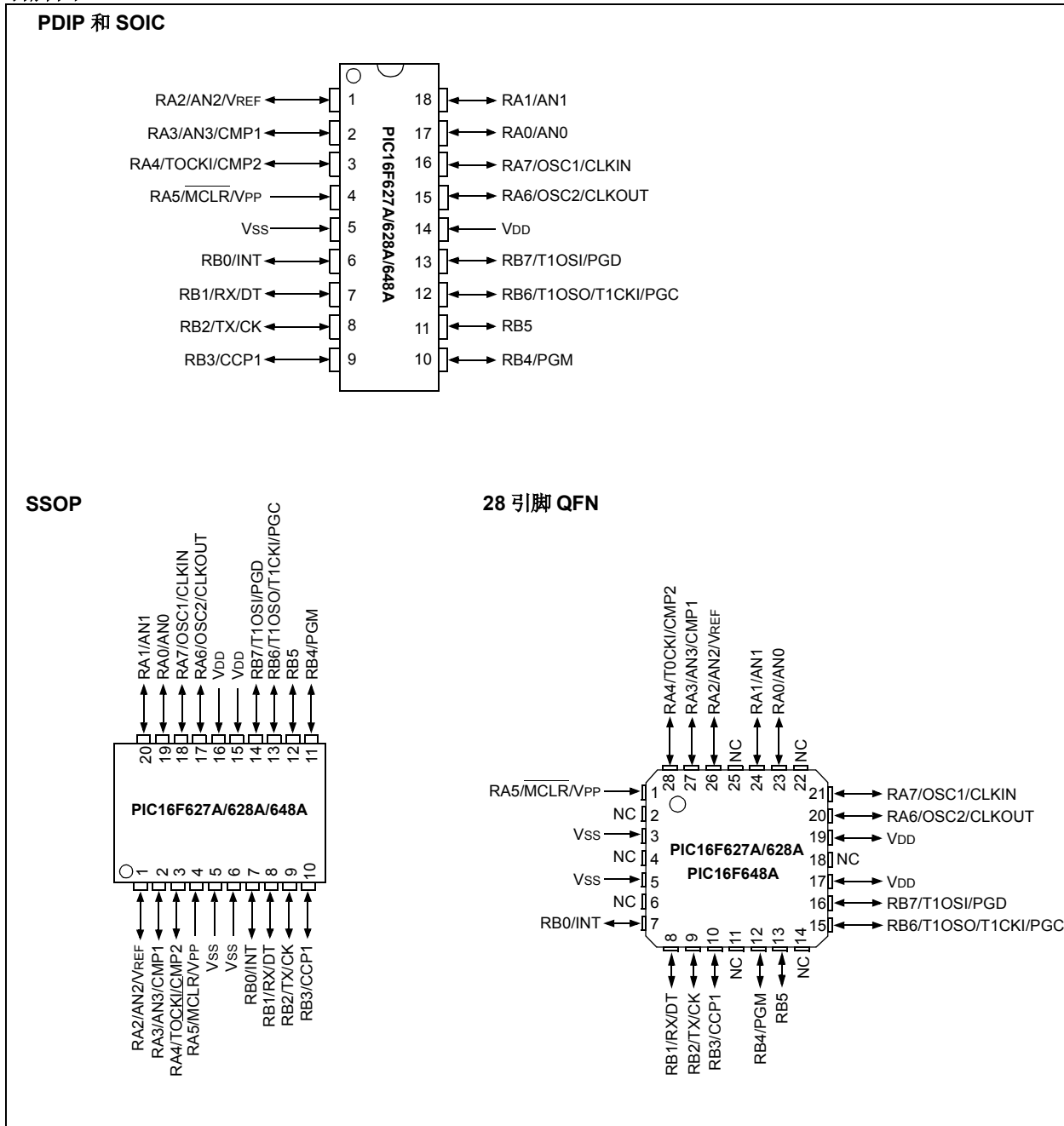
### 外设功能:

- 16 个具有独立方向控制的 I/O 引脚
- 较高灌 / 拉电流用于直接驱动 LED
- 模拟比较器模块带有:
  - 两个模拟比较器
  - 可编程的片上参考电压 (VREF) 模块。
  - 可选择的内部或外部参考电压
  - 可外部访问比较器输出
- Timer0: 带 8 位可编程预分频器的 8 位定时器 / 计数器
- Timer1: 带有外部晶振 / 时钟源功能的 16 位定时器 / 计数器
- Timer2: 带 8 位周期寄存器、预分频器和后分频器的 8 位定时器 / 计数器
- 捕捉、比较、PWM 模块
  - 16 位捕捉 / 比较
  - 10 位 PWM
- 可寻址的通用同步 / 异步收发器 USART/SCI

器件	程序存储器	数据存储器		I/O	CCP (PWM)	USART	比较器	定时器 8/16 位
	闪存 (字)	SRAM (字节)	EEPROM (字节)					
PIC16F627A	1024	224	128	16	1	有	2	2/1
PIC16F628A	2048	224	128	16	1	有	2	2/1
PIC16F648A	4096	256	256	16	1	有	2	2/1

# PIC16F627A/628A/648A

## 引脚图



## 目录

1.0 概述 .....	5
2.0 PIC16F627A/628A/648A 器件种类 .....	7
3.0 架构综述 .....	9
4.0 存储器构成 .....	15
5.0 I/O 口 .....	31
6.0 Timer0 模块 .....	45
7.0 Timer1 模块 .....	48
8.0 Timer2 模块 .....	52
9.0 捕捉 / 比较 / PWM (CCP) 模块 .....	55
10.0 比较器模块 .....	61
11.0 参考电压模块 .....	67
12.0 通用同步 / 异步收发器 (USART) 模块 .....	71
13.0 数据 EEPROM 存储器 .....	89
14.0 CPU 的特殊功能 .....	95
15.0 指令集综述 .....	115
16.0 开发支持 .....	129
17.0 电气规范 .....	135
18.0 直流和交流特性图表 .....	151
19.0 封装信息 .....	163
附录 A: 数据手册版本历史 .....	169
附录 B: 器件差异 .....	169
附录 C: 器件移植 .....	170
附录 D: 从其他 PICmicro® 器件移植到此系列 .....	170
Microchip 网站 .....	171
变更通知客户服务 .....	171
客户支持 .....	171
读者反馈表 .....	172
产品标识体系 .....	177

## 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 [CTRC@microchip.com](mailto:CTRC@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](http://www.microchip.com) 上注册。

# PIC16F627A/628A/648A

---

---

注:

# PIC16F627A/628A/648A

## 1.0 概述

PIC16F627A/628A/648A 系列单片机是 18 引脚的 8 位 CMOS 闪存单片机，具有多用途、低成本、高性能和全静态的特点。

所有 PICmicro® 单片机均采用先进的 RISC 架构。PIC16F627A/628A/648A 具有增强的内核功能、8 级深度的堆栈以及多种内部和外部中断源。哈佛架构独立的指令总线和数据总线，允许同时取 14 位宽指令字与独立的 8 位宽数据。两级指令流水线使得除了程序转移指令（需要两个周期）以外的所有指令都能在单个周期内执行。总共有 35 条指令（精简指令集）可用，辅之以大的寄存器组。

PIC16F627A/628A/648A 单片机与同类的其他 8 位单片机相比，通常能实现 2:1 的代码压缩率和 4 倍的速度提升。

PIC16F627A/628A/648A 器件集成了很多功能部件，从而减少了外部元件的使用，因此降低了系统成本，提高了系统可靠性，并降低了功耗。

PIC16F627A/628A/648A 有 8 种振荡器配置。单引脚的 RC 振荡器提供了低成本的解决方案。LP 振荡器可将功耗降至最低，XT 是标准晶振，而 INTOSC 是独立的高

精度双速内部振荡器。HS 模式是高速晶振。EC 模式则是采用外部时钟源。

休眠（断电）模式可以节能。用户可以通过几种外部中断、内部中断以及复位将芯片从休眠状态唤醒。

高可靠性的看门狗自带了片上 RC 振荡器，能够避免程序锁死。

表 1-1 给出了 PIC16F627A/628A/648A 中档单片机系列的特性。

图 3-1 是 PIC16F627A/628A/648A 的简化框图。

PIC16F627A/628A/648A 系列适合从电池充电器到低功耗远程传感器的一系列应用。闪存技术使得定制应用程序（如检测电平、脉冲产生、定时器等）变得十分便捷。小尺寸的封装使该系列单片机对于有空间限制的所有应用都很理想。低成本、低功耗、高性能、易使用以及 I/O 的灵活性使得 PIC16F627A/628A/648A 用途非常广泛。

## 1.1 开发支持

支持 PIC16F627A/628A/648A 系列的工具有：功能全面的宏汇编器、软件模拟器、在线仿真器、低成本在线调试器、低成本开发编程器和功能前面的编程器。还可使用第三方 C 编译器支持工具。

表 1-1: PIC16F627A/628A/648A 系列器件

		PIC16F627A	PIC16F628A	PIC16F648A	PIC16LF627A	PIC16LF628A	PIC16LF648A
时钟	最大工作频率 (MHz)	20	20	20	20	20	20
	闪存程序存储器 (字)	1024	2048	4096	1024	2048	4096
存储器	RAM 数据存储器 (字节)	224	224	256	224	224	256
	EEPROM 数据存储器 (字节)	128	128	256	128	128	256
	定时器模块	TMR0、TMR1、TMR2	TMR0、TMR1、TMR2	TMR0、TMR1、TMR2	TMR0、TMR1、TMR2	TMR0、TMR1、TMR2	TMR0、TMR1、TMR2
外设	比较器	2	2	2	2	2	2
	捕捉 / 比较 / PWM 模块	1	1	1	1	1	1
	串行通信	USART	USART	USART	USART	USART	USART
	内部参考电压	有	有	有	有	有	有
	中断源	10	10	10	10	10	10
特性	I/O 引脚	16	16	16	16	16	16
	电压范围 (V)	3.0-5.5	3.0-5.5	3.0-5.5	2.0-5.5	2.0-5.5	2.0-5.5
	欠压复位	有	有	有	有	有	有
	封装	18 引脚 DIP 和 SOIC、20 引脚 SSOP、28 引脚 QFN	18 引脚 DIP 和 SOIC、20 引脚 SSOP、28 引脚 QFN	18 引脚 DIP 和 SOIC、20 引脚 SSOP、28 引脚 QFN	18 引脚 DIP 和 SOIC、20 引脚 SSOP、28 引脚 QFN	18 引脚 DIP 和 SOIC、20 引脚 SSOP、28 引脚 QFN	18 引脚 DIP 和 SOIC、20 引脚 SSOP、28 引脚 QFN

PICmicro® 系列的所有器件都有上电复位、可选择的看门狗、可选择的代码保护以及高 I/O 电流性能。PIC16F627A/628A/648A 系列的所有器件使用时钟引脚 RB6 和数据引脚 RB7 进行串行编程。

# PIC16F627A/628A/648A

---

---

注:



## 2.0 PIC16F627A/628A/648A 器件种类

有多种频率和封装类型可供选择。根据应用和生产要求，可以通过器件数据手册最末的“PIC16F627A/628A/648A 产品识别体系”来选择适当的器件。在订购器件的时候，也请使用数据手册的这一页来指定正确的器件编号。

### 2.1 闪存器件

闪存器件是电可擦除和再编程的。这样可以在样片开发、试用和产品生产中都使用相同的器件。

电可擦除闪存更大的优点是它可以在线或使用器件编程器（例如 Microchip 的 PICSTART<sup>®</sup> Plus 或 PRO MATE<sup>®</sup> II 编程器）进行擦除或重新编程。

### 2.2 快速批量生产（Quick-Turnaround-Production, QTP）器件

Microchip 为工厂生产订单提供 QTP 编程服务。这种服务适用于那些不想对中到大批量单片机编程，并且代码已经相对稳定的用户。这些器件是标准的闪存器件，只是所有的编程单元和配置已在出厂前设定。Microchip 还在出厂前对某些代码和样片进行了校验。欲了解更详细信息，请联系当地 Microchip 销售办事处。

### 2.3 带序列号的快速批量生产（Serialized Quick-Turnaround-Production, SQTP<sup>SM</sup>）器件

Microchip 提供一种独特的编程服务，可用不同的序列号对每个器件中的几处用户自定义单元进行编程。这些序列号可以是随机数、伪随机数或连续数。

这种串行编程使每个器件具有唯一的序列号，可以作为登录码、口令或用户识别码。

# PIC16F627A/628A/648A

---

---

注:

## 3.0 架构综述

PIC16F627A/628A/648A 系列的高性能可以归功于 RISC 微处理器中普遍采用的架构特点。首先，PIC16F627A/628A/648A 采用了哈佛架构，在这个架构中，分别使用独立的总线从独立的存储器中存取程序和数据。与传统的程序和数据存储器合二为一的冯·诺依曼架构相比，哈佛架构具有更加优良的总线带宽。独立的程序和数据存储器允许指令宽度超过 8 位。由于指令操作码为 14 位宽，所以所有指令都可以是单字指令。通过 14 位宽的程序存储器总线可以在单周期内取一条 14 位的指令。两级流水线可以使取指和执行指令同时进行。因此，除了程序转移指令以外，所有指令（35 条）都可以在单周期（时钟频率为 20 MHz 时为 200ns）内执行。

表 3-1 所列为器件的存储器容量（闪存、数据存储器 and EEPROM）。

**表 3-1: 器件存储器列表**

器件	存储器		
	闪存程序	数据 RAM	数据 EEPROM
PIC16F627A	1024 x 14	224 x 8	128 x 8
PIC16F628A	2048 x 14	224 x 8	128 x 8
PIC16F648A	4096 x 14	256 x 8	256 x 8
PIC16LF627A	1024 x 14	224 x 8	128 x 8
PIC16LF628A	2048 x 14	224 x 8	128 x 8
PIC16LF648A	4096 x 14	256 x 8	256 x 8

PIC16F627A/628A/648A 可以直接或间接的寻址它的文件寄存器或数据存储器。所有特殊功能寄存器（Special Function Register, SFR），包括程序计数器，都映射到数据存储器空间。PIC16F627A/628A/648A 有一个正交（对称）指令集，因此它可以使用任何寻址模式对任一寄存器执行任何操作。指令集的对称特性以及无“特别理想状态”让使用 PIC16F627A/628A/648A 编程更简单有效。此外，所有这些都显著地简化了学习过程。

PIC16F627A/628A/648A 器件包含一个 8 位 ALU 和工作寄存器。ALU 是一个通用的算术单元。它对工作寄存器和任何文件寄存器中的数据进行算术和布尔运算。

ALU 为 8 位宽，能够进行加、减、移位和逻辑操作。除非特别指明，否则算术运算一般是以 2 进制补码的形式进行的。在双操作数指令中，一般情况下，其中的一个操作数是在工作寄存器（W 寄存器）中。另一个操作数存放在一个文件寄存器中或是立即数。在单操作数指令中，操作数放在 W 寄存器中或某个文件寄存器中。

W 寄存器是一个 8 位宽、用于 ALU 运算的工作寄存器。该寄存器不可寻址。

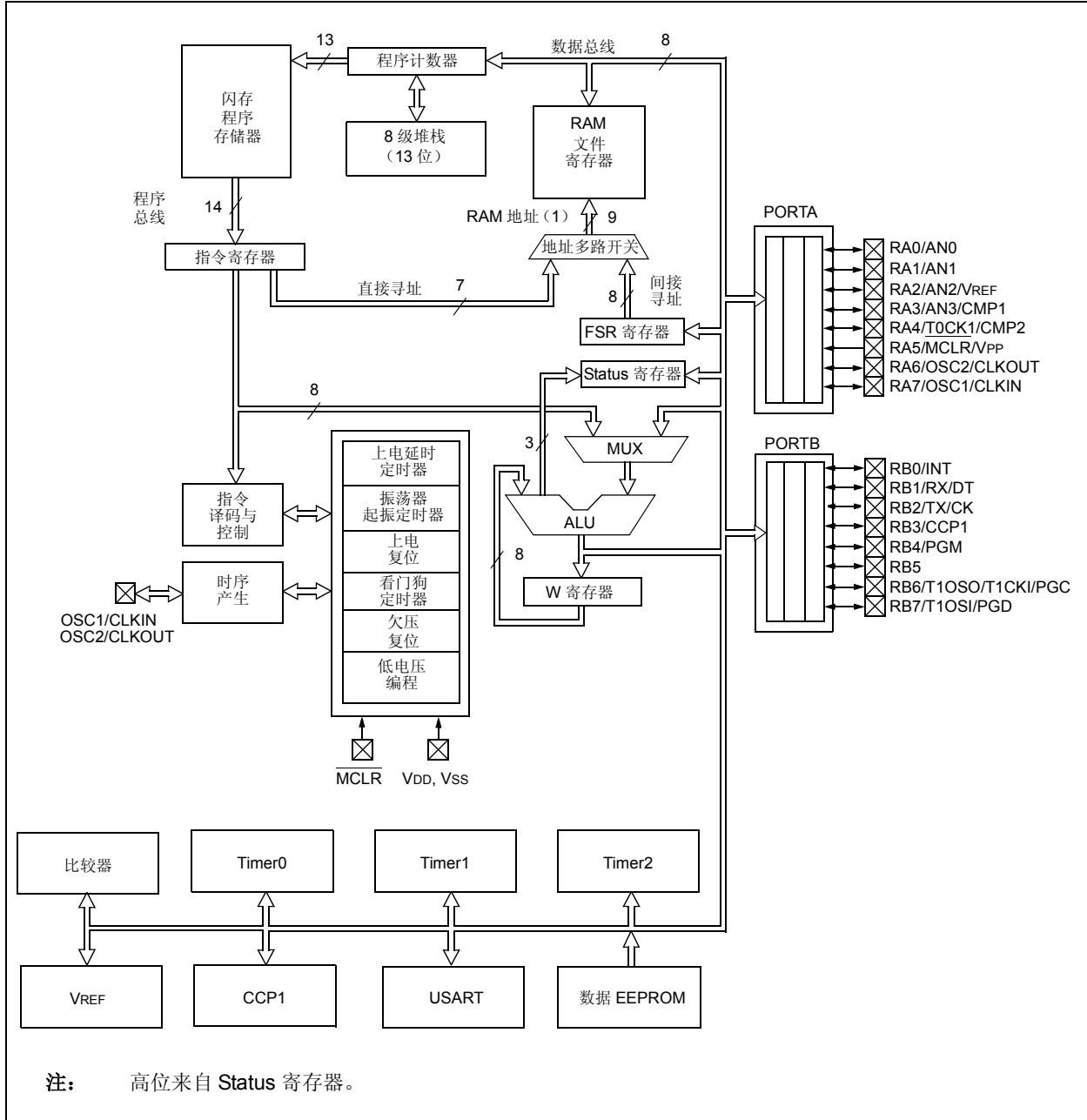
根据所执行的指令，ALU 可能影响 Status 寄存器中的进位标志位 C、辅助进位标志位 DC 和全零标志位 Z。在减法操作中，C 和 DC 位分别作为借位和辅助借位标志位。例如指令 SUBLW 和 SUBWF。

图 3-1 给出了简化框图，表 3-2 对器件引脚作了描述。

PIC16F627A/628A/648A 器件上提供了两种类型的数据存储器。所提供的非易失性 EEPROM 数据存储器用于长期存储数据，例如校准值、查表数据以及其他可能需要现场定期更新的数据。这些数据类型不会在掉电时丢失。所提供的另一个数据存储器是常规的 RAM 数据存储器。常规的 RAM 数据存储器用于对正常操作中的数据进行临时存储。这些数据会在掉电时丢失。

# PIC16F627A/628A/648A

图 3-1: 结构框图



# PIC16F627A/628A/648A

表 3-2: PIC16F627A/628A/648A 引脚配置描述

名称	功能	输入类型	输出类型	描述
RA0/AN0	RA0	ST	CMOS	双向 I/O 端口
	AN0	AN	—	模拟比较器输入
RA1/AN1	RA1	ST	CMOS	双向 I/O 端口
	AN1	AN	—	模拟比较器输入
RA2/AN2/VREF	RA2	ST	CMOS	双向 I/O 端口
	AN2	AN	—	模拟比较器输入
	VREF	—	AN	VREF 输出
RA3/AN3/CMP1	RA3	ST	CMOS	双向 I/O 端口
	AN3	AN	—	模拟比较器输入
	CMP1	—	CMOS	比较器 1 输出
RA4/T0CKI/CMP2	RA4	ST	OD	双向 I/O 端口
	T0CKI	ST	—	Timer0 时钟输入
	CMP2	—	OD	比较器 2 输出
RA5/ $\overline{\text{MCLR}}$ /VPP	RA5	ST	—	输入端口
	$\overline{\text{MCLR}}$	ST	—	主复位。如果将引脚配置为 $\overline{\text{MCLR}}$ ，这是一个低电平有效的引脚，即低电平时器件复位。器件正常运行时 $\overline{\text{MCLR}}$ /VPP 上的电压不能超过 VDD。
	VPP	—	—	编程电压输入
RA6/OSC2/CLKOUT	RA6	ST	CMOS	双向 I/O 端口
	OSC2	—	XTAL	振荡器晶振输出。在晶体振荡器模式连接到晶振和谐振器。
	CLKOUT	—	CMOS	在 RC/INTOSC 模式，OSC2 引脚可以输出 CLKOUT，其频率为 OSC1 的 1/4。
RA7/OSC1/CLKIN	RA7	ST	CMOS	双向 I/O 端口
	OSC1	XTAL	—	振荡器晶振输入
	CLKIN	ST	—	外部时钟源输入。RC 偏置引脚。
RB0/INT	RB0	TTL	CMOS	双向 I/O 端口。可以软件编程实现内部弱上拉。
	INT	ST	—	外部中断。
RB1/RX/DT	RB1	TTL	CMOS	双向 I/O 端口。可以软件编程实现内部弱上拉。
	RX	ST	—	USART 接收引脚
	DT	ST	CMOS	同步数据 I/O
RB2/TX/CK	RB2	TTL	CMOS	双向 I/O 端口。可以软件编程实现内部弱上拉。
	TX	—	CMOS	USART 发送引脚
	CK	ST	CMOS	同步时钟 I/O
RB3/CCP1	RB3	TTL	CMOS	双向 I/O 端口。可以软件编程实现内部弱上拉。
	CCP1	ST	CMOS	捕捉 / 比较 / PWM I/O

图注: O = 输出  
— = 未使用  
TTL = TTL 输入

CMOS = CMOS 输出  
I = 输入  
OD = 漏极开路输出

P = 电源  
ST = 施密特触发器输入  
AN = 模拟

# PIC16F627A/628A/648A

表 3-2: PIC16F627A/628A/648A 引脚配置描述 (续)

名称	功能	输入类型	输出类型	描述
RB4/PGM	RB4	TTL	CMOS	双向 I/O 端口。引脚电平变化可触发中断。可以软件编程实现内部弱上拉。
	PGM	ST	—	低电压编程输入引脚。当低电压编程被使能时，禁止引脚电平变化触发中断以及弱上拉电阻。
RB5	RB5	TTL	CMOS	双向 I/O 端口。引脚电平变化可触发中断。可以软件编程实现内部弱上拉。
RB6/T1OSO/T1CKI/PGC	RB6	TTL	CMOS	双向 I/O 端口。引脚电平变化可触发中断。可以软件编程实现内部弱上拉。
	T1OSO	—	XTAL	Timer1 振荡器输出。
	T1CKI	ST	—	Timer1 时钟输入。
	PGC	ST	—	ICSPTM 编程时钟。
RB7/T1OSI/PGD	RB7	TTL	CMOS	双向 I/O 端口。引脚电平变化可触发中断。可以软件编程实现内部弱上拉。
	T1OSI	XTAL	—	Timer1 振荡器输入。
	PGD	ST	CMOS	ICSP 数据 I/O
VSS	VSS	电源	—	逻辑和 I/O 引脚的接地参考点
VDD	VDD	电源	—	逻辑和 I/O 引脚的正电源

图注:     O = 输出  
           — = 未使用  
           TTL = TTL 输入

CMOS = CMOS 输出  
       I   = 输入  
       OD  = 漏极开路输出

P   = 电源  
      ST = 施密特触发器输入  
      AN = 模拟

## 3.1 时序图 / 指令周期

时钟输入（OSC1/CLKIN/RA7 引脚）信号在器件内部经过 4 分频后产生 4 个不重叠的正交时钟信号，即 Q1、Q2、Q3 和 Q4。在此过程中，程序计数器（PC）在每个 Q1 时递增 1，并在 Q4 时从程序存储器取指并将指令锁存到指令寄存器中。在接下来的 Q1 到 Q4 周期中进行指令的译码和执行。图 3-2 所示为时钟和指令执行流程。

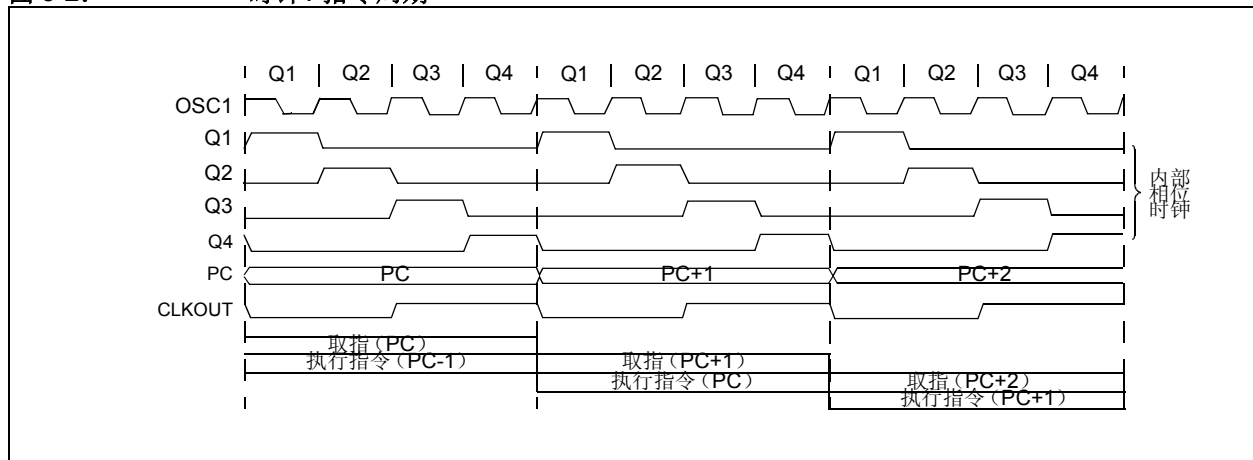
## 3.2 指令流 / 流水线

一个指令周期由 4 个 Q 周期组成（Q1、Q2、Q3 和 Q4）。取指和执行指令是流水线操作的，用一个指令周期来取指，而用另一个指令周期来译码和执行取到的指令。但由于是流水线操作，所以每条指令的等效执行时间都是一个指令周期。如果指令（例如，GOTO）改变了程序计数器，则需要两个指令周期来完成指令（例 3-1）。

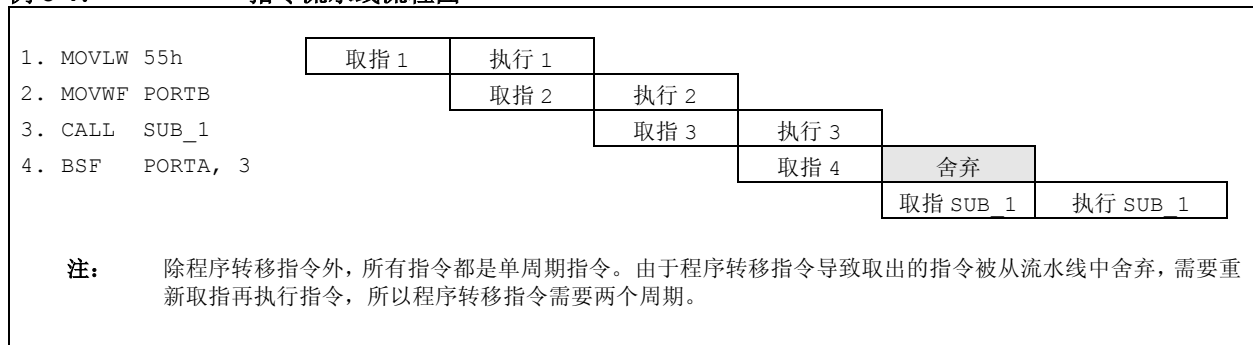
在 Q1 周期开始取指操作，程序计数器（PC）加 1。

指令的执行过程：在 Q1 周期，将所取指令锁存到指令寄存器（IR）。在 Q2、Q3 和 Q4 周期中进行指令的译码和执行。其中读数据存储器（读操作数）发生在 Q2 周期，写操作发生在 Q4 周期（写入目标单元）。

图 3-2: 时钟 / 指令周期



例 3-1: 指令流水线流程图



# PIC16F627A/628A/648A

---

---

注:

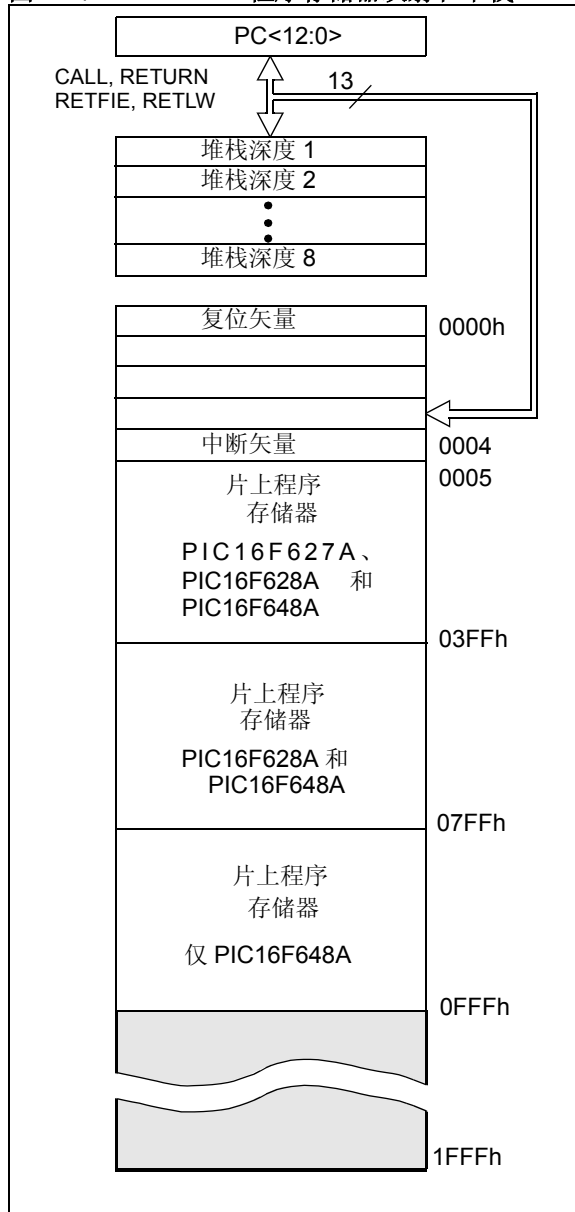


## 4.0 存储器构成

### 4.1 程序存储器构成

PIC16F627A/628A/648A 有一个 13 位的程序计数器，能寻址 8K x 14 的程序存储器空间。物理实现的只有 PIC16F627A 使用的第一个 1K x 14 (0000h – 03FFh) 单元、PIC16F628A 使用的 2K x 14 (0000h – 07FFh) 单元和 PIC16F648A 使用的 4K x 14 (0000h-0FFFh) 单元。对超出以上范围的单元寻址时，将会导致在第一个 1K x 14 空间 (PIC16F627A)、2K x 14 空间 (PIC16F628A) 或 4K x 14 空间 (PIC16F648A) 内部溢出返回。复位矢量位于 0000h，中断矢量位于 0004h (图 4-1)。

图 4-1: 程序存储器映射和堆栈



### 4.2 数据存储器的构成

数据存储区 (图 4-2 和图 4-3) 被划分为 4 块存储区，其中包含通用寄存器 (General Purpose Register, GPR) 和特殊功能寄存器 (Special Function Register, SFR)。SFR 位于每个存储区的前 32 个存储单元。每个存储区中的通用寄存器都是以静态 RAM 的形式实现的。表 4-1 列出了这 4 个存储区中可用作通用寄存器的区域。

表 4-1: 通用静态 RAM 寄存器

	PIC16F627A/628A	PIC16F648A
存储区 0	20-7Fh	20-7Fh
存储区 1	A0h-FF	A0h-FF
存储区 2	120h-14Fh, 170h-17Fh	120h-17Fh
存储区 3	1F0h-1FFh	1F0h-1FFh

地址为 F0h-FFh、170h-17Fh 和 1F0h-1FFh 的存储单元是以通用 RAM 的形式实现的，并映射回地址为 70h-7Fh 的单元。

表 4-2 说明了如何通过 Status 寄存器的 RP1 和 RP0 位访问这 4 个存储区中的寄存器。

表 4-2: 访问存储区中的寄存器

存储区	RP1	RP0
0	0	0
1	0	1
2	1	0
3	1	1

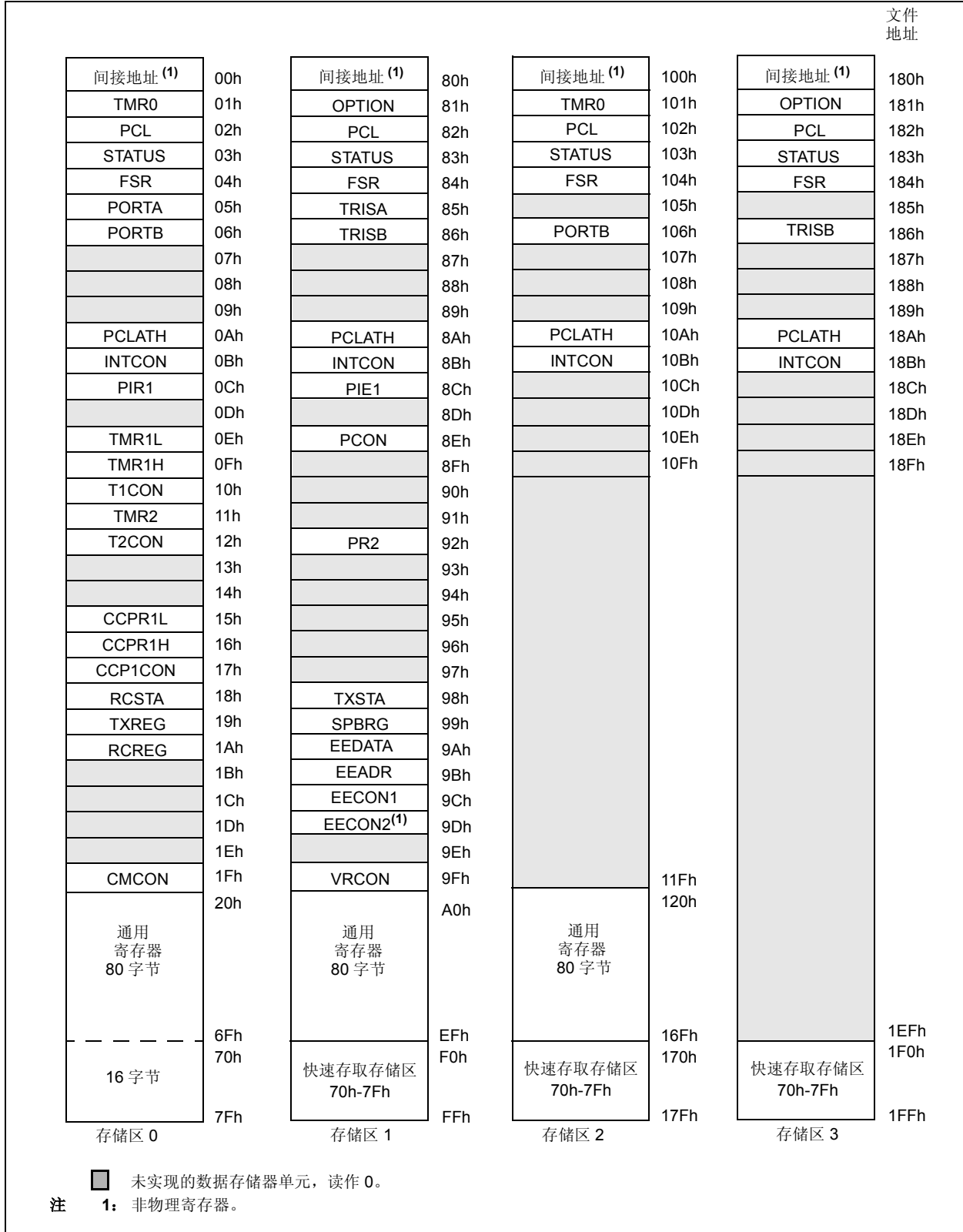
#### 4.2.1 通用文件寄存器

文件寄存器在 PIC16F627A/628A 中被组织为 224 x 8，而在 PIC16F648A 中则是 256 x 8。可直接访问或通过指针寄存器 (File Select Register, FSR) 间接访问每个寄存器，参见第 4.4 节“间接寻址、INDF 和 FSR 寄存器”。



# PIC16F627A/628A/648A

图 4-3: PIC16F648A 的数据存储器映射



# PIC16F627A/628A/648A

## 4.2.2 特殊功能寄存器

SFR 用于 CPU 和外设功能，控制器件执行期望操作的寄存器（表 4-3）。这些寄存器是静态 RAM。

这些特殊功能寄存器可以分成两类（内核与外设）。本节讲述与“内核”有关的 SFR。与外设功能部件操作有关的 SFR 将外设功能部件章节中描述。

**表 4-3: 特殊功能寄存器综述存储区 0**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 复位时的值 <sup>(1)</sup>	详情请见: (页)
<b>存储区 0</b>											
00h	INDF	对此存储单元寻址会使用 FSR 的内容对数据存储器寻址（非物理寄存器）								xxxx xxxx	28
01h	TMR0	Timer0 模块寄存器								xxxx xxxx	45
02h	PCL	程序计数器（PC）最低有效字节								0000 0000	28
03h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	22
04h	FSR	间接数据存储器地址指针								xxxx xxxx	28
05h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx 0000	31
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	36
07h	—	未实现位								—	—
08h	—	未实现位								—	—
09h	—	未实现位								—	—
0Ah	PCLATH	—	—	—	程序计数器高 5 位的写缓冲器			---	0 0000	28	
0Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	24
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	26
0Dh	—	未实现位								—	—
0Eh	TMR1L	16 位 TMR1 最低有效字节的保持寄存器								xxxx xxxx	48
0Fh	TMR1H	16 位 TMR1 最高有效字节的保持寄存器								xxxx xxxx	48
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	48
11h	TMR2	TMR2 模块寄存器								0000 0000	52
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	52
13h	—	未实现位								—	—
14h	—	未实现位								—	—
15h	CCPR1L	捕捉 / 比较 / PWM 寄存器（LSB）								xxxx xxxx	55
16h	CCPR1H	捕捉 / 比较 / PWM 寄存器（MSB）								xxxx xxxx	55
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	55
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	71
19h	TXREG	USART 发送数据寄存器								0000 0000	77
1Ah	RCREG	USART 接收数据寄存器								0000 0000	80
1Bh	—	未实现位								—	—
1Ch	—	未实现位								—	—
1Dh	—	未实现位								—	—
1Eh	—	未实现位								—	—
1Fh	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	61

**图注:** — = 未实现单元，读作 0，u = 未改变，x = 未知，q = 值随条件变化，阴影 = 未实现

**注 1:** 如需了解寄存器表的初始化状态，请参见表 14-6 和表 1-7。

# PIC16F627A/628A/648A

表 4-4: 特殊功能寄存器综述存储区 1

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 复位时的值 <sup>(1)</sup>	详情请见: (页)
存储区 1											
80h	INDF	对此存储单元寻址会使用 FSR 的内容对数据存储寄存器寻址。(非物理寄存器)								xxxx xxxx	28
81h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	23
82h	PCL	程序计数器 (PC) 最低有效字节								0000 0000	28
83h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	22
84h	FSR	间接数据存储寄存器地址指针								xxxx xxxx	28
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	31
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	36
87h	—	未实现位								—	—
88h	—	未实现位								—	—
89h	—	未实现位								—	—
8Ah	PCLATH	—	—	—	程序计数器高 5 位的写缓冲器					---0 0000	28
8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	24
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	25
8Dh	—	未实现位								—	—
8Eh	PCON	—	—	—	—	OSCF	—	$\overline{POR}$	$\overline{BOR}$	---- 1-0x	27
8Fh	—	未实现位								—	—
90h	—	未实现位								—	—
91h	—	未实现位								—	—
92h	PR2	Timer2 周期寄存器								1111 1111	52
93h	—	未实现位								—	—
94h	—	未实现位								—	—
95h	—	未实现位								—	—
96h	—	未实现位								—	—
97h	—	未实现位								—	—
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	73
99h	SPBRG	波特率发生器寄存器								0000 0000	73
9Ah	EEDATA	EEPROM 数据寄存器								xxxx xxxx	89
9Bh	EEADR	EEPROM 地址寄存器								xxxx xxxx	90
9Ch	EECON1	—	—	—	—	WRERR	WREN	WR	RD	---- x000	90
9Dh	EECON2	EEPROM 控制寄存器 2 (非物理寄存器)								---- ----	90
9Eh	—	未实现位								—	—
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	67

图注: — = 未实现单元, 读作 0, u = 未改变, x = 未知, q = 值随条件变化, 阴影 = 未实现  
注 1: 如需了解寄存器表的初始化状态, 请参见表 14-6 和表 1-7。

# PIC16F627A/628A/648A

表 4-5: 特殊功能寄存器综述存储区 2

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 复位时的值 <sup>(1)</sup>	详情请见: (页)
存储区 2											
100h	INDF	对此存储单元寻址会使用 FSR 的内容对数据存储器寻址。(非物理寄存器)								xxxx xxxx	28
101h	TMR0	Timer0 模块寄存器								xxxx xxxx	45
102h	PCL	程序计数器 (PC) 最低有效字节								0000 0000	28
103h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	22
104h	FSR	间接数据存储器地址指针								xxxx xxxx	28
105h	—	未实现位								—	—
106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	36
107h	—	未实现位								—	—
108h	—	未实现位								—	—
109h	—	未实现位								—	—
10Ah	PCLATH	—	—	—	程序计数器高 5 位的写缓冲器					---0 0000	28
10Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	24
10Ch	—	未实现位								—	—
10Dh	—	未实现位								—	—
10Eh	—	未实现位								—	—
10Fh	—	未实现位								—	—
110h	—	未实现位								—	—
111h	—	未实现位								—	—
112h	—	未实现位								—	—
113h	—	未实现位								—	—
114h	—	未实现位								—	—
115h	—	未实现位								—	—
116h	—	未实现位								—	—
117h	—	未实现位								—	—
118h	—	未实现位								—	—
119h	—	未实现位								—	—
11Ah	—	未实现位								—	—
11Bh	—	未实现位								—	—
11Ch	—	未实现位								—	—
11Dh	—	未实现位								—	—
11Eh	—	未实现位								—	—
11Fh	—	未实现位								—	—

图注: — = 未实现单元, 读作 0, u = 未改变, x = 未知, q = 值随条件变化, 阴影 = 未实现

注 1: 如需了解寄存器表的初始化状态, 请参见表 14-6 和表 1-7。

# PIC16F627A/628A/648A

表 4-6: 特殊功能寄存器综述存储区 3

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 复位时的值 <sup>(1)</sup>	详情请见: (页)
存储区 3											
180h	INDF	对此存储单元寻址会使用 FSR 的内容对数据存储单元寻址 (非物理寄存器)								xxxx xxxx	28
181h	OPTION	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	23
182h	PCL	程序计数器 (PC) 最低有效字节								0000 0000	28
183h	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	22
184h	FSR	间接数据存储单元地址指针								xxxx xxxx	28
185h	—	未实现位								—	—
186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	36
187h	—	未实现位								—	—
188h	—	未实现位								—	—
189h	—	未实现位								—	—
18Ah	PCLATH	—	—	—	程序计数器高 5 位的写缓冲器					---0 0000	28
18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	24
18Ch	—	未实现位								—	—
18Dh	—	未实现位								—	—
18Eh	—	未实现位								—	—
18Fh	—	未实现位								—	—
190h	—	未实现位								—	—
191h	—	未实现位								—	—
192h	—	未实现位								—	—
193h	—	未实现位								—	—
194h	—	未实现位								—	—
195h	—	未实现位								—	—
196h	—	未实现位								—	—
197h	—	未实现位								—	—
198h	—	未实现位								—	—
199h	—	未实现位								—	—
19Ah	—	未实现位								—	—
19Bh	—	未实现位								—	—
19Ch	—	未实现位								—	—
19Dh	—	未实现位								—	—
19Eh	—	未实现位								—	—
19Fh	—	未实现位								—	—

图注: — = 未实现单元, 读作 0, u = 未改变, x = 未知, q = 值随条件变化, 阴影 = 未实现  
注 1: 如需了解寄存器表的初始化状态, 请参见表 14-6 和表 1-7。

# PIC16F627A/628A/648A

## 4.2.2.1 STATUS 寄存器

Status 寄存器（如寄存器 4-1 所示）含有 ALU 的算术运算结果状态、复位状态及数据存储器（SRAM）存储区选择位。

与其他寄存器一样，Status 寄存器也可以是任何指令的目标寄存器。如果 Status 寄存器作为一条影响 Z、DC 或 C 标志位的指令的目标寄存器，则会禁止写入这三个标志位。根据器件逻辑，这些位会置位或清零。而且，TO 和 PD 位不可写。所以执行一条把 Status 寄存器作为目标寄存器的指令所得到的结果可能和预想的不一样。

例如，执行指令 CLRF STATUS 将会把 STATUS 寄存器的高 3 位清零，并将 Z 位置位。这将使 Status 寄存器的结果为 000uu1uu（其中 u 表示未变化）。

因此，建议仅使用 BCF、BSF、SWAPF 和 MOVWF 指令去改变 Status 寄存器，因为这些指令不会影响任何状态位。对于其他不影响任何状态位的指令，请参见“指令集综述”。

**注 1:** 在减法运算中，C 和 DC 位分别作为借位和辅助借位位。例如 SUBLW 和 SUBWF 指令。

寄存器 4-1:

**STATUS — 状态寄存器（地址：03h、83h、103h、183h）**

	R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
	IRP	RP1	RP0	TO	PD	Z	DC	C
bit 7								bit 0

- bit 7 **IRP:** 寄存器存储区选择位（用于间接寻址）  
1= 存储区 2, 3 (100h - 1FFh)  
0= 存储区 0, 1 (00h - FFh)
  - bit 6-5 **RP<1:0>:** 寄存器存储区选择位（用于直接寻址）  
00= 存储区 0 (00h - 7Fh)  
01= 存储区 1 (80h - FFh)  
10= 存储区 2 (100h - 17Fh)  
11= 存储区 3 (180h - 1FFh)
  - bit 4 **TO:** 超时位  
1= 上电后，执行 CLRWDW 指令或 SLEEP 指令  
0= 发生了 WDT 超时
  - bit 3 **PD:** 掉电标志位  
1= 在上电后或执行 CLRWDW 指令  
0= 执行 SLEEP 指令
  - bit 2 **Z:** 全零标志位  
1= 算术运算或逻辑运算结果为零  
0= 算术运算或逻辑运算结果不是零
  - bit 1 **DC:** 辅助进位 / 借位标志位（ADDWF、ADDLW、SUBLW、SUBWF 指令影响这些标志位）（对于借位，其极性是相反的）  
1= 结果中发生了第 4 低位向高位的进位  
0= 结果中第 4 低位没有向高位进位
  - bit 0 **C:** 进位 / 借位标志位（ADDWF、ADDLW、SUBLW、SUBWF 指令影响这些标志位）  
1= 结果中发生了最高位进位  
0= 结果中未发生最高位进位
- 注:** 对于借位，极性是相反的。执行减法是通过加上第二个操作数的二进制补码来完成的。对于移位指令（RRF 和 RLF），此位值来自源寄存器的最高位或最低位。

**图注:**

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



# PIC16F627A/628A/648A

## 4.2.2.2 OPTION 寄存器

OPTION 寄存器是可读写的寄存器，它包含各个控制位，用来配置 TMR0/WDT 预分频器、外部 RB0/INT 中断、TMR0 和 PORTB 弱上拉。

**注：** 要使 TMR0 的预分频比达到 1:1，应将预分频器分配给 WDT (PSA=1)。参见第 6.3.1 节“切换预分频器的分配”。

### 寄存器 4-2: OPTION\_REG — 选择寄存器 (地址: 81h, 181h)

	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7								bit 0

- bit 7 **RBPU:** PORTB 上拉使能位  
1= 禁止 PORTB 上拉  
0= 按各端口锁存器值使能 PORTB 上拉
- bit 6 **INTEDG:** 中断触发边沿选择位  
1= 在 RB0/INT 引脚的上升沿触发中断  
0= 在 RB0/INT 引脚的下降沿触发中断
- bit 5 **T0CS:** TMR0 时钟源选择位  
1= RA4/T0CKI/CMP2 引脚的信号作为时钟源  
0= 内部指令周期时钟 (CLKOUT) 作为时钟源
- bit 4 **T0SE:** TMR0 计数边沿选择位  
1= 在 RA4/T0CKI/CMP2 引脚电平发生下跳变时递增  
0= 在 RA4/T0CKI/CMP2 引脚电平发生上跳变时递增
- bit 3 **PSA:** 预分频器分配位  
1= 将预分频器分配给 WDT  
0= 将预分频器分配给 Timer0 模块
- bit 2-0 **PS<2:0>:** 预分频器分频比选择位

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

**图注:**

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

# PIC16F627A/628A/648A

## 4.2.2.3 INTCON 寄存器

INTCON 寄存器是可读写的寄存器，其中包含除比较器模块外的所有中断源的允许和标志位。如需了解比较器中断允许和标志位的说明，请参见第 4.2.2.4 节“PIE1 寄存器”和第 4.2.2.5 节“PIR1 寄存器”。

**注：** 当一个中断条件发生时，不管相应的中断允许位或全局允许位 GIE (INTCON<7>) 的状态如何，中断标志位都将置位。

### 寄存器 4-3: INTCON — 中断控制寄存器 (地址: 0Bh、8Bh、10Bh、18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit 7							bit 0

- bit 7 **GIE:** 全局中断允许位  
1= 允许所有未屏蔽的中断  
0= 禁止所有中断
- bit 6 **PEIE:** 外设中断允许位  
1= 允许所有未屏蔽的外设中断  
0= 禁止所有外设中断
- bit 5 **TOIE:** TMR0 溢出中断允许位  
1= 允许 TMR0 中断  
0= 禁止 TMR0 中断
- bit 4 **INTE:** RB0/INT 外部中断允许位  
1= 允许 RB0/INT 外部中断  
0= 禁止 RB0/INT 外部中断
- bit 3 **RBIE:** RB 端口电平变化中断允许位  
1= 允许 RB 端口电平变化中断  
0= 禁止 RB 端口电平变化中断
- bit 2 **TOIF:** TMR0 溢出中断标志位  
1=TMR0 寄存器已经溢出 (必须用软件清零)  
0=TMR0 寄存器未发生溢出
- bit 1 **INTE:** RB0/INT 外部中断标志位  
1= 发生了 RB0/INT 外部中断 (必须用软件清零)  
0= 未发生 RB0/INT 外部中断
- bit 0 **RBIF:** RB 端口电平变化中断标志位  
1=RB<7:4> 引脚中至少有一个引脚改变了状态 (必须用软件清零)  
0=RB<7:4> 各引脚的状态都没有改变

**图注:**

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

# PIC16F627A/628A/648A

## 4.2.2.4 PIE1 寄存器

此寄存器包含中断允许位。

### 寄存器 4-4: PIE1 — 外设中断允许寄存器 1 (地址: 8Ch)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

- bit 7 **EEIE:** EE 写操作完成中断允许位  
1= 允许 EE 写操作完成中断  
0= 禁止 EE 写操作完成中断
- bit 6 **CMIE:** 比较器中断允许位  
1= 允许比较器中断  
0= 禁止比较器中断
- bit 5 **RCIE:** USART 接收中断允许位  
1= 允许 USART 接收中断  
0= 禁止 USART 接收中断
- bit 4 **TXIE:** USART 发送中断允许位  
1= 允许 USART 发送中断  
0= 禁止 USART 发送中断
- bit 3 **未实现位:** 读作 0
- bit 2 **CCP1IE:** CCP1 中断允许位  
1= 允许 CCP1 中断  
0= 禁止 CCP1 中断
- bit 1 **TMR2IE:** TMR2 与 PR2 匹配中断允许位  
1= 允许 TMR2 与 PR2 匹配中断  
0= 禁止 TMR2 与 PR2 匹配中断
- bit 0 **TMR1IE:** TMR1 溢出中断允许位  
1= 允许 TMR1 溢出中断  
0= 禁止 TMR1 溢出中断

#### 图注:

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

# PIC16F627A/628A/648A

## 4.2.2.5 PIR1 寄存器

此寄存器包含中断标志位。

**注：** 当一个中断条件发生时，不管相应的中断允许位或全局允许位 **GIE** (**INTCON<7>**) 的状态如何，中断标志位都将置位。用户软件应在允许一个中断之前，确保先将相应的中断标志位清零。

### 寄存器 4-5: PIR1 — 外设中断 1 寄存器 (地址: 0Ch)

R/W-0	R/W-0	R-0	R-0	U-0	R/W-0	R/W-0	R/W-0	
EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	
bit 7					bit 0			

- bit 7     **EEIF:** EEPROM 写操作中断标志位  
1= 写操作完成 (必须用软件清零)  
0= 写操作未完成或未开始
- bit 6     **CMIF:** 比较器中断标志位  
1= 比较器输出发生了改变  
0= 比较器输出没有改变
- bit 5     **RCIF:** USART 接收中断标志位  
1=USART 接收缓冲器已满  
0=USART 接收缓冲器是空的
- bit 4     **TXIF:** USART 发送中断标志位  
1=USART 发送缓冲器是空的  
0=USART 发送缓冲器已满
- bit 3     **未实现位:** 读作 0
- bit 2     **CCP1IF:** CCP1 中断标志位  
捕捉模式  
1= 发生了 TMR1 寄存器捕捉 (必须用软件清零)  
0= 没有发生 TMR1 寄存器捕捉  
比较模式  
1= 发生了 TMR1 寄存器比较匹配 (必须用软件清零)  
0= 没有发生 TMR1 寄存器比较匹配  
PWM 模式  
在此模式下未使用
- bit 1     **TMR2IF:** TMR2 与 PR2 匹配中断标志位  
1= 发生了 TMR2 与 PR2 匹配 (必须用软件清零)  
0=TMR2 与 PR2 不匹配
- bit 0     **TMR1IF:** TMR1 溢出中断标志位  
1=TMR1 寄存器溢出 (必须用软件清零)  
0=TMR1 寄存器没有溢出

**图注:**

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

# PIC16F627A/628A/648A

## 4.2.2.6 PCON 寄存器

PCON 寄存器包含用来区分上电复位、外部  $\overline{\text{MCLR}}$  复位、WDT 复位或欠压复位的标志位。

**注：**  $\overline{\text{BOR}}$  在上电复位时未知。它必须在上电复位后由用户置位，并在接下来的复位中检查  $\overline{\text{BOR}}$  位是否为清零状态，从而指示是否发生的是欠压复位。如果禁止了欠压电路（将配置字中的  $\overline{\text{BOREN}}$  位清零），则  $\overline{\text{BOR}}$  状态位可忽略，也没必要对其内容作出预测。

### 寄存器 4-6: PCON — 电源控制寄存器（地址：8Eh）

U-0	U-0	U-0	U-0	R/W-1	U-0	R/W-0	R/W-x
—	—	—	—	OSCF	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7				bit 0			

- bit 7-4 **未实现位：** 读作 0
- bit 3 **OSCF：** INTOSC 振荡器频率位  
1= 典型值 4 MHz  
0= 典型值 48 kHz
- bit 2 **未实现位：** 读作 0
- bit 1  **$\overline{\text{POR}}$ ：** 上电复位状态位  
1= 没有发生上电复位  
0= 发生了上电复位（在上电复位发生之后必须用软件置位）
- bit 0  **$\overline{\text{BOR}}$ ：** 欠压复位状态位  
1= 没有发生欠压复位  
0= 发生了欠压复位（必须在发生欠压复位后用软件置位）

#### 图注：

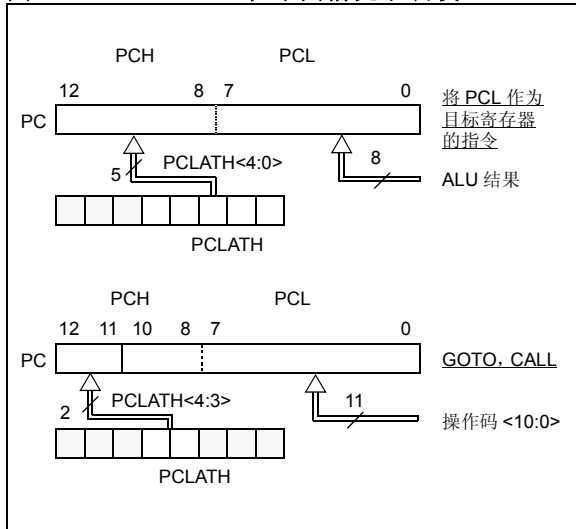
R= 可读位	W= 可写位	U= 未实现位，读作 0
-n=POR 时的值	1= 置位	0= 清零      x= 未知

# PIC16F627A/628A/648A

## 4.3 PCL 和 PCLATH

程序计数器 (PC) 为 13 位宽。它的低字节来自可读写的 PCL 寄存器。高字节 (PC<12:8>) 不可直接读写, 来自 PCLATH。任何复位时, PC 都会被清零。图 4-4 给出了装载 PC 的两种情况。图 4-4 中上面的示例给出了在写 PCL (PCLATH<4:0> → PCH) 时, 装载 PC 的过程。图 4-4 中下面的示例给出了在 CALL 或 GOTO 指令执行时 (PCLATH<4:3> → PCH) 装载 PC 的过程。

图 4-4: 在不同情况下装载 PC



### 4.3.1 相对 GOTO

相对 GOTO 是通过向程序计数器加一个偏移量 (ADDWF PCL) 来实现的。当使用相对 GOTO 指令方法进行读表操作时, 要注意表地址是否超过了 PCL 的存储边界 (每块 256 个字节)。请参见应用笔记 AN556 “Implementing a Table Read” (DS00556)。

### 4.3.2 堆栈

PIC16F627A/628A/648A 系列有一个 8 级深度, 13 位宽的硬件堆栈 (图 4-1)。该堆栈既不占用程序存储空间也不占用数据存储空间, 且栈指针不能读写。当执行 CALL 指令或中断引起程序转移时, PC 的值会被压入堆栈。而执行 RETURN、RETLW 或 RETFIE 指令时, PC 值会从堆栈弹出。压栈 (PUSH) 或出栈 (POP) 的操作不会影响 PCLATH。

堆栈是作为循环缓冲器使用的。也就是说, 压栈 8 次之后, 第 9 次压栈时进栈的数据将覆盖第 1 次压栈存储的数据。而第 10 次压栈时进栈的数据将覆盖第 2 次压栈存储的数据, 依此类推。

- 注**
- 1: 没有用于表示堆栈溢出或堆栈下溢情形的状态位。
  - 2: 没有称为 PUSH 或 POP 的指令或助记符。这些是执行 CALL、RETURN、RETLW 和 RETFIE 指令, 或转到中断向量地址时发生的操作。

## 4.4 间接寻址、INDF 和 FSR 寄存器

INDF 寄存器不是物理寄存器。对 INDF 寄存器寻址将导致间接寻址。

使用 INDF 寄存器可以实现间接寻址。任何使用 INDF 寄存器的指令实际上访问的是由指针寄存器 (FSR) 所指向的寄存器。间接读 INDF 本身会返回 00h。而使用间接寻址对 INDF 寄存器进行写操作将导致执行一个空操作 (虽然可能会影响状态位)。有效的 9 位地址是通过组合 8 位 FSR 寄存器和 IRP 位 (STATUS<7>) 获得的, 如图 4-5 所示。

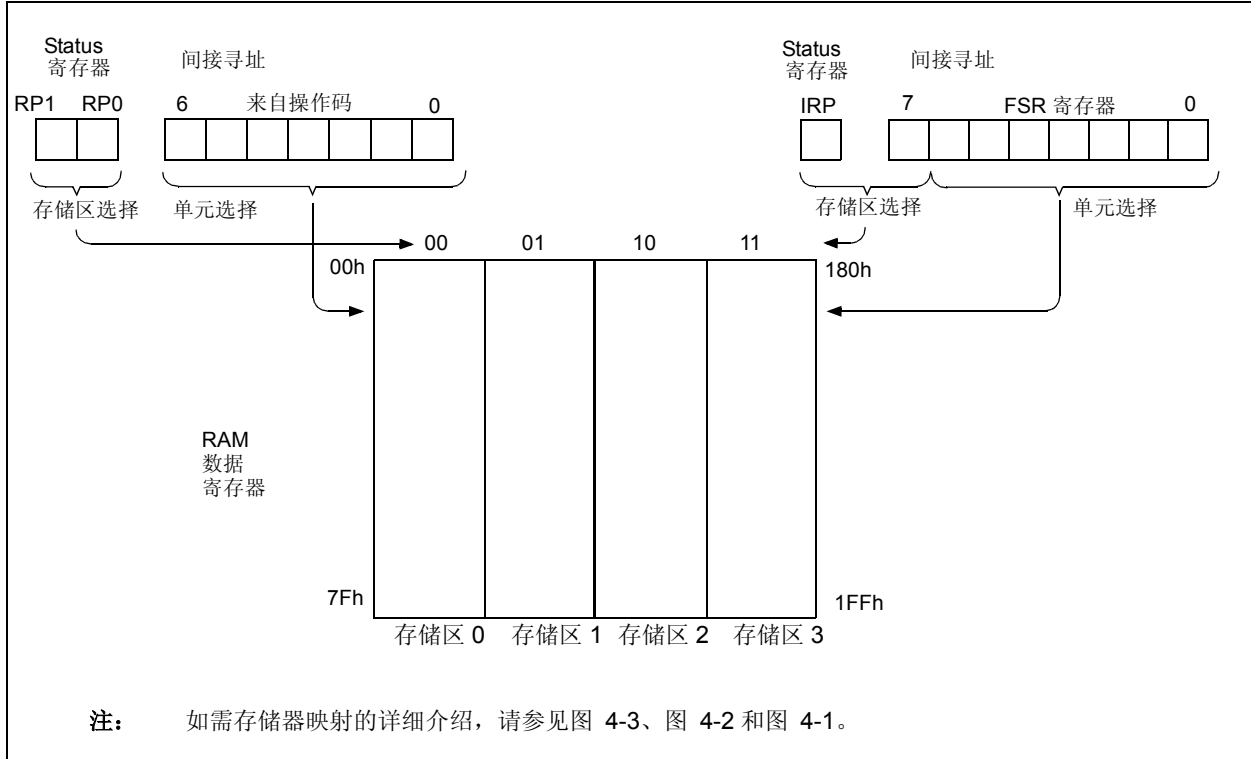
例 4-1 所示为一个使用间接寻址清零 RAM 单元 20h-2Fh 的简单程序。

例 4-1: 间接寻址

```
MOVLW 0x20 ;initialize pointer
MOVWF FSR ;to RAM
NEXT CLRF INDF ;clear INDF register
INCF FSR ;inc pointer
BTFSS FSR,4 ;all done?
GOTO NEXT ;no clear next
;yes continue
```

# PIC16F627A/628A/648A

图 4-5: PIC16F627A/628A/648A 的直接 / 间接寻址



# PIC16F627A/628A/648A

---

---

注:



## 5.0 I/O 口

PIC16F627A/628A/648A 有两个端口, PORTA 和 PORTB。这些 I/O 端口的部分引脚与器件外设功能部件的备用功能复用。通常, 当外设使能时, 其对应的引脚就可能无法用作一般的 I/O 引脚。

### 5.1 IPORTA 和 TRISA 寄存器

PORTA 是一个 8 位宽的锁存器。RA4 为施密特触发器输入和漏极开路输出引脚。它与 TOCKI 时钟输入复用。RA5<sup>(1)</sup> 只是施密特触发器输入, 没有输出驱动器。所有其他的 RA 端口引脚都有施密特触发器输入电平和完整的 CMOS 输出驱动器。所有引脚都有数据方向控制位 (在 TRIS 寄存器中), 这些位可以将引脚配置为输入或输出。

将 TRISA 寄存器的某位置 1, 就可以将相应的输出驱动器置为高阻态模式。将 TRISA 寄存器的某位清零, 则将输出锁存器的内容锁存到指定的引脚。

读 PORTA 寄存器是读取引脚上的电平状态, 而写 PORTA 寄存器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, 对一个端口进行写操作就意味着总是先读取端口引脚电平, 然后修改这个值, 最后将该值写入端口数据锁存器。

PORTA 引脚与比较器和参考电压功能复用。这些引脚的工作状态是通过 CMCON (比较器控制寄存器) 寄存器和 VRCON 寄存器 (参考电压控制寄存器) 中的控制位选择的。当选择它们作为比较器输入时, 这些引脚将读作 0。

**注 1:** RA5 与 VPP 复用。当在 RA5 上加 VPP 电压时, 器件将进入编程模式。

**注 2:** 在复位时, TRISA 寄存器各位全部被置为输入。禁止数字输入 (RA<3:0>) 并强制比较器输入接地以降低电流消耗。

**注 3:** 振荡器配置会改写 TRISA<6:7>。当改写 PORTA<6:7> 时, 数据读为 0, 并且忽略 TRISA<6:7> 位。

TRISA 控制 RA 引脚的输入 / 输出方向, 即使引脚作为比较器输入时也是如此。当将这些引脚作为比较器输入时, 用户必须确保将它们配置为输入引脚。

RA2 引脚也能作为参考电压的输出。当处于此模式时, VREF 引脚是极高阻抗的输出。用户必须配置 TRISA<2> 位为输入并使用高阻抗负载。

在 CMCON 寄存器定义的其中一种比较器模式下, RA3 和 RA4 引脚作为比较器的输出。为了使输出能使用此功能, 必须将 TRISA<4:3> 位清零。

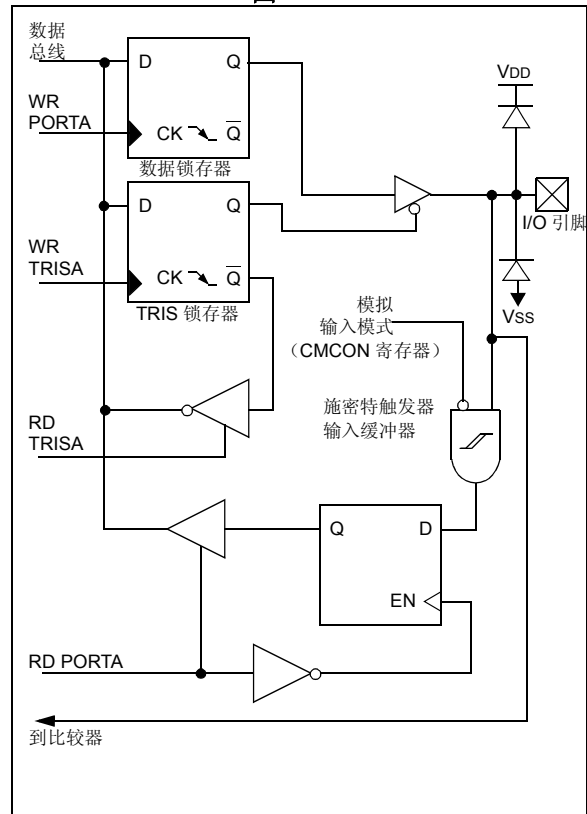
#### 例 5-1: 初始化 PORTA

```

CLRf   PORTA      ;Initialize PORTA by
                ;setting
                ;output data latches
MOVLW  0x07      ;Turn comparators off and
MOVWF  CMCON      ;enable pins for I/O
                ;functions

BCF    STATUS, RP1
BSF    STATUS, RP0;Select Bank1
MOVLW  0x1F      ;Value used to initialize
                ;data direction
MOVWF  TRISA      ;Set RA<4:0> as inputs
                ;TRISA<5> always
                ;read as '1'.
                ;TRISA<7:6>
                ;depend on oscillator
                ;mode
    
```

图 5-1: RA0/AN0:RA1/AN1 引脚框图



# PIC16F627A/628A/648A

图 5-2: RA2/AN2/VREF 引脚框图

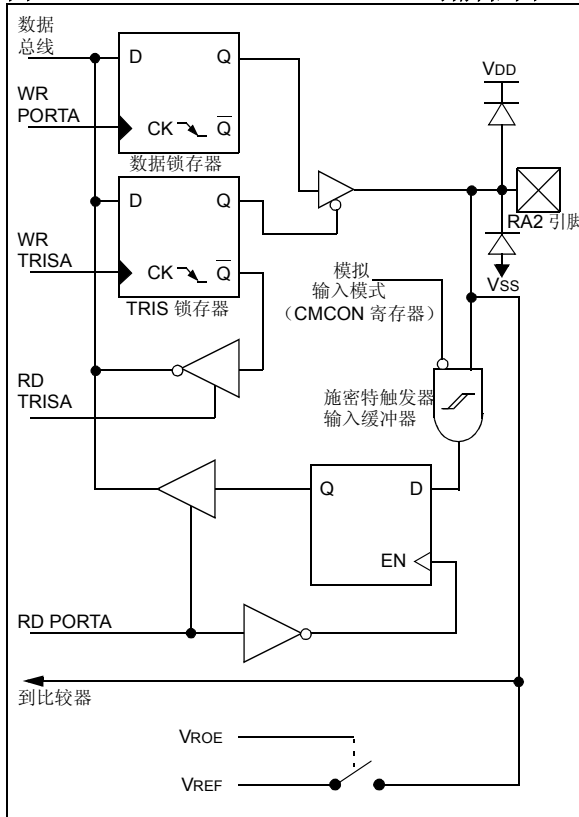


图 5-3: RA3/AN3/CMP1 引脚框图

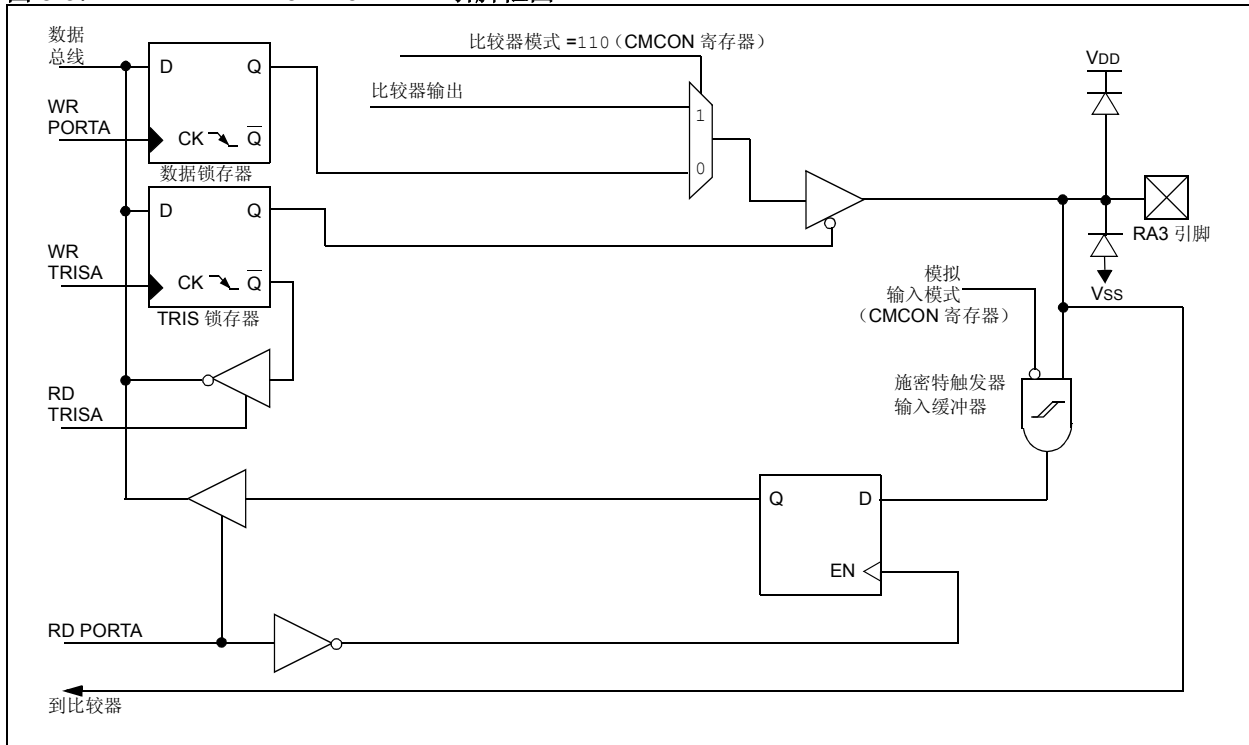


图 5-4: RA4/T0CKI/CMP2 引脚框图

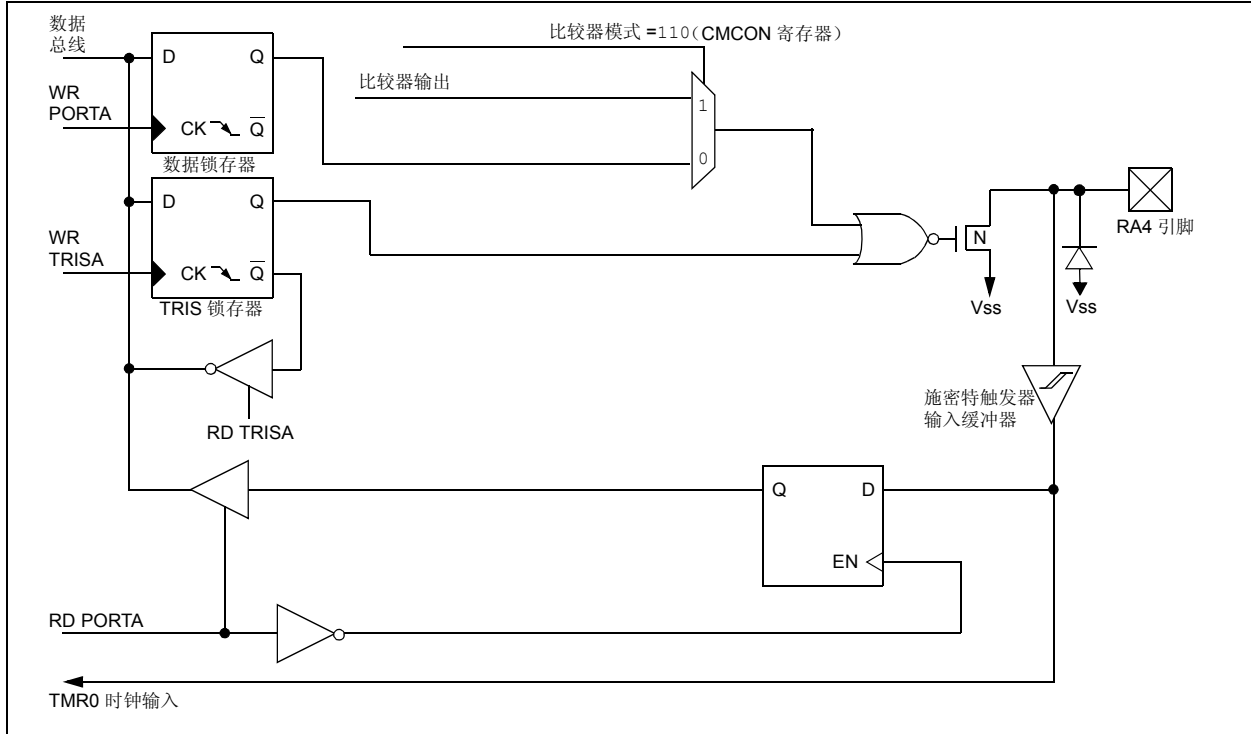


图 5-5: RA5/MCLR/VPP 引脚框图

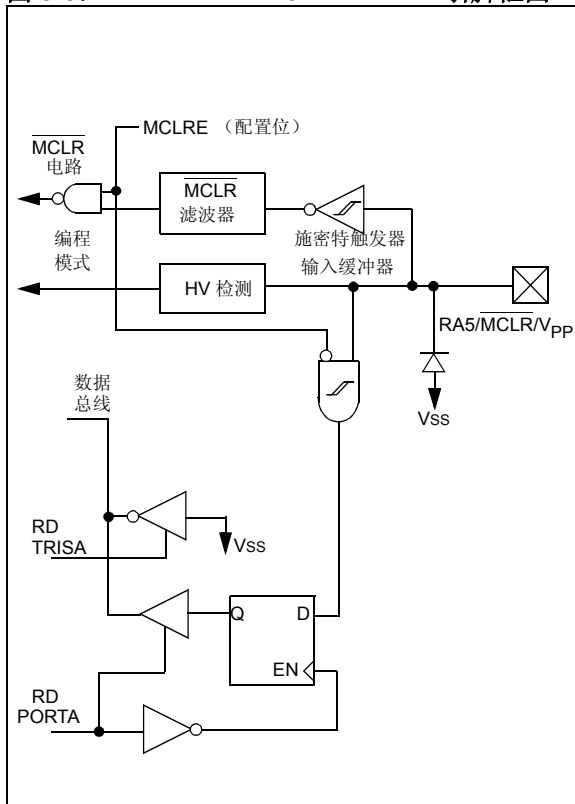
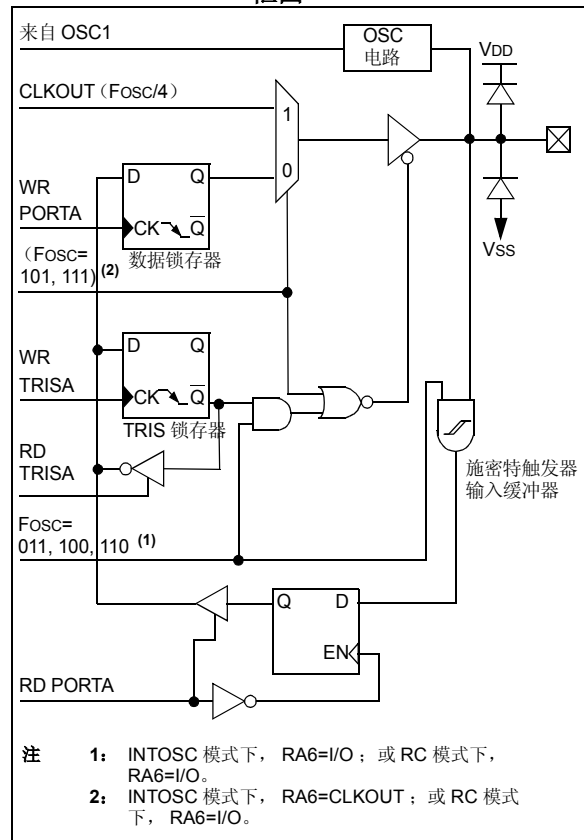
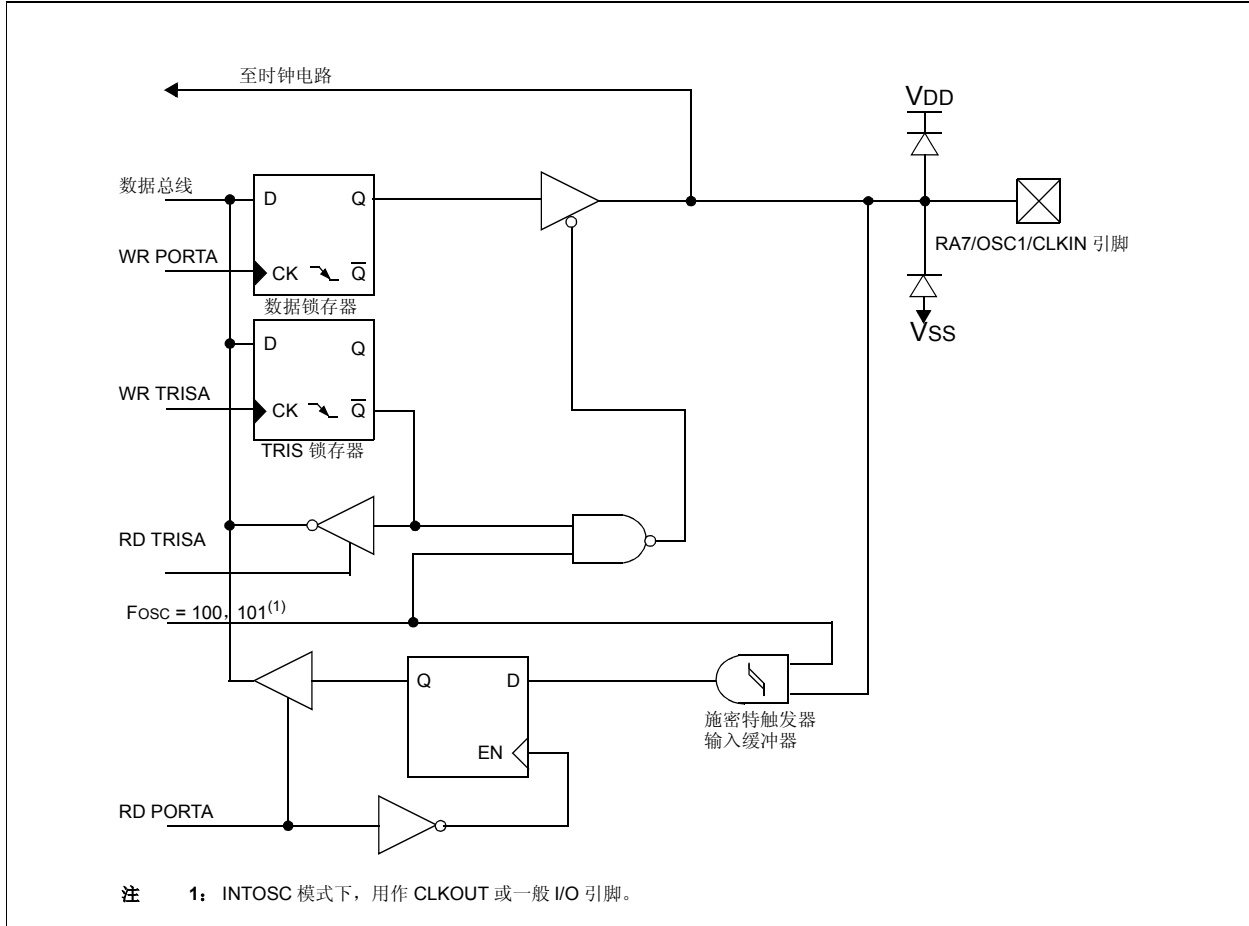


图 5-6: RA6/OSC2/CLKOUT 引脚框图



# PIC16F627A/628A/648A

图 5-7: RA7/OSC1/CLKIN 引脚框图





# PIC16F627A/628A/648A

## 5.2 PORTB 和 TRISB 寄存器

PORTB 是一个 8 位宽的双向端口。相应的数据方向寄存器是 TRISB。将 TRISB 寄存器的某位置 1，可以使相应的输出驱动器为高阻态。而将 TRISB 寄存器的某位清零，则将输出锁存器的内容锁存到选定的引脚上。

PORTB 与外部中断、USART、CCP 模块和 TMR1 时钟输入 / 输出复用。其标准端口功能和备用端口功能如表 5-3 所示。使能时，备用端口功能可能会使 TRIS 设置无效。

读 PORTB 寄存器可读取引脚的电平状态，而写 PORTB 寄存器是将数据写入端口锁存器。所有写操作都是读一修改一写操作。因此，对一个端口进行写操作就意味着总是先读取端口引脚电平，再修改这个值，然后将该值写入端口数据锁存器。

每个 PORTB 引脚都有一个内部弱上拉电路（通常约为 200 $\mu$ A）。单个控制位可以接通所有的上拉电路。这是通过将 RBPU（OPTION<7>）位清零实现的。当将 PORTB 端口的引脚配置为输出时，其弱上拉电路会自动切断。在上电复位后，弱上拉会被禁止。

PORTB 引脚中有 4 个（RB<7:4>）具有电平变化触发中断的功能。只有当引脚配置为输入时才会发生此中断（即，当 RB<7:4> 的任何一个引脚被配置为输出时，则该引脚就不会具有电平变化触发中断的功能）。当前 RB7:RB4 输入引脚上的电平与上次读 PORTB 时锁存的旧值进行比较。将 RB7:RB4 上与旧值相比的“失配”输出进行或运算，以产生 RBIF 中断（在 INTCON<0> 中锁存该中断标志）。

该中断可将器件从休眠中唤醒。在中断服务程序中，用户可以通过下面的方式清除中断：

- 对 PORTB 进行读 / 写操作。这将结束失配状态。
- 将 RBIF 标志位清零。

失配状态会不断将 RBIF 标志位置位。而读 PORTB 将结束失配状态，并且允许将 RBIF 标志位清零。

失配触发中断这个功能和这四个引脚上的可软件配置的上拉功能，通过这些功能可以很容易地与键盘接口，从而实现按键唤醒功能（参见应用笔记 AN552 “Implementing Wake-up on Key Strokes” (DS00552)）。

**注：** 如果 I/O 引脚的电平在执行读取操作时（Q2 周期的开始）发生改变，则 RBIF 中断标志可能不会置位。

对于按键唤醒以及其他仅使用 PORTB 的电平变化触发中断功能的操作，建议使用此电平变化触发中断功能来实现。在使用电平变化触发中断功能时，建议不要查询 PORTB 的状态。

图 5-8: RB0/INT 引脚框图

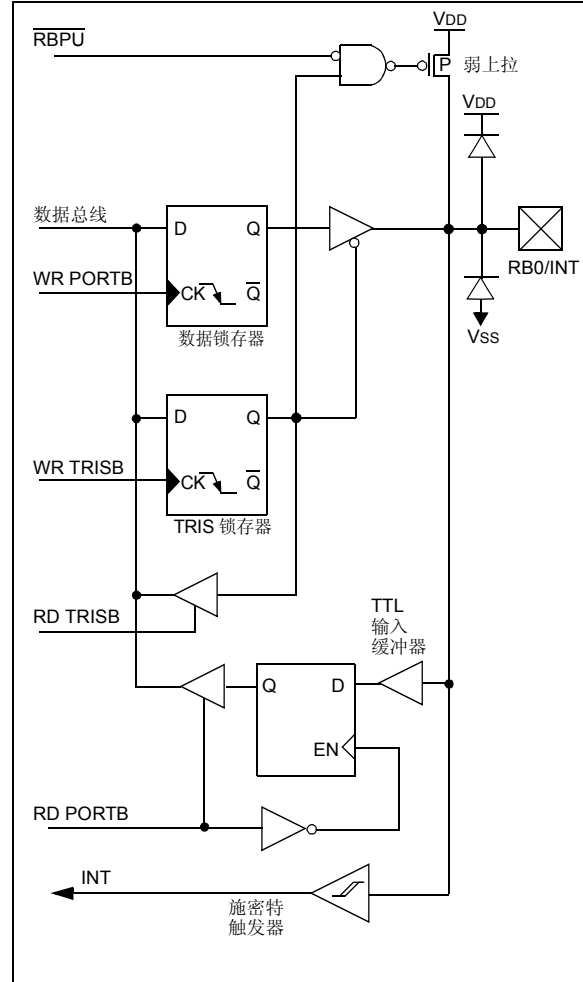


图 5-9: RB1/RX/DT 引脚框图

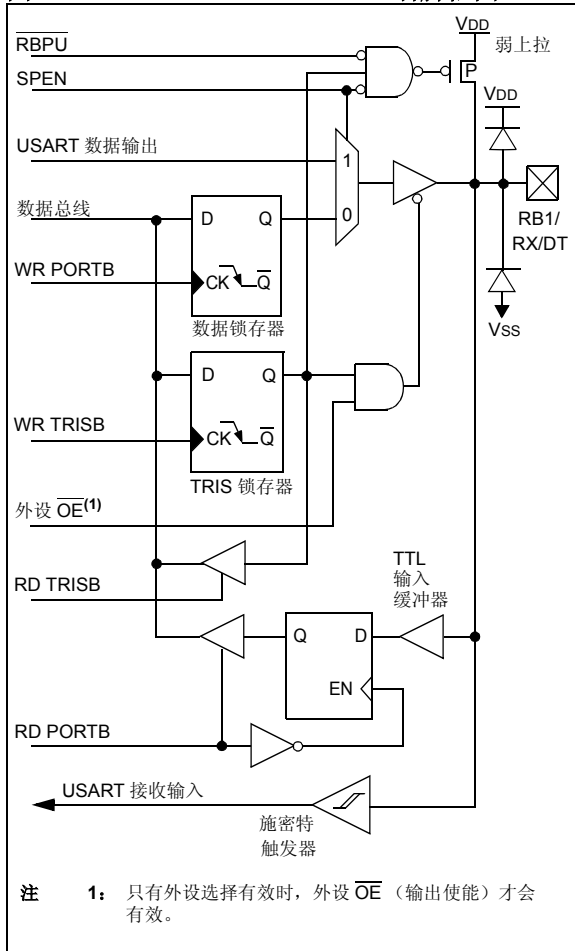
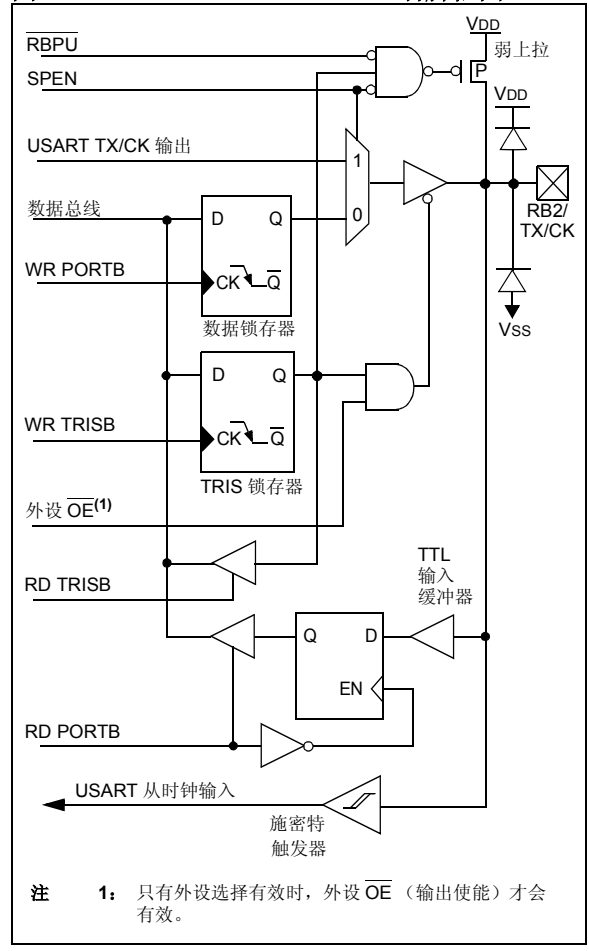


图 5-10: RB2/TX/CK 引脚框图



# PIC16F627A/628A/648A

图 5-11: RB3/CCP1 引脚框图

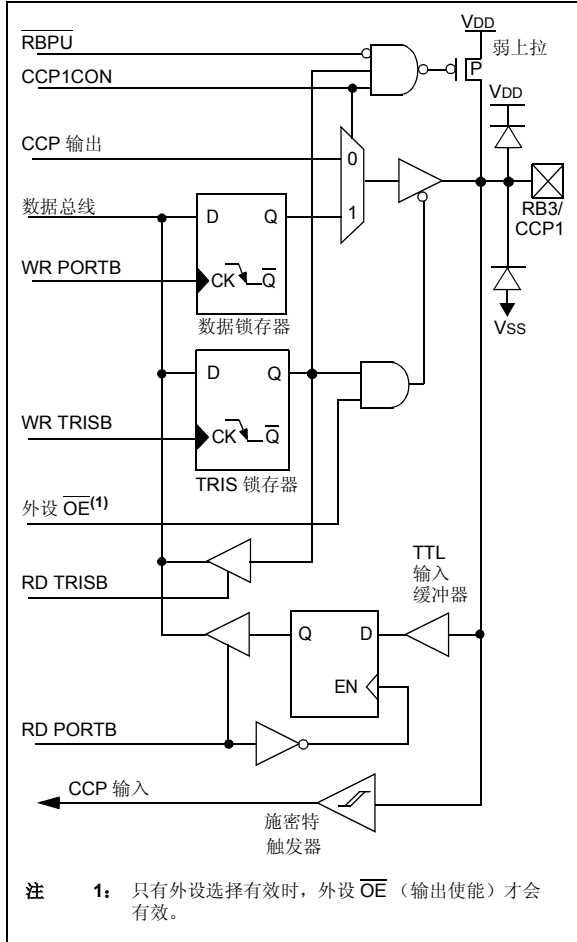
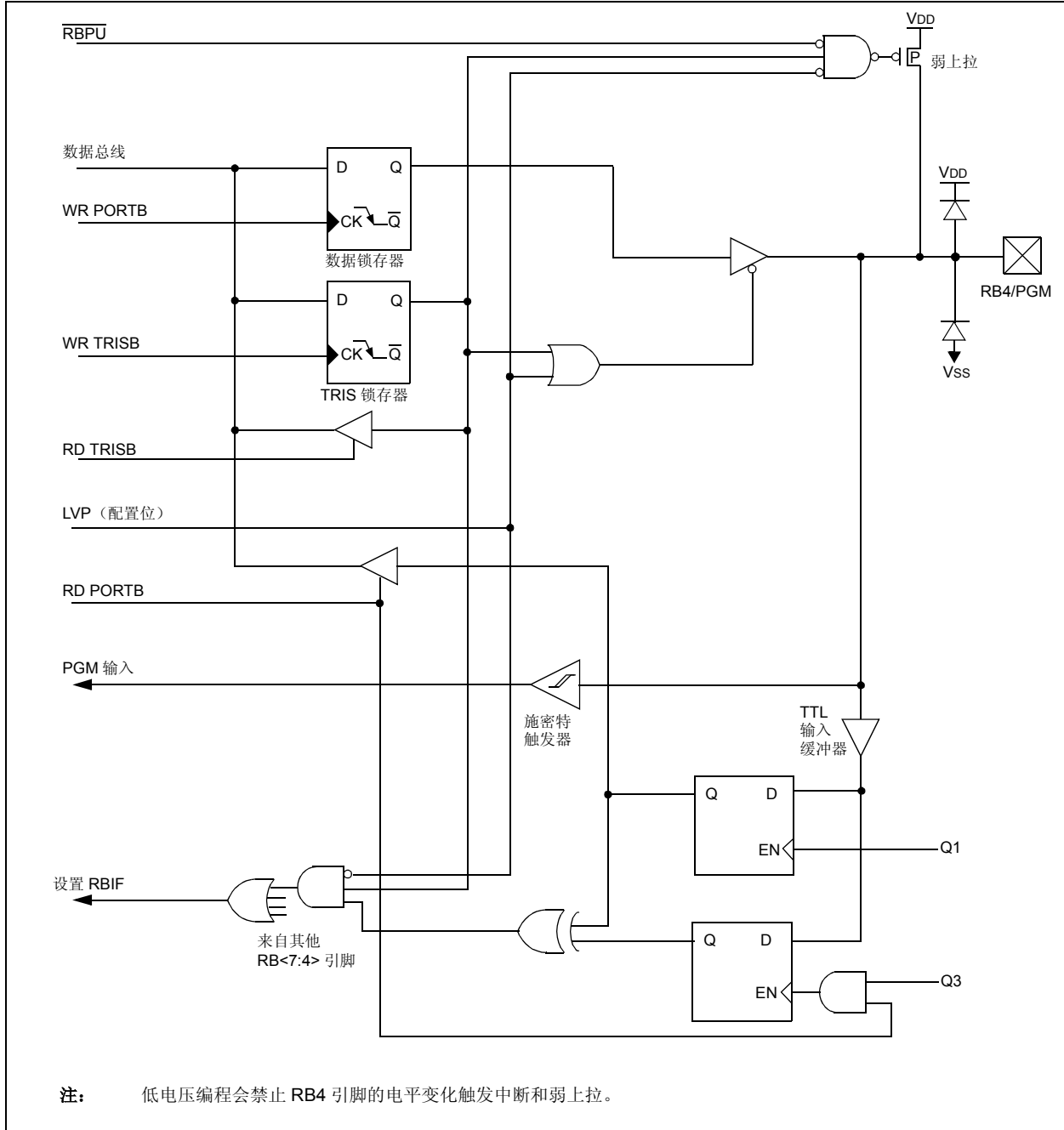




图 5-12: RB4/PGM 引脚框图



# PIC16F627A/628A/648A

图 5-13: RB5 引脚框图

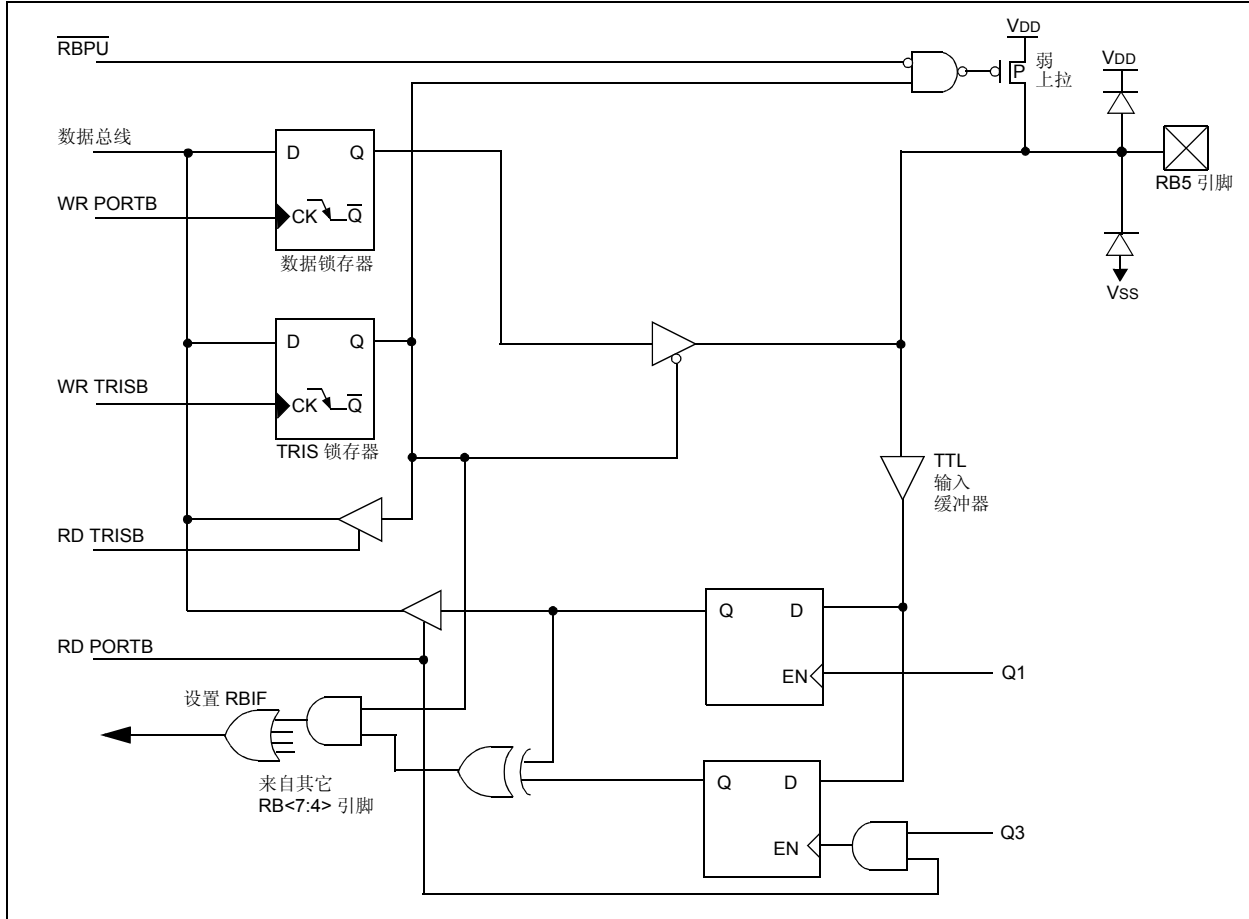
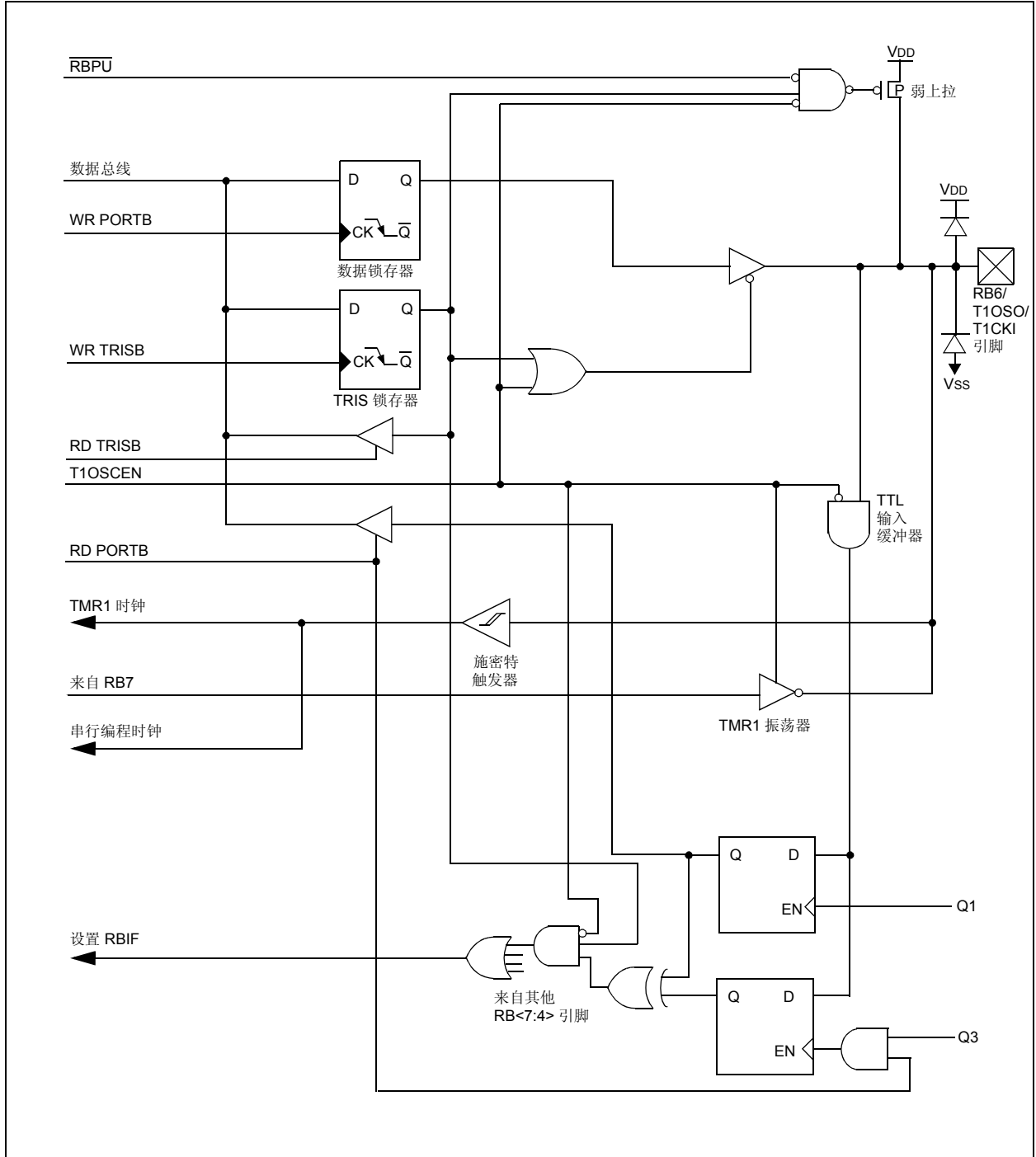
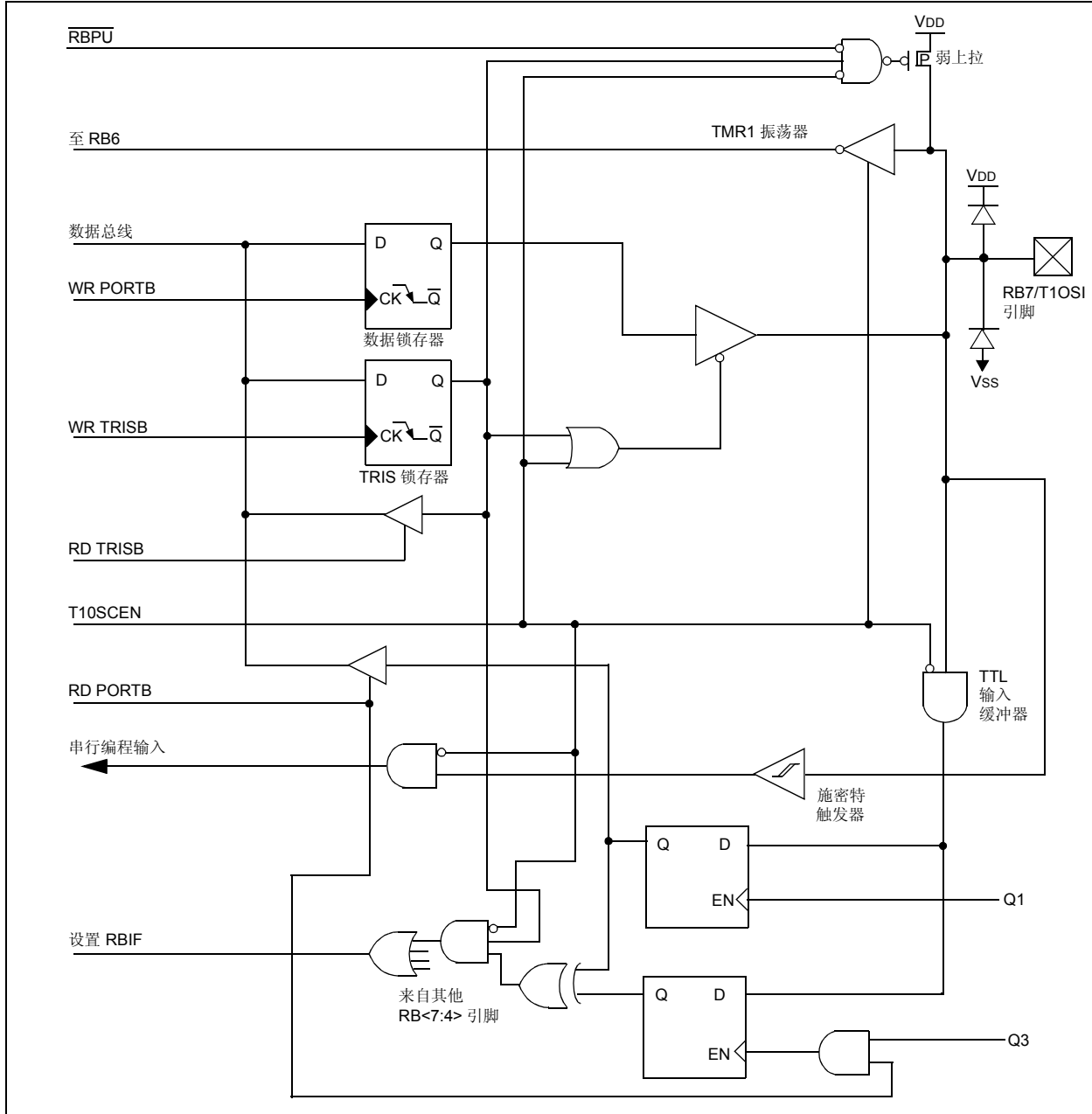


图 5-14: RB6/T1OSO/T1CKI 引脚框图



# PIC16F627A/628A/648A

图 5-15: RB7/T10SI 引脚框图





# PIC16F627A/628A/648A

## 5.3 I/O 编程注意事项

### 5.3.1 双向 I/O 端口

任何写操作指令，其内部操作都是读操作后跟一个写操作。例如，BCF和BSF指令先将寄存器的值读入CPU，然后执行位操作，最后将结果写回寄存器。当某个端口定义为既能输入又能输出时，对这些端口执行这些指令时必须小心。例如，对PORTB的bit 5执行BSF操作将使PORTB的所有8个位都被读入CPU。然后在bit 5上进行BSF操作，并且PORTB被写入输出锁存器。如果PORTB的另一个位用作双向I/O引脚（例如bit 0），并且此时被定义为输入，则该引脚上的输入信号本身将被读入CPU，并将该值重写入该特定引脚的数据锁存器，覆盖了先前的内容。只要引脚一直是输入模式，则不会有问题。但是，如果bit 0随后切换为输出模式，则数据锁存器的内容将是未知的。

读端口寄存器将读取端口引脚的值。写端口寄存器将把值写入端口锁存器。当对某个端口使用读-修改-写指令（例如BCF、BSF等）时，首先会读取该端口引脚的值，然后对读入值执行所需的操作，最后将操作结果写入端口锁存器。

例 5-2 给出了对 I/O 端口连续执行两个读-修改-写指令（例如 BCF、BSF 等）的结果。

当某引脚正在输出低电平或高电平信号时，不应再同时由外部器件驱动该引脚改变该引脚上的电平（“线与”或“线与”）。产生的高输出电流可能会损坏芯片。

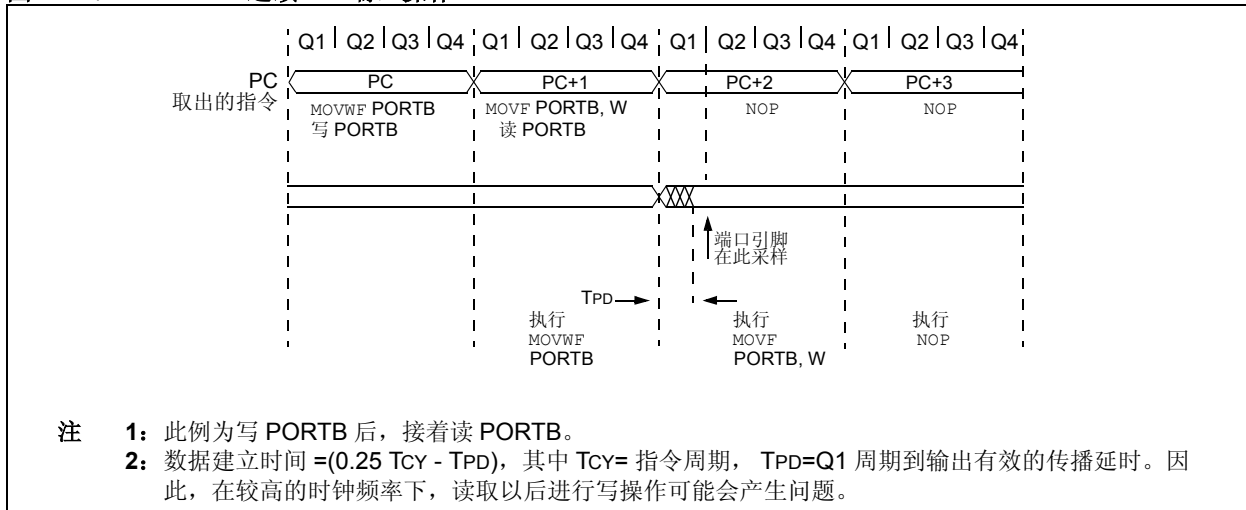
### 例 5-2: 对 I/O 端口执行读-修改-写指令

```
;Initial PORT settings:PORTB<7:4> Inputs
;                               PORTB<3:0> Outputs
;PORTB<7:6> have external pull-up and are
;not connected to other circuitry
;
;                               PORT latchPORT Pins
;                               -----
BCF STATUS, RP0 ;
BCF PORTB, 7 ;01pp pppp 11pp pppp
BSF STATUS, RP0 ;
BCF TRISB, 7 ;10pp pppp 11pp pppp
BCF TRISB, 6 ;10pp pppp 10pp pppp
;
;Note that the user may have expected the
;pin values to be 00pp pppp. The 2nd BCF
;caused RB7 to be latched as the pin value
;(High).
```

### 5.3.2 对 I/O 端口的连续操作

对 I/O 端口的写操作实际发生在指令周期的末尾；但对于读操作，数据必须在指令周期的开始时就有效（图 5-16）。因此，如果进行写操作之后，接着对同一个 I/O 端口执行读操作，就要特别注意。指令的执行顺序应该允许在下一条导致数据写入 CPU 的指令执行之前，引脚电压达到稳定（与负载有关）。否则，读入 CPU 的可能是引脚的前一个状态而非新状态。当状态不确定时，最好用一个 NOP 指令或其他不访问该 I/O 端口的指令隔开这些指令。

图 5-16: 连续 I/O 端口操作



## 6.0 TIMER0 模块

Timer0 模块定时器 / 计数器具有如下特点：

- 8 位定时器 / 计数器
- 读 / 写能力
- 8 位可软件编程的预分频器
- 内部或外部时钟选择
- FFh 到 00h 的溢出中断
- 外部时钟边沿选择

图 6-1 是 Timer0 模块的简化框图。更多信息，可查阅《PICmicro<sup>®</sup> 中档单片机系列参考手册》(DS33023A\_CN)。

可通过将 T0CS 位 (OPTION<5>) 清零选择定时器模式。在定时器模式下，TMR0 寄存器的值在每个指令周期递增 1 (不使用预分频器)。如果对 TMR0 寄存器执行写操作，那么在接下来的两个指令周期将禁止对其进行递增计数。用户可通过将校正值写入 TMR0 寄存器以避开这一问题。

可通过将 T0CS 位置 1 选择计数器模式。在该模式下，TMR0 寄存器的值将在 RA4/T0CKI/CMP2 引脚的每个上升沿或下降沿递增计数。具体的递增沿由时钟源边沿 (T0SE) 控制位 (OPTION<4>) 决定。将 T0SE 位清零选择上升沿。第 6.2 节“Timer0 用于外部时钟计数”中详细讨论了外部时钟输入的限制性。

预分频器由 Timer0 模块和看门狗共用。预分频器的分配可通过软件设置 PSA 控制位 (OPTION<3>) 来控制。清零 PSA 位可将预分频器分配给 Timer0。预分频器是不可读写的。当将预分频器分配给 Timer0 模块时，可供选择的预分频比有 1:2, 1:4, ……，1:256。预分频器的具体操作参见第 6.3 节“Timer0 预分频器”。

### 6.1 Timer0 中断

当 TMR0 寄存器发生从 FFh 到 00h 的溢出时，即产生 Timer0 中断。该溢出将 TOIF 位置 1。中断可以通过清零 TOIE 位 (INTCON<5>) 来屏蔽。在重新允许中断前，必须在 Timer0 模块的中断服务程序中用软件方法将 TOIF 位 (INTCON<2>) 清零。休眠状态时，由于定时器被关闭，所以 Timer0 中断无法唤醒单片机。

## 6.2 Timer0 用于外部时钟计数

当 Timer0 用于外部时钟输入的计数时，必须满足一定要求。这是由于内部相位时钟 (Tosc) 同步引起的。同样，在同步之后 Timer0 仍然要经过一段延时才会真正执行递增操作。

### 6.2.1 外部时钟同步

当不使用预分频器时，外部时钟输入与预分频器输出相同。在内部相位时钟的 Q2 和 Q4 周期对预分频器输出进行采样可实现 T0CKI 与内部相位时钟的同步 (图 6-1)。因此，要求 T0CKI 引脚上的信号高、低电平都至少保持 2Tosc (加上一小段 20ns 的 RC 延时)。具体请参考所用器件的电气规范。

当使用预分频器时，外部时钟输入要先经过异步计数型预分频器分频，而使预分频器的输出对称。为了使外部时钟满足采样要求，必须考虑计数器。因此，要求 T0CKI 引脚上信号的周期至少为 4Tosc (加上一小段 40 ns 的 RC 延时) 供预分频器分频。T0CKI 引脚的高低电平只须满足最小脉宽为 10 ns 的要求。具体请参考所用器件电气规范中的参数 40、41 和 42。参见表 17-8。

# PIC16F627A/628A/648A

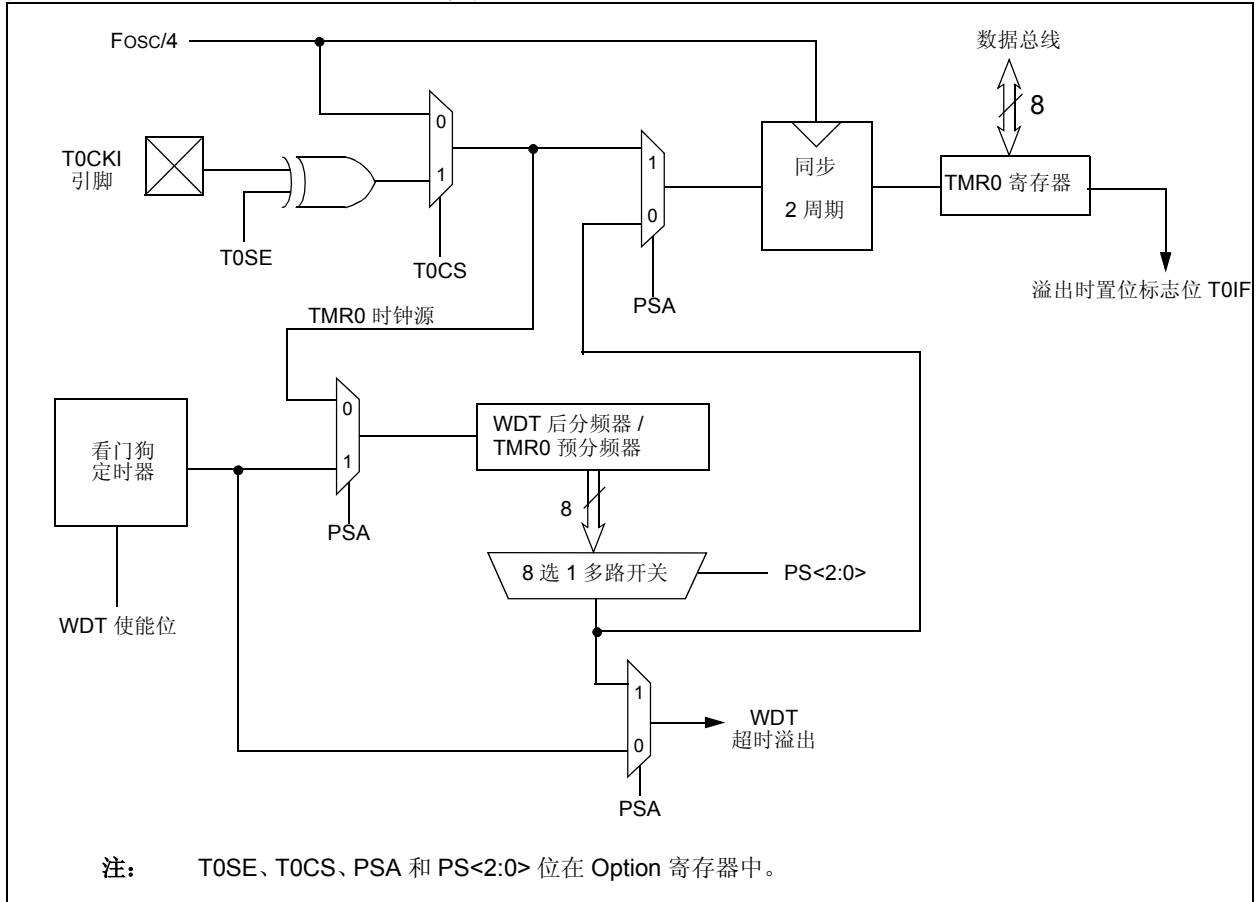
## 6.3 Timer0 预分频器

提供了一个 8 位的计数器作为 Timer0 模块的预分频器，或作为看门狗的后分频器。如果把预分频器分配给 Timer0 模块就意味着看门狗无后分频器可用，反之亦然。

PSA 和 PS<2:0> 位 (OPTION<3:0>) 决定预分频器的分配和分频比。

当将预分频器分配给 Timer0 模块时，所有对 TMR0 寄存器执行的写操作指令 (如 CLRF 1、MOVWF 1 和 BSF 1, x 等) 都将清零预分频器。当将预分频器分配给 WDT 时，执行 CLRWDT 指令将同时清零预分频器和看门狗。预分频器是不可读写的。

图 6-1: TIMER0/WDT 的框图





## 6.3.1 切换预分频器的分配

预分频器的分配完全由软件控制（比如，它可在程序执行期间“随时”被改变）。当把预分频器从 Timer0 分配给 WDT 时，必须执行例 6-1 中所示的指令序列，以避免器件意外复位。

**例 6-1: 改变预分频器 (TIMER0 → WDT)**

```
BCF STATUS, RP0 ;Skip if already in
                    ;Bank 0
CLRWDWDT          ;Clear WDT
CLRFB TMR0        ;Clear TMR0 and
                    ;Prescaler
BSF STATUS, RP0 ;Bank 1
MOVLW '00101111'b ;These 3 lines
                    ;(5, 6, 7)
MOVWF OPTION_REG ;are required only
                    ;if desired PS<2:0>
                    ;are
CLRWDWDT          ;000 or 001
MOVLW '00101xxx'b ;Set Postscaler to
MOVWF OPTION_REG ;desired WDT rate
BCF STATUS, RP0 ;Return to Bank 0
```

要将预分频器从 WDT 分配给 Timer0 模块，使用例 6-2 所示的指令序列。即使 WDT 被禁止也要采取该预防措施。

**例 6-2: 改变预分频器 (WDT → TIMER0)**

```
CLRWDWDT          ;Clear WDT and
                    ;prescaler
BSF STATUS, RP0
MOVLW b'xxxx0xxx' ;Select TMR0, new
                    ;prescale value and
                    ;clock source
MOVWF OPTION_REG
BCF STATUS, RP0
```

**表 6-1: 与 TIMER0 相关的寄存器**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
01h、101h	TMR0	Timer0 模块寄存器								xxxx xxxx	uuuu uuuu
0Bh、8Bh、10Bh、18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
81h、181h	OPTION <sup>(2)</sup>	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111

图注: - = 未实现的地址单元，读作 0，u = 未改变，x = 未知。阴影单元表示在 Timer0 模块中未使用。

注 1: Option 寄存器在 MPLAB<sup>®</sup> IDE 中用 OPTION\_REG 表示。

# PIC16F627A/628A/648A

## 7.0 TIMER1 模块

Timer1 模块是由两个可读写 8 位寄存器（TMR1H 和 TMR1L）组成的 16 位定时器 / 计数器。TMR1 寄存器对（TMR1H:TMR1L）从 0000h 开始递增计数，一直递增到 FFFFh，然后从 0000h 重新开始递增计数。如果允许 Timer1 中断，那么当用于锁存中断标志位 TMR1IF（PIR1<0>）的 TMR1 寄存器对发生溢出时就会产生 Timer1 中断。可以通过对 Timer1 中断允许位 TMR1IE（PIE1<0>）置位 / 清零来允许 / 禁止该中断。

Timer1 有下列两种工作模式：

- 定时器模式
- 计数器模式

工作模式由时钟选择位 TMR1CS（T1CON<1>）决定。

在定时器模式下，TMR1 寄存器对的值在每个指令周期递增计数。而在计数器模式下，它在外部时钟输入引脚的每个上升沿递增计数。

可以通过对控制位 TMR1ON（T1CON<0>）置位 / 清零来使能 / 禁止 Timer1。

Timer1 还有一个内部“复位输入”。这个复位可由 CCP 模块产生。（参见第 9.0 节“捕捉 / 比较 / PWM（CCP）模块”）寄存器 7-1 中是 Timer1 控制寄存器。

对于 PIC16F627A/628A/648A 而言，当使能 Timer1 的振荡器（T1OSCEN 位置 1）时，RB7/T1OSI/PGD 和 RB6/T1OSO/T1CKI/PGC 引脚就变为输入引脚。即

寄存器 7-1:

**T1CON — TIMER1 控制寄存器（地址：10h）**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	
bit 7								bit 0

bit 7-6 **未实现：** 读作 0

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：** Timer1 振荡器使能控制位

1= 使能振荡器  
0= 关闭振荡器<sup>(1)</sup>

bit 2 **T1SYNC：** Timer1 外部时钟输入同步控制位

**TMR1CS=1**

1= 不同步外部时钟输入  
0= 同步外部时钟输入

**TMR1CS=0**

此位被忽略。TMR1CS=0 时，Timer1 使用内部时钟。

bit 1 **TMR1CS：** Timer1 时钟源选择位

1= 选择来自 RB6/T1OSO/T1CKI/PGC 引脚的外部时钟（上升沿有效）  
0= 选择内部时钟（Fosc/4）

bit 0 **TMR1ON：** Timer1 开启位

1= 使能 Timer1  
0= 停止 Timer1

**注 1：** 关断振荡器的反相器和反馈电阻以降低功耗。

**图注：**

R= 可读位

W= 可写位

U= 未实现位，读作 0

-n=POR 时的值

1= 置位

0= 清零

x= 未知

## 7.1 Timer1 工作在定时器模式

通过将 TMR1CS (T1CON<1>) 位清零选择定时器模式。在这种模式下, 定时器的输入时钟是  $F_{osc}/4$ 。因为内部时钟总是同步的, 所以同步控制位 T1SYNC (T1CON<2>) 此时不起作用。

## 7.2 Timer1 工作在同步计数器模式

通过将 TMR1CS 位置 1 选择计数器模式。在这种模式下, TMR1 寄存器对的值在引脚 RB7/T1OSI/PGD (T1OSCN 位置 1 时) 或 RB6/T1OSO/T1CKI/PGC (T1OSCN 位清零时) 上的每个输入时钟上升沿到来时递增 1。

如果 T1SYNC 位清零, 那么外部时钟输入与内部相位时钟同步。同步是在预分频之后完成的。预分频器是一个异步计数器。

在该模式配置下, 由于在休眠模式下关断同步电路, 所以即使使用外部时钟, TMR1 寄存器对的值也不能递增计数。但是预分频器会继续递增计数。

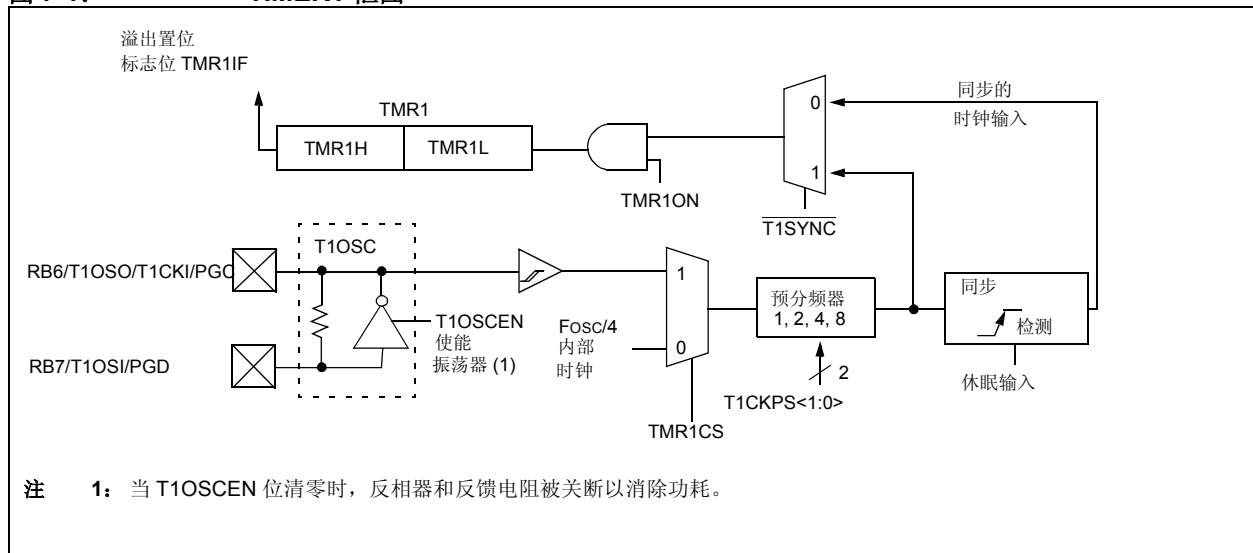
## 7.2.1 同步计数器模式下的外部时钟输入

当外部时钟输入用于同步计数器模式下的 Timer1 时, 必须满足一定要求。这是由于内部相位时钟 (Tosc) 同步引起的。而且在同步之后, TMR1 寄存器对的值仍然要经过一段延时才会进行递增 1 操作。

当预分频比为 1:1 时, 外部时钟输入和预分频器的输出相同。在内部相位时钟的 Q2 和 Q4 周期, 对预分频器输出进行采样, 可实现 T1CKI 与内部相位时钟的同步。因此, 必须要求 T1CKI 引脚上信号的高、低电平都至少保持  $2T_{osc}$  (加上一小段 20ns 的 RC 延时)。具体请参考电气规范章节中的参数 45、46 和 47。

当使用的预分频比不是 1:1 时, 外部时钟输入要先经过异步计数型预分频器的分频, 从而使预分频器的输出对称。为了使外部时钟满足采样要求, 必须考虑计数器。因此, 必须要求 T1CKI 引脚上信号的周期至少为  $4T_{osc}$  (加上一小段 40ns 的 RC 延时) 供预分频器分频。T1CKI 引脚上信号的高低电平时间只需满足最小脉宽为 10ns 的要求。具体请参考适用电气规范中的参数 45、46 和 47。

图 7-1: TIMER1 框图



# PIC16F627A/628A/648A

## 7.3 Timer1 工作在异步计数器模式

如果控制位  $\overline{T1SYNC}$  ( $T1CON<2>$ ) 置 1，外部时钟输入就不同步。计数器继续进行与内部相位时钟异步的递增计数。在休眠状态下计数器将仍继续运行，并在溢出时产生中断，唤醒处理器。但在软件中应特别注意对计数器的读写（第 7.3.2 节“异步计数器模式下对 Timer1 的读写操作”）。

**注：** 在异步计数器模式下，Timer1 不能用作捕捉器或比较器工作的时基。

### 7.3.1 异步时钟的外部时钟输入时序

如果控制位  $\overline{T1SYNC}$  被置 1，计数器将完全异步地进行递增计数。输入时钟必须满足高电平和低电平的最小脉宽要求。参阅电气规范一章中的表 17-8 以及时序参数 45、46 和 47。

### 7.3.2 异步计数器模式下对 TIMER1 的读写操作

当计数器工作在外部异步时钟下时，读 TMR1H 或 TMR1L 寄存器能够产生有效的读取（由硬件完成）。但用户应牢记，用读两个 8 位值来读一个 16 位定时器会出现问题，这是因为在两次读取操作之间计数器可能会溢出。

对于写操作，建议用户停止计数器后再写入需要的数值。而寄存器正在递增计数时，向定时器寄存器写入数据可能会产生写争用。这可能在定时器寄存器中产生无法预知的值。

读 16 位值时应非常注意。例 7-1 是读取 16 位定时器值的程序示例。在定时器无法停止时，该程序很有帮助。

### 例 7-1: 读取独立运行的 16 位定时器

```
; All interrupts are disabled
MOVWF TMR1H, W ;Read high byte
MOVWF TMPH ;
MOVWF TMR1L, W ;Read low byte
MOVWF TMPL ;
MOVWF TMR1H, W ;Read high byte
SUBWF TMPH, W ;Sub 1st read with
;2nd read
BTFSC STATUS,Z ;Is result = 0
GOTO CONTINUE ;Good 16-bit read
;
; TMR1L may have rolled over between the
; read of the high and low bytes. Reading
; the high and low bytes now will read a good
; value.
;
MOVWF TMR1H, W ;Read high byte
MOVWF TMPH ;
MOVWF TMR1L, W ;Read low byte
MOVWF TMPL ;
; Re-enable the Interrupts (if required)
CONTINUE ;Continue with your
;code
```

## 7.4 Timer1 振荡器

在 T1OSI（输入）和 T1OSO（放大器输出）引脚之间内接有晶振电路。通过将控制位 T1OSCEN（T1CON<3>）置 1 使能该电路。它在休眠状态下可以继续工作。一般倾向于使用 32.768 kHz 的时钟晶振。表 7-1 给出了 Timer1 振荡器的电容选择。

用户必须提供软件时间延迟来确保振荡器正确起振。

**表 7-1: TIMER1 振荡器的电容选择**

频率	C1	C2
32.768 kHz	15 pF	15 pF

**注：** 这些值仅供设计参考。  
想要获取更多有关晶振 / 电容选择的信息，请查阅 AN826 “Crystal Oscillator Basics and Crystal Selection for rfPIC® and PICmicro® Devices” (DS00826)。

## 7.5 用 CCP 触发器输出复位 Timer1

如果在比较模式下将 CCP1 模块配置为产生“特殊事件触发信号”（CCP1M<3:0> =1011），则该信号会复位 Timer1。

**注：** CCP1 模块的特殊事件触发信号不会将中断标志位 TMR1IF（PIR1<0>）置 1。

为了利用这一特点，Timer1 必须设置为定时器或同步计数器模式。如果 Timer1 在异步计数器模式下运行，该复位操作可能不起作用。

当对 Timer1 的写操作与 CCP1 的特殊事件触发信号同时发生时，写操作优先。

在这种工作方式下，寄存器对 CCPRxH:CCPRxL 实际上变成了 Timer1 的周期寄存器。

## 7.6 复位 Timer1 寄存器对（TMR1H, TMR1L）

除了 CCP1 特殊事件触发信号（参见第 9.2.4 节“特殊事件触发”）外，POR 或其他复位都不会将 TMR1H 和 TMR1L 复位至 00h。

T1CON 寄存器在上电复位或欠压复位时复位为 00h，这样就关闭了定时器并得到 1:1 的预分频比。在所有其他复位中，寄存器都不受影响。

## 7.7 Timer1 预分频器

在对 TMR1H 或 TMR1L 寄存器进行写操作时，预分频器计数器被清零。

**表 7-2: 与作为定时器 / 计数器的 TIMER1 有关的寄存器**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
0Bh、8Bh、10Bh、18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
0Eh	TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNCR	TMR1CS	TMR1ON	--00 0000	--uu uuuu

**图注：** x= 未知， u= 未改变， — = 未实现，读作 0。阴影部分表示在 Timer1 模块中未使用。



# PIC16F627A/628A/648A

**寄存器 8-1: T2CON — TIMER2 控制寄存器 (地址: 12h)**

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

- bit 7 **未实现位:** 读作 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  
 .  
 .  
 .  
 1111 = 1:16 后分频比
- bit 2 **TMR2ON:** Timer2 使能位  
 1 = 使能 Timer2  
 0 = 停止 Timer2
- bit 1-0 **T2CKPS<1:0>:** Timer2 时钟预分频比选择位  
 00 = 1:1 预分频比  
 01 = 1:4 预分频比  
 1x = 1:16 预分频比

<b>图注:</b>			
R= 可读位	W= 可写位	U= 未实现位, 读作 0	
-n=POR 时的值	1= 置位	0= 清零	x= 未知

**表 8-1: 与作为定时器 / 计数器的 TIMER2 有关的寄存器**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
0Bh、8Bh、10Bh、18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
11h	TMR2	Timer2 模块的寄存器								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
92h	PR2	Timer2 周期寄存器								1111 1111	1111 1111

**图注:** x = 未知, u = 未改变, — = 未实现, 读作 0。阴影部分表示在 Timer2 模块中未使用。

# PIC16F627A/628A/648A

---

---

注:



## 9.0 捕捉 / 比较 / PWM (CCP) 模块

CCP (捕捉 / 比较 / PWM) 模块包含一个 16 位寄存器, 它可以作为 16 位捕捉寄存器、16 位比较寄存器或作为 PWM 主 / 从占空比寄存器。表 9-1 所示为 CCP 模块模式的定时器资源。

### CCP1 模块

捕捉 / 比较 / PWM 寄存器 1 (CCPR1) 由两个 8 位寄存器组成, 这两个寄存器为: CCPR1L (低字节) 和 CCPR1H (高字节)。CCP1CON 寄存器控制 CCP1 的工作。所有位均可读写。

有关 CCP 模块的更多信息, 可参阅《PICmicro® 中档单片机系列参考手册》(DS33023A\_CN)。

表 9-1: CCP 模式 — 定时器资源

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

### 寄存器 9-1: CCP1CON — CCP 操作寄存器 (地址: 17h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	
bit 7								bit 0

bit 7-6 **未实现位:** 读作 0

bit 5-4 **CCP1X:CCP1Y:** PWM 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 占空比的两个 LSB。8 个 MSb 在 CCPRxL 中。

bit 3-0 **CCP1M<3:0>:** CCPx 模式选择位

0000= 捕捉 / 比较 / PWM 关闭 (复位 CCP1 模块)

0100= 捕捉模式, 每个下降沿发生

0101= 捕捉模式, 每个上升沿发生

0110= 捕捉模式, 每 4 个上升沿发生

0111= 捕捉模式, 每 16 个上升沿发生

1000= 比较模式, 相符时置位输出位 (置位 CCP1IF 位)

1001= 比较模式, 相符时清零输出位 (置位 CCP1IF 位)

1010= 比较模式, 相符时产生软件中断 (置位 CCP1IF 位, 而 CCP1 引脚不受影响)

1011= 比较模式, 触发特殊事件 (置位 CCP1IF 位; CCP1 复位 TMR1)

11xx= PWM 模式

#### 图注:

R= 可读位

W= 可写位

U= 未实现位, 读作 0

-n=POR 时的值

1= 置位

0= 清零

x= 未知

# PIC16F627A/628A/648A

## 9.1 捕捉模式

在捕捉模式下，RB3/CCP1 引脚上发生事件时，CCPR1H:CCPR1L 捕捉 TMR1 寄存器的 16 位值。事件定义为：

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

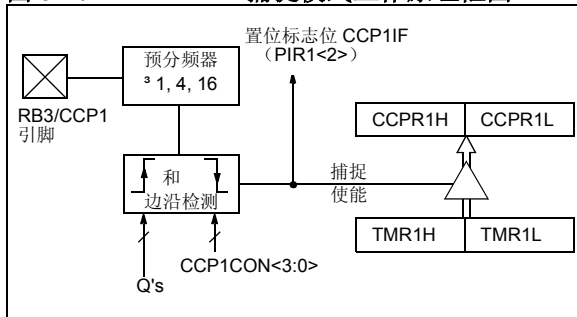
控制位 CCP1M<3:0> (CCP1CON<3:0>) 选择事件。当捕捉发生时，中断请求标志位 CCP1IF (PIR1<2>) 置位。此位必须用软件清零。如果在读取寄存器 CCPR1 中的值之前发生另一个捕捉，那么之前捕捉的数据将会丢失。

### 9.1.1 CCP 引脚配置

在捕捉模式下，应该通过置位 TRISB<3> 位将 RB3/CCP1 引脚配置为输入。

**注：** 如果 RB3/CCP1 引脚配置为输出，那么对该端口的写操作可能引发一个捕捉事件。

图 9-1: 捕捉模式工作原理框图



### 9.1.2 TIMER1 模式选择

欲使 CCP 模块使用捕捉功能，Timer1 必须工作在定时器或同步计数器模式。在异步计数器模式下，可能无法进行捕捉操作。

### 9.1.3 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 CCP1IE (PIE1<2>) 位为零以避免产生错误中断，而且应该在工作模式发生任何改变之后清零标志位 CCP1IF。

### 9.1.4 CCP 预分频器

通过设置 CCP1M<3:0> 可以选择四种预分频比设置。只要 CCP 模块被关闭或者 CCP 模块不工作在捕捉模式，就将预分频器的计数器清零。这就意味着任何复位都可以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捕捉预分频比可能产生中断。而且，也不会将预分频器的计数器清零，因此第一个捕捉可能是从一个非零的预分频比开始的。例 9-1 是切换捕捉预分频比时建议采用的方法。该例也使预分频器的计数器清零且不会产生“错误”中断。

例 9-1: 改变捕捉预分频器比

```
CLRF    CCP1CON    ;Turn CCP module off
MOVLW  NEW_CAPT_PS;Load the W reg with
                ; the new prescaler
                ; mode value and CCP ON
MOVWF  CCP1CON    ;Load CCP1CON with this
                ; value
```

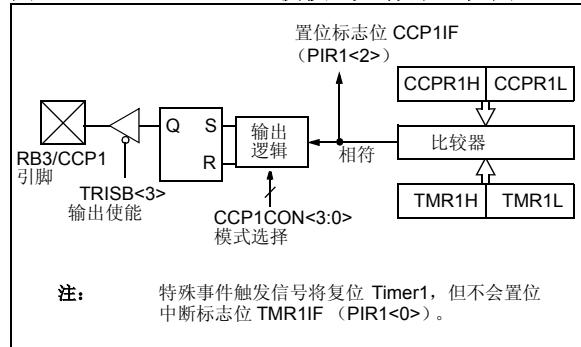
## 9.2 比较模式

在比较模式下，始终用 16 位 CCPR1 寄存器值与 TMR1 寄存器对的值相比较。当两者相符时，RB3/CCP1 引脚将：

- 输出高电平
- 输出低电平
- 保持不变

引脚上的操作取决于控制位 CCP1M<3:0> (CCP1CON<3:0>) 的值。同时，中断标志位 CCP1IF 置位。

图 9-2: 比较模式工作原理框图



**注：** 特殊事件触发信号将复位 Timer1，但不会置位中断标志位 TMR1IF (PIR1<0>)。

## 9.2.1 CCP 引脚配置

用户必须通过清零 TRISB<3> 位将 RB3/CCP1 引脚配置为输出。

**注：** 清零 CCP1CON 寄存器将强制 RB3/CCP1 比较输出锁存到缺省低电平。这不是数据锁存器。

## 9.2.2 TIMER1 模式选择

欲使 CCP 模块使用比较功能，Timer1 必须工作在定时器或同步计数器模式。在异步计数器模式下，可能无法进行比较操作。

## 9.2.3 软件中断模式

当选择了产生软件中断时，CCP1 引脚上的电平不受影响。只会产生一个 CCP 中断（允许该中断时）。

## 9.2.4 特殊事件触发

在这一模式下，将产生一个内部硬件触发信号，可用于触发一个操作。参见寄存器 9-1。

CCP1 的特殊事件触发输出引起 TMR1 寄存器对复位。这使 CCPR1 寄存器有效地成为 Timer1 的 16 位可编程周期寄存器。

当 TMR1H、TMR1L 寄存器对与 CCPR1H、CCPR1L 寄存器对相符时，CCP 立即产生特殊事件触发输出。直到 TMR1 时钟的下一个上升沿，TMR1H、TMR1L 寄存器对才被复位。这使 CCPR1 寄存器对实际上等同于 Timer1 的 16 位周期控制寄存器。如果使能了 A/D 模块，特殊事件触发输出还可以启动 A/D 转换。

**注：** 在产生特殊事件触发的时钟边沿和产生 TMR1 复位的时钟边沿之间，通过改变 CCPR1H、CCPR1L 寄存器对的内容来去除相符条件，将会阻止复位的发生。

**表 9-2: 与捕捉、比较和 TIMER1 相关的寄存器**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
86h, 186h	TRISB	PORTB 数据方向寄存器								1111 1111	1111 1111
0Eh	TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
15h	CCPR1L	捕捉 / 比较 / PWM 寄存器 1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	捕捉 / 比较 / PWM 寄存器 1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000

**图注：** x = 未知，u = 未改变，— = 未实现，读作 0。阴影部分表示未被捕捉和 Timer1 使用。

# PIC16F627A/628A/648A

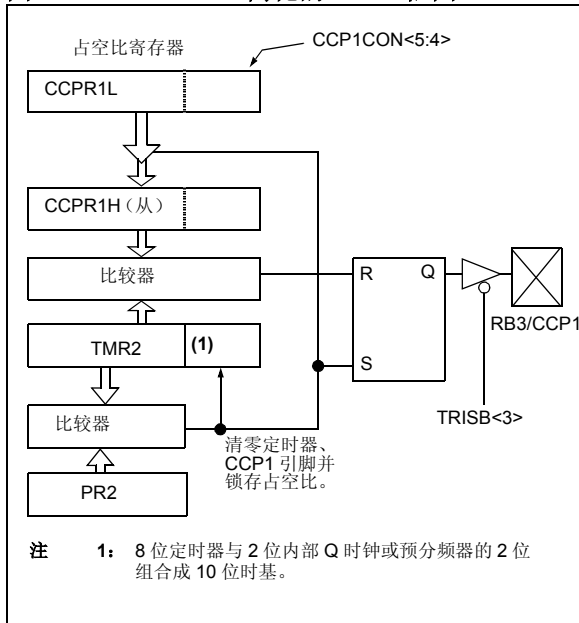
## 9.3 PWM 模式

在脉宽调制 (PWM) 模式下, CCP1 引脚产生的 PWM 输出可达 10 位分辨率。由于 CCP1 引脚与 PORTB 数据锁存器复用, 必须清零 TRISB<3> 位将 CCP1 引脚作为输出引脚。

**注:** 清零 CCP1CON 寄存器将使 CCP1 PWM 输出锁存到缺省低电平。这不是 PORTB I/O 数据锁存器。

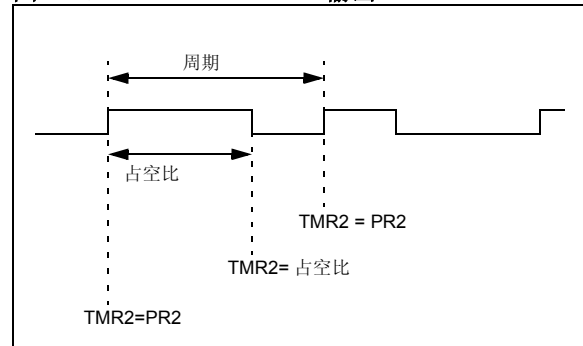
图 9-3 所示为 CCP 模块在 PWM 模式时的简化框图。如需循序渐进地了解如何为执行 PWM 操作而设置 CCP 模块, 请参阅第 9.3.3 节 “PWM 操作的设置”。

图 9-3: 简化的 PWM 框图



一个 PWM 输出 (图 9-4) 包含一个时基 (周期) 和一段输出保持高电平的时间 (占空比)。PWM 频率是周期的倒数 (频率 = 1/周期)。

图 9-4: PWM 输出



### 9.3.1 PWM 周期

PWM 周期可以通过写 PR2 寄存器来指定。可以用以下公式计算:

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

PWM 频率定义为  $1 / [PWM \text{ 周期}]$ 。

当 TMR2 等于 PR2 时, 在下一递增计数周期中将产生下面三个事件:

- TMR2 清零
- CCP1 引脚置位 (例外: 如果 PWM 占空比为 0%, CCP1 引脚将不会被置位)
- PWM 占空比从 CCPR1L 锁存到 CCPR1H

**注:** Timer2 后分频器对于确定 PWM 频率不起作用 (参阅第 8.0 节 “Timer2 模块”)。可以利用后分频器来产生频率与 PWM 输出不同的特定波形。

## 9.3.2 PWM 占空比

PWM 占空比可通过写 CCPR1L 寄存器和 CCP1CON<5:4> 位来指定。最高分辨率可达 10 位；其中高 8 位包含在 CCPR1L 中，而低 2 位是 CCP1CON<5:4>。由 CCPR1L:CCP1CON<5:4> 表示这个 10 位值。计算占空比的公式如下：

$$PWM \text{ 占空比} = (CCPR1L:CCP1CON<5:4>) \cdot T_{osc} \cdot TMR2 \text{ 预分频值}$$

可以在任何时候写入 CCPR1L 和 CCP1CON<5:4>，但直到 PR2 和 TMR2 中的值相符（即当周期结束时），占空比的值才被锁存到 CCP1H。在 PWM 模式下，CCP1H 是只读寄存器。

CCP1H 寄存器和一个 2 位的内部锁存器用于双重缓冲 PWM 占空比。双重缓冲对于减少 PWM 的毛刺是极其重要的。

当 CCP1H 和 2 位锁存器的值与 TMR2 与内部 2 位 Q 时钟或 TMR2 预分频器的 2 位的组合相符时，CCP1 引脚被清零。

对于给定的 PWM 频率，其最大分辨率（位）为：

$$PWM \text{ 分辨率} = \frac{\log\left(\frac{F_{osc}}{F_{pwm} \times TMR2 \text{ 预分频值}}\right)}{\log(2)} \text{ 位}$$

**注：** 如果 PWM 占空比的值大于 PWM 周期，CCP1 引脚将不会被清零。

PWM 周期和占空比计算的示例，请参阅《PICmicro® 中档单片机系列参考手册》(DS33023A\_CN)。

## 9.3.3 PWM 操作的设置

通过以下步骤配置 CCP 模块的 PWM 操作：

1. 通过写 PR2 寄存器设置 PWM 周期。
2. 通过写 CCPR1L 寄存器和 CCP1CON<5:4> 位来设置 PWM 占空比。
3. 通过清零 TRISB<3> 位将 CCP1 引脚配置为输出。
4. 通过写 T2CON 设置 TMR2 预分频比并使能 Timer2。

**表 9-3: 20 MHz 时的 PWM 频率与分辨率示例**

PWM 频率	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
定时器预分频比 (1,4,16)	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	6.5

**表 9-4: 与 PWM 和 TIMER2 相关的寄存器**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
86h, 186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
11h	TMR2	Timer2 模块寄存器								0000 0000	0000 0000
92h	PR2	Timer2 模块周期寄存器								1111 1111	1111 1111
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	uuuu uuuu
15h	CCPR1L	捕捉 / 比较 / PWM 寄存器 1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	捕捉 / 比较 / PWM 寄存器 1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000

**图注：** x = 未知，u = 未改变，— = 未实现，读作 0。阴影部分表示未被 PWM 和 Timer2 使用。

# PIC16F627A/628A/648A

---

---

注:

## 10.0 比较器模块

比较器模块包含两个模拟比较器。比较器的输入端与 RA0 至 RA3 引脚复用。片上参考电压（参见第 11.0 节“参考电压模块”）也能作为比较器的输入。

如寄存器 10-1 所示，CMCON 寄存器控制比较器的输入和输出选择。图 10-1 是比较器的框图。

寄存器 10-1: **CMCON** — 比较器配置寄存器（地址：01Fh）

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
bit 7						bit 0	

bit 7 **C2OUT**: 比较器 2 输出位

当 C2INV = 0 时:  
 1 = C2 VIN+ > C2 VIN-  
 0 = C2 VIN+ < C2 VIN-

当 C2INV = 1 时:  
 1 = C2 VIN+ < C2 VIN-  
 0 = C2 VIN+ > C2 VIN-

bit 6 **C1OUT**: 比较器 1 输出位

当 C1INV = 0 时:  
 1 = C1 VIN+ > C1 VIN-  
 0 = C1 VIN+ < C1 VIN-

当 C1INV = 1 时:  
 1 = C1 VIN+ < C1 VIN-  
 0 = C1 VIN+ > C1 VIN-

bit 5 **C2INV**: 比较器 2 输出反相位

1 = C2 输出反相  
 0 = C2 输出不反相

bit 4 **C1INV**: 比较器 1 输出反相位

1 = C1 输出反相  
 0 = C1 输出不反相

bit 3 **CIS**: 比较器 1 输入切换位

当 CM<2:0> = 001 时  
 则:  
 1 = C1 VIN- 连接到 RA3  
 0 = C1 VIN- 连接到 RA0

当 CM<2:0> = 010 时  
 则:  
 1 = C1 VIN- 连接到 RA3  
     C2 VIN- 连接到 RA2  
 0 = C1 VIN- 连接到 RA0  
     C2 VIN- 连接到 RA1

bit 2-0 **CM<2:0>**: 比较器模式位

图 10-1 显示了比较器的各种模式以及 CM<2:0> 位的设置

<b>图注:</b>			
R = 可读位	W = 可写位	U = 未实现位, 读作 0	
-n = POR 时的值	1 = 置位	0 = 清零	x = 未知

# PIC16F627A/628A/648A

## 10.1 比较器配置

比较器共有 8 种工作模式。CMCON 寄存器用于选择这些模式，如图 10-1 所示。TRISA 寄存器控制每种模式下比较器引脚的数据方向。

如果改变了比较器模式，由于存在特定的模式改变延迟（如表 17-2 所示），可能导致比较器的输出电平无效。

**注 1:** 改变比较器模式的过程中，应禁止比较器中断，以免产生误中断。

**2:** 比较器输出可以反相。参见图 10-1。

图 10-1: 比较器 I/O 工作模式

<p>比较器复位 (POR 时的缺省值) CM&lt;2:0&gt; = 000</p> <p>RA0/AN0     A    VIN- RA3/AN3/CMP1   A    VIN+     C1     关闭 (读作 0)</p> <p>RA1/AN1     A    VIN- RA2/AN2/VREF   A    VIN+     C2     关闭 (读作 0)</p>	<p>比较器关闭 CM&lt;2:0&gt; = 111</p> <p>RA0/AN0     D    VIN- RA3/AN3/CMP1   D    VIN+     C1     关闭 (读作 0)</p> <p>RA1/AN1     D    VIN- RA2/AN2/VREF   D    VIN+     C2     关闭 (读作 0)</p> <p style="text-align: center;">VSS</p>
<p>两个独立的比较器 CM&lt;2:0&gt; = 100</p> <p>RA0/AN0     A    VIN- RA3/AN3/CMP1   A    VIN+     C1     C1VOUT</p> <p>RA1/AN1     A    VIN- RA2/AN2/VREF   A    VIN+     C2     C2VOUT</p>	<p>四路输入的双比较器 CM&lt;2:0&gt; = 010</p> <p>RA0/AN0     A    VIN- RA3/AN3/CMP1   A    VIN+     C1     C1VOUT</p> <p style="margin-left: 20px;">CIS = 0    CIS = 1</p> <p>RA1/AN1     A    VIN- RA2/AN2/VREF   A    VIN+     C2     C2VOUT</p> <p style="margin-left: 20px;">CIS = 0    CIS = 1</p> <p style="text-align: right;">来自 VREF 模块</p>
<p>两个具有公共参考端的比较器 CM&lt;2:0&gt; = 011</p> <p>RA0/AN0     A    VIN- RA3/AN3/CMP1   D    VIN+     C1     C1VOUT</p> <p>RA1/AN1     A    VIN- RA2/AN2/VREF   A    VIN+     C2     C2VOUT</p>	<p>两个具有公共参考端且带输出的比较器 CM&lt;2:0&gt; = 110</p> <p>RA0/AN0     A    VIN- RA3/AN3/CMP1   D    VIN+     C1     C1VOUT</p> <p>RA1/AN1     A    VIN- RA2/AN2/VREF   A    VIN+     C2     C2VOUT</p> <p>RA4/T0CKI/CMP2   漏极开路</p>
<p>单个独立的比较器 CM&lt;2:0&gt; = 101</p> <p>RA0/AN0     D    VIN- RA3/AN3/CMP1   D    VIN+     C1     关闭 (读作 0)</p> <p style="text-align: center;">VSS</p> <p>RA1/AN1     A    VIN- RA2/AN2/VREF   A    VIN+     C2     C2VOUT</p>	<p>三路输入的双比较器 CM&lt;2:0&gt; = 001</p> <p>RA0/AN0     A    VIN- RA3/AN3/CMP1   A    VIN+     C1     C1VOUT</p> <p style="margin-left: 20px;">CIS = 0    CIS = 1</p> <p>RA1/AN1     A    VIN- RA2/AN2/VREF   A    VIN+     C2     C2VOUT</p>
<p>A = 模拟输入，端口始终读作 0。                      D = 数字输入。                      CIS (CMCON&lt;3&gt;) 是比较器输入切换位。</p>	



例 10-1 中的代码描述了配置比较器模块所必需的步骤。RA3 和 RA4 配置为数字输出。RA0 和 RA1 配置为两个比较器的 V- 输入，RA2 配置为两个比较器的 V+ 输入。

## 例 10-1: 初始化比较器模块

```

FLAG_REG EQU 0X20
CLRF FLAG_REG ;Init flag register
CLRF PORTA ;Init PORTA
MOVF CMCON, W ;Load comparator bits
ANDLW 0xC0 ;Mask comparator bits
IORWF FLAG_REG, F ;Store bits in flag register
MOVLW 0x03 ;Init comparator mode
MOVWF CMCON ;CM<2:0> = 011
BSF STATUS, RP0 ;Select Bank1
MOVLW 0x07 ;Initialize data direction
MOVWF TRISA ;Set RA<2:0> as inputs
;RA<4:3> as outputs
;TRISA<7:5> always read

&#xD4;0&#xD5;
BCF STATUS, RP0 ;Select Bank 0
CALL DELAY10 ;10&#xB5;s delay
MOVF CMCON, F ;Read CMCON to end change
;condition
BCF PIR1, CMIF ;Clear pending interrupts
BSF STATUS, RP0 ;Select Bank 1
BSF PIE1, CMIE ;Enable comparator interrupts
BCF STATUS, RP0 ;Select Bank 0
BSF INTCON, PEIE ;Enable peripheral interrupts
    
```

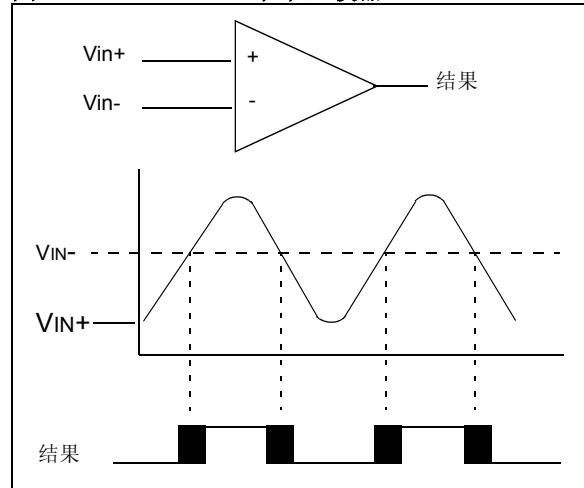
## 10.2 比较器工作原理

图 10-2 所示为单比较器及模拟输入电平与数字输出之间的关系。当 VIN+ 端的模拟输入电压低于 VIN- 时，比较器输出数字低电平。当 VIN+ 端的模拟输入电压高于 VIN- 时，比较器输出数字高电平。图 10-2 中比较器输出部分的阴影区代表由于输入失调电压和响应时间所引起的不确定区。关于共模电压，请参见表 17-2。

## 10.3 比较器参考电压源

根据不同的工作模式，比较器可以使用外部或内部的参考电压源。加在 VIN- 上的模拟信号与 VIN+ 上的信号相比较，比较器的数字输出也随着比较的结果作出相应调整（图 10-2）。

图 10-2: 单个比较器



### 10.3.1 外部参考信号

当使用外部参考电压时，比较器模块可以配置为两个比较器在同一个参考电压源或不同的参考电压源下工作。然而，阈值检测器应用中可能要求使用同一个参考电压源。参考电压信号必须介于 VSS 和 VDD 之间，且可施加到比较器的任一引脚上。

### 10.3.2 内部参考电压信号

比较器模块也可以为比较器选择使用内部产生的参考电压。第 11.0 节“参考电压模块”对提供这一信号的参考电压模块进行了详细描述。在 CM<2:0> = 010 模式（图 10-1）下，比较器使用内部参考电压信号。这种模式下，内部参考电压加到两个比较器的 VIN+ 引脚上。

## 10.4 比较器的响应时间

响应时间是指比较器从选定一个新的参考电压或输入电压源到输出达到一个有效电平时的最短时间。如果内部参考电压发生了改变，在使用比较器输出时必须考虑到内部参考电压的最大延迟的影响。否则，应使用比较器的最大延迟（第 142 页的表 17-2）。

# PIC16F627A/628A/648A

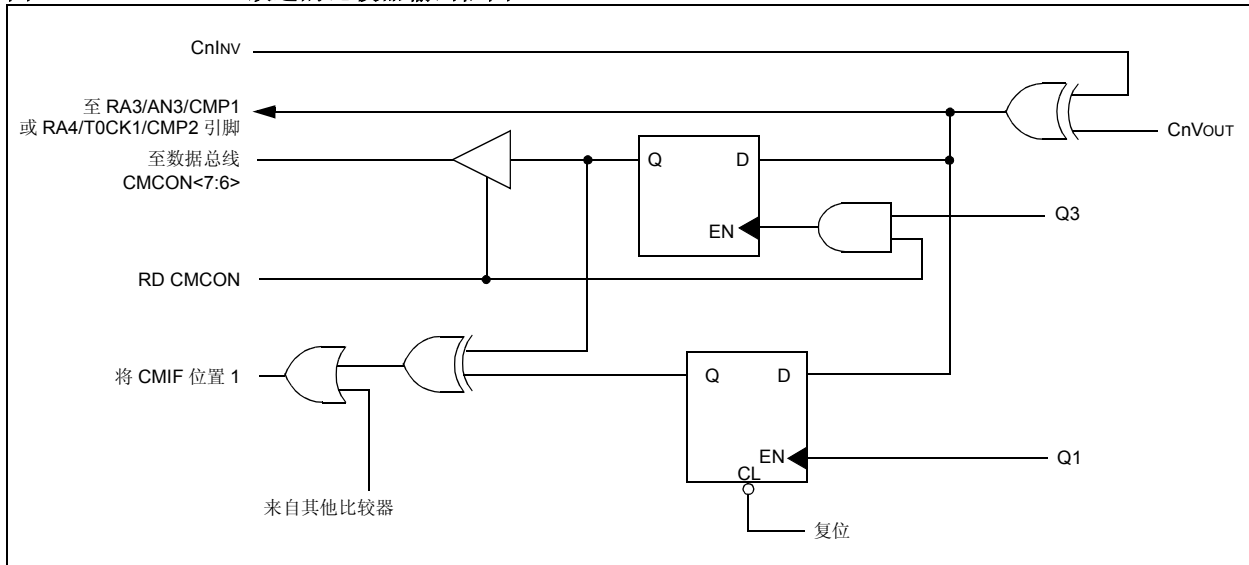
## 10.5 比较器输出

通过读 CMCON 寄存器的相应位，可以得到比较器的输出。这些位都是只读的。比较器的输出也可以直接输出到 I/O 引脚 RA3 和 RA4。当  $CM\langle 2:0 \rangle = 110$  或  $001$  时，RA3 和 RA4/T0CK1/CMP2 引脚输出路径上的多路开关将发生切换，那么各个引脚的输出将为比较器的不同步输出。每个比较器输出的不确定区与规范中给出的输入失调电压和响应时间有关。图 10-3 为比较器的输出框图。

在该模式下，TRISA 寄存器的相应位仍然可以允许或禁止 RA3/AN3/CMP1 和 RA4/T0CK1/CMP2 为输出。

- 注 1:** 当读端口寄存器时，所有配置为模拟输入的引脚都读作“0”。而配置为数字输入的引脚将根据施密特触发器输入规范对模拟输入进行转换。
- 2:** 模拟电平加在定义为数字输入的引脚上，将导致输入缓冲器的电流消耗超出规定值。

图 10-3: 改进的比较器输出框图



## 10.6 比较器中断

一旦两个比较器中的任意一个的输出值发生变化，就会将该比较器的中断标志位置 1。当从 CMCON<7:6> 读取数据的时候，需要用软件来保持输出位的状态信息以判断实际发生的变化。CMIF 位 PIR1<6> 是比较器中断标志位。CMIF 位必须通过清零复位。因为也可以把“1”写入该寄存器，所以可以产生模拟中断。

必须将 CMIE 位 (PIE1<6>) 和 PEIE 位 (INTCON<6>) 置位以允许中断。此外，GIE 位也必须置位。只要这些位中的任何一位被清零，虽然当有中断条件产生时 CMIF 位仍会置位，但却仍然不允许中断。

<b>注：</b>	当执行一个读操作时（Q2 周期开始时），如果 CMCON 寄存器（C1OUT 或 C2OUT）发生变化，那么 CMIF（PIR1<6>）中断标志位可能不会被置位。
-----------	---

在中断服务程序中，用户可以通过以下方式清除中断：

- a) 任何对 CMCON 寄存器的读写操作都将结束失配状态。
- b) 将中断标志位 CMIF 清零。

失配状态会一直不断地将 CMIF 标志位置 1。读 CMCON 寄存器将结束这种失配状态，并允许清零 CMIF 标志位。

## 10.7 休眠状态下比较器的操作

当比较器处于有效状态而器件处于休眠模式时，比较器仍能继续工作，此时如果允许中断，则中断同样有效。在允许中断时，中断会把器件从休眠模式唤醒。当比较器上电时，消耗的休眠电流将比掉电电流规范中指定的高。每个比较器工作时都会消耗额外的电流（如比较器的规范中所示）。若要使休眠模式下的功耗最小，可在进入休眠状态前，配置 CM<2:0> = 111，关断比较器。器件从休眠状态中唤醒时，CMCON 寄存器的内容不受影响。

## 10.8 复位的影响

器件复位会强制 CMCON 寄存器为其复位值。这将使比较器模块处于比较器复位模式 (CM<2:0> = 000)。确保所有可能的输入都为模拟输入。当复位时引脚为模拟输入将使器件电流达到最小。在复位期间，比较器将掉电。

## 10.9 模拟输入连接注意事项

图 10-4 是一个简化的模拟输入电路。由于模拟引脚和数字输出相连，因而它们与 VDD 及 VSS 之间加有反向偏置二极管。因此模拟输入电压被限制在 VSS 和 VDD 之间。如果输入电压在任一方向上偏离这个范围超过 0.6V，就会有一个二极管正偏从而使输入电压被钳位。推荐模拟信号源的最大内阻为 10 kΩ。任何连接到模拟输入引脚上的外部元件（如电容器、稳压二极管等），在引脚上的泄漏电流应该极小。

# PIC16F627A/628A/648A

图 10-4: 模拟输入模式

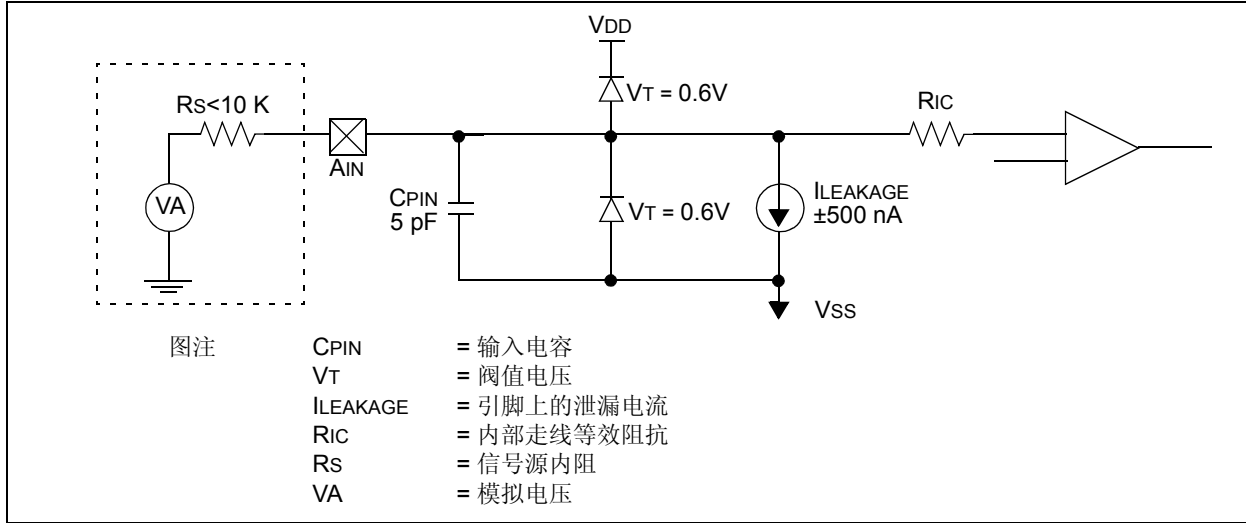


表 10-1: 与比较器模块相关的寄存器

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
1Fh	CMCON	C2OUT	C1OUT	C2INV	C1NV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111

图注: x = 未知, u = 未改变, — = 未实现, 读作 0。

## 11.0 参考电压模块

参考电压模块是由一个 16 级的梯形电阻网络组成，提供了可选择的参考电压。将梯形电阻网络分段来提供两个不同范围的VREF，并且在不需要使用参考电压时可以将其关闭以降低功耗。如图 11-1 所示，VRCON 寄存器对参考电压模块进行控制。图 11-1 为该参考电压模块的框图。

### 11.1 参考电压配置

在每个范围内，参考电压模块均可输出 16 种不同的电压值。

下面是计算参考电压模块输出的公式：

如果 VRR = 1:

$$V_{REF} = \frac{VR<3:0>}{24} \times V_{DD}$$

如果 VRR = 0:

$$V_{REF} = \left( V_{DD} \times \frac{1}{4} \right) + \frac{VR<3:0>}{32} \times V_{DD}$$

当改变 VREF 输出时，必需考虑参考电压模块的稳定时间（表 17-3）。例 11-1 说明了当 VDD 为 5.0V 时，欲获得输出电压 1.25V，如何配置参考电压模块。

寄存器 11-1:

**VRCON — 参考电压模块控制寄存器（地址：9Fh）**

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VREN	VROE	VRR	—	VR3	VR2	VR1	VR0
bit 7				bit 0			

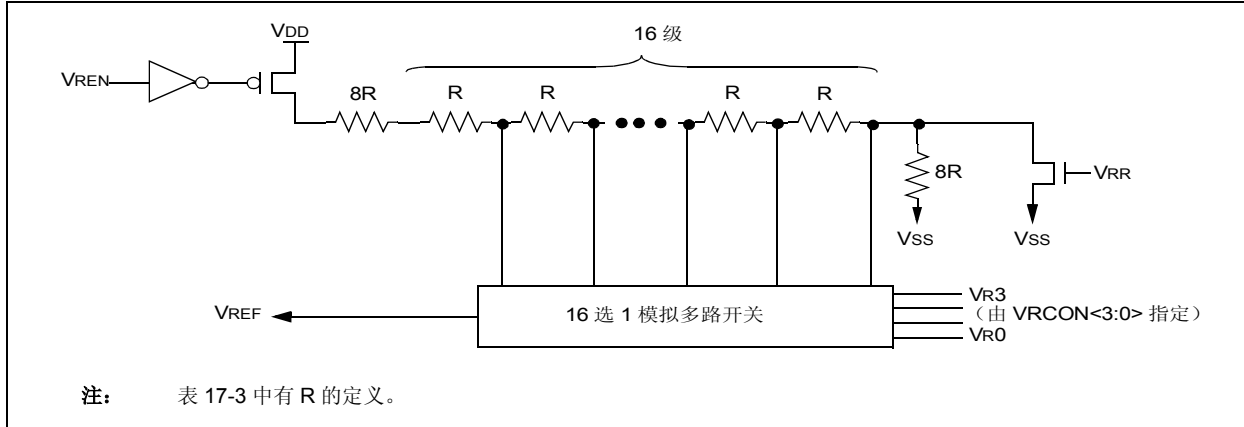
- bit7 **VREN:** VREF 使能位  
1 = VREF 电路上电  
0 = VREF 电路断电，无 IDD 漏电流
- bit6 **VROE:** VREF 输出使能位  
1 = VREF 在 RA2 引脚输出  
0 = VREF 从 RA2 引脚断开
- bit5 **VRR:** VREF 范围选择位  
1 = 低参考电压范围  
0 = 高参考电压范围
- bit4 **未实现:** 读作 0
- bit3-0 **VR<3:0>:** VREF 值选择位， $0 \leq VR<3:0> \leq 15$   
当 VRR = 1 时:  $V_{REF} = (VR<3:0>/24) * V_{DD}$   
当 VRR = 0 时:  $V_{REF} = 1/4 * V_{DD} + (VR<3:0>/32) * V_{DD}$

**图注:**

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

# PIC16F627A/628A/648A

图 11-1: 参考电压模块框图



例 11-1: 参考电压配置

```
MOVLW 0x02 ;4 Inputs Muxed
MOVWF CMCON ;to 2 comps.
BSF STATUS,RP0 ;go to Bank 1
MOVLW 0x07 ;RA3-RA0 are
MOVWF TRISA ;outputs
MOVLW 0xA6 ;enable VREF
MOVWF VRCON ;low range set VR<3:0>=6
BCF STATUS,RP0 ;go to Bank 0
CALL DELAY10 ;10ms delay
```

## 11.2 参考电压精度 / 误差

由于结构上的限制,参考电压模块不能实现从  $V_{SS}$  到  $V_{DD}$  的全范围输出。梯形电阻网络顶部和底部的晶体管(图 11-1)使  $V_{REF}$  的值不能达到  $V_{SS}$  或  $V_{DD}$ 。由于参考电压是由  $V_{DD}$  产生的,因此  $V_{REF}$  输出随着  $V_{DD}$  的波动而变化。经测试的参考电压绝对精度参见表 17-3。

## 11.3 休眠期间工作原理

如果因中断或看门狗超时将器件从休眠模式下唤醒,  $VRCON$  寄存器内容将不受影响。为了使休眠模式下的电流消耗达到最小,应关闭参考电压模块。

## 11.4 复位的影响

单片机复位会使  $VREN$  ( $VRCON<7>$ ) 位清零,从而禁止参考电压模块。复位还将  $VROE$  位 ( $VRCON<6>$ ) 清零,从而使参考电压与  $RA2$  引脚断开;同时通过将  $VRR$  位 ( $VRCON<5>$ ) 清零选择高电压范围。 $VREF$  值选择位  $VRCON<3:0>$  也被清零。

## 11.5 连接注意事项

参考电压模块的操作是独立于比较器模块的。如果  $TRISA<2>$  位置位并且  $VROE$  位 ( $VRCON<6>$ ) 也置位的话,参考电压发生器的输出可以连接到  $RA2$  引脚。参考电压输出与  $RA2$  引脚相连时,如果该引脚上还有输入信号,将会增大电流消耗。使能  $VREF$  时,连接  $RA2$  作为数字输出也会增加电流消耗。

$RA2$  引脚可以作为简单的 D/A 输出,但是其驱动能力有限。因为其驱动能力有限,所以必须在  $VREF$  输出端外接一个缓冲器。图 11-2 举例说明了这一缓冲技术。

图 11-2: 参考电压输出缓冲示例

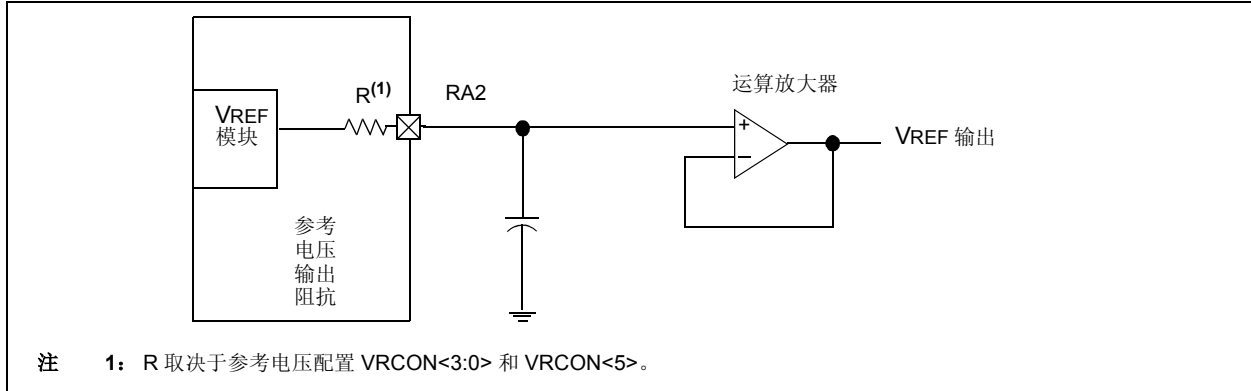


表 11-1: 与参考电压相关的寄存器

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000
1Fh	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111

图注: — = 未实现, 读作 0。

# PIC16F627A/628A/648A

---

---

注:



## 12.0 通用同步 / 异步收发器 (USART) 模块

通用同步 / 异步收发器 (Universal Synchronous Asynchronous Receiver Transmitter, USART) 也是一种串行通信接口 (Serial Communications Interface, SCI)。USART 可以配置为全双工异步系统, 使之能与 CRT 终端和个人计算机等外设进行通信; 也可将其配置为半双工同步系统, 使之能与 A/D 或 D/A 集成电路以及串行 EEPROM 等外设进行通信。

USART 可配置为以下几种工作模式:

- 全双工异步模式
- 同步 — 主模式 (半双工)
- 同步 — 从模式 (半双工)

要将 RB2/TX/CK 引脚和 RB1/RX/DT 引脚配置为通用同步 / 异步收发器引脚, 必须置位 SPEN (RCSTA<7>) 位和 TRISB<2:1> 位。

寄存器 12-1 所示为发送状态和控制寄存器 (TXSTA), 寄存器 12-2 所示为接收状态和控制寄存器 (RCSTA)。

寄存器 12-1: TXSTA — 发送状态和控制寄存器 (地址: 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

- bit 7 **CSRC:** 时钟源选择位  
异步模式  
 忽略  
同步模式  
 1 = 主模式 (由内部 BRG 产生时钟信号)  
 0 = 从模式 (由外部时钟源提供时钟信号)
- bit 6 **TX9:** 9 位发送使能位  
 1 = 选择 9 位发送  
 0 = 选择 8 位发送
- bit 5 **TXEN:** 发送使能位 <sup>(1)</sup>  
 1 = 使能发送  
 0 = 禁止发送
- bit 4 **SYNC:** USART 模式选择位  
 1 = 同步模式  
 0 = 异步模式
- bit 3 **未实现位:** 读作 0
- bit 2 **BRGH:** 高速波特率选择位  
异步模式  
 1 = 高速  
 0 = 低速  
同步模式  
 在此模式下未使用
- bit 1 **TRMT:** 发送移位寄存器状态位  
 1 = TSR 空  
 0 = TSR 满
- bit 0 **TX9D:** 发送数据的第 9 位。可作为奇偶校验位。  
**注 1:** 在同步模式下, SREN/CREN 位比 TXEN 位优先级高。

**图注:**

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

# PIC16F627A/628A/648A

## 寄存器 12-2: RCSTA — 接收状态和控制寄存器 (地址: 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D
bit 7					bit 0		

- bit 7 **SPEN:** 串行端口使能位  
(当 TRISB<2:1> 位置位时, RB1/RX/DT 和 RB2/TX/CK 引脚将被配置为串行端口引脚)  
1 = 使能串行端口  
0 = 禁止串行端口
- bit 6 **RX9:** 9 位接收使能位  
1 = 选择 9 位接收  
0 = 选择 8 位接收
- bit 5 **SREN:** 单字节接收使能位  
异步模式:  
忽略  
同步主模式:  
1 = 使能单字节接收  
0 = 禁止单字节接收  
此位在接收完成后清零。  
同步从模式:  
在此模式下未使用
- bit 4 **CREN:** 连续接收使能位  
异步模式:  
1 = 允许连续接收  
0 = 禁止连续接收  
同步模式:  
1 = 使能连续接收, 直到 CREN 使能位被清零 (CREN 位的优先级比 SREN 位高)  
0 = 禁止连续接收
- bit 3 **ADEN:** 地址检测使能位  
9 位同步模式 (RX9 = 1):  
1 = 使能地址检测, RSR<8> 置位时, 允许中断并装入接收缓冲器  
0 = 禁止地址检测, 已接收所有字节, 第 9 位可作为奇偶校验位  
8 位异步模式 (RX9 = 0):  
在此模式下未使用  
同步模式  
在此模式下未使用
- bit 2 **FERR:** 帧错误标志位  
1 = 帧错误 (读 RCREG 寄存器可更新该位并可接收下一个有效字节)  
0 = 无帧错误
- bit 1 **OERR:** 溢出错误标志位  
1 = 有溢出错误 (清零 CREN 位可将此位清零)  
0 = 无溢出错误
- bit 0 **RX9D:** 接收数据的第 9 位 (可作为奇偶校验位)

### 图注:

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

## 12.1 USART 波特率发生器

波特率发生器 (Baud Rate Generator, BRG) 支持 USART 的同步模式和异步模式。这是一个专用的 8 位波特率发生器。SPBRG 寄存器控制着自由运行的 8 位定时器的周期。在异步模式下, BRGH (TXSTA<2>) 位还用于控制波特率。在同步模式下可忽略 BRGH 位。表 12-1 所示为仅应用于主模式 (内部时钟) 下的不同 USART 工作模式的波特率计算公式。

在理想的波特率和 Fosc 下, 可以使用表 12-1 中的公式计算 SPBRG 寄存器的最近似整数值。因此, 可以判断波特率的误差。

公式 12-1 说明了在下列条件下波特率误差的计算方法

Fosc = 16 MHz  
 目标波特率 = 9600  
 BRGH = 0  
 SYNC = 0

公式 12-1: 计算波特率误差

$$\begin{aligned} \text{目标波特率} &= \frac{F_{osc}}{64(x+1)} \\ 9600 &= \frac{16000000}{64(x+1)} \\ x &= 25.042 \\ \text{计算波特率} &= \frac{16000000}{64(25+1)} = 9615 \\ \text{误差} &= \frac{(\text{计算波特率} - \text{目标波特率})}{\text{目标波特率}} \\ &= \frac{9615 - 9600}{9600} = 0.16\% \end{aligned}$$

即使对于低波特率时钟, 使用高波特率公式 (BRGH=1) 也可能会有好处。这是因为公式  $F_{osc}/(16(X+1))$  在某些情况下会降低波特率误差。

向 SPBRG 寄存器写入一个新值将导致 BRG 定时器复位 (或清零), 这可以确保 BRG 不必等待定时器溢出再输出新波特率。

表 12-1: 波特率公式

SYNC	BRGH = 0 (低速)	BRGH = 1 (高速)
0	(异步) 波特率 = $F_{osc}/(64(X+1))$	波特率 = $F_{osc}/(16(X+1))$
1	(同步) 波特率 = $F_{osc}/(4(X+1))$	NA

图注: X 为 SPBRG 寄存器中的值 (0 到 255)

表 12-2: 与波特率发生器有关的寄存器

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	波特率发生器寄存器								0000 0000	0000 0000

图注: x = 未知, — = 未实现, 读作 0。阴影部分表示未被 BRG 使用。

# PIC16F627A/628A/648A

表 12-3: 同步模式下的波特率

目标波特率 (Kbps)	Fosc = 20 MHz			16 MHz			10 MHz		
	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	NA	—	—	NA	—	—	NA	—	—
2.4	NA	—	—	NA	—	—	NA	—	—
9.6	NA	—	—	NA	—	—	9.766	+1.73%	255
19.2	19.53	+1.73%	255	19.23	+0.16%	207	19.23	+0.16%	129
76.8	76.92	+0.16%	64	76.92	+0.16%	51	75.76	-1.36%	32
96	96.15	+0.16%	51	95.24	-0.79%	41	96.15	+0.16%	25
300	294.1	-1.96	16	307.69	+2.56%	12	312.5	+4.17%	7
500	500	0	9	500	0	7	500	0	4
高	5000	—	0	4000	—	0	2500	—	0
低	19.53	—	255	15.625	—	255	9.766	—	255

目标波特率 (Kbps)	Fosc = 7.15909 MHz			5.0688 MHz			4 MHz		
	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	NA	—	—	NA	—	—	NA	—	—
2.4	NA	—	—	NA	—	—	NA	—	—
9.6	9.622	+0.23%	185	9.6	0	131	9.615	+0.16%	103
19.2	19.24	+0.23%	92	19.2	0	65	19.231	+0.16%	51
76.8	77.82	+1.32	22	79.2	+3.13%	15	75.923	+0.16%	12
96	94.20	-1.88	18	97.48	+1.54%	12	1000	+4.17%	9
300	298.3	-0.57	5	316.8	5.60%	3	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
高	1789.8	—	0	1267	—	0	100	—	0
低	6.991	—	255	4.950	—	255	3.906	—	255

目标波特率 (Kbps)	Fosc = 3.579545 MHz			1 MHz			32.768 kHz		
	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)
0.3	NA	—	—	NA	—	—	0.303	+1.14%	26
1.2	NA	—	—	1.202	+0.16%	207	1.170	-2.48%	6
2.4	NA	—	—	2.404	+0.16%	103	NA	—	—
9.6	9.622	+0.23%	92	9.615	+0.16%	25	NA	—	—
19.2	19.04	-0.83%	46	19.24	+0.16%	12	NA	—	—
76.8	74.57	-2.90%	11	83.34	+8.51%	2	NA	—	—
96	99.43	+3.57%	8	NA	—	—	NA	—	—
300	298.3	0.57%	2	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
高	894.9	—	0	250	—	0	8.192	—	0
低	3.496	—	255	0.9766	—	255	0.032	—	255

# PIC16F627A/628A/648A

表 12-4: 异步模式下的波特率 (BRGH = 0)

目标波特率 (Kbps)	Fosc = 20 MHz			16 MHz			10 MHz		
	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	1.221	+1.73%	255	1.202	+0.16%	207	1.202	+0.16%	129
2.4	2.404	+0.16%	129	2.404	+0.16%	103	2.404	+0.16%	64
9.6	9.469	-1.36%	32	9.615	+0.16%	25	9.766	+1.73%	15
19.2	19.53	+1.73%	15	19.23	+0.16%	12	19.53	+1.73%	7
76.8	78.13	+1.73%	3	83.33	+8.51%	2	78.13	+1.73%	1
96	104.2	+8.51%	2	NA	—	—	NA	—	—
300	312.5	+4.17%	0	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
高	312.5	—	0	250	—	0	156.3	—	0
低	1.221	—	255	0.977	—	255	0.6104	—	255

目标波特率 (Kbps)	Fosc = 7.15909 MHz			5.0688 MHz			4 MHz		
	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)
0.3	NA	—	—	0.31	+3.13%	255	0.3005	-0.17%	207
1.2	1.203	+0.23%	92	1.2	0	65	1.202	+1.67%	51
2.4	2.380	-0.83%	46	2.4	0	32	2.404	+1.67%	25
9.6	9.322	-2.90%	11	9.9	+3.13%	7	NA	—	—
19.2	18.64	-2.90%	5	19.8	+3.13%	3	NA	—	—
76.8	NA	—	—	79.2	+3.13%	0	NA	—	—
96	NA	—	—	NA	—	—	NA	—	—
300	NA	—	—	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
高	111.9	—	0	79.2	—	0	62.500	—	0
低	0.437	—	255	0.3094	—	255	3.906	—	255

目标波特率 (Kbps)	Fosc = 3.579545 MHz			1 MHz			32.768 kHz		
	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)
0.3	0.301	+0.23%	185	0.300	+0.16%	51	0.256	-14.67%	1
1.2	1.190	-0.83%	46	1.202	+0.16%	12	NA	—	—
2.4	2.432	+1.32%	22	2.232	-6.99%	6	NA	—	—
9.6	9.322	-2.90%	5	NA	—	—	NA	—	—
19.2	18.64	-2.90%	2	NA	—	—	NA	—	—
76.8	NA	—	—	NA	—	—	NA	—	—
96	NA	—	—	NA	—	—	NA	—	—
300	NA	—	—	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
高	55.93	—	0	15.63	—	0	0.512	—	0
低	0.2185	—	255	0.0610	—	255	0.0020	—	255

# PIC16F627A/628A/648A

表 12-5: 异步模式下的波特率 (BRGH = 1)

目标波特率 (Kbps)	Fosc = 20 MHz			16 MHz			10 MHz		
	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)
9.600	9.615	+0.16%	129	9.615	+0.16%	103	9.615	+0.16%	64
19.200	19.230	+0.16%	64	19.230	+0.16%	51	18.939	-1.36%	32
38.400	37.878	-1.36%	32	38.461	+0.16%	25	39.062	+1.7%	15
57.600	56.818	-1.36%	21	58.823	+2.12%	16	56.818	-1.36%	10
115.200	113.636	-1.36%	10	111.111	-3.55%	8	125	+8.51%	4
250	250	0	4	250	0	3	NA	—	—
625	625	0	1	NA	—	—	625	0	0
1250	1250	0	0	NA	—	—	NA	—	—

目标波特率 (Kbps)	Fosc = 7.16 MHz			5.068 MHz			4 MHz		
	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)
9.600	9.520	-0.83%	46	9.598	0.016%	32	9.615	0.160%	25
19.200	19.454	+1.32%	22	18.632	-2.956%	16	19.23	0.160%	12
38.400	37.286	-2.90%	11	39.594	3.109%	7	35.714	-6.994%	6
57.600	55.930	-2.90%	7	52.792	-8.348%	5	62.5	8.507%	3
115.200	111.860	-2.90%	3	105.583	-8.348%	2	125	8.507%	1
250	NA	—	—	316.75	26.700%	0	250	0.000%	0
625	NA	—	—	NA	—	—	NA	—	—
1250	NA	—	—	NA	—	—	NA	—	—

目标波特率 (Kbps)	Fosc = 3.579 MHz			1 MHz			32.768 kHz		
	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)	计算波特率 (Kbps)	误差	SPBRG 值 (10 进制)
9.600	9.726	1.308%	22	8.928	-6.994%	6	NA	NA	NA
19.200	18.641	-2.913%	11	20.833	8.507%	2	NA	NA	NA
38.400	37.281	-2.913%	5	31.25	-18.620%	1	NA	NA	NA
57.600	55.922	-2.913%	3	62.5	+8.507	0	NA	NA	NA
115.200	111.244	-2.913%	1	NA	—	—	NA	NA	NA
250	223.688	-10.525%	0	NA	—	—	NA	NA	NA
625	NA	—	—	NA	—	—	NA	NA	NA
1250	NA	—	—	NA	—	—	NA	NA	NA

## 12.2 USART 异步模式

在此模式下，USART 使用标准的“非归零码”（non-return-to-zero, NRZ）格式（1 个起始位，8 个或 9 个数据位，1 个停止位）。最常用的数据格式为 8 位。专用的 8 位波特率发生器用于从振荡器产生波特率频率。USART 首先发送和接收 LSB。USART 的发送器和接收器在功能上是独立的，但采用相同的数据格式和波特率。波特率发生器可以根据 BRGH 位（TXSTA<2>）的状态产生相当于移位速率 16 倍或 64 倍的波特率时钟。USART 硬件不支持奇偶校验，但该功能可由软件实现（奇偶校验位是第 9 个数据位）。在休眠模式下，异步模式会停止工作。

通过对 SYNC 位（TXSTA<4>）清零，可选择异步工作模式。

USART 异步模块中的重要组成部分有：

- 波特率发生器
- 采样电路
- 异步发送器
- 异步接收器

### 12.2.1 USART 异步发送器

图 12-1 所示是 USART 发送器的框图。发送器的核心是发送（串行）移位寄存器（TSR）。移位寄存器从读 / 写发送缓冲器 TXREG 中获取数据。TXREG 寄存器要用软件装入数据。在前一次装入数据的停止位发送完成前，不会向 TSR 寄存器装入数据。一旦停止位发送完毕，TSR 就会自动装入 TXREG 中的新数据（如果 TXREG 中有数据的话）。一旦 TXREG 寄存器向 TSR 寄存器传输完数据（在 1 个 Tcy 内发生），TXREG 寄存器就为空，同时标志位 TXIF（PIR1<4>）置位。可以通过置位或清零 TXIE 使能位（PIE1<4>）来使能或禁止该中断。不管使能位 TXIE 的状态如何，标志位 TXIF 都将置位，且无法用软件清零。只有向 TXREG 寄存器装入新数据此位才会复位。标志位 TXIF 表示 TXREG 寄存器的状态，而另一个位 TRMT（TXSTA<1>）则表示 TSR 寄存器的状态。状态位 TRMT 是只读位，当 TSR 寄存器为空时该位置位。因为没有与此位相关的中断逻辑，所以用户必须查询此位以判断 TSR 寄存器是否为空。

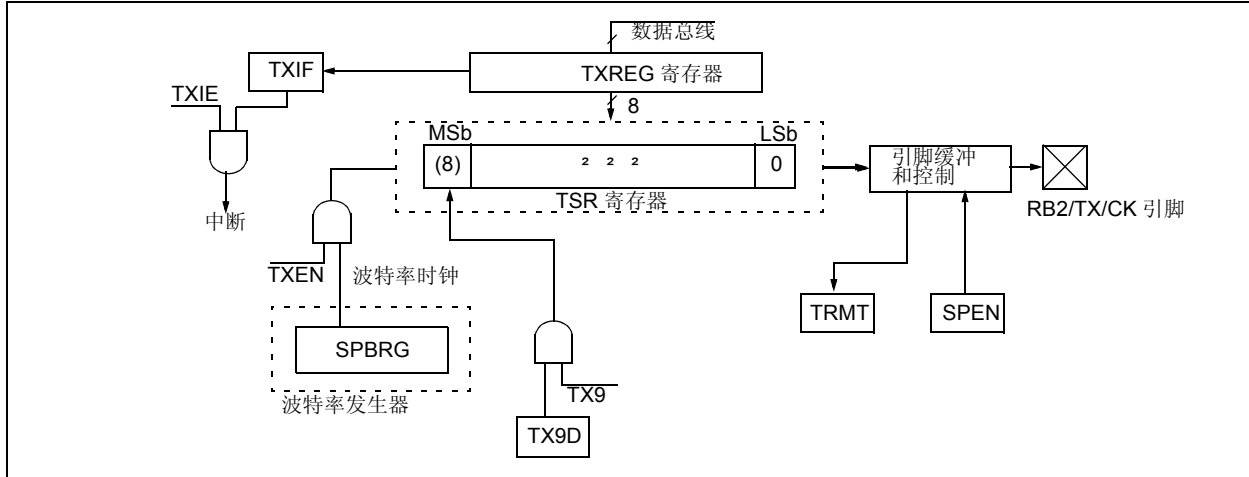
- |          |   |
|----------|---|
| <b>注</b> | <b>1:</b> TSR 寄存器并未映射到数据存储寄存器中，因此用户不能直接访问它。 |
|          | <b>2:</b> 当使能位 TXEN 置位时，标志位 TXIF 置位。        |

通过将使能位 TXEN（TXSTA<5>）置位使能发送。但是只有在 TXREG 寄存器装入数据和波特率发生器（BRG）产生移位时钟之后才能实际进行数据发送（图 12-1）。也可以通过先把发送数据装入 TXREG 寄存器，再将使能位 TXEN 位置位来启动发送。通常，在第一次启动发送时，TSR 寄存器为空，因此送到 TXREG 寄存器的数据会被立即送到 TSR 寄存器，从而导致 TXREG 寄存器为空。因此可以进行背对背发送（图 12-3）。在发送过程中将 TXEN 位清零会导致发送中止，同时复位发送器。因此 TX/CK 引脚会恢复到高阻态。

为选择 9 位数据发送方式，应将发送位 TX9 位（TXSTA<6>）置 1，并且第 9 位应写入 TX9D 位（TXSTA<0>）。必须先写第 9 位数据，然后再将 8 位数据写入 TXREG 寄存器。这是因为如果 TSR 寄存器为空，向 TXREG 寄存器写数据会导致数据立即送入 TSR 寄存器。在这种情况下，装入 TSR 寄存器的第 9 位数据可能是错误的。

# PIC16F627A/628A/648A

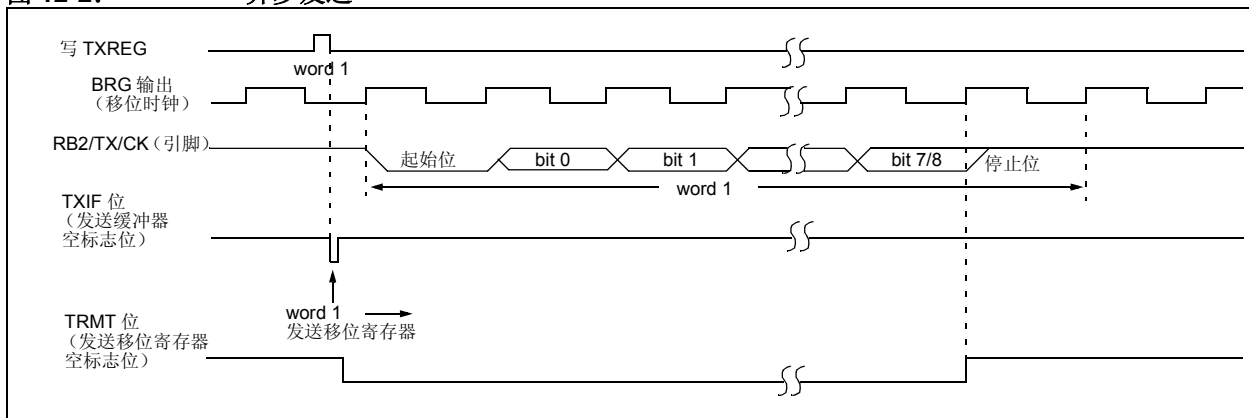
图 12-1: USART 发送框图



在设置异步发送模式时应遵循下面这些步骤：

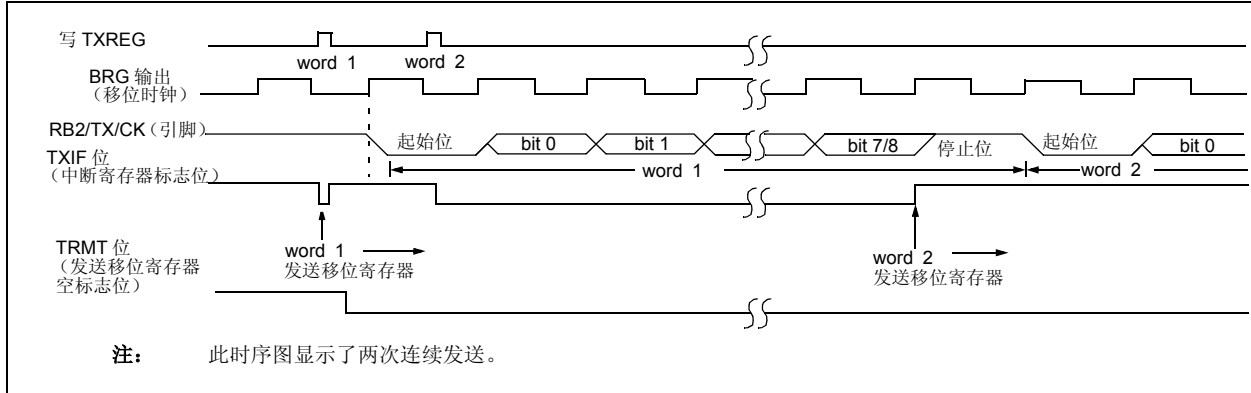
1. 应同时置位 TRISB<1> 位和 TRISB<2> 位，将 RB1/RX/DT 和 RB2/TX/CK 引脚配置为输入。需要时，通过外设电路控制输出驱动。
2. 用正确的波特率初始化 SPBRG 寄存器。如果需要高速波特率，应将 BRGH 位置位。（参见第 12.1 节“USART 波特率发生器”）。
3. 通过将 SYNC 位清零并将 SPEN 位置位，使能异步串行口。
4. 如果需要中断，应将使能位 TXIE 置位。
5. 如果需要发送 9 位数据，应将发送位 TX9 置位。
6. 通过将 TXEN 位置位使能发送，此操作同时也会将 TXIF 位置位。
7. 如果选择发送 9 位数据，应该将第 9 位数据装入 TX9D 位。
8. 将数据装入 TXREG 寄存器（开始发送）。

图 12-2: 异步发送





**图 12-3: 异步发送 (背对背)**



**表 12-6: 与异步发送相关的寄存器**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	USART 发送数据寄存器								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	波特率发生器寄存器								0000 0000	0000 0000

图注: x = 未知, — = 未实现, 读作 0。  
阴影部分表示在异步发送模式下未使用。

# PIC16F627A/628A/648A

## 12.2.2 USART 异步接收器

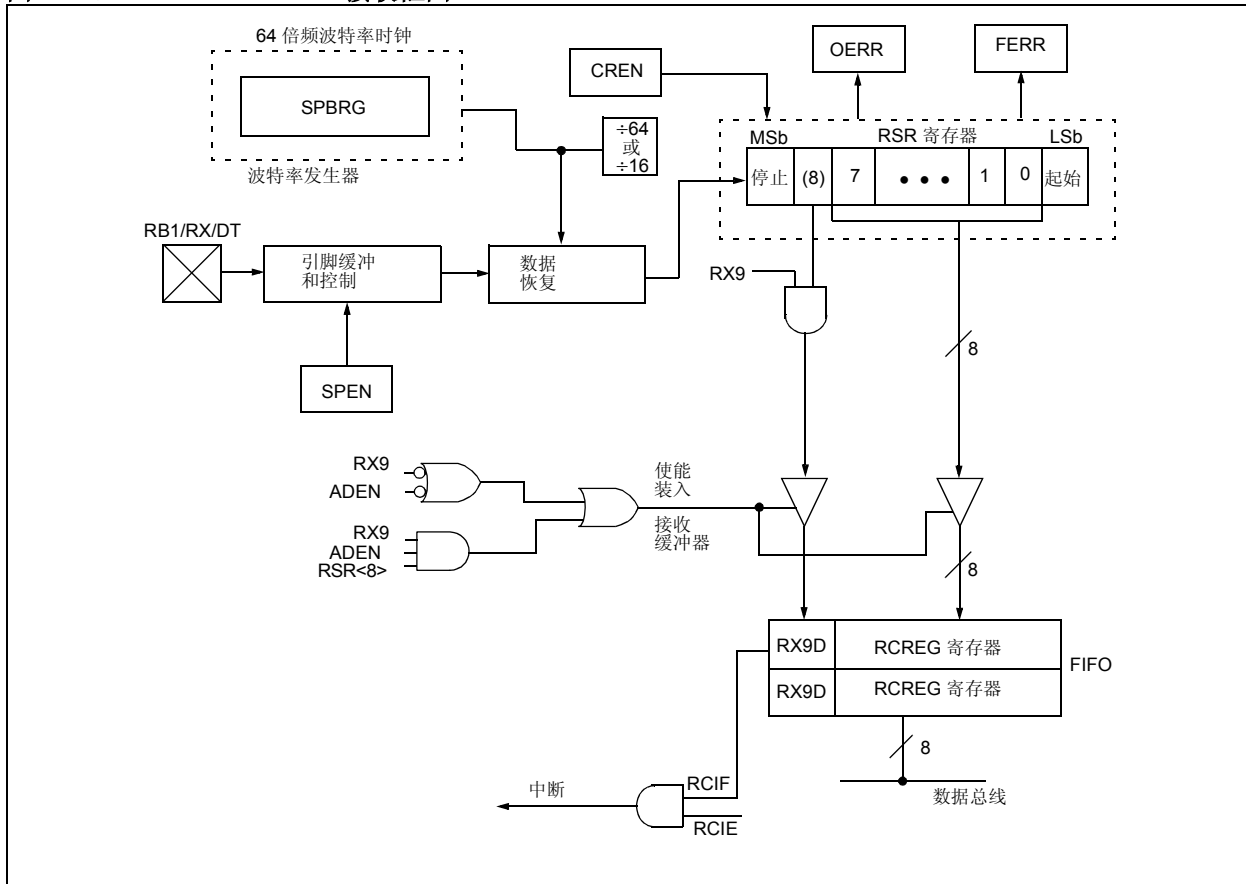
图 12-4 所示为接收器框图。在 RB1/RX/DT 引脚上接收数据，并驱动数据恢复模块。数据恢复模块实际上是一个高速移位器，工作频率为波特率的 16 倍频，但是主接收串行移位寄存器工作频率为位速率或 Fosc。

选择异步模式后，将 CREN 位（RCSTA<4>）置位将使能接收。

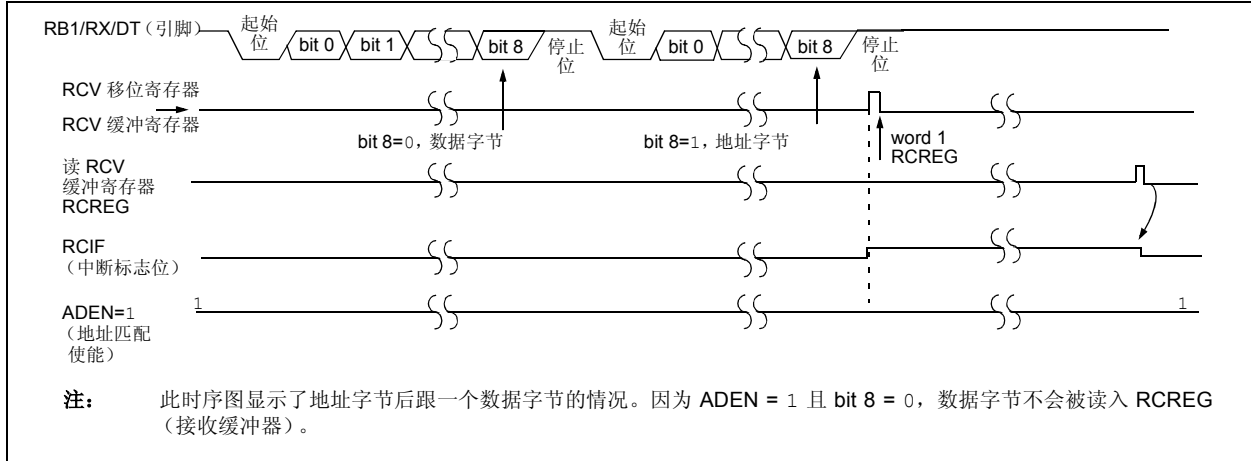
接收器的核心部件是接收（串行）移位寄存器（RSR）。当采样到停止位之后，RSR 中接收到的数据被送到 RCREG 寄存器（如果 RCREG 寄存器为空）。发送完成后，标志位 RCIF（PIR1<5>）会置位。通过置位/清零使能位 RCIE（PIE1<5>），可以允许或禁止该中断。标志位 RCIF 是只读位，由硬件清零。在读 RCREG 寄存器且 RCREG 寄存器为空时，它会被硬件清零。RCREG 寄存器是一个双缓冲寄存器（即两级深度的

FIFO）。因此可以实现接收两个字节的数据并传送到 RCREG FIFO，然后第三个字节开始移位到 RSR 寄存器。在检测到第三个字节的停止位后，如果 RCREG 寄存器仍然是满的，则溢出错误标志位 OERR（RCSTA<1>）会被置位。RSR 寄存器中的数据将会丢失。可以读两次 RCREG 寄存器以重新获得 FIFO 中的两个字节。溢出标志位 OERR 必须用软件清零。这可以通过复位接收逻辑（将 CREN 位清零后再置位）实现。如果 OERR 位被置位，则禁止将 RSR 寄存器中的数据传送到 RCREG 寄存器，因此如果 OERR 位已置位，则必须将它清零。如果检测到停止位为零，则帧出错标志位 FERR（RCSTA<2>）将被置位。FERR 位和接收到的第 9 位以与接收数据相同的方式被缓冲。读 RCREG 会将新值装入 RX9D 和 FERR 位，因此，为了避免丢失 RX9D 和 FERR 位的原值，用户必须在读 RCREG 寄存器前读 RCSTA 寄存器。

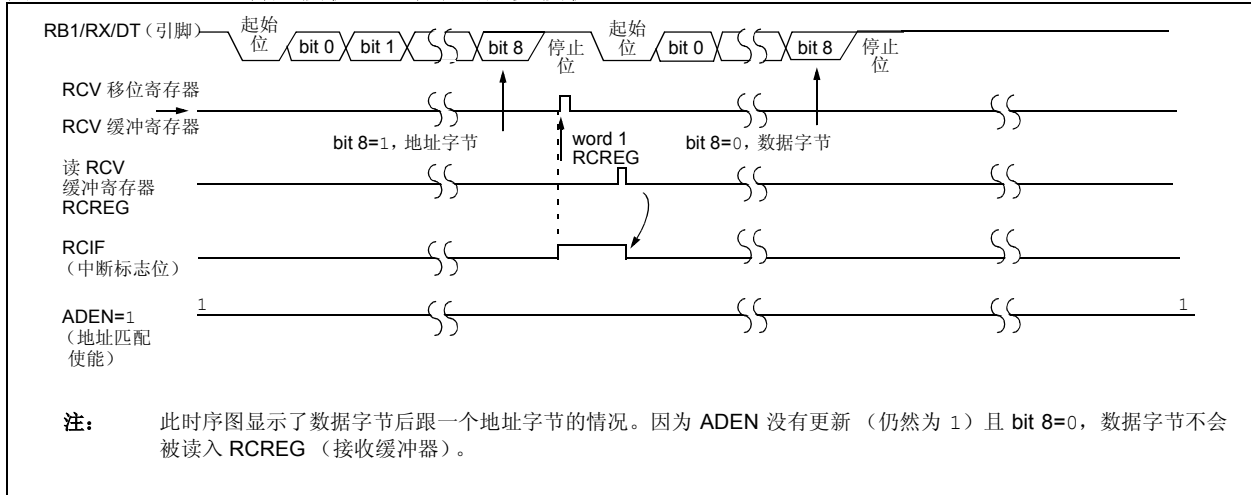
图 12-4: USART 接收框图



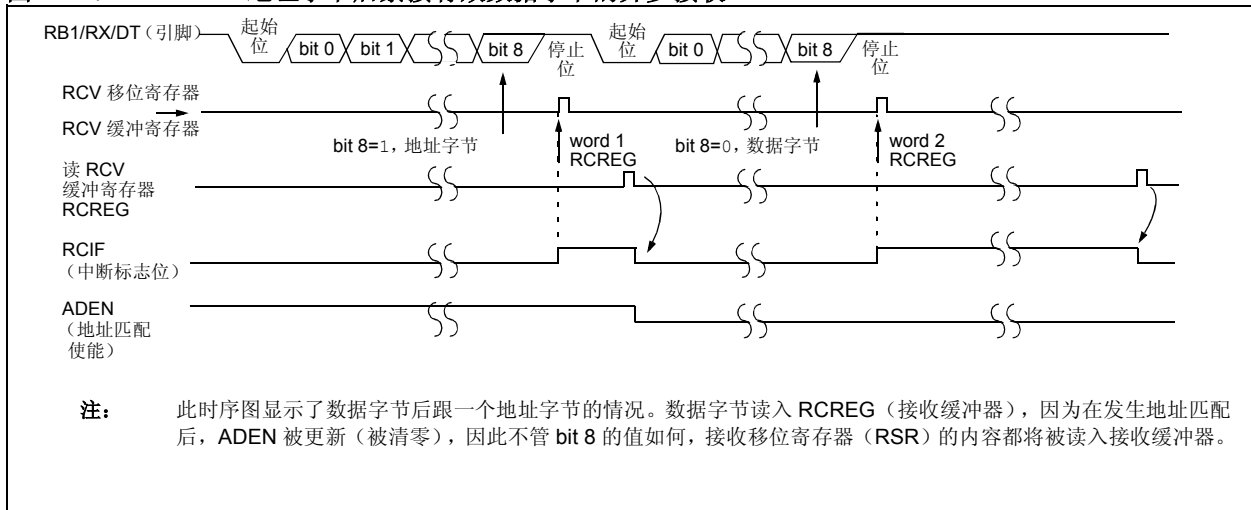
**图 12-5: 带地址检测的异步接收**



**图 12-6: 首先接收地址字节的异步接收**



**图 12-7: 地址字节后紧接有效数据字节的异步接收**



# PIC16F627A/628A/648A

在设置异步接收模式时应遵循以下步骤：

1. 应同时置位 TRISB<1> 位和 TRISB<2> 位，将 RB1/RX/DT 和 RB2/TX/CK 引脚配置为输入。需要时，通过外设电路控制输出驱动。
2. 用正确的波特率初始化 SPBRG 寄存器。如果需要高速波特率，应将 BRGH 位置位。（参见第 12.1 节“USART 波特率发生器”）。
3. 通过将 SYNC 位清零并将 SPEN 位置位，使能异步串行口。
4. 如果需要中断，应将使能位 RCIE 置位。
5. 如果需要接收 9 位数据，则将 RX9 位置位。
6. 通过将 CREN 位置位使能接收。
7. 当接收完成时 RCIF 位将置位，此时如果使能位 RCIE 置位，将会产生一个中断。
8. 读 RCSTA 寄存器获取第 9 位数据（如果已使能），并判断在接收操作中是否发生错误。
9. 通过读RCREG寄存器来读取接收到的8位数据。
10. 如果发生错误，通过将使能位 CREN 清零以清除错误。

表 12-7: 与异步接收有关的寄存器

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART 接收数据寄存器								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	波特率发生器寄存器								0000 0000	0000 0000

图注： x = 未知， — = 未实现，读作 0。阴影部分表示在异步接收模式下未使用。

## 12.3 USART 地址检测功能

### 12.3.1 带地址检测的 USART 9 位接收器

当 RCSTA 寄存器中的 RX9 位置位时，接收到 9 个位，并且第 9 位被置入 RSTA 寄存器中的 RX9D 位。USART 具有多处理器通信的特殊功能。将 ADEN 位 (RCSTA<3>) 与 RX9 位置位可启用多处理器通信。端口目前被这样设置：当接收到最后一位时，接收移位寄存器 (RSR) 的内容移入接收缓冲器中，RSR 的第 9 位 (RSR<8>) 移入 RX9D，并且当且仅当 RSR<8>=1 时允许接收中断。此功能可用于如下多处理器系统：

主处理器希望将一个数据块发送给许多从处理器中的一个。它必须首先发出一个地址字节来标识目标从处理器。地址字节是通过将第 9 位 (RSR<8>) 置 1 标识的 (如果为 0 则是数据字节)。如果从处理器的 RCSTA 寄存器中 ADEN 和 RX9 位被置位，启用多处理器通信，将忽略所有的数据字节。然而，如果接收到的第 9 位为 1，即表示接收到的字节是地址，此时从处理器会被中断并且 RSR 寄存器的内容将被移入接收缓冲器。这使从处理器仅被地址中断，因此，从处理器可以检查接收到的字节以查看自己的地址是否与该地址匹配。地址匹配的从处理器随后将清零它的 ADEN 位，并准备接收来自自主处理器的数据字节。

当 ADEN 使能时 (=1)，所有数据字节都会被忽略。在停止位后的数据将不会被装入接收缓存器，且不会产生中断。如果另一个字节被移入了 RSR 寄存器，那么以前的数据字节将被丢弃。

只有在接收器被配置为 9 位模式 (RX9=1) 时，ADEN 位才起作用。当禁止 ADEN 位 (=0) 时，会接收所有的数据字节，并且第 9 位可作为奇偶校验位。

图 12-4 所示为接收框图。

将 CREN 位 (RCSTA<4>) 置位可启用接收。

#### 12.3.1.1 设置带有地址检测的 9 位模式

在使能地址检测时，要设置异步接收，请遵循以下步骤

1. 应同时置位 TRISB<1> 位和 TRISB<2> 位，将 RB1/RX/DT 和 RB2/TX/CK 引脚配置为输入。需要时，通过外设电路控制输出驱动。
2. 用正确的波特率初始化 SPBRG 寄存器。如果需要高速波特率，应将 BRGH 位置位。
3. 通过将 SYNC 位清零并将 SPEN 位置位，使能异步通信。
4. 如果需要中断，应将使能位 RCIE 置位。
5. 如果需要接收 9 位数据，则将 RX9 位置位。
6. 将 ADEN 置位以启用地址检测。
7. 将 CREN 或 SREN 位置位使能接收。
8. 当接收完成时标志位 RCIF 将置位，此时如果使能位 RCIE 已置位，将会产生一个中断。
9. 通过读 RCREG 寄存器读取 8 位接收到的数据，以判断器件是否正在被寻址。
10. 如果发生 OERR 错误，通过将使能位 CREN 清零 (如果已经置位) 以清除错误。
11. 如果器件被寻址 (RSR<8>=1，使能地址匹配)，将 ADEN 和 RCIF 位清零以允许将数据字节和地址字节读入接收缓冲器并中断 CPU。

表 12-8: 与异步接收有关的寄存器

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART 接收数据寄存器								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	波特率发生器寄存器								0000 0000	0000 0000

图注: x = 未知, — = 未实现, 读作 0。阴影部分表示在异步接收模式下未使用。

## 12.4 USART 同步主模式

在同步主模式下，数据以半双工方式传输（即收发不同步进行）。发送数据时，禁止接收，反之亦然。将 SYNC 位（TXSTA<4>）置位可进入同步模式。此外为了把 RB2/TX/CK 和 RB1/RX/DT I/O 引脚分别配置为时钟线 CK 和数据线 DT，应该将使能位 SPEN（RCSTA<7>）置位。主模式意味着处理器在 CK 时钟线上发送主时钟信号。将 CSRC 位（TXSTA<7>）置位可进入主模式。

### 12.4.1 USART 同步主发送

图 12-1 所示是 USART 发送器的框图。发送器的核心是发送（串行）移位寄存器（TSR）。移位寄存器从读/写发送缓冲寄存器 TXREG 中获取数据。TXREG 寄存器要用软件装入数据。在前一次装入数据的最后一位发送完成前，不会向 TSR 寄存器装入新数据。一旦最后一位发送完成，就会将 TXREG 寄存器中的新数据装入 TSR（如果 TXREG 中有新数据的话）。当 TXREG 寄存器把数据送入 TSR 后（在一个 T 周期内完成），TXREG 就变为空，同时中断标志位 TXIF（PIR1<4>）置位。通过置位/清零使能位 TXIE（PIE1<4>），可以允许或禁止该中断。不管使能位 TXIE 的状态如何，都将置位中断标志位 TXIF，且其不能由软件清零。只有向 TXREG 寄存器装入新数据此位才会复位。TXIF 标志位表明 TXREG 寄存器的状态，而 TRMT 位（TXSTA<1>）表明 TSR 寄存器的状态。TRMT 位是一个只读位，当 TSR 为空时，该位置位。没有任何与 TRMT 位相关的中断逻辑，所以用户必须通过查询 TRMT 位来判断 TSR 寄存器是否为空。TSR 并未映射到数据存储寄存器中，所以用户不能对它进行访问。

通过将使能位 TXEN（TXSTA<5>）置位使能发送。但是直到向 TXREG 寄存器装入数据之后才开始实际的发送。第一位数据将在 CK 线上下一个有效时钟上升沿被移出。数据输出在同步时钟下降沿前后是稳定的（图 12-8）。也可以通过先将发送数据装入 TXREG 寄存器，再将 TXEN 位置位的方式来启动发送（图 12-9）。这在选择低速波特率时是有利的，因为当 TXEN、CREN 和 SREN 位清零时 BRG 保持复位状态。将 TXEN 位置位将启动 BRG，从而立即产生移位时钟。通常，在第一次开始发送时，TSR 寄存器为空，因此送到 TXREG 寄存器的数据会被立即送到 TSR 寄存器，从而导致 TXREG 寄存器为空。可以进行背对背发送。

在发送过程中将 TXEN 位清零会导致发送中止，同时发送器被复位。DT 和 CK 引脚将恢复到高阻态。在发送过程中，无论 CREN 或 SREN 位被置位都会中止发送，同时 DT 引脚恢复到高阻态（准备接收）。如果 CSRC 位被置位（内部时钟），CK 引脚将保持输出状态。虽然发送器的逻辑电路与这两个引脚断开，但它不会复位。要将发送器复位，用户必须将 TXEN 位清零。如果 SREN 位被置位（以便中断正在进行的发送并接收一个字），在收到一个字后，SREN 位会被清零，由于 TXEN 位仍然为置位状态，串口将恢复到发送模式。DT 线将立即从高阻抗接收模式切换到发送模式并开始发送。为避免这种情况，必须将 TXEN 位清零。

为选择 9 位发送模式，TX9 位（TXSTA<6>）应被置位，而且第 9 位数据应当写入 TX9D 位（TXSTA<0>）。必须先写第 9 位数据，然后将 8 位数据写入 TXREG 寄存器。这是因为当 TSR 为空时，向 TXREG 寄存器写数据会导致数据立即送入 TSR 寄存器。如果 TSR 空且在向 TX9D 位写新值之前写 TXREG 寄存器，此时送入 TSR 的第 9 位数据是 TX9D 的“当前”值。

设置同步主发送模式应遵循以下步骤：

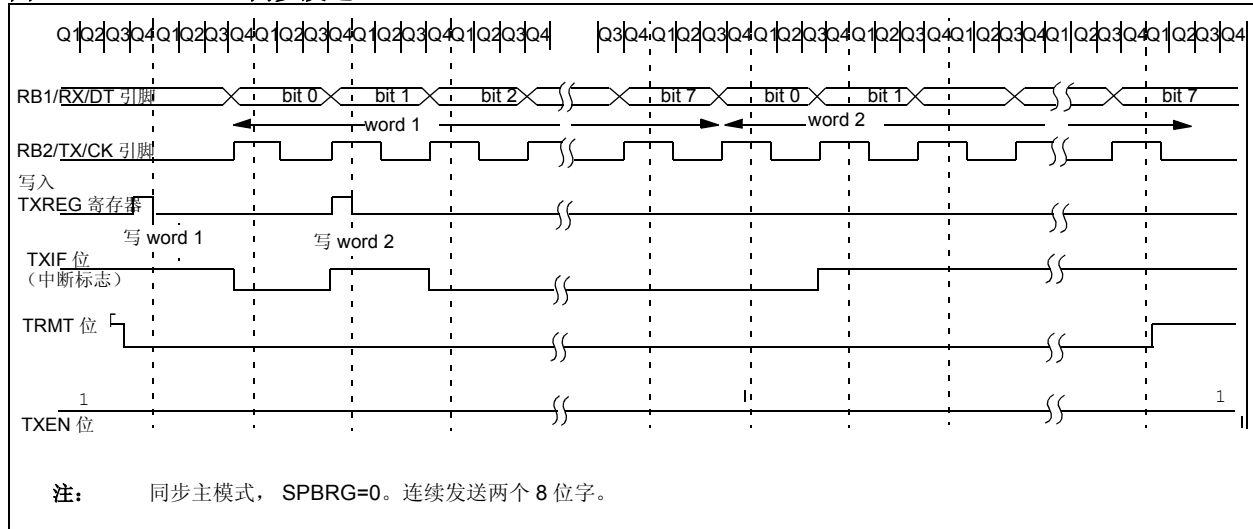
1. 应同时置位 TRISB<1> 位和 TRISB<2> 位，将 RB1/RX/DT 和 RB2/TX/CK 引脚配置为输入。需要时，通过外设电路控制输出驱动。
2. 用正确的波特率初始化 SPBRG 寄存器（参见第 12.1 节“USART 波特率发生器”）。
3. 将 SYNC、SPEN 和 CSRC 位置位，使能同步串行口。
4. 如果需要中断，应将使能位 TXIE 置位。
5. 如果需要发送 9 位数据，应将 TX9 置位。
6. 将 TXEN 位置位以使能发送。
7. 如果选择发送 9 位数据，应该将第 9 位数据装入 TX9D 位。
8. 将数据装入 TXREG 寄存器启动发送。

**表 12-9: 与同步主发送有关的寄存器**

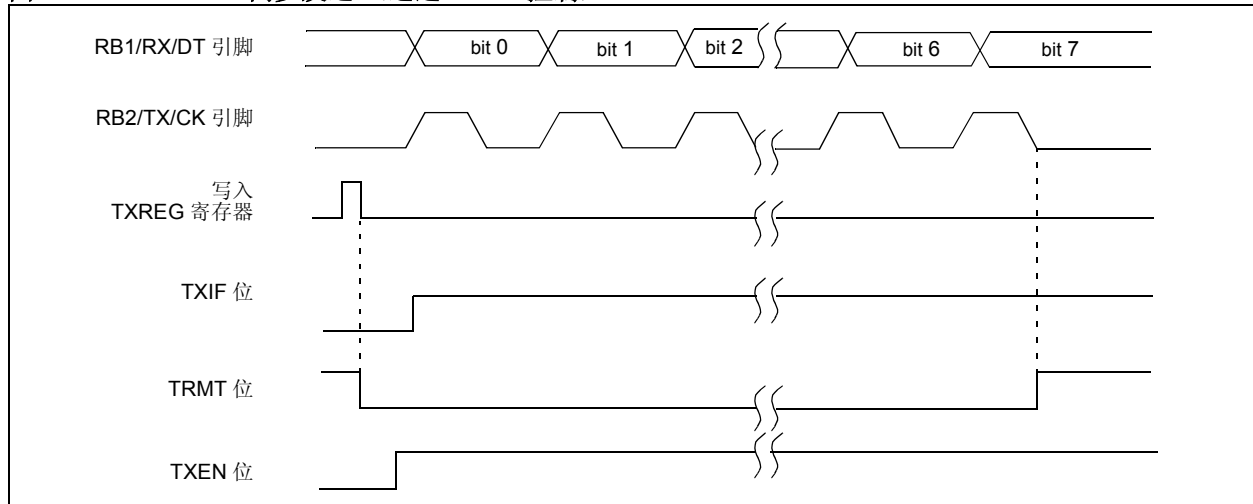
地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	USART 发送数据寄存器								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	波特率发生器寄存器								0000 0000	0000 0000

图注: x= 未知, — = 未实现, 读作 0。阴影部分表示在同步主发送模式下未使用。

**图 12-8: 同步发送**



**图 12-9: 同步发送 (通过 TXEN 控制)**



# PIC16F627A/628A/648A

## 12.4.2 USART 同步主接收模式

一旦选择同步模式后，使能位 SREN 位（RCSTA<5>）和 CREN 位（RCSTA<4>）两者之一被置位都可以使能接收。在时钟的下降沿采样 RB1/RX/DT 引脚上的数据。如果使能位 SREN 已置位，则只接收一个字。如果使能位 CREN 位置位，则可连续接收数据，直到 CREN 位清零。如果两个位同时置位，则 CREN 位优先。采样得到最后的数据后，接收移位寄存器（RSR）中接收到的数据被送到 RCREG 寄存器（如果该寄存器为空）。传送完成后，中断标志位 RCIF（PIR1<5>）会置位。通过置位 / 清零使能位 RCIE（PIE1<5>），可以允许或禁止该中断。RCIF 标志位是只读位，由硬件复位。它在 RCREG 寄存器被读之后或为空时被复位。RCREG 寄存器是一个双缓冲寄存器（即两级深度的 FIFO）。这可以允许在接收到两个字节的的数据并传送给 RCREG FIFO 后，第三个字节再移位到 RSR 寄存器。在接收到第三个字节的最后一位后，如果 RCREG 寄存器仍然是满的，则溢出错误标志位 OERR（RCSTA<1>）会被置位。RSR 寄存器中的数据将会丢失。可以对 RCREG 寄存器读两次，以重新获得 FIFO 中的两个字节。OERR 位必须由软件清零（将 CREN 位清零）。如果 OERR 位置位，则禁止 RSR 中的数据传送到 RCREG 寄存器，因此如果 OERR 位置位，则必须将它清零。接收到的第 9 位数据以与其他接收数据同样的方式被缓冲。读 RCREG 寄存器将会给 RX9D 位装入新值，因此为了不丢失 RX9D 位原来的信息，用户必须在读 RCREG 寄存器之前读 RCSTA 寄存器。

设置同步主接收模式应遵循以下步骤：

1. 应同时置位 TRISB<1> 位和 TRISB<2> 位，将 RB1/RX/DT 和 RB2/TX/CK 引脚配置为输入。需要时，通过外设电路控制输出驱动。
2. 用正确的波特率初始化 SPBRG 寄存器（参见第 12.1 节“USART 波特率发生器”）。
3. 将 SYNC、SPEN 和 CSRC 位置位，使能同步主串行口。
4. 确保将 CREN 和 SREN 位清零。
5. 如果需要中断，应将使能位 RCIE 置位。
6. 如果需要接收 9 位数据，则将 RX9 位置位。
7. 如果需要单字节接收，则将 SREN 位置位；如果需要连续接收，则将 CREN 位置位。
8. 当接收完成时中断标志 RCIF 位将置位，如果使能位 RCIE 置位，则会产生一个中断。
9. 读 RCSTA 寄存器获取第 9 位数据（如果已使能），并判断在接收操作中是否发生错误。
10. 通过读 RCREG 寄存器来读取接收到的 8 位数据。
11. 如果发生 OERR 错误，将 CREN 位清零以清除错误。

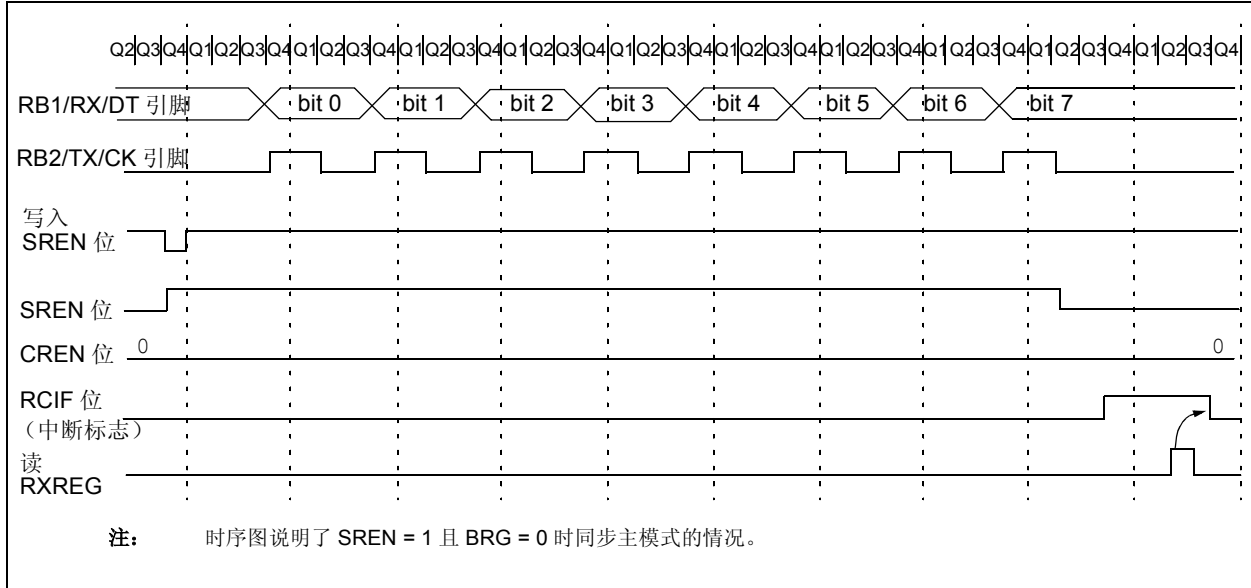
表 12-10: 与同步主接收模式有关的寄存器

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART 接收数据寄存器								0000 0000	0000 0000
8Ch	PIE1	EEPIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	波特率发生器寄存器								0000 0000	0000 0000

图注： x= 未知， — = 未实现，读作 0。阴影部分表示在同步主接收方式中未使用。



图 12-10: 同步主接收模式 (SREN)



## 12.5 USART 同步从模式

同步从模式与主模式的不同之处在于，其移位时钟信号是通过 TX/CK 引脚外部提供的（主模式由内部提供移位时钟）。这就可以让器件在休眠状态下发送或接收数据。将 CSRC 位（TXSTA<7>）清零即可进入从模式。

### 12.5.1 USART 同步从发送

除了休眠模式以外，同步主、从模式的工作原理是相同的。

如果向 TXREG 写入两个字，然后执行 SLEEP 指令，则

- 第一个字立即传送到 TSR 寄存器并发送。
- 第二个字留在 TXREG 寄存器中。
- 标志位 TXIF 不会置位。
- 当第一个字移出 TSR 后，TXREG 寄存器将把第二个字送入 TSR，标志位 TXIF 置位。
- 如果中断允许位 TXIE 为 1，中断将把芯片从休眠状态唤醒，如果允许全局中断，则程序会跳转到中断矢量（0004h）。

设置同步从发送模式应遵循以下步骤：

- 应同时置位 TRISB<1> 位和 TRISB<2> 位，将 RB1/RX/DT 和 RB2/TX/CK 引脚配置为输入。需要时，通过外设电路控制输出驱动。
- 通过将 SYNC 和 SPEN 位置位并清零 CSRC 位，使能同步从串行口。
- 将 CREN 和 SREN 位清零。
- 如果需要中断，应将使能位 TXIE 置位。
- 如果需要发送 9 位数据，应将 TX9 位置位。
- 将使能位 TXEN 位置位以使能发送。
- 如果选择发送 9 位数据，应该将第 9 位数据装入 TX9D 位。
- 将数据装入 TXREG 寄存器启动发送。

# PIC16F627A/628A/648A

## 12.5.2 USART 同步从接收

除了休眠模式以外，同步主、从模式的工作原理是相同的。另外，SREN 位在从模式下未使用。

如果在执行 SLEEP 指令之前已使能接收模式（即将 CREN 位置位），那么在休眠状态下仍可以接收一个数据字。当接收完一个数据字后，RSR 寄存器将把数据送入 RCREG 寄存器，并且如果置位了 RCIE 使能位，将产生中断并将芯片从休眠状态中唤醒。如果允许全局中断，那么程序将跳转至中断矢量（0004h）。

设置同步从接收模式应遵循以下步骤：

1. 应同时置位 TRISB<1> 位和 TRISB<2> 位，将 RB1/RX/DT 和 RB2/TX/CK 引脚配置为输入。需要时，通过外设电路控制输出驱动。
2. 通过将 SYNC 和 SPEN 位置位并清零 CSRC 位，使能同步主串行口。
3. 如果需要中断，应将使能位 RCIE 置位。
4. 如果需要接收 9 位数据，则将 RX9 位置位。
5. 将使能位 CREN 置位以使能接收。
6. 当接收完成时标志位 RCIF 将置位，此时如果使能位 RCIE 置位，将会产生一个中断。
7. 读 RCSTA 寄存器获取第 9 位数据（如果已使能），并判断在接收操作中是否发生错误。
8. 通过读 RCREG 寄存器来读取接收到的 8 位数据。
9. 如果发生 OERR 错误，将 CREN 位清零以清除错误。

表 12-11: 与同步从发送有关的寄存器

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	USART 发送数据寄存器								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	波特率发生器寄存器								0000 0000	0000 0000

图注： x = 未知，— = 未实现，读作 0。阴影部分表示在同步从发送模式下未使用。

表 12-12: 与同步从接收有关的寄存器

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART 接收数据寄存器								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	波特率发生器寄存器								0000 0000	0000 0000

图注： x = 未知，— = 未实现，读作 0。阴影部分表示在同步从接收模式下未使用。

## 13.0 数据 EEPROM 存储器

EEPROM 数据存储器在正常工作状态下（整个 VDD 范围内）是可读写的。这种存储器并不直接映射到文件寄存器空间。而是通过特殊功能寄存器（Special Function Register, SFR）对其进行间接寻址。有 4 个 SFR 用于读写该存储器。它们是：

- EECON1
- EECON2（不是物理实现的寄存器）
- EEDATA
- EEADR

EEDATA 保存要读 / 写的 8 位数据，EEADR 则保存要被寻址的 EEPROM 单元的地址。PIC16F627A/628A 器件具有 128 字节的数据 EEPROM，地址范围从 0h 到 7Fh；PIC16F648A 器件具有 256 字节的数据 EEPROM，地址范围从 0h 到 FFh。

EEPROM 数据存储器允许字节读写。字节写操作会自动擦除目标单元并写入新数据（先擦再写）。EEPROM 数据存储器以其高擦 / 写次数而闻名。写入时间由片上定时器控制。它会随着电压、温度以及具体芯片的不同而变化。如需了解具体的限制，请参见 AC 规范。

当器件被代码保护时，CPU 仍可继续读写数据 EEPROM 存储器。但器件编程器将再也无法访问此存储器。

有关数据 EEPROM 的更多信息，可参见《PICmicro® 中档单片机系列参考手册》（DS33023A\_CN）。

### 寄存器 13-1: EEDATA — EEPROM 数据寄存器（地址：9Ah）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0
bit 7							bit 0

bit 7-0 **EEDATn**: 读取或写入数据 EEPROM 存储器单元的字节值

#### 图注:

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

### 寄存器 13-2: EEADR — EEPROM 地址寄存器（地址：9Bh）

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EADR7	EADR6	EADR5	EADR4	EADR3	EADR2	EADR1	EADR0
bit 7							bit 0

bit 7 **PIC16F627A/628A**  
未实现地址：必须设置为 0

**PIC16F648A**  
 bit 7-0 **EEADR**: 置 1 指定将读写 EEPROM 128 个高地址单元（128 到 255）  
 bit 6-0 **EEADR**: 指定 EEPROM 读写操作的目标地址单元（共有 128 个地址单元）

#### 图注:

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

# PIC16F627A/628A/648A

## 13.1 EEADR

PIC16F648A 的 EEADR 寄存器寻址数据 EEPROM 的 256 字节。寄存器中所有的 8 个位 (EEADR<7:0>) 都是必需的。

PIC16F627A/628A 的 EEADR 寄存器只寻址数据 EEPROM 的前 128 字节，因此只需要使用此寄存器 8 个位中的 7 个 (EEADR<6:0>)。最高位也参加地址译码。这就意味着此位必须始终为 0，以确保地址处于 128 字节存储空间内。

## 13.2 EECON1 和 EECON2 寄存器

EECON1 控制寄存器的低 4 位是物理实现的。其高 4 位为不存在，读作 0。

控制位 RD 和 WR 分别启动读和写。用软件只能将这些位置位而无法清零。在读或写操作完成后，由硬件将它们清零。由于无法用软件将 WR 位清零，从而避免了写操作过早地意外终止。

置位 WREN 位将允许一次写操作。上电时，WREN 位被清零。正常工作时，当写操作被 MCLR 复位或 WDT 超时复位中断时，WRERR 位置位。在这些情况下，复位后用户可以检查 WRERR 位的状态并重写该单元。EEDATA 和 EEADR 寄存器中的数据和地址将保持不变。

写操作完成时，PIR1 寄存器中的中断标志位 EEIF 会置位。此位必须用软件清零。

EECON2 不是物理寄存器。读 EECON2 将返回全 0。EECON2 寄存器仅在数据 EEPROM 写过程中使用。

寄存器 13-3:

EECON1 — EEPROM 控制寄存器 1 (地址: 9Ch) 器件

U-0	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
—	—	—	—	WRERR	WREN	WR	RD
bit 7				bit 0			

bit 7-4 未实现位: 读作 0

bit 3 **WRERR:** EEPROM 错误标志位

1= 提前终止写操作 (由于正常操作过程中的任何 MCLR 复位、WDT 复位或 BOR 复位)  
0= 写操作完成

bit 2 **WREN:** EEPROM 写使能位

1= 允许写周期  
0= 禁止写数据 EEPROM

bit 1 **WR:** 写控制位

1= 开始写周期 (一旦写入完成, 该位将由硬件清零。用软件只能置位 WR 位, 而不能将其清零)。  
0= 数据 EEPROM 写周期完成

bit 0 **RD:** 读控制位

1= 开始读 EEPROM (读取需要一个周期。RD 由硬件清零。用软件只能置位 RD 位, 而不能将其清零)。  
0= 没有启动 EEPROM 读操作

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读作 0

-n = POR 时的值

1 = 置位

0 = 清零

x = 未知

## 13.3 读取 EEPROM 数据存储器

要读取数据存储器单元，用户必须将地址写入 EEDR 寄存器，然后置位控制位 RD (EECON1<0>)。随后的一个周期数据就出现 EEDATA 寄存器中，因此该数据可在下一条指令中读取。EEDATA 将把此值保存至下一次用户向该单元读取或写入数据时（在写操作过程中）为止。

### 例 13-1: 读取数据 EEPROM

```
BSF STATUS, RP0 ;Bank 1
MOVLW CONFIG_ADDR ;
MOVWF EEDR ;Address to read
BSF EECON1, RD ;EE Read
MOVF EEDATA, W ;W = EEDATA
BCF STATUS, RP0 ;Bank 0
```

## 13.4 写入 EEPROM 数据存储器

要写入 EEPROM 数据单元，用户必须首先将地址写入 EEDR 寄存器，并将数据写入 EEDATA 寄存器。然后用户必须按特定顺序开始写入每个字节。

### 例 13-2: 写入数据 EEPROM

```
BSF STATUS, RP0 ;Bank 1
BSF EECON1, WREN ;Enable write
BCF INTCON, GIE ;Disable INTs.
BTFSC INTCON, GIE ;See AN576
GOTO $-2
MOVLW 55h ;
MOVWF EECON2 ;Write 55h
MOVLW AAh ;
MOVWF EECON2 ;Write AAh
BSF EECON1, WR ;Set WR bit
```

如果未完全按照上例中的顺序（先将 55h 写入 EECON2，再将 AAh 写入 EECON2，最后将 WR 位置 1）逐字节写入，写操作将不会开始。强烈建议在这一代码段中禁止中断。在执行必需的指令序列时也会执行周期计数。如果周期计数与执行必需指令序列所需的周期数不等，那么数据将不会写入 EEPROM。

此外，必须将 EECON1 中的 WREN 位置位以启用写操作。这种机制可防止由于执行错误（不期望的）代码（例如程序跑飞）导致误写入 EEPROM。除了更新 EEPROM 时以外，用户应该始终保持 WREN 位清零。硬件不能清零 WREN 位。

一个写过程开始后，将 WREN 位清零将不会影响此写周期。除非 WREN 位置位，否则 WR 位将无法置位。

写周期完成时，WR 位由硬件清零，并且 EE 写完成中断标志位（EE Write Complete Interrupt Flag bit, EEIF）置位。用户可以使能此中断或查询此位。PIR1 寄存器的 EEIF 位必须用软件清零。

## 13.5 写校验

视应用而定，好的编程习惯可能规定将写入数据 EEPROM 的值对照期望值进行校验（例 13-3）。当 EEPROM 的数据位擦写次数接近规范极限值的应用中，就应进行写校验。

### 例 13-3: 写校验

```
BSF STATUS, RP0 ;Bank 1
MOVF EEDATA, W
BSF EECON1, RD ;Read the
;value written
;
;Is the value written (in W reg) and
;read (in EEDATA) the same?
;
SUBWF EEDATA, W ;
BTFSS STATUS, Z ;Is difference 0?
GOTO WRITE_ERR ;NO, Write error
: ;YES, Good write
: ;Continue program
```

## 13.6 防止误写

有些情况下，不宜向数据 EEPROM 写入数据。为了防止误写 EEPROM，器件内建了各种保护机制。在上电时，WREN 位被清零。同时，如果使能了上电延时定时器（持续时间为 72 ms），也可防止误写 EEPROM。

在掉电、电源毛刺或软件故障期间，写操作启动顺序和 WREN 位可共同防止发生误写操作。

# PIC16F627A/628A/648A

## 13.7 使用数据 EEPROM

数据 EEPROM 是高耐用性，通过字节寻址的阵列，优化为用来保存经常变更的信息（例如，程序变量或其他经常更新的数据）。如果一个段中的变量经常发生改变，而另一个段中的变量不发生改变，就可能造成对整个 EEPROM 的总擦写次数超出规范（参数 D124），而对某个字节的总擦写次数不会超出规范（参数 D120 和 D120A）如果是这样，必须执行一次阵列刷新。因此，应将不经常改变的变量（如常量、ID 和校准值等）存储到闪存程序存储器中。

例 13-4 所示为简单的数据 EEPROM 刷新程序。

**注：** 如果数据 EEPROM 仅用于存储常量和 / 或很少修改的数据，可能就不需要进行阵列刷新。请参见规范 D124。

### 例 13-4: 数据 EEPROM 刷新程序

```
BANKSEL    0X80           ;select Bank1
CLRF      EEADR          ;start at address 0
BCF       INTCON, GIE    ;disable interrupts
BTFSC    INTCON, GIE     ;see AN576
GOTO     $ - 2
BSF      EECON1, WREN    ;enable EE writes
Loop
BSF      EECON1, RD      ;retrieve data into EEDATA
MOVLW   0x55             ;first step of ...
MOVWF   EECON2           ;... required sequence
MOVLW   0xAA             ;second step of ...
MOVWF   EECON2           ;... required sequence
BSF     EECON1, WR       ;start write sequence
BTFSC   EECON1, WR      ;wait for write complete
GOTO    $ - 1

#IFDEF   __16F648A       ;256 bytes in 16F648A
    INCFSZ    EEADR, f    ;test for end of memory
#ELSE
    INCF     EEADR, f     ;next address
    BTFSS    EEADR, 7    ;test for end of memory
#ENDIF

GOTO    Loop            ;repeat for all locations
```

## 13.8 代码保护时的数据 EEPROM 操作

当器件代码保护时，CPU 仍能够读写数据 EEPROM。

**表 13-1: 与数据 EEPROM 相关的寄存器 / 位**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	上电复位时的值	所有其他复位时的值
9Ah	EEDATA	EEPROM 数据寄存器								xxxx xxxx	uuuu uuuu
9Bh	EEADR	EEPROM 地址寄存器								xxxx xxxx	uuuu uuuu
9Ch	EECON1	—	—	—	—	WRERR	WREN	WR	RD	---- x000	---- q000
9Dh	EECON2 <sup>(1)</sup>	EEPROM 控制寄存器 2								---- ----	---- ----

图注: x = 未知, u = 未改变, — = 未实现, 读作 0, q = 值随条件变化。  
 阴影单元表示未被数据 EEPROM 使用。

**注 1:** EECON2 不是物理存在的寄存器。

# PIC16F627A/628A/648A

---

---

注:



## 14.0 CPU 的特殊功能

单片机内部集成了处理实时应用需要的特殊电路，这也是单片机与其他处理器的区分。PIC16F627A/628A/648A 系列具有以下功能，旨在最大限度地提高系统可靠性，通过避免使用外部元器件将成本降至最低，提供低功耗工作模式以及代码保护功能。

这些功能包括：

1. 振荡源选择
2. 复位
3. 上电复位 (POR)
4. 上电延时定时器 (PWRT)
5. 振荡器起振定时器 (Oscillator Start-Up Timer, OST)
6. 欠压复位 (BOR)
7. 中断
8. 看门狗 (WDT)
9. 休眠
10. 代码保护
11. ID 单元
12. 在线串行编程 (In-Circuit Serial Programming™, ICSP™)

PIC16F627A/628A/648A 配备了一个看门狗，由配置位来控制。看门狗依靠自身的 RC 振荡器来运行，增加了可靠性。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)，旨在确保芯片在晶振达到稳定前处于复位状态。另外一个为上电延时定时器 (PWRT)，仅在上电时提供 72 ms (标称值) 的固定延时，确保器件在电源电压稳定之前处于复位状态。此外，还有专门的电路在发生欠压时，使器件复位。由于片上已具有这三种功能，对大多数应用而言，不需要外部复位电路。

休眠模式是为提供一种电流消耗很低的掉电工作模式而设计的。用户可通过外部复位、看门狗唤醒或中断将器件从休眠状态唤醒。还有几种振荡模式供选择，使器件能符合不同应用的需要。选择 RC 振荡模式可节省系统的成本，而选择 LP 晶振模式可降低系统功耗。可以使用一组配置位来选择不同的模式。

## 14.1 配置位

可以通过对配置位编程 (读作“0”)或不编程 (读作“1”)来选择不同的器件配置。这些配置位映射到程序存储器地址 2007h。

用户会注意到地址 2007h 超出了用户程序存储空间范围。事实上，它属于特殊配置存储空间 (2000h - 3FFFh)，仅可在编程时对其进行访问。更多信息参见 *PIC16F627A/628A/648A EEPROM Memory Programming Specification* (DS41196)。

# PIC16F627A/628A/648A

## 寄存器 14-1: CONFIG — 配置字寄存器

CP	—	—	—	—	CPD	LVP	BOREN	MCLR $\overline{E}$	FOSC2	$\overline{PWRTE}$	WDTE	FOSC1	FOSC0
bit 13													bit 0

bit 13: **CP**: 闪存程序存储器代码保护位<sup>(2)</sup>  
(PIC16F648A)  
1= 代码保护关闭  
0=0000h 到 0FFFh 有代码保护  
(PIC16F628A)  
1= 代码保护关闭  
0=0000h 到 07FFh 有代码保护  
(PIC16F627A)  
1= 代码保护关闭  
0=0000h 到 03FFh 有代码保护

bit 12-9: **未实现位**: 读作 0

bit 8: **CPD**: 数据代码保护位<sup>(3)</sup>  
1= 数据存储器代码保护关闭  
0= 数据存储器有代码保护

bit 7: **LVP**: 低电压编程使能位  
1=RB4/PGM 引脚具有 PGM 功能, 低电压编程使能。  
0=RB4/PGM 为数字 I/O, 如要编程, 必须在 MCLR 引脚加高电压。

bit 6: **BOREN**: 欠压复位使能位<sup>(1)</sup>  
1= 使能 BOR 复位  
0= 禁止 BOR 复位

bit 5: **MCLR $\overline{E}$** : RA5/MCLR/VPP 引脚功能选择位  
1=RA5/MCLR/VPP 引脚功能为 MCLR  
0=RA5/MCLR/VPP 引脚功能为数字输入, MCLR 内部连接到 VDD

bit 3:  **$\overline{PWRTE}$** : 上电延时定时器使能位<sup>(1)</sup>  
1= 禁止 PWRT  
0= 使能 PWRT

bit 2: **WDTEN**: 看门狗使能位  
1= 使能 WDT  
0= 禁止 WDT

bit 4, 1-0: **FOSC<2:0>**: 振荡模式选择位<sup>(4)</sup>  
111= RC 振荡模式: RA6/OSC2/CLKOUT 引脚为 CLKOUT 功能, RA7/OSC1/CLKIN 引脚连接电阻和电容  
110= RC 振荡模式: RA6/OSC2/CLKOUT 引脚为 I/O 功能, RA7/OSC1/CLKIN 引脚连接电阻和电容  
101= INTOSC 振荡模式: RA6/OSC2/CLKOUT 引脚为 CLKOUT 功能, RA7/OSC1/CLKIN 引脚为 I/O 功能  
100= INTOSC 振荡模式: RA6/OSC2/CLKOUT 引脚为 I/O 功能, RA7/OSC1/CLKIN 引脚为 I/O 功能  
011= EC 振荡模式: RA6/OSC2/CLKOUT 引脚为 I/O 功能, RA7/OSC1/CLKIN 引脚为 CLKIN 功能。  
010= HS 振荡模式: RA6/OSC2/CLKOUT 和 RA7/OSC1/CLKIN 引脚接高速晶振 / 谐振器  
001= XT 振荡模式: RA6/OSC2/CLKOUT 和 RA7/OSC1/CLKIN 引脚接晶振 / 谐振器  
000= LP 振荡模式: RA6/OSC2/CLKOUT 和 RA7/OSC1/CLKIN 引脚接低功耗晶振

- 注
- 1: 使能欠压复位不会像在 PIC16F627/628 中一样自动使能上电延时定时器 (PWRT)。
  - 2: 该系列对代码保护方案作了修改, 不同于 PIC16F627/628 原来所用的代码保护方案。为了将 CP 位置 1 (关闭代码保护), 需要批量擦除整个闪存程序存储器。详情参见 *PIC16F627A/628A/648A EEPROM Memory Programming Specification* (DS41196)。
  - 3: 为了将 CPD 位置 1 (关闭代码保护), 需要批量擦除整个数据 EEPROM。详情参见 *PIC16F627A/628A/648A EEPROM Memory Programming Specification* (DS41196)。
  - 4: 如果 MCLR 在 INTOSC 模式下被拉为低电平, 则内部时钟振荡器将被禁止。

### 图注:

R= 可读位	W= 可写位	U= 未实现位, 读作 “0”
-n=POR 时的值	1= 置位	0= 清零
		x= 未知

## 14.2 振荡器配置

### 14.2.1 振荡器类型

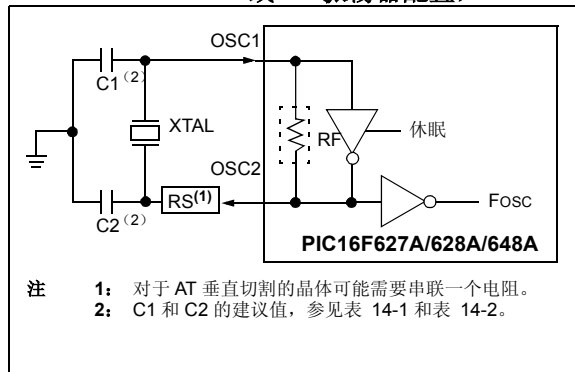
PIC16F627A/628A/648A 可工作在 8 种不同的振荡模式。用户可以对三个配置位（FOSC2 到 FOSC0）编程来选择其中任何一种模式。

- LP 低功耗晶振
- XT 晶振 / 谐振器
- HS 高速晶振 / 谐振器
- RC 外部电阻 / 电容（2 种模式）
- INTOSC 高精度内部振荡器（2 种模式）
- EC 外部时钟输入

### 14.2.2 晶振 / 陶瓷谐振器

在 XT、LP 或 HS 模式下，OSC1 和 OSC2 引脚连接一个晶振或陶瓷谐振器以产生振荡（图 14-1）。PIC16F627A/628A/648A 振荡器的设计要求使用平行切割的晶体。而采用顺序切割的晶体，可能使振荡器产生的频率超出晶体制造商给出的范围。在 XT、LP 或 HS 模式下，器件可用一个外部时钟源来驱动 OSC1 引脚（图 14-4）。

**图 14-1:** 晶振工作原理  
(或陶瓷谐振器) (HS、XT 或 LP 振荡器配置)



**表 14-1:** 陶瓷谐振器的电容选择

模式	频率	OSC1(C1)	OSC1(C2)
XT	455 kHz	22 - 100 pF	22 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	10 - 68 pF	10 - 68 pF
	16.0 MHz	10 - 22 pF	10 - 22 pF

**注:** 采用较大的电容虽然可以提高振荡器的稳定性，但是同时会延长振荡器的起振时间。这些值仅供设计参考。由于谐振器的特性各不相同，因此用户应向谐振器厂商咨询外部元件的正确参数。

**表 14-2:** 晶振的电容选择

模式	频率	OSC1(C1)	OSC1(C2)
LP	32 kHz	15 - 30 pF	15 - 30 pF
	200 kHz	0 - 15 pF	0 - 15 pF
XT	100 kHz	68 - 150 pF	150 - 200 pF
	2 MHz	15 - 30 pF	15 - 30 pF
	4 MHz	15 - 30 pF	15 - 30 pF
HS	8 MHz	15 - 30 pF	15 - 30 pF
	10 MHz	15 - 30 pF	15 - 30 pF
	20 MHz	15 - 30 pF	15 - 30 pF

**注:** 采用较大的电容值虽然可以提高振荡器的稳定性，但是同时会延长振荡器的起振时间。这些值仅供设计参考。为避免对要求低驱动电压的晶振造成过驱动，在 HS 和 XT 模式下，可能需要串联电阻（RS）。由于晶振的特性各不相同，因此用户应向晶振厂商咨询外部元件的正确参数。

# PIC16F627A/628A/648A

## 14.2.3 外部晶振电路

既可以使用现成的振荡器，也可构建一个 TTL 的简单振荡电路。现成的振荡器有较宽的工作范围和更好的稳定性。而一个设计良好的晶振可以通过 TTL 门提供优良的性能。可使用两种类型的晶振电路：串联谐振电路或并联谐振电路。

图 14-2 为并联谐振振荡电路的实现原理。该电路设计为使用晶振的基频。74AS04 反相器提供并联振荡器所需的  $180^\circ$  相移。4.7k $\Omega$  电阻则提供负反馈来提高电路稳定性。10k $\Omega$  电位器将 74AS04 偏置在线性工作区。这可用于外部振荡器的设计。

图 14-2: 外部并联谐振器 / 晶振的振荡电路

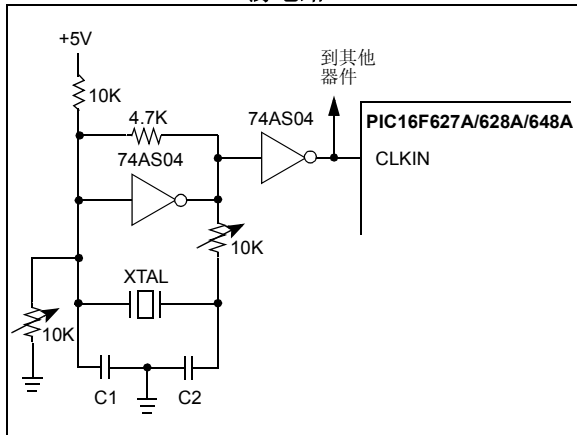
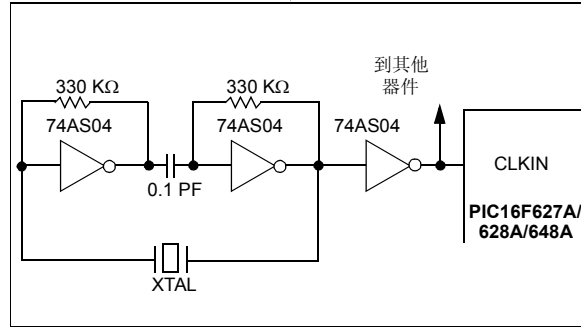


图 14-3 为串联谐振振荡电路。该电路也设计为使用晶振的基频。反相器为串联谐振振荡电路提供  $180^\circ$  相移。330 k $\Omega$  电阻提供负反馈，将反相器偏置在线性工作区。

图 14-3: 外部串联谐振 / 晶振的振荡电路



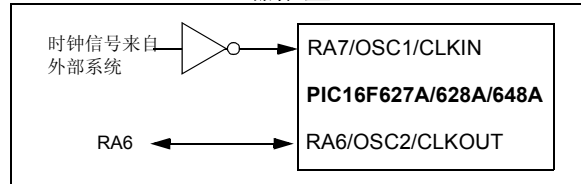
## 14.2.4 高精度 4 MHz 内部振荡器

在  $V_{DD} = 5\text{ V}$ 、 $25^\circ\text{C}$  条件下，高精度内部振荡器提供一个固定的 4 MHz（标称值）系统时钟。其振荡频率随工作电压和温度变化的信息，参见第 17.0 节“电气规范”。

## 14.2.5 外部时钟输入

对于已经有了外部时钟信号的应用，如果外部时钟源满足第 17.6 节“时序图和规范”中列出的 AC/DC 时序要求，则用户可以使用它来直接驱动 PIC16F627A/628A/648A。下面的图 14-4 说明了应当如何配置外部时钟电路。

图 14-4: 外部时钟输入工作原理（EC、HS、XT 或 LP 振荡器配置）



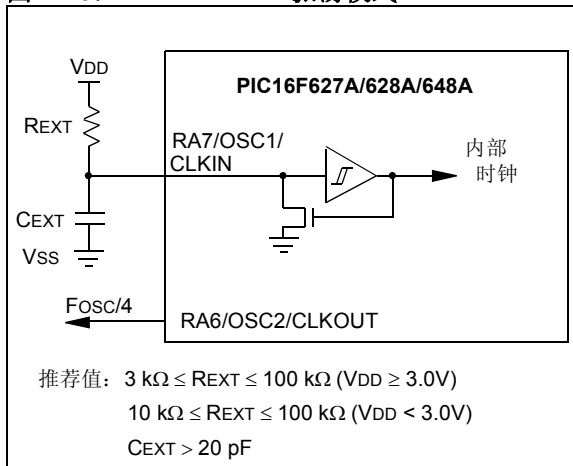
## 14.2.6 RC 振荡器

对于不需要精确时序的应用，可选用 RC 振荡器。RC 振荡器的工作和功能取决于几个变量。RC 振荡器频率随以下几个值变化：

- 电源电压
- 电阻（REXT）和电容（CEXT）值
- 工作温度。

由于正常的制造工艺参数的差异，每个器件的振荡频率也会有所不同。而各种封装类型的引线电容不同，也会影响振荡频率，特别是当 CEXT 值较小时。用户还需要考虑外部元件（电阻 R 和电容 C）的容差。图 14-5 显示了如何外接 R/C 电路。

图 14-5: RC 振荡模式



RC 振荡模式对于未使用的 OSC2 引脚有两个控制选择。第一个选择允许将它用作通用 I/O 端口。另一个选择把该引脚配置为输出端，为测试或外部同步提供 Fosc 信号（内部时钟的四分频）。

## 14.2.7 CLKOUT

PIC16F627A/628A/648A 可以通过对配置字编程，使器件提供时钟输出信号。该振荡器频率的四分频可用于检测或者同步其他逻辑电路。

## 14.2.8 特殊功能：双速振荡器模式

当 PIC16F627A/628A/648A 配置为 INTOSC 振荡模式时，可使用可软件编程的双速振荡器模式。在 INTOSC 模式下，该功能使用户能够动态地在 4 MHz 与 48 kHz（标称值）之间切换振荡器速度。对于要求低电流消耗，却不允许器件进入休眠状态的应用，可以使用该模式。

当在快速振荡和慢速振荡之间进行切换时，存在延迟。振荡器速度转换延迟包括两个现有时钟脉冲和八个新的速度时钟脉冲。在时钟速度转换延迟期间，系统时钟暂停工作，使单片机也及时地停止工作。延迟期间，程序计数器和时钟输出 CLKOUT 停止。

PCON 寄存器中的 OSCF 位用来控制双速模式。参见第 4.2.2.6 节“PCON 寄存器”，寄存器 4-6。

## 14.3 复位

PIC16F627A/628A/648A 有以下几种复位方式：

- 上电复位（POR）
- 正常工作期间的 MCLR 复位
- 休眠模式下的 MCLR 复位
- WDT 复位（正常工作状态）
- WDT 唤醒（休眠状态）
- 欠压复位（BOR）

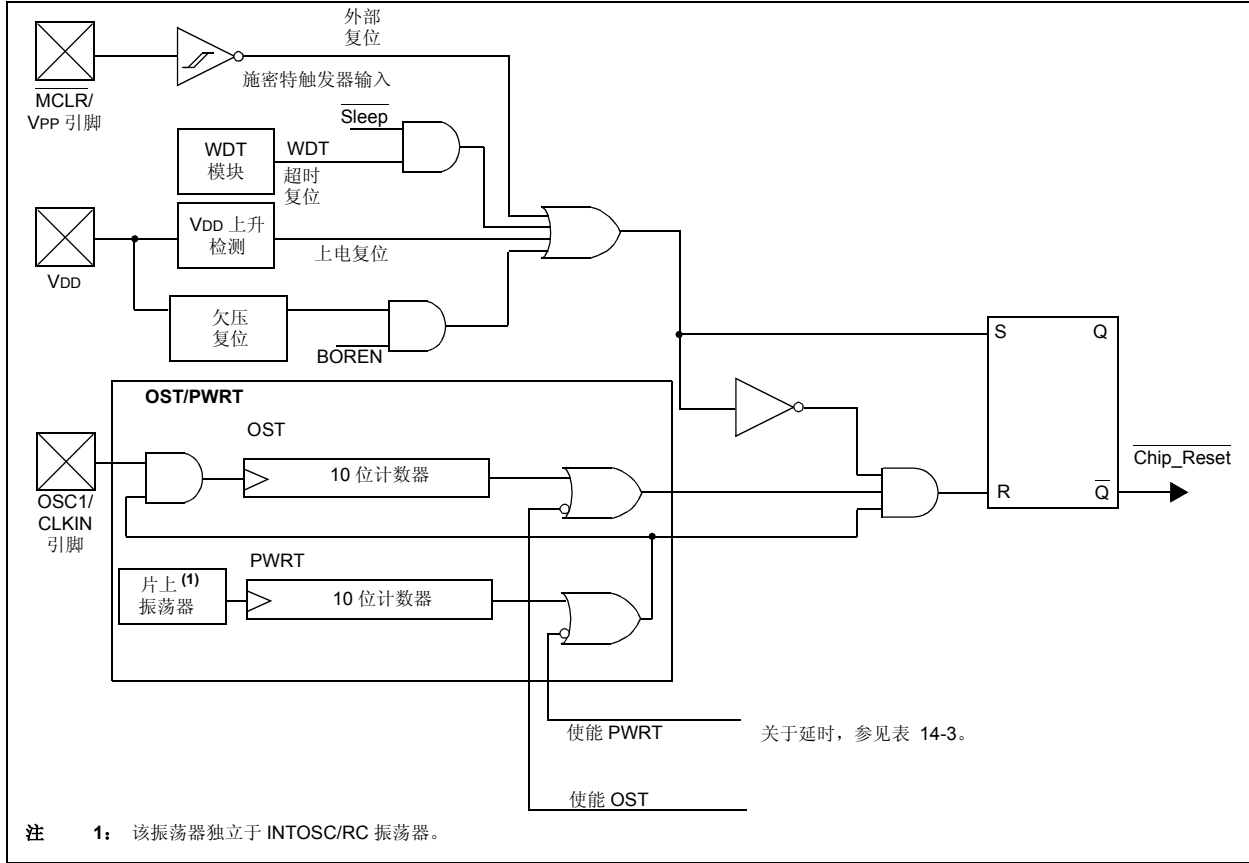
有些寄存器不受任何复位的影响；在 POR 时它们的状态未知，而在其他复位时状态不改变。而大多数其他的寄存器在上电复位、欠压复位、MCLR 复位、WDT 复位以及休眠期间的 MCLR 复位发生时，复位到“复位状态”。它们不受 WDT 唤醒的影响，因为这被视为恢复正常工作。如表 14-4 所示，TO 和 PD 状态位在不同的复位情况下会分别被置位或清零。这些状态位在软件中用于判断复位的性质。表 14-7 对所有寄存器的复位状态作了完整的介绍。

图 14-6 显示了一个片上复位电路的简化框图。

在 MCLR 复位信号传输路径上有一个噪声滤波器，它可以检测和滤除小脉冲的干扰。关于脉宽规范，参见表 17-7。

# PIC16F627A/628A/648A

图 14-6: 片上复位电路的简化框图



## 14.4 上电复位 (POR)、上电延时定时器 (PWRT)、振荡器起振定时器 (OST) 和欠压复位 (BOR)

### 14.4.1 上电复位 (POR)

片上 POR 电路使器件保持复位状态，直到检测到 VDD 上升（到 1.2-1.7V 范围内）。VDD 上升时间不应超过最大上升时间，参见电气规范。

当 VDD 降低时，POR 电路不会产生内部复位。

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

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

### 14.4.2 上电延时定时器 (PWRT)

上电复位 (POR) 或欠压复位 (若使能) 时，PWRT 提供一个 72 ms（标称值）的固定延时。PWRT 以内部 RC 振荡器为工作时钟。只要 PWRT 有效，器件就保持在复位状态。PWRT 延时可以使 VDD 上升到适当的电平。配置位 **PWRTE** 可以禁止（如果被置位）或使能（如果被清零或编程）PWRT。建议在欠压复位使能时，使能 PWRT。

由于 VDD、温度和制造工艺的差异，不同器件的上电延时时间也会有所不同。详细信息，参见 DC 参数（表 17-7）。

### 14.4.3 振荡器起振定时器 (OST)

当 PWRT 延时结束后，OST 提供一个 1024 个振荡周期（从 OSC1 输入）的延时。直到 OST 延时完成，才开始执行程序。这可以确保晶振或谐振器已经起振并且稳定。

只有在 XT、LP 和 HS 振荡模式下，并且只有在上电复位或器件从休眠状态唤醒时，才启动 OST 延时。参见表 17-7。

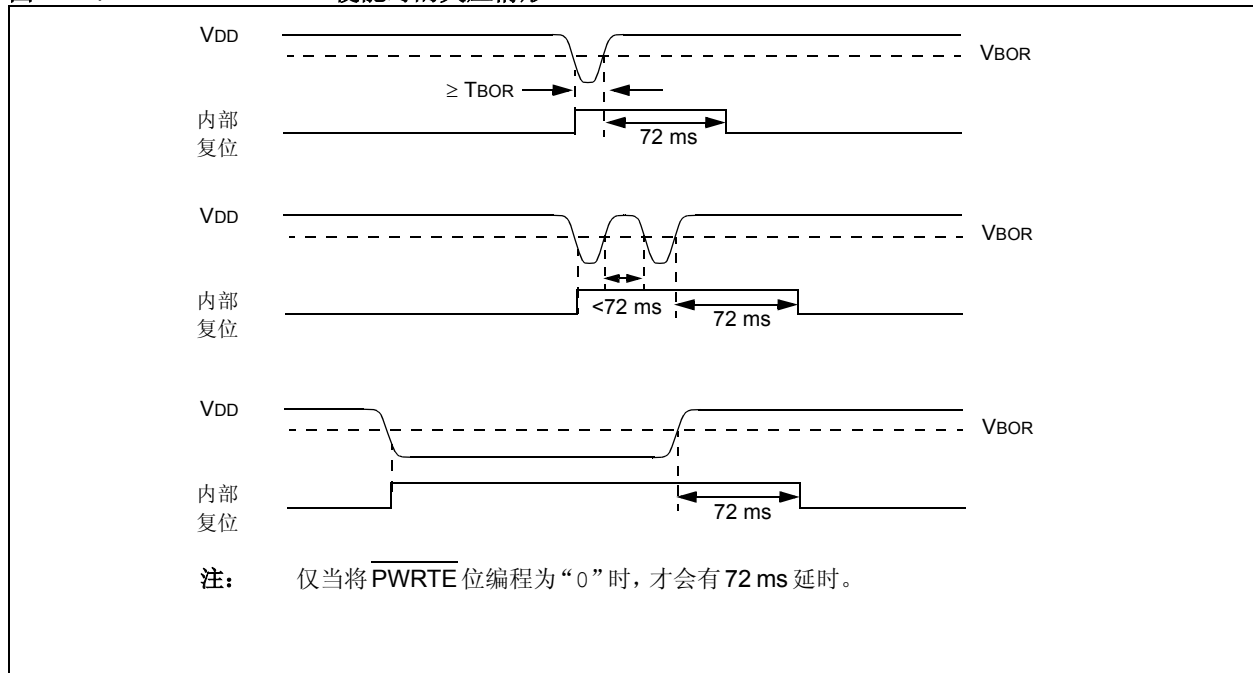
### 14.4.4 欠压复位 (BOR)

PIC16F627A/628A/648A 具有片上 BOR 电路。配置位 **BOREN** 可以禁止（如果被清零或编程）或使能（如果被置位）BOR 电路。如果 VDD 下降到 **VBOR** 以下的持续时间超过 **TBOR**，欠压条件将使器件复位。如果 VDD 低于 **VBOR** 的时间少于 **TBOR**，不能保证是否会发生复位。在表 17-2 和表 17-7 中分别定义了 **VBOR** 和 **TBOR**。

任何复位（上电复位、欠压复位、看门狗复位等）发生时，器件将保持复位状态直到 VDD 上升到 **VBOR** 以上。（参见图 14-7）。如果使能了上电延时定时器，此时它将启动，并且会使器件保持复位状态的时间延长 72 ms。

如果上电延时定时器运行时，VDD 下降到 **VBOR** 以下，芯片将重新回到欠压复位状态，且上电延时定时器将被重新初始化。一旦 VDD 上升到 **VBOR** 以上，上电延时定时器将执行一个 72 ms 的延时复位。图 14-7 所示为典型欠压情形。

图 14-7: PWRT 使能时的欠压情形



# PIC16F627A/628A/648A

## 14.4.5 延时过程

上电时的延时过程为：首先，POR 结束后启动 PWRT 延时。然后，OST 被激活。总延时时间取决于振荡器的配置和 PWRTE 位的状态。例如，在 RC 模式且 PWRTE 位置 1（PWRT 禁止）的情况下，根本不会出现延时。图 14-8、图 14-9 和图 14-10 说明了延时过程。

POR 脉冲引发延时，因此若 MCLR 引脚保持足够长时间的低电平，延时将结束。拉高 MCLR 将使程序立即执行（参见图 14-9）。这对于测试或同步多个并行工作的 PIC16F627A/628A/648A 器件非常有用。

表 14-6 给出了一些特殊寄存器的复位状态，而表 14-7 给出了所有寄存器的复位状态。

## 14.4.6 电源控制（PCON）状态寄存器

电源控制（Power Control, PCON）/状态寄存器 PCON（地址 8Eh）包含两个位。

bit 0 是 BOR（欠压复位）位。上电复位后，BOR 的状态未知。然后，用户必须将该位置 1，并在随后的复位发生时检查 BOR 是否为 0，如果是，则表示已经发生过欠压复位。如果禁止欠压电路（通过在配置字中设置 BOREN 位 = 0），则不必关心 BOR 状态位的取值，也没有必要预见该状态位。

bit 1 是 POR（上电复位）位。该位在上电复位时清零，并且不受其他条件影响。用户必须在上电复位之后，将 1 写入该位。当随后的复位发生时，如果 POR 为 0，则表示一定发生过上电复位（VDD 可能曾经处于过低的状态）。

表 14-3: 不同情况下的延时

振荡器配置	上电延时定时器		欠压复位		从休眠状态唤醒
	PWRTE=0	PWRTE=1	PWRTE=0	PWRTE=1	
XT、HS、LP	72 ms+ 1024•Tosc	1024•Tosc	72 ms+ 1024•Tosc	1024•Tosc	1024•Tosc
RC, EC	282.8 ms	—	282.8 ms	—	—
INTOSC	72 ms	—	72 ms	—	6 μs

表 14-4: 状态 /PCON 位及其含义

POR	BOR	TO	PD	条件
0	X	1	1	上电复位
0	X	0	X	非法，POR 时 TO 置位
0	X	X	0	非法，POR 时 PD 置位
1	0	X	X	欠压复位
1	1	0	u	WDT 复位
1	1	0	0	WDT 唤醒
1	1	u	u	正常工作期间 MCLR 复位
1	1	1	0	休眠状态下 MCLR 复位

图注： u = 未改变， x = 未知



# PIC16F627A/628A/648A

**表 14-5: 与欠压复位相关的寄存器综述**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 复位时的值	所有其他复位时的值 <sup>(1)</sup>
03h, 83h, 103h, 183h	STATUS	IRP	RP1	RPO	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
8Eh	PCON	—	—	—	—	OSCF	—	$\overline{POR}$	$\overline{BOR}$	---- 1-0x	---- u-uq

**图注:** x = 未知, u = 未改变, — = 未实现, 读作 0, q = 值随条件变化。  
阴影部分表示在欠压复位中未使用。

**注 1:** 正常工作中的其他复位 (非上电复位) 包括  $\overline{MCLR}$  复位、欠压复位和看门狗复位。

**表 14-6: 特殊寄存器的初始化状态**

条件	程序计数器	状态寄存器	PCON 寄存器
上电复位	000h	0001 1xxx	---- 1-0x
正常工作期间 $\overline{MCLR}$ 复位	000h	000u uuuu	---- 1-uu
休眠状态下的 $\overline{MCLR}$ 复位	000h	0001 0uuu	---- 1-uu
WDT 复位	000h	0000 uuuu	---- 1-uu
WDT 唤醒	PC+1	uuu0 0uuu	---- u-uu
欠压复位	000h	000x xuuu	---- 1-u0
休眠时的中断唤醒	PC+1 <sup>(1)</sup>	uuu1 0uuu	---- u-uu

**图注:** u = 未改变, x = 未知, — = 未实现位, 读作 0。

**注 1:** 如果中断唤醒器件且全局允许位 GIE 置位, 则执行 PC+1 后, PC 装入中断矢量 (0004h)。

# PIC16F627A/628A/648A

表 14-7: 寄存器的初始化状态

寄存器	地址	上电复位	<ul style="list-style-type: none"> <li>• 正常工作期间的 MCLR 复位</li> <li>• 休眠期间的 MCLR 复位</li> <li>• WDT 复位</li> <li>• 欠压复位<sup>(1)</sup></li> </ul>	<ul style="list-style-type: none"> <li>• 由中断从休眠状态唤醒<sup>(7)</sup></li> <li>• 由于 WDT 超时从休眠状态唤醒<sup>(7)</sup></li> </ul>
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h, 80h, 100h, 180h	—	—	—
TMR0	01h, 101h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h, 82h, 102h, 182h	0000 0000	0000 0000	PC + 1 <sup>(3)</sup>
STATUS	03h, 83h, 103h, 183h	0001 1xxx	000q quuu <sup>(4)</sup>	uuuq 0uuu <sup>(4)</sup>
FSR	04h, 84h, 104h, 184h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	xxxx 0000	xxxx 0000	uuuu uuuu
PORTB	06h, 106h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	0Ah, 8Ah, 10Ah, 18Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh, 8Bh, 10Bh, 18Bh	0000 000x	0000 000u	uuuu uqqq <sup>(2)</sup>
PIR1	0Ch	0000 -000	0000 -000	qqqq -qqq <sup>(2)</sup>
TMR1L	0Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	0Fh	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	10h	--00 0000	--uu uuuu <sup>(6)</sup>	--uu uuuu
TMR2	11h	0000 0000	0000 0000	uuuu uuuu
T2CON	12h	-000 0000	-000 0000	-uuu uuuu
CCPR1L	15h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	16h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	17h	--00 0000	--00 0000	--uu uuuu
RCSTA	18h	0000 000x	0000 000x	uuuu uuuu
TXREG	19h	0000 0000	0000 0000	uuuu uuuu
RCREG	1Ah	0000 0000	0000 0000	uuuu uuuu
CMCON	1Fh	0000 0000	0000 0000	uu-- uuuu
OPTION	81h, 181h	1111 1111	1111 1111	uuuu uuuu
TRISA	85h	1111 1111	1111 1111	uuuu uuuu
TRISB	86h, 186h	1111 1111	1111 1111	uuuu uuuu
PIE1	8Ch	0000 -000	0000 -000	uuuu -uuu
PCON	8Eh	---- 1-0x	---- 1-uq <sup>(1,5)</sup>	---- u-uu
PR2	92h	1111 1111	1111 1111	uuuu uuuu
TXSTA	98h	0000 -010	0000 -010	uuuu -uuu
SPBRG	99h	0000 0000	0000 0000	uuuu uuuu
EEDATA	9Ah	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADR	9Bh	xxxx xxxx	uuuu uuuu	uuuu uuuu
EECON1	9Ch	---- x000	---- q000	---- uuuu
EECON2	9Dh	—	—	—
VRCON	9Fh	000- 0000	000- 0000	uuu- uuuu

- 图注: u=未改变, x=未知, -=未实现位, 读作 0, q=值随条件变化。
- 注:
- 1: 如果 VDD 过低, 上电复位将被激活, 寄存器将受到不同的影响。
  - 2: INTCON 和 PIR1 寄存器中的一位或多位将受到影响 (而引起唤醒)。
  - 3: 当中断将器件唤醒且 GIE 位置位时, PC 装入中断矢量 (0004h)。
  - 4: 关于具体条件的复位值, 参见表 14-6。
  - 5: 如果由欠压引起复位, 则 bit 0=0。所有其它复位使 bit 0=u。
  - 6: 发生欠压复位 (BOR) 时, 复位至 "--00 0000"。
  - 7: 如果外设产生的中断将器件从休眠状态唤醒, 会改变相关寄存器中的相应位。

图 14-8: 上电时的延时时序 ( $\overline{\text{MCLR}}$  未连接到 VDD): 情形 1

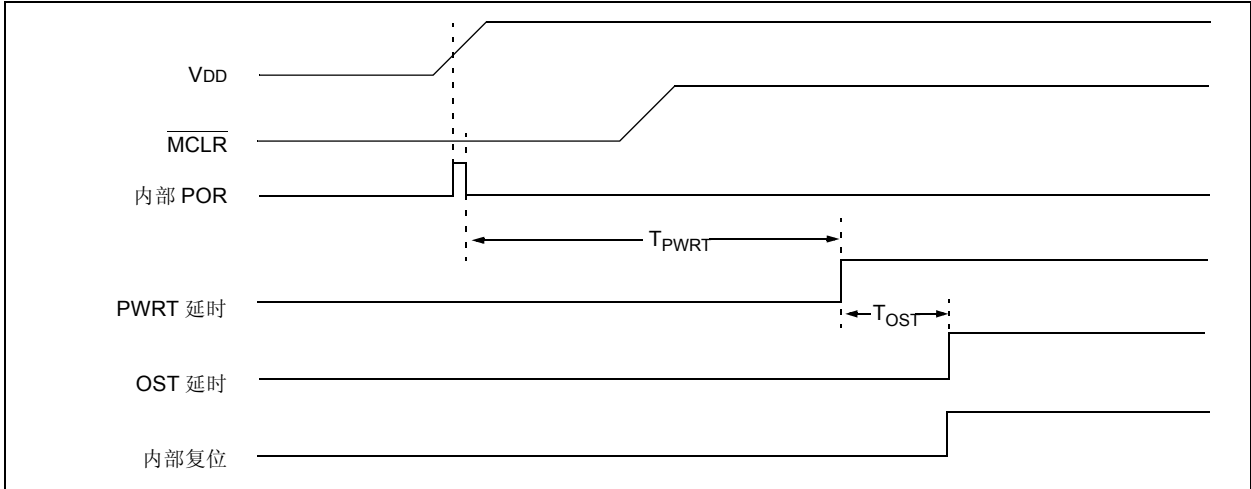


图 14-9: 上电时的延时时序 ( $\overline{\text{MCLR}}$  未连接到 VDD): 情形 2

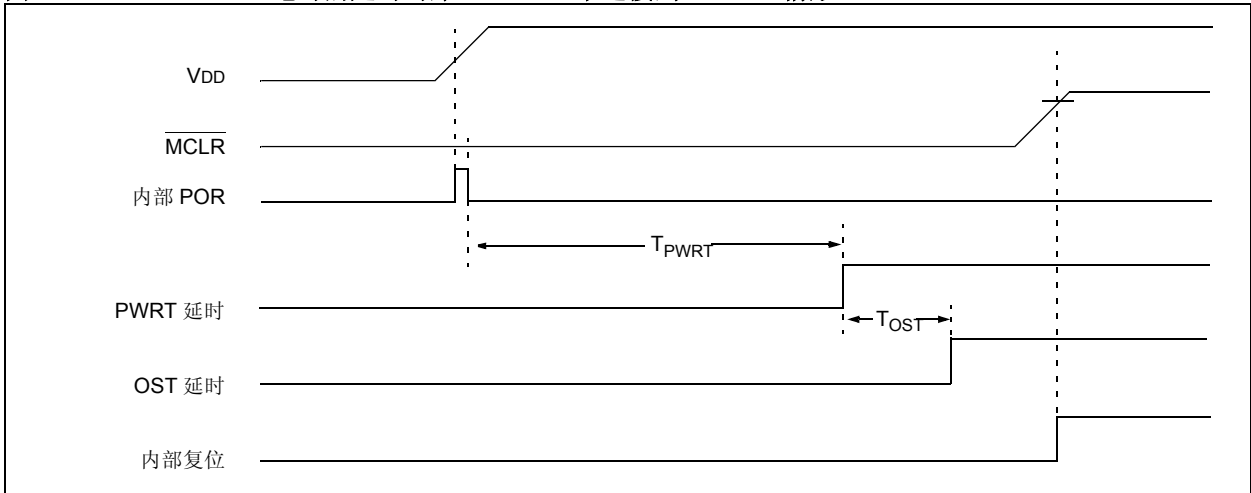
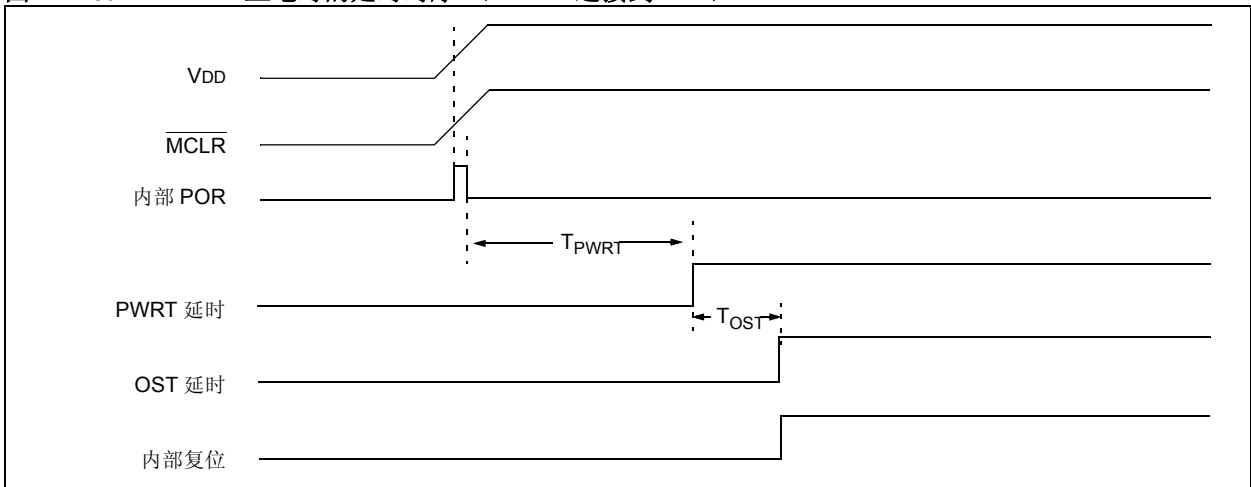


图 14-10: 上电时的延时时序 ( $\overline{\text{MCLR}}$  连接到 VDD)



# PIC16F627A/628A/648A

图 14-11: 外部上电复位电路 (VDD 上电速率过慢时)

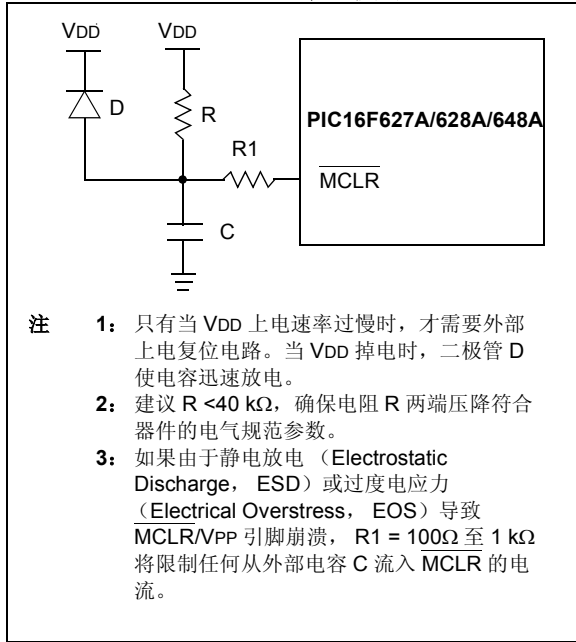


图 14-13: 外部欠压保护电路 2

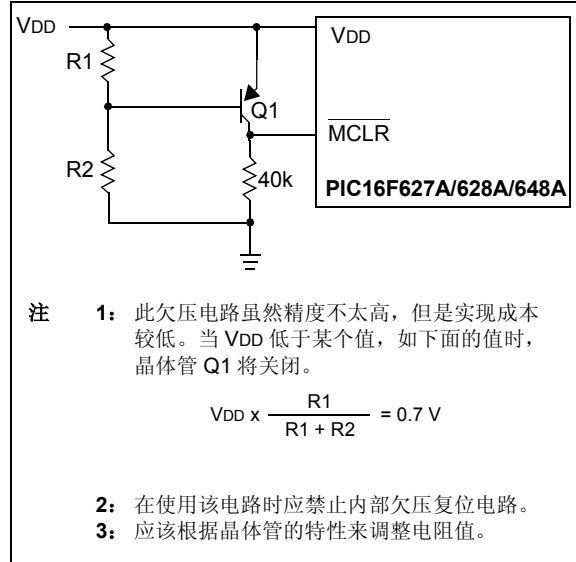
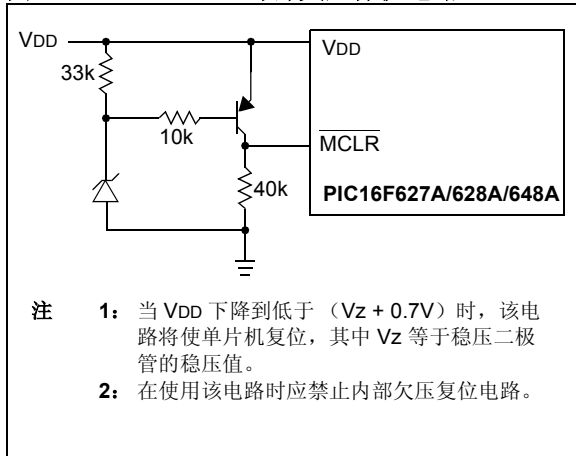


图 14-12: 外部欠压保护电路 1



## 14.5 中断

PIC16F627A/628A/648A 有 10 个中断源：

- 外部中断 RB0/INT
- TMR0 溢出中断
- PORTB 电平变化中断（引脚 RB<7:4>）
- 比较器中断
- USART 中断 TX
- USART 中断 RX
- CCP 中断
- TMR1 溢出中断
- TMR2 匹配中断
- 数据 EEPROM 中断

中断控制寄存器（INTCON）在标志位中记录各个中断请求。它还包括各个中断允许位和全局中断允许位。

全局中断允许位，GIE（INTCON<7>）在置位时允许所有未屏蔽的中断，在清零时，禁止所有中断。可以通过各个中断在INTCON寄存器中相应的使能位来禁止各个中断。复位时 GIE 清零。

执行“中断返回”指令 RETFIE，将退出中断服务程序，同时将置位 GIE 位，从而重新允许 RB0/INT 中断。INTCON 寄存器中包含 INT 引脚中断、RB 端口电平变换中断和 TMR0 溢出中断标志。

特殊的 PTR1 寄存器包含各外设中断的标志位。特殊 PIE1 寄存器包含相应的中断允许位。

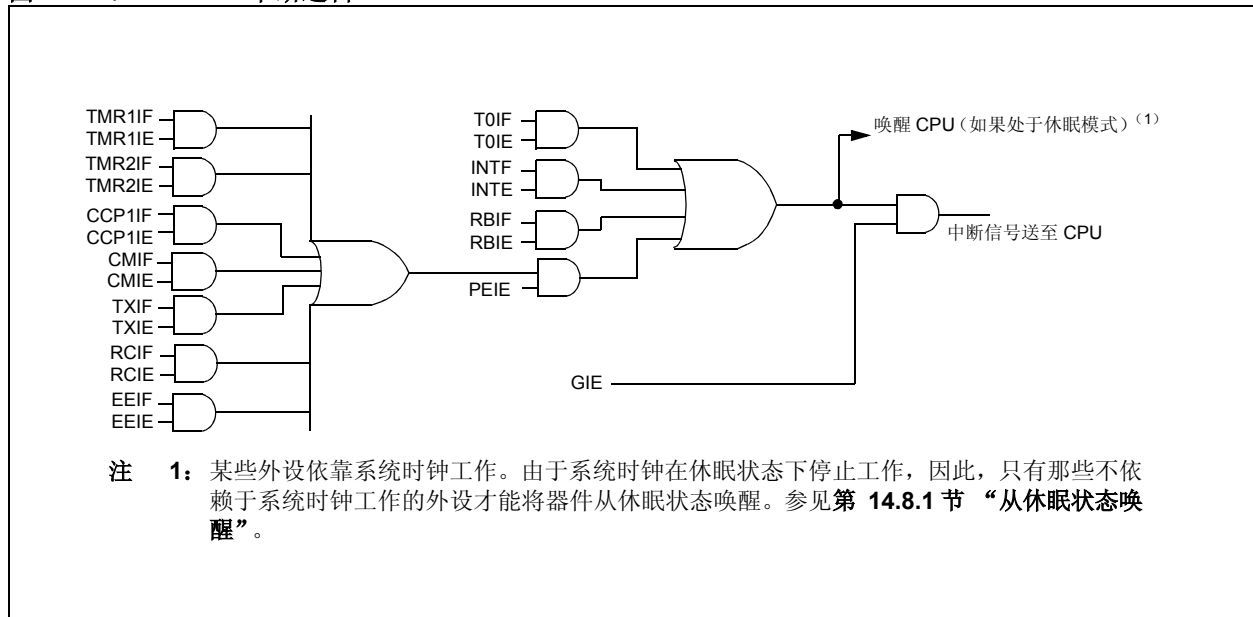
当响应一个中断时，GIE 位被清零以禁止其他中断，返回地址压入堆栈，PC 中装入 0004h。只要在中断服务程序中，就可以通过查询中断标志位来判断中断源。在重新允许中断前，必须用软件将中断标志位清零，以避免 RB0/INT 重复请求中断。

对于外部中断事件，例如 INT 引脚中断或者 PORTB 输入电平变化中断，中断响应延时为 3 到 4 个指令周期。确切的延时时间取决于中断事件何时发生（图 14-15）。对于单周期或双周期指令，中断响应延时完全相同。在中断服务程序中，通过查询中断标志位可判断中断源。在重新允许中断前，必须用软件将中断标志位清零，以避免多个中断请求。

**注 1:** 各中断标志位的设置不受对应的中断屏蔽位和 GIE 位的状态影响。

**注 2:** 当执行一条清零 GIE 位的指令时，将忽略任何等待在下一个周期处理的中断。在执行了清零 GIE 位的指令之后，CPU 立即在下一个周期中执行一条 NOP 指令。当 GIE 位被再次置位时，被忽略的中断仍然等待处理。

图 14-14: 中断逻辑



# PIC16F627A/628A/648A

## 14.5.1 RB0/INT 中断

RB0/INT 引脚上的外部中断是边沿触发的：当 INTEDG 位 (OPTION<6>) 置位时在上升沿触发，而当 INTEDG 位被清零时在下降沿触发。当 RB0/INT 引脚上出现有效沿时，INTF 位 (INTCON<1>) 置位。可以通过清零 INTE 控制位 (INTCON<4>) 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位，则 RB0/INT 中断能把处理器从休眠状态中唤醒。GIE 位的状态取决于处理器在唤醒之后是否转入中断服务程序。有关休眠的详细信息，请参阅第 14.8 节“低功耗 (休眠) 模式”，而有关 RB0/INT 中断将处理器从休眠状态唤醒的时序请参阅图 14-17。

## 14.5.2 TMR0 中断

TMR0 寄存器中的溢出 (FFh → 00h) 将置位 TOIF (INTCON<2>) 位。可以通过置位 / 清零 TOIE (INTCON<5>) 位来允许 / 禁止中断。有关 Timer0 模块的工作，请参阅第 6.0 节“Timer0 模块”。

## 14.5.3 PORTB 中断

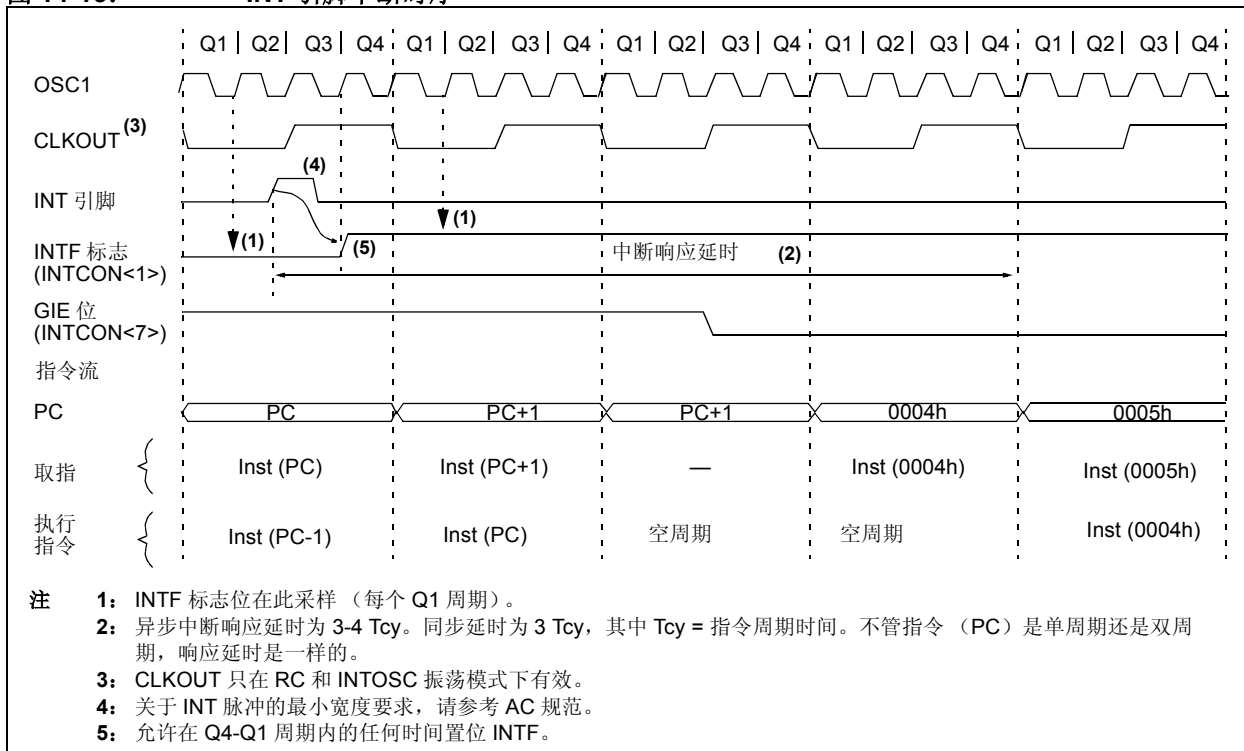
PORTB<7:4> 的输入电平变化将置位 RBIF (INTCON<0>) 位。可以通过置位 / 清零 RBIE (INTCON<3>) 位来允许 / 禁止中断。有关 PORTB 的工作，请参阅第 5.2 节“PORTB 和 TRISB 寄存器”。

**注：** 如果 I/O 引脚上的电平在执行读操作时 (在 Q2 周期开始，在 Q3 周期开始前结束) 发生变化，则 RBIF 中断标志可能不会置位。

## 14.5.4 比较器中断

有关完整的比较器中断描述，请参阅第 10.6 节“比较器中断”。

图 14-15: INT 引脚中断时序



**表 14-8: 中断寄存器综述**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值 <sup>(1)</sup>
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000

注 1: 正常工作期间的其他复位（非上电复位）包括  $\overline{\text{MCLR}}$  复位、欠压复位和看门狗复位。

## 14.6 中断现场保护

进入中断时，仅将返回的 PC 值压入堆栈。通常情况下，用户可能希望保存中断服务程序中使用的关键寄存器（例如，W 寄存器和 Status 寄存器）。这必须用软件实现。

例 14-1 说明了如何保存、恢复 Status 寄存器和 W 寄存器。用户寄存器，W\_TEMP，必须定义到公共存储地址单元中（即 W\_TEMP 定义到 Bank0 的 0x70，因此它在 0xF0、0x170 和 0x1F0 都可进行存取）。例 14-1:

- 保存 W 寄存器的内容
- 保存 Status 寄存器的内容
- 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

### 例 14-1: 在 RAM 中保存 Status 和 W 寄存器的内容

```

MOVWF W_TEMP      ;copy W to temp register,
                  ;could be in any bank
SWAPF STATUS,W    ;swap status to be saved
                  ;into W
BCF STATUS,RP0    ;change to bank 0
                  ;regardless of current
                  ;bank
MOVWF STATUS_TEMP ;save status to bank 0
                  ;register
:
:(ISR)
:

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
    
```

## 14.7 看门狗 (WDT)

看门狗依靠一个片上 RC 振荡器自由运行，它不需要外接任何元器件。该 RC 振荡器独立于 CLKIN 引脚上的 RC 振荡器。这意味着即使已经停止器件 OSC1 和 OSC2 引脚上的时钟信号（例如，因执行 SLEEP 指令而引起），WDT 仍会继续运行。正常工作状态下，WDT 超时会引起器件复位。如果器件处于休眠模式，WDT 超时将唤醒器件，使其继续正常工作。可以通过编程清零配置位 WDTE 来永久地禁止 WDT（第 14.1 节“配置位”）。

### 14.7.1 WDT 周期

在没有后分频器时，WDT 的超时周期标称值为 18 ms。每个器件的超时周期随温度、VDD 和工艺的差异而各不相同（参见 DC 规范，表 17-7）。如果希望得到更长的超时周期，可以用软件写 OPTION 寄存器，将分频比最多可达 1:128 的后分频器分配给 WDT。这样可以实现最长 2.3 秒的超时周期。

如果将后分频器分配给 WDT，CLRWDTC 和 SLEEP 指令会将 WDT 和后分频器清零，并阻止 WDT 超时产生器件复位。

Status 寄存器中的  $\overline{\text{TO}}$  位，将在看门狗超时溢出时被清零。

### 14.7.2 WDT 编程注意事项

在最恶劣的情况下（VDD 最小，温度最高，WDT 预分频比最大），要过几秒钟才会发生超时溢出，因此在编写程序时要考虑到这一点。

# PIC16F627A/628A/648A

图 14-16: 看门狗框图

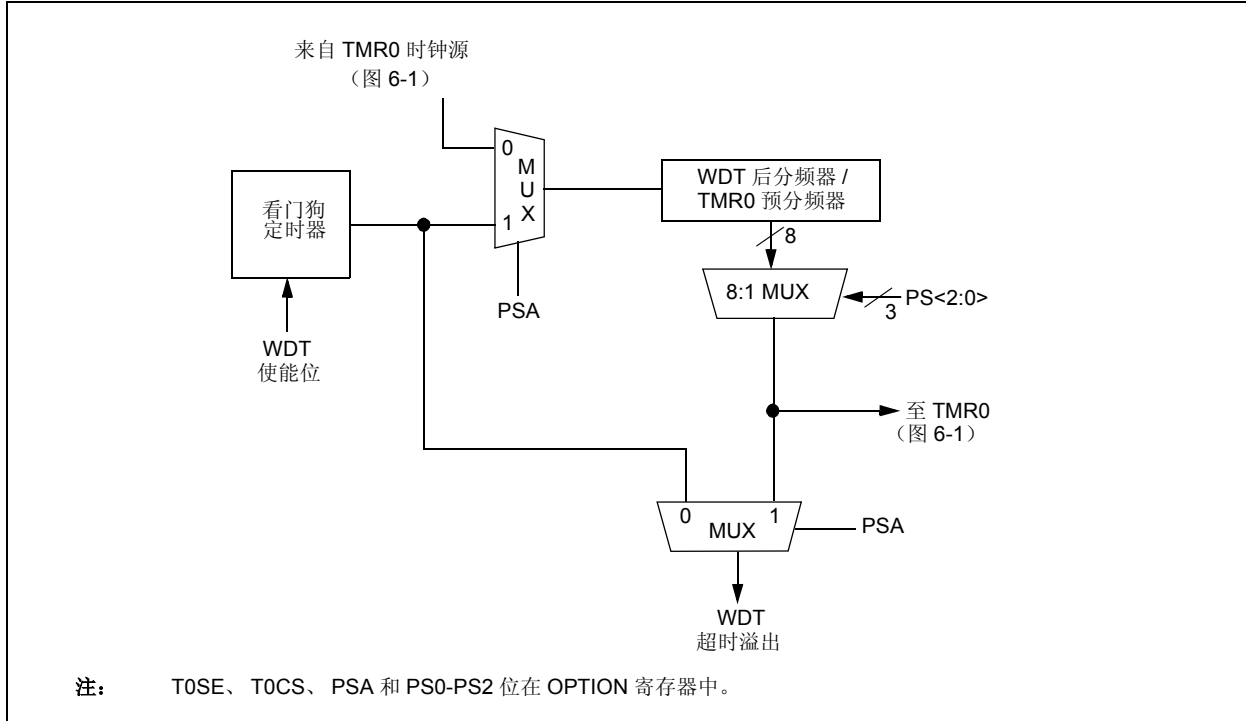


表 14-9: 看门狗寄存器综述

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 时的值	所有其他复位时的值
2007h	CONFIG	LVP	BOREN	MCLRE	FOSC2	PWRTE	WDTE	FOSC1	FOSC0	uuuu uuuu	uuuu uuuu
81h, 181h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

图注： x = 未知，u = 未改变，- = 未实现，读作 0，q = 值随条件变化。

注： 阴影部分表示在看门狗中未使用。

## 14.8 低功耗（休眠）模式

可通过执行 SLEEP 指令进入低功耗模式。

如果看门狗使能的话，将清零看门狗但保持其继续运行，还将清零 Status 寄存器中的 PD 位，置位 TO 位并且关闭振荡驱动器。I/O 端口保持它们在执行 SLEEP 指令之前的状态（输出高电平、低电平或者呈现高阻态）。

在休眠模式下，为了尽量降低电流消耗，所有 I/O 引脚都应该保持为 VDD 或 VSS 电平，不要有外部电路通过 I/O 引脚和比较器消耗电流，同时应禁止 VREF。为了避免悬空输入引起开关电流，应拉高或拉低高阻输入的 I/O 引脚。为了将电流消耗降至最低，TOCKI 输入也应该保持为 VDD 或 VSS 电平。还应考虑到 PORTB 片内上拉的影响。

MCLR 引脚必须保持为逻辑高电平（VIHMC）。

注： 应该注意到 WDT 超时溢出产生的复位不会拉低 MCLR 引脚电平。



## 14.8.1 从休眠状态唤醒

可以通过下列任一事件将器件从休眠状态唤醒：

1.  $\overline{\text{MCLR}}$  引脚上的外部复位输入
2. 看门狗唤醒（如果 WDT 使能）
3. RB0/INT 引脚中断、RB 端口电平变化中断或休眠状态下有效的外设中断。

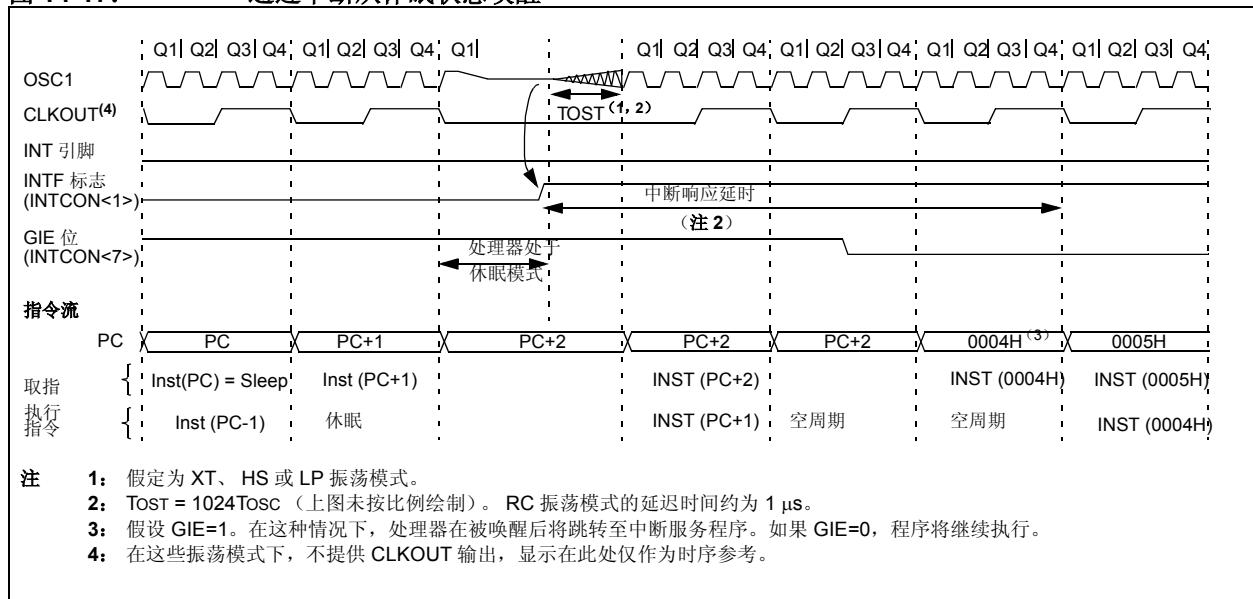
第一种事件会导致器件复位。后两种事件被认为是程序执行的延续。Status 寄存器中的 TO 和 PD 位可以用来确定器件复位的原因。PD 位在上电时被置位，在调用 SLEEP 指令时被清零。TO 位在发生 WDT 唤醒时被清零。

在执行 SLEEP 指令时，下一条指令（PC+1）被预取出。要通过中断事件唤醒器件，必须将对应的中断允许位置位（即允许中断）。唤醒与 GIE 位的状态无关。如果 GIE 位清零（禁止），器件继续执行 SLEEP 指令以后的指令。如果 GIE 位被置位（允许），器件执行 SLEEP 指令之后的指令，然后跳转到中断地址（0004h）。如果不想执行 SLEEP 指令以后的指令，用户应该在 SLEEP 指令后面放置一条 NOP 指令。

**注：** 如果全局中断禁止（GIE 清零），但所有中断源都将它们各自的中断允许位以及对应的中断标志位置位，器件将不会进入休眠模式。此时 SLEEP 指令的执行效果与 NOP 指令相同。

不管唤醒是由什么引起的，器件从休眠状态唤醒时，WDT 都会被清零。

图 14-17: 通过中断从休眠状态唤醒



## 14.9 代码保护

如果将代码保护位清零（代码保护使能），程序存储器单元的内容将读作“0”。详情参见 PIC16F627A/628A/648A EEPROM Memory Programming Specification (DS41196)。

**注：** 只有批量擦除功能可以通过关闭代码保护将 CP 和 CPD 位置位。将擦除整个数据 EEPROM 和闪存程序存储器来关闭代码保护。

## 14.10 用户 ID 单元

器件中有 4 个存储单元（2000h-2003h）被指定为用户 ID 单元，用户可在其中存放校验和或者其他代码识别数字。这些地址单元在正常执行过程中不能存取，但在编程和校验过程中可读写。只可使用用户 ID 单元中的低 4 位来进行校验和计算，尽管每个地址单元共有 14 位。

# PIC16F627A/628A/648A

## 14.11 在线串行编程

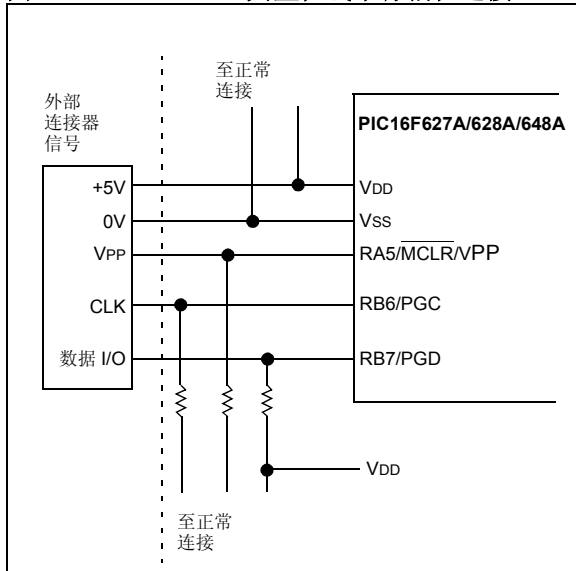
在最终应用电路中可对 PIC16F627A/628A/648A 单片机进行串行编程。这只需要一根时钟线、一根数据线以及一根电源线、一根接地线和一根编程电压线就可实现。这允许用户使用未编程器件来制造电路板，他们可以在产品装运前才对单片机进行编程。这样允许使用编程最新的固件或用户定制的固件。

当  $\overline{\text{MCLR}}$  ( $V_{PP}$ ) 引脚的电平从  $V_{IL}$  上升到  $V_{IH}$  时，保持  $\text{RB6}$  和  $\text{RB7}$  引脚为低电平，可以使器件进入编程 / 校验模式（参见编程规范）。此时  $\text{RB6}$  为编程时钟引脚而  $\text{RB7}$  为编程数据引脚。在此模式下， $\text{RB6}$  和  $\text{RB7}$  均为施密特触发输入。

复位后，将器件置于编程 / 校验模式，程序计数器 (PC) 处于地址单元  $00h$ 。然后向器件发送一条 6 位命令。根据具体命令是执行装载还是读取操作，可向器件提供一个 14 位的程序数据或是从器件送出一个 14 位的程序数据。有关串行编程的具体细节，请参阅 *PIC16F627A/628A/648A EEPROM Memory Programming Specification* (DS41196)。

图 14-18 所示为典型的在线串行编程连接。

图 14-18: 典型在线串行编程连接



## 14.12 低电压编程

配置字的  $\text{LVP}$  位使能低电压编程。在此模式下，用户可仅使用  $5V$  的电压源通过  $\text{ICSP}$  对单片机进行编程。此模式使得不必对  $\overline{\text{MCLR}}$  引脚施加  $V_{IH}$  电压。通常将使能低电压编程的  $\text{LVP}$  位擦除为  $1$ 。在这个模式下， $\text{RB4/PGM}$  引脚专用于编程功能，而不再作为通用的 I/O 引脚。当  $\text{RB4/PGM}$  引脚被置  $1$  时，器件将进入编程模式。对  $\overline{\text{MCLR}}$  引脚施加  $V_{IH}$  电压，仍然可以进入高电压 (HV) 编程模式。

- 注 1:** 在此模式下， $\text{RB4}$  引脚不再用作通用 I/O 引脚。
- 注 2:** 在擦除操作过程中， $V_{DD}$  电压必须为  $5.0V \pm 10\%$ 。

如果不使用低电压编程模式，应将  $\text{LVP}$  位编程为  $0$ ，这样  $\text{RB4/PGM}$  就成为数字 I/O 引脚。要编程器件，必须在编程过程中对  $\overline{\text{MCLR}}$  引脚施加  $V_{IH}$ 。只有通过  $\overline{\text{MCLR}}$  引脚施加  $V_{IH}$  电压进入编程模式时，才能编程  $\text{LVP}$  位。如果通过  $\text{RB4/PGM}$  引脚进入编程模式，则不能对  $\text{LVP}$  位进行编程。

应该注意的是，一旦  $\text{LVP}$  位被编程为  $0$ ，就只能使用高电压编程模式对器件编程。

## 14.13 在线调试器

在线调试需要占用时钟、数据和 MCLR 引脚，因此采用 MPLAB<sup>®</sup> ICD 2 来开发 18 引脚器件是不切实际的。一种特殊的 28 引脚 PIC16F648A-ICD 器件可配合 MPLAB ICD 2 来进行调试，提供独立的时钟、数据和 MCLR 引脚，因此所有通常可用的引脚都可供用户使用。PIC16F648A-ICD 支持 PIC16F627A/628A/648A 所有三个版本的调试。

这种特殊的 ICD 器件安装在适配头的上端，其信号连接到 MPLAB ICD 2 连接器。适配头的下端是一个 18 引脚的插座，通过一个独立的 18 引脚连接器，可将这个插座接插到用户的目标板中。

当 PIC16F648A-ICD 的 ICD 引脚为低电平时，使能在线调试功能。当与 MPLAB ICD 2 配合使用时，这一功能允许简单的调试功能。当单片机使能这一功能时，某些资源被占用，不能供普通的其他功能使用。表 14-10 列出了后台调试器所占用的资源。

**表 14-10: 调试器占用的资源**

I/O 引脚	ICDCLK, ICDDATA
堆栈	1 级深度
程序存储器	地址 0h 必须存放 NOP 指令 300h-3FEh

带适配头的 PIC16F648A-ICD 器件作为一个组件提供。详情请参见 Microchip 部件编号 AC162053。

# PIC16F627A/628A/648A

---

---

注:

## 15.0 指令集综述

每条PIC16F627A/628A/648A指令都是一个14位的字，由指定指令类型的操作码和进一步说明指令具体操作的一个或多个操作数组成。表 15-2 中的 PIC16F627A/628A/648A 指令集综述列出了**字节操作类指令**、**位操作类指令**、**立即数和控制操作类指令**。表 15-1 给出了操作码字段的描述。

对于**字节操作指令**，“f”表示文件寄存器标识符，而“d”则表示目标标识符。文件寄存器标识符指定指令将使用哪一个文件寄存器。

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

对于**位操作指令**，“b”表示位域标识符，选择将受操作影响的位，而“f”则表示该位处于哪个文件寄存器中。

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

### 15.1 读 - 修改 - 写操作

任何指定文件寄存器作为指令一部分的指令，都执行读 - 修改 - 写 (R-M-W) 操作。先读寄存器，然后修改数据，根据指令或目标标识符“d”存储结果。即使指令是写寄存器，但也要对该寄存器进行读操作。

例如，指令 `clrf PORTB` 将读 `PORTB`，清零所有数据位，然后将结果写回 `PORTB`。对于配置为输入并使用 `PORTB` 电平变化中断功能的引脚来说，这个例子会产生意外结果，置位 `RBIF` 标志位的条件会被清除。

表 15-1: 操作码字段描述

字段	描述
f	文件寄存器地址 (0x00 到 0x7F)
W	工作寄存器 (累加器)
b	某 8 位文件寄存器内的位地址
k	立即数、常数或标号
x	与取值无关的位 (=0 或 1) 对于 x=0，将由汇编器产生代码。为了与所有的 Microchip 软件工具兼容，建议使用这种格式。
d	目标寄存器选择； d=0：结果保存至 W， d=1：结果保存至文件寄存器 f。 默认 d=1
TO	超时标志位
PD	掉电标志位

指令集是高度正交的，分为三种基本类型：

- 字节操作类指令
- 位操作类指令
- 立即数和控制操作类指令

除条件测试为真的程序跳转或执行后会改变程序计数器值的指令以外，其他指令都是单周期指令。对于前面的指令，执行指令需要两个指令周期，第二个周期中执行的是一条 NOP 指令。每个指令周期包括 4 个振荡周期。因此，对于频率为 4 MHz 的振荡器，正常指令执行时间为 1 μs。如果条件测试为真或指令执行改变了程序计数器的值，则指令的执行时间为 2 μs。

表 15-2 列出了能被 MPASM™ 汇编器识别的指令。

图 15-1 给出了指令的三种通用格式。

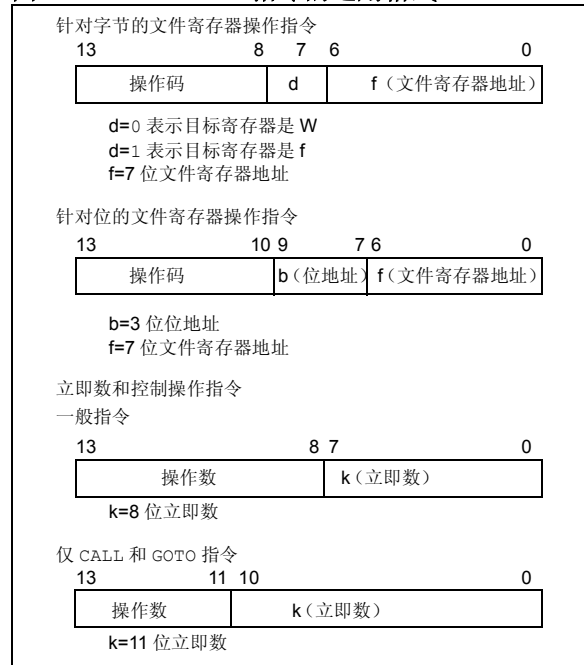
- 注 1:** 任何未使用的操作码都是保留的。使用任何保留的操作码将会导致异常操作。
- 2:** 为保持与将来 PICmicro® 单片机的向上兼容，不要使用 `OPTION` 和 `TRIS` 指令。

所有的示例均使用下面的格式来表示十六进制数：

0xhh

其中 h 代表一个 16 进制位。

图 15-1: 指令的通用格式



# PIC16F627A/628A/648A

表 15-2: PIC16F627A/628A/648A 指令集

助记符, 操作数	描述	周期	14 位指令字				受影响的状态位	注	
			MSb		LSb				
针对字节的文件寄存器操作									
ADDWF	f, d	W 和 f 相加	1	00	0111	dfff	ffff	C、DC、Z	1,2
ANDWF	f, d	W 和 f 进行“与”运算	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	f 清零	1	00	0001	1fff	ffff	Z	2
CLRWF	—	W 清零	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	f 取反	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	f 递减 1	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	f 递减 1, 为 0 则跳过	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	f 递增 1	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	f 递增 1, 为 0 则跳过	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	W 和 f 进行“或”运算	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	f 内容传送到目标寄存器	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	将 W 的内容传送到 f	1	00	0000	1fff	ffff		
NOP	—	空操作	1	00	0000	0xx0	0000		
RLF	f, d	f 寄存器带进位标志位循环左移	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	f 寄存器带进位标志位循环右移	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	f 减 W	1	00	0010	dfff	ffff	C、DC、Z	1,2
SWAPF	f, d	f 半字节交换	1	00	1110	dfff	ffff		1,2
XORWF	f, d	W 和 f 进行“异或”运算	1	00	0110	dfff	ffff	Z	1,2
针对位的文件寄存器操作指令									
BCF	f, b	f 中某位清零	1	01	00bb	bfff	ffff		1,2
BSF	f, b	f 中某位置 1	1	01	01bb	bfff	ffff		1,2
BTFSF	f, b	检测 f 中的某位, 为 0 则跳过	1(2)	01	10bb	bfff	ffff		3
BTFSF	f, b	检测 f 中的某位, 为 1 则跳过	1(2)	01	11bb	bfff	ffff		3
立即数和控制操作指令									
ADDLW	k	立即数与 W 的内容相加	1	11	111x	kkkk	kkkk	C、DC、Z	
ANDLW	k	立即数与 W 进行“与”运算	1	11	1001	kkkk	kkkk	Z	
CALL	k	调用子程序	2	10	0kkk	kkkk	kkkk		
CLRWDT	—	看门狗清零	1	00	0000	0110	0100	TO, PD	
GOTO	k	跳转到地址	2	10	1kkk	kkkk	kkkk		
IORLW	k	立即数与 W 进行“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	返回时将立即数送入 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	TO, PD	
SUBLW	k	立即数减 W	1	11	110x	kkkk	kkkk	C、DC、Z	
XORLW	k	立即数与 W 进行“异或”运算	1	11	1010	kkkk	kkkk	Z	

- 注
- 1: 当 I/O 寄存器用自身内容修改自身时 (例如: MOVF PORTB, 1), 使用的值为出现在引脚上的值。例如, 如果一引脚配置为输入, 其数据锁存器中的值为“1”, 但此时外部器件将该引脚拉为低电平, 则被写回数据锁存器的数据将是“0”。
  - 2: 如果将预分频器分配给 Timer0 模块, 则对 TMR0 寄存器执行这条指令 (适当时, d=1) 将清零预分频器。
  - 3: 如果程序计数器 (PC) 被修改或者条件检测为真, 则指令需要两个指令周期。第二个指令周期执行一条 NOP 指令。

## 15.2 指令描述

### ADDLW 立即数与 W 的内容相加

语法:	[label] ADDLW k				
操作数:	$0 \leq k \leq 255$				
操作:	$(W)+k \rightarrow (W)$				
受影响的状态位:	C、DC、Z				
机器码:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">11</td> <td style="padding: 2px;">111x</td> <td style="padding: 2px;">kkkk</td> <td style="padding: 2px;">kkkk</td> </tr> </table>	11	111x	kkkk	kkkk
11	111x	kkkk	kkkk		
描述:	W 寄存器的内容与 8 位立即数“k”相加, 结果存入 W 寄存器。				
指令字数:	1				
指令周期:	1				
示例:	<pre>ADDLW 0x15 执行指令前 W = 0x10 执行指令后 W = 0x25</pre>				

### ADDWF W 和 f 相加

语法:	[label] ADDWF f,d				
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$				
操作:	$(W)+(f) \rightarrow (dest)$				
受影响的状态位:	C、DC、Z				
机器码:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">00</td> <td style="padding: 2px;">0111</td> <td style="padding: 2px;">dfff</td> <td style="padding: 2px;">ffff</td> </tr> </table>	00	0111	dfff	ffff
00	0111	dfff	ffff		
描述:	W 寄存器的内容与“f”寄存器的内容相加。如果“d”为 0, 结果存入 W 寄存器。如果“d”为 1, 结果存回“f”寄存器。				
指令字数:	1				
指令周期:	1				
示例:	<pre>ADDWF REG1, 0 执行指令前 W = 0x17 REG1 = 0xC2 执行指令后 W = 0xD9 REG1 = 0xC2 Z = 0 C = 0 DC = 0</pre>				

### ANDLW 立即数与 W 进行“与”运算

语法:	[label] ANDLW k				
操作数:	$0 \leq k \leq 255$				
操作:	$(W).AND.(k) \rightarrow (W)$				
受影响的状态位:	Z				
机器码:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">11</td> <td style="padding: 2px;">1001</td> <td style="padding: 2px;">kkkk</td> <td style="padding: 2px;">kkkk</td> </tr> </table>	11	1001	kkkk	kkkk
11	1001	kkkk	kkkk		
描述:	W 寄存器的内容与 8 位立即数“k”相与。结果存入 W 寄存器。				
指令字数:	1				
指令周期:	1				
示例:	<pre>ANDLW 0x5F 执行指令前 W = 0xA3 执行指令后 W = 0x03</pre>				

### ANDWF W 和 f 进行“与”运算

语法:	[label] ANDWF f,d				
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$				
操作:	$(W).AND.(f) \rightarrow (dest)$				
受影响的状态位:	Z				
机器码:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">00</td> <td style="padding: 2px;">0101</td> <td style="padding: 2px;">dfff</td> <td style="padding: 2px;">ffff</td> </tr> </table>	00	0101	dfff	ffff
00	0101	dfff	ffff		
描述:	W 寄存器与“f”寄存器相与。如果“d”为 0, 结果存入 W 寄存器。如果“d”为 1, 结果存回“f”寄存器。				
指令字数:	1				
指令周期:	1				
示例:	<pre>ANDWF REG1, 1 执行指令前 W = 0x17 REG1 = 0xC2 执行指令后 W = 0x17 REG1 = 0x02</pre>				

# PIC16F627A/628A/648A

## BCF 清零 f 中的某位

语法: [label] BCF f,b  
 操作数:  $0 \leq f \leq 127$   
 $0 \leq b \leq 7$   
 操作:  $0 \rightarrow (f<b>)$   
 受影响的状态位: 无  
 机器码: 

01	00bb	bfff	ffff
----	------	------	------

  
 描述: 将寄存器“f”的位“b”清零。  
 指令字数: 1  
 指令周期: 1  
 示例: BCF REG1, 7

执行指令前  
 REG1 = 0xC7  
 执行指令后  
 REG1 = 0x47

## BSF 置位 f 中的某位

语法: [label] BSF f,b  
 操作数:  $0 \leq f \leq 127$   
 $0 \leq b \leq 7$   
 操作:  $1 \rightarrow (f<b>)$   
 受影响的状态位: 无  
 机器码: 

01	01bb	bfff	ffff
----	------	------	------

  
 描述: 将寄存器“f”的位“b”置 1。  
 指令字数: 1  
 指令周期: 1  
 示例: BSF REG1, 7

执行指令前  
 REG1 = 0x0A  
 执行指令后  
 REG1 = 0x8A

## BTFSC 检测 f 中的某位, 为 0 则跳过

语法: [label] BTFSC f,b  
 操作数:  $0 \leq f \leq 127$   
 $0 \leq b \leq 7$   
 操作: 如果  $(f<b>)=0$  则跳过  
 受影响的状态位: 无  
 机器码: 

01	10bb	bfff	ffff
----	------	------	------

  
 描述: 如果寄存器“f”的位“b”为 0, 则跳过下一条指令。  
 如果位“b”为 1, 那么下一条指令 (在当前指令执行期间取指) 被丢弃, 转而执行一条 NOP 指令, 使该指令变成双周期指令。  
 指令字数: 1  
 指令周期: 1(2)  
 示例: 

```
HERE    BTFSC   REG1
FALSE   GOTO   PROCESS_CODE
TRUE    :
```

执行指令前  
 PC = 地址 HERE  
 执行指令后  
 如果  $REG<1>=0$ ,  
 PC = 地址 TRUE  
 如果  $REG<1>=1$ ,  
 PC = 地址 FALSE



# PIC16F627A/628A/648A

## BTFSS 检测 f 中的某位，为 1 则跳过

语法: [label] BTFSS f,b

操作数:  $0 \leq f \leq 127$   
 $0 \leq b < 7$

操作: 如果 (f<b>)=1 则跳过

受影响的状态位: 无

机器码:

01	11bb	bfff	ffff
----	------	------	------

描述: 如果寄存器“f”的位“b”为 1, 则跳过下一条指令。  
如果位“b”为 0, 那么下一条指令 (在当前指令执行期间取指) 被丢弃, 转而执行一条 NOP 指令, 使该指令变成双周期指令。

指令字数: 1

指令周期: 1(2)

示例:

```

HERE   BTFSS   REG1
FALSE  GOTO   PROCESS_CODE
TRUE   .
        .
        .
    
```

执行指令前  
PC = 地址      HERE

执行指令后  
如果 FLAG<1>=0,  
PC= 地址      FALSE  
如果 FLAG<1>=1,  
PC= 地址      TRUE

## CALL 调用子程序

语法: [label] CALL k

操作数:  $0 \leq k \leq 2047$

操作: (PC)+1 → TOS,  
k → PC<10:0>,  
(PCLATH<4:3>) → PC<12:11>

受影响的状态位: 无

机器码:

10	0kkk	kkkk	kkkk
----	------	------	------

描述: 调用子程序。首先, 返回地址 (PC+1) 被压入堆栈。11 位立即数地址被装入 PC<10:0>。PC 的高位从 PCLATH 装入。CALL 为一条双周期指令。

指令字数: 1

指令周期: 2

示例:

```

HERE   CALL   THERE
    
```

执行指令前  
PC = 地址      HERE

执行指令后  
PC = 地址      THERE  
TOS = 地址      HERE+1

## CLRF 清零 f

语法: [label] CLRF f

操作数:  $0 \leq f \leq 127$

操作: 00h → (f)  
1 → Z

受影响的状态位: Z

机器码:

00	0001	1fff	ffff
----	------	------	------

描述: 寄存器“f”的内容被清零, Z 位置 1。

指令字数: 1

指令周期: 1

示例:

```

CLRF   REG1
    
```

执行指令前  
REG1 = 0x5A

执行指令后  
REG1 = 0x00  
Z = 1

# PIC16F627A/628A/648A

CLRW	W 清零				
语法:	[label] CLRW				
操作数:	无				
操作:	00h → (W) 1 → Z				
受影响的状态位:	Z				
机器码:	<table border="1"> <tr> <td>00</td> <td>0001</td> <td>0000</td> <td>0011</td> </tr> </table>	00	0001	0000	0011
00	0001	0000	0011		
描述:	W 寄存器被清零。全零标志位 (Z) 置 1。				
指令字数:	1				
指令周期:	1				
示例:	<pre>CLRW 执行指令前 W = 0x5A 执行指令后 W = 0x00 Z = 1</pre>				

CLRWDT	看门狗清零				
语法:	[label] CLRWDT				
操作数:	无				
操作:	00h → WDT 0 → $\overline{\text{WDT}}$ 预分频器, 1 → $\overline{\text{TO}}$ 1 → $\overline{\text{PD}}$				
受影响的状态位:	$\overline{\text{TO}}$ , $\overline{\text{PD}}$				
机器码:	<table border="1"> <tr> <td>00</td> <td>0000</td> <td>0110</td> <td>0100</td> </tr> </table>	00	0000	0110	0100
00	0000	0110	0100		
描述:	CLRWDT 指令复位看门狗。还将复位 $\overline{\text{WDT}}$ 的预分频器。状态位 $\overline{\text{TO}}$ 和 $\overline{\text{PD}}$ 被置 1。				
指令字数:	1				
指令周期:	1				
示例:	<pre>CLRWDT 执行指令前 WDT 计数器 = ? 执行指令后 WDT 计数器 = 0x00 <math>\overline{\text{WDT}}</math> 预分频器 = 0 <math>\overline{\text{TO}}</math> = 1 <math>\overline{\text{PD}}</math> = 1</pre>				

COMF	f 取反				
语法:	[label] COMF f,d				
操作数:	0 ≤ f ≤ 127 d ∈ [0,1]				
操作:	(f) → (dest)				
受影响的状态位:	Z				
机器码:	<table border="1"> <tr> <td>00</td> <td>1001</td> <td>dfff</td> <td>ffff</td> </tr> </table>	00	1001	dfff	ffff
00	1001	dfff	ffff		
描述:	寄存器 “f” 的内容取反。如果 “d” 为 0, 结果存入 W。如果 “d” 为 1, 结果存回寄存器 “f”。				
指令字数:	1				
指令周期:	1				
示例:	<pre>COMF    REG1, 0 执行指令前 REG1 = 0x13 执行指令后 REG1 = 0x13 W = 0xEC</pre>				

DECF	f 减 1				
语法:	[label] DECF f,d				
操作数:	0 ≤ f ≤ 127 d ∈ [0,1]				
操作:	(f)-1 → (dest)				
受影响的状态位:	Z				
机器码:	<table border="1"> <tr> <td>00</td> <td>0011</td> <td>dfff</td> <td>ffff</td> </tr> </table>	00	0011	dfff	ffff
00	0011	dfff	ffff		
描述:	寄存器 “f” 的内容减 1。如果 “d” 为 0, 结果存入 W 寄存器。如果 “d” 为 1, 结果存回 “f” 寄存器。				
指令字数:	1				
指令周期:	1				
示例:	<pre>DECF    CNT, 1 执行指令前 CNT = 0x01 Z = 0 执行指令后 CNT = 0x00 Z = 1</pre>				

# PIC16F627A/628A/648A

DECFSZ	f 递减, 为 0 则跳过				
语法:	[label] DECFSZ f,d				
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$				
操作:	$(f)-1 \rightarrow (\text{dest})$ ; 如果结果为 0 则跳过				
受影响的状态位:	无				
机器码:	<table border="1"><tr><td>00</td><td>1011</td><td>dfff</td><td>ffff</td></tr></table>	00	1011	dfff	ffff
00	1011	dfff	ffff		
描述:	寄存器“f”的内容递减。如果“d”为 0, 结果存入 W 寄存器。如果“d”为 1, 结果存回“f”寄存器。 如果结果为 0, 则丢弃已经取出的下一条指令。转而执行一条 NOP 指令, 使该指令变成双周期指令。				
指令字数:	1				
指令周期:	1(2)				
示例:	HERE      DECFSZ   REG1, 1 GOTO      LOOP CONTINUE : :  执行指令前 PC     = 地址 HERE 执行指令后 REG1  = REG1-1 如果 REG1=0, PC    = 地址 CONTINUE 如果 REG1≠0, PC    = 地址 HERE+1				

GOTO	无条件转移				
语法:	[label] GOTO k				
操作数:	$0 \leq k \leq 2047$				
操作:	$k \rightarrow \text{PC}<10:0>$ $\text{PCLATH}<4:3> \rightarrow \text{PC}<12:11>$				
受影响的状态位:	无				
机器码:	<table border="1"><tr><td>10</td><td>1kkk</td><td>kkkk</td><td>kkkk</td></tr></table>	10	1kkk	kkkk	kkkk
10	1kkk	kkkk	kkkk		
描述:	GOTO 是无条件转移指令。11 位立即数值装入 PC<10:0>。PC 高位从 PCLATH<4:3> 装入。GOTO 为一条双周期指令。				
指令字数:	1				
指令周期:	2				
示例:	GOTO THERE  执行指令后 PC = 地址      THERE				

# PIC16F627A/628A/648A

**INCF**                    **f 递增 1**

---

语法:                    [label] INCF f,d

操作数:                     $0 \leq f \leq 127$   
                               $d \in [0,1]$

操作:                       $(f)+1 \rightarrow (\text{dest})$

受影响的状态位:        Z

机器码:                    

00	1010	dfff	ffff
----	------	------	------

描述:                      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“f”寄存器。

指令字数:                1

指令周期:                1

示例:                      INCF     REG1, 1

                              执行指令前  
                                  REG1 = 0xFF  
                                  Z     = 0

                              执行指令后  
                                  REG1 = 0x00  
                                  Z     = 1

**INCFSZ**                    **f 递增 1，为 0 则跳过**

---

语法:                    [label] INCFSZ f,d

操作数:                     $0 \leq f \leq 127$   
                               $d \in [0,1]$

操作:                       $(f)+1 \rightarrow (\text{dest})$ ，如果结果为 0 则跳过

受影响的状态位:        无

机器码:                    

00	1111	dfff	ffff
----	------	------	------

描述:                      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“f”寄存器。  
如果结果为 0，则舍弃已经取出的下一条指令。转而执行一条 NOP 指令，使该指令变成双周期指令。

指令字数:                1

指令周期:                1(2)

示例:                      HERE        INCFSZ     REG1, 1  
  GOTO        LOOP

                              CONTINUE  
                                  :  
                                  :

                              执行指令前  
                                  PC    = 地址 HERE

                              执行指令后  
                                  REG1 = REG1+1  
                                  如果 CNT=0,  
                                  PC    = 地址 CONTINUE  
                                  如果 REG1≠0,  
                                  PC    = 地址 HERE+1

## IORLW 立即数与 W 进行“或”运算

语法: [label] IORLW k

操作数:  $0 \leq k \leq 255$

操作: (W).OR. k → (W)

受影响的状态位: Z

机器码: 

11	1000	kkkk	kkkk
----	------	------	------

描述: W 寄存器的内容与 8 位立即数“k”作“或”运算。结果存入 W 寄存器。

指令字数: 1

指令周期: 1

示例: IORLW 0x35

执行指令前  
W = 0x9A  
执行指令后  
W = 0xBF  
Z = 0

## MOVLW 将立即数传送到 W

语法: [label] MOVLW k

操作数:  $0 \leq k \leq 255$

操作: k → (W)

受影响的状态位: 无

机器码: 

11	00xx	kkkk	kkkk
----	------	------	------

描述: 8 位立即数“k”被装入 W 寄存器。无关的位置为 0。

指令字数: 1

指令周期: 1

示例: MOVLW 0x5A

执行指令后  
W = 0x5A

## IORWF W 和 f 进行“或”运算

语法: [label] IORWF f,d

操作数:  $0 \leq f \leq 127$   
 $d \in [0,1]$

操作: (W).OR. (f) → (dest)

受影响的状态位: Z

机器码: 

00	0100	dfff	ffff
----	------	------	------

描述: W 寄存器与“f”寄存器进行“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“f”寄存器。

指令字数: 1

指令周期: 1

示例: IORWF REG1, 0

执行指令前  
REG1 = 0x13  
W = 0x91  
执行指令后  
REG1 = 0x13  
W = 0x93  
Z = 1

## MOVF 传送 f 的内容到目标寄存器

语法: [label] MOVF f,d

操作数:  $0 \leq f \leq 127$   
 $d \in [0,1]$

操作: (f) → (dest)

受影响的状态位: Z

机器码: 

00	1000	dfff	ffff
----	------	------	------

描述: 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 受到指令结果的影响，d=1 可用于检测文件寄存器内容是否为全 0。

指令字数: 1

指令周期: 1

示例: MOVF REG1, 0

执行指令后  
W = REG1 寄存器中的值  
Z = 1

# PIC16F627A/628A/648A

## MOVWF 将 W 的内容传送到 f

语法:	[label] MOVWF f				
操作数:	$0 \leq f \leq 127$				
操作:	(W) → (f)				
受影响的状态位:	无				
机器码:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">00</td> <td style="padding: 2px;">0000</td> <td style="padding: 2px;">1fff</td> <td style="padding: 2px;">ffff</td> </tr> </table>	00	0000	1fff	ffff
00	0000	1fff	ffff		
描述:	将 W 寄存器中的数据传送到寄存器“f”。				
指令字数:	1				
指令周期:	1				
示例:	<pre>MOVWF  REG1 执行指令前 REG1 = 0xFF W     = 0x4F 执行指令后 REG1 = 0x4F W     = 0x4F</pre>				

## NOP 空操作

语法:	[label] NOP				
操作数:	无				
操作:	无操作				
受影响的状态位:	无				
机器码:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">00</td> <td style="padding: 2px;">0000</td> <td style="padding: 2px;">0xx0</td> <td style="padding: 2px;">0000</td> </tr> </table>	00	0000	0xx0	0000
00	0000	0xx0	0000		
描述:	无操作。				
指令字数:	1				
指令周期:	1				
示例:	NOP				

## OPTION 装入 Option 寄存器

语法:	[label] OPTION				
操作数:	无				
操作:	(W) → OPTION				
受影响的状态位:	无				
机器码:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">00</td> <td style="padding: 2px;">0000</td> <td style="padding: 2px;">0110</td> <td style="padding: 2px;">0010</td> </tr> </table>	00	0000	0110	0010
00	0000	0110	0010		
描述:	将 W 寄存器的内容装入到 OPTION 寄存器。支持该指令是为了与 PIC16C5X 产品代码兼容。OPTION 是可读写的寄存器，用户可对其直接寻址。只能对其使用寄存器指令，例如 MOVWF。				
指令字数:	1				
指令周期:	1				
示例:	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>为保持与将来 PICmicro® 单片机的向上兼容性，不要使用该指令。</td> </tr> </table>	为保持与将来 PICmicro® 单片机的向上兼容性，不要使用该指令。			
为保持与将来 PICmicro® 单片机的向上兼容性，不要使用该指令。					

## RETFIE 从中断返回

语法:	[label] RETFIE				
操作数:	无				
操作:	TOS → PC, 1 → GIE				
受影响的状态位:	无				
机器码:	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">00</td> <td style="padding: 2px;">0000</td> <td style="padding: 2px;">0000</td> <td style="padding: 2px;">1001</td> </tr> </table>	00	0000	0000	1001
00	0000	0000	1001		
描述:	从中断返回。执行出栈操作，将栈顶 (TOS) 内容装入 PC。通过置位全局中断允许位 GIE (INTCON<7>) 允许中断。这是一条双周期指令。				
指令字数:	1				
指令周期:	2				
示例:	<pre>RETFIE 中断后 PC = TOS GIE = 1</pre>				

# PIC16F627A/628A/648A

## RETLW 返回时将立即数传送到 W

语法: [label] RETLW k

操作数:  $0 \leq k \leq 255$

操作:  $k \rightarrow (W)$ ;  
TOS  $\rightarrow$  PC

受影响的状态位: 无

机器码: 

11	01xx	kkkk	kkkk
----	------	------	------

描述: 8 位立即数 “k” 被装入 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: 2

示例:

```
CALL TABLE;W contains table
    ;offset value
    ;W now has table value
    .
    .
TABLE ADDWF PC;W = offset
RETLW k1;Begin table
RETLW k2;
    .
    .
RETLW kn; End of table
```

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: [label] RETURN

操作数: 无

操作: TOS  $\rightarrow$  PC

受影响的状态位: 无

机器码: 

00	0000	0000	1000
----	------	------	------

描述: 从子程序返回。执行出栈操作，将栈顶（TOS）单元内容装入程序计数器。这是一条双周期指令。

指令字数: 1

指令周期: 2

示例:

```
RETURN
```

执行后  
PC = TOS

## RLF f 寄存器带进位标志位循环左移

语法: [label] RLF f,d

操作数:  $0 \leq f \leq 127$   
 $d \in [0,1]$

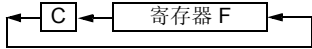
操作: 参见下面的描述

受影响的状态位: C

机器码: 

00	1101	dfff	ffff
----	------	------	------

描述: 寄存器 “f” 的内容带进位标志位循环左移 1 位。如果 “d” 为 0，结果存入 W 寄存器。如果 “d” 为 1，结果存回 “f” 寄存器。



指令字数: 1

指令周期: 1

示例:

```
RLF    REG1, 0
```

执行指令前  
REG1=1110 0110  
C = 0

执行指令后  
REG1=1110 0110  
W = 1100 1100  
C = 1

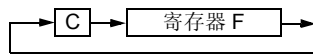
# PIC16F627A/628A/648A

## RRF f 寄存器带进位标志位循环右移

语法: [label] RRF f,d  
 操作数:  $0 \leq f \leq 127$   
 $d \in [0,1]$   
 操作: 参见下面的描述  
 受影响的状态位: C  
 机器码: 

00	1100	dfff	ffff
----	------	------	------

  
 描述: 寄存器“f”的内容带进位标志位循环右移1位。如果“d”为0, 结果存入W寄存器。如果“d”为1, 结果存回“f”寄存器。



指令字数: 1  
 指令周期: 1  
 示例: RRF REG1, 0

执行指令前  
 REG1 = 1110 0110  
 C = 0  
 执行指令后  
 REG1 = 1110 0110  
 W = 0111 0011  
 C = 0

## SLEEP

语法: [label] SLEEP  
 操作数: 无  
 操作: 00h → WDT,  
 0 → WDT 预分频器,  
 1 →  $\overline{TO}$ ,  
 0 → PD  
 受影响的状态位:  $\overline{TO}$ ,  $\overline{PD}$   
 机器码: 

00	0000	0110	0011
----	------	------	------

描述: 掉电状态位 ( $\overline{PD}$ ) 清零。超时状态位 ( $\overline{TO}$ ) 置位。看门狗及其预分频器清零。振荡器停振, 单片机进入休眠模式。详细情况, 请参见第 14.8 节“低功耗 (休眠) 模式”。  
 指令字数: 1  
 指令周期: 1  
 示例: SLEEP

## SUBLW 立即数减 W

语法: [label] SUBLW k  
 操作数:  $0 \leq k \leq 255$   
 操作:  $k-(W) \rightarrow (W)$   
 受影响的状态位: C、DC、Z  
 机器码: 

11	110x	kkkk	kkkk
----	------	------	------

  
 描述: 将 8 位立即数“k”减去 W 寄存器的内容 (通过二进制补码方法进行运算)。结果存入 W 寄存器。

指令字数: 1  
 指令周期: 1  
 示例 1: SUBLW 0x02

执行指令前  
 W = 1  
 C = ?  
 执行指令后  
 W = 1  
 C = 1; 结果为正

示例 2: 执行指令前  
 W = 2  
 C = ?  
 执行指令后  
 W = 0  
 C = 1; 结果为零

示例 3: 执行指令前  
 W = 3  
 C = ?  
 执行指令后  
 W = 0xFF  
 C = 0; 结果为负



# PIC16F627A/628A/648A

## SUBWF f 减 W

语法: [label] SUBWF f,d

操作数:  $0 \leq f \leq 127$   
 $d \in [0,1]$

操作: (f)-(W) → (dest)

受影响的状态位: C、DC 和 Z

机器码:	00	0010	dfff	ffff
------	----	------	------	------

描述: 用寄存器“f”减去 W 寄存器的内容（采用二进制补码方式进行）。如果“d”为 0，结果存入 W 寄存器。如果“d”为 1，结果存回“f”寄存器。

指令字数: 1

指令周期: 1

示例 1: SUBWF REG1, 1

执行指令前

REG1 = 3  
W = 2  
C = ?

执行指令后

REG1 = 1  
W = 2  
C = 1; 结果为正  
DC = 1  
Z = 0

示例 2: 执行指令前

REG1 = 2  
W = 2  
C = ?

执行指令后

REG1 = 0  
W = 2  
C = 1; 结果为零  
Z = DC=1

示例 3: 执行指令前

REG1 = 1  
W = 2  
C = ?

执行指令后

REG1 = 0xFF  
W = 2  
C = 0; 结果为负  
Z = DC=0

## SWAPF f 半字节交换

语法: [label] SWAPF f,d

操作数:  $0 \leq f \leq 127$   
 $d \in [0,1]$

操作: (f<3:0>) → (dest<7:4>),  
(f<7:4>) → (dest<3:0>)

受影响的状态位: 无

机器码:	00	1110	dfff	ffff
------	----	------	------	------

描述: 交换“f”寄存器的高半字节和低半字节。如果“d”为 0，结果存入 W 寄存器。如果“d”为 1，结果存回“f”寄存器。

指令字数: 1

指令周期: 1

示例: SWAPF REG1, 0

执行指令前

REG1 = 0xA5

执行指令后

REG1 = 0xA5  
W = 0x5A

TRIS	装载 TRIS 寄存器				
语法:	[label] TRIS f				
操作数:	$5 \leq f \leq 7$				
操作:	(W) → TRIS 寄存器 f;				
受影响的状态位:	无				
机器码:	<table border="1"> <tr> <td>00</td> <td>0000</td> <td>0110</td> <td>0fff</td> </tr> </table>	00	0000	0110	0fff
00	0000	0110	0fff		
描述:	支持该指令是为了与 PIC16C5X 产品代码兼容。TRIS 是可读写的寄存器，用户可对其直接寻址。				
指令字数:	1				
指令周期:	1				
示例:	<div style="border: 1px solid black; padding: 5px;">                     为保持与将来 PICmicro® 单片机的向上兼容性，不要使用该指令。                 </div>				

# PIC16F627A/628A/648A

## XORLW 立即数与 W 进行“异或”运算

语法: [label] XORLW k

操作数:  $0 \leq k \leq 255$

操作: (W).OR. k  $\rightarrow$  (W)

受影响的状态位: Z

机器码: 

11	1010	kkkk	kkkk
----	------	------	------

描述: W 寄存器的内容与 8 位立即数“k”进行异或运算。结果存入 W 寄存器。

指令字数: 1

指令周期: 1

示例: XORLW 0xAF

执行指令前

W = 0xB5

执行指令后

W = 0x1A

## XORWF W 和 f 进行“异或”运算

语法: [label] XORWF f,d

操作数:  $0 \leq f \leq 127$   
 $d \in [0,1]$

操作: (W).XOR. (f)  $\rightarrow$  (dest)

受影响的状态位: Z

机器码: 

00	0110	dfff	ffff
----	------	------	------

描述: W 寄存器的内容与“f”寄存器的内容进行异或运算。如果“d”为 0, 结果存入 W 寄存器。如果“d”为 1, 结果存回“f”寄存器。

指令字数: 1

指令周期: 1

示例: XORWF REG1, 1

执行指令前

REG1 = 0xAF

W = 0xB5

执行指令后

REG1 = 0x1A

W = 0xB5

## 16.0 开发支持

一系列硬件及软件开发工具对 PICmicro® 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 ICE 4000 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
- 低成本演示和开发板及评估工具包

## 16.1 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 PICmicro 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。

# PIC16F627A/628A/648A

---

## 16.2 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于所有的 PICmicro 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

## 16.3 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 系列单片机和 dsPIC30F 系列数据信号控制器。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。

## 16.4 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

## 16.5 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

## 16.6 MPLAB SIM 软件模拟器

MPLAB SIM 软件模拟器在指令级对 PICmicro 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 的动作以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 MPLAB ASM30 汇编器的符号调试。该软件模拟器可用于在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。

## 16.7 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 PICmicro 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 PICmicro 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。

## 16.8 MPLAB ICE 4000 高性能在线仿真器

MPLAB ICE 4000 在线仿真器旨在为产品开发工程师提供一整套用于高端 PICmicro MCU 和 dsPIC DSC 的设计工具。MPLAB ICE 4000 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 4000 是高级的仿真系统，除具备 MPLAB ICE 2000 的所有功能外，它还增加了适用于 dsPIC30F 和 PIC18XXXX 器件的仿真存储容量以及高速性能。该仿真器的先进特性包括复杂触发和定时功能及高达 2 Mb 的仿真存储容量。

MPLAB ICE 4000 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在更加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程序中得以很好的利用。

## 16.9 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 PICmicro MCU，可用于开发本系列及其他 PICmicro 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 PICmicro 器件的开发编程器。

## 16.10 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 VDDMIN 和 VDDMAX 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 PICmicro 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。

# PIC16F627A/628A/648A

---

## 16.11 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 PICmicro 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。

## 16.12 演示、开发和评估板

有许多演示、开发和评估板可用于各种 PICmicro 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、 $\Sigma$ - $\Delta$  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](http://www.microchip.com)) 以及最新的 “*Product Selector Guide (产品选型指南)*” (DS00148)。

注:

# PIC16F627A/628A/648A

---

---

注:



## 17.0 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
VDD 相对于 VSS 的电压.....	-0.3 至 +6.5V
MCLR 和 RA4 相对于 VSS 的电压.....	-0.3 至 +14V
所有其他引脚相对于 VSS 的电压.....	-0.3V 至 VDD+ 0.3V
总功耗 (1).....	800 mW
VSS 引脚最大电流.....	300 mA
VDD 引脚的最大电流.....	250 mA
输入箝位电流, I <sub>IK</sub> (V <sub>I</sub> < 0 或 V <sub>I</sub> > VDD).....	± 20 mA
输出箝位电流, I <sub>OK</sub> (V <sub>O</sub> < 0 或 V <sub>O</sub> > VDD).....	± 20 mA
任一 I/O 引脚的最大灌电流.....	25 mA
任一 I/O 引脚的最大拉电流.....	25 mA
PORTA 和 PORTB 的最大灌电流总和.....	200 mA
PORTA 和 PORTB 的最大拉电流总和.....	200 mA

注 1: 功耗按如下公式计算:  $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$

†注意: 如果运行条件超过上述“绝对最大额定值”可能会对器件造成永久的损坏。上述值仅为运行条件极大值, 我们不建议器件在该极大值或该规范范围以外运行。长时间运行在最大额定值可能会影响器件的可靠性。

注: 如果 MCLR 引脚上的尖峰电压低于 VSS, 会导致感应电流大于 80 mA, 可能引起锁死。因此, 如果要在 MCLR 引脚施加“低电平”, 应串连一个 50-100 Ω 的电阻, 而不是将引脚直接拉到 VSS。

# PIC16F627A/628A/648A

图 17-1: PIC16F627A/628A/648A 电压 - 频率关系图,  $-40^{\circ}\text{C} \leq \text{TA} \leq +125^{\circ}\text{C}$

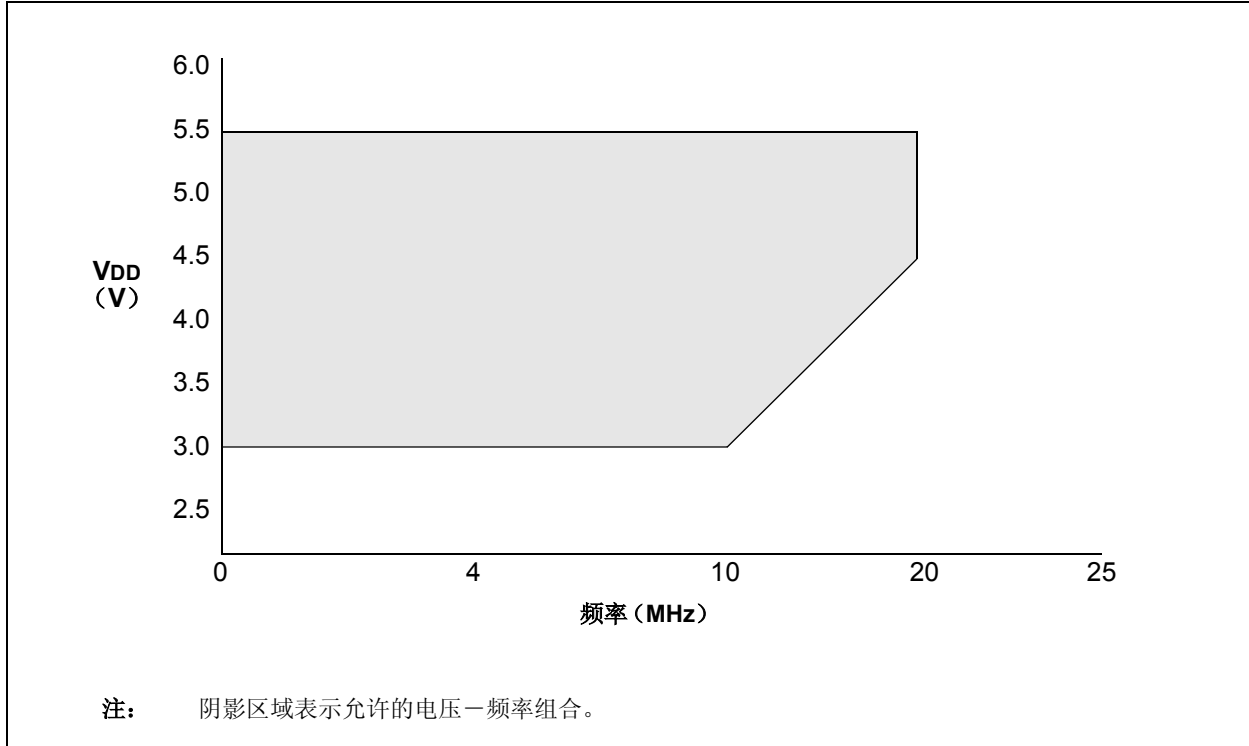
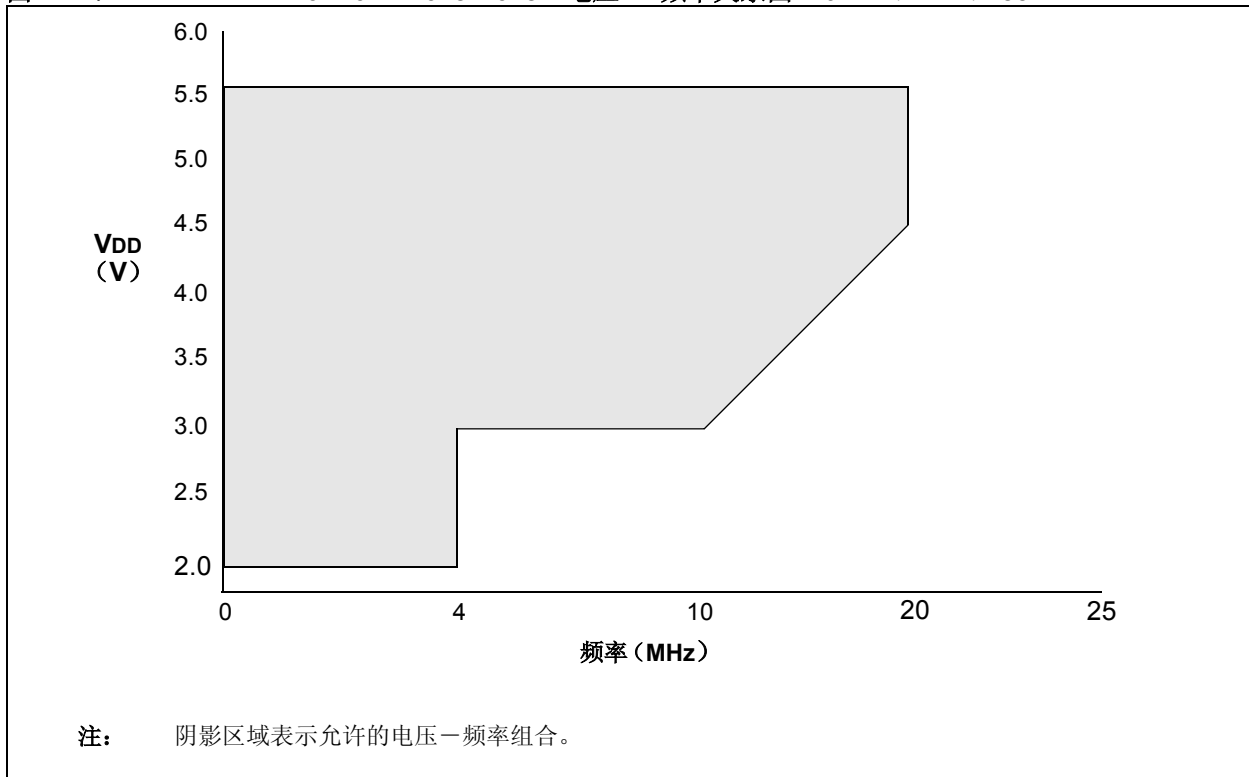


图 17-2: PIC16LF627A/628A/648A 电压 - 频率关系图  $-40^{\circ}\text{C} \leq \text{TA} \leq +85^{\circ}\text{C}$



# PIC16F627A/628A/648A

## 17.1 DC 特性: PIC16F627A/628A/648A (工业级, 扩展级) PIC16LF627A/628A/648A (工业级)

PIC16LF627A/628A/648A (工业级)		标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_a \leq +85^{\circ}\text{C}$ 工业级					
PIC16F627A/628A/648A (工业级, 扩展级)		标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_a \leq +85^{\circ}\text{C}$ 工业级和 $-40^{\circ}\text{C} \leq T_a \leq +125^{\circ}\text{C}$ 扩展级					
参数编号	符号	特性 / 器件	最小值	典型值	最大值	单位	条件
D001	VDD	电源电压					
		PIC16LF627A/628A/648A	2.0	—	5.5	V	
		PIC16F627A/628A/648A	3.0	—	5.5	V	
D002	VDR	RAM 数据保持电压 <sup>(1)</sup>	—	1.5*	—	V	器件工作在休眠模式
D003	VPOR	确保上电复位的 VDD 起始电压	—	VSS	—	V	有关上电复位的细节请参阅第 14.4 节“上电复位 (POR)、上电延时定时器 (PWRT)、振荡器起振定时器 (OST) 和欠压复位 (BOR)”
D004	SVDD	确保上电复位的 VDD 上升率	0.05*	—	—	V/ms	有关上电复位的细节请参阅第 14.4 节“上电复位 (POR)、上电延时定时器 (PWRT)、振荡器起振定时器 (OST) 和欠压复位 (BOR)”
D005	VBOR	欠压复位电压	3.65	4.0	4.35	V	BOREN 配置位置位
			3.65	4.0	4.4	V	BOREN 配置位置位, 扩展级

图注: 仅标准电压器件数据的行用阴影表示, 以提高可读性。

\* 这些参数为特性值, 未经测试。

† 除非另外说明, 否则在“典型值”栏中的数据都是在 5.0 V、25°C 的条件下测得的。这些参数仅供设计参考, 未经测试。

注 1: 这是在休眠模式下, 不丢失 RAM 数据的 VDD 电压下限。

# PIC16F627A/628A/648A

## 17.2 DC 特性: PIC16F627A/628A/648A (工业级)

### PIC16LF627A/628A/648A (工业级)

DC 特性		标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_a \leq +85^{\circ}\text{C}$ 工业级					
参数 编号	LF 和 F 器件特性	最小 值†	典型 值	最大 值	单位	条件	
						VDD	注
<b>电源电压 (VDD)</b>							
D001	LF	2.0	—	5.5	V	—	
	LF/F	3.0	—	5.5	V	—	
<b>断电基本电流 (IPD)</b>							
D020	LF	—	0.01	0.80	$\mu\text{A}$	2.0	WDT、BOR、比较器、VREF 和 T1OSC: 禁止
	LF/F	—	0.01	0.85	$\mu\text{A}$	3.0	
		—	0.02	2.7	$\mu\text{A}$	5.0	
<b>外设模块电流 (<math>\Delta\text{IMOD}</math>)<sup>(1)</sup></b>							
D021	LF	—	1	2.0	$\mu\text{A}$	2.0	WDT 电流
	LF/F	—	2	3.4	$\mu\text{A}$	3.0	
		—	9	17.0	$\mu\text{A}$	5.0	
D022	LF/F	—	29	52	$\mu\text{A}$	4.5	BOR 电流
		—	30	55	$\mu\text{A}$	5.0	
D023	LF	—	15	22	$\mu\text{A}$	2.0	比较器电流 (两个比较器都使能)
	LF/F	—	22	37	$\mu\text{A}$	3.0	
		—	24	68	$\mu\text{A}$	5.0	
D024	LF	—	34	55	$\mu\text{A}$	2.0	VREF 电流
	LF/F	—	50	75	$\mu\text{A}$	3.0	
		—	80	110	$\mu\text{A}$	5.0	
D025	LF	—	1.2	2.0	$\mu\text{A}$	2.0	T1Osc 电流
	LF/F	—	1.3	2.2	$\mu\text{A}$	3.0	
		—	1.8	2.9	$\mu\text{A}$	5.0	
<b>供电电流 (IDD)</b>							
D010	LF	—	10	15	$\mu\text{A}$	2.0	Fosc = 32 kHz LP 振荡器模式
	LF/F	—	15	25	$\mu\text{A}$	3.0	
		—	18	48	$\mu\text{A}$	5.0	
D011	LF	—	125	190	$\mu\text{A}$	2.0	Fosc = 1 MHz XT 振荡器模式
	LF/F	—	175	340	$\mu\text{A}$	3.0	
		—	320	520	$\mu\text{A}$	5.0	
D012	LF	—	250	350	$\mu\text{A}$	2.0	Fosc = 4 MHz XT 振荡器模式
	LF/F	—	450	600	$\mu\text{A}$	3.0	
		—	710	995	$\mu\text{A}$	5.0	
D012A	LF	—	395	465	$\mu\text{A}$	2.0	Fosc = 4 MHz INTOSC
	LF/F	—	565	785	$\mu\text{A}$	3.0	
		—	0.895	1.3	mA	5.0	
D013	LF/F	—	2.5	2.9	mA	4.5	Fosc = 20 MHz HS 振荡器模式
		—	2.75	3.3	mA	5.0	

注 1: “ $\Delta$ ” 电流是外设使能时的额外消耗电流。在测量基本电流 IDD 或 IPD 时, 应该加上这个电流。在计算总的电流消耗时应使用最大值。

# PIC16F627A/628A/648A

## 17.3 DC 特性: PIC16F627A/628A/648A (扩展级)

DC 特性		标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_a \leq +125^{\circ}\text{C}$ 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
<b>电源电压 (VDD)</b>							
D001	—	3.0	—	5.5	V	—	
<b>断电基本电流 (IPD)</b>							
D020E	—	—	0.01	4	$\mu\text{A}$	3.0	WDT、BOR、比较器、VREF 和 T1OSC: 禁止
		—	0.02	8	$\mu\text{A}$	5.0	
<b>外设模块电流 (<math>\Delta\text{IMOD}</math>)<sup>(1)</sup></b>							
D021E	—	—	2	9	$\mu\text{A}$	3.0	WDT 电流
		—	9	20	$\mu\text{A}$	5.0	
D022E	—	—	29	52	$\mu\text{A}$	4.5	BOR 电流
		—	30	55	$\mu\text{A}$	5.0	
D023E	—	—	22	37	$\mu\text{A}$	3.0	比较器电流 (两个比较器都使能)
		—	44	68	$\mu\text{A}$	5.0	
D024E	—	—	50	75	$\mu\text{A}$	3.0	VREF 电流
		—	83	110	$\mu\text{A}$	5.0	
D025E	—	—	1.3	4	$\mu\text{A}$	3.0	T1Osc 电流
		—	1.8	6	$\mu\text{A}$	5.0	
<b>供电电流 (IDD)</b>							
D010E	—	—	15	28	$\mu\text{A}$	3.0	Fosc = 32 kHz LP 振荡器模式
		—	28	54	$\mu\text{A}$	5.0	
D011E	—	—	175	340	$\mu\text{A}$	3.0	Fosc = 1 MHz XT 振荡器模式
		—	320	520	$\mu\text{A}$	5.0	
D012E	—	—	450	650	$\mu\text{A}$	3.0	Fosc = 4 MHz XT 振荡器模式
		—	0.710	1.1	mA	5.0	
D012AE	—	—	565	785	$\mu\text{A}$	3.0	Fosc = 4 MHz INTOSC
		—	0.895	1.3	mA	5.0	
D013E	—	—	2.5	2.9	mA	4.5	Fosc = 20 MHz HS 振荡器模式
		—	2.75	3.5	mA	5.0	

**注 1:** “ $\Delta$ ” 电流是此外设使能时的额外消耗电流。在测量基本电流 IDD 或 IPD 时, 应该加上这个电流。在计算总的电流消耗时应使用最大值。

# PIC16F627A/628A/648A

## 17.4 DC 特性: PIC16F627A/628A/648A (工业级, 扩展级) PIC16LF627A/628A/648A (工业级)

DC 特性		标准工作条件 (除非另有声明)					
		工作温度 -40°C ≤ TA ≤ +85°C 工业级和 -40°C ≤ TA ≤ +125°C 扩展级					
		工作电压 V <sub>DD</sub> 范围如 DC 规范表 17-2 和表 17-3 所述					
参数编号	符号	特性 / 器件	最小值	典型值 †	最大值	单位	条件
	V <sub>IL</sub>	<b>输入低电压</b>					
D030		I/O 端口 带 TTL 缓冲	V <sub>SS</sub>	—	0.8	V	V <sub>DD</sub> = 4.5V 至 5.5V 否则 (注 1)
D031		带施密特触发器输入 (4)	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	
D032		MCLR、RA4/T0CKI、OSC1 (在 RC 模式下)	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	
D033		OSC1 (在 HS 模式下)	V <sub>SS</sub>	—	0.3 V <sub>DD</sub>	V	
		OSC1 (在 LP 和 XT 模式下)	V <sub>SS</sub>	—	0.6	V	
	V <sub>IH</sub>	<b>输入高电压</b>					
D040		I/O 端口 带 TTL 缓冲	2.0V	—	V <sub>DD</sub>	V	V <sub>DD</sub> = 4.5V 至 5.5V 否则 (注 1)
D041		带施密特触发器输入 (4)	.25 V <sub>DD</sub> + 0.8V	—	V <sub>DD</sub>	V	
D042		MCLR RA4/T0CKI	0.8 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
D043		OSC1 (XT 和 LP)	1.3	—	V <sub>DD</sub>	V	
D043A		OSC1 (在 RC 模式下)	0.9 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
D043B		OSC1 (在 HS 模式下)	0.7 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
D070	IPURB	<b>PORTB 弱上拉 电流</b>	50	200	400	μA	V <sub>DD</sub> = 5.0V, V <sub>PIN</sub> = V <sub>SS</sub>
	I <sub>IL</sub>	<b>输入泄漏电流 (2), (3)</b>					
D060		I/O 端口 (PORTA 除外) PORTA (4)	—	—	±1.0	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , 引脚处于高阻态 V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , 引脚处于高阻态 V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , XT、HS 和 LP 振荡器配置
D061		RA4/T0CKI	—	—	±0.5	μA	
D063		OSC1, MCLR	—	—	±1.0	μA	
				—	—	±5.0	
	V <sub>OL</sub>	<b>输出低电压</b>					
D080		I/O 端口 (4)	—	—	0.6	V	I <sub>OL</sub> =8.5 mA, V <sub>DD</sub> =4.5 V, -40°C 至 +85°C
			—	—	0.6	V	I <sub>OL</sub> =7.0 mA, V <sub>DD</sub> =4.5 V, +85°C 至 +125°C
	V <sub>OH</sub>	<b>输出高电压 (3)</b>					
D090		I/O 端口 (RA4 (4) 除外)	V <sub>DD</sub> -0.7	—	—	V	I <sub>OH</sub> =-3.0 mA, V <sub>DD</sub> =4.5 V, -40°C 至 +85°C
			V <sub>DD</sub> -0.7	—	—	V	I <sub>OH</sub> =-2.5 mA, V <sub>DD</sub> =4.5 V, +85°C 至 +125°C
D150	V <sub>OD</sub>	<b>开漏高电压</b>	—	—	8.5*	V	RA4 引脚 PIC16F627A/628A/648A, PIC16LF627A/628A/648A
		<b>输出引脚上容性负载规范</b>					
D100*	C <sub>OSC2</sub>	OSC2 引脚	—	—	15	pF	在 XT、HS 和 LP 模式下, 当使用外部时钟来驱动 OSC1 时。
D101*	C <sub>IO</sub>	所有 I/O 引脚 / OSC2 (在 RC 模式下)	—	—	50	pF	

\* 这些参数为特性值, 未经测试。

† 除非另外说明, 否则“典型值”栏中的数据都是在 5.0V、25°C 的条件下测得的。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 引脚为施密特触发器输入引脚。在 RC 模式下, 建议不要使用外部时钟驱动 PIC16F627A/628A/648A。  
 2: MCLR 引脚上的泄漏电流很大程度上取决于施加的电平。规定的电平表示正常工作条件。可以在不同的输入电压下测得更高的泄漏电流。  
 3: 负电流定义为自引脚流出的电流。  
 4: 配置为 I/O 引脚、CLKIN 或 CLKOUT 时, 包括 OSC1 和 OSC2。

# PIC16F627A/628A/648A

表 17-1: DC 特性: PIC16F627A/628A/648A (工业级, 扩展级)  
PIC16LF627A/628A/648A (工业级)

DC 特性		标准工作条件 (除非另有声明)					
		工作温度	-40°C ≤ TA ≤ +85°C 工业级和 -40°C ≤ TA ≤ +125°C 扩展级				
		工作电压 VDD	范围如 DC 规范表 17-2 和表 17-3 所述				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
<b>数据 EEPROM 存储器</b>							
D120	ED	耐用性	100K	1M	—	E/W	-40°C ≤ TA ≤ 85°C
D120A	ED	耐用性	10K	100K	—	E/W	85°C ≤ TA ≤ 125°C
D121	VDRW	读写操作时的 VDD	VMIN	—	5.5	V	VMIN 为最小工作电压
D122	TDEW	擦写周期时间	—	4	8*	ms	
D123	TRETD	特性保持	40	—	—	年	假如没有违反其他规范
D124	TREF	刷新之前总的擦写次数 <sup>(1)</sup>	1M	10M	—	E/W	-40°C 至 +85°C
<b>闪存程序存储器</b>							
D130	EP	耐用性	10K	100K	—	E/W	-40°C ≤ TA ≤ 85°C
D130A	EP	耐用性	1000	10K	—	E/W	85°C ≤ TA ≤ 125°C
D131	VPR	读操作时的 VDD	VMIN	—	5.5	V	VMIN 为最小工作电压
D132	VIE	块擦除时的 VDD	4.5	—	5.5	V	
D132A	VPEW	写操作时的 VDD	VMIN	—	5.5	V	VMIN 为最小工作电压
D133	TIE	块擦除周期时间	—	4	8*	ms	VDD > 4.5V
D133A	TPEW	写周期时间	—	2	4*	ms	
D134	TRETP	特性保持	40	—	—	年	假如没有违反其他规范

\* 这些参数为特性值, 未经测试。

† 除非另外说明, 否则“典型值”栏中的数据都是在 5.0 V、25°C 的条件下测得的。这些参数仅供设计参考, 未经测试。

注 1: 有关数据 EEPROM 耐用性的更多细节, 请参阅第 13.7 节“使用数据 EEPROM”。

# PIC16F627A/628A/648A

**表 17-2: 比较器规范**

工作条件: $2.0V < V_{DD} < 5.5V$ , $-40^{\circ}C < T_A < +125^{\circ}C$ , 除非另有声明。							
参数编号	特性	符号	最小值	典型值	最大值	单位	备注
D300	输入失调电压	V <sub>IOFF</sub>	—	±5.0	±10	mV	
D301	输入共模电压	V <sub>ICM</sub>	0	—	$V_{DD} - 1.5^*$	V	
D302	共模抑制比	CMRR	55*	—	—	db	
D303	响应时间 <sup>(1)</sup>	T <sub>RESP</sub>	—	300	400*	ns	V <sub>DD</sub> = 3.0V 至 5.5V -40°C 至 +85°C V <sub>DD</sub> = 3.0V 至 5.5V -85°C 至 +125°C V <sub>DD</sub> = 2.0V 至 3.0V -40°C 至 +85°C
			—	400	600*	ns	
			—	400	600*	ns	
D304	比较器模式变为输出有效	T <sub>MC2OV</sub>	—	300	10*	μs	

\* 这些参数为特性值, 未经测试。

注 1: 当一个比较器输入电压为  $(V_{DD} - 1.5) / 2$ , 而另一个比较器输入从 V<sub>SS</sub> 跳变到 V<sub>DD</sub> 时所测得的响应时间。

**表 17-3: 参考电压规范**

工作条件: $2.0V < V_{DD} < 5.5V$ , $-40^{\circ}C < T_A < +125^{\circ}C$ , 除非另有声明。							
参数编号	特性	符号	最小值	典型值	最大值	单位	备注
D310	精度	V <sub>RES</sub>	—	—	V <sub>DD</sub> /24	LSb	低量程 (VRR = 1) 高量程 (VRR = 0)
					V <sub>DD</sub> /32	LSb	
D311	绝对精度	V <sub>RAA</sub>	—	—	$1/4^{(2)*}$	LSb	低量程 (VRR = 1) 高量程 (VRR = 0)
					$1/2^{(2)*}$	LSb	
D312	单位电阻值 (R)	V <sub>RUR</sub>	—	2k*	—	Ω	
D313	稳定时间 <sup>(1)</sup>	T <sub>SET</sub>	—	—	10*	μs	

\* 这些参数为特性值, 未经测试。

注 1: 稳定时间是在 VRR = 1 且 VR<3:0> 从 0000 跳变到 1111 时测得的。

注 2: 当 V<sub>DD</sub> 在 2.0V 到 3.0V 之间时, RA2 上的 V<sub>REF</sub> 输出电平用以下公式计算:  $[V_{DD}/2 \pm (3 - V_{DD})/2]$ , 它可能导致 RA2 上 V<sub>REF</sub> 输出信号的绝对精度 (V<sub>RAA</sub>) 大于规范中所给出的最大值。



## 17.5 时序参数符号

时序参数符号根据以下格式中的一种创建：

1. TppS2ppS
2. TppS

<b>T</b>			
F	频率	T	时间

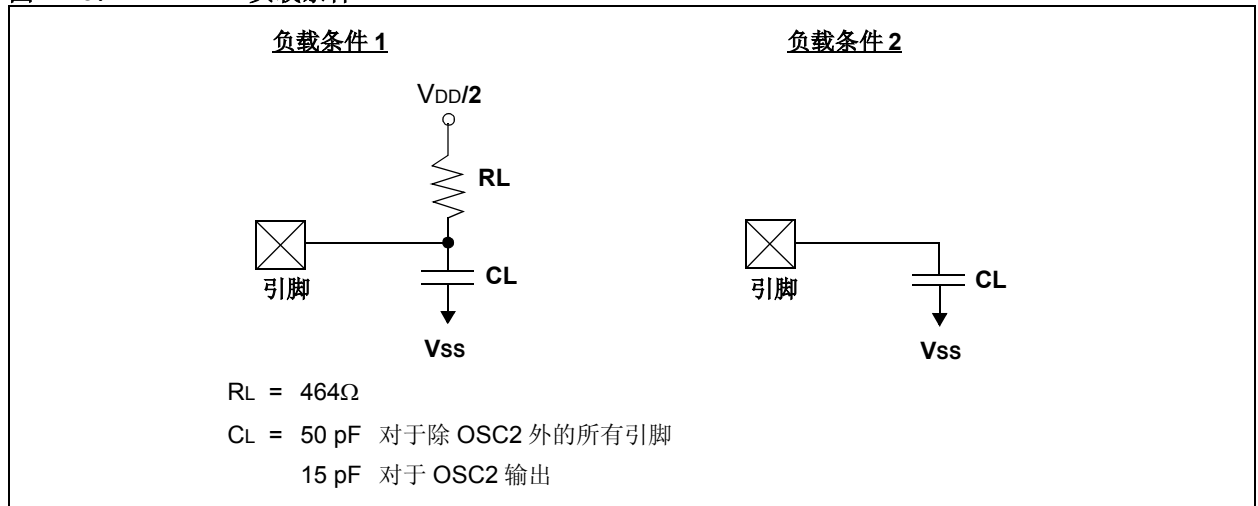
小写下标 (pp) 及其含义：

<b>pp</b>			
ck	CLKOUT	osc	OSC1
io	I/O 端口	t0	T0CKI
mc	MCLR		

大写字母及其含意：

<b>S</b>			
F	下降	P	周期
H	高电平	R	上升
I	无效 (高阻态)	V	有效
L	低电平	Z	高阻态

**图 17-3: 负载条件**



# PIC16F627A/628A/648A

## 17.6 时序图和规范

图 17-4: 外部时钟时序

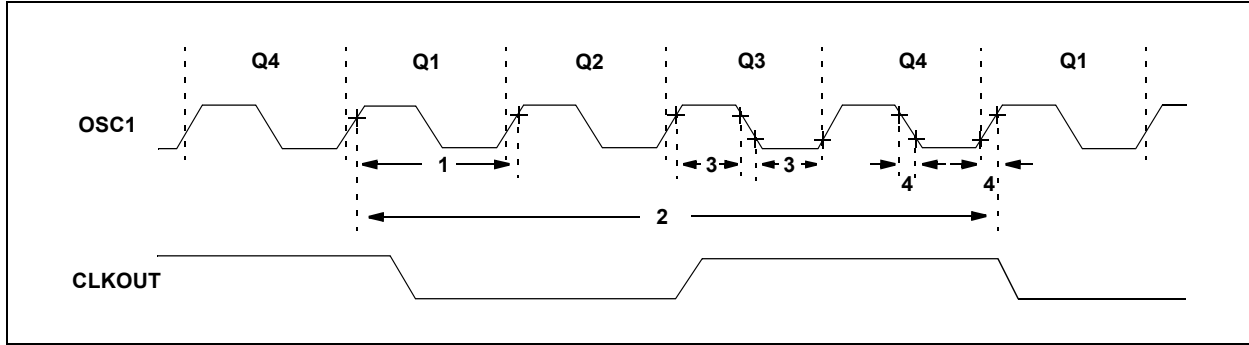


表 17-4: 外部时钟时序要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
	$F_{OSC}$	外部 CLKIN 频率 (1)	DC	—	4	MHz	XT 和 RC 振荡器模式, $V_{DD} = 5.0\text{ V}$
			DC	—	20	MHz	HS 和 EC 振荡器模式
			DC	—	200	kHz	LP 振荡器模式
		振荡器频率 (1)	—	—	4	MHz	RC 振荡器模式 $V_{DD} = 5.0\text{ V}$
			0.1	—	4	MHz	XT 振荡器模式
			1	—	20	MHz	HS 振荡器模式
			—	—	200	kHz	LP 振荡器模式
			—	4	—	MHz	INTOSC 模式 (快)
			—	48	—	kHz	INTOSC 模式 (慢)
			—	—	—	—	—
1	$T_{OSC}$	外部 CLKIN 周期 (1)	250	—	—	ns	XT 和 RC 振荡器模式
			50	—	—	ns	HS 和 EC 振荡器模式
			5	—	—	$\mu\text{s}$	LP 振荡器模式
		振荡器周期 (1)	250	—	—	ns	RC 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	
		5	—	—	$\mu\text{s}$	LP 振荡器模式	
		—	250	—	ns	INTOSC 模式 (快)	
—	21	—	$\mu\text{s}$	INTOSC 模式 (慢)			
2	$T_{CY}$	指令周期时间	200	$T_{CY}$	DC	ns	$T_{CY} = 4/F_{OSC}$
3	$T_{oS_L}$ , $T_{oS_H}$	外部 CLKIN (OSC1) 高电平 外部 CLKIN 低电平	100*	—	—	ns	XT 振荡器, $T_{oS_C}$ L/H 占空比
4	RC	外部 RC 频率	10 kHz*	—	4 MHz	—	$V_{DD} = 5.0\text{ V}$

\* 这些参数为特性值, 未经测试。

† 除非另外说明, 否则“典型值”栏中的数据都是在 5.0V、25°C 的条件下测得的。这些参数仅供设计参考, 未经测试。

**注 1:** 指令周期时间 ( $T_{CY}$ ) 等于输入振荡器基本时基周期的四倍。所有规定值都基于标准运行条件下器件代码执行过程中特定振荡器类型的特征数据。超出这些规定的限定值可能导致振荡器运行不稳定 / 或导致电流消耗超出预期值。所有器件的测试都是在“最小”值条件下, 在 OSC1 引脚施加外部时钟进行的。对于所有器件, 当使用外部时钟输入时, “最大”周期时间限制为“DC”(无时钟)。

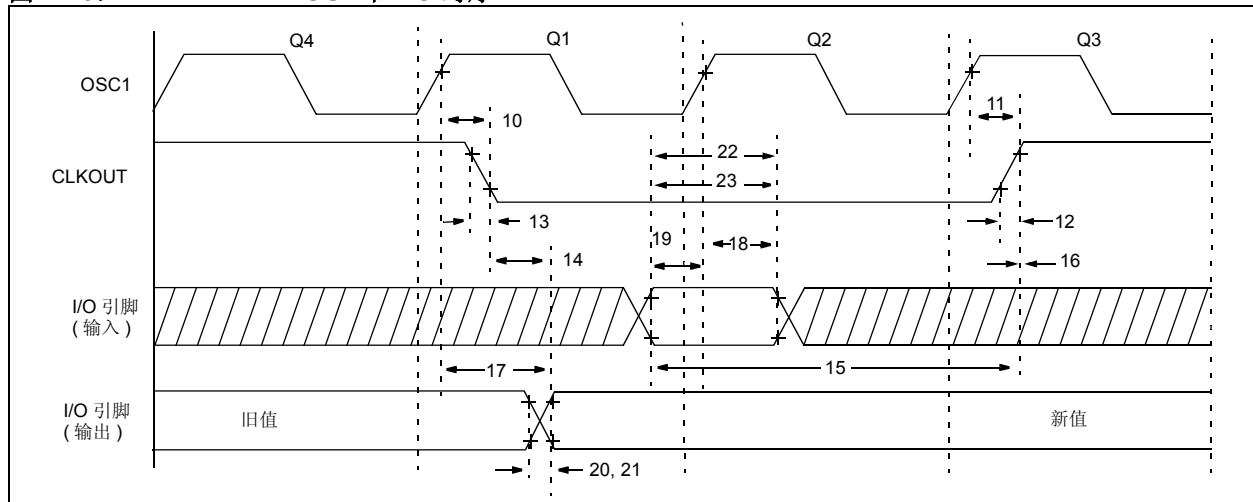
**表 17-5: 高精度内部振荡器参数**

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
F10	$f_{osc}$	振荡器中心频率	—	4	—	MHz	
F13	$\Delta f_{osc}$	振荡器精度	3.96	4	4.04	MHz	$V_{DD} = 3.5\text{ V}$ , $25^\circ\text{ C}$
			3.92	4	4.08	MHz	$2.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ $0^\circ\text{ C} \leq T_A \leq +85^\circ\text{ C}$
			3.80	4	4.20	MHz	$2.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ $-40^\circ\text{ C} \leq T_A \leq +85^\circ\text{ C (IND)}$ $-40^\circ\text{ C} \leq T_A \leq +125^\circ\text{ C (EXT)}$
F14*	$T_{oscst}$	从休眠模式唤醒的振荡器起振时间	—	6	8	$\mu\text{S}$	$V_{DD} = 2.0\text{ V}$ , $-40^\circ\text{ C}$ 到 $+85^\circ\text{ C}$
			—	4	6	$\mu\text{S}$	$V_{DD} = 3.0\text{ V}$ , $-40^\circ\text{ C}$ 到 $+85^\circ\text{ C}$
			—	3	5	$\mu\text{S}$	$V_{DD} = 5.0\text{ V}$ , $-40^\circ\text{ C}$ 到 $+85^\circ\text{ C}$

图注: TBD = 待定。

\* 特性数据, 未经测试。

**图 17-5: CLKOUT 和 I/O 时序**



# PIC16F627A/628A/648A

表 17-6: CLKOUT 和 I/O 时序要求

参数编号	符号	特性	最小值	典型值 †	最大值	单位	
10	TosH2ckL	OSC1 ↑ 到 CLKOUT ↓	PIC16F62XA	—	75	200*	
10A			PIC16LF62XA	—	—	400*	
11	TosH2ckH	OSC1 ↑ 到 CLKOUT ↑	PIC16F62XA	—	75	200*	
11A			PIC16LF62XA	—	—	400*	
12	TckR	CLKOUT 上升时间	PIC16F62XA	—	35	100*	
12A			PIC16LF62XA	—	—	200*	
13	TckF	CLKOUT 下降时间	PIC16F62XA	—	35	100*	
13A			PIC16LF62XA	—	—	200*	
14	TckL2ioV	CLKOUT ↓ 到端口输出有效	—	—	20*	ns	
15	TioV2ckH	在 CLKOUT ↑ 之前端口输入有效	PIC16F62XA	Tosc+200 ns*	—	—	ns
			PIC16LF62XA	Tosc+400 ns*	—	—	ns
16	TckH2ioI	在 CLKOUT ↑ 之后保持端口输入	0	—	—	ns	
17	TosH2ioV	OSC1 ↑ (Q1 周期) 到端口输出有效	PIC16F62XA	—	50	150*	ns
			PIC16LF62XA	—	—	300*	ns
18	TosH2ioI	OSC1 ↑ (Q2 周期) 到端口输入无效 (I/O 输入保持时间)	100*	—	—	ns	
			200*	—	—	ns	

\* 这些参数为特性值，未经测试。

† 除非另外说明，否则“典型值”栏中的数据都是在 5.0V、25°C 的条件下测得的。这些参数仅供设计参考，未经测试。

图 17-6: 复位、看门狗、振荡器起振定时器和上电延时定时器时序

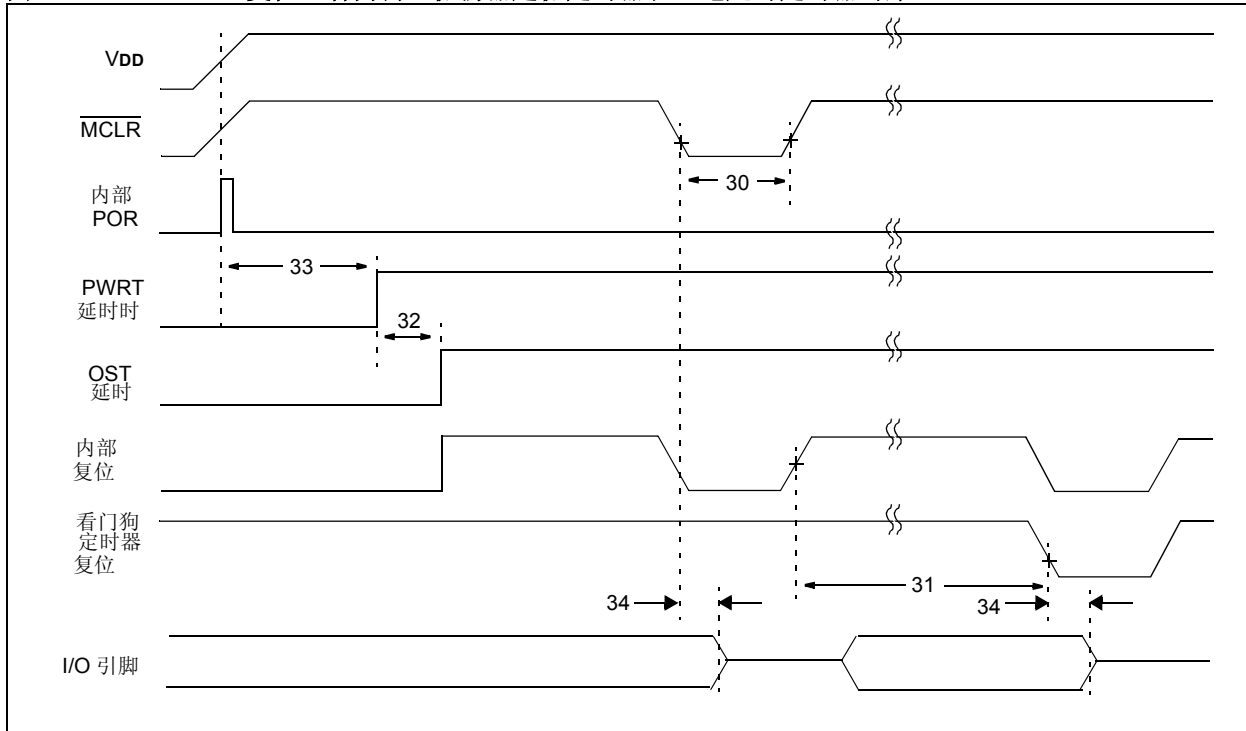


图 17-7: 欠压复位时序

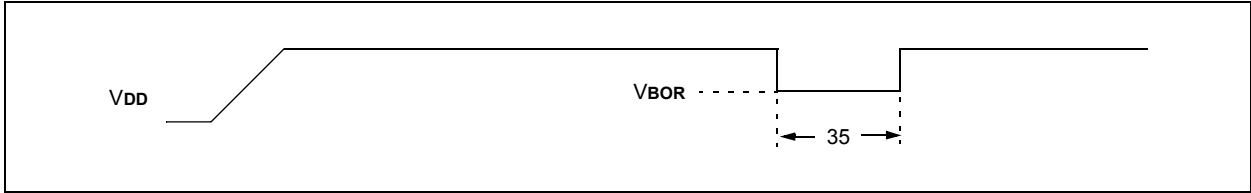


表 17-7: 复位、看门狗、振荡器起振定时器和上电延时定时器时序

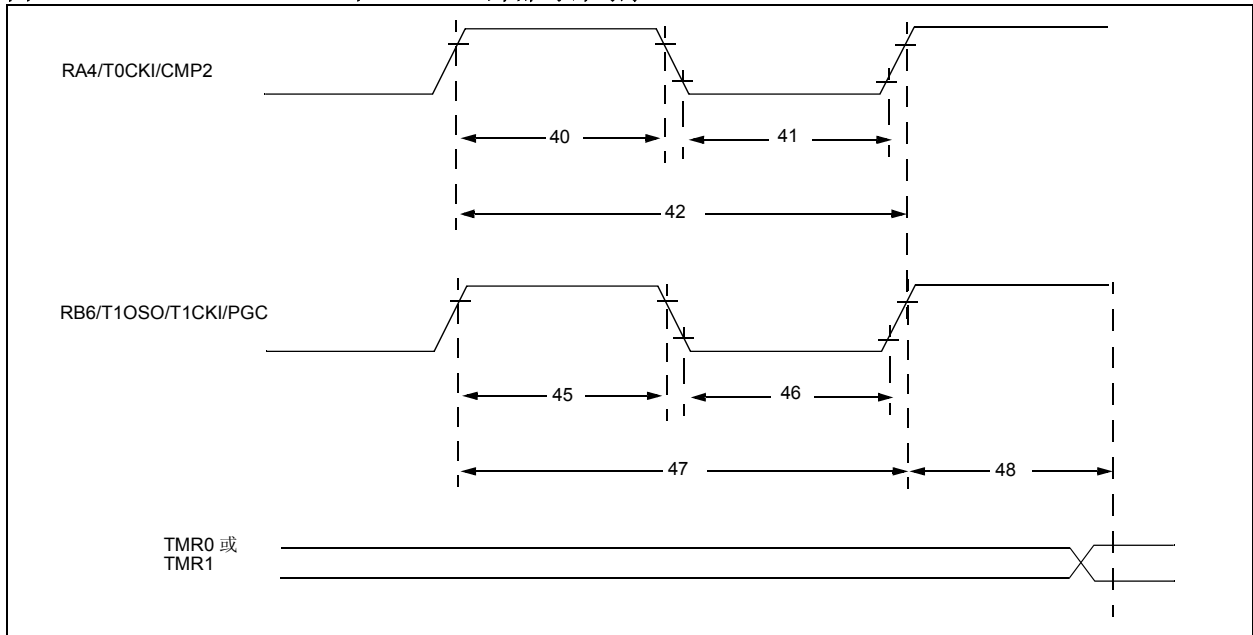
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
30	TMCL	MCLR 脉冲宽度 (低电平)	2000	—	—	ns	VDD = 5V, -40°C 至 +85°C
31	TWDT	看门狗超时周期 (无预分频器)	7*	18	33*	ms	VDD = 5V, -40°C 至 +85°C
32	TOST	振荡器起振定时器周期	—	1024 TOSC	—	—	TOSC = OSC1 周期
33	TPWRT	上电延时定时器周期	28*	72	132*	ms	VDD = 5V, -40°C 至 +85°C
34	TIOZ	由于 MCLR 低电平和看门狗复位引起的 I/O 高阻抗	—	—	2.0*	μs	
35	TBOR	欠压复位脉冲宽度	100*	—	—	μs	VDD ≤ VBOR (D005)

图注: TBD = 待定。

\* 这些参数为特性数据, 未经测试。

† 除非另外说明, 否则“典型值”栏中的数据都是在 5.0V、25°C 的条件下测得的。这些参数仅供设计参考, 未经测试。

图 17-8: TIMER0 和 TIMER1 外部时钟时序



# PIC16F627A/628A/648A

表 17-8: TIMER0 和 TIMER1 外部时钟要求

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	$0.5T_{CY} + 20^*$	—	—	ns		
			有预分频器	$10^*$	—	—	ns		
41	Tt0L	T0CKI 低电平脉冲宽度	无预分频器	$0.5T_{CY} + 20^*$	—	—	ns		
			有预分频器	$10^*$	—	—	ns		
42	Tt0P	T0CKI 周期		取较大值: $20$ 或 $\frac{T_{CY} + 40^*}{N}$	—	—	ns	N = 预分频比 (2, 4, ..., 256)	
45	Tt1H	T1CKI 高电平时间	同步, 无预分频器	$0.5T_{CY} + 20^*$	—	—	ns		
			同步, 有预分频器	PIC16F62X	$15^*$	—	—	ns	
				PIC16LF62X	$25^*$	—	—	ns	
			异步	PIC16F62X	$30^*$	—	—	ns	
PIC16LF62X	$50^*$	—		—	ns				
46	Tt1L	T1CKI 低电平时间	同步, 无预分频器	$0.5T_{CY} + 20^*$	—	—	ns		
			同步, 有预分频器	PIC16F62X	$15^*$	—	—	ns	
				PIC16LF62X	$25^*$	—	—	ns	
			异步	PIC16F62X	$30^*$	—	—	ns	
PIC16LF62X	$50^*$	—		—	ns				
47	Tt1P	T1CKI 输入周期	同步	PIC16F62X	取较大值: $20$ 或 $\frac{T_{CY} + 40^*}{N}$	—	—	ns	N = 预分频比 (1, 2, 4, 8)
				PIC16LF62X	取较大值: $20$ 或 $\frac{T_{CY} + 40^*}{N}$	—	—	—	
			异步	PIC16F62X	$60^*$	—	—	ns	
				PIC16LF62X	$100^*$	—	—	ns	
	Ft1	Timer1 振荡器输入频率范围 (置位 T1OSCEN 位使能振荡器)		—	$32.7^{(1)}$	—	kHz		
48	TCKEZTMR1	从外部时钟边沿到定时器递增的延时		$2T_{osc}$	—	$7T_{osc}$	—		

\* 这些参数为特性值, 未经测试。

† 除非另外说明, 否则“典型值”栏中的数据都是在 5V、25°C 的条件下测得的。这些参数仅供设计参考, 未经测试。

注 1: 该振荡器仅与 32.768 kHz 的时钟晶振配合使用, 具有制造容差。较高的晶振频率可能与此晶振驱动不兼容。

图 17-9: 捕捉 / 比较 / PWM 时序

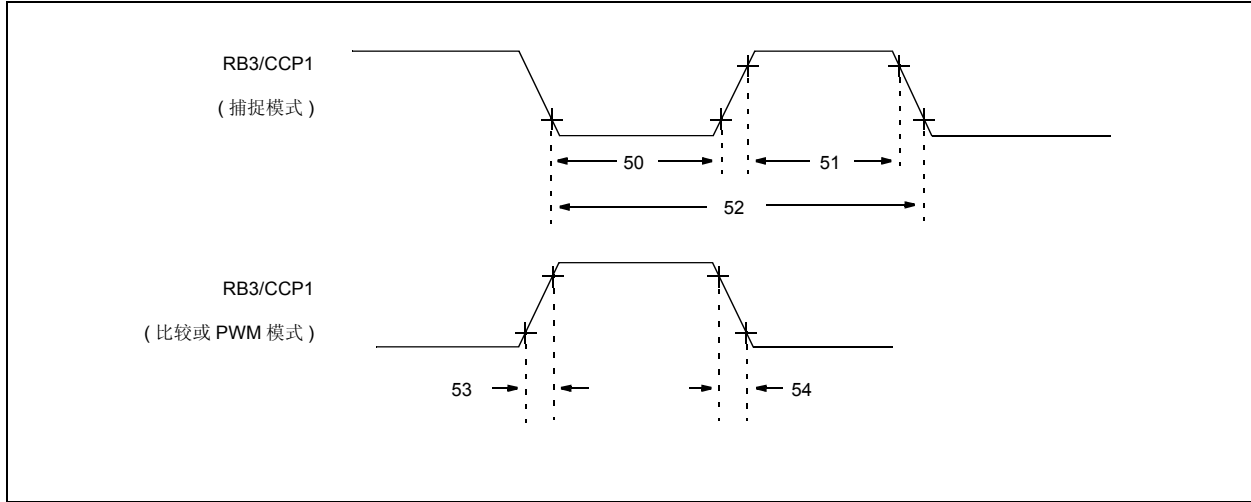


表 17-9: 捕捉 / 比较 / PWM 要求

参数编号	符号	特性		最小值	典型值 †	最大值	单位	条件
50	TccL	CCP 输入低电平时间	无预分频器	$0.5T_{CY} + 20^*$	—	—	ns	
			有预分频器	PIC16F62X	10*	—	—	
51	TccH	CCP 输入高电平时间	无预分频器	$0.5T_{CY} + 20^*$	—	—	ns	
			有预分频器	PIC16F62X	10*	—	—	
52	TccP	CCP 输入周期		$\frac{3T_{CY} + 40^*}{N}$	—	—	ns	N = 预分频比 (1、4 或 16)
53	TccR	CCP 输出上升时间	PIC16F62X		10	25*	ns	
			PIC16LF62X		25	45*	ns	
54	TccF	CCP 输出下降时间	PIC16F62X		10	25*	ns	
			PIC16LF62X		25	45*	ns	

\* 这些参数为特性值，未经测试。

† 除非另外说明，否则“典型值”栏中的数据都是在 5V、25°C 的条件下测得的。这些参数仅供设计参考，未经测试。

# PIC16F627A/628A/648A

---

---

注:



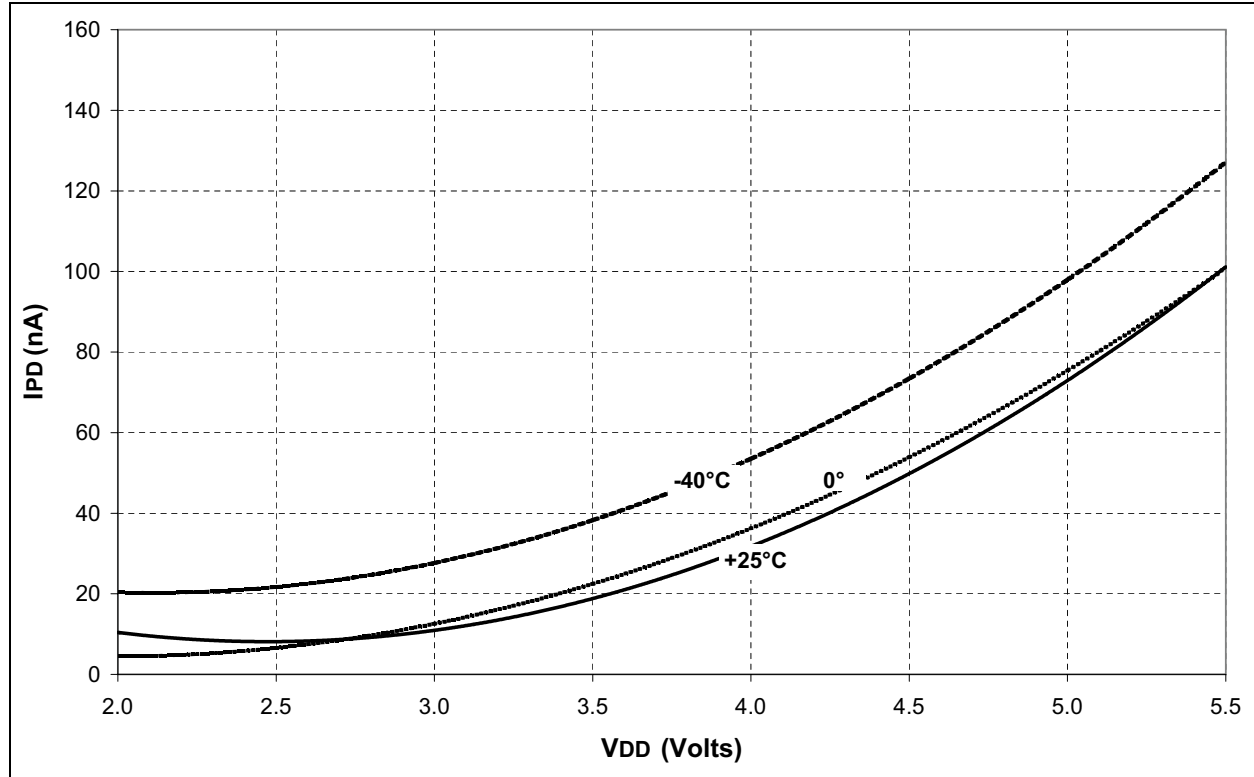
## 18.0 直流和交流特性图表

本章提供的图表仅供设计参考，未经测试。

某些图表中的数据超出了规定工作范围（如超出了规定的 VDD 范围），这仅用于提供信息，器件仅在规定的范围内才能确保正常工作。

本章所提供的数据是从不同批次的器件样片，经过一段时间收集的。“典型值”表示 25°C 时的分布平均值。“最大值”或“最小值”分别代表（平均值 + 3 $\sigma$ ）或（平均值 - 3 $\sigma$ ），其中  $\sigma$  为整个温度范围内的标准偏差。

图 18-1: 基本 IPD 典型值 — VDD 关系曲线（-40°C 至 25°C）



# PIC16F627A/628A/648A

图 18-2: 基本 IPD 典型值 — VDD 关系曲线 (85°C)

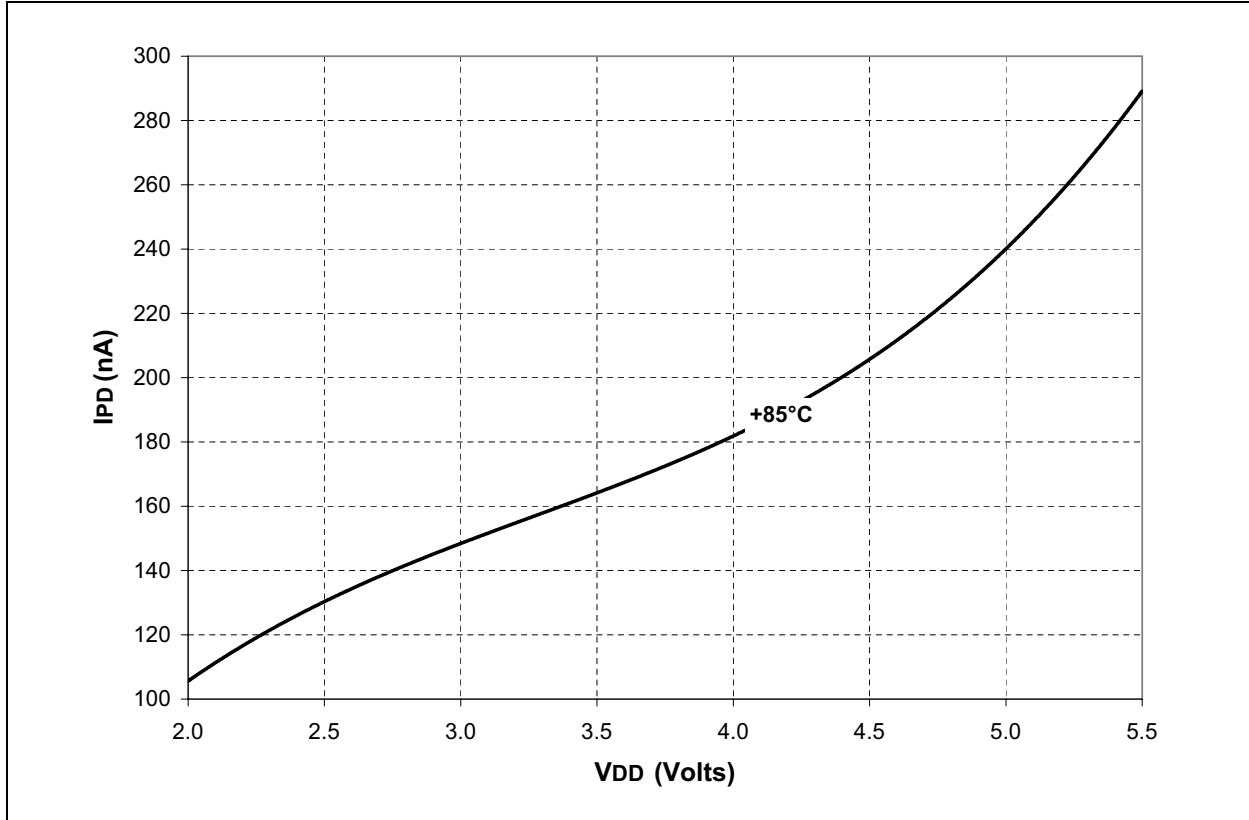


图 18-3: 基本 IPD 电流典型值 — VDD 关系曲线 (125°C)

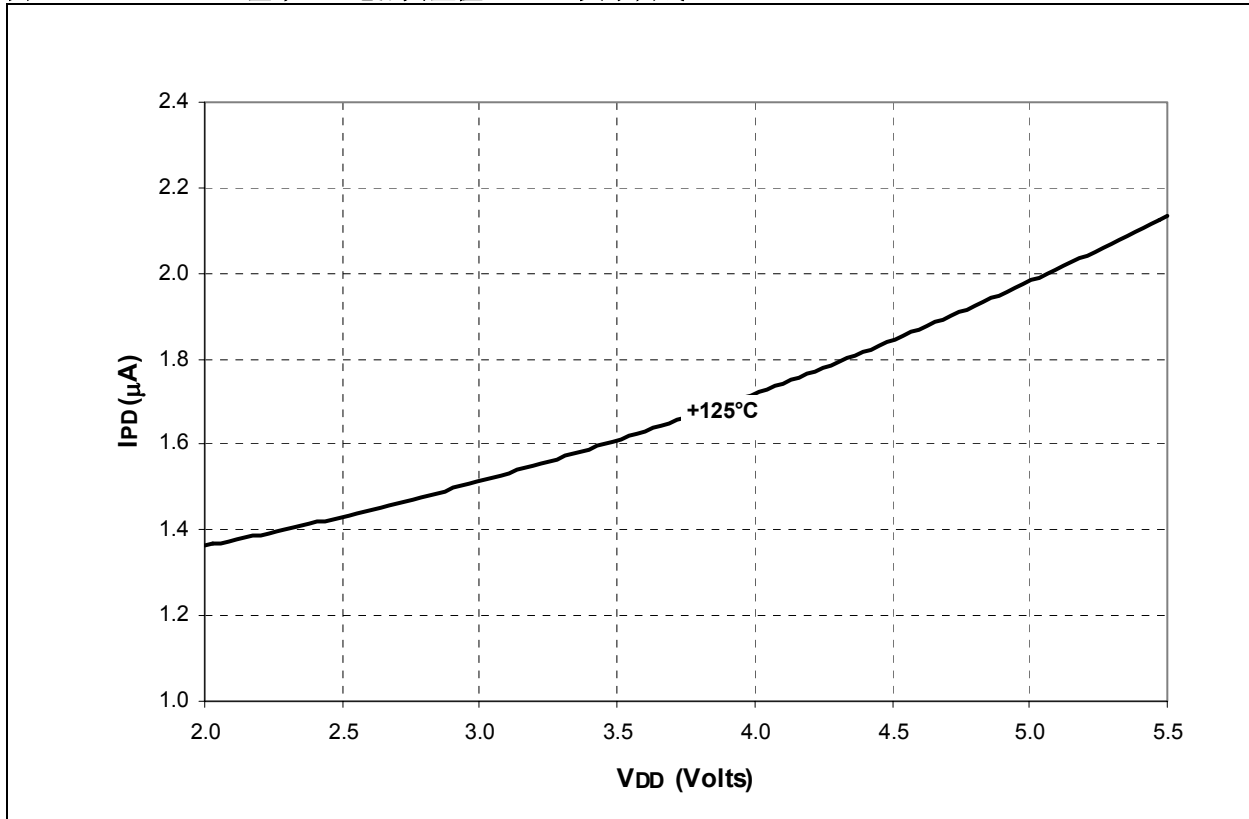


图 18-4: BOR IPD 典型值 — VDD 关系曲线

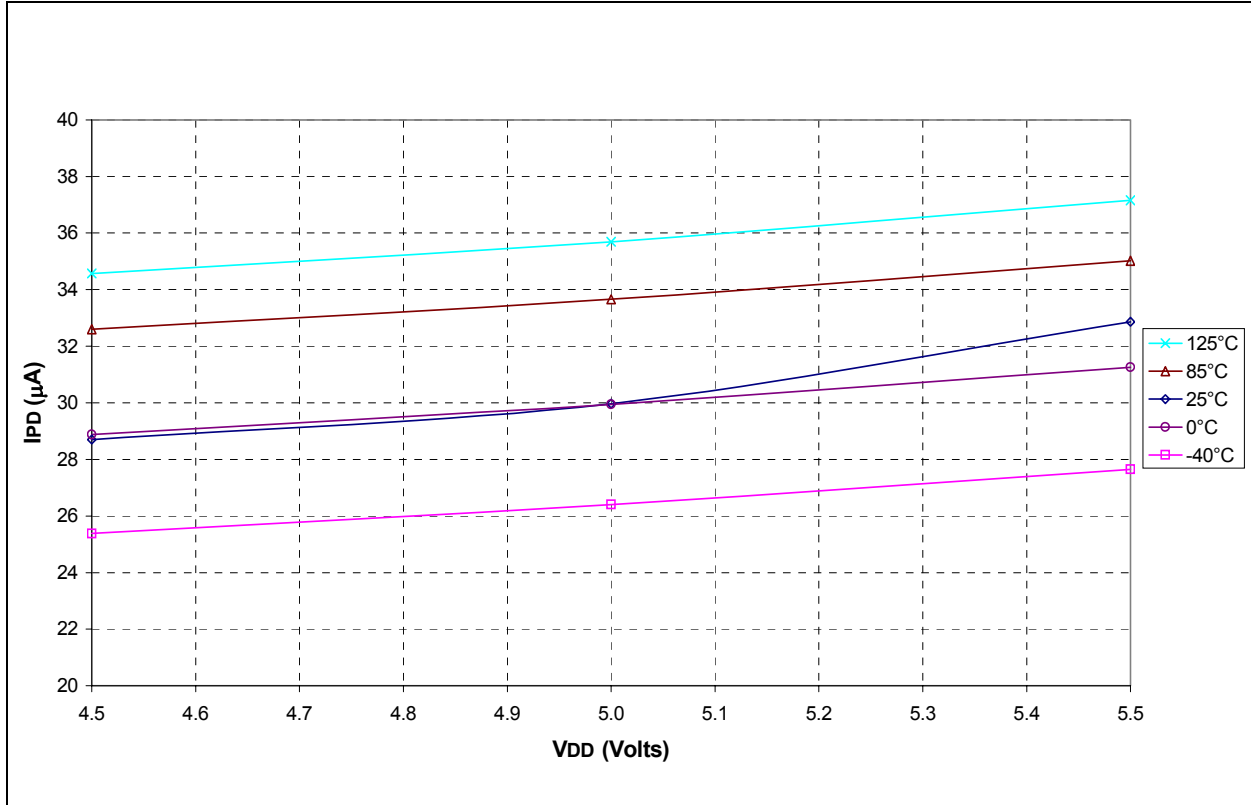
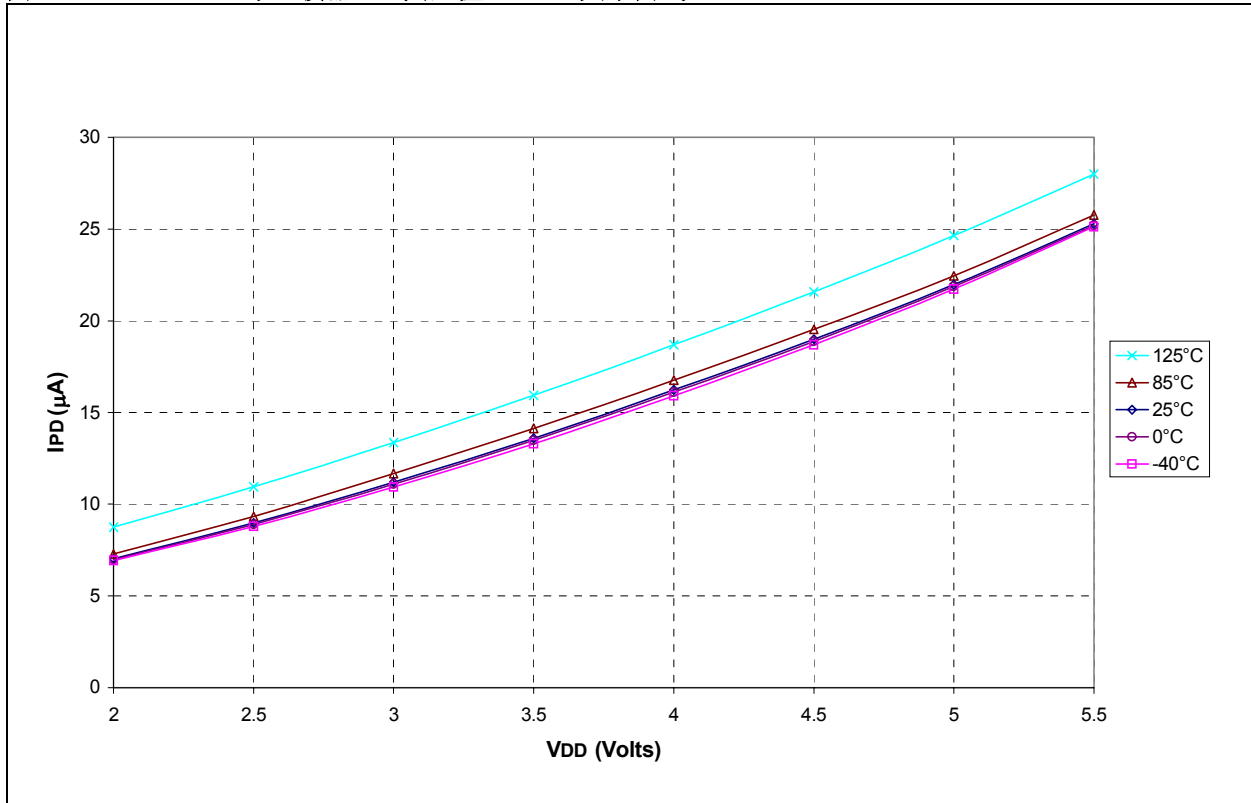


图 18-5: 单比较器 IPD 典型值 — VDD 关系曲线



# PIC16F627A/628A/648A

图 18-6: VREF IPD 典型值 — VDD 关系曲线

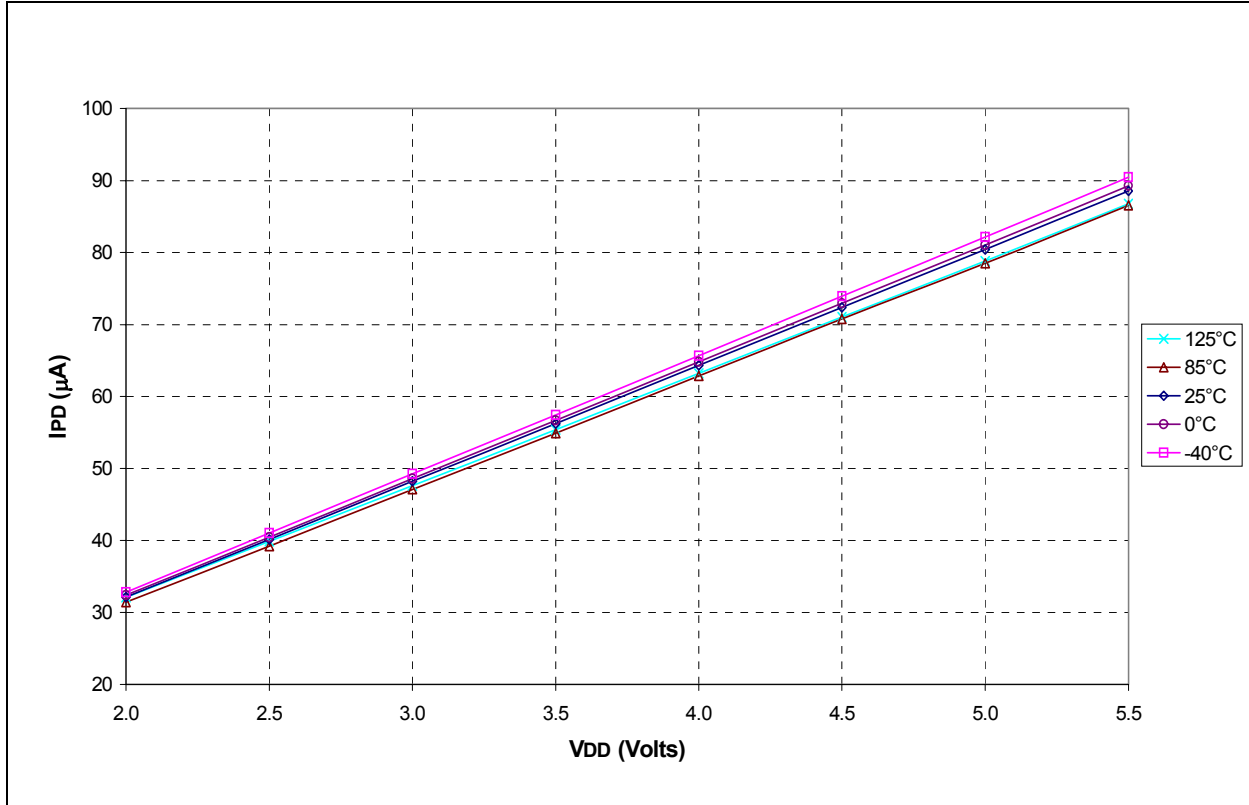


图 18-7: WDT IPD 典型值 — VDD 关系曲线

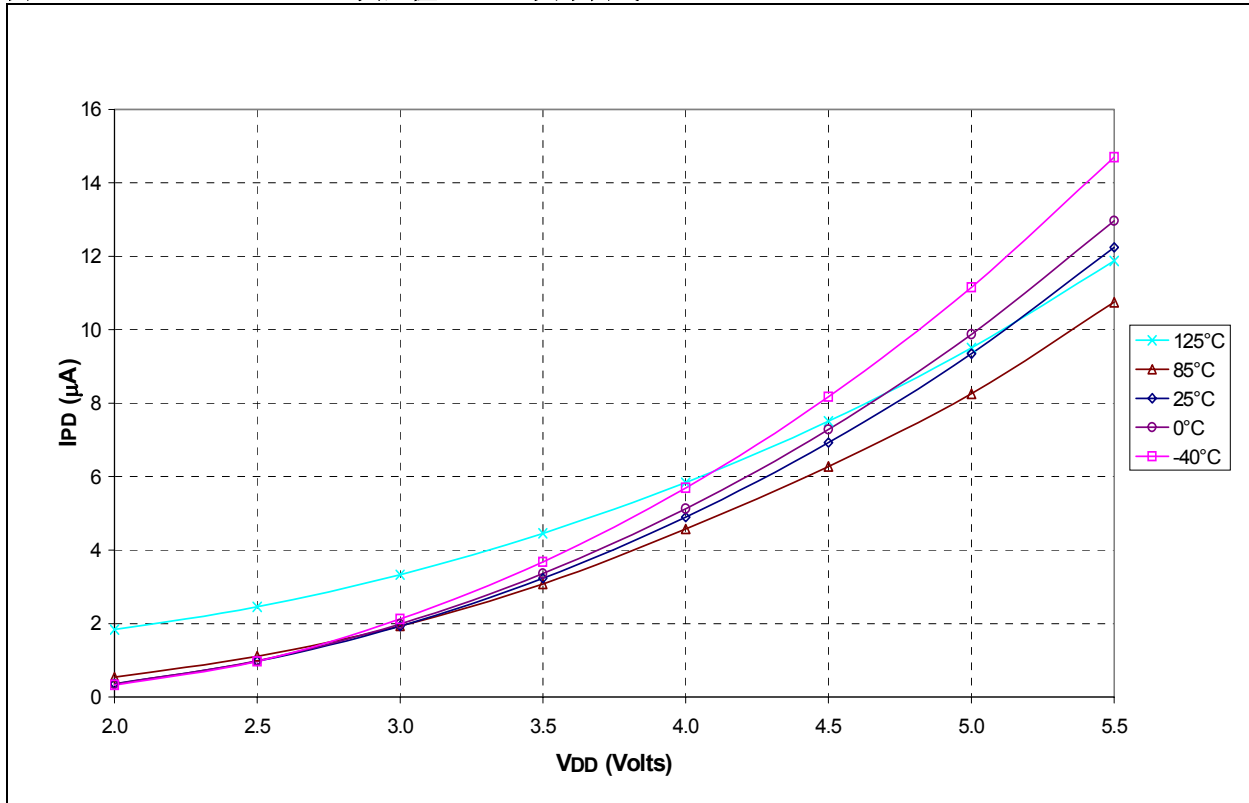


图 18-8: TIMER1 IPD 平均值

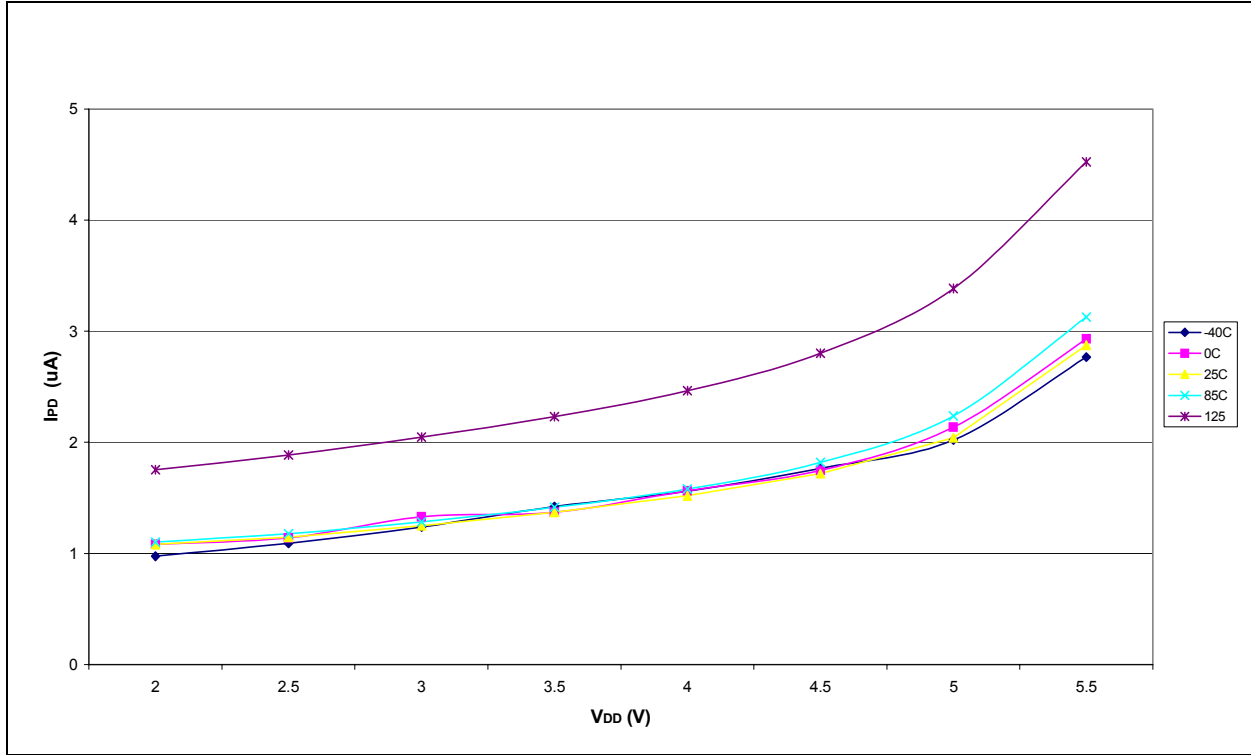
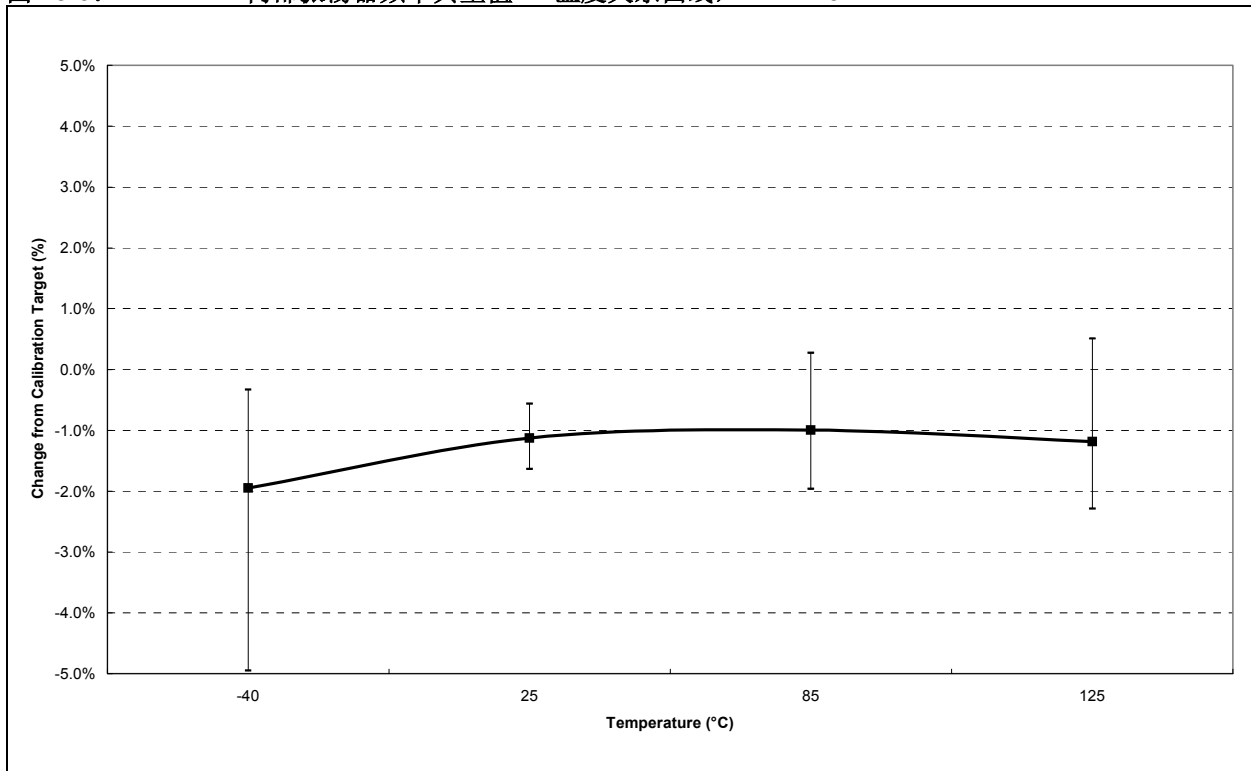


图 18-9: 内部振荡器频率典型值 — 温度关系曲线, VDD = 5 V



# PIC16F627A/628A/648A

图 18-10: 内部振荡器频率典型值 — 温度关系曲线,  $V_{DD} = 3\text{ V}$

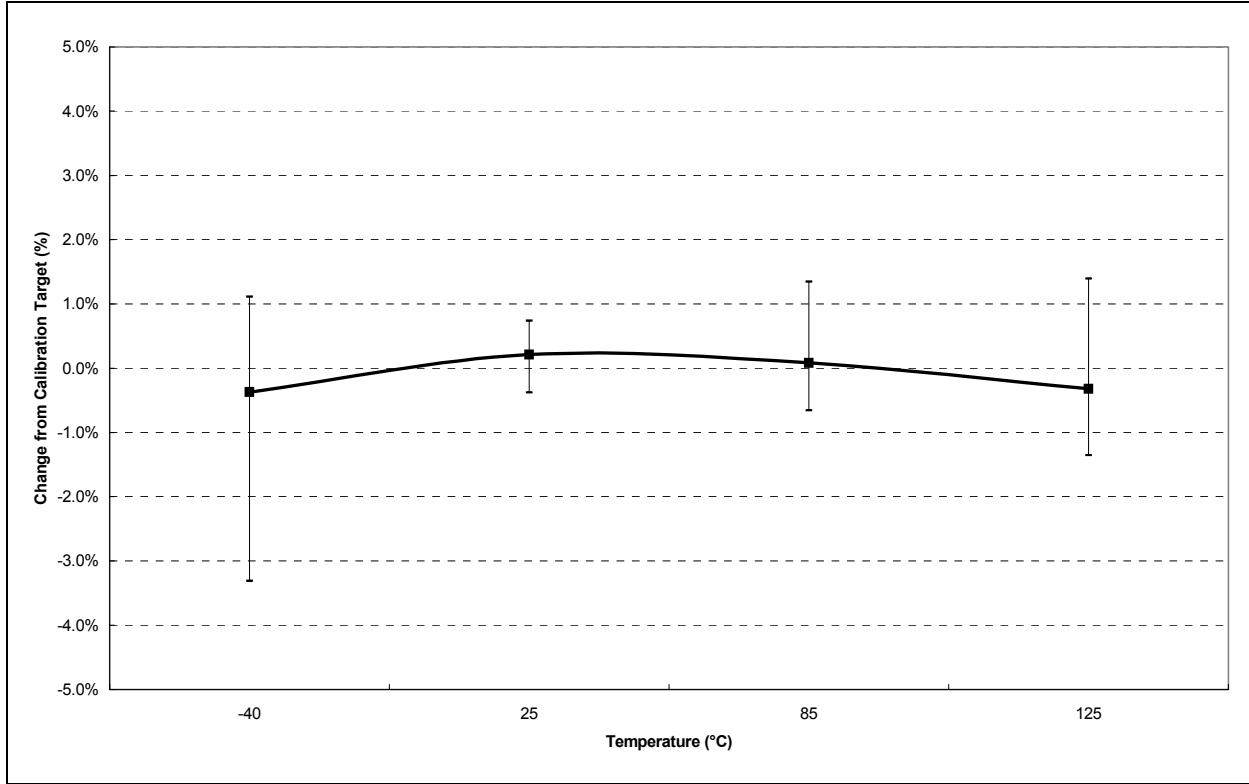


图 18-11: 内部振荡器频率典型值 — 温度关系曲线,  $V_{DD} = 2\text{ V}$

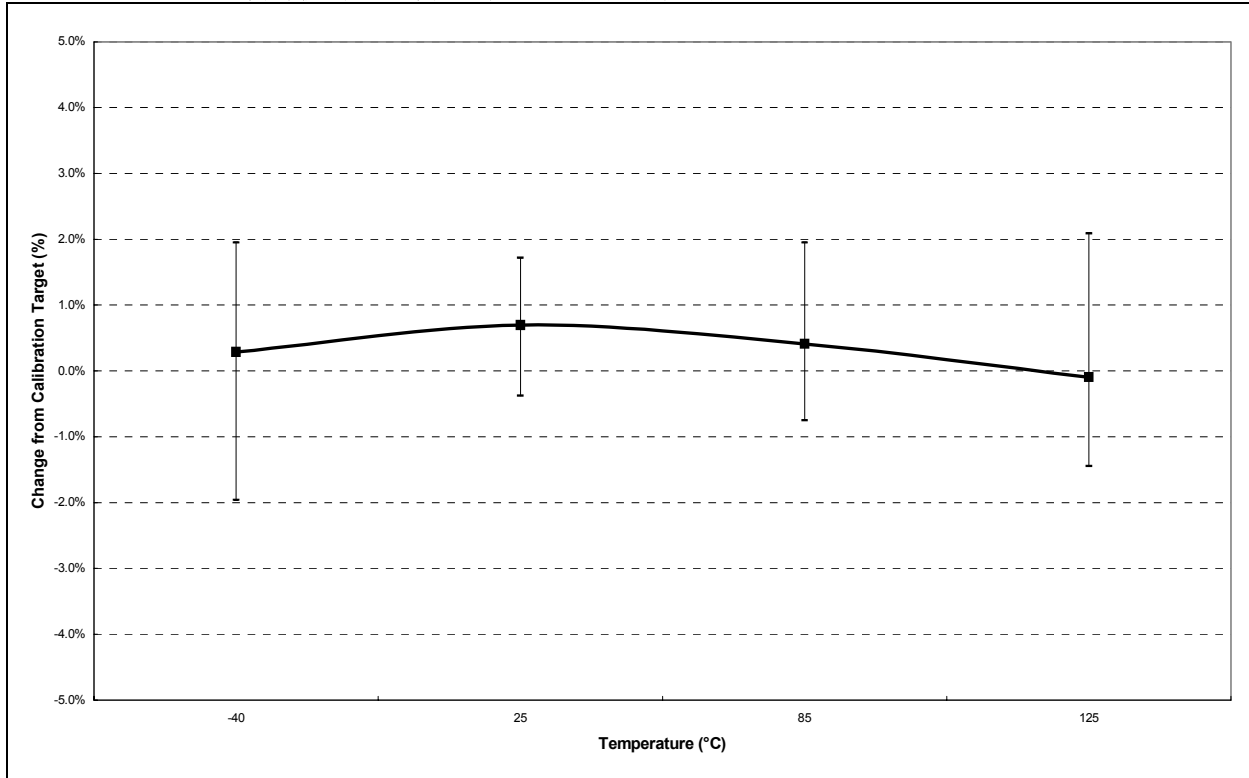


图 18-12: 内部振荡器偏差典型值 — VDD 关系曲线, 25°C – 4 MHz 模式

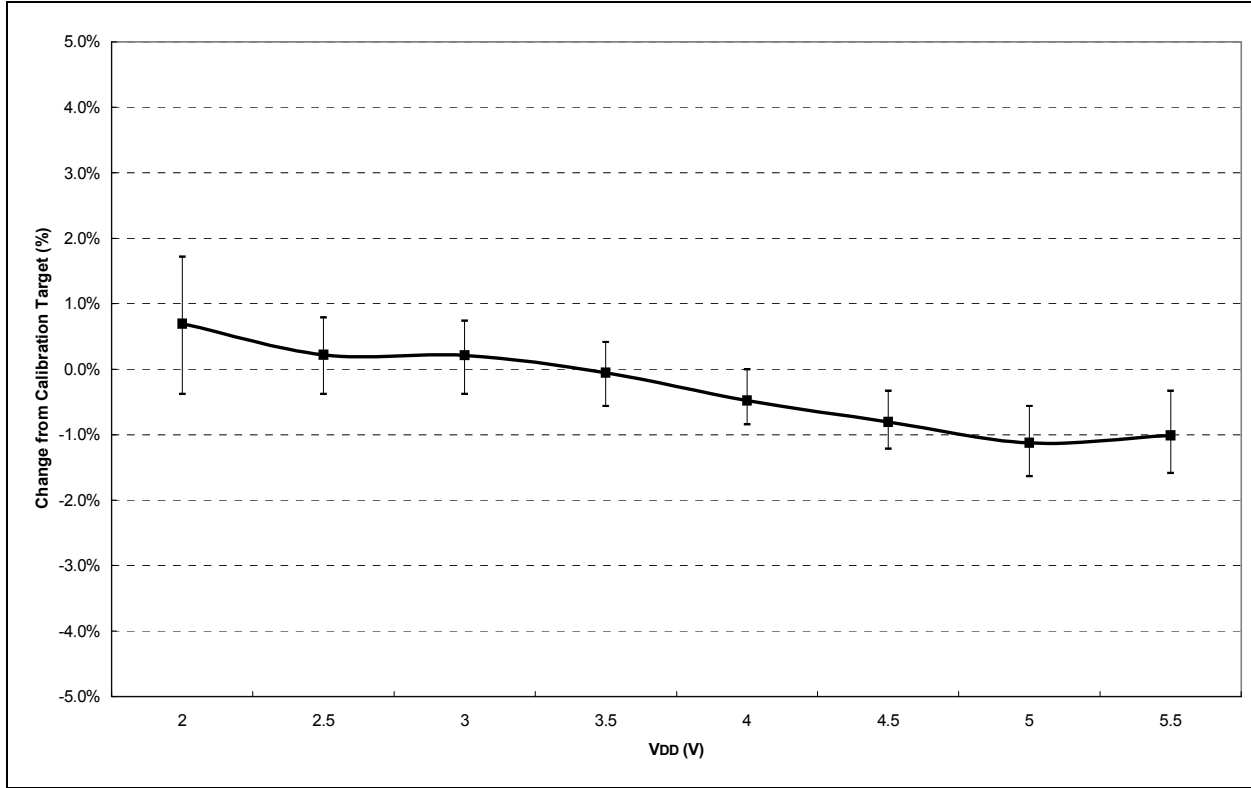
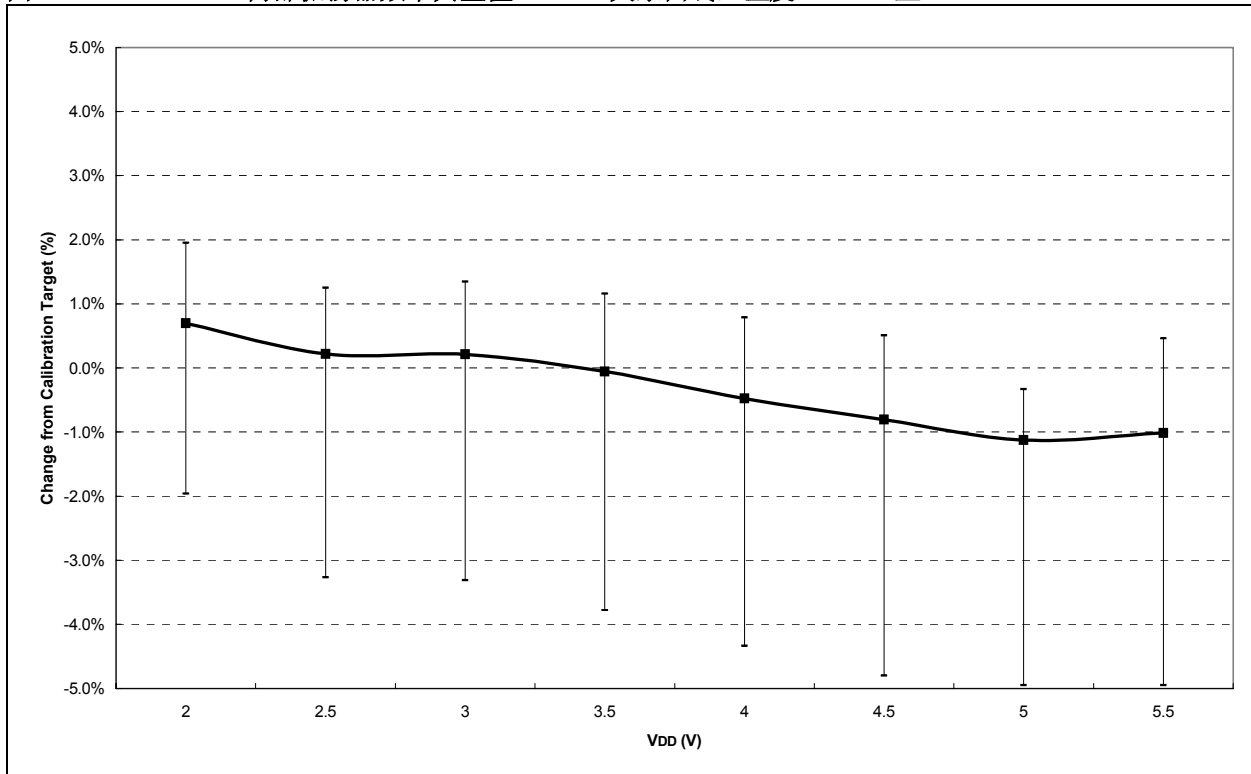


图 18-13: 内部振荡器频率典型值 — VDD 关系曲线, 温度 = -40°C 至 85°C



# PIC16F627A/628A/648A

图 18-14: 内部振荡器  $I_{DD}$  —  $V_{DD}$  关系曲线, 4 MHz 模式

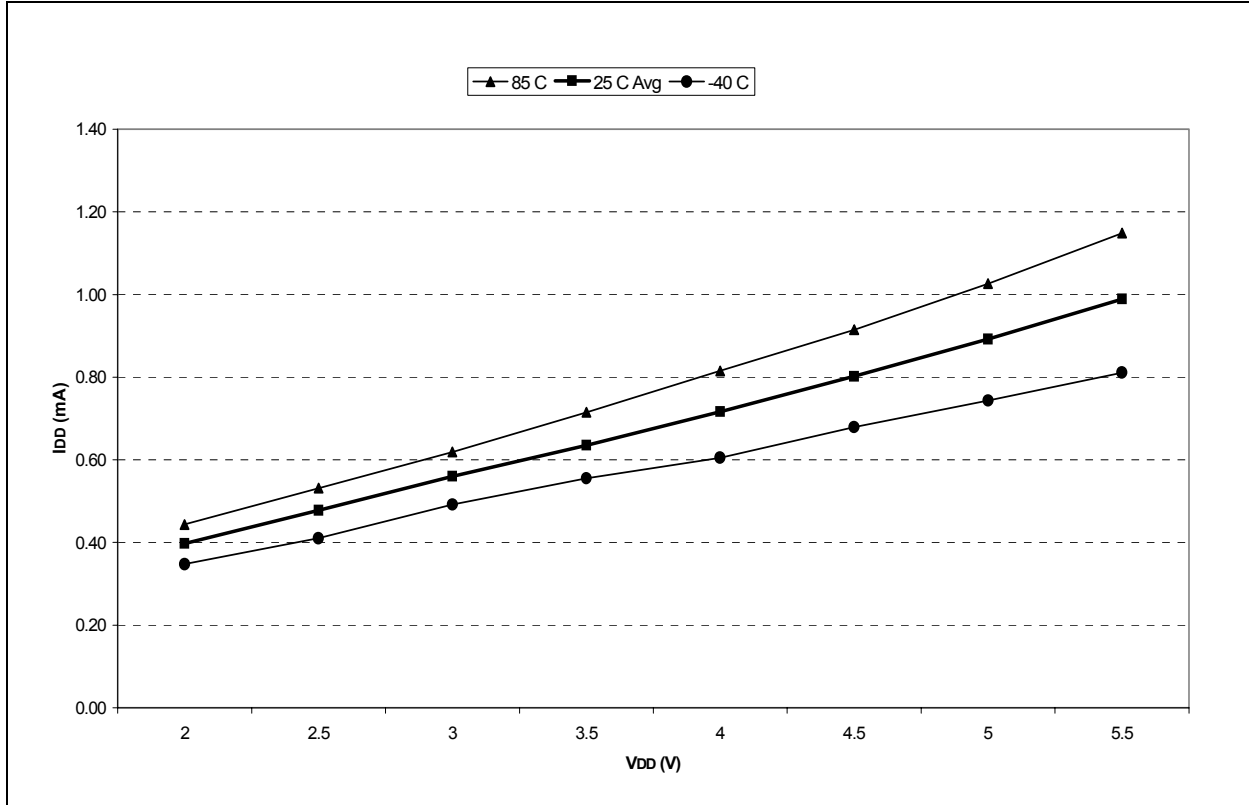


图 18-15: 内部振荡器频率典型值 —  $V_{DD}$  关系曲线, 25°C — 低速模式

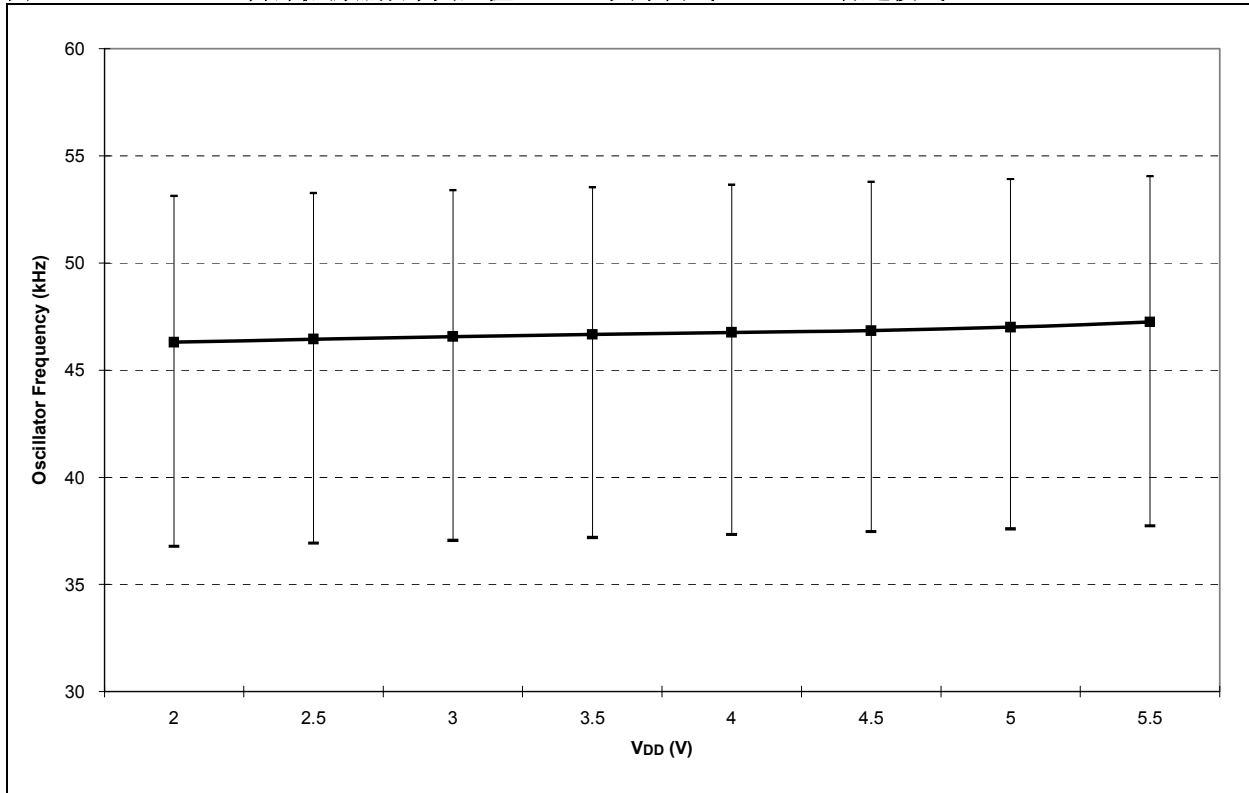




图 18-16: 内部振荡器  $I_{DD}$  —  $V_{DD}$  关系曲线, 低速模式

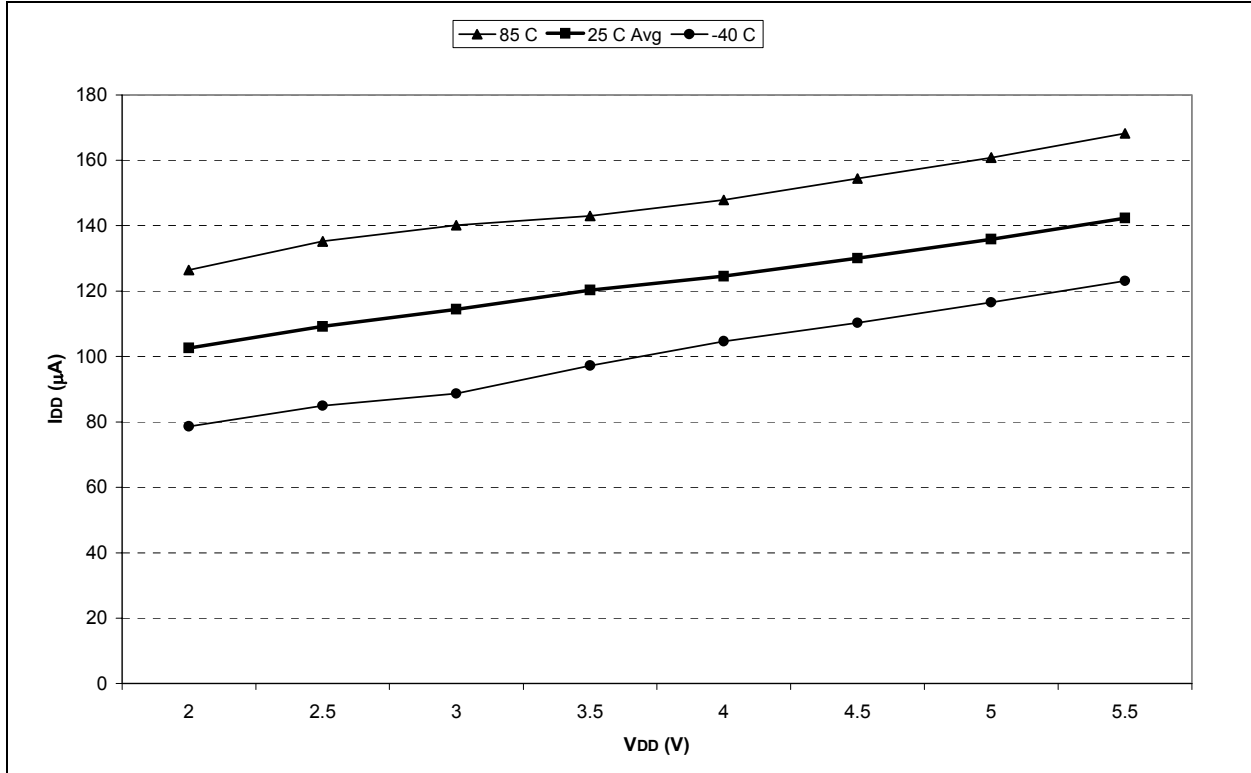
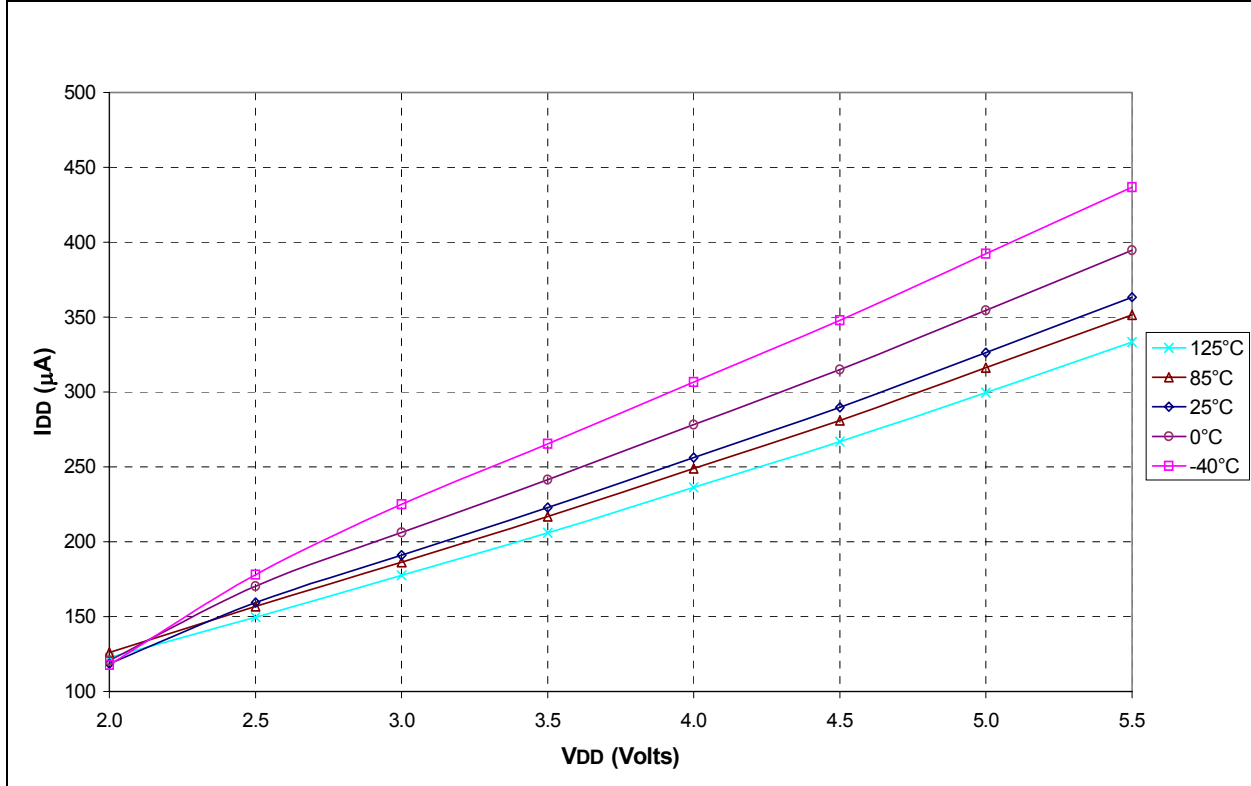


图 18-17: 供电电流  $I_{DD}$  —  $V_{DD}$  关系曲线,  $F_{osc} = 1$  MHz (XT 振荡器模式)



# PIC16F627A/628A/648A

图 18-18: 供电电流  $I_{DD}$  —  $V_{DD}$  关系曲线,  $F_{osc} = 4 \text{ MHz}$  (XT 振荡器模式)

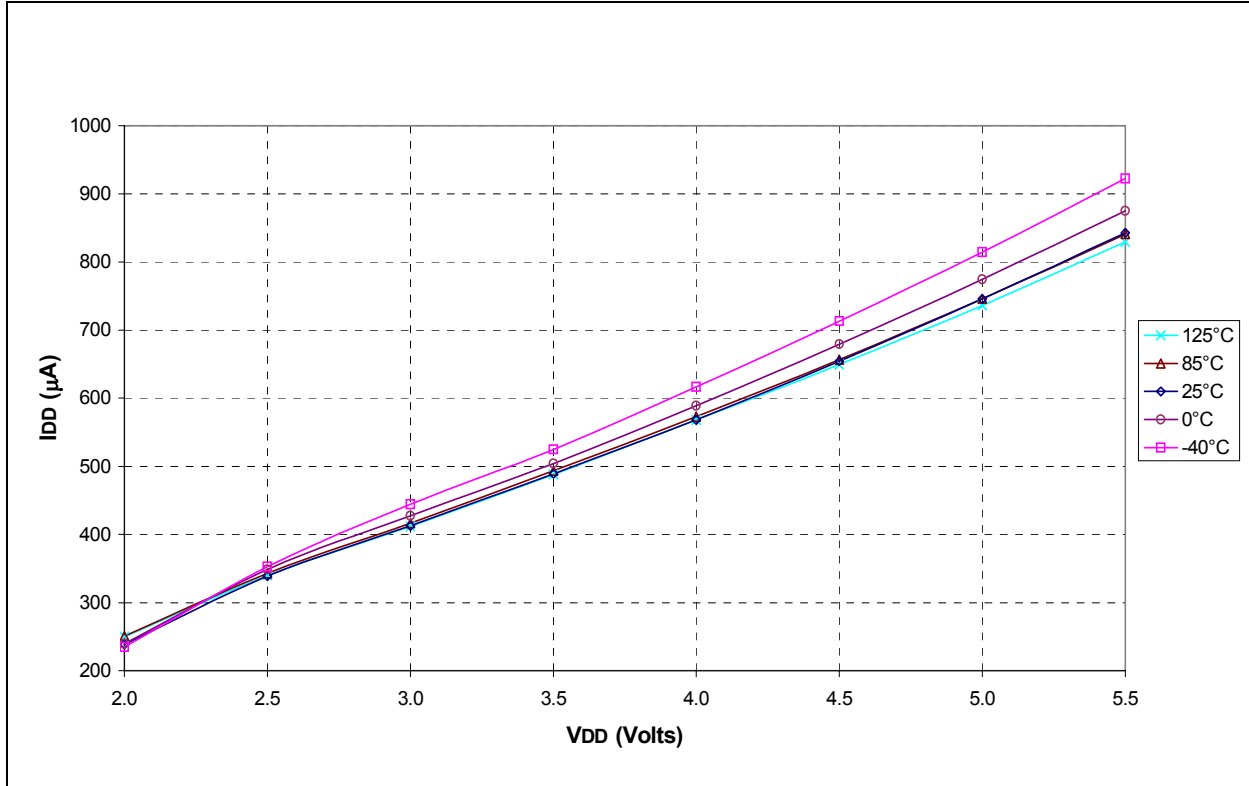


图 18-19: 供电电流  $I_{DD}$  —  $V_{DD}$  关系曲线,  $F_{osc} = 20 \text{ MHz}$  (HS 振荡器模式)

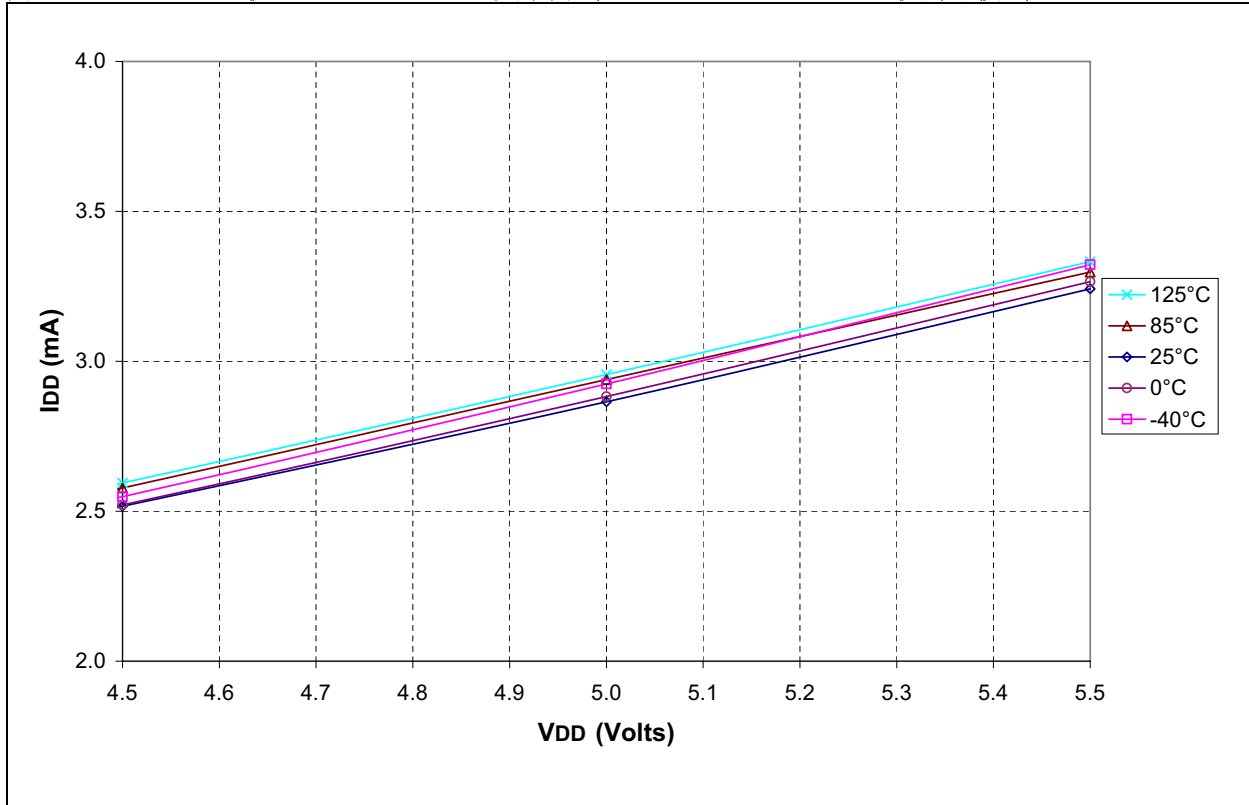
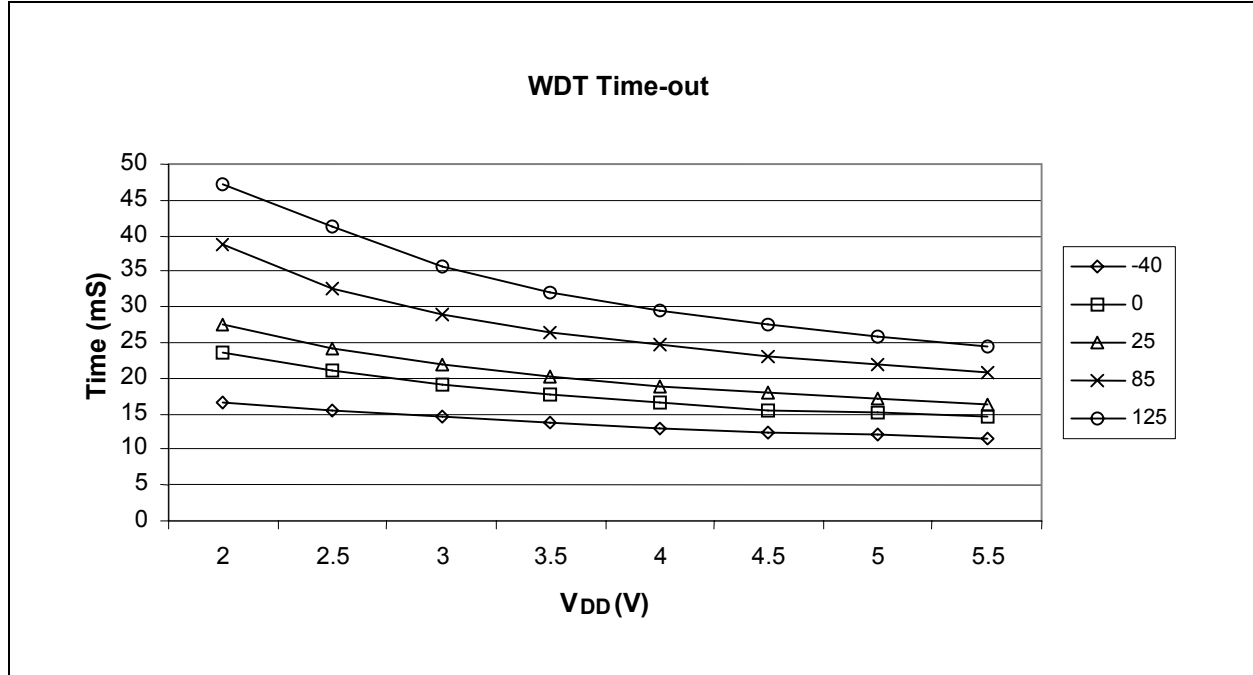


图 18-20: WDT 周期典型值 — V<sub>DD</sub> 关系曲线, (-40°C 至 +125°C)



# PIC16F627A/628A/648A

---

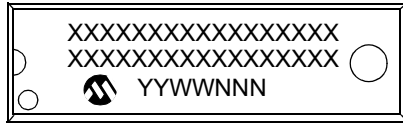
---

注:

## 19.0 封装信息

### 19.1 封装标识信息

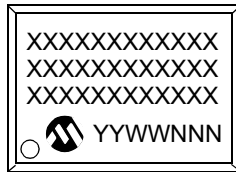
18 引脚 PDIP



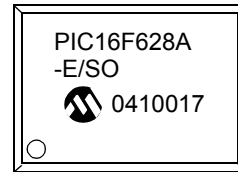
示例



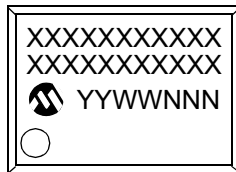
18 引脚 SOIC (.300")



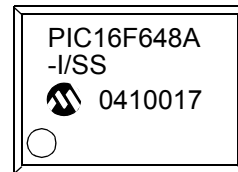
示例



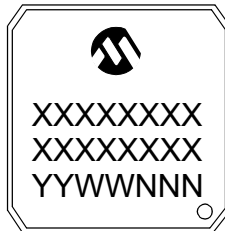
20 引脚 SSOP



示例



28 引脚 QFN



示例



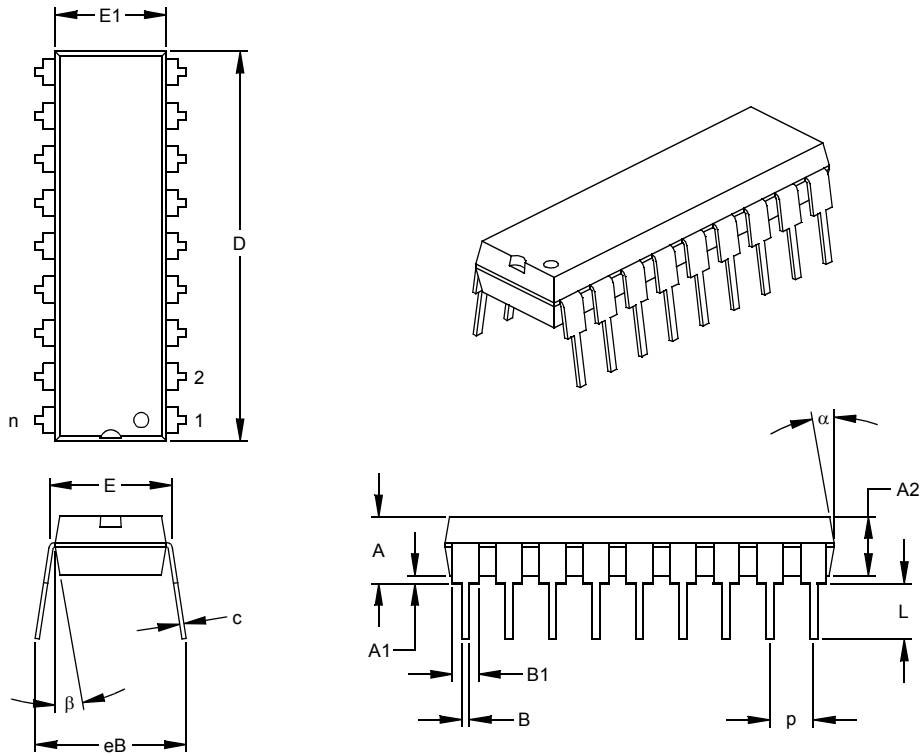
**图注:** XX...X 客户信息 \*  
 Y 年份代码 (日历年的最后一位数字)  
 YY 年份代码 (日历年的最后两位数字)  
 WW 星期代码 (一月一日的星期代码为“01”)  
 NNN 以字母数字排序的追踪代码

**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 PICmicro 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 PICmicro 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。

# PIC16F627A/628A/648A

## 18 引脚塑料双列直插封装 (P) —— 主体 300 mil (PDIP)



单位		英寸*			毫米		
尺寸限制		最小值	标准值	最大值	最小值	标准值	最大值
引脚数	n		18			18	
引线间距	p		.100			2.54	
顶端至底板面高度	A	.140	.155	.170	3.56	3.94	4.32
塑模封装厚度	A2	.115	.130	.145	2.92	3.30	3.68
塑封底端至底板面高度	A1	.015			0.38		
塑封最大宽度	E	.300	.313	.325	7.62	7.94	8.26
塑模封装总宽	E1	.240	.250	.260	6.10	6.35	6.60
塑模封装总长	D	.890	.898	.905	22.61	22.80	22.99
底脚长度	L	.125	.130	.135	3.18	3.30	3.43
引线厚度	c	.008	.012	.015	0.20	0.29	0.38
引线上部宽度	B1	.045	.058	.070	1.14	1.46	1.78
引线下部宽度	B	.014	.018	.022	0.36	0.46	0.56
封装引线跨度	§ eB	.310	.370	.430	7.87	9.40	10.92
塑模顶端倾斜角	α	5	10	15	5	10	15
塑模底端倾斜角	β	5	10	15	5	10	15

\* 控制参数

§ 重要特性

注

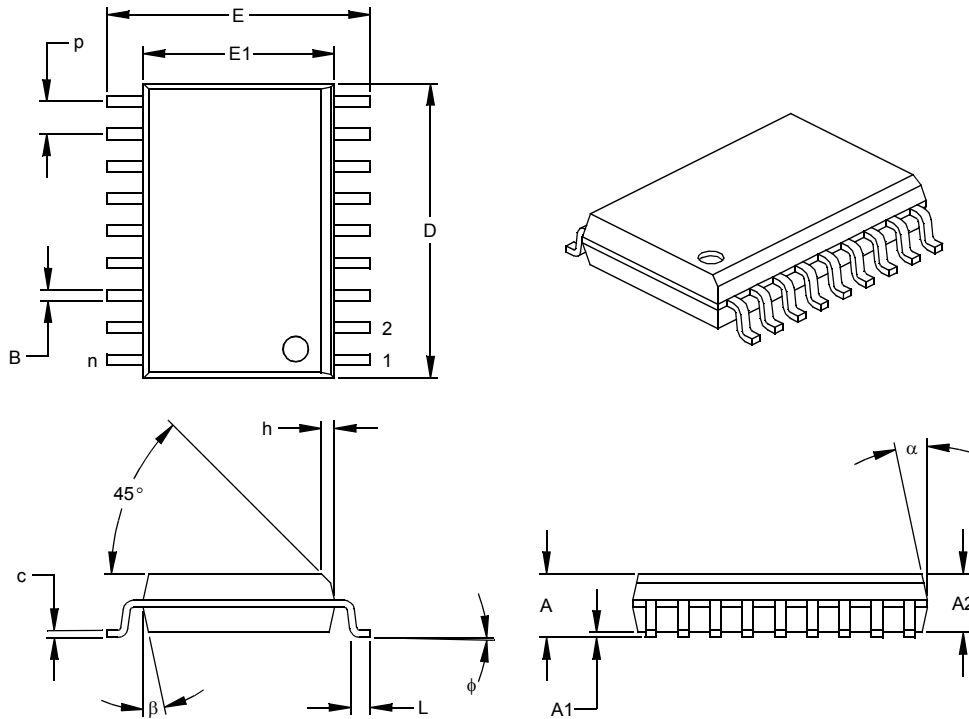
尺寸 D 和 E1 不包括塑模的毛边或突起。每侧毛边或突起不得超过 0.010 英寸 (0.254 毫米)。

等效 JEDEC 规范 MS-001

图号 C04-007

# PIC16F627A/628A/648A

18 引脚塑料小外形封装 (Small Outline, SO) —— 宽体, 主体 300 mil (SOIC)



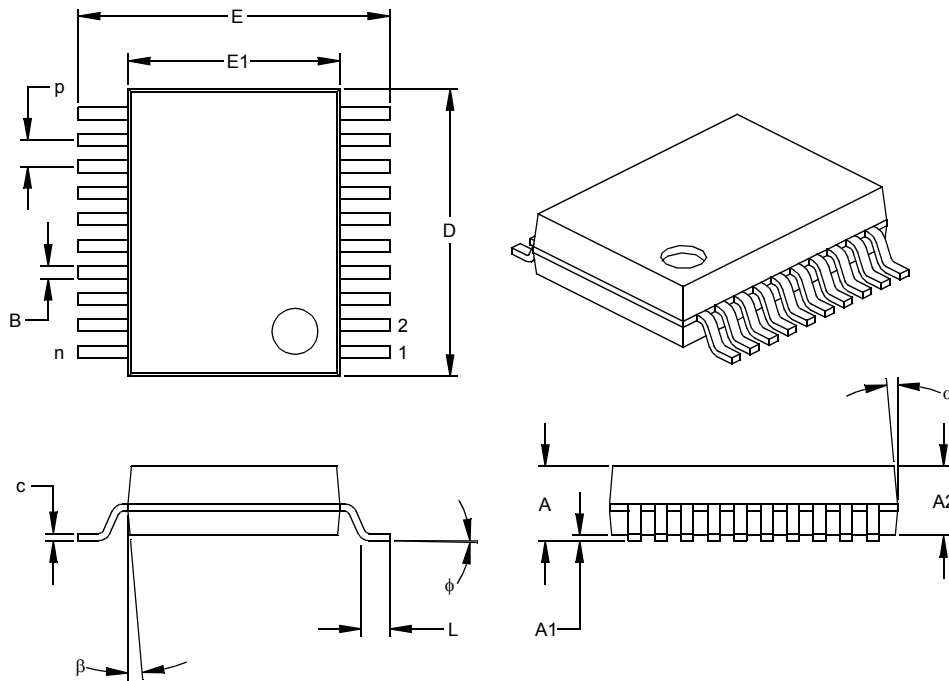
单位		英寸*			毫米		
尺寸限制		最小值	标准值	最大值	最小值	标准值	最大值
引脚数	n		18			18	
引线间距	p		.050			1.27	
总高度	A	.093	.099	.104	2.36	2.50	2.64
塑模封装厚度	A2	.088	.091	.094	2.24	2.31	2.39
托起高度	A1	.004	.008	.012	0.10	0.20	0.30
总宽度	E	.394	.407	.420	10.01	10.34	10.67
塑模封装宽度	E1	.291	.295	.299	7.39	7.49	7.59
总长度	D	.446	.454	.462	11.33	11.53	11.73
斜面距离	h	.010	.020	.029	0.25	0.50	0.74
底脚长度	L	.016	.033	.050	0.41	0.84	1.27
底脚倾斜角	φ	0	4	8	0	4	8
引线厚度	c	.009	.011	.012	0.23	0.27	0.30
引脚宽度	B	.014	.017	.020	0.36	0.42	0.51
塑模顶端倾斜角	α	0	12	15	0	12	15
塑模底端倾斜角	β	0	12	15	0	12	15

\* 控制参数  
§ 重要特性

注  
尺寸 D 和 E1 不包括塑模毛边或突起。模模每边的毛边或突起不得超过 .010 英寸 (0.254mm)。  
等效 JEDEC 规范 MS-013  
图号 C04-051

# PIC16F627A/628A/648A

20 引脚塑料缩小型小外形封装 (Shrink Small, SS) —— 主体 209 mil, 5.30 mm (SSOP)



单位		英寸*			毫米		
尺寸限制		最小值	标准值	最大值	最小值	标准值	最大值
引脚数	n		20			20	
引线间距	p		.026			0.65	
总高度	A	.068	.073	.078	1.73	1.85	1.98
塑模封装厚度	A2	.064	.068	.072	1.63	1.73	1.83
托起高度 §	A1	.002	.006	.010	0.05	0.15	0.25
总宽度	E	.299	.309	.322	7.59	7.85	8.18
塑模封装宽度	E1	.201	.207	.212	5.11	5.25	5.38
总长度	D	.278	.284	.289	7.06	7.20	7.34
底脚长度	L	.022	.030	.037	0.56	0.75	0.94
引线厚度	c	.004	.007	.010	0.10	0.18	0.25
底脚倾斜角	φ	0	4	8	0.00	101.60	203.20
引线宽度	B	.010	.013	.015	0.25	0.32	0.38

\* 控制参数

§ 重要特性

注:

尺寸 D 和 E1 不包括塑模毛边或突起。塑模每边的毛边或突起不得超过 .010 英寸 (0.254mm)。

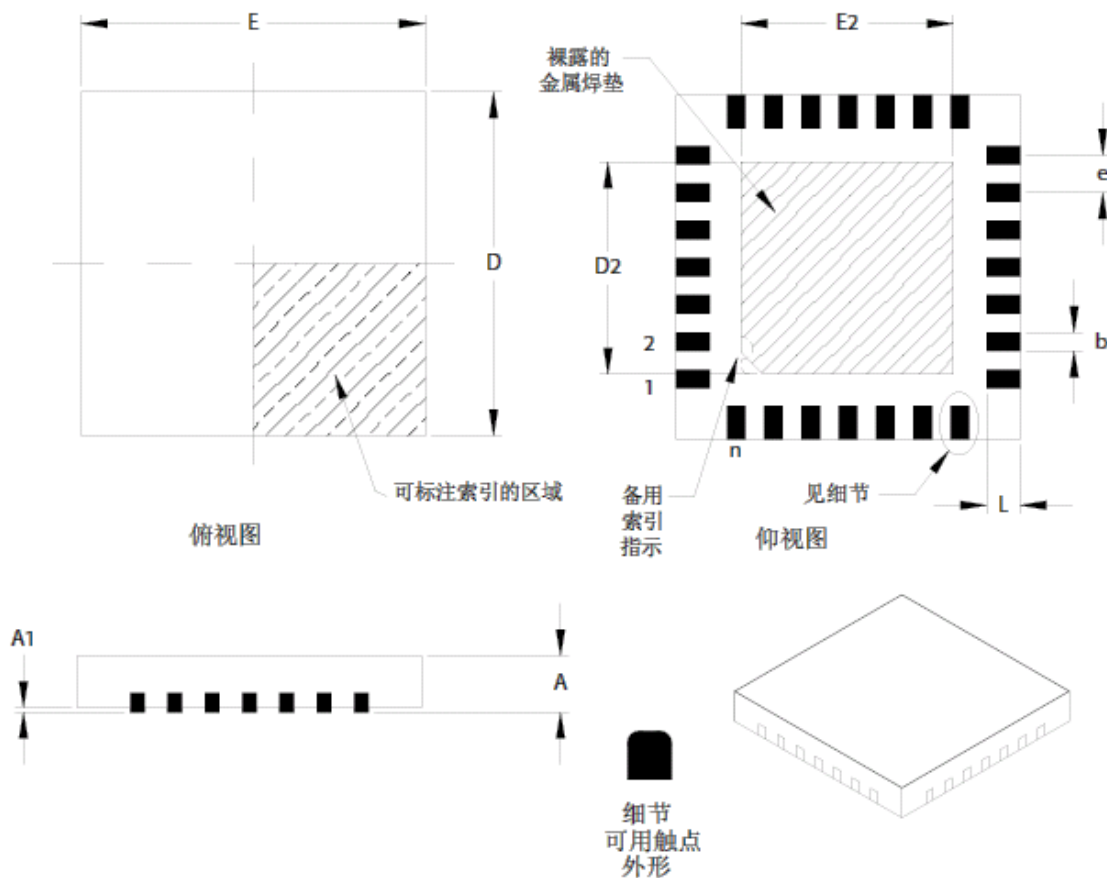
等效 JEDEC 规范 MO-150

图号: C04-072



# PIC16F627A/628A/648A

28 引脚塑料四方扁平无引线封装 (ML) 主体 6x6 mm (QFN) —— 触点长度 0.55 mm (Saw Singulated)



	单位 尺寸范围	英寸			毫米*		
		最小	正常	最大	最小	正常	最大
引脚数	n	28			28		
引脚间距	e	.026 BSC			0.65 BSC		
总高度	A	.031	.035	.039	0.80	0.90	1.00
基空间隙	A1	.000	.001	.002	0.00	0.02	0.05
触点厚度	A3	.008 REF			0.20 REF		
总宽度	E	.232	.236	.240	5.90	6.00	6.10
裸露的焊垫宽度	E2	.153	.167	.169	3.89	4.24	4.29
总长度	D	.232	.236	.240	5.90	6.00	6.10
裸露的焊垫长度	D2	.153	.167	.169	3.89	4.24	4.29
触点宽度	b	.009	.011	.013	0.23	0.28	0.33
触点长度	L	.018	.022	.024	0.45	0.55	0.65

\* 控制参数

注  
等效 JEDEC 规范: MO-220  
图号: C04-105

修订于 02-18-05

# PIC16F627A/628A/648A

---

---

注:

# PIC16F627A/628A/648A

## 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

修改了 28 引脚 QFN 的引脚图

修改了图 5-4 框图

修改了寄存器 7-1 TMR1ON

修改了例 13-4 数据 EEPROM 刷新程序

修改了指令集 SUBWF, 例 1

修改了 DC 特性 17-2 和 17-3

修改了表 17-4 和 17-6

校正了第 17.0 节中的图表编号

### 版本 C

整个文档从头至尾都有修改的版本。主要修改了第 14 章 — CPU 的特性, 并修改了第 18 章中的图。

### 版本 D

修改了例 13-2、数据 EEPROM 写操作

修改了第 17.2 节的参数 D020 以及第 17.3 节的参数 D020E

第 18 章增加了曲线图

## 附录 B: 器件差异

表 B-1 所示为此数据手册中所列 PIC16F627A/628A/648A 器件之间的差异。

表 B-1: 器件差异

器件	存储器		
	闪存程序	RAM 数据	EEPROM 数据
PIC16F627A	1024 x 14	224 x 8	128 x 8
PIC16F628A	2048 x 14	224 x 8	128 x 8
PIC16F648A	4096 x 14	256 x 8	256 x 8

# PIC16F627A/628A/648A

---

## 附录 C: 器件移植

本节描述在功能相似的器件之间进行移植时的功能和电气规范差异。(比如从 PIC16F627 移植到 PIC16F627A)。

### C.1 从 PIC16F627/628 移植到 PIC16F627A/628A

1. ER 模式现在是 RC 模式。
2. 程序存储器的代码保护从部分存储器的代码保护变为整个存储器的代码保护。PIC16F627/628 中的配置位 CP0 和 CP1 在 PIC16F627A/628A 中不存在。它们已经被配置位 <13>CP 所取代。
3. 术语“欠压检测 (Brown-out Detect, BOD)”改为“欠压复位 (Brown-out Reset, BOR)”可以更好地表示欠压电路的功能。
4. 使能欠压复位 (BOR) 不会像在 PIC16F627/628 中一样自动使能上电延时定时器。
5. INTRC 现在被称为 INTOSC。
6. 现在的 Timer1 振荡器的工作频率设计为 32.768 kHz。在 PIC16F627/628 中, Timer1 振荡器的工作频率设计为最大 200 kHz。
7. 双速振荡器模式只能在 INTOSC 振荡器模式下工作。而在 PIC16F627/628 中, 双速振荡器模式在 INTRC 和 ER 振荡器模式下都能工作。

## 附录 D: 从其他 PICmicro® 器件移植到此系列

本节将讨论从其他 PICmicro 单片机移植到 PIC16F627A/628A/648A 系列器件的一些问题。

### D.1 PIC16C62X/CE62X 到 PIC16F627A/628A/648A 的移植

参见 Microchip 网站 ([www.microchip.com](http://www.microchip.com))。

### D.2 PIC16C622A 到 PIC16F627A/628A/648A 的移植

参见 Microchip 网站 ([www.microchip.com](http://www.microchip.com))。

<p><b>注:</b> 已将此器件设计为按照数据手册上的参数来运行。它已经通过电气规范测试, 电气规范用于确定器件与这些参数的一致性。由于器件制造时的工艺差异, 此器件和它的早期版本之间可能会存在一些运行特性方面的差异。这些差异可能会导致器件与其早期版本在应用中的性能差异。</p>
--

## MICROCHIP 网站

Microchip 网站 ([www.microchip.com](http://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、应用笔记和样本程序、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、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

## 变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请登录 Microchip 网站 [www.microchip.com](http://www.microchip.com)，点击“变更通知客户 (Customer Change Notification)”服务后按照注册说明完成注册。

## 客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持
- 开发系统信息热线

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。

# PIC16F627A/628A/648A

---

---

## 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: PIC16F627A/628A/648A 文献编号: DS40044D\_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?

---

---

# PIC16F627A/628A/648A

## 索引

### A

#### A/D

特殊事件触发器 (CCP)	57
ADDLW 指令	117
ADDWF 指令	117
ANDLW 指令	117
ANDWF 指令	117

### B

BCF 指令	118
BRGH 位	73
BSF 指令	118
BTFSC 指令	118
BTFSS 指令	119
版本历史	169
比较器	
比较器模块	61
参考源	63
工作原理	63
框图	
改进的比较器输出	64
I/O 工作模式	62
配置	62
中断	65
比较 (CCP 模块)	56
CCPR1H:CCPR1L 寄存器	56
CCP 引脚配置	56
框图	56
软件中断	57
Timer1 模式选择	57
特殊事件触发器	57
波特率公式	73
波特率误差	73
捕捉 / 比较 / PWM (CCP)	55
比较模式。参见比较	
捕捉模式。参见捕捉	
CCP1	55
CCPR1H 寄存器	55
CCPR1L 寄存器	55
CCP2	55
定时器资源	55
PWM 模式。参见 PWM	
捕捉 (CCP 模式)	56
CCPR1H:CCPR1L 寄存器	56
CCP 引脚配置	56
改变捕捉预分频器	56
框图	56
软件中断	56
Timer1 模式选择	56
预分频器	56

### C

CALL 指令	119
C 编译器	
MPLAB C17	130
MPLAB C18	130
MPLAB C30	130
CCP1CON 寄存器	
CCP1X:CCP1Y 位	55
CCP2CON 寄存器	
CCP2X:CCP2Y 位	55
CLRF 指令	119

CLRWDT 指令	120
CLRW 指令	120
COMF 指令	120
CPU 的特殊功能	95
参考电压	
参考电压模块	67
程序存储器组织	15
串行通信接口模块, 参见 USART	
从其他 PICmicro 器件移植	170
存储器组织	
数据 EEPROM 存储器	89, 91, 93

### D

DECFSZ 指令	121
DECF 指令	120
代码保护	111
代码示例	
数据 EEPROM 刷新程序	92
参考电压	
配置	67
电源控制 / 状态寄存器 (PCON)	102
掉电 (休眠) 模式	110
端口 RB 中断	108

### E

EECON1 寄存器	90
EECON2 寄存器	90

### F

封装标识信息	163
封装信息	163
复位	99

### G

GOTO 指令	121
更改客户通知服务	171

### H

汇编器	
MPASM 汇编器	130

### I

ID 地址单元	111
INCF 指令	122
INCF 指令	122
INTCON 寄存器	24
INT 中断	108
I/O 端口	31
编程注意事项	44
框图	
RB0/INT 引脚	36
RB1/RX/DT 引脚	37
RB2/TX/CK 引脚	37
RB3/CCP1 引脚	38
RB4/PGM 引脚	39
RB5 引脚	40
RB6/T1OSO/T1CKI 引脚	41
RB7/T1OSI 引脚	42
连续操作	44
PORTA	31
PORTB	36
双向	44
TRISA	31
TRISB	36

# PIC16F627A/628A/648A

## J

### 寄存器

OPTION_REG (选择)	23
PCON (电源控制)	27
PIR1 (外设中断寄存器 1)	26
Status	22
映射	
PIC16F627A	16, 17
PIC16F628A	16, 17

架构综述	9
间接寻址、INDF 和 FSR 寄存器	28
晶振工作原理	97
绝对最大额定值	135

## K

开发支持	129
看门狗定时器 (WDT)	109
勘误表	3
客户支持	171
快速批量生产 (QTP) 器件	7
框图	

### 比较器

改进的比较器输出	64
I/O 工作模式	62

### I/O 端口

RB0/INT 引脚	36
RB1/RX/DT 引脚	37
RB2/TX/CK 引脚	37
RB3/CCP1 引脚	38
RB4/PGM 引脚	39
RB5 引脚	40
RB6/T1OSO/T1CKI 引脚	41
RB7/T1OSI 引脚	42

RC 振荡模式	99
USART 发送	78
USART 接收	80

## M

MOVF 指令	123
MOVLW 指令	123
MOVWF 指令	124
MPLAB ASM30 汇编器、链接器和库管理器	130
MPLAB ICD 2 在线调试器	131
MPLAB ICE 2000 高性能通用在线仿真器	131
MPLAB ICE 4000 高性能通用在线仿真器	131
MPLAB PM3 器件编程器	131
MPLAB 集成开发环境软件	129
MPLINK 目标链接器 /MPLIB 目标库管理器	130

## N

NOP 指令	124
--------	-----

## O

OPTION 寄存器	23
OPTION_REG 寄存器	23
OPTION 指令	124

## P

PCL 和 PCLATH	28
堆栈	28
PCON 寄存器	27
PICSTART Plus 开发编程器	132
PIE1 寄存器	25

PIR1 寄存器	26
PORTA	31
PORTB	36
PR2 寄存器	52, 58
PRO MATE II 通用器件编程器	131
PWM (CCP 模块)	
占空比	59
PWM (CCP 模块)	58
CCPR1H:CCPR1L 寄存器	58
框图	58
简化 PWM	58
频率 / 分辨率示例	59
设置执行 PWM 操作	59
TMR2 与 PR2 匹配	58
占空比	59
周期	58
配置位	95

## Q

Q 时钟	59
器件差异	169
器件移植	170
欠压检测 (Brown-Out Detect, BOD)	101

## R

RC 振荡模式	
框图	99
RC 振荡器	99
RETFIE 指令	124
RETLW 指令	125
RETURN 指令	125
RLF 指令	125
RRF 指令	126
软件模拟器 (MPLAB SIM)	130

## S

SLEEP 指令	126
STATUS 寄存器	22
Status 寄存器	22
SWAPF 指令	127
SUBLW 指令	126
SUBWF 指令	127
上电复位 (POR)	101
上电延迟定时器 (PWRT)	101

### 时序图

Timer0	147
Timer1	147

### USART

异步接收器	81
USART 同步接收	87
USART 同步传输	85
USART 异步接收	81
USART 异步主控发送	78

时序图 / 指令周期	13
------------	----

时序图和规范	144
--------	-----

数据存储组织	15
--------	----

数据 EEPROM 存储器	89
---------------	----

代码保护时的操作	93
----------	----

读取	91
----	----

EECON1 寄存器	89
------------	----

EECON2 寄存器	89
------------	----

使用	92
----	----

误写保护	91
------	----

写入	91
----	----

写校验	91
-----	----



# PIC16F627A/628A/648A

双速振荡模式 .....	99	框图	
<b>T</b>		发送 .....	78
T1CKPS0 位 .....	48	USART 接收 .....	80
T1CKPS1 位 .....	48	同步从动发送 .....	87
T1OSCEN 位 .....	48	同步从动接收 .....	88
T1SYNC 位 .....	48	同步从动模式 .....	87
T2CKPS0 位 .....	53	同步主控发送 .....	84
T2CKPS1 位 .....	53	同步主控接收模式 .....	86
Timer0		同步主控模式 .....	84
框图		异步发送 .....	78
Timer0/WDT .....	46	异步发送器 .....	77
切换预分频器的分配 .....	47	异步接收 .....	82
Timer0 模块 .....	45	异步接收器 .....	80
外部时钟输入 .....	45	异步模式 .....	77
预分频器 .....	46	WWW 在线技术支持 .....	3
中断 .....	45	外部晶振电路 .....	98
Timer2		<b>X</b>	
后分频器 .....	52	XORLW 指令 .....	128
框图 .....	52	XORWF 指令 .....	128
PR2 寄存器 .....	52	<b>Y</b>	
Timer2 模块 .....	52	引脚功能	
TMR2 输出 .....	52	RC6/TX/CK .....	71-87
TMR2 与 PR2 匹配中断 .....	58	RC7/RX/DT .....	71-87
预分频器 .....	52, 59	<b>Z</b>	
Timer1		在线串行编程 .....	112
电容选择 .....	51	振荡器配置 .....	97
定时器模式 .....	49	振荡器起振定时器 (OST) .....	101
复位 Timer1 .....	51	指令集	
复位 Timer1 寄存器 .....	51	ADDLW .....	117
TMR1H .....	50	ADDWF .....	117
TMR1L .....	50	ANDLW .....	117
特殊事件触发器 (CCP) .....	57	ANDWF .....	117
同步计数器模式 .....	49	BCF .....	118
外部时钟输入 .....	49	BSF .....	118
外部时钟输入时序 .....	50	BTFSC .....	118
异步计数器模式 .....	50	BTFSS .....	119
预分频器 .....	49, 51	CALL .....	119
振荡器 .....	51	CLRF .....	119
TMR0 中断 .....	108	CLRW .....	120
TMR1CS 位 .....	48	CLRWDT .....	120
TMR1ON 位 .....	48	COMF .....	120
TMR2ON 位 .....	53	DECF .....	120
TOUTPS0 位 .....	53	DECFSZ .....	121
TOUTPS1 位 .....	53	GOTO .....	121
TOUTPS2 位 .....	53	INCF .....	122
TOUTPS3 位 .....	53	INCFSZ .....	122
TRISA .....	31	IORLW .....	123
TRISB .....	36	IORWF .....	123
TRIS 指令 .....	127	MOVF .....	123
特殊功能寄存器 .....	18	MOVLW .....	123
特殊事件触发器。参见比较		MOVWF .....	124
通用寄存器数据 .....	15	NOP .....	124
通用同步 / 异步收发器 (USART)		OPTION .....	124
异步接收器		RETFIE .....	124
设置接收 .....	83	RETLW .....	125
异步接收器模式		RETURN .....	125
地址检测 .....	83	RLF .....	125
框图 .....	83	RRF .....	126
<b>W</b>		SLEEP .....	126
USART		SWAPF .....	127
BRGH 位 .....	73	SUBLW .....	126
波特率发生器 (BRG) .....	73		
采样 .....	74, 75, 76		

# PIC16F627A/628A/648A

---

SUBWF .....	127
TRIS .....	127
XORLW .....	128
XORWF .....	128
指令集综述 .....	115
指令流 / 流水线 .....	13
中断 .....	107
中断源	
比较完成 (CCP) .....	57
捕捉完成 (CCP) .....	56
TMR2 与 PR2 匹配 (PWM) .....	58
中断, 标志位	
CCP1 标志 (CCP1IF 位) .....	56
中断, 使能位	
CCP1 使能 (CCP1IE 位) .....	56

# PIC16F627A/628A/648A

## 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	示例:
器件	温度范围	封装	模式	
<b>器件:</b>	PIC16F627A/628A/648A: 标准 VDD 范围: 3.0V 至 5.5V PIC16F627A/628A/648AT: VDD 范围: 3.0V 至 5.5V (卷带式) PIC16LF627A/628A/648A: VDD 范围: 2.0V 至 5.5V PIC16LF627A/628A/648AT: VDD 范围: 2.0V 至 5.5V (卷带式)			a) PIC16F627A - E/P 301 = 扩展级温度, PDIP 封装, 20 MHz, 正常 VDD 限制, QTP 模式 #301。 b) PIC16LF627A - I/SO = 工业级温度, SOIC 封装, 20 MHz, 扩展 VDD 限制。
<b>温度范围:</b>	I = -40°C 至 +85°C E = -40°C 至 +125°C			
<b>封装:</b>	P = PDIP SO = SOIC (翅形, 300 mil) SS = SSOP (209 mil) ML = QFN (28 引脚)			
<b>模式:</b>	对于 QTP, 为 3 位数字的模式编码 (其他情况空白)			



## 全球销售及服务中心

### 美洲

公司总部 **Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199

Tel: 1-480-792-7200  
Fax: 1-480-792-7277

技术支持:

<http://support.microchip.com>

网址: [www.microchip.com](http://www.microchip.com)

#### 亚特兰大 Atlanta

Alpharetta, GA  
Tel: 1-770-640-0034  
Fax: 1-770-640-0307

#### 波士顿 Boston

Westborough, MA  
Tel: 1-774-760-0087  
Fax: 1-774-760-0088

#### 芝加哥 Chicago

Itasca, IL  
Tel: 1-630-285-0071  
Fax: 1-630-285-0075

#### 达拉斯 Dallas

Addison, TX  
Tel: 1-972-818-7423  
Fax: 1-972-818-2924

#### 底特律 Detroit

Farmington Hills, MI  
Tel: 1-248-538-2250  
Fax: 1-248-538-2260

#### 科科莫 Kokomo

Kokomo, IN  
Tel: 1-765-864-8360  
Fax: 1-765-864-8387

#### 洛杉矶 Los Angeles

Mission Viejo, CA  
Tel: 1-949-462-9523  
Fax: 1-949-462-9608

#### 圣何塞 San Jose

Mountain View, CA  
Tel: 1-650-215-1444  
Fax: 1-650-961-0286

#### 加拿大多伦多 Toronto

Mississauga, Ontario,  
Canada  
Tel: 1-905-673-0699  
Fax: 1-905-673-6509

### 亚太地区

中国 - 北京  
Tel: 86-10-8528-2100  
Fax: 86-10-8528-2104

中国 - 成都  
Tel: 86-28-8676-6200  
Fax: 86-28-8676-6599

中国 - 福州  
Tel: 86-591-8750-3506  
Fax: 86-591-8750-3521

中国 - 香港特别行政区  
Tel: 852-2401-1200  
Fax: 852-2401-3431

中国 - 青岛  
Tel: 86-532-8502-7355  
Fax: 86-532-8502-7205

中国 - 上海  
Tel: 86-21-5407-5533  
Fax: 86-21-5407-5066

中国 - 沈阳  
Tel: 86-24-2334-2829  
Fax: 86-24-2334-2393

中国 - 深圳  
Tel: 86-755-8203-2660  
Fax: 86-755-8203-1760

中国 - 顺德  
Tel: 86-757-2839-5507  
Fax: 86-757-2839-5571

中国 - 武汉  
Tel: 86-27-5980-5300  
Fax: 86-27-5980-5118

中国 - 西安  
Tel: 86-29-8833-7252  
Fax: 86-29-8833-7256

台湾地区 - 高雄  
Tel: 886-7-536-4818  
Fax: 886-7-536-4803

台湾地区 - 台北  
Tel: 886-2-2500-6610  
Fax: 886-2-2508-0102

台湾地区 - 新竹  
Tel: 886-3-572-9526  
Fax: 886-3-572-6459

### 亚太地区

澳大利亚 **Australia - Sydney**  
Tel: 61-2-9868-6733  
Fax: 61-2-9868-6755

印度 **India - Bangalore**  
Tel: 91-80-2229-0061  
Fax: 91-80-2229-0062

印度 **India - New Delhi**  
Tel: 91-11-5160-8631  
Fax: 91-11-5160-8632

印度 **India - Pune**  
Tel: 91-20-2566-1512  
Fax: 91-20-2566-1513

日本 **Japan - Yokohama**  
Tel: 81-45-471-6166  
Fax: 81-45-471-6122

韩国 **Korea - Gumi**  
Tel: 82-54-473-4301  
Fax: 82-54-473-4302

韩国 **Korea - Seoul**  
Tel: 82-2-554-7200  
82-2-558-5932 或  
82-2-558-5934

马来西亚 **Malaysia - Penang**  
Tel: 604-646-8870  
Fax: 604-646-5086

菲律宾 **Philippines - Manila**  
Tel: 632-634-9065  
Fax: 632-634-9069

新加坡 **Singapore**  
Tel: 65-6334-8870  
Fax: 65-6334-8850

泰国 **Thailand - Bangkok**  
Tel: 66-2-694-1351  
Fax: 66-2-694-1350

### 欧洲

奥地利 **Austria - Weis**  
Tel: 43-7242-2244-399  
Fax: 43-7242-2244-393

丹麦 **Denmark - Copenhagen**  
Tel: 45-4450-2828  
Fax: 45-4485-2829

法国 **France - Paris**  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

德国 **Germany - Munich**  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

意大利 **Italy - Milan**  
Tel: 39-0331-742611  
Fax: 39-0331-466781

荷兰 **Netherlands - Drunen**  
Tel: 31-416-690399  
Fax: 31-416-690340

西班牙 **Spain - Madrid**  
Tel: 34-91-352-30-52  
Fax: 34-91-352-11-47

英国 **UK - Wokingham**  
Tel: 44-118-921-5869  
Fax: 44-118-921-5820