



PIC16F882/883/884/886/887 数据手册

采用纳瓦技术的 28/40/44 引脚
增强型闪存 8 位
CMOS 单片机

请注意以下有关 Microchip 器件代码保护功能的要点:

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信: 在正常使用的情况下, Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前, 仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知, 所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

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

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分, 因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利, 它们可能由更新之信息所替代。确保应用符合技术规范, 是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保, 包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和 / 或生命安全应用, 一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时, 会维护和保障 Microchip 免于承担法律责任, 并加以赔偿。在 Microchip 知识产权保护下, 不得暗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、KEELOQ 徽标、microID、MPLAB、PIC、PICmicro、PICSTART、PRO MATE、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、Linear Active Thermistor、Migratable Memory、MXDEV、MXLAB、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、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、FlexROM、fuzzyLAB、In-Circuit Serial Programming、ICSP、ICEPIC、Mindi、MiWi、MPASM、MPLAB Certified 徽标、MPLIB、MPLINK、PICkit、PICDEM、PICDEM.net、PICLAB、PICtail、PowerCal、PowerInfo、PowerMate、PowerTool、REAL ICE、rfLAB、Select Mode、Smart Serial、SmartTel、Total Endurance、UNI/O、WiperLock 和 ZENA 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。

© 2007, 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 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2002 认证。公司在 PIC[®] MCU 与 dsPIC[®] DSC、KEELOQ[®] 跳码器件、串行 EEPROM、单片机外设、非易失性存储器 and 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和生产方面的质量体系也已通过了 ISO 9001:2000 认证。



PIC16F882/883/884/886/887

采用纳瓦技术的 28/40/44 引脚 8 位 CMOS 闪存单片机

高性能 RISC CPU:

- 仅需学习 35 条指令:
 - 除跳转指令外, 所有指令均为单周期指令
- 工作速度
 - 振荡器 / 时钟输入为 DC – 20 MHz
 - 指令周期为 DC – 200 ns
- 中断功能
- 8 级深的硬件堆栈
- 直接、间接和相对寻址模式

单片机的特殊性能:

- 高精度内部振荡器:
 - 出厂时精度校准为 $\pm 1\%$
 - 可通过软件选择的频率范围为: 31 kHz 至 8 MHz
 - 可通过软件调节
 - 双速启动模式
 - 关键应用的晶振故障检测
 - 在工作期间切换时钟模式以节能
- 节能休眠模式
- 宽工作电压范围 (2.0V-5.5V)
- 工业级及扩展级温度范围
- 上电复位 (Power-on Reset, POR)
- 上电延时定时器 (Power-up Timer, PWRT) 和振荡器起振定时器 (Oscillator Start-up Timer, OST)
- 带有软件控制选项的掉电复位 (Brown-out Reset, BOR)
- 带有片上振荡器的增强型低电流看门狗定时器 (Watchdog Timer, WDT), 可软件使能 (在软件选择最大分频比时, 标称周期为 268 秒)
- 带有上拉的主复位引脚, 可复用作输入引脚
- 可编程代码保护
- 高耐用性闪存 /EEPROM 单元:
 - 闪存可承受 10 万次擦写
 - EEPROM 可承受 100 万次擦写
 - 闪存 / 数据 EEPROM 的数据保持时间: > 40 年
- 运行时读 / 写程序存储器
- 在线调试器 (板上)

低功耗特性:

- 待机电流:
 - 2.0V 时典型值为 50 nA
- 工作电流:
 - 32 kHz、2.0V 时典型值为 11 μ A
 - 4 MHz、2.0V 时典型值为 220 μ A
- 看门狗定时器电流:
 - 2.0V 时典型值为 1 μ A

外设特性:

- 24/35 个带有方向可单独控制的 I/O 引脚:
 - 高灌 / 拉电流可直接驱动 LED
 - 电平变化中断引脚
 - 可单独编程的弱上拉引脚
 - 超低功耗唤醒 (Ultra Low-Power Wake-up, ULPWU)
- 模拟比较器模块具有:
 - 两个模拟比较器
 - 可编程片上参考电压 (CVREF) 模块 (占 VDD 的百分比)
 - 固定的参考电压 (0.6V)
 - 可从外部访问比较器的输入和输出
 - SR 锁存模式
 - 外部定时器选通 (使能计数)
- A/D 转换器:
 - 10 位分辨率和 11/14 个通道
- Timer0: 带 8 位可编程预分频器的 8 位定时器 / 计数器
- 增强型 Timer1:
 - 带预分频器的 16 位定时器 / 计数器
 - 外部选通输入模式
 - 专用低功耗 32 kHz 振荡器
- 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
- 增强型捕捉、比较和 PWM+ 模块:
 - 16 位捕捉, 最大分辨率为 12.5 ns
 - 比较, 最大分辨率为 200 ns
 - 带有 1、2 或 4 个输出通道和可编程“死区时间”的 10 位 PWM, 最大频率为 20 kHz
 - PWM 输出转向 (steering) 控制
- 捕捉、比较和 PWM 模块:
 - 16 位捕捉, 最大分辨率为 12.5 ns
 - 16 位比较, 最大分辨率为 200 ns
 - 10 位 PWM, 最大频率为 20 kHz
- 增强型 USART 模块:
 - 支持 RS-485、RS-232 和 LIN 2.0
 - 自动波特率检测
 - 遇到起始位时自动唤醒
- 通过两个引脚进行在线串行编程 (In-Circuit Serial Programming™, ICSP™)
- 主同步串行口 (Master Synchronous Serial Port, MSSP) 模块支持 3 线 SPI (总共 4 种模式) 和带有 I²C 地址屏蔽功能的 I²C™ 主 / 从模式

PIC16F882/883/884/886/887

器件	程序存储器	数据存储器		I/O	10 位 A/D (通道数)	ECCP/ CCP	EUSART	MSSP	比较器	8/16 位 定时器
	闪存 (字)	SRAM (字节)	EEPROM (字节)							
PIC16F882	2048	128	128	28	11	1/1	1	1	2	2/1
PIC16F883	4096	256	256	24	11	1/1	1	1	2	2/1
PIC16F884	4096	256	256	35	14	1/1	1	1	2	2/1
PIC16F886	8192	368	256	24	11	1/1	1	1	2	2/1
PIC16F887	8192	368	256	35	14	1/1	1	1	2	2/1

PIC16F882/883/884/886/887

引脚图—— PIC16F882/883/886 (28 引脚 PDIP、SOIC 和 SSOP)

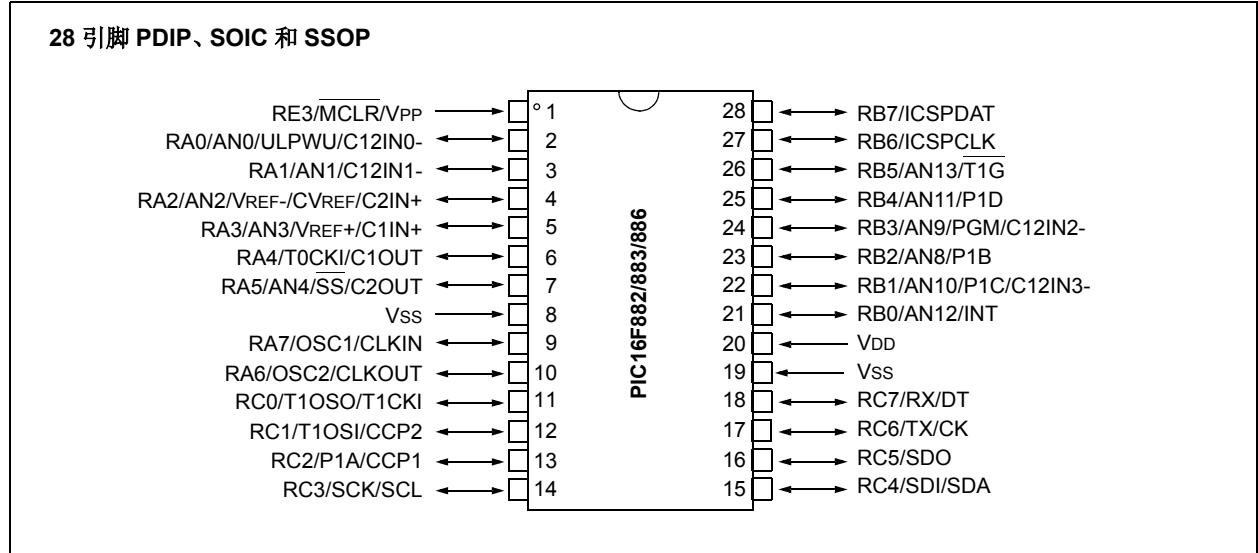


表 1: PIC16F882/883/886 28 引脚汇总 (PDIP、SOIC 和 SSOP)

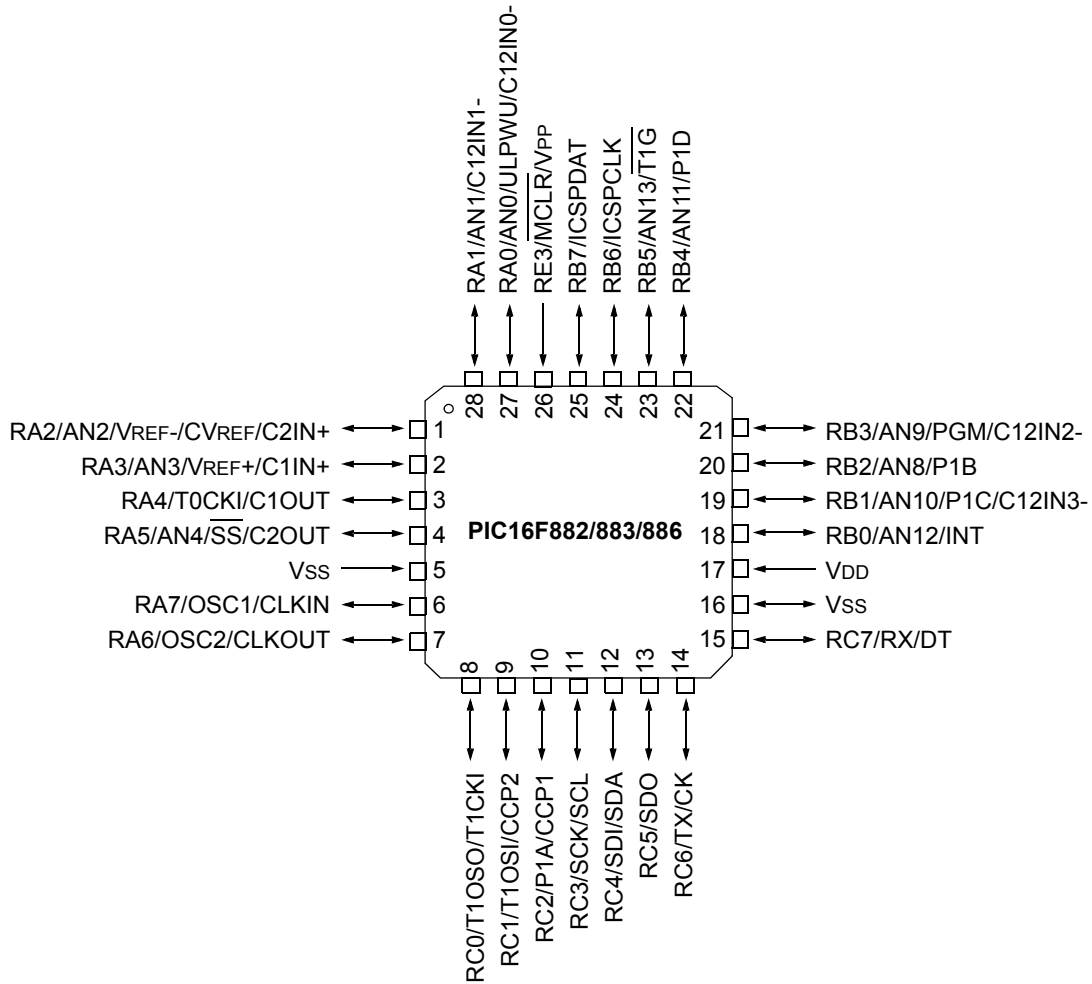
I/O	引脚	模拟	比较器	定时器	ECCP	EUSART	MSSP	中断	上拉	基准
RA0	2	AN0/ULPWU	C12IN0-	—	—	—	—	—	—	—
RA1	3	AN1	C12IN1-	—	—	—	—	—	—	—
RA2	4	AN2	C2IN+	—	—	—	—	—	—	VREF-/CVREF
RA3	5	AN3	C1IN+	—	—	—	—	—	—	VREF+
RA4	6	—	C1OUT	T0CKI	—	—	—	—	—	—
RA5	7	AN4	C2OUT	—	—	—	SS	—	—	—
RA6	10	—	—	—	—	—	—	—	—	OSC2/CLKOUT
RA7	9	—	—	—	—	—	—	—	—	OSC1/CLKIN
RB0	21	AN12	—	—	—	—	—	IOC/INT	Y	—
RB1	22	AN10	C12IN3-	—	P1C	—	—	IOC	Y	—
RB2	23	AN8	—	—	P1B	—	—	IOC	Y	—
RB3	24	AN9	C12IN2-	—	—	—	—	IOC	Y	PGM
RB4	25	AN11	—	—	P1D	—	—	IOC	Y	—
RB5	26	AN13	—	T1G	—	—	—	IOC	Y	—
RB6	27	—	—	—	—	—	—	IOC	Y	ICSPCLK
RB7	28	—	—	—	—	—	—	IOC	Y	ICSPDAT
RC0	11	—	—	T1OSO/T1CKI	—	—	—	—	—	—
RC1	12	—	—	T1OSI	CCP2	—	—	—	—	—
RC2	13	—	—	—	CCP1/P1A	—	—	—	—	—
RC3	14	—	—	—	—	—	SCK/SCL	—	—	—
RC4	15	—	—	—	—	—	SDI/SDA	—	—	—
RC5	16	—	—	—	—	—	SDO	—	—	—
RC6	17	—	—	—	—	TX/CK	—	—	—	—
RC7	18	—	—	—	—	RX/DT	—	—	—	—
RE3	1	—	—	—	—	—	—	—	Y ⁽¹⁾	MCLR/VPP
—	20	—	—	—	—	—	—	—	—	VDD
—	8	—	—	—	—	—	—	—	—	VSS
—	19	—	—	—	—	—	—	—	—	VSS

注 1: 只有在采用外部 MCLR 配置时才能激活上拉。

PIC16F882/883/884/886/887

引脚图——PIC16F882/883/886 (28 引脚 QFN)

28 引脚 QFN



PIC16F882/883/884/886/887

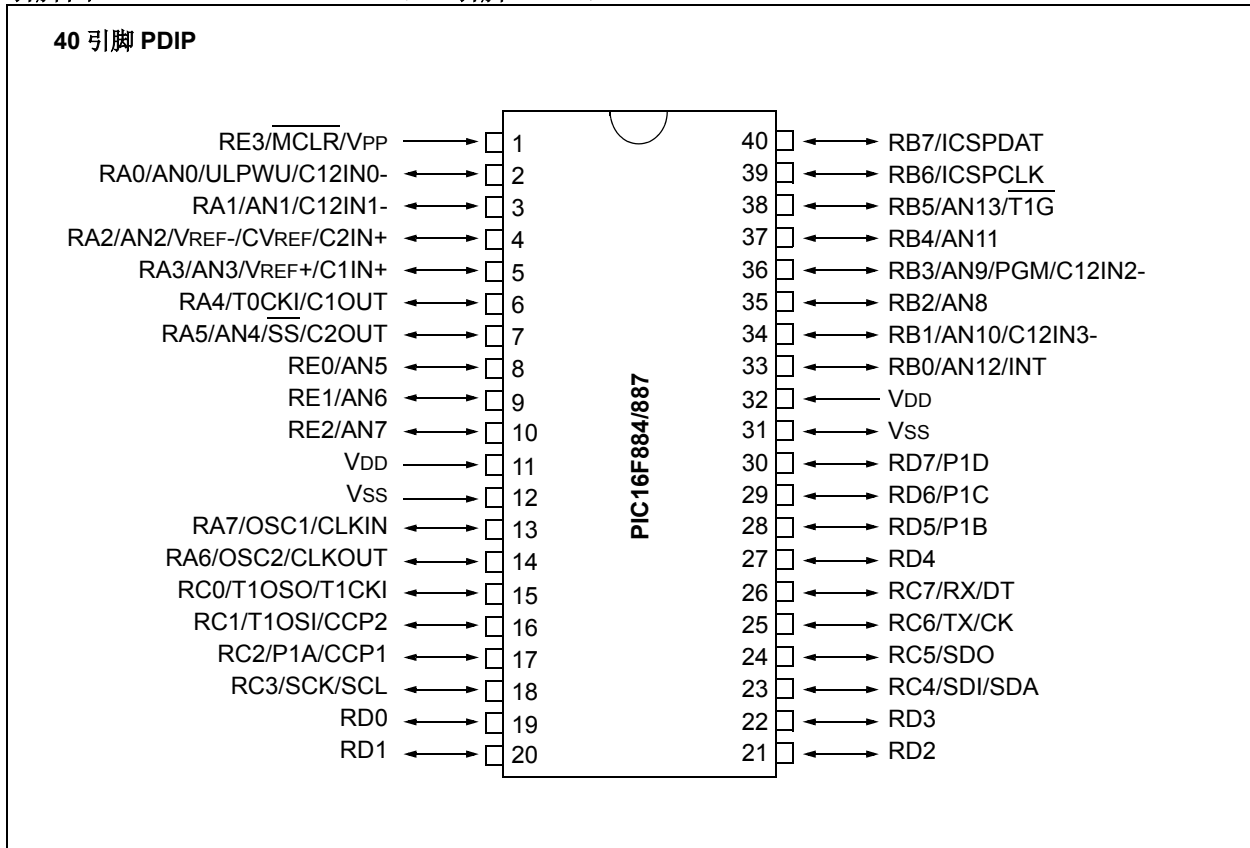
表 2: PIC16F882/883/886 28 引脚汇总 (QFN)

I/O	引脚	模拟	比较器	定时器	ECCP	EUSART	MSSP	中断	上拉	基准
RA0	27	AN0/ULPWU	C12IN0-	—	—	—	—	—	—	—
RA1	28	AN1	C12IN1-	—	—	—	—	—	—	—
RA2	1	AN2	C2IN+	—	—	—	—	—	—	VREF-/CVREF
RA3	2	AN3	C1IN+	—	—	—	—	—	—	VREF+
RA4	3	—	C1OUT	T0CKI	—	—	—	—	—	—
RA5	4	AN4	C2OUT	—	—	—	SS	—	—	—
RA6	7	—	—	—	—	—	—	—	—	OSC2/CLKOUT
RA7	6	—	—	—	—	—	—	—	—	OSC1/CLKIN
RB0	18	AN12	—	—	—	—	—	IOC/INT	Y	—
RB1	19	AN10	C12IN3-	—	P1C	—	—	IOC	Y	—
RB2	20	AN8	—	—	P1B	—	—	IOC	Y	—
RB3	21	AN9	C12IN2-	—	—	—	—	IOC	Y	PGM
RB4	22	AN11	—	—	P1D	—	—	IOC	Y	—
RB5	23	AN13	—	T1G	—	—	—	IOC	Y	—
RB6	24	—	—	—	—	—	—	IOC	Y	ICSPCLK
RB7	25	—	—	—	—	—	—	IOC	Y	ICSPDAT
RC0	8	—	—	T1OSO/T1CKI	—	—	—	—	—	—
RC1	9	—	—	T1OSI	CCP2	—	—	—	—	—
RC2	10	—	—	—	CCP1/P1A	—	—	—	—	—
RC3	11	—	—	—	—	—	SCK/SCL	—	—	—
RC4	12	—	—	—	—	—	SDI/SDA	—	—	—
RC5	13	—	—	—	—	—	SDO	—	—	—
RC6	14	—	—	—	—	TX/CK	—	—	—	—
RC7	15	—	—	—	—	RX/DT	—	—	—	—
RE3	26	—	—	—	—	—	—	—	γ ⁽¹⁾	MCLR/VPP
—	17	—	—	—	—	—	—	—	—	VDD
—	5	—	—	—	—	—	—	—	—	VSS
—	16	—	—	—	—	—	—	—	—	VSS

注 1: 只有在采用外部 MCLR 配置时才能激活上拉。

PIC16F882/883/884/886/887

引脚图 —— PIC16F884/887 (40 引脚 PDIP)



PIC16F882/883/884/886/887

表 3: PIC16F884/887 40 引脚汇总 (PDIP)

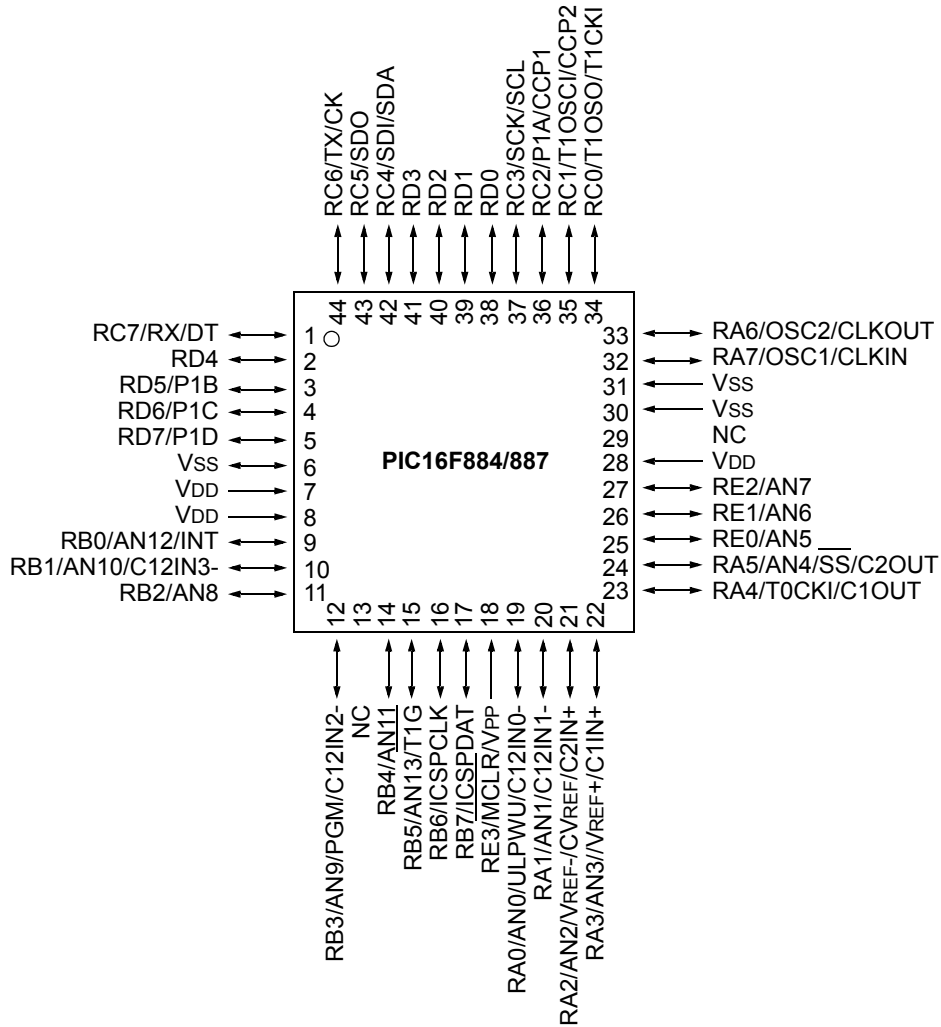
I/O	引脚	模拟	比较器	定时器	ECCP	EUSART	MSSP	中断	上拉	基准
RA0	2	AN0/ULPWU	C12IN0-	—	—	—	—	—	—	—
RA1	3	AN1	C12IN1-	—	—	—	—	—	—	—
RA2	4	AN2	C2IN+	—	—	—	—	—	—	VREF-/CVREF
RA3	5	AN3	C1IN+	—	—	—	—	—	—	VREF+
RA4	6	—	C1OUT	T0CKI	—	—	—	—	—	—
RA5	7	AN4	C2OUT	—	—	—	SS	—	—	—
RA6	14	—	—	—	—	—	—	—	—	OSC2/CLKOUT
RA7	13	—	—	—	—	—	—	—	—	OSC1/CLKIN
RB0	33	AN12	—	—	—	—	—	IOC/INT	Y	—
RB1	34	AN10	C12IN3-	—	—	—	—	IOC	Y	—
RB2	35	AN8	—	—	—	—	—	IOC	Y	—
RB3	36	AN9	C12IN2-	—	—	—	—	IOC	Y	PGM
RB4	37	AN11	—	—	—	—	—	IOC	Y	—
RB5	38	AN13	—	T1G	—	—	—	IOC	Y	—
RB6	39	—	—	—	—	—	—	IOC	Y	ICSPCLK
RB7	40	—	—	—	—	—	—	IOC	Y	ICSPDAT
RC0	15	—	—	T1OSO/T1CKI	—	—	—	—	—	—
RC1	16	—	—	T1OSI	CCP2	—	—	—	—	—
RC2	17	—	—	—	CCP1/P1A	—	—	—	—	—
RC3	18	—	—	—	—	—	SCK/SCL	—	—	—
RC4	23	—	—	—	—	—	SDI/SDA	—	—	—
RC5	24	—	—	—	—	—	SDO	—	—	—
RC6	25	—	—	—	—	TX/CK	—	—	—	—
RC7	26	—	—	—	—	RX/DT	—	—	—	—
RD0	19	—	—	—	—	—	—	—	—	—
RD1	20	—	—	—	—	—	—	—	—	—
RD2	21	—	—	—	—	—	—	—	—	—
RD3	22	—	—	—	—	—	—	—	—	—
RD4	27	—	—	—	—	—	—	—	—	—
RD5	28	—	—	—	P1B	—	—	—	—	—
RD6	29	—	—	—	P1C	—	—	—	—	—
RD7	30	—	—	—	P1D	—	—	—	—	—
RE0	8	AN5	—	—	—	—	—	—	—	—
RE1	9	AN6	—	—	—	—	—	—	—	—
RE2	10	AN7	—	—	—	—	—	—	—	—
RE3	1	—	—	—	—	—	—	—	Y ⁽¹⁾	MCLR/VPP
—	11	—	—	—	—	—	—	—	—	VDD
—	32	—	—	—	—	—	—	—	—	VDD
—	12	—	—	—	—	—	—	—	—	VSS
—	31	—	—	—	—	—	—	—	—	VSS

注 1: 只有在采用外部 MCLR 配置时才能激活上拉。

PIC16F882/883/884/886/887

引脚图—— PIC16F884/887 (44 引脚 QFN)

44 引脚 QFN



PIC16F882/883/884/886/887

表 4: PIC16F884/887 44 引脚汇总 (QFN)

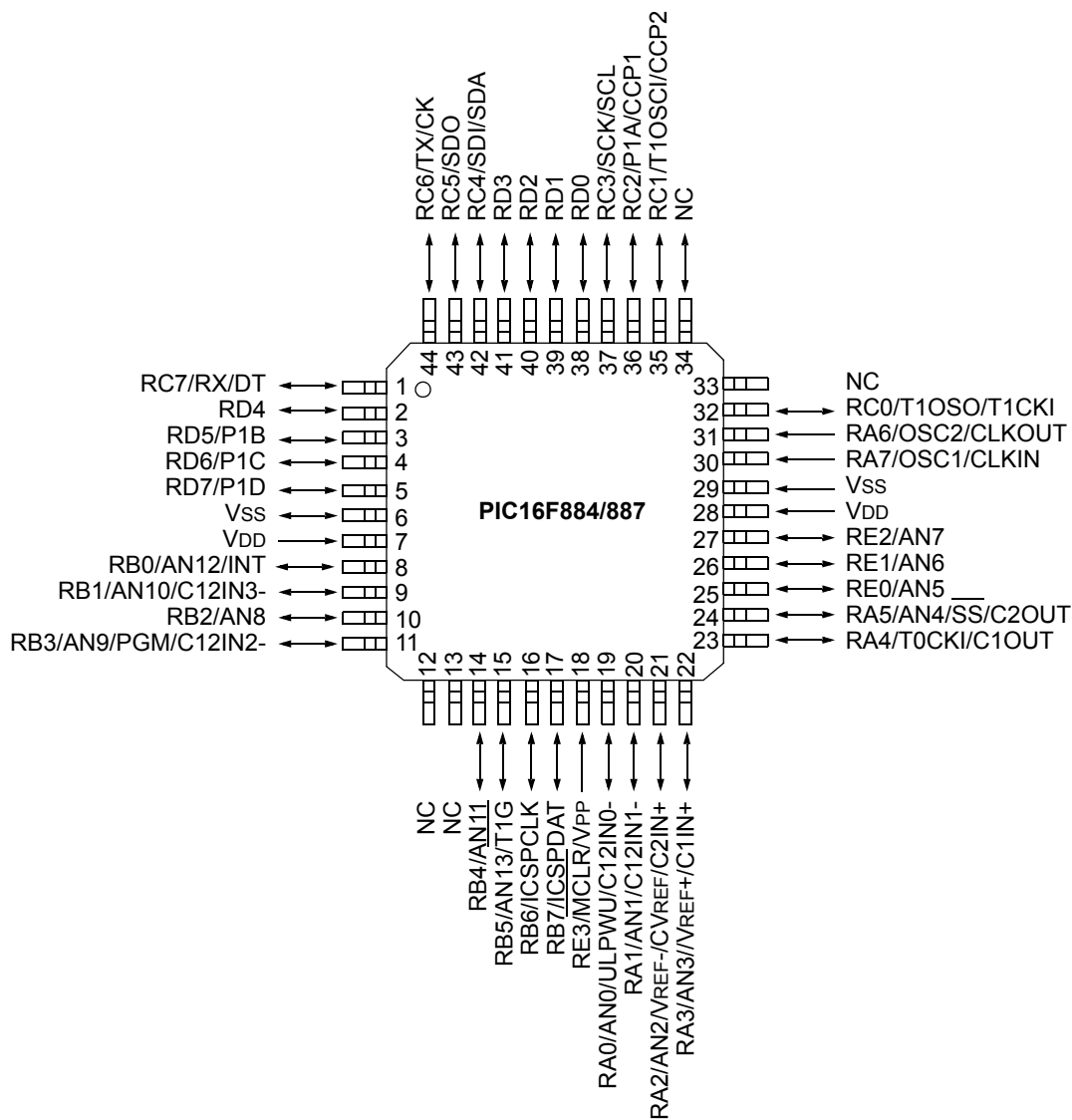
I/O	引脚	模拟	比较器	定时器	ECCP	EUSART	MSSP	中断	上拉	基准
RA0	19	AN0/ULPWU	C12IN0-	—	—	—	—	—	—	—
RA1	20	AN1	C12IN1-	—	—	—	—	—	—	—
RA2	21	AN2	C2IN+	—	—	—	—	—	—	VREF-/CVREF
RA3	22	AN3	C1IN+	—	—	—	—	—	—	VREF+
RA4	23	—	C1OUT	T0CKI	—	—	—	—	—	—
RA5	24	AN4	C2OUT	—	—	—	SS	—	—	—
RA6	33	—	—	—	—	—	—	—	—	OSC2/CLKOUT
RA7	32	—	—	—	—	—	—	—	—	OSC1/CLKIN
RB0	9	AN12	—	—	—	—	—	IOC/INT	Y	—
RB1	10	AN10	C12IN3-	—	—	—	—	IOC	Y	—
RB2	11	AN8	—	—	—	—	—	IOC	Y	—
RB3	12	AN9	C12IN2-	—	—	—	—	IOC	Y	PGM
RB4	14	AN11	—	—	—	—	—	IOC	Y	—
RB5	15	AN13	—	T1G	—	—	—	IOC	Y	—
RB6	16	—	—	—	—	—	—	IOC	Y	ICSPCLK
RB7	17	—	—	—	—	—	—	IOC	Y	ICSPDAT
RC0	34	—	—	T1OSO/T1CKI	—	—	—	—	—	—
RC1	35	—	—	T1OSI	CCP2	—	—	—	—	—
RC2	36	—	—	—	CCP1/P1A	—	—	—	—	—
RC3	37	—	—	—	—	—	SCK/SCL	—	—	—
RC4	42	—	—	—	—	—	SDI/SDA	—	—	—
RC5	43	—	—	—	—	—	SDO	—	—	—
RC6	44	—	—	—	—	TX/CK	—	—	—	—
RC7	1	—	—	—	—	RX/DT	—	—	—	—
RD0	38	—	—	—	—	—	—	—	—	—
RD1	39	—	—	—	—	—	—	—	—	—
RD2	40	—	—	—	—	—	—	—	—	—
RD3	41	—	—	—	—	—	—	—	—	—
RD4	2	—	—	—	—	—	—	—	—	—
RD5	3	—	—	—	P1B	—	—	—	—	—
RD6	4	—	—	—	P1C	—	—	—	—	—
RD7	5	—	—	—	P1D	—	—	—	—	—
RE0	25	AN5	—	—	—	—	—	—	—	—
RE1	26	AN6	—	—	—	—	—	—	—	—
RE2	27	AN7	—	—	—	—	—	—	—	—
RE3	18	—	—	—	—	—	—	—	Y ⁽¹⁾	MCLR/VPP
—	7	—	—	—	—	—	—	—	—	VDD
—	8	—	—	—	—	—	—	—	—	VDD
—	28	—	—	—	—	—	—	—	—	VDD
—	6	—	—	—	—	—	—	—	—	VSS
—	30	—	—	—	—	—	—	—	—	VSS
—	31	—	—	—	—	—	—	—	—	VSS
—	13	—	—	—	—	—	—	—	—	NC (无连接)
—	29	—	—	—	—	—	—	—	—	NC (无连接)

注 1: 只有在采用外部 MCLR 配置时才能激活上拉。

PIC16F882/883/884/886/887

引脚图——PIC16F884/887 (44 引脚 TQFP)

44 引脚 TQFP



PIC16F882/883/884/886/887

表 5: PIC16F884/887 44 引脚汇总 (TQFP)

I/O	引脚	模拟	比较器	定时器	ECCP	EUSART	MSSP	中断	上拉	基准
RA0	19	AN0/ULPWU	C12IN0-	—	—	—	—	—	—	—
RA1	20	AN1	C12IN1-	—	—	—	—	—	—	—
RA2	21	AN2	C2IN+	—	—	—	—	—	—	VREF-/CVREF
RA3	22	AN3	C1IN+	—	—	—	—	—	—	VREF+
RA4	23	—	C1OUT	T0CKI	—	—	—	—	—	—
RA5	24	AN4	C2OUT	—	—	—	SS	—	—	—
RA6	31	—	—	—	—	—	—	—	—	OSC2/CLKOUT
RA7	31	—	—	—	—	—	—	—	—	OSC1/CLKIN
RB0	8	AN12	—	—	—	—	—	IOC/INT	Y	—
RB1	9	AN10	C12IN3-	—	—	—	—	IOC	Y	—
RB2	10	AN8	—	—	—	—	—	IOC	Y	—
RB3	11	AN9	C12IN2-	—	—	—	—	IOC	Y	PGM
RB4	14	AN11	—	—	—	—	—	IOC	Y	—
RB5	15	AN13	—	T1G	—	—	—	IOC	Y	—
RB6	16	—	—	—	—	—	—	IOC	Y	ICSPCLK
RB7	17	—	—	—	—	—	—	IOC	Y	ICSPDAT
RC0	32	—	—	T1OSO/T1CKI	—	—	—	—	—	—
RC1	35	—	—	T1OSI	CCP2	—	—	—	—	—
RC2	36	—	—	—	CCP1/P1A	—	—	—	—	—
RC3	37	—	—	—	—	—	SCK/SCL	—	—	—
RC4	42	—	—	—	—	—	SDI/SDA	—	—	—
RC5	43	—	—	—	—	—	SDO	—	—	—
RC6	44	—	—	—	—	TX/CK	—	—	—	—
RC7	1	—	—	—	—	RX/DT	—	—	—	—
RD0	38	—	—	—	—	—	—	—	—	—
RD1	39	—	—	—	—	—	—	—	—	—
RD2	40	—	—	—	—	—	—	—	—	—
RD3	41	—	—	—	—	—	—	—	—	—
RD4	2	—	—	—	—	—	—	—	—	—
RD5	3	—	—	—	P1B	—	—	—	—	—
RD6	4	—	—	—	P1C	—	—	—	—	—
RD7	5	—	—	—	P1D	—	—	—	—	—
RE0	25	AN5	—	—	—	—	—	—	—	—
RE1	26	AN6	—	—	—	—	—	—	—	—
RE2	27	AN7	—	—	—	—	—	—	—	—
RE3	18	—	—	—	—	—	—	—	Y ⁽¹⁾	MCLR/VPP
—	7	—	—	—	—	—	—	—	—	VDD
—	28	—	—	—	—	—	—	—	—	VDD
—	6	—	—	—	—	—	—	—	—	VSS
—	13	—	—	—	—	—	—	—	—	NC (无连接)
—	29	—	—	—	—	—	—	—	—	VSS
—	34	—	—	—	—	—	—	—	—	NC (无连接)
—	33	—	—	—	—	—	—	—	—	NC (无连接)
—	12	—	—	—	—	—	—	—	—	NC (无连接)

注 1: 只有在采用外部 MCLR 配置时才能激活上拉。

PIC16F882/883/884/886/887

目录

1.0 器件概述	13
2.0 存储器的构成	21
3.0 I/O 端口	39
4.0 振荡器模块（带故障保护时钟监视器）	61
5.0 Timer0 模块	73
6.0 带有门控的 Timer1 模块	76
7.0 Timer2 模块	81
8.0 比较器模块	83
9.0 模数转换器（ADC）模块	99
10.0 数据 EEPROM 和闪存程序存储器控制	111
11.0 增强型捕捉 / 比较 / PWM 模块	123
12.0 增强型通用同步异步收发器（EUSART）	149
13.0 主同步串行端口（MSSP）模块	175
14.0 CPU 的特殊性能	205
15.0 指令集综述	225
16.0 开发支持	235
17.0 电气规范	239
18.0 DC 和 AC 特性图表	261
19.0 封装信息	263
附录 A: 数据手册的版本历史	273
附录 B: 从其他 PIC® 器件移植	273
索引	275
Microchip 网站	283
变更通知客户服务	283
客户支持	283
读者反馈表	284
产品标识体系	285

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，或将本数据手册后附的《读者反馈表》传真到 86-21-5407 5066。我们期待您的反馈。

最新数据手册

欲获得本数据手册的最新版本，请查询我公司的网站：

<http://www.microchip.com>

查看数据手册中任意一页下边角处的文献编号即可确定其版本。文献编号中数字串后的字母是版本号，例如：DS30000A 是 DS30000 的 A 版本。

勘误表

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

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

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

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

客户通知系统

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

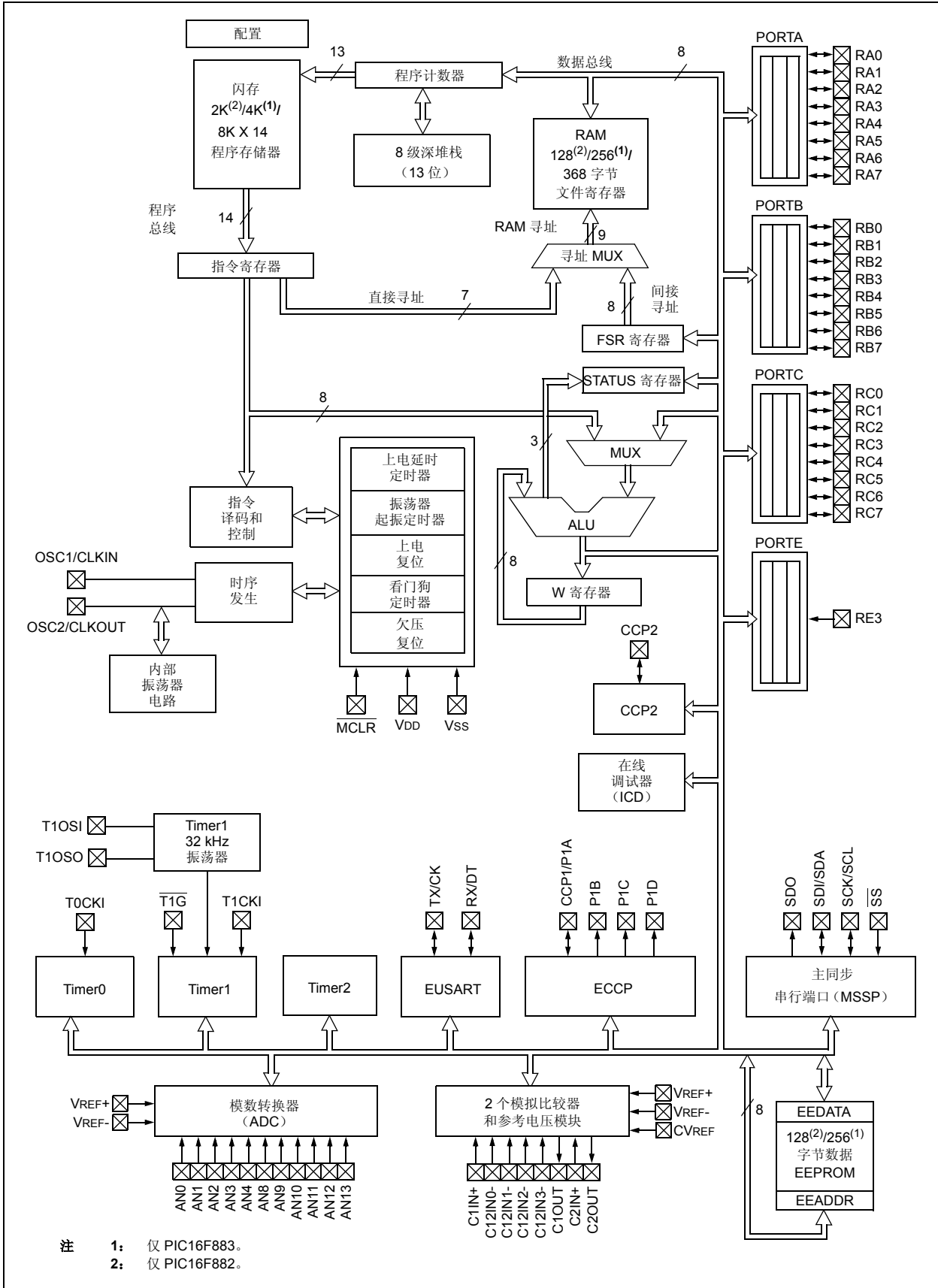
PIC16F882/883/884/886/887

1.0 器件概述

本数据手册涵盖了 PIC16F882/883/884/886/887 器件。PIC16F882/883/886 的封装类型有：28 引脚 PDIP、SOIC、SSOP 和 QFN。PIC16F884/887 的封装类型有：40 引脚 PDIP 以及 44 引脚 QFN 和 TQFP。图 1-1 给出了 PIC16F882/883/886 器件的框图，图 1-2 给出了 PIC16F884/887 器件的框图。表 1-1 和表 1-2 给出了相应的引脚配置说明。

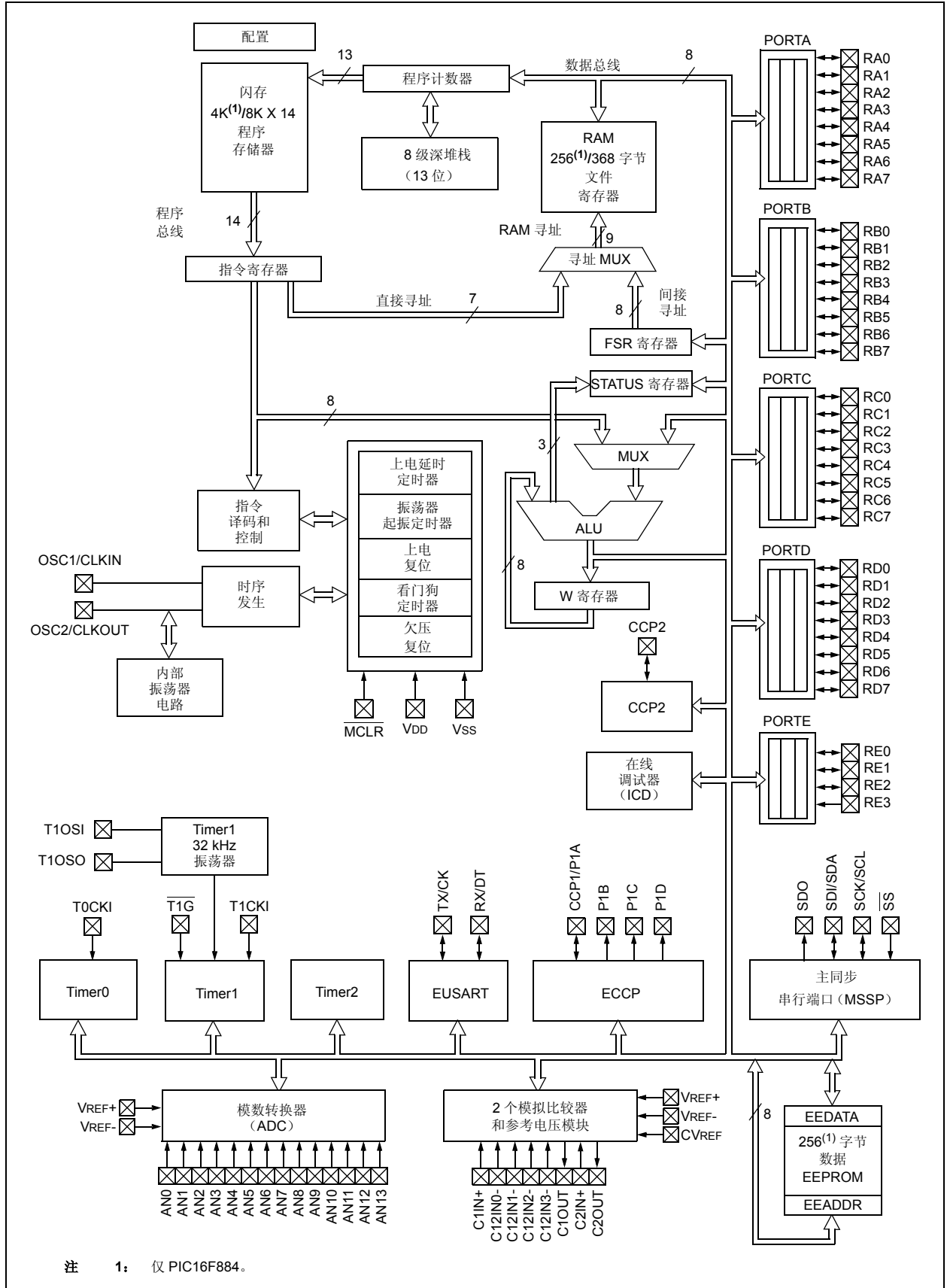
PIC16F882/883/884/886/887

图 1-1: PIC16F882/883/886 框图



PIC16F882/883/884/886/887

图 1-2: PIC16F884/887 框图



PIC16F882/883/884/886/887

表 1-1: PIC16F882/883/886 引脚配置说明

名称	功能	输入类型	输出类型	描述
RA0/AN0/ULPWU/C12IN0-	RA0	TTL	CMOS	通用 I/O。
	AN0	AN	—	A/D 通道 0 输入。
	ULPWU	AN	—	超低功耗唤醒输入。
	C12IN0-	AN	—	比较器 C1 或 C2 的负输入。
RA1/AN1/C12IN1-	RA1	TTL	CMOS	通用 I/O。单独使能的上拉。
	AN1	AN	—	A/D 通道 1 输入。
	C12IN1-	AN	—	比较器 C1 或 C2 的负输入。
RA2/AN2/VREF-/CVREF/C2IN+	RA2	TTL	CMOS	通用 I/O。
	AN2	AN	—	A/D 通道 2。
	VREF-	AN	—	A/D 负参考电压输入。
	CVREF	—	AN	比较器参考电压输出。
	C2IN+	AN	—	比较器 C2 的正输入。
RA3/AN3/VREF+/C1IN+	RA3	TTL	—	通用 I/O。
	AN3	AN	—	A/D 通道 3。
	VREF+	AN	—	编程电压。
	C1IN+	AN	—	比较器 C1 的正输入。
RA4/T0CKI/C1OUT	RA4	TTL	CMOS	通用 I/O。单独使能的上拉。
	T0CKI	ST	—	Timer0 时钟输入。
	C1OUT	—	CMOS	比较器 C1 输出。
RA5/AN4/SS/C2OUT	RA5	TTL	CMOS	通用 I/O。
	AN4	AN	—	A/D 通道 4。
	SS	ST	—	从动选择输入。
	C2OUT	—	CMOS	比较器 C2 输出。
RA6/OSC2/CLKOUT	RA6	TTL	CMOS	通用 I/O。
	OSC2	—	XTAL	带有内部上拉的主复位。
	CLKOUT	—	CMOS	Fosc/4 输出。
RA7/OSC1/CLKIN	RA7	TTL	CMOS	通用 I/O。
	OSC1	XTAL	—	晶振 / 谐振器。
	CLKIN	ST	—	外部时钟输入 /RC 振荡器连接。
RB0/AN12/INT	RB0	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	AN12	AN	—	A/D 通道 12。
	INT	ST	—	外部中断。
RB1/AN10/P1C/C12IN3-	RB1	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	AN10	AN	—	A/D 通道 10。
	P1C	—	CMOS	PWM 输出。
	C12IN3-	AN	—	比较器 C1 或 C2 的负输入。
RB2/AN8/P1B	RB2	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	AN8	AN	—	A/D 通道 8。
	P1B	—	CMOS	PWM 输出。

图注: AN = 模拟输入或输出
TTL = TTL 兼容输入
HV = 高电压

CMOS = CMOS 兼容输入或输出
ST = CMOS 电平的施密特触发器输入
XTAL = 晶振

OD = 漏极开路

PIC16F882/883/884/886/887

表 1-1: PIC16F882/883/886 引脚配置说明 (续)

名称	功能	输入类型	输出类型	描述
RB3/AN9/PGM/C12IN2-	RB3	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	AN9	AN	—	A/D 通道 9。
	PGM	ST	—	低电压 ICSP™ 编程使能引脚。
	C12IN2-	AN	—	比较器 C1 或 C2 的负输入。
RB4/AN11/P1D	RB4	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	AN11	AN	—	A/D 通道 11。
	P1D	—	CMOS	PWM 输出。
RB5/AN13/T1G	RB5	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	AN13	AN	—	A/D 通道 13。
	T1G	ST	—	Timer1 选通输入。
RB6/ICSPCLK	RB6	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	ICSPCLK	ST	—	串行编程时钟。
RB7/ICSPDAT	RB7	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	ICSPDAT	ST	CMOS	ICSP™ 数据 I/O。
RC0/T1OSO/T1CKI	RC0	ST	CMOS	通用 I/O。
	T1OSO	—	CMOS	Timer1 振荡器输出。
	T1CKI	ST	—	Timer1 时钟输入。
RC1/T1OSI/CCP2	RC1	ST	CMOS	通用 I/O。
	T1OSI	ST	—	Timer1 振荡器输入。
	CCP2	ST	CMOS	捕捉 / 比较 / PWM2。
RC2/P1A/CCP1	RC2	ST	CMOS	通用 I/O。
	P1A	—	CMOS	PWM 输出。
	CCP1	ST	CMOS	捕捉 / 比较 / PWM1。
RC3/SCK/SCL	RC3	ST	CMOS	通用 I/O。
	SCK	ST	CMOS	SPI 时钟。
	SCL	ST	OD	I ² C™ 时钟。
RC4/SDI/SDA	RC4	ST	CMOS	通用 I/O。
	SDI	ST	—	SPI 数据输入。
	SDA	ST	OD	I ² C 数据输入 / 输出。
RC5/SDO	RC5	ST	CMOS	通用 I/O。
	SDO	—	CMOS	SPI 数据输出。
RC6/TX/CK	RC6	ST	CMOS	通用 I/O。
	TX	—	CMOS	EUSART 异步发送。
	CK	ST	CMOS	EUSART 同步时钟。
RC7/RX/DT	RC7	ST	CMOS	通用 I/O。
	RX	ST	—	EUSART 异步输入。
	DT	ST	CMOS	EUSART 同步数据。
RE3/MCLR/VPP	RE3	TTL	—	通用输入。
	MCLR	ST	—	带有内部上拉的主复位。
	VPP	HV	—	编程电压。
VSS	VSS	电源	—	接地参考端。
VDD	VDD	电源	—	正电源。

图注: AN = 模拟输入或输出 CMOS = CMOS 兼容输入或输出 OD = 漏极开路
TTL = TTL 兼容输入 ST = CMOS 电平的施密特触发器输入
HV = 高电压 XTAL = 晶振

PIC16F882/883/884/886/887

表 1-2: PIC16F884/887 引脚配置说明

名称	功能	输入类型	输出类型	描述
RA0/AN0/ULPWU/C12IN0-	RA0	TTL	CMOS	通用 I/O。
	AN0	AN	—	A/D 通道 0 输入。
	ULPWU	AN	—	超低功耗唤醒输入。
	C12IN0-	AN	—	比较器 C1 或 C2 的负输入。
RA1/AN1/C12IN1-	RA1	TTL	CMOS	通用 I/O。
	AN1	AN	—	A/D 通道 1 输入。
	C12IN1-	AN	—	比较器 C1 或 C2 的负输入。
RA2/AN2/VREF-/CVREF/C2IN+	RA2	TTL	CMOS	通用 I/O。
	AN2	AN	—	A/D 通道 2。
	VREF-	AN	—	A/D 负参考电压输入。
	CVREF	—	AN	比较器参考电压输出。
	C2IN+	AN	—	比较器 C2 的正输入。
RA3/AN3/VREF+/C1IN+	RA3	TTL	CMOS	通用 I/O。
	AN3	AN	—	A/D 通道 3。
	VREF+	AN	—	A/D 正参考电压输入。
	C1IN+	AN	—	比较器 C1 的正输入。
RA4/T0CKI/C1OUT	RA4	TTL	CMOS	通用 I/O。
	T0CKI	ST	—	Timer0 时钟输入。
	C1OUT	—	CMOS	比较器 C1 输出。
RA5/AN4/SS/C2OUT	RA5	TTL	CMOS	通用 I/O。
	AN4	AN	—	A/D 通道 4。
	SS	ST	—	从动选择输入。
	C2OUT	—	CMOS	比较器 C2 输出。
RA6/OSC2/CLKOUT	RA6	TTL	CMOS	通用 I/O。
	OSC2	—	XTAL	晶振 / 谐振器。
	CLKOUT	—	CMOS	Fosc/4 输出。
RA7/OSC1/CLKIN	RA7	TTL	CMOS	通用 I/O。
	OSC1	XTAL	—	晶振 / 谐振器。
	CLKIN	ST	—	外部时钟输入 / RC 振荡器连接。
RB0/AN12/INT	RB0	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	AN12	AN	—	A/D 通道 12。
	INT	ST	—	外部中断。
RB1/AN10/C12IN3-	RB1	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	AN10	AN	—	A/D 通道 10。
	C12IN3-	AN	—	比较器 C1 或 C2 的负输入。
RB2/AN8	RB2	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	AN8	AN	—	A/D 通道 8。
RB3/AN9/PGM/C12IN2-	RB3	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	AN9	AN	—	A/D 通道 9。
	PGM	ST	—	低电压 ICSP™ 编程使能引脚。
	C12IN2-	AN	—	比较器 C1 或 C2 的负输入。

图注: AN = 模拟输入或输出 CMOS = CMOS 兼容输入或输出 OD = 漏极开路
TTL = TTL 兼容输入 ST = CMOS 电平的施密特触发器输入
HV = 高电压 XTAL = 晶振

PIC16F882/883/884/886/887

表 1-2: PIC16F884/887 引脚配置说明 (续)

名称	功能	输入类型	输出类型	描述
RB4/AN11	RB4	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	AN11	AN	—	A/D 通道 11。
RB5/AN13/T1G	RB5	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	AN13	AN	—	A/D 通道 13。
	T1G	ST	—	Timer1 选通输入。
RB6/ICSPCLK	RB6	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	ICSPCLK	ST	—	串行编程时钟。
RB7/ICSPDAT	RB7	TTL	CMOS	通用 I/O。单独控制的电平变化中断。单独使能的上拉。
	ICSPDAT	ST	TTL	ICSP™ 数据 I/O。
RC0/T1OSO/T1CKI	RC0	ST	CMOS	通用 I/O。
	T1OSO	—	XTAL	Timer1 振荡器输出。
	T1CKI	ST	—	Timer1 时钟输入。
RC1/T1OSI/CCP2	RC1	ST	CMOS	通用 I/O。
	T1OSI	XTAL	—	Timer1 振荡器输入。
	CCP2	ST	CMOS	捕捉 / 比较 / PWM2。
RC2/P1A/CCP1	RC2	ST	CMOS	通用 I/O。
	P1A	ST	CMOS	PWM 输出。
	CCP1	—	CMOS	捕捉 / 比较 / PWM1。
RC3/SCK/SCL	RC3	ST	CMOS	通用 I/O。
	SCK	ST	CMOS	SPI 时钟。
	SCL	ST	OD	I ² C™ 时钟。
RC4/SDI/SDA	RC4	ST	CMOS	通用 I/O。
	SDI	ST	—	SPI 数据输入。
	SDA	ST	OD	I ² C 数据输入 / 输出。
RC5/SDO	RC5	ST	CMOS	通用 I/O。
	SDO	—	CMOS	SPI 数据输出。
RC6/TX/CK	RC6	ST	CMOS	通用 I/O。
	TX	—	CMOS	EUSART 异步发送。
	CK	ST	CMOS	EUSART 同步时钟。
RC7/RX/DT	RC7	ST	CMOS	通用 I/O。
	RX	ST	—	EUSART 异步输入。
	DT	ST	CMOS	EUSART 同步数据。
RD0	RD0	TTL	CMOS	通用 I/O。
RD1	RD1	TTL	CMOS	通用 I/O。
RD2	RD2	TTL	CMOS	通用 I/O。
RD3	RD3	TTL	CMOS	通用 I/O。
RD4	RD4	TTL	CMOS	通用 I/O。
RD5/P1B	RD5	TTL	CMOS	通用 I/O。
	P1B	—	CMOS	PWM 输出。
RD6/P1C	RD6	TTL	CMOS	通用 I/O。
	P1C	—	CMOS	PWM 输出。

图注: AN = 模拟输入或输出 CMOS = CMOS 兼容输入或输出 OD = 漏极开路
TTL = TTL 兼容输入 ST = CMOS 电平的施密特触发器输入
HV = 高电压 XTAL = 晶振

PIC16F882/883/884/886/887

表 1-2: PIC16F884/887 引脚配置说明 (续)

名称	功能	输入类型	输出类型	描述
RD7/P1D	RD7	TTL	CMOS	通用 I/O。
	P1D	—	CMOS	PWM 输出。
RE0/AN5	RE0	TTL	CMOS	通用 I/O。
	AN5	AN	—	A/D 通道 5。
RE1/AN6	RE1	ST	CMOS	通用 I/O。
	AN6	AN	—	A/D 通道 6。
RE2/AN7	RE2	TTL	CMOS	通用 I/O。
	AN7	AN	—	A/D 通道 7。
RE3/MCLR/VPP	RE3	TTL	—	通用输入。
	MCLR	ST	—	带有内部上拉的主复位。
	VPP	HV	—	编程电压。
VSS	VSS	电源	—	接地参考端。
VDD	VDD	电源	—	正电源。

图注: AN = 模拟输入或输出 CMOS = CMOS 兼容输入或输出 OD = 漏极开路
TTL = TTL 兼容输入 ST = CMOS 电平的施密特触发器输入
HV = 高电压 XTAL = 晶振

PIC16F882/883/884/886/887

2.0 存储器的构成

2.1 程序存储器的构成

PIC16F882/883/884/886/887 具备 13 位的程序计数器 (program counter, PC), 可寻址 PIC16F882 的 2K x 14 (0000h-07FFh) 的程序存储空间, PIC16F883 和 PIC16F884 的 4K x 14 (0000h-0FFFh) 的程序存储空间以及 PIC16F886 和 PIC16F887 的 8K x 14 (0000h-1FFFh) 的程序存储空间。访问超出以上边界的单元会导致折回到第一个 8K x 14 空间。复位向量位于 0000h 处, 中断向量位于 0004h 处 (见图 2-2 和图 2-3)。

图 2-1: PIC16F882 程序存储器映射和堆栈

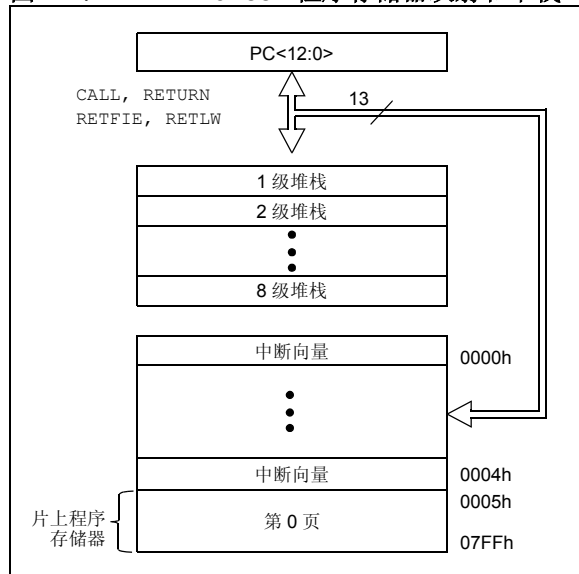


图 2-2: PIC16F883/PIC16F884 的程序存储器映射和堆栈

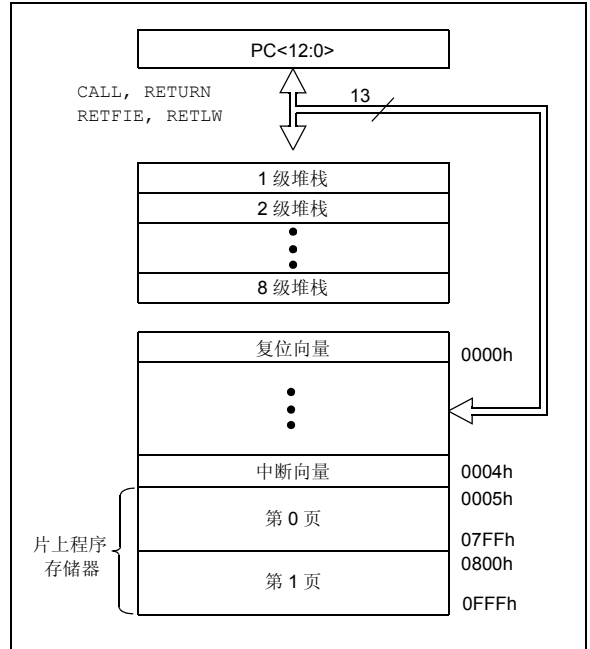
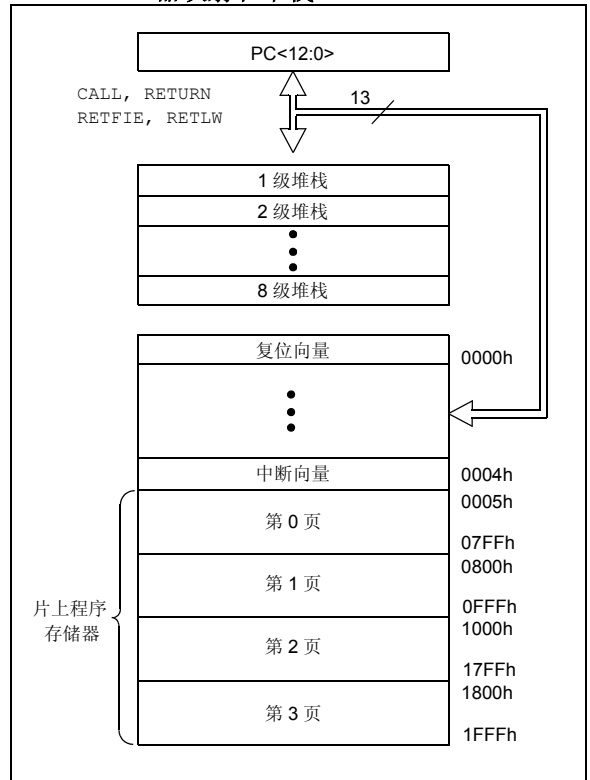


图 2-3: PIC16F886/PIC16F887 的程序存储器映射和堆栈



PIC16F882/883/884/886/887

2.2 数据存储器的构成

数据存储器（见图 2-2 和图 2-3）分为四个存储区（bank），包含通用寄存器（General Purpose Register, GPR）和特殊功能寄存器（Special Function Register, SFR）。特殊功能寄存器位于各个存储区的前 32 个单元。通用寄存器实现为静态 RAM，位于各个存储区的后 96 个单元。Bank 1 中的 F0h-FFh、Bank 2 中的 170h-17Fh 和 Bank 3 中的 1F0h-1FFh 寄存器单元指向 Bank 0 中的地址 70h-7Fh。在各存储区中实现的通用寄存器（GPR）的实际数量取决于具体器件。详细信息如图 2-5 和图 2-6 所示。所有其他 RAM 单元均未实现，读取时返回 0。STATUS 寄存器的 RP<1:0> 为存储区选择位：

RP1 RP0

- 0 0 → 选择 Bank 0
- 0 1 → 选择 Bank 1
- 1 0 → 选择 Bank 2
- 1 1 → 选择 Bank 3

2.2.1 通用文件寄存器

在 PIC16F882 中文件寄存器以 128 x 8 的形式构成，在 PIC16F883/PIC16F884 中以 256 x 8 的形式构成，而在 PIC16F886/PIC16F887 中以 368 x 8 的形式构成。通过文件选择寄存器（File Select Register, FSR）直接或间接访问每个寄存器（见第 2.4 节“间接寻址、INDF 和 FSR 寄存器”）。

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

2.2.2 特殊功能寄存器

特殊功能寄存器是 CPU 和外设模块用来控制器件进行所需操作的寄存器（见表 2-1）。这类寄存器均实现为静态 RAM。

特殊功能寄存器可分成两类：内核和外设 SFR。本节将介绍与内核功能相关的寄存器。与外设功能相关的寄存器将在相应外设功能的章节介绍。

PIC16F882/883/884/886/887

图 2-5: PIC16F883/PIC16F884 的特殊功能寄存器

文件地址	文件地址	文件地址	文件地址
间接寻址 ⁽¹⁾ 00h	间接寻址 ⁽¹⁾ 80h	间接寻址 ⁽¹⁾ 100h	间接寻址 ⁽¹⁾ 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h	WDTCON 105h	SRCON 185h
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h	CM1CON0 107h	BAUDCTL 187h
PORTD ⁽²⁾ 08h	TRISD ⁽²⁾ 88h	CM2CON0 108h	ANSEL 188h
PORTE 09h	TRISE 89h	CM2CON1 109h	ANSELH 189h
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 ⁽¹⁾ 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	保留 18Eh
TMR1H 0Fh	OSCCON 8Fh	EEADRH 10Fh	保留 18Fh
T1CON 10h	OSCTUNE 90h		
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPADDD 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h	WPUB 95h		
CCPR1H 16h	IOCB 96h		
CCP1CON 17h	VRCON 97h		
RCSTA 18h	TXSTA 98h		
TXREG 19h	SPBRG 99h		
RCREG 1Ah	SPBRGH 9Ah		
CCPR2L 1Bh	PWM1CON 9Bh		
CCPR2H 1Ch	ECCPAS 9Ch		
CCP2CON 1Dh	PSTRCON 9Dh		
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
	A0h		
通用寄存器 96 字节	通用寄存器 80 字节	通用寄存器 80 字节	
	快速操作存储区 70h-7Fh	快速操作存储区 70h-7Fh	快速操作存储区 70h-7Fh
7Fh	EFh	16Fh	1EFh
	F0h	170h	1F0h
	FFh	17Fh	1FFh
Bank 0	Bank 1	Bank 2	Bank 3

■ 未实现的数据存储单元，读为 0。

注 1: 非物理寄存器。
注 2: 仅 PIC16F884。

PIC16F882/883/884/886/887

图 2-6: PIC16F886/PIC16F887 的特殊功能寄存器

文件地址	文件地址	文件地址	文件地址
间接寻址 ⁽¹⁾ 00h	间接寻址 ⁽¹⁾ 80h	间接寻址 ⁽¹⁾ 100h	间接寻址 ⁽¹⁾ 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h	WDTCON 105h	SRCON 185h
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h	CM1CON0 107h	BAUDCTL 187h
PORTD ⁽²⁾ 08h	TRISD ⁽²⁾ 88h	CM2CON0 108h	ANSEL 188h
PORTE 09h	TRISE 89h	CM2CON1 109h	ANSELH 189h
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 ⁽¹⁾ 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	保留 18Eh
TMR1H 0Fh	OSCCON 8Fh	EEADRH 10Fh	保留 18Fh
T1CON 10h	OSCTUNE 90h		
TMR2 11h	SSPCON2 91h	通用寄存器 110h	通用寄存器 190h
T2CON 12h	PR2 92h	16 字节 111h	16 字节 191h
SSPBUF 13h	SSPADD 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h	WPUB 95h		
CCPR1H 16h	IOCB 96h		
CCP1CON 17h	VRCON 97h		
RCSTA 18h	TXSTA 98h		
TXREG 19h	SPBRG 99h		
RCREG 1Ah	SPBRGH 9Ah		
CCPR2L 1Bh	PWM1CON 9Bh		
CCPR2H 1Ch	ECCPAS 9Ch		
CCP2CON 1Dh	PSTRCON 9Dh		
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
通用寄存器 20h	通用寄存器 A0h	通用寄存器 120h	通用寄存器 1A0h
通用寄存器 3Fh	通用寄存器	通用寄存器	通用寄存器
通用寄存器 40h	80 字节	80 字节	80 字节
通用寄存器 6Fh			
快速操作存储区 70h-7Fh	快速操作存储区 70h-7Fh	快速操作存储区 70h-7Fh	快速操作存储区 70h-7Fh
7Fh	FFh	17Fh	1FFh
Bank 0	Bank 1	Bank 2	Bank 3

■ 未实现的数据存储单元，读为 0。

注 1: 非物理寄存器。
2: 仅 PIC16F887。

PIC16F882/883/884/886/887

表 2-1: PIC16F882/883/884/886/887 特殊功能寄存器汇总 (BANK 0)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	页
Bank 0											
00h	INDF	寻址该单元会使用 FSR 的内容寻址数据存储器 (不是物理寄存器)								xxxx xxxx	37,213
01h	TMR0	Timer0 模块寄存器								xxxx xxxx	73,213
02h	PCL	程序计数器的低字节								0000 0000	37,213
03h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	29,213
04h	FSR	间接数据存储器地址指针								xxxx xxxx	37,213
05h	PORTA ⁽³⁾	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx xxxx	39,213
06h	PORTB ⁽³⁾	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	48,213
07h	PORTC ⁽³⁾	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	53,213
08h	PORTD ^(3,4)	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	57,213
09h	PORTE ⁽³⁾	—	—	—	—	RE3	RE2 ⁽⁴⁾	RE1 ⁽⁴⁾	RE0 ⁽⁴⁾	---- xxxx	59,213
0Ah	PCLATH	—	—	—	程序计数器高 5 位的写缓冲器				---0 0000	37,213	
0Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF ⁽¹⁾	0000 000x	31,213
0Ch	PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	34,213
0Dh	PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	ULPWUIF	—	CCP2IF	0000 00-0	35,213
0Eh	TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								xxxx xxxx	76,213
0Fh	TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								xxxx xxxx	76,213
10h	T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	0000 0000	79,213
11h	TMR2	Timer2 模块寄存器								0000 0000	81,213
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	82,213
13h	SSPBUF	同步串行端口接收缓冲器 / 发送寄存器								xxxx xxxx	179,213
14h	SSPCON ⁽²⁾	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	177,213
15h	CCPR1L	捕捉 / 比较 / PWM 寄存器 1 的低字节 (LSB)								xxxx xxxx	126,213
16h	CCPR1H	捕捉 / 比较 / PWM 寄存器 1 的高字节 (MSB)								xxxx xxxx	126,213
17h	CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	124,213
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	159,213
19h	TXREG	EUSART 发送数据寄存器								0000 0000	151,213
1Ah	RCREG	EUSART 接收数据寄存器								0000 0000	156,213
1Bh	CCPR2L	捕捉 / 比较 / PWM 寄存器 2 的低字节 (LSB)								xxxx xxxx	126,213
1Ch	CCPR2H	捕捉 / 比较 / PWM 寄存器 2 的高字节 (MSB)								xxxx xxxx	126,213
1Dh	CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	125,214
1Eh	ADRESH	A/D 结果寄存器的高字节								xxxx xxxx	99,214
1Fh	ADCON0	ADCS1	ADCS0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	0000 0000	104,214

- 图注: — = 未实现单元 (读为 0), u = 不变, x = 未知, q = 取值视具体情况而定, 阴影 = 未实现
- 注:
- 1: MCLR 和 WDT 复位不会对数据锁存器中先前的值产生影响。RBIF 位将在复位时清零, 但如果存在不匹配, 它将再次置 1。
 - 2: 当 SSPCON 寄存器中的 SSPM<3:0> 位 = 1001 时, 任何对 SSPADD SFR 地址的读或写操作都通过 SSPMSK 寄存器进行。更多详细信息请参见寄存器 13-3 和 13-4。
 - 3: 具有模拟功能并由 ANSEL 和 ANSELH 寄存器控制的端口引脚将在复位后立即读为 0, 即使数据锁存器未定义 (POR) 或不变 (其他复位) 时也是如此。
 - 4: 仅 PIC16F884/PIC16F887。

PIC16F882/883/884/886/887

表 2-2: PIC16F882/883/884/886/887 特殊功能寄存器汇总 (BANK 1)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	页
Bank 1											
80h	INDF	寻址该地址单元会使用 FSR 的内容寻址数据存储器 (不是物理寄存器)								xxxx xxxx	37,213
81h	OPTION_REG	RBP0	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	29,214
82h	PCL	程序计数器 (PC) 的低字节								0000 0000	37,213
83h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	29,213
84h	FSR	间接数据存储器地址指针								xxxx xxxx	37,213
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	39,214
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	48,214
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	53,214
88h	TRISD ⁽³⁾	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	57,214
89h	TRISE	—	—	—	—	TRISE3	TRISE2 ⁽³⁾	TRISE1 ⁽³⁾	TRISE0 ⁽³⁾	---- 1111	59,214
8Ah	PCLATH	—	—	—	程序计数器高 5 位的写缓冲器				---0 0000	37,213	
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF ⁽¹⁾	0000 000x	31,213
8Ch	PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	32,214
8Dh	PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	ULPWUIE	—	CCP2IE	0000 00-0	33,214
8Eh	PCON	—	—	ULPWUE	SBOREN	—	—	\overline{POR}	\overline{BOR}	--01 --qq	36,214
8Fh	OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS	-110 q000	62,214
90h	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	66,214
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	177,214
92h	PR2	Timer2 周期寄存器								1111 1111	81,214
93h	SSPAD ⁽²⁾	同步串行端口 (I ² C 模式) 地址寄存器								0000 0000	185,214
93h	SSPMSK ⁽²⁾	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	1111 1111	185,214
94h	SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	185,214
95h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	1111 1111	49,214
96h	IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0	0000 0000	49,214
97h	VRCON	VREN	VROE	VRR	VRSS	VR3	VR2	VR1	VR0	0000 0000	97,214
98h	TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	158,214
99h	SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	161,214
9Ah	SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	161,214
9Bh	PWM1CON	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000	144,214
9Ch	ECCPAS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000	141,214
9Dh	PSTRCON	—	—	—	STRSYNC	STRD	STRC	STRB	STRA	---0 0001	145,214
9Eh	ADRESL	A/D 结果寄存器的低字节								xxxx xxxx	99,214
9Fh	ADCON1	ADFM	—	VCFG1	VCFG0	—	—	—	—	0-00 ----	105,214

- 图注:** — = 未实现单元 (读为 0), u = 不变, x = 未知, q = 取值视具体情况而定, 阴影 = 未实现
- 注:**
- 1: MCLR 和 WDT 复位不会对数据锁存器中先前的值产生影响。RBIF 位将在复位时清零, 但如果存在不匹配, 它将再次置 1。
 - 2: 仅当 SSPCON 寄存器的 SSPM<3:0> 位 = 1001 时才可访问。
 - 3: 仅 PIC16F884/PIC16F887。

PIC16F882/883/884/886/887

表 2-3: PIC16F882/883/884/886/887 特殊功能寄存器汇总 (BANK 2)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	页
Bank 2											
100h	INDF	寻址该地址单元会使用 FSR 的内容寻址数据存储器 (不是物理寄存器)								xxxx xxxx	37,213
101h	TMR0	Timer0 模块寄存器								xxxx xxxx	73,213
102h	PCL	程序计数器 (PC) 的低字节								0000 0000	37,213
103h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	29,213
104h	FSR	间接数据存储器地址指针								xxxx xxxx	37,213
105h	WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 1000	221,214
106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	48,213
107h	CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0	0000 -000	88,214
108h	CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0	0000 -000	89,214
109h	CM2CON1	MC1OUT	MC2OUT	C1RSEL	C2RSEL	—	—	T1GSS	C2SYNC	0000 --10	91,214
10Ah	PCLATH	—	—	—	程序计数器高 5 位的写缓冲器				---	0000	37,213
10Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF ⁽¹⁾	0000 000x	31,213
10Ch	EEDATA	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0	0000 0000	112,214
10Dh	EEADR	EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	0000 0000	112,215
10Eh	EEDATH	—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0	--00 0000	112,215
10Fh	EEADRH	—	—	—	EEADRH4 ⁽²⁾	EEADRH3	EEADRH2	EEADRH1	EEADRH0	---- 0000	112,215

图注: — = 未实现单元 (读为 0), u = 不变, x = 未知, q = 取值视条件而定, 阴影 = 未实现
 注 1: MCLR 和 WDT 复位不会对数据存储器中先前的值产生影响。RBIF 位将在复位时清零, 但如果存在不匹配, 它将再次置 1。
 2: 仅 PIC16F886/PIC16F887。

表 2-4: PIC16F882/883/884/886/887 特殊功能寄存器汇总 (BANK 3)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	页
Bank 3											
180h	INDF	寻址该地址单元会使用 FSR 的内容寻址数据存储器 (不是物理寄存器)								xxxx xxxx	37,213
181h	OPTION_REG	\overline{RBPU}	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	30,214
182h	PCL	程序计数器 (PC) 的低字节								0000 0000	37,213
183h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	29,213
184h	FSR	间接数据存储器地址指针								xxxx xxxx	37,213
185h	SRCON	SR1	SR0	C1SEN	C2REN	PULSS	PULSR	—	FVREN	0000 00-0	93,215
186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	48,214
187h	BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	160,215
188h	ANSEL	ANS7 ⁽²⁾	ANS6 ⁽²⁾	ANS5 ⁽²⁾	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	40,215
189h	ANSELH	—	—	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8	--11 1111	99,215
18Ah	PCLATH	—	—	—	程序计数器高 5 位的写缓冲器				---	0000	37,213
18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF ⁽¹⁾	0000 000x	31,213
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	113,215
18Dh	EECON2	EEPROM 控制寄存器 2 (不是物理寄存器)								---- ----	111,215

图注: — = 未实现单元 (读为 0), u = 不变, x = 未知, q = 取值视具体情况而定, 阴影 = 未实现
 注 1: MCLR 和 WDT 复位不会对数据存储器中先前的值产生影响。RBIF 位将在复位时清零, 但如果存在不匹配, 它将再次置 1。
 2: 仅 PIC16F886/PIC16F887。

PIC16F882/883/884/886/887

2.2.2.1 状态寄存器

STATUS 寄存器如寄存器 2-1 所示，包含：

- ALU 的算术状态
- 复位状态
- 数据存储区（GPR 和 SFR）的存储区选择位

与其他寄存器一样，STATUS 寄存器可以是任何指令的目标寄存器。如果一条影响 Z、DC 或 C 位的指令以 STATUS 寄存器作为目标寄存器，则不能写这 3 个状态位。这些位根据器件逻辑被置 1 或清零。而且也不能写 TO 和 PD 位。因此将 STATUS 作为目标寄存器的指令可能无法得到预期的结果。

例如，CLRF STATUS 会清零高 3 位，并将 Z 位置 1。这样 STATUS 的值为 000u u1uu（其中 u = 不变）。

因此，建议仅使用 BCF、BSF、SWAPF 和 MOVWF 指令来改变 STATUS 寄存器，因为这些指令不会影响任何状态位。对于其他不影响状态位的指令，请参见第 15.0 节“指令集综述”。

注 1: 在减法运算中，C 和 DC 分别作为借位（Borrow）和半借位（Digit borrow）位。

寄存器 2-1: STATUS: 状态寄存器

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 ⁽¹⁾	C ⁽¹⁾
bit 7							bit 0

图注:

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

- bit 7 **IRP:** 寄存器存储区选择位（用于间接寻址）
 1 = Bank 2 和 Bank 3（100h-1FFh）
 0 = Bank 0 和 Bank 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 = Bank 0（00h-7Fh）
 01 = Bank 1（80h-FFh）
 10 = Bank 2（100h-17Fh）
 11 = Bank 3（180h-1FFh）
- bit 4 **TO:** 超时位
 1 = 上电或是执行了 CLRWDT 指令或 SLEEP 指令
 0 = 发生了 WDT 超时
- bit 3 **PD:** 掉电位
 1 = 上电或执行了 CLRWDT 指令
 0 = 执行了 SLEEP 指令
- bit 2 **Z:** 结果为零位
 1 = 算术或逻辑运算的结果为零
 0 = 算术或逻辑运算的结果不为零
- bit 1 **DC:** 半进位 / 借位位（ADDWF、ADDLW、SUBLW 或 SUBWF 指令）⁽¹⁾
 1 = 发生了结果的第 4 低位向高位进位
 0 = 结果的第 4 低位没有向高位进位
- bit 0 **C:** 进位 / 借位位（ADDWF、ADDLW、SUBLW 或 SUBWF 指令）⁽¹⁾
 1 = 结果的最高位发生了进位
 0 = 结果的最高位没有发生进位

注 1: 对于借位，极性是相反的。减法是通过加上第二个操作数的二进制补码实现的。对于移位指令（RRF 或 RLF），此位的值来自源寄存器的最高位或最低位。

PIC16F882/883/884/886/887

2.2.2.2 OPTION 寄存器

如寄存器2-2所示，OPTION寄存器是可读写的寄存器，包括各种控制位用于配置：

- Timer0/WDT 预分频器
- 外部 INT 中断
- Timer0
- PORTB 上的弱上拉

注： 要使 Timer0 获取 1:1 的预分频比配置，可通过将选项寄存器的 PSA 位置 1 将预分频器分配给 WDT。请参见第 6.3 节“Timer1 预分频器”。

寄存器 2-2: OPTION_REG: 选项寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBP}}\text{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

图注：

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

- bit 7 **$\overline{\text{RBP}}\text{U}$** : PORTB 上拉使能位
 1 = 禁止 PORTB 上拉
 0 = 由端口的各个锁存值使能 PORTB 上拉
- bit 6 **INTEDG**: 触发中断的边沿选择位
 1 = INT 引脚上升沿触发中断
 0 = INT 引脚下降沿触发中断
- bit 5 **T0CS**: Timer0 时钟源选择位
 1 = T0CKI 引脚上的跳变沿
 0 = 内部指令周期时钟 (Fosc/4)
- bit 4 **T0SE**: Timer0 时钟源边沿选择位
 1 = 在 T0CKI 引脚信号从高电平跳变到低电平时递增
 0 = 在 T0CKI 引脚信号从低电平跳变到高电平时递增
- bit 3 **PSA**: 预分频器分配位
 1 = 预分频器分配给 WDT
 0 = 预分频器分配给 Timer0 模块
- bit 2-0 **PS<2:0>**: 预分频比选择位

位值	Timer0 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

PIC16F882/883/884/886/887

2.2.2.3 IINTCON 寄存器

如寄存器 2-3 所示，INTCON 寄存器是可读写的寄存器，包含 TMR0 寄存器溢出、PORTB 端口变化和外部 INT 引脚中断等的允许和标志位。

注： 当有中断条件产生时，无论对应的中断允许位或（INTCON 寄存器中的）全局允许位 GIE 的状态如何，中断标志位都将置 1。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 2-3: INTCON: 中断控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TOIE	INTE	RBIE ^(1,3)	T0IF ⁽²⁾	INTF	RBIF
bit 7							bit 0

图注：

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

- bit 7 **GIE:** 全局中断允许位
1 = 允许所有未被屏蔽的中断
0 = 禁止所有中断
- bit 6 **PEIE:** 外设中断允许位
1 = 允许所有未被屏蔽的外设中断
0 = 禁止所有外设中断
- bit 5 **TOIE:** Timer0 溢出中断允许位
1 = 允许 Timer0 中断
0 = 禁止 Timer0 中断
- bit 4 **INTE:** INT 外部中断允许位
1 = 允许 INT 外部中断
0 = 禁止 INT 外部中断
- bit 3 **RBIE:** PORTB 电平变化中断允许位
1 = 允许 PORTB 电平变化中断
0 = 禁止 PORTB 电平变化中断
- bit 2 **T0IF:** Timer0 溢出中断标志位 ⁽²⁾
1 = TMR0 寄存器已经溢出（必须由软件清零）
0 = TMR0 寄存器未发生溢出
- bit 1 **INTF:** INT 外部中断标志位
1 = 发生 INT 外部中断（必须由软件清零）
0 = 未发生 INT 外部中断
- bit 0 **RBIF:** PORTB 电平变化中断标志位
1 = PORB 端口中至少有一个引脚的电平状态发生了改变（必须由软件清零）
0 = 没有一个 PORTB 通用 I/O 引脚的状态发生了改变

- 注**
- 1: IOCB 寄存器也必须使能。
 - 2: T0IF 位在 Timer0 计满归 0 时置 1。复位不会使 Timer0 发生改变，应在将 T0IF 位清零前对其进行初始化。
 - 3: 包括 ULPWU 中断。

PIC16F882/883/884/886/887

2.2.2.5 PIE2 寄存器

PIE2 寄存器包含各中断允许位，如寄存器 2-5 所示。

注： 在允许任何外设中断前，必须先将 INTCON 寄存器的 PEIE 位置 1。

寄存器 2-5: PIE2: 外设中断允许寄存器 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
OSFIE	C2IE	C1IE	EEIE	BCLIE	ULPWUIE	—	CCP2IE
bit 7						bit 0	

图注：

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

- bit 7 **OSFIE:** 振荡器故障中断允许位
1 = 允许振荡器故障中断
0 = 禁止振荡器故障中断
- bit 6 **C2IE:** 比较器 C2 中断允许位
1 = 允许比较器 C2 中断
0 = 禁止比较器 C2 中断
- bit 5 **C1IE:** 比较器 C1 中断允许位
1 = 允许比较器 C1 中断
0 = 禁止比较器 C1 中断
- bit 4 **EEIE:** EEPROM 写操作中断允许位
1 = 允许 EEPROM 写操作中断
0 = 禁止 EEPROM 写操作中断
- bit 3 **BCLIE:** 总线冲突中断允许位
1 = 允许总线冲突中断
0 = 禁止总线冲突中断
- bit 2 **ULPWUIE:** 超低功耗唤醒中断允许位
1 = 允许超低功耗唤醒中断
0 = 禁止超低功耗唤醒中断
- bit 1 **未实现:** 读为 0
- bit 0 **CCP2IE:** CCP2 中断允许位
1 = 允许 CCP2 中断
0 = 禁止 CCP2 中断

PIC16F882/883/884/886/887

2.2.2.7 PIR2 寄存器

PIR2 寄存器包括各中断标志位，如寄存器 2-7 所示。

注： 当有中断条件产生时，无论对应的中断允许位或（INTCON 寄存器中的）全局允许位 GIE 的状态如何，中断标志位都将置 1。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 2-7: PIR2: 外设中断请求寄存器 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
OSFIF	C2IF	C1IF	EEIF	BCLIF	ULPWUIF	—	CCP2IF
bit 7						bit 0	

图注：

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

- bit 7 **OSFIF:** 振荡器故障中断标志位
 1 = 系统振荡器发生故障，时钟输入更改为 INTOSC（必须由软件清零）
 0 = 系统时钟正常工作
- bit 6 **C2IF:** 比较器 C2 中断标志位
 1 = 比较器输出（C2OUT 位）发生了改变（必须由软件清零）
 0 = 比较器输出（C2OUT 位）不变
- bit 5 **C1IF:** 比较器 C1 中断标志位
 1 = 比较器输出（C1OUT 位）发生了改变（必须由软件清零）
 0 = 比较器输出（C1OUT 位）不变
- bit 4 **EEIF:** EE 写操作中断标志位
 1 = 写操作完成（必须由软件清零）
 0 = 写操作未完成或尚未启动
- bit 3 **BCLIF:** 总线冲突中断标志位
 1 = 当配置为 I²C 主控模式时，MSSP 中发生了总线冲突
 0 = 未发生总线冲突
- bit 2 **ULPWUIF:** 超低功耗唤醒中断标志位
 1 = 产生了唤醒条件（必须由软件清零）
 0 = 未产生唤醒条件
- bit 1 **未实现:** 读为 0
- bit 0 **CCP2IF:** CCP2 中断标志位
捕捉模式:
 1 = 发生了 TMR1 寄存器的捕捉（必须由软件清零）
 0 = 未发生 TMR1 寄存器的捕捉
比较模式:
 1 = 发生了 TMR1 寄存器的比较匹配（必须由软件清零）
 0 = 未发生 TMR1 寄存器的比较匹配
PWM 模式:
 在此模式下未用

PIC16F882/883/884/886/887

2.2.2.8 PCON 寄存器

电源控制 (PCON) 寄存器 (见寄存器 2-8) 包含区别以下内容的标志位:

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

PCON 寄存器还控制超低功耗唤醒和软件使能 $\overline{\text{BOR}}$ 。

寄存器 2-8: PCON: 电源控制寄存器

U-0	U-0	R/W-0	R/W-1	U-0	U-0	R/W-0	R/W-x
—	—	ULPWUE	SBOREN ⁽¹⁾	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7						bit 0	

图注:

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

- bit 7-6 **未实现:** 读为 0
- bit 5 **ULPWUE:** 超低功耗唤醒使能位
1 = 使能超低功能唤醒
0 = 禁止超低功能唤醒
- bit 4 **SBOREN:** 软件 BOR 使能位 ⁽¹⁾
1 = BOR 使能
0 = BOR 禁止
- bit 3-2 **未实现:** 读为 0
- bit 1 **$\overline{\text{POR}}$:** 上电复位状态位
1 = 未发生上电复位
0 = 已发生上电复位 (必须在发生上电复位后用软件置 1)
- bit 0 **$\overline{\text{BOR}}$:** 欠压复位状态位
1 = 未发生欠压复位
0 = 已发生欠压复位 (必须在发生欠压复位后用软件置 1)

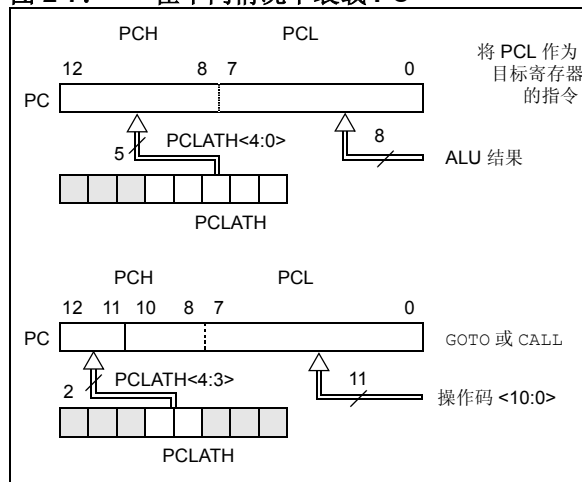
注 1: 当配置字寄存器 1 中的 BOREN<1:0> = 01 时, 该位用来控制 $\overline{\text{BOR}}$ 。

PIC16F882/883/884/886/887

2.3 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, 但可通过 PCLATH 寄存器间接进行。任何复位时都将清零 PC 的高位。图 2-7 给出了装载 PC 的两种情况。图中上方的示例给出了在写 PCL (PCLATH<4:0> → PCH) 时装载 PC 的方式。图中下方的示例给出了在执行 CALL 或 GOTO 指令期间装载 PC (PCLATH<4:3> → PCH) 的方式。

图 2-7: 在不同情况下装载 PC



2.3.1 修改 PCL

执行任何将 PCL 寄存器作为目标寄存器的指令会导致程序计数器的 PC<12:8> 位 (PCH) 被 PCLATH 寄存器的内容替代, 从而允许通过将所需的高 5 位写入 PCLATH 寄存器来更改整个程序计数器。当将低 8 位写入 PCL 寄存器时, 程序计数器的全部 13 位将更改为 PCLATH 寄存器所包含的值和正在写入到 PCL 寄存器的值。

通过向程序计数器添加一个偏移量 (ADDWF PCL) 实现计算 GOTO。通过修改 PCL 寄存器跳转到查找表或程序跳转表 (计算 GOTO) 时需要小心。假设将 PCLATH 设置为表的起始地址, 如果表长度大于 255 条指令或存储器地址的低 8 位在表的中间从 0xFF 溢出到 0x00, 则当存储器地址从表起始地址变化至表内目标单元时, 每发生一次溢出 PCLATH 都必须递增 1。

欲知更多信息请参见应用笔记 AN556, “Implementing a Table Read” (DS00556)

2.3.2 堆栈

PIC16F882/883/884/886/887 具有一个 8 级深 x 13 位宽的硬件堆栈 (见图 2-2 和图 2-3)。该堆栈既不占用程序存储空间也不占用数据存储空间, 并且栈指针不能读写。当执行 CALL 指令或由于中断导致程序跳转时, PC 的值会被压入 (PUSH) 堆栈。当执行 RETURN、RETLW 或 RETFIE 指令时, PC 值从堆栈弹出 (POP)。PCLATH 的值不受压栈或出栈操作的影响。

堆栈的工作如循环缓冲器。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: 不存在用于表示堆栈上溢或堆栈下溢条件的状态位。
- 注 2: 不存在称为 PUSH 或 POP 的指令或助记符。这两个操作是在执行 CALL、RETURN、RETLW 和 RETFIE 指令或跳转到中断向量地址时发生的。

2.4 间接寻址、INDF 和 FSR 寄存器

INDF 寄存器不是物理寄存器。对 INDF 寄存器进行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文件选择寄存器 (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 位地址是通过连接 8 位 FSR 寄存器和 STATUS 寄存器中的 IRP 位获得的, 如图 2-8 所示。

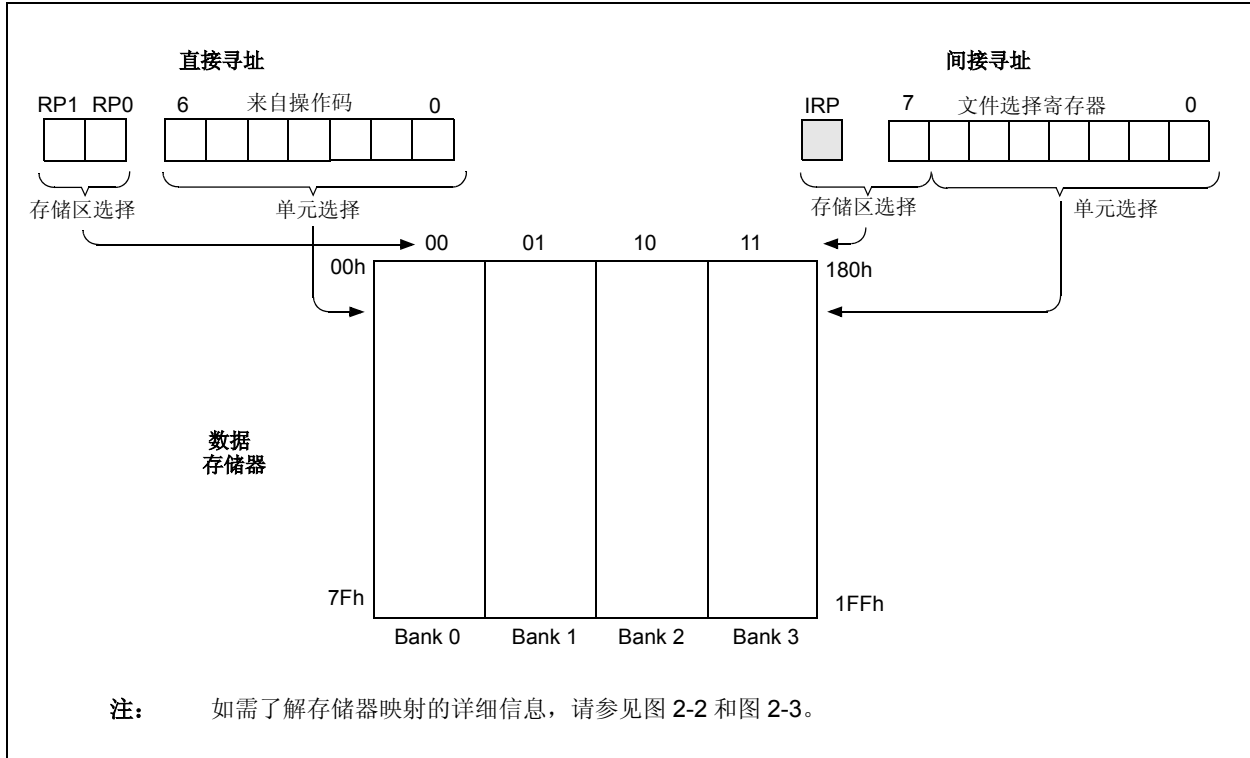
例 2-1 给出了使用间接寻址清零 RAM 地址单元 20h-2Fh 的简单程序。

例 2-1: 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
CONTINUE ;yes continue
```

PIC16F882/883/884/886/887

图 2-8: 直接 / 间接寻址 PIC16F882/883/884/886/887



PIC16F882/883/884/886/887

3.2 该引脚的其他功能

RA0 还具有超低功耗唤醒选项。以下三节将说明这些功能。

3.2.1 ANSEL 寄存器

ANSEL 寄存器（寄存器 3-3）用于将 I/O 引脚的输入模式配置为模拟模式。将 ANSEL 中适当的位置 1 将导致对相应引脚的所有数字读操作返回 0，并使引脚的模拟功能正常工作。

ANSEL 位的状态对数字输出功能没有影响。TRIS 清零且 ANSEL 置 1 的引脚仍作为数字输出，但输入模式将成为模拟模式。这会导致在受影响的端口上执行读—修改—写操作时产生不可预计的结果。

寄存器 3-3: ANSEL: 模拟选择寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANS7 ⁽²⁾	ANS6 ⁽²⁾	ANS5 ⁽²⁾	ANS4	ANS3	ANS2	ANS1	ANS0
bit 7							bit 0

图注:

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

bit 7-0 **ANS<7:0>**: 模拟选择位
分别选择引脚 AN<7:0> 的模拟或数字功能。
1 = 模拟输入。引脚被分配为模拟输入 ⁽¹⁾。
0 = 数字 I/O。引脚被分配给端口或特殊功能。

- 注 1:** 将引脚设置为模拟输入将自动禁止数字输入电路、弱上拉电路和电平变化中断（如果有的话）。相应的 TRIS 位必须置 1 以将引脚设置为输入模式，从而允许从外部控制引脚电压。
- 注 2:** 在 PIC16F883/886 上未实现。

PIC16F882/883/884/886/887

3.2.2 超低功耗唤醒

RA0 上的超低功耗唤醒 (ULPWU) 允许缓慢下降的电压在 RA0 上的电平发生变化时产生中断而不消耗过多的电流。通过将 PCON 寄存器中的 ULPWUE 位置 1 选择此模式。这将产生一个小的灌电流 (current sink)，可用于将 RA0 上的电容放电。

执行以下步骤以使用这一功能：

- 通过将 RA0 引脚配置为输出 (= 1) 给 RA0 上的电容充电。
- 将 RA0 配置为输入。
- 允许 RA0 上的电平变化中断。
- 将 PCON 寄存器中的 ULPWUE 位置 1 开始给电容放电。
- 执行 SLEEP 指令。

当 RA0 上的电压下降到 V_{IL} 以下时，将产生中断使器件唤醒并执行下一条指令。如果 INTCON 寄存器中的 GIE 位置 1，器件将调用中断向量 (0004h)。更多信息请参见第 3.4.3 节“电平变化中断”。

此功能提供了一种周期性地将器件从休眠状态唤醒的低功耗技术。超时时间取决于对 RA0 上的 RC 电路放电的时间。有关如何初始化超低功耗唤醒模块，请参见例 3-2。

在 RA0 和外部电容间串联一个电阻可为 RA0/AN0/ULPWU/C12IN0 引脚提供过电流保护并允许通过软件对超时进行校准 (见图 3-1)。可使用一个定时器测量电容的充电时间和放电时间。随后可调节充电时间以提供所需的中断延时。此技术可补偿温度、电压和元件精度造成的影响。超低功耗唤醒外设也可被配置为简单的可编程低电压检测或温度传感器。

注： 更多信息，请参考应用笔记 AN879 (DS00879)。

例 3-2: 超低功耗唤醒初始化

```
BANKSEL PORTA          ;
BSF    PORTA,0         ;Set RA0 data latch
BANKSEL ANSEL          ;
BCF    ANSEL,0         ;RA0 to digital I/O
BANKSEL TRISA          ;
BCF    TRISA,0         ;Output high to
CALL   CapDelay        ;charge capacitor
BANKSEL PIR2           ;
BCF    PIR2,ULPWUIF    ;Clear flag
BSF    PCON,ULPWUE     ;Enable ULP Wake-up
BSF    IOCB,0          ;Select RA0 IOC
BSF    TRISA,0         ;RA0 to input
MOVLW  B'10001000'     ;Enable interrupt
MOVWF  INTCON          ;and clear flag
SLEEP                    ;Wait for IOC
NOP                      ;
```

PIC16F882/883/884/886/887

3.2.3 引脚说明和框图

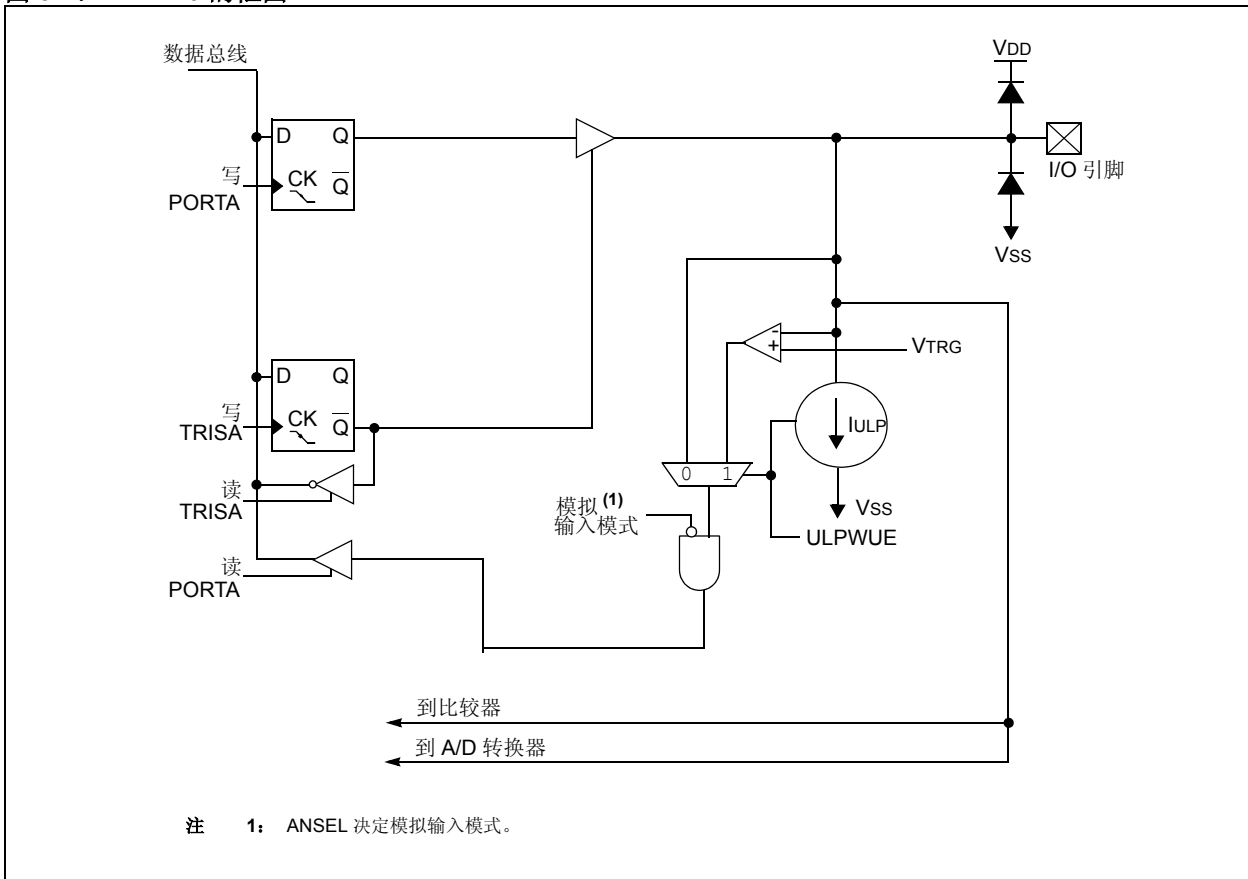
每个 PORTA 引脚都与其他功能复用。这里将简要地描述这些引脚和与它们复用的功能。欲知有关各个功能（如比较器和 A/D 转换器）的具体信息，请参见此数据手册中的相应章节。

3.2.3.1 RA0/AN0/ULPWU/C12IN0-

图 3-1 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- ADC 的一个模拟输入通道
- 比较器 C1 或 C2 的负模拟输入
- 用于超低功耗唤醒的模拟输入

图 3-1: RA0 的框图



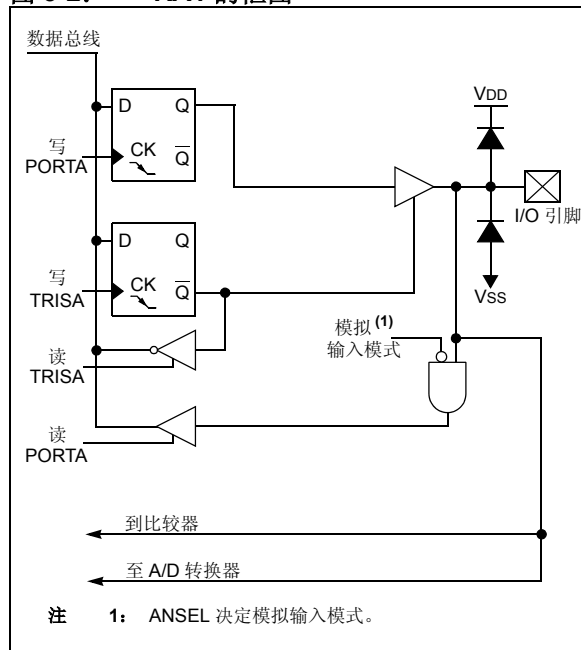
PIC16F882/883/884/886/887

3.2.3.2 RA1/AN1/C12IN1-

图 3-2 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- ADC 的模拟输入
- 比较器 C1 或 C2 的负模拟输入

图 3-2: RA1 的框图

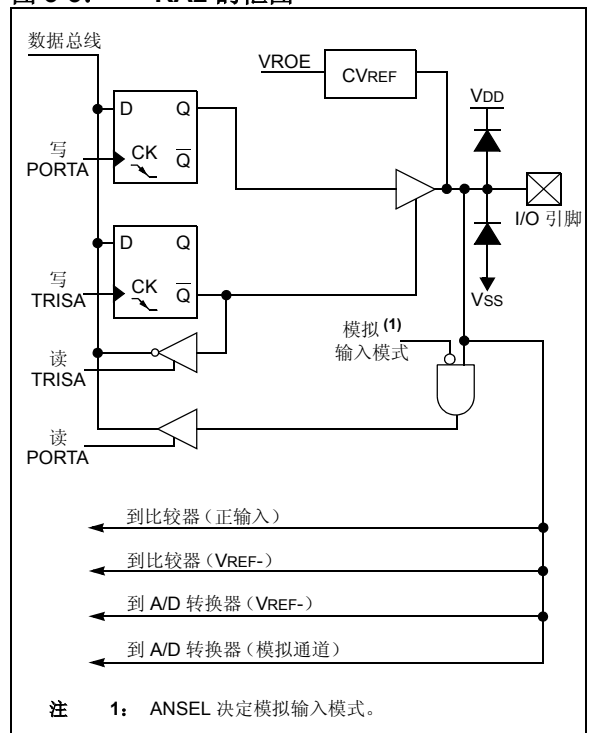


3.2.3.3 RA2/AN2/VREF-/CVREF/C2IN+

图 3-3 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- ADC 的模拟输入
- ADC 和 CVREF 的负参考电压输入
- 比较器的参考电压输出
- 比较器 C2 的正模拟输入

图 3-3: RA2 的框图



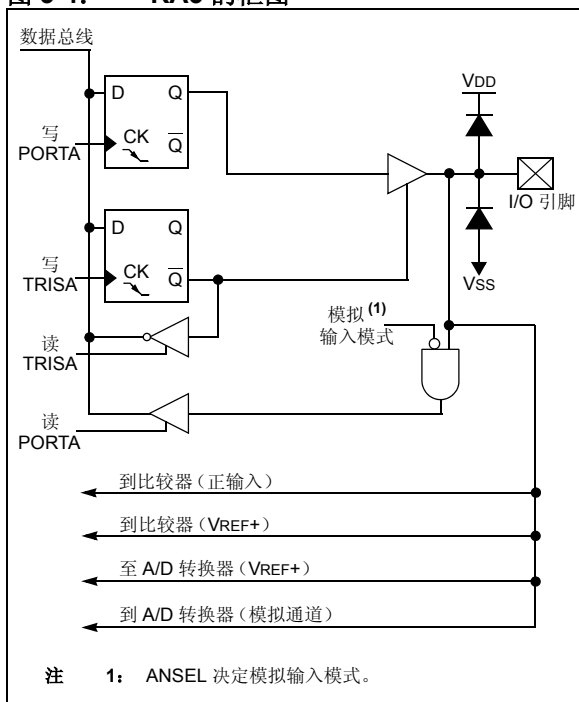
PIC16F882/883/884/886/887

3.2.3.4 RA3/AN3/VREF+/C1IN+

图 3-4 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用输入
- ADC 的模拟输入
- ADC 和 CVREF 的正参考电压输入
- 比较器 C1 的正模拟输入

图 3-4: RA3 的框图

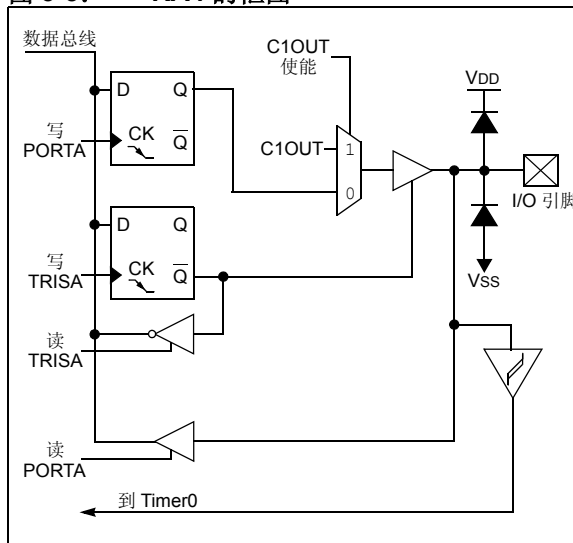


3.2.3.5 RA4/T0CKI/C1OUT

图 3-5 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- Timer0 的时钟输入
- 比较器 C1 的数字输出

图 3-5: RA4 的框图



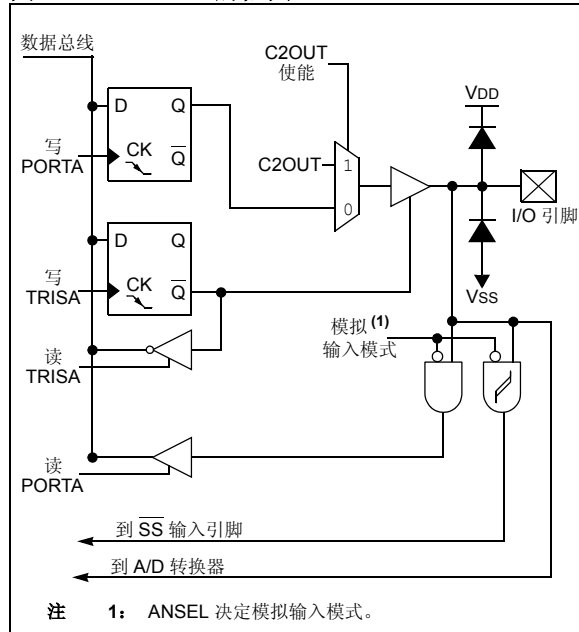
PIC16F882/883/884/886/887

3.2.3.6 RA5/AN4/ \overline{SS} /C2OUT

图 3-6 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- ADC 的模拟输入
- 从动选择输入
- 比较器 C2 的数字输出

图 3-6: RA5 的框图

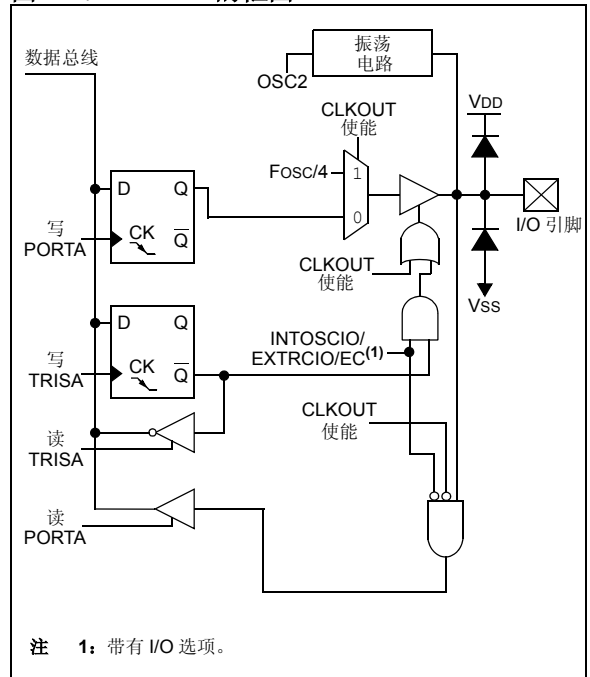


3.2.3.7 RA6/OSC2/CLKOUT

图 3-7 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- 晶振 / 谐振器连接
- 时钟输出

图 3-7: RA6 的框图



PIC16F882/883/884/886/887

3.2.3.8 RA7/OSC1/CLKIN

图 3-8 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- 晶振 / 谐振器连接
- 时钟输入

图 3-8: RA7 的框图

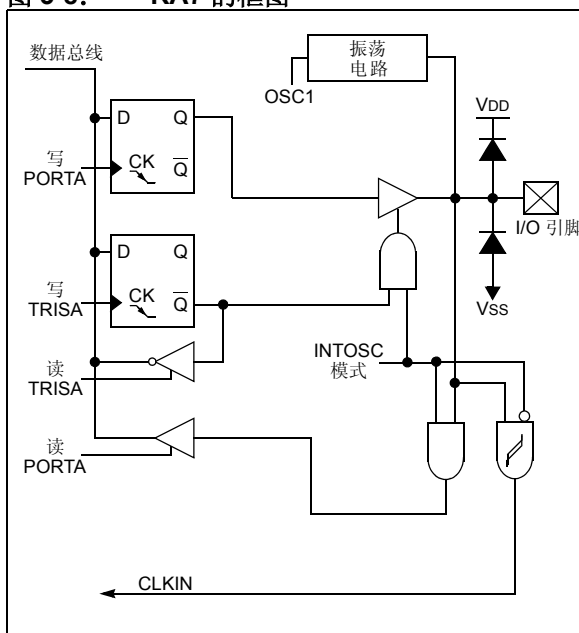


表 3-1: 与 PORTA 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
ADCON0	ADCS1	ADCS0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	0000 0000	0000 0000
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0	0000 -000	0000 -000
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0	0000 -000	0000 -000
CM2CON1	MC1OUT	MC2OUT	C1RSEL	C2RSEL	—	—	T1GSS	C2SYNC	0000 --10	0000 --10
PCON	—	—	ULPWUE	SBOREN	—	—	POR	BOR	--01 --qq	--0u --uu
OPTION_REG	RBPV	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx xxxx	uuuu uuuu
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111

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

PIC16F882/883/884/886/887

3.3 PORTB 和 TRISB 寄存器

PORTB 是一个 8 位宽的双向端口。对应的数据方向寄存器为 TRISB（寄存器 3-6）。将 TRISB 中的某个位置 1（= 1）可以使对应的 PORTB 引脚作为输入引脚（即将相应的输出驱动器置于高阻态）。将 TRISB 中的某个位清零（= 0）将使对应的 PORTB 引脚作为输出引脚（即将输出锁存器的内容从选定引脚输出）。例 3-3 给出了初始化 PORTB 的过程。

读 PORTB 寄存器（寄存器 3-5）读的是引脚的状态而写该寄存器将会写入端口锁存器。所有写操作都是读—修改—写操作。因此，写一个端口就意味着先读该端口的引脚电平，修改读到的值，然后再将改好的值写入端口数据锁存器。

即使在 PORTB 引脚用作模拟输入时，TRISB 寄存器（寄存器 3-6）仍然控制 PORTB 引脚的方向。当将 PORTB 引脚用作模拟输入时，用户必须确保 TRISB 寄存器中的位保持为置 1 状态。配置为模拟输入的 I/O 引脚总是读为 0。例 3-3 给出了初始化 PORTB 的过程。

例 3-3: 初始化 PORTB

```
BANKSEL PORTB      ;
CLRF   PORTB       ;Init PORTB
BANKSEL TRISB      ;
MOVLW  B'11110000';Set RB<7:4> as inputs
                                ;and RB<3:0> as outputs
MOVWF  TRISB       ;
```

注： 必须初始化 ANSELH 寄存器以将模拟通道配置为数字输入。配置为模拟输入的引脚将读为 0。

3.4 PORTB 引脚的其他功能

该系列器件的 PORTB 引脚 RB<7:0> 具有电平变化中断选项和弱上拉选项。以下三节将说明 PORTB 引脚的这些功能。

该系列器件中的每个 PORTB 引脚都具有电平变化中断选项和弱上拉选项。

3.4.1 ANSELH 寄存器

ANSELH 寄存器（寄存器 3-4）用于将 I/O 引脚的输入模式配置为模拟模式。将 ANSELH 中适当的位置 1 将导致对相应引脚的所有数字读操作返回 0，并使引脚的模拟功能正常工作。

ANSELH 位的状态对数字输出功能没有影响。TRIS 清零且 ANSELH 置 1 的引脚仍作为数字输出，但输入模式将成为模拟模式。这会导致在受影响的端口上执行读—修改—写操作产生不可预计的结果。

3.4.2 弱上拉

每个 PORTB 引脚都有可单独配置的内部弱上拉。控制位 WPUB<7:0> 使能或禁止每个弱上拉（见寄存器 3-7）。当将端口引脚配置为输出时，其弱上拉会自动切断。在上电复位时，弱上拉由 OPTION_REG 寄存器的 RBPU 位禁止。

3.4.3 电平变化中断

所有的 PORTB 引脚都可以被单独配置为电平变化中断引脚。控制位 IOCB<7:4> 允许或禁止每个引脚的该中断功能。请参见寄存器 3-8。上电复位时禁止引脚的电平变化中断功能。

对于已允许电平变化中断的引脚，则将该引脚上的值与上次读 PORTB 时锁存的旧值进行比较。将与上次读操作“不匹配”的输出一起进行逻辑或运算，以将 INTCON 寄存器中的 PORTB 电平变化中断标志位（RBIF）置 1。

该中断可将器件从休眠中唤醒。用户可在中断服务程序中通过以下方式清除中断：

- 对 PORTB 进行读或写操作。这将结束引脚电平的不匹配状态。
- 将标志位 RBIF 清零。

不匹配状态会不断将 RBIF 标志位置 1。而读或写 PORTB 将结束不匹配状态，并且允许将 RBIF 标志位清零。锁存器将保持最后一次读取的值不受 MCLR 和欠压复位的影响。在这些复位之后，如果不匹配仍然存在，RBIF 标志位将继续置 1。

注： 如果在执行读取操作时（Q2 周期的开始）I/O 引脚的电平发生变化，则 RBIF 中断标志位不会被置 1。此外，由于对端口的读或写影响到该端口的所有位，所以在电平变化中断模式下使用多个引脚的时候必须特别小心。在处理一个引脚电平变化的时候可能不会注意到另一个引脚上的电平变化。

PIC16F882/883/884/886/887

寄存器 3-4: ANSELH: 模拟选择寄存器的高字节

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8
bit 7							bit 0

图注:

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

bit 7-6 **未实现:** 读为 0
 bit 5-0 **ANS<13:8>:** 模拟选择位
 分别选择引脚 **AN<13:8>** 的模拟或数字功能。
 1 = 模拟输入。引脚被分配为模拟输入 **(1)**。
 0 = 数字 I/O。引脚被分配给端口或特殊功能。

注 1: 将引脚设置为模拟输入将自动禁止数字输入电路、弱上拉电路和电平变化中断（如果有的话）。相应的 TRIS 位必须置 1 以将引脚设置为输入模式，从而允许从外部控制引脚电压。

寄存器 3-5: PORTB: PORTB 寄存器

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
bit 7							bit 0

图注:

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

bit 7-0 **RA<7:0>:** PORTB I/O 引脚位
 1 = 端口引脚电平 > V_{IH}
 0 = 端口引脚电平 < V_{IL}

寄存器 3-6: TRISB: PORTB 三态寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
bit 7							bit 0

图注:

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

bit 7-0 **TRISB<7:0>:** PORTB 三态控制位
 1 = PORTB 引脚被配置为输入（三态）
 0 = PORTB 引脚被配置为输出

PIC16F882/883/884/886/887

寄存器 3-7: WPUB: 弱上拉 PORTB 寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0
bit 7							bit 0

图注:

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

bit 7-0 **WPUB<7:0>**: 弱上拉寄存器位
 1 = 使能上拉
 0 = 禁止上拉

- 注 1: 要单独使能任一个上拉, OPTION 寄存器的全局 $\overline{\text{RBPU}}$ 位必须清零。
 2: 如果引脚被配置为输出, 将自动禁止弱上拉。

寄存器 3-8: IOCB: PORTB 电平变化中断寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0
bit 7							bit 0

图注:

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

bit 7-0 **IOCB<7:0>**: PORTB 的电平变化中断控制位
 1 = 允许电平变化中断
 0 = 禁止电平变化中断

PIC16F882/883/884/886/887

3.4.4 引脚说明和框图

每个 PORTB 引脚都与其他功能复用。这里将简要地描述这些引脚和与它们复用的功能。欲知有关各个功能（如 SSP、I²C 或中断）的具体信息，请参见此数据手册中的相应章节。

3.4.4.1 RB0/AN12/INT

图 3-9 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- ADC 的模拟输入
- 由外部边沿触发的中断

3.4.4.2 RB1/AN10/P1C⁽¹⁾/C12IN3-

图 3-9 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- ADC 的模拟输入
- PWM 输出 ⁽¹⁾
- 比较器 C1 或 C2 的模拟输入

注 1： 只有 PIC16F882/883/886 上有 P1C。

3.4.4.3 RB2/AN8/P1B⁽¹⁾

图 3-9 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- ADC 的模拟输入
- PWM 输出 ⁽¹⁾

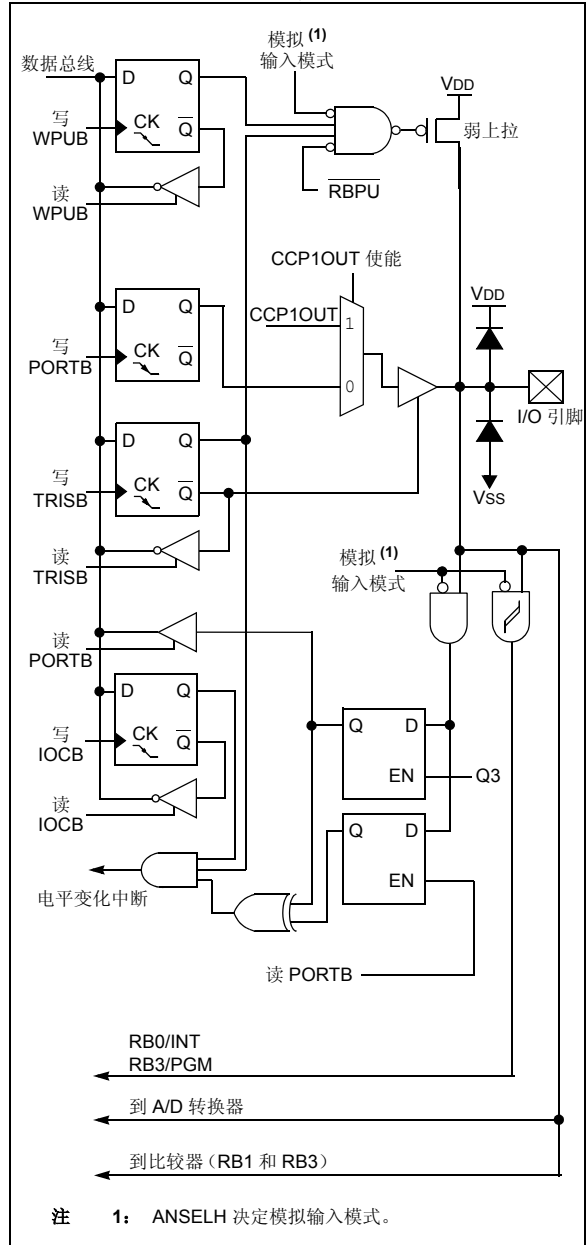
注 1： 只有 PIC16F882/883/886 上有 P1B。

3.4.4.4 RB3/AN9/PGM/C12IN2-

图 3-9 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- ADC 的模拟输入
- 低电压在线串行编程使能引脚
- 比较器 C1 或 C2 的模拟输入

图 3-9: RB<3:0> 的框图



PIC16F882/883/884/886/887

3.4.4.5 RB4/AN11/P1D⁽¹⁾

图 3-10 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- ADC 的模拟输入
- PWM 输出⁽¹⁾

注 1: P1D 仅在 PIC16F882/883/886 上可用。

3.4.4.6 RB5/AN13/T1G

图 3-10 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- ADC 的模拟输入
- Timer1 门控输入

3.4.4.7 RB6/ICSPCLK

图 3-10 显示了此引脚的框图。此引脚可被配置为以下功能之一：

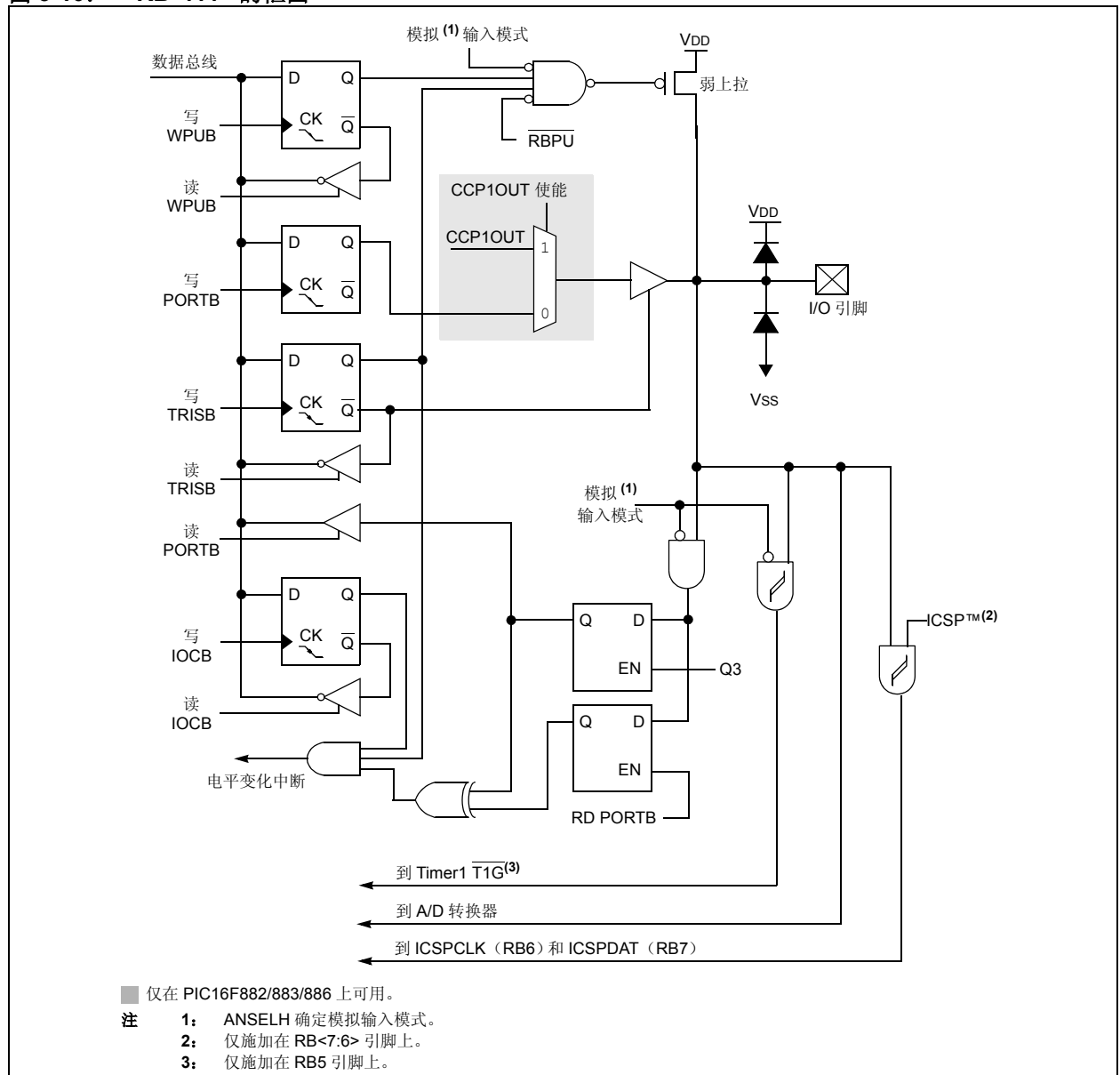
- 通用 I/O
- 在线串行编程时钟

3.4.4.8 RB7/ICSPDAT

图 3-10 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- 在线串行编程数据

图 3-10: RB<7:4> 的框图



PIC16F882/883/884/886/887

表 3-2: 与 PORTB 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
ANSELH	—	—	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8	--11 1111	--11 1111
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
CM2CON1	MC1OUT	MC2OUT	C1RSEL	C2RSEL	—	—	T1GSS	C2SYNC	0000 --10	0000 --10
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0	0000 0000	0000 0000
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
OPTION_REG	RBP _U	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	1111 1111	1111 1111

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

PIC16F882/883/884/886/887

3.5 PORTC 和 TRISC 寄存器

PORTC 是一个 8 位宽的双向端口。对应的数据方向寄存器为 TRISC（寄存器 3-10）。将 TRISC 中的某个位置 1（= 1）可以使对应的 PORTC 引脚作为输入引脚（即将相应的输出驱动器置于高阻态）。将 TRISC 中的某个位清零（= 0）将使对应的 PORTC 引脚作为输出引脚（即将输出锁存器的内容从选定引脚输出）。例 3-4 给出了初始化 PORTC 的过程。

读 PORTC 寄存器（寄存器 3-9）读的是引脚的状态而写该寄存器将会写入端口锁存器。所有写操作都是读—修改—写操作。因此，写一个端口就意味着先读该端口的引脚电平，修改读到的值，然后再将改好的值写入端口数据锁存器。

即使在 PORTC 引脚用作模拟输入时，TRISC 寄存器（寄存器 3-10）仍然控制 PORTC 引脚的方向。当将 PORTC 引脚用作模拟输入时，用户必须确保 TRISC 寄存器中的位保持为置 1 状态。配置为模拟输入的 I/O 引脚总是读为 0。

例 3-4: 初始化 PORTC

```
BANKSEL PORTC      ;
CLRF   PORTC       ;Init PORTC
BANKSEL TRISC      ;
MOVLW  B'00001100' ;Set RC<3:2> as inputs
MOVWF  TRISC       ;and set RC<7:4,1:0>
                          ;as outputs
```

寄存器 3-9: PORTC: PORTC 寄存器

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
bit 7							bit 0

图注:

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

bit 7-0 **RC<7:0>**: PORTC 通用 I/O 引脚位
 1 = 端口引脚电平 > V_{IH}
 0 = 端口引脚电平 < V_{IL}

寄存器 3-10: TRISC: PORTC 三态寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1 ⁽¹⁾	R/W-1 ⁽¹⁾
TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
bit 7							bit 0

图注:

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

bit 7-0 **TRISC<7:0>**: PORTC 三态控制位
 1 = PORTC 引脚被配置为输入（三态）
 0 = PORTC 引脚被配置为输出

注 1: TRISC<1:0> 在 LP 振荡器模式下总是读为 1。

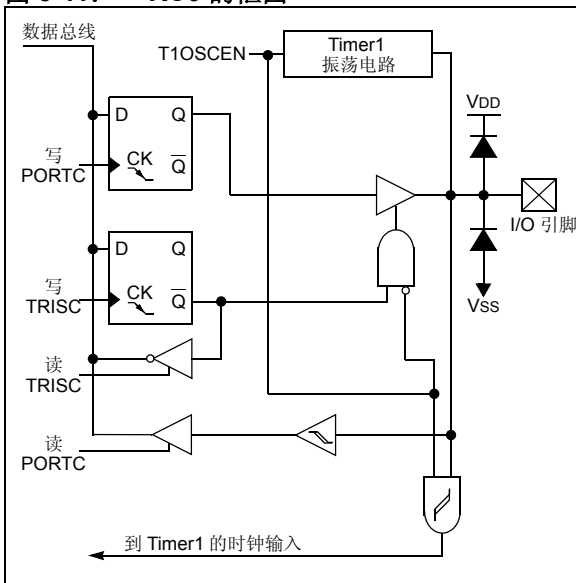
PIC16F882/883/884/886/887

3.5.1 RC0/T1OSO/T1CKI

图 3-11 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- Timer1 的振荡器输出
- Timer1 的时钟输入

图 3-11: RC0 的框图

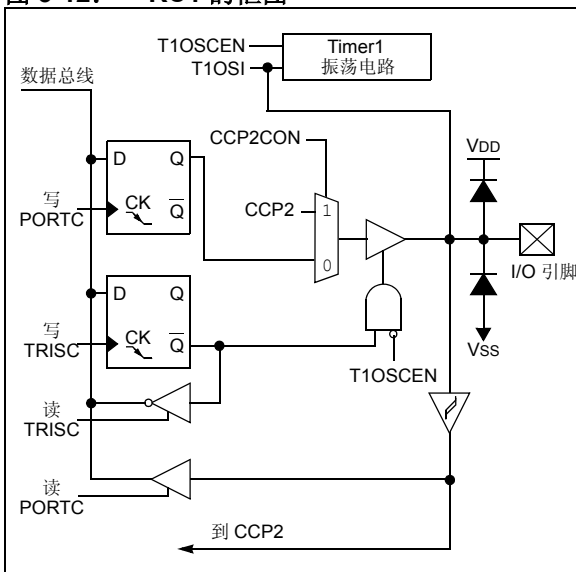


3.5.2 RC1/T1OSI/CCP2

图 3-12 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- Timer1 的振荡器输入
- 比较器 C2 的捕捉输入和比较 /PWM 输出

图 3-12: RC1 的框图

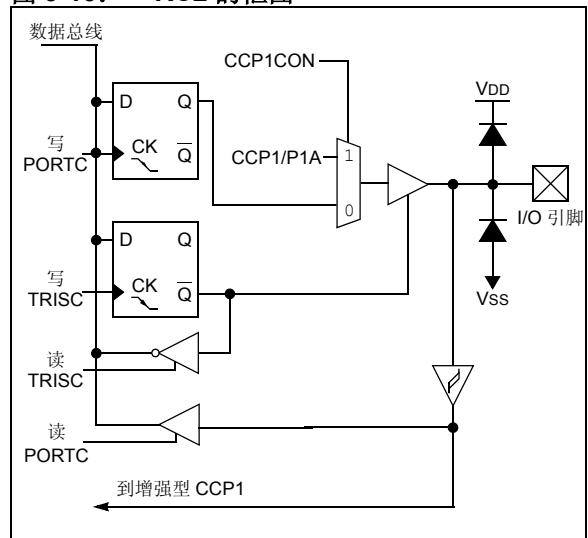


3.5.3 RC2/P1A/CCP1

图 3-13 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- PWM 输出
- 比较器 C1 的捕捉输入和比较输出

图 3-13: RC2 的框图



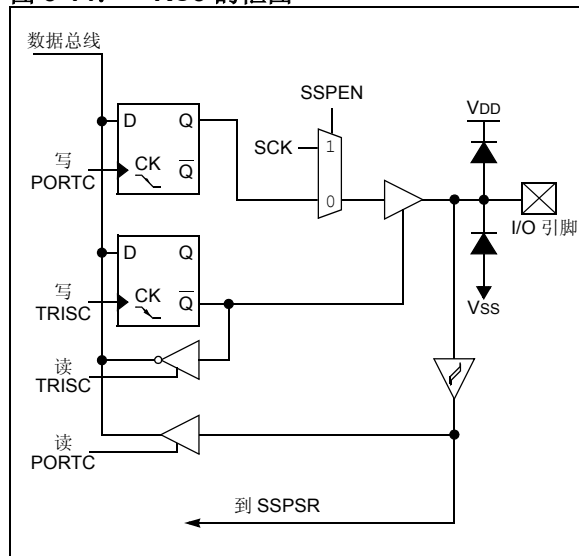
PIC16F882/883/884/886/887

3.5.4 RC3/SCK/SCL

图 3-14 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- SPI 时钟
- I²C™ 时钟

图 3-14: RC3 的框图

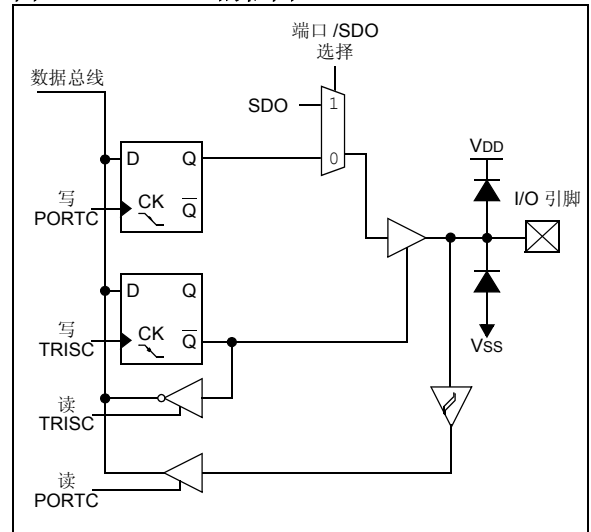


3.5.6 RC5/SDO

图 3-16 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- 串行数据输出

图 3-16: RC5 的框图

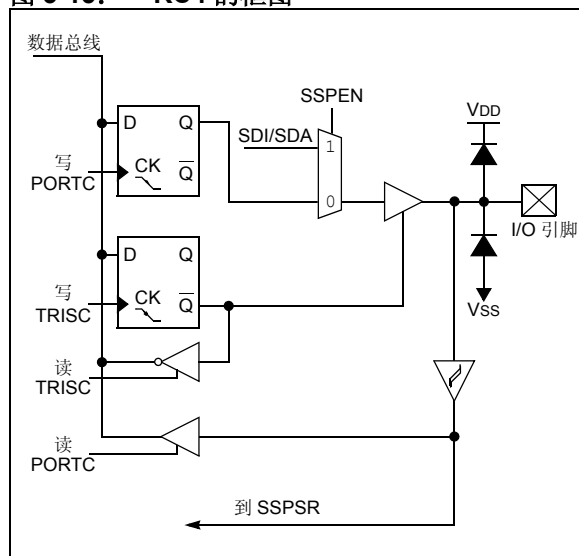


3.5.5 RC4/SDI/SDA

图 3-15 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- SPI 数据 I/O
- I²C 数据 I/O

图 3-15: RC4 的框图



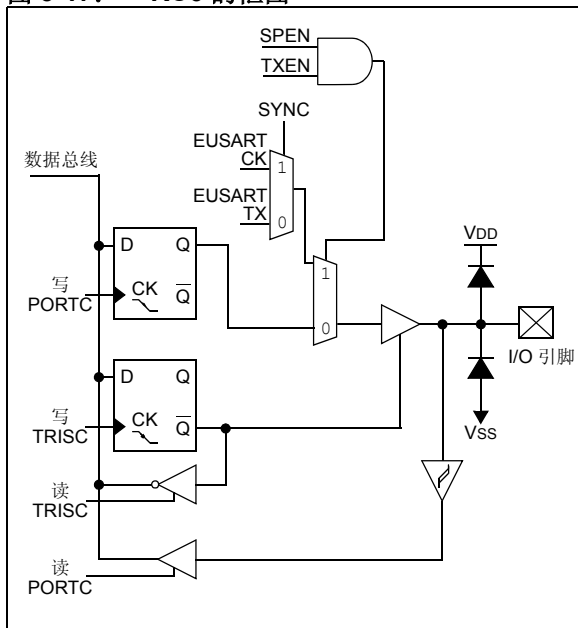
PIC16F882/883/884/886/887

3.5.7 RC6/TX/CK

图 3-17 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- 异步串行输出
- 同步时钟 I/O

图 3-17: RC6 的框图



3.5.8 RC7/RX/DT

图 3-18 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- 异步串行输入
- 同步串行数据 I/O

图 3-18: RC7 的框图

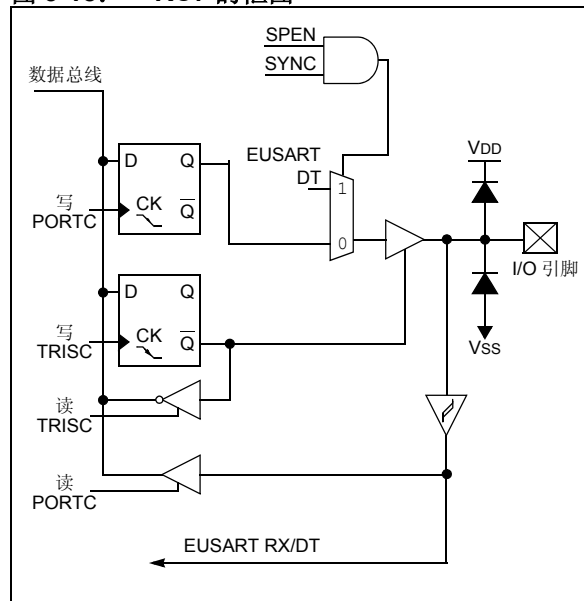


表 3-3: 与 PORTC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
PSTRCON	—	—	—	STRSYNC	STRD	STRC	STRB	STRA	---0 0001	---0 0001
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNCR	TMR1CS	TMR1ON	0000 0000	0000 0000
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111

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

PIC16F882/883/884/886/887

3.6 PORTD 和 TRISD 寄存器

PORTD⁽¹⁾ 是一个 8 位宽的双向端口。对应的数据方向寄存器为 TRISD（寄存器 3-12）。将 TRISD 中的某个位置 1 (= 1) 可以使对应的 PORTD 引脚作为输入引脚（即将相应的输出驱动器置于高阻态）。将 TRISD 中的某个位置 0 (= 0) 将使对应的 PORTD 引脚作为输出引脚（即将输出锁存器的内容从选定引脚输出）。例 3-5 给出了初始化 PORTD 的过程。

读 PORTD 寄存器（寄存器 3-11）读的是引脚的状态而写该寄存器将会写入端口锁存器。所有写操作都是读—修改—写操作。因此，写一个端口就意味着先读该端口的引脚电平，修改读到的值，然后再将改好的值写入端口数据锁存器。

注 1: 只有 PIC16F884/887 上有 PORTD。

即使在 PORTD 引脚用作模拟输入时，TRISD 寄存器（寄存器 3-12）仍然控制 PORTD 引脚的方向。当将 PORTD 引脚用作模拟输入时，用户必须确保 TRISD 寄存器中的位保持为置 1 状态。配置为模拟输入的 I/O 引脚总是读为 0。

例 3-5: 初始化 PORTD

```
BANKSEL PORTD      ;
CLRWF  PORTD       ;Init PORTD
BANKSEL TRISD      ;
MOVLW  B'00001100' ;Set RD<3:2> as inputs
MOVWF  TRISD       ;and set RD<7:4,1:0>
                          ;as outputs
```

寄存器 3-11: PORTD: PORTD 寄存器

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
bit 7							bit 0

图注:

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

bit 7-0 **RD<7:0>**: PORTD 通用 I/O 引脚位
 1 = 端口引脚电平 > V_{IH}
 0 = 端口引脚电平 < V_{IL}

寄存器 3-12: TRISD: PORTD 三态寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0
bit 7							bit 0

图注:

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

bit 7-0 **TRISD<7:0>**: PORTD 三态控制位
 1 = PORTD 引脚被配置为输入（三态）
 0 = PORTD 引脚被配置为输出

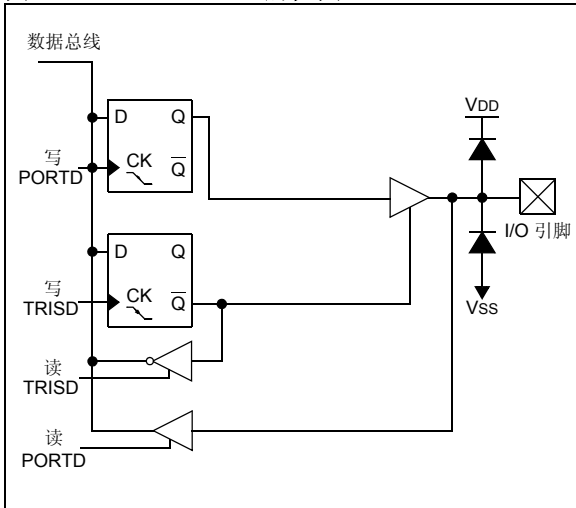
PIC16F882/883/884/886/887

3.6.1 RD<4:0>

图 3-19 显示了这些引脚的框图。这些引脚被配置为通用 I/O。

注： 只有 PIC16F884/887 上有 RD<4:0>。

图 3-19: RD<4:0> 的框图



3.6.2 RD5/P1B⁽¹⁾

图 3-20 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- PWM 输出

注 1： 只有 PIC16F884/887 上有 RD5/P1B。PIC16F882/883/886 上的此功能请参见 RB2/AN8/P1B。

3.6.3 RD6/P1C⁽¹⁾

图 3-20 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- PWM 输出

注 1： 只有 PIC16F884/887 上有 RD6/P1C。PIC16F882/883/886 上的此功能请参见 RB1/AN10/P1C/C12IN3。

3.6.4 RD7/P1D⁽¹⁾

图 3-20 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用 I/O
- PWM 输出

注 1： 只有 PIC16F884/887 上有 RD7/P1D。PIC16F882/883/886 上的此功能请参见 RB4/AN11/P1D。

图 3-20: RD<7:5> 的框图

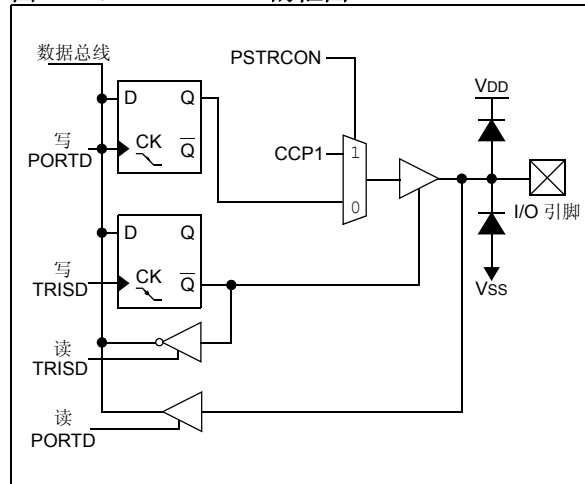


表 3-4: 与 PORTD 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
PSTRCON	—	—	—	STRSYNC	STRD	STRC	STRB	STRA	---0 0001	---0 0001
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	1111 1111

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

PIC16F882/883/884/886/887

3.7 PORTE 和 TRISE 寄存器

PORTE⁽¹⁾ 是一个 4 位宽的双向端口。对应的数据方向寄存器为 TRISE。将 TRISE 中的某个位置 1 (= 1) 可以使对应的 PORTE 引脚作为输入引脚（即将相应的输出驱动器置于高阻态）。将 TRISE 中的某个位清零 (= 0) 将使对应的 PORTE 引脚作为输出引脚（即将输出锁存器的内容从选定引脚输出）。RE3 例外，它是只能输入的引脚，其 TRIS 位将始终读为 1。例 3-6 给出了初始化 PORTE 的过程。

读 PORTE 寄存器（寄存器 3-13）读的是引脚的状态而写该寄存器将会写入端口锁存器。所有写操作都是读—修改—写操作。因此，写一个端口就意味着先读该端口的引脚电平，修改读到的值，然后再将改好的值写入端口数据锁存器。当 MCLRE = 1 时，RE3 读为 0。

注 1: 只有 PIC16F884/887 上有 RE<2:0> 引脚。

即使在 PORTE 引脚用作模拟输入时，TRISE 寄存器（寄存器 3-14）仍然控制 PORTE 引脚的方向。当将 PORTE 引脚用作模拟输入时，用户必须确保 TRISE 寄存器中的位保持为置 1 状态。配置为模拟输入的 I/O 引脚总是读为 0。

注: 必须初始化 ANSEL 寄存器以将模拟通道配置为数字输入。配置为模拟输入的引脚将读为 0。

例 3-6: 初始化 PORTE

```
BANKSEL PORTE      ;
CLRF  PORTE        ;Init PORTE
BANKSEL ANSEL      ;
CLRF  ANSEL        ;digital I/O
BCF   STATUS,RP1   ;Bank 1
BANKSEL TRISE      ;
MOVLW B'00001100' ;Set RE<3:2> as inputs
MOVWF TRISE        ;and set RE<1:0>
                          ;as outputs
```

寄存器 3-13: PORTE: PORTE 寄存器

U-0	U-0	U-0	U-0	R-x	R/W-x	R/W-x	R/W-x
—	—	—	—	RE3	RE2	RE1	RE0
bit 7				bit 0			

图注:

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

bit 7-4 未实现: 读为 0
 bit 3-0 RD<3:0>: PORTE 通用 I/O 引脚位
 1 = 端口引脚电平 > VIH
 0 = 端口引脚电平 < VIL

寄存器 3-14: TRISE: PORTE 三态寄存器

U-0	U-0	U-0	U-0	R-1 ⁽¹⁾	R/W-1	R/W-1	R/W-1
—	—	—	—	TRISE3	TRISE2	TRISE1	TRISE0
bit 7				bit 0			

图注:

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

bit 7-4 未实现: 读为 0
 bit 3-0 TRISE<3:0>: PORTE 三态控制位
 1 = PORTE 引脚被配置为输入（三态）
 0 = PORTE 引脚被配置为输出

注 1: TRISE<3:0> 始终读为 1。

PIC16F882/883/884/886/887

3.7.1 RE0/AN5⁽¹⁾

此引脚可被配置为以下功能之一：

- 通用 I/O
- ADC 的模拟输入

注 1： 只有 PIC16F884/887 上有 RE0/AN5。

3.7.2 RE1/AN6⁽¹⁾

此引脚可被配置为以下功能之一：

- 通用 I/O
- ADC 的模拟输入

注 1： 只有 PIC16F884/887 上有 RE1/AN6。

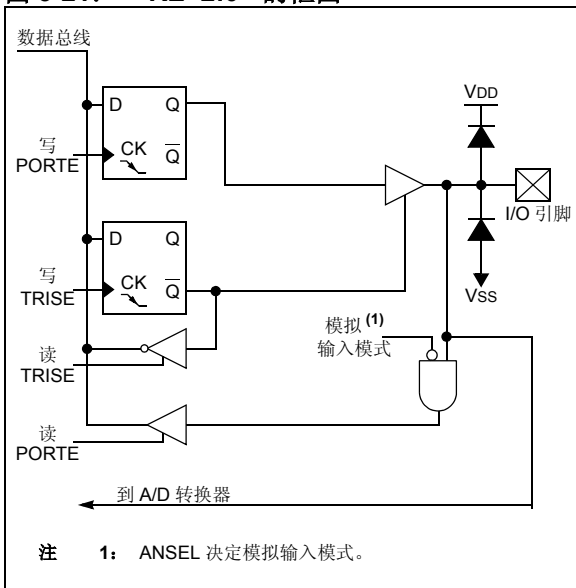
3.7.3 RE2/AN7⁽¹⁾

此引脚可被配置为以下功能之一：

- 通用 I/O
- ADC 的模拟输入

注 1： 只有 PIC16F884/887 上有 RE2/AN7。

图 3-21: RE<2:0> 的框图



3.7.4 RE3/MCLR/VPP

图 3-22 显示了此引脚的框图。此引脚可被配置为以下功能之一：

- 通用输入
- 带有弱上拉的主复位

图 3-22: RE3 的框图

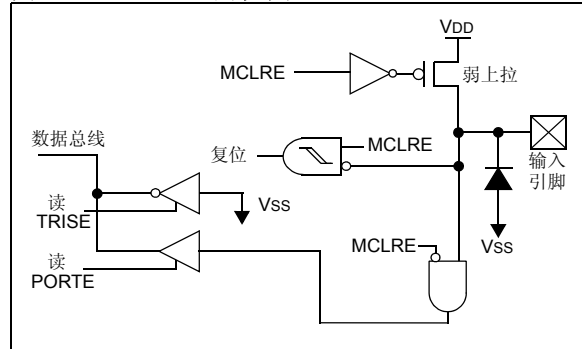


表 3-5: 与 PORTE 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
PORTE	—	—	—	—	RE3	RE2	RE1	RE0	---- xxxx	---- uuuu
TRISE	—	—	—	—	TRISE3	TRISE2	TRISE1	TRISE0	---- 1111	---- 1111

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

PIC16F882/883/884/886/887

4.0 振荡器模块（带故障保护时钟监视器）

4.1 概述

振荡器模块具有多种时钟源和选择功能，从而使其得到广泛应用，并可最大限度地提高性能和降低功耗。图 4-1 说明了振荡器模块的框图。

时钟源可以配置为由外部振荡器、石英晶体谐振器、陶瓷谐振器以及阻容（RC）电路提供。此外，系统时钟源可以配置为由两个内部振荡器之一提供，并可以通过软件选择速度。其他时钟功能包括：

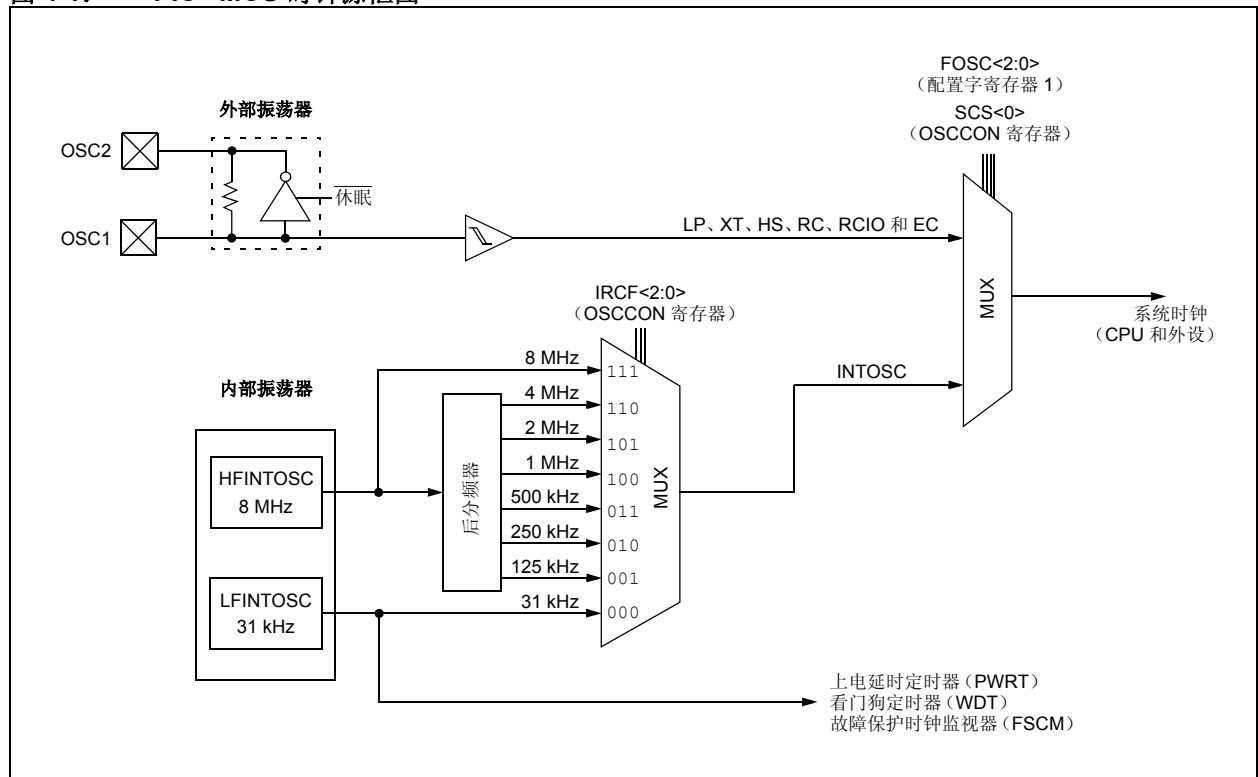
- 通过软件可以选择外部或内部系统时钟源。
- 双速时钟启动模式，使外部振荡器起振到代码执行之间的延时达到最小。
- 故障保护时钟监视器（Fail-Safe Clock Monitor, FSCM）旨在检测外部时钟源（LP、XT、HS、EC 或 RC 模式）的故障并切换到内部振荡器。

可将振荡器模块配置为以下八种时钟模式之一。

1. EC——OSC2/CLKOUT 为 I/O 引脚的外部时钟模式。
2. LP——32 kHz 低功耗晶振模式。
3. XT——中等增益晶振或陶瓷谐振器振荡模式。
4. HS——高增益晶振或陶瓷谐振器振荡模式。
5. RC——外部阻容（RC）振荡模式，且 OSC2/CLKOUT 输出频率为 $F_{osc}/4$ 的时钟信号。
6. RCIO——OSC2/CLKOUT 为 I/O 引脚的外部阻容（RC）振荡模式。
7. INTOSC——OSC2 输出频率为 $F_{osc}/4$ 的时钟信号而 OSC1/CLKIN 为 I/O 引脚的内部振荡器模式。
8. INTOSCIO——OSC1/CLKIN 和 OSC2/CLKOUT 作为 I/O 引脚的内部振荡器模式。

时钟源模式由配置字寄存器 1（CONFIG1）中的 FOSC<2:0> 位配置。内部时钟可以由两个振荡器产生。HFINTOSC 是高频已校准的振荡器。LFINTOSC 是低频未校准的振荡器。

图 4-1: PIC® MCU 时钟源框图



PIC16F882/883/884/886/887

4.2 振荡器控制

振荡器控制 (OSCCON) 寄存器 (图 4-1) 控制系统时钟和频率选择。OSCCON 寄存器包含以下位:

- 频率选择位 (IRCF)
- 频率状态位 (HTS 和 LTS)
- 系统时钟控制位 (OSTS 和 SCS)

寄存器 4-1: **OSCCON: 振荡器控制寄存器**

U-0	R/W-1	R/W-1	R/W-0	R-1	R-0	R-0	R/W-0
—	IRCF2	IRCF1	IRCF0	OSTS ⁽¹⁾	HTS	LTS	SCS
bit 7							bit 0

图注:

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

bit 7 **未实现:** 读为 0

bit 6-4 **IRCF<2:0>:** 内部振荡器频率选择位

111 = 8 MHz
110 = 4 MHz (默认)
101 = 2 MHz
100 = 1 MHz
011 = 500 kHz
010 = 250 kHz
001 = 125 kHz
000 = 31 kHz (LFINTOSC)

bit 3 **OSTS:** 振荡器起振延时状态位⁽¹⁾

1 = 器件依靠由 CONFIG1 寄存器中的 FOSC<2:0> 定义的外部时钟源运行
0 = 器件依靠内部振荡器 (HFINTOSC 或 LFINTOSC) 运行

bit 2 **HTS:** HFINTOSC 状态位 (高频——8 MHz 到 125 kHz)

1 = HFINTOSC 稳定
0 = HFINTOSC 不稳定

bit 1 **LTS:** LFINTOSC 稳定位 (低频——31 kHz)

1 = LFINTOSC 稳定
0 = LFINTOSC 不稳定

bit 0 **SCS:** 系统时钟选择位

1 = 内部振荡器用作系统时钟
0 = 时钟源由 CONFIG1 寄存器中的 FOSC<2:0> 定义

注 1: 双速启动时如果选择了 LP、XT 或 HS 振荡模式或者使能了故障保护模式会使该位复位为 0。

PIC16F882/883/884/886/887

4.3 时钟源模式

时钟源模式可以分为外部或内部两类。

- 外部时钟模式依赖外部电路作为时钟源。示例有振荡器模块（EC 模式）、石英晶体谐振器或陶瓷谐振器（LP、XT 和 HS 模式）以及阻容（RC）模式电路。
- 振荡器模块内部包含了内部时钟源。振荡器模块有两个内部振荡器：8 MHz 高频内部振荡器（HFINTOSC）和 31 kHz 低频内部振荡器（LFINTOSC）。

可通过 OSCCON 寄存器的系统时钟选择（SCS）位选择外部或内部时钟源。更多信息请参见第 4.6 节“时钟切换”。

4.4 外部时钟模式

4.4.1 振荡器起振定时器（OST）

如果振荡器模块被配置为 LP、XT 或 HS 模式，当发生上电复位（POR）且上电延时定时器（PWRT）延时已结束（如果配置了此延时），或从休眠状态唤醒时，振荡器起振定时器（OST）将对 OSC1 引脚进行 1024 次振荡计数。在这段时间内，程序计数器不递增计数，并且程序执行暂停。OST 确保使用石英晶体谐振器或陶瓷谐振器的振荡电路已起振并且为振荡器模块提供稳定的系统时钟。在不同时钟源之间切换时需要一个延时以使新的时钟稳定下来。表 4-1 显示了这些振荡器延时。

为了让外部振荡器起振和代码执行之间的延时缩到最短，可以选择双速时钟启动模式（见第 4.7 节“双速时钟启动模式”）。

表 4-1: 振荡器延时示例

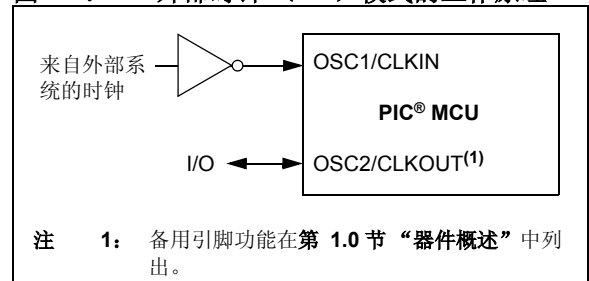
切换自	切换到	频率	振荡器延时
休眠 /POR	LFINTOSC HFINTOSC	31 kHz 125 kHz 至 8 MHz	振荡器预热延时（TWARM）
休眠 /POR	EC 或 RC	DC - 20 MHz	2 个周期
LFINTOSC（31 kHz）	EC 或 RC	DC - 20 MHz	各一个周期
休眠 /POR	LP、XT 或 HS	32 kHz 至 20 MHz	1024 个时钟周期（OST）
LFINTOSC（31 kHz）	HFINTOSC	125 kHz 至 8 MHz	1 μs（大约）

4.4.2 EC 模式

外部时钟（EC）模式将外部产生的逻辑电平作为系统时钟源。当在此模式下工作时，外部时钟源连接到 OSC1 输入引脚，而 OSC2 引脚用作通用 I/O 引脚。图 4-2 显示了 EC 模式的引脚连接。

当选择 EC 模式时，振荡器起振定时器（OST）被禁止。因此，在上电复位（POR）后或从休眠状态唤醒后，不会有延时操作。由于 PIC[®] MCU 的设计是全静态的，停止外部时钟输入可以停止器件并同时保持所有数据的完整性。外部时钟重新起振之后，器件将恢复工作就像没有时间流逝一样。

图 4-2: 外部时钟（EC）模式的工作原理



PIC16F882/883/884/886/887

4.4.3 LP、XT、HS 模式

LP、XT 和 HS 模式支持在 OSC1 和 OSC2 引脚间连接石英晶体谐振器或陶瓷谐振器（图 4-3）。该模式选择内部反相放大器的低、中等或高增益设置以支持多种谐振器类型和速度。

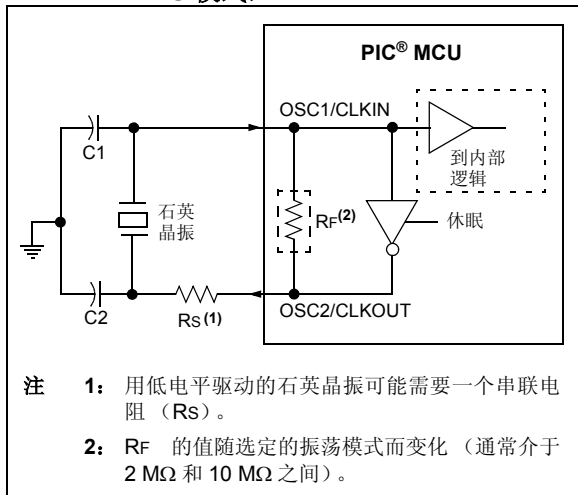
LP 振荡模式选择内部反相放大器的最低增益设置。LP 模式的电流消耗是三种模式中最低的。此模式只用于驱动 32.768 kHz 的调节音叉型（fork type）晶振。

XT 振荡模式选择内部反相放大器的中等增益设置。XT 模式的电流消耗在三种模式中处于中等水平。此模式最适合于驱动具有中等驱动电平规格的谐振器。

HS 振荡模式选择内部反相放大器的最高增益设置。HS 模式的电流消耗是三种模式中最高。此模式最适合于要求高电平驱动设置的谐振器。

图 4-3 和 4-4 分别显示了典型的石英晶体谐振器和陶瓷谐振器电路。

图 4-3: 石英晶振的工作原理 (LP、XT 或 HS 模式)



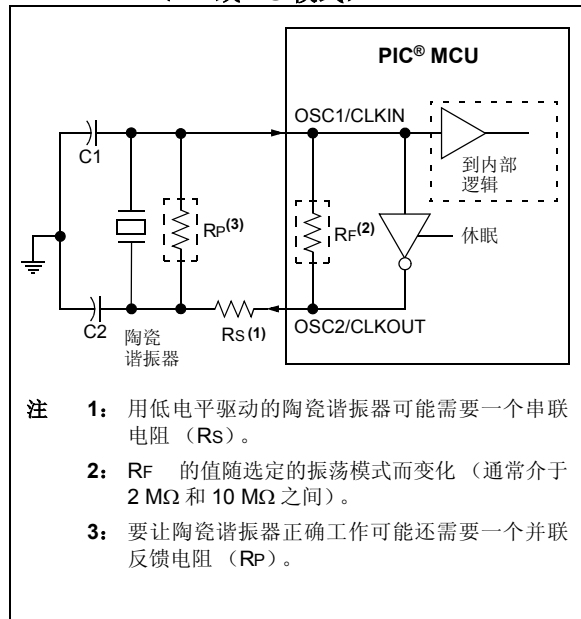
注 1: 石英晶振的特性取决于类型、封装以及制造商。用户应该查阅制造商的数据手册以获知规范和推荐的应用场合。

注 2: 请总是在应用期望的 VDD 和温度范围下验证振荡器的性能。

注 3: 要获取振荡器设计帮助, 请参见以下 Microchip 应用笔记:

- AN826 “Crystal Oscillator Basics and Crystal Selection for *rfPIC*® and *PIC*® Devices” (DS00826)
- AN849 “Basic *PIC*® Oscillator Design” (DS00849)
- AN943 “Practical *PIC*® Oscillator Analysis and Design” (DS00943)
- AN949 “Making Your Oscillator Work” (DS00949)

图 4-4: 陶瓷谐振器的工作原理 (XT 或 HS 模式)



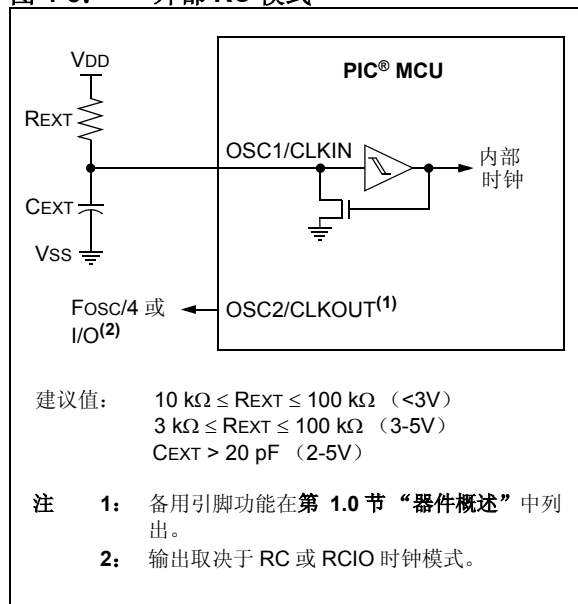
PIC16F882/883/884/886/887

4.4.4 外部 RC 模式

外部阻容 (RC) 模式支持使用外部 RC 电路。当对时钟精度要求不高时, 外部 RC 模式可以让设计人员在选择频率上有最大的灵活性, 同时将成本保持在最低。这种模式分为两种, RC 和 RCIO。

在 RC 模式下, RC 电路与 OSC1 引脚相连。OSC2/CLKOUT 引脚输出 RC 振荡器频率的四分频。此信号可以为外部电路、同步、校准、测试或其他应用要求提供时钟源。图 4-5 显示了外部 RC 模式的连接。

图 4-5: 外部 RC 模式



在 RCIO 模式下, RC 电路与 OSC1 引脚相连。OSC2 引脚变成了另一个通用 I/O 引脚。

RC 振荡器的频率是供电电压、电阻 (R_{EXT})、电容 (C_{EXT}) 值以及工作温度的函数。其他影响振荡频率的因素有:

- 门限电压差异
- 元件公差
- 电容封装差异

用户还需要考虑由于所使用的外部 RC 元件的容差所引起的频率差异。

4.5 内部时钟模式

振荡器模块有两个独立的内部振荡器, 可被配置或选定为系统时钟源。

1. **HFINTOSC** (高频内部振荡器) 已经过厂家校准, 工作频率为 8 MHz。用户可以通过软件使用 OSCTUNE 寄存器 (寄存器 4-2) 调节 HFINTOSC 的频率。
2. **LFINTOSC** (低频内部振荡器) 未经过厂家校准, 工作频率为 31 kHz。

可通过软件使用 OSCCON 寄存器中的内部振荡器频率选择位 $IRCF<2:0>$ 选择系统时钟速度。

可通过 OSCCON 寄存器的系统时钟选择 (SCS) 位选择外部或内部时钟源。更多信息请参见第 4.6 节“时钟切换”。

4.5.1 INTOSC 和 INTOSCIO 模式

当使用配置字寄存器 1 (CONFIG1) 中的振荡器选择 ($FOSC<2:0>$) 位对器件进行编程时, INTOSC 和 INTOSCIO 模式将内部振荡器配置为系统时钟源。

在 INTOSC 模式下, OSC1/CLKIN 引脚可用作通用 I/O 引脚。OSC2/CLKOUT 引脚输出选定的内部振荡器频率的四分频。CLKOUT 信号可以为外部电路、同步、校准、测试或其他应用需求提供时钟源。

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

4.5.2 HFINTOSC

高频内部振荡器 (HFINTOSC) 是经过厂家校准、工作频率为 8 MHz 的内部时钟源。可以通过软件使用 OSCTUNE 寄存器 (寄存器 4-2) 对 HFINTOSC 的频率在大约 $\pm 12\%$ 范围内进行调节。

HFINTOSC 的输出连接到后分频器和多路开关 (见图 4-1)。可以通过软件使用 OSCCON 寄存器的 $IRCF<2:0>$ (见第 4.5.4 节“频率选择位 (IRCF)”) 位在七种频率中选择一种频率。

通过选择 8 MHz 和 125 kHz 之间的任何频率 (设置 OSCCON 寄存器的 $IRCF<2:0>$ 位, 使其 $\neq 000$) 使能 HFINTOSC。之后, 将 OSCCON 寄存器中的系统时钟源选择位 SCS 置 1 或通过配置字寄存器 1 (CONFIG1) 中的 IESO 位置 1 使能双速启动。

OSCCON 寄存器中的 HF 内部振荡器 (HTS) 位指示 HFINTOSC 是否稳定。

PIC16F882/883/884/886/887

4.5.2.1 OSCTUNE 寄存器

HFINTOSC 已经过厂家的校准，但可用软件通过写 OSCTUNE 寄存器（寄存器 4-2）对其进行调节。

OSCTUNE 寄存器的默认值为 0。写入的值是一个 5 位的二进制补码数字。

当修改 OSCTUNE 寄存器时，HFINTOSC 将开始改变到新的频率。在此变动期间，代码会继续执行。不会有任何迹象表明发生了时钟变化。

OSCTUNE 不影响 LFINTOSC 的频率。依赖 LFINTOSC 时钟源频率工作的部件，如上电延时定时器（PWRT）、看门狗定时器（WDT）、故障保护时钟监视器（FSCM）以及外设等的工作不受频率更改的影响。

寄存器 4-2: OSCTUNE: 振荡器调节寄存器

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0
bit 7							bit 0

图注:

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

bit 7-5 **未实现:** 读为 0

bit 4-0 **TUN<4:0>:** 频率调节位

- 01111 = 最高频率
- 01110 =
-
-
-
- 00001 =
- 00000 = 振荡器模块以校准后的频率运行。
- 11111 =
-
-
-
- 10000 = 最低频率

PIC16F882/883/884/886/887

4.5.3 LFINTOSC

低频内部振荡器（LFINTOSC）是未经校准的 31 kHz 的内部时钟源。

LFINTOSC 的输出连接到后分频器和多路开关（见图 4-1）。可以通过软件使用 OSCCON 寄存器的 IRCF<2:0> 位选择 31 kHz。更多信息请参见第 4.5.4 节“频率选择位（IRCF）”。LFINTOSC 输出的频率也是上电延时定时器（PWRT）、看门狗定时器（WDT）和故障保护时钟监视器（FSCM）的时钟频率。

通过将 OSCCON 寄存器的 IRCF<2:0> 位设置为 000 选择 31 kHz 作为系统时钟频率（OSCCON 寄存器的 SCS 位 = 1）或使能以下任何一项都可以使能 LFINTOSC：

- 配置字寄存器 1 的双速启动 IESO 位 = 1 且 OSCCON 寄存器的 IRCF<2:0> 位 = 000
- 上电延时定时器（PWRT）
- 看门狗定时器（WDT）
- 故障保护时钟监视器（FSCM）

OSCCON 寄存器中的 LF 内部振荡器（LTS）位指示 LFINTOSC 是否稳定。

4.5.4 频率选择位（IRCF）

8 MHz HFINTOSC 的输出和 31 kHz LFINTOSC 的输出连接到后分频器和多路开关（见图 4-1）。OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0> 选择内部振荡器的输出频率。可通过软件选择 8 种频率之一：

- 8 MHz
- 4 MHz（复位后的默认值）
- 2 MHz
- 1 MHz
- 500 kHz
- 250 kHz
- 125 kHz
- 31 kHz（LFINTOSC）

注： 发生任何复位后，OSCCON 寄存器的 IRCF<2:0> 位被设置为 110 且频率选择被设置为 4 MHz。用户可修改 IRCF 位以选择其他频率。

4.5.5 HFINTOSC 和 LFINTOSC 时钟切换时序

在 LFINTOSC 和 HFINTOSC 之间切换时，新的振荡器可能已被关闭以节省功耗（见图 4-6）。如果情况是这样，在 OSCCON 寄存器的 IRCF<2:0> 位被修改后到频率选择生效之前将有一个 10 μs 的延时。OSCCON 寄存器的 LTS 和 HTS 位将反映 LFINTOSC 和 HFINTOSC 振荡器的当前状态。频率选择的时序如下：

1. 修改 OSCCON 寄存器的 IRCF<2:0> 位。
2. 如果新的时钟已关闭，将启动一段时钟起振延时。
3. 时钟切换电路等待当前时钟的下降沿。
4. CLKO 保持低电平，时钟切换电路等待新的时钟的上升沿。
5. CLKO 现在与新的时钟连接。按照要求更新 OSCCON 寄存器的 LTS 和 HTS 位。
6. 时钟切换完成。

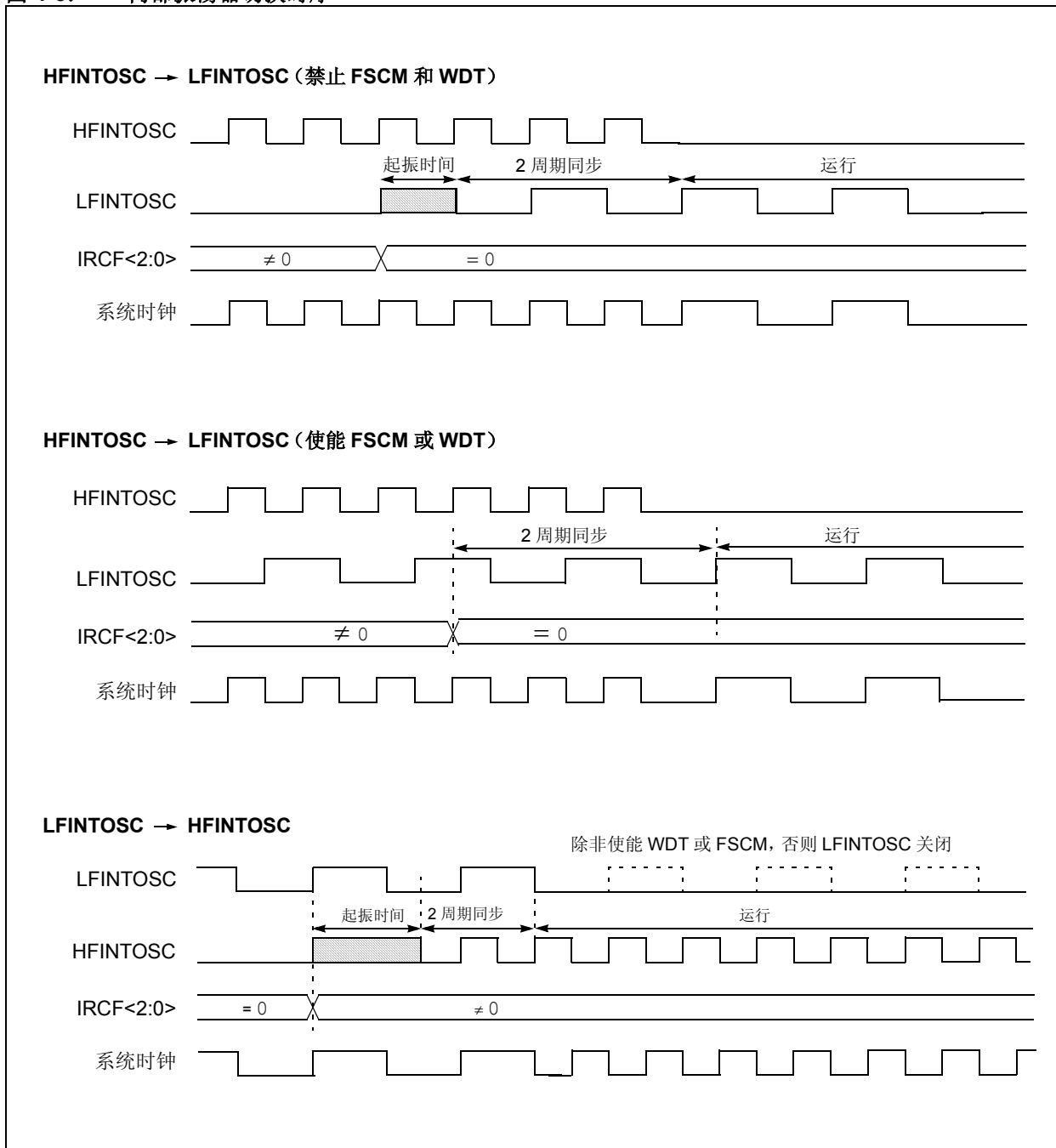
更多详细信息请参见图 4-1。

如果选定的内部振荡器速度介于 125 kHz 和 8 MHz 之间，在选定新的频率之前将没有起振延时。这是因为旧的和新的频率都是由 HFINTOSC 通过后分频器和多路开关后得到的。

第 17.0 节“电气特性”中的振荡器表给出了起振延时规范。

PIC16F882/883/884/886/887

图 4-6: 内部振荡器切换时序



PIC16F882/883/884/886/887

4.6 时钟切换

通过软件使用 OSCCON 寄存器的系统时钟选择 (SCS) 位可以在外部和内部时钟源之间切换系统时钟源。

4.6.1 系统时钟选择 (SCS) 位

OSCCON 寄存器的系统时钟选择 (SCS) 位供 CPU 和外设使用的系统时钟源。

- 当 OSCCON 寄存器的 SCS 位 = 0 时，系统时钟源由配置字寄存器 1 (CONFIG1) 中的 FOSC<2:0> 位决定。
- 当 OSCCON 寄存器的 SCS 位 = 1 时，系统时钟源由 OSCCON 寄存器的 IRCF<2:0> 位选择的内部振荡器频率选择。复位后，OSCCON 寄存器的 SCS 位总是清零。

注： 可能由双速启动或故障保护时钟监视器引起的任何自动时钟切换均不会更新 OSCCON 寄存器的 SCS 位。用户可监视 OSCCON 寄存器的 OSTS 位以确定当前的系统时钟源。

4.6.2 振荡器起振超时状态 (OSTS) 位

OSCCON 寄存器的振荡器起振超时状态 (OSTS) 位指示系统时钟是来自于由配置字寄存器 1 (CONFIG1) 中的 FOSC<2:0> 位定义的外部时钟源还是来自于内部时钟源。特别地，当处于 LP、XT 或 HS 模式时，OSTS 表示振荡器起振定时器 (OST) 已经超时。

4.7 双速时钟启动模式

双速时钟启动模式通过使外部振荡器起振到代码执行之间的延时达到最小而进一步降低功耗。在大量利用休眠模式的应用程序中，双速启动将使唤醒所花费的时间中不包含振荡器的起振时间，并能降低器件的总功耗。

此模式允许应用从休眠状态唤醒，使用 INTOSC 作为时钟源来执行一些指令，然后返回休眠状态，无需等待主振荡器稳定。

注： 执行 SLEEP 指令将中止振荡器起振延时并将使 OSCCON 寄存器的 OSTS 位保持清零。

如果振荡器模块被配置为 LP、XT 或 HS 模式，则使能振荡器起振定时器 (OST) (见第 4.4.1 节“振荡器起振定时器 (OST)”)。OST 定时器将暂停程序执行直到计数完 1024 次振荡为止。双速启动模式通过在 OST 计数时使用内部振荡器作为时钟源，最大限度地缩短代码执行的延时。当 OST 计数达到 1024 且 OSCCON 的 OSTS 位置 1 时，程序执行将切换到由外部振荡器充当时钟源。

4.7.1 双速启动模式的配置

通过下列设置配置双速启动模式：

- IESO (配置字寄存器 1) = 1；内部/外部切换位 (使能双速启动模式)。
- SCS (OSCCON 寄存器) = 0。
- 配置字寄存器 1 (CONFIG1) 中的 FOSC<2:0> 位被配置为 LP、XT 或 HS 模式。

发生下列事件后进入双速启动模式：

- 上电复位 (POR) 后，上电延时定时器 (PWRT) 超时 (如果使能) 后，或者
- 从休眠状态唤醒后。

如果外部时钟振荡器被配置为除 LP、XT 或 HS 以外的任何其他模式，那么双速启动将被禁止。这是因为在上电复位后或从休眠状态退出后，外部时钟振荡器将不需要任何稳定时间。

4.7.2 双速启动时序

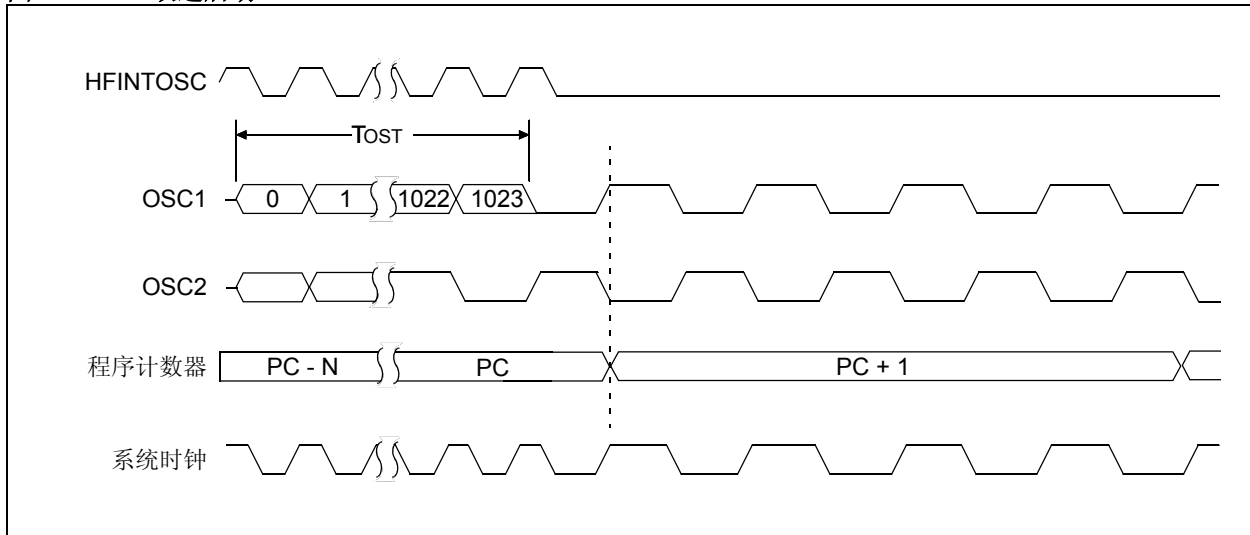
1. 发生上电复位或从休眠状态唤醒。
2. 以内部振荡器作为时钟源 (以 OSCCON 寄存器的 IRCF<2:0> 位设置的频率) 开始执行指令。
3. 使能 OST 计数 1024 个时钟周期。
4. OST 超时，等待内部振荡器的下降沿。
5. OSTS 置 1。
6. 系统时钟保持低电平直到新的时钟的下一个下降沿 (LP、XT 或 HS 模式)。
7. 系统时钟切换到外部时钟源。

PIC16F882/883/884/886/887

4.7.3 检查双速时钟状态

检查 OSCCON 寄存器的 OSTS 位可确定单片机使用的时钟源是配置字寄存器 1 (CONFIG1) 中 FOSC<2:0> 位定义的外部时钟源还是内部振荡器。

图 4-7: 双速启动

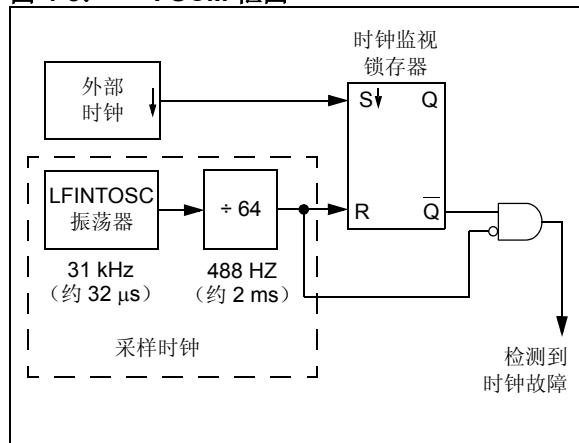


PIC16F882/883/884/886/887

4.8 故障保护时钟监视器

故障保护时钟监视器（FSCM）旨在使器件能在振荡器发生故障时继续运行。FSCM 可以检测出在振荡器起振定时器（OST）延时结束后的任何时刻发生的振荡器故障。通过将配置字寄存器 1（CONFIG1）中的 FCMEN 位置 1 使能 FSCM。它适用于所有外部振荡器模式（LP、XT、HS、EC、RC 和 RCIO）。

图 4-8: FSCM 框图



4.8.1 故障保护检测

FSCM 模块通过比较外部振荡器和 FSDCM 采样时钟检测有故障的振荡器。通过对 INTOSC 时钟进行 64 分频得到 FSCM 采样时钟。参见图 4-8。故障检测电路内部有一个锁存器。在外部时钟的每个下降沿将锁存器置 1。在采样时钟的每个上升沿将锁存器清零。如果采样时钟的半个周期在主时钟变为低电平之前完成则将检测到一个故障。

4.8.2 故障保护工作原理

当外部时钟出现故障时，FSCM 将器件时钟切换到内部时钟源并将 PIR2 寄存器中的 OSFIF 标志位置 1。如果 PIE2 寄存器中的 OSFIE 位也置 1 时，OSFIF 标志位置 1 将产生中断。器件固件可采取措施以减轻可能由故障时钟造成的问题。系统时钟将继续采用内部时钟源，直到器件固件成功重启外部振荡器并使时钟重新切换到外部振荡器为止。

FSCM 选择的内部时钟源由 OSCCON 寄存器的 IRCF<2:0> 位决定，从而允许在故障发生前配置内部振荡器。

4.8.3 清除故障保护条件

在复位、执行了 SLEEP 指令或修改了 OSCCON 寄存器的 SCS 位后将清除故障保护条件。改变 SCS 位时，OST 将重新起振。OST 运行时，器件继续依靠在 OSCCON 中选择的 INTOSC 运行。OST 超时后，故障保护条件被清除，器件将依靠外部时钟源运行。必须先清除故障保护条件才能清零 OSFIF 标志位。

4.8.4 复位或从休眠状态唤醒

FSCM 设计为检测振荡器起振定时器（OST）延时结束后的任何时刻发生的振荡器故障。从休眠中唤醒或任何类型的复位后使用 OST。OST 不与 EC 或 RC 时钟模式一起使用，因此 FSCM 将在复位或唤醒后立即生效。使能 FSCM 也将使能双速启动。因此，在 OST 运行时，器件将始终执行代码。

注： 由于振荡器的起振时间范围很广，故障保护电路在振荡器起振期间（即从复位或休眠状态退出后）不工作。在一段适当的时间后，用户应该检查 OSCCON 寄存器的 OSTS 位以验证振荡器起振和系统时钟切换是否已经成功完成。

PIC16F882/883/884/886/887

图 4-9: FSCM 时序图

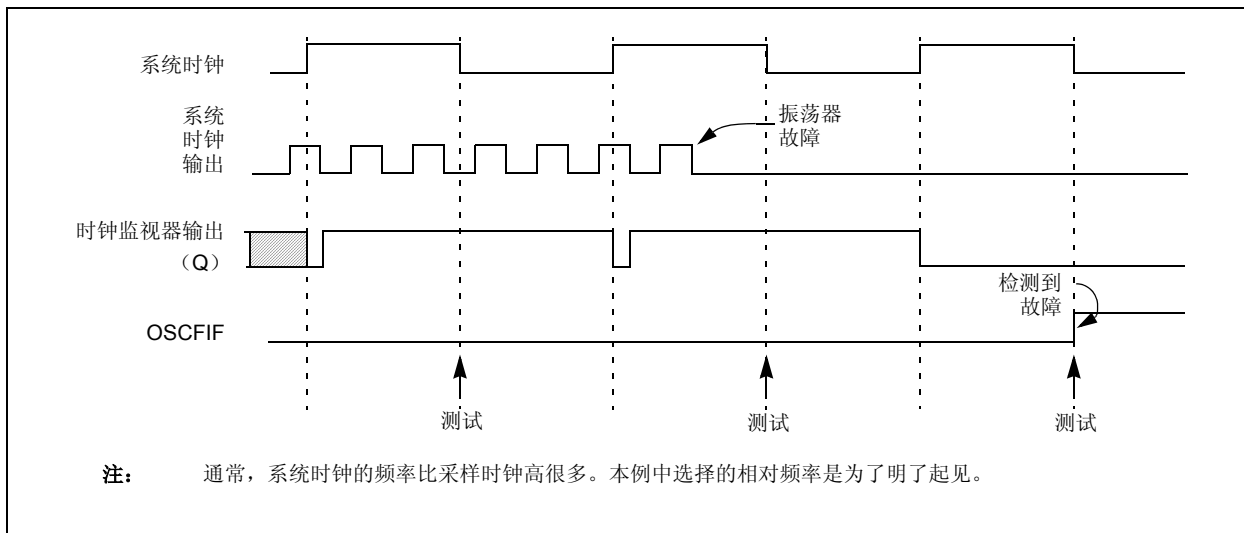


表 4-2: 与时钟源相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值 ⁽¹⁾
CONFIG1 ⁽²⁾	CPD	CP	MCLRE	PWRTE	WDTE	FOSC2	FOSC1	FOSC0	—	—
OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS	-110 x000	-110 x000
OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	---u uuuu
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	ULPWUIE	—	CCP2IE	0000 00-0	0000 00-0
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	ULPWUIF	—	CCP2IF	0000 00-0	0000 00-0

图注: x = 未知, u = 不变, — = 未实现单元 (读为 0)。振荡器未使用阴影单元。

注 1: 其他 (非上电) 复位包括正常运行期间的 MCLR 复位和看门狗定时器复位。

注 2: 所有寄存器位的操作请参见配置字寄存器 1 (寄存器 14-1)。

PIC16F882/883/884/886/887

5.0 TIMER0 模块

Timer0 模块是一个 8 位定时器/计数器，具有如下特点：

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

图 5-1 为 Timer0 模块的框图。

5.1 Timer0 的工作原理

Timer0 模块既可用作 8 位定时器也可用作 8 位计数器。

5.1.1 8 位定时器模式

用作定时器时，Timer0 模块将在每个指令周期递增（不带预分频器）。通过将 OPTION_REG 寄存器的 T0CS 位置 0 可选择定时器模式。

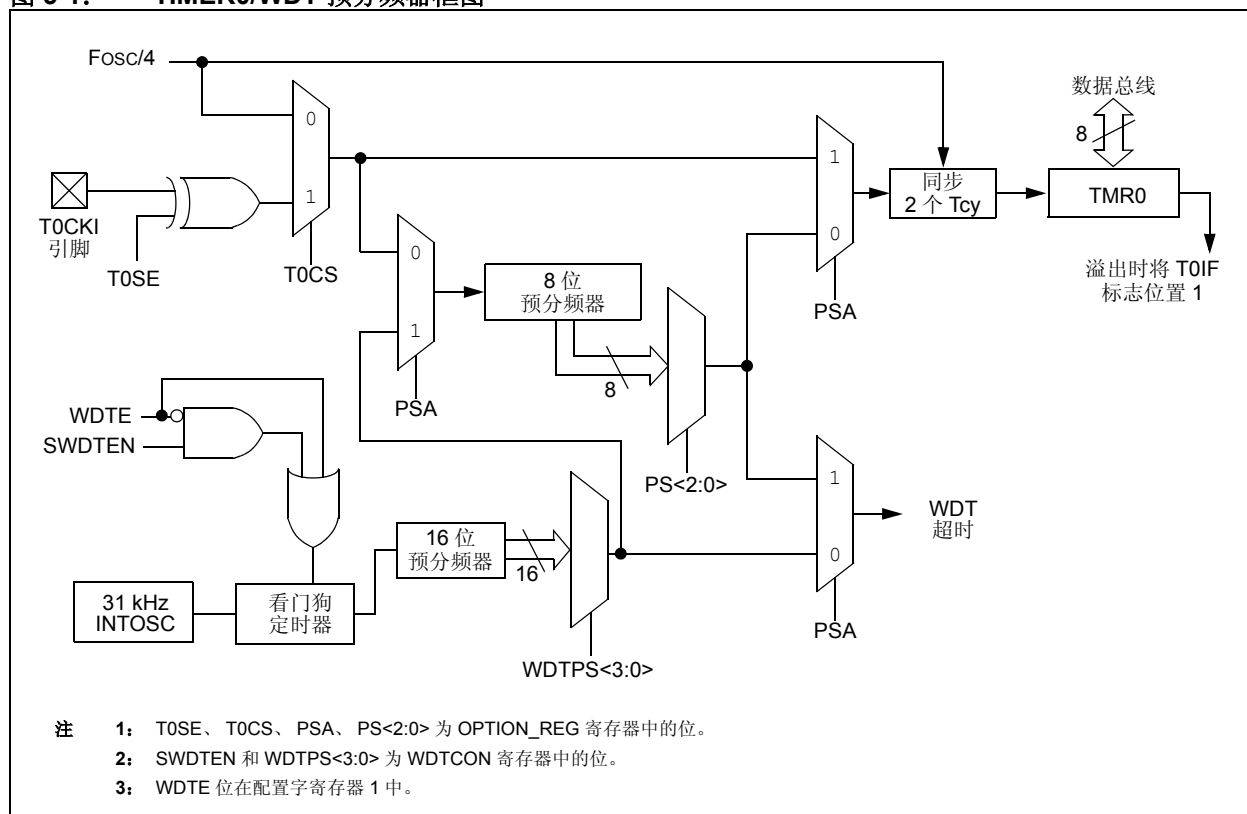
如果对 TMR0 寄存器执行写操作，则在接下来的两个指令周期将禁止递增。

注： 可调整写入 TMR0 寄存器的值，使得在写入 TMR0 时计入两个指令周期的延时。

5.1.2 8 位计数器模式

用作计数器时，Timer0 模块将在 T0CKI 引脚的每个上升沿或下降沿递增。递增的边沿取决于 OPTION_REG 寄存器的 T0SE 位。通过将 OPTION_REG 寄存器的 T0CS 位置 1 可选择计数器模式。

图 5-1: TIMER0/WDT 预分频器框图



PIC16F882/883/884/886/887

5.1.3 软件可编程预分频器

Timer0 和看门狗定时器 (WDT) 共用一个软件可编程预分频器, 但不能同时使用。预分频器的分配由 OPTION_REG 寄存器的 PSA 位控制。要将预分频器分配给 Timer0, PSA 位必须清 0。

Timer0 模块具有 8 种预分频比选择, 范围为 1:2 至 1:256。可通过 OPTION_REG 寄存器的 PS<2:0> 位选择预分频比。要使 Timer0 模块具有 1:1 的预分频比, 必须将预分频器分配给 WDT 模块。

预分频器不可读写。当预分频器分配给 Timer0 模块时, 所有写入 TMR0 寄存器的指令都将使预分频器清零。

当预分频器分配给 WDT 时, CLRWDT 指令将同时清零预分频器和 WDT。

5.1.3.1 在 Timer0 和 WDT 模块间切换预分频器

将预分频器分配给 Timer0 或 WDT 后, 在切换预分频比时可能会产生无意的器件复位。要将预分频器从分配给 Timer0 改为分配给 WDT 模块时, 必须执行如例 5-1 所示的指令序列。

例 5-1: 更改预分频器 (TIMER0 → WDT)

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

要将预分频器从分配给 WDT 改为分配给 Timer0 模块, 必须执行以下指令序列 (见例 5-2)。

例 5-2: 更改预分频器 (WDT → TIMER0)

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

5.1.4 TIMER0 中断

当 TMR0 寄存器从 FFh 溢出至 00h 时, 产生 Timer0 中断。每次 TMR0 寄存器溢出时, 不论是否允许 Timer0 中断, INTCON 寄存器的 T0IF 中断标志位都会置 1。T0IF 位必须在软件中清零。Timer0 中断允许位是 INTCON 寄存器的 T0IE 位。

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

5.1.5 Timer0 与外部时钟配合使用

当 Timer0 工作在计数器模式下时, 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 输入与内部相位时钟的同步。因此, 外部时钟源信号的高、低电平时间必须符合第 17.0 节“电气特性”给出的时序要求。

PIC16F882/883/884/886/887

寄存器 5-1: OPTION_REG: 选项寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

图注:

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

- bit 7 **$\overline{\text{RBPU}}$** : 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**: 中断边沿选择位
 1 = INT 引脚的上升沿触发中断
 0 = INT 引脚的下降沿触发中断
- bit 5 **T0CS**: TMR0 时钟源选择位
 1 = T0CKI 引脚上的跳变沿
 0 = 内部指令周期时钟 (Fosc/4)
- bit 4 **T0SE**: TMR0 时钟源边沿选择位
 1 = 在 T0CKI 引脚上电平发生由高到低的跳变时递增
 0 = 在 T0CKI 引脚上电平发生由低到高的跳变时递增
- bit 3 **PSA**: 预分频器分配位
 1 = 预分频器分配给 WDT
 0 = 预分频器分配给 Timer0 模块
- bit 2-0 **PS<2:0>**: 预分频比选择位

位值	TMR0 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

注 1: 有一个专用的 16 位 WDT 后分频器。更多信息, 请参见第 14.5 节“看门狗定时器 (WDT)”。

表 5-1: 与 TIMER0 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 与 BOR 时的值	所有其他复位时的值
TMR0	Timer0 模块寄存器								xxxx xxxx	uuuu uuuu
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111

图注: - = 未实现单元 (读为 0), u = 不变, x = 未知。Timer0 模块未使用阴影单元。

PIC16F882/883/884/886/887

6.0 带门控控制的 TIMER1 模块

Timer1 模块是一个 16 位定时器/计数器，具有以下特性

- 16 位定时器 / 计数器寄存器 (TMR1H:TMR1L)
- 可编程内部或外部时钟源
- 3 位预分频器
- 可选 LP 振荡器
- 同步或异步操作
- 通过比较器或 T1G 引脚门控 Timer1 (使能计数)
- 溢出中断
- 溢出时唤醒 (仅外部时钟异步模式)
- 捕捉 / 比较功能的时基
- 特殊事件触发功能 (带有 ECCP)
- 比较器输出与 Timer1 时钟同步

图 6-1 为 Timer1 模块的框图。

6.1 Timer1 的工作原理

Timer1 模块是一个通过一对寄存器 TMR1H:TMR1L 访问的 16 位递增计数器。写入 TMR1H 或 TMR1L 可直接更新该计数器。

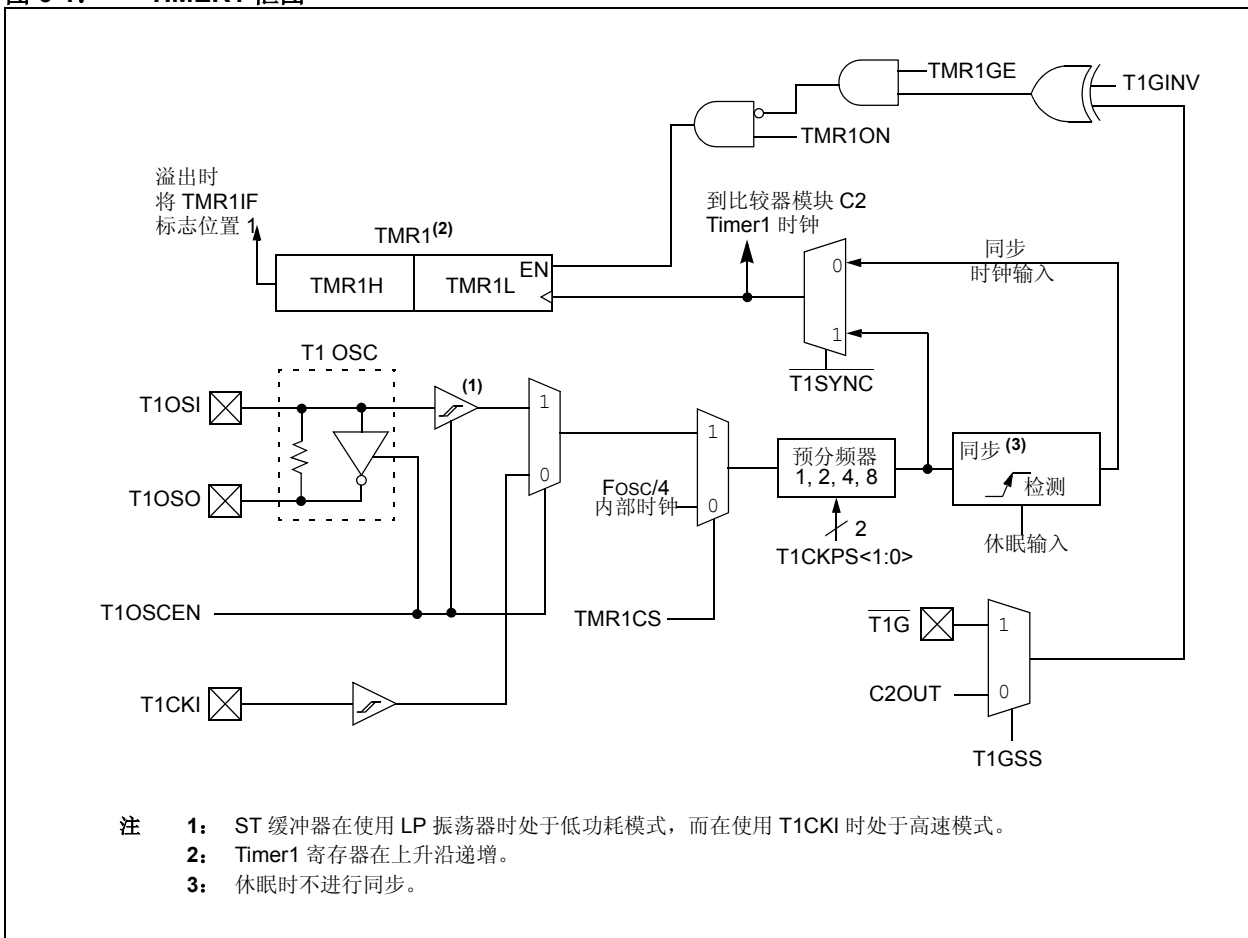
当与内部时钟源一同使用时，此模块用作计数器。当与外部时钟源一同使用时，此模块可用作定时器或计数器。

6.2 时钟源选择

T1CON 寄存器的 TMR1CS 位用于选择时钟源。当 TMR1CS = 0 时，时钟源的频率为 $F_{osc}/4$ 。当 TMR1CS = 1 时，时钟源由外部提供。

时钟源	TMR1CS
$F_{osc}/4$	0
T1CKI 引脚	1

图 6-1: TIMER1 框图



PIC16F882/883/884/886/887

6.2.1 内部时钟源

选择内部时钟源后，TMR1H:TMR1L 寄存器将以 F_{osc} 的倍数为频率递增，具体倍数由 Timer1 预分频器决定。

6.2.2 外部时钟源

选择外部时钟源后，Timer1 模块可作为定时器或计数器。

计数时，Timer1 在外部时钟输入 T1CKI 的上升沿递增。此外，计数器模式下的时钟可与单片机系统时钟同步或异步。

如需一个外部时钟振荡器（且单片机正在使用不带 CLKOUT 的 INTOSC），Timer1 可使用 LP 振荡器作为时钟源。

注： 在计数器模式下，必须先经过一个下降沿，计数器才可以在随后的上升沿进行第一次递增计数。

6.3 Timer1 预分频器

Timer1 具有四种预分频比选择，允许对时钟输入进行 1、2、4 或 8 分频。T1CON 寄存器的 T1CKPS 位控制预分频计数器。不能直接对预分频计数器进行读或写操作；但是，通过写入 TMR1H 或 TMR1L 可清零预分频计数器。

6.4 Timer1 振荡器

在 T1OSI（输入）引脚和 T1OSO（放大器输出）引脚之间连接有一个内置的低功耗 32.768 kHz 晶振。将 T1CON 寄存器的 T1OSCEN 控制位置 1 可使能该振荡器。此振荡器将在休眠模式下继续运行。

Timer1 振荡器与 LP 振荡器完全相同。用户必须提供软件延时，以保证振荡器正常起振。

使能 Timer1 振荡器时 TRISC0 和 TRISC1 位被置 1。RC0 和 RC1 位读为 0 且 TRISC0 和 TRISC1 位读为 1。

注： 振荡器需要经过一段起振和稳定时间后才能使用。因此，在使能 Timer1 前应将 T1OSCEN 置 1 并经过适当的延时。

6.5 在异步计数器模式下的 Timer1 工作原理

如果 T1CON 寄存器中的控制位 T1SYNC 被置 1，外部时钟输入就不同步。定时器继续进行与内部相位时钟异步的递增计数。在休眠状态下定时器仍将继续运行，并在溢出时产生中断，从而唤醒处理器。但是，在用软件对定时器进行读/写操作时应该特别小心（见第 6.5.1 节“异步计数器模式下对 Timer1 的读写操作”）。

注： 当从同步操作切换到异步操作时，有可能漏过一个递增。当从异步操作切换到同步操作时，有可能产生一个误递增。

6.5.1 异步计数器模式下对 TIMER1 的读写操作

当定时器采用外部异步时钟工作时，对 TMR1H 或 TMR1L 的读操作将确保有效（由硬件负责）。但用户应牢记，用读两个 8 位值来读一个 16 位定时器本身就会存在问题，这是因为在两次读操作之间定时器可能会溢出。

对于写操作，建议用户停止定时器后再写入所需数值。当寄存器正在递增计数时，向定时器的寄存器写入数据可能会产生写争用。从而会在 TMR1H:TMR1L 这对寄存器中产生不可预测的值。

6.6 Timer1 门控

可用软件将 Timer1 门控信号源配置为 T1G 引脚或比较器 C2 的输出。这让器件可以直接使用 T1G 为外部事件定时或者使用比较器 C2 为模拟事件定时。有关如何选择 Timer1 门控信号源的信息，请参见 CM2CON1（寄存器 8-2）。此功能部件可以仅仅是 Δ - Σ A/D 转换器的软件，也可以是很多其他应用。欲知有关 Δ - Σ A/D 转换器的更多信息，请访问 Microchip 网站（www.microchip.com）。

注： 必须将 T1CON 寄存器的 TMR1GE 位（T1CON<6>）置 1 以使用 T1G 或 C2OUT 作为 Timer1 的门控信号源。欲知有关选择 Timer1 门控信号源的更多信息，请参见寄存器 8-2。

可使用 T1CON 寄存器的 T1GINV 位来设置 Timer1 门控信号的极性，门控信号可以来自 T1G 引脚也可以来自比较器 C2 的输出。该位可将 Timer1 配置为对两个事件之间的高电平时间或低电平时间进行计时。

PIC16F882/883/884/886/887

6.7 Timer1 中断

一对 Timer1 寄存器 (TMR1H:TMR1L) 递增计数到 FFFFh 后, 将溢出返回 0000h。当 Timer1 溢出时, PIR1 寄存器的 Timer1 中断标志位被置 1。要允许该溢出中断, 用户应将以下位置 1:

- PIE1 寄存器中的 Timer1 中断允许位
- INTCON 寄存器中的 PEIE 位
- INTCON 寄存器中的 GIE 位

在中断服务程序中将 TMR1IF 位清零可以清除该中断。

注: 再次允许该中断前, 应将 TMR1H:TMR1L 这对寄存器以及 TMR1IF 位清零。

6.8 休眠期间的 Timer1 工作原理

只有设置为异步计数器模式时, Timer1 才可在休眠模式下工作。在该模式下, 可使用外部晶振或时钟源使计数器进行递增计数。通过如下设置使定时器能够唤醒器件:

- T1CON 寄存器中的 TMR1ON 位必须置 1
- PIE1 寄存器中的 TMR1IE 位必须置 1
- INTCON 寄存器中的 PEIE 位必须置 1

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

6.9 ECCP 捕捉 / 比较时基

ECCP 模块使用 TMR1H:TMR1L 这对寄存器作为其工作在捕捉或比较模式下的时基。

在捕捉模式下, TMR1H:TMR1L 这对寄存器的值在配置事件发生时被复制到 CCPRxH:CCPRxL 这对寄存器中。

在比较模式下, 当 CCPRxH:CCPRxL 这对寄存器中的值与 TMR1H:TMR1L 这对寄存器中的值匹配时将触发一个事件。此事件可用于触发特殊事件。

更多信息请参见第 11.0 节“捕捉 / 比较 / PWM 模块 (CCP1 和 CCP2)”。

6.10 ECCP 特殊事件触发器

如果将 ECCP 配置为触发一个特殊事件, 触发器将清零 MR1H:TMR1L 这对寄存器。此特殊事件不会导致 Timer1 中断。可仍将 ECCP 模块配置为产生一个 ECCP 中断。

在此工作模式下, CCPRxH:CCPRxL 这对寄存器实际上成为了 Timer1 的周期寄存器。

要使用特殊事件触发器应使 Timer1 与 FOSC 同步。Timer1 在异步模式下工作可导致丢失特殊事件触发信号。

当写入 TMR1H 或 TMR1L 的操作与来自 ECCP 的特殊事件触发信号同时发生时, 写操作具有优先权。

更多信息, 请参见第 11.0 节“捕捉 / 比较 / PWM 模块 (CCP1 和 CCP2)”。

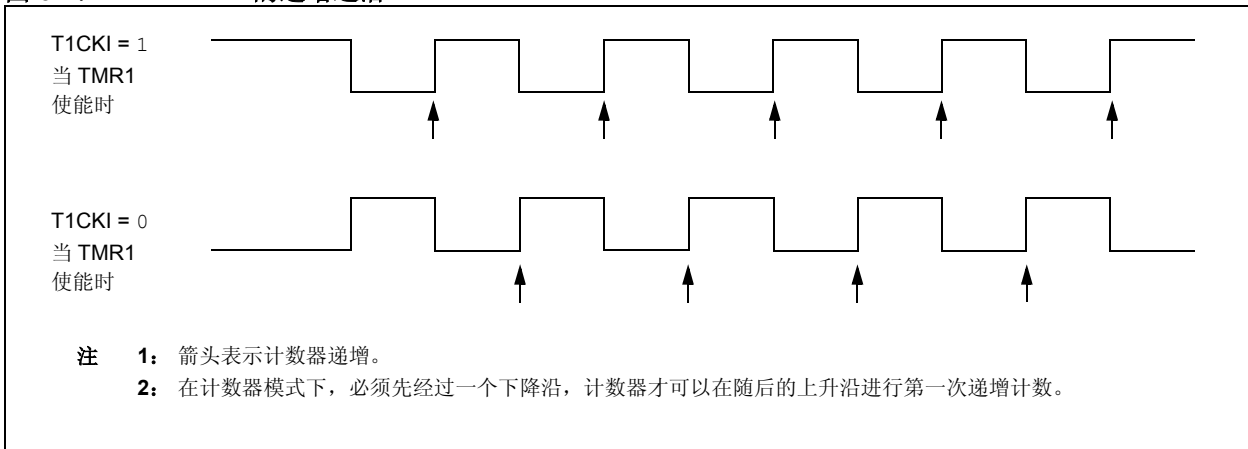
6.11 比较器同步

使 Timer1 递增的同一时钟还可用来同步比较器输出。此功能在比较器模块中使能。

使用比较器进行 Timer1 门控时, 比较器输出应与 Timer1 同步。这样会确保当比较器输出改变时 Timer1 不会错过一次递增计数。

更多信息请参见第 8.0 节“比较器模块”。

图 6-2: TIMER1 的递增边沿



PIC16F882/883/884/886/887

6.12 Timer1 控制寄存器

Timer1 控制寄存器 (T1CON)，如寄存器 6-1 所示，用于控制 Timer1 并选择 Timer1 模块的各种功能。

寄存器 6-1: T1CON: TIMER1 控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T1GINV ⁽¹⁾	TMR1GE ⁽²⁾	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
bit 7							bit 0

图注:

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

- bit 7 **T1GINV:** Timer1 门控信号极性位 ⁽¹⁾
1 = Timer1 门控信号高电平有效 (当门控信号为高电平时 Timer1 计数)
0 = Timer1 门控信号低电平有效 (当门控信号为低电平时 Timer1 计数)
- bit 6 **TMR1GE:** Timer1 门控使能位 ⁽²⁾
如果 TMR1ON = 0:
此位被忽略。
如果 TMR1ON = 1:
1 = 当 Timer1 门控无效时使能 Timer1 门控
0 = 使能 Timer1 门控
- bit 5-4 **T1CKPS<1:0>:** Timer1 输入时钟预分频比选择位
11 = 1:8 预分频比
10 = 1:4 预分频比
01 = 1:2 预分频比
00 = 1:1 预分频比
- bit 3 **T1OSCEN:** LP 振荡器使能控制位
1 = 使能 LP 振荡器作为 Timer1 的时钟源
0 = LP 振荡器关闭
- bit 2 **T1SYNC:** Timer1 外部时钟输入同步控制位
TMR1CS = 1:
1 = 不与外部时钟输入同步
0 = 与外部时钟输入同步
TMR1CS = 0:
忽略此位。Timer1 使用内部时钟。
- bit 1 **TMR1CS:** Timer1 时钟源选择位
1 = 来自 T1CKI 引脚的外部时钟源 (上升沿触发)
0 = 内部时钟源 (FOSC/4)
- bit 0 **TMR1ON:** Timer1 使能位
1 = 使能 Timer1
0 = 禁止 Timer1

- 注 1: T1GINV 位可使 Timer1 门控信号的逻辑电平反相，而不管门控信号源如何。
2: TMR1GE 位必须置 1，以使用 T1G 引脚或 C2OUT (根据 CM2CON1 寄存器的 T1GSS 位选择) 作为 Timer1 的门控信号源。

PIC16F882/883/884/886/887

表 6-1: 与 TIMER1 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
CM2CON1	MC1OUT	MC2OUT	C1RSEL	C2RSEL	—	—	T1GSS	C2SYNC	0000 --10	0000 --10
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								xxxx xxxx	uuuu uuuu
TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								xxxx xxxx	uuuu uuuu
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	uuuu uuuu

图注: x = 未知, u = 不变, - = 未实现 (读为 0)。Timer1 模块未使用阴影单元。

PIC16F882/883/884/886/887

7.0 TIMER2 模块

Timer2 模块是一个 8 位定时器/计数器，具有以下特性：

- 8 位定时器寄存器（TMR2）
- 8 位周期寄存器（PR2）
- TMR2 与 PR2 匹配时中断
- 软件可编程预分频比（1:1、1:4 和 1:16）
- 软件可编程后分频比（1:1 至 1:16）

Timer2 的框图请参见图 7-1。

7.1 Timer2 的工作原理

到 Timer2 模块的时钟输入是系统指令时钟（Fosc/4）。时钟被输入到 Timer2 预分频器，有如下几种分频比可供选择：1:1、1:4 或 1:16。预分频器的输出随后用于使 TMR2 寄存器递增。

持续将 TMR2 和 PR2 的值做比较以确定它们何时匹配。TMR2 将从 00h 开始递增直至与 PR2 中的值匹配。匹配发生时，会发生以下两个事件：

- TMR2 在下一递增周期被复位为 00h
- Timer2 后分频器递增

Timer2 与 PR2 比较器的匹配输出随后输入给 Timer2 的后分频器。后分频器具有 1:1 至 1:16 的预分频比可供选择。Timer2 后分频器的输出用于使 PIR1 寄存器的 TMR2IF 中断标志位置 1。

TMR2 和 PR2 寄存器均可读写。任何复位时，TMR2 寄存器均被设置为 00h 且 PR2 寄存器被设置为 FFh。

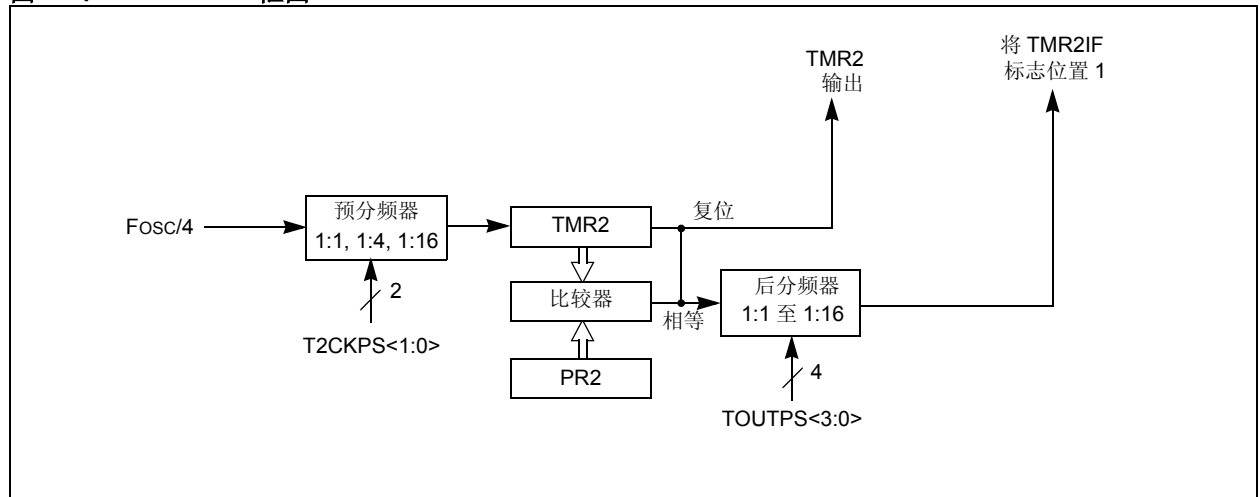
通过将 T2CON 寄存器的 TMR2ON 位置 1 使能 Timer2。通过将 TMR2ON 位清零禁止 Timer2。

Timer2 预分频器由 T2CON 寄存器的 T2CKPS 位控制。Timer2 后分频器由 T2CON 寄存器的 TOUTPS 位控制。预分步器和后分步器计数器在以下情况下被清零：

- 对 TMR2 寄存器执行写操作
- 对 T2CON 寄存器执行写操作
- 发生任何器件复位（上电复位、MCLR 复位、看门狗定时器复位或欠压复位）。

注： 写 T2CON 不会将 TMR2 清零。

图 7-1: TIMER2 框图



PIC16F882/883/884/886/887

寄存器 7-1: T2CON: TIMER2 控制寄存器

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

图注:

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

- bit 7 **未实现:** 读为 0
- bit 6-3 **TOUTPS<3:0>:** Timer2 输出后分频比选择位
 - 0000 = 1:1 后分频比
 - 0001 = 1:2 后分频比
 - 0010 = 1:3 后分频比
 - 0011 = 1:4 后分频比
 - 0100 = 1:5 后分频比
 - 0101 = 1:6 后分频比
 - 0110 = 1:7 后分频比
 - 0111 = 1:8 后分频比
 - 1000 = 1:9 后分频比
 - 1001 = 1:10 后分频比
 - 1010 = 1:11 后分频比
 - 1011 = 1:12 后分频比
 - 1100 = 1:13 后分频比
 - 1101 = 1:14 后分频比
 - 1110 = 1:15 后分频比
 - 1111 = 1:16 后分频比
- bit 2 **TMR2ON:** Timer2 使能位
 - 1 = 使能 Timer2
 - 0 = 禁止 Timer2
- bit 1-0 **T2CKPS<1:0>:** Timer2 时钟预分频比选择位
 - 00 = 预分频值为 1
 - 01 = 预分频值为 4
 - 1x = 预分频值为 16

表 7-1: 与 TIMER2 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
PR2	Timer2 模块周期寄存器								1111 1111	1111 1111
TMR2	8 位 TMR2 寄存器的保持寄存器								0000 0000	0000 0000
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000

图注: x = 未知, u = 不变, — = 未实现 (读为 0)。Timer2 模块未使用阴影单元。

PIC16F882/883/884/886/887

8.0 比较器模块

比较器用作模拟电路与数字电路的接口，通过比较两个模拟电压的大小并输出一个数字量以指示输入量的相对大小。比较器是非常有用的复合信号电路的组成部分，因为它可以提供与执行程序无关的模拟功能。模拟比较器模块包含如下特性：

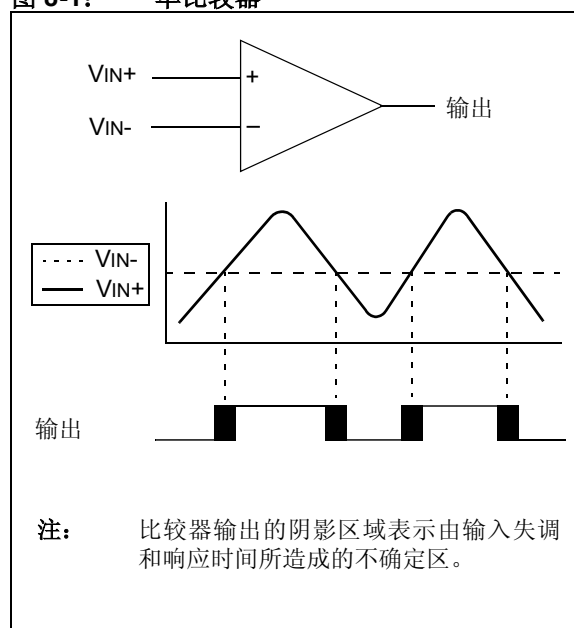
- 独立的比较器控制
- 可编程输入选择
- 可从外部或内部获取比较器输出
- 可编程输出极性
- 电平变化中断
- 从休眠唤醒
- PWM 关闭
- Timer1 选通（使能计数）
- 输出与 Timer1 时钟输入同步
- SR 锁存器
- 可编程和固定参考电压

注： 只有比较器 C2 可与 Timer1 相关联。

8.1 比较器概述

图 8-1 给出了某个比较器及其模拟输入电压与数字输出电平之间的关系。当 V_{IN+} 端的模拟输入电压低于 V_{IN-} 上的模拟电压时，比较器输出数字低电平。当 V_{IN+} 端的模拟输入电压高于 V_{IN-} 上的模拟电压时，比较器输出数字高电平。

图 8-1: 单比较器



PIC16F882/883/884/886/887

图 8-2: 比较器 C1 的简化框图

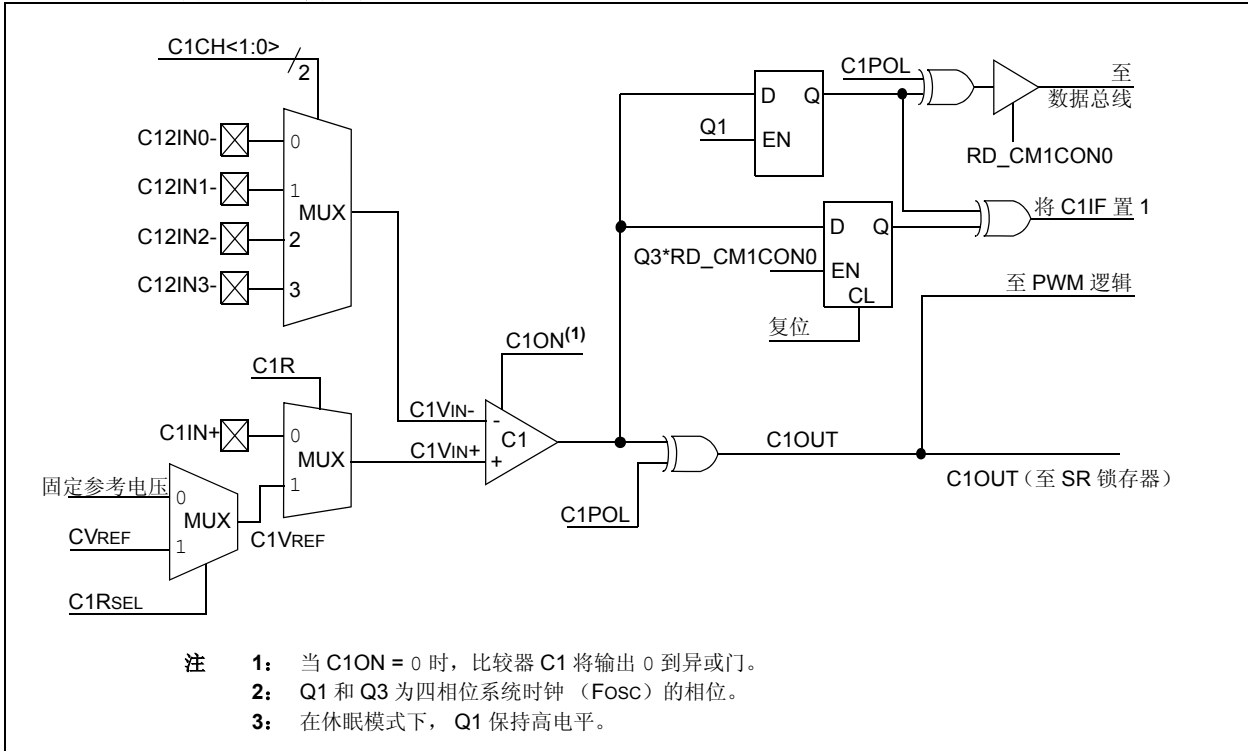
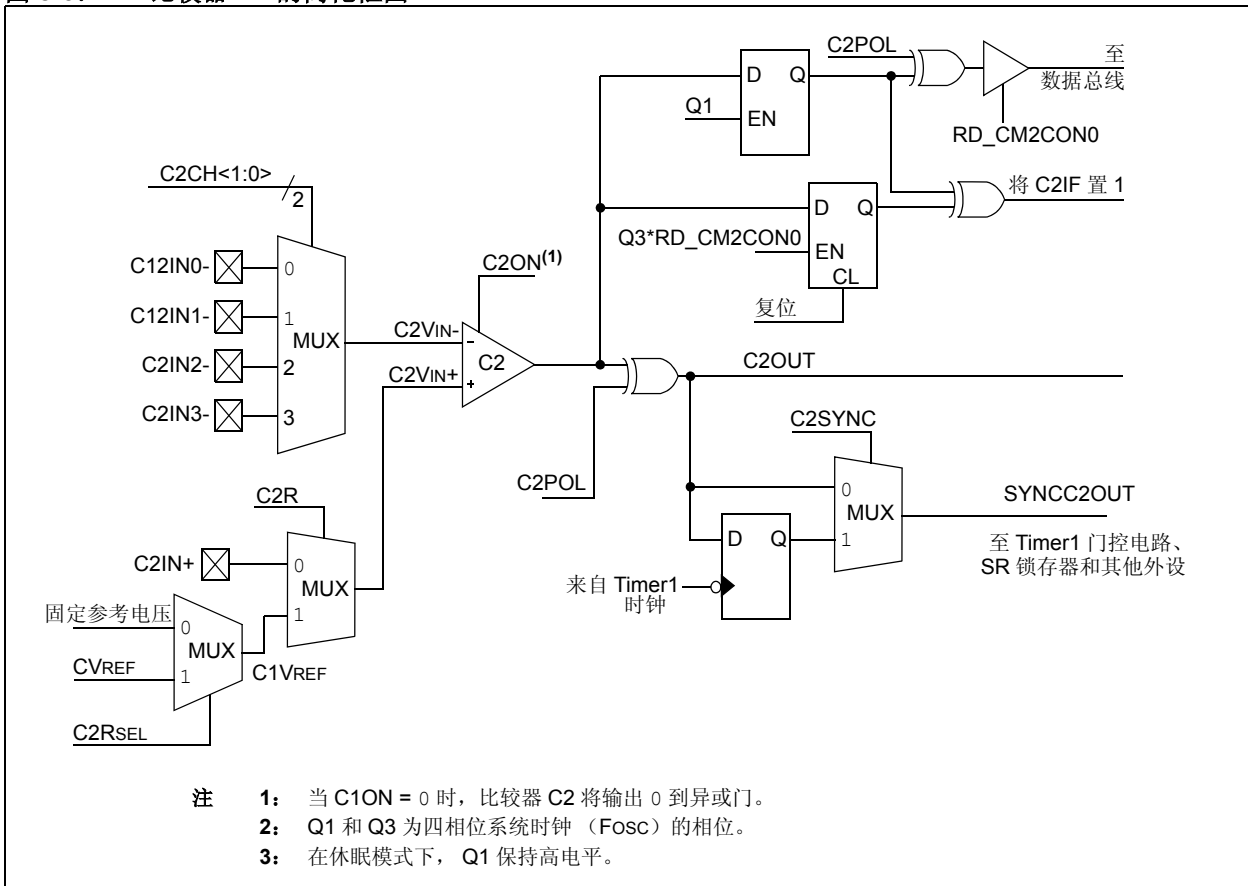


图 8-3: 比较器 C2 的简化框图



PIC16F882/883/884/886/887

8.2 比较器控制

每个比较器都具有独立的控制和配置寄存器，比较器 C1 的控制和配置寄存器为 CM1CON0，比较器 C2 的控制和配置寄存器为 CM2CON0。此外，C2 还有一个辅助控制寄存器 CM2CON1，该寄存器用于控制与 Timer1 的交互以及同时读取两个比较器的读出。

CM1CON0 和 CM2CON0 寄存器（分别见寄存器 8-1 和寄存器 8-2）包含如下控制位和状态位：

- 使能位
- 输入选择位
- 参考选择位
- 输出选择位
- 输出极性位

8.2.1 使能比较器

将 CMxCON0 寄存器中的 CxON 位置 1，使能比较器。将 CxON 位清零禁止比较器，因此可使电流消耗最小。

8.2.2 比较器输入选择

CMxCON0 寄存器的 CxCH<1:0> 位将四个模拟输入引脚之一直接连接到比较器的反向输入端。

注： 要将 CxIN+ 和 CxIN- 用作模拟输入引脚，必须将 ANSEL 和 ANSELH 寄存器的相应位置 1。还必须将 TRIS 位置 1 以禁止输出驱动器。

8.2.3 比较器参考电压选择

将 CMxCON0 寄存器的 CxR 位置 1，将内部参考电压或模拟输入引脚连接到比较器的正向输入端。请参见第 8.10 节“比较器参考电压”获得有关内部参考电压模块的更多信息。

8.2.4 比较器输出选择

通过读 CMxCON0 寄存器的 CxOUT 位或 CM2CON1 寄存器的 MCxOUT 位可以监视比较器的输出。为了保证输出可与外部电路相连，如下条件必须成立：

- CMxCON0 寄存器的 CxOE 位必须置 1
- 相应的 TRIS 位必须清零。
- CMxCON0 寄存器的 CxOE 位必须置 1

- 注**
- 1: CxOE 位掌握端口数据锁存器的主控制权。将 CxON 置 1 不会改写端口。
 - 2: 在每个指令执行周期内，比较器输出会在内部锁存。而外部输出不锁存，除非另外说明。

8.2.5 比较器输出极性

将比较器输出反向与交换比较器的两个输入端功能相同。将 CMxCON0 寄存器的 CxPOL 置 1 可以翻转比较器的输出极性。清零 CxPOL 位则不翻转输出极性。

表 8-1 显示不同输入条件下的输出状态，包括极性控制。

表 8-1: 不同输入条件下的输出状态

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

8.3 比较器响应时间

当改变输入源或选择了新的参考电压后的一段时间，比较器的输出为不确定状态。该段时间称为响应时间。响应时间根据参考电压达到稳定值的时间不同而不同。因此，当输入电压发生变化后要确定比较器的响应时间时，必须将这些时间考虑在内。请参见第 17.0 节“电气特性”的比较器和参考电压规范获得更多详细信息。

PIC16F882/883/884/886/887

8.4 比较器中断工作原理

只要一个比较器的输出值发生了变化，比较器的中断标志位就会被置 1。通过失配电路识别输出端变化，该电路包含两个锁存器和一个异或门（见图 8-2 和图 8-3）。当读 CMxCON0 寄存器时使用比较器的输出电平更新其中一个锁存器。此锁存器保留更新后的值，直到下一次读取 CMxCON0 寄存器或发生复位为止。失配电路的另一个锁存器在每个 Q1 系统时钟更新。如果比较器的输出变化在 Q1 时钟周期内通过第二个锁存器锁存则引发不匹配条件。在该点，失配的锁存器具有相反的输出电平，此电平由异或门进行检测并馈送到中断电路。失配条件将持续存在，直到读 CMxCON0 寄存器或比较器输出返回以前状态为止。

注 1: 对 CMxCON0 寄存器执行操作也将清除失配条件，这是因为在每个写周期开始会首先执行读操作。

2: 比较器中断操作不受 CxOE 状态位的影响。

比较器中断由失配边沿而不是失配电平设置。这表明不需要额外的读或写 CMxCON0 寄存器的操作来清零失配寄存器，就可以使中断标志复位。清零失配寄存器后，如果比较器返回前一状态，则产生中断，否则不产生中断。

当从 CMxCON0 或 CM2CON1 寄存器读数据时，需要用软件来保持比较器输出的状态信息以判断实际发生的变化。

PIR2 寄存器的 CxIF 位为比较器中断标志位。必须在软件中将该位清零使其复位。因为也可以向该寄存器写入 1，因而可模拟中断的产生。

PIE2 寄存器的 CxIE 位和 INTCON 寄存器的 PEIE 和 GIE 位必须置 1，以允许比较器中断。只要这些位中的任何位被清零，尽管当中断条件产生时 PIR2 寄存器的 CxIF 位仍会置 1，但中断却是被禁止的。

图 8-4: 比较器中断时序（读或不读 CMxCON0 寄存器时）

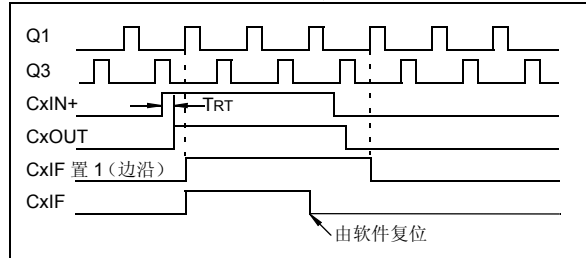
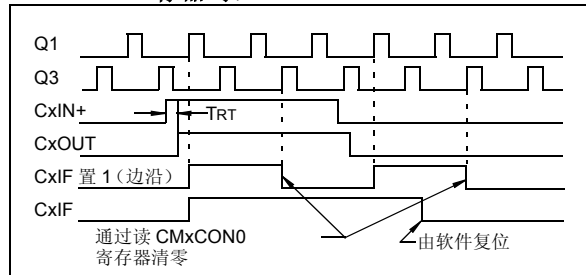


图 8-5: 比较器中断时序（读 CMxCON0 寄存器时）



注 1: 如果在执行一个读操作时（Q2 周期的开始）CMxCON0 寄存器（CxOUT）的值发生了改变，那么 PIR2 寄存器的 CxIF 中断标志位可能不会被置 1。

2: 当首先使能任一比较器时，比较器模块的偏置电路可能导致比较器产生无效的输出现象，直至偏置电路达到稳态。允许大约 1 μs 的偏置稳定时间，然后在允许比较器中断之前，清除失配条件和中断标志。

8.5 休眠期间工作原理

如果在进入休眠模式之前已经使能了比较器，它将在休眠模式下继续工作。比较器额外消耗的电流由**第 17.0 节“电气特性”**单独给出。如果不使用比较器唤醒器件，则可通过关闭比较器使休眠模式的功耗最小。将 CMxCON0 寄存器的 CxON 位清零可以关闭比较器。

比较器输出发生改变可将器件从休眠模式唤醒。将 PIE2 寄存器的 CxIE 位和 INTCON 寄存器的 PEIE 位置 1 以允许比较器将器件从休眠模式唤醒。当器件从休眠模式唤醒后，将始终执行跟随在 SLEEP 指令之后的指令。如果 INTCON 寄存器的 GIE 也置 1，则器件将执行中断服务程序。

8.6 复位的影响

器件复位强制 CMxCON0 和 CM2CON1 寄存器进入复位状态。同时强制比较器和参考电压模块为关闭状态。

PIC16F882/883/884/886/887

寄存器 8-1: CM1CON0: 比较器 C1 控制寄存器 0

R/W-0	R-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0
bit 7							bit 0

图注:

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

- bit 7 **C1ON:** 比较器 C1 使能位
 1 = 使能比较器 C1
 0 = 禁止比较器 C1
- bit 6 **C1OUT:** 比较器 C1 输出位
 如果 C1POL = 1 (极性相反):
 当 C1VIN+ > C1VIN- 时 C1OUT = 0
 当 C1VIN+ < C1VIN- 时 C1OUT = 1
 如果 C1POL = 0 (极性相同):
 当 C1VIN+ > C1VIN- 时 C1OUT = 1
 当 C1VIN+ < C1VIN- 时 C1OUT = 0
- bit 5 **C1OE:** 比较器 C1 输出使能位
 1 = C1OUT 引脚有输出 ⁽¹⁾
 0 = C1OUT 引脚无外部连接
- bit 4 **C1POL:** 比较器 C1 输出极性选择位
 1 = C1OUT 逻辑为反
 0 = C1OUT 逻辑为正
- bit 3 **未实现:** 读为 0
- bit 2 **C1R:** 比较器 C1 参考电压选择位 (正向输入)
 1 = C1VIN+ 引脚与 C1VREF 输出引脚连接
 0 = C1VIN+ 引脚与 C1IN+ 引脚连接
- bit 1-0 **C1CH<1:0>:** 比较器 C1 通道选择位
 00 = C1 的 C12IN0- 引脚连接到 C1VIN- 时
 01 = C1 的 C12IN1- 引脚连接到 C1VIN- 时
 10 = C1 的 C12IN2- 引脚连接到 C1VIN- 时
 11 = C1 的 C12IN3- 引脚连接到 C1VIN- 时

注 1: 若要比比较器产生输出, 需满足如下三个条件: C1OE = 1、C1ON = 1 以及相应端口的 TRIS 位 = 0。

PIC16F882/883/884/886/887

寄存器 8-2: **CM2CON0: 比较器 C2 控制寄存器 0**

R/W-0	R-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0
bit 7							bit 0

图注:

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

- bit 7 **C2ON:** 比较器 C2 使能位
 1 = 使能比较器 C2
 0 = 禁止比较器 C2
- bit 6 **C2OUT:** 比较器 C2 输出位
如果 C2POL = 1 (极性相反):
 当 C2VIN+ > C2VIN- 时 C2OUT = 0
 当 C2VIN+ < C2VIN- 时 C2OUT = 1
如果 C2POL = 0 (极性相同):
 当 C2VIN+ > C2VIN- 时 C2OUT = 1
 当 C2VIN+ < C2VIN- 时 C2OUT = 0
- bit 5 **C2OE:** 比较器 C2 输出使能位
 1 = C2OUT 引脚有输出 ⁽¹⁾
 0 = C2OUT 引脚无外部连接
- bit 4 **C2POL:** 比较器 C2 输出极性选择位
 1 = C2OUT 逻辑为反
 0 = C2OUT 逻辑为正
- bit 3 **未实现:** 读为 0
- bit 2 **C2R:** 比较器 C2 参考电压选择位 (正向输入)
 1 = C2VIN+ 引脚与 C2VREF 输出引脚连接
 0 = C2VIN+ 引脚与 C2IN+ 引脚连接
- bit 1-0 **C2CH<1:0>:** 比较器 C2 通道选择位
 00 = C2 的 C12IN0- 引脚连接到 C2VIN- 时
 01 = C2 的 C12IN1- 引脚连接到 C2VIN- 时
 10 = C2 的 C12IN2- 引脚连接到 C2VIN- 时
 11 = C2 的 C12IN3- 引脚连接到 C2VIN- 时

注 1: 若要比较器产生输出, 需要满足如下三个条件: C2OE = 1、C2ON = 1 以及相应端口的 TRIS 位 = 0。

PIC16F882/883/884/886/887

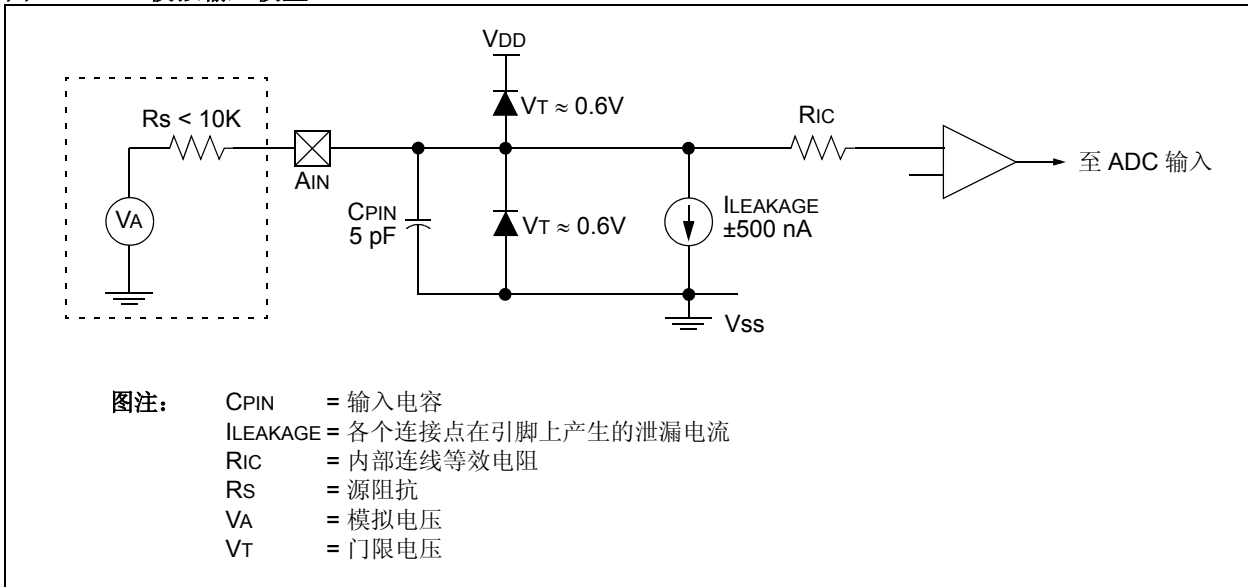
8.7 模拟输入连接注意事项

图 8-6 是一个简化的模拟输入电路。由于模拟输入引脚和数字输出端相连，因此它们与 VDD 和 VSS 之间接有反向偏置二极管，从而将模拟输入电压限制在 VSS 和 VDD 之间。一旦输入电压超出该范围 0.6V，其中一个二极管就会发生正向偏置，从而使输入电压被钳位。

模拟信号源的最大阻抗的建议值为 10 kΩ。连接到模拟输入引脚的任何外部元件（如电容器或齐纳二极管），要保证其漏电流极小，从而使引入的误差最小。

- 注 1:** 读端口寄存器时，所有配置为模拟输入的引脚将读为 0。配置为数字输入的引脚根据输入规范，对模拟输入信号进行相应的转换。
- 注 2:** 施加在数字输入引脚上的模拟电平会使其输入缓冲器消耗的电流超过规定值。

图 8-6: 模拟输入模型



PIC16F882/883/884/886/887

8.8 比较器的其他功能

比较器还有另外三个功能，如下：

- 使能 Timer1 计数（门控）
- 与 Timer1 同步输出
- 同步读取比较器输出

8.8.1 比较器 C2 门控 TIMER1

本功能可以用来为模拟事件的持续时间和间隔定时。将 CM2CON1 寄存器的 T1GSS 位清零将使 Timer1 以比较器 C2 的输出为基准递增。此操作要求打开 Timer1 并使能门控功能。请参见第 6.0 节“带门控控制的 Timer1 模块”获得详细信息。

当比较器用作 Timer1 的门控源时，建议通过将 C2SYNC 位置 1 使比较器与 Timer1 同步。这样可以确保比较器在 Timer1 递增期间改变输出状态时，Timer1 不会丢失计数。

8.8.2 比较器 C2 输出与 TIMER1 同步

将 CM2CON1 寄存器的 C2SYNC 位置 1 可使比较器 C2 的输出与 Timer1 同步。使能时，C2 的输出在 Timer1 时钟源的下降沿被锁存。如果预分频被分配给 Timer1，则比较器的输出将在预分频后锁存。为了防止出现争用，比较器输出在 Timer1 时钟源的下降沿被锁存，而 Timer1 在其时钟源的上升沿递增。请参见比较器框图（图 8-2 和图 8-3）和 Timer1 框图（图 6-1）获取更多信息。

8.8.3 同时读取比较器输出

CM2CON1 寄存器的 MC1OUT 和 MC2OUT 位是比较器两个输出的镜像拷贝。由于能从单个寄存器同时读取两个比较器的输出，因此消除了分别读两个寄存器所造成的时序偏差（timing skew）。

注 1：通过读 CM2CON1 寄存器以获取 C1OUT 或 C2OUT 的状态不会影响比较器的中断失配寄存器。

寄存器 8-3: CM2CON1: 比较器 C2 控制寄存器 1

R-0	R-0	R/W-0	R/W-0	U-0	U-0	R/W-1	R/W-0
MC1OUT	MC2OUT	C1RSEL	C2RSEL	—	—	T1GSS	C2SYNC
bit 7							bit 0

图注:

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

- bit 7 **MC1OUT:** C1OUT 位的镜像拷贝
- bit 6 **MC2OUT:** C2OUT 位的镜像拷贝
- bit 5 **C1RSEL:** 比较器 C1 的参考选择位
1 = CVREF 连接到比较器 C1 的 C1VREF 输入端
0 = 连接到比较器 C1 的 C1VREF 输入端的绝对参考电压 (0.6V) (或器件自带的 1.2V 高精度参考源)
- bit 4 **C2RSE:** 比较器 C2 参考选择位
1 = CVREF 连接到比较器 C2 的 C2VREF 输入端
0 = 连接到比较器 C2 的 C2VREF 输入端的绝对参考电压 (0.6V) (或器件自带的 1.2V 高精度参考源)
- bit 3-2 **未实现:** 读为 0
- bit 1 **T1GSS:** Timer1 门控源选择位
1 = Timer1 门控源为 $\overline{T1G}$
0 = Timer1 门控源为 SYNC2OUT。
- bit 0 **C2SYNC:** 比较器 C2 输出同步位
1 = 输出与 Timer1 时钟的下降沿同步
0 = 输出异步

PIC16F882/883/884/886/887

8.9 比较器 SR 锁存器

SR 锁存器模块提供对比较器输出的额外控制。该模块由单个 SR 锁存器和输出多路开关组成。可根据比较器输出使 SR 锁存器置 1、复位或翻转。也可以通过设置 SRCON 控制寄存器的控制位来使 SR 锁存器置 1 和复位，而与比较器输出无关。SR 锁存器输出多路开关选择将锁存器输出还是比较器输出连接到 I/O 逻辑端口以最终输出到引脚。

8.9.1 锁存器工作原理

该锁存器为不依赖于时钟源的置 1 和复位锁存器。每个置 1 和复位输入引脚都为高电平有效。每个锁存器的输入引脚都与比较器的一个输出引脚和一个由软件控制的脉冲发生器产生。锁存器可通过 C1OUT 或 SRCON 寄存器的 PULSS 位置 1，C2OUT 或 SRCON 寄存器的 PULSR 位复位。锁存器的复位是显性的，也就是说，如果置 1 和复位输入都为高电平，锁存器将进入复位状态。PULSS 和 PULSR 位均可自动复位，也就是说，完成该锁存器的置 1 或复位操作，唯一要做的就是直接写入这 2 位。

8.9.2 锁存器输出

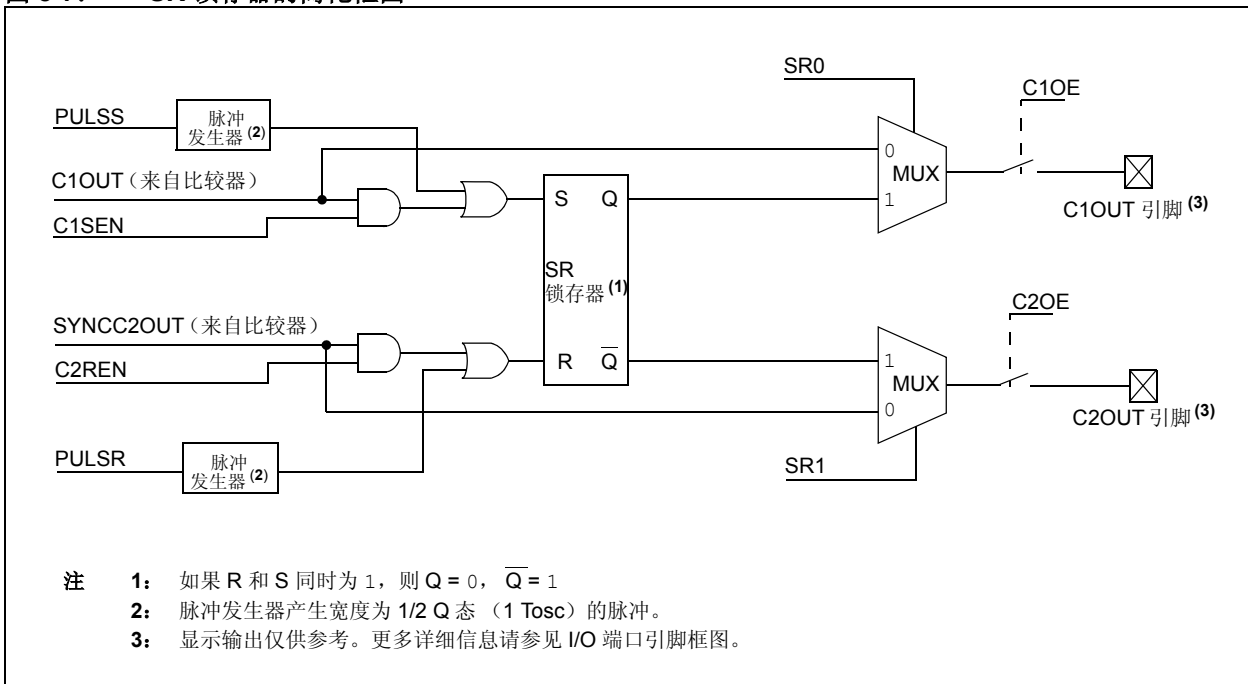
SRCON 寄存器的 SR<1:0> 位控制锁存器的输出多路开关，并决定四种可能的输出配置。在这四种可能的配置中，CxOUT I/O 逻辑端口分别连接：

- C1OUT 和 C2OUT
- C1OUT 和 SR 锁存器 \bar{Q}
- C2OUT 和 SR 锁存器 Q
- SR 锁存器 Q 和 \bar{Q}

任何复位后，默认的输出配置为未锁存的 C1OUT 和 C2OUT 模式，这是为了保持与不具有 SR 锁存器的器件兼容。

必须将相应端口的可用 TRIS 位清零，以使能端口引脚输出驱动器。此外，必须将 CMxCON0 寄存器的 CxOE 比较器输出使能位置 1，以确保比较器或锁存器输出可供输出引脚使用。锁存器配置使能状态与比较器的使能状态毫无关系。

图 8-7: SR 锁存器的简化框图



PIC16F882/883/884/886/887

寄存器 8-4: **SRCON: SR 锁存器控制寄存器**

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

图注:	S = 只可置 1 的位
R = 可读位	W = 可写位
-n = 上电复位时的值	U = 未实现位, 读为 0
	0 = 清零
	1 = 置 1
	x = 未知

- bit 7 **SR1: SR 锁存器配置位 (2)**
 1 = C2OUT 引脚为锁存器 \bar{Q} 输出
 0 = C2OUT 引脚为 C2 比较器输出
- bit 6 **SR0: SR 锁存器配置位 (2)**
 1 = C1OUT 引脚为锁存器 Q 输出
 0 = C1OUT 引脚为 C1 比较器输出
- bit 5 **C1SEN: 由 C1 置 1 使能位**
 1 = 由 C1 比较器输出使 SR 锁存器置 1
 0 = C1 比较器输出不影响 SR 锁存器
- bit 4 **C2SEN: 由 C2 复位使能位**
 1 = 由 C2 比较器输出使 SR 锁存器复位
 0 = C2 比较器输出不影响 SR 锁存器
- bit 3 **PULSS: 是否触发脉冲以将“置 1”命令输入给 SR 锁存器的位**
 1 = 触发脉冲发生器使 SR 锁存器置 1。该位由硬件立即复位。
 0 = 不触发脉冲发生器
- bit 2 **PULSR: 是否触发脉冲以将“复位”命令输入给 SR 锁存器的位**
 1 = 触发脉冲发生器使 SR 锁存器复位。该位由硬件立即复位。
 0 = 不触发脉冲发生器
- bit 1 **未实现:** 读为 0
- bit 0 **FVREN: 固定参考电压使能位**
 1 = 使能来自 INTOSC LDO 的 0.6V 参考电压
 0 = 禁用来自 INTOSC LDO 的 0.6V 参考电压

- 注 1: CMxCON0 寄存器的 CxOUT 位总是反映比较器的真实输出 (而不是引脚电平), 该输出不受锁存器操作的影响。
- 2: 要使 SR 锁存器输出到引脚, 相应的 CxOE 和 TRIS 必须被正确配置。

PIC16F882/883/884/886/887

8.10 比较器参考电压

比较器参考电压模块为比较器提供内部产生的参考电压。它具有如下特性：

- 与比较器操作无关
- 两个具有 16 阶电平的电压范围
- 输出电压钳位为 V_{SS}
- 与 V_{DD} 成比例
- 固定参考电压 (0.6V)

VRCON 寄存器 (寄存器 8-5) 控制参考电压模块, 如图 8-8 所示。

可以通过梯形电阻网络的 16 个端点选择电压源。VRCON 寄存器的 VRSS 位用来选择内部电压源或外部电压源。

PIC16F882/883/884/886/887 仅允许 CVREF 信号在特定的配置条件下输出到 PORTA 的 RA2 引脚。如需更多详细信息, 请参见图 8-9。

8.10.1 独立操作

比较器参考电压与比较器配置无关。将 VRCON 寄存器的 VREN 位置 1 将使能参考电压。

8.10.2 输出电压选择

CVREF 参考电压有 2 个范围, 每个范围都有 16 阶电平。由 VRCON 寄存器的 VRR 位控制范围选择。由 VRCON 寄存器的 VR<3:0> 位设置 16 阶电平。

CVREF 输出电压由如下公式确定:

公式 8-1: CVREF 输出电压

$VRR = 1$ (low range):

$$CVREF = (VR<3:0>/24) \times VLADDER$$

$VRR = 0$ (high range):

$$CVREF = (VLADDER/4) + (VR<3:0> \times VLADDER/32)$$

$$VLADDER = V_{DD} \text{ 或 } ([V_{REF+}] - [V_{REF-}]) \text{ 或 } V_{REF+}$$

由于模块结构的限制, 无法实现从 V_{SS} 到 V_{DD} 满量程输出。请参见图 8-8。

8.10.3 输出电压钳位为 V_{SS}

按如下配置 VRCON, 将 CVREF 输出电压设置为 V_{SS} 而无额外功耗。

- $VREN = 0$
- $VRR = 1$
- $VR<3:0> = 0000$

此操作允许比较器进行过零检测, 而不会使 CVREF 模块消耗额外的电流。

8.10.4 输出与 V_{DD} 成比例

由于比较器的参考电压是依据 V_{DD} 产生的, 因此 CVREF 的输出随着 V_{DD} 的波动而变化。经测试的比较器参考电压绝对准确度可参见第 17.0 节 “电气特性”。

8.10.5 固定参考电压

固定参考电压与 V_{DD} 无关, 其标称输出值为 0.6V。将 SRCON 寄存器的 FVREN 位置 1 可使能该参考电压。只要 HFINTOSC 振荡器处于工作状态, 该参考电压就一直使能。

8.10.6 固定参考电压稳定周期

使能固定参考电压模块后, 需要一段时间使参考电压及其放大电路稳定。用户程序必须包含一个短暂的延时子程序以允许模块稳定。请参见电气规范部分了解最小延时要求。

8.10.7 选择参考电压

可使用参考电压模块输出端上的多路开关选择 CVREF 或固定参考电压以供比较器使用。

将 VRCON 寄存器的 C1VREN 位置 1, 允许电流通过 CVREF 分压器, 并选择 CVREF 电压供 C1 使用。将 C1VREN 位清零选择固定参考电压供 C1 使用。

将 VRCON 寄存器的 C2VREN 位置 1, 允许电流通过 CVREF 分压器, 并选择 CVREF 电压供 C2 使用。将 C2VREN 位清零选择固定参考电压供 C2 使用。

如果 C1VREN 和 C2VREN 都清零, 则禁止电流流入 CVREF 分压器, 从而使参考电压外设的功耗最小。

PIC16F882/883/884/886/887

图 8-8: 比较器参考电压框图

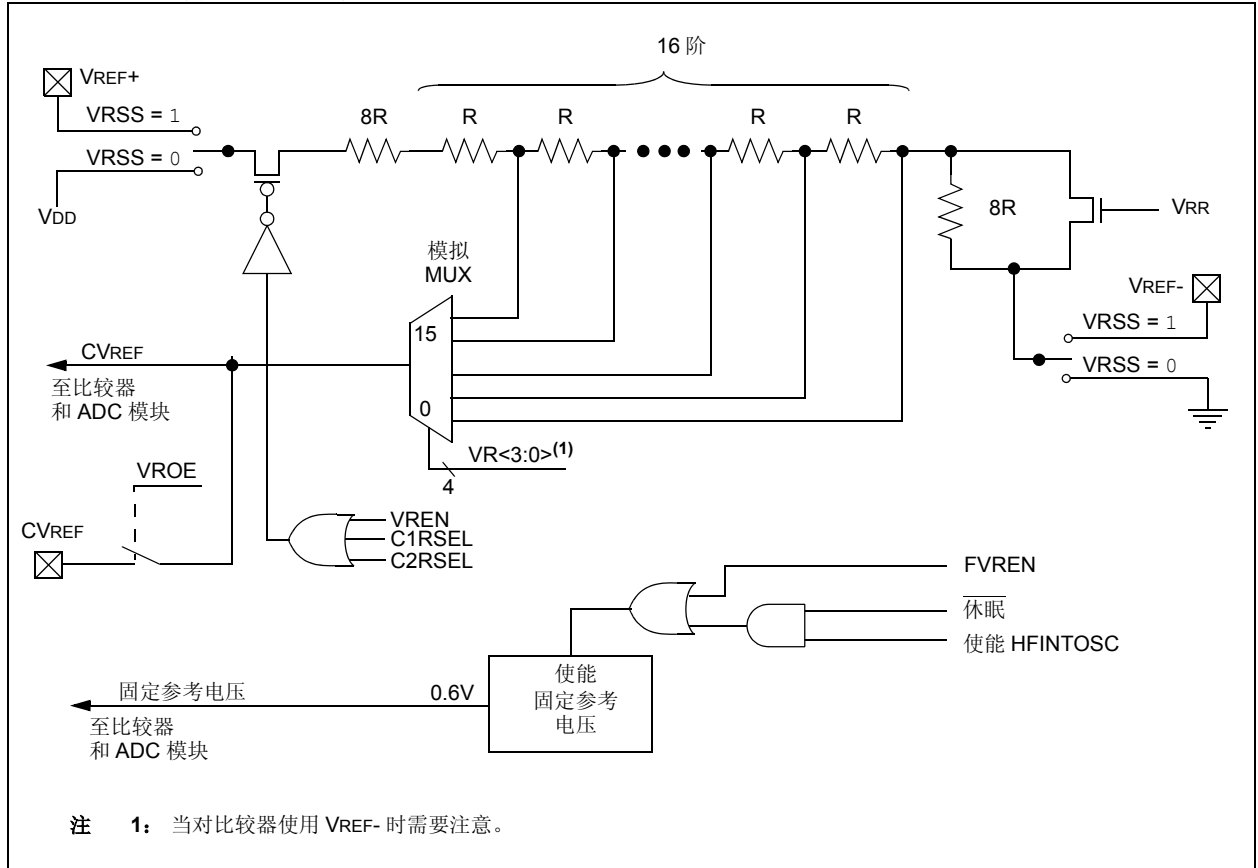
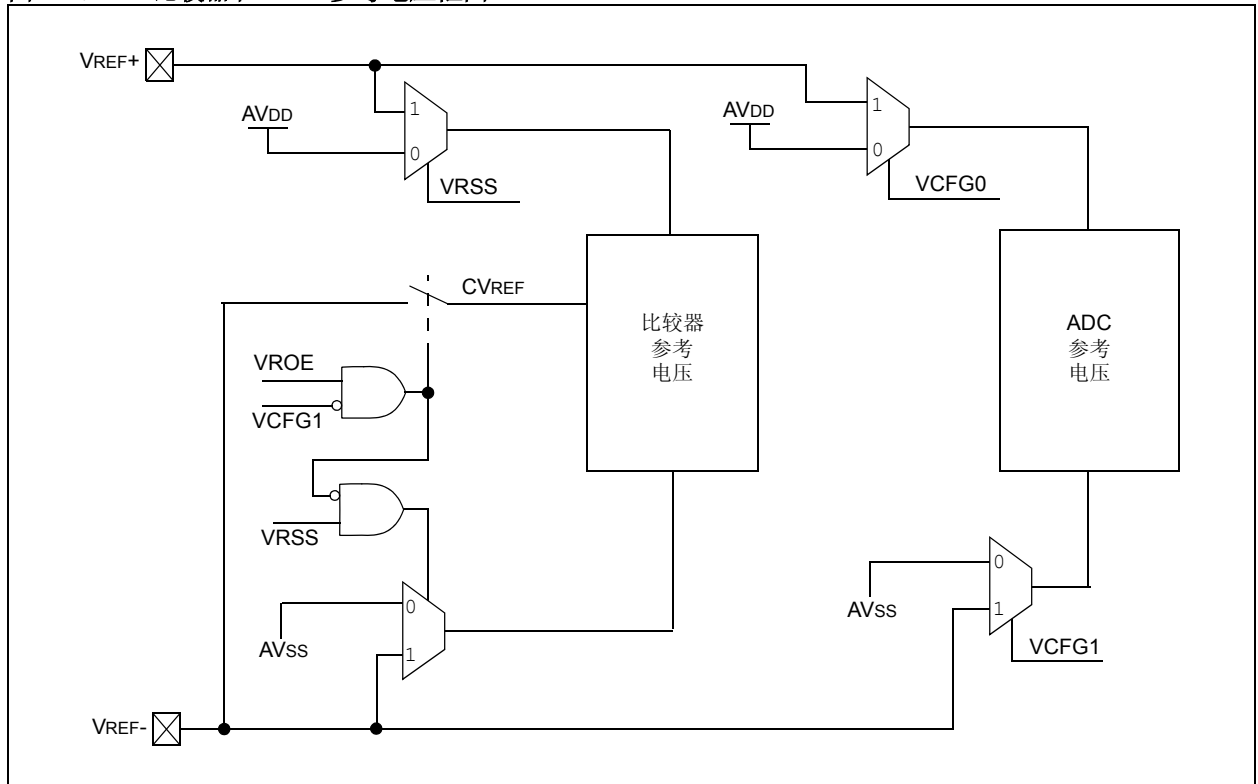


图 8-9: 比较器和 ADC 参考电压框图



PIC16F882/883/884/886/887

表 8-2: 比较器和 ADC 的参考电压优先级

RA3	RA2	比较器 参考电压 (+)	比较器 参考电压 (-)	ADC 参考电压 (+)	ADC 参考电压 (-)	CFG1	CFG0	VRSS	VROE
I/O	I/O	AVDD	AVSS	AVDD	AVSS	0	0	0	0
I/O	CVREF	AVDD	AVSS	AVDD	AVSS	0	0	0	1
VREF+	VREF-	VREF+	VREF-	AVDD	AVSS	0	0	1	0
VREF+	CVREF	VREF+	AVSS	AVDD	AVSS	0	0	1	1
VREF+	I/O	AVDD	AVSS	VREF+	AVSS	0	1	0	0
VREF+	CVREF	AVDD	AVSS	VREF+	AVSS	0	1	0	1
VREF+	VREF-	VREF+	VREF-	VREF+	AVSS	0	1	1	0
VREF+	CVREF	VREF+	AVSS	VREF+	AVSS	0	1	1	1
I/O	VREF-	AVDD	AVSS	AVDD	VREF-	1	0	0	0
I/O	VREF-	AVDD	AVSS	AVDD	VREF-	1	0	0	1
VREF+	VREF-	VREF+	VREF-	AVDD	VREF-	1	0	1	0
VREF+	VREF-	VREF+	VREF-	AVDD	VREF-	1	0	1	1
VREF+	VREF-	AVDD	AVSS	VREF+	VREF-	1	1	0	0
VREF+	VREF-	AVDD	AVSS	VREF+	VREF-	1	1	0	1
VREF+	VREF-	VREF+	VREF-	VREF+	VREF-	1	1	1	0
VREF+	VREF-	VREF+	VREF-	VREF+	VREF-	1	1	1	1

PIC16F882/883/884/886/887

寄存器 8-5: **VRCON: 参考电压控制寄存器**

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

图注:

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

- bit 7 **VREN:** 比较器 C1 参考电压使能位
 1 = CVREF 电路上电
 0 = CVREF 电路断电
- bit 6 **VREN:** 比较器 C2 参考电压使能位
 1 = CVREF 电压也是 RA2/AN2/VREF-/CVREF/C2IN+ 引脚的输出
 0 = CVREF 电压与 RA2/AN2/VREF-/CVREF/C2IN+ 引脚断开
- bit 5 **VRR:** CVREF 范围选择位
 1 = 低电压范围
 0 = 高电压范围
- bit 4 **VRSS:** 比较器 VREF 范围选择位
 1 = 比较器参考源, CVRSRC = (VREF+) - (VREF-)
 0 = 比较器参考源, CVRSRC = VDD - VSS
- bit 3-0 **VR<3:0>:** CVREF 值选择 (0 ≤ VR<3:0> ≤ 15)
 当 VRR = 1 时: CVREF = (VR<3:0>/24) * VDD
 当 VRR = 0 时: CVREF = VDD/4 + (VR<3:0>/32) * VDD

表 8-3: 与比较器和参考电压模块相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 与 BOR 时的值	所有其他复位时的值
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
ANSELH	Ñ	Ñ	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8	--11 1111	--11 1111
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0	0000 -000	0000 -000
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0	0000 -000	0000 -000
CM2CON1	MC1OUT	MC2OUT	C1RSEL	C2RSEL	—	—	T1GSS	C2SYNC	0000 --10	0000 --10
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	ULPWUIE	—	CCP2IE	0000 00-0	0000 00-0
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	ULPWUIF	—	CCP2IF	0000 00-0	0000 00-0
PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx xxxx	uuuu uuuu
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
SRCON	SR1	SR0	C1SEN	C2SEN	PULSS	PULSR	—	FVREN	0000 00-0	0000 00-0
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
VRCON	VREN	VROE	VRR	VRSS	VR3	VR2	VR1	VR0	0000 0000	0000 0000

图注: x = 未知, u = 不变, - = 未实现 (读为 0)。比较器未使用阴影单元。

PIC16F882/883/884/886/887

注:

PIC16F882/883/884/886/887

9.0 模数转换器 (ADC) 模块

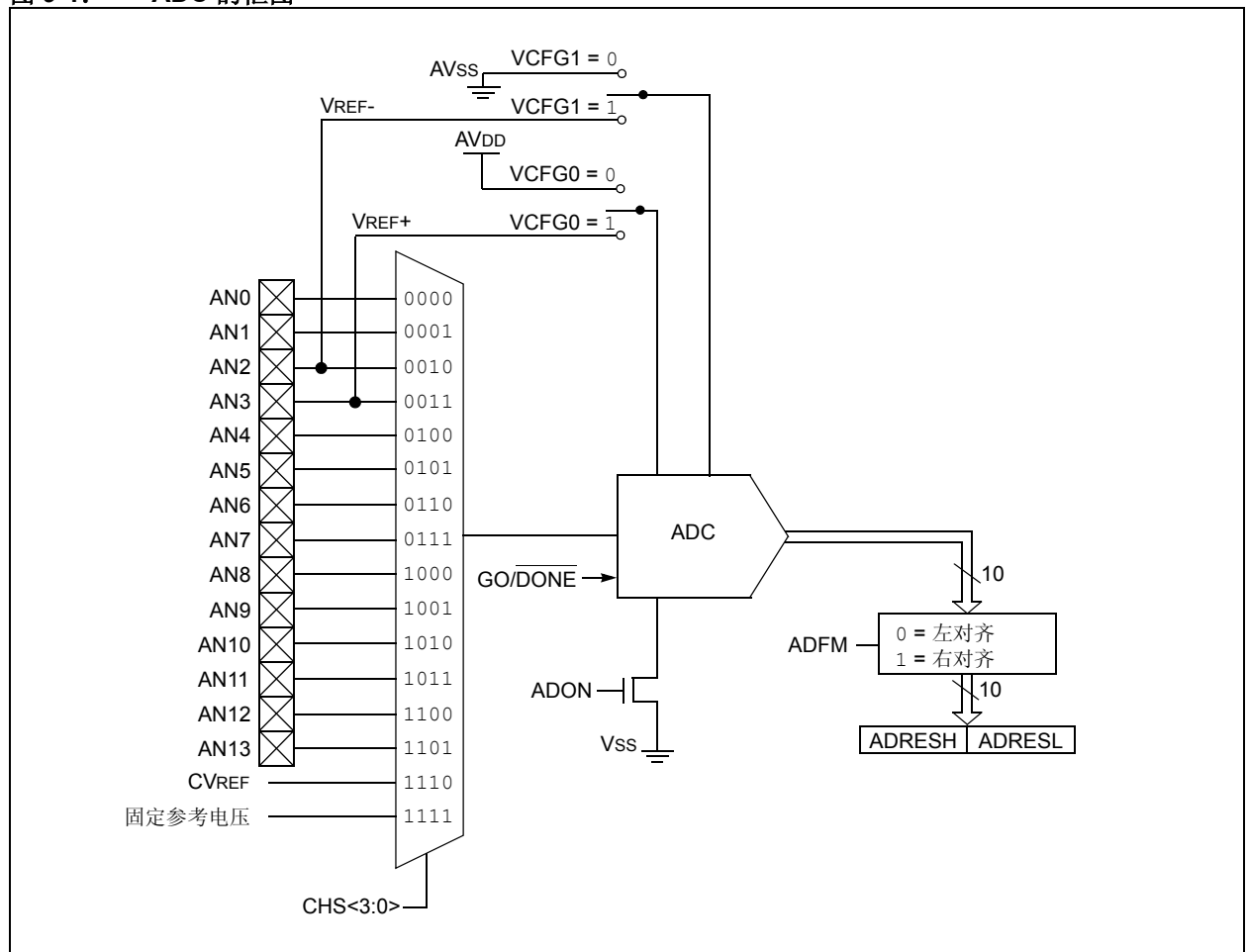
模数转换器 (ADC) 可以将模拟输入信号转换为表示该信号的一个 10 位二进制数。器件使用的模拟输入通道共用一个采样保持电路。采样保持电路的输出与模数转换器的输入相连。模数转换器采用逐次逼近法产生一个 10 位二进制结果, 并将该结果保存在 ADC 结果寄存器 (ADRESL 和 ADRESH) 中。

可用软件选择转换所使用的参考电压为 VDD 或者外部参考引脚提供的电压。

ADC 在转换完成之后可以产生一个中断。此中断可以用来唤醒处于休眠状态的器件。

图 9-1 所示为 ADC 的框图。

图 9-1: ADC 的框图



PIC16F882/883/884/886/887

9.1 ADC 配置

配置和使用 ADC 时，必须考虑如下因素：

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

9.1.1 端口配置

ADC 既可以转换模拟信号，又可以转换数字信号。当转换模拟信号时，应该通过将相应的 TRIS 和 ANSEL 位置 1，将 I/O 引脚配置为模拟输入引脚。更多信息请参见相应的端口章节。

注： 对定义为数字输入的引脚施加模拟电压可能导致输入缓冲器出现过电流。

9.1.2 通道选择

由 ADCON0 寄存器的 CHS 位决定将哪个通道连接到采样保持电路。

如果更改了通道，在下一转换开始前需要一定的延迟。更多信息请参见第 9.2 节“ADC 工作原理”。

9.1.3 ADC 参考电压

由 ADCON0 的 VCFG 位独立控制正或负参考电压。正参考电压可以是 VDD 或外部电压源。同样的，负参考电压可以是 VSS 或外部电压源。

9.1.4 转换时钟

可以通过软件设置 ADCON0 寄存器的 ADCS 位来选择转换的时钟源。有以下 4 种可能的时钟频率可供选择：

- FOSC/2
- FOSC/8
- FOSC/32
- FRC（专用内部振荡器）

完成一位转换的时间定义为 TAD。一个完整的 10 位转换需要 11 个 TAD 周期，如图 9-3 所示。

必须符合相应的 TAD 规范，才能获得正确的转换结果。请参见第 17.0 节“电气特性”中的 A/D 转换要求获得更多相关信息。表 9-1 给出了正确选择 ADC 时钟的示例。

注： 除非使用 FRC，否则系统时钟频率的任何改变都会改变 ADC 时钟的频率，从而对 ADC 转换结果产生负面影响。

PIC16F882/883/884/886/887

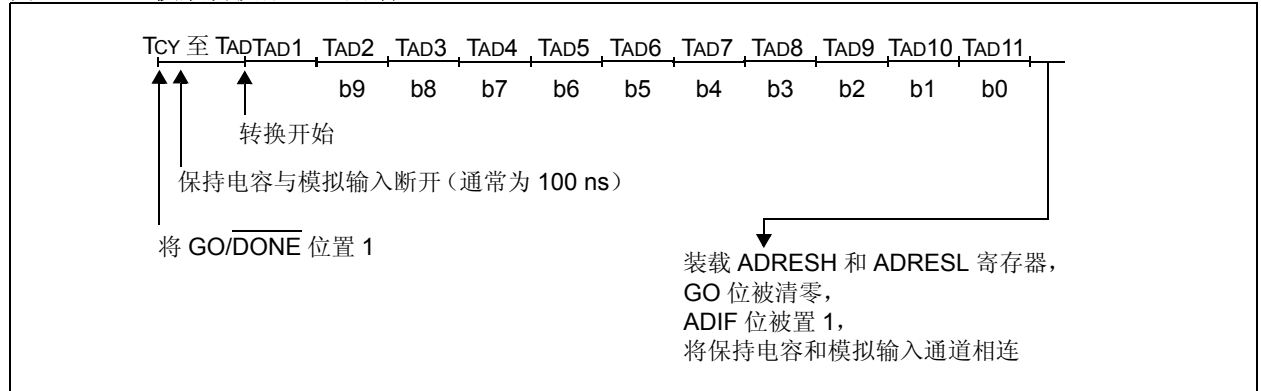
表 9-1: ADC 时钟周期 (TAD) 与器件工作频率的关系 (VDD ≥ 3.0V)

ADC 时钟周期 (TAD)		器件频率 (Fosc)			
ADC 时钟源	ADCS<2:0>	20 MHz	8 MHz	4 MHz	1 MHz
Fosc/2	000	100 ns ⁽²⁾	250 ns ⁽²⁾	500 ns ⁽²⁾	2.0 μs
Fosc/8	001	400 ns ⁽²⁾	1.0 μs ⁽²⁾	2.0 μs	8.0 μs ⁽³⁾
Fosc/32	010	1.6 μs	4.0 μs	8.0 μs ⁽³⁾	32.0 μs ⁽³⁾
FRC	x11	2-6 μs ^(1,4)	2-6 μs ^(1,4)	2-6 μs ^(1,4)	2-6 μs ^(1,4)

图注: 建议不要使用阴影单元内的值。

- 注 1:** 对于 VDD > 3.0V 的情况, FRC 时钟源的典型 TAD 时间为 4 μs。
注 2: 这些值均违反了最小 TAD 时间要求。
注 3: 为了加快转换速度, 建议选用其他时钟源。
注 4: 当器件的工作频率高于 1 MHz 时, 仅当在休眠期间进行转换时才建议使用 FRC 时钟源。

图 9-2: 模数转换器 TAD 周期



9.1.5 中断

ADC 模块允许在完成模数转换后产生一个中断。ADC 中断标志位是 PIR1 寄存器中的 ADIF 位。ADC 中断允许位是 PIE1 寄存器中的 ADIE 位。ADIF 位必须用软件清零。

注: 每次转换结束后 ADIF 位都会被置 1, 与是否允许 ADC 中断无关。

不管器件处于工作模式还是休眠模式都可以产生中断。如果器件处于休眠模式, 该中断可将器件唤醒。当将器件从休眠状态唤醒后, 总是执行 SLEEP 指令后的下一条指令。如果用户尝试使器件从休眠模式唤醒并按顺序恢复代码执行, 则必须禁止全局中断。如果允许全局中断, 程序将跳转到中断服务程序处执行。

请参见第 14.3 节“中断”了解更多信息。

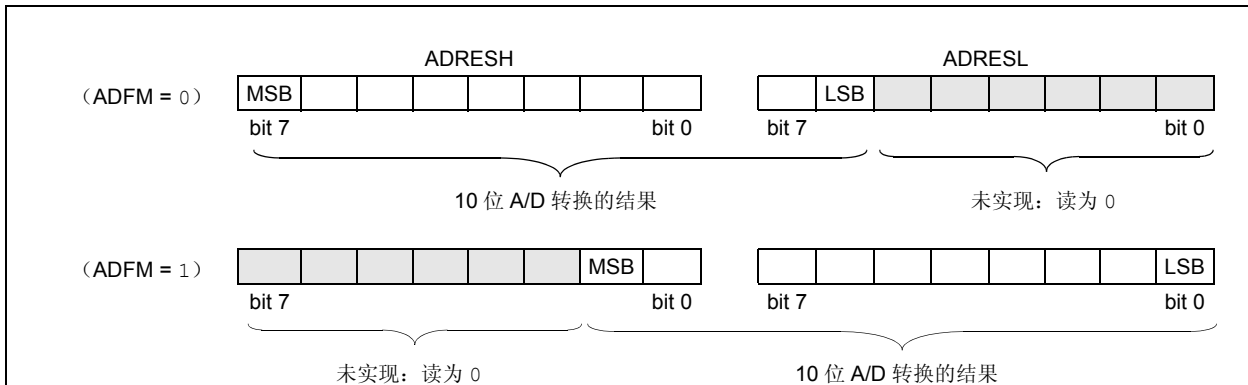
PIC16F882/883/884/886/887

9.1.6 结果格式化

10 位 A/D 转换的结果可采用两种格式：左对齐或右对齐。由 ADCON0 寄存器的 ADFM 位控制输出格式。

图 9-3 给出了这两种输出格式。

图 9-3: 10 位 A/D 转换结果的格式



9.2 ADC 工作原理

9.2.1 启动转换

要启用 ADC 模块，必须将 ADCON0 寄存器的 ADON 位置 1，将 ADCON0 寄存器的 GO/DONE 位置 1 开始模数转换。

注： 不能用开启 A/D 模块的同一指令将 GO/DONE 位置 1。请参见第 9.2.6 节“A/D 转换步骤”。

9.2.2 完成转换

当转换完成时，ADC 模块将：

- 清零 GO/DONE 位
- 将 ADIF 标志位置 1
- 用转换的新结果更新 ADRESH:ADRESL 寄存器

9.2.3 终止转换

如果必须要在转换完成前终止转换，则可用软件清零 GO/DONE 位。不会用尚未完成的模数转换结果更新 ADRESH:ADRESL 寄存器。因此，ADRESH:ADRESL 寄存器将保持上次转换所得到的值。此外，在 A/D 转换终止以后，必须经过 2 个 T_{AD} 的延时才能开始下一次采集。延时过后，将自动开始对选定通道的输入信号进行采集。

注： 器件复位将强制所有寄存器进入复位状态。因此，复位会关闭 ADC 模块并且终止任何待处理的转换。

9.2.4 ADC 在休眠模式下的工作原理

ADC 模块可以工作在休眠模式下。此操作需要将 ADC 时钟源设置为 FRC 选项。如果选择了 FRC 时钟源，ADC 在开始转换之前要多等待一个指令周期。从而允许执行 SLEEP 指令，以降低转换中的系统噪声。如果允许 ADC 中断，当转换结束时，将使器件从休眠模式唤醒。如果禁止 ADC 中断，即使 ADON 位保持置 1，则转换结束后也还是会关闭 ADC 模块。

如果 ADC 时钟源不是 FRC，即使 ADON 位仍保持置 1，执行 SLEEP 指令还是会中止当前的转换并关闭 A/D 模块。

9.2.5 特殊事件触发器

ECCP 特殊事件触发器允许在没有软件介入的情况下，周期性地对 ADC 测量。引发触发时，由硬件将 GO/DONE 位置 1，并且 Timer1 计数器将复位为零。

使用特殊事件触发器不能确保正确的 ADC 时序。由用户来确保满足 ADC 的时序要求。

请参见第 11.0 节“捕捉/比较/PWM 模块 (CCP1 和 CCP2)”获取更多相关信息。

PIC16F882/883/884/886/887

9.2.6 A/D 转换步骤

如下步骤给出了使用 ADC 进行模数转换的示例：

1. 端口配置：
 - 禁止引脚输出驱动器（见 TRIS 寄存器）
 - 将引脚配置为模拟输入引脚
2. 配置 ADC 模块：
 - 选择 ADC 转换时钟
 - 配置参考电压
 - 选择 ADC 输入通道
 - 选择结果的格式
 - 启动 ADC 模块
3. 配置 ADC 中断（可选）：
 - 清零 ADC 中断标志位
 - 允许 ADC 中断
 - 允许外设中断
 - 允许全局中断 ⁽¹⁾
4. 等待所需的采集时间 ⁽²⁾。
5. 将 GO/DONE 置 1 启动转换。
6. 由如下方法之一等待 ADC 转换结束：
 - 查询 GO/DONE 位
 - 等待 ADC 中断（允许中断）
7. 读 ADC 结果
8. 将 ADC 中断标志位清零（如果允许中断的话，需要进行此操作）。

注 1： 如果用户尝试在使器件从休眠模式唤醒后恢复顺序代码执行，则必须禁止全局中断。

2： 请参见第 9.3 节“A/D 采集要求”。

例 9-1: A/D 转换

```
;This code block configures the ADC
;for polling, Vdd and Vss as reference, Frc
clock and AN0 input.
;
;Conversion start & polling for completion
;are included.
;
BANKSEL    ADCON1        ;
MOVLW     B'10000000'    ;right justify
MOVWF    ADCON1         ;Vdd and Vss as Vref
BANKSEL    TRISA         ;
BSF      TRISA,0        ;Set RA0 to input
BANKSEL    ANSEL        ;
BSF      ANSEL,0       ;Set RA0 to analog
BANKSEL    ADCON0       ;
MOVLW     B'11000001'   ;ADC Frc clock,
MOVWF    ADCON0        ;AN0, On
CALL     SampleTime    ;Acquisiton delay
BSF     ADCON0,GO      ;Start conversion
BTFSC   ADCON0,GO     ;Is conversion done?
GOTO    $-1           ;No, test again
BANKSEL  ADRESH       ;
MOV     ADRESH,W      ;Read upper 2 bits
MOVWF  RESULTHI      ;store in GPR space
BANKSEL ADRESL       ;
MOV     ADRESL,W     ;Read lower 8 bits
MOVWF  RESULTLO     ;Store in GPR space
```

PIC16F882/883/884/886/887

9.2.7 ADC 寄存器的定义

以下寄存器用于控制 ADC 的操作。

注： ANSEL 和 ANSELH 寄存器请分别参见寄存器 3-3 和寄存器 3-4。

寄存器 9-1: ADCON0: A/D 控制寄存器 0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS1	ADCS0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0

图注:

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

bit 7-6 **ADCS<1:0>**: A/D 转换时钟选择位

00 = Fosc/2
01 = Fosc/8
10 = Fosc/32
11 = FRC (由专用的内部振荡器产生频率最高为 500 kHz 的时钟)

bit 5-2 **CHS<3:0>**: 模拟通道选择位

0000 = AN0
0001 = AN1
0010 = AN2
0011 = AN3
0100 = AN4
0101 = AN5
0110 = AN6
0111 = AN7
1000 = AN8
1001 = AN9
1010 = AN10
1011 = AN11
1100 = AN12
1101 = AN13
1110 = CVREF
1111 = 固定参考电压 (0.6V 固定参考电压)

bit 1 **GO/DONE**: A/D 转换状态位

1 = A/D 转换正在进行。将该位置 1 启动 A/D 转换。当 A/D 转换完成以后, 该位由硬件自动清零。
0 = A/D 转换完成 / 或不在进行中

bit 0 **ADON**: ADC 使能位

1 = 使能 ADC
0 = 禁止 ADC, 不消耗工作电流

PIC16F882/883/884/886/887

寄存器 9-2: **ADCON1: A/D 控制寄存器 1**

R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
ADFM	—	VCFG1	VCFG0	—	—	—	—
bit 7							bit 0

图注:

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

- bit 7 **ADFM:** A/D 转换结果格式选择位
 1 = 右对齐
 0 = 左对齐
- bit 6 **未实现:** 读为 0
- bit 5 **VCFG1:** 参考电压位
 1 = VREF- 引脚
 0 = VSS
- bit 4 **VCFG0:** 参考电压位
 1 = VREF+ 引脚
 0 = VDD
- bit 3-0 **未实现:** 读为 0

PIC16F882/883/884/886/887

9.3 A/D 采集要求

为了使 A/D 转换达到规定的准确度，必须使充电保持电容（CHOLD）充满至输入通道的电平。图 9-4 给出了模拟输入电路模型。信号源阻抗（Rs）和内部采样开关阻抗（Rss）直接影响为电容 CHOLD 充电所需要的时间。采样开关阻抗（Rss）随器件电压 VDD 不同而有所不同，请参见图 9-4。模拟信号源最大阻抗建议值为 10 kΩ。采集时间随着阻抗的降低而缩短。选择（或改变）模拟输入通道后，在启动转换前必须对通道进行采集。

可以使用公式 9-1 来计算最小采集时间。该公式假设误差为 1/2 LSB（A/D 转换需要 1024 步）。1/2 LSB 误差是 A/D 达到规定分辨率所允许的最大误差。

公式 9-1: 采集时间示例

假设 温度 = 50°C，外部阻抗为 10 kΩ，VDD 为 5V

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

可以使用如下公式估算 Tc 的值

$$V_{APPLIED} \left(1 - \frac{1}{2047} \right) = V_{CHOLD} \quad ; [1] \text{ 在 } 1/2 \text{ lsb 误差范围内对 } V_{CHOLD} \text{ 进行充电}$$

$$V_{APPLIED} \left(1 - e^{-\frac{T_C}{RC}} \right) = V_{CHOLD} \quad ; [2] \text{ 依照 } V_{APPLIED} \text{ 对 } V_{CHOLD} \text{ 进行充电}$$

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

计算 Tc:

$$\begin{aligned} T_C &= -CHOLD(RIC + RSS + RS) \ln(1/2047) \\ &= -10\text{pF}(1\text{k}\Omega + 7\text{k}\Omega + 10\text{k}\Omega) \ln(0.0004885) \\ &= 1.37 \mu\text{s} \end{aligned}$$

因此

$$\begin{aligned} T_{ACQ} &= 2 \mu\text{s} + 1.37 \mu\text{s} + [(50^\circ\text{C} - 25^\circ\text{C}) (0.05 \mu\text{s}/^\circ\text{C})] \\ &= 4.67 \mu\text{s} \end{aligned}$$

注 1: 由于可以将参考电压（VREF）消掉，因此它对公式的结果不会产生影响。

2: 在每次转换后，充电保持电容（CHOLD）并不放电。

3: 模拟信号的最大阻抗建议值为 10 kΩ。这样要求是为了符合引脚泄漏电流规范。

PIC16F882/883/884/886/887

图 9-4: 模拟输入模型

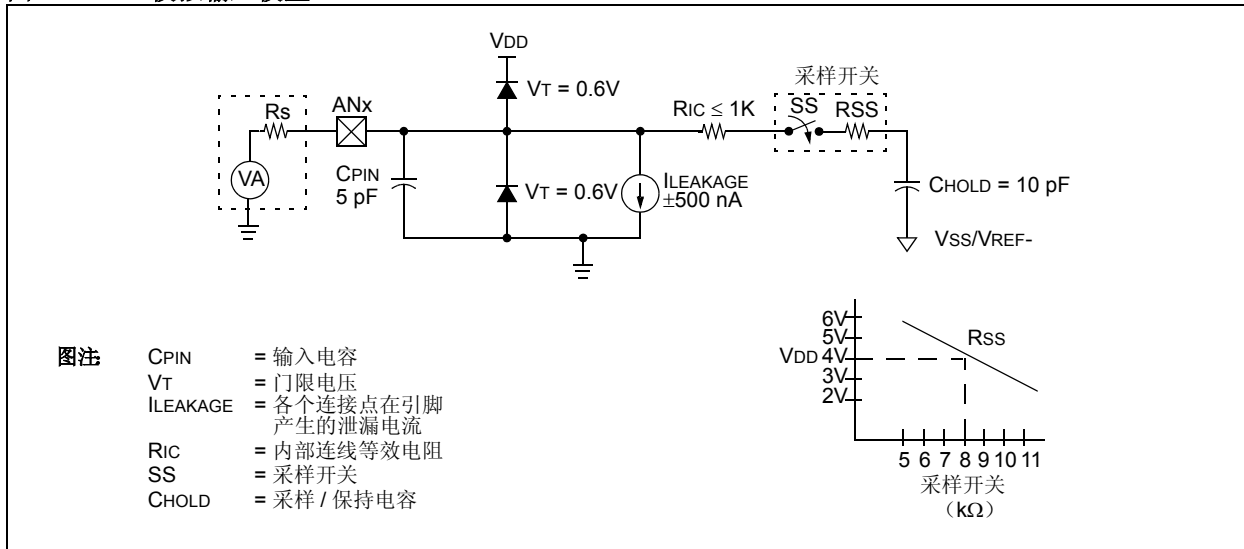
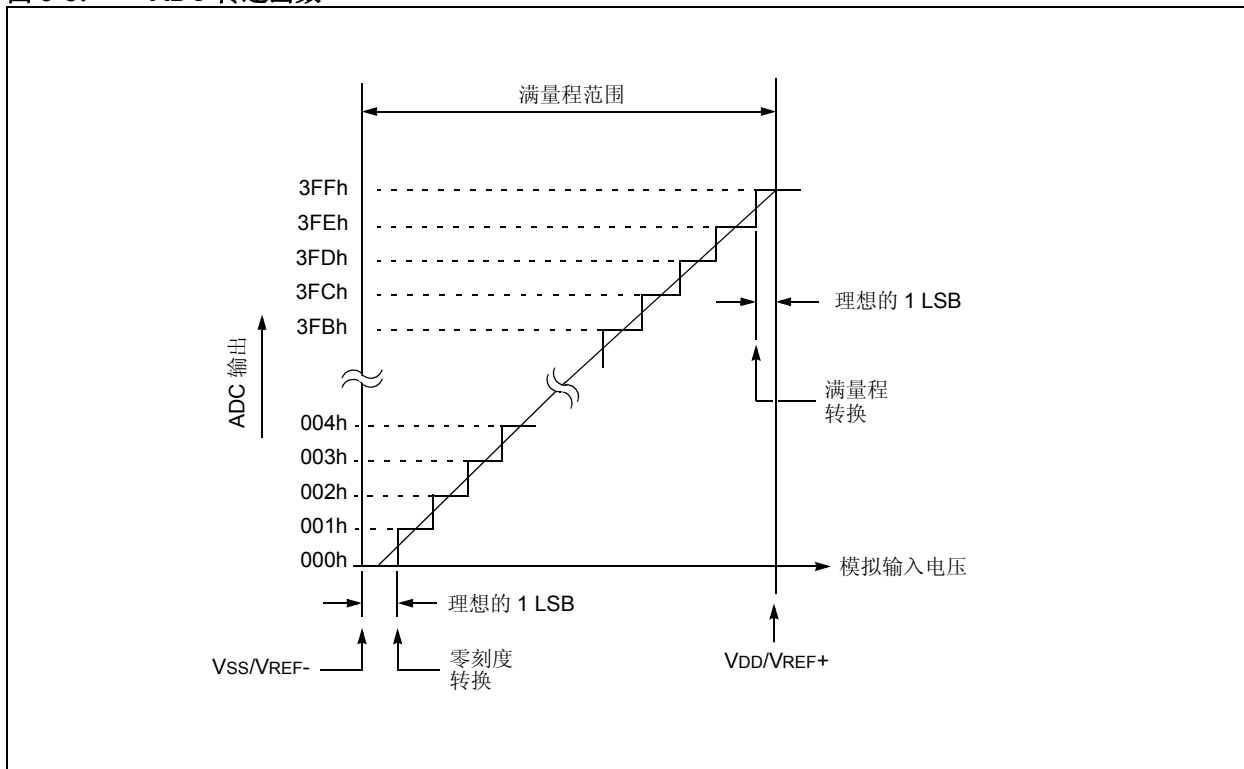


图 9-5: ADC 传递函数



PIC16F882/883/884/886/887

表 9-2: 与 ADC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
ADCON0	ADCS1	ADCS0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	0000 0000	0000 0000
ADCON1	ADFM	—	VCFG1	VCFG0	—	—	—	—	0-00 ----	-000 ----
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
ANSELH	—	—	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8	--11 1111	--11 1111
ADRESH	A/D 结果寄存器的高字节								xxxx xxxx	uuuu uuuu
ADRESL	A/D 结果寄存器的低字节								xxxx xxxx	uuuu uuuu
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx xxxx	uuuu uuuu
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
PORTE	—	—	—	—	RE3	RE2	RE1	RE0	---- xxxx	---- uuuu
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 111
TRISE	—	—	—	—	TRISE3	TRISE2	TRISE1	TRISE0	---- 1111	---- 111

图注: x = 未知, u = 不变, - = 未实现 (读为 0)。ADC 模块未使用阴影单元。

PIC16F882/883/884/886/887

注:

PIC16F882/883/884/886/887

10.0 数据 EEPROM 和闪存程序存储器控制

数据 EEPROM 和闪存程序存储器在正常工作状态下（整个 VDD 范围内）是可读写的。这些存储器并不直接映射到寄存器文件空间，而是通过特殊功能寄存器（SFR）对其进行间接寻址。共有 6 个 SFR 寄存器用于访问这些存储器：

- EECON1
- EECON2
- EEDAT
- EEDATH
- EEADR
- EEADRH（其 bit 4 仅供 PIC16F886/PIC16F887 使用）

当与数据存储器模块接口时，EEDATL 寄存器存放 8 位读写的数据，而 EEADR 寄存器存放被访问的 EEDAT 单元的地址。该系列中的器件具有 256 字节的数据 EEPROM，地址范围为 0h 到 0FFh。

访问 PIC16F886/PIC16F887 器件的程序存储器时，EEDAT 和 EEDATH 寄存器形成一个双字节字用于保存要读/写的 14 位数据，EEADR 和 EEADRH 寄存器组成一个双字节字用于保存待读取的 12 位 EEPROM 单元地址。PIC16F882 器件具有 2K 字的程序 EEPROM，地址范围从 0h 至 07FFh。PIC16F883/PIC16F884 器件具有 4K 字的程序 EEPROM，地址范围从 0h 到 0FFFh。程序存储器允许以字为单位读取。

EEPROM 数据存储器允许字节读写。字节写操作可自动擦除目标单元并写入新数据（在写入前擦除）。

写入时间由片上定时器控制。写入和擦除电压是由片上电荷泵产生的，此电荷泵额定工作在器件的电压范围内，用于进行字节或字操作。

器件能否对程序存储器的特定块执行写操作取决于配置字寄存器 2 的闪存程序存储器自写使能位 WRT<1:0> 的设置。但是，始终允许对程序存储器执行读操作。

当器件受代码保护时，CPU 仍可继续读写数据 EEPROM 和闪存程序存储器。代码保护时，器件编程器将不再能访问数据或程序存储器。

10.1 EEADR 和 EEADRH 寄存器

EEADRL 和 EEADRH 寄存器能寻址最大 256 字节的数据 EEPROM 或最大 8K 字的程序 EEPROM。

当选择程序地址值时，地址的高字节被写入 EEADRH 寄存器而低字节被写入 EEADR 寄存器。当选择数据地址值时，只将地址的低字节写入 EEADR 寄存器。

10.1.1 EECON1 和 EECON2 寄存器

EECON1 是访问 EE 存储器的控制寄存器。

控制位 EEPGD 决定访问的是程序存储器还是数据存储器。该位被清零时，和复位时一样，任何后续操作都将针对数据存储器进行。该位置 1 时，任何后续操作都将针对程序存储器进行。程序存储器是只读的。

控制位 RD 和 WR 分别启动读和写。用软件只能将这些位置 1 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，从而可避免意外地过早终止写操作。

当 WREN 置 1 时，允许对数据 EEPROM 执行写操作。上电时，WREN 位被清零。当正常的写入操作被 MCLR 复位或 WDT 超时复位中断时，WRERR 位会置 1。在这些情况下，复位后用户可以检查 WRERR 位并重写相应的单元。

当写操作完成时 PIR2 寄存器中的中断标志位 EEIF 被置 1。此标志位必须用软件清零。

EECON2 不是物理寄存器。读 EECON2 得到的是全 0。EECON2 寄存器仅在执行数据 EEPROM 写序列时使用。

PIC16F882/883/884/886/887

寄存器 10-1: EEDAT: EEPROM 数据寄存器

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

图注:

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

bit 7-0 **EEDAT<7:0>**: 要从数据 EEPROM 中读取或向数据 EEPROM 写入的地址的低 8 位, 或者要从程序存储器中读取的地址的低 8 位

寄存器 10-2: EEADR: EEPROM 地址寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0
bit 7							bit 0

图注:

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

bit 7-0 **EEADR<7:0>**: 指定 EEPROM 读 / 写操作⁽¹⁾ 的地址的低 8 位, 或程序存储器读操作的地址的低 8 位

寄存器 10-3: EEDATH: EEPROM 数据的高字节寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0
bit 7							bit 0

图注:

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

bit 7-6 **未实现**: 读为 0

bit 5-0 **EEDATH<5:0>**: 从程序存储器读出的数据的高 6 位

寄存器 10-4: EEADRH: EEPROM 地址的高字节寄存器

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	EEADRH4 ⁽¹⁾	EEADRH3	EEADRH2	EEADRH1	EEADRH0
bit 7							bit 0

图注:

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

bit 7-5 **未实现**: 读为 0

bit 4-0 **EEADRH<4:0>**: 指定程序存储器读操作的高 4 位地址或高 3 位。

注 1: 仅用于 PIC16F886/PIC16F887。

PIC16F882/883/884/886/887

寄存器 10-5: EECON1: EEPROM 控制寄存器

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

图注:

S = 只能被置 1 的位

R = 可读位

-n = 上电复位时的值

W = 可写位

1 = 置 1

U = 未实现位, 读为 0

0 = 清零

x = 未知

- bit 7 **EEPGD:** 程序 / 数据 EEPROM 选择位
 1 = 访问程序存储器
 0 = 访问数据存储器
- bit 6-4 **未实现:** 读为 0
- bit 3 **WRERR:** EEPROM 错误标志位
 1 = 写操作过早终止 (正常工作期间的任何 $\overline{\text{MCLR}}$ 复位、WDT 复位或欠压复位)
 0 = 写操作完成
- bit 2 **WREN:** EEPROM 写使能位
 1 = 允许写周期
 0 = 禁止写入数据 EEPROM
- bit 1 **WR:** 写控制位
 1 = 启动写周期 (写操作一旦完成, 由硬件清零该位。用软件只能将 WR 位置 1, 但不能清零。)
 0 = 数据 EEPROM 写周期完成
- bit 0 **RD:** 读控制位
 1 = 启动存储器读操作 (由硬件清零 RD。用软件只能将 RD 位置 1, 但不能清零。)
 0 = 不启动存储器读操作

PIC16F882/883/884/886/887

10.1.2 读数据 EEPROM 存储器

要读取数据存储器单元，用户必须将地址写入 EEADR 寄存器，清零 EECON1 寄存器的 EEPGD 控制位，然后将控制位 RD 置 1。在紧接着的下一个周期，EEDAT 寄存器中就有数据了，因此该数据可由下一条指令读取。EEDAT 将保存此值直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 10-1: 读数据 EEPROM

```
BSF    STATUS,RP1    ;
BCF    STATUS,RP0    ; Bank 2
MOVF   DATA_EE_ADDR,W ; Data Memory
MOVWF  EEADRL        ; Address to read
BSF    STATUS,RP0    ; Bank 3
BCF    EECON1,EEPGD  ; Point to Data
                    ; memory
BSF    EECON1,RD     ; EE Read
BCF    STATUS,RP0    ; Bank 2
MOVF   EEDATL,W      ; W = EEDATA
```

10.1.3 写数据 EEPROM 存储器

要写 EEPROM 数据存储器单元，用户应首先将该单元的地址写入 EEADR 寄存器并将数据写入 EEDAT 寄存器。然后用户必须按特定顺序开始写入每个字节。

如果没有完全按照下面的指令顺序（即首先将 55h 写入 EECON2，随后将 AAh 写入 EECON2，最后将 WR 位置 1）写每个字节，将不会启动写操作。在该代码段中应禁止中断。

此外，必须将 EECON1 中的 WREN 位置 1 以能使写操作。这种机制可防止由于代码执行错误（异常）（即程序跑飞）导致误写 EEPROM。在不更新 EEPROM 时，用户应该始终保持 WREN 位清零。WREN 位不能被硬件清零。

一个写过程启动后，将 WREN 位清零将不会影响此写周期。除非 WREN 位置 1，否则 WR 位将无法置 1。

写周期完成时，WR 位由硬件清零并且 EE 写完成中断标志位（EEIF）置 1。用户可以允许此中断或查询此位。EEIF 必须用软件清零。

例 10-2: 写数据 EEPROM 存储器

```
BANKSEL EEADR        ;
MOVLW  DATA_EE_ADDR ;
MOVWF  EEADR         ;Data Memory Address to write
MOVLW  DATA_EE_DATA ;
MOVWF  EEDAT        ;Data Memory Value to write
BANKSEL EECON1       ;
BCF    EECON1, EEPGD ;Point to DATA memory
BSF    EECON1, WREN  ;Enable writes

BCF    INTCON, GIE   ;Disable INTs.
BTFSK  INTCON, GIE   ;SEE AN576
GOTO   $-2

MOVWF  55h           ;
MOVWF  EECON2        ;Write 55h
MOVWF  AAh           ;
MOVWF  EECON2        ;Write AAh
BSF    EECON1, WR    ;Set WR bit to begin write
BSF    INTCON, GIE   ;Enable INTs.

SLEEP                ;Wait for interrupt to signal write complete
BCF    EECON1, WREN  ;Disable writes
BCF    STATUS, RP0   ;Bank 0
BCF    STATUS, RP1
```

必需的
序列

PIC16F882/883/884/886/887

10.1.4 读闪存程序存储器

要读取程序存储器单元，用户必须将地址的高位和低位分别写入 EEDR 和 EEADRH 寄存器，将 EECON1 寄存器的 EEPGD 位置 1，然后将控制位 RD 置 1。一旦设置好读控制位，闪存程序存储器控制器将使用第二个指令周期来读数据。这会导致紧随“BSF EECON1,RD”指令的第二条指令被忽略。在紧接着的下一个周期 EEDAT 和 EEDATH 寄存器中就有数据了，因此在随后的指令中将该数据读作两个字节。

EEDATL 和 EEDATH 寄存器将保存此值直至下一次用户向该单元读取或写入数据时为止。

- 注 1:** 程序存储器读操作后的两条指令必须为 NOP。这可阻止用户在 RD 位置 1 后的下一条指令执行双周期指令。
- 2:** 当 EEPGD = 1 时如果 WR 位置 1，它会立即复位为 0，而不执行任何操作。

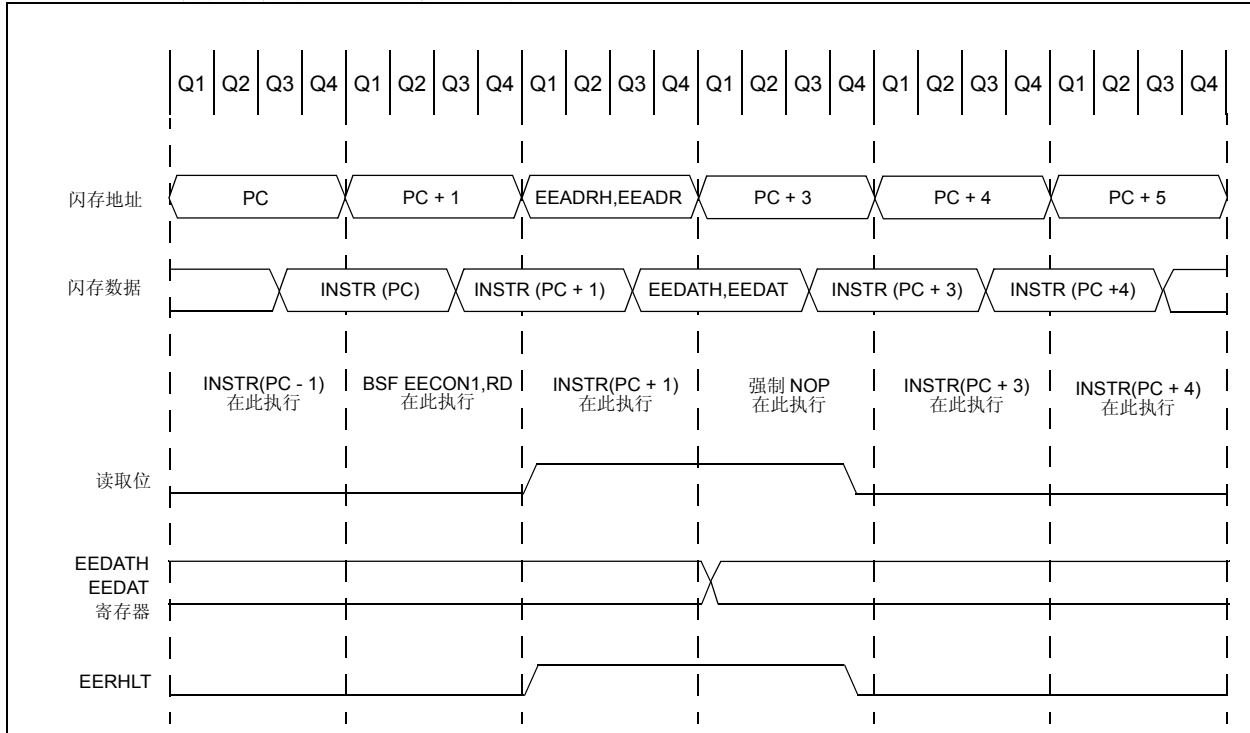
例 10-3: 读闪存程序存储器

```
BANKSEL EEADR          ;
MOVLW  MS_PROG_EE_ADDR ;
MOVWF  EEADRH          ;MS Byte of Program Address to read
MOVLW  LS_PROG_EE_ADDR ;
MOVWF  EEADR           ;LS Byte of Program Address to read
BANKSEL EECON1         ;
BSF    EECON1, EEPGD   ;Point to PROGRAM memory
BSF    EECON1, RD      ;EE Read
                               ;First instruction after BSF EECON1,RD executes normally
NOP
NOP                               ;Any instructions here are ignored as program
                               ;memory is read in second cycle after BSF EECON1,RD
;
BANKSEL EEDAT          ;
MOVF   EEDAT, W        ;W = LS Byte of Program Memory
MOVWF  LOWPMBYTE       ;
MOVF   EEDATH, W       ;W = MS Byte of Program EEDAT
MOVWF  HIGHPMBYTE      ;
BCF    STATUS, RP1     ;Bank 0
```

必需的
序列

PIC16F882/883/884/886/887

图 10-1: 闪存程序存储器读周期执行时序



10.2 写闪存程序存储器

只有当目标地址位于未被写保护的存储器段时（如配置字寄存器 2 中对 WRT<1:0> 位的定义），才能对闪存存储器执行写操作。闪存程序存储器只能以每 8 个字为一块的块为单位写入（对于存储器为 4K 的器件以 4 字块为单位）。请参见图 10-2 和图 10-3 获得更多相关信息。存储器的块由具有连续地址的 8 个字组成，由 EEADR<2:0> = 000 定义地址的低边界。所有程序存储器的块写操作都是通过先擦除 16 字，然后写入 8 字完成的。写操作是边界对齐的，且不产生跨边界操作。

要对程序存储器写入数据，必须首先将数据载入缓冲寄存器（见图 10-2）。这是通过将目标地址写入 EEADR 和 EEADRH 寄存器，再将数据写入 EEDATA 和 EEDATH 寄存器完成的。然后，执行如下事件序列：

1. 将 EECON1 寄存器的 EEPGD 控制位置 1。
2. 先后将 55h 和 AAh 写入 EECON2（闪存编程序列）。
3. 将 EECON1 寄存器的 WR 控制位置 1。

所有 8 个缓冲寄存器单元均必须写入正确数据。如果写入 8 字存储块的字不满 8 个，则必须对未被写入的程序存储单元执行读操作。此操作从未被写入的程序存储单元取出数据，并将其装入 EEDATA 和 EEDATH 寄存器。然后必须执行将数据传送到缓冲寄存器的事件序列。

要将数据从缓冲寄存器传送到程序存储器，EEADR 和 EEADRH 必须指向 8 字存储块的最后一个单元（EEADR<2:0> = 111）。然后必须执行如下事件序列：

1. 将 EECON1 寄存器的 EEPGD 控制位置 1。
2. 先后将 55h 和 AAh 写入 EECON2（闪存编程序列）。
3. 将 EECON1 寄存器的 WR 控制位置 1，开始写操作。

用户必须按照同样的特定顺序依次开始写入程序存储块的每个字，写入每个程序字顺序为 000、001、010、011、100、101、110 和 111。当最后一个字（EEADR<2:0> = 111）的写操作完成时，将自动擦除一个 16 字的块并将 8 字缓冲寄存器的内容写入程序存储器。

执行了 BSF EECON1, WR 指令之后，处理器需要 2 个指令周期以设置擦除 / 写操作。用户必须在将 WR 位置 1 的指令后放置两条 NOP 指令。由于数据被写入缓冲寄存器，因而可立即实现第一个 7 字块的写操作。只有在擦除周期时（即 16 字块的最后一个字被擦除时），处理器才会使内部操作暂停 4 ms（典型值）时间。因为时钟和外设仍继续工作，所以这并不是休眠模式。8 字写周期结束后，处理器将从 EECON1 写指令后的第三条指令恢复工作。必须对高 8 字重复执行上述事件序列。

PIC16F882/883/884/886/887

图 10-2: 2K 和 4K 闪存程序存储器块写操作

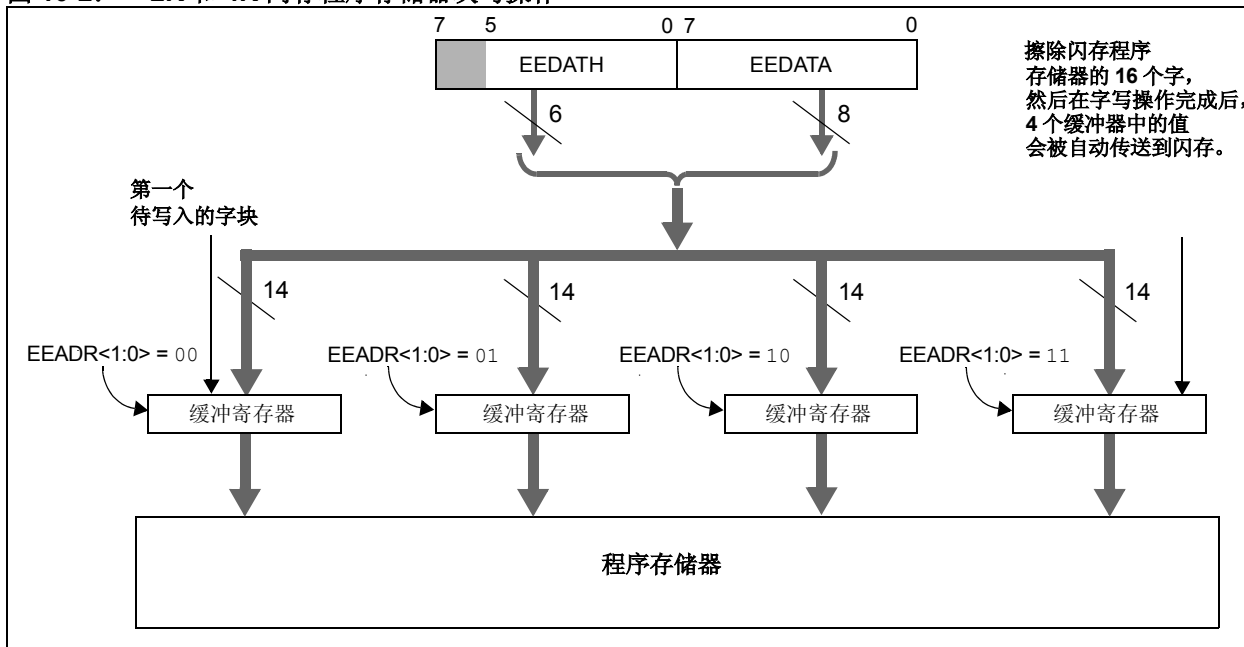
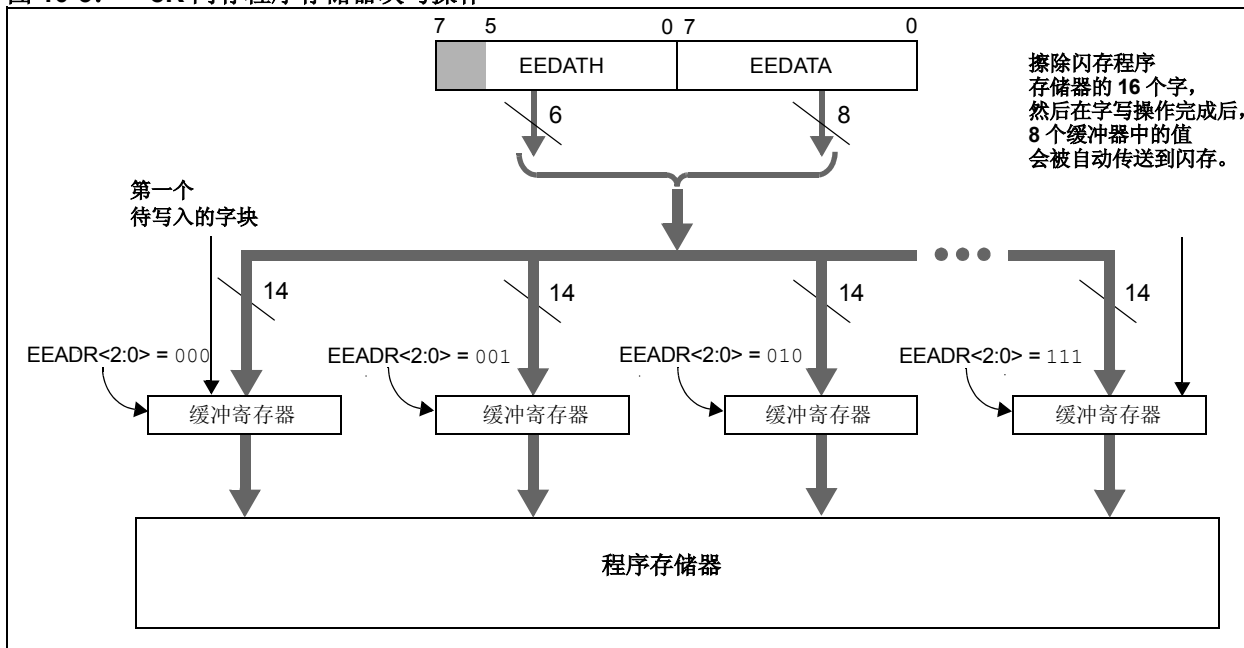


图 10-3: 8K 闪存程序存储器块写操作



PIC16F882/883/884/886/887

例 10-4 给出了完整的 8 字写序列的示例。将起始地址装入 EEADRH 和 EEADR 这对寄存器，使用间接寻址装入 8 字数据。

例 10-4: 写闪存程序存储器

```
; This write routine assumes the following:
;
; 1. A valid starting address (the least significant bits = '00') is loaded in ADDRH:ADDRL
; 2. The 8 bytes of data are loaded, starting at the address in DATADDR
; 3. ADDRH, ADDRL and DATADDR are all located in shared data memory 0x70 - 0x7f
;
      BSF    STATUS,RP1      ;
      BCF    STATUS,RP0      ; Bank 2
      MOVF   ADDRH,W         ; Load initial address
      MOVWF  EEADRH          ;
      MOVF   ADDRL,W         ;
      MOVWF  EEADR           ;
      MOVF   DATAADDR,W     ; Load initial data address
      MOVWF  FSR             ;
LOOP   MOVF   INDF,W         ; Load first data byte into lower
      MOVWF  EEDATA          ;
      INCF   FSR,F           ; Next byte
      MOVF   INDF,W         ; Load second data byte into upper
      MOVWF  EEDATH          ;
      INCF   FSR,F           ;
      BSF    STATUS,RP0      ; Bank 3
      BSF    EECON1,EEPGD     ; Point to program memory
      BSF    EECON1,WREN      ; Enable writes
      BCF    INTCON,GIE       ; Disable interrupts (if using)
      BTFSC  INTCON,GIE       ; See AN576
      GOTO   1-2
      MOVLW  55h              ; Start of required write sequence:
      MOVWF  EECON2           ; Write 55h
      MOVLW  AAh              ;
      MOVWF  EECON2           ; Write AAh
      BSF    EECON1,WR        ; Set WR bit to begin write
      NOP                    ; Any instructions here are ignored as processor
      NOP                    ; halts to begin write sequence
      NOP                    ; processor will stop here and wait for write complete
      NOP                    ; after write processor continues with 3rd instruction
      BCF    EECON1,WREN      ; Disable writes
      BSF    INTCON,GIE       ; Enable interrupts (if using)
      BCF    STATUS,RP0      ; Bank 2
      INCF   EEADR,F          ; Increment address
      MOVF   EEADR,W         ; Check if lower two bits of address are '00'
      ANDLW  0x07             ; Indicates when four words have been programmed
      XORLW  0x07             ;
      BTFSC  STATUS,Z         ; Exit if more than eight words,
      GOTO   LOOP            ; Continue if less than eight words
```

必需的
序列

PIC16F882/883/884/886/887

10.3 写校验

根据具体的应用，好的编程习惯一般要求将写入数据 EEPROM 的值对照期望值进行校验（见例 10-5）。

例 10-5: 写校验

```
BANKSEL EEDAT      ;
MOVWF  EEDAT, W    ;EEDAT not changed
                    ;from previous write

BANKSEL EECON1     ;
BSF    EECON1, RD   ;YES, Read the
                    ;value written

BANKSEL EEDAT      ;
XORWF  EEDAT, W    ;
BTFS   STATUS, Z    ;Is data the same
GOTO   WRITE_ERR   ;No, handle error
:      ;Yes, continue
BCF    STATUS, RP1  ;Bank 0
```

当数据存储器被代码保护时，只有 CPU 可以对数据 EEPROM 执行读/写操作。对数据存储器进行代码保护的同时，建议用户也对程序存储器进行代码保护。这将防止有人通过在已有代码上写入零（这将作为 NOP 执行），来访问一个在未使用的程序存储器中编写的额外程序，以达到输出数据存储器内容的目的。将程序存储器中未使用的地址单元编程为 0 有助于防止数据存储器的代码保护受到破坏。

10.3.1 使用数据 EEPROM

数据 EEPROM 是具有高耐用性，可字节寻址的阵列，并针对存储频繁更改的信息（例如，程序变量或其他经常更新的数据）进行了优化。当某部分变量发生频繁的更改，而其他部分的变量不变时，则可能会发生超出 EEPROM 的总写周期数（规范 D124），但却没有超出单个字节的总写周期数（规范 D120 和 D120A）的情况。如果出现上述情况，则必须执行阵列刷新。正因如此，不常修改的变量（例如常数、ID 和校准值等）应该存储在闪存程序存储器中。

10.4 避免误写的保护

有些情况下，用户可能不希望向数据 EEPROM 存储器写入数据。为防止误写 EEPROM，芯片内嵌了各种保护机制。上电时清零 WREN 位。而且，上电延时定时器（延迟时间为 64 ms）会防止对 EEPROM 执行写操作。

写操作的启动序列以及 WREN 位将共同防止在以下情况下发生误写操作：

- 欠压
- 电源毛刺
- 软件故障

10.5 代码保护下的数据 EEPROM 操作

通过编程将配置寄存器 1（寄存器 14-1）中的 CPD 位清零，可以将数据存储器代码保护。

PIC16F882/883/884/886/887

表 10-1: 与数据 EEPROM 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 与 BOR 时的值	所有其他复位时的值
EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	0--- q000
EECON2	EEPROM 控制寄存器 2 (不是物理寄存器)								---- ----	---- ----
EEADR	EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	0000 0000	0000 0000
EEADRH	—	—	—	EEADRH4 ⁽¹⁾	EEADRH3	EEADRH2	EEADRH1	EEADRH0	---0 0000	---0 0000
EEDATA	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0	0000 0000	0000 0000
EEDATH	—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0	--00 0000	--00 0000
INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000x
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	ULPWUIE	—	CCP2IE	0000 00-0	0000 00-0
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	ULPWUIF	—	CCP2IF	0000 00-0	0000 00-0

图注: x = 未知, u = 不变, - = 未实现 (读为 0), q = 取值视条件而定。
数据 EEPROM 模块未使用阴影单元。

注 1: 仅用于 PIC16F886/PIC16F887。

PIC16F882/883/884/886/887

注:

PIC16F882/883/884/886/887

11.0 捕捉 / 比较 / PWM 模块 (CCP1 和 CCP2)

该器件包含一个增强型捕捉 / 比较 / PWM (CCP1) 和捕捉 / 比较 / PWM 模块 (CCP2)。CCP1 和 CCP2 模块的操作基本相同。不同之处在于只有 CCP1 模块才具有增强型 PWM 功能。请参见第 11.6 节 “PWM (增强模式)” 了解更多信息。

注： 本文档中的 CCPRx 和 CCPx 分别指 CCPR1 或 CCPR2 和 CCP1 或 CCP2。

PIC16F882/883/884/886/887

11.1 增强型捕捉 / 比较 PWM (CCP1)

增强型捕捉 / 比较 / PWM 模块是允许用户定时和控制不同事件的外设。在捕捉模式下，该外设能对事件的持续时间计时。捕捉模式允许用户在预先确定的定时时间结束后触发一个外部事件。PWM 模式可产生频率和占空比都可变化的脉宽调制信号。

表 11-1 给出了 ECCP 模块所需的定时器资源。

表 11-1: ECCP 模式所需的定时器资源

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

寄存器 11-1: CCP1CON: 增强型 CCP1 控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = 上电复位是的值	1 = 置 1	0 = 清零
		x = 未知

bit 7-6 **P1M<1:0>**: PWM 输出配置位
 如果 CCP1M<3:2> = 00、01 或 10:
 xx = 指定 P1A 为捕捉 / 比较输入引脚, P1B、P1C 和 P1D 为端口引脚
 如果 CCP1M<3:2> = 11:
 00 = 单输出: P1A 调制输出; P1B、P1C 和 P1D 被分配为端口引脚
 01 = 全桥正向输出: P1D 调制输出; P1A 有效; P1B 和 P1C 无效
 10 = 半桥输出: P1A 和 P1B 为带死区控制的调制输出; P1C 和 P1D 被分配为端口引脚
 11 = 全桥反向输出: P1B 调制输出; P1C 有效; P1A 和 P1D 无效

bit 5-4 **DC1B<1:0>**: PWM 占空比的低两位
捕捉模式:
 未使用。
比较模式:
 未使用。
PWM 模式:
 这两位是 10 位 PWM 占空比的低 2 位。占空比的高 8 位在 CCPR1L 中。

bit 3-0 **CCP1M<3:0>**: ECCP 模式选择位
 0000 = 捕捉 / 比较 / PWM 关闭 (复位 ECCP 模块)
 0001 = 未使用 (保留)
 0010 = 比较模式, 匹配时输出电平翻转 (CCP1IF 置 1)
 0011 = 未使用 (保留)
 0100 = 捕捉模式, 在每个下降沿发生捕捉
 0101 = 捕捉模式, 在每个上升沿发生捕捉
 0110 = 捕捉模式, 每 4 个上升沿发生捕捉
 0111 = 捕捉模式, 每 16 个上升沿发生捕捉
 1000 = 比较模式, 比较匹配时输出高电平 (CCP1IF 置 1)
 1001 = 比较模式, 比较匹配时输出低电平 (CCP1IF 置 1)
 1010 = 比较模式, 比较匹配时产生软件中断 (CCP1IF 位置 1, CCP1 引脚不受影响)
 1011 = 比较模式, 触发特殊事件 (CCP1IF 位置 1, CCP1 复位 TMR1 或 TMR2)
 1100 = PWM 模式; P1A 和 P1C 为高电平有效; P1B 和 P1D 也为高电平有效
 1101 = PWM 模式; P1A 和 P1C 为高电平有效; P1B 和 P1D 为低电平有效
 1110 = PWM 模式; P1A 和 P1C 为低电平有效; P1B 和 P1D 为高电平有效
 1111 = PWM 模式; P1A 和 P1C 为低电平有效; P1B 和 P1D 也为低电平有效

PIC16F882/883/884/886/887

11.2 捕捉 / 比较 / PWM (CCP2)

捕捉 / 比较 / PWM 模块是允许用户定时和控制不同事件的外设。在捕捉模式下，该外设能对事件的持续时间计时。捕捉模式允许用户在预先确定的定时时间结束后触发一个外部事件。PWM 模式可产生频率和占空比都可变化的脉宽调制信号。

表 11-2 给出了该模块使用的定时器资源。

有关 CCP 模块的更多信息，请参见应用笔记 AN594 “Using the CCP Modules” (DS00594)。

表 11-2: CCP 模式所需的定时器资源

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

寄存器 11-2: CCP2CON: CCP2 控制寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位，读为 0
 -n = 上电复位是的值 1 = 置 1 0 = 清零 x = 未知

bit 7-6 **未实现:** 读为 0

bit 5-4 **DC2B<1:0>:** PWM 占空比的低两位

捕捉模式:

未使用。

比较模式:

未使用。

PWM 模式:

这两位是 10 位 PWM 占空比的低 2 位。占空比的高 8 位在 CCPR2L 中。

bit 3-0 **CCP2M<3:0>:** CCP2 模式选择位

0000 = 捕捉 / 比较 / PWM 关闭 (复位 CCP2 模块)

0001 = 未使用 (保留)

0010 = 未使用 (保留)

0011 = 未使用 (保留)

0100 = 捕捉模式，在每个下降沿发生捕捉

0101 = 捕捉模式，在每个上升沿发生捕捉

0110 = 捕捉模式，每 4 个上升沿发生捕捉

0111 = 捕捉模式，每 16 个上升沿发生捕捉

1000 = 比较模式，比较匹配时输出高电平 (CCP2IF 置 1)

1001 = 比较模式，比较匹配时输出低电平 (CCP2IF 置 1)

1010 = 比较模式，比较匹配时产生软件中断 (CCP2IF 位置 1, CCP2 引脚不受影响)

1011 = 比较模式，触发特殊事件 (CCP2IF 位置 1, TMR1 复位，如果使能 ADC 模块还将启动 A/D 转换。

CCP2 引脚不受影响)

11xx = PWM 模式

PIC16F882/883/884/886/887

11.3 捕捉模式

在捕捉模式下，当对应的 CCPx 引脚发生事件时，CCPRxH:CCPRxL 这对寄存器捕捉 TMR1 寄存器的 16 位值。触发捕捉的事件可被定义为以下四者之一，并且由 CCP1CON 寄存器中的 CCP1M<3:0> 位配置：

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

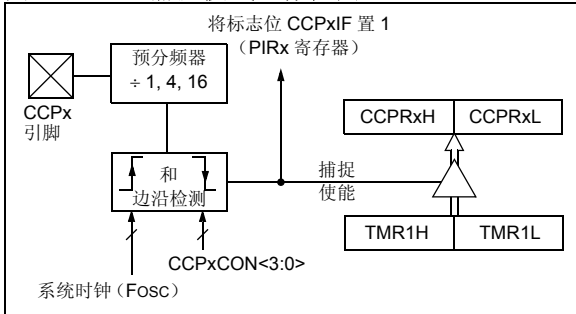
通过模式选择位 CCPxM3:CCPxM0 (CCPxCON<3:0>) 选择事件类型。当一个捕捉发生时，中断请求标志位 PIRx 寄存器中的 CCPxIF 置 1；它必须用软件清零。如果在 CCPRxH 和 CCPRxL 这对寄存器中的值被读取之前发生另一次捕捉，那么之前捕捉的值将被新捕捉的值覆盖（见图 11-1）。

11.3.1 CCP 引脚配置

在捕捉模式下，应通过将对应的 TRIS 控制位置 1 来将相应的 CCPx 引脚配置为输入。

注： 如果 CCPx 引脚被配置为输出，对该端口的写操作可能引发一个捕捉事件。

图 11-1: 捕捉模式工作框图



11.3.2 TIMER1 模式选择

Timer1 必须运行在定时器模式或同步计数器模式下 CCP 模块才能使用捕捉功能。在异步计数器模式下无法进行捕捉操作。

11.3.3 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 PIRx 寄存器中的 CCPxIE 中断允许位清零以避免产生误中断。在操作模式发生任何改变之后也应清零 PIRx 寄存器中的中断标志位 CCPxIF。

11.3.4 CCP 预分频器

CCPxCON 寄存器中的 CCPxM<3:0> 位指定了 4 种预分频器设置。每当关闭 CCP 模块或禁止捕捉模式时，就会清零预分频器计数器。这意味着任何复位都将清零预分频计数器。

从一种捕捉预分频比切换到另一种捕捉预分频比不会将预分频计数器清零，但可能会产生误中断。要避免出现这种不期望的操作，应在改变预分频比前通过将 CCPxCON 寄存器清零关闭该模块（见例 11-1）。

例 11-1: 改变捕捉预分频比

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


PIC16F882/883/884/886/887

11.4 比较模式

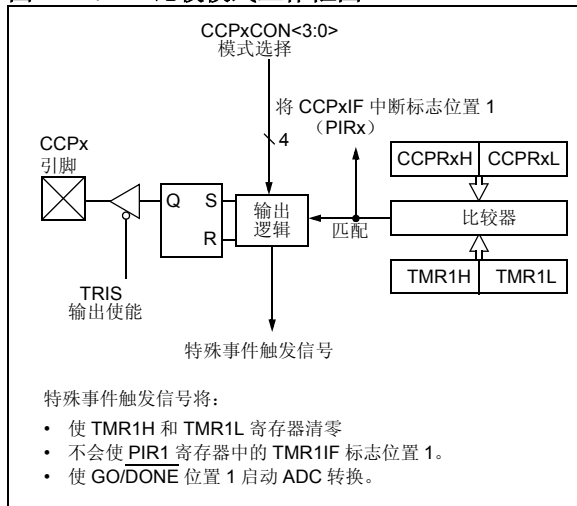
在比较模式下，16 位 CCPRx 寄存器的值将不断与一对 TMR1 寄存器的值相比较。当两者匹配时，CCPx 模块可能会出现以下几种情况：

- CCPx 的输出电平翻转
- CCPx 输出高电平
- CCPx 输出低电平
- 产生特殊事件触发信号
- 产生软件中断

引脚的动作取决于 CCPx1CON 寄存器中 CCPxM<3:0> 控制位的值。

所有捕捉模式都会产生中断。

图 11-2: 比较模式工作框图



11.4.1 CCP 引脚配置

用户必须通过将相应的 TRIS 位清零来将 CCPx 引脚配置为输出。

注： 清零 CCP1CON 寄存器会将 CCPx 比较输出锁存器强制为默认的低电平。这不是端口 I/O 数据锁存器。

11.4.2 TIMER1 模式选择

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

11.4.3 软件中断模式

当选择了产生软件中断模式时 (CCPxM<3:0> = 1010)，CCPx 模块不会控制 CCPx 引脚 (见 CCP1CON 寄存器)。

11.4.4 特殊事件触发信号

当选择了特殊事件触发模式 (CCPxM<3:0> = 1011) 时，CCPx 模块将完成以下操作：

- 复位 Timer1
- 如果使能了 ADC 还将启动 ADC 转换

在该模式下，CCPx 模块不控制 CCPx 引脚 (见 CCPxCON 寄存器)。

当 TMR1H/TMR1L 寄存器对和 CCPRxH/CCPRxL 寄存器对匹配时 CCP 会立即产生特殊事件触发输出。TMR1H/TMR1L 寄存器对不会复位直到 Timer1 时钟的下一个上升沿才复位。从而使 CCPRxH/CCPRxL 寄存器对实际上成为了 Timer1 的 16 位可编程周期寄存器。

- 注 1：** 来自 CCP 模块的特殊事件触发信号不会使 PIR1 寄存器中的 TMRxIF 中断标志位置 1。
- 注 2：** 在产生特殊事件触发信号的边沿和导致 Timer1 复位的时钟边沿之间改变 CCPRxH 和 CCPRxL 寄存器对的内容可清除匹配条件，从而阻值复位发生。

PIC16F882/883/884/886/887

11.5 PWM 模式

PWM 模式在 CCPx 引脚上产生脉宽调制信号。由以下寄存器确定占空比、周期和分辨率：

- PR2
- T2CON
- CCPRxL
- CCPxCON

在脉宽调制 (PWM) 模式下, CCP 模块可在 CCPx 引脚上输出分辨率高达 10 位的 PWM 信号。由于 CCPx 引脚与端口数据锁存器复用, 必须清零相应的 TRIS 位才能启用 CCPx 引脚的输出驱动器。

注: 清零 CCPxCON 寄存器将放弃 CCPx 对 CCPx 引脚的控制权。

图 11-3 给出了 PWM 操作的简化框图。

图 11-4 给出了 PWM 信号的典型波形。

如需了解设置 CCP 模块以进行 PWM 操作的详细步骤, 请参见第 11.5.7 节“设置 PWM 操作”。

PWM 输出 (图 11-4) 包含一个时基 (周期) 和一段输出保持高电平的时间 (占空比)。

图 11-4: CCP PWM 输出

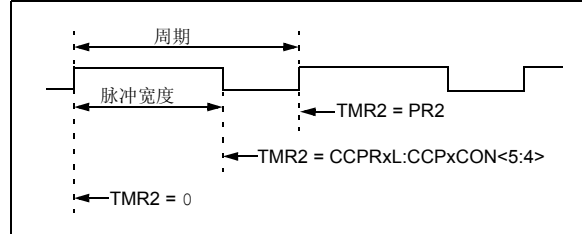
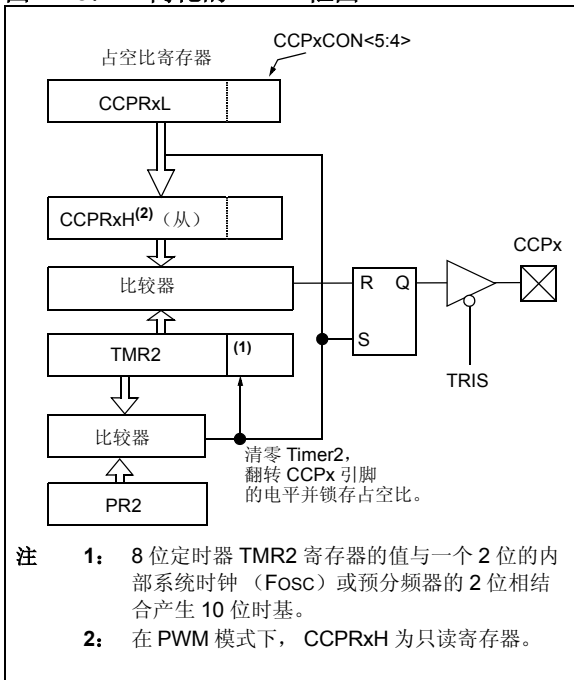


图 11-3: 简化的 PWM 框图



PIC16F882/883/884/886/887

11.5.1 PWM 周期

PWM 周期是通过写 Timer2 的 PR2 寄存器来指定的。可以使用公式 11-1 计算 PWM 周期：

公式 11-1: PWM 周期

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

当 TMR2 等于 PR2 时，在下一个递增计数周期中会发生以下 3 个事件：

- TMR2 被清零
- CCPx 引脚被置 1（例外情况：如果 PWM 占空比 = 0%，CCPx 引脚将不被置 1）
- PWM 占空比从 CCPRxL 被锁存到 CCPxH

注： 在确定 PWM 频率时不使用 Timer2 后分频比（见第 7.1 节“Timer2 的工作原理”）。

11.5.2 PWM 占空比

可通过将一个 10 位值写入以下多个寄存器来指定 PWM 占空比：CCPRxL 寄存器和 CCPxCON 寄存器的 DCxB<1:0> 位。CCPRxL 保存占空比的高 8 位，而 CCPxCON 寄存器的 DCxB<1:0> 位保存占空比的低 2 位。可以在任何时候写入 CCPRxL 和 CCPxCON 寄存器的 DCxB<1:0> 位，但直到 PR2 和 TMR2 中的值匹配（即周期结束）时，占空比的值才被锁存到 CCPxH 中。在 PWM 模式下，CCPRxH 是只读寄存器。

公式 11-2 用于计算 PWM 脉冲的宽度。

公式 11-3 用于计算 PWM 占空比。

公式 11-2: 脉冲宽度

$$脉冲宽度 = (CCPRxL:CCPxCON<5:4>) \cdot T_{osc} \cdot (TMR2 \text{ 预分频值})$$

公式 11-3: 占空比

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

CCPRxH 寄存器和一个 2 位的内部锁存器用于为 PWM 占空比提供双重缓冲。这种双重缓冲结构极其重要，可以避免在 PWM 操作过程中产生毛刺。

8 位定时器 TMR2 寄存器的值与一个 2 位的内部系统时钟（Fosc）或预分频器的 2 位相结合，产生 10 位时基。当 Timer2 预分频比为 1:1 时使用系统时钟。

当 10 位时基与 CCPRxH 和 2 位锁存器相结合的值匹配时，CCPx 引脚被清零（见图 11-3）。

PIC16F882/883/884/886/887

11.5.3 PWM 分辨率

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

当 PR2 为 255 时，PWM 的最大分辨率为 10 位。如公式 11-4 所示，分辨率是 PR2 寄存器值的函数。

公式 11-4: PWM 分辨率

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

注：如果脉冲宽度大于周期值，指定的 PWM 引脚将保持不变。

表 11-3: PWM 频率和分辨率示例 (Fosc = 20 MHz)

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

表 11-4: PWM 频率和分辨率示例 (Fosc = 8 MHz)

PWM 频率	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
定时器预分频值 (1、4 或 16)	16	4	1	1	1	1
PR2 值	0x65	0x65	0x65	0x19	0x0C	0x09
最高分辨率 (位)	8	8	8	6	5	5

PIC16F882/883/884/886/887

11.5.4 休眠模式下的操作

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

11.5.5 系统时钟频率的改变

PWM 频率是由系统时钟频率产生的。系统时钟频率发生任何改变都会使 PWM 频率发生变化。更多详细信息，请参见第 4.0 节“振荡器模块（带故障保护时钟监视器）”。

11.5.6 复位的影响

任何复位都会将所有端口强制为输入模式，并强制 CCP 寄存器进入其复位状态。

11.5.7 设置 PWM 操作

在将 CCP 模块配置为 PWM 操作模式时应该执行以下步骤：

1. 通过将相应的 TRIS 位置 1，禁止 PWM 引脚（CCPx）的输出驱动器，使之成为输入引脚。
2. 通过装载 PR2 寄存器设置 PWM 周期。
3. 通过用适当的值装载 CCPxCON 寄存器配置 CCP 模块的 PWM 模式。
4. 通过装载 CCPRxL 寄存器和 CCPxCON 寄存器中的 DCxB<1:0> 位设置 PWM 占空比。
5. 配置并启动 Timer2:
 - 清零 PIR1 寄存器中的 TMR2IF 中断标志位。
 - 通过装载 T2CON 寄存器的 T2CKPS 位来设置 Timer2 预分频比。
 - 通过将 T2CON 寄存器中的 TMR2ON 位置 1 来使能 Timer2。
6. 在新的 PWM 周期开始后，使能 PWM 输出：
 - 等待 Timer2 溢出（PIR1 寄存器中的 TMR2IF 位置 1）。
 - 通过将相应的 TRIS 位清零，使能 CCPx 引脚输出驱动器。

PIC16F882/883/884/886/887

11.6 PWM（增强模式）

增强型 PWM 模式能在最多 4 个不同的引脚上产生分辨率最高为 10 位的 PWM 信号。它是通过以下 4 种 PWM 输出模式做到这一点的：

- 单输出
- 半桥输出
- 全桥输出，正向模式
- 全桥输出，反向模式

要选择一种增强型 PWM 模式，必须对 CCP1CON 寄存器中的 P1M 位进行适当设置。

注： 只有增强型捕捉/比较/PWM 模块 (CCP1) 才可使用增强型 PWM 模式。

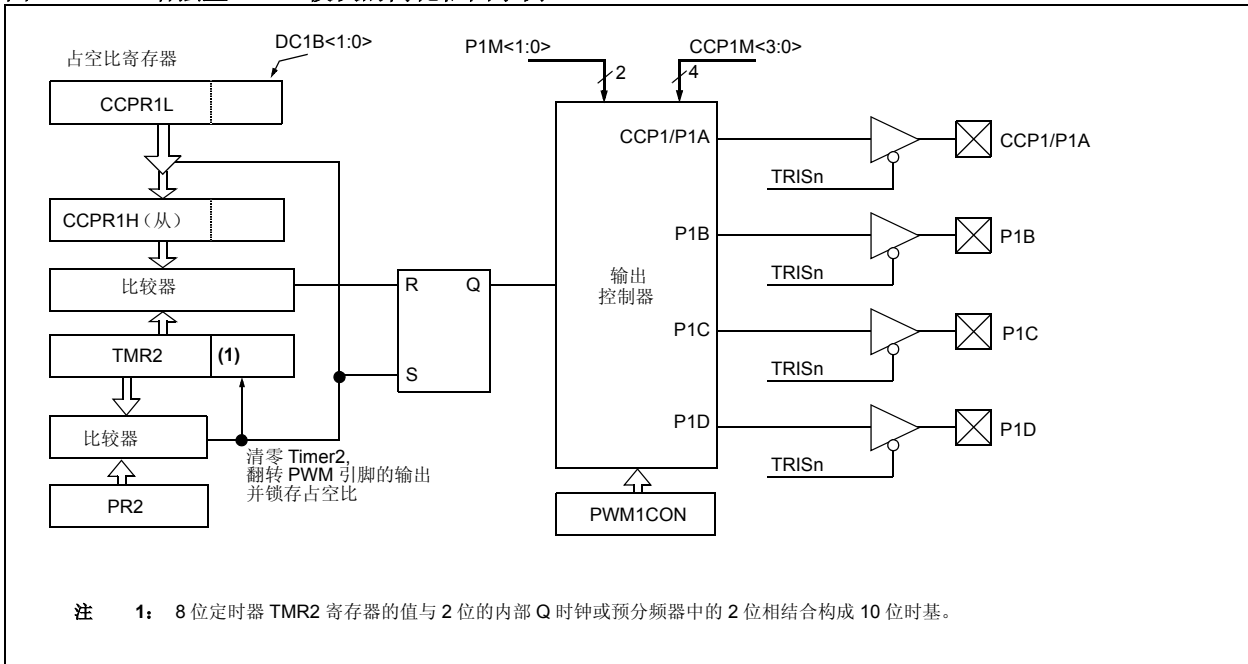
PWM 输出与 I/O 引脚复用，并被称为 P1A、P1B、P1C 和 P1D。PWM 引脚的极性是可配置的，通过对 CCP1CON 寄存器中的 CCP1M 位进行适当设置来选择极性。

表 11-5 给出了每种增强型 PWM 模式的引脚配置。

图 11-5 给出了增强型 PWM 模块的简化框图示例。

注： 为避免在第一次使能 PWM 时产生不完整的波形，ECCP 模块将等待新的 PWM 周期开始后才产生 PWM 信号。

图 11-5: 增强型 PWM 模块的简化框图示例



- 注 1：** 必须为每个 PWM 输出适当配置 TRIS 寄存器的值。
- 2：** 将 CCPxCON 寄存器清零将放弃 ECCP 对所有 PWM 输出引脚的控制权。
- 3：** 任何不被增强型 PWM 模式使用的引脚都可用作其他引脚功能。

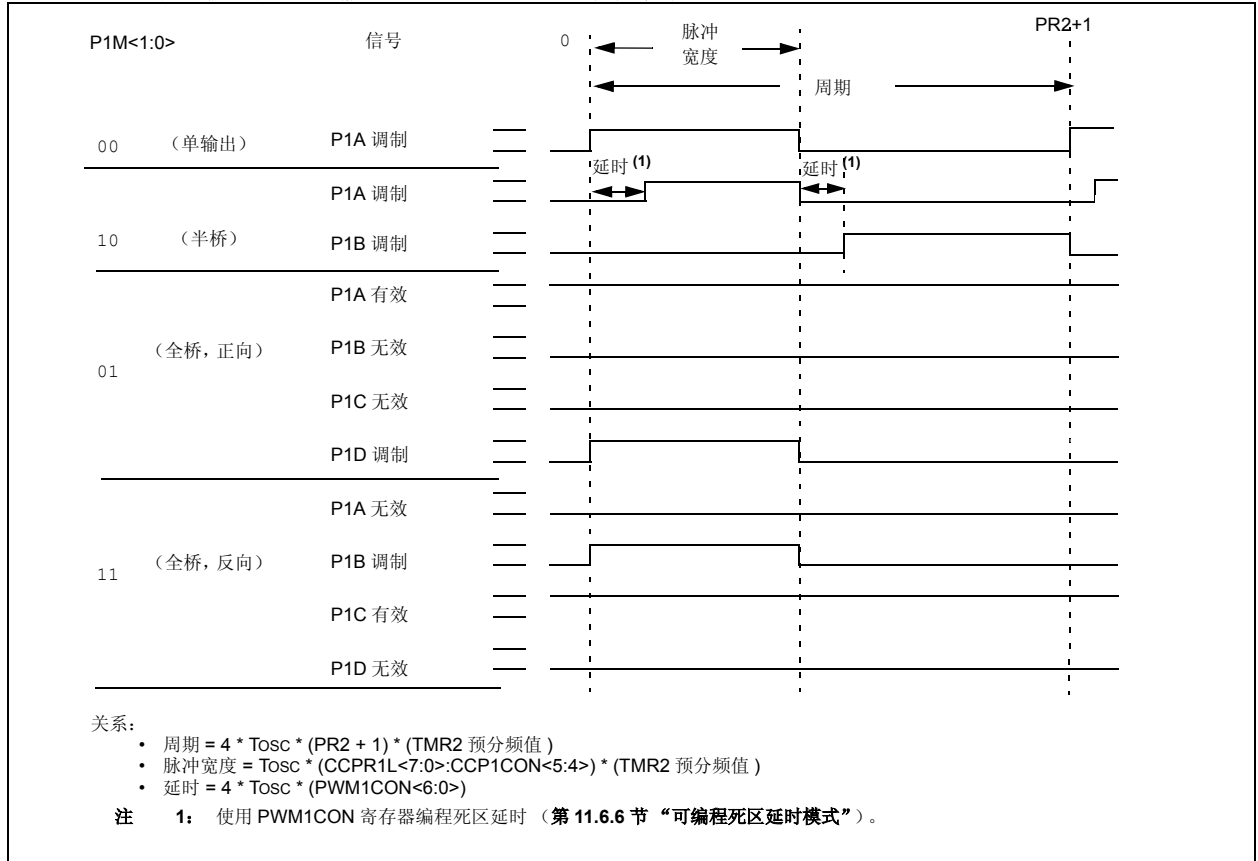
表 11-5: 各种 PWM 增强模式的引脚配置示例

ECCP 模式	P1M<1:0>	CCP1/P1A	P1B	P1C	P1D
单输出	00	有 ⁽¹⁾	有 ⁽¹⁾	有 ⁽¹⁾	有 ⁽¹⁾
半桥输出	10	有	有	无	无
全桥输出，正向模式	01	有	有	有	有
全桥输出，反向模式	11	有	有	有	有

注 1： 脉冲转向可启用单输出模式的输出。

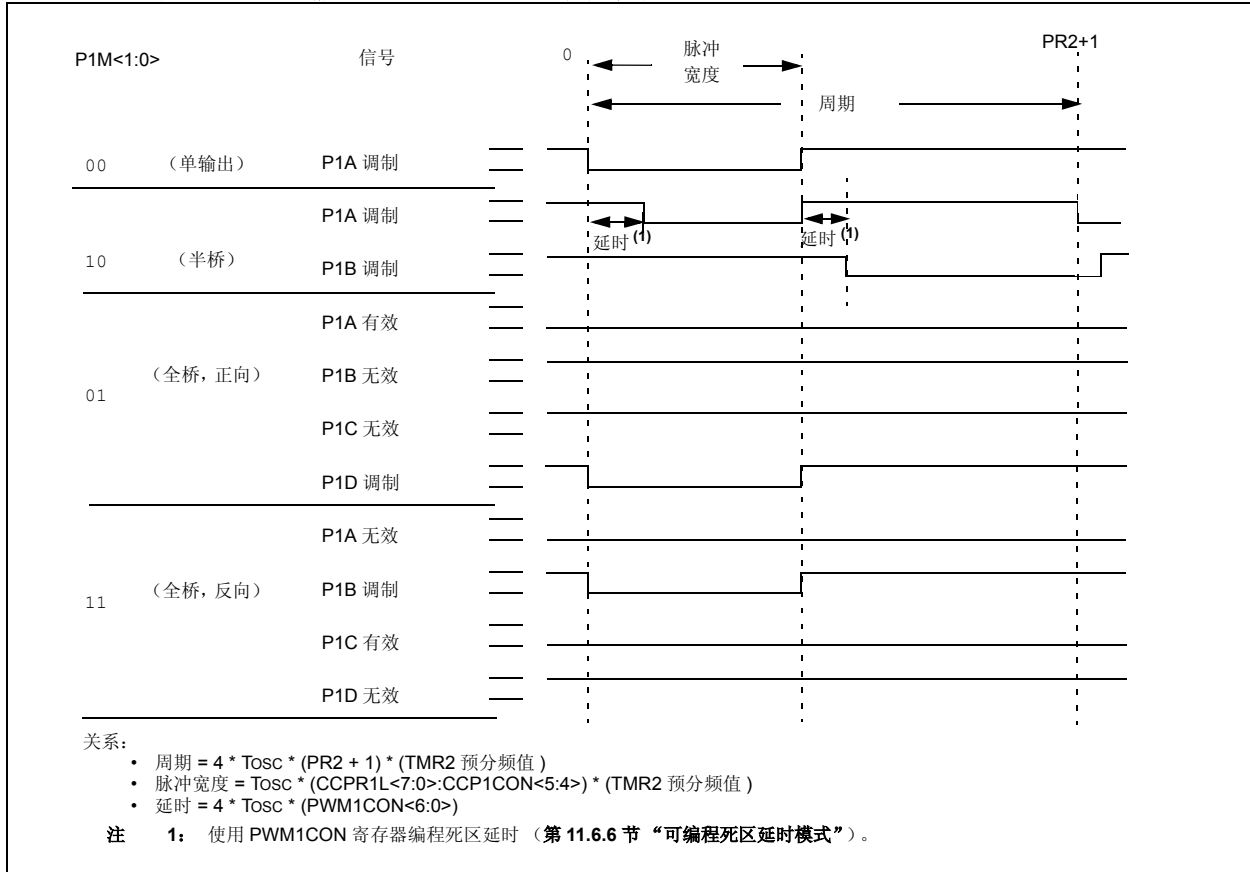
PIC16F882/883/884/886/887

图 11-6: 增强模式 PWM 输出关系示例 (高电平有效状态)



PIC16F882/883/884/886/887

图 11-7: 增强型 PWM 输出关系示例 (低电平有效状态)



PIC16F882/883/884/886/887

11.6.1 半桥模式

在半桥输出模式下，两个引脚用作输出端来驱动推挽式负载。PWM 输出信号在 CCPx/P1A 引脚上输出，而互补的 PWM 输出信号在 P1B 引脚上输出（见图 11-9）。如图 11-9 所示，此模式可用于半桥应用，或那些使用 2 个 PWM 信号来调制 4 个电源开关的全桥应用。

在半桥输出模式下，可编程的死区延时用来防止在半桥电源设备中产生直通电流。PWM1CON 寄存器中 PDC<6:0> 位的值用来设置输出被驱动为有效前的指令周期数。如果该值大于占空比，在整个周期内对应的输出将保持无效。有关死区延时操作的更多详细信息，请参见第 11.6.6 节“可编程死区延时模式”。

由于 P1A 和 P1B 输出与端口数据锁存器复用，必须清零相应的 TRISC 位将 P1A 和 P1B 配置为输出。

图 11-8: 半桥 PWM 输出示例

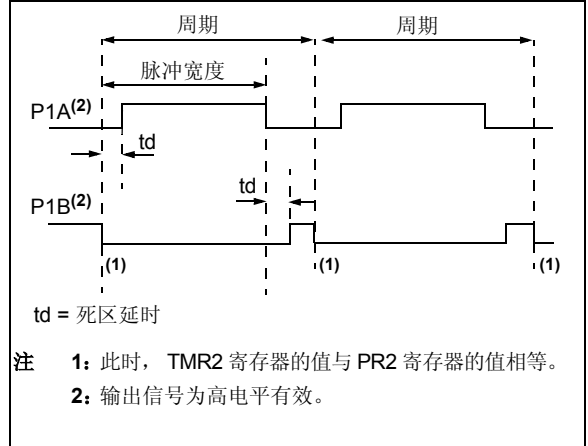
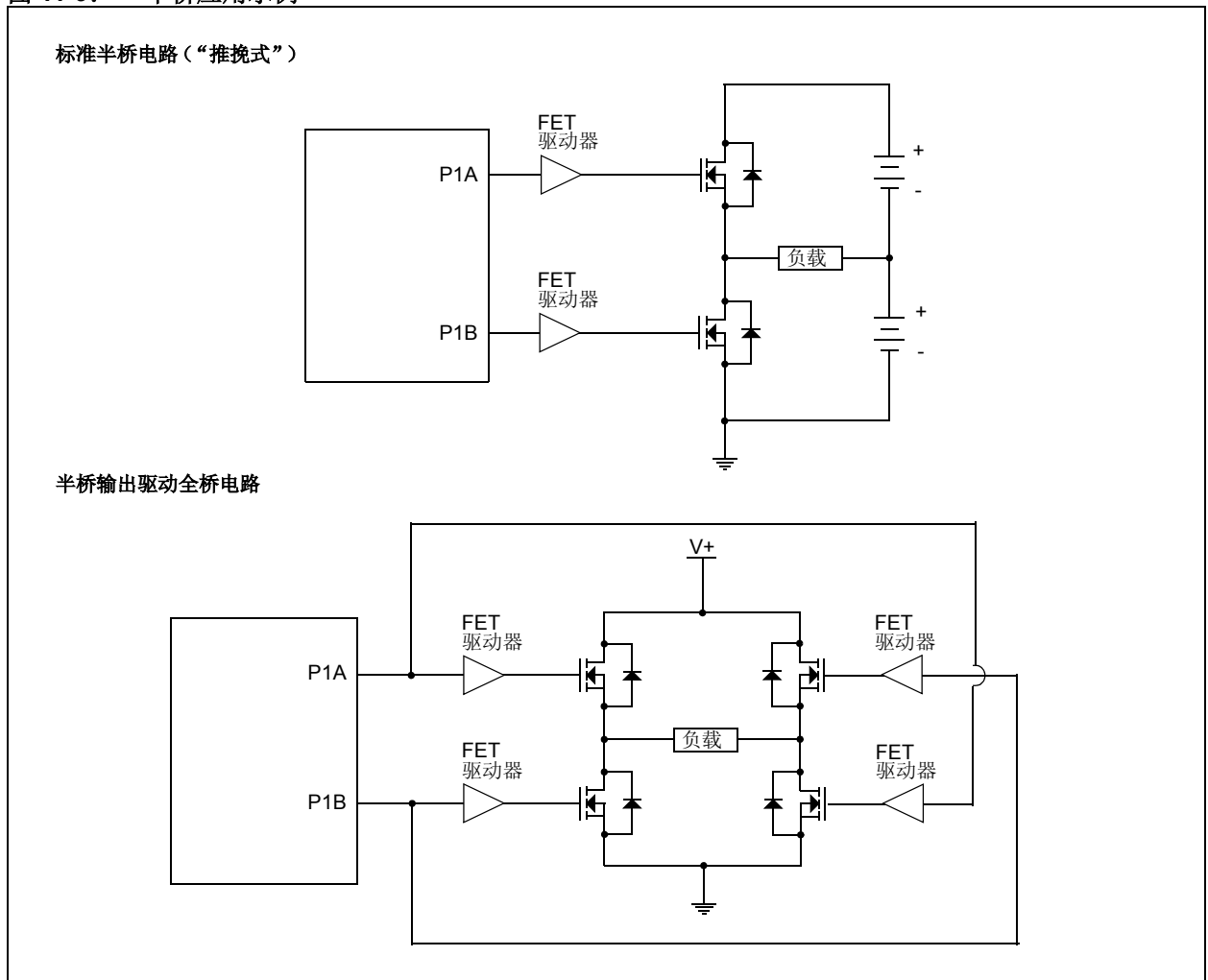


图 11-9: 半桥应用示例



PIC16F882/883/884/886/887

11.6.2 全桥模式

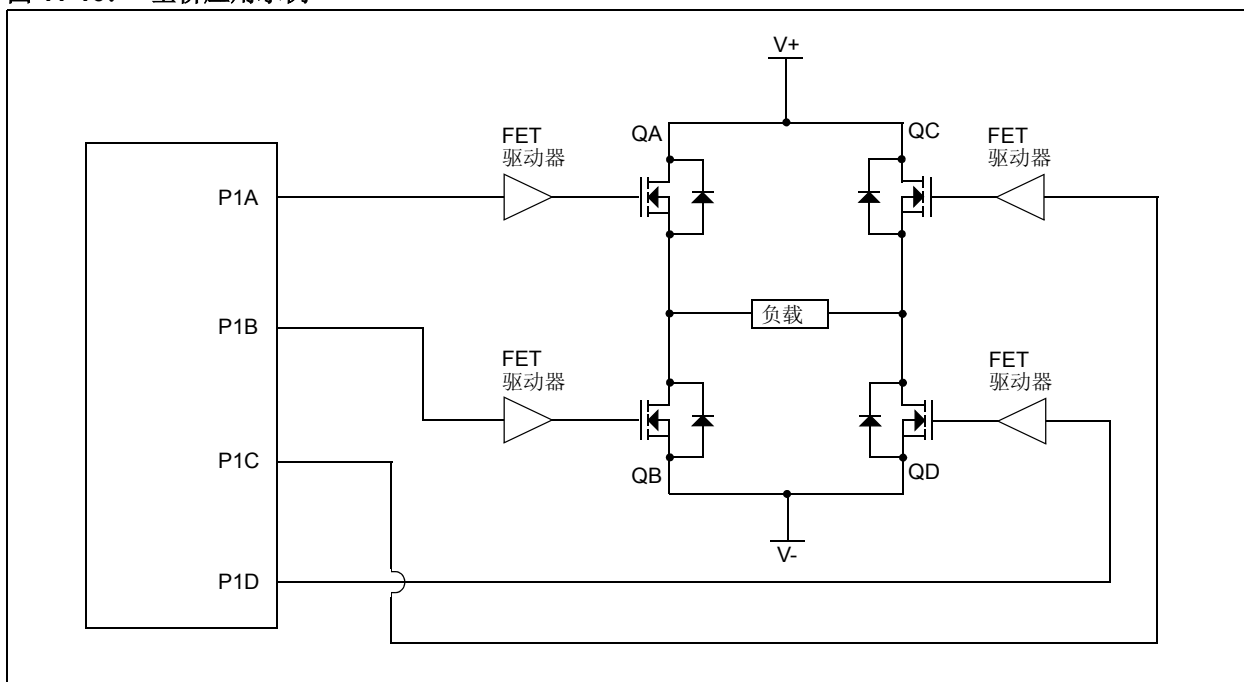
在全桥输出模式下，所有 4 个引脚均用作输出。图 11-10 给出了全桥应用的示例。

如图 11-11 所示，在正向模式下，引脚 CCP1/P1A 被驱动为其有效状态，引脚 P1D 为调制输出，而引脚 P1B 和 P1C 将被驱动为其无效状态。

如图 11-1 所示，在反向模式下，引脚 P1C 被驱动为其有效状态，引脚 P1B 为调制输出，而引脚 P1A 和 P1D 将被驱动为其无效状态。

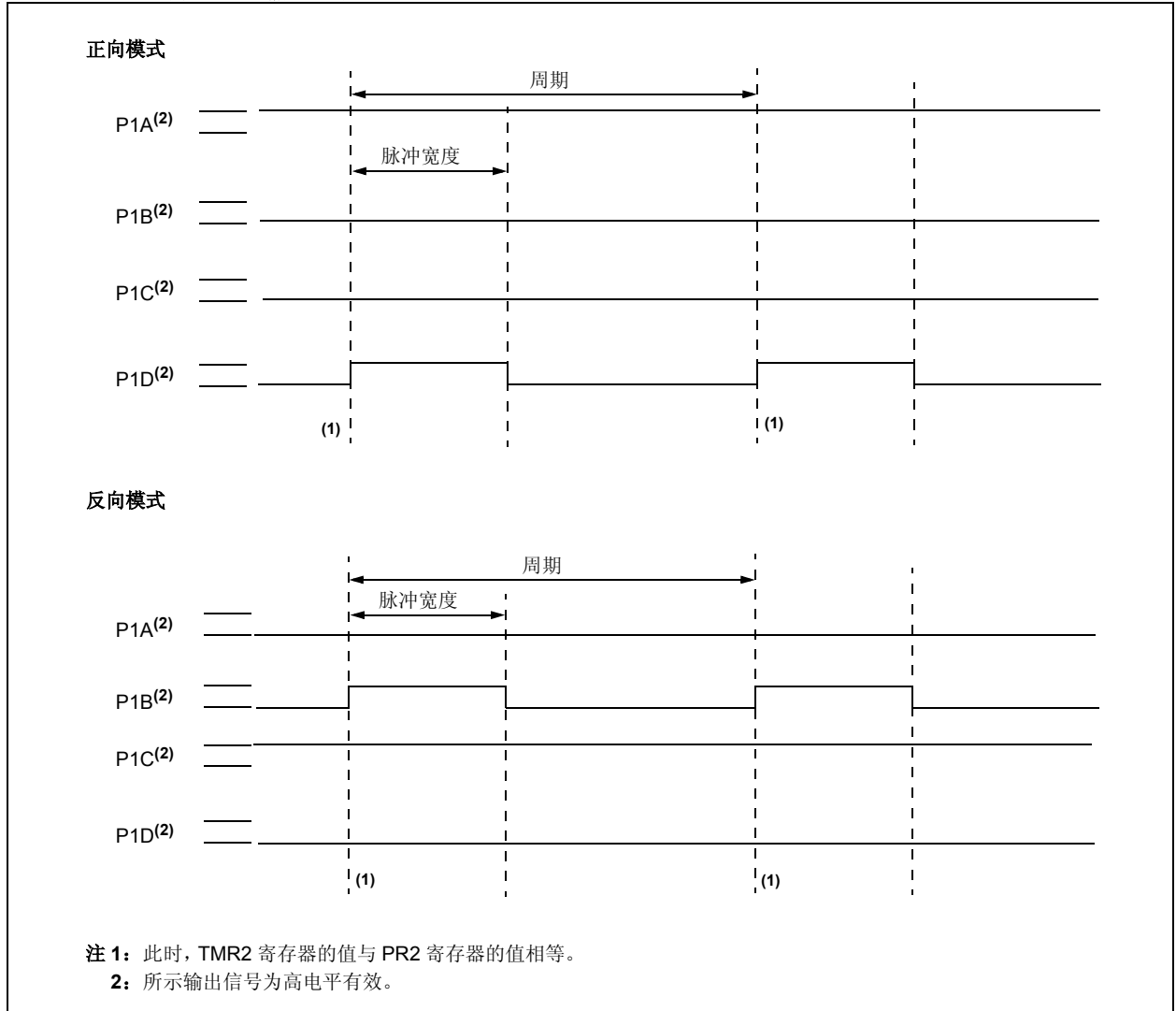
P1A、P1B、P1C 和 P1D 输出与端口数据锁存器复用。必须清零 TRISC 位将 P1A、P1B、P1C 和 P1D 配置为输出引脚。

图 11-10: 全桥应用示例



PIC16F882/883/884/886/887

图 11-11: 全桥 PWM 输出示例



注 1: 此时, TMR2 寄存器的值与 PR2 寄存器的值相等。

注 2: 所示输出信号为高电平有效。

PIC16F882/883/884/886/887

11.6.2.1 全桥模式下的方向更改

在全桥输出模式下，CCP1CON 寄存器中的 P1M1 位使用户能控制负载中电流为正向还是反向。当应用固件更改此方向控制位时，模块将在下一个 PWM 周期采用新的方向。

通过在软件中改变 CCP1CON 寄存器中的 P1M1 位来改变方向。在当前 PWM 周期结束前，以下序列发生的时间为 4 个 Timer2 周期：

- 调制的输出（P1B 和 P1D）被置于其无效状态。
- 相关的非调制输出（P1A 和 P1C）被切换到以相反的方向驱动负载。
- 在下一个周期的开始恢复 PWM 调制。

有关该序列的说明请参见图 11-12。

在全桥输出模式下不提供任何死区延时。通常，由于在任何时间只调制一个输出，所以不需要死区延时。然而，有一种情形可能需要死区延时，即，当以下两个条件都成立时的情况：

1. 当输出信号的占空比接近或等于 100% 时，PWM 输出的方向发生改变。
2. 电源开关（包括电源设备和驱动电路）的关断时间大于导通时间。

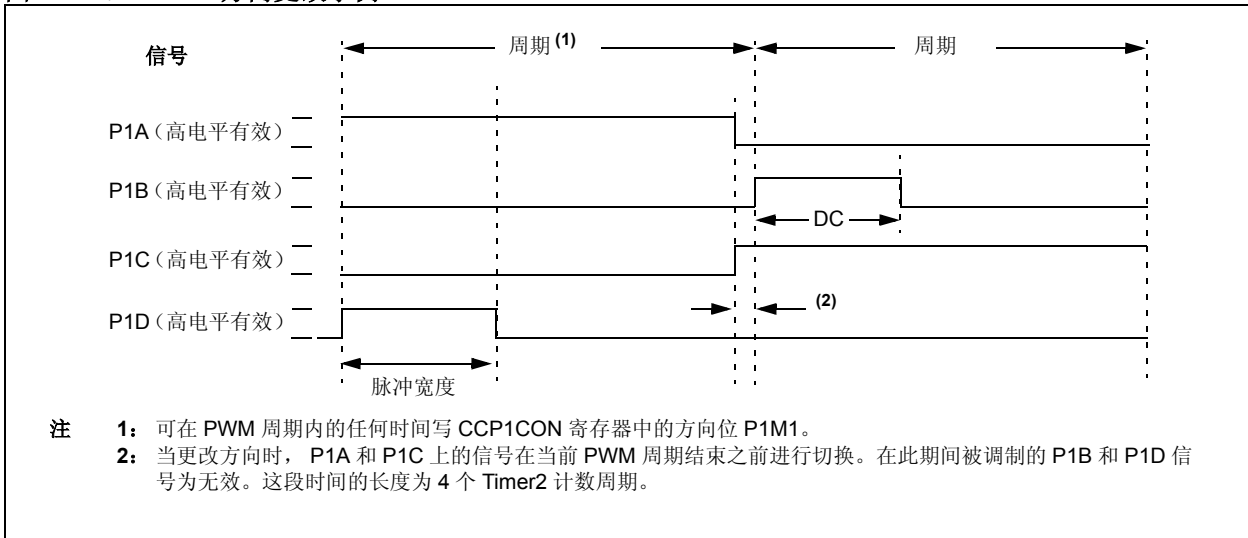
图 11-13 所示为占空比接近 100% 时，PWM 方向从正向更改为反向的示例。在时间 t1 时，输出 P1A 和 P1D 变为无效，而输出 P1C 变为有效。在此示例中，由于电源设备的关断时间比导通时间长，直通电流可能在时间段“t”内流过电源设备 QC 和 QD（见图 11-10）。如果 PWM 方向从反向更改为正向，电源设备 QA 和 QB 将出现相同的现象。

如果需要在高占空比情况下更改 PWM 方向，消除直通电流的两种可能的解决方案如下：

1. 在更改方向前的一个 PWM 周期减小 PWM 的占空比。
2. 使用可使开关元件的关断速度比导通速度更快的开关驱动器。

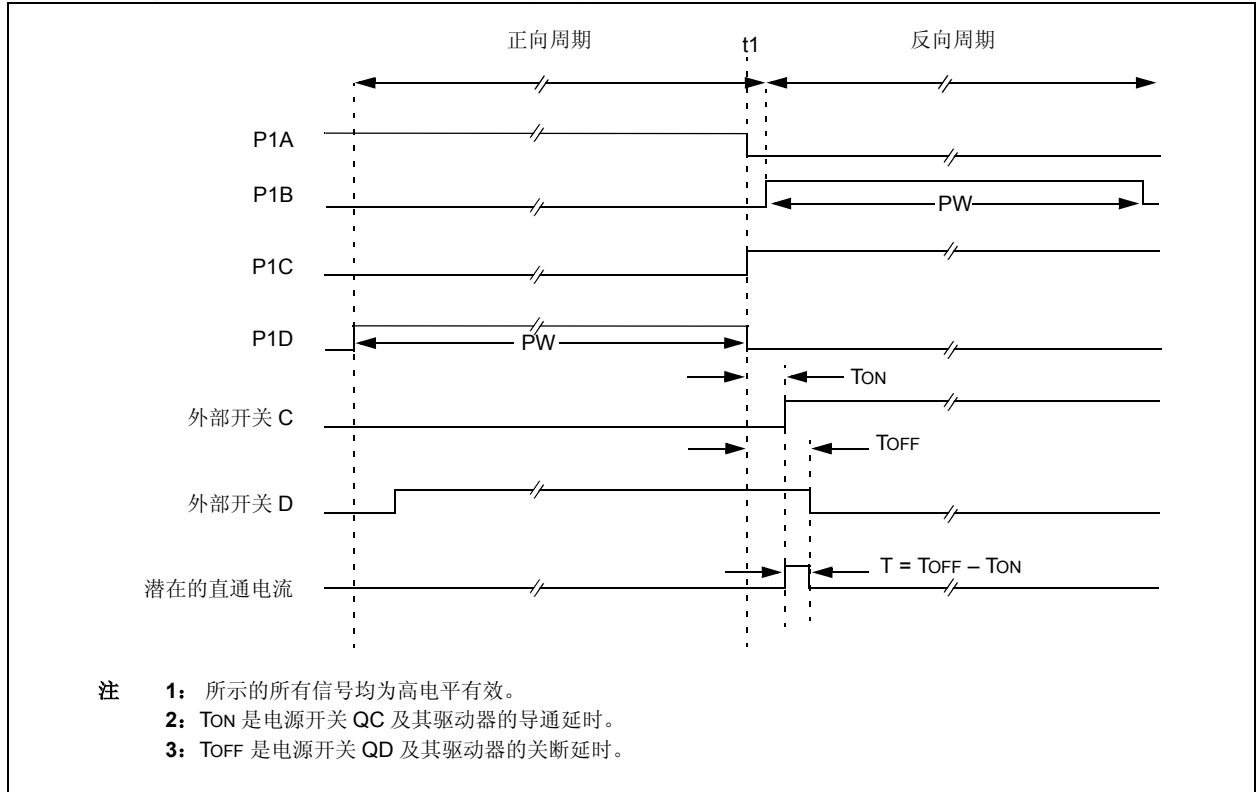
可能还存在防止直通电流的其他方法。

图 11-12: PWM 方向更改示例



PIC16F882/883/884/886/887

图 11-13: 在接近 100% 占空比时 PWM 更改方向的示例



PIC16F882/883/884/886/887

11.6.3 启动注意事项

使用任何 PWM 模式时，应用的硬件时必须在 PWM 输出引脚上使用适当的外部上拉和 / 或下拉电阻。

注： 当单片机复位结束时，所有的 I/O 引脚都处于高阻态。外部电路必须将电源开关元件保持在关断状态，直到单片机用适当的信号电平驱动 I/O 引脚，或激活 PWM 输出为止。

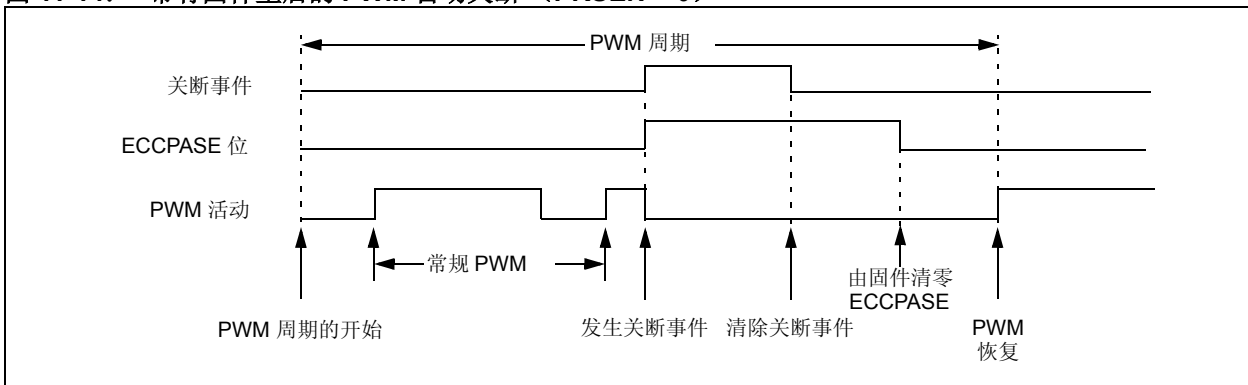
CCP1CON 寄存器中的 CCP1M<1:0> 位允许用户为每对 PWM 输出引脚 (P1A/P1C 和 P1B/P1D) 选择 PWM 输出信号是高电平有效还是低电平有效。必须在使能 PWM 引脚的输出驱动器之前选择 PWM 输出的极性。建议不要在使能 PWM 引脚输出驱动器时更改极性配置，因为这可能会损坏应用电路。

当初始化 PWM 模块时，P1A、P1B、P1C 和 P1D 的输出锁存器可能处于不正确的状态。使能增强型 PWM 模式的同时使能 PWM 引脚的输出驱动器有可能损坏应用电路。必须先将 ECCP 模块使能为正确的输出模式，并且必须经过一个完整的 PWM 周期后，才能使能 PWM 引脚的输出驱动器。当第 2 个 PWM 周期开始时，如果 PIR1 寄存器中的 TMR2IF 位置 1，即表明经过了一个完整的 PWM 周期。

PIC16F882/883/884/886/887

- 注 1:** 自动关断条件是基于电平而不是基于边沿的事件。只要关断电平存在，自动关断就将保持。
- 2:** 当存在自动关断条件时，禁止写入 ECCPASE 位。
- 3:** 一旦清除了自动关断条件并重启了 PWM（通过固件或自动重启），PWM 信号将始终从下一个 PWM 周期开始时重启。

图 11-14: 带有固件重启的 PWM 自动关断 (PRSEN = 0)

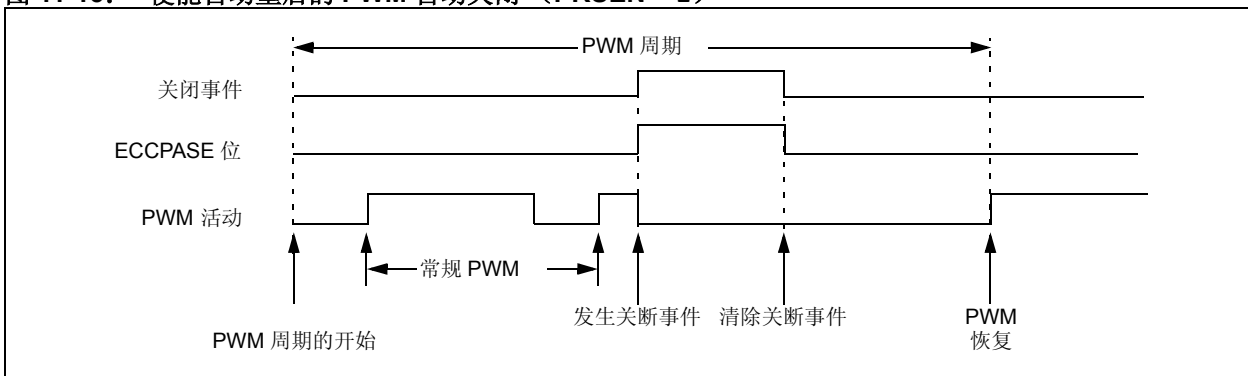


11.6.5 自动重启模式

可以将增强型 PWM 配置为一旦清除自动关断条件就自动重启 PWM 信号。通过将 PWM1CON 寄存器中的 PRSEN 位置 1 使能自动重启。

如果使能自动重启，只要自动关断条件有效，ECCPASE 位就将保持置 1。当清除自动关闭条件时，将通过硬件将 ECCPASE 位清零，并且将恢复常规操作。

图 11-15: 使能自动重启的 PWM 自动关闭 (PRSEN = 1)



PIC16F882/883/884/886/887

11.6.6 可编程死区延时模式

在半桥应用中，模块一直以 PWM 频率调制所有的电源开关，关闭电源开关通常比打开它需要更多的时间。如果上方的电源开关和下方的电源开关同时开关（一个打开，另一个关闭），两个开关可能会在一段很短的时间内都处于打开状态，直到一个开关完全关闭为止。在这很短的间隔内，很大的电流（直通电流）可能流过两个电源开关，从而导致半桥供电电路短路。为了避免在开关期间流过这种潜在的破坏性直通电流，一般延迟打开其中的一个电源开关以允许另一个开关完全关闭。

在半桥输出模式下，可数字编程的死区延时可用来避免直通电流破坏电源开关。该延时在信号从非有效状态到有效状态转换时发生。参见图 11-6。相关 PWM1CON 寄存器（寄存器 11-4）的低 7 位以单片机指令周期为单位设置延时（Tcy 或 4 Tosc）。

图 11-16: 半桥 PWM 输出示例

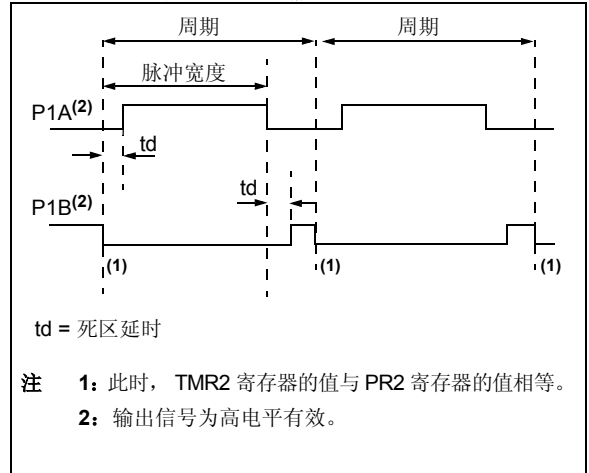
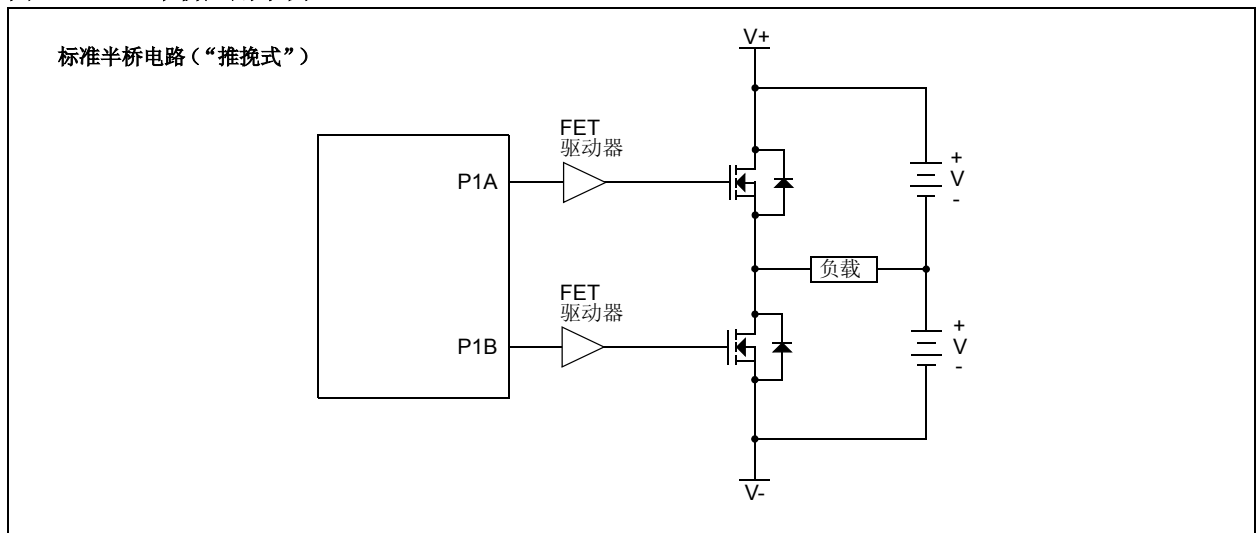


图 11-17: 半桥应用示例



PIC16F882/883/884/886/887

寄存器 11-4: PWM1CON: 增强型 PWM 控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 7 **PRSEN:** PWM 重启使能位

1 = 自动关断时, 一旦关断事件被清除, ECCPASE 位将立即自动清零; PWM 自动重启

0 = 自动关断时, 必须用软件清零 ECCPASE 位以重启 PWM

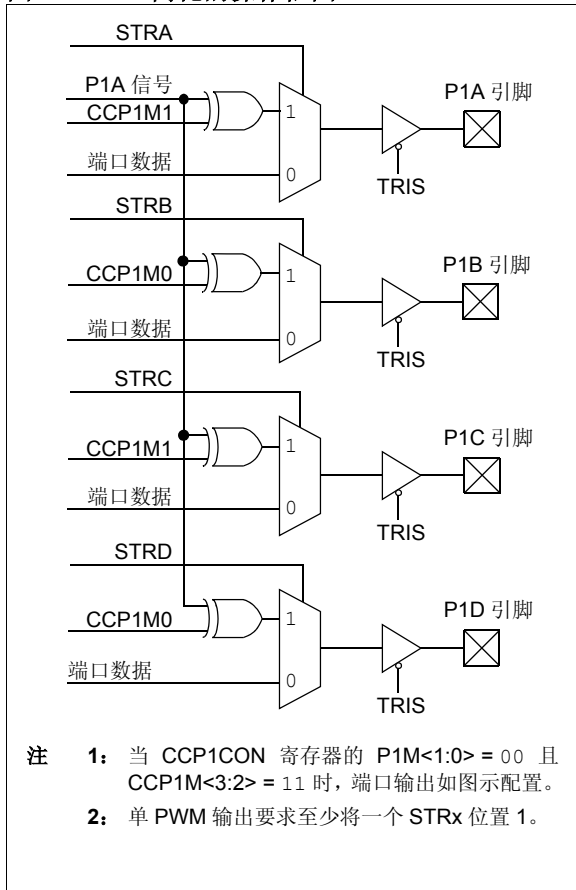
bit 6-0 **PDC<6:0>:** PWM 延时计数位

PDCn = PWM 信号转换到有效的**预计时间**和**实际时间**之差, 以 $F_{osc}/4(4 * T_{osc})$ 周期的倍数表示

注 1: 当使能故障保护模式或选择 LP、XT 或 HS 振荡模式时, 发生双速启动会使该位复位为 0。

PIC16F882/883/884/886/887

图 11-18: 简化的操作框图



PIC16F882/883/884/886/887

11.6.7.1 转向同步

PSTRCON 寄存器中的 STRSYNC 位向用户提供了何时发生转向事件的两种选择。当 STRSYNC 位为 0 时，转向事件将会在写入 PSTRCON 寄存器的指令结束前发生。在此情形下，P1<D:A> 引脚可能输出不完整的 PWM 波形。当用户固件需要使引脚立即停止 PWM 信号输出时，该操作很有用。

当 STRSYNC 位为 1 时，实际的转向更新将发生在下一个 PWM 周期的开始。在此情形下，转向开 / 关 PWM 输出将始终产生完整的 PWM 波形。

图 11-19 和图 11-20 说明了 PWM 转向在不同 STRSYNC 设置下的时序图。

图 11-19: 发生在指令结束前的换向事件示例 (STRSYNC = 0)

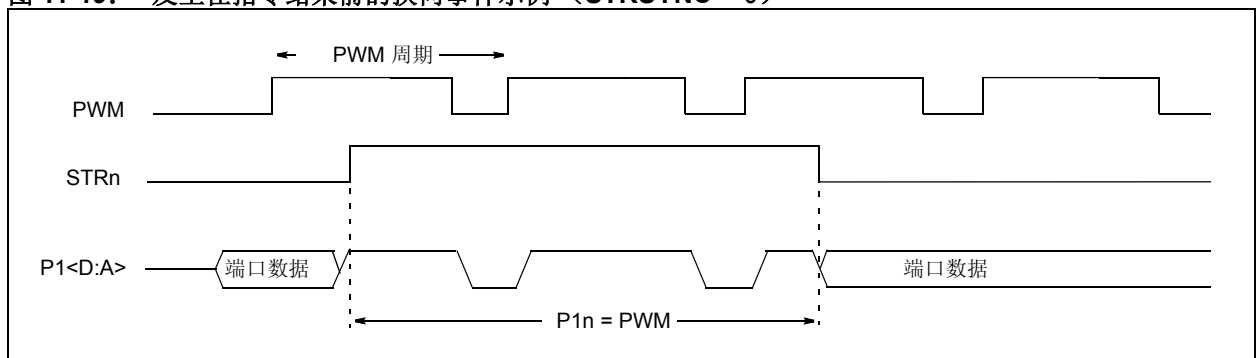
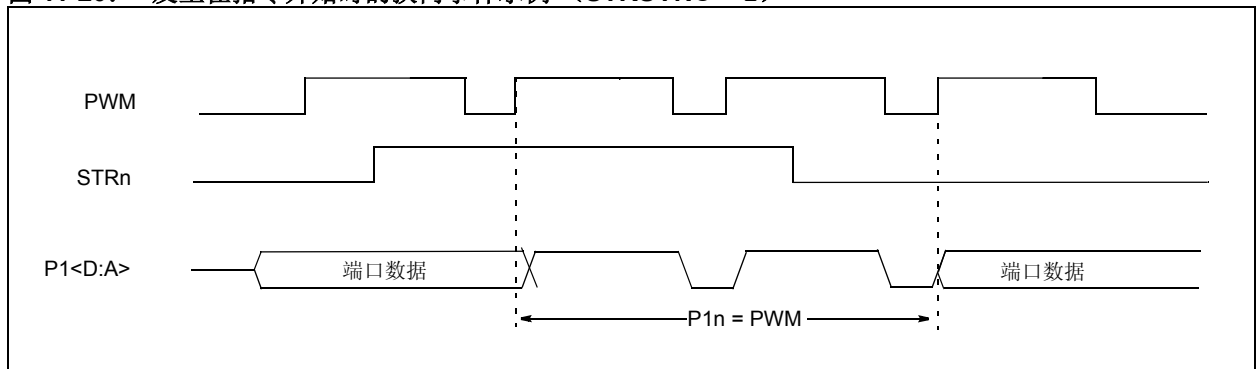


图 11-20: 发生在指令开始时的换向事件示例 (STRSYNC = 1)



PIC16F882/883/884/886/887

表 11-6: 与捕捉、比较和 TIMER1 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR和BOR时的值	所有其他复位时的值
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000
CCPR1L	捕捉 / 比较 / PWM 寄存器 1 的低字节 (LSB)								xxxx xxxx	xxxx xxxx
CCPR1H	捕捉 / 比较 / PWM 寄存器 1 的高字节 (MSB)								xxxx xxxx	xxxx xxxx
CCPR2L	捕捉 / 比较 / PWM 寄存器 2 的低字节 (LSB)								xxxx xxxx	xxxx xxxx
CCPR2H	捕捉 / 比较 / PWM 寄存器 2 的高字节 (MSB)								xxxx xxxx	xxxx xxxx
CM2CON1	MC1OUT	MC2OUT	C1RSEL	C2RSEL	—	—	T1GSS	C2SYNC	0000 --10	0000 --10
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	ULPWUIE	—	CCP2IE	0000 00-0	0000 00-0
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	ULPWUIF	—	CCP2IF	0000 00-0	0000 00-0
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNĀ	TMR1CS	TMR1ON	0000 0000	0000 0000
TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								xxxx xxxx	xxxx xxxx
TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								xxxx xxxx	xxxx xxxx
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111

图注: - = 未实现单元 (读为 0), u = 不变, x = 未知。捕捉和比较操作不使用阴影单元。

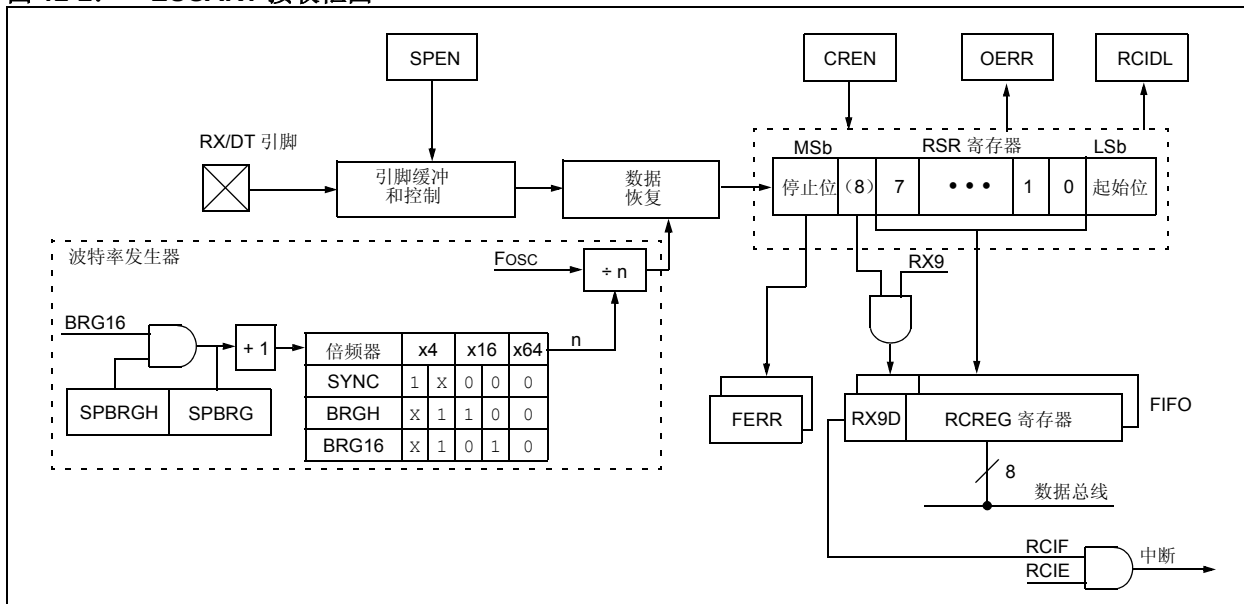
表 11-7: 与 PWM 和 TIMER2 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR和BOR时的值	所有其他复位时的值
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000
ECCPAS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000	0000 0000
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PR2	Timer2 周期寄存器								1111 1111	1111 1111
PSTRCON	—	—	—	STRSYNC	STRD	STRC	STRB	STRA	---0 0001	---0 0001
PWM1CON	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000	0000 0000
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
TMR2	Timer2 模块寄存器								0000 0000	0000 0000
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	1111 1111

图注: - = 未实现单元 (读为 0), u = 不变, x = 未知。PMW 操作不使用阴影单元。

PIC16F882/883/884/886/887

图 12-2: EUSART 接收框图



EUSART 模块的操作是通过 3 个寄存器控制的:

- 发送状态和控制寄存器 (TXSTA)
- 接收状态和控制寄存器 (RCSTA)
- 波特率控制寄存器 (BAUDCTL)

寄存器 12-1、寄存器 12-2 和寄存器 12-3 分别详细介绍了这些寄存器。

12.1 EUSART 异步模式

EUSART 使用标准不归零码 (non-return-to-zero, NRZ) 格式发送和接收数据。使用 2 种电平实现 NRZ: 代表 1 数据位的 V_{OH} 标号状态 (mark state), 和代表 0 数据位的 V_{OL} 空格状态 (space state)。采用 NRZ 格式连续发送相同值的数据位时, 输出电平将保持该位的电平, 而不会在发送完每个位后返回中间电平值。NRZ 发送端口在标号状态空闲。每个发送的字符都包括一个起始位, 后面跟有 8 个或 9 个数据位和一个或多个终止字符发送的停止位。起始位总是处于空格状态, 停止位总是处于标号状态。最常用的数据格式为 8 位。每个发送位的持续时间为 $1/(波特率)$ 。片上专用 8 位/16 位波特率发生器可用于通过系统振荡器产生标准波特率频率。请参见表 12-5 了解波特率配置示例。

EUSART 首先发送和接收 LSB。EUSART 的发送器和接收器在功能上是相互独立的, 但采用相同的数据格式和波特率。硬件不支持奇偶校验, 但可以用软件实现 (奇偶校验位是第 9 个数据位)。

12.1.1 EUSART 异步发生器

图 12-1 所示为 EUSART 发送器的框图。发送器的核心是串行发送移位寄存器 (TSR), 该寄存器不能由软件直接访问。TSR 从 TXREG 发送缓冲寄存器获取数据。

12.1.1.1 使能发送器

通过配置如下三个控制位使能 EUSART 发送器, 以用于异步操作:

- TXEN = 1
- SYNC = 0
- SPEN = 1

假设所有其他 EUSART 控制位处于其默认状态。

将 TXSTA 寄存器的 TXEN 位置 1, 使能 EUSART 发送器电路。将 TXSTA 寄存器的 SYNC 位清零, 将 EUSART 配置用于异步操作。将 RCSTA 寄存器的 SPEN 位置 1, 使能 EUSART 并自动将 TX/CK I/O 引脚配置为输出引脚。如果与模拟外设共用 TX/CK 引脚, 必须清零相应的 ANSEL 位禁止模拟 I/O 功能。

注 1: 将 SPEN 位置 1 会自动将 RX/DT I/O 引脚配置为输入引脚时, 无论相关的 TRIS 位的状态如何以及 EUSART 接收器使能与否。可以通过普通端口读 RX/DT 引脚数据, 但却无法使用该端口锁存输出数据。

2: 如果 TXEN 使能位置 1, TXIF 发送器中断标志位会置 1。

12.1.1.2 发送数据

向 TXREG 寄存器写入一个字符, 以启动发送。如果这是第一个字符, 或者前一个字符已经完全从 TSR 中移出, TXREG 中的数据会立即发送给 TSR 寄存器。如果 TSR 中仍保存全部或部分前一字符, 新的字符数据将保存在 TXREG 中, 直到发送完前一字符的停止位为止。然后, 在停止位发送完毕后经过一个 T_{CY} , TXREG 中待处理的数据将被传输到 TSR。当数据从 TXREG 传输至 TSR 后, 立即开始进行起始位、数据位和停止位序列的发送。

12.1.1.3 发送中断标志

只要使能 EUSART 发送器且 TXREG 中没有待发送数据, 就将 PIR1 寄存器的 TXIF 中断标志位置 1。换句话说, 只有当 TSR 忙于处理字符和 TXREG 中有排队等待发送的新字符时, TXIF 位才处于清零状态。写 TXREG 时, 不立即清零 TXIF 标志位。TXIF 在写指令后的第 2 个指令周期清零。在写 TXREG 后立即查询 TXIF 会返回无效结果。TXIF 为只读位, 不能由软件置 1 或清零。

可通过将 PIE1 寄存器的 TXIE 中断允许位置 1 允许 TXIF 中断。然而, 只要 TXREG 为空, 不管 TXIE 允许位的状态如何都会将 TXIF 标志位置 1。

如果要在发送数据时使用中断, 只有在有待发送数据时, 才将 TXIE 位置 1。当将待发送的最后一个字符写入 TXREG 后, 将 TXIE 中断允许位清零。

PIC16F882/883/884/886/887

12.1.1.4 TSR 状态

TXSTA 寄存器的 TRMT 位指示 TSR 寄存器的状态。TRMT 位为只读位。当 TSR 寄存器为空时，TRMT 位被置 1，当有字符从 TXREG 传输到 TSR 寄存器时，TRMT 被清零。TRMT 位保持清零状态，直到所有位从 TSR 寄存器移出为止。没有任何中断逻辑与该位有关，所以用户必须查询该位来确定 TSR 位的状态。

注： TSR 寄存器并未映射到数据存储中，因此用户不能直接访问它。

12.1.1.5 发送 9 位字符

EUSART 支持 9 位字符发送。当 TXSTA 寄存器的 TX9 位置 1 时，EUSART 将移出每个待发送字符的 9 位。TXSTA 寄存器的 TX9D 位为第 9 位，即最高数据位。当发送 9 位数据时，必须在将 8 个最低位写入 TXREG 之前，写 TX9D 数据位。在写入 TXREG 寄存器后会立即将 9 个数据位传输到 TSR 移位寄存器。

使用多个接收器时可使用特殊的 9 位地址模式。请参见第 12.1.2.7 节“地址检测”获得有关地址模式的更多信息。

12.1.1.6 设置异步发送：

1. 初始化 SPBRGH 和 SPBRG 这对寄存器以及 BRGH 和 BRG16 位，以获得所需的波特率（见第 12.3 节“EUSART 波特率发生器 (BRG)”）。
2. 通过将 SYNC 位清零并将 SPEN 位置 1 使能异步串口。
3. 如果需要 9 位发送，将 TX9 控制位置 1。当接收器被设置为进行地址检测时，将数据位的第 9 位置 1，指示 8 个最低数据位为地址。
4. 将 TXEN 控制位置 1，使能发送；这将导致 TXIF 中断标志位置 1。
5. 如果需要中断，将 TXIE 中断允许位置 1；如果 INTCON 寄存器的 GIE 和 PEIE 位也置 1 将立即产生中断。
6. 若选择发送 9 位数据，第 9 位应该被装入 TX9D 数据位。
7. 将 8 位数据装入 TXREG 寄存器开始发送数据。

图 12-3: 异步发送

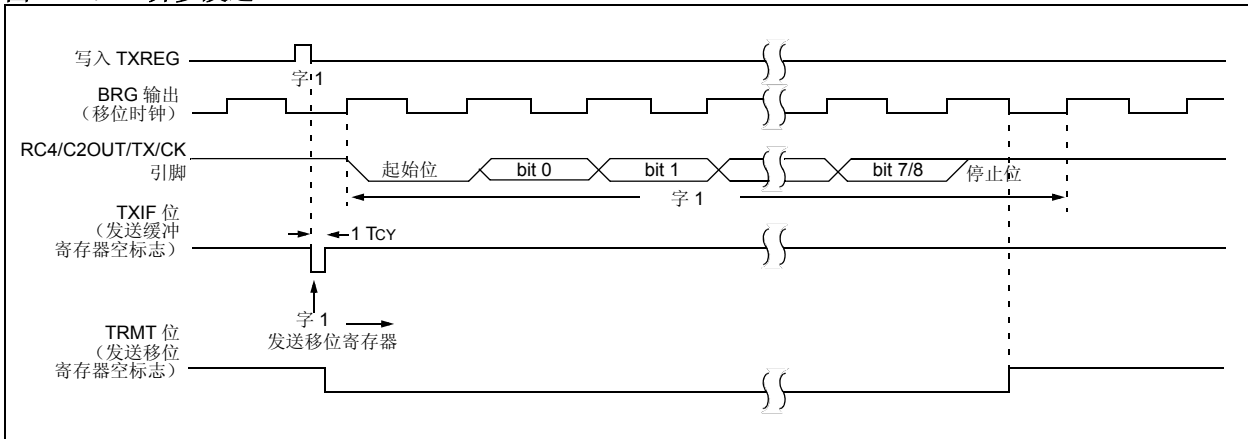
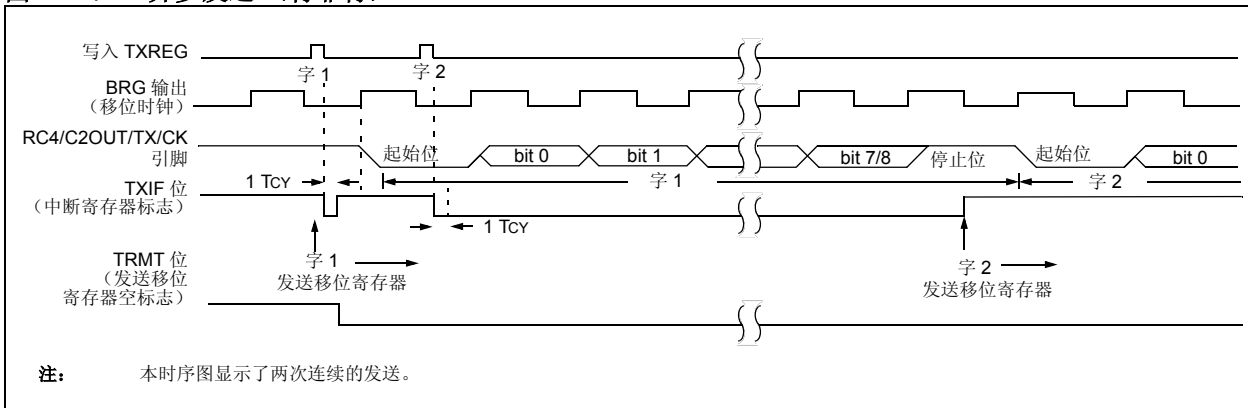


图 12-4: 异步发送 (背靠背)



注： 本时序图显示了两次连续的发送。

PIC16F882/883/884/886/887

表 12-1: 与异步发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
RCREG	EUSART 接收数据寄存器								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
TXREG	EUSART 发送数据寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

图注: x = 未知, - = 未实现 (读为 0)。异步发送不使用阴影单元。

PIC16F882/883/884/886/887

12.1.2 EUSART 异步接收器

异步模式通常用于 RS-232 系统。图 12-2 给出了接收器的框图。在 RX/DT 引脚上接收数据和驱动数据恢复电路。数据恢复电路实际上是一个以 16 倍波特率为工作频率的高速移位器，而串行接收移位寄存器（Receive Shift Register, RSR）则以比特率工作。当字符的全部 8 位或 9 位数据位被移入后，立即将它们传输到一个 2 字符的先入先出（FIFO）缓冲器。FIFO 缓冲器允许接收 2 个完整的字符和第 3 个字符的起始位，然后必须由软件将接收到的数据提供给 EUSART 接收器。FIFO 和 RSR 寄存器不能直接由软件访问。通过 RCREG 寄存器访问接收到的数据。

12.1.2.1 使能接收器

通过配置如下三个控制位使能 EUSART 接收器，以用于异步操作。

- CREN = 1
- SYNC = 0
- SPEN = 1

假设所有其他 EUSART 控制位都处于默认状态。

将 RCSTA 寄存器的 CREN 位置 1，使能 EUSART 接收器电路。将 TXSTA 寄存器的 SYNC 位清零，配置 EUSART 以用于异步操作。将 RCSTA 寄存器的 SPEN 位置 1，使能 EUSART 并自动将 RX/DT I/O 引脚配置为输入引脚。如果 RX/DT 引脚与模拟外设共用，必须清零相应的 ANSEL 位禁止模拟 I/O 功能。

注： 当将 SPEN 位置 1，TX/CK I/O 引脚被自动配置为输出引脚时，无需考虑相应 TRIS 位的状态以及 EUSART 发送器使能与否。端口锁存器与输出驱动器是断开的，从而不能将 TX/CK 引脚用作通用输出引脚。

12.1.2.2 接收数据

接收器数据恢复电路在第一个位的下降沿开始接收字符。第一个位，通常称为起始位，始终为 0。由数据恢复电路计数半个位时间，到起始位的中心位置，校验该位是否仍为零。如果该位不为零，数据恢复电路放弃接收该字符，而不会产生错误，并且继续查找起始位的下降沿。如果起始位零校验通过，则数据恢复电路计数一个完整的位时间，到达下一位的中心位置。由多检测电路对该位进行采样，将相应的采样结果 0 或 1 移入 RSR。重复该过程，直到完成所有数据位的采样并将其全部移入 RSR 寄存器。测量最后一个位的时间并采样其电平。此位为停止位，总是为 1。如果数据恢复电路在停止位的位置采样到 0，则该字符的帧错误标志将置 1，反之，该字符的帧错误标志会清零。请参见第 12.1.2.4 节“接收帧错误”获得有关帧错误的更多相关信息。

当接收到所有数据位和停止位后，RSR 中的字符会被立即传输到 EUSART 的接收 FIFO 并将 PIR1 寄存器的 RCIF 中断标志位置 1。通过读 RCREG 寄存器将 FIFO 最顶端的字符移出 FIFO。

注： 如果接收 FIFO 溢出，则不能再继续接收其他字符，直到溢出条件被清除。请参见第 12.1.2.5 节“接收溢出错误”获得有关溢出错误的更多相关信息。

12.1.2.3 接收中断

只要使能 EUSART 接收器且在接收 FIFO 中没有未读数据，PIR1 寄存器中的 RCIF 中断标志位就会置 1。RCIF 中断标志位为只读，不能由软件置 1 或清零。

通过将下列位置 1 允许 RCIF 中断：

- PIE1 寄存器的 RCIE 中断允许位
- INTCON 寄存器的 PEIE 外设中断允许位
- INTCON 寄存器的 GIE 全局中断允许位

如果 FIFO 中有未读数据，无论中断允许位的状态如何，都会将 RCIF 中断标志位置 1。

PIC16F882/883/884/886/887

12.1.2.4 接收帧错误

接收 FIFO 缓冲器中的每个字符都有一个相应的帧错误状态位。帧错误指示未在预期的时间内接收到停止位。由 RCSTA 寄存器的 FERR 位获取帧错误状态。FERR 位代表接收 FIFO 最顶端未读字符的状态。因此，必须在读 RCREG 寄存器之前读 FERR 位。

FERR 位为只读位，且只能用于接收 FIFO 的最顶端未读字符。帧错误 (FERR = 1) 并不会阻止接收更多的字符。无需清零 FERR 位。从 FIFO 缓冲器读下一字符会使 FIFO 指针前进至下一字符和下一个相应的帧错误。

清零 RCSTA 寄存器的 SPEN 位会复位 EUSART，并强制清零 FERR 位。清零 RCSTA 寄存器的 CREN 位不影响 FERR 位。帧错误本身不会产生中断。

注： 如果接收 FIFO 缓冲器中所有接收到的字符都有帧错误，重复读 RCREG 不会清零 FERR 位。
--

12.1.2.5 接收溢出错误

接收 FIFO 缓冲器可以保存 2 个字符。但如果在访问 FIFO 之前，接收到完整的第 3 个字符，则会产生溢出错误。此时，RCSTA 寄存器的 OERR 位会置 1。可以读取 FIFO 缓冲器内的字符，但是在错误清除之前，不能再接收其他字符。可以通过清零 RCSTA 寄存器的 CREN 位或通过清零 RCSTA 寄存器的 SPEN 位使 EUSART 复位来清除错误。

12.1.2.6 接收 9 位字符

EUSART 支持 9 位数据接收。将 RCSTA 寄存器的 RX9 位置 1 时，EUSART 将接收到的每个字符的 9 位移入 RSR。RCSTA 寄存器的 RX9D 位是接收 FIFO 顶端未读字符的第 9 位，同时也是最高数据位。当从接收 FIFO 缓冲器读取 9 位数据时，必须在读 RCREG 中的低 8 位之前，读取 RX9D 数据位。

12.1.2.7 地址检测

当多个接收器共享同一传输线时（如在 RS-485 系统中），可使用特殊地址检测模式。将 RCSTA 寄存器的 ADDEN 位置 1，使能地址检测模式。

地址检测要求接收 9 位字符。使能地址检测后，只有第 9 位数据位被置 1 的字符可以被传输到接收 FIFO 缓冲器，从而使 RCIF 中断标志位置 1。所有其他字符将被忽略。

由用户软件判断接收到的地址字符是否与其匹配。如果匹配，用户软件必须在下一个停止位产生之前，清零 ADDEN 位以禁止地址检测。当用户软件检测到信息的末尾（由所使用的信息协议判断）时，由软件将 ADDEN 位置 1，从而使接收器返回地址检测模式。

PIC16F882/883/884/886/887

12.1.2.8 异步接收设置

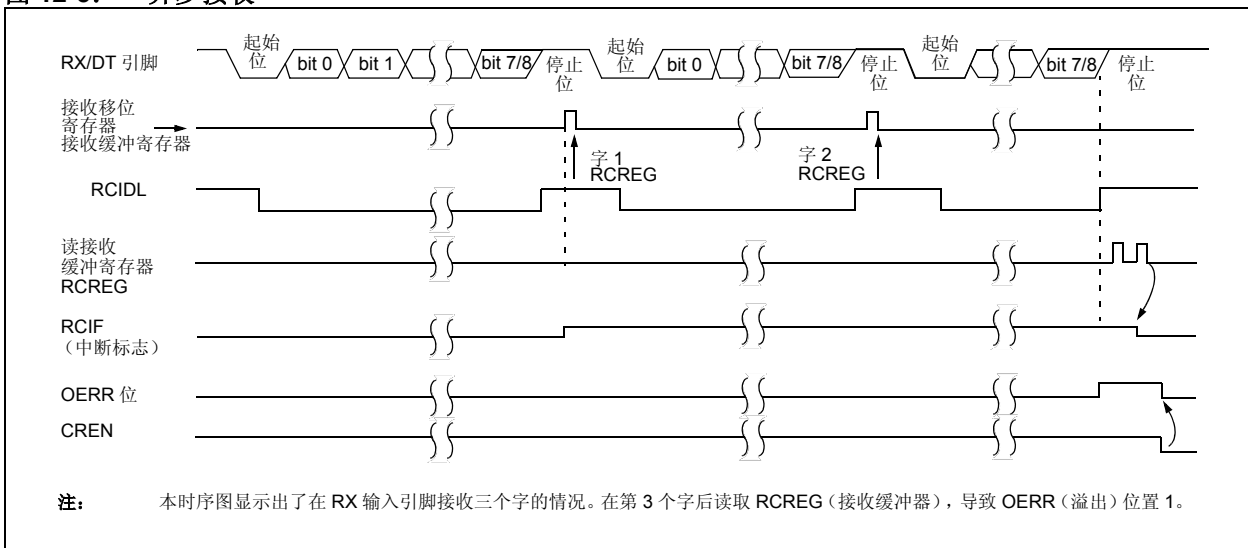
1. 初始化 SPBRGH 和 SPBRG 这对寄存器以及 BRGH 和 BRG16 位，以获得所需的波特率（见第12.3节“EUSART波特率发生器(BRG)”）。
2. 将 SPEN 位置 1，使能串行端口。必须清零 SYNC 位以执行异步操作。
3. 如果需要中断，将 RCIE 中断允许位和 INTCON 寄存器的 GIE 和 PEIE 位置 1。
4. 如果需要接收 9 位数据，将 RX9 位置 1。
5. 将 CREN 位置 1 使能接收。
6. 当一个字符从 RSR 传输到接收缓冲器时，将 RCIF 中断标志位置 1。如果 RCIE 中断允许位也置 1 还将产生中断。
7. 读 RCSTA 寄存器获取错误标志位和第 9 位数据位（如果使能 9 位数据接收）。
8. 读 RCREG 寄存器，从接收缓冲器获取接收到的 8 个低数据位。
9. 如果发生溢出，通过清零 CREN 接收器使能位清零 OERR 标志。

12.1.2.9 9 位地址检测模式设置

此模式通常用在 RS-485 系统中。要设置使能地址检测的异步接收：

1. 初始化 SPBRGH 和 SPBRG 这对寄存器以及 BRGH 和 BRG16 位，以获得所需的波特率（见第12.3节“EUSART波特率发生器(BRG)”）。
2. 将 SPEN 位置 1，使能串行端口。必须清零 SYNC 位以执行异步操作。
3. 如果需要中断，将 RCIE 中断允许位和 INTCON 寄存器的 GIE 和 PEIE 位置 1。
4. 将 RX9 位置 1，使能 9 位数据接收。
5. 将 ADDEN 位置 1，使能地址检测。
6. 将 CREN 位置 1 使能接收。
7. 当一个第 9 位置 1 的字符从 RSR 传输到接收缓冲器时，将 RCIF 中断标志位置 1。如果 RCIE 中断允许位也置 1 还将产生中断。
8. 读 RCSTA 寄存器获取错误标志位。第 9 个数据位始终置 1。
9. 读 RCREG 寄存器，从接收缓冲器获取接收到的 8 个低数据位。由软件判断此地址是否为本地器件的地址。
10. 如果发生溢出，通过清零 CREN 接收器使能位清零 OERR 标志。
11. 如果是对当前器件寻址，将 ADDEN 位清零以允许所有接收到的数据进入接收缓冲器并产生中断。

图 12-5: 异步接收



PIC16F882/883/884/886/887

表 12-2: 与异步接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
RCREG	EUSART 接收数据寄存器								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
TXREG	EUSART 发送数据寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

图注: x = 未知, - = 未实现 (读为 0)。异步接收不使用阴影单元。

PIC16F882/883/884/886/887

12.2 异步操作时的时钟准确度

由厂家校准内部振荡电路（INTOSC）的输出。但在VDD或温度变化时，INTOSC会发生频率漂移，从而会直接影响异步波特率。有两种方法可用于调整波特率时钟，但这两种方法都需要某种类型的参考时钟源。

第一种（首选）方法使用OSCTUNE寄存器调整INTOSC输出。调整OSCTUNE寄存器的值可以微调系统时钟源的频率。请参见第4.5节“内部时钟模式”获得更多相关信息。

另一种方法为调整波特率发生器的值。此操作可由自动波特率特性自动完成（见第12.3.1节“自动波特率检测”）。当调整波特率发生器时，调节的分辨率可能不足以弥补外设时钟频率的渐进变化。

寄存器 12-1: TXSTA: 发送状态和控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN ⁽¹⁾	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7						bit 0	

图注:

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

- bit 7 **CSRC:** 时钟源选择位
 异步模式:
 任意值
 同步模式:
 1 = 主控模式（由内部 BRG 产生时钟信号）
 0 = 从动模式（由外部时钟源产生时钟）
- 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⁽¹⁾
 1 = 使能发送
 0 = 禁止发送
- bit 4 **SYNC:** EUSART 模式选择位
 1 = 同步模式
 0 = 异步模式
- bit 3 **SENDB:** 发送间隔字符位
 异步模式:
 1 = 在下次发送时发送同步间隔字符（在完成时由硬件清零）
 0 = 同步间隔字符发送完成
 同步模式:
 任意值
- bit 2 **BRGH:** 高波特率选择位
 异步模式:
 1 = 高速
 0 = 低速
 同步模式:
 在此模式下未用
- bit 1 **TRMT:** 发送移位寄存器状态位
 1 = TSR 为空
 0 = TSR 为满
- bit 0 **TX9D:** 发送数据的第 9 位
 可以是地址 / 数据位或奇偶校验位。

注 1: 同步模式下，SREN/CREN 会覆盖 TXEN 的值。

PIC16F882/883/884/886/887

寄存器 12-2: **RCSTA: 接收状态和控制寄存器** ⁽¹⁾

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	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

图注:

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

- bit 7 SPEN:** 串行端口使能位
 1 = 使能串行端口 (将 RX/DT 和 TX/CK 引脚配置为串行端口引脚)
 0 = 禁止串行端口 (保持在复位状态)
- 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:
 任意值
同步主控模式:
 1 = 使能单字节接收
 0 = 禁止单字节接收
 接收完成后清零该位。
同步从动模式:
 任意值
- bit 4 CREN:** 连续接收使能位
异步模式:
 1 = 使能接收
 0 = 禁止接收
同步模式:
 1 = 使能连续接收直到清零 CREN 使能位 (CREN 覆盖 SREN)
 0 = 禁止连续接收
- bit 3 ADDEN:** 地址检测使能位
异步 9 位模式 (RX9 = 1):
 1 = 当 RSR<8> 置 1 时, 使能地址检测、允许中断并装载接收缓冲器。
 0 = 禁止地址检测, 所有字节都可以被接收, 第 9 位可被用作奇偶校验位。
异步 8 位模式 (RX9 = 0):
 任意值
- bit 2 FERR:** 帧错误位
 1 = 帧错误 (可通过读 RCREG 寄存器更新并接收下一个有效字节)
 0 = 没有帧错误
- bit 1 OERR:** 溢出错误位
 1 = 溢出错误 (可通过清零 CREN 位清零)
 0 = 没有溢出错误
- bit 0 RX9D:** 接收到数据的第 9 位
 此位可以是地址 / 数据位或奇偶校验位, 必须由用户固件计算得到。

PIC16F882/883/884/886/887

寄存器 12-3: BAUDCTL: 波特率控制寄存器

R-0	R-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

图注:

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

- bit 7 **ABDOVF:** 波特率检测溢出位
异步模式:
 1 = 波特率定时器溢出
 0 = 波特率定时器没有溢出
同步模式:
 任意值
- bit 6 **RCIDL:** 接收空闲标志位
异步模式:
 1 = 接收器空闲
 0 = 已接收到起始位, 接收器正在接收数据
同步模式:
 任意值
- bit 5 **未实现:** 读为 0
- bit 4 **SCKP:** 同步时钟极性选择位
异步模式:
 1 = 将数据字符的电平取反后发送到 RB7/TX/CK 引脚
 0 = 直接将数据字符发送到 RB7/TX/CK 引脚
同步模式:
 1 = 在时钟上升沿传输数据
 0 = 在时钟下降沿传输数据
- bit 3 **BRG16:** 16 位波特率发生器位
 1 = 使用 16 位波特率发生器
 0 = 使用 8 位波特率发生器
- bit 2 **未实现:** 读为 0
- bit 1 **WUE:** 唤醒使能位
异步模式:
 1 = 接收器等待时钟下降沿。此时不接收任何字节, RCIF 位置 1。RCIF 置 1 后, 自动清零 WUE。
 0 = 接收器正常工作
同步模式:
 任意值
- bit 0 **ABDEN:** 自动波特率检测使能位
异步模式:
 1 = 使能自动波特率检测模式 (检测完成后清零)
 0 = 禁止自动波特率检测模式
同步模式:
 任意值

PIC16F882/883/884/886/887

12.3 EUSART 波特率发生器 (BRG)

波特率发生器 (BRG) 是一个 8 位或 16 位定时器，专用于支持 EUSART 的异步和同步工作模式。默认情况下，BRG 工作在 8 位模式下。将 BAUDCTL 寄存器中的 BRG16 位置 1 可选择 16 位模式。

SPBRGH 和 SPBRG 寄存器对决定自由运行的波特率定时器的周期。在异步模式下，由 TXSTA 寄存器的 BRGH 寄存器和 BAUDCTL 寄存器的 BRG16 位决定波特率周期的倍数。在同步模式下，忽略 BRGH 位。

表 12-3 包含了计算波特率的公式。例 12-1 给出了一个计算波特率和波特率误差的示例。

表 12-3 中给出了已经计算好的各种异步模式下的典型波特率和波特率误差值，可便于您使用。使用高波特率 (BRGH = 1) 或 16 位 BRG (BRG16 = 1) 有利于减少波特率误差。16 位 BRG 模式用来在快速振荡频率条件下实现低波特率。

向 SPBRGH 和 SPBRG 寄存器对写入新值会导致 BRG 定时器复位 (或清零)。这可以确保 BRG 无需等待定时器溢出就可以输出新的波特率。

如果系统时钟在有效的接收过程中发生了变化，可能会产生接收错误或导致数据丢失。为了避免此问题，应该检查 RCIDL 位的状态以确保改变系统时钟之前，接收操作处于空闲状态。

例 12-1: 计算波特率误差

对于 Fosc 为 16 MHz，目标波特率为 9600 bps，异步模式，采用 8 位 BRG 的器件：

$$\text{目标波特率} = \frac{F_{osc}}{64([\text{SPBRGH}:\text{SPBRG}] + 1)}$$

求解 SPBRGH:SPBRG:

$$\begin{aligned} X &= \frac{\frac{F_{osc}}{\text{目标波特率}}}{64} - 1 \\ &= \frac{\frac{16000000}{9600}}{64} - 1 \\ &= [25.042] = 25 \end{aligned}$$

$$\begin{aligned} \text{计算波特率} &= \frac{16000000}{64(25 + 1)} \\ &= 9615 \end{aligned}$$

$$\begin{aligned} \text{误差} &= \frac{\text{计算波特率} - \text{目标波特率}}{\text{目标波特率}} \\ &= \frac{(9615 - 9600)}{9600} = 0.16\% \end{aligned}$$

表 12-3: 波特率公式

配置位			BRG/EUSART 模式	波特率公式
SYNC	BRG16	BRGH		
0	0	0	8 位 / 异步	Fosc/[64 (n+1)]
0	0	1	8 位 / 异步	Fosc/[16 (n+1)]
0	1	0	16 位 / 异步	
0	1	1	16 位 / 异步	Fosc/[4 (n+1)]
1	0	x	8 位 / 同步	
1	1	x	16 位 / 同步	

图注: x = 任意值, n = SPBRGH:SPBRG 寄存器的值

表 12-4: 与波特率发生器相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

图注: x = 未知, - = 未读 (读为 0)。波特率发生器不使用阴影单元。

PIC16F882/883/884/886/887

表 12-5: 异步模式下的波特率

目标 波特率	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz			Fosc = 8.000 MHz		
	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	1221	1.73	255	1200	0.00	239	1200	0.00	143	1202	0.16	103
2400	2404	0.16	129	2400	0.00	119	2400	0.00	71	2404	0.16	51
9600	9470	-1.36	32	9600	0.00	29	9600	0.00	17	9615	0.16	12
10417	10417	0.00	29	10286	-1.26	27	10165	-2.42	16	10417	0.00	11
19.2k	19.53k	1.73	15	19.20k	0.00	14	19.20k	0.00	8	—	—	—
57.6k	—	—	—	57.60k	0.00	7	57.60k	0.00	2	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

目标 波特率	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)
300	300	0.16	207	300	0.00	191	300	0.16	103	300	0.16	51
1200	1202	0.16	51	1200	0.00	47	1202	0.16	25	1202	0.16	12
2400	2404	0.16	25	2400	0.00	23	2404	0.16	12	—	—	—
9600	—	—	—	9600	0.00	5	—	—	—	—	—	—
10417	10417	0.00	5	—	—	—	10417	0.00	2	—	—	—
19k	—	—	—	19.20k	0.00	2	—	—	—	—	—	—
57.6k	—	—	—	57.60k	0.00	0	—	—	—	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

目标 波特率	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz			Fosc = 8.000 MHz		
	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	2404	0.16	207
9600	9615	0.16	129	9600	0.00	119	9600	0.00	71	9615	0.16	51
10417	10417	0.00	119	10378	-0.37	110	10473	0.53	65	10417	0.00	47
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35	19231	0.16	25
57.6k	56.82k	-1.36	21	57.60k	0.00	19	57.60k	0.00	11	55556	-3.55	8
115.2k	113.64	-1.36	10	115.2k	0.00	9	115.2k	0.00	5	—	—	—

PIC16F882/883/884/886/887

表 12-5: 异步模式下的波特率 (续)

目标波特率	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)
300	—	—	—	—	—	—	—	—	—	300	0.16	207
1200	1202	0.16	207	1200	0.00	191	1202	0.16	103	1202	0.16	51
2400	2404	0.16	103	2400	0.00	95	2404	0.16	51	2404	0.16	25
9600	9615	0.16	25	9600	0.00	23	9615	0.16	12	—	—	—
10417	10417	0.00	23	10473	0.53	21	10417	0.00	11	10417	0.00	5
19.2k	19.23k	0.16	12	19.2k	0.00	11	—	—	—	—	—	—
57.6k	—	—	—	57.60k	0.00	3	—	—	—	—	—	—
115.2k	—	—	—	115.2k	0.00	1	—	—	—	—	—	—

目标波特率	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz			Fosc = 8.000 MHz		
	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)
300	300.0	-0.01	4166	300.0	0.00	3839	300.0	0.00	2303	299.9	-0.02	1666
1200	1200	-0.03	1041	1200	0.00	959	1200	0.00	575	1199	-0.08	416
2400	2399	-0.03	520	2400	0.00	479	2400	0.00	287	2404	0.16	207
9600	9615	0.16	129	9600	0.00	119	9600	0.00	71	9615	0.16	51
10417	10417	0.00	119	10378	-0.37	110	10473	0.53	65	10417	0.00	47
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35	19.23k	0.16	25
57.6k	56.818	-1.36	21	57.60k	0.00	19	57.60k	0.00	11	55556	-3.55	8
115.2k	113.63	-1.36	10	115.2k	0.00	9	115.2k	0.00	5	—	—	—

目标波特率	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)
300	300.1	0.04	832	300.0	0.00	767	299.8	-0.108	416	300.5	0.16	207
1200	1202	0.16	207	1200	0.00	191	1202	0.16	103	1202	0.16	51
2400	2404	0.16	103	2400	0.00	95	2404	0.16	51	2404	0.16	25
9600	9615	0.16	25	9600	0.00	23	9615	0.16	12	—	—	—
10417	10417	0.00	23	10473	0.53	21	10417	0.00	11	10417	0.00	5
19.2k	19.23k	0.16	12	19.20k	0.00	11	—	—	—	—	—	—
57.6k	—	—	—	57.60k	0.00	3	—	—	—	—	—	—
115.2k	—	—	—	115.2k	0.00	1	—	—	—	—	—	—

PIC16F882/883/884/886/887

表 12-5: 异步模式下的波特率 (续)

目标 波特率	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz			Fosc = 8.000 MHz		
	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)
300	300.0	0.00	16665	300.0	0.00	15359	300.0	0.00	9215	300.0	0.00	6666
1200	1200	-0.01	4166	1200	0.00	3839	1200	0.00	2303	1200	-0.02	1666
2400	2400	0.02	2082	2400	0.00	1919	2400	0.00	1151	2401	0.04	832
9600	9597	-0.03	520	9600	0.00	479	9600	0.00	287	9615	0.16	207
10417	10417	0.00	479	10425	0.08	441	10433	0.16	264	10417	0	191
19.2k	19.23k	0.16	259	19.20k	0.00	239	19.20k	0.00	143	19.23k	0.16	103
57.6k	57.47k	-0.22	86	57.60k	0.00	79	57.60k	0.00	47	57.14k	-0.79	34
115.2k	116.3k	0.94	42	115.2k	0.00	39	115.2k	0.00	23	117.6k	2.12	16

目标 波特率	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1											
	Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)
300	300.0	0.01	3332	300.0	0.00	3071	299.9	-0.02	1666	300.1	0.04	832
1200	1200	0.04	832	1200	0.00	767	1199	-0.08	416	1202	0.16	207
2400	2398	0.08	416	2400	0.00	383	2404	0.16	207	2404	0.16	103
9600	9615	0.16	103	9600	0.00	95	9615	0.16	51	9615	0.16	25
10417	10417	0.00	95	10473	0.53	87	10417	0.00	47	10417	0.00	23
19.2k	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	25	19.23k	0.16	12
57.6k	56.82k	2.12	16	57.60k	0.00	15	55.56k	-3.55	8	—	—	—
115.2k	111.1k	-3.55	8	115.2k	0.00	7	—	—	—	—	—	—

PIC16F882/883/884/886/887

12.3.1 自动波特率检测

EUSART 模块支持波特率自动检测和校准。

在自动波特率检测 (Auto-Baud Rate Detect, ABD) 模式下, BRG 的时钟是反向的。不是由 BRG 为 RX 输入信号提供时钟源, 而是由 RX 信号为 BRG 定时。波特率发生器用来测量接收 55h 字符的周期 (ASCII “U”, 也是 LIN 总线的同步字符), 该字符的特性在于它包含 5 个上升沿 (包括停止位边沿)。

将 BAUDCTL 寄存器的 ABDEN 位置 1, 启动自动校准序列 (图 12-6)。发生 ABD 序列时, EUSART 状态机保持空闲状态。在起始位后, 在接收信号的第一个上升沿, SPBRG 使用 BRG 计数器时钟开始计数, 如表 12-6 所示。在 RX 引脚传输了 8 个位之后, 产生第 5 个上升沿。此时, 会将在相应 BRG 周期内累计的值保存到相应的 SPBRGH:SPBRG 这对寄存器中, 自动清零 ABDEN 位, 并将 RCIF 中断标志位置 1。要清除 RCIF 中断, 需要读取 RCREG 中的值。应丢弃 RCREG 中的内容。当在不需要使用 SPBRGH 寄存器的模式下进行校准时, 用户可以通过检查 SPBRGH 寄存器的值是否为 00h 位校验 SPBRG 寄存器是否溢出。

由 BRG16 和 BRGH 位决定 BRG 自动波特率时钟, 如表 12-6 所示。在 ABD 过程中, 无论 BRG16 位的设置如何, SPBRGH 和 SPBRG 寄存器均用作 16 位计数器。

器。当校准波特率周期时, 以 BRG 基本时钟频率的 1/8 设置 SPBRGH 和 SPBRG 寄存器。当将波特率设置为全速时, 字节测量结果为平均位时间。

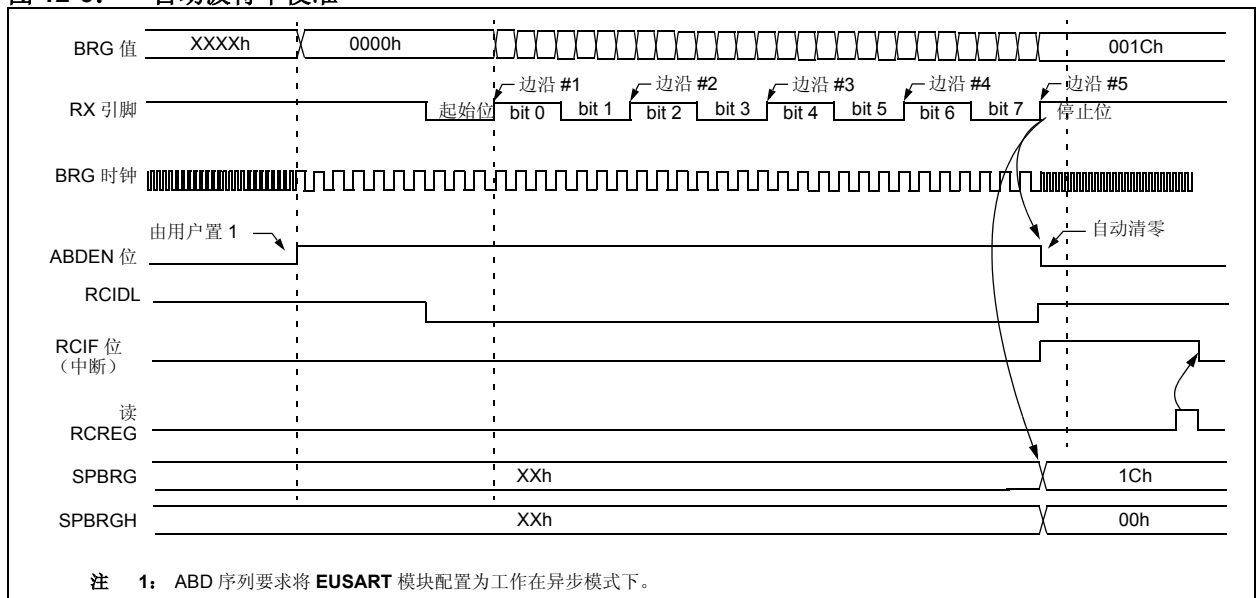
- 注 1:** 如果 WUE 位与 ABDEN 位同时置 1, 自动波特率检测会在间隔字符之后的字节开始 (见第 12.3.2 节 “接收到间隔字符时自动唤醒”)。
- 注 2:** 由用户来判断进入的字符波特率是否处于所选 BRG 时钟源范围内。由于位误差率的原因, 某些振荡频率和 EUSART 波特率的组合是无法实现的。在使用自动波特率检测功能时, 必须考虑系统总体时序和通信波特率。
- 注 3:** 完成自动波特率序列后, 计算得到的波特率将为实际波特率加 1。

表 12-6: BRG 计数器时钟速率

BRG16	BRGH	BRG 基本时钟	BRGABD 时钟
0	0	Fosc/64	Fosc/512
0	1	Fosc/16	Fosc/128
1	0	Fosc/16	Fosc/128
1	1	Fosc/4	Fosc/32

注: 在 ABD 过程中, 无论 BRG16 位的设置如何, SPBRGH 和 SPBRG 寄存器均用作 16 位计数器。

图 12-6: 自动波特率校准



PIC16F882/883/884/886/887

12.3.2 接收到间隔字符时自动唤醒

在休眠模式下，EUSART 的所有时钟都会暂停。因此，波特率发生器处于无效状态，且无法进行正确的字符接收。自动唤醒功能允许在 RX/DT 线上活动时唤醒控制器，只有在 EUSART 工作在异步模式下时才可使用该功能。

通过将 BAUDCTL 寄存器的 WUE 位置 1，使能自动唤醒功能。置 1 后，将禁止 RX/DT 上的正常接收序列，并且 EUSART 保持在空闲状态，监视唤醒事件而与 CPU 的工作模式无关。唤醒事件是指 RX/DT 线上从高电平到低电平的转换。（这与“同步间隔”字符或 LIN 协议唤醒信号字符的启动条件一致）。

EUSART 模块在检测到唤醒事件时，产生一个 RCIF 中断。在 CPU 正常工作模式下，中断会与 Q 时钟同步产生（图 12-7）；如果器件处于休眠模式，则两者不同步（图 12-8）。通过读 RCREG 寄存器清除中断条件。

WUE 会被间隔字符末尾的 RX 线从低电平向高电平的转换自动清零。这将通知用户“同步间隔”事件结束。此时，EUSART 模块处于空闲模式，等待接收下一字符。

12.3.2.1 特别注意事项

间隔字符

为了避免唤醒事件期间的字符错误或字符分段，唤醒字符必须为全零。

如果使能唤醒功能，无论数据流的低电平持续时间的长短，该功能都可起作用。如果将 WUE 位置 1，且收到一个有效的非零字符，从起始位到第一个上升沿之间的低电平时间被解释为唤醒事件。将以分段字符的方式接收当前字符的剩余位，而后续的字符会导致帧错误或溢出错误。

因此，发送的初始字符必须为全 0，且其持续时间必须为 10 个或更多的位时间。针对 LIN 总线建议为 13 个位时间，而对于标准 RS-232 器件可为任何数量的位时间。

振荡器起振时间

另外还必须考虑振荡器起振时间，尤其是在采用起振时间较长的振荡器（比如 LP、XT 或 HS/PLL 模式）的应用中。“同步间隔”（或唤醒信号）字符必须足够长，并且跟有足够长的时间间隔，以使选定振荡器有足够的时间起振并使 EUSART 正确初始化。

WUE 位

唤醒事件通过将 RCIF 位置 1 产生接收中断。在 RX/DT 的上升沿，由硬件清零 WUE 位。通过在软件中读 RCREG 寄存器并丢弃其内容清零中断条件。

要确保没有丢失数据，在将 WUE 位置 1 之前，应检查 RCIDL 位以验证是否正在进行接收。如果不在进行接收，则可将 WUE 位置 1，使器件立即进入休眠模式。

图 12-7: 正常工作下的自动唤醒位 (WUE) 时序

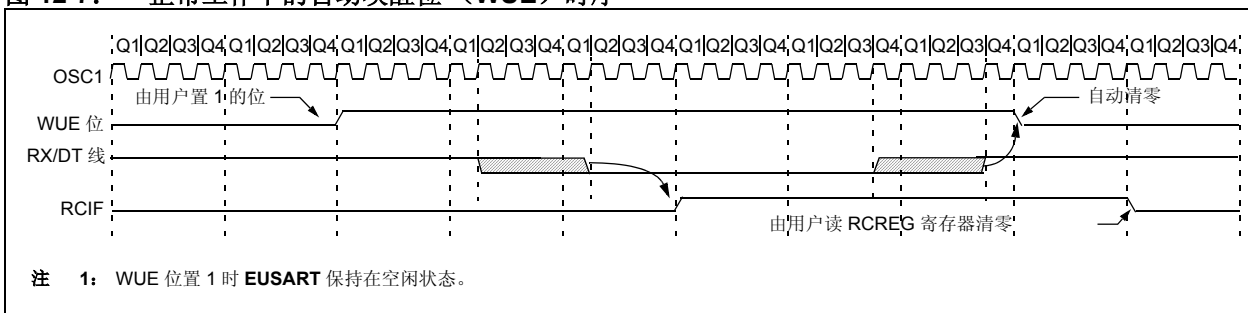
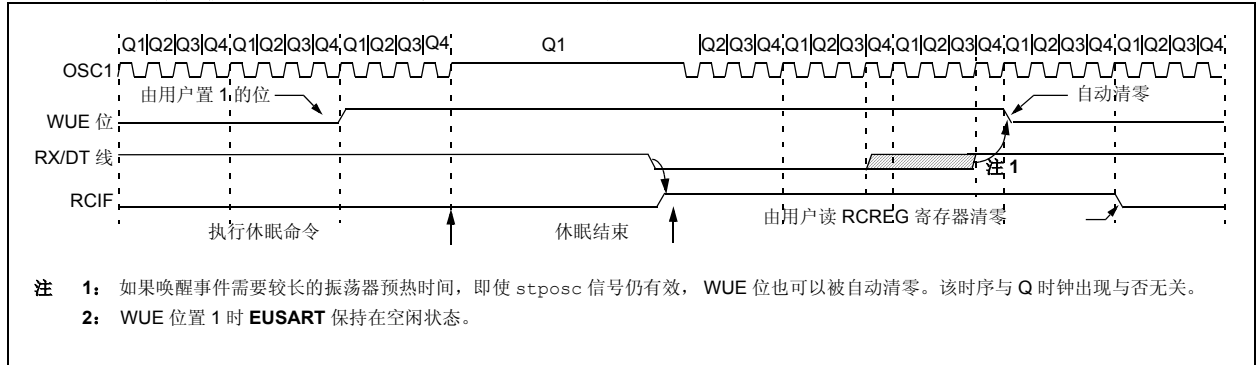


图 12-8: 休眠模式下的自动唤醒位 (WUE) 时序



12.3.3 间隔字符时序

EUSART 模块能够发送符合 LIN 总线标准的特殊间隔字符序列。发送的间隔字符包括 1 个起始位, 后面跟有 12 个 0 位和一个停止位。

要发送间隔字符, 应将 TXSTA 寄存器的 SENDB 和 TXEN 位置 1。之后对 TXREG 寄存器的写操作将启动间隔字符的发送。写入 TXREG 的数据值会被忽略, 并会发送全 0。

在发送了相应的停止位后, 硬件会自动将 SENDB 位清零。这样用户可以在发送完间隔字符 (在 LIN 规范中通常是同步字符) 后将下一个要发送的字节预先装入发送 FIFO。

正如其在正常发送操作中一样, TXSTA 寄存器的 TRMT 位表明发送正在进行还是处于空闲状态。关于发送间隔字符的时序, 请参见图 12-9。

12.3.3.1 间隔和同步发送序列

以下序列会发送一个报文帧头, 包括一个间隔字符和其后的自动波特率同步字节。此序列适用于典型的 LIN 总线主器件。

1. 将 EUSART 配置为所需的模式。
2. 将 TXEN 和 SENDB 位置 1, 以设置间隔字符。
3. 将无效字符装入 TXREG, 启动发送 (该值会被忽略)。
4. 将 55h 写入 TXREG, 以便把同步字符装入发送 FIFO 缓冲器。
5. 间隔字符发送后, 硬件会将 SENDB 位复位。开始发送同步字符。

当 TXIF 指示 TXREG 为空时, 下一个数据字节会写入 TXREG。

12.3.4 接收间隔字符

增强型 EUSART 模块接收间隔字符有两种方法。

第一种检测间隔字符的方法是使用 RCSTA 寄存器的 FERR 位和 RCREG 指示的接收数据。假设已将波特率发生器初始化为预期的波特率。

当

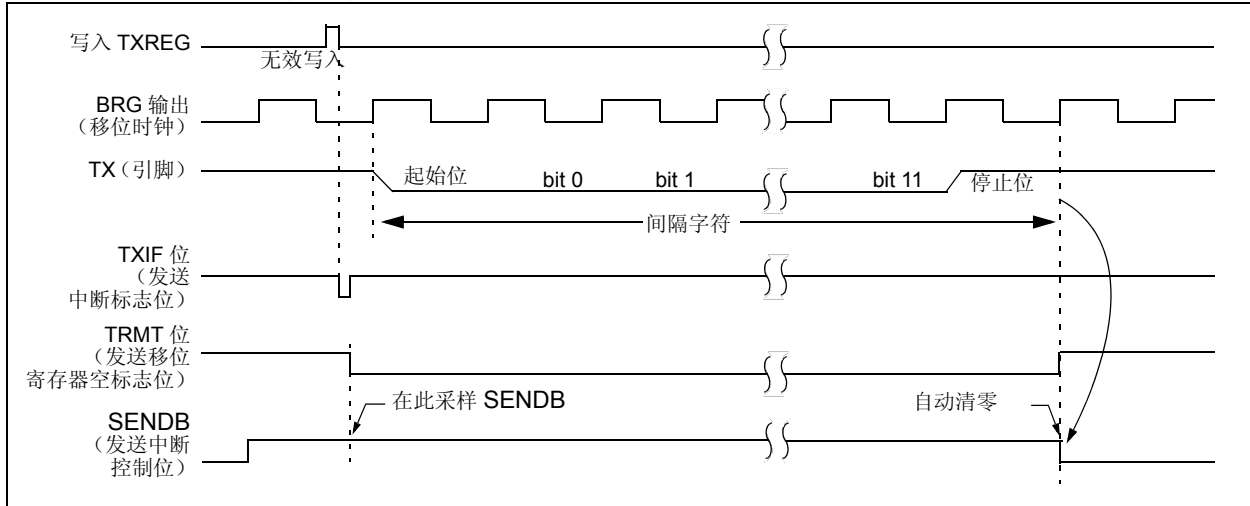
- RCIF 位置 1
- FERR 位置 1
- RCREG = 00h 时, 接收到间隔字符。

第二种方法使用第 12.3.2 节“接收到间隔字符时自动唤醒”中描述的自动唤醒功能。通过使能此功能, EUSART 将采样 RX/DT 上电平的下两个跳变, 产生一个 RCIF 中断, 接收下一个数据字节, 之后产生另一个中断。

请注意, 在间隔字符后, 用户通常希望使能自动波特率检测功能。无论使用哪种方法, 用户都可以在 EUSART 进入休眠模式之前, 将 BAUDCTL 寄存器的 ABDEN 位置 1。

PIC16F882/883/884/886/887

图 12-9: 发送间隔字符时序



PIC16F882/883/884/886/887

12.4 EUSART 同步模式

同步串行通信通常用在具有一个主控制器件和一个或多个从动器件的系统中。主控制器件包含产生波特率时钟所必需的电路，并为系统中的所有器件提供时钟。从动器件可以使用主控时钟，因此无需内部时钟发生电路。

在同步模式下，有 2 条信号线：双向数据线和时钟线。从动器件使用主控制器件提供的外部时钟，将数据串行移入或移出相应的接收和发送移位寄存器。因为使用双向数据线，所以同步操作只能采用半双工方式。半双工是指：主控制器件和从动器件都可以接收和发送数据，但是不能同时进行接收或发送。EUSART 既可以作为主控制器件，也可以作为从动器件。

同步发送无需使用起始位和停止位。

12.4.1 同步主控模式

下列位用来将 EUSART 配置为同步主控操作：

- SYNC = 1
- CSRC = 1
- SREN = 0（用于发送）； SREN = 1（用于接收）
- CREN = 0（用于发送）； CREN = 1（用于接收）
- SPEN = 1

将 TXSTA 寄存器的 SYNC 位置 1，可将 EUSART 配置用于同步操作。将 TXSTA 寄存器的 CSRC 位置 1，将器件配置为主控制器件。将 RCSTA 寄存器的 SREN 和 CREN 位清零，以确保器件处于发送模式，否则器件配置为接收模式。将 RCSTA 寄存器的 SPEN 位置 1，使能 EUSART。如果 RX/DT 或 TX/CK 引脚与模拟外设共用，必须清零相应的 ANSEL 位禁止模拟 I/O 功能。

12.4.1.1 主控时钟

同步数据传输使用独立的时钟线同步传输数据。配置为主控制器件的器件在 TX/CK 引脚发送时钟信号。当 EUSART 被配置为同步发送操作时，TX/CK 引脚自动配置为输出引脚。串行数据位在每个时钟的上升沿发生改变，以确保它们在下降沿有效。每个数据位的时间为一个时钟周期，有多少数据位就只能产生多少个时钟周期。

12.4.1.2 时钟极性

器件提供时钟极性选项以与 Microwire 兼容。由 BAUDCTL 寄存器的 SCKP 位选择时钟极性。将 SCKP 位置 1 将时钟空闲状态设置为高电平。当 SCKP 位置 1 时，数据在每个时钟的下降沿发生改变。清零 SCKP 位，将时钟空闲状态设置为低电平。当清零 SCKP 位时，数据在每个时钟的上升沿发生改变。

12.4.1.3 同步主控发送

由器件的 RX/DT 引脚输出数据。当 EUSART 配置为同步主控发送操作时，器件的 RX/DT 和 TX/CK 输出引脚自动使能。

向 TXREG 寄存器写入一个字符开始发送。如果 TSR 中仍保存全部或部分前一字符，新的字符数据保存在 TXREG 中，直到发送完前一字符的停止位为止。如果这是第一个字符，或者前一个字符已经完全从 TSR 中移出，则 TXREG 中的数据会被立即传输到 TSR 寄存器。当字符从 TXREG 传输到 TSR 后会立即开始发送数据。每个数据位在主控时钟的上升沿发生改变，并保持有效，直至下一个时钟的上升沿为止。

注： 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。

12.4.1.4 同步主控发送设置

1. 初始化 SPBRGH 和 SPBRG 这对寄存器以及 BRGH 和 BRG16 位，以获得所需的波特率（见第 12.3 节“EUSART 波特率发生器 (BRG)”）。
2. 将 SYNC、SPEN 和 CSRC 位置 1，使能同步主控串行端口。
3. 将 SREN 和 CREN 位清零，禁止接收模式。
4. 将 TXEN 位置 1 使能发送模式。
5. 如果需要发送 9 位字符，将 TX9 置 1。
6. 若需要中断，将 TXIE、GIE 和 PEIE 中断允许位置 1。
7. 如果选择发送 9 位字符，应该将第 9 位数据装入 TX9D 位。
8. 通过将数据装入 TXREG 寄存器启动发送。

PIC16F882/883/884/886/887

图 12-10: 同步发送

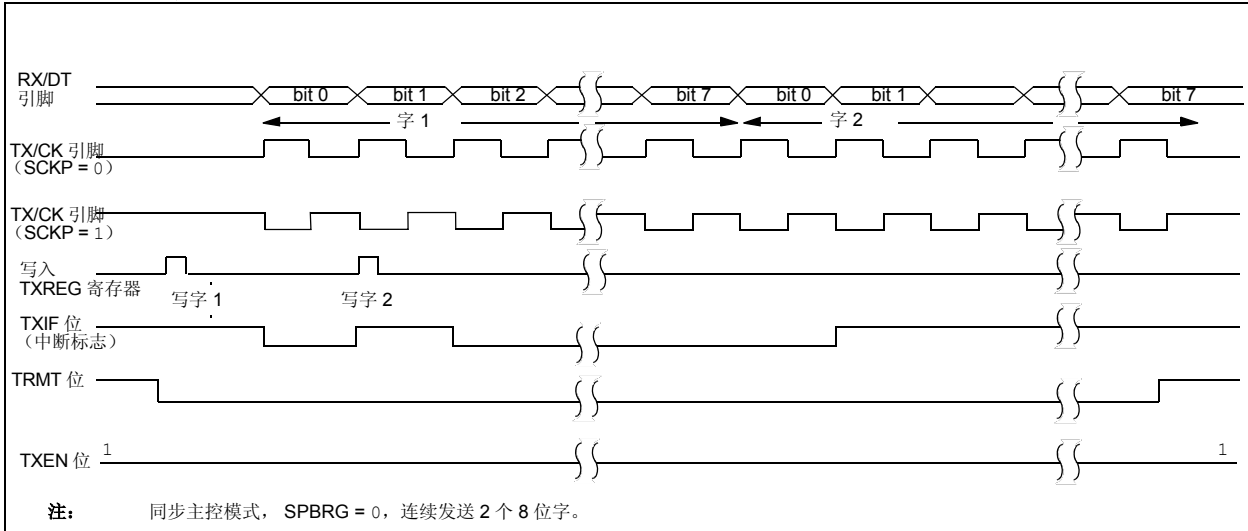


图 12-11: 同步发送 (通过 TXEN)

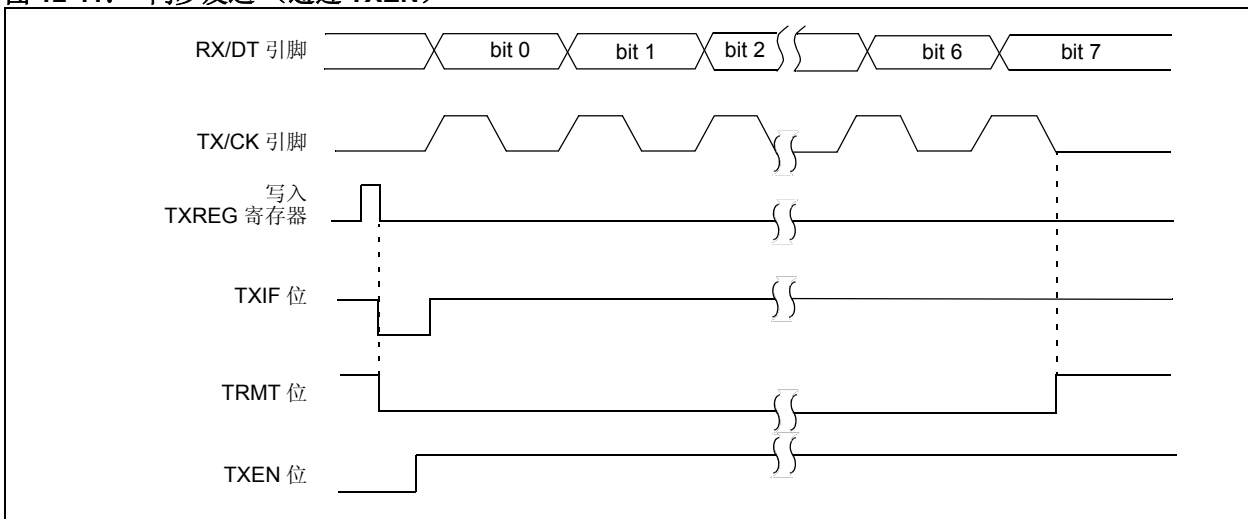


表 12-7: 与同步主控发送有关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
RCREG	EUSART 接收数据寄存器								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
TXREG	EUSART 发送数据寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

图注: x = 未知, - = 未实现 (读为 0)。同步主控发送不使用阴影单元。

PIC16F882/883/884/886/887

12.4.1.5 同步主控接收

在 RX/DT 引脚接收数据。当 EUSART 配置为同步主控接收时，自动禁止器件的 RX/DT 和 TX/CK 引脚的输出驱动器。

在同步模式下，将单字接收使能位（RCSTA 寄存器的 SREN 位）或连续接收使能位（RCSTA 寄存器的 CREN 位）置 1 使能接收。

当将 SREN 置 1，CREN 位清零时，一个单字符中有多少数据位就只能产生多少时钟周期。一个字符传输结束后，自动清零 SREN 位。当 CREN 置 1 时，将产生连续时钟，直到清零 CREN 为止。如果 CREN 在一个字符的传输过程中清零，则 CK 时钟立即停止，并丢弃该不完整的字符。如果 SREN 和 CREN 都置 1，则当第一个字符传输完成时，SREN 位被清零，CREN 优先。

将 SREN 或 CREN 位置 1，启动接收。在 TX/CK 时钟引脚信号的下降沿采样 RX/DT 引脚上的数据，并将采样到的数据移入接收移位寄存器（RSR）。当 RSR 接收到一个完整字符时，将 RCIF 位置 1，字符自动移入 2 字节接收 FIFO。接收 FIFO 中最顶端字符的低 8 位可通过 RCREG 读取。只要接收 FIFO 中仍有未读字符，则 RCIF 位就保持置 1 状态。

12.4.1.6 接收溢出错误

接收 FIFO 缓冲器可以保存 2 个字符。在读 RCREG 以访问 FIFO 之前，若完整地接收到第 3 个字符，则产生溢出错误。此时，RCSTA 寄存器的 OERR 位会置 1。FIFO 中先前的数据不会被改写。可以读取 FIFO 缓冲器内的 2 个字符，但是在错误被清除前，不能再接收其他字符。只能通过清除溢出条件，将 OERR 位清零。如果发生溢出时，SREN 位为置 1 状态，CREN 位为清零状态，则通过读 RCREG 寄存器清除错误。如果溢出时，CREN 为置 1 状态，则可以清零 RCSTA 寄存器的 CREN 位或清零 SPEN 位以复位 EUSART，从而清除错误。

12.4.1.7 接收 9 位字符

EUSART 支持接收 9 位字符。当 RCSTA 寄存器的 RX9 位置 1 时，EUSART 将接收到的每个字符的 9 位数据移入 RSR。RCSTA 寄存器的 RX9D 位是接收 FIFO 顶端未读字符的第九位数据位（最高数据位）。当从接收 FIFO 缓冲器读取 9 位数据时，必须在读 RCREG 的 8 个低位之前，读取 RX9D 数据位。

12.4.1.8 同步主控接收设置

1. 用正确的波特率初始化一对 SPBRGH:SPBRG 寄存器。按需要将 BRGH 和 BRG16 位置 1 或清零，以获得所需的波特率。
2. 将 SYNC、SPEN 和 CSRC 位置 1 使能同步主控串行端口。
3. 确保将 CREN 和 SREN 位清零。
4. 如果使用中断，将 INTCON 寄存器的 GIE 和 PEIE 位置 1，将 RCIE 置 1。
5. 如果需要接收 9 位字符，将 RX9 位置 1。
6. 将 SREN 位置 1，启动接收，或将 CREN 位置 1 使能连续接收。
7. 当字符接收完毕后，将 RCIF 中断标志位置 1。如果允许位 RCIE 置 1，还会产生一个中断。
8. 读 RCSTA 寄存器以获取第 9 个数据位（使能 9 位接收时），并判断接收过程中是否产生错误。
9. 读 RCREG 寄存器获取接收到的 8 位数据。
10. 如果产生溢出错误，清零 RCSTA 寄存器的 CREN 位或清零 SPEN 以复位 EUSART 来清除错误。

PIC16F882/883/884/886/887

图 12-12: 同步接收 (主控模式, SREN)

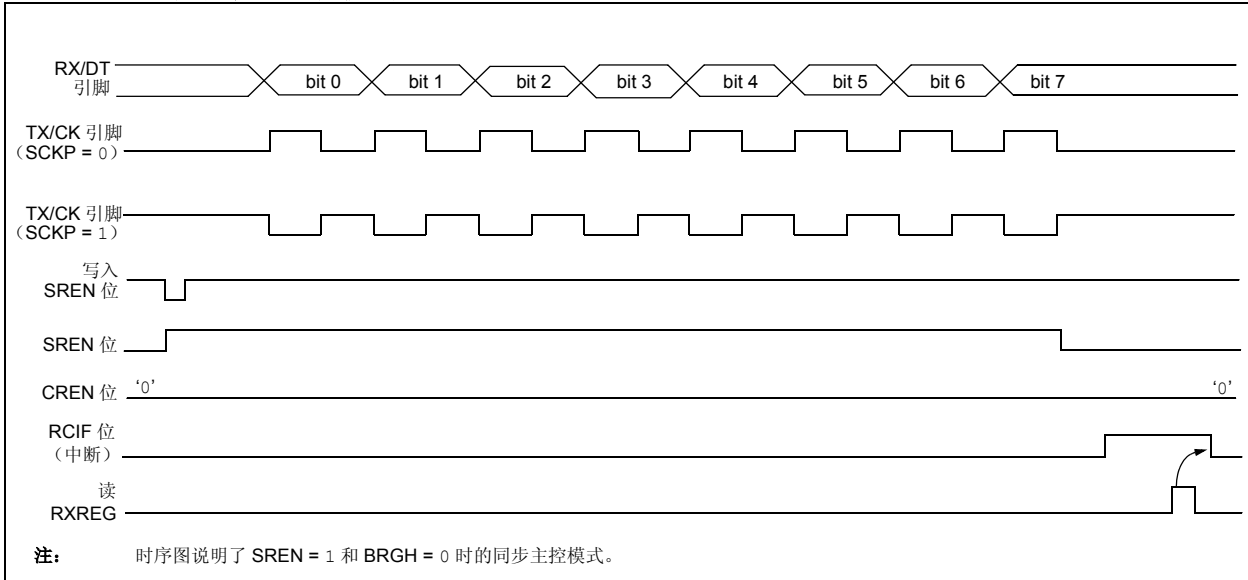


表 12-8: 与同步主控接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
RCREG	EUSART 接收数据寄存器								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
TXREG	EUSART 发送数据寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

图注: x = 未知, - = 未实现, 读为 0。同步主控接收不使用阴影单元。

PIC16F882/883/884/886/887

12.4.2 同步从动模式

下列位用来将 EUSART 配置为同步从动操作：

- SYNC = 1
- CSRC = 0
- SREN = 0（用于发送）； SREN = 1（用于接收）
- CREN = 0（用于发送）； CREN = 1（用于接收）
- SPEN = 1

将 TXSTA 寄存器的 SYNC 位置 1，可将器件配置用于同步操作。将 TXSTA 寄存器的 CSRC 位置 1，将器件配置为从动器件。将 RCSTA 寄存器的 SREN 和 CREN 位清零，以确保器件处于发送模式，否则器件将被配置为接收模式。将 RCSTA 寄存器的 SPEN 位置 1，使能 EUSART。如果 RX/DT 或 TX/CK 引脚与模拟外设共用，必须清零相应的 ANSEL 位禁止模拟 I/O 功能。

12.4.2.1 EUSART 同步从动发送

除休眠模式外，同步主控和从动模式的工作原理是相同的（见第 12.4.1.3 节“同步主控发送”）。

如果向 TXREG 写入 2 个字，然后执行 SLEEP 指令，则会出现下列情况：

1. 第一个字立即传输到 TSR 寄存器并进行发送。
2. 第二个字留在 TXREG 寄存器中。
3. TXIF 中断标志位不会置 1。
4. 当第一个字符移出 TSR 时，TXREG 寄存器将把第二个字符传输到 TSR，然后标志位 TXIF 置 1。
5. 如果 PEIE 和 TXIE 位都置 1，则由中断将器件从休眠模式唤醒，然后执行下一条指令。如果 GIE 位也置 1，程序将调用中断服务程序。

12.4.2.2 同步从动发送设置：

1. 将 SYNC 和 SPEN 位置 1 并将 CSRC 位清零。
2. 将 CREN 和 SREN 位清零。
3. 如果使用中断，将 INTCON 寄存器的 GIE 和 PEIE 位置 1，并将 TXIE 位置 1。
4. 如果需要发送 9 位数据，将 TX9 位置 1。
5. 将 TXEN 位置 1 使能发送。
6. 若选择发送 9 位数据，将最高位写入 TX9D 位。
7. 将低 8 位数据写入 TXREG 寄存器开始传输。

表 12-9: 与同步从动发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
RCREG	EUSART 接收数据寄存器								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
TXREG	EUSART 发送数据寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

图注： x = 未知， - = 未实现（读为 0）。同步从动发送不使用阴影单元。

PIC16F882/883/884/886/887

12.4.2.3 EUSART 同步从动接收

除了以下不同外，同步主控和从动模式的工作原理相同。（见第 12.4.1.5 节“同步主控接收”）。

- 休眠模式
- CREN 位总是置 1，因此接收器不能进入空闲状态。
- SREN 位，在从动模式可为“任意值”。

如果在进入休眠模式之前，已经将 CREN 位置 1，则在休眠模式仍可接收字符。RSR 寄存器接收到字后，就会立即将接收到的数据传输到 RCREG 寄存器。如果将 RCIE 允许位置 1，则产生的中断将使器件从休眠模式唤醒，然后执行下一条指令。如果 GIE 位也置 1，则程序将跳转到中断向量处执行。

12.4.2.4 同步从动接收设置：

1. 将 SYNC 和 SPEN 位置 1 并将 CSRC 位清零。
2. 若要使用中断，确保将 INTCON 寄存器的 GIE 和 PEIE 位置 1，并将 RCIE 位置 1。
3. 如果需要接收 9 位字符，将 RX9 位置 1。
4. 将 CREN 位置 1，使能接收。
5. 当接收完成后，将 RCIF 位置 1。如果 RCIE 已置 1，还会产生一个中断。
6. 如果使能 9 位模式，从 RCSTA 寄存器的 RX9D 位获取最高位。
7. 读 RCREG 寄存器，从接收 FIFO 缓冲器获取接收到的 8 个低数据位。
8. 如果产生溢出错误，清零 RCSTA 寄存器的 CREN 位或清零 SPEN 位以复位 EUSART 来清除错误。

表 12-10: 与同步从动接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
RCREG	EUSART 接收数据寄存器								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
TXREG	EUSART 发送数据寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

图注： x = 未知， - = 未实现（读为 0）。同步从动接收不使用阴影单元。

13.0 主控同步串行端口（MSSP）模块

13.1 主控 SSP（MSSP）模块概述

主控同步串行端口（Master Synchronous Serial Port, MSSP）模块是用于同其他外设或单片机进行通信的串行接口。这些外设器件可以是串行 EEPROM、移位寄存器、显示驱动器或 A/D 转换器等。MSSP 模块有下列两种工作模式：

- 串行外设接口（SPI）
- I²C™
 - 全主控模式
 - 从动模式（支持广播地址呼叫）

I²C 接口在硬件上支持下列模式：

- 主控模式
- 多主机模式
- 从动模式

13.2 控制寄存器

MSSP 模块具有三个相关的寄存器。包括一个 STATUS 寄存器和两个控制寄存器。

寄存器 13-1 给出了 MSSP 的状态寄存器（SSPSTAT），寄存器 13-2 给出了 MSSP 控制寄存器 1（SSPCON1），寄存器 13-3 给出了 MSSP 控制寄存器 2（SSPCON2）。

PIC16F882/883/884/886/887

寄存器 13-1: SSPSTAT: SSP 状态寄存器

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P	S	R/W	UA	BF
bit 7							bit 0

图注:

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

- bit 7 **SMP:** 采样位
SPI 主控模式:
 1 = 在数据输出时间的末尾采样输入数据
 0 = 在数据输出时间的中间采样输入数据
SPI 从动模式:
 当使用 SPI 的从动模式时, 必须将 SMP 清零。
在 I²C 主控或从动模式下:
 1 = 禁止变化率控制, 标准速度模式 (100 kHz 和 1 MHz)
 0 = 使能变化率控制, 高速模式 (400 kHz)
- bit 6 **CKE:** SPI 时钟边沿选择位
CKP = 0:
 1 = 在 SCK 引脚的上升沿发送数据
 0 = 在 SCK 引脚的下降沿发送数据
CKP = 1:
 1 = 在 SCK 引脚的下降沿发送数据
 0 = 在 SCK 引脚的上升沿发送数据
- bit 5 **D/A:** 数据 / 地址位 (仅限 I²C 模式)
 1 = 表示最后接收或发送的字节是数据
 0 = 表示最后接收或发送的字节是地址
- bit 4 **P:** 停止位
 (仅限 I²C 模式。禁止 MSSP 模块 (SSPEN 清零) 时此位被清零。)
 1 = 表示最后检测到了停止位 (复位时该位为 0)。
 0 = 表示最后未检测到停止位
- bit 3 **S:** 起始位
 (仅限 I²C 模式。禁止 MSSP 模块 (SSPEN 清零) 时此位被清零。)
 1 = 表示最后检测到了起始位 (复位时该位为 0)。
 0 = 最后未检测到起始位
- bit 2 **R/W:** 读 / 写位信息 (仅限 I²C 模式)
 该位用来保存在最后一次地址匹配后的 R/W 位信息。该位仅在从地址匹配开始到下一个起始位、停止位或非 ACK 位时有效。
在 I²C 从动模式下:
 1 = 读
 0 = 写
在 I²C 主控模式下:
 1 = 正在发送
 0 = 不在进行发送
 此位与 SEN、RSEN、PEN、RCEN 或 ACKEN 做逻辑或运算的结果将指示 MSSP 是否在空闲模式下。
- bit 1 **UA:** 更新地址位 (仅限 10 位 I²C 模式)
 1 = 表示用户需要更新 SSPADD 寄存器中的地址
 0 = 不需要更新地址
- bit 0 **BF:** 缓冲器满状态位
接收 (SPI 和 I²C 模式):
 1 = 接收完成, SSPBUF 满
 0 = 接收未完成, SSPBUF 空
发送 (仅限 I²C 模式):
 1 = 数据正在发送 (不包括 ACK 和停止位), SSPBUF 满
 0 = 数据发送完成 (不包括 ACK 和停止位), SSPBUF 空

PIC16F882/883/884/886/887

寄存器 13-3: **SSPCON2: SSP 控制寄存器 2**

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

图注:

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

- bit 7 **GCEN:** 广播呼叫使能位 (仅限 I²C 从动模式)
 1 = 允许在 SSPSR 中接收到广播呼叫地址 (0000h) 时产生中断
 0 = 禁止广播呼叫地址
- bit 6 **ACKSTAT:** 应答状态位 (仅限于 I²C 主控模式)
在主控发送模式下:
 1 = 未接收到来自从动器件的应答
 0 = 已接收到来自从动器件的应答
- bit 5 **ACKDT:** 应答数据位 (仅限于 I²C 主控模式)
在主控接收模式下:
 用户在接收完成后发送的应答序列的值
 1 = 不应答
 0 = 应答
- bit 4 **ACKEN:** 应答序列使能位 (仅限 I²C 主控模式)
在主控接收模式下:
 1 = 在 SDA 和 SCL 引脚启动应答序列, 发送 ACKDT 数据位。由硬件自动清零。
 0 = 应答序列空闲
- bit 3 **RCEN:** 接收使能位 (仅限 I²C 主控模式)
 1 = 使能 I²C 接收模式
 0 = 接收空闲
- bit 2 **PEN:** 停止条件使能位 (仅限于 I²C 主控模式)
SCK 释放控制:
 1 = 在 SDA 和 SCL 引脚启动停止条件。由硬件自动清零。
 0 = 停止条件空闲
- bit 1 **RCEN:** 重复启动条件使能位 (仅限 I²C 主控模式)
 1 = 在 SDA 和 SCL 引脚启动重复启动条件。由硬件自动清零。
 0 = 重复启动条件空闲
- bit 0 **SEN:** 启动条件使能位 (仅限 I²C 主控模式)
 1 = 在 SDA 和 SCL 引脚启动启动条件。由硬件自动清零。
 0 = 启动条件空闲

注 1: 对于 ACKEN、RCEN、PEN、RSEN 和 SEN 位: 如果 I²C 模块不处在空闲模式, 此位可能无法被置 1 (没有假脱机 (spooling)) 且可能无法对 SSPBUF 进行写操作 (禁止写 SSPBUF)。

PIC16F882/883/884/886/887

13.3 SPI 模式

SPI 模式允许同时同步发送和接收 8 位数据。支持 SPI 的所有 4 种模式。通常使用以下三个引脚来完成通信：

- 串行数据输出 (SDO) ——RC5/SDO
- 串行数据输入 (SDI) ——RC4/SDI/SDA
- 串行时钟 (SCK) ——RC3/SCK/SCL

此外，在任意一种从动工作模式下均可使用第 4 个引脚：

- 从动选择 (\overline{SS}) ——RA5/ \overline{SS} /AN4

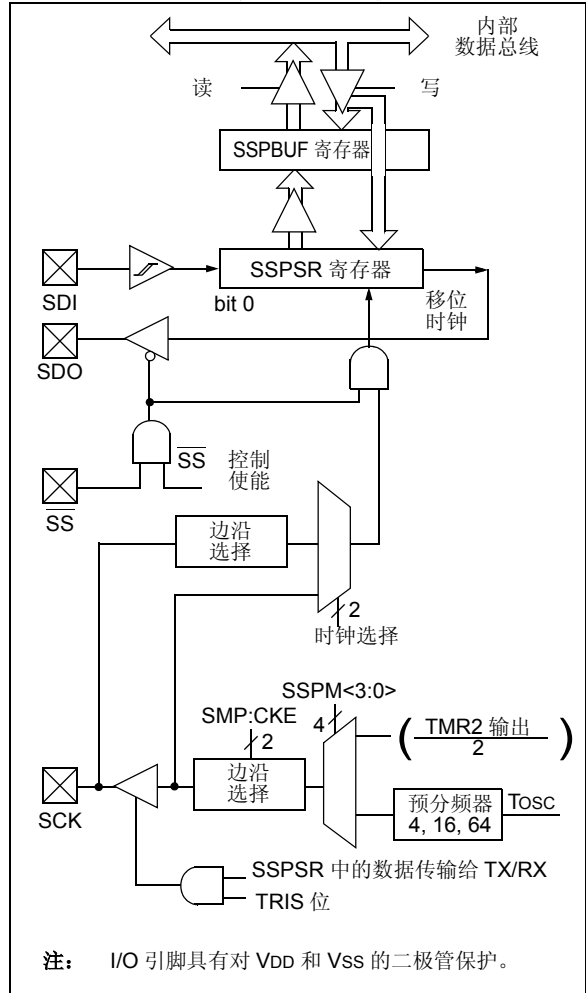
13.3.1 工作原理

当初初始化 SPI 时，需要指定几个选项。可以通过对相应的控制位 (SSPCON<5:0> 和 SSPSTAT<7:6>) 编程来指定。这些控制位用于指定以下选项：

- 主控模式 (SCK 作为时钟输出)
- 从动模式 (SCK 作为时钟输入)
- 时钟极性 (SCK 的空闲状态)
- 输入数据的采样相位 (数据输出时间的中间或末尾)
- 时钟边沿 (在 SCK 的上升沿 / 下降沿输出数据)
- 时钟速率 (仅限主控模式)
- 从动选择模式 (仅限于从动模式)

图 13-1 显示了在 SPI 模式下 MSSP 模块的框图。

图 13-1: MSSP 框图 (SPI 模式)



MSSP 模块由一个发送 / 接收移位寄存器 (SSPSR) 和一个缓冲寄存器 (SSPBUF) 组成。SSPSR 将数据移入和移出器件，最高有效位在前。SSPBUF 保存上次写入 SSPSR 的数据直到新接收到的数据就绪为止。一旦 8 位数据接收完毕，该字节就被移入 SSPBUF 寄存器。然后，SSPSTAT 寄存器的缓冲器满检测位 BF 和 PIR1 寄存器的中断标志位 SSPIF 被置 1。这种双重缓冲数据接收方式 (SSPBUF) 允许在读取刚接收的数据之前，就开始接收下一个字节。在数据发送 / 接收期间，任何试图写 SSPBUF 寄存器的操作都会被忽略，并将 SSPCON 寄存器的写冲突检测位 WCOL 置 1。此时用户必须用软件将 WCOL 位清零，否则无法判别下一次对 SSPBUF 的写操作是否成功完成。

PIC16F882/883/884/886/887

当应用软件等待接收有效数据时，应在下一个要传输的数据字节写入 SSPBUF 之前，将 SSPBUF 中的前一个数据读出。缓冲器满标志位 BF（SSPSTAT 寄存器）用于表示何时 SSPBUF 已经载入了接收到的数据（发送完成）。当 SSPBUF 中的数据被读取后，BF 位即被清零。如果 SPI 仅作为发送器，则不必理会接收的数据。通常可用 MSSP 中断来判断发送或接收何时完成。必须读和 / 或写 SSPBUF。如果不使用中断来处理数据的收发，用软件查询方法同样可确保不会发生写冲突。例 13-1 显示了装载 SSPBUF（SSPSR）进行数据发送的过程。

不能直接读写 SSPSR 寄存器，只能通过寻址 SSPBUF 寄存器间接访问它。此外，MSSP 状态寄存器 SSPSTAT 将指示各种状态。

13.3.2 使能 SPI I/O

要使能串行端口，SSPCON 寄存器的 MSSP 使能位 SSPEN 必须置 1。要复位或重新配置 SPI 模式，要先将 SSPEN 位清零，重新初始化 SSPCON 寄存器，然后将 SSPEN 位置 1。这将把 SDI、SDO、SCK 和 SS 引脚配置为串行端口引脚。要将这些引脚用作串行端口，还必须将其数据方向位（在 TRIS 寄存器中）正确编程，方法如下：

- SDI 由 SPI 模块自动控制
- 必须将 SDO 的 TRISC<5> 清零
- 必须将 SCK（主控模式）的 TRISC<3> 位清零
- 必须将 SCK（从动模式）的 TRISC<3> 位清零
- 必须将 \overline{SS} 的 TRISA<5> 置 1

对于任何不想要的串行端口功能，可通过将对应的数据方向（TRIS）寄存器设置为相反值来跳过。

例 13-1: 装载 SSPBUF（SSPSR）寄存器

LOOP	BTFS	SSPSTAT, BF	;Has data been received (transmit complete)?
	GOTO	LOOP	;No
	MOVF	SSPBUF, W	;WREG reg = contents of SSPBUF
	MOVWF	RXDATA	;Save in user RAM, if data is meaningful
	MOVF	TXDATA, W	;W reg = contents of TXDATA
	MOVWF	SSPBUF	;New data to xmit

PIC16F882/883/884/886/887

13.3.3 主控模式

主器件控制 SCK，因此可以随时启动数据传输。主器件根据软件协议确定从器件应在何时广播数据。

在主控模式下，数据一旦写入 SSPBUF 寄存器就开始发送或接收。如果 SPI 仅作为接收器，则可以禁止 SDO 输出（将其编程设定为输入）。SSPSR 寄存器按设置的时钟速率对 SDI 引脚上的信号进行连续移位输入。每个字节接收完后，都会被当作普通的接收字节装入 SSPBUF 寄存器（相应的中断和状态位置 1）。这可以在接收器应用中作为“线路活动监控（Line Activity Monitor）”模式，是很有用的。

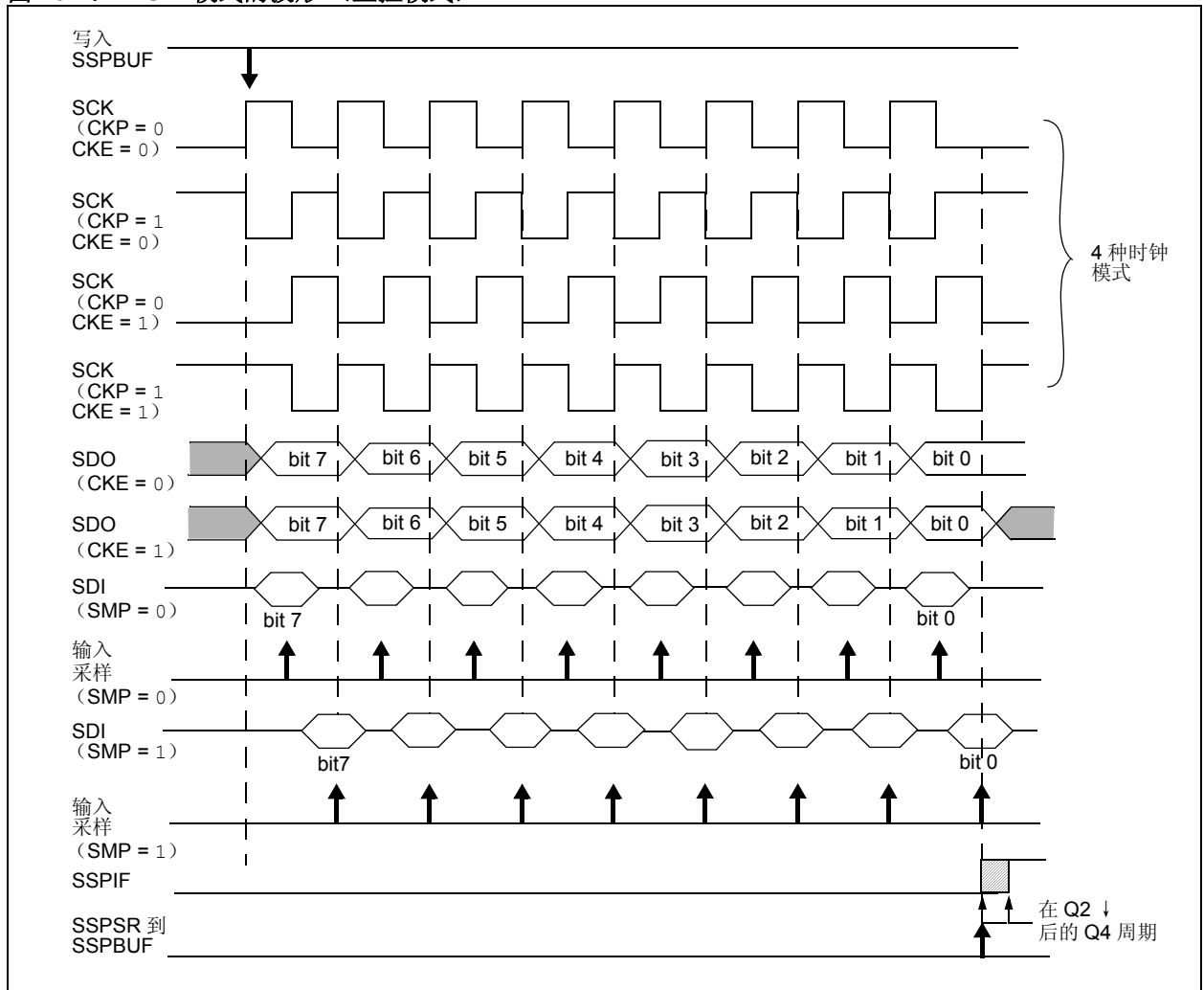
可通过对 SSPCON 寄存器的 CKP 位进行相应的编程来选择时钟极性。图 13-2、图 13-4 和图 13-5 给出了 SPI 通信的波形图，其中 MSb 被首先发送。在**主控模式**下，SPI 时钟速率（比特率）可由用户编程设定为下列速率之一：

- $F_{osc}/4$ （或 T_{cy} ）
- $F_{osc}/16$ （或 $4 \cdot T_{cy}$ ）
- $F_{osc}/64$ （或 $16 \cdot T_{cy}$ ）
- Timer2 输出 /2

这里允许的最大数据速率是 10.00 Mbps（当晶振为 40 MHz 时）。

图 13-2 给出了主控模式的波形图。当 SSPSTAT 寄存器的 CKE 位置 1 时，SDO 数据在 SCK 上出现时钟边沿前就有效。图中所示输入采样的变化由 SSPSTAT 寄存器的 SMP 位的状态决定。图中指出了将接收到的数据装入 SSPBUF 的时间。

图 13-2: SPI 模式的波形（主控模式）



PIC16F882/883/884/886/887

13.3.4 从动模式

在从动模式下，当 SCK 引脚上出现外部时钟脉冲时，发送和接收数据。当最后一位数据被锁存后，PIR1 寄存器的 SSPIF 中断标志位置 1。

在从动模式下，时钟由 SCK 引脚上的外部时钟源提供。外部时钟必须满足电气规范中规定的高电平和低电平的最短时间要求。

在休眠状态下，从器件仍可发送 / 接收数据。当收到一个字节时，器件从休眠状态被唤醒。

13.3.5 从动选择同步

SS 引脚允许器件工作在同步从动模式。SPI 必须工作在从动模式下，并使能 SS 引脚控制 ($SSPxCON1<3:0> = 04h$)。要使 SS 引脚用作输入引脚，不能将该引脚驱动为低电平。当 SS 引脚为低电平时，使能数据的发送和接收，

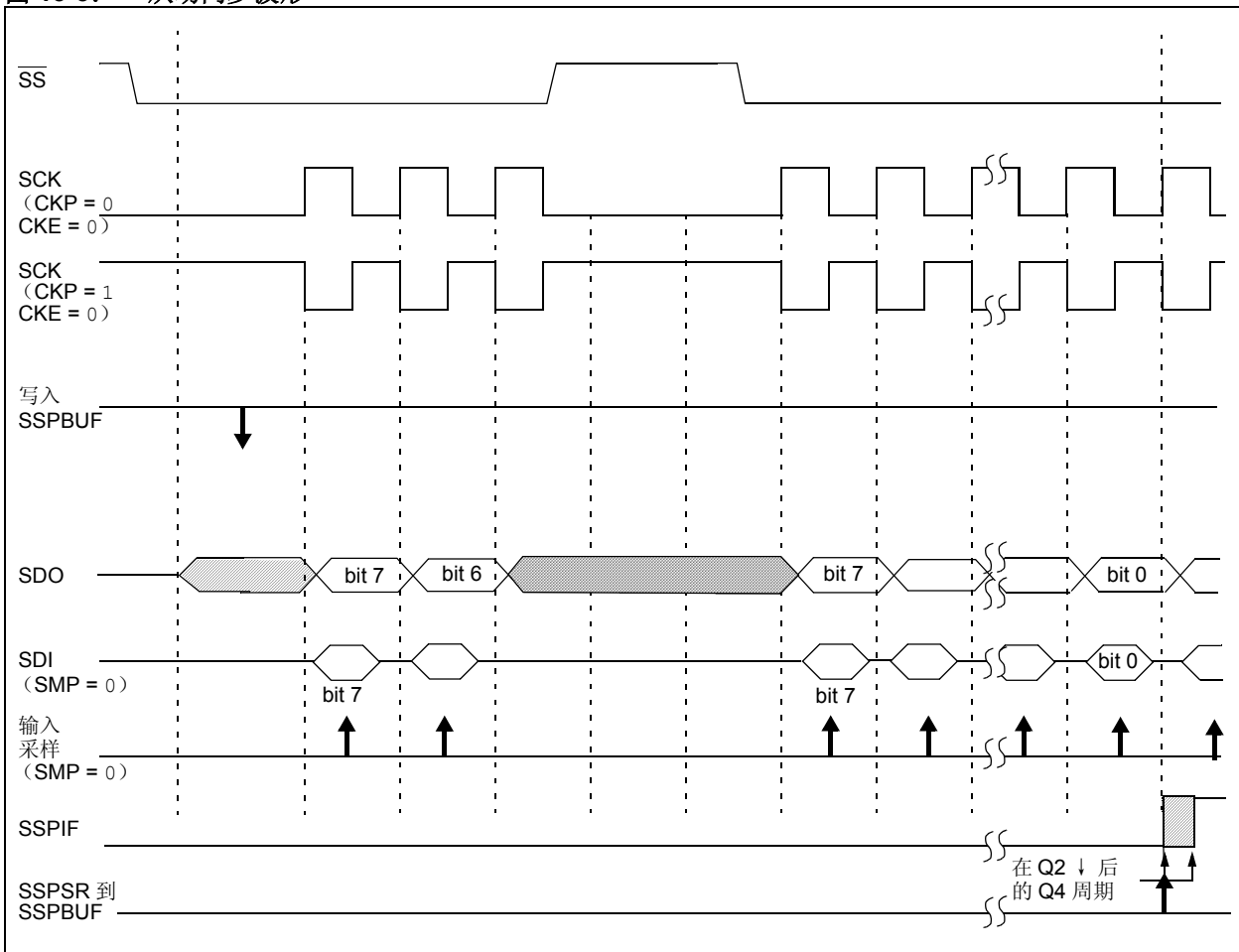
同时 SDO 引脚被驱动。当 SS 引脚为高电平时，即使是在数据的发送过程中，SDO 引脚也不再被驱动，而是变成悬空输出。根据应用的需要，可外接上拉 / 下拉电阻。

- 注 1:** 当 SPI 工作在从动模式下，并且 SS 引脚控制使能 ($SSPxCON1<3:0> = 0100$) 时，如果 SS 引脚置为 VDD 电平，SPI 模块将被复位。
- 2:** 如果在 CKE 置 1 (SSPSTAT 寄存器) 的从动模式下使用 SPI，则必须使能 SS 引脚控制。

当 SPI 模块复位后，位计数器被强制归 0。这可以通过强制将 SS 引脚拉为高电平或将 SSPEN 位清零来实现。

将 SDO 引脚和 SDI 引脚相连可以仿真双线制通信。当 SPI 需要作为接收器工作时，SDO 引脚可以被配置为输入。这样就禁止了从 SDO 发送数据。因为 SDI 不会引起总线冲突，因而总是可以将其保留为输入 (SDI 功能)。

图 13-3: 从动同步波形



PIC16F882/883/884/886/887

图 13-4: SPI 模式波形 (从动模式, CKE = 0)

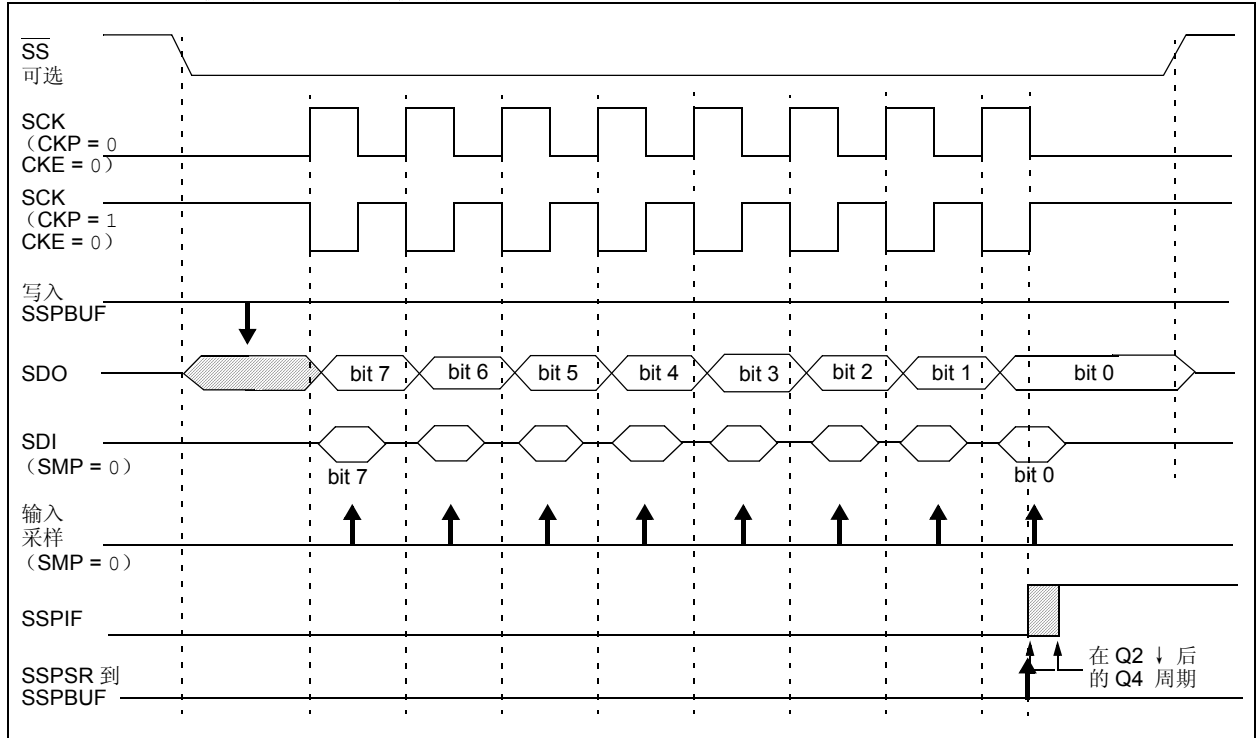
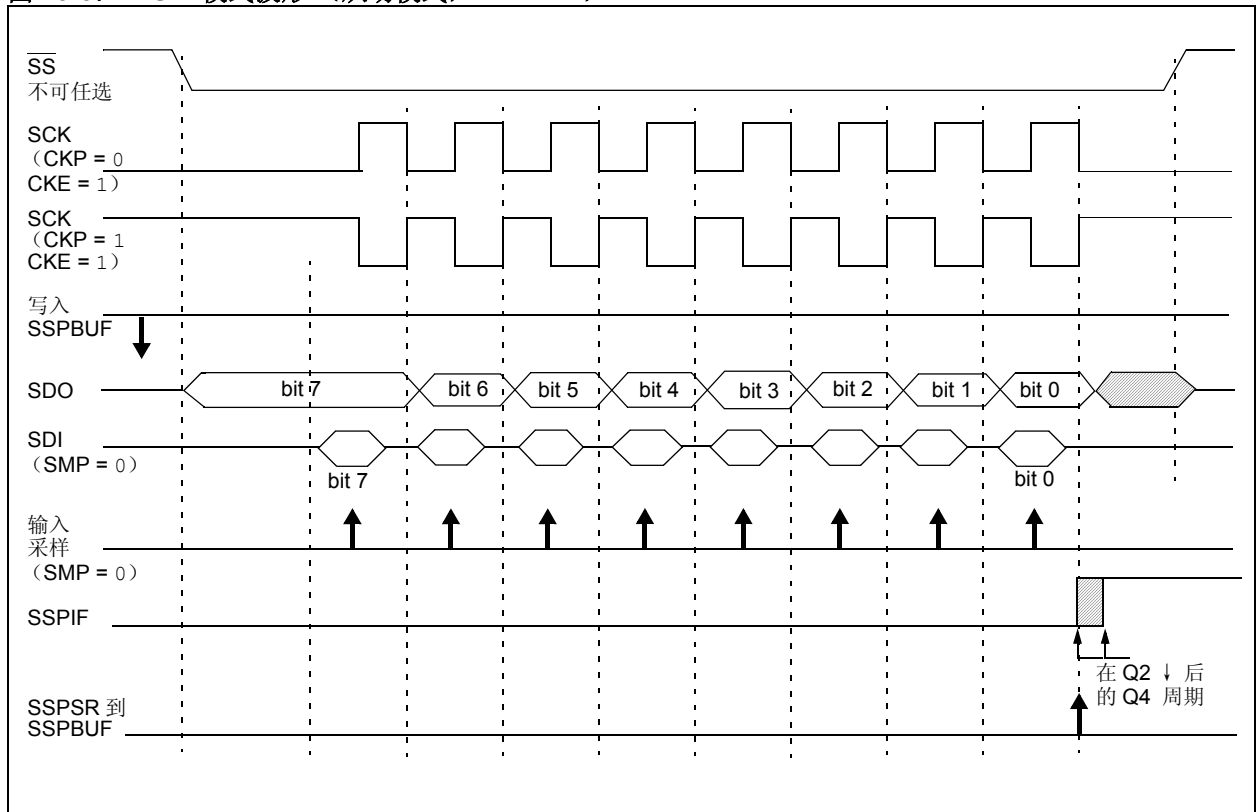


图 13-5: SPI 模式波形 (从动模式, CKE = 1)



PIC16F882/883/884/886/887

13.3.6 休眠操作

在主动模式下，如果选择了休眠模式，所有模块的时钟都将停止，并且在器件被唤醒前，发送 / 接收将保持此停滞状态。当器件返回到运行模式后，该模块将恢复发送和接收数据。

在从动模式下，SPI 发送 / 接收移位寄存器与器件异步工作。这可以使器件处于休眠模式下，而且数据仍可被移入 SPI 发送 / 接收移位寄存器。当 8 位数据全部接收到后，MSSP 中断标志位将置 1，并且如果允许中断的话，将唤醒器件。

13.3.7 复位的影响

复位会禁止 MSSP 模块并终止当前的传输。

13.3.8 总线模式兼容性

表 13-1 给出了标准 SPI 模式和 CKP 与 CKE 控制位状态的对应情况。

表 13-1: SPI 总线模式

标准 SPI 模式术语	控制位状态	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

还有一个 SMP 位用来控制何时采样数据。

表 13-2: 与 SPI 操作相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	0000 0000
SSPBUF	同步串行端口接收缓冲器 / 发送寄存器								xxxx xxxx	uuuu uuuu
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	0000 0000
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111

图注: x = 未知, u = 不变, - = 未实现, 读为 0。SPI 模式下的 MSSP 模块不使用阴影单元。

注 1: PORTA、LATA 和 TRISA 中的 bit 6 仅在 ECIO 和 RCIO 振荡模式下被使能。在所有其他振荡模式下，它们被禁止并读为 0。

PIC16F882/883/884/886/887

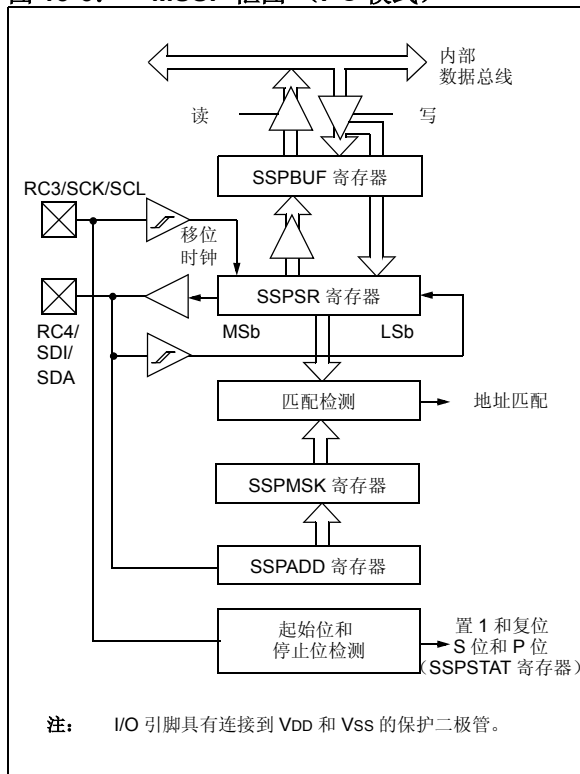
13.4 MSSP I²C™ 操作

MSSP 模块工作在 I²C 模式时，可以实现所有的主控和从动功能（包括广播呼叫支持），并且用硬件提供起始位和停止位的中断来判断总线何时空闲（多主机功能）。MSSP 模块实现了标准模式规范，以及 7 位和 10 位寻址。

有两个引脚用于数据传输。它们是时钟引脚（SCL）——RC3/SCK/SCL 引脚，和数据引脚（SDA）——RC4/SDI/SDA 引脚。用户必须通过 TRISC<4:3> 位将这些引脚配置为输入或输出引脚。

通过将 SSPCON 寄存器的 MSSP 使能位 SSPEN 置 1，使能 MSSP 模块的功能。

图 13-6: MSSP 框图 (I²C 模式)



MSSP 模块具有 7 个用于 I²C 操作的寄存器。它们是：

- MSSP 控制寄存器 1 (SSPCON)
- MSSP 控制寄存器 2 (SSPCON2)
- MSSP 状态寄存器 (SSPSTAT)
- 串行接收 / 发送缓冲寄存器 (SSPBUF)
- MSSP 移位寄存器 (SSPSR)：不能直接访问
- MSSP 地址寄存器 (SSPADD)
- MSSP 屏蔽寄存器 (SSPMASK)

可使用 SSPCON 寄存器控制 I²C 的操作。可使用 SSPM<3:0> 模式选择位 (SSPCON 寄存器) 选择以下 I²C 模式之一：

- I²C 主控模式，时钟 = OSC/4 (SSPADD + 1)
- I²C 从动模式 (7 位地址)
- I²C 从动模式 (10 位地址)
- I²C 从动模式，7 位地址，允许起始位和停止位中断
- I²C 从动模式，10 位地址，允许起始位和停止位中断
- I²C 固件控制的主控操作，从器件空闲

如果已将 SCL 和 SDA 引脚编程为输入引脚（将相应的 TRISC 位置 1），选择任何 I²C 模式且 SSPEN 位置 1 将强制 SCL 和 SDA 引脚为漏极开路。

13.4.1 从动模式

在从动模式下，SCL 引脚和 SDA 引脚必须被配置为输入 (TRISC<4:3> 置 1)。需要时（如从发送器）MSSP 模块将用输出数据改写输入状态。

当地址匹配时或在地址匹配后传输的数据被接收时，硬件会自动产生一个应答 (ACK) 脉冲，并把当时 SSPSR 寄存器中接收到的数据装入 SSPBUF 寄存器。

只要满足下列条件之一，MSSP 模块就不会产生此 ACK 脉冲：

- a) 缓冲器满标志位 BF (SSPCON 寄存器) 在接收到传输的数据前置 1。
- b) 在接收到传输的数据之前，溢出标志位 SSPOV (SSPCON 寄存器) 已被置 1。

在这种情况下，SSPSR 寄存器的值不会载入 SSPBUF，但是 PIR1 寄存器的 SSPIF 位会置 1。BF 位是通过读取 SSPBUF 寄存器清零的，而 SSPOV 位是通过软件清零的。

为确保正常工作，SCL 时钟输入必须满足最小高电平时间和最小低电平时间要求。关于 I²C 规范所规定的高电平和低电平时间以及对 MSSP 模块的具体要求，请参见时序参数 100 和 101。

PIC16F882/883/884/886/887

13.4.1.1 寻址

一旦使能了 MSSP 模块，它就会等待启动条件产生。在启动条件出现后，8 位数据被移入 SSPSR 寄存器。在时钟（SCL）线的上升沿采样所有的输入位。寄存器 SSPSR<7:1> 的值会和 SSPADD 寄存器的值比较，该比较是在第 8 个时钟脉冲（SCL）的下降沿进行的。如果地址匹配，并且 BF 位和 SSPOV 位为零，会发生下列事件：

- SSPSR 寄存器的值被装入 SSPBUF 寄存器。
- 缓冲器满标志位 BF 置 1。
- 产生 $\overline{\text{ACK}}$ 脉冲。
- 在第 9 个 SCL 脉冲的下降沿，PIR1 寄存器的 MSSP 中断标志位 SSPIF 置 1（如果允许中断则产生中断）。

在 10 位地址模式下，从器件需要接收两个地址字节。第一个地址字节的高 5 位指定这是否为 10 位地址。R/W（SSPSTAT 寄存器）必须指定写操作，这样从器件才能接收到第二个地址字节。对于 10 位地址，第一个字节应该是 11110 A9 A8 0，其中 A9 和 A8 是该地址的两个最高有效位。

10 位地址的工作时序如下，其中 7-9 步是针对从动发送器而言的：

- 接收地址的第一（高）字节（PIR1 寄存器的 SSPIF 位和 SSPSTAT 寄存器的 BF 和 UA 位置 1）。
- 用地址的第二个（低）字节更新 SSPADD 寄存器（UA 位清零并释放 SCL 线）。
- 读 SSPBUF 寄存器（BF 位清零）并将标志位 SSPIF 清零。
- 接收地址的第二个（低）字节（SSPIF 位、BF 位和 UA 位置 1）。
- 用地址的第一（高）字节更新 SSPADD 寄存器。如果匹配，释放 SCL 线，并将 UA 位清零。
- 读 SSPBUF 寄存器（BF 位清零）并将标志位 SSPIF 清零。
- 接收重复启动条件。
- 接收地址的第一个（高）字节（SSPIF 位和 BF 位置 1）。
- 读 SSPBUF 寄存器（BF 位清零）并将标志位 SSPIF 清零。

13.4.1.2 接收

当地址字节的 $\overline{\text{R/W}}$ 位清零并发生地址匹配时，SSPSTAT 寄存器的 R/W 位清零。接收到的地址被装入 SSPBUF 寄存器。

当存在地址字节溢出条件时，则不会产生应答脉冲（ACK）。溢出条件是指 BF 位（SSPSTAT 寄存器）置 1，或者 SSPOV 位（SSPCON 寄存器）置 1。

每个数据传输字节都会产生一个 MSSP 中断。必须用软件将 PIR1 寄存器的中断标志位 SSPIF 清零。SSPSTAT 寄存器用于确定该字节的状态。

13.4.1.3 发送

当接收的地址字节的 $\overline{\text{R/W}}$ 位置 1 并发生地址匹配时，SSPSTAT 寄存器的 $\overline{\text{R/W}}$ 位置 1。接收到的地址被装入 SSPBUF 寄存器。ACK 脉冲在第 9 位上发送，同时 RC3/SCK/SCL 引脚保持低电平。传输的数据必须装入 SSPBUF 寄存器，同时也被装入 SSPSR 寄存器。随后应通过将 CKP 位（SSPCON 寄存器）置 1 使能 RC3/SCK/SCL 引脚。在发送另一个时钟脉冲前，主控器件必须监视 SCL 引脚。从动器件可以通过延长时钟，暂停与主控器件的数据传输。8 个数据位在 SCL 输入的下降沿移出。这可确保在 SCL 为高电平期间 SDA 信号是有效的（见图 13-8）。

每个数据传输字节都会产生一个 MSSP 中断。SSPIF 标志位必须由软件清零，SSPSTAT 寄存器用于确定字节的状态。SSPIF 位在第 9 个时钟脉冲的下降沿被置 1。

来自主接收器的 $\overline{\text{ACK}}$ 脉冲将在 SCL 输入第 9 个脉冲的上升沿锁存。如果 SDA 线为高电平（无 ACK），那么表示数据传输已完成。在这种情况下，如果从器件锁存了 ACK，将复位从动逻辑（复位 SSPSTAT 寄存器），同时从器件监视下一个起始位的出现。如果 SDA 线为低电平（ACK），则必须将接下去要发送的数据装入 SSPBUF 寄存器，这也将装载 SSPSR 寄存器。应将 CKP 置 1 使能 RC3/SCK/SCL。

PIC16F882/883/884/886/887

图 13-7: I²C™ 从动模式接收时序 (7 位地址)

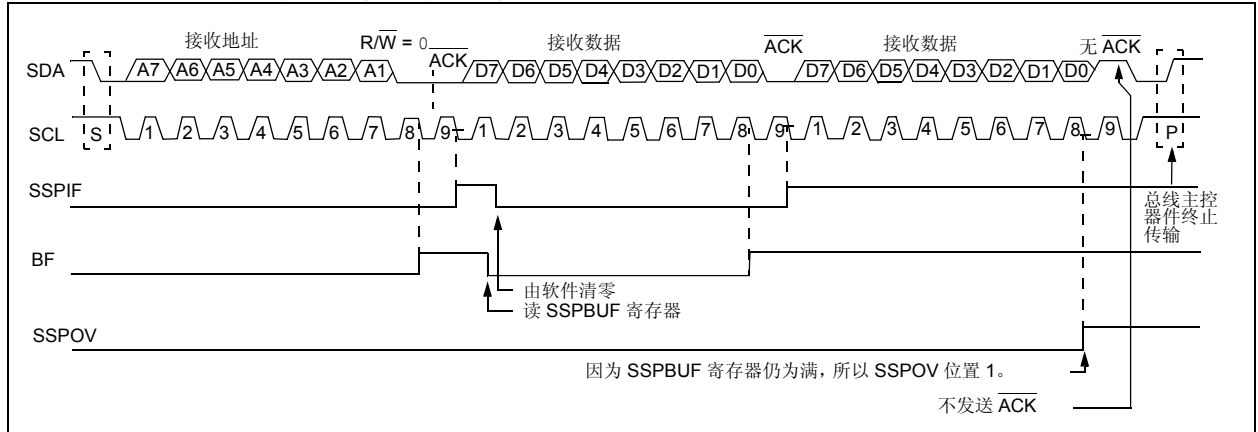
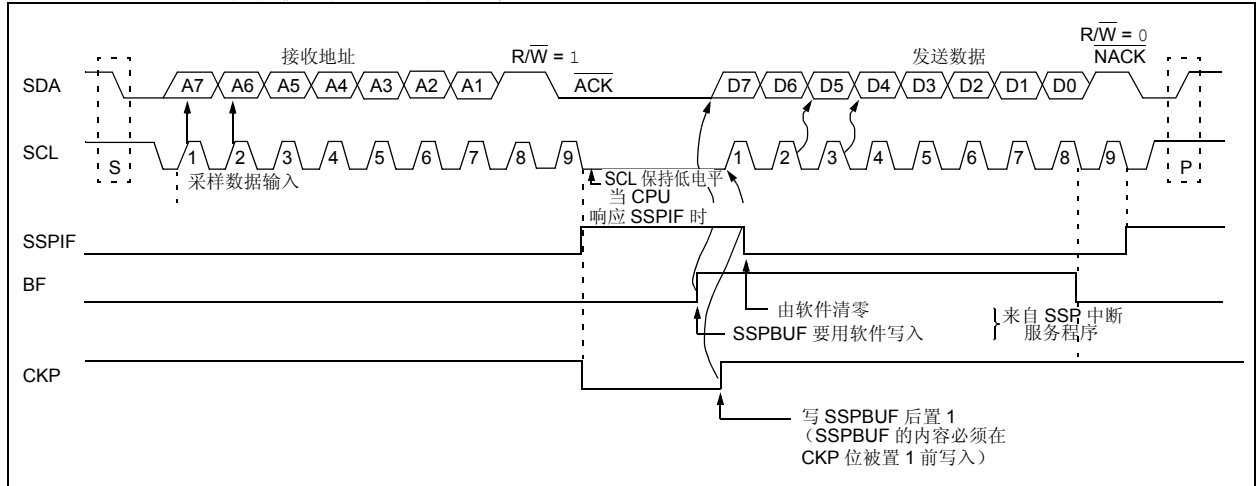


图 13-8: I²C™ 从动模式发送时序 (7 位地址)



PIC16F882/883/884/886/887

13.4.2 广播呼叫地址支持

在 I²C 总线的寻址过程中，通常由启动条件后的第一个字节决定主器件将寻址哪个从器件。但广播呼叫地址例外，它能寻址所有器件。当使用这个地址时，理论上所有的器件都应该发送一个应答响应。

广播呼叫地址是根据 I²C 协议为特定目的保留的八个地址之一。它由全 0 组成，且 R/W = 0。

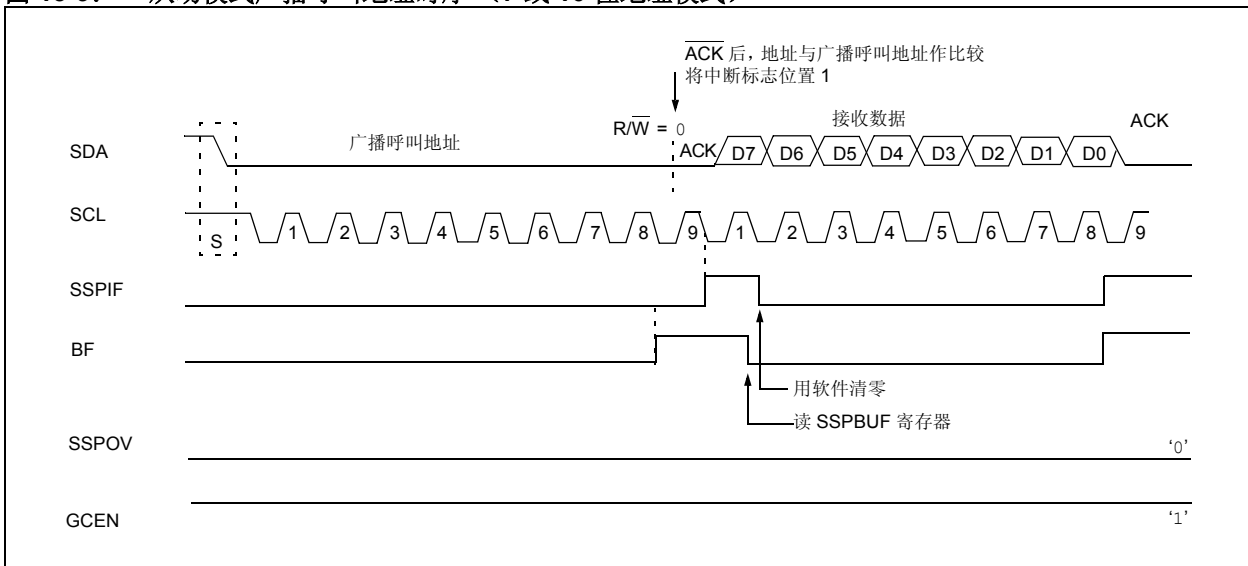
广播呼叫使能位 (GCEN) 使能 (SSPCON2 寄存器) 时，即可识别广播呼叫地址。检测到起始位后，8 位数据会移入 SSPSR，同时将该地址与 SSPADD 进行比较。它还会与广播呼叫地址进行比较并用硬件设定。

如果与广播呼叫地址匹配，SSPSR 的值将传输到 SSPBUF，BF 标志位 (第 8 位) 置 1，并且 SSPIF 中断标志位在第 9 位 (ACK 位) 的下降沿置 1。

当响应中断时，可以通过读取 SSPBUF 的内容来判断中断源。该值可以用于判断地址是特定器件的还是一个广播呼叫地址。

在 10 位模式下，需要更新 SSPADD 以使地址的后半部分匹配，同时 UA 位 (SSPSTAT 寄存器) 置 1。如果 GCEN 位置 1 时采样到广播呼叫地址，同时从器件被配置为 10 位地址模式，则不再需要地址的后半部分，也不会将 UA 位置 1，从器件将在应答后开始接收数据 (见图 13-9)。

图 13-9: 从动模式广播呼叫地址时序 (7 或 10 位地址模式)



PIC16F882/883/884/886/887

13.4.4.1 I²C™ 主控模式操作

所有串行时钟脉冲和启动 / 停止条件均由主器件产生。停止条件或重复启动条件能结束传输。因为重复启动条件也是下一次串行传输的开始，因此不会释放I²C总线。

在主控发送器模式下，串行数据通过 SDA 输出，而串行时钟由 SCL 输出。发送的第一个字节包括接收器件的地址（7 位）和读 / 写（R/W）位。在这种情况下，R/W 位将是逻辑 0。串行数据每次发送 8 位。每发送一个字节，会收到一个应答位。启动和停止条件的输出表明串行传输的开始和结束。

在主控接收模式下，发送的第一个字节包括发送器件的地址（7 位）和 R/W 位。在这种情况下，R/W 位将是逻辑 1。因此，发送的第一个字节是一个 7 位从器件地址，后面跟 1 表示接收。串行数据通过 SDA 接收，而串行时钟由 SCL 输出。每次接收 8 位串行数据。每接收到一个字节，都会发送一个应答位。启动和停止条件分别表明发送的开始和结束。

在 I²C 模式下，在 SPI 模式中使用的波特率发生器被用于将 SCL 时钟频率设置为 100 kHz、400 kHz 或 1 MHz。波特率发生器的重载值位于 SSPADD 寄存器的低 7 位。当发生对 SSPBUF 的写操作时，波特率发生器将自动开始计数。如果指定操作完成（即，发送的最后一个数据位后面跟着 ACK），内部时钟将自动停止计数，SCL 引脚将保持在其最后的状态。

下面是一个典型的发送事件序列：

- a) 用户通过将启动使能位 SEN（SSPCON2 寄存器）置 1 产生启动条件。
- b) SSPIF 位置 1。在进行任何其他操作前，MSSP 模块将等待所需的启动时间。
- c) 用户将从器件地址装入 SSPBUF 进行发送。
- d) 地址从 SDA 引脚移出，直到发送完所有 8 位为止。
- e) MSSP 模块移入来自器件的 ACK 位，并将它的值写入 SSPCON2 寄存器的 ACKSTAT 位。
- f) MSSP 模块在第 9 个时钟周期的末尾将 SSPIF 位置 1，产生一个中断。
- g) 用户将 8 位数据装入 SSPBUF。
- h) 数据从 SDA 引脚移出，直到发送完所有 8 位为止。
- i) MSSP 模块移入来自器件的 ACK 位，并将它的值写入 SSPCON2 寄存器的 ACKSTAT 位。
- j) MSSP 模块在第 9 个时钟的末尾将 SSPIF 位置 1，产生一个中断。
- k) 用户通过将停止使能位（PEN）位（SSPCON2 寄存器）置 1 产生停止条件。
- l) 一旦停止条件完成，将产生一个中断。

PIC16F882/883/884/886/887

13.4.5 波特率发生器

在 I²C 主控模式下，波特率发生器的重载值位于 SSPADD 寄存器的低 7 位（图 13-11）。当装载了该值后，波特率发生器将自动开始计数并递减至 0，然后停止直到下次重载为止。BRG 会在每个指令周期（T_{cy}）中的 Q2 和 Q4 时钟周期上进行两次减计数。在 I²C 主控模式下，会自动重载 BRG。例如，在发生时钟仲裁时，BRG 将在 SCL 引脚采样到高电平时重载（图 13-12）。

图 13-11： 波特率发生器框图

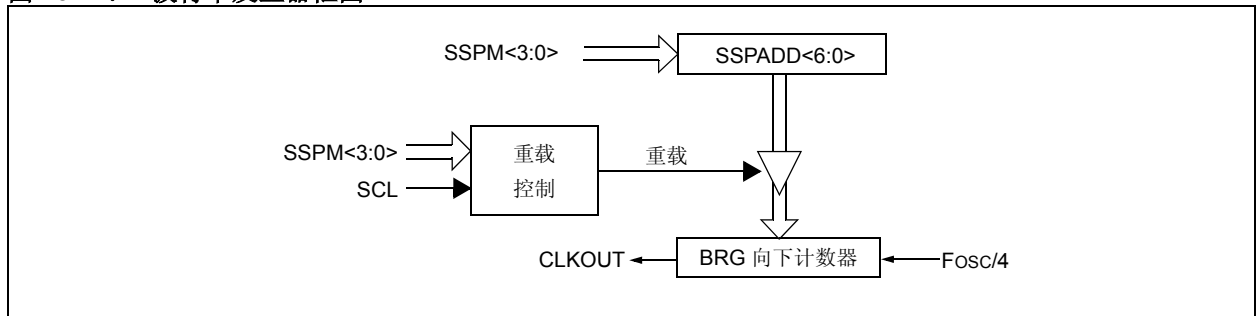
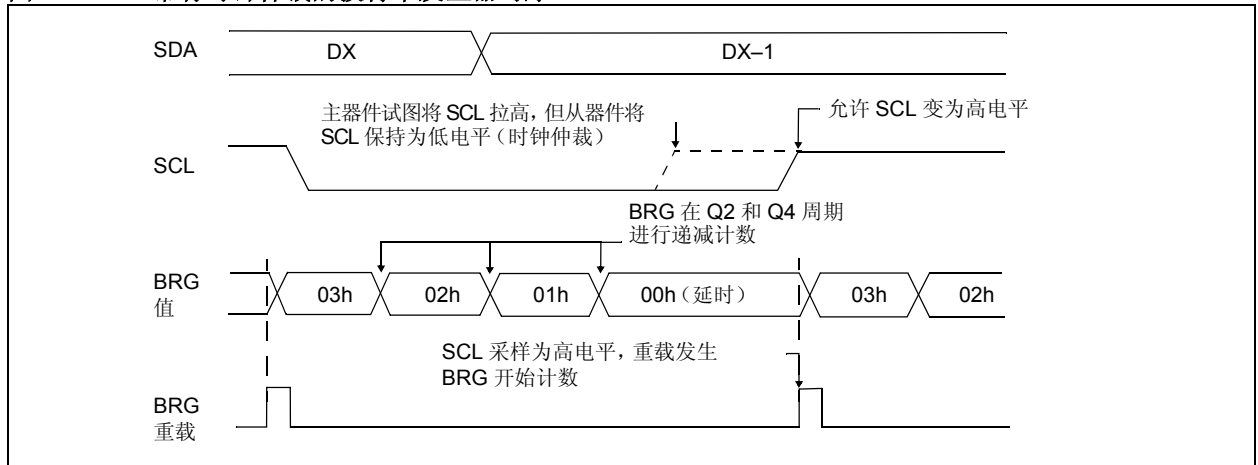


图 13-12： 带有时钟仲裁的波特率发生器时序



PIC16F882/883/884/886/887

13.4.6 I²C™ 主控模式启动条件时序

要发起启动条件，用户应将 SSPCON2 寄存器的启动条件使能位 SEN 置 1。当 SDA 和 SCL 引脚都采样为高电平时，波特率发生器重新装入 SSPADD<6:0> 的内容并开始计数。当波特率发生器发生超时 (TBRG) 时，如果 SCL 和 SDA 都采样为高电平，则 SDA 引脚被驱动为低电平。当 SCL 为高电平时，将 SDA 驱动为低电平就是启动条件，将使 S 位 (SSPSTAT 寄存器) 置 1。随后波特率发生器重新装入 SSPADD<6:0> 的内容并恢复计数。当波特率发生器超时 (TBRG) 时，SSPCON2 寄存器的 SEN 位将自动被硬件清零。波特率发生器暂停工作，SDA 线保持低电平，启动条件结束。

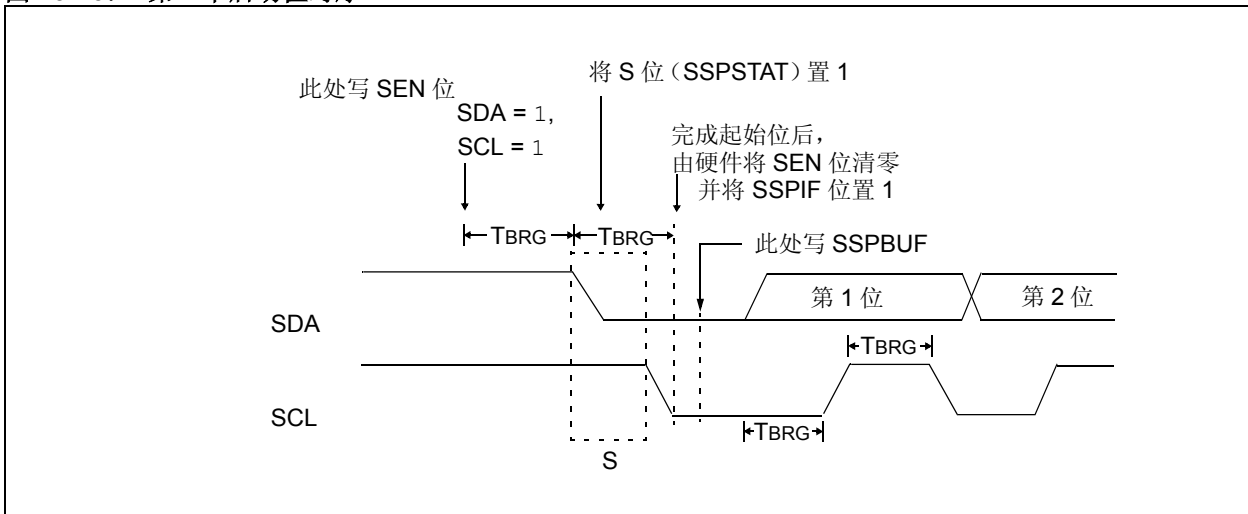
注： 如果在启动条件开始时，SDA 和 SCL 引脚已经采样为低电平，或者在启动条件期间，SCL 在 SDA 线被驱动为低电平之前已经采样为低电平，则会发生总线冲突。总线冲突中断标志位 BCLIF 置 1，启动条件中止，I²C 模块复位到空闲状态。

13.4.6.1 WCOL 状态标志

当启动序列进行时，如果用户写 SSPBUF，则 WCOL 被置 1，同时缓冲器内容不变（未发生写操作）。

注： 由于不允许事件排队，在启动条件结束之前，不能对 SSPCON2 的低 5 位进行写操作。

图 13-13: 第一个启动位时序



PIC16F882/883/884/886/887

13.4.7 I²C™ 主控模式重复启动条件时序

将 RSEN 位 (SSPCON2 寄存器) 编程为高电平, 并且 I²C 逻辑模块处于空闲状态时, 就会产生重复启动条件。当 RSEN 位置 1 时, SCL 引脚被拉为低电平。当 SCL 引脚采样为低电平时, 波特率发生器装入 SSPADD<6:0> 的内容, 并开始计数。在一个波特率发生器计数周期 (TBRG) 内 SDA 引脚被释放 (其引脚电平被拉高)。当波特率发生器超时, 如果 SDA 采样为高电平, SCL 引脚将被拉高。当 SCL 引脚采样为高电平时, 波特率发生器将被重新装入 SSPADD<6:0> 的内容并开始计数。SDA 和 SCL 必须在一个计数周期 TBRG 内采样为高电平。随后将 SDA 引脚拉为低电平 (SDA = 0) 并保持一个计数周期 TBRG, 同时 SCL 为高电平。然后 RSEN 位 (SSPCON2 寄存器) 将自动清零, 波特率发生器不会重载, SDA 引脚保持低电平。一旦在 SDA 和 SCL 引脚上检测到启动条件, S 位 (SSPSTAT 寄存器) 将被置 1。直到波特率发生器超时后, SSPIF 位才会置 1。

- 注 1:** 有任何其他事件进行时, 对 RSEN 的编程无效。
- 注 2:** 在重复启动条件期间, 下列事件将会导致总线冲突:
- 当 SCL 由低电平变为高电平时, SDA 采样为低电平。
 - 在 SDA 被拉低之前, SCL 变为低电平。这表示可能有另一个主器件正尝试发送数据 1。

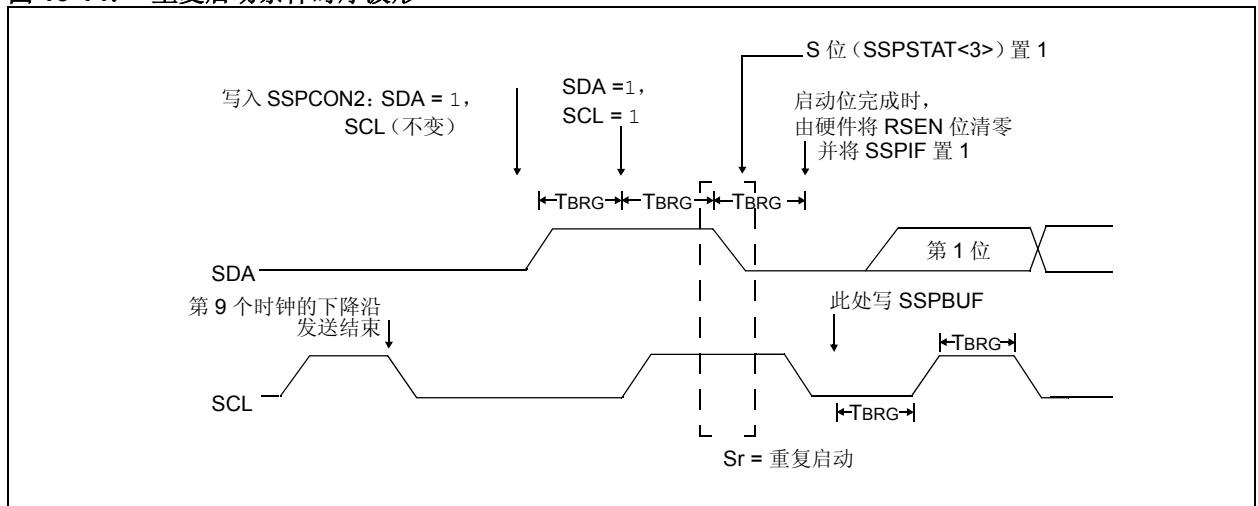
一旦 SSPIF 位被置 1, 用户便可以在 7 位地址模式下将 7 位地址写入 SSPBUF, 或者在 10 位地址模式下写入默认的第一个地址字节。当发送完第一个 8 位并接收到一个 ACK 后, 用户可以发送另外 8 位地址 (10 位地址模式下) 或 8 位数据 (7 位地址模式下)。

13.4.7.1 WCOL 状态标志

当重复启动序列进行时, 如果用户写 SSPBUF, 则 WCOL 被置 1, 同时缓冲器内容不变 (未发生写操作)。

- 注:** 由于不允许事件排队, 在重复启动条件结束之前, 不能对 SSPCON2 的低 5 位进行写操作。

图 13-14: 重复启动条件时序波形



PIC16F882/883/884/886/887

13.4.8 I²C™ 主控模式发送

发送一个数据字节、一个 7 位地址或一个 10 位地址的另一半，都可以直接通过写一个值到 SSPBUF 寄存器来实现。该操作将使缓冲器满标志位 BF 置 1，并且波特率发生器开始计数，同时启动下一次发送。在 SCL 的下降沿有效后（见数据保持时间规范参数 106），地址 / 数据的每一位将被移出至 SDA 引脚。在一个波特率发生器计满返回计数周期（TBRG）内，SCL 保持低电平。数据应该在 SCL 释放为高电平前保持有效（见数据建立时间规范参数 107）。当 SCL 引脚被释放为高电平时，它将在整个 TBRG 中保持高电平状态。在此期间以及下一个 SCL 下降沿之后的一段时间内，SDA 引脚上的数据必须保持稳定。在第 8 位被移出（第 8 个时钟周期的下降沿）之后，BF 标志位清零，同时主器件释放 SDA。此时如果发生地址匹配或是数据被正确接收，被寻址的从器件将在第 9 位的时间以一个 ACK 位响应。ACK 的状态在第 9 个时钟周期的下降沿写入 ACKDT 位。主器件接收到应答之后，应答状态位 ACKSTAT 会被清零；如果未收到应答，则该位被置 1。第 9 个时钟之后，SSPIF 位会置 1，主控时钟（波特率发生器）暂停，直到下一个数据字节装入 SSPBUF 为止，SCL 引脚保持低电平，SDA 保持不变（图 13-14）。

在写 SSPBUF 之后，地址的每一位在 SCL 的下降沿被移出，直至地址的所有 7 位和 R/W 位都被移出为止。在第 8 个时钟的下降沿，主器件将 SDA 引脚拉为高电平，以允许从器件发出应答响应。在第 9 个时钟的下降沿，主器件通过采样 SDA 引脚来判断地址是否被从器件识别。ACK 位的状态被装入 ACKSTAT 状态位（SSPCON2 寄存器）。在发送地址的第 9 个时钟下降沿之后，SSPIF 置 1，BF 标志位清零，波特率发生器关闭直到下一次写 SSPBUF，且 SCL 引脚保持低电平，允许 SDA 引脚悬空。

13.4.8.1 BF 状态标志

在发送模式下，BF 位（SSPSTAT 寄存器）在 CPU 写 SSPBUF 时置 1，在所有 8 位数据移出后清零。

13.4.8.2 WCOL 状态标志位

如果用户在发送过程中（即，SSPSR 仍在移出数据字节时）写 SSPBUF，则 WCOL 置 1 且缓冲器的内容保持不变（未发生写操作）。WCOL 必须由软件清零。

13.4.8.3 ACKSTAT 状态标志

在发送模式下，当从器件发送应答响应（ACK = 0）时，ACKSTAT 位（SSPCON2 寄存器）清零；当从器件没有应答（ACK = 1）时，该位置 1。从器件在识别出其地址（包括广播呼叫地址）或正确接收数据后，会发送一个应答。

13.4.9 I²C™ 主控模式接收

通过编程接收使能位 RCEN（SSPCON2 寄存器）使能主控模式接收。

注： RCEN 位置 1 前，MSSP 模块必须处于空闲状态，否则 RCEN 位将被忽略。

波特率发生器开始计数，每次计满返回时，SCL 引脚的状态都发生改变（由高变低或由低变高），且数据被移入 SSPSR。第 8 个时钟的下降沿之后，接收使能标志位自动清零，SSPSR 的内容装入 SSPBUF，BF 标志位置 1，SSPIF 标志位置 1，波特率发生器暂停计数，SCL 保持为低电平。此时 MSSP 处于空闲状态，等待下一条命令。当 CPU 读缓冲器时，BF 标志位将自动清零。通过将应答序列使能位 ACKEN（SSPCON2 寄存器）置 1，用户可以在接收结束后发送应答位。

13.4.9.1 BF 状态标志

接收时，当将地址或数据字节从 SSPSR 装入 SSPBUF 时，BF 位置 1；在读 SSPBUF 寄存器时 BF 位清零。

13.4.9.2 SSPOV 状态标志

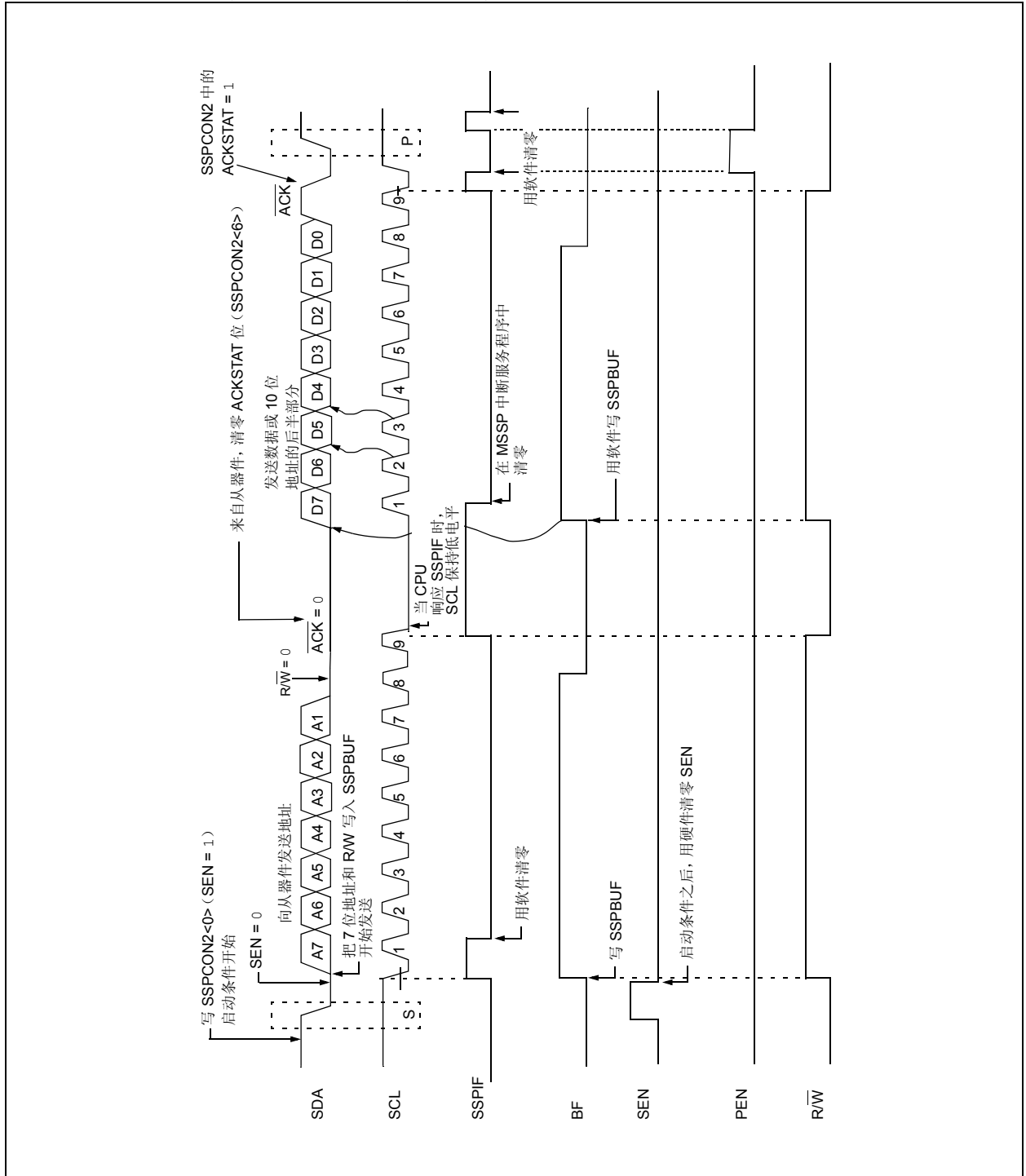
接收时，当 SSPSR 接收到 8 位数据时，SSPOV 位置 1，BF 标志位已经在上一次接收时置 1。

13.4.9.3 WCOL 状态标志

如果用户在接收过程中（即，SSPSR 仍在移入数据字节时）写 SSPBUF，则 WCOL 位置 1，缓冲器内容不变（未发生写操作）。

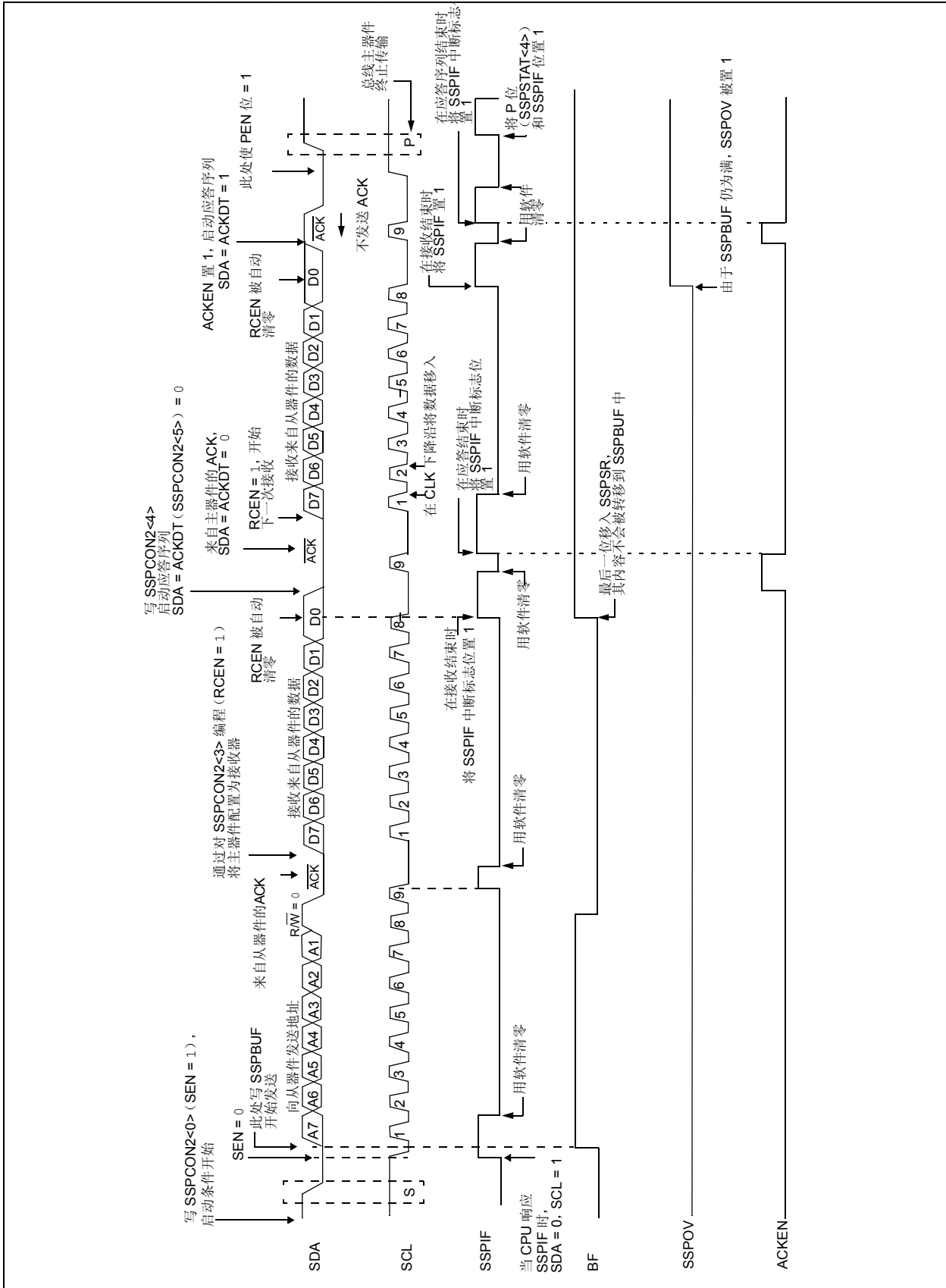
PIC16F882/883/884/886/887

图 13-15: I²C™ 主控模式发送时序 (7 位或 10 位地址)



PIC16F882/883/884/886/887

图 13-16: I²C™ 主控模式接收时序 (7 位地址)



PIC16F882/883/884/886/887

13.4.10 应答序列时序

将应答序列使能位 **ACKEN** (**SSPCON2** 寄存器) 置 1 即可使能应答序列。当该位被置 1 时, **SCL** 引脚被拉低, 应答数据位的内容出现在 **SDA** 引脚上。如果用户希望产生一个应答, 则应该将 **ACKDT** 位置 1。然后波特率发生器进行一个计满返回周期 (**TBRG**) 的计数, 随后 **SCL** 引脚电平被拉高。当 **SCL** 引脚采样为高电平时 (时钟仲裁), 波特率发生器再进行一个 **TBRG** 周期的计数。然后 **SCL** 引脚被拉低。在这之后, **ACKEN** 位自动清零, 波特率发生器关闭, **MSSP** 模块进入空闲模式 (图 13-17)。

13.4.10.1 WCOL 状态标志位

如果用户在应答序列正在进行时写 **SSPBUF**, **WCOL** 将被置 1 且缓冲器的内容保持不变 (未发生写操作)。

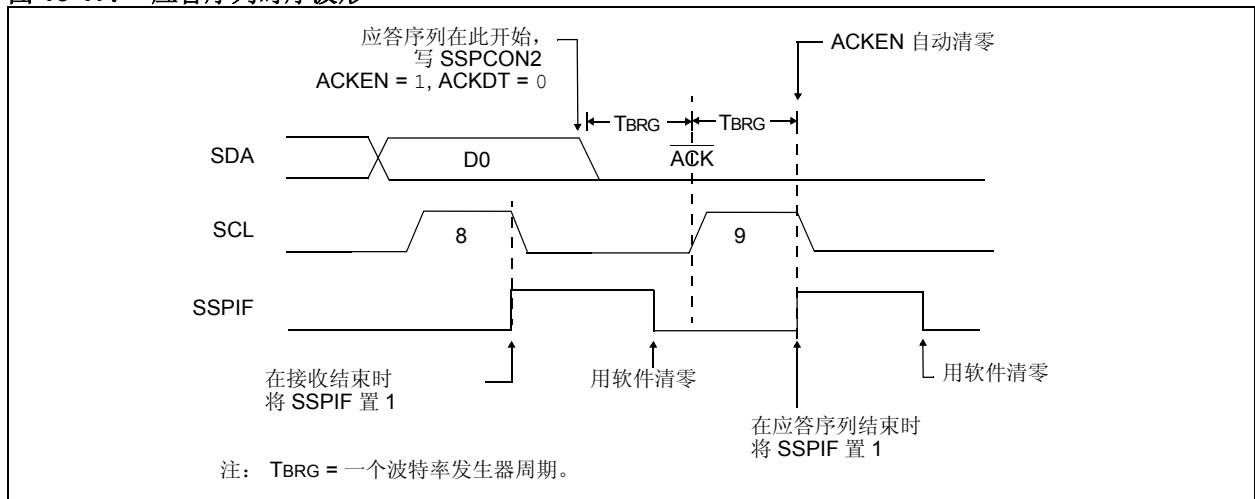
13.4.11 停止条件序列

在接收 / 发送结束时, 通过置停止序列的使能位, **PEN** (**SSPCON2** 寄存器), **SDA** 引脚将产生一个停止位。在接收 / 发送结束时, **SCL** 引脚在第 9 个时钟的下降沿后保持低电平。当 **PEN** 位置 1 时, 主控器件将 **SDA** 置为低电平。当 **SDA** 线采样为低电平时, 波特率发生器被重新装入值并递减计数至 0。波特率发生器发生超时时, **SCL** 引脚被拉到高电平, 且一个 **TBRG** (波特率发生器计满回零) 后, **SDA** 引脚被重新拉到高电平。当 **SDA** 引脚采样为高电平且 **SCL** 也是高电平时, **P** 位 (**SSPSTAT** 寄存器) 置 1。一个 **TBRG** 周期后, **PEN** 位清零且 **SSPIF** 位置 1 (图 13-17)。

13.4.11.1 WCOL 状态标志

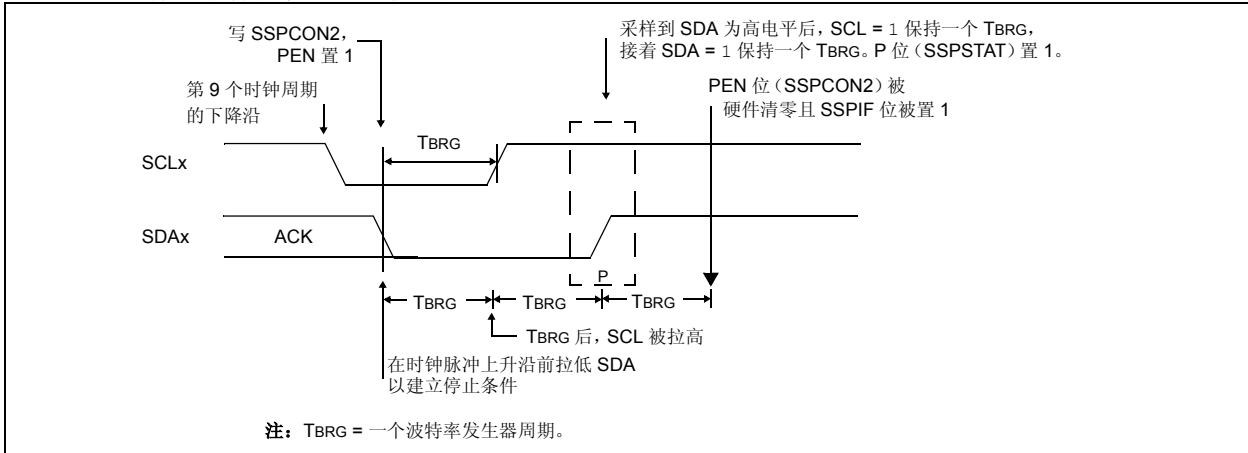
如果用户在停止序列进行过程中试图写 **SSPBUF**, 则 **WCOL** 位将置 1, 缓冲器的内容不会改变 (未发生写操作)。

图 13-17: 应答序列时序波形



PIC16F882/883/884/886/887

图 13-18: 停止条件接收或发送模式



13.4.12 时钟仲裁

如果在任何接收、发送或重复启动 / 停止条件期间, 主器件拉高了 SCL 引脚 (允许 SCL 引脚悬空为高电平), 就会发生时钟仲裁。如果允许 SCL 引脚悬空为高电平, 波特率发生器 (BRG) 将暂停计数, 直到实际采样到 SCL 引脚为高电平为止。当 SCL 引脚采样为高电平时, 波特率发生器中将被重新装入 SSPADD<6:0> 的内容并开始计数。这可以保证当外部器件将时钟拉低时, SCL 始终保持至少一个 BRG 计满返回周期的高电平 (图 13-19)。

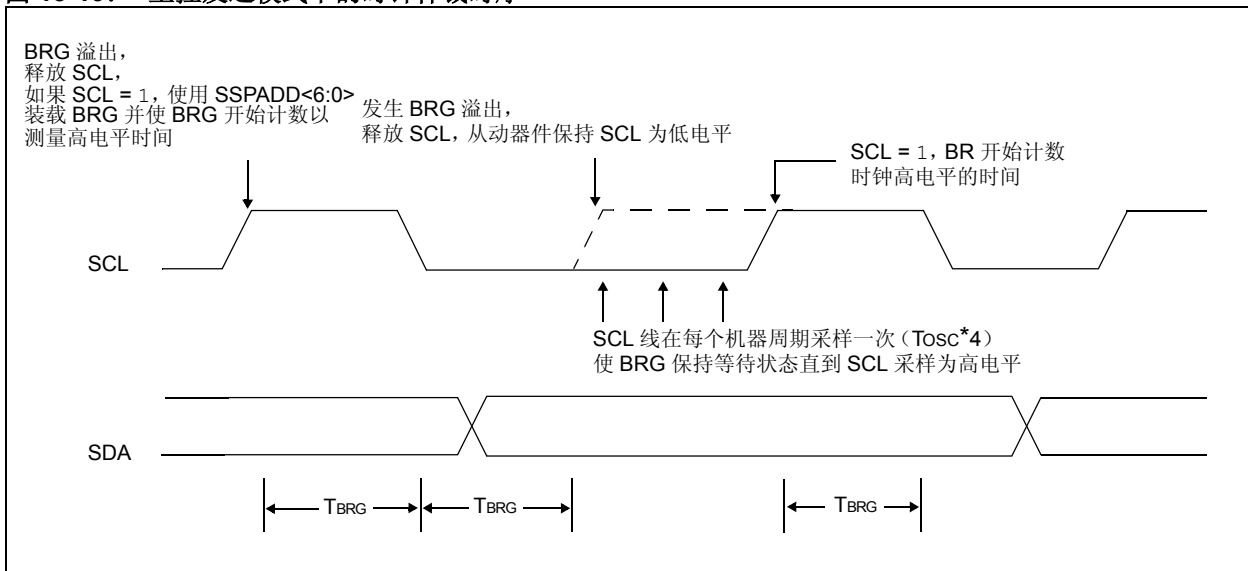
13.4.13 休眠模式下的操作

在休眠模式下, I²C 模块能够接收地址或数据。并且在地址匹配或字节传输完成后, 将唤醒处理器 (如果允许了 MSSP 中断)。

13.4.14 复位的影响

复位会禁止 MSSP 模块并终止当前的传输。

图 13-19: 主控发送模式下的时钟仲裁时序



PIC16F882/883/884/886/887

13.4.15 多主机模式

在多主机模式下，通过在检测到启动和停止条件时产生中断可以确定总线何时空闲。停止（P）位和启动（S）位在复位时或禁止 MSSP 模块时清零。当 P 位置 1 时，可以取得 I²C 总线的控制权；否则总线处于空闲状态，且 P 位和 S 位清零。当总线忙时，如果出现停止条件，则将产生中断（若允许 MSSP 中断）。

在多主机模式下工作时，必须监视 SDA 线来进行仲裁，查看信号电平是否为期望的输出电平。此检查由硬件完成，其结果放在 BCLIF 位。

在以下状态下仲裁可能失败：

- 地址传输
- 数据传输
- 启动条件
- 重复启动条件
- 应答条件

13.4.16 多主机通信、总线冲突与总线仲裁

多主机模式是通过总线仲裁来支持的。当主器件将地址 / 数据位输出到 SDA 引脚时，如果一个主器件通过将 SDA 引脚悬空为高电平以在 SDA 上输出 1，而另一个

主器件输出 0，就会发生总线仲裁。如果 SDA 引脚上期望的数据是 1，而实际在 SDA 引脚上采样到的数据是 0，则发生了总线冲突。主器件将把总线冲突中断标志位 BCLIF 置 1，并将 I²C 端口复位到空闲状态（图 13-19）。

如果在发送过程中发生总线冲突，则发送停止，BF 标志位清零，SDA 和 SCL 线被拉高，并且允许对 SSPBUF 进行写操作。当执行完总线冲突中断服务程序后，如果 I²C 总线空闲，用户可通过发出启动条件恢复通信。

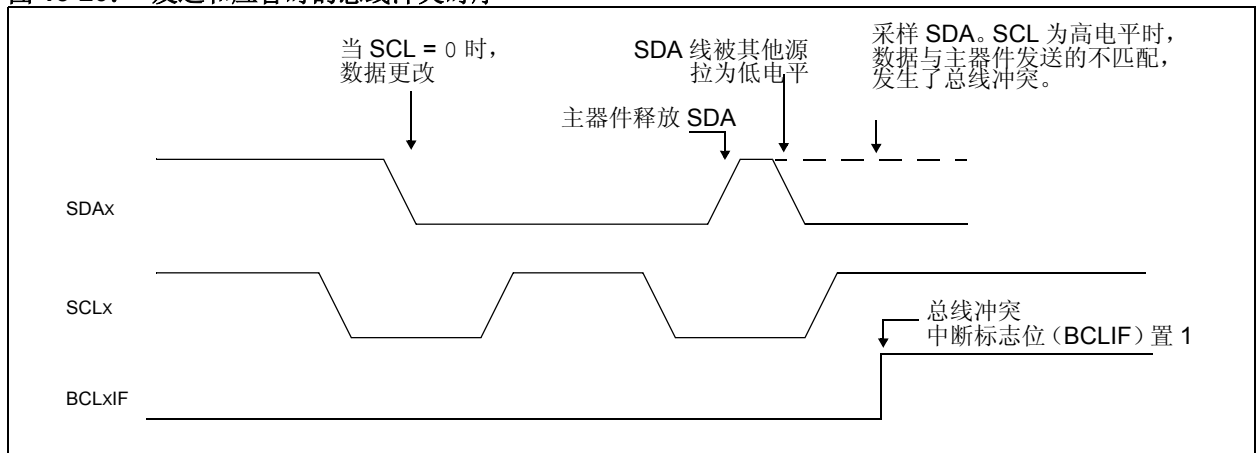
如果在启动、重复启动、停止或应答条件的进行过程中发生总线冲突，则该条件被中止，SDA 和 SCL 线被拉高，SSPCON2 寄存器中的对应控制位清零。当执行完总线冲突中断服务程序后，如果 I²C 总线空闲，用户可以通过发出启动条件恢复通信。

主器件将继续监视 SDA 和 SCL 引脚。如果出现停止条件，SSPIF 位将被置 1。

无论发生总线冲突时发送的进度如何，写 SSPBUF 都会从第一个数据位开始发送数据。

在多主机模式下，通过在检测到启动和停止条件时产生中断可以确定总线何时空闲。P 位置 1 时，可以获取 I²C 总线的控制权，否则总线空闲且 S 和 P 位清零。

图 13-20： 发送和应答时的总线冲突时序



PIC16F882/883/884/886/887

13.4.16.1 启动条件期间的总线冲突

启动条件期间，以下事件将导致总线冲突：

- a) 在启动条件开始时，SDA 或 SCL 被采样为低电平（图 13-20）。
- b) SDA 被拉低之前，SCL 采样为低电平（图 13-22）。

在启动条件期间，SDA 和 SCL 引脚都会被监视。

如果 SDA 引脚已经是低电平，或 SCL 引脚已经是低电平，则：

中止启动条件，

并且 BCLIF 标志位置 1，

并且 MSSP 模块复位为空闲状态（图 13-20）。

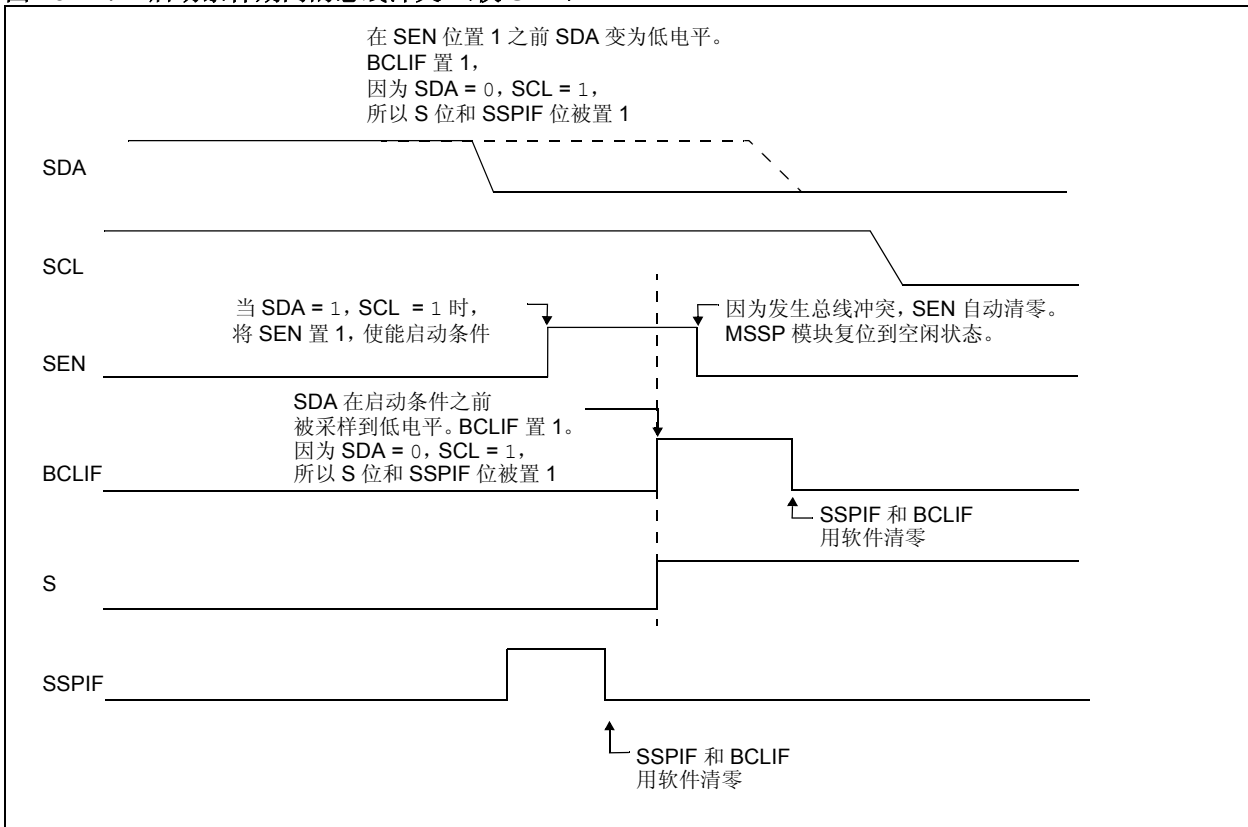
启动条件从 SDA 和 SCL 引脚被拉高开始。当 SDA 引脚采样为高电平时，波特率发生器装入 SSPADD<6:0> 的值并递减计数到 0。如果在 SDA 为高电平时，SCL 引

脚采样为低电平，则发生总线冲突，因为这表示另一个主器件在启动条件期间试图发送一个数据 1。

如果 SDA 引脚在该计数周期内采样为低电平，则 BRG 复位，同时 SDA 线保持原值（图 13-22）。但是，如果 SDA 引脚采样为 1，SDA 引脚将在 BRG 计数结束时被置为低电平。随后波特率发生器被重新装入值并递减计数至 0。在此期间，如果 SCL 引脚采样到 0，则没有发生总线冲突。在 BRG 计数结束时，SCL 引脚被拉为低电平。

注： 在启动条件期间不会发生总线冲突是因为两个总线主器件不可能精确地在同一时刻发出启动条件。因此总是有一个主器件先于另一个主器件将 SDA 拉低。但是这一情况不会引起总线冲突，因为允许两个主器件对启动条件后的第一个地址进行仲裁。如果地址是相同的，将继续对数据部分、重复启动条件或停止条件进行仲裁。

图 13-21： 启动条件期间的总线冲突（仅 SDA）



PIC16F882/883/884/886/887

图 13-22: 启动条件期间的总线冲突 (SCL = 0)

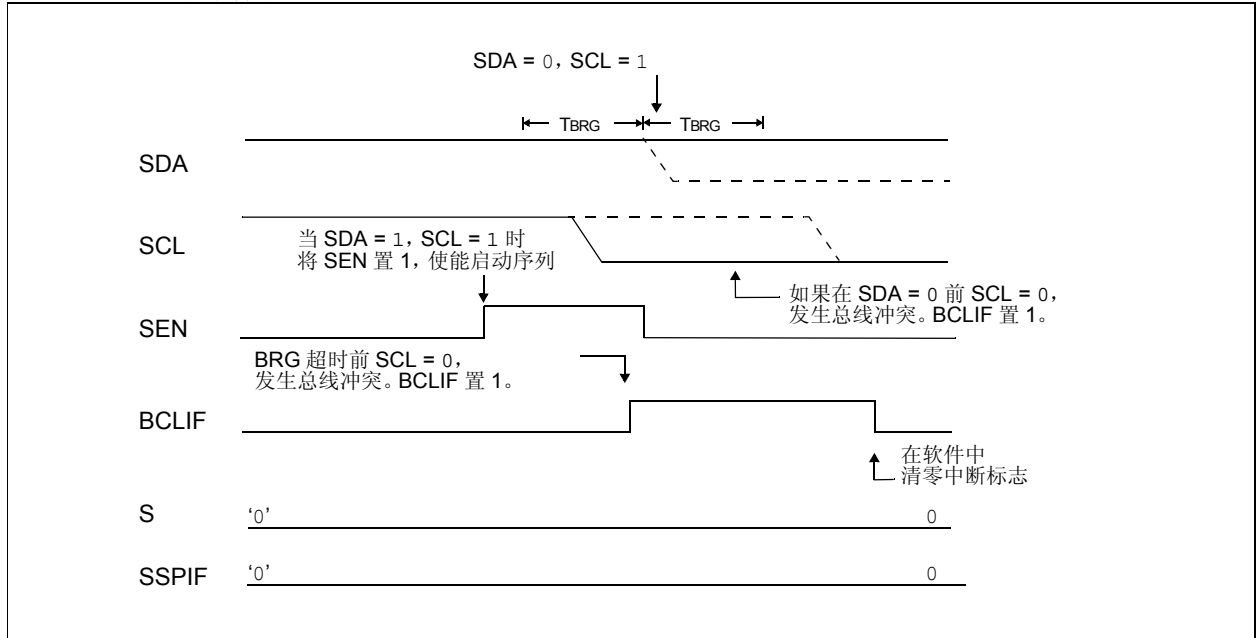
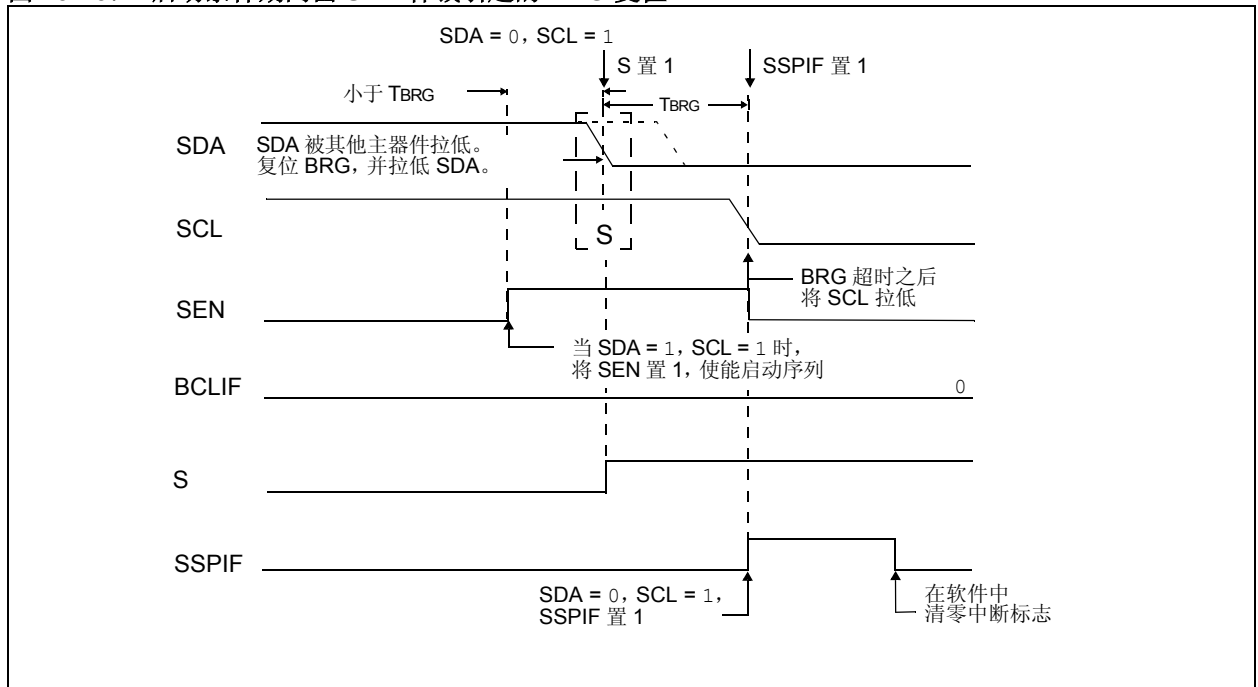


图 13-23: 启动条件期间由 SDA 仲裁引起的 BRG 复位



PIC16F882/883/884/886/887

13.4.16.2 重复启动条件期间的总线冲突

在下列情况中，重复启动条件期间会发生总线冲突：

- a) 在 SCL 由低电平变为高电平的过程中，SDA 采样到低电平。
- b) 在 SDA 被拉为低电平之前，SCL 变为低电平，表示另一个主器件正试图发送一个数据 1。

当用户拉高 SDA 并允许该引脚悬空时，BRG 中装入 SSPADD<6:0> 中的值并递减计数至 0。接着 SCL 引脚被置为高电平，当 SCL 采样到高电平时，对 SDA 引脚进行采样。

如果 SDA 为低电平，则已发生了总线冲突（即，另一个主器件正试图发送一个数据 0，见图 13-24）。如果 SDA 采样为高电平，则 BRG 被重新装入值并开始计数。如果 SDA 在 BRG 超时之前从高电平变为低电平，则没有发生总线冲突，因为两个主器件不可能精确地在同一时刻将 SDA 拉低。

如果 SCL 在 BRG 超时之前从高电平变为低电平，且 SDA 尚未变为低电平，表示发生了总线冲突。在此情况下，在重复启动条件期间另一个主器件正试图发送一个数据 1（见图 13-25）。

如果在 BRG 超时结束时 SCL 和 SDA 都仍然是高电平，则 SDA 引脚被拉低，BRG 重新装入值并开始计数。在计数结束时，无论 SCL 引脚的状态如何，SCL 引脚都被拉低，重复启动条件结束。

图 13-24： 重复启动条件期间的总线冲突（情形 1）

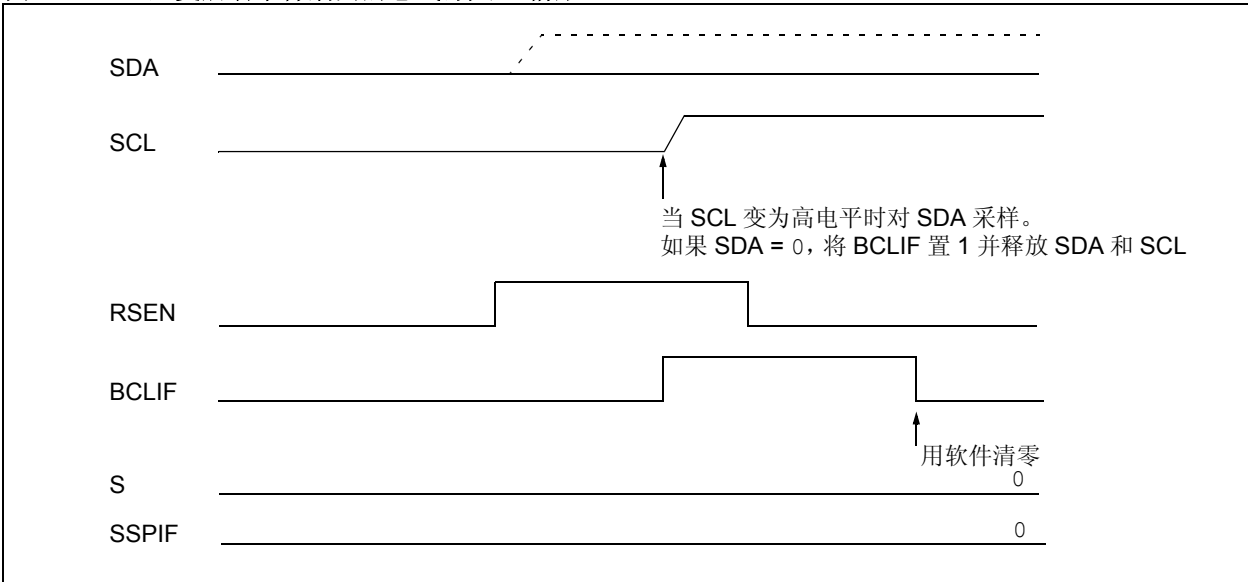
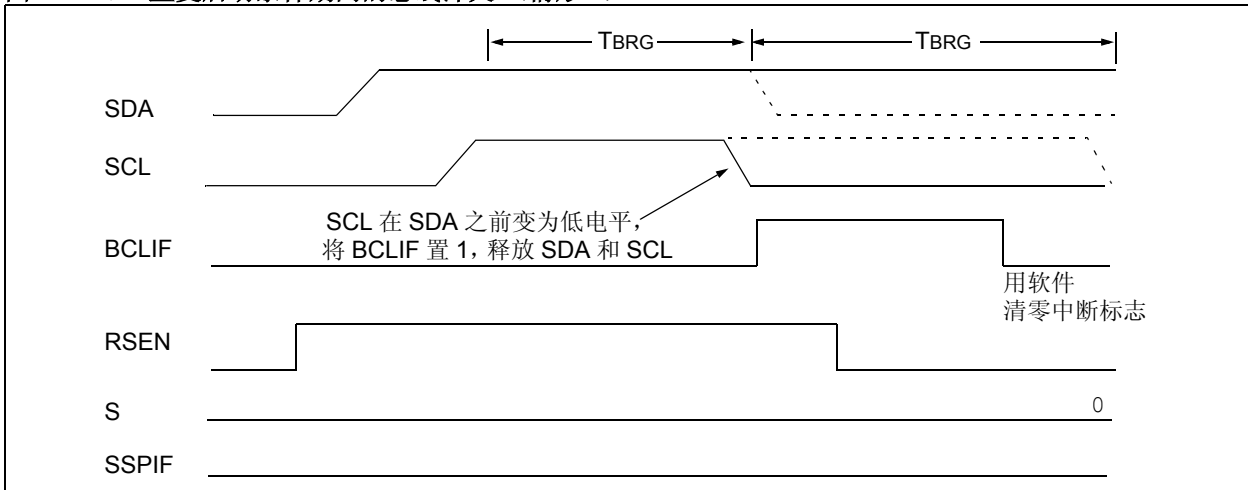


图 13-25： 重复启动条件期间的总线冲突（情形 2）



PIC16F882/883/884/886/887

13.4.16.3 停止条件期间的总线冲突

以下事件会导致停止条件期间的总线冲突：

- SDA 已被拉高并允许悬空为高电平之后，SDA 在 BRG 超时后被采样到低电平。
- SCL 引脚被拉高之后，SCL 在 SDA 变成高电平之前被采样到低电平。

停止条件从 SDA 被拉低开始。当 SDA 采样为低电平时，SCL 引脚就可以悬空为高电平。当引脚被采样到高电平时（时钟仲裁），波特率发生器中装入 SSPADD<6:0> 的内容并递减计数到 0。BRG 超时后，采样 SDA。如果 SDA 采样到低电平，则已发生总线冲突。这是因为另一个主器件正试图发送一个数据 0（图 13-26）。如果 SCL 引脚在允许 SDA 悬空为高电平前被采样到低电平，也会发生总线冲突。这是另一个主器件正试图发送一个数据 0 的又一种情况（图 13-27）。

图 13-26: 停止条件期间的总线冲突（情形 1）

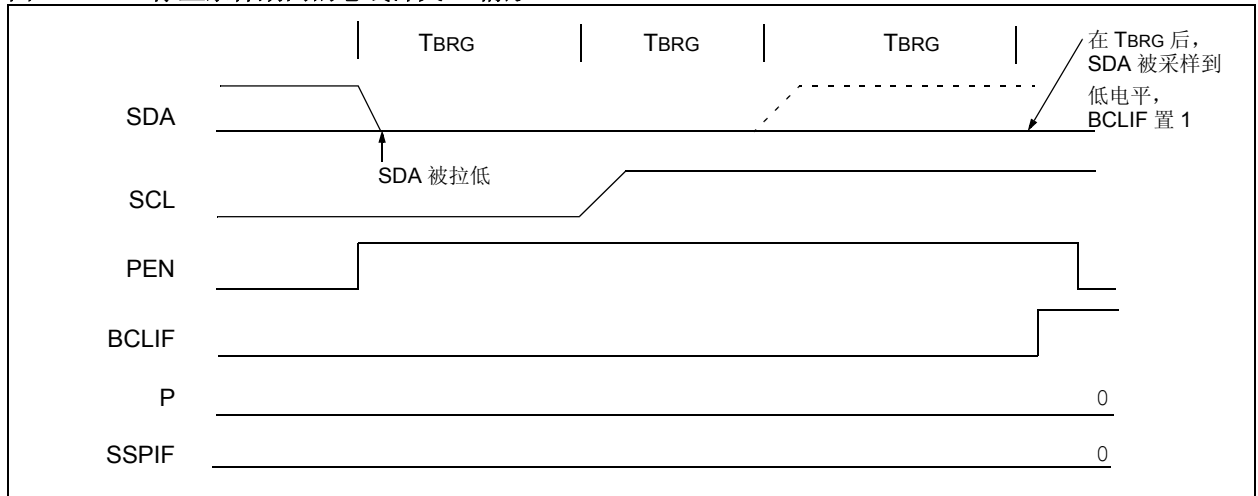
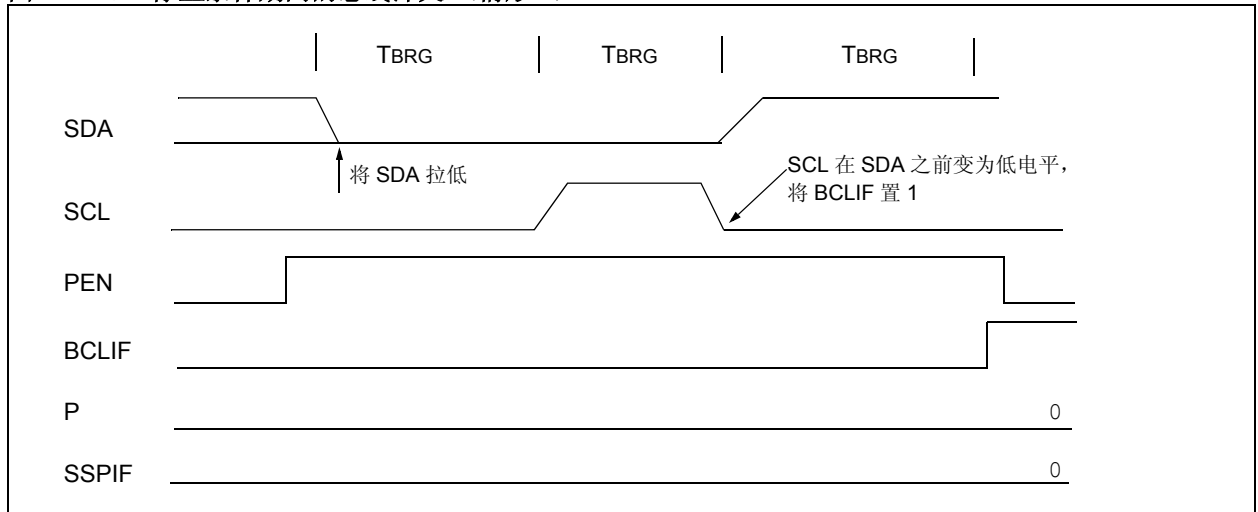


图 13-27: 停止条件期间的总线冲突（情形 2）



PIC16F882/883/884/886/887

13.4.17 SSP 屏蔽寄存器

在 I²C 从动模式下，SSP 屏蔽 (SSPMSK) 寄存器用于在地址比较操作下屏蔽 SSPSR 寄存器中的值。SSPMSK 寄存器中某位为 0 会使 SSPSR 寄存器中相应的位成为“无关位”。

此寄存器在任何复位条件发生时均复位为全 1，因此，在写入屏蔽值前，它对标准 SSP 操作没有影响。

必须在通过设置 SSPM<3:0> 位以选择 I²C 从动模式 (7 位或 10 位地址) 之前对此寄存器进行初始化。

只有通过 SSPCON 的 SSPM<3:0> 位选择了适当的模式后才可访问此寄存器。

SSP 屏蔽寄存器在以下情况下有效：

- 7 位地址模式：与 A<7:1> 进行地址比较。
- 10 位地址模式：仅与 A<7:0> 进行地址比较。
SSP 屏蔽在接收到地址的第一个 (高) 字节期间无效。

寄存器 13-4: SSPMSK: SSP 屏蔽寄存器⁽¹⁾

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0 ⁽²⁾
bit 7							bit 0

图注：

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

bit 7-1 **MSK<7:1>**: 屏蔽位
 1 = 接收到的地址的 bit n 与 SSPADD<n> 比较以检测 I²C 的地址匹配情况
 0 = 接收到的地址的 bit n 不用于检测 I²C 的地址匹配情况

bit 0 **MSK<0>**: I²C 从动模式 10 位地址的屏蔽位⁽²⁾
 I²C 从动模式，10 位地址 (SSPM<3:0> = 0111):
 1 = 接收到的地址的 bit 0 与 SSPADD<0> 比较以检测 I²C 的地址匹配情况
 0 = 接收到的地址的 bit 0 不用于检测 I²C 的地址匹配情况

注 1: 当 SSPCON 位 SSPM<3:0> = 1001 时，任何对 SSPADD SFR 地址的读或写操作都通过 SSPMSK 寄存器进行。

注 2: 在所有其他 SSP 模式下，此位无效。

PIC16F882/883/884/886/887

14.0 CPU 的特性

PIC16F882/883/884/886/887 系列器件包含的许多特性旨在最大限度地提高系统的可靠性，通过减少外部元件将成本降至最低，并且还提供了低功耗和代码保护功能。

这些功能包括：

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

PIC16F882/883/884/886/887 系列器件有两个用于提供必要的上电延时的定时器。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始终处于复位状态。另一个是上电延时定时器 (PWRT)，仅在上电时提供 64 ms (标称值) 的固定延时，用来确保器件在供电电压稳定之前处于复位状态。如果出现欠压条件，同样有可使器件复位的电路，该电路使用上电延时定时器提供至少 64 ms 的复位延时。具有这三种片上功能，大多数应用将不再需要外部复位电路。

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。用户可通过以下方法将器件从休眠模式唤醒：

- 外部复位
- 看门狗定时器唤醒
- 中断

还有几种振荡器可供选择，以使器件适应各种应用。选择 INTOSC 可节约系统的成本，而选择 LP 晶振可以节能。可以使用一组配置位来选择各种时钟选项 (见寄存器 14-3)。

PIC16F882/883/884/886/887

14.1 配置位

可以通过对配置位编程（读为 0）或不编程（读为 1）来选择不同的器件配置，如寄存器 14-1 所示。这些配置位被映射到程序存储器中地址为 2007h 的单元中。

注： 地址 2007h 超过了用户程序存储空间，该地址属于特殊配置存储空间（2000h-3FFFh），仅可在编程期间进行访问。请参见“PIC16F88X Memory Programming Specification”（DS41287）了解更多信息。

寄存器 14-1: CONFIG1: 配置字寄存器 1

		$\overline{\text{DEBUG}}$	LVP	FCMEN	IESO	BOREN1	BOREN0
bit 15							bit 8
$\overline{\text{CPD}}$	$\overline{\text{CP}}$	MCLR $\overline{\text{E}}$	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0
bit 7							bit 0

- bit 15-14 **未实现：** 读为 1
- bit 13 **DEBUG：** 在线调试器模式位
1 = 禁止在线调试器，RB6/ICSPCLK 和 RB7/ICSPDAT 为通用 I/O 引脚
0 = 使能在线调试器，RB6/ICSPCLK 和 RB7/ICSPDAT 专用于调试器
- bit 12 **LVP：** 低电压编程使能位
1 = RB3/PGM 引脚具有 PGM 功能，使能低电压编程
0 = RB3 引脚为数字 I/O，MCLR 上的 HV 必须用于编程
- bit 11 **FCMEN：** 故障保护时钟监视器使能位
1 = 使能故障保护时钟监视器
0 = 禁止故障保护时钟监视器
- bit 10 **IESO：** 内外时钟切换位
1 = 使能内外时钟切换模式
0 = 禁止内外时钟切换模式
- bit 9-8 **BOREN<1:0>：** 欠压复位选择位⁽¹⁾
11 = 使能欠压复位
10 = 使能正常工作期间的欠压复位而禁止休眠状态下的欠压复位
01 = 由 PCON 寄存器中的 SBOREN 位控制欠压复位
00 = 禁止欠压复位
- bit 7 **CPD：** 数据代码保护位⁽²⁾
1 = 禁止数据存储代码保护
0 = 使能数据存储代码保护
- bit 6 **CP：** 代码保护位⁽³⁾
1 = 禁止程序存储器代码保护
0 = 使能程序存储器代码保护
- bit 5 **MCLR $\overline{\text{E}}$ ：** RB3/MCLR 引脚功能选择位⁽⁴⁾
1 = RB3/MCLR 引脚功能为 MCLR
0 = RB3/MCLR 引脚功能为数字输入，MCLR 在内部被连接到 VDD
- bit 4 **PWRTE $\overline{\text{N}}$ ：** 上电延时定时器使能位
1 = 禁止 PWRTE
0 = 使能 PWRTE
- bit 3 **WDTE：** 看门狗定时器使能位
1 = 使能 WDT
0 = WDT 被禁止，但可通过 WDTCON 寄存器中的 SWDTEN 位使能
- bit 2-0 **FOSC<2:0>：** 振荡器选择位
111 = RC 振荡器：RA6/OSC2/CLKOUT 引脚功能为 CLKOUT，RA7/OSC1/CLKIN 引脚连接 RC
110 = RCIO 振荡器：RA6/OSC2/CLKOUT 为 IO 引脚，RA7/OSC1/CLKIN 引脚连接 RC
101 = INTOSC 振荡器：RA6/OSC2/CLKOUT 引脚功能为 CLKOUT，RA7/OSC1/CLKIN 为 I/O 引脚
100 = INTOSCIO 振荡器：RA6/OSC2/CLKOUT 为 I/O 引脚，RA7/OSC1/CLKIN 也为 I/O 引脚
011 = EC：RA6/OSC2/CLKOUT 为 I/O 引脚，RA7/OSC1/CLKIN 引脚功能为 CLKIN
010 = HS 振荡模式：高速晶振 / 谐振器连接到 RA6/OSC2/CLKOUT 和 RA7/OSC1/CLKIN 引脚
001 = XT 振荡器：晶振 / 谐振器连接到 RA6/OSC2/CLKOUT 和 RA7/OSC1/CLKIN 引脚
000 = LP 振荡器：低功耗晶振连接到 RA6/OSC2/CLKOUT 和 RA7/OSC1/CLKIN 引脚

- 注**
- 1: 使能欠压复位并不能自动使能上电延时定时器。
 - 2: 当禁止代码保护时，将擦除整个数据 EEPROM 的内容。
 - 3: 当禁止代码保护时，将擦除整个程序存储器的内容。
 - 4: 当 MCLR 在 INTOSC 或者 RC 模式下被拉为低电平时，将禁止内部时钟振荡器。

PIC16F882/883/884/886/887

寄存器 14-2: CONFIG2: 配置字寄存器 2

—	—	—	—	—	WRT1	WRT0	BOR4V
bit 15					bit 8		
—	—	—	—	—	—	—	—
bit 7					bit 0		

bit 15-11 未实现: 读为 1

bit 10-9 **WRT<1:0>**: 闪存程序存储器自写使能位

PIC16F883/PIC16F884

00 = 0000h 至 07FFh 被写保护, 0800h 至 0FFFh 可由 EECON 控制寄存器修改
 01 = 0000h 至 03FFh 被写保护, 0400h 至 0FFFh 可由 EECON 控制寄存器修改
 10 = 0000h 至 00FFh 被写保护, 0100h 至 0FFFh 可由 EECON 控制寄存器修改
 11 = 关闭写保护

PIC16F886/PIC16F887

00 = 0000h 至 0FFFh 被写保护, 1000h 至 1FFFh 可由 EECON 控制寄存器修改
 01 = 0000h 至 07FFh 被写保护, 0800h 至 1FFFh 可由 EECON 控制寄存器修改
 10 = 0000h 至 00FFh 被写保护, 0100h 至 1FFFh 可由 EECON 控制寄存器修改
 11 = 关闭写保护

PIC16F882

00 = 0000h 至 03FFh 被写保护, 0400h 至 07FFh 可由 EECON 控制寄存器修改
 01 = 0000h 至 00FFh 被写保护, 0100h 至 07FFh 可由 EECON 控制寄存器修改
 11 = 关闭写保护

bit 8 **BOR4V**: 欠压复位选择位

0 = 欠压复位值被设置为 2.1V
 1 = 欠压复位值被设置为 4.0V

bit 7-0 未实现: 读为 1

PIC16F882/883/884/886/887

14.2 复位

PIC16F882/883/884/886/887 器件有以下几种不同类型的复位方式：

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

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

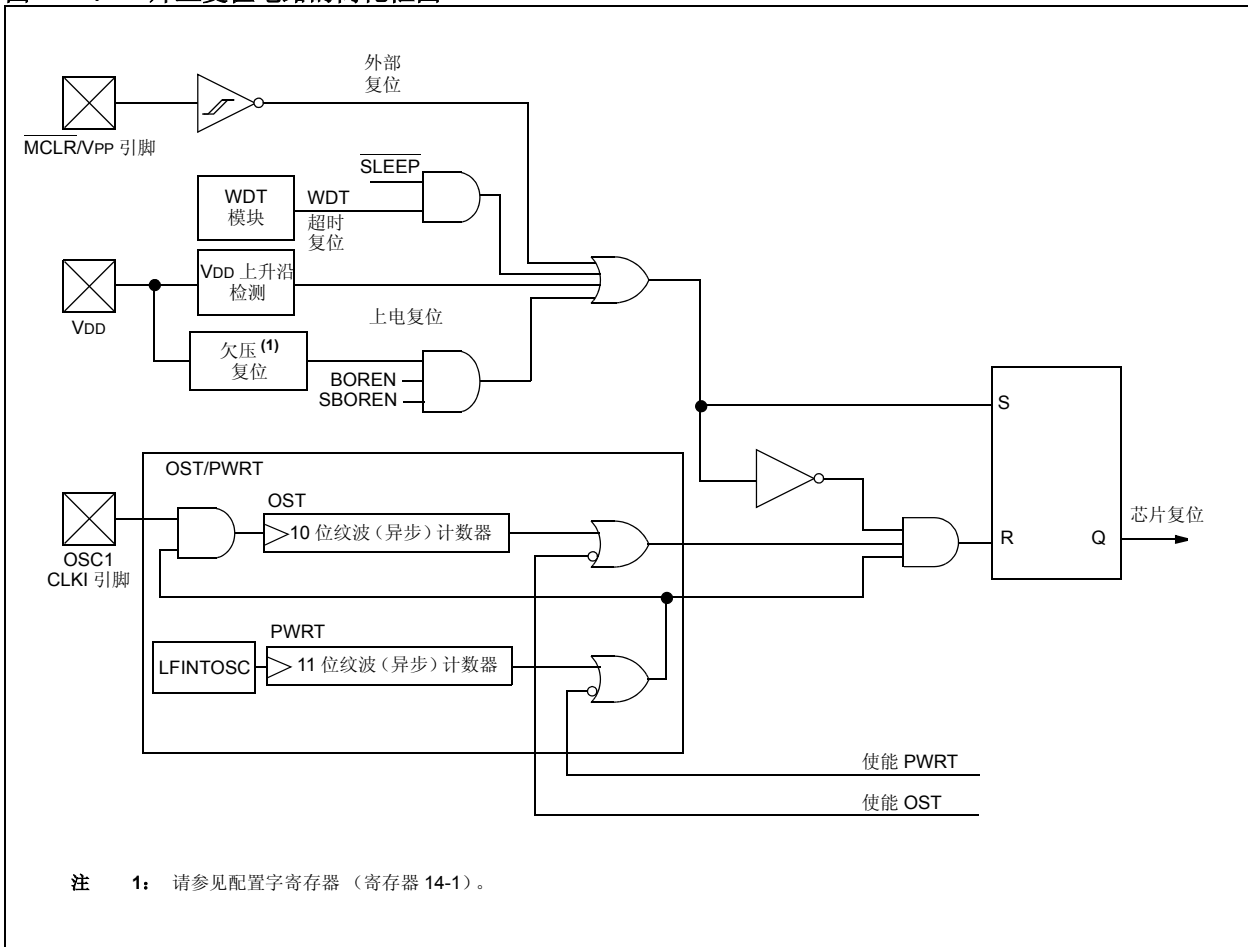
- 上电复位
- $\overline{\text{MCLR}}$ 复位
- 休眠期间的 $\overline{\text{MCLR}}$ 复位
- WDT 复位
- 欠压复位 (BOR)

它们不受 WDT 唤醒的影响，因为这被视为恢复正常工作。如表 14-2 所示， $\overline{\text{TO}}$ 和 $\overline{\text{PD}}$ 位在不同的复位情形下会分别被置 1 或清零。这些状态位在软件中用于判断复位的性质。表 14-5 对所有寄存器的复位状态做了完整的说明。

图 14-1 给出了片上复位电路的简化框图。

$\overline{\text{MCLR}}$ 复位电路中有一个噪声滤波器，它可以检测并滤除小脉冲干扰信号。请参见第 17.0 节“电气特性”了解脉宽规范。

图 14-1： 片上复位电路的简化框图



PIC16F882/883/884/886/887

14.2.1 上电复位 (POR)

在 VDD 达到足以使器件正常工作的电平以前，片上上电复位电路使器件保持复位状态。需要一个最大上升时间才能达到 VDD。详情请参见第 17.0 节“电气特性”。如果使能了欠压复位，那么该最大上升时间规范将不再适用。欠压复位电路在 VDD 达到 VBOR 以前将保持器件为复位状态（见第 14.2.4 节“欠压复位 (BOR)”）。

注： 当 VDD 降低时，上电复位电路不会产生内部复位。要重新使能上电复位，VDD 必须达到 VSS 电压并至少保持 100 μ s。

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

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

14.2.2 MCLR

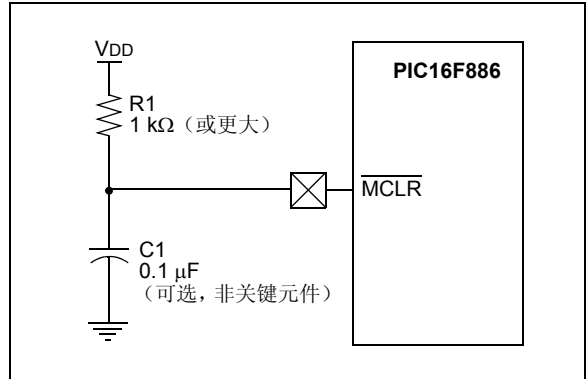
PIC16F882/883/884/886/887 器件在 MCLR 复位电路中有一个噪声滤波器。该滤波器检测并滤除小的干扰脉冲。

应该注意 WDT 复位不会将 MCLR 引脚驱动为低电平。

MCLR 引脚上 ESD 保护的工作原理与该系列早期的器件有所不同。该引脚上电压超过规范值将导致 MCLR 复位，并且在 ESD 事件中产生的电流也将超过器件的规范值。因此，Microchip 建议不要把 MCLR 引脚直接连接到 VDD。建议使用图 14-2 所示的 RC 网络。

通过清零配置寄存器 1 中的 MCLRE 位可使能内部 MCLR 选项。当 MCLRE = 0 时，由内部产生芯片的复位信号。当 MCLRE = 1 时，RA3/MCLR 引脚变为外部复位引脚。在该模式下，RA3/MCLR 引脚具备到 VDD 的弱上拉。

图 14-2: 推荐的 MCLR 电路



14.2.3 上电延时定时器 (PWRT)

上电延时定时器仅在上电时（上电复位或欠压复位）提供 64 ms（标称值）的固定延时。上电延时定时器采用 LFINTOSC 振荡器作为时钟源，工作频率为 31 kHz。更多信息，请参见第 4.5 节“内部时钟模式”。只要 PWRT 处于工作状态，器件就保持在复位状态。PWRT 延时使 VDD 有足够的时间上升到所需的电平。配置位 PWRTE 可以禁止（如果置 1）或使能（如果清零或被编程）上电延时定时器。虽然不是必需的，但是在使能欠压复位时也应使能上电延时定时器。

由于以下原因不同芯片的上电延时定时器的延迟时间也互不相同：

- VDD 差异
- 温度差异
- 制造工艺差异

详情请参见直流参数（第 17.0 节“电气特性”）。

PIC16F882/883/884/886/887

14.2.4 欠压复位 (BOR)

配置字寄存器中的 **BOREN0** 和 **BOREN1** 位用于选择四种欠压复位模式中的一种。其中增加了两种允许使用软件或硬件对欠压复位的使能与否进行控制的模式。当 **BOREN<1:0> = 01** 时, 可由 **SBOREN** 位 (**PCON<4>**) 使能或禁止欠压复位, 从而能用软件对其进行控制。通过对 **BOREN<1:0>** 进行选择, 可自动在休眠时禁止 **POR** 以节约功耗, 而在唤醒后使能 **POR**。在此模式下, **SBOREN** 位无效。关于配置字的定义, 请参见寄存器 14-3。

配置字寄存器 2 中的 **BOR4V** 位选择两种欠压复位电压之一。当 **BOR4B = 1** 时, **VBOR** 被设置为 **4V**。当 **BOR4V = 0** 时, **VBOR** 被设置为 **2.1V**。

如果 **VDD** 下降到 **VBOR** 以下, 且持续时间超过参数值 (**TBOR**) (见第 17.0 节 “电气特性”), 欠压状况将使器件复位。不管 **VDD** 的变化速率如何, 上述情况都会发

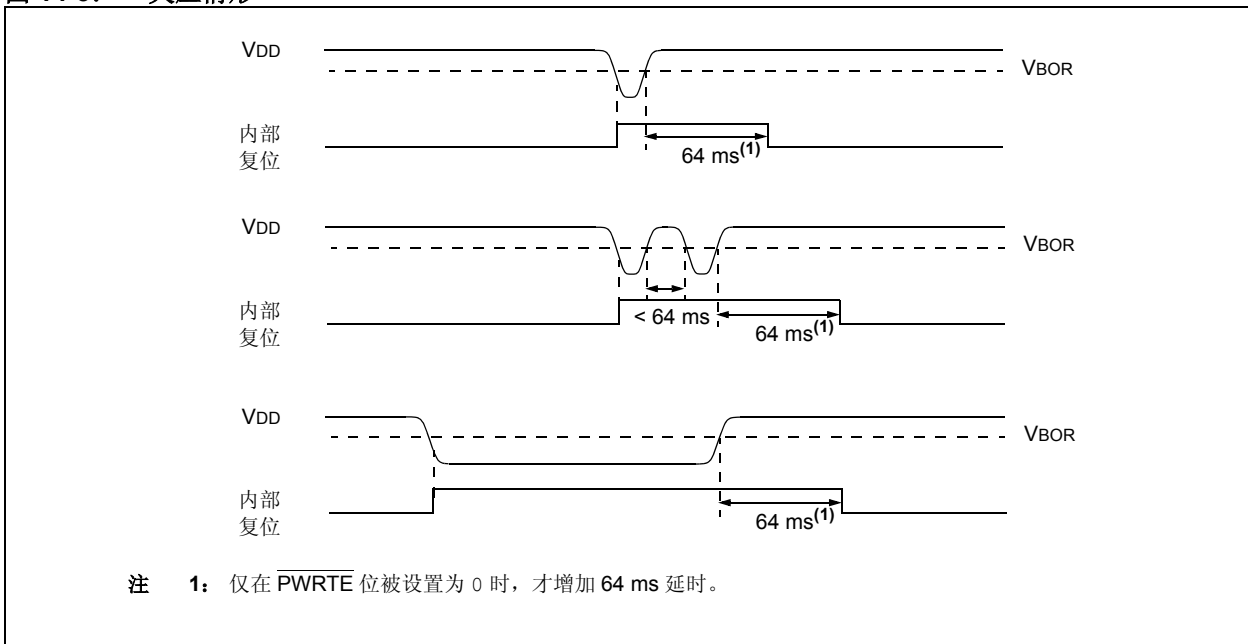
生。如果 **VDD** 低于 **VBOR** 的时间少于参数 (**TBOR**), 则不一定发生复位。

任何复位 (上电复位、欠压复位和看门狗定时器等) 都会使器件保持复位状态, 直到 **VDD** 上升到 **VBOR** 以上 (见图 14-3)。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 **64 ms**。

注: 配置字寄存器 1 中的 **PWRTE** 位用于使能上电延时定时器。

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

图 14-3: 欠压情形



PIC16F882/883/884/886/887

14.2.5 超时时序

上电时的超时时序如下：首先，在 POR 延时结束后启动一段 PWRT 延时，随后在 PWRT 延时结束后振荡器起振。总延时时间取决于振荡器的配置和 PWRTE 位的状态。例如，在 EC 模式且 PWRTE 位置 1（PWRT 禁止）的情况下，根本不会出现延时。图 14-4、图 14-5 和图 14-6 分别给出了各种情形下的超时时序。通过使能双速启动或故障保护时钟监视器，当振荡器起振后，器件将以 INTOSC 作为时钟源来执行代码（见第 4.7.2 节“双速启动时序”和第 4.8 节“故障保护时钟监视器”）。

由于超时是由上电复位脉冲触发的，因此如果 MCLR 保持足够长时间的低电平，所有延时都将结束。将 MCLR 电平拉高后器件将立即开始执行代码（见图 14-5）。这对于测试或同步多个并行工作的 PIC16F882/883/884/886/887 器件来说是非常有用的。

表 14-5 给出了一些特殊寄存器的复位条件，而表 14-4 给出了所有寄存器的复位条件。

14.2.6 电源控制（PCON）寄存器

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

Bit 0 是 $\overline{\text{BOR}}$ （欠压复位）标志位。 $\overline{\text{BOR}}$ 在上电复位时未知。然后，用户必须将该位置 1，并在随后的复位发生时检查 BOR 是否为 0，如果是，则表示已经发生过欠压复位。当禁止欠压复位电路（配置字寄存器中的 BOREN<1:0> = 00）时，BOR 状态位被“忽略”并且不必对其进行预测。

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

更多信息请参见第 3.2.2 节“超低功耗唤醒”和第 14.2.4 节“欠压复位（BOR）”。

表 14-1: 各种情形下的延时

振荡器配置	上电延时		欠压复位延时		从休眠状态唤醒的延时
	$\overline{\text{PWRTE}} = 0$	$\overline{\text{PWRTE}} = 1$	$\overline{\text{PWRTE}} = 0$	$\overline{\text{PWRTE}} = 1$	
XT、HS 和 LP	TPWRT + 1024 · TOSC	1024 · TOSC	TPWRT + 1024 · TOSC	1024 · TOSC	1024 · TOSC
LP, T1OSCIN = 1	TPWRT	—	TPWRT	—	—
RC、EC 和 INTOSC	TPWRT	—	TPWRT	—	—

表 14-2: STATUS/PCON 寄存器中位及其意义

POR	BOR	$\overline{\text{TO}}$	$\overline{\text{PD}}$	状态
0	x	1	1	上电复位
u	0	1	1	欠压复位
u	u	0	u	WDT 复位
u	u	0	0	WDT 唤醒
u	u	u	u	正常工作期间的 $\overline{\text{MCLR}}$ 复位
u	u	1	0	休眠期间的 $\overline{\text{MCLR}}$ 复位

图注: u = 不变, x = 未知

表 14-3: 与欠压有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
PCON	—	—	ULPWUE	SBOREN	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	--01 --qq	--0u --uu
STATUS	IRP	RP1	RPO	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	000q quuu

图注: u = 不变, x = 未知, - = 未实现位 (读为 0), q = 取值视条件而定。欠压复位未使用阴影单元。
注 1: 其他 (非上电) 复位包括正常工作期间的 $\overline{\text{MCLR}}$ 复位和看门狗定时器复位。

PIC16F882/883/884/886/887

图 14-4: 上电时的超时时序 ($\overline{\text{MCLR}}$ 延时): 情形 1

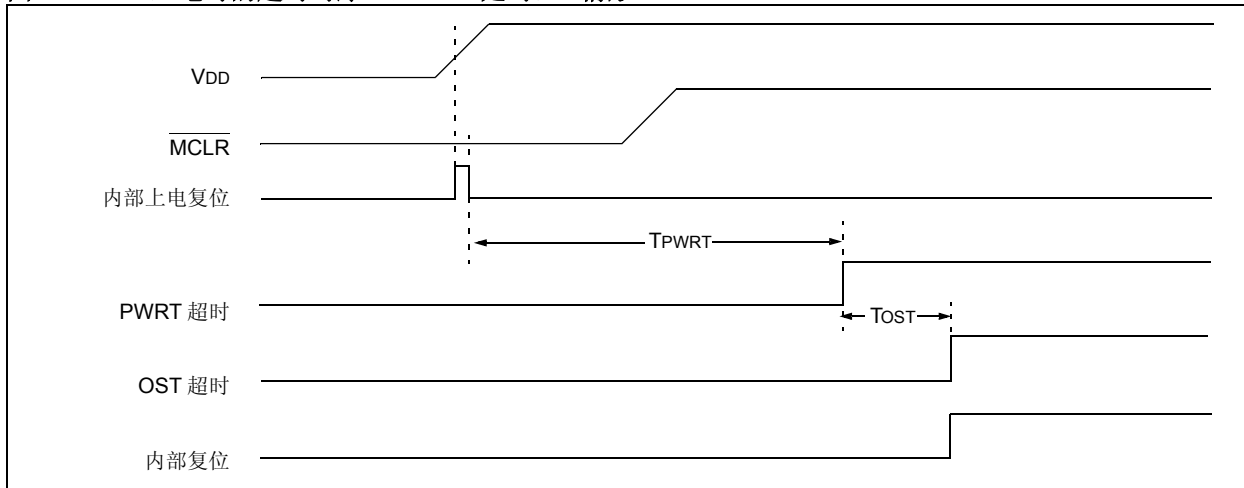


图 14-5: 上电时的超时时序 ($\overline{\text{MCLR}}$ 延时): 情形 2

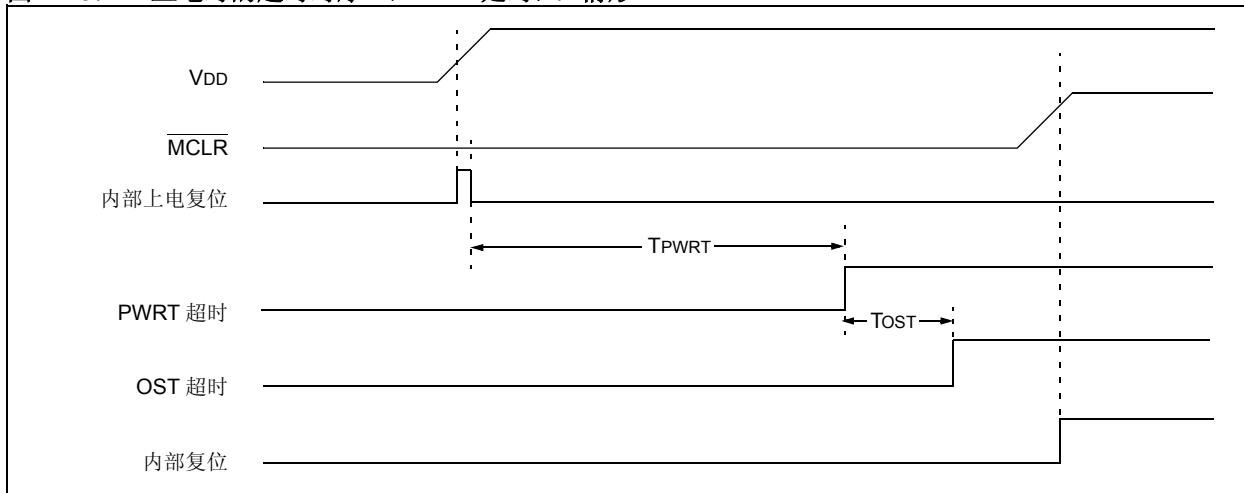
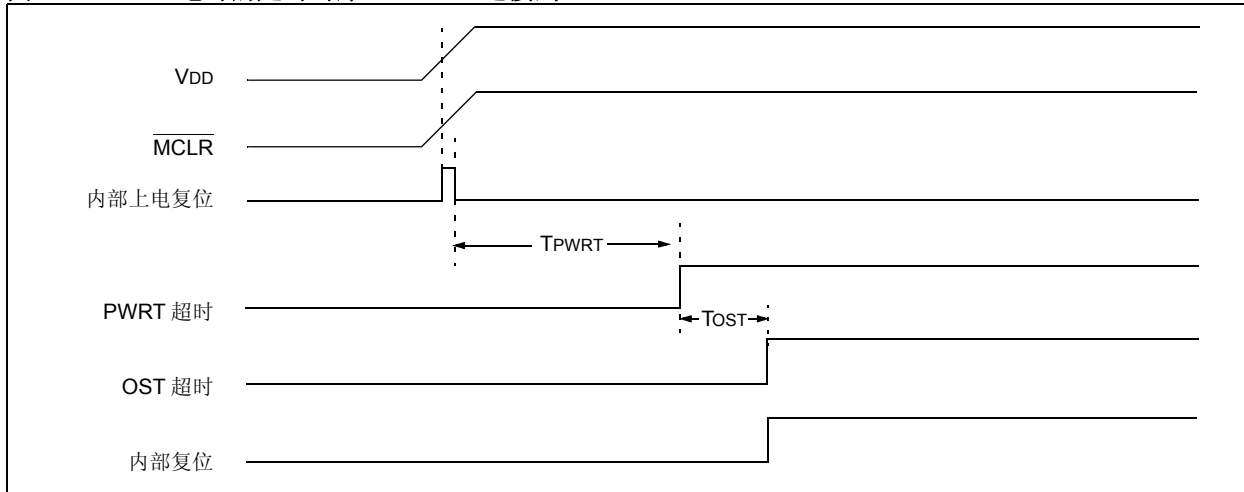


图 14-6: 上电时的延时时序 ($\overline{\text{MCLR}}$ 连接到 VDD)



PIC16F882/883/884/886/887

表 14-4: 各个寄存器的初始状态

寄存器	地址	上电复位	<ul style="list-style-type: none"> • MCLR 复位 • WDT 复位 • 欠压复位⁽¹⁾ 	<ul style="list-style-type: none"> • 通过中断将器件从休眠模式唤醒 • 通过 WDT 超时溢出将器件从休眠模式唤醒
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h/80h/ 100h/180h	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR0	01h/101h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h/82h/ 102h/182h	0000 0000	0000 0000	PC + 1 ⁽³⁾
STATUS	03h/83h/ 103h/183h	0001 1xxx	000q quuu ⁽⁴⁾	uuuq quuu ⁽⁴⁾
FSR	04h/84h/ 104h/184h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	xxxx xxxx	0000 0000	uuuu uuuu
PORTB	06h/106h	xxxx xxxx	0000 0000	uuuu uuuu
PORTC	07h	xxxx xxxx	0000 0000	uuuu uuuu
PORTD	08h	xxxx xxxx	0000 0000	uuuu uuuu
PORTE	09h	---- xxxx	---- 0000	---- uuuu
PCLATH	0Ah/8Ah/ 10Ah/18Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh/8Bh/ 10Bh/18Bh	0000 000x	0000 000u	uuuu uuuu ⁽²⁾
PIR1	0Ch	0000 0000	0000 0000	uuuu uuuu ⁽²⁾
PIR2	0Dh	0000 0000	0000 0000	uuuu uuuu ⁽²⁾
TMR1L	0Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	0Fh	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	10h	0000 0000	uuuu uuuu	-uuu uuuu
TMR2	11h	0000 0000	0000 0000	uuuu uuuu
T2CON	12h	-000 0000	-000 0000	-uuu uuuu
SSPBUF	13h	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	14h	0000 0000	0000 0000	uuuu uuuu
CCPR1L	15h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	16h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	17h	0000 0000	0000 0000	uuuu uuuu
RCSTA	18h	0000 000x	0000 0000	uuuu uuuu
TXREG	19h	0000 0000	0000 0000	uuuu uuuu
RCREG	1Ah	0000 0000	0000 0000	uuuu uuuu
CCPR2L	1Bh	xxxx xxxx	uuuu uuuu	uuuu uuuu

图注: u = 不变, x = 未知, - = 未实现位 (读为 0), q = 取值视条件而定。

- 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 / 或 PIR1 寄存器中的 1 位或多位会受到影响 (引起唤醒)。
- 3: 当器件被中断唤醒且 GIE 位置 1 时, PC 中装入中断向量 (0004h)。
- 4: 关于特定条件下的复位值, 请参见表 14-5。
- 5: 如果复位是由于欠压引起的, 则 bit 0 = 0。其他复位将使 bit 0 = u。
- 6: 只有在 SSPCON 寄存器的 SSPM<3:0> 位 = 1001 时才可能被访问。

PIC16F882/883/884/886/887

表 14-4: 各个寄存器的初始状态 (续)

寄存器	地址	上电复位	<ul style="list-style-type: none"> • $\overline{\text{MCLR}}$ 复位 • WDT 复位 • 欠压复位⁽¹⁾ 	<ul style="list-style-type: none"> • 通过中断将器件从休眠模式唤醒 • 通过 WDT 超时溢出将器件从休眠模式唤醒
CCPR2H	1Ch	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	1Dh	--00 0000	--00 0000	--uu uuuu
ADRESH	1Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	1Fh	00-0 0000	00-0 0000	uu-u uuuu
OPTION_REG	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
TRISC	87h	1111 1111	1111 1111	uuuu uuuu
TRISD	88h	1111 1111	1111 1111	uuuu uuuu
TRISE	89h	---- 1111	---- 1111	---- uuuu
PIE1	8Ch	0000 0000	0000 0000	uuuu uuuu
PIE2	8Dh	0000 0000	0000 0000	uuuu uuuu
PCON	8Eh	--01 --0x	--0u --uu ^(1, 5)	--uu --uu
OSCCON	8Fh	-110 q000	-110 q000	-uuu uuuu
OSCTUNE	90h	---0 0000	---u uuuu	---u uuuu
SSPCON2	91h	0000 0000	0000 0000	uuuu uuuu
PR2	92h	1111 1111	1111 1111	1111 1111
SSPADD ⁽⁶⁾	93h	0000 0000	0000 0000	uuuu uuuu
SSPMSK ⁽⁶⁾	93h	1111 1111	1111 1111	1111 1111
SSPSTAT	94h	0000 0000	0000 0000	uuuu uuuu
WPUB	95h	1111 1111	1111 1111	uuuu uuuu
IOCB	96h	0000 0000	0000 0000	uuuu uuuu
VRCON	97h	0000 0000	0000 0000	uuuu uuuu
TXSTA	98h	0000 -010	0000 -010	uuuu -uuu
SPBRG	99h	0000 0000	0000 0000	uuuu uuuu
SPBRGH	9Ah	0000 0000	0000 0000	uuuu uuuu
PWM1CON	9Bh	0000 0000	0000 0000	uuuu uuuu
ECCPAS	9Ch	0000 0000	0000 0000	uuuu uuuu
PSTRCON	9Dh	---0 0001	---0 0001	---u uuuu
ADRESL	9Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	9Fh	0-00 ----	0-00 ----	u-uu ----
WDTCON	105h	---0 1000	---0 1000	---u uuuu
CM1CON0	107h	0000 0-00	0000 0-00	uuuu u-uu
CM2CON0	108h	0000 0-00	0000 0-00	uuuu u-uu

图注: u = 不变, x = 未知, - = 未实现位 (读为 0), q = 取值视条件而定。

- 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。
 注 2: INTCON 和 / 或 PIR1 寄存器中的 1 位或多位会受到影响 (引起唤醒)。
 注 3: 当器件被中断唤醒且 GIE 位置 1 时, PC 中装入中断向量 (0004h)。
 注 4: 关于特定条件下的复位值, 请参见表 14-5。
 注 5: 如果复位是由于欠压引起的, 则 bit 0 = 0。其他复位将使 bit 0 = u。
 注 6: 只有在 SSPCON 寄存器的 SSPM<3:0> 位 = 1001 时才可能被访问。

PIC16F882/883/884/886/887

表 14-4: 各个寄存器的初始状态 (续)

寄存器	地址	上电复位	<ul style="list-style-type: none"> • $\overline{\text{MCLR}}$ 复位 • WDT 复位 • 欠压复位⁽¹⁾ 	<ul style="list-style-type: none"> • 通过中断将器件从休眠模式唤醒 • 通过 WDT 超时溢出将器件从休眠模式唤醒
CM2CON1	109h	0000 0--0	0000 0--0	uuuu u--u
EEDAT	10Ch	0000 0000	0000 0000	uuuu uuuu
EEADR	10Dh	0000 0000	0000 0000	uuuu uuuu
EEDATH	10Eh	--00 0000	--00 0000	--uu uuuu
EEADRH	10Fh	---0 0000	---0 0000	---u uuuu
SRCON	185h	0000 00-0	0000 00-0	uuuu uu-u
BAUDCTL	187h	01-0 0-00	01-0 0-00	uu-u u-uu
ANSEL	188h	1111 1111	1111 1111	uuuu uuuu
ANSELH	189h	1111 1111	1111 1111	uuuu uuuu
EECON1	18Ch	---- x000	---- q000	---- uuuu
EECON2	18Dh	---- ----	---- ----	---- ----

图注: u = 不变, x = 未知, - = 未实现位 (读为 0), q = 取值视条件而定。

- 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 / 或 PIR1 寄存器中的 1 位或多位会受到影响 (引起唤醒)。
- 3: 当器件被中断唤醒且 GIE 位置 1 时, PC 中装入中断向量 (0004h)。
- 4: 关于特定条件下的复位值, 请参见表 14-5。
- 5: 如果复位是由于欠压引起的, 则 bit 0 = 0。其他复位将使 bit 0 = u。
- 6: 只有在 SSPCON 寄存器的 SSPM<3:0> 位 = 1001 时才可能被访问。

表 14-5: 特殊寄存器的初始状态

条件	程序计数器	STATUS 寄存器	PCON 寄存器
上电复位	000h	0001 1xxx	--01 --0x
正常工作期间的 $\overline{\text{MCLR}}$ 复位	000h	000u uuuu	--0u --uu
休眠期间的 $\overline{\text{MCLR}}$ 复位	000h	0001 0uuu	--0u --uu
WDT 复位	000h	0000 uuuu	--0u --uu
WDT 唤醒	PC + 1	uuu0 0uuu	--uu --uu
欠压复位	000h	0001 1uuu	--01 --u0
通过中断从休眠唤醒	PC + 1 ⁽¹⁾	uuu1 0uuu	--uu --uu

图注: u = 不变, x = 未知, - = 未实现位 (读为 0)。

- 注 1: 如果器件被中断唤醒且全局中断允许位 GIE 置 1, 则执行 PC + 1 后, PC 中装入中断向量 (0004h)。

PIC16F882/883/884/886/887

14.3 中断

PIC16F882/883/884/886/887 器件具有以下多种中断源：

- 外部中断 RB0/INT
- Timer0 溢出中断
- PORTB 电平变化中断
- 两个比较器中断
- A/D 中断
- Timer1 溢出中断
- Timer2 匹配中断
- 数据 EEPROM 写中断
- 故障保护时钟监视器中断
- 增强型 CCP 中断
- EUSART 接收和发送中断
- 超低功耗唤醒中断
- MSSP 中断

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

全局中断允许位 GIE (INTCON<7>) 在置 1 时允许所有未屏蔽的中断，而在清零时，禁止所有中断。可以通过 INTCON、PIE1 和 PIE2 寄存器中相应的允许位来禁止各个中断。复位时 GIE 被清零。

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

INTCON 寄存器包含下列中断标志位：

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

外设中断标志位在特殊功能寄存器 PIR1 和 PIR2 中。相应的中断允许位在特殊功能寄存器 PIE1 和 PIE2 中。

PIR1 寄存器包含下列中断标志位：

- A/D 中断
- EUSART 接收和发送中断
- Timer1 溢出中断
- 同步串行端口 (SSP) 中断
- 增强型 CCP1 中断
- Timer1 溢出中断
- Timer2 匹配中断

PIR2 寄存器包含下列中断标志位：

- 故障保护时钟监视器中断
- 2 个比较器中断
- EEPROM 数据写中断
- 超低功耗唤醒中断
- CCP2 中断

当响应一个中断时：

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

对于外部中断事件，如 INT 引脚中断或 PORTB 电平变化中断，中断响应延时将为 3 到 4 个指令周期。确切的响应延时取决于发生中断事件的时间（见图 14-8）。单周期或双周期指令的中断响应延时完全相同。进入中断服务程序之后，就可以通过查询中断标志位来确定中断源。在重新允许中断前，必须用软件将中断标志位清零，以避免重复响应该中断。

- | |
|---|
| 注 1: 各中断标志位的置 1 不受对应的中断允许位和 GIE 位状态的影响。 |
| 2: 当执行一条清零 GIE 位的指令后，任何一条等待在下一周期执行的中断都将被忽略。当 GIE 位被再次置 1 后，被忽略的中断仍会继续等待处理。 |

欲知更多有关 Timer1、Timer2、比较器、A/D、数据 EEPROM、EUSART、MSSP 或增强型 CCP 模块的信息，请参见相应的外设章节。

14.3.1 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>) 被置 1 时在上升沿触发，而当 INTEDG 位被清零时在下降沿触发。当 RB0/INT 引脚上出现有效边沿时，INTF 位 (INTCON<1>) 置 1。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 1，则 RB0/INT 中断能将处理器从休眠状态唤醒。GIE 位的状态决定处理器在被唤醒之后是否会跳转到中断向量 (0004h) 处执行代码。有关休眠的详细信息，请参见第 14.6 节“掉电模式 (休眠)”；而有关 RB0/INT 中断将处理器从休眠状态唤醒的时序请参见图 14-10。

PIC16F882/883/884/886/887

14.3.2 TMR0 中断

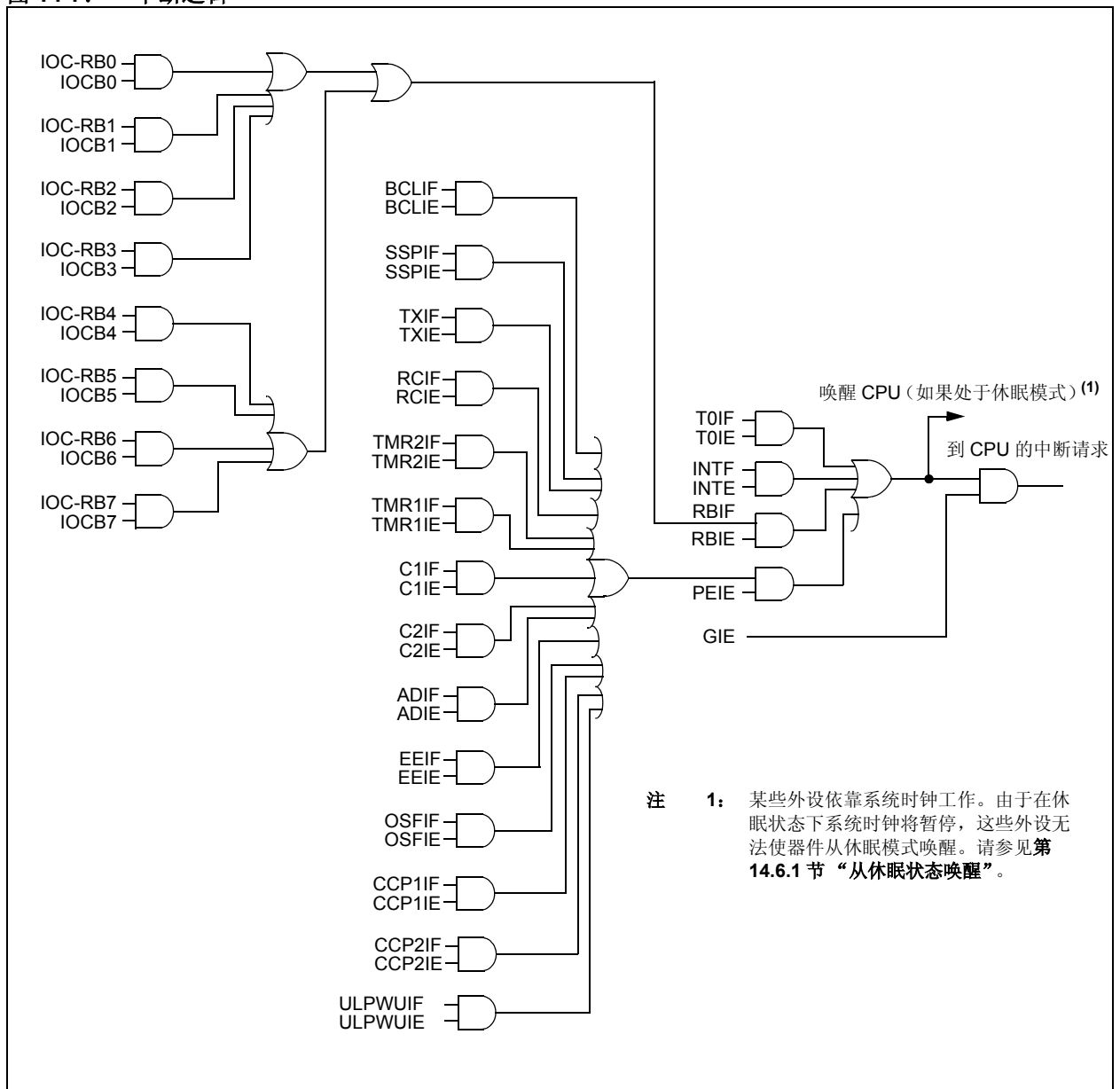
TMR0 寄存器溢出 (FFh → 00h) 会将 T0IF (INTCON<2>) 位置 1。可以通过置 1/ 清零 T0IE (INTCON<5>) 位来允许 / 禁止该中断。欲知有关 Timer0 模块的操作, 请参见第 5.0 节“Timer0 模块”。

14.3.3 PORTB 中断

PORTB 输入电平的变化会使 RBIF (INTCON<0>) 位置 1。可以通过置 1/ 清零 RBIE (INTCON<3>) 位来允许 / 禁止该中断。此外, 可通过 IOCB 寄存器对该端口的各个引脚进行配置。

注: 如果在执行读操作的过程中 (Q2 周期的开始) I/O 引脚的电平发生了改变, 那么 RBIF 中断标志可能就不会置 1。更多信息请参见第 3.4.3 节“电平变化中断”。

图 14-7: 中断逻辑



PIC16F882/883/884/886/887

图 14-8: INT 引脚中断时序

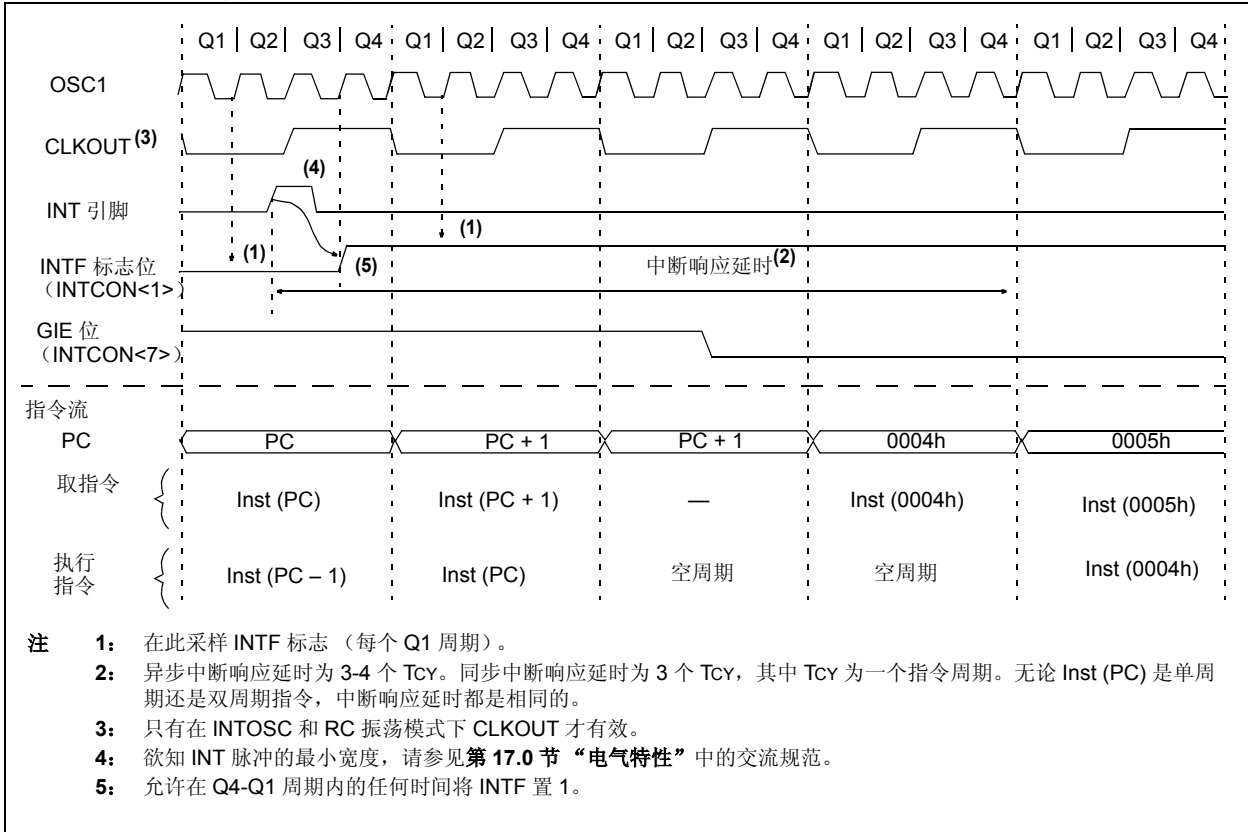


表 14-6: 中断寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	ULPWUIE	—	CCP2IE	0000 00-0	0000 00-0
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	ULPWUIF	—	CCP2IF	0000 00-0	0000 00-0

图注: x = 未知, u = 不变, - = 未实现 (读为 0), q = 取值视条件而定。中断模块不使用阴影单元。

PIC16F882/883/884/886/887

14.4 中断的现场保护

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

由于在 PIC16F882/883/884/886/887 中所有 GPR 存储区的高 16 个字节都是公用的（见图 2-2 和图 2-3），临时保存寄存器 W_TEMP 和 STATUS_TEMP 都应该被放在这里。这 16 个存储单元不需要分区，因此更加便于现场保护和恢复。与例 14-1 中相同的代码可被用于：

- 存储 W 寄存器
- 存储 STATUS 寄存器
- 执行中断服务程序代码
- 恢复 STATUS 寄存器（和存储区选择寄存器）
- 恢复 W 寄存器

注： PIC16F882/883/884/886/887 器件通常不需要保存 PCLATH。但是，如果要在中断服务程序和主函数中使用计算 GOTO，就必须在中断服务程序中保存和恢复 PCLATH。

例 14-1： 将 STATUS 和 W 寄存器保存在 RAM 中

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

PIC16F882/883/884/886/887

14.5 看门狗定时器 (WDT)

WDT 具有以下特性:

- 依靠 LFINTOSC (31 kHz) 工作
- 包含一个 16 位预分频器
- 与 Timer0 共用 8 位预分频器
- 超时周期从 1 ms 至 268 秒
- 可由配置位和软件控制

在表 14-7 中的条件下, WDT 会被清零。

14.5.1 WDT 振荡器

WDT 以 31 kHz 的 LFINTOSC 作为其工作的时基。LFINTOSC 使能与否不会在 OSCCON 寄存器的 LTS 位上有所反映。

在所有复位后, WDTCON 的值都为 “---0 1000”。这相当于 17 ms 的标称时基。

注: 当振荡器起振延时定时器 (OST) 启动时, 由于 OST 需要使用 WDT 脉动计数器来对振荡器延时进行计数, 因此 WDT 仍将保持复位状态。当 OST 计数结束后, WDT 将开始计数 (如果使能)。

14.5.2 WDT 控制

WDTE 位位于配置字寄存器 1 中。当该位置 1 时, WDT 持续运行。

当配置字寄存器 1 中的 WDTE 位置 1 时, WDTCON 寄存器的 SWDTEN 位无效。当 WDTE 清零时, 则可使用 SWDTEN 位使能和禁止 WDT。SWDTEN 位置 1 使能 WDT, SWDTEN 位清零则禁止 WDT。

OPTION_REG 寄存器的 PSA 和 PS<2:0> 位具有与早期 PIC16F882/883/884/886/887 系列单片机中相应位同样的功能。更多信息, 请参见第 5.0 节“Timer0 模块”。

图 14-9: 看门狗定时器框图

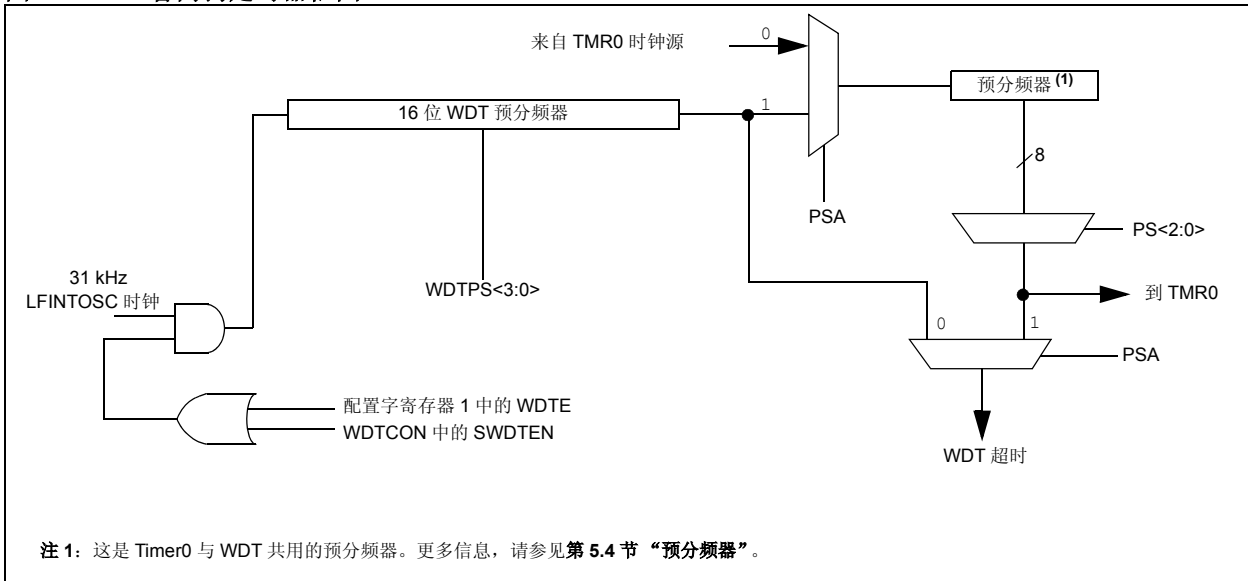


表 14-7: WDT 状态

条件	WDT
WDTE = 0	清零
CLRWDT 指令	
检测到振荡器故障	
退出休眠 + 系统时钟 = T1OSC、EXTRC、INTOSC 或 EXTCLK	
退出休眠 + 系统时钟 = XT、HS 或 LP	清零直到 OST 结束

PIC16F882/883/884/886/887

寄存器 14-3: WDTCON: 看门狗定时器控制寄存器

U-0	U-0	U-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0
—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN ⁽¹⁾
bit 7							bit 0

图注:

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

- bit 7-5 **未实现:** 读为 0
- bit 4-1 **WDTPS<3:0>:** 看门狗定时器周期选择位
 位值 = 预分频比
 0000 = 1:32
 0001 = 1:64
 0010 = 1:128
 0011 = 1:256
 0100 = 1:512 (复位值)
 0101 = 1:1024
 0110 = 1:2048
 0111 = 1:4096
 1000 = 1:8192
 1001 = 1:16384
 1010 = 1:32768
 1011 = 1:65536
 1100 = 保留
 1101 = 保留
 1110 = 保留
 1111 = 保留
- bit 0 **SWDTEN:** 软件使能或禁止看门狗定时器位 ⁽¹⁾
 1 = 使能 WDT
 0 = 禁止 WDT (复位值)

注 1: 如果 WDTE 配置位 = 1, 则 WDT 始终被使能, 而与该控制位的状态无关。如果 WDTE 配置位 = 0, 则可以使用该控制位使能或禁止 WDT。

表 14-8: 看门狗定时器寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
CONFIG1 ⁽¹⁾	CPD	CP	MCLRE	PWRTE	WDTE	FOSC2	FOSC1	FOSC0	—	—
OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 1000	---0 1000

图注: 看门狗定时器不使用阴影单元。
注 1: 关于配置字寄存器 1 中的所有位的操作, 请参见寄存器 14-1。

PIC16F882/883/884/886/887

14.6 掉电模式（休眠）

执行 SLEEP 指令可进入掉电模式。

如果使能了看门狗定时器：

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

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，没有外部电路从 I/O 引脚消耗电流，同时应禁止比较器和 CVREF。为了避免输入引脚悬空而引入开关电流，应在外部将高阻输入的 I/O 引脚拉为高电平或低电平。为了将电流消耗降至最低，TOCKI 输入也应该保持为 VDD 或 VSS。还应考虑 PORTA 片上上拉的影响。

MCLR 引脚必须处于逻辑高电平。

注： 应该注意到 WDT 超时溢出导致的复位并不会将 MCLR 引脚驱动为低电平。

14.6.1 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. MCLR 引脚上的外部复位输入。
2. 看门狗定时器唤醒（如果 WDT 使能）
3. RB0/INT 引脚中断、PORTB 电平变化中断或外设中断。

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

下列外设中断可以将器件从休眠状态唤醒：

1. TMR1 中断。Timer1 必须用作异步计数器。
2. ECCP 捕捉模式中断。
3. A/D 转换（当 A/D 时钟源为 FRC 振荡器时）。
4. EEPROM 写操作完成。
5. 比较器输出状态变化。
6. 电平变化中断。
7. 来自 INT 引脚的外部中断
8. EUSART 间隔检测，I²C 从动模式。

由于在休眠状态期间没有片内时钟处于工作状态，因此其他外设不能产生中断。

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

注： 如果禁止了全局中断（GIE 被清零），但若有任一中断源的中断允许位和相应的中断标志位同时置 1，器件将立即从休眠状态唤醒。SLEEP 指令被完整执行。

器件从休眠状态唤醒时，WDT 都将被清零，而与唤醒的原因无关。

14.6.2 使用中断唤醒

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

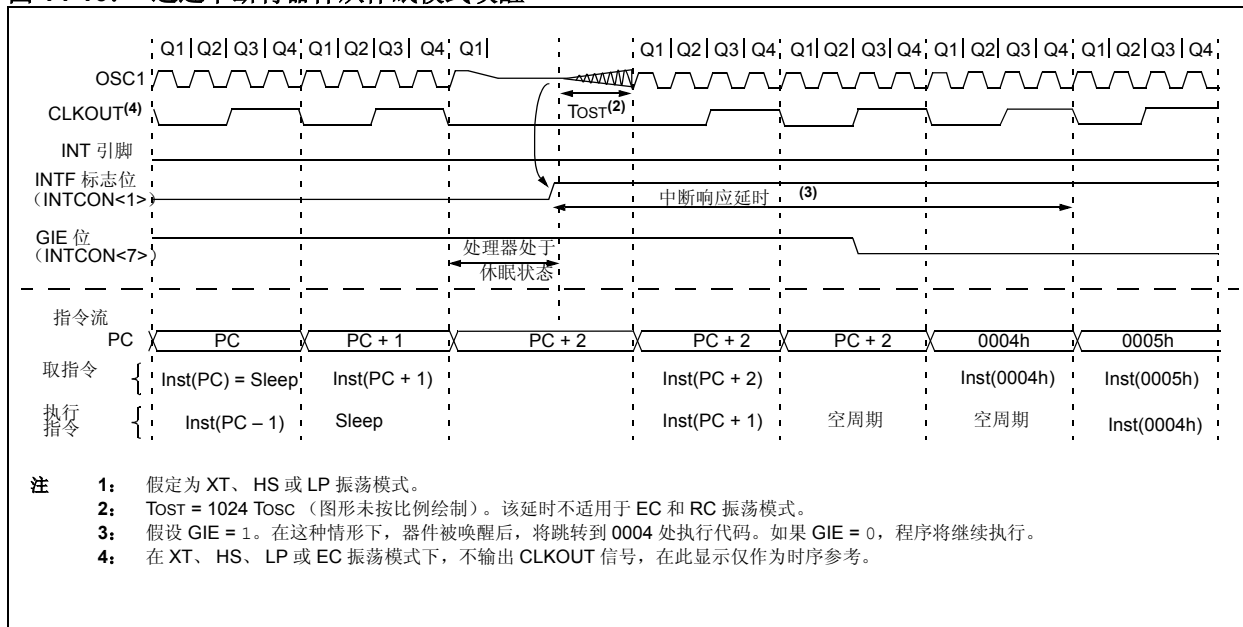
- 如果在执行 SLEEP 指令之前产生了中断，那么 SLEEP 指令将被作为一条 NOP 指令执行。因此，WDT 及其预分频器和后分频器（如果使能）将不会被清零，并且 \overline{TO} 位将不会被置 1，同时 \overline{PD} 也不会被清零。
- 如果在执行 SLEEP 指令期间或之后产生了中断，那么器件将被立即从休眠模式唤醒。SLEEP 指令将在唤醒之前执行完毕。因此，WDT 及其预分频器和后分频器（如果使能）将被清零，并且 \overline{TO} 位将被置 1，同时 \overline{PD} 也将被清零。

即使在执行 SLEEP 指令之前检查到标志位为 0，它也可能在 SLEEP 指令执行完毕之前被置 1。要确定是否执行了 SLEEP 指令，可以测试 \overline{PD} 位。如果 \overline{PD} 位置 1，则说明 SLEEP 指令被作为一条 NOP 指令执行了。

在执行 SLEEP 指令之前，必须先执行一条 CLRWDT 指令，来确保将 WDT 清零。

PIC16F882/883/884/886/887

图 14-10: 通过中断将器件从休眠模式唤醒



14.7 代码保护

如果代码保护位未编程, 校验时可通过使用 ICSP™ 读出片上程序存储器。

注: 当代码保护从开到关时, 将擦除整个数据 EEPROM 和闪存程序存储器的内容。请参见“PIC16F88X Memory Programming Specification”(DS41287) 了解更多信息。

14.8 ID 地址单元

有 4 个存储单元 (2000h-2003h) 被指定为 ID 地址单元, 供用户存储校验和其他代码标识号。在正常执行过程中不能访问这些单元, 但可在编程 / 校验模式中对它们进行读写。只可使用 ID 地址单元的低 7 位。

14.9 在线串行编程

可在最终应用电路中对 PIC16F882/883/884/886/887 单片机进行串行编程。编程可以简单地通过一根时钟线、一根数据线和以下三种其他的线完成:

- 电源线
- 接地线
- 编程电压线

这使用户可使用未编程的器件制造电路板, 而仅在产品交付前才对单片机进行编程。从而可以将最新版本的固件或者定制固件烧写到单片机中。

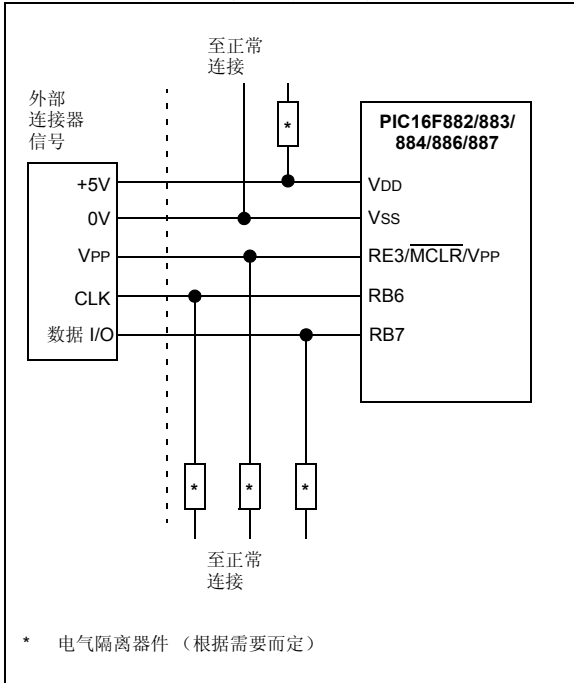
通过将 RB6/ICSPCLK 和 RB7/ICSPDAT 引脚拉至并保持低电平, 并同时 MCLR (VPP) 引脚从 V_{IL} 升到 V_{IH} , 可将器件置于编程 / 校验模式。请参见“PIC16F88X Memory Programming Specification”(DS41287) 了解更多信息。此时 RB7 成为编程数据线而 RB6/ 则成为编程时钟线。在该模式下, RB7 和 RB6 均为施密特触发输入方式。

复位后, 为将器件置于编程 / 校验模式, 程序计数器 (PC) 指向地址单元 00h。随后向器件发送一条 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读取一个 14 位的程序数据。关于串行编程的完整细节, 请参见“PIC16F88X Memory Programming Specification”(DS41287)。

图 14-11 给出了典型的在线串行编程的连接方式。

PIC16F882/883/884/886/887

图 14-11: 典型的在线串行编程连接方式



14.10 在线调试器

PIC16F882/883/884/886/887-ICD 可用于任何封装类型的器件。器件将被固定在目标应用板上，有 3 或 4 根导线连接到 ICD 工具。

当配置字中的调试位 (CONFIG<13>) 被编程为 0 时，将使能在线调试功能。该功能允许使用 MPLAB® ICD 2 实现简单的调试功能。当单片机使能了此功能后，某些资源将不再具有常规功能。详情请参见表 14-9。

注： 用户应用必须具有支持 ICD 功能所需的电路。一旦使能了 ICD 电路，RB6/ICSPCLK 和 RB7/ICSPDAT 引脚的常规器件引脚功能将无法使用。ICD 电路使用这些引脚与 ICD 2 外部调试器通信。

更多信息可参见 Microchip 网站 (www.microchip.com) 上的 “Using MPLAB® ICD 2” (DS51265)。

14.10.1 ICD 引脚排列

PIC16F88X 系列器件具有片上在线调试器电路，以及用于在线调试器的引脚。这使 ICD 器件不必采用另外的管芯或封装。ICD 器件的引脚排列与器件相同 (见第 1.0 节 “器件概述” 以获取完整的引脚及其排列形式的说明)。表 14-9 给出了 28 和 40 引脚器件中与 ICD 相关的引脚的位置和功能。

表 14-9: PIC16F882/883/884/886/887-ICD 引脚说明

引脚 (PDIP)		名称	类型	上拉	说明
PIC16F884/887	PIC16F882/883/886				
40	28	ICDDATA	TTL	—	在线调试器双向数据引脚
39	27	ICDCLK	ST	—	在线调试器双向时钟引脚
1	1	MCLR/VPP	HV	—	编程电压
11, 32	20	VDD	P	—	
12, 31	8, 19	VSS	P	—	

图注: TTL = TTL 输入缓冲器, ST = 施密特触发器输入缓冲器, P = 电源, HV = 高电压

PIC16F882/883/884/886/887

表 15-2: PIC16F882/883/884/886/887 指令集

助记符, 操作数	说明	周期	14 位操作码				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 作逻辑与运算	1	00	0101	dfff	ffff	Z	1, 2
CLRF	f	将 f 清零	1	00	0001	1fff	ffff	Z	2
CLRWF	—	将 W 寄存器清零	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	f 取反	1	00	1001	dfff	ffff	Z	1, 2
DECWF	f, d	f 递减 1	1	00	0011	dfff	ffff	Z	1, 2
DECFSZ	f, d	f 递减 1, 为 0 则跳过	1(2)	00	1011	dfff	ffff		1, 2, 3
INCF	f, d	f 递增 1	1	00	1010	dfff	ffff	Z	1, 2
INCFSZ	f, d	f 递增 1, 为 0 则跳过	1(2)	00	1111	dfff	ffff		1, 2, 3
IORWF	f, d	W 和 f 作逻辑或运算	1	00	0100	dfff	ffff	Z	1, 2
MOVF	f, d	将 f 的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	对 f 执行带进位的左移	1	00	1101	dfff	ffff	C	1, 2
RRF	f, d	对 f 执行带进位的右移	1	00	1100	dfff	ffff	C	1, 2
SUBWF	f, d	f 减去 W	1	00	0010	dfff	ffff	C, DC, Z	1, 2
SWAPF	f, d	将 f 中的两个半字节交换	1	00	1110	dfff	ffff		1, 2
XORWF	f, d	W 和 f 作逻辑异或运算	1	00	0110	dfff	ffff	Z	1, 2
针对位的文件寄存器操作									
BCF	f, b	将 f 中的某位清零	1	01	00bb	bfff	ffff		1, 2
BSF	f, b	将 f 中的某位置 1	1	01	01bb	bfff	ffff		1, 2
BTFSC	f, b	检测 f 中的某位, 为 0 则跳过	1(2)	01	10bb	bfff	ffff		3
BTFSS	f, b	检测 f 中的某位, 为 1 则跳过	1(2)	01	11bb	bfff	ffff		3
立即数和控制操作									
ADDLW	k	立即数和 W 相加	1	11	111x	kkkk	kkkk	C, DC, Z	
ANDLW	k	立即数与 W 作逻辑与运算	1	11	1001	kkkk	kkkk	Z	
CALL	k	调用子程序	2	10	0kkk	kkkk	kkkk		
CLRWDAT	—	清零看门狗定时器	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$	
GOTO	k	跳转到地址	2	10	1kkk	kkkk	kkkk		
IORLW	k	立即数与 W 作逻辑或运算	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	将立即数传送到 W	1	11	00xx	kkkk	kkkk		
RETFIE	—	从中断返回	2	00	0000	0000	1001		
RETLW	k	返回并将立即数传送到 W	2	11	01xx	kkkk	kkkk		
RETURN	—	从子程序返回	2	00	0000	0000	1000		
SLEEP	—	进入待机模式	1	00	0000	0110	0011	$\overline{TO}, \overline{PD}$	
SUBLW	k	从立即数中减去 W 的内容	1	11	110x	kkkk	kkkk	C, DC, Z	
XORLW	k	立即数与 W 作逻辑异或运算	1	11	1010	kkkk	kkkk	Z	

- 注 1: 当 I/O 寄存器用自身内容修改自身时 (例如: MOVF PORTB, 1), 使用的值是出现在引脚上的值。例如, 如果将一引脚配置为输入, 虽然其对应数据锁存器中的值为 1, 但此时若有外部器件将该引脚驱动为低电平, 则被写回的数据值将是 0。
- 2: 如果对 TMR0 寄存器执行这条指令 (并且适用时 d = 1), 预分频器分配给 Timer0 模块时将被清零。
- 3: 如果程序计数器 (PC) 被修改或条件测试为真, 则执行该指令需要两个周期。第二个周期执行一条 NOP 指令。

PIC16F882/883/884/886/887

15.2 指令说明

ADDLW 立即数与 W 相加

语法: [标号] 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 和 Z
说明: 将 W 寄存器的内容与 8 位立即数 k 相加, 结果存入 W 寄存器。

ADDWF W 和 f 相加

语法: [标号] ADDWF f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(W) + (f) \rightarrow (\text{目标寄存器})$
受影响的状态位: C、DC 和 Z
说明: 将 W 寄存器的内容与 f 寄存器的内容相加。如果 d 等于 0, 结果存放在 W 寄存器中。如果 d 为 1, 结果存回寄存器 f。

ANDLW 立即数与 W 作逻辑与运算

语法: [标号] ANDLW k
操作数: $0 \leq k \leq 255$
操作: $(W) .AND.(k) \rightarrow (W)$
受影响的状态位: Z
说明: 将 W 寄存器的内容与 8 位立即数 k 作逻辑与运算。结果存入 W 寄存器。

ANDWF W 和 f 作逻辑与运算

语法: [标号] ANDWF f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(W) .AND.(f) \rightarrow (\text{目标寄存器})$
受影响的状态位: Z
说明: W 寄存器与 f 寄存器作逻辑与运算。如果 d 等于 0, 结果存放在 W 寄存器中。如果 d 为 1, 结果存回寄存器 f。

BCF 将 f 中的某位清零

语法: [标号] BCF f,b
操作数: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
操作: $0 \rightarrow (f)$
受影响的状态位: 无
说明: 将寄存器 f 中的 b 位清零。

BSF 将 f 中的某位置 1

语法: [标号] BSF f,b
操作数: $0 \leq f \leq 127$
 $0 \leq b \leq 7$
操作: $1 \rightarrow (f)$
受影响的状态位: 无
说明: 将寄存器 f 的 b 位置 1。

BTFSC 检测位, 为 0 则跳过

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

PIC16F882/883/884/886/887

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

CALL	调用子程序
语法:	[标号] 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
说明:	调用子程序。首先，将返回地址 (PC+1) 压入堆栈。11 位立即数地址被装入 PC 位 <10:0>。将 PCLATH 装入 PC 的高位。CALL 是双周期指令。

CLRF	将 f 清零
语法:	[标号] CLRF f
操作数:	$0 \leq f \leq 127$
操作:	00h → (f) 1 → Z
受影响的状态位:	Z
说明:	寄存器 f 的内容被清零，Z 位置 1。

CLRW	将 W 寄存器清零
语法:	[标号] CLRW
操作数:	无
操作:	00h → (W) 1 → Z
受影响的状态位:	Z
说明:	W 寄存器被清零。全零标志位 (Z) 置 1。

CLRWDT	清零看门狗定时器
语法:	[标号] CLRWDT
操作数:	无
操作:	00h → WDT 0 → WDT 预分频器, 1 → \overline{TO} 1 → \overline{PD}
受影响的状态位:	\overline{TO} 和 \overline{PD}
说明:	CLRWDT 指令用于复位看门狗定时器。它还将复位 WDT 的预分频器。状态位 \overline{TO} 和 \overline{PD} 置 1。

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

DECF	f 递减 1
语法:	[标号] DECF f,d
操作数:	$0 \leq f \leq 127$ d ∈ [0,1]
操作:	(f) - 1 → (目标寄存器)
受影响的状态位:	Z
说明:	将寄存器 f 的内容减 1。如果 d 为 0，结果存储到 W 寄存器。如果 d 为 1，结果存回寄存器 f。

PIC16F882/883/884/886/887

DECFSZ **f 递减 1, 为 0 则跳过**

语法: [标号] DECFSZ f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(f) - 1 \rightarrow$ (目标寄存器), 结果为 0 时跳过
受影响的状态位: 无
说明: 将寄存器 f 的内容递减 1。如果 d 为 0, 结果存放在 W 寄存器中。如果 d 为 1, 结果存回寄存器 f。如果结果为 1, 则执行下一条指令。如果结果为 0, 则执行 NOP, 从而使该指令变为双周期指令。

GOTO **无条件跳转**

语法: [标号] GOTO k
操作数: $0 \leq k \leq 2047$
操作: $k \rightarrow PC<10:0>$
 $PCLATH<4:3> \rightarrow PC<12:11>$
受影响的状态位: 无
说明: GOTO 是无条件跳转指令。11 位立即数地址被装入 PC 位 $<10:0>$ 。PC 高位从 $PCLATH<4:3>$ 装入。GOTO 是双周期指令。

INCF **f 递增 1**

语法: [标号] INCF f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(f)+1 \rightarrow$ (目标寄存器)
受影响的状态位: Z
说明: 将寄存器 f 的内容递增 1。如果 d 为 0, 结果存放在 W 寄存器中。如果 d 为 1, 结果存回寄存器 f。

INCFSZ **f 递增 1, 为 0 则跳过**

语法: [标号] INCFSZ f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(f) + 1 \rightarrow$ (目标寄存器), 结果为 0 时跳过
受影响的状态位: 无
说明: 将寄存器 f 的内容递增 1。如果 d 为 0, 结果存放在 W 寄存器中。如果 d 为 1, 结果存回寄存器 f。如果结果为 1, 则执行下一条指令。如果结果为 0, 则执行 NOP, 从而使该指令变为双周期指令。

IORLW **立即数与 W 作逻辑或运算**

语法: [标号] IORLW k
操作数: $0 \leq k \leq 255$
操作: $(W) .OR. k \rightarrow (W)$
受影响的状态位: Z
说明: 将 W 寄存器的内容与 8 位立即数 k 作逻辑或运算。结果存入 W 寄存器。

IORWF **将 W 和 f 作逻辑或运算**

语法: [标号] IORWF f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(W) .OR. (f) \rightarrow$ (目标寄存器)
受影响的状态位: Z
说明: W 寄存器与 f 寄存器作逻辑或运算。如果 d 为 0, 结果存放在 W 寄存器中。如果 d 为 1, 结果存回寄存器 f。

PIC16F882/883/884/886/887

MOVF 将 f 的内容传送到目标寄存器

语法: [标号] MOVF f,d

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

操作: (f) → (目标寄存器)

受影响的状态位: Z

机器码:

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

说明: 根据 d 的状态, 将寄存器 f 的内容送入目标寄存器。如果 d 为 0, 目标寄存器为 W 寄存器。如果 d 为 1, 目标寄存器为寄存器 f。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。

指令字数: 1

指令周期数: 1

示例: MOVF FSR, 0

MOVLW 将立即数传送到 W

语法: [标号] 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
----	------	------	------

说明: 将 8 位立即数 k 装入 W 寄存器。“无关位”被汇编为 0。

指令字数: 1

指令周期数: 1

示例: MOVLW 0x5A

执行指令后
W = 0x5A

MOVWF 将 W 的内容传送到 f

语法: [标号] 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:

00	0000	1fff	ffff
----	------	------	------

说明: 将 W 寄存器中的数据传送到寄存器 f。

指令字数: 1

指令周期数: 1

示例: MOVWF OPTION

NOP 空操作

语法: [标号] NOP

操作数: 无

操作: 空操作

受影响的状态位: 无

机器码:

00	0000	0xx0	0000
----	------	------	------

说明: 无任何操作。

指令字数: 1

指令周期数: 1

示例: NOP

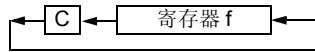
PIC16F882/883/884/886/887

RLF 对 f 执行带进位的循环左移

语法: [标号] RLF f,d
 操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
 操作: 参见下面的说明
 受影响的状态位: C
 机器码:

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

 说明: 将寄存器 f 的内容连同进位标志位一起左移 1 位。如果 d 为 0, 结果存放在 W 寄存器中。如果 d 为 1, 结果存回寄存器 f。

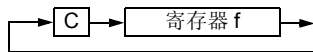


指令字数: 1
 指令周期数: 1
 示例: RLF REG1,0

执行指令前
 REG1 = 1110 0110
 C = 0
 执行指令后
 REG1 = 1110 0110
 W = 1100 1100
 C = 1

RRF 对 f 执行带进位的循环右移

语法: [标号] RRF f,d
 操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
 操作: 参见下面的说明
 受影响的状态位: C
 说明: 将寄存器 f 的内容连同进位标志位一起右移 1 位。如果 d 为 0, 结果存放在 W 寄存器中。如果 d 为 1, 结果存回寄存器 f。



SLEEP 进入休眠模式

语法: [标号] SLEEP
 操作数: 无
 操作: 00h → WDT,
 0 → WDT 预分频器,
 1 → \overline{TO} ,
 0 → \overline{PD}
 受影响的状态位: \overline{TO} 和 \overline{PD}
 说明: 掉电状态位 (\overline{PD}) 清零。超时状态位 (\overline{TO}) 位置 1。看门狗定时器及其预分频器被清零。振荡器停振, 处理器进入休眠模式。

SUBLW 从立即数中减去 W 寄存器的内容

语法: [标号] 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 和 Z
 说明: 从 8 位立即数 k 中减去 W 寄存器的内容 (采用二进制补码方法运算)。结果存入 W 寄存器。

C = 0	$W > k$
C = 1	$W \leq k$
DC = 0	$W\langle 3:0 \rangle > k\langle 3:0 \rangle$
DC = 1	$W\langle 3:0 \rangle \leq k\langle 3:0 \rangle$

PIC16F882/883/884/886/887

SUBWF **f 减去 W**

语法: [标号] SUBWF f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(f) - (W) \rightarrow$ (目标寄存器)
受影响的状态位: C、DC 和 Z
说明: 从寄存器 f 中减去 W 寄存器的内容 (采用二进制补码方法运算)。如果 d 为 0, 结果存储到 W 寄存器。如果 d 为 1, 结果存回寄存器 f。

C = 0	$W > f$
C = 1	$W \leq f$
DC = 0	$W\langle 3:0 \rangle > f\langle 3:0 \rangle$
DC = 1	$W\langle 3:0 \rangle \leq f\langle 3:0 \rangle$

SWAPF **将 f 中的两个半字节交换**

语法: [标号] SWAPF f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(f\langle 3:0 \rangle) \rightarrow$ (目标寄存器 $\langle 7:4 \rangle$),
 $(f\langle 7:4 \rangle) \rightarrow$ (目标寄存器 $\langle 3:0 \rangle$)
受影响的状态位: 无
说明: 将寄存器 f 的高半字节和低半字节交换。如果 d 为 0, 结果存放在 W 寄存器中。如果 d 为 1, 结果存回寄存器 f。

XORLW **立即数与 W 作逻辑异或运算**

语法: [标号] XORLW k
操作数: $0 \leq k \leq 255$
操作: $(W).XOR.k \rightarrow (W)$
受影响的状态位: Z
说明: 将 W 寄存器的内容与 8 位立即数 k 作逻辑异或运算。结果存入 W 寄存器。

XORWF **W 和 f 作逻辑异或运算**

语法: [标号] XORWF f,d
操作数: $0 \leq f \leq 127$
 $d \in [0,1]$
操作: $(W).XOR.(f) \rightarrow$ (目标寄存器)
受影响的状态位: Z
说明: 将 W 寄存器的内容与 f 寄存器的内容作逻辑异或运算。如果 d 等于 0, 结果存放在 W 寄存器中。如果 d 为 1, 结果存回寄存器 f。

PIC16F882/883/884/886/887

注:

PIC16F883/884/886/887

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
 - PICKit™ 2 开发编程器
- 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。

PIC16F883/884/886/887

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 系列单片机及 dsPIC30F、dsPIC33 和 PIC24 系列数字信号控制器。这些编译器可提供其他编译器并不具备的强大的集成功能和出众的代码优化能力，且使用方便。

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

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，或与其他可重定位目标文件和存档链接以生成可执行文件。该汇编器有如下显著特征：

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

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、大部分的外设及内部寄存器的状况。

MPLAB SIM 软件模拟器完全支持使用 MPLAB C18 和 MPLAB C30 C 编译器以及 MPASM 和 MPLAB ASM30 汇编器的符号调试。该软件模拟器可用于在硬件实验室环境外灵活地开发和调试代码，是一款完美且经济的软件开发工具。

PIC16F883/884/886/887

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。

PIC16F883/884/886/887

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 PICKit 2 开发编程器

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

16.13 演示、开发和评估板

有许多演示、开发和评估板可用于各种 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、 Σ - Δ ADC、流速传感器，等等。

有关演示、开发和评估工具包的完整列表，请查阅 Microchip 公司网页 (www.microchip.com) 以及最新的“*Product Selector Guide (产品选型指南)*” (DS00148)。

PIC16F882/883/884/886/887

17.0 电气特性

绝对最大额定值 (†)

偏置时的环境温度	-40°C 至 +125°C
储存温度	-65°C 至 +150°C
VDD 相对于 VSS 的电压	-0.3V 至 +6.5V
MCLR 相对于 VSS 的电压	-0.3V 至 +13.5V
所有其他引脚相对于 VSS 的电压	-0.3V 至 (VDD + 0.3V)
总功耗 (1)	800 mW
VSS 引脚的最大输出电流	300 mA
VDD 引脚的最大输入电流	250 mA
输入钳位电流 I _{IK} (V _I < 0 或 V _I > VDD)	±20 mA
输出钳位电流 I _{OK} (V _O < 0 或 V _O > VDD)	±20 mA
任一 I/O 引脚的最大输出灌电流	25 mA
任一 I/O 引脚的最大输出拉电流	25 mA
PORTA、PORTB 和 PORTE (组合) 的最大灌电流 (2)	200 mA
PORTA、PORTB 和 PORTE (组合) 的最大拉电流 (2)	200 mA
PORTC 和 PORTD (组合) 的最大灌电流 (2)	200 mA
PORTC 和 PORTD (组合) 的最大拉电流 (2)	200 mA

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

2: PIC16F886/PIC16F887 器件上没有 PORTD 和 PORTE。

† 注意: 如果器件的工作条件超过“绝对最大额定值”, 就可能会对器件造成永久性损坏。上述值仅为运行条件极大值, 我们建议不要使器件在该规范规定的范围以外运行。器件长时间工作在最大值条件下, 其稳定性会受到影响。

PIC16F882/883/884/886/887

图 17-1: PIC16F882/883/884/886/887 电压—频率关系图 ($-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$)

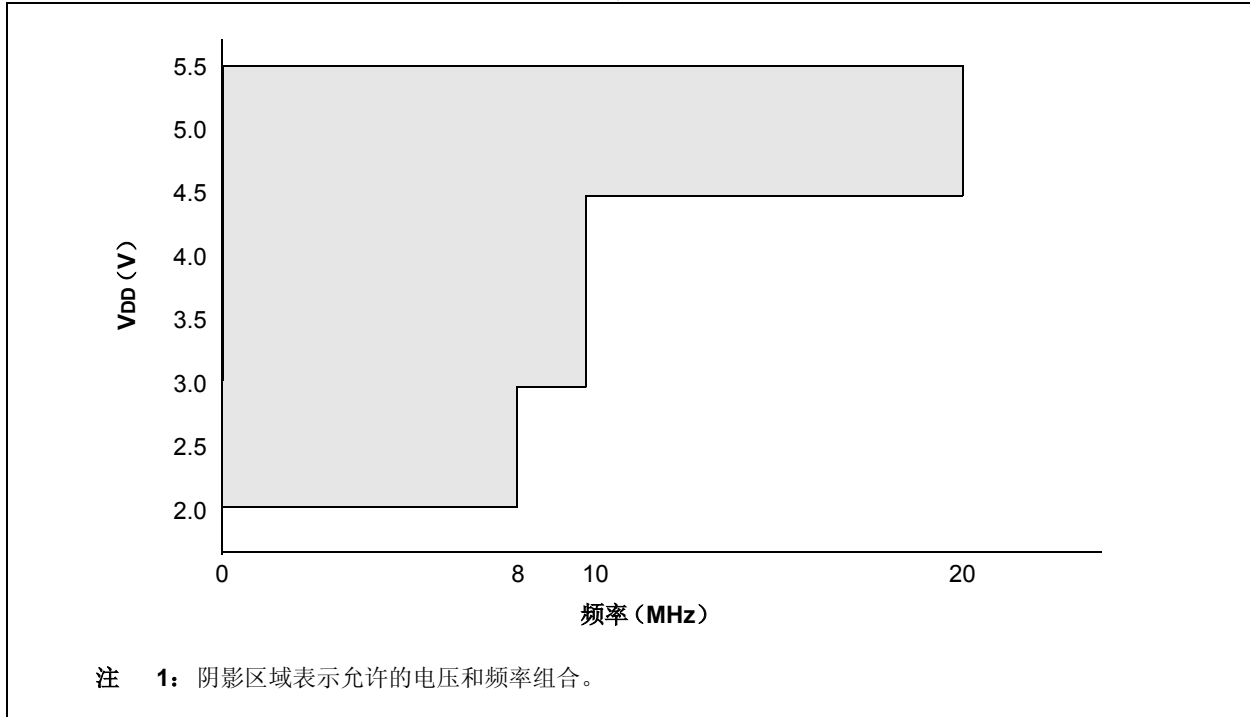
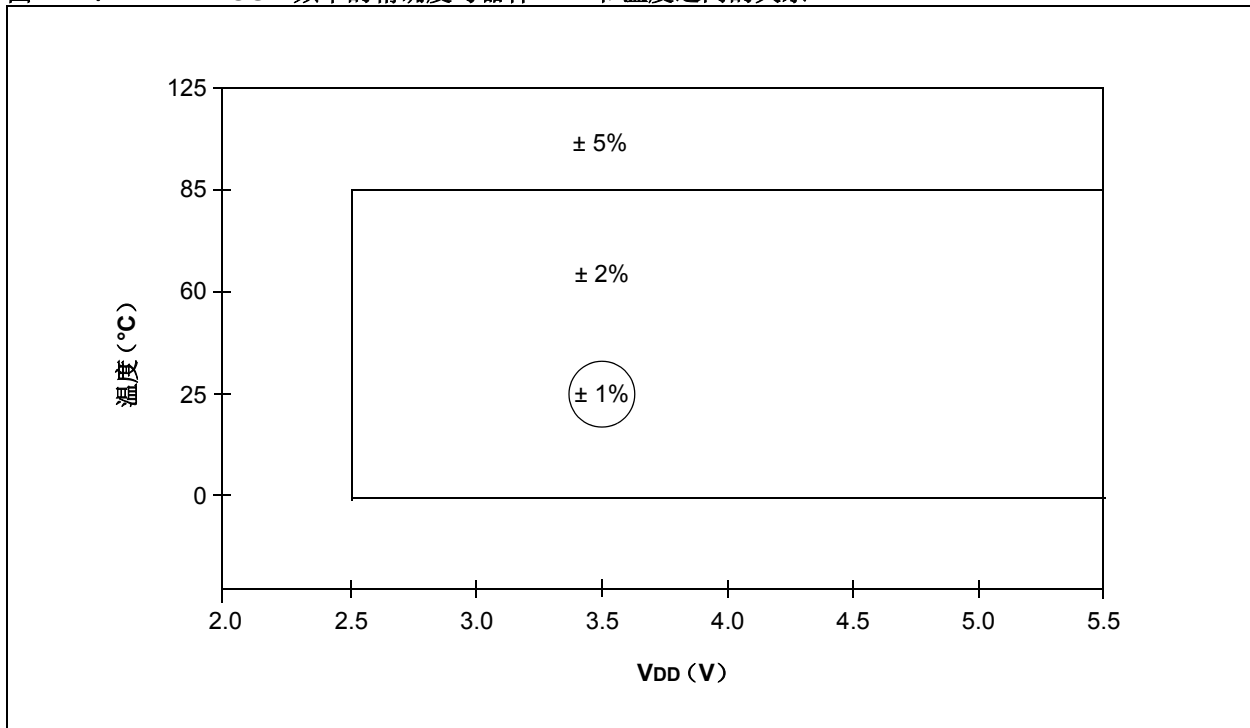


图 17-2: HFINTOSC 频率的精确度与器件 VDD 和温度之间的关系



PIC16F882/883/884/886/887

17.1 直流特性：PIC16F882/883/884/886/887-I（工业级）

PIC16F882/883/884/886/887-E（扩展级）

直流特性			标准工作条件（除非另外说明）				
			工作温度				
			-40°C ≤ TA ≤ +85°C（工业级）				
			-40°C ≤ TA ≤ +125°C（扩展级）				
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
D001 D001C D001D	VDD	供电电压	2.0 2.0 3.0 4.5	— — — —	5.5 5.5 5.5 5.5	V V V V	Fosc ≤ 8 MHz: HFINTOSC 和 EC Fosc ≤ 4 MHz Fosc ≤ 10 MHz Fosc ≤ 20 MHz
D002*	VDR	RAM 数据保持电压 ⁽¹⁾	1.5	—	—	V	器件处于休眠模式
D003	VPOR	确保能够产生内部上电复位信号的 VDD 起始电压	—	VSS	—	V	详细信息请参见第 14.2.1 节“上电复位 (POR)”。
D004*	SVDD	确保能够产生内部上电复位信号的 VDD 上升速率	0.05	—	—	V/ms	详细信息请参见第 14.2.1 节“上电复位 (POR)”。

* 这些参数仅为特征值，未经测试。

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

注 1：这是在不丢失 RAM 数据的前提下 VDD 在休眠模式下的最小值。

PIC16F882/883/884/886/887

17.2 直流特性：PIC16F882/883/884/886/887-I（工业级）

PIC16F882/883/884/886/887-E（扩展级）

直流特性		标准工作条件（除非另外说明） 工作温度					条件	
参数编号	器件特性	最小值	典型值†	最大值	单位	VDD	注	
D010	供电电流 (IDD) (1, 2)	—	11	23	μA	2.0	FOSC = 32 kHz LP 振荡模式	
		—	18	38	μA	3.0		
		—	35	75	μA	5.0		
D011*		—	140	250	μA	2.0	FOSC = 1 MHz XT 振荡模式	
		—	220	400	μA	3.0		
		—	380	650	μA	5.0		
D012		—	260	380	μA	2.0	FOSC = 4 MHz XT 振荡模式	
		—	420	670	μA	3.0		
		—	0.8	1.4	mA	5.0		
D013*		—	130	220	μA	2.0	FOSC = 1 MHz EC 振荡模式	
		—	215	360	μA	3.0		
		—	360	520	μA	5.0		
D014		—	220	340	μA	2.0	FOSC = 4 MHz EC 振荡模式	
		—	375	550	μA	3.0		
		—	0.65	1.0	mA	5.0		
D015		—	8	20	μA	2.0	FOSC = 31 kHz LFINTOSC 模式	
		—	16	40	μA	3.0		
		—	31	65	μA	5.0		
D016*		—	340	400	μA	2.0	FOSC = 4 MHz HFINTOSC 模式	
		—	500	650	μA	3.0		
		—	0.8	1.2	mA	5.0		
D017		—	410	0.7	μA	2.0	FOSC = 8 MHz HFINTOSC 模式	
		—	700	1	μA	3.0		
		—	1.30	1.8	mA	5.0		
D018		—	230	580	μA	2.0	FOSC = 4 MHz EXTRC 模式 (3)	
		—	400	950	μA	3.0		
		—	0.63	1.6	mA	5.0		
D019		—	2.6	3.7	mA	4.5	FOSC = 20 MHz HS 振荡器模式	
		—	2.8	3.8	mA	5.0		

* 这些参数仅为特征值，未经测试。

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

- 注 1: 在正常工作模式下，所有 IDD 测量的测试条件为：OSC1 = 外部方波，满幅；所有 I/O 引脚均为三态，拉至 VDD；MCLR = VDD；禁止 WDT。
- 2: 供电电流主要由工作电压和频率决定。其他因素，如 I/O 引脚负载和开关频率、振荡器类型、内部代码执行模式和温度也会影响电流消耗。
- 3: 配置为 RC 振荡模式时，该电流不包括流经 REXT 的电流。流经该电阻的电流可由公式 $I_R = V_{DD}/2R_{EXT}$ (mA) 来估算，其中 REXT 的单位是 kΩ。

PIC16F882/883/884/886/887

17.3 直流特性：PIC16F882/883/884/886/887-I（工业级）

直流特性		标准工作条件（除非另外说明） 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ （工业级）					条件	
参数编号	器件特性	最小值	典型值 †	最大值	单位	VDD		
						注		
D020	基本掉电电流 (IPD) ⁽²⁾	—	0.05	1.2	μA	2.0	禁止 WDT、BOR、比较器、VREF 和 T1OSC	
		—	0.15	1.5	μA	3.0		
		—	0.35	1.8	μA	5.0		
		—	150	500	nA	3.0		$-40^{\circ}\text{C} \leq T_A \leq +25^{\circ}\text{C}$
D021		—	1.0	2.2	μA	2.0	WDT 电流 ⁽¹⁾	
		—	2.0	4.0	μA	3.0		
		—	3.0	7.0	μA	5.0		
D022		—	42	60	μA	3.0	BOR 电流 ⁽¹⁾	
		—	85	122	μA	5.0		
D023		—	32	45	μA	2.0	比较器电流 ⁽¹⁾ 。同时使能两个比较器	
		—	60	78	μA	3.0		
		—	120	160	μA	5.0		
D024		—	30	36	μA	2.0	CVREF 电流 ⁽¹⁾ （高范围）	
		—	45	55	μA	3.0		
		—	75	95	μA	5.0		
D025*		—	39	47	μA	2.0	CVREF 电流 ⁽¹⁾ （低范围）	
		—	59	72	μA	3.0		
		—	98	124	μA	5.0		
D026		—	4.5	7.0	μA	2.0	T1OSC 电流 ⁽¹⁾ ， 32.768 kHz	
		—	5.0	8.0	μA	3.0		
		—	6.0	12	μA	5.0		
D027		—	0.30	1.6	μA	3.0	A/D 电流 ⁽¹⁾ ， 不在进行转换	
		—	0.36	1.9	μA	5.0		

* 这些参数仅为特征值，未经测试。

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

- 注 1: 外设电流是基本 IDD 或 IPD 电流与使能该外设时额外消耗的电流的和。外设的 Δ 电流可通过将该参数的值减去基本 IDD 或 IPD 电流确定。当计算总电流消耗时应该使用最大值。
- 2: 振荡器类型不决定休眠模式下的掉电电流。掉电电流是在器件休眠时，所有 I/O 引脚处于高阻态并且连接到 VDD 时测得的。

PIC16F882/883/884/886/887

17.4 直流特性: PIC16F882/883/884/886/887-E (扩展级)

直流特性		标准工作条件 (除非另外说明) 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
D020E	基本掉电电流 (IPD) (2)	—	0.05	9	μA	2.0	禁止 WDT、BOR、比较器、VREF 和 T1OSC
		—	0.15	11	μA	3.0	
		—	0.35	15	μA	5.0	
D021E		—	1	28	μA	2.0	WDT 电流 (1)
		—	2	30	μA	3.0	
		—	3	35	μA	5.0	
D022E		—	42	65	μA	3.0	BOR 电流 (1)
		—	85	127	μA	5.0	
D023E		—	32	45	μA	2.0	比较器电流 (1)。同时使能两个比较器
		—	60	78	μA	3.0	
		—	120	160	μA	5.0	
D024E		—	30	70	μA	2.0	CVREF 电流 (1) (高范围)
		—	45	90	μA	3.0	
		—	75	120	μA	5.0	
D025E*		—	39	91	μA	2.0	CVREF 电流 (1) (低范围)
		—	59	117	μA	3.0	
		—	98	156	μA	5.0	
D026E		—	4.5	25	μA	2.0	T1OSC 电流 (1), 32.768 kHz
		—	5	30	μA	3.0	
		—	6	40	μA	5.0	
D027E		—	0.30	12	μA	3.0	A/D 电流 (1), 不在进行转换
		—	0.36	16	μA	5.0	

* 这些参数仅为特征值, 未经测试。

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

- 注 1: 外设电流是基本 IDD 或 IPD 电流与使能该外设时额外消耗的电流的和。外设的 Δ 电流可通过将该参数的值减去基本 IDD 或 IPD 电流确定。当计算总电流消耗时应该使用最大值。
- 2: 振荡器类型不决定休眠模式下的掉电电流。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并且连接到 VDD 时测得的。

PIC16F882/883/884/886/887

17.5 直流特性:

PIC16F882/883/884/886/887-I (工业级)
PIC16F882/883/884/886/887-E (扩展级)

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

* 这些参数仅为特征值, 未经测试。

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

- 注
- 1: 在 RC 振荡器配置中, OSC1/CLKIN 引脚为施密特触发器输入。在 RC 模式中建议不要使用外部时钟。
 - 2: 负电流定义为自引脚输出的电流。
 - 3: MCLR 引脚上的泄漏电流主要取决于施加的电平。规定电平为正常工作条件下的电平。在不同的输入电压下可测得更高的泄漏电流。
 - 4: 更多信息请参见第 10.3.1 节“使用数据 EEPROM”。
 - 5: 包括 CLKOUT 模式下的 OSC2 引脚。

PIC16F882/883/884/886/887

17.5 直流特性:

PIC16F882/883/884/886/887-I (工业级)

PIC16F882/883/884/886/887-E (扩展级) (续)

直流特性			标准工作条件 (除非另外说明)				
			工作温度				
			-40°C ≤ TA ≤ +85°C (工业级)				
			-40°C ≤ TA ≤ +125°C (扩展级)				
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
D100	IULP	超低功耗唤醒电流	—	200	—	nA	请参见应用笔记 AN879 “Using the Microchip Ultra Low-Power Wake-up Module” (DS00879)
D101*	COSC2	输出引脚上的容性负载规范 OSC2 引脚	—	—	15	pF	使用外部时钟驱动 OSC1 并处于 XT、HS 和 LP 模式
D101A*	CIO	所有 I/O 引脚	—	—	50	pF	
数据 EEPROM 存储器							
D120	Ed	字节耐用性	100K	1M	—	E/W	-40°C ≤ TA ≤ +85°C
D120A	Ed	字节耐用性	10K	100K	—	E/W	+85°C ≤ TA ≤ +125°C
D121	VDRW	用于读写操作的 VDD	VMIN	—	5.5	V	使用 EECON1 用于读 / 写 VMIN = 最小工作电压
D122	TDEW	擦 / 写周期时间	—	5	6	ms	
D123	TRETD	数据保存时间	40	—	—	年	假设没有违反其他规范
D124	TREF	更新前的总擦 / 写周期数 ⁽⁴⁾	1M	10M	—	E/W	-40°C ≤ TA ≤ +85°C
闪存程序存储器							
D130	EP	单元耐用性	10K	100K	—	E/W	-40°C ≤ TA ≤ +85°C
D130A	Ed	单元耐用性	1K	10K	—	E/W	+85°C ≤ TA ≤ +125°C
D131	VPR	用于读操作的 VDD	VMIN	—	5.5	V	VMIN = 最小工作电压
D132	VPEW	用语擦 / 写操作的 VDD	4.5	—	5.5	V	
D133	TPEW	擦 / 写周期时间	—	2	2.5	ms	
D134	TRETD	数据保存时间	40	—	—	年	假设没有违反其他规范

* 这些参数仅为特征值, 未经测试。

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

- 注
- 1: 在 RC 振荡器配置中, OSC1/CLKIN 引脚为施密特触发器输入。在 RC 模式中建议不要使用外部时钟。
 - 2: 负电流定义为自引脚输出的电流。
 - 3: MCLR 引脚上的泄漏电流主要取决于施加的电平。规定电平为正常工作条件下的电平。在不同的输入电压下可测得更高的泄漏电流。
 - 4: 更多信息请参见第 10.3.1 节“使用数据 EEPROM”。
 - 5: 包括 CLKOUT 模式下的 OSC2 引脚。

PIC16F882/883/884/886/887

17.6 散热考虑

标准工作条件（除非另外说明） 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$					
参数编号	符号	特性	典型值	单位	条件
TH01	θ_{JA}	结点与环境之间的热阻	47.2	$^{\circ}\text{C}/\text{W}$	40 引脚 PDIP 封装
			24.4	$^{\circ}\text{C}/\text{W}$	44 引脚 QFN 封装
			45.8	$^{\circ}\text{C}/\text{W}$	44 引脚 TQFP 封装
			60.2	$^{\circ}\text{C}/\text{W}$	28 引脚 PDIP 封装
			80.2	$^{\circ}\text{C}/\text{W}$	28 引脚 SOIC 封装
			89.4	$^{\circ}\text{C}/\text{W}$	28 引脚 SSOP 封装
			29	$^{\circ}\text{C}/\text{W}$	28 引脚 QFN 封装
TH02	θ_{JC}	结点与封装外壳之间的热阻	24.7	$^{\circ}\text{C}/\text{W}$	40 引脚 PDIP 封装
			TBD	$^{\circ}\text{C}/\text{W}$	44 引脚 QFN 封装
			14.5	$^{\circ}\text{C}/\text{W}$	44 引脚 TQFP 封装
			29	$^{\circ}\text{C}/\text{W}$	28 引脚 PDIP 封装
			23.8	$^{\circ}\text{C}/\text{W}$	28 引脚 SOIC 封装
			23.9	$^{\circ}\text{C}/\text{W}$	28 引脚 SSOP 封装
			TBD	$^{\circ}\text{C}/\text{W}$	28 引脚 QFN 封装
TH03	T_J	结温	150	$^{\circ}\text{C}$	用于计算降低了的额定功率
TH04	PD	功耗	—	W	$PD = P_{INTERNAL} + P_{I/O}$
TH05	$P_{INTERNAL}$	内部功耗	—	W	$P_{INTERNAL} = I_{DD} \times V_{DD}$ (注 1)
TH06	$P_{I/O}$	I/O 功耗	—	W	$P_{I/O} = \sum (I_{OL} \times V_{OL}) + \sum (I_{OH} \times (V_{DD} - V_{OH}))$
TH07	P_{DER}	降低了的额定功率	—	W	$P_{DER} = (T_J - T_A) / \theta_{JA}$ (注 2 和 3)

图注: TBD = 待定。

- 注 1: I_{DD} 为单独运行芯片而不驱动输出引脚上的任何负载的电流。
 注 2: T_A = 环境温度。
 注 3: 所允许的最大功耗是绝对最大总功耗或降低了的额定功率 (P_{DER}) 中较小的值。

PIC16F882/883/884/886/887

17.7 时序参数符号

可根据以下任一格式来创建时序参数符号：

1. TppS2ppS
2. TppS

T		
F	频率	T
		时间

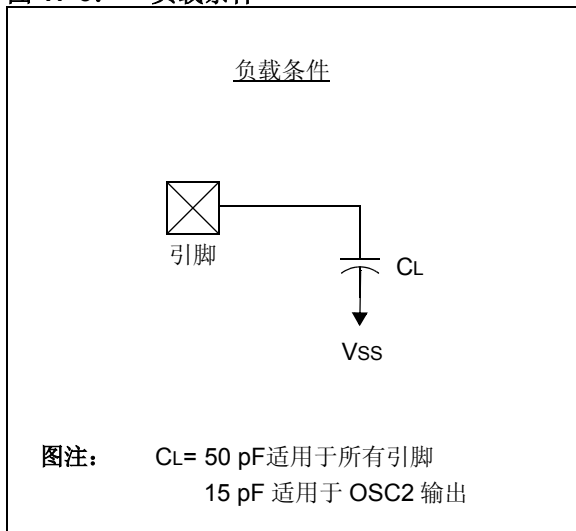
小写字母 (pp) 及其含意：

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

大写字母及其含意：

S		
F	下降	P
H	高	R
I	无效 (高阻态)	V
L	低	Z
		周期
		上升
		有效
		高阻态

图 17-3: 负载条件



PIC16F882/883/884/886/887

17.8 交流特性: PIC16F882/883/884/886/887 (工业级和扩展级)

图 17-4: 时钟时序

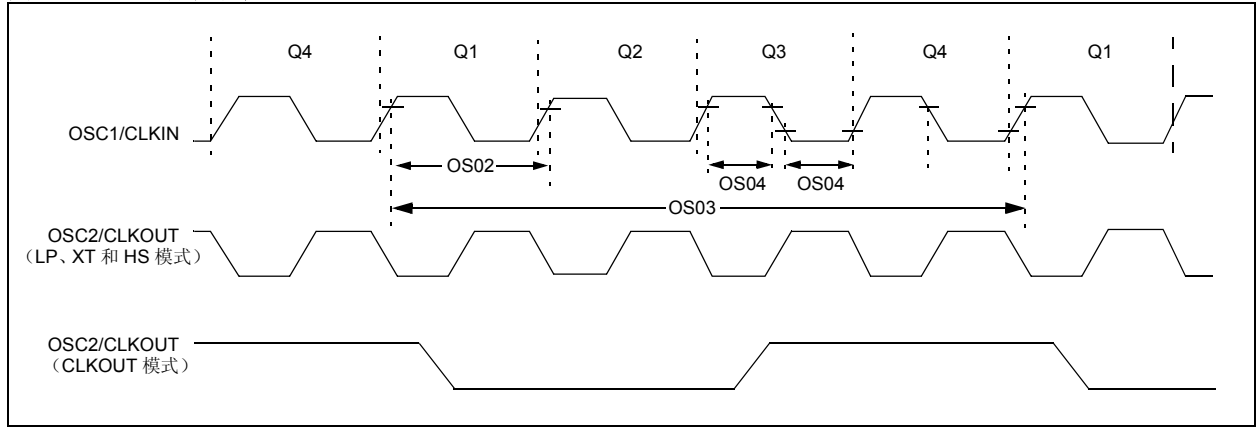


表 17-1: 时钟振荡器时序要求

标准工作条件 (除非另外说明)							
工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
OS01	Fosc	外部 CLKIN 频率 ⁽¹⁾	DC	—	37	kHz	LP 振荡模式
			DC	—	4	MHz	XT 振荡模式
			DC	—	20	MHz	HS 振荡模式
			DC	—	20	MHz	EC 振荡模式
		振荡频率 ⁽¹⁾	—	32.768	—	kHz	LP 振荡模式
			0.1	—	4	MHz	XT 振荡模式
			1	—	20	MHz	HS 振荡模式
			DC	—	4	MHz	RC 振荡模式
OS02	Tosc	外部 CLKIN 周期 ⁽¹⁾	27	—	•	μs	LP 振荡模式
			250	—	•	ns	XT 振荡模式
			50	—	•	ns	HS 振荡模式
			50	—	•	ns	EC 振荡模式
		振荡周期 ⁽¹⁾	—	30.5	—	μs	LP 振荡模式
			250	—	10,000	ns	XT 振荡模式
			50	—	1,000	ns	HS 振荡模式
			250	—	—	ns	RC 振荡模式
OS03	Tcy	指令周期 ⁽¹⁾	200	Tcy	DC	ns	Tcy = 4/Fosc
OS04*	TosH, TosL	外部 CLKIN 高电平, 外部 CLKIN 低电平	2	—	—	μs	LP 振荡模式
			100	—	—	ns	XT 振荡模式
			20	—	—	ns	HS 振荡模式
OS05*	TosR, TosF	外部 CLKIN 上升, 外部 CLKIN 下降	0	—	•	ns	LP 振荡器
			0	—	•	ns	XT 振荡器
			0	—	•	ns	HS 振荡器

* 这些参数仅为特征值, 未经测试。

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

注 1: 指令周期时间 (Tcy) 等于输入振荡器时基周期的 4 倍。所有规范值均基于器件在标准工作条件下执行代码所对应的特定振荡器类型的特征数据。超过这些规范值可能导致振荡器运行不稳定和 / 或电流消耗超出预期。所有器件在测试“最小值”时都在 OSC1 引脚连接了外部时钟。当使用了外部时钟输入时, 所有器件的“最大值”周期时限为“DC”(没有时钟)。

PIC16F882/883/884/886/887

表 17-2: 振荡器参数

标准工作条件 (除非另外说明)								
工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
参数编号	符号	特性	频率公差	最小值	典型值 †	最大值	单位	条件
OS06	TWARM	运行时的内部振荡器切换 ⁽³⁾	—	—	—	2	Tosc	最慢时钟
OS07	Tsc	故障保护采样时钟周期 ⁽¹⁾	—	—	21	—	ms	LFINTOSC/64
OS08	HFosc	内部校准的 HFINTOSC 频率 ⁽²⁾	±1%	7.92	8.0	8.08	MHz	V _{DD} = 3.5V, 25°C 2.5V ≤ V _{DD} ≤ 5.5V, 0°C ≤ T _A ≤ +85°C 2.0V ≤ V _{DD} ≤ 5.5V, -40°C ≤ T _A ≤ +85°C (工业级) -40°C ≤ T _A ≤ +125°C (扩展级)
			±2%	7.84	8.0	8.16	MHz	
			±5%	7.60	8.0	8.40	MHz	
OS09*	LFosc	内部校准的 LFINTOSC 频率	—	15	31	45	kHz	
OS10*	TIOSCST	HFINTOSC 振荡器从休眠模式唤醒的起振时间	—	5.5	12	24	μs	V _{DD} = 2.0V, -40°C 至 +85°C
			—	3.5	7	14	μs	V _{DD} = 3.0V, -40°C 至 +85°C
			—	3	6	11	μs	V _{DD} = 5.0V, -40°C 至 +85°C

* 这些参数仅为特征值, 未经测试。

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

- 注 1: 指令周期时间 (T_{cy}) 等于输入振荡器时钟周期的 4 倍。所有规范值均基于器件在标准工作条件下执行代码所对应的特定振荡器类型的特征数据。超过这些规范值可能导致振荡器运行不稳定和 / 或电流消耗超出预期。所有器件在测试“最小值”时都在 OSC1 引脚连接了外部时钟。当使用了外部时钟输入时, 所有器件的“最大值”周期时限为“DC” (没有时钟)。
- 2: 要确保参数值位于这些振荡频率公差范围内, 必须在 V_{DD} 和 V_{SS} 与器件之间在尽可能靠近器件的地方连接耦合电容。建议使用值为 0.1 μF 和 0.01 μF 的并联电容。
- 3: 设计值。

PIC16F882/883/884/886/887

图 17-5: CLKOUT 和 I/O 时序

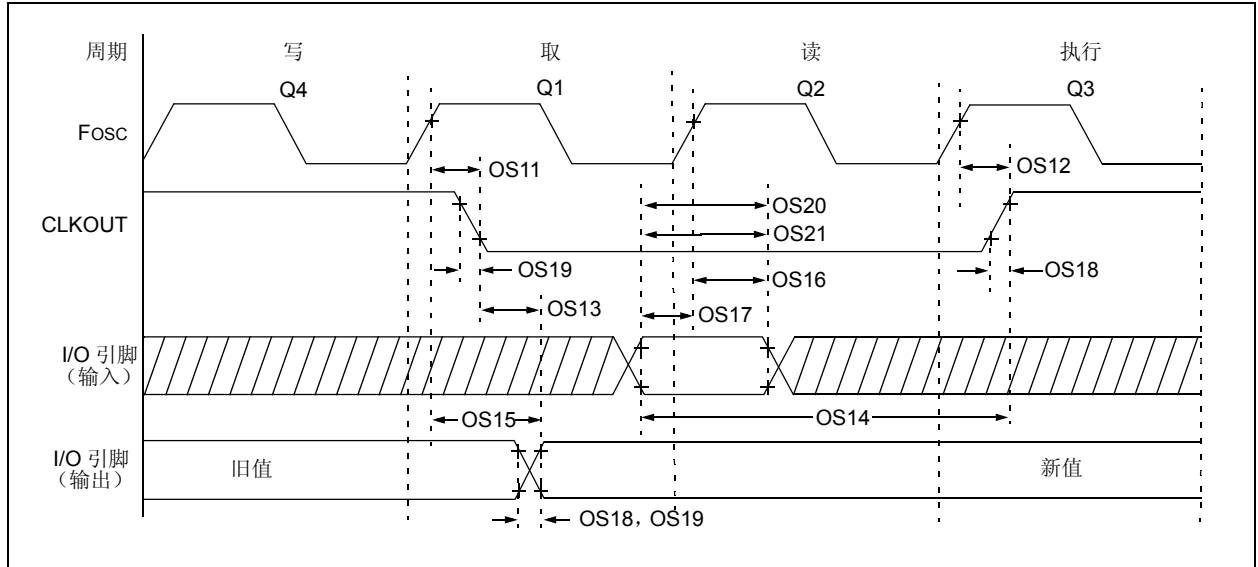


表 17-3: CLKOUT 和 I/O 时序参数

标准工作条件（除非另外说明）							
工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
OS11	TosH2ckL	Fosc↑ 到 CLKOUT↓ 的时间 (1)	—	—	70	ns	VDD = 5.0V
OS12	TosH2ckH	Fosc↑ 到 CLKOUT↑ 的时间 (1)	—	—	72	ns	VDD = 5.0V
OS13	TckL2ioV	CLKOUT↓ 到端口输出有效的的时间 (1)	—	—	20	ns	
OS14	TioV2ckH	在出现 CLKOUT↑ 之前端口输入有效的的时间 (1)	Tosc + 200 ns	—	—	ns	
OS15*	TosH2ioV	Fosc↑ (Q1 周期) 至端口输出有效的的时间	—	50	70	ns	VDD = 5.0V
OS16	TosH2ioI	Fosc↑ (Q2 周期) 至端口输入无效的时间 (I/O 输入保持时间)	50	—	—	ns	VDD = 5.0V
OS17	TioV2osH	端口输入有效至出现 Fosc↑ (Q2 周期) 的时间 (I/O 输入建立时间)	20	—	—	ns	
OS18	TioR	端口输出上升时间 (2)	—	15 40	72 32	ns	VDD = 2.0V VDD = 5.0V
OS19	TioF	端口输出下降时间 (2)	—	28 15	55 30	ns	VDD = 2.0V VDD = 5.0V
OS20*	TINP	INT 引脚高电平或低电平时间	25	—	—	ns	
OS21*	TRAP	PORTA 引脚电平变化中断信号的新输入电平时间	Tcy	—	—	ns	

* 这些参数仅为特征值，未经测试。

† 除非另外说明，否则“典型值”栏中的数据均为 5V、25°C 条件下的值。

注 1: 测量是在 RC 模式下进行的，其中 CLKO 输出为 $4 \times T_{osc}$ 。

2: 包含 CLKOUT 模式下的 OSC2。

PIC16F882/883/884/886/887

图 17-6: 复位、看门狗定时器、振荡器起振定时器和上电延时定时器时序

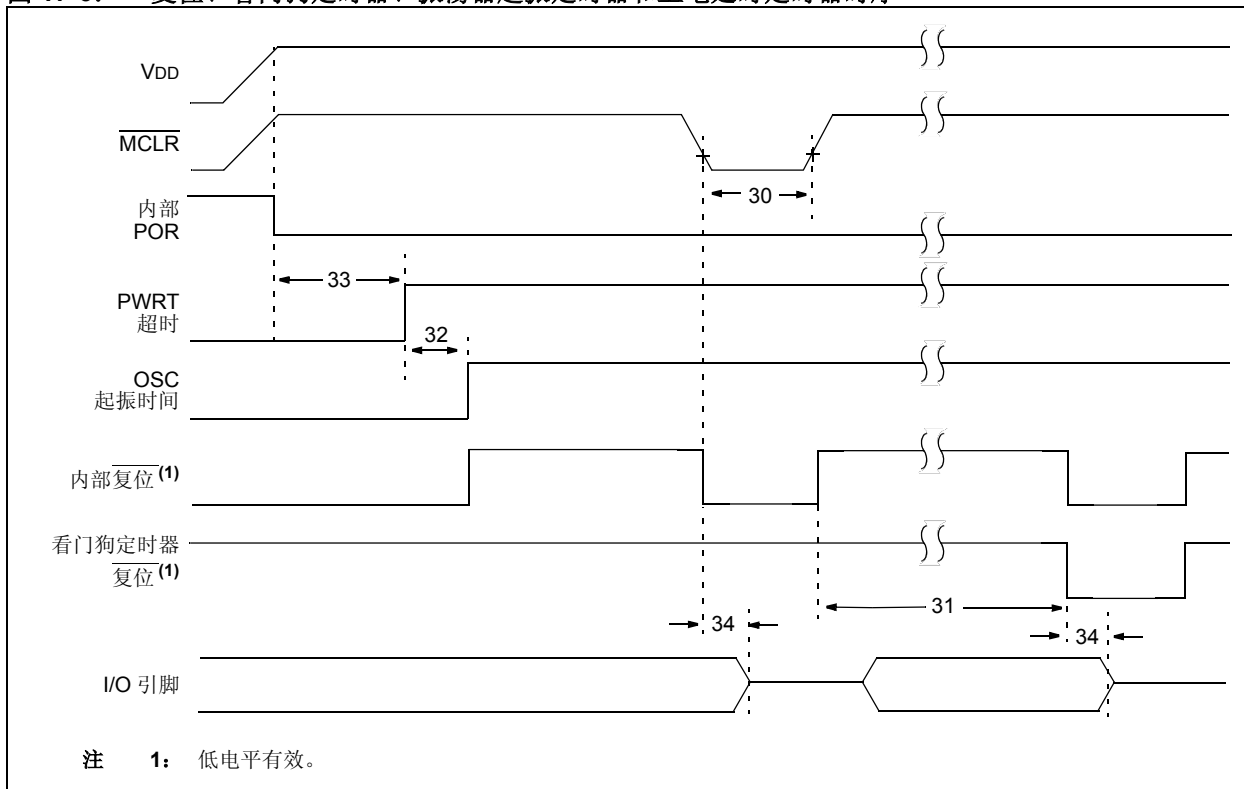
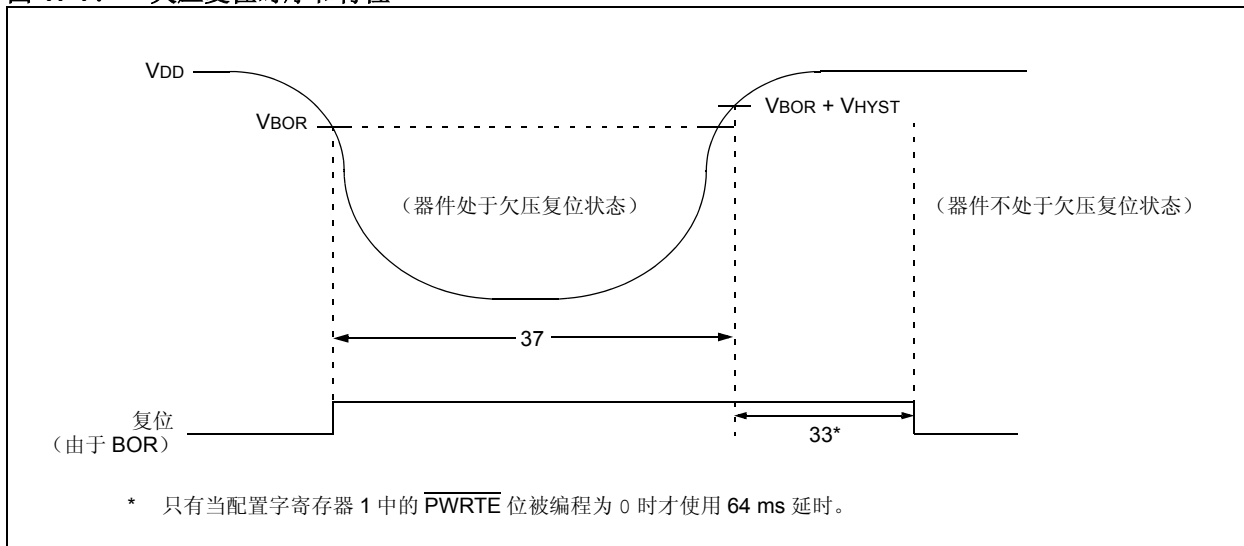


图 17-7: 欠压复位时序和特性



PIC16F882/883/884/886/887

表 17-4: 复位、看门狗定时器、振荡器起振定时器、上电延时定时器和欠压复位参数

标准工作条件 (除非另外说明)							
工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
30	T _{MCL}	MCLR 脉冲宽度 (低电平)	2 5	— —	— —	μs μs	V _{DD} = 5V, -40°C 至 $+85^{\circ}\text{C}$ V _{DD} = 5V
31	T _{WDT}	看门狗定时器超时周期 (无预分频器)	10 10	16 16	29 31	ms ms	V _{DD} = 5V, -40°C 至 $+85^{\circ}\text{C}$ V _{DD} = 5V
32	T _{OSt}	振荡器起振定时器周期 (1, 2)	—	1024	—	T _{OSC}	(注 3)
33*	T _{PWRT}	上电延时定时器周期	40	65	140	ms	
34*	T _{IOZ}	MCLR 低电平或看门狗定时器复位引起的 I/O 高阻态时间	—	—	2.0	μs	
35	V _{BOR}	欠压复位电压	2.0	—	2.2	V	BOR4V 位 = 0 (注 4)
35	V _{BOR}	欠压复位电压	TBD	4.0	TBD	V	BOR4V 位 = 1 (注 4)
36*	V _{HYST}	欠压复位迟滞	—	50	—	mV	
37*	T _{BOR}	欠压复位最小检测周期	100	—	—	μs	V _{DD} ≤ V _{BOR}

* 这些参数仅为特征值, 未经测试。

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

注 1: 指令周期时间 (T_{cy}) 等于输入振荡器时基周期的 4 倍。所有规范值均基于器件在标准工作条件下执行代码所对应的特定振荡器类型的特征数据。超过这些规范值可能导致振荡器运行不稳定和 / 或电流消耗超出预期。所有器件在测试“最小值”时都在 OSC1 引脚连接了外部时钟。当使用了外部时钟输入时, 所有器件的“最大值”周期时限为“DC” (没有时钟)。

2: 设计值。

3: 较慢的时钟周期。

4: 要确保电压值不超出公差范围, 必须在 V_{DD} 和 V_{SS} 与器件之间在尽可能靠近器件的地方连接去耦电容。建议使用值为 0.1 μF 和 0.01 μF 的并联电容。

PIC16F882/883/884/886/887

图 17-8: TIMER0 和 TIMER1 的外部时钟时序

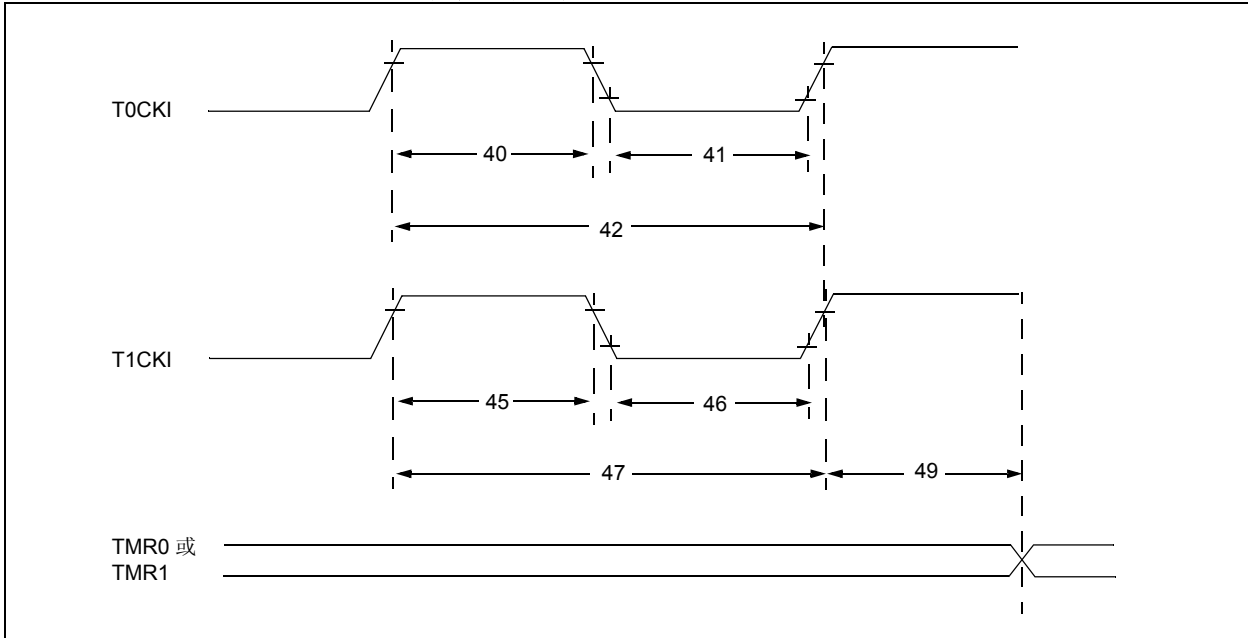


表 17-5: TIMER0 和 TIMER1 的外部时钟要求

标准工作条件 (除非另外说明)								
工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
参数编号	符号	特性		最小值	典型值†	最大值	单位	条件
40*	T _{T0H}	T0CKI 高电平脉冲宽度	无预分频器	$0.5 T_{CY} + 20$	—	—	ns	
			有预分频器	10	—	—	ns	
41*	T _{T0L}	T0CKI 低电平脉冲宽度	无预分频器	$0.5 T_{CY} + 20$	—	—	ns	
			有预分频器	10	—	—	ns	
42*	T _{T0P}	T0CKI 周期		取以下较大的值: 20 或 $T_{CY} + 40$ N	—	—	ns	N = 预分频值 (2,4,...,256)
45*	T _{T1H}	T1CKI 高电平时间	同步, 无预分频器	$0.5 T_{CY} + 20$	—	—	ns	
			同步, 带有预分频器	15	—	—	ns	
			异步	30	—	—	ns	
46*	T _{T1L}	T1CKI 低电平时间	同步, 无预分频器	$0.5 T_{CY} + 20$	—	—	ns	
			同步, 带有预分频器	15	—	—	ns	
			异步	30	—	—	ns	
47*	T _{T1P}	T1CKI 输入周期	同步	取以下较大的值: 30 或 $T_{CY} + 40$ N	—	—	ns	N = 预分频值 (1,2,4,8)
			异步	60	—	—	ns	
48	F _{T1}	Timer1 振荡器输入频率范围 (将 T10SCEN 位置 1, 使能振荡器)		—	32.768	—	kHz	
49*	TCKEZTMR1	出现外部时钟边沿到定时器递增的延时		$2 T_{OSC}$	—	$7 T_{OSC}$	—	定时器处于同步工作模式

* 这些参数仅为特征值, 未经测试。

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

PIC16F882/883/884/886/887

图 17-9: 捕捉 / 比较 / PWM 时序 (ECCP)

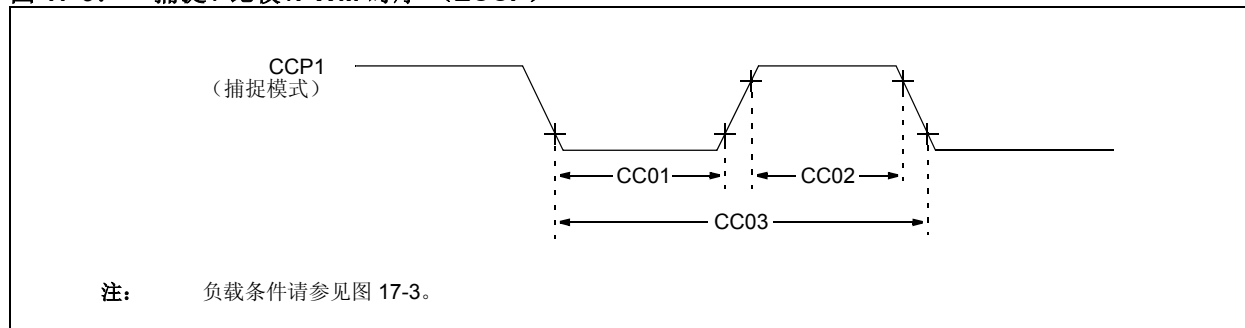


表 17-6: 捕捉 / 比较 / PWM 要求 (ECCP)

标准工作条件 (除非另外说明)								
工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
参数编号	符号	特性		最小值	典型值 †	最大值	单位	条件
CC01*	TccL	CCP1 输入低电平时间	无预分频器	$0.5T_{CY} + 20$	—	—	ns	
			有预分频器	20	—	—	ns	
CC02*	TccH	CCP1 输入高电平时间	无预分频器	$0.5T_{CY} + 20$	—	—	ns	
			有预分频器	20	—	—	ns	
CC03*	TccP	CCP1 输入周期		$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = 预分频值 (1、4 或 16)

* 这些参数仅为特征值, 未经测试。

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

PIC16F882/883/884/886/887

表 17-7: 比较器规范

标准工作条件（除非另外说明） 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$									
参数编号	符号	特性		最小值	典型值 †	最大值	单位	备注	
CM01	VOS	输入失调电压		—	±5.0	±10	mV	(VDD - 1.5)/2	
CM02	VCM	输入共模电压		0	—	VDD - 1.5	V		
CM03*	CMRR	共模抑制比		+55	—	—	dB		
CM04*	TRT	响应时间		下降	—	150	600	ns	(注 1)
				上升	—	200	1000	ns	
CM05*	Tmc2coV	比较器模式改变到输出有效的 时间		—	—	10	µs		

* 这些参数仅为特征值，未经测试。

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

注 1: 响应时间是在比较器一个输入端的电压从 (VDD - 1.5)/2 - 100 mV 变化到 (VDD - 1.5)/2 + 20 mV 时测得的。

表 17-8: 比较器参考电压 (CVREF) 规范

标准工作条件（除非另外说明） 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
参数编号	符号	特性		最小值	典型值 †	最大值	单位	
CV01*	CLSB	量化台阶大小 (2)		—	VDD/24	—	V	低电压范围 (VRR = 1)
				—	VDD/32	—	V	高电压范围 (VRR = 0)
CV02*	CACC	绝对准确度		—	±1/2	LSb	LSb	低电压范围 (VRR = 1)
				—	±1/2	LSb	LSb	高电压范围 (VRR = 0)
CV03*	CR	单位电阻值 (R)		—	2k	—	Ω	
CV04*	CST	稳定时间 (1)		—	—	10	µs	

* 这些参数仅为特征值，未经测试。

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

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

2: 更多信息请参见第 8.10 节“比较器参考电压”。

PIC16F882/883/884/886/887

表 17-9: PIC16F882/883/884/886/887 A/D 转换器 (ADC) 特性

标准工作条件 (除非另外说明)							
工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
AD01	NR	分辨率	—	—	10 位	位	
AD02	EIL	积分误差	—	—	± 1	LSb	$V_{REF} = 5.12\text{V}$
AD03	EDL	微分误差	—	—	± 1	LSb	不丢失编码至 10 位 $V_{REF} = 5.12\text{V}$
AD04	E _{OFF}	失调误差	—	1.5	TBD	LSb	$V_{REF} = 5.12\text{V}$
AD07	E _{GN}	增益误差	—	± 1	TBD	LSb	$V_{REF} = 5.12\text{V}$
AD06 AD06A	V _{REF}	参考电压 ⁽³⁾	2.2 2.7	—	— V _{DD}	V	确保 1 LSb 精度的绝对最小值
AD07	V _{AIN}	满量程范围	V _{SS}	—	V _{REF}	V	
AD08	Z _{AIN}	建议的模拟电压源阻抗	—	—	10	k Ω	
AD09*	I _{REF}	V _{REF} 输入电流 ⁽³⁾	10	—	1000	μA	在 V _{AIN} 采集期间。 根据 V _{HOLD} 与 V _{AIN} 的差值。
			—	—	50	μA	在 A/D 转换期间。

表注: TBD = 待定。

* 这些参数仅为特征值, 未经测试。

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

注 1: 总绝对误差包括积分、微分、失调和增益误差。

2: A/D 转换结果不会因输入电压的递增而递减, 并且不会丢失编码。

3: ADC V_{REF} 来自选作参考输入引脚的 V_{REF} 引脚或 V_{DD} 引脚。

4: 关闭 A/D 后, 除泄漏电流外, A/D 不消耗任何其他电流。掉电电流规范包含 ADC 模块的所有泄漏电流。

PIC16F882/883/884/886/887

表 17-10: PIC16F882/883/884/886/887 A/D 转换要求

标准工作条件 (除非另外说明)							
工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
参数编号	符号	特性		最小值	典型值 †	最大值	单位
AD130*	TAD	A/D 时钟周期	1.6	—	9.0	μs	基于 TOSC, $V_{\text{REF}} \geq 3.0\text{V}$
			3.0	—	9.0	μs	基于 TOSC, V_{REF} 满量程
		A/D 内部 RC 振荡周期	3.0	6.0	9.0	μs	ADCS<1:0> = 11 (ADRC 模式)
			1.6	4.0	6.0	μs	当 $V_{\text{DD}} = 2.5\text{V}$ 时 当 $V_{\text{DD}} = 5.0\text{V}$ 时
AD131	TCNV	转换时间 (不包括采集时间) (1)	—	11	—	TAD	将 GO/DONE 置 1 将新数据保存到 A/D 结果寄存器中
AD132*	TACQ	采集是		11.5	—	μs	
AD133*	TAMP	放大器稳定时间	—	—	5	μs	
AD134	TGO	Q4 至 A/D 时钟启动的时间	—	Tosc/2	—	—	如果选择 RC 作为 A/D 时钟源, 在 A/D 时钟开始前要加上一个 Tcy 时间, 用于执行 SLEEP 指令。
			—	Tosc/2 + Tcy	—	—	

* 这些参数仅为特征值, 未经测试。

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

注 1: 将在接下来的 Tcy 周期读 ADRESH 和 ADRESL 寄存器。

2: 最小条件请参见第 9.3 节“A/D 采集要求”。

PIC16F882/883/884/886/887

图 17-10: PIC16F882/883/884/886/887 A/D 转换时序 (正常模式)

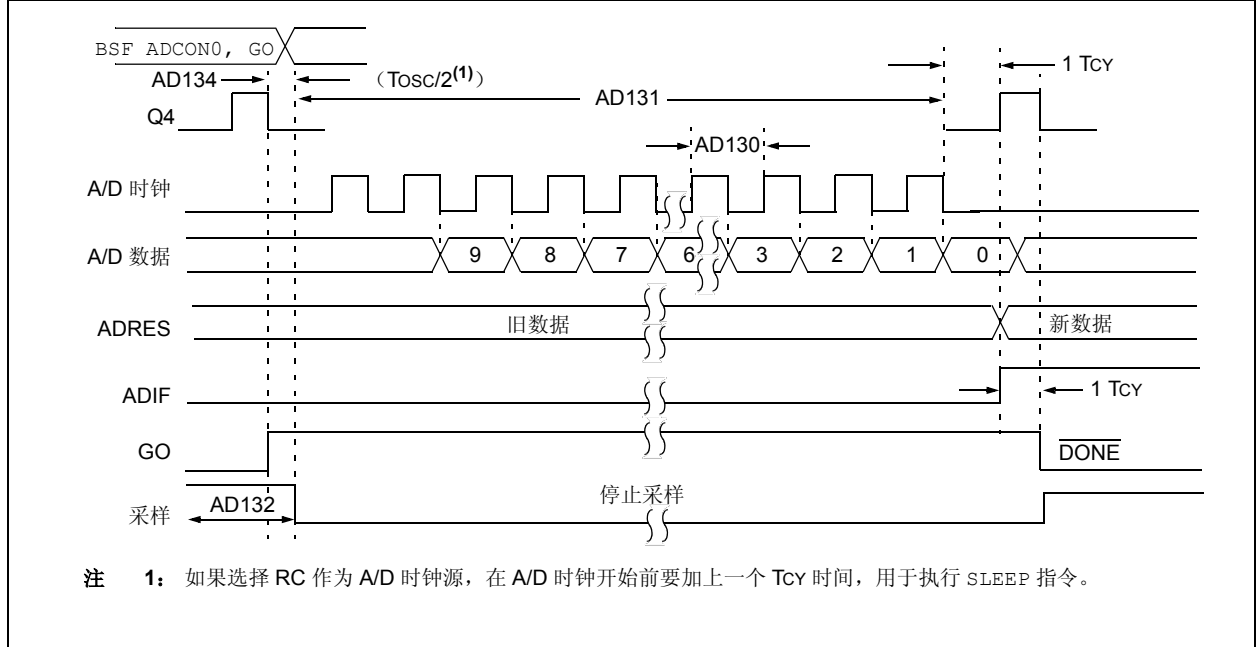
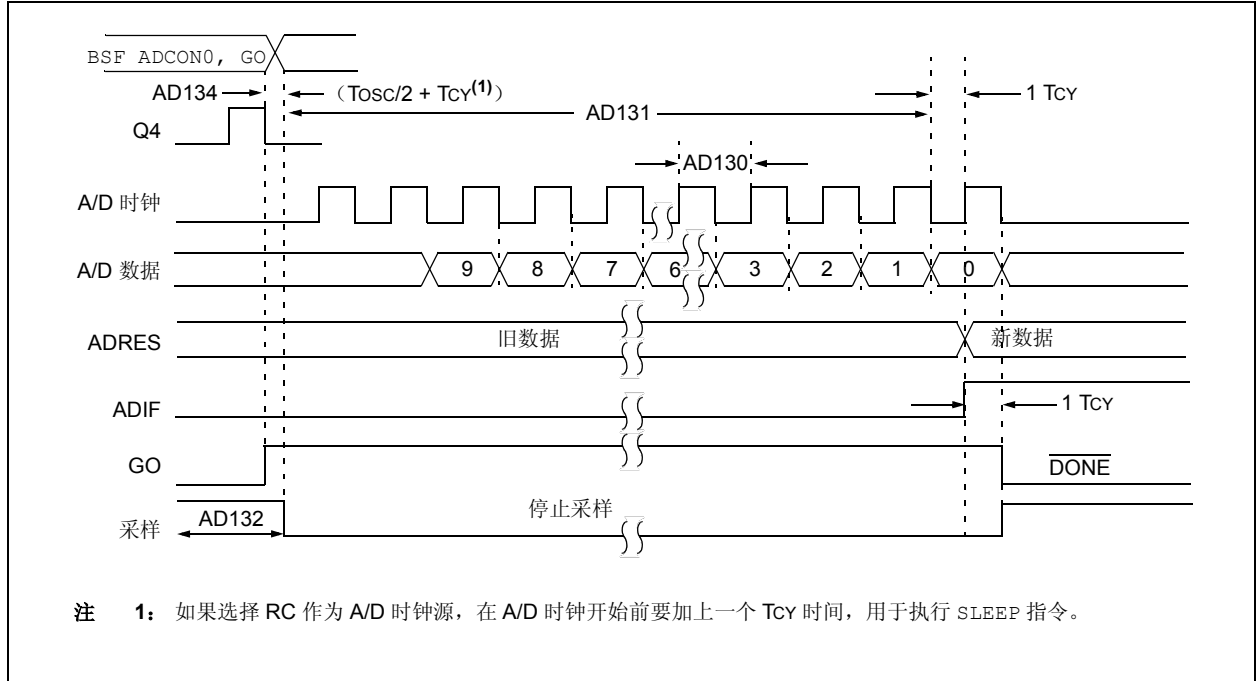


图 17-11: PIC16F882/883/884/886/887 A/D 转换时序 (休眠模式)



PIC16F882/883/884/886/887

注:

PIC16F882/883/884/886/887

18.0 直流和交流特性图表

目前暂无图表。

PIC16F882/883/884/886/887

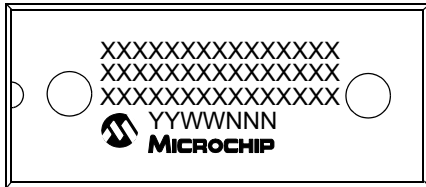
注:

PIC16F882/883/884/886/887

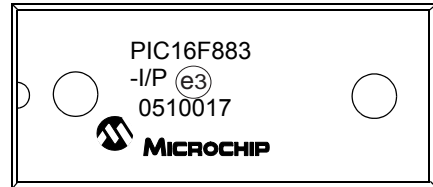
19.0 封装信息

19.1 封装标识信息

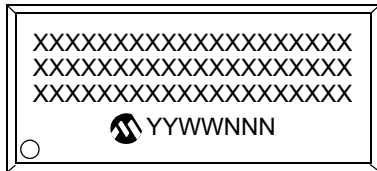
28 引脚 PDIP



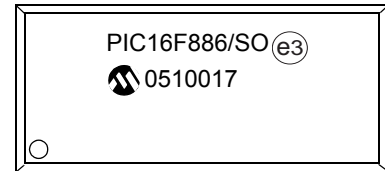
示例



28 引脚 SOIC (7.50 mm)



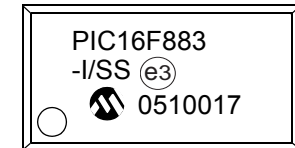
示例



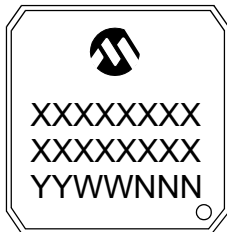
28 引脚 SSOP



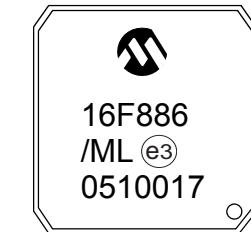
示例



28 引脚 QFN



示例



图注:

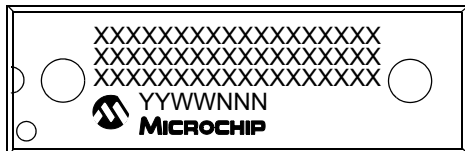
XX...X	客户信息
Y	年份代码 (日历年的最后一位数字)
YY	年份代码 (日历年的最后两位数字)
WW	星期代码 (一月一日的星期代码为“01”)
NNN	以字母数字排序的追踪代码
(e3)	雾锡 (Matte Tin, Sn) 的 JEDEC 无铅标志
*	表示无铅封装。JEDEC 无铅标志 ((e3)) 标示于此种封装的外包装上。

注: Microchip 元器件编号如果无法在同一行内完整标注, 将换行标出, 因此会限制表示客户信息的字符数。

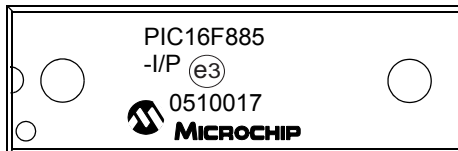
PIC16F882/883/884/886/887

19.1 封装标识信息（续）

40 引脚 PDIP



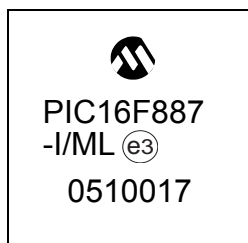
示例



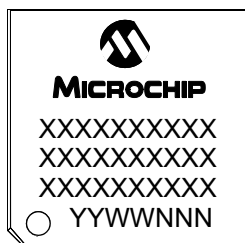
44 引脚 QFN



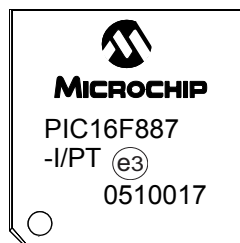
示例



44 引脚 TQFP



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
 (e3) 雾锡（Matte Tin, Sn）的 JEDEC 无铅标志
 * 表示无铅封装。JEDEC 无铅标志（e3）标示于此种封装的外包装上。

注: Microchip 元器件编号如果无法在同一行内完整标注，将换行标出，因此会限制表示客户信息的字符数。

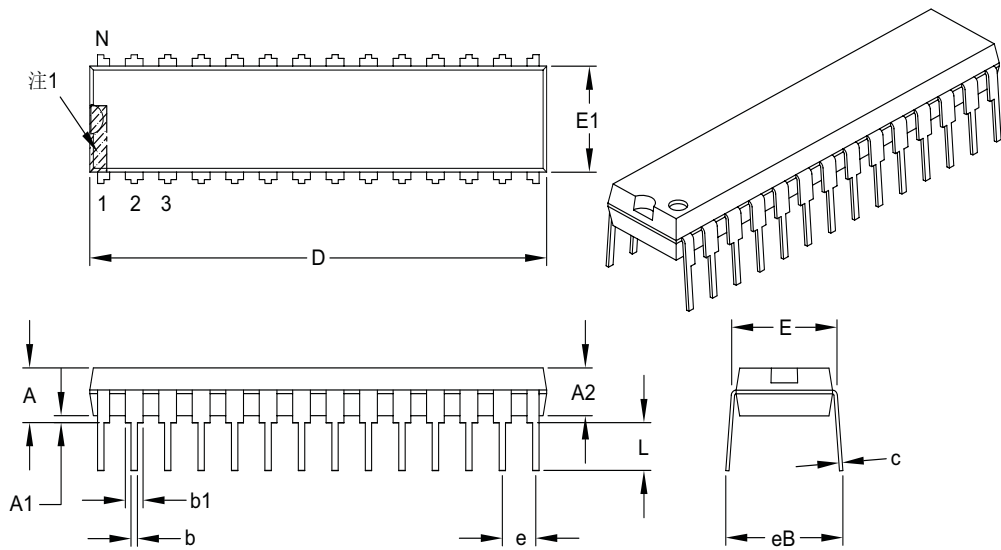
PIC16F882/883/884/886/887

19.2 封装详细信息

以下部分将介绍各种封装的技术细节。

28 引脚小型塑封双列直插封装（SP 或 PJ）——300 mil 主体 [SPDIP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



	单位	英寸		
		尺寸范围	最小值	正常值
引脚数	N	28		
引脚间距	e	.100 BSC		
顶端到底座平面距离	A	-	-	.200
塑模封装厚度	A2	.120	.135	.150
基面到底座平面距离	A1	.015	-	-
肩宽	E	.290	.310	.335
塑模封装宽度	E1	.240	.285	.295
总长度	D	1.345	1.365	1.400
足尖到底座平面距离	L	.110	.130	.150
引脚厚度	c	.008	.010	.015
引脚上部宽度	b1	.040	.050	.070
引脚下部宽度	b	.014	.018	.022
总行间距 §	eB	-	-	.430

注：

1. 引脚1外观定位特性可以变化，但必须位于阴影区域内。
2. § 为重要特性。
3. 尺寸D和E1不含塑模的毛边或突起。塑模的毛边或突起每侧不得超过0.010"。
4. 尺寸和公差依据ASME Y14.5M。

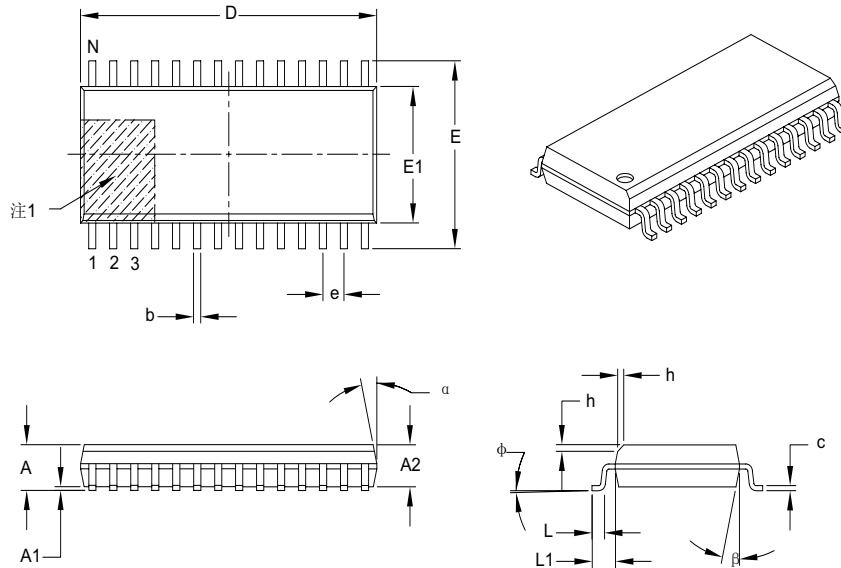
BSC: 基本尺寸。理论精确值，无公差。

Microchip 图号: C04-070B

PIC16F882/883/884/886/887

28 引脚塑封小外形封装 (SO 或 OI) —— 宽型, 7.50 mm 主体 [SOIC]

注: 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



尺寸范围	单位	毫米		
		最小值	正常值	最大值
引脚数	N	28		
引脚间距	e	1.27 BSC		
总高度	A	—	—	2.65
塑模封装厚度	A2	2.05	—	—
悬空间隙 §	A1	0.10	—	0.30
总宽度	E	10.30 BSC		
塑模封装宽度	E1	7.50 BSC		
总长度	D	17.90 BSC		
倒角 (可选)	h	0.25	—	0.75
底足长度	L	0.40	—	1.27
占位长度	L1	1.40 REF		
底足上部倾角	φ	0°	—	8°
引脚厚度	c	0.18	—	0.33
引脚宽度	b	0.31	—	0.51
塑模上部锥度	α	5°	—	15°
塑模下部锥度	β	5°	—	15°

注:

1. 引脚1外观定位特性可以改变, 但必须位于阴影区域内。
2. § 为重要特性。
3. 尺寸D和E1不含塑模的毛边或突起。塑模的毛边或突起每侧不得超过0.15 mm。
4. 尺寸和公差依据ASME Y14.5M。

BSC: 基本尺寸。理论精确值, 无公差。

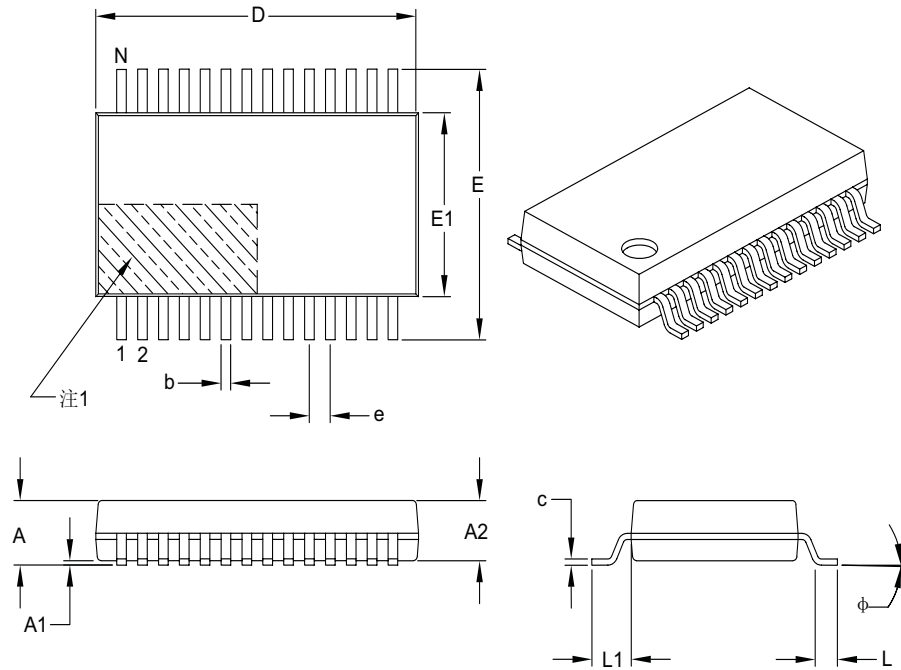
REF: 参考尺寸, 通常无公差, 仅供参考。

Microchip 图号: C04-052B

PIC16F882/883/884/886/887

28 引脚塑封缩小外形封装 (SS) —— 5.30 mm 主体 [SSOP]

注： 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



单位		毫米		
尺寸范围		最小值	正常值	最大值
引脚数	N	28		
引脚间距	e	0.65 BSC		
总高度	A	-	-	2.00
塑模封装厚度	A2	1.65	1.75	1.85
悬空间隙	A1	0.05	-	-
总宽度	E	7.40	7.80	8.20
塑模封装宽度	E1	5.00	5.30	5.60
总长度	D	9.90	10.20	10.50
底足长度	L	0.55	0.75	0.95
占位长度	L1	1.25 REF		
引脚厚度	c	0.09	-	0.25
底足倾角	ϕ	0°	4°	8°
引脚宽度	b	0.22	-	0.38

注：

1. 引脚1外观定位特性可以改变，但必须位于阴影区域内。
2. 尺寸D和E1不包含塑模的毛边或突起。塑模的毛边或突起每侧不得超过0.20 mm。
3. 尺寸和公差依据ASME Y14.5M。

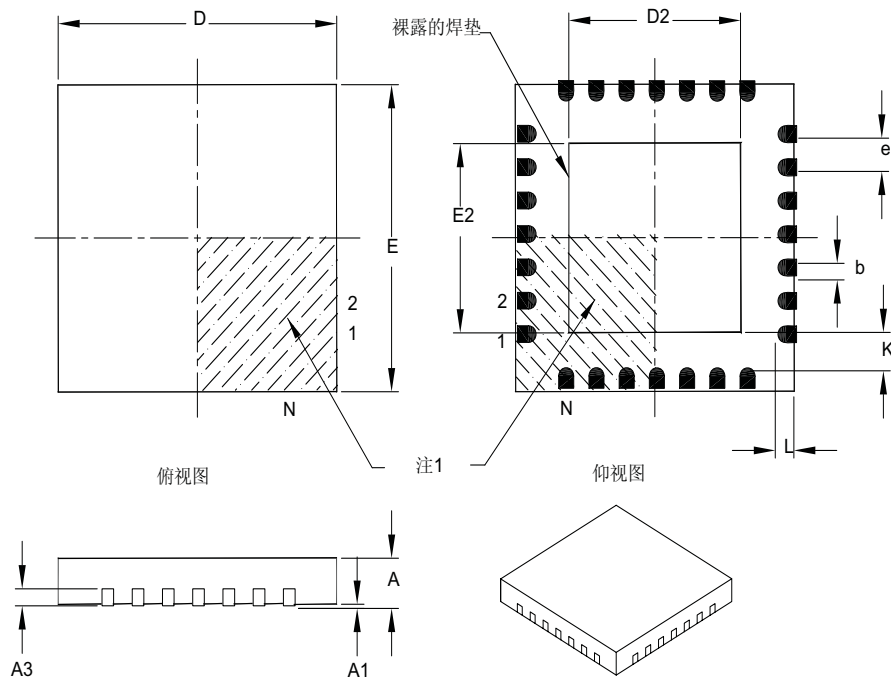
BSC: 基本尺寸。理论精确值，无公差。
REF: 参考尺寸，通常无公差，仅供参考。

Microchip 图号: C04-073B

PIC16F882/883/884/886/887

28 引脚塑封正方形扁平无引脚封装 (MM) —— 6x6x0.9 mm 主体 [QFN-S]

注： 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



尺寸范围	单位	毫米		
		最小值	正常值	最大值
引脚数	N	28		
引脚间距	e	0.65 BSC		
总高度	A	0.80	0.90	1.00
悬空间隙	A1	0.00	0.02	0.05
触点厚度	A3	0.20 REF		
总宽度	E	6.00 BSC		
裸露焊垫宽度	E2	3.65	3.70	4.70
总长度	D	6.00 BSC		
裸露焊垫长度	D2	3.65	3.70	4.70
触点宽度	b	0.23	0.38	0.43
触点长度	L	0.30	0.40	0.50
触点至裸露焊垫距离	K	0.20	-	-

注：

1. 引脚1外观定位特性可以变化，但必须位于阴影区域内。
封装为切割分离。
3. 尺寸和公差依据ASME Y14.5M。

BSC: 基本尺寸。理论精确值，无公差。

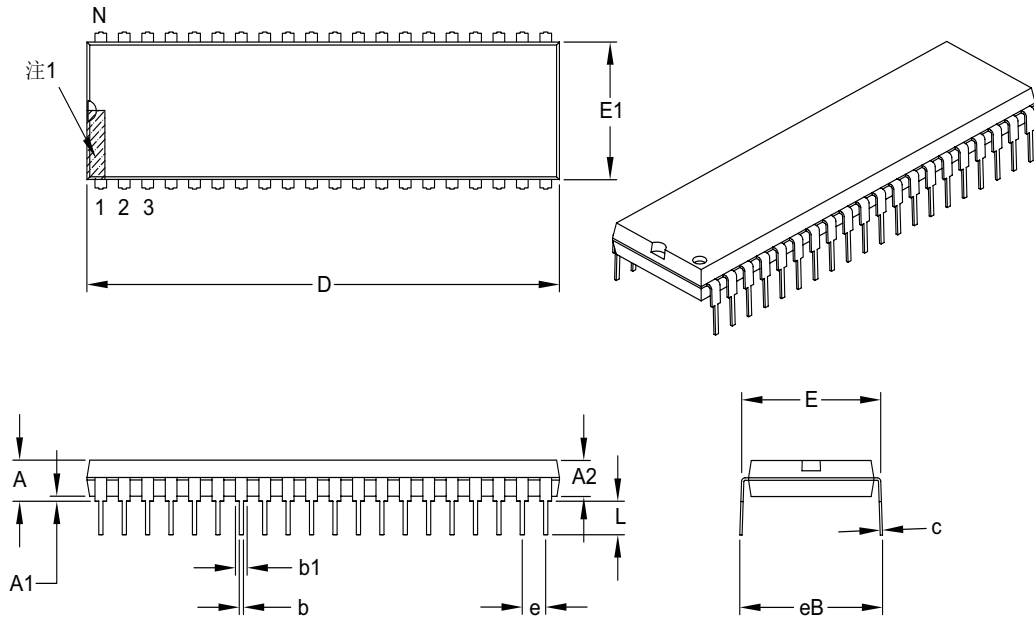
REF: 参考尺寸，通常无公差，仅供参考。

Microchip 图号: C04-124B

PIC16F882/883/884/886/887

40 引脚塑封双列直插 (P 或 PL) ——600 mil 主体 [PDIP]

注： 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



尺寸范围	单位	英寸		
		最小值	正常值	最大值
引脚数	N	40		
引脚间距	e	.100 BSC		
顶部至底座平面距离	A	-	-	.250
塑模封装厚度	A2	.125	-	.195
基座至底座平面距离	A1	.015	-	-
肩宽	E	.590	-	.625
塑模封装宽度	E1	.485	-	.580
总长度	D	1.980	-	2.095
足尖至底座平面距离	L	.115	-	.200
引脚厚度	c	.008	-	.015
引脚上部宽度	b1	.030	-	.070
引脚下部宽度	b	.014	-	.023
总行间距 §	eB	-	-	.700

注：

1. 引脚1的外观定位特性可以变化，但必须位于阴影区域内。
2. § 为重要特性。
3. 尺寸D和E1不包含塑模的毛边或突起。塑模的毛边或突起每侧不得超过0.010"。
4. 尺寸和公差依据ASME Y14.5M。

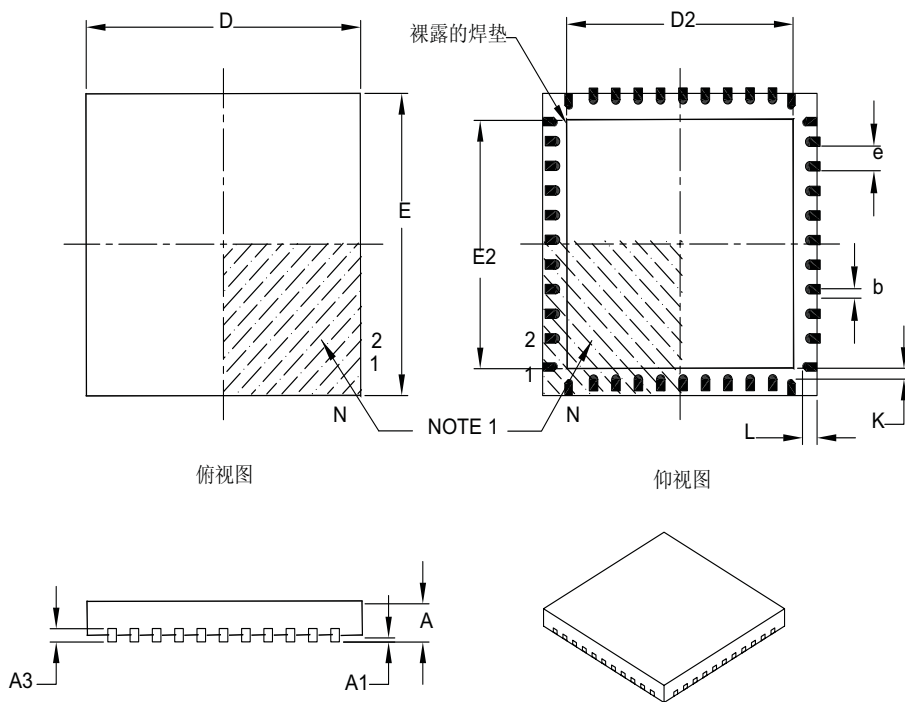
BSC：基本尺寸。理论精确值，无公差。

Microchip 图号：C04-016B

PIC16F882/883/884/886/887

44 引脚塑封正方形扁平无引脚封装 (ML) —— 8x8 mm 主体 [QFN]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



尺寸范围	单位	毫米		
		最小值	正常值	最大值
引脚数	N	44		
引脚间距	e	0.65 BSC		
总高度	A	0.80	0.90	1.00
悬空间隙	A1	0.00	0.02	0.05
触点厚度	A3	0.20 REF		
总宽度	E	8.00 BSC		
裸露焊垫宽度	E2	6.30	6.45	6.80
总长度	D	8.00 BSC		
裸露焊垫长度	D2	6.30	6.45	6.80
触点宽度	b	0.25	0.30	0.38
触点长度	L	0.30	0.40	0.50
触点至裸露焊垫距离	K	0.20	-	-

注：

1. 引脚1外观定位特性可以变化，但必须位于阴影区域内。
2. 封装为切割分离。
3. 尺寸和公差依据ASME Y14.5M。

BSC: 基本尺寸。理论精确值，无公差。

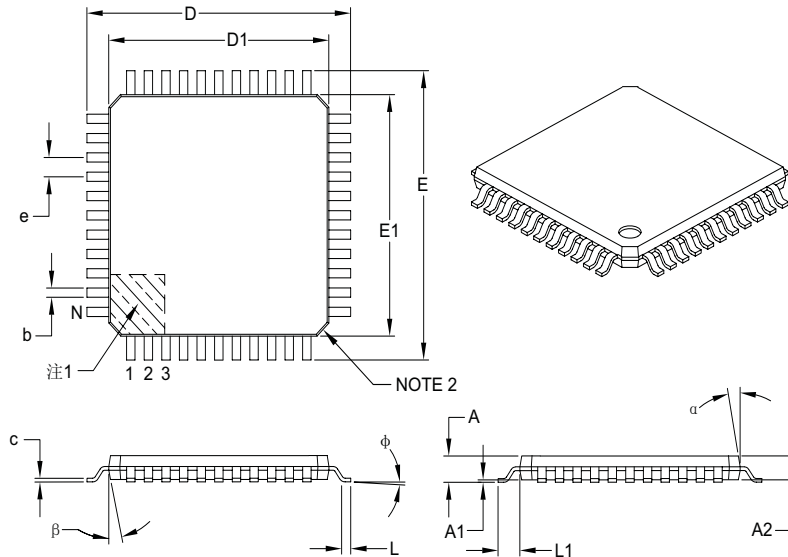
REF: 参考尺寸，通常无公差，仅供参考。

Microchip 图号: C04-103

PIC16F882/883/884/886/887

44 引脚塑封薄型正方形扁平封装 (PT) —— 10x10x1 mm 主体, 2.00 mm 占位 [TQFP]

注: 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



尺寸范围	单位	毫米		
		最小值	正常值	最大值
引脚数	N	44		
引脚间距	e	0.80 BSC		
总高度	A	-	-	1.20
塑模封装厚度	A2	0.95	1.00	1.05
悬空间隙	A1	0.05	-	0.15
底足长度	L	0.45	0.60	0.75
占位长度	L1	1.00 REF		
底足倾角	φ	0°	3.5°	7°
总宽度	E	12.00 BSC		
总长度	D	12.00 BSC		
塑模封装宽度	E1	10.00 BSC		
塑模封装长度	D1	10.00 BSC		
引脚厚度	c	0.09	-	0.20
引脚宽度	b	0.30	0.37	0.45
塑模顶部锥度	α	11°	12°	13°
塑模底部锥度	β	11°	12°	13°

注:

1. 引脚1的外观定位特性可以变化, 但必须位于阴影区域内。
2. 四个角的倒角均为可选, 尺寸可以变化。
3. 尺寸D1和E1不包含塑模的毛边或突起。塑模的毛边或突起每侧不得超过0.25 mm。
4. 尺寸和公差依据ASME Y14.5M。

BSC: 基本尺寸。理论精确值, 无公差。
REF: 参考尺寸, 通常无公差, 仅供参考。

Microchip 图号: C04-076B

PIC16F882/883/884/886/887

注:

PIC16F882/883/884/886/887

附录 A: 数据手册和版本历史

版本 A (2006 年 5 月)

本数据手册的初始发行版。

版本 B (2006 年 7 月)

引脚图 (44 引脚 QFN 图); 修改了表 2-1 中的地址 1Dh (CCP2CON); 第 3.0、3.1 节; 第 3.4.4.6 节; 表 3; 表 3-1 (ANSEL); 表 3-3 (CCP2CON); 寄存器 3-1; 寄存器 3-2; 寄存器 3-3; 寄存器 3-4; 寄存器 3-9; 寄存器 3-10; 寄存器 3-11; 寄存器 3-12; 寄存器 3-14; 表 3-5 (ANSEL); 图 3-5; 图 3-11; 图 8-2; 图 8-3; 图 9-1; 寄存器 9-1; 第 9.1.4 节; 例 10-4; 图 11-5; 表 11-5 (P1M); 第 11.5.2 节; 第 11.5.7 节中第 4 项; 表 11-7 (CCP2CON); 第 12.3.1 节 (参数 3); 图 12-6 (标题); 第 14.2、14.3 和 14.4 节直流特性 (最大值); 表 14-4 (OSCCON); 第 14.3 节 (TMR0) 以及第 14.3.2 节 (TMR0)。

版本 C

第 19.0 节封装信息: 替换了封装图并加了注。

增加了 PIC16F882 器件编号。

将 PICmicro 替换为 PIC。

版本 D

替换了封装图 (Rev. AM); 替换了开发支持一节。修改了产品标识一节。

附录 B: 从其他 PIC® 器件移植

本节讨论了从其他 PIC 器件移植到 PIC16F88X 系列器件的一些问题。

B.1 PIC16F87X 到 PIC16F88X

表 B-1: 特性比较

特性	PIC16F87X	PIC16F88X
最大工作速度	20 MHz	20 MHz
最大程序存储容量 (字)	8192	8192
SRAM (字节)	368	368
A/D 分辨率	10 位	10 位
数据 EEPROM (字节)	256	256
定时器 (8/16 位)	2/1	2/1
振荡器模式	4	8
欠压复位	有	有 (2.1V/4V)
WDT/BOR 的软件控制选项	无	有
内部上拉	RB<7:4>	RB<7:0> 和 MCLR
电平变化中断	RB<7:4>	RB<7:0>
比较器	2	2
参考	CVREF	CVREF 和 VP6
ECCP/CCP	0/2	1/1
超低功耗唤醒	无	有
扩展的 WDT	无	有
INTOSC 频率	无	32 kHz 至 8 MHz
时钟切换	无	有
MSSP	标准	带从动地址屏蔽
USART	AUSART	EUSART
ADC 通道	8	14

注: 该器件设计为按照数据手册上的参数运行。它已经通过电气规范测试, 该规范用于确定器件与这些参数的一致性。由于制造时的工艺差异, 此器件可能与早期版本具有不同的性能特性。这些差异可能会导致此器件在应用中的性能与其早期版本有所不同。

PIC16F882/883/884/886/887

注:

PIC16F882/883/884/886/887

索引

A

A/D	
规范	257, 258
内部采样开关阻抗 (R _{ss})	107
源阻抗	107
ACKSTAT	194
ACKSTAT 状态标志	194
ADC	99
采集要求	107
参考电压 (V _{REF})	100
端口配置	100
工作原理	102
计算采集时间	107
结果格式化	102
开始 A/D 转换	102
框图	99
配置	100
配置中断	103
特殊事件触发器	102
通道选择	100
相关的寄存器	109
在休眠模式下的工作原理	102
中断	101
转换步骤	103
转换时钟	100
ADCON0 寄存器	104
ADCON1 寄存器	105
ADRESH 寄存器 (ADFM = 0)	106
ADRESH 寄存器 (ADFM = 1)	106
ADRESL 寄存器 (ADFM = 0)	106
ADRESL 寄存器 (ADFM = 1)	106
ANSEH 寄存器	48
ANSEL 寄存器	40

B

BAUDCTL 寄存器	160
BF	194
BF 状态标志	194
BRG	191
版本历史	273
保险丝。参见配置位	
比较模块。参见增强型捕捉 / 比较 / PWM (ECCP)	
比较器	
C2OUT 作为 T1 的门控信号线	77
C2OUT 作为 T1 的门控信号源	91
复位影响	87
工作原理	83
规范	256
带 Timer1 的同步 COUT	91
响应时间	85
休眠期间工作原理	87
比较器参考电压 (CVREF)	
规范	256
比较器参考电压 (CVREF)	94
比较器参考电压 (CVREF)	85
比较器参考电压 (CVREF)	
复位影响	87
比较器模块	83
C1 不同输入条件下的输出状态	85
相关的寄存器	97
编程, 器件指令	225
变更通知客户服务	283
波特率发生器	191

捕捉 / 比较 / PWM (CCP)

比较模式	127
CCP 引脚配置	127
软件中断模式	126
Timer1 模式选择	126, 127
比较模式软	
件中断模式	127
捕捉模式	126
定时器资源	125
PWM 模式	128
PWM 频率和分辨率示例, 20 MHz	130
PWM 频率和分辨率示例, 8 MHz	130
设置操作	131
休眠模式下的操作	131
占空比	129
设置 PWM 操作	131
与捕捉、比较和 Timer1 相关的寄存器	148
与 PWM 和 Timer2 相关的寄存器	148
捕捉 / 比较器 / PWM (CCP)	
比较模式	
特殊事件触发信号	127
PWM 模式	
复位的影响	131
系统时钟频率改变	131
PWM 周期	129
捕捉模式。参见增强型捕捉 / 比较 / PWM (ECCP)	
捕捉 (CCP 模块)	
CCP 引脚配置	126

C

C 编译器	
MPLAB C18	236
MPLAB C30	236
CCP1CON (增强型) 寄存器	124
CCP2CON 寄存器	125
CM1CON0 寄存器	88
CM2CON0 寄存器	89
CM2CON1 寄存器	91
CONFIG1 寄存器	206
CONFIG2 寄存器	207
CPU 的特性	205
参考电压	
VP6 稳定周期	94
相关的寄存器	97
参考电压。参见比较器参考电压 (CVREF)	
操作码字段说明	225
超低功耗唤醒	16, 18, 40, 41
程序存储器	21
映射图和堆栈 (PIC16F883/884)	21
映射图和堆栈 (PIC16F886/887)	21
串行时钟, SCK	179
串行数据输出, SDO	179
串行数据输入, SDI	179
串行外设接口。参见 SPI	
从动模式广播呼叫地址时序波形	188
从动模式广播呼叫地址序列	188
从动选择同步	182
从动选择, \overline{SS}	179
从其他 PIC 器件移值	273
存储器构成	21
程序	21
数据	22

PIC16F882/883/884/886/887

D

代码保护	223
代码示例	
A/D 转换	103
超低功耗唤醒初始化	41
初始化 PORTA	39
初始化 PORTB	47
初始化 PORTC	53
初始化 PORTD	57
初始化 PORTE	59
改变捕捉预分频比	126
间接寻址	36
将 Status 和 W 寄存器保存在 RAM 中	219
将预分频器分配给 Timer0	74
将预分频器分配给 WDT	74
写入闪存程序存储器	119
写校验	120
装载 SSPBUF 寄存器	180
电气规范	239
掉电模式（休眠）	222
定时器	
Timer2	
T2CON	82
Timer1	
T1CON	79
读者反馈表	284
读—修改—写操作	225
多主机模式	199
多主机通信，总线冲突和总线仲裁	199

E

ECCPAS 寄存器	141
ECCP. 参见增强型捕捉 / 比较 / PWM	
EEADRH 寄存器	111
EEADR 寄存器	111, 112
EECON1 寄存器	111, 113
EECON2 寄存器	111
EEDATH 寄存器	112
EEDAT 寄存器	112
EEPROM 数据存储	
避免误写	120
写校验	120
EUSART	149
波特率发生器（BRG）	
波特率误差，计算	161
波特率，异步模式	162
高波特率选择位（BRGH 位）	161
公式	161
自动波特率检测	165
同步从动模式	173
发送	173
接收	174
相关的寄存器	
发送	173
接收	174
同步主控模式	169
发送	169
接收	171
相关的寄存器	
发送	170
接收	172
相关的寄存器	
波特率发生器	161
异步模式	151
12 位间隔发送和接收	167

波特率发生器（BRG）	161
发送器	151
接收器	154
设置带地址检测的 9 位模式	156
时钟精确度	158
收到间隔字符时自动唤醒	166
相关的寄存器	
发送	153
相关寄存器	
接收	157

F

封装	263
标识	263, 264
PDIP 详细信息	265
复位	208
复位的影响	
PWM 模式	131
负载条件	248

G

高精度内部振荡器参数	251
固件指令	225
故障安全时钟监视	
清除故障安全条件	71
故障保护时钟监视	71
复位或从休眠状态唤醒	71
故障保护工作原理	71
故障保护检测	71
广播呼叫地址支持	188

H

汇编器	
MPASM 汇编器	236

I

i ² C 模块	
7 位发送时序	187
BRG 时序	191
多主机模式	199
启动条件期间由 SDA 仲裁引起的 BRG 复位	201
停止条件接收或发送时序	198
重复启动条件时序波形	193
主控模式 7 位地址接收时序	196
主控模式发送	194
总线冲突	
启动条件时序	200
i ² C（MSSP 模块）	
ACK 脉冲	185
i ² C（MSSP 模块）	
从动模式	185
读 / 写位信息（R/W 位）	186
i ² C 模块	
7 位接收时序	187
BRG 框图	191
波特率发生器	191
复位的影响	198
广播呼叫地址支持	188
时钟仲裁	198
时钟仲裁时序（主控发送）	198
停止条件时序	197
休眠模式下的操作	198
应答序列时序	197
主控模式操作	190
主控模式发送序列	190
主控模式启动条件时序	192

PIC16F882/883/884/886/887

主控模式支持	189	OPTION_REG (选项)	75
总线冲突		OSCCON (振荡器控制)	62
启动条件	200	OSCTUNE (振荡器调节)	66
启动条件时序	201	PCON (电源控制寄存器)	35
停止条件	203	PCON (电源控制)	211
停止条件时序 (情形 1)	203	PIE1 (外设中断允许 1)	31
停止条件时序 (情形 2)	203	PIE2 (外设中断允许 2)	32
应答	199	PIR1 (外设中断请求寄存器 1)	33
重复启动条件	202	PIR2 (外设中断请求寄存器 2)	34
重复启动条件的时序 (情形 2)	202	PORTA	39
重复启动条件时序 (情形 1)	202	PORTB	48
总线冲突时序	199	PORTC	53
i ² C 模式		PORTD	57
主控模式	189	PORTE	59
i ² C 模式 (MSSP)		PSTRCON (脉冲换向控制寄存器)	145
读/写信息位 (R/W 位)	186	PWM1CON (增强型 PWM 控制寄存器)	144
i ² C 主控模式接收	194	RCREG 寄存器	165
i ² C 主控模式重复启动条件时序	193	RCSTA (接收状态和控制寄存器)	159
i ² C (MSSP 模块)		SRCON (SR 锁存控制寄存器)	93
ACK 脉冲	186	SSPCON (SSP 控制寄存器 1)	177
ACK 脉冲	186	SSPCON2	
i ² C (MSSP 模块)		(SSP 控制寄存器 2)	178
串行时钟 (RC3/SCK/SCL)	186	SSPMSK (SSP 屏蔽寄存器)	204
发送	186	SSPSTAT (SSP 状态寄存器)	176
接收	186	STATUS	28
寻址	186	T1CON	79
ID 地址单元	223	T2CON	82
INTCON 寄存器	30	TRISA (三态 PORTA)	39
INTOSC		TRISB (三态 PORTB)	48
规范	250	TRISC (三态 PORTC)	53
INTOSC 规范	251	TRISD (三态 PORTD)	57
INTOSC 规范	250	TRISE (三态 PORTE)	59
IOCB 寄存器	49	TXSTA (发送状态和控制寄存器)	158
J		特殊功能寄存器	22
寄存器		特殊功能寄存器汇总	
ADCON0 (ADC 控制寄存器 0)	104	Bank 0	25
ADCON1 (ADC 控制寄存器 1)	105	特殊功能寄存器映射图	
ADRESH (ADC 结果寄存器的高字节) (ADFM = 0)		PIC16F883/884	23
.....	106	PIC16F886/887	24
ADRESH (ADC 结果寄存器的高字节) (ADFM = 1)		特殊寄存器小结	
.....	106	Bank 1	26
ADRESL (ADC 结果寄存器的低字节) (ADFM = 1)		Bank 2	27
.....	106	Bank 3	27
ADRESL (ADC 结果寄存器的低字节) (ADFM = 0)		WDTCON (看门狗定时器控制)	221
.....	106	WPUB (弱上拉 PORTB)	49
ANSELH (模拟选择高字节)	48	WRCON (参考电压控制寄存器)	97
ANSEL (模拟选择)	40	间隔时唤醒	166
BAUDCTL (波特率控制寄存器)	160	间隔字符 (12 位) 发送和接收	167
CCP1CON (增强型 CCP1 控制)	124	间接寻址、INDF 和 FSR 寄存器	36
CCP2CON (CCP2 控制寄存器)	125	交流特性	
CM1CON0 (C1 控制寄存器)	88	负载条件	248
CM2CON0 (C2 控制寄存器)	89	工业级和扩展级	249
CM2CON1 (C2 控制寄存器)	91	绝对最大额定值	239
CONFIG1 (配置字寄存器 1)	206	K	
CONFIG2 (配置字寄存器 2)	207	开发支持	235
ECCPAS (增强型 CCP 自动关闭控制寄存器)	141	看门狗定时器 (WDT)	220
EEDR (EEPROM 地址寄存器)	112	规范	253
EECON1 (EEPROM 控制寄存器 1)	113	模式	220
EEDATH (EEPROM 数据寄存器)	112	时钟源	220
EEDAT (EEPROM 数据寄存器)	112	相关的寄存器	221
复位值	213	周期	220
复位值 (特殊寄存器)	215	勘误表	12
INTCON (中断控制)	30	客户通知服务	283
IOCB (电平变化中断 PORTB)	49	客户支持	283
OPTION_REG (OPTION)	29	框图	

PIC16F882/883/884/886/887

ADC	99	框图 (SPI 模式)	179
ADC 传递函数	108	SPI 模式	179
比较模式工作	127	SPI 模式。参见 SPI	
比较器 C1	84	SSPBUF 寄存器	181
比较器 C1 和 ADC 参考电压	95	SSPSR 寄存器	181
比较器 C2	84	MSSP 模块	
波特率发生器	191	I ² C 操作	185
CCP PWM	128	控制寄存器	175
EUSART 发送	149	SPI 从动模式	182
EUSART 接收	150	SPI 主控模式	181
故障保护时钟监视器 (FSCM)	71	脉冲换向	145
晶振工作原理	64	模拟输入连接注意事项	90
看门狗定时器 (WDT)	220	模数转换器。参见 ADC	
MCLR 电路	209	N	
MSSP (I ² C 模式)	185	内部采样开关阻抗 (R _{ss})	107
MSSP 模式 (SPI 模式)	179	内部振荡器电路	250
MSSP (I ² C 主控模式)	189	INTOSC	
模拟输入模型	90, 108	规范	251
PIC16F883/886	14	O	
PIC16F884/887	15	OPTION_REG 寄存器	75
PWM (增强型)	132	OSCCON 寄存器	62
片上复位电路	208	OSCTUNE 寄存器	66
RA0 引脚	42	P	
RA1 引脚	43	P1A/P1B/P1C/P1D。参见增强型 捕捉 / 比较 / PWM (ECCP)	
RA2 引脚	43	132
RA3 引脚	44	PCL 和 PCLATH	36
RA4 引脚	44	堆栈	36
RA5 引脚	45	PCON 寄存器	35, 211
RA6 引脚	45	PICSTART 2 开发编程器	238
RA7 引脚	46	PICSTART Plus 开发编程器	238
RB0、RB1、RB2 和 RB3 引脚	50	PIE1 寄存器	31
RB4、RB5、RB6 和 RB7 引脚	51	PIE2 寄存器	32
RC0 引脚	54	PIR1 寄存器	33
RC1 引脚	54	PIR2 寄存器	34
RC2 引脚	54	PORTA	39
RC3 引脚	55	更多引脚功能	40
RC4 引脚	55	ANSEL 寄存器	40
RC5 引脚	55	超低功耗唤醒	40, 41
RC6 引脚	56	规范	251
RC7 引脚	56	RA0	42
RD0、RD1、RD2、RD3 和 RD4 引脚	58	RA1	43
RD5、RD6 和 RD7 引脚	58	RA2	43
RE3 引脚	60	RA3	44
时钟源	61	RA4	44
Timer2	81	RA5	45
Timer1	76	RA6	45
TMR0/WDT 预分频器	73	RA7	46
外部 RC 模式	65	相关的寄存器	46
谐振器工作原理	64	引脚说明和框图	42
在线串行编程连接	224	PORTA 寄存器	39
中断逻辑	217	PORTB	47
(CCP) 捕捉模式工作	126	电平变化时中断	47
M		电平变化中断	47
MCLR	209	P1B/P1C/P1D。参见增强型捕捉 / 比较 // PWM+	
内部	209	(ECCP+)	47
Microchip 因特网网站	283	其他引脚功能	47
MPLAB ASM30 汇编器、链接器和库管理器	236	ANSEH 寄存器	47
MPLAB ICD 2 在线调试器	237	弱上拉	47
MPLAB ICE 2000 高性能通用在线仿真器	237	RB0	50
MPLAB ICE 4000 高性能通用在线仿真器	237	RB1	50
MPLAB PM3 器件编程器	237	RB2	50
MPLAB 集成开发环境软件	235	RB3	50
MPLINK 目标链接器 / MPLIB 目标库管理器	236	RB4	51
MSSP	175		
I ² C 模式。参见 I ² C			

PIC16F882/883/884/886/887

RB5	51	串行数据输入	179
RB6	51	从动选择	179
RB7	51	SPI 模式	179
相关的寄存器	52	SPI 时钟	181
引脚说明和框图	50	主控模式	181
PORTB 寄存器	48	SPI 模块	
PORTC	53	从动时序, CKE = 0	183
规范	251	从动时序, CKE = 1	183
P1A。参见增强型捕捉 / 比较 //PWM+ (ECCP+) ...	53	从动同步时序	182
RC0	54	从动选择同步	182
RC1	54	SPI 模式	
RC2	54	从动模式	182
RC3	55	复位的影响	184
RC3 引脚	186	工作原理	179
RC4	55	使能 SPI /IO	180
RC5	55	休眠操作	184
RC6	56	与 SPI 操作相关的寄存器	184
RC7	56	总线模式兼容性	184
相关的寄存器	56	SPI 总线模式	184
PORTC 寄存器	53	SRCON 寄存器	93
PORTD	57	SS	179
P1B/P1C/P1D。参见增强型捕捉 / 比较 //PWM+ (ECCP+)	57	SSPCON2 寄存器	178
RD0、RD1、RD2、RD3 和 RD4	58	SSPCON 寄存器	177
RD5	58	SSPMSK 寄存器	204
RD6	58	SSPOV	194
RD7	58	SSPOV 状态标志	194
相关的寄存器	58	SSPSTAT 寄存器	176
PORTD 寄存器	57	R/W 位	186
PORTE	59	散热考虑	247
RE0	60	闪存程序存储器	111
RE1	60	写操作	117
RE2	60	上电复位 (POR)	209
RE3	60	上电延时定时器 (PWRT)	209
相关的寄存器	60	规范	253
PORTE 寄存器	59	时序参数符号	248
PSTRCON 寄存器	145	时序图	
PWM1CON 寄存器	144	A/D 转换	259
PWM 模式。参见增强型捕捉 / 比较 //PWM	132	A/D 转换 (休眠模式)	259
PWM (ECCP 模块)		半桥 PWM 输出	135, 143
换向同步	147	比较器输出	83
脉冲换向	145	CLKOUT 和 I/O	251
配置位	206	从动同步	182
Q		带有时钟仲裁的波特率发生器	191
启动条件期间的总线冲突	200	发送和应答时的总线冲突	199
器件概述	13	发送间隔字符时序	168
欠压复位 (BOR)	210	复位、WDT、OST 和上电延时定时器	252
规范	253	故障保护时钟监视器 (FSCM)	72
时序和特性	252	I ² C 模块	
相关的寄存器	211	总线冲突发送时序	199
R		I ² C 主控模式第一启动位时序	192
RCREG	156	I ² C 主控模式发送时序	195
RCSTA 寄存器	159	I ² C 主控模式接收时序	196
软件模拟器 (MPLAB SIM)	236	INT 引脚中断	218
S		内部振荡器切换时序	68
SCK	179	PWM 方向更改	138
SDI	179	PWM 输出 (低电平有效)	134
SDO	179	PWM 输出 (高电平有效)	133
SPBRG	161	PWM 自动关闭	
SPBRGH	161	固件重启	142
SPI		使能自动重启	142
串行时钟	179	启动条件期间的总线冲突 (SCL = 0)	201
串行数据输出	179	欠压复位 (BOR)	252
		欠压复位情形	210
		全桥 PWM 输出	137
		SPI 模式时序 (从动模式, CKE = 0)	183

PIC16F882/883/884/886/887

SPI 模式时序 (从动模式, CKE = 1)	183	相关的寄存器	75
SPI 模式时序 (主控模式) SPI 模式		中断	75
主控模式时序图	181	Timer2	
时钟时序	249	相关的寄存器	82
双速启动	70	Timer1	76
Timer0 和 Timer1 外部时钟	254	工作原理	76
Timer1 递增边沿	78	规范	254
停止条件接收或发送	198	时钟源选择	76
停止条件期间的总线冲突	203	Timer1 门控	
同步发送	170	非门	77
同步发送 (通过 TXEN)	170	SR 锁存器	92
同步接收 (主控模式, SREN)	172	时钟源	77
通过中断唤醒	223	同步 COUT w/Timer1	91
休眠模式下的自动唤醒位 (WUE)	167	选择源	91
延时时序		TMR1H 寄存器	76
情形 1	212	TMR1L 寄存器	76
情形 2	212	相关的寄存器	80
情形 3	212	休眠期间的工作原理	78
异步发送	152	异步计数器模式	77
异步发送 (背靠背)	152	读写	77
异步接收	156	预分频器	77
应答序列时序	197	振荡器	77
由 SDA 仲裁引起的 BRG 复位	201	中断	78
在接近 100% 占空比时 PWM 更改方向	139	TRISA	39
增强型捕捉 / 比较 / PWM 时序 (ECCP)	255	TRISA 寄存器	39
正常工作下的自动唤醒位 (WUE)	166	TRISB	47
重复启动条件	193	TRISB 寄存器	48
重复启动条件期间的总线冲突 (情形 1)	202	TRISC	53
重复启动条件期间的总线冲突 (情形 2)	202	TRISC 寄存器	53
主控发送模式下的时钟仲裁	198	TRISD	57
自动波特率校准	165	TRISD 寄存器	57
总线冲突		TRISE	59
启动条件时序	200	TRISE 寄存器	59
使用中唤醒	222	TXREG	151
时钟切换	69	TXSTA 寄存器	158
时钟源		BRGH 位	161
内部模式	65	特殊功能寄存器	22
HFINTOSC	65	特殊事件触发器	102
HFINTOSC/LFINTOSC 切换时序图	67	停止条件期间的总线冲突	203
INTOSC	65	通用文件寄存器	22
INTOSCIO	65	W	
LFINTOSC	67	WCOL	192, 194, 197
频率选择	67	WCOL 状态标志	192, 194, 197
外部模式	63	WDTCON 寄存器	221
EC	63	WPUB 寄存器	49
HS	64	VREF. 参见 ADC 参考电压	
LP	64	WWW 地址	283
OST	63	WWW 在线技术支持	12
RC	65	X	
XT	64	响应时间	85
数据存储器	22	休眠	
数据 EEPROM 存储器	111	唤醒	222
代码保护	120	使用中唤醒	222
的写操作	114	选项寄存器	29
读	114	Y	
相关的寄存器	121	延时时序	211
双速时钟启动模式	69	异步操作时的时钟精确度	158
T		引脚配置说明	
T1CON 寄存器	79	PIC16F883/886	16
T2CON 寄存器	82	PIC16F884/887	18
Timer0	73	引脚图	
工作原理	73	PIC16F883/886, 28 引脚 (QFN)	4
规范	254	PIC16F883/886, 28 引脚 (PDIP、SOIC 和 SSOP)	3
TOCKI	74		
外部时钟	74		

PIC16F882/883/884/886/887

PIC16F884/887, 40 引脚 (PDIP)	6	CLRWDT	228
PIC16F884/887, 44 引脚 (QFN)	8	COMF	228
PIC16F884/887, 44 引脚 (TQFP)	10	DECF	228
因特网地址	283	GOTO	229
预分频器		INCF	229
切换预分频器分配	74	INCFSZ	229
WDT/Timer0 共用	74	IORLW	229
Z		IORWF	229
在线串行编程 (ICSP)	223	MOVF	230
在线调试器	224	MOVLW	230
增强型捕捉 / 比较 / PWM	124	MOVWF	230
增强型捕捉 / 比较 / PWM (ECCP)		NOP	230
增强型 PWM 模式		RETFIE	231
直通电流	143	RETLW	231
增强型捕捉 / 比较 / PWM (ECCP)		RETURN	231
规范	255	RLF	232
增强型 PWM 模式	132	RRF	232
半桥模式	135	SLEEP	232
半桥应用	135	SWAPF	233
半桥应用示例	143	SUBLW	232
可编程死区延时	143	SUBWF	233
启动注意事项	140	XORLW	233
全桥模式	136	XORWF	233
全桥输出模式下的方向更改	138	总表	226
全桥应用	136	直流特性	
输出关系图	134	工业级	243
输出关系 (高电平有效和低电平有效)	133	工业级和扩展级	241, 242, 245
自动关闭	141	扩展级	244
自动重启	142	直通电流	143
增强型捕捉 / 比较器 / PWM (ECCP)		中断	216
Timer 资源	124	ADC	103
增强型通用同步 / 异步收发器 (EUSART)	149	电平变化时中断	47
振荡器		电平变化中断	47
RCIO	61	PORTB 电平变化中断	217
相关的寄存器	72, 80	RB0/INT	216
振荡器参数	250	TMR0	217
振荡器规范	249	TMR1	78
振荡器模块	61	现场保护	219
EC	61	相关的寄存器	218
HFINTOSC	61	重复启动条件期间的总线冲突	202
HS	61	主控模式	189
INTOSC	61	主控模式支持	189
INTOSCIO	61	主控同步串行端口 (MSSP)。参见 MSSP	
LFINTOSC	61	状态寄存器	28
LP	61		
RC	61		
XT	61		
振荡器起振延时定时器 (OST)			
规范	253		
振荡器切换			
故障保护时钟监视	71		
双速时钟启动	69		
指令格式	225		
指令集	225		
ADDLW	227		
ADDWF	227		
ANDLW	227		
ANDWF	227		
BCF	227		
BSF	227		
BTFSC	228		
BTFSS	227		
CALL	228		
CLRF	228		
CLRW	228		

PIC16F882/883/884/886/887

注:

PIC16F882/883/884/886/887

MICROCHIP 网站

Microchip 网站 (www.microchip.com) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的因特网浏览器即可访问。网站提供以下信息:

- **产品支持**——数据手册和勘误表、应用笔记和样本程序、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, 点击“变更通知客户 (Customer Change Notification)”服务后按照注册说明完成注册。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助:

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持

客户应联系其代理商、代表或应用工程师 (FAE) 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 <http://support.microchip.com> 获得网上技术支持。

PIC16F882/883/884/886/887

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: PIC16F882/883/884/886/887 文献编号: DS41291D_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?

PIC16F882/883/884/886/887

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销售办事处联系。

器件编号	X	/XX	XXX
器件	温度范围	封装	模板
器件:	PIC16F883F ⁽¹⁾ , PIC16F883FT ^(1,2) , PIC16F884F ⁽¹⁾ , PIC16F884FT ^(1,2) , PIC16F886F ⁽¹⁾ , PIC16F886FT ^(1,2) , PIC16F887F ⁽¹⁾ , PIC16F887FT ^(1,2) VDD 范围为 2.0V 至 5.5V		
温度范围:	I = -40°C 至 +85°C (工业级) E = -40°C 至 +125°C (扩展级)		
封装:	ML = 正方扁平无引脚封装 (QFN) P = 塑封 DIP PT = 塑封薄型正方扁平封装 (TQFP) SO = 塑封小外型封装 (SOIC) (7.50 mm) SS = 塑封缩小型小外形封装		
模板:	QTP、SQTP、代码或特殊要求 (其他情况空白)		

示例:

a) PIC16F883-E/P 301 = 扩展温度级, PDIP 封装, 20 MHz, QTP 模板 #301

b) PIC16F883-I/SO = 工业温度级, SOIC 封装, 20 MHz

注 1: F = 标准电压范围
注 2: T = 仅卷带式 SSOP、SOIC 和 QFN 封装。

