



# PIC18F85J11 系列 数据手册

采用纳瓦技术的  
64/80 引脚高性能单片机

---

---

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。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. 的英文原本文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利, 它们可能由更新之信息所替代。确保应用符合技术规范, 是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保, 包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和/或生命安全应用, 一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时, 会维护和保障 Microchip 免于承担法律责任, 并加以赔偿。在 Microchip 知识产权保护下, 不得暗中或以其他方式转让任何许可证。

#### 商标

Microchip 的名称和徽标组合、Microchip 徽标、dsPIC、KEELOQ、KEELOQ 徽标、MPLAB、PIC、PICmicro、PICSTART、PIC<sup>32</sup> 徽标、rfPIC 和 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MXDEV、MXLAB、SEEVAL 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、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPLAB Certified 徽标、MPLIB、MPLINK、mTouch、Octopus、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、REAL ICE、rfLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、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。

© 2010, Microchip Technology Inc. 版权所有。

ISBN: 978-1-60932-433-9

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<sup>®</sup> MCU 与 dsPIC<sup>®</sup> DSC、KEELOQ<sup>®</sup> 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品方面的质量体系流程均符合 ISO/TS-16949:2002。此外, Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

## 采用纳瓦技术的 64/80 引脚高性能单片机

### 低功耗特性:

- 功耗管理模式: 运行模式、空闲模式和休眠模式
- 运行模式电流可低至 7  $\mu\text{A}$  (典型值)
- 空闲模式电流可低至 2.5  $\mu\text{A}$  (典型值)
- 休眠模式电流可低至 100 nA (典型值)
- 退出休眠模式时的快速 INTOSC 启动时间为 1  $\mu\text{s}$  (典型值)
- 振荡器双速启动能够减少晶振稳定等待时间

### 外设特点:

- 高灌 / 拉电流 25 mA/25 mA (PORTB 和 PORTC)
- 最多 4 个外部中断
- 4 个 8 位 /16 位定时器 / 计数器模块
- 实时时钟 (Real-Time Clock, RTC) 软件模块:
  - 使用 Timer1 可配置 24 小时时钟、日历、自动 100 年或 12800 年星期日期计算器
- 两个捕捉 / 比较 / PWM (Capture/Compare/PWM, CCP) 模块:
  - 捕捉模块为 16 位, 最大分辨率为 6.25 ns ( $T_{CY}/16$ )
  - 比较模块为 16 位, 最大分辨率为 100 ns ( $T_{CY}$ )
  - PWM 输出: PWM 分辨率最高可达 10 位
- 具有两种工作模式的主同步串口 (Master Synchronous Serial Port, MSSP):
  - 3 线 /4 线 SPI (支持所有 4 种 SPI 模式)
  - I<sup>2</sup>C™ 主 / 从模式
- 一个可寻址 USART 模块
- 一个增强型可寻址 USART 模块:
  - 支持 LIN/J2602
  - 起始位和间隔字符自动唤醒
  - 自动波特率检测 (Auto-Baud Detect, ABD)
- 最多 12 路通道的 10 位 A/D 转换器:
  - 自动采集
  - 休眠期间可进行转换
- 两个模拟比较器
- 比较器的可编程参考电压

### 外部存储器总线 (仅 PIC18F8XJ11):

- 可寻址范围最多 2 MB
- 8 位或 16 位接口
- 12 位、16 位和 20 位寻址模式

### 灵活的振荡器结构:

- 两种晶振模式, 频率为 4-25 MHz
- 两种外部时钟模式, 频率最高可达 40 MHz
- 4 倍频锁相环 (Phase Lock Loop, PLL)
- 内部振荡器模块:
  - 8 个可由用户选择的频率, 从 31.25 kHz 到 8 MHz
- 辅助振荡器使用 Timer1 (工作频率为 32 kHz)
- 故障保护时钟监视器:
  - 外设时钟出现故障时允许安全关闭

### 单片机的特殊性能:

- 可承受 1000 次 (典型值) 擦 / 写的闪存程序存储器
- 闪存的数据保持时间至少为 20 年
- 可在软件控制下自编程
- 中断优先级
- 8 x 8 单周期硬件乘法器
- 扩展的看门狗定时器 (Watchdog Timer, WDT):
  - 可编程周期从 4 ms 到 131s
- 通过两个引脚进行在线串行编程 (In-Circuit Serial Programming™, ICSP™)
- 通过两个引脚进行在线调试
- 工作电压范围: 2.0V 到 3.6V
- 可承受 5.5V 输入电压 (仅限数字引脚)
- 用于串行通信的可选漏极开路配置以及用于将输出最高驱动到 5V 的 CCP 引脚
- 片上 2.5V 稳压器

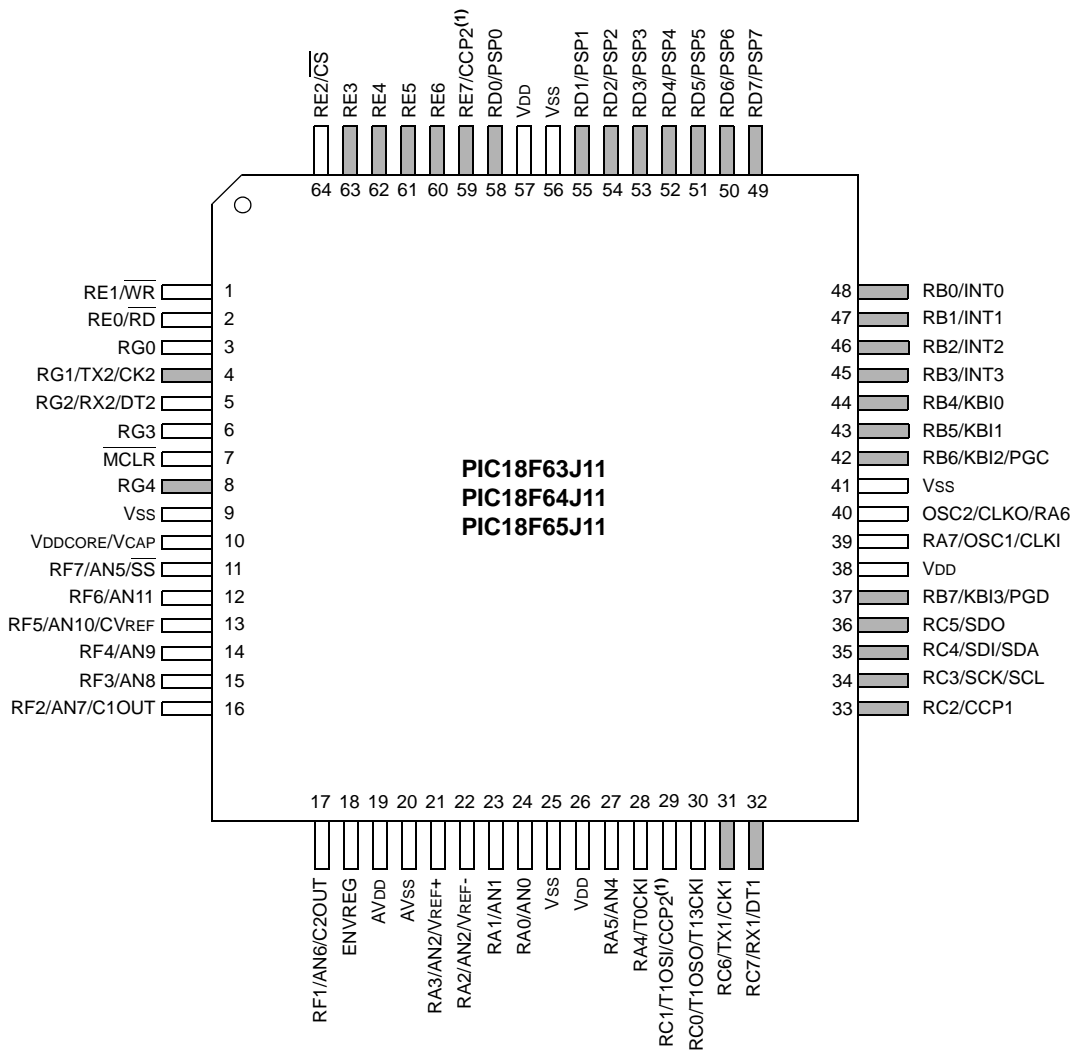
器件	程序存储器		SRAM 数据存储器 (字节)	I/O	8/16 位定时器	CCP	MSSP		EUSART/AUSART	10 位 A/D (通道数)	比较器	BOR/LVD	外部总线	PSP
	闪存 (字节)	单字指令数					SPI	主 I <sup>2</sup> C™						
PIC18F63J11	8K	4096	1K	52	1/3	2	有	有	1/1	12	2	有	无	有
PIC18F64J11	16K	8192	1K	52	1/3	2	有	有	1/1	12	2	有	无	有
PIC18F65J11	32K	16384	2K	52	1/3	2	有	有	1/1	12	2	有	无	有
PIC18F83J11	8K	4096	1K	68	1/3	2	有	有	1/1	12	2	有	有	有
PIC18F84J11	16K	8192	1K	68	1/3	2	有	有	1/1	12	2	有	有	有
PIC18F85J11	32K	16384	2K	68	1/3	2	有	有	1/1	12	2	有	有	有

# PIC18F85J11 系列

## 引脚图

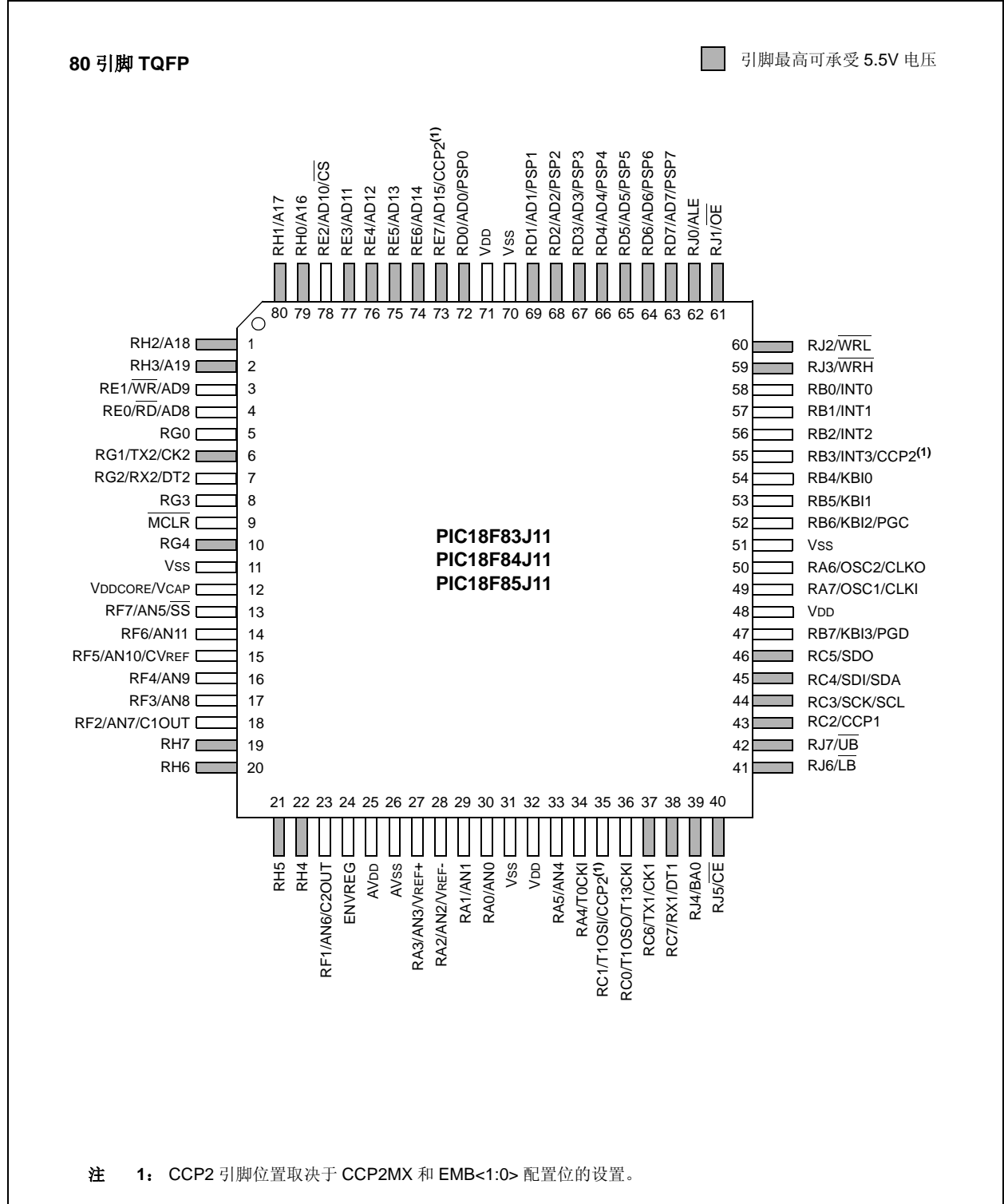
64 引脚 TQFP

引脚最高可承受 5.5V 电压



注 1: CCP2 引脚位置取决于 CCP2MX 配置位的设置。

引脚图 (续)



# PIC18F85J11 系列

---

## 目录

1.0 器件概述 .....	9
2.0 PIC18FJ 单片机入门指南 .....	31
3.0 振荡器配置 .....	35
4.0 功耗管理模式 .....	43
5.0 复位 .....	51
6.0 存储器构成 .....	63
7.0 闪存程序存储器 .....	89
8.0 外部存储器总线 .....	99
9.0 8 x 8 硬件乘法器 .....	111
10.0 中断 .....	113
11.0 I/O 端口 .....	129
12.0 Timer0 模块 .....	153
13.0 Timer1 模块 .....	157
14.0 Timer2 模块 .....	163
15.0 Timer3 模块 .....	165
16.0 捕捉 / 比较 / PWM (CCP) 模块 .....	169
17.0 主同步串行端口 (MSSP) 模块 .....	179
18.0 增强型通用同步 / 异步收发器 (EUSART) .....	223
19.0 可寻址通用同步 / 异步收发器 (AUSART) .....	245
20.0 10 位模数转换器 (A/D) 模块 .....	259
21.0 比较器模块 .....	269
22.0 比较器参考电压模块 .....	275
23.0 CPU 的特殊功能 .....	279
24.0 开发支持 .....	295
25.0 指令集汇总 .....	299
26.0 电气特性 .....	349
27.0 封装信息 .....	387
附录 A: 版本历史 .....	393
附录 B: 高档器件系列之间的移植 .....	393
索引 .....	395
Microchip 网站 .....	405
变更通知客户服务 .....	405
客户支持 .....	405
读者反馈表 .....	406
产品标识体系 .....	407

## 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或建议，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 是文档 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档版本。

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）

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

## 客户通知系统

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

# PIC18F85J11 系列

---

注:



## 1.0 器件概述

该文档包含以下器件的特定信息：

- PIC18F63J11
- PIC18F64J11
- PIC18F65J11
- PIC18F83J11
- PIC18F84J11
- PIC18F85J11

该系列器件在保持极具竞争力的价格的同时，结合了 PIC18 单片机固有的优点——即出色的计算性能和丰富的功能。这些特性使得 PIC18F85J11 系列器件成为许多将价格作为首选因素的高性能应用的明智选择。

## 1.1 内核特性

### 1.1.1 纳瓦技术

PIC18F85J11 系列的所有器件都具有一系列能在工作时显著降低功耗的功能。主要包括以下几项：

- **备用运行模式：**通过将 Timer1 或内部 RC 振荡器作为单片机时钟源，可使代码执行时的功耗大约降低 90%。
- **多种空闲模式：**单片机还可在其 CPU 内核禁止而外设仍然工作的情况下运行。在这些状态下，功耗能降得更低，只有正常工作时的 4%。
- **动态模式切换：**在器件工作期间可由用户代码调用该功耗管理模式，允许用户将节能理念融入到他们的应用软件设计中。

### 1.1.2 振荡器选项和特性

PIC18F85J11 系列的所有器件提供 6 种不同的振荡器选项，使用户在开发应用硬件时有很大的选择范围。这些选项包括：

- 两种晶振模式，即使用晶振或陶瓷谐振器。
- 两种外部时钟模式，提供 4 分频时钟输出选项。
- 一个锁相环（PLL）倍频器，可用于外部振荡器模式，可提供高达 40 MHz 的时钟速率。
- 一个内部振荡器模块，它提供一个 8 MHz 的时钟源（精度为  $\pm 2\%$ ）和一个 INTRC 时钟源（频率大约为 31 kHz，温度和 VDD 变化时频率保持稳定），以及 6 种用户可选择的时钟频率（从 125 kHz 到 4 MHz），因此，共有 8 种时钟频率可供选用。此选项可以空出两个振荡器引脚作为额外的通用 I/O 引脚。

内部振荡器模块提供了一个稳定的参考源，增加了以下功能以使器件更可靠地工作：

- **故障保护时钟监视器：**该选项不停地监视主时钟源，将其与内部振荡器提供的参考信号作比较。如果发生了时钟故障，会切换为内部振荡器，使器件可继续低速工作或安全地关闭。
- **双速启动：**该选项允许器件在上电复位或从休眠模式唤醒时将内部振荡器用作时钟源，直到主时钟源可正常工作为止。

### 1.1.3 存储器选项

PIC18F85J11 系列器件提供了一系列程序存储器选项（从 8 KB 到 32 KB 的代码空间）。程序存储器的闪存单元可反复擦写最多 1000 次。如果不刷新，数据保存期保守地估计在 20 年以上。

PIC18F85J11 系列器件还为动态应用数据提供了足够的空间和最多 2048 字节的数据 RAM。

### 1.1.4 扩展指令集

PIC18F85J11 系列器件在 PIC18 指令集的基础上进行了扩展，添加了 8 条新指令和变址寻址模式。此扩展指令集可以作为一个器件配置选项使能，它是为优化使用高级语言（如 C 语言）开发的重入应用程序代码而特别设计的。

### 1.1.5 外部存储器总线

当 32 KB 存储器不能满足应用要求时，可使用 PIC18F85J11 系列的 80 引脚器件以实现外部存储器总线。这使得单片机的内部程序计数器可寻址最大 2 MB 的存储空间，提供了 8 位器件很少能够提供的访问能力。这使得有更多的存储器选项可供选择，包括：

- 结合使用片上和片外存储器，存储容量最多可达 2 MB
- 使用外部闪存来存储可重新编程的应用程序代码或较大的数据表
- 使用外部 RAM 器件存储大量可变数据

# PIC18F85J11 系列

---

## 1.1.6 易于移植

无论存储器大小如何，所有器件均共享相同的丰富外设集，使得在更改和升级应用时能方便地移植。

整个系列的引脚排列一致也有助于向下一代更大的器件移植。在相同引脚数的器件间（引脚数为 64 和 80）移植，甚至是从 64 引脚器件向 80 引脚器件的移植都是可行的。

PIC18F85J11 系列器件的引脚也同其他 PIC18 通用系列器件（如 PIC18F8720 和 PIC18F8722）的引脚兼容。这为应用的更新换代开拓了新的视野，使开发人员能在保留相同功能集的同时在 Microchip PIC18 系列中选择不同价格的器件。

## 1.2 其他特殊性能

- **通信：**PIC18F85J11 系列具有一系列串行通信外设，包括一个可寻址 USART、一个支持 LIN 规范 LIN/J2602 的独立增强型 USART 和一个具备 SPI 和 I<sup>2</sup>C™（主和从）工作模式的主 SSP（MSSP）模块。
- **CCP 模块：**该系列中的所有器件均包含 2 个捕捉 / 比较 / PWM（CCP）模块。一次最多可能会采用 4 个不同的时基执行不同的操作。
- **10 位 A/D 转换器：**该模块具备可编程采集时间，从而不必在选择通道和启动转换之间等待一个采样周期，因而减少了代码开销。
- **扩展的看门狗定时器（WDT）：**该增强型版本增加了一个 16 位预分频器，使得延时范围得到扩展，在整个工作电压和温度内延时范围保持稳定。延时周期请参见第 26.0 节“电气特性”。

## 1.3 系列中各器件的详细信息

PIC18F85J11 系列器件具有 64 引脚和 80 引脚两种封装形式。图 1-1 和图 1-2 为这两类器件的框图。

这两类器件在以下 4 个方面存在差异：

1. 闪存程序存储器（3 种规格，范围从 PIC18FX3J11 器件的 8 KB 到 PIC18FX5J11 器件的 32 KB）。
2. 数据 RAM（PIC18FX3J11 器件和 PIC18FX4J11 器件的 1024 字节，以及 PIC18FX5J11 器件的 2048 字节）。
3. I/O 端口（64 引脚器件有 7 个双向端口，80 引脚器件有 9 个双向端口）。
4. 外部存储器总线（仅在 80 引脚器件中实现）

该系列器件的其他功能都是相同的。表 1-1 和表 1-2 汇总了这些功能。

表 1-3 和表 1-4 列出了所有器件的引脚说明。

**表 1-1: PIC18F6XJ11 系列器件（64 引脚器件）的特性**

特性	PIC18F63J11	PIC18F64J11	PIC18F65J11
工作频率	DC – 40 MHz		
程序存储器（字节）	8K	16K	32K
程序存储器（指令）	4096	8192	16384
数据存储器（字节）	1024	1024	2048
中断源	27		
I/O 端口	端口 A、B、C、D、E、F 和 G		
定时器	4		
捕捉 / 比较 / PWM 模块	2		
串行通信	MSSP、可寻址 USART 和增强型 USART		
并行通信（PSP）	有		
外部存储器总线	无		
10 位模数转换模块	12 路输入通道		
复位（和延时）	POR、BOR、RESET 指令、堆栈满、堆栈下溢、 $\overline{\text{MCLR}}$ 和 WDT（PWRT 和 OST）		
指令集	75 条指令，使能扩展指令集后总共为 83 条		
封装	64 引脚 TQFP		

**表 1-2: PIC18F8XJ11 系列器件（80 引脚器件）的特性**

特性	PIC18F83J11	PIC18F84J11	PIC18F85J11
工作频率	DC – 40 MHz		
程序存储器（字节）	8K	16K	32K
程序存储器（指令）	4096	8192	16384
数据存储器（字节）	1024	1024	2048
中断源	27		
I/O 端口	端口 A、B、C、D、E、F、G、H 和 J		
定时器	4		
捕捉 / 比较 / PWM 模块	2		
串行通信	MSSP、可寻址 USART 和增强型 USART		
并行通信（PSP）	有		
外部存储器总线	有		
10 位模数转换模块	12 路输入通道		
复位（和延时）	POR、BOR、RESET 指令、堆栈满、堆栈下溢、 $\overline{\text{MCLR}}$ 和 WDT（PWRT 和 OST）		
指令集	75 条指令，使能扩展指令集后总共为 83 条		
封装	80 引脚 TQFP		

# PIC18F85J11 系列

图 1-1: PIC18F6XJ11 (64 引脚) 框图

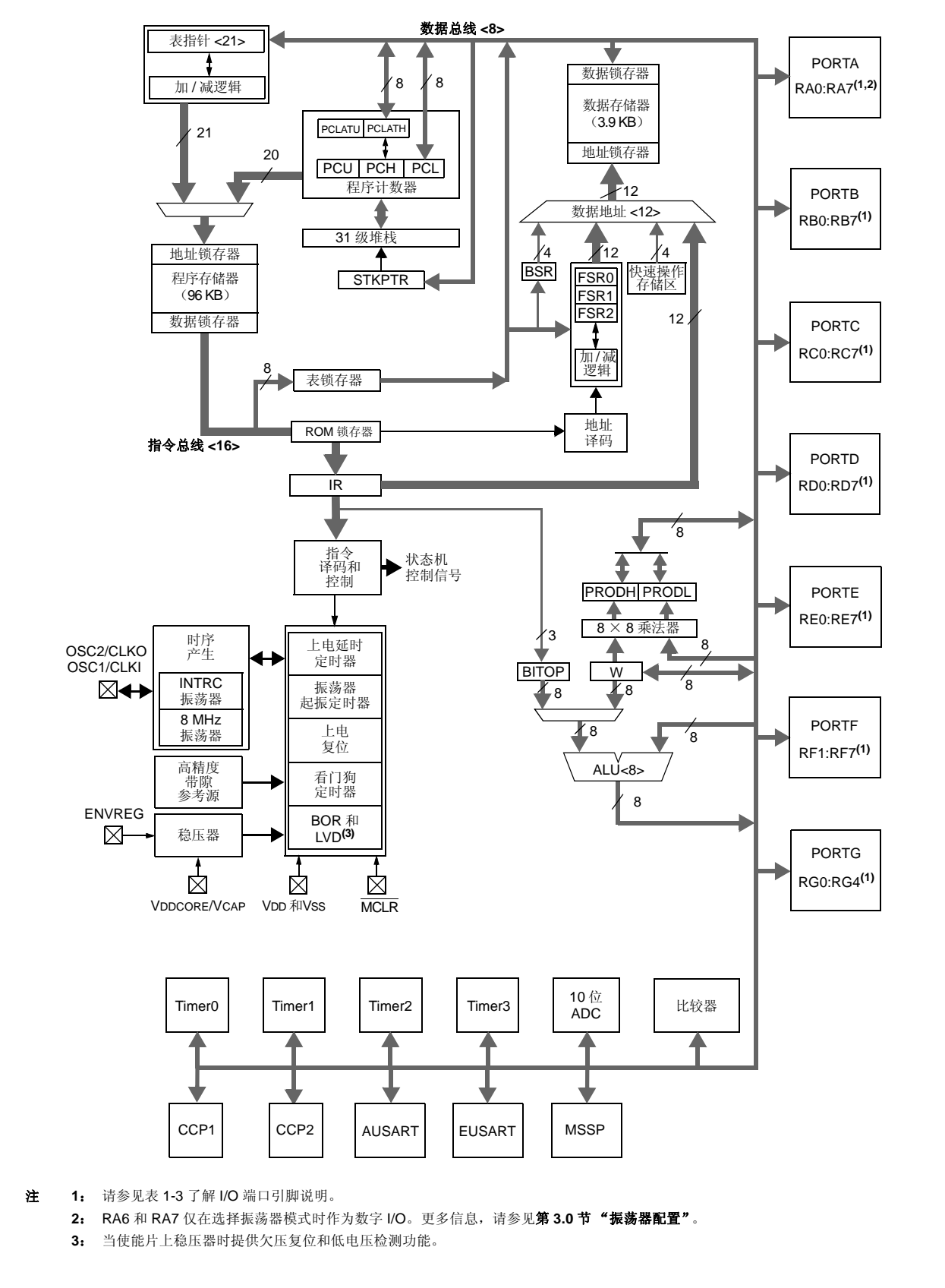
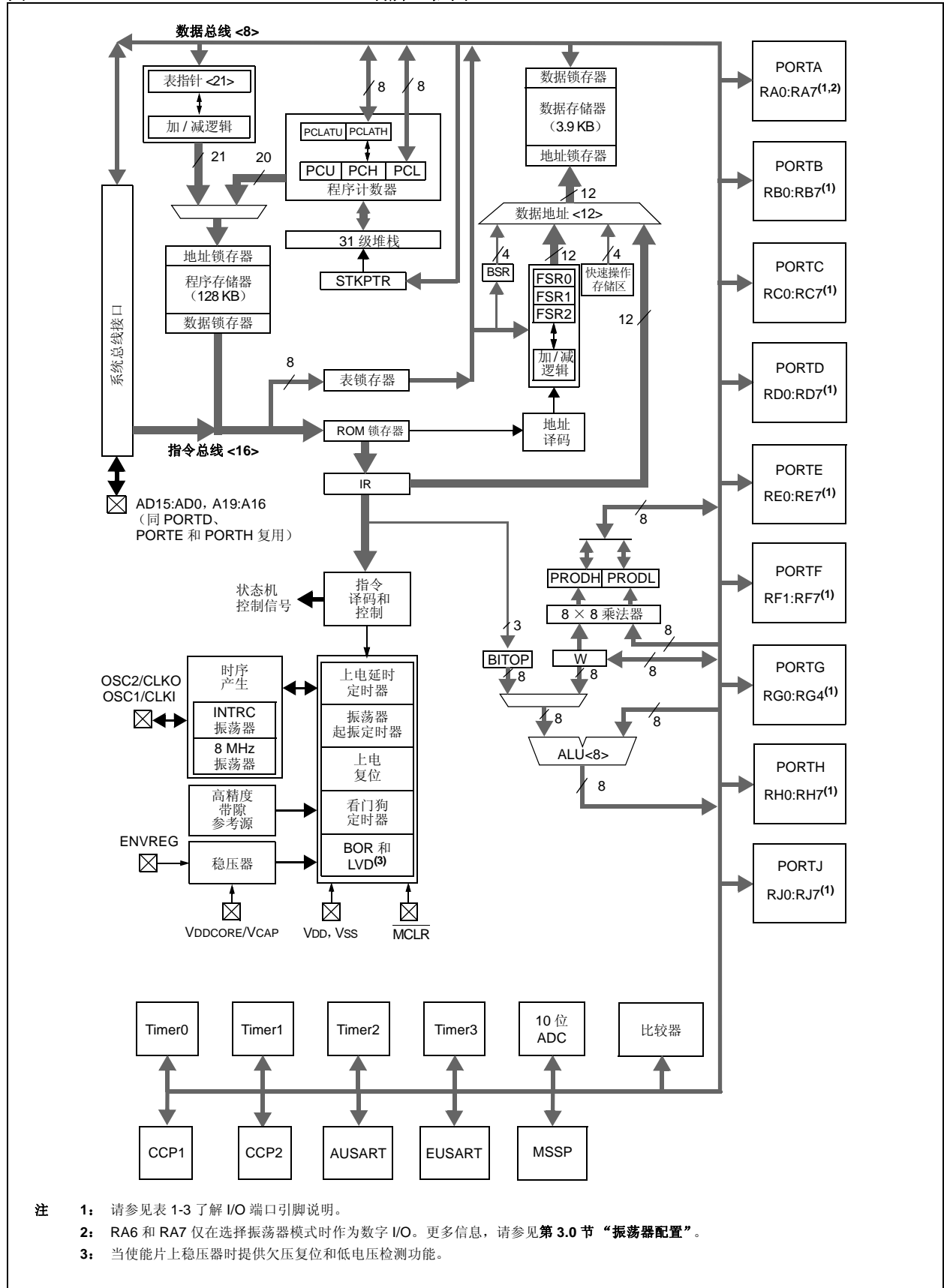


图 1-2: PIC18F8XJ11 (80 引脚) 框图



---

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
MCLR	7	I	ST	主复位（输入）或编程电压（输入）。此引脚为低电平时器件复位。
RA7/OSC1/CLKI RA7 OSC1 CLKI	39	I/O I I	TTL CMOS CMOS	振荡器晶振或外部时钟输入。 通用 I/O 引脚。 振荡器晶振输入。 外部时钟源输入。总是与 OSC1 引脚功能复用。（见相关的 OSC1/CLKI 和 OSC2/CLKO 引脚。）
RA6/OSC2/CLKO RA6 OSC2 CLKO	40	I/O O O	TTL — —	振荡器晶振或时钟输出。 通用 I/O 引脚。 振荡器晶振输出。在晶振模式下，与晶振和谐振器相连。 在 EC 模式下，OSC2 引脚输出 CLKO 信号，该信号是 OSC1 引脚上信号的 4 分频，其频率用于表示指令周期的快慢。
RA0/AN0 RA0 AN0	24	I/O I	TTL Analog	PORTA 是双向 I/O 端口。  数字 I/O。 模拟输入 0。
RA1/AN1 RA1 AN1	23	I/O I	TTL Analog	
RA2/AN2/VREF- RA2 AN2 VREF-	22	I/O I I	TTL Analog Analog	
RA3/AN3/VREF+ RA3 AN3 VREF+	21	I/O I I	TTL Analog Analog	
RA4/T0CKI RA4 T0CKI	28	I/O I	ST/OD ST	
RA5/AN4 RA5 AN4	27	I/O I	TTL Analog	
RA6				
RA7				参见 OSC2/CLKO/RA6 引脚。 参见 OSC1/CLKI/RA7 引脚。

**注**

- 1:** 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。
- 2:** 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。

**表 1-3: PIC18F6XJ11 器件 I/O 引脚说明 (续)**

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
RB0/INT0 RB0 INT0	48	I/O I	TTL ST	PORTB 是双向 I/O 端口。PORTB 在所有的输入端都可以软件编程为内部弱上拉。  数字 I/O。 外部中断 0。
RB1/INT1 RB1 INT1	47	I/O I	TTL ST	数字 I/O。 外部中断 1。
RB2/INT2 RB2 INT2	46	I/O I	TTL ST	数字 I/O。 外部中断 2。
RB3/INT3 RB3 INT3	45	I/O I	TTL ST	数字 I/O。 外部中断 3。
RB4/KBI0 RB4 KBI0	44	I/O I	TTL TTL	数字 I/O。 电平变化中断引脚。
RB5/KBI1 RB5 KBI1	43	I/O I	TTL TTL	数字 I/O。 电平变化中断引脚。
RB6/KBI2/PGC RB6 KBI2 PGC	42	I/O I I/O	TTL TTL ST	数字 I/O。 电平变化中断引脚。 在线调试器和 ICSP™ 编程时钟引脚。
RB7/KBI3/PGD RB7 KBI3 PGD	37	I/O I I/O	TTL TTL ST	数字 I/O。 电平变化中断引脚。 在线调试器和 ICSP 编程数据引脚。

图注:	TTL = TTL 兼容输入	CMOS = CMOS 兼容输入或输出
	ST = CMOS 电平的施密特触发器输入	Analog = 模拟输入
	I = 输入	O = 输出
	P = 电源	OD = 漏极开路 (无 P 型二极管连接到 VDD)
	I <sup>2</sup> C <sup>TM</sup> = I <sup>2</sup> C/SMBus	

**注 1:** 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。  
**注 2:** 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。

# PIC18F85J11 系列

表 1-3: PIC18F6XJ11 器件 I/O 引脚说明 (续)

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
RC0/T1OSO/T13CKI	30	I/O	ST	PORTC 是双向 I/O 端口。
RC0		O	—	数字 I/O。
T1OSO		I	ST	Timer1 振荡器输出。
T13CKI				Timer1/Timer3 外部时钟输入。
RC1/T1OSI/CCP2	29	I/O	ST	数字 I/O。
RC1		I	CMOS	Timer1 振荡器输入。
T1OSI		I/O	ST	捕捉 2 输入 / 比较 2 输出 / PWM2 输出。
CCP2 <sup>(1)</sup>				
RC2/CCP1	33	I/O	ST	数字 I/O。
RC2		I/O	ST	捕捉 1 输入 / 比较 1 输出 / PWM1 输出。
CCP1				
RC3/SCK/SCL	34	I/O	ST	数字 I/O。
RC3		I/O	ST	SPI 模式的同步串行时钟输入 / 输出
SCK		I/O	ST	I <sup>2</sup> C™ 模式的同步串行时钟输入 / 输出
SCL		I/O	I <sup>2</sup> C	
RC4/SDI/SDA	35	I/O	ST	数字 I/O。
RC4		I	ST	SPI 数据输入。
SDI		I/O	I <sup>2</sup> C	I <sup>2</sup> C 数据 I/O。
SDA				
RC5/SDO	36	I/O	ST	数字 I/O。
RC5		O	—	SPI 数据输出。
SDO				
RC6/TX1/CK1	31	I/O	ST	数字 I/O。
RC6		O	—	EUSART 异步发送。
TX1		I/O	ST	EUSART 同步时钟 (见相关的 RX1/DT1 引脚信息)。
CK1				
RC7/RX1/DT1	32	I/O	ST	数字 I/O。
RC7		I	ST	EUSART 异步接收。
RX1		I/O	ST	EUSART 同步数据 (见相关的 TX1/CK1 引脚信息)。
DT1				

图注: TTL = TTL 兼容输入  
ST = CMOS 电平的施密特触发器输入  
I = 输入  
P = 电源  
I<sup>2</sup>C™ = I<sup>2</sup>C/SMBus  
CMOS = CMOS 兼容输入或输出  
Analog = 模拟输入  
O = 输出  
OD = 漏极开路 (无 P 型二极管连接到 V<sub>DD</sub>)

- 注 1: 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。  
2: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。



表 1-3: PIC18F6XJ11 器件 I/O 引脚说明 (续)

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
RD0/PSP0 RD0 PSP0	58	I/O I/O	ST TTL	PORTA 是双向 I/O 端口。  数字 I/O。 并行从端口数据。
RD1/PSP1 RD1 PSP1	55	I/O I/O	ST TTL	数字 I/O。 并行从端口数据。
RD2/PSP2 RD2 PSP2	54	I/O I/O	ST TTL	数字 I/O。 并行从端口数据。
RD3/PSP3 RD3 PSP3	53	I/O I/O	ST TTL	数字 I/O。 并行从端口数据。
RD4/PSP4 RD4 PSP4	52	I/O I/O	ST TTL	数字 I/O。 并行从端口数据。
RD5/PSP5 RD5 PSP5	51	I/O I/O	ST TTL	数字 I/O。 并行从端口数据。
RD6/PSP6 RD6 PSP6	50	I/O I/O	ST TTL	数字 I/O。 并行从端口数据。
RD7/PSP7 RD7 PSP7	49	I/O I/O	ST TTL	数字 I/O。 并行从端口数据。

图注: TTL = TTL 兼容输入  
ST = CMOS 电平的施密特触发器输入  
I = 输入  
P = 电源  
I<sup>2</sup>C<sup>TM</sup> = I<sup>2</sup>C/SMBus  
CMOS = CMOS 兼容输入或输出  
Analog = 模拟输入  
O = 输出  
OD = 漏极开路 (无 P 型二极管连接到 V<sub>DD</sub>)

- 注 1: 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。  
2: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。

# PIC18F85J11 系列

**表 1-3: PIC18F6XJ11 器件 I/O 引脚说明 (续)**

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
RE0/ $\overline{\text{RD}}$ RE0 RD	2	I/O I	ST TTL	<p>PORTE 是双向 I/O 端口。</p> <p>数字 I/O。 并行从端口读控制。</p>
RE1/ $\overline{\text{WR}}$ RE1 WR	1	I/O I	ST TTL	<p>数字 I/O。 并行从端口写控制。</p>
RE2/ $\overline{\text{CS}}$ RE2 CS	64	I/O I	ST TTL	<p>数字 I/O。 并行从端口片选控制。</p>
RE3	63	I/O	ST	数字 I/O。
RE4	62	I/O	ST	数字 I/O。
RE5	61	I/O	ST	数字 I/O。
RE6	60	I/O	ST	数字 I/O。
RE7/CCP2 RE7 CCP2 <sup>(2)</sup>	59	I/O I/O	ST ST	<p>数字 I/O。 捕捉 2 输入 / 比较 2 输出 /PWM2 输出。</p>

图注:	TTL = TTL 兼容输入	CMOS = CMOS 兼容输入或输出
	ST = CMOS 电平的施密特触发器输入	Analog = 模拟输入
	I = 输入	O = 输出
	P = 电源	OD = 漏极开路 (无 P 型二极管连接到 VDD)
	I <sup>2</sup> C <sup>TM</sup> = I <sup>2</sup> C/SMBus	

**注 1:** 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。  
**2:** 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。

表 1-3: PIC18F6XJ11 器件 I/O 引脚说明 (续)

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
RF1/AN6/C2OUT RF1 AN6 C2OUT	17	I/O I O	ST Analog —	PORTF 是双向 I/O 端口。  数字 I/O。 模拟输入 6。 比较器 2 输出。
RF2/AN7/C1OUT RF2 AN7 C1OUT	16	I/O I O	ST Analog —	
RF3/AN8 RF3 AN8	15	I/O I	ST Analog	
RF4/AN9 RF4 AN9	14	I/O I	ST Analog	
RF5/AN10/CVREF RF5 AN10 CVREF	13	I/O I O	ST Analog Analog	
RF6/AN11 RF6 AN11	12	I/O I	ST Analog	
RF7/AN5/SS RF7 AN5 SS	11	I/O O I	ST Analog TTL	

图注: TTL = TTL 兼容输入  
ST = CMOS 电平的施密特触发器输入  
I = 输入  
P = 电源  
I<sup>2</sup>C™ = I<sup>2</sup>C/SMBus  
CMOS = CMOS 兼容输入或输出  
Analog = 模拟输入  
O = 输出  
OD = 漏极开路 (无 P 型二极管连接到 VDD)

注 1: 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。  
2: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。

\_\_\_\_\_

--	--	--	--	--

引脚名称	引脚号	缓冲器类型	说明	
	TQFP			
RG0	3	I/O	ST	PORTG 是双向 I/O 端口。 数字 I/O。
RG1/TX2/CK2	4	I/O	ST	数字 I/O。
RG1		O	—	AUSART 异步发送。
TX2		I/O	ST	AUSART 同步时钟 （见相关的 RX2/DT2 引脚信息）。
CK2				
RG2/RX2/DT2	5	I/O	ST	数字 I/O。
RG2		I	ST	AUSART 异步接收。
RX2		I/O	ST	AUSART 同步数据 （见相关的 TX2/CK2 引脚信息）。
DT2				
RG3	6	I/O	ST	数字 I/O。
RG4	8	I/O	ST	数字 I/O。
VSS	9, 25, 41, 56	P	—	逻辑电路和 I/O 引脚的参考地。
VDD	26, 38, 57	P	—	逻辑电路和 I/O 引脚的正电源。
AVSS	20	P	—	模拟模块的参考地。
AVDD	19	P	—	模拟模块的正电源。
ENVREG	18	I	ST	使能片内稳压器。
VDDCORE/VCAP	10			内核逻辑电源或外部滤波电容连接端。
VDDCORE		P	—	单片机内核逻辑正电源 （禁止稳压器）。
VCAP		P	—	外部滤波电容连接 （使能稳压器）。

图注:	TTL = TTL 兼容输入	CMOS = CMOS 兼容输入或输出
	ST = CMOS 电平的施密特触发器输入	Analog = 模拟输入
	I = 输入	O = 输出
	P = 电源	OD = 漏极开路 (无 P 型二极管连接到 VDD)
	I <sup>2</sup> C <sup>TM</sup> = I <sup>2</sup> C/SMBus	

**注 1:** 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。  
**注 2:** 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。

**表 1-4: PIC18F8XJ11 器件 I/O 引脚说明**

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
MCLR	9	I	ST	主复位（输入）或编程电压（输入）。此引脚为低电平时器件复位。
RA7/OSC1/CLKI RA7 OSC1 CLKI	49	I/O I I	TTL CMOS CMOS	振荡器晶振或外部时钟输入。 通用 I/O 引脚。 振荡器晶振输入。 外部时钟源输入。总是与 OSC1 引脚功能复用。（见相关的 OSC1/CLKI 和 OSC1/CLKO 引脚。）
RA6/OSC2/CLKO RA6 OSC2 CLKO	50	I/O O O	TTL — —	振荡器晶振或时钟输出。 通用 I/O 引脚。 振荡器晶振输出。在晶振模式下，与晶振或谐振器相连。 在 EC 模式下，OSC2 引脚输出 CLKO 信号，该信号是 OSC1 引脚上信号的 4 分频，其频率用于表示指令周期的快慢。
RA0/AN0 RA0 AN0	30	I/O I	TTL Analog	PORTA 是双向 I/O 端口。  数字 I/O。 模拟输入 0。
RA1/AN1 RA1 AN1	29	I/O I	TTL Analog	
RA2/AN2/VREF- RA2 AN2 VREF-	28	I/O I I	TTL Analog Analog	
RA3/AN3/VREF+ RA3 AN3 VREF+	27	I/O I I	TTL Analog Analog	
RA4/T0CKI RA4 T0CKI	34	I/O I	ST/OD ST	
RA5/AN4 RA5 AN4	33	I/O I	TTL Analog	
RA6				
RA7				参见 OSC2/CLKO/RA6 引脚。  参见 OSC1/CLKI/RA7 引脚。

图注:	TTL = TTL 兼容输入	CMOS = CMOS 兼容输入或输出
	ST = CMOS 电平的施密特触发器输入	Analog = 模拟输入
	I = 输入	O = 输出
	P = 电源	OD = 漏极开路 (无 P 型二极管连接到 VDD)
	I <sup>2</sup> C <sup>TM</sup> = I <sup>2</sup> C/SMBus	

- 注 1:** 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配 (仅限 80 引脚器件的扩展单片机模式)。
- 注 2:** 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。
- 注 3:** 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。

# PIC18F85J11 系列

表 1-4: PIC18F8XJ11 器件 I/O 引脚说明 (续)

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
RB0/INT0 RB0 INT0	58	I/O I	TTL ST	PORTB 是双向 I/O 端口。PORTB 在所有的输入端都可以软件编程为内部弱上拉。  数字 I/O。 外部中断 0。
RB1/INT1 RB1 INT1	57	I/O I	TTL ST	数字 I/O。 外部中断 1。
RB2/INT2 RB2 INT2	56	I/O I	TTL ST	数字 I/O。 外部中断 2。
RB3/INT3/CCP2 RB3 INT3 CCP2 <sup>(1)</sup>	55	I/O I I/O	TTL ST ST	数字 I/O。 外部中断 3。 捕捉 2 输入 / 比较 2 输出 / PWM2 输出。
RB4/KBI0 RB4 KBI0	54	I/O I	TTL TTL	数字 I/O。 电平变化中断引脚。
RB5/KBI1 RB5 KBI1	53	I/O I	TTL TTL	数字 I/O。 电平变化中断引脚。
RB6/KBI2/PGC RB6 KBI2 PGC	52	I/O I I/O	TTL TTL ST	数字 I/O。 电平变化中断引脚。 在线调试器和 ICSP™ 编程时钟引脚。
RB7/KBI3/PGD RB7 KBI3 PGD	47	I/O I I/O	TTL TTL ST	数字 I/O。 电平变化中断引脚。 在线调试器和 ICSP 编程数据引脚。

图注: TTL = TTL 兼容输入  
ST = CMOS 电平的施密特触发器输入  
I = 输入  
P = 电源  
I<sup>2</sup>C™ = I<sup>2</sup>C/SMBus  
CMOS = CMOS 兼容输入或输出  
Analog = 模拟输入  
O = 输出  
OD = 漏极开路 (无 P 型二极管连接到 VDD)

- 注 1: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配 (仅限 80 引脚器件的扩展单片机模式)。  
2: 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。  
3: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。

表 1-4: PIC18F8XJ11 器件 I/O 引脚说明 (续)

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
RC0/T1OSO/T13CKI RC0 T1OSO T13CKI	36	I/O O I	ST — ST	PORTC 是双向 I/O 端口。  数字 I/O。 Timer1 振荡器输出。 Timer1/Timer3 外部时钟输入。
RC1/T1OSI/CCP2 RC1 T1OSI CCP2 <sup>(2)</sup>	35	I/O I I/O	ST CMOS ST	数字 I/O。 Timer1 振荡器输入。 捕捉 2 输入 / 比较 2 输出 / PWM2 输出。
RC2/CCP1 RC2 CCP1	43	I/O I/O	ST ST	数字 I/O。 捕捉 1 输入 / 比较 1 输出 / PWM1 输出。
RC3/SCK/SCL RC3 SCK SCL	44	I/O I/O I/O	ST ST I <sup>2</sup> C	数字 I/O。 SPI 模式的同步串行时钟输入 / 输出。 I <sup>2</sup> C™ 模式的同步串行时钟输入 / 输出。
RC4/SDI/SDA RC4 SDI SDA	45	I/O I I/O	ST ST I <sup>2</sup> C	数字 I/O。 SPI 数据输入。 I <sup>2</sup> C 数据 I/O。
RC5/SDO RC5 SDO	46	I/O O	ST —	数字 I/O。 SPI 数据输出。
RC6/TX1/CK1 RC6 TX1 CK1	37	I/O O I/O	ST — ST	数字 I/O。 EUSART 异步发送。 EUSART 同步时钟 (见相关的 RX1/DT1 引脚信息)。
RC7/RX1/DT1 RC7 RX1 DT1	38	I/O I I/O	ST ST ST	数字 I/O。 EUSART 异步接收。 EUSART 同步数据 (见相关的 TX1/CK1 引脚信息)。

图注: TTL = TTL 兼容输入  
ST = CMOS 电平的施密特触发器输入  
I = 输入  
P = 电源  
I<sup>2</sup>C™ = I<sup>2</sup>C/SMBus  
CMOS = CMOS 兼容输入或输出  
Analog = 模拟输入  
O = 输出  
OD = 漏极开路 (无 P 型二极管连接到 VDD)

- 注 1: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配 (仅限 80 引脚器件的扩展单片机模式)。  
2: 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。  
3: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。

# PIC18F85J11 系列

表 1-4: PIC18F8XJ11 器件 I/O 引脚说明 (续)

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
RD0/AD0/PSP0 RD0 AD0 PSP0	72	I/O I/O I/O	ST TTL TTL	PORTA 是双向 I/O 端口。  数字 I/O。 外部存储器地址 / 数据 0。 并行从端口数据。
RD1/AD1/PSP1 RD1 AD1 PSP1	69	I/O I/O I/O	ST TTL TTL	数字 I/O。 外部存储器地址 / 数据 1。 并行从端口数据。
RD2/AD2/PSP2 RD2 AD2 PSP2	68	I/O I/O I/O	ST TTL TTL	数字 I/O。 外部存储器地址 / 数据 2。 并行从端口数据。
RD3/AD3/PSP3 RD3 AD3 PSP3	67	I/O I/O I/O	ST TTL TTL	数字 I/O。 外部存储器地址 / 数据 3。 并行从端口数据。
RD4/AD4/PSP4 RD4 AD4 PSP4	66	I/O I/O I/O	ST TTL TTL	数字 I/O。 外部存储器地址 / 数据 4。 并行从端口数据。
RD5/AD5/PSP5 RD5 AD5 PSP5	65	I/O I/O I/O	ST TTL TTL	数字 I/O。 外部存储器地址 / 数据 5。 并行从端口数据。
RD6/AD6/PSP6 RD6 AD6 PSP6	64	I/O I/O I/O	ST TTL TTL	数字 I/O。 外部存储器地址 / 数据 6。 并行从端口数据。
RD7/AD7/PSP7 RD7 AD7 PSP7	63	I/O I/O I/O	ST TTL TTL	数字 I/O。 外部存储器地址 / 数据 7。 并行从端口数据。

图注: TTL = TTL 兼容输入  
ST = CMOS 电平的施密特触发器输入  
I = 输入  
P = 电源  
I<sup>2</sup>C™ = I<sup>2</sup>C/SMBus  
CMOS = CMOS 兼容输入或输出  
Analog = 模拟输入  
O = 输出  
OD = 漏极开路 (无 P 型二极管连接到 V<sub>DD</sub>)

- 注 1: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配 (仅限 80 引脚器件的扩展单片机模式)。  
2: 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。  
3: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。



表 1-4: PIC18F8XJ11 器件 I/O 引脚说明 (续)

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
RE0/ $\overline{\text{RD}}$ /AD8	4	I/O I I/O	ST TTL TTL	PORTC 是双向 I/O 端口。  数字 I/O。 并行从端口读控制。 外部存储器地址 / 数据 8。
RE1/ $\overline{\text{WR}}$ /AD9	3	I/O I I/O	ST TTL TTL	数字 I/O。 并行从端口写控制。 外部存储器地址 / 数据 9。
RE2/AD10/ $\overline{\text{CS}}$	78	I/O I/O I	ST TTL TTL	数字 I/O。 外部存储器地址 / 数据 10。 并行从端口片选控制。
RE3/AD11	77	I/O I/O	ST TTL	数字 I/O。 外部存储器地址 / 数据 11。
RE4/AD12	76	I/O I/O	ST TTL	数字 I/O。 外部存储器地址 / 数据 12。
RE5/AD13	75	I/O I/O	ST TTL	数字 I/O。 外部存储器地址 / 数据 13。
RE6/AD14	74	I/O I/O	ST TTL	数字 I/O。 外部存储器地址 / 数据 14。
RE7/AD15/CCP2	73	I/O I/O I/O	ST TTL ST	数字 I/O。 外部存储器地址 / 数据 15。 捕捉 2 输入 / 比较 2 输出 / PWM2 输出。

图注: TTL = TTL 兼容输入  
ST = CMOS 电平的施密特触发器输入  
I = 输入  
P = 电源  
I<sup>2</sup>C<sup>TM</sup> = I<sup>2</sup>C/SMBus  
CMOS = CMOS 兼容输入或输出  
Analog = 模拟输入  
O = 输出  
OD = 漏极开路 (无 P 型二极管连接到 VDD)

- 注 1: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配 (仅限 80 引脚器件的扩展单片机模式)。  
2: 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。  
3: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。

# PIC18F85J11 系列

表 1-4: PIC18F8XJ11 器件 I/O 引脚说明 (续)

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
RF1/AN6/C2OUT	23	I/O	ST	PORTF 是双向 I/O 端口。
RF1		I	Analog	
AN6		O	—	
C2OUT				
RF2/AN7/C1OUT	18	I/O	ST	
RF2		I	Analog	
AN7		O	—	
C1OUT				
RF3/AN8	17	I/O	ST	数字 I/O。
RF3		I	Analog	
AN8				
RF4/AN9	16	I/O	ST	
RF4		I	Analog	
AN9				
RF5/AN10/CVREF	15	I/O	ST	
RF5		I	Analog	
AN10		O	Analog	
CVREF				
RF6/AN11	14	I/O	ST	数字 I/O。
RF6		I	Analog	
AN11				
RF7/AN5/SS	13	I/O	ST	
RF7		O	Analog	
AN5		I	TTL	
SS				

图注: TTL = TTL 兼容输入 CMOS = CMOS 兼容输入或输出  
ST = CMOS 电平的施密特触发器输入 Analog = 模拟输入  
I = 输入 O = 输出  
P = 电源 OD = 漏极开路 (无 P 型二极管连接到 VDD)  
I<sup>2</sup>C™ = I<sup>2</sup>C/SMBus

- 注 1: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配 (仅限 80 引脚器件的扩展单片机模式)。  
2: 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。  
3: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。

表 1-4: PIC18F8XJ11 器件 I/O 引脚说明 (续)

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
RG0	5	I/O	ST	PORTG 是双向 I/O 端口。  数字 I/O。  数字 I/O。 AUSART 异步发送。 AUSART 同步时钟 (见相关的 RX2/DT2 引脚信息)。  数字 I/O。 AUSART 异步接收。 AUSART 同步数据 (见相关的 TX2/CK2 引脚信息)。  数字 I/O。  数字 I/O。
RG1/TX2/CK2	6	I/O	ST	
RG1		I/O	ST	
TX2		O	—	
CK2		I/O	ST	
RG2/RX2/DT2	7	I/O	ST	
RG2		I/O	ST	
RX2		I	ST	
DT2		I/O	ST	
RG3	8	I/O	ST	
RG4	10	I/O	ST	

图注: TTL = TTL 兼容输入      CMOS = CMOS 兼容输入或输出  
ST = CMOS 电平的施密特触发器输入      Analog = 模拟输入  
I = 输入      O = 输出  
P = 电源      OD = 漏极开路 (无 P 型二极管连接到 VDD)  
I<sup>2</sup>C™ = I<sup>2</sup>C/SMBus

- 注 1: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配 (仅限 80 引脚器件的扩展单片机模式)。  
2: 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。  
3: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。

# PIC18F85J11 系列

**表 1-4: PIC18F8XJ11 器件 I/O 引脚说明 (续)**

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
RH0/A16 RH0 A16	79	I/O I/O	ST TTL	PORTH 是双向 I/O 端口。  数字 I/O。 外部存储器地址 / 数据 16。
RH1/A17 RH1 A17	80	I/O I/O	ST TTL	数字 I/O。 外部存储器地址 / 数据 17。
RH2/A18 RH2 A18	1	I/O I/O	ST TTL	数字 I/O。 外部存储器地址 / 数据 18。
RH3/A19 RH3 A19	2	I/O I/O	ST TTL	数字 I/O。 外部存储器地址 / 数据 19。
RH4	22	I/O	ST	数字 I/O。
RH5	21	I/O	ST	数字 I/O。
RH6	20	I/O	ST	数字 I/O。
RH7	19	I/O	ST	数字 I/O。

图注:

TTL	= TTL 兼容输入	CMOS	= CMOS 兼容输入或输出
ST	= CMOS 电平的施密特触发器输入	Analog	= 模拟输入
I	= 输入	O	= 输出
P	= 电源	OD	= 漏极开路 (无 P 型二极管连接到 VDD)
I <sup>2</sup> C™	= I <sup>2</sup> C/SMBus		

- 注 1: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配 (仅限 80 引脚器件的扩展单片机模式)。  
2: 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。  
3: 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。

**表 1-4: PIC18F8XJ11 器件 I/O 引脚说明 (续)**

引脚名称	引脚号	引脚类型	缓冲器类型	说明
	TQFP			
RJ0/ALE RJ0 ALE	62	I/O O	ST —	PORTJ 是双向 I/O 端口。  数字 I/O。 外部存储器地址锁存器使能引脚。
RJ1/ $\overline{\text{OE}}$ RJ1 $\overline{\text{OE}}$	61	I/O O	ST —	数字 I/O。 外部存储器输出使能引脚。
RJ2/ $\overline{\text{WRL}}$ RJ2 $\overline{\text{WRL}}$	60	I/O O	ST —	数字 I/O。 外部存储器写低地址空间控制引脚。
RJ3/ $\overline{\text{WRH}}$ RJ3 $\overline{\text{WRH}}$	59	I/O O	ST —	数字 I/O。 外部存储器写高地址空间控制引脚。
RJ4/BA0 RJ4 BA0	39	I/O O	ST —	数字 I/O。 外部存储器字节地址 0 控制引脚。
RJ5/ $\overline{\text{CE}}$ RJ5 $\overline{\text{CE}}$	40	I/O O	ST —	数字 I/O。 外部存储器片选使能控制引脚。
RJ6/ $\overline{\text{LB}}$ RJ6 $\overline{\text{LB}}$	41	I/O O	ST —	数字 I/O。 外部存储器低字节控制引脚。
RJ7/ $\overline{\text{UB}}$ RJ7 $\overline{\text{UB}}$	42	I/O O	ST —	数字 I/O。 外部存储器高字节控制引脚。
VSS	11, 31, 51, 70	P	—	逻辑电路和 I/O 引脚的参考地。
VDD	32, 48, 71	P	—	逻辑电路和 I/O 引脚的正电源。
AVSS	26	P	—	模拟模块的参考地。
AVDD	25	P	—	模拟模块的正电源。
ENVREG	24	I	ST	使能片内稳压器。
VDDCORE/VCAP VDDCORE VCAP	12	P P	— —	内核逻辑电源或外部滤波电容连接端。 单片机内核逻辑的正电源（禁止稳压器）。 外部滤波电容连接（使能稳压器）。

图注:

TTL	= TTL 兼容输入	CMOS	= CMOS 兼容输入或输出
ST	= CMOS 电平的施密特触发器输入	Analog	= 模拟输入
I	= 输入	O	= 输出
P	= 电源	OD	= 漏极开路 (无 P 型二极管连接到 VDD)
I <sup>2</sup> C <sup>TM</sup>	= I <sup>2</sup> C/SMBus		

- 注 1:** 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配 (仅限 80 引脚器件的扩展单片机模式)。  
**注 2:** 当 CCP2MX 配置位置 1 时, 对 CCP2 进行默认分配。  
**注 3:** 当 CCP2MX 配置位清零时, 对 CCP2 进行其他分配。

# PIC18F85J11 系列

---

注:

## 2.0 PIC18FJ 单片机入门指南

### 2.1 基本连接要求

8位单片机PIC18F85J11系列入门要求在继续开发之前，必须至少注意一些器件引脚的连接。

以下引脚必须始终处于连接状态：

- 所有 VDD 和 VSS 引脚（见第 2.2 节“电源引脚”）
- 所有 AVDD 和 AVSS 引脚，而与是否使用模拟器件功能无关（见第 2.2 节“电源引脚”）
- MCLR 引脚（见第 2.3 节“主复位（MCLR）引脚”）
- ENVREG（如果存在）和 VCAP/VDDCORE 引脚（见第 2.4 节“稳压器引脚（ENVREG 和 VCAP/VDDCORE）”）

如果在最终应用中使用了以下引脚，那么也必须连接这些引脚：

- 用于在线串行编程（ICSP™）和调试的 PGC/PGD 引脚（见第 2.5 节“ICSP 引脚”）
- 使用外部振荡器时用到的 OSCI 和 OSCO 引脚（见第 2.6 节“外部振荡器引脚”）

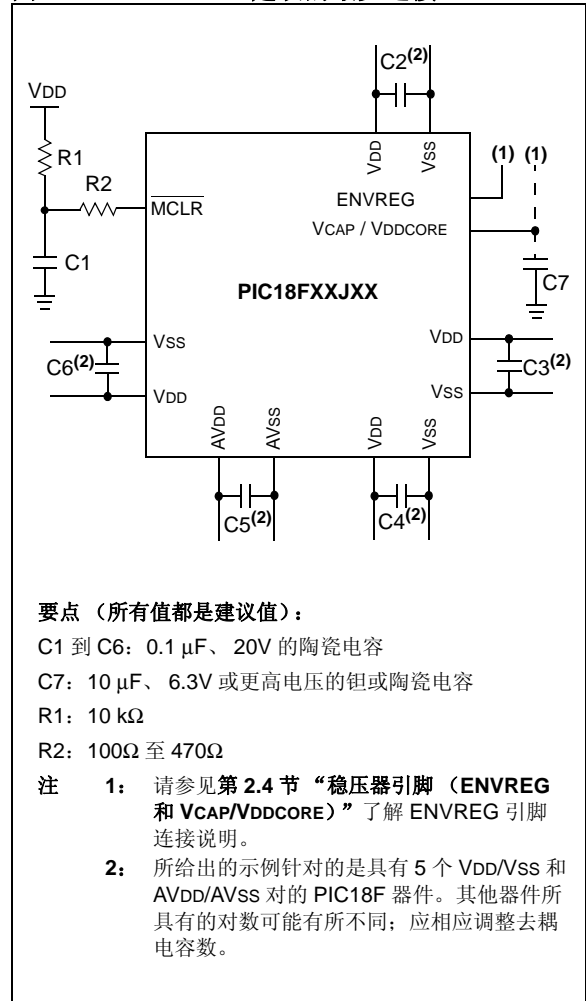
此外，可能需要以下引脚：

- 实现模拟模块的外部参考电压时使用的 VREF+/VREF- 引脚

**注：** 无论是否使用任何模拟模块，AVDD 和 AVSS 引脚都必须始终处于已连接状态。

图 2-1 中给出了必需的最少连接。

图 2-1： 建议的最少连接



## 2.2 电源引脚

### 2.2.1 去耦电容

需要在每对电源引脚（如 VDD 与 VSS，以及 AVDD 与 AVSS）上使用去耦电容。

使用去耦电容时，考虑以下条件：

- **电容的值和类型：**建议使用 0.1  $\mu\text{F}$ （100 nF）、10-20V 的电容。应使用谐振频率在 200 MHz 及更高范围内的低 ESR 电容。建议使用陶瓷电容。
- **印刷电路板上的放置：**去耦电容应尽可能靠近相应的引脚放置。建议将电容放到电路板上与器件相同的一侧。如果空间有限，可使用过孔将电容放到 PCB 的另一层上；但是，需要确保从引脚到电容的走线长度不超过 0.25 英寸（6 mm）。
- **处理高频噪声：**如果电路板产生高达几十兆赫兹的高频噪声，请在上述去耦电容旁并联一个陶瓷类型的辅助电容。辅助电容值的范围为 0.01  $\mu\text{F}$  至 0.001  $\mu\text{F}$ 。请将这个辅助电容靠近每个主去耦电容放置。在高速电路设计中，请考虑在尽可能靠近电源和接地引脚的地方布置 10 对电容（例如，一个 0.1  $\mu\text{F}$  的电容与一个 0.001  $\mu\text{F}$  的电容并联构成一对）。
- **最大程度提高性能：**在从电源电路布置电路板时，请首先连接电源并将走线返回到去耦电容，然后再连接到器件引脚。这可以确保去耦电容在电源链中处于第一位。保持电容和电源引脚之间走线长度尽可能短也同样重要，因为这可以减少 PCB 的走线感抗。

### 2.2.2 槽路电容

在电源走线长度大于 6 英寸的电路板上，建议在包含单片机的集成电路中使用槽路电容，以提供本地电源。槽路电容的值应根据以下因素确定：连接电源和器件的走线的电阻值以及应用中器件消耗的最大电流。也就是说，选择槽路电容使之满足器件的可接受的电压骤降要求。典型值范围为 4.7  $\mu\text{F}$  到 47  $\mu\text{F}$ 。

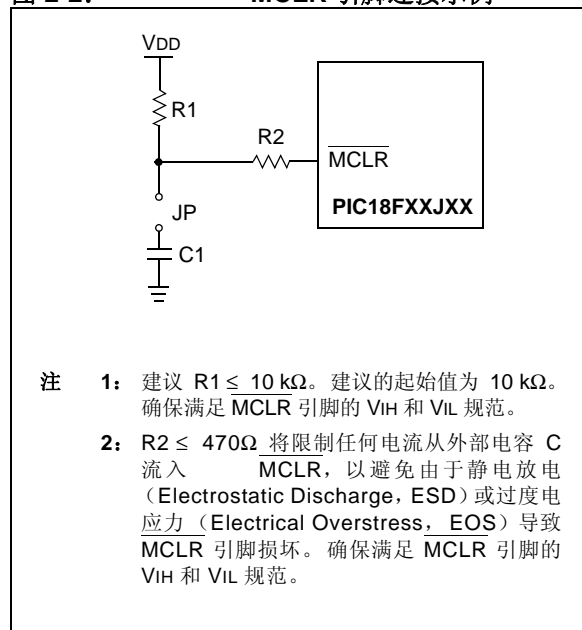
## 2.3 主复位（ $\overline{\text{MCLR}}$ ）引脚

$\overline{\text{MCLR}}$  引脚提供两个特定的器件功能：器件复位以及器件编程与调试。如果在最终应用中不需要编程和调试，则只需直接将该引脚连接到 VDD。为帮助增加应用的电阻以避免因电压骤降而意外复位，可能需要添加其他元件。图 2-1 给出了典型的配置。可根据应用的要求实现其他电路设计。

在编程和调试期间，必须考虑可以添加到引脚的电阻和电容。器件编程器和调试器驱动  $\overline{\text{MCLR}}$  引脚。因此，不允许对特殊电压电平（ $V_{IH}$  和  $V_{IL}$ ）以及快速信号变化造成不良影响。这就需要根据应用和 PCB 的要求调整 R1 和 C1 的具体值。例如，建议在编程和调试操作期间，通过使用跳线将电容 C1 和  $\overline{\text{MCLR}}$  引脚隔离（图 2-2）。正常的运行时操作不需要这样的跳线。

与  $\overline{\text{MCLR}}$  引脚相关的任何元件应放置在距离该引脚 0.25 英寸（6 mm）的范围内。

图 2-2:  $\overline{\text{MCLR}}$  引脚连接示例





## 2.4 稳压器引脚（ENVREG 和 VCAP/VDDCORE）

片上稳压器使能引脚 ENVREG 必须始终直接连接到电源电压或接地。将 ENVREG 连接到 VDD 会使能稳压器，而将其接地会禁止稳压器。连接和使用片上稳压器的详情，请参见第 23.3 节“片上稳压器”。

使能稳压器后，VCAP/VDDCORE 引脚上需要连接一个低 ESR ( $< 5\Omega$ ) 的电容来稳定稳压器输出电压。VCAP/VDDCORE 引脚不能连接到 VDD，并且必须使用一个  $10\mu\text{F}$  的电容接地。电容类型可以是陶瓷电容或钽电容。Murata GRM21BF50J106ZE01 ( $10\mu\text{F}$ , 6.3V) 或同等规格的电容就很适合。设计人员可以使用图 2-3 来计算备选电容的 ESR。

建议走线长度不超过 0.25 英寸 (6 mm)。更多信息请参见第 26.0 节“电气特性”。

禁止稳压器时，必须将 VCAP/VDDCORE 引脚连接到电压为 VDDCORE 的电源。关于 VDD 和 VDDCORE 的信息，请参见第 26.0 节“电气特性”。

请注意，一些低引脚数 PIC18FJ 器件的“LF”版本（例如，PIC18LF45J10）没有 ENVREG 引脚。这些器件的稳压器将永久禁止；且必须始终通过 VDDCORE 引脚为这些器件供电。

## 2.5 ICSP 引脚

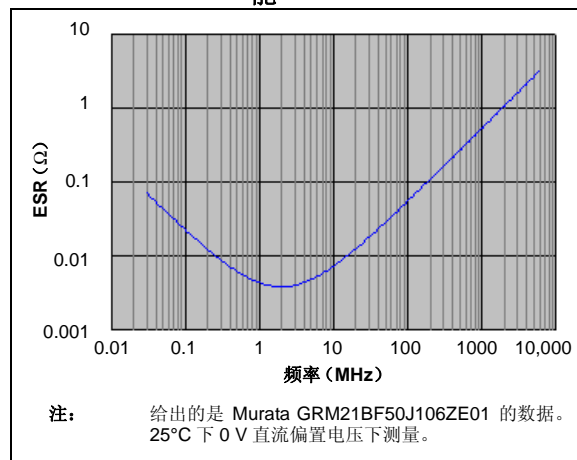
PGC 和 PGD 引脚用于在线串行编程（ICSP™）和调试。建议保持器件上的 ICSP 连接器和 ICSP 引脚之间的走线长度尽可能短。如果预期 ICSP 连接器上会发生 ESD 事件，建议使用一个串联电阻，且电阻值在几十欧姆范围内，不要超过  $100\Omega$ 。

建议不要在 PGC 和 PGD 引脚上使用上拉电阻、串联二极管和电容，因为它们会干扰编程器 / 调试器与器件之间的通信。如果应用需要这类分立元件，应在编程和调试期间将它们从电路中除去。或者，参阅相关器件闪存编程规范中的交流 / 直流特性和时序要求信息，以了解关于容性负载限制以及引脚高输入电压 ( $V_{IH}$ ) 和低输入电压 ( $V_{IL}$ ) 要求的信息。

对于器件仿真，请确保编程到器件中的“通信通道选择”（即，PGCx/PGDx 引脚）与 ICSP 到 Microchip 调试器 / 仿真器工具的物理连接一致。

关于可用 Microchip 开发工具连接要求的更多信息，请参见第 24.0 节“开发支持”。

图 2-3: 建议 VCAP 的频率和 ESR 性能



## 2.6 外部振荡器引脚

许多单片机都至少有两个振荡器选项：高频主振荡器和低频辅助振荡器（详情请参见第 3.0 节“振荡器配置”）。

振荡器电路应放到电路板上与器件相同的一侧。将振荡器电路靠近相关振荡器引脚放置，且电路元件与引脚之间的距离不要超过 0.5 英寸（12 mm）。负载电容应在电路板的同一侧挨着振荡器本身放置。

在振荡器电路周围设置接地敷铜区，将其与周围电路隔离。接地敷铜区应直接连接到 MCU 地。不要在接地敷铜区内使用任何信号线或电源线。而且，如果使用双面的电路板，请避免在电路板上放置晶振的另一面布线。

图 2-4 给出了建议的电路板布局。可以使用包含完整振荡器引脚的单面布局来处理直插式封装器件。对于引脚排列紧密的器件，单面布局则可能无法始终完全包含所有的引脚和元件。适当的解决方案是将含有保护走线的部分连接到反面的接地层。在所有情况下，保护走线必须返回到地。

在规划应用的走线和 I/O 分配过程中，确保临近的端口引脚和离振荡器非常近的其他信号状态良好（即，没有很高的频率、陡升和陡降以及其他类似的噪声）。

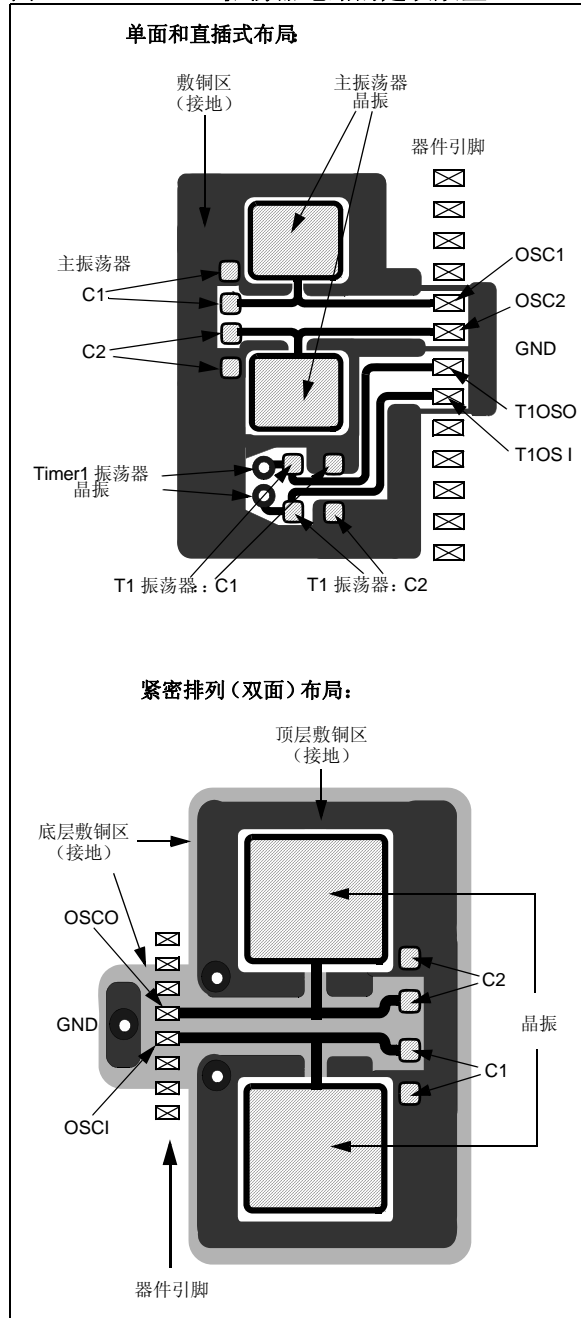
关于振荡器电路的其他信息和设计指导，请参考我公司网站（[www.microchip.com](http://www.microchip.com)）上提供的以下应用笔记：

- AN826, “Crystal Oscillator Basics and Crystal Selection for *rfPIC™* and *PICmicro®* Devices”
- AN849, “Basic *PICmicro®* Oscillator Design”
- AN943, “Practical *PICmicro®* Oscillator Analysis and Design”
- AN949, “Making Your Oscillator Work”

## 2.7 未使用的 I/O

未使用的 I/O 引脚应该配置为输出，并驱动为逻辑低电平状态。或者，使用一个 1 k $\Omega$  到 10 k $\Omega$  电阻将未使用的引脚连接到 VSS，并将输出驱动为逻辑低电平。

图 2-4：振荡器电路的建议放置



## 3.0 振荡器配置

### 3.1 振荡器类型

PIC18F85J11 系列器件可以在 6 种不同的振荡器模式下工作：

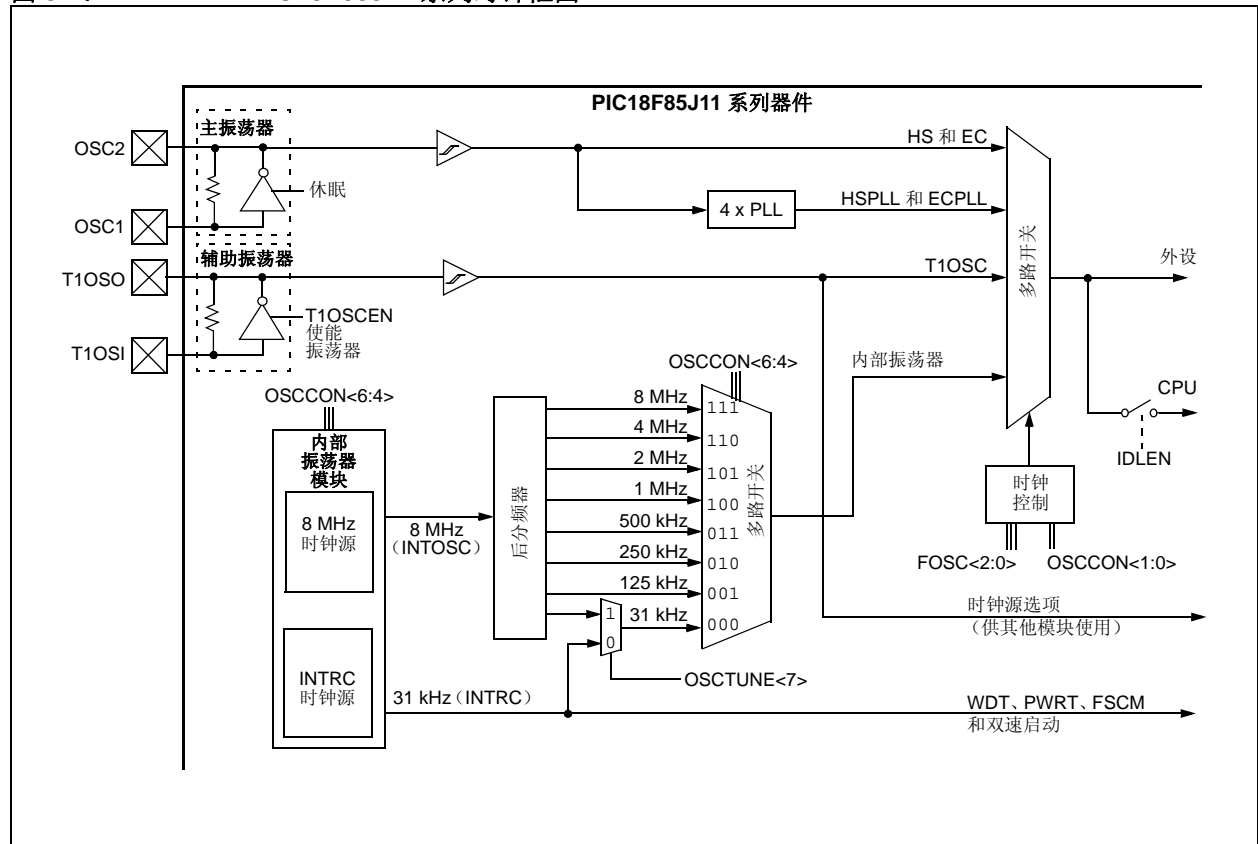
1. HS 高速晶振 / 谐振器
2. HSPLL 具有软件 PLL 控制的高速晶振 / 谐振器
3. EC 具有  $F_{osc}/4$  输出的外部时钟
4. ECPLL 具有软件 PLL 控制的外部时钟
5. INTOSC 内部快速 RC (8 MHz) 振荡器
6. INTRC 内部 31 kHz 振荡器

用户通过对  $FOSC<2:0>$  配置位进行编程选择其中 5 种模式。第 6 种模式 (INTRC) 可在软件控制下使用；该模式也可在器件复位后配置为默认模式。

此外，PIC18F85J11 系列器件可以在软件控制下或在某些条件下自动在不同时钟源之间进行切换。从而可通过实时管理器件时钟速度而无需复位应用，进一步节省功耗。

图 3-1 中显示了 PIC18F85J11 系列器件的时钟源。

图 3-1: PIC18F85J11 系列时钟框图



# PIC18F85J11 系列

## 3.2 控制寄存器

OSCCON 寄存器（寄存器 3-1）控制器件时钟运行的主要方面。它选择要使用的振荡器类型、要使用的功耗管理模式以及 INTOSC 源的输出频率。它也提供振荡器的状态。

OSCTUNE 寄存器（寄存器 3-2）控制内部振荡器模块的频率调节和工作。它还实现了 PLEN 位，该位用于控制内部振荡器模式下锁相环（PLL）的操作（见第 3.4.3 节“PLL 倍频器”）。

寄存器 3-1: OSCCON: 振荡器控制寄存器

R/W-0	R/W-1	R/W-0	R/W-0	R <sup>(1)</sup>	R-0	R/W-0	R/W-0
IDLEN	IRCF2 <sup>(2)</sup>	IRCF1 <sup>(2)</sup>	IRCF0 <sup>(2)</sup>	OSTS	IOFS	SCS1 <sup>(4)</sup>	SCS0 <sup>(4)</sup>
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	<b>IDLEN:</b> 空闲使能位 1 = 在执行 SLEEP 指令后器件进入空闲模式 0 = 在执行 SLEEP 指令后器件进入休眠模式
bit 6-4	<b>IRCF&lt;2:0&gt;:</b> INTOSC 源频率选择位 <sup>(2)</sup> 111 = 8 MHz (INTOSC 直接驱动时钟) 110 = 4 MHz 101 = 2 MHz 100 = 1 MHz (默认值) 011 = 500 kHz 010 = 250 kHz 001 = 125 kHz 000 = 31 kHz (来自 INTOSC/256 或 INTRC) <sup>(3)</sup>
bit 3	<b>OSTS:</b> 振荡器起振延时状态位 <sup>(1)</sup> 1 = 振荡器起振定时器 (OST) 延时已结束; 主振荡器正在运行 0 = 振荡器起振定时器 (OST) 正在进行延时; 主振荡器尚未就绪
bit 2	<b>IOFS:</b> INTOSC 频率稳定位 1 = 快速 RC 振荡器频率稳定 0 = 快速 RC 振荡器频率不稳定
bit 1-0	<b>SCS&lt;1:0&gt;:</b> 系统时钟选择位 <sup>(4)</sup> 11 = 内部振荡器模块 10 = 主振荡器 01 = Timer1 振荡器 <u>当 FOSC2 = 1 时:</u> 00 = 主振荡器 <u>当 FOSC2 = 0 时:</u> 00 = 内部振荡器

- 注 1: 复位状态取决于 IESO 配置位的状态。
- 2: 如果正由内部振荡器提供器件时钟, 更改这些位将导致时钟频率立刻切换。
- 3: 时钟源由 INTSRC 位 (OSCTUNE<7>) 选择, 请参见下文。
- 4: 更改这些位将导致时钟源立刻切换。

寄存器 3-2: OSCTUNE: 振荡器调节寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTSRC	PLLEN <sup>(1)</sup>	TUN5	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	<b>INTSRC:</b> 内部振荡器低频源选择位 1 = 源自 8 MHz INTOSC 源的 31.25 kHz 器件时钟 (使能 256 分频) 0 = 源自 INTRC 31 kHz 振荡器的 31 kHz 器件时钟
bit 6	<b>PLLEN:</b> PLL 倍频器使能位 <sup>(1)</sup> 1 = 使能 PLL 0 = 禁止 PLL
bit 5-0	<b>TUN&lt;5:0&gt;:</b> 快速 RC 振荡器 (INTOSC) 频率调节位 011111 = 最高频率 • • • • 000001 000000 = 中心频率。快速 RC 振荡器在校准后的频率下运行。 111111 • • • • 100000 = 最低频率

注 1: 仅在 ECPLL 和 HSPLL 振荡器配置下可用; 其他情况, 该位不可用且读为 0。

3.3 时钟源和振荡器切换

实质上, PIC18F85J11 系列器件具有 3 个独立的时钟源:

- 主振荡器
- 辅助振荡器
- 内部振荡器

**主振荡器**可以认为是主要的器件振荡器。这些是指与 OSC1 和 OSC2 引脚连接的任何外部振荡器, 包括外部晶振和谐振器模式以及外部时钟模式。在某些情况下, 内部振荡器模块也可认为是主振荡器。特定的模式由 FOSC 配置位定义。第 3.4 节 “外部振荡器模式” 中包含了这些模式的详情。

**辅助振荡器**是未连接到 OSC1 或 OSC2 引脚的外部时钟源。即使在控制器处于功耗管理模式时, 这些时钟源仍

可继续工作。PIC18F85J11 系列器件将 Timer1 振荡器作为辅助振荡器源。该振荡器在所有功耗管理模式下通常被用作实时时钟等功能的时基。第 13.3 节 “Timer1 振荡器” 中对 Timer1 振荡器进行了更为详细的讨论。

除了在某些情况下作为主时钟源之外, **内部振荡器**还可以作为功耗管理模式的时钟源。INTRC 源也可以作为几种特殊功能部件 (例如 WDT 和故障保护时钟监视器) 的时钟源。第 3.5 节 “内部振荡器模块” 对内部振荡器模块进行了更为详细的讨论。

PIC18F85J11 系列器件包含允许器件时钟源从主振荡器 (由器件配置选择) 切换到其他时钟源的功能。当使能其他时钟源时, 可以使用多种功耗管理工作模式。

# PIC18F85J11 系列

## 3.3.1 时钟源选择

系统时钟选择位  $SCS<1:0>$  ( $OSCCON<1:0>$ ) 选择时钟源。可用的时钟源包括主时钟（由  $FOSC<1:0>$  配置位定义）、辅助时钟（**Timer1** 振荡器）和内部振荡器。当写入一个或多个位之后，接下来是一段很短的时钟转换延时，然后，时钟源改变。

$OSTS$  ( $OSCCON<3>$ ) 和  $T1RUN$  ( $T1CON<6>$ ) 位指明当前提供器件时钟的时钟源。 $OSTS$  位置 1 表明振荡器起振定时器 ( $OST$ ) 已超时，且主时钟在主时钟模式下提供器件时钟。 $T1RUN$  位置 1 表明 **Timer1** 振荡器在辅助时钟模式下提供器件时钟。在功耗管理模式下，任何时候这两个位中只有一个会置 1。如果这两位都没有置 1，则表示当前时钟源是  $INTRC$ ，或内部振荡器刚刚起振且尚未稳定。

$IDLEN$  位决定当执行  $SLEEP$  指令时，器件是进入休眠模式还是某种空闲模式。

第 4.0 节“功耗管理模式”中详细讨论了  $OSCCON$  寄存器中的标志位和控制位的使用。

**注 1:** 要选择辅助时钟源，必须使能 **Timer1** 振荡器。通过将 **Timer1** 控制寄存器中的  $T1OSCEN$  位 ( $T1CON<3>$ ) 置 1，可以使能 **Timer1** 振荡器。如果未使能 **Timer1** 振荡器，在执行  $SLEEP$  指令时选择辅助时钟源的任何尝试都会被忽略。

**2:** 建议在 **Timer1** 振荡器稳定工作之后再执行  $SLEEP$  指令，因为当 **Timer1** 振荡器起振时会有很长的延时。

## 3.3.1.1 系统时钟选择位和 $FOSC2$ 配置位

在所有形式的复位中  $SCS$  位都被清零。在器件的默认配置中，这表示由  $FOSC<1:0>$  定义的主振荡器（即， $HS$  或  $EC$  模式）作为器件复位后的主时钟源。

可以使用  $FOSC2$  配置位来更改复位后的默认时钟配置。该位确定在随后的器件复位时是由外部振荡器还是由内部振荡器作为默认时钟源。此外，它还会影响  $SCS<1:0>$  在其复位状态 ( $= 00$ ) 时时钟源的选择。若  $FOSC2 = 1$ （默认值），在  $SCS<1:0> = 00$  时选择由  $FOSC<1:0>$  定义的振荡器源。若  $FOSC2 = 0$ ，在  $SCS<2:1> = 00$  时选择内部振荡器模块。

在内部振荡器模块作为复位时默认时钟的情况下，快速  $RC$  振荡器 ( $INTOSC$ ) 将被用作器件时钟源。它将从 1 MHz 开始起振，这是对应于  $IRCF<2:0>$  位复位值 ( $100$ ) 的后分频比选项。

无论  $FOSC2$  的设置如何，器件上电后将总是使能  $INTRC$ 。它将作为时钟源直到器件从存储器中装入了它的配置值。此时  $FOSC$  配置位被读取并选择振荡器的工作模式。

注意主时钟或内部振荡器在任何时刻下都会有两种可能的  $SCS<1:0>$  位设置选项，具体取决于  $FOSC2$  的值。

## 3.3.2 振荡器转换

**PIC18F85J11** 系列器件包含在时钟源切换时防止时钟产生“毛刺”的电路。在切换时钟时，系统时钟会有短暂的停顿。该停顿的时间长度是旧时钟源的两个周期加上新时钟源的三到四个周期的和。此公式假设新时钟源是稳定的。

在第 4.1.2 节“进入功耗管理模式”中详细讨论了时钟转换。

3.4 外部振荡器模式

3.4.1 晶振 / 陶瓷谐振器（HS 模式）

在 HS 或 HSPLL 振荡器模式下，将晶振或陶瓷谐振器连接到 OSC1 和 OSC2 引脚来产生振荡信号。图 3-2 中显示了引脚连接。

振荡器的设计要求使用平行切割的晶体。

**注：** 使用顺序切割的晶体，可能会使振荡器产生的频率超出晶体制造厂商所给的参数范围。

表 3-1：陶瓷谐振器的电容选择

使用的典型电容值：			
模式	频率	OSC1	OSC2
HS	8.0 MHz	27 pF	27 pF
	16.0 MHz	22 pF	22 pF

上述电容值仅供设计参考。

要得到理想的振荡器工作状态，可能需要不同的电容值。用户应当在应用期望的 VDD 和温度条件下测试振荡器的性能。请参见以下应用笔记以获取振荡器的具体信息：

- AN588, “PIC® Microcontroller Oscillator Design Guide”
- AN826, “Crystal Oscillator Basics and Crystal Selection for rPIC® and PIC® Devices”
- AN849, “Basic PIC® Oscillator Design”
- AN943, “Practical PIC® Oscillator Analysis and Design”
- AN949, “Making Your Oscillator Work”

欲知更多信息，请参见表 3-2 后面的“注”。

表 3-2：晶振的电容选择

振荡类型	晶振频率	已测试的典型电容值：	
		C1	C2
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

上述电容值仅供设计参考。

要得到理想的振荡器工作状态，可能需要不同的电容值。用户应当在应用期望的 VDD 和温度条件下测试振荡器的性能。

振荡器的具体信息，请参见表 3-1 列出的 Microchip 应用笔记。欲知更多信息，请参见本表后面的“注”。

**注**

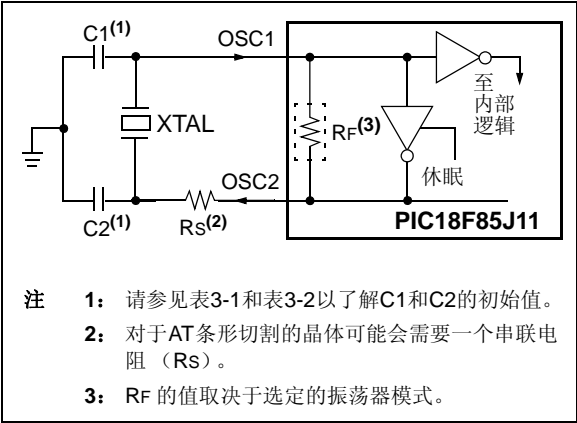
1： 电容值越大，振荡器的稳定性越高，但同时起振时间也越长。

2： 因为每种谐振器 / 晶振都有其自身特性，用户应当向谐振器 / 晶振制造厂商询问外部元件的适当值。

3： 可能需要使用电阻 Rs 以避免对低驱动规格的晶体造成过驱动。

4： 请始终在应用期望的 VDD 和温度范围下验证振荡器性能。

图 3-2：晶振 / 陶瓷谐振器工作原理（HS 或 HSPLL 配置）



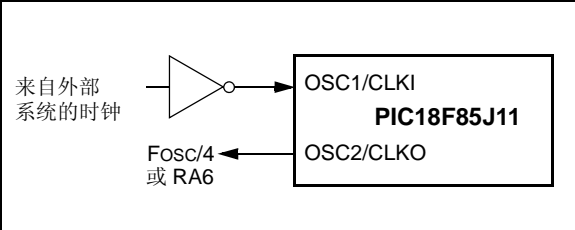
# PIC18F85J11 系列

## 3.4.2 外部时钟输入（EC 模式）

EC 和 ECPLL 振荡器模式要求 OSC1 引脚与一个外部时钟源相连。在上电复位后或从休眠模式退出后，不需要振荡器起振时间。

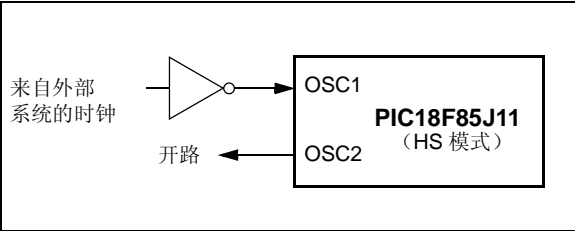
在 EC 振荡器模式下，振荡器频率的 4 分频信号可由 OSC2 引脚输出。此信号可用于测试或同步其他逻辑。图 3-3 给出了 EC 振荡器模式的引脚连接方式。

图 3-3: 外部时钟输入工作原理 (EC 配置)



在 HS 模式下，OSC1 引脚可能也需要连接一个外部时钟源，如图 3-4 所示。在此配置下，OSC2 的 4 分频输出不可用。

图 3-4: 外部时钟输入工作原理 (HS 振荡器配置)

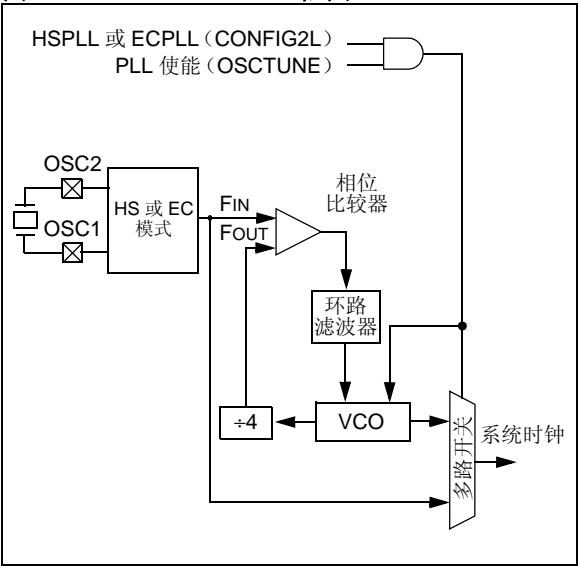


## 3.4.3 PLL 倍频器

如果用户希望使用低频振荡器电路或通过晶振将器件频率调节至其最高额定频率，可以选择使用锁相环 (PLL) 电路。对于担心高频晶振引起 EMI 或需要内部振荡器提供高速时钟的用户而言，这样做可能会有用。因此，HSPLL 和 ECPLL 模式可用。

HSPLL 和 ECPLL 模式可以使器件以外部振荡源频率的 4 倍运行，从而使最高频率可达 40 MHz。通过将 FOSC<2:0> 配置位 (CONFIG2L<2:0>) 编程为 110 (对于 ECPLL) 或 100 (对于 HSPLL) 来使能 PLL。另外，Pllen 位 (OSCTUNE<6>) 也必须置 1。不管选定的振荡器配置为何，清零 Pllen 位都会禁止 PLL。它还允许软件更灵活地控制应用的时钟速度。

图 3-5: PLL 框图





## 3.5 内部振荡器模块

PIC18F85J11 系列器件包含一个内部振荡器模块，该模块可产生两种不同的时钟信号；这两种信号均可充当单片机的时钟源，从而避免在 OSC1 和 / 或 OSC2 引脚上使用外部振荡器电路。

主输出为快速 RC 振荡器或 INTOSC（可用来直接驱动器件时钟的 8 MHz 时钟源）。它还可以驱动后分频器，该后分频器可以提供从 31 kHz 到 4 MHz 的时钟频率。当选择了频率为 125 kHz 到 8 MHz 的时钟时，会使能 INTOSC。当选择了频率为 31 kHz 的时钟时，也会使能 INTOSC 输出，这取决于 INTSRC 位（OSCTUNE<7>）。

另一个时钟源是内部 RC 振荡器（INTRC），它提供了标称值为 31 kHz 的输出。如果 INTRC 被选中为器件时钟源，则使能 INTRC。当使能以下任一功能时，也将自动使能 INTRC：

- 上电延时定时器
- 故障保护时钟监视器
- 看门狗定时器
- 双速启动

第 23.0 节“CPU 的特殊功能”中详细讨论了这些功能。

通过配置 OSCCON 寄存器的 IRCF 位，可以选择时钟源频率（INTOSC 频率，INTOSC 后分频频率或 INTRC 频率）。器件复位后的默认频率为 1 MHz。

### 3.5.1 OSC1 和 OSC2 引脚配置

每当内部振荡器配置为默认时钟源（FOSC2 = 0）时，OSC1 和 OSC2 引脚都会被自动重新配置为端口引脚 RA6 和 RA7。在此模式下，它们用作通用数字 I/O。这些引脚上的所有振荡器功能被禁止。

### 3.5.2 内部振荡器输出频率与调节

出厂时已校准了内部振荡器模块使之能够产生 8 MHz 的 INTOSC 输出频率。可以通过写 OSCTUNE 寄存器（寄存器 3-2）中的 TUN<5:0>（OSCTUNE<5:0>），在用户应用中对其进行调整。

当修改了 OSCTUNE 寄存器后，INTOSC 频率将逐渐向新的频率转变。该振荡器会在 1 ms 内稳定下来。在此变化期间，代码会继续执行。不会有任何迹象表明时钟频率发生了改变。

INTRC 振荡器的工作独立于 INTOSC 源。电压和温度变化导致的 INTOSC 变化并不一定会使 INTRC 变化，反之亦然。INTRC 的频率不受 OSCTUNE 的影响。

### 3.5.3 INTOSC 频率漂移

INTOSC 频率可能会随着 VDD 电压或温度的改变而发生漂移，并能以各种方式影响控制器运行。通过修改 OSCTUNE 寄存器的值可以调节 INTOSC 的频率。这对 INTRC 时钟源频率没有影响。

调节 INTOSC 时钟源需要了解调节时间、调节方向以及在某些情况下的调整量。这里给出了三种补偿技术。

#### 3.5.3.1 用 EUSART 进行补偿

当 EUSART 在异步模式下开始产生帧错误，或者接收到错误数据时可能需要进行调节。帧错误表示器件时钟的频率太高。要对此进行调节，可以减小 OSCTUNE 中的值来降低时钟频率。另一方面，数据中有错误可能表明时钟速度太低。要进行补偿，可以增大 OSCTUNE 中的值来提高时钟频率。

#### 3.5.3.2 用定时器进行补偿

此技术是将器件时钟的速度与某一个参考时钟进行比较。可能要用到两个定时器；一个由外设时钟提供时钟源，而另一个由一个固定的参考源（如 Timer1 振荡器）提供时钟源。

两个定时器都被清零，但由参考源提供时钟信号的定时器产生中断。当中断发生时，使用内部时钟源的定时器值被读取且两个定时器均被清零。如果使用内部时钟源的定时器的值比期望值大很多，则表示内部振荡器模块运行过快。要对此进行调节，需减小 OSCTUNE 寄存器中的值。

#### 3.5.3.3 在捕捉模式下用 CCP 模块进行补偿

CCP 模块可使用由内部振荡器模块提供时钟源的独立运行的 Timer1（或 Timer3）和周期已知的外部事件（例如，交流电源频率）。CCPRxH:CCPRxL 寄存器捕捉第一个事件的时间，并进行记录以备后用。当第二个事件导致捕捉时，要用第二个事件的时间减去第一个事件的时间。由于外部事件周期是已知的，因此可以计算事件之间的时间差。

如果测量所得时间远大于计算所得时间，说明内部振荡器模块运行过快。要进行补偿，应减小 OSCTUNE 寄存器的值。如果测量所得时间远小于计算所得时间，说明内部振荡器模块运行过慢。要进行补偿，应增大 OSCTUNE 寄存器的值。

### 3.6 功耗管理模式对各种时钟源的影响

当选择了 PRI\_IDLE 模式时，指定的主振荡器会继续运行而不中断。对于所有其他功耗管理模式，使用 OSC1 引脚的振荡器会被禁止。OSC1 引脚（以及由振荡器使用的 OSC2 引脚）将停止振荡。

在辅助时钟模式（SEC\_RUN和SEC\_IDLE）下，Timer1 振荡器作为器件时钟源工作。如果需要，Timer1 振荡器也可以运行在所有功耗管理模式下为 Timer1 或 Timer3 提供时钟源。

在 RC\_RUN 和 RC\_IDLE 模式下，由内部振荡器提供器件时钟源。可直接使用 31 kHz 的 INTRC 输出来提供时钟，并且可被使能来支持多种特殊功能，而与功耗管理模式无关（欲知更多有关 WDT、故障保护时钟监视器和双速启动的信息，请参见第 23.2 节“看门狗定时器（WDT）”到第 23.5 节“故障保护时钟监视器”

如果选择了休眠模式，所有的时钟源都会停止。因为休眠模式消除了所有的晶体管开关电流，休眠模式能实现最小的器件电流消耗（仅有泄漏电流）。

在休眠期间使能任何片上功能将会增加休眠时的电流消耗。需要使能 INTRC 来支持 WDT 工作。Timer1 振荡

器可以用来为实时时钟（RTC）提供时钟源。不需要系统时钟源的其他功能部件（即 MSSP 从器件、PSP 和 INTn 引脚等）也可以工作。第 26.2 节“直流特性：掉电和供电电流”列出了会明显增加电流消耗的外设。

### 3.7 上电延时

由两个定时器控制上电延时，这样大部分应用都无需外接复位电路。上电延时可以确保在器件电源稳定（常规环境下）和主时钟稳定工作之前器件保持复位状态。欲知有关上电延时的更多信息，请参见第 5.6 节“上电延时定时器（PWRT）”。

第一个定时器是上电延时定时器（PWRT），在上电时它提供了固定的延迟（表 26-13 中的参数 33）。它总是使能的。

第二个定时器是振荡器起振定时器（OST），用于在晶振稳定前使芯片保持在复位状态（HS 模式）。OST 在计数 1024 个振荡周期后允许振荡器为器件提供时钟。

在上电复位之后，会有一段 TcSD（表 26-13 中的参数 38）间隔的延时，此延时期间控制器为执行指令做准备。

表 3-3: 休眠模式下 OSC1 和 OSC2 引脚的状态

振荡器模式	OSC1 引脚	OSC2 引脚
EC 和 ECPLL	悬空，由外部时钟驱动	处于逻辑低电平（时钟的 4 分频输出）
HS 和 HSPLL	反馈反相器被禁止，处于静态电平	反馈反相器被禁止，处于静态电平
INTOSC	I/O 引脚 RA6，由 TRISA<6> 控制方向	I/O 引脚 RA7，由 TRISA<7> 控制方向

注：有关由于休眠和 MCLR 复位引起的延时的信息，请参见第 5.0 节“复位”中的表 5-2。

## 4.0 功耗管理模式

PIC18F85J11 系列器件提供了只需通过管理 CPU 和外设的时钟源就可以管理功耗的功能。通常，降低时钟频率和减少由时钟源驱动的电路数目会使功耗降低。为了在应用中管理功耗，提供了三种主要的工作模式：

- 运行模式
- 空闲模式
- 休眠模式

这些模式定义了器件的哪些部分由时钟源驱动，以及以多高的时钟速度驱动。运行和空闲模式可以使用三种时钟源（主时钟源、辅助时钟源或内部振荡电路）中的任意一种；而休眠模式则不使用时钟源。

功耗管理模式包括几种在以前的 PIC® MCU 器件上提供的节约功耗的功能。其中之一是其他 PIC18 器件也提供的时钟切换功能，该功能允许控制器使用 Timer1 振荡器代替主振荡器。节约功耗的功能还包括所有 PIC MCU 都提供的休眠模式，在此模式下所有器件时钟均停止。

### 4.1 选择功耗管理模式

选择功耗管理模式需要考虑两个因素：是否用时钟源驱动 CPU，以及使用哪个时钟源。IDLEN 位（OSCCON<7>）控制是否为 CPU 提供时钟，而 SCS<1:0> 位（OSCCON<1:0>）选择时钟源。表 4-1 总结了各个模式下的位设置、时钟源和受影响的模块。

表 4-1： 功耗管理模式

模式	OSCCON 位		模块时钟控制		可用时钟和振荡器源
	IDLEN<7> <sup>(1)</sup>	SCS<1:0>	CPU	外设	
休眠	0	N/A	关闭	关闭	无——所有时钟被禁止
PRI_RUN	N/A	10	提供时钟	提供时钟	主时钟源——HS、EC、HSPLL 和 ECPLL；这是正常的全功耗执行模式
SEC_RUN	N/A	01	提供时钟	提供时钟	辅助时钟源——Timer1 振荡器
RC_RUN	N/A	11	提供时钟	提供时钟	内部振荡器
PRI_IDLE	1	10	关闭	提供时钟	主时钟源——HS、EC、HSPLL 和 ECPLL
SEC_IDLE	1	01	关闭	提供时钟	辅助时钟源——Timer1 振荡器
RC_IDLE	1	11	关闭	提供时钟	内部振荡器

注 1： 执行 SLEEP 指令时，IDLEN 的值得到反映。

### 4.1.1 时钟源

SCS<1:0> 位可以为功耗管理模式选择三个时钟源中的一个。它们是：

- 主时钟源，由 FOSC<2:0> 配置位定义
- 辅助时钟源（Timer1 振荡器）
- 内部振荡器

### 4.1.2 进入功耗管理模式

通过装载 OSCCON 寄存器可从一种功耗管理模式切换到另一种功耗管理模式。SCS<1:0> 位选择时钟源并决定使用哪一种运行模式或空闲模式。更改这些位将会立即切换到新的时钟源（假定新的时钟源正在运行）。切换时，可能会有时钟转换延迟。第 4.1.3 节“时钟转换和状态指示位”及其后续章节将会讨论这些问题。

执行 SLEEP 指令可以触发进入功耗管理空闲模式或休眠模式。最后实际进入了哪个模式由 IDLEN 位的状态决定。

更改功耗管理模式并不总是需要设置所有这些位，这由当前的模式和将要切换到的模式决定。可以通过在发出 SLEEP 指令之前更改振荡器选择位或更改 IDLEN 位来进行多种模式转换。如果已经正确地配置了 IDLEN 位，可能只须通过执行 SLEEP 指令就可切换到所需的模式。

## 4.1.3 时钟转换和状态指示位

时钟源转换的时间长度是旧时钟源的 2 个周期与新时钟源的 3 到 4 个周期的和。此公式假设新时钟源是稳定的。

这两个位表示当前的时钟源及其状态：OSTS (OSCCON<3>) 和 T1RUN (T1CON<6>)。通常，在给定功耗管理模式下，这两个位中只有一位会被置 1。当 OSTS 位置 1 时，表明由主时钟提供器件时钟。当 T1RUN 位置 1 时，表明由 Timer1 振荡器提供器件时钟。如果这两位都不置 1，则由 INTRC 为器件提供时钟信号。

**注：** 执行 SLEEP 指令不一定会使器件进入休眠模式。该指令充当触发条件，根据 IDLEN 位的设置，使控制器进入休眠模式和某种空闲模式。

## 4.1.4 多条 SLEEP 命令

SLEEP 指令启动的功耗管理模式由这条指令执行时 IDLEN 位的设置决定。如果执行另一条 SLEEP 指令，器件将进入那时由 IDLEN 位指定的功耗管理模式。如果 IDLEN 位已更改，器件将进入由新的设置指定的新功耗管理模式。

## 4.2 运行模式

在运行模式下，内核和外设的时钟均有效。不同运行模式之间的差异在于时钟源的不同。

### 4.2.1 PRI\_RUN 模式

PRI\_RUN 模式是单片机正常工作的全功耗运行模式。如果未使能双速启动，PRI\_RUN 模式也是器件复位后的默认模式（详情请见第 23.4 节“双速启动”）。在此模式下，OSTS 位置 1（见第 3.2 节“控制寄存器”）。

### 4.2.2 SEC\_RUN 模式

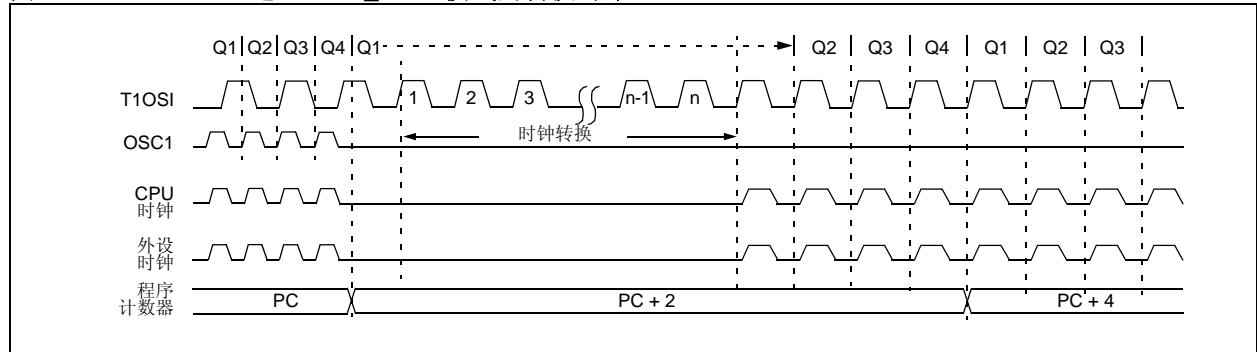
SEC\_RUN 模式与其他 PIC18 器件所提供的“时钟切换”功能兼容。在此模式下，CPU 和外设由 Timer1 振荡器提供时钟。这让用户在仍使用高精度时钟源的情况下适当地降低功耗。

通过将 SCS<1:0> 位设置为 01 进入 SEC\_RUN 模式。器件时钟源切换到 Timer1 振荡器（见图 4-1），关闭主振荡器，T1RUN 位 (T1CON<6>) 置 1，OSTS 位清零。

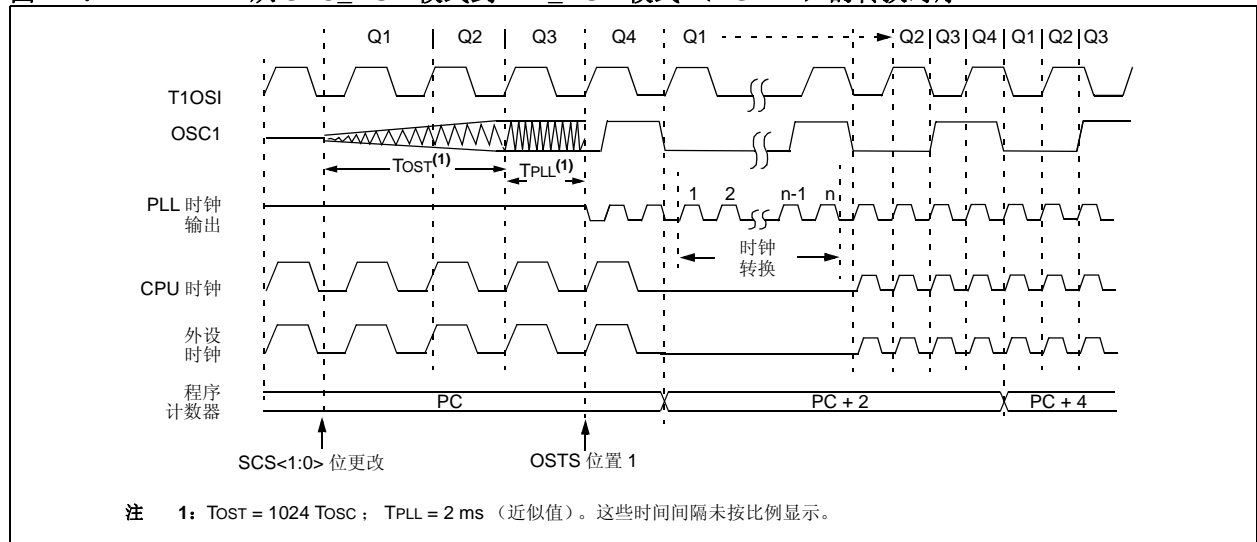
**注：** Timer1 振荡器应该在进入 SEC\_RUN 模式之前已经开始运行。如果当 SCS<1:0> 位被设置为 01 时，T1OSCEN 位未置 1，将不会进入 SEC\_RUN 模式。如果 Timer1 振荡器已经被使能，但没有开始运行，器件时钟将被延时直到该振荡器起振为止。在这种情况下，最初振荡器运行很不稳定，可能会导致无法预料的结果。

从 SEC\_RUN 模式转换到 PRI\_RUN 模式时，在主时钟启动期间，外设和 CPU 继续将 Timer1 振荡器作为时钟源。当主时钟准备就绪以后，时钟切换回主时钟（见图 4-2）。当时钟切换完成后，T1RUN 位被清零，OSTS 位被置 1，并由主时钟提供器件时钟。唤醒不会影响 IDLEN 和 SCS 位；Timer1 振荡器继续运行。

**图 4-1：** 进入 SEC\_RUN 模式的转换时序



**图 4-2：** 从 SEC\_RUN 模式到 PRI\_RUN 模式（HSPLL）的转换时序



# PIC18F85J11 系列

## 4.2.3 RC\_RUN 模式

在 RC\_RUN 模式下，使用内部振荡器作为 CPU 和外设的时钟源；主时钟关闭。此模式是在代码执行期间所有运行模式中最节省功耗的运行模式。它非常适合于那些对时序要求不高或不是一直需要高速时钟的应用。

通过将 SCS 设置为 11 可以进入此模式。当时钟源切换为 INTRC（见图4-3）时，主振荡器关闭，OSTS 位清零。

从 RC\_RUN 模式转换到 PRI\_RUN 模式时，在主时钟启动期间，器件继续使用 INTRC 电路作为时钟源。当主时钟准备就绪以后，时钟切换回主时钟（见图 4-4）。当时钟切换完成后，OSTS 位被置 1，主时钟提供器件时钟。切换不会影响 IDLEN 和 SCS 位。如果使能了 WDT 或故障保护时钟监视器，INTRC 源将继续运行。

图 4-3: 到 RC\_RUN 模式的转换时序

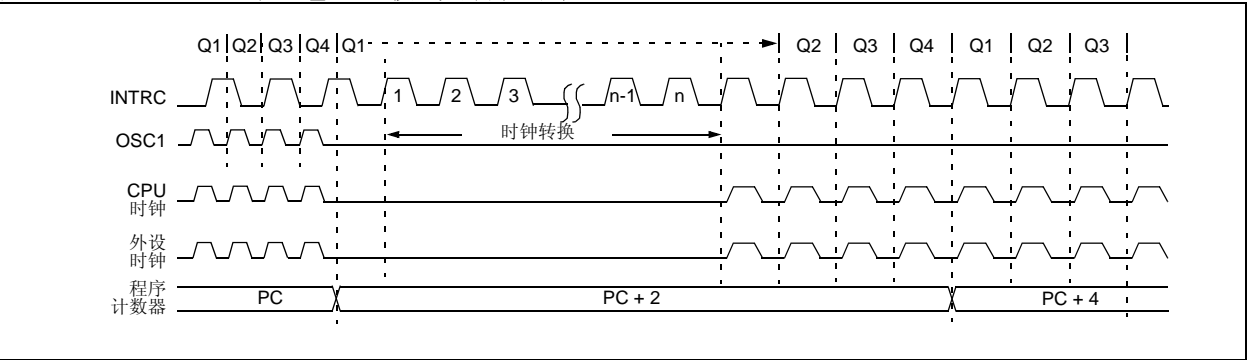
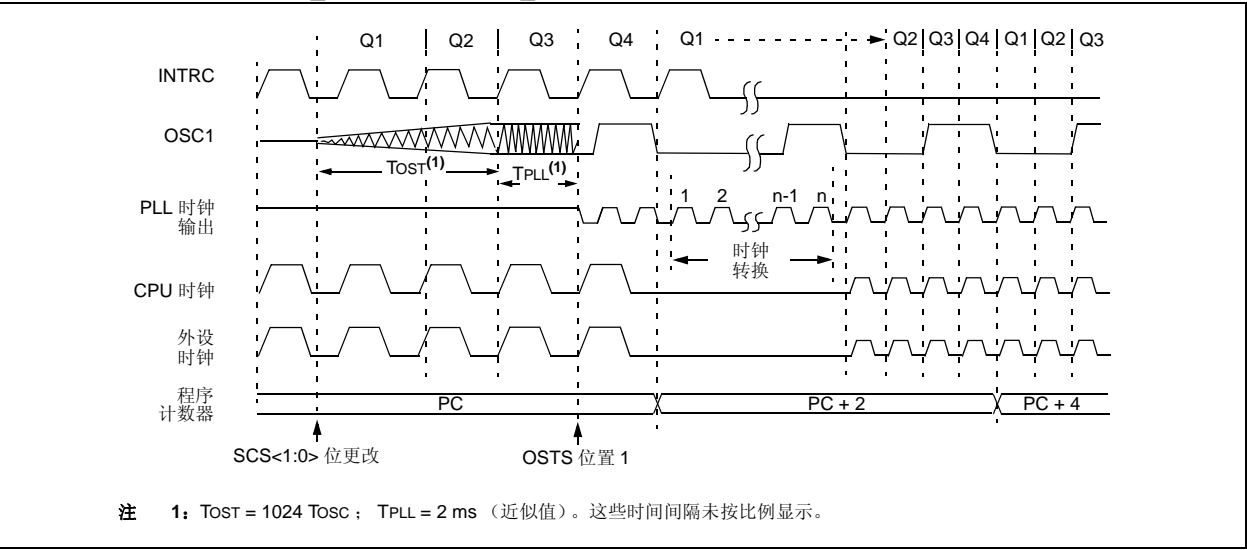


图 4-4: 从 RC\_RUN 模式到 PRI\_RUN 模式的转换时序



## 4.3 休眠模式

功耗管理休眠模式与所有其他 PIC 单片机所提供的传统休眠模式相同。通过清零 **IDLEN** 位（器件复位时的默认状态）并执行 **SLEEP** 指令进入该模式。这将关闭选定的振荡器（图 4-5），并清零所有的时钟源状态位。

从任何其他模式进入休眠模式不需要时钟切换。这是因为一旦控制器进入休眠模式后就不需要时钟了。如果选择了 **WDT**，**INTRC** 源将继续运行。如果使能了 **Timer1** 振荡器，它也将继续运行。

在休眠模式下发生唤醒事件时（由于中断、复位或 **WDT** 超时），器件将没有时钟源直到由 **SCS<1:0>** 位选定的时钟源准备就绪为止（见图 4-6），或者如果使能了双速启动或故障保护时钟监视器，它会使用内部振荡器作为时钟源（见第 23.0 节“CPU 的特殊功能”）。在这两种情况下，当主时钟提供器件时钟时，**OSTS** 位被置 1。唤醒事件不会影响 **IDLEN** 和 **SCS** 位。

## 4.4 空闲模式

空闲模式允许在外设继续工作时关闭单片机的 **CPU**。选择某种特定的空闲模式使用户能进一步管理功耗。

若在 **SLEEP** 指令执行时，**IDLEN** 已置 1，外设将使用由 **SCS<1:0>** 位选择的时钟源而不会为 **CPU** 提供时钟。时钟源状态位不受影响。置 1 **IDLEN** 并执行 **SLEEP** 指令将快速从给定运行模式切换到相应的空闲模式。

如果选择了 **WDT**，**INTRC** 时钟源将继续运行。如果使能了 **Timer1** 振荡器，它也将继续运行。

由于 **CPU** 没有执行指令，器件只能通过中断、**WDT** 超时或复位从任一空闲模式退出。当发生唤醒事件时，会有一个 **Tcsd** 间隔（表 26-13 中的参数 38）的延时，之后 **CPU** 恢复代码执行。当 **CPU** 开始执行代码时，它将沿用当前空闲模式使用的时钟源。例如，当从 **RC\_IDLE** 模式唤醒时，将使用内部振荡器电路作为 **CPU** 和外设的时钟源（即 **RC\_RUN** 模式）。唤醒事件不会影响 **IDLEN** 和 **SCS** 位。

当处于任何空闲或休眠模式时，**WDT** 超时将导致 **WDT** 唤醒并进入由当前 **SCS<1:0>** 位指定的运行模式。

图 4-5: 进入休眠模式的转换时序

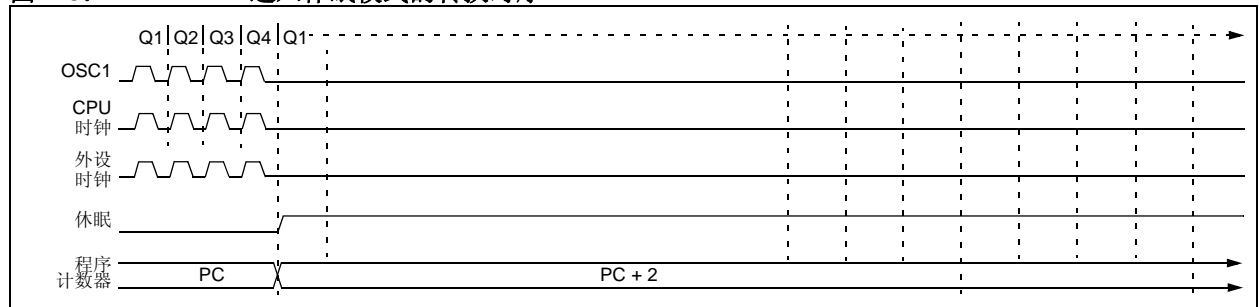
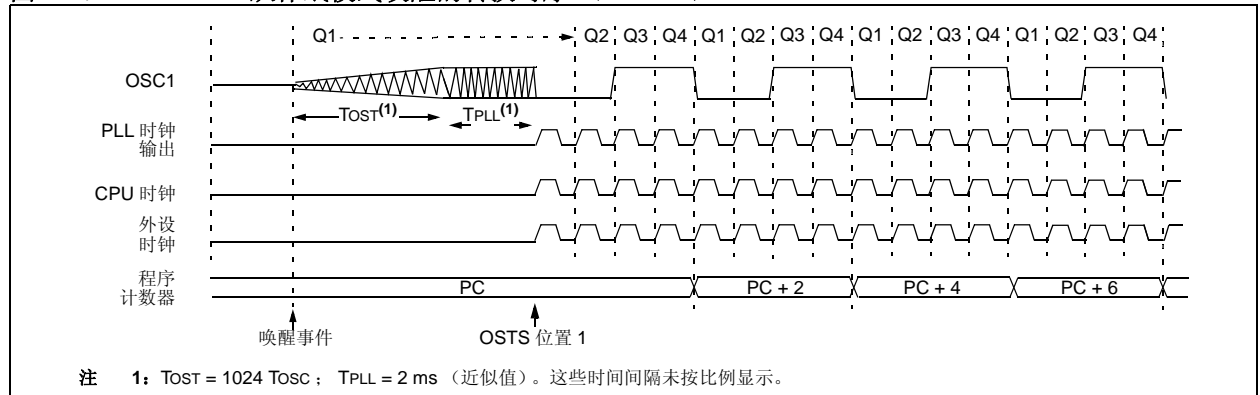


图 4-6: 从休眠模式唤醒的转换时序（HSPLL）



# PIC18F85J11 系列

## 4.4.1 PRI\_IDLE 模式

在三种低功耗空闲模式中，只有该模式不会禁止主器件时钟。由于时钟源不需要“预热”或从其他振荡器转换过来，选用此模式可以使对时序要求较高的应用以最快的速度恢复器件运行并使用较精确的主时钟源。

通过将 IDLEN 位置 1 并执行 SLEEP 指令，可以从 PRI\_RUN 模式进入 PRI\_IDLE 模式。如果器件处于其他运行模式，请先将 IDLEN 位置 1，然后将 SCS 位置为 10 并执行 SLEEP。虽然 CPU 已被禁止，但外设仍可继续使用由 FOSC<1:0> 配置位指定的主时钟源为其提供时钟信号。OSTS 位保持置 1（见图 4-7）。

当发生唤醒事件时，由主时钟源为 CPU 提供时钟。在唤醒事件和代码开始执行之间需要一段 T<sub>CSD</sub> 间隔的延时。这段时间可以使 CPU 做好执行指令的准备。在唤醒之后，OSTS 位保持置 1。唤醒事件不会影响 IDLEN 和 SCS 位（见图 4-8）。

## 4.4.2 SEC\_IDLE 模式

在 SEC\_IDLE 模式中，CPU 被禁止，但外设继续将 Timer1 振荡器作为时钟源。通过将 IDLEN 位置 1 并执行 SLEEP 指令，可以从 SEC\_RUN 进入此模式。如果器件处于其他运行模式，请先将 IDLEN 位置 1，然后将 SCS<1:0> 位置为 01 并执行 SLEEP。当时钟源切换到 Timer1 振荡器时，主振荡器关闭，OSTS 位清零，T1RUN 位置 1。

当唤醒事件发生时，外设继续将 Timer1 振荡器作为时钟源。在唤醒事件后经过一个 T<sub>CSD</sub> 的时间间隔，CPU 开始使用 Timer1 振荡器作为时钟源执行代码。唤醒不会 IDLEN 和 SCS 位。Timer1 振荡器继续运行（见图 4-8）。

**注：** Timer1 振荡器应该在进入 SEC\_IDLE 模式之前就已经运行了。如果执行 SLEEP 指令时 T1OSCEN 位没有置 1，则会忽略 SLEEP 指令并且不会进入 SEC\_IDLE 模式。如果使能了 Timer1 振荡器，但它尚未运行，外设时钟将被延时直到该振荡器起振为止。在这种情况下，最初振荡器运行很不稳定，可能会导致无法预料的结果。

图 4-7: 进入空闲模式的转换时序

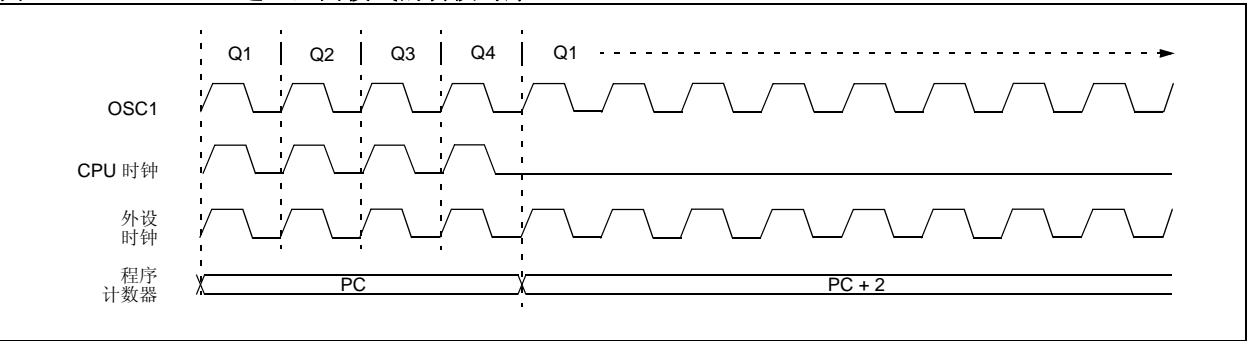
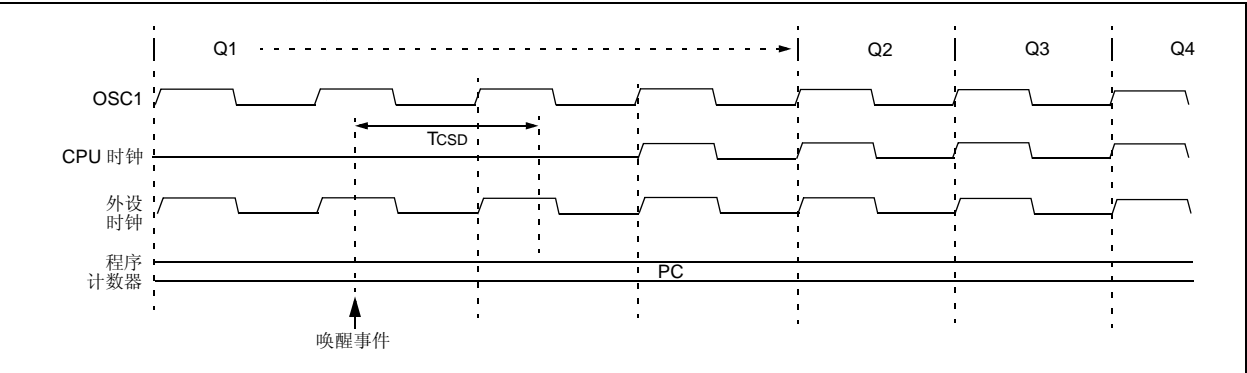


图 4-8: 从空闲模式唤醒进入运行模式的转换时序





## 4.4.3 RC\_IDLE 模式

在 RC\_IDLE 模式中，CPU 被禁止，但外设仍继续使用内部振荡器作为时钟源。该模式允许在空闲期间对功耗进行控制。

通过将 IDLEN 位置 1 并执行 SLEEP 指令，可以从 RC\_RUN 模式进入此模式。如果器件处于其他运行模式，请先将 IDLEN 位置 1，然后将 SCS 位清零并执行 SLEEP。当时钟源切换到 INTRC 电路时，主振荡器被关闭，OSTS 位被清零。

当唤醒事件发生时，外设继续将 INTOSC 作为时钟源。在唤醒事件后经过一个 T<sub>CSD</sub> 的延时，CPU 开始使用 INTOSC 作为时钟源执行代码。唤醒事件不会影响 IDLEN 和 SCS 位。如果使能了 WDT 或故障保护时钟监视器，INTOSC 源将继续运行。

## 4.5 退出空闲和休眠模式

由中断、复位或 WDT 超时触发从休眠模式或任何空闲模式退出。本节将讨论导致从功耗管理模式退出的触发条件。在每种功耗管理模式章节中我们已经讨论了时钟源子系统的作用（见第 4.2 节“运行模式”、第 4.3 节“休眠模式”和第 4.4 节“空闲模式”）。

### 4.5.1 通过中断退出

任何可用的中断源可以使器件从空闲模式或休眠模式退出，并进入运行模式。要使能此功能，必须通过将某个 INTCON 或 PIE 寄存器中的相应中断允许位置 1 来允许中断源。当相应的中断标志位置 1 时，触发退出操作。

当通过中断从空闲或休眠模式退出时，如果 GIE/GIEH 位（INTCON<7>）置 1，代码就会跳转到中断向量处执行。否则，代码就会按顺序执行或从头开始执行，而不发生跳转（见第 10.0 节“中断”）。

唤醒事件之后需要一个固定的 T<sub>CSD</sub> 间隔的延时，器件才会退出休眠和空闲模式。CPU 需要此延时来为执行代码做准备。在此延时后的第一个时钟周期恢复指令执行。

### 4.5.2 通过 WDT 超时退出

根据 WDT 超时发生时器件所处的功耗管理模式会引发不同的操作。

如果器件不在执行代码（所有空闲模式和休眠模式），超时将导致从功耗管理模式退出（见第 4.2 节“运行模式”和第 4.3 节“休眠模式”）。如果器件正在执行代码（所有运行模式），超时将导致 WDT 复位（见第 23.2 节“看门狗定时器（WDT）”）。

看门狗定时器和后分频器可由以下任一事件清零：

- 执行 SLEEP 或 CLRWDI 指令
- 当前选定的时钟源失效（如果使能了故障保护时钟监视器）

### 4.5.3 通过复位退出

通过复位从空闲或休眠模式退出，自动强制器件使用 INTRC 作为时钟源运行。

### 4.5.4 在没有振荡器起振延时的情况下退出

从功耗管理模式退出的某些情况根本不会启动 OST。有两种情形：

- 主时钟源一直工作的 PRI\_IDLE 模式；
- 主时钟源处于 EC 模式或 ECPLL 模式。

在这两种情况下，主时钟源不需要振荡器起振延时，因为它已经在运行（PRI\_IDLE 模式），或者它本来就不需要振荡器起振延时（EC 模式）。然而，当退出休眠和空闲模式时，在唤醒事件后仍需要一个固定的延时 T<sub>CSD</sub>，以便让 CPU 为执行代码做好准备。在此延时后的第一个时钟周期恢复指令执行。

# PIC18F85J11 系列

---

注:

## 5.0 复位

PIC18F85J11 系列器件有以下几种不同的复位方式：

- 上电复位（POR）
- 正常工作状态下的  $\overline{\text{MCLR}}$  复位
- 功耗管理模式下的  $\overline{\text{MCLR}}$  复位
- 看门狗定时器（WDT）复位（代码执行期间）
- 欠压复位（BOR）
- RESET 指令
- 堆栈满复位
- 堆栈下溢复位

本节将讨论由  $\overline{\text{MCLR}}$ 、POR 和 BOR 产生的各种复位以及各种起振定时器的操作。堆栈复位事件将在第 6.1.6.4 节“堆栈满和下溢复位”中讨论。WDT 复位将在第 23.2 节“看门狗定时器（WDT）”中讨论。

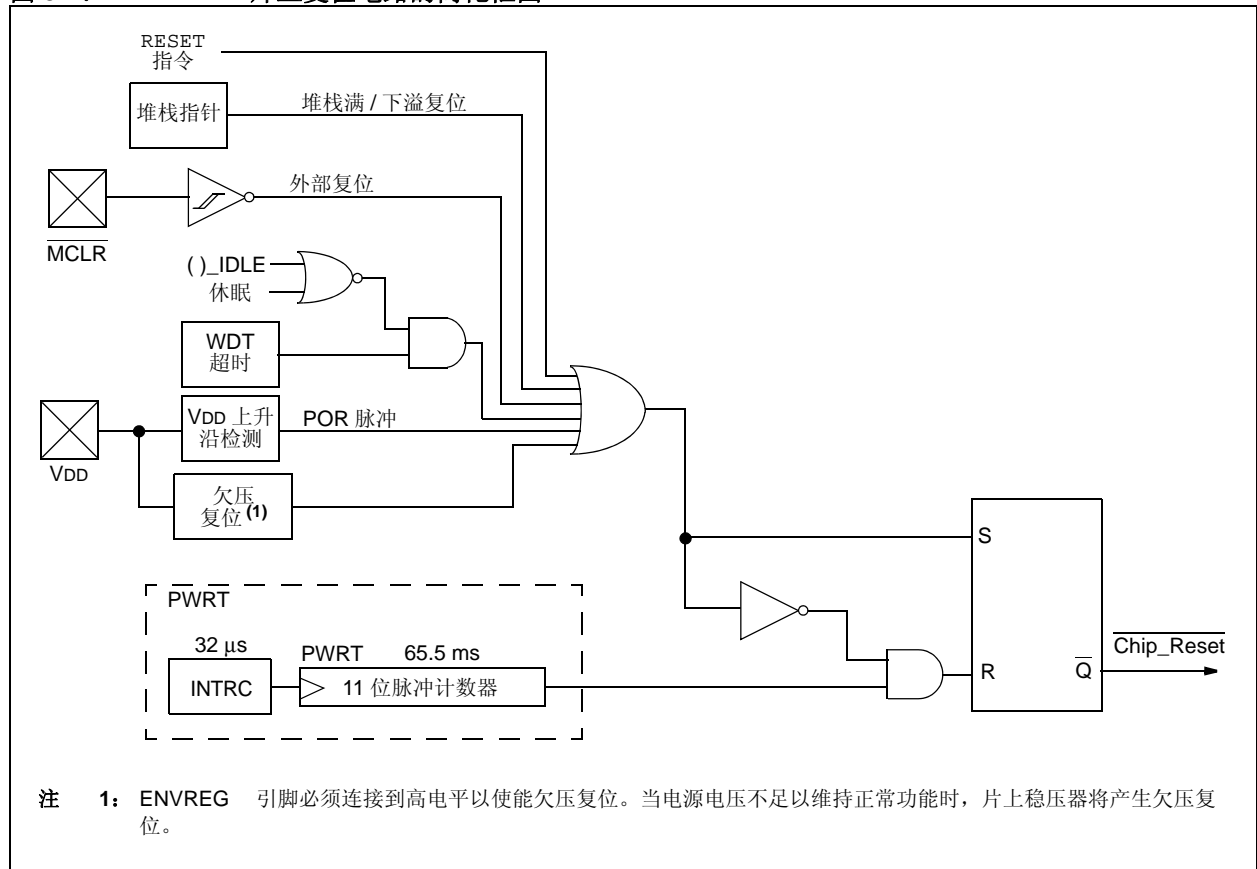
图 5-1 给出了片上复位电路的简化框图。

## 5.1 RCON 寄存器

可通过 RCON 寄存器（寄存器 5-1）跟踪器件复位事件。该寄存器的低 5 位表示特定的复位事件是否已经发生。在大部分情况下，只能通过复位事件将这些位置 1，而且随后必须在应用程序中将它们清零。读这 5 个标志位可以获知刚刚发生的复位类型。第 5.7 节“寄存器的复位状态”中对此进行了更详细的说明。

RCON 寄存器还有一个设置中断优先级的控制位（IPEN）。第 10.0 节“中断”将讨论中断优先级。

图 5-1： 片上复位电路的简化框图



# PIC18F85J11 系列

寄存器 5-1: RCON: 复位控制寄存器

R/W-0	U-0	R/W-1	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	CM	RI	TO	PD	POR	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	<b>IPEN:</b> 中断优先级允许位 1 = 允许中断优先级 0 = 禁止中断优先级 (PIC16XXXX 兼容模式)
bit 6	<b>未实现:</b> 读为 0
bit 5	<b>CM:</b> 配置不匹配标志位 1 = 未发生配置不匹配复位 0 = 发生了配置不匹配复位。必须在配置不匹配复位后由软件置 1。
bit 4	<b>RI:</b> RESET 指令标志位 1 = 未执行 RESET 指令 (只能由固件置 1) 0 = 已执行 RESET 指令, 导致器件复位 (必须在欠压复位发生之后由软件置 1)
bit 3	<b>TO:</b> 看门狗超时标志位 1 = 通过上电、CLRWDT 指令或 SLEEP 指令置 1 0 = 发生了 WDT 超时
bit 2	<b>PD:</b> 掉电检测标志位 1 = 通过上电或 CLRWDT 指令置 1 0 = 通过执行 SLEEP 指令置 1
bit 1	<b>POR:</b> 上电复位状态位 1 = 未发生上电复位 (只能由固件置 1) 0 = 已发生上电复位 (必须在发生上电复位后由软件置 1)
bit 0	<b>BOR:</b> 欠压复位状态位 1 = 未发生欠压复位 (只能由固件置 1) 0 = 已发生欠压复位 (必须在发生欠压复位后由软件置 1)

- 注 1: 建议在检测到上电复位后, 将 **POR** 位置 1, 以便检测后续的上电复位。
- 2: 如果禁止了片上稳压器, **BOR** 则总是保持为 0。更多信息, 请参见第 5.4.1 节 “检测 BOR”。
- 3: 当 **BOR** 为 0 且 **POR** 为 1 时 (假定在上电复位之后立即由软件将 **POR** 置 1), 表示已发生了欠压复位。

## 5.2 主复位 ( $\overline{\text{MCLR}}$ )

$\overline{\text{MCLR}}$  引脚提供了用外部硬件触发器件复位的方法。将该引脚拉低可以产生复位信号。PIC18 扩展单片机在  $\overline{\text{MCLR}}$  复位路径上有一个噪声滤波器，该滤波器检测并滤除小的干扰脉冲。

任何内部复位，包括 WDT 复位，都不能将  $\overline{\text{MCLR}}$  引脚驱动为低电平。

## 5.3 上电复位 (POR)

只要当  $V_{DD}$  超过设定的门限值后，就会在片上产生上电复位脉冲。这使得器件会在  $V_{DD}$  达到满足器件正常工作的数值时以初始化状态启动。

要使用 POR 电路，需要将  $\overline{\text{MCLR}}$  引脚通过一个电阻（阻值为 1 k $\Omega$  到 10 k $\Omega$ ）连接到  $V_{DD}$ 。这样可以省去产生上电复位延时通常需要的外部 RC 元件。 $V_{DD}$  的最小上升速率在参数 D004 中指定。对于上升速率缓慢的情况，请参见图 5-2。

当器件开始正常工作（即退出复位状态）时，必须满足特定的工作参数要求（电压、频率和温度等），才能确保其正常工作。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。

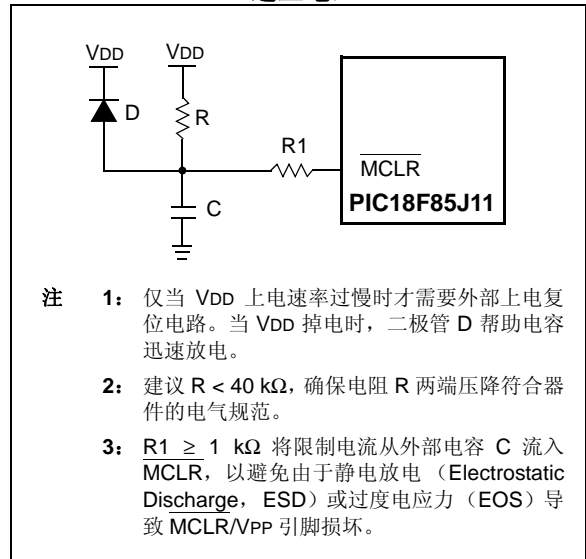
上电复位事件由  $\overline{\text{POR}}$  位 RCON<1> 捕捉。每当发生上电复位时，该位的状态就会被设置为 0，任何其他复位事件均不能改变它。 $\overline{\text{POR}}$  不能被硬件事件复位为 1。要捕捉多个事件，用户需要在任何上电复位后，手动在软件中将该位复位为 1。

## 5.4 欠压复位 (BOR)

当内部稳压器被使能时（ENVREG 引脚连接到  $V_{DD}$ ），PIC18F85J11 系列器件就具备了简单的 BOR 功能。当稳压器到器件内核的输出接近器件全速运行的临界电压时，稳压器将会触发欠压复位。随着  $V_{DD}$  上升，BOR 电路会保持器件处于复位状态，直到稳压器的输出电平足够支持器件全速运行为止。

一旦发生 BOR，上电延时定时器将使芯片在  $\text{TPWRT}$ （参数 33）期间保持复位。如果在上电延时定时器运行期间， $V_{DD}$  电压降到器件全速运行所需的门限电压以下，那么芯片将重新回到欠压复位状态且将初始化上电延时定时器。一旦  $V_{DD}$  电压上升到稳压器的输出足以满足器件正常工作的电压时，上电延时定时器将执行额外的延时。

图 5-2: 外部上电复位电路 ( $V_{DD}$  慢速上电)



### 5.4.1 检测 BOR

欠压复位或上电复位事件都会使  $\overline{\text{BOR}}$  位复位为 0。因此仅通过读  $\overline{\text{BOR}}$  的状态很难确定是否发生了欠压复位事件。更可靠的方法是同时检查  $\overline{\text{POR}}$  和  $\overline{\text{BOR}}$  的状态。假定在发生上电复位事件后，立即用软件将  $\overline{\text{POR}}$  位复位为 1。如果  $\overline{\text{BOR}}$  为 0 而  $\overline{\text{POR}}$  为 1，就可以明确断定已经发生了欠压复位事件。

如果稳压器被禁止，欠压复位功能也会被禁止。在这种情况下，不能使用  $\overline{\text{BOR}}$  位来确定欠压复位事件。上电复位事件仍然会清零  $\overline{\text{BOR}}$  位。

## 5.5 配置不匹配 (CM)

配置不匹配 (Configuration Mismatch, CM) 复位用于检测随机存储器损坏事件, 并尝试从存储器损坏事件中恢复。这些事件包括静电放电 (ESD) 事件, 它可使器件中多个位发生变化, 从而导致灾难性故障。

在 PIC18FXXJXX 闪存器件中, 在操作期间通过将器件配置寄存器 (位于配置存储空间) 的值与配对的影子寄存器进行比较来持续监视此配置寄存器。

如果在两组寄存器中检测到不匹配现象, 则自动发生 CM 复位。这些事件由 CM 位 (RCON<5>) 捕捉, 每当发生 CM 事件时, 该位的状态就会被置为 0。任何其他复位事件均不能改变它。

CM 复位操作与主复位、RESET 指令、WDT 超时或堆栈事件复位相似。与所有硬件和电源复位事件相同, 器件重启时将从程序存储器中的闪存配置字重新加载器件配置字。

## 5.6 上电延时定时器 (PWRT)

PIC18F85J11 系列器件包含了片上上电延时定时器 (PWRT), 有助于调节上电复位过程。PWRT 总是使能的。其主要功能是确保代码执行之前器件电压是稳定的。

PIC18F85J11 系列器件的上电延时定时器 (PWRT) 是一个 11 位计数器, 使用 INTRC 时钟源作为时钟输入。产生约  $2048 \times 32 \mu\text{s} = 65.6 \text{ ms}$  的时间间隔。当 PWRT 计数时, 器件保持在复位状态。

上电延时时间取决于 INTRC 时钟, 并且由于温度和工艺的不同, 不同芯片的延时时间也各不相同。详情请参见直流参数 33。

### 5.6.1 延时时序

在 POR 脉冲被清零后调用 PWRT 延时 (如果使能)。总延时根据 PWRT 的状态而有所不同。图 5-3、图 5-4、图 5-5 和图 5-6 分别描述了上电延时定时器使能时在上电期间的延时序列。

由于延时是由 POR 脉冲触发的, 因此如果  $\overline{\text{MCLR}}$  保持足够长时间的低电平, PWRT 将结束。将  $\overline{\text{MCLR}}$  电平拉高后程序将立即开始执行 (图 5-5)。这对于测试或同步多个并行工作的 PIC18FXXXX 器件来说非常有用。

图 5-3: 上电时的延时序列 ( $\overline{\text{MCLR}}$  连接到 VDD, VDD 电压上升时间 < TPWRT)

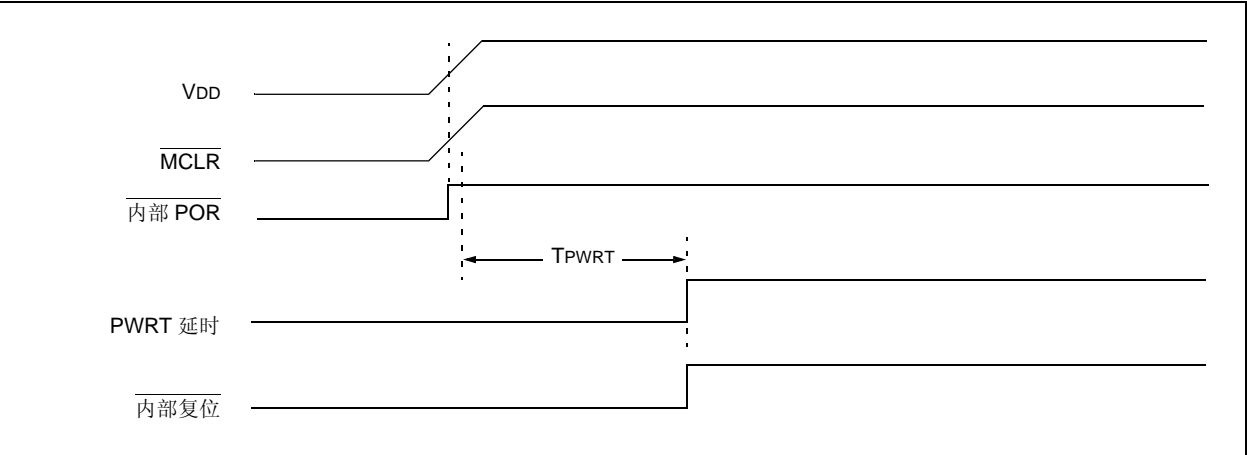


图 5-4: 上电时的延时序列 ( $\overline{\text{MCLR}}$  未连接到  $\text{VDD}$ ): 情形 1

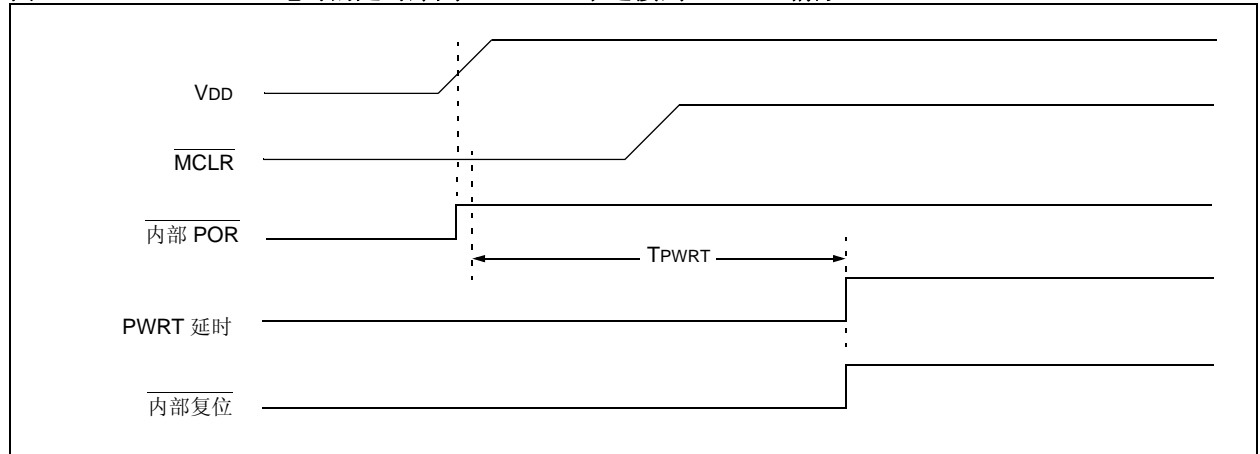


图 5-5: 上电时的延时序列 ( $\overline{\text{MCLR}}$  未连接到  $\text{VDD}$ ): 情形 2

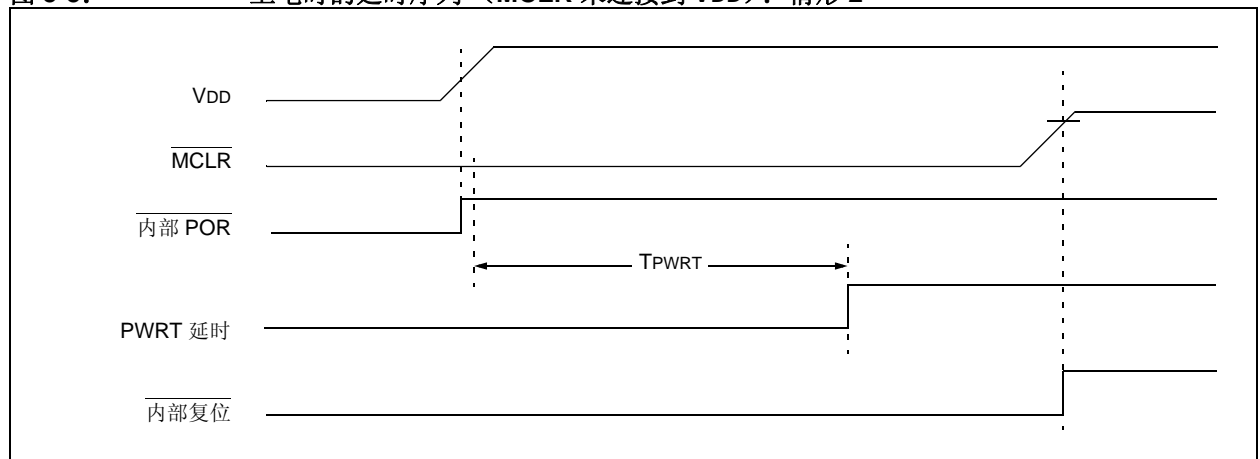
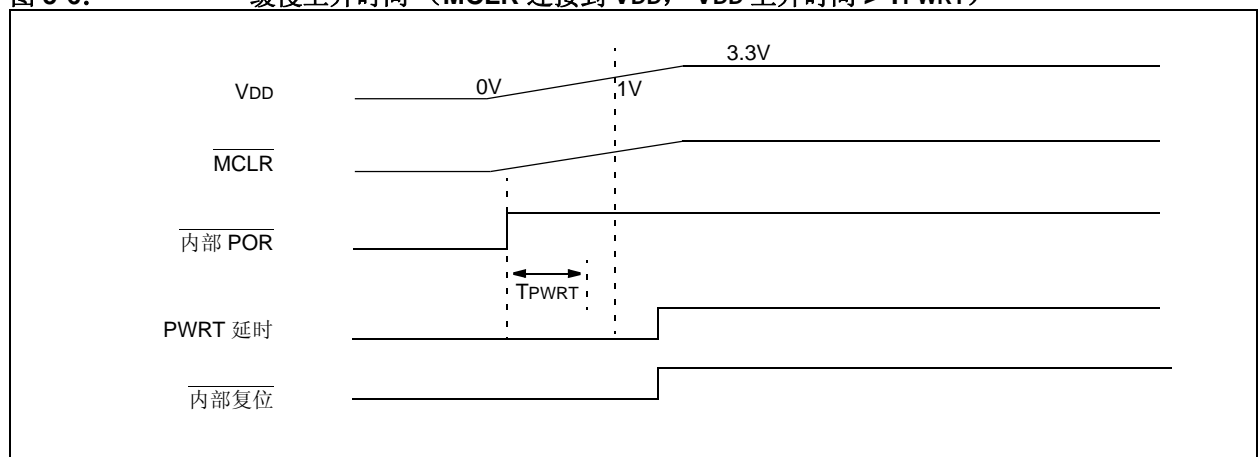


图 5-6: 缓慢上升时间 ( $\overline{\text{MCLR}}$  连接到  $\text{VDD}$ ,  $\text{VDD}$  上升时间  $> \text{TPWRT}$ )



# PIC18F85J11 系列

## 5.7 寄存器的复位状态

大多数寄存器不受复位的影响。在 POR 时这些寄存器的状态不确定，而在其他复位时它们的状态不变。而其他寄存器则根据发生的不同复位类型被强制为相应的“复位状态”。

因为 WDT 唤醒被视为恢复正常的工作，所以大部分寄存器不受 WDT 唤醒的影响。如表 5-1 所示，RCON 寄存器中的状态位（CM、RI、TO、PD、POR 和 BOR）在不同的复位情形中会分别被置 1 或清零。可在软件中使用这些状态位判断复位的类型。

表 5-2 给出了所有特殊功能寄存器的复位状态。这些复位被分为上电和欠压复位、主复位和 WDT 复位以及 WDT 唤醒复位。

表 5-1: RCON 寄存器中的状态位及其含义和初始化状态

条件	程序计数器 (1)	RCON 寄存器						STKPTR 寄存器	
		CM	RI	TO	PD	POR	BOR	STKFUL	STKUNF
上电复位	0000h	1	1	1	1	0	0	0	0
RESET 指令	0000h	u	0	u	u	u	u	u	u
欠压复位	0000h	1	1	1	1	u	0	u	u
功耗管理运行模式期间的 MCLR	0000h	u	u	1	u	u	u	u	u
功耗管理空闲模式和休眠模式期间的 MCLR	0000h	u	u	1	0	u	u	u	u
全速运行或功耗管理运行模式期间的 WDT 超时	0000h	u	u	0	u	u	u	u	u
全速执行期间的 MCLR	0000h	u	u	u	u	u	u	u	u
堆栈满复位 (STVREN = 1)	0000h	u	u	u	u	u	u	1	u
堆栈下溢复位 (STVREN = 1)	0000h	u	u	u	u	u	u	u	1
堆栈下溢错误 (不是真正的复位, STVREN = 0)	0000h	u	u	u	u	u	u	u	1
功耗管理空闲或休眠模式下的 WDT 超时	PC +2	u	u	0	0	u	u	u	u
通过中断从功耗管理模式退出	PC +2	u	u	u	0	u	u	u	u

图注: u = 不变

注 1: 当器件被中断唤醒且 GIEH 或 GIEL 位置 1 时，PC 装入中断向量（0008h 或 0018h）。



表 5-2: 所有寄存器的初始化状态

寄存器	适用器件		上电复位, 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位 CM 复位	通过 WDT 或中断唤醒器件
TOSU	PIC18F6XJ11	PIC18F8XJ11	---0 0000	---0 0000	---0 uuuu <sup>(1)</sup>
TOSH	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu <sup>(1)</sup>
TOSL	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu <sup>(1)</sup>
STKPTR	PIC18F6XJ11	PIC18F8XJ11	uu-0 0000	00-0 0000	uu-u uuuu <sup>(1)</sup>
PCLATU	PIC18F6XJ11	PIC18F8XJ11	---0 0000	---0 0000	---u uuuu
PCLATH	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
PCL	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	PC + 2 <sup>(2)</sup>
TBLPTRU	PIC18F6XJ11	PIC18F8XJ11	--00 0000	--00 0000	--uu uuuu
TBLPTRH	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
TABLAT	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
PRODH	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	PIC18F6XJ11	PIC18F8XJ11	0000 000x	0000 000u	uuuu uuuu <sup>(3)</sup>
INTCON2	PIC18F6XJ11	PIC18F8XJ11	1111 1111	1111 1111	uuuu uuuu <sup>(3)</sup>
INTCON3	PIC18F6XJ11	PIC18F8XJ11	1100 0000	1100 0000	uuuu uuuu <sup>(3)</sup>
INDF0	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A
POSTINC0	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A
POSTDEC0	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A
PREINC0	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A
PLUSW0	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A
FSR0H	PIC18F6XJ11	PIC18F8XJ11	---- xxxx	---- uuuu	---- uuuu
FSR0L	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A
POSTINC1	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A
POSTDEC1	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A
PREINC1	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A
PLUSW1	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A

图注: u = 不变, x = 未知, - = 未实现位, 读为 0, q = 取值视情况而定。  
阴影单元表示不适用于指定器件。

- 注 1: 当器件被中断唤醒且 GIEL 或 GIEH 位置 1 时, 用 PC 的当前值更新 TOSU、TOSH 和 TOSL。将 STKPTR 修改为指向硬件堆栈的下一个单元。
- 2: 当器件被中断唤醒且 GIEL 或 GIEH 位置 1 时, PC 装入中断向量 (0008h 或 0018h)。
- 3: INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 (引起唤醒)。
- 4: 具体条件下的复位值, 请参见表 5-1。
- 5: 根据选定的振荡器模式使能 PORTA、LATA 和 TRISA 中的 Bit 6 和 Bit 7。若未将这两位对应的引脚配置为 PORTA 引脚, 则它们将被禁止且读为 0。

# PIC18F85J11 系列

表 5-2: 所有寄存器的初始化状态 (续)

寄存器	适用器件		上电复位, 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位 CM 复位	通过 WDT 或中断唤醒器件
FSR1H	PIC18F6XJ11	PIC18F8XJ11	---- xxxx	---- uuuu	---- uuuu
FSR1L	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
BSR	PIC18F6XJ11	PIC18F8XJ11	---- 0000	---- 0000	---- uuuu
INDF2	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A
POSTINC2	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A
POSTDEC2	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A
PREINC2	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A
PLUSW2	PIC18F6XJ11	PIC18F8XJ11	N/A	N/A	N/A
FSR2H	PIC18F6XJ11	PIC18F8XJ11	---- xxxx	---- uuuu	---- uuuu
FSR2L	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	PIC18F6XJ11	PIC18F8XJ11	---x xxxx	---u uuuu	---u uuuu
TMR0H	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
TMR0L	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	PIC18F6XJ11	PIC18F8XJ11	1111 1111	1111 1111	uuuu uuuu
OSCCON	PIC18F6XJ11	PIC18F8XJ11	0100 q000	0100 q000	uuuu quuu
WDTCON	PIC18F6XJ11	PIC18F8XJ11	0--- ---0	0--- ---0	u--- ---u
RCON <sup>(4)</sup>	PIC18F6XJ11	PIC18F8XJ11	0-11 11q0	0-uq qquu	u-uu qquu
TMR1H	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	PIC18F6XJ11	PIC18F8XJ11	0000 0000	u0uu uuuu	uuuu uuuu
TMR2	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
PR2	PIC18F6XJ11	PIC18F8XJ11	1111 1111	1111 1111	1111 1111
T2CON	PIC18F6XJ11	PIC18F8XJ11	-000 0000	-000 0000	-uuu uuuu
SSPBUF	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPADD	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
SSPCON1	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
SSPCON2	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu

图注: u = 不变, x = 未知, - = 未实现位, 读为 0, q = 取值视情况而定。  
阴影单元表示不适用于指定器件。

- 注 1: 当器件被中断唤醒且 GIEL 或 GIEH 位置 1 时, 用 PC 的当前值更新 TOSU、TOSH 和 TOSL。将 STKPTR 修改为指向硬件堆栈的下一个单元。
- 2: 当器件被中断唤醒且 GIEL 或 GIEH 位置 1 时, PC 装入中断向量 (0008h 或 0018h)。
- 3: INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 (引起唤醒)。
- 4: 具体条件下的复位值, 请参见表 5-1。
- 5: 根据选定的振荡器模式使能 PORTA、LATA 和 TRISA 中的 Bit 6 和 Bit 7。若未将这两位对应的引脚配置为 PORTA 引脚, 则它们将被禁止且读为 0。

表 5-2: 所有寄存器的初始化状态 (续)

寄存器	适用器件		上电复位, 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位 CM 复位	通过 WDT 或中断唤醒器件
ADRESH	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	PIC18F6XJ11	PIC18F8XJ11	0-00 0000	0-00 0000	u-uu uuuu
ADCON1	PIC18F6XJ11	PIC18F8XJ11	--00 0000	--00 0000	--uu uuuu
ADCON2	PIC18F6XJ11	PIC18F8XJ11	0-00 0000	0-00 0000	u-uu uuuu
CVRCON	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
CMCON	PIC18F6XJ11	PIC18F8XJ11	0000 0111	0000 0111	uuuu uuuu
TMR3H	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
T3CON	PIC18F6XJ11	PIC18F8XJ11	0000 0000	uuuu uuuu	uuuu uuuu
PSPCON	PIC18F6XJ11	PIC18F8XJ11	0000 ----	0000 ----	uuuu ----
SPBRG1	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
RCREG1	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
TXREG1	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
TXSTA1	PIC18F6XJ11	PIC18F8XJ11	0000 0010	0000 0010	uuuu uuuu
RCSTA1	PIC18F6XJ11	PIC18F8XJ11	0000 000x	0000 000x	uuuu uuuu
EECON2	PIC18F6XJ11	PIC18F8XJ11	---- ----	---- ----	---- ----
EECON1	PIC18F6XJ11	PIC18F8XJ11	---0 x00-	---0 u00-	---0 u00-
IPR3	PIC18F6XJ11	PIC18F8XJ11	--00 -11-	--00 -11-	--uu -uu-
PIR3	PIC18F6XJ11	PIC18F8XJ11	--00 -00-	--00 -00-	--uu -00-(3)
PIE3	PIC18F6XJ11	PIC18F8XJ11	--00 -00-	--00 -00-	--uu -00-
IPR2	PIC18F6XJ11	PIC18F8XJ11	11-- 111-	11-- 111-	uu-- uuu-
PIR2	PIC18F6XJ11	PIC18F8XJ11	00-- 000-	00-- 000-	uu-- uuu-(3)
PIE2	PIC18F6XJ11	PIC18F8XJ11	00-- 000-	00-- 000-	uu-- uuu-
IPR1	PIC18F6XJ11	PIC18F8XJ11	1111 1-11	1111 1-11	uuuu u-uu
PIR1	PIC18F6XJ11	PIC18F8XJ11	0000 0-00	0000 0-00	uuuu u-uu(3)
PIE1	PIC18F6XJ11	PIC18F8XJ11	0000 0-00	0000 0-00	uuuu u-uu
MEMCON	PIC18F6XJ11	PIC18F8XJ11	0-00 --00	0-00 --00	u-uu --uu
OSCTUNE	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu

图注: u = 不变, x = 未知, - = 未实现位, 读为 0, q = 取视情况而定。  
阴影单元表示不适用于指定器件。

- 注 1: 当器件被中断唤醒且 GIEL 或 GIEH 位置 1 时, 用 PC 的当前值更新 TOSU、TOSH 和 TOSL。将 STKPTR 修改为指向硬件堆栈的下一个单元。
- 2: 当器件被中断唤醒且 GIEL 或 GIEH 位置 1 时, PC 装入中断向量 (0008h 或 0018h)。
- 3: INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 (引起唤醒)。
- 4: 具体条件下的复位值, 请参见表 5-1。
- 5: 根据选定的振荡器模式使能 PORTA、LATA 和 TRISA 中的 Bit 6 和 Bit 7。若未将这两位对应的引脚配置为 PORTA 引脚, 则它们将被禁止且读为 0。

# PIC18F85J11 系列

表 5-2: 所有寄存器的初始化状态 (续)

寄存器	适用器件		上电复位, 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位 CM 复位	通过 WDT 或中断唤醒器件
TRISJ	PIC18F6XJ11	PIC18F8XJ11	1111 1111	1111 1111	uuuu uuuu
TRISH	PIC18F6XJ11	PIC18F8XJ11	1111 1111	1111 1111	uuuu uuuu
TRISG	PIC18F6XJ11	PIC18F8XJ11	0001 1111	0001 1111	uuuu uuuu
TRISF	PIC18F6XJ11	PIC18F8XJ11	1111 111-	1111 111-	uuuu uuu-
TRISE	PIC18F6XJ11	PIC18F8XJ11	1111 1-11	1111 1-11	uuuu u-uu
TRISD	PIC18F6XJ11	PIC18F8XJ11	1111 1111	1111 1111	uuuu uuuu
TRISC	PIC18F6XJ11	PIC18F8XJ11	1111 1111	1111 1111	uuuu uuuu
TRISB	PIC18F6XJ11	PIC18F8XJ11	1111 1111	1111 1111	uuuu uuuu
TRISA <sup>(5)</sup>	PIC18F6XJ11	PIC18F8XJ11	1111 1111 <sup>(5)</sup>	1111 1111 <sup>(5)</sup>	uuuu uuuu <sup>(5)</sup>
LATJ	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATH	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATG	PIC18F6XJ11	PIC18F8XJ11	00-x xxxx	00-u uuuu	uu-u uuuu
LATF	PIC18F6XJ11	PIC18F8XJ11	xxxx xxx-	uuuu uuu-	uuuu uuu-
LATE	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATD	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATC	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATB	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATA <sup>(5)</sup>	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx <sup>(5)</sup>	uuuu uuuu <sup>(5)</sup>	uuuu uuuu <sup>(5)</sup>
PORTJ	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTH	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTG	PIC18F6XJ11	PIC18F8XJ11	000x xxxx	000u uuuu	000u uuuu
PORTF	PIC18F6XJ11	PIC18F8XJ11	xxxx xxx-	uuuu uuu-	uuuu uuu-
PORTE	PIC18F6XJ11	PIC18F8XJ11	xxxx x-xx	uuuu u-uu	uuuu u-uu
PORTD	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTB	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA <sup>(5)</sup>	PIC18F6XJ11	PIC18F8XJ11	xx0x 0000 <sup>(5)</sup>	uu0u 0000 <sup>(5)</sup>	uuuu uuuu <sup>(5)</sup>
SPBRGH1	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
BAUDCON1	PIC18F6XJ11	PIC18F8XJ11	0100 0-00	0100 0-00	uuuu u-uu
CCPR1H	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	PIC18F6XJ11	PIC18F8XJ11	--00 0000	--00 0000	--uu uuuu

图注: u = 不变, x = 未知, - = 未实现位, 读为 0, q = 取值视情况而定。  
阴影单元表示不适用于指定器件。

- 注 1: 当器件被中断唤醒且 GIEL 或 GIEH 位置 1 时, 用 PC 的当前值更新 TOSU、TOSH 和 TOSL。将 STKPTR 修改为指向硬件堆栈的下一个单元。
- 2: 当器件被中断唤醒且 GIEL 或 GIEH 位置 1 时, PC 装入中断向量 (0008h 或 0018h)。
- 3: INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 (引起唤醒)。
- 4: 具体条件下的复位值, 请参见表 5-1。
- 5: 根据选定的振荡器模式使能 PORTA、LATA 和 TRISA 中的 Bit 6 和 Bit 7。若未将这两位对应的引脚配置为 PORTA 引脚, 则它们将被禁止且读为 0。

**表 5-2: 所有寄存器的初始化状态 (续)**

寄存器	适用器件		上电复位, 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位 CM 复位	通过 WDT 或中断唤醒器件
CCPR2H	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2L	PIC18F6XJ11	PIC18F8XJ11	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	PIC18F6XJ11	PIC18F8XJ11	--00 0000	--00 0000	--uu uuuu
SPBRG2	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
RCREG2	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
TXREG2	PIC18F6XJ11	PIC18F8XJ11	0000 0000	0000 0000	uuuu uuuu
TXSTA2	PIC18F6XJ11	PIC18F8XJ11	0000 -010	0000 -010	uuuu -uuu
RCSTA2	PIC18F6XJ11	PIC18F8XJ11	0000 000x	0000 000x	uuuu uuuu

**图注:** u = 不变, x = 未知, - = 未实现位, 读为 0, q = 取值视情况而定。  
阴影单元表示不适用于指定器件。

- 注 1:** 当器件被中断唤醒且 GIEL 或 GIEH 位置 1 时, 用 PC 的当前值更新 TOSU、TOSH 和 TOSL。将 STKPTR 修改为指向硬件堆栈的下一个单元。
- 2:** 当器件被中断唤醒且 GIEL 或 GIEH 位置 1 时, PC 装入中断向量 (0008h 或 0018h)。
- 3:** INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 (引起唤醒)。
- 4:** 具体条件下的复位值, 请参见表 5-1。
- 5:** 根据选定的振荡器模式使能 PORTA、LATA 和 TRISA 中的 Bit 6 和 Bit 7。若未将这两位对应的引脚配置为 PORTA 引脚, 则它们将被禁止且读为 0。

# PIC18F85J11 系列

---

注:

6.0 存储器构成

PIC18 闪存单片机有两种类型的存储器：

- 程序存储器
- 数据 RAM

在哈佛架构的器件中，数据和程序存储器使用不同的总线，因而可同时访问这两种存储空间。

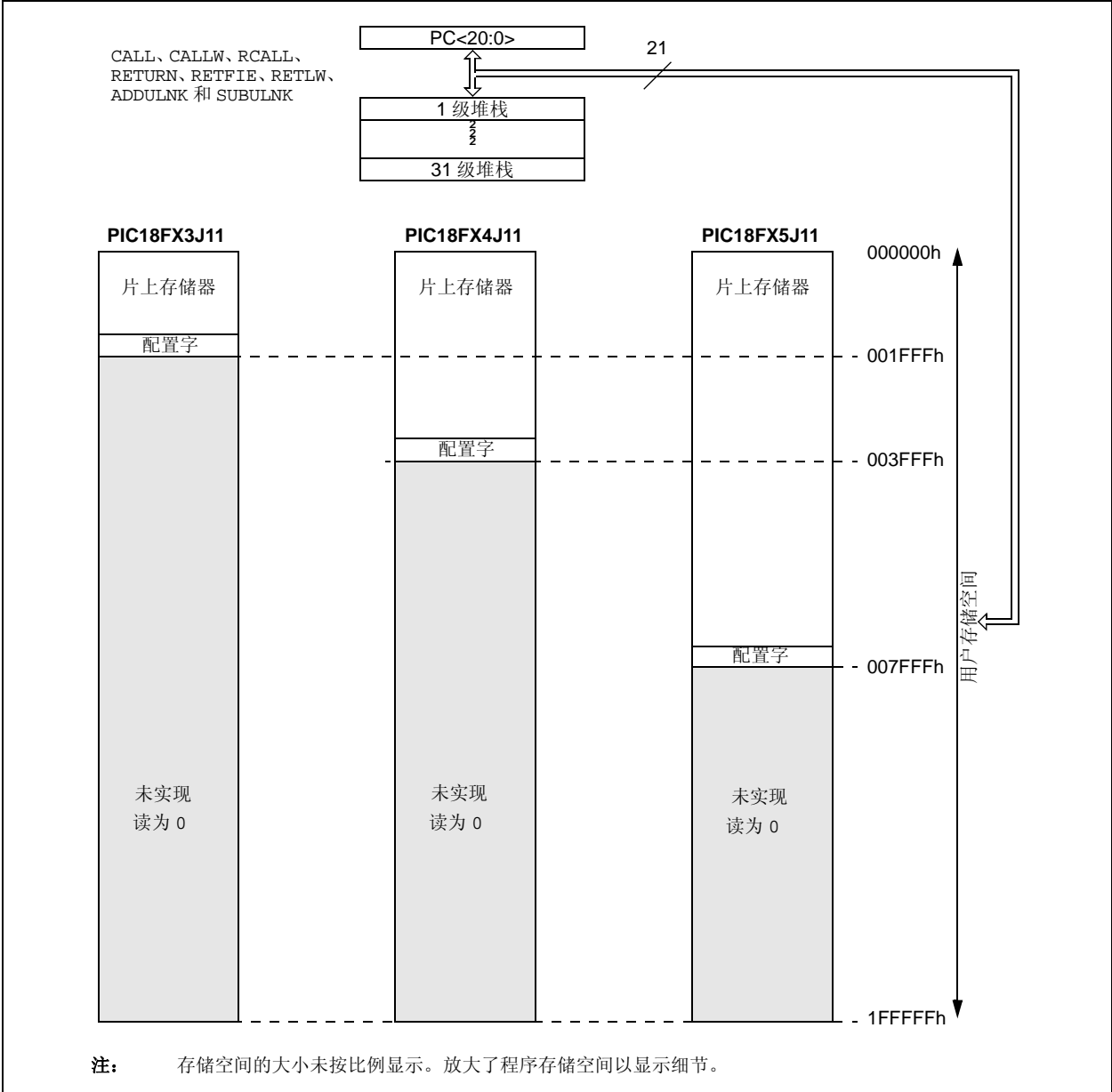
第 7.0 节“闪存程序存储器”提供了关于闪存程序存储器操作的更多详细信息。

6.1 程序存储器构成

PIC18 单片机具有一个 21 位程序计数器，可以对 2 MB 的程序存储空间进行寻址。访问物理实现的存储器的上边界和 2 MB 地址之间的存储单元将会返回全 0（NOP 指令）。

整个 PIC18F85J11 系列器件提供了一系列容量的片上闪存程序存储空间，从 8 KB（至多 4096 条单字指令）到 32 KB（32768 条单字指令）。该系列各个器件的程序存储器映射如图 6-1 所示。

图 6-1: PIC18F85J11 系列器件的存储器映射



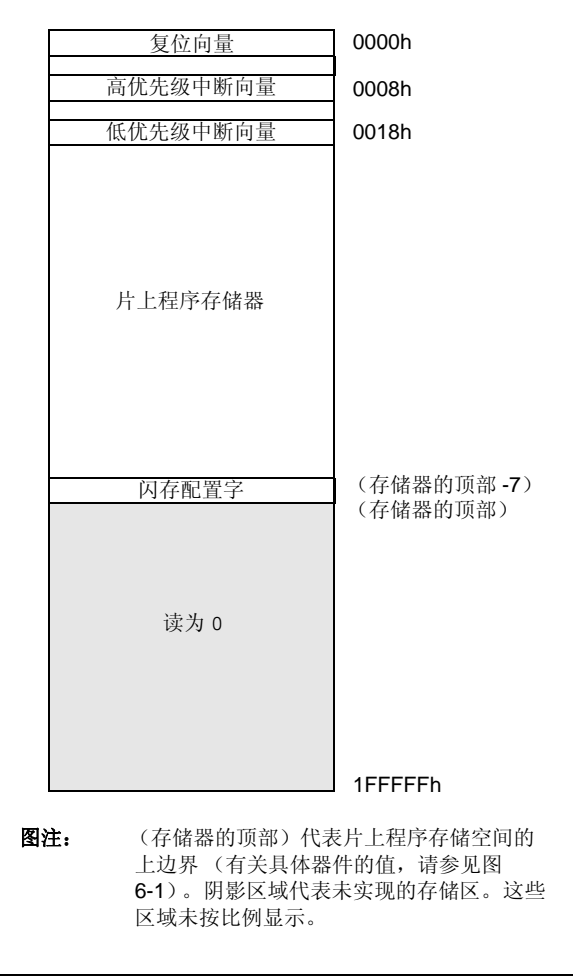
# PIC18F85J11 系列

## 6.1.1 存储器硬编码向量

所有的 PIC18 器件在它们的程序存储空间内共有 3 个硬编码的返回向量。复位向量地址是在器件发生任何复位时程序计数器返回的默认值；它位于 0000h。

PIC18 器件还有两个中断向量地址，用于处理高优先级和低优先级中断。高优先级中断向量位于 0008h，低优先级中断向量位于 0018h。它们在程序存储器映射中的相对位置如图 6-2 所示。

图 6-2: PIC18F85J11 系列器件的硬编码向量和配置字单元



## 6.1.2 闪存配置字

由于 PIC18F85J11 系列器件没有固定的配置存储器，所以保留片上程序存储器顶部的 4 个字来保存配置信息。复位时，该配置信息被复制到配置寄存器中。

配置字以由低到高的顺序存储在程序存储单元内，从最低地址开始存放 CONFIG1 的低字节，到 CONFIG4 的高字节结束。对于这些器件，只使用从 CONFIG1 到 CONFIG3 的配置字；保留 CONFIG4。PIC18F85J11 系列器件的闪存配置字的实际地址如表 6-1 所示。图 6-2 中显示了闪存配置字以及其他的存储器向量在存储器映射中的位置。

第 23.1 节“配置位”中提供了有关器件配置字的更多详细信息。

表 6-1: PIC18F85J11 系列器件的闪存配置字

器件	程序存储器 (KB)	配置字地址
PIC18F63J11	8	1FF8h 到 1FFFh
PIC18F83J11		
PIC18F64J11	16	3FF8h 到 3FFFh
PIC18F84J11		
PIC18F65J11	32	7FF8h 到 7FFFh
PIC18F85J11		



## 6.1.3 PIC18F8XJ11 程序存储器模式

该系列中的 80 引脚器件最多可寻址 2 MB 的程序存储器。这是通过外部存储器总线实现的。控制器有两种不同的工作模式：

- 单片机（MC）
- 扩展单片机（EMC）

通过设置 EMB 配置位（CONFIG3L<5:4>）决定程序存储器的模式，如寄存器 6-1 所示。（如需有关器件配置位的详细信息，还可参见第 23.1 节“配置位”。）

程序存储器模式的工作方式如下：

- **单片机模式**只访问片上闪存程序存储器。尝试读取片上存储器的顶部以上的地址单元会导致读到全 0（相当于执行 NOP 指令）。

单片机模式也是 64 引脚器件惟一可用的工作模式。

- **扩展单片机模式**允许将内部和外部程序存储器作为一个整体进行访问。器件可以访问整个片上程序存储器；除此之外，器件还可以访问外部程序存储器，寻址范围限制在 2 MB 范围内。程序执行可以按照需要自动在两个存储器之间切换。

EMB 配置位的设置还能控制外部存储器总线的地址总线宽度。在第 8.0 节“外部存储器总线”中对此有更详细的论述。

在所有模式下，单片机都能访问数据 RAM。

图 6-3 比较了不同程序存储器模式的存储器映射。表 6-2 中更充分的说明了片上存储器和外部存储器在访问限制方面的差异。

**寄存器 6-1: CONFIG3L: 配置寄存器 3 的低字节（字节地址为 300004h）<sup>(1)</sup>**

R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	U-0	U-0	U-0
WAIT	BW	EMB1	EMB0	EASHFT	—	—	—
bit 7							bit 0

### 图注:

R = 可读位                      WO = 一次性写入位                      U = 未实现位，读为 0  
 -n = 未对器件编程时的值                      1 = 置 1                      0 = 清零

- bit 7            **WAIT:** 外部总线等待使能位  
 1 = 不可使用 WAIT<1:0> 位（MEMCON<5:4>）选择等待时间，器件将不会等待  
 0 = 通过 WAIT<1:0> 位（MEMCON<5:4>）编程等待时间
- bit 6            **BW:** 数据总线宽度选择位  
 1 = 16 位外部总线模式  
 0 = 8 位外部总线模式
- bit 5:4          **EMB<1:0>:** 外部存储器总线配置位  
 00 = 扩展单片机模式——20 位地址模式  
 01 = 扩展单片机模式——16 位地址模式  
 10 = 扩展单片机模式——12 位地址模式  
 11 = 单片机模式——禁止外部总线
- bit 3            **EASHFT:** 外部地址总线平移使能位  
 1 = 使能地址平移——外部地址总线移位到起始地址为 000000h 的单元  
 0 = 禁止地址平移——外部地址总线反映 PC 值
- bit 2-0          **未实现:** 读为 0

**注 1:** 仅在 80 引脚器件上实现 CONFIG3L 及其相关位。

# PIC18F85J11 系列

## 6.1.4 扩展单片机模式和地址平移

默认情况下，处于扩展单片机模式的器件将指向外部存储空间范围内的地址的程序计数器值直接映射到外部地址总线上。实际上，这意味着外部存储器中低于片上存储器顶部的地址都不能被访问。

为了避免这种情况，扩展单片机模式使用了一个地址平移功能以使能自动地址转换。在此模式下，外部总线上的地址值减去片上程序存储器的大小并重新映射到从 0000h 开始的地址处。这样就可以利用外部存储器的全部存储空间。

图 6-3: PIC18F85J11 系列程序存储器模式下的存储器映射

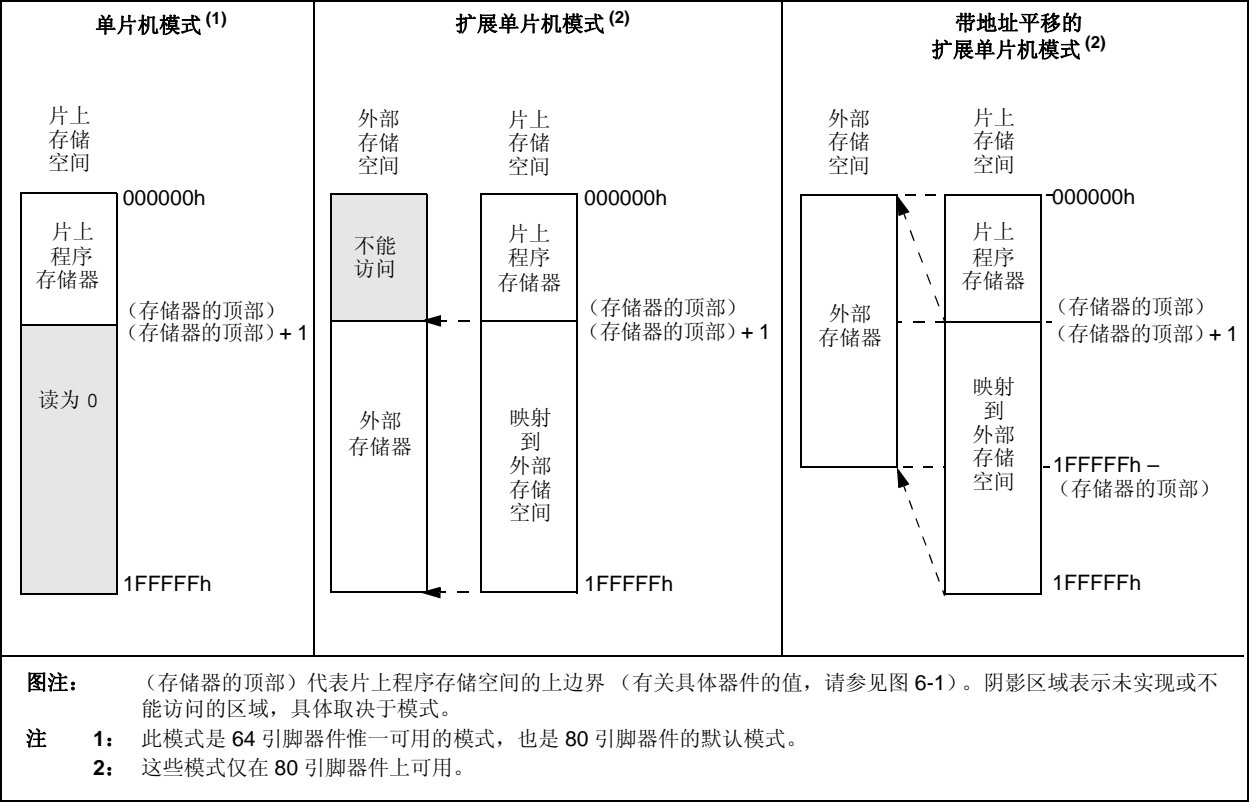


表 6-2: PIC18F8XJ11 各种程序存储器模式下的存储器访问

工作模式	内部程序存储器			外部程序存储器		
	程序执行	表读	表写	程序执行	表读	表写
单片机	可以	可以	可以	不可以	不可以	不可以
扩展单片机	可以	可以	可以	可以	可以	可以

## 6.1.5 程序计数器

程序计数器（Program Counter, PC）指定要取出执行的指令的地址。PC 为 21 位宽，保存在 3 个独立的 8 位寄存器中。存储低字节的寄存器称为 PCL 寄存器，该寄存器是可读写的。存储高字节的寄存器即 PCH 寄存器，存储 PC<15:8> 位，不可直接读写。可以通过 PCLATH 寄存器更新 PCH 寄存器。存储最高字节的寄存器称为 PCU。该寄存器储存 PC<20:16> 位，它也不可以直接读写。可以通过 PCLATU 寄存器更新 PCU 寄存器。

通过执行写 PCL 的操作，可以将 PCLATH 和 PCLATU 的内容传送到程序计数器。类似的，通过执行读 PCL 的操作，可以将程序计数器的两个高字节传送到 PCLATH 和 PCLATU。这对于计算 PC 的偏移量很有用（见第 6.1.8.1 节“计算 GOTO”）。

PC 按字节寻址程序存储器。为防止 PC 不能正确获取指令字，需要将 PCL 的最低有效位固定取值为 0。PC 以 2 为增量来连续寻址程序存储器中的指令。

CALL、RCALL、GOTO 和程序转移指令直接将地址写入程序计数器。对于这些指令，PCLATH 和 PCLATU 的内容将不会被传送到程序计数器。

## 6.1.6 返回地址堆栈

用于存放返回地址的堆栈可保存最多 31 次程序调用和中断时的 PC 值。当执行 CALL 或 RCALL 指令或响应中断时，PC 值被压入堆栈。而当执行 RETURN、RETLW 或 RETFIE 指令（若使能了扩展指令集，则还包括 ADDULNK 和 SUBULNK 指令）时，PC 值从堆栈中弹出。PCLATU 和 PCLATH 不受 RETURN 或 CALL 指令的影响。

通过 21 位的 RAM 和 5 位的堆栈指针（STKPTR）来实现 31 级的堆栈操作。堆栈既不占用程序存储空间也不占用数据存储空间。堆栈指针是可读写的，并且通过栈顶的特殊功能寄存器可以读写栈顶地址。也可使用这些寄存器将数据压入堆栈，或将数据从堆栈弹出。

执行 CALL 类型指令引起进栈操作。首先加 1 堆栈指针，并且 PC 的内容被写入堆栈指针指向的单元（PC 已指向 CALL 的下一条指令）。执行 RETURN 类型指令时，引起出栈操作。STKPTR 所指向的单元的内容会被传送给 PC，然后堆栈指针减 1。

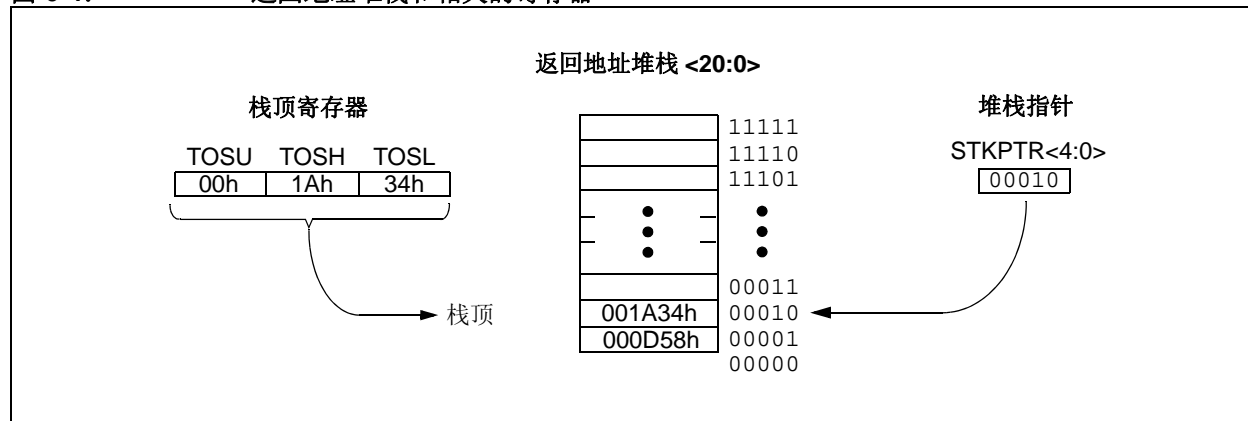
所有复位后，堆栈指针被初始化为 00000。堆栈指针值 00000 不指向任何 RAM 单元，它只是一个复位值。状态位表明堆栈是满、上溢还是下溢。

### 6.1.6.1 访问栈顶

只有返回地址堆栈的栈顶（Top-of-Stack, TOS）是可读写的。3 个寄存器 TOSU:TOSH:TOSL 用于保存 STKPTR 寄存器所指向的堆栈单元的内容（图 6-4）。这可以让用户在必要时实现软件堆栈。在 CALL、RCALL 或中断（如果使能了扩展指令集，则还包括 ADDULNK 和 SUBULNK 指令）后，软件可以通过读取 TOSU:TOSH:TOSL 寄存器来读取进栈值。这些值可以存放由用户定义的软件堆栈。返回时，软件将这些值存回 TOSU:TOSH:TOSL 并执行返回。

为防止对堆栈的意外操作，访问堆栈时用户必须禁止全局中断允许位。

图 6-4: 返回地址堆栈和相关的寄存器



# PIC18F85J11 系列

## 6.1.6.2 返回堆栈指针 (STKPTR)

STKPTR 寄存器 (寄存器 6-2) 包含堆栈指针值、STKFUL (堆栈满) 状态位和 STKUNF (堆栈下溢) 状态位。堆栈指针值可为 0 到 31 之间的值。向堆栈压入值前, 堆栈指针加 1; 而从堆栈弹出值后, 堆栈指针减 1。复位时, 堆栈指针值为零。用户可以读写堆栈指针的值。实时操作系统 (Real-Time Operating System, RTOS) 可以利用此特性对返回堆栈进行维护。

当向堆栈压入 PC 值 31 次 (且没有值从堆栈弹出) 后, STKFUL 位就会置 1。通过软件或 POR 清零 STKFUL 位。

当堆栈满时执行的操作由 STVREN (堆栈上溢复位使能) 配置位的状态决定。(有关器件配置位的说明, 请参见第 23.1 节 “配置位”。) 如果 STVREN 位已经置 1 (默认), 第 31 次进栈把 (PC + 2) 处的值压入堆栈, 将 STKFUL 位置 1, 并复位器件。STKFUL 位将保持置 1, 而堆栈指针将被清零。

如果 STVREN 位被清零, 第 31 次进栈时 STKFUL 位会被置 1, 堆栈指针则加 1 变为 31。后续进栈操作不会覆盖第 31 次进栈的值, 并且 STKPTR 将保持为 31。

当堆栈弹出次数足够卸空堆栈时, 下一次出栈会向 PC 返回一个零值, 并将 STKUNF 位置 1, 而堆栈指针则保持为 0。STKUNF 位将保持置 1, 直到被软件清零或发生 POR。

**注:** 下溢时, 将零值返回给 PC, 会使程序执行指向复位向量, 此时可以验证堆栈状态并采取相应的操作。这与复位不同, 因为 SFR 的内容不受影响。

## 6.1.6.3 PUSH 和 POP 指令

因为栈顶是可以读写的, 因此将值压入堆栈或从堆栈弹出值而不影响程序的正常执行是非常理想的。PIC18 指令集包括两条指令 PUSH 和 POP, 它们允许在软件控制下对 TOS 进行操作。可以通过修改 TOSU、TOSH 和 TOSL, 将数据或返回地址压入堆栈。

PUSH 指令将当前的 PC 值压入堆栈。这将使堆栈指针加 1, 并将当前 PC 值装入堆栈。

POP 指令通过将堆栈指针减 1 来丢弃当前的 TOS 值。然后前一个进栈值成为了 TOS 值。

寄存器 6-2: STKPTR: 堆栈指针寄存器

R/C-0	R/C-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STKFUL <sup>(1)</sup>	STKUNF <sup>(1)</sup>	—	SP4	SP3	SP2	SP1	SP0
bit 7							bit 0

<b>图注:</b>	C = 只可清零位		
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	<b>STKFUL:</b> 堆栈满标志位 <sup>(1)</sup>
	1 = 堆栈满或上溢
	0 = 堆栈未满或未上溢
bit 6	<b>STKUNF:</b> 堆栈下溢标志位 <sup>(1)</sup>
	1 = 发生堆栈下溢
	0 = 未发生堆栈下溢
bit 5	<b>未实现:</b> 读为 0
bit 4-0	<b>SP&lt;4:0&gt;:</b> 堆栈指针地址位

**注 1:** 通过用户软件或 POR 清零 bit 7 和 bit 6。

## 6.1.6.4 堆栈满和下溢复位

通过将配置寄存器 1L 中的 STVREN 位置 1 可以使得器件在出现堆栈上溢和堆栈下溢条件时复位。当 STVREN 位置 1 时，堆栈满或堆栈下溢条件会将相应的 STKFUL 或 STKUNF 位置 1，然后使器件复位。当 STVREN 位清零时，堆栈满或堆栈下溢条件会将相应的 STKFUL 或 STKUNF 位置 1，但不会使器件复位。通过用户软件或上电复位清零 STKFUL 或 STKUNF 位。

## 6.1.7 快速寄存器堆栈

为 STATUS、WREG 和 BSR 寄存器提供了快速寄存器堆栈，从而可实现从中断“快速返回”。此堆栈深度仅为 1 级，并且不可读写。当处理器转入中断向量处执行时，此堆栈装入对应寄存器的当前值。所有中断源都会将值压入此堆栈寄存器。如果使用 RETFIE，FAST 指令从中断返回，这些寄存器中的值就会被装回相应的工作寄存器。

如果同时允许了低优先级中断和高优先级中断，从低优先级中断返回时，无法可靠地使用堆栈寄存器。如果在处理低优先级中断时，发生了高优先级中断，则低优先级中断储存在堆栈寄存器中的值将被覆盖。在这种情况下，用户必须在低优先级中断期间用软件保存关键寄存器的值。

如果未使用中断优先级，所有中断都可以使用快速寄存器堆栈从中断返回。如果没有使用中断，快速寄存器堆栈可以用于在子程序调用结束后恢复 STATUS、WREG 和 BSR 寄存器。要将快速寄存器堆栈用于子程序调用，必须执行 ALL label，FAST 指令将 STATUS、WREG 和 BSR 寄存器的内容存入快速寄存器堆栈。然后执行 RETURN，FAST 指令，从快速寄存器堆栈恢复这些寄存器。

例 6-1 给出了一个在子程序调用和返回期间使用快速寄存器堆栈的源代码示例。

**例 6-1: 快速寄存器堆栈代码示例**

CALL SUB1, FAST	;STATUS, WREG, BSR
	;SAVED IN FAST REGISTER
	;STACK
•	
•	
SUB1	•
	•
RETURN FAST	;RESTORE VALUES SAVED
	;IN FAST REGISTER STACK

## 6.1.8 程序存储器中的查找表

可能在有些编程场合中需要在程序存储器中创建数据结构或查找表。对于 PIC18 器件，有两种方法可以实现查找表：

- 计算 GOTO
- 表读

### 6.1.8.1 计算 GOTO

计算 GOTO 是通过向程序计数器加一个偏移量来实现的。例 6-2 给出了一个示例。

使用 ADDWF PCL 指令和一组 RETLW nn 指令可以创建一个查找表。在调用该表前，会先将查找表的偏移量装入 W 寄存器。被调用程序的第一条指令是 ADDWF PCL 指令。接下去执行的是一条 RETLW nn 指令，它将值 nn 返回给调用函数。

偏移量（WREG 中的值）指定程序计数器应该增加的字节数，其值应当为 2 的整数倍（LSb = 0）。

在这种方法中，每个指令单元只能存储一个数据字节，并且要求返回地址堆栈还有空闲单元。

**例 6-2: 使用偏移量的计算 GOTO**

	MOVWF	OFFSET, W
	CALL	TABLE
ORG	nn00h	
TABLE	ADDWF	PCL
	RETLW	nnh
	RETLW	nnh
	RETLW	nnh
	•	
	•	
	•	

### 6.1.8.2 表读

有一种更好的方法可以将数据存储在程序存储器中，该方法允许在每个指令单元存储 2 个字节的数据。

编程时，每个程序字可以存储 2 个字节的查找表数据。表指针（TBLPTR）指定字节地址，而表锁存器（TABLAT）则储存从程序存储器读取的数据。一次只能从程序存储器读取一个字节。

在第 7.1 节“表读和表写”中将进一步讨论表读操作。

# PIC18F85J11 系列

## 6.2 PIC18 指令周期

### 6.2.1 时钟机制

单片机时钟输入信号，无论来自内部或外部时钟源，都会在器件内部被 4 分频用来产生 4 个不重叠的正交时钟信号，即 Q1、Q2、Q3 和 Q4。在内部，程序计数器在每个 Q1 递增；并在 Q4 期间，从程序存储器取指并将指令锁存到指令寄存器（Instruction Register, IR）中。指令的译码和执行在下一个 Q1 到 Q4 周期完成。图 6-5 所示为时钟和指令执行的流程图。

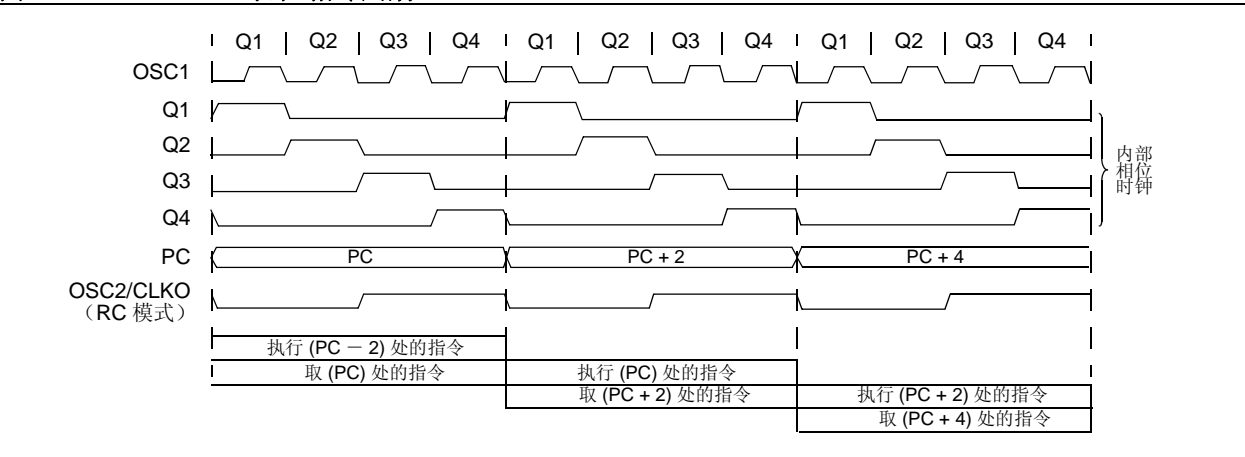
### 6.2.2 指令流 / 流水线

一个“指令周期”由 4 个 Q 周期组成（即 Q1 到 Q4）。指令的取指和执行是以流水线的形式进行的，在一个指令周期进行取指，而在下一个指令周期译码并执行指令。但由于是流水线操作，所以每条指令的等效执行时间都是一个指令周期。如果某条指令改变了程序计数器的值（如 GOTO 指令），则需要两个指令周期才能完成该指令（例 6-3）。

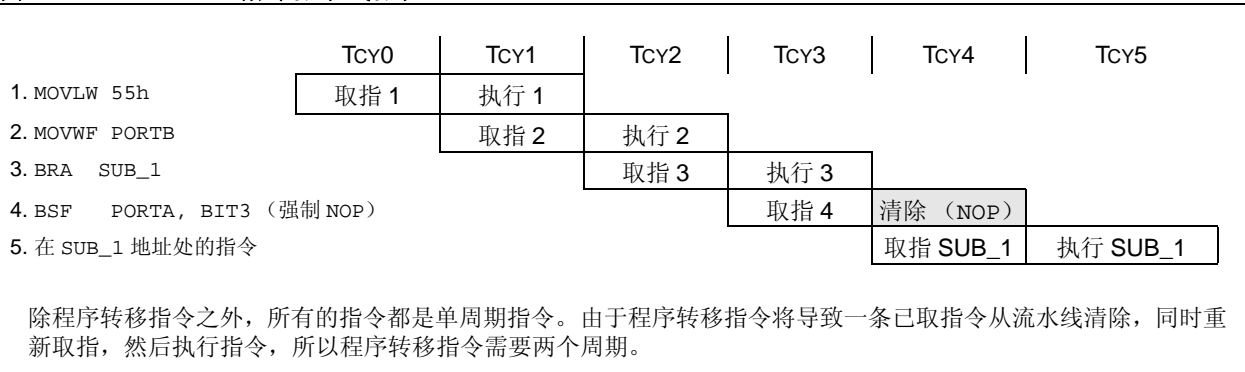
在 Q1 周期，程序计数器（PC）递增，开始取指。

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

图 6-5: 时钟 / 指令周期



例 6-3: 指令流水线流程



## 6.2.3 程序存储器中的指令

程序存储器按字节寻址。指令以 2 字节或 4 字节形式存储在程序存储器中。指令字的最低有效字节始终存储在地址为偶数的程序存储单元中（LSB = 0）。要保证正确指向指令单元，PC 必须以 2 为单位递增，并且 LSB 总是读为 0（见第 6.1.5 节“程序计数器”）。

图 6-6 给出了指令字存储在程序存储器中的一个示例。

CALL 和 GOTO 指令在指令中嵌入了程序存储器的绝对地址。指令总是存储为一个字长，因而指令所包含的数据为字地址。字地址会写入 PC<20:1>，由 PC 在程序存储器中访问目标字节地址。图 6-6 中的指令 2 给出了指令 GOTO 0006h 在程序存储器中的译码过程。程序转移指令也采取同样的方式对相对地址偏移量进行译码。在转移指令中的偏移量代表单字指令数，PC 将以此作为偏移量跳转到指定的地址单元。第 25.0 节“指令集汇总”提供了指令集的更多详情。

图 6-6: 程序存储器中的指令

				LSB = 1	LSB = 0	字地址 ↓
程序存储器 字节单元 →						000000h
						000002h
						000004h
						000006h
指令 1:	MOVLW	055h		0Fh	55h	000008h
指令 2:	GOTO	0006h		EFh	03h	00000Ah
				F0h	00h	00000Ch
指令 3:	MOVFF	123h, 456h		C1h	23h	00000Eh
				F4h	56h	000010h
						000012h
						000014h

## 6.2.4 双字指令

标准的 PIC18 指令集有 4 条双字指令：CALL、MOVFF、GOTO 和 LSFR。在所有情况下，这些指令第二个字的高 4 位总是 1111，而其余 12 位是立即数数据，通常为一个数据存储器地址。

指令的高 4 位 1111 代表一条特殊的 NOP 指令。指令的正确执行顺序为：执行完第一个字之后立即按顺序访问并使用第二个字中的数据。如果由于某些原因跳过了第

一个字并自行执行指令的第二个字，那么将转而执行一条 NOP 指令。如果双字指令跟在更改 PC 的条件指令后，就有必要执行此操作。例 6-4 说明了其执行过程。

**注：** 欲知扩展指令集中的双字指令的信息，请参见第 6.5 节“程序存储器和扩展指令集”。

例 6-4: 双字指令

情形 1:			
目标代码	源代码		
0110 0110 0000 0000	TSTFSZ	REG1	; is RAM location 0?
1100 0001 0010 0011	MOVFF	REG1, REG2	; No, skip this word
1111 0100 0101 0110			; Execute this word as a NOP
0010 0100 0000 0000	ADDWF	REG3	; continue code
情形 2:			
目标代码	源代码		
0110 0110 0000 0000	TSTFSZ	REG1	; is RAM location 0?
1100 0001 0010 0011	MOVFF	REG1, REG2	; Yes, execute this word
1111 0100 0101 0110			; 2nd word of instruction
0010 0100 0000 0000	ADDWF	REG3	; continue code

## 6.3 数据存储器构成

**注：** 当使能 PIC18 扩展指令集时，数据存储器某些方面的操作会有所更改。更多信息，请参见第 6.6 节“数据存储器 and 扩展指令集”。

PIC18 器件中的数据存储器以静态 RAM 的方式实现。在数据存储器中，每个寄存器都有一个 12 位的地址，数据存储器容量最高可达 4096 个字节。存储空间被分为 16 个存储区，每个存储区包含 256 个字节。PIC18FX3J11/X4J11 器件包含最多 16 KB 的程序存储器，实现了 4 个完整的存储区，总共 1024 个字节。PIC18FX5J11 器件包含 32 KB 的程序存储器，实现了 8 个完整的存储区，总共 2048 个字节。图 6-7 和图 6-8 显示了器件的数据存储器构成。

数据存储器由特殊功能寄存器（SFR）和通用寄存器（General Purpose Register, GPR）组成。SFR 用于单片机和外设功能模块的控制和状态显示，GPR 则用于在用户应用程序中存储数据和高速暂存操作。任何未实现单元的读取值均为 0。

此指令集和架构支持对所有存储区的操作。可以通过直接、间接或变址寻址模式访问整个数据存储器。本节后面的部分将讨论寻址模式。

为了确保能在一个周期存取常用寄存器（选定的 SFR 和 GPR），PIC18 器件实现了一个快速操作存储区。这是一个 256 字节的存储空间，可实现对选定 SFR 和 GPR Bank 0 的低地址部分的快速存取，而无需使用 BSR。第 6.3.2 节“快速操作存储区”提供了对快速操作 RAM 的详细说明。

### 6.3.1 存储区选择寄存器

存储空间很大的数据存储器需要有效的寻址机制，以便对所有地址进行快速存取。理想状况下，这意味着不必为每次读写操作提供整个地址。PIC18 器件是使用 RAM 分区机制实现快速存取的。该机制将存储空间分成连续的 16 个 256 字节的存储区。根据不同的指令，可以通过完整的 12 位地址、或通过 8 位低字节地址和 4 位存储区指针来直接寻址每个地址单元。

PIC18 指令集中的大部分指令都使用存储区指针，即存储区选择寄存器（Bank Select Register, BSR）。此 SFR 保存地址的高 4 位；而指令本身则包括低 8 位。仅使用了 BSR 的低 4 位（BSR<3:0>）。其高 4 位未用；始终读为 0 且不能被写入。可以通过 MOVLB 指令直接装入 BSR。

BSR 的值指向数据存储器中的存储区。指令中的 8 位地址表示存储区中的存储单元，可以将其看做是以存储区的下边界为起点的偏移量。图 6-9 所示是 BSR 的值与数据存储器中的分区之间的关系。

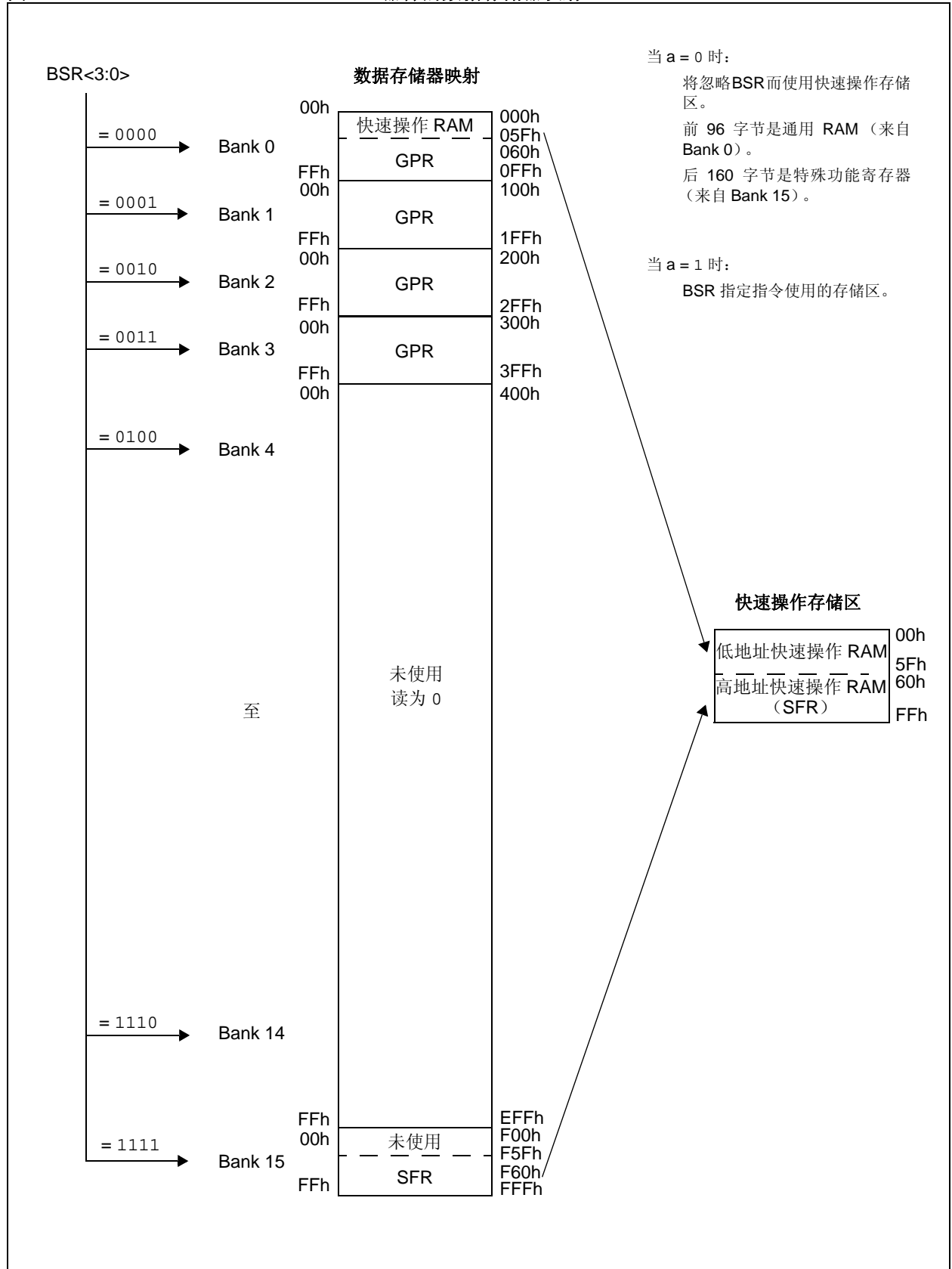
由于最多有 16 个寄存器共享同一个低位地址，用户必须非常仔细以确保在执行数据读或写之前选择正确的存储区。例如，当 BSR 为 0Fh 时将程序数据写入地址为 F9h 的 8 位地址单元，将导致程序计数器复位。

虽然可以选择任何存储区，但只有实际实现的存储区才可以被读写。对未实现存储区的写入操作将被忽略，而读未实现的存储区会返回 0。虽然是这样，STATUS 寄存器仍然会受到影响，如同成功执行了读写操作一样。图 6-7 中的数据存储器映射指出了可实现的存储区。

在 PIC18 的内核指令集中，只有 MOVFF 指令指定源寄存器和目标寄存器的完整 12 位地址。此指令在执行时完全忽略 BSR 的值。所有其他指令仅包含作为操作数的低位地址，而且必须使用 BSR 或快速操作存储区来寻址目标寄存器。



图 6-7: PIC18FX3J11/X4J11 器件的数据存储器映射



# PIC18F85J11 系列

图 6-8: PIC18FX5J11 器件的数据存储器映射

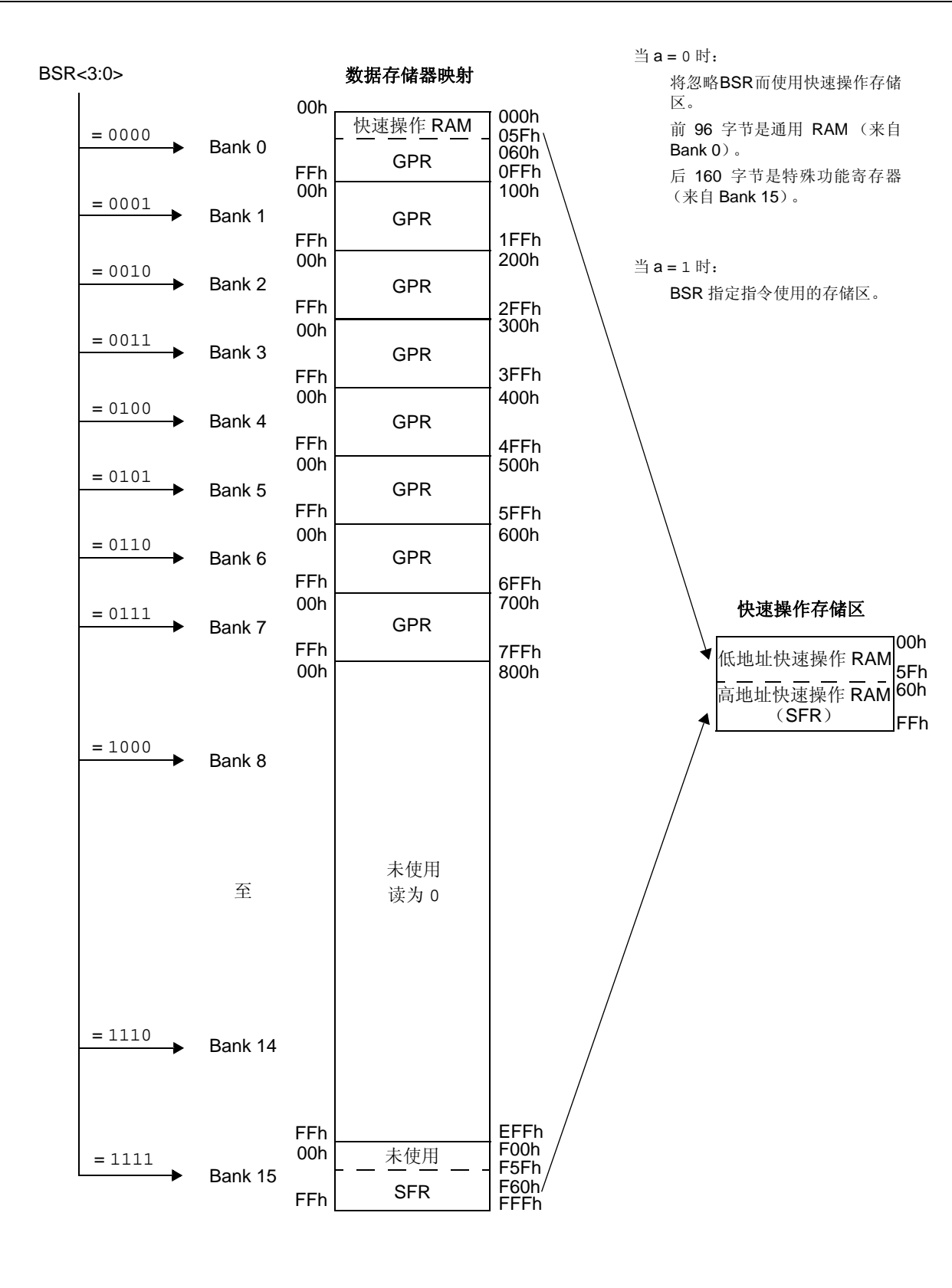
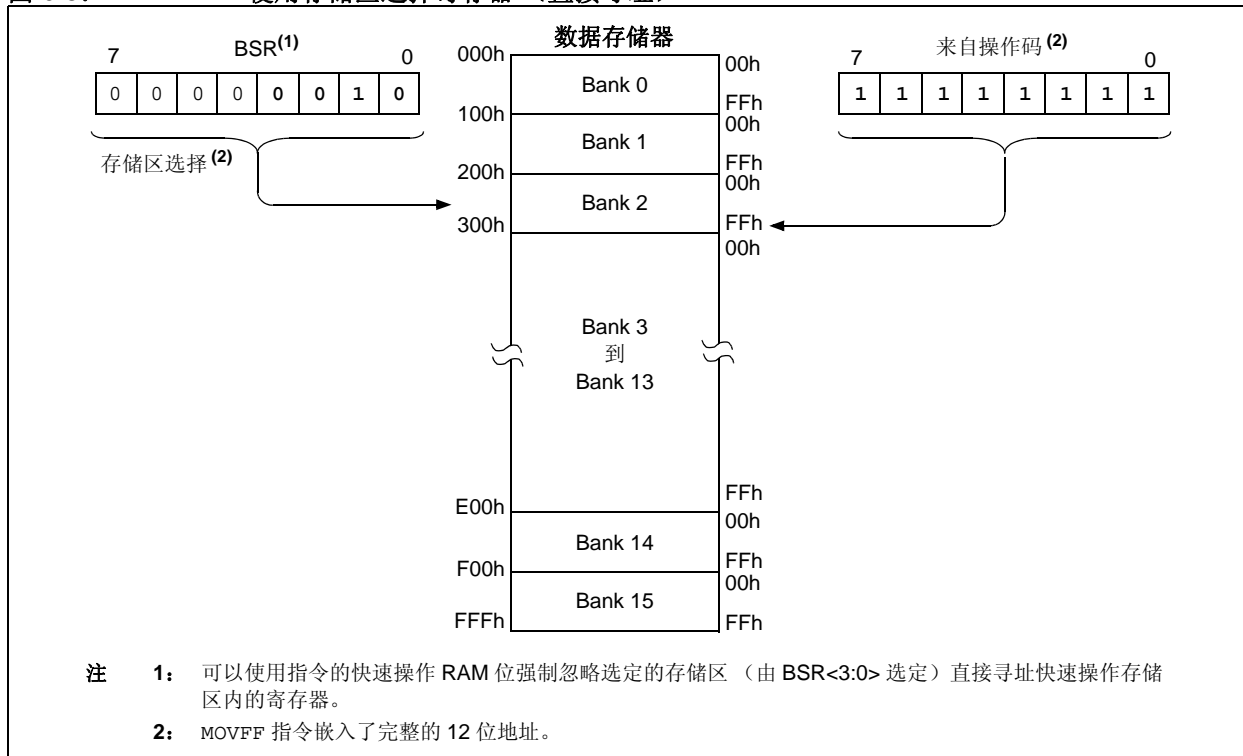


图 6-9: 使用存储区选择寄存器（直接寻址）



## 6.3.2 快速操作存储区

使用 BSR 和指令内嵌的 8 位地址可以使用户对整个数据存储区进行寻址，这同时意味着用户必须始终确保选择了正确的存储区。否则，可能会从错误的单元读取数据或将数据写入错误的单元。如果本来是向 GPR 进行操作，却写入了 SFR，后果将是非常严重的。但是在每次向数据存储器进行读或写操作时，验证和/或更改 BSR 可能会降低代码的执行效率。

为了连续访问大多数常用的数据存储单元，必须为数据存储器配置快速操作存储区，这使得用户无需指定 BSR 即可访问被映射的存储区。快速操作存储区由 Bank 0 的前 96 个字节（00h-5Fh）和 Bank 15 的后 160 个字节（60h-FFFh）组成。低半部分就是“快速操作 RAM”，由 GPR 组成。高半部分被映射为器件的 SFR。这两个区域在快速操作存储区中连续映射，并且可以用 8 位地址进行线性寻址（图 6-7）。

快速操作存储区供包括快速操作 RAM 位（指令中的 a 参数）的 PIC18 内核指令使用。当 a 等于 1 时，指令使用 BSR 和包含在操作码中的 8 位地址来对数据存储器进行寻址。但是当 a 为 0 时，指令被强制使用快速操作

存储区地址映射；BSR 的当前值被忽略。

该“强制”寻址方式可使指令在一个周期内对数据地址进行操作，而无需首先更新 BSR。这意味着用户可以更有效地对 8 位地址为 60h 及以上的 SFR 进行取值和操作。地址低于 60h 的快速操作 RAM 非常适合于存储那些用户可能需要快速存取的数据值（如直接计算结果或常用程序变量）。快速操作 RAM 还可以实现更快速、代码效率更高的现场保护和变量切换。

当使能扩展指令集（XINST 配置位 = 1）时，快速操作存储区的映射略有不同。第 6.6.3 节“在立即数变址寻址模式下映射快速操作存储区”中将对此进行更详细的说明。

## 6.3.3 通用寄存器

PIC18 器件可能在 GRP 区中划分了一部分存储区。这部分存储区为数据 RAM，所有指令都可以访问它。GPR 区从 Bank 0 的底部（地址 000h）开始向上直到 SFR 区的底部。上电复位不会初始化 GPR，其他复位也不会改变其内容。

# PIC18F85J11 系列

## 6.3.4 特殊功能寄存器

特殊功能寄存器（SFR）是 CPU 和外设模块用来控制器件操作的寄存器。这类寄存器以静态 RAM 的形式实现。SFR 起始于数据存储器的顶部（FFFh）并且向下扩展到 Bank 15 的上半部分（F60h 到 FFFh）。表 6-3 和表 6-4 列出了这些寄存器。

SFR 可分为两类：一类与“内核”器件功能（ALU、复位和中断）有关，另一类与外设功能有关。在相关的章节中将对复位和中断寄存器进行说明，而本章后面的部分将对 ALU 的 STATUS 寄存器进行说明。与外设功能部件的操作相关的寄存器在相应的外设章节中进行说明。

SFR 通常位于受其控制的外设中。未使用的 SFR 单元未实现，且读为 0。

表 6-3: PIC18F85J11 系列器件的特殊功能寄存器映射

地址	名称	地址	名称	地址	名称	地址	名称	地址	名称
FFFh	TOSU	FDFh	INDF2 <sup>(1)</sup>	FBFh	— <sup>(2)</sup>	F9Fh	IPR1	F7Fh	SPBRGH1
FFEh	TOSH	FDEh	POSTINC2 <sup>(1)</sup>	FBEh	— <sup>(2)</sup>	F9Eh	PIR1	F7Eh	BAUDCON1
FFDh	TOSL	FDDh	POSTDEC2 <sup>(1)</sup>	FBDh	— <sup>(2)</sup>	F9Dh	PIE1	F7Dh	— <sup>(2)</sup>
FFCh	STKPTR	FDCh	PREINC2 <sup>(1)</sup>	FBCCh	— <sup>(2)</sup>	F9Ch	MEMCON <sup>(3)</sup>	F7Ch	— <sup>(2)</sup>
FFBh	PCLATU	FDBh	PLUSW2 <sup>(1)</sup>	FBCh	— <sup>(2)</sup>	F9Bh	OSCTUNE	F7Bh	— <sup>(2)</sup>
FFAh	PCLATH	FDAh	FSR2H	FBAh	— <sup>(2)</sup>	F9Ah	TRISJ <sup>(3)</sup>	F7Ah	— <sup>(2)</sup>
FF9h	PCL	FD9h	FSR2L	FB9h	— <sup>(2)</sup>	F99h	TRISH <sup>(3)</sup>	F79h	— <sup>(2)</sup>
FF8h	TBLPTRU	FD8h	STATUS	FB8h	— <sup>(2)</sup>	F98h	TRISG	F78h	— <sup>(2)</sup>
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	— <sup>(2)</sup>	F97h	TRISF	F77h	— <sup>(2)</sup>
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	— <sup>(2)</sup>	F96h	TRISE	F76h	— <sup>(2)</sup>
FF5h	TABLAT	FD5h	T0CON	FB5h	CVRCON	F95h	TRISD	F75h	— <sup>(2)</sup>
FF4h	PRODH	FD4h	— <sup>(2)</sup>	FB4h	CMCON	F94h	TRISC	F74h	— <sup>(2)</sup>
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB	F73h	— <sup>(2)</sup>
FF2h	INTCON	FD2h	— <sup>(2)</sup>	FB2h	TMR3L	F92h	TRISA	F72h	— <sup>(2)</sup>
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	LATJ <sup>(3)</sup>	F71h	— <sup>(2)</sup>
FF0h	INTCON3	FD0h	RCON	FB0h	PSPCON	F90h	LATH <sup>(3)</sup>	F70h	— <sup>(2)</sup>
FEFh	INDF0 <sup>(1)</sup>	FCFh	TMR1H	FAFh	SPBRG1	F8Fh	LATG	F6Fh	— <sup>(2)</sup>
FEeh	POSTINC0 <sup>(1)</sup>	FCEh	TMR1L	FAeh	RCREG1	F8Eh	LATF	F6Eh	— <sup>(2)</sup>
FEDh	POSTDEC0 <sup>(1)</sup>	FCDh	T1CON	FADh	TXREG1	F8Dh	LATE	F6Dh	— <sup>(2)</sup>
FECh	PREINC0 <sup>(1)</sup>	FCCh	TMR2	FACH	TXSTA1	F8Ch	LATD	F6Ch	— <sup>(2)</sup>
FEbh	PLUSW0 <sup>(1)</sup>	FCBh	PR2	FABh	RCSTA1	F8Bh	LATC	F6Bh	— <sup>(2)</sup>
FEAh	FSR0H	FCAh	T2CON	FAAh	— <sup>(2)</sup>	F8Ah	LATB	F6Ah	CCPR1H
FE9h	FSR0L	FC9h	SSPBUF	FA9h	— <sup>(2)</sup>	F89h	LATA	F69h	CCPR1L
FE8h	WREG	FC8h	SSPADDD	FA8h	— <sup>(2)</sup>	F88h	PORTJ <sup>(3)</sup>	F68h	CCP1CON
FE7h	INDF1 <sup>(1)</sup>	FC7h	SSPSTAT	FA7h	EECON2	F87h	PORTH <sup>(3)</sup>	F67h	CCPR2H
FE6h	POSTINC1 <sup>(1)</sup>	FC6h	SSPCON1	FA6h	EECON1	F86h	PORTG	F66h	CCPR2L
FE5h	POSTDEC1 <sup>(1)</sup>	FC5h	SSPCON2	FA5h	IPR3	F85h	PORTF	F65h	CCP2CON
FE4h	PREINC1 <sup>(1)</sup>	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE	F64h	SPBRG2
FE3h	PLUSW1 <sup>(1)</sup>	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD	F63h	RCREG2
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC	F62h	TXREG2
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB	F61h	TXSTA2
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA	F60h	RCSTA2

- 注 1: 非物理寄存器。  
2: 未实现的寄存器，读为 0。  
3: 此寄存器在 64 引脚器件上不存在。

表 6-4: PIC18F85J11 系列寄存器汇总

寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 与 BOR 时的值	详情请见 (页)
TOSU	—	—	—	栈顶最高字节 (TOS<20:16>)					---0 0000	57, 67
TOSH	栈顶高字节 (TOS<15:8>)								0000 0000	57, 67
TOSL	栈顶低字节 (TOS<7:0>)								0000 0000	57, 67
STKPTR	STKFUL	STKUNF	—	返回堆栈指针					uu-0 0000	57, 68
PCLATU	—	—	bit 21 <sup>(1)</sup>	PC<20:16> 的保持寄存器					---0 0000	57, 67
PCLATH	PC<15:8> 的保持寄存器								0000 0000	57, 67
PCL	PC 的低字节 (PC<7:0>)								0000 0000	57, 67
TBLPTRU	—	—	bit 21	程序存储器表指针最高字节 (TBLPTR<20:16>)					--00 0000	57, 92
TBLPTRH	程序存储器表指针高字节 (TBLPTR<15:8>)								0000 0000	57, 92
TBLPTRL	程序存储器表指针低字节 (TBLPTR<7:0>)								0000 0000	57, 92
TABLAT	程序存储器表锁存器								0000 0000	57, 92
PRODH	乘积寄存器高字节								xxxx xxxx	57, 111
PRODL	乘积寄存器低字节								xxxx xxxx	57, 111
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	57, 115
INTCON2	RBPV	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	1111 1111	57, 116
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	1100 0000	57, 117
INDF0	使用 FSR0 的内容来寻址数据存储器——FSR0 的值不变 (不是物理寄存器)								N/A	57, 83
POSTINC0	使用 FSR0 的内容来寻址数据存储器——FSR0 的值后增 (不是物理寄存器)								N/A	57, 84
POSTDEC0	使用 FSR0 的内容来寻址数据存储器——FSR0 的值后减 (不是物理寄存器)								N/A	57, 84
PREINC0	使用 FSR0 的内容来寻址数据存储器——FSR0 的值预增 (不是物理寄存器)								N/A	57, 84
PLUSW0	使用 FSR0 的内容来寻址数据存储器——FSR0 的值预增 (不是物理寄存器)，FSR0 的偏移量由 W 寄存器提供								N/A	57, 84
FSR0H	—	—	—	—	间接数据存储器地址指针 0 的高字节				---- xxxx	57, 83
FSR0L	间接数据存储器地址指针 0 的低字节								xxxx xxxx	57, 83
WREG	工作寄存器								xxxx xxxx	57
INDF1	使用 FSR1 的内容来寻址数据存储器——FSR1 的值不变 (不是物理寄存器)								N/A	57, 83
POSTINC1	使用 FSR1 的内容来寻址数据存储器——FSR1 的值后增 (不是物理寄存器)								N/A	57, 84
POSTDEC1	使用 FSR1 的内容来寻址数据存储器——FSR1 的值后减 (不是物理寄存器)								N/A	57, 84
PREINC1	使用 FSR1 的内容来寻址数据存储器——FSR1 的值预增 (不是物理寄存器)								N/A	57, 84
PLUSW1	使用 FSR1 的内容来寻址数据存储器——FSR1 的值预增 (不是物理寄存器)，FSR1 的偏移量由 W 寄存器提供								N/A	57, 84
FSR1H	—	—	—	—	间接数据存储器地址指针 1 的高字节				---- xxxx	58, 83
FSR1L	间接数据存储器地址指针 1 的低字节								xxxx xxxx	58, 83
BSR	—	—	—	—	存储区选择寄存器				---- 0000	58, 72
INDF2	使用 FSR2 的内容来寻址数据存储器——FSR2 的值不变 (不是物理寄存器)								N/A	58, 83
POSTINC2	使用 FSR2 的内容来寻址数据存储器——FSR2 的值后增 (不是物理寄存器)								N/A	58, 84
POSTDEC2	使用 FSR2 的内容来寻址数据存储器——FSR2 的值后减 (不是物理寄存器)								N/A	58, 84
PREINC2	使用 FSR2 的内容来寻址数据存储器——FSR2 的值预增 (不是物理寄存器)								N/A	58, 84
PLUSW2	使用 FSR2 的内容来寻址数据存储器——FSR2 的值预增 (不是物理寄存器)，FSR2 的偏移量由 W 寄存器提供								N/A	58, 84
FSR2H	—	—	—	—	间接数据存储器地址指针 2 的高字节				---- xxxx	58, 83
FSR2L	间接数据存储器地址指针 2 的低字节								xxxx xxxx	58, 83
STATUS	—	—	—	N	OV	Z	DC	C	---x xxxx	58, 81

图注: x = 未知, u = 不变, - = 未实现, q = 取值视情况而定, r = 保留, 不能修改

注 1: PC 的 bit 21 仅在测试模式和串行编程模式下可用。

2: 这些寄存器和 / 或位仅在 80 引脚器件中可用; 在其他器件上, 未实现且读为 0。此处给出了 80 引脚器件的复位值。

3: 当 MSSP 模块在 I<sup>2</sup>C™ 从模式下工作时, 这些位的备用名称和定义。详细信息请参见第 17.4.3.2 节“地址掩码”。

4: 只有在特定振荡器配置中才可以使用 PLEN 位; 否则, 它被禁止且读为 0。详细信息请参见第 3.4.3 节“PLL 倍频器”。

5: 只有在选择内部振荡器作为默认时钟源 (FOSC2 配置位 = 0) 时才将 RA6/RA7 配置为端口引脚并设置其相关锁存器和方向位; 否则, 它们将禁止, 且这些位将读为 0。

# PIC18F85J11 系列

表 6-4: PIC18F85J11 系列寄存器汇总 (续)

寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 与 BOR 时的值	详情请见 (页)
TMR0H	Timer0 寄存器的高字节								0000 0000	58, 155
TMR0L	Timer0 寄存器的低字节								xxxx xxxx	58, 155
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	58, 153
OSCCON	IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0	0100 q000	36, 58
WDTCON	REGSLP	—	—	—	—	—	—	SWDTEN	0--- --0	58, 287
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR	0-11 11q0	52, 58
TMR1H	Timer1 寄存器的高字节								xxxx xxxx	58, 161
TMR1L	Timer1 寄存器的低字节								xxxx xxxx	58, 161
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN	TMR1CS	TMR1ON	0000 0000	58, 157
TMR2	Timer2 寄存器								0000 0000	58, 164
PR2	Timer2 周期寄存器								1111 1111	58, 164
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	58, 163
SSPBUF	MSSP 接收缓冲器 / 发送寄存器								xxxx xxxx	58, 187, 222
SSPADD	MSSP 地址寄存器 (I <sup>2</sup> C™ 从模式)。MSSP 波特率重载寄存器 (I <sup>2</sup> C 主模式)。								0000 0000	58, 222
SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	58, 180, 189
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	58, 181, 190
SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	58, 191, 192
	GCEN	ACKSTAT	ADMSK5 <sup>(3)</sup>	ADMSK4 <sup>(3)</sup>	ADMSK3 <sup>(3)</sup>	ADMSK2 <sup>(3)</sup>	ADMSK1 <sup>(3)</sup>	SEN		
ADRESH	A/D 结果寄存器的高字节								xxxx xxxx	59, 267
ADRESL	A/D 结果寄存器的低字节								xxxx xxxx	59, 267
ADCON0	ADCAL	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	0-00 0000	59, 259
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	--00 0000	59, 260
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0-00 0000	59, 261
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000 0000	59, 275
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	59, 269
TMR3H	Timer3 寄存器的高字节								xxxx xxxx	59, 167
TMR3L	Timer3 寄存器的低字节								xxxx xxxx	59, 167
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYN	TMR3CS	TMR3ON	0000 0000	59, 165
PSPCON	IBF	OBF	IBOV	PSPMODE	—	—	—	—	0000 ----	59, 165
SPBRG1	EUSART 波特率发生器寄存器								0000 0000	59, 228
RCREG1	EUSART 接收寄存器								0000 0000	59, 236
TXREG1	EUSART 发送寄存器								0000 0000	59, 234
TXSTA1	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	0000 0010	59, 224
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	59, 225
EECON2	EEPROM 控制寄存器 2 (不是物理寄存器)								---- ----	59, 90
EECON1	—	—	—	FREE	WRERR	WREN	WR	—	---0 x00-	59, 91

图注: x = 未知, u = 不变, - = 未实现, q = 取值视情况而定, r = 保留, 不能修改

注 1: PC 的 bit 21 仅在测试模式和串行编程模式下可用。

2: 这些寄存器和 / 或位仅在 80 引脚器件中可用; 在其他器件上, 未实现且读为 0。此处给出了 80 引脚器件的复位值。

3: 当 MSSP 模块在 I<sup>2</sup>C™ 从模式下工作时, 这些位的备用名称和定义。详细信息请参见第 17.4.3.2 节 “地址掩码”。

4: 只有在特定振荡器配置中才可以使用 PLEN 位; 否则, 它被禁止且读为 0。详细信息请参见第 3.4.3 节 “PLL 倍频器”。

5: 只有在选择内部振荡器作为默认时钟源 (FOSC2 配置位 = 0) 时才将 RA6/RA7 配置为端口引脚并设置其相关锁存器和方向位; 否则, 它们将禁止, 且这些位将读为 0。

表 6-4: PIC18F85J11 系列寄存器汇总 (续)

寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 与 BOR 时的值	详情请见 (页)
IPR3	—	—	RC2IP	TX2IP	—	CCP2IP	CCP1IP	—	--00 -11-	59, 126
PIR3	—	—	RC2IF	TX2IF	—	CCP2IF	CCP1IF	—	--00 -00-	59, 120
PIE3	—	—	RC2IE	TX2IE	—	CCP2IE	CCP1IE	—	--00 -00-	59, 123
IPR2	OSCFIP	CMIP	—	—	BCLIP	LVDIP	TMR3IP	—	11-- 111-	59, 125
PIR2	OSCFIF	CMIF	—	—	BCLIF	LVDIF	TMR3IF	—	00-- 000-	59, 119
PIE2	OSCFIE	CMIE	—	—	BCLIE	LVDIE	TMR3IE	—	00-- 000-	59, 122
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	—	TMR2IP	TMR1IP	1111 1-11	59, 124
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	—	TMR2IF	TMR1IF	0000 0-00	59, 118
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	—	TMR2IE	TMR1IE	0000 0-00	59, 121
MEMCON <sup>(2)</sup>	EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0	0-00 --00	59, 100
OSCTUNE	INTSRC	PLLEN <sup>(4)</sup>	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	0000 0000	37, 59
TRISJ <sup>(2)</sup>	TRISJ7	TRISJ6	TRISJ5	TRISJ4	TRISJ3	TRISJ2	TRISJ1	TRISJ0	1111 1111	60, 149
TRISH <sup>(2)</sup>	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	1111 1111	60, 147
TRISG	SPIOD	CCP2OD	CCP1OD	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	0001 1111	60, 146
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	1111 111-	60, 144
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	—	TRISE1	TRISE0	1111 1-11	60, 142
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	60, 139
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	60, 136
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	60, 133
TRISA	TRISA7 <sup>(5)</sup>	TRISA6 <sup>(5)</sup>	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	60, 131
LATJ <sup>(2)</sup>	LATJ7	LATJ6	LATJ5	LATJ4	LATJ3	LATJ2	LATJ1	LATJ0	xxxx xxxx	60, 149
LATH <sup>(2)</sup>	LATH7	LATH6	LATH5	LATH4	LATH3	LATH2	LATH1	LATH0	xxxx xxxx	60, 147
LATG	U2OD	U1OD	—	LATG4	LATG3	LATG2	LATG1	LATG0	00-x xxxxx	60, 146
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	—	xxxx xxx-	60, 144
LATE	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	xxxx xxxxx	60, 142
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx xxxxx	60, 139
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx xxxxx	60, 136
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx xxxxx	60, 133
LATA	LATA7 <sup>(5)</sup>	LATA6 <sup>(5)</sup>	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx xxxxx	60, 131
PORTJ <sup>(2)</sup>	RJ7	RJ6	RJ5	RJ4	RJ3	RJ2	RJ1	RJ0	xxxx xxxxx	60, 149
PORTH <sup>(2)</sup>	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	xxxx xxxxx	60, 147
PORTG	RDPU	REPU	RJPU <sup>(2)</sup>	RG4	RG3	RG2	RG1	RG0	000x xxxxx	60, 146
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	xxxx xxx-	60, 144
PORTE	RE7	RE6	RE5	RE4	RE3	—	RE1	RE0	xxxx x-xx	60, 142
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxxx	60, 139
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxxx	60, 136
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxxx	60, 133
PORTA	RA7 <sup>(5)</sup>	RA6 <sup>(5)</sup>	RA5	RA4	RA3	RA2	RA1	RA0	xx0x 0000	60, 131

图注: x = 未知, u = 不变, - = 未实现, q = 取值视情况而定, r = 保留, 不能修改

注 1: PC 的 bit 21 仅在测试模式和串行编程模式下可用。

2: 这些寄存器和/或位仅在 80 引脚器件中可用; 在其他器件上, 未实现且读为 0。此处给出了 80 引脚器件的复位值。

3: 当 MSSP 模块在 I<sup>2</sup>C™ 从模式下工作时, 这些位的备用名称和定义。详细信息请参见第 17.4.3.2 节“地址掩码”。

4: 只有在特定振荡器配置中才可以使用 PLLEN 位; 否则, 它被禁止且读为 0。详细信息请参见第 3.4.3 节“PLL 倍频器”。

5: 只有在选择内部振荡器作为默认时钟源 (FOSC2 配置位 = 0) 时才将 RA6/RA7 配置为端口引脚并设置其相关锁存器和方向位; 否则, 它们将禁止, 且这些位将读为 0。

# PIC18F85J11 系列

表 6-4: PIC18F85J11 系列寄存器汇总 (续)

寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 与 BOR 时的值	详情请见 (页)
SPBRGH1	EUSART 波特率发生器的高字节								0000 0000	60, 228
BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	01-0 0-00	60, 226
CCPR1H	捕捉 / 比较 / PWM 寄存器 1 的高字节								xxxx xxxx	60, 170
CCPR1L	捕捉 / 比较 / PWM 寄存器 1 的低字节								xxxx xxxx	60, 170
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	60, 169
CCPR2H	捕捉 / 比较 / PWM 寄存器 2 的高字节								xxxx xxxx	61, 170
CCPR2L	捕捉 / 比较 / PWM 寄存器 2 的低字节								xxxx xxxx	61, 170
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	61, 169
SPBRG2	AUSART 波特率发生器寄存器								0000 0000	61, 248
RCREG2	AUSART 接收寄存器								0000 0000	61, 253
TXREG2	AUSART 发送寄存器								0000 0000	61, 251
TXSTA2	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	61, 246
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	61, 247

图注: x = 未知, u = 不变, - = 未实现, q = 取值视情况而定, r = 保留, 不能修改

- 注
- 1: PC 的 bit 21 仅在测试模式和串行编程模式下可用。
  - 2: 这些寄存器和 / 或位仅在 80 引脚器件中可用; 在其他器件上, 未实现且读为 0。此处给出了 80 引脚器件的复位值。
  - 3: 当 MSSP 模块在 I<sup>2</sup>C™ 从模式下工作时, 这些位的备用名称和定义。详细信息请参见第 17.4.3.2 节 “地址掩码”。
  - 4: 只有在特定振荡器配置中才可以使用 PLEN 位; 否则, 它被禁止且读为 0。详细信息请参见第 3.4.3 节 “PLL 倍频器”。
  - 5: 只有在选择内部振荡器作为默认时钟源 (FOSC2 配置位 = 0) 时才将 RA6/RA7 配置为端口引脚并设置其相关锁存器和方向位; 否则, 它们将禁止, 且这些位将读为 0。



## 6.3.5 STATUS 寄存器

如寄存器 6-3 所示，STATUS 寄存器包含 ALU 的算术运算状态。和其他寄存器一样，STATUS 寄存器可以是任何指令的操作数。如果一条影响 Z、DC、C、OV 或 N 位的指令的目标寄存器是 STATUS 寄存器，则指令执行的结果将不会被直接写入。

根据器件逻辑，这些位会被置 1 或清零。所以当执行一条把 STATUS 寄存器作为目标寄存器的指令后，STATUS 寄存器的结果可能和预想的不一樣。例如，CLRF STATUS 会将 Z 位置 1，而保留其余位不变。此时，

读取 STATUS 寄存器的值得到的是 000u u1uu。因此，建议只用 BCF、BSF、SWAPF、MOVFF 和 MOVWF 指令来改变 STATUS 寄存器的值，因为这些指令不会影响该寄存器中的 Z、C、DC、OV 或 N 位。

关于其他不影响任何状态位的指令，请参见表 25-2 和表 25-3 中的指令集汇总。

**注：** 在减法运算中，C 和 DC 位分别作为借位和半借位标志位。

**寄存器 6-3: STATUS 寄存器**

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	N	OV	Z	DC <sup>(1)</sup>	C <sup>(2)</sup>
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 **N：** 负标志位  
此位用于有符号的算术运算（通过 2 进制补码方式进行）。它表示结果是否为负（ALU MSB = 1）。  
1 = 结果为负  
0 = 结果为正

bit 3 **OV：** 溢出位  
此位用于有符号的算术运算（通过 2 进制补码方式进行）。表明运算结果超出了 7 位二进制数据的范围，溢出将导致符号位（bit 7）发生改变。  
1 = 有符号的算术运算中发生溢出（本次运算）  
0 = 没有发生溢出

bit 2 **Z：** 全零标志位  
1 = 算术运算或逻辑运算结果为零  
0 = 算术运算或逻辑运算结果不为零

bit 1 **DC：** 半进位 / 借位 <sup>(1)</sup>  
对于 ADDWF、ADDLW、SUBLW 和 SUBWF 指令：  
1 = 结果的第 4 低位发生了进位  
0 = 结果的第 4 低位未发生进位

bit 0 **C：** 进位 / 借位 <sup>(2)</sup>  
对于 ADDWF、ADDLW、SUBLW 和 SUBWF 指令：  
1 = 结果的最高有效位发生了进位  
0 = 结果的最高有效位未发生进位

- 注** 1：对于借位，极性是相反的。减法是通过加上第二个操作数的 2 进制补码来实现的。对于移位指令（RRF 和 RLF），此位的值来自源寄存器的 bit 4 或 bit 3。
- 2：对于借位，极性是相反的。减法是通过加上第二个操作数的 2 进制补码来实现的。对于移位指令（RRF 和 RLF），此位的值来自源寄存器的高位或低位。

## 6.4 数据寻址模式

**注：** 当使能 PIC18 扩展指令集时，PIC18 内核指令集中的某些指令的执行会发生改变。更多信息，请参见第 6.6 节“数据存储器和扩展指令集”。

虽然只能用一种方法（通过程序计数器）对程序存储器进行寻址，但是可以用多种方法来对数据存储空间进行寻址。对于大多数指令，寻址模式是固定的。其余的指令最多可以使用三种模式，这取决于使用的操作数及是否使能了扩展指令集。

寻址模式有：

- 固有寻址
- 立即数寻址
- 直接寻址
- 间接寻址

当使能扩展指令集（XINST 配置位 = 1）时，可以使用另一种寻址模式，即使用立即数作为偏移量进行变址寻址的模式。第 6.6.1 节“使用立即数作为偏移量进行变址寻址”中对其操作进行了更详细的讨论。

### 6.4.1 固有和立即数寻址

很多 PIC18 控制类指令根本不需要任何参数，执行这些指令要么对整个器件造成影响，要么仅针对一个寄存器进行操作。该寻址模式称为固有寻址，例如 SLEEP、RESET 和 DAW。

其他指令的工作方式与此类似，但需要在操作码中明确指定参数。由于需要一些立即数作为参数，这种寻址模式被称为立即数寻址模式，例如 ADDLW 和 MOVLW，它们分别将立即数值加到或移入 W 寄存器。其他的立即数寻址指令，例如 CALL 和 GOTO，包含一个 20 位的程序存储器地址。

### 6.4.2 直接寻址

直接寻址在操作码中指定操作的全部或部分源地址和 / 或目标地址。这些选项由指令附带的参数指定。

在 PIC18 内核指令集中，位操作类指令和字节操作类指令默认情况下使用直接寻址。所有这些指令都包含某个 8 位的直接地址作为它们的最低有效字节。此地址指定数据 RAM 的某个存储区中的寄存器地址（第 6.3.3 节“通用寄存器”）或快速操作存储区中的地址单元（第 6.3.2 节“快速操作存储区”）作为指令的数据源。

快速操作 RAM 位 a 决定地址的解析方式。当 a 为 1 时，BSR（第 6.3.1 节“存储区选择寄存器”）的内容和地址一起用于确定寄存器完整的 12 位地址。当 a 为 0 时，此地址将被解析为快速操作存储区中的一个寄存器。使用快速操作 RAM 寻址有时候也被称为直接强制寻址模式。

有几条指令，比如 MOVFF，在操作码中包含完整的 12 位地址（源地址或目标地址）。在这些情况下，BSR 完全被忽略。

操作结果的目标寄存器由目标位 d 确定。当 d 为 1 时，结果被存储到源寄存器并覆盖它原来的内容。当 d 为 0 时，结果被存回 W 寄存器中。不带 d 参数的指令的目标地址隐含在指令中，它们是操作的目标寄存器或 W 寄存器。

### 6.4.3 间接寻址

间接寻址允许用户访问数据存储区中的单元而不需要在指令中给出一个固定的地址。通过使用文件选择寄存器（File Select Register, FSR）作为指针指向被读取或写入的单元实现的。由于 FSR 本身作为特殊功能寄存器位于 RAM 中，所以也可以在程序控制下直接对它们进行操作。这使得 FSR 对于在数据存储区中实现诸如表和数组等数据结构非常有用。

也可以使用间接指针操作数（Indirect File Operand, INDF）对寄存器进行间接寻址。此操作允许自动递增、递减或偏移指针，从而自动控制指针的值。它通过循环提高代码执行效率，如例 6-5 所示的清零整个 RAM 存储区的操作。它也允许用户在数据存储区中进行变址寻址和其他针对程序存储器的堆栈指针操作。

#### 例 6-5: 使用间接寻址清零 RAM (BANK 1) 的方法

LFSR	FSR0, 100h ;
NEXT	CLRF POSTINC0 ; Clear INDF
	; register then
	; inc pointer
	BTFSS FSR0H, 1 ; All done with
	; Bank1?
	BRA NEXT ; NO, clear next
CONTINUE	; YES, continue

## 6.4.3.1 FSR 寄存器和 INDF 操作数

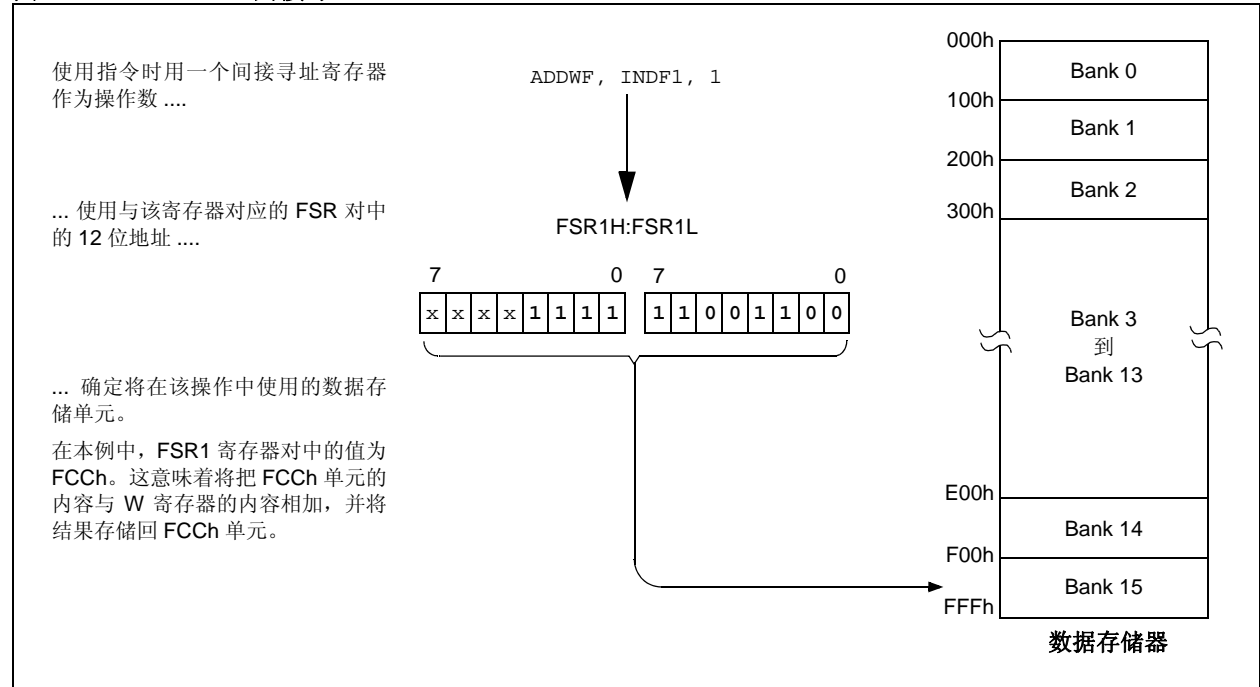
间接寻址的核心是三组寄存器 **FSR0**、**FSR1** 和 **FSR2**。每组寄存器都代表一对 8 位寄存器 **FSRnH** 和 **FSRnL**。**FSRnH** 寄存器的高四位未使用，所以每对 **FSR** 只保存一个 12 位值，从而可以线性寻址整个数据存储单元。因此，**FSR** 寄存器对将作为指向数据存储单元的指针。

间接寻址可用一组间接指针操作数（从 **INDF0** 到 **INDF2**）完成。这些操作数可以被看作是“虚拟”寄存器：它们被映射到 **SFR** 空间中而不是通过物理方式实

现。对特定的 **INDF** 寄存器执行读或写操作实际上访问的是相应的 **FSR** 寄存器对。例如，读 **INDF1** 就是读 **FSR1H:FSR1L** 指向的地址单元中的数据。使用 **INDF** 寄存器作为操作数的指令实际上是使用相应的 **FSR** 的内容作为指向目标地址的指针。**INDF** 操作数只是使用指针的一种简便方法。

由于间接寻址使用完整的 12 位地址，因此没有必要进行数据 **RAM** 分区。故 **BSR** 和快速操作 **RAM** 位的当前内容对于确定目标地址没有影响。

图 6-10: 间接寻址



## 6.4.3.2 FSR 寄存器和 POSTINC、POSTDEC、PREINC 以及 PLUSW

除了 INDF 操作数之外，每对 FSR 寄存器还有四个额外的间接操作数。和 INDF 一样，这些也是不能直接读写的“虚拟”寄存器。访问这些寄存器其实就是访问相关的 FSR 寄存器对，并对其存储的数据进行特定的操作。这些寄存器是：

- **POSTDEC**：访问 FSR 的值，然后将它自动减 1
- **POSTINC**：访问 FSR 的值，然后将它自动加 1
- **PREINC**：将 FSR 的值加 1，然后在操作中使用该值
- **PLUSW**：将 W 寄存器中有符号值（从 -127 到 128）与 FSR 寄存器中的值相加，并在操作中使用得到的新值。

如前所述，访问 INDF 寄存器使用的是 FSR 寄存器中的值，但不更改此值。同样，访问 PLUSW 寄存器是将 W 寄存器中的值作为 FSR 值的偏移量，该操作不会改变这两个寄存器中的值。访问其他虚拟寄存器会更改 FSR 寄存器的值。

用 POSTDEC、POSTINC 和 PREINC 在 FSR 上进行操作会影响整对寄存器；也就是说一旦 FSRnL 寄存器从 FFh 溢出到 00h，会向 FSRnH 寄存器进位。但这些操作的结果不会更改 STATUS 寄存器中的任何标志位（如 Z、N 和 OV 等）。

PLUSW 寄存器可以用于在数据存储空间实现变址寻址。通过控制 W 寄存器中的值，用户可以访问相对当前指针地址有固定偏移量的地址单元。在某些应用中，该功能可以被用于在数据存储器内部实现某些非常有用的程序控制结构，如软件堆栈。

## 6.4.3.3 通过 FSR 对其他 FSR 进行操作

在某些特殊情况下，间接寻址操作以其他 FSR 或虚拟寄存器作为寻址目标。例如，使用 FSR 指向一个虚拟寄存器会导致操作不成功。假如如下特殊情况：FSR0H:FSR0L 保存的是 INDF1 的地址 FE7h。尝试使用 INDF0 作为操作数读取 INDF1 的值，将返回 00h。尝试使用 INDF0 作为操作数写入 INDF1，将会导致执行一条 NOP。

另一方面，使用虚拟寄存器对 FSR 寄存器对进行写操作可能会产生与预期不同的结果。在这些情形下，值将被写入 FSR 寄存器对，但 FSR 中的值不会有任何增减。因此，写入 INDF2 或 POSTDEC2 时会把相同的值写入 FSR2H:FSR2L。

由于 FSR 是在 SFR 空间中映射的物理寄存器，所以可以通过直接寻址对它们进行操作。用户在使用这些寄存器时应特别小心，尤其是当代码使用间接寻址时。

同样，通常允许通过间接寻址对所有其他的 SFR 进行操作。用户在进行此类操作时应特别小心，以免更改设置从而影响器件操作。

## 6.5 程序存储器和扩展指令集

使用扩展指令集不会影响程序存储器的操作。

使能扩展指令集会向现有的 PIC18 指令集添加五个额外的双字指令：ADDFSR、CALLW、MOVSF、MOVSS 和 SUBFSR。第 6.2.4 节“双字指令”说明了这些指令的执行方式。

## 6.6 数据存储器和扩展指令集

使能 PIC18 扩展指令集（XINST 配置位 = 1）明显更改了数据存储及其寻址的方式。特别是很多 PIC18 内核指令使用快速操作存储区的方式会有所不同。这是由于扩展指令集引入了对数据存储空间的新寻址模式。此模式还会改变使用 FSR2 及其相关的操作数进行间接寻址的方式。

了解哪些部分保持不变同样重要。数据存储空间的大小及其线性寻址方式都不会改变。SFR 映射也保持不变。PIC18 内核指令仍然以直接和间接寻址模式进行操作；固有和立即数指令操作照旧。FSR0 和 FSR1 的间接寻址方式也保持不变。

### 6.6.1 使用立即数作为偏移量进行变址寻址

使能 PIC18 扩展指令集将更改使用 FSR2 寄存器对其相关的文件操作数进行间接寻址的方式。在适当的条件下，使用快速操作存储区的指令（即大多数位和字节操作类指令）可以利用指令中指定的偏移量来执行变址寻址。这种特定的寻址模式被称为使用立即数偏移量的变址寻址或立即数变址寻址模式。

在使用扩展指令集时，这种寻址模式有如下要求：

- 强制使用快速操作存储区（a = 0）；且
- 指针地址参数要小于或等于 5Fh。

在这些条件下，指令的指针地址不被解析为地址的低字节（在直接寻址中和 BSR 一起使用），或快速操作存储区中的 8 位地址。相反，该值被解析为由 FSR2 指定的地址指针的偏移量。将偏移量与 FSR2 的内容相加以获取操作的目标地址。

### 6.6.2 受立即数变址寻址模式影响的指令

任何可以使用直接寻址的 PIC18 内核指令均会受到立即数变址寻址模式的潜在影响。包括所有字节和位操作类指令，即标准 PIC18 指令集中几乎一半的指令。只有使用固有或立即数寻址模式的指令不受影响。

此外，如果字节和位操作类指令不使用快速操作存储区（快速操作 RAM 位为 1）或包含 60h 或以上的指针地址，它们也不会受影响。符合这些标准的指令会像以前一样继续执行。图 6-11 给出了当使能扩展指令集时，各种可能的寻址模式的对比。

那些想要在立即数变址寻址模式中使用字节或位操作类指令的用户，应该注意此模式下汇编语法的改变。第 25.2.1 节“扩展指令的语法”中将对此进行更详细的说明。

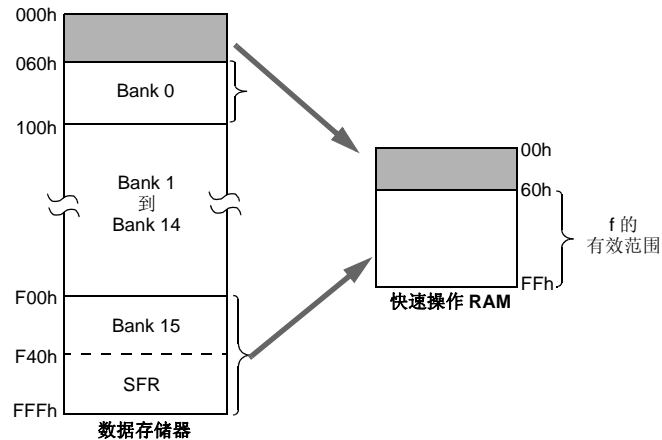
图 6-11: 位和字节操作类指令的寻址方式对比（使能了扩展指令集）

指令示例 `ADDWF, f, d, a` (操作码 `0010 01da ffff ffff`)

当  $a = 0$  且  $f \geq 60h$  时:

此指令以直接强制模式执行。 $f$  被解析为快速操作 RAM 中 `060h` 到 `FFFh` 之间的单元。这实际上是数据存储器中从 `F60h` 到 `FFFh` (Bank 15) 之间的存储单元。

此寻址模式中不可以使用地址低于 `060h` 的单元。

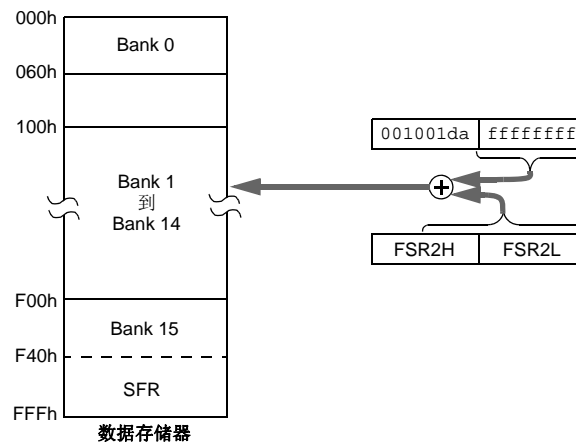


当  $a = 0$  且  $f \leq 5Fh$  时:

该指令以立即数变址寻址模式执行。 $f$  被解析为 `FSR2` 中地址值的偏移量。这两个值相加以获得指令的目标寄存器的地址。此地址可以位于数据存储空间的任何位置。

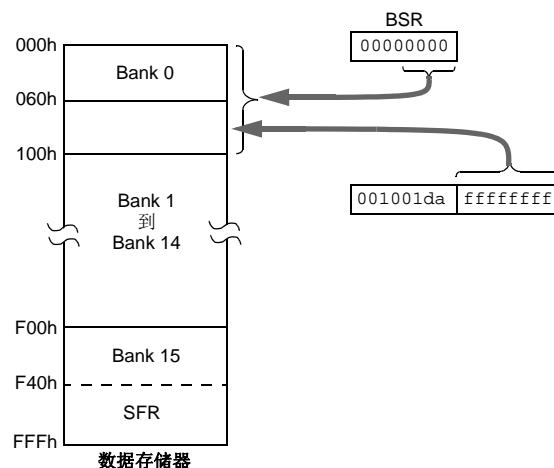
注意在此模式中，正确的语法是:

`ADDWF [k], d`  
其中  $k$  就是  $f$ 。



当  $a = 1$  ( $f$  为所有值) 时:

指令以直接模式执行（也被称为直接长地址寻址模式）。 $f$  被解析为数据存储空间的 16 个存储区中的某个单元的地址。此存储区由存储区选择寄存器 (BSR) 指定。此地址可以位于数据存储空间的任何已实现区域。



6.6.3 在立即数变址寻址模式下映射快速操作存储区

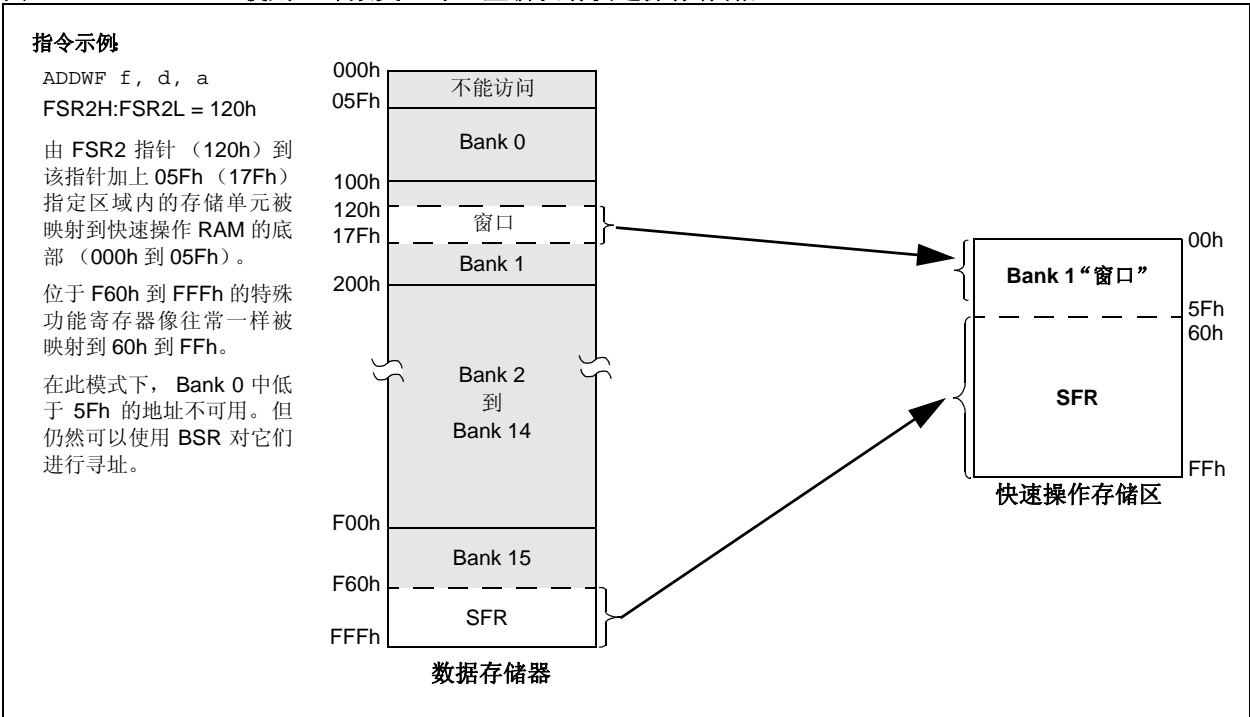
使用立即数变址寻址模式能改变快速操作 RAM 的低地址部分（00h 到 5Fh）的映射方式。除了包含 Bank 0 底部的内容之外，此模式还映射 Bank 0 的内容和由用户定义的可以位于数据存储空间中任何位置的“窗口”内容。FSR2 的值确定了映射到窗口的地址的下边界，而上边界则由 FSR2 加 95（5Fh）确定。地址为 5Fh 以上的快速操作 RAM 的映射方式如前所述（见第 6.3.2 节“快速操作存储区”）。图 6-12 所示为在此寻址模式中重新映射的快速操作存储区的示例。

快速操作存储区的重新映射仅适用于立即数变址寻址模式。使用 BSR（快速操作 RAM 位为 1）的操作和前面一样继续使用直接寻址模式。任何使用间接指针操作数（包括 FSR2）进行的间接或变址寻址操作都将继续以标准间接寻址模式进行操作。任何使用快速操作存储区、但是包括大于 05Fh 的寄存器地址的指令都将使用直接寻址和常规快速操作存储区映射。

6.6.4 立即数变址寻址模式中的 BSR

虽然当使能了扩展指令集时会重新映射快速操作存储区，但 BSR 的操作将保持不变。使用 BSR 来选择数据存储区的直接寻址模式的操作方式与前面描述的相同。

图 6-12: 使用立即数变址寻址重新映射快速操作存储区



# PIC18F85J11 系列

---

注:



## 7.0 闪存程序存储器

在整个 VDD 范围内，闪存程序存储器在正常工作状态下都是可读写并可擦除的。

对程序存储器的读操作按照每次 1 个字节来执行。对程序存储器的写操作按照每次 64 字节块来执行。对程序存储器的擦除操作按照每次 1024 字节块执行。用户代码不能执行批量擦除操作。

写或擦除程序存储器将中止取指操作，直到写或擦除操作完成为止。在写或擦除期间不能访问程序存储器，因此无法执行代码。内部编程定时器可终止对程序存储器的写入和擦除操作。

写入程序存储器的值不必是有效指令。执行存无效指令的程序存储单元会导致执行 NOP 指令。

## 7.1 表读和表写

为了读写程序存储器，有两种操作可以让处理器在程序存储空间和数据 RAM 之间传送字节：

- 表读 (TBLRD)
- 表写 (TBLWT)

程序存储空间为 16 位宽，而数据 RAM 空间为 8 位宽。表读和表写操作通过一个 8 位寄存器 (TABLAT) 在这两个存储空间之间传送数据。

表读操作从程序存储器获取数据并将其存入数据 RAM 空间。图 7-1 显示了程序存储器和数据 RAM 之间的表读操作。

表写操作将数据存储空间中的数据存入程序存储器中的保持寄存器。第 7.5 节“写闪存程序存储器”将详细介绍将保持寄存器中的内容写入程序存储器的过程。图 7-2 显示了程序存储器和数据 RAM 之间的表写操作。

表操作是以字节为单位进行的。一个包含数据而非程序指令的表块不必按字对齐。因此，表块可在任何字节地址处开始和结束。如果使用表写操作将可执行代码写入程序存储器，程序指令就需要按字对齐。

图 7-1: 表读操作

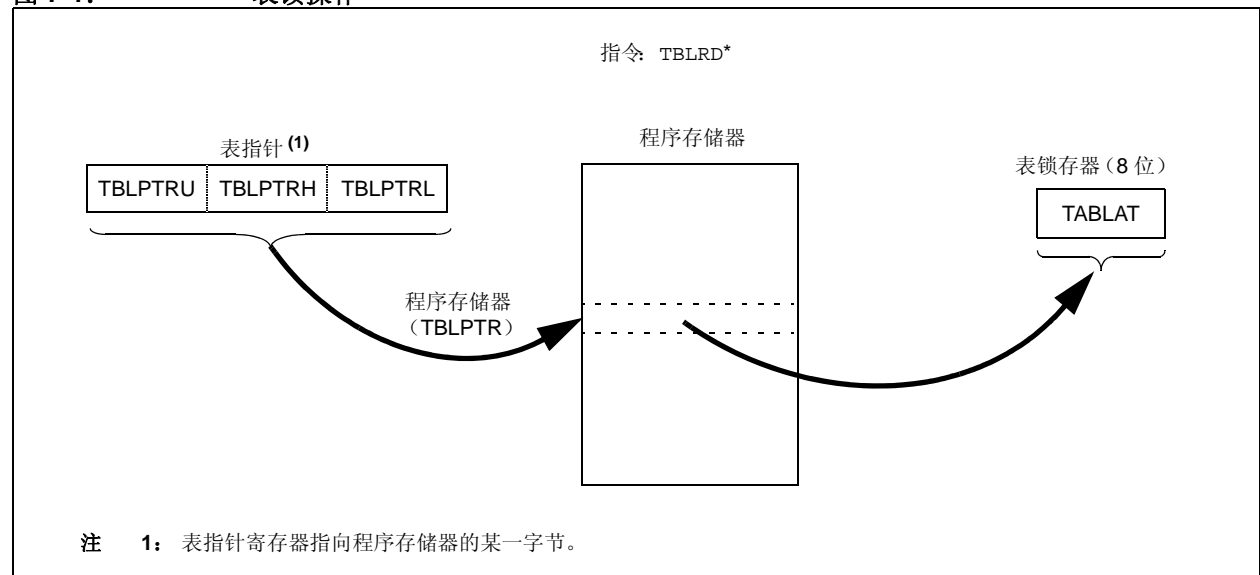
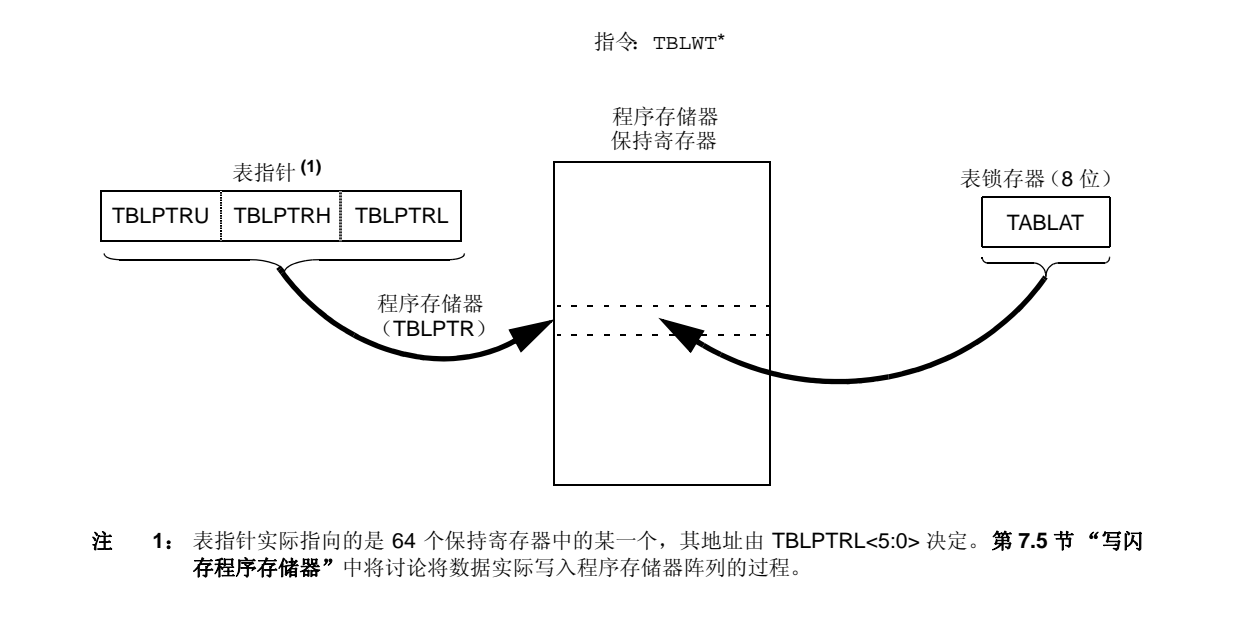


图 7-2: 表写操作



## 7.2 控制寄存器

有几个控制寄存器与 TBLRD 和 TBLWT 指令一起使用。其中包括:

- EECON1 寄存器
- EECON2 寄存器
- TABLAT 寄存器
- TBLPTR 寄存器

### 7.2.1 EECON1 和 EECON2 寄存器

EECON1 寄存器 (寄存器 7-1) 是访问存储器的控制寄存器。EECON2 寄存器不是物理寄存器, 它专用于存储器的擦写操作。读 EECON2 得到的是全 0。

FREE 位置 1 时允许对程序存储器执行擦除操作。FREE 位置 1 时, 擦除操作由下一个 WR 命令触发。FREE 位清零时, 则只使能写操作。

WREN 位置 1 时允许写操作。上电时将清零 WREN 位。当 WR 位置 1 时, WRERR 位将被硬件置 1; 当内部编程定时器超时并且写操作完成时, 清零 WRERR 位。

**注:** 如果在正常工作期间, WRERR 的读取值为 1, 则表明写操作因复位而提早终止或进行了非法的写操作。

控制位 WR 用于启动写操作。该位不能由软件清零, 但可由软件置 1。该位在写操作完成时由硬件清零。

## 寄存器 7-1: EECON1: EEPROM 控制寄存器 1

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

图注: U = 未实现位, 读为 0

R = 可读位

W = 可写位

S = 只能置 1 的位 (不能由软件清零)

-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 **FREE:** 闪存擦除使能位

1 = 在下一个 WR 命令时擦除由 TBLPTR 寻址的程序存储器块 (擦除操作完成时清零)

0 = 仅执行写操作

bit 3 **WRERR:** 闪存程序错误标志位

1 = 写操作提早终止 (由于正常工作中自定时编程期间的任何复位, 或非法写入)

0 = 写操作完成

bit 2 **WREN:** 闪存程序写使能位

1 = 允许写入闪存程序存储器

0 = 禁止写入闪存程序存储器

bit 1 **WR:** 写控制位

1 = 启动程序存储器的擦除或写周期

(该操作是自定时的, 一旦写入完成即由硬件将该位清零。软件只能将 WR 位置 1 而不能清零。)

0 = 写周期完成

bit 0 未实现: 读为 0

# PIC18F85J11 系列

## 7.2.2 表锁存寄存器 (TABLAT)

表锁存器 (TABLAT) 是映射到 SFR 空间的一个 8 位寄存器。表锁存寄存器用于保存在程序存储器和数据 RAM 之间传输的 8 位数据。

## 7.2.3 表指针寄存器 (TBLPTR)

表指针 (TBLPTR) 寄存器指向程序存储器中的一个字节。TBLPTR 由 3 个 SFR 寄存器组成，即：表指针最高字节、表指针高字节和表指针低字节 (TBLPTRU:TBLPTRH:TBLPTRL)。这 3 个寄存器合起来组成一个 22 位宽的指针。器件使用该寄存器的低 21 位寻址最多 2 MB 的程序存储空间。第 22 位可用于访问器件 ID、用户 ID 以及配置位。

TBLRD 和 TBLWT 指令要用到表指针寄存器 TBLPTR。根据不同的表操作，上述指令可使用四种方式更新 TBLPTR。表 7-1 列出了这些操作。这些操作只会影响 TBLPTR 的低 21 位。

## 7.2.4 表指针边界

TBLPTR 用于读、写和擦除闪存程序存储器。

当执行 TBLRD 时，由 TBLPTR 的全部 22 位决定将程序存储器的哪个字节读入 TABLAT。

当执行 TBLWT 时，由表指针寄存器的 7 个最低有效位 (TBLPTR<6:0>) 决定要写入 64 个程序存储器保持寄存器的哪一个。当对程序存储器的定时写操作开始后 (通过 WR 位启动)，由 TBLPTR 的 12 个最高有效位 (TBLPTR<21:10>) 决定要写入哪一个 1024 字节的程序存储块。如需更多详情，请参见第 7.5 节“写闪存程序存储器”。

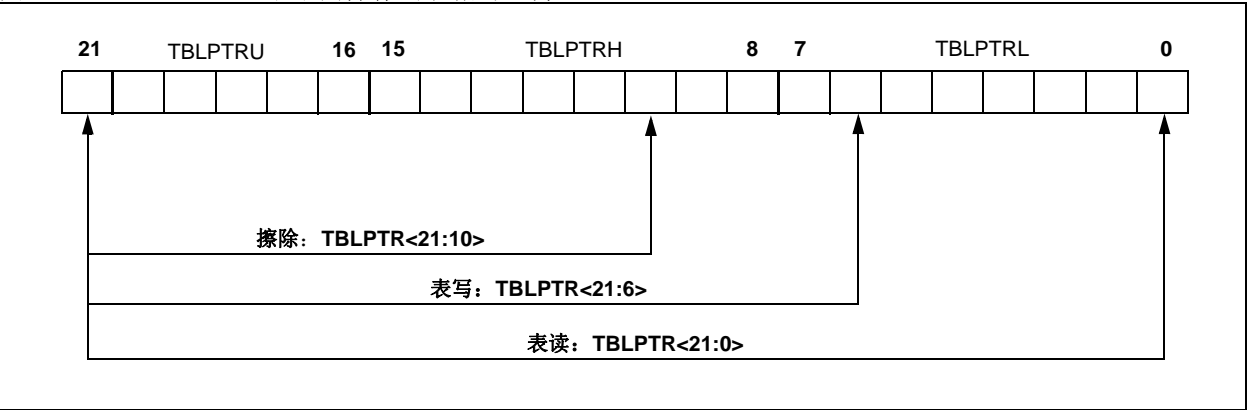
当擦除程序存储器时，表指针寄存器的 12 个最高有效位指向将要擦除的 1024 字节的块，而忽略最低有效位。

图 7-3 给出了基于闪存程序存储器操作的 TBLPTR 相关边界。

表 7-1: 使用 TBLRD 和 TBLWT 指令对表指针进行操作

示例	表指针操作
TBLRD* TBLWT*	不修改 TBLPTR
TBLRD*+ TBLWT*+	读 / 写后，TBLPTR 递增
TBLRD*- TBLWT*-	读 / 写后，TBLPTR 递减
TBLRD*+* TBLWT*+*	读 / 写前，TBLPTR 递增

图 7-3: 基于不同操作的表指针边界



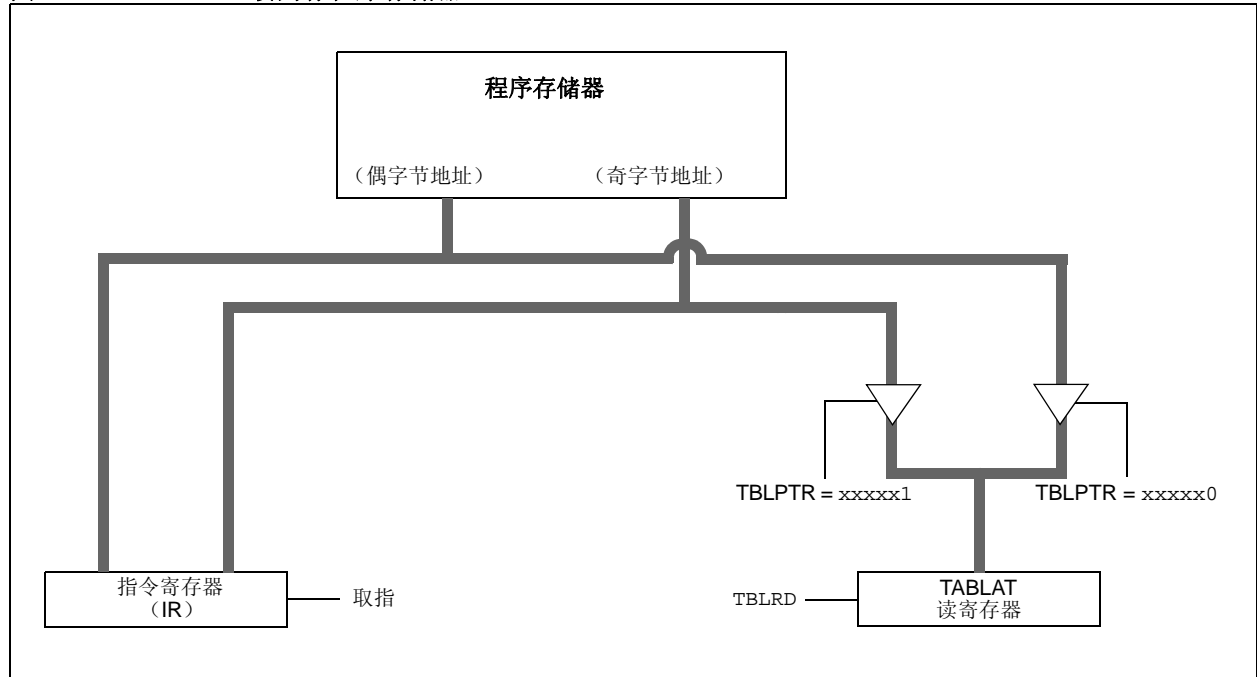
## 7.3 读闪存程序存储器

TBLRD 指令用于从程序存储器获取数据并把数据存入数据 RAM。对程序存储器执行表读操作时，每次读取一个字节。

TBLPTR 指向程序空间内的某个字节地址。执行 TBLRD 会将指向的字节存入 TABLAT。同时自动修改 TBLPTR 以进行下次表读操作。

内部程序存储器通常是以字为单位构成的。由地址的最低有效位选择字的高字节和低字节。图 7-4 显示了内部程序存储器和 TABLAT 之间的接口。

图 7-4: 读闪存程序存储器



例 7-1: 读一个闪存程序存储器字

```

MOV LW    CODE_ADDR_UPPER           ; Load TBLPTR with the base
MOV WF    TBLPTRU                    ; address of the word
MOV LW    CODE_ADDR_HIGH
MOV WF    TBLPTRH
MOV LW    CODE_ADDR_LOW
MOV WF    TBLPTRL

READ_WORD
TBLRD*+           ; read into TABLAT and increment
MOV F    TABLAT, W ; get data
MOV WF    WORD_EVEN
TBLRD*+           ; read into TABLAT and increment
MOV F    TABLAT, W ; get data
MOV WF    WORD_ODD
    
```

# PIC18F85J11 系列

## 7.4 擦除闪存程序存储器

擦除的最小数据块为 512 个字（即 1024 字节）。只有通过使用外部编程器或 ICSP 控制，才能够批量擦除更大的程序存储器块。不支持闪存阵列的字擦除。

当单片机启动擦除操作时，将擦除程序存储器的一个 1024 字节块。TBLPTR 的 12 个最高有效位 (TBLPTR<21:10>) 指向将被擦除的数据块。TBLPTR<9:0> 被忽略。

擦除操作由 EECON1 寄存器控制。必须将 WREN 位置 1 以使能写操作。必须将 FREE 位置 1 以选择擦除操作。为了安全起见，必须采用 EECON2 的写操作启动序列。

擦除内部闪存需要使用长写周期。在长写周期中，指令执行暂停。由内部编程定时器终止长周期写操作。

### 7.4.1 闪存程序存储器擦除序列

擦除内部程序存储器数据块的步骤如下：

1. 将要擦除的块地址装入表指针寄存器。
2. 将 WREN 和 FREE 位 (EECON1<2,4>) 置 1 以使能擦除操作。
3. 禁止中断。
4. 向 EECON2 写入 55h。
5. 向 EECON2 写入 0AAh。
6. 将 WR 位置 1，启动擦除周期。
7. 在擦除操作期间，CPU 将在 TIE（见参数 D133B）时间段内停止工作。
8. 重新允许中断。

例 7-2: 擦除闪存程序存储器块

	MOVLW	CODE_ADDR_UPPER	; load TBLPTR with the base
	MOVWF	TBLPTRU	; address of the memory block
	MOVLW	CODE_ADDR_HIGH	
	MOVWF	TBLPTRH	
	MOVLW	CODE_ADDR_LOW	
	MOVWF	TBLPTRL	
ERASE_BLOCK			
	BSF	EECON1, WREN	; enable write to memory
	BSF	EECON1, FREE	; enable Erase operation
	BCF	INTCON, GIE	; disable interrupts
必需的 序列	MOVLW	55h	
	MOVWF	EECON2	; write 55h
	MOVLW	0AAh	
	MOVWF	EECON2	; write 0AAh
	BSF	EECON1, WR	; start erase (CPU stall)
	BSF	INTCON, GIE	; re-enable interrupts

## 7.5 写闪存程序存储器

最小编程块大小为 32 个字（即 64 字节）。不支持字或字节编程。

表写操作用于在内部将数据装入编程闪存存储器所需的保持寄存器。表写操作使用 64 个保持寄存器进行编程。

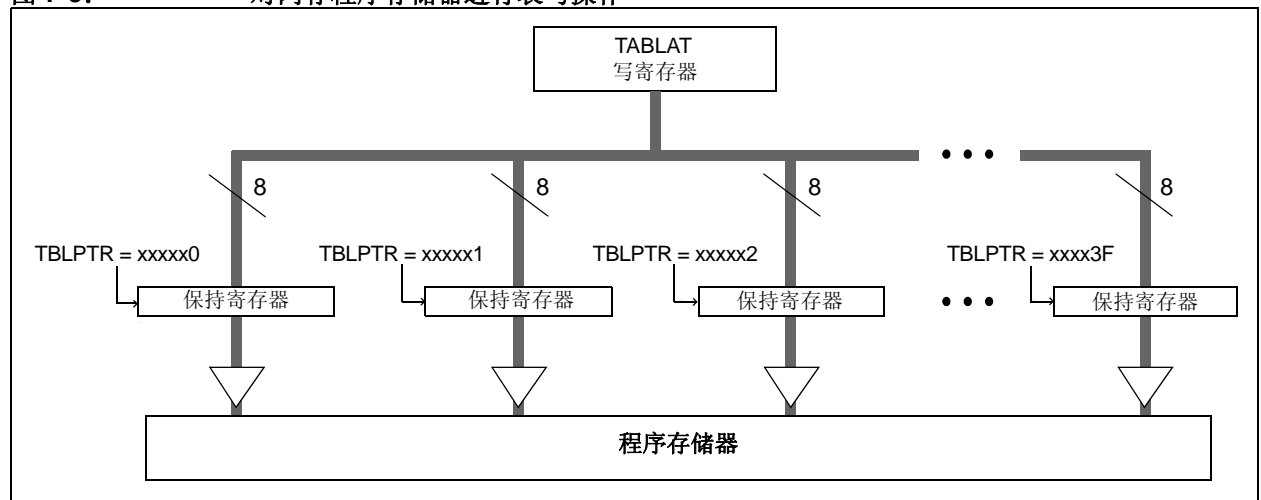
因为表锁存器（TABLAT）仅一个单字节大小，所以每次编程操作必须执行 64 次 TBLWT 指令。由于只写入保持寄存器，所有表写操作实际上都是短周期写操作。在更新完 64 个保持寄存器后，必须写入 EECON1 寄存器，以便启动长写周期编程操作。

对内部闪存进行编程需要使用长写周期。在长写周期中，指令暂停执行。由内部编程定时器终止长周期写操作。

由片上定时器控制写入的时间。写 / 擦除电压由可在器件电压范围内工作的片上电荷泵产生。

- 注 1:** 与以前的 PIC® 单片机不同，PIC18F85J11 系列单片机在写周期开始后并不复位保持寄存器。必须在编程序列开始之前清零或重写保持寄存器。
- 2:** 为了保证程序存储单元的耐擦写次数，在两次擦除操作之间，只允许对每个闪存字节进行 1 次编程。在对某一目标单元的内容做第二次修改之前，必须对其所在的目标块执行块擦除或对整个存储器执行批量擦除。

图 7-5: 对闪存程序存储器进行表写操作



### 7.5.1 闪存程序存储器写操作序列

对内部程序存储单元编程的步骤应为：

1. 将 1024 个字节读入 RAM。
2. 必要时更新 RAM 中的数据值。
3. 把要擦除的目标地址装入表指针寄存器。
4. 执行块擦除过程。
5. 把要写入的第一个字节的地址装入表指针寄存器，表指针减 1。
6. 采用自动递增的方式把 64 个字节写入保持寄存器。
7. 将 WREN 位（EECON1<2>）置 1，使能字节写操作。

8. 禁止中断。
9. 向 EECON2 写入 55h。
10. 向 EECON2 写入 0AAh。
11. 将 WR 位置 1，启动写周期。
12. 在写操作期间，CPU 将在 T<sub>tw</sub>（见参数 D133A）时间段内停止工作。
13. 重新允许中断。
14. 重复第 6 到 13 步，直到全部 1024 个字节都被写入程序存储器。
15. 对存储器进行校验（表读）。

例 7-3 给出了所需的代码示例。

**注：** 在 WR 位置 1 前，表指针地址应处于保持寄存器中所需的 64 字节地址范围内。

# PIC18F85J11 系列

例 7-3: 写闪存程序存储器

ERASE_BLOCK	MOVLW	CODE_ADDR_UPPER	; Load TBLPTR with the base address
	MOVWF	TBLPTRU	; of the memory block, minus 1
	MOVLW	CODE_ADDR_HIGH	
	MOVWF	TBLPTRH	
	MOVLW	CODE_ADDR_LOW	
	MOVWF	TBLPTRL	
	BSF	EECON1, WREN	; enable write to memory
	BSF	EECON1, FREE	; enable Erase operation
	BCF	INTCON, GIE	; disable interrupts
	MOVLW	55h	
	MOVWF	EECON2	; write 55h
	MOVLW	0AAh	
	MOVWF	EECON2	; write 0AAh
	BSF	EECON1, WR	; start erase (CPU stall)
	BSF	INTCON, GIE	; re-enable interrupts
RESTART_BUFFER	MOVLW	D'16'	
	MOVWF	WRITE_COUNTER	; Need to write 16 blocks of 64 to write
			; one erase block of 1024
	MOVLW	D'64'	
	MOVWF	COUNTER	
	MOVWF	BUFFER_ADDR_HIGH	; point to buffer
FILL_BUFFER	MOVWF	FSR0H	
	MOVLW	BUFFER_ADDR_LOW	
WRITE_BUFFER	MOVWF	FSR0L	
	...		; read the new data from I2C, SPI,
			; PSP, USART, etc.
WRITE_BYTE_TO_HREGS	MOVLW	D'64	; number of bytes in holding register
	MOVWF	COUNTER	
PROGRAM_MEMORY	MOVFF	POSTINC0, WREG	; get low byte of buffer data
	MOVWF	TABLAT	; present data to table latch
	TBLWT*		; write data, perform a short write
			; to internal TBLWT holding register.
	DECFSZ	COUNTER	; loop until buffers are full
	BRA	WRITE_BYTE_TO_HREGS	
	BSF	EECON1, WREN	; enable write to memory
	BCF	INTCON, GIE	; disable interrupts
	MOVLW	55h	
	MOVWF	EECON2	; write 55h
必需的 序列	MOVLW	0AAh	
	MOVWF	EECON2	; write 0AAh
	BSF	EECON1, WR	; start program (CPU stall)
	BSF	INTCON, GIE	; re-enable interrupts
	BCF	EECON1, WREN	; disable write to memory
	DECFSZ	WRITE_COUNTER	; done with one write cycle
	BRA	RESTART_BUFFER	; if not done replacing the erase block



## 7.5.2 写校验

根据具体的应用，好的编程习惯一般要求使用原始值对写入存储器的值进行校验。在写入次数过多已接近规范极限值的应用中，就应该采用写校验。

## 7.5.3 写操作意外终止

如果写操作因意外事件（如掉电或意外复位）终止，那么应对刚刚编程的存储单元进行校验，如果需要，还需对其进行重新编程。如果写操作在正常操作过程中由 MCLR 复位或 WDT 超时复位而中断，用户就可以检测 WRERR 位的状态并重新写入（如果需要）。

## 7.6 代码保护期间的闪存程序存储器操作

如需了解有关闪存程序存储器代码保护的详情，请参见第 23.6 节“程序校验和代码保护”。

**表 7-2: 与闪存程序存储器相关的寄存器**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
TBLPTRU	—	—	bit 21	程序存储器表指针最高字节（TBLPTR<20:16>）					57
TBPLTRH	程序存储器表指针高字节（TBLPTR<15:8>）								57
TBLPTRL	程序存储器表指针低字节（TBLPTR<7:0>）								57
TABLAT	程序存储器表锁存器								57
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
EECON2	EEPROM 控制寄存器 2（非物理寄存器）								59
EECON1	—	—	—	FREE	WRERR	WREN	WR	—	59

图注： — = 未实现，读为 0。访问程序存储器时不使用阴影单元。

# PIC18F85J11 系列

---

注:

## 8.0 外部存储器总线

**注：** 64 引脚器件未实现外部存储器总线。

外部存储器总线使器件能像访问程序存储器或数据存储器一样访问外部存储器件（如闪存、EPROM 和 SRAM 等）。外部存储器总线支持 8 位和 16 位数据宽度模式，以及 3 种最宽为 20 位的地址宽度模式。

该总线与 4 个 I/O 端口复用，占 28 个引脚。三个端口（PORTD、PORTE 和 PORTH）与地址 / 数据总线复用，总共可使用 20 根线，而 PORTJ 与总线控制信号复用。

表 8-1 列出了与外部存储器总线有关的各个引脚及其功能。

**表 8-1: PIC18F85J11 系列器件外部总线——I/O 端口功能**

名称	端口	位	外部存储器总线功能
RD0/AD0	PORTD	0	地址 bit 0 或数据 bit 0
RD1/AD1	PORTD	1	地址 bit 1 或数据 bit 1
RD2/AD2	PORTD	2	地址 bit 2 或数据 bit 2
RD3/AD3	PORTD	3	地址 bit 3 或数据 bit 3
RD4/AD4	PORTD	4	地址 bit 4 或数据 bit 4
RD5/AD5	PORTD	5	地址 bit 5 或数据 bit 5
RD6/AD6	PORTD	6	地址 bit 6 或数据 bit 6
RD7/AD7	PORTD	7	地址 bit 7 或数据 bit 7
RE0/AD8	PORTE	0	地址 bit 8 或数据 bit 8
RE1/AD9	PORTE	1	地址 bit 9 或数据 bit 9
RE2/AD10	PORTE	2	地址 bit 10 或数据 bit 10
RE3/AD11	PORTE	3	地址 bit 11 或数据 bit 11
RE4/AD12	PORTE	4	地址 bit 12 或数据 bit 12
RE5/AD13	PORTE	5	地址 bit 13 或数据 bit 13
RE6/AD14	PORTE	6	地址 bit 14 或数据 bit 14
RE7/AD15	PORTE	7	地址 bit 15 或数据 bit 15
RH0/A16	PORTH	0	地址 bit 16
RH1/A17	PORTH	1	地址 bit 17
RH2/A18	PORTH	2	地址 bit 18
RH3/A19	PORTH	3	地址 bit 19
RJ0/ALE	PORTJ	0	地址锁存使能（ALE）控制引脚
RJ1/OE	PORTJ	1	输出使能（OE）控制引脚
RJ2/WRL	PORTJ	2	低字节写（WRL）控制引脚
RJ3/WRH	PORTJ	3	高字节写（WRH）控制引脚
RJ4/BA0	PORTJ	4	字节地址 bit 0（BA0）
RJ5/CE	PORTJ	5	芯片使能（CE）控制引脚
RJ6/LB	PORTJ	6	低字节使能（LB）控制引脚
RJ7/UB	PORTJ	7	高字节使能（UB）控制引脚

**注：** 为清楚起见，这里只显示了 I/O 端口和外部总线的分配情况。在一些引脚上可能还有 1 个或多个复用功能。

# PIC18F85J11 系列

## 8.1 外部存储器总线控制

接口的操作由 MEMCON 寄存器（寄存器 8-1）控制。MEMCON 寄存器在所有程序存储器工作模式（单片机模式除外）下可用。在单片机模式中，该寄存器被禁止且不能写入。

EBDIS 位（MEMCON<7>）控制总线的操作和相关端口的功能。将 EBDIS 位清零会使能此接口，而且会禁止端口的 I/O 功能及其他在这些引脚上复用的任何功能。将该位置 1 会使能 I/O 端口和其他在这些引脚上复用的功能，但允许接口在要求进行外部存储器操作时，改写引脚上的数据。默认情况下，总是使能外部总线并禁止其他所有的 I/O 功能。

对 EBDIS 位的操作也受当前所用的程序存储器模式影响。第 8.5 节“程序存储器模式和外部存储器总线”将详细讨论 EBDIS 位的操作。

WAIT 位支持给外部存储器操作增加等待状态。第 8.3 节“等待状态”将讨论这些位的用法。

WM 位选择在总线工作在 16 位数据宽度模式时所使用的特定工作模式。第 8.6 节“16 位数据宽度模式”中对此进行更详细的讨论。当选择使用 8 位数据宽度模式时，这些位不起作用。

寄存器 8-1: MEMCON: 外部存储器总线控制寄存器

R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0
bit 15						bit 8	

### 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                      **EBDIS:** 外部总线禁止位  
1 = 当单片机访问外部存储器时使能外部总线，否则，所有的外部总线驱动器被映射为 I/O 端口  
0 = 总是使能外部总线，禁止 I/O 端口
- bit 6                      **未实现:** 读为 0
- bit 5-4                      **WAIT<1:0>:** 表读和表写操作总线等待周期计数位  
11 = 表读和表写操作将等待 0 个 T<sub>CY</sub>  
10 = 表读和表写操作将等待 1 个 T<sub>CY</sub>  
01 = 表读和表写操作将等待 2 个 T<sub>CY</sub>  
00 = 表读和表写操作将等待 3 个 T<sub>CY</sub>
- bit 3-2                      **未实现:** 读为 0
- bit 1-0                      **WM<1:0>:** 16 位数据总线宽度的 TBLWT 操作选择位  
1x = 字写模式: TABLAT0 和 TABLAT1 字输出; 当写入 TABLAT1 时 WRH 有效  
01 = 字节选择模式: 复制 TABLAT 数据的 MSB 和 LSB, WRH 和 (UB 或 LB) 将有效  
00 = 字节写模式: 复制 TABLAT 数据的 MSB 和 LSB, WRH 或 WRL 将有效

8.2 地址和数据宽度

PIC18F85J11 系列器件可以将同一存储器总线配置成不同的地址和数据宽度。两者均由 CONFIG3L 寄存器中的配置位设置。作为配置位，这意味着这些选项只能通过器件编程才能进行配置，而不能由软件控制。

BW 位选择 8 位或 16 位数据总线宽度。该位置 1（默认）会选择 16 位数据宽度。

EMB<1:0> 位决定程序存储器工作模式和地址总线宽度。可用选项有：20 位、16 位、12 位和默认单片机模式（禁止外部总线）。选择 16 位或 12 位宽度时，多出的高位线用于 I/O 功能。这些引脚不再受 EBDIS 位设置的影响。例如，选择 16 位地址模式（EMB<1:0> = 01）会禁止 A<19:16> 并允许 PORTH<3:0> 工作不受总线影响。使用较小地址宽度使用户可为特定的设计定制存储器总线以适应外部存储空间的大小，而使空出的引脚专用于 I/O 操作。

因为 EMB 位能禁止引脚用于存储总线操作，因此要注意选择的地址宽度要始终大于等于数据宽度。如果 12 位地址宽度配以 16 位数据宽度，那么总线上将无法得到数据的高 4 位。

地址和数据宽度的所有组合都要求在相同的线上复用地址和数据信息。表 8-2 总结了地址和数据复用以及使用较小地址宽度时可用的 I/O 端口。

8.2.1 外部总线的地址平移

默认情况下，PC 值就是外部总线上的地址。也就是说对单片机而言，外部存储器中地址低于片内存储器上边界的单元是不可用的。要访问这些物理单元，单片机和外部存储器间的连接逻辑必须设法对地址进行转换。

为简化接口，外部总线提供了一种扩展的单片机模式，它能自动进行地址平移。此功能由 EASHFT 配置位控制。将该位置 1 会将总线上的地址进行平移，移动量为单片机片内程序存储器的容量，且下限地址设为 0000h。这样器件就可以使用外部存储器的所有物理地址了。

8.2.2 21 位寻址

作为 20 位地址宽度操作的扩展，外部存储器总线还能寻址整个 2 MB 的存储空间。将总线地址 bit 0（BA0）控制线用作地址的最低位可以实现上述功能。在某些存储器件中，还可以使用 UB 和 LB 控制信号来选择 16 位宽数据字中的高字节和低字节。

此寻址模式可用于 8 位和某些 16 位数据宽度模式。第 8.6.3 节“16 位字节选择模式”和第 8.7 节“8 位数据宽度模式”中提供了更多详细信息。

表 8-2: 不同地址和数据宽度的地址和数据线

数据宽度	地址宽度	复用的数据和地址线 (及相应的端口)	仅地址线 (及相应的端口)	可用作 I/O 的端口
8 位	12 位	AD<7:0> (PORTD<7:0>)	AD<11:8> (PORTE<3:0>)	PORTE<7:4>, PORTH 的全部引脚
	16 位		AD<15:8> (PORTE<7:0>)	PORTH 的全部引脚
	20 位		A<19:16>, AD<15:8> (PORTH<3:0>, PORTE<7:0>)	—
16 位	16 位	AD<15:0> (PORTD<7:0>, PORTE<7:0>)	—	PORTH 的全部引脚
	20 位		A<19:16> (PORTH<3:0>)	—

## 8.3 等待状态

人们可能认为外部存储器件将工作在单片机的时钟速率下，但通常不是这样。事实上，许多器件写数据或检索数据所需的时间长于表读或表写操作允许的时间。

为补偿时间差，可以配置外部存储器总线使得在每个使用总线的表操作中添加一个固定的延时。置 1 WAIT 配置位将使能等待状态。当使能时，可通过 WAIT<1:0> 位 (MEMCON<5:4>) 设置延迟时间。延时必须是单片机指令周期的整数倍，添加在执行表操作的指令周期后。延时范围为 0 至 3 个 Tcy (默认值)。

## 8.4 端口引脚弱上拉

除高地址线 A<19:16> 外，与外部存储器总线相关的引脚均配有弱上拉。PORTG 寄存器的高 3 位控制上拉。这三位分别命名为 RDPU、REPU 和 RJPU，用于控制 PORTD、PORTE 和 PORTJ 上的弱上拉。将其中任一位置 1 会使能相应端口的弱上拉。默认情况下，器件复位时禁止所有的上拉。

## 8.5 程序存储器模式和外部存储器总线

同时使用片内和片外程序存储器时，PIC18F85J11 系列器件能工作在两种程序存储器模式下。复用端口引脚的功能取决于选定的程序存储器模式和 EBDIS 位的设置。

在**单片机模式**下，总线无效，引脚只具有端口功能。不允许写 MEMCON 寄存器。EBDIS 的复位值 (0) 被忽略， EMB 引脚用作 I/O 端口。

在**扩展单片机模式**下，外部程序存储器总线与引脚的 I/O 端口功能复用。当器件在外部程序存储空间进行取指或表读 / 表写操作时，引脚将具有外部总线功能。

如果器件只是对内部程序存储单元进行取指和访问， EBDIS 控制位将把引脚从外部存储器功能改为 I/O 端口功能。 EBDIS = 0 时，引脚用作外部总线。 EBDIS = 1 时，引脚用作 I/O 端口。

当 EBDIS = 1 时，如果器件对外部存储器进行取指或访问，引脚将切换到外部总线功能。如果在外部存储器中

执行的程序将 EBDIS 位置 1，那么将该位置 1 的操作将会延迟到程序跳转到内部存储器才得以执行。那时，引脚将由外部总线功能更改为 I/O 端口功能。

当 EBDIS = 0 时，如果器件针对内部存储器执行操作，存储器总线的地址 / 数据和控制引脚将失效。它们进入下面的状态：有效地址 / 数据引脚处于三态； CE、OE、WRH、WRL、UB 和 LB 信号为 1， ALE 和 BA0 为 0。注意，只有那些与当前地址宽度相关的引脚被强制为三态；其他引脚继续用作 I/O 功能。例如在 16 位地址宽度情况下，仅 AD<15:0> (PORTD 和 PORTE) 会受到影响； A<19:16> (PORTH<3:0>) 继续用作 I/O 功能。

在所有的外部存储器模式下，总线的优先级高于任何其他与总线共享引脚的外设，包括并行从端口和串行通信模块，此两者的优先级高于 I/O 端口。

## 8.6 16 位数据宽度模式

在 16 位数据宽度模式下，外部存储器接口能以三种不同的配置连接到外部存储器：

- 16 位字节写
- 16 位字写
- 16 位字节选择

MEMCON 寄存器中的 WM<1:0> (MEMCON<1:0>) 决定要使用的配置。这三种不同的配置使设计人员能够在使用 8 位和 16 位器件处理 16 位数据时获得最大的灵活性。

对于所有的 16 位数据宽度模式，地址锁存使能 (ALE) 引脚表明地址位 AD<15:0> 可用在外部存储器接口总线上。在地址锁存之后，输出使能信号 (OE) 将同时使能程序存储器的两个字节来形成一个 16 位的指令字。在单片机访问外部存储器 (不论读或写) 的任何时间内芯片使能信号 (CE) 始终有效；只要器件一进入休眠模式，信号就变为无效 (变为高电平)。

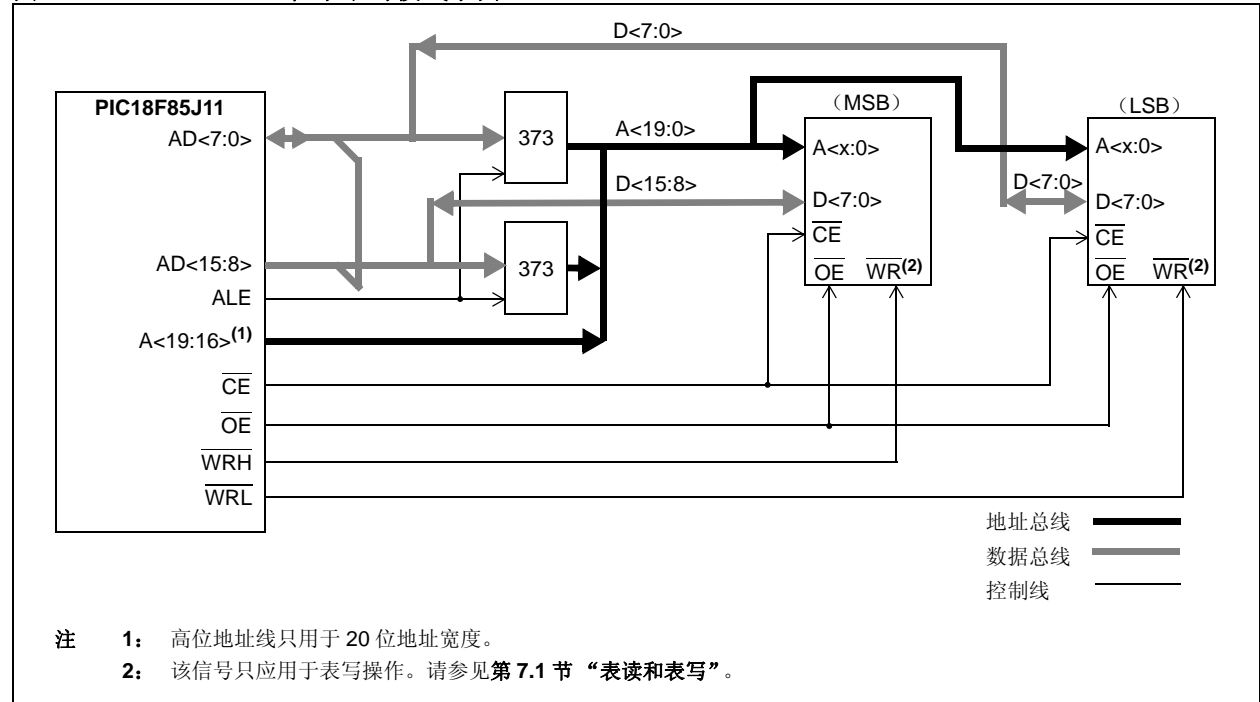
在字节选择模式下， JEDEC 标准闪存存储器需要 BA0 作为字节地址线，并使用一条 I/O 线来在字节和字模式间进行选择。其他 16 位数据宽度模式不需要 BA0。 JEDEC 标准静态 RAM 存储器将使用 UB 或 LB 信号选择字节。

## 8.6.1 16 位字节写模式

图 8-1 举例说明了 PIC18F85J11 系列器件的 16 位字节写模式。该模式用于将 2 个不同的 8 位存储器相连以进行 16 位操作。通常包括基本 EPROM 和闪存器件。允许对字节宽的外部存储器进行表写操作。

在 TBLWT 指令周期内，TABLAT 中的数据送到 AD<15:0> 总线的高字节和低字节中。TBLPTR 的 LSb 选通相应的 WRH 或 WRL 控制线。

图 8-1: 16 位字节写模式示例



# PIC18F85J11 系列

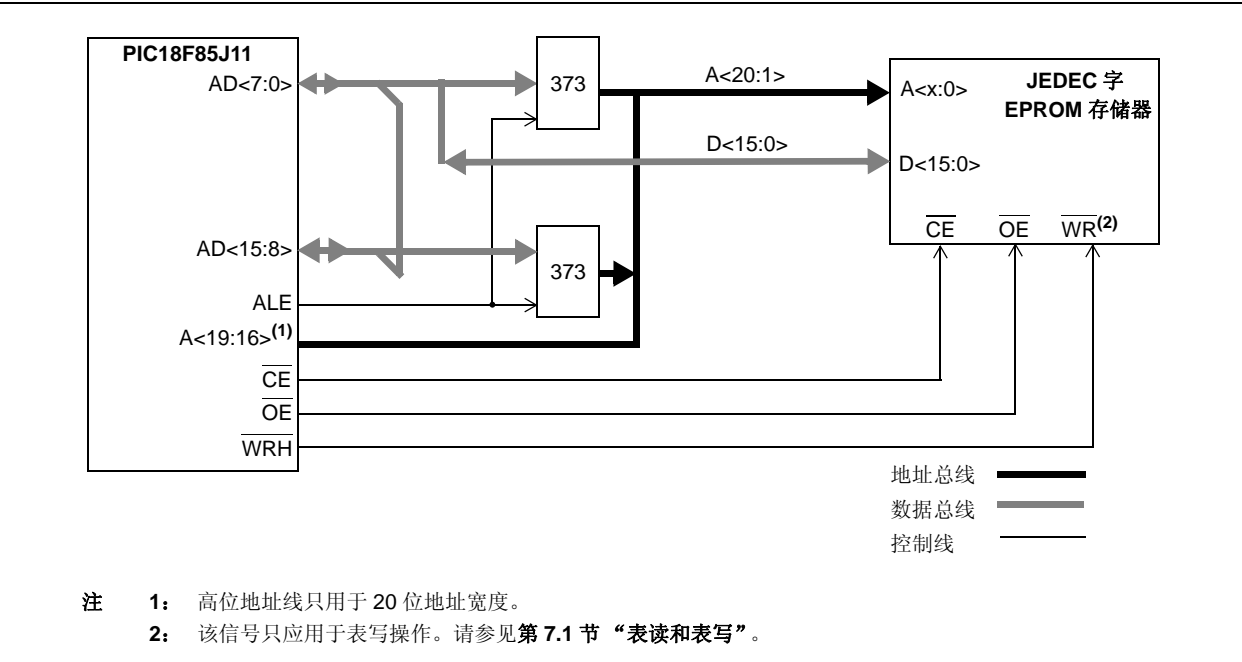
## 8.6.2 16 位字写模式

图 8-2 举例说明了 PIC18F85J11 系列器件的 16 位字写模式。该模式用于字宽度的存储器，包括一些 EPROM 和闪存存储器。该模式允许从所有的 16 位存储器取操作码和执行表读操作，而对所有的字宽度的外部存储器执行表写操作。这种方法将访问奇地址和偶地址的 TBLWT 周期区分开来。

在写入偶地址 ( $TBLPTR<0> = 0$ ) 的 TBLWT 周期中，TABLAT 数据被传送到保持锁存器中，外部地址数据总线在总线周期中传送数据的时刻呈现为三态。写信号不会被激活。

在写入奇地址 ( $TBLPTR<0> = 1$ ) 的 TBLWT 周期中，TABLAT 中的数据被传送到  $AD<15:0>$  总线的高字节。保持锁存器的内容被传送到  $AD<15:0>$  总线的低字节。 $\overline{WRH}$  信号在每个写周期被选通，未使用  $\overline{WRL}$  引脚。 $\overline{BA0}$  引脚上的信号表示  $TBLPTR$  的  $LSb$ ，但该引脚是悬空的。实际上使用  $\overline{UB}$  和  $\overline{LB}$  信号来选择两个字节。这种方法的明显缺陷是表写操作必须在特定字边界上成对完成，才能正确地写一个字存储单元。

图 8-2: 16 位字写模式示例





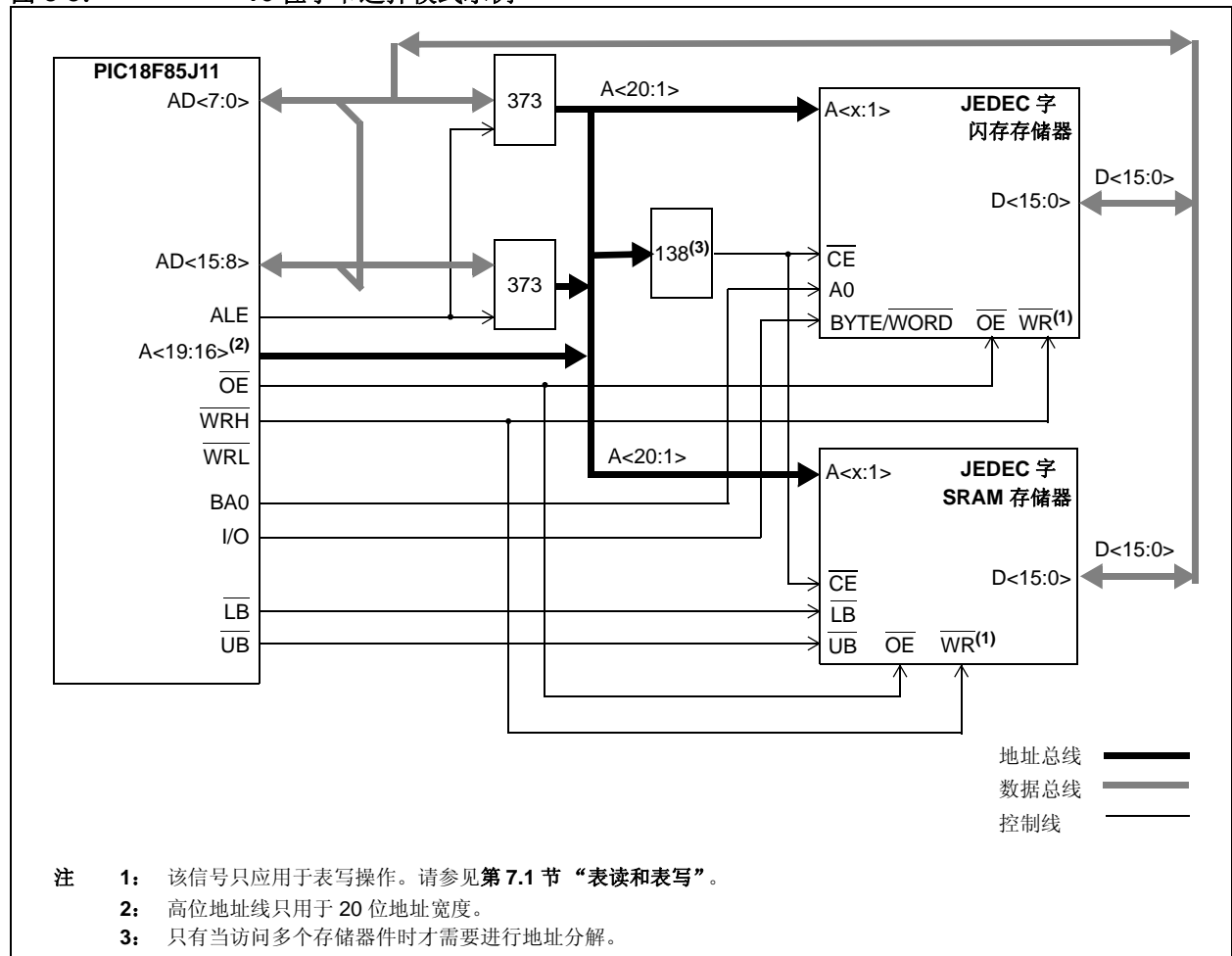
## 8.6.3 16 位字节选择模式

图 8-3 举例说明了 16 位字节选择模式。该模式允许对字宽度的外部存储器进行表写操作，同时还可以进行字节选择。这通常包括字宽度的闪存和 SRAM 器件。

在 TBLWT 周期内，TABLAT 中的数据送到 AD<15:0> 总线的高字节和低字节中。WRH 信号在每个写周期被选通；WRL 引脚未被使用。BA0 或 UB/LB 信号用于根据 TBLPTR 寄存器中的最低有效位选择要写的字节。

闪存和 SRAM 器件使用不同的控制信号组合来实现字节选择模式。JEDEC 标准闪存存储器要求将一个控制器 I/O 端口引脚连接到存储器的 BYTE/WORD 引脚，以提供选择信号。它们也使用控制器的 BA0 信号作为字节地址，另一方面 JEDEC 标准静态 RAM 存储器将使用 UB 或 LB 信号选择字节。

图 8-3: 16 位字节选择模式示例



# PIC18F85J11 系列

## 8.6.4 16 位模式时序

送到外部存储器总线上的控制信号在不同的操作模式下是不同的。图 8-4 和图 8-5 所示为典型的信号时序图。

图 8-4: 执行 TBLRD 指令时的外部存储器总线时序（扩展单片机模式）

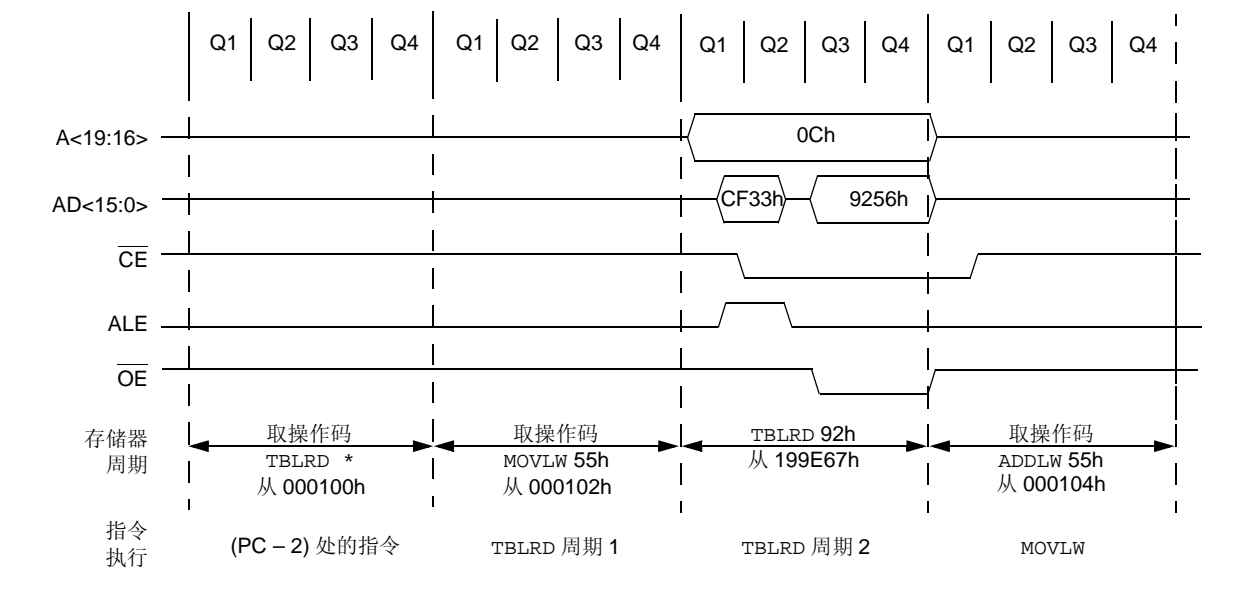
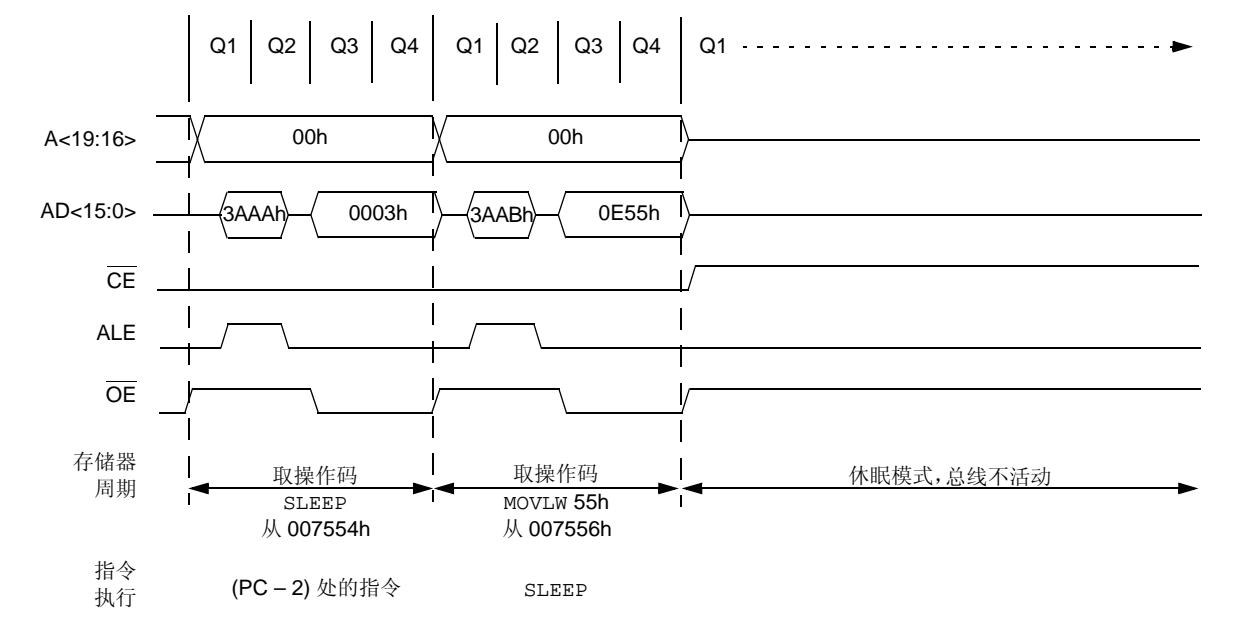


图 8-5: 执行 SLEEP 指令时的外部存储器总线时序（扩展单片机模式）



## 8.7 8 位数据宽度模式

在 8 位数据宽度模式下，外部存储器总线只工作在复用模式下；即数据总线与地址总线的低 8 位复用。

图 8-6 举例说明了 80 引脚器件的 8 位复用模式。该模式用于连接 1 个 8 位存储器以进行 16 位的操作。在数据 / 地址复用的总线上将一条指令作为 2 个 8 位字节取出，这两个字节在一个指令周期 ( $T_{CY}$ ) 内依次被取出。所以，设计人员必须以  $1/2 T_{CY}$  (2 倍指令速率) 为单位计算时序，并据此选择外部存储器。为了选择合适的存储器速度，在考虑建立时间和保持时间的同时，还必须考虑连接逻辑传输延时。

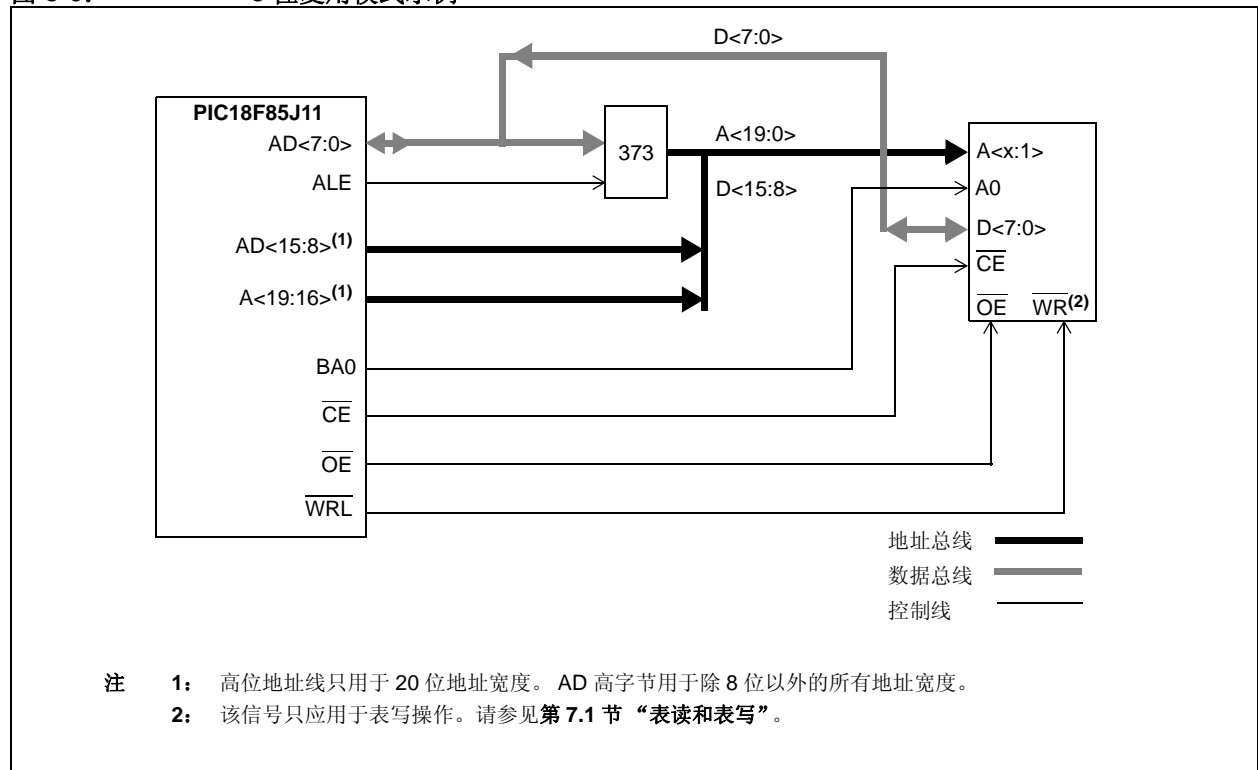
地址锁存使能 (Address Latch Enable, ALE) 引脚表明地址位  $AD<15:0>$  在外部存储器接口总线上是否可用。输出使能信号 ( $\overline{OE}$ ) 将在指令周期的部分时间内

使能程序存储器的一个字节，然后更改  $BA0$ ，从而使能第二个字节以构成 16 位的指令字。在该模式下，地址的最低有效位  $BA0$  必须连接到存储器件。在单片机访问外部存储器 (不论读或写) 的任何时间内芯片使能信号 ( $\overline{CE}$ ) 始终有效。只要器件一进入休眠模式，信号就变为无效 (变为高电平)。

通常包括基本 EPROM 和闪存器件。允许对字节宽的外部存储器进行表写操作。

在 TBLWT 指令周期内，TABLAT 中的数据送到  $AD<15:0>$  总线的高字节和低字节中。TBLPTR 的 LSB 选通  $BA0$  控制线的相应电平。

图 8-6: 8 位复用模式示例



# PIC18F85J11 系列

## 8.7.1 8 位模式时序

送到外部存储器总线上的控制信号在不同的操作模式下是不同的。图 8-7 和图 8-8 所示为典型的信号时序图。

图 8-7: 执行 TBLRD 指令时的外部存储器总线时序（扩展单片机模式）

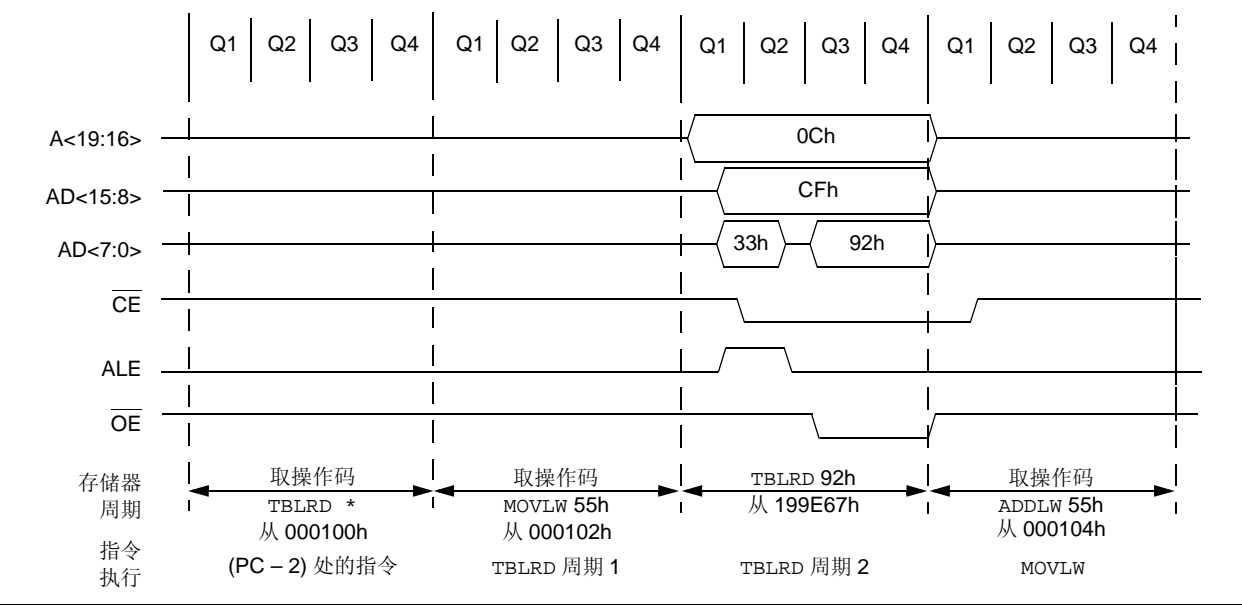
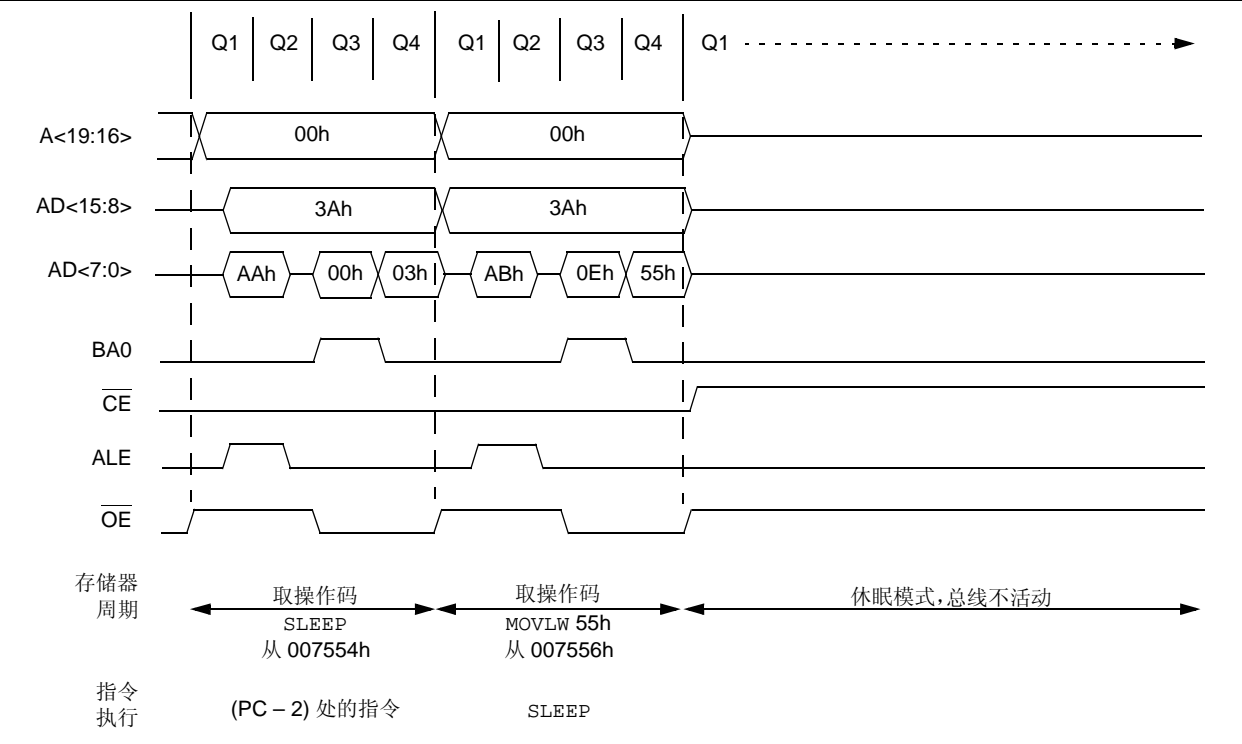


图 8-8: 执行 SLEEP 指令时的外部存储器总线时序（扩展单片机模式）



## 8.8 功耗管理模式下的操作

在所有功耗管理运行模式下，外部总线都能继续正常工作。如果选择一个低速的时钟源，总线操作将运行在该速率下。在这些情况中，如果使能了等待状态并把它添加到外部存储器操作中，可能会导致对外部存储器的访问时间过长。如果希望工作在低功耗运行模式下，用户应该在应用程序中调整在低时钟速率下访问存储器的时间。

在休眠和空闲模式下，单片机内核不需访问数据；总线操作暂停。外部总线的状态被冻结，地址 / 数据引脚和多数控制引脚保持在进入该模式前的状态。惟一可能变化的是 CE、LB 和 UB 引脚，它们保持逻辑高电平。

# PIC18F85J11 系列

---

注:

## 9.0 8 x 8 硬件乘法器

### 9.1 简介

所有的 PIC18 器件均包含一个 8 x 8 硬件乘法器（乘法器是 ALU 的一部分）。该乘法器可执行无符号运算并产生一个 16 位运算结果，该结果存储在 16 位乘积寄存器 PRODH:PRODL 中。该乘法器执行的运算不会影响 STATUS 寄存器中的任何标志。

通过硬件执行乘法运算只需要 1 个指令周期。硬件乘法器具有更高的计算吞吐量并减少了乘法算法的代码长度，从而可在许多先前仅能使用数字信号处理器的应用中使用 PIC18 器件。表 9-1 给出了各种硬件和软件乘法运算的比较，包括所需存储空间和执行时间。

### 9.2 工作原理

例 9-1 给出了一个 8 x 8 无符号乘法运算的指令序列。当已在 WREG 寄存器中装入了一个乘数时，实现该运算仅需一条指令。

例 9-2 给出了一个 8 x 8 有符号乘法运算的指令序列。要弄清结果的符号位，必须检查每个乘数的最高有效位（MSb），并做相应的减法。

**例 9-1: 8 x 8 无符号乘法程序**

```
MOVF    ARG1, W    ;
MULWF   ARG2        ; ARG1 * ARG2 ->
                        ; PRODH:PRODL
```

**例 9-2: 8 x 8 有符号乘法程序**

```
MOVF    ARG1, W
MULWF   ARG2        ; ARG1 * ARG2 ->
                        ; PRODH:PRODL

BTFSC   ARG2, SB    ; Test Sign Bit
SUBWF   PRODH, F    ; PRODH = PRODH
                        ; - ARG1

MOVF    ARG2, W
BTFSC   ARG1, SB    ; Test Sign Bit
SUBWF   PRODH, F    ; PRODH = PRODH
                        ; - ARG2
```

**表 9-1: 各种乘法运算的性能比较**

程序	乘法实现方法	程序 存储空间 (字)	周期数 (最多)	时间		
				40 MHz 时	10 MHz 时	4 MHz 时
8 x 8 无符号	软件乘法	13	69	6.9 μs	27.6 μs	69 μs
	硬件乘法	1	1	100 ns	400 ns	1 μs
8 x 8 有符号	软件乘法	33	91	9.1 μs	36.4 μs	91 μs
	硬件乘法	6	6	600 ns	2.4 μs	6 μs
16 x 16 无符号	软件乘法	21	242	24.2 μs	96.8 μs	242 μs
	硬件乘法	28	28	2.8 μs	11.2 μs	28 μs
16 x 16 有符号	软件乘法	52	254	25.4 μs	102.6 μs	254 μs
	硬件乘法	35	40	4.0 μs	16.0 μs	40 μs

# PIC18F85J11 系列

例 9-3 给出了一个 16 x 16 无符号乘法运算的指令序列。  
公式 9-1 为所使用的算法。32 位结果存储在 4 个寄存器 (RES3:RES0) 中。

## 公式 9-1: 16 x 16 无符号乘法算法

$$\begin{aligned}\text{RES3:RES0} &= \text{ARG1H:ARG1L} \cdot \text{ARG2H:ARG2L} \\ &= (\text{ARG1H} \cdot \text{ARG2H} \cdot 2^{16}) + \\ &\quad (\text{ARG1H} \cdot \text{ARG2L} \cdot 2^8) + \\ &\quad (\text{ARG1L} \cdot \text{ARG2H} \cdot 2^8) + \\ &\quad (\text{ARG1L} \cdot \text{ARG2L})\end{aligned}$$

## 例 9-3: 16 x 16 无符号乘法程序

```
MOVF ARG1L, W
MULWF ARG2L          ; ARG1L * ARG2L->
                      ; PRODH:PRODL

MOVFF PRODH, RES1    ;
MOVFF PRODL, RES0    ;
;

MOVF ARG1H, W
MULWF ARG2H          ; ARG1H * ARG2H->
                      ; PRODH:PRODL

MOVFF PRODH, RES3    ;
MOVFF PRODL, RES2    ;
;

MOVF ARG1L, W
MULWF ARG2H          ; ARG1L * ARG2H->
                      ; PRODH:PRODL

MOVF PRODL, W        ;
ADDWF RES1, F         ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F        ;
CLRF WREG             ;
ADDWFC RES3, F        ;
;

MOVF ARG1H, W        ;
MULWF ARG2L          ; ARG1H * ARG2L->
                      ; PRODH:PRODL

MOVF PRODL, W        ;
ADDWF RES1, F         ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F        ;
CLRF WREG             ;
ADDWFC RES3, F        ;
```

例 9-4 给出了一个 16 x 16 有符号乘法运算的指令序列。  
公式 9-2 为所使用的算法。32 位结果存储在 4 个寄存器 (RES3:RES0) 中。要弄清结果的符号位, 必须检查每个乘数的最高有效位 (MSb), 并做相应的减法。

## 公式 9-2: 16 x 16 有符号乘法算法

$$\begin{aligned}\text{RES3:RES0} &= \text{ARG1H:ARG1L} \cdot \text{ARG2H:ARG2L} \\ &= (\text{ARG1H} \cdot \text{ARG2H} \cdot 2^{16}) + \\ &\quad (\text{ARG1H} \cdot \text{ARG2L} \cdot 2^8) + \\ &\quad (\text{ARG1L} \cdot \text{ARG2H} \cdot 2^8) + \\ &\quad (\text{ARG1L} \cdot \text{ARG2L}) + \\ &\quad (-1 \cdot \text{ARG2H} < 7 > \cdot \text{ARG1H:ARG1L} \cdot 2^{16}) + \\ &\quad (-1 \cdot \text{ARG1H} < 7 > \cdot \text{ARG2H:ARG2L} \cdot 2^{16})\end{aligned}$$

## 例 9-4: 16 x 16 有符号乘法程序

```
MOVF ARG1L, W
MULWF ARG2L          ; ARG1L * ARG2L ->
                      ; PRODH:PRODL

MOVFF PRODH, RES1    ;
MOVFF PRODL, RES0    ;
;

MOVF ARG1H, W
MULWF ARG2H          ; ARG1H * ARG2H ->
                      ; PRODH:PRODL

MOVFF PRODH, RES3    ;
MOVFF PRODL, RES2    ;
;

MOVF ARG1L, W
MULWF ARG2H          ; ARG1L * ARG2H ->
                      ; PRODH:PRODL

MOVF PRODL, W        ;
ADDWF RES1, F         ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F        ;
CLRF WREG             ;
ADDWFC RES3, F        ;
;

MOVF ARG1H, W        ;
MULWF ARG2L          ; ARG1H * ARG2L ->
                      ; PRODH:PRODL

MOVF PRODL, W        ;
ADDWF RES1, F         ; Add cross
MOVF PRODH, W        ; products
ADDWFC RES2, F        ;
CLRF WREG             ;
ADDWFC RES3, F        ;
;

BTFSF ARG2H, 7        ; ARG2H:ARG2L neg?
BRA SIGN_ARG1         ; no, check ARG1

MOVF ARG1L, W        ;
SUBWF RES2            ;
MOVF ARG1H, W        ;
SUBWFB RES3           ;
;

SIGN_ARG1
BTFSF ARG1H, 7        ; ARG1H:ARG1L neg?
BRA CONT_CODE         ; no, done
MOVF ARG2L, W        ;
SUBWF RES2            ;
MOVF ARG2H, W        ;
SUBWFB RES3           ;
;

CONT_CODE
:
```



## 10.0 中断

PIC18F85J11 系列器件的各个成员提供多个中断源及一个中断优先级功能，该功能可以给大多数中断源分配高优先级或者低优先级。高优先级中断向量位于 0008h，低优先级中断向量位于 0018h。高优先级中断事件将中断任何正在处理的低优先级中断。

有 13 个寄存器用于控制中断操作。它们是：

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1、PIR2 和 PIR3
- PIE1、PIE2 和 PIE3
- IPR1、IPR2 和 IPR3

推荐使用随 MPLAB® IDE 提供的 Microchip 头文件命名这些寄存器中的位。这使得汇编器 / 编译器能够自动识别指定寄存器内的这些位。

通常，用三个位来控制中断源的操作。它们是：

- **标志位**表明发生了中断事件
- **允许位**允许程序跳转到中断向量地址处执行（当标志位置 1 时）
- **优先级位**用于选择高优先级或低优先级

通过将 IPEN 位（RCON<7>）置 1，可允许中断优先级功能。当允许了中断优先级时，有 2 位可允许全局中断。将 GIEH 位（INTCON<7>）置 1，可允许所有优先级位已置 1（高优先级）的中断。将 GIEL 位（INTCON<6>）置 1，可允许所有优先级位已清零（低优先级）的中断。当中断标志位、允许位及相应的全局中断允许位均被置 1 时，中断将立即跳转到地址 0008h 或 0018h，具体地址取决于优先级位的设置。通过设置相应的允许位可以禁止各个中断。

当 IPEN 位被清零（默认状态）时，便会禁止中断优先级功能，此时中断与 PIC® MCU 中档系列器件相兼容。在兼容模式下，各个中断源的中断优先级位均不起作用。INTCON<6> 是 PEIE 位，用于允许 / 禁止所有的外设中断源。INTCON<7> 是 GIE 位，用于允许 / 禁止所有中断源。在兼容模式下，所有中断均跳转到地址 0008h。

当响应中断时，全局中断允许位被清零以禁止其他中断。如果清零 IPEN 位，全局中断允许位就是 GIE 位。如果使用了中断优先级，这个位就是 GIEH 位或者 GIEL 位。高优先级中断源可以中断低优先级中断。在处理高优先级中断时，低优先级中断将不被响应。

返回地址被压入堆栈，中断向量地址（0008h 或 0018h）被装入 PC。可以在中断服务程序（Interrupt Service Routine, ISR）中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，以避免重复响应中断。

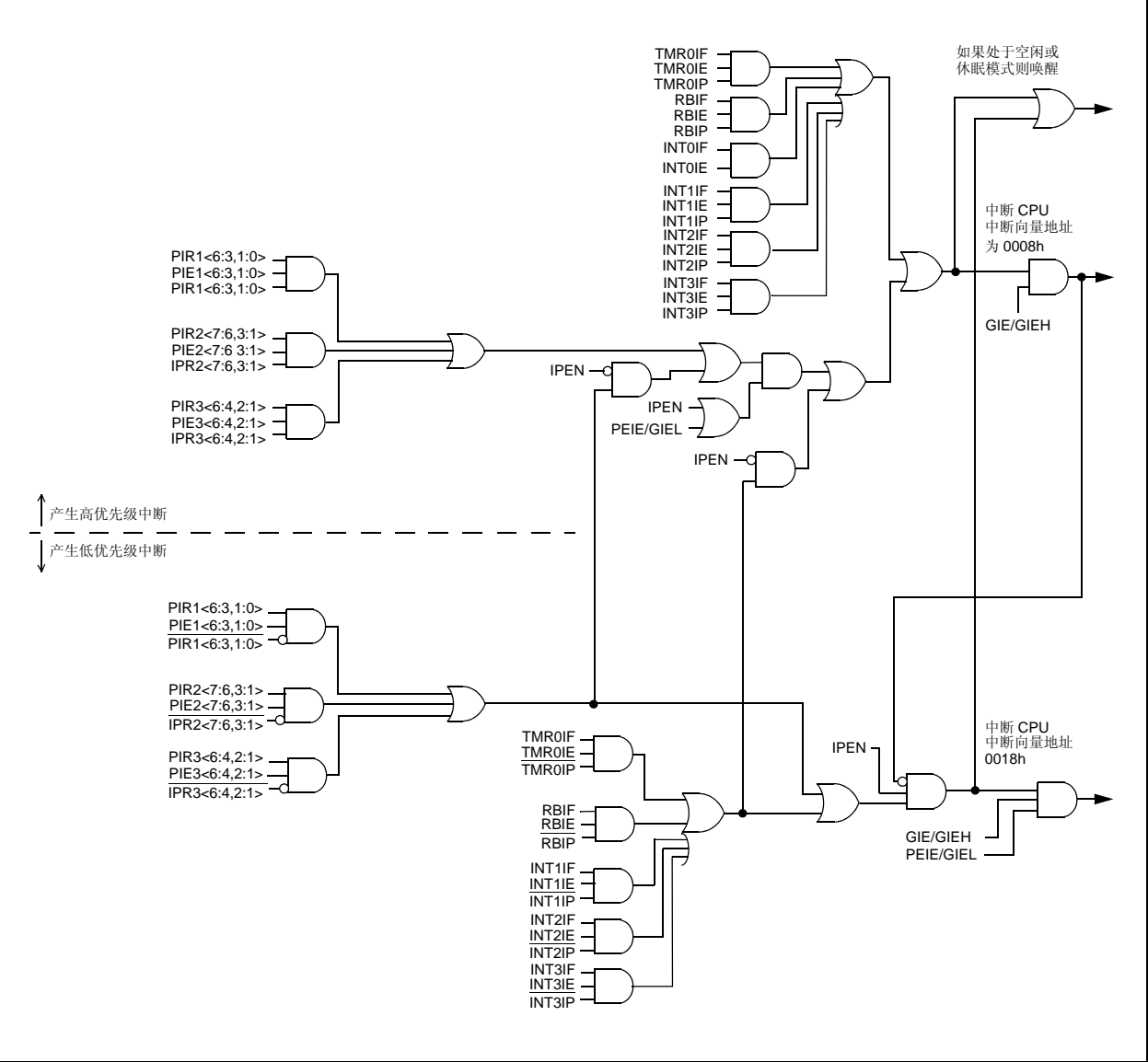
执行“从中断返回”指令 RETFIE 将退出中断程序，同时将 GIE 位（若使用中断优先级，则为 GIEH 或 GIEL 位）置 1，从而重新允许中断。

对于外部中断事件，诸如 INTx 引脚中断或者 PORTB 输入电平变化中断，中断响应延时将会是 3 到 4 个指令周期。对于单周期或双周期指令，中断响应延时完全相同。不管对应的中断允许位和 GIE 位状态如何，各中断标志位均被置 1。

<b>注：</b>	当任何中断被允许时，不要使用 MOVFF 指令来修改中断控制寄存器，否则可能导致单片机操作出错。
-----------	--

# PIC18F85J11 系列

图 10-1: PIC18F85J11 系列中断逻辑



## 10.1 INTCON 寄存器

INTCON 寄存器是可读写的寄存器，包含各个允许位、优先级位和标志位。

**注：** 当中断条件产生时，不管相应的中断允许位或全局中断允许位的状态如何，中断标志位都将置 1。用户软件应在允许一个中断前，确保先将相应的中断标志位清零。中断标志位可由软件查询。

寄存器 10-1: 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/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF <sup>(1)</sup>
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/GIEH:** 全局中断允许位  
             当 IPEN = 0:  
             1 = 允许所有未屏蔽的中断  
             0 = 禁止所有中断  
             当 IPEN = 1:  
             1 = 允许所有高优先级中断  
             0 = 禁止所有中断
- bit 6      **PEIE/GIEL:** 外设中断允许位  
             当 IPEN = 0:  
             1 = 允许所有未屏蔽的外设中断  
             0 = 禁止所有外设中断  
             当 IPEN = 1:  
             1 = 允许所有低优先级的外设中断  
             0 = 禁止所有低优先级的外设中断
- bit 5      **TMR0IE:** TMR0 溢出中断允许位  
             1 = 允许 TMR0 溢出中断  
             0 = 禁止 TMR0 溢出中断
- bit 4      **INT0IE:** INT0 外部中断允许位  
             1 = 允许 INT0 外部中断  
             0 = 禁止 INT0 外部中断
- bit 3      **RBIE:** RB 端口电平变化中断允许位  
             1 = 允许 RB 端口电平变化中断  
             0 = 禁止 RB 端口电平变化中断
- bit 2      **TMR0IF:** TMR0 溢出中断标志位  
             1 = TMR0 寄存器已发生溢出 （必须由软件清零）  
             0 = TMR0 寄存器未发生溢出
- bit 1      **INT0IF:** INT0 外部中断标志位  
             1 = 发生了 INT0 外部中断 （必须由软件清零）  
             0 = 未发生 INT0 外部中断
- bit 0      **RBIF:** RB 端口电平变化中断标志位 <sup>(1)</sup>  
             1 = RB<7:4> 引脚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 = RB<7:4> 引脚的电平状态没有改变

**注 1:** 引脚上电平变化会一直不断地将此位置 1。读取 PORTB，然后等待一个额外的指令周期可以结束这种情况，并将该位清零。

# PIC18F85J11 系列

寄存器 10-2: INTCON2: 中断控制寄存器 2

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBP <sub>U</sub>	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP
bit 7							bit 0

<b>图注:</b>			
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	<b>RBP<sub>U</sub></b> : PORTB 上拉使能位 1 = 禁止所有 PORTB 上拉 0 = 按各个端口锁存值使能 PORTB 上拉
bit 6	<b>INTEDG0</b> : 外部中断 0 边沿选择位 1 = 上升沿触发中断 0 = 下降沿触发中断
bit 5	<b>INTEDG1</b> : 外部中断 1 边沿选择位 1 = 上升沿触发中断 0 = 下降沿触发中断
bit 4	<b>INTEDG2</b> : 外部中断 2 边沿选择位 1 = 上升沿触发中断 0 = 下降沿触发中断
bit 3	<b>INTEDG3</b> : 外部中断 3 边沿选择位 1 = 上升沿触发中断 0 = 下降沿触发中断
bit 2	<b>TMR0I</b> : TMR0 溢出中断优先级位 1 = 高优先级 0 = 低优先级
bit 1	<b>INT3IP</b> : INT3 外部中断优先级位 1 = 高优先级 0 = 低优先级
bit 0	<b>RBIP</b> : RB 端口电平变化中断优先级位 1 = 高优先级 0 = 低优先级

<b>注:</b>	当中断条件产生时, 不管相应的中断允许位或全局中断允许位的状态如何, 中断标志位都将置 1。用户软件应在允许一个中断前, 确保先将相应的中断标志位清零。中断标志位可由软件查询。
-----------	--

## 寄存器 10-3: INTCON3: 中断控制寄存器 3

R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF
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            **INT2IP:** INT2 外部中断优先级位  
                  1 = 高优先级  
                  0 = 低优先级
- bit 6            **INT1IP:** INT1 外部中断优先级位  
                  1 = 高优先级  
                  0 = 低优先级
- bit 5            **INT3IE:** INT3 外部中断允许位  
                  1 = 允许 INT3 外部中断  
                  0 = 禁止 INT3 外部中断
- bit 4            **INT2IE:** INT2 外部中断允许位  
                  1 = 允许 INT2 外部中断  
                  0 = 禁止 INT2 外部中断
- bit 3            **INT1IE:** INT1 外部中断允许位  
                  1 = 允许 INT1 外部中断  
                  0 = 禁止 INT1 外部中断
- bit 2            **INT3IF:** INT3 外部中断标志位  
                  1 = 发生了 INT3 外部中断 (必须由软件清除)  
                  0 = 未发生 INT3 外部中断
- bit 1            **INT2IF:** INT2 外部中断标志位  
                  1 = 发生了 INT2 外部中断 (必须由软件清除)  
                  0 = 未发生 INT2 外部中断
- bit 0            **INT1IF:** INT1 外部中断标志位  
                  1 = 发生了 INT1 外部中断 (必须由软件清零)  
                  0 = 未发生 INT1 外部中断

**注:** 当中断条件产生时, 不管相应的中断允许位或全局中断允许位的状态如何, 中断标志位都将置 1。用户软件应在允许一个中断前, 确保先将相应的中断标志位清零。中断标志位可由软件查询。

# PIC18F85J11 系列

## 10.2 PIR 寄存器

PIR 寄存器包含各外设中断的标志位。根据外设中断源的数量，有三个外设中断请求（标志）寄存器（PIR1、PIR2 和 PIR3）。

- 注 1:** 当中断条件产生时，不管相应的中断允许位或全局中断允许位 **GIE**（INTCON<7>）的状态如何，中断标志位都将置 1。
- 2:** 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；同时在处理完中断后，也应该将相应的中断标志位清零。

寄存器 10-4: PIR1: 外设中断请求（标志）寄存器 1

R/W-0	R/W-0	R-0	R-0	R/W-0	U-0	R/W-0	R/W-0
PSPIF	ADIF	RC1IF	TX1IF	SSPIF	—	TMR2IF	TMR1IF
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      **PSPIF:** 并行从端口读 / 写中断标志位  
            1 = 发生了读或写操作（必须由软件清零）  
            0 = 未发生读或写操作
- bit 6      **ADIF:** A/D 转换器中断标志位  
            1 = A/D 转换已完成（必须由软件清零）  
            0 = A/D 转换未完成
- bit 5      **RC1IF:** EUSART 接收中断标志位  
            1 = EUSART 接收寄存器 RCREG1 已满（当读 RCREG1 时清零）  
            0 = EUSART 接收缓冲器为空
- bit 4      **TX1IF:** EUSART 发送中断标志位  
            1 = EUSART 发送缓冲器 TXREG1 为空（当写 TXREG1 时清零）  
            0 = EUSART 发送缓冲器已满
- bit 3      **SSPIF:** 主同步串口中断标志位  
            1 = 发送 / 接收已完成（必须由软件清零）  
            0 = 等待发送 / 接收
- bit 2      **未实现:** 读为 0
- bit 1      **TMR2IF:** TMR2 与 PR2 匹配中断标志位  
            1 = TMR2 与 PR2 发生匹配（必须由软件清零）  
            0 = TMR2 与 PR2 未发生匹配
- bit 0      **TMR1IF:** TMR1 溢出中断标志位  
            1 = TMR1 寄存器已发生溢出（必须由软件清零）  
            0 = TMR1 寄存器未发生溢出

## 寄存器 10-5: PIR2: 外设中断请求 (标志) 寄存器 2

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
OSCFIF	CMIF	—	—	BCLIF	LVDIF	TMR3IF	—
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              **OSCFIF:** 振荡器故障中断标志位  
                     1 = 器件振荡器发生故障, 改由 INTOSC 作为时钟输入 (必须由软件清零)  
                     0 = 器件时钟正常运行
- bit 6              **CMIF:** 比较器中断标志位  
                     1 = 比较器输入已改变 (必须由软件清零)  
                     0 = 比较器输入未改变
- bit 5-4            **未实现:** 读为 0
- bit 3              **BCLIF:** 总线冲突中断标志位  
                     1 = 发生了总线冲突 (必须由软件清零)  
                     0 = 未发生总线冲突
- bit 2              **LVDIF:** 低电压检测中断标志位  
                     1 = 发生低电压条件 (必须由软件清零)  
                     0 = 器件电压高于稳压器低电压检测跳变点
- bit 1              **TMR3IF:** TMR3 溢出中断标志位  
                     1 = TMR3 寄存器已溢出 (必须由软件清零)  
                     0 = TMR3 寄存器未溢出
- bit 0              **未实现:** 读为 0

# PIC18F85J11 系列

寄存器 10-6:      **PIR3: 外设中断请求（标志）寄存器 3**

U-0	U-0	R-0	R-0	U-0	R/W-0	R/W-0	U-0
—	—	RC2IF	TX2IF	—	CCP2IF	CCP1IF	—
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	<b>未实现:</b> 读为 0
bit 5	<b>RC2IF:</b> AUSART 接收中断标志位 1 = AUSART 接收缓冲器 RCREG2 已满 (当读 RCREG2 时清零) 0 = AUSART 接收缓冲器为空
bit 4	<b>TX2IF:</b> AUSART 发送中断标志位 1 = AUSART 发送缓冲器 TXREG2 为空 (当写 TXREG2 时清零) 0 = AUSART 发送缓冲器已满
bit 3	<b>未实现:</b> 读为 0
bit 2	<b>CCP2IF:</b> CCP2 中断标志位 <u>捕捉模式:</u> 1 = 发生了 TMR1/TMR3 寄存器捕捉 (必须由软件清零) 0 = 未发生 TMR1/TMR3 寄存器捕捉 <u>比较模式:</u> 1 = 发生了 TMR1/TMR3 寄存器的比较匹配 (必须由软件清零) 0 = 未发生 TMR1/TMR3 寄存器的比较匹配 <u>PWM 模式:</u> 在此模式下未使用。
bit 1	<b>CCP1IF:</b> CCP1 中断标志位 <u>捕捉模式:</u> 1 = 发生了 TMR1/TMR3 寄存器捕捉 (必须由软件清零) 0 = 未发生 TMR1/TMR3 寄存器捕捉 <u>比较模式:</u> 1 = 发生了 TMR1/TMR3 寄存器的比较匹配 (必须由软件清零) 0 = 未发生 TMR1/TMR3 寄存器的比较匹配 <u>PWM 模式:</u> 在此模式下未使用。
bit 0	<b>未实现:</b> 读为 0



## 10.3 PIE 寄存器

PIE 寄存器包含各外设中断的允许位。根据外设中断源的数量，有 3 个外设中断允许寄存器（**PIE1**、**PIE2** 和 **PIE3**）。当 **IPEN = 0** 时，要允许任何外设中断就必须将 **PEIE** 位置 1。

**寄存器 10-7:        **PIE1**: 外设中断允许寄存器 1**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
<b>PSPIE</b>	<b>ADIE</b>	<b>RC1IE</b>	<b>TX1IE</b>	<b>SSPIE</b>	—	<b>TMR2IE</b>	<b>TMR1IE</b>
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        **PSPIE**: 并行从端口读 / 写中断允许位  
              1 = 允许 PSP 读 / 写中断  
              0 = 禁止 PSP 读 / 写中断
- bit 6        **ADIE**: A/D 转换器中断允许位  
              1 = 允许 A/D 中断  
              0 = 禁止 A/D 中断
- bit 5        **RC1IE**: EUSART 接收中断允许位  
              1 = 允许 EUSART 接收中断  
              0 = 禁止 EUSART 接收中断
- bit 4        **TX1IE**: EUSART 发送中断允许位  
              1 = 允许 EUSART 发送中断  
              0 = 禁止 EUSART 发送中断
- bit 3        **SSPIE**: 主同步串口中断允许位  
              1 = 允许 MSSP 中断  
              0 = 禁止 MSSP 中断
- bit 2        **未实现**: 读为 0
- bit 1        **TMR2IE**: TMR2 与 PR2 匹配中断允许位  
              1 = 允许 TMR2 与 PR2 匹配中断  
              0 = 禁止 TMR2 与 PR2 匹配中断
- bit 0        **TMR1IE**: TMR1 溢出中断允许位  
              1 = 允许 TMR1 溢出中断  
              0 = 禁止 TMR1 溢出中断

# PIC18F85J11 系列

寄存器 10-8:        **PIE2: 外设中断允许寄存器 2**

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
OSCFIE	CMIE	—	—	BCLIE	LVDIE	TMR3IE	—
bit 7							bit 0

<b>图注:</b>			
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	<b>OSCFIE:</b> 振荡器故障中断允许位 1 = 允许 0 = 禁止
bit 6	<b>CMIE:</b> 比较器中断允许位 1 = 允许 0 = 禁止
bit 5-4	<b>未实现:</b> 读为 0
bit 3	<b>BCLIE:</b> 总线冲突中断允许位 1 = 允许 0 = 禁止
bit 2	<b>LVDIE:</b> 低电压检测中断允许位 1 = 允许 0 = 禁止
bit 1	<b>TMR3IE:</b> TMR3 溢出中断允许位 1 = 允许 0 = 禁止
bit 0	<b>未实现:</b> 读为 0

## 寄存器 10-9: PIE3: 外设中断允许寄存器 3

U-0	U-0	R-0	R-0	U-0	R/W-0	R/W-0	U-0
—	—	RC2IE	TX2IE	—	CCP2IE	CCP1IE	—
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      **RC2IE:** AUSART 接收中断允许位  
             1 = 允许  
             0 = 禁止  
 bit 4      **TX2IE:** AUSART 发送中断允许位  
             1 = 允许  
             0 = 禁止  
 bit 3      **未实现:** 读为 0  
 bit 2      **CCP2IE:** CCP2 中断允许位  
             1 = 允许  
             0 = 禁止  
 bit 1      **CCP1IE:** CCP1 中断允许位  
             1 = 允许  
             0 = 禁止  
 bit 0      **未实现:** 读为 0

# PIC18F85J11 系列

## 10.4 IPR 寄存器

IPR 寄存器包含各外设中断的优先级位。根据外设中断源的数量，有 3 个外设中断优先级寄存器（IPR1、IPR2 和 IPR3）。使用优先级位要求将中断优先级允许（IPEN）位置 1。

寄存器 10-10: IPR1: 外设中断优先级寄存器 1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	R/W-1
PSPIP	ADIP	RC1IP	TX1IP	SSPIP	—	TMR2IP	TMR1IP
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	<b>PSPIP:</b> 并行从端口读 / 写中断优先级位 1 = 高优先级 0 = 低优先级
bit 6	<b>ADIP:</b> A/D 转换器中断优先级位 1 = 高优先级 0 = 低优先级
bit 5	<b>RC1IP:</b> EUSART 接收中断优先级位 1 = 高优先级 0 = 低优先级
bit 4	<b>TX1IP:</b> EUSART 发送中断优先级位 1 = 高优先级 0 = 低优先级
bit 3	<b>SSPIP:</b> 主同步串口中断优先级位 1 = 高优先级 0 = 低优先级
bit 2	<b>未实现:</b> 读为 0
bit 1	<b>TMR2IP:</b> TMR2 与 PR2 匹配中断优先级位 1 = 高优先级 0 = 低优先级
bit 0	<b>TMR1IP:</b> TMR1 溢出中断优先级位 1 = 高优先级 0 = 低优先级

## 寄存器 10-11: IPR2: 外设中断优先级寄存器 2

R/W-1	R/W-1	U-0	U-0	R/W-1	R/W-1	R/W-1	U-0
OSCFIP	CMIP	—	—	BCLIP	LVDIP	TMR3IP	—
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 **OSCFIP:** 振荡器故障中断优先级位

1 = 高优先级

0 = 低优先级

bit 6 **CMIP:** 比较器中断优先级位

1 = 高优先级

0 = 低优先级

bit 5-4 **未实现:** 读为 0

bit 3 **BCLIP:** 总线冲突中断优先级位

1 = 高优先级

0 = 低优先级

bit 2 **LVDIP:** 低电压检测中断优先级位

1 = 高优先级

0 = 低优先级

bit 1 **TMR3IP:** TMR3 溢出中断优先级位

1 = 高优先级

0 = 低优先级

bit 0 **未实现:** 读为 0

# PIC18F85J11 系列

寄存器 10-12:      IPR3: 外设中断优先级寄存器 3

U-0	U-0	R-1	R-1	U-0	R/W-1	R/W-1	U-0
—	—	RC2IP	TX2IP	—	CCP2IP	CCP1IP	—
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	<b>RC2IP:</b> AUSART 接收中断优先级位
	1 = 高优先级
	0 = 低优先级
bit 4	<b>TX2IP:</b> AUSART 发送中断优先级位
	1 = 高优先级
	0 = 低优先级
bit 3	未实现: 读为 0
bit 2	<b>CCP2IP:</b> CCP2 中断优先级位
	1 = 高优先级
	0 = 低优先级
bit 1	<b>CCP1IP:</b> CCP1 中断优先级位
	1 = 高优先级
	0 = 低优先级
bit 0	未实现: 读为 0

## 10.5 RCON 寄存器

RCON 寄存器中包含的位用来确定器件上次复位或从空闲或休眠模式唤醒的原因。RCON 还包含一个可允许中断优先级的位（IPEN）。

寄存器 10-13: RCON: 复位控制寄存器

R/W-0	U-0	R/W-1	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	$\overline{\text{CM}}$	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
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      **IPEN:** 中断优先级允许位  
             1 = 允许中断优先级  
             0 = 禁止中断优先级（PIC16CXXX 兼容模式）
- bit 6      **未实现:** 读为 0
- bit 5       **$\overline{\text{CM}}$ :** 配置不匹配标志位  
             欲知位操作的详细信息，请参见寄存器 5-1。
- bit 4       **$\overline{\text{RI}}$ :** RESET 指令标志位  
             欲知位操作的详细信息，请参见寄存器 5-1。
- bit 3       **$\overline{\text{TO}}$ :** 看门狗定时器超时标志位  
             欲知位操作的详细信息，请参见寄存器 5-1。
- bit 2       **$\overline{\text{PD}}$ :** 掉电检测标志位  
             欲知位操作的详细信息，请参见寄存器 5-1。
- bit 1       **$\overline{\text{POR}}$ :** 上电复位状态位  
             欲知位操作的详细信息，请参见寄存器 5-1。
- bit 0       **$\overline{\text{BOR}}$ :** 欠压复位状态位  
             欲知位操作的详细信息，请参见寄存器 5-1。

## 10.6 INTx 引脚中断

RB0/INT0、RB1/INT1、RB2/INT2 和 RB3/INT3 引脚上的外部中断都是边沿触发的。如果 INTCON2 寄存器中相应的 INTEDGx 位被置 1 (= 1)，则为上升沿触发；如果该位被清零，则为下降沿触发。当 RBx/INTx 引脚上出现一个有效边沿时，相应的标志位 INTxIF 被置 1。通过清零相应的允许位 INTxIE，可禁止该中断。在重新允许该中断前，必须在中断服务程序中先用软件将 INTxIF 标志位清零。

如果 INTxIE 位在进入功耗管理模式前被置 1，则所有的外部中断（INT0、INT1、INT2 和 INT3）均能把处理器从功耗管理模式唤醒。如果全局中断允许位 GIE 被置 1，则处理器将在唤醒之后跳转到中断向量处执行程序。

INT1、INT2 和 INT3 的中断优先级由中断优先级位 INT1IP（INTCON3<6>）、INT2IP（INTCON3<7>）和 INT3IP（INTCON2<1>）中的值决定。没有与 INT0 相关的优先级位。INT0 始终是一个高优先级的中断源。

## 10.7 TMR0 中断

在 8 位模式（默认模式）下，TMR0 寄存器的溢出（FFh → 00h）会使标志位 TMR0IF 置 1。在 16 位模式下，TMR0H:TMR0L 寄存器对的溢出（FFFFh → 0000h）会使 TMR0IF 标志位置 1。通过将允许位 TMR0IE（INTCON<5>）置 1 或清零，可以允许或禁止该中断。Timer0 的中断优先级由中断优先级位 TMR0IP（INTCON2<2>）中的值决定。欲进一步了解 Timer0 模块的详细信息，请参见第 12.0 节“Timer0 模块”。

## 10.8 PORTB 电平变化中断

PORTB<7:4> 上的输入电平变化会将标志位 RBIF（INTCON<0>）置 1。通过将允许位 RBIE（INTCON<3>）置 1 或清零，可以允许或禁止该中断。PORTB 电平变化中断的优先级由中断优先级位 RBIP（INTCON2<0>）中的值决定。

## 10.9 中断的现场保护

在中断期间，PC 返回的地址被保存在堆栈中。另外，WREG、STATUS 和 BSR 寄存器的值被压入快速返回堆栈。如果未使用从中断快速返回功能（见第 6.3 节“数据存储器构成”），那么用户可能需要在进入中断服务程序前，保存 WREG、STATUS 和 BSR 寄存器的值。根据用户的具体应用，还可能需保存其他寄存器的值。例 10-1 在执行中断服务程序期间，保存并恢复 WREG、STATUS 和 BSR 寄存器的值。

**例 10-1:** 将 STATUS、WREG 和 BSR 寄存器的值保存在 RAM 中

MOVWF	W_TEMP	; W_TEMP is in virtual bank
MOVFF	STATUS, STATUS_TEMP	; STATUS_TEMP located anywhere
MOVFF	BSR, BSR_TEMP	; BSR_TMEP located anywhere
;		
; USER ISR CODE		
;		
MOVFF	BSR_TEMP, BSR	; Restore BSR
MOVF	W_TEMP, W	; Restore WREG
MOVFF	STATUS_TEMP, STATUS	; Restore STATUS



## 11.0 I/O 端口

根据选定的器件和使能的功能，最多有 9 个端口可供使用。I/O 端口的一些引脚与器件上外设功能复用。通常，当外设使能时，其对应的引脚就无法作为通用 I/O 引脚。

每个端口都有 3 个存储器映射的寄存器：

- TRIS 寄存器（数据方向寄存器）
- PORT 寄存器（读取器件引脚的电平）
- LAT 寄存器（输出锁存器）

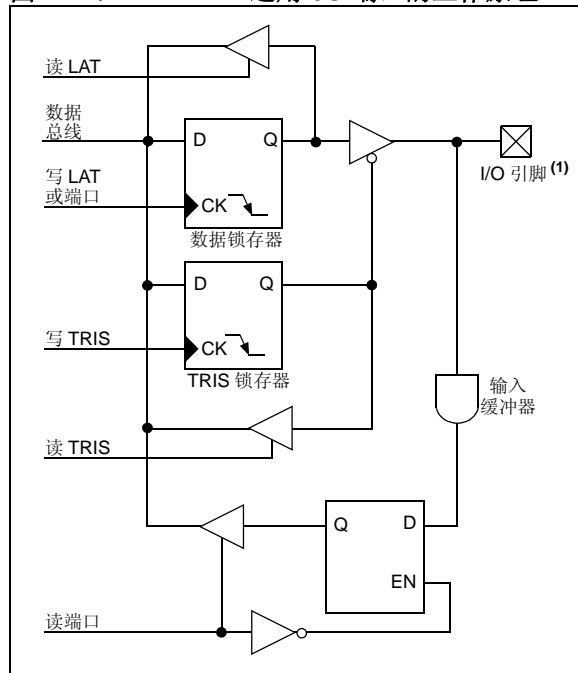
读 PORT 寄存器读入的是引脚的当前状态，而写 PORT 寄存器则是将数据写入输出锁存（LAT）寄存器。

将 TRIS 位置 1（= 1）可以将相应的 PORT 引脚配置为输入引脚（即，将对应的输出驱动器置于高阻态）。将 TRIS 位清零（= 0）可以将相应的 PORT 引脚配置为输出引脚（即，将对应 LAT 位的内容从选定引脚输出）。

在对 I/O 引脚驱动的值进行读 - 修改 - 写操作时会用到输出锁存器（LAT 寄存器）。对 LAT 寄存器执行读 - 修改 - 写操作将读 / 写 PORT 寄存器的锁存输出值。

图 11-1 给出了通用 I/O 端口的简化模型，图中没有显示到其他外设的接口。

图 11-1：通用 I/O 端口的工作原理



## 11.1 I/O 端口引脚的驱动能力

在开发应用时，必须考虑端口引脚的驱动能力。某些引脚上的输出驱动能力要高于其他引脚。同样，某些引脚可以接受高于 VDD 的输入电平。

### 11.1.1 输入引脚和电压注意事项

用作器件输入的引脚所能承受的电压取决于该引脚的输入功能。大部分仅用作数字输入的引脚能够承受最高 5.5V 的直流电压，这个电压值是数字逻辑电路的典型电压值。不能承受超过 VDD 电压的数字引脚有 RE0、RE1、RE2、RG0、RG2 和 RG3。

而还具有各种类型模拟输入功能的引脚只可以承受最高为 VDD 的电压值。应避免在这些引脚上施加超过 VDD 的电压。

表 11-1 总结了引脚的输入电压能力。更多详细信息，请参见第 26.0 节“电气特性”。

表 11-1：输入电压承受能力

端口或引脚	可承受的最高输入电压	说明
PORTA<7:5>	VDD	只能承受 VDD 输入电压。
PORTA<3:0>		
PORTC<1:0>		
PORTE<2:0>		
PORTF<7:1>		
PORTG<3:2,0>		
PORTA<4>	5.5V	可承受高于 VDD 的输入电压，适用于大多数标准逻辑。
PORTB<7:0>		
PORTC<7:2>		
PORTD<7:0>		
PORTE<7:3>		
PORTG<4,1>		
PORTH<7:0>(1)		
PORTJ<7:0>(1)		

注 1：这些端口在 64 引脚器件上不可用。

### 11.1.2 引脚输出驱动能力

作为数字 I/O 使用时，各引脚组的输出引脚驱动能力是不同的，这样可以满足不同应用的需求。通常，输出引脚按驱动能力分为三类。

PORTB、PORTC 以及 PORTA<7:6> 用来驱动电流较高的负载（如 LED）。PORTD、PORTE 和 PORTJ 可驱动与外部存储器件相关的数字电路。它们也可驱动 LED，但只适用于那些电流需求较小的 LED。PORTF、PORTG、PORTH 以及 PORTA<5:0> 的驱动电平最低，但可驱动具有高输入阻抗的正常数字电路负载。

# PIC18F85J11 系列

表 11-2 总结了端口的输出能力。更多详细信息请参见第 26.0 节“电气特性”中的“绝对最大值”。

表 11-2: 各端口的输出驱动电压

低	中	高
PORTA<5:0>	PORTD	PORTA<7:6>
PORTF	PORTE	PORTB
PORTG	PORTJ <sup>(1)</sup>	PORTC
PORTH <sup>(1)</sup>		

注 1: 这些端口在 64 引脚器件上不可用。

## 11.1.3 上拉配置

4 个 I/O 端口 (PORTB、PORTD、PORTE 和 PORTJ) 在所有引脚上实现可配置的弱上拉电路。内部上拉电路可使悬空的数字输入引脚在不使用外部电阻的情况下被拉至一个稳定电平。

用单个位使能以下各端口的上拉电路: 对于 PORTB, 是 RBPU (INTCON2<7>) 位, 对于其他端口, 分别是 RDPu、REPU 和 RJPU (PORTG<7:5>)。

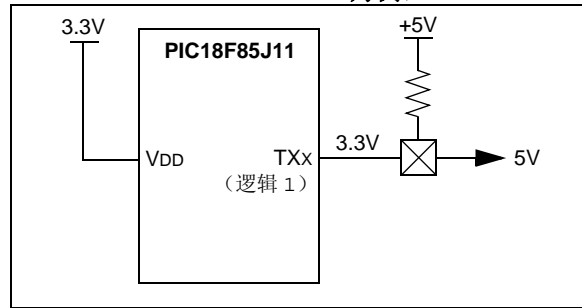
## 11.1.4 漏极开路输出

某些外设的输出引脚还配备了可配置的漏极开路输出选项。这使得外设可与工作在较高电平下的外部数字逻辑进行通信, 而无需电平转换器。

可在某些端口引脚 (尤其是与 USART、MSSP 模块 (SPI 模式下), 以及 CCP 模块的数据和时钟输出相关的端口引脚) 上实现漏极开路选项。将 TRISG 和 LATG 中对应模块的漏极开路控制位置 1, 可有选择地使能此选项。将在关于 PORTC、PORTE 和 PORTG 的一节中详细讨论它们的配置。

需要漏极开路选项时, 输出引脚必须通过由用户提供的外部上拉电阻连接到最高为 5V 的电平 (图 11-2)。当输出数字逻辑高电平信号时, 它被拉高至此高电平。

图 11-2: 使用漏极开路输出 (以 USART 为例)



## 11.2 PORTA、TRISA 和 LATA 寄存器

PORTA 是 8 位宽的双向端口。相应的数据方向寄存器和输出锁存器分别为 TRISA 和 LATA。

RA4/T0CKI 引脚为施密特触发器输入。PORTA 的所有其他引脚都具有 TTL 输入电平和完整的 CMOS 输出驱动器。

RA4 引脚与 Timer0 模块的时钟输入复用。RA5 和 RA<3:0> 与 A/D 转换器的模拟输入复用。

通过清零或置 1 ADCON1 寄存器中的 PCFG<3:0> 控制位, 可选择将模拟输入作为 A/D 转换器的输入通道。即使在 PORTA 引脚被用作模拟输入时, 相应的 TRISA 位仍控制 PORTA 引脚的方向。在将它们用作模拟输入时, 用户必须确保 TRISA 寄存器中相应的位保持为置 1 状态。

**注:** 在任何复位时, RA5 和 RA<3:0> 均被配置为模拟输入且读为 0。RA4 被配置为数字输入通道。

RA6/OSC2/CLKO 和 RA7/OSC1/CLKI 通常用作外部 (主) 振荡器电路 (HS 振荡器模式) 或外部时钟输入和输出 (EC 振荡器模式) 的外部电路连接端。在这些情况下, RA6 和 RA7 不可用作数字 I/O 并且它们对应的 TRIS 和 LAT 位读为 0。当器件配置为使用 INTOSC 或 INTRC 作为默认振荡器模式 (FOSC2 配置位为 0) 时, RA6 和 RA7 会被自动配置为数字 I/O, 并且振荡器和时钟输入或输出功能被禁止。

例 11-1: 初始化 PORTA

```
CLRF    PORTA    ; Initialize PORTA by
                  ; clearing output latches
CLRF    LATA      ; Alternate method to
                  ; clear output data latches
MOVLW   07h      ; Configure A/D
MOVWF   ADCON1   ; for digital inputs
MOVLW   0Bfh     ; Value used to initialize
                  ; data direction
MOVWF   TRISA    ; Set RA<7, 5:0> as inputs,
                  ; RA<6> as output
```

表 11-3: PORTA 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RA0/AN0	RA0	0	O	DIG	LATA<0> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTA<0> 数据输入；使能模拟输入时被禁止。
	AN0	1	I	ANA	A/D 输入通道 0。POR 时默认配置为输入；不影响数字输出。
RA1/AN1	RA1	0	O	DIG	LATA<1> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTA<1> 数据输入；使能模拟输入时被禁止。
	AN1	1	I	ANA	A/D 输入通道 1。POR 时默认配置为输入；不影响数字输出。
RA2/AN2/VREF-	RA2	0	O	DIG	LATA<2> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTA<2> 数据输入。当使能模拟功能时被禁止。
	AN2	1	I	ANA	A/D 输入通道 2。POR 时默认配置为输入。
	VREF-	1	I	ANA	A/D 和比较器低参考电压输入。
RA3/AN3/VREF+	RA3	0	O	DIG	LATA<3> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTA<3> 数据输入；使能模拟输入时被禁止。
	AN3	1	I	ANA	A/D 输入通道 3。POR 时默认配置为输入。
	VREF+	1	I	ANA	A/D 和比较器高参考电压输入。
RA4/T0CKI	RA4	0	O	DIG	LATA<4> 数据输出。
		1	I	ST	PORTA<4> 数据输入；POR 时为默认配置。
	T0CKI	x	I	ST	Timer0 时钟输入。
RA5/AN4	RA5	0	O	DIG	LATA<5> 数据输出；不受模拟输入影响。
		1	I	TTL	PORTA<5> 数据输入；使能模拟输入时被禁止。
	AN4	1	I	ANA	A/D 输入通道 4。POR 时为默认配置。
RA6/OSC2/CLKO	RA6	0	O	DIG	LATA<6> 数据输出；FOSC2 配置位置 1 时被禁止。
		1	I	TTL	PORTA<6> 数据输入；FOSC2 配置位置 1 时被禁止。
	OSC2	x	O	ANA	主振荡器反馈输出连接（HS 和 HSPLL 模式）。
	CLKO	x	O	DIG	系统周期时钟输出，Fosc/4（EC 和 ECPLL 模式）。
RA7/OSC1/CLKI	RA7	0	O	DIG	LATA<7> 数据输出；FOSC2 配置位置 1 时被禁止。
		1	I	TTL	PORTA<7> 数据输入；FOSC2 配置位置 1 时被禁止。
	OSC1	x	I	ANA	主振荡器输入连接（HS 和 HSPLL 模式）。
	CLKI	x	I	ANA	主外部时钟源输入（EC 和 ECPLL 模式）。

图注： O = 输出，I = 输入，ANA = 模拟信号，DIG = 数字输出，ST = 施密特缓冲器输入，TTL = TTL 缓冲器输入，x = 任意值（TRIS 位不影响端口方向或在此可被忽略）。

表 11-4: 与 PORTA 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
PORTA	RA7 <sup>(1)</sup>	RA6 <sup>(1)</sup>	RA5	RA4	RA3	RA2	RA1	RA0	60
LATA	LATA7 <sup>(1)</sup>	LATA6 <sup>(1)</sup>	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	60
TRISA	TRISA7 <sup>(1)</sup>	TRISA6 <sup>(1)</sup>	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	60
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	59

图注： — = 未实现，读为 0，x = 任意值。PORTA 不使用阴影单元。

注 1： 根据所选择的振荡器模式使能这些位。若未将这些位配置为 PORTA 引脚，则它们将被禁止且读为 x。

# PIC18F85J11 系列

## 11.3 PORTB、TRISB 和 LATB 寄存器

PORTB 是 8 位宽的双向端口。相应的数据方向寄存器和输出锁存器分别是 TRISB 和 LATB。PORTB 上所有引脚都只能作为数字引脚，最多可承受 5.5V 的电压。

PORTB 的每个引脚都有内部弱上拉电路。单个控制位可以接通所有上拉电路。通过将 RBPU (INTCON2<7>) 位清零可实现此功能。当 PORTB 端口的引脚配置为输出时，其弱上拉电路会自动切断。此弱上拉电路在上电复位时被禁止。

4 个 PORTB 引脚 (RB<7:4>) 具有电平变化中断功能。仅当将这些引脚配置为输入时，才具有此中断功能 (即：当 RB<7:4> 中的任一引脚被配置为输出时，该引脚就不再具有电平变化中断功能)。当前 RB<7:4> 输入引脚上的电平与 PORTB 上次读入锁存器的旧值进行比较。RB<7:4> 输出的“不匹配”值作逻辑或运算，以将标志位 RBIF (INTCON<0>) 置 1，并产生 RB 端口电平变化中断。

该中断可将器件从功耗管理模式中唤醒。在中断服务程序中，用户可通过以下方式清除该中断：

- 读或写 PORTB (MOVFF (ANY), PORTB 指令除外)。这将结束不匹配的情况。
- 等待一个指令周期 (例如执行一条 NOP 指令)。
- 清零标志位 RBIF。

电平不匹配的状态将会持续地将 RBIF 标志位置 1。而读 PORTB 将结束不匹配状态，并且允许在一个 Tcy 延迟后将 RBIF 标志位清零。

建议使用电平变化中断功能实现按键唤醒操作以及其他 PORTB 仅用作该中断功能的操作。在使用电平变化中断功能时，建议不要查询 PORTB 的状态。

### 例 11-2: 初始化 PORTB

```
CLRF    PORTB    ; Initialize PORTB by
                ; clearing output
                ; data latches
CLRF    LATB      ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh     ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISB     ; Set RB<3:0> as inputs
                ; RB<5:4> as outputs
                ; RB<7:6> as inputs
```

表 11-5: PORTB 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RB0/INT0	RB0	0	O	DIG	LATB<0> 数据输出。
		1	I	TTL	PORTB<0> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。
	INT0	1	I	ST	外部中断 0 输入。
RB1/INT1	RB1	0	O	DIG	LATB<1> 数据输出。
		1	I	TTL	PORTB<1> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。
	INT1	1	I	ST	外部中断 1 输入。
RB2/INT2	RB2	0	O	DIG	LATB<2> 数据输出。
		1	I	TTL	PORTB<2> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。
	INT2	1	I	ST	外部中断 2 输入。
RB3/INT3/ CCP2	RB3	0	O	DIG	LATB<3> 数据输出。
		1	I	TTL	PORTB<3> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。
	INT3	1	I	ST	外部中断 3 输入。
	CCP2 <sup>(1)</sup>	0	O	DIG	CCP2 比较输出和 CCP2 PWM 输出；优先级高于端口数据。
		1	I	ST	CCP2 捕捉输入。
RB4/KBI0	RB4	0	O	DIG	LATB<4> 数据输出。
		1	I	TTL	PORTB<4> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。
	KBI0		I	TTL	引脚电平变化中断。
RB5/KBI1	RB5	0	O	DIG	LATB<5> 数据输出。
		1	I	TTL	PORTB<5> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。
	KBI1		I	TTL	引脚电平变化中断。
RB6/KBI2/PGC	RB6	0	O	DIG	LATB<6> 数据输出。
		1	I	TTL	PORTB<6> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。
	KBI2	1	I	TTL	引脚电平变化中断。
	PGC	x	I	ST	供 ICSP 和 ICD 操作使用的串行执行（ICSP™）时钟输入。 <sup>(2)</sup>
RB7/KBI3/PGD	RB7	0	O	DIG	LATB<7> 数据输出。
		1	I	TTL	PORTB<7> 数据输入；当 $\overline{\text{RBPU}}$ 位被清零时启用弱上拉。
	KBI3	1	I	TTL	引脚电平变化中断。
	PGD	x	O	DIG	供 ICSP 和 ICD 操作使用的串行执行数据输出。 <sup>(2)</sup>
		x	I	ST	供 ICSP 和 ICD 操作使用的串行执行数据输入。 <sup>(2)</sup>

图注： O = 输出，I = 输入，DIG = 数字输出，ST = 施密特缓冲器输入，TTL = TTL 缓冲器输入，x = 任意值（TRIS 位不影响端口方向或在此可被忽略）。

注 1： 当 CCP2MX 配置位被清零时，CCP2 的备用分配（仅限扩展单片机模式下的 80 引脚器件）。默认分配为 RC1。

2： 当使能 ICSP™ 或 ICD 时，所有其他引脚功能都被禁止。

表 11-6: 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	复位值所在页
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	60
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	60
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	60
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	57
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	57

图注： PORTB 不使用阴影单元。

## 11.4 PORTC、TRISC 和 LATC 寄存器

PORTC 是 8 位宽的双向端口。相应的数据方向寄存器和输出锁存器分别是 TRISC 和 LATC。PORTC 引脚 RC2 到 RC7 只能作为数字引脚并且最高可承受 5.5V 的输入电压。

PORTC 与 CCP、MSSP 和 EUSART 外设功能复用（表 11-7）。这些引脚配有施密特触发器输入缓冲器。只要 CCP、SPI 和 EUSART 功能有效，则相关的引脚还可配置为漏极开路输出。通过将 SPIOD、CCPxOD 和 U1OD 控制位（分别为 TRISG<7:5> 和 LATG<6>）置 1 可选择漏极开路配置。

RC1 通常被配置为 CCP2 模块的默认外设引脚。配置位 CCP2MX（默认状态，CCP2MX = 1）控制 CCP2 的分配。

当使能外设功能时，应小心定义每个 PORTC 引脚的 TRIS 位。某些外设会改写 TRIS 位的设置而将引脚定义为输出，而另一些外设也会改写 TRIS 位的设置，但将引脚定义为输入。用户应参考相应的外设章节来正确设置 TRIS 位。

**注：** 在任何器件复位时，这些引脚都被配置为数字输入。

外设对引脚的改写会影响 TRISC 寄存器的内容。尽管某个外设可能会改写一个或多个引脚，但读 TRISC 总是会返回其当前内容。

### 例 11-3: 初始化 PORTC

```
CLRF    PORTC    ; Initialize PORTC by
                ; clearing output
                ; data latches
CLRF    LATC     ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh     ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISC    ; Set RC<3:0> as inputs
                ; RC<5:4> as outputs
                ; RC<7:6> as inputs
```

**表 11-7: PORTC 功能**

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RC0/T1OSO/ T13CKI	RC0	0	O	DIG	LATC<0> 数据输出。
		1	I	ST	PORTC<0> 数据输入。
	T1OSO	x	O	ANA	Timer1 振荡器输出；当使能 Timer1 振荡器时被使能。禁止数字 I/O。
	T13CKI	1	I	ST	Timer1/Timer3 计数器输入。
RC1/T1OSI/ CCP2	RC1	0	O	DIG	LATC<1> 数据输出。
		1	I	ST	PORTC<1> 数据输入。
	T1OSI	x	I	ANA	Timer1 振荡器输入；当使能 Timer1 振荡器时被使能。禁止数字 I/O。
	CCP2 <sup>(1)</sup>	0	O	DIG	CCP2 比较输出和 CCP2 PWM 输出；优先级高于端口数据。
		1	I	ST	CCP2 捕捉输入。
RC2/CCP1	RC2	0	O	DIG	LATC<2> 数据输出。
		1	I	ST	PORTC<2> 数据输入。
	CCP1	0	O	DIG	CCP1 比较输出和 CCP1 PWM 输出；优先级高于端口数据。
		1	I	ST	CCP1 捕捉输入。
RC3/SCK/SCL	RC3	0	O	DIG	LATC<3> 数据输出。
		1	I	ST	PORTC<3> 数据输入。
	SCK	0	O	DIG	SPI 时钟输出（MSSP 模块）；优先级高于端口数据。
		1	I	ST	SPI 时钟输入（MSSP 模块）。
	SCL	0	O	DIG	I <sup>2</sup> C™ 时钟输出（MSSP 模块）；优先级高于端口数据。
		1	I	I <sup>2</sup> C	I <sup>2</sup> C 时钟输入（MSSP 模块）；输入类型取决于模块设置。
RC4/SDI/SDA	RC4	0	O	DIG	LATC<4> 数据输出。
		1	I	ST	PORTC<4> 数据输入。
	SDI	0	O	ST	SPI 数据输入（MSSP 模块）。
	SDA	1	I	DIG	I <sup>2</sup> C 数据输出（MSSP 模块）；优先级高于端口数据。
		1	I	I <sup>2</sup> C	I <sup>2</sup> C 数据输入（MSSP 模块）；输入类型取决于模块设置。
RC5/SDO	RC5	1	O	DIG	LATC<5> 数据输出。
		1	I	ST	PORTC<5> 数据输入。
	SDO	0	O	DIG	SPI 数据输出（MSSP 模块）；优先级高于端口数据。
RC6/TX1/CK1	RC6	1	I	DIG	LATC<6> 数据输出。
		0	O	ST	PORTC<6> 数据输入。
	TX1	0	O	DIG	同步串行数据输出（EUSART 模块）；优先级高于端口数据。
	CK1	1	I	DIG	同步串行数据输入（EUSART 模块）。用户必须将其配置为输入。
		1	O	ST	同步串行时钟输入（EUSART 模块）。
RC7/RX1/DT1	RC7	0	O	DIG	LATC<7> 数据输出。
		1	I	ST	PORTC<7> 数据输入。
	RX1	1	I	ST	异步串行接收数据输入（EUSART 模块）。
	DT1	1	O	DIG	同步串行数据输出（EUSART 模块）；优先级高于端口数据。
		1	I	ST	同步串行数据输入（EUSART 模块）。用户必须将其配置为输入。

**图注：** O = 输出，I = 输入，ANA = 模拟信号，DIG = 数字输出，ST = 施密特缓冲器输入，x = 任意值（TRIS 位不影响端口方向或在此可被忽略）。

**注 1：** 当配置位 CCP2MX 被置 1 时，CCP2 的默认分配。

# PIC18F85J11 系列

表 11-8: 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	复位值 所在页
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	60
LATC	LATC7	LATBC6	LATC5	LATCB4	LATC3	LATC2	LATC1	LATC0	60
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	60
LATG	U2OD	U1OD	—	LATG4	LATG3	LATG2	LATG1	LATG0	60
TRISG	SPIOD	CCP2OD	CCP1OD	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	60

图注： — = 未实现，读为 0。PORTC 不使用阴影单元。



11.5 PORTD、TRISD 和 LATD 寄存器

PORTD 是 8 位宽的双向端口。相应的数据方向寄存器和输出锁存器分别是 TRISD 和 LATD。PORTD 上所有引脚都只能作为数字引脚，且最高可承受 5.5V 的电压。PORTD 上的所有引脚都配有施密特触发器输入缓冲器。每个引脚都可被单独地设置为输入或输出。

**注：** 在任何器件复位时，这些引脚都被配置为数字输入。

PORTD 的每个引脚都有内部弱上拉电路。单个控制位可以关闭所有上拉电路。这可通过清零 RDPU (PORTG<7>) 实现。当 PORTD 端口的引脚配置为输出时，其弱上拉电路会自动切断。在所有器件复位时上拉功能被禁止。

在 80 引脚器件上，PORTD 与系统总线复用作为外部存储器接口的一部分。I/O 端口和其他功能只有在通过将 EBDIS 位 (MEMCON<7>) 置 1 而禁止接口功能时才可用。当使能接口时，PORTD 是复用的地址 / 数据总线的低位字节 (AD<7:0>)。TRISD 位要被改写。

PORTD也可通过将PSPMODE控制位 (PSPCON<4>) 置 1 而配置成具有 8 位宽的并行主端口的数据功能。在这个模式中，并行端口数据的优先级高于其他数字 I/O (但外部存储器接口除外)。当并行端口有效时，输入缓冲器为 TTL。更多信息，请参见第 11.11 节 “并行从端口”。

例 11-4: 初始化 PORTD

CLRF	PORTD	; Initialize PORTD by ; clearing output ; data latches
CLRF	LATD	; Alternate method ; to clear output ; data latches
MOVLW	0CFh	; Value used to ; initialize data ; direction
MOVWF	TRISD	; Set RD<3:0> as inputs ; RD<5:4> as outputs ; RD<7:6> as inputs

# PIC18F85J11 系列

表 11-9: PORTD 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RD0/AD0/PSP0	RD0	0	O	DIG	LATD<0> 数据输出。
		1	I	ST	PORTD<0> 数据输入。
	AD0 <sup>(2)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 0 输出。 <sup>(1)</sup>
		x	I	TTL	外部存储器接口, 数据 bit 0 输入。 <sup>(1)</sup>
	PSP0		O	DIG	PSP 读数据输出 (LATD<0>); 优先级高于端口数据。
			I	TTL	PSP 写数据输入。
RD1/AD1/PSP1	RD1	0	O	DIG	LATD<1> 数据输出。
		1	I	ST	PORTD<1> 数据输入。
	AD1 <sup>(2)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 1 输出。 <sup>(1)</sup>
		x	I	TTL	外部存储器接口, 数据 bit 1 输入。 <sup>(1)</sup>
	PSP1	x	O	DIG	PSP 读数据输出 (LATD<1>); 优先级高于端口数据。
		x	I	TTL	PSP 写数据输入。
RD2/AD2/PSP2	RD2	0	O	DIG	LATD<2> 数据输出。
		1	I	ST	PORTD<2> 数据输入。
	AD2 <sup>(2)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 2 输出。 <sup>(1)</sup>
		x	I	TTL	外部存储器接口, 数据 bit 2 输入。 <sup>(1)</sup>
	PSP2	x	O	DIG	PSP 读数据输出 (LATD<2>); 优先级高于端口数据。
		x	I	TTL	PSP 写数据输入。
RD3/AD3/PSP3	RD3	0	O	DIG	LATD<3> 数据输出。
		1	I	ST	PORTD<3> 数据输入。
	AD3 <sup>(2)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 3 输出。 <sup>(1)</sup>
		x	I	TTL	外部存储器接口, 数据 bit 3 输入。 <sup>(1)</sup>
	PSP3	x	O	DIG	PSP 读数据输出 (LATD<3>); 优先级高于端口数据。
		x	I	TTL	PSP 写数据输入。
RD4/AD4/PSP4	RD4	0	O	DIG	LATD<4> 数据输出。
		1	I	ST	PORTD<4> 数据输入。
	AD4 <sup>(2)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 4 输出。 <sup>(1)</sup>
		x	I	TTL	外部存储器接口, 数据 bit 4 输入。 <sup>(1)</sup>
	PSP4	x	O	DIG	PSP 读数据输出 (LATD<4>); 优先级高于端口数据。
		x	I	TTL	PSP 写数据输入。
RD5/AD5/PSP5	RD5	0	O	DIG	LATD<5> 数据输出。
		1	I	ST	PORTD<5> 数据输入。
	AD5 <sup>(2)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 5 输出。 <sup>(1)</sup>
		x	I	TTL	外部存储器接口, 数据 bit 5 输入。 <sup>(1)</sup>
	PSP5	x	O	DIG	PSP 读数据输出 (LATD<5>); 优先级高于端口数据。
		x	I	TTL	PSP 写数据输入。
RD6/AD6/PSP6	RD6	0	O	DIG	LATD<6> 数据输出。
		1	I	ST	PORTD<6> 数据输入。
	AD6 <sup>(2)</sup>	x	O	DIG-3	外部存储器接口, 地址 / 数据 bit 6 输出。 <sup>(1)</sup>
		x	I	TTL	外部存储器接口, 数据 bit 6 输入。 <sup>(1)</sup>
	PSP6	x	O	DIG	PSP 读数据输出 (LATD<6>); 优先级高于端口数据。
		x	I	TTL	PSP 写数据输入。

图注: O = 输出, I = 输入, DIG = 数字输出, ST = 施密特缓冲器输入, TTL = TTL 缓冲器输入, x = 任意值 (TRIS 位不影响端口方向或在此可被忽略)。

注 1: 外部存储器接口 I/O 的优先级高于其他所有的数字和 PSP I/O。

2: 仅在 80 引脚器件上可用。

**表 11-9: PORTD 功能 (续)**

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RD7/AD7/PSP7	RD7	0	O	DIG	LATD<7> 数据输出。
		1	I	ST	PORTD<7> 数据输入。
	AD7 <sup>(2)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 7 输出。 <sup>(1)</sup>
		x	I	TTL	外部存储器接口, 数据 bit 7 输入。 <sup>(1)</sup>
	PSP7	x	O	DIG	PSP 读数据输出 (LATD<7>) ; 优先级高于端口数据。
		x	I	TTL	PSP 写数据输入。

**图注:** O = 输出, I = 输入, DIG = 数字输出, ST = 施密特缓冲器输入, TTL = TTL 缓冲器输入, x = 任意值 (TRIS 位不影响端口方向或在此可被忽略)。

**注 1:** 外部存储器接口 I/O 的优先级高于其他所有的数字和 PSP I/O。

**2:** 仅在 80 引脚器件上可用。

**表 11-10: 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	复位值 所在页
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	60
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	60
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	60
PORTG	RDPU	REPU	RJPU <sup>(1)</sup>	RG4	RG3	RG2	RG1	RG0	60

**图注:** PORTD 不使用阴影单元。

**注 1:** 在 64 引脚器件上未实现, 读为 0。

## 11.6 PORTE、TRISE 和 LATE 寄存器

PORTE 是 8 位宽的双向端口。相应的数据方向寄存器和输出锁存器分别是 TRISE 和 LATE。PORTE 上所有引脚都只能作为数字引脚，且最高可承受 5.5V 的电压。

PORTE 上的所有引脚都配有施密特触发器输入缓冲器。每个引脚都可被单独地设置为输入或输出。当 CCP2 在 RE7 引脚上有效时，也可将其配置为漏极开路输出。通过将 CCP2OD 控制位 (TRISG<6>) 置 1，可选择配置为漏极开路。

**注：** 在任何器件复位时，这些引脚都被配置为数字输入。

PORTE 的每个引脚都有内部弱上拉电路。单个控制位可以关闭所有上拉电路。这可通过清零 REPU (PORTG<6>) 实现。当 PORTE 端口的引脚配置为输出时，其弱上拉电路会自动切断。在任何器件复位时上拉功能被禁止。

在 80 引脚器件上，PORTE 与系统总线复用作为外部存储器接口的一部分。I/O 端口和其他功能只有在通过将 EBDIS 位 (MEMCON<7>) 置 1 而禁止接口功能时才可用。当使能接口时，PORTE 是复用的地址 / 数据总线的高位字节 (AD<15:8>)。TRISE 位也会被改写。

当 PORTD 的并行从端口有效时，PORTE 的三个引脚 (RE0、RE1 和 RE2) 被配置为该端口的数字控制输入。表 11-11 总结了控制功能。当将 PSPMODE 控制位 (PSPCON<4>) 置 1 时，自动进行重新配置。用户仍必须确保相应的 TRISE 位被置 1 以将这些引脚配置为数字输入。

RE7 也可配置成 CCP2 模块的备用外设引脚。这通过清零 CCP2MX 配置位完成。

### 例 11-5: 初始化 PORTE

```
CLRf    PORTE    ; Initialize PORTE by
                ; clearing output
                ; data latches
CLRf    LATE     ; Alternate method
                ; to clear output
                ; data latches
MOVLW   03h      ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISE    ; Set RE<1:0> as inputs
                ; RE<7:2> as outputs
```

表 11-11: PORTE 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RE0/RD $\overline{\text{AD8}}$	RE0	0	O	DIG	LATE<0> 数据输出。
		1	I	ST	PORTE<0> 数据输入。
	$\overline{\text{RD}}$	1	I	TTL	并行从端口读使能控制输入。
	AD8 <sup>(1)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 8 输出。(2)
		x	I	TTL	外部存储器接口, 数据 bit 8 输入。(2)
RE1/WR $\overline{\text{AD9}}$	RE1	0	O	DIG	LATE<1> 数据输出。
		1	I	ST	PORTE<1> 数据输入。
	$\overline{\text{WR}}$	1	I	TTL	并行从端口写使能控制输入。
	AD9 <sup>(1)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 9 输出。(2)
		x	I	TTL	外部存储器接口, 数据 bit 9 输入。(2)
RE2/AD10/ $\overline{\text{CS}}$	RE2	0	O	DIG	LATE<2> 数据输出。
		1	I	ST	PORTE<2> 数据输入。
	AD10 <sup>(1)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 10 输出。(2)
		x	I	TTL	外部存储器接口, 数据 bit 10 输入。(2)
	$\overline{\text{CS}}$	1	I	TTL	并行从端口片选控制输入。
RE3/AD11	RE3	0	O	DIG	LATE<3> 数据输出。
		1	I	ST	PORTE<3> 数据输入。
	AD11 <sup>(1)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 11 输出。(2)
		x	I	TTL	外部存储器接口, 数据 bit 11 输入。(2)
RE4/AD12	RE4	0	O	DIG	LATE<4> 数据输出。
		1	I	ST	PORTE<4> 数据输入。
	AD12 <sup>(1)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 12 输出。(2)
		x	I	TTL	外部存储器接口, 数据 bit 12 输入。(2)
RE5/AD13	RE5	0	O	DIG	LATE<5> 数据输出。
		1	I	ST	PORTE<5> 数据输入。
	AD13 <sup>(1)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 13 输出。(2)
		x	I	TTL	外部存储器接口, 数据 bit 13 输入。(2)
RE6/AD14	RE6	0	O	DIG	LATE<6> 数据输出。
		1	I	ST	PORTE<6> 数据输入。
	AD14 <sup>(1)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 14 输出。(2)
		x	I	TTL	外部存储器接口, 数据 bit 14 输入。(2)
RE7/AD15/ CCP2	RE7	0	O	DIG	LATE<7> 数据输出。
		1	I	ST	PORTE<7> 数据输入。
	AD15 <sup>(1)</sup>	x	O	DIG	外部存储器接口, 地址 / 数据 bit 15 输出。(2)
		x	I	TTL	外部存储器接口, 数据 bit 15 输入。(2)
	CCP2 <sup>(3)</sup>	0	O	DIG	CCP2 比较输出和 CCP2 PWM 输出; 优先级高于端口数据。
		1	I	ST	CCP2 捕捉输入。

图注: O = 输出, I = 输入, DIG = 数字输出, ST = 施密特缓冲器输入, TTL = TTL 缓冲器输入, x = 任意值 (TRIS 位不影响端口方向或在此可被忽略)。

- 注 1: 仅在 80 引脚器件上可用。  
 2: 外部存储器接口 I/O 的优先级高于其他所有的数字和 PSP I/O。  
 3: 当 CCP2MX 配置位被清零时, CCP2 的备用配置 (适用于工作在单片机模式下的所有器件)。

# PIC18F85J11 系列

表 11-12: 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	复位值 所在页
PORTE	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	60
LATE	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	60
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	60
PORTG	RDPU	REPU	RJPU <sup>(1)</sup>	RG4	RG3	RG2	RG1	RG0	60
TRISG	SPIOD	CCP2OD	CCP1OD	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	60

图注: — = 未实现, 读为 0。PORTE 不使用阴影单元。

注 1: 在 64 引脚器件上未实现, 读为 0。

11.7 PORTF、LATF 和 TRISF 寄存器

PORTF 是 7 位宽的双向端口。相应的数据方向寄存器和输出锁存器分别是 TRISF 和 LATF。PORTF 上的所有引脚都配有施密特触发器输入缓冲器。每个引脚都可被单独地设置为输入或输出。

PORTF 与模拟外设功能复用。通过将 CMCON 寄存器中相应的位置 1 还可以将 RF1 到 RF6 引脚用作比较器输入或输出。要将 RF<6:3> 用作数字输入，也必须关闭比较器。

- 注

  - 1: 当器件复位时，引脚 RF<6:1> 都配置成模拟输入，并且读为 0。
  - 2: 要将 PORTF 配置为数字 I/O，应关闭比较器并将 ADCON1 置 1。

例 11-6: 初始化 PORTF

```
CLRF    PORTF    ; Initialize PORTF by
                  ; clearing output
                  ; data latches
CLRF    LATF      ; Alternate method
                  ; to clear output
                  ; data latches
MOVLW   07h      ;
MOVWF   CMCON     ; Turn off comparators
MOVLW   0Fh;
MOVWF   ADCON1    ; Set PORTF as digital I/O
MOVLW   0CEh     ; Value used to
                  ; initialize data
                  ; direction
MOVWF   TRISF     ; Set RF3:RF1 as inputs
                  ; RF5:RF4 as outputs
                  ; RF7:RF6 as inputs
```

# PIC18F85J11 系列

表 11-13: PORTF 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RF1/AN6/ C2OUT	RF1	0	O	DIG	LATF<1> 数据输出；不受模拟输入影响。
		1	I	ST	PORTF<1> 数据输入；使能模拟输入时被禁止。
	AN6	1	I	ANA	A/D 输入通道 6。POR 时为默认配置。
	C2OUT	0	O	DIG	比较器 2 输出；优先级高于端口数据。
RF2/AN7/ C1OUT	RF2	0	O	DIG	LATF<2> 数据输出；不受模拟输入影响。
		1	I	ST	PORTF<2> 数据输入；使能模拟输入时被禁止。
	AN7	1	I	ANA	A/D 输入通道 7。POR 时为默认配置。
	C1OUT	0	O	TTL	比较器 1 输出；优先级高于端口数据。
RF3/AN8	RF3	0	O	DIG	LATF<3> 数据输出；不受模拟输入影响。
		1	I	ST	PORTF<3> 数据输入；使能模拟输入时被禁止。
	AN8	1	I	ANA	A/D 输入通道 8 和比较器 C2+ 输入。发生 POR 时，默认配置为输入；不受模拟输出影响。
RF4/AN9	RF4	0	O	DIG	LATF<4> 数据输出；不受模拟输入影响。
		1	I	ST	PORTF<4> 数据输入；使能模拟输入时被禁止。
	AN9	1	I	ANA	A/D 输入通道 9 和比较器 C2- 输入。发生 POR 时，默认配置为输入；不影响数字输出。
RF5/AN10/ CVREF	RF5	0	O	DIG	LATF<5> 数据输出；不受模拟输入影响。当使能 CVREF 输出时被禁止。
		1	I	ST	PORTF<5> 数据输入；使能模拟输入时被禁止。当使能 CVREF 输出时被禁止。
	AN10	1	I	ANA	A/D 输入通道 10 和比较器 C1+ 输入。发生 POR 时，默认配置为输入。
	CVREF	x	O	ANA	比较器参考电压输出。使能此功能将禁止数字 I/O。
RF6/AN11	RF6	0	O	DIG	LATF<6> 数据输出；不受模拟输入影响。
		1	I	ST	PORTF<6> 数据输入；使能模拟输入时被禁止。
	AN11	1	I	ANA	A/D 输入通道 11 和比较器 C1- 输入。发生 POR 时，默认配置为输入；不影响数字输出。
RF7/AN5/SS	RF7	0	O	DIG	LATF<7> 数据输出。
		1	I	ST	PORTF<7> 数据输入。
	AN5	1	I	ANA	A/D 输入通道 5。POR 时为默认配置。
	SS	1	I	TTL	MSSP 模块的从选择输入。

图注： O = 输出，I = 输入，ANA = 模拟信号，DIG = 数字输出，ST = 施密特缓冲器输入，TTL = TTL 缓冲器输入，  
x = 任意值（TRIS 位不影响端口方向或者在此可被忽略）。

表 11-14: 与 PORTF 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	复位值 所在页
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	60
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	—	60
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	60
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	59
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	59
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	59

图注： — = 未实现，读为 0。PORTF 不使用阴影单元。



11.8 PORTG、TRISG 和 LATG 寄存器

PORTG 是 5 位宽的双向端口。相应的数据方向寄存器和输出锁存器分别是 TRISG 和 LATG。PORTG 上所有引脚都只能用作数字功能，最高可承受 5.5V 的电压。

当作为 I/O 进行操作时，所有 PORTG 引脚都有施密特触发器输入缓冲器。RG1 和 RG2 引脚与 AUSART 模块复用。当 AUSART 有效时，RG1 引脚也可配置为漏极开路输出。通过将 U2OD 控制位（LATG<7>）置 1，可选择配置为漏极开路。

当使能外设功能时，应小心定义每个 PORTG 引脚的 TRIS 位。某些外设会改写 TRIS 位的设置而将引脚定义为输出，而另一些外设也会改写 TRIS 位的设置，但将引脚定义为输入。用户应参考相应的外设章节来正确设置 TRIS 位。改写引脚的值不会被装入 TRIS 寄存器。这将允许对 TRIS 寄存器执行读 - 修改 - 写操作而无需担心外设改写其内容。

虽然此端口仅有 5 位宽，但仍实现了 PORTG<7:5> 位，从而控制与 PORTD、PORTE 和 PORTJ 相关的 I/O 端口上的弱上拉电路。将这些位清零将使能各个端口的弱上拉电路。在所有器件复位时上拉功能默认被禁止。

大部分相应的 TRISG 和 LATG 位实现为 CCP1、CCP2 以及 SPI（TRISG<7:5>）和 USART（LATG<7:6>）的漏极开路控制位。将这些位置 1 可将输出引脚配置为漏极开路操作的相应外设。LATG<5> 未实现。

例 11-7: 初始化 PORTG

```
CLRF    PORTG    ; Initialize PORTG by
                ; clearing output
                ; data latches
CLRF    LATG      ; Alternate method
                ; to clear output
                ; data latches
MOVLW   04h       ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISG     ; Set RG1:RG0 as outputs
                ; RG2 as input
                ; RG4:RG3 as inputs
```

# PIC18F85J11 系列

表 11-15: PORTG 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RG0	RG0	0	O	DIG	LATG<0> 数据输出。
		1	I	ST	PORTG<0> 数据输入。
RG1/TX2/CK2	R21	0	O	DIG	LATG<1> 数据输出。
		1	I	ST	PORTG<1> 数据输入。
	TX2	1	O	DIG	同步串行数据输出 (AUSART2 模块)；优先级高于端口数据。
	CK2	1	O	DIG	同步串行数据输入 (AUSART2 模块)。用户必须将其配置为输入。
		1	I	ST	同步串行时钟输入 (AUSART2 模块)。
RG2/RX2/DT2	RG2	0	O	DIG	LATG<2> 数据输出。
		1	I	ST	PORTG<2> 数据输入。
	RX2	1	I	ST	异步串行接收数据输入 (AUSART2 模块)。
	DT2	1	O	DIG	同步串行数据输出 (AUSART2 模块)；优先级高于端口数据。
		1	I	ST	同步串行数据输入 (AUSART2 模块)。用户必须将其配置为输入。
RG3	RG3	0	O	DIG	LATG<3> 数据输出。
		1	I	ST	PORTG<3> 数据输入。
RG4	RG4	0	O	DIG	LATG<4> 数据输出。
		1	I	ST	PORTG<4> 数据输入。

图注: O = 输出, I = 输入, DIG = 数字输出, ST = 施密特缓冲器输入, x = 任意值 (TRIS 位不影响端口方向或在此可被忽略)。

表 11-16: 与 PORTG 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	复位值 所在页
PORTG	RDPU	REPU	RJPU <sup>(1)</sup>	RG4	RG3	RG2	RG1	RG0	60
LATG	U2OD	U1OD	—	LATG4	LATG3	LATG2	LATG1	LATG0	60
TRISG	SPIOD	CCP2OD	CCP1OD	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	60

图注: — = 未实现, 读为 0。PORTG 不使用阴影单元。

注 1: 在 64 引脚器件上未实现, 读为 0。

## 11.9 PORTH、LATH 和 TRISH 寄存器

**注：** PORTH 只在 80 引脚器件上可用。

PORTH 是 8 位宽的双向 I/O 端口。相应的数据方向寄存器和输出锁存器分别是 TRISH 和 LATH。所有引脚都只能用作数字功能，最高可承受 5.5V 的电压。

PORTH 上的所有引脚都配有施密特触发器输入缓冲器。每个引脚都可被单独地设置为输入或输出。

当使能外部存储器接口时，PORTH 的 4 个引脚被用作该接口的高位地址线。接口的地址输出的优先级高于其他数字 I/O。相应的 TRISH 位也将被改写。

### 例 11-8: 初始化 PORTH

```
CLRF    PORTH    ; Initialize PORTH by
                ; clearing output
                ; data latches
CLRF    LATH      ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0Fh      ; Configure PORTH as
MOVWF   ADCON1    ; digital I/O
MOVLW   0CFh     ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISH     ; Set RH3:RH0 as inputs
                ; RH5:RH4 as outputs
                ; RH7:RH6 as inputs
```

**表 11-17: PORTH 功能**

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RH0/A16	RH0	0	O	DIG	LATH<0> 数据输出。
		1	I	ST	PORTH<0> 数据输入。
	A16	x	O	DIG	外部存储器接口，地址线 16。优先级高于端口数据。
RH1/A17	RH1	0	O	DIG	LATH<1> 数据输出。
		1	I	ST	PORTH<1> 数据输入。
	A17	x	O	DIG	外部存储器接口，地址线 17。优先级高于端口数据。
RH2/A18	RH2	0	O	DIG	LATH<2> 数据输出。
		1	I	ST	PORTH<2> 数据输入。
	A18	x	O	DIG	外部存储器接口，地址线 18。优先级高于端口数据。
RH3/A19	RH3	0	O	DIG	LATH<3> 数据输出。
		1	I	ST	PORTH<3> 数据输入。
	A19	x	O	DIG	外部存储器接口，地址线 19。优先级高于端口数据。
RH4	RH4	0	O	DIG	LATH<4> 数据输出。
		1	I	ST	PORTH<4> 数据输入。
RH5	RH5	0	O	DIG	LATH<5> 数据输出。
		1	I	ST	PORTH<5> 数据输入。
RH6	RH6	0	O	DIG	LATH<6> 数据输出。
		1	I	ST	PORTH<6> 数据输入。
RH7	RH7	0	O	DIG	LATH<7> 数据输出。
		1	I	ST	PORTH<7> 数据输入。

**图注：** O = 输出，I = 输入，DIG = 数字输出，ST = 施密特缓冲器输入，x = 任意值（TRIS 位不影响端口方向或在此可被忽略）。

**表 11-18: 与 PORTH 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	复位值 所在页
PORTH	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	60
LATH	LATH7	LATH6	LATH5	LATH4	LATH3	LATH2	LATH1	LATH0	60
TRISH	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	60

# PIC18F85J11 系列

## 11.10 PORTJ、TRISJ 和 LATJ 寄存器

**注：** PORTJ 只在 80 引脚器件上可用。

PORTJ 是 8 位宽的双向端口。相应的数据方向寄存器和输出锁存器分别是 TRISJ 和 LATJ。PORTJ 上所有引脚都只能用作数字功能，最高可承受 5.5V 的电压。

PORTJ 上的所有引脚都配有施密特触发器输入缓冲器。每个引脚都可被单独地设置为输入或输出。

**注：** 在任何器件复位时，这些引脚都被配置为数字输入。

PORTJ 的每个引脚都有内部弱上拉电路。提供上拉电路是为了保持上电时外部存储器接口上的输入电平处于已知的状态。单个控制位可以关闭所有上拉电路。这可通过清零 RJPU (PORTG<5>) 实现。当 PORTJ 端口的引脚配置为输出时，其弱上拉电路会自动切断。在任何器件复位时上拉功能都被禁止。

当使能外部存储器接口时，所有的 PORTJ 引脚都用于接口的控制输出。当通过清零 EBDIS 控制位 (MEMCON<7>) 而使能此接口时，这将自动发生。TRISJ 位也会被改写。

### 例 11-9: 初始化 PORTJ

```
CLRF    PORTJ    ; Initialize PORTJ by
                  ; clearing output latches
CLRF    LATJ      ; Alternate method
                  ; to clear output latches
MOVLW   0CFh     ; Value used to
                  ; initialize data
                  ; direction
MOVWF   TRISJ     ; Set RJ3:RJ0 as inputs
                  ; RJ5:RJ4 as output
                  ; RJ7:RJ6 as inputs
```

表 11-19: PORTJ 功能

引脚名称	功能	TRIS 设置	I/O	I/O 类型	说明
RJ0/ALE	RJ0	0	O	DIG	LATJ<0> 数据输出。
		1	I	ST	PORTJ<0> 数据输入。
	ALE	x	O	DIG	外部存储器接口地址锁存器使能控制输出；优先级高于数字 I/O。
RJ1/OE	RJ1	0	O	DIG	LATJ<1> 数据输出。
		1	I	ST	PORTJ<1> 数据输入。
	OE	x	O	DIG	外部存储器接口输出使能控制输出；优先级高于数字 I/O。
RJ2/WRL	RJ2	0	O	DIG	LATJ<2> 数据输出。
		1	I	ST	PORTJ<2> 数据输入。
	WRL	x	O	DIG	外部存储器总线低字节写控制输出；优先级高于数字 I/O。
RJ3/WRH	RJ3	0	O	DIG	LATJ<3> 数据输出。
		1	I	ST	PORTJ<3> 数据输入。
	WRH	x	O	DIG	外部存储器接口高字节写控制输出；优先级高于数字 I/O。
RJ4/BA0	RJ4	0	O	DIG	LATJ<4> 数据输出。
		1	I	ST	PORTJ<4> 数据输入。
	BA0	x	O	DIG	外部存储器接口字节地址 0 控制输出；优先级高于数字 I/O。
RJ5/CE	RJ5	0	O	DIG	LATJ<5> 数据输出。
		1	I	ST	PORTJ<5> 数据输入。
	CE	x	O	DIG	外部存储器接口芯片使能控制输出；优先级高于数字 I/O。
RJ6/LB	RJ6	0	O	DIG	LATJ<6> 数据输出。
		1	I	ST	PORTJ<6> 数据输入。
	LB	x	O	DIG	外部存储器接口低字节使能控制输出；优先级高于数字 I/O。
RJ7/UB	RJ7	0	O	DIG	LATJ<7> 数据输出。
		1	I	ST	PORTJ<7> 数据输入。
	UB	x	O	DIG	外部存储器接口高字节使能控制输出；优先级高于数字 I/O。

图注： O = 输出， I = 输入， DIG = 数字输出， ST = 施密特缓冲器输入， x = 任意值（TRIS 位不影响端口方向或在此可被忽略）。

表 11-20: 与 PORTJ 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	复位值所在页
PORTJ	RJ7	RJ6	RJ5	RJ4	RJ3	RJ2	RJ1	RJ0	60
LATJ	LATJ7	LATJ6	LATJ5	LATJ4	LATJ3	LATJ2	LATJ1	LATJ0	60
TRISJ	TRISJ7	TRISJ6	TRISJ5	TRISJ4	TRISJ3	TRISJ2	TRISJ1	TRISJ0	60
PORTG	RDPU	REPU	RJPU	RG4	RG3	RG2	RG1	RG0	60

图注： PORTJ 不使用阴影单元。

# PIC18F85J11 系列

## 11.11 并行从端口

当将控制位  $PSPMODE$  ( $PSPCON<4>$ ) 置 1 时,  $PORTD$  也可用作 8 位宽的并行从端口或微处理器端口。 $PORTD$  可从外部通过  $\overline{RD}$  控制输入引脚  $RE0/\overline{RD}$  和  $\overline{WR}$  控制输入引脚  $RE1/\overline{WR}$  进行异步读写。

**注:** 对于 80 引脚器件, 并行从端口仅在单片机模式下可用。

$PSP$  可与 8 位微处理器数据总线直接连接。外部微处理器可以将  $PORTD$  锁存器作为一个 8 位锁存器进行读或写。将  $PSPMODE$  位置 1 将使能端口引脚  $RE0/\overline{RD}$  为  $\overline{RD}$  输入、 $RE1/\overline{WR}$  为  $\overline{WR}$  输入和  $RE2/\overline{CS}$  为  $\overline{CS}$  (片选) 输入。用作此功能时, 必须将  $TRISE$  寄存器中相应的数据方向位 ( $TRISE<2:0>$ ) 配置为输入 (置 1)。

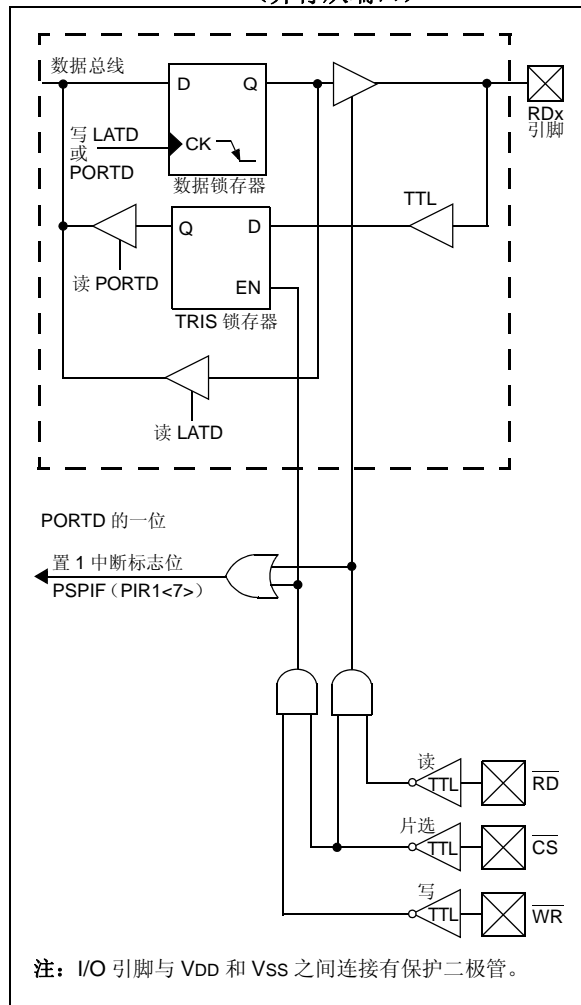
当第一次检测到  $\overline{CS}$  和  $\overline{WR}$  线均为低电平时, 开始写  $PSP$ ; 当检测到两者之一为高电平时, 写  $PSP$  结束。当写操作结束时,  $PSPIF$  和  $IBF$  标志位都被置 1。

当第一次检测到  $\overline{CS}$  和  $\overline{RD}$  线均为低电平时, 开始读  $PSP$ 。 $PORTD$  中的数据被读出且  $OBF$  位被置 1。如果用户将新数据写入  $PORTD$  从而将  $OBF$  置 1, 该数据将被立即读出, 但  $OBF$  位不会被置 1。

当检测到  $\overline{CS}$  或  $\overline{RD}$  线为高电平时,  $PORTD$  引脚返回到输入状态并且  $PSPIF$  位被置 1。用户应用程序应等到  $PSPIF$  被置 1 后再处理  $PSP$ 。当发生这种情况时, 可轮询  $IBF$  和  $OBF$  位并且采取适当的操作。

图 11-4 和图 11-5 分别给出了在读和写两种模式下的控制信号的时序。

图 11-3:  $PORTD$  和  $PORTE$  框图 (并行从端口)



寄存器 11-1: **PSPCON**: 并行从端口控制寄存器

R-0	R-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
IBF	OBF	IBOV	PSPMODE	—	—	—	—
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

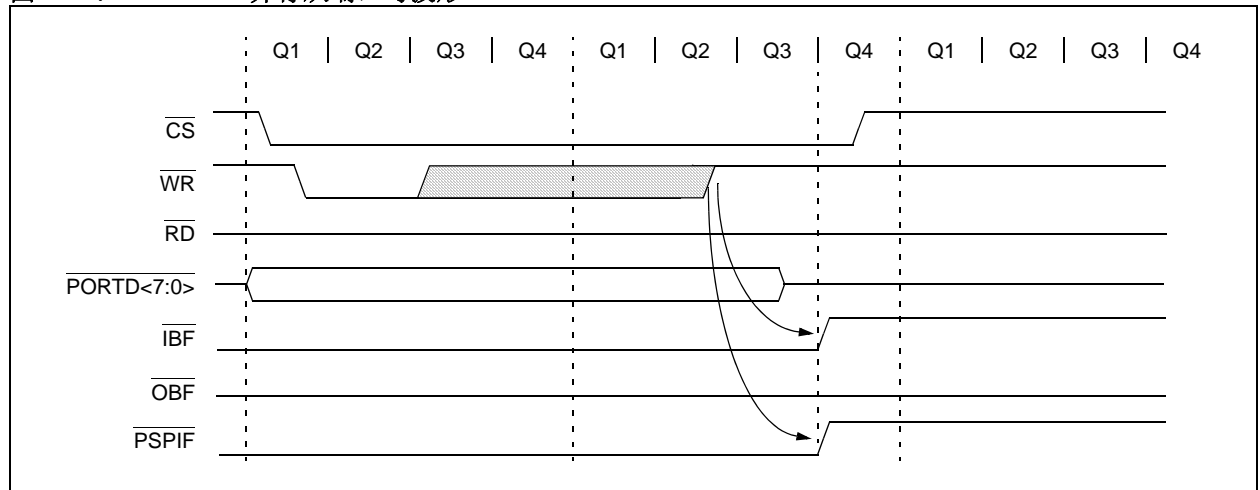
1 = 置 1

0 = 清零

x = 未知

- bit 7      **IBF**: 输入缓冲器满状态位  
 1 = 已接收到一个字, 等待 CPU 读取  
 0 = 未接收到任何字
- bit 6      **OBF**: 输出缓冲器满状态位  
 1 = 输出缓冲器仍保存着上一次写入的字  
 0 = 已读取输出缓冲器
- bit 5      **IBOV**: 输入缓冲器溢出检测位  
 1 = 在尚未读取上一次输入的字时, 又发生了一次写入 (必须由软件清零)  
 0 = 未发生溢出
- bit 4      **PSPMODE**: 并行从端口模式选择位  
 1 = 并行从端口模式  
 0 = 通用 I/O 模式
- bit 3-0    **未实现**: 读为 0

图 11-4: 并行从端口写波形



# PIC18F85J11 系列

图 11-5: 并行从端口读波形

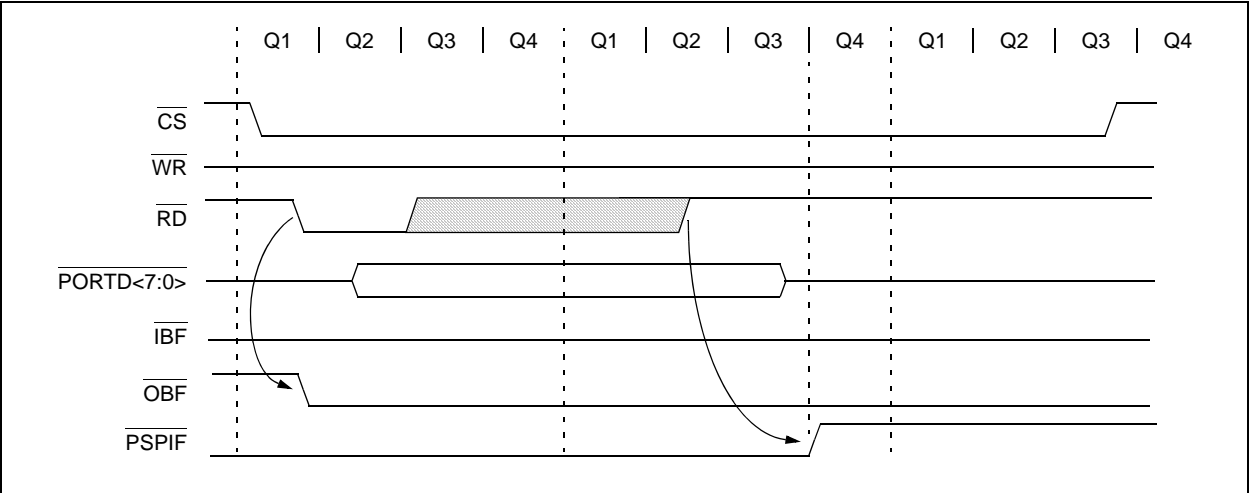


表 11-21: 与并行从端口相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	60
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	60
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	60
PORTE	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	60
LATE	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	60
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE1	60
PSPCON	IBF	OBF	IBOV	PSPMODE	—	—	—	—	59
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	—	TMR2IF	TMR1IF	59
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	—	TMR2IE	TMR1IE	59
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	—	TMR2IP	TMR1IP	59

图注: — = 未实现, 读为 0。并行从端口不使用阴影单元。



## 12.0 TIMER0 模块

Timer0 模块具有以下特征：

- 可通过软件选择作为 8 位或 16 位定时器 / 计数器
- 可读写的寄存器
- 专用的 8 位软件可编程预分频器
- 可选的时钟源（内部或外部）
- 外部时钟的边沿选择
- 溢出中断

T0CON 寄存器（寄存器 12-1）控制模块工作的所有方面，包括预分频比选择；该寄存器是可读写的。

图 12-1 给出了 8 位模式下 Timer0 模块的简化框图。

图 12-2 给出了 16 位模式下 Timer0 模块的简化框图。

**寄存器 12-1: T0CON: TIMER0 控制寄存器**

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
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 **TMR0ON:** Timer0 开关控制位

1 = 使能 Timer0

0 = 禁止 Timer0

bit 6 **T08BIT:** Timer0 8 位 /16 位模式控制位

1 = Timer0 被配置为 8 位定时器 / 计数器

0 = Timer0 被配置为 16 位定时器 / 计数器

bit 5 **T0CS:** Timer0 时钟源选择位

1 = T0CKI 引脚输入电平的跳变沿作为时钟源

0 = 内部时钟（Fosc/4）作为时钟源

bit 4 **T0SE:** Timer0 时钟源边沿选择位

1 = 在 T0CKI 引脚上电平的下降沿递增

0 = 在 T0CKI 引脚上电平的上升沿递增

bit 3 **PSA:** Timer0 预分频器分配位

1 = 未给 Timer0 分配预分频器。Timer0 时钟输入不经过预分频器。

0 = 已给 Timer0 分配了预分频器。Timer0 时钟输入信号来自预分频器的输出。

bit 2-0 **T0PS<2:0>:** Timer0 预分频值选择位

111 = 1:256 预分频值

110 = 1:128 预分频值

101 = 1:64 预分频值

100 = 1:32 预分频值

011 = 1:16 预分频值

010 = 1:8 预分频值

001 = 1:4 预分频值

000 = 1:2 预分频值

## 12.1 Timer0 工作原理

Timer0 既可用于作定时器也可用作计数器。具体的模式由 T0CS 位 (T0CON<5>) 选择。在定时器模式下 (T0CS = 0)，除非选择了不同的预分频值 (见第 12.3 节“预分频器”)，否则，默认情况下，在每个时钟周期该模块都会递增。如果写入 TMR0 寄存器，那么在随后的两个指令周期，计数将不再递增。用户可通过将调整值写入 TMR0 寄存器来解决上述问题。

通过将 T0CS 位置 1 (= 1) 选择计数器模式。在计数器模式下，Timer0 可在 RA4/T0CKI 引脚信号的每个上升沿或下降沿递增。触发递增的边沿由 Timer0 时钟源边沿选择位 T0SE (T0CON<4>) 决定。清零此位选择上升沿触发。下面讨论外部时钟输入的限制条件。

可以使用外部时钟源来驱动 Timer0。但是必须满足一定的要求以确保外部时钟与内部相位时钟 (Tosc) 同步。

在同步之后，仍需要一定的延时才会引发定时器 / 计数器的递增操作。

## 12.2 Timer0 的 16 位读写模式

TMR0H 不是 16 位模式 Timer0 的高字节。它实际上是 Timer0 高字节的缓冲寄存器，因为 Timer0 的高字节不可以被直接读写 (见图 12-2)。在读 TMR0L 时使用 Timer0 高字节的内容更新 TMR0H。这样可以一次读取 Timer0 的全部 16 位，而无需验证读到的高字节和低字节的有效性。(在高、低字节分两次连续读取的情况下，由于可能存在进位，因此需要验证读到字节的有效性)。

同样，写入 Timer0 的高字节也是通过 TMR0H 缓冲寄存器来操作的。在写入 TMR0L 的同时，使用 TMR0H 的内容更新 Timer0 的高字节。这样一次就可以完成 Timer0 全部 16 位的更新。

图 12-1: TIMER0 框图 (8 位模式)

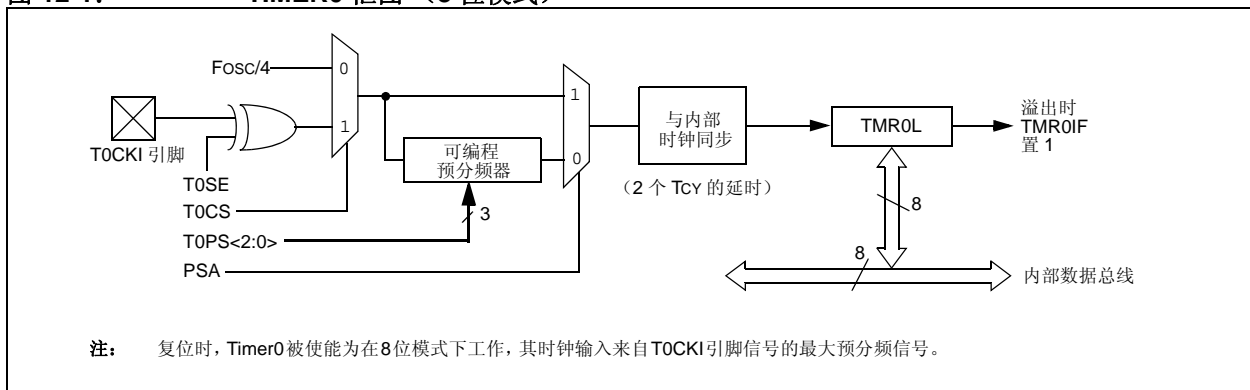
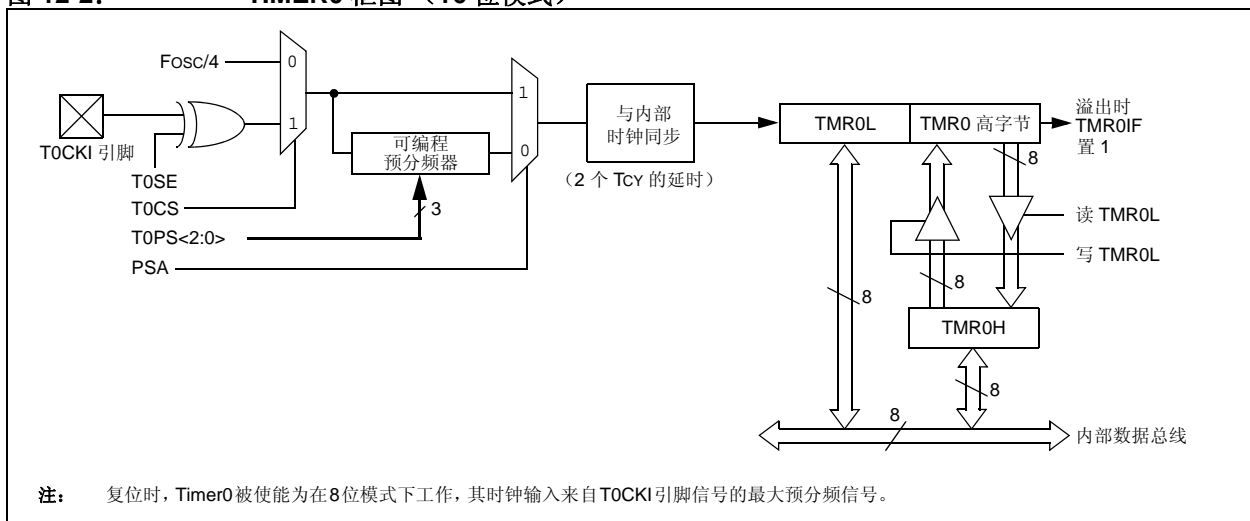


图 12-2: TIMER0 框图 (16 位模式)



## 12.3 预分频器

Timer0 模块的预分频器为一个 8 位计数器。此预分频器不可直接读写。其值通过 PSA 和 T0PS<2:0> 位 (T0CON<3:0>) 设置, 这两位决定预分频器的分配和预分频比值。

将 PSA 位清零可将预分频器分配给 Timer0 模块。在分配预分频器时, 预分频比值可以从 1:2 到 1:256 之间进行选择, 该比值以 2 的整数次幂递增。

若将预分频器分配给 Timer0 模块, 所有以 TMR0 寄存器为写入对象的指令 (如 CLRF TMR0、MOVWF TMR0 和 BSF TMR0 等) 都将使预分频器的计数值清零。

**注:** 若将预分频器分配给 Timer0, 写入 TMR0 会将预分频器的计数值清零, 但不会改变预分频器的分配。

### 12.3.1 切换预分频器的分配

预分频器的分配完全由软件控制, 并且在程序执行期间可以随时更改。

## 12.4 Timer0 中断

当 TMR0 寄存器发生溢出时 (8 位模式下, 从 FFh 到 00h; 或 16 位模式下, 从 FFFFh 到 0000h), 将产生 TMR0 中断。这种溢出会将标志位 TMR0IF 置 1。可以通过清零 TMR0IE 位 (INTCON<5>) 来屏蔽此中断。在重新允许该中断前, 必须在中断服务程序中用软件清零 TMR0IF 位。

由于 Timer0 在休眠模式下是关闭的, 所以 TMR0 中断无法将处理器从休眠状态唤醒。

**表 12-1: 与 TIMER0 相关的寄存器**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
TMR0L	Timer0 寄存器的低字节								58
TMR0H	Timer0 寄存器的高字节								58
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	58
TRISA	TRISA7 <sup>(1)</sup>	TRISA6 <sup>(1)</sup>	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	60

**图注:** — = 未实现, 读为 0。Timer0 不使用阴影单元。

**注 1:** 仅当选择内部振荡器作为默认时钟源 (FOSC2 配置位 = 0) 时, 才将 RA<7:6> 及其相关的锁存和方向位配置为端口引脚, 否则将禁止它们并且这些位读为 0。

# PIC18F85J11 系列

---

注:

## 13.0 TIMER1 模块

Timer1 定时器 / 计数器模块具有以下特征：

- 可通过软件选择作为 16 位定时器或计数器
- 可读写的 8 位寄存器（TMR1H 和 TMR1L）
- 可选择使用器件时钟或 Timer1 内部振荡器作为内部或外部时钟源
- 溢出中断
- CCPx 特殊事件触发复位
- 器件时钟状态标志位（T1RUN）

图 13-1 给出了 Timer1 模块的简化框图。图 13-2 所示为此模块在读 / 写模式下的工作原理框图。

该模块自身具有低功耗振荡器，可提供额外的时钟。Timer1 振荡器也可作为单片机处于功耗管理模式下的低功耗时钟源。

仅需增加极少量的外部元件和代码开销，Timer1 就可为应用提供实时时钟（RTC）。

Timer1 的工作由 T1CON 控制寄存器（寄存器 13-1）控制。该寄存器还包含 Timer1 振荡器使能位（T1OSCEN）。可以通过将控制位 TMR1ON（T1CON<0>）置 1 或清零来使能或禁止 Timer1。

寄存器 13-1: T1CON: TIMER1 控制寄存器

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON
bit 7							bit 0

图注：

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

- bit 7      **RD16:** 16 位读 / 写模式使能位  
 1 = 使能通过一次 16 位操作对 Timer1 寄存器进行读写  
 0 = 使能通过两次 8 位操作对 Timer1 寄存器进行读写
- bit 6      **T1RUN:** Timer1 系统时钟状态位  
 1 = 器件时钟由 Timer1 振荡器产生  
 0 = 器件时钟由另一个时钟源产生
- bit 5-4      **T1CKPS<1:0>:** Timer1 输入时钟预分频比选择位  
 11 = 1:8 预分频值  
 10 = 1:4 预分频值  
 01 = 1:2 预分频值  
 00 = 1:1 预分频值
- bit 3      **T1OSCEN:** Timer1 振荡器使能位  
 1 = 使能 Timer1 振荡器  
 0 = 关闭 Timer1 振荡器  
 关断振荡器的反相器和反馈电阻以降低功耗。
- bit 2       **$\overline{T1SYNC}$ :** Timer1 外部时钟输入同步选择位  
 当 TMR1CS = 1 时：  
 1 = 不与外部时钟输入同步  
 0 = 与外部时钟输入同步  
 当 TMR1CS = 0 时：  
 忽略此位。当 TMR1CS = 0 时，Timer1 使用内部时钟。
- bit 1      **TMR1CS:** Timer1 时钟源选择位  
 1 = 使用 RC0/T1OSO/T13CKI 引脚上的外部时钟（上升沿触发计数）  
 0 = 内部时钟（Fosc/4）
- bit 0      **TMR1ON:** Timer1 使能位  
 1 = 使能 Timer1  
 0 = 停止 Timer1

# PIC18F85J11 系列

## 13.1 Timer1 工作原理

Timer1 可工作在以下模式：

- 定时器
- 同步计数器
- 异步计数器

工作模式由时钟选择位 TMR1CS (T1CON<1>) 决定。当 TMR1CS 清零 (= 0) 时，Timer1 在每个内部指令周期 ( $F_{osc}/4$ ) 递增。当 TMR1CS 位

置 1 时，Timer1 在 Timer1 外部时钟输入信号或 Timer1 振荡器信号（如果使能）的每个上升沿递增。

当使能 Timer1 时，RC1/T1OSI 和 RC0/T1OSO/T13CKI 引脚变为输入引脚。这意味着 TRISC<1:0> 的值被忽略并且这些引脚的读取值为 0。

图 13-1: TIMER1 框图 (8 位模式)

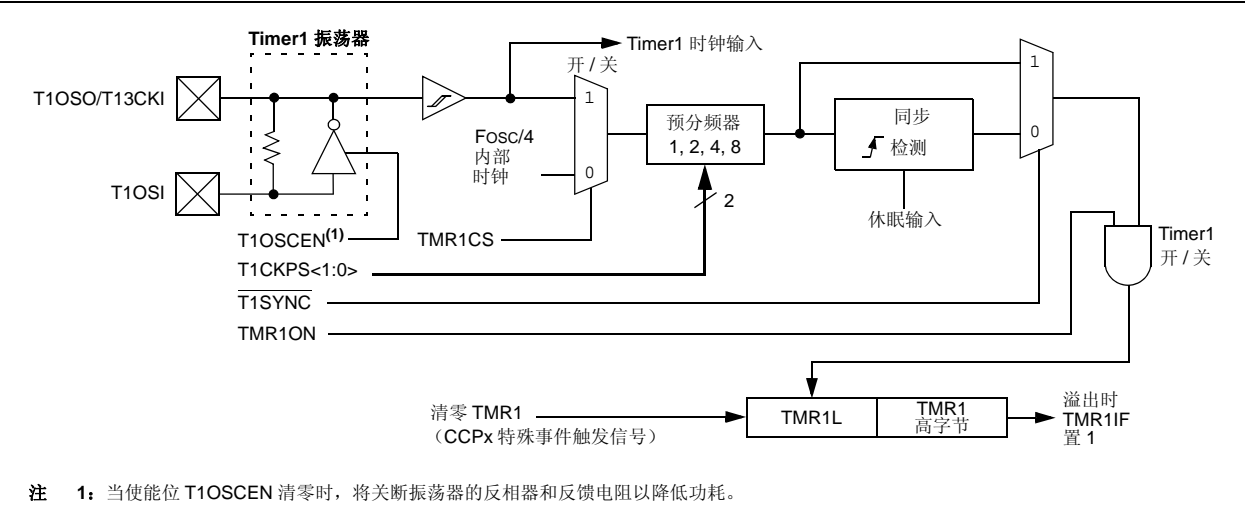
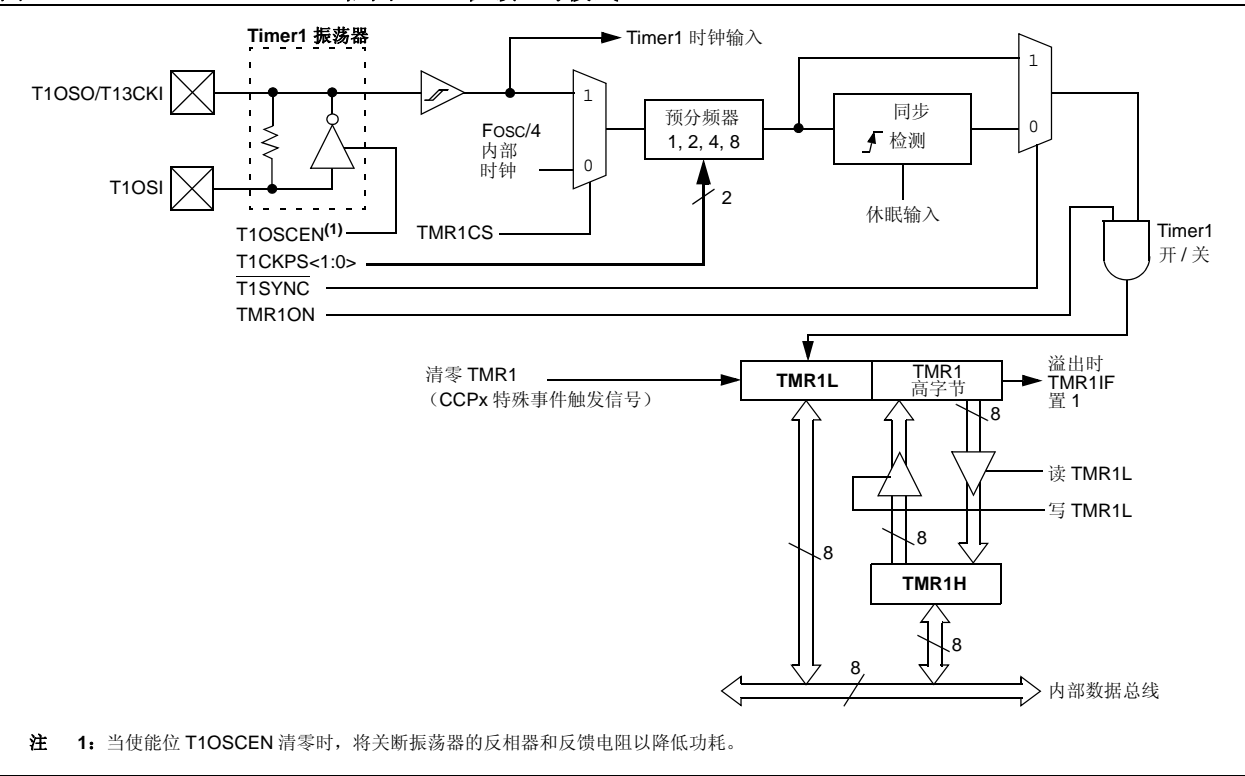


图 13-2: TIMER1 框图 (16 位读/写模式)



## 13.2 Timer1 16 位读 / 写模式

可将 Timer1 配置为 16 位读写模式（见图 13-2）。当 RD16 控制位（T1CON<7>）置 1 时，TMR1H 的地址被映射到 Timer1 的高字节缓冲寄存器。对 TMR1L 的读操作将把 Timer1 的高字节内容装入 Timer1 高字节缓冲寄存器。这种方式使用户可以精确地读取 Time1 的全部 16 位，而不需要像先读高字节再读低字节那样：由于两次读取之间可能存在进位，而不得不验证读取的有效性。

对 Timer1 的高字节进行写操作也必须通过 TMR1H 缓冲器进行。在写入 TMR1L 的同时，使用 TMR1H 的内容更新 Timer1 的高字节。这样允许用户将 16 位值一次写入 Timer1 的高字节和低字节。

在该模式下不能直接读写 Timer1 的高字节。所有读写都必须通过 Timer1 高字节缓冲寄存器进行。写入 TMR1H 不会清零 Timer1 预分频器。只有在写 TMR1L 时才会清零该预分频器。

## 13.3 Timer1 振荡器

片上晶振电路连接在 T1OSI（输入）引脚和 T1OSO（放大器输出）引脚之间。通过将 Timer1 振荡器使能位 T1OSCEN（T1CON<3>）置 1 可使能该振荡电路。此振荡电路是一种低功耗电路，它采用了额定振荡频率为 32 kHz 的晶振。在所有功耗管理模式下该振荡电路都将继续运行。图 13-3 所示为典型的 LP 振荡电路。表 13-1 给出了供 Timer1 振荡器选择的电容值。

用户必须提供软件延时来确保 Timer1 振荡器的正常起振。

表 13-1: TIMER1 振荡器的电容选择 (2,3,4)

振荡器类型	频率	C1	C2
LP	32.768 kHz	27 pF <sup>(1)</sup>	27 pF <sup>(1)</sup>

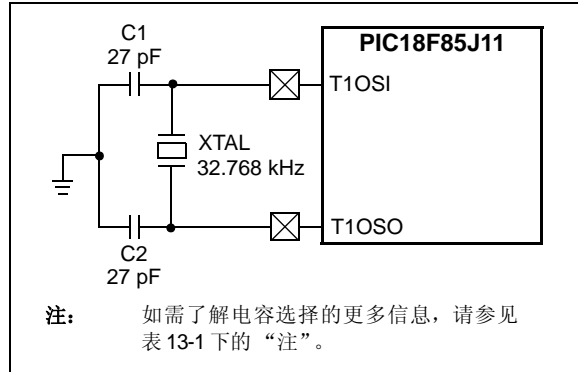
- 注
- 1: Microchip 建议在验证振荡电路时从这些值开始。
  - 2: 选用较大的电容值虽然可以提高振荡器的稳定性，但同时也会延长起振时间。
  - 3: 由于谐振器 / 晶振的特性各不相同，因此用户应当向谐振器 / 晶振制造厂商咨询外部元件相应的值。
  - 4: 上述电容值仅供设计参考。

### 13.3.1 使用 TIMER1 作为时钟源

在功耗管理模式中也可以将 Timer1 振荡器用作时钟源。通过将系统时钟选择位 SCS<1:0>（OSCCON<1:0>）设置为 01，器件可以切换到 SEC\_RUN 模式，此时 CPU 和外设都以 Timer1 振荡器作为时钟源。如果 IDLEN 位（OSCCON<7>）被清零并且执行了 SLEEP 指令，器件将进入 SEC\_IDLE 模式。欲知更多详情，请参见第 4.0 节“功耗管理模式”。

只要 Timer1 振荡器提供时钟源，Timer1 系统时钟状态标志位 T1RUN（T1CON<6>）就会置 1。这可用于确定控制器的当前时钟模式。该位也可指示故障保护时钟监视器当前正使用的时钟源。如果使能了故障保护时钟监视器并且 Timer1 振荡器在提供时钟信号时发生了故障，查询 T1RUN 位可以确定时钟源是 Timer1 振荡器还是其他时钟源。

图 13-3: TIMER1 LP 振荡器的外部元件



# PIC18F85J11 系列

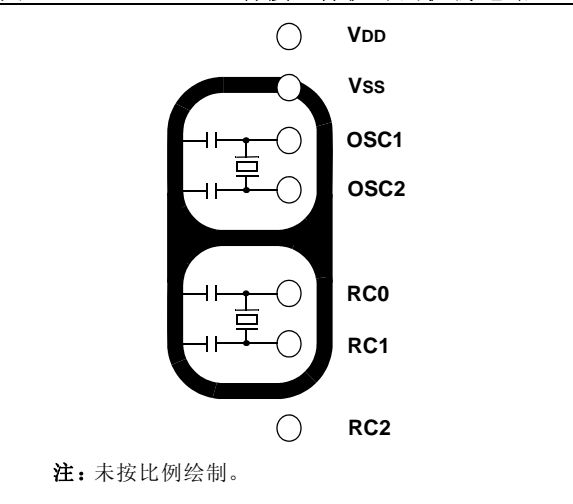
## 13.3.2 TIMER1 振荡器布线注意事项

Timer1 振荡电路在工作期间的功耗极小。鉴于此振荡器的低功耗特性，它对附近变化较快的信号可能会比较敏感。

如图 13-3 所示，振荡电路应该尽可能靠近单片机。除了 Vss 或 VDD 外，在振荡电路区域内不应有其他电路。

对于单面 PCB 板，如果必须要在该振荡器附近布置高速电路（如输出比较模式或 PWM 模式下的 CCP1 引脚，或使用 OSC2 引脚的主振荡器），那么在该振荡电路周围布置接地保护环（如图 13-4 所示），或外加接地层可能会有帮助。

图 13-4: 有接地保护环的振荡电路



## 13.4 Timer1 中断

TMR1 寄存器对（TMR1H:TMR1L）从 0000h 开始递增，一直到 FFFFh，然后溢出从 0000h 重新开始计数。如果允许 Timer1 中断，该中断就会在溢出时产生，并置 1 中断标志位 TMR1IF（PIR1<0>）。可以通过置 1 或清零 Timer1 中断允许位 TMR1IE（PIE1<0>）来允许或禁止该中断。

## 13.5 使用 CCPx 特殊事件触发信号来复位 Timer1

如果 CCP1 或 CCP2 被配置为使用 Timer1，并在比较模式下产生特殊事件触发信号（CCPxM<3:0> = 1011），触发信号将复位 Timer3。如果使能了 A/D 模块，来自 CCP2 的触发信号还将启动 A/D 转换（欲知更多信息，请参见第 16.3.4 节“特殊事件触发信号”）。

要使用这一功能，必须将 Timer1 模块配置为定时器或同步计数器。在这种情况下，CCPRxH:CCPRxL 寄存器对实际上变成了 Timer1 的周期寄存器。

如果 Timer1 工作在异步计数器模式下，复位操作将不起作用。

如果对 Timer1 的写操作和特殊事件触发信号同时发生，则写操作优先。

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

## 13.6 使用 Timer1 作为实时时钟

为 Timer1 外接一个 LP 振荡器（如第 13.3 节“Timer1 振荡器”中所述），用户就可以在其应用中包含 RTC 功能。这是通过一个提供精确时基的廉价时钟晶振以及几行计算时间的应用程序代码实现的。当器件工作在休眠模式下并使用电池或超大容量电容作为电源时，可以完全省去额外的 RTC 器件和备用电池。

应用代码程序 RTCisr（如例 13-1 所示）给出了使用中断服务程序以 1 秒的间隔递增计数器的简单方法。递增 TMR1 寄存器对的值直至溢出，将触发中断并调用中断服务程序，该程序会使秒计数器加 1。而分钟和小时计数器则会在前面的计数器溢出时加 1。

由于这对寄存器为 16 位宽，因此使用 32.768 kHz 时钟，将其计数到溢出需要 2 秒。要使溢出按所需的 1 秒间隔进行，必须预先装载这对寄存器，最简单的方法是使用 BSF 指令将 TMR1H 的最高有效位置 1。请注意绝不要预先加载或改变 TMR1L 寄存器，这样做可能会引起多个周期的累积错误。

要使此方法精确，Timer1 必须工作在异步模式且必须允许 Timer1 溢出中断（PIE1<0> = 1），如程序 RTCinit 所示。同时 Timer1 振荡器也必须使能并始终保持运行。



**例 13-1: 使用 TIMER1 中断服务程序实现实时时钟**

```

RTCinit
    MOVLW    80h           ; Preload TMR1 register pair
    MOVWF    TMR1H         ; for 1 second overflow
    CLRF     TMR1L
    MOVLW    b'00001111'   ; Configure for external clock,
    MOVWF    T1CON         ; Asynchronous operation, external oscillator
    CLRF     secs          ; Initialize timekeeping registers
    CLRF     mins          ;
    MOVLW    .12
    MOVWF    hours
    BSF      PIE1, TMR1IE   ; Enable Timer1 interrupt
    RETURN

RTCisr
    BSF      TMR1H, 7       ; Preload for 1 sec overflow
    BCF      PIR1, TMR1IF   ; Clear interrupt flag
    INCF     secs, F        ; Increment seconds
    MOVLW    .59            ; 60 seconds elapsed?
    CPFSGT   secs
    RETURN                                ; No, done
    CLRF     secs          ; Clear seconds
    INCF     mins, F        ; Increment minutes
    MOVLW    .59            ; 60 minutes elapsed?
    CPFSGT   mins
    RETURN                                ; No, done
    CLRF     mins          ; clear minutes
    INCF     hours, F       ; Increment hours
    MOVLW    .23            ; 24 hours elapsed?
    CPFSGT   hours
    RETURN                                ; No, done
    CLRF     hours          ; Reset hours
    RETURN                                ; Done
    
```

**表 13-2: TIMER1 作为定时器 / 计数器时相关的寄存器**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	—	TMR2IF	TMR1IF	59
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	—	TMR2IE	TMR1IE	59
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	—	TMR2IP	TMR1IP	59
TMR1L	Timer1 寄存器的低字节								58
TMR1H	Timer1 寄存器的高字节								58
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCN	T1SYNC	TMR1CS	TMR1ON	58

图注: Timer1 模块不使用阴影单元。

# PIC18F85J11 系列

---

注:

## 14.0 TIMER2 模块

Timer2 模块具有以下特征：

- 8 位定时器和周期寄存器（分别为 TMR2 和 PR2）
- 可读写（以上两个寄存器）
- 可软件编程的预分频器（分频比为 1:1、1:4 和 1:16）
- 可软件编程的后分频器（分频比为 1:1 至 1:16）
- 当 TMR2 与 PR2 匹配时产生中断
- 作为 MSSP 模块的可选移位时钟

此模块的工作由 T2CON 寄存器（寄存器 14-1）控制，此寄存器用于使能或禁止定时器并配置预分频器和后分频器。可以通过清零控制位 TMR2ON（T2CON<2>）关闭 Timer2，以使功耗最小。

图 14-1 所示为该模块的简化框图。

## 14.1 Timer2 工作原理

在正常操作中，TMR2 从 00h 开始，在每个时钟周期（Fosc/4）递增 1。4 位计数器 / 预分频器提供了对时钟输入不分频、4 分频和 16 分频三种预分频选项。可通过预分频控制位 T2CKPS<1:0>（T2CON<1:0>）进行选择。在每个时钟周期，TMR2 的值都会与周期寄存器 PR2 中的值进行比较。当两个值匹配时，由比较器产生匹配信号作为定时器的输出。此信号也会使 TMR2 的值在下一个周期复位到 00h，并驱动输出计数器 / 后分频器（见第 14.2 节“Timer2 中断”）。

TMR2 和 PR2 寄存器均可直接读写。在任何方式的器件复位时，TMR2 寄存器都会清零，而 PR2 寄存器则初始化为 FFh。发生以下事件时，预分频和后分频计数器均会清零：

- 对 TMR2 寄存器进行写操作
- 对 T2CON 寄存器进行写操作
- 任何方式的器件复位（上电复位、MCLR 复位、看门狗定时器复位或欠压复位）

写 T2CON 时 TMR2 不会清零。

寄存器 14-1: 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
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	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	<b>T2OUTPS&lt;3:0&gt;</b> ：Timer2 输出后分频比选择位 0000 = 1:1 后分频比 0001 = 1:2 后分频比 • • • 1111 = 1:16 后分频比
bit 2	<b>TMR2ON</b> ：Timer2 使能位 1 = 使能 Timer2 0 = 禁止 Timer2
bit 1-0	<b>T2CKPS&lt;1:0&gt;</b> ：Timer2 时钟预分频值选择位 00 = 预分频值为 1 01 = 预分频值为 4 1x = 预分频值为 16



## 15.0 TIMER3 模块

Timer3 定时器 / 计数器模块具有以下特征：

- 可通过软件选择作为 16 位定时器或计数器
- 可读写的 8 位寄存器（TMR3H 和 TMR3L）
- 可选择使用器件时钟或 Timer1 内部振荡器作为内部或外部时钟源
- 溢出中断
- CCPx 特殊事件触发模块复位

图 15-1 所示为 Timer3 模块的简化框图。图 15-2 所示为此模块在读 / 写模式下的工作原理框图。

Timer3 模块由 T3CON 寄存器（寄存器 15-1）控制。该控制寄存器还用于为 CCP 模块选择时钟源。欲知更多信息，请参见第 16.2.2 节“Timer1/Timer3 模式选择”。

寄存器 15-1: T3CON: TIMER3 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
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON
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	<b>RD16:</b> 16 位读 / 写模式使能位 1 = 使能通过一次 16 位操作对 Timer3 寄存器进行读写 0 = 使能通过两次 8 位操作对 Timer3 寄存器进行读写
bit 6,3	<b>T3CCP&lt;2:1&gt;:</b> CCPx 的时钟源（Timer3 和 Timer1）使能位 1x = Timer3 是 CCP 模块的比较 / 捕捉时钟源 01 = Timer3 是 CCP2 模块的比较 / 捕捉时钟源； Timer1 是 CCP1 模块的比较 / 捕捉时钟源 00 = Timer1 是 CCP 模块的比较 / 捕捉时钟源
bit 5-4	<b>T3CKPS&lt;1:0&gt;:</b> Timer3 输入时钟预分频比选择位 11 = 1:8 预分频值 10 = 1:4 预分频值 01 = 1:2 预分频值 00 = 1:1 预分频值
bit 2	<b>T3SYNC:</b> Timer3 外部时钟输入同步控制位 （不适用于器件时钟来自 Timer1/Timer3。） 当 <b>TMR3CS = 1</b> 时： 1 = 不与外部时钟输入同步 0 = 与外部时钟输入同步 当 <b>TMR3CS = 0</b> 时： 忽略此位。当 TMR3CS = 0 时，Timer3 使用内部时钟。
bit 1	<b>TMR3CS:</b> Timer3 时钟源选择位 1 = 使用 Timer1 振荡器或 T13CKI 引脚信号作为外部时钟输入（在第一个下降沿之后的上升沿开始计数） 0 = 内部时钟（Fosc/4）
bit 0	<b>TMR3ON:</b> Timer3 使能位 1 = 使能 Timer3 0 = 禁止 Timer3



## 15.2 Timer3 16 位读 / 写模式

可将 Timer3 配置为 16 位读写模式（见图 15-2）。当 RD16 控制位（T3CON<7>）置 1 时，TMR3H 的地址被映射到 Timer3 的高字节缓冲寄存器。对 TMR3L 的读操作将把 Timer3 的高字节内容装入 Timer3 高字节缓冲寄存器。这种方式使用户可以精确地读取 Time1 的全部 16 位，而不需要像先读高字节再读低字节那样：由于两次读取之间可能存在进位，而不得不验证读取的有效性。

对 Timer3 的高字节进行写操作也必须通过 TMR3H 缓冲器进行。在写入 TMR3L 的同时，使用 TMR3H 的内容更新 Timer3 的高字节。这样允许用户将所有的 16 位值一次写入 Timer3 的高字节和低字节。

在该模式下不能直接读写 Timer3 的高字节。所有读写都必须通过 Timer3 高字节缓冲寄存器进行。

写入 TMR3H 不会清零 Timer3 预分频器。只有在写入 TMR3L 时才会清零该预分频器。

## 15.3 使用 Timer1 振荡器作为 Timer3 的时钟源

Timer1 内部振荡器可用作 Timer3 的时钟源。通过将 T1OSCEN（T1CON<3>）位置 1，可使能 Timer1 振荡器。要将其用作 Timer3 的时钟源，还必须将 TMR3CS 位置 1。如前所述，这样做也会将 Timer3 配置为在振荡器时钟源的每个上升沿递增。

在第 13.0 节“Timer1 模块”中对 Timer1 振荡器做出了说明。

## 15.4 Timer3 中断

TMR3 寄存器对（TMR3H:TMR3L）从 0000h 开始递增直到 FFFFh，然后溢出返回到 0000h。如果允许 Timer3 中断，该中断就会在溢出时产生，并置 1 中断标志位 TMR3IF（PIR2<1>）。可以通过将 Timer3 中断允许位 TMR3IE（PIE2<1>）置 1 或清零来允许或禁止该中断。

## 15.5 使用 CCPx 特殊事件触发信号来复位 Timer3

如果 CCP1 或 CCP2 模块被配置为使用 Timer3，并在比较模式下产生特殊事件触发信号（CCPxM<3:0> = 1011），该触发信号将复位 Timer3。如果使能了 A/D 模块，来自 CCP2 的触发信号还将启动 A/D 转换（欲知更多信息，请参见第 16.3.4 节“特殊事件触发信号”）。

要使用这一功能，必须将 Timer3 模块配置为定时器或同步计数器。在这种情况下，CCPRxH:CCPRxL 寄存器对实际上变成了 Timer3 的周期寄存器。

如果 Timer3 在异步计数器模式下运行，复位操作将不起作用。

如果对 Timer3 的写操作和来自 CCPx 模块的特殊事件触发信号同时发生，则写操作优先。

**注：** 来自 CCPx 模块的特殊事件触发信号不会将中断标志位 TMR3IF（PIR2<1>）置 1。

表 15-1: TIMER3 作为定时器 / 计数器时相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR2	OSCFIF	CMIF	—	—	BCLIF	LVDIF	TMR3IF	—	59
PIE2	OSCFIE	CMIE	—	—	BCLIE	LVDIE	TMR3IE	—	59
IPR2	OSCFIP	CMIP	—	—	BCLIP	LVDIP	TMR3IP	—	59
TMR3L	Timer3 寄存器的低字节								59
TMR3H	Timer3 寄存器的高字节								59
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN $\overline{C}$	TMR1CS	TMR1ON	58
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYN $\overline{C}$	TMR3CS	TMR3ON	59

图注： — = 未实现，读为 0。Timer3 模块不使用阴影单元。

# PIC18F85J11 系列

---

注:



## 16.0 捕捉 / 比较 / PWM (CCP) 模块

PIC18F85J11 系列器件有两个 CCP (捕捉/比较/PWM) 模块 CCP1 和 CCP2。这两个模块均实现标准的捕捉、比较和脉宽调制 (Pulse-Width Modulation, PWM) 模式。

每个 CCP 模块都有一个 16 位寄存器，它可以用作 16 位捕捉寄存器、16 位比较寄存器或 PWM 主 / 从占空比寄存器。为了清楚起见，以下章节中描述的所有 CCP 模块操作均是针对 CCP2 的，但同样也适用于 CCP1。

寄存器 16-1: CCPxCON: CCPx 控制寄存器 (CCP1 和 CCP2 模块)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
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 5-4 **DCxB<1:0>**: CCPx 模块的 PWM 占空比 bit 1 和 bit 0

#### 捕捉模式:

未用。

#### 比较模式:

未用。

#### PWM 模式:

这些位是 10 位 PWM 占空比的低 2 位 (bit 1 和 bit 0)。占空比的高 8 位 (DCx9:DCx2) 在 CCPxL 中。

bit 3-0 **CCPxM<3:0>**: CCPx 模块模式选择位

0000 = 禁止捕捉 / 比较 / PWM (复位 CCPx 模块)

0001 = 保留

0010 = 比较模式, 匹配时输出电平翻转 (CCPxIF 位置 1)

0011 = 保留

0100 = 捕捉模式, 在每个下降沿捕捉

0101 = 捕捉模式, 在每个上升沿捕捉

0110 = 捕捉模式, 每 4 个上升沿捕捉

0111 = 捕捉模式, 每 16 个上升沿捕捉

1000 = 比较模式: 初始化 CCPx 引脚为低电平; 比较匹配时强制 CCPx 引脚为高电平 (CCPxIF 位置 1)

1001 = 比较模式: 初始化 CCPx 引脚为高电平; 比较匹配时强制 CCPx 引脚为低电平 (CCPxIF 位置 1)

1010 = 比较模式: 比较匹配时产生软件中断 (CCPxIF 位置 1, CCPx 引脚反映 I/O 状态)

1011 = 比较模式: 发生 CCPx 匹配时, 触发特殊事件、复位定时器并启动 A/D 转换 (CCPxIF 位置 1) <sup>(1)</sup>

11xx = PWM 模式

注 1: 如果 CCPxM<3:0> = 1011, 在发生 CCPx 匹配时, 将只复位定时器而不启动 A/D 转换。

## 16.1 CCP 模块配置

每个捕捉 / 比较 / PWM 模块均与一个控制寄存器（通常为 CCPxCON）和一个数据寄存器（CCPRx）相关联。数据寄存器由 2 个 8 位寄存器组成：CCPRxL（低字节）和 CCPRxH（高字节）。所有寄存器都是可读取写的。

### 16.1.1 CCP 模块和定时器资源

CCP 模块根据选定的工作模式使用 Timer1、Timer2 或 Timer3。该模块在捕捉和比较模式下使用 Timer1 和 Timer3，而在 PWM 模式下使用 Timer2。

表 16-1: CCPx 模式一定时器资源

CCPx 模式	定时资源
捕捉	Timer1 或 Timer3
比较	Timer1 或 Timer3
PWM	Timer2

要将哪个特定的定时器分配给 CCP 模块是由 T3CON 寄存器（寄存器 15-1）中的“将定时器分配给 CCPx 模块”使能位决定的。如果同时将两个模块配置为在相同的模式（捕捉 / 比较或 PWM）下工作，那么这两个模块在任何特定时间可同时处于激活状态且可共用相同的定时器资源。表 16-2 总结了这两个模块的相互关系。

根据选定的配置，至多同时可以有 4 个定时器处于激活状态，具有相同配置（捕捉 / 比较或 PWM）的模块可共用定时器资源。图 16-1 给出了可能的配置。

### 16.1.2 漏极开路输出选项

在输出模式（即比较或 PWM 模式）下工作时，可将 CCPx 引脚的驱动器配置为漏极开路输出。该功能允许通过外部上拉电阻将 CCPx 引脚上的电平拉为高电平，并允许漏极开路输出与外部电路通信，而无需额外的电平转换器。

漏极开路输出选项由 CCP2OD 和 CCP1OD 位（TRISG<6:5>）控制。通过置 1 适当的位来配置相应模块的引脚以执行漏极开路操作。

### 16.1.3 CCP2 引脚分配

可根据器件配置改变 CCP2 的引脚分配（捕捉输入、比较和 PWM 输出）。CCP2MX 配置位决定哪个引脚与 CCP2 复用。默认情况下，CCP2 引脚被分配给 RC1（CCP2MX = 1）。如果该配置位清零，则 CCP2 与 RE7 复用。

改变 CCP2 的引脚分配不会自动更改端口引脚的配置。无论 CCP2 分配给了哪个引脚，用户必须总是确认已为 CCP2 操作正确地配置了相应的 TRIS 寄存器。

图 16-1: CCPx 和定时器互连配置

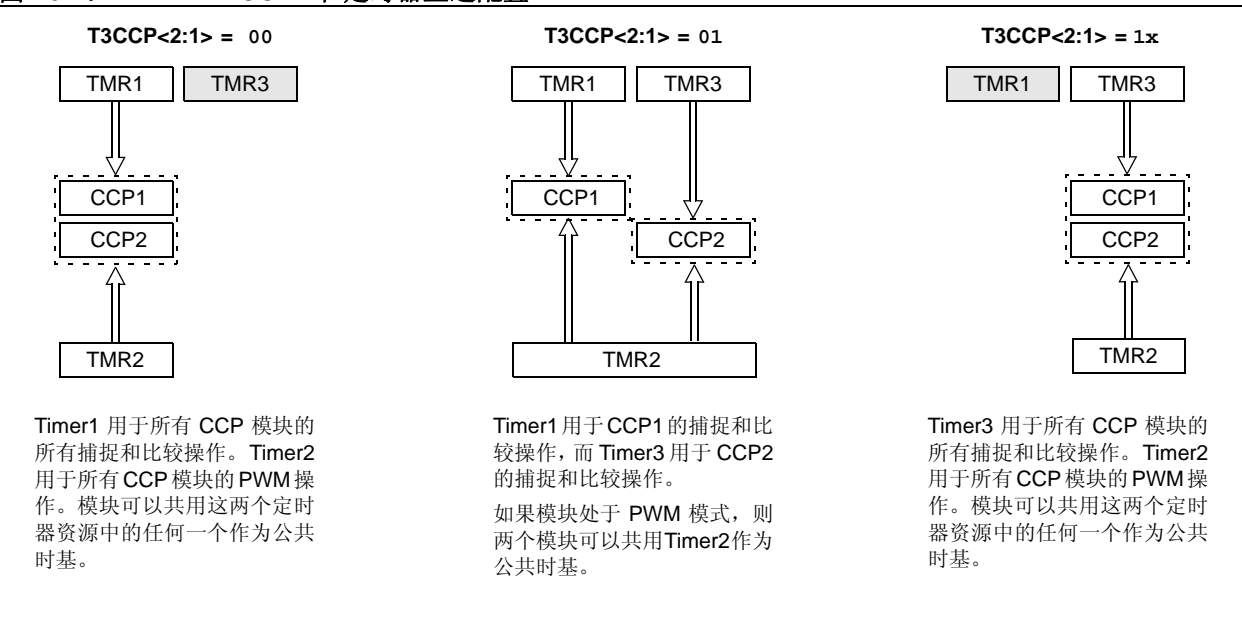


表 16-2: CCP1 和 CCP2 在使用定时器资源方面的相互关系

CCP1 模式	CCP2 模式	相互关系
捕捉	捕捉	每个模块都可以使用 TMR1 或 TMR3 作为时基。每个 CCP 模块的时基可以不同。
捕捉	比较	根据使用哪个时基，可将 CCP2 配置为特殊事件触发信号从而复位 TMR1 或 TMR3。特殊事件触发信号还可自动触发 A/D 转换。如果 CCP2 与 CCP1 使用相同的定时器作为时基，上述操作可能会对 CCP1 产生影响。
比较	捕捉	根据使用哪个时基，可将 CCP1 配置为特殊事件触发信号从而复位 TMR1 或 TMR3。如果 CCP1 与 CCP2 使用相同的定时器作为时基，上述操作可能会对 CCP2 产生影响。
比较	比较	两个模块都可被配置为特殊事件触发信号从而复位时基。CCP2 的特殊事件触发信号还可自动触发 A/D 转换。如果两个模块使用同一个时基，可能会发生冲突。
捕捉	PWM	无
比较	PWM	无
PWM	捕捉	无
PWM	比较	无
PWM	PWM	两个 PWM 具有相同的频率和更新速率（TMR2 中断）。

# PIC18F85J11 系列

## 16.2 捕捉模式

在捕捉模式下，当对应的 CCP2 引脚（根据器件配置不同，可为 RB3、RC1 或 RE7）上发生以下事件时，CCPR2H:CCPR2L 寄存器对即捕捉 TMR1 寄存器或 TMR3 寄存器的 16 位值。事件定义为下列情况之一：

- 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 CCP2M<3:0>（CCP2CON<3:0>）选择事件。当一次捕捉完成时，中断请求标志位 CCP2IF（PIR3<2>）置 1；它必须用软件清零。如果在读取寄存器 CCPR2 之前发生了另一次捕捉，那么之前捕捉的值将被新捕捉的值覆盖。

### 16.2.1 CCPx 引脚配置

在捕捉模式下，应该通过将对应的 TRIS 方向位置 1 来将相应的 CCPx 引脚配置为输入。

**注：** 如果 RB3/INT3/CCP2、RC1/T1OSI/CCP2 或 RE7/CCP2 引脚被配置为输出，对该端口的写操作会产生捕捉条件。

### 16.2.2 TIMER1/TIMER3 模式选择

用于捕捉功能的定时器（Timer1 和 / 或 Timer3）必须运行在定时器模式或同步计数器模式下。在异步计数器模式下，无法进行捕捉操作。用于每个 CCP 模块的定时器由 T3CON 寄存器进行选择（见第 16.1.1 节“CCP 模块和定时器资源”）。

### 16.2.3 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 CCP2IE（PIE3<2>）中断允许位清零，以避免错误中断，还应该在工作模式发生任何改变之后清零中断标志位 CCP2IF。

### 16.2.4 CCP 预分频器

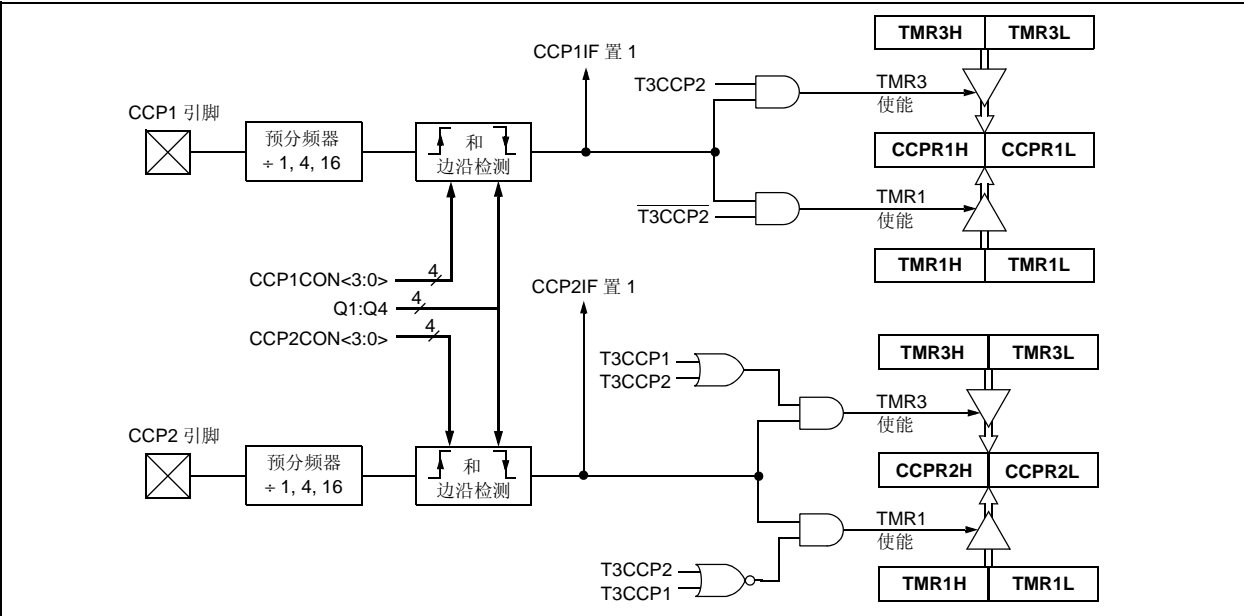
在捕捉模式下有 4 种预分频比设置。它们作为工作模式的一部分由模式选择位（CCP2M<3:0>）选择。当关闭 CCP2 模块或 CCP2 模块不处于捕捉模式下时，预分频计数器就会被清零。这意味着任何复位都将清零预分频计数器。

从一种捕捉预分频比切换到另一种捕捉预分频比可能会产生中断，且不会清零预分频计数器；因此切换后的第一次捕捉可能来自于一个非零的预分频器。例 16-1 是切换捕捉预分频比时建议采用的方法。该示例使预分频计数器清零且不会产生“误”中断。

#### 例 16-1： 改变捕捉预分频比

```
CLRF    CCP2CON    ; Turn CCP module off
MOVLW   NEW_CAPT_PS ; Load WREG with the
                        ; new prescaler mode
                        ; value and CCP ON
MOVWF   CCP2CON    ; Load CCP2CON with
                        ; this value
```

图 16-2： 捕捉模式工作原理框图



## 16.3 比较模式

在比较模式下，16位CCPR2寄存器的值将不断与TMR1或TMR3寄存器对的值相比较。当两者匹配时，CCP2引脚将被：

- 驱动为高电平
- 驱动为低电平
- 翻转电平（高电平变为低电平或低电平变为高电平）
- 保持不变（即反映 I/O 锁存器的状态）

引脚动作取决于模式选择位 CCP2M<3:0> 的值。同时，中断标志位 CCP2IF 置 1。

### 16.3.1 CCPx 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 CCP2CON 寄存器会将 RB3、RC1 或 RE7 比较输出锁存器（取决于器件配置）强制为默认的低电平。这不是 PORTB、PORTC 或 PORTE I/O 数据锁存器。

### 16.3.3 软件中断模式

当选择了产生软件中断模式时（CCP2M<3:0> = 1010），CCP2 引脚将不受影响。如果中断被允许（CCP2IE 位置 1），将只产生一个 CCP2 中断。

### 16.3.4 特殊事件触发信号

两个 CCP 模块都配备一个特殊事件触发器。该触发信号为内部硬件信号，是在比较模式下产生以触发其他模块的动作。通过选择比较特殊事件触发模式（CCP2M<3:0> = 1011）使能特殊事件触发器。

对于任何一个 CCP 模块，无论当前使用哪个定时器资源作为模块的时基，特殊事件触发信号将把对应的定时器寄存器对复位。这样 CCPRx 寄存器就可用作任何一个定时器的可编程周期寄存器。

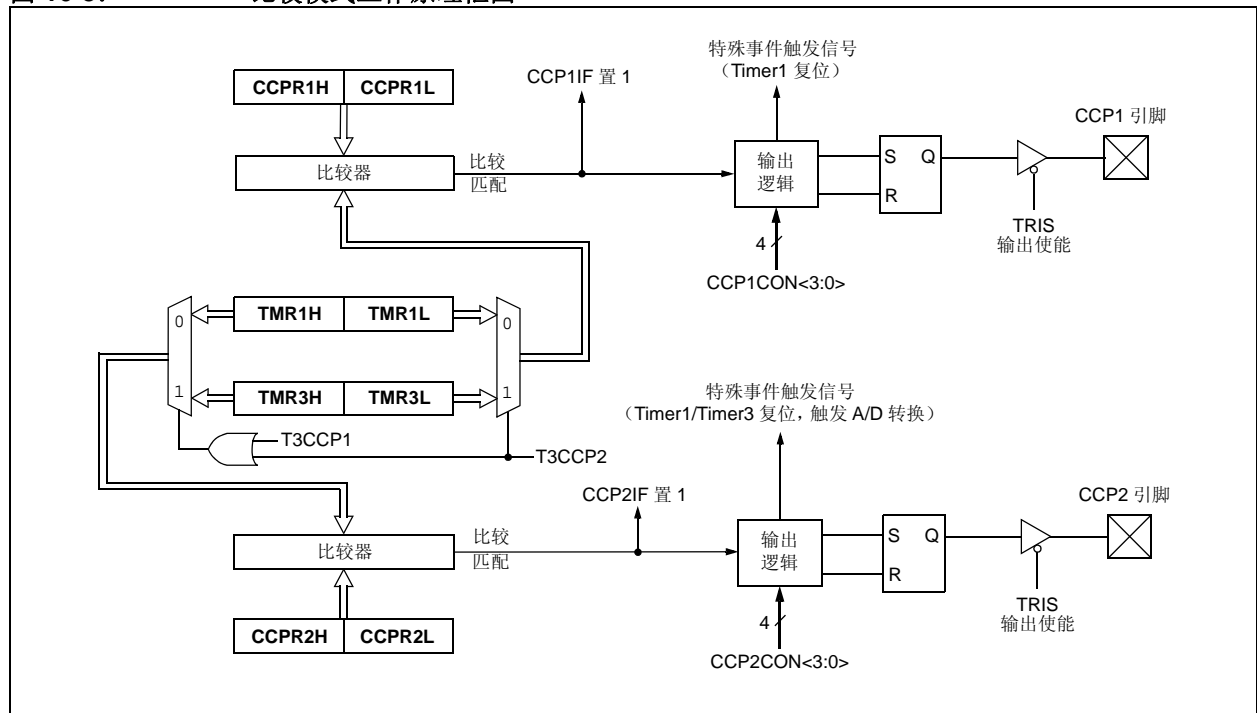
CCP2 的特殊事件触发信号还能启动 A/D 转换。要实现此功能，必须首先使能 A/D 转换器。

**注：** CCP1 的特殊事件触发信号仅复位 Timer1/Timer3，并且即使已使能 A/D 转换器也不会启动 A/D 转换。

### 16.3.2 TIMER1/TIMER3 模式选择

如果 CCPx 模块使用比较功能，Timer1 和 / 或 Timer3 必须运行在定时器模式或同步计数器模式下。在异步计数器模式下，无法进行比较操作。

图 16-3: 比较模式工作原理框图



# PIC18F85J11 系列

表 16-3: 与捕捉、比较、TIMER1 和 TIMER3 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR	58
PIR3	—	—	RC2IF	TX2IF	—	CCP2IF	CCP1IF	—	59
PIE3	—	—	RC2IE	TX2IE	—	CCP2IE	CCP1IE	—	59
IPR3	—	—	RC2IP	TX2IP	—	CCP2IP	CCP1IP	—	59
PIR2	OSCFIF	CMIF	—	—	BCLIF	LVDIF	TMR3IF	—	59
PIE2	OSCFIE	CMIE	—	—	BCLIE	LVDIE	TMR3IE	—	59
IPR2	OSCFIP	CMIP	—	—	BCLIP	LVDIP	TMR3IP	—	59
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	60
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	—	TRISE1	TRISE0	60
TRISG	SPIOD	CCP2OD	CCP1OD	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	60
TMR1L	Timer1 寄存器的低字节								58
TMR1H	Timer1 寄存器的高字节								58
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	58
TMR3H	Timer3 寄存器的高字节								59
TMR3L	Timer3 寄存器的低字节								59
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON	59
CCPR1L	捕捉 / 比较 / PWM 寄存器 1 的低字节								60
CCPR1H	捕捉 / 比较 / PWM 寄存器 1 的高字节								60
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	60
CCPR2L	捕捉 / 比较 / PWM 寄存器 2 的低字节								61
CCPR2H	捕捉 / 比较 / PWM 寄存器 2 的高字节								61
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	61

图注: — = 未实现, 读为 0。捕捉 / 比较、Timer1 或 Timer3 不使用阴影单元。

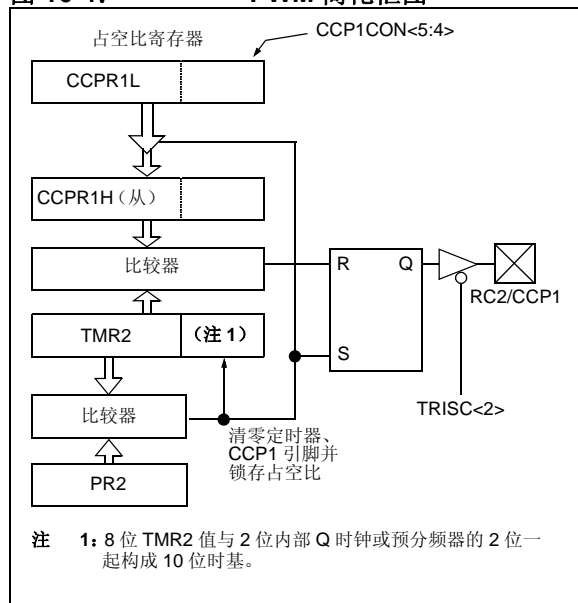
## 16.4 PWM 模式

在脉宽调制 (PWM) 模式下, CCP2 引脚会产生最高 10 位分辨率的 PWM 输出信号。由于 CCP2 引脚与 PORTB、PORTC 或 PORTE 数据锁存器复用, 必须清零相应的 TRIS 位才能使 CCP2 引脚用作输出引脚。

**注:** 清零 CCP2CON 寄存器会将 RB3、RC1 或 RE7 输出锁存器 (取决于器件配置) 强制为默认的低电平。这不是 PORTB、PORTC 或 PORTE I/O 数据锁存器。

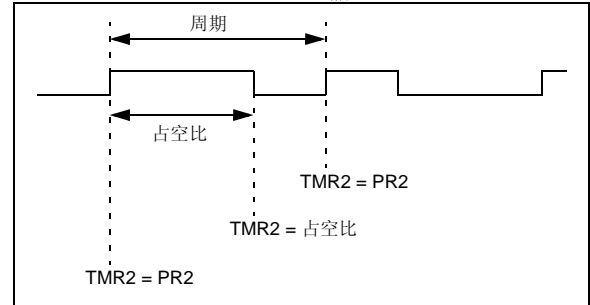
图16-4所示为在PWM模式下的CCP1模块的简化框图。如需了解如何设置 CCP 模块使之工作于 PWM 模式的详细步骤, 请参见第 16.4.3 节 “设置 PWM 操作”。

图 16-4: PWM 简化框图



PWM 输出 (图 16-5) 包含一个时基 (周期) 和一段输出保持为高电平的时间 (占空比)。PWM 频率为周期的倒数 (1/周期)。

图 16-5: PWM 输出



### 16.4.1 PWM 周期

通过写 PR2 寄存器来指定 PWM 周期。可以使用以下公式来计算 PWM 周期：

公式 16-1:

$$\text{PWM 周期} = (\text{PR2} + 1) \cdot 4 \cdot \text{Tosc} \cdot (\text{TMR2 预分频值})$$

PWM 频率定义为  $1/[\text{PWM 周期}]$ 。

当 TMR2 等于 PR2 时, 在下一递增周期中发生以下三个事件:

- 清零 TMR2
- CCP2 引脚被置 1 (例外情况: 如果 PWM 占空比 = 0%, 则 CCP2 引脚不被置 1)
- PWM 占空比从 CCPR2L 锁存到 CCPR2H

**注:** 确定 PWM 频率时不会用到 Timer2 后分频器 (见第 14.0 节 “Timer2 模块”)。使用后分频器时, 可用不同于 PWM 输出频率的频率进行数据更新。

# PIC18F85J11 系列

## 16.4.2 PWM 占空比

PWM 占空比可通过写 CCPR2L 寄存器和 CCP2CON<5:4> 位来指定。最高分辨率可达 10 位。CCPR2L 包含占空比的高 8 位，而 CCP1CON<5:4> 包含低 2 位。CCPR2L:CCP2CON<5:4> 表示这个 10 位值。下面的公式用于计算 PWM 占空比（以时间为单位）：

### 公式 16-2:

$$\text{PWM 占空比} = (\text{CCPR2L:CCP2CON<5:4>}) \cdot T_{\text{osc}} \cdot (\text{TMR2 预分频值})$$

可在任何时候写入 CCPR2L 和 CCP2CON<5:4>，但直到 PR2 与 TMR2 中的值匹配（即周期结束）时，占空比的值才被锁存到 CCPR2H。在 PWM 模式下，CCPR2H 是只读寄存器。

CCPR2H 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。这种双重缓冲结构非常重要，可以避免在 PWM 操作中产生毛刺。

当 CCPR2H 和 2 位锁存器的值与 TMR2（加上内部 2 位 Q 时钟或 TMR2 预分频器的 2 位）匹配时，CCP2 引脚被清零。

在给定 PWM 频率的情况下，其最大 PWM 分辨率（位）由以下公式计算：

### 公式 16-3:

$$\text{PWM 分辨率（最大）} = \frac{\log\left(\frac{F_{\text{OSC}}}{F_{\text{PWM}}}\right)}{\log(2)} \text{ 位}$$

**注：** 如果 PWM 占空比值大于 PWM 周期，将不会清零 CCP2 引脚。

表 16-4: 40 MHz 时的 PWM 频率和分辨率示例

PWM 频率	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
定时器预分频值（1、4 或 16）	16	4	1	1	1	1
PR2 值	FFh	FFh	FFh	3Fh	1Fh	17h
最高分辨率（位）	14	12	10	8	7	6.58



## 16.4.3 设置 PWM 操作

在将 CCP 模块配置为 PWM 操作时应该遵循以下步骤：

1. 通过写 PR2 寄存器设置 PWM 周期。
2. 通过写入 CCPR2L 寄存器和 CCP2CON<5:4> 位来设置 PWM 占空比。
3. 通过清零相应的 TRIS 位将 CCP2 引脚配置为输出。
4. 通过写 T2CON 设置 TMR2 预分频值并使能 Timer2。
5. 配置 CCP2 模块使之工作于 PWM 模式。

**表 16-5: 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	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR	58
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	—	TMR2IF	TMR1IF	59
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	—	TMR2IE	TMR1IE	59
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	—	TMR2IP	TMR1IP	59
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	60
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	—	TRISE1	TRISE0	60
TRISG	SPIOD	CCP2OD	CCP1OD	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	60
TMR2	Timer2 寄存器								58
PR2	Timer2 周期寄存器								58
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	58
CCPR1L	捕捉 / 比较 / PWM 寄存器 1 的低字节								60
CCPR1H	捕捉 / 比较 / PWM 寄存器 1 的高字节								60
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	60
CCPR2L	捕捉 / 比较 / PWM 寄存器 2 的低字节								61
CCPR2H	捕捉 / 比较 / PWM 寄存器 2 的高字节								61
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	61

图注： — = 未实现，读为 0。PWM 或 Timer2 不使用阴影单元。

# PIC18F85J11 系列

---

注:

## 17.0 主同步串行端口（MSSP）模块

### 17.1 主 SSP（MSSP）模块概述

主同步串行端口（Master Synchronous Serial Port, MSSP）模块是用于同其他外设或单片机器件进行通信的串行接口。这些外设包括串行 EEPROM、移位寄存器、显示驱动器和 A/D 转换器等。MSSP 模块有以下两种工作模式：

- 串行外设接口（SPI）
- I<sup>2</sup>C™
  - 全主模式
  - 从模式（广播地址呼叫）

I<sup>2</sup>C 接口硬件上支持下列模式：

- 主模式
- 多主器件模式
- 从模式

### 17.2 控制寄存器

MSSP 模块有 3 个相关的控制寄存器。其中包括一个状态寄存器（SSPSTAT）和两个控制寄存器（SSPCON1 和 SSPCON2）。根据 MSSP 模块是在 SPI 模式还是 I<sup>2</sup>C 模式下工作，这些寄存器的用途及其各自的位将完全不同。

下面各节提供了更多详细信息。

### 17.3 SPI 模式

SPI 模式允许同时同步发送和接收 8 位数据。器件支持 SPI 的所有四种模式。通常使用以下三个引脚来实现通信：

- 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

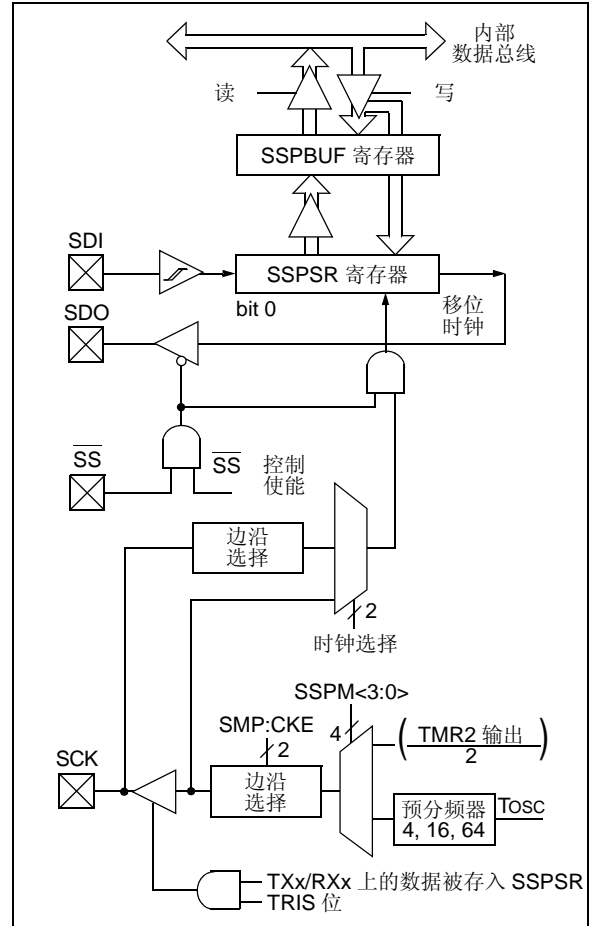
此外，当工作在从模式时要使用第 4 根引脚：

- 从选择（ $\overline{SS}$ ）– RF7/AN5/SS

图 17-1 给出了 MSSP 模块在 SPI 模式下工作时的原理图。

**注：** 通过将 SSPEN（SSPCON1<5>）位清零禁止 MSSP 模块可能不会将该模块复位。建议将 SSPSTAT、SSPCON1 和 SSPCON2 寄存器清零，并在将 SSPEN 位置 1 前选择该模式以使能 MSSP 模块。

图 17-1: MSSP 框图（SPI 模式）



# PIC18F85J11 系列

## 17.3.1 寄存器

MSSP 模块有四个寄存器用于 SPI 模式。这些寄存器包括：

- MSSP 控制寄存器 1 (SSPCON1)
- 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) – 不可直接访问

SSPCON1 和 SSPSTAT 是在 SPI 模式下工作的控制寄存器和状态寄存器。SSPCON1 寄存器是可读写的。SSPSTAT 的低 6 位是只读位，而 SSPSTAT 的高 2 位是可读写的。

SSPSR 是用来将数据移入或移出的移位寄存器。SSPBUF 是缓冲寄存器，用于写入和读出数据字节。

接收时，SSPSR 和 SSPBUF 共同构成一个双重缓冲接收器。当 SSPSR 接收到一个完整的字节后，该字节被送入 SSPBUF 寄存器，同时将中断标志位 SSPIF 置 1。

在发送过程中，SSPBUF 并不是双重缓冲的。对 SSPBUF 的写操作将同时写入 SSPBUF 和 SSPSR。

寄存器 17-1: SSPSTAT: MSSP 状态寄存器 (SPI 模式)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R0	R-0
SMP	CKE <sup>(1)</sup>	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 主模式:  
            1 = 在数据输出时间的末端采样输入数据  
            0 = 在数据输出时间的中间采样输入数据  
            SPI 从模式:  
            当 SPI 工作在从模式时，必须将 SMP 清零。
- bit 6      **CKE:** SPI 时钟选择位 <sup>(1)</sup>  
            1 = 当时钟状态从有效变为空闲时开始发送  
            0 = 当时钟状态从空闲变为有效时开始发送
- bit 5      **D/A:** 数据 / 地址位  
            仅在 I<sup>2</sup>C™ 模式下使用。
- bit 4      **P:** 停止位  
            仅在 I<sup>2</sup>C 模式下使用。当禁止 MSSP 模块 (SSPEN 清零) 时该位被清零。
- bit 3      **S:** 启动位  
            仅在 I<sup>2</sup>C 模式下使用。
- bit 2      **R/W:** 读 / 写信息位  
            仅在 I<sup>2</sup>C 模式下使用。
- bit 1      **UA:** 更新地址位  
            仅在 I<sup>2</sup>C 模式下使用。
- bit 0      **BF:** 缓冲器满状态位 (仅用于接收模式)  
            1 = 接收完成, SSPBUF 满  
            0 = 接收未完成, SSPBUF 空

注 1: 时钟极性由 CKP 位 (SSPCON1<4>) 设置。

**寄存器 17-2: SSPCON1: MSSP 控制寄存器 1 (SPI 模式)**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV <sup>(1)</sup>	SSPEN <sup>(2)</sup>	CKP	SSPM3 <sup>(3)</sup>	SSPM2 <sup>(3)</sup>	SSPM1 <sup>(3)</sup>	SSPM0 <sup>(3)</sup>
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            **WCOL:** 写冲突检测位 (仅用于发送模式)  
1 = 正在发送前一个字时, 又有数据写入 SSPBUF 寄存器 (必须用软件清零)  
0 = 未发生冲突
- bit 6            **SSPOV:** 接收溢出指示位 <sup>(1)</sup>  
SPI 从模式:  
1 = 当 SSPBUF 寄存器中仍保存前一数据时, 又接收到一个新字节。如果溢出, SSPSR 中的数据会丢失。  
溢出只发生在从模式下。即使只是发送数据, 用户也必须读 SSPBUF, 以避免将溢出标志位置 1 (必须用软件清零)。  
0 = 无溢出
- bit 5            **SSPEN:** 主同步串行端口使能位 <sup>(2)</sup>  
1 = 使能串行端口并将 SCK、SDO、SDI 和  $\overline{SS}$  配置为串行端口引脚  
0 = 禁止串行端口并将这些引脚配置为 I/O 端口引脚
- bit 4            **CKP:** 时钟极性选择位  
1 = 空闲状态时, 时钟为高电平  
0 = 空闲状态时, 时钟为低电平
- bit 3-0          **SSPM<3:0>:** 主同步串行端口模式选择位 <sup>(3)</sup>  
0101 = SPI 从模式, 时钟 = SCK 引脚, 禁止  $\overline{SS}$  引脚控制, 可将  $\overline{SS}$  用作 I/O 引脚  
0100 = SPI 从模式, 时钟 = SCK 引脚, 使能  $\overline{SS}$  引脚控制  
0011 = SPI 主模式, 时钟 = TMR2 输出 /2  
0010 = SPI 主模式, 时钟 = Fosc/64  
0001 = SPI 主模式, 时钟 = Fosc/16  
0000 = SPI 主模式, 时钟 = Fosc/4

- 注**    **1:** 在主模式下, 溢出位不会被置 1, 因为每次接收 (和发送) 新数据都是通过写 SSPBUF 寄存器启动的。  
**2:** 当该位使能时, 必须将此引脚正确地配置为输入或输出。  
**3:** 在此未列出的位组合是保留的或仅在 I<sup>2</sup>C™ 模式下使用。

# PIC18F85J11 系列

## 17.3.2 工作原理

当初始化 **SPI** 时，需要指定几个选项。可通过编程相应的控制位 (**SSPCON1<5:0>** 和 **SSPSTAT<7:6>**) 来指定。这些控制位用于指定以下选项：

- 主模式 (**SCK** 作为时钟输出)
- 从模式 (**SCK** 作为时钟输入)
- 时钟极性 (**SCK** 处于空闲状态)
- 输入数据的采样阶段 (数据输出时间的中间或末端)
- 时钟边沿 (在 **SCK** 的上升沿 / 下降沿输出数据)
- 时钟速率 (仅用于主模式)
- 从选择模式 (仅用于从模式)

**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 / 移出器件，最高有效位在前。在新数据接收完毕前，**SSPBUF** 保存上次写入 **SSPSR** 的数据。一旦 8 位数据接收完毕，该字节就被移入 **SSPBUF** 寄存器。然后，缓冲器满检测位 **BF** (**SSPSTAT<0>**) 和 **MSSP** 中断标志位 **SSPIF** 均被置 1。这种双重缓冲数据接收方式 (使用 **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** 寄存器的操作均无效，并且将写冲突检测位 **WCOL** (**SSPCON1<7>**) 置 1。用户必须用软件将 **WCOL** 位清零才能判断以后对 **SSPBUF** 寄存器的写入是否成功。

为确保应用软件能有效地接收数据，应该在下一个要发送的数据字节写入 **SSPBUF** 之前，读取 **SSPBUF** 中现有的数据。缓冲器满检测位 **BF** (**SSPSTAT<0>**) 用于表示何时 **SSPBUF** 载入了已接收的数据 (发送完成)。当 **SSPBUF** 中的数据被读取后，**BF** 位即被清零。如果 **SPI** 仅作为一个发送器，则不必理会该数据。通常可用 **MSSP** 中断来判断发送 / 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，用软件查询的方法同样可确保不会发生写冲突。例 17-1 说明了装载 **SSPBUF** (**SSPSR**) 进行数据发送的过程。

用户不能直接读写 **SSPSR** 寄存器，只能通过寻址 **SSPBUF** 寄存器来访问。此外，**SSPSTAT** 寄存器用于指示各种状态。

**注：** 为避免在主模式下丢失数据，在每次发送之间都必须读 **SSPBUF** 以将缓冲器满检测位 **BF** (**SSPSTAT<0>**) 清零。

例 17-1: 装载 **SSPBUF** (**SSPSR**) 寄存器

LOOP	BTFSS	SSPSTAT, BF	;Has data been received (transmit complete)?
	BRA	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

## 17.3.3 使能 SPI I/O

要使能串行端口，MSSP使能位SSPEN (SSPCON1<5>) 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。这样就将 SDI、SDO、SCK 和 SS 引脚配置为串行端口引脚。要让上述引脚充当串行端口功能，必须按如下说明正确设置其中一些引脚的数据方向位（在 TRIS 寄存器中）：

- SDI 由 SPI 模块自动控制
- 必须将 SDO 对应的 TRISC<5> 位清零
- 必须将 SCK（主模式）对应的 TRISC<3> 位清零
- 必须将 SCK（从模式）对应的 TRISC<3> 位置 1
- 必须将 SS 对应的 TRISF<7> 位置 1

对于不需要的串行端口功能，可通过将对应的数据方向（TRIS）寄存器设置为相反值来屏蔽。

## 17.3.4 漏极开路输出选项

可选择将 SDO 输出和 SCK 时钟引脚的驱动器配置为漏极开路输出。此功能允许通过外部上拉电阻将引脚上的

电平拉至更高的电平，以允许输出与外部电路通信，而无需额外的电平转换器。

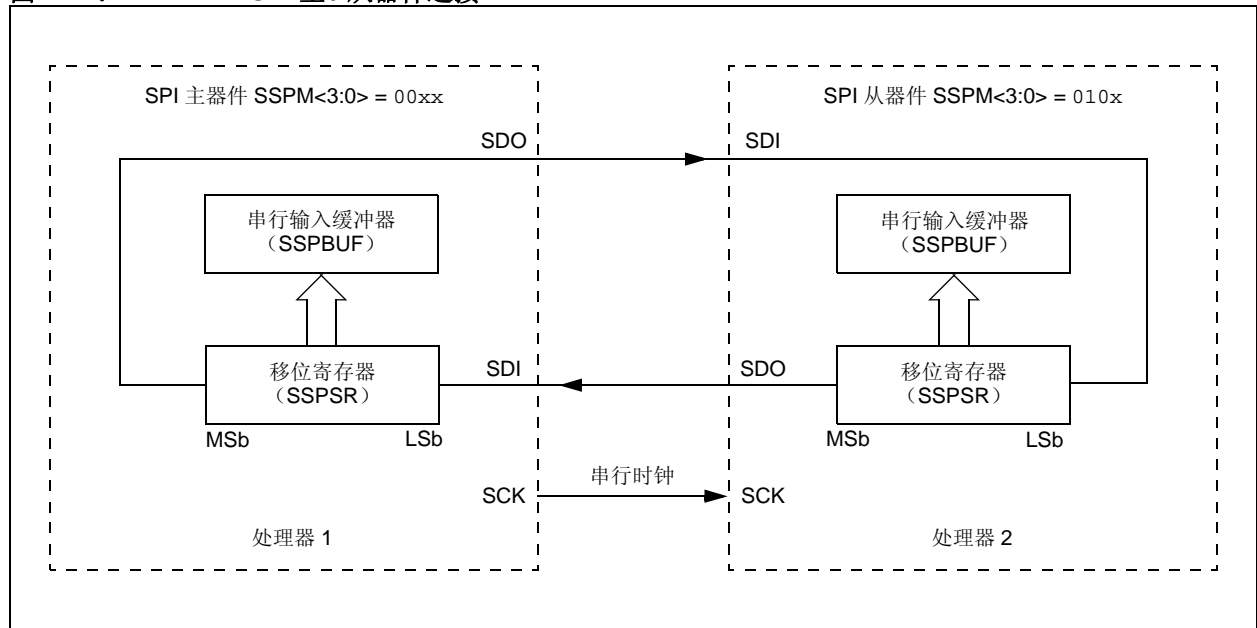
漏极开路输出选项由 SPIOD 位（TRISG<7>）控制。置 1 SPIOD 位可将这两个引脚配置为漏极开路操作。

## 17.3.5 典型连接

图 17-2 给出了两个单片机之间的典型连接。主控制器（处理器 1）通过发送 SCK 信号来启动数据传输。在两个处理器的移位寄存器之间，数据在编程设定的时钟边沿被传送，并在相反的时钟边沿被锁存。必须将两个处理器的时钟极性（CKP）设定为相同，这样两个控制器就可以同时收发数据。数据是否有效取决于应用软件。这就导致以下三种数据传输场景：

- 主器件发送数据 — 从器件发送无效（Dummy）数据
- 主器件发送数据 — 从器件发送数据
- 主器件发送无效数据 — 从器件发送数据

图 17-2: SPI 主 / 从器件连接



## 17.3.6 主模式

因为由主器件控制 SCK 信号，所以它可以随时启动数据传输。主器件根据软件协议确定从器件（图 17-2 中的处理器 2）应在何时广播数据。

在主模式下，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（通过编程将其设置为输入）。SSPSR 寄存器按设定的时钟速率连续移入 SDI 引脚上的信号。每收到一个字节，就将其装入 SSPBUF 寄存器，就像接收到普通字节一样（中断和状态位相应置 1）。这在以“线路活动监视器”（Line Activity Monitor）方式工作的接收器应用中很有用。

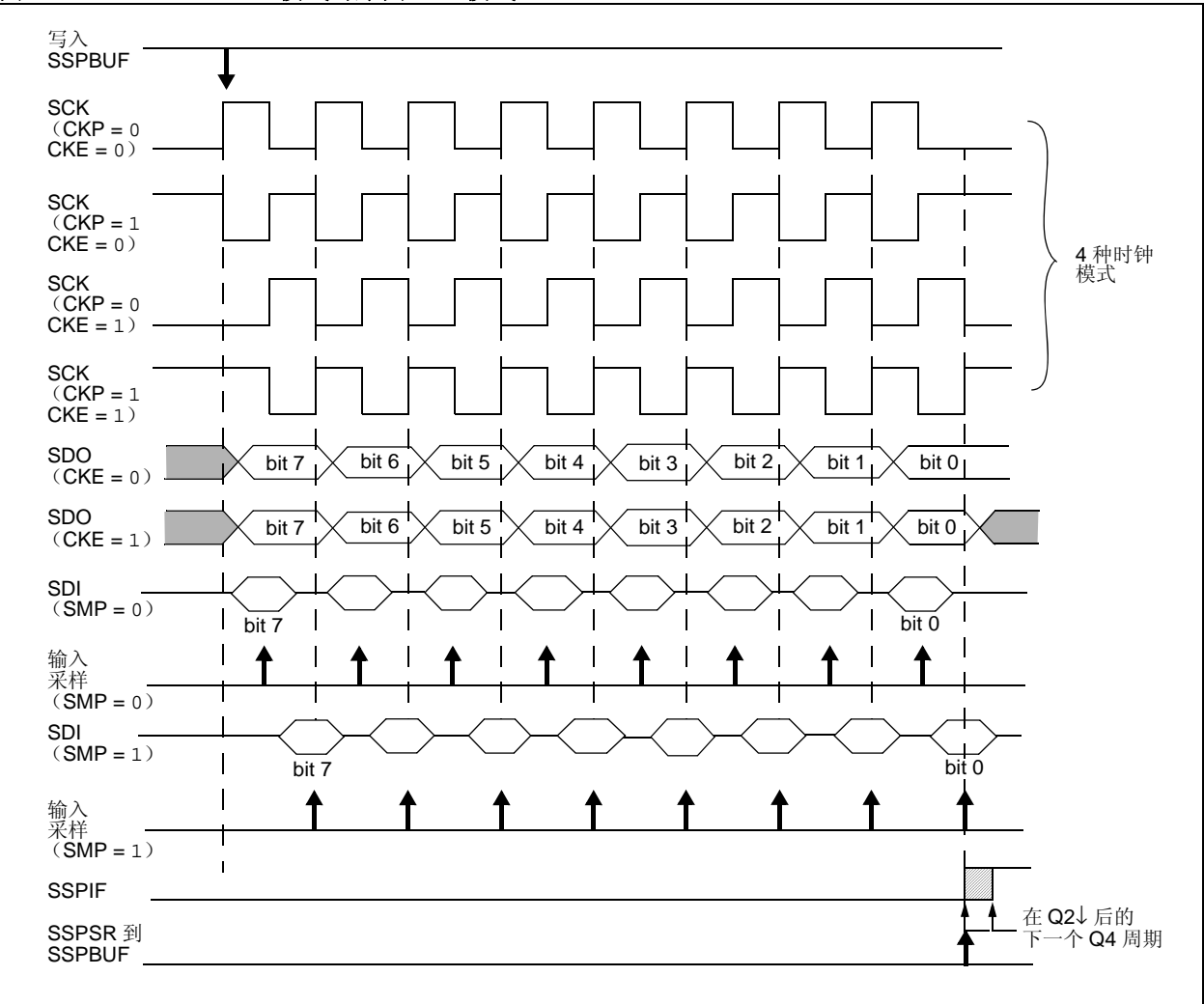
可通过对 CKP 位（SSPCON1<4>）进行适当的编程来选择时钟极性。图 17-3、图 17-5 和图 17-6 将给出 SPI 通信的时序图，其中最先发送最高有效字节。在主模式下，SPI 时钟速率（比特率）可由用户编程设定为下面某种速率：

- $F_{osc}/4$ （或  $T_{CY}$ ）
- $F_{osc}/16$ （或  $4 \cdot T_{CY}$ ）
- $F_{osc}/64$ （或  $16 \cdot T_{CY}$ ）
- Timer2 输出 /2

这样可使数据速率最高为 10.00 Mbps（时钟速率为 40 MHz）。

图 17-3 给出了主模式的时序图。当 CKE 位置 1 时，SDO 数据在 SCK 出现时钟边沿前一直有效。图中所示的输入采样的变化由 SMP 位的状态决定。图中给出了接收到的数据装入 SSPBUF 的时间。

图 17-3: SPI 模式时序图（主模式）





## 17.3.7 从模式

在从模式下，当 **SCK** 引脚上有外部时钟脉冲出现时发送和接收数据。当最后一位数据被锁存后，中断标志位 **SSPIF** 置 1。

在使能该模块的 **SPI** 从模式之前，时钟线必须符合正确的空闲状态。可以通过读 **SCK** 引脚来观察时钟线。由 **CKP** 位 (**SSPCON1<4>**) 确定空闲状态。

在从模式下，外部时钟由 **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。

在休眠模式下，从器件仍可发送 / 接收数据。当收到一个字节时，会将器件从休眠模式下唤醒。

## 17.3.8 从选择同步

**SS** 引脚允许器件工作于同步从模式。**SPI** 必须处于从模式，并使能 **SS** 引脚控制 (**SSPCON1<3:0> = 04h**)。当 **SS** 引脚为低电平时，使能数据的发送和接收，同时

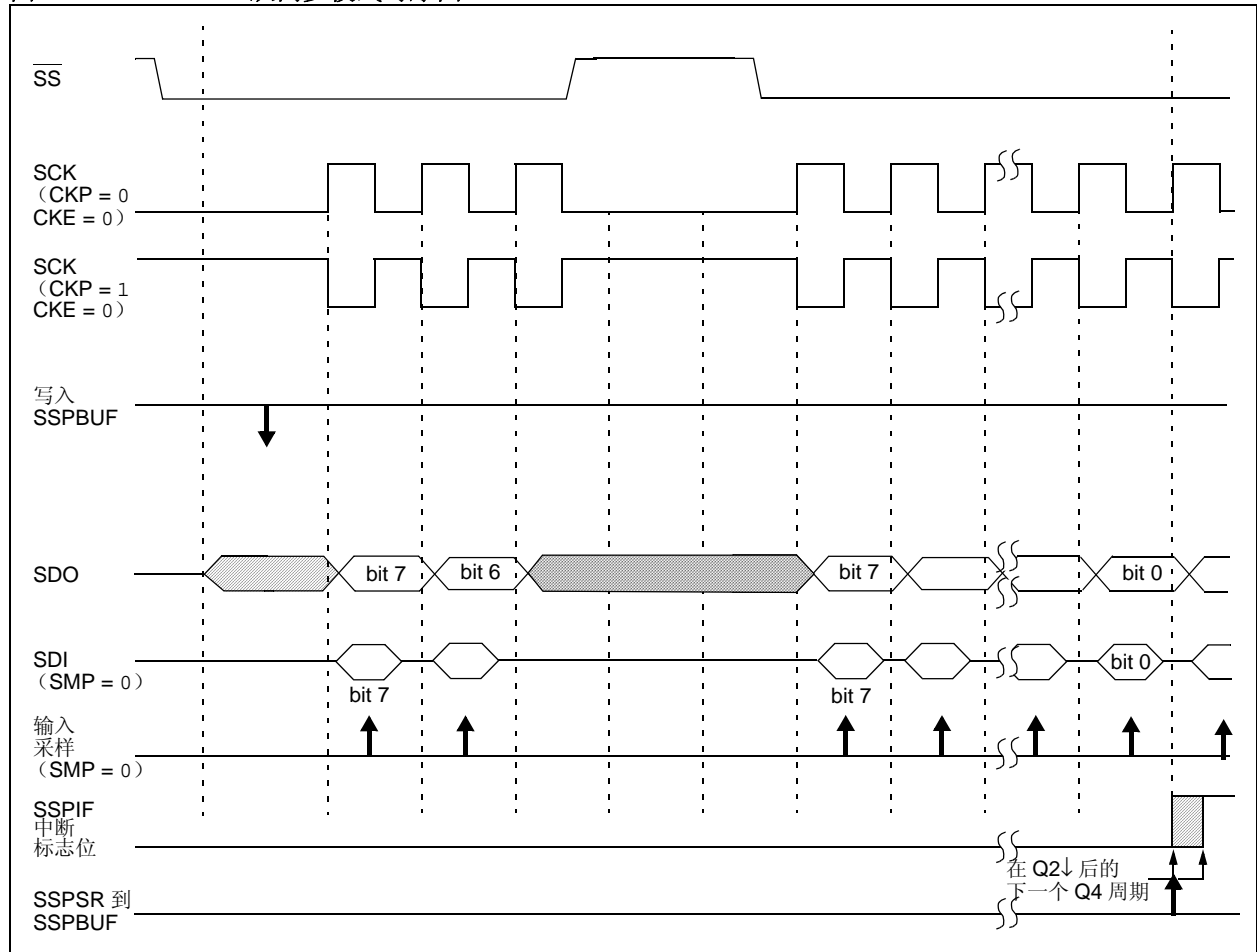
驱动 **SDO** 引脚。当 **SS** 引脚变为高电平时，即使是在字节的发送过程中，也不再驱动 **SDO** 引脚，而是将其变成悬空输出状态。根据应用需要，可在 **SDO** 引脚上外接上拉 / 下拉电阻。

- 注 1:** 当 **SPI** 处于从模式，并且 **SS** 引脚控制功能使能 (**SSPCON1<3:0> = 0100**) 时，如果 **SS** 引脚置为 **VDD** 电平将使 **SPI** 模块复位。
- 2:** 如果 **SPI** 工作在从模式下并且 **CKE** 置 1，则必须使能 **SS** 引脚控制功能。

当 **SPI** 模块复位后，位计数器被强制为 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，可以仿真二线制通信。当 **SPI** 需要作为接收器工作时，**SDO** 引脚可以被配置为输入端。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 17-4: 从同步模式时序图



# PIC18F85J11 系列

图 17-5: SPI 模式时序图 (从模式, CKE = 0)

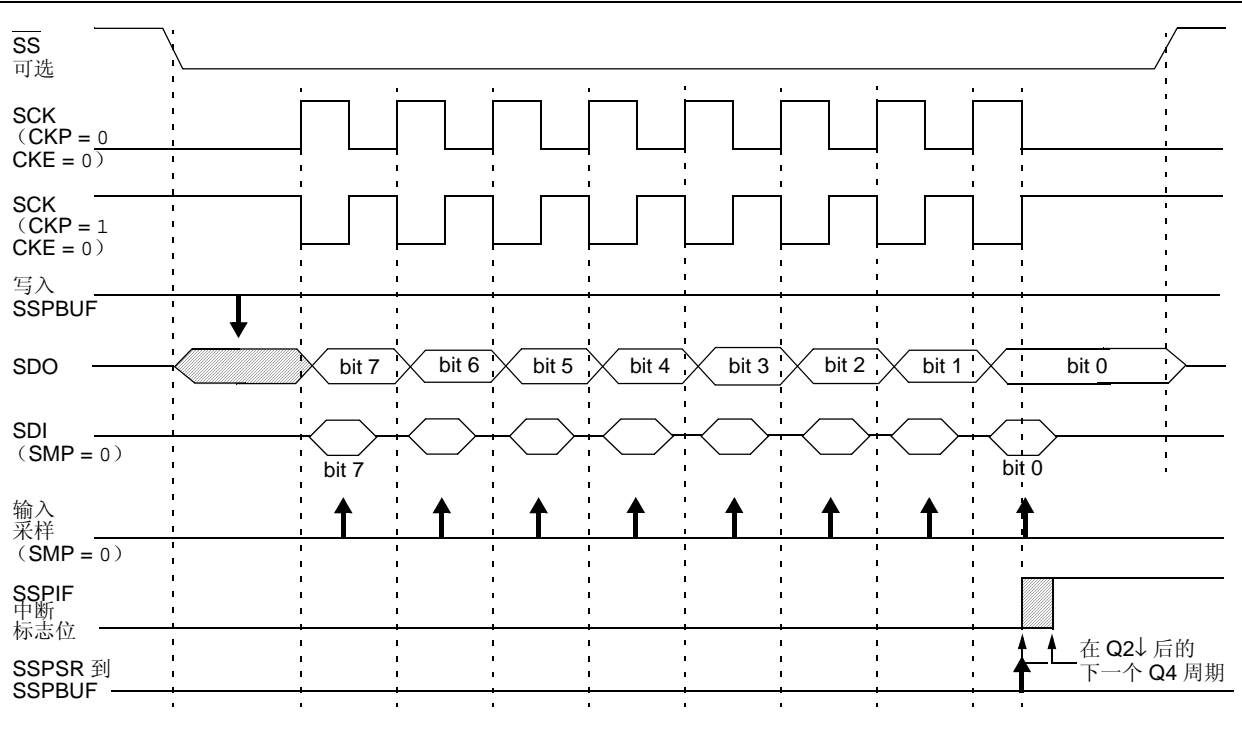
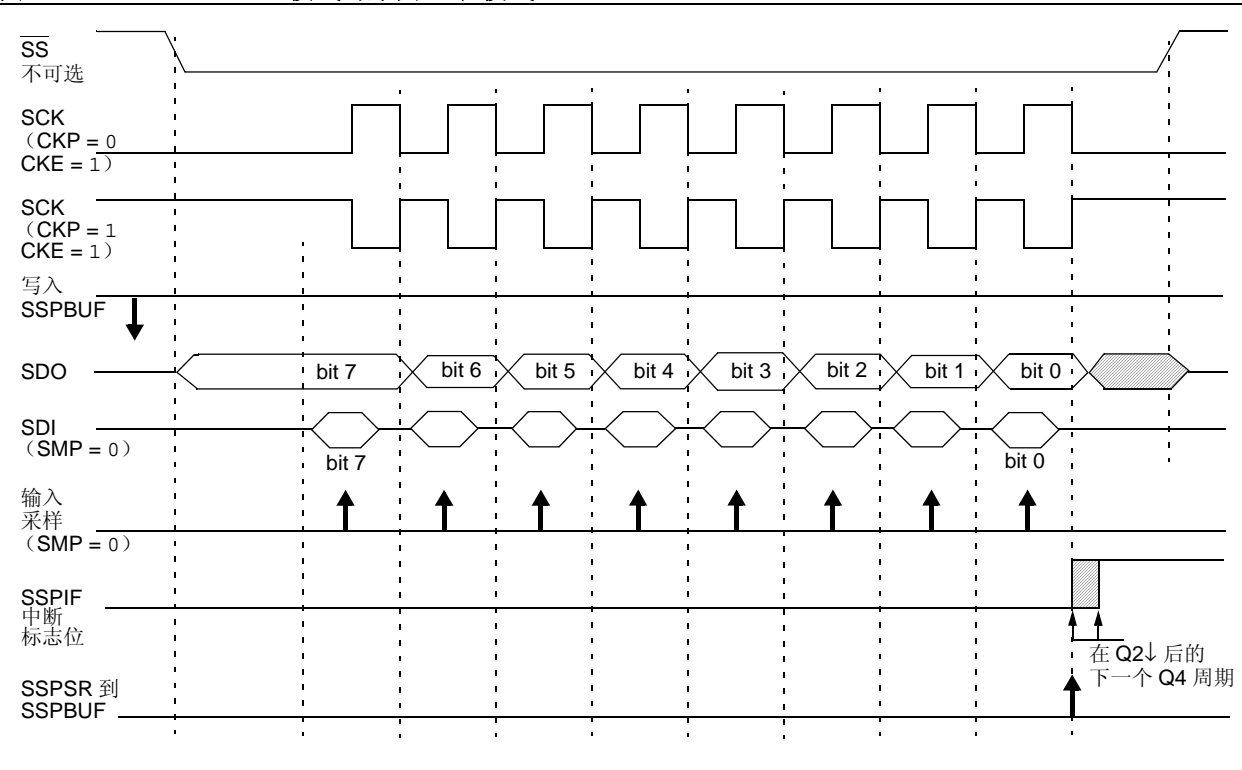


图 17-6: SPI 模式时序图 (从模式, CKE = 1)



## 17.3.9 在功耗管理模式下的操作

在 SPI 主模式下，模块时钟速度与全功耗模式下的不同；处于休眠模式时，所有时钟都停止。

在空闲模式下，需要为外设提供一个时钟。该时钟可以来自于主时钟源、辅助时钟源（32.768 kHz 的 Timer1 振荡器）或 INTRC 时钟源。更多信息请参见第 3.3 节“时钟源和振荡器切换”。

在大多数情况下，主器件为 SPI 数据提供的时钟速率并不重要；但是，每个系统都应该评估此因素。

如果允许了 MSSP 中断，那么当主器件发送完数据时这些中断可以将控制器从休眠模式或某种空闲模式唤醒。如果不想从休眠或空闲模式退出，应该禁止 MSSP 中断。

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 SPI 从模式下，SPI 发送 / 接收移位寄存器与器件异步工作。这可以使器件处于任何功耗管理模式，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，还将唤醒器件。

## 17.3.10 复位的影响

复位会禁止 MSSP 模块并终止当前的传输。

## 17.3.11 总线模式兼容性

表 17-1 中所示是标准 SPI 模式与 CKP 和 CKE 控制位状态的对应关系。

表 17-1: 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 17-2: 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	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	—	TMR2IF	TMR1IF	59
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	—	TMR2IE	TMR1IE	59
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	—	TMR2IP	TMR1IP	59
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	60
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	60
TRISG	SPIOD	CCP2OD	CCP1OD	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	60
SSPBUF	MSSP 接收缓冲器 / 发送寄存器								58
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	58
SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	58

图注：— = 未实现，读为 0。SPI 模式下的 MSSP 模块不使用阴影单元。

## 17.4 I<sup>2</sup>C 模式

MSSP 模块工作在 I<sup>2</sup>C 模式时，可以实现所有的主和从功能（包括广播呼叫支持），并且用硬件提供启动位和停止位的中断来判断总线何时空闲（多主器件功能）。MSSP 模块实现了标准模式规范以及 7 位和 10 位寻址。

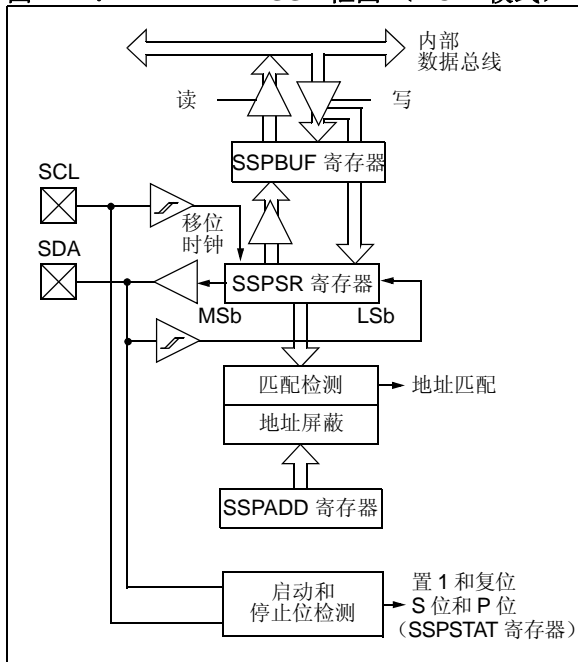
有两个引脚用于数据传输：

- 串行时钟（SCL）— RC3/SCK/SCL
- 串行数据（SDA）— RC4/SDI/SDA

用户必须通过将 TRISC<4:3> 位置 1 将上述引脚配置为输入。

**注：** 通过将 SSPEN（SSPCON1<5>）位清零禁止 MSSP 模块可能不会将该模块复位。建议将 SSPSTAT、SSPCON1 和 SSPCON2 寄存器清零，并在将 SSPEN 位置 1 前选择该模式以使能 MSSP 模块。

图 17-7: MSSP 框图 (I<sup>2</sup>C™ 模式)



### 17.4.1 寄存器

MSSP 模块有 6 个寄存器用于 I<sup>2</sup>C 操作。这些寄存器包括：

- MSSP 控制寄存器 1（SSPCON1）
- 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）— 不可直接访问
- MSSP 地址寄存器（SSPADD）

SSPCON1、SSPCON2 和 SSPSTAT 是 I<sup>2</sup>C 模式下的控制寄存器和状态寄存器。SSPCON1 和 SSPCON2 寄存器是可读写的。SSPSTAT 的低 6 位是只读位，而 SSPSTAT 的高 2 位是可读写的。

根据模块是工作在主模式还是从模式，SSPCON2 的许多位都有不同的功能；SSPCON2<5:2> 位在从模式下的名称也不同。SSPCON2 的不同功能分别在寄存器 17-5（针对主模式）和寄存器 17-6（针对从模式）中给出。

SSPSR 是用来将数据移入或移出的移位寄存器。SSPBUF 是缓冲寄存器，用于数据字节的写入和读出。

当 MSSP 被配置为 I<sup>2</sup>C 从模式时，SSPADD 寄存器将保存从器件地址。MSSP 被配置为主模式时，SSPADD 的低 7 位用作波特率发生器的重载值。

接收时，SSPSR 和 SSPBUF 共同构成一个双重缓冲接收器。当 SSPSR 接收到一个完整的字节后，该字节被送入 SSPBUF 寄存器，同时将中断标志位 SSPIF 置 1。

在发送过程中，SSPBUF 并不是双重缓冲的。对 SSPBUF 的写操作将同时写入 SSPBUF 和 SSPSR。

**寄存器 17-3: SSPSTAT: MSSP 状态寄存器 (I<sup>2</sup>C™ 模式)**

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R0	R-0
SMP	CKE	D/A	P <sup>(1)</sup>	S <sup>(1)</sup>	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:** 压摆率控制位  
在主或从模式下:  
 1 = 标准速率模式 (100 kHz 和 1 MHz) 下禁止压摆率控制  
 0 = 高速模式 (400 kHz) 下使能压摆率控制
- bit 6      **CKE:** SMBus 选择位  
在主或从模式下:  
 1 = 使能 SMBus 特定输入  
 0 = 禁止 SMBus 特定输入
- bit 5      **D/A:** 数据 / 地址位  
在主模式下:  
 保留。  
在从模式下:  
 1 = 表示上一个接收或发送的字节是数据  
 0 = 表示上一个接收或发送的字节是地址
- bit 4      **P:** 停止位 <sup>(1)</sup>  
 1 = 表示上一次检测到了停止位  
 0 = 表示上一次未检测到停止位
- bit 3      **S:** 启动位 <sup>(1)</sup>  
 1 = 表示上一次检测到了启动位  
 0 = 表示上一次未检测到启动位
- bit 2      **R/W:** 读 / 写信息位 (仅用于 I<sup>2</sup>C 模式)  
在从模式下: <sup>(2)</sup>  
 1 = 读  
 0 = 写  
在主模式下: <sup>(3)</sup>  
 1 = 正在进行发送  
 0 = 不在进行发送
- bit 1      **UA:** 更新地址位 (仅用于 10 位从模式)  
 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  
在发送模式下:  
 1 = SSPBUF 满  
 0 = SSPBUF 空  
在接收模式下:  
 1 = SSPBUF 满 (不包括 ACK 位和停止位)  
 0 = SSPBUF 空 (不包括 ACK 位和停止位)

- 注**    **1:** 当 SSPEN 被清零时该位在复位时清零。  
**2:** 该位用来保存在最近一次地址匹配后的 R/W 位信息。该位仅在从地址匹配开始到下一个启动位、停止位或非 ACK 位期间有效。  
**3:** 该位与 SEN、RSEN、PEN、RCEN 或 ACKEN 进行逻辑“或”运算的结果表示 MSSP 是否处于活动模式。

# PIC18F85J11 系列

寄存器 17-4: **SSPCON1: MSSP 控制寄存器 1 (I<sup>2</sup>C™ 模式)**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN <sup>(1)</sup>	CKP	SSPM3	SSPM2	SSPM1	SSPM0
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 **WCOL:** 写冲突检测位

在主发送模式下:

1 = 当 I<sup>2</sup>C 不满足启动发送数据的条件时, 试图向 SSPBUF 寄存器写入数据 (必须用软件清零)

0 = 未发生冲突

在从发送模式下:

1 = 正在发送前一个字时, 又有数据写入 SSPBUF 寄存器 (必须用软件清零)

0 = 未发生冲突

在接收模式下 (主或从模式):

此位为无关位。

bit 6 **SSPOV:** 接收溢出指示位

在接收模式下:

1 = SSPBUF 寄存器仍在保存前一字节时, 又接收到一个新的字节 (必须用软件清零)

0 = 无溢出

在发送模式下:

在发送模式下, 此位为无关位。

bit 5 **SSPEN:** 主同步串行端口使能位 <sup>(1)</sup>

1 = 使能串行端口并将 SDA 和 SCL 引脚配置为串行端口引脚

0 = 禁止串行端口并将这些引脚配置为 I/O 端口引脚

bit 4 **CKP:** SCK 释放控制位

在从模式下:

1 = 释放时钟

0 = 保持时钟低电平 (时钟延长); 用来确保数据建立时间

在主模式下:

在此模式下未使用。

bit 3-0 **SSPM<3:0>:** 同步串行端口模式选择位

1111 = I<sup>2</sup>C 从模式, 10 位地址, 并允许启动位和停止位中断

1110 = I<sup>2</sup>C 从模式, 7 位地址, 并允许启动位和停止位中断

1011 = I<sup>2</sup>C 由固件控制的主模式 (从器件空闲)

1000 = I<sup>2</sup>C 主模式, 时钟 = Fosc/(4 \* (SSPADD + 1))

0111 = I<sup>2</sup>C 从模式, 10 位地址

0110 = I<sup>2</sup>C 从模式, 7 位地址

此处未列出的位组合为保留或只用于 SPI 模式。

**注 1:** 当该位被使能时, 必须将 SDA 和 SCL 引脚配置为输入引脚。

**寄存器 17-5: SSPCON2: MSSP 控制寄存器 2 (I<sup>2</sup>C™ 主模式)**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT <sup>(1)</sup>	ACKEN <sup>(2)</sup>	RCEN <sup>(2)</sup>	PEN <sup>(2)</sup>	RSEN <sup>(2)</sup>	SEN <sup>(2)</sup>
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  
在主模式下未使用。
- bit 6      **ACKSTAT:** 应答状态位 (仅用于主发送模式)  
1 = 未收到来自从器件的应答  
0 = 收到来自从器件的应答
- bit 5      **ACKDT:** 应答数据位 (仅用于主接收模式) <sup>(1)</sup>  
1 = 无应答  
0 = 应答
- bit 4      **ACKEN:** 应答序列使能位 <sup>(2)</sup>  
1 = 在 SDA 和 SCL 引脚上发起应答序列, 并发送 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 (仅用于主接收模式) <sup>(2)</sup>  
1 = 使能 I<sup>2</sup>C 接收模式  
0 = 接收空闲
- bit 2      **PEN:** 停止条件使能位 <sup>(2)</sup>  
1 = 在 SDA 和 SCL 引脚上发起停止条件; 由硬件自动清零。  
0 = 停止条件空闲
- bit 1      **RSEN:** 重复启动条件使能位 <sup>(2)</sup>  
1 = 在 SDA 和 SCL 引脚上发起重复启动条件; 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 <sup>(2)</sup>  
1 = 在 SDA 和 SCL 引脚上发起启动条件; 由硬件自动清零。  
0 = 启动条件空闲

- 注 1:** 用户在接收结束时发起一个应答时序, 同时发送该值。
- 2:** 如果 I<sup>2</sup>C 模块处于活动状态, 这些位可能不能被置 1 (没有缓存), 并且也可能不会写入 SSPBUF (或者禁止写 SSPBUF)。

# PIC18F85J11 系列

寄存器 17-6: **SSPCON2: MSSP 控制寄存器 2 (I<sup>2</sup>C™ 从模式)**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ADMSK5	ADMSK4	ADMSK3	ADMSK2	ADMSK1	SEN <sup>(1)</sup>
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	<b>GCEN:</b> 广播呼叫使能位 1 = 允许在 SSPSR 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	<b>ACKSTAT:</b> 应答状态位 在从模式下未使用。
bit 5-2	<b>ADMSK&lt;5:2&gt;:</b> 从地址掩码选择位 1 = 使能掩码 SSPADD 的相应位 0 = 禁止掩码 SSPADD 的相应位
bit 1	<b>ADMSK1:</b> 从地址最低有效位掩码选择位 <u>在 7 位寻址模式下:</u> 1 = 使能仅掩码 SSPADD<1> 0 = 禁止仅掩码 SSPADD<1> <u>在 10 位寻址模式下:</u> 1 = 使能仅掩码 SSPADD<1:0> 0 = 禁止仅掩码 SSPADD<1:0>
bit 0	<b>SEN:</b> 时钟延长使能位 <sup>(1)</sup> 1 = 为从发送和从接收使能时钟延长 (已使能时钟延长) 0 = 时钟延长被禁止

**注 1:** 如果 I<sup>2</sup>C 模块处于活动状态, 此位可能不能被置 1 (没有缓存), 并且也可能不会写入 SSPBUF (或者禁止写 SSPBUF)。



## 17.4.2 工作原理

通过将 MSSP 使能位 SSPEN (SSPCON1<5>) 置 1, 可使能 MSSP 模块。

SSPCON1 寄存器用于控制 I<sup>2</sup>C 的工作模式。可通过设置 4 个模式选择位 (SSPCON1<3:0>) 选择以下 I<sup>2</sup>C 模式之一:

- I<sup>2</sup>C 主模式, 时钟 = (Fosc/4) x (SSPADD + 1)
- I<sup>2</sup>C 从模式 (7 位地址)
- I<sup>2</sup>C 从模式 (10 位地址)
- I<sup>2</sup>C 从模式 (7 位地址), 允许启动位和停止位中断
- I<sup>2</sup>C 从模式 (10 位地址), 允许启动位和停止位中断
- I<sup>2</sup>C 由固件控制的主模式, 从器件空闲

通过将相应的 TRISC 或 TRISD 位置 1, 将 SCL 和 SDA 引脚编程为输入引脚; 在 SSPEN 位置 1 时选择任何 I<sup>2</sup>C 模式, 将强制上述引脚为漏极开路。要确保此模块正常工作, 必须为 SCL 和 SDA 引脚提供外接上拉电阻。

## 17.4.3 从模式

在从模式下, SCL 引脚和 SDA 引脚必须被配置为输入 (TRISC<4:3> 置 1)。必要时, MSSP 模块将使用输出数据改写输入状态 (从发送器)。

I<sup>2</sup>C 从模式硬件总是在地址匹配时产生中断。此外, 地址掩码还允许硬件在多个地址发生匹配时 (7 位寻址模式下最多 31 个地址, 10 位寻址模式下最多 63 个地址) 产生中断。用户也可以通过模式选择位, 选择启动位或停止位中断。

当地址匹配时或在地址匹配后发送的数据被接收时, 硬件会自动产生一个应答 (ACK) 脉冲, 并把当前 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:

- 在接收到传输的数据前, 缓冲器满标志位 BF (SSPSTAT<0>) 置 1。
- 在接收到传输的数据前, MSSP 溢出位 SSPOV (SSPCON1<6>) 置 1。

在这种情况下, SSPSR 寄存器的值不会被装入 SSPBUF, 但是 SSPIF 位会置 1。BF 位是通过读取 SSPBUF 寄存器清零的, 而 SSPOV 位通过软件清零。

为确保正常工作, SCL 时钟输入必须满足最小高电平和最小低电平时间要求。关于 I<sup>2</sup>C 规范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。

## 17.4.3.1 寻址

一旦使能了 MSSP 模块, 它就会等待启动条件出现。在启动条件出现后, 8 位数据被移入 SSPSR 寄存器。在时钟线 (SCL) 的上升沿采样所有的输入位。寄存器 SSPSR<7:1> 的值会和 SSPADD 寄存器的值进行比较。地址会在第 8 个时钟 (SCL) 脉冲的下降沿进行比较。如果地址匹配, 并且 BF 位和 SSPOV 位都被清零, 会发生下列事件:

1. SSPSR 寄存器的值被装入 SSPBUF 寄存器。
2. 缓冲器满标志位 BF 置 1。
3. 产生 ACK 脉冲。
4. 在第 9 个 SCL 脉冲的下降沿, MSSP 中断标志位 SSPIF 置 1 (如果允许中断, 则产生中断)。

在 10 位寻址模式下, 从器件需要接收两个地址字节。第一个地址字节的高五位指定这是否为 10 位地址。R/W 位 (SSPSTAT<2>) 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 位寻址的操作步骤如下, 其中 7-9 步是针对从发送器而言的:

1. 接收地址的第一个 (高) 字节 (SSPIF 位、BF 位和 UA 位 (SSPSTAT<1>) 置 1)。
2. 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)。
3. 读 SSPBUF 寄存器 (BF 位清零) 并将标志位 SSPIF 清零。
4. 接收地址的第二个 (低) 字节 (SSPIF 位、BF 位和 UA 位置 1)。
5. 使用地址的第一个 (高) 字节更新 SSPADD 寄存器。如果匹配的话就释放 SCL 线, 这将清零 UA 位。
6. 读 SSPBUF 寄存器 (BF 位清零) 并将标志位 SSPIF 清零。
7. 接收重复启动条件。
8. 接收地址的第一个 (高) 字节 (SSPIF 位和 BF 位置 1)。
9. 读 SSPBUF 寄存器 (BF 位清零) 并将标志位 SSPIF 清零。

# PIC18F85J11 系列

## 17.4.3.2 地址掩码

将地址的某一位掩码意味着该位可以为任意值。若将某个地址位掩码，则将应答两个地址并产生中断。由于可同时掩码多个地址位，所以在 7 位寻址模式中最多可应答 31 个地址，在 10 位寻址模式中最多可应答 63 个地址（见例 17-2）。

无论是否使用地址掩码，I<sup>2</sup>C 从器件的操作方式都不会改变。但是，当使用了地址掩码时，I<sup>2</sup>C 从器件可应答多个地址并产生中断。当出现这样的情况时，需要通过检查 SSPBUF 来确定是哪个地址引起的中断。

在 7 位寻址模式下，地址掩码位 ADMSK<5:1>（SSPCON2<5:1>）掩码 SSPADD 寄存器中相应的地址位。对于置 1 的任何 ADMSK 位（ADMSK<n> = 1），

都将忽略相应的地址位（SSPADD<n> = x）。要使模块发出地址应答，只要与没被掩码的地址位匹配就可以了。

在 10 位寻址模式下，ADMSK<5:2> 位将掩码 SSPADD 寄存器中相应的地址位。此外，ADMSK1 同时还掩码该地址低 2 位（SSPADD<1:0>）。对于任何置 1 的 ADMSK 位（ADMSK<n> = 1），将忽略相应的地址位（SSPADD<n> = x）。另请注意，虽然在 10 位寻址模式下，高地址位会与 SSPADD 寄存的某几位复用，但是地址掩码位不会影响这些位，而只影响低地址位。

- 注
- 1: ADMSK1 掩码该地址的低 2 位。

2: 地址掩码不会对该地址的高 2 位产生影响。

### 例 17-2: 地址掩码示例

#### 7 位寻址模式:

SSPADD<7:1> = A0h (1010000) (SSPADD<0> 假定为 0)

ADMSK<5:1> = 00111

应答的地址: A0h、A2h、A4h、A6h、A8h、AAh、ACh 和 AEh

#### 10 位寻址模式:

SSPADD<7:0> = A0h (10100000) (由于掩码不会对地址的高 2 位产生影响，所以在本示例中将其忽略)

ADMSK<5:1> = 00111

应答的地址: A0h、A1h、A2h、A3h、A4h、A5h、A6h、A7h、A8h、A9h、AAh、ABh、ACh、ADh、AEh 和 AFh

## 17.4.3.3 接收

当地址字节的  $\overline{R/\overline{W}}$  位清零且地址匹配时，SSPSTAT 寄存器的  $\overline{R/\overline{W}}$  位清零。接收的地址被装入 SSPBUF 寄存器，且 SDA 信号线保持低电平 ( $\overline{ACK}$ )。

当存在地址字节溢出时，会产生无应答脉冲 ( $\overline{ACK}$ )。溢出条件是指 BF 位 (SSPSTAT<0>) 置 1，或者 SSPOV 位 (SSPCON1<6>) 置 1。

每个数据传输字节都会产生一个 MSSP 中断。必须用软件将中断标志位 SSPIF 清零。使用 SSPSTAT 寄存器可以确定该字节的状态。

如果 SEN 被使能 (SSPCON2<0> = 1)，SCK/SCL 将在每个数据传输之后保持为低电平 (时钟延长)。必须将 CKP 位 (SSPCON1<4>) 置 1 才能释放时钟。更多详情，请参见第 17.4.4 节 “时钟延长”。

## 17.4.3.4 发送

当输入的地址字节的  $\overline{R/\overline{W}}$  位置 1 并发生地址匹配时，SSPSTAT 寄存器的  $\overline{R/\overline{W}}$  位置 1。接收到的地址被装入 SSPBUF 寄存器。 $\overline{ACK}$  脉冲在第 9 位上发送，同时不管 SEN 的值如何，RC3 引脚都将保持低电平 (如需了解更多细节，请参见第 17.4.4 节 “时钟延长”)。通过延长时钟，主器件只有在从器件准备好发送数据时才能发送另一个时钟脉冲。发送的数据必须装入 SSPBUF 寄存器，同时也被装入 SSPSR 寄存器。然后，通过将 CKP 位 (SSPCON1<4>) 置 1 来使能 RC3 引脚。8 个数据位在 SCL 时钟输入的下降沿被移出。这可确保在 SCL 为高电平期间 SDA 信号是有效的 (图 17-10)。

主接收器发出的  $\overline{ACK}$  脉冲将在第 9 个 SCL 输入脉冲的上升沿被锁存。如果 SDA 线为高电平 (无  $\overline{ACK}$  应答信号)，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  $\overline{ACK}$ ，将复位从逻辑，同时从器件监视下一个启动位的出现。如果 SDA 线为低电平 ( $\overline{ACK}$ )，则必须将下一个要发送的数据装入 SSPBUF 寄存器。同样，必须通过将 CKP 位置 1 来使能 RC3 引脚。

每个数据传输字节都会产生一个 MSSP 中断。SSPIF 位必须用软件清零，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。

图 17-8: I<sup>2</sup>C™ 从模式接收时序 (SEN = 0, 7 位地址)

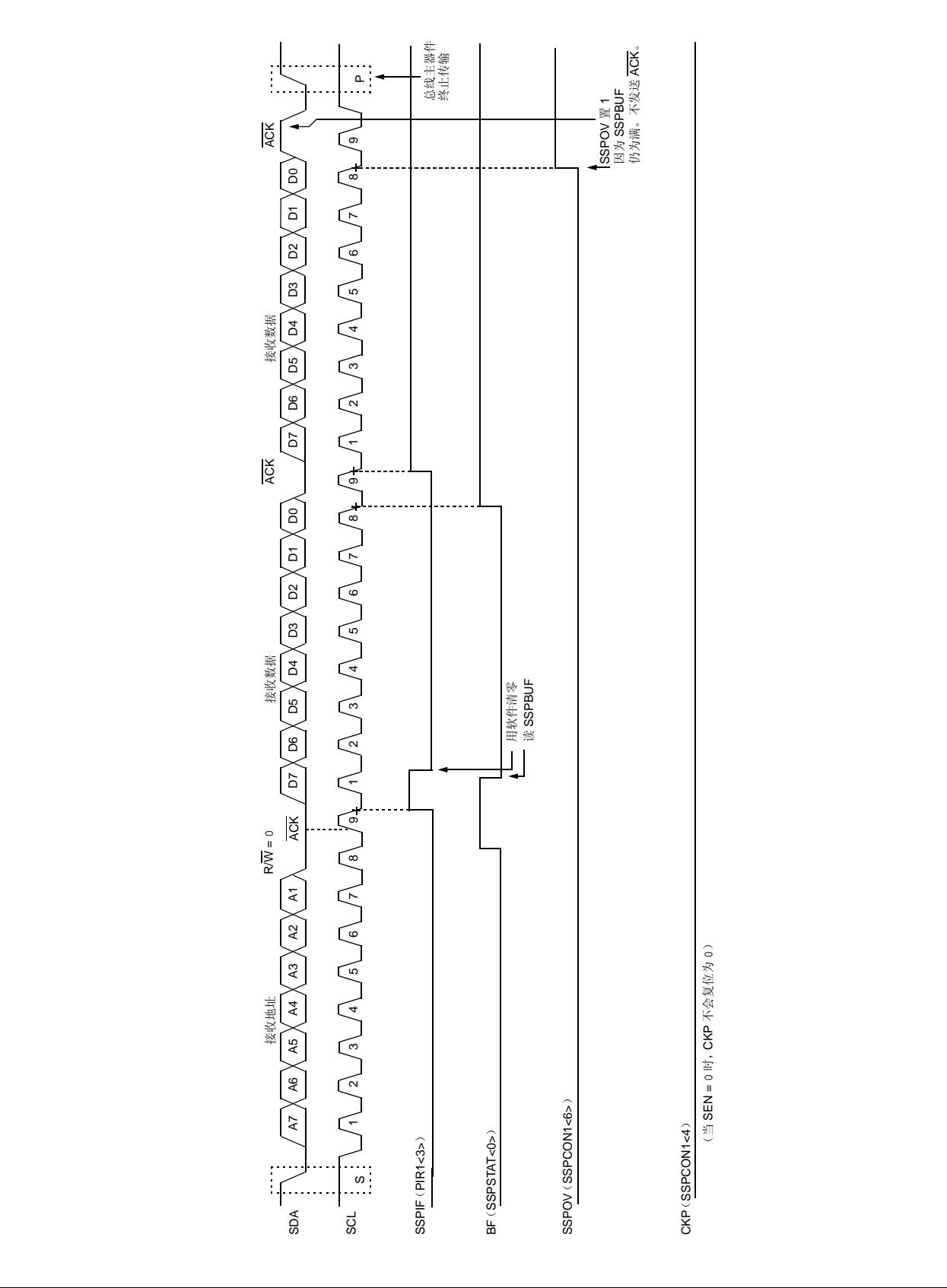


图 17-9: I<sup>2</sup>C™ 从模式接收时序 (SEN = 0 且 ADMSK<5:1> = 01011, 7 位地址)

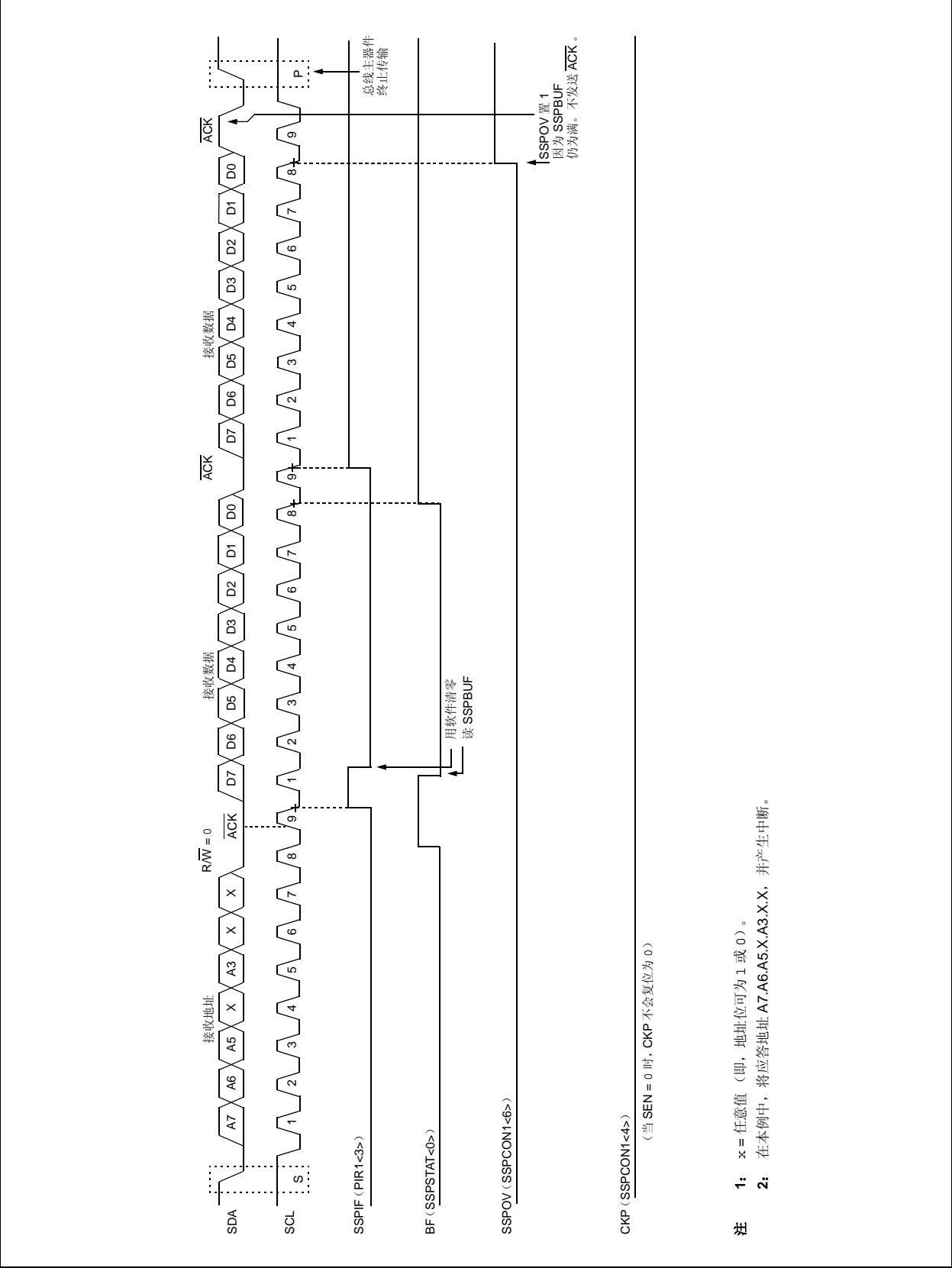


图 17-10: I<sup>2</sup>C™ 从模式发送时序 (7 位地址)

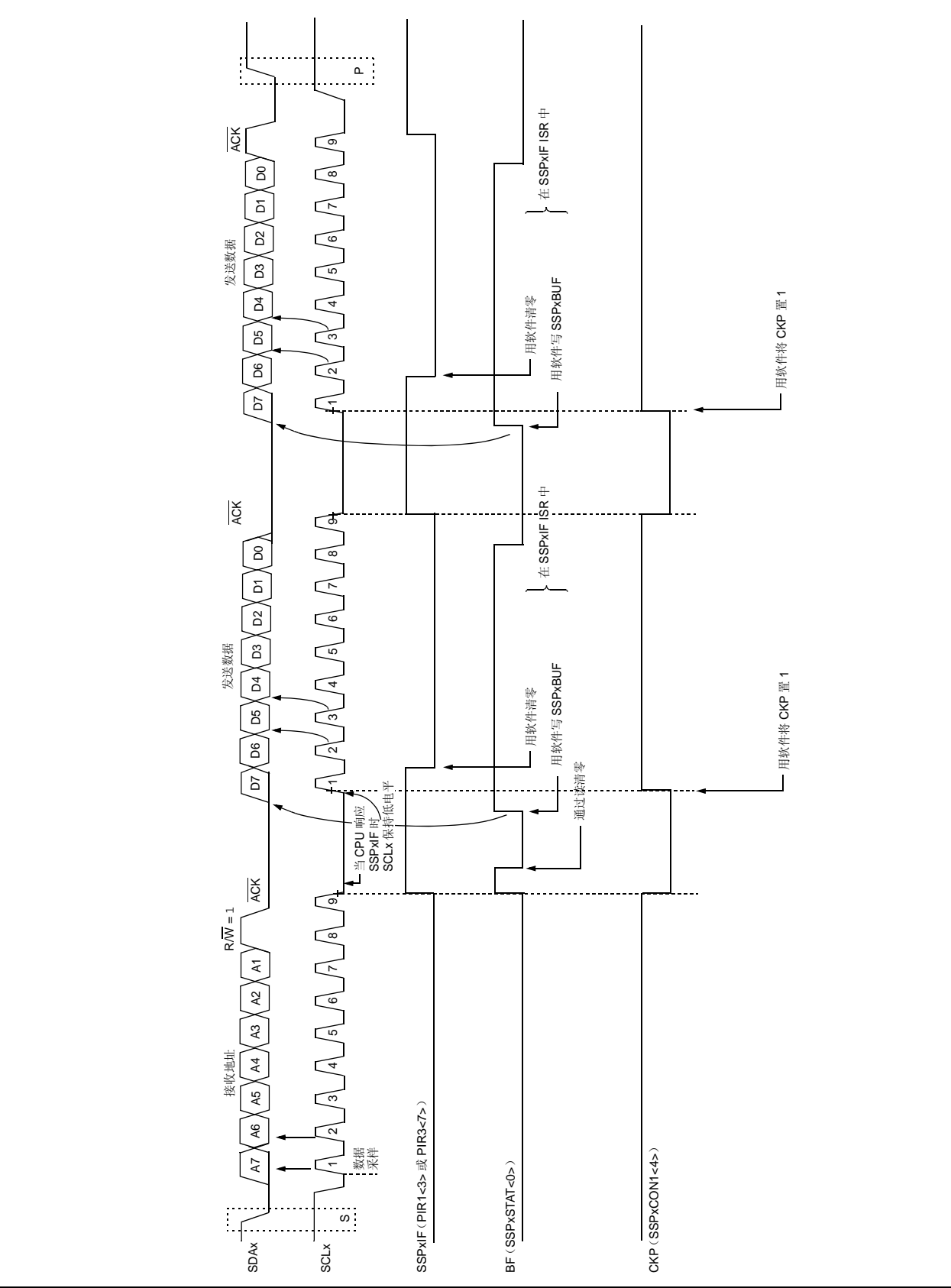
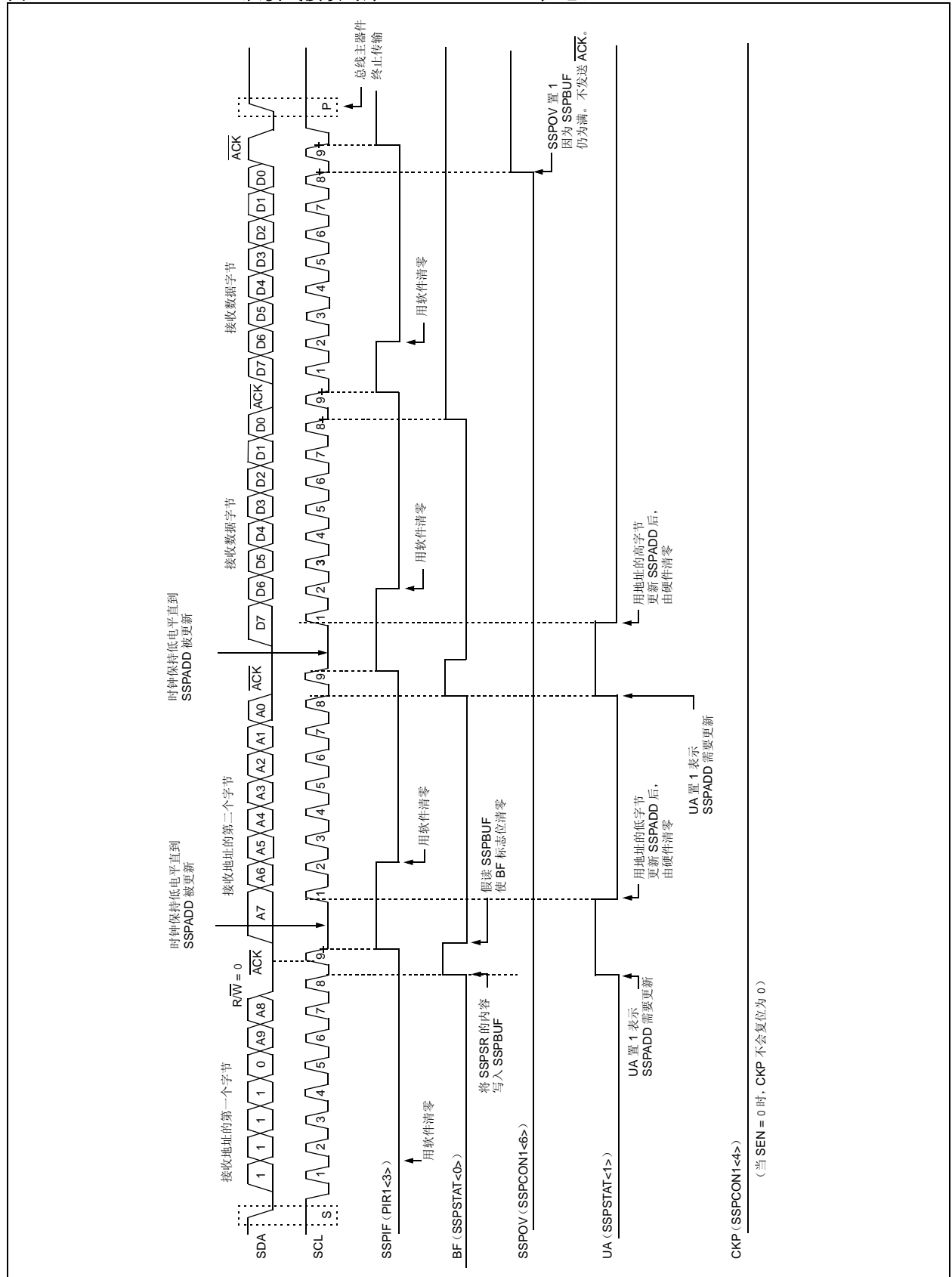
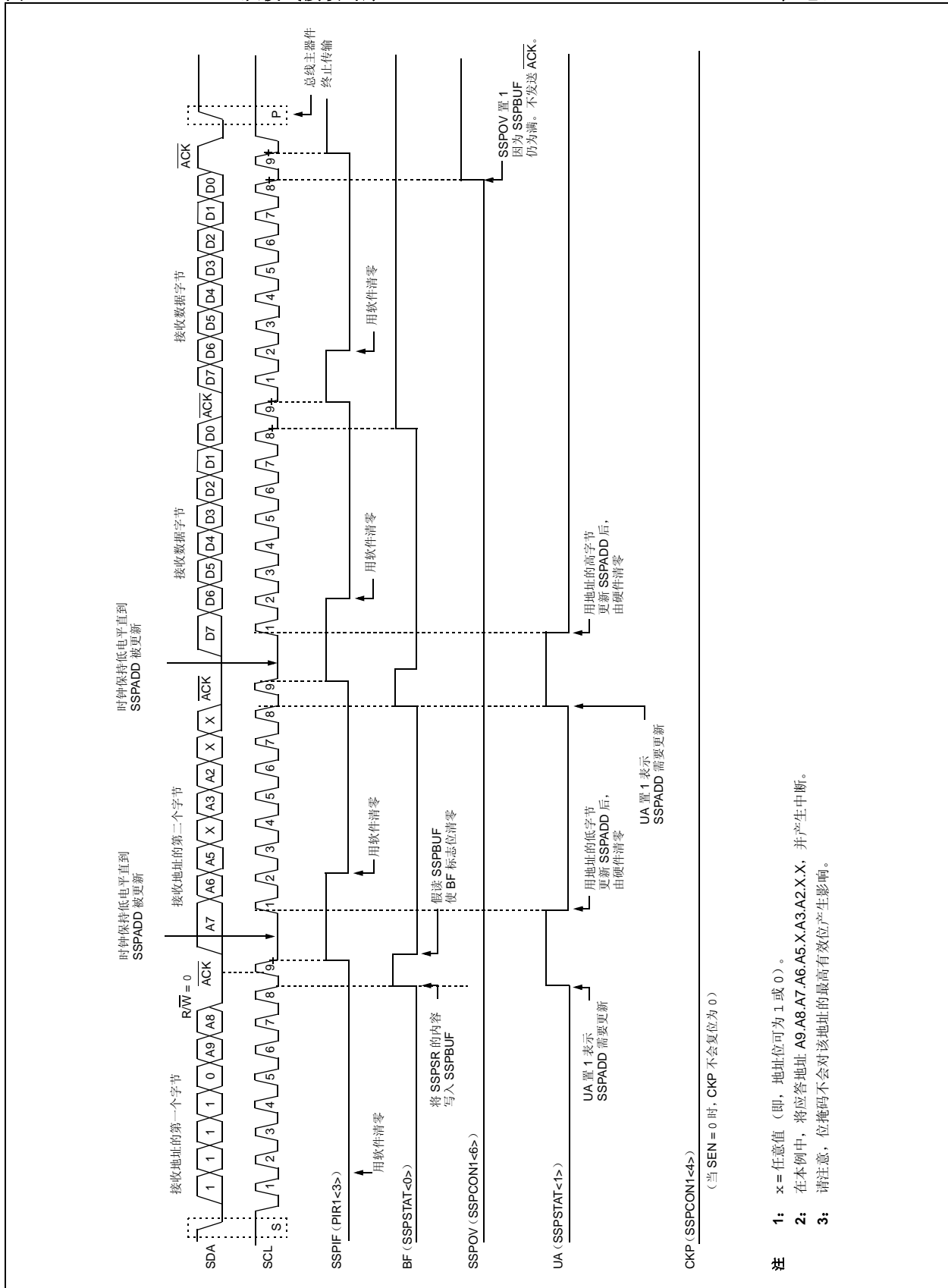


图 17-11: I<sup>2</sup>C™ 从模式接收时序 (SEN = 0, 10 位地址)

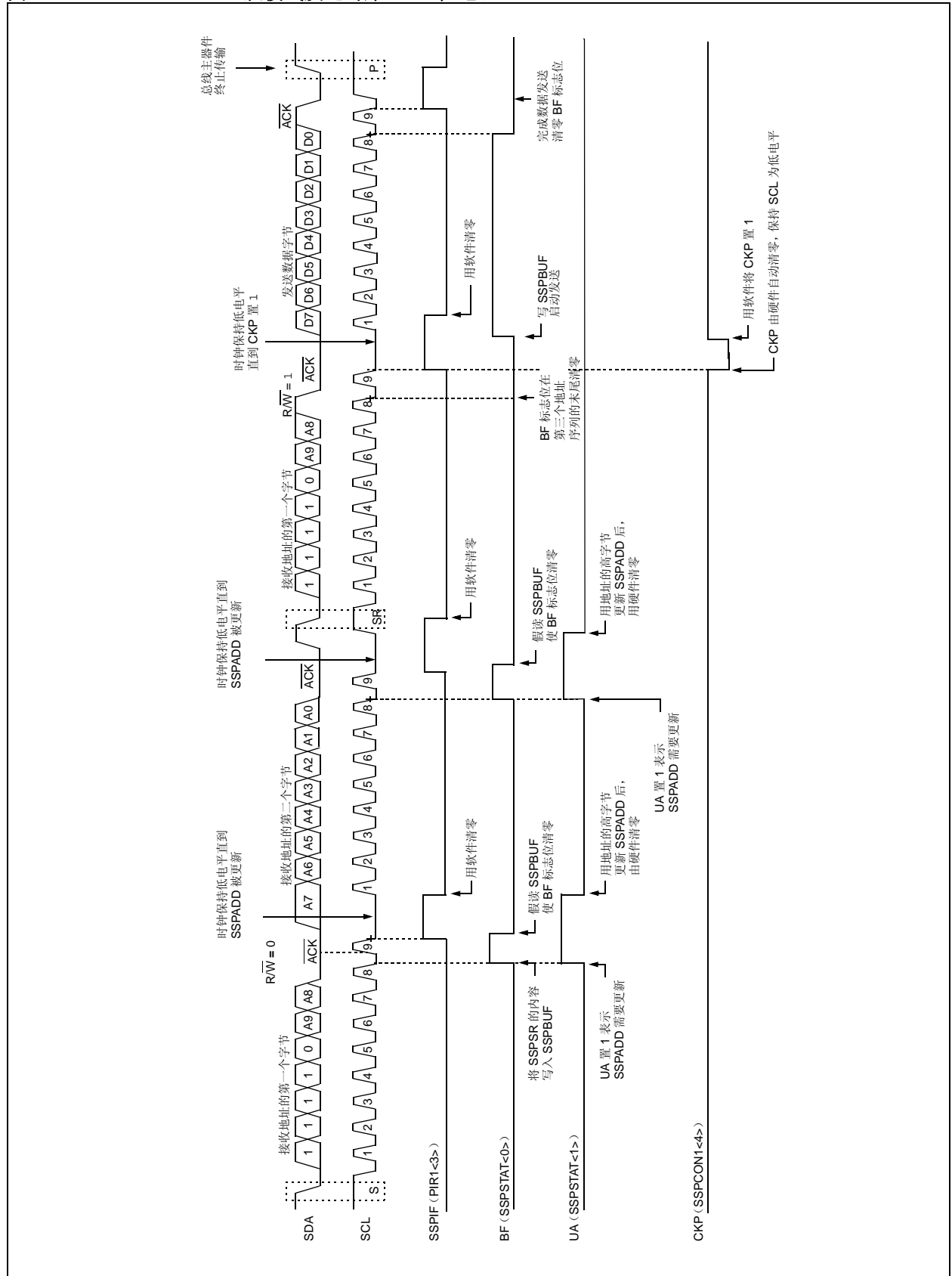


**图 17-12: I<sup>2</sup>C™ 从模式接收时序 (SEN = 0 且 ADMSK<5:1> = 01001, 10 位地址)**





**图 17-13:** **I<sup>2</sup>C™ 从模式发送时序 (10 位地址)**



# PIC18F85J11 系列

## 17.4.4 时钟延长

7位和10位从模式都在发送序列中实现了自动时钟延长。

SEN 位 (SSPCON2<0>) 允许在接收过程中使能时钟延长。将 SEN 置 1 会导致在每个数据接收序列的末尾将 SCL 引脚保持在低电平。

### 17.4.4.1 7 位从接收模式的时钟延长 (SEN = 1)

在 7 位从接收模式下，如果在  $\overline{\text{ACK}}$  序列末尾的第 9 个时钟的下降沿 BF 位置 1，则 SSPCON1 寄存器中的 CKP 位就会自动清零，强制 SCL 输出保持低电平。CKP 位被清为 0 会将 SCL 线拉为低电平。在允许继续接收之前，必须在用户的中断服务程序中将 CKP 位置 1。保持 SCL 线为低电平，让用户在主器件发起另一个接收序列之前，有时间处理中断服务程序并读取 SSPBUF 的内容。这将防止发生缓冲器溢出（见图 17-15）。

- 注 1:** 如果用户在第 9 个时钟的下降沿前读取了 SSPBUF 的内容而导致 BF 位被清零，则 CKP 位将不会被清零，也不会产生时钟延长。
- 2:** 不管 BF 位的状态如何，CKP 位都可以用软件置 1。在下一个接收序列开始之前，用户要注意在中断服务程序中清零 BF 位，以避免溢出。

### 17.4.4.2 10 位从接收模式的时钟延长 (SEN = 1)

在 10 位从接收模式下，在地址序列中会自动发生时钟延长，但是 CKP 位不会被清零。在这期间，如果 UA 位在第 9 个时钟之后置 1，就会启动时钟延长。在接收到 10 位地址的高字节后，UA 位被置 1，然后接收 10 位地址的第二个字节且 R/W 位清为 0。在更新 SSPADD 的时候释放时钟线。如同 7 位模式那样，在每个数据接收序列中会发生时钟延长。

**注:** 如果用户在第 9 个时钟的下降沿出现之前查询 UA 位，并通过更新 SSPADD 寄存器清零 UA 位，而且在此之前用户没有读取 SSPBUF 寄存器使 BF 位清零，则 CKP 位的电平仍然不会被拉低。基于 BF 位状态的时钟延长仅在传输数据的序列中出现，不会出现在地址序列中。

### 17.4.4.3 7 位从发送模式的时钟延长

如果 BF 位被清零，7 位从发送模式将通过在第 9 个时钟的下降沿出现后清零 CKP 位，来实现时钟延长。上述情形与 SEN 位的状态无关。

用户的中断服务程序必须先将 CKP 位置 1 才可以继续发送。保持 SCL 线为低电平，让用户在主器件发起另一个发送序列之前，有时间处理中断服务程序并装入 SSPBUF 的内容（见图 17-10）。

- 注 1:** 如果用户在第 9 个时钟的下降沿之前就装入 SSPBUF 的内容，使 BF 位置 1，CKP 位就不会被清零，也不会发生时钟延长。
- 2:** 不管 BF 位的状态如何，CKP 位都可以用软件置 1。

### 17.4.4.4 10 位从发送模式的时钟延长

在 10 位从发送模式下，在前两个地址序列中由 UA 位的状态来控制时钟延长，正如同 10 位从接收模式一样。前两个地址后跟着第三个地址序列，该地址序列包含 10 位地址的高位和被置 1 的 R/W 位。在执行完第三个地址序列后，UA 位不置 1，此时模块被配置为发送模式，BF 标志位控制时钟延长，正如 7 位从发送模式一样（见图 17-13）。

## 17.4.4.5 时钟同步和 CKP 位

当 CKP 位被清零时，SCL 输出被强制为 0。然而，清零 CKP 位不会将 SCL 输出拉为低电平，除非已经采样到 SCL 输出为低电平。因此，CKP 位将不会把 SCL 线拉为低电平，除非外部 I<sup>2</sup>C 主器件已将 SCL 线拉低。

SCL 输出将保持低电平，直到 CKP 位置 1 且 I<sup>2</sup>C 总线上的所有其他器件将 SCL 电平拉高为止。这可以确保对 CKP 位的写操作不会违反 SCL 的最小高电平时间要求（见图 17-14）。

图 17-14: 时钟同步时序

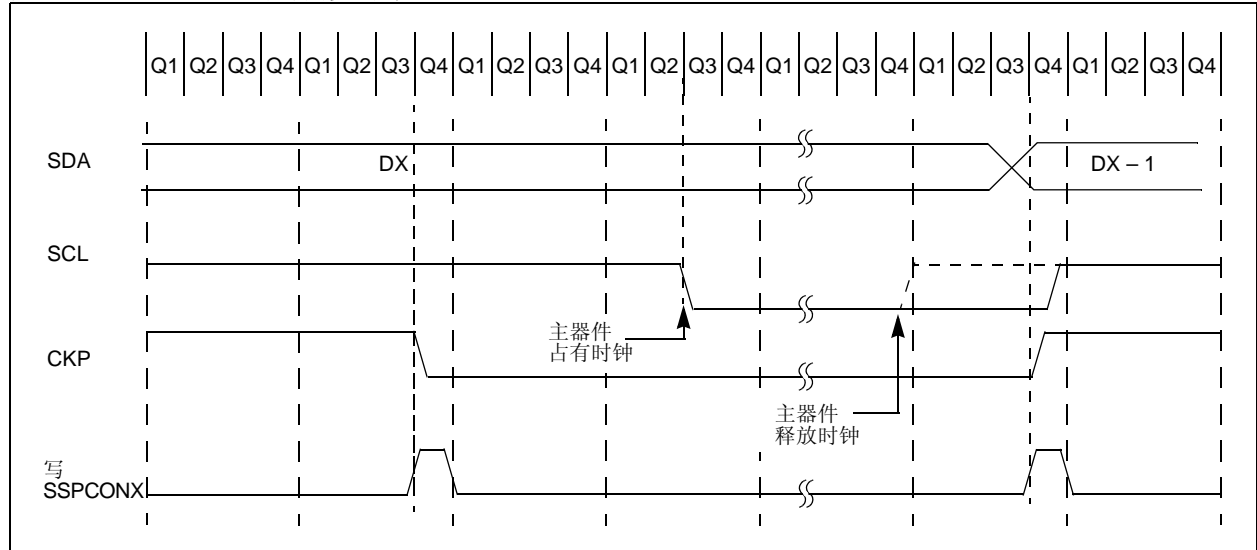


图 17-15: I<sup>2</sup>C™ 从模式接收时序 (SEN = 1, 7 位地址)

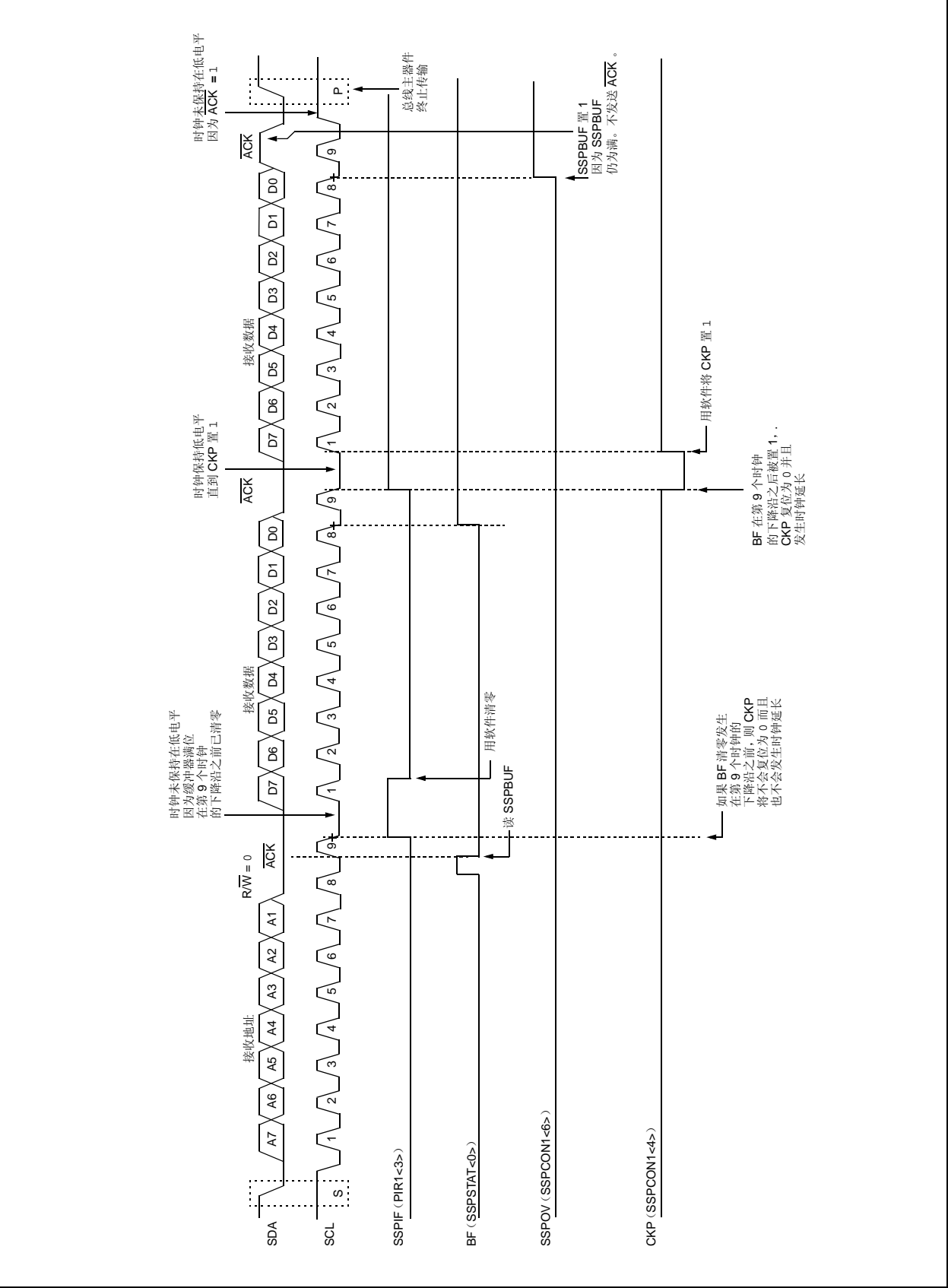
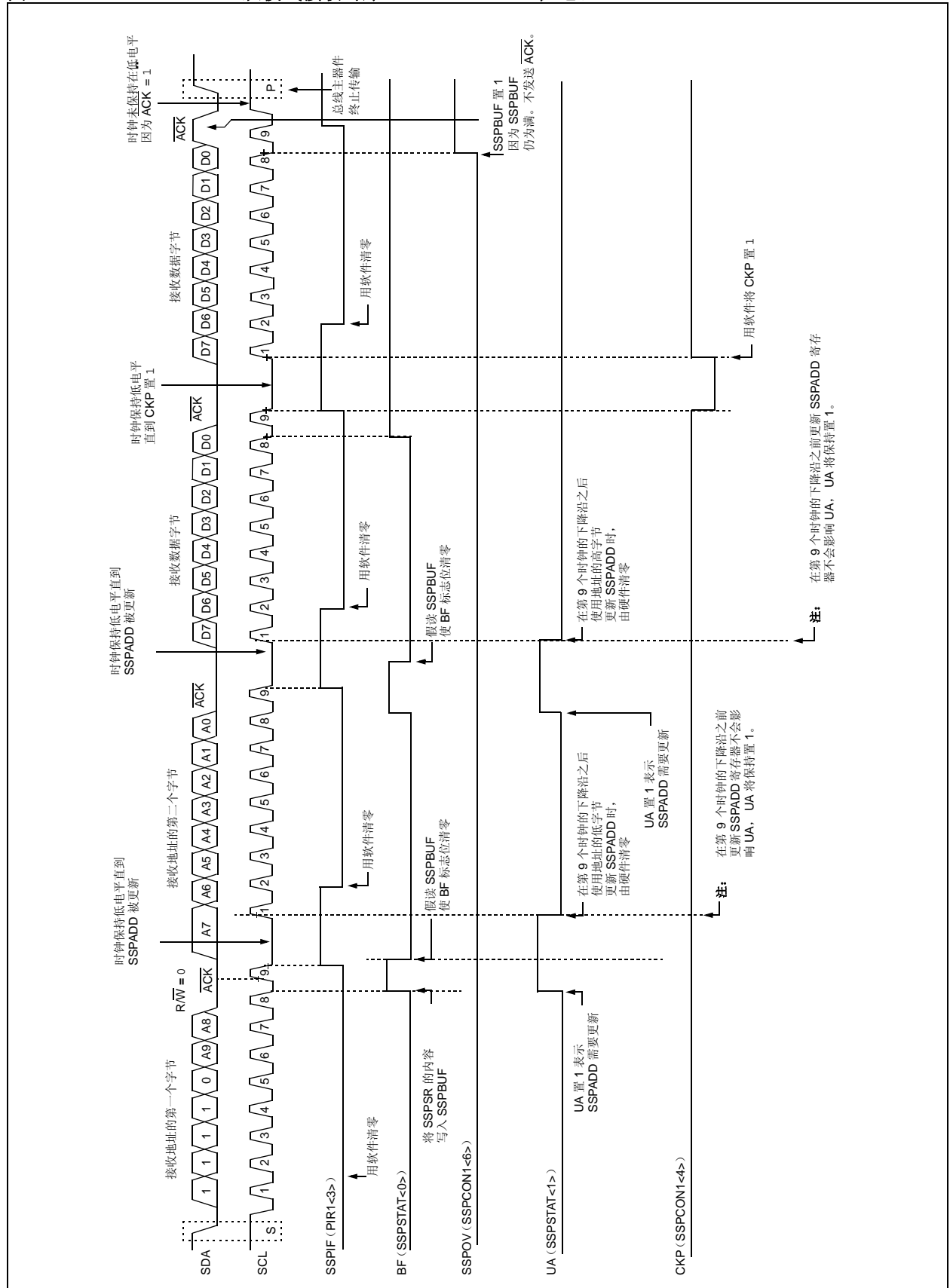


图 17-16: I<sup>2</sup>C™ 从模式接收时序 (SEN = 1, 10 位地址)



## 17.4.5 支持广播呼叫地址

在 I<sup>2</sup>C 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信号来响应。

广播呼叫地址是由 I<sup>2</sup>C 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。

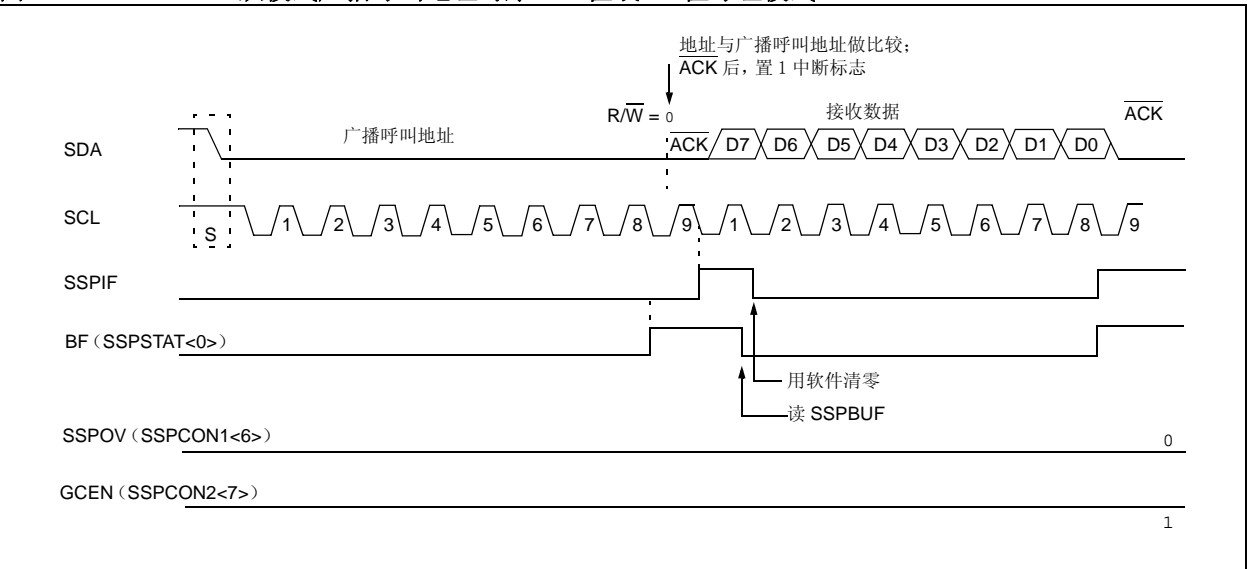
当使能广播呼叫使能位 GCEN (SSPCON2<7> 置 1) 时，即可识别广播呼叫地址。检测到启动位后，8 位数据会被移入 SSPSR，同时将该地址与 SSPADD 的内容进行比较。它还会与广播呼叫地址进行比较并用硬件设定。

如果与广播呼叫地址匹配，SSPSR 的值将被传输到 SSPBUF，BF 标志位（第 8 位）置 1，并且在第 9 位（ACK 位）的下降沿 SSPIF 中断标志位置 1。

当响应中断时，可以通过读取 SSPBUF 的内容来检查中断源。该值可以用于判断是特定器件的地址还是一个广播呼叫地址。

在 10 位寻址模式下，需要更新 SSPADD 以匹配地址的后半部分，同时 UA 位 (SSPSTAT<1>) 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 位寻址模式，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（图 17-17）。

图 17-17: 从模式广播呼叫地址时序（7 位或 10 位寻址模式）



## 17.4.6 主模式

通过将 SSPCON1 中的相应 SSPM 位置 1 和清零，同时将 SSPEN 位置 1，可以使能主模式。在主模式下，SCL 和 SDA 信号线由 MSSP 硬件控制。

主模式通过在检测到启动和停止条件时产生中断来工作。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，可以获得 I<sup>2</sup>C 总线的控制权；否则 P 位和 S 位都清零，总线处于空闲状态。

在固件控制的主模式下，用户代码根据启动和停止位条件执行所有的 I<sup>2</sup>C 总线操作。

一旦使能主模式，用户即可选择以下 6 项操作：

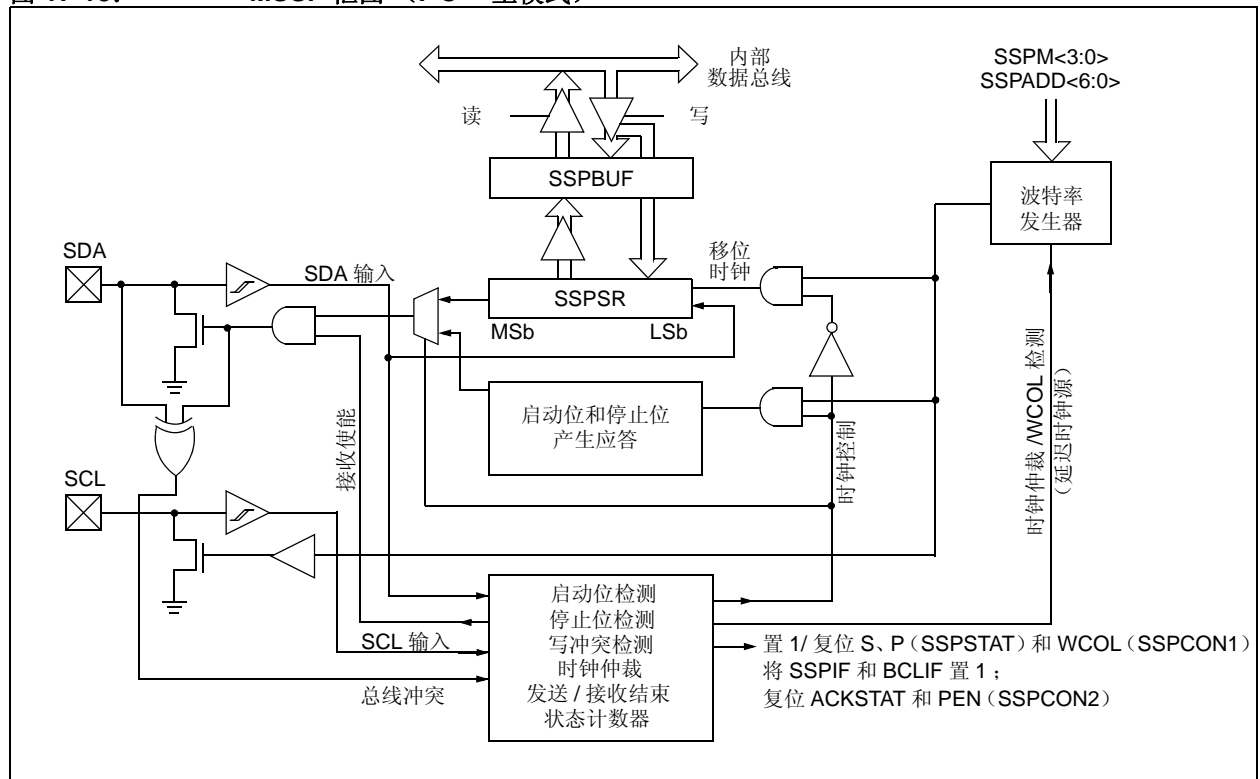
1. 在 SDA 和 SCL 上发出启动条件。
2. 在 SDA 和 SCL 上发出重复启动条件。
3. 写入 SSPBUF 寄存器，启动数据 / 地址的发送。
4. 配置 I<sup>2</sup>C 端口用于接收数据。
5. 在接收到数据字节后产生应答条件。
6. 在 SDA 和 SCL 上产生停止条件。

**注：** 当被配置为 I<sup>2</sup>C 主模式时，MSSP 模块不允许事件排队。例如，在启动条件结束前，不允许用户立即写 SSPBUF 寄存器以启动传输。在这种情况下，将不会写 SSPBUF，WCOL 位将被置 1，这表明没有发生对 SSPBUF 的写操作。

下列事件会使 MSSP 中断标志位 SSPIF 置 1（如果允许 MSSP 中断，则产生该中断）：

- 启动条件
- 停止条件
- 数据传输字节发送 / 接收
- 发送应答
- 重复启动

图 17-18: MSSP 框图 (I<sup>2</sup>C™ 主模式)



## 17.4.6.1 I<sup>2</sup>C 主模式工作原理

主器件产生所有串行时钟脉冲和启动 / 停止条件。以停止条件或重复启动条件结束传输。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 I<sup>2</sup>C 总线。

在主发送器模式下，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作为接收方的从器件地址（7 位）和读 / 写（R/W）位。在这种情况下，R/W 位将是逻辑 0。一次发送 8 位串行数据。每发送一个字节，都会收到一个应答位。输出启动和停止条件来指示串行传输的开始和结束。

在主接收模式下，发送的第一个字节包括作为发送方的从器件地址（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代表发送的开始和结束。

在 I<sup>2</sup>C 模式下，在 SPI 模式中使用的波特率发生器用于将 SCL 时钟频率设置为 100 kHz、400 kHz 或 1 MHz。更多详情请参见第 17.4.7 节“波特率”。

下面是一个典型的发送序列：

1. 用户通过将启动使能位 SEN（SSPCON2<0>）置 1，产生启动条件。
2. SSPIF 置 1。在进行下一步操作前，MSSP 模块将等待所需的启动时间。
3. 用户将从器件地址装入 SSPBUF 进行发送。
4. 地址从 SDA 引脚移出，直到发送完所有 8 位地址数据。
5. MSSP 模块移入从器件的 ACK 位，并将它的值写入 SSPCON2 寄存器（SSPCON2<6>）。
6. MSSP 模块在第 9 个时钟周期的末尾通过将 SSPIF 位置 1，产生一个中断。
7. 用户将 8 位数据装入 SSPBUF。
8. 数据从 SDA 引脚移出，直到发送完所有 8 位数据。
9. MSSP 模块移入从器件的 ACK 位，并将它的值写入 SSPCON2 寄存器（SSPCON2<6>）。
10. MSSP 模块在第 9 个时钟周期的末尾通过将 SSPIF 位置 1，产生一个中断。
11. 用户通过将停止使能位 PEN（SSPCON2<2>）置 1，产生停止条件。
12. 一旦停止条件完成，将产生一个中断。



## 17.4.7 波特率

在 I<sup>2</sup>C 主模式下，波特率发生器（Baud Rate Generator, BRG）的重载值位于 SSPADD 寄存器的低 7 位（图 17-19）。当发生对 SSPBUF 的写操作时，波特率发生器将自动开始计数。BRG 会递减计数到 0，然后停止，直到再次发生重载。BRG 的值会在每个指令周期（Tcy）中的 Q2 和 Q4 时钟周期上进行两次减计数。在 I<sup>2</sup>C 主模式下，会自动重载 BRG。

如果指定操作完成（即，在传输的最后一个数据位后传输 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。

表 17-3 显示了不同的指令周期下的时钟频率以及装入 SSPADD 的 BRG 值。不支持 SSPADD BRG 值为 0x00。

### 17.4.7.1 功耗管理模式下的波特率发生

器件在某种功耗管理模式下工作时，BRG 的时钟源频率可能会发生变化甚至停止，具体取决于选择的模式和时钟源。从辅助时钟或内部振荡器切换到运行或空闲模式可能会更改 BRG 的时钟频率。在休眠模式下，将不对 BRG 提供时钟。

图 17-19: 波特率发生器框图

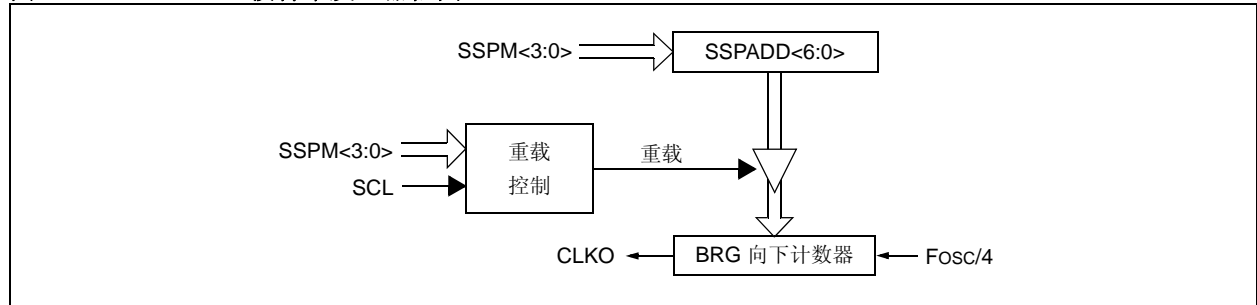


表 17-3: 使用 BRG 的 I<sup>2</sup>C™ 时钟频率

Fcy	Fcy * 2	BRG 值	Fscl (两次 BRG 计满返回)
10 MHz	20 MHz	18h	400 kHz <sup>(1)</sup>
10 MHz	20 MHz	1Fh	312.5 kHz
10 MHz	20 MHz	63h	100 kHz
4 MHz	8 MHz	09h	400 kHz <sup>(1)</sup>
4 MHz	8 MHz	0Ch	308 kHz
4 MHz	8 MHz	27h	100 kHz
1 MHz	2 MHz	02h	333 kHz <sup>(1)</sup>
1 MHz	2 MHz	09h	100 kHz

注 1: 虽然 I<sup>2</sup>C 接口各方面都不符合 400 kHz I<sup>2</sup>C 规范（该规范适用于大于 100 kHz 的频率），但在需要较高频率的应用场合可以慎重使用。

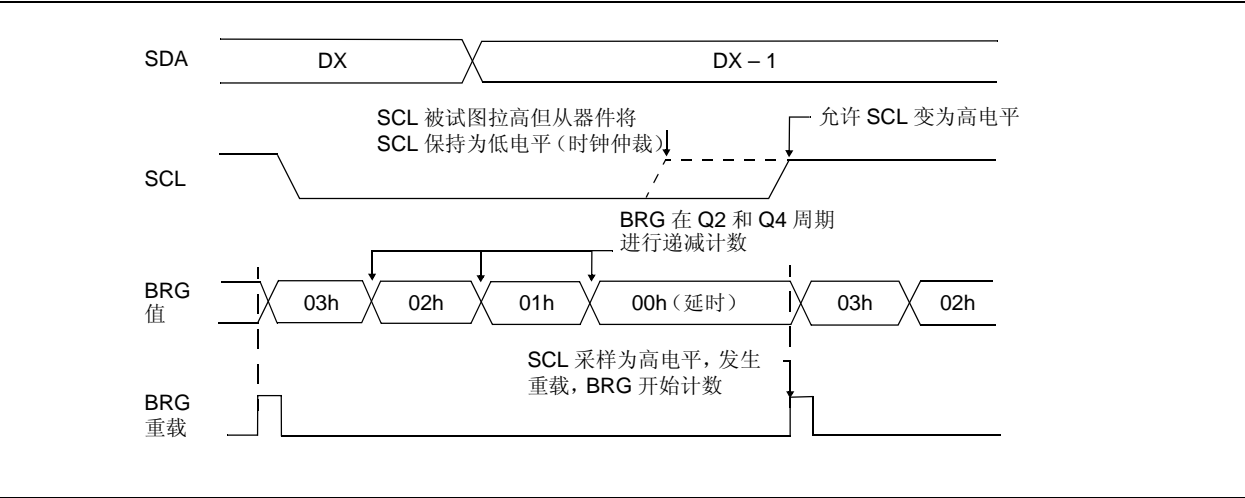
# PIC18F85J11 系列

## 17.4.7.2 时钟仲裁

如果在任何接收、发送或重复启动 / 停止条件过程中，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，波特率发生器将重新装入 SSPADD<6:0> 的值并开始计数。这可以保证当外部器件将时钟拉低时，SCL 始终在至少一个 BRG 计满返回周期内保持高电平（图 17-20）。

图 17-20: 带有时钟仲裁的波特率发生器时序



## 17.4.8 I<sup>2</sup>C 主模式启动条件时序

要产生启动条件，用户应将启动使能位 SEN (SSPCON2<0>) 置 1。当 SDA 和 SCL 引脚采样为高电平时，波特率发生器重新装入 SSPADD<6:0> 的值并开始计数。当波特率发生器发生超时 (TBRG) 时，如果 SCL 和 SDA 都被采样为高电平，则 SDA 引脚将被驱动为低电平。当 SCL 为高电平时将 SDA 驱动为低电平将产生启动条件，并使 S 位 (SSPSTAT<3>) 置 1。随后波特率发生器重新装入 SSPADD<6:0> 的值并恢复计数。当波特率发生器再次发生超时 (TBRG) 时，SEN 位 (SSPCON2<0>) 将自动由硬件清零。波特率发生器暂停工作，SDA 线保持低电平，启动条件结束。

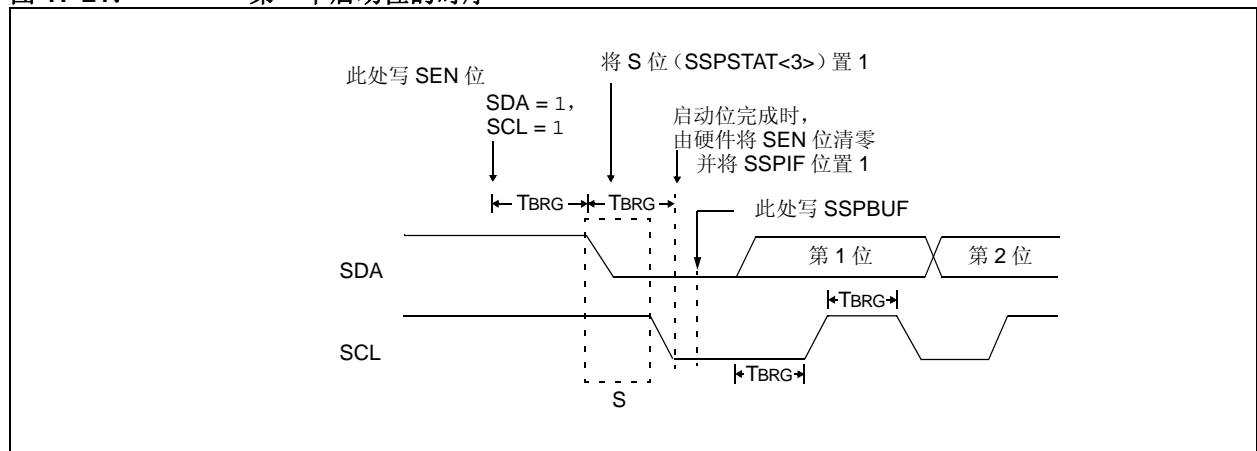
**注：** 如果在启动条件开始时，SDA 和 SCL 引脚已采样为低电平，或者在启动条件期间，SCL 线在 SDA 线被驱动为低电平之前已经采样为低电平，则会产生总线冲突。总线冲突中断标志位 BCLIF 置 1，启动条件中止，I<sup>2</sup>C 模块复位为空闲状态。

### 17.4.8.1 WCOL 状态标志

在启动时序进行当中，如果用户写 SSPBUF，则 WCOL 被置 1，同时缓冲器内容不变（写操作无效）。

**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 17-21: 第一个启动位的时序



## 17.4.9 I<sup>2</sup>C 主模式重复启动条件时序

将 RSEN 位 (SSPCON2<1>) 编程为高电平, 并且 I<sup>2</sup>C 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, 波特率发生器将装入 SSPADD<6:0> 的值, 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 (引脚电平被拉高)。当波特率发生器超时, 如果 SDA 被采样为高电平, SCL 引脚将被释放 (引脚电平被拉高)。当 SCL 被采样为高电平时, 波特率发生器重新装入 SSPADD<6:0> 的值并开始计数。SDA 和 SCL 必须在一个 TBRG 内被采样为高电平。接下来, 在一个 TBRG 中将 SDA 引脚拉为低电平 (SDA = 0), 同时 SCL 保持高电平。然后 RSEN 位 (SSPCON2<1>) 将自动清零, 波特率发生器不会被重载并且 SDA 引脚保持低电平。一旦在 SDA 和 SCL 引脚上检测到启动条件, 就会置 1 S 位 (SSPSTAT<3>)。直到波特率发生器发生超时后, SSPIF 位才会置 1。

- 注 1:** 有任何其他事件在进行时, 编程设置对 RSEN 无效。
- 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, 采样到 SDA 为低电平。
  - 在 SDA 被拉低之前, SCL 变为低电平。这表示另一个主器件正尝试发送数据 1。

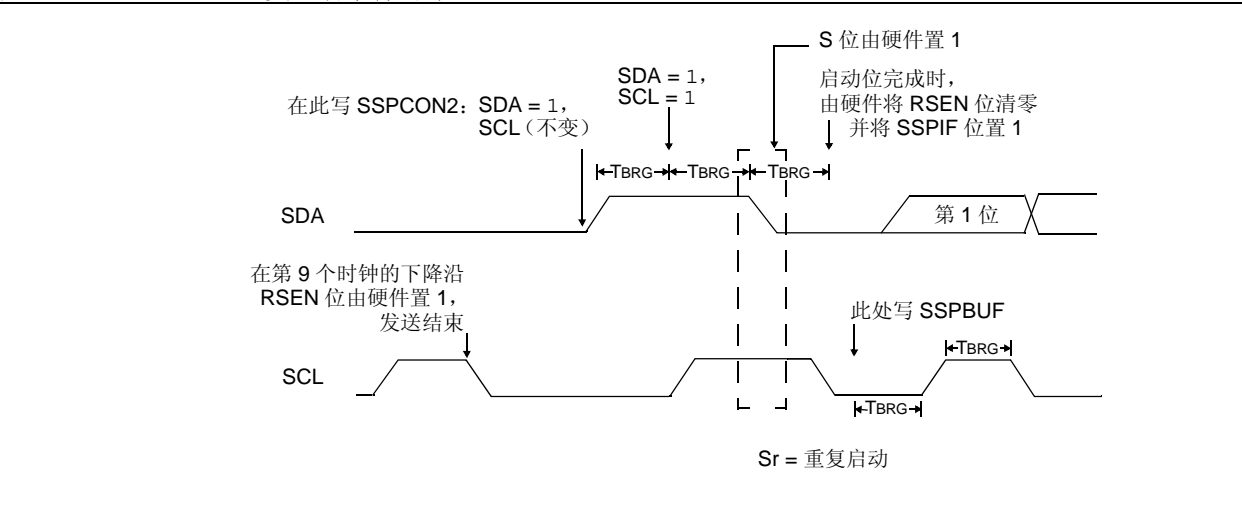
一旦 SSPIF 位被置 1, 用户便可以在 7 位寻址模式下将 7 位地址写入 SSPBUF, 或者在 10 位寻址模式下将默认的 1 个地址字节写入 SSPBUF。当发送完第 1 个 8 位数据并接收到一个 ACK 后, 用户可以发送另外 8 位地址 (10 位寻址模式) 或 8 位数据 (7 位寻址模式)。

### 17.4.9.1 WCOL 状态标志

在重复启动时序进行当中, 如果用户写 SSPBUF, 则 WCOL 被置 1, 同时缓冲器内容不变 (写操作无效)。

**注:** 由于不允许事件排队, 在重复启动条件完成之前, 不能对 SSPCON2 的低 5 位进行写操作。

图 17-22: 重复启动条件时序



## 17.4.10 I<sup>2</sup>C 主模式发送

发送一个数据字节、一个 7 位地址或一个 10 位地址的一半，都可以通过写一个值到 SSPBUF 寄存器来实现。该操作将使缓冲器满标志位 BF 置 1，波特率发生器开始计数，同时启动下一次发送。在出现 SCL 的下降沿后（见数据保持时间规范参数 106），地址/数据的每一位被移出至 SDA 引脚。在一个波特率发生器计满返回周期（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，它将在一个 TBRG 内保持为高电平。在此期间以及 SCL 的下一个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 位数据被移出（第 8 个时钟周期的下降沿）之后，BF 标志位被清零，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 位时以一个 ACK 位做出响应。ACK 的状态在第 9 个时钟的下降沿写入 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 个时钟周期之后，SSPIF 位会置 1，主时钟（波特率发生器）暂停，直到下一个数据字节装入 SSPBUF。SCL 引脚保持低电平，并且 SDA 保持不变（图 17-23）。

在写 SSPBUF 之后，地址的每一位在 SCL 的下降沿被移出，直至所有 7 位地址和 R/W 位都被移出。在第 8 个时钟的下降沿，主器件将释放 SDA 引脚，以允许从器件发出一个应答响应。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（SSPCON2<6>）位。在发送地址的第 9 个时钟下降沿之后，SSPIF 位置 1，BF 标志位清零，波特率发生器关闭直到下一次写 SSPBUF，同时 SCL 引脚保持低电平并且允许 SDA 引脚悬空。

### 17.4.10.1 BF 状态标志

在发送模式下，BF 位（SSPSTAT<0>）在 CPU 写 SSPBUF 时置 1，在所有 8 位数据移出后清零。

### 17.4.10.2 WCOL 状态标志

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 位置 1 并且在写入 SSPBUF 之后的 2 个 Tcy 内缓冲器内容不变（写操作无效）。如果在 2 个 Tcy 内重写 SSPBUF，则 WCOL 位置 1 并且更新 SSPBUF。这可能导致传输被破坏。

在每次写 SSPBUF 之后，用户应该验证 WCOL 位是否清零以确保传输是正确的。在所有情况下，WCOL 必须用软件清零。

### 17.4.10.3 ACKSTAT 状态标志

在发送模式下，当从器件发送应答（ $\overline{ACK} = 0$ ）时，ACKSTAT 位（SSPCON2<6>）清零；当从器件没有应答（ $\overline{ACK} = 1$ ）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。

## 17.4.11 I<sup>2</sup>C 主模式接收

通过编程接收使能位 RCEN（SSPCON2<3>）使能主模式接收。

**注：** RCEN 位置 1 前，MSSP 模块必须处于空闲状态，否则对 RCEN 位的置 1 操作将无效。

波特率发生器开始计数，每次计满返回时，SCL 引脚的状态发生改变（由高变低或由低变高），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 并且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<4>）置 1，用户可以在接收结束后发送应答位。

### 17.4.11.1 BF 状态标志

接收过程中，当将地址或数据字节从 SSPSR 装入 SSPBUF 时，BF 位置 1；在读 SSPBUF 寄存器时清零。

### 17.4.11.2 SSPOV 状态标志

接收过程中，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。

### 17.4.11.3 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 并且缓冲器内容不变（写操作无效）。

图 17-23: I<sup>2</sup>C™ 主模式发送时序 (7 位或 10 位地址)

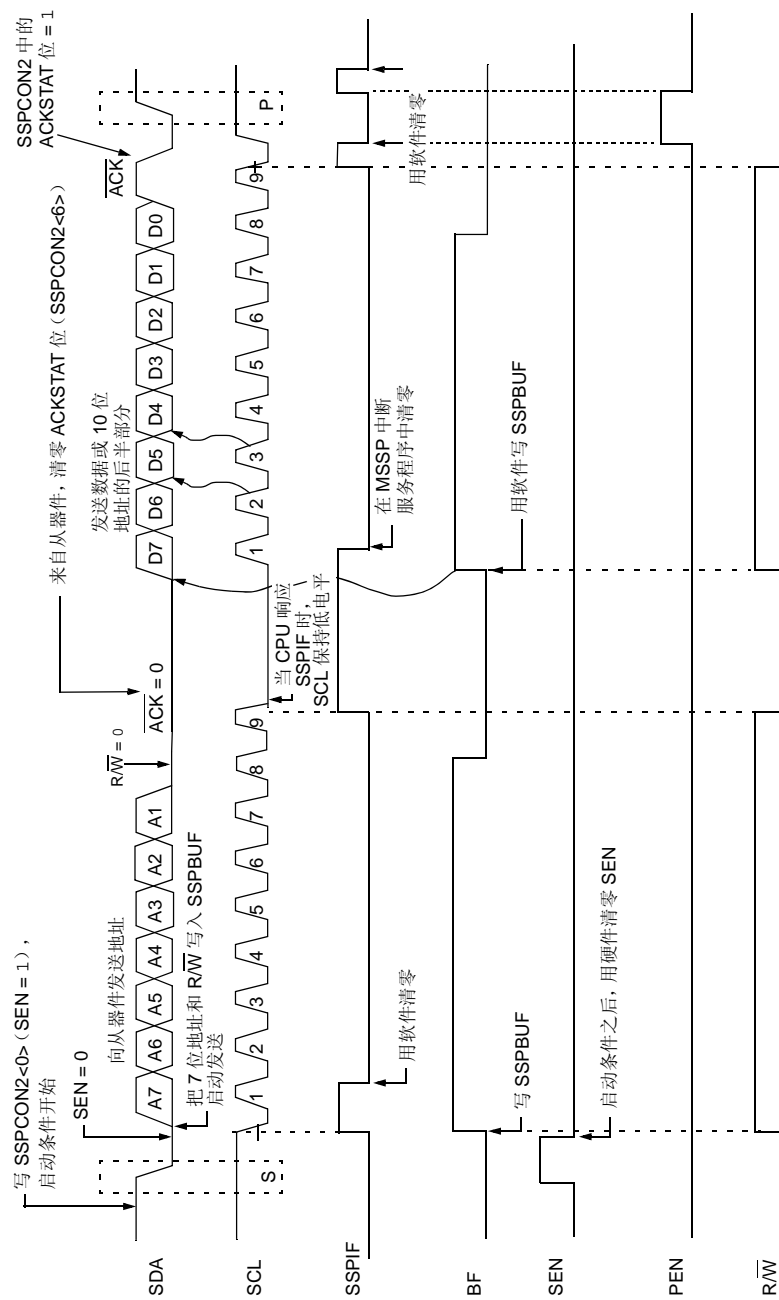
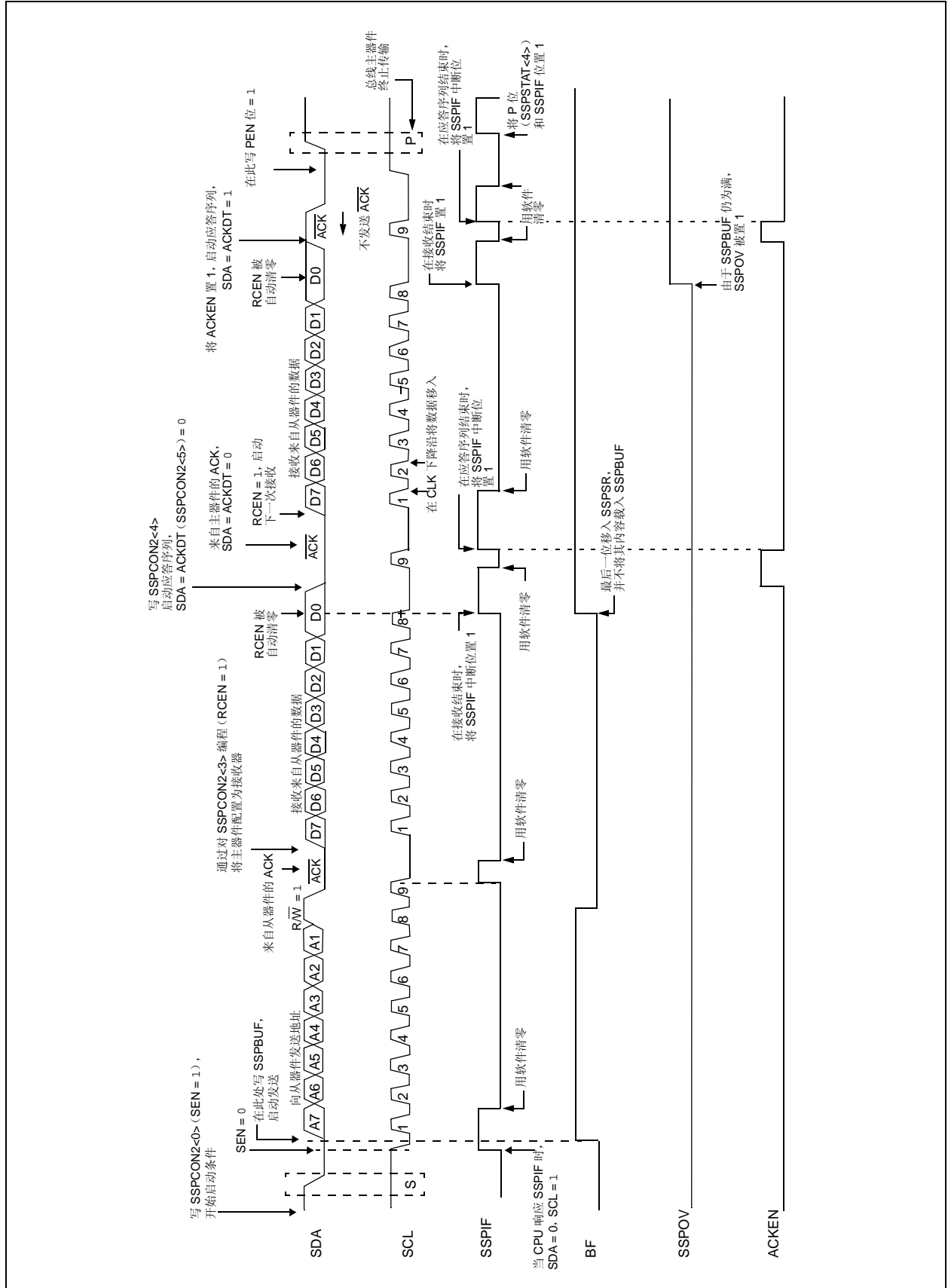


图 17-24: I<sup>2</sup>C™ 主模式接收时序 (7 位地址)



## 17.4.12 应答序列时序

将应答序列使能位 **ACKEN** (**SSPCON2<4>**) 置 1 即可使能应答序列。当该位被置 1 时, **SCL** 引脚被拉低, 应答数据位的内容输出到 **SDA** 引脚。如果用户希望产生一个应答, 则应该将 **ACKDT** 位清零。否则, 用户应在应答序列开始前将 **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** 引脚被采样为高电平时 (时钟仲裁), 波特率发生器进行一个 **TBRG** 周期的计数。然后将 **SCL** 引脚拉低。在这之后, **ACKEN** 位自动清零, 波特率发生器关闭并且 **MSSP** 模块进入空闲模式 (图 17-25)。

### 17.4.12.1 WCOL 状态标志

如果用户在应答序列进行过程中试图写 **SSPBUF**, 则 **WCOL** 将置 1, 且缓冲器的内容不会改变 (写操作无效)。

## 17.4.13 停止条件时序

将停止序列使能位 **PEN** (**SSPCON2<2>**) 置 1, 在接收 / 发送结束后, **SDA** 引脚上将产生停止位。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, 主器件将 **SDA** 线置为低电平。当 **SDA** 线被采样为低电平时, 波特率发生器发生重载并递减计数到 0。当波特率发生器发生超时, **SCL** 引脚被拉为高电平, 在一个 **TBRG** (波特率发生器计满返回周期) 之后, **SDA** 引脚将被拉高。当 **SDA** 引脚被采样为高电平且 **SCL** 也是高电平时, **P** 位 (**SSPSTAT<4>**) 置 1。另一个 **TBRG** 之后, **PEN** 位被清零, 同时 **SSPIF** 位被置 1 (图 17-26)。

### 17.4.13.1 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 (写操作无效)。

图 17-25: 应答序列时序

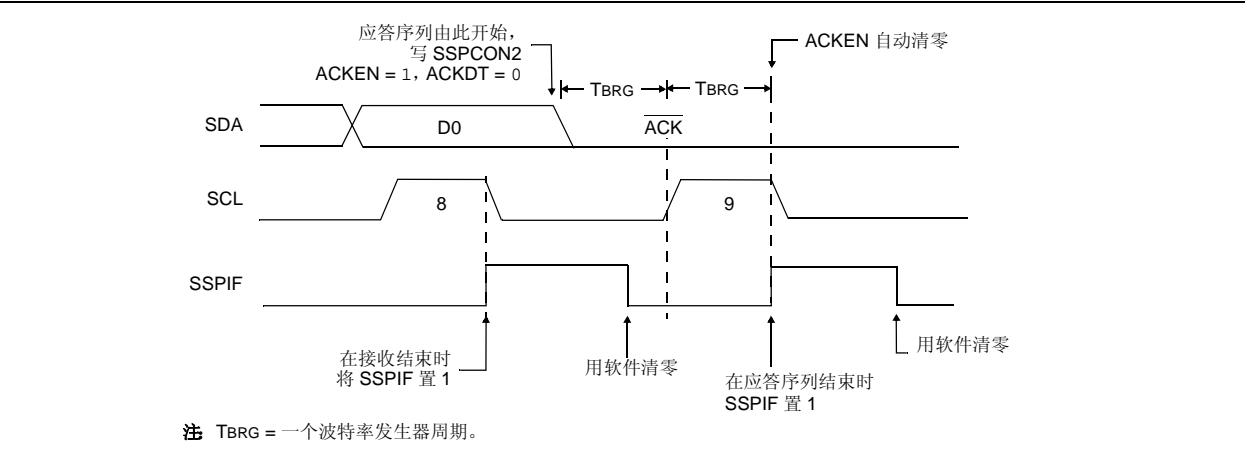
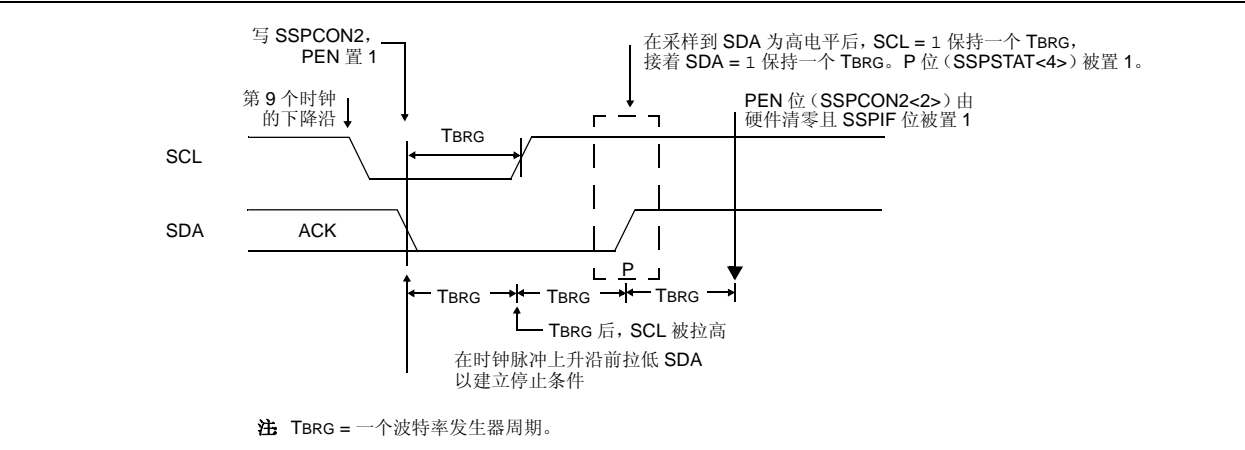


图 17-26: 停止条件接收或发送模式





17.4.14 休眠状态下的操作

在休眠模式下，I<sup>2</sup>C 模块能够接收地址或数据，并且在地址匹配时或字节传输完成后，如果允许 MSSP 中断，将唤醒处理器。

17.4.15 复位的影响

复位会禁止 MSSP 模块并终止当前的传输。

17.4.16 多主器件模式

在多主器件模式下，在检测到启动和停止条件时将产生中断，这可用于判断总线何时空闲。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（SSPSTAT<4>）置 1 时，可取得 I<sup>2</sup>C 总线的控制权；否则，P 位和 S 位均为零且总线空闲。当总线忙时，一旦出现停止条件，在允许 MSSP 中断的情况下将产生中断。

在多主器件模式下，必须一直监视 SDA 线来进行仲裁，以查看信号电平是否为期望的输出电平。此操作由硬件实现，其结果保存在 BCLIF 位中。

可能导致仲裁失败的情况是：

- 地址传输
- 数据传输
- 启动条件
- 重复启动条件
- 应答条件

17.4.17 多主器件通信、总线冲突与总线仲裁

通过总线仲裁来支持多主器件模式。当主器件将地址/数据位输出到 SDA 引脚时，如果主器件通过将 SDA 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，而实际采样到的数据是 0，则表示发生了总线冲突。主器件将把总线冲突中断标志位 BCLIF 置 1，并将 I<sup>2</sup>C 端口复位到空闲状态（图 17-27）。

如果在发送过程中发生总线冲突，则会暂停发送，且 BF 标志位被清零，SDA 和 SCL 线被拉高同时将 SSPBUF 置于可写入状态。当执行完总线冲突中断服务程序后，如果 I<sup>2</sup>C 总线空闲，用户可通过发出启动条件恢复通信。

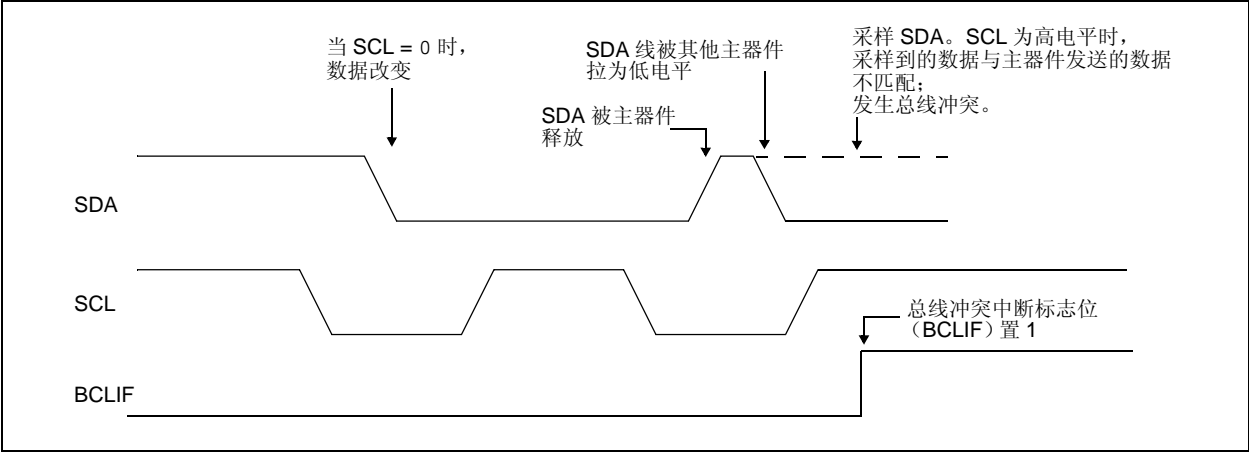
如果在启动、重复启动、停止或应答条件进行过程中发生总线冲突，那么相应的条件将被中止，SDA 和 SCL 线被拉高，SSPCON2 寄存器中的对应控制位清零。当执行完总线冲突中断服务程序后，如果 I<sup>2</sup>C 总线空闲，用户可通过发出启动条件恢复通信。

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，写入 SSPBUF 都会重新从第一个数据位开始发送数据。

在多主器件模式下，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。SSPSTAT 寄存器中的 P 位置 1 时，可以取得 I<sup>2</sup>C 总线的控制权；否则 P 位和 S 位清零，总线空闲。

图 17-27: 发送和应答时的总线冲突时序



# PIC18F85J11 系列

## 17.4.17.1 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：

- 在启动条件开始时，SDA 或 SCL 被采样为低电平（图 17-28）。
- 将 SDA 拉低之前，SCL 被采样为低电平（图 17-29）。

在启动条件期间，将监视 SDA 和 SCL 引脚。

如果 SDA 引脚或 SCL 引脚已经是低电平，则会发生下面的情况：

- 中止启动条件；
- BCLIF 标志位置 1；且
- MSSP 模块复位为空闲状态（图 17-28）。

拉高 SDA 和 SCL 引脚将建立启动条件。当 SDA 引脚被采样为高电平时，波特率发生器装入 `SSPADD<6:0>` 的值并递减计数到 0。如果在 SDA 引脚为高电平时，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 17-30）。但是，如果 SDA 引脚被采样为 1，SDA 引脚将在 BRG 计数结束时被拉为低电平。接着，波特率发生器被重载并递减计数到 0。在此期间，如果 SCL 引脚采样到 0，则不会发生总线冲突。在 BRG 计数结束时，SCL 引脚被拉为低电平。

**注：** 在启动条件期间不太可能发生总线冲突，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，一个主器件将总是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，因为两个主器件一定不会对启动条件后的第一个地址进行仲裁。如果地址是相同的，必须继续对数据部分、重复启动或停止条件进行仲裁。

图 17-28: 启动条件期间的总线冲突（仅 SDA）

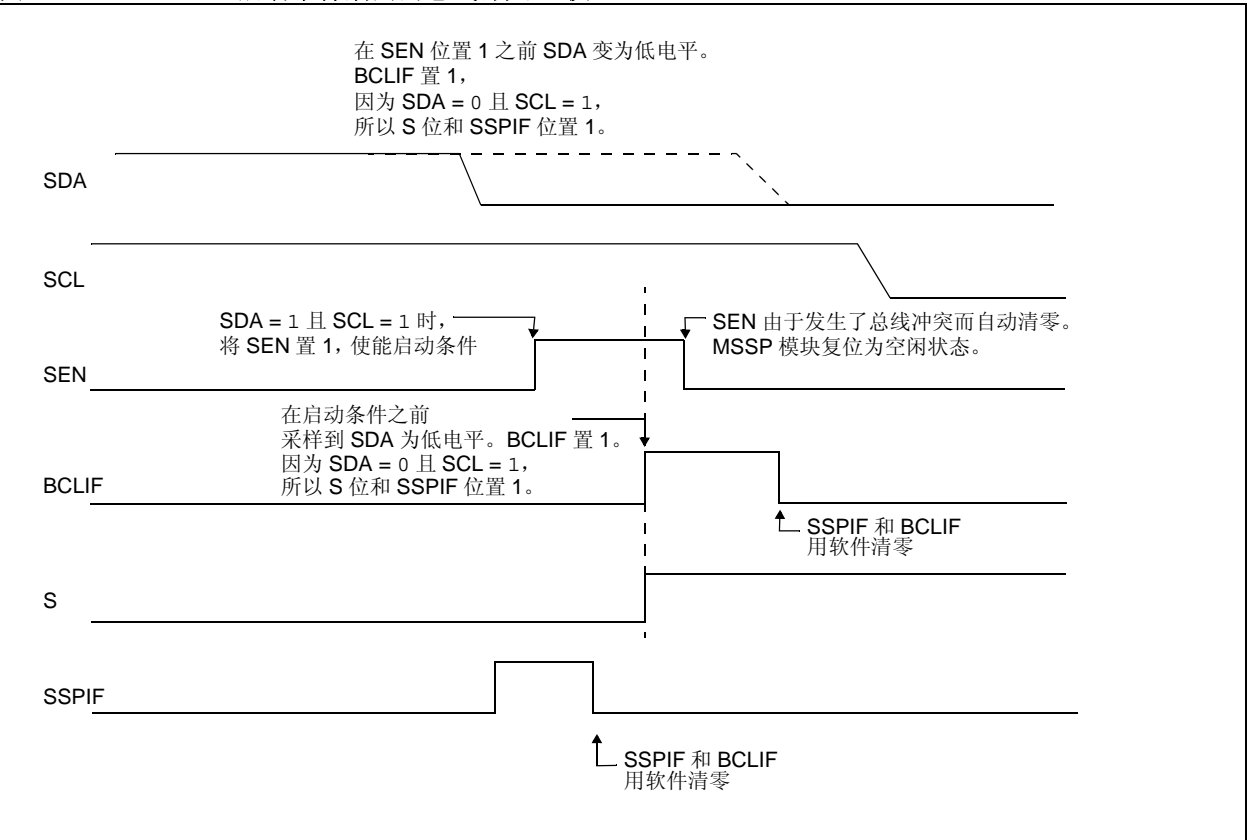


图 17-29: 启动条件期间的总线冲突 (SCL = 0)

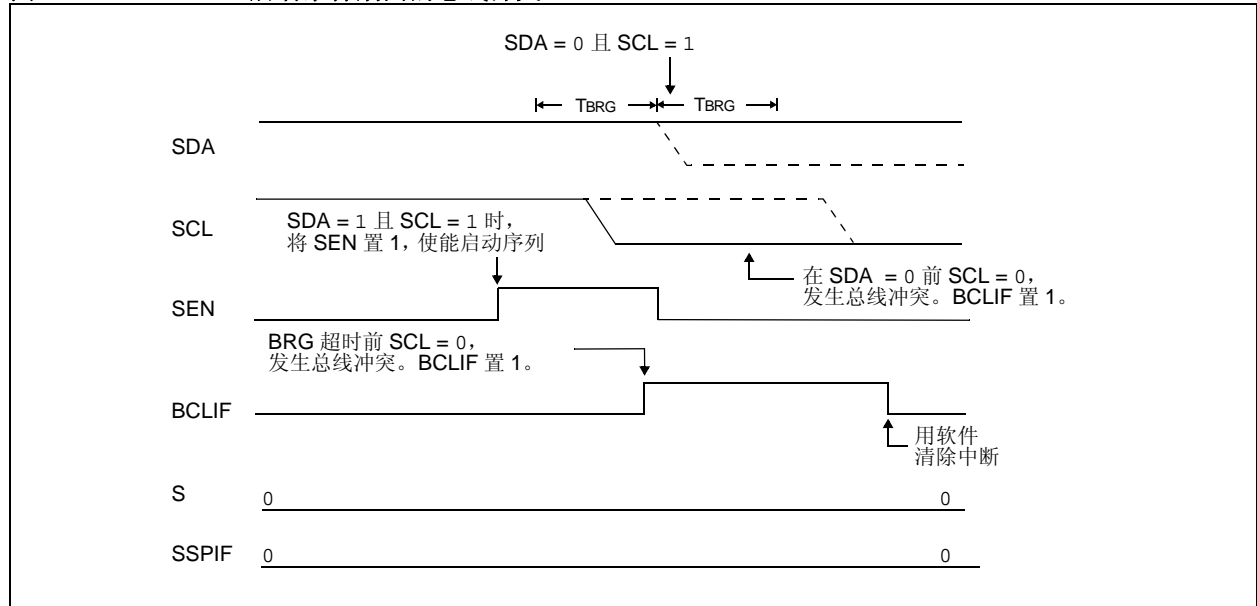
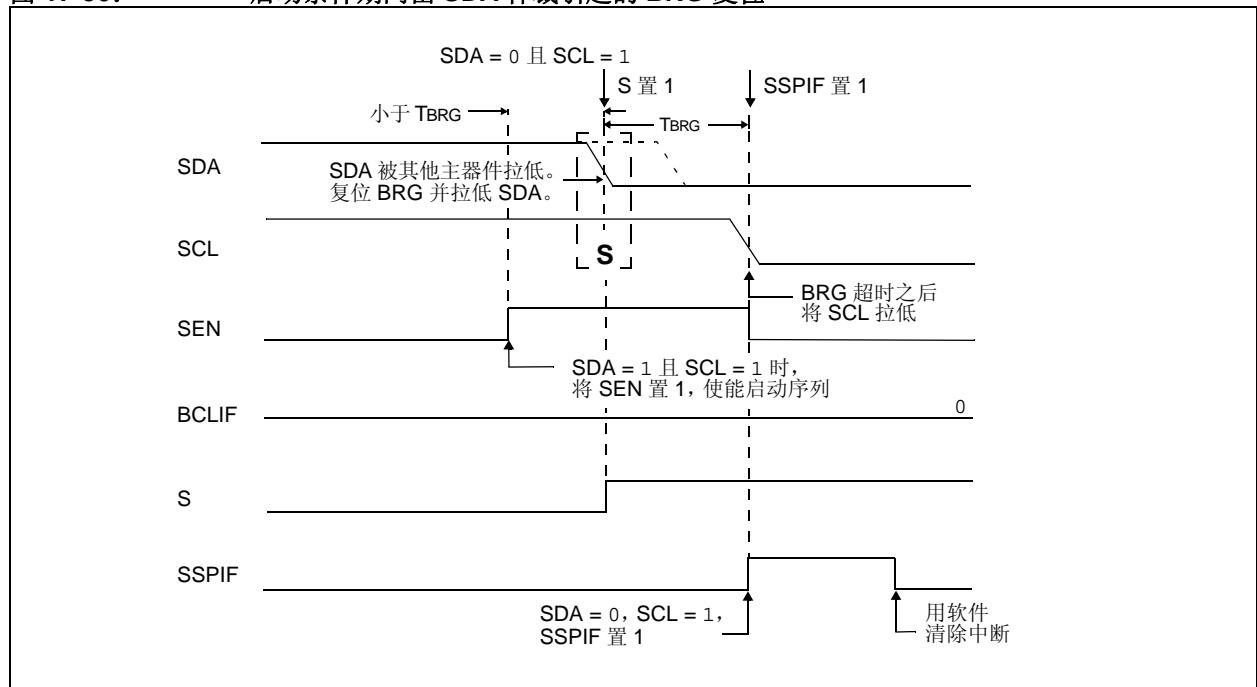


图 17-30: 启动条件期间由 SDA 仲裁引起的 BRG 复位



# PIC18F85J11 系列

## 17.4.17.2 重复启动条件期间的总线冲突

在重复启动条件期间，以下事件会导致发生总线冲突：

- a) 在 SCL 由低电平变为高电平期间，在 SDA 上采样到低电平。
- 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数据 1。

当用户释放 SDA 且允许该引脚悬空为高电平时，BRG 装入 SSPADD<6:0> 中的值并递减计数到 0。接着 SCL 引脚被拉为高电平，当 SCL 采样到高电平时，对 SDA 引脚进行采样。

如果 SDA 为低电平，则表示已发生总线冲突（即，另一个主器件正试图发送数据 0，见图 17-31）。如果 SDA 被采样为高电平，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，则不会发生总线冲突，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 尚未被拉低，那么将发生总线冲突。在此情况下，另一个主器件在重复启动条件期间正试图发送数据 1（见图 17-32）。

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，不管 SCL 引脚的状态如何，SCL 引脚都被拉低，重复启动条件结束。

图 17-31: 重复启动条件期间的总线冲突（情形 1）

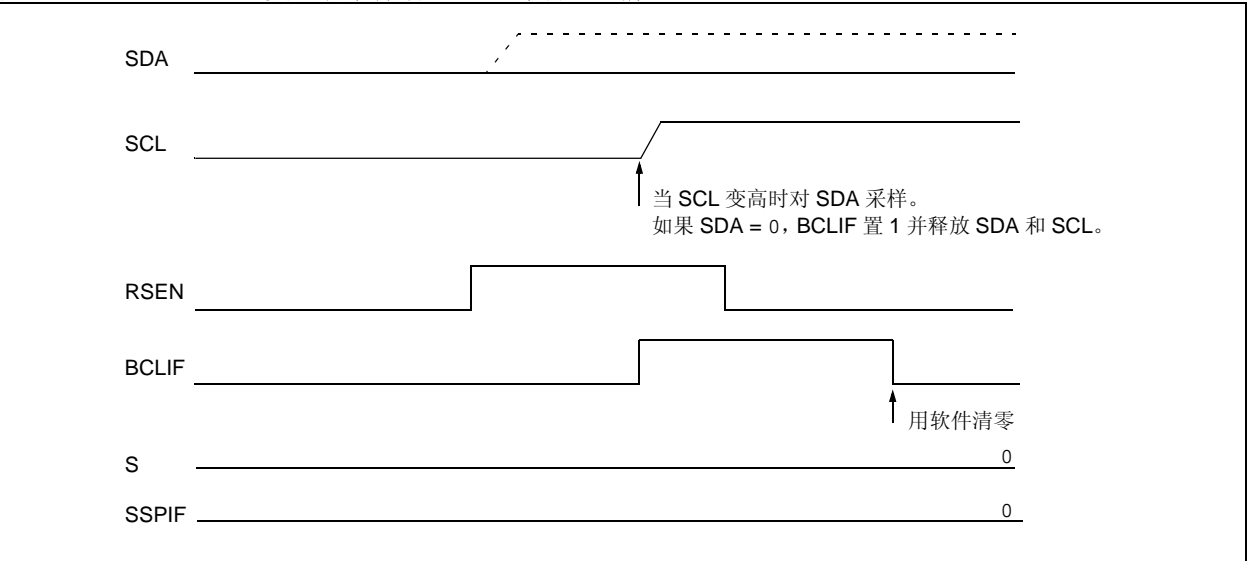
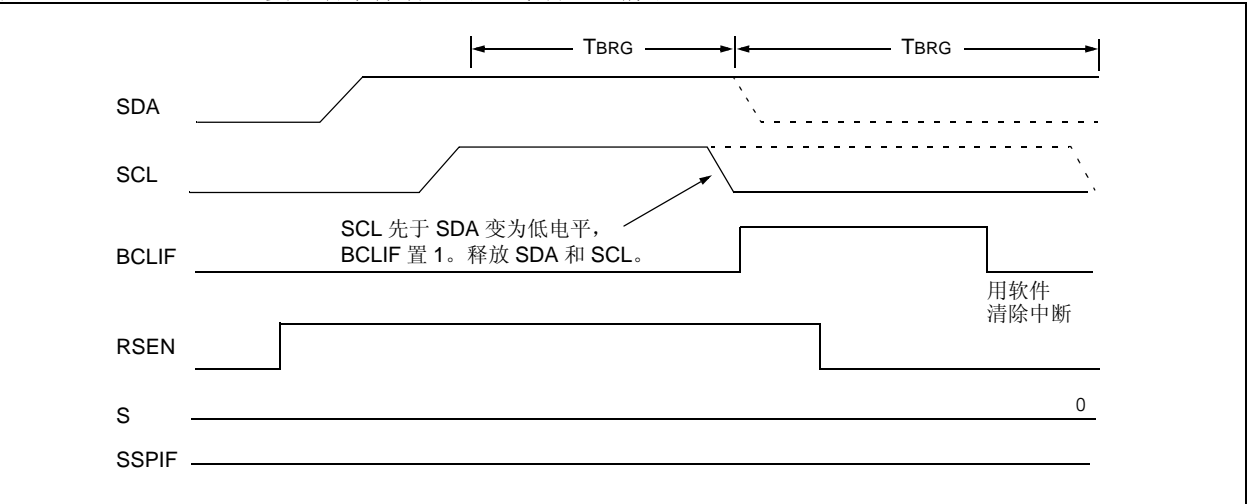


图 17-32: 重复启动条件期间的总线冲突（情形 2）



## 17.4.17.3 停止条件期间的总线冲突

以下事件会导致停止条件期间发生总线冲突：

- SDA 被释放并允许悬空为高电平之后，SDA 在 BRG 超时后采样为低电平。
- SCL 引脚被拉高之后，SCL 在 SDA 变为高电平之前采样为低电平。

停止条件从 SDA 拉低开始。当 SDA 被采样为低电平时，SCL 引脚就可以悬空。当 SDA 引脚被采样为高电平时（时钟仲裁），波特率发生器被载入 SSPADD<6:0> 的值并递减计数到 0。BRG 超时后，采样 SDA。如果 SDA 被采样为低电平，则表示已发生总线冲突。这是因为另一个主器件正试图发送一个数据 0（图 17-33）。如果 SCL 引脚在允许 SDA 悬空为高电平前被采样为低电平，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 的另一个情形（图 17-34）。

图 17-33: 停止条件期间的总线冲突（情形 1）

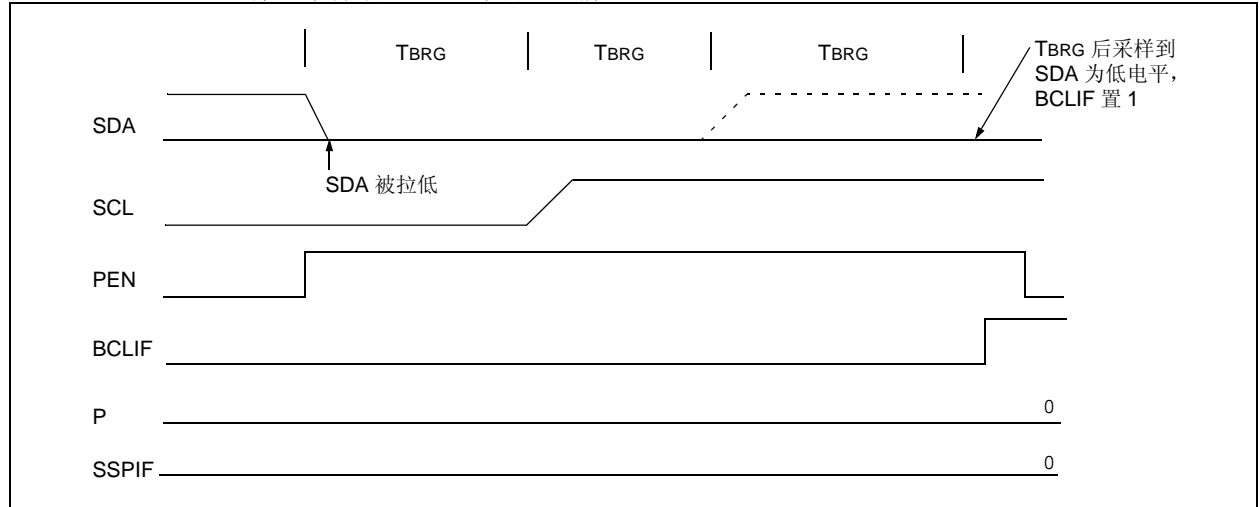
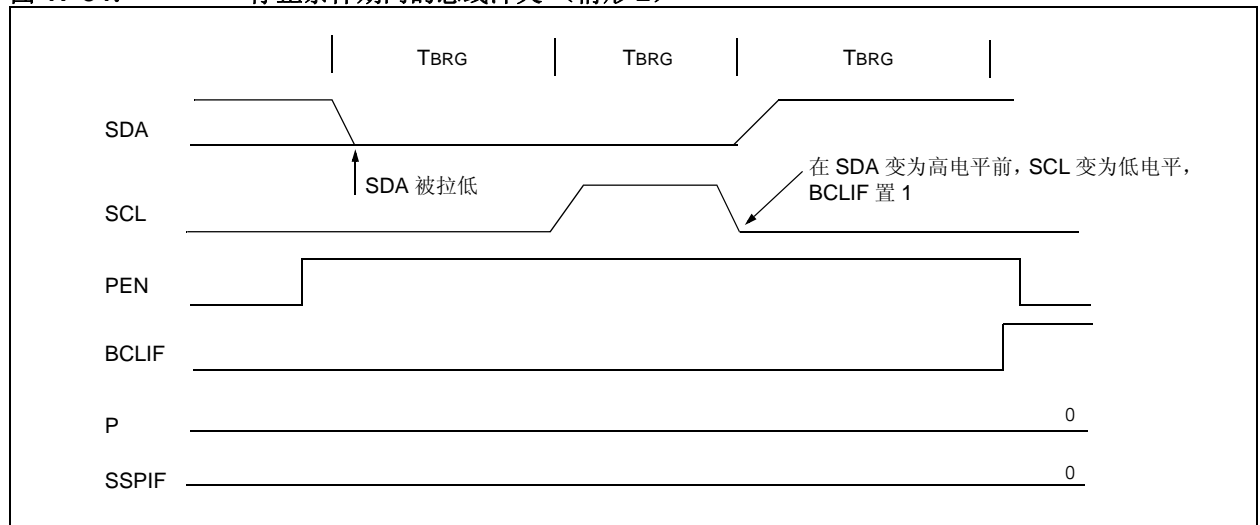


图 17-34: 停止条件期间的总线冲突（情形 2）



# PIC18F85J11 系列

表 17-4: 与 I<sup>2</sup>C™ 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	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	—	TMR2IF	TMR1IF	59
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	—	TMR2IE	TMR1IE	59
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	—	TMR2IP	TMR1IP	59
PIR2	OSCFIF	CMIF	—	—	BCLIF	LVDIF	TMR3IF	—	59
PIE2	OSCFIE	CMIE	—	—	BCLIE	LVDIE	TMR3IE	—	59
IPR2	OSCFIP	CMIP	—	—	BCLIP	LVDIP	TMR3IP	—	59
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	60
SSPBUF	MSSP 接收缓冲器 / 发送寄存器								58
SSPADD	MSSP 地址寄存器 (I <sup>2</sup> C™ 从模式), MSSP 波特率重载寄存器 (I <sup>2</sup> C 主模式)								58
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	58
SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	58
	GCEN	ACKSTAT	ADMSK5 <sup>(1)</sup>	ADMSK4 <sup>(1)</sup>	ADMSK3 <sup>(1)</sup>	ADMSK2 <sup>(1)</sup>	ADMSK1 <sup>(1)</sup>	SEN	
SSPSTAT	SMP	CKE	D/Ā	P	S	R/Ā	UA	BF	58

图注: — = 未实现, 读为 0。I<sup>2</sup>C™ 模式下的 MSSP 模块不使用阴影单元。

注 1: 只有在 I<sup>2</sup>C 从模式下工作时才可使用备用位定义。

## 18.0 增强型通用同步 / 异步收发器 (EUSART)

PIC18F85J11 系列器件具有 3 个串行 I/O 模块：前一章中讨论的 MSSP 模块和两个通用同步 / 异步收发器 (Universal Synchronous Asynchronous Receiver Transmitter, USART) 模块。(通常, USART 也被称为串行通信接口或 SCI。) 可以将 EUSART 配置为能与 CRT 终端和个人计算机等外设通信的全双工异步系统, 也可以将它配置成能够与 A/D 或 D/A 集成电路、串行 EEPROM 等外设通信的半双工同步系统。

这些器件的 USART 模块有两种不同的实现方法：本章讨论的增强型 USART (EUSART) 和将在下一章讨论的可寻址 USART。对于该系列器件来说, USART1 总是指 EUSART, 而 USART2 总是指 AUSART。

EUSART 和 AUSART 模块的串行通信使用相同的核心功能, 其基本操作基本上是相同的。EUSART 模块还提供其他功能, 包括自动波特率检测和校准、在接收到“同步间隔”字符时自动唤醒和 12 位间隔字符发送。这些功能使 EUSART 模块成为局域互连网络 (LIN/J2602) 总线系统的理想选择。

EUSART 可配置为以下几种工作模式：

- 带有以下功能的全双工异步模式：
  - 字符接收自动唤醒
  - 自动波特率校准
  - 12 位间隔字符发送
- 时钟极性可选的半双工同步主模式
- 时钟极性可选的半双工同步从模式

EUSART 的引脚与 PORTC (RC6/TX1/CK1 和 RC7/RX1/DT1) 功能复用。为了将这些引脚配置为 EUSART：

- SPEN (RCSTA<7>) 位必须置 1 (= 1)
- TRISC<7> 位必须置 1 (= 1)
- TRISC<6> 位必须置 1 (= 1)

**注：** EUSART 控制逻辑在需要时会自动将引脚从输入重新配置为输出。

TX1 输出引脚的驱动器也可选择配置为漏极开路输出。此功能允许通过外部上拉电阻将该引脚上的电平拉为高电平, 并允许漏极开路输出与外部电路通信, 而无需额外的电平转换器。

漏极开路输出选项由 U1OD 位 (LATG<6>) 控制。将该位置 1 可将引脚配置为漏极开路操作。

### 18.1 控制寄存器

增强型 USART 模块的操作由以下 3 个寄存器控制：

- 发送状态和控制寄存器 1 (TXSTA1)
- 接收状态和控制寄存器 1 (RCSTA1)
- 波特率控制寄存器 1 (BAUDCON1)

这些寄存器如寄存器 18-1、寄存器 18-2 和寄存器 18-3 所述。

# PIC18F85J11 系列

寄存器 18-1: TXSTA1: EUSART 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 <sup>(1)</sup>	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	<b>CSRC:</b> 时钟源选择位 <u>异步模式:</u> 无关位。 <u>同步模式:</u> 1 = 主模式 (时钟来自内部 BRG) 0 = 从模式 (时钟来自外部时钟源)
bit 6	<b>TX9:</b> 9 位发送使能位 1 = 选择 9 位发送 0 = 选择 8 位发送
bit 5	<b>TXEN:</b> 发送使能位 <sup>(1)</sup> 1 = 使能发送 0 = 禁止发送
bit 4	<b>SYNC:</b> EUSART 模式选择位 1 = 同步模式 0 = 异步模式
bit 3	<b>SENDB:</b> 发送间隔字符位 <u>异步模式:</u> 1 = 在下一次发送时发送 “同步间隔” 字符 (在完成时用硬件清零) 0 = “同步间隔” 字符发送完成 <u>同步模式:</u> 无关位。
bit 2	<b>BRGH:</b> 高波特率选择位 <u>异步模式:</u> 1 = 高速 0 = 低速 <u>同步模式:</u> 在此模式下未使用。
bit 1	<b>TRMT:</b> 发送移位寄存器状态位 1 = TSR 空 0 = TSR 满
bit 0	<b>TX9D:</b> 发送数据的第 9 位 该位可以是地址 / 数据位或奇偶校验位。

注 1: 同步模式下 SREN/CREN 的优先级高于 TXEN。



**寄存器 18-2: RCSTA1: EUSART 接收状态和控制寄存器**

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:** 串口使能位  
1 = 使能串口  
0 = 禁止串口 (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:  
无关位。  
同步主模式:  
1 = 使能单字节接收  
0 = 禁止单字节接收  
此位在接收完成后清零。  
同步从模式:  
无关位。
- bit 4      CREN:** 连续接收使能位  
异步模式:  
1 = 使能接收器  
0 = 禁止接收器  
同步模式:  
1 = 使能连续接收, 直到使能位 CREN 清零 (CREN 的优先级高于 SREN)  
0 = 禁止连续接收
- bit 3      ADDEN:** 地址检测使能位  
9 位异步模式 (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, 接收到所有字节并且第 9 位可用作奇偶校验位  
9 位异步模式 (RX9 = 0):  
无关位。
- bit 2      FERR:** 帧错误位  
1 = 帧错误 (可以通过读 RCREG1 寄存器清除, 并接收下一个有效字节)  
0 = 无帧错误
- bit 1      OERR:** 溢出错误位  
1 = 溢出错误 (可以通过清零 CREN 位清除)  
0 = 无溢出错误
- bit 0      RX9D:** 接收数据的第 9 位  
该位可以是地址 / 数据位或奇偶校验位, 必须由用户固件计算得到。

# PIC18F85J11 系列

寄存器 18-3: **BAUDCON1: 波特率控制寄存器 1**

R/W-0	R-1	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	RXDTP	TXCKP	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:** 自动波特率采集计满返回状态位  
1 = 在自动波特率检测模式下发生了 BRG 计满返回 (必须用软件清零)  
0 = 没有发生 BRG 计满返回
- bit 6      **RCIDL:** 接收操作空闲状态位  
1 = 接收操作处于空闲状态  
0 = 接收操作处于活动状态
- bit 5      **RXDTP:** 接收数据极性选择位 (仅异步模式)  
异步模式:  
1 = 接收的数据 (RXx) 反相 (低电平有效)  
0 = 接收的数据 (RXx) 未反相 (高电平有效)
- bit 4      **TXCKP:** 时钟和数据极性选择位  
异步模式:  
1 = 发送 (TXx) 空闲状态为低电平  
0 = 发送 (TXx) 空闲状态为高电平  
同步模式:  
1 = 时钟 (CKx) 空闲状态为高电平  
0 = 时钟 (CKx) 空闲状态为低电平
- bit 3      **BRG16:** 16 位波特率寄存器使能位  
1 = 16 位波特率发生器——SPBRGH1 和 SPBRG1  
0 = 8 位波特率发生器——仅 SPBRG1 (兼容模式), 忽略 SPBRGH1 的值
- bit 2      **未实现:** 读为 0
- bit 1      **WUE:** 唤醒使能位  
异步模式:  
1 = EUSART 将继续采样 RX1 引脚——中断在下降沿产生, 在下一个上升沿由硬件清零该位  
0 = 未监视 RX1 引脚或检测到了上升沿  
同步模式:  
在此模式下未使用。
- bit 0      **ABDEN:** 自动波特率检测使能位  
异步模式:  
1 = 在下一个字符使能波特率检测。需要收到“同步”字段 (55h), 完成时由硬件清零。  
0 = 禁止波特率检测或检测已完成  
同步模式:  
在此模式下未使用。

18.2 EUSART 波特率发生器 (BRG)

BRG 是一个专用的 8 位或 16 位发生器，支持 EUSART 的异步和同步模式。默认情况下，BRG 工作在 8 位模式下，将 BRG16 位 (BAUDCON1<3>) 置 1 可选择 16 位模式。

SPBRGH1:SPBRG1 寄存器对控制自由运行定时器的周期。在异步模式下，BRGH (TXSTA1<2>) 和 BRG16 (BAUDCON1<3>) 位也控制波特率。在同步模式下，会忽略 BRGH 位。表 18-1 所示为不同 EUSART 模式下的波特率计算公式，但仅适用于主模式 (由内部产生时钟信号)。

给定目标波特率和 Fosc 的情况下，可以使用表 18-1 中的公式来计算 SPBRGH1:SPBRG1 寄存器的最近似整数值，从而确定波特率误差。例 18-1 给出了一个计算示例。表 18-2 中给出了各种异步模式下典型的波特率和误差。

差值。使用高波特率 (BRGH = 1) 或 16 位 BRG 有利于减少波特率误差，或者在快速振荡频率条件下实现低波特率。

向 SPBRGH1:SPBRG1 寄存器写入新值会引起 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。当工作在同步模式下时，不支持 0000h 和 0001h 的 SPBRGH:SPBRG 值。在异步模式下，可使用所有 BRG 值。

18.2.1 功耗管理模式下的操作

器件时钟用于产生所需的波特率。当进入一种功耗管理模式时，新时钟源可能会工作在与先前不同的频率下。这可能需要调整 SPBRG1 寄存器对中的值。

18.2.2 采样

择多检测电路对 RX1 引脚数据采样三次，以判定 RX1 引脚上出现的是高电平还是低电平。

表 18-1: 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	$F_{osc}/[64 (n + 1)]$
0	0	1	8 位 / 异步	$F_{osc}/[16 (n + 1)]$
0	1	0	16 位 / 异步	
0	1	1	16 位 / 异步	$F_{osc}/[4 (n + 1)]$
1	0	x	8 位 / 同步	
1	1	x	16 位 / 同步	

图注: x = 任意值, n = SPBRGH1:SPBRG1 寄存器对的值

# PIC18F85J11 系列

## 例 18-1: 计算波特率误差

针对工作在异步模式下、工作频率 Fosc 为 16 MHz，采用 8 位 BRG、目标波特率为 9600 bps 的器件：  
目标波特率 =  $F_{osc}/(64 ([SPBRGH1:SPBRG1] + 1))$   
求解 SPBRGH1:SPBRG1：  
$$X = ((F_{osc}/\text{目标波特率})/64) - 1$$
$$= ((16000000/9600)/64) - 1$$
$$= [25.042] = 25$$
  
波特率计算结果 =  $16000000/(64 (25 + 1))$ 
$$= 9615$$
  
误差 =  $(\text{波特率计算结果} - \text{目标波特率}) / \text{目标波特率}$ 
$$= (9615 - 9600)/9600 = 0.16\%$$

表 18-2: 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	复位值所在页
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	59
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	59
BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	60
SPBRGH1	EUSART 波特率发生器寄存器的高字节								60
SPBRG1	EUSART 波特率发生器寄存器的低字节								59

图注： — = 未实现，读为 0。BRG 不使用阴影单元。

表 18-3: 异步模式下的波特率

目标波特率 (K)	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际波特率 (K)	误差 %	SPBRG 值 (十进制)	实际波特率 (K)	误差 %	SPBRG 值 (十进制)	实际波特率 (K)	误差 %	SPBRG 值 (十进制)	实际波特率 (K)	误差 %	SPBRG 值 (十进制)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1.201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2.403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9.615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-32.18	1	—	—	—

目标波特率 (K)	SYNC = 0, BRGH = 0, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际波特率 (K)	误差 %	SPBRG 值 (十进制)	实际波特率 (K)	误差 %	SPBRG 值 (十进制)	实际波特率 (K)	误差 %	SPBRG 值 (十进制)
0.3	0.300	0.16	207	0.300	-0.16	103	0.300	-0.16	51
1.2	1.202	0.16	51	1.201	-0.16	25	1.201	-0.16	12
2.4	2.404	0.16	25	2.403	-0.16	12	—	—	—
9.6	8.929	-6.99	6	—	—	—	—	—	—
19.2	20.833	8.51	2	—	—	—	—	—	—
57.6	62.500	8.51	0	—	—	—	—	—	—
115.2	62.500	-45.75	0	—	—	—	—	—	—

目标波特率 (K)	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际波特率 (K)	误差 %	SPBRG 值 (十进制)	实际波特率 (K)	误差 %	SPBRG 值 (十进制)	实际波特率 (K)	误差 %	SPBRG 值 (十进制)	实际波特率 (K)	误差 %	SPBRG 值 (十进制)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	1.73	255	2.403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

目标波特率 (K)	SYNC = 0, BRGH = 1, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际波特率 (K)	误差 %	SPBRG 值 (十进制)	实际波特率 (K)	误差 %	SPBRG 值 (十进制)	实际波特率 (K)	误差 %	SPBRG 值 (十进制)
0.3	—	—	—	—	—	—	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

# PIC18F85J11 系列

表 18-3: 异步模式下的波特率 (续)

目标 波特率 (K)	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际波 特率 (K)	误差 %	SPBRG 值 (十进制)	实际波 特率 (K)	误差 %	SPBRG 值 (十进制)	实际波 特率 (K)	误差 %	SPBRG 值 (十进制)	实际波 特率 (K)	误差 %	SPBRG 值 (十进制)
0.3	0.300	0.00	8332	0.300	0.02	4165	0.300	0.02	2082	0.300	-0.04	1665
1.2	1.200	0.02	2082	1.200	-0.03	1041	1.200	-0.03	520	1.201	-0.16	415
2.4	2.402	0.06	1040	2.399	-0.03	520	2.404	0.16	259	2.403	-0.16	207
9.6	9.615	0.16	259	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

目标 波特率 (K)	SYNC = 0, BRGH = 0, BRG16 = 1								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际波 特率 (K)	误差 %	SPBRG 值 (十进制)	实际波 特率 (K)	误差 %	SPBRG 值 (十进制)	实际波 特率 (K)	误差 %	SPBRG 值 (十进制)
0.3	0.300	0.04	832	0.300	-0.16	415	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

目标 波特率 (K)	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际波 特率 (K)	误差 %	SPBRG 值 (十进制)	实际波 特率 (K)	误差 %	SPBRG 值 (十进制)	实际波 特率 (K)	误差 %	SPBRG 值 (十进制)	实际波 特率 (K)	误差 %	SPBRG 值 (十进制)
0.3	0.300	0.00	33332	0.300	0.00	16665	0.300	0.00	8332	0.300	-0.01	6665
1.2	1.200	0.00	8332	1.200	0.02	4165	1.200	0.02	2082	1.200	-0.04	1665
2.4	2.400	0.02	4165	2.400	0.02	2082	2.402	0.06	1040	2.400	-0.04	832
9.6	9.606	0.06	1040	9.596	-0.03	520	9.615	0.16	259	9.615	-0.16	207
19.2	19.193	-0.03	520	19.231	0.16	259	19.231	0.16	129	19.230	-0.16	103
57.6	57.803	0.35	172	57.471	-0.22	86	58.140	0.94	42	57.142	0.79	34
115.2	114.943	-0.22	86	116.279	0.94	42	113.636	-1.36	21	117.647	-2.12	16

目标 波特率 (K)	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际波 特率 (K)	误差 %	SPBRG 值 (十进制)	实际波 特率 (K)	误差 %	SPBRG 值 (十进制)	实际波 特率 (K)	误差 %	SPBRG 值 (十进制)
0.3	0.300	0.01	3332	0.300	-0.04	1665	0.300	-0.04	832
1.2	1.200	0.04	832	1.201	-0.16	415	1.201	-0.16	207
2.4	2.404	0.16	415	2.403	-0.16	207	2.403	-0.16	103
9.6	9.615	0.16	103	9.615	-0.16	51	9.615	-0.16	25
19.2	19.231	0.16	51	19.230	-0.16	25	19.230	-0.16	12
57.6	58.824	2.12	16	55.555	3.55	8	—	—	—
115.2	111.111	-3.55	8	—	—	—	—	—	—

18.2.3 自动波特率检测

增强型 USART 模块支持波特率自动检测和校准。此功能仅在异步模式下当 WUE 位清零时有效。

只要接收到起始位并且 ABDEN 位已置 1，就会开始自动波特率检测（图 18-1）。波特率计算采用自平均的方式。

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 为输入 RX1 的信号提供时钟源，而是由 RX1 信号为 BRG 定时。在 ABD 模式下，内部波特率发生器被用作计数器来计算输入的串行字节流的位间隔时间。

一旦 ABDEN 位置 1，状态机就会将 BRG 清零并寻找起始位。为了正确计算比特率，自动波特率检测必须接收到一个值为 55h（ASCII 字符“U”，也是 LIN/J2602 总线的同步字符）的字节。为了尽量减少输入信号不对称所造成的影响，在接收低位和高位的时间内都要进行检测。在起始位后，SPBRG1 使用预先选择的时钟源在 RX1 引脚上的第一个上升沿开始计数。在 RX1 引脚传输了 8 个位，或在检测到第 5 个上升沿后，会将在相应 BRG 周期内累加的值保存在 SPBRGH1:SPBRG1 寄存器对中。当第 5 个边沿出现时（应与停止位对应），ABDEN 位会自动清零。

如果发生了 BRG 计满返回（从 FFFFh 到 0000h 的溢出），会在 ABDOVF 状态位（BAUDCON<7>）有所反映。当 BRG 计满返回时，该位由硬件置 1，用户也可用软件将其置 1 或清零。在发生计满返回事件后，继续保持 ABD 模式，ABDEN 位保持置 1（图 18-2）。

在校准波特率周期时，BRG 寄存器时钟频率为预配置时钟频率的 1/8。请注意 BRG 时钟可由 BRG16 和 BRGH 位配置。必须将 BRG16 置 1 以使 SPBRG1 和 SPBRGH1 都被用作一个 16 位计数器。通过检查 SPBRGH1 寄存器中的值是否为 00h，用户可以验证在 8 位模式下是否发生了进位。表 18-4 所示为 BRG 计数器的时钟速率。

当产生 ABD 序列时，EUSART 状态机保持在空闲状态。一旦在 RX1 上检测到第 5 个上升沿，中断标志位 RC1IF 就会置 1。需要读取 RCREG1 中的值，来清零中断标志位 RC1IF。应丢弃 RCREG1 的值。

- 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。
- 2:

需要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。由于位错误率的原因，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，必须综合考虑系统总的时序和通信波特率。

表 18-4: BRG 计数器时钟速率

BRG16	BRGH	BRG 计数器时钟
0	0	Fosc/512
0	1	Fosc/128
1	0	Fosc/128
1	1	Fosc/32

注：在产生 ABD 序列时，不管 BRG16 的设置如何，SPBRG1 和 SPBRGH1 都被用作一个 16 位计数器。

18.2.3.1 ABD 和 EUSART 发送

由于在 ABD 采集期间 BRG 时钟是反向的，因此在 ABD 期间不能使用 EUSART 发送器。这意味着只要 ABDEN 位置 1，就不能写入 TXREG1。用户还应确保在发送期间 ABDEN 不能为置 1 状态，否则可能会导致无法预料的 EUSART 操作。

# PIC18F85J11 系列

图 18-1: 自动波特率计算

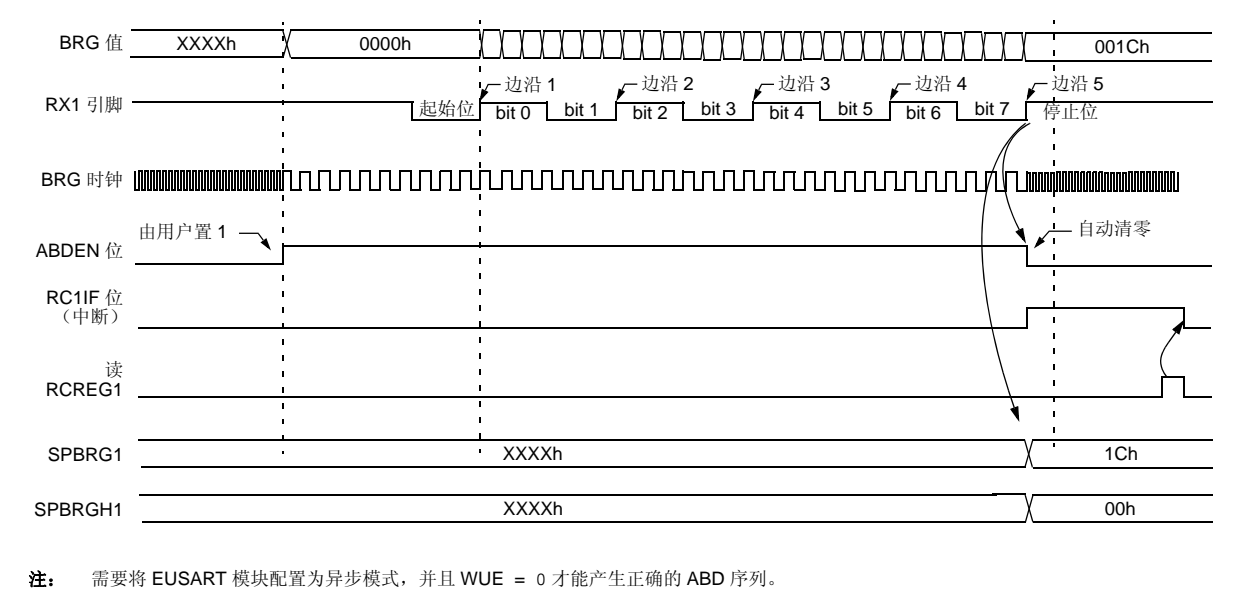
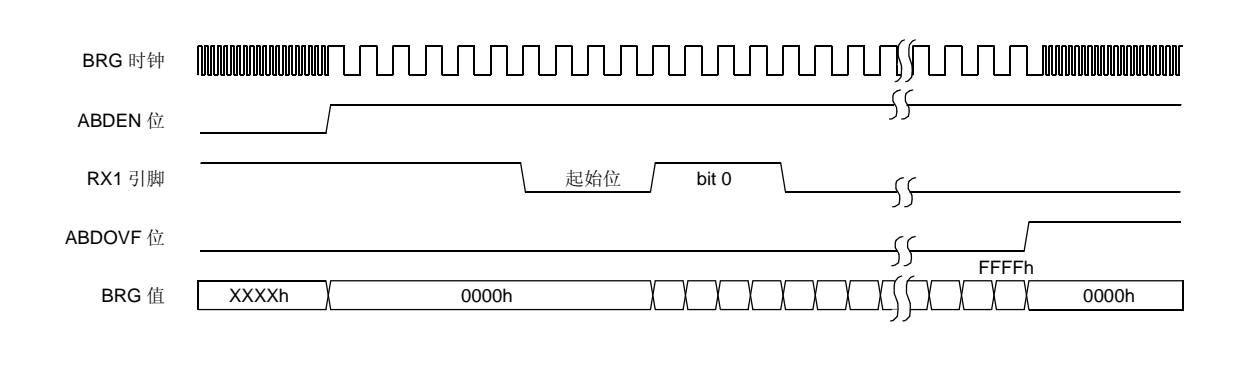


图 18-2: BRG 溢出时序







# PIC18F85J11 系列

图 18-4: 异步发送

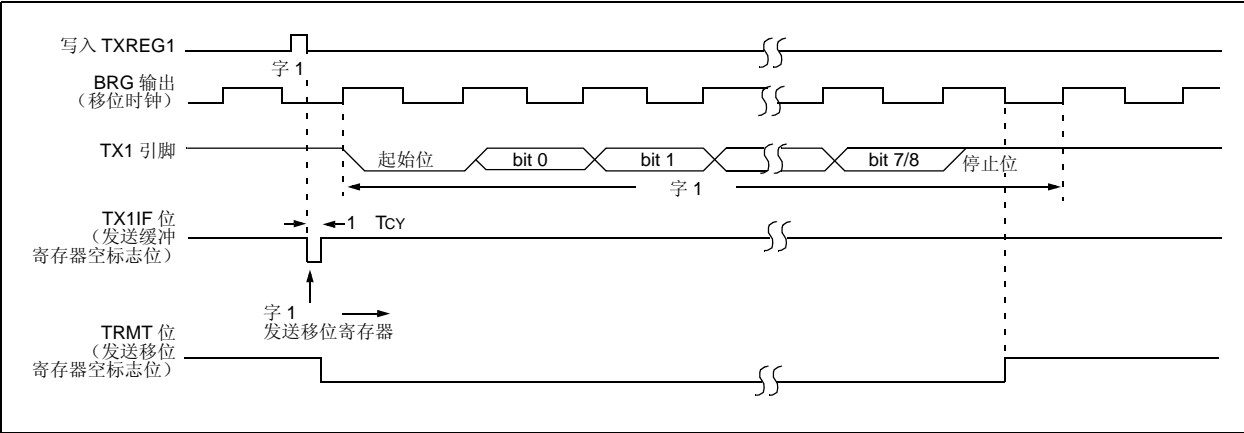


图 18-5: 异步发送（背对背）

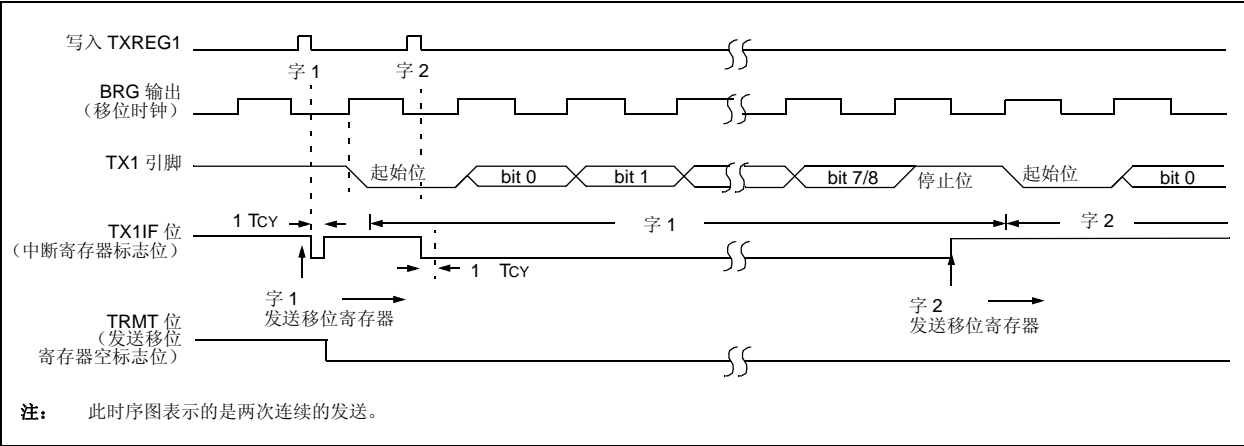


表 18-5: 与异步发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	—	TMR2IF	TMR1IF	59
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	—	TMR2IE	TMR1IE	59
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	—	TMR2IP	TMR1IP	59
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	59
TXREG1	EUSART 发送寄存器								59
TXSTA1	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	59
BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	60
SPBRGH1	EUSART 波特率发生器寄存器的高字节								60
SPBRG1	EUSART 波特率发生器寄存器的低字节								59
LATG	U2OD	U1OD	—	LATG4	LATG3	LATG2	LATG1	LATG0	60

图注： — = 未实现单元，读为 0。异步发送不使用阴影单元。

## 18.3.2 EUSART 异步接收器

图 18-6 显示了接收器的框图。在 RX1 引脚上接收数据，并驱动数据恢复电路。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，而主接收串行移位器的工作频率等于比特率或 Fosc。此模式通常用于 RS-232 系统。

设置异步接收操作的步骤如下：

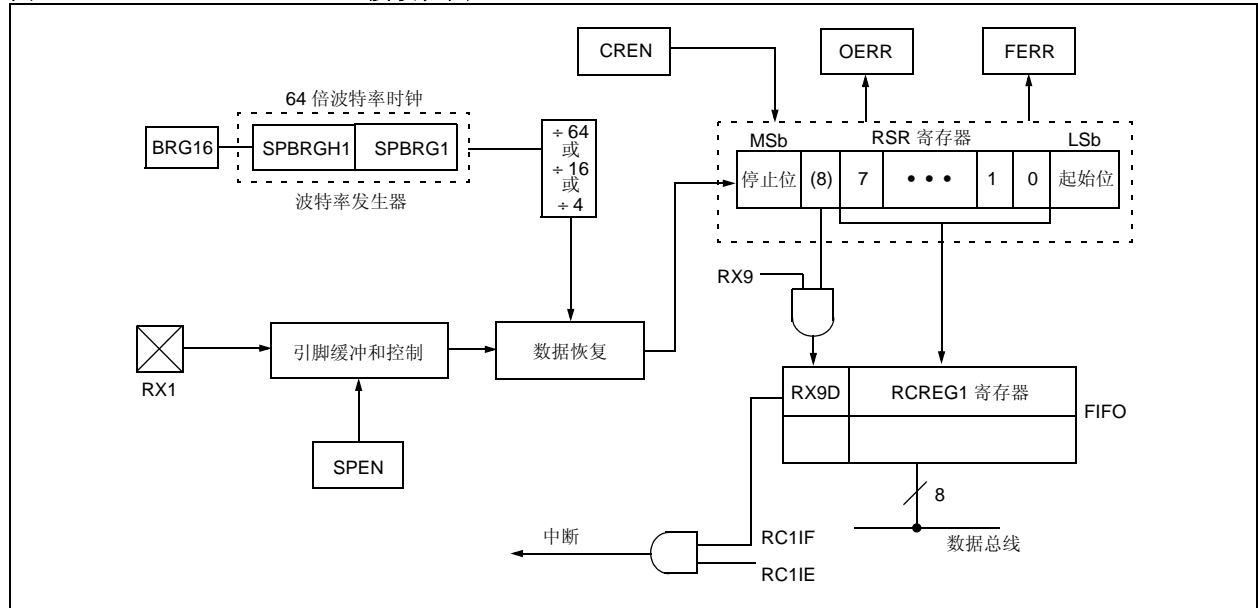
1. 初始化 SPBRGH1:SPBRG1 寄存器，设置合适的波特率。按需要将 BRGH 和 BRG16 位置 1 或清零，以获得目标波特率。
2. 通过将 SYNC 位清零并将 SPEN 位置 1 使能异步串口。
3. 如果需要中断，将中断允许位 RC1IE 置 1。
4. 若需要接收 9 位数据，将 RX9 位置 1。
5. 通过将 CREN 位置 1，使能接收。
6. 当接收完成时标志位 RC1IF 将置 1，此时如果允许位 RC1IE 已置 1，将会产生一个中断。
7. 读 RCSTA1 寄存器以获取第 9 位数据（如果已使能），并判断是否在接收过程中发生了错误。
8. 通过读 RCREG1 寄存器来读取接收到的 8 位数据。
9. 如果发生错误，通过将使能位 CREN 清零来清除错误。
10. 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）置 1。

## 18.3.3 设置带有地址检测功能的 9 位模式

此模式通常用于 RS-485 系统。按如下步骤设置带有地址检测功能的异步接收操作：

1. 初始化 SPBRGH1:SPBRG1 寄存器，设置合适的波特率。按需要将 BRGH 和 BRG16 位置 1 或清零，以获得目标波特率。
2. 通过将 SYNC 位清零并将 SPEN 位置 1 使能异步串口。
3. 若需要中断，请将 RCEN 位置 1 并使用 RC1IP 位设置优先级。
4. 将 RX9 位置 1，使能 9 位接收。
5. 将 ADDEN 位置 1，使能地址检测。
6. 将 CREN 位置 1，使能接收。
7. 当接收完成时 RC1IF 位将被置 1。此时如果 RC1IE 和 GIE 位已置 1，还将响应中断。
8. 读 RCSTA1 寄存器判断在接收时是否发生了错误，同时读取第 9 位数据（如果适用）。
9. 读 RCREG1 来判断是否正在对器件进行寻址。
10. 如果发生错误，将 CREN 位清零。
11. 如果已寻址到器件，将 ADDEN 位清零，允许接收到的所有数据进入接收缓冲器，并中断 CPU。

图 18-6: EUSART 接收框图



# PIC18F85J11 系列

图 18-7: 异步接收

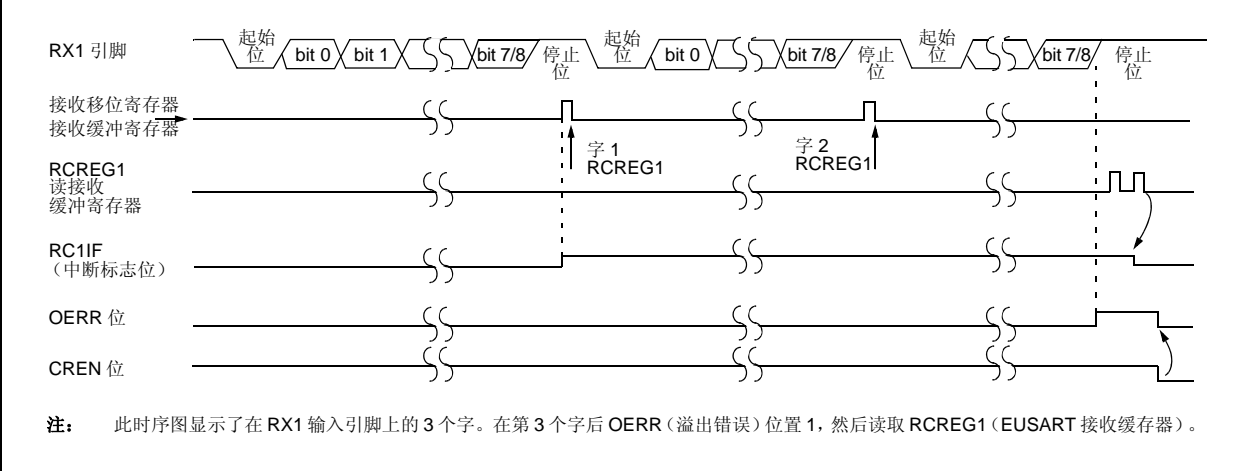


表 18-6: 与异步接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	—	TMR2IF	TMR1IF	59
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	—	TMR2IE	TMR1IE	59
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	—	TMR2IP	TMR1IP	59
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	59
RCREG1	EUSART 接收寄存器								59
TXSTA1	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	59
BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	60
SPBRGH1	EUSART 波特率发生器寄存器的高字节								60
SPBRG1	EUSART 波特率发生器寄存器的低字节								59

图注： — = 未实现，读为 0。异步接收不使用阴影单元。

## 18.3.4 同步间隔字符自动唤醒

在休眠模式下，EUSART 的所有时钟都会暂停。因此，波特率发生器处于非激活状态，并且无法进行正确的字节接收。自动唤醒功能允许在 RX1/DT1 线上有事件发生时唤醒控制器，该功能需要 EUSART 工作在异步模式下。

通过将 WUE 位 (BAUDCON<1>) 置 1，使能自动唤醒功能。该功能启用后，将禁止 RX1/DT1 上的典型接收操作，且 EUSART 保持在空闲状态并监视唤醒事件（与 CPU 运行模式无关）。唤醒事件是指 RX1/DT1 线上发生由高向低的电平跳变。（这与“同步间隔”字符或 LIN/J2602 协议唤醒信号字符的启动条件一致。）

唤醒事件后，模块产生一个 RC1IF 中断。在正常工作模式下，中断会与 Q 时钟同步产生（图 18-8）；如果器件处于休眠模式，则两者不同步（图 18-9）。通过读 RCREG1 寄存器可清除中断条件。

唤醒事件后，当 RX1 线上出现由低向高的电平跳变时，WUE 位自动清零。此时，EUSART 模块将从空闲状态返回正常工作模式，由此用户可知“同步间隔”事件已经结束。

### 18.3.4.1 使用自动唤醒功能的特别注意事项

因为自动唤醒功能是通过检测 RX1/DT1 上的上升沿跳变实现的，所以在停止位前该引脚上任何的状态改变都可能产生错误的字符结束（End-Of-Character, EOC）

信号并导致数据或帧错误。因此，为了确保正确的传输，必须首先发送全 0 字符。对于标准的 RS-232 器件，这可以是 00h（8 位），而对于 LIN/J2602 总线器件则是 000h（12 位）。

另外还必须考虑振荡器起振时间，尤其是在采用起振延时较长的振荡器（即 XT 或 HS 模式）的应用中更要注意这一点。“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，以便使振荡器有充足的时间起振并保证 EUSART 正确初始化。

### 18.3.4.2 使用 WUE 位时的特殊注意事项

用 WUE 和 RC1IF 事件的时序来判断接收数据的有效性可能会引起混淆。如前所述，将 WUE 位置 1 会使 EUSART 进入空闲模式。唤醒事件会通过将 RC1IF 位置 1 产生一个接收中断。此后当 RX1/DT1 出现上升沿时 WUE 位清零。然后通过读 RCREG1 寄存器清除中断条件。一般情况下，RCREG1 中的数据是无效数据，应该丢弃。

WUE 位清零（或仍然置 1）且 RC1IF 标志位置 1 并不能表明 RCREG1 中数据接收是完整的。用户还应该考虑使用固件验证是否完整地接收了数据。

要确保没有丢失有效数据，应检查 RCIDL 位来验证是否还在接收数据。如果不在接收数据，则可将 WUE 位置 1，使器件立即进入休眠模式。

图 18-8: 正常工作模式下的自动唤醒位 (WUE) 时序

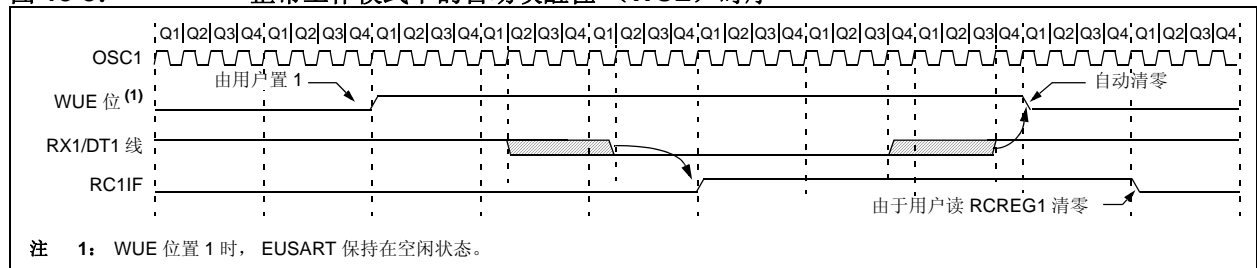
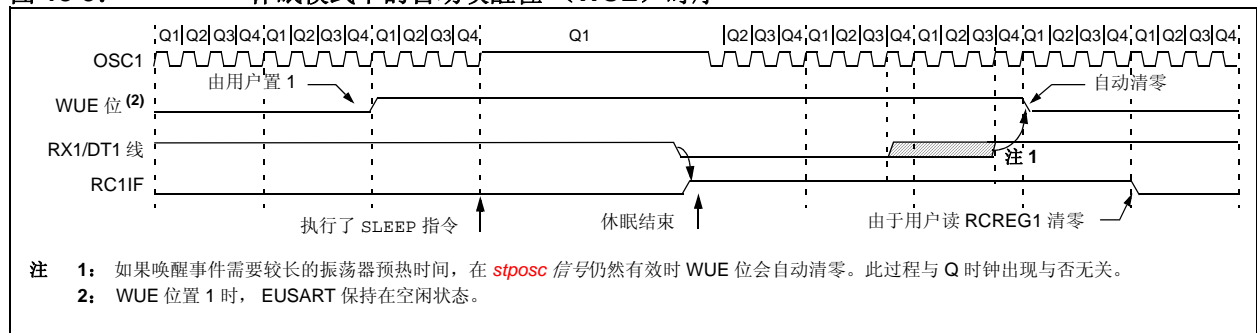


图 18-9: 休眠模式下的自动唤醒位 (WUE) 时序



## 18.3.5 间隔字符序列

增强型 USART 模块能够发送符合 LIN/J2602 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。当发送移位寄存器装有数据时，只要 SENDB 和 TXEN 位（TXSTA1<3> 和 TXSTA1<5>）置 1，就会发送帧间隔字符。请注意写入 TXREG1 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/J2602 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。

请注意在发送间隔字符时写入 TXREG1 的数据值会被忽略。写入仅仅是为了启动正确的序列。

如其在正常发送操作中一样，TRMT 位用于指示发送正在进行还是处于空闲状态。关于间隔字符序列，请参见图 18-10。

### 18.3.5.1 间隔和同步发送序列

下列序列会发送一个报文帧头，其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/J2602 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 TXREG1，启动发送（该值会被忽略）。
4. 将 55h 写入 TXREG1，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。此时，同步字符会以预先配置的模式发送。

当 TX1IF 指出 TXREG1 为空以后，下一个数据字节会被写入 TXREG1。

## 18.3.6 接收间隔字符

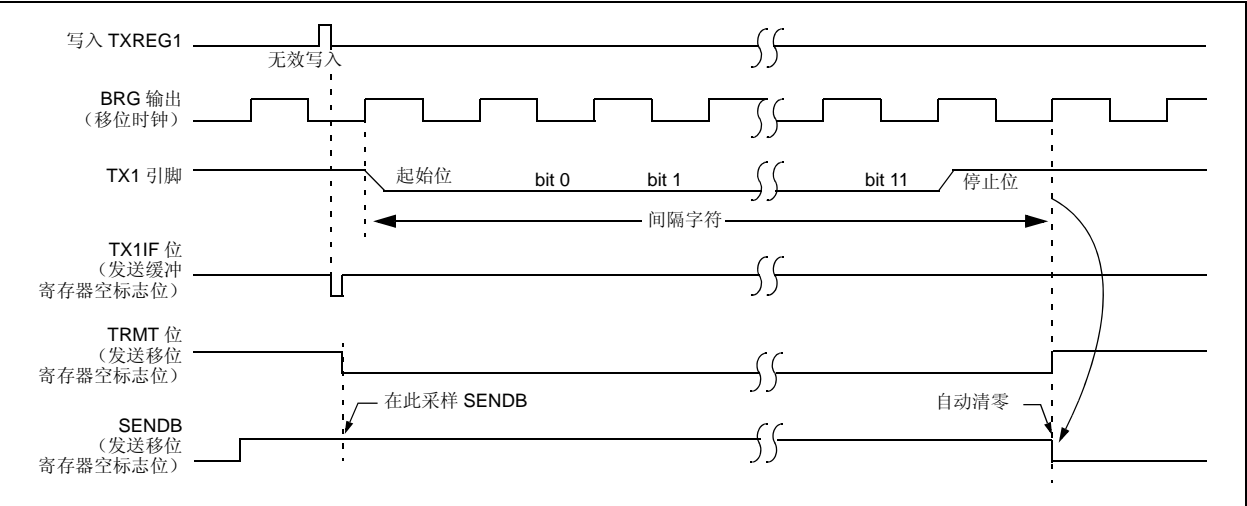
增强型 USART 模块接收间隔字符有两种方法。

第一种方法是强制将波特率配置为典型速率的 9/13。这可以使停止位在正确的采样点（对于间隔字符为起始位之后的 13 位，对于典型数据则是 8 个数据位）产生。

第二种方法使用第 18.3.4 节“同步间隔字符自动唤醒”中描述的自动唤醒功能。通过使能此功能，EUSART 将采样 RX1/DT1 引脚上电平的下两次跳变，产生一个 RC1IF 中断，接收下一个数据字节，并在随后产生另一个中断。

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，用户都可以在检测到 TX1IF 中断时马上将 ABDEN 位置 1。

图 18-10: 发送间隔字符序列



## 18.4 EUSART 同步主模式

将 CSRC 位 (TXSTA1<7>) 置 1 可以进入同步主模式。在此模式中, 数据以半双工方式 (即发送和接收不同时进行) 发送。发送数据时, 禁止接收, 反之亦然。将 SYNC 位 (TXSTA1<4>) 置 1 可进入同步模式。此外, 应将使能位 SPEN (RCSTA1<7>) 置 1, 分别把 TX1 和 RX1 引脚配置为 CK1 (时钟) 和 DT1 (数据) 线。

主模式意味着处理器在 CK1 时钟线上发送主时钟信号。时钟极性是通过 TXCKP 位 (BAUDCON1<4>) 选择的。将 TXCKP 置 1 可将 CK1 的空闲状态设置为高电平, 将该位清零则可将空闲状态设置为低电平。此选项支持将本模块与 Microwire 器件配合使用。

### 18.4.1 EUSART 同步主发送

图 18-3 显示了 EUSART 发送器的框图。发送器的核心是发送 (串行) 移位寄存器 (Transmit Shift Register, TSR)。移位寄存器从读 / 写发送缓冲寄存器 TXREG1 中获取数据。而 TXREG1 寄存器中的数据由软件装入。在前一次装入数据的最后一位发送完成后, 才会向 TSR 寄存器装入新数据。一旦最后一位发送完成, 就会将 TXREG1 寄存器中的新数据 (如果有的话) 装入 TSR。

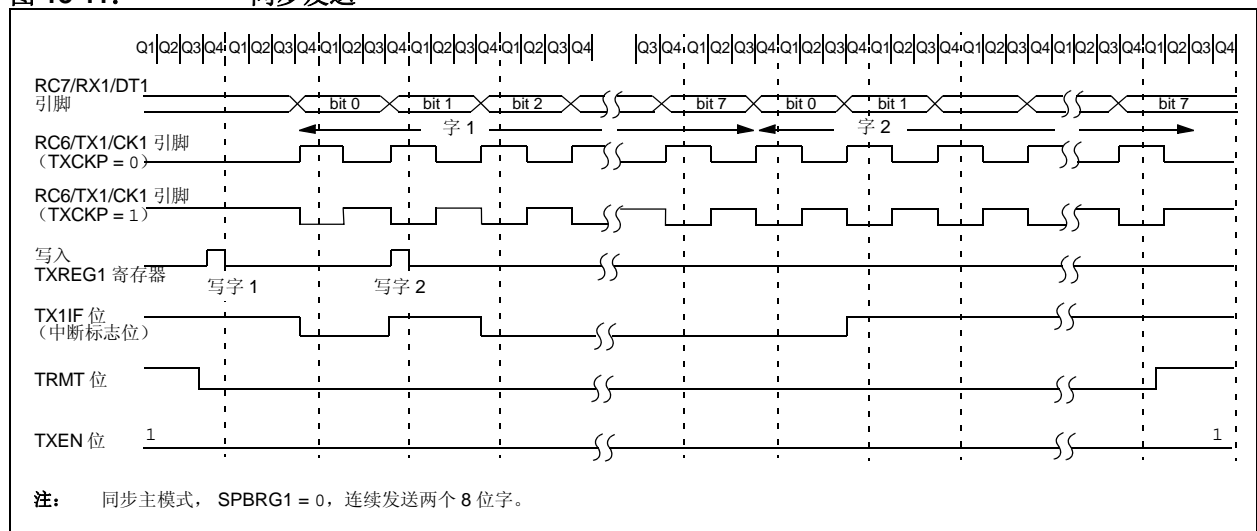
一旦 TXREG1 寄存器向 TSR 寄存器传输了数据 (在 1 个 Tcycle 内发生), TXREG1 寄存器就为空, 同时标志位 TX1IF (PIR1<4>) 置 1。可以通过将中断允许位 TX1IE (PIE1<4>) 置 1 或清零来允许 / 禁止该中断。不管 TX1IE 的状态如何, 只要中断发生, TX1IF 就会置 1, 且不能用软件清零。只有在新数据写入 TXREG1 寄存器时, TX1IF 才会复位。

TX1IF 标志位表示的是 TXREG1 寄存器的状态, 而另一个位 TRMT (TXSTA1<1>) 则指示 TSR 寄存器的状态。TRMT 是只读位, 它在 TSR 寄存器为空时被置 1。TRMT 位与任何中断逻辑均无关联, 因此要判断 TSR 寄存器是否为空, 用户只能对此位进行轮询。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。

设置同步主发送操作的步骤如下:

1. 初始化 SPBRGH1:SPBRG1 寄存器, 设置合适的波特率。按需要将 BRG16 位置 1 或清零, 以获得目标波特率。
2. 通过将 SYNC、SPEN 和 CSRC 位置 1, 使能同步主串口。
3. 如果需要中断, 将中断允许位 TX1IE 置 1。
4. 若需要发送 9 位数据, 将 TX9 位置 1。
5. 将 TXEN 位置 1, 使能发送。
6. 如果选择发送 9 位数据, 应该将第 9 位数据装入 TX9D 位。
7. 将数据装入 TXREG1 寄存器, 启动发送。
8. 若想使用中断, 请确保将 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 置 1。

图 18-11: 同步发送



# PIC18F85J11 系列

图 18-12: 同步发送（由 TXEN 位控制）

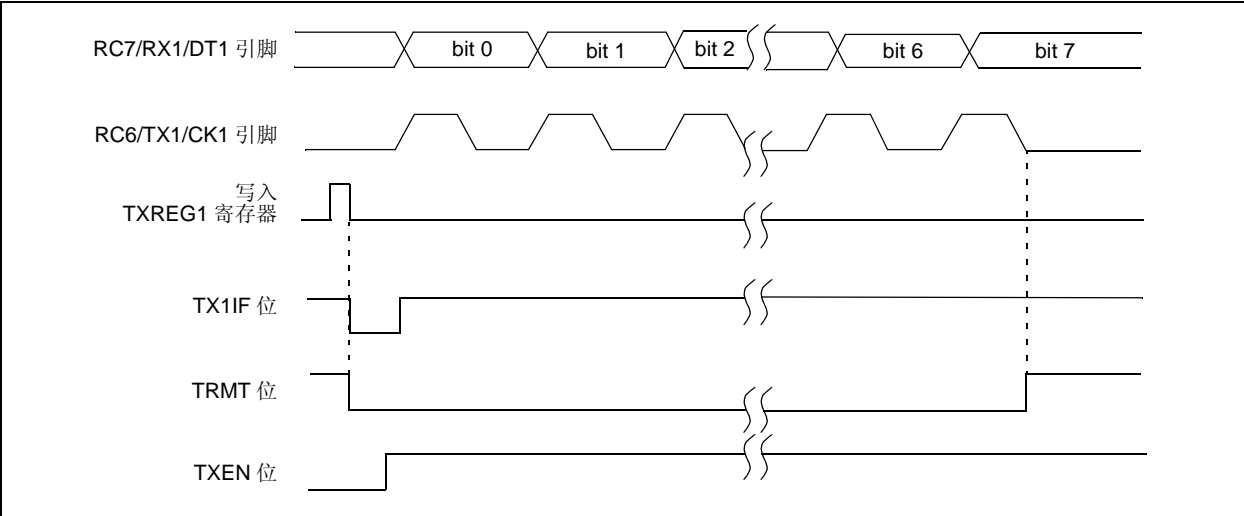


表 18-7: 与同步主发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	—	TMR2IF	TMR1IF	59
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	—	TMR2IE	TMR1IE	59
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	—	TMR2IP	TMR1IP	59
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	59
TXREG1	EUSART 发送寄存器								59
TXSTA1	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	59
BAUDCON1	ABDOVF	RCIDL	RxDTP	TXCKP	BRG16	—	WUE	ABDEN	60
SPBRGH1	EUSART 波特率发生器寄存器的高字节								60
SPBRG1	EUSART 波特率发生器寄存器的低字节								59
LATG	U2OD	U1OD	—	LATG4	LATG3	LATG2	LATG1	LATG0	60

图注： — = 未实现，读为 0。同步主发送不使用阴影单元。



## 18.4.2 EUSART 同步主接收

一旦选择了同步模式，只要将单字节接收使能位 **SREN** (**RCSTA1<5>**) 或连续接收使能位 **CREN** (**RCSTA1<4>**) 置 1，即可使能接收。在时钟的下降沿采样 **RX1** 引脚上的数据。

如果将使能位 **SREN** 置 1，则只接收单个字。如果将使能位 **CREN** 置 1，则会连续接收数据，直到将 **CREN** 位清零。如果两个位均被置 1，则 **CREN** 具有优先权。

设置同步主接收操作的步骤如下：

1. 初始化 **SPBRGH1:SPBRG1** 寄存器，设置合适的波特率。按需要将 **BRG16** 位置 1 或清零，以获得目标波特率。
2. 通过将 **SYNC**、**SPEN** 和 **CSRC** 位置 1，使能同步主串口。

3. 确保将 **CREN** 和 **SREN** 位清零。
4. 如果需要中断，将中断允许位 **RC1IE** 置 1。
5. 若需要接收 9 位数据，将 **RX9** 位置 1。
6. 若需要单字节接收，将 **SREN** 位置 1；若需要连续接收，将 **CREN** 位置 1。
7. 当接收完成时中断标志位 **RC1IF** 将置 1，此时如果允许位 **RC1IE** 已置 1，还将产生一个中断。
8. 读 **RCSTA1** 寄存器以获取第 9 位数据（如果已使能），并判断是否在接收过程中发生了错误。
9. 通过读 **RCREG1** 寄存器来读取接收到的 8 位数据。
10. 如果发生错误，通过将 **CREN** 位清零来清除错误。
11. 若想使用中断，请确保将 **INTCON** 寄存器中的 **GIE** 和 **PEIE** 位 (**INTCON<7:6>**) 置 1。

图 18-13: 同步主接收模式（由 **SREN** 位控制）

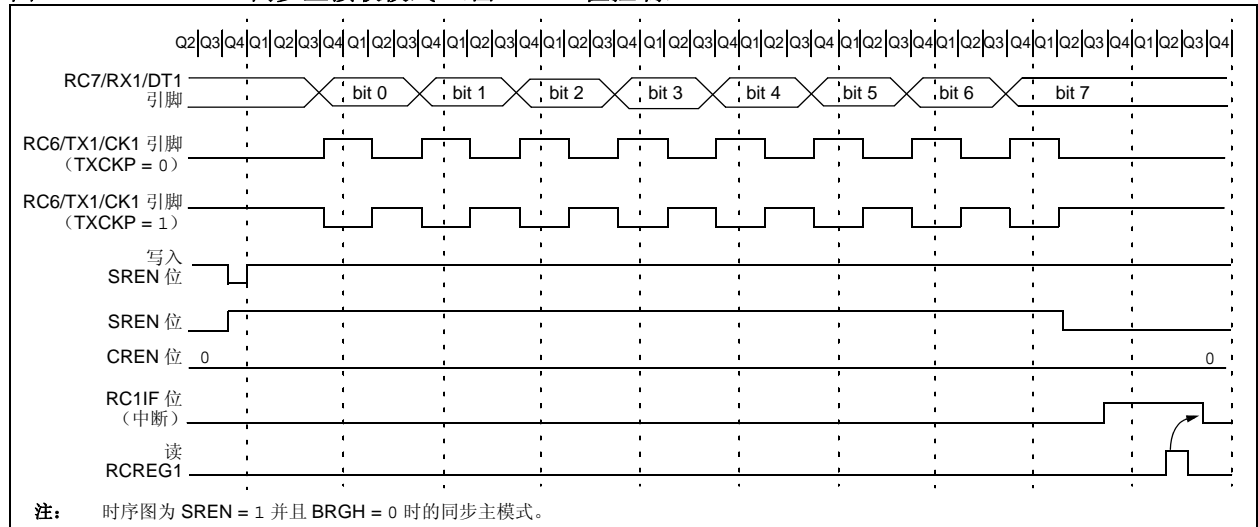


表 18-8: 与同步主接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	—	TMR2IF	TMR1IF	59
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	—	TMR2IE	TMR1IE	59
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	—	TMR2IP	TMR1IP	59
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	59
RCREG1	EUSART 接收寄存器								59
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	59
BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	60
SPBRGH1	EUSART 波特率发生器寄存器的高字节								60
SPBRG1	EUSART 波特率发生器寄存器的低字节								59

图注：— = 未实现，读为 0。同步主接收不使用阴影单元。

# PIC18F85J11 系列

## 18.5 EUSART 同步从模式

将 CSRC (TXSTA<7>) 位清零可进入同步从模式。此模式与同步主模式的区别在于移位时钟由 CK1 引脚上的外部时钟提供 (主模式下由内部时钟提供)。这使得器件能在任何低功耗模式下发送或接收数据。

### 18.5.1 EUSART 同步从发送

除了休眠模式以外，同步主模式和从模式的工作方式是完全相同的。

如果向 TXREG1 寄存器写入两个字，然后执行 SLEEP 指令，则将发生以下事件：

- 第一个字立即传送到 TSR 寄存器进行发送。
- 第二个字仍保留在 TXREG1 寄存器中。
- 不会将标志位 TX1IF 置 1。
- 当第一个字移出 TSR 后，TXREG1 寄存器将把第二个字送入 TSR，同时将标志位 TX1IF 置 1。
- 如果中断允许位 TX1IE 已置 1，中断将把器件从休眠状态唤醒。如果允许了全局中断，程序则会跳转到中断向量处执行。

设置同步从发送操作的步骤如下：

- 通过将 SYNC 和 SPEN 位置 1 并将 CSRC 位清零使能同步从串口。
- 将 CREN 和 SREN 位清零。
- 如果需要中断，将中断允许位 TX1IE 置 1。
- 若需要发送 9 位数据，将 TX9 位置 1。
- 将使能位 TXEN 置 1，使能发送。
- 如果选择发送 9 位数据，应该将第 9 位数据装入 TX9D 位。
- 将数据装入 TXREG1 寄存器，启动发送。
- 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 置 1。

表 18-9: 与同步从发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	—	TMR2IF	TMR1IF	59
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	—	TMR2IE	TMR1IE	59
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	—	TMR2IP	TMR1IP	59
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	59
TXREG1	EUSART 发送寄存器								59
TXSTA1	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	59
BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	60
SPBRGH1	EUSART 波特率发生器寄存器的高字节								60
SPBRG1	EUSART 波特率发生器寄存器的低字节								59
LATG	U2OD	U1OD	—	LATG4	LATG3	LATG2	LATG1	LATG0	60

图注： — = 未实现，读为 0。同步从发送不使用阴影单元。

## 18.5.2 EUSART 同步从接收

除了休眠模式、任何空闲模式以及在从模式下为无关位的 SREN 位以外，同步主模式和同步从模式的工作方式完全相同。

如果在进入休眠或空闲模式前将 CREN 位置 1 使能接收，那么在低功耗模式下可以接收到一个数据字。接收到该字后，RSR 寄存器将把数据传输到 RCREG1 寄存器。如果中断允许位 RC1IE 已置 1，则产生的中断将把器件从低功耗模式唤醒。如果允许了全局中断，程序则会跳转到中断向量处执行。

设置同步从接收操作的步骤如下：

1. 通过将 SYNC 和 SPEN 位置 1 并将 CSRC 位清零使能同步主串口。
2. 如果需要中断，将中断允许位 RC1IE 置 1。
3. 若需要接收 9 位数据，将 RX9 位置 1。
4. 将使能位 CREN 置 1，使能接收。
5. 当接收完成时，RC1IF 标志位将被置 1。如果中断允许位 RC1IE 已置 1，还将产生一个中断。
6. 读 RCSTA1 寄存器以获取第 9 位数据（如果已使能），并判断是否在接收过程中发生了错误。
7. 通过读 RCREG1 寄存器来读取接收到的 8 位数据。
8. 如果发生错误，通过将 CREN 位清零来清除错误。
9. 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）置 1。

表 18-10: 与同步从接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	—	TMR2IF	TMR1IF	59
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	—	TMR2IE	TMR1IE	59
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	—	TMR2IP	TMR1IP	59
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	59
RCREG1	EUSART 接收寄存器								59
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	59
BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	60
SPBRGH1	EUSART 波特率发生器寄存器的高字节								60
SPBRG1	EUSART 波特率发生器寄存器的低字节								59

图注： — = 未实现，读为 0。同步从接收不使用阴影单元。

# PIC18F85J11 系列

---

注:

## 19.0 可寻址通用同步异步收发器 (AUSART)

可寻址通用同步异步收发器 (Addressable Universal Synchronous Asynchronous Receiver Transmitter, AUSART) 模块与前一章中讨论的增强型 USART 模块在功能上非常相似。它提供了与外部器件进行串行通信的额外通道, 适用于那些不需要自动波特率检测或 LIN/J2602 总线支持的情况。

AUSART 可配置为以下几种工作模式:

- 全双工异步模式
- 半双工同步主模式
- 半双工同步从模式

AUSART 模块的引脚分别与 PORTG (RG1/TX2/CK2 和 RG2/RX2/DT2) 功能复用。为了将这些引脚配置为 AUSART:

- SPEN (RCSTA<7>) 位必须置 1 (= 1)
- TRISC<2> 位必须置 1 (= 1)
- 要使模块工作在同步 / 异步主模式, TRISG<1> 位必须清零 (= 0)
- 要使模块工作在同步从模式, TRISC<1> 位必须置 1 (= 1)

**注:** AUSART 控制逻辑在需要时会自动将引脚从输入重新配置为输出。

TX2 输出引脚的驱动器也可选择配置为漏极开路输出。该功能支持通过外部上拉电阻将该引脚上的电平拉至较高级别, 且可实现输出与外部电路通信而不需要额外电平转换器。

U2OD 位 (LATG<7>) 控制漏极开路输出选项。将该位置 1 可将引脚配置为漏极开路操作。

### 19.1 控制寄存器

可寻址 USART 模块的操作由 TXSTA2 和 RCSTA2 两个寄存器控制。寄存器 19-1 和寄存器 19-2 分别对这两个寄存器进行了详细说明。

# PIC18F85J11 系列

寄存器 19-1: TXSTA2: AUSART 发送状态和控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN <sup>(1)</sup>	SYNC	—	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	<b>CSRC:</b> 时钟源选择位 <u>异步模式:</u> 无关位。 <u>同步模式:</u> 1 = 主模式 (时钟来自内部 BRG) 0 = 从模式 (时钟来自外部时钟源)
bit 6	<b>TX9:</b> 9 位发送使能位 1 = 选择 9 位发送 0 = 选择 8 位发送
bit 5	<b>TXEN:</b> 发送使能位 <sup>(1)</sup> 1 = 使能发送 0 = 禁止发送
bit 4	<b>SYNC:</b> AUSART 模式选择位 1 = 同步模式 0 = 异步模式
bit 3	<b>未实现:</b> 读为 0
bit 2	<b>BRGH:</b> 高波特率选择位 <u>异步模式:</u> 1 = 高速 0 = 低速 <u>同步模式:</u> 在此模式下未使用。
bit 1	<b>TRMT:</b> 发送移位寄存器状态位 1 = TSR 空 0 = TSR 满
bit 0	<b>TX9D:</b> 发送数据的第 9 位 该位可以是地址 / 数据位或奇偶校验位。

注 1: 同步模式下 SREN/CREN 的优先级高于 TXEN。

**寄存器 19-2: RCSTA2: AUSART 接收状态和控制寄存器**

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:** 串口使能位  
1 = 使能串口 (将 RX2/DT2 和 TX2/CK2 (TXEN = 1) 引脚配置为串口引脚)  
0 = 禁止串口 (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:  
无关位。  
同步主模式:  
1 = 使能单字节接收  
0 = 禁止单字节接收  
此位在接收完成后清零。  
同步从模式:  
无关位。
- bit 4      CREN:** 连续接收使能位  
异步模式:  
1 = 使能接收器  
0 = 禁止接收器  
同步模式:  
1 = 使能连续接收, 直到使能位 CREN 清零 (CREN 的优先级高于 SREN)  
0 = 禁止连续接收
- bit 3      ADDEN:** 地址检测使能位  
9 位异步模式 (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, 接收到所有字节并且第 9 位可用作奇偶校验位  
9 位异步模式 (RX9 = 0):  
无关位。
- bit 2      FERR:** 帧错误位  
1 = 帧错误 (可以通过读 RCREGx 寄存器清除, 并接收下一个有效字节)  
0 = 无帧错误
- bit 1      OERR:** 溢出错误位  
1 = 溢出错误 (可以通过清零 CREN 位清除)  
0 = 无溢出错误
- bit 0      RX9D:** 接收数据的第 9 位  
该位可以是地址 / 数据位或奇偶校验位, 必须由用户固件计算得到。

# PIC18F85J11 系列

## 19.2 AUSART 波特率发生器 (BRG)

BRG 是一个专用的 8 位发生器，支持 AUSART 的异步和同步模式。

SPBRG2 寄存器控制自由运行定时器的周期。在异步模式下，BRGH (TXSTA<2>) 位也控制波特率。在同步模式下，将忽略 BRGH 位。表 19-1 所示为不同 AUSART 模式的波特率计算公式，但仅适用于主模式（由内部产生时钟信号）。

给定目标波特率和 Fosc 的情况下，可以使用表 19-1 中的公式计算 SPBRG2 寄存器的最近似整数值，从而确定波特率误差。例 19-1 给出了一个计算示例。表 19-2 给出

了各种异步模式下典型的波特率和误差值。使用高波特率 (BRGH = 1) 有利于减少波特率误差，或者在快速振荡频率条件下实现低波特率。

向 SPBRG2 寄存器写入新值会引起 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。

### 19.2.1 在功耗管理模式下的操作

器件时钟用于产生所需的波特率。当进入一种功耗管理模式时，新时钟源可能会工作在与先前不同的频率下。这可能需要调整 SPBRG2 寄存器中的值。

### 19.2.2 采样

择多检测电路对 RX2 引脚数据采样三次，以判定 RX2 引脚上出现的是高电平还是低电平。

表 19-1: 波特率公式

配置位		BRG/AUSART 模式	波特率公式
SYNC	BRGH		
0	0	异步	$F_{osc}/[64 (n + 1)]$
0	1	异步	$F_{osc}/[16 (n + 1)]$
1	x	同步	$F_{osc}/[4 (n + 1)]$

图注: x = 任意值, n = SPBRG2 寄存器的值

例 19-1: 计算波特率误差

针对工作在异步模式下、工作频率 Fosc 为 16 MHz、目标波特率为 9600，BRGH = 0 的器件：	
目标波特率	= $F_{osc}/(64 ([SPBRG2] + 1))$
求解 SPBRG2:	
X	= $((F_{osc}/\text{Desired Baud Rate})/64) - 1$
	= $((16000000/9600)/64) - 1$
	= $[25.042] = 25$
波特率计算结果	= $16000000/(64 (25 + 1))$
	= 9615
误差	= $(\text{Calculated Baud Rate} - \text{Desired Baud Rate})/\text{Desired Baud Rate}$
	= $(9615 - 9600)/9600 = 0.16\%$

表 19-2: 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	复位值所在页
TXSTA2	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	61
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	61
SPBRG2	AUSART 波特率发生器寄存器								61

图注: — = 未实现单元，读为 0。BRG 不使用阴影单元。



表 19-3: 异步模式下的波特率

目标 波特率 (K)	BRGH = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际 波特率 (K)	误差 %	SPBRG 值 (十进制)	实际 波特率 (K)	误差 %	SPBRG 值 (十进制)	实际 波特率 (K)	误差 %	SPBRG 值 (十进制)	实际 波特率 (K)	误差 %	SPBRG 值 (十进制)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1.201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2.403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9.615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	125.00	8.51	4	104.167	-9.58	2	78.125	-32.18	1	—	—	—

目标 波特率 (K)	BRGH = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际 波特率 (K)	误差 %	SPBRG 值 (十进制)	实际 波特率 (K)	误差 %	SPBRG 值 (十进制)	实际 波特率 (K)	误差 %	SPBRG 值 (十进制)
0.3	0.300	0.16	207	0.300	-0.16	103	0.300	-0.16	51
1.2	1.202	0.16	51	1.201	-0.16	25	1.201	-0.16	12
2.4	2.404	0.16	25	2.403	-0.16	12	—	—	—
9.6	8.929	-6.99	6	—	—	—	—	—	—
19.2	20.833	8.51	2	—	—	—	—	—	—
57.6	62.500	8.51	0	—	—	—	—	—	—
115.2	62.500	-45.75	0	—	—	—	—	—	—

目标 波特率 (K)	BRGH = 1											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	实际 波特率 (K)	误差 %	SPBRG 值 (十进制)	实际 波特率 (K)	误差 %	SPBRG 值 (十进制)	实际 波特率 (K)	误差 %	SPBRG 值 (十进制)	实际 波特率 (K)	误差 %	SPBRG 值 (十进制)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	1.73	255	2.403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

目标 波特率 (K)	BRGH = 1								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	实际 波特率 (K)	误差 %	SPBRG 值 (十进制)	实际 波特率 (K)	误差 %	SPBRG 值 (十进制)	实际 波特率 (K)	误差 %	SPBRG 值 (十进制)
0.3	—	—	—	—	—	—	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.00	8.51	1	—	—	—	—	—	—

## 19.3 AUSART 异步模式

通过将 SYNC 位 (TXSTA2<4>) 清零可选择异步工作模式。在此模式下, AUSART 使用标准的不归零 (NRZ) 格式 (1 个起始位, 8 个或 9 个数据位, 1 个停止位)。最常用的数据格式为 8 位。片上专用 8 位波特率发生器可借助于振荡器产生标准波特率频率。

AUSART 首先发送和接收的是最低有效位。AUSART 的发送器和接收器在功能上是独立的, 但采用相同的数据格式和波特率。波特率发生器可根据 BRGH 位 (TXSTA2<2>) 产生两种不同的波特率时钟, 频率分别为移位速率的 16 倍和 64 倍。硬件不支持奇偶校验, 但可以用软件实现, 校验位保存在第 9 个数据位中。

当工作在异步模式时, AUSART 模块包括以下重要组成部分:

- 波特率发生器
- 采样电路
- 异步发送器
- 异步接收器

### 19.3.1 AUSART 异步发送器

图 19-1 显示了 AUSART 发送器的框图。发送器的核心是发送 (串行) 移位寄存器 (TSR)。移位寄存器从读/写发送缓冲寄存器 TXREG2 中获取数据。TXREG2 寄存器中的数据由软件装入。直到前一次装入的停止位已被发送, 才会向 TSR 寄存器装入新数据。一旦停止位发送完毕, TXREG2 寄存器中的新数据 (如果有的话) 就会被装入 TSR。

一旦 TXREG2 寄存器向 TSR 寄存器传输了数据 (在 1 个 Tcy 内发生), TXREG2 寄存器就为空, 同时标志位 TX2IF (PIR3<4>) 置 1。可以通过将中断允许位 TX2IE (PIE3<4>) 置 1 或清零来允许/禁止该中断。不管 TX2IE 的状态如何, 只要中断发生, TX2IF 就会置 1 并且不能用软件清零。TX2IF 不会在 TXREG2 装入新数据时立即被清零, 而是在装入指令后的第二个指令周期被清零。因此在 TXREG2 装入新数据后立即查询 TX2IF, 会返回无效结果。

TX2IF 指示的是 TXREG2 寄存器的状态, 而另一个位 TRMT (TXSTA2<1>) 则指示 TSR 寄存器的状态。TRMT 是只读位, 它在 TSR 寄存器为空时被置 1。TRMT 位与任何中断逻辑均无关联, 因此要确定 TSR 寄存器是否为空, 用户只能对此位进行轮询。

- 注 1:** 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。

**2:** 当使能位 TXEN 置 1 时, 标志位 TX2IF 也置 1。

设置异步发送操作的步骤如下:

1. 初始化 SPBRG2 寄存器, 设置合适的波特率。按需要将 BRGH 位置 1 或清零, 以获得目标波特率。
2. 通过将 SYNC 位清零并将 SPEN 位置 1 使能异步串口。
3. 如果需要中断, 将中断允许位 TX2IE 置 1。
4. 若需要发送 9 位数据, 将发送位 TX9 置 1。发送的第 9 位可以是地址位也可以是数据位。
5. 通过将 TXEN 位置 1 使能发送, 此操作同时也会将 TX2IF 位置 1。
6. 如果选择发送 9 位数据, 将第 9 位数据装入 TX9D 位。
7. 将数据装入 TXREG2 寄存器 (开始发送)。
8. 若想使用中断, 请确保将 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 置 1。

图 19-1: AUSART 发送框图

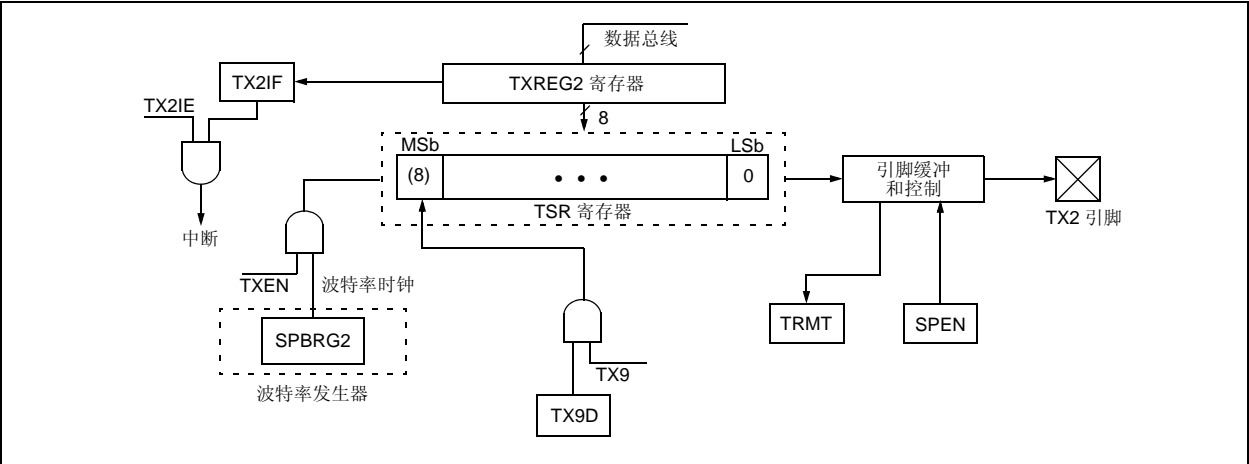


图 19-2: 异步发送

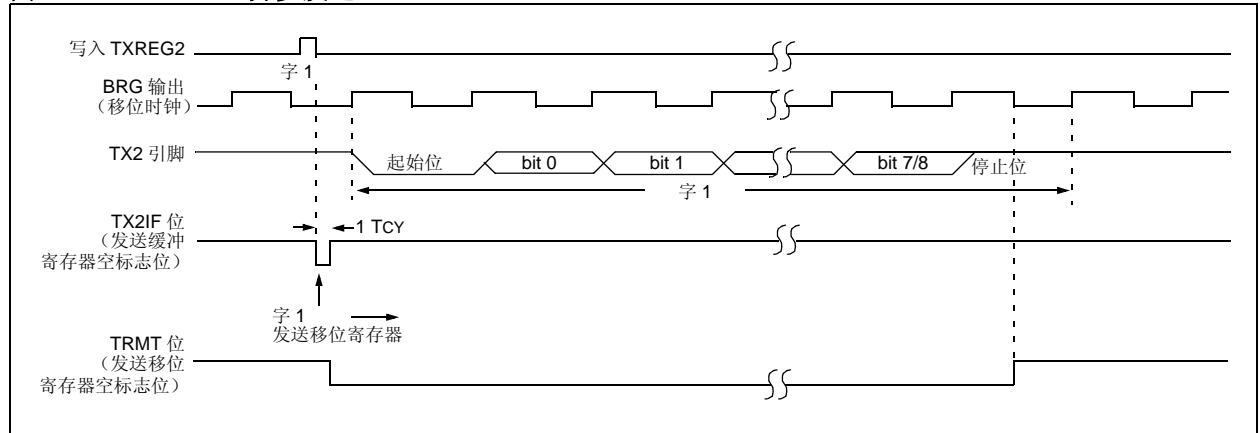


图 19-3: 异步发送（背对背）

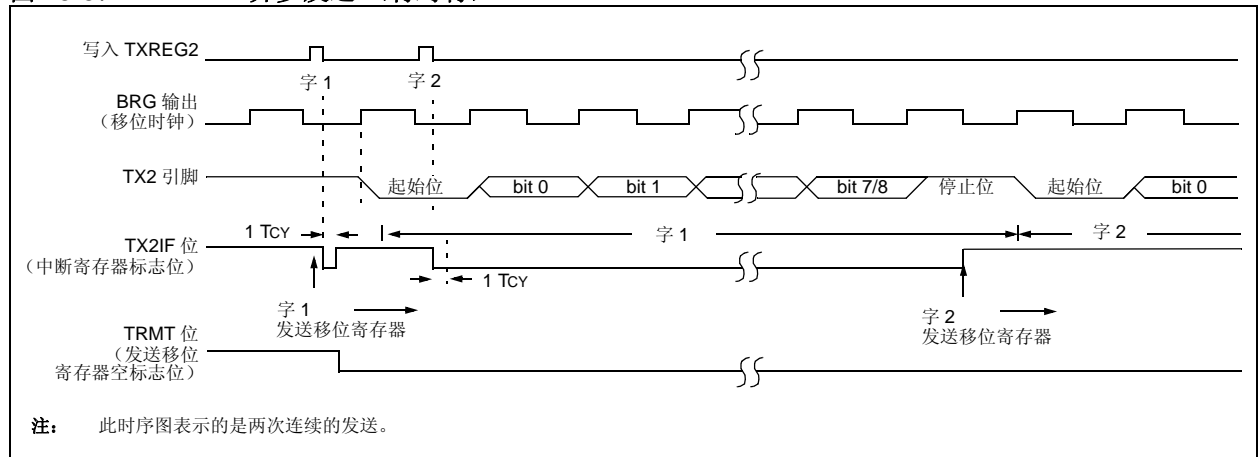


表 19-4: 与异步发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR3	—	—	RC2IF	TX2IF	—	CCP2IF	CCP1IF	—	59
PIE3	—	—	RC2IE	TX2IE	—	CCP2IE	CCP1IE	—	59
IPR3	—	—	RC2IP	TX2IP	—	CCP2IP	CCP1IP	—	59
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	61
TXREG2	AUSART 发送寄存器								61
TXSTA2	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	61
SPBRG2	AUSART 波特率发生器寄存器								61
LATG	U2OD	U1OD	—	LATG4	LATG3	LATG2	LATG1	LATG0	60

图注： — = 未实现，读为 0。异步发送不使用阴影单元。

## 19.3.2 AUSART 异步接收

图 19-4 显示了接收器的框图。在 RX2 引脚上接收数据，并驱动数据恢复电路。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，而主接收串行移位器的工作频率等于比特率或  $F_{osc}$ 。此模式通常用于 RS-232 系统。

设置异步接收操作的步骤如下：

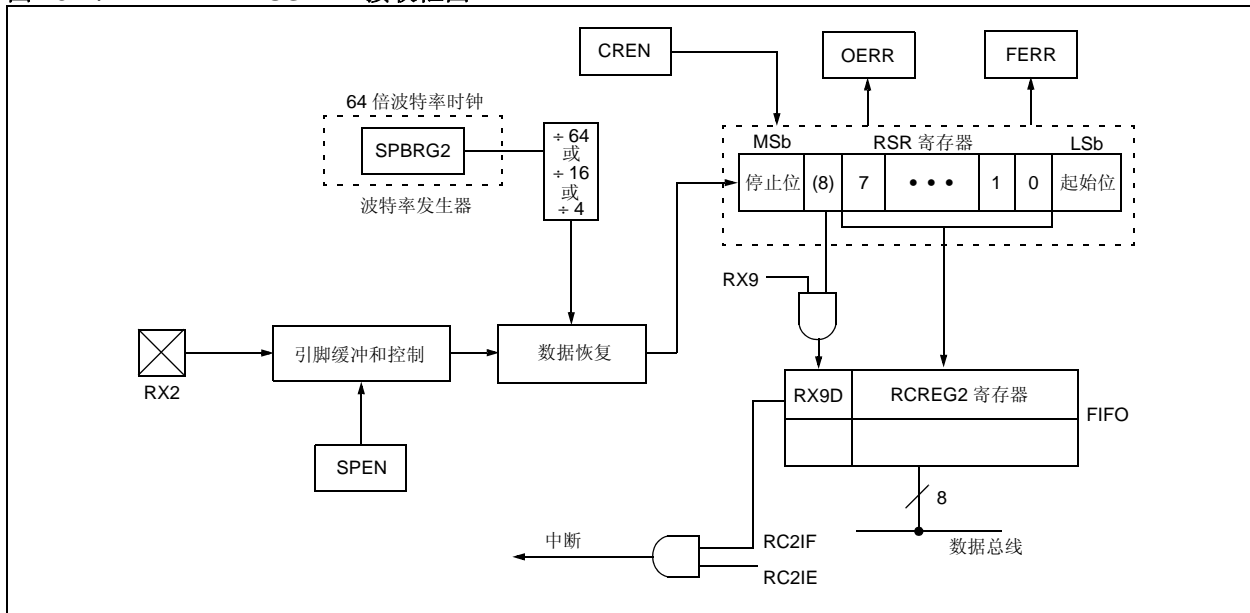
1. 初始化 SPBRG2 寄存器，设置合适的波特率。按需要将 BRGH 位置 1 或清零，以获得目标波特率。
2. 通过将 SYNC 位清零并将 SPEN 位置 1 使能异步串口。
3. 如果需要中断，将中断允许位 RC2IE 置 1。
4. 若需要接收 9 位数据，将 RX9 位置 1。
5. 通过将 CREN 位置 1，使能接收。
6. 当接收完成时标志位 RC2IF 将置 1，此时如果中断允许位 RC2IE 已置 1，将会产生中断。
7. 读 RCSTA2 寄存器以获取第 9 位数据（如果已使能），并判断是否在接收过程中发生了错误。
8. 通过读 RCREG2 寄存器来读取接收到的 8 位数据。
9. 如果发生错误，通过将使能位 CREN 清零来清除错误。
10. 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）置 1。

## 19.3.3 设置带有地址检测功能的 9 位模式

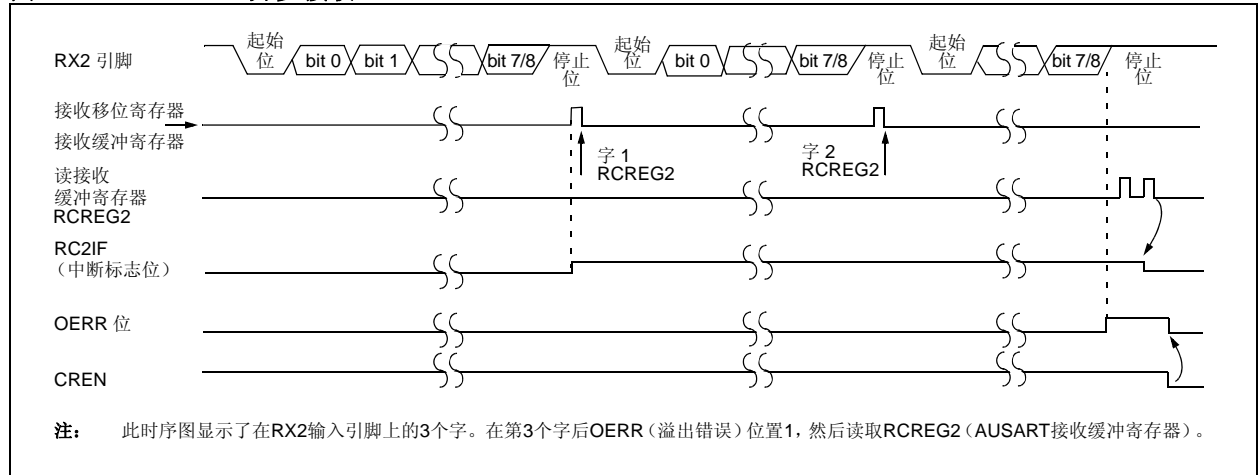
此模式通常用于 RS-485 系统。按如下步骤设置带有地址检测功能的异步接收操作：

1. 初始化 SPBRG2 寄存器，设置合适的波特率。按需要将 BRGH 和 BRG16 位置 1 或清零，以获得目标波特率。
2. 通过将 SYNC 位清零并将 SPEN 位置 1 使能异步串口。
3. 若需要中断，请将 RCEN 位置 1 并使用 RC2IP 位设置所需优先级。
4. 将 RX9 位置 1，使能 9 位接收。
5. 将 ADDEN 位置 1，使能地址检测。
6. 将 CREN 位置 1，使能接收。
7. 当接收完成时 RC2IF 位将被置 1。此时如果 RC2IE 和 GIE 位已置 1，还将响应中断。
8. 读 RCSTA2 寄存器判断在接收时是否发生了错误，同时读取第 9 位数据（如果适用）。
9. 读 RCREG2 来判断是否正在对器件进行寻址。
10. 如果发生错误，将 CREN 位清零。
11. 如果已经寻址到器件，将 ADDEN 位清零，允许接收到的所有数据进入接收缓冲器，并中断 CPU。

图 19-4: AUSART 接收框图



**图 19-5: 异步接收**



**表 19-5: 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	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR3	—	—	RC2IF	TX2IF	—	CCP2IF	CCP1IF	—	59
PIE3	—	—	RC2IE	TX2IE	—	CCP2IE	CCP1IE	—	59
IPR3	—	—	RC2IP	TX2IP	—	CCP2IP	CCP1IP	—	59
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	61
RCREG2	AUSART 接收寄存器								61
TXSTA2	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	61
SPBRG2	AUSART 波特率发生器寄存器								61

**图注:** — = 未实现单元, 读为 0。异步接收不使用阴影单元。

## 19.4 AUSART 同步主模式

将 CSRC 位 (TXSTA2<7>) 置 1 可以进入同步主模式。在此模式中, 数据以半双工方式 (即发送和接收不同时进行) 发送。发送数据时, 禁止接收, 反之亦然。将 SYNC 位 (TXSTA2<4>) 置 1 可进入同步模式。此外, 应将使能位 SPEN (RCSTA2<7>) 置 1, 分别把 TX2 和 RX2 引脚配置为 CK2 (时钟) 和 DT2 (数据) 线。

主模式意味着处理器在 CK2 时钟线上发送主时钟信号。

### 19.4.1 AUSART 同步主发送

图 19-1 显示了 AUSART 发送器的框图。发送器的核心是发送 (串行) 移位寄存器 (TSR)。移位寄存器从读 / 写发送缓冲寄存器 TXREG2 中获取数据。而 TXREG2 寄存器中的数据由软件装入。在前一次装入数据的最后一位发送完成后, 才会向 TSR 寄存器装入新数据。一旦最后一位发送完成, 就会将 TXREG2 寄存器中的新数据 (如果有的话) 装入 TSR。

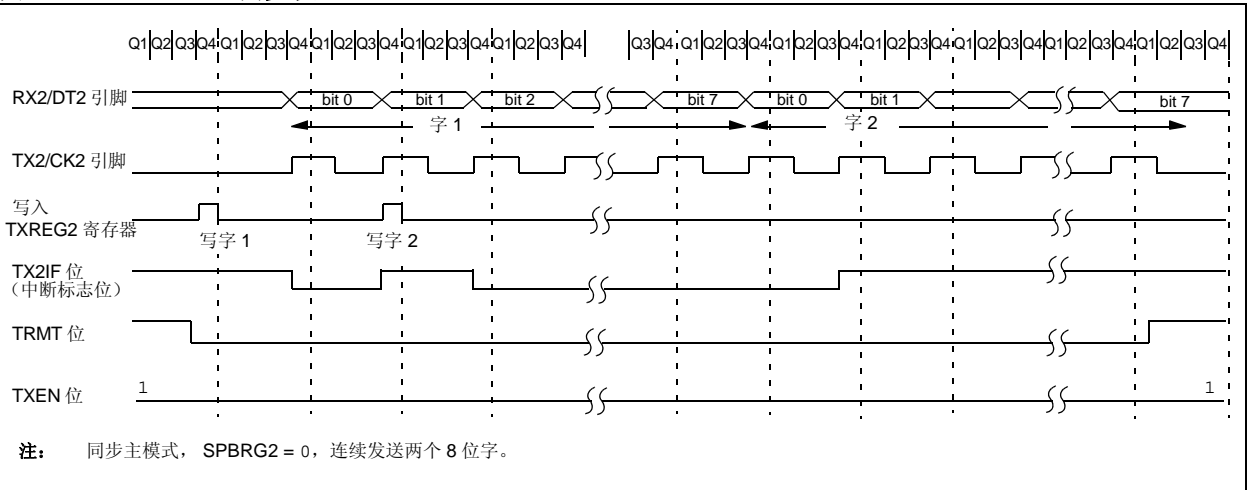
一旦 TXREG2 寄存器向 TSR 寄存器传输了数据 (在 1 个 Tcycle 内发生), TXREG2 寄存器就为空, 同时标志位 TX2IF (PIR3<4>) 置 1。可以通过将中断允许位 TX2IE (PIE3<4>) 置 1 或清零来允许 / 禁止该中断。不管 TX2IE 的状态如何, 只要中断发生, TX2IF 就会置 1, 且不能用软件清零。只有在新数据写入 TXREG2 寄存器时, TX2IF 才会复位。

标志位 TX2IF 指示的是 TXREG2 寄存器的状态, 而另一个位 TRMT (TXSTA2<1>) 则指示 TSR 寄存器的状态。TRMT 是只读位, 它在 TSR 寄存器为空时被置 1。TRMT 位与任何中断逻辑均无关联, 因此要判断 TSR 寄存器是否为空, 用户只能对此位进行轮询。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。

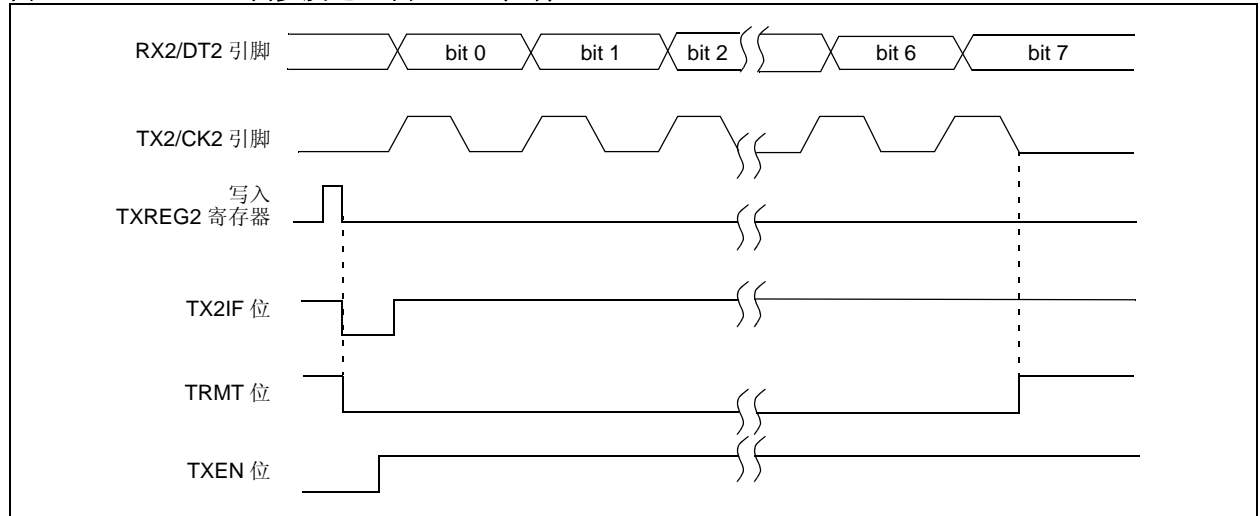
设置同步主发送操作的步骤如下:

1. 初始化 SPBRG2 寄存器, 设置合适的波特率。
2. 通过将 SYNC、SPEN 和 CSRC 位置 1, 使能同步主串口。
3. 如果需要中断, 将中断允许位 TX2IE 置 1。
4. 若需要发送 9 位数据, 将 TX9 位置 1。
5. 将 TXEN 位置 1, 使能发送。
6. 如果选择发送 9 位数据, 应该将第 9 位数据装入 TX9D 位。
7. 将数据装入 TXREG2 寄存器, 启动发送。
8. 若想使用中断, 请确保将 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 置 1。

图 19-6: 同步发送



**图 19-7:** 同步发送（由 TXEN 控制）



**表 19-6:** 与同步主发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR3	—	—	RC2IF	TX2IF	—	CCP2IF	CCP1IF	—	59
PIE3	—	—	RC2IE	TX2IE	—	CCP2IE	CCP1IE	—	59
IPR3	—	—	RC2IP	TX2IP	—	CCP2IP	CCP1IP	—	59
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	61
TXREG2	AUSART 发送寄存器								61
TXSTA2	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	61
SPBRG2	AUSART 波特率发生器寄存器								61
LATG	U2OD	U1OD	—	LATG4	LATG3	LATG2	LATG1	LATG0	60

**图注:** — = 未实现，读为 0。同步主发送不使用阴影单元。

# PIC18F85J11 系列

## 19.4.2 AUSART 同步主接收

一旦选择了同步模式，只要将单字节接收使能位 **SREN** (**RCSTA2<5>**) 或连续接收使能位 **CREN** (**RCSTA2<4>**) 置 1，即可使能接收。在时钟的下降沿采样 **RX2** 引脚上的数据。

如果将使能位 **SREN** 置 1，则只接收单个字。如果将使能位 **CREN** 置 1，则会连续接收数据，直到将 **CREN** 位清零。如果两个位均被置 1，则 **CREN** 具有优先权。

设置同步主接收操作的步骤如下：

1. 初始化 **SPBRG2** 寄存器，设置合适的波特率。
2. 通过将 **SYNC**、**SPEN** 和 **CSRC** 位置 1，使能同步主串口。
3. 确保将 **CREN** 和 **SREN** 位清零。
4. 如果需要中断，将中断允许位 **RC2IE** 置 1。
5. 若需要接收 9 位数据，将 **RX9** 位置 1。
6. 若需要单字节接收，将 **SREN** 位置 1；若需要连续接收，将 **CREN** 位置 1。
7. 当接收完成时中断标志位 **RC2IF** 将置 1，此时如果允许位 **RC2IE** 也置 1，还将产生中断。
8. 读 **RCSTA2** 寄存器以获取第 9 位数据（如果已使能），并判断是否在接收过程中发生了错误。
9. 通过读 **RCREG2** 寄存器来读取接收到的 8 位数据。
10. 如果发生错误，通过将 **CREN** 位清零来清除错误。
11. 若想使用中断，请确保将 **INTCON** 寄存器中的 **GIE** 和 **PEIE** 位 (**INTCON<7:6>**) 置 1。

图 19-8： 同步主接收模式（由 **SREN** 位控制）

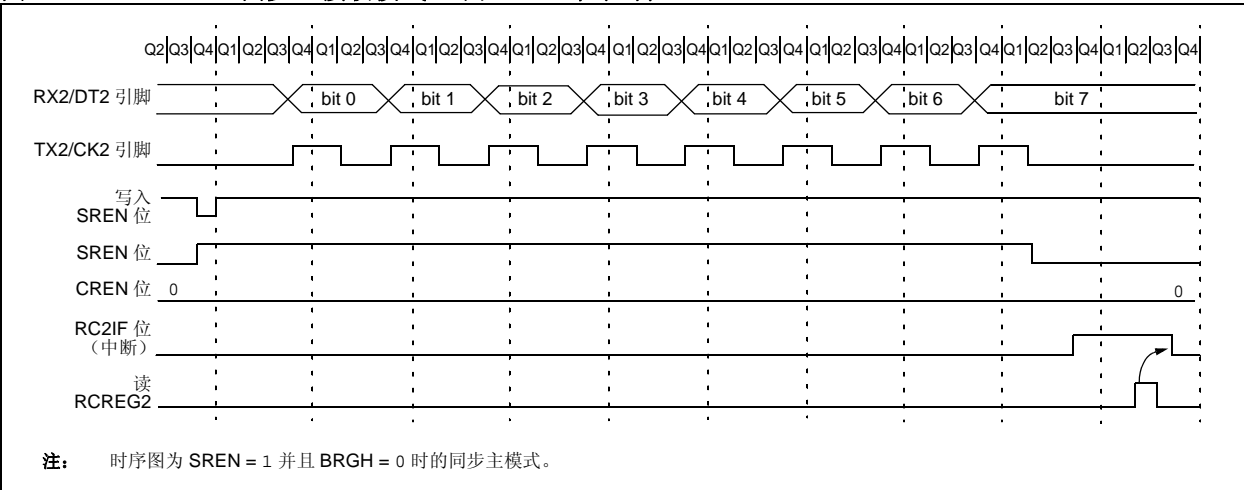


表 19-7： 与同步主接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR3	—	—	RC2IF	TX2IF	—	CCP2IF	CCP1IF	—	59
PIE3	—	—	RC2IE	TX2IE	—	CCP2IE	CCP1IE	—	59
IPR3	—	—	RC2IP	TX2IP	—	CCP2IP	CCP1IP	—	59
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	61
RCREG2	AUSART 接收寄存器								61
TXSTA2	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	61
SPBRG2	AUSART 波特率发生器寄存器								61

图注： — = 未实现，读为 0。同步主接收不使用阴影单元。



## 19.5 AUSART 同步从模式

将 CSRC (TXSTA2<7>) 位清零可进入同步从模式。此模式与同步主模式的区别在于移位时钟由CK2引脚上的外部时钟提供 (而主模式下由内部时钟提供)。这使得器件能在任何低功耗模式下发送或接收数据。

### 19.5.1 AUSART 同步从发送

除了休眠模式以外, 同步主模式和从模式的工作方式是完全相同的。

如果向 TXREG2 中写入两个字, 然后执行 SLEEP 指令, 则将发生以下事件:

- 第一个字立即传送到 TSR 寄存器进行发送。
- 第二个字仍保留在 TXREG2 寄存器中。
- 不会将标志位 TX2IF 置 1。
- 当第一个字移出 TSR 后, TXREG2 寄存器将把第二个字送入 TSR, 同时将标志位 TX2IF 置 1。
- 如果中断允许位 TX2IE 已置 1, 中断将把器件从休眠状态唤醒。如果允许了全局中断, 程序则会跳转到中断向量处执行。

设置同步从发送操作的步骤如下:

1. 通过将 SYNC 和 SPEN 位置 1 并将 CSRC 位清零使能同步从串口。
2. 将 CREN 和 SREN 位清零。
3. 如果需要中断, 将中断允许位 TX2IE 置 1。
4. 若需要发送 9 位数据, 将 TX9 位置 1。
5. 将使能位 TXEN 置 1, 使能发送。
6. 如果选择发送 9 位数据, 将第 9 位数据装入 TX9D 位。
7. 将数据装入 TXREG2 寄存器, 启动发送。
8. 若想使用中断, 请确保将 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 置 1。

表 19-8: 与同步从发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR3	—	—	RC2IF	TX2IF	—	CCP2IF	CCP1IF	—	59
PIE3	—	—	RC2IE	TX2IE	—	CCP2IE	CCP1IE	—	59
IPR3	—	—	RC2IP	TX2IP	—	CCP2IP	CCP1IP	—	59
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	61
TXREG2	AUSART 发送寄存器								61
TXSTA2	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	61
SPBRG2	AUSART 波特率发生器寄存器								61
LATG	U2OD	U1OD	—	LATG4	LATG3	LATG2	LATG1	LATG0	60

图注: — = 未实现, 读为 0。同步从发送不使用阴影单元。

# PIC18F85J11 系列

## 19.5.2 AUSART 同步从接收

除了休眠模式、任何空闲模式以及在从模式下为无关位的 SREN 位以外，同步主模式和同步从模式的工作方式完全相同。

如果通过在进入休眠或空闲模式前将 CREN 位置 1 来使能接收，那么在低功耗模式下可以接收到一个数据字。接收到该字后，RSR 寄存器将把数据传输到 RCREG2 寄存器。如果中断允许位 RC2IE 已置 1，则产生的中断将把器件从低功耗模式唤醒。如果允许了全局中断，程序则会跳转到中断向量处执行。

设置同步从接收操作的步骤如下：

1. 通过将 SYNC 和 SPEN 位置 1 并将 CSRC 位清零使能同步主串口。
2. 如果需要中断，将中断允许位 RC2IE 置 1。
3. 若需要接收 9 位数据，将 RX9 位置 1。
4. 将使能位 CREN 置 1，使能接收。
5. 当接收完成时，RC2IF 标志位将被置 1。如果中断允许位 RC2IE 已置 1，还将产生中断。
6. 读 RCSTA2 寄存器以获取第 9 位数据（如果已使能），并判断是否在接收过程中发生了错误。
7. 通过读 RCREG2 寄存器来读取接收到的 8 位数据。
8. 如果发生错误，通过将 CREN 位清零来清除错误。
9. 若想使用中断，请确保将 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）置 1。

表 19-9: 与同步从接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR3	—	—	RC2IF	TX2IF	—	CCP2IF	CCP1IF	—	59
PIE3	—	—	RC2IE	TX2IE	—	CCP2IE	CCP1IE	—	59
IPR3	—	—	RC2IP	TX2IP	—	CCP2IP	CCP1IP	—	59
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	61
RCREG2	AUSART 接收寄存器								61
TXSTA2	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	61
SPBRG2	AUSART 波特率发生器寄存器								61

图注： — = 未实现，读为 0。同步从接收不使用阴影单元。

## 20.0 10 位模数转换器 (A/D) 模块

所有 PIC18F85J11 系列器件的模数转换器 (A/D) 模块都有 12 路输入。此模块能将一个模拟输入信号转换成相应的 10 位数字信号。

此模块有 5 个寄存器：

- A/D 结果寄存器高字节 (ADRESH)
- A/D 结果寄存器低字节 (ADRESL)
- A/D 控制寄存器 0 (ADCON0)
- A/D 控制寄存器 1 (ADCON1)
- A/D 控制寄存器 2 (ADCON2)

ADCON0 寄存器 (如寄存器 20-1 所示) 控制 A/D 模块的工作。ADCON1 寄存器 (如寄存器 20-2 所示) 配置端口引脚的功能。ADCON2 寄存器 (如寄存器 20-3 所示) 配置 A/D 时钟源、可编程采集时间和输出结果的对齐方式。

**寄存器 20-1: ADCON0: A/D 控制寄存器 0**

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCAL	—	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            **ADCAL:** A/D 校准位  
 1 = 在下一次 A/D 转换时进行校准  
 0 = 正常的 A/D 转换器操作 (没有进行转换)
- bit 6            **未实现:** 读为 0
- bit 5-2        **CHS<3:0>:** 模拟通道选择位  
 0000 = 通道 00 (AN0)  
 0001 = 通道 01 (AN1)  
 0010 = 通道 02 (AN2)  
 0011 = 通道 03 (AN3)  
 0100 = 通道 04 (AN4)  
 0101 = 通道 05 (AN5)  
 0110 = 通道 06 (AN6)  
 0111 = 通道 07 (AN7)  
 1000 = 通道 08 (AN8)  
 1001 = 通道 09 (AN9)  
 1010 = 通道 10 (AN10)  
 1011 = 通道 11 (AN11)  
 11xx = 未用
- bit 1            **GO/DONE:** A/D 转换状态位  
 当 **ADON = 1** 时:  
 1 = A/D 转换正在进行  
 0 = A/D 空闲
- bit 0            **ADON:** A/D 使能位  
 1 = 使能 A/D 转换器模块  
 0 = 禁止 A/D 转换器模块

---

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
bit 7		bit 0					

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 5      **VCFG1**: 参考电压配置位 (VREF- 参考电压源)  
             1 = VREF- (AN2)  
             0 = AVSS

bit 4      **VCFG0**: 参考电压配置位 (VREF+ 参考电压源)  
             1 = VREF+ (AN3)  
             0 = AVDD

[illegible]

D = 数字 I/O

## 寄存器 20-3: ADCON2: A/D 控制寄存器 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

### 图注:

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

bit 7              **ADFM:** A/D 结果格式选择位

1 = 右对齐

0 = 左对齐

bit 6              **未实现:** 读为 0

bit 5-3              **ACQT<2:0>:** A/D 采集时间选择位

111 = 20 个 TAD

110 = 16 个 TAD

101 = 12 个 TAD

100 = 8 个 TAD

011 = 6 个 TAD

010 = 4 个 TAD

001 = 2 个 TAD

000 = 0 个 TAD<sup>(1)</sup>

bit 2-0              **ADCS<2:0>:** A/D 转换时钟选择位

111 = FRC (时钟来自 A/D 模块的 RC 振荡器) <sup>(1)</sup>

110 = FOSC/64

101 = FOSC/16

100 = FOSC/4

011 = FRC (时钟来自 A/D 模块的 RC 振荡器) <sup>(1)</sup>

010 = FOSC/32

001 = FOSC/8

000 = FOSC/2

**注 1:** 如果选择了 A/D 模块的 FRC 时钟源, 在 A/D 时钟启动之前会加上一个 T<sub>CY</sub> (指令周期) 的延迟。这可以保证在开始转换之前执行 SLEEP 指令。

# PIC18F85J11 系列

可通过软件选择器件的正电源电压和负电源电压 (AVDD 和 AVSS) 或 RA3/AN3/VREF+ 和 RA2/AN2/VREF- 引脚上的电压作为模拟参考电压。

A/D 转换器具有可在器件休眠模式下工作的独特特性。要使 A/D 转换器在休眠模式下工作，A/D 转换时钟必须来自 A/D 模块内部的 RC 振荡器。

采样保持电路的输出是转换器的输入，此转换器采用逐次逼近法得到转换结果。

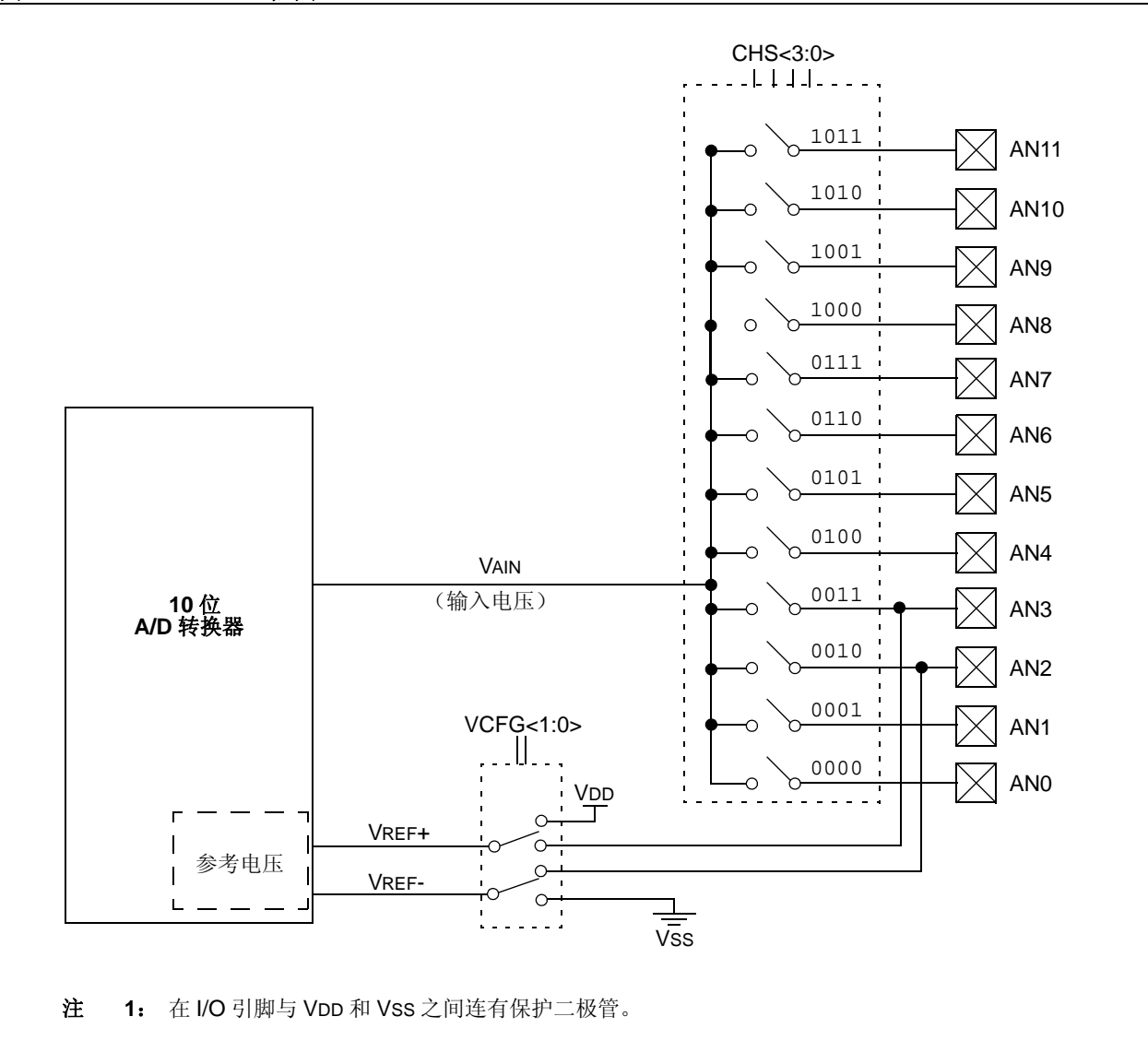
与 A/D 转换器相关的每个端口引脚都可以被配置为模拟输入或数字 I/O。ADRESH 和 ADRESL 寄存器保存

A/D 转换的结果。当 A/D 转换完成之后，转换结果被装入 ADRESH:ADRESL 寄存器对，GO/DONE 位 (ADCON0<1>) 被清零，且 A/D 中断标志位 ADIF 置 1。

器件复位操作强制所有寄存器进入复位状态。这将迫使 A/D 模块关闭，并中止所有正在进行的转换。上电复位时，ADRESH:ADRESL 寄存器对中的值保持不变。上电复位后，这两个寄存器中的数据不确定。

A/D 模块的框图见图 20-1。

图 20-1: A/D 框图 (1)

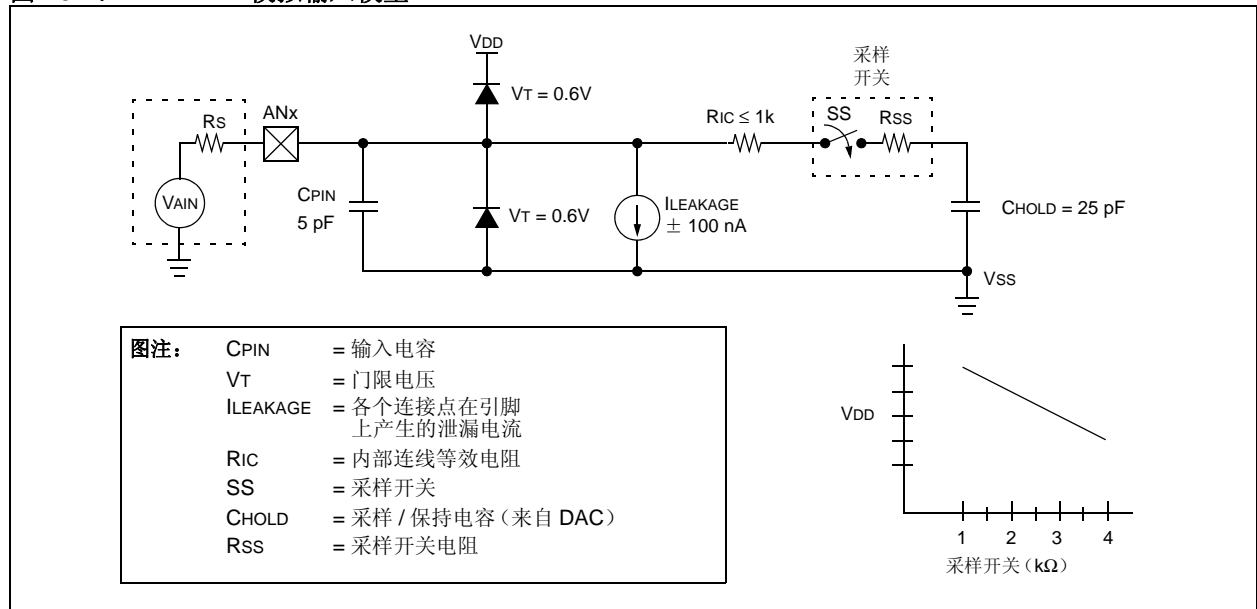


在根据需要配置好 A/D 模块后，在开始转换之前必须先对所选择的通道进行采集。必须将模拟输入通道相应的 TRIS 位设置为输入。采集时间的确定请参见第 20.1 节“A/D 采集要求”。在采集完成之后，即可启动 A/D 转换。采集时间可以被编程为位于 GO/DONE 位置 1 和实际启动转换之间。

执行 A/D 转换时应该遵循以下步骤：

1. 配置 A/D 模块：
  - 配置模拟引脚、参考电压和数字 I/O（通过 ADCON1）
  - 选择 A/D 输入通道（通过 ADCON0）
  - 选择 A/D 采集时间（通过 ADCON2）
  - 选择 A/D 转换时钟（通过 ADCON2）
  - 使能 A/D 模块（通过 ADCON0）
2. 配置 A/D 中断（如果需要）：
  - 清零 ADIF 位
  - 将 ADIE 位置 1
  - 将 GIE 位置 1
3. 等待所需的采集时间（如果需要）。
4. 启动转换：
  - 将 GO/DONE 位（ADCON0<1>）置 1
5. 等待 A/D 转换完成，通过以下两种方法之一判断转换是否完成：
  - 查询 GO/DONE 位是否被清零
 或
  - 等待 A/D 中断
6. 读取 A/D 结果寄存器（ADRESH:ADRESL），需要时将 ADIF 位清零。
7. 如需再次进行 A/D 转换，请根据需从步骤 1 或步骤 2 重新开始执行。每位的 A/D 转换时间被定义为  $T_{AD}$ 。在下次采集开始之前至少需要等待 2 个  $T_{AD}$ 。

图 20-2: 模拟输入模型



# PIC18F85J11 系列

## 20.1 A/D 采集要求

为了使 A/D 转换器达到规定精度，必须使充电保持电容（CHOLD）充满至输入通道的电平。模拟输入的电路模型如图 20-2 所示。模拟信号源的阻抗（Rs）和内部采样开关阻抗（Rss）直接影响给电容 CHOLD 充电所需的时间。采样开关阻抗（Rss）的值随器件电压（VDD）不同而变化。信号源阻抗影响模拟输入的失调电压（由于引脚泄漏电流引起）。**模拟信号源的最大阻抗推荐值为 2.5 kΩ。**选择（改变）模拟输入通道后，必须对通道进行采样才能启动转换，采样时间必须大于最小采集时间。

**注：** 当开始转换时，要將保持电容与输入引脚断开。

可使用公式 20-1 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 的误差是 A/D 模块达到规定分辨率所能允许的最大误差。

公式 20-3 为计算所需的最小采集时间 TACQ 的过程。计算结果基于以下应用系统假设：

CHOLD	=	25 pF
Rs	=	2.5 kΩ
转换误差	≤	1/2 LSb
VDD	=	3V → Rss = 2 kΩ
温度	=	85°C（系统最大值）

### 公式 20-1: A/D 采集时间

$$\begin{aligned} \text{TACQ} &= \text{放大器稳定时间} + \text{保持电容充电时间} + \text{温度系数} \\ &= \text{TAMP} + \text{TC} + \text{Tcoff} \end{aligned}$$

### 公式 20-2: A/D 最小充电时间

$$\begin{aligned} \text{V}_{\text{HOLD}} &= (\text{V}_{\text{REF}} - (\text{V}_{\text{REF}}/2048)) \cdot (1 - e^{-(\text{TC}/\text{CHOLD}(\text{RIC} + \text{RSS} + \text{RS})))} \\ \text{或} \\ \text{TC} &= -(\text{CHOLD})(\text{RIC} + \text{RSS} + \text{RS}) \ln(1/2048) \end{aligned}$$

### 公式 20-3: 计算所需的最小采集时间

$$\begin{aligned} \text{TACQ} &= \text{TAMP} + \text{TC} + \text{Tcoff} \\ \text{TAMP} &= 0.2 \mu\text{s} \\ \text{Tcoff} &= (\text{Temp} - 25^\circ\text{C})(0.02 \mu\text{s}/^\circ\text{C}) \\ &\quad (85^\circ\text{C} - 25^\circ\text{C})(0.02 \mu\text{s}/^\circ\text{C}) \\ &\quad 1.2 \mu\text{s} \end{aligned}$$

只有在温度 > 25°C 时才需要温度系数。当温度低于 25°C 时，Tcoff = 0 ms。

$$\begin{aligned} \text{TC} &= -(\text{CHOLD})(\text{RIC} + \text{RSS} + \text{RS}) \ln(1/2048) \mu\text{s} \\ &\quad -(25 \text{ pF})(1 \text{ k}\Omega + 2 \text{ k}\Omega + 2.5 \text{ k}\Omega) \ln(0.0004883) \mu\text{s} \\ &\quad 1.05 \mu\text{s} \end{aligned}$$

$$\begin{aligned} \text{TACQ} &= 0.2 \mu\text{s} + 1 \mu\text{s} + 1.2 \mu\text{s} \\ &\quad 2.4 \mu\text{s} \end{aligned}$$



20.2 选择和配置自动采集时间

用户可以利用 `ADCON2` 寄存器选择采集时间，该采集时间发生在每次 `GO/DONE` 位置 1 之后。

当 `GO/DONE` 位置 1 时，采样停止并启动转换。用户应确保在选定所需要的输入通道之后到 `GO/DONE` 位置 1 之前经过了所需要的采集时间。这在 `ACQT<2:0>` 位 (`ADCON2<5:3>`) 保持在复位状态 (000) 时发生，并且与不提供可编程采集时间的器件兼容。

如果需要，可设置 `ACQT` 位来选择 A/D 模块的可编程采集时间。当 `GO/DONE` 位置 1 时，A/D 模块继续对输入进行采样，采样时间为所选择的采集时间，然后自动启动转换。由于采集时间已被编程，因此在选择通道和 `GO/DONE` 位置 1 之间无需另外等待一个采集时间。

在这两种情况下，当转换完成时，`GO/DONE` 位被清零、`ADIF` 标志位被置 1 且 A/D 再次开始采样当前选择的通道。如果采集时间被编程，那么将不会有任何标志显示采集时间是否结束或转换是否开始。

20.3 选择 A/D 转换时钟

每位的 A/D 转换时间被定义为 `TAD`。每完成一次 10 位 A/D 转换需要 11 个 `TAD`。可用软件选择 A/D 转换时钟源。

`TAD` 有以下 7 种可能的选择：

- 2 个 `Tosc`
- 4 个 `Tosc`
- 8 个 `Tosc`
- 16 个 `Tosc`
- 32 个 `Tosc`
- 64 个 `Tosc`
- 内部 RC 振荡器

为了实现正确的 A/D 转换，A/D 转换时钟 (`TAD`) 必须尽可能小，但它必须大于最小 `TAD`（欲知更多的信息，请参见表 26-27 中的参数 130）。

表 20-1 所示为器件在不同的工作频率下和选择不同的 A/D 时钟源时得到的 `TAD`。

表 20-1: 不同器件工作频率下的 `TAD`

AD 时钟源 ( <code>TAD</code> )		最高器件频率
工作状态	<code>ADCS&lt;2:0&gt;</code>	
2 个 <code>Tosc</code>	000	2.86 MHz
4 个 <code>Tosc</code>	100	5.71 MHz
8 个 <code>Tosc</code>	001	11.43 MHz
16 个 <code>Tosc</code>	101	22.86 MHz
32 个 <code>Tosc</code>	010	40.0 MHz
64 个 <code>Tosc</code>	110	40.0 MHz
RC <sup>(1)</sup>	x11	1.00 MHz <sup>(2)</sup>

- 注 1: RC 时钟源的典型 `TAD` 时间为 4  $\mu$ s。
- 注 2: 器件工作频率高于 1 MHz 时，整个转换过程必须在休眠模式下进行，否则 A/D 转换精度可能超出规范允许范围。

20.4 配置模拟端口引脚

`ADCON1`、`TRISA`、`TRISF` 和 `TRISH` 寄存器均可控制 A/D 端口引脚的操作。若希望端口引脚为模拟输入，则必须将相应的 `TRIS` 位置 1（输入）。如果将 `TRIS` 位清零（输出），则转换数字输出电平 (`VOH` 或 `VOL`)。

A/D 转换操作与 `CHS<3:0>` 位以及 `TRIS` 位的状态无关。

注 1:	当读取 <code>PORT</code> 寄存器时，所有配置为模拟输入通道的引脚均读为 0（低电平）。配置为数字输入的引脚将转换模拟输入信号。配置为数字输入的引脚上的模拟电平会被精确转换。
注 1:	定义为数字输入的引脚上的模拟电平可能会导致数字输入缓冲器消耗的电流超出器件规范的限制。

# PIC18F85J11 系列

## 20.5 A/D 转换

图 20-3 所示为  $\overline{\text{GO/DONE}}$  位置 1 且  $\text{ACQT}\langle 2:0 \rangle$  位被清零后 A/D 转换器的工作状态。转换在下一条指令执行之后开始，以允许器件在转换开始之前进入休眠模式。

图 20-4 所示为  $\overline{\text{GO/DONE}}$  位置 1、 $\text{ACQT}\langle 2:0 \rangle$  位被设置为 010、且在转换开始之前选择了 4 TAD 的采集时间之后 A/D 转换器的工作状态。

在转换期间将  $\overline{\text{GO/DONE}}$  位清零将中止当前转换。不会用尚未完成的 A/D 转换结果来更新 A/D 结果寄存器对。这意味着  $\text{ADRESH:ADRESL}$  寄存器仍将保持上一次的转换结果（即上一次写入  $\text{ADRESH:ADRESL}$  寄存器的值）。

在 A/D 转换完成或中止以后，需要等待 2 个 TAD 才能开始下一次采集。等待时间一到，将自动开始对所选通道进行采集。

**注：** 不应使用启动 A/D 模块的指令将  $\overline{\text{GO/DONE}}$  位置 1。

## 20.6 CCP2 触发信号的使用

CCP2 模块的“特殊事件触发信号”可以启动 A/D 转换。这需要将  $\text{CCP2M}\langle 3:0 \rangle$  位（ $\text{CCP2CON}\langle 3:0 \rangle$ ）设置为 1011，并使能 A/D 模块（ $\overline{\text{ADON}}$  位置 1）。发生触发事件时， $\overline{\text{GO/DONE}}$  位被置 1，启动 A/D 采集和转换，并将 Timer1（或 Timer3）计数器复位为 0。复位 Timer1（或 Timer3）可自动重复 A/D 采集周期，最大限度地降低了软件开销（将  $\text{ADRESH/ADRESL}$  内容移到目标单元）。在特殊事件触发信号将  $\overline{\text{GO/DONE}}$  位置 1（启动转换）前，用户必须选择正确的模拟输入通道，并通过手动或选择相应的  $\text{TACQ}$  时间来设定最小采集时间。

如果 A/D 模块未使能（ $\overline{\text{ADON}}$  清零），则特殊事件触发信号对 A/D 模块不产生任何影响，但它仍会将 Timer1（或 Timer3）计数器复位。

图 20-3: A/D 转换 TAD 周期（ $\text{ACQT}\langle 2:0 \rangle = 000$ ， $\text{TACQ} = 0$ ）

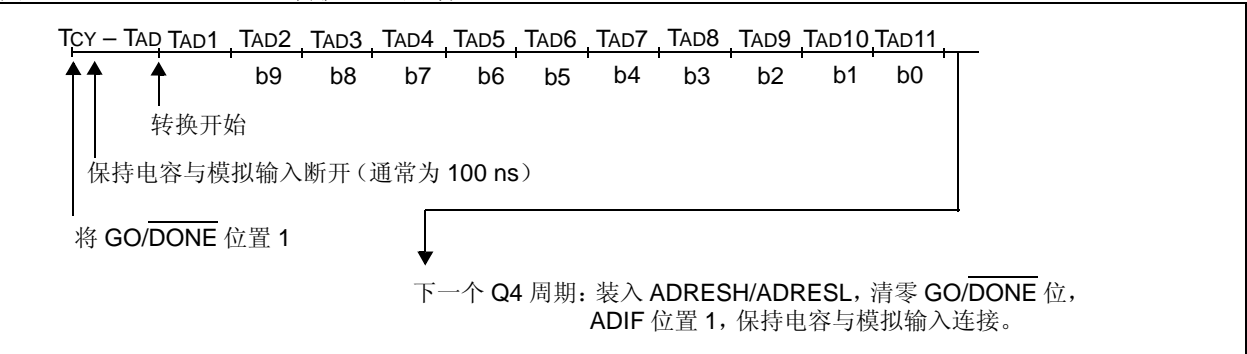
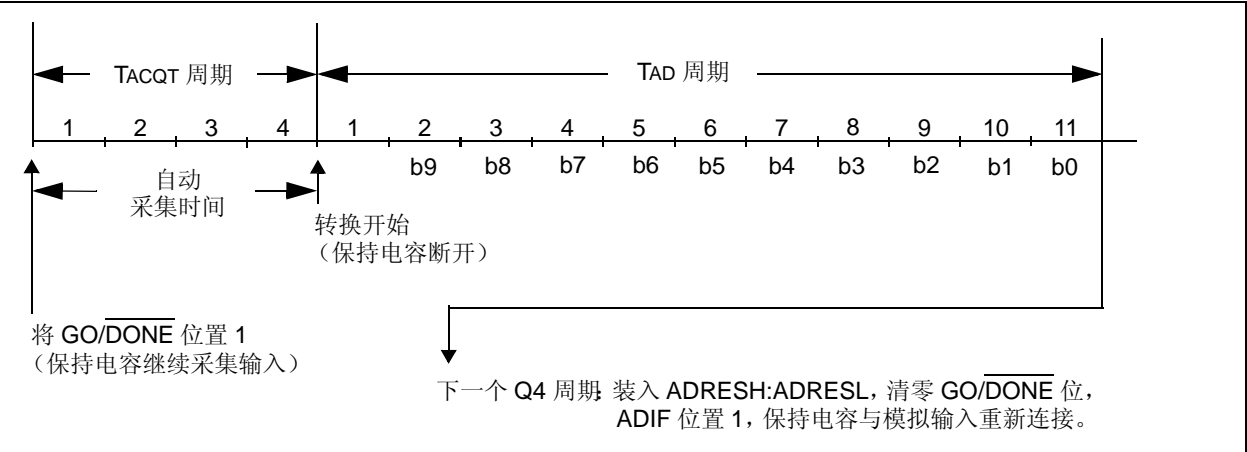


图 20-4: A/D 装换 TAD 周期（ $\text{ACQT}\langle 2:0 \rangle = 010$ ， $\text{TACQ} = 4 \text{ TAD}$ ）



## 20.7 A/D 转换器校准

PIC18F85J11 系列器件中的 A/D 转换器具有自动校准功能，它可以补偿模块中所产生的任何偏移。校准过程是自动进行的，并可以通过置 1 ADCAL 位 (ADCON0<7>) 来启动校准。下一次 GO/DONE 位置 1 时，此模块将执行“假”转换（即不读输入通道），并内部存储结果值以补偿偏移。由此，后续的偏移都将得到补偿。

在校准过程中假设器件处于相对稳定的工作状态。如果使用 A/D 校准，则应在每次器件复位之后或在运行条件有重要变化时执行校准操作。

## 20.8 在功耗管理模式下工作

处于功耗管理模式时，自动采集时间和 A/D 转换时钟的选择一定程度上可由时钟源和频率决定。

如果希望器件处于功耗管理模式时进行 A/D 转换，ADCON2 中的 ACQT<2:0> 和 ADCS<2:0> 位就应该根据该模式下使用的时钟进行更新。在进入功耗管理模式之后（某种功耗管理运行模式），就可以开始 A/D 采集或转换。采集或转换开始以后，器件应继续使用功耗管理模式中的时钟源直到转换完成。如果需要，在转换期间也可以将器件置于相应的功耗管理空闲模式。

如果功耗管理模式时钟频率小于 1 MHz，就应该选择 A/D RC 时钟源。

在休眠模式下工作需要选择 A/D 模块的 RC 时钟。如果 ACQT<2:0> 位被设置为 000 并启动转换，转换将延迟一个指令周期以允许执行 SLEEP 指令并进入休眠模式。OSCCON 寄存器中的 IDLEN 和 SCS<1:0> 位必须在转换开始之前被清零。

表 20-2: A/D 寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值 所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	—	TMR2IF	TMR1IF	59
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	—	TMR2IE	TMR1IE	59
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	—	TMR2IP	TMR1IP	59
PIR3	—	—	RC2IF	TX2IF	—	CCP2IF	CCP1IF	—	59
PIE3	—	—	RC2IE	TX2IE	—	CCP2IE	CCP1IE	—	59
IPR3	—	—	RC2IP	TX2IP	—	CCP2IP	CCP1IP	—	59
ADRESH	A/D 结果寄存器的高字节								59
ADRESL	A/D 结果寄存器的低字节								59
ADCON0	ADCAL	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	59
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	59
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	59
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	60
PORTA	RA7 <sup>(1)</sup>	RA6 <sup>(1)</sup>	RA5	RA4	RA3	RA2	RA1	RA0	60
TRISA	TRISA7 <sup>(1)</sup>	TRISA6 <sup>(1)</sup>	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	60
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	60
TRISF	TRISF5	TRISF4	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	60

图注： — = 未实现，读为 0。A/D 转换不使用阴影单元。

注 1： 仅当选择内部振荡器作为默认时钟源（FOSC2 配置位 = 0）时，才将 RA6/RA7 及其相关的锁存和方向位配置为端口引脚，否则将禁止它们并且这些位读为 0。

# PIC18F85J11 系列

---

注:

## 21.0 比较器模块

模拟比较器模块包含两个比较器，可以用多种方式对它们进行配置。该比较器的输入可以是与引脚 RF1 至 RF6 复用的模拟输入，也可以是片上参考电压（见第 22.0 节“比较器参考电压源模块”）。数字输出（正常或反相输出）可从引脚电平获取，也可通过控制寄存器读取。

CMCON 寄存器（寄存器 21-1）选择比较器的输入和输出配置。各比较器配置的框图如图 21-1 所示。

寄存器 21-1: CMCON: 比较器模块控制寄存器

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1
C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
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	<b>C2OUT:</b> 比较器 2 输出位 当 C2INV = 0 时: 1 = C2 VIN+ > C2 VIN- 0 = C2 VIN+ < C2 VIN- 当 C2INV = 1 时: 1 = C2 VIN+ < C2 VIN- 0 = C2 VIN+ > C2 VIN-
bit 6	<b>C1OUT:</b> 比较器 1 输出位 当 C1INV = 0 时: 1 = C1 VIN+ > C1 VIN- 0 = C1 VIN+ < C1 VIN- 当 C1INV = 1 时: 1 = C1 VIN+ < C1 VIN- 0 = C1 VIN+ > C1 VIN-
bit 5	<b>C2INV:</b> 比较器 2 输出反相位 1 = C2 输出反相 0 = C2 输出同相
bit 4	<b>C1INV:</b> 比较器 1 输出反相位 1 = C1 输出反相 0 = C1 输出同相
bit 3	<b>CIS:</b> 比较器输入切换位 当 CM<2:0> = 110 时: 1 = C1 VIN- 连接到 RF5/AN10/CVREF, C2 VIN- 连接到 RF3/AN8 0 = C1 VIN- 连接到 RF6/AN11, C2 VIN- 连接到 RF4/AN9
bit 2-0	<b>CM&lt;2:0&gt;:</b> 比较器模式位 图 21-1 给出了比较器模式和 CM<2:0> 位设置。

# PIC18F85J11 系列

## 21.1 比较器配置

图 21-1 为比较器的八种操作模式。使用 CMCON 寄存器的 CM<2:0> 位来选择这些模式。TRISF 寄存器控制每个模式下的比较器引脚的数据方向。比较器模式改变

时，比较器输出电平可能会对第 26.0 节“电气特性”中所列的特定模式改变延时无效。

**注：** 应在比较器模式改变期间禁止比较器中断，否则，可能发生错误中断。

图 21-1: 比较器 I/O 工作模式

<p><b>禁止比较器输出</b> CM&lt;2:0&gt; = 000</p>	<p><b>比较器关闭 (POR 默认值)</b> CM&lt;2:0&gt; = 111</p>
<p><b>两个独立的比较器</b> CM&lt;2:0&gt; = 010</p>	<p><b>带输出的两个独立比较器</b> CM&lt;2:0&gt; = 011</p>
<p><b>具有公共参考端的双比较器</b> CM&lt;2:0&gt; = 100</p>	<p><b>带输出的两个具有公共参考端的比较器</b> CM&lt;2:0&gt; = 101</p>
<p><b>一个带输出的独立比较器</b> CM&lt;2:0&gt; = 001</p>	<p><b>复用四路输入的双比较器</b> CM&lt;2:0&gt; = 110</p>
<p>A = 模拟输入，端口始终读为 0    D = 数字输入    CIS (CMCON&lt;3&gt;) 为比较器输入切换位 *TRISF&lt;2:1&gt; 位置 1 将通过配置引脚为输入从而禁止比较器输出。</p>	

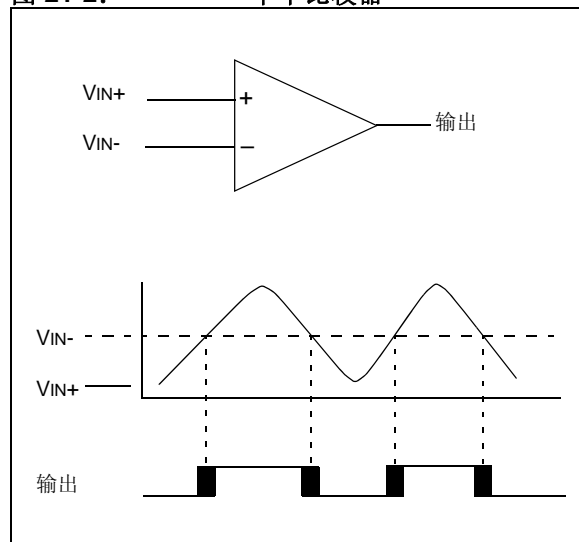
## 21.2 比较器工作原理

图 21-2 显示了单个比较器，以及其模拟输入电平和数字输出之间的关系。如果  $V_{IN+}$  上的模拟输入电平值小于  $V_{IN-}$  上的模拟输入值，那么比较器将输出数字低电平。如果  $V_{IN+}$  上的模拟输入电平值高于  $V_{IN-}$  上的模拟输入值，那么比较器将输出数字高电平。图 21-2 中比较器输出的阴影部分表示因输入失调和响应时间所造成的输出不确定区域。

## 21.3 比较器参考电压

根据不同的比较器工作模式可使用外部或内部参考电压。比较  $V_{IN-}$  上的模拟信号与  $V_{IN+}$  上的信号，对比较器的数字输出作相应的调整（图 21-2）。

图 21-2: 单个比较器



### 21.3.1 外部参考信号

使用外部参考电压时，比较器模块可配置为使两个比较器使用相同或不同的参考源。然而，阈值探测器应用可能要求使用相同的参考电压。参考信号必须位于  $V_{SS}$  和  $V_{DD}$  之间，且可用于比较器的任一引脚。

### 21.3.2 内部参考信号

比较器模块也允许选择来自比较器参考电压模块的内部产生的参考电压。第 22.0 节“比较器参考电压源模块”中将对此模块进行更详细的说明。

内部参考电压仅在两个比较器复用四路输入（ $CM<2:0> = 110$ ）的模式下可用。在此模式下，内部参考电压连接到两个比较器的  $V_{IN+}$  引脚。

## 21.4 比较器响应时间

响应时间是指从选定一个新的参考电压或输入源到比较器输出达到一个有效电平的最短时间。如果内部参考电压发生改变，使用比较器输出时必须考虑内部参考电压的最大延时。否则，应使用比较器的最大响应延时（见第 26.0 节“电气特性”）。

## 21.5 比较器输出

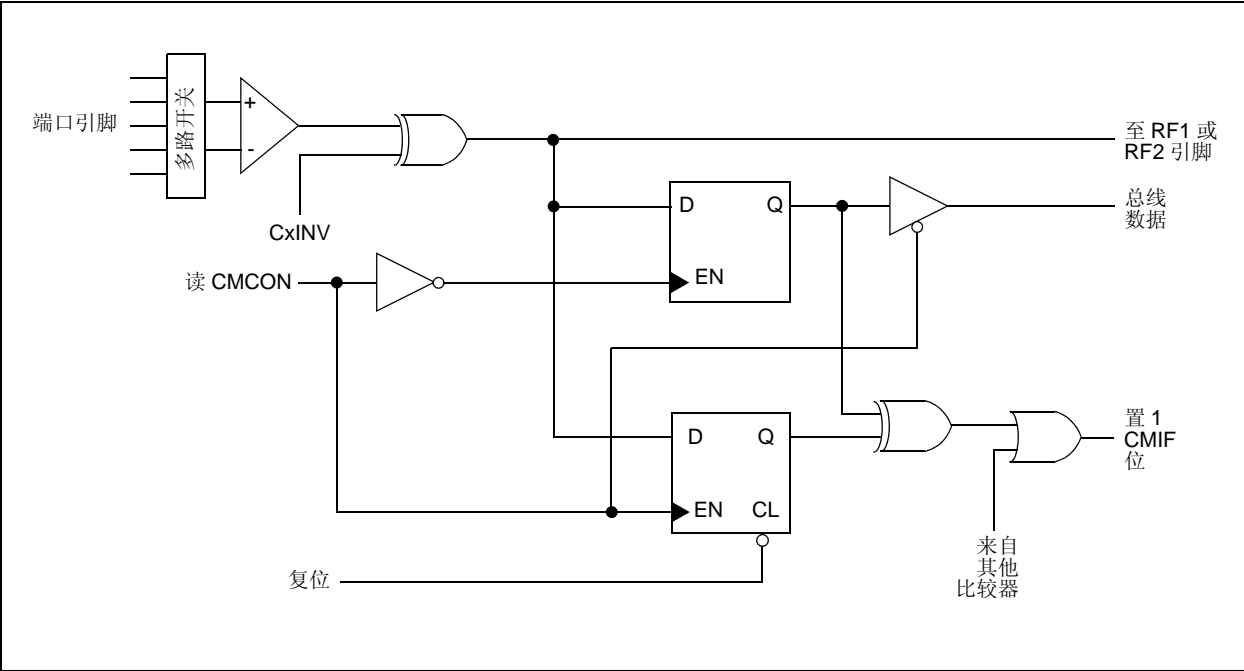
通过  $CMCON$  寄存器可读取比较器的输出。这些位为只读位。比较器输出也可直接输出到  $RF1$  和  $RF2$  I/O 引脚。使能时， $RF1$  和  $RF2$  引脚输出路径中的多路开关将进行切换且每个引脚的输出为比较器的非同步输出。每个比较器输出的不确定区域的大小与电气规范中给出的输入失调电压和响应时间有关。图 21-3 为比较器输出框图。

在该模式下， $TRISF$  位仍用作  $RF1$  和  $RF2$  引脚的输出使能 / 禁止位。

使用  $C2INV$  和  $C1INV$  位（ $CMCON<5:4>$ ）可翻转比较器输出的极性。

- |   |
|---|
| <p><b>注 1:</b> 当读取 <math>PORT</math> 寄存器时，所有配置为模拟输入的引脚均读为 0。配置为数字输入的引脚将根据施密特触发器输入规范转换模拟输入信号。</p> <p><b>2:</b> 定义为数字输入的引脚上的模拟电平可能会导致输入缓冲器消耗的电流超出器件规范的限制。</p> |
|---|

图 21-3: 比较器输出框图



## 21.6 比较器中断

任何一个比较器的输出电平一旦发生了变化，就会将比较器的中断标志位置 1。为确定实际发生的变化，需要用软件来保存输出位的状态信息（从 CMCON<7:6> 中读取数据）。CMIF 位（PIR2<6>）为比较器中断标志位，必须通过清零 CMIF 位来将其复位。因为可以人为向 PIR2 寄存器写入 1，所以也可以模拟中断产生。

必须将 CMIE 位（PIE2<6>）和 PEIE 位（INTCON<6>）置 1 以允许中断。此外，还必须将 GIE 位（INTCON<7>）置 1。只要这些位中的任何一位被清零，虽然当有中断条件产生时 CMIF 位仍会置 1，但中断是被禁止的。

**注：** 当正在执行读操作（Q2 周期的开始）时，若 CMCON 寄存器（C1OUT 或 C2OUT）发生变化，那么 CMIF（PIR2<6>）中断标志位可能无法被置 1。

用户可用以下方式在中断服务程序中清除该中断：

- 任一读或写 CMCON 的操作都将结束电平不匹配状态。
- 清零标志位 CMIF。

电平不匹配的状态将会持续地将 CMIF 标志位置 1。读 CMCON 将结束不匹配状态，并且允许将 CMIF 标志位清零。

## 21.7 休眠模式下的比较器工作原理

当比较器处于运行模式而器件处于休眠模式时，比较器仍保持工作并可使用比较器中断（如果中断被允许）。当允许中断时，中断会把器件从休眠模式唤醒。每个比较器工作时都会消耗额外的电流，如比较器规范中所列。若要把休眠模式下的功耗减少到最小，可在进入休眠模式前关闭比较器模块（CM<2:0> = 111）。当器件从休眠模式被唤醒时，CMCON 寄存器的内容不受影响。

## 21.8 复位的影响

器件复位强制 CMCON 寄存器进入复位状态，导致比较器模块处于关闭状态（CM<2:0> = 111）。然而，器件复位时，默认将输入引脚（RF3 至 RF6）配置为模拟输入。这些引脚的 I/O 配置由 PCFG<3:0> 位（ADCON1<3:0>）的设置决定。因此，若复位时为模拟输入，则器件的电流最小。



## 21.9 模拟输入连接注意事项

图 21-4 是一个简化的模拟输入电路。由于模拟引脚和数字输出端相连，因此在  $V_{DD}$  和  $V_{SS}$  之间连有反向偏置二极管。这样，即将模拟输入电压限制在  $V_{SS}$  和  $V_{DD}$  之间。一旦输入电压超出该范围 0.6V 以上，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推荐值为  $10\text{ k}\Omega$ 。任何连接到模拟输入引脚上的外部元件（如电容或齐纳二极管等），要保证其泄漏电流极小。

图 21-4： 比较器模拟输入模型

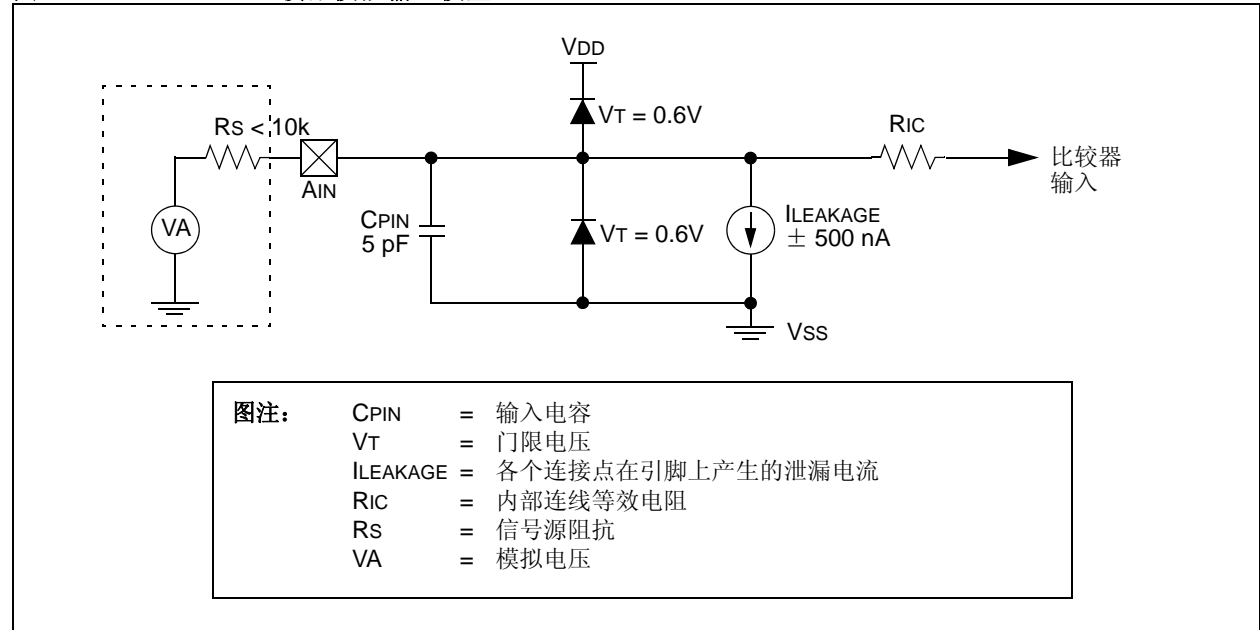


表 21-1： 与比较器模块相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	57
PIR2	OSCFIF	CMIF	—	—	BCLIF	LVDIF	TMR3IF	—	59
PIE2	OSCFIE	CMIE	—	—	BCLIE	LVDIE	TMR3IE	—	59
IPR2	OSCFIP	CMIP	—	—	BCLIP	LVDIP	TMR3IP	—	59
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	59
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	59
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	60
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	—	60
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	60

图注： — = 未实现，读为 0。比较器模块不使用阴影单元。

# PIC18F85J11 系列

---

注:

## 22.0 比较器参考电压源模块

比较器参考电压源模块是一个16阶的梯形电阻网络，可提供多个参考电压供选择。虽然它的主要目的是为模拟比较器提供参考电压，但也可将它用于其他场合。

图22-1显示了此模块的框图。梯形电阻经过分段可提供两种范围的CVREF值，并具有断电功能，可以在不使用参考电压时节省功耗。此模块的参考电源可由VDD/VSS或外部参考电压提供。

### 22.1 配置比较器参考电压

比较器参考电压模块由CVRCON寄存器（寄存器22-1）控制。比较器参考电压模块提供两种范围的输出电压，每种范围都具有16种电压。

CVRR位（CVRCON<5>）用于选择要使用的输出电压范围。这两种范围的主要区别在于其电压值之间的步长不同（其中一种范围可提供更高的分辨率），该步长由CVREF选择位（CVR<3:0>）来选择。计算比较器参考电压输出值的公式如下所示：

$$\text{当 CVRR} = 1: \\ \text{CVREF} = ((\text{CVR} < 3:0 >)/24) \times (\text{CVRSRC})$$

$$\text{当 CVRR} = 0: \\ \text{CVREF} = (\text{CVRSRC}/4) + ((\text{CVR} < 3:0 >)/32) \times (\text{CVRSRC})$$

比较器参考电压模块的电源可来自VDD和VSS，也可来自与RA2和RA3复用的外部VREF+和VREF-。电压源由CVRSS位（CVRCON<4>）选择。

更改CVREF输出值时，必须考虑比较器参考电压的稳定时间（见第26.0节“电气特性”中的表26-3）。

寄存器 22-1: CVRCON: 比较器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
CVREN	CVROE <sup>(1)</sup>	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
bit 7							bit 0

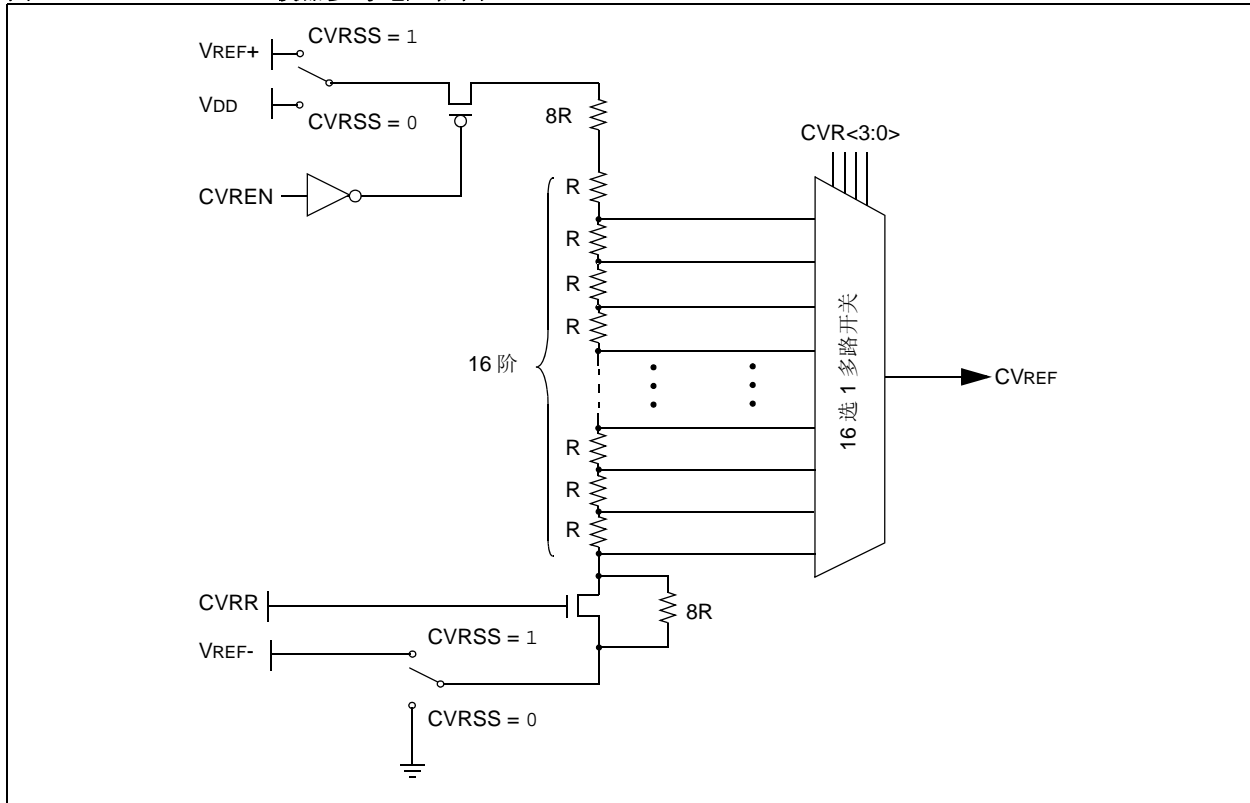
图注:

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      **CVREN:** 比较器参考电压使能位  
1 = CVREF 电路上电  
0 = CVREF 电路断电
- bit 6      **CVROE:** 比较器 VREF 输出使能位 <sup>(1)</sup>  
1 = CVREF 电压也是 RF5/AN10/CVREF 引脚上的输出  
0 = CVREF 电压与 RF5/AN10/CVREF 引脚断开
- bit 5      **CVRR:** 比较器 VREF 范围选择位  
1 = 0 到 0.667 CVRSRC，步长为 CVRSRC/24（低电平范围）  
0 = 0.25 CVRSRC 到 0.75 CVRSRC，步长为 CVRSRC/32（高电平范围）
- bit 4      **CVRSS:** 比较器 VREF 源选择位  
1 = 比较器参考电压源，CVRSRC = (VREF+) – (VREF-)  
0 = 比较器参考电压源，CVRSRC = VDD – VSS
- bit 3-0    **CVR<3:0>:** 比较器 VREF 值选择位（0 ≤ (CVR<3:0>) ≤ 15）  
当 CVRR = 1 时:  
 $\text{CVREF} = ((\text{CVR} < 3:0 >)/24) \cdot (\text{CVRSRC})$   
当 CVRR = 0 时:  
 $\text{CVREF} = (\text{CVRSRC}/4) + ((\text{CVR} < 3:0 >)/32) \cdot (\text{CVRSRC})$

注 1: CVROE 的优先级高于 TRISF<5> 位。

图 22-1: 比较器参考电压框图



## 22.2 比较器参考电压精度 / 误差

由于模块结构的限制，并不能实现整个参考电压范围的满量程输出。梯形电阻网络顶部和底部的晶体管（图 22-1）使  $CVREF$  值不能达到参考电压源的上下限。参考电压是由参考电压源分压而来的，因此  $CVREF$  输出随参考电压源的波动而变化。测试到的参考电压的绝对精度请参考第 26.0 节“电气特性”。

## 22.3 在休眠模式下工作

如果器件因中断或看门狗定时器超时从休眠模式唤醒， $CVRCON$  寄存器的内容将不受影响。为了降低休眠模式下的电流消耗，应禁止比较器参考电压模块。

## 22.4 复位的影响

器件复位通过将  $CVREN$ （ $CVRCON<7>$ ）位清零从而禁止比较器参考电压模块。此复位功能还通过将  $CVROE$ （ $CVRCON<6>$ ）清零，使参考电压与  $RA2$  引脚断开；同时通过将  $CVRR$ （ $CVRCON<5>$ ）位清零选择高电压范围。 $CVR<3:0>$  值选择位也将清零。

## 22.5 连接注意事项

比较器参考电压模块的工作独立于比较器模块。如果  $CVROE$  位置 1，则参考电压发生器的输出可能与  $RF5$  引脚连接。当  $RA2$  配置为数字输入时，将参考电压输出连接到  $RA2$  引脚上将增加电流消耗。使能  $CVRSS$  时，将  $RF5$  用作数字输出引脚也将增加电流消耗。

$RF5$  引脚可被直接用作 D/A 输出，但是其驱动能力有限。由于电流驱动能力有限，参考电压输出端必须外接一个缓冲器到  $VREF$ 。图 22-2 举例说明了这一缓冲技术。

图 22-2: 比较器参考电压输出缓冲器示例

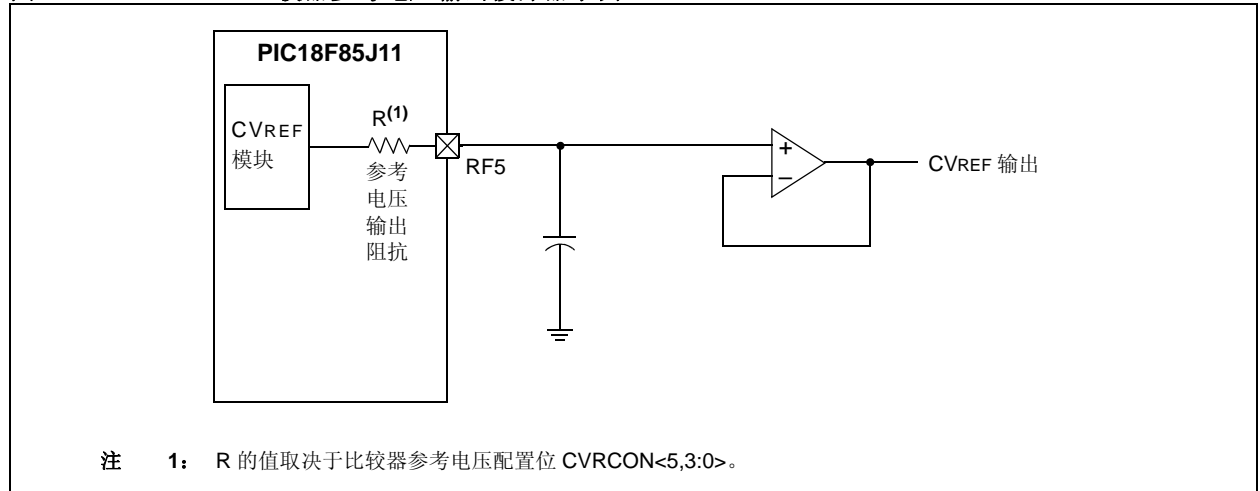


表 22-1: 与比较器参考电压相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在页
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	59
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	59
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	60

图注: — = 未实现, 读为 0。比较器参考电压不使用阴影单元。

# PIC18F85J11 系列

---

注:

## 23.0 CPU 的特殊功能

PIC18F85J11 系列器件具有几项特殊的功能，旨在最大限度地提高系统可靠性，并通过减少外部元件把成本降到最低。这些功能包括：

- 振荡器选择
- 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）
- 故障保护时钟监视器
- 双速启动
- 代码保护
- 在线串行编程

根据具体应用对频率、功耗、精度和成本的要求来配置振荡器。在**第 3.0 节“振荡器配置”**中详细讨论了所有的选项。

在本数据手册的前面几章中已经完整地讨论了器件的复位和中断。

除了为复位提供上电延时定时器和振荡器起振定时器之外，PIC18F85J11 系列器件还有一个可配置的看门狗定时器，该定时器由软件控制。

器件自带的 RC 振荡器还提供了故障保护时钟监视器（FSCM）和双速启动这两个额外的功能。FSCM 对外设时钟进行后台监视，并在外设时钟发生故障时自动切换时钟源。双速启动使得代码几乎可在起振发生时立即执行，而此时主时钟源刚好完成起振延时。

通过设置相应的配置寄存器位可以使能和配置所有这些功能。

### 23.1 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。这些配置位被映射到程序存储器从 300000h 开始的单元中。表 23-2 为配置位地址的完整列表。从寄存器 23-1 到寄存器 23-6 详细解释了各配置位的不同功能。

#### 23.1.1 配置 PIC18F85J11 系列器件时的注意事项

与先前的 PIC18 单片机不同，PIC18F85J11 系列器件不再使用非易失性存储寄存器存储配置信息。配置寄存器 CONFIG1L 到 CONFIG4H 以易失性存储方式实现。

就在上电或器件复位后，单片机硬件使用存储在非易失性闪存程序存储器中的配置数据自动装载 CONFIG1L 到 CONFIG4L 寄存器。闪存程序存储器的最后 4 个字，被称为闪存配置字（Flash Configuration Words, FCW），这些字用来存储配置数据。表 23-1 提供闪存程序存储器，其将被载入相应的配置寄存器。

当为这些器件创建应用程序时，用户就该为配置数据分配特定的闪存配置字单元。以确保当编译代码时程序代码不会存储在该地址上。

在上电复位时用于配置位的易失性存储单元始终复位为 1。对于所有其他类型的复位事件，将保留先前已编程的值，因此无需从程序存储器重新装入即可使用。

程序存储器中 CONFIG1H、CONFIG2H 和 CONFIG3H 的高 4 位也应为 1111。这样当这些配置字被远程事件意外执行时，被当作一条 NOP 指令。由于配置位不是在对应的单元内实现的，因此向这些单元写 1 不会影响器件工作。

为了避免代码执行期间的意外配置变化，所有的可编程配置位只可被写入一次。在上电周期内对位进行初始化编程之后，就不能再次写入该位了。要改变器件的配置需要对器件周期性地上电。

**表 23-1： 闪存配置字映射到配置寄存器**

配置字节	代码空间地址	配置寄存器地址
CONFIG1L	XXXF8h	300000h
CONFIG1H	XXXF9h	300001h
CONFIG2L	XXXFAh	300002h
CONFIG2H	XXXFBh	300003h
CONFIG3L	XXXFCh	300004h
CONFIG3H	XXXFDh	300005h

# PIC18F85J11 系列

表 23-2: 配置位和器件 ID

寄存器名		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	缺省 / 未编程值 <sup>(1)</sup>
300000h	CONFIG1L	DEBUG	XINST	STVREN	—	—	—	—	WDTEN	111- -1
300001h	CONFIG1H	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(3)</sup>	CP0	—	—	---- 01--
300002h	CONFIG2L	IESO	FCMEN	—	—	—	FOSC2	FOSC1	FOSC0	11-- -111
300003h	CONFIG2H	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	WDTPS3	WDTPS2	WDTPS1	WDTPS0	---- 1111
300004h	CONFIG3L <sup>(4)</sup>	WAIT	BW	EMB1	EMB0	EASHFT	—	—	—	1111 1---
300005h	CONFIG3H	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	— <sup>(2)</sup>	—	—	—	CCP2MX	---- -1
3FFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xx <sup>(5)</sup>
3FFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0000 10x1 <sup>(5)</sup>

图注: x = 未知, — = 未实现。阴影单元未实现, 读为 0。

- 注
- 1: 这些值反映出厂时和上电复位后的未编程状态。在所有其他复位状态中, 配置字节保持原先的编程状态。
  - 2: 程序存储器中这些位的值应始终为 1。这样可确保如果意外地执行了这些单元将会执行 NOP 指令。
  - 3: 该位应始终保持为 0。
  - 4: CONFIG3L 仅在 80 引脚器件中实现。
  - 5: DEVID 的值, 请参阅寄存器 23-7 和寄存器 23-8。这些寄存器为只读寄存器, 用户不能对其进行编程。



**寄存器 23-1: CONFIG1L: 配置寄存器 1 的低字节 (字节地址为 300000h)**

R/WO-1	R/WO-1	R/WO-1	U-0	U-0	U-0	U-0	R/WO-1
DEBUG	XINST	STVREN	—	—	—	—	WDTEN
bit 7							bit 0

**图注:**

R = 可读位                      WO = 一次性写入位                      U = 未实现位, 读为 0  
 -n = 未对器件编程时的值                      1 = 置 1                      0 = 清零

- bit 7                      **DEBUG:** 后台调试器使能位  
                             1 = 禁止后台调试器, RB6 和 RB7 被配置为通用 I/O 引脚  
                             0 = 使能后台调试器, RB6 和 RB7 专用于在线调试
- bit 6                      **XINST:** 扩展指令集使能位  
                             1 = 使能指令集扩展和变址寻址模式  
                             0 = 禁止指令集扩展和变址寻址模式 (传统模式)
- bit 5                      **STVREN:** 堆栈上溢 / 下溢复位使能位  
                             1 = 使能堆栈上溢 / 下溢复位  
                             0 = 禁止堆栈上溢 / 下溢复位
- bit 4-1                      **未实现:** 读为 0
- bit 0                      **WDTEN:** 看门狗定时器使能位  
                             1 = 使能 WDT  
                             0 = 禁止 WDT (控制位为 SWDTEN 位)

**寄存器 23-2: CONFIG1H: 配置寄存器 1 的高字节 (字节地址为 300001h)**

U-0	U-0	U-0	U-0	U-0	R/WO-1	U-0	U-0
— <sup>(1)</sup>	— <sup>(1)</sup>	— <sup>(1)</sup>	— <sup>(1)</sup>	— <sup>(2)</sup>	CP0	—	—
bit 7							bit 0

**图注:**

R = 可读位                      WO = 一次性写入位                      U = 未实现位, 读为 0  
 -n = 未对器件编程时的值                      1 = 置 1                      0 = 清零

- bit 7-4                      **未实现:** 读为 1<sup>(1)</sup>
- bit 3                      **未实现:** 读为 0<sup>(2)</sup>
- bit 2                      **CP0:** 代码保护位  
                             1 = 程序存储器未受代码保护  
                             0 = 程序存储器受代码保护
- bit 1-0                      **未实现:** 读为 0

**注**      1: 程序存储器中这些位的值应始终为 1。这样可确保如果意外地执行了这些单元将会执行 NOP 指令。  
             2: 该位应始终保持为 0。

# PIC18F85J11 系列

寄存器 23-3:        **CONFIG2L: 配置寄存器 2 的低字节**（字节地址为 300002h）

R/WO-1	R/WO-1	U-0	U-0	U-0	R/WO-1	R/WO-1	R/WO-1
IESO	FCMEN	—	—	—	FOSC2	FOSC1	FOSC0
bit 7							bit 0

图注:

R = 可读位                      WO = 一次性写入位                      U = 未实现位, 读为 0  
-n = 未对器件编程时的值                      1 = 置 1                      0 = 清零

- bit 7        **IESO:** 双速启动（内部 / 外部振荡器切换）控制位  
             1 = 使能双速启动  
             0 = 禁止双速启动
- bit 6        **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 5-3      **未实现:** 读为 0
- bit 2-0      **FOSC<2:0>:** 振荡器选择位  
             111 = OSC1/OSC2 作为主振荡器; 具有 CLKO 功能和软件控制 PLL 的 EC 振荡器（ECPLL）  
             110 = OSC1/OSC2 作为主振荡器; 具有 CLKO 功能的 EC 振荡器（EC）  
             101 = OSC1/OSC2 作为主振荡器; 具有软件控制 PLL 的 HS 振荡器（HSPLL）  
             100 = OSC1/OSC2 作为主振荡器; HS 振荡器（HS）  
             011 = 具有 CLKO 的 INTOSC 作为主振荡器; RA7 作为端口引脚; 具有 CLKO 功能和软件控制 PLL 的 EC 振荡器（ECPLL）  
             011 = 具有 CLKO 的 INTOSC 作为主振荡器; RA7 作为端口引脚; 具有 CLKO 功能的 EC 振荡器  
             001 = INTOSC 作为主振荡器, RA<7:6> 作为端口引脚; 具有软件控制 PLL 的 HS 振荡器（HSPLL）  
             000 = INTOSC 作为主振荡器, RA<7:6> 作为端口引脚; HS 振荡器（HS）

寄存器 23-4: **CONFIG2H: 配置寄存器 2 的高字节** (字节地址 300003h)

U-0	U-0	U-0	U-0	R/WO-1	R/WO-1	R/WO-1	R/WO-1
—(1)	—(1)	—(1)	—(1)	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7				bit 0			

**图注:**

R = 可读位                      WO = 一次性写入位                      U = 未实现位, 读为 0  
 -n = 未对器件编程时的值                      1 = 置 1                      0 = 清零

bit 7-4                      **未实现:** 读为 1<sup>(1)</sup>

bit 3-0                      **WDTPS<3:0>:** 看门狗定时器后分频值选择位

1111 = 1:32,768  
 1110 = 1:16,384  
 1101 = 1:8,192  
 1100 = 1:4,096  
 1011 = 1:2,048  
 1010 = 1:1,024  
 1001 = 1:512  
 1000 = 1:256  
 0111 = 1:128  
 0110 = 1:64  
 0101 = 1:32  
 0100 = 1:16  
 0011 = 1:8  
 0010 = 1:4  
 0001 = 1:2  
 0000 = 1:1

**注 1:** 程序存储器中这些位的值应始终为 1。这样可确保如果意外地执行了这些单元将会执行 NOP 指令。

# PIC18F85J11 系列

寄存器 23-5: CONFIG3L: 配置寄存器 3 的低字节 (字节地址为 300004h) <sup>(1)</sup>

R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	U-0	U-0	U-0
WAIT	BW	EMB1	EMB0	EASHFT	—	—	—
bit 7							bit 0

**图注:**

R = 可读位

WO = 一次性写入位

U = 未实现位, 读为 0

-n = 未对器件编程时的值

1 = 置 1

0 = 清零

- bit 7      **WAIT:** 外部总线等待使能位  
1 = 不可使用 WAIT<1:0> 位 (MEMCON<5:4>) 选择等待时间, 器件将不会等待  
0 = 通过 WAIT<1:0> 位 (MEMCON<5:4>) 编程等待时间
- bit 6      **BW:** 数据总线宽度选择位  
1 = 16 位外部总线模式  
0 = 8 位外部总线模式
- bit 5-4    **EMB<1:0>:** 外部存储器总线配置位  
00 = 扩展单片机模式——20 位寻址模式  
01 = 扩展单片机模式——16 位寻址模式  
10 = 扩展单片机模式——12 位寻址模式  
11 = 单片机模式——禁止外部总线
- bit 3      **EASHFT:** 外部地址总线移位使能位  
1 = 使能地址移位——外部地址总线移位到起始地址为 000000h 的单元  
0 = 禁止地址移位——外部地址总线反映 PC 值
- bit 2-0    **未实现:** 读为 0

注 1: CONFIG3L 及其相关位仅在 80 引脚器件中实现。

寄存器 23-6: CONFIG3H: 配置寄存器 3 的高字节 (字节地址为 300005h)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/WO-1
— <sup>(1)</sup>	— <sup>(1)</sup>	— <sup>(1)</sup>	— <sup>(1)</sup>	—	—	—	CCP2MX
bit 7							bit 0

**图注:**

R = 可读位

WO = 一次性写入位

U = 未实现位, 读为 0

-n = 未对器件编程时的值

1 = 置 1

0 = 清零

- bit 7-1    **未实现:** 读为 1<sup>(1)</sup>
- bit 0      **CCP2MX:** CCP2 多路复用位  
1 = CCP2 与 RC1 复用  
0 = CCP2 在单片机模式下与 RE7 复用 (所有器件), 或在扩展单片机模式下与 RB3 复用 (仅限 80 引脚器件)

注 1: 程序存储器中这些位的值应始终为 1。这样可确保如果意外地执行了这些单元将会执行 NOP 指令。

## 寄存器 23-7: DEVID1: PIC18F85J11 系列器件的器件 ID 寄存器 1

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
bit 7							bit 0

图注:

R = 只读位

bit 7-5 **DEV<2:0>:** 器件 ID 位

111 = PIC18F85J11

101 = PIC18F84J11

100 = PIC18F83J11

011 = PIC18F65J11

001 = PIC18F64J11

000 = PIC18F63J11

bit 4-0 **REV<4:0>:** 版本 ID 位

这些位用于表明器件的版本。

## 寄存器 23-8: DEVID2: PIC18F85J11 系列器件的器件 ID 寄存器 2

R	R	R	R	R	R	R	R
DEV10 <sup>(1)</sup>	DEV9 <sup>(1)</sup>	DEV8 <sup>(1)</sup>	DEV7 <sup>(1)</sup>	DEV6 <sup>(1)</sup>	DEV5 <sup>(1)</sup>	DEV4 <sup>(1)</sup>	DEV3 <sup>(1)</sup>
bit 7							bit 0

图注:

R = 只读位

bit 7-0 **DEV<10:3>:** 器件 ID 位 <sup>(1)</sup>

这些位与器件 ID 寄存器 1 中的 DEV<2:0> 位结合使用以标识器件编号。

0011 1001 = PIC18F6XJ11/8XJ11 器件

**注 1:** DEV<10:3> 的值可能会与其他器件系列共用。始终通过使用整个 DEV<10:0> 位序列标识具体器件。

# PIC18F85J11 系列

## 23.2 看门狗定时器 (WDT)

对于 PIC18F85J11 系列器件, WDT 由 INTRC 振荡器驱动。当使能 WDT 时, 也将同时使能时钟源。WDT 超时溢出周期的标称值为 4 ms, 其稳定性与 INTRC 振荡器相同。

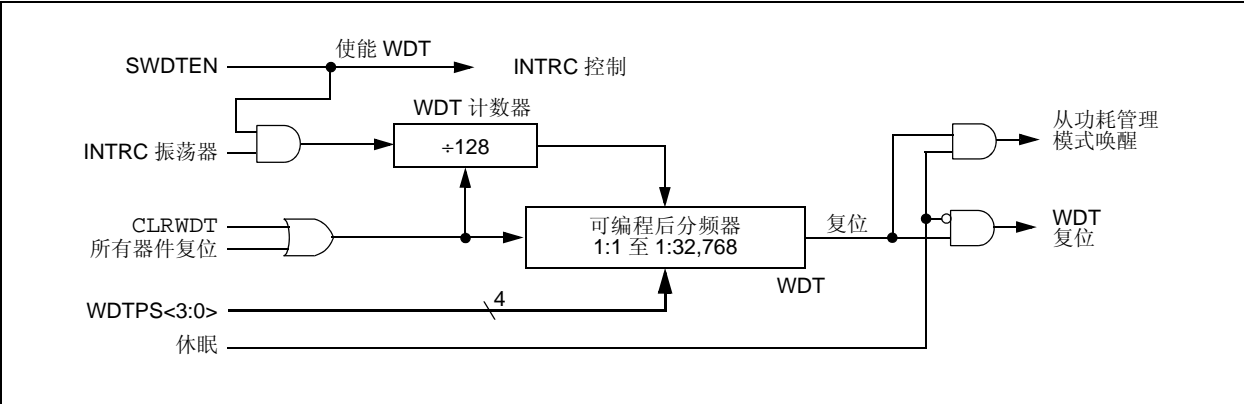
4 ms 的 WDT 超时溢出周期与 16 位后分频器的值相乘。通过配置寄存器 2H 中的 WDTPS 位控制一个多路开关以对 WDT 后分频器的输出进行选择。因此可获得的超时溢出周期范围为 4 ms 到 131.072 秒 (2.18 分钟)。当执行 SLEEP 或 CLRWDT 指令, 或发生时钟故障时 (主时钟或 Timer1 振荡器), WDT 和后分频器将被清零。

- 注 1: 当执行 CLRWDT 和 SLEEP 指令时, WDT 和后分频器的计数值将被清零。
- 2: 当执行 CLRWDT 指令时, 后分频器的计数值将被清零。

### 23.2.1 控制寄存器

WDTCON 寄存器 (寄存器 23-9) 为可读写寄存器。SWDTEN 位用于使能或禁止 WDT 操作。仅当 WDTEN 配置位禁止 WDT 时, 才允许使用软件改写该配置位并使能 WDT。

图 23-1: WDT 框图



寄存器 23-9: WDTCON: 看门狗定时器控制寄存器

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
REGSLP <sup>(1)</sup>	—	—	—	—	—	—	SWDTEN <sup>(2)</sup>
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              **REGSLP:** 稳压器低功耗工作模式使能位  
1 = 当器件进入休眠模式后, 片上稳压器进入低功耗工作模式  
0 = 片上稳压器在休眠模式下依然正常工作
- bit 6-1              **未实现:** 读为 0
- bit 0              **SWDTEN:** 由软件控制的看门狗定时器使能位 <sup>(1)</sup>  
1 = 使能看门狗定时器  
0 = 禁止看门狗定时器

- 注    1: 当满足低电压检测条件时, 自动清零 REGSLP 位。  
2: 当使能 WDTEN 配置位时该位不起作用。

表 23-3: 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	复位值 所在页
RCON	IPEN	—	CM	RI	TO	PD	POR	BOR	58
WDTCON	REGSLP	—	—	—	—	—	—	SWDTEN	58

图注:    — = 未实现, 读为 0。看门狗定时器不使用阴影单元。

# PIC18F85J11 系列

## 23.3 片上稳压器

所有 PIC18F85J11 系列器件使用标称值为 2.5V 的电压为其内核数字逻辑供电。对于需要更高典型工作电压的设计（如 3.3V），PIC18F85J11 系列中的所有器件均包含一个片上稳压器，可使器件内核逻辑运行在 VDD 下。

ENVREG 引脚控制稳压器。把 VDD 连到该引脚使能稳压器，然后稳压器通过其他 VDD 引脚向内核供电。当使能稳压器时，低 ESR 滤波器电容必须连接到 VDDCORE/VCAP 引脚（图 23-2）。这有利于保持稳压器的稳定性。第 26.3 节“直流特性：PIC18F85J11 系列（工业级）”提供了该滤波电容的推荐值。

如果 ENVREG 与 VSS 相连，则禁止稳压器。在这种情况下，独立的 2.5V 标称值的内核逻辑电源必须通过 VDDCORE/VCAP 引脚向器件供电，从而将 I/O 引脚驱动为一个较高的电平，通常为 3.3V。另外，VDDCORE/VCAP 和 VDD 引脚可以连在一起，使器件工作在较低的标称电压下。请参见图 23-2 以了解可能的配置。

### 23.3.1 稳压和低电压检测

当使能时，片上稳压器为数字内核逻辑提供标称值为 2.5V 的恒定电压。稳压器可提供从约为 2.5V 的 VDD，到器件 VDDMAX 的电压范围。该稳压器无法将低于 2.5V 的 VDD 电压拉高。

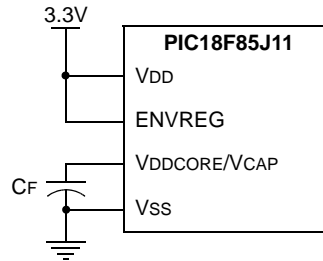
为了防止出现“欠压”条件，当电压被拉至远低于稳压器时，稳压器将进入到跟踪模式。在跟踪模式下，稳压器输出跟随 VDD，通常比 VDD 低 100 mV。

片上稳压器包括一个简单的低电压检测（LVD）电路。当 VDD 降低使 VDDCORE 上的电压约为 2.45V 时，该电路将低电压检测中断标志位 LVDIF（PIR2<2>）置 1 并将已置 1 的 REGSLP（WDTCON<7>）位清零。

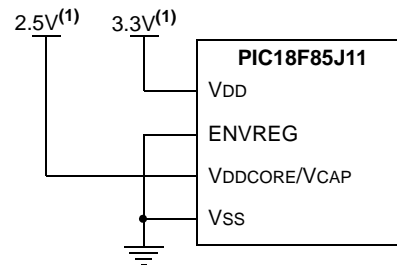
这可被用于产生中断并使应用进入低功耗工作模式，或触发常规关断。低电压检测仅在稳压器使能的情况下可用。

图 23-2: 片上稳压器的连接

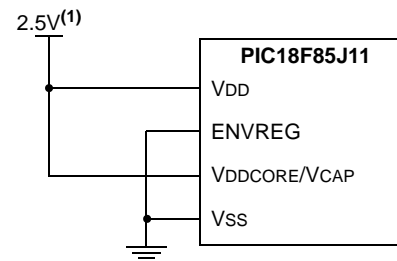
使能稳压器（ENVREG 连接到 VDD）:



禁止稳压器（ENVREG 接地）:



禁止稳压器（VDD 连接到 VDDCORE）:



注 1: 这些为典型的工作电压。请参见第 26.1 节“DC 特性：供电电压”以了解 VDD 和 VDDCORE 的全部工作电压范围。



## 23.3.2 片上稳压器和 BOR

当使能片上稳压器时，PIC18F85J11 系列器件也会有一个简单的欠压复位功能。如果向稳压器提供的电压不足以维持器件全速操作所需的稳定输出电平，那么稳压器复位电路将产生欠压复位。BOR 标志位 (RCON<0>) 捕捉该事件。

在第 5.4 节“欠压复位 (BOR)”和第 5.4.1 节“检测 BOR”中详细介绍了欠压复位的工作原理。

## 23.3.3 上电要求

片上稳压器是为了满足器件的上电要求而设计的。如果应用不使用稳压器，那就必须严格遵守上电条件。上电时，VDDCORE 不能比 VDD 高出 0.3V 以上。

## 23.3.4 休眠模式下的操作

片上稳压器使能时，总会消耗 I<sub>DD</sub> 上的一小部分电流。当器件处于休眠模式时也是如此，尽管此时内核数字逻辑不需要电源。对于电源资源要求高的应用，为了节省更多的资源，应将稳压器配置为只要器件进入休眠模式就会自动禁止。REGSLP 位 (WDTCN<7>) 控制此功能。将此位置 1 将会在休眠模式中禁止稳压器，并将电流消耗降至最低。

将 REGSLP 位置 1 可明显降低休眠模式下的功耗，但由于要确保稳压器经过足够的时间稳定下来，器件的唤醒时间将会增加。

当满足低电压检测条件时，硬件将自动清零 REGSLP 位。REGSLP 位可在软件中再次置 1，这样可以保持稳压器在低压模式下工作。不推荐这样做，但在对闪存执行任何写操作时例外。

## 23.4 双速启动

双速启动功能允许单片机在主时钟源可用之前使用 INTRC 振荡器作为时钟源，从而帮助器件最大限度地缩短从振荡器起振到代码执行之间的延时。通过将 IESO 配置位置 1 可启用该功能。

仅当主振荡器模式为 HS 或 HSPLL（基于晶振）模式时才可使能双速启动。由于 EC 和 ECPLL 模式不需要振荡器起振定时器延时，因此应禁止双速启动。

当使能双速启动时，在发生上电复位并且上电延时定时器发生超时时，器件复位和从休眠模式唤醒都会使器件将自身配置为使用内部振荡器模块作为时钟源来运行。这样几乎可使在主振荡器起振、OST 运行的同时立即执行代码。一旦 OST 超时，器件就自动切换到 PRI\_RUN 模式。

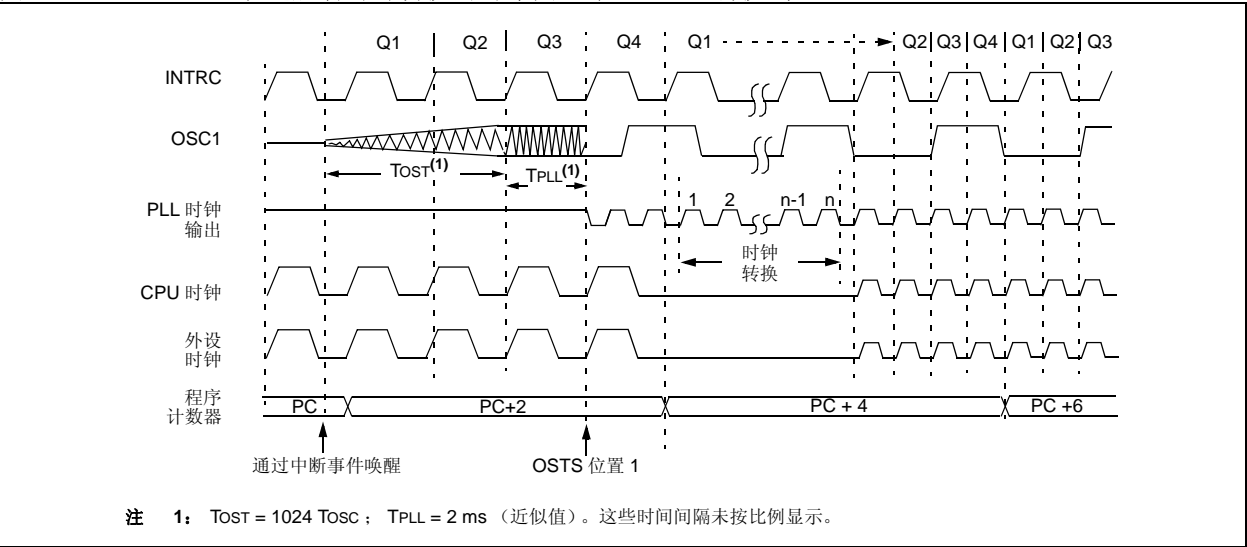
在其他所有的功耗管理模式不使用双速启动。器件将使用当前选定的时钟源直到主时钟源可用为止。该操作与 IESO 位的设置无关。

### 23.4.1 使用双速启动的特别注意事项

当在双速启动中使用 INTRC 振荡器时，器件仍将遵守进入功耗管理模式的正常命令时序，包括执行多条 SLEEP 指令（见第 4.1.4 节“多条 sleep 命令”）。实际上，这意味着在 OST 超时前用户代码可以改变 SCS<1:0> 位的设置或执行 SLEEP 指令。这就使应用程序能短暂地唤醒器件，执行“日常事务”并在器件开始使用主振荡器前返回休眠状态。

用户代码还能通过检查 OSTS 位（OSCCON<3>）的状态来确定当前主时钟源是否正在为系统提供时钟。若该位置 1，则表示主振荡器正在为系统提供时钟。否则，表示当器件从复位或休眠模式唤醒期间由内部振荡电路为系统提供时钟。

图 23-3: 双速启动时钟转换的时序图（从 INTRC 切换到 HSPLL）

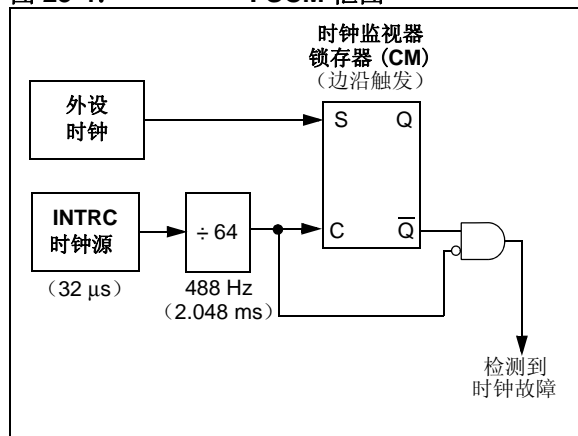


## 23.5 故障保护时钟监视器

故障保护时钟监视器（FSCM）可使单片机在发生外部振荡器故障时，自动将器件时钟切换到内部振荡器模块以保持器件继续运行。将 FCMEN 配置位置 1 可使能 FSCM 功能。

当使能 FSCM 时，INTRC 振荡器将一直保持运行以监视外设时钟，并且在外设时钟发生故障时立即提供备用时钟。时钟监视（如图 23-4 所示）通过创建一个采样时钟信号实现，该信号为 INTRC 输出的 64 分频。这样就使得 FSCM 采样时钟沿之间有充足的时间间隔，从而保证在此时间段内必然会有外设时钟沿出现。外设器件时钟和采样时钟作为时钟监视器锁存器（CM）的输入。CM 在器件时钟源的下降沿置 1，在采样时钟的上升沿清零。

图 23-4: FSCM 框图



在采样时钟的下降沿检测时钟故障。如果在出现采样时钟的下降沿时，CM 仍置 1，就表示检测到时钟故障（如图 23-5 所示）。这将引发以下事件：

- 通过将 OSCFIF（PIR2<7>）置 1，由 FSCM 产生振荡器故障中断；
- 器件时钟源切换为内部振荡器模块（OSCCON 不会被更新，因此无法显示当前时钟源，这就是故障保护状态）；并且
- WDT 复位

切换过程中，对于时序要求较高的应用，内部振荡器模块的后分频频率可能不够稳定。在这些情况下，最好选择另一种时钟配置并进入其他功耗管理模式。可以尝试部分恢复或执行安全关闭。更多详情，请参阅第 4.1.4 节“多条 sleep 命令”和第 23.4.1 节“使用双速启动的特别注意事项”。

FSCM 只能检测出主时钟源或辅助时钟源的故障。如果内部振荡器模块发生故障，将无法检测到故障，当然也就不可能采取任何措施。

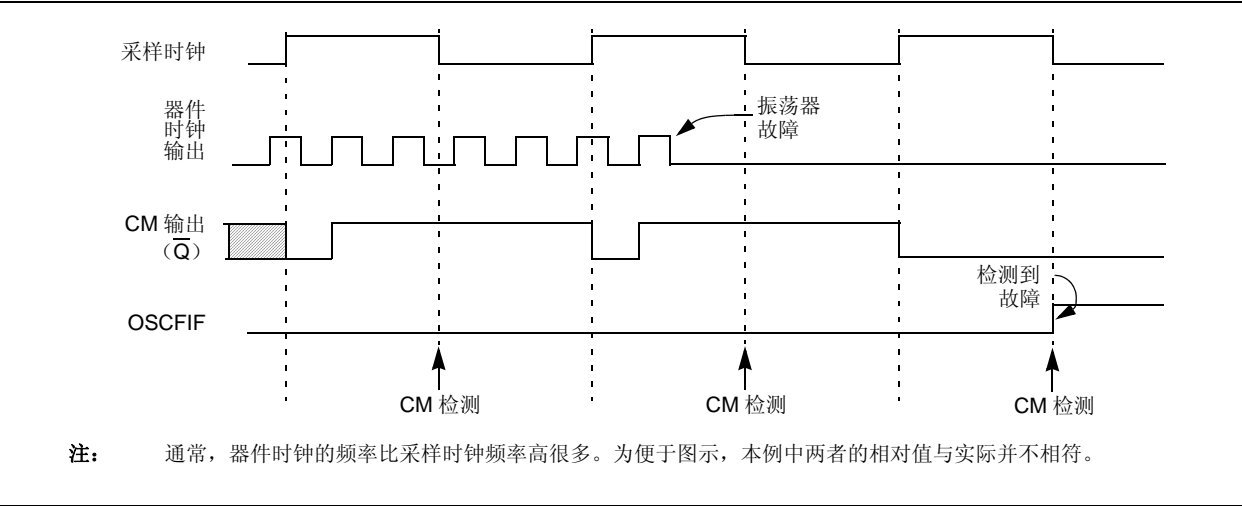
### 23.5.1 FSCM 和看门狗定时器

FSCM 和 WDT 均以 INTRC 振荡器作为时钟源。由于 WDT 使用独立的分频器和计数器工作，当使能 FSCM 时，禁止 WDT 不影响 INTRC 振荡器的运行。

如前所述，当检测到时钟故障时，时钟源将切换到 INTRC 时钟源。这可能意味着代码执行速率会发生很大变化。如果用小分频值的时钟源使能 WDT，时钟速率的下降将使 WDT 发生超时并在随后使器件复位。由于这个原因，故障保护时钟监视器事件也会使 WDT 和后分频器复位，从而使 WDT 从执行速度发生变化的那一刻开始计数，因而减少了发生错误超时的可能性。

如果禁止了该中断，空闲模式下后续中断将使 CPU 开始执行指令，此时由 INTOSC 时钟源为 CPU 提供时钟。

图 23-5: FSCM 时序框图



### 23.5.2 退出故障保护运行模式

器件复位或进入功耗管理模式均可终止故障保护状态。发生复位时，控制器启动在配置寄存器 2H 中指定的主时钟源（伴有如 OST 或 PLL 定时器等振荡器模式所需的起振延时）。INTOSC 振荡器将在主时钟源就绪前提供器件时钟（类似于双速启动）。当主时钟源可用时，时钟源将切换回主时钟（OSCCON 寄存器中的 OSTS 位置 1 以表明当前使用的是主时钟源）。然后，故障保护时钟监视器恢复对外设时钟的监视。

在启动期间，主时钟源可能永远不能就绪。在这种情况下，器件运行将以 INTOSC 振荡器作为时钟源。OSCCON 寄存器将保持复位状态直到进入功耗模式为止。

### 23.5.3 功耗管理模式下的 FSCM 中断

进入功耗管理模式时，时钟多路开关选择由 OSCCON 寄存器选定的时钟源。在功耗管理模式下将恢复对功耗管理时钟源的故障保护监视。

如果在功耗管理运行期间发生了振荡器故障，随后将会发生的事件取决于是否允许了振荡器故障中断。如果允许中断（OSCFIF = 1），代码执行将以 INTRC 复用器作为时钟源。并不会自动转回到发生故障的时钟源。

### 23.5.4 POR 或从休眠中唤醒

FSCM 在器件退出上电复位（POR）或低功耗休眠模式后开始检测振荡器故障。当系统主时钟为 EC 或 INTRC 模式时，监视会在这些事件发生后立即开始。

对于 HS 或 HSPLL 模式，情况会有所不同。由于这类振荡器需要的起振时间可能比 FSCM 采样时钟的时间长很多，因此可能会检测到假的时钟故障。为了避免这一情况，内部振荡器模块会被自动配置为器件时钟并一直工作直到主时钟稳定下来为止（OST 和 PLL 定时器已超时）。这与双速启动模式相同。一旦主时钟稳定下来，INTRC 就将重新作为 FSCM 时钟源。

**注：**防止在发生 POR 或从休眠状态唤醒时发生假振荡器故障中断的逻辑同样也将阻止在发生这些事件后对振荡器故障的检测。通过监视 OSTS 位，并使用定时程序来确定振荡器起振时间是否过长可避免这个问题。即使如此，在检测到振荡器故障时也不会标记任何振荡器故障中断。

正如第 23.4.1 节“使用双速启动的特别注意事项”中所述，在等待系统主时钟稳定的过程中，可以选择另一种时钟配置并进入另一种功耗管理模式。当选择了新的功耗管理模式时，主时钟将被禁止。

## 23.6 程序校验和代码保护

对于 PIC18F85J11 系列中的所有器件，将片内程序存储空间视为一个独立的存储区。配置位 CP0 控制该存储区的代码保护。该位阻止对程序存储空间的外部读写。但对正常的执行模式没有直接影响。

### 23.6.1 配置寄存器保护

有两种方法保护配置寄存器使其免遭破坏性的改写或读取。主要的保护方式是配置位的一次写入功能，该功能阻止对在上电周期内完成编程的位再次进行配置。要阻止不可预见的事件，由于电池故障（如 ESD 事件）产生的配置位更改将导致奇偶校验错误并触发器件复位。

配置寄存器的数据来自于程序存储器中的闪存配置字。当 CP0 位置 1 时，也将保护器件配置的源数据。

## 23.7 在线串行编程

PIC18F85J11 系列单片机可以在最终的应用电路中进行串行编程。只需要 5 根线就可完成这一操作，其中时钟线、数据线各一根，其余 3 根分别是电源线、接地线和编程电压线。这允许用户使用未编程器件制造电路板，仅在产品交付前才对单片机进行编程，从而可使用最新固件或定制固件进行编程。

## 23.8 在线调试器

当  $\overline{\text{DEBUG}}$  配置位被编程为 0 时，可使能在线调试器功能。这一功能允许使用 MPLAB® IDE 进行一些简单的调试。当使能了单片机的这项功能时，某些资源就不再是通用的了。表 23-4 显示了后台调试器所需的资源。

表 23-4: 调试器资源

I/O 引脚:	RB6 和 RB7
堆栈:	2 级
程序存储器:	512 字节
数据存储器:	10 字节

# PIC18F85J11 系列

---

注:

## 24.0 开发支持

一系列软件及硬件开发工具对 PIC® 单片机和 dsPIC® 数字信号控制器提供支持：

- 集成开发环境
  - MPLAB® IDE 软件
- 编译器 / 汇编器 / 链接器
  - 适用于各种器件系列的 MPLAB C 编译器
  - 适用于各种器件系列的 HI-TECH C 编译器
  - MPASM™ 汇编器
  - MPLINK™ 目标链接器 / MPLIB™ 目标库管理器
  - 适用于各种器件系列的 MPLAB 汇编器 / 链接器 / 库管理器
- 模拟器
  - MPLAB SIM 软件模拟器
- 仿真器
  - MPLAB REAL ICE™ 在线仿真器
- 在线调试器
  - MPLAB ICD 3
  - PICKit™ 3 Debug Express
- 器件编程器
  - PICKit™ 2 编程器
  - MPLAB PM3 器件编程器
- 低成本演示 / 开发板、评估工具包及入门工具包

## 24.1 MPLAB 集成开发环境软件

MPLAB IDE 软件为 8/16/32 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）
  - 在线仿真器（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
- 鼠标停留在变量上进行查看的功能
- 将变量从源代码窗口拖放到 Watch（观察）窗口
- 丰富的在线帮助
- 集成了可选的第三方工具，如 IAR C 编译器

MPLAB IDE 可以让您：

- 编辑源文件（C 语言或汇编语言）
- 点击一次即可完成编译或汇编，并将代码下载到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（C 语言或汇编语言）
  - 混合 C 语言和汇编语言
  - 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强大的工具时的学习时间。

## 24.2 适用于各种器件系列的 MPLAB C 编译器

MPLAB C 编译器代码开发系统是完整的 ANSI C 编译器，适用于 Microchip 的 PIC18、PIC24 和 PIC32 系列单片机及 dsPIC30 和 dsPIC33 系列数字信号控制器。这些编译器提供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，编译器提供针对 MPLAB IDE 调试器优化的符号信息。

## 24.3 适用于各种器件系列的 HI-TECH C 编译器

HI-TECH C 编译器代码开发系统是完整的 ANSI C 编译器，适用于 Microchip 的 PIC 系列单片机及 dsPIC 系列数字信号控制器。这些编译器提供强大的集成功能和全知代码生成能力，且使用方便。

为便于源代码调试，编译器提供针对 MPLAB IDE 调试器优化的符号信息。

编译器包括一个宏汇编器、链接器、预处理程序和单步驱动程序，可以在多种平台上运行。

## 24.4 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于 PIC10/12/16/18 MCU。

MPASM 汇编器可生成用于 MPLINK 目标链接器的可重定位目标文件、Intel® 标准 HEX 文件、详细描述存储器使用状况和符号参考的 MAP 文件、包含源代码行及生成机器码的绝对 LST 文件以及用于调试的 COFF 文件。

MPASM 汇编器具有如下特性：

- 集成在 MPLAB 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

## 24.5 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程序。这样可使大型库在许多不同应用中被高效地利用。

目标链接器 / 库管理器具有如下特性：

- 高效地连接单个的库而不是许多小文件
- 通过将相关的模块组合在一起来增强代码的可维护性
- 只要列出、替换、删除和抽取模块，便可灵活地创建库

## 24.6 适用于各种器件系列的 MPLAB 汇编器、链接器和库管理器

MPLAB 汇编器为 PIC24、PIC32 和 dsPIC 器件从符号汇编语言生成可重定位机器码。MPLAB 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性：

- 支持整个器件指令集
- 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



## 24.7 MPLAB SIM 软件模拟器

MPLAB SIM 软件模拟器通过在指令级对 PIC MCU 和 dsPIC® DSC 进行模拟，可在 PC 主机环境下进行代码开发。对于任何给定的指令，都可以对数据区进行检查或修改，并通过一个全面的激励控制器来施加激励。可以将各寄存器记录在文件中，以便进行进一步的运行时分析。跟踪缓冲区和逻辑分析器的显示使软件模拟器还能记录和跟踪程序的执行、I/O 的动作、大部分的外设及内部寄存器。

MPLAB SIM 软件模拟器完全支持使用 MPLAB C 编译器以及 MPASM 和 MPLAB 汇编器的符号调试。该软件模拟器可用于在硬件实验室环境外灵活地开发和调试代码，是一款完美且经济的软件开发工具。

## 24.8 MPLAB REAL ICE 在线仿真器系统

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

该仿真器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与在线调试器系统兼容的连接器和 (RJ11) 或新型抗噪声、高速低压差分信号 (LVDS) 互连电缆 (CAT5) 与目标板相连。

可通过 MPLAB IDE 下载将来版本的固件，对该仿真器进行现场升级。在即将推出的 MPLAB IDE 版本中，会支持许多新器件，还将增加一些新特性。在同类仿真器中，MPLAB REAL ICE 的优势十分明显：低成本、全速仿真、运行时变量查看、跟踪分析、复杂断点、耐用的探针接口及较长（长达 3 米）的互连电缆。

## 24.9 MPLAB ICD 3 在线调试器系统

MPLAB ICD 3 在线调试器系统是 Microchip 成本效益最高的高速硬件调试器 / 编程器，适用于 Microchip 闪存数字信号控制器 (DSC) 和单片机 (MCU) 器件。结合 MPLAB 集成开发环境 (IDE) 所具有的功能强大但易于使用的图形用户界面，该调试器可对 PIC® 闪存单片机和 dsPIC® DSC 进行调试和编程。

MPLAB ICD 3 在线调试器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与 MPLAB ICD 2 或 MPLAB REAL ICE 系统兼容的连接器和 (RJ-11) 与目标板相连。MPLAB ICD 3 支持所有 MPLAB ICD 2 转接器。

## 24.10 PICkit 3 在线调试器 / 编程器及 PICkit 3 Debug Express

结合 MPLAB 集成开发环境 (IDE) 所具有的功能强大的图形用户界面，MPLAB PICkit 3 可对 PIC® 闪存单片机和 dsPIC® 数字信号控制器进行调试和编程，且价位较低。MPLAB PICkit 3 通过全速 USB 接口与设计工程师的 PC 相连，并利用 Microchip 调试 (RJ-11) 连接器 (与 MPLAB ICD 3 和 MPLAB REAL ICE 兼容) 与目标板相连。连接器使用两个器件 I/O 引脚和复位线来实现在线调试和在线串行编程。

PICkit 3 Debug Express 包括 PICkit 3、演示板和单片机、连接电缆和光盘 (内含用户指南、课程、教程、编译器和 MPLAB IDE 软件)。

## 24.11 PICkit 2 开发编程器 / 调试器及 PICkit 2 Debug Express

PICkit™ 2 开发编程器 / 调试器是一款低成本开发工具，具有易于使用的界面，适用于对 Microchip 的闪存系列单片机进行编程和调试。这一全功能的 Windows® 编程界面支持低档（PIC10F、PIC12F5xx 和 PIC16F5xx）、中档（PIC12F6xx 和 PIC16F）、PIC18F、PIC24、dsPIC30、dsPIC33 和 PIC32 系列的 8 位、16 位及 32 位单片机，以及许多 Microchip 串行 EEPROM 产品。结合 Microchip 功能强大的 MPLAB 集成开发环境 (IDE)，PICkit 2 可对大多数 PIC® 单片机进行在线调试。即使 PIC 单片机已嵌入应用，在线调试功能仍可以运行、暂停和单步执行程序。在断点处暂停时，可以检查和修改文件寄存器。

PICkit 2 Debug Express 包括 PICkit 2、演示板和单片机、连接电缆和光盘（内含用户指南、课程、教程、编译器 and MPLAB IDE 软件）。

## 24.12 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款符合 CE 规范的通用器件编程器，在 VDDMIN 和 VDDMAX 点对其可编程电压进行校验以确保可靠性最高。它有一个用来显示菜单和错误消息的大 LCD 显示器（128 x 64），以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSP™ 电缆。在单机模式下，MPLAB PM3 器件编程器不必与 PC 相连即可对 PIC 器件进行读取、校验和编程。在该模式下它还可设置代码保护。MPLAB PM3 通过 RS-232 或 USB 电缆连接到 PC 主机上。MPLAB PM3 具备高速通信能力以及优化算法，可对具有大存储器的器件进行快速编程。它还包含了 MMC 卡，用于文件存储及数据应用。

## 24.13 演示 / 开发板、评估工具包及入门工具包

有许多演示、开发和评估板可用于各种 PIC MCU 和 dsPIC DSC，实现对全功能系统的快速应用开发。大多数的演示、开发和评估板都有实验布线区，供用户添加定制电路；还有应用固件和源代码，用于检查和修改。

这些板支持多种功能部件，包括 LED、温度传感器、开关、扬声器、RS-232 接口、LCD 显示器、电位计和附加 EEPROM 存储器。

演示和开发板可用于教学环境，在实验布线区设计定制电路，从而掌握各种单片机应用。

除了 PICDEM™ 和 dsPICDEM™ 演示 / 开发板系列电路外，Microchip 还有一系列评估工具包和演示软件，适用于模拟滤波器设计、KEELOQ® 数据安全产品 IC、CAN、IrDA®、PowerSmart 电池管理、SEEVAL® 评估系统、Σ-Δ ADC、流速传感器，等等。

同时还提供入门工具包，其中包含体验指定器件功能所需的所有软硬件。通常提供单个应用以及调试功能，都包含在一块电路板上。

有关演示、开发和评估工具包的完整列表，请访问 Microchip 网站（[www.microchip.com](http://www.microchip.com)）。

## 25.0 指令集汇总

PIC18F85J11 系列器件具有一个含有 75 条 PIC18 内核指令的标准指令集和一个含有 8 条新指令（优化递归或软件堆栈代码）的扩展指令集。本章后面的部分将讨论该扩展指令集。

### 25.1 标准指令集

标准的 PIC18 指令集与以前的 PIC<sup>®</sup>MCU 指令集相比，添加了很多增强功能，并保持了易于从 PIC MCU 指令集移植的特点。大部分指令只占用一个程序存储字（16 位），只有 4 条指令需要两个程序存储单元。

每条单字指令都是一个 16 位字，由操作码（指明指令类型）和一个或多个操作数（进一步指定指令操作）组成。

整个指令集具有高度的正交性，分为以下 4 种基本类型：

- 字节操作类指令
- 位操作类指令
- 立即数操作类指令
- 控制操作类指令

表 25-2 为 PIC18 指令集汇总，列出了字节操作类指令、位操作类指令以及立即数和控制操作类指令。表 25-1 给出了对操作码字段的说明。

大部分字节操作类指令含有三种操作数：

1. 文件选择寄存器（由 **f** 指定）。
2. 保存结果的目标寄存器（由 **d** 指定）。
3. 被访问的寄存器（由 **a** 指定）。

文件选择寄存器指示符 **f** 指定了指令将会使用哪一个文件选择寄存器。目标寄存器指示符 **d** 指定了操作结果的存放位置。如果 **d** 为 0，操作结果存入 WREG 寄存器。如果 **d** 为 1，操作结果将存入指令指定的文件选择寄存器中。

所有位操作类指令都含有三种操作数：

1. 文件选择寄存器（由 **f** 指定）。
2. 文件选择寄存器中的位（由 **b** 指定）。
3. 被访问的寄存器（由 **a** 指定）。

位域指示符 **b** 选择操作所影响的位的编号，而文件选择寄存器指示符 **f** 则代表该位所在寄存器的编号。

立即数操作类指令可以使用以下操作数：

- 要装入文件选择寄存器中的立即数（由 **k** 指定）。
- 要装入立即数的目标 FSR 寄存器（由 **f** 指定）。
- 不需要操作数（由 — 指定）。

控制操作类指令可以使用以下操作数：

- 程序存储器地址（由 **n** 指定）。
- CALL 或 RETURN 指令的模式（由 **s** 指定）。
- 表读和表写指令的模式（由 **m** 指定）。
- 不需要操作数（由 — 指定）。

除了 4 条双字指令外，所有的指令都是单字指令。双字指令将所需的信息保存在 32 位中。第二个字的高 4 位都是 1。如果第二个字作为一条指令执行，它会执行 NOP 指令。

除非条件测试结果为 true 或者指令执行改变了程序计数器的值，否则执行所有的单字指令都只需要一个指令周期。对于上述两种特殊情况，执行指令需要两个指令周期，第二个指令周期中执行一条 NOP 指令。

执行双字指令需要两个指令周期。

每个指令周期由 4 个振荡周期组成。因此，对于频率为 4 MHz 的振荡器，其正常的指令执行时间为 1 μs。如果条件测试为 true 或指令执行改变了程序计数器的值，则该指令的执行时间为 2 μs。双字跳转指令（如果为 true）的执行需要 3 μs。

图 25-1 给出了指令的几种通用格式。所有示例均使用 nnn 约定来表示十六进制数。

指令集汇总（见表 25-2）列出了可被 Microchip MPASM<sup>™</sup> 汇编器识别的标准指令。

第 25.1.1 节“标准指令集”对每条指令进行了介绍。

# PIC18F85J11 系列

表 25-1: 操作码字段说明

字段	说明
a	快速操作 RAM 位： a = 0: 快速操作 RAM 内的 RAM 单元（BSR 寄存器被忽略） a = 1: 由 BSR 寄存器指定的 RAM 存储区
bbb	8 位文件选择寄存器的位地址（0 到 7）。
BSR	存储区选择寄存器。用于选择当前的 RAM 存储区。
C, DC, Z, OV, N	ALU 状态位: 进位标志位、半进位标志位、全零标志位、溢出标志位和负标志位。
d	目标寄存器选择位： d = 0: 结果保存至 WREG 寄存器 d = 1: 结果保存至文件选择寄存器 f
dest	目标寄存器: 可以是 WREG 寄存器或指定的寄存器单元。
f	8 位寄存器地址（00h 到 FFh）或 2 位 FSR 指示符（0h 到 3h）。
f <sub>s</sub>	12 位寄存器地址（000h 到 FFFh）。这是源地址。
f <sub>d</sub>	12 位寄存器地址（000h 到 FFFh）。这是目标地址。
GIE	全局中断允许位。
k	立即数字段、常数或者标号（可能是 8 位、12 位或 20 位的值）。
label	标号名称。
mm	表读和表写指令的 TBLPTR 寄存器模式。 只与表读与表写指令一起使用：
*	不改变寄存器（如用于表读和表写的 TBLPTR）
*+	后增寄存器（如用于表读和表写的 TBLPTR）
*-	后减寄存器（如用于表读和表写的 TBLPTR）
++	预增寄存器（如用于表读和表写的 TBLPTR）
n	相对跳转指令的相对地址（二进制补码）或调用/跳转和返回指令的直接地址。
PC	程序计数器。
PCL	程序计数器的低字节。
PCH	程序计数器的高字节。
PCLATH	程序计数器的高字节锁存器。
PCLATU	程序计数器的最高字节锁存器。
$\overline{PD}$	掉电位。
PRODH	乘积的高字节。
PRODL	乘积的低字节。
s	快速调用/返回模式选择位： s = 0: 不对影子寄存器进行更新，也不用影子寄存器的内容更新其他寄存器 s = 1: 将寄存器的值存入影子寄存器或把影子寄存器的值装入其对应的寄存器（快速模式）
TBLPTR	21 位表指针（指向程序存储单元）。
TABLAT	8 位表锁存器。
$\overline{TO}$	超时溢出位。
TOS	栈顶。
u	未用或不变。
WDT	看门狗定时器。
WREG	工作寄存器（累加器）。
x	任意值（0 或 1）。汇编器将产生 x = 0 的代码。为了与所有 Microchip 软件工具兼容，建议使用这种格式。
z <sub>s</sub>	对寄存器（源）进行间接寻址的 7 位偏移量。
z <sub>d</sub>	对寄存器（目标）进行间接寻址的 7 位偏移量。
{ }	可选参数。
[text]	表示变址地址。
(text)	text 的内容。
[expr]<n>	表示由指针 expr 指定的寄存器中的位 n。
→	赋值。
< >	寄存器位域。
∈	表示属于某个集合。
斜体文字	用户定义项（字体为 Courier New）。

图 25-1: 指令的一般格式

面向字节的文件选择寄存器操作		指令示例
15	10 9 8 7	0
操作码	d a f (寄存器地址)	ADDWF MYREG, W, B
d = 0 表示结果存入 WREG 寄存器 d = 1 表示结果存入文件选择寄存器 (f) a = 0 强制使用快速操作存储区 a = 1 根据 BSR 选择存储区 f = 8 位文件选择寄存器地址		
字节到字节的传送操作 (双字)		
15	12 11	0
操作码	f (源寄存器地址)	MOVFF MYREG1, MYREG2
15	12 11	0
1111	f (目标寄存器地址)	
f = 12 位文件选择寄存器地址		
面向位的文件选择寄存器操作		
15	12 11 9 8 7	0
操作码	b (位地址) a f (寄存器地址)	BSF MYREG, bit, B
b = 占 3 位, 表示文件选择寄存器 (f) 中位的位置 a = 0 强制使用快速操作存储区 a = 1 根据 BSR 选择存储区 f = 8 位文件选择寄存器地址		
立即数操作		
15	8 7	0
操作码	k (立即数)	MOVLW 7Fh
k = 8 位立即数值		
控制操作		
CALL、GOTO 和跳转类操作		
15	8 7	0
操作码	n<7:0> (立即数)	GOTO Label
15	12 11	0
1111	n<19:8> (立即数)	
n = 20 位立即数值		
15	8 7	0
操作码	S n<7:0> (立即数)	CALL MYFUNC
15	12 11	0
1111	n<19:8> (立即数)	
S = 快速位		
15	11 10	0
操作码	n<10:0> (立即数)	BRA MYFUNC
15	8 7	0
操作码	n<7:0> (立即数)	BC MYFUNC

# PIC18F85J11 系列

表 25-2: PIC18F85J11 系列指令集

助记符， 操作数	说明	周期数	16 位指令字				受影响的状态位	注
			MSb		LSb			
字节操作类指令								
ADDWF f, d, a	WREG 与 f 相加	1	0010	01da	ffff	ffff	C, DC, Z, OV, N	1, 2
ADDWFC f, d, a	WREG 与 f 带进位相加	1	0010	00da	ffff	ffff	C, DC, Z, OV, N	1, 2
ANDWF f, d, a	WREG 与 f 作逻辑与运算	1	0001	01da	ffff	ffff	Z, N	1, 2
CLRF f, a	将 f 清零	1	0110	101a	ffff	ffff	Z	2
COMF f, d, a	将 f 取反	1	0001	11da	ffff	ffff	Z, N	1, 2
CPFSEQ f, a	f 与 WREG 做比较，相等则跳过	1 (2 或 3)	0110	001a	ffff	ffff	无	4
CPFSGT f, a	f 与 WREG 做比较，大于则跳过	1 (2 或 3)	0110	010a	ffff	ffff	无	4
CPFSLT f, a	f 与 WREG 做比较，小于则跳过	1 (2 或 3)	0110	000a	ffff	ffff	无	1, 2
DECF f, d, a	f 减 1	1	0000	01da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4
DECFSZ f, d, a	f 减 1，为 0 则跳过	1 (2 或 3)	0010	11da	ffff	ffff	无	1, 2, 3, 4
DCFSNZ f, d, a	f 减 1，非 0 则跳过	1 (2 或 3)	0100	11da	ffff	ffff	无	1, 2
INCF f, d, a	f 加 1	1	0010	10da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4
INCFSZ f, d, a	f 加 1，为 0 则跳过	1 (2 或 3)	0011	11da	ffff	ffff	无	4
INFSNZ f, d, a	f 加 1，非 0 则跳过	1 (2 或 3)	0100	10da	ffff	ffff	无	1, 2
IORWF f, d, a	WREG 与 f 作逻辑或运算	1	0001	00da	ffff	ffff	Z, N	1, 2
MOVF f, d, a	移动 f	1	0101	00da	ffff	ffff	Z, N	1
MOVFF f <sub>s</sub> , f <sub>d</sub>	f <sub>s</sub> （源）地址移入（第一个字） f <sub>d</sub> （目标）地址移入（第二个字）	2	1100	ffff	ffff	ffff	无	
			1111	ffff	ffff	ffff		
MOVWF f, a	将 WREG 移入 f	1	0110	111a	ffff	ffff	无	
MULWF f, a	WREG 乘以 f	1	0000	001a	ffff	ffff	无	1, 2
NEGF f, a	对 f 取补	1	0110	110a	ffff	ffff	C, DC, Z, OV, N	
RLCF f, d, a	对 f 执行带进位的循环左移	1	0011	01da	ffff	ffff	C, Z, N	1, 2
RLNCF f, d, a	f 循环左移（不带进位）	1	0100	01da	ffff	ffff	Z, N	
RRCF f, d, a	对 f 执行带进位的循环右移	1	0011	00da	ffff	ffff	C, Z, N	
RRNCF f, d, a	f 循环右移（不带进位）	1	0100	00da	ffff	ffff	Z, N	
SETF f, a	将 f 置为全 1	1	0110	100a	ffff	ffff	无	1, 2
SUBFWB f, d, a	WREG 减去 f（带借位）	1	0101	01da	ffff	ffff	C, DC, Z, OV, N	
SUBWF f, d, a	f 减去 WREG	1	0101	11da	ffff	ffff	C, DC, Z, OV, N	1, 2
SUBWFB f, d, a	f 减去 WREG（带借位）	1	0101	10da	ffff	ffff	C, DC, Z, OV, N	
SWAPF f, d, a	将 f 中的两个半字节进行交换	1	0011	10da	ffff	ffff	无	4
TSTFSZ f, a	测试 f，为 0 则跳过	1 (2 或 3)	0110	011a	ffff	ffff	无	1, 2
XORWF f, d, a	WREG 与 f 作逻辑异或运算	1	0001	10da	ffff	ffff	Z, N	

- 注 1: PORT 寄存器的值随端口状态的变化而不断修改 (例如, MOVF PORTB, 1, 0), 修改时使用的值是引脚上的当前值。例如, 如果将一个引脚配置为输入, 虽然其数据锁存器中的值为 1, 但如果此时外部器件将该引脚驱动为低电平, 则被写回数据总线的值将是 0。
- 2: 当对 TMR0 寄存器 (并且 d = 1) 执行该指令时, 如果已将预分频器分配给了 TMR0, 则将其清零。
- 3: 如果程序计数器 (PC) 被修改或者条件检测为 true, 则该指令需要两个周期。第二个周期执行一条 NOP 指令。
- 4: 某些指令是双字指令。除非指令的第一个字获取这 16 位中包含的信息, 否则第二个字将作为 NOP 指令执行。这将确保所有程序存储单元存储的都是合法的指令。

表 25-2: PIC18F85J11 系列指令集 (续)

助记符, 操作数		说明	周期数	16 位指令字				受影响的状态位	注
				MSb		LSb			
位操作类指令									
BCF	f, b, a	将 f 中的某位清零	1	1001	bbba	ffff	ffff	无	1, 2
BSF	f, b, a	将 f 中的某位置 1	1	1000	bbba	ffff	ffff	无	1, 2
BTFSC	f, b, a	检测 f 中的某位, 为 0 则跳过	1 (2 或 3)	1011	bbba	ffff	ffff	无	3, 4
BTFSS	f, b, a	检测 f 中的某位, 为 1 则跳过	1 (2 或 3)	1010	bbba	ffff	ffff	无	3, 4
BTG	f, b, a	将 f 中的某位取反	1	0111	bbba	ffff	ffff	无	1, 2
控制操作类指令									
BC	n	进位则跳转	1 (2)	1110	0010	nnnn	nnnn	无	4
BN	n	为负则跳转	1 (2)	1110	0110	nnnn	nnnn	无	
BNC	n	无进位则跳转	1 (2)	1110	0011	nnnn	nnnn	无	
BNN	n	不为负则跳转	1 (2)	1110	0111	nnnn	nnnn	无	
BNOV	n	不溢出则跳转	1 (2)	1110	0101	nnnn	nnnn	无	
BNZ	n	不为零则跳转	1 (2)	1110	0001	nnnn	nnnn	无	
BOV	n	溢出则跳转	1 (2)	1110	0100	nnnn	nnnn	无	
BRA	n	无条件跳转	2	1101	0nnn	nnnn	nnnn	无	
BZ	n	为零则跳转	1 (2)	1110	0000	nnnn	nnnn	无	
CALL	n, s	调用子程序	2	1110	110s	kkkk	kkkk	无	
		第一个字		1111	kkkk	kkkk	kkkk		
		第二个字							
CLRWDT	—	将看门狗定时器清零	1	0000	0000	0000	0100	$\overline{\text{TO}}$ 和 $\overline{\text{PD}}$	
DAW	—	对 WREG 进行十进制调整	1	0000	0000	0000	0111	C	
GOTO	n	跳转到地址	2	1110	1111	kkkk	kkkk	无	
		第一个字		1111	kkkk	kkkk	kkkk		
		第二个字							
NOP	—	空操作	1	0000	0000	0000	0000	无	
NOP	—	空操作	1	1111	xxxx	xxxx	xxxx	无	
POP	—	弹出返回堆栈栈顶 (TOS) 内容	1	0000	0000	0000	0110	无	
PUSH	—	压入返回堆栈栈顶 (TOS)	1	0000	0000	0000	0101	无	
RCALL	n	相对调用	2	1101	1nnn	nnnn	nnnn	无	
RESET		用软件使器件复位	1	0000	0000	1111	1111	全部	
RETFIE	s	中断返回允许	2	0000	0000	0001	000s	GIE/GIEH, PEIE/GIEL	
RETLW	k	返回时将立即数送入 WREG	2	0000	1100	kkkk	kkkk	无	
RETURN	s	从子程序返回	2	0000	0000	0001	001s	无	
SLEEP	—	进入待机模式	1	0000	0000	0000	0011	$\overline{\text{TO}}$ 和 $\overline{\text{PD}}$	

注 1: PORT 寄存器的值随端口状态的变化而不断修改 (例如, MOVF PORTB, 1, 0), 修改时使用的值是引脚上的当前值。例如, 如果将一个引脚配置为输入, 虽然其数据锁存器中的值为 1, 但如果此时外部器件将该引脚驱动为低电平, 则被写回数据总线的数值将是 0。

2: 当对 TMR0 寄存器 (并且 d = 1) 执行该指令时, 如果已将预分频器分配给了 TMR0, 则将其清零。

3: 如果程序计数器 (PC) 被修改或者条件检测为 true, 则该指令需要两个周期。第二个周期执行一条 NOP 指令。

4: 某些指令是双字指令。除非指令的第一个字获取这 16 位中包含的信息, 否则第二个字将作为 NOP 指令执行。这将确保所有程序存储单元存储的都是合法的指令。

# PIC18F85J11 系列

表 25-2: PIC18F85J11 系列指令集 (续)

助记符, 操作数	说明	周期数	16 位指令字				受影响的状态位	注
			MSb		LSb			
立即数操作类指令								
ADDLW k	WREG 与立即数相加	1	0000	1111	kkkk	kkkk	C, DC, Z, OV, N	
ANDLW k	立即数与 WREG 作逻辑与运算	1	0000	1011	kkkk	kkkk	Z, N	
IORLW k	立即数与 WREG 作逻辑或运算	1	0000	1001	kkkk	kkkk	Z, N	
LFSR f, k	将立即数 (12 位) 第二个字 移入 FSR(f) 第一个字	2	1110	1110	00ff	kkkk	无	
MOVLB k	将立即数移入 BSR<3:0>	1	0000	0001	0000	kkkk	无	
MOVLW k	将立即数移入 WREG	1	0000	1110	kkkk	kkkk	无	
MULLW k	立即数与 WREG 相乘	1	0000	1101	kkkk	kkkk	无	
RETLW k	返回时将立即数送入 WREG	2	0000	1100	kkkk	kkkk	无	
SUBLW k	立即数减去 WREG	1	0000	1000	kkkk	kkkk	C, DC, Z, OV, N	
XORLW k	立即数与 WREG 作逻辑异或运算	1	0000	1010	kkkk	kkkk	Z, N	
数据存储器 ↔ 程序存储器操作类指令								
TBLRD*	表读	2	0000	0000	0000	1000	无	
TBLRD*+	后增表读	2	0000	0000	0000	1001	无	
TBLRD*-	后减表读		0000	0000	0000	1010	无	
TBLRD+*	预增表读		0000	0000	0000	1011	无	
TBLWT*	表写		0000	0000	0000	1100	无	
TBLWT*+	后增表写		0000	0000	0000	1101	无	
TBLWT*-	后减表写		0000	0000	0000	1110	无	
TBLWT+*	预增表写	0000	0000	0000	1111	无		

- 注 1: PORT 寄存器的值随端口状态的变化而不断修改 (例如, MOVF PORTB, 1, 0), 修改时使用的值是引脚上的当前值。例如, 如果将一个引脚配置为输入, 虽然其数据锁存器中的值为 1, 但如果此时外部器件将该引脚驱动为低电平, 则被写回数据总线的数据值将是 0。
- 2: 当对 TMR0 寄存器 (并且 d = 1) 执行该指令时, 如果已将预分频器分配给了 TMR0, 则将其清零。
- 3: 如果程序计数器 (PC) 被修改或者条件检测为 true, 则该指令需要两个周期。第二个周期执行一条 NOP 指令。
- 4: 某些指令是双字指令。除非指令的第一个字获取这 16 位中包含的信息, 否则第二个字将作为 NOP 指令执行。这将确保所有程序存储单元存储的都是合法的指令。





# PIC18F85J11 系列

## ADDWFC W 与 f 带进位相加

语法: ADDWFC f {,d {,a}}

操作数:  $0 \leq f \leq 255$

$d \in [0, 1]$

$a \in [0, 1]$

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

受影响的状态位: N、OV、C、DC 和 Z

机器码:

0010	00da	ffff	ffff
------	------	------	------

说明: 将 W、进位标志位与数据存储单元 f 的内容相加。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存储在数据存储单元 f 中。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要  $f \leq 95$  (5Fh)，指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

示例: ADDWFC REG, 0, 1

执行指令前

进位标志位 = 1  
REG = 02h  
W = 4Dh

执行指令后

进位标志位 = 0  
REG = 02h  
W = 50h

## ANDLW 立即数与 W 寄存器作逻辑与运算

语法: ANDLW k

操作数:  $0 \leq k \leq 255$

操作:  $(W) .AND. k \rightarrow W$

受影响的状态位: N 和 Z

机器码:

0000	1011	kkkk	kkkk
------	------	------	------

说明: W 的内容与 8 位立即数 k 作逻辑与运算。结果保存在 W 寄存器中。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 W

示例: ANDLW 05Fh

执行指令前

W = A3h

执行指令后

W = 03h

ANDWF

将 W 与 f 作逻辑与运算

语法:

ANDWF f {,d {,a}}

操作数:

0 ≤ f ≤ 255  
d ∈ [0, 1]  
a ∈ [0, 1]

操作:

(W) .AND. (f) → dest

受影响的状态位:

N 和 Z

机器码:

0001	01da	ffff	ffff
------	------	------	------

说明:

将 W 的内容与寄存器 f 的内容进行逻辑与运算。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例:

ANDWF	REG, 0, 0
-------	-----------

执行指令前

W = 17h

REG = C2h

执行指令后

W = 02h

REG = C2h

BC

进位则跳转

语法:

BC n

操作数:

-128 ≤ n ≤ 127

操作:

如果进位标志位为 1，  
(PC) + 2 + 2n → PC

受影响的状态位:

无

机器码:

1110	0010	nnnn	nnnn
------	------	------	------

说明:

如果进位标志位为 1，那么程序将跳转。二进制补码 2n 与 PC 相加。因为 PC 要先递增以便取出下一条指令，所以新地址都将为 PC + 2 + 2n。这种情况下，该指令是一条双周期指令。

指令字数:

1

指令周期数:

1(2)

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例:

HERE	BC	5
------	----	---

执行指令前

PC = 地址 (HERE)

执行指令后

如果进位标志位 = 1 ;

PC = 地址 (HERE + 12)

如果进位标志位 = 0 ;

PC = 地址 (HERE + 2)

# PIC18F85J11 系列

## BCF 将 f 中的某位清零

语法: BCF f, b {,a}

操作数:  $0 \leq f \leq 255$

$0 \leq b \leq 7$

$a \in [0, 1]$

操作:  $0 \rightarrow f \langle b \rangle$

受影响的状态位: 无

机器码: 

1001	bbba	ffff	ffff
------	------	------	------

说明: 将寄存器 f 中的位 b 清零。

如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集, 只要  $f \leq 95$  (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例: BCF FLAG\_REG, 7, 0

执行指令前

FLAG\_REG = C7h

执行指令后

FLAG\_REG = 47h

## BN 为负则跳转

语法: BN n

操作数:  $-128 \leq n \leq 127$

操作: 如果负标志位为 1,  
 $(PC) + 2 + 2n \rightarrow PC$

受影响的状态位: 无

机器码: 

1110	0110	nnnn	nnnn
------	------	------	------

说明: 如果负标志位为 1, 那么程序将跳转。

二进制补码 2n 与 PC 相加。因为 PC 要先递增以便取出下一条指令, 所以新地址都将是  $PC + 2 + 2n$ 。这种情况下, 该指令是一条双周期指令。

指令字数: 1

指令周期数: 1(2)

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例: HERE BN Jump

执行指令前

PC = 地址 (HERE)

执行指令后

如果负标志位 = 1;

PC = 地址 (Jump)

如果负标志位 = 0;

PC = 地址 (HERE + 2)

BNC	无进位则跳转				
语法:	BNC n				
操作数:	-128 ≤ n ≤ 127				
操作:	如果进位标志位为 0, (PC) + 2 + 2n → PC				
受影响的状态位:	无				
机器码:	<table><tr><td>1110</td><td>0011</td><td>nnnn</td><td>nnnn</td></tr></table>	1110	0011	nnnn	nnnn
1110	0011	nnnn	nnnn		
说明:	如果进位标志位为 0, 那么程序将跳转。  二进制补码 2n 与 PC 相加。因为 PC 要先递增以便取出下一条指令, 所以新地址都将为 PC + 2 + 2n。这种情况下, 该指令是一条双周期指令。				
指令字数:	1				
指令周期数:	1(2)				
Q 周期操作:					
如果跳转:					

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例:                    HERE            BNC    Jump

执行指令前  
PC                    =    地址 (HERE)

执行指令后  
如果进位标志位 = 0 ;  
PC                    =    地址 (Jump)  
如果进位标志位 = 1 ;  
PC                    =    地址 (HERE + 2)

BNN	不为负则跳转				
语法:	BNN n				
操作数:	$-128 \leq n \leq 127$				
操作:	如果负标志位为 0, $(PC) + 2 + 2n \rightarrow PC$				
受影响的状态位:	无				
机器码:	<table><tr><td>1110</td><td>0111</td><td>nnnn</td><td>nnnn</td></tr></table>	1110	0111	nnnn	nnnn
1110	0111	nnnn	nnnn		
说明:	如果负标志位为 0, 那么程序将跳转。  二进制补码 $2n$ 与 $PC$ 相加。因为 $PC$ 要先递增以便取出下一条指令, 所以新地址都将为 $PC + 2 + 2n$ 。这种情况下, 该指令是一条双周期指令。				
指令字数:	1				
指令周期数:	1(2)				
Q 周期操作:					
如果跳转:					

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例:                    HERE            BNN    Jump

执行指令前  
PC                    =    地址 (HERE)

执行指令后  
如果负标志位 = 0 ;  
PC                    =    地址 (Jump)  
如果负标志位 = 1 ;  
PC                    =    地址 (HERE + 2)

# PIC18F85J11 系列

## BNOV 不溢出则跳转

语法: BNOV n

操作数:  $-128 \leq n \leq 127$

操作: 如果溢出标志位为 0,  
(PC) + 2 + 2n → PC

受影响的状态位: 无

机器码: 

1110	0101	nnnn	nnnn
------	------	------	------

说明: 如果溢出标志位为 0, 那么程序将跳转。  
二进制补码 2n 与 PC 相加。因为 PC 要先递增以便取出下一条指令, 所以新地址都将为 PC + 2 + 2n。这种情况下, 该指令是一条双周期指令。

指令字数: 1

指令周期数: 1(2)

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例:                    HERE            BNOV   Jump

执行指令前

PC                    =    地址 (HERE)

执行指令后

如果溢出标志位= 0;

PC                    =    地址 (Jump)

如果溢出标志位= 1;

PC                    =    地址 (HERE + 2)

## BNZ 不为零则跳转

语法: BNZ n

操作数:  $-128 \leq n \leq 127$

操作: 如果全零标志位为 0,  
(PC) + 2 + 2n → PC

受影响的状态位: 无

机器码: 

1110	0001	nnnn	nnnn
------	------	------	------

说明: 如果全零标志位为 0, 那么程序将跳转。  
二进制补码 2n 与 PC 相加。因为 PC 要先递增以便取出下一条指令, 所以新地址都将为 PC + 2 + 2n。这种情况下, 该指令是一条双周期指令。

指令字数: 1

指令周期数: 1(2)

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例:                    HERE            BNZ    Jump

执行指令前

PC                    =    地址 (HERE)

执行指令后

如果全零标志位= 0;

PC                    =    地址 (Jump)

如果全零标志位= 1;

PC                    =    地址 (HERE + 2)

**BRA**

无条件跳转

语法:

BRA n

操作数:

-1024 ≤ n ≤ 1023

操作:

(PC) + 2 + 2n → PC

受影响的状态位:

无

机器码:

1101	0nnn	nnnn	nnnn
------	------	------	------

说明:

将二进制补码 2n 与 PC 相加。因为 PC 要先递增才能取下一条指令，所以新地址都将为 PC + 2 + 2n。该指令是一条双周期指令。

指令字数:

1

指令周期数:

2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

示例:

执行指令前

PC

=

地址 (HERE)

执行指令后

PC

=

地址 (Jump)

**BSF**

将 f 中的某位置 1

语法:

BSF f, b {,a}

操作数:

0 ≤ f ≤ 255  
0 ≤ b ≤ 7  
a ∈ [0, 1]

操作:

1 → f<b>

受影响的状态位:

无

机器码:

1000	bbba	ffff	ffff
------	------	------	------

说明:

将寄存器 f 中的位 b 置 1。  
  
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。  
  
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95 (5Fh)，指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例:

执行指令前

FLAG\_REG

=

0Ah

执行指令后

FLAG\_REG

=

8Ah

# PIC18F85J11 系列

## BTFSC 测试寄存器中的位，为 0 则跳过

语法: BTFSC f, b {,a}

操作数:  $0 \leq f \leq 255$   
 $0 \leq b \leq 7$   
 $a \in [0, 1]$

操作: 如果  $(f < b) = 0$  则跳过

受影响的状态位: 无

机器码: 

1011	bbba	ffff	ffff
------	------	------	------

说明: 如果寄存器 f 中的位 b 为 0，则跳过下一条指令。即在位 b 为 0 时，丢弃下一条指令（执行当前指令期间取的指令）而执行一条 NOP 指令，使该指令变成双周期指令。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要  $f \leq 95$  (5Fh)，指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1(2)  
注: 如果跳过的指令后面跟有双字指令，则执行该指令需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例: 

HERE	BTFSC	FLAG, 1, 0
FALSE	:	
TRUE	:	

执行指令前

PC = 地址 (HERE)

执行指令后

如果  $FLAG < 1 = 0$ :

PC = 地址 (TRUE)

如果  $FLAG < 1 = 1$ :

PC = 地址 (FALSE)

## BTFSS 测试寄存器中的位，为 1 则跳过

语法: BTFSS f, b {,a}

操作数:  $0 \leq f \leq 255$   
 $0 \leq b < 7$   
 $a \in [0, 1]$

操作: 如果  $(f < b) = 1$  则跳过

受影响的状态位: 无

机器码: 

1010	bbba	ffff	ffff
------	------	------	------

说明: 如果寄存器 f 中的位 b 为 1，则跳过下一条指令。即在位 b 为 1 时，丢弃下一条指令（执行当前指令期间取的指令）而执行一条 NOP 指令，使该指令变成双周期指令。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要  $f \leq 95$  (5Fh)，指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1(2)  
注: 如果跳过的指令后面跟有双字指令，则执行该指令需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例: 

HERE	BTFSS	FLAG, 1, 0
FALSE	:	
TRUE	:	

执行指令前

PC = 地址 (HERE)

执行指令后

如果  $FLAG < 1 = 0$ :

PC = 地址 (FALSE)

如果  $FLAG < 1 = 1$ :

PC = 地址 (TRUE)



BTG

将 f 中的某位取反

语法:

BTG f, b {,a}

操作数:

0 ≤ f ≤ 255  
0 ≤ b < 7  
a ∈ [0, 1]

操作:

( $\bar{f} < b >$ ) → f < b >

受影响的状态位:

无

机器码:

0111	bbba	ffff	ffff
------	------	------	------

说明:

将数据存储单元 f 中的位 b 取反。  
  
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。  
  
如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例: BTG PORTC, 4, 0

执行指令前:  
PORTC = 0111 0101 [75h]

执行指令后:  
PORTC = 0110 0101 [65h]

BOV

溢出则跳转

语法:

BOV n

操作数:

-128 ≤ n ≤ 127

操作:

如果溢出标志位为 1，  
(PC) + 2 + 2n → PC

受影响的状态位:

无

机器码:

1110	0100	nnnn	nnnn
------	------	------	------

说明:

如果溢出标志位为 1，那么程序将跳转。  
  
二进制补码 2n 与 PC 相加。因为 PC 要先递增以便取出下一条指令，所以新地址都将为 PC + 2 + 2n。这种情况下，该指令是一条双周期指令。

指令字数:

1

指令周期数:

1(2)

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例: HERE BOV Jump

执行指令前  
PC = 地址 (HERE)

执行指令后  
如果溢出标志位 = 1 ;  
PC = 地址 (Jump)  
如果溢出标志位 = 0 ;  
PC = 地址 (HERE + 2)

# PIC18F85J11 系列

## BZ 为零则跳转

语法: BZ n

操作数:  $-128 \leq n \leq 127$

操作: 如果全零标志位为 1,  
(PC) + 2 + 2n → PC

受影响的状态位: 无

机器码:

1110	0000	nnnn	nnnn
------	------	------	------

说明: 如果全零标志位为 1, 那么程序将跳转。  
二进制补码 2n 与 PC 相加。因为 PC 要先递增以便取出下一条指令, 所以新地址都将为 PC + 2 + 2n。这种情况下, 该指令是一条双周期指令。

指令字数: 1

指令周期数: 1(2)

Q 周期操作:

如果跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写入 PC
空操作	空操作	空操作	空操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	空操作

示例:                    HERE            BZ    Jump

执行指令前  
PC = 地址 (HERE)  
执行指令后  
如果全零标志位= 1;  
PC = 地址 (Jump)  
如果全零标志位= 0;  
PC = 地址 (HERE + 2)

## CALL 调用子程序

语法: CALL k {,s}

操作数:  $0 \leq k \leq 1048575$   
 $s \in [0, 1]$

操作: (PC) + 4 → TOS,  
k → PC<20:1>;  
如果 s = 1,  
(W) → WS,  
(STATUS) → STATUSS,  
(BSR) → BSRS

受影响的状态位: 无

机器码:

第一个字 (k<7:0>)	1110	110s	k <sub>7</sub> kkk	kkkk <sub>0</sub>
第二个字 (k<19:8>)	1111	k <sub>19</sub> kkk	kkkk	kkkk <sub>8</sub>

说明: 可在整个 2 MB 的存储器范围内进行子程序调用。首先, 将返回地址 (PC+4) 压入返回堆栈。如果 s = 1, 还会将 W、STATUS 和 BSR 寄存器的内容存入它们各自的影子寄存器、WS、STATUSS 和 BSRS。如果 s = 0, 将不会进行任何更新。然后将 k 的 20 位值装入 PC<20:1>。CALL 是一条双周期指令。

指令字数: 2

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k<7:0>	将 PC 压入堆栈	读立即数 k<19:8>, 写入 PC
空操作	空操作	空操作	空操作

示例:                    HERE            CALL    THERE, 1

执行指令前  
PC = 地址 (HERE)  
执行指令后  
PC = 地址 (HERE)  
TOS = 地址 (HERE + 4)  
WS = W  
BSRS = BSR  
STATUSS = STATUS

CLRf

将 f 清零

语法:

CLRf f{,a}

操作数:

0 ≤ f ≤ 255  
a ∈ [0, 1]

操作:

000h → f,  
1 → Z

受影响的状态位:

Z

机器码:

0110	101a	ffff	ffff
------	------	------	------

说明:

清零指定寄存器的内容。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例: CLRf FLAG\_REG, 1

执行指令前  
FLAG\_REG = 5Ah

执行指令后  
FLAG\_REG = 00h

CLRWDt

将看门狗定时器清零

语法:

CLRWDt

操作数:

无

操作:

000h → WDT,  
000h → WDT 后分频器,  
1 →  $\overline{TO}$ ,  
1 → PD

受影响的状态位:

$\overline{TO}$  和  $\overline{PD}$

机器码:

0000	0000	0000	0100
------	------	------	------

说明:

CLRWDt 指令复位看门狗定时器。而且还会复位 WDT 的后分频器。状态位  $\overline{TO}$  和  $\overline{PD}$  被置 1。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	处理数据	空操作

示例: CLRWDt

执行指令前  
WDT 计数器 = ?

执行指令后  
WDT 计数器 = 00h  
WDT 后分频器 = 0  
 $\overline{TO}$  = 1  
 $\overline{PD}$  = 1

# PIC18F85J11 系列

COMF

将 f 取反

语法:

COMF f {,d {,a}}

操作数:

$0 \leq f \leq 255$   
 $d \in [0, 1]$   
 $a \in [0, 1]$

操作:

$\bar{f} \rightarrow \text{dest}$

受影响的状态位:

N 和 Z

机器码:

0001	11da	ffff	ffff
------	------	------	------

说明:

将寄存器 f 的内容取反。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要  $f \leq 95$ （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例:

COMF	REG, 0, 0
执行指令前 REG	= 13h
执行指令后 REG	= 13h
W	= ECh

CPFSEQ		比较 f 和 W，如果 f = W 则跳过							
语法:	CPFSEQ f {,a}								
操作数:	$0 \leq f \leq 255$ $a \in [0, 1]$								
操作:	$(f) - (W)$ ， 如果 $(f) = (W)$ 则跳过 (无符号比较)								
受影响的状态位:	无								
机器码:	<table border="1"><tr><td>0110</td><td>001a</td><td>ffff</td><td>ffff</td></tr></table>					0110	001a	ffff	ffff
0110	001a	ffff	ffff						
说明:	通过执行无符号减法，将数据存储单元 f 的内容与 W 的内容做比较。  如果 $f = W$ ，则所取的指令被丢弃，转而执行一条 NOP 指令，从而使该指令变成双周期指令。  如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。  如果 a 为 0 且使能了扩展指令集，只要 $f \leq 95$ (5Fh)，指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。								
指令字数:	1								
指令周期数:	1(2)								
注:	如果跳过的指令后面跟有双字指令，则执行该指令需要 3 个周期。								

示例:

HERE	CPFSEQ REG, 0
NEQUAL	:
EQUAL	:

执行指令前

PC 地址	=	HERE
W	=	?
REG	=	?

执行指令后

如果 REG	=	W;
PC	=	地址 (EQUAL)
如果 REG	≠	W;
PC	=	地址 (NEQUAL)

CPFSGT		比较 f 和 W，如果 f > W 则跳过							
语法:	CPFSGT f {,a}								
操作数:	0 ≤ f ≤ 255 a ∈ [0, 1]								
操作:	(f) − (W), 如果 (f) > (W) 则跳过 (无符号比较)								
受影响的状态位:	无								
机器码:	<table border="1"><tr><td>0110</td><td>010a</td><td>ffff</td><td>ffff</td></tr></table>					0110	010a	ffff	ffff
0110	010a	ffff	ffff						
说明:	通过执行无符号减法，将数据存储单元 f 的内容与 W 的内容做比较。  如果 f 的内容大于 WREG 的内容，则所取的指令会被丢弃，转而执行一条 NOP，从而使该指令变成双周期指令。  如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。  如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。								
指令字数:	1								
指令周期数:	1(2)								
注:	如果跳过的指令后面跟有双字指令，则执行该指令需要 3 个周期。								

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

**示例:**

```

HERE      CPFSGT REG, 0
NGREATER  :
GREATER   :
```

执行指令前

```

PC      = 地址 (HERE)
W       = ?
```

执行指令后

```

如果 REG > W ;
PC      = 地址 (GREATER)
如果 REG ≤ W ;
PC      = 地址 (NGREATER)
```

CPFSLT		比较 f 和 W，如果 f < W 则跳过							
语法:	CPFSLT f {,a}								
操作数:	$0 \leq f \leq 255$ $a \in [0, 1]$								
操作:	(f) - (W), 如果 (f) < (W) 则跳过 (无符号比较)								
受影响的状态位:	无								
机器码:	<table border="1"><tr><td>0110</td><td>000a</td><td>ffff</td><td>ffff</td></tr></table>					0110	000a	ffff	ffff
0110	000a	ffff	ffff						
说明:	<p>通过执行无符号减法，将数据存储单元 f 的内容与 W 的内容做比较。</p> <p>如果 f 的内容小于 W 的内容，则所取指令会被丢弃，转而执行一条 NOP，使该指令变成双周期指令。</p> <p>如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。</p>								
指令字数:	1								
指令周期数:	1(2)								
注:	如果跳过的指令后面跟有双字指令，则执行该指令需要 3 个周期。								

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	空操作

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

**示例:**

```

HERE      CPFSLT REG, 1
NLESS    :
LESS     :
```

执行指令前

```

PC      = 地址 (HERE)
W       = ?
```

执行指令后

```

如果 REG < W ;
PC      = 地址 (LESS)
如果 REG ≥ W ;
PC      = 地址 (NLESS)
```

# PIC18F85J11 系列

## DAW 对 W 寄存器进行十进制调整

语法: DAW

操作数: 无

操作: 如果  $[W<3:0> > 9]$  或  $[DC = 1]$ , 那么  $(W<3:0>) + 6 \rightarrow W<3:0>$ ; 否则  $(W<3:0>) \rightarrow W<3:0>$

如果  $[W<7:4> > 9]$  或  $[C = 1]$ , 那么  $(W<7:4>) + 6 \rightarrow W<7:4>$ ;  $C = 1$ ; 否则  $(W<7:4>) \rightarrow W<7:4>$

受影响的状态位: C

机器码: 

0000	0000	0000	0111
------	------	------	------

说明: DAW 指令调整 W 内的 8 位数值, 即前两个压缩 BCD 格式的变量之和, 并产生一个正确的压缩 BCD 格式结果。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 W	处理数据	写 W

### 示例 1: DAW

执行指令前

W = A5h

C = 0

DC = 0

执行指令后

W = 05h

C = 1

DC = 0

### 示例 2:

执行指令前

W = CEh

C = 0

DC = 0

执行指令后

W = 34h

C = 1

DC = 0

## DECF f 减 1

语法: DECF f {,d {,a}}

操作数:  $0 \leq f \leq 255$   
 $d \in [0, 1]$   
 $a \in [0, 1]$

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

受影响的状态位: C、DC、N、OV 和 Z

机器码: 

0000	01da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的内容减 1。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f。

如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集, 只要  $f \leq 95$  (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

### 示例: DECF CNT, 1, 0

执行指令前

CNT = 01h

Z = 0

执行指令后

CNT = 00h

Z = 1

## DECFSZ f 减 1, 为 0 则跳过

语法:	DECFSZ f {,d {,a}}				
操作数:	$0 \leq f \leq 255$ $d \in [0, 1]$ $a \in [0, 1]$				
操作:	$(f) - 1 \rightarrow \text{dest}$ . 结果为 0 时跳过				
受影响的状态位:	无				
机器码:	<table border="1"><tr><td>0010</td><td>11da</td><td>ffff</td><td>ffff</td></tr></table>	0010	11da	ffff	ffff
0010	11da	ffff	ffff		
说明:	<p>将寄存器 <math>f</math> 的内容减 1。如果 <math>d</math> 为 0，结果存储在 <math>W</math> 中。如果 <math>d</math> 为 1，结果存回寄存器 <math>f</math>。</p> <p>如果结果为 0 时，则丢弃已取的下一条指令，转而执行一条 NOP 指令，使该指令变成双周期指令。</p> <p>如果 <math>a</math> 为 0，选择快速操作存储区。如果 <math>a</math> 为 1，使用 BSR 选择 GPR 存储区。</p> <p>如果 <math>a</math> 为 0 且使能了扩展指令集，只要 <math>f \leq 95</math> (5Fh)，指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。</p>				
指令字数:	1				
指令周期数:	1(2)				
注:	如果跳过的指令后面跟有双字指令，则执行该指令需要 3 个周期。				

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例:                   HERE           DECFSZ   CNT, 1, 1  
                                  GOTO       LOOP  
                                  CONTINUE

执行指令前  
PC = 地址 (HERE)  
执行指令后  
CNT = CNT - 1  
如果 CNT = 0;  
PC = 地址 (CONTINUE)  
如果 CNT ≠ 0;  
PC = 地址 (HERE + 2)

## DCFSNZ f 减 1, 非 0 则跳过

语法:	DCFSNZ f {,d {,a}}				
操作数:	$0 \leq f \leq 255$ $d \in [0, 1]$ $a \in [0, 1]$				
操作:	$(f) - 1 \rightarrow \text{dest}$ , 结果为 $\neq 0$ 时跳过				
受影响的状态位:	无				
机器码:	<table border="1"><tr><td>0100</td><td>11da</td><td>ffff</td><td>ffff</td></tr></table>	0100	11da	ffff	ffff
0100	11da	ffff	ffff		
说明:	<p>将寄存器 <b>f</b> 的内容减 1。如果 <b>d</b> 为 0，结果存储在 <b>W</b> 中。如果 <b>d</b> 为 1，结果存回寄存器 <b>f</b>。</p> <p>如果结果不为 0，则丢弃已取的下一条指令，转而执行一条 NOP，使该指令变成双周期指令。</p> <p>如果 <b>a</b> 为 0，选择快速操作存储区。如果 <b>a</b> 为 1，使用 BSR 选择 GPR 存储区。</p> <p>如果 <b>a</b> 为 0 且使能了扩展指令集，只要 <math>f \leq 95</math> (5Fh)，指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。</p>				
指令字数:	1				
指令周期数:	1(2) <b>注:</b> 如果跳过的指令后面跟有双字指令，则执行该指令需要 3 个周期。				

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例:                   HERE       DCFSNZ   TEMP, 1, 0  
                                  ZERO       :  
                                  NZERO       :

执行指令前  
TEMP = ?  
执行指令后  
TEMP = TEMP - 1  
如果 TEMP = 0;  
PC = 地址 (ZERO)  
如果 TEMP ≠ 0;  
PC = 地址 (NZERO)

# PIC18F85J11 系列

## GOTO 无条件跳转

语法: GOTO k

操作数:  $0 \leq k \leq 1048575$

操作:  $k \rightarrow PC<20:1>$

受影响的状态位: 无

机器码:

第一个字 ( $k<7:0>$ )

1110

1111

$k_7kkk$

$kkkk_0$

第二个字 ( $k<19:8>$ )

1111

$k_{19}kkk$

$kkkk$

$kkkk_8$

说明: GOTO指令允许无条件跳转到整个2 MB存储器范围中的任何位置。将k的20位值装入PC<20:1>。GOTO始终为双周期指令。

指令字数: 2

指令周期数: 2

Q周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 $k<7:0>$	空操作	读立即数 $k<19:8>$ , 写入PC
空操作	空操作	空操作	空操作

示例: GOTO THERE

执行指令后

PC = 地址 (THERE)

## INCF f加1

语法: INCF f {,d {,a}}

操作数:  $0 \leq f \leq 255$

$d \in [0, 1]$

$a \in [0, 1]$

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

受影响的状态位: C、DC、N、OV和Z

机器码:

0010

10da

ffff

ffff

说明: 将寄存器f的内容加1。如果d为0, 结果存储在W中。如果d为1, 结果存回寄存器f。

如果a为0, 选择快速操作存储区。如果a为1, 使用BSR选择GPR存储区。

如果a为0且使能了扩展指令集, 只要 $f \leq 95$  (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第25.2.3节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1

Q周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器f	处理数据	写入目标寄存器

示例: INCF CNT, 1, 0

执行指令前

CNT = FFh  
Z = 0  
C = ?  
DC = ?

执行指令后

CNT = 00h  
Z = 1  
C = 1  
DC = 1



## INCFSZ f 加 1, 为 0 则跳过

语法: INCFSZ f {,d {,a}}

操作数:  $0 \leq f \leq 255$   
 $d \in [0, 1]$   
 $a \in [0, 1]$

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

受影响的状态位: 无

机器码: 

0011	11da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的内容加 1。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f。

如果结果为 0 时, 则丢弃已取的下一条指令, 转而执行一条 NOP 指令, 使该指令变成双周期指令。

如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集, 只要  $f \leq 95$  (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1(2)  
 注: 如果跳过的指令后面跟有双字指令, 则执行该指令需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例: 

HERE	INCFSZ	CNT, 1, 0
NZERO	:	
ZERO	:	

执行指令前  
 PC = 地址 (HERE)

执行指令后  
 CNT = CNT + 1  
 如果 CNT = 0;  
 PC = 地址 (ZERO)  
 如果 CNT ≠ 0;  
 PC = 地址 (NZERO)

## INFSNZ f 加 1, 非 0 则跳过

语法: INFSNZ f {,d {,a}}

操作数:  $0 \leq f \leq 255$   
 $d \in [0, 1]$   
 $a \in [0, 1]$

操作:  $(f) + 1 \rightarrow \text{dest}$ ,  
 结果为 ≠ 0 时跳过

受影响的状态位: 无

机器码: 

0100	10da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的内容加 1。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f。

如果结果不为 0, 则丢弃已取的下一条指令, 转而执行一条 NOP, 使该指令变成双周期指令。

如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展的指令集, 只要  $f \leq 95$  (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1(2)  
 注: 如果跳过的指令后面跟有双字指令, 则执行该指令需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例: 

HERE	INFSNZ	REG, 1, 0
ZERO	:	
NZERO	:	

执行指令前  
 PC = 地址 (HERE)

执行指令后  
 REG = REG + 1  
 如果 REG ≠ 0;  
 PC = 地址 (NZERO)  
 如果 REG = 0;  
 PC = 地址 (ZERO)

# PIC18F85J11 系列

## IORLW 立即数与 W 作逻辑或运算

语法: IORLW k

操作数:  $0 \leq k \leq 255$

操作: (W) .OR. k  $\rightarrow$  W

受影响的状态位: N 和 Z

机器码:

0000	1001	kkkk	kkkk
------	------	------	------

说明: W 的内容与 8 位立即数 k 作逻辑或运算。结果保存在 W 寄存器中。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 立即数 k	处理数据	写入 W

示例: IORLW 35h

执行指令前  
W = 9Ah

执行指令后  
W = BFh

## IORWF 将 W 与 f 作逻辑或运算

语法: IORWF f {,d {,a}}

操作数:  $0 \leq f \leq 255$

$d \in [0, 1]$

$a \in [0, 1]$

操作: (W) .OR. (f)  $\rightarrow$  dest

受影响的状态位: N 和 Z

机器码:

0001	00da	ffff	ffff
------	------	------	------

说明: 将 W 的内容与寄存器 f 的内容进行逻辑或运算。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要  $f \leq 95$  (5Fh)，指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

示例: IORWF RESULT, 0, 1

执行指令前  
RESULT = 13h  
W = 91h

执行指令后  
RESULT = 13h  
W = 93h

LFSR

装入 FSR

语法:

LFSR f, k

操作数:

0 ≤ f ≤ 2  
0 ≤ k ≤ 4095

操作:

k → FSRf

受影响的状态位:

无

机器码:

1110	1110	00ff	k <sub>11</sub> kkk
1111	0000	k <sub>7</sub> kkk	kkkk

说明:

将 12 位的立即数 k 装入由 f 指向的文件选择寄存器。

指令字数:

2

指令周期数:

2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k 的高字节	处理数据	将立即数 k 的高字节写入 FSRfH
译码	读立即数 k 的低字节	处理数据	将立即数 k 写入 FSRfL

示例:

LFSR 2, 3ABh

执行指令后

FSR2H	=	03h
FSR2L	=	ABh

MOVF

移动 f

语法:

MOVF f {,d {,a}}

操作数:

0 ≤ f ≤ 255  
d ∈ [0, 1]  
a ∈ [0, 1]

操作:

f → dest

受影响的状态位:

N 和 Z

机器码:

0101	00da	ffff	ffff
------	------	------	------

说明:

根据 d 的状态，将寄存器 f 的内容移入目标单元。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f。f 可以为 256 字节存储区中的任何单元。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写 W

示例:

MOVF REG, 0, 0

执行指令前

REG	=	22h
W	=	FFh

执行指令后

REG	=	22h
W	=	22h

# PIC18F85J11 系列

## MOVFF 将源寄存器的内容移入目标寄存器

语法: MOVFF  $f_s, f_d$

操作数:  $0 \leq f_s \leq 4095$   
 $0 \leq f_d \leq 4095$

操作:  $(f_s) \rightarrow f_d$

受影响的状态位: 无

机器码:				
第一个字 (源)	1100	ffff	ffff	ffff $f_s$
第二个字 (目标)	1111	ffff	ffff	ffff $f_d$

说明: 将源寄存器  $f_s$  的内容移入目标寄存器  $f_d$ 。源寄存器  $f_s$  可以是 4096 字节数据空间 (000h 到 FFFh) 中的任何单元, 目标寄存器  $f_d$  也可以是 000h 到 FFFh 中的任何单元。

源或目标寄存器都可以是 W (这是个有用的特例)。

MOVFF 指令对于将数据存储单元中的内容移入外设寄存器 (如发送缓冲器或 I/O 端口) 的场合非常有用。

MOVFF 指令不能使用 PCL、TOSU、TOSH 或 TOSL 作为目标寄存器。

指令字数: 2

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 $f$ (源寄存器)	处理数据	空操作
译码	空操作 非无效读取	空操作	写寄存器 $f$ (目标寄存器)

示例: MOVFF REG1, REG2

执行指令前  
REG1 = 33h  
REG2 = 11h

执行指令后  
REG1 = 33h  
REG2 = 33h

## MOVLB 将立即数移入 BSR 的低半字节

语法: MOVLW  $k$

操作数:  $0 \leq k \leq 255$

操作:  $k \rightarrow \text{BSR}$

受影响的状态位: 无

机器码:	0000	0001	kkkk	kkkk
------	------	------	------	------

说明: 将 8 位立即数  $k$  装入存储区选择寄存器 (BSR)。BSR<7:4> 的值将始终保持为 0, 不管  $k_7:k_4$  的值如何都如此。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 $k$	处理数据	将立即数 $k$ 写入 BSR

示例: MOVLB 5

执行指令前  
BSR 寄存器 = 02h  
执行指令后  
BSR 寄存器 = 05h

**MOVLW**

将立即数移入 W

语法:

MOVLW k

操作数:

$0 \leq k \leq 255$

操作:

$k \rightarrow W$

受影响的状态位:

无

机器码:

0000	1110	kkkk	kkkk
------	------	------	------

说明:

将 8 位立即数 k 装入 W。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 W

示例:                   MOVLW       5Ah

      执行指令后  
      W       =   5Ah

**MOVWF**

将 W 的内容移入 f

语法:

MOVWF f {,a}

操作数:

$0 \leq f \leq 255$   
 $a \in [0, 1]$

操作:

$(W) \rightarrow f$

受影响的状态位:

无

机器码:

0110	111a	ffff	ffff
------	------	------	------

说明:

将 W 寄存器中的数据移入寄存器 f。  
f 可以为 256 字节存储区中的任何单元。  
  
如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。  
  
如果 a 为 0 且使能了扩展指令集，只要  $f \leq 95$ （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例:                   MOVWF     REG, 0

      执行指令前  
      W       =   4Fh  
      REG     =   FFh

      执行指令后  
      W       =   4Fh  
      REG     =   4Fh

# PIC18F85J11 系列

## MULLW 将立即数与 W 的内容相乘

语法: MULLW k

操作数:  $0 \leq k \leq 255$

操作:  $(W) \times k \rightarrow \text{PRODH:PRODL}$

受影响的状态位: 无

机器码:

0000	1101	kkkk	kkkk
------	------	------	------

说明: 将 W 的内容和 8 位立即数 k 进行无符号的乘法运算。16 位的结果存储在 PRODH:PROD 寄存器对中。其中 PRODH 存储高字节。

W 的内容不变。

所有状态标志位都不受影响。

请注意此操作不可能发生溢出或进位。结果有可能为零，但不会反映到相应的标志位。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写寄存器 PRODH: PRODL

示例: MULLW 0C4h

执行指令前

W	=	E2h
PRODH	=	?
PRODL	=	?

执行指令后

W	=	E2h
PRODH	=	ADh
PRODL	=	08h

## MULWF 将 W 与 f 的内容相乘

语法: MULWF f {,a}

操作数:  $0 \leq f \leq 255$

$a \in [0, 1]$

操作:  $(W) \times (f) \rightarrow \text{PRODH:PRODL}$

受影响的状态位: 无

机器码:

0000	001a	ffff	ffff
------	------	------	------

说明: 将 W 的内容和寄存器单元 f 的内容执行无符号的乘法运算。运算的 16 位结果存储在 PRODH:PRODL 寄存器对中。其中 PRODH 存储高字节。W 和 f 的内容都不变。

所有状态标志位都不受影响。

请注意此操作不可能发生溢出或进位。结果有可能为零，但不会反映到相应的标志位。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要  $f \leq 95$  (5Fh)，指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 PRODH: PRODL

示例: MULWF REG, 1

执行指令前

W	=	C4h
REG	=	B5h
PRODH	=	?
PRODL	=	?

执行指令后

W	=	C4h
REG	=	B5h
PRODH	=	8Ah
PRODL	=	94h

NEGf

对 f 取补

语法:

NEGf f {,a}

操作数:

0 ≤ f ≤ 255  
a ∈ [0, 1]

操作:

$(\bar{f}) + 1 \rightarrow f$

受影响的状态位:

N、OV、C、DC 和 Z

机器码:

0110	110a	ffff	ffff
------	------	------	------

说明:

用二进制补码对单元 f 取补。结果存储在数据存储单元 f 中。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例:

	NEGf	REG, 1
执行指令前	REG	= 0011 1010 [3Ah]
执行指令后	REG	= 1100 0110 [C6h]

NOP

空操作

语法:

NOP

操作数:

无

操作:

空操作

受影响的状态位:

无

机器码:

0000	0000	0000	0000
1111	xxxx	xxxx	xxxx

说明:

空操作。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	空操作	空操作

示例:

无。

# PIC18F85J11 系列

## POP 弹出返回堆栈栈顶的内容

语法:	POP			
操作数:	无			
操作:	(TOS) → 丢弃			
受影响的状态位:	无			
机器码:	0000	0000	0000	0110
说明:	从返回堆栈弹出 <b>TOS</b> 值并丢弃。然后，前一个压入返回堆栈的值成为 <b>TOS</b> 值。此指令可以让用户正确管理返回堆栈，从而实现软件堆栈。			

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	弹出 TOS 值	空操作

示例: POP  
GOTO NEW

执行指令前		
TOS	=	0031A2h
堆栈 (下一级)	=	014332h

执行指令后		
TOS	=	014332h
PC	=	新值

## PUSH 将数据压入返回堆栈栈顶

语法:	PUSH				
操作数:	无				
操作:	(PC + 2) → TOS				
受影响的状态位:	无				
机器码:	<table border="1"><tr><td>0000</td><td>0000</td><td>0000</td><td>0101</td></tr></table>	0000	0000	0000	0101
0000	0000	0000	0101		
说明:	PC + 2 的值被压入返回堆栈的栈顶。原先的 TOS 值被压入堆栈的下一级。 此指令允许通过修改 TOS 并将其压入返回堆栈来实现软件堆栈。				

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	将 PC + 2 压入返回堆栈	空操作	空操作

示例: PUSH

执行指令前		
TOS	=	345Ah
PC	=	0124h

执行指令后		
PC	=	0126h
TOS	=	0126h
堆栈 (下一级)	=	345Ah



RCALL

相对调用

语法:

RCALL n

操作数:

-1024 ≤ n ≤ 1023

操作:

(PC) + 2 → TOS,  
(PC) + 2 + 2n → PC

受影响的状态位:

无

机器码:

1101	1nnn	nnnn	nnnn
------	------	------	------

说明:

从当前地址单元（最多 1KB）来调用子程序。首先，将返回地址（PC + 2）压入返回堆栈。然后，将二进制补码 2n 与 PC 相加。因为 PC 要先递增才能取下一条指令，因此新地址将为 PC + 2 + 2n。该指令是一条双周期指令。

指令字数:

1

指令周期数:

2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 n 将 PC 压入堆栈	处理数据	写入 PC
空操作	空操作	空操作	空操作

示例:                    HERE        RCALL Jump

执行指令前  
PC =    地址 (HERE)

执行指令后  
PC =    地址 (Jump)  
TOS =   地址 (HERE + 2)

RESET

复位

语法:

RESET

操作数:

无

操作:

将所有  $\overline{\text{MCLR}}$  复位影响的寄存器和标志位复位。

受影响的状态位:

全部

机器码:

0000	0000	1111	1111
------	------	------	------

说明:

此指令可实现用软件执行  $\overline{\text{MCLR}}$  复位。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	启动复位	空操作	空操作

示例:

RESET

执行指令后  
寄存器    =    复位值  
标志位 \* =    复位值

# PIC18F85J11 系列

## RETFIE 从中断返回

语法: RETFIE {s}

操作数:  $s \in [0, 1]$

操作: (TOS)  $\rightarrow$  PC,  
1  $\rightarrow$  GIE/GIEH 或 PEIE/GIEL ;  
如果  $s = 1$ ,  
(WS)  $\rightarrow$  W,  
(STATUS)  $\rightarrow$  STATUS 寄存器,  
(BSRS)  $\rightarrow$  BSR,  
PCLATU 和 PCLATH 保持不变

受影响的状态位: GIE/GIEH 和 PEIE/GIEL。

机器码:

0000	0000	0001	000s
------	------	------	------

说明: 从中断返回。执行出栈操作时, 将栈顶 (TOS) 的内容装入 PC。通过将高或低优先级全局中断允许位置 1 来允许中断。如果  $s = 1$ , 则影子寄存器 WS、STATUS 和 BSR 的内容将被装入对应的寄存器 W、STATUS 和 BSR。如果  $s = 0$ , 则不更新这些寄存器。

指令字数: 1

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	空操作	从堆栈弹出 PC 值 将 GIEH 或 GIEL 置 1
空操作	空操作	空操作	空操作

示例: RETFIE 1

中断后  
PC = TOS  
W = WS  
BSR = BSR  
STATUS = STATUS  
GIE/GIEH 和 PEIE/GIEL = 1

## RETLW 返回时将立即数送入 W

语法: RETLW k

操作数:  $0 \leq k \leq 255$

操作:  $k \rightarrow$  W,  
(TOS)  $\rightarrow$  PC,  
PCLATU 和 PCLATH 保持不变

受影响的状态位: 无

机器码:

0000	1100	kkkk	kkkk
------	------	------	------

说明: 将 8 位立即数 k 装入 W。将栈顶内容 (返回地址) 装入程序计数器。高位地址锁存器 (PCLATH) 的内容保持不变。

指令字数: 1

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	从堆栈弹出 PC 值, 写入 W
空操作	空操作	空操作	空操作

示例:

```
CALL TABLE ; W contains table  
              ; offset value  
              ; W now has  
              ; table value
```

```
:  
TABLE  
    ADDWF PCL ; W = offset  
    RETLW k0  ; Begin table  
    RETLW k1  ;  
:  
:  
    RETLW kn  ; End of table
```

执行指令前  
W = 07h

执行指令后  
W = kn 的值

RETURN 从子程序返回

语法: RETURN {s}

操作数:  $s \in [0, 1]$

操作: (TOS) → PC ;  
如果  $s = 1$ ,  
(WS) → W,  
(STATUS) → STATUS 寄存器,  
(BSRS) → BSR,  
PCLATU 和 PCLATH 保持不变

受影响的状态位: 无

机器码: 

0000	0000	0001	001s
------	------	------	------

说明: 从子程序返回。执行出栈操作, 将栈顶 (TOS) 的内容装入程序计数器。如果  $s = 1$ , 将影子寄存器 WS、STAUTSS 和 BSRS 的内容装入相应的 W、STATUS 和 BSR 寄存器。如果  $s = 0$ , 则不更新这些寄存器。

指令字数: 1

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	处理数据	从堆栈弹出 PC 值
空操作	空操作	空操作	空操作

示例: RETURN

执行指令后:  
PC = TOS

RLCF f 带进位循环左移

语法: RLCF f {,d {,a}}

操作数:  $0 \leq f \leq 255$   
 $d \in [0, 1]$   
 $a \in [0, 1]$

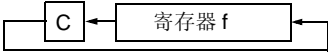
操作: (f<n>) → dest<n + 1>,  
(f<7>) → C,  
(C) → dest<0>

受影响的状态位: C、N 和 Z

机器码: 

0011	01da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的内容连同进位标志位一起循环左移 1 位。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f。  
如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区。  
如果 a 为 0 且使能了扩展指令集, 只要  $f \leq 95$  (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。



指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例: RLCF REG, 0, 0

执行指令前  
REG = 1110 0110  
C = 0

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

# PIC18F85J11 系列

## RLNCF f 循环左移（不带进位）

语法: RLNCF f {,d {,a}}

操作数:  $0 \leq f \leq 255$   
 $d \in [0, 1]$   
 $a \in [0, 1]$

操作:  $(f < n) \rightarrow \text{dest} < n + 1 >$ ,  
 $(f < 7) \rightarrow \text{dest} < 0 >$

受影响的状态位: N 和 Z

机器码: 

0100	01da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的内容循环左移 1 位。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f。

如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集, 只要  $f \leq 95$  (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。



指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例: RLNCF REG, 1, 0

执行指令前  
REG = 1010 1011

执行指令后  
REG = 0101 0111

## RRCF f 带进位循环右移

语法: RRCF f {,d {,a}}

操作数:  $0 \leq f \leq 255$   
 $d \in [0, 1]$   
 $a \in [0, 1]$

操作:  $(f < n) \rightarrow \text{dest} < n - 1 >$ ,  
 $(f < 0) \rightarrow C$ ,  
 $(C) \rightarrow \text{dest} < 7 >$

受影响的状态位: C、N 和 Z

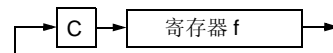
机器码: 

0011	00da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的内容连同进位标志位一起循环右移 1 位。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f。

如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集, 只要  $f \leq 95$  (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。



指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例: RRCF REG, 0, 0

执行指令前  
REG = 1110 0110  
C = 0

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

**RRNCF** **f 循环右移（不带进位）**

语法: RRNCF f {,d {,a}}

操作数:  $0 \leq f \leq 255$   
 $d \in [0, 1]$   
 $a \in [0, 1]$

操作:  $(f < n) \rightarrow \text{dest} < n - 1 >$ ,  
 $(f < 0 >) \rightarrow \text{dest} < 7 >$

受影响的状态位: N 和 Z

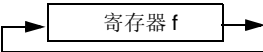
机器码: 

0100	00da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的内容循环右移 1 位。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f。

如果 a 为 0，选择快速操作存储区，覆盖 BSR 的值。如果 a 为 1，使用 BSR 选择存储区。

如果 a 为 0 且使能了扩展指令集，只要  $f \leq 95$  (5Fh)，指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。



指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例 1: RRNCF REG, 1, 0

执行指令前	
REG	= 1101 0111
执行指令后	
REG	= 1110 1011

示例 2: RRNCF REG, 0, 0

执行指令前	
W	= ?
REG	= 1101 0111
执行指令后	
W	= 1110 1011
REG	= 1101 0111

**SETF** **将 f 的内容置为全 1**

语法: SETF f {,a}

操作数:  $0 \leq f \leq 255$   
 $a \in [0, 1]$

操作: FFh  $\rightarrow$  f

受影响的状态位: 无

机器码: 

0110	100a	ffff	ffff
------	------	------	------

说明: 将指定寄存器的内容置为 FFh。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要  $f \leq 95$  (5Fh)，指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例: SETF REG, 1

执行指令前	
REG	= 5Ah
执行指令后	
REG	= FFh

# PIC18F85J11 系列

## SLEEP 进入休眠模式

语法:	SLEEP				
操作数:	无				
操作:	00h → WDT, 0 → WDT 后分频器, 1 → $\overline{TO}$ , 0 → PD				
受影响的状态位:	$\overline{TO}$ 和 $\overline{PD}$				
机器码:	<table border="1"><tr><td>0000</td><td>0000</td><td>0000</td><td>0011</td></tr></table>	0000	0000	0000	0011
0000	0000	0000	0011		
说明:	掉电状态位 ( $\overline{PD}$ ) 清零。超时状态位 ( $\overline{TO}$ ) 置 1。看门狗定时器及其后分频器清零。  振荡器停振，处理器进入休眠模式。				
指令字数:	1				
指令周期数:	1				
Q 周期操作:					

Q1	Q2	Q3	Q4
译码	空操作	处理数据	进入休眠模式

示例: SLEEP

执行指令前  
 $\overline{TO}$  = ?  
 $\overline{PD}$  = ?

执行指令后  
 $\overline{TO}$  = 1†  
 $\overline{PD}$  = 0

† 如果由 WDT 引起唤醒, 则此位将被清零。

## SUBFWB W 减去 f (带借位)

语法:	SUBFWB f {,d {,a}}				
操作数:	$0 \leq f \leq 255$ $d \in [0, 1]$ $a \in [0, 1]$				
操作:	$(W) - (f) - (\overline{C}) \rightarrow \text{dest}$				
受影响的状态位:	N、OV、C、DC 和 Z				
机器码:	<table border="1"><tr><td>0101</td><td>01da</td><td>ffff</td><td>ffff</td></tr></table>	0101	01da	ffff	ffff
0101	01da	ffff	ffff		
说明:	将 W 的内容减去寄存器 f 和进位（借				

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例 1: SUBFWB REG, 1, 0

执行指令前  
REG = 3  
W = 2  
C = 1  
执行指令后  
REG = FF  
W = 2  
C = 0  
Z = 0  
N = 1 ; 结果为负

示例 2: SUBFWB REG, 0, 0

执行指令前  
REG = 2  
W = 5  
C = 1  
执行指令后  
REG = 2  
W = 3  
C = 1  
Z = 0  
N = 0 ; 结果为正

示例 3: SUBFWB REG, 1, 0

执行指令前  
REG = 1  
W = 2  
C = 0  
执行指令后  
REG = 0  
W = 2  
C = 1  
Z = 1  
N = 0 ; 结果为零

## SUBLW 立即数减去 W 的内容

语法: SUBLW k

操作数:  $0 \leq k \leq 255$

操作:  $k - (W) \rightarrow W$

受影响的状态位: N、OV、C、DC 和 Z

机器码:

0000	1000	kkkk	kkkk
------	------	------	------

说明: 用 8 位立即数 k 减去 W。结果保存在 W 寄存器中。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写入 W

示例 1: SUBLW 02h

执行指令前

W	=	01h
C	=	?

执行指令后

W	=	01h
C	=	1 ; 结果为正
Z	=	0
N	=	0

示例 2: SUBLW 02h

执行指令前

W	=	02h
C	=	?

执行指令后

W	=	00h
C	=	1 ; 结果为零
Z	=	1
N	=	0

示例 3: SUBLW 02h

执行指令前

W	=	03h
C	=	?

执行指令后

W	=	FFh ; (二进制补码)
C	=	0 ; 结果为负
Z	=	0
N	=	1

## SUBWF f 减去 W

语法: SUBWF f {,d {,a}}

操作数:  $0 \leq f \leq 255$   
 $d \in [0, 1]$   
 $a \in [0, 1]$

操作:  $(f) - (W) \rightarrow \text{dest}$

受影响的状态位: N、OV、C、DC 和 Z

机器码:

0101	11da	ffff	ffff
------	------	------	------

说明: 用寄存器 f 中的内容减去 W 寄存器的内容 (通过二进制补码方式进行运算)。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f。

如果 a 为 0, 选择快速操作存储区。如果 a 为 1, 使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集, 只要  $f \leq 95$  (5Fh), 指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例 1: SUBWF REG, 1, 0

执行指令前

REG	=	3
W	=	2
C	=	?

执行指令后

REG	=	1
W	=	2
C	=	1 ; 结果为正
Z	=	0
N	=	0

示例 2: SUBWF REG, 0, 0

执行指令前

REG	=	2
W	=	2
C	=	?

执行指令后

REG	=	2
W	=	0
C	=	1 ; 结果为零
Z	=	1
N	=	0

示例 3: SUBWF REG, 1, 0

执行指令前

REG	=	1
W	=	2
C	=	?

执行指令后

REG	=	FFh ; (二进制补码)
W	=	2
C	=	0 ; 结果为负
Z	=	0
N	=	1

# PIC18F85J11 系列

## SUBWFB f 减去 W（带借位）

语法: SUBWFB f {,d {,a}}

操作数:  $0 \leq f \leq 255$

$d \in [0, 1]$

$a \in [0, 1]$

操作:  $(f) - (W) - (\overline{C}) \rightarrow \text{dest}$

受影响的状态位: N、OV、C、DC 和 Z

机器码:

0101	10da	ffff	ffff
------	------	------	------

说明: 用 f 寄存器的内容减去 W 的内容和进位（借位）（通过二进制补码方式进行运算）。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要  $f \leq 95$ （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

示例 1: SUBWFB REG, 1, 0

执行指令前

REG = 19h (0001 1001)  
W = 0Dh (0000 1101)  
C = 1

执行指令后

REG = 0Ch (0000 1011)  
W = 0Dh (0000 1101)  
C = 1  
Z = 0  
N = 0 ; 结果为正

示例 2: SUBWFB REG, 0, 0

执行指令前

REG = 1Bh (0001 1011)  
W = 1Ah (0001 1010)  
C = 0

执行指令后

REG = 1Bh (0001 1011)  
W = 00h  
C = 1  
Z = 1 ; 结果为零  
N = 0

示例 3: SUBWFB REG, 1, 0

执行指令前

REG = 03h (0000 0011)  
W = 0Eh (0000 1101)  
C = 1

执行指令后

REG = F5h (1111 0100)  
; [二进制补码]  
W = 0Eh (0000 1101)  
C = 0  
Z = 0  
N = 1 ; 结果为负

## SWAPF 将 f 的高半字节和低半字节相交换

语法: SWAPF f {,d {,a}}

操作数:  $0 \leq f \leq 255$

$d \in [0, 1]$

$a \in [0, 1]$

操作:  $(f<3:0>) \rightarrow \text{dest}<7:4>$ ,  
 $(f<7:4>) \rightarrow \text{dest}<3:0>$

受影响的状态位: 无

机器码:

0011	10da	ffff	ffff
------	------	------	------

说明: 将寄存器 f 的高半字节和低半字节互相交换。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要  $f \leq 95$ （5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写入 目标寄存器

示例: SWAPF REG, 1, 0

执行指令前

REG = 53h

执行指令后

REG = 35h



## TBLRD 表读

语法: TBLRD (\*; \*+; \*-; +\*)

操作数: 无

操作: 如果执行 TBLRD \*,  
(程序存储器 (TBLPTR)) → TABLAT ;  
TBLPTR 不改变  
如果执行 TBLRD \*+,  
(程序存储器 (TBLPTR)) → TABLAT ;  
(TBLPTR) + 1 → TBLPTR  
如果执行 TBLRD \*-,  
(程序存储器 (TBLPTR)) → TABLAT ;  
(TBLPTR) - 1 → TBLPTR  
如果执行 TBLRD +\*,  
(TBLPTR) + 1 → TBLPTR ;  
(程序存储器 (TBLPTR)) → TABLAT

受影响的状态位: 无

机器码:	0000	0000	0000	10nn nn=0 * =1 *+ =2 *- =3 +*
------	------	------	------	---

说明: 此指令用于读取程序存储器 (P.M.)。要对程序存储器进行寻址, 需要使用表指针 (TBLPTR)。

TBLPTR (一个 21 位指针) 可以指向程序存储器中的每个字节。TBLPTR 的寻址范围为 2 MB。

TBLPTR<0> = 0: 程序存储器字的低有效字节

TBLPTR<0> = 1: 程序存储器字的高有效字节

TBLRD 指令可以按如下方法来修改 TBLPTR 的值:

- 不变
- 后增
- 后减
- 预增

指令字数: 1

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	空操作	空操作	空操作
空操作	空操作 (读程序存储器)	空操作	空操作 (写 TABLAT)

## TBLRD 表读 (续)

示例 1: TBLRD \*+ ;

执行指令前

TABLAT	=	55h
TBLPTR	=	00A356h
存储单元 (00A356h)	=	34h

执行指令后

TABLAT	=	34h
TBLPTR	=	00A357h

示例 2: TBLRD +\* ;

执行指令前

TABLAT	=	AAh
TBLPTR	=	01A357h
存储单元 (01A357h)	=	12h
存储单元 (01A358h)	=	34h

执行指令后

TABLAT	=	34h
TBLPTR	=	01A358h

# PIC18F85J11 系列

TBLWT	表写				
语法:	TBLWT (*; *+; *-; +*)				
操作数:	无				
操作:	如果执行 TBLWT*, (TABLAT) → 保持寄存器; TBLPTR 不改变 如果执行 TBLWT*+*, (TABLAT) → 保持寄存器; (TBLPTR) + 1 → TBLPTR 如果执行 TBLWT*-, (TABLAT) → 保持寄存器; (TBLPTR) - 1 → TBLPTR 如果执行 TBLWT*+*, (TBLPTR) + 1 → TBLPTR ; (TABLAT) → 保持寄存器				
受影响的状态位:	无				
机器码:	<table><tr><td>0000</td><td>0000</td><td>0000</td><td>11nn nn=0 * =1 *+ =2 *- =3 +*</td></tr></table>	0000	0000	0000	11nn nn=0 * =1 *+ =2 *- =3 +*
0000	0000	0000	11nn nn=0 * =1 *+ =2 *- =3 +*		
说明:	<p>此指令使用 TBLPTR 的低 3 个字节来确定要将 TABLAT 写入 8 个保持寄存器中的哪一个。保持寄存器用于对程序存储器 (P.M.) 的内容编程。(关于对闪存程序存储器编程的更多详情, 请参见<b>第 6.0 节“存储器构成”</b>。)</p> <p>TBLPTR (一个 21 位指针) 可以指向程序存储器中的每个字节。TBLPTR 的寻址范围为 2 MB。TBLPTR 的 LSb 选择要访问的程序存储单元字节。</p> <p>TBLPTR&lt;0&gt; = 0: 程序存储器字的低有效字节</p> <p>TBLPTR&lt;0&gt; = 1: 程序存储器字的高有效字节</p> <p>TBLWT 指令可用如下方法来修改 TBLPTR 的值:</p> <ul style="list-style-type: none"><li>• 不变</li><li>• 后增</li><li>• 后减</li><li>• 预增</li></ul>				
指令字数:	1				
指令周期数:	2				
Q 周期操作:					

Q1	Q2	Q3	Q4
译码	空操作	空操作	空操作
空操作	空操作 (读 TABLAT)	空操作	空操作 (写入保持 寄存器)

TBLWT	表写 (续)
示例 1:	TBLWT *+;
执行指令前	
TABLAT	= 55h
TBLPTR	= 00A356h
保持寄存器 (00A356h)	= FFh
执行指令后 (表写操作完成)	
TABLAT	= 55h
TBLPTR	= 00A357h
保持寄存器 (00A356h)	= 55h
示例 2:	TBLWT +*;
执行指令前	
TABLAT	= 34h
TBLPTR	= 01389Ah
保持寄存器 (01389Ah)	= FFh
保持寄存器 (01389Bh)	= FFh
执行指令后 (表写操作完成)	
TABLAT	= 34h
TBLPTR	= 01389Bh
保持寄存器 (01389Ah)	= FFh
保持寄存器 (01389Bh)	= 34h

TSTFSZ

测试 f，为 0 则跳过

语法:

TSTFSZ f {,a}

操作数:

0 ≤ f ≤ 255  
a ∈ [0, 1]

操作:

如果 f = 0 则跳过

受影响的状态位:

无

机器码:

0110	011a	ffff	ffff
------	------	------	------

说明:

如果 f 为 0 时，丢弃下一指令（在当前指令执行期间获取）而执行一条 NOP 指令，使这条指令变成双周期指令。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要 f ≤ 95（5Fh），指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数:

1

指令周期数:

1(2)

注:

如果跳过的指令后面跟有双字指令，则执行该指令需要 3 个周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	空操作

如果跳过:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作

如果跳过的指令后面跟有双字指令:

Q1	Q2	Q3	Q4
空操作	空操作	空操作	空操作
空操作	空操作	空操作	空操作

示例:

HERE     TSTFSZ   CNT, 1

NZERO   :

ZERO    :

执行指令前

PC       =    地址（HERE）

执行指令后

如果 CNT   =   00h,

PC       =    地址（ZERO）

如果 CNT   ≠   00h,

PC       =    地址（NZERO）

XORLW

将立即数与 W 作逻辑异或运算

语法:

XORLW k

操作数:

0 ≤ k ≤ 255

操作:

(W) .XOR. k → W

受影响的状态位:

N 和 Z

机器码:

0000	1010	kkkk	kkkk
------	------	------	------

说明:

将 W 的内容与 8 位立即数 k 作逻辑异或运算。结果保存在 W 寄存器中。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 立即数 k	处理数据	写入 W

示例:

XORLW   0AFh

执行指令前

W       =    B5h

执行指令后

W       =    1Ah

# PIC18F85J11 系列

## XORWF

W 与 f 作逻辑异或运算

语法: XORWF f {,d {,a}}

操作数:  $0 \leq f \leq 255$   
 $d \in [0, 1]$   
 $a \in [0, 1]$

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

受影响的状态位: N 和 Z

机器码:

0001	10da	ffff	ffff
------	------	------	------

说明: W 的内容与寄存器 f 的内容进行逻辑异或运算。如果 d 为 0，结果存储在 W 中。如果 d 为 1，结果存回寄存器 f。

如果 a 为 0，选择快速操作存储区。如果 a 为 1，使用 BSR 选择 GPR 存储区。

如果 a 为 0 且使能了扩展指令集，只要  $f \leq 95$  (5Fh)，指令就将以立即数变址寻址模式进行操作。详情请参见第 25.2.3 节“立即数变址寻址模式中字节和位操作类指令”。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例: XORWF REG, 1, 0

执行指令前  
REG = AFh  
W = B5h

执行指令后  
REG = 1Ah  
W = B5h

25.2 扩展指令集

除了 PIC18 指令集的 75 条标准指令之外，PIC18F85J11 系列器件还提供针对内核 CPU 功能的可选扩展指令。这些新增的功能包括 8 条额外的指令，它们可以实现间接和变址寻址操作，并使得许多标准 PIC18 指令可以实现立即数变址寻址。

默认情况下，这些扩展指令集的额外功能在未编程器件上是使能的。用户必须在编程过程中将 XINST 配置位置 1 或清零来使能或禁止这些功能。

扩展指令集中的指令可以全部被归为立即数操作类指令，它们既可以控制文件选择寄存器，也可以使用这些寄存器进行变址寻址。其中两条指令 ADDFSR 和 SUBFSR 的一种特殊用法是直接对 FSR2 进行操作，而 ADDULNK 和 SUBULNK 指令允许在执行后自动返回。

这些扩展指令专门用于优化用高级语言，特别是 C 语言编写的重入程序代码（也就是递归调用或使用软件堆栈的代码）。此外，它们使用户能更有效地用高级语言对数据结构执行特定的操作。包括：

- 在进入和退出子程序时对软件堆栈空间进行动态分配和释放
- 函数指针调用
- 对软件堆栈指针进行控制
- 对软件堆栈中的变量进行控制

表 25-3 提供了扩展指令集中的指令汇总。第 25.2.2 节“扩展指令集”对这些指令进行了详细说明。表 25-1（第 300 页）提供了标准和扩展的 PIC18 指令集的操作码字段说明。

注：扩展的指令集和立即数变址寻址模式是专为优化用 C 语言编写的应用程序而设计的，用户可能不会在汇编器中直接使用这些指令。对于那些查看编译器生成代码的用户，这些命令的语法可作为参考。

25.2.1 扩展指令的语法

大部分扩展指令都使用变址参数，同时使用一个文件选择寄存器和某一偏移量来指定源寄存器或目标寄存器。当指令的参数作为变址寻址的一部分时，会用方括号（“[]”）把它括起来。这时表示此参数用作变址地址或偏移量。如果 MPASM™ 汇编器发现一个变址地址或偏移量没有被括起来，它就会给出错误信息。

当使能扩展指令集时，括号也用于表示面向字节和面向位的指令中的变址参数。这是对指令语法的额外更改。欲知更多信息，请参见第 25.2.3.1 节“标准 PIC18 命令的扩展指令语法”。

注：以前，在 PIC18 和早期的指令集中使用方括号来表示可选参数。在此文本中，可选参数将用大括号（“{}”）表示。

表 25-3: PIC18 指令集的扩展

助记符， 操作数		说明	周期数	16 位指令字				受影响的状态位
				MSb		LSb		
ADDFSR	f, k	将立即数与 FSR 相加	1	1110	1000	ffkk	kkkk	无
ADDULNK	k	将立即数与 FSR2 相加并返回	2	1110	1000	11kk	kkkk	无
CALLW		使用 WREG 调用子程序	2	0000	0000	0001	0100	无
MOVSF	z <sub>s</sub> , f <sub>d</sub>	将 z <sub>s</sub> （源）移入第一个字	2	1110	1011	0zzz	zzzz	无
		f <sub>d</sub> （目标）移入第二个字		1111	ffff	ffff	ffff	
MOVSS	z <sub>s</sub> , z <sub>d</sub>	将 z <sub>s</sub> （源）移入第一个字	2	1110	1011	1zzz	zzzz	无
		z <sub>d</sub> （目标）移入第二个字		1111	xxxx	xzzz	zzzz	
PUSHL	k	将立即数保存在 FSR2 后，FSR2 减 1	1	1110	1010	kkkk	kkkk	无
SUBFSR	f, k	FSR 减去立即数	1	1110	1001	ffkk	kkkk	无
SUBULNK	k	FSR2 减去立即数并返回	2	1110	1001	11kk	kkkk	无

# PIC18F85J11 系列

## 25.2.2 扩展指令集

ADDFSR	FSR 的内容与立即数相加			
语法:	ADDFSR f, k			
操作数:	$0 \leq k \leq 63$ $f \in [0, 1, 2]$			
操作:	$FSR(f) + k \rightarrow FSR(f)$			
受影响的状态位:	无			
机器码:	1110	1000	ffkk	kkkk
说明:	将由 f 指定的 FSR 的内容加上一个 6 位的立即数 k。			
指令字数:	1			
指令周期数:	1			
Q 周期操作:				
Q1	Q2	Q3	Q4	
译码	读立即数 k	处理数据	写入 FSR	

示例:                   ADDFSR 2, 23h

执行指令前  
FSR2     =   03FFh  
执行指令后  
FSR2     =   0422h

ADDULNK		FSR2 的内容与立即数相加并返回							
语法:		ADDULNK k							
操作数:		0 ≤ k ≤ 63							
操作:		FSR2 + k → FSR2, (TOS) → PC							
受影响的状态位:		无							
机器码:		<table border="1"><tr><td>1110</td><td>1000</td><td>11kk</td><td>kkkk</td></tr></table>				1110	1000	11kk	kkkk
1110	1000	11kk	kkkk						
说明:		将 FSR2 的内容加上一个 6 位立即数 k。然后通过将 TOS 装入 PC，执行一条 RETURN 指令。							
		执行该指令需要两个周期：在第二个周期执行一条 NOP 指令。							
		该指令可以被认为是 ADDFSR 指令的特例，其中 f = 3（二进制 11）；它仅针对 FSR2 进行操作。							
指令字数:		1							
指令周期数:		2							
Q 周期操作:									
Q1		Q2		Q3		Q4			
译码		读立即数 k		处理数据		写入 FSR			
空操作		空操作		空操作		空操作			

示例:                   ADDULNK 23h

执行指令前  
FSR2     =   03FFh  
PC       =   0100h  
执行指令后  
FSR2     =   0422h  
PC       =   (TOS)

**注：** 所有的 PIC18 指令都可能在其指令助记符之前使用可选的标号参数，用于符号寻址。如果使用了标号，那么指令语法将变为: {label} 指令参数。

## CALLW 使用 WREG 调用子程序

语法:	CALLW												
操作数:	无												
操作:	(PC + 2) → TOS, (W) → PCL, (PCLATH) → PCH, (PCLATU) → PCU												
受影响的状态位:	无												
机器码:	<table><tr><td>0000</td><td>0000</td><td>0001</td><td>0100</td></tr></table>	0000	0000	0001	0100								
0000	0000	0001	0100										
说明	<p>首先, 返回地址 (PC + 2) 被压入返回堆栈。接下来, 将 W 寄存器的内容写入 PCL; PCL 现有的值被丢弃。然后, PCLATH 和 PCLATU 的内容被分别锁存到 PCH 和 PCU。第二个周期执行一条 NOP 指令, 并同时取下一条指令。</p> <p>和 CALL 指令不一样, 该指令没有更新 W、STATUS 或 BSR 寄存器的选项。</p>												
指令字数:	1												
指令周期数:	2												
Q 周期操作:	<table><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>译码</td><td>读 WREG</td><td>将 PC 压入 堆栈</td><td>空操作</td></tr><tr><td>空操作</td><td>空操作</td><td>空操作</td><td>空操作</td></tr></table>	Q1	Q2	Q3	Q4	译码	读 WREG	将 PC 压入 堆栈	空操作	空操作	空操作	空操作	空操作
Q1	Q2	Q3	Q4										
译码	读 WREG	将 PC 压入 堆栈	空操作										
空操作	空操作	空操作	空操作										

示例:                    HERE      CALLW

执行指令前

PC        =    地址 (HERE)  
PCLATH   =   10h  
PCLATU   =   00h  
W        =   06h

执行指令后

PC        =   001006h  
TOS       =   地址 (HERE + 2)  
PCLATH   =   10h  
PCLATU   =   00h  
W        =   06h

## MOVSF 将变址寻址单元内容移入 f

语法:  $MOVSF [z_s], f_d$

操作数:  $0 \leq z_s \leq 127$   
 $0 \leq f_d \leq 4095$

操作:  $((FSR2) + z_s) \rightarrow f_d$

受影响的状态位: 无

机器码:

1110	1011	0zzz	zzzz <sub>s</sub>
1111	ffff	ffff	ffff <sub>d</sub>

第一个字 (源)

第二个字 (目标)

说明:

源寄存器的内容移入目标寄存器  $f_d$ 。通过将第一个字中的 7 位立即数偏移量  $z_s$  与  $FSR2$  的值相加, 来确定源寄存器的实际地址。第二个字中的 12 位立即数  $f_d$  指向目标寄存器的地址。两个地址均可 以是 4096 字节的数据空间 (000h 到 FFFh) 中的任何单元。

$MOVSF$  指令中的目标寄存器不能是  $PCL$ 、 $TOSU$ 、 $TOSH$  或  $TOSL$ 。

如果计算得到的源地址指向间接寻址寄存器, 将返回 00h。

指令字数: 2

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	确定源地址	确定源地址	读源寄存器
译码	空操作	空操作	写寄存器 $f$ (目标寄存器)
	非无效读取		

示例:                    MOVSF    [05h], REG2

执行指令前

FSR2        =    80h  
85h 单元的  
内容        =    33h  
REG2        =    11h

执行指令后

FSR2        =    80h  
85h 单元的  
内容        =    33h  
REG2        =    33h

# PIC18F85J11 系列

## MOVSS 变址寻址传送数据

语法: MOVSS [ $z_s$ ], [ $z_d$ ]

操作数:  $0 \leq z_s \leq 127$   
 $0 \leq z_d \leq 127$

操作:  $((FSR2) + z_s) \rightarrow ((FSR2) + z_d)$

受影响的状态位: 无

机器码:				
第一个字 (源)	1110	1011	1zzz	zzzz $z_s$
第二个字 (目标)	1111	xxxx	xzzz	zzzz $z_d$

说明

将源寄存器的内容送入目标寄存器。通过将 FSR2 中的值分别加上 7 位立即数偏移量  $z_s$  或  $z_d$  来确定源寄存器和目标寄存器的地址。两个寄存器都可以是 4096 字节的数据空间 (000h 到 FFFh) 中的任意单元。

MOVSS 指令中的目标寄存器不能是 PCL、TOSU、TOSH 或 TOSL。

如果计算得到的源地址指向间接寻址寄存器, 将返回 00h。如果计算得到的目标地址指向间接寻址寄存器, 指令将作为一条 NOP 指令执行。

指令字数: 2

指令周期数: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	确定源地址	确定源地址	读源寄存器
译码	确定目标地址	确定目标地址	写目标寄存器

示例: MOVSS [05h], [06h]

执行指令前

FSR2 = 80h

85h 单元的内容 = 33h

86h 单元的内容 = 11h

执行指令后

FSR2 = 80h

85h 单元的内容 = 33h

86h 单元的内容 = 33h

87h 单元的内容 = 33h

## PUSHL 将立即数保存在 FSR2 后, FSR2 减 1

语法: PUSHL k

操作数:  $0 \leq k \leq 255$

操作:  $k \rightarrow (FSR2)$ ,  
 $FSR2 - 1 \rightarrow FSR2$

受影响的状态位: 无

机器码:	1111	1010	kkkk	kkkk
------	------	------	------	------

说明: 8 位立即数 k 被写入由 FSR2 指定的数据存储单元。操作完成后 FSR2 减 1。

该指令允许用户将值压入软件堆栈。

指令字数: 1

指令周期数: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读取 k	处理数据	写入目标寄存器

示例: PUSHL 08h

执行指令前

FSR2H:FSR2L = 01ECh

存储单元 (01ECh) = 00h

执行指令后

FSR2H:FSR2L = 01EBh

存储单元 (01ECh) = 08h



SUBFSR

FSR 减去立即数

语法:

SUBFSR f, k

操作数:

$0 \leq k \leq 63$

$f \in [0, 1, 2]$

操作:

$FSRf - k \rightarrow FSRf$

受影响的状态位:

无

机器码:

1110	1001	ffkk	kkkk
------	------	------	------

说明:

用寄存器 f 指定的 FSR 的内容减去 6 位立即数 k。

指令字数:

1

指令周期数:

1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器

示例: SUBFSR 2, 23h

执行指令前	FSR2	=	03FFh
执行指令后	FSR2	=	03DCh

SUBULNK

FSR2 减去立即数并返回

语法:

SUBULNK k

操作数:

$0 \leq k \leq 63$

操作:

$FSR2 - k \rightarrow FSR2,$   
 $(TOS) \rightarrow PC$

受影响的状态位:

无

机器码:

1110	1001	11kk	kkkk
------	------	------	------

说明:

用 FSR2 的内容减去 6 位立即数 k。然后通过将 TOS 装入 PC 执行一条 RETURN 指令。

指令字数:

1

指令周期数:

2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写入目标寄存器
空操作	空操作	空操作	空操作

示例: SUBULNK 23h

执行指令前	FSR2	=	03FFh
	PC	=	0100h
执行指令后	FSR2	=	03DCh
	PC	=	(TOS)

## 25.2.3 立即数变址寻址模式中字节和位操作类指令

**注：** 使能 PIC18 扩展指令集可能导致常规应用程序运行不正常或完全失败。

一旦使能扩展指令集，除了可以使用 8 条新指令之外，还可以使用立即数变址寻址模式（见第 6.6.1 节“使用立即数作为偏移量进行变址寻址”）。这将导致很多标准 PIC18 指令的地址解析方法有很大变化。

当禁用扩展指令集时，被嵌入在操作码中的地址被视作立即数存储单元：可以是快速操作存储区中的单元（ $a = 0$ ），或由 BSR 指定的 GPR 存储区中的单元（ $a = 1$ ）。当使能扩展指令集且  $a = 0$  时，地址为 5Fh 或以下的文件选择寄存器参数被解析为 FSR2 中的指针值的偏移量，而不是一个立即数地址。对于实际应用来说，这意味着所有使用快速操作 RAM 位作为参数的指令（即所有字节或位操作类指令或几乎半数的 PIC18 内核指令）在使能了扩展指令集时操作都会有所不同。

当 FSR2 的内容为 00h，快速操作 RAM 的边界会被重新映射到它们的原始值。这对于编写向下兼容的代码很有用处。如果使用此技术，有必要在 C 程序中调用汇编子程序时保存 FSR2 的值并在返回时将它恢复，这样做的目的是保护堆栈指针。用户还必须记住扩展指令集的语法要求（见第 25.2.3.1 节“标准 PIC18 命令的扩展指令语法”）。

虽然立即数变址寻址模式对于动态堆栈和指针控制很有用处，但是如果不小心在简单的算术运算中误用了寄存器，也会非常麻烦。已经习惯使用 PIC18 编程的用户必须记住，在使能了扩展指令集后，地址小于或等于 5Fh 的寄存器用于立即数变址寻址。

下面是在立即数变址寻址模式中，一些典型的字节和位操作类指令示例，通过示例可以看出指令如何受到影响。示例中的操作数条件适用于所有这一类的指令。

### 25.2.3.1 标准 PIC18 命令的扩展指令语法

当使能了扩展指令集时，立即数偏移量  $k$  被用来替换标准的字节和位操作类命令中的文件选择寄存器参数  $f$ 。如前所述，只有在  $f$  小于或等于 5Fh 时才会发生这种情况。当使用偏移量时，该偏移量必须用方括号（“[]”）标出。因为在扩展指令集中，方括号指示编译器将括号中的数值解析为变址地址或偏移量。省略括号，或在括号内使用大于 5Fh 的值会在 MPASM 汇编器中产生错误。

如果变址寻址模式下的变址参数已被加上括号，那么就不再需要指定快速操作 RAM 参数；自动将此参数假定为 0。在  $a$  根据目标地址置 1 时，这与标准操作（禁止扩展指令集）正好相反。在变址寻址模式中，声明快速操作 RAM 位也将在 MPASM 汇编器中产生错误。

目标参数  $d$  的操作和以前一样。

在 MPASM 汇编器的最新版本中，必须明确调用对扩展的指令集的语言支持。可以通过命令行选项 `/Y` 或在源代码中加入 PE 伪指令进行调用。

### 25.2.4 使能扩展指令集时的注意事项

需要注意的是并非所有用户都有必要使用扩展指令集，尤其是那些不使用软件堆栈编写代码的用户，使用扩展指令集并不会获得益处。

此外，立即数变址寻址模式可能会给写入 PIC18 汇编器的常规应用程序带来问题。这是因为常规代码中的指令会尝试寻址快速操作存储区中地址低于 5Fh 的寄存器。当使能了扩展指令集时，这些地址会被解析为相对于 FSR2 的立即数偏移量，所以应用程序会读或写错误的地址。

将应用程序移植到 PIC18F85J11 系列器件时，代码的类型是非常重要的。在使用扩展指令集时，代码较长的重入应用程序（用 C 语言编写且可高效编译）会运行的很好，而大量使用快速操作存储区的常规应用程序不会获得任何益处。

ADDWF		将 W 与变址寻址单元的内容相加 (立即数变址寻址模式)											
语法:	ADDWF [k] {,d}												
操作数:	$0 \leq k \leq 95$ $d \in [0, 1]$												
操作:	$(W) + ((FSR2) + k) \rightarrow dest$												
受影响的状态位:	N、OV、C、DC 和 Z												
机器码:	<table><tr><td>0010</td><td>01d0</td><td>kkkk</td><td>kkkk</td></tr></table>					0010	01d0	kkkk	kkkk				
0010	01d0	kkkk	kkkk										
说明:	将 W 的内容与由 FSR2 加上偏移量 k 指定的寄存器的内容相加。  如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存储回寄存器 f。												
指令字数:	1												
指令周期数:	1												
Q 周期操作:	<table><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>译码</td><td>读取 k</td><td>处理数据</td><td>写入 目标寄存器</td></tr></table>					Q1	Q2	Q3	Q4	译码	读取 k	处理数据	写入 目标寄存器
Q1	Q2	Q3	Q4										
译码	读取 k	处理数据	写入 目标寄存器										

示例:                    ADDWF    [OFST] , 0

执行指令前	
W	= 17h
OFST	= 2Ch
FSR2	= 0A00h
0A2Ch 单元的内容	= 20h
执行指令后	
W	= 37h
0A2Ch 单元的内容	= 20h

BSF		将变址寻址单元相应位置 1 (立即数变址寻址模式)						
语法:	BSF [k], b							
操作数:	$0 \leq f \leq 95$ $0 \leq b \leq 7$							
操作:	$1 \rightarrow ((FSR2) + k) < b >$							
受影响的状态位:	无							
机器码:	<table border="1"><tr><td>1000</td><td>bbb0</td><td>kkkk</td><td>kkkk</td></tr></table>				1000	bbb0	kkkk	kkkk
1000	bbb0	kkkk	kkkk					
说明:	将由 FSR2 加上偏移量 k 指定的寄存器中的位 b 置 1。							
指令字数:	1							
指令周期数:	1							
Q 周期操作:								
	Q1	Q2	Q3	Q4				
	译码	读寄存器 f	处理数据	写入目标寄存器				

示例:                    BSF        [FLAG\_OFST] , 7

执行指令前	
FLAG_OFST	= 0Ah
FSR2	= 0A00h
0A0Ah 单元的内容	= 55h
执行指令后	
0A0Ah 单元的内容	= D5h

SETF		将变址寻址单元置全 1 (立即数变址寻址模式)							
语法:	SETF [k]								
操作数:	$0 \leq k \leq 95$								
操作:	$FFh \rightarrow ((FSR2) + k)$								
受影响的状态位:	无								
机器码:	<table border="1"><tr><td>0110</td><td>1000</td><td>kkkk</td><td>kkkk</td></tr></table>					0110	1000	kkkk	kkkk
0110	1000	kkkk	kkkk						
说明:	将由 FSR2 加上偏移量 k 指定的寄存器的内容置为 FFh。								
指令字数:	1								
指令周期数:	1								
Q 周期操作:									
	Q1	Q2	Q3	Q4					
	译码	读取 k	处理数据	写寄存器					

示例:                    SETF        [OFST]

执行指令前	
OFST	= 2Ch
FSR2	= 0A00h
0A2Ch 单元的内容	= 00h
执行指令后	
0A2Ch 单元的内容	= FFh

# PIC18F85J11 系列

---

## 25.2.5 使用 MICROCHIP MPLAB® IDE 工具的注意事项

最新版本的 Microchip 软件工具完全支持 PIC18F85J11 系列的扩展指令集，包括 MPLAB C18 C 编译器、MPASM 汇编器和 MPLAB 集成开发环境（IDE）。

在选择了使用软件开发的目標器件后，MPLAB IDE 将把该器件的默认配置位自动置 1。XINST 配置位的默认设置是 1，即使能扩展指令集和立即数变址寻址模式。在编程过程中必须将 XINST 位置 1 才能正确地利用扩展的指令集开发应用程序。

要使用扩展指令集开发软件，用户必须设置他们的语言工具以实现对扩展指令和变址寻址模式的支持。根据所使用的环境，可以通过以下几种方法实现：

- 开发环境中的菜单选项或对话框，允许用户配置项目的语言工具及其设置
- 命令行选项
- 源代码中的伪指令

这些选项在不同的编译器、汇编器和开发环境中将有所不同。建议用户在其开发系统所附带的文档中查询相应的信息。

## 26.0 电气特性

### 绝对最大值 (†)

环境温度.....	-40°C 至 +100°C
储存温度.....	-65°C 至 +150°C
任何数字 I/O 引脚或 $\overline{\text{MCLR}}$ ( $V_{DD}$ 除外) 相对于 $V_{SS}$ 的电压.....	-0.3V 至 5.6V
任何模数组合引脚 ( $V_{DD}$ 和 $\overline{\text{MCLR}}$ 除外) 相对于 $V_{SS}$ 的电压 .....	-0.3V 至 ( $V_{DD} + 0.3V$ )
$V_{DDCORE}$ 相对于 $V_{SS}$ 的电压.....	-0.3V 至 2.75V
$V_{DD}$ 相对于 $V_{SS}$ 的电压 .....	-0.3V 至 3.6V
总功耗 (注 1).....	1.0W
$V_{SS}$ 引脚最大输出电流 .....	300 mA
$V_{DD}$ 引脚最大输入电流.....	250 mA
PORTA<7:6> 以及任何 PORTB 和 PORTC I/O 引脚的最大输出灌电流.....	25 mA
任何 PORTD、PORTE 和 PORTJ I/O 引脚的最大输出灌电流.....	8 mA
PORTA<5:0> 以及任何 PORTF、PORTG 和 PORTH I/O 引脚的最大输出灌电流.....	2 mA
PORTA<7:6> 以及任何 PORTB 和 PORTC I/O 引脚的最大输出拉电流.....	25 mA
任何 PORTD、PORTE 和 PORTJ I/O 引脚的最大输出拉电流.....	8 mA
PORTA<5:0> 以及任何 PORTF、PORTG 和 PORTH I/O 引脚的最大输出拉电流.....	2 mA
所有组合端口的最大灌电流.....	200 mA

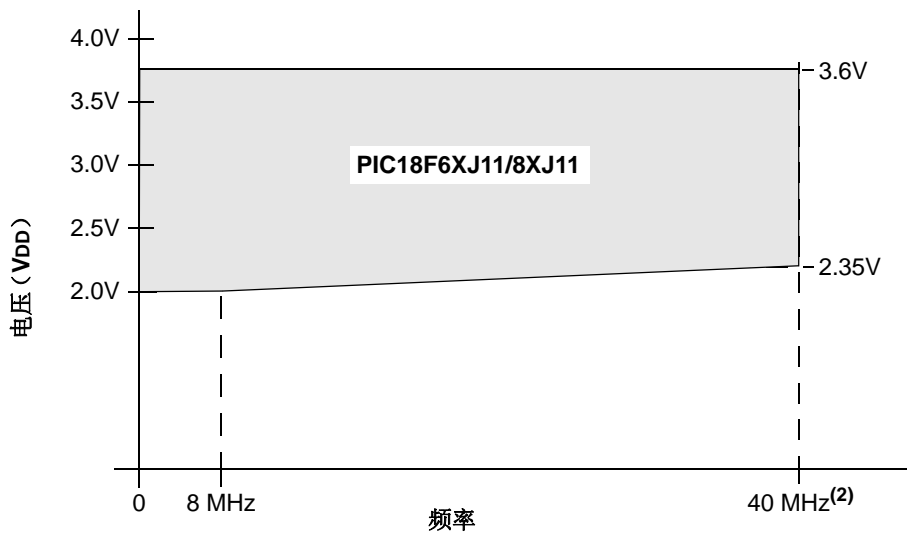
注 1: 功耗按如下公式计算:

$$P_{dis} = V_{DD} \times \{I_{DD} - I_{OH}\} + \{(V_{DD} - V_{OH}) \times I_{OH}\} + (V_{OL} \times I_{OL})$$

† 注: 如果器件工作条件超过上述“绝对最大值”, 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, 我们不建议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。

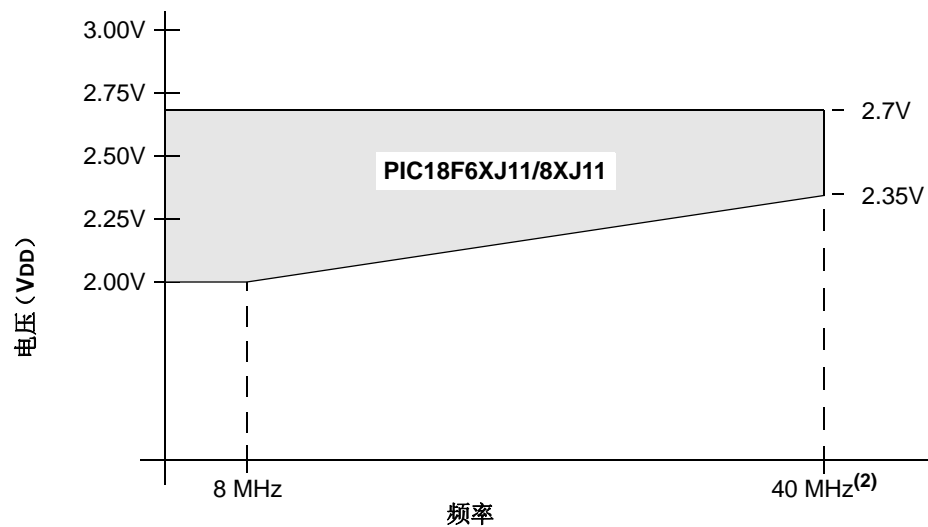
# PIC18F85J11 系列

图 26-1: PIC18F85J11 系列电压——频率关系图（使能稳压器，工业级）<sup>(1)</sup>



- 注 1: 使能片上稳压器时，在 V<sub>DD</sub> 达到无法全速运行的电平之前，稳压器的 BOR 将自动触发器件复位。
- 2: 在 8 位外部存储器模式下，F<sub>MAX</sub> = 25 MHz。

图 26-2: PIC18F85J11 系列电压——频率关系图（禁止稳压器，工业级）<sup>(1,3)</sup>



- 注 1: 对于位于 4 MHz 和 40 MHz 之间的频率， $F_{MAX} = (51.42 \text{ MHz/V}) * (V_{DDCORE} - 2V) + 4 \text{ MHz}$ 。
- 2: 在 8 位外部存储器模式下，F<sub>MAX</sub> = 25 MHz。
- 3: 当禁止片上稳压器时，V<sub>DD</sub> 和 V<sub>DDCORE</sub> 必须保持原来的关系，从而保证  $V_{DDCORE} \leq V_{DD} \leq 3.6V$ 。

## 26.1 直流特性:

供电电压

PIC18F85J11 系列 (工业级)

PIC18F85J11 系列 (工业级)			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)				
参数 编号	符号	特性	最小值	典型值	最大值	单位	条件
D001	VDD	供电电压	VDDCORE 2.0	— —	3.6 3.6	V V	ENVREG 连接到 Vss ENVREG 连接到 VDD
D001B	VDDCORE	单片机内核的外部供电电压	2.0	—	2.70	V	ENVREG 连接到 Vss
D001C	AVDD	模拟电源	VDD 0.3	—	VDD + 0.3	V	
D001D	AVSS	模拟地的电势	Vss 0.3	—	Vss + 0.3	V	
D002	VDR	RAM 数据保存电压 <sup>(1)</sup>	1.5	—	—	V	
D003	VPOR	VDD 启动电压 确保产生内部上电复位信号	—	—	0.7	V	详情请参见第 5.3 节 “上电复位 (POR)”。
D004	SVDD	VDD 上升率 确保产生内部上电复位信号	0.05	—	—	V/ms	详情请参见第 5.3 节 “上电复位 (POR)”。
D005	VBOR	欠压复位电压	—	1.9	—	V	

注 1: 该电压是休眠模式或器件复位状态下, 在不丢失 RAM 数据的前提下的最小 VDD。

# PIC18F85J11 系列

## 26.2 直流特性： 掉电电流和供电电流 PIC18F85J11 系列（工业级）

PIC18F85J11 系列 (工业级)		标准工作条件（除非另外说明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）				
参数 编号	器件	典型值	最大值	单位	条件	
	掉电电流 (IPD) <sup>(1)</sup>					
	所有器件	0.2	0.9	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V (休眠模式) <sup>(4)</sup>
		0.1	0.9	μA	+25°C	
		0.3	3	μA	+60°C	
		2.4	5	μA	+85°C	
	所有器件	0.5	0.9	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V (休眠模式) <sup>(4)</sup>
		0.1	0.9	μA	+25°C	
		0.4	3	μA	+60°C	
		2.7	5	μA	+85°C	
	所有器件	2.7	6	μA	-40°C	VDD = 3.3V, (休眠模式) <sup>(5)</sup>
		3.5	6	μA	+25°C	
		4.1	8	μA	+60°C	
		6.7	12	μA	+85°C	

- 注 1: 休眠模式下的掉电电流不是由振荡器类型决定的。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  $V_{DD}$  或  $V_{SS}$ ，以及禁止所有会带来新增电流的功能部件（比如 WDT、Timer1 振荡器或 BOR 等）时测得的。
- 2: 供电电流主要是由工作电压、频率和模式一起决定的。其他因素，如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。  
在正常工作模式下，所有  $I_{DD}$  测量的测试条件为：  
 $\text{OSC1} = \text{外部方波，满幅；所有 I/O 引脚均为三态，拉至 } V_{DD}$ ；  
 $\text{MCLR} = V_{DD}$ ；根据具体应用禁止或使能 WDT。
- 3: 标准低成本 32 kHz 晶振的工作温度范围为  $-10^{\circ}\text{C}$  到  $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。
- 4: 禁止稳压器（ENVREG 连接到  $V_{SS}$ ）。
- 5: 使能稳压器（ENVREG 连接到  $V_{DD}$ ）。



## 26.2 直流特性:

### 掉电电流和供电电流

#### PIC18F85J11 系列 (工业级) (续)

PIC18F85J11 系列 (工业级)		标准工作条件 (除非另外说明) 工作温度 -40°C ≤ Ta ≤ +85°C (工业级)					
参数 编号	器件	典型值	最大值	单位	条件		
	供电电流 (I <sub>DD</sub> ) (2)						
	所有器件	6.5	16	μA	-40°C	V <sub>DD</sub> = 2.0V, V <sub>DDCORE</sub> = 2.0V(4)	FOSC = 31 kHz (RC_RUN 模式, 内部振荡器时钟源)
		7	16	μA	+25°C		
		9.5	20	μA	+85°C		
	所有器件	10	18	μA	-40°C	V <sub>DD</sub> = 2.5V, V <sub>DDCORE</sub> = 2.5V(4)	
		10.5	18	μA	+25°C		
		12.5	24	μA	+85°C		
	所有器件	41	100	μA	-40°C	V <sub>DD</sub> = 3.3V(5)	
		52	100	μA	+25°C		
		71	110	μA	+85°C		
	所有器件	359	750	μA	-40°C	V <sub>DD</sub> = 2.0V, V <sub>DDCORE</sub> = 2.0V(4)	FOSC = 1 MHz (INTOSC_RUN 模式, 内部振荡器时钟源)
		387	750	μA	+25°C		
		407	840	μA	+85°C		
	所有器件	438	850	μA	-40°C	V <sub>DD</sub> = 2.5V, V <sub>DDCORE</sub> = 2.5V(4)	
		470	850	μA	+25°C		
		491	910	μA	+85°C		
	所有器件	486	900	μA	-40°C	V <sub>DD</sub> = 3.3V(5)	
		526	900	μA	+25°C		
		564	990	μA	+85°C		
	所有器件	0.76	1.45	mA	-40°C	V <sub>DD</sub> = 2.0V, V <sub>DDCORE</sub> = 2.0V(4)	FOSC = 4 MHz (INTOSC_RUN 模式, 内部振荡器时钟源)
		0.84	1.45	mA	+25°C		
		0.9	1.6	mA	+85°C		
	所有器件	1.1	1.63	mA	-40°C	V <sub>DD</sub> = 2.5V, V <sub>DDCORE</sub> = 2.5V(4)	
		1.18	1.63	mA	+25°C		
		1.24	1.75	mA	+85°C		
	所有器件	1.25	1.86	mA	-40°C	V <sub>DD</sub> = 3.3V(5)	
		1.29	1.86	mA	+25°C		
1.37		1.94	mA	+85°C			

**注 1:** 休眠模式下的掉电电流不是由振荡器类型决定的。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 或 VSS, 以及禁止所有会带来新增电流的功能部件 (比如 WDT、Timer1 振荡器或 BOR 等) 时测得的。

**2:** 供电电流主要是由工作电压、频率和模式一起决定的。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。

在正常工作模式下, 所有  $I_{DD}$  测量的测试条件为:

$\overline{\text{OSC1}}$  = 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 VDD;

$\overline{\text{MCLR}}$  = VDD; 根据具体应用禁止或使能 WDT。

**3:** 标准低成本 32 kHz 晶振的工作温度范围为  $-10^{\circ}\text{C}$  到  $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。

**4:** 禁止稳压器 (ENVREG 连接到 VSS)。

**5:** 使能稳压器 (ENVREG 连接到 VDD)。

# PIC18F85J11 系列

## 26.2 直流特性: 掉电电流和供电电流 PIC18F85J11 系列 (工业级) (续)

PIC18F85J11 系列 (工业级)		标准工作条件（除非另外说明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）					
参数 编号	器件	典型值	最大值	单位	条件		
	供电电流（IDD）(2)						
	所有器件	2.4	8	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 31 kHz (RC_IDLE 模式, 内部振荡器时钟源)
		2.5	8	μA	+25°C		
		4.8	12	μA	+85°C		
	所有器件	3.2	9	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		3.2	9	μA	+25°C		
		6	14	μA	+85°C		
	所有器件	62	82	μA	-40°C	VDD = 3.3V(5)	
		42	82	μA	+25°C		
		59	97	μA	+85°C		
	所有器件	251	570	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 1 MHz (INTOSC_IDLE 模式, 内部振荡器时钟源)
		264	570	μA	+25°C		
		272	590	μA	+85°C		
	所有器件	284	610	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		284	610	μA	+25°C		
		293	650	μA	+85°C		
	所有器件	295	710	μA	-40°C	VDD = 3.3V(5)	
		323	710	μA	+25°C		
		392	790	μA	+85°C		
	所有器件	368	760	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 4 MHz (INTOSC_IDLE 模式, 内部振荡器时钟源)
		362	760	μA	+25°C		
		370	800	μA	+85°C		
	所有器件	400	850	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		410	850	μA	+25°C		
		418	900	μA	+85°C		
	所有器件	460	950	μA	-40°C	VDD = 3.3V(5)	
		462	950	μA	+25°C		
486		1,000	μA	+85°C			

- 注 1: 休眠模式下的掉电电流不是由振荡器类型决定的。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 或 VSS, 以及禁止所有会带来新增电流的功能部件 (比如 WDT、Timer1 振荡器或 BOR 等) 时测得的。
- 2: 供电电流主要是由工作电压、频率和模式一起决定的。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。  
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;  
 $MCLR = V_{DD}$ ; 根据具体应用禁止或使能 WDT。
- 3: 标准低成本 32 kHz 晶振的工作温度范围为  $-10^{\circ}\text{C}$  到  $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。
- 4: 禁止稳压器 (ENVREG 连接到 VSS)。
- 5: 使能稳压器 (ENVREG 连接到 VDD)。

## 26.2 直流特性:

### 掉电电流和供电电流

#### PIC18F85J11 系列 (工业级) (续)

PIC18F85J11 系列 (工业级)		标准工作条件（除非另外说明） 工作温度 -40°C ≤ Ta ≤ +85°C（工业级）					
参数 编号	器件	典型值	最大值	单位	条件		
	供电电流 (IDD) (2)						
	所有器件	165	490	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 1 MHz (PRI_RUN 模式, EC 振荡器)
		180	490	mA	+25°C		
		200	490	mA	+85°C		
	所有器件	256	670	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		260	670	mA	+25°C		
		280	670	mA	+85°C		
	所有器件	460	850	mA	-40°C	VDD = 3.3V(5)	
		456	850	mA	+25°C		
		482	850	mA	+85°C		
	所有器件	0.63	2.2	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 4 MHz (PRI_RUN 模式, EC 振荡器)
		0.68	2.2	mA	+25°C		
		0.74	2.2	mA	+85°C		
	所有器件	0.91	2.5	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		1.04	2.5	mA	+25°C		
		1.04	2.5	mA	+85°C		
	所有器件	1.32	3.0	mA	-40°C	VDD = 3.3V(5)	
		1.32	3.0	mA	+25°C		
		1.41	3.0	mA	+85°C		
	所有器件	7.47	14	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	FOSC = 40 MHz (PRI_RUN 模式, EC 振荡器)
		5.81	14	mA	+25°C		
		6.32	13	mA	+85°C		
	所有器件	8.84	18	mA	-40°C	VDD = 3.3V(5)	
		8.66	18	mA	+25°C		
7.97		16	mA	+85°C			

- 注 1:** 休眠模式下的掉电电流不是由振荡器类型决定的。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 或 VSS, 以及禁止所有会带来新增电流的功能部件 (比如 WDT、Timer1 振荡器或 BOR 等) 时测得的。
- 2:** 供电电流主要是由工作电压、频率和模式一起决定的。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。  
在正常工作模式下, 所有  $I_{DD}$  测量的测试条件为:  
 $\overline{\text{OSC1}}$  = 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 VDD;  
 $\overline{\text{MCLR}}$  = VDD; 根据具体应用禁止或使能 WDT。
- 3:** 标准低成本 32 kHz 晶振的工作温度范围为  $-10^{\circ}\text{C}$  到  $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。
- 4:** 禁止稳压器 (ENVREG 连接到 VSS)。
- 5:** 使能稳压器 (ENVREG 连接到 VDD)。

# PIC18F85J11 系列

## 26.2 直流特性: 掉电电流和供电电流 PIC18F85J11 系列 (工业级) (续)

PIC18F85J11 系列 (工业级)		标准工作条件 (除非另外说明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)					
参数 编号	器件	典型值	最大值	单位	条件		
	供电电流 (IDD) (2)						
	所有器件	2.8	3.8	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 4 MHz, 16 MHz 内部 (PRI_RUN 模式, HSPLL 振荡器)
		3.02	3.8	mA	+25°C		
		3.01	4.5	mA	+85°C		
	所有器件	4.5	5.4	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	FOSC = 4 MHz, 16 MHz 内部 (PRI_RUN 模式, HSPLL 振荡器)
		4.8	5.6	mA	+25°C		
		4.54	5.6	mA	+85°C		
	所有器件	5.72	6.7	mA	-40°C	VDD = 3.3V(5)	FOSC = 4 MHz, 16 MHz 内部 (PRI_RUN 模式, HSPLL 振荡器)
		5.55	6.5	mA	+25°C		
		5.3	6.5	mA	+85°C		
	所有器件	7.4	8.5	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	FOSC = 10 MHz, 40 MHz 内部 (PRI_RUN 模式, HSPLL 振荡器)
		7.23	8.5	mA	+25°C		
		6.55	7.5	mA	+85°C		
	所有器件	9.74	11.6	mA	-40°C	VDD = 3.3V(5)	FOSC = 10 MHz, 40 MHz 内部 (PRI_RUN 模式, HSPLL 振荡器)
		9.43	11.6	mA	+25°C		
		8.89	10.5	mA	+85°C		

- 注 1: 休眠模式下的掉电电流不是由振荡器类型决定的。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  $V_{DD}$  或  $V_{SS}$ , 以及禁止所有会带来新增电流的功能部件 (比如 WDT、Timer1 振荡器或 BOR 等) 时测得的。
- 2: 供电电流主要是由工作电压、频率和模式一起决定的。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。  
在正常工作模式下, 所有  $I_{DD}$  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  $V_{DD}$  ;  
 $MCLR$  =  $V_{DD}$  ; 根据具体应用禁止或使能 WDT。
- 3: 标准低成本 32 kHz 晶振的工作温度范围为  $-10^{\circ}\text{C}$  到  $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。
- 4: 禁止稳压器 (ENVREG 连接到  $V_{SS}$ )。
- 5: 使能稳压器 (ENVREG 连接到  $V_{DD}$ )。

## 26.2 直流特性:

### 掉电电流和供电电流

#### PIC18F85J11 系列（工业级）（续）

PIC18F85J11 系列 (工业级)		标准工作条件（除非另外说明） 工作温度 -40°C ≤ Ta ≤ +85°C （工业级）					
参数 编号	器件	典型值	最大值	单位	条件		
	供电电流 (IDD) (2)						
	所有器件	50	120	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 1 MHz (PRI_IDLE 模式, EC 振荡器)
		51	120	μA	+25°C		
		54	130	μA	+85°C		
	所有器件	223	480	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		134	300	μA	+25°C		
		110	270	μA	+85°C		
	所有器件	307	550	μA	-40°C	VDD = 3.3V(5)	
		254	500	μA	+25°C		
		194	460	μA	+85°C		
	所有器件	307	850	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 4 MHz (PRI_IDLE 模式, EC 振荡器)
		200	850	μA	+25°C		
		202	800	μA	+85°C		
	所有器件	483	950	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		318	950	μA	+25°C		
		343	900	μA	+85°C		
	所有器件	0.52	1.3	mA	-40°C	VDD = 3.3V(5)	
		0.47	1.2	mA	+25°C		
		0.47	1.2	mA	+85°C		
	所有器件	2.38	8	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	FOSC = 40 MHz (PRI_IDLE 模式, EC 振荡器)
		2.04	8	mA	+25°C		
		2.52	9	mA	+85°C		
	所有器件	3.02	10	mA	-40°C	VDD = 3.3V(5)	
		2.99	10	mA	+25°C		
4.23		11	mA	+85°C			

- 注 1:** 休眠模式下的掉电电流不是由振荡器类型决定的。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 或 VSS，以及禁止所有会带来新增电流的功能部件（比如 WDT、Timer1 振荡器或 BOR 等）时测得的。
- 2:** 供电电流主要是由工作电压、频率和模式一起决定的。其他因素，如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。  
在正常工作模式下，所有  $I_{DD}$  测量的测试条件为：  
 $\text{OSC1}$  = 外部方波，满幅；所有 I/O 引脚均为三态，拉至 VDD；  
 $\text{MCLR}$  = VDD；根据具体应用禁止或使能 WDT。
- 3:** 标准低成本 32 kHz 晶振的工作温度范围为  $-10^{\circ}\text{C}$  到  $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。
- 4:** 禁止稳压器（ENVREG 连接到 VSS）。
- 5:** 使能稳压器（ENVREG 连接到 VDD）。

# PIC18F85J11 系列

## 26.2 直流特性：掉电电流和供电电流 PIC18F85J11 系列（工业级）（续）

PIC18F85J11 系列 (工业级)		标准工作条件（除非另外说明） 工作温度 -40°C ≤ TA ≤ +85°C （工业级）					
参数 编号	器件	典型值	最大值	单位	条件		
	供电电流 (IDD) (2)						
	所有器件	10.5	22	μA	-10°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 32 kHz(3) ( <b>SEC_RUN</b> 模式, Timer1 作为时钟源)
		13.4	28	μA	+25°C		
		17.6	40	μA	+70°C		
	所有器件	13.2	30	μA	-10°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		16.2	35	μA	+25°C		
		20.7	50	μA	+70°C		
	所有器件	39	120	μA	-10°C	VDD = 3.3V(5)	
		58	150	μA	+25°C		
		75	190	μA	+70°C		
	所有器件	5.7	15	μA	-10°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 32 kHz(3) ( <b>SEC_IDLE</b> 模式, Timer1 作为时钟源)
		8.9	20	μA	+25°C		
		12.8	26	μA	+70°C		
	所有器件	6.6	17	μA	-10°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		9.7	24	μA	+25°C		
		13.7	30	μA	+70°C		
	所有器件	39	115	μA	-10°C	VDD = 3.3V(5)	
		52.8	145	μA	+25°C		
		72.7	185	μA	+70°C		

- 注 1: 休眠模式下的掉电电流不是由振荡器类型决定的。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 或 VSS，以及禁止所有会带来新增电流的功能部件（比如 WDT、Timer1 振荡器或 BOR 等）时测得的。
- 2: 供电电流主要是由工作电压、频率和模式一起决定的。其他因素，如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。  
在正常工作模式下，所有 IDD 测量的测试条件为：  
 $\text{OSC1} = \text{外部方波，满幅；所有 I/O 引脚均为三态，拉至 } V_{DD}$ ；  
 $\text{MCLR} = V_{DD}$ ；根据具体应用禁止或使能 WDT。
- 3: 标准低成本 32 kHz 晶振的工作温度范围为  $-10^{\circ}\text{C}$  到  $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。
- 4: 禁止稳压器（ENVREG 连接到 VSS）。
- 5: 使能稳压器（ENVREG 连接到 VDD）。

## 26.2 直流特性:

### 掉电电流和供电电流

#### PIC18F85J11 系列 (工业级) (续)

PIC18F85J11 系列 (工业级)		标准工作条件（除非另外说明） 工作温度-40°C ≤ TA ≤ +85°C（工业级）						
参数 编号	器件	典型值	最大值	单位	条件			
D022 (ΔIWDT)	模块差分电流（ΔIWDT、ΔIOSCB 和 ΔIAD）							
	看门狗定时器	1.6	4	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V <sup>(4)</sup>		
		1.7	4	μA	+25°C			
		1.6	4	μA	+85°C			
		2.5	5	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V <sup>(4)</sup>		
		2.5	5	μA	+25°C			
		2.3	5	μA	+85°C			
		3.8	6	μA	-40°C	VDD = 3.3V <sup>(5)</sup>		
		2.6	6	μA	+25°C			
		2.4	6	μA	+85°C			
D025 (ΔIOSCB)	Timer1 振荡器	6.6	12.5	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V <sup>(4)</sup>	Timer1 为 32 kHz <sup>(3)</sup>	
		7.9	12.5	μA	+25°C			
		11.5	18	μA	+85°C			
		7.2	12.5	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V <sup>(4)</sup>	Timer1 为 32 kHz <sup>(3)</sup>	
		8.1	12.5	μA	+25°C			
		11.9	18.5	μA	+85°C			
		7	12.5	μA	-40°C	VDD = 3.3V <sup>(5)</sup>	Timer1 为 32 kHz <sup>(3)</sup>	
		9	12.5	μA	+25°C			
11	18.5	μA	+85°C					
D026 (ΔIAD)	A/D 转换器	1	1.5	μA	-40°C 至 +85°C	VDD = 2.0V, VDDCORE = 2.0V <sup>(4)</sup>	A/D 启动，但不进行转换	
		1	1.5	μA	-40°C 至 +85°C			VDD = 2.5V, VDDCORE = 2.5V <sup>(4)</sup>
		1	1.5	μA	-40°C 至 +85°C			

- 注 1: 休眠模式下的掉电电流不是由振荡器类型决定的。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  $V_{DD}$  或  $V_{SS}$ , 以及禁止所有会带来新增电流的功能部件 (比如 WDT、Timer1 振荡器或 BOR 等) 时测得的。
- 2: 供电电流主要是由工作电压、频率和模式一起决定的。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。  
在正常工作模式下, 所有  $I_{DD}$  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  $V_{DD}$  ;  
 $MCLR$  =  $V_{DD}$  ; 根据具体应用禁止或使能 WDT。
- 3: 标准低成本 32 kHz 晶振的工作温度范围为  $-10^{\circ}\text{C}$  到  $+70^{\circ}\text{C}$ 。扩展级温度的晶振成本要高很多。
- 4: 禁止稳压器 (ENVREG 连接到  $V_{SS}$ )。
- 5: 使能稳压器 (ENVREG 连接到  $V_{DD}$ )。

# PIC18F85J11 系列

## 26.3 直流特性: PIC18F85J11 系列（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）			
参数编号	符号	特性	最小值	最大值	单位	条件
D030 D030A D031 D031A D031B D032 D033 D033A D034	$V_{IL}$	输入低电压 所有 I/O 端口： 带 TTL 缓冲器  带施密特触发缓冲器 RC3 和 RC4  $\overline{\text{MCLR}}$ OSC1 OSC1 T13CKI	$V_{SS}$ — $V_{SS}$ $V_{SS}$ $V_{SS}$ $V_{SS}$ $V_{SS}$ $V_{SS}$ $V_{SS}$	$0.15 V_{DD}$ 0.8 $0.2 V_{DD}$ $0.3 V_{DD}$ 0.8 $0.2 V_{DD}$ $0.3 V_{DD}$ $0.2 V_{DD}$ 0.3	V V V V V V V V V	$V_{DD} < 3.3\text{V}$ $3.3\text{V} \leq V_{DD} \leq 3.6\text{V}$  使能 $I^2C^{\text{TM}}$ 使能 SMBus  HS 和 HSPLL 模式 EC 和 ECPLL 模式 <sup>(1)</sup>
D040 D040A D041 D041A D041B  Dxxx DxxxA Dxxx D042 D043 D043A D044	$V_{IH}$	输入高电压 不能承受 5.5V 电压的 I/O 端口: <sup>(2)</sup> 带 TTL 缓冲器 带施密特触发缓冲器 RC3 和 RC4  能承受 5.5V 电压的 I/O 端口: <sup>(2)</sup> 带 TTL 缓冲器 带施密特触发缓冲器 $\overline{\text{MCLR}}$ OSC1 OSC1 T13CKI	$0.25 V_{DD} + 0.8\text{V}$ 2.0 $0.8 V_{DD}$ $0.7 V_{DD}$ 2.1  $0.25 V_{DD} + 0.8\text{V}$ 2.0 $0.8 V_{DD}$ $0.8 V_{DD}$ $0.7 V_{DD}$ $0.8 V_{DD}$ 1.6	$V_{DD}$ $V_{DD}$ $V_{DD}$ $V_{DD}$ $V_{DD}$  5.5 5.5 5.5 $V_{DD}$ $V_{DD}$ $V_{DD}$ $V_{DD}$	V V V V V  V V V V V V V	$V_{DD} < 3.3\text{V}$ $3.3\text{V} \leq V_{DD} \leq 3.6\text{V}$  使能 $I^2C$ 使能 SMBus, $V_{DD} > 3.3\text{V}$  $V_{DD} < 3.3\text{V}$ $3.3\text{V} \leq V_{DD} \leq 3.6\text{V}$  HS 和 HSPLL 模式 EC 和 ECPLL 模式
D060  D061 D063	$I_{IL}$	输入泄漏电流 <sup>(1)</sup> 不能承受 5.5V 电压的 I/O 端口 <sup>(2)</sup>  能承受 5.5V 电压的 I/O 端口 <sup>(2)</sup>  $\overline{\text{MCLR}}$ OSC1	— — — —	200 200 $\pm 1$ $\pm 1$	nA nA $\mu\text{A}$ $\mu\text{A}$	$V_{SS} \leq V_{PIN} \leq V_{DD}$ , 引脚处于高阻态 $V_{SS} \leq V_{PIN} \leq 5.5\text{V}$ , 引脚处于高阻态 $V_{SS} \leq V_{PIN} \leq V_{DD}$ $V_{SS} \leq V_{PIN} \leq V_{DD}$
D070	$I_{PU}$ $I_{PURB}$	弱上拉电流 PORTB 弱上拉电流	30	400	$\mu\text{A}$	$V_{DD} = 3.3\text{V}$ , $V_{PIN} = V_{SS}$

注 1: 负电流定义为引脚的拉电流。  
2: 关于相应引脚的可承受电压限制请参见表 11-1。



## 26.3 直流特性: PIC18F85J11 系列（工业级）（续）

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）			
参数编号	符号	特性	最小值	最大值	单位	条件
D080	VOL	输出低电压 I/O 端口:				
		PORTA、PORTF、PORTG 和 PORTH	—	0.4	V	$I_{OL} = 3.4\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ 至 $+85^{\circ}\text{C}$
		PORTD、PORTE 和 PORTJ	—	0.4	V	$I_{OL} = 3.4\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ 至 $+85^{\circ}\text{C}$
		PORTB 和 PORTC	—	0.4	V	$I_{OL} = 8.5\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ 至 $+85^{\circ}\text{C}$
D083		OSC2/CLKO (EC 和 ECPLL 模式)	—	0.4	V	$I_{OL} = 1.6\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ 至 $+85^{\circ}\text{C}$
D090	VOH	输出高电压 <sup>(1)</sup> I/O 端口:			V	
		PORTA、PORTF、PORTG 和 PORTH	2.4	—	V	$I_{OH} = -2\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ 至 $+85^{\circ}\text{C}$
		PORTD、PORTE 和 PORTJ	2.4	—	V	$I_{OH} = -2\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ 至 $+85^{\circ}\text{C}$
		PORTB 和 PORTC	2.4	—	V	$I_{OH} = -6\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ 至 $+85^{\circ}\text{C}$
D092		OSC2/CLKO (INTOS、EC 和 ECPLL 模式)	2.4	—	V	$I_{OH} = -1\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ 至 $+85^{\circ}\text{C}$
输出引脚上的容性负载规范						
D100	COSC2	OSC2 引脚	—	15	pF	在外部时钟用于驱动 OSC1 时处于 HS 模式
D101	CIO	所有 I/O 引脚和 OSC2	—	50	pF	满足交流时序规范
D102	CB	SCL 和 SDA	—	400	pF	I <sup>2</sup> C™ 规范

注 1: 负电流定义为引脚的拉电流。

2: 关于相应引脚的可承受电压限制请参见表 11-1。

# PIC18F85J11 系列

表 26-1: 存储器编程要求

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）				
参数 编号	符号	特性	最小值	典型值 †	最大值	单位	条件
D130	EP	闪存程序存储器					
		芯片的耐擦写能力	100	1k	—	E/W	-40°C 至 +85°C
D131	VPR	用于读操作的 VDD	V <sub>MIN</sub>	—	3.6	V	V <sub>MIN</sub> = 最小工作电压
D132	VPEW	用于自定时擦除或写操作的电压					
		VDD	2.35	—	3.6	V	ENVREG 连接到 VDD
		VDDCORE	2.25	—	2.7	V	ENVREG 连接到 VSS
D133A	TiW	自定时写周期	—	2.8	—	ms	
D133B	TiE	自定时块擦除周期	—	33	—	ms	
D134	TRETD	保存时间	20	—	—	年	假设没有违反其他规范
D135	IDDP	编程期间的供电电流	—	3	7	mA	
D1xxx	TWE	每个擦除周期中的写时间	—	—	1		每一个物理字地址

† 除非另外说明，否则“典型值”栏中的数据均为 3.3V、25°C 下的值。这些参数仅供设计参考，未经测试。

**表 26-2: 比较器规范**

工作条件: $3.0V \leq V_{DD} \leq 3.6V$ , $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ (除非另外说明)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
D300	VIOFF	输入失调电压	—	$\pm 5.0$	$\pm 25$	mV	
D301	VICM	输入共模电压	0	—	$AV_{DD} - 1.5$	V	
D302	CMRR	共模抑制比	55	—	—	dB	
D303	TRESP	响应时间 <sup>(1)</sup>	—	150	400	ns	
D304	TMC2OV	比较器模式改变到输出有效的时间	—	—	10	$\mu s$	
D305	VIRV	内部参考电压	—	1.2	—	V	

注 1: 响应时间是在比较器的一个输入端施加电压  $(AV_{DD} - 1.5)/2$ , 而另一个输入端从  $V_{SS}$  跳变到  $V_{DD}$  时测得的。

**表 26-3: 参考电压规范**

工作条件: $3.0V \leq V_{DD} \leq 3.6V$ , $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ (除非另外说明)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
D310	VRES	分辨率	$V_{DD}/24$	—	$V_{DD}/32$	LSb	
D311	VRAA	绝对精度	—	—	1/2	LSb	
D312	VRUR	单位电阻值 (R)	—	2k	—	$\Omega$	
D313	TSET	稳定时间 <sup>(1)</sup>	—	—	10	$\mu s$	

注 1: 稳定时间是在  $CVRR = 1$  并且  $CVR<3:0>$  位从 0000 跳变到 1111 时测得的。

**表 26-4: 内部稳压器规范**

工作条件: $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ (除非另外说明)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
	VRGOUT	稳压器输出电压	—	2.5	—	V	
	CEFC	外部滤波器电容值	4.7	10	—	$\mu F$	电容的等效串联阻抗必须很低 ( $<5\Omega$ )

# PIC18F85J11 系列

## 26.4 交流（时序）特性

### 26.4.1 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
3. Tcc:ST（仅用于 I<sup>2</sup>C 规范）
4. Ts（仅用于 I<sup>2</sup>C 规范）

T		T	
F	频率	T	时间

小写字母（pp）及其含义：

pp		osc	OSC1
cc	CCP1	rd	$\overline{RD}$
ck	CLKO	rw	$\overline{RD}$ 或 $\overline{WR}$
cs	$\overline{CS}$	sc	SCK
di	SDI	ss	$\overline{SS}$
do	SDO	t0	T0CKI
dt	数据输入	t1	T13CKI
io	I/O 端口	wr	$\overline{WR}$
mc	MCLR		

大写字母及其含义：

S		P	周期
F	下降	R	上升
H	高	V	有效
I	无效（高阻态）	Z	高阻态
L	低		
仅用于 I <sup>2</sup> C 模式		High	高
AA	输出通道	Low	低
BUF	总线空闲		

Tcc:ST（仅用于 I<sup>2</sup>C 规范）

CC		SU	建立
HD	保持		
ST		STO	停止条件
DAT	数据输入保持		
STA	启动条件		

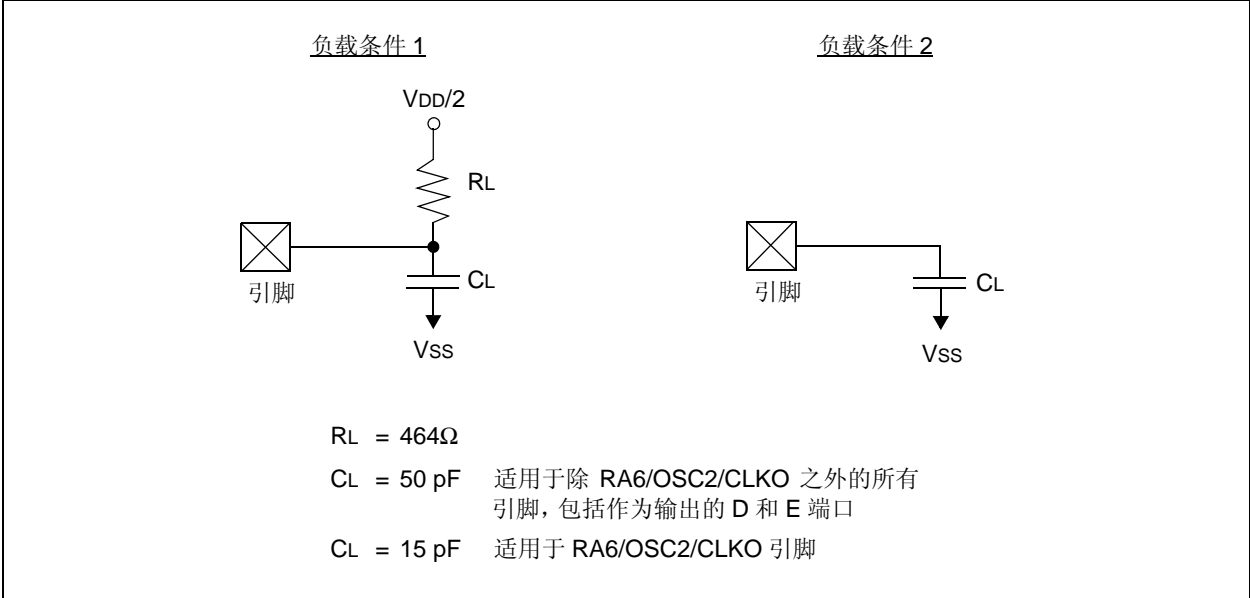
26.4.2 时序条件

除非另外说明，表 26-5 中指定的温度和电压适用于所有时序规范。图 26-3 规定了时序规范的负载条件。

表 26-5: 温度和电压规范——交流

交流特性	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）
	工作电压 $V_{DD}$ 范围如第 26.1 节和第 26.3 节中所示。

图 26-3: 器件时序规范的负载条件



# PIC18F85J11 系列

## 26.4.3 时序图和规范

图 26-4: 外部时钟时序（除 PLL 之外的所有模式）

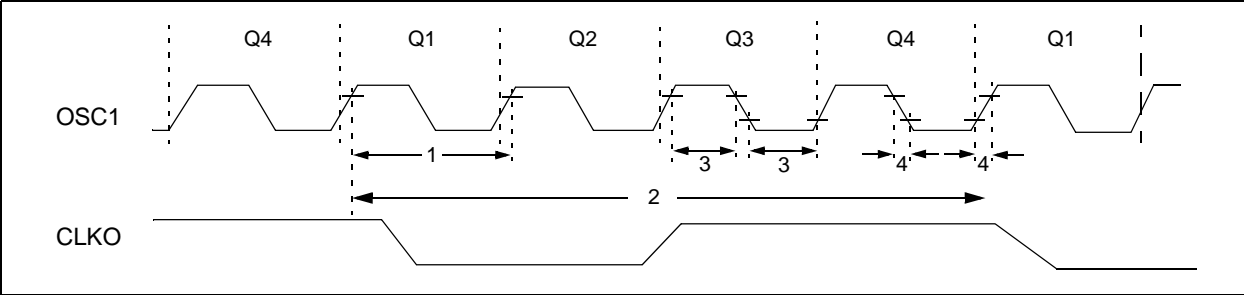


表 26-6: 外部时钟时序要求

参数编号	符号	特性	最小值	最大值	单位	条件
1A	FOSC	外部 CLKI 频率 <sup>(1)</sup> 振荡器频率 <sup>(1)</sup>	DC	40	MHz	ECPLL 振荡器模式 HSPLL 振荡器模式
1	TOSC	外部 CLKI 周期 <sup>(1)</sup> 振荡器周期 <sup>(1)</sup>	25	—	ns	EC 振荡器模式
			25	250	ns	HS 振荡器模式
2	Tcy	指令周期 <sup>(1)</sup>	100	—	ns	Tcy = 4/FOSC, 工业级
3	TosL, TosH	外部时钟输入 (OSC1) 的 高电平或低电平时间	10	—	ns	EC 振荡器模式
4	TosR, TosF	外部时钟输入 (OSC1) 的 上升或下降时间	—	7.5	ns	EC 振荡器模式

注 1: 对于除 PLL 的所有配置来说, 指令周期 (Tcy) 等于输入振荡器时基周期的 4 倍。所有指定值均为在特定振荡器模式下, 器件在标准工作条件下执行代码时获得的特征数据。超过规定值可能导致振荡器运行不稳定和 / 或电流消耗超出预期值。所有器件在测试 “最小值” 时, 都在 OSC1/CLKI 引脚连接了外部时钟。当使用了外部时钟输入时, 所有器件的 “最大” 周期时间限制为 “DC” (没有时钟)。

表 26-7: PLL 时钟时序规范 (VDD = 2.15V 至 3.6V)

参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
F10	FOSC	振荡器频率范围	4	—	10	MHz	仅 HS 模式
F11	FSys	片上 VCO 系统频率	16	—	40	MHz	仅 HS 模式
F12	t <sub>rc</sub>	PLL 起振时间 (锁定时间)	—	—	2	ms	
F13	ΔCLK	CLKO 稳定性 (抗抖动性能)	-2	—	+2	%	

† 除非另外说明, 否则“典型值”栏中的数据均为 3.3V、25°C 下的值。这些参数仅供设计参考, 未经测试。

表 26-8: 内部 RC 精度 (INTOSC 和 INTRC 时钟源)

PIC18F85J11 系列 (工业级)		标准工作条件（除非另外说明） 工作温度-40°C ≤ Ta ≤ +85°C（工业级）					
参数 编号	器件	最小值	典型值	最大值	单位	条件	
	频率为 8 MHz、4 MHz、2 MHz、1 MHz、500 kHz、250 kHz、125 kHz 和 31 kHz 时的 INTOSC 精度 <sup>(1)</sup>						
	所有器件	-2	+/-1	2	%	+25°C	VDD = 2.7-3.3V
		-5	—	5	%	-10°C 至 +85°C	VDD = 2.0-3.3V
		-10	+/-1	10	%	-40°C 至 +85°C	VDD = 2.0-3.3V
	频率为 31 kHz 时的 INTRC 精度 <sup>(1)</sup>						
	所有器件	26.562	—	35.938	kHz	-40°C 至 +85°C	VDD = 2.0-3.3V

注 1: 31 kHz 时钟的精度规范由在给定时间内所提供的时钟源来确定。当 INTSRC (OSCTUNE<7>) 为 1 时, 使用 INTOSC 精度规范。当 INTSRC 为 0 时, 使用 INTRC 精度规范。

# PIC18F85J11 系列

图 26-5: CLKO 和 I/O 时序

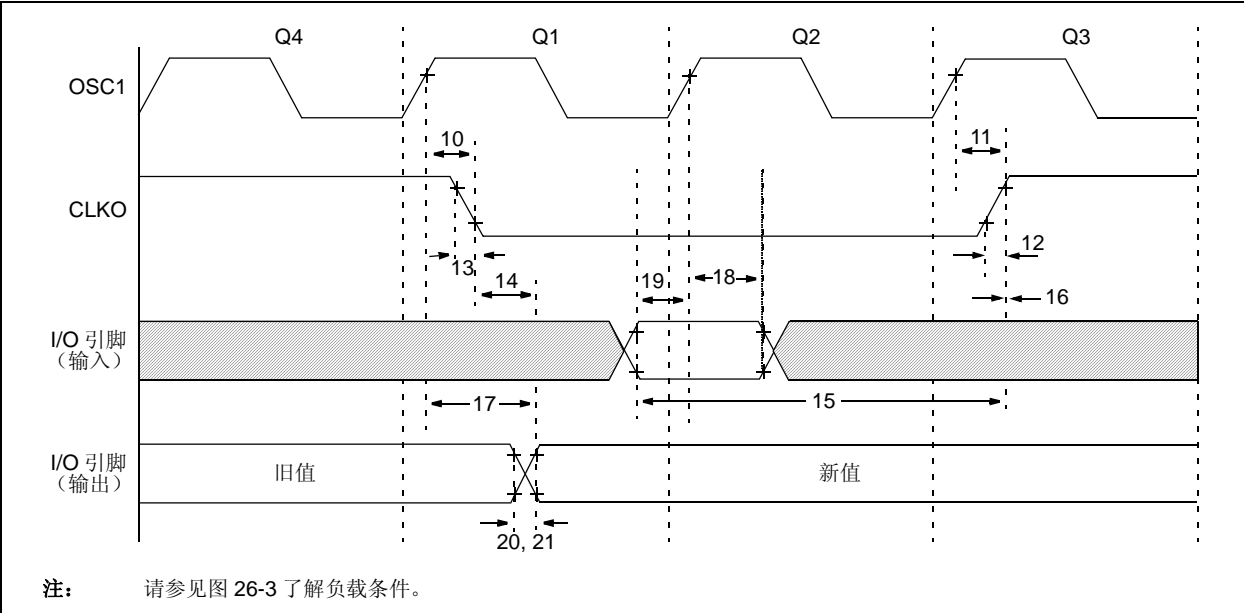


表 26-9: CLKO 和 I/O 时序要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
10	TosH2ckL	OSC1↑ 到 CLKO ↓	—	75	200	ns	(注 1)
11	TosH2ckH	OSC1↑ 到 CLKO ↓	—	75	200	ns	(注 1)
12	TckR	CLKO 上升时间	—	15	30	ns	(注 1)
13	TckF	CLKO 下降时间	—	15	30	ns	(注 1)
14	TckL2ioV	CLKO ↓ 至端口输出有效的时间	—	—	$0.5 T_{CY} + 20$	ns	
15	TioV2ckH	在 CLKO ↑ 之前端口输入有效的时间	$0.25 T_{CY} + 25$	—	—	ns	
16	TckH2ioI	在 CLKO ↑ 之后端口输入保持的时间	0	—	—	ns	
17	TosH2ioV	OSC1 ↑ (Q1 周期) 至端口输出有效的时间	—	50	150	ns	
18	TosH2ioI	OSC1 ↑ (Q2 周期) 至端口输入无效的时间 (I/O 输入保持时间)	100	—	—	ns	
19	TioV2osH	端口输入有效至 OSC1↑ 的时间 (I/O 输入建立时间)	0	—	—	ns	
20	TioR	端口输出上升时间	—	—	6	ns	
21	TioF	端口输出下降时间	—	—	5	ns	
22†	TINP	INTx 引脚高电平或低电平时间	$T_{CY}$	—	—	ns	
23†	TRBP	RB<7:4> 电平变化 INTx 高电平或低电平时间	$T_{CY}$	—	—	ns	

† 这些参数是与任何内部时钟边沿无关的异步事件。

注 1: 测量是在 EC 模式下进行的, 其中 CLKO 输出为  $4 \times T_{OSC}$ 。



图 26-6: 程序存储器取指时序图 (8 位)

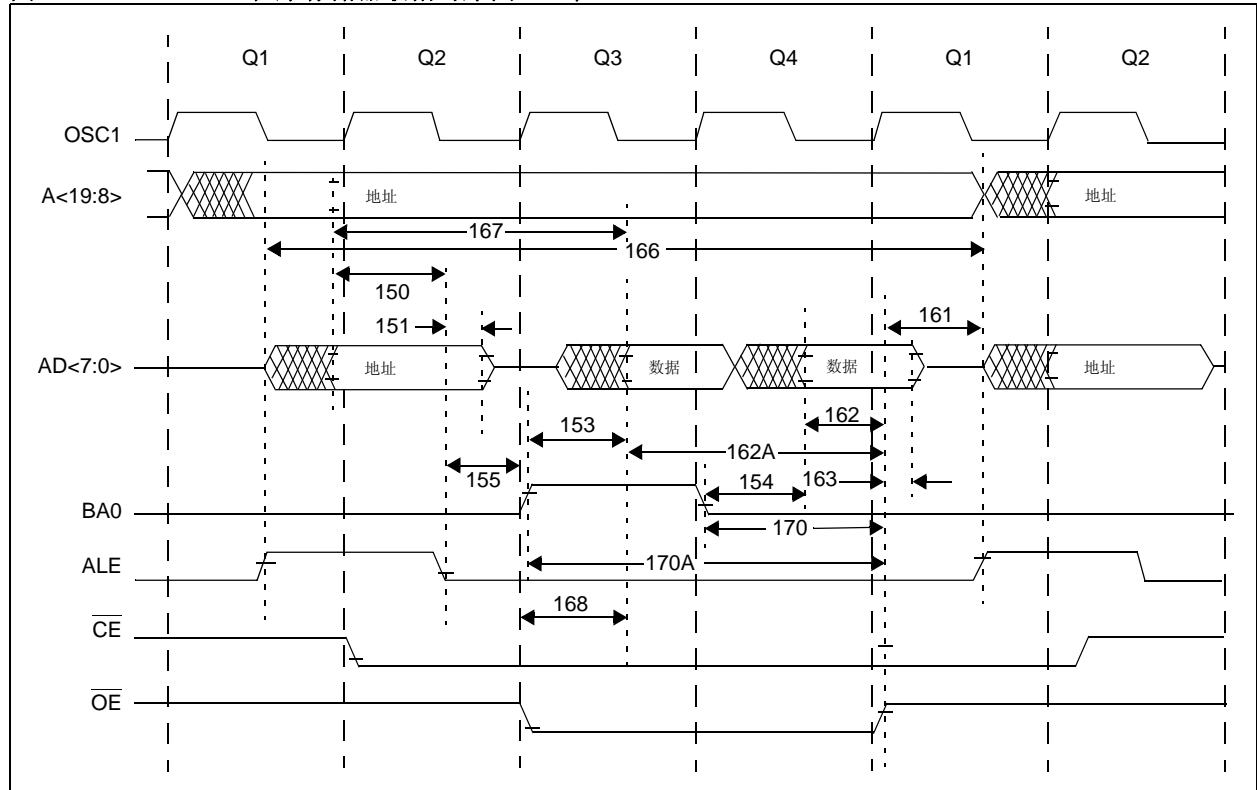


表 26-10: 程序存储器取指时序要求 (8 位)

参数编号	符号	特性	最小值	典型值	最大值	单位
150	TadV2alL	地址输出有效至 ALE ↓ 的时间 (地址建立时间)	0.25 Tcy - 10	—	—	ns
151	TalL2adl	ALE ↓ 至地址输出无效的时间 (地址保持时间)	5	—	—	ns
153	BA01	BA0 ↑ 至最高数据位有效的的时间	0.125 Tcy	—	—	ns
154	BA02	BA0 ↓ 至最低数据位有效的的时间	0.125 Tcy	—	—	ns
155	TalL2oeL	ALE ↓ 至 OE ↓ 的时间	0.125 Tcy	—	—	ns
161	ToeH2adD	OE ↑ 至 A/D 驱动的时间	0.125 Tcy - 5	—	—	ns
162	TadV2oeH	在 OE ↑ 之前最低数据位有效的的时间 (数据建立时间)	20	—	—	ns
162A	TadV2oeH	在 OE ↑ 之前最高数据位有效的的时间 (数据建立时间)	0.25 Tcy + 20	—	—	ns
163	ToeH2adl	OE ↑ 至数据输入无效的时间 (数据保持时间)	0	—	—	ns
166	TalH2alH	ALE ↑ 至 ALE ↑ 的时间 (周期)	—	Tcy	—	ns
167	TACC	地址有效至数据有效的的时间	0.5 Tcy - 10	—	—	ns
168	Toe	OE ↓ 至数据有效的的时间	—	—	0.125 Tcy + 5	ns
170	TubH2oeH	在 OE ↑ 之前 BA0 = 0 有效的的时间	0.25 Tcy	—	—	ns
170A	TubL2oeH	在 OE ↑ 之前 BA0 = 1 有效的的时间	0.5 Tcy	—	—	ns

# PIC18F85J11 系列

图 26-7: 程序存储器读时序图

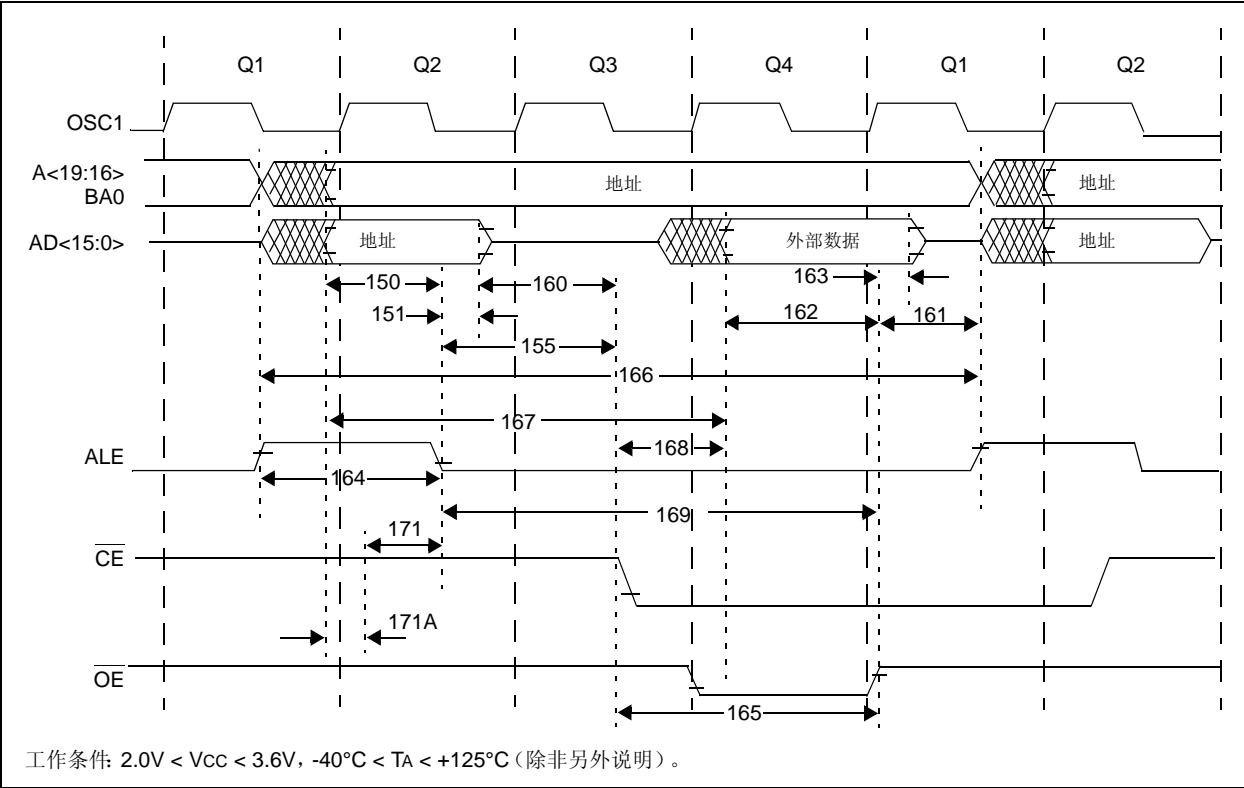


表 26-11: CLKO 和 I/O 时序要求

参数编号	符号	特性	最小值	典型值	最大值	单位
150	TadV2a1L	地址输出有效至 ALE↓ 的时间 (地址建立时间)	0.25 Tcy - 10	—	—	ns
151	TalL2adl	ALE↓ 至地址输出无效的时间 (地址保持时间)	5	—	—	ns
155	TalL2oeL	ALE↓ 至 $\overline{OE}$ ↓ 的时间	10	0.125 Tcy	—	ns
160	TadZ2oeL	AD 高阻态至 $\overline{OE}$ ↓ 的时间 (总线释放给 $\overline{OE}$ )	0	—	—	ns
161	ToeH2adD	$\overline{OE}$ ↑ 至 AD 驱动的时间	0.125 Tcy - 5	—	—	ns
162	TadV2oeH	在 $\overline{OE}$ ↑ 之前最低数据位有效的时间 (数据建立时间)	20	—	—	ns
163	ToeH2adl	$\overline{OE}$ ↑ 至数据输入无效的时间 (数据保持时间)	0	—	—	ns
164	TalH2a1L	ALE 脉冲宽度	—	0.25 Tcy	—	ns
165	ToeL2oeH	$\overline{OE}$ 脉冲宽度	0.5 Tcy - 5	0.5 Tcy	—	ns
166	TalH2a1H	ALE ↑ 至 ALE ↑ 的时间 (周期)	—	Tcy	—	ns
167	Tacc	地址有效至数据有效的时间	0.75 Tcy - 25	—	—	ns
168	Toe	$\overline{OE}$ ↓ 至数据有效的时间	—	—	0.5 Tcy - 25	ns
169	TalL2oeH	ALE ↓ 至 $\overline{OE}$ ↑ 的时间	0.625 Tcy - 10	—	0.625 Tcy + 10	ns
171	TalH2csL	芯片使能有效至 ALE ↓ 的时间	0.25 Tcy - 20	—	—	ns
171A	TubL2oeH	AD 有效至芯片使能有效的时间	—	—	10	ns

图 26-8: 程序存储器写时序图

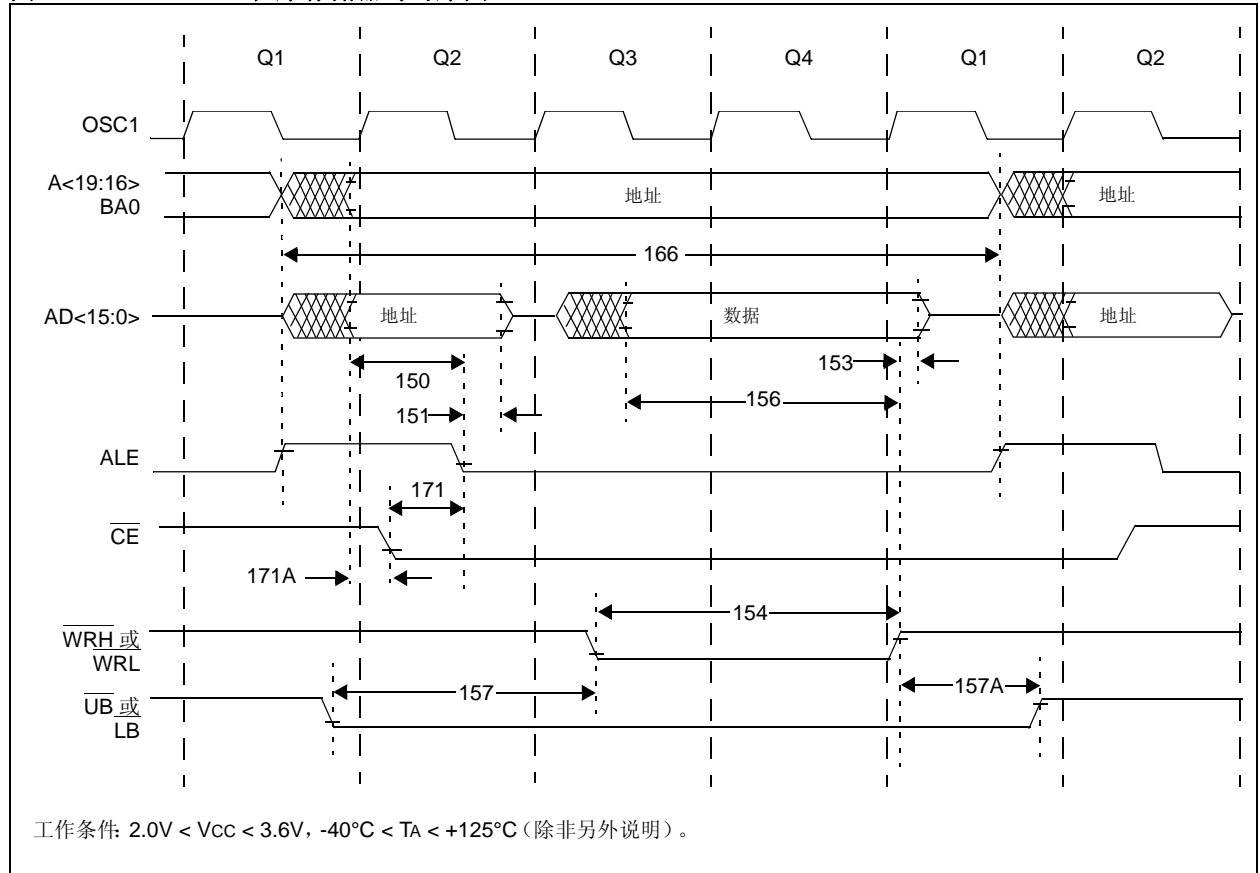


表 26-12: 程序存储器写时序要求

参数编号	符号	特性	最小值	典型值	最大值	单位
150	TadV2alL	地址输出有效至 ALE ↓ 的时间 (地址建立时间)	$0.25 T_{CY} - 10$	—	—	ns
151	TalL2adl	ALE ↓ 至地址输出无效的时间 (地址保持时间)	5	—	—	ns
153	TwrH2adl	$\overline{WRn} \uparrow$ 至数据输出无效的时间 (数据保持时间)	5	—	—	ns
154	TwrL	$\overline{WRn}$ 脉冲宽度	$0.5 T_{CY} - 5$	$0.5 T_{CY}$	—	ns
156	TadV2wrH	在 $\overline{WRn} \uparrow$ 之前数据有效的的时间 (数据建立时间)	$0.5 T_{CY} - 10$	—	—	ns
157	TbsV2wrL	在 $\overline{WRn} \downarrow$ 之前字节选择有效的的时间 (字节选择建立时间)	$0.25 T_{CY}$	—	—	ns
157A	TwrH2bsl	$\overline{WRn} \uparrow$ 至字节选择无效的时间 (字节选择保持时间)	$0.125 T_{CY} - 5$	—	—	ns
166	TalH2alH	ALE ↑ 至 ALE ↑ 的时间 (周期)	—	$T_{CY}$	—	ns
171	TalH2csL	芯片使能有效至 ALE ↓ 的时间	$0.25 T_{CY} - 20$	—	—	ns
171A	TubL2oeH	AD 有效至芯片使能有效的时间	—	—	10	ns

# PIC18F85J11 系列

图 26-9: 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

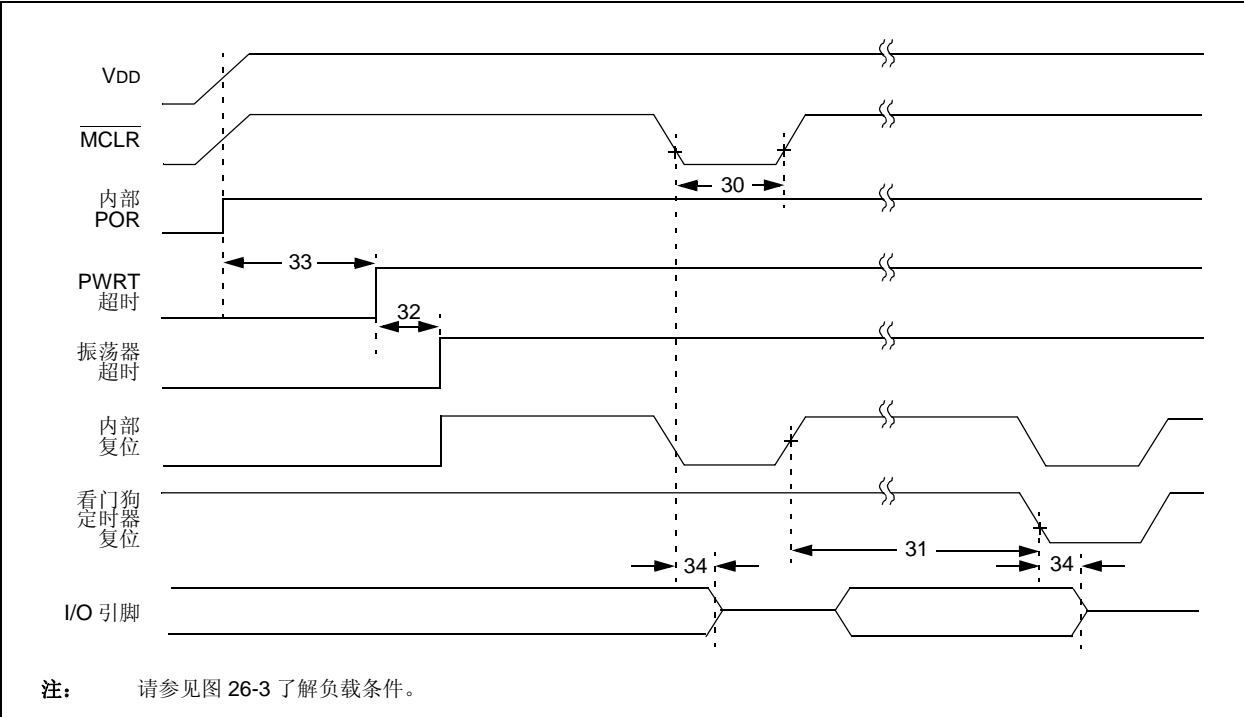


表 26-13: 复位、看门狗定时器、振荡器起振定时器、上电延时定时器和欠压复位要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
30	TMCL	MCLR 脉冲宽度（低电平）	2 Tcy	10 Tcy	—		(注 1)
31	TWDT	看门狗定时器超时周期（无后分频器）	3.4	4.0	4.6	ms	
32	TOST	振荡器起振定时器周期	1024 TOSC	—	1024 TOSC		TOSC = OSC1 周期
33	TPWRT	上电延时定时器周期	45.8	65.5	85.2	ms	
34	TIOZ	自 MCLR 低电平或看门狗定时器复位起 I/O 处于高阻态的时间	—	2	—	μs	
38	TCSD	CPU 启动时间	—	10	—	μs	使能稳压器并进入休眠模式
			—	200	—	μs	
39	TIOBST	INTOSC 电路稳定时间	—	1	—	μs	

注 1: 为了确保器件已复位，MCLR 必须为低电平且保持时间为 2 Tcy 或 400 μs 中较小的一个。

图 26-10: **TIMER0 和 TIMER1 外部时钟时序**

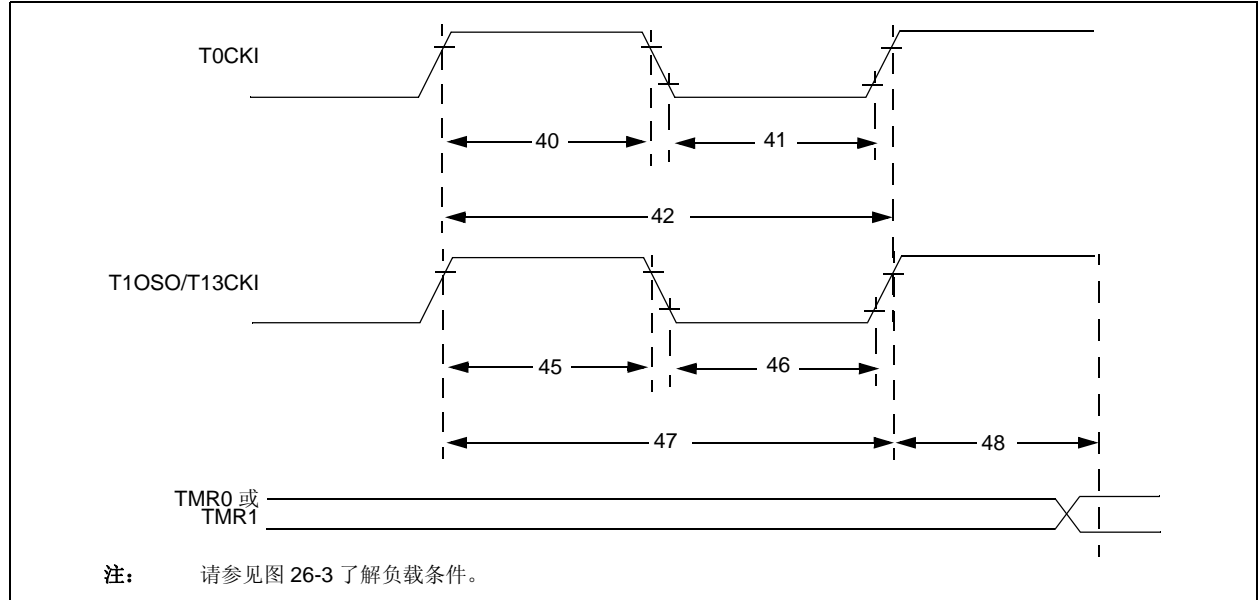


表 26-14: **TIMER0 和 TIMER1 外部时钟要求**

参数编号	符号	特性		最小值	最大值	单位	条件
40	Tt0H	T0CKI 高电平脉冲宽度	无预分频器	$0.5 T_{CY} + 20$	—	ns	
			有预分频器	10	—	ns	
41	Tt0L	T0CKI 低电平脉冲宽度	无预分频器	$0.5 T_{CY} + 20$	—	ns	
			有预分频器	10	—	ns	
42	Tt0P	T0CKI 周期	无预分频器	$T_{CY} + 10$	—	ns	N = 预分频值 (1、2、4,..., 256)
			有预分频器	取较大值: 20 ns 或 $(T_{CY} + 40)/N$	—	ns	
45	Tt1H	T13CKI 高电平时间	同步, 无预分频器	$0.5 T_{CY} + 20$	—	ns	
			同步, 有预分频器	10	—	ns	
			异步	30	—	ns	
46	Tt1L	T13CKI 低电平时间	同步, 无预分频器	$.5 T_{CY} + 5$	—	ns	
			同步, 有预分频器	10	—	ns	
			异步	30	—	ns	
47	Tt1P	T13CKI 输入周期	同步	取较大值: 20 ns 或 $(T_{CY} + 40)/N$	—	ns	N = 预分频值 (1、2、4 或 8)
			异步	60	—	ns	
	Ft1	T13CKI 振荡器输入频率范围		DC	50	kHz	
48	Tcke2TMRI	从外部 T13CKI 时钟边沿到定时器递增的延时		$2 T_{OSC}$	$7 T_{OSC}$	—	

# PIC18F85J11 系列

图 26-11: 捕捉 / 比较 / PWM 时序 (CCP1 和 CCP2 模块)

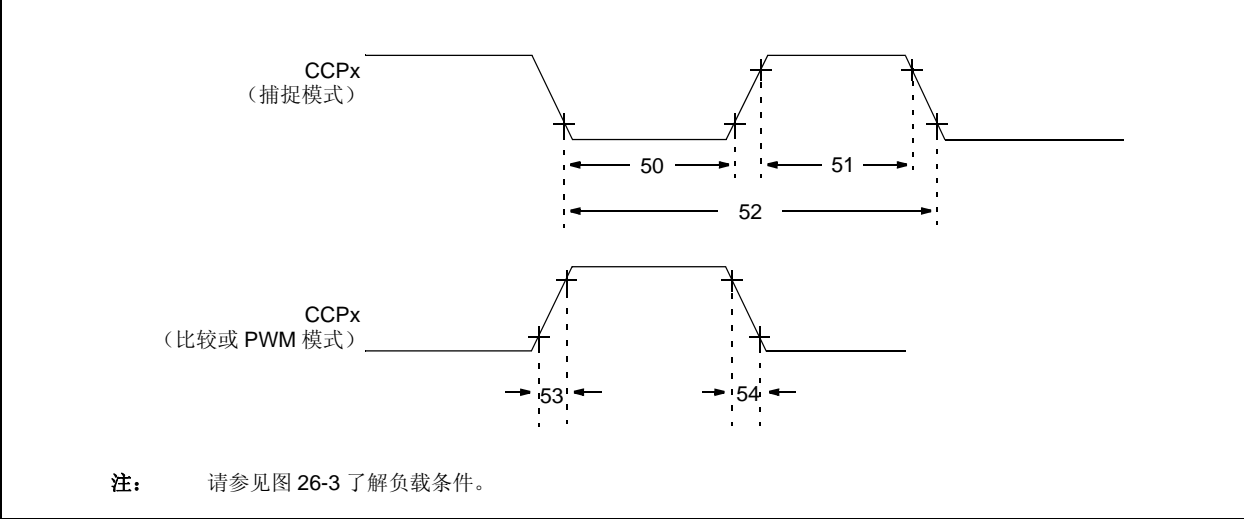


表 26-15: 捕捉 / 比较 / PWM 要求 (CCP1 和 CCP2 模块)

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

图 26-12: SPI 主模式时序示例 (CKE = 0)

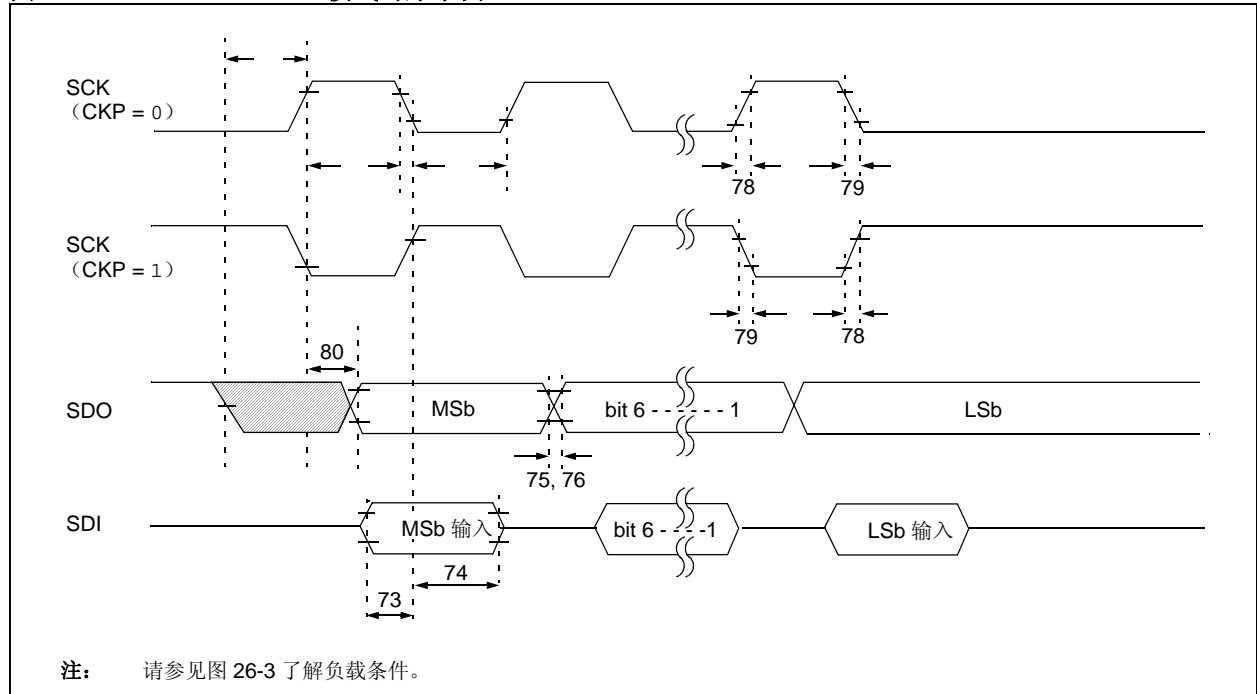


表 26-16: SPI 模式要求示例 (主模式, CKE = 0)

参数编号	符号	特性	最小值	最大值	单位	条件
73	TdIV2scH, TdIV2scL	SDI 数据输入到 SCK 边沿的建立时间	20	—	ns	
73A	Tb2B	字节 1 的最后一个时钟边沿到字节 2 的第一个时钟边沿的时间	1.5 Tcy + 40	—	ns	(注 1)
74	Tsch2diL, TscL2diL	SDI 数据输入到 SCK 边沿的保持时间	40	—	ns	
75	TdoR	SDO 数据输出上升时间	—	25	ns	
76	TdoF	SDO 数据输出下降时间	—	25	ns	
78	TscR	SCK 输出上升时间 (主模式)	—	25	ns	
79	TscF	SCK 输出下降时间 (主模式)	—	25	ns	
80	Tsch2doV, TscL2doV	SCK 边沿后 SDO 数据输出有效的的时间	—	50	ns	

注 1: 仅当使用参数 71A 和 72A 时。

# PIC18F85J11 系列

图 26-13: SPI 主模式时序示例 (CKE = 1)

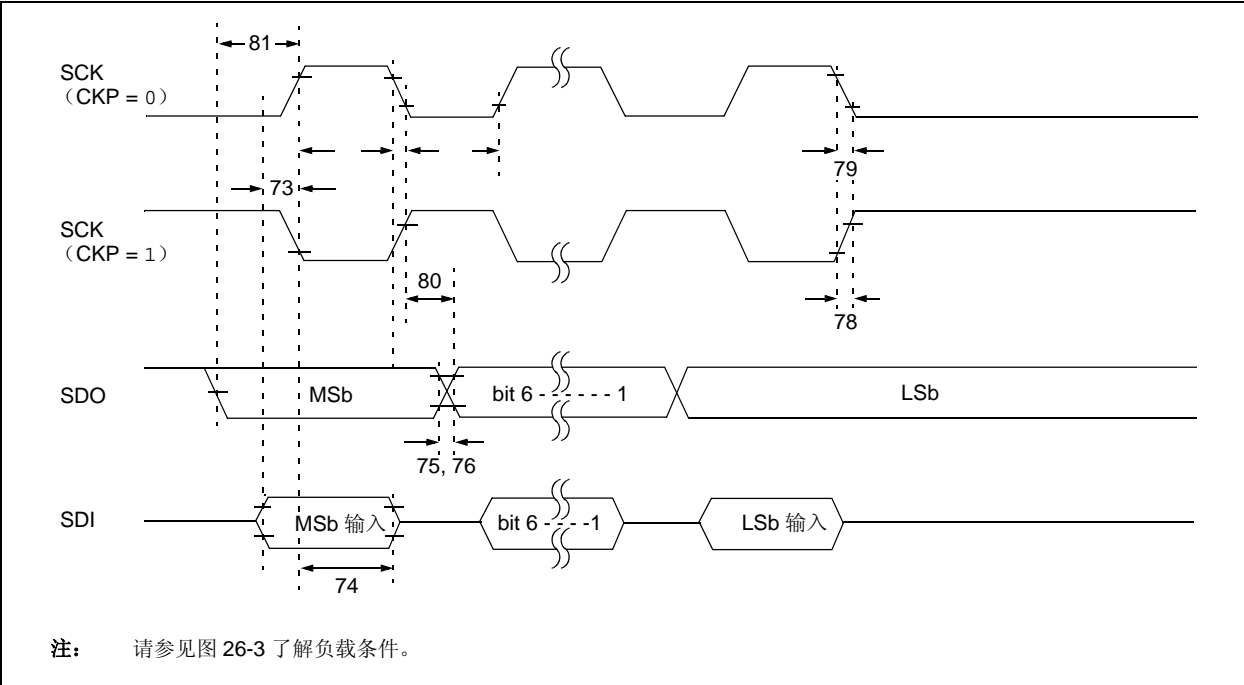


表 26-17: SPI 模式要求示例 (主模式, CKE = 1)

参数编号	符号	特性	最小值	最大值	单位	条件
73	TdIV2scH, TdIV2scL	SDI 数据输入到 SCK 边沿的建立时间	20	—	ns	
73A	Tb2B	字节 1 的最后一个时钟边沿到字节 2 的第一个时钟边沿的时间	1.5 Tcy + 40	—	ns	(注 2)
74	Tsch2diL, TscL2diL	SDI 数据输入到 SCK 边沿的保持时间	40	—	ns	
75	TdoR	SDO 数据输出上升时间	—	25	ns	
76	TdoF	SDO 数据输出下降时间	—	25	ns	
78	TscR	SCK 输出上升时间 (主模式)	—	25	ns	
79	TscF	SCK 输出下降时间 (主模式)	—	25	ns	
80	Tsch2doV, TscL2doV	SCK 边沿后 SDO 数据输出有效的时间	—	50	ns	
81	TdoV2scH, TdoV2scL	SDO 数据输出到 SCK 边沿的建立时间	Tcy	—	ns	

注 1: 要求使用参数 73A。  
2: 仅当使用参数 71A 和 72A 时。



图 26-14: SPI 从模式时序示例 (CKE = 0)

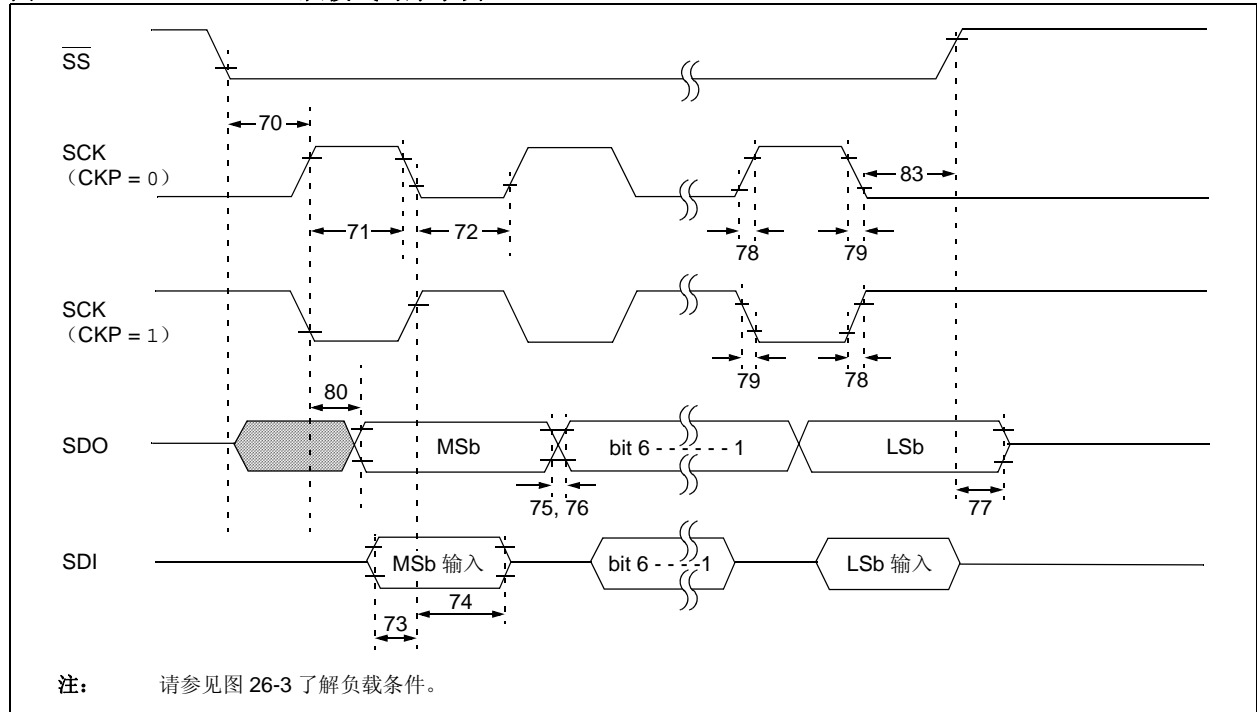


表 26-18: SPI 模式要求示例 (从模式时序, CKE = 0)

参数编号	符号	特性	最小值	最大值	单位	条件
70	TssL2scH, TssL2scL	$\overline{SS} \downarrow$ 至 SCK $\downarrow$ 或 SCK $\uparrow$ 输入的时间	3 Tcy	—	ns	
70A	TssL2WB	$\overline{SS}$ 至写 SSPBUF 的时间	3 Tcy	—	ns	
71	Tsch	SCK 输入高电平时间	1.25 Tcy + 30	—	ns	
71A		(从模式)	连续	—	ns	(注 1)
72	Tscl	SCK 输入低电平时间	1.25 Tcy + 30	—	ns	
72A		(从模式)	连续	—	ns	(注 1)
73	TdiV2scH, TdiV2scL	SDI 数据输入到 SCK 边沿的建立时间	20	—	ns	
73A	Tb2B	字节 1 的最后一个时钟边沿到字节 2 的第一个时钟边沿	1.5 Tcy + 40	—	ns	(注 2)
74	Tsch2diL, TscL2diL	SDI 数据输入到 SCK 边沿的保持时间	40	—	ns	
75	TdoR	SDO 数据输出上升时间	—	25	ns	
76	TdoF	SDO 数据输出下降时间	—	25	ns	
77	TssH2doZ	$\overline{SS} \uparrow$ 至 SDO 输出呈现高阻态的时间	10	50	ns	
78	TscR	SCK 输出上升时间 (主模式)	—	25	ns	
79	TscF	SCK 输出下降时间 (主模式)	—	25	ns	
80	Tsch2doV, TscL2doV	SCK 边沿后 SDO 数据输出有效的时间	—	50	ns	
83	Tsch2ssH, TscL2ssH	SCK 边沿后 $\overline{SS} \uparrow$ 的时间	1.5 Tcy + 40	—	ns	

注 1: 要求使用参数 73A。

注 2: 仅当使用参数 71A 和 72A 时。

# PIC18F85J11 系列

图 26-15: SPI 从模式时序示例 (CKE = 1)

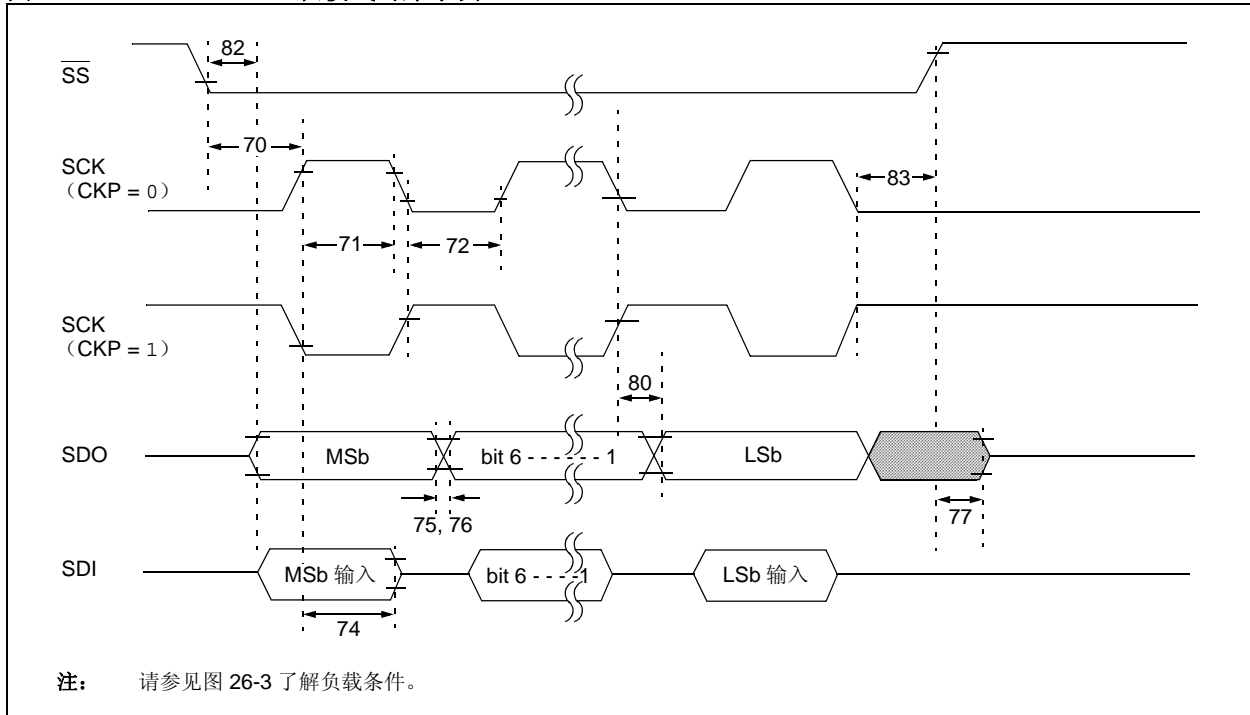


表 26-19: SPI 从模式要求示例 (CKE = 1)

参数编号	符号	特性	最小值	最大值	单位	条件
70	TssL2sCH, TssL2sCL	$\overline{SS} \downarrow$ 至 SCK $\downarrow$ 或 SCK $\uparrow$ 输入的时间	3 Tcy	—	ns	
70A	TssL2WB	$\overline{SS}$ 至写 SSPBUF 的时间	3 Tcy	—	ns	
71	Tsch	SCK 输入高电平时间	连续	1.25 Tcy + 30	ns	
71A		(从模式)	单字节	40	ns	(注 1)
72	Tscl	SCK 输入低电平时间	连续	1.25 Tcy + 30	ns	
72A		(从模式)	单字节	40	ns	(注 1)
73A	Tb2B	字节 1 的最后一个时钟边沿到字节 2 的第一个时钟边沿	1.5 Tcy + 40	—	ns	(注 2)
74	Tsch2diL, TscL2diL	SDI 数据输入到 SCK 边沿的保持时间	40	—	ns	
75	TdoR	SDO 数据输出上升时间	—	25	ns	
76	TdoF	SDO 数据输出下降时间	—	25	ns	
77	TssH2doZ	$\overline{SS} \uparrow$ 至 SDO 输出呈现高阻态的时间	10	50	ns	
78	TscR	SCK 输出上升时间 (主模式)	—	25	ns	
79	TscF	SCK 输出下降时间 (主模式)	—	25	ns	
80	Tsch2doV, TscL2doV	SCK 边沿后 SDO 数据输出有效的时间	—	50	ns	
82	TssL2doV	$\overline{SS} \downarrow$ 边沿后 SDO 数据输出有效的时间	—	50	ns	
83	Tsch2ssH, TscL2ssH	SCK 边沿后 $\overline{SS} \uparrow$ 的时间	1.5 Tcy + 40	—	ns	

注 1: 要求使用参数 73A。

注 2: 仅当使用参数 71A 和 72A 时。

图 26-16: I<sup>2</sup>C™ 总线启动 / 停止位时序

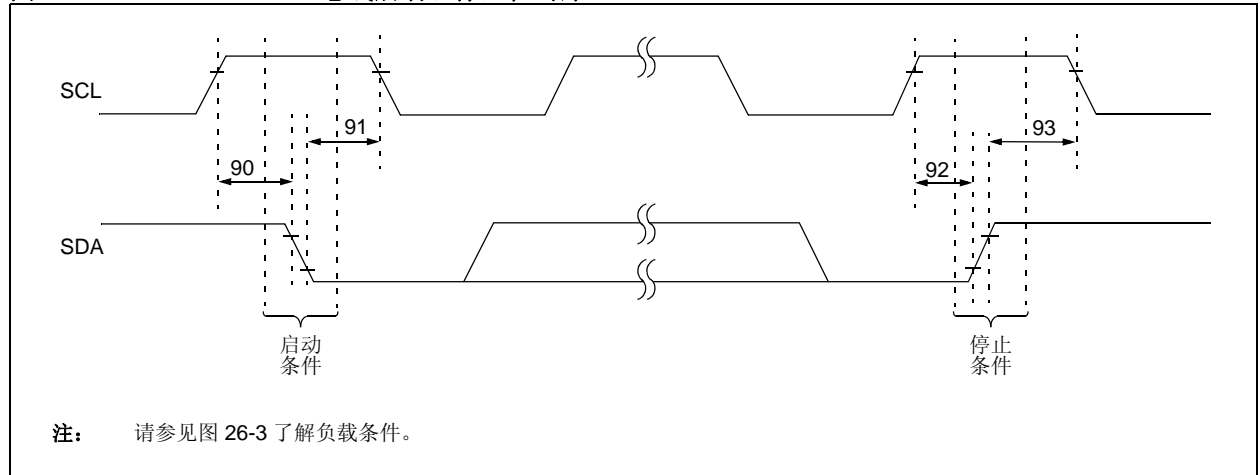


表 26-20: I<sup>2</sup>C 总线启动 / 停止位要求 (从模式)

参数编号	符号	特性		最小值	最大值	单位	条件
90	TSU:STA	启动条件 建立时间	100 kHz 模式	4700	—	ns	仅与重复启动条件相关
			400 kHz 模式	600	—		
91	THD:STA	启动条件 保持时间	100 kHz 模式	4000	—	ns	这个周期后产生第一个时钟脉冲
			400 kHz 模式	600	—		
92	TSU:STO	停止条件 建立时间	100 kHz 模式	4700	—	ns	
			400 kHz 模式	600	—		
93	THD:STO	停止条件 保持时间	100 kHz 模式	4000	—	ns	
			400 kHz 模式	600	—		

# PIC18F85J11 系列

图 26-17: I<sup>2</sup>C™ 总线数据时序

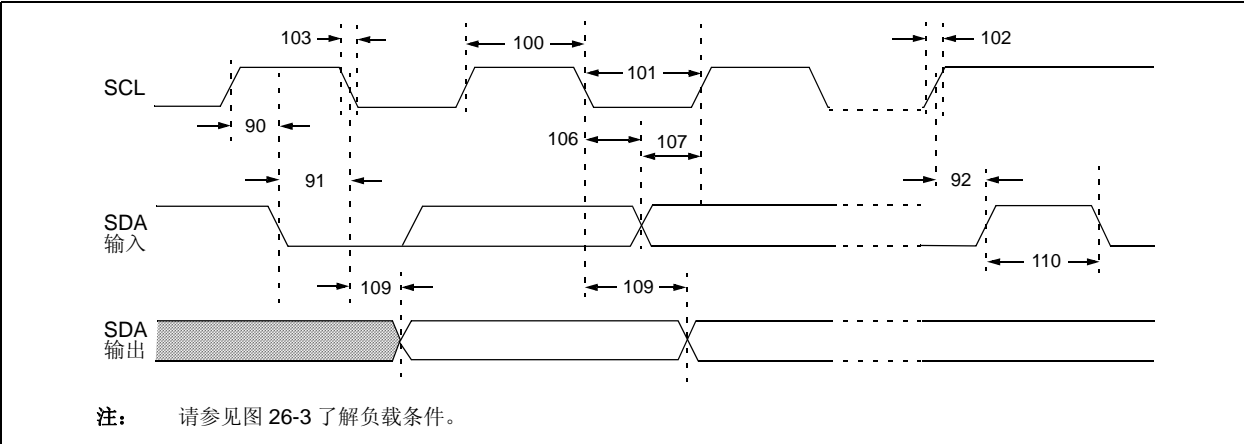


表 26-21: I<sup>2</sup>C™ 总线数据要求 (从模式)

参数编号	符号	特性	最小值	最大值	单位	条件
100	T <sub>HIGH</sub>	时钟高电平时间	100 kHz 模式	4.0	—	μs
			400 kHz 模式	0.6	—	μs
			MSSP 模块	1.5 T <sub>CY</sub>	—	
101	T <sub>LOW</sub>	时钟低电平时间	100 kHz 模式	4.7	—	μs
			400 kHz 模式	1.3	—	μs
			MSSP 模块	1.5 T <sub>CY</sub>	—	
102	T <sub>R</sub>	SDA 和 SCL 上升时间	100 kHz 模式	—	1000	ns
			400 kHz 模式	20 + 0.1 C <sub>B</sub>	300	ns C <sub>B</sub> 值在 10 到 400 pF 之间
103	T <sub>F</sub>	SDA 和 SCL 下降时间	100 kHz 模式	—	300	ns
			400 kHz 模式	20 + 0.1 C <sub>B</sub>	300	ns C <sub>B</sub> 值在 10 到 400 pF 之间
90	T <sub>SU:STA</sub>	启动条件建立时间	100 kHz 模式	4.7	—	μs 仅与重复启动条件相关
			400 kHz 模式	0.6	—	μs
91	T <sub>HD:STA</sub>	启动条件保持时间	100 kHz 模式	4.0	—	μs 这个周期后产生第一个时钟脉冲
			400 kHz 模式	0.6	—	μs
106	T <sub>HD:DAT</sub>	数据输入保持时间	100 kHz 模式	0	—	ns
			400 kHz 模式	0	0.9	μs
107	T <sub>SU:DAT</sub>	数据输入建立时间	100 kHz 模式	250	—	ns (注 2)
			400 kHz 模式	100	—	ns
92	T <sub>SU:STO</sub>	停止条件建立时间	100 kHz 模式	4.7	—	μs
			400 kHz 模式	0.6	—	μs
109	T <sub>AA</sub>	时钟输出有效时间	100 kHz 模式	—	3500	ns (注 1)
			400 kHz 模式	—	—	ns
110	T <sub>BUF</sub>	总线空闲时间	100 kHz 模式	4.7	—	μs 在启动一个新的传输前总线必须保持空闲的时间
			400 kHz 模式	1.3	—	μs
D102	C <sub>B</sub>	总线容性负载	—	400	pF	

- 注 1: 为避免意外产生启动或停止条件, 作为发送器的器件必须提供此内部最小延迟时间, 以覆盖 SCL 下降沿的未定义区域 (最小值 300 ns)。
- 2: 在标准模式的 I<sup>2</sup>C 总线系统中, 可使用快速模式的 I<sup>2</sup>C™ 总线器件, 但必须满足 T<sub>SU:DAT</sub> ≥ 250 ns。如果快速模式器件没有延长 SCL 信号的低电平时间, 则必然满足此条件。如果该器件延长了 SCL 信号的低电平时间, 必须将下一个数据位输出到 SDA 线, SCL 线被释放前, 根据标准模式 I<sup>2</sup>C 总线规范, T<sub>R max.</sub> + T<sub>SU:DAT</sub> = 1000 + 250 = 1250 ns。

图 26-18: MSSP I<sup>2</sup>C™ 总线启动 / 停止位时序波形

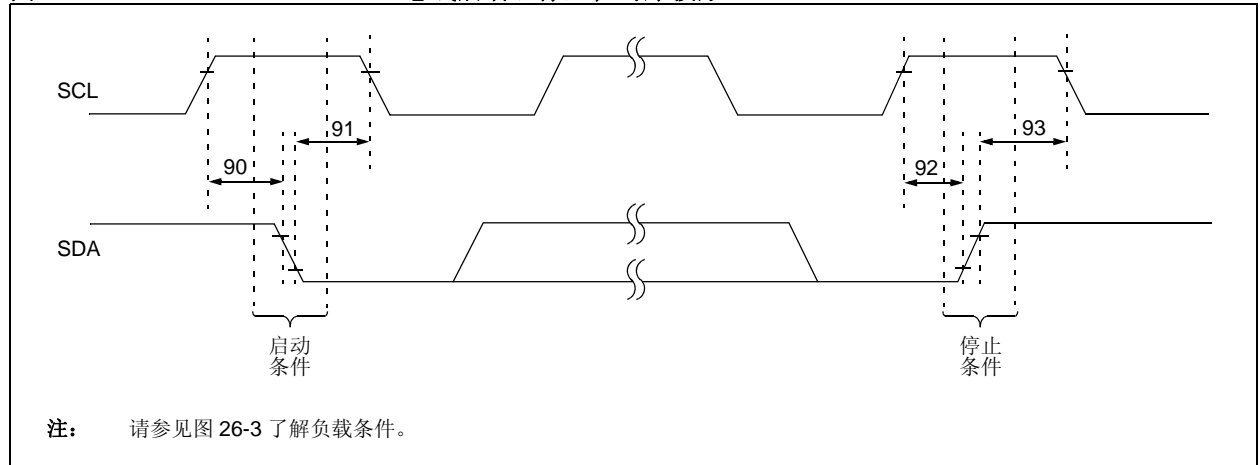
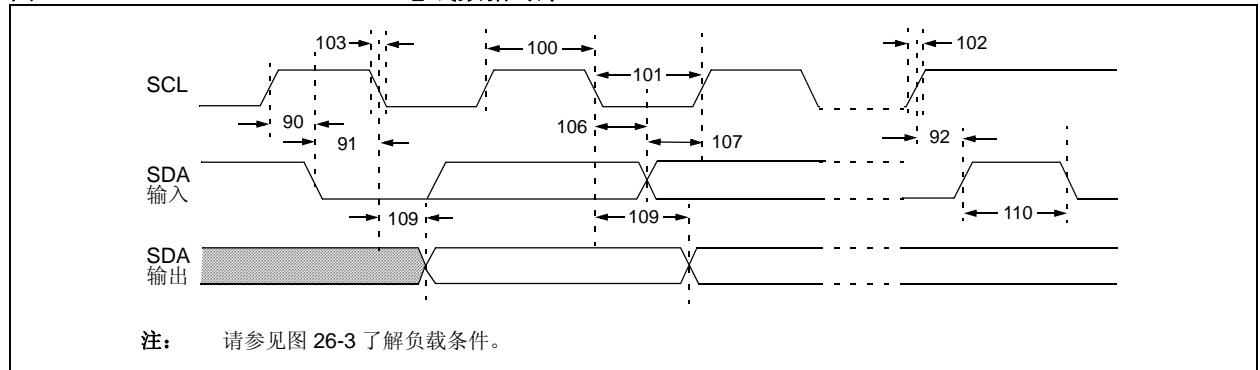


表 26-22: MSSP I<sup>2</sup>C™ 总线启动 / 停止位要求

参数编号	符号	特性	最小值	最大值	单位	条件
90	TSU:STA	启动条件建立时间	100 kHz 模式	$2(T_{osc})(BRG + 1)$	—	ns 仅与重复启动条件相关
			400 kHz 模式	$2(T_{osc})(BRG + 1)$	—	
			1 MHz 模式 (1,2)	$2(T_{osc})(BRG + 1)$	—	
91	THD:STA	启动条件保持时间	100 kHz 模式	$2(T_{osc})(BRG + 1)$	—	ns 这个周期后产生第一个时钟脉冲
			400 kHz 模式	$2(T_{osc})(BRG + 1)$	—	
			1 MHz 模式 (1,2)	$2(T_{osc})(BRG + 1)$	—	
92	TSU:STO	停止条件建立时间	100 kHz 模式	$2(T_{osc})(BRG + 1)$	—	ns
			400 kHz 模式	$2(T_{osc})(BRG + 1)$	—	
			1 MHz 模式 (1,2)	$2(T_{osc})(BRG + 1)$	—	
93	THD:STO	停止条件保持时间	100 kHz 模式	$2(T_{osc})(BRG + 1)$	—	ns
			400 kHz 模式	$2(T_{osc})(BRG + 1)$	—	
			1 MHz 模式 (1,2)	$2(T_{osc})(BRG + 1)$	—	

注 1: 对于所有 I<sup>2</sup>C™ 引脚, 最大引脚电容均为 10 pF。  
 2: 1 MHz I<sup>2</sup>C 至少需要 16 MHz Fosc。

图 26-19: MSSP I<sup>2</sup>C™ 总线数据时序



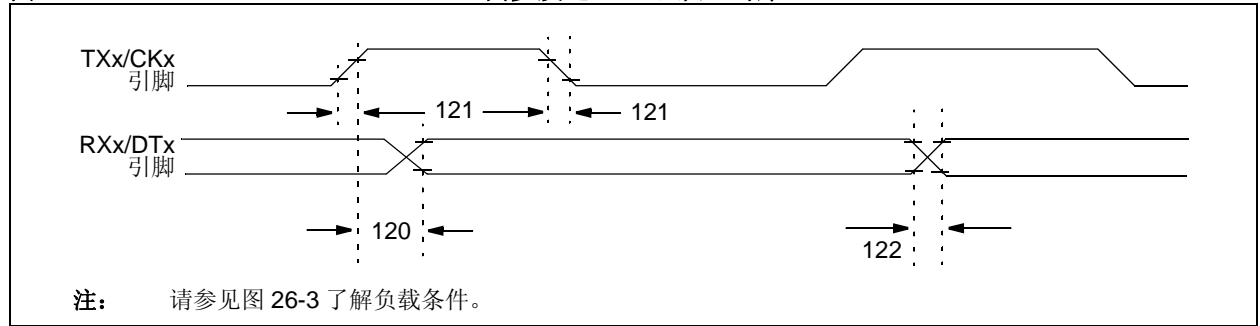
# PIC18F85J11 系列

表 26-23: MSSP I<sup>2</sup>C™ 总线数据要求

参数编号	符号	特性	最小值	最大值	单位	条件
100	T <sub>HIGH</sub>	时钟高电平时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	—
			400 kHz 模式	2(Tosc)(BRG + 1)	—	—
			1 MHz 模式 (1,2)	2(Tosc)(BRG + 1)	—	—
101	T <sub>LOW</sub>	时钟低电平时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	—
			400 kHz 模式	2(Tosc)(BRG + 1)	—	—
			1 MHz 模式 (1,2)	2(Tosc)(BRG + 1)	—	—
102	T <sub>R</sub>	SDA 和 SCL 上升时间	100 kHz 模式	—	1000	ns
			400 kHz 模式	20 + 0.1 C <sub>B</sub>	300	ns
			1 MHz 模式 (1,2)	—	300	ns
103	T <sub>F</sub>	SDA 和 SCL 下降时间	100 kHz 模式	—	300	ns
			400 kHz 模式	20 + 0.1 C <sub>B</sub>	300	ns
			1 MHz 模式 (1,2)	—	100	ns
90	T <sub>SU:STA</sub>	启动条件建立时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	—
			400 kHz 模式	2(Tosc)(BRG + 1)	—	—
			1 MHz 模式 (1,2)	2(Tosc)(BRG + 1)	—	—
91	T <sub>HD:STA</sub>	启动条件保持时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	—
			400 kHz 模式	2(Tosc)(BRG + 1)	—	—
			1 MHz 模式 (1,2)	2(Tosc)(BRG + 1)	—	—
106	T <sub>HD:DAT</sub>	数据输入保持时间	100 kHz 模式	0	—	ns
			400 kHz 模式	0	0.9	μs
			1 MHz 模式 (1,2)	—	—	ns
107	T <sub>SU:DAT</sub>	数据输入建立时间	100 kHz 模式	250	—	ns
			400 kHz 模式	100	—	ns
			1 MHz 模式 (1,2)	—	—	ns
92	T <sub>SU:STO</sub>	停止条件建立时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	—
			400 kHz 模式	2(Tosc)(BRG + 1)	—	—
			1 MHz 模式 (1,2)	2(Tosc)(BRG + 1)	—	—
109	T <sub>AA</sub>	时钟输出有效时间	100 kHz 模式	—	3500	ns
			400 kHz 模式	—	1000	ns
			1 MHz 模式 (1,2)	—	—	ns
110	T <sub>BUF</sub>	总线空闲时间	100 kHz 模式	4.7	—	μs
			400 kHz 模式	1.3	—	μs
			1 MHz 模式 (1,2)	—	—	μs
D102	C <sub>B</sub>	总线容性负载	—	400	pF	

- 注 1: 对于所有 I<sup>2</sup>C™ 引脚, 最大引脚电容均为 10 pF。
- 2: 1 MHz I<sup>2</sup>C 至少需要 16 MHz Fosc。
- 3: 在标准模式的 I<sup>2</sup>C 总线系统中, 可以使用快速模式的 I<sup>2</sup>C 总线器件, 但必须满足参数 107 ≥ 250 ns 的要求。如果快速模式器件没有延长 SCL 信号的低电平时间, 则必然满足此条件。如果该器件延长了 SCL 信号的低电平时间, 它必须将下一个数据位输出到 SDA 线, SCL 线被释放前, 参数 102 + 参数 107 = 1000 + 250 = 1250 ns (100 kHz 模式下)。

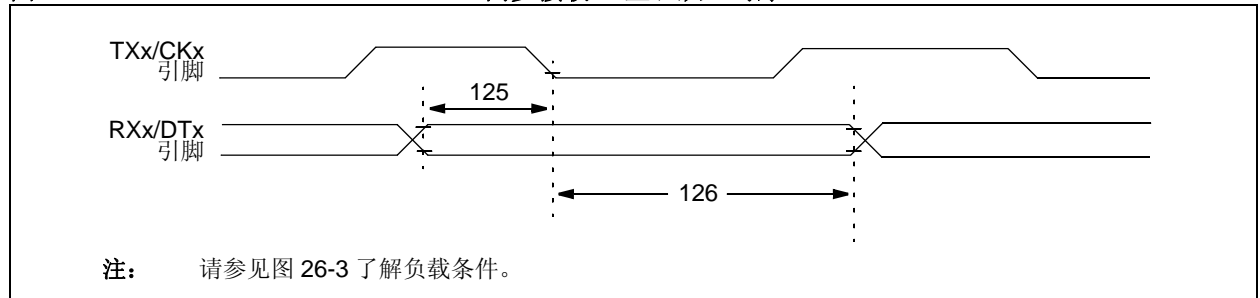
**图 26-20: EUSART/AUSART 同步发送（主 / 从）时序**



**表 26-24: EUSART/AUSART 同步发送要求**

参数编号	符号	特性	最小值	最大值	单位	条件
120	TckH2DTV	同步发送（主和从） 时钟高电平至数据输出有效的时间	—	40	ns	
121	TCKRF	时钟输出上升时间和下降时间（主模式）	—	20	ns	
122	TDTRF	数据输出上升时间和下降时间	—	20	ns	

**图 26-21: EUSART/AUSART 同步接收（主 / 从）时序**



**表 26-25: EUSART/AUSART 同步接收要求**

参数编号	符号	特性	最小值	最大值	单位	条件
125	TDTV2CKL	同步接收（主和从） 在 CKx ↓ 前数据保持时间（DTx 保持时间）	10	—	ns	
126	TCKL2DTL	在 CKx ↓ 后数据保持时间（DTx 保持时间）	15	—	ns	

# PIC18F85J11 系列

表 26-26: A/D 转换器特性: PIC18F85J11 系列 (工业级)

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
A01	NR	分辨率	—	—	10	位	
A03	EIL	积分线性误差	—	—	$< \pm 1$	LSb	$\Delta V_{REF} \geq 3.0V$
A04	EDL	微分线性误差	—	—	$< \pm 1$	LSb	$\Delta V_{REF} \geq 3.0V$
A06	E0FF	失调误差	—	—	$< \pm 3$	LSb	$\Delta V_{REF} \geq 3.0V$
A07	EGN	增益误差	—	—	$< \pm 3$	LSb	$\Delta V_{REF} \geq 3.0V$
A10	—	单调性	保证 <sup>(1)</sup>			—	$V_{SS} \leq V_{AIN} \leq V_{REF}$
A20	$\Delta V_{REF}$	参考电压范围 ( $V_{REFH} - V_{REFL}$ )	2.0	—	—	V	$V_{DD} < 3.0V$
			3	—	—	V	$V_{DD} \geq 3.0V$
A21	$V_{REFH}$	参考电压高电平	$V_{SS} + \Delta V_{REF}$	—	$V_{DD}$	V	
A22	$V_{REFL}$	参考电压低电平	$V_{SS} - 0.3V$	—	$V_{DD} - 3.0V$	V	
A25	$V_{AIN}$	模拟输入电压	$V_{REFL}$	—	$V_{REFH}$	V	
A30	$Z_{AIN}$	模拟电源阻抗推荐值	—	—	2.5	k $\Omega$	
A50	$I_{REF}$	$V_{REF}$ 输入电流 <sup>(2)</sup>	—	—	5	$\mu A$	在 $V_{AIN}$ 采集期间 在 A/D 转换期间
			—	—	150	$\mu A$	

注 1: A/D 转换结果不会因输入电压的增加而减小, 并且不会丢失编码。

2:  $V_{REFH}$  电流来自作为  $V_{REFH}$  源的 RA3/AN3/ $V_{REF+}$  引脚或  $V_{DD}$ 。

$V_{REFL}$  电流来自作为  $V_{REFL}$  源的 RA2/AN2/ $V_{REF-}$  引脚或  $V_{SS}$ 。



图 26-22: A/D 转换时序

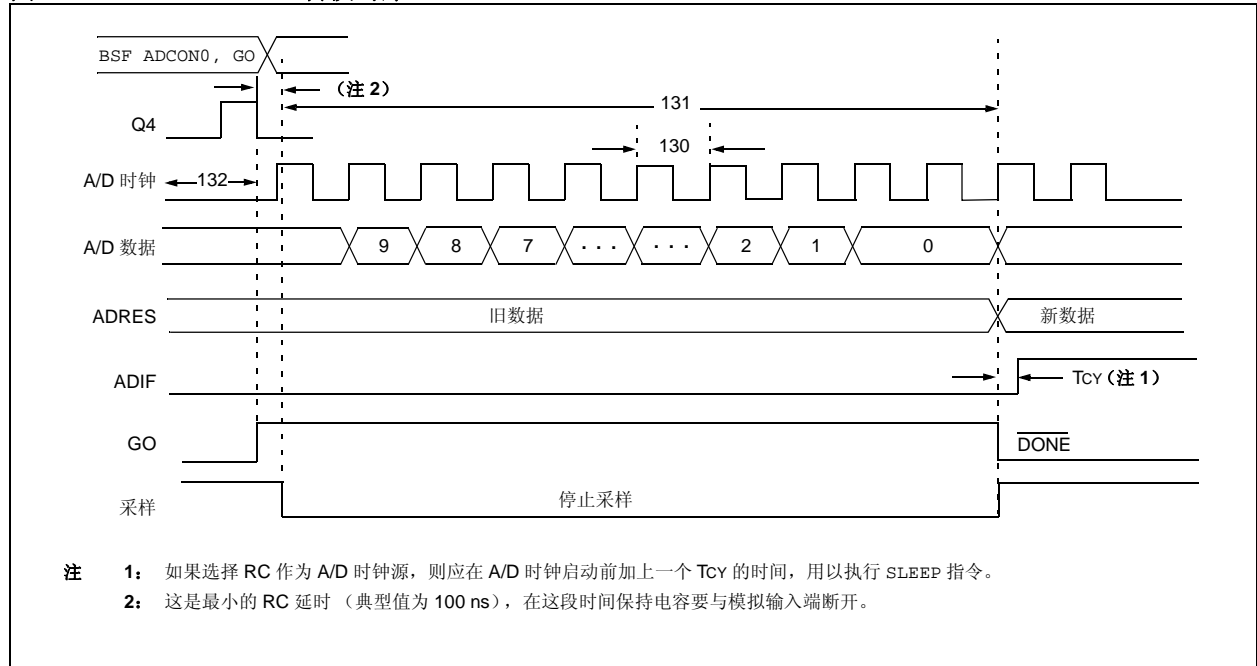


表 26-27: A/D 转换要求

参数编号	符号	特性	最小值	最大值	单位	条件
130	TAD	A/D 时钟周期	0.7	25.0 <sup>(1)</sup>	μs	基于 TOSC, VREF ≥ 3.0V
			—	1	μs	A/D RC 模式
131	TCNV	转换时间 (不包括采集时间) (2)	11	12	TAD	
132	TACQ	采集时间 (3)	1.4	—	μs	-40°C 至 +85°C
135	TSWC	转换 → 采样的切换时间	—	(注 4)		
137	TDIS	放电时间	0.2	—	μs	

- 注 1:** A/D 时钟周期取决于器件频率和 TAD 时钟分频比。  
**注 2:** 可在后续 Tcy 周期内读 ADRES 寄存器。  
**注 3:** 转换完成后当电压满量程变化时 (VDD 至 VSS, 或 VSS 至 VDD), 保持电容采样一个 “新” 输入电压所需的时间。输入通道上的信号源阻抗 (Rs) 为 50Ω。  
**注 4:** 在器件时钟的下一个周期上。

# PIC18F85J11 系列

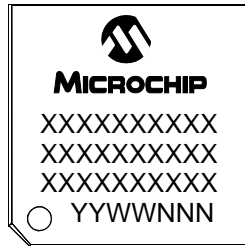
---

注:

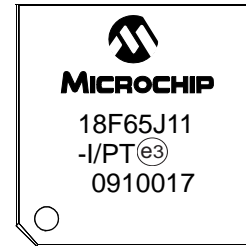
## 27.0 封装信息

### 27.1 封装标识信息

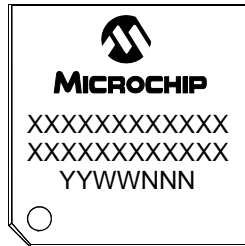
64 引脚 TQFP



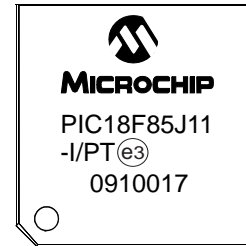
示例



80 引脚 TQFP



示例



图注: XX...X 客户指定信息  
Y 年份代码（日历年的最后一位数字）  
YY 年份代码（日历年的最后两位数字）  
WW 星期代码（一月一日的星期代码为“01”）  
NNN 以字母数字排序的追踪代码  
(e3) 雾锡（Matte Tin, Sn）的 JEDEC 无铅标志  
\* 表示无铅包装。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。

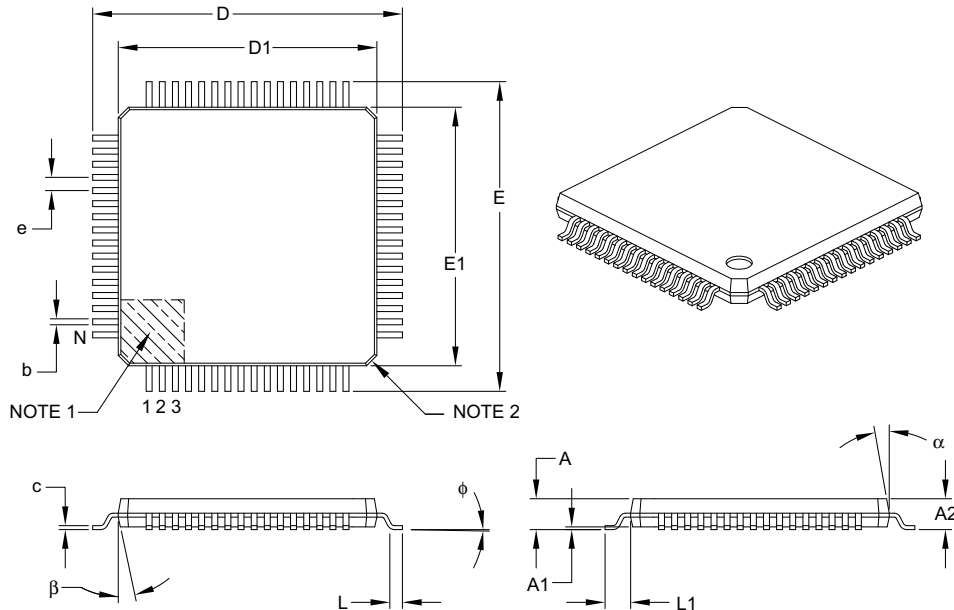
# PIC18F85J11 系列

## 27.2 封装详细信息

以下部分给出了封装的技术详细信息。

### 64 引脚塑封薄型四方扁平封装（PT）——主体 10x10x1 mm，底脚占位 2.00 mm [TQFP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N	64		
Lead Pitch	e	0.50 BSC		
Overall Height	A	—	—	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	—	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	$\phi$	0°	3.5°	7°
Overall Width	E	12.00 BSC		
Overall Length	D	12.00 BSC		
Molded Package Width	E1	10.00 BSC		
Molded Package Length	D1	10.00 BSC		
Lead Thickness	c	0.09	—	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	$\alpha$	11°	12°	13°
Mold Draft Angle Bottom	$\beta$	11°	12°	13°

#### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Chamfers at corners are optional; size may vary.
- Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

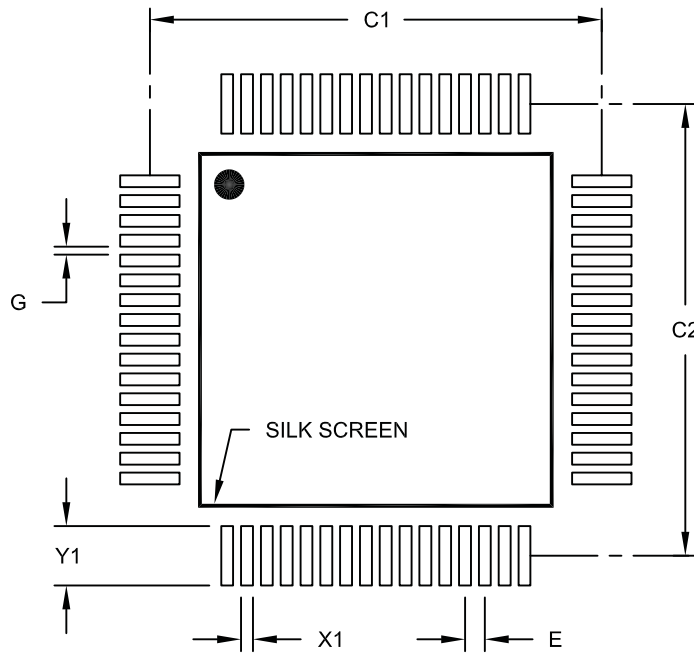
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-085B

## 64 引脚塑封薄型四方扁平封装（PT）——主体 10x10x1 mm，底脚占位 2.00 mm [TQFP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X64)	X1			0.30
Contact Pad Length (X64)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

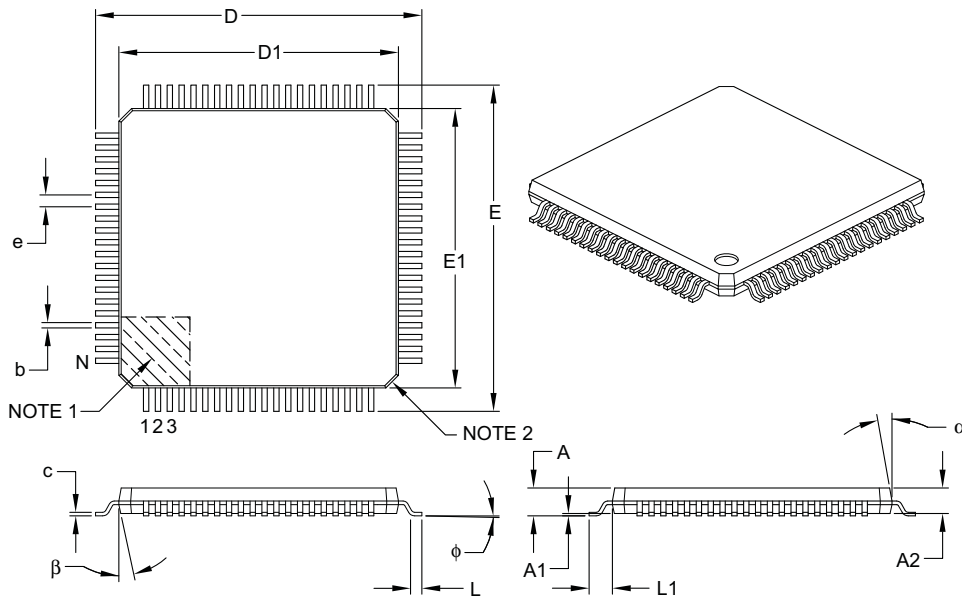
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2085A

# PIC18F85J11 系列

## 80 引脚塑封薄型四方扁平封装（PT）——主体 12x12x1 mm，底脚占位 2.00 mm [TQFP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N	80		
Lead Pitch	e	0.50 BSC		
Overall Height	A	—	—	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	—	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	φ	0°	3.5°	7°
Overall Width	E	14.00 BSC		
Overall Length	D	14.00 BSC		
Molded Package Width	E1	12.00 BSC		
Molded Package Length	D1	12.00 BSC		
Lead Thickness	c	0.09	—	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Chamfers at corners are optional; size may vary.
- Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

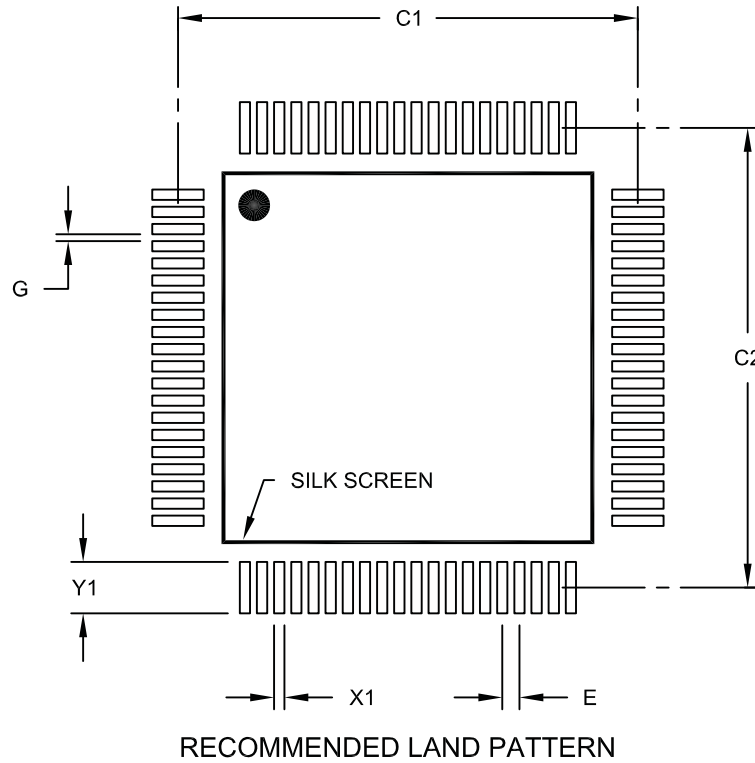
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-092B

## 80 引脚塑封薄型四方扁平封装（PT）——主体 12x12x1 mm，底脚占位 2.00 mm [TQFP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Contact Pad Spacing	C1		13.40	
Contact Pad Spacing	C2		13.40	
Contact Pad Width (X80)	X1			0.30
Contact Pad Length (X80)	Y1			1.50
Distance Between Pads	G	0.20		

### Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2092A

# PIC18F85J11 系列

---

注:



## 附录 A：版本历史

### 版本 A（2006 年 10 月）

PIC18F85J11 系列器件数据手册的最初版本。

### 版本 B（2007 年 3 月）

更新了掉电和供电电流的电气特性和封装细节图。

### 版本 C（2007 年 4 月）

更新了电气特性。

### 版本 D（2010 年 2 月）

更新了电气特性和封装的详细描述。对整个文档进行了较小的文本编辑。

## 附录 B：高档器件系列之间的移植

PIC18F85J1 和 PIC18F8722 系列器件具有非常相似的功能和特性集。然而，在不同系列器件之间移植应用以实现新的设计目标时，还需要考虑这些器件之间的一些潜在的重要差异。表 B-1 汇总了这些差异。本节后面更详细地讨论了对移植有重大影响的各方面差异。

表 B-1：PIC18F85J11 和 PIC18F8722 系列之间的显著差异

特性	PIC18F85J11 系列	PIC18F8722 系列
工作频率	40 MHz（2.35V 时）	40 MHz（4.2V 时）
供电电压	2.0V-3.6V，双电压要求	2.0V-5.5V
工作电流	低	较低
程序存储器耐擦写性	1,000 个写 / 擦除周期（典型值）	100,000 个写 / 擦除周期（典型值）
25 mA 的 I/O 拉灌电流	仅限 PORTB 和 PORTC	所有端口
I/O 引脚的输入电压容限	仅数字引脚为 5.5V	所有 I/O 引脚均为 V <sub>DD</sub>
I/O	68（RF0 不可用）	70
上拉	PORTB、PORTD、PORTE 和 PORTJ	PORTB
振荡器选项	有限选项（EC、HS、PLL 和灵活的 INTRC）	更多选项（EC、HS、XT、LP、RC、PLL 和灵活的 INTRC）
程序存储器保存时间	20 年（最小值）	40 年（最小值）
自写程序存储器	可用	可用
编程时间（标称值）	156 μs/ 字节（10 ms/ 64 字节块）	15.6 μs/ 字节（1 ms/ 64）
编程模式入口	低电压、密钥序列	V <sub>PP</sub> 和 LVP
代码保护	单个块、全部和不保护	多个代码保护块
配置字	保存在程序存储空间最后 4 个字中	保存在配置空间中，起始地址为 300000h
上电延时定时器	始终保持开启	可配置的
数据 EEPROM	使用自编程	可用
BOR	通过稳压器提供简单 BOR	可编程 BOR
LVD	通过稳压器提供简单 LVD	可用
A/D 通道	12	16
A/D 校准	需要	不需要
扩展单片机模式（EMB）	自校准功能	可用
外部存储器寻址	地址移位可用	地址移位不可用
在线仿真	不可用	可用

# PIC18F85J11 系列

## B.1 功耗要求差异

PIC18F85J11 和 PIC18F8722 系列器件之间的最显著差异是功耗要求。PIC18F85J11 器件是基于较小的处理能力设计的，从而导致具有较低的最大电压和较高的泄漏电流。

PIC18F85J11 器件的工作电压范围是 2.0V 至 3.6V。另外，这些器件还有分离电源要求：一个为内核逻辑提供电压，而另一个为 I/O 提供电压。其中一个 VDD 引脚单独为内核逻辑供电，即 VDDCORE。该引脚具有特定的电压和电容要求，如第 26.0 节“电气特性”所述。

PIC18F85J11 器件的电流规范尚未确定。

## B.2 引脚差异

PIC18F85J11 和 PIC18F8722 系列的引脚布局存在以下一些差异：

- 输入电压容限
- 输出电流能力
- 可用 I/O

PIC18F85J11 上的仅具有数字输入能力的引脚可承受高达 5.5V 的电压，从而可承受高于 VDD 的电压。第 11.1 节“**I/O 端口引脚的驱动能力**”中的表 11-1 包含了完整的清单。

除了输入差异，还存在输出差异。PIC18F85J11 器件具有三个等级的引脚输出电流能力：高、中和低。不是所有 I/O 引脚都可拉灌同等级别的电流。仅 PORTB 和 PORTC 具有 25 mA 的拉灌电流能力，而 PIC18F8722 的所有输出引脚都具有这个能力。第 11.1 节“**I/O 端口引脚的驱动能力**”中的表 11-2 包含了输出能力的完整清单。

在如何实现 PIC18F85J11 器件的一些引脚功能上存在额外差异。首先，MCLR 引脚仅专用于 MCLR 且不能配置为输入（RG5）。最后，PIC18F85J11 器件上不存在 RF0。

当在 PIC18F8722 和 PIC18F85J11 器件之间进行转换时，应考虑所有这些引脚差异（包括电源引脚差异）。

## B.3 振荡器差异

PIC18F8722 和 PIC18F85J11 系列器件具有相同的振荡器选项范围。主要差异是：PIC18F85J11 系列器件支持更少数量的主（外部）振荡器选项，即 HS 和 EC 振荡器模式。

尽管这两个器件系列均有可与主振荡器一起使用的内部 PLL，但是 PIC18F85J11 系列的 PLL 不能通过器件配置选项使能，却必须用软件使能。

在 PIC18F8722 和 PIC18F85J11 器件系列之间进行转换时，应考虑它们的时钟差异。

## B.4 外设

在 PIC18F85J11 和 PIC18F8722 之间进行转换时，还必须考虑以下外设差异：

- **外部存储总线：**PIC18F85J11 上的外部存储总线（External Memory Bus, EMB）不支持单片机模式；然而，它支持外部地址偏移。
- **A/D 转换器：**PIC18F85J11 器件只有 12 路通道。这些器件的转换器在正常工作之前还需要进行校准。
- **数据 EEPROM：**PIC18F85J11 器件不具有该模块。
- **BOR：**PIC18F85J11 器件不具有可编程 BOR。通过使用内部稳压器提供简单的欠压复位能力。
- **LVD：**PIC18F85J11 器件不具有单独的可编程 LVD 模块。通过使用内部稳压器提供简单的具有可配置中断的低电压检测能力。

## 索引

### A

A/D .....	259
A/D 转换器中断, 配置 .....	263
ADCAL 位 .....	267
ADCON0 寄存器 .....	259
ADCON1 寄存器 .....	259
ADCON2 寄存器 .....	259
ADRESH 寄存器 .....	259, 262
ADRESL 寄存器 .....	259
CCP2 触发信号的使用 .....	266
采集要求 .....	264
模拟端口引脚, 配置 .....	265
配置模块 .....	263
特殊事件触发信号 (CCP2) .....	266
相关的寄存器 .....	267
在功耗管理模式下工作 .....	267
转换 .....	266
转换器特性 .....	384
转换器校准 .....	267
转换时钟 (TAD) .....	265
转换要求 .....	385
转换状态 (GO/DONE 位) .....	262
自动采集时间 .....	265
ACKSTAT .....	213
ACKSTAT 状态标志 .....	213
ADCAL 位 .....	267
ADCON0 寄存器 .....	259
GO/DONE 位 .....	262
ADCON1 寄存器 .....	259
ADCON2 寄存器 .....	259
ADDFSR .....	342
ADDLW .....	305
ADDWF .....	305
ADDWFC .....	306
ADDULNK .....	342
ADRESH 寄存器 .....	259
ADRESL 寄存器 .....	259, 262
ANDLW .....	306
ANDWF .....	307
AUSART .....	
波特率发生器 (BRG) .....	248
波特率误差, 计算 .....	248
波特率, 异步模式 .....	249
采样 .....	248
高波特率选择位 (BRGH 位) .....	248
相关的寄存器 .....	248
在功耗管理模式下的操作 .....	248
控制寄存器 .....	245
同步从模式 .....	257
发送 .....	257
发送设置 .....	257
接收 .....	258
接收设置 .....	258
相关寄存器, 发送 .....	257
相关寄存器, 接收 .....	258
同步主模式 .....	254
发送 .....	254
发送设置 .....	254
接收 .....	256
接收设置 .....	256
相关寄存器, 发送 .....	255
相关寄存器, 接收 .....	256

异步模式 .....	250
发送器 .....	250
发送设置 .....	250
接收器 .....	252
接收设置 .....	252
设置带有地址检测功能的 9 位模式 .....	252
带有地址检测使能设置的接收 .....	252
相关寄存器, 发送 .....	251
相关寄存器, 接收 .....	253

### B

BC .....	307
BCF .....	308
BF .....	213
BF 状态标志 .....	213
BN .....	308
BNC .....	309
BNN .....	309
BN OV .....	310
BNZ .....	310
BOR. 参阅欠压复位。	
BOV .....	313
BRA .....	311
BRGH 位 .....	
TXSTA1 寄存器 .....	227
TXSTA2 寄存器 .....	248
BRG. 参阅波特率发生器。	
BSF .....	311
BTFSC .....	312
BTFSS .....	312
BTG .....	313
BZ .....	314
版本历史 .....	393
比较器 .....	269
参考 .....	271
内部信号 .....	271
外部信号 .....	271
复位的影响 .....	272
工作原理 .....	271
模拟输入连接注意事项 .....	273
配置 .....	270
输出 .....	271
相关的寄存器 .....	273
响应时间 .....	271
休眠模式下的工作原理 .....	272
中断 .....	272
比较器参考电压 .....	275
复位的影响 .....	276
精度和误差 .....	276
连接注意事项 .....	276
配置 .....	275
相关的寄存器 .....	277
在休眠模式下工作 .....	276
比较器规范 .....	363
比较 (CCP2 模块) .....	
特殊事件触发信号 .....	266
比较 (CCP 模块) .....	173
CCPR2 寄存器 .....	173
CCPx 引脚配置 .....	173
软件中断 .....	173
Timer1/Timer3 模式选择 .....	173
特殊事件触发信号 .....	167, 173
相关的寄存器 .....	174
编程, 器件指令 .....	299

# PIC18F85J11 系列

变更通知客户服务 .....	405
表读 / 表写 .....	69
表指针操作（表） .....	92
并行从端口（PSP） .....	150
RE0/RD 引脚 .....	150
RE1/WR 引脚 .....	150
RE2/CS 引脚 .....	150
相关的寄存器 .....	152
波特率发生器 .....	209
捕捉 / 比较 / PWM（CCP） .....	169
比较模式。参见比较。	
捕捉模式。参见捕捉。	
CCP1 和 CCP2 在使用定时器资源方面的相互关系 ..	171
CCPRxH 寄存器 .....	170
CCPRxL 寄存器 .....	170
CCPx 模式和定时器资源 .....	170
互连配置 .....	170
配置 .....	170
捕捉（CCP 模块） .....	172
CCPR2H:CCPR2L 寄存器 .....	172
CCPx 引脚配置 .....	172
软件中断 .....	172
Timer1/Timer3 模式选择 .....	172
相关的寄存器 .....	174
<b>C</b>	
CALL .....	314
CALLW .....	343
C 编译器	
MPLAB C18 .....	296
CLRF .....	315
CLRWDT .....	315
COMF .....	316
CPFSEQ .....	316
CPFSGT .....	317
CPFSLT .....	317
CPU 特性 .....	279
参考电压规范 .....	363
程序存储器	
查找表 .....	69
存储器硬编码向量 .....	64
存储器映射 .....	63
硬编码向量和配置字 .....	64
复位向量 .....	64
扩展的指令集 .....	84
模式 .....	65
存储器访问（表） .....	66
单片机 .....	65
扩展单片机 .....	65
扩展单片机（地址平移） .....	66
闪存配置字 .....	64
指令 .....	71
双字 .....	71
中断向量 .....	64
程序存储器模式	
外部存储器总线的工作原理 .....	102
程序计数器 .....	67
PCLATH 和 PCLATU 寄存器 .....	67
PCL、PCH 和 PCU 寄存器 .....	67
程序校验和代码保护 .....	293
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 模式。	
从选择（SS） .....	179
存储器编程要求 .....	362

存储器构成 .....	63
程序存储器 .....	63
数据存储器 .....	72

## D

DAW .....	318
DCFSNZ .....	319
DECf .....	318
DECFSZ .....	319
代码保护 .....	279
代码示例	
16 x 16 无符号乘法程序 .....	112
16 x 16 有符号乘法程序 .....	112
8 x 8 无符号乘法程序 .....	111
8 x 8 有符号乘法程序 .....	111
擦除闪存程序存储器块 .....	94
初始化 PORTA .....	130
初始化 PORTB .....	132
初始化 PORTC .....	134
初始化 PORTD .....	137
初始化 PORTE .....	140
初始化 PORTF .....	143
初始化 PORTG .....	145
初始化 PORTH .....	147
初始化 PORTJ .....	148
读一个闪存程序存储器字 .....	93
改变捕捉预分频比 .....	172
将 Status、WREG 和 BSR 寄存器的值保存	
在 RAM 中 .....	128
快速寄存器堆栈 .....	69
使用间接寻址清零 RAM（Bank 1）的方法 .....	82
使用偏移量计算 GOTO .....	69
使用 Timer1 中断服务程序实现实时时钟 .....	161
写闪存程序存储器 .....	96
装载 SSPBUF（SSPSR）寄存器 .....	182
单片机模式 .....	102
电气特性 .....	349
读者反馈表 .....	406
对标准 PIC18 指令的影响 .....	346
堆栈满 / 下溢复位 .....	69

## E

ENVREG 引脚 .....	288
EUSART	
波特率发生器（BRG） .....	227
波特率误差，计算 .....	228
波特率，异步模式 .....	229
采样 .....	227
高波特率选择位（BRGH 位） .....	227
相关的寄存器 .....	228
在功耗管理模式下的操作 .....	227
自动波特率检测 .....	231
控制寄存器 .....	223
同步从模式 .....	242
发送 .....	242
发送设置 .....	242
接收 .....	243
接收设置 .....	243
相关寄存器，发送 .....	242
相关寄存器，接收 .....	243
同步主模式 .....	239
发送 .....	239
发送设置 .....	239
接收 .....	241
接收设置 .....	241
相关寄存器，发送 .....	240

相关寄存器, 接收 .....	241	16 x 16 有符号乘法算法 .....	112
异步模式 .....	233	A/D 采集时间 .....	264
发送器 .....	233	A/D 最小充电时间 .....	264
发送设置 .....	233	计算所需的最小采集时间 .....	264
间隔和同步发送序列 .....	238	固件指令 .....	299
间隔字符序列 .....	238	故障保护时钟监视器 .....	279, 291
接收器 .....	235	功耗管理模式下的中断 .....	292
接收设置 .....	235	POR 或从休眠中唤醒 .....	292
设置带有地址检测功能的 9 位模式 .....	235	退出故障保护运行模式 .....	292
同步间隔字符自动唤醒 .....	237	振荡器故障期间 WDT .....	291
相关寄存器, 发送 .....	234		
相关寄存器, 接收 .....	236	<b>H</b>	
<b>F</b>		汇编器	
FSCM。参阅故障保护时钟监视器。		MPASM 汇编器 .....	296
返回地址堆栈 .....	67	<b>I</b>	
返回堆栈指针 (STKPTR) .....	68	I/O 端口 .....	129
访问栈顶 .....	67	漏极开路输出 .....	130
封装 .....	387	上拉配置 .....	130
标识 .....	387	输入引脚和电压注意事项 .....	129
详细信息 .....	388	引脚驱动能力 .....	129
复位 .....	51, 279	引脚输出驱动能力 .....	129
堆栈满复位 .....	51	I/O 引脚说明	
堆栈下溢复位 .....	51	PIC18F6XJ11 .....	14
功耗管理模式下的 MCLR 复位 .....	51	PIC18F8XJ11 .....	21
看门狗定时器 (WDT) 复位 .....	51	I <sup>2</sup> C 模式 (MSSP) .....	188
欠压复位 (BOR) .....	51, 279	波特率发生器 .....	209
RESET 指令 .....	51	串行时钟 (SCK/SCL) .....	195
上电复位 (POR) .....	51, 279	从模式 .....	193
上电延迟定时器 (PWRT) .....	279	地址掩码模式 .....	194
振荡器起振定时器 (OST) .....	279	发送 .....	195
正常工作状态下的 MCLR 复位 .....	51	接收 .....	195
		寻址 .....	193
<b>G</b>		带有 BRG 的 I <sup>2</sup> C 时钟速率 .....	209
GOTO .....	320	读/写位信息 (R/W 位) .....	193, 195
高档器件系列之间的移植 .....	393	多主器件模式 .....	217
功耗要求差异 .....	394	多主器件通信、总线冲突与总线仲裁 .....	217
外设 .....	394	复位的影响 .....	217
引脚差异 .....	394	工作原理 .....	193
振荡器差异 .....	394	寄存器 .....	188
功耗管理模式对各种时钟源的影响 .....	42	时钟同步和 CKP 位 .....	203
功耗管理模式 .....	43	时钟延长 .....	202
多条 Sleep 指令 .....	44	10 位从发送模式 .....	202
和 SPI 模式的工作原理 .....	187	10 位从接收模式 (SEN = 1) .....	202
汇总 (表) .....	43	7 位从发送模式 .....	202
进入 .....	43	7 位从接收模式 (SEN = 1) .....	202
空闲模式 .....	47	时钟仲裁 .....	210
PRI_IDLE .....	48	停止条件时序 .....	216
RC_IDLE .....	49	相关的寄存器 .....	222
SEC_IDLE .....	48	休眠状态下的操作 .....	217
时钟源 .....	43	应答序列时序 .....	216
时钟转换和状态指示位 .....	44	支持广播呼叫地址 .....	206
退出空闲和休眠模式 .....	49	主模式 .....	207
通过复位 .....	49	波特率发生器 .....	209
通过 WDT 超时 .....	49	发送 .....	213
通过中断 .....	49	工作原理 .....	208
在没有振荡器起振延时的情况下 .....	49	接收 .....	213
休眠模式 .....	47	启动条件时序 .....	211
选择 .....	43	重复启动条件时序 .....	212
运行模式 .....	44	总线冲突	
PRI_RUN .....	44	停止条件期间 .....	221
RC_RUN .....	46	在重复启动条件期间 .....	220
SEC_RUN .....	44	INCF .....	320
功耗管理模式对各种时钟源的影响 .....	42	INCFSZ .....	321
公式		INFSNZ .....	321
16 x 16 无符号乘法算法 .....	112	INTCON 寄存器 .....	115

# PIC18F85J11 系列

INTCON 寄存器.....	115
RBIF 位 .....	132
INTOSC, INTRC. 参见内部振荡器电路。	
IORLW .....	322
IORWF .....	322
IPR 寄存器 .....	124
<b>J</b>	
寄存器	
ADCON0 (A/D 控制寄存器 0) .....	259
ADCON1 (A/D 控制寄存器 1) .....	260
ADCON2 (A/D 控制寄存器 2) .....	261
BAUDCON1 (波特率控制寄存器 1) .....	226
CCPxCON (CCPx 控制寄存器) .....	169
CMCON (比较器控制寄存器) .....	269
CONFIG1H (配置寄存器 1 的高字节) .....	281
CONFIG1L (配置寄存器 1 的低字节) .....	281
CONFIG2H (配置寄存器 2 的高字节) .....	283
CONFIG2L (配置寄存器 2 的低字节) .....	282
CONFIG3H (配置寄存器 3 的高字节) .....	284
CONFIG3L (配置寄存器 3 的低字节) .....	65, 284
CVRCON (比较器参考电压控制寄存器) .....	275
DEVID1 (器件 ID 寄存器 1) .....	285
DEVID2 (器件 ID 寄存器 2) .....	285
EECON1 (EEPROM 控制 1) .....	91
INTCON2 (中断控制 2) .....	116
INTCON3 (中断控制 3) .....	117
INTCON (中断控制) .....	115
IPR1 (外设中断优先级 1) .....	124
IPR2 (外设中断优先级 2) .....	125
IPR3 (外设中断优先级 3) .....	126
MEMCON (外部存储器总线控制) .....	100
OSCCON (振荡器控制) .....	36
OSCTUNE (振荡器调节) .....	37
PIE1( 外设中断允许 1) .....	121
PIE2 (外设中断允许 2) .....	122
PIE3( 外设中断允许 3) .....	123
PIR1 (外设中断请求标志 1) .....	118
PIR2 (外设中断请求标志 2) .....	119
PIR3 (外设中断请求标志 3) .....	120
PSPCON (并行从端口控制) .....	151
RCON (复位控制) .....	52, 127
RCSTA1 (EUSART 接收状态和控制寄存器) .....	225
RCSTA2 (AUSART 接收状态和控制寄存器) .....	247
SSPCON1 (MSSP 控制寄存器 1, I <sup>2</sup> C 模式) .....	190
SSPCON1 (MSSP 控制寄存器 1, SPI 模式) .....	181
SSPCON2 (MSSP 控制寄存器 2, I <sup>2</sup> C 从模式) .....	192
SSPCON2 (MSSP 控制寄存器 2, I <sup>2</sup> C 主模式) .....	191
SSPSTAT (MSSP 状态寄存器, SPI 模式) .....	180
SSPSTAT (MSSP 状态, I <sup>2</sup> C 模式) .....	189
STATUS .....	81
STKPTR (堆栈指针) .....	68
T0CON (Timer0 控制寄存器) .....	153
T1CON (Timer1 控制寄存器) .....	157
T2CON (Timer2 控制寄存器) .....	163
T3CON (Timer3 控制寄存器) .....	165
TXSTA1 (EUSART 发送状态和控制寄存器) .....	224
TXSTA2 (AUSART 发动状态和控制寄存器) .....	246
WDTCON (看门狗定时器控制) .....	287
寄存器汇总 .....	77 - 80
寄存器文件 .....	75
集成电路. 见 I <sup>2</sup> C 模式	
计算 GOTO .....	69
间接寻址 .....	83
交流 (时序) 特性 .....	364
参数符号 .....	364

器件时序规范的负载条件 .....	365
时序条件 .....	365
温度和电压规范 .....	365
结构框图	
表读操作 .....	89
表写操作 .....	90
读闪存程序存储器 .....	93
对闪存程序存储器进行表写操作 .....	95
故障保护时钟监视器 .....	291
看门狗定时器 .....	286
PIC18F6XJ11 .....	12
PIC18F8XJ11 .....	13
PLL .....	40
片内稳压器的连接 .....	288
片上复位电路 .....	51
器件时钟 .....	35
外部上电复位电路 (V <sub>DD</sub> 慢速上电) .....	53
晶振 / 陶瓷谐振器 .....	39
进入空闲模式的转换时序 .....	48
绝对最大值 .....	349
<b>K</b>	
开发支持 .....	295
看门狗定时器 (WDT) .....	279, 286
编程注意事项 .....	286
控制寄存器 .....	286
相关的寄存器 .....	287
勘误表 .....	7
客户通知服务 .....	405
客户支持 .....	405
可寻址通用同步异步收发器 (AUSART) 参见 AUSART。	
快速寄存器堆栈 .....	69
框图	
16 位模式 Timer0 .....	154
16 位字节写模式 .....	103
16 位字节选择模式 .....	105
16 位字写模式 .....	104
8 位复用模式 .....	107
8 位模式 Timer0 .....	154
A/D .....	262
AUSART 发送 .....	250
AUSART 接收 .....	252
比较模式工作原理 .....	173
比较器参考电压 .....	276
比较器参考电压输出缓冲器示例 .....	277
比较器 I/O 工作模式 .....	270
比较器模拟输入模型 .....	273
比较器输出 .....	272
波特率发生器 .....	209
捕捉模式工作原理 .....	172
单个比较器 .....	271
EUSART 发送 .....	233
EUSART 接收 .....	235
MSSP (I <sup>2</sup> C 模式) .....	188
MSSP (I <sup>2</sup> C 主模式) .....	207
MSSP (SPI 模式) .....	179
模拟输入模型 .....	263
PORTD 和 PORTE (并行从端口) .....	150
PWM 工作原理 (简化) .....	175
Timer2 .....	164
Timer3 (16 位读 / 写模式) .....	166
Timer3 (8 位模式) .....	166
Timer1 (16 位读 / 写模式) .....	158
Timer1 (8 位模式) .....	158
通用 I/O 端口的工作原理 .....	129
中断逻辑 .....	114

扩展单片机模式 .....	102
扩展指令集 .....	341
ADDFSR .....	342
ADDULNK .....	342
CALLW .....	343
MOVSF .....	343
MOVSS .....	344
PUSHL .....	344
SUBFSR .....	345
SUBULNK .....	345
使能时的注意事项 .....	346
语法 .....	341
与 MPLAB IDE 工具一起使用 .....	348

## L

LFSR .....	323
立即数变址寻址 .....	
和标准 PIC18 指令 .....	346
立即数变址寻址模式 .....	346

## M

Microchip 因特网网站 .....	405
MOVF .....	323
MOVFF .....	324
MOVLB .....	324
MOVLW .....	325
MOVSF .....	343
MOVSS .....	344
MOVWF .....	325
MPLAB ASM30 汇编器、链接器和库管理器 .....	296
MPLAB PM3 器件编程器 .....	298
MPLAB REAL ICE 在线仿真器系统 .....	297
MPLAB 集成开发环境软件 .....	295
MPLINK 目标链接器 /MPLIB 目标库管理器 .....	296
MSSP .....	
ACK 脉冲 .....	193, 195
控制寄存器（通用） .....	179
模块概述 .....	179
SPI 主 / 从器件连接 .....	183
SSPBUF 寄存器 .....	184
SSPSR 寄存器 .....	184
MULLW .....	326
MULWF .....	326
脉宽调制。参见 PWM（CCP 模块）。	
默认系统时钟 .....	38
模数转换器。参见 A/D	

## N

NEGF .....	327
NOP .....	327
内部 RC 振荡器 .....	
与 WDT 一起使用 .....	286
内部稳压器规范 .....	363
内部振荡器电路 .....	41
调节 .....	41
INTOSC 频率漂移 .....	41
INTOSC 输出频率 .....	41
OSC1, OSC2 引脚配置 .....	41
内核功能 .....	
存储器选项 .....	9
扩展指令集 .....	9
纳瓦技术 .....	9
外部存储器总线 .....	9
易于移植 .....	10
振荡器选项和功能 .....	9

## P

PIE 寄存器 .....	121
PIR 寄存器 .....	118
PLL .....	40
ECPLL 振荡器模式 .....	40
HSPLL 振荡器模式 .....	40
POP .....	328
PORTA .....	
LATA 寄存器 .....	130
PORTA 寄存器 .....	130
TRISA 寄存器 .....	130
相关的寄存器 .....	131
PORTB .....	
LATB 寄存器 .....	132
PORTB 寄存器 .....	132
RB7:RB4 电平变化中断标志（RBIF 位） .....	132
TRISB 寄存器 .....	132
相关的寄存器 .....	133
PORTC .....	
LATC 寄存器 .....	134
PORTC 寄存器 .....	134
RC3/SCK/SCL 引脚 .....	195
TRISC 寄存器 .....	134
相关的寄存器 .....	136
PORTD .....	150
LATD 寄存器 .....	137
PORTD 寄存器 .....	137
TRISD 寄存器 .....	137
相关的寄存器 .....	139
PORTE .....	
LATE 寄存器 .....	140
PORTE 寄存器 .....	140
RE0/RD 引脚 .....	150
RE1/WR 引脚 .....	150
RE2/CS 引脚 .....	150
TRISE 寄存器 .....	140
相关的寄存器 .....	142
PORTF .....	
LATF 寄存器 .....	143
PORTF 寄存器 .....	143
TRISF 寄存器 .....	143
相关的寄存器 .....	144
PORTG .....	
LATG 寄存器 .....	145
PORTG 寄存器 .....	145
TRISG 寄存器 .....	145
相关的寄存器 .....	146
PORTH .....	
LATH 寄存器 .....	147
PORTH 寄存器 .....	147
TRISH 寄存器 .....	147
相关的寄存器 .....	147
PORTJ .....	
LATJ 寄存器 .....	148
PORTJ 寄存器 .....	148
TRISJ 寄存器 .....	148
相关的寄存器 .....	149
POR。参阅上电复位。	
PRI_IDLE 模式 .....	48
PRI_RUN 模式 .....	44
PSPMODE 位（PSPCON 寄存器） .....	150
PSP。参见并行从端口。	
PWM（CCP 模块） .....	
频率 / 分辨率示例 .....	176
设置 PWM 操作 .....	177
TMR2 与 PR2 匹配 .....	175

# PIC18F85J11 系列

相关的寄存器 .....	177
占空比 .....	176
周期 .....	175
PUSH .....	328
PUSH 和 POP 指令 .....	68
PUSHL .....	344
配置寄存器保护 .....	293
配置位 .....	279
配置位, 器件 ID	
相关的寄存器 .....	280
片内稳压器 .....	288
低电压检测 (LVD) .....	288
欠压复位 (BOR) .....	289
上电要求 .....	289
休眠模式下的操作 .....	289
<b>Q</b>	
器件概述 .....	9
功能 (64 引脚器件) .....	11
功能 (80 引脚器件) .....	11
Q 时钟 .....	176
欠压复位 (BOR) .....	53
和片内稳压器 .....	289
检测 .....	53
<b>R</b>	
RAM. 参见数据存储器。	
RCALL .....	329
RC_IDLE 模式 .....	49
RCON 寄存器	
初始化期间的位状态 .....	56
RC_RUN 模式 .....	46
RESET .....	329
RETFIE .....	330
RETLW .....	330
RETURN .....	331
RLCF .....	331
RLNCF .....	332
RRCF .....	332
RRNCF .....	333
软件模拟器 (MPLAB SIM) .....	297
<b>S</b>	
SCK .....	179
SDI .....	179
SDO .....	179
SEC_IDLE 模式 .....	48
SEC_RUN 模式 .....	44
SETF .....	333
SLEEP .....	334
SPI 模式 (MSSP)	
串行时钟 .....	179
串行数据输出 .....	179
串行数据输入 .....	179
从模式 .....	185
从选择 .....	179
从选择同步 .....	185
典型连接 .....	183
复位的影响 .....	187
工作原理 .....	182
SPI 时钟 .....	184
使能 SPI I/O .....	183
相关的寄存器 .....	187
在功耗管理模式下的操作 .....	187
主 / 从器件连接 .....	183
主模式 .....	184

总线模式兼容性 .....	187
SS .....	179
SSPOV .....	213
SSPOV 状态标志 .....	213
SSPSTAT 寄存器	
R/W 位 .....	193, 195
STATUS 寄存器 .....	81
SWAPF .....	336
SUBFSR .....	345
SUBFWB .....	334
SUBLW .....	335
SUBWF .....	335
SUBWFB .....	336
SUBULNK .....	345
闪存程序存储器 .....	89
表读和表写 .....	89
表指针	
基于不同操作的表指针 .....	92
表指针边界 .....	92
擦除 .....	94
擦除序列 .....	94
代码保护期间的操作 .....	97
读 .....	93
控制寄存器 .....	90
EECON1 和 EECON2 .....	90
TABLAT (表锁存器) 寄存器 .....	92
TBLPTR (表指针) 寄存器 .....	92
相关的寄存器 .....	97
写 .....	95
写校验 .....	97
意外终止 .....	97
写操作序列 .....	95
闪存配置字	
映射 .....	279
上电复位 (POR) .....	53
上电延迟定时器 (PWRT) .....	54
延时时序 .....	54
上电延时 .....	42
上电延时定时器 (PWRT) .....	42
时序图	
A/D 转换 .....	385
BRG 溢出时序 .....	232
并行从端口 (PSP) 读 .....	152
并行从端口 (PSP) 写 .....	151
捕捉 / 比较 / PWM (CCP1 和 CCP2) .....	374
CLKO 和 I/O .....	368
程序存储器读 .....	370
从空闲模式唤醒进入运行模式的转换 .....	48
从 RC_RUN 模式到 PRI_RUN 模式的转换 .....	46
从 SEC_RUN 模式到 PRI_RUN 模式 (HSPLL)	
的转换 .....	45
从同步 .....	185
从休眠模式唤醒的转换 (HSPLL) .....	47
带有时钟仲裁的波特率发生器 .....	210
到 RC_RUN 模式的转换 .....	46
第一个启动位的时序 .....	211
EUSART/AUSART 同步发送 (主 / 从) 时序 .....	383
EUSART/AUSART 同步接收 (主 / 从) 时序 .....	383
发送和应答时的总线冲突 .....	217
发送间隔字符时序 .....	238
复位、看门狗定时器 (WDT)、振荡器起振定时器	
(OST) 和上电延时定时器 (PWRT) .....	372
故障保护时钟监视器 .....	292
缓慢上升时间 (MCLR 连接到 VDD, VDD 上升时间 >	
TPWRT) .....	55



i <sup>2</sup> C 从模式广播呼叫地址时序 (7 位或 10 位寻址模式) .....	206	程序存储器写时序要求 .....	371
i <sup>2</sup> C 从模式 (10 位发送) .....	201	EUSART/AUSART 同步发送要求 .....	383
i <sup>2</sup> C 从模式 (7 位发送) .....	198	EUSART/AUSART 同步接收要求 .....	383
i <sup>2</sup> C 从模式 (7 位或 10 位发送) .....	214	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要求 .....	372
i <sup>2</sup> C 从模式 (7 位接收, SEN = 0) .....	196	i <sup>2</sup> C 总线启动 / 停止位要求 (从模式) .....	379
i <sup>2</sup> C 从模式 (7 位接收, SEN = 0, ADMSK = 01011) .....	197	i <sup>2</sup> C 总线数据要求 (从模式) .....	380
i <sup>2</sup> C 从模式 ((10 位接收, SEN = 1) .....	205	MSSP i <sup>2</sup> C 总线启动 / 停止位要求 .....	381
i <sup>2</sup> C 从模式 (10 接收, SEN = 0) .....	199	MSSP i <sup>2</sup> C 总线数据要求 .....	382
i <sup>2</sup> C 从模式 (10 位接收, SEN = 0 且 ADMSK = 01001) .....	200	内部 RC 精度 .....	367
i <sup>2</sup> C 从模式 (7 位接收, SEN = 1) .....	204	PLL 时钟 .....	367
i <sup>2</sup> C 停止条件接收或发送模式 .....	216	SPI 从模式要求 (CKE = 1) .....	378
i <sup>2</sup> C 主模式 (7 位接收) .....	215	SPI 模式要求示例 (从模式, CKE = 0) .....	377
i <sup>2</sup> C 总线启动 / 停止位 .....	379	SPI 模式要求示例 (主模式, CKE = 0) .....	375
i <sup>2</sup> C 总线数据 .....	380	SPI 模式要求示例 (主模式, CKE = 1) .....	376
进入 SEC_RUN 模式的转换 .....	45	Timer0 和 Timer1 外部时钟要求 .....	373
进入休眠模式的转换 .....	47, 48	外部时钟要求 .....	366
MSSP i <sup>2</sup> C 总线启动 / 停止位 .....	381	时钟源 .....	37
MSSP i <sup>2</sup> C 总线数据 .....	381	复位后的默认系统时钟 .....	38
PWM 输出 .....	175	使用 OSCCON 寄存器进行选择 .....	38
启动条件期间的总线冲突 (仅 SDA) .....	218	数据存储器 .....	72
启动条件期间的总线冲突 (SCL = 0) .....	219	存储器映射 .....	
启动条件期间由 SDA 仲裁引起的 BRG 复位 .....	219	PIC18FX3J11/X4J11 器件 .....	73
SPI 从模式示例 (CKE = 0) .....	377	PIC18FX5J11 器件 .....	74
SPI 从模式示例 (CKE = 1) .....	378	特殊功能寄存器 .....	76
SPI 模式 (从模式, CKE = 0) .....	186	存储区选择寄存器 (BSR) .....	72
SPI 模式 (从模式, CKE = 1) .....	186	快速操作存储区 .....	75
SPI 模式 (主模式) .....	184	扩展的指令集 .....	85
SPI 主模式示例 (CKE=0) .....	375	特殊功能寄存器 .....	76
SPI 主模式示例 (CKE=1) .....	376	通用寄存器 .....	75
上电时的延时序列 (MCLR 连接到 VDD, VDD 电压上升时间 < TPWRT) .....	54	数据寻址模式 .....	82
上电时的延时序列 (MCLR 未连接到 VDD), 情形 1 .....	55	变址立即数偏移量 .....	85
上电时的延时序列 (MCLR 未连接到 VDD), 情形 2 .....	55	BSR .....	87
时钟 / 指令周期 .....	70	映射快速访问存储区 .....	87
时钟同步 .....	203	影响的指令 .....	85
双速启动时钟转换 (从 INTRC 切换到 HSPLL) .....	290	固有和立即数 .....	82
Timer0 和 Timer1 外部时钟 .....	373	间接寻址 .....	82
停止条件期间的总线冲突 (情形 1) .....	221	与使能了扩展的指令集的寻址模式对比 .....	86
停止条件期间的总线冲突 (情形 2) .....	221	直接寻址 .....	82
同步发送 .....	239, 254	双速启动 .....	279, 290
同步发送 (由 TXEN 控制) .....	255	双字指令 .....	
同步发送 (由 TXEN 位控制) .....	240	示例情形 .....	71
外部时钟 (除 PLL 之外的所有模式) .....	366	所有寄存器的初始化状态 .....	57 - 61
休眠模式下的自动唤醒位 (WUE) .....	237	<b>T</b> .....	
异步发送 .....	234, 251	TBLRD .....	337
异步发送 (背对背) .....	234, 251	TBLWT .....	338
异步接收 .....	236, 253	Timer0 .....	153
应答序列 .....	216	16 位读写模式 .....	154
正常工作状态下的自动唤醒位 (WUE) .....	237	工作原理 .....	154
执行 Sleep 指令时的外部存储器总线时序 (扩展单片机模式) .....	106, 108	时钟源边沿选择位 (T0SE 位) .....	154
执行 TBLRD 指令时的外部存储器总线时序 (扩展单片机模式) .....	106, 108	时钟源选择位 (T0CS 位) .....	154
重复启动条件 .....	212	相关的寄存器 .....	155
重复启动条件期间的总线冲突 (情形 1) .....	220	预分频比选择 (T0PS2:T0PS0 位) .....	155
重复启动条件期间的总线冲突 (情形 2) .....	220	预分频器 .....	155
主模式同步接收 (由 SREN 位控制) .....	241, 256	切换分配 .....	155
自动波特率计算 .....	232	预分频器分配 (PSA 位) .....	155
时序图和规范 .....		预分频器。参见预分频器, Timer0. .....	
捕捉 / 比较 / PWM 要求 (CCP1 和 CCP2) .....	374	中断 .....	155
CLKO 和 I/O 要求 .....	368, 370	Timer2 .....	163
		工作原理 .....	163
		PR2 寄存器 .....	175
		输出 .....	164
		TMR2 与 PR2 匹配中断 .....	175
		相关的寄存器 .....	164
		中断 .....	164

# PIC18F85J11 系列

Timer3 .....	165
16 位读 / 写模式 .....	167
工作原理 .....	166
TMR3H 寄存器 .....	165
TMR3L 寄存器 .....	165
特殊事件触发信号 (CCP) .....	167
相关的寄存器 .....	167
溢出中断 .....	165
振荡器 .....	165, 167
中断 .....	167
Timer1 .....	157
16 位读 / 写模式 .....	159
复位, 使用 CCPx 特殊事件触发信号 .....	160
工作原理 .....	158
TMR1H 寄存器 .....	157
TMR1L 寄存器 .....	157
相关的寄存器 .....	161
溢出中断 .....	157
振荡器 .....	157, 159
布线注意事项 .....	160
中断 .....	160
作为实时时钟使用 .....	160
作为时钟源使用 .....	159
TSTFSZ .....	339
特殊事件触发信号。参见比较 (CCP 模块)。	
同步间隔字符自动唤醒 .....	237
同步主接收模式 (由 SREN 位控制) .....	241, 256
<b>W</b>	
WCOL .....	211, 212, 213, 216
WCOL 状态标志 .....	211, 212, 213, 216
VDDCORE/VCAP 引脚 .....	288
WWW 地址 .....	405
WWW 在线技术支持 .....	7
外部存储器总线 .....	99
16 位模式时序 .....	106
16 位数据宽度模式 .....	102
16 位字节写模式 .....	103
16 位字节选择模式 .....	105
16 位字写模式 .....	104
21 位寻址 .....	101
8 位模式时序 .....	108
8 位数据宽度模式 .....	107
等待状态 .....	102
地址和数据宽度 .....	101
地址和数据线用法 (表) .....	101
地址平移 .....	101
端口引脚弱上拉 .....	102
和程序存储器模式 .....	102
I/O 端口功能 .....	99
控制 .....	100
在功耗管理模式下的操作 .....	109
外部振荡器模式 .....	39
HS 模式 .....	39
(EC 模式) .....	40
<b>X</b>	
系列中各器件的详细信息 .....	10
XORLW .....	339
XORWF .....	340
休眠	
OSC1 和 OSC2 引脚状态 .....	42
休眠模式下	
的比较器操作 .....	272

## Y

### 引脚功能

AVDD .....	20, 29
AVSS .....	20, 29
ENVREG .....	20, 29
MCLR .....	14, 21
RA0/AN0 .....	14, 21
RA1/AN1 .....	14, 21
RA2/AN2/VREF- .....	14, 21
RA3/AN3/VREF+ .....	14, 21
RA4/T0CKI .....	14, 21
RA5/AN4 .....	14, 21
RA6/OSC2/CLKO .....	14, 21
RA7/OSC1/CLKI .....	14, 21
RB0/INT0 .....	15, 22
RB1/INT1 .....	15, 22
RB2/INT2 .....	15, 22
RB3/INT3 .....	15
RB3/INT3/CCP2 .....	22
RB4/KBI0 .....	15, 22
RB5/KBI1 .....	15, 22
RB6/KBI2/PGC .....	15, 22
RB7/KBI3/PGD .....	15, 22
RC0/T1OSO/T13CKI .....	16, 23
RC1/T1OSI/CCP2 .....	16, 23
RC2/CCP1 .....	16, 23
RC3/SCK/SCL .....	16, 23
RC4/SDI/SDA .....	16, 23
RC5/SDO .....	16, 23
RC6/TX1/CK1 .....	16, 23
RC7/RX1/DT1 .....	16, 23
RD0/AD0/PSP0 .....	24
RD0/PSP0 .....	17
RD1/AD1/PSP1 .....	24
RD1/PSP1 .....	17
RD2/AD2/PSP2 .....	24
RD2/PSP2 .....	17
RD3/AD3/PSP3 .....	24
RD3/PSP3 .....	17
RD4/AD4/PSP4 .....	24
RD4/PSP4 .....	17
RD5/AD5/PSP5 .....	24
RD5/PSP5 .....	17
RD6/AD6/PSP6 .....	24
RD6/PSP6 .....	17
RD7/AD7/PSP7 .....	24
RD7/PSP7 .....	17
RE0/RD .....	18
RE0/RD/AD8 .....	25
RE1/WR .....	18
RE1/WR/AD9 .....	25
RE2/AD10/CS .....	25
RE2/CS .....	18
RE3 .....	18
RE3/AD11 .....	25
RE4 .....	18
RE4/AD12 .....	25
RE5 .....	18
RE5/AD13 .....	25
RE6 .....	18
RE6/AD14 .....	25
RE7/AD15/CCP2 .....	25
RE7/CCP2 .....	18
RF1/AN6/C2OUT .....	19, 26
RF2/AN7/C1OUT .....	19, 26

RF3/AN8 .....	19, 26	BC .....	307
RF4/AN9 .....	19, 26	BCF .....	308
RF5/AN10/CVREF .....	19, 26	BN .....	308
RF6/AN11 .....	19, 26	BNC .....	309
RF7/AN5/SS .....	19, 26	BNN .....	309
RG0 .....	20, 27	BNOV .....	310
RG1/TX2/CK2 .....	20, 27	BNZ .....	310
RG2/RX2/DT2 .....	20, 27	BOV .....	313
RG3 .....	20, 27	BRA .....	311
RG4 .....	20, 27	BSF .....	311
RH0/A16 .....	28	BSF (立即数变址寻址模式) .....	347
RH1/A17 .....	28	BTFSC .....	312
RH2/A18 .....	28	BTFSS .....	312
RH3/A19 .....	28	BTG .....	313
RH4 .....	28	BZ .....	314
RH5 .....	28	标准指令 .....	299
RH6 .....	28	CALL .....	314
RH7 .....	28	CLRF .....	315
RJ0/ALE .....	29	CLRWDT .....	315
RJ1/OE .....	29	COMF .....	316
RJ2/WRL .....	29	CPFSEQ .....	316
RJ3/WRH .....	29	CPFSGT .....	317
RJ4/BA0 .....	29	CPFSLT .....	317
RJ5/CE .....	29	操作码字段说明 .....	300
RJ6/LB .....	29	DAW .....	318
RJ7/UB .....	29	DCFSNZ .....	319
VDD .....	20, 29	DECF .....	318
VDDCORE/VCAP .....	20, 29	DECFSZ .....	319
Vss .....	20, 29	GOTO .....	320
因特网地址 .....	405	INCF .....	320
硬件乘法器 .....	111	INCFSZ .....	321
工作原理 .....	111	INFSNZ .....	321
简介 .....	111	IORLW .....	322
性能比较 .....	111	IORWF .....	322
预分频器, 捕捉 .....	172	LFSR .....	323
预分频器, Timer0 .....	155	MOVF .....	323
预分频器, Timer2 .....	176	MOVFF .....	324
<b>Z</b>		MOVLB .....	324
在线串行编程 (ICSP) .....	279, 293	MOVLW .....	325
在线调试器 .....	293	MOVWF .....	325
增强型通用同步 / 异步收发器 (EUSART)。参见 EUSART。		MULLW .....	326
振荡器		MULWF .....	326
切换 .....	37	NEGF .....	327
Timer1 为辅助时钟 .....	37	NOP .....	327
转换 .....	38	PIC18F85J11 系列 (表) .....	302
振荡器配置 .....	35	POP .....	328
EC .....	35	PUSH .....	328
ECPLL .....	35	RCALL .....	329
HS .....	35	RESET .....	329
HSPLL .....	35	RETFIE .....	330
INTOSC .....	35	RETLW .....	330
INTRC .....	35	RETURN .....	331
内部振荡器电路 .....	41	RLCF .....	331
振荡器起振定时器 (OST) .....	42	RLNCF .....	332
振荡器选择 .....	279	RRCF .....	332
振荡器, Timer3 .....	165	RRNCF .....	333
振荡器, Timer1 .....	157, 167	SETF .....	333
直接寻址 .....	83	SETF (立即数变址寻址模式) .....	347
指令集 .....	299	SLEEP .....	334
ADDLW .....	305	SWAPF .....	336
ADDWF .....	305	SUBFWB .....	334
ADDWFC .....	306	SUBLW .....	335
ADDWF (立即数变址寻址模式) .....	347	SUBWF .....	335
ANDLW .....	306	SUBWFB .....	336
ANDWF .....	307	TBLRD .....	337

# PIC18F85J11 系列

---

TBLWT .....	338
TSTFSZ .....	339
一般格式 .....	301
XORLW .....	339
XORWF .....	340
指令周期 .....	70
流 / 流水线 .....	70
时钟机制 .....	70
直流特性	
掉电电流和供电电流 .....	352
供电电压 .....	351
PIC18F85J11 系列 .....	360
中断 .....	113
INTx 引脚 .....	128
PORTB 电平变化中断 .....	128
期间, 现场保护 .....	128
TMR0 .....	128
中断源 .....	279
A/D 转换完成 .....	263
比较完成 (CCP) .....	173
捕捉完成 (CCP) .....	172
电平变化中断 (RB7:RB4) .....	132
TMR1 中断 .....	157
TMR2 与 PR2 匹配 (PWM) .....	175
TMR3 溢出 .....	165
中断, 标志位	
电平变化中断 (RB7:RB4) 标志 (RBIF 位) .....	132
主复位 (MCLR) .....	53
主同步串行端口 (MSSP)。参见 MSSP。	

## MICROCHIP 网站

Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的因特网浏览器即可访问。网站提供以下信息:

- **产品支持**——数据手册和勘误表、应用笔记和样本程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及存档软件
- **一般技术支持**——常见问题 (FAQ)、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

## 变更通知客户服务

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

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

## 客户支持

Microchip 产品的用户可通过以下渠道获得帮助:

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持

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。

# PIC18F85J11 系列

---

## 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  
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: PIC18F85J11 系列 文献编号: DS39774D\_CN

问题

1. 本文档中哪些部分最有特色?

---

---

2. 本文档是否满足了您的软硬件开发要求? 如何满足的?

---

---

3. 您认为本文档的组织结构便于理解吗? 如果不便于理解, 那么问题何在?

---

---

4. 您认为本文档应该添加哪些内容以改善其结构和主题?

---

---

5. 您认为本文档中可以删减哪些内容, 而又不会影响整体使用效果?

---

---

6. 本文档中是否存在错误或误导信息? 如果存在, 请指出是什么信息及其具体页数。

---

---

7. 您认为本文档还有哪些方面有待改进?

---

---

产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或销售办事处联系。

器件编号	X	/XX	XXX
器件	温度范围	封装	模式
器件	PIC18F63J11/64J11/65J11 <sup>(1)</sup> 、 PIC18F83J11/84J11/85J11 <sup>(1)</sup> 、 PIC18F63J11/64J11/65J11T <sup>(2)</sup> 、 PIC18F83J11/84J11/85J11T <sup>(2)</sup>		
温度范围	I = -40°C 至 +85°C （工业级）		
封装	PT = TQFP （薄型四方扁平封装）		
模式	QTP、SQTP、代码或特殊要求 （空白为其他情况）		

示例：

a) PIC18F85J11-I/PT 301 表示工业级温度，TQFP 封装，QTP 模式 301。

b) PIC18F63J11T-I/PT 表示卷带式，工业级温度，TQFP 封装。

注 1: F = 标准电压范围  
2: T = 仅卷带式封装

## 全球销售及服务中心

### 美洲

公司总部 **Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 1-480-792-7200  
Fax: 1-480-792-7277

技术支持:  
<http://support.microchip.com>  
网址: [www.microchip.com](http://www.microchip.com)

**亚特兰大 Atlanta**  
Duluth, GA  
Tel: 1-678-957-9614  
Fax: 1-678-957-1455

**波士顿 Boston**  
Westborough, MA  
Tel: 1-774-760-0087  
Fax: 1-774-760-0088

**芝加哥 Chicago**  
Itasca, IL  
Tel: 1-630-285-0071  
Fax: 1-630-285-0075

**克里夫兰 Cleveland**  
Independence, OH  
Tel: 1-216-447-0464  
Fax: 1-216-447-0643

**达拉斯 Dallas**  
Addison, TX  
Tel: 1-972-818-7423  
Fax: 1-972-818-2924

**底特律 Detroit**  
Farmington Hills, MI  
Tel: 1-248-538-2250  
Fax: 1-248-538-2260

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

**洛杉矶 Los Angeles**  
Mission Viejo, CA  
Tel: 1-949-462-9523  
Fax: 1-949-462-9608

**圣克拉拉 Santa Clara**  
Santa Clara, CA  
Tel: 1-408-961-6444  
Fax: 1-408-961-6445

**加拿大多伦多 Toronto**  
Mississauga, Ontario,  
Canada  
Tel: 1-905-673-0699  
Fax: 1-905-673-6509

### 亚太地区

亚太总部 **Asia Pacific Office**  
Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon  
Hong Kong  
Tel: 852-2401-1200  
Fax: 852-2401-3431

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

**中国 - 成都**  
Tel: 86-28-8665-5511  
Fax: 86-28-8665-7889

**中国 - 重庆**  
Tel: 86-23-8980-9588  
Fax: 86-23-8980-9500

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

**中国 - 南京**  
Tel: 86-25-8473-2460  
Fax: 86-25-8473-2470

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

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

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

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

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

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

**中国 - 厦门**  
Tel: 86-592-238-8138  
Fax: 86-592-238-8130

**中国 - 珠海**  
Tel: 86-756-321-0040  
Fax: 86-756-321-0049

**台湾地区 - 高雄**  
Tel: 886-7-213-7830

Fax: 886-7-330-9305

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

### 亚太地区

**台湾地区 - 新竹**  
Tel: 886-3-6578-300  
Fax: 886-3-6578-370

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

**印度 India - Bangalore**  
Tel: 91-80-3090-4444  
Fax: 91-80-3090-4123

**印度 India - New Delhi**  
Tel: 91-11-4160-8631  
Fax: 91-11-4160-8632

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

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

**韩国 Korea - Daegu**  
Tel: 82-53-744-4301  
Fax: 82-53-744-4302

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

**马来西亚 Malaysia - Kuala Lumpur**  
Tel: 60-3-6201-9857  
Fax: 60-3-6201-9859

**马来西亚 Malaysia - Penang**  
Tel: 60-4-227-8870  
Fax: 60-4-227-4068

**菲律宾 Philippines - Manila**  
Tel: 63-2-634-9065  
Fax: 63-2-634-9069

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

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

### 欧洲

**奥地利 Austria - Wels**  
Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

**丹麦 Denmark-Copenhagen**  
Tel: 45-4450-2828  
Fax: 45-4485-2829

**法国 France - Paris**  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

**德国 Germany - Munich**  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

**意大利 Italy - Milan**  
Tel: 39-0331-742611  
Fax: 39-0331-466781

**荷兰 Netherlands - Drunen**  
Tel: 31-416-690399  
Fax: 31-416-690340

**西班牙 Spain - Madrid**  
Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

**英国 UK - Wokingham**  
Tel: 44-118-921-5869  
Fax: 44-118-921-5820