



PIC16F87X

PIC16F87X 28/40 引脚, 8 位 CMOS FLASH 单片机

单片机包括:

PIC16F873

PIC16F874

PIC16F876

PIC16F877

单片机主要性能

- 具有高性能 RISC CPU。
- 仅有 35 条单字指令
- 除程序分支指令为两个周期外, 其余均为单周期指令
- 运行速度: DC-20MHZ 时钟输入
DC-200ns 指令周期
- 8K*14 个 FLASH 程序存储器
368*8 个数据存储器 (RAM) 字节
256*8EEPROM 数据存储器字节
- 引脚输出和 PIC16C73B/74B/76/77 兼容
- 中断能力 (达到 14 个中断源)
- 8 级深度的硬件堆栈
- 直接, 间接和相对寻址方式
- 上电复位 (POR)
- 上电定时器 (PWRT) 和振荡启动定时器 (OST)
- 监视定时器 (WDT), 它带有片内可靠运行的 RC 振荡器
- 可编程的代码保护
- 低功耗睡眠方式
- 可选择的振荡器,
- 低功耗、高速 CMOS FLASH/EEPROM 工艺
- 全静态设计
- 在线串行编程 (ICSP)
- 单独 5V 的内部电路串行编程 (ICSP) 能力。

处理机读/写访问程序存储器

- 运行电压范围 2.0V 到 5.5V
- 高吸入/拉出电流 25mA
- 商用, 工业用温度范围
- 低功耗:
 - 在 5V, 4MHz 时典型值小于 2mA
 - 在 3V, 32KHz 时典型值小于 20 μ A
 - 典型的稳态电流值小于 1 μ A

外围特征:

- **Timer0:** 带有预分频器的 8 位定时器/计数器
- **Timer1:** 带有预分频器的 16 位定时器/计数器, 在使用外部晶体振荡时钟时, 在 **Sleep** 期间仍能工作
- **Timer2:** 带有 8 位周期寄存器, 预分频器和后分频器的 8 位定时器/计数器
- 2 个捕捉器, 比较器, **PWM** 模块
 - 其中: 捕捉器是 16 位, 最大分辨率为 12.5ns
 - 比较器是 16 位, 最大分辨率为 200ns
 - PWM** 最大分辨率为 10 位
- 10 位多通道模/数转换器
- 带有 **SPI** (主模式) 和 **I2C** (主/从) 模式的 **SSP**
- 带有 9 位地址探测的通用同步异步接收/发送器 (**USART/SCI**)
- 带有 **RD**, **WR** 和 **CS** 控制 (只 40/44 引脚) 8 位字宽的并行从端口
- 带有降压复位的降压检测电路

目录:

1. 0 概述
2. 0 存储器结构
3. 0 输入/输出端口
4. 0 数据 EEPROM 和 FLASH 程序存储器
5. 0 **Timer0** 模块
6. 0 **Timer1** 模块
7. 0 **Timer2** 模块
8. 0 捕捉/比较/**PWM** 模块
9. 0 主同步串行端口模块
10. 0 通用同步异步接收/发送器
11. 0 模/数转换器模块
12. 0 **CPU** 特性
13. 0 指令系统
14. 0 支持开发工具
15. 0 电气特性
16. 0 **DC** 和 **AC** 特征图表
17. 0 封装情况

1. 0 单片机概述

这份资料包括单片机的特殊情况。另外可以从 PICmicro™ Mid-Range Reference Manual, (DS33023) 中获得信息, 也可以从当地供应商获得或者从 MICROCHIP WEBSITE 下载。参考手册应该看成是这份资料的补充, 为更好的理解单片机的结构和外部模块的特征, 推荐读参考手册。

这份资料包括 4 种单片机 (**PIC16F873**, **PIC16F874**, **PIC16F876**, **PIC16F877**), **PIC16F876/873** 是一个 40 引脚封装单片机。28 引脚的单片机没有一个并行从端口。

下面是单片机的两个结构框图; 28 引脚见图 1-1, 40 引脚的见图 1-2。28 引脚和 40 引脚的输出各自见表 1-1 和 1-2。

2. 0 存储器结构

在这些 PICmicro 处理器里有 3 个存储器块。程序存储器和数据存储器有它自己的总线，以便能够并行访问。详见这节。EEPROM 数据存储器块见 4. 0 节。

关于单片机存储器的附加情况可从 PICmicro-TM Mid-Range Reference Manual, (DS33023) 中发现信息。

2. 1: 程序存储器结构

PIC16F87X 有一个 13 位的程序计数器，它的寻址能力达 $8K \times 14$ 个程序存储器空间。PIC16F877/876 有 $8K \times 14$ 个 FLASH 程序存储器字和 PIC16F873/874 有 $4K \times 14$ 个 FLASH 程序存储器字。寻址超出上面可执行的地址将导致回绕。

复位向量在 0000h 种中断向量在 0004h。

2.2 数据存储器结构

数据存储器被分成多个块，它包括通用寄存器（GPR）和特殊功能寄存器（SFR），位 RP1 和 RP0 是块选择位。

RP1, RP0 (STATUS <6:5>)

=00 块 0

=01 块 1

=10 块 2

=11 块 3

每个块的范围达到 7Fh(128bit)，每个块的较低位存储单元保存 SFRS,在 SFRS 上面是 GPRS 做为静态 RAM 操作的。所有执行的块包括 SFR。一些经常使用的 SFRS 可以从一个块镜像到另外一个块来减少代码和实现快速访问。

注：EEPROM 数据存储器可在这份资料的第 4 节描述。

2.2.1 通用存储器

通用寄存器文件可以通过文件选择寄存器直接或间接的访问。

2. 2. 2 特殊功能寄存器

特殊功能寄存器（SFRS）是通过 CPU 和外围模块来控制单片机运行的寄存器。这些寄存器是做为静态 RAM 操作的，

特殊功能寄存器被分成两类：一类是与这节所说的“核心”功能有关的特殊功能寄存器，另一类是在这节所说与外围特性有关的外设寄存器。

2.2.2.1 状态寄存器

状态寄存器包括 ALU 的算术运算状态、复位状态和数据寄存器的块选择位。

和其它任何寄存器一样，状态寄存器可以做为任一指令的目标寄存器。如果在一个影响 Z、DC、C 位的指令中状态寄存器是一个目标寄存器的话，那么写入状态寄存器的这三位是禁止的，这些位可通过单片机的逻辑运算而清零或置 1。此外 TO 和 PD 位是不可写入的，因此，用状态寄存器作为目标寄存器，一条指令的执行结果跟预想的也许不同。

如 C LRF、STATUS 将清除高三位和将 Z 位置 1。这将使状态寄存器成为 000U U1UU (U 表示未变)

因此推荐用 BCF、BSF, SWAPF 和 MOVWF 指令来改变状态寄存器，因为这些指令不影响状态寄存器的 Z、DC 和 C 位。对于其它不影响状态位的指令见指令一览表。

注 1: C 和 DC 位在减法中作为各自的借位和辅助借位。例见 SUBLW 和 SUBWF 指令

2.2.2.2 选择寄存器

OPTION-REG 寄存器是一个可读写的寄存器，它包括各种控制位用来设定 TMR0/WDT 预分频器，（做为预分频器的分配寄存器）外部 INT 中断，TMR0 和在 PORTB 上的弱上拉。

注：为得到一个给 TMR0 寄存器的 1: 1 预分频器分配，将预分频器分配给监视定时器。

2.2.2.3 中断控制寄存器

INTCON 寄存器是可读写的寄存器,它包括各种对于 TMR0 寄存器溢出, RB 端口变化和外部 RB0 引脚中断的使能位和标志位。

注:当中断条件发生时,中断标志位置 1,不管它的相应使能位或全体使能位 GIE (INTCON <7>) 的状态。用户软件应该在使能一个中断之前保证合适的中断标志位清零。

2.2.2.4 PIE1 寄存器

这个寄存器包括进行外部中断的使能位。

注: (INTCON<6>)为使任何外部中断使能, 位 PEIE 必须置 1。

2.2.2.5 PIR1 寄存器

这个寄存器包括进行外部中断的单个标志位。

注:当中断条件发生时,中断标志位置 1,不管它的相应使能位或全体使能位 GIE (INTCON <7>) 的状态。用户软件应该在使能一个中断之前保证合适的中断标志位清零。

寄存器 2-5 (略)

2.2.2.6 PIE2 寄存器

这个寄存器包括对于 SSP 总线冲突和低电压探测中断的使能位。

寄存器 2-6 (略)

2.2.2.7 PIR2 寄存器

这个寄存器包括对于 SSP 总线冲突和低电压探测中断的标志位。

注:当中断条件发生时,中断标志位置 1,不管它的相应使能位或全体使能位 GIE (INTCON <7>) 的状态。用户软件应该在使能一个中断之前保证合适的中断标志位清零。

寄存器 2-7 (略)

2.2.2.8: PCON 寄存器

电源控制 (PCON) 寄存器包括在一个上电复位 (POR), 一个降压复位, 外部 MCLR 复位, 监视定时器 (WDT) 之间不同的标志位。

注: BOR 在 POR 上未知, 它必须通过用户置 1 和后继复位上检查看看 BOR 是否清零, 以表明出现一个溢出, 如果溢出电路不能使用, BOR 是一个随意位和不可预测的。

2.3: PCL 和 PCLATH

PC 是 13 位字宽, 它的低位字节叫做 PCL 寄存器, 它是一个可读写的寄存器。它的高位 (PC <12: 8>) 是不可读的。但通过 PCLATH 寄存器是可以间接写的。在任何复位上 PC 的较高位将被清零。图 2-3 给出了装入 PC 的两种情况。在上面的例子给出了在一个写入 PCL (PCLATH <4: 0> 到 PCL) 时 PC 是如何被装入的。在下面的例子给出了在一个 CALL 或 GOTO 指令期间 (PCLATH <4: 0> 到 PCH) 时 PC 是如何被装入的。

2.3.1: 计算 GOTO

一个计算 GOTO 指令通过向程序计数器增加一个位移来完成 (ADDWF PCL)。当用计算 GOTO 方法进行一个读表, 如果这个表的存储单元通过一个 PCL 存储器边界 (每个 256 字节块) 时, 必须小心操作。参考 “implementiy a Table Read” 的应用。

2. 3. 2: 堆栈

PIC16CXX 系列有一个 8 级深度 13 位字宽的硬件堆栈, 堆栈空间既不是程序空间的一部分, 也不是数据空间的一部分, 并且堆栈指针是不可读写的, 当执行一个 CALL 指令或者中断引起一个分支时, 程序计数器值被压入堆栈, 通过执行 RETURN、RETLW 或 RETFIE 指令, 堆栈内容被弹出, PCLATH 的内容不受 PUCH 或 POP 指令操作的影响。

堆栈是作为一个循环缓冲器来运行的, 这意味着在向堆栈压入 8 次内容之后, 第九次压入的内容

将覆盖第一次压入的存储在堆栈里的内容，第十次压入内容将覆盖第二次压入内容。（如此等等）

注：1.没有状态位表明堆栈溢出或堆栈不溢出的条件

注 2：没有指令助记符叫 PUSH 或 POP，它是通过执行 CALL，RETURN， RETLW 和 RETFIE 指令来产生动作或指向一个中断地址。

2.4:程序存储器的分页

PIC16CXX 单片机有连续寻址 8K 程序存储器字块的能力，在任何 2K 程序存储器页面之内，CaLL 和 GOTO 指令只能提供 11 个地址位来允许分支程序。当执行 CaLL 和 GOTO 指令时，地址的高 2 位由 PCLATH 〈4：3〉提供。当执行 CaLL 和 GOTO 指令时，用户必须保证页面选择位被合适编程以便所要求的程序存储器页面被寻址。如果执行从一个 CaLL 指令（或中断）的返回，整个 13 位程序计数器的内容被弹出堆栈。因此 PCLATH 〈4：3〉位的操作对于返回指令是不要求的。

和例 2-1 给出了在程序存储器的页面 1 里的一个子程序的访问。这个例子假定 PCLATH 通过中断服务程序被保存和重新存储。

2.5：间接寻址，INDF 和 FSR 寄存器

INDF 寄存器不是一个物理上的寄存器，寻址 INDF 寄存器将导致间接寻址。

间接寻址通过使用 INDF 寄存器而成为可能。任何使用 INDF 寄存器的指令实际上是通过文件选择寄存器 FSR 来访问所指向的寄存器。间接地读 INDF 寄存器（FSR=“0”）自身将会读 00h，间接地写入 INDF 寄存器将导致一个不运行（尽管状态位会被影响），一个有效的 9 位地址可通过连接 8 位 FSR 寄存器和 IRP 位（STATUS 〈7〉）来获得。见图 2-6（略）

用间接寻址的一个简单的清除 RAM 20h~2Fh 存储单元的程序见例 2-2。

3. 0 输入/输出端口

这些 I/O 端口引脚是多路复用的，它对于在单片机的外部特性具有可选择功能。通常当使用外部功能时，这个引脚就不可以作为正常 I/O 引脚使用。

3.1 端口 A 和 TRISA 寄存器

端口 A 是一个六位双向端口。相应的数据方向寄存器是 TRISA，将 TRISA 寄存器的一位置 1，把相应的端口 A 引脚变为输入，即把相应的输出驱动器置成高阻抗方式。对 TRISA 寄存器里的一位清零会把输出锁存器的内容放到所选择的引脚上。

读端口 A 寄存器读的是引脚的状态，反之，写入端口将是写向端口锁存器。所有写操作都是先读后写操作，因此，写入一个端口意味着这个端口引脚是先读入的，这个值经更改再写向端口数据锁存器。

引脚 RA4 对于 Timero 模块时钟输入变成 RA4/T0CK1 引脚是复用的。RA4/T0CK1 引脚是一个施密特触发器输入和一个漏极开路输出。

其它 端口 A 引脚对于模拟输入和模拟 VREF 输入是复用的。每个引脚的操作是通过在 ADCON1 寄存器里（A/D 控制寄存器 1）里对控制位清零/置 1 来选择。

注：在一个上电复位上，这些引脚都设定为模拟输入和读为 0。

TRISA 寄存器控制 RA 引脚的方向，即使当他们正被用作模拟输入时也一样。当用他们作为模拟输入时，用户必须保证在 TRISA 寄存器里的此位是置 1 的。

3. 2 端口 B 和 TRISB 寄存器

端口 B 是一个 8 位字宽的双向端口，相应的数据方向寄存器是 TRISB，把 TRISB 寄存器里的一位置 1 会把相应的输出驱动变成为一个高阻输入方式。把 TRISB 寄存器里的一位置 0 将会把输出锁存器的内容送到所选的引脚上。

PORTB 的 3 个引脚对于低电压编程功能；RB3/PGM，RB6/PGC 和 RB7/PGD 是复用的。这些引脚的选择功能在特别功能节描述。

端口 B 的每一个引脚都有一个内部弱上拉，一个单独控制位能够打开所有的弱上拉。这个可以通过对 RBPU (OPTION—REG<7>) 位清零来实现。当端口设置为一个输出时弱上拉自动关闭。上拉不能使用在上电复位。

这个中断能够将单片机从睡眠中唤醒。用户在中断服务程序中用下列方法清零中断。

a) 端口 B 的任何读写，这将结束不匹配条件；

b) 将标志 RBIF 位清零：

一个不匹配条件将持续使标志位 RBIF 置 1，读端口 B 将结束不匹配条件和允许标志位 RBIF 被清零。

变化中断特征要求对于按键唤醒操作和端口 B 只用作变化中断特征的操作。当使用变化中断特征时不要求对端口 B 的探测。

这种不匹配中断特征和在这四个引脚上可用软件设置上拉一起，使得很容易与键盘接口，从而使通过按键唤醒器件成为可能。参考 Embedded Control Handbook, “IMPLEMENTING WAKE ON KEY STROKE”。

RB0/INT 是一个外部中断输入引脚和通过使用 INTEDG 位 (OPTION-REG<6>) 来设置。

RB0/INT 在 12.10.1 节详细讨论。

3.3 端口 C 和 TRISC 寄存器

端口 C 是一个 8 位字宽的双向端口，相应的数据方向寄存器是 TRISC，把 TRISC 寄存器里的一位置 1 把相应的端口 C 引脚变为输入，即把相应的输出驱动变成为一个高阻输入方式。把 TRISC 寄存器里的一位置 0 将会把相应的端口 C 引脚变为输出，即把输出锁存器的内容送到所选的引脚上。

端口 C 对于几种外部功能是复用的，(表 3-5)，端口 C 引脚有施密特触发器输入缓冲器。

当 I2C 模块使用时，PORTC <3: 4> 引脚可以通过 CKE 位 (SSPSTAT <6>) 用标准 I2C 或用 SMBUS 来设置。

当外围功能使能时，对每个端口 C 引脚在设定 TRIS 位时应该小心，当其它外部超越 TRIS 位使一个引脚成为输入时，一些外部超越 TRIS 位使一个引脚成为输出，因为当外部功能能使时，TRIS 位超越是有效的，带有 TRISC 作为目标的先读后写指令 (BSF、BCF, XORWF) 应该避免，用户应该参考相关的正确设置 TRIS 位的章节。

3.4: 端口 D 和 TRISD 寄存器

注：PIC16F873/876 不提供端口 D。

端口 D 是一个 8 位端口，它带有施密特触发器输入缓冲器，每个引脚可单独设定为输入或输出。

端口 D 可以通过设置控制位 PSPMODE (TRISE <4>) 而被设置为一个 8 位字宽的单片机端口 (并行从端口)。用这种方式输入缓冲器是 TTL。

3. 5: 并行从端口 (PSP)

并行从端口在 PIC16F873/876 上不执行。

端口 D 作为一个 8 字宽并行从端口 (PSP)，或当控制位 PSP、MODE (TRISE <4>) 置 1 时作为微处理器端口。用从方式，通过外部字：用 RD 控制输入引脚 RE0/RD 和用 WR 控制输入引脚 RE1/WR 使它可同步读写。

它可以直接接口到 8 位单片机的数据总线上，外部单片机能够读或写作为 8 位锁存器的端口 D 锁存器。将位 PSPMODE 置 1 能使端口引脚 RE0/RD 成为 RD 输入，RE1/WR 成为 WR 输入和 RE2/CS 成为 CS (片选) 输入，对这种功能相应的 TRISE 寄存器 (TRISE <2: 0>) 数据方向位必须设置为输入 (置 1) 和 A/D 端口设置位 PCFG <2: 0> (ADCON1 <2: 0>) 必须置 1，它将设置引脚 RE <2: 0> 设定为数字输入/输出。

实际上有两个 8 位锁存器，一个用作数据输入 (从 PICMICRO 微处理器) 和一个用作数据输出。用户写 8 位数据到端口 D 数据锁存器和从端口引脚锁存器读数据 (注意他们有相同的地址)。用这种方式，TRISD 寄存器被忽略了，因为微处理器正在控制数据流的方向。

当 CS 和 WR 线都被首先探测为低电平时，发生一个写入并行从属端口。当 CS 或 WR 线两者变成高电平 (电平触发器) 时，这时输入触发器全静态 (IBF) 标志位 (TRISE <7>) 在 Q4 时钟周期时被置 1，在下一个 Q2 周期之后，标志写完成 (图 3-10)。中断标志位 PSPIF (PIR1 <7>) 在相同的 Q4 时钟上也被置 1，IBF 通过读端口 D 输入锁存器也能被清零。如果当先前字节还没有被读出缓冲器时，尝试第二次写入 PSP，这时输入缓冲器溢出 (IBOV) 状态标志位 (TRISE <5>) 被置 1。

当 CS 和 RD 线是首先被探测为低电平时，发生一个从 PSP 读出。输出缓冲器全静态标志位 (TRISE <6>) 被立即清零 (图 3-11) 以表明端口 D 锁存器通过外部总线正在等待被读。当 CS 和 RD 引脚都变成高电平 (电平触发器) 时，中断标志位 PSPIF 在 Q4 时钟周期上是置 1 的，在下一个 Q2 周期之后，表明读操作完成。OBF 仍保持低电平直到通过用户固件将数据写向端口 D。

当不用 PSP 方式时，IBF 和 OBF 位是保持清零的，然而，如果状态位 IBOV 先前被置 1，它必须用固件清零。

当一个读写操作完成后，产生一个中断和被锁存到标志位 PSPIF。PSPIF 必须通过用户用软件清零，通过中断使能位 PSPIF (PIE <7>) 清零而使中断使能。

4. 0: EEPROM 数据存储器 and 快速程序寄存器

Data EEPROM 和快速程序寄存器中 (在整个 VDD 范围上) 在。的正常运行期间是可读写的。大容量的擦写不可以从用户代码中产生 (包括移动代码保护)。数据存储器不直接映射在寄存器文件空间里，而是通过特殊功能寄存器 (SFR) 间接寻址的。

有六个 SFRS 寄存器用来读和写 EEPROM 程序和数据存储器，它们是：

- EECON1
- EECON2
- EEDATA
- EEDATH
- EEADR
- EEADRH

EEPROM 数据存储器允许字节读和写，当和数据存储器块接口时，EEDATA 保存 8 位读和写数

据，EEADR 保存被访问的 EEPROM 存储单元的地址。寄存器 EEDATH 和 EEADRH 不用作 EEPROM 数据存储器的访问。这些单片机有高达 256 个带有 0h~FFh 的地址范围的 EEPROM 数据存储器字节。

EEPROM 数据存储器有高速的擦和写周期。写时间由片内定时器控制。写时间随着电压和温度以及不同的单片机将有变化，请参考限制说明。

程序存储器允许按字读和写。程序存储器允许对检查计算和校准表存储器的访问。写一个字节或字会自动地擦除存储单元（擦除写之前的存储单元）和写入一个新的数据。写入程序存储器将停止单片机运行直到写入完成。程序存储器在写期间不能被访问，因此代码不能执行。在写操作期间，振荡器继续运行外部时钟，因此它们继续运行。中断事件将被探测到和需要“排队”，直到写完成。当写完成时，在流水线上的下一个指令被执行，程序将分支到中断向量地址。

当和程序存储器块接口时，EEDATH: EEDATA 寄存器组成一个两字节字。它为读/写保存 14 位数据。EEADRH: EEADR 寄存器组成一个两字节字，它保存被访问的 EEPROM 存储单元的 13 位地址。这些单片机能有达到 8K 带有从 0h 到 3FFFh 地址范围的 EEPROM 程序字。在 EEDATH 和 EEDATA 寄存器上未用的较高位都应读作“0”。

4. 8. 1: EEPROM 数据存储器

当单片机不要求写入 EEPROM 数据存储器时有一些条件，为了防止寄生的 EEPROM 写，各种结构已建立，在上电时，WREN 位被清零，并且上电定时器（72ms）持续时间防止 EEPROM 写。

写初始化顺序和 WREN 位一起帮助防止电压降低，电源误操作和软件故障期间的意外写入。

4. 8. 2: 快速程序存储器

为防止寄生写入快速程序存储器，在设定字里的 WRT 位可以通过编程为 0 以防止写入，写初始顺序也必须跟在后面，WRT 和设定字通过用户代码不能被编程，只有通过外部编程器的使用被编程。

4. 9: 在代码保护期间的操作

每一个重新可编程存储器块有它自己的代码保护结构，外部的读和写操作不能使，如果这些结构中的任一个能使。

4. 9. 1: EEPROM 数据存储器

微处理器自己能够到内部 EEPROM 数据存储器里读和写。而不管代码保护设定位的状态如何。

4. 9. 2: 程序快速存储器

单片机脱离内部快速程序存储器能够读和执行指令，而不管代码保护设定位的状态如何，然而 WRT 设定位和代码保护位在写入程序存储器上有不同的影响，表 4-1 给出了各种设定和读写状态。为了擦除 WRT 或在设定字里代码保护位要求单片机全部被擦除。表 4-1、表 4-2 略

5. 0 : Timer0 模块

Timer0 模块定时器/计数器有下列特征：

- 8 位定时器/计数器寄存器。
- 可读写。
- 8 位软件可编程预分频器。
- 内部或外部时钟选择。
- 中断在从 FFH 到 00H 溢出
- 外部时钟为边缘选择。

图 5-1 是 Timer0 模块的方块框图和预分频器分给 WDT。

在 PICMICROTTM 中等 MCU 系列参考资料中关于 TIMERO 模块的附加情况在 PICMICROTTM 中等 MCU 系列参考资料中可得到的。

通过对 TOCS 位（OPTION-REG 〈5〉）清零可选择定时器方式，在定时器方式时，Timer0 模块在每个指令周期加 1（无预分频器）。如果是在写入 TMRO 寄存器时，递增将会延迟两个指令周期发生。用

户可以通过向 TMR0 寄存器写入一个调整值来解决。

计数器方式通过对 TOCS 位 (OPTION-REG <5>) 置 1 来选择, 用计数器方式时, Timer0 在 TOCK1 引脚的每个上升/下降边沿递增 1。递增边沿通过 Timer0 源边沿选择位 TOSE (OPTION-REG <4>) 来决定, 通过对 TOSE 位清 0 来选择上升边沿, 外部时钟的输入限定在 5.2 节详细讨论。

预分频器可以在定时器模块和监视定时器之间分配, 预分频器是不可读写的, 5.3 节将详细说明预分频器的运行。

5.1 Timer0 中断

当 TMR0 寄存器从 FFh 到 00h 溢出时将产生 TMR0 中断。这个溢出将位 TOIF (INTCON <2>) 置 1。可以通过对 TOIE (INTCON <5>) 位清零来屏蔽中断, 在重新使能这个中断之前, 位 TOIF 必须通过 TIMER0 模块中断服务程序清零。TMR0 中断不能将单片机从睡眠中唤醒, 因为在睡眠期间, 定时器是关闭的。

5.2: 使用带有外部时钟的 Timer0。

当没有使用预分频器时, 外部时钟输入与预分频器输出相同。通过采样在内部相位时钟的 Q2 和 Q4 周期时的预分频器输出, 来实现 TOCK1 与内部相位时钟 (TOSC) 同步。因此, 对 TOCK1 变成高电平至少需要 2TOSC (一个 20ns 的 RC 延迟) 和成为低电平至少需要 2TOSC (一个 20ns 的 RC 延迟)。参考所要求单片机的电气特性。

5.3: 预分频器

只有一个预分频器在 Timer0 模块和监视定时器之间相互独自分享, 当预分频器分配给 Timer0 模块, 意味着将没有预分频器分给监视定时器和 Vice-Versa 使用, 这种预分频器是不可读写的, 见图 5-1

PSA 和 PS0~PS2 位 (OPTION<3:0>) 决定了预分频器的分配对象和分频频率。

当预分频器分配给 Timer0 模块时, 所有写入 Timer0 寄存器的指令将使预分频器清零 (如 CLRF1, MOVWF 1, BSF 1, X 等) 当预分频器分配给 WDT 时, CLRWDI 指令在对预分频器清零时, 也对 WDT 清零。预分频器既不可读也不可写。

注: 当预分频器分配给 Timer0 模块时, 写入 TMR0 将预分频器清零, 但不改变预分频器分配。

6.0 :Timer1 模块

Timer1 模块由两个 8 位可读写的寄存器 (TMR1H 和 TMR1L) 组成的一个 16 位定时器/计数器。寄存器对 (TMR1H: TMR1L) 从 0000H 递增到 FFFFH 然后再转回到 0000H。当相应中断能使时, 可产生溢出中断, 并锁存在中断标志位 TMR1IF (PIR1 <0>)。通过对 TMR1 中断使能位 TMR1IE (PIE1 <0>) 置 1 和清零来确定这种中断使能或不使能。

Timer1 能够下列两种方法之一来操作。

1. 定时器方式
2. 计数器方式

运行方式可通过时钟选择位 TMR1CS (T1CON <1>) 来决定。

在定时器方式时, TIMER1 在每个指令周期递增 1, 在计数器方式时, 它在每个外部时钟输入的上升延递增 1。

Timer1 可通过对控制位 TMR1ON (T1CON <0>) 置 1 和清零来确定这种中断使能或不使能。

Timer1 也有一个中断“复位输入”。这种复位可以通过两个 CCP 模块中的一个来产生。(8.0 节) 寄存器 7-1 表示 Timer1 控制寄存器。

当 Timer1 振荡器使能时 (T1OSCEN 置 1), RC1/T1SI。CCP2 和 RC0/T1OSO/T1CK1 引脚变为输入, 这时, TRISC <1> 的值是可忽略的。

在中等系列参考手册(DS33023)中, 在定时器模块上的另一个情况是可用的。

6.1 Timer1 在定时器方式的运行

通过对 TMR1CS (T1CON <1>) 清零来选择定时器方式。用这种方式, 向定时器输入的时钟

是 FOSO/4。因为中断时钟一直是同步的，所以同步控制位 T1SYNC (T1CON <2>) 没有受影响。

6.2: Timer1 计数器运行

根据对位 TMR1CS 置 1，Timer1 可以运行在同步或异步模式。

当 Timer1 正在执行一个外部时钟源时，在一个上升边沿发生递增。在 Timer1 用计数器模式能运行时，在计数器开始递增之前，模块必须首先有一个下降边沿。

6.3: Timer1 在同步计数器方式的运行

通过对位 TMR1CS 置 1 来选择计数器方式，用这种方式，当 T1OSCEN 位置 1 时，定时器在 RC1/T1OSI 引脚上每个时钟输入的上升边沿递增加 1。或者当 T1OSCEN 位清零时，定时器在 RC0/T1OSO/T1CK1 引脚上每个时钟输入的上升也延递增加 1。

如果 T1SYNC 是清零的，那么外部时钟输入和内部相位时钟是同步的。在预分频器级之后同步运行，预分频齐级是一个同步脉动计数器。

在这种设置时，在 SLEEP 期间，既使出现外部时钟，TIMER1 将不会递增加 1，这是因为同步线路被切断，然而预分频器将继续递增。

6.4: Timer1 在异步计数器模式下操作

如果位 T1SYNC (T1CON <2>) 置 1，外部时钟输入不同步。定时器继续递增以便和内部相位时钟同步。定时器在睡眠期间继续运行和在溢出时产生中断，这个中断能够唤醒处理器。然而，用软件读/写定时器时需要注意。

异步计数器方式时，Timer1 不能用作捕捉器或比价器运行的时基。

6.4.1 用异步计数器模式读和写 TIMER1

当定时器从一个外部异步时钟运行时，读 TMR1H 或 TMR1L 将保证一个有效的（用硬件小心）。然而，用户应该记住用两个 8 位值本身读 16 位造成一定的问题，因为定时器在读之间可以溢出。

对于写，建议用户简单的停止定时器后写入所要求的值。当寄存器的值递增时，通过写入定时器寄存器出现一个写竞争。这样在定时器寄存器里可以产生一个预想不到的值。

读 16 位值要求一些小心，在 PICMICROTM MID-RANGE MCU 系列参考资料里的例 12-2 和例 12-3 给出当 TIMER1 运行在异步计数器模式时如何读和写。

6. 5: TIMER1 振荡器

一个晶体电路建立在 T1OSI(输入)和 T1OSO(放大器输出)。它通过对控制位 T1OSCEN(T1CON <3>) 置 1 而使能。振荡器是一个低功耗振荡器，其频率可达 200KHZ。在睡眠期间继续运行。基本预想使用一个 32KHZ 的晶体振荡器，表 6-1 给出 TIMER1 振荡器的电容选择。

Timer1 振荡器和 LP 振荡器是有区别的。用户必须提供一个软件时间延迟以保证的振荡器合适启动。

6.6: 使用一个 CCP 输出复位 TIMER1

如果 CCP1 模块设置为比较器模块来产生一个“特别事件触发器”(CCP1M <3: 0>=1011)，这个信号将复位 TIMER1。

注：从 CCP1 模块里产生的特别事件触发器不会将中断标志位 TMR1IF (PIR <0>) 置 1。

为充分利用着个特征，TIMER1 必须设置为定时器或异步计数器模式。如果 TIMER1 运行在异步计数器模式时，这个复位操作可以不工作。

在写入 TIMER1 事件里与从 CCP1 里产生的特别事件触发器相符合，写将优先。

在这种操作模式中，CCPR1H: CCPR1L 寄存器对使 TIMER1 有效地成为周期寄存器。

6.7: TIMER1 寄存器对 (TMR1H, TMR1L) 的复位

除通过 CCP1 特别事件触发器外。TMR1H, TMR1L 寄存器在 POR 或其它复位时都不会复位到 00h。

T1CON 寄存器在一个上电复位或一个降压复位时，复位到 00H。它关闭定时器和产生一个 1: 1 预分频器。在所有其它复位时，寄存器不受影响。

6.8 TIMER1 预分频器

预分频器计数器在写入 TMR1H 或 TMR1L 寄存器时被清零。

7. 0: Timer2 模块

Timer2 是一个带有预分频器和后分频器的 8 位定时器，它能用作 CCP 模块的 PWM 模式 PWM 的时基，TMR2 寄存器是可读写的，和任何单片机的复位都有将其清零。

输入时钟 (FOSC/4) 有 1:1, 1:4 或 1:16 的分频值选择，可通过控制位 T2CKPS <1:0> (T2CON <1:0>) 来选择。

Timer2 模块有一个 8 位周期 PR2。Timer2 从 00h 递增到它与 PR2 内数值相等时，然后复位到 00h 进行下一次递增周期，PR2 是可读写寄存器，PR2 寄存器通过复位初始化到 FFh。

TMR2 的匹配输出通过一个 4 位后分频器 (给出 1:1 到 1:16 的范围) 产生一个 TMR2 中断 (被锁存在标志位 TMR2IF, (PIR1 <1>))。

通过对控制位 TMR2ON (T2CON <2>) 的清零来关闭 Timer2 以达到最小功率消耗。

寄存器 7-1 给出了 Timer2 控制寄存器。

详见 DS33023

7.1 Timer2 预分频器和后分频器

当下列任何一条发生时，预分频器和后分频器计数器都会被清零。

- 。写入 TMR2 寄存器
- 。写入 T2CON 寄存器
- 。任何单片机复位 (POR、MCLR 复位、WDT 复位或 BOR)

当 T2CON 被写时，TMR2 不被清零。

7.2 TMR2 输出

TMR2 的输出 (在后分频器之前) 也可用作同步串口通信的模块，操作使用它将产生一个移位时钟。

8.0 捕捉器/比较器/PWM 模块

每个捕捉器/比较器/PWM 模块都包括一个 16 位的寄存器，这个寄存器可以作为：

- 16 位捕捉寄存器
- 16 位比较寄存器
- PWM 主/从运行周期寄存器

特别事件触发器的运行除外，CCP1 和 CCP2 两个模块在运行中是有区别的。

表 8-1 给出了 CCP 模块的资源和相互作用，在后面章节中，描述 CCP 模块的操作。

CCP1 模块

捕捉器/比较器/PWM 寄存器 1 (CCPR1) 是由两个 8 位寄存器组成，即 CCPR1L (低位字节) 和 CCPR1H (高位字节)。CCP1CON 寄存器控制 CCP1 的运行，通过比较相等时产生特别事件触发器和将对 Timer1 复位和启动一个 A/D 转换器。(如果 A/D 模块使能)

CCP2 模块

捕捉器/比较器/PWM 寄存器 2 (CCPR2) 是由两个 8 位寄存器组成，即 CCPR2L (低位字节) 和 CCPR2H (高位字节)，CCP2CON 寄存器控制 CCP2 的运行，通过比较相等时产生一个特别事件触发器和将 Timer1 复位以及 A/D 转换器开始转换 (如果 A/D 模块能使)。

关于 CCP 模块的其它情况可在 PICMICRO™ 中等 MCU 系列参考资料中和在“CCP 模块的使用”中可得到

8.1 捕捉器模式

在捕捉器模式时，当在引脚 RC2/CCP1 上发生一个事件时 CCPR1H:CCPR1L 将会捕捉 16 位 TMR1 寄存器的值。

一个事件指的是：

1. 每个下降边沿.
2. 每个上升边沿.
3. 每第 4 个上升边沿
4. 每第 16 个上升边沿

通过控制位 CCP1M <3:0> (CCP1CON <3:0>) 来选择上述事件，当捕捉器使用时，中断请求标志

位 CCP1IF(PIR1 <2>) 是置 1 的, 中断标志位必须用软件清零。如果在寄存器 CCPR1 里的值被读之前发生另外一个捕捉事件, 那么旧的捕捉值将会丢失。

8.1.1 CCP 引脚设置

在捕捉器方式时, RC2/CCP1 引脚通过对 TRISC <2> 位置 1 而被设置为输入。

注: 如果 RC2/CCP1 引脚设置为输出, 一个写入这个端口能导致捕捉发生。

8.1.2 Timer1 模式选择

为 CCP 模块使用捕捉功能时, Timer1 必须工作在定时器方式或者同步计数器方式, 在异步计数器方式时, 捕捉器不会工作。

8.1.3 软件中断

当捕捉方式改变时, 一个错误捕捉中断或许产生。用户应该保持 CCP1IE (PIE1 <2>) 位清零以避免错误的中断, 和在运行方式中的任何这样变化之后应该对标志位 CCP1IF 清零。

8.1.4 CCP 预分频器

有四种预分频器设置可通过 CCP1M <3:0> 位来指定。无论 CCP 模块是关闭还是 CCP 模块不在捕捉方式, 预分频计数器都是清零的, 任何复位将会对预分频计数器清零。

从一个捕捉预分频器向另一个捕捉预分频器转换时将产生一个中断, 同时预分频计数器将不被清零。因此, 首次捕捉可以从一个非零预分频器开始。例 8-1 给出了在捕捉预分频器之间转换的推荐方法。

8.2 比较器模块

在比较器模式中, 16 位 CCPR1 寄存器的值不断地和 TMR1 寄存器对的值相比较, 当两者出现相等时, RC2/CCP1 引脚是:

1. 变为高
2. 变为低
3. 保持不变

在这个引脚上的动作是基于控制位 CCP1M <3:0> (CCP1CON <3:0>) 的值。同时, 中断标志位 CCP1IF 是置 1R 的。

8.2.1 CCP 引脚的设定

用户必须通过对 TRISC <2> 位清零来设置 RC2/CCP1 引脚,

注: 对 CCP1CON 寄存器清零将迫使 RC2/CCP1 比较器输出锁存到不足低电平, 这不是数据锁存器。

8.2.2: Timer1 模式选择

如果 CCP 模块正在使用比较器功能时, Timer1 必须运行在定时器方式或同步计数器方式, 在异步计数器方式时, 比较器运行不会工作。

8.2.3: 软件中断方式

当通用软件中断方式被选择时, CCP1 引脚不受影响, CCPIF 位置 1 会引起一个 CCP 中断 (如果能使)

8.2.4 特别事件触发器

用这种方式, 产生一个内部硬件触发器, 它可用作初始化一次动作。

CCP1 的特别事件触发器的输出将 TMR1 寄存器对复位, 这样为 Timer1 允许 CCPR1 寄存器有效地成为一个 16 位可编程周期寄存器。

CCP2 的特别事件触发器输出将 TMR1 寄存器对复位和开始一个 A/D 转换 (如果 A/D 模块能使用时)

注: 从 CCP1 和 CCP2 模块中的特别事件触发器将不会中断标志位 TMR1IF (PIR1 <0>)

8.3: PWM 模块 (PWM)

在宽脉冲调制方式中, CCPX 引脚可产生一个 10 位分辨率的 PWM 输出, 因为 CCP1 引脚对端口数据锁存器是复用的, 因此 TRISC <2> 位必须清零以使 CCP1 引脚成为输出。

注: 对 CCP1CON 寄存器清零将会强迫 CCP1PWM 输出锁存器达到不足低电平。这个不是端口 C 输入/输出数据锁存器。

图 8-3 给出了 CCP 模块在 PWM 方式时的简单方块结构图。

8.3.3 节给出了对于 PWM 如何建立 CCP 模块的步骤过程。

一个 PWM 输出 (图 8-4) 有一个时间基准 (周期) 和一个输出高电平阶段的时间 (运行周期), PWM 的频率是周期的倒数 (即 1/周期)。

8.3.1: PWM 周期

通过写入 PR2 寄存器来确定 PWM 周期, PWM 周期可以用下列公式计算:

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

$$\text{PWM 频率} = 1 / [\text{PWM 周期}]$$

当 TMR2 等于 PR2 时，在下一个递增周期将产生下面三个事件：

1. TMR2 被清零。
2. CCP1 引脚被置 1（例外的是：如果 PWM 的运行周期=0%，CCP1 引脚将不会置 1）
3. PWM 运行周期从 CCP1L 到 CCP1H 被锁存。

注：Timer2 后分频器（见 8.1 节）不用来决定 PWM 频率，能被使用的后分频器与 PWM 输出相比，在不同频率时有一个伺服修改速率。

8.3.2: PWM 运行周期

PWM 运行周期可通过写入 CCP1L 寄存器和到 CCP1CON <5:4> 位来指定。它可达到 10 位分辨率，CCP1L 包括 8 个 MSBS 和 CCP1CON <5:4> 包括两个 MSBS，这个 10 位值通过 CCP1L:CCP1CON <5:4> 表示，下面等式被用作计算 PWM 运行周期

$$\text{PWM 运行周期} = (\text{CCP1L:CCP1CON} \langle 5:4 \rangle) \times T_{OSC} \cdot (\text{TMR2 预分频值})$$

CCP1L:CCP1CON <5:4> 在任何时间都是可写的，但是运行周期值直到 PR2 和 TMR2 之间相等发生后才能锁存到 CCP1H，（即周期完成）在用 PWM 方式时，CCP1H 是一个只读寄存器。

CCP1H 寄存器和一个 2 位内部锁存器是用作双重缓冲 PWM 运行周期，这个双重缓冲对于误操作 PWM 是需要的。

当 CCP1H 寄存器和 2 位匹配 TMR2 的锁存器用一个内部 2 位 Q 时钟或 2 位 TMR2 预分频器相连时，CCP1 引脚被清零。

所给 PWM 频率的 PWM 的最大分辨率：

$$\text{分辨率} = \log(\text{FINT}/\text{FPWM}) / \log(2) \quad (\text{位})$$

注：如果 PWM 运行周期值比 PWM 周期长，CCP1 引脚将不会清零。

8.3.3: PWM 运行的建立

设定 CCP 模块为 PWM 运行的可采用下面几步：

- 1 通过写向 PR2 寄存器将 PWM 周期置位。
- 2 通过写入 CCP1L 寄存器和 CCP1CON <5:4> 位将 PWM 运行周期置位。
- 3 通过对 TRISC <2> 位清零将 CCP1 引脚变为输出。
- 4 通过写入 T2CON 将 TMR2 预分频值和 Timer2 使能置位。
- 5 为 PWM 运行设定 CCP1 模块。

9.0: 主同步串行端口模块

同步串行端口模块 是一个串行接口，它对于和其它外部器件或单片机之间通讯非常有用。这些外部器件可以是串行 EEPROMS，移位寄存器，显示驱动器，A/D 转换器，等等。MSSP 模块能够用下列两种模块之一运行：

- 串行外部接口（SPI）
- 内部集成电路（I²C）

图 9-1 给出 SPI 模式的方框图，而图 9-5 和图 9-9 给出两种不同的 I²C 模式的运行框图。

9.1: SPI 模式

SPI 方式允许 8 位数据同时的同步发送和接收。所有 SPI 的四种模式都被支持。为完成通讯使用典型的三个引脚是：

- 串行数据输出（SDO）
- 串行数据输入（SDI）
- 串行时钟（SCK）

当用从方式运行时，另外第 4 个引脚可以使用。

- 从选择（SS）

当初始化 SPI 时，几种选项需要确定，这可通过用 SSPCON 寄存器（SSPXON <5:0> 和 SSPSTAT <7:6>）编程合适的控制位来进行。这些控制位允许指定下列情况：

- 主模式（SCK 是时钟输出）
- 从模式（SCK 是时钟输入）
- 时钟极性（SCK 在空闲状态）
- 数据输入采样相位（数据输出时间的中间或最后）

- 时钟边沿（在 SCK 的上升/或下降边沿）
- 时钟速率（只主方式有）
- 从选择方式（只从方式有）

图 9-4 给出了当在 SPI 模式时，MSSP 模块的方框图。

为使串行端口能使，MSSP 能使位 SSPEN（SSPCON<5>）必须置 1，为设定和复位 SPI 方式，对使能位 SSPEN 清零，对 SSPCON 寄存器重新初始化，然后对使能位 SSPEN 置 1，这样设定 SDI、SDO、SCK 和 SS 引脚作为串行端口引脚，对于引脚作为端口功能运转，他们必须对他们的数据方向位（在 TRIS 寄存器里）适合编程，这些是：

- SDI 通过 SPI 模块自动控制
- SDO 必须有 TRISC <5> 位清零
- SCK（主模式）必须有 TRISC <3> 位清零
- SCK（从模式）必须有 TRISC <3> 位置 1
- SS 必须有 TRISB <5> 置 1，

任何那些不需要的串行端口功能通过对合适的数据方向寄存器编程产生相反值可以被超

9. 1. 1 主模式

主机在任何时间里都能够初始化数据传送，因为它控制 SCK，主机决定从机（处理器 2，图 9-5）通过软件协议什么时候广播数据。

在主机方式时，数据一写入 SSPBUF 时，数据就被发送/接收。如果 SPI 只是接收，SDO 输出将不使能（作为一个输入编程），SSPSR 寄存器在 SDI 引脚上在当前的信号所编程的时钟速率连续地移位。因每个字节被接收，它将被装入到 SSPBUF 寄存器就象普通的接收字节一样（中断和状态位合适的置 1），这在作为“线路活动监视器”方式中一个接收的应用是很有用。

通过合适的编程 CKP 位（SSPCON <4>）来选择时钟极性，用图 9-6 和图 9-8 和图 9-9 给出 SPI 通讯的波形，在这里 MSB 是首次发送，用主模式中 SPI 时钟速率（位速率）对于下列几种之一是用用户编程选择的。

- FOSC/4（或 Tcy）
- FOSC/16（或 4.Tcy）
- FOSC/64（或 16.Tcy）
- Timer2 输出/2

这样允许一个最大 5MHz 的位时钟频率（在 20MHz 时）。

图 9-6 给出了主模式的波形。当 CKE=1 时，在 SCK 上有一个时钟边沿之前，SDO 数据是有效的。给出了在 SMP 位的状态上的基础输入采样的变化。给出了 SSPBUF 装入所接受的数据的时间。

9. 1. 2 从模式

在从方式时，数据的发送和接收作为外部时钟脉冲出现在 SCK 上。当最后一位被锁存，中断标志位 SSPIF（PIR1 <3>）是置 1 的。

当在从模式是外部时钟由在 SCK 引脚上的外部时钟源供给。这个外部时钟必须符合在电特性里规定的最小的最高或最低时间。

当在睡眠方式时，从模式可以发送/接受数据。当接受一个字节时，单片机将从睡眠中唤醒。

注：当 SPI 是从模式时，SS 引脚使能，（SSPCON <3: 0>=0100）如果 SS 引脚置位到 VDD，SPI 模块将复位。

注：如果 SPI 在从模式时使用，CKE=“1”，那么 SS 引脚控制必须使能。

9.2: SSP I²C 操作

用 I²C 方式的 SSP 模块除支持通常请求之外，它还可实现所有从功能，和在开始和停止位上用硬件提供中断以决定一个自由总线（多主机功能）。MSSP 模块可实现标准模式下的 7 位和 10 位寻址。

参考应用说明 AN578，“SSP 模块在 I²C 多主机环境的使用。”

在 SCL 和 SDA 引脚上当他们作为输入时有一个“低频干扰”过滤器。这个过滤器在 100KHZ 和 400KHZ 两种模式里运行。在 100KHZ 时，当这些引脚是输出时，有一个不依靠单片机频率的引脚的步进速度控制。

两个引脚用做数据传送，它们是 SCL 引脚，它是时钟（SCL）引脚。SDA 引脚，它是数据引脚。当 I²C 模式使用时，SDA 和 SCL 引脚自动设置。SSP 模块功能通过对 SSP 使能位 SSPEN（SSPCON <5>）置 1 来使能。

对于 I²C 操作的 SSP 模块有 5 个寄存器：

- SSP 控制寄存器 (SSPCON)
- SSP 控制寄存器 2 (SSPCON2)
- SSP 状态寄存器 (SSPSTAT)
- 串行接收/发送缓冲器 (SSPBUF)
- SSP 移位寄存器 (SSPSR)，不可直接访问的
- SSP 地址寄存器 (SSPAD)

SSPCON 寄存器允许 I²C 操作的控制，四个方式选择位 (SSPCON <3:0>) 允许下列 I²C 方式之一被选择：

- I²C 从模式 (7 位寻址)
- I²C 从模式 (10 位寻址)
- I²C 从模式 (7 位寻址)，时钟=OSC/4 (SSPAD+1)

在选择任何 I²C 模式之前，SCL 和 SDA 引脚必须可通过对合适的 TRIS 位置 1 编程为输入。通过对 SSPEN 位置 1 来选择 I²C 模式。使能 SCL 和 SDA 引脚以用作在 I²C 模式的时钟和数据线。

CKE 位在主模式或从模式里，置 (SSPSTAT <6:7>) SDA 和 SCL 引脚的电平置 1。当 CKE=1，电平将符合 SMBUS 要求。当 CKE=0 时，电平将符合 I2C 要求。

SSPSTAT 寄存器给出数据传送的状态。这种情况包括 START 和 STOP 位的检测，确定所接收的字节是数据还是地址，下一个字节是否完成了 10 位地址传送；以及数据传送是读还是写。SSPSTAT 寄存器是只读的。

SSPBUF 是一个寄存器，传送数据写入或从 SSPBUF 读出。SSPSR 寄存器将数据从单片机中移出单片机。在接收操作时，SSPBUF 和 SSPSR 创建一个双重缓冲接收器。这种允许下一个字节在读所接收的数据的最后一个字节之前开始。当字节接收完成后，它被传送到 SSPBUF 寄存器和标志位 SSPIF 是置 1 的。如果另一个字节在 SSPBUF 寄存器被读之前接收完成，这时会产生一个接收溢出，位 SSPOV (SSPCON <6>) 是置 1 的和在 SSPSR 的字节丢失。

SSPAD 寄存器保存从地址，在 10 位寻址模式下，用户必须先写地址的高字节 (1111 0 A9 A8 0)，在高字节匹配之后，需要装入地址的低字节。

9.2.1 从模式

在从模式中，SCL 和 SDA 引脚必须设定为输入 (TRISC <4:3> 置 1) 当要求从模式时 MSSP 模块将超越带有输出数据的输入状态 (从发送器)。

在接收到一个地址匹配之后，当一个地址匹配或数据传送时，硬件将自动地产生应答 (ACK) 脉冲，然后用在 SSPSR 寄存器接收到的当前值装入 SSPBUF 寄存器。

导致 SSP 模块不产生 ACK 脉冲有一定的条件，它们两者之一或都是：

- a): 在传送接收到之前，缓冲器装满位 BF (SSPSTAT <0>) 置 1。
- b): 在传送接收到之前，溢出位 SSPOV (SSPCON <6>) 置 1。

如果 BF 位是置 1，SSPSR 寄存器值不被装入 SSPBUF，但位 SSPIF (PIR1 <3>) 是置 1 的，表 9-2 表示当接收到一个数据传送时所发生的情况：给出位 BF 和 SSPOV 的状态，阴影部分给出了用户软件没有对溢出条件清零的条件。标志位通过读 SSPBUF 寄存器清零，同时 SSPOV 位通过软件清零。

SCL 时钟输入对于适当的操作必须有一个最小的高或低电平。I²C 指定的高和低电平时间以及 MSSP 模块的要求在时序参数 #100 和 #101 中说明。

9.2.1.1: 寻址

一旦 SSP 模块使能，它等待一个开始条件发生。在开始条件之后，8 位地址数据移入 SSPSR 寄存器。所有接收位在时钟线上升沿采样。SSPSR <7:1> 寄存器的值和 SSPADD 寄存器的值进行比较。地址比较在第八个时钟脉冲 (SCL) 下降边沿进行。如果地址匹配，则 BF 和 SSPOV 位是清零的，产生下列操作：

- a) SSPSR 寄存器值在 SCL 脉冲的第 8 个下降边沿装入到 SSPBUF 寄存器
- b) 缓冲器装满位 BF 在 SCL 脉冲的第 8 个下降边沿置 1
- c) 产生 ACK 脉冲
- d) SSP 中断标志位，SSPIF (PIR1 <3>) (如果能使中断产生) 在第九个 SCL 脉冲下降沿置 1。

在 10 位地址模式中，通过从设备需要接收两个地址字节，(图 11-16) 如果这是一个 10 位地址，首先的地址字节中的 5 个最有效位 (MSBs)。指定 R/W (SSPSTAT <2>) 必须指定一个写入，以便

从设备将接收第二个地址字节。对于一个 10 位地址，第一个字节将等于 ‘1111 0 A9 A8 0’，其中 A9 和 A8 是地址中的两个 MSBs，10 位地址工作顺序如下，从发送器带有 7~9 步。

1. 接收地址的第一个字节（位 SSPIF，BF 和（SSPSTAT <1>）是置 1）。
2. 用地址的第 2 个字节更新 SSPADD 寄存器（对位 UA 清零和释放 SCL 线）。
3. 读 SSPBUF 寄存器（对 BF 位清零）和对标志位 SSPIF 清零。
4. 接收地址的第二个字节（位 SSPIF，BF 和 UA 是置 1 的）。
5. 用地址的第一个字节更新 SSPADD 寄存器，如果匹配将释放 SCL 线，对位 UA 清零。
6. 读 SSPBUF 寄存器（对 BF 清零）和对标志位 SSPIF 清零。
7. 接收重复的开始条件。
8. 接收地址的第一个字节（高位）（位 SSPIF 和 BF 置 1）。
9. 读 SSPBUF 寄存器（BF 位清零）和对 SSPIF 标志位清零。

注：在 10 位模式里在重复开始条件之后，用户只需要和开始的 7 位地址相匹配。用户对于地址的另一半不需要更新 SSPADD。

9.2.1.2: 从接收

当地址字节中的 R/W 位是 “0” 且地址匹配时，SSPSAT 的 R/W 位被清零，接收地址装入 SSPBUF 寄存器。

当地址字节溢出条件产生时，将不产生 ACK 认可信号，溢出条件定义为位 BF（SSPSTAT <0>）置 1 或者 SSPOV 位（SSPCON <6>）是置 1 的。

对于每一个数据发送字节产生一个 MSSP 中断，标志位 SSPIF（PIR1 <3>）必须用软件清零，SSPSTAT 用作决定字节的状态。

注：如果 SSPOV 位是置 1 的和 BF 标志位是清零时，SSPBUF 将被装入。如果执行 SSPBUF 读，但用户在下一个接收发生之前不对 SSPOV 位的状态清零，ACK 不发送和 SSPBUF 被更新。

9.2.1.3: 从发送

当所引入的地址字节的 R/W 位是置 1 和一个地址匹配时，SSPSTAT 寄存器的 R/W 位是置 1 的。所接收的地址装入 SSPBUF 寄存器，ACK 脉冲在第九位时发送，引脚 RC3/SCK/SCL 应通过对位 CKP（SSPCON <4>）置 1 来使能。主机必须监视 SCL 引脚在先前表明的另一个时钟脉冲。从单片机可以通过扩充时钟离开主机。8 位数据在 SCL 时钟下降沿时被移位出去。这样保证 SDA 信号在 SCL 高电平时间内是有效的（图 9-7）。

对于每个数据传送字产生一个 MSSP 中断，标志位 SSPIF 必须用软件清零。SSPSTAT 寄存器用作决定字节的状态，标志位 SSPIF 在第九个时钟脉冲的下降边沿置 1。

作为一个从发射器，从主接收器中的 ACK 脉冲被锁存在第九个 SCL 输入脉冲的上升边沿。如果 SDA 线是高电平（不认可）那么数据传送完成。当 ACK 通过从设备锁存时，从设备逻辑是复位的（复位 SSPSTAT 寄存器）和从设备然后监视另一个开始位的出现，如果 SDA 线是低电平（ACK），发送数据必须装入 SSPBUF 寄存器，同时也装入 SSPSR 寄存器，那么引脚 RC3/SCK/SCL 应通过位 CKP 置 1 使能。

9.2.2: 一般请求地址支持

对于 I2C 总线的寻址过程是这样的，在开始条件之后的第一个字节决定通过主机单片机将被从寻址。例外的是一般请求寻址，它能够寻址所有单片机，当这个地址被使用时，所有单片机在理论上应该用一个认可来响应。

一般请求寻址是通过 I²C 协议的指定目标的 8 个反向地址之一。它由带有 R/W=0 的所有 0 组成。

当一般访问能使位（GCEN）能使时（SSPCON2 <7> 是置 1）一般访问地址被识别。在一个开始探测之后，8 位被移位到 SSPSR 和地址与 SSPADD 相比较。它也和一般访问地址相比较，用硬件固定。

如果一般访问地址匹配，SSPSR 被传送到 SSPBUF、BF 标志置 1（第 8 位）和在第九位（ACK 位）的下降边沿时，SSPBUF 标志置 1。

当中断服务时，通过读 SSPBUF 的内容能够检查中断源，以决定其地址是单片机指定的还是一般访问地址。

在 10 位模式里，SSPADD 对于第二位地址匹配的一半和 UA 位置 1（SSPSTAT <1>）时要求被更新。如果一般请求地址当 GCEN 置 1 时被采样，而这时从设备用 10 位地址模式设置。那么不需要第二个一半地址，UA 位将不被置 1。和从设备在认可之后开始接收数据。

9.2.3: 睡眠操作

当在睡眠方式时，I²C 模块能够接收地址或数据和当一个地址匹配或者全部字节传送出现时，将单片机从睡眠中唤醒（如果 SSP 中断位能使）。

9. 2. 4: 复位影响

一个复位将不能使 MSSP 模块和终止当前传送。

9. 2. 5: 主模式

主模式的操作通过探测到 START 和 STOP 条件时产生的中断来支持。停止和开始位通过复位和当 MSSP 模块不使能时来清零。在当 STOP 位置 1 时，或在总线空闲和 STOP 和 START 位是清零时，I²C 总线的控制启动。

在主模式中 SCL 和 SDA 线是通过 MSSP 硬件进行操作的。

下列事件将引起 SSP 中断标志位，SSPIF 置 1（SSP 中断能使）：

- 起始条件
- 停止条件
- 数据传送字节发送/接收
- 认可发送
- 重复开始

9.2.6: 多主机模式

在多主机模式中，在探测到起始和停止条件时产生中断，它用来判断总线什么时候空闲。停止和起始位通过复位或者当 MSSP 模块不使能时来清零。停止和起始位在 START 和 STOP 条件的基础上转换。当位 P（SSPSTAT <4>）置 1 或总线空闲和 S 和 P 位是清零时，I²C 总线控制可以启动。当总线忙时，停止条件出现时，使能 SSP 中断将产生中断。

在多主机操作中，SDA 线必须被监测看看是否有所要求的输出电平的信号电平。这种检查用硬件完成，结果放在 BCLIF 位里。

竞争失败的状态，它们是：

- 地址传送
- 数据传送
- 一个开始条件
- 一个重复开始条件
- 一个认可条件

9. 2. 7: I²C 主操作支持

通过对 SSPCON 里的适当的 SSPM 位置 1 和清零将 SSPEN 位置 1 来使能主模式，一旦主模式使能，用户有六种选择。

——在 SDA 和 SCL 线上维护一个开始条件。

——维护一个重复开始条件。

——写入 SSPBUF 寄存器初始化数据/地址的发送。

——在 SDA 和 SCL 上产生一个停止条件。

——设置 I²C 端口以接收数据。

——在所接收的数据字节的结尾产生一个认可条件。

注：MSSP 模块，当用 I²C 主模式设置时，不允许事件排队，例如：用户不允许初始化一个开始条件和立即写入 SSPBUF 寄存器在开始条件完成前初始化传送。在这种情况下，SSPBUF 将不写入和 WCOL 位将置 1，表明写入 SSPBUF 并未发生。

9. 2. 7. 1: I²C 主模式操作

主机产生所有串行时钟脉冲和开始和停止条件。传送随着停止条件或随着重复开始条件而结束。因为重复开始条件也是下一个串行传送的开始，I²C 总线将不会释放。

在主发送器模式，在 SCL 输出是串行时钟时，串行数据通过 SDA 输出。第一个发送的字节包括接收单片机（7 位）的从地址和读/写（R/W）位。在这种情况下，R/W 位将是逻辑“0”，串行数据在一个时间发送 8 位。在每个字节发送之后，接收一个认可位，开始和结束条件输出表明串行传送的开始和结束。

在主接收模式，第一个发送的字节包括发送单片机（7 位）的从地址和读/写（R/W）位。在这种情况下，R/W 位将是逻辑“1”，那么第一个发送的字节是一个 7 位从地址紧跟在后面的一个 1 表明接收位。串行数据在一个时间接收 8 位。在每个字节接收之后，发送一个认可位。开始和结束条件输出表明串行传送的开始和结束。

对于 SPI 模式操作的波特率发生器现在用作将 SCL 时钟频率置为 100KHZ、400 KHZ 或 1 KHZ I²C 运行。波特率发生器包括用 SSPADD 寄存器的低 7 位重新装入的值。波特率发生器在写入 SSPBUF 时将重新自动开始计数，一旦所给的操作完成（即最后数据的发送通过 ACK 紧跟着）内部时钟将自动地停止计数和 SCL 引脚将保持在它的最后状态。一个典型的发送顺列如下：

- a) 用户通过在 SSPCON2 里的开始使能位 (SEN) 置 1 来产生开始条件。
- b) SSPIF 置 1，模块在任何其它操作发生时，将等待所要求的开始时间。
- c) 用户用地址装入 SSPBUF 来发送。
- d) 地址移位出 SDA 引脚直到所有 8 位被发送。
- e) MSSP 模块从从单片机用 ACK 位移位，和写它的值到 SSPCON2 寄存器 (SSPCON2 <6>)。
- f) 模块在第九位时钟周期的结尾通过对 SSPIF 置 1 而产生一个中断。
- g) 用户用数据的 8 位装入 SSPBUF。
- h) 数据移出 SDA 引脚直到所有 8 位被发送。
- i) MSSP 模块从从单片机用 ACK 位移位和写它的值到 SSPCON2 寄存器 (SSPCON2 <6>)。
- j) MSSP 在第九个时钟周期结尾通过对 SSPIF 置 1 来产生一个中断。
- k) 用户通过对在 SSPCON2 里的停止能使位 PEN 置 1 来产生一个停止条件。
- l) 一旦停止条件完成就产生中断。

9. 2. 8: 波特率发生器

用 I²C 主模式时，对于 BRG 重新装入值被确定在 SSPADD 寄存器 (图 9-10) 的低 7 位里。当 BRG 用这个值装入时，BRG 计数下降到 0 和停止，直到另一个重新装入发生。BRG 计数在 Q2 和 Q4 时钟每个指令周期 (TCY) 衰减两次。

用 I²C 主模式，BRG 自动重新装入，作为例子如果时钟判优发生，当 SCL 引脚被采样为高电平时，BRG 将重新装入。

9. 2. 9: I²C 主模式开始条件时序

为初始化开始条件，用户将开始条件使能位 SEN (SSPCON2 <0>) 置 1，如果 SDA 和 SCL 引脚被采样为高电平，波特率发生器用 SSPADD <6:0> 的内容重新装入和开始它的计数 (当波特率发生器溢出时)。如果 SCL 和 SDA 都被采样为高电平，则 SDA 引脚被驱动为低电平。当 SCL 是高电平时，被驱动为低电平的 SDA 的活动是一个开始条件，和导致 S 位 (SSPSTAT <3>) 置 1。紧跟这个波特率发生器用 SSPADD <6:0> 的内容重新装入和恢复它的计数。当波特率发生器溢出时，SEN 位 (SSPCON <2:0>) 将自动清零。波特率发生器暂停释放 SDA 线保持低电平和开始条件完成。

注：如果在开始条件期间，SDA 和 SCL 引脚已经采样到低电平，或者如果在开始条件期间，在 SDA 线被驱动为低电平之前 SCL 线采样到低电平，这时发生总冲突。总线冲突中断标志 (BCLIF) 置 1，开始条件中断和 I²C 模块复位到 IDLE 状态。

9. 2. 9. 1: WCOL 状态标志

如果当开始顺序正在进行时，用户写入 SSPBUF，那么 WCOL 置 1 和缓冲器的内容不变（写入未发生）。

注：因为事件的排队不允许，写到 SSPCON2 的低 5 位是不能的，直到开始条件完成。

9. 2. 10: I²C 主模式重复开始条件时序

当 RSEN 位 (SSPCON2 <1>) 置为高电平和 I²C 模块在空闲状态时，将出现重复开始条件。当 RSEN 位置 1 时，SCL 引脚肯定是低电平。当 SCL 引脚采样为低电平时，波特率发生器用 SSPADD <6:0> 的内容装入，并且开始计数。对于一个波特率发生器计数 (T_{BRG}) SDA 引脚被释放。当波特率发生器溢出时，如果 SDA 采样为高电平，SCL 引脚将不被肯定。当 SCL 采样为高电平时，波特率发生器用 SSPADD <6:0> 的内容重新装入和开始计数。SDA 和 SCL 对于 TBRG 必须采样高电平。对于当 SCL 高电平时的一个 TBRG 通过 SDA 引脚 (SDA 是低电平) 确定紧跟着这个活动。在这之后，在 SSPCON2 寄存器里的 RSEN 位将被自动清零，和波特率发生器不被重新装入。离开 SDA 引脚保持低电平。一旦在 SDA 和 SCL 引脚上探测到开始条件，S 位 (SSPSTAT <3>) 将置 1，SSPIF 位不被置 1 直到波特率发生器已溢出。

注 1. 如果 RSEN 置 1 当任何其它事件在进行，它将不产生影响。

2. 在重复开始条件发生期间，如果出现下面事件，总线冲突。

- 当 SCL 从低变高时，SDA 采样低电平。

- 在 SDA 断定为低电平之前，SCL 变为低电平。这样可以表明另外一个主机正试图发送一个数据 “1”。

在 SSPIF 位变成置 1 之后，用户可以在 7 位模式中用 7 位地址写入 SSPBUF，或者用 10 位模式缺有第一个地址。在第一个 8 位发送之后和接收一个认可，那么用户可以发送另外一个八位地址（10 位模式）或者 8 位数据（7 位模式）。

9. 2. 10. 1: WCOL 状态标志

如果当一个重复开始顺序正在进行时，用户写入 SSPBUF，那么 WCOL 置 1 和缓冲器的内容不变（写不发生）。

注：因为不允许事件排队，写入 SSPCON2 的低 5 位不能够直到重复开始条件完成。

9. 2. 11: I²C 主模式发送

数据字节的发送，一个 7 位的地址，或者是通过简单地写一个值到 SSPBUF 寄存器来实现一个 10 位地址的一半。这种活动将缓冲器装满标志 BF 置 1 和允许波特率发生器开始计数和开始下一个发送。地址/数据的每一位在 SCL 下降边沿被断定之后将被移到 SDA 引脚（见数据保持时间说明书）。对于一个波特率发生器转回计数器（TBRG），SCL 保持低电平。在 SCL 被释放为高电平时，它对 TBRG 保持那种方法。在 SDA 引脚上的数据对于在下一个 SCL 下降边沿之后的那个时期和一些保持时间来说必须保持稳定。在第八位被移出之后（第八个时钟的下降边沿）BF 标志被清零和主机释放 SDA 以允许被寻址的从单片机在第九位时间用一个 ACK 响应。如果发生一个地址匹配，或者如果数据被接收。ACK 的状态在第九个时钟下降边沿被读到 ACKDT。如果主机接收到一个认可，认可状态位（ACKSTAT）被清零。如果未接收到一个认可，这位被置 1。在第九个时钟 SSPIF 置 1 之后，和主机时钟（波特率发生器）被暂停，直到下一个数据字节被装入到 SSPBUF，剩余的 SCL 和 SDA 未变（图 9-14）。

在写入 SSPBUF 之后，地址的每一位将移到 SCL 的下降边沿，直到所有七个地址位和 R/W 位完成为止。在第八个时钟的下降边沿，主机将不肯定 SDA 引脚以允许从机用一个认可来响应。在第九个时钟的下降边沿，主机将采样 SDA 引脚看看是否通过从机识别地址。ACK 的状态位被装进 ACKSTAT 状态位（SSPCON2 <6>）。在地址发送的第九个时钟的下降边沿之后，SSPIF 置 1，BF 标志清零和波特率发生器关闭，直到另一个写入 SSPBUF 发生，保持 SCL 低电平和允许 SDA 浮动。

9. 2. 11. 1: BF 状态标志

在发送模式时，当 CPU 写到 SSPBUF，BF 位（SSPSTAT <0>）置 1。当所有 8 位被移出时，BF 位被清零。

9. 2. 11. 2: WCOL 状态标志

当发送已在进行（即：SSPBUF 仍在移出一个数据字节），如果用户写入 SSPBUF，那么 WCOL 置 1 和缓冲器的内容不变（写入未发生）。

WCOL 必须用软件清零。

9. 2. 11. 3: ACKSTAT 状态标志

在发送模式里，当从机已发送一个认可时（ACK=0），ACKSTAT 位（SSPCON2 <6>）是清零的，和当从机没有认可（ACK=1）时，ACKSTAT 置 1，当从机已识别到他的地址时（包括一般访问），或者当从机已接收到它的数据时，从机发出一个认可。

9. 2. 12: I²C 主模式接收

通过对接收位 RCEN（SSPCON2 <3>）置 1，使能主机模式接收。

注：MSSP 模块在 RCEN 位置 1 或者 RCEN 位不承认之前必须是在一个空闲状态。

波特率发生器开始计数和在每个转回时，SCL 引脚的状态的改变（由高变低或由低变高）和数据被移进 SSPSR。在第八个时钟的下降边沿，接收使能标志自动清零。SSPSR 的内容被装入 SSPBUF，BF 标志被置 1，SSPIF 被置 1，和波特率发生器暂停计数。保持 SCL 低电平，SSP 现在在空闲状态，等待下一个命令。当通过 CPU 读缓冲器时，BF 标志自动清零。用户在接收的结尾通过对认可顺序使能位，ACKEN（SSPCON2 <4>）置 1 时发生一个认可位。

9. 2. 12. 1: BF 状态标志

在接收操作时，当一个地址或数据字节从 SSPSR 装入 SSPBUF 时 BF 被置 1，当 SSPBUF 被读时，BF 位被清零。

9. 2. 12. 2: SSPOV 状态标志

在接收操作中，当 8 位接收到 SSPSR，和 BF 标志已从先前的接收中置 1 时，SSPOV 被置 1。

9. 2. 12. 3: WCOL 状态标志

当一个接收已在进行时（即 SSPSR 仍在用数据字移位）如果用户写入 SSPBUF，那么 WCOL 将置 1 和缓冲器的内容不变（写未发生）。

9. 2. 13: 认可顺序时序

一个认可顺序通过对认可顺序使能位 ACKEN (SSPCON2 <4>) 置 1 能使。SCL 引脚被拉向低电平和认可数据位的内容保持在 SDA 引脚上。如果用户希望产生一个认可, 那么 ACKDT 位应被清零。如果不希望, 用户应该在开始一个认可顺序之前将 ACKDT 位置 1。波特率发生器对于一个回转周期 TBRG 计数, 和 SCL 引脚是不肯定的 (拉向高电平)。当 SCL 引脚被采样为高电平时 (时钟仲裁), 波特率发生器对于 TBRG 计数。SCL 引脚被拉向低电平, 在这之后, ACKEN 位自动清零, 波特率发生器关闭, 和 MSSP 模块变为空闲模式。(图 9-16)

9. 2. 13. 1: WCOL 状态标志

当一个认可顺序在进行时, 如果用户写入 SSPBUF, 那么 WCOL 被置 1 和缓冲器的内容不变 (写未发生)。

9. 2. 14: 停止条件时序

在 SDA 引脚上, 通过对停止使能位 PEN (SSPCON <2>) 置 1 来结束一个接收/发送以确定一个停止位。在接收/发送结束时, SCL 线在第九个时钟的下降边沿之后保持低电平。当 PEN 位置 1, 主机将确定 SDA 线低电平。当 SDA 线被采样为低电平时, 波特率发生器被重新装入和计数下降到 0。当波特率发生器溢出时, SCL 引脚将被带到高电平和一个较迟的 TBRG (波特率发生器重新计数), SDA 引脚将不被肯定。当 SDA 引脚被重新采样为高电平时, 这时 SCL 是高电平。P 位 (SSPSTAT <4>) 是置 1 的, 一个较迟的 TBRG、PEN 位被清零和 SSPIF 位被置 1。

无论什么时候固件决定采取总线控制, 它将首先通过检查在 SSPSTAT 寄存器里的 S 和 P 位以决定总线是否忙。如果总线是忙的, 那么当一个停止位被检测到时 (即总线空线), CPU 会被中断 (被通知)。

9. 2. 14. 1: WCOL 状态标志

改当一个 STOP 顺序在进行时, 如果用户写入 SSPBUF, 那么 WCOL 被置 1 和缓冲器的内容不变 (写未发生)。

9. 2. 15: 时钟仲裁

当主机在任何接收和发送期间或按重复开始/停止条件, 不能肯定 SCL 引脚 (允许 SCL 浮动高电平) 时, 将出现仲裁。当 SCL 引脚被允许浮动高电平时, 波特率发生器从计数中暂停, 直到 SCL 引脚被实际采样高电平时为止。当 SCL 引脚采样为高电平时, 波特率发生器重新用 SSPADD <6:0> 的内容装入和开始计数。这样保证在通过外部器件来保持时钟为低电平的事件中, SCL 一直是高电平时间至少是一个 BRG 重新计数。

9. 2. 16: 睡眠操作

当在睡眠方式时, I²C 模块能够接收地址和数据, 当地址匹配时或全部字节传送发生时, 单片机从睡眠中唤醒 (如果 SSP 中断能使)。

9. 2. 17: 复位影响

一个复位不能使 MSSP 模块和终止当前传送。

9. 2. 18: 多机通讯、总线冲突和总线仲裁

通过总线竞争来取得多主机模式支持。当主机输出地址/数据到 SDA 引脚时, 发生竞争。当主机在 SDA 上通过使 SDA 浮动高电平输出为 1 时和另一个主机确定为 0, 当 SCL 引脚浮动高电平时, 数据应该稳定。如果期望在 SDA 上的数据是 1 和所采样到的数据在 SDA 引脚等于 “0”, 那么已经发生总线冲突。主机将总线冲突中断标志 BCLIF 置 1, 和复位 I²C 端口到它的空闲状态 (图 9-19)。

如果当总线冲突发生时发送在进行, 发送将被终止, BF 标志被清零。SDA 和 SCL 线不能确定和 SSPBUF 会被写入。当用户使用总线冲突中断服务程序时和如果 I²C 总线是空闲时, 用户通过确定一个开始条件能够恢复通讯。

如果一个开始、重复开始, 停止或认可条件在进行, 当总线冲突发生时, 条件被中止。SDA 和 SCL 不能确定和在 SSPCON2 寄存器里的各自的控制位被清零。当用户使用总线冲突中断服务程序时, 和如果 I²C 总线是自由的, 用户通过确定一个开始条件恢复通讯。

主机将继续监视 SDA 和 SCL 引脚, 和如果出现停止条件, SSPIF 位将被置 1。

一个写入 SSPBUF 将在第一个数据位上开始数据发送。而不管当总线冲突出现时, 发送停止。

在多机模式中, 在开始和停止条件的探测上中断产生以允许决定什么时候总线是空闲的。当在 SSPSATA 寄存器里的 P 位置 1 时或总线空闲和 S、P 位被清零时, I²C 总线的控制能够被执行。

9. 2. 18. 1: 在开始条件期间总线冲突

在开始条件期间, 如果出现下列情况将发生总线冲突。

A) 在开始条件的开始,SDA 或 SCL 采样为低电平。

B) 在 SDA 确定为低电平之前, SCL 采样为低电平。

在开始条件期间, SDA 和 SCL 引脚都被监视, 如果 SDA 引脚已经是低电平或 SCL 已经是低电平, 那么开始条件中止和 BCLIF 标志置 1 和 SSP 模块复位到空闲状态。(图 9-20)

开始条件从不确定的 SDA 和 SCL 引脚的开始。当 SDA 引脚被采样为高电平时, 波特率发生器从 SSPADD <6:0> 里被装入和计数下降到 0。如果 SCL 引脚采样为低电平, 而 SDA 为高电平, 则出现总线冲突。因为它被恢复另外一个主机在开始条件期间正试图驱动一个数据“1”。

如果 SDA 引脚在这个计数期间被采样为低电平, BRG 被复位和 SDA 线被早早确定。如果在 SDA 引脚上采样到一个“1”, SDA 引脚在 BRG 计数结尾被确定为低电平, 波特率发生器被重新装入和计数下降到 0。和在这期间, 如果 SCL 引脚被采样为“0”, 总线冲突不会发生。在 BRG 计数结束时, SCL 引脚被确定为低电平。

注: 总线冲突的原因不是在一个开始条件期间的一个因素, 而是在精确的同一时间无两个总线主机能确定一个开始条件。因此一个主机在另一个主机之前将一直确定 SDA。这个条件不会引起总线冲突, 因为两个主机必须被允许竞争在开始条件之后的第一个地址。如果地址是相同的, 竞争必须允许继续一部分数据, 重新开始或停止条件。

9. 2. 18. 2: 在重复开始条件期间总线冲突

在重复开始期间, 如果出现下列情况, 发生总线冲突:

A) 当 SCL 从低电平变到高电平时, 在 SDA 上采样一个低电平。

B) 在 SDA 被确定为低电平之前, SCL 变为低电平, 表示另一个主机正试图发送一个数据“1”。当用户不确定 SDA 时和引脚被允许浮动高电平, BRG 用 SSPADD <6:0> 装入和计数下降到 0, SCL 引脚被确定。和当 SCL 采样为高电平时, SDA 引脚被采样。如果 SDA 是低电平, 一个总线冲突已经发生(即另一个主机正试图发送一个数据“0”)。如果 SDA 采样高电平, 那么 BRG 被重新装入和开始计数。如果 SDA 在 BRG 溢出之前由高变低, 那么无总线冲突出现, 因为两个主机在精确的同一时间能确定 SDA。

如果 SCL 在 BRG 溢出之前由高变低和 SDA 已不能确定, 那么出现总线冲突, 在这种情况下另一个主机在重复开始条件期间正试图发送一个数据 1。

如果在 BRG 溢出结束时, SCL 和 SDA 都是高电平, 那么 SDA 引脚被驱动为低电平, BRG 重新装入和开始计数。在计数的结束时, 不管 SCL 引脚的状态如何, SCL 引脚被驱动为低电平和重复开始条件被完成。图 9-23 略

9. 2. 18. 3: 在停止条件期间总线冲突

如果出现下列情况, 在停止条件期间将出现总线冲突:

a) 在 SDA 引脚已经被否定和允许浮动高电平之后, SDA 在 BRG 溢出之后采样为低电平。

b) 在 SCL 引脚被否定之后, SCL 在 SDA 变成高电平之前采样为低电平。

停止条件从 SDA 确定为低电平开始, 当 SDA 采样为低电平时, SCL 引脚允许浮动, 当引脚采样为高电平时(时钟竞争), 波特率发生器用 SSPADD <6:0> 装入和计数下降到 0。如果 SCL 引脚在 SDA 允许浮动高电平之前被采样到, 发生总线冲突, 这是另外一个主机试图驱动一个数据“0”。(图 9-25)

9. 3: I²C 总线连接注意事项

对于标准模式 I²C 总线单片机在图 9-39 中电阻 R_P 和 R_S 的值依靠下列参数:

- 电源电压
- 总线容量
- 所连接的单片机数量 (输入电流加漏电流)

由于对于所指定的输出等级在 VOLmax=0.4V 时, 所指定的最小 3mA 的拉出电流, 所以电源电压限制电阻 R_P 最小值。例如: Vdd=5V±10% 的电源电压和在 3mA 时 VOLmax=0.4V。R_{Pmin}=(5.5-0.4)/0.003=1.7KΩ。Vdd 作为 R_P 的一个功能见图 9-39 说明所要求的 0.1Vdd 的噪音边界对于低电平限制 R_S 系列电阻的最小值是可选择和被用作提高 ESD 灵敏度。

总线容量是线路连接和引脚的总容量。由于所指定的上升时间(图 9-27), 这个容量限制 R_P 的最大值。

SMP 位是转换速率控制使能位, 这位是在 SSPSTAT 寄存器里, 和当在 I²C 模式时(主或从)控制 I/O 引脚的转换速率。

10: 通用同步异步接收/发送器

通用同步异步接收发送器模块是两个串行 I/O 模块之一，(USART 也被称作串行通讯接口或 SCI)，USART 可被设置为全双工异步系统，它能和外部器件通讯，如 CRT 终端和个人计算机，或者它可设置为一个半双工同步系统，它能够和外部器件通讯，如 A/D 或 D/A 集成电路，串行 EEPROMS 等等。

USART 可通过下列方式设定：

- 异步(全双工)
- 同步—主机 (半双工)
- 同步—从机 (半双工)

为了将引脚 RC6/TX/CK 和 RC7/RX/DT 设定为通用同步、异步接收发送器，位 SPEN (RCSTA <7>) 和 TRISC <7:6> 必须置 1。

USART 模块也有一个使用 9 位地址探测的多主机通讯能力。

10. 1: USART 波特率发生器 (BRG)

BRG 支持包括 USART 的同步和异步方式，它是一个专用的 8 位波特率发生器，SPBRG 寄存器控制一个自由运行的 8 位定时器的周期、用异步方式，位 BRGH (TXSTA <2>) 也控制波特率。用同步方式，位 BRGH 被忽略，表 10-1 给出了于不同的 USART 模式波特率的计算，它仅用在主机方式 (内部时钟)

给出所期望的波特率和 FOSC，对于 SPBRG 寄存器按最近的整数值可用表 10-1 里的公式计算，在这一点上，在波特率里的误差能被确定。

它甚至会对于较低波特率时钟使用高波特率 (BRGH=1) 有帮助，这是因为 $FIM / (16(X+1))$ 方程式在一些情况能够减少波特率错误。

写入一个新值到 SPBRG 寄存器会导致 BRG 定时器复位 (或清零) 这样保证 BRG 在输出新的波特率之前不等待一个定时器溢出。

10. 1. 1: 采样

在 RCT/RX/DT 引脚上的数据，在每一位时间中心附近被采样三次，如果一个高或低电平出现在 RX 引脚上，这种采样通过一个主要的探测电路来决定。

10. 2: USART 异步方式

用这种方式，USART 使用标准的不归零 (NRZ) 格式 (一个开始位，八个或九个数据位，和一个停止位)。最常用的数据格式是 8 位，一个所指定的片内 8 位波特率发生器用作从振荡器里导出标准的波特率频率。USART 首次发送和接收 LSB。USART 的发送器和接收器在功能上是相互独立的，但它们使用相同的数据格式和波特率，波特率发生器依靠 BRGH 位 (TXSTA <2>) 产生一个 16 倍或 64 倍的位移位速率时钟，通过硬件不支持奇偶性，但用软件能够执行 (作为第九位数据存储)。异步方式在 SLEEP 期间停止。

异步方式通过对位 SYNC (TXSTA <4>) 清零来选择。

USART 异步模块由下列重要元件组成：

- 波特率发生器
- 采样电路
- 异步发送器
- 异步接收器

10. 2. 1: USART 异步发送器

USART 发送器方框图见图 10-1，发送器的关键部件是发器移位寄存器 (TSR)，移位寄存器从读/写发送器缓冲器 TXREG 中获得数据，TXREG 寄存器用软件装入数据，TSR 寄存器不装入新的数据直到先前装入数据的停止位已被发送。停止位一发送，TSR 就从 TXREG 寄存器 (如果有效) 里装入新的数据。一旦 TXREG 寄存器将数据传送到 TSR 寄存器 (发生在一个 TCY)，TXREG 寄存器变空，标志位 TXIF (PIR1 <4>) 被置 1。通过对标志位 TXIE (PIE1 <4>) 的置 1 和清零来决定中断使能或不能。不管使能位 TXIE 的状态和用软件不能被清零，标志位 TXIF 将被置 1，只有当新的数据装入 TXREG 寄存器时它将被复位，当标志位 TXIF 表明 TXREG 寄存器的状态时，另一位 TRMT (TXSTA <1>) 表明 TSR 寄存器的状态，状态位 TRMT 是一个只读位，当 TSR 寄存器变空时，这位被置 1。没有中断逻辑连向这位，因此用户不得不探询这位来决定 TSR 寄存器是否变空。

注：1、TSR 寄存器没有被映象在数据寄存器里，因此对用户没有用。

2、当使能位 TXEN 置 1 时，标志位 TXIF 被置 1，TXIF 通过装入 TXREG 而被清零。

通过将使能位 TXEN (TXSTA <5>) 置 1 使能发送功能, 实际上发送直到 TXREG 寄存器装入数据和波特器发生器 (BRG) 已产生一个移位时钟时才会发生 (图 10-2)。发送也可以通过首次装入 TXREG 寄存器然后将使能位 TXEN 置 1 来启动, 通常, 当发送被启动时, TSR 寄存器变空。在这点上, 发送 TXREG 寄存器时将导致立即传送到 TSR, 结果 TXREG 始终是一个空的, 这样一个背靠背的传送是可能的 (图 10-3), 在发送期间对使能位 TXEN 清零将导致发送中断和将发送器复位, 结果, RC6/TX/CK 引脚将回转至高阻抗状态。

为了选择 9 位发送, 发送位 TX9 (TXSTA <6>) 应该被置 1 和第九位应该写入 TX9D (TXSTA <0>)。第九位必须在写 8 位数据到 TXREG 寄存器之前写入。这是因为一个数据写入 TXREG 寄存器将产生一个数据立即传送到 TSR 寄存器 (如果 TSR 是空的), 在这种情况下, 一个不正确的第九位数据位或许被装入到 TSR 寄存器。

用下面的步骤建立一个异步发送:

1. 用适当的波特率初始化 SPBRG, 如果需要高的波特率, 将 BRGH 位置 1。
2. 通过对 SYNC 位清零和 SPEN 位置 1 使异步串行端口能使。
3. 如果需要中断, 那么将发送位 TX9 置 1。
4. 如果需要 9 位发送, 那么将发送位 TX9 置 1。
5. 通过将 TXEN 位置 1 使能发送, 它也将位 TXIF 置 1。
6. 如果选择 9-位发送, 第九位应该用位 TX9D 装入。
7. 向 TXREG 寄存器装入数据 (开始转送)。

10. 2. 2: USART 异步接收器

接收器结构方框图见图 10-4, 数据在 RCI/RX/DT 引脚上接收和驱动数据恢复程序块。数据恢复程序块实际上是一个高速移位器, 它以 16 倍波特率运行, 反之主要的接收串行移位器以位速率或 FINT 运行。

一旦异步方式被选择, 通过将位 CREN (RCSTA <4>) 置 1 使能接收。

接收器的心脏是接收 (串行) 移位寄存器 (RSR)。在采样 STOP 位之后, 在 RSR 里所接收的数据传送到 RCREG 寄存器 (如果它是空的)。如果传送完成, 标志位 RCIF (PIR1 <5>) 被置 1。通过对使能位 RCIE (PIE1 <5>) 置 1/清零, 使实际中断使能和不使能。标志位 RCIF 是只读位, 通过硬件它被清零。RCREG 是一个双重缓冲寄存器, 即它是两个深 FIFO。对于被接收两个数据字节和传送 RCREG FIFO 和第三个字节开始移位到 RSR 寄存器是可能的。在第三个字节停止位的探测上, 如果 RCREG 寄存器仍就是满的, 那么超时运行错误位 OERR (RCSTA <1>) 将被置 1。在 RSR 里的字将被丢失, RCREG 寄存器可以读两次以检索在 FIFO 里的两个字节。超时运行位 OERR 得用软件清零, 这个通过对接收逻辑复位来进行 (CREN 被清零然后置 1)。如果位 OERR 是置 1 的, 从 RSR 寄存器传送到 RCREG 寄存器是禁止的。因此如果位 OERR 是置 1 的, 需要清零错误位。如果一个停止位被探测为清零, 那么组帧错误位 FERR (RCSTA <2>) 被置 1。位 FERR 和第 9 个接收位和接收数据同样是缓冲的。读 RCREG, 将用新值装入 RX9D 和 FERR 位, 因此为了不丢失过去的 FERR 和 RX9D 的信息, 对于用户读 RCSTA 寄存器之前去读 RCSTA 寄存器是有必要的。

用下列步骤建立一个异步接收:

1. 用适当的波特率初始化 SPBRG 寄存器, 如果需要高速波特率将 BRGH 置 1。
2. 通过对 SYNC 位清 0 和对 SPEN 位置 1 使异步串行端口能使。
3. 如果需要中断, 那么将能使位 RCIE 置 1。
4. 如果需要 9 位接收, 那么将 RX9 位置 1。
5. 通过对 CREN 位置 1 使能接收。
6. 当接收完成时, 标志位将被置 1 和如果使能位 RCIE 置 1, 将产生一个中断。
7. 读 RCTA 寄存器以获得第 9 位和决定在接收期间是否发生任何错误。
8. 通过读 RCREG 寄存器来读 8 位接收数据。
9. 如果出现错误, 通过对能使位 CREN 清零来清除错误。

10. 2. 3: 用地址探测建立 9 位模式

用下列步骤建立一个用地址探测的异步接收:

1. 用适当的波特率初始化 SPBRG 寄存器, 如果需要高速波特率将 BRGH 置 1。
2. 通过对 SYNC 位清 0 和对 SPEN 位置 1 使异步串行端口能使。
3. 如果需要中断, 那么将能使位 RCIE 置 1。
4. 如果需要 9 位接收, 那么将 RX9 位置 1。

5. 对 ADDEN 位置 1 使能地址探测
6. 通过对 CREN 位置 1 使能接收。
7. 当接收完成时，标志位将被置 1 和如果使能位 RCIE 置 1，将产生一个中断。
8. 读 RCSTA 寄存器以获得第 9 位和决定在接收期间是否发生任何错误。
9. 通过读 RCREG 寄存器来读 8 位接收数据。
10. 如果出现错误，通过对能使位 CREN 清零来清除错误。
11. 如果单片机已被寻址，对 ADDEN 位清零以允许数据字和地址字读进接收缓冲器，和中断 CPU。

10. 3: USART 同步主模式

用同步主模式时，数据用半双工方法发送，即发送和接收不在同一时间，当发送数据时，接收是禁止的和 Vice Versa。通过对 SYNC (TXSTA <4>) 位置 1 进入同步方式，另外，为了设定 RC6/TX/CK 和 RC7/RX/DT I/O 引脚为各自的 CK(时钟)和 DT/数据)线，使能位 SPEN(RCSTA <7>)是置 1，主模式表明在 CK 线上处理机发送主时钟，通过对 CSRC(TXSTA <7>)位置 1 进入主模式。

10. 3. 1: USART 同步主发送

USART 发送器方框图见图 10-6，发送器的核心是发送移位寄存器 (TSR)，移位寄存器从读/写发送缓冲寄存器 TXREG 里获得数据，TXREG 寄存器用软件装入数据，TSR 寄存器直到从先前装入的最后一位已经传送之前是不会装入，当最后一位一发送，TSR 就从 TXREG (如果可用)用新的数据装入，一旦 TXREG 寄存器传送数据到 TSR 寄存器(在一个 TCYCLE 里出现)TXREG 变空和中断位 TXIF (PIR1 <4>)置 1，通过对使能位 TXIE (PIE <4>)的置 1 和清零来使中断使能和不使能。

不管使能位 TXIE 的状态和用软件不会被清零，标志位 TXIF 将被置 1。只有当新的数据装入到 TXREG 寄存器时它才复位。当标志位 TXIF 表明 TXREG 寄存器的状态时，另一位 TRMT (TXSTA <1>)表示 TSR 寄存器的状态。TRMT 是只读位，当 TSR 变空时，它被置 1。没有中断逻辑连向这位，因此用户得探询这位以决定 TSR 寄存器是否变空，TSR 不被映射到数据存储里，因此对于用户它是不可用的。

通过对能使位 TXEN (TXSTA <5>)置 1 使能发送，实际上发送将不会发生直到 TXREG 寄存器已被装入数据。第一个数据位在 CK 线上的下一个可用的时钟上升沿时将被移出。数据输出在同步时钟下降稳定(图 10-9)。发送也可通过首次装入 TXREG 寄存器然后将位 TXEN 置 1 来启动。(图 10-10)，当低波特率被选择时，这是一个优点，因为当位 TXEN、CREN 和 SREN 清零时，BRG 被保持在复位状态。将使能位 TXEN 置 1 将启动 BRG，立即建立一个移位时钟。一般，当发送是首次启动时，TSR 寄存器是空的，因此一个传送到 TXREG 寄存器将导致一个立即传送到 TSR，产生一个空的 TXREG。背靠背传送是可能的。

在发送期间，对使能位 TXEN 清零，将导致发送中断和发送器复位。DT 和 CK 引脚将转回到高阻状态，如果 CREN 位和 SREN 位在一个发送期间是置 1 的，发送被中断和 DT 引脚转回到高阻状态(一个接收)。如果 CSRC 位是置 1 的(内部时钟)CK 引脚将仍保持输出。然而，发射器逻辑上是不复位的，尽管它与此引脚不相连。为了发送，用户得将 TXEN 位清零。如果 SREN 位是置 1 (中断一个正在进行的一单字节发送和接收)那么在单字节接收后，位 SREN 将被清零和串行端口将转回到发送。因为位 TXEN 仍就是置 1 的，DT 线将立即从高阻接收方式中打开发送和开始驱动，为避免这点，TXEN 位应该清零。

为了选择 9 位发送，发送位 TX9 (TXSTA <6>)应该被置 1 和第九位应该写入 TX9D (TXSTA <0>)。第九位必须在写 8 位数据到 TXREG 寄存器之前写入。这是因为一个数据写入 TXREG 寄存器将产生一个数据立即传送到 TSR 寄存器(如果 TSR 是空的)。如果 TSR 是空的和 TXREG 在写新的 TX9D 之前写入位 TX9D 的目前值被装入

用下面的步骤建立一个同步主发送：

1. 用适当的波特率初始化 SPBRG 寄存器。
2. 通过对位 SYNC, SPEN 和 CSRC 置 1 使同步主串行端口能使。
3. 如果需要中断，那么将使能位 TXIE 置 1。
4. 如果需要 9 位发送，那么将位 TX9 置 1。
5. 通过将位 TXEN 置 1 使能发送，
6. 如果选择 9-位发送，第九位应该用位 TX9D 装入。
7. 向 TXREG 寄存器装入数据开始转送。

10. 3. 2: USART 同步主接收

一旦同方式被选择, 通过将使能位 SREN (RCSTA <5>) 和使能位 CREN (RCSTA <4>) 置 1 使能接收。数据在 RC7/RX/DT 引脚上, 在时钟的下降边沿被采样。如果使能位 SREN 置 1, 只接收一个单字节。如果使能位 CREN 置 1, 接收连续进行直到 CREN 清零。如果两位都置 1, 那么 CREN 优先。在计时最后一位时, 在接受移位寄存器 (RSR) 的数据传送到 RCREG 寄存器 (如果它是空的)。如果传送完成, 标志位 RCIF (PIR1 <5>) 被置 1。通过对使能位 RCIE (PIE1 <5>) 置 1/清零, 使实际中断能使和不能使。标志位 RCIF 是只读位, 通过硬件它被清零。在这种情况下, 当 RCREG 寄存器已被读和变空时它被复位。RCREG 是一个双重缓冲器寄存器, 即它是两个深 FIFO。对于被接收两个数据字节和传送 RCREG FIFO 和第三个字节开始移到 RSR 寄存器是可能的。在第三个字节最后计时上, 如果 RCREG 寄存器仍就是满的, 那么超时运行错误位 OERR (RCSTA <1>) 将被置 1。在 RSR 里的字将被丢失, RSR 寄存器可以读两次以检索在 FIFO 里的两个字节。超时运行位 OERR 得用软件清零 (通过对 CREN 清零)。如果位 OERR 是置 1 的, 从 RSR 寄存器传送到 RCREG 寄存器是禁止的。因此如果位 OERR 是置 1 的, 需要清零错误位。第 9 个接收位和接收数据同样是缓冲的。读 RCREG, 将用新值装入 RX9D 位, 因此为了不丢失过去的 FERR 和 RX9D 的信息, 对于用户读 RCSTA 寄存器之前去读 RCSTA 寄存器是有必要的。

用下列步骤建立一个异步接收:

1. 用适当的波特率初始化 SPBRG 寄存器
2. 通过对位 SYNC, SPEN 和 CSRC 置 1 使能同步串行端口。
3. 保证位 CREN 和 SREN 清零。
4. 如果需要中断, 那么将能使位 RCIE 置 1。
5. 如果需要 9 位接收, 那么将 RX9 位置 1。
6. 如果需要一个单字接收, 那么将 SREN 位置 1, 如果需要连续接收, 那么将 CREN 置 1。
7. 当接收完成时, 中断标志位将被置 1 和如果使能位 RCIE 置 1, 将产生一个中断。
8. 读 RCSTA 寄存器以获得第 9 位和决定在接收期间是否发生任何错误。
9. 通过读 RCREG 寄存器来读 8 位接收数据。
10. 如果出现错误, 通过对能使位 CREN 清零来清除错误。

10. 4: USART 同步从模式

同步从模式和主模式的区别实际上在于在 RC6/TX/CK 引脚上所供的是外部移位时钟 (而主模式所供的是内部时钟) 这将允许在睡眠方式时单片机发送或接收数据, 通过对 CSRC (TXSTA <7>) 位清零进入从模式。

10. 4. 1: USART 同步从发送

同步主操作和从操作除了在睡眠方式下不同外, 其余是相同的, 如果两个字写入 TXREG, 然后执行一条 Sleep 指令, 将会发生下面情况:

- a) 第一个字立即传送到 TSR 寄存器和发送。
- b) 第二个字仍保留在 TXREG 寄存器里。
- c) 标志位 TXIF 将不被置 1。
- d) 当第一个字已移出 TSR 时, TXREG 寄存器将传送第三个字到 TSR 和标志位 TXIF 将被置 1。
- e) 如果能使位 TXIE 被置 1, 中断将单片机从睡眠中唤醒, 和如果全体中断被使用, 程序将分支到中断向量 (0004h)。

用下列各步建立同步从发送方式:

- 1) 通过对 SYNC 位和 SPEN 位置 1 和 CSRC 位清零来使能同步从串行端口。
- 2) 对位 CREN 和 SREN 位清零。
- 3) 如果需要中断, 那么将使能位 TXIE 置 1
- 4) 如果需要 9 位发送, 将 TX9 位置 1。
- 5) 通过对使能位 TXEN 置 1 能使发送。
- 6) 如果选择 9 位发送, 第九位应该用位 TX9D 装入。
- 7) 通过装入数据到 TXREG 寄存器启动发送。

10. 4. 2: USART 同步从接收

除了在睡眠情况下, 同步从接收操作与主接收是相同的, 同时位 SREN 在从方式时不用考虑。

如果在 Sleep 指令之前位 CREN 置 1 使能接收, 那么在睡眠期间可以接收一个字, 在完全接收这个字上, RSR 寄存器将传送数据到 RCREG 寄存器, 如果使能位 RCLE 位是置 1 的, 所产生的中断将

芯片从睡眠中唤醒，如果全体中断能使，程序将分支到中断向量（0004h）。

用下列步骤建立同步从接收：

1. 通过对 SYNC 和 SPEN 位置 1 和 CSRC 清零来使同步主串行端口能使。
2. 如果需要中断，将 RCIE 位置 1。
3. 如果需要 9 位接收，将 RX9 位置 1。
4. 为使能接收，将使能位 CREN 置 1。
5. 当接收完成和中断发生时，如果使能位 RCIE 是置 1 的，标志位 RCIF 将置 1。
6. 读 RCSTA 寄存器到第九位（如果使能）和决定在接收期间发生任何错误。
7. 通过读 RCREG 寄存器读 8 位接收数据。
8. 如果出现任何错误，通过对 CREN 位清零来清除错误。

11: 模拟，数字转换器模块

模数转换器对于 28 引脚有 5 个输入，对于其它有 8 个输入。

模拟输入充电一个采样和保持电容，采样和保持输出是输入到转换器。转换器 Via 逐次近似法产生一个模拟电平的数字结果。模拟输入信号的 A/D 转换产生一个相应的 10 位数字数。A/D 模块有高或低的参考电压输入，它是可用软件选择的 VDD，VSS，RA2 或 RA3 的组合。

A/D 转换器具有一个独一无二的特征，即在单片机处在睡眠方式时，A/D 转换器仍能够运行，为了在睡眠下运行，A/D 转换器的时钟必须从 A/D 转换器指定的内部 RC 振荡器中导出。

A/D 转换器模块有三个寄存器，这些寄存器是：

A/D 结果高位寄存器（ADRESH）

A/D 结果低位寄存器（ADRESL）

A/D 控制寄存器 0（ADCON0）

A/D 控制寄存器 1（ADCON1）

ADCON0 寄存器控制 A/D 模块的操作（见寄 11-1），ADCON1 寄存器设端口引脚的功能（见寄 11-2），端口引脚能被设置为模拟输入（RA3 也可当作参考电压）或为数字输入/输出。

另外使用 A/D 模块的情况可用 PICMCRO-TM 中等 MCD 系列参考手册（DS33023）和 AN546 的应用说明。

ADRESH: ADRESL 寄存器包括 A/D 转换的 10 位结果。当 A/D 转换完成时，结果放到 A/D 结果寄存器对，GO/DONE 位（ADCON0 <2>）被清零，A/D 中断标志位 ADIF 置 1，A/D 模块方框图见图 11-1。

在 A/D 模块已按要求设置后，所选择的通道在转换之前必须被采集。模拟输入通路必须有所选择的作为输入的相应 TRIS 位。为决定采样时间在这个采集时间已经过去之后 A/D 转换开始。见 11.1 节，

下列步骤用作进行 A/D 转换：

1. 设置 A/D 模块
 - 设定模拟引脚/参考电压/数字 I/O（ADCON1）
 - 选择 A/D 输入通路（ADCON0）
 - 选择 A/D 转换时钟（ADCON0）
 - 接通 A/D 模块（ADCON0）
2. 设置 A/D 中断（如果需要）
 - 对 ADIF 位清零
 - 对 ADIE 位置 1
 - 对 GIE 位置 1
3. 0: 等待所要求的采集时间
4. 0: 开始转换
 - 将 GO/DONE 位置 1（ADCON0）
5. 等待 A/D 转换完成，既可通过：
 - 。探询 GO/DONE 被清零
 - 。等待 A/D 中断。
6. 读 A/D 结果寄存器（ADRES），如果需要对 ADIF 位清零。
7. 对下一个转换，如果需要到第一步或第二步，每位的转换时间定义为 TAD。在下一个采集开

始要求一个最小 2 TAD 等待。

11. 1: A/D 转换器采集要求

为使 A/D 转换器符合它的指定精度，充电保持电容（CHOLD）必须允许完全充电到输入通道电压等级，模拟输入模型见图 11-2。源阻抗（RS）和内部采样开关（RSS）阻抗直接影响所要求的充电电容 CHOLD 的时间，采样开关（RSS）阻抗在单片机电压上的变化，见图 11-2。源阻抗在模拟输入时影响偏移电压（由于引脚漏电流）。所推荐的最大模拟源阻抗是 10K Ω ，在模块输入通道被选择后，在转换可以开始之前采集必须通过。

为计算最小时间，可以使用算式 11-1，这个算式假定有 1/2LSB 错误被使用（1024 级 A/D）对于 A/D 转换器符合它的指定的分辨率 1/2LSB 错误是所允许的最大出错。

为计算最小采集时间，TACQ，见 PicmicroTM 中等 MCU 参考手册（CS33023）。

11. 2 选择 A/D 转换器时钟

A/D 转换器时间每位定义为 TAD，A/D 转换器每 10 位转换要求 12TAD，A/D 转换器时钟源是用软件可选择的，TAD 的四种可能选择是：

- 2TOSC
- 8TOSC
- 32TOSC
- 内部 RC 振荡器

对于正确的 A/D 转换，A/D 转换时钟（TAD）必须被选择以保证 1.6 μ s 的最小 TAD 时间。

11. 3: 设定模拟端口

ADCON1、TRISA 和 TRISE 寄存器控制 A/D 端口引脚的操作。这个需做为模拟输入的端口引脚必须有他们的相应位 TRIS 置 1（输入）。如果 TRIS 位清零（输出），数字输出电平（VOH 或 VOL）将被转换。

A/D 操作是独立于（CHS〈2:0〉）位和 TRIS 位的状态之外的。

注：1. 当读端口寄存器时，所有设定为模拟输入通道引脚将读作零（低电平），设定为数字输入的引脚将转换一个模拟输入。在数字设定输入上的模拟电平将不会影响转换精度。

2. 在任何引脚上定做数字输入的模拟电平，不做为模拟输入，可以导致输入缓冲器消耗不是指定的电流。

11. 4: A/D 转换

在转换期间对 GO/DONE 位清零将中断电流转换。ADRES 寄存器将不会用局部完成 A/D 转换采样来更新。这是因为 ADRES 寄存器继续包含最后完成转换的值（或最后写向 ADRESH: ADRESL 寄存器的值）。在 A/D 转换中断之后，在下一个采集开始之前，在 A/D 转换结束之后，要求一个 2TAD 等待。在 2TAD 等待之后，一个采集在所选的通道上自动地开始。

在图 11-3 里，在 GO 位置 1 之后，第一个时间段有一个最小的 TCY 和一个最大 TAD 的实践。

注：GO/DONE 位不应该置 1 用打开 A/D 的相同的指令。

11. 4. 1A/D 结果寄存器

ADRESL: ADRESH 寄存器对是在 A/D 转换完成时，10 位 A/D 结果放入的存储单元。着个寄存器对是 16 位字宽。A/D 模块在 16 位结果寄存器里给出灵活的左或右调整的结果。A/D 格式选择位（ADFM）控制着个调整。图 11-4 给出了 A/D 结果调整的操作。额外位用“0”装入。当一个 A/D 结果不会覆盖这些存储单元时，这些寄存器可以用作两个普通的 8 位寄存器。

11. 5: 在睡眠期间 A/D 操作

在睡眠期间 A/D 模块可以操作，这种 A/D 时钟源要求对 RC 置 1，（ADCS〈1:0〉=11），当选择 RC 时钟源时，A/D 模块在开始转换之前等待一个指令周期。这种允许执行睡眠指令，消除在转换中产生的所有数字开关噪声。当转换完成时，GO/DONE 位将被清零，和结果装入 ADRES 寄存器，如果 A/D 中断使能，单片机将从睡眠中唤醒，如果 A/D 中断不使能，尽管 ADON 位仍保持置 1，A/D 模块将被关闭。

当 A/D 时钟源是另一种时钟选择（不是 RC），尽管 ADIN 位仍保持 1，睡眠指令将导致目前的转换中断和 A/D 模块被关闭。

关闭 A/D 把 A/D 模块放到它的最低电流消耗状态，

注：对于 A/D 模块在睡眠时运行，A/D 时钟源 RC 必须置 1（ADCS〈1:0〉=11），为了在睡眠时执行一个 A/D 转换，保证睡眠指令即刻跟在将 GO/DONE 位置 1 的指令之后。

11. 6: 复位影响

单片机复位强迫所有寄存器复位到他们的复位状态，强迫 A/D 转换模块关闭和在进行的转换被中断。

在 ADRESH: ADRESL 寄存器的值在上电复位时不变。ADRESH: ADRESL 寄存器在上电复位之后将包含未知数据。

12. 0: CPU 特性

这些单片机是一个有这样特征的主机。通过消除外部元件使系统可靠性趋向最大，成本趋向最低，提供低功耗运行方式和允许代码保护。

这些特征是：

- 振荡器可选择
- 复位：
 - 上电复位 (POR)，
 - 上电定时器 (PWRT)。
 - 振荡器启动定时器 (OST)。
 - 降压复位
- 中断
- 监视定时器 WDT
- 睡眠
 - 代码保护
 - ID 存储单元
 - 在线串行编程
 - 低电压内部电路串行编程
 - 内部电路调试器

这些单片机有一个监视定时器，它只能通过配置位 WDTE 来关闭。它自己带有 RC 振荡器以增加它的可靠性。有两个提供所需要的上电延迟定时器。一个是振荡器启动定时器 (OST)，它保证芯片处在复位状态直到晶体振荡器稳定。另一个是上电定时器 (PWRT)，它只在上电时提供一个固定 72ms 的延迟。和设计保证部件处在复位状态直到所供电源稳定。带有这两个片内定时器，大多数的应用不再需要外外部复位电路。

睡眠方式被设计成低电流省电方式。用户可以通过外部复位，WDT 唤醒或通过中断来唤醒睡眠状态。几种振荡器可供选出适合应用的一种，EC 振荡器允许用户直接驱动单片机，而 HS 振荡器允许使用高速。晶体振荡器。一组设定位可用来选择不同的振荡器。

12. 1 设定位

设定位可通过编程（读 0）或不编程（读 1）来选择不同的单片机配置。这些位被映射在程序存储器的 2007h 存储单元。

用户将要注意，地址 2007h 是在用户程序存储器空间之外。实际上，它属于特殊测试/设置存储器空间 (2000h~3FFFh)，它只有在编程期间才被访问。

12. 2 设定振荡器

12. 2. 1 振荡器的类型

PIC16F872 可以用四种不同的振荡器的模式来运行，用户可以通过对设置位 (FOSC <1: 0>) 的编程来选择四种类型中的一种。

- LP: 低功耗振荡器
- XT: 晶体/陶瓷振荡器
- HS 高速晶体/陶瓷振荡器
- RC: RC 振荡器

12. 2. 2 晶体/陶瓷振荡器

在 XT 或 LP 或 HS 方式中晶体或陶瓷谐振器和 OSC1/CLKIN 和 OSC2/CLKOUT 引脚相连来建立振荡（见图 12-1）。PIC16F87X 振荡器的设计需要用一个并行切割晶体。使用串行切割晶体可以给出一个不是生产厂家所指定的频率。当在 XT, LP 或 HS 方式时，单片机有一个外部时钟源来驱动/OSC1/CLKIN 引脚。

12. 2. 3 RC 振荡器

对于定时不需要很精确的应用，采用 RC 振荡器具可节约成本。RC 振荡器的频率是电源电压，电阻和电容值以及工作的环境温度的一个函数，另外，由于生产过程中参数的变化，振荡器频率对于不同的芯片也有所不同，再者，封装类型不同所导致的不同的分布电容也将影响振荡器具的频率，特别对于小电容值。用户还需要考虑由于所用外部电阻和电容的误差所引起的偏差。图 12-3 给出 R/C 组合的振荡器如何与 PIC16F872 连接。

12.3: 复位

PIC16F87X 单片机在各种复位间的区别：

- a) 上电复位 (POR)
- b) MCLR 在正常运行下复位
- c) MCLR 复位在 SLEEP 期间
- d) WDT 在正常运行期间溢出复位
- e) WDT 唤醒 (在 SLEEP 期间)
- f) 降压复位

一些寄存器用任何复位方法都有不会复位。它们在 POR 上未知，在其它任何复位时不会改变。其它大多数寄存器在运行期间都能通过上电复位 (POR)，MCLR、WDT 复位，在睡眠期间 MCLR 复位和在 BOR 上复位到“复位状态”。它们不受 WDT 唤醒的影响，因为这些复位被当成正常运行的继续。TO、PD 位，它们在不同的复位状态中被置 1 或清 0 见表 12-4，用软件来判别这些位以决定复位的性质，见表 12-6 对所有寄存器的复位状态的描述。

一个片内复位电路的方框图见图 12-4 略。

单片机在 MCLR 复位通道上有一个 MCLR 噪音过滤器。过滤器将探测和忽略小的脉冲。然而一个有效的 MCLR 脉冲必须符合最小脉冲宽度。

应该注意一个 WDT 复位不会将 MCLR 引脚驱动为低电平。

12.4: 上电复位 (POR)

当 VDD 被探测到上升时 (在 1.2V~1.7V 之间)，产生一个上电复位脉冲。为了充分利用 POR，把 MCLR 引脚连到 Vdd (通过一个电阻器)。这将省去通常用来产生上电复位的外部 RC 元件。需要确定一个 Vdd 最大上升时间 (详见电气特性)。对于一个缓慢上升的时间。

当单片机开始运行时 (退出复位条件)，单片机运行参数必须达到要求以保证运行 (象电压、频率、温度)，如果这些条件达不到，单片机必须一直保持复位状态直到运行参数达到要求。降压复位可用来满足启动条件。

12.5: 上电定时器 PWRT

上电定时器在从 POR 上电时提供一个固定的 72ms 正常溢出。PWRT 运行在一个内部的 RC 振荡器上。随着 PWRT 一起作用，单片机一直保持在复位状态。PWRT 的时间延时允许 VDD 上升到一个可接受的电平，提供一个设定位使能/不使能 PWRT。

上电时间延时随着单片机的不同，VDD，温度以及生产过程中的变化而不同..详细地参见 DC 参数

12.6: 晶体振荡器起动定时器 (OST)

在 PWRT 延时结束之后晶体振荡器起动定时器提供一个 1024 个振荡器周期的延时，(从 OSC1 输入) 这样保证晶体振荡器或者谐振器开始振荡和稳定。

OST 溢出只在为 XT，LP 和 HS 方式下和只在上电复位或从睡眠中唤醒时才起作用。

12.7: 降压复位 (BOR)

一个设置位 BODEN，能够不使能 (如果编程清零) 或使能 (如果置 1) 降压复位电路。如果 VDD 下降低于 4.0V，参考 VBOR 参数 D005 (VBOR) 比在表 12-6 里的参数大，降压情况将复位芯片。如果 VDD 下降低于 4.0V，比参数 (TBOR) 小，复位芯或许不会发生。

一旦降压发生，芯片将保持在降压复位状态直到 VDD 上升高于 VBOR。采用上电定时器保持单片机在复位状态 72ms。如果在 TPWRT 期间，当带有上电定时器复位的 VDD 上升高于 VBOR 时，降压复位将重新开始。当降压复位能使时，上电定时器应该一直被使用而不管 PWRT 设置位的状态。

12.8: 溢出顺序

在上电时，溢出顺序是：在 POR 复位出现时，

当一个 POR 复位发生时，PWRT 延迟开始。当 PWRT 结束 (LP，XT，HS) 时，OST 开始计数 1204 个振荡器周期。当 OST 结束时，单片机脱离复位。

如果 MCLR 保持较长时间低电平，溢出将会中止。把 MCLR 变为高电平时，溢出立即运行。这一点在为测试目的或达到同步时比用并行运行的 PIC16CX 单片机更加有用。

表 12-5 给出了特殊寄存器复位条件，表 12-6 给出了所有寄存器复位条件。

12. 9: 电源控制/状态寄存器 (PCON)

电源控制/状态寄存器，根据单片机，PCON 有两位。

位 0 是 BOR (降压复位状态位) POR 在上电复位时是未知的。用户必须将这位置 1。在随后的复位上如果 BOR 是 0，它表明发生降压复位。BOR 状态位是随意位，和如果降压复位电路不能使，BOR 位是不能断定的。(通过对在设置字里的 BODEN 位清零)

位 1 是 POR (上电复位)。在 POR 时是被清零的和在其它方面不受影响。在一个 POR 之后，用户必须将这位置 1。在随后的复位上如果 POR 是 0，它表明发生 POR 复位。

12. 10: 中断

PIC16F87X 有 10 个中断源。中断控制寄存器 (INTCON) 用标志位记录单个中断请求，它有单个和全体中断使能位。

注：不管和他们相关的屏蔽位或 GIE 位状态，单个中断标志位是置 1 的。

一个全体中断使能位 GIE (INTCON <7>) 使能或不使能所有的中断。当位 GIE 使能时，一个中断标志位和屏蔽位置 1 时，中断将立即引导。单个中断通过他们在各种寄存器里相应使能位能够不使能。不管 GIE 位的状态如何，单个中断都是置 1 的，GIE 位在复位时清零。

中断返回指令，RETFIE 将 GIE 位置 1 来重新使能中断。

RB0/INT 引脚中断，RB 端口改变中断和 TMR。溢出中断标志都包括在 INTCON 寄存器里。

外部中断标志是包含在特殊功能寄存器 PIR1 和 PIR2 里，相应的的中断使能位包含在特殊功能寄存器 PIE1 和 PIE2 里，和外部中断使能位包含在特殊功能寄存器 INTCON 里。

当一个中断响应时，GIE 位清零使任何更多的中断不能使，返回地址压入堆栈，程序计数器 PC 装入 0004h，一旦用中断服务程序，通过查询中断标志位来决定中断源，中断标志位在重新使能之前必须用软件清零以避免重复中断。

对于外部中断事件，象 INT 引脚或端口 B 的变化中断，中断等待时间将有 3~4 个指令周期。当中断事件发生时精确的等待时间不定，等待时间的差别为一个或两个指令周期指令。不管与他们相应的屏蔽位和 GIE 位的状态如何，单个中断标志位置 1

12. 10. 1: INT 中断

在引脚 RB0/INT 引脚上的外部中断为边沿触发，如果位 INTEDG (OPTION <6>) 是置 1 时，为上升沿触发，如果 INTEDG 位是清零的，为下降沿触发。当在 RB0/INT 引脚上出现一个有效边沿时，标志位 INTF (INTCON <1>) 被置 1。通过对使能位 INTE (INTCON <4>) 清零，使这个中断不能使。标志位 INTF 在重新使能这个中断之前必须用中断服务程序这个软件来清，如果 INTE 在进入睡眠之前 3 被置 1，INT 中断能将处理器从睡眠中唤醒，全体中断使能位 GIE 的状态决定着 CPU 在唤醒之后是否进入中断，详细见 12.13 节的睡眠方式。

12. 10. 2: TMR0 中断

在 TMR0 寄存器里的一个溢出 (FFh~00h) 将标志位 T0IF (INTCON <2>) 置 1。通过对使能位 T0IE (INTCON <5>) 置 1 或清零来决定中断使能或不使能。(见 5.0 节)

12.10.3: 端口 B 变化中断

一个在 PORTB<7:4>上的输入变化将标志位 RBIF (INTCON <0>) 置 1。通过对使能位 RBIF (INTCON <4>) 置 1/清 0 来决定中断使能/不使能。

12.11: 在中断期间上下内容的保存

在一个中断期间，只有返回程序计数的值被保存在堆栈里。典型的是用户希望在一个中断期间保存关键的寄存器。即工作寄存器和状态寄存器。这将在软件来实现。

对于 PIC16F873/874 单片机，寄存器 W-TEMP 必须用块 0 和块 1 来定义和必须从块基本地址以相同的偏移量来定义。(即：如果 W-TEMP 用块 0 的 0*20 来定义，它必须也能用在块 1 里的 0*A0 来定义。

因为在 PIC16F876/877 单片机里每个块的高 16 位字节是公用的，临时保存寄存器 W-TEMP，STATUS-TEMP 和 PCLATH-TEMP 应该放在这里。这 16 个存储单元不要求分块和因此使它存储和重新存储内容较容易。例 12-1 可以用作存储和重新存储中断内容。

12. 12: 监视定时器 (WDT)

监视定时器是一个自由运行的片内专用振荡器，它不需要任何外部元件。这个 RC 振荡器和

OSC1/CLKIN 引脚的外部 RC 振荡器是分离的。这就意味着即使在单片机的 OSC1/CLKIN 和 OSC2/CLKOUT 引脚上的时钟已停止的情况下，WDT 仍能运行。例如，通过执行一条 Sleep 指令。

在正常运行期间，一个 WDT 溢出将使单片机复位（EDT 复位）。如果单片机在睡眠方式，一个 WDT 溢出将使单片机唤醒和恢复正常通过运行（WDT 唤醒）。在 STATUS 寄存器里的 TO 位通过 WDT 溢出将被清零。

通过编程设定 WDTE 位为 0，使 WDT 永久不能使用。

WDT 溢出周期值在电特性节可以发现。当把预分频器分配给 WDT，预分频器分配和预分频器值在 OPTION-REG 寄存器里是置 1 的。

注：CLRWDT 指令和 SLEEP 指令清零 WDT 和后分频器，如果分配给 WDT 时，和阻止它溢出和产生一个单片机复位条件。

注：当执行一个 CLRWDT 指令和预分频器分配给 WDT 时，预分频器计数器将被清零，但预分频器分配不会改变。

12. 13 低功耗方式 Sleep

通过执行一条睡眠指令就进入低功耗方式。

如果使能，WDT 将被清 0，但仍保持运行。To (STATUS<4>) 位被置 1，PD 位 (STATUS<3>) 被清 0，和振荡器驱动器被停止。所有 I/O 端口保持 sleep 指令执行前它们所有的状态（驱动高、驱动低或高阻）。

在这种方式时为使电流消耗最低，把所有的 I/O 引脚放在 VDD 或 VSS 电平上，以保证外部电路从 I/O 引脚，低功耗的 A/D，不使能的外部时钟中拉出电流。把所有呈高阻输入 I/O 引脚上拉或下拉以避免输入浮空所带来的开关电流。为使电流消耗最低，T0CK1 输入电平应该在 VDD 或 VSS 上。在 PORTB 上的片内上拉应该考虑。

MCLR 引脚必须在逻辑高电平（VIHMC）。

12. 13. 1 睡眠唤醒

单片机通过下列事件之一唤醒睡眠。

1. 在 MCLR 引脚上的外部复位输入。
2. 监视定时器唤醒（如果 WDT 使能）
3. INT 引脚中断、RB 端口变化中断和一些外部中断。

外部 MCLR 复位将引起单片机复位，象一个程序执行结果和引起唤醒的所有其它事件被考虑，在状态寄存器里的 TO 和 PD 位用来决定 单片机复位的原因，在上电复位时，PD 置 1。当执行睡眠时被将被清零，如果 WDT 发生溢出，TO 位将被清零和引起唤醒。

下面的是能将单片机从睡眠中唤醒的外部中断。

1. PSP 读或写
2. TMR1 中断，TIMER1 必须作为一个异步计数器运行。
3. CCP 捕捉方式中断
4. 特别事件触发器（TIMER1 使用外部时钟异步模式，CCP1 是用比较器模式）
5. SSP（START/STOP）位探测中断。
6. SSP 用从模式（SPI/I2C）发送和接收
7. USART RX 或 TX（同步从模式）
8. A/D 转换（当 A/D 时钟源是 RC 时）
9. EEPROM 写操作完成

因为在睡眠期间，无片内时钟出现，所以其它外部不能产生中断。

当执行 Sleep 指令时，下一个指令（PC+1）被预取，对于通过中断事件唤醒的单片机，相应的中断使能位置的 1（使能），唤醒是不管 GIE 位的状态的。如果 GIE 位清零（不使能），单片机在 Sleep 指令之后连续执行指令。如果 GIE 位是置 1 的（使能），单片机在 Sleep 指令之后执行指令，然后分支到中断地址（0004h）。如不期望执行紧跟在 Sleep 指令之后的指令，用户应该在 Sleep 指令之后有一个 NOP 指令。

12. 13. 2: 用中断唤醒

当全体中断不使能和任何中断源已包括它的中断使能位和中断标志位是置 1 的时，将发生下面之一：

- 如果在 Sleep 指令执行之前发生中断，Sleep 指令将做为一个 NOP 完成，因此 WDT 和 WDT

后分频器将不被清零，TD 位将不会置 1 和 PD 位将不会清零。

- 如果在 Sleep 指令执行期间或之后发生中断，单片机将立即唤醒睡眠，在唤醒之前 Sleep 指令将完全执行，因此，WDT 和 WDT 后分频器将被清零，TD 位将被置 1 和 PD 位将被清零。

即使在执行 Sleep 指令之前标志位被检查，在 Sleep 指令完成之前标志位变成 1 是可能的。为了决定 Sleep 指令是否执行，应该检测 PD 位，如果 PD 位是置 1 的，Sleep 指令将做为 NOP 执行。

为保证 WDT 清零，在 Sleep 指令之前应该执行 CLRWDT 指令。

12. 14: 内部电路调试器

当在设置字里的 DEBUG 位被编程为 0 时，内部电路调试器功能使能。当使用 MPLAB 时，这种功能允许简单的功能调试，当微处理器有这种使能功能时，其中一些资源对于通常使用是不可使用的。表 11-7 给出了通过后台调试器的功能消耗。

为了使用单片机的内部电路调试器功能，设计必须实现内部电路串行编程以连接 MCLR/VPP，VDD，GND，RB7，RB6。这样将内部电路调试器和微芯片或其他公司的开发工具接口。

12. 15: 程序验证/代码保护

如果代码保护位不被编程，片内程序存储器可以被读出以便验证。

12.16: 识别码 ID 存储单元

四个存储器单元被设定为 ID 存储单元，在这里用户可以储存检查和以及其它代码识别数。这些存储单元在正常执行中是不可访问的，但在编程/修改期间是可读写的。推荐使用 ID 存储单元的低 4 位。

12. 17: 内部电路串行编程

PIC16F87X 微处理器在应用电路结束时能够被串行编程。可用两根线作为时钟线和数据线和其它多种线用作电源，接地和编程电压线来简单地完成。可提供给用户来编程的单片机制造板，然后在单片机即将卖出时，对它进行编程，这将允许提供最新固件和用户定制的可编程固件。

当使用 ICSP 时，如果执行一个大量的擦除，部件必须用 4.5V 到 5.5V 供电。这包括从一个打开状态到关闭状态两个的代码保护的重新编程。对于 ICSP 的所有其它情况，部件可以在通常操作电压下被编程。这意味着振荡器值，独一无二的用户代码或用户码可以被重新编程或增加。

对于详细情况见 DS30277。

12. 18: 低电压 ICSP 编程

设置位里的 LVP 位使能低电压 ICSP 编程。这种模式允许微处理器在操作电压范围内使用一个 VDD 源通过 ICSP 编程。这意味着 VPP 不必带到 VIHh，但在标准操作电压时能够替换。在这种模式里，RB3/PGM 引脚被制定为编程功能和停止一个普通的 I/O 引脚。在编程期间，VDD 应用到 MCLR 引脚。为了进入编程模式，VDD 必须应用到提供 LVP 位置 1 的 RB3/PGM。从工厂里 LVP 位的缺省值是 1。

注：1：通过应用 VIHh 到 MCLR 引脚，不管 LVP 位的状态，都一直可用高电压编程模式。

2：当在低电压 ICSP 模式时，RB3 引脚可不再用作普通的 I/O 引脚。

3：当使用低电压编程和在 PORTB 上拉被使用时，在 TRISB 寄存器里的位 3 必须清零以在 RB3 上的上拉不能使和保证单片机合适的操作。

如果低电压编程模式不能使，那么 LVP 位不能编程到 0 和 RB3/PGM 变成一个数字 I/O 引脚。然而，当在 MCLR 上用 VIHh 进入编程时，LVP 位只可以被编程。当 MCLR 上使用高电压时，LVP 位只可以被充电。

应该注意，一旦 LVP 位被编程到 0，只有高电压编程模式可用和只有高电压编程模式可用到编程单片机。

当使用低电压 ICSP 时，如果执行一个大量的擦除，部件必须用 4.5V 到 5.5V 供电。这包括从一个打开状态到关闭状态两个的代码保护的重新编程。对于 ICSP 的所有其它情况，部件可以在通常操作电压下被编程。这意味着振荡器值，独一无二的用户代码或用户码可以被重新编程或增加。

13. 0: 指令系统

PIC16CXX 的每条 14 位字长的指令被分为一个操作码和一个或多个操作数，其中操作码用于指令类型，操作数用于确定指令的操作。PIC16CXX 指令一览表见表 13-2，一般指令可分成面向字节操作，面向位操作和常数和控制类操作，表 13-1 给出了操作码字段功能描述。

对于面向字节操作指令，“f”表示文件寄存器指针。“d”表示目标指针，文件寄存器指针通过指令用于确定文件寄存器中的一个，目标指针确定操作结果存放在哪里，如果“d”是“0”，结果存放在工作寄存器中，如果是“1”，结果存放在通过指令指定的文件寄存中。

对于面向位的操作指令,“b”代表一个“位”指针,它选择受影响的位,“f”表示该位所在的文件寄存器地址。

对于常数和控制操作而言,K表示一个8位或9位的常数或文字值。

指令系统是高度对称的,分成三个基本种类

- 。面向字节操
- 。面向位的操作
- 。对于常数和控制操作

所有指令均在一个指令周期内完成,除非条件检测为真时,或指令执行结果使程序计数器改变时,在这种情况下,完成指令需要两个指令周期。它的第二个周期作为一个NOP执行。一个指令周期由四个振荡周期组成,这样对于一4MHz的振荡器频率其指令执行时间为1μS,如果条件检测为真或当指令执行结果使程序计数器值改变时,这时指令执行周期为2μS。

表13-2列出了通过MPASM汇编器所知道的指令。

图13-1表示三种指令所有的普通格式,在图中所有的例子用十六进制数表示。0×hhh。h表示一个十六进制数。

14. 0 支持开发工具

PICmicro 单片机可得到全范围的硬件和软件开发工具的支持

- 集成开发环境
 - MPLABTM IDE 软件
- 汇编器/编译器/连接器
 - MPASM 汇编器
 - MPLAB-C17 MPLAB-C18 (C 编译器)
 - MPLINK/MPLIB 连接器/程序管理器
- 模拟器
 - MPLABTM SIM 软件模拟器
- 仿真器
 - MPLAB-ICE 实时内部电路仿真器。
 - PICMASTER/PICMASTER-CE 内部电路仿真器。
 - ICEPIC-TM
- 内部电路调试器
 - PIC16F877 的 MPLAB-ICD
- 单片机编程器
 - PROMATE II 通用编程器。
 - PICSTART PLUS 基本样机编程器,
- 低成本演示板
 - SIMICE
 - PICDEM-1
 - PICDEM-2
 - PICDEM-3
 - PICDEM-17
 - SEEVAL
 - KEELOQ

14. 1 MPLAB 集成开发环境软件

• MPLAB IDE 软件给 8 位单片机市场上带来前所未有的容易的软件开发。MPLAB 是一个以窗口为基础的应用。它包括:

- 多种功能
 - 编辑器
 - 模拟器
 - 编程器(固态分离)

-仿真器（固态分离）

- 全特征的编辑器
- 项目管理器
- 可定制的工具条和键位映射
- 状态条
- 在线帮助

MPLAB 允许你做下面事情：

- 编辑你的源文件（包括汇编和 C 文件）
- 一种触摸汇编（或编译）和下载到 PICMICRO 工具（自动修正所有项目信息）
- 调试使用
 - 源文件
 - 绝对的列表文件
 - 目标代码

使用带有 MICROCHIP 公司的模拟器的 MPLAB 的能力，MPLAB-SIM 它允许一个兼容的平台；用很少的重新培训就很容易掌握从低价的模拟器到全功能的仿真模拟器。

14. 2 MPASM 汇编器

MPASM 对于所有 Picmicro Mcus 是一个全特征的通用 MACRO 宏汇编器，对于单片机编程器在 HEX 文件的组成中它能直接产生绝对代码，或者为 MPLINK 产生浮动目标。

MPASM 有一个命令线接口和一个窗口外壳，它能够在 WINDOW 3. X 或更高系统作为独立应用，MPASM 产生浮动目标文件，intel 标准 HEX 文件，MAP 文件以详述存储器使用和参考符号，一个包含绝对源程序的 LST 文件，和产生机器代码，和一个为 MPASM 调试的 COD 文件。

MPASM 特征包括：

- MPASM 和 MPLINK 是被集成到 MPLAB 项目中
- MPASM 允许用户定义为流线汇编而创宏。
- MPASM 允许为多重目标源文件的条件汇编
- MPASM 伪指令完全允许控制汇编过程。

14. 3 MPLAB-C17 和 MPLAB-C18 编译器

MPLAB-C17 和 MPLAB-C18 代码开发系统对 MICROCHIP 的 PIC17CXXX 和 PIC18CXXX 系列的单片机是一个完全的 ANSI' C' 编译器和集成开发环境，编译器提供了强大的集成能力以至没有比它更易使用的编译器。

对于较容易的源级调试，编译器提供符号情况以及和 MPLAB 1DE 存储显示器兼容的情况。

14. 4 MPLINK/MPLIB 连接器/程序库管理器

MPLINK 是 MPASM 和 MPLAB-C17 和 MPLAB-C18 的一个浮动的连接器，它可以从汇编或 C 源文件中连接浮动目标以及用从连接程序引脚中使用伪指令连接预编译程序库。

MPLIB 是一个使用 MPLINK 的预编译码程序库，当一个程序库中的程序被称为另一个源文件，只有那些含有那个程序的模块将和应用连接，这将允许在很多不同应用中有效地使用大型程序库。MPLIB 管理程序库文件的创建和修改。

MPLINK 特征包括：

- MPLINK 和 MPASM 和 MPLAB-C17 和 MPLAB-C18 一起工作。
- MPLINK 允许所有的存储器区域被定成节以提供灵活的连接时间。

MPLIB 特征包括：

- 因为用包括单独的程序库代替很多小的文件，因此 MPLIB 使连接更容易。
- MPLIB 通过将相关的模块分组在一起，MPLIB 有助于代码的维护。
- MPLIB 命令允许创建程序库和增加，列表，取代，删除或抽取模块。

14. 5 MPLAB-SIM 软件模拟器

MPLAB-SIM 软件模拟器允许代码在 PC 主机环境下开发，它允许用户在一个指令级上模拟 PICMICRO 系列单片机，对于所给的任何指令，用户可以修改或者测试任何数据部分或者对于任何引脚提供外部模拟，输入/输出基数可由用户设置和可以用单步，执行中中断，或用跟踪方式来完成执行。

MPLAB-SIM 用 MPLAB-C17，MPLAB-C18 和 MPASM 来完全支持符号调试，软件模拟器提供低成本

的灵活性以开发和调试实验室环境之外的代码，使它成为最佳多项目软件开发工具。

14. 6 MPLAB-IEC 高性能、带有 MPLAB IDE 通用内部仿真器。

MPLAB-IEC 通用内部电路仿真器是提供给产品开发工程师的开发工具。对于 PICMICRO 单片机，它带有一个完善的单片机设计工具系统，MPLAB-IEC 带有 MPLAB 集成开发环境，它允许编辑，“MAKE”和下载以及在独立环境下进行源程序调试。

可交换单片机模块允许系统极容易地设定为不同的单片机仿真。MPLAB-IEC 通用结构允许扩展支持所有新的 MICRO 单片机。

MPLAB-IEC 仿真系统被设计成实时仿真系统，它带有许多先进的特征，这些特征通常建立在更加昂贵的开发工具上。选择 PC 机或兼容机 386（或更高）的计算机平台和 MICROSOFT WINDOWS 3. X 和 WINDOWS95 操作系统使这些特征对你这个最终用户更好地使用。

MPLAB-ICE 2000 是一个全特征的仿真系统，它带有增强跟踪、触发器和数据监视等特点。两个系统通过在整个 PICMICRO MCU 的运行速度范围内将会运行。

14. 7: PICMASTER/PICMASTER CE

采用 Microchip 技术的 PICMASTER 系统是一个全特征、专业品质的仿真器系统，这种灵活的内部电路仿真器为仿真 Microchip 的 8 位 Picmicro 单片机(MCUS)提供一个高品质的通用平台，PICMASTER 系统是在世界范围销售，它带有一个 CE 服从性模型可用于很多欧盟国家。

14. 8: ICEPIC

ICEPIC 是一个低价的内部仿真器，用它解决 Microchip 公司的 PIC16C5X、PIC16C6X、PIC16C7X 和 PIC16CXXX 系列 8 位一次性可编程单片机，通过使用可交换的专用模块或子插件板，模块化系统可以支持 PIC16C5X 或 PIC16CXX 产品的子设备，没有给予目标应用，电路仿真器就不能仿真。

14. 9: MPLAB-ICD 内部电路调试器，

Microchip 公司的内部电路调试器，MPLAB-ICD 是一个功能强大的低成本运行时间开发工具，这个工具是基于快速的 PIC16F877 和用作这种或那种 PIC16CXX 系列中的 Picmicro 单片机的开发。MPLAB-ICD 利用内部电路调试能力建立在 PIC16F87X 里，这种特征以及 Microchip 公司的内部电路串行编程协议，MPLAB 集成开发环境的图形用户接口提供价格可取得内部电路快速编程和调试，这将使设计者通过观察变化，单步试通和设置断点来开发和调试源代码，全速运行能够实时测试硬件，MPLAB-ICD 对快速 PIC16F87X 系列了是一个编程器。

14. 10. PROMATE II 通用编程器

PIC12CXXX II 通用编程器是一个全特征编程器，它能够独立运行或在 PC 机上运行，PROMATE 是 CE 的附属产品。

PROMATE II 具有可编程的 Vdd 和 Vpp，为了增加可靠性，允许在 Vdd 最小和 Vdd 最大时改变程序存储器，它有 LCD 显示，用来显示错误信息，用钥匙进入命令和可拆开的插接装配模块以支持不同的封装类型。用于独立方式的 PROMATE II 够读，更改或编程 PICMICRO 单片机，用这种方式也可以设定配置和代码保护值。

14. 11: PICSTART PLUS 基本开发系统

PICSTART 编程器是很容易使用的低成本样机编程器，它通过 CMOS (RS-232) 端口之一与 PC 机相连。MPLAB 集成开发环境软件使编程器应用起来简单有效，

PICSTART PLUS 不要求对产品编程。

PICSTART PLUS 支持所有的 PICMICRO 带有多达 40 个引脚的单片机，大的引脚数单片机象 PIC16C92X 和 PIC17C76X 可以用适配器插件支持。PICSTART PLUS 是 CE 的附属产品。

14. 12: SIMICE 基本硬件仿真器

SIMICE 是一个基本硬件开发系统，它设计运行在 PC 机为基础的环境中，它带有 MICROCHIP 公司的 MPLAB-SM 仿真器。SIMICE 和 MPLAB-SIM 在 MICROCHIP Technology 公司的 MPLAB 集成开发环境软件中运行。要说明的是 SIMICE 为 PIC12C5XX、PIC12CE5XX、PIC16C5X 系列的 8 位单片机提供硬件仿真。SIMICE 用插件和 MPLAB-SIM 连接以提供不实时的 I/O 端口仿真。SIMICE 结构使开发者运行仿真器代码以驱动目标系统。另外目标系统能够提供输入到仿真器的代码，这种能力允许简单地交互调试，而没有必须产生 MPLAB-SIM 激励文件，SIMICE 对于基本系统的开发是一个有价值的调试工具。

14. 13 低成本 PICMICRO-1 演示板

PICDEM-1 是一个简单的演示板，它具有能演示好几种 MICROCHIP 公司的单片机的能力，所支持的单片机是 PIC16C5X(PIC16C54 到 PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 和 PIC17C44. 它包括运行基本演示程序所需要的所有硬件和软件。用户

可以用所提供的 PICDEM-1 板在 PROMATE. II 或 PPICSTART-PLUS 编程器上对样品单片机编程或很容易地测试固件。用户也可以把 PICDEM-1 演示板连接到 MPLAB-ICE 仿真器上和为测试而下载固件到仿真器，另外用户可通过建立一些附加硬件并和单片机插接件相连，用来增加样机的使用范围。其中一些特征包括 RS-232 接口；仿真模拟输入电位器；弹出按钮开关和 8 个连向端口的 LEDS。

14. 14 PICDEM-2 低成本 PIC16CXX 演示板

PICDEM-2 是一个支持 PIC16C62, PIC16C64, PPIC16C65 和 PIC16C73 和 PIC16C74 单片机的简单演示板。它包括所有运行基本演示程序所需要的所有硬件和软件。用户可以用所提供的 PICDEM-2 板在 PROMATE. II 或 PPICSTART-PLUS 对样品单片机编程和很容易地测试固件。MPLAB-ICE 仿真器可以用 PICDEM-2 板测试固件。用户可以通过附加硬件，并和单片机插接件相连，用来增加一些样机的使用范围。其中一些特征包括 RS-232 接口；弹出按钮开关；仿真模拟输入电位器；串行 EEPROM 演示；I2C 总线的应用；连接 LCD 模块和辅足键盘的首标示分开的。

14. 15 PICDEM-3 低成本 PIC16CXXX 演示板

PICDEM-3 是一个用 PLCC 封装，支持 PIC16C923 和 PIC16C924 单片机的简单演示板。它也支持未来带有 LCD 模块的 44 引脚的 PLCC 单片机，它包括运行基本演示程序所需要的所有硬件和软件。用户可以用所提供的 PICDEM-3 板在 PROMATE. II 编程器或带有适配器插件的 PICSTART PLUS 上对样品单片机编程和极容易地测试固件。可以用带有 MPLAB-ICE 仿真器测试固件。用户可以通过增加硬件和把它连接到单片机的插接件上来增加样机的使用范围。其中一些特征象 RS-232 接口；弹出按钮开关；仿真模拟输入电位器；热敏电阻；接向外部的 LCD 模块和辅足键盘的首标示分开的。在 PICDEM-3 板上提供一个 LCD 显示板，它带有 4 个普通的和 12 个程序段，它能够显示时间、温度和星期的天数，PICDEM-3 接供一个附加的 RS-232 接口和 WINDOWS 3.1 软件用以在 PC 机上显示出已译码的 LCD 信号。一个简单的串行接口允许用户对 LCD 信号建立一个硬件译码器。

14. 16: PICDEM-17

PICDEM-17 是一个鉴定板，它具有演示好几种 Microchip 单片机的能力。单片机包括 PIC17C752、PIC17C756、PIC17C762 和 PIC17C766，包括运行基本演示程序在内的所需的所有硬件，都提供在一个 3.5 英寸的软盘上。包括编程取样，和用户可以用其它使用 PROMATE II 或者 PICSTART PLUS 单片机编程器取样程序来对它擦除和编程并且很容易地调试和检测取样代码，另外 PICDEM-17 对于 MPLAB-ICE 或 PICMASTER 仿真器是不能使用的，和使用任何一个仿真器能够运行或修改采样程序，另外，对用户硬件通用样机范围是可用的。

14. 17 SEEVAL 鉴定和可编程系统

SEEVAL EEPROM 设计者的成套工具，它支持所有的 2 线和 3 线串行 EEPROMS。成套工具包括所有需要读，写，擦除或对于任何 MICROCHIP EEPROM 产品的特殊性能编程，这些产品包括 SMART SERIALTM 和安全系列。THE TOTAL ENDURANCE TM DISK 被包括用来帮助综合分析和可靠计数。总得成套工具能够有效地减少到市场时间和产生一个优化系统。

14. 18 KEELOQ 鉴定和可编程程序

KEELOQ 鉴定和可编程工具支持 MICROCHIP 的安全数据产品，HCS 鉴定成套工具包括一个 LCD 显示器以显示变化的代码，一个译码器进行译码传递和一个可编程接口来编程测试发送器。

15. 0~17.0 略