PIC24FJ256GA705系列

具有256 KB闪存和16 KB RAM的低引脚数16位通用单片机

高性能CPU

- 改进型哈佛架构
- 最高运行速度可达16 MIPS（32 MHz时）
- 8 MHz快速RC内部振荡器：
  - 96 MHz PLL选项
  - 多个时钟分频选项
  - 快速起振
- 17位 x 17位单周期硬件小数/整数乘法器
- 32位/16位硬件除法器
- 16位 x 16位工作寄存器阵列
- 优化的C编译器指令集架构
- 2个地址发生单元，分别用于数据存储器的读/写寻址
- 6通道DMA控制器

模拟特性

- 最多14个通道、可通过软件选择10/12位模数转换器：
  - 12位，200 ksps的转换速率（单个采样/保持）
  - 可在休眠模式下工作
  - 用于在较低AVDD下工作的电荷泵
  - 带隙参考输入功能
  - 窗口阈值比较功能
  - 自动扫描功能
- 3个带输入复用功能的模拟比较器：
  - 比较器参考电压可编程
- 高于/低于可编程VLVD值时产生LVD中断
- 充电时间测量单元（Charge Time Measurement Unit，CTMU）：
  - 可测量电容和时间
  - 可在休眠模式下工作

低功耗特性

- 休眠和空闲模式选择性关闭外设和/或内核，实现大幅降低功耗和快速唤醒
- 打盹模式允许CPU以低于外设的时钟速度运行
- 备用时钟模式允许在运行时切换到低时钟速度以选择性降低功耗

单片机的特性

- 电源电压范围为2.0V至3.6V
- 双稳压器：
  - 1.8V内核稳压器
  - 用于保持休眠模式的1.2V稳压器
- 工作环境温度范围为-40°C至+85°C
- ECC闪存（256 KB）：
  - 单位错误修正（Single Error Correction，SEC）
  - 双位错误检测（Double Error Detection，DED）
  - 耐擦/写次数达10,000次（典型值）
  - 数据保存时间：至少20年
  - 可在软件控制下自编程
- 16 KB SRAM
- 可编程参考时钟输出
- 可通过2个引脚进行在线串行编程（In-Circuit Serial Programming™，ICSP™）和在线仿真（In-Circuit Emulation，ICE）
- 支持JTAG边界扫描
- 故障保护时钟监视器操作：
  - 检测时钟故障并切换至片上低功耗RC（Low-Power RC，LPRC）振荡器
- 上电复位（Power-on Reset，POR）、欠压复位（Brown-out Reset，BOR）和振荡器起振定时器（Oscillator Start-up Timer，OST）
- 可编程低电压检测（Low-Voltage Detect，LVD）
- 灵活的看门狗定时器（Watchdog Timer，WDT）自带RC振荡器，以实现可靠运行

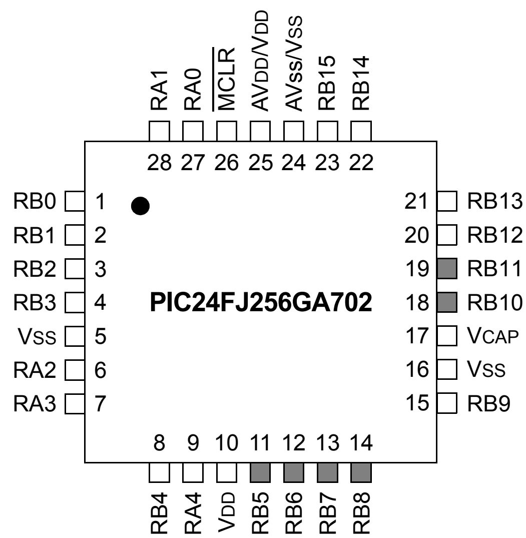
外设特性

- 所有I/O引脚均具有18 mA/18 mA的高灌/拉电流能力
- 独立低功耗32 kHz定时器振荡器
- Timer1：带外部晶振的16位定时器/计数器；Timer1可提供A/D触发信号
- Timer2和Timer3：16位定时器/计数器，可构成32位定时器；Timer3可提供A/D触发信号
- 3个输入捕捉模块，均带有一个16位定时器
- 3个输出比较/PWM模块，均带有一个16位定时器
- 4个MCCP模块，均带有一个专用16/32位定时器：
  - 1个6输出MCCP模块
  - 3个2输出MCCP模块
- 所有器件上均带有3个可变宽度的同步外设接口（Synchronous Peripheral Interface，SPI）端口；3种工作模式：
  - 3线SPI（支持全部4种SPI模式）
  - 8 x 16位或8 x 8位FIFO
  - I^2S模式
- 2个I^2C主/从模块，支持地址掩码和IPMI
- 2个UART模块：
  - 支持LIN/J2602总线（支持自动唤醒、自动波特率检测（Auto-Baud Detect，ABD）和间隔字符）
  - 支持RS-232和RS-485
  - IrDA®模式（硬件编码器/解码器功能）
- 5个外部中断引脚
- 并行主端口/增强型并行从端口（Parallel Master Port/Enhanced Parallel Slave Port，PMP/EPSP），8位数据，支持外部可编程控制（极性和协议）
- 增强型CRC模块
- 具有可编程分频比的参考时钟输出
- 2个可配置逻辑单元（Configurable Logic Cell，CLC）模块：
  - 2个输入和1个输出，均可映射到外设或I/O引脚
  - AND/OR/XOR逻辑和D/JK触发器功能
- 外设引脚选择（Peripheral Pin Select，PPS），允许对许多外设进行独立的I/O映射

器件	存储器		引脚	GPIO	DMA通道	外设													JTAG
	程序（字节）	SRAM（字节）				10/12位A/D通道	比较器	CRC	MCCP 6输出/2输出	IC/OC/PWM	16位定时器	I^2C	可变宽度SPI	LIN-USART/IrDA®	CTMU通道	EPMP（地址/数据线）	CLC	RTCC	
PIC24FJ64GA705	64K	16K	48	40	6	14	3	有	1/3	3/3	3	2	3	2	13	10/8	2	有	有
PIC24FJ128GA705	128K	16K	48	40	6	14	3	有	1/3	3/3	3	2	3	2	13	10/8	2	有	有
PIC24FJ256GA705	256K	16K	48	40	6	14	3	有	1/3	3/3	3	2	3	2	13	10/8	2	有	有
PIC24FJ64GA704	64K	16K	44	36	6	14	3	有	1/3	3/3	3	2	3	2	13	10/8	2	有	有
PIC24FJ128GA704	128K	16K	44	36	6	14	3	有	1/3	3/3	3	2	3	2	13	10/8	2	有	有
PIC24FJ256GA704	256K	16K	44	36	6	14	3	有	1/3	3/3	3	2	3	2	13	10/8	2	有	有
PIC24FJ64GA702	64K	16K	28	22	6	10	3	有	1/3	3/3	3	2	3	2	12	无	2	有	有
PIC24FJ128GA702	128K	16K	28	22	6	10	3	有	1/3	3/3	3	2	3	2	12	无	2	有	有
PIC24FJ256GA702	256K	16K	28	22	6	10	3	有	1/3	3/3	3	2	3	2	12	无	2	有	有

引脚图（PIC24FJ256GA702器件）

28引脚QFN和UQFN



图注： 有关引脚功能的完整描述，请参见表1。引脚排列可能变更。

注： 灰色阴影表示可承受5.5V电压的输入引脚。

表1： 引脚功能的完整描述（PIC24FJ256GA702 QFN和UQFN）

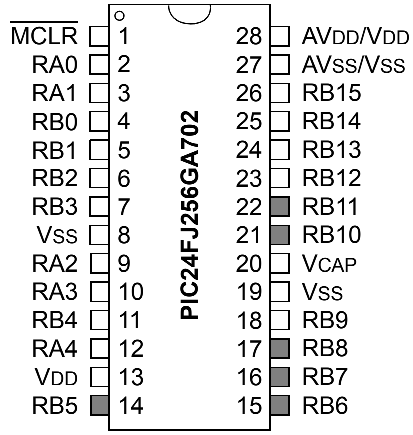
引脚	功能	引脚	功能
1	PGD1/AN2/CTCMP/C2INB/**RP0**/RB0	15	TDO/C1INC/C2INC/C3INC/$\overline{\text{TMPRN}}$/**RP9**/SDA1/T1CK/CTED4/RB9
2	PGC1/AN1-/AN3/C2INA/**RP1**/CTED12/RB1	16	VSS
3	AN4/C1INB/**RP2**/SDA2/CTED13/RB2	17	VCAP
4	AN5/C1INA/**RP3**/SCL2/CTED8/RB3	18	PGD2/TDI/**RP10**/OCM1C/CTED11/RB10
5	VSS	19	PGC2/TMS/REFI1/**RP11**/CTED9/RB11
6	OSCI/CLKI/C1IND/RA2	20	AN8/LVDIN/**RP12**/RB12
7	OSCO/CLKO/C2IND/RA3	21	AN7/C1INC/**RP13**/OCM1D/CTPLS/RB13
8	SOSCI/**RP4**/RB4	22	CVREF/AN6/C3INB/**RP14**/CTED5/RB14
9	SOSCO/PWRLCLK/RA4	23	AN9/C3INA/**RP15**/CTED6/RB15
10	VDD	24	AVSS/VSS
11	PGD3/**RP5**/ASDA1/OCM1E/RB5	25	AVDD/VDD
12	PGC3/**RP6**/ASCL1/OCM1F/RB6	26	$\overline{\text{MCLR}}$
13	**RP7**/OCM1A/CTED3/INT0/RB7	27	VREF+/CVREF+/AN0/C3INC/**RP26**/CTED1/RA0
14	TCK/**RP8**/SCL1/OCM1B/CTED10/RB8	28	VREF-/CVREF-/AN1/C3IND/**RP27**/CTED2/RA1

图注： **RPn**表示用于外设引脚选择（PPS）功能的可重映射引脚。

注： 引脚排列可能变更。

引脚图（PIC24FJ256GA702器件）

28引脚SOIC、SSOP和SPDIP



图注： 有关引脚功能的完整描述，请参见表2。引脚排列可能变更。

注： 灰色阴影表示可承受5.5V电压的输入引脚。

表2：引脚功能的完整描述（PIC24FJ256GA702 SOIC、SSOP和SPDIP）

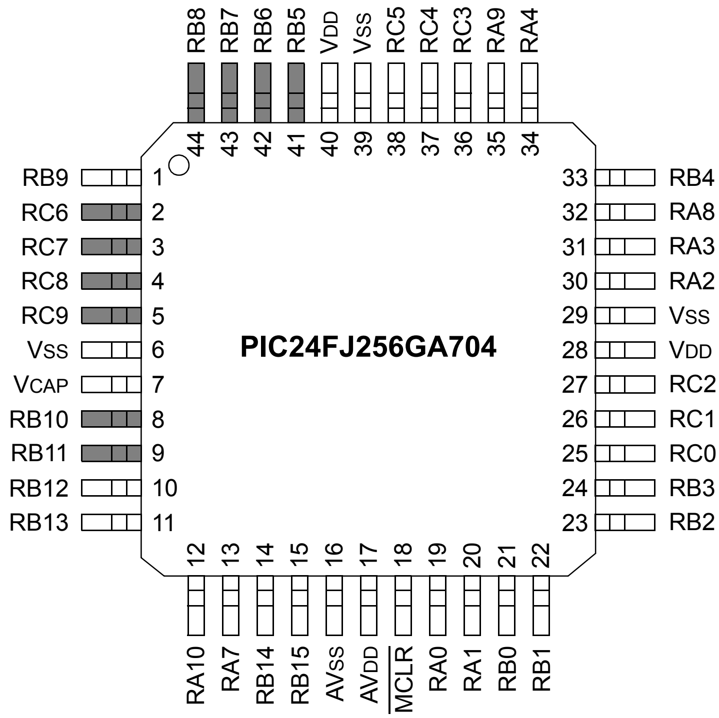
引脚	功能	引脚	功能
1	$\overline{\text{MCLR}}$	15	PGC3/**RP6**/ASCL1/OCM1F/RB6
2	VREF+/CVREF+/AN0/C3INC/**RP26**/CTED1/RA0	16	**RP7**/OCM1A/CTED3/INT0/RB7
3	VREF-/CVREF-/AN1/C3IND/**RP27**/CTED2/RA1	17	TCK/**RP8**/SCL1/OCM1B/CTED10/RB8
4	PGD1/AN2/CTCMP/C2INB/**RP0**/RB0	18	TDO/C1INC/C2INC/C3INC/$\overline{\text{TMPRN}}$/**RP9**/SDA1/T1CK/CTED4/RB9
5	PGC1/AN1-/AN3/C2INA/**RP1**/CTED12/RB1	19	VSS
6	AN4/C1INB/**RP2**/SDA2/CTED13/RB2	20	VCAP
7	AN5/C1INA/**RP3**/SCL2/CTED8/RB3	21	PGD2/TDI/**RP10**/OCM1C/CTED11/RB10
8	VSS	22	PGC2/TMS/REFI1/**RP11**/CTED9/RB11
9	OSCI/CLKI/C1IND/RA2	23	AN8/LVDIN/**RP12**/RB12
10	OSCO/CLKO/C2IND/RA3	24	AN7/C1INC/**RP13**/OCM1D/CTPLS/RB13
11	SOSCI/**RP4**/RB4	25	CVREF/AN6/C3INB/**RP14**/CTED5/RB14
12	SOSCO/PWRLCLK/RA4	26	AN9/C3INA/**RP15**/CTED6/RB15
13	VDD	27	AVSS/VSS
14	PGD3/**RP5**/ASDA1/OCM1E/RB5	28	AVDD/VDD

图注： **RPn**表示用于外设引脚选择（PPS）功能的可重映射引脚。

注： 引脚排列可能变更。

引脚图（PIC24FJ256GA704器件）

44引脚TQFP



图注： 有关引脚功能的完整描述，请参见表3。引脚排列可能变更。

注： 灰色阴影表示可承受5.5V电压的输入引脚。

表3：　　引脚功能的完整描述（PIC24FJ256GA704 TQFP）

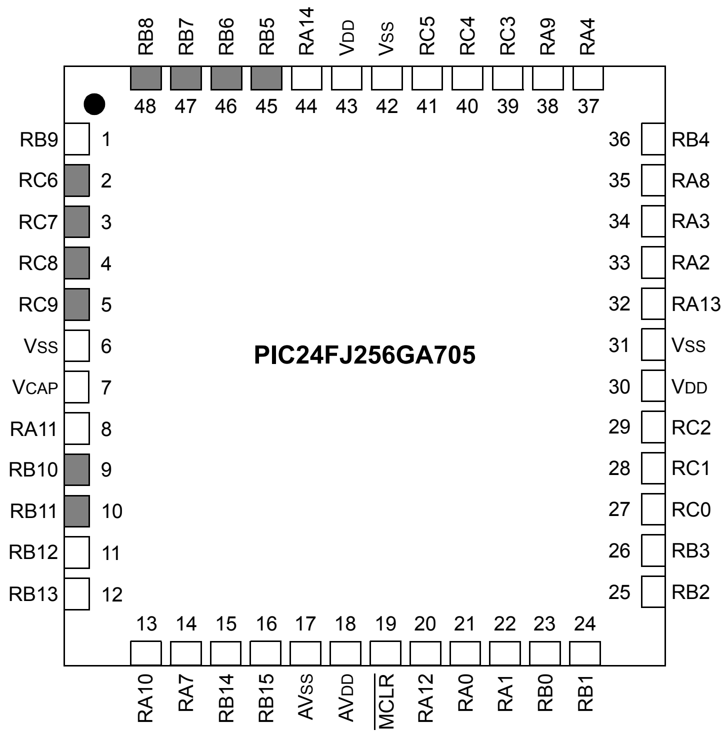
引脚	功能	引脚	功能
1	C1INC/C2INC/C3INC/$\overline{\text{TMPRN}}$/**RP9**/SDA1/T1CK/CTED4/PMD3/RB9	23	AN4/C1INB/**RP2**/SDA2/CTED13/RB2
2	**RP22**/PMA1/PMALH/RC6	24	AN5/C1INA/**RP3**/SCL2/CTED8/RB3
3	**RP23**/PMA0/PMALL/RC7	25	AN10/**RP16**/PMBE1/RC0
4	**RP24**/PMA5/RC8	26	AN11/**RP17**/PMA15/PMCS2/RC1
5	**RP25**/CTED7/PMA6/RC9	27	AN12/**RP18**/PMACK1/RC2
6	VSS	28	VDD
7	VCAP	29	VSS
8	PGD2/**RP10**/OCM1C/CTED11/PMD2/RB10	30	OSCI/CLKI/C1IND/RA2
9	PGC2/REFI1/**RP11**/CTED9/PMD1/RB11	31	OSCO/CLKO/C2IND/RA3
10	AN8/LVDIN/**RP12**/PMD0/RB12	32	TDO/PMA8/RA8
11	AN7/C1INC/**RP13**/OCM1D/CTPLS/PMRD/$\overline{\text{PMWR}}$/RB13	33	SOSCI/**RP4**/RB4
12	TMS/**RP28**/PMA2/PMALU/RA10	34	SOSCO/PWRLCLK/RA4
13	TCK/PMA7/RA7	35	TDI/PMA9/RA9
14	CVREF/AN6/C3INB/**RP14**/CTED5/PMWR/PMENB/RB14	36	AN13/**RP19**/PMBE0/RC3
15	AN9/C3INA/**RP15**/CTED6/PMA14/PMCS/PMCS1/RB15	37	**RP20**/PMA4/RC4
16	AVSS	38	**RP21**/PMA3/RC5
17	AVDD	39	VSS
18	$\overline{\text{MCLR}}$	40	VDD
19	VREF+/CVREF+/AN0/C3INC/**RP26**/CTED1/RA0	41	PGD3/**RP5**/ASDA1/OCM1E/PMD7/RB5
20	VREF-/CVREF-/AN1/C3IND/**RP27**/CTED2/RA1	42	PGC3/**RP6**/ASCL1/OCM1F/PMD6/RB6
21	PGD1/AN2/CTCMP/C2INB/**RP0**/RB0	43	**RP7**/OCM1A/CTED3/PMD5/INT0/RB7
22	PGC1/AN1-/AN3/C2INA/**RP1**/CTED12/RB1	44	**RP8**/SCL1/OCM1B/CTED10/PMD4/RB8

图注：　**RPn**表示用于外设引脚选择（PPS）功能的可重映射引脚。

注：　引脚排列可能变更。

引脚图（PIC24FJ256GA705器件）

48引脚UQFN



图注： 有关引脚功能的完整描述，请参见表4。引脚排列可能变更。

注： 灰色阴影表示可承受5.5V电压的输入引脚。

表4：　　引脚功能的完整描述（**PIC24FJ256GA705 UQFN**）

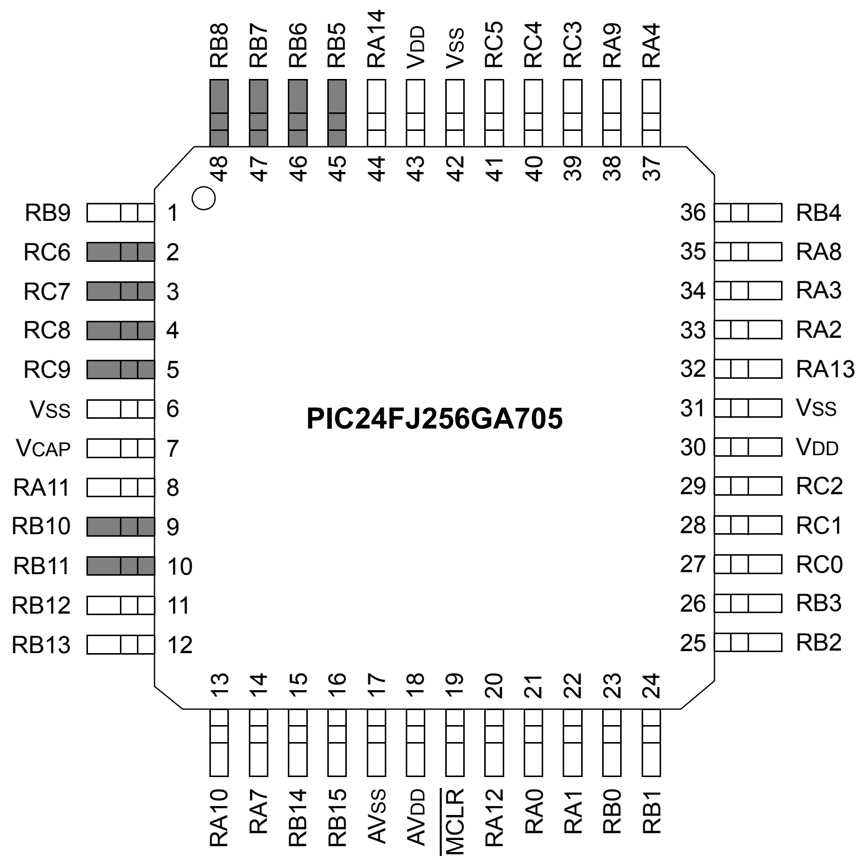
引脚	功能	引脚	功能
1	C1INC/C2INC/C3INC/$\overline{\text{TMPRN}}$/**RP9**/SDA1/T1CK/CTED4/PMD3/RB9	25	AN4/C1INB/**RP2**/SDA2/CTED13/RB2
2	**RP22**/PMA1/PMALH/RC6	26	AN5/C1INA/**RP3**/SCL2/CTED8/RB3
3	**RP23**/PMA0/PMALL/RC7	27	AN10/**RP16**/PMBE1/RC0
4	**RP24**/PMA5/RC8	28	AN11/**RP17**/PMA15/PMCS2/RC1
5	**RP25**/CTED7/PMA6/RC9	29	AN12/**RP18**/PMACK1/RC2
6	VSS	30	VDD
7	VCAP	31	VSS
8	**RPI29**/RA11	32	**RPI31**/RA13
9	PGD2/**RP10**/OCM1C/CTED11/PMD2/RB10	33	OSCI/CLKI/C1IND/RA2
10	PGC2/REFI1/**RP11**/CTED9/PMD1/RB11	34	OSCO/CLKO/C2IND/RA3
11	AN8/LVDIN/**RP12**/PMD0/RB12	35	TDO/PMA8/RA8
12	AN7/C1INC/**RP13**/OCM1D/CTPLS/PMRD/$\overline{\text{PMWR}}$/RB13	36	SOSCI/**RP4**/RB4
13	TMS/**RP28**/PMA2/PMALU/RA10	37	SOSCO/PWRLCLK/RA4
14	TCK/PMA7/RA7	38	TDI/PMA9/RA9
15	CVREF/AN6/C3INB/**RP14**/CTED5/PMWR/PMENB/RB14	39	AN13/**RP19**/PMBE0/RC3
16	AN9/C3INA/**RP15**/CTED6/PMA14/PMCS/PMCS1/RB15	40	**RP20**/PMA4/RC4
17	AVSS	41	**RP21**/PMA3/RC5
18	AVDD	42	VSS
19	$\overline{\text{MCLR}}$	43	VDD
20	**RPI30**/RA12	44	**RPI32**/RA14
21	VREF+/CVREF+/AN0/C3INC/**RP26**/CTED1/RA0	45	PGD3/**RP5**/ASDA1/OCM1E/PMD7/RB5
22	VREF-/CVREF-/AN1/C3IND/**RP27**/CTED2/RA1	46	PGC3/**RP6**/ASCL1/OCM1F/PMD6/RB6
23	PGD1/AN2/CTCMP/C2INB/**RP0**/RB0	47	**RP7**/OCM1A/CTED3/PMD5/INT0/RB7
24	PGC1/AN1-/AN3/C2INA/**RP1**/CTED12/RB1	48	**RP8**/SCL1/OCM1B/CTED10/PMD4/RB8

图注：　**RPn**和**RPIn**表示用于外设引脚选择（PPS）功能的可重映射引脚。

注：　引脚排列可能变更。

引脚图（PIC24FJ256GA705器件）

48 引脚 TQFP



图注： 有关引脚功能的完整描述，请参见表5。引脚排列可能变更。

注： 灰色阴影表示可承受5.5V电压的输入引脚。

表5：　　引脚功能的完整描述（**PIC24FJ256GA705 TQFP**）

引脚	功能	引脚	功能
1	C1INC/C2INC/C3INC/$\overline{\text{TMPRN}}$/**RP9**/SDA1/T1CK/CTED4/PMD3/RB9	25	AN4/C1INB/**RP2**/SDA2/CTED13/RB2
2	**RP22**/PMA1/PMALH/RC6	26	AN5/C1INA/**RP3**/SCL2/CTED8/RB3
3	**RP23**/PMA0/PMALL/RC7	27	AN10/**RP16**/PMBE1/RC0
4	**RP24**/PMA5/RC8	28	AN11/**RP17**/PMA15/PMCS2/RC1
5	**RP25**/CTED7/PMA6/RC9	29	AN12/**RP18**/PMACK1/RC2
6	VSS	30	VDD
7	VCAP	31	VSS
8	**RPI29**/RA11	32	**RPI31**/RA13
9	PGD2/**RP10**/OCM1C/CTED11/PMD2/RB10	33	OSCI/CLKI/C1IND/RA2
10	PGC2/REFI1/**RP11**/CTED9/PMD1/RB11	34	OSCO/CLKO/C2IND/RA3
11	AN8/LVDIN/**RP12**/PMD0//RB12	35	TDO/PMA8/RA8
12	AN7/C1INC/**RP13**/OCM1D/CTPLS/PMRD/$\overline{\text{PMWR}}$/RB13	36	SOSCI/**RP4**/RB4
13	TMS/**RP28**/PMA2/PMALU/RA10	37	SOSCO/PWRLCLK/RA4
14	TCK/PMA7/RA7	38	TDI/PMA9/RA9
15	CVREF/AN6/C3INB/**RP14**/CTED5/PMWR/PMENB/RB14	39	AN13/**RP19**/PMBE0/RC3
16	AN9/C3INA/**RP15**/CTED6/PMA14/PMCS/PMCS1/RB15	40	**RP20**/PMA4/RC4
17	AVSS	41	**RP21**/PMA3/RC5
18	AVDD	42	VSS
19	$\overline{\text{MCLR}}$	43	VDD
20	**RPI30**/RA12	44	**RPI32**/RA14
21	VREF+/CVREF+/AN0/C3INC/**RP26**/CTED1/RA0	45	PGD3/**RP5**/ASDA1/OCM1E/PMD7/RB5
22	VREF-/CVREF-/AN1/C3IND/**RP27**/CTED2/RA1	46	PGC3/**RP6**/ASCL1/OCM1F/PMD6/RB6
23	PGD1/AN2/CTCMP/C2INB/**RP0**/RB0	47	**RP7**/OCM1A/CTED3/PMD5/INT0/RB7
24	PGC1/AN1-/AN3/C2INA/**RP1**/CTED12/RB1	48	**RP8**/SCL1/OCM1B/CTED10/PMD4/RB8

图注：　**RPn**和**RPIn**表示用于外设引脚选择（PPS）功能的可重映射引脚。

注：　引脚排列可能变更。

目录

致 客 户

我们旨在提供最佳文档供客户正确使用 Microchip 产品。为此，我们将不断改进出版物的内容和质量，使之更好地满足您的需求。出版物的质量将随新文档及更新版本的推出而得到提升。

如果您对本出版物有任何问题和建议，请通过电子邮件联系我公司 TRC 经理，电子邮件地址为 **CTRC@microchip.com**。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，请访问我公司网站：

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。文献编号中紧跟数字串后的字母是版本号，例如：DS30000000A_CN 是文档的 A 版本。

勘误表

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

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

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

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

客户通知系统

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

参考资料

本器件数据手册基于《dsPIC33/PIC24 系列参考手册》的以下各个章节。这些文档应作为某个特定模块或器件特性的工作原理的一般参考。

> **注 1：** 要访问下列文档，请浏览Microchip网站（www.microchip.com）上 PIC24FJ256GA705 产品页面上的文档部分或从以下列表中选择一个器件系列参考手册章节。
>
> 除了参数、特性和其他文档，结果页面还提供相关系列参考手册章节的链接。

- **“带扩展数据空间（EDS）的CPU”**（DS39732）
- **“PIC24F数据存储器”**（DS30009717）
- **“直接存储器访问控制器（DMA）”**（DS39742）
- **“PIC24F闪存程序存储器”**（DS30009715）
- **“带扩展数据空间（EDS）的数据存储器”**（DS39733）
- **“复位”**（DS39712）
- **“中断”**（DS70000600）
- **“振荡器”**（DS39700）
- **“节能特性”**（DS39698）
- **“带外设引脚选择（PPS）功能的I/O端口”**（DS39711）
- **“定时器”**（DS39704）
- **“带专用定时器的输入捕捉”**（DS70000352）
- **“带专用定时器的输出比较”**（DS70005159）
- **“捕捉/比较/PWM/定时器（MCCP和SCCP）”**（DS33035）
- **“串行外设接口（SPI）”**（DS70005185）
- **“I^2C”**（DS70000195）
- **“UART”**（DS39708）
- **“增强型并行主端口（EPMP）”**（DS39730）
- **“带时间戳的RTCC”**（DS70005193）
- **“32位可编程循环冗余校验（CRC）”**（DS30009729）
- **“可配置逻辑单元（CLC）”**（DS33949）
- **“带阈值检测功能的12位A/D转换器”**（DS39739）
- **“可扩展的比较器模块”**（DS39734）
- **“双比较器模块”**（DS39710）
- **“带阈值检测功能的充电时间测量单元（CTMU）及CTMU操作”**（DS30009743）
- **“高度集成的可编程高/低电压检测（HLVD）模块”**（DS39725）
- **“看门狗定时器（WDT）”**（DS39697）
- **“CodeGuard™中级安全性”**（DS70005182）
- **“高级器件集成”**（DS39719）
- **“编程和诊断”**（DS39716）

注：

1.0 器件概述

本文档包含以下器件的具体信息：

- PIC24FJ64GA705
- PIC24FJ128GA705
- PIC24FJ256GA705
- PIC24FJ64GA704
- PIC24FJ128GA704
- PIC24FJ256GA704
- PIC24FJ64GA702
- PIC24FJ128GA702
- PIC24FJ256GA702

PIC24FJ256GA705系列在较小尺寸封装内引入了较大的闪存和SRAM存储器。该16位单片机系列具有丰富的外设功能和增强的计算性能。该系列还为某些高性能应用提供了新的移植选项，这些高性能应用可能超出了其8位平台的数字处理能力，但又不需要数字信号处理器（Digital Signal Processor，DSP）的数字处理能力。

表1-3列出了引脚图中显示的各引脚的功能。

1.1 内核特性

1.1.1 16位架构

所有PIC24F器件的内核都采用了16位改进型哈佛架构，该架构最初是在Microchip的dsPIC®数字信号控制器（Digital Signal Controller，DSC）中引入的。PIC24F CPU内核提供了大量增强功能，例如：

- 16位数据路径和24位地址路径，可在数据空间和程序空间之间传送信息
- 线性寻址能力最高可达12 MB（程序空间）和32 KB（数据空间）
- 带内建软件堆栈支持的16元工作寄存器阵列
- 支持整数算术运算的17位 x 17位硬件乘法器
- 支持32位/16位除法运算的硬件
- 支持多种寻址模式并针对高级语言（如C语言）进行优化的指令集
- 工作性能最高可达16 MIPS

1.1.2 节能技术

PIC24FJ256GA705系列器件包含保持休眠模式，这是一种由单独的低压稳压器为基本电路供电的低功耗模式。

这种新的低功耗模式还支持低功耗片上实时时钟/日历（Real-Time Clock/Calendar，RTCC）的连续工作，使应用可在器件处于休眠状态时计时。

除这一新特性外，PIC24FJ256GA705系列器件还包括以前PIC24F单片机的所有传统节能特性，例如：

- 动态时钟切换，允许在运行时选择低功耗时钟
- 打盹工作模式，在降低CPU时钟速度的同时保持外设时钟速度
- 基于指令的节能模式，快速调用空闲模式和休眠模式

1.1.3 振荡器选项和特性

PIC24FJ256GA705系列中的所有器件均提供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
- 外部时钟（External Clock，EC）模式
- 一个锁相环（Phase-Locked Loop，PLL）倍频器，最高可使处理器速度达到32 MHz
- 内部快速RC（Fast RC，FRC）振荡器（标称8 MHz输出），具有多个分频比选项
- 独立内部低功耗RC（Low-Power RC，LPRC）振荡器（31 kHz标称值），用于对时序要求不高的低功耗应用。

内部振荡器模块还为故障保护时钟监视器（Fail-Safe Clock Monitor，FSCM）提供了一个稳定的参考源。故障保护时钟监视器不断地监视主时钟源与内部振荡器提供的参考信号。一旦发生时钟故障，允许控制器将时钟源切换到内部振荡器，继续保持低速工作或安全地关闭应用。

1.1.4 易于移植

无论存储器容量如何，所有器件均共享同一组丰富外设，使得应用在扩展和升级时移植变得很方便。整个系列使用相同的引脚配置方案也有助于从一个器件向下一代更大型器件移植。

PIC24F系列器件的引脚同dsPIC33系列器件的引脚是兼容的，并与PIC18和dsPIC30器件的引脚配置方案部分兼容。这样仍然可以选用Microchip器件而将相对简单的应用移植到功能强大而复杂的应用。

1.2 DMA控制器

PIC24FJ256GA705系列器件具有一个直接存储器访问（Direct Memory Access，DMA）控制器。此模块的工作与CPU互不冲突，允许数据在数据存储器与外设之间传送而无需CPU干预，从而提高数据吞吐量并缩短执行时间。6条独立可编程通道几乎可同时为多个外设提供服务，每个通道外设都执行不同的操作。支持多种类型的数据传输操作。

1.3 其他特性

- **外设引脚选择：**外设引脚选择（PPS）功能允许大部分的数字外设通过一组固定的数字I/O引脚进行映射。用户可将许多数字外设之一的输入和/或输出独立地映射到其中的任一I/O引脚。
- **可配置逻辑单元：**可配置逻辑单元（CLC）模块允许用户将信号组合指定为逻辑功能的输入以及允许用户使用逻辑输出来控制其他外设或I/O引脚。
- **时序模块：**PIC24FJ256GA705系列提供3个独立的通用16位定时器（可将其中2个定时器组合为1个32位定时器）。该系列器件还包括4个多输出高级捕捉/比较/PWM/定时器外设以及3个独立传统输入捕捉模块和3个独立传统输出比较模块。
- **通信：**PIC24FJ256GA705 系列集成了一系列串行通信外设以满足各种应用需求，并且提供了两个均支持主从工作模式的独立I^2C模块。通过PPS功能，器件还具有2个包含内置IrDA编码器/解码器的独立UART以及3个SPI模块。
- **模拟特性：**PIC24FJ256GA705 系列的所有器件都包含一个12位A/D转换器（A/D）模块和一个三比较器模块。A/D模块包含一系列新功能，允许转换器对传入数据进行访问和判断，从而降低常规A/D转换的CPU开销。比较器模块包含3个模拟比较器，它们可针对多种操作进行配置。
- **CTMU接口：**除了其他模拟特性外，PIC24FJ256GA705系列器件还包含CTMU接口模块。此模块为精确时间测量和脉冲生成提供了一个便利的方法，并可用作容性传感器的接口。
- **增强型并行主/并行从端口：**此模块允许快速透明地访问单片机数据总线，并允许CPU直接对外部数据存储器进行寻址。并行端口可在主模式或从模式下工作，接受4或8位的数据宽度，可在主模式下进行最高10位寻址。
- **实时时钟和日历（Real-Time Clock and Calendar，RTCC）：**此模块通过硬件实现带有闹钟功能的全功能时钟和日历，从而释放了定时器资源和程序存储器空间供核心应用程序使用。

1.4 系列中各器件的详细说明

PIC24FJ256GA705系列器件具有28引脚、44引脚和48引脚三类封装形式。图1-1给出了所有器件的一般框图。

这些器件在以下5个方面存在差异：

1. 闪存程序存储器（PIC24FJ64GA70X器件为64 KB，PIC24FJ128GA70X器件为128 KB，PIC24FJ256GA70X器件为256 KB）。
2. 可用I/O引脚和端口（28引脚器件有2个端口22个引脚，44引脚/48引脚器件有3个端口36/40个引脚）。
3. 仅44引脚/48引脚器件上提供增强型并行主端口（Enhanced Parallel Master Port，EPMP）。
4. 模拟输入通道（28引脚器件有10通道，44引脚/48引脚器件有14通道）。
5. CTMU输入通道（28引脚器件有12通道，44引脚/48引脚器件有13通道）。

本系列器件的所有其他特性都是相同的。表1-1和表1-2对此做了总结。

表1-3给出了PIC24FJ256GA705系列器件上可用的引脚功能的列表，按功能排序。请注意，此表只显示了各个外设功能所使用的引脚的位置，而没有显示同一引脚上的多种功能的复用方式。在本数据手册开始部分的引脚图中提供了有关引脚复用的信息。复用的功能按功能的优先级排列，最前面的是优先级最高的外设功能。

表1-1： PIC24FJXXXGA702器件的特性：28引脚器件

特性	PIC24FJ64GA702	PIC24FJ128GA702	PIC24FJ256GA702
工作频率	DC – 32 MHz		
程序存储器（字节数）	64K	128K	256K
程序存储器（指令字数，24位）	22,528	45,056	88,064
数据存储器（字节数）	16K		
中断源（软向量数/NMI陷阱数）	124		
I/O端口	端口A和B		
I/O引脚总数	22		
可重映射的引脚数	18（18个I/O，0个输入）		
DMA	1个6通道		
16位定时器	3[1]		
实时时钟和日历（RTCC）	有		
循环冗余校验（CRC）	有		
输入捕捉通道	3[1]		
输出比较/PWM通道	3[1]		
输入电平变化通知中断	21（可重映射的引脚数）		
串行通信：			
UART	2[1]		
SPI（3线/4线）	3[1]		
I^2C	2		
可配置逻辑单元（CLC）	2[1]		
并行通信（EPMP/PSP）	无		
捕捉/比较/PWM/定时器模块	4个多输出MCCP 1个6输出MCCP，3个2输出MCCP		
JTAG边界扫描	有		
10/12位模数转换器（A/D）模块（输入通道）	10		
模拟比较器	3		
CTMU接口	有		
通用串行总线控制器	无		
复位（和延时）	内核POR、V_{DD} POR、BOR、RESET指令、 $\overline{\text{MCLR}}$、WDT、非法操作码、REPEAT指令、 硬件陷阱以及配置字不匹配（OST和PLL锁定）		
指令集	76条基本指令和多种寻址模式		
封装	28引脚QFN、UQFN、SOIC、SSOP和SPDIP		

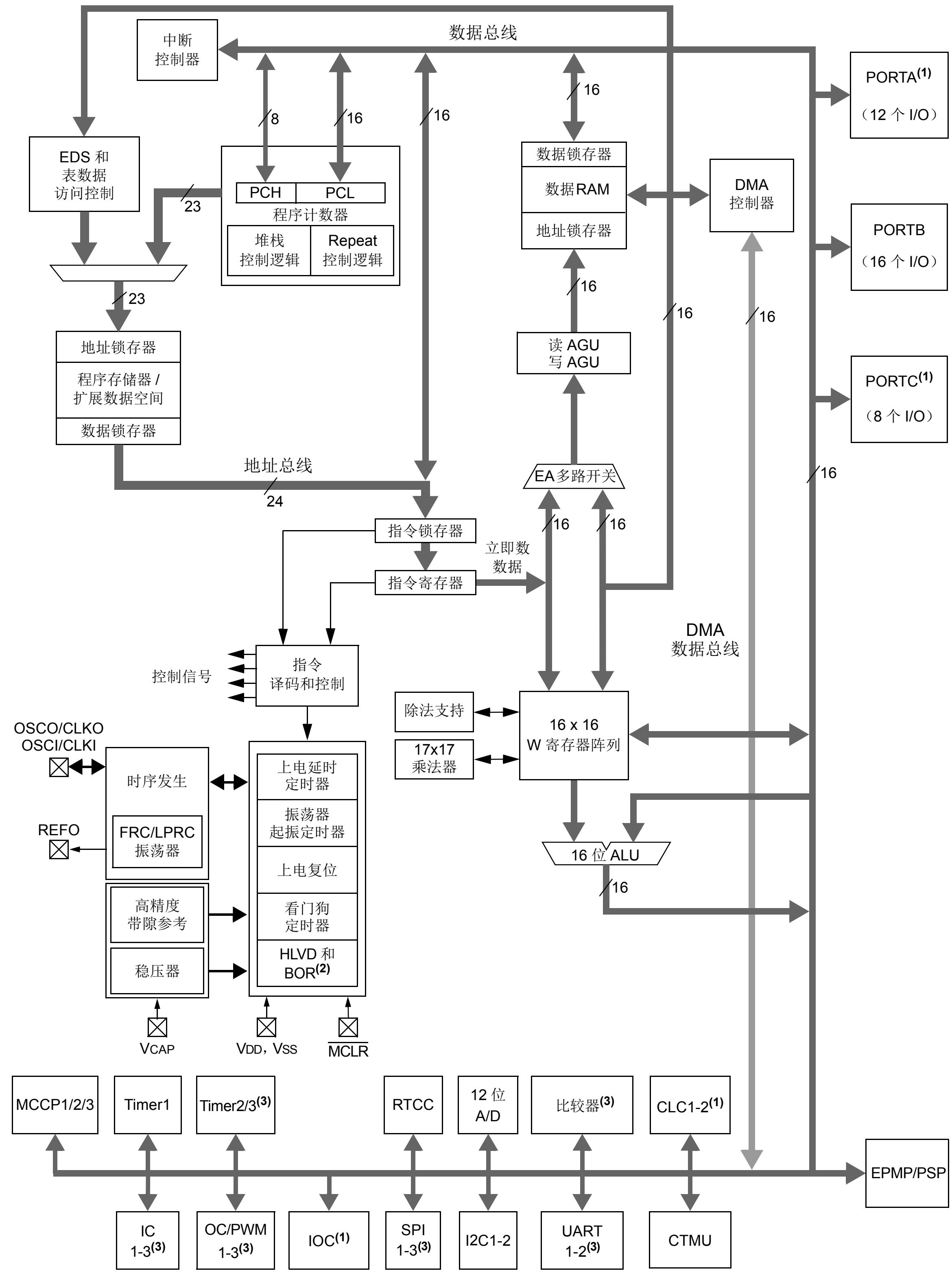
注 1： 一些外设可通过可重映射的引脚访问。

表1-2： PIC24FJXXXGA70X器件的特性：44引脚和48引脚器件

特性	PIC24FJ64GA70X	PIC24FJ128GA70X	PIC24FJ256GA70X
工作频率	DC – 32 MHz		
程序存储器（字节数）	64K	128K	256K
程序存储器（指令字数，24位）	22,528	45,056	88,064
数据存储器（字节数）	16K		
中断源（软向量数/NMI陷阱数）	124		
I/O端口	端口A、B和C		
I/O引脚总数：			
44引脚	35	35	35
48引脚	39	39	39
可重映射的引脚数：			
44引脚	29（29个I/O，0个输入）		
48引脚	33（29个I/O，仅4个输入）		
DMA（6通道）	1		
16位定时器	3(1)		
实时时钟和日历（RTCC）	有		
循环冗余校验（CRC）	有		
输入捕捉通道	3(1)		
输出比较/PWM通道	3(1)		
输入电平变化通知中断	25（可重映射的引脚数）		
串行通信：			
UART	2(1)		
SPI（3线/4线）	3(1)		
I^2C	2		
可配置逻辑单元（CLC）	2(1)		
并行通信（EPMP/PSP）	有		
捕捉/比较/PWM/定时器模块（MCCP）	4个模块 1个6输出MCCP，3个2输出MCCP		
JTAG边界扫描	有		
10/12位模数转换器（A/D）模块（输入通道）	14		
模拟比较器	3		
CTMU接口	有		
通用串行总线控制器	无		
复位（和延时）	内核POR、VDD POR、BOR、RESET指令、 $\overline{\text{MCLR}}$、WDT、非法操作码、REPEAT指令、 硬件陷阱以及配置字不匹配（OST和PLL锁定）		
指令集	76条基本指令和多种寻址模式		
封装	44引脚TQFP以及48引脚TQFP和QFN		

注 1： 一些外设可通过可重映射的引脚访问。

图1-1：　PIC24FJ256GA705系列一般框图



注 1： 不是所有I/O引脚或特性在所有器件引脚配置中都实现。请参见表1-3以了解按引脚数分类的具体实现。
2： 当使能片内稳压器时提供BOR功能。
3： 只能通过可重映射引脚访问这些外设I/O。

表1-3： PIC24FJ256GA705系列引脚说明

引脚功能	引脚编号/网格定位符 28引脚 SOIC、SSOP 和SPDIP	28引脚QFN 和UQFN	44引脚 TQFP	48引脚 QFN/TQFP	I/O	输入 缓冲器	说明
AN0	2	27	19	21	I	ANA	A/D模拟输入
AN1	3	28	20	22	I	ANA	
AN2	4	1	21	23	I	ANA	
AN3	5	2	22	24	I	ANA	
AN4	6	3	23	25	I	ANA	
AN5	7	4	24	26	I	ANA	
AN6	25	22	14	15	I	ANA	
AN7	24	21	11	12	I	ANA	
AN8	23	20	10	11	I	ANA	
AN9	26	23	15	16	I	ANA	
AN10	—	—	25	27	I	ANA	
AN11	—	—	26	28	I	ANA	
AN12	—	—	27	29	I	ANA	
AN13	—	—	36	39	I	ANA	
AVDD	28	25	17	18	P	—	模拟模块的正电源
AVSS	27	24	16	17	P	—	模拟模块的参考地
C1INA	7	4	24	26	I	ANA	比较器1的输入A
C1INB	6	3	23	25	I	ANA	比较器1的输入B
C1INC	18，24	15，21	1，11	1，12	I	ANA	比较器1的输入C
C1IND	9	6	30	33	I	ANA	比较器1的输入D
C2INA	5	2	22	24	I	ANA	比较器2的输入A
C2INB	4	1	21	23	I	ANA	比较器2的输入B
C2INC	18	15	1	1	I	ANA	比较器2的输入C
C2IND	10	7	31	34	I	ANA	比较器2的输入D
C3INA	26	23	15	16	I	ANA	比较器3的输入A
C3INB	25	22	14	15	I	ANA	比较器3的输入B
C3INC	2，18	15，27	1，19	1，21	I	ANA	比较器3的输入C
C3IND	3	28	20	22	I	ANA	比较器3的输入D
CLKI	9	6	30	33	—	—	主时钟输入连接
CLKO	10	7	31	34	O	DIG	系统时钟输出
CTCMP	4	1	21	23	O	ANA	CTMU比较器2输入（脉冲模式）

图注： TTL = TTL输入缓冲器 ST = 施密特触发器输入缓冲器
ANA = 模拟电压输入/输出 I^2C = I^2C/SMBus输入缓冲器
DIG = 数字输入/输出 XCVR = 专用收发器

表1-3： PIC24FJ256GA705系列引脚说明（续）

引脚功能	引脚编号/网格定位符				I/O	输入缓冲器	说明
	28引脚 SOIC、SSOP和SPDIP	28引脚QFN和UQFN	44引脚 TQFP	48引脚 QFN/TQFP			
CTED1	2	27	19	21	I	ST	CTMU外部边沿输入
CTED2	3	28	20	22	I	ST	
CTED3	16	13	43	47	I	ST	
CTED4	18	15	1	1	I	ST	
CTED5	25	22	14	15	I	ST	
CTED6	26	23	15	16	I	ST	
CTED7	—	—	5	5	I	ST	
CTED8	7	4	24	26	I	ST	
CTED9	22	19	9	10	I	ST	
CTED10	17	14	44	48	I	ST	
CTED11	21	18	8	9	I	ST	
CTED12	5	2	22	24	I	ST	
CTED13	6	3	23	25	I	ST	
CTPLS	24	21	11	12	O	DIG	CTMU脉冲输出
CVREF	25	22	14	15	O	ANA	比较器参考电压输出
CVREF+	2	27	19	21	I	ANA	比较器参考电压（高）输入
CVREF-	3	28	20	22	I	ANA	比较器参考电压（低）输入
INT0	16	13	43	47	I	ST	外部中断输入0
IOCA0	2	27	19	21	I	ST	PORTA电平变化中断
IOCA1	3	28	20	22	I	ST	
IOCA2	9	6	30	33	I	ST	
IOCA3	10	7	31	34	I	ST	
IOCA4	12	9	34	37	I	ST	
IOCA7	—	—	13	14	I	ST	
IOCA8	—	—	32	35	I	ST	
IOCA9	—	—	35	38	I	ST	
IOCA10	—	—	12	13	I	ST	
IOCA11	—	—	—	8	I	ST	
IOCA12	—	—	—	20	I	ST	
IOCA13	—	—	—	32	I	ST	
IOCA14	—	—	—	44	I	ST	

图注： TTL = TTL输入缓冲器　　ST = 施密特触发器输入缓冲器
ANA = 模拟电压输入/输出　　I^2C = I^2C/SMBus输入缓冲器
DIG = 数字输入/输出　　XCVR = 专用收发器

表1-3： PIC24FJ256GA705系列引脚说明（续）

引脚功能	引脚编号/网格定位符				I/O	输入缓冲器	说明
	28引脚 SOIC、SSOP和SPDIP	28引脚QFN和UQFN	44引脚 TQFP	48引脚 QFN/TQFP			
IOCB0	4	1	21	23	I	ST	PORTB电平变化中断
IOCB1	5	2	22	24	I	ST	
IOCB2	6	3	23	25	I	ST	
IOCB3	7	4	24	26	I	ST	
IOCB4	11	8	33	36	I	ST	
IOCB5	14	11	41	45	I	ST	
IOCB6	15	12	42	46	I	ST	
IOCB7	16	13	43	47	I	ST	
IOCB8	17	14	44	48	I	ST	
IOCB9	18	15	1	1	I	ST	
IOCB10	21	18	8	9	I	ST	
IOCB11	22	19	9	10	I	ST	
IOCB12	23	20	10	11	I	ST	
IOCB13	24	21	11	12	I	ST	
IOCB14	25	22	14	15	I	ST	
IOCB15	26	23	15	16	I	ST	
IOCC1	—	—	26	28	I	ST	PORTC电平变化中断
IOCC2	—	—	27	29	I	ST	
IOCC3	—	—	36	39	I	ST	
IOCC4	—	—	37	40	I	ST	
IOCC5	—	—	38	41	I	ST	
IOCC6	—	—	2	2	I	ST	
IOCC7	—	—	3	3	I	ST	
IOCC8	—	—	4	4	I	ST	
IOCC9	—	—	5	5	I	ST	
$\overline{\text{MCLR}}$	1	26	18	19	I	ST	主复位（器件复位）输入。将此引脚拉为低电平可导致器件复位。
OCM1A	16	13	43	47	O	DIG	MCCP1输出
OCM1B	17	14	44	48	O	DIG	
OCM1C	21	18	8	9	O	DIG	
OCM1D	24	21	11	12	O	DIG	
OCM1E	14	11	41	45	O	DIG	
OCM1F	15	12	42	46	O	DIG	
OSCI	9	6	30	33	I	ANA/ST	主振荡器输入连接
OSCO	10	7	31	34	O	ANA	主振荡器输出连接

图注： TTL = TTL输入缓冲器　　ST = 施密特触发器输入缓冲器
ANA = 模拟电压输入/输出　　I^2C = I^2C/SMBus输入缓冲器
DIG = 数字输入/输出　　XCVR = 专用收发器

表1-3： PIC24FJ256GA705系列引脚说明（续）

引脚功能	引脚编号/网格定位符				I/O	输入缓冲器	说明
	28引脚 SOIC、SSOP 和SPDIP	28引脚QFN 和UQFN	44引脚 TQFP	48引脚 QFN/TQFP			
PGC1	5	2	22	24	I	ST	ICSP™ 编程时钟
PGC2	22	19	9	10	I	ST	
PGC3	15	12	42	46	I	ST	
PGD1	4	1	21	23	I/O	DIG/ST	ICSP编程数据
PGD2	21	18	8	9	I/O	DIG/ST	
PGD3	14	11	41	45	I/O	DIG/ST	
PMA0	—	—	3	3	I/O	DIG/ST/TTL	并行主端口地址<0>/地址锁存器低选通端
PMA1	—	—	2	2	I/O	DIG/ST/TTL	并行主端口地址<1>/地址锁存器高选通端
PMA2	—	—	12	13	I/O	DIG/ST/TTL	并行主端口地址<2>
PMA3	—	—	38	41	I/O	DIG/ST/TTL	并行主端口地址<3>
PMA4	—	—	37	40	I/O	DIG/ST/TTL	并行主端口地址<4>
PMA5	—	—	4	4	I/O	DIG/ST/TTL	并行主端口地址<5>
PMA6	—	—	5	5	I/O	DIG/ST/TTL	并行主端口地址<6>
PMA7	—	—	13	14	I/O	DIG/ST/TTL	并行主端口地址<7>
PMA8	—	—	32	35	I/O	DIG/ST/TTL	并行主端口地址<8>
PMA9	—	—	35	38	I/O	DIG/ST/TTL	并行主端口地址<9>
PMA14/PMCS/PMCS1	—	—	15	16	I/O	DIG/ST/TTL	并行主端口地址<14>/从片选/片选1选通端

图注： TTL = TTL输入缓冲器　ST = 施密特触发器输入缓冲器
ANA = 模拟电压输入/输出　I^2C = I^2C/SMBus输入缓冲器
DIG = 数字输入/输出　XCVR = 专用收发器

表1-3： PIC24FJ256GA705系列引脚说明（续）

引脚功能	引脚编号/网格定位符				I/O	输入缓冲器	说明
	28引脚SOIC、SSOP和SPDIP	28引脚QFN和UQFN	44引脚TQFP	48引脚QFN/TQFP			
PMD0	—	—	10	11	I/O	DIG/ST/TTL	并行主端口数据（非复用主模式）或地址/数据（复用主模式）
PMD1	—	—	9	10	I/O	DIG/ST/TTL	
PMD2	—	—	8	9	I/O	DIG/ST/TTL	
PMD3	—	—	1	1	I/O	DIG/ST/TTL	
PMD4	—	—	44	48	I/O	DIG/ST/TTL	
PMD5	—	—	43	47	I/O	DIG/ST/TTL	
PMD6	—	—	42	46	I/O	DIG/ST/TTL	
PMD7	—	—	41	45	I/O	DIG/ST/TTL	
PMRD/$\overline{\text{PMWR}}$	—	—	11	12	I/O	DIG/ST/TTL	并行主端口读选通/写选通
PMWR/PMENB	—	—	14	15	I/O	DIG/ST/TTL	并行主端口写选通/使能选通
PWRGT	—	—	—	—	O	DIG	实时时钟电源控制输出
PWRLCLK	12	9	34	37	I	ST	实时时钟50/60 Hz时钟输入

图注： TTL = TTL输入缓冲器　　ST = 施密特触发器输入缓冲器
ANA = 模拟电压输入/输出　　I^2C = I^2C/SMBus输入缓冲器
DIG = 数字输入/输出　　XCVR = 专用收发器

表1-3： PIC24FJ256GA705系列引脚说明（续）

引脚功能	引脚编号/网格定位符 28引脚 SOIC、SSOP 和SPDIP	28引脚QFN 和UQFN	44引脚 TQFP	48引脚 QFN/TQFP	I/O	输入 缓冲器	说明
RA0	2	27	19	21	I/O	DIG/ST	PORTA数字I/O
RA1	3	28	20	22	I/O	DIG/ST	
RA2	9	6	30	33	I/O	DIG/ST	
RA3	10	7	31	34	I/O	DIG/ST	
RA4	12	9	34	37	I/O	DIG/ST	
RA7	—	—	13	14	I/O	DIG/ST	
RA8	—	—	32	35	I/O	DIG/ST	
RA9	—	—	35	38	I/O	DIG/ST	
RA10	—	—	12	13	I/O	DIG/ST	
RA11	—	—	—	8	I/O	DIG/ST	
RA12	—	—	—	20	I/O	DIG/ST	
RA13	—	—	—	32	I/O	DIG/ST	
RA14	—	—	—	44	I/O	DIG/ST	
RB0	4	1	21	23	I/O	DIG/ST	PORTB数字I/O
RB1	5	2	22	24	I/O	DIG/ST	
RB2	6	3	23	25	I/O	DIG/ST	
RB3	7	4	24	26	I/O	DIG/ST	
RB4	11	8	33	36	I/O	DIG/ST	
RB5	14	11	41	45	I/O	DIG/ST	
RB6	15	12	42	46	I/O	DIG/ST	
RB7	16	13	43	47	I/O	DIG/ST	
RB8	17	14	44	48	I/O	DIG/ST	
RB9	18	15	1	1	I/O	DIG/ST	
RB10	21	18	8	9	I/O	DIG/ST	
RB11	22	19	9	10	I/O	DIG/ST	
RB12	23	20	10	11	I/O	DIG/ST	
RB13	24	21	11	12	I/O	DIG/ST	
RB14	25	22	14	15	I/O	DIG/ST	
RB15	26	23	15	16	I/O	DIG/ST	
RC1	—	—	26	28	I/O	DIG/ST	PORTC数字I/O
RC2	—	—	27	29	I/O	DIG/ST	
RC3	—	—	36	39	I/O	DIG/ST	
RC4	—	—	37	40	I/O	DIG/ST	
RC5	—	—	38	41	I/O	DIG/ST	
RC6	—	—	2	2	I/O	DIG/ST	
RC7	—	—	3	3	I/O	DIG/ST	
RC8	—	—	4	4	I/O	DIG/ST	
RC9	—	—	5	5	I/O	DIG/ST	

图注： TTL = TTL输入缓冲器　　ST = 施密特触发器输入缓冲器
ANA = 模拟电压输入/输出　　I^2C = I^2C/SMBus输入缓冲器
DIG = 数字输入/输出　　XCVR = 专用收发器

表1-3：　PIC24FJ256GA705系列引脚说明（续）

引脚功能	引脚编号/网格定位符				I/O	输入缓冲器	说明
	28引脚SOIC、SSOP和SPDIP	28引脚QFN和UQFN	44引脚TQFP	48引脚QFN/TQFP			
RP0	4	1	21	23	I/O	DIG/ST	可重映射的外设（输入或输出）
RP1	5	2	22	24	I/O	DIG/ST	
RP2	6	3	23	25	I/O	DIG/ST	
RP3	7	4	24	26	I/O	DIG/ST	
RP4	11	8	33	36	I/O	DIG/ST	
RP5	14	11	41	45	I/O	DIG/ST	
RP6	15	12	42	46	I/O	DIG/ST	
RP7	16	13	43	47	I/O	DIG/ST	
RP8	17	14	44	48	I/O	DIG/ST	
RP9	18	15	1	1	I/O	DIG/ST	
RP10	21	18	8	9	I/O	DIG/ST	
RP11	22	19	9	10	I/O	DIG/ST	
RP12	23	20	10	11	I/O	DIG/ST	
RP13	24	21	11	12	I/O	DIG/ST	
RP14	25	22	14	15	I/O	DIG/ST	
RP15	26	23	15	16	I/O	DIG/ST	
RP16	—	—	25	27	I/O	DIG/ST	
RP17	—	—	26	28	I/O	DIG/ST	
RP18	—	—	27	29	I/O	DIG/ST	
RP19	—	—	36	39	I/O	DIG/ST	
RP20	—	—	37	40	I/O	DIG/ST	
RP21	—	—	38	41	I/O	DIG/ST	
RP22	—	—	2	2	I/O	DIG/ST	
RP23	—	—	3	3	I/O	DIG/ST	
RP24	—	—	4	4	I/O	DIG/ST	
RP25	—	—	5	5	I/O	DIG/ST	
RP26	2	27	19	21	I/O	DIG/ST	
RP27	3	28	20	22	I/O	DIG/ST	
RP28	—	—	12	13	I/O	DIG/ST	
RPI29	—	—	—	8	I	DIG/ST	可重映射的外设（仅输入）
RPI30	—	—	—	20	I	DIG/ST	
RPI31	—	—	—	32	I	DIG/ST	
RPI32	—	—	—	44	I	DIG/ST	

图注：　TTL = TTL输入缓冲器　　ST = 施密特触发器输入缓冲器
ANA = 模拟电压输入/输出　　I^2C = I^2C/SMBus输入缓冲器
DIG = 数字输入/输出　　XCVR = 专用收发器

表1-3： PIC24FJ256GA705系列引脚说明（续）

引脚功能	引脚编号/网格定位符				I/O	输入缓冲器	说明
	28引脚SOIC、SSOP和SPDIP	28引脚QFN和UQFN	44引脚TQFP	48引脚QFN/TQFP			
SCL1	17	14	44	48	I/O	I^2C	I2C1同步串行时钟输入/输出
SCL2	7	4	24	26	I/O	I^2C	I2C2同步串行时钟输入/输出
SDA1	18	15	1	1	I/O	I^2C	I2C1数据输入/输出
SDA2	6	3	23	25	I/O	I^2C	I2C2数据输入/输出
SOSCI	11	8	33	36	I	ANA/ST	辅助振荡器/Timer1时钟输入
SOSCO	12	9	34	37	O	ANA	辅助振荡器/Timer1时钟输出
T1CK	18	15	1	1	I	ST	Timer1时钟
TCK	17	14	13	14	I	ST	JTAG测试时钟/编程时钟输入
TDI	21	18	35	38	I	ST	JTAG测试数据/编程数据输入
TDO	18	15	32	35	O	DIG	JTAG测试数据输出
$\overline{\text{TMPRN}}$	18	15	1	1	I	ST	篡改检测输入
TMS	22	19	12	13	I	ST	JTAG测试模式选择输入
VCAP	20	17	7	7	P	—	外部滤波电容连接（已使能稳压器）
VDD	13，28	10，25	28，40	30，43	P	—	外设数字逻辑和I/O引脚的正电源
VREF+	2	27	19	21	I	ANA	比较器和A/D参考电压（高）输入
VREF-	3	28	20	22	I	ANA	比较器和A/D参考电压（低）输入
VSS	8，19，27	5，16，24	6，29，39	6，31，42	P	—	外设数字逻辑和I/O引脚的参考地

图注： TTL = TTL输入缓冲器　ST = 施密特触发器输入缓冲器
ANA = 模拟电压输入/输出　I^2C = I^2C/SMBus输入缓冲器
DIG = 数字输入/输出　XCVR = 专用收发器

注：

2.0 16位单片机入门指南

2.1 基本连接要求

在开始使用PIC24FJ256GA705系列16位单片机进行开发之前，需要注意最基本的器件引脚连接要求。

必须始终连接以下引脚：

- 所有VDD和VSS引脚（见**第2.2节“电源引脚”**）
- 所有AVDD和AVSS引脚，不论是否使用模拟器件功能（见**第2.2节“电源引脚”**）
- MCLR引脚（见**第2.3节“主复位（MCLR）引脚”**）
- VCAP引脚（见**第2.4节“稳压器引脚（VCAP）”**）

如果在最终应用中使用了以下引脚，则也必须连接它们：

- 用于在线串行编程（ICSP）和调试的PGCx/PGDx引脚（见**第2.5节“ICSP引脚”**）
- OSCI和OSCO引脚（使用外部振荡器源时）（见**第2.6节“外部振荡器引脚”**）

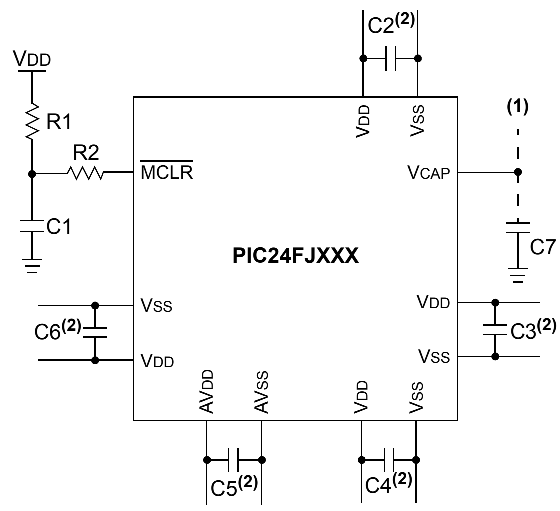
此外，可能还需要连接以下引脚：

- 实现模拟模块的外部参考电压时使用的VREF+/VREF-引脚

注：	不论是否使用模拟模块，都必须始终连接AVDD和AVSS引脚。

图2-1给出了最基本的连接要求。

图2-1： 建议的最基本连接



关键参数（所有值均为建议值）：

C1至C6：0.1 μF，50V，陶瓷电容

C7：10 μF，16V或以上，陶瓷电容

R1：10 kΩ

R2：100Ω到470Ω

注 1： 关于稳压器引脚连接的说明，请参见**第2.4节“稳压器引脚（VCAP）”**。

2： 此处所示示例针对的是具有5个VDD/VSS和AVDD/AVSS引脚对的PIC24F器件。其他器件的引脚对可能增多或减少，请相应地调整去耦电容的数量。

2.2 电源引脚

2.2.1 去耦电容

需要在每对电源引脚（例如，V_{DD}、V_{SS}、AV_{DD}和AV_{SS}）上使用去耦电容。

使用去耦电容时，需要考虑以下标准：

- **电容的类型和电容值：**建议使用参数为0.1 µF（100 nF）、25-50V的电容。应使用自谐振频率≥ 200 MHz的低ESR电容。建议使用陶瓷电容。
- **在印刷电路板上的放置：**去耦电容应尽可能靠近引脚。建议将电容与器件放置在电路板的同一层。如果空间受到限制，可以使用过孔将电容放置在PCB的另一层，但请确保从引脚到电容的走线长度不超出0.25英寸（6毫米）。
- **高频噪声处理：**如果电路板遇到高频噪声（频率高于数十MHz），则另外添加一个陶瓷电容，与上述去耦电容并联。第二个电容的电容值可以介于0.001 µF至0.01 µF之间。请将第二个电容放置在靠近每个主去耦电容的位置。在高速电路设计中，需要考虑尽可能靠近电源和接地引脚放置两个电容（例如，0.1 µF电容与0.001 µF电容并联）。
- **最大程度提高性能：**对于从电源电路开始的电路板布线，需要将电源和返回走线先连接到去耦电容，然后再与器件引脚连接。这可以确保去耦电容是电源链中的第一个元件。同等重要的是尽可能减小电容和电源引脚之间的走线长度，从而降低PCB走线电感。

2.2.2 大容量电容

在电源走线长度大于6英寸的电路板上，建议靠近MCU使用10 µF或更大的大容量电容。电容的电容值应根据连接电源与器件的走线电阻和应用中的器件的最大电流确定。典型值的范围为10 µF至47 µF。应使用额定电压为25V或更高值的陶瓷电容，以减小直流偏置的影响（见**第2.4.1节“陶瓷电容注意事项”**）。

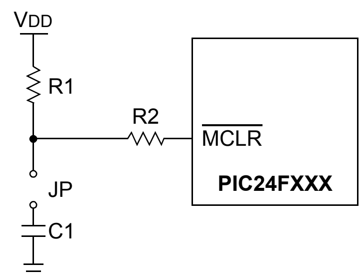
2.3 主复位（$\overline{\text{MCLR}}$）引脚

$\overline{\text{MCLR}}$引脚提供两种特定的器件功能：器件复位以及器件编程和调试。如果最终应用中不需要进行编程和调试，则只需直接连接到V_{DD}即可。添加其他元件有助于提高应用抵抗由于电压骤降导致意外复位的能力。图2-1给出了一种典型配置。根据应用的需求，还可以实现其他电路设计。

在编程和调试过程中，必须考虑到引脚上可能会增加的电阻和电容。器件编程器和调试器会驱动$\overline{\text{MCLR}}$引脚。因此，特定电压（V_{IH}和V_{IL}）和快速信号跳变一定不能受到不利影响。所以，需要根据应用和PCB需求来调整R1和C1的具体值。例如，在编程和调试操作期间，建议通过使用跳线将电容C1与$\overline{\text{MCLR}}$引脚隔离（图2-2）。对于正常的运行时操作，可以将跳线放回原处。

与$\overline{\text{MCLR}}$引脚关联的所有元件都应放置在距离该引脚0.25英寸（6毫米）的范围内。

图2-2： $\overline{\text{MCLR}}$引脚连接示例



注 1： 建议R1 ≤ 10 kΩ。建议的起始值为10 kΩ。请确保满足$\overline{\text{MCLR}}$引脚V_{IH}和V_{IL}规范。

2： R2 ≤ 470Ω用于限制由于静电放电（Electrostatic Discharge，ESD）或电过载（Electrical Overstress，EOS）导致$\overline{\text{MCLR}}$引脚损坏时从外部电容C流入$\overline{\text{MCLR}}$的任何电流。请确保满足$\overline{\text{MCLR}}$引脚V_{IH}和V_{IL}规范。

2.4 稳压器引脚（VCAP）

注：	本节仅适用于具有片上稳压器的PIC24FJ器件。

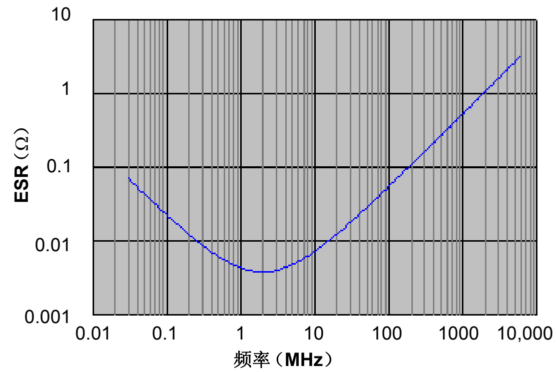
关于连接和使用片上稳压器的详细信息，请参见第29.3节“片上稳压器”。

VCAP引脚需要一个低ESR（< 5Ω）电容来稳定稳压器的输出电压。VCAP引脚一定不能与VDD连接，并且必须使用10 µF的电容接地。可以使用陶瓷电容或钽电容。表2-1列出了一些适用电容的示例。同等规格的电容都可以使用。

设计人员可以使用图2-3来评估备选电容的ESR。

此电容应靠近VCAP放置。建议走线长度不要超出0.25英寸（6毫米）。更多信息，请参见第32.0节“电气特性”。

图2-3： 推荐VCAP的频率—ESR性能曲线



注： 在25°C、0V直流偏置条件下测量的典型数据。

表2-1： 适用的等效电容（0805外壳尺寸）

制造商	部件编号	标称电容	基本容差	额定电压
TDK	C2012X5R1E106K085AC	10 µF	±10%	25V
TDK	C2012X5R1C106K085AC	10 µF	±10%	16V
Kemet	C0805C106M4PACTU	10 µF	±10%	16V
Murata	GRM21BR61E106KA3L	10 µF	±10%	25V
Murata	GRM21BR61C106KE15	10 µF	±10%	16V

2.4.1 陶瓷电容注意事项

近年来，大容值、低电压的表面贴装式陶瓷电容极具成本效益，其容值最高可达数十微法。低ESR、小外形尺寸及其他一些属性，使得陶瓷电容对于许多类型的应用颇具吸引力。

陶瓷电容适用于本系列单片机的内部稳压器。但在选择具体电容时，仍需注意确保电容在应用的目标工作条件范围内，始终能保持足够的电容值。

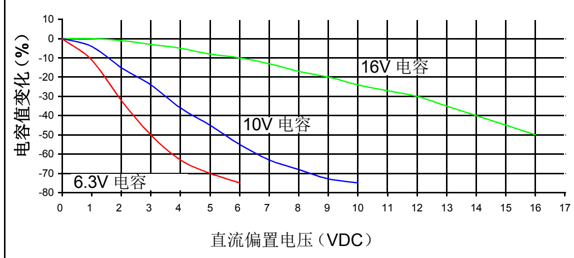
典型的低成本10 µF陶瓷电容有X5R、X7R和Y5V等介电特性（也存在其他型号，不过不太常见）。此类电容的初始公差通常指定为±10%至±20%（X5R和X7R），或-20%/+80%（Y5V）。然而，这些电容在应用中提供的有效电容值还会根据其他因素产生变化，比如施加的直流偏压和温度。因此，电容在电路中的总公差会比初始公差规范指定的范围宽很多。

X5R和X7R电容通常能表现良好的温度稳定性（例如，在很宽的温度范围内，电容值变化量在±15% 之间。具体的参数请参见制造商提供的数据手册）。但是，Y5V电容在温度变化时的电容公差通常很糟糕，为+22%/-82%。因此，标称值为10 µF 的Y5V型电容可能无法提供足够的总电容值来满足内部稳压器稳定性和瞬态响应的最低要求。所以，当应用必须在较宽温度范围内工作时，建议不要将Y5V 电容用于内部稳压器。

除了温度的影响，大容值陶瓷电容的有效电容值随着施加给电容的直流电压，会有很大的变化。直流偏压的效果非常明显，但却常常被人们忽视，或常常没有在文档中说明。

X7R型电容的典型直流偏压和电容的关系图见图2-4。

图2-4： 直流偏置电压—电容特性曲线



在选择要用于内部稳压器的陶瓷电容时，建议选择额定电压较高的电容，这样工作电压占电容最大额定电压的比例就比较小。例如对于1.8V内核电压，至少选择额定电压为16V的陶瓷电容。表2-1列出了推荐的电容。

2.5 ICSP引脚

PGCx和PGDx引脚用于进行在线串行编程（ICSP）和调试。建议尽可能减小ICSP连接器与器件ICSP引脚之间的走线长度。如果ICSP连接器会遇到ESD事件，则建议添加一个串联电阻，电阻值介于几十欧姆的范围，不要超出100Ω。

建议不要在PGCx和PGDx引脚上连接上拉电阻、串联二极管和电容，因为它们会影响编程器/调试器与器件之间的通信。如果应用需要此类分立元件，则在编程和调试期间应将它们从电路中去除。或者，请参见相应器件闪存编程规范中的交流/直流特性与时序要求信息，了解关于容性负载限制、引脚输入高电压（V_{IH}）和输入低电压（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”引脚（即，PGCx/PGDx）与ICSP 到Microchip 调试器/仿真器工具的物理连接一致。

有关可用的Microchip开发工具连接要求的详细信息，请参见**第30.0节“开发支持”**。

2.6 外部振荡器引脚

许多单片机都有至少两个振荡器可供选择：高频主振荡器和低频辅助振荡器（详细信息请参见第 9.0 节 “振荡器配置”）。

振荡器电路与器件应放置在电路板的同一层。请将振荡器电路放置在靠近相应振荡器引脚的位置，电路元件与引脚之间的距离不要超出 0.5 英寸（12 毫米）。负载电容应靠近振荡器本身，位于电路板的同一层。

请在振荡器电路周围使用接地灌铜区，以将其与周围电路隔离。接地灌铜区应与 MCU 地直接连接。不要在接地覆铜区内安排任何信号走线或电源走线。此外，如果使用双面电路板，请避免在电路板上晶振所在位置的背面有任何走线。

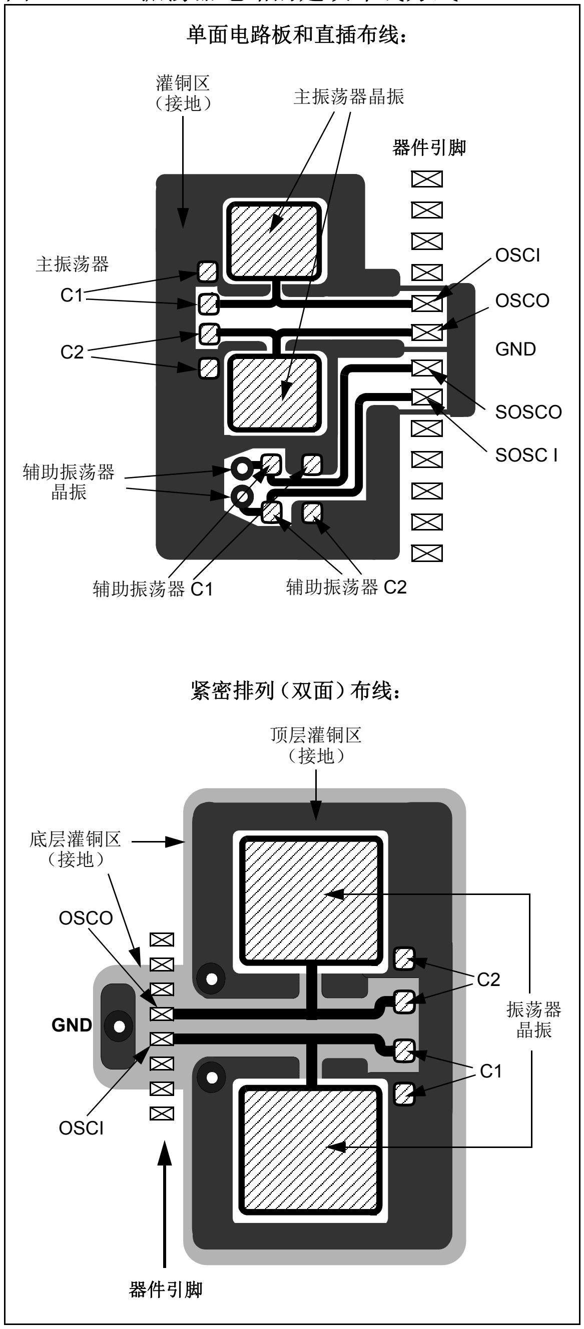
图 2-5 给出了一些布线建议。直插式封装可以采用完全包围振荡器引脚的单面布线来处理。对于引脚排列紧密的封装形式，单面布局则可能无法始终完全地包围所有引脚和元件。一种适合的解决方案是将各断开的地线连接到镜像接地层。在所有情形中，保护走线都必须回接到地。

在规划应用的走线和 I/O 分配时，需要确保相邻端口引脚和其他邻近振荡器的信号是无害的（即，无高频、无短暂上升和下降时间，以及无其他类似噪声）。

关于振荡器电路的其他信息和设计指南，请参见 Microchip 公司网站（www.microchip.com）上提供的以下应用笔记：

- *AN943，“Practical PICmicro® Oscillator Analysis and Design”*
- *AN949，“Making Your Oscillator Work”*
- *AN1798，“Crystal Selection for Low-Power Secondary Oscillator”*

图 2-5： 振荡器电路的建议布线方式



2.7 ICSP工作期间模拟和数字引脚配置

如果将符合ICSP的仿真器选择为调试器，它会自动初始化所有A/D输入引脚（ANx）为“数字”引脚，具体是通过将ANSx寄存器中的所有位清零来完成。更多具体信息，请参见**第11.2节“配置模拟端口引脚（ANSx）”**。

用户应用固件不得更改这些寄存器中与初始化仿真器的A/D引脚相对应的位；否则将导致调试器和器件之间发生通信错误。

如果在调试会话期间应用需要使用某些A/D引脚作为模拟输入引脚，那么用户应用必须在A/D模块初始化期间修改相应位，具体如下：

- 将要配置为模拟用途的引脚所对应的位置1。在任何时候都不要更改任何其他位，特别是与PGCx/PGDx引脚对对应的位。

当Microchip调试器/仿真器用作编程器时，用户应用固件必须正确配置ANSx寄存器。仅在调试器操作期间自动初始化这些寄存器。如果未能正确配置这些寄存器，将导致所有A/D引脚被识别为模拟输入引脚，以致端口值被读为逻辑0，从而可能影响用户应用的功能。

2.8 未用I/O

未用I/O引脚应配置为输出，并驱动为逻辑低电平状态。或者，将未用引脚通过一个1 kΩ至10 kΩ的电阻与Vss连接，并将输出驱动为逻辑低电平。

3.0 CPU

> **注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。有关CPU的更多信息，请参见《dsPIC33/PIC24系列参考手册》的**"具有扩展数据空间（EDS）的CPU"**（DS39732），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

PIC24F CPU采用16位（数据）的改进型哈佛架构，具有增强的指令集和带有长度可变的操作码字段的24位指令字。程序计数器（Program Counter，PC）为23位宽，可以寻址最大4M指令字的用户程序存储器空间。单周期指令预取机制可帮助维持吞吐量，并使指令的执行具有预测性。除了改变程序流的指令、双字传送（MOV.D）指令和表指令以外，所有指令都在单个周期内执行。使用REPEAT指令可以支持无开销的程序循环结构，REPEAT指令在任何时候都可被中断。

PIC24F器件的编程模型中有16个16位的工作寄存器。每个工作寄存器都可以充当数据、地址或地址偏移量寄存器。第16个工作寄存器（W15）用作软件堆栈指针（Software Stack Pointer，SSP），用于中断和调用。

可对数据空间（Data Space，DS）的低32 KB进行线性访问。数据空间的高32 KB是扩展数据RAM、EPMP存储空间或程序存储器可映射到的扩展数据空间（Extended Data Space，EDS）。

指令集架构（Instruction Set Architecture，ISA）与PIC18相比有了显著的提升，并保持了可接受的向后兼容性。该架构直接或通过简单的宏支持所有的PIC18指令和寻址模式。对编译器执行效率的需求促使了对ISA的许多改进。

内核支持固有（无操作数）寻址、相对寻址、立即数寻址和存储器直接寻址及三组寻址模式。所有模式都支持寄存器直接寻址和各种寄存器间接寻址模式。每组都提供了最多7种寻址模式。指令根据其功能要求，与预定义的寻址模式相关。

对于大多数指令，内核能在每个指令周期内执行一次数据（或程序数据）存储器读操作、一次工作寄存器（数据）读操作、一次数据存储器写操作和一次程序（指令）存储器读操作。因此，可以支持3个参数的指令，使3操作数的操作（如，A + B = C）能在单周期内执行。

包含了一个高速17位 x 17位乘法器，显著提高了内核的运算能力和吞吐量。此乘法器支持有符号、无符号和混合符号模式的16位 x 16位或8位 x 8位整数乘法。所有的乘法指令都在单周期内执行。

已对16位ALU进行了改进使其具备一个支持整数除法的硬件，该硬件支持迭代的不可恢复的除法算法。它可以和REPEAT指令循环机制和迭代除法指令（可选）一起工作，支持32位（或16位）除以16位有符号和无符号整数的除法运算。所有除法运算都需要19个周期来完成，但是可在任何周期边界被中断。

PIC24F具有向量异常机制，具有最多8个不可屏蔽的陷阱源和118个中断源。可以为每个中断源分配7个优先级之一。

CPU的框图如图3-1所示。

3.1 编程模型

图3-2中所示为PIC24F的编程模型。编程模型中的所有寄存器都是存储器映射的，并且可以由指令直接操作。

表3-1中提供了对每个寄存器的描述。所有与编程模型相关联的寄存器都是存储器映射的。

图3-1：　　PIC24F CPU内核框图

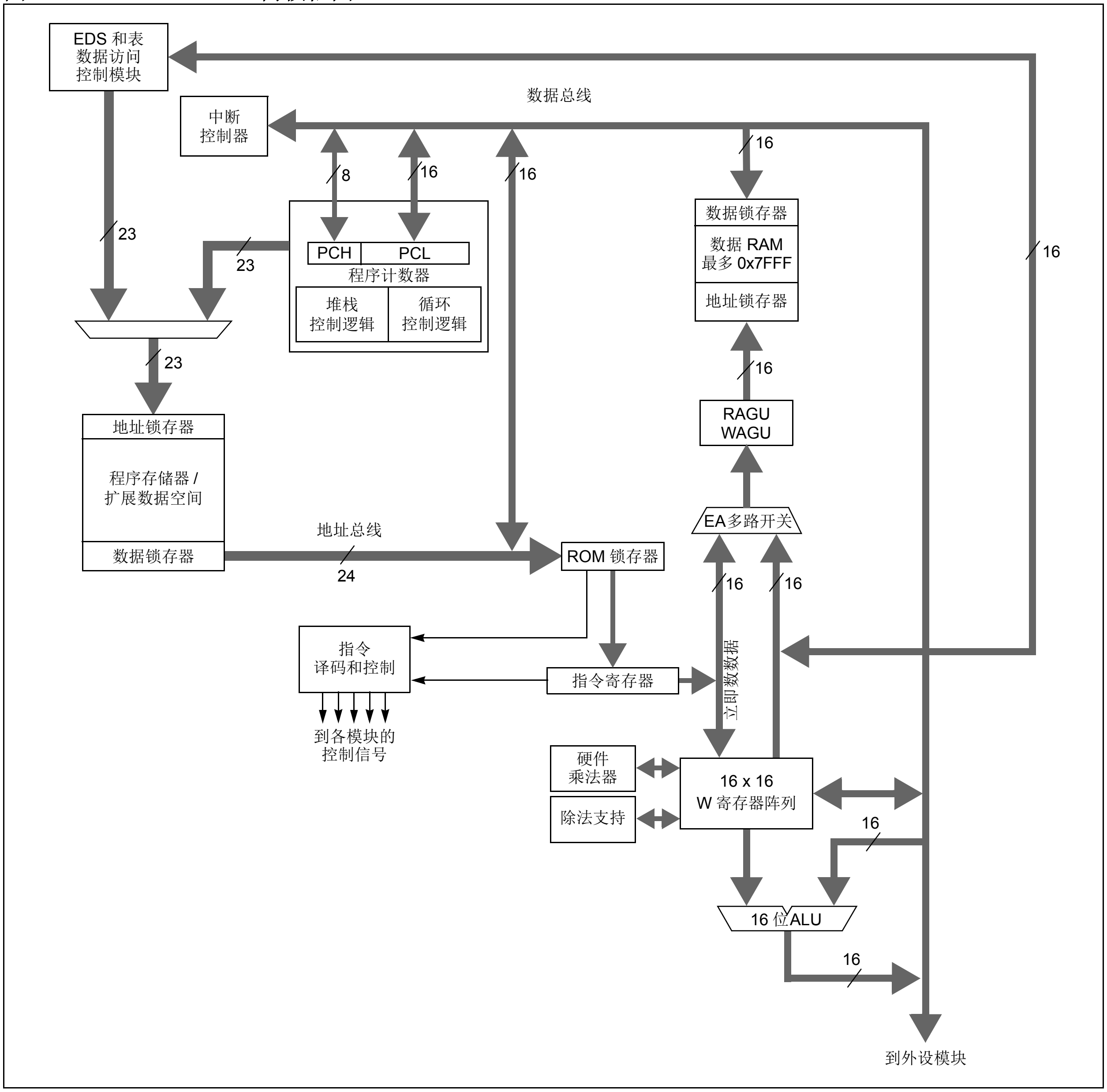
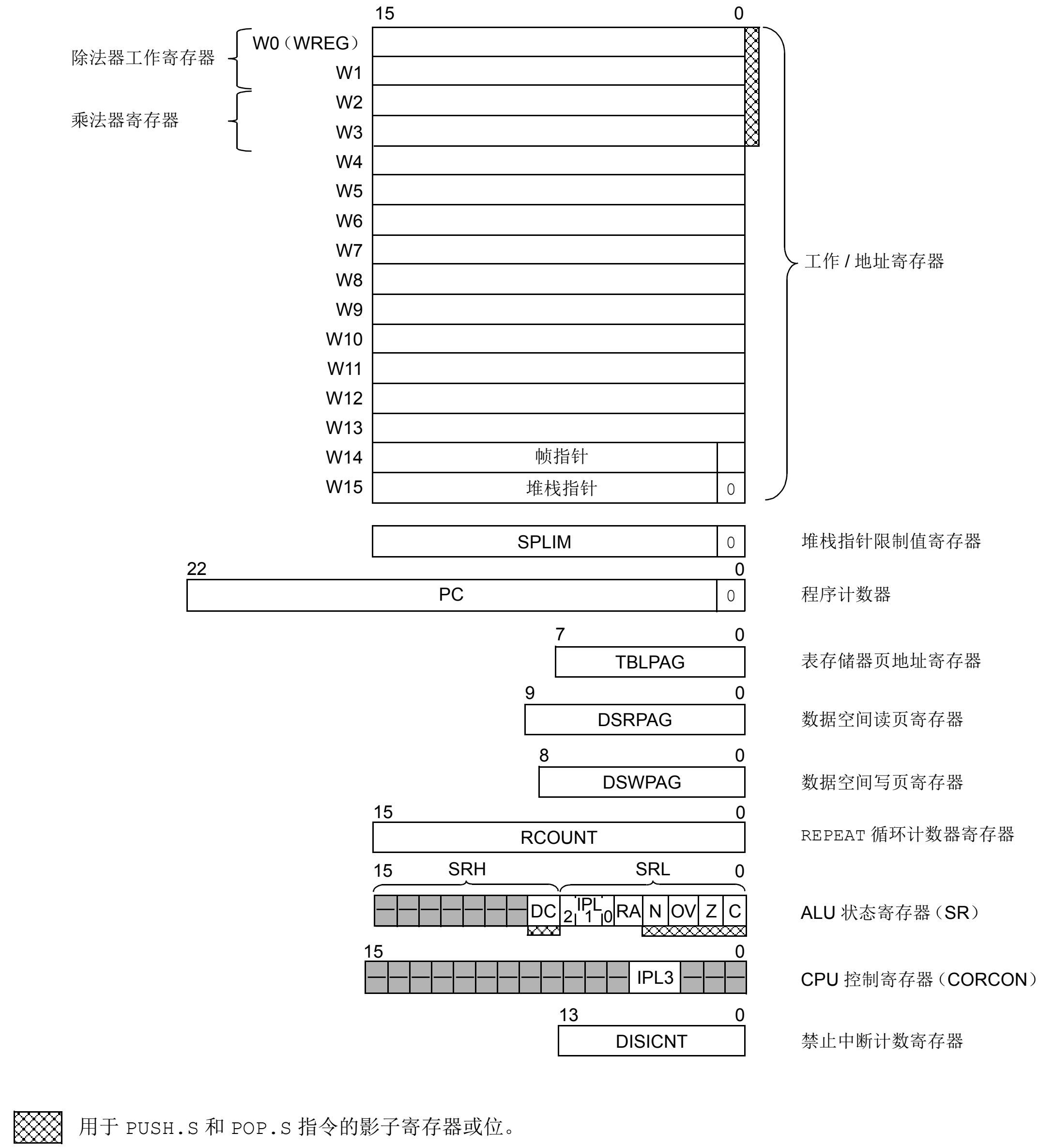


表3-1：　　CPU内核寄存器

寄存器名称	说明
W0到W15	工作寄存器阵列
PC	23位程序计数器
SR	ALU状态寄存器
SPLIM	堆栈指针限制值寄存器
TBLPAG	表存储器页地址寄存器
RCOUNT	REPEAT循环计数寄存器
CORCON	CPU控制寄存器
DISICNT	禁止中断计数寄存器
DSRPAG	数据空间读页寄存器
DSWPAG	数据空间写页寄存器

图3-2： 编程模型


15
0
除法器工作寄存器
W0（WREG）
W1
乘法器寄存器
W2
W3
W4
W5
W6
W7
W8
W9
W10
W11
W12
W13
W14
帧指针
W15
堆栈指针
0
工作/地址寄存器
SPLIM
0
堆栈指针限制值寄存器
22
0
PC
0
程序计数器
7
0
TBLPAG
表存储器页地址寄存器
9
0
DSRPAG
数据空间读页寄存器
8
0
DSWPAG
数据空间写页寄存器
15
0
RCOUNT
REPEAT 循环计数器寄存器
15
SRH
SRL
0
DC
IPL
2
1
0
RA
N
OV
Z
C
ALU 状态寄存器（SR）
15
0
IPL3
CPU 控制寄存器（CORCON）
13
0
DISICNT
禁止中断计数寄存器
用于 PUSH.S 和 POP.S 指令的影子寄存器或位。

3.2 CPU控制寄存器

寄存器3-1： SR：ALU状态寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	DC
bit 15							bit 8

R/W-0[1]	R/W-0[1]	R/W-0[1]	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2[2]	IPL1[2]	IPL0[2]	RA	N	OV	Z	C
bit 7							bit 0

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

bit 15-9 **未实现：**读为0

bit 8 **DC：**ALU半进位/借位标志位
1 = 结果的第4个低位（对于字节大小的数据）或第8个低位（对于字大小的数据）发生了向高位的进位
0 = 结果的第4个低位或第8个低位未发生向高位的进位

bit 7-5 **IPL<2:0>：**CPU中断优先级状态位[1,2]
111 = CPU中断优先级为7（15）；禁止用户中断
110 = CPU中断优先级为6（14）
101 = CPU中断优先级为5（13）
100 = CPU中断优先级为4（12）
011 = CPU中断优先级为3（11）
010 = CPU中断优先级为2（10）
001 = CPU中断优先级为1（9）
000 = CPU中断优先级为0（8）

bit 4 **RA：**REPEAT循环活动位
1 = 正在进行REPEAT循环
0 = 不在进行REPEAT循环

bit 3 **N：**ALU负标志位
1 = 结果为负
0 = 结果为非负（零或正值）

bit 2 **OV：**ALU溢出标志位
1 = 在本次算术运算中有符号运算（以二进制补码方式进行）发生了溢出
0 = 未发生溢出

bit 1 **Z：**ALU全零标志位
1 = 影响Z位的任何运算在过去某时已将该位置1
0 = 影响Z位的最近一次运算已经将该位清零（即运算结果非零）

bit 0 **C：**ALU进位/$\overline{\text{借位}}$标志位
1 = 结果的最高位（Most Significant bit，MSb） 发生了进位
0 = 结果的最高位未发生进位

注 1： 当NSTDIS（INTCON1<15>）= 1时，IPLx状态位只读。
2： IPLx状态位与IPL3状态位（CORCON<3>）组合形成CPU中断优先级（Interrupt Priority Level，IPL）。如果IPL3 = 1，则括号中的值表示IPL。

寄存器3-2： CORCON：CPU内核控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/C-0	R/W-1	U-0	U-0
—	—	—	—	IPL3(1)	PSV(2)	—	—
bit 7							bit 0

图注：	C = 可清零位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15-4　**未实现：**读为0

bit 3　**IPL3：**CPU中断优先级状态位(1)
　1 = CPU中断优先级大于7
　0 = CPU中断优先级小于或等于7

bit 2　**PSV：**数据空间中的程序空间可视性（Program Space Visibility，PSV）使能位(2)
　1 = 程序空间在数据空间中可视
　0 = 程序空间在数据空间中不可视

bit 1-0　**未实现：**读为0

注 **1：** IPL3位与IPL<2:0>位（SR<7:5>）共同决定CPU中断优先级；关于位描述，请参见寄存器3-1。

2： 如果PSV = 0，则对数据存储器的0x8000及更高地址的任何读操作都将导致地址陷阱错误，而不是读取程序存储器的PSV部分。该位不可单独寻址。

3.3 算术逻辑单元（ALU）

PIC24F ALU为16位宽，能进行加法、减法、移位和逻辑运算。除非另外声明，算术运算实际上是以二进制补码方式进行的。根据不同的运算，ALU可能会影响SR寄存器中的进位标志位（C）、全零标志位（Z）、负标志位（N）、溢出标志位（OV）和半进位标志位（DC）的值。在减法运算中，C和DC状态位分别作为借位位和半借位位。

根据所使用的指令模式，ALU可执行8位或16位运算。依据指令的寻址模式，ALU运算的数据可以来自W寄存器阵列或数据存储器。同样，ALU的输出数据可被写入W寄存器阵列或数据存储单元。

PIC24F CPU融入了对乘法和除法的硬件支持。这包括专门的硬件乘法器以及支持16位除数除法的硬件。

3.3.1 乘法器

ALU包含一个高速的17位x 17位乘法器。它支持几种乘法模式下的无符号、有符号或混合符号运算：

- 16位有符号 x 16位有符号
- 16位无符号 x 16位无符号
- 16位有符号 x 5位（立即数）无符号
- 16位无符号 x 16位无符号
- 16位无符号 x 5位（立即数）无符号
- 16位无符号 x 16位有符号
- 8位无符号 x 8位无符号

3.3.2 除法器

除法模块支持具有以下数据长度的32位/16位和16位/16位有符号和无符号整数除法运算：

1. 32位有符号/16位有符号
2. 32位无符号/16位无符号
3. 16位有符号/16位有符号
4. 16位无符号/16位无符号

所有除法指令的商都放在W0中，余数放在W1中。16位有符号和无符号DIV指令可为16位除数指定任一W寄存器（Wn），为32位被除数指定任意两个连续的W寄存器（W(m+1):Wm）。除法运算中处理除数的每一位需要一个周期，因此32位/16位和16位/16位指令的执行周期数相同。

3.3.3 多位移位支持

PIC24F ALU支持单位和单周期多位算术和逻辑移位操作。由一个移位寄存器实现多位移位，在单个周期内最多可将数据算术右移或左移15位。所有的多位移位指令仅支持操作数源寄存器和结果目标寄存器的寄存器直接寻址模式。

在下面的表3-2中汇总了所有使用移位操作的指令。

表3-2： 使用单位和多位移位操作的指令

指令	说明
ASR	将源寄存器算术右移一位或多位。
SL	将源寄存器左移一位或多位。
LSR	将源寄存器逻辑右移一位或多位。

4.0 存储器构成

注：	本数据手册总结了该PIC24F系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的**“PIC24F闪存程序存储器”**（DS30009715），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

作为哈佛架构器件，PIC24F单片机具有独立的程序和数据存储器空间以及总线。此架构还允许在代码执行过程中直接通过数据空间访问程序空间。

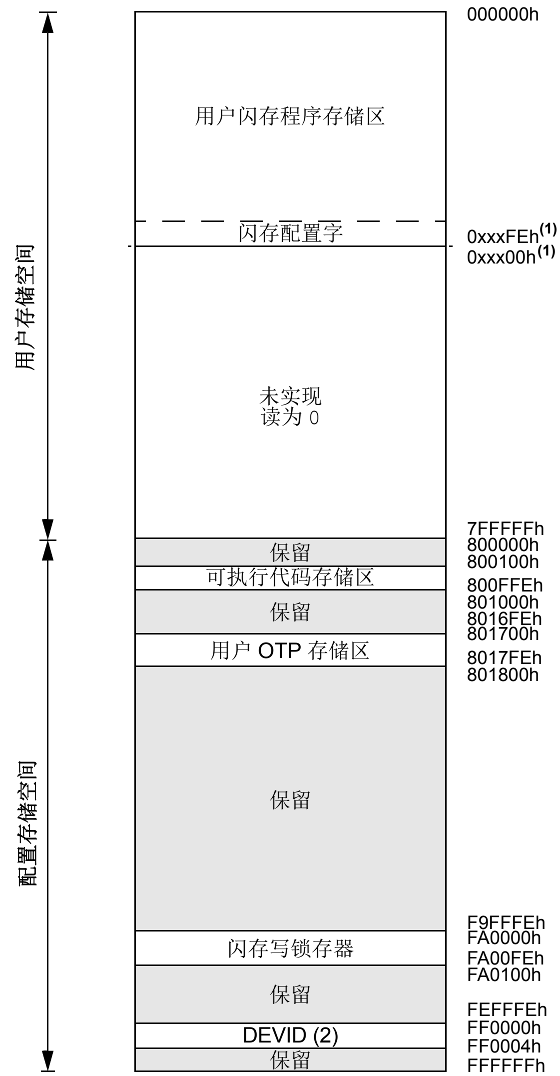
4.1 程序存储空间

PIC24FJ256GA705系列器件的程序地址存储空间可存储4M指令字。可通过由程序执行过程中的23位程序计数器（PC）或者**第4.3节“程序存储空间与数据存储空间的接口”**中所述的表操作或数据空间重映射得到的24位值寻址这一空间。

用户只能访问程序存储空间的低半地址部分（地址范围为000000h至7FFFFFh）。使用`TBLRD/TBLWT`指令时，情况会有所不同，这两条指令使用TBLPAG<7>以允许访问配置存储空间的配置位和用户OTP部分。

图4-1给出了PIC24FJ256GA705系列器件的存储器映射。

图 4-1： PIC24FJ256GA705 器件的程序存储器映射



图注： 存储区未按比例显示。

注 1： 精确的边界地址由已实现的程序存储器的大小决定（表 4-1）。

表 4-1： 程序存储器大小和边界[2]

器件	程序存储器上边界（指令字）	写入块[1]	擦除块[1]
PIC24FJ256GA70X	02AFFEh（88,064 x 24）	1376	172
PIC24FJ128GA70X	015FFEh（45,056 x 24）	704	88
PIC24FJ64GA70X	00AFFEh（22,528 x 24）	352	44

注 1： 1 个写入块 = 128 个指令字；1 个擦除块（页）= 1024 个指令字。
2： 为了保持整数页大小，存储器大小之间不是精确的二分之一关系。

4.1.1 程序存储器构成

程序存储空间由可字寻址的块构成。虽然它被视为24位宽，但将程序存储器的每个存储单元视作一个低位字和一个高位字的组合更加合理，其中高位字的高字节部分未实现。低位字的地址始终为偶数，而高位字的地址为奇数（图4-2）。

程序存储器地址始终在低位字处按字对齐，并且在代码执行过程中地址将递增或递减2。这种对齐方式与数据存储空间寻址兼容，且为访问程序存储空间中的数据提供了可能。

4.1.2 存储器硬编码向量

所有PIC24F器件中从000000h到000200h之间的地址空间都是保留的，用来存储硬编码的程序执行向量。提供了一个硬件复位向量将代码执行从器件复位时PC的默认值重定位到代码实际起始处。用户可在000000h地址编写一条GOTO指令以将代码的实际起始地址设置为000002h。

PIC24FJ256GA705器件最多可以有两个中断向量表（Interrupt Vector Table，IVT）。第一个表的地址范围为000004h至0000FFh。备用中断向量表（Alternate Interrupt Vector Table，AIVT）可通过AIVTDIS配置位使能，如果不存在引导段（Boot Segment，BS），则其地址范围为000104h至0001FFh。如果用户已配置了引导段（BS），则AIVT位于以下地址：(BSLIM<12:0> x 1024) – 508。这两个中断向量表允许使用不同的中断服务程序（Interrupt Service Routine，ISR）处理每个器件中断源。有关中断向量表的更详细讨论，请参见**第8.1节“中断向量表”**。

4.1.3 配置位概述

配置位存储在已实现的程序存储器的最后一页。可将这些位置1或清零以选择不同的器件配置。有两种配置位类型：系统操作位和代码保护位。系统操作位确定系统级元件的上电设置，如振荡器和看门狗定时器。代码保护位防止读写程序存储器。

表4-2列出了所有的配置寄存器及其配置寄存器位置。有关每个具体器件的完整配置寄存器描述，请参见**第29.0节“特殊功能”**。

表4-2： 配置字地址

配置寄存器	PIC24FJ256GA70X	PIC24FJ128GA70X	PIC24FJ64GA70X
FSEC	02AF00h	015F00h	00AF00h
FBSLIM	02AF10h	015F10h	00AF10h
FSIGN	02AF14h	015F14h	00AF14h
FOSCSEL	02AF18h	015F18h	00AF18h
FOSC	02AF1Ch	015F1Ch	00AF1Ch
FWDT	02AF20h	015F20h	00AF20h
FPOR	02AF24h	015F24h	00AF24h
FICD	02AF28h	015F28h	00AF28h
FDEVOPT1	02AF2Ch	015F2Ch	00AF2Ch

4.1.4 代码保护配置位

器件实现了由FSEC寄存器定义的中等安全功能。引导段（BS）权限较高，通用段（General Segment，GS）权限较低。总用户代码存储器可以分为BS或GS。段大小取决于BSLIM<12:0>位。用户空间内段的相对位置不会改变，因此BS（如果存在）占用中断向量表（IVT）后的存储区，GS占用BS后的空间（如果使能了备用IVT，则GS占用AIVT后的空间）。

配置段（Configuration Segment，CS）是用户闪存地址空间内的一个较小的段（小于一页，通常只有一行）。该段包含所有用户配置数据，在复位序列期间，NVM控制器会加载这些数据。

4.1.5 用户OTP存储器

PIC24FJ256GA705系列器件提供256字节的可一次性编程（One-Time-Programmable，OTP）存储器，其地址范围为801700h至8017FEh。该存储器可用于持久存储针对特定应用的信息，这些信息不会因对器件进行重新编程而被擦除。这包括许多类型的信息，例如（但不限于）：

- 应用程序校验和
- 代码版本信息
- 产品信息
- 序列号
- 系统制造日期
- 制造批号

用户OTP存储器可在包括用户RTSP模式在内的任何模式下进行编程，但不能擦除。全片擦除不会清除数据。

注：	请勿多次写入OTP存储器。多次写入OTP存储器可能导致永久性ECC双位错误（ECC Double-Bit Error，ECCDBE）陷阱。

4.2 数据存储空间

> **注：** 本数据手册总结了该 PIC24F 系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》的 **“具有扩展数据空间（EDS）的数据存储器”**（DS39733）。本数据手册中的信息取代了FRM中的信息。

PIC24F内核具有一个16位宽的数据存储空间，可将其作为一个线性空间寻址。使用两个地址发生单元（Address Generation Unit，AGU）对数据空间执行读写操作。图4-2给出了数据存储空间映射。

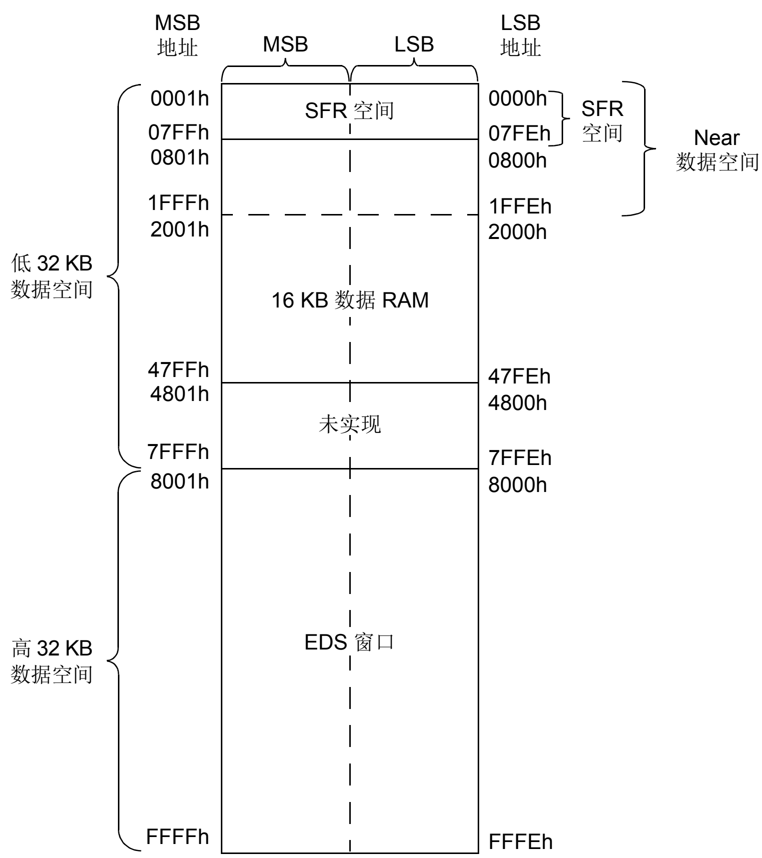
数据存储空间中的 16 位宽数据地址指向数据空间（DS）中的字节。这种构成方式使得数据空间的地址范围为16K字节或8K字。低半部分（0000h至7FFFh）供实现的（片上）存储器地址使用。

数据存储器地址空间的高半部分（8000h至FFFFh）用作通向扩展数据空间（EDS）的窗口。这使单片机能够直接访问比标准16位地址范围更大的数据范围。**第4.2.5节“扩展数据空间（EDS）”**详细介绍了EDS。

4.2.1 数据空间宽度

数据存储空间组织为可字节寻址的16位宽的块。在数据存储器和寄存器中的数据是以16位字为单位对齐的，但所有数据空间EA都解析为字节。每个字的低字节（Least Significant Byte，LSB）具有偶地址，而高字节（MSB）则具有奇地址。

图4-2： PIC24FJ256GA705器件的数据存储空间映射



注： 存储区未按比例显示。

4.2.2 数据存储器构成和对齐方式

为了保持与PIC® MCU的向后兼容和提高数据存储空间使用效率，PIC24F指令集既支持字操作，也支持字节操作。字节访问会在内部对按字对齐的存储空间的所有有效地址计算进行调整。例如，对于执行后修改寄存器间接寻址模式[Ws++]的结果，字节操作时，内核将其识别为值Ws + 1；而字操作时，内核将其识别为值Ws + 2。

数据字节读操作将读取包含字节的整个字，使用任何EA的LSB来确定要选取的字节。选中的字节将被放在数据总线的LSB处。也就是说，数据存储器和寄存器被组织为两个字节宽的并行实体，它们共享（字）地址译码，而写入线相互独立。数据字节写操作仅写入存储阵列和寄存器中与字节地址匹配的相应部分。

所有字访问必须按偶地址对齐。不支持不对齐的字数据取操作，因此当混合字节和字操作或从8位MCU代码移植时，必须要小心。如果试图进行不对齐的读或写操作，则会产生地址错误陷阱。如果在读操作时产生错误，正在执行的指令将完成；而如果在写操作时产生错误，指令仍将执行，但不会进行写入。无论是哪种情况，都会执行陷阱，从而系统和/或用户能够检查地址错误发生之前的机器状态。

所有装入任意W寄存器的字节都将装入W寄存器的低字节（LSB）。W寄存器的高字节（MSB）不变。

提供了一条符号扩展（SE）指令，允许用户将8位有符号数据转换为16位有符号值。或者，对于16位无符号数据，用户可以清零任何W寄存器的MSB，方法是在相应的地址处执行一条零扩展（ZE）指令。

尽管大多数指令能够对字或字节大小的数据进行操作，但需要注意的是，某些指令只对字大小的数据进行操作。

4.2.3 NEAR数据空间

0000h和1FFFh之间的8 KB的区域被称为Near数据空间。可以使用所有存储器直接寻址指令中的13位绝对地址字段直接寻址这一空间中的存储单元。剩余的数据空间通过间接寻址访问。此外，还可以使用MOV指令寻址整个数据空间，MOV指令支持使用16位地址字段的存储器直接寻址模式。

4.2.4 特殊功能寄存器（SFR）空间

Near数据空间的前2 KB（0000h至07FFh）主要被特殊功能寄存器（Special Function Register，SFR）占用。PIC24F的内核和外设模块使用这些寄存器来控制器件的工作。

SFR分布在受其控制的模块中，通常一个模块会使用一组SFR。大部分SFR空间包含未用的地址单元；它们读为0。表4-3给出了SFR空间的布局表，显示了实际实现SFR的位置。每个实现的区域代表一个32字节区域，其中至少有一个地址实现为一个SFR。表4-4至表4-11给出了所有实现的SFR及其地址的完整列表。

表4-3： SFR数据空间已实现区域[2]

	SFR空间地址															
	xx00	xx10	xx20	xx30	xx40	xx50	xx60	xx70	xx80	xx90	xxA0	xxB0	xxC0	xxD0	xxE0	xxF0
000h	内核															
100h	OSC	复位[1]	EPMP			CRC	REFO	PMD		定时器		—	CTMU	RTCC		
200h	捕捉		比较				MCCP								比较器	ANCFG
300h	MCCP				—	—	—	—	—	UART				—	—	— SPI
400h	SPI					—	CLC		—	—	I^2C			DMA		
500h	DMA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
600h	—	—	—	—	—	I/O										—
700h	—	A/D					NVM	—	—	PPS						

图注： — = 在该存储块中未实现任何SFR。

注 1： 包括HLVD控制。

2： 所显示的区域均为近似地址。关于精确地址，请参见表4-4至表4-11。

表4-4： SFR映射：0000h块

寄存器名称	地址	所有复位时的状态	寄存器名称	地址	所有复位时的状态
CPU内核			中断控制器（续）		
WREG0	0000	0000	IEC1	009A	0000
WREG1	0002	0000	IEC2	009C	0000
WREG2	0004	0000	IEC3	009E	0000
WREG3	0006	0000	IEC4	00A0	0000
WREG4	0008	0000	IEC5	00A2	0000
WREG5	000A	0000	IEC6	00A4	0000
WREG6	000C	0000	IEC7	00A6	0000
WREG7	000E	0000	IPC0	00A8	4444
WREG8	0010	0000	IPC1	00AA	4444
WREG9	0012	0000	IPC2	00AC	4444
WREG10	0014	0000	IPC3	00AE	4444
WREG11	0016	0000	IPC4	00B0	4444
WREG12	0018	0000	IPC5	00B2	4404
WREG13	001A	0000	IPC6	00B4	4444
WREG14	001C	0000	IPC7	00B6	4444
WREG15	001E	0800	IPC8	00B8	0044
SPLIM	0020	xxxx	IPC9	00BA	4444
PCL	002E	0000	IPC10	00BC	4444
PCH	0030	0000	IPC11	00BE	4444
DSRPAG	0032	0000	IPC12	00C0	4444
DSWPAG	0034	0000	IPC13	00C2	0440
RCOUNT	0036	xxxx	IPC14	00C4	4400
SR	0042	0000	IPC15	00C6	4444
CORCON	0044	0004	IPC16	00C8	4444
DISICNT	0052	xxxx	IPC17	00CA	4444
TBLPAG	0054	0000	IPC18	00CC	0044
中断控制器			IPC19	00CE	0040
INTCON1	0080	0000	IPC20	00D0	4440
INTCON2	0082	8000	IPC21	00D2	4444
INTCON4	0086	0000	IPC22	00D4	4444
IFS0	0088	0000	IPC23	00D6	4400
IFS1	008A	0000	IPC24	00D8	4444
IFS2	008C	0000	IPC25	00DA	0440
IFS3	008E	0000	IPC26	00DC	0400
IFS4	0090	0000	IPC27	00DE	4440
IFS5	0092	0000	IPC28	00E0	4444
IFS6	0094	0000	IPC29	00E2	0044
IFS7	0096	0000	INTTREG	00E4	0000
IEC0	0098	0000			

图注： x = 未定义。复位值以十六进制显示。

表4-5： SFR映射：0100h块

寄存器名称	地址	所有复位时的状态	寄存器名称	地址	所有复位时的状态
振荡器			**PMD**（续）		
OSCCON	0100	xxx0	PMD3	017C	0000
CLKDIV	0102	30x0	PMD4	017E	0000
OSCTUN	0108	0000	PMD5	0180	0000
OSCDIV	010C	0001	PMD6	0182	0000
OSCFDIV	010E	0000	PMD7	0184	0000
复位			PMD8	0186	0000
RCON	0110	0003	定时器		
HLVD			TMR1	0190	0000
HLVDCON	0114	0000	PR1	0192	FFFF
PMP			T1CON	0194	0000
PMCON1	0128	0000	TMR2	0196	0000
PMCON2	012A	0000	TMR3HLD	0198	0000
PMCON3	012C	0000	TMR3	019A	0000
PMCON4	012E	0000	PR2	019C	FFFF
PMCS1CF	0130	0000	PR3	019E	FFFF
PMCS1BS	0132	0000	T2CON	01A0	0x00
PMCS1MD	0134	0000	T3CON	01A2	0x00
PMCS2CF	0136	0000	**CTMU**		
PMCS2BS	0138	0000	CTMUCON1L	01C0	0000
PMCS2MD	013A	0000	CTMUCON1H	01C2	0000
PMDOUT1	013C	xxxx	CTMUCON2L	01C4	0000
PMDOUT2	013E	xxxx	实时时钟和日历（**RTCC**）		
PMDIN1	0140	xxxx	RTCCON1L	01CC	xxxx
PMDIN2	0142	xxxx	RTCCON1H	01CE	xxxx
PMSTAT	0144	008F	RTCCON2L	01D0	xxxx
CRC			RTCCON2H	01D2	xxxx
CRCCON1	0158	00x0	RTCCON3L	01D4	xxxx
CRCCON2	015A	0000	RTCSTATL	01D8	00xx
CRCXORL	015C	0000	TIMEL	01DC	xx00
CRCXORH	015E	0000	TIMEH	01DE	xxxx
CRCDATL	0160	xxxx	DATEL	01E0	xx0x
CRCDATH	0162	xxxx	DATEH	01E2	xxxx
CRCWDATL	0164	xxxx	ALMTIMEL	01E4	xx00
CRCWDATH	0166	xxxx	ALMTIMEH	01E6	xxxx
REFO			ALMDATEL	01E8	xx0x
REFOCONL	0168	0000	ALMDATEH	01EA	xxxx
REFOCONH	016A	0000	TSATIMEL	01EC	xx00
REFOTRIML	016C	0000	TSATIMEH	01EE	xxxx
PMD			TSADATEL	01F0	xx0x
PMD1	0178	0000	TSADATEH	01F2	xxxx
PMD2	017A	0000			

图注： x = 未定义。复位值以十六进制显示。

表4-6： SFR映射：0200h块

寄存器名称	地址	所有复位时的状态	寄存器名称	地址	所有复位时的状态
输入捕捉			**多输出捕捉/比较/PWM（续）**		
IC1CON1	0200	0000	CCP1RAH	0286	0000
IC1CON2	0202	000D	CCP1RBL	0288	0000
IC1BUF	0204	0000	CCP1RBH	028A	0000
IC1TMR	0206	0000	CCP1BUFL	028C	0000
IC2CON1	0208	0000	CCP1BUFH	028E	0000
IC2CON2	020A	000D	CCP2CON1L	0290	0000
IC2BUF	020C	0000	CCP2CON1H	0292	0000
IC2TMR	020E	0000	CCP2CON2L	0294	0000
IC3CON1	0210	0000	CCP2CON2H	0296	0100
IC3CON2	0212	000D	CCP2CON3L	0298	0000
IC3BUF	0214	0000	CCP2CON3H	029A	0000
IC3TMR	0216	0000	CCP2STATL	029C	00x0
输出比较			CCP2STATH	029E	0000
OC1CON1	0230	0000	CCP2TMRL	02A0	0000
OC1CON2	0232	000C	CCP2TMRH	02A2	0000
OC1RS	0234	xxxx	CCP2PRL	02A4	FFFF
OC1R	0236	xxxx	CCP2PRH	02A6	FFFF
OC1TMR	0238	xxxx	CCP2RAL	02A8	0000
OC2CON1	023A	0000	CCP2RAH	02AA	0000
OC2CON2	023C	000C	CCP2RBL	02AC	0000
OC2RS	023E	xxxx	CCP2RBH	02AE	0000
OC2R	0240	xxxx	CCP2BUFL	02B0	0000
OC2TMR	0242	xxxx	CCP2BUFH	02B2	0000
OC3CON1	0244	0000	CCP3CON1L	02B4	0000
OC3CON2	0246	000C	CCP3CON1H	02B6	0000
OC3RS	0248	xxxx	CCP3CON2L	02B8	0000
OC3R	024A	xxxx	CCP3CON2H	02BA	0100
OC3TMR	024C	xxxx	CCP3CON3L	02BC	0000
多输出捕捉/比较/PWM			CCP3CON3H	02BE	0000
CCP1CON1L	026C	0000	CCP3STATL	02C0	00x0
CCP1CON1H	026E	0000	CCP3STATH	02C2	0000
CCP1CON2L	0270	0000	CCP3TMRL	02C4	0000
CCP1CON2H	0272	0100	CCP3TMRH	02C6	0000
CCP1CON3L	0274	0000	CCP3PRL	02C8	FFFF
CCP1CON3H	0276	0000	CCP3PRH	02CA	FFFF
CCP1STATL	0278	00x0	CCP3RAL	02CC	0000
CCP1STATH	027A	0000	CCP3RAH	02CE	0000
CCP1TMRL	027C	0000	CCP3RBL	02D0	0000
CCP1TMRH	027E	0000	CCP3RBH	02D2	0000
CCP1PRL	0280	FFFF	CCP3BUFL	02D4	0000
CCP1PRH	0282	FFFF	CCP3BUFH	02D6	0000
CCP1RAL	0284	0000			

图注： x = 未定义。复位值以十六进制显示。

表4-6： SFR映射：0200h块（续）

寄存器名称	地址	所有复位时的状态	寄存器名称	地址	所有复位时的状态
比较器			比较器（续）		
CMSTAT	02E6	0000	CM3CON	02EE	0000
CVRCON	02E8	00xx	模拟配置		
CM1CON	02EA	0000	ANCFG	02F4	0000
CM2CON	02EC	0000			

图注： x = 未定义。复位值以十六进制显示。

表4-7： SFR映射：0300h块

寄存器名称	地址	所有复位时的状态	寄存器名称	地址	所有复位时的状态
多输出捕捉/比较/PWM			**UART**		
CCP4CON1L	0300	0000	U1MODE	0398	0000
CCP4CON1H	0302	0000	U1STA	039A	0110
CCP4CON2L	0304	0000	U1TXREG	039C	x0xx
CCP4CON2H	0306	0100	U1RXREG	039E	0000
CCP4CON3L	0308	0000	U1BRG	03A0	0000
CCP4CON3H	030A	0000	U1ADMD	03A2	0000
CCP4STATL	030C	00x0	U2MODE	03AE	0000
CCP4STATH	030E	0000	U2STA	03B0	0110
CCP4TMRL	0310	0000	U2TXREG	03B2	xxxx
CCP4TMRH	0312	0000	U2RXREG	03B4	0000
CCP4PRL	0314	FFFF	U2BRG	03B6	0000
CCP4PRH	0316	FFFF	U2ADMD	03B8	0000
CCP4RAL	0318	0000	**SPI**		
CCP4RAH	031A	0000	SPI1CON1L	03F4	0x00
CCP4RBL	031C	0000	SPI1CON1H	03F6	0000
CCP4RBH	031E	0000	SPI1CON2L	03F8	0000
CCP4BUFL	0320	0000	SPI1STATL	03FC	0028
CCP4BUFH	0322	0000	SPI1CON2H	03F8	0000
			SPI1STATH	03FE	0000

图注： x = 未定义。复位值以十六进制显示。

表4-8： SFR映射：0400h块

寄存器名称	地址	所有复位时的状态	寄存器名称	地址	所有复位时的状态
SPI（续）			**I²C**（续）	0498	0000
SPI1BUFL	0400	0000	I2C1BRG	0498	0000
SPI1BUFH	0402	0000	I2C1CONL	049A	1000
SPI1BRGL	0404	xxxx	I2C1CONH	049C	0000
SPI1IMSKL	0408	0000	I2C1STAT	049E	0000
SPI1IMSKH	040A	0000	I2C1ADD	04A0	0000
SPI1URDTL	040C	0000	I2C1MSK	04A2	0000
SPI1URDTH	040E	0000	I2C2RCV	04A4	0000
SPI2CON1L	0410	0x00	I2C2TRN	04A6	00FF
SPI2CON1H	0412	0000	I2C2BRG	04A8	0000
SPI2CON2L	0414	0000	I2C2CONL	04AA	1000
SPI2STATL	0418	0028	I2C2CONH	04AC	0000
SPI2STATH	041A	0000	I2C2STAT	04AE	0000
SPI2BUFL	041C	0000	I2C2ADD	04B0	0000
SPI2BUFH	041E	0000	I2C2MSK	04B2	0000
SPI2BRGL	0420	xxxx	**DMA**		
SPI2IMSKL	0424	0000	DMACON	04C4	0000
SPI2IMSKH	0426	0000	DMABUF	04C6	0000
SPI2URDTL	0428	0000	DMAL	04C8	0000
SPI2URDTH	042A	0000	DMAH	04CA	0000
SPI3CON1L	042C	0x00	DMACH0	04CC	0000
SPI3CON1H	042E	0000	DMAINT0	04CE	0000
SPI3CON2L	0430	0000	DMASRC0	04D0	0000
SPI3STATL	0434	0028	DMADST0	04D2	0000
SPI3STATH	0436	0000	DMACNT0	04D4	0001
SPI3BUFL	0438	0000	DMACH1	04D6	0000
SPI3BUFH	043A	0000	DMAINT1	04D8	0000
SPI3BRGL	043C	xxxx	DMASRC1	04DA	0000
SPI3IMSKL	0440	0000	DMADST1	04DC	0000
SPI3IMSKH	0442	0000	DMACNT1	04DE	0001
SPI3URDTL	0444	0000	DMACH2	04E0	0000
SPI3URDTH	0446	0000	DMAINT2	04E2	0000
可配置逻辑单元（**CLC**）			DMASRC2	04E4	0000
CLC1CONL	0464	0000	DMADST2	04E6	0000
CLC1CONH	0466	0000	DMACNT2	04E8	0001
CLC1SEL	0468	0000	DMACH3	04EA	0000
CLC1GLSL	046C	0000	DMAINT3	04EC	0000
CLC1GLSH	046E	0000	DMASRC3	04EE	0000
CLC2CONL	0470	0000	DMADST3	04F0	0000
CLC2CONH	0472	0000	DMACNT3	04F2	0001
CLC2SEL	0474	0000	DMACH4	04F4	0000
CLC2GLSL	0478	0000	DMAINT4	04F6	0000
CLC2GLSH	047A	0000	DMASRC4	04F8	0000
I²C			DMADST4	04FA	0000
I2C1RCV	0494	0000	DMACNT4	04FC	0001
I2C1TRN	0496	00FF	DMACH5	04FE	0000

图注： x = 未定义。复位值以十六进制显示。

表4-9：　SFR映射：0500h块

寄存器名称	地址	所有复位时的状态	寄存器名称	地址	所有复位时的状态
DMA（续）			**DMA**（续）		
DMAINT5	0500	0000	DMADST5	0504	0000
DMASRC5	0502	0000	DMACNT5	0506	0001

图注：　x = 未定义。复位值以十六进制显示。

表4-10：　SFR映射：0600h块

寄存器名称	地址	所有复位时的状态	寄存器名称	地址	所有复位时的状态
I/O			**PORTB**（续）		
PADCON	065E	0000	ANSB	067E	FFFF
IOCSTAT	0660	0000	IOCPB	0680	0000
PORTA			IOCNB	0682	0000
TRISA	0662	FFFF	IOCFB	0684	0000
PORTA	0664	0000	IOCPUB	0686	0000
LATA	0666	0000	IOCPDB	0688	0000
ODCA	0668	0000	**PORTC**		
ANSA	066A	FFFF	TRISC	068A	FFFF
IOCPA	066C	0000	PORTC	068C	0000
IOCNA	066E	0000	LATC	068E	0000
IOCFA	0670	0000	ODCC	0690	0000
IOCPUA	0672	0000	ANSC	0692	FFFF
IOCPDA	0674	0000	IOCPC	0694	0000
PORTB			IOCNC	0696	0000
TRISB	0676	FFFF	IOCFC	0698	0000
PORTB	0678	0000	IOCPUC	069A	0000
LATB	067A	0000	IOCPDC	069C	0000
ODCB	067C	0000			

图注：　x = 未定义。复位值以十六进制显示。

表4-11： SFR映射：0700h块

寄存器名称	地址	所有复位时的状态	寄存器名称	地址	所有复位时的状态
A/D			外设引脚选择		
ADC1BUF0	0712	xxxx	RPINR0	0790	3F3F
ADC1BUF1	0714	xxxx	RPINR1	0792	3F3F
ADC1BUF2	0716	xxxx	RPINR2	0794	3F3F
ADC1BUF3	0718	xxxx	RPINR3	0796	3F3F
ADC1BUF4	071A	xxxx	RPINR5	079A	3F3F
ADC1BUF5	071C	xxxx	RPINR6	079C	3F3F
ADC1BUF6	071E	xxxx	RPINR7	079E	3F3F
ADC1BUF7	0720	xxxx	RPINR8	07A0	003F
ADC1BUF8	0722	xxxx	RPINR11	07A6	3F3F
ADC1BUF9	0724	xxxx	RPINR12	07A8	3F3F
ADC1BUF10	0726	xxxx	RPINR18	07B4	3F3F
ADC1BUF11	0728	xxxx	RPINR19	07B6	3F3F
ADC1BUF12	072A	xxxx	RPINR20	07B8	3F3F
ADC1BUF13	072C	xxxx	RPINR21	07BA	3F3F
ADC1BUF14	072E	xxxx	RPINR22	07BC	3F3F
ADC1BUF15	0730	xxxx	RPINR23	07BE	3F3F
AD1CON1	0746	xxxx	RPINR25	07C2	3F3F
AD1CON2	0748	xxxx	RPINR28	07C8	3F3F
AD1CON3	074A	xxxx	RPINR29	07CA	003F
AD1CHS	074C	xxxx	RPOR0	07D4	0000
AD1CSSH	074E	xxxx	RPOR1	07D6	0000
AD1CSSL	0750	xxxx	RPOR2	07D8	0000
AD1CON4	0752	xxxx	RPOR3	07DA	0000
AD1CON5	0754	xxxx	RPOR4	07DC	0000
AD1CHITL	0758	xxxx	RPOR5	07DE	0000
AD1CTMENH	075A	0000	RPOR6	07E0	0000
AD1CTMENL	075C	0000	RPOR7	07E2	0000
AD1RESDMA	075E	0000	RPOR8	07E4	0000
NVM			RPOR9	07E6	0000
NVMCON	0760	0000	RPOR10	07E8	0000
NVMADR	0762	xxxx	RPOR11	07EA	0000
NVMADRU	0764	00xx	RPOR12	07EC	0000
NVMKEY	0766	0000	RPOR13	07EE	0000
			RPOR14	07F0	0000

图注： x = 未定义。复位值以十六进制显示。

4.2.5 扩展数据空间（EDS）

扩展数据空间（EDS）使PIC24F器件能够对更大的数据范围进行寻址，否则只可对16位地址范围进行寻址。EDS包括低32 KB数据地址空间不可直接访问的任何其他内部数据存储器空间和任何通过EPMP访问的外部存储空间。

此外，EDS还允许对程序存储器空间进行读访问。此功能称为程序空间可视性（PSV），第**4.3.3**节**“使用EDS从程序存储器读数据”**中有详细介绍。

图4-3显示了整个EDS空间。EDS由页组成，称为EDS页，单页大小等于EDS窗口的大小（32 KB）。通过数据空间读页寄存器（DSRPAG）或数据空间写页寄存器（DSWPAG）选择特定EDS页。对于PSV，只使用DSRPAG寄存器。DSRPAG寄存器值和16位宽数据地址组合形成了24位有效地址（Effective Address，EA）。

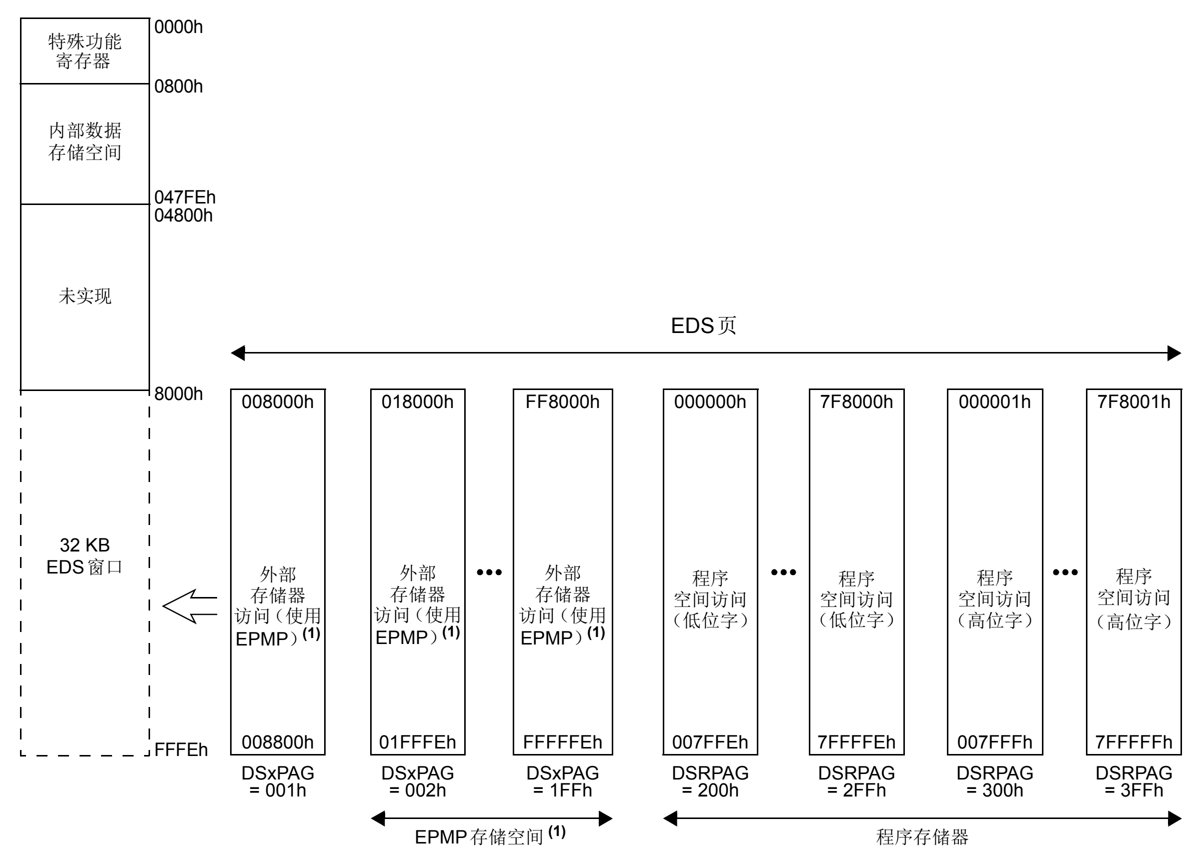
PIC24FJ256GA705系列器件的数据寻址范围取决于特定器件上实现的增强型并行主端口版本；同时这也与器件引脚数有关。表4-12列出了此系列中各器件可访问的总存储空间。有关使用EPMP访问外部存储空间的更多详细信息，请参见《dsPIC33/PIC24系列参考手册》的**“增强型并行主端口（EPMP）”**（DS39730）。

表4-12： 可访问的总数据存储器空间

系列	内部RAM	外部RAM访问（使用EPMP）
PIC24FJXXXGA70X	16K	1K

注：	访问EDS窗口的Page 0将生成地址错误陷阱，因为Page 0是基础数据存储器（低数据空间中的数据存储单元0800h至7FFFh）。

图4-3： 扩展数据空间



注 1： 可寻址的存储器范围取决于器件引脚数和EPMP实现。

4.2.5.1 从EDS读取数据

为了从EDS空间读取数据，首先，通过将所需的EDS页号装入到DSRPAG寄存器并将偏移地址分配给其中一个W寄存器来设置地址指针。一旦完成上述分配任务，通过将分配了偏移地址的工作寄存器的bit 15置1使能EDS窗口；然后，将读取所指EDS存储单元的内容。

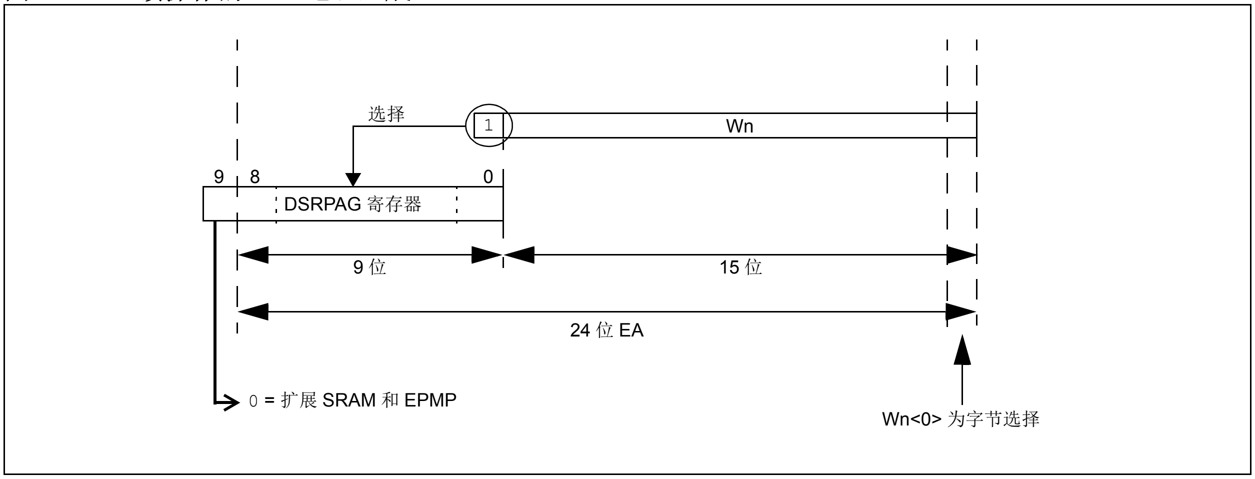
图4-4描述了为读操作生成EDS空间地址的方式。

当EA的最高有效位（MSb）为1且DSRPAG<9> = 0时，DSRPAG的低9位与EA的低15位共同形成一个24位EDS空间地址以进行读操作。

例4-1显示了如何从EDS读取一个字节、字和双字。

注：	从EDS空间的所有读操作均有一个指令周期的额外开销。因此，完成一个EDS读操作至少需要两个指令周期。在REPEAT指令中进行EDS读操作，前两次访问每次需要三个周期而后续的访问每次需要一个周期。

图4-4： 读操作的EDS地址生成



例4-1： EDS读代码（汇编语言）

```
; Set the EDS page from where the data to be read
    mov     #0x0002, w0
    mov     w0, DSRPAG      ;page 2 is selected for read
    mov     #0x0800, w1     ;select the location (0x800) to be read
    bset    w1, #15         ;set the MSB of the base address, enable EDS mode

;Read a byte from the selected location
    mov.b   [w1++], w2      ;read Low byte
    mov.b   [w1++], w3      ;read High byte

;Read a word from the selected location
    mov     [w1], w2        ;

;Read Double - word from the selected location
    mov.d   [w1], w2        ;two word read, stored in w2 and w3
```

4.2.5.2 数据写入EDS

要将数据写入EDS空间，与EDS读操作类似，通过将所需的EDS页号装入到DSWPAG寄存器并将偏移地址分配给其中一个W寄存器来设置地址指针。一旦完成上述分配任务，通过将分配了偏移地址的工作寄存器的bit 15置1使能EDS窗口，从而写入所访问的存储单元。

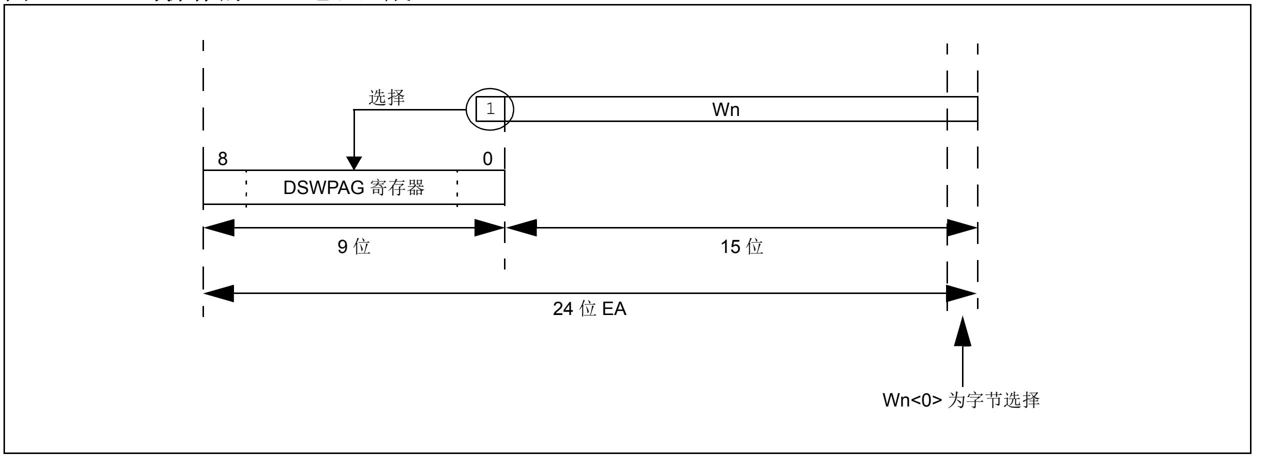
图4-5描述了如何为写操作生成EDS地址。

当EA的MSb为1时，DSWPAG的低9位与EA的低15位共同组成一个24位EDS地址以进行写操作。例4-2显示了如何将一个字节、字或双字写入EDS。

数据空间页寄存器（DSRPAG/DSWPAG）在跨页边界（发生从0xFFFF至0x8000的计满返回）时不会自动更新。用汇编语言开发代码时，在地址指针跨页边界时更新数据空间页寄存器必须谨慎。“C”编译器会跟踪寻址，并在访问连续数据存储单元时相应递增或递减页寄存器值。

> **注 1：** 所有对EDS的写操作均在一个单周期内执行。
>
> **2：** 不支持在 REPEAT 指令中对任何 EDS 存储单元执行读/修改/写操作。例如：BCLR、BSW、BTG、RLC f、RLNC f、RRC f、RRNC f、ADD f、SUB f、SUBR f、AND f、IOR f、XOR f、ASR f和ASL f。
>
> **3：** 在执行读/修改/写操作时，请使用DSRPAG寄存器。

图4-5： 写操作的EDS地址生成



例4-2： EDS写代码（汇编语言）

```
; Set the EDS page where the data to be written
    mov     #0x0002, w0
    mov     w0, DSWPAG      ;page 2 is selected for write
    mov     #0x0800, w1     ;select the location (0x800) to be written
    bset    w1, #15         ;set the MSB of the base address, enable EDS mode

;Write a byte to the selected location
    mov     #0x00A5, w2
    mov     #0x003C, w3
    mov.b   w2, [w1++]      ;write Low byte
    mov.b   w3, [w1++]      ;write High byte

;Write a word to the selected location
    mov     #0x1234, w2     ;
    mov     w2, [w1]        ;

;Write a Double - word to the selected location
    mov     #0x1122, w2
    mov     #0x4455, w3
    mov.d   w2, [w1]        ;2 EDS writes
```

表4-13： 具有不同页和地址的EDS存储地址

DSRPAG（数据空间读寄存器）	DSWPAG（数据空间写寄存器）	间接寻址时的源/目标地址	指向EDS的24位EA	备注
x[1]	x[1]	0000h至1FFFh	000000h至001FFFh	Near数据空间[2]
		2000h至7FFFh	002000h至007FFFh	
001h	001h	8000h至FFFFh	008000h至00FFFEh	EPMP存储空间
002h	002h		010000h至017FFEh	
003h • • • • • 1FFh	003h • • • • • 1FFh		018000h至0187FEh • • • • FF8000h至FFFFFEh	
000h	000h		无效地址	地址错误陷阱[3]

注 1： 如果源/目标地址低于8000h，将忽略DSRPAG和DSWPAG寄存器。
2： 该数据空间也可通过直接寻址访问。
3： 当源/目标地址高于8000h且DSRPAG/DSWPAG为0时，将发生地址错误陷阱。

4.2.6 软件堆栈

除了用作工作寄存器外，PIC24F器件中的W15寄存器也可用作软件堆栈指针（Software Stack Pointer，SSP）。此指针总是指向第一个可用的空字，从低地址向高地址方向增长。它在弹出堆栈之前递减，而在压入堆栈后递增，如图4-6所示。注意，对于执行任何CALL指令期间的PC压栈操作，在压入堆栈之前，PC的MSB要进行零扩展，从而确保MSB始终清零。

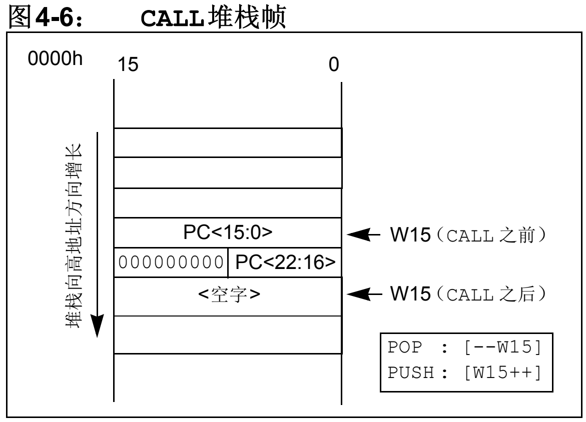
注：	在异常处理期间，在PC压入堆栈之前，要先将PC的MSB与SRL寄存器组合在一起。

堆栈指针限制值寄存器（SPLIM）与堆栈指针相关联，它设置堆栈上边界的地址。SPLIM在复位时不会被初始化。与堆栈指针一样，SPLIM<0>被强制为0，因为所有的堆栈操作必须是字对齐的。每当使用W15作为源指针或目标指针产生有效地址时，产生的有效地址会与SPLIM中的值进行比较。如果堆栈指针（W15）的内容与SPLIM寄存器的内容相等，则会执行压栈操作而不产生堆栈错误陷阱。但在随后的压栈操作时将会产生堆栈错误陷阱。因此，例如如果想要在堆栈增长到超过RAM中的地址2000h时产生堆栈错误陷阱，则需用值1FFEh来初始化SPLIM。

同理，当堆栈指针地址小于0800h时，将产生堆栈指针下溢（堆栈错误）陷阱。这可防止堆栈进入特殊功能寄存器空间。

在对SPLIM寄存器进行写操作后，不应紧跟使用W15进行间接读操作的指令。

图4-6： CALL堆栈帧



4.3 程序存储空间与数据存储空间的接口

PIC24F架构采用24位宽的程序空间和16位宽的数据空间。该架构也是一种改进型哈佛结构，这意味着数据也能存放在程序空间内。要成功使用该数据，在访问数据时必须确保这两种存储空间中的信息是对齐的。

除了正常执行外，PIC24F架构还提供了两种可在操作过程中访问程序空间的方法：

- 使用表指令访问程序空间中任意位置的各个字节或字
- 将程序空间的一部分重新映射到数据空间（程序空间可视性）

表指令允许应用程序读写程序存储器中的一小块区域。这一功能对于访问需要时常更新的数据表来说非常理想。也可通过表操作访问一个程序字的所有字节。重映射方式允许应用程序访问一大块数据，但只限于读操作，它非常适合于在一个大的静态数据表中进行查找。这一方式只能访问程序字的低位字。

4.3.1 对程序空间进行寻址

由于数据和程序空间的地址范围分别为16位和24位，因此需要一个从16位数据寄存器创建一个23位或24位程序地址的方法。方法取决于所采用的接口方式。

对于表操作，使用8位的表存储器页地址寄存器（TBLPAG）定义程序空间中一个32K字的区域。TBLPAG寄存器的8位与16位EA组合形成了一个完整的24位程序空间地址。在这种地址形式下， TBLPAG的MSB用来决定操作是发生在用户存储区 （TBLPAG<7> = 0）中还是配置存储区 （TBLPAG<7> = 1）中。

对于重映射操作，使用10位的扩展数据空间读寄存器（DSRPAG）定义程序空间中的16K字页。当EA的最高位（MSb）为1且DSRPAG的MSB（bit 9）为1时，DSRPAG的低8位与EA的低15位组合形成一个23位的程序空间地址。DSRPAG<8>位决定是映射程序存储空间的低字位（当该位为0时）还是高字位（当该位为1时）。与表操作不同，重映射操作被严格限制在用户存储区中。

表4-14和图4-7显示了如何从数据EA创建程序EA来进行表操作和重映射访问。本文中，P<23:0>指的是程序空间字，而D<15:0>指的是数据空间字。

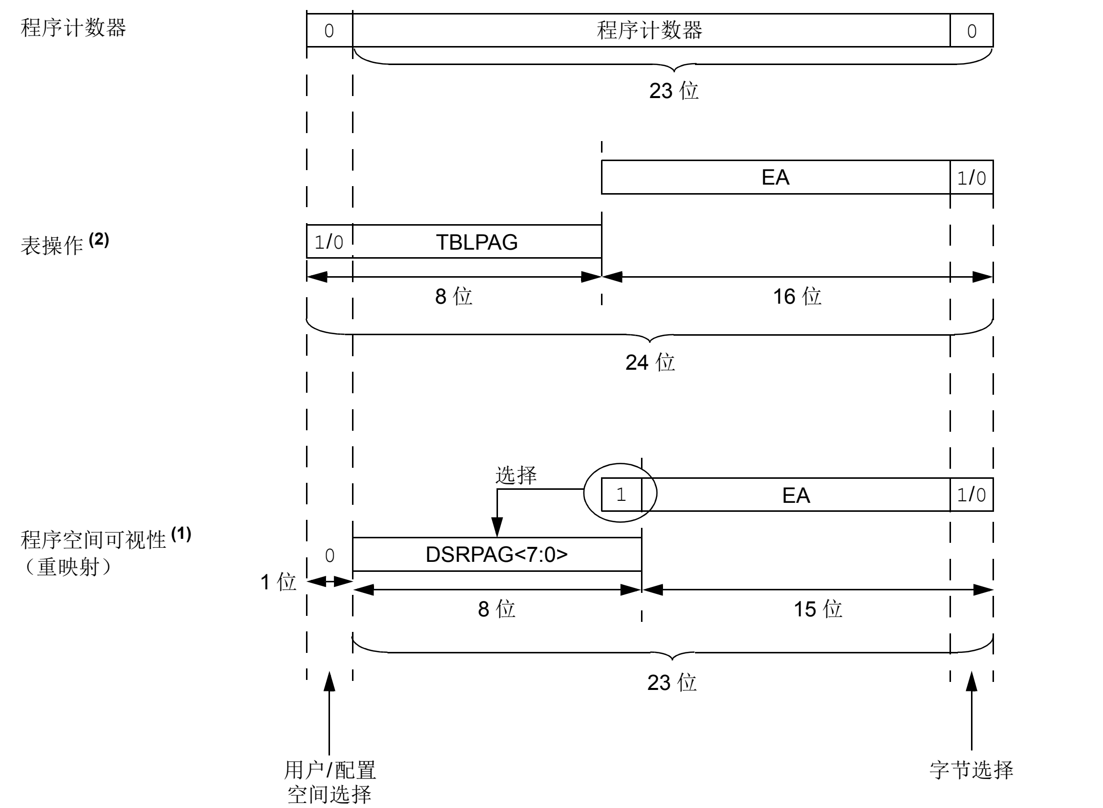
表4-14： 程序空间地址构成

访问类型	访问空间	程序空间地址				
		<23>	<22:16>	<15>	<14:1>	<0>
通过指令访问（代码执行）	用户	0	PC<22:1>			0
		0xx xxxx xxxx xxxx xxxx xxx0				
TBLRD/TBLWT（读/写字节或字）	用户	TBLPAG<7:0>		数据EA<15:0>		
		0xxx xxxx		xxxx xxxx xxxx xxxx		
	配置	TBLPAG<7:0>		数据EA<15:0>		
		1xxx xxxx		xxxx xxxx xxxx xxxx		
程序空间可视性（块重映射/读）	用户	0	DSRPAG<7:0>[(2)]		数据EA<14:0>[(1)]	
		0	xxxx xxxx		xxx xxxx xxxx xxxx	

注 1： 在这种情况下，数据EA<15>始终为1，但并不用它来计算程序空间地址。地址的bit 15为DSRPAG<0>。

2： 在这种情况下，DSRPAG<9> 始终为 1。DSRPAG<8> 决定读取的是程序存储空间的低位字还是高位字。当DSRPAG<8>为0时，读低位字；当DSRPAG<8>为1时，读高位字。

图4-7： 访问程序空间内数据的地址生成方式



注 1： DSRPAG<8>充当字选择。DSRPAG<9>应始终为1以将程序存储器映射到数据存储器。

2： 指令TBLRDH/TBLWTH/TBLRDL/TBLWTL决定是访问程序存储器的高位字还是低位字。TBLRDH/TBLWTH指令访问高位字，而TBLRDL/TBLWTL指令访问低位字。允许对配置存储空间执行表读操作。

4.3.2 使用表指令访问程序存储器中的数据

TBLRDL和TBLWTL指令提供了直接读或写程序空间内任意地址的低位字的方法，无需通过数据空间。TBLRDH和TBLWTH指令是可将一个程序空间字的最高8位作为数据读写的惟一方法。

对于每个连续的24位程序字，PC的递增量为2。这使得程序存储器地址能够直接映射到数据空间地址。于是，程序存储器可以被看作是两个16位字宽的地址空间，它们并排放置，具有相同的地址范围。TBLRDL和TBLWTL访问包含最低有效数据字的空间，而TBLRDH和TBLWTH访问包含高数据字节的空间。

提供了两条表指令来对程序空间执行字节或字（16位）大小的数据读写。读和写都可以采用字节或字操作的形式。

1. TBLRDL（表读低位字）：在字模式下，该指令将程序空间地址中的低位字（P<15:0>）映射到数据地址（D<15:0>）中。

   在字节模式下，低位程序字的高字节或低字节被映射到数据地址的低字节中。当字节选择位为1时映射高字节；当字节选择位为0时映射低字节。

2. TBLRDH（表读高位字）：在字模式下，该指令将程序地址中的整个高位字（P<23:16>）映射到数据地址中。注意，D<15:8>为“虚拟”字节，始终为0。

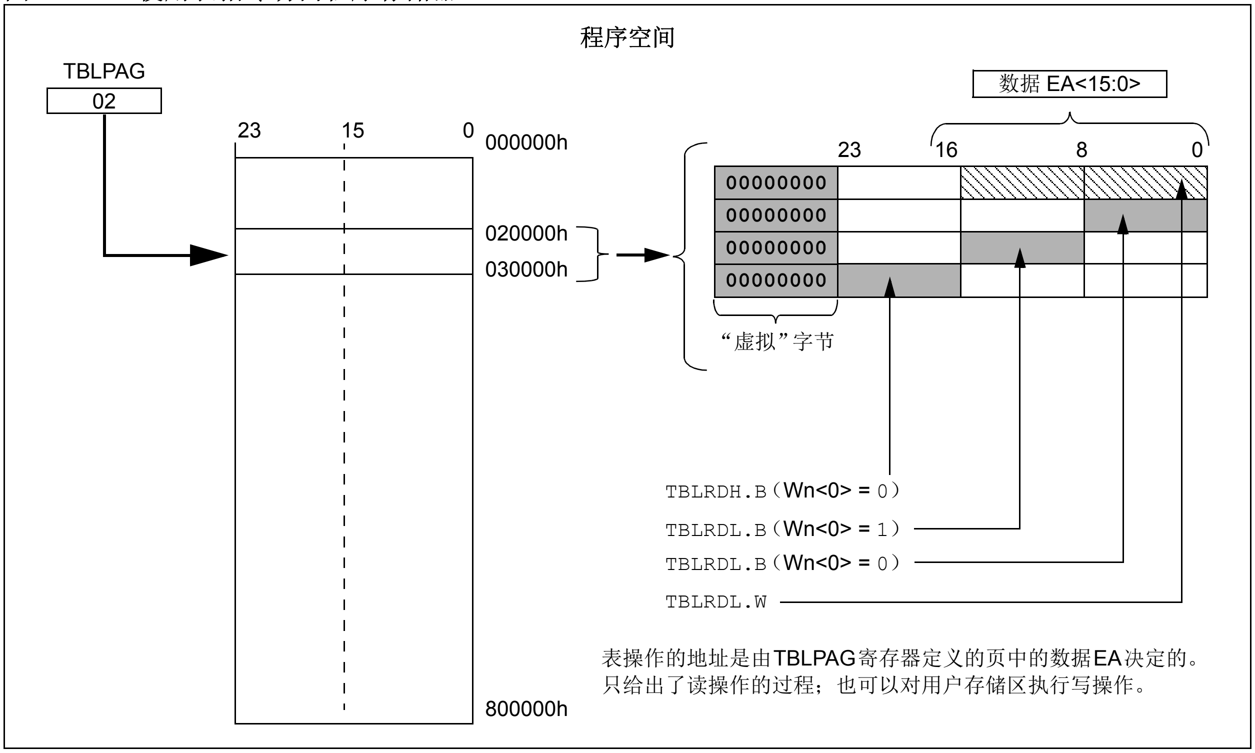
   在字节模式下，程序字的高字节或低字节被映射到数据地址的D<7:0>中，这与上述相同。注意当选择高位“虚拟”字节（字节选择位 = 1）时，数据将始终为0。

表指令TBLWTH和TBLWTL以类似的方式向程序空间地址写入各字节或字。第6.0节“闪存程序存储器”对这两条指令的详细操作给出了说明。

对于所有的表操作，要访问程序存储器空间的哪个区域是由表存储器页地址寄存器（TBLPAG）决定的。TBLPAG可寻址器件的整个程序存储空间，包括用户空间和配置空间。当TBLPAG<7> = 0时，表页位于用户存储空间中。当TBLPAG<7> = 1时，表页位于配置空间中。

注：	仅表读操作可在存放器件ID的配置存储空间内执行。不允许表写操作。

图4-8： 使用表指令访问程序存储器



4.3.3 使用EDS从程序存储器读数据

可选择将数据空间的高32 KB映射到程序空间中的任一16K字页中。这提供了通过数据空间对存储的常量数据的透明访问，而无需使用特殊指令（即TBLRDL/H指令）。

当EA的MSb为1且DSRPAG<9>位也为1时，就可以通过数据空间访问程序空间。DSRPAG的低8位和Wn<14:0>位组合形成了一个23位EA来访问程序存储器。DSRPAG<8>决定应访问哪个字；当该位为0时，访问程序存储器的低位字，当该位为1时，访问程序存储器的高位字。

整个程序存储器分为512个EDS页，从200h至3FFh，每个页由16K字数据组成。页200h至2FFh对应程序存储器的低位字，而300h至3FFh对应程序存储器的高位字。

使用EDS技术，可访问整个程序存储器。此前，不支持对程序存储器高位字的访问。

表4-15提供了与EDS页和源地址对应的程序存储器的23位EDS地址。

对于使用PSV且在REPEAT循环外执行的指令，MOV和MOV.D指令除了规定的执行时间之外，还需要一个额外的指令周期。其他所有的指令，除了规定的指令执行时间之外，还需要两个额外的指令周期。

对于使用PSV且在REPEAT循环内执行的指令，下列情况，除了规定的指令执行时间之外，还需要两个额外的指令周期：

- 在第一次迭代中执行的指令
- 在最后一次迭代中执行的指令
- 由于中断而退出循环之前执行的指令
- 中断得到处理后再次进入循环时执行的指令

REPEAT循环的所有其他各次迭代，都允许使用PSV访问数据的指令在一个周期内执行。

表4-15： 不同页和地址的EDS程序地址

DSRPAG（数据空间读寄存器）	间接寻址时的源地址	指向EDS的23位EA	备注
200h • • • 2FFh	8000h至FFFFh	000000h至007FFEh • • • 7F8000h至7FFFFEh	仅用于读操作的低4M字程序指令（8 MB）。
300h • • • 3FFh		000001h至007FFFh • • • 7F8001h至7FFFFFh	仅用于读操作的高4M字程序指令（4 MB保留；4 MB为虚拟字节）。
000h		无效地址	地址错误陷阱。(1)

注 1： 当源/目标地址高于8000h且DSRPAG/DSWPAG为0时，将发生地址错误陷阱。

例4-3： 从程序存储器进行EDS读操作的代码（汇编语言）

```
; Set the EDS page from where the data to be read
    mov     #0x0202, w0
    mov     w0, DSRPAG                ;page 0x202, consisting lower words, is selected for read
    mov     #0x000A, w1               ;select the location (0x0A) to be read
    bset    w1, #15                   ;set the MSB of the base address, enable EDS mode
;Read a byte from the selected location
    mov.b   [w1++], w2                ;read Low byte
    mov.b   [w1++], w3                ;read High byte
;Read a word from the selected location
    mov     [w1], w2                  ;
;Read Double - word from the selected location
    mov.d   [w1], w2                  ;two word read, stored in w2 and w3
```

图4-9： 访问低位字的程序空间可视性

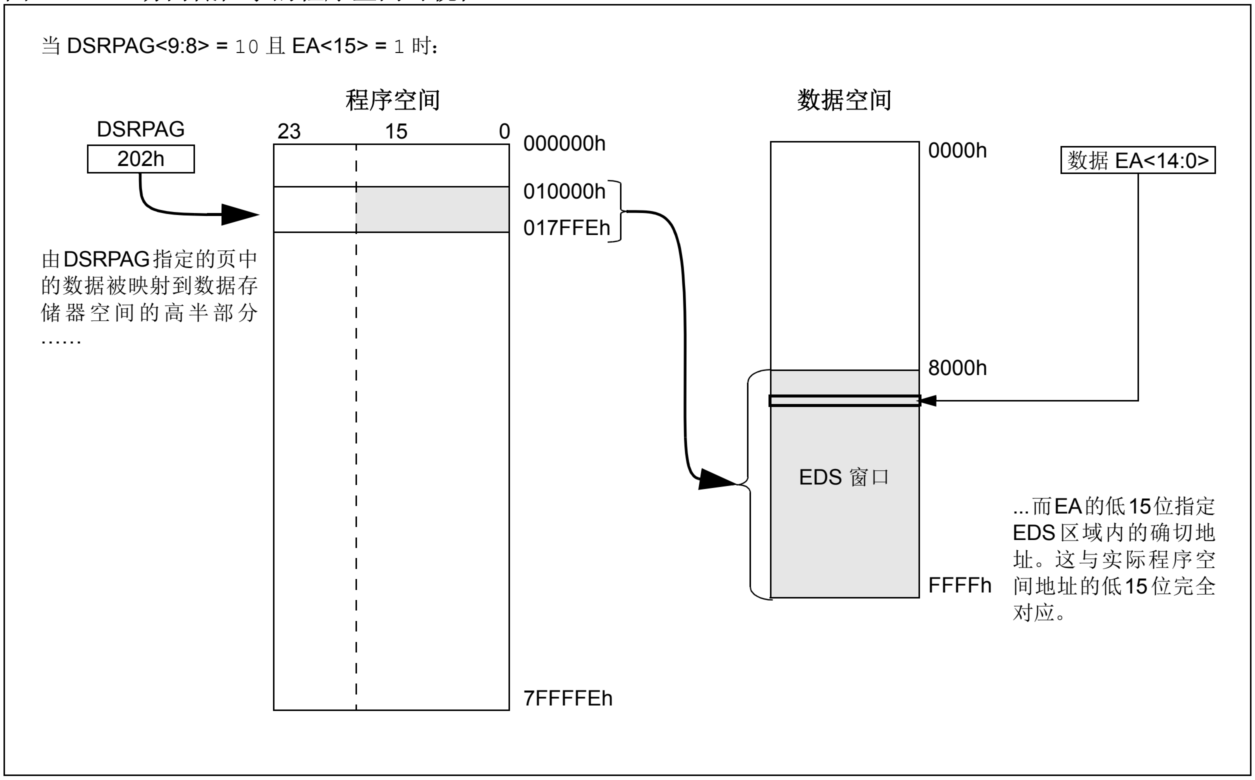
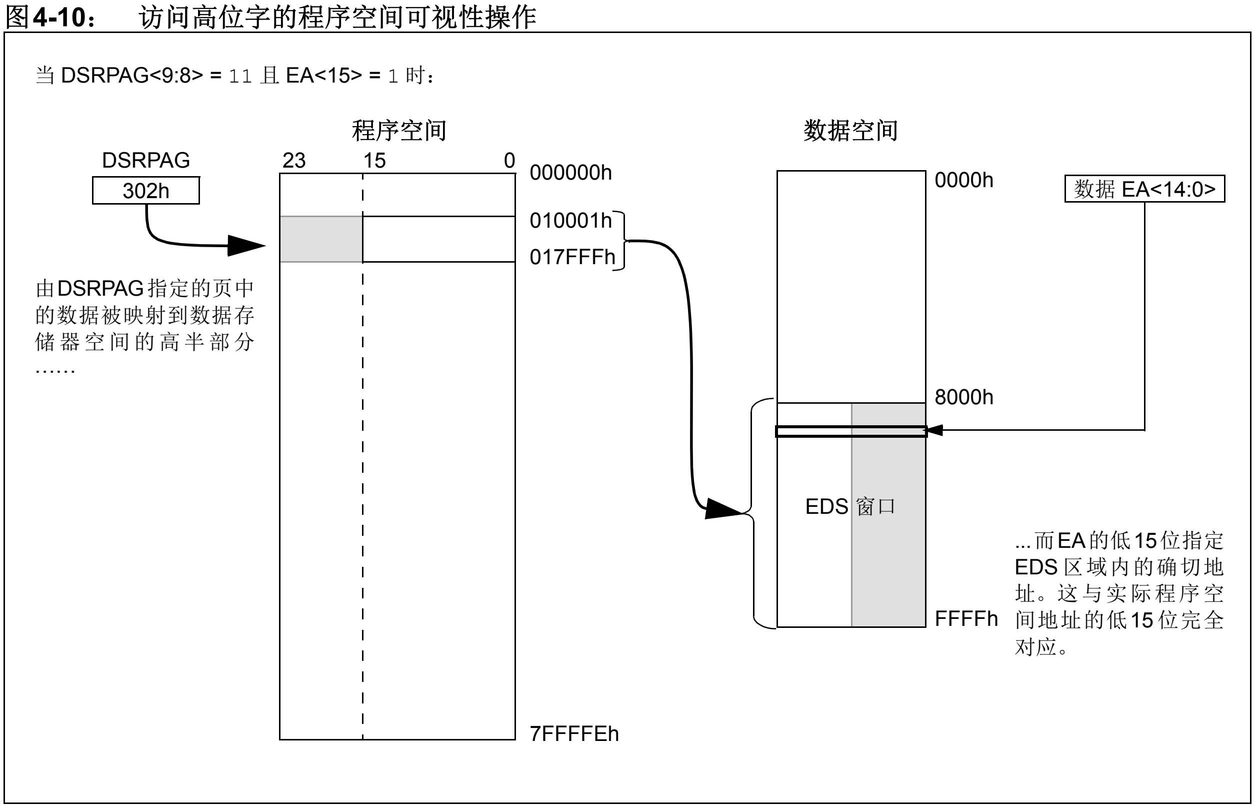


图4-10： 访问高位字的程序空间可视性操作



5.0 直接存储器访问控制器（DMA）

> 注： 本数据手册总结了PIC24FJ256GA705系列器件的功能。但是不应把本手册当做无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24系列参考手册》的**“直接存储器访问控制器（DMA）”**（DS39742），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

直接存储器访问（DMA）控制器旨在为工作在SFR总线上的高数据吞吐量外设提供服务，使这些外设可直接访问数据存储器，而降低对CPU密集型管理的需要。通过使这些数据密集型外设共享各自的数据路径，也可减轻主数据总线负荷，从而进一步降低功耗。

DMA控制器充当外设及CPU的直接扩展。DMA控制器位于CPU和使能DMA的外设之间的单片机数据总线上，可直接访问SRAM。这将SFR总线分为两个总线，使DMA控制器可访问位于新的DMA SFR总线上具有DMA功能的外设。该控制器充当DMA SFR总线上的主器件，控制具有DMA功能的外设的数据流。

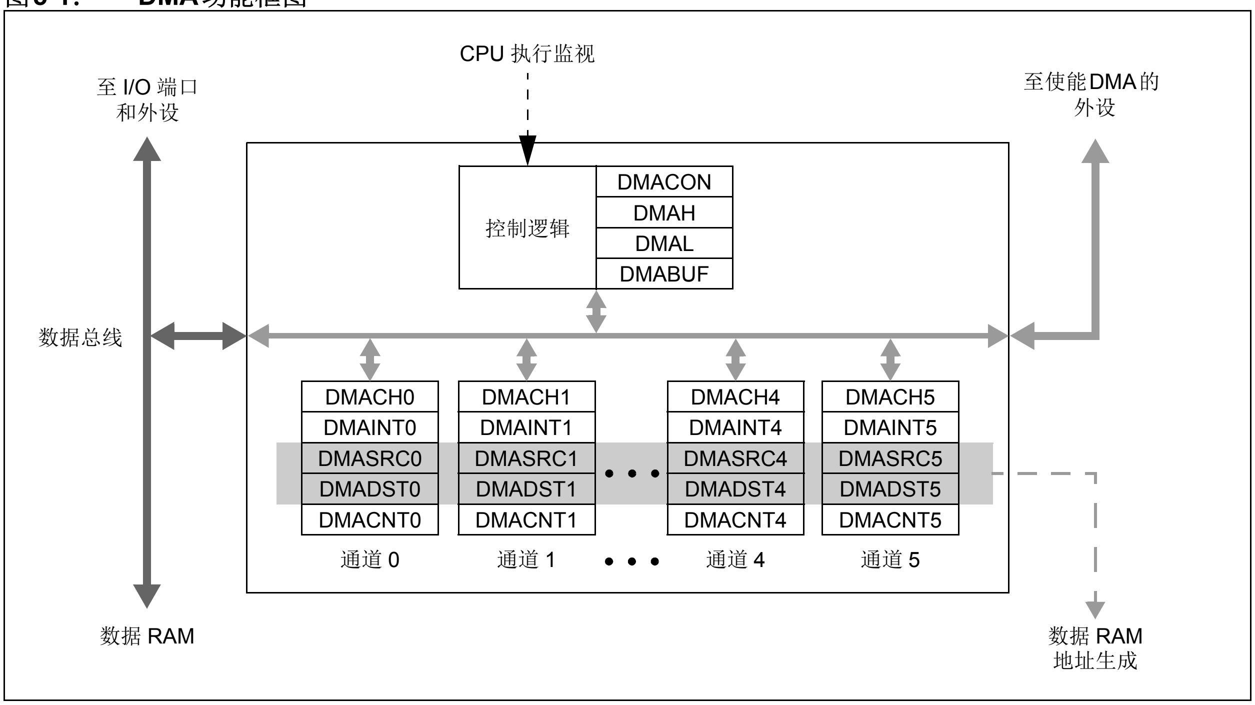
该控制器还直接监视CPU指令处理，使其知道何时CPU需要访问DMA总线上的外设，并根据需要自动放弃对CPU的控制。这会增加数据处理有效带宽，而DMA操作又不会造成处理器停顿。这使得控制器对用户基本透明。

DMA控制器具有以下特性：

- 6条独立的可独立编程通道
- 与CPU并行操作（无DMA导致的等待状态）
- DMA总线仲裁
- 5个可编程地址模式
- 4个可编程传输模式
- 4个灵活的内部数据传输模式
- 支持字节或字数据传输
- 每个通道具有16位源地址和目标地址寄存器，可动态更新和重载
- 16位事务计数寄存器，可动态更新和重载
- 高/低地址限制寄存器
- 计数器半满中断
- 软件触发传输
- 用于对称缓冲操作的空写模式

图5-1显示了DMA控制器的简化框图。

图5-1： DMA功能框图



5.1 DMA操作汇总

DMA控制器可根据多个不同的参数在各地址间传输数据。可为任一事务独立配置这些参数中的每一个，此外，任一或所有DMA通道均可同时独立执行不同的事务。事务按以下参数分类：

- 源和目标（SFR和数据RAM）
- 数据大小（字节或字）
- 触发源
- 传输模式（单次、重复或连续）
- 寻址模式（固定地址或地址块，带或不带地址递增/递减）

此外，DMA控制器还为所有通道提供通道优先级仲裁。

5.1.1 源和目标

使用DMA控制器，数据可在数据空间中任意两个地址之间传输。SFR空间（0000h至07FFh）或数据RAM空间（0800h或FFFFh）既可作为源，又可作为目标。数据可在这些区域之间双向传输或在任一区域内的各地址之间传输。图5-2给出了四种不同的组合。

如有必要保护数据RAM的区域，DMA控制器允许用户为高于SFR空间地址的数据空间操作设置高/低地址边界。边界由DMAH和DMAL限制寄存器设置。如果DMA通道尝试在地址边界之外执行操作，则终止事务并产生中断。

5.1.2 数据大小

DMA控制器可处理8位和16位事务。用户可使用SIZE位（DMACHn<1>）选择大小。默认情况下，每条通道都配置为字大小事务。选择字节大小事务时，源和/或目标地址的LSb确定数据代表数据RAM单元的高字节还是低字节。

5.1.3 触发源

DMA控制器可使用器件中断源中的任意一个来发起事务。DMA触发源以其自然中断优先级的反向顺序列出，如表5-1所示。

由于任何事务的源和目标地址均可独立于触发源进行编程，所以DMA控制器可使用任何触发信号对任何外设执行操作。这还可使要级联的DMA通道执行更复杂的传输操作。

5.1.4 传输模式

根据每次触发时要传输的数据量，DMA控制器支持四种类型的数据传输。

- 单次：每次触发时发生一个事务。
- 连续：每次触发时发生一系列连续事务，事务数量由DMACNTn事务计数器确定。
- 重复单次：重复执行一个事务，每次触发执行一次，直到DMA通道不可用。
- 重复连续：重复执行一系列事务，每次触发执行一轮，直到DMA通道不可用。

所有传输模式都允许选择在事务完成后自动重载源地址、目标地址和计数器值。重复传输模式自动完成此操作。

5.1.5 寻址模式

DMA控制器还支持在单独地址或地址范围之间进行传输。这四种基本选项是：

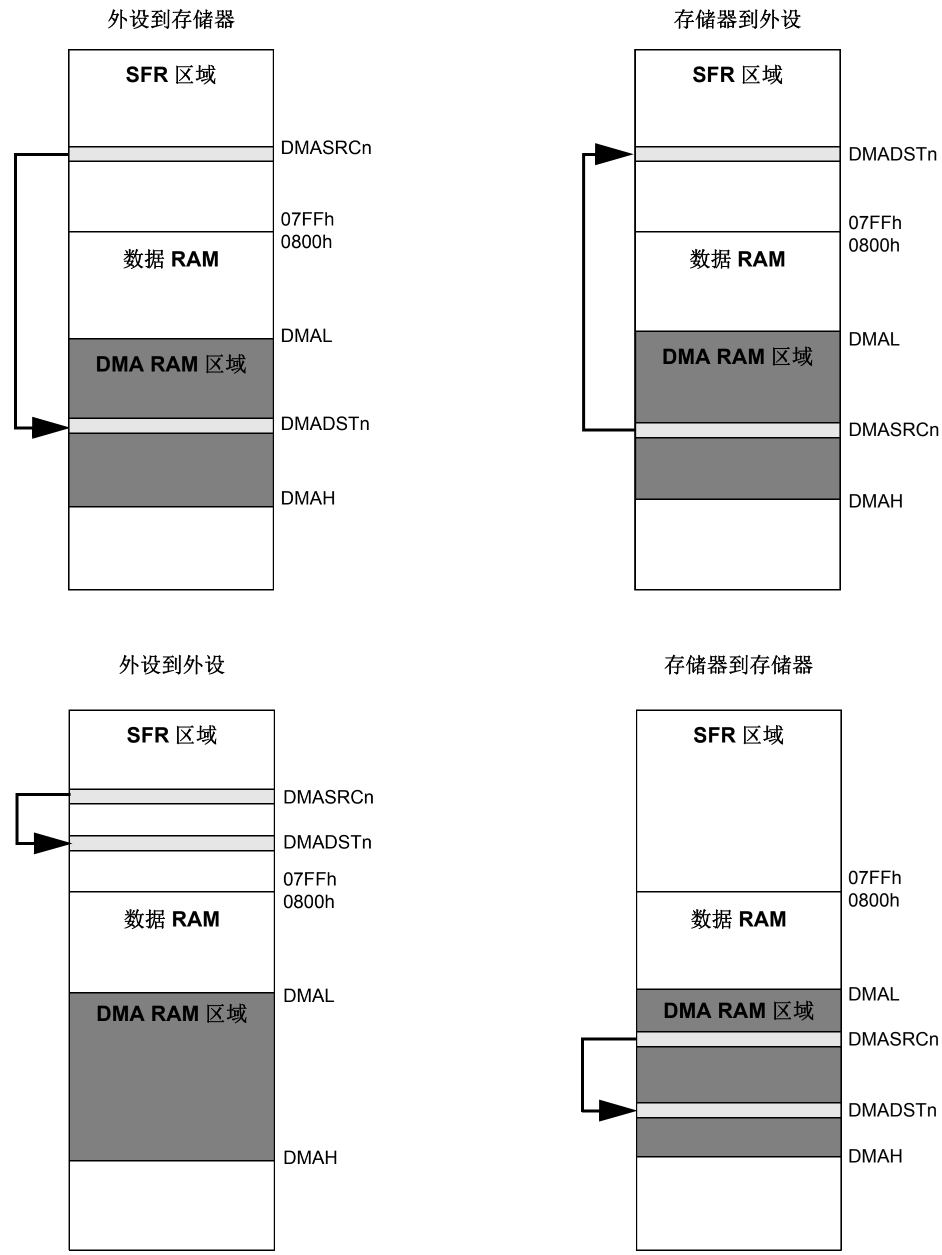
- 固定地址到固定地址：两个恒定地址之间
- 固定地址到地址块：从恒定源地址到目标地址范围
- 地址块到固定地址：从源地址范围到单个恒定目标地址
- 地址块到地址块：从源地址范围到目标地址范围

块寻址模式提供源和/或目标地址的自动递增和自动递减选项。

除了这四种基本模式之外，DMA控制器还支持外设间接寻址（PIA）模式，其中源或目标地址由DMA控制器和支持PIA的外设共同生成。使能时，DMA通道提供基本源和/或目标地址，而外设提供固定范围的偏移地址。

对于PIC24FJ256GA705系列器件，12位A/D转换器模块是惟一支持PIA的外设。有关其在PIA模式下应用的详细信息，请参见**第24.0节“带阈值检测功能的12位A/D转换器”**。

图5-2： DMA数据传输类型



注： 存储器区域的相对大小未按比例显示。

5.1.6 通道优先级

各条DMA通道相互独立工作，但为访问数据和DMA总线也相互竞争。发生访问冲突时，DMA控制器使用用户可选择的优先级机制对各个通道进行仲裁。有两种机制：

- 轮转：两条或多条通道发生冲突时，编号较低的通道在第一次冲突时获得优先权。随后再次冲突时，编号较高的通道根据通道编号依次获得优先权。
- 固定：两条或多条通道发生冲突时，编号最低的通道始终获得优先权，无论之前发生过什么情况；但是，正在接受处理的通道无法立即重新触发。如果较高优先级的通道不断地请求服务，则它将被安排在具有待处理请求的下一个较低优先级通道之后进行服务。

5.2 典型设置

要设置DMA通道用于基本数据传输：

1. 使能DMA控制器（DMAEN = 1）并通过将PRSSEL置1或清零选择适当的通道优先级机制。
2. 使用用于数据RAM操作的适当高/低地址边界设置DMAH和DMAL。
3. 选择要使用的DMA通道并禁止其操作（CHEN = 0）。
4. 将事务的相应源和目标地址编程到通道的DMASRCn和DMADSTn寄存器。对于PIA寻址模式，使用基地址值。
5. 设置DMACNTn寄存器以获得每次传输的触发数量（单次或连续模式）或要传输的字（字节）数（重复模式）。
6. 置1或清零SIZE位以选择数据大小。
7. 编程TRMODE<1:0>位以选择数据传输模式。
8. 编程SAMODE<1:0>和DAMODE<1:0>位以选择寻址模式。
9. 通过将CHEN位置1使能DMA通道。
10. 允许触发源中断。

5.3 外设模块禁止

与其他外设模块不同，DMA控制器的通道无法使用外设模块禁止（PMD）寄存器单独断电，而是将通道分为两组进行控制。DMA0MD位（PMD7<4>）有选择地控制DMACH0到DMACH3。DMA1MD位（PMD7<5>）控制DMACH4和DMACH5。将这两位都置1可有效禁止DMA控制器。

5.4 寄存器

DMA控制器使用一些寄存器来控制其操作。寄存器的数量取决于特定器件所实现的通道数。

始终有4个模块级寄存器（1个控制寄存器和3个缓冲/地址寄存器）：

- DMACON：DMA引擎控制寄存器（寄存器5-1）
- DMAH和DMAL：DMA高/低地址限制寄存器
- DMABUF：DMA数据缓冲区

每个DMA通道实现5个寄存器（2个控制寄存器和3个缓冲/地址寄存器）：

- DMACHn：DMA通道n控制寄存器（寄存器5-2）
- DMAINTn：DMA通道n中断寄存器（寄存器5-3）
- DMASRCn：通道n的DMA数据源地址指针
- DMADSTn：通道n的DMA数据目标地址指针
- DMACNTn：通道n的DMA事务计数器

PIC24FJ256GA705系列器件共有34个寄存器。

寄存器5-1： DMACON：DMA引擎控制寄存器

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
DMAEN	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	PRSSEL
bit 7							bit 0

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

bit 15 **DMAEN**：DMA模块使能位
1 = 使能模块
0 = 禁止模块并终止所有正在进行的DMA操作

bit 14-1 **未实现**：读为0

bit 0 **PRSSEL**：通道优先级机制选择位
1 = 轮转机制
0 = 固定优先级机制

寄存器5-2： DMACHn：DMA通道n控制寄存器

U-0	U-0	U-0	r-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	NULLW	RELOAD[(1)]	CHREQ[(3)]
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	SIZE	CHEN
bit 7							bit 0

图注：	r = 保留位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15-13 **未实现：**读为0

bit 12 **保留：**保持为0

bit 11 **未实现：**读为0

bit 10 **NULLW：**空写模式位
1 = 每次写DMADSTn时，启动对DMASRCn的假写操作
0 = 不启动任何假写操作

bit 9 **RELOAD：**地址和计数重载位[(1)]
1 = 在启动下一个操作时，DMASRCn、DMADSTn和DMACNTn寄存器都重新装载其先前的值
0 = 在启动下一个操作时，DMASRCn、DMADSTn和DMACNTn不会重载[(2)]

bit 8 **CHREQ：**DMA通道软件请求位[(3)]
1 = DMA请求由软件启动，在DMA传输结束时自动清零
0 = 无任何DMA请求在等待处理

bit 7-6 **SAMODE<1:0>：**源地址模式选择位
11 = DMASRCn用于外设间接寻址且保持不变
10 = 传输完成后，DMASRCn根据SIZE位递减
01 = 传输完成后，DMASRCn根据SIZE位递增
00 = 传输完成后，DMASRCn保持不变

bit 5-4 **DAMODE<1:0>：**目标地址模式选择位
11 = DMADSTn用于外设间接寻址且保持不变
10 = 传输完成后，DMADSTn根据SIZE位递减
01 = 传输完成后，DMADSTn根据SIZE位递增
00 = 传输完成后，DMADSTn保持不变

bit 3-2 **TRMODE<1:0>：**传输模式选择位
11 = 重复连续模式
10 = 连续模式
01 = 重复单次模式
00 = 单次模式

bit 1 **SIZE：**数据大小选择位
1 = 字节（8位）
0 = 字（16位）

bit 0 **CHEN：**DMA通道使能位
1 = 使能相应的通道
0 = 禁止相应的通道

注 1： 只需存储DMACNTn的原始值来恢复DMASRCn和DMADSTn的原始值。
2： DMASRCn、DMADSTn和DMACNTn始终在重复传输模式（DMACHn<2> = 1）时重载，与RELOAD位的状态无关。
3： CHREQ置1时执行的传输数量取决于TRMODE<1:0>的配置。

寄存器5-3： DMAINTn：DMA通道n中断寄存器

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DBUFWF[(1)]	CHSEL6	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
HIGHIF[(1,2)]	LOWIF[(1,2)]	DONEIF[(1)]	HALFIF[(1)]	OVRUNIF[(1)]	—	—	HALFEN
bit 7							bit 0

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

bit 15 **DBUFWF**：DMA缓冲数据写标志位[(1)]
1 = 在空写模式下，DMA缓冲区的内容未被写入DMADSTn或DMASRCn中指定的地址单元
0 = 在空写模式下，DMA缓冲区的内容被写入DMADSTn或DMASRCn中指定的地址单元

bit 14-8 **CHSEL<6:0>**：DMA通道触发源选择位
完整列表请参见表5-1。

bit 7 **HIGHIF**：DMA高地址限制中断标志位[(1,2)]
1 = DMA通道尝试访问高于DMAH或数据RAM空间上限的地址
0 = DMA通道未调用高地址限制中断

bit 6 **LOWIF**：DMA低地址限制中断标志位[(1,2)]
1 = DMA通道尝试访问低于DMAL但超出SFR范围（07FFh）的DMA SFR地址
0 = DMA通道未调用低地址限制中断

bit 5 **DONEIF**：DMA完成操作中断标志位[(1)]
如果CHEN = 1：
1 = 前一DMA会话结束且完成
0 = 当前DMA会话尚未完成
如果CHEN = 0：
1 = 前一DMA会话结束且完成
0 = 前一DMA会话结束但未完成

bit 4 **HALFIF**：DMA 50%水印值中断标志位[(1)]
1 = DMACNTn已达到与0000h之间的中点
0 = DMACNTn未达到中点

bit 3 **OVRUNIF**：DMA通道溢出标志位[(1)]
1 = DMA通道仍在完成基于前一触发信号的操作时被触发
0 = 未发生溢出条件

bit 2-1 **未实现**：读为0

bit 0 **HALFEN**：完成一半水印位
1 = 当DMACNTn达到其中点和完成时调用中断
0 = 仅在传输完成时调用中断

注 1： 用软件将这些标志位置1不会产生中断。
2： 实际访问之前未完成地址超限测试（DMASRCn或DMADSTn大于DMAH或小于DMAL）。

表5-1： DMA触发源

CHSEL<6:0>	触发信号（中断）	CHSEL<6:0>	触发信号（中断）
0000000	关闭	1000001	UART2发送中断
0001001	MCCP4 IC/OC中断	1000010	UART2接收中断
0001010	MCCP4定时器中断	1000011	UART2错误中断
0001011	MCCP3 IC/OC中断	1000100	UART1发送中断
0001100	MCCP3定时器中断	1000101	UART1接收中断
0001101	MCCP2 IC/OC中断	1000110	UART1错误中断
0001110	MCCP2定时器中断	1001011	DMA通道5中断
0001111	MCCP1 IC/OC中断	1001100	DMA通道4中断
0010000	MCCP1定时器中断	1001101	DMA通道3中断
0010100	OC3中断	1001110	DMA通道2中断
0010101	OC2中断	1001111	DMA通道1中断
0010110	OC1中断	1010000	DMA通道0中断
0011010	IC3中断	1010001	A/D中断
0011011	IC2中断	1010011	PMP中断
0011100	IC1中断	1010100	HLVD中断
0100000	SPI3接收中断	1010101	CRC中断
0100001	SPI3发送中断	1011011	CLC2输出
0100010	SPI3一般中断	1011100	CLC1输出
0100011	SPI2接收中断	1011110	RTCC闹钟中断
0100100	SPI2发送中断	1100001	TMR3中断
0100101	SPI2一般中断	1100010	TMR2中断
0100110	SPI1接收中断	1100011	TMR1中断
0100111	SPI1发送中断	1100110	CTMU触发信号
0101000	SPI1一般中断	1100111	比较器中断
0101111	I2C2从中断	1101000	INT4中断
0110000	I2C2主中断	1101001	INT3中断
0110001	I2C2总线冲突中断	1101010	INT2中断
0110010	I2C1从中断	1101011	INT1中断
0110011	I2C1主中断	1101100	INT0中断
0110100	I2C1总线冲突中断	1101101	电平变化中断（IOC）中断

6.0 闪存程序存储器

> **注：** 本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的 **"PIC24F闪存程序存储器"**（DS30009715），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

PIC24FJ256GA705系列器件包含内部闪存程序存储器，用于存储和执行应用程序代码。程序存储器是可读写并可擦除的。闪存有以下四种编程方式：

- 在线串行编程（ICSP）
- 运行时自编程（Run-Time Self-Programming，RTSP）
- JTAG
- 增强型在线串行编程（增强型ICSP）

ICSP允许在最终应用电路中对PIC24FJ256GA705系列器件进行串行编程。只需五根线就可以完成编程，它们分别是编程时钟线（PGCx）、编程数据线（PGDx）、电源线（VDD）、接地线（VSS）和主复位线（$\overline{MCLR}$）。这允许用户在生产电路板时使用未编程器件，而仅在产品交付之前才对单片机进行编程，从而可以使用最新版本的固件或者定制固件进行编程。

RTSP是通过`TBLRD`（表读）和`TBLWT`（表写）指令完成的。使用RTSP，用户可以一次将128条指令（384字节）的数据块写入程序存储器，也可以一次擦除1024条指令（3072字节）的数据块。

该器件实现了7位纠错码（Error Correcting Code，ECC）。NVM块包含写入和读取闪存中的ECC位的逻辑。闪存与相应的ECC奇偶校验位同时编程。ECC提供了更强的闪存纠错功能。ECC单位错误可透明纠正；ECC双位错误会导致陷阱。

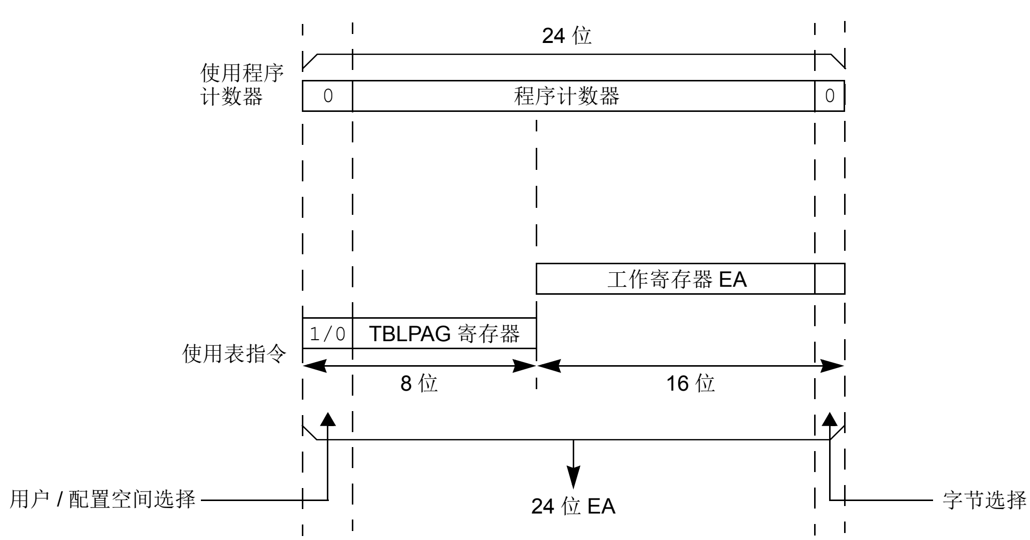
6.1 表指令和闪存编程

闪存的所有编程都是通过表读和表写指令完成的，与使用的编程方法无关。这些指令允许器件在正常工作模式下通过数据存储器直接读写程序存储空间。24位程序存储器目标地址由TBLPAG<7:0>位和表指令中指定的W寄存器中的有效地址（EA）组成，如图6-1所示。

`TBLRDL`和`TBLWTL`指令用于读或写程序存储器的bit<15:0>。`TBLRDL`和`TBLWTL`能以字或字节模式访问程序存储器。

`TBLRDH`和`TBLWTH`指令用于读或写程序存储器的bit<23:16>。`TBLRDH`和`TBLWTH`指令也能以字或字节模式访问程序存储器。

图6-1： 表寄存器寻址



6.2 RTSP操作

PIC24F闪存程序存储器阵列以128条指令或384字节的行为单位构成。RTSP允许用户一次擦除8行（1024条指令）的块并一次编程一行，还可以编程两个指令字块。

8行擦除块和单行写入块都是边界对齐的，从程序存储器起始地址开始，分别到3072字节边界和384字节边界。

当使用TBLWT指令将数据写入程序存储器时，数据并未直接写入存储器。实际上，使用表写指令写入的数据在执行编程序列之前一直存储在保持锁存器中。

可以执行任意数量的TBLWT指令，且写操作都将成功执行。但是，需要128条TBLWT指令来写存储器的整行。

要确保在写操作期间没有数据被破坏，应将所有未使用的地址编程为FFFFFFh。这是因为保持锁存器复位为未知状态，因此当地址处于复位状态时，就可能改写未被覆写的行上的存储单元。

RTSP编程的基本步骤是将表指针设置为指向编程锁存器，执行一系列TBLWT指令来加载缓冲区，并将NVMADRU/NVMADR寄存器设置为指向目标地址。通过将NVMCON寄存器的控制位置1来执行编程。

可按任何顺序装入数据，且在执行写操作之前可以多次写保持寄存器。但后续写操作将覆盖掉先前的所有写操作。

注：	不推荐多次写入一个存储单元而不擦除其内容。

由于只写缓冲区，因此所有表写操作都是单字写操作（2个指令周期）。编程每行都需要一个编程周期。

6.3 JTAG操作

PIC24F系列器件支持JTAG边界扫描。边界扫描可以通过验证引脚到PCB的连通性改进制造工艺。

6.4 增强型在线串行编程

增强型在线串行编程使用称为编程执行程序（Program Executive，PE）的片上自举程序来管理编程过程。通过使用SPI数据帧格式，编程执行程序可以擦除、编程和校验程序存储器。有关增强型ICSP的更多信息，请参见器件编程规范。

6.5 控制寄存器

有四个用于读写闪存程序存储器的SFR：NVMCON、NVMADRU、NVMADR和NVMKEY。

NVMCON寄存器（寄存器6-1）控制要擦除的块、要编程的存储器类型以及编程周期的开始时间。

NVMKEY是只写寄存器，用于写保护。要启动编程或擦除过程，用户必须把55h和AAh连续写入NVMKEY寄存器。更多详细信息，请参见**第6.6节“编程操作”**。

NVMADRU/NVMADR寄存器包含NVM写操作或擦除操作目标地址的最高字节和低位字。有一些操作（例如，芯片擦除）只针对固定存储单元，无需地址值。

6.6 编程操作

在RTSP模式下，对内部闪存进行编程或擦除需要执行完整的编程序列。在编程或擦除操作期间，处理器将停止（等待），直到操作完成。将WR位（NVMCON<15>）置1启动操作，当操作完成时WR位会自动清零。

寄存器6-1： NVMCON：闪存控制寄存器

R/S-0, HC[1]	R/W-0[1]	R-0, HSC[1]	R/W-0	r-0	r-0	U-0	U-0
WR	WREN	WRERR	NVMSIDL	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0[1]	R/W-0[1]	R/W-0[1]	R/W-0[1]
—	—	—	—	NVMOP3[2]	NVMOP2[2]	NVMOP1[2]	NVMOP0[2]
bit 7							bit 0

图注：	S = 可置1位	HC = 硬件清零位	r = 保留位
R = 可读位	W = 可写位	0 = 清零	x = 未知
-n = POR时的值	1 = 置1	U = 未实现位，读为0	
HSC = 硬件置1/清零位			

bit 15 **WR：**写控制位[1]
1 = 启动闪存编程或擦除操作；该操作是自定时的，一旦操作完成此位即由硬件清零
0 = 编程或擦除操作完成，并处于停止状态

bit 14 **WREN：**写使能位[1]
1 = 使能闪存编程/擦除操作
0 = 禁止闪存编程/擦除操作

bit 13 **WRERR：**写序列错误标志位[1]
1 = 试图执行错误的编程或擦除序列，或者发生终止（任何试图将WR位置1的操作都会自动置1此位）
0 = 编程或擦除操作正常完成

bit 12 **NVMSIDL：**NVM空闲模式停止位
1 = 当器件进入空闲模式时，不为程序存储器供电
0 = 当器件进入空闲模式时，为程序存储器供电，使其处于待机模式

bit 11-10 **保留：**保持为0

bit 9-4 **未实现：**读为0

bit 3-0 **NVMOP<3:0>：**NVM操作选择位[1,2]
1110 = 整片擦除用户存储器（不擦除器件ID、用户OTP或可执行程序存储区）
0100 = 未使用
0011 = 擦除程序存储器或可执行程序存储区的一页
0010 = 行编程操作
0001 = 双字编程操作

注 1： 这些位只能在上电复位时复位。
2： NVMOP<3:0>的所有其他组合均未实现。

6.6.1 闪存程序存储器的编程算法

用户一次可编程闪存程序存储器的一行。要实现此操作，必须擦除包含该行在内的一个8行擦除块。一般步骤如下：

1. 读程序存储器中的8行（1024条指令），并将其存储在数据RAM中。
2. 用需要的新数据更新RAM中对应的程序数据。
3. 擦除块（见例6-1）：
   a) 将NVMOP<3:0>位（NVMCON<3:0>）设置为0011，以配置为块擦除操作。将WREN（NVMCON<14>）位置1。
   b) 将要擦除的块的起始地址写入NVMADRU/NVMADR寄存器。
   c) 将55h写入NVMKEY。
   d) 将AAh写入NVMKEY。
   e) 将WR位（NVMCON<15>）置1。启动擦除周期，CPU停止工作等待擦除周期结束。当擦除操作完成时，WR位自动清零。
4. 更新TBLPAG寄存器以指向器件的编程锁存器。更新NVMADRU/NVMADR寄存器以指向程序存储器中的目标地址。
5. 将数据RAM的前128条指令写入程序存储器缓冲区（见表6-1）。
6. 将程序块写入闪存：
   a) 将NVMOPx位设置为0010，以配置为行编程操作。将WREN位置1。
   b) 将55h写入NVMKEY。
   c) 将AAh写入NVMKEY。
   d) 将WR位置1。启动编程周期，CPU停止工作等待写周期完成。当写闪存的操作完成时，WR位自动清零。
7. 将NVMADRU/NVMADR的值递增1，使用数据RAM中下一个128条指令的块重复步骤4到6，直到将所有1024条指令写回闪存。

为防止意外操作，必须向NVMKEY写入写启动序列，以允许执行任何擦除或编程操作。执行了编程命令以后，用户必须等待一定的编程时间，直到编程操作完成。编程序列开始后紧跟的两条指令必须为NOP，如例6-2所示。

表6-1： 页擦除示例

步骤1：设置NVMCON寄存器，以进行页擦除。

```
MOV    #0x4003, W0
MOV    W0, NVMCON
```

步骤2：将要擦除的页的地址装入NVMADR寄存器对。

```
MOV    #PAGE_ADDR_LO, W0
MOV    W0, NVMADR
MOV    #PAGE_ADDR_HI, W0
MOV    W0, NVMADRU
```

步骤3：将WR位置1。

```
MOV    #0x55, W0
MOV    W0, NVMKEY
MOV    #0xAA, W0
MOV    W0, NVMKEY
BSET   NVMCON, #WR
NOP
NOP
NOP
```


例6-1： 擦除程序存储器块（C语言代码）

```
// C example using MPLAB XC16
    unsigned   long progAddr = 0xXXXXXX;          // Address of row to write
    unsigned   int  offset;
//Set up pointer to the first memory location to be written
    NVMADRU = progAddr>>16;                       // Initialize PM Page Boundary SFR
    NVMADR  = progAddr & 0xFFFF;                  // Initialize lower word of address
    NVMCON  = 0x4003;                             // Initialize NVMCON
    asm("DISI #5");                               // Block all interrupts with priority <7
                                                  // for next 5 instructions
    __builtin_write_NVM();                        // check function to perform unlock
                                                  // sequence and set WR
```

表6-2： 代码存储器编程示例：行写

步骤1：设置NVMCON寄存器，以进行128指令字编程。

```
MOV      #0x4002, W0
MOV      W0, NVMCON
```

步骤2：初始化TBLPAG寄存器以写入锁存器。

```
MOV      #0xFA, W12
MOV      W12, TBLPAG
```

步骤3：将下一组4个指令字装入W0:W5，以进行编程。

```
MOV      #<LSW0>, W0
MOV      #<MSB1:MSB0>, W1
MOV      #<LSW1>, W2
MOV      #<LSW2>, W3
MOV      #<MSB3:MSB2>, W4
MOV      #<LSW3>, W5
```

步骤4：设置读指针（W6）并装载（下一组）写锁存器。

```
CLR      W6
CLR      W7
TBLWTL   [W6++], [W7]
TBLWTH.B [W6++], [W7++]
TBLWTH.B [W6++], [++W7]
TBLWTL   [W6++], [W7++]
TBLWTL   [W6++], [W7]
TBLWTH.B [W6++], [W7++]
TBLWTH.B [W6++], [++W7]
TBLWTL   [W6++], [W7++]
```

步骤5：重复执行步骤3和步骤4，总共执行32次，以将128个指令字装入写锁存器。

步骤6：设置NVMADRU/NVMADR寄存器对，以指向正确的地址。

```
MOV      #DestinationAddress<15:0>, W3
MOV      #DestinationAddress<23:16>, W4
MOV      W3, NVMADR
MOV      W4, NVMADRU
```

步骤7：执行WR位解锁序列并启动写周期。

```
MOV      #0x55, W0
MOV      W0, NVMKEY
MOV      #0xAA, W0
MOV      W0, NVMKEY
BSET     NVMCON, #WR
NOP
NOP
NOP
```


例6-2： 启动编程序列

```
        DISI    #5                          ; Block all interrupts with priority <7
                                            ; for next 5 instructions
        MOV.B   #0x55, W0
        MOV     W0, NVMKEY                  ; Write the 0x55 key
        MOV.B   #0xAA, W1                   ;
        MOV     W1, NVMKEY                  ; Write the 0xAA key
        BSET    NVMCON, #WR                 ; Start the programming sequence
        NOP                                 ; Required delays
        NOP
        BTSC    NVMCON, #15                 ; and wait for it to be
        BRA     $-2                         ; completed
```

6.6.2 编程闪存程序存储器的一个双字

如果已擦除一个闪存单元，则可以使用表写指令对此单元进行编程，从而将两个指令字（2 x 24位）写入写锁存器。TBLPAG 寄存器装载写锁存器的地址，NVMADRU/NVMADR寄存器装载两个要编程的指令字中的第一个指令字的地址。TBLWTL和TBLWTH指令将所需数据写入写锁存器。要将NVMCON寄存器配置为双字写操作，需将NVMOPx位（NVMCON<3:0>）设置为0001。双字写操作是通过执行解锁序列并将WR位置1来执行。例6-3给出了C语言形式的等效程序（使用MPLAB® XC16编译器和内置硬件功能）。

表6-3： 编程闪存程序存储器的一个双字

步骤1： 初始化TBLPAG寄存器以写入锁存器。

```
MOV       #0xFA, W12
MOV       W12, TBLPAG
```

步骤2： 将下一组2个指令字装入W0:W2，以进行编程。

```
MOV       #<LSW0>, W0
MOV       #<MSB1:MSB0>, W1
MOV       #<LSW1>, W2
```

步骤3： 设置读指针（W6）和写指针（W7）并装载（下一组）写锁存器。

```
CLR       W6
CLR       W7
TBLWTL    [W6++], [W7]
TBLWTH.B  [W6++], [W7++]
TBLWTH.B  [W6++], [++W7]
TBLWTL.W  [W6++], [W7++]
```

步骤4： 设置NVMADRU/NVMADR寄存器对，以指向正确的地址。

```
MOV       #DestinationAddress<15:0>, W3
MOV       #DestinationAddress<23:16>, W4
MOV       W3, NVMADR
MOV       W4, NVMADRU
```

步骤5： 设置NVMCON寄存器，以编程两个指令字。

```
MOV       #0x4001, W10
MOV       W10, NVMCON
NOP
```

步骤6： 启动写周期。

```
MOV       #0x55, W1
MOV       W1, NVMKEY
MOV       #0xAA, W1
MOV       W1, NVMKEY
BSET      NVMCON, #WR
NOP
NOP
NOP
```

例6-3：　编程闪存程序存储器的一个双字（C语言代码）

```
// C example using MPLAB XC16
unsigned   long progAddr = 0xXXXXXX;         // Address of word to program
unsigned   int  progData1L = 0xXXXX;         // Data to program lower word of word 1
unsigned   char progData1H = 0xXX;           // Data to program upper byte of word 1
unsigned   int  progData2L = 0xXXXX;         // Data to program lower word  of word 2
unsigned   char progData2H = 0xXX;           // Data to program upper byte of word 2

//Set up NVMCON for word programming
NVMCON = 0x4001;                             // Initialize NVMCON
TBLPAG = 0xFA;                               // Point TBLPAG to the write latches

//Set up pointer to the first memory location to be written
NVMADRU = progAddr>>16;                      // Initialize PM Page Boundary SFR
NVMADR  = progAddr & 0xFFFF;                 // Initialize lower word of address

//Perform TBLWT instructions to write latches
__builtin_tblwtl(0, progData1L);             // Write word 1 to address low word
__builtin_tblwth(0, progData2H);             // Write word 1 to upper byte
__builtin_tblwtl(1, progData2L);             // Write word 2 to address low word
__builtin_tblwth(1, progData2H);             // Write word 2 to upper byte
asm("DISI #5");                              // Block interrupts with priority <7 for next 5
                                             // instructions
__builtin_write_NVM();                       // XC16 function to perform unlock sequence and set WR
```

7.0 复位

> **注：** 本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的“复位”（DS39712），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

复位模块结合了所有复位源并控制器件的主复位信号SYSRST。下面列出了器件的复位源：

- POR：上电复位
- MCLR：主复位引脚复位
- SWR：RESET指令
- WDT：看门狗定时器复位
- BOR：欠压复位
- CM：配置不匹配复位
- TRAPR：陷阱冲突复位
- IOPUWR：非法操作码复位
- UWR：未初始化的W寄存器复位

图7-1给出了复位模块的简化框图。

任何有效的复位源都将使SYSRST信号有效。许多与CPU和外设有关的寄存器被强制为一个已知的复位状态。大多数寄存器不受复位的影响；在POR时它们的状态未知，而在所有其他复位时它们的状态不变。

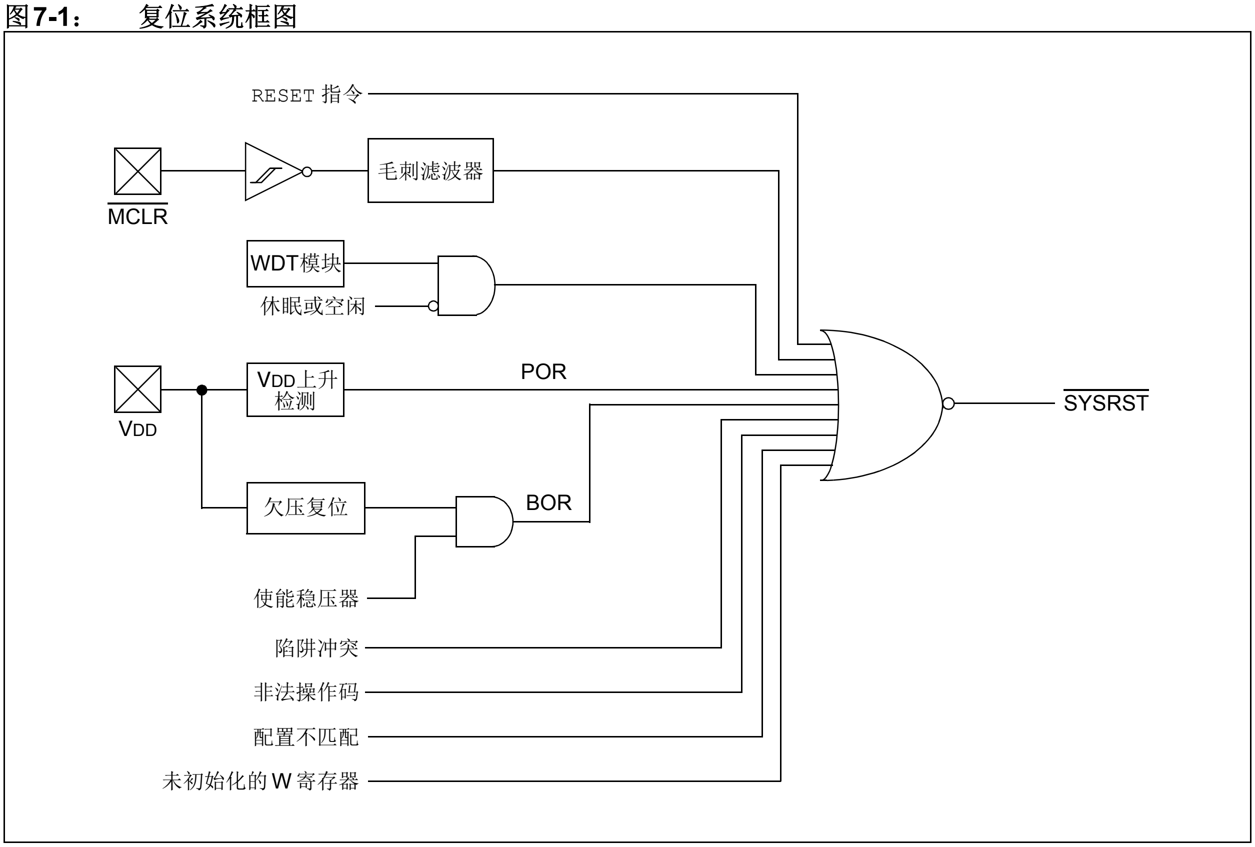
> **注：** 有关寄存器复位状态的信息，请参见本手册中的特定外设或CPU章节。

任何类型的器件复位都会将RCON寄存器中相应的状态位置1以表明复位类型（见寄存器7-1）。上电复位会将除BOR和POR（RCON<1:0>）位之外的所有位清零，而BOR和POR位会被置1。在代码执行过程中，用户随时可将任何位置1或清零。RCON位仅用作状态位。用软件将特定的复位状态位置1不会导致器件发生复位。

RCON寄存器还包含与看门狗定时器和器件节能状态相关的其他位。本数据手册的其他章节中将讨论这些位的功能。

> **注：** RCON寄存器中的状态位应该在被读取后清零，这样下一次器件复位后的RCON寄存器值才会有意义。

图7-1： 复位系统框图



寄存器7-1： RCON：复位控制寄存器

R/W-0	R/W-0	R/W-1	R/W-0	U-0	U-0	R/W-0	R/W-0
TRAPR[(1)]	IOPUWR[(1)]	SBOREN[(5)]	RETEN[(2)]	—	—	CM[(1)]	VREGS[(3)]
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1
EXTR[(1)]	SWR[(1)]	SWDTEN[(4)]	WDTO[(1)]	SLEEP[(1)]	IDLE[(1)]	BOR[(1)]	POR[(1)]
bit 7							bit 0

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

bit 15 **TRAPR：**陷阱复位标志位[(1)]
1 = 已发生陷阱冲突复位
0 = 未发生陷阱冲突复位

bit 14 **IOPUWR：**非法操作码或因访问未初始化的W寄存器而复位的标志位[(1)]
1 = 检测到非法操作码、非法地址模式或未初始化的W寄存器用作地址指针而导致的复位
0 = 未发生由非法操作码或未初始化的W寄存器而导致的复位

bit 13 **SBOREN：**软件控制BOR功能位[(5)]
1 = 使能BOR
0 = 禁止BOR

bit 12 **RETEN：**保持模式使能位[(2)]
1 = 在器件处于休眠模式时（1.2V稳压器为内核供电）使能保持模式
0 = 禁止保持模式；存在正常电压

bit 11-10 **未实现：**读为0

bit 9 **CM：**配置字不匹配复位标志位[(1)]
1 = 已发生配置字不匹配复位
0 = 未发生配置字不匹配复位

bit 8 **VREGS：**从休眠中快速唤醒位[(3)]
1 = 禁止快速唤醒（低功耗）
0 = 使能快速唤醒（高功耗）

bit 7 **EXTR：**外部复位（$\overline{\text{MCLR}}$）引脚位[(1)]
1 = 已发生主复位（引脚）复位
0 = 未发生主复位（引脚）复位

bit 6 **SWR：**软件RESET（指令）标志位[(1)]
1 = 已执行RESET指令
0 = 未执行RESET指令

注 1： 所有复位状态位都可以用软件置1或清零。用软件将这些位中的某一位置1不会导致器件复位。
2： 如果$\overline{\text{LPCFG}}$配置位为1（未编程），则保持稳压器被禁止且RETEN位不起作用。在休眠期间，保持模式可以保留SRAM内容。
3： 在稳压器进入待机模式后将其重新使能，将在从休眠中唤醒时增加延时T_{VREG}。不使用稳压器的应用应该将此位置1，以防止发生此延时。
4： 如果FWDTEN<1:0>配置位为11（未编程），则WDT总是使能，而与SWDTEN位的设置无关。
5： BOREN<1:0>（FPOR<1:0>）配置位必须设置为01，以使SBOREN生效。

寄存器7-1： RCON：复位控制寄存器 （续）

bit 5 **SWDTEN**：软件使能/禁止WDT位[4]
1 = 使能WDT
0 = 禁止WDT

bit 4 **WDTO**：看门狗定时器超时标志位[1]
1 = 已发生WDT超时
0 = 未发生WDT超时

bit 3 **SLEEP**：从休眠模式唤醒标志位[1]
1 = 器件处于休眠模式
0 = 器件未处于休眠模式

bit 2 **IDLE**：从空闲模式唤醒标志位[1]
1 = 器件处于空闲模式
0 = 器件不处于空闲模式

bit 1 **BOR**：欠压复位标志位[1]
1 = 已发生欠压复位（上电复位后也置1）
0 = 未发生欠压复位

bit 0 **POR**：上电复位标志位[1]
1 = 已发生上电复位
0 = 未发生上电复位

注 1： 所有复位状态位都可以用软件置1或清零。用软件将这些位中的某一位置1不会导致器件复位。
2： 如果$\overline{\text{LPCFG}}$配置位为1（未编程），则保持稳压器被禁止且RETEN位不起作用。在休眠期间，保持模式可以保留SRAM内容。
3： 在稳压器进入待机模式后将其重新使能，将在从休眠中唤醒时增加延时T_{VREG}。不使用稳压器的应用应该将此位置1，以防止发生此延时。
4： 如果FWDTEN<1:0>配置位为11（未编程），则WDT总是使能，而与SWDTEN位的设置无关。
5： BOREN<1:0>（FPOR<1:0>）配置位必须设置为01，以使SBOREN生效。

表7-1： 复位标志位操作

标志位	置1事件	清零事件
TRAPR（RCON<15>）	陷阱冲突事件	POR
IOPUWR（RCON<14>）	非法操作码或访问未初始化的W寄存器	POR
CM（RCON<9>）	配置不匹配复位	POR
EXTR（RCON<7>）	$\overline{\text{MCLR}}$复位	POR
SWR（RCON<6>）	RESET指令	POR
WDTO（RCON<4>）	WDT超时	CLRWDT指令、PWRSAV指令以及POR
SLEEP（RCON<3>）	PWRSAV #0指令	POR
IDLE（RCON<2>）	PWRSAV #1指令	POR
BOR（RCON<1>）	POR和BOR	—
POR（RCON<0>）	POR	—

注： 所有复位标志位可由用户软件置1或清零。

7.1 特殊功能寄存器的复位状态

大部分与PIC24F CPU和外设有关的特殊功能寄存器（SFR）会在器件复位时复位为某个特定值。SFR是按其外设或CPU功能分组的，其复位值在本手册的各章节中有所说明。

除了四个寄存器之外，其他SFR的复位值都不受复位类型的影响。复位控制寄存器RCON的复位值取决于器件复位的类型。振荡器控制寄存器OSCCON的复位值取决于复位的类型和FOSCSEL闪存配置字中FNOSC<2:0>位的编程值（见表7-2）。RCFGCAL和NVMCON寄存器只受POR影响。

7.2 器件复位时间

表7-3总结了各种类型的器件复位所需的时间。注意，在POR延时结束后会发出主复位信号$\overline{\mathrm{SYSRST}}$。

器件实际开始执行代码的时间还取决于系统振荡器延时，它包括振荡器起振定时器（OST）延时和PLL锁定时间。OST和PLL锁定时间与相应的$\overline{\mathrm{SYSRST}}$延时同时发生。

故障保护时钟监视器（FSCM）延时决定在$\overline{\mathrm{SYSRST}}$信号发出后FSCM开始监视系统时钟源的时间。

7.3 欠压复位（BOR）

PIC24FJ256GA705系列器件带有一个BOR电路，它将为用户提供多个配置和节能选项。BOR由BOREN<1:0>（FPOR<1:0>）配置位控制。

BOR使能时，只要V_{DD}降至BOR阈值以下，就会导致器件欠压复位。有关欠压复位阈值电压，请参见**第32.1节“直流特性”**。

7.4 复位时的时钟源选择

如果使能了时钟切换，则器件复位时的系统时钟源可按照表7-2进行选择。如果禁止了时钟切换，则总是根据振荡器配置位选择系统时钟源。更多信息，请参见《dsPIC33/PIC24系列参考手册》中的**“振荡器”**（DS39700）。

表7-2： 振荡器选择与复位类型（使能时钟切换）

复位类型	时钟源的确定
POR	FNOSC<2:0>配置位（FOSCSEL<2:0>）
BOR	
$\overline{\mathrm{MCLR}}$	COSC<2:0>控制位（OSCCON<14:12>）
WDTO	
SWR	

表7-3： 各种器件复位的复位延时

复位类型	时钟源	$\overline{\text{SYSRST}}$延时	系统时钟延时	注
POR	EC	TPOR + TSTARTUP + TRST	—	1, 2, 3
	ECPLL	TPOR + TSTARTUP + TRST	TLOCK	1, 2, 3, 5
	XT、HS和SOSC	TPOR + TSTARTUP + TRST	TOST	1, 2, 3, 4
	XTPLL和HSPLL	TPOR + TSTARTUP + TRST	TOST + TLOCK	1, 2, 3, 4, 5
	FRC和OSCFDIV	TPOR + TSTARTUP + TRST	TFRC	1, 2, 3, 6, 7
	FRCPLL	TPOR + TSTARTUP + TRST	TFRC + TLOCK	1, 2, 3, 5, 6
	LPRC	TPOR + TSTARTUP + TRST	TLPRC	1, 2, 3, 6
BOR	EC	TSTARTUP + TRST	—	2, 3
	ECPLL	TSTARTUP + TRST	TLOCK	2, 3, 5
	XT、HS和SOSC	TSTARTUP + TRST	TOST	2, 3, 4
	XTPLL和HSPLL	TSTARTUP + TRST	TOST + TLOCK	2, 3, 4, 5
	FRC和OSCFDIV	TSTARTUP + TRST	TFRC	2, 3, 6, 7
	FRCPLL	TSTARTUP + TRST	TFRC + TLOCK	2, 3, 5, 6
	LPRC	TSTARTUP + TRST	TLPRC	2, 3, 6
$\overline{\text{MCLR}}$	任何时钟	TRST	—	3
WDT	任何时钟	TRST	—	3
软件	任何时钟	TRST	—	3
非法操作码	任何时钟	TRST	—	3
未初始化的W寄存器	任何时钟	TRST	—	3
陷阱冲突	任何时钟	TRST	—	3

注 1： TPOR = 上电复位延时（标称值为10 μs）。
2： TSTARTUP = TVREG。
3： TRST = 内部状态复位时间（标称值为2 μs）。
4： TOST = 振荡器起振定时器（OST）延时。10位计数器计满1024个振荡器周期后，才将振荡器时钟释放给系统使用。
5： TLOCK = PLL锁定时间。
6： TFRC和TLPRC = RC振荡器起振时间。
7： 如果使能了双速启动，无论选择了哪种主振荡器，器件都将使用FRC启动，因此，系统时钟延时即为TFRC，在这些情况下，FRC起振时间有效；器件将在其相应时钟延时之后切换到主振荡器。

7.4.1 POR和长振荡器起振时间

振荡器起振电路及其相关的延时定时器与上电时发生的器件复位延时没有关系。某些晶振电路（尤其是低频晶振）的起振时间会相对较长。因此，在发出$\overline{\text{SYSRST}}$之后，可能会发生以下一种或多种情况：

- 振荡器电路未起振。
- 振荡器起振定时器尚未超时（如果使用了晶振）。
- PLL还未锁定（如果使用了PLL）。

在有效时钟源供系统使用之前，器件不会开始执行代码。因此，如果必须确定复位延时，还须考虑振荡器和PLL起振延时。

7.4.2 故障保护时钟监视器（FSCM）和器件复位

如果使能了FSCM，它将在发出$\overline{\text{SYSRST}}$时开始监视系统时钟源。如果此时没有可用的有效时钟源，器件将会自动切换到FRC振荡器，且用户可在陷阱服务程序（Trap Service Routine，TSR）中将系统时钟源切换到所需的晶振。

8.0 中断控制器

> **注 1:** 本数据手册总结了PIC24FJ256GA705系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24系列参考手册》的"中断"（DS70000600），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。
>
> **2:** 本章中描述的一些寄存器及其相关的位并非在所有器件上都可用。关于具体器件的寄存器和位信息，请参见本数据手册中的**第4.0节"存储器构成"**。

PIC24FJ256GA705系列中断控制器将诸多外设中断请求信号缩减为一个送往PIC24FJ256GA705系列CPU的中断请求信号。

中断控制器具有以下特性：

- 最多8个处理器异常和软件陷阱
- 7个可由用户选择的优先级
- 中断向量表（IVT），其中每个中断或异常源对应惟一的向量
- 在指定的用户优先级内具有固定的优先级
- 固定的中断进入和返回延时

8.1 中断向量表

PIC24FJ256GA705系列的中断向量表（IVT）（如图8-1所示）位于程序存储器中，起始单元地址是000004h。IVT包含6个不可屏蔽陷阱向量和最多118个中断源。一般来说，每个中断源都有自己的中断向量。每个中断向量都包含一个24位宽的地址。每个中断向量单元中编程的值是其相关的中断服务程序（ISR）的起始地址。

中断向量根据其自然优先级区分优先顺序。自然优先级与中断向量在向量表中的位置有关。一般来说，较低地址的中断向量具有较高的自然优先级。例如，与向量0相关的中断比任何其他向量地址的中断具有更高的自然优先级。

8.1.1 备用中断向量表

如图8-1所示，备用中断向量表（AIVT）位于IVT之后。AIVTEN（INTCON2<8>）控制位提供对AIVT的访问。如果AIVTEN位置1，所有中断和异常处理将使用备用向量而不是默认的向量。备用向量与默认向量的构成方式相同。

AIVT支持仿真和调试功能，它提供了一种不需要将中断向量再编程就可以在应用程序和支持环境之间切换的方法。此特性也支持运行时在不同应用程序之间切换以便评估各种不同的软件算法。如果不需要AIVT，应该用IVT中使用的相同地址编程AIVT。

8.2 复位序列

由于复位过程中不涉及到中断控制器，所以器件复位并不是真的异常。作为对复位的响应，PIC24FJ256GA705系列器件清零其寄存器，强制PC为零。然后器件从地址0x000000处开始执行程序。用户可以在复位地址处编程一条`GOTO`指令，将程序执行重定向到相应的启动程序。

> **注：** 应使用包含`RESET`指令的默认中断处理程序的地址编程IVT中所有未实现或未使用的向量单元。

图8-1：　PIC24F中断向量表

中断向量表（IVT）(1)

自然优先级降序排列

向量	地址
复位——GOTO指令	000000h
复位——GOTO地址	000002h
振荡器故障陷阱向量	000004h
地址错误陷阱向量	
通用硬陷阱向量	
堆栈错误陷阱向量	
数学错误陷阱向量	
保留	
通用软陷阱向量	
保留	
中断向量0	000014h
中断向量1	
—	
—	
中断向量52	00007Ch
中断向量53	00007Eh
中断向量54	000080h
—	
—	
中断向量116	0000FCh
中断向量117	0000FEh

备用中断向量表（AIVT）(1,2)

向量	地址
保留	BOA+00h
保留	BOA+02h
振荡器故障陷阱向量	BOA+04h
地址错误陷阱向量	
通用硬陷阱向量	
堆栈错误陷阱向量	
数学错误陷阱向量	
保留	
通用软陷阱向量	
保留	
中断向量0	BOA+14h
中断向量1	
—	
—	
中断向量52	BOA+7Ch
中断向量53	BOA+7Eh
中断向量54	BOA+80h
—	
—	
中断向量116	
中断向量117	BOA+FEh
（代码起始单元）	（BOA+100h）

图注： BOA：AIVT的基准偏移地址，该地址是引导段最后一页的起始地址。所有地址均采用十六进制格式。

注 1： 有关中断向量列表的信息，请参见表8-2。

2： AIVT仅在实现引导段的情况下可用。

表8-1：　陷阱向量详细信息

向量编号	IVT地址	AIVT地址	陷阱源
0	000004h	BOA+04h	振荡器故障
1	000006h	BOA+06h	地址错误
2	000008h	BOA+08h	一般硬件错误
3	00000Ah	BOA+0Ah	堆栈错误
4	00000Ch	BOA+0Ch	数学错误
5	00000Eh	BOA+0Eh	保留
6	000010h	BOA+10h	一般软件错误
7	000012h	BOA+12h	保留

图注： BOA = AIVT段的基准偏移地址，该地址是引导段最后一页的起始地址。

表**8-2**： 中断向量详细信息

中断源	IRQ编号	IVT地址	中断位的位置		
			标志	使能	优先级
最高自然顺序优先级					
INT0——外部中断0	0	000014h	IFS0<0>	IEC0<0>	INT0Interrupt
IC1——输入捕捉1	1	000016h	IFS0<1>	IEC0<1>	IC1Interrupt
OC1——输出比较1	2	000018h	IFS0<2>	IEC0<2>	OC1Interrupt
T1——Timer1	3	00001Ah	IFS0<3>	IEC0<3>	T1Interrupt
DMA0——直接存储器访问0	4	00001Ch	IFS0<4>	IEC0<4>	DMA0Interrupt
IC2——输入捕捉2	5	00001Eh	IFS0<5>	IEC0<5>	IC2Interrupt
OC2——输出比较2	6	000020h	IFS0<6>	IEC0<6>	OC2Interrupt
T2——Timer2	7	000022h	IFS0<7>	IEC0<7>	T2Interrupt
T3——Timer3	8	000024h	IFS0<8>	IEC0<8>	T3Interrupt
SPI1——SPI1通用	9	000026h	IFS0<9>	IEC0<9>	SPI1Interrupt
SPI1TX——SPI1传输完成	10	000028h	IFS0<10>	IEC0<10>	SPI1TXInterrupt
U1RX——UART1接收器	11	00002Ah	IFS0<11>	IEC0<11>	U1RXInterrupt
U1TX——UART1发送器	12	00002Ch	IFS0<12>	IEC0<12>	U1TXInterrupt
ADC1——A/D转换器1	13	00002Eh	IFS0<13>	IEC0<13>	ADC1Interrupt
DMA1——直接存储器访问1	14	000030h	IFS0<14>	IEC0<14>	DMA1Interrupt
NVM——NVM编程/擦除完成	15	000032h	IFS0<15>	IEC0<15>	NVMInterrupt
SI2C1——I2C1从事件	16	000034h	IFS1<0>	IEC1<0>	SI2C1Interrupt
MI2C1——2C1主事件	17	000036h	IFS1<1>	IEC1<1>	MI2C1Interrupt
Comp——比较器	18	000038h	IFS1<2>	IEC1<2>	CompInterrupt
IOC——电平变化中断	19	00003Ah	IFS1<3>	IEC1<3>	IOCInterrupt
INT1——外部中断1	20	00003Ch	IFS1<4>	IEC1<4>	INT1Interrupt
—	21	—	—	—	—
—	22	—	—	—	—
—	23	—	—	—	—
DMA2——直接存储器访问2	24	000044h	IFS1<8>	IEC1<8>	DMA2Interrupt
OC3——输出比较3	25	000046h	IFS1<9>	IEC1<9>	OC3Interrupt
—	26	—	—	—	—
—	27	—	—	—	—
—	28	—	—	—	—
INT2——外部中断2	29	00004Eh	IFS1<13>	IEC1<13>	INT2Interrupt
U2RX——UART2接收器	30	000050h	IFS1<14>	IEC1<14>	U2RXInterrupt
U2TX——UART2发送器	31	000052h	IFS1<15>	IEC1<15>	U2TXInterrupt
SPI2——SPI2通用	32	000054h	IFS2<0>	IEC2<0>	SPI2Interrupt
SPI2TX——SPI2传输完成	33	000056h	IFS2<1>	IEC2<1>	SPI2TXInterrupt
—	34	—	—	—	—
—	35	—	—	—	—
DMA3——直接存储器访问3	36	00005Ch	IFS2<4>	IEC2<4>	DMA3Interrupt
IC3——输入捕捉3	37	00005Eh	IFS2<5>	IEC2<5>	IC3Interrupt
—	38	—	—	—	—
—	39	—	—	—	—
—	40	—	—	—	—
CCT3——捕捉/比较定时器3	43	00006Ah	IFS2<11>	IEC2<11>	CCT3Interrupt

表8-2： 中断向量详细信息（续）

中断源	IRQ编号	IVT地址	中断位的位置		
			标志	使能	优先级
PMP——并行主端口	45	00006Eh	IFS2<13>	IEC2<13>	PMPInterrupt
DMA4——直接存储器访问4	46	000070h	IFS2<14>	IEC2<14>	DMA4Interrupt
—	47	—	—	—	—
—	48	—	—	—	—
SI2C2——I2C2从事件	49	000076h	IFS3<1>	IEC3<1>	SI2C2Interrupt
MI2C2——I2C2主事件	50	000078h	IFS3<2>	IEC3<2>	MI2C2Interrupt
—	51	—	—	—	—
—	52	—	—	—	—
INT3——外部中断3	53	00007Eh	IFS3<5>	IEC3<5>	INT3Interrupt
INT4——外部中断4	54	000080h	IFS3<6>	IEC3<6>	INT4Interrupt
—	55	—	—	—	—
—	56	—	—	—	—
—	57	—	—	—	—
SPI1RX——SPI1接收完成	58	000088h	IFS3<10>	IEC3<10>	SPI1RXInterrupt
SPI2RX——SPI2接收完成	59	00008Ah	IFS3<11>	IEC3<11>	SPI2RXInterrupt
SPI3RX——SPI3接收完成	60	00008Ch	IFS3<12>	IEC3<12>	SPI3RXInterrupt
DMA5——直接存储器访问5	61	00008Eh	IFS3<13>	IEC3<13>	DMA5Interrupt
RTCC——实时时钟和日历	62	000090h	IFS3<14>	IEC3<14>	RTCCInterrupt
CCP1——捕捉/比较1	63	000092h	IFS3<15>	IEC3<15>	CCP1Interrupt
CCP2——捕捉/比较2	64	000094h	IFS4<0>	IEC4<0>	CCP2Interrupt
U1E——UART1错误	65	000096h	IFS4<1>	IEC4<1>	U1EInterrupt
U2E——UART2错误	66	000098h	IFS4<2>	IEC4<2>	U2EInterrupt
CRC——循环冗余校验	67	00009Ah	IFS4<3>	IEC4<3>	CRCInterrupt
—	68	—	—	—	—
—	69	—	—	—	—
—	70	—	—	—	—
—	71	—	—	—	—
HLVD——高/低压检测	72	0000A4h	IFS4<8>	IEC4<8>	HLVDInterrupt
—	73	—	—	—	—
—	74	—	—	—	—
—	75	—	—	—	—
—	76	—	—	—	—
CTMU——中断	77	0000AEh	IFS4<13>	IEC4<13>	CTMUInterrupt
—	78	—	—	—	—
—	79	—	—	—	—
—	80	—	—	—	—
—	81	—	—	—	—
—	82	—	—	—	—
—	83	—	—	—	—

表8-2： 中断向量详细信息（续）

中断源	IRQ 编号	IVT地址	中断位的位置		
			标志	使能	优先级
I2C1BC——I2C1总线冲突	84	0000BCh	IFS5<4>	IEC5<4>	I2C1BCInterrupt
I2C2BC——I2C2总线冲突	85	0000BEh	IFS5<5>	IEC5<5>	I2C2BCInterrupt
—	86	—	—	—	—
—	87	—	—	—	—
—	88	—	—	—	—
—	89	—	—	—	—
SPI3——SPI3通用	90	0000C8h	IFS5<10>	IEC5<10>	SPI3Interrupt
SPI3TX——SPI3传输完成	91	0000CAh	IFS5<11>	IEC5<11>	SPI3TXInterrupt
—	92	92	—	—	—
—	93	93	—	—	—
CCP3——捕捉/比较3	94	0000D0h	IFS5<14>	IEC5<14>	CCP3Interrupt
CCP4——捕捉/比较4	95	0000D2h	IFS5<15>	IEC5<15>	CCP4Interrupt
CLC1——可配置逻辑单元1	96	0000D4h	IFS6<0>	IEC6<0>	CLC1Interrupt
CLC2——可配置逻辑单元2	97	0000D6h	IFS6<1>	IEC6<1>	CLC2Interrupt
—	98	—	—	—	—
—	99	—	—	—	—
—	100	—	—	—	—
CCT1——捕捉/比较定时器1	101	0000DEh	IFS6<5>	IEC6<5>	CCT1Interrupt
CCT2——捕捉/比较定时器2	102	0000E0h	IFS6<6>	IEC6<6>	CCT2Interrupt
—	103	—	—	—	—
—	104	—	—	—	—
—	105	—	—	—	—
FST——FRC自调整中断	106	0000E8h	IFS6<10>	IEC6<10>	FSTInterrupt
—	107	—	—	—	—
ECCIE——ECC单位错误	108	0000ECh	IFS6<12>	IEC6<12>	ECCIEInterrupt
—	109	—	—	—	—
RTCCTS——实时时钟时间戳	110	0000F0h	IFS6<14>	IEC6<14>	RTCCTSInterrupt
—	111	—	—	—	—
—	112	—	—	—	—
—	113	—	—	—	—
—	114	—	—	—	—
—	115	—	—	—	—
—	116	—	—	—	—
JTAG——JTAG	117	0000FEh	IFS7<5>	IEC7<5>	JTAGInterrupt

8.3 中断资源

对于本数据手册中列出的器件，Microchip网站的主产品页面中提供了许多有用的资源。该产品页面可使用此链接进行访问，其中包含最新更新和附加信息。

注：	如果您无法使用上述链接访问产品页面，请在浏览器中输入此URL：http://www.microchip.com/wwwproducts/en/pic24fj256ga705

8.3.1 主要资源

- 《dsPIC33/PIC24系列参考手册》中的 **"中断"**（DS70000600）
- 代码示例
- 应用笔记
- 软件库
- 网上研讨会
- 所有相关的《dsPIC33/PIC24系列参考手册》章节
- 开发工具

8.4 中断控制和状态寄存器

PIC24FJ256GA705系列器件实现了以下用于中断控制器的寄存器：

- INTCON1
- INTCON2
- INTCON4
- IFS0-IFS7
- IEC0-IEC7
- IPC0-ICP29
- INTTREG

8.4.1 INTCON1至INTCON4

INTCON1和INTCON2控制全局中断。INTCON1包含中断嵌套禁止（NSTDIS）位以及处理器陷阱源的控制和状态标志。

INTCON2寄存器控制全局中断的产生、外部中断请求信号行为和备用中断向量表（AIVT）的使用。

INTCON4寄存器包含软件生成的硬陷阱（Software Generated Hard Trap，SGHT）位和ECC双位错误（ECC Double-Bit Error， ECCDBE）陷阱。

8.4.2 IFSx

IFSx寄存器维护所有中断请求标志。每个中断源都具有一个中断标志状态位，该状态位由相应的外设中断或外部中断信号置1，通过软件进行清零。

8.4.3 IECx

IECx寄存器维护所有中断允许位。这些控制位用于单独允许外设中断或外部中断信号。

8.4.4 IPCx

IPCx寄存器用于设置每个中断源的中断优先级（IPL）。可以给每个用户中断源分配8个优先级之一。

8.4.5 INTTREG

INTTREG寄存器包含相关的中断向量编号和新的CPU中断优先级，分别锁存在INTTREG寄存器中的向量编号（VECNUM<7:0>）和中断优先级（ILR<3:0>）位域中。新的中断优先级是等待处理中断的优先级。

中断源按表8-2中的顺序分配给IFSx、IECx和IPCx寄存器。例如，INT0（外部中断0）向量编号为8、自然顺序优先级为0。所以，INT0IF位在IFS0<0>中，INT0IE位在IEC0<0>中，INT0IPx位在IPC0的第一个位置（IPC0<2:0>）中。

8.4.6 状态/控制寄存器

尽管这些寄存器不是中断控制硬件的特定组成部分，但其中两个CPU控制寄存器包含控制中断功能的位。关于这些寄存器的详细信息，请参见《dsPIC33/PIC24系列参考手册》的 **"带扩展数据空间（EDS）的CPU"**（DS39732）。

- CPU状态寄存器SR包含IPL<2:0>位（SR<7:5>）。这些位指示当前CPU中断优先级。用户软件可以通过写IPLx位来更改当前CPU中断优先级。
- CORCON寄存器包含IPL3位，此位与IPL<2:0>位一起指示当前CPU中断优先级。IPL3是只读位，所以用户软件不能屏蔽陷阱事件。

下面各页中的寄存器8-3至寄存器8-6介绍了所有中断寄存器。

寄存器8-1： SR：ALU状态寄存器[1]

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	DC
bit 15							bit 8

R/W-0[3]	R/W-0[3]	R/W-0[3]	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2[2]	IPL1[2]	IPL0[2]	RA	N	OV	Z	C
bit 7							bit 0

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

bit 7-5 **IPL<2:0>**：CPU中断优先级状态位[2,3]
`111` = CPU中断优先级为7（15）；禁止用户中断
`110` = CPU中断优先级为6（14）
`101` = CPU中断优先级为5（13）
`100` = CPU中断优先级为4（12）
`011` = CPU中断优先级为3（11）
`010` = CPU中断优先级为2（10）
`001` = CPU中断优先级为1（9）
`000` = CPU中断优先级为0（8）

注 1： 如需了解整个寄存器的详细信息，请参见寄存器3-1。
2： IPL<2:0>状态位与IPL3状态位（CORCON<3>）组合形成CPU中断优先级（IPL）。如果IPL3 = `1`，则括号中的值表示IPL。当IPL3 = `1`时，禁止用户中断。
3： 当NSTDIS位（INTCON1<15>）= `1`时，IPL<2:0>状态位是只读的。

寄存器8-2： **CORCON：CPU内核控制寄存器[1]**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/C-0	R/W-1	U-0	U-0
—	—	—	—	IPL3[2]	PSV	—	—
bit 7							bit 0

图注：	C = 可清零位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15-4 **未实现：**读为0

bit 3 **IPL3：**CPU中断优先级状态位[2]
1 = CPU中断优先级大于7
0 = CPU中断优先级小于或等于7

bit 2 **PSV：**未用作中断模块的一部分

bit 1-0 **未实现：**读为0

注 1： 如需了解整个寄存器的详细信息，请参见寄存器3-2。
2： IPL<2:0>状态位与IPL3状态位（CORCON<3>）组合形成CPU中断优先级（IPL）。如果IPL3 = 1，则括号中的值表示IPL。当IPL3 = 1时，禁止用户中断。

寄存器8-3： INTCON1：中断控制寄存器1

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
NSTDIS	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—
bit 7							bit 0

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

bit 15 **NSTDIS：**中断嵌套禁止位
1 = 禁止中断嵌套
0 = 使能中断嵌套

bit 14-5 **未实现：**读为0

bit 4 **MATHERR：**数学错误状态位
1 = 发生了数学错误陷阱
0 = 未发生数学错误陷阱

bit 3 **ADDRERR：**地址错误陷阱状态位
1 = 发生了地址错误陷阱
0 = 未发生地址错误陷阱

bit 2 **STKERR：**堆栈错误陷阱状态位
1 = 发生了堆栈错误陷阱
0 = 未发生堆栈错误陷阱

bit 1 **OSCFAIL：**振荡器故障陷阱状态位
1 = 发生了振荡器故障陷阱
0 = 未发生振荡器故障陷阱

bit 0 **未实现：**读为0

寄存器8-4： INTCON2：中断控制寄存器2

R/W-1	R-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
GIE	DISI	SWTRAP	—	—	—	—	AIVTEN
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP
bit 7							bit 0

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

bit 15 **GIE**：全局中断允许位
1 = 允许中断并将相关的中断允许位置1
0 = 禁止中断，但仍然允许陷阱

bit 14 **DISI**：DISI指令状态位
1 = DISI指令有效
0 = DISI指令无效

bit 13 **SWTRAP**：软件陷阱状态位
1 = 允许软件陷阱
0 = 禁止软件陷阱

bit 12-9 **未实现**：读为0

bit 8 **AIVTEN**：备用中断向量表使能位
1 = 使用备用中断向量表（如果已在配置位中使能）
0 = 使用标准中断向量表（默认）

bit 7-5 **未实现**：读为0

bit 4 **INT4EP**：外部中断4边沿检测极性选择位
1 = 下降沿中断
0 = 上升沿中断

bit 3 **INT3EP**：外部中断3边沿检测极性选择位
1 = 下降沿中断
0 = 上升沿中断

bit 2 **INT2EP**：外部中断2边沿检测极性选择位
1 = 下降沿中断
0 = 上升沿中断

bit 1 **INT1EP**：外部中断1边沿检测极性选择位
1 = 下降沿中断
0 = 上升沿中断

bit 0 **INT0EP**：外部中断0边沿检测极性选择位
1 = 下降沿中断
0 = 上升沿中断

寄存器8-5： INTCON4：中断控制寄存器4

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/C-0	R/C-0
—	—	—	—	—	—	ECCDBE	SGHT
bit 7							bit 0

图注：	C = 可清零位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15-2 **未实现：**读为0

bit 1 **ECCDBE：**ECC双位错误陷阱位
1 = 发生了ECC双位错误陷阱
0 = 未发生ECC双位错误陷阱

bit 0 **SGHT：**软件生成的硬陷阱状态位
1 = 发生了软件生成的硬陷阱
0 = 未发生软件生成的硬陷阱

寄存器8-6： **INTTREG：中断控制和状态寄存器**

R-0	U-0	R/W-0	U-0	R-0	R-0	R-0	R-0
CPUIRQ	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
VECNUM7	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0
bit 7							bit 0

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

bit 15 **CPUIRQ**：从中断控制器至CPU的中断请求响应位
1 = 产生了中断请求，但CPU尚未响应；这发生在CPU的优先级高于所请求的中断优先级时
0 = 没有未响应的中断请求

bit 14 **未实现**：读为0

bit 13 **VHOLD**：向量编号捕捉配置位
1 = VECNUMx位保存优先级最高的待处理中断的值
0 = VECNUMx位保存上一次应答的中断的值（即已发生且优先级高于CPU的上一个中断，即使有其他待处理中断时也是如此）

bit 12 **未实现**：读为0

bit 11-8 **ILR<3:0>**：新的CPU中断优先级位
1111 = CPU中断优先级为15
•
•
•
0001 = CPU中断优先级为1
0000 = CPU中断优先级为0

bit 7-0 **VECNUM<7:0>**：待处理中断的向量编号位
11111111 = 255，保留；不要使用
•
•
•
00001001 = 9，IC1——输入捕捉1
00001000 = 8，INT0——外部中断0
00000111 = 7，保留；不要使用
00000110 = 6，通用软陷阱
00000101 = 5，保留；不要使用
00000100 = 4，数学错误陷阱
00000011 = 3，堆栈错误陷阱
00000010 = 2，通用硬陷阱
00000001 = 1，地址错误陷阱
00000000 = 0，振荡器故障陷阱

9.0 振荡器配置

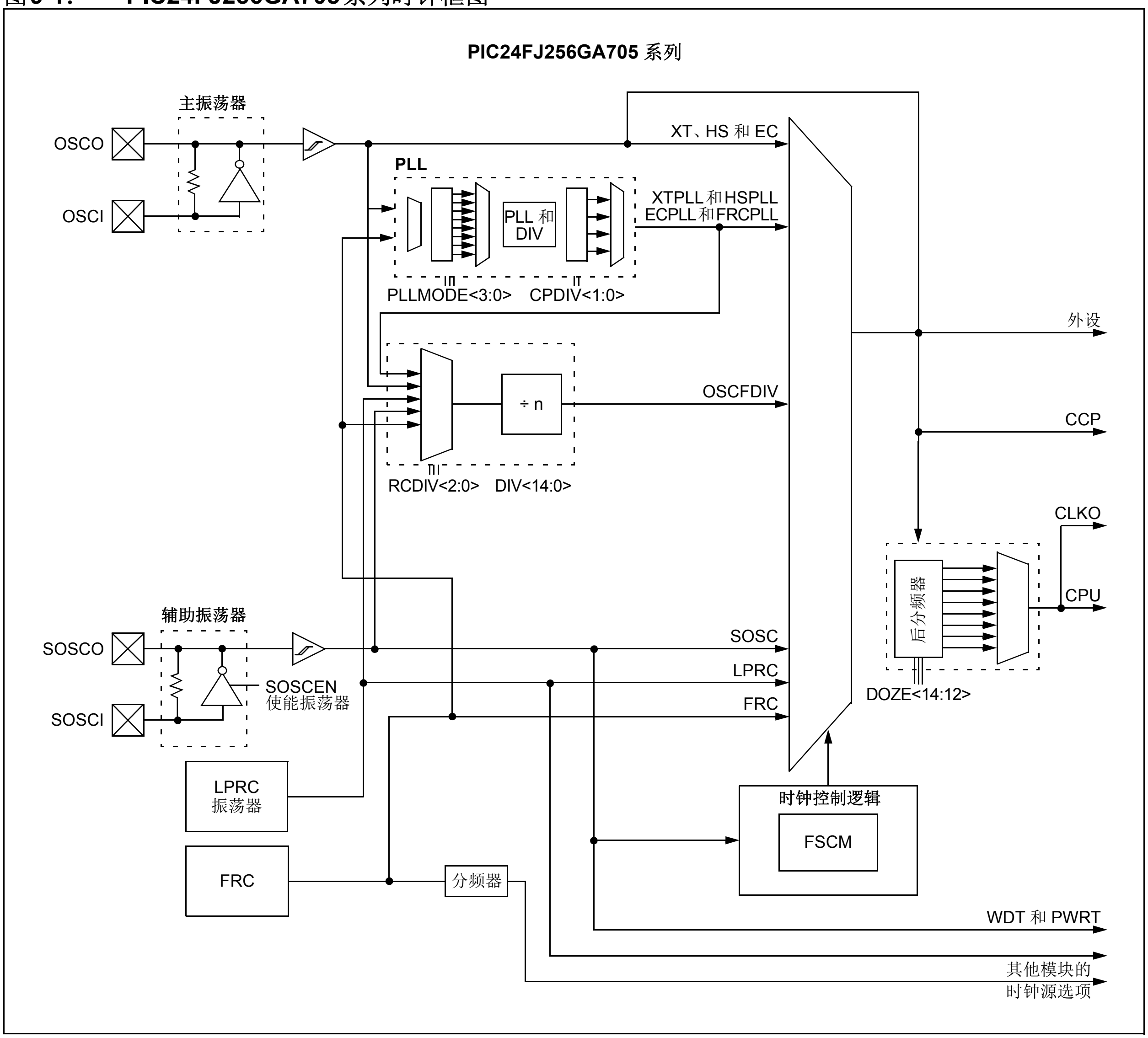
> **注：** 本数据手册总结了该组PIC24F器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的“振荡器”（DS39700），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

PIC24FJ256GA705系列器件的振荡器系统具有以下特性：

- 片上PLL模块可为系统时钟提供一系列频率选项
- 可采用软件控制在多个时钟源之间切换
- 由软件控制的后分频器对 CPU 的时钟进行选择以节省系统的功耗
- 具有故障保护时钟监视器（FSCM），可检测时钟故障，以使应用可安全地恢复或关闭
- 可单独使用且独立配置的系统时钟输出，以同步外部硬件

图9-1给出了振荡器系统的简化框图。

图9-1： PIC24FJ256GA705系列时钟框图



9.1 CPU时钟机制

系统时钟源可由以下四种时钟源之一提供：

- OSCI和OSCO引脚上的主振荡器（POSC）
- SOSCI和SOSCO引脚上的辅助振荡器（SOSC）
- 内部快速RC（FRC）振荡器
- 内部低功耗RC（LPRC）振荡器

主振荡器和FRC源可使用内部PLL模块，从而能够生成4x、6x或8x PLL时钟。如果使用了PLL，则必要时可对PLL时钟进行后分频并用作系统时钟。更多信息，请参见**第9.5节“振荡器模式”**。内部FRC提供8 MHz时钟源。

每个时钟源（PRIPLL、FRCPLL、PRI、FRC、LPRC和SOSC）均可用作附加分频器的输入，然后可以使用附加分频器产生一个分频时钟源，以用作系统时钟（OSCFDIV）。

选择的时钟源提供处理器和外设的时钟。将处理器时钟源二分频可以产生内部指令周期时钟FCY。在本文档中，指令周期时钟还由FOSC/2表示。在OSCO I/O引脚上可以提供内部指令周期时钟FOSC/2，用于主振荡器的某些工作模式。

9.2 上电复位时的初始配置

通过使用配置位设置选择器件发生上电复位事件时使用的振荡器源（和工作模式）。振荡器配置位设置位于程序存储器中的配置寄存器中（更多详细信息，请参见**第29.1节“配置位”**）。主振荡器配置位POSCMD<1:0>（FOSC<1:0>）和振荡器选择配置位 FNOSC<2:0>（FOSCSEL<2:0>）选择上电复位时使用的振荡器源。OSCFDIV时钟源为默认（未编程）选择；OSCFDIV分频器的默认输入源为FRC时钟源。可通过编程这些位来选择其他振荡器。

配置位允许用户在多个时钟模式之间进行选择（见表9-1）。

9.2.1 时钟切换模式配置位

FCKSM<1:0>配置位（FOSC<7:6>）用于联合配置器件时钟切换和故障保护时钟监视器（FSCM）。只有将FCKSM1编程为0时才可以使能时钟切换功能。只有将FCKSM<1:0>编程为00时才可以使能FSCM。

表9-1： 用于时钟选择的配置位值

振荡器模式	振荡器源	POSCMD<1:0>	FNOSC<2:0>	注
带分频的振荡器（OSCFDIV）	内部/外部	11	111	1, 2, 3
低功耗RC振荡器（LPRC）	内部	11	101	3
辅助（Timer1）振荡器（SOSC）	辅助	11	100	3
带有PLL模块的主振荡器（XT）（XTPLL）	主	01	011	
带有PLL模块的主振荡器（EC）（ECPLL）	主	00	011	
主振荡器（HS）	主	10	010	
主振荡器（XT）	主	01	010	
主振荡器（EC）	主	00	010	
带有PLL模块的快速RC振荡器（FRCPLL）	内部	11	001	3
快速RC振荡器（FRC）	内部	11	000	3

注 1： OSCFDIV时钟模式的输入振荡器由RCDIV<2:0>（CLKDIV<10:8>）位决定。上电复位时，默认值会选择FRC模块。

2： 这是未编程（已擦除）器件的默认振荡器模式。

3： OSCO引脚功能由OSCIOFCN配置位决定。

9.3 控制寄存器

振荡器的操作由五个特殊功能寄存器控制：

- OSCCON
- CLKDIV
- OSCTUN
- OSCDIV
- OSCFDIV

OSCCON寄存器（寄存器9-1）是振荡器的主控制寄存器。它控制时钟源的切换并允许监视时钟源。OSCCON受写锁定保护，以防止意外时钟切换。更多信息，请参见第9.4节“时钟切换操作”。

CLKDIV寄存器（寄存器9-2）控制与打盹模式相关的特性，以及用于OSCFDIV时钟模式的后分频器和PLL模块。

OSCTUN寄存器（寄存器9-3）允许用户在大约±1.5%的范围内精确地调整FRC振荡器。

OSCDIV和OSCFDIV寄存器提供对系统振荡器分频器的控制。

寄存器9-1： OSCCON：振荡器控制寄存器[1]

U-0	R-x[2]	R-x[2]	R-x[2]	U-0	R/W-x[2]	R/W-x[2]	R/W-x[2]
—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0
bit 15							bit 8

R/W-0	R/W-0	R-0[4]	U-0	R/CO-0	R/W-0	R/W-0	R/W-0
CLKLOCK	IOLOCK[3]	LOCK	—	CF	POSCEN	SOSCEN	OSWEN
bit 7							bit 0

图注：	CO = 只可清零位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15 **未实现：**读为0

bit 14-12 **COSC<2:0>：**当前振荡器选择位[2]
111 = 带分频器的振荡器（OSCFDIV）
110 = 保留
101 = 低功耗RC振荡器（LPRC）
100 = 辅助振荡器（SOSC）
011 = 带PLL模块的主振荡器（XTPLL和ECPLL）
010 = 主振荡器（XT、HS和EC）
001 = 带PLL模块的快速RC振荡器（FRCPLL）
000 = 快速RC振荡器（FRC）

bit 11 **未实现：**读为0

bit 10-8 **NOSC<2:0>：**新振荡器选择位[2]
111 = 带分频器的振荡器（OSCFDIV）
110 = 保留
101 = 低功耗RC振荡器（LPRC）
100 = 辅助振荡器（SOSC）
011 = 带PLL模块的主振荡器（XTPLL和ECPLL）
010 = 主振荡器（XT、HS和EC）
001 = 带PLL模块的快速RC振荡器（FRCPLL）
000 = 快速RC振荡器（FRC）

bit 7 **CLKLOCK：**时钟选择锁定使能位
如果使能了FSCM（FCKSM<1:0> = 00）：
1 = 时钟和PLL选择被锁定
0 = 时钟和PLL选择不锁定，可通过将OSWEN位置1来进行修改
如果禁止了FSCM（FCKSM<1:0> = 1x）：
时钟和PLL选择始终不锁定，可通过将OSWEN位置1来进行修改

bit 6 **IOLOCK：**I/O锁定使能位[3]
1 = I/O锁定有效
0 = I/O锁定无效

bit 5 **LOCK：**PLL锁定状态位[4]
1 = PLL模块处于锁定状态或PLL模块的起振定时器延时结束
0 = PLL模块不处于锁定状态，PLL起振定时器正在运行或PLL被禁止

注 1： OSCCON受写锁定保护，以防止意外时钟切换。更多信息，请参见**第9.4节“时钟切换操作”**。
2： 这些位的复位值由FNOSCx配置位决定。
3： 只有在执行解锁序列后才能更改IOLOCK位的状态。另外，如果IOL1WAY配置位为1，一旦IOLOCK位置1，就不能清零。
4： 在进行有效的时钟切换或选择非PLL时钟模式时，此位也复位为0。

寄存器9-1： OSCCON：振荡器控制寄存器[1]（续）

bit 4	**未实现：**读为0
bit 3	**CF：**时钟故障检测位 1 = FSCM已检测到时钟故障 0 = 未检测到时钟故障
bit 2	**POSCEN：**主振荡器休眠使能位 1 = 主振荡器在休眠模式期间继续工作 0 = 主振荡器在休眠模式期间被禁止
bit 1	**SOSCEN：**32 kHz辅助振荡器（SOSC）使能位 1 = 使能辅助振荡器 0 = 禁止辅助振荡器
bit 0	**OSWEN：**振荡器切换使能位 1 = 启动振荡器切换，切换到由NOSC<2:0>位指定的时钟源 0 = 完成振荡器切换

注 1： OSCCON受写锁定保护，以防止意外时钟切换。更多信息，请参见**第9.4节“时钟切换操作”**。
2： 这些位的复位值由FNOSCx配置位决定。
3： 只有在执行解锁序列后才能更改IOLOCK位的状态。另外，如果IOL1WAY配置位为1，一旦IOLOCK位置1，就不能清零。
4： 在进行有效的时钟切换或选择非PLL时钟模式时，此位也复位为0。

寄存器9-2： **CLKDIV：时钟分频器寄存器**

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
ROI	DOZE2	DOZE1	DOZE0	DOZEN(1)	RCDIV2	RCDIV1	RCDIV0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
CPDIV1	CPDIV0	PLLEN	—	—	—	—	—
bit 7							bit 0

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

bit 15 **ROI：**中断恢复位
1 = 中断时清零DOZEN位并将CPU和外设的时钟频率比复位为1:1
0 = 中断对DOZEN位没有影响

bit 14-12 **DOZE<2:0>：**CPU/外设的时钟频率比选择位
111 = 1:128
110 = 1:64
101 = 1:32
100 = 1:16
011 = 1:8（默认值）
010 = 1:4
001 = 1:2
000 = 1:1

bit 11 **DOZEN：**打盹使能位(1)
1 = DOZE<2:0>位指定CPU和外设时钟频率比
0 = 将CPU和外设的时钟频率比设置为1:1

bit 10-8 **RCDIV<2:0>：**系统分频器时钟源选择位
111 = 保留；不要使用
110 = 保留
101 = 低功耗RC振荡器（LPRC）
100 = 辅助振荡器（SOSC）
011 = 带PLL模块的主振荡器（XT、HS和EC）（XTPLL、HSPLL和ECPLL）
010 = 主振荡器（XT、HS和EC）
001 = 带PLL模块的快速RC振荡器（FRC）（FRCPLL）
000 = 快速RC振荡器（FRC）

bit 7-6 **CPDIV<1:0>：**系统时钟选择位（PLL 32 MHz时钟分支的后分频比选择）
11 = 4 MHz（8分频）
10 = 8 MHz（4分频）
01 = 16 MHz（2分频）
00 = 32 MHz（1分频）

bit 5 **PLLEN：**PLL使能位
1 = PLL始终有效
0 = PLL只有在选择PLL振荡器模式（OSCCON<14:12> = 011或001）时有效

bit 4-0 **未实现：**读为0

注 1： 当ROI位置1并发生中断时，该位会自动清零。

寄存器9-3： OSCTUN：FRC振荡器调节寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN<5:0>					
bit 7							bit 0

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

bit 15-6 **未实现：**读为0

bit 5-0 **TUN<5:0>：**FRC振荡器调节位

011111 = 最大频率偏差
011110 =
•
•
•
000001 =
000000 = 中心频率，振荡器运行在经过出厂前校准的频率上
111111 =
•
•
•
100001 =
100000 = 最小频率偏差

寄存器9-4： OSCDIV：振荡器分频比寄存器

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	DIV<14:8>						
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
DIV<7:0>							
bit 7							bit 0

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

bit 15 **未实现：**读为0

bit 14-0 **DIV<14:0>：**参考时钟分频比位
指定参考时钟的1/2周期（参考时钟源时钟数）
（例如：ref_clk_output周期 = [参考时钟源 * 2] * DIV<14:0>）。
111111111111111 = 振荡器频率65,534（32,767 * 2）分频
111111111111110 = 振荡器频率65,532（32,766 * 2）分频
•
•
•
000000000000011 = 振荡器频率6（3 * 2）分频
000000000000010 = 振荡器频率4（2 * 2）分频
000000000000001 = 振荡器频率2（1 * 2）分频（默认值）
000000000000000 = 振荡器频率不变（不分频）

寄存器9-5：　OSCFDIV：振荡器小数分频比寄存器[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TRIM<0:7>							
bit 15							bit 8

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
TRIM8	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15-7　**TRIM<0:8>**：微调位

为振荡器时钟的1/2周期提供加到DIV<14:0>位值中的小数。

`0000_0000_0` = 将0/512（0.0）分频值加到DIVx值中
`0000_0000_1` = 将1/512（0.001953125）分频值加到DIVx值中
`0000_0001_0` = 将2/512（0.00390625）分频值加到DIVx值中
•
•
•
`100000000` = 将256/512（0.5000）分频值加到DIVx值中
•
•
•
`1111_1111_0` = 将510/512（0.99609375）分频值加到DIVx值中
`1111_1111_1` = 将511/512（0.998046875）分频值加到DIVx值中

bit 6-0　**未实现**：读为0

注 1：　大于零的TRIMx值只有在DIVx值大于零时才有效。

9.4 时钟切换操作

应用可在软件控制下随时在四个时钟源（POSC、SOSC、FRC和LPRC）间自由切换，几乎没有什么限制。为限制该灵活性可能带来的负面影响，PIC24F器件在时钟切换过程中采用了安全锁定。

> **注：** 主振荡器模式有三种不同的子模式（XT、HS和EC），这三种子模式由POSCMDx配置位决定。在应用中可以用软件实现从主振荡器模式切换到其他模式，或从其他模式切换到主振荡器模式，但不能在不对器件进行再编程的情况下在主振荡器模式的不同子模式之间进行切换。

9.4.1 使能时钟切换

要使能时钟切换，必须将FOSC中的FCKSM1配置位编程为0。（更多详细信息，请参见**第29.1节“配置位”**。）如果没有对FCKSM1配置位编程（1），则时钟切换功能和故障保护时钟监视器功能都将被禁止；这是默认设置。

在时钟切换被禁止的情况下，NOSCx控制位（OSCCON<10:8>）不控制时钟选择。但是，COSC<2:0>位（OSCCON<14:12>）将会反映FNOSCx配置位选择的时钟源。

禁止了时钟切换功能时，OSWEN控制位（OSCCON<0>）不起作用；它始终保持为0。

9.4.2 振荡器切换序列

执行时钟切换至少需要下列基本序列：

1. 如果需要，读COSCx位（OSCCON<14:12>）确定当前振荡器源。
2. 执行解锁序列以允许写入OSCCON寄存器的高字节。
3. 将新振荡器源的相应值写入NOSCx位（OSCCON<10:8>）。
4. 执行解锁序列以允许写入 OSCCON 寄存器的低字节。
5. 将OSWEN位置1来启动振荡器切换。

一旦基本序列完成，系统时钟硬件将自动进行如下响应：

1. 时钟切换硬件将NOSCx位的新值和COSCx位做比较。如果相等，时钟切换为冗余操作。在这种情况下，OSWEN位被自动清零且时钟切换被中止。
2. 如果启动了有效时钟切换，则LOCK（OSCCON<5>）和CF（OSCCON<3>）位被清零。
3. 如果新振荡器现在不运行，硬件会将其开启。如果必须开启晶振，硬件将等待直至OST超时。如果新振荡源正在使用PLL，硬件将等待直到检测到PLL锁定（LOCK = 1）。
4. 硬件会等待新时钟源的10个时钟周期，然后执行时钟切换。
5. 硬件清零OSWEN位以指示时钟切换成功。此外，NOSCx位的值被传送到COSCx位中。
6. 此时旧时钟源被关闭，但LPRC（如果使能了WDT或FSCM）或SOSC（如果SOSCEN位保持置1状态）除外。

> **注 1：** 在整个时钟切换过程中，处理器将继续执行代码。对时序敏感的代码不应在此时执行。
>
> **2：** 不允许直接在使能PLL的任何主振荡器模式和FRCPLL模式之间进行时钟切换。这适用于任一方向的时钟切换。在这些情况下，应用必须首先切换到FRC模式将其作为两个PLL模式之间的过渡时钟源。

时钟切换的建议代码序列通常包括：

1. 在OSCCON寄存器解锁和写序列过程中禁止中断。
2. 通过两条连续的指令将78h和9Ah写入OSCCON<15:8>执行OSCCON高字节的解锁序列。
3. 执行解锁序列之后，立即使用指令将新振荡器源写入NOSCx位。
4. 通过两条连续的指令将46h和57h写入OSCCON<7:0>执行OSCCON低字节的解锁序列。
5. 执行解锁序列之后，立即使用指令将OSWEN位置1。
6. 继续执行对时钟不敏感的代码（可选）。
7. 调用适当时间的软件延时（周期计数），允许选定的振荡器和/或PLL启动并稳定下来。
8. 检查OSWEN是否为0。如果是，则切换成功。如果OSWEN位仍为置1状态，则检查LOCK位以确定故障原因。

解锁OSCCON寄存器并启动时钟切换的核心序列如例9-1所示。

例9-1： 时钟切换的基本代码序列

```
;Place the new oscillator selection in W0
;OSCCONH (high byte) Unlock Sequence
MOV         #OSCCONH, w1
MOV         #0x78, w2
MOV         #0x9A, w3
MOV.b       w2, [w1]
MOV.b       w3, [w1]
;Set new oscillator selection
MOV.b       WREG, OSCCONH
;OSCCONL (low byte) unlock sequence
MOV         #OSCCONL, w1
MOV         #0x46, w2
MOV         #0x57, w3
MOV.b       w2, [w1]
MOV.b       w3, [w1]
;Start oscillator switch operation
BSET        OSCCON, #0
```

9.5 振荡器模式

PLL模块如图9-2所示。在此系统中，PLL预分频器对主振荡器的输入进行分频，从而产生4 MHz输出。该输出用于驱动片上96 MHz PLL倍频器，进而驱动固定的3分频分频器和可配置的PLL预分频器/分频器来生成一系列系统时钟频率。CPDIV<1:0>位选择系统时钟速度。表9-2中列出了可用的时钟选项。

用户必须使用PLLMODE<3:0>配置位手动配置PLL分频器才能生成所需的4 MHz输出。这将可选的主振荡器频率总共限制为8种，如表9-3所示。

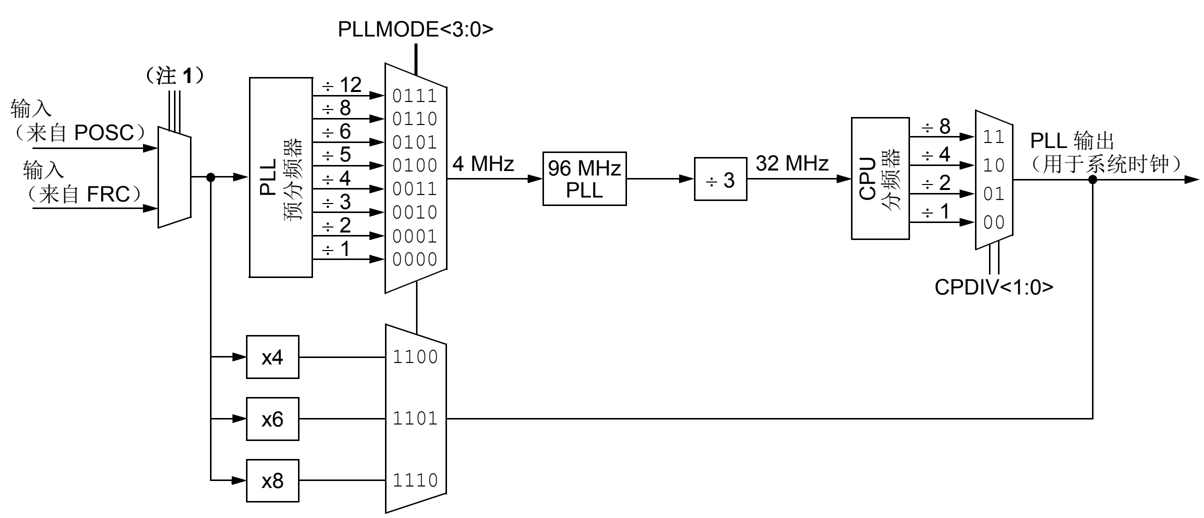
表9-2：　系统时钟选项

MCU时钟分频（CPDIV<1:0>）	单片机时钟频率
无（00）	32 MHz
÷2（01）	16 MHz
÷4（10）	8 MHz
÷8（11）	4 MHz

表9-3：　有效的主振荡器配置

振荡器输入频率	时钟模式	PLL模式 (PLLMODE<3:0>)
48 MHz	ECPLL	÷12（0111）
32 MHz	HSPLL和ECPLL	÷8（0110）
24 MHz	HSPLL和ECPLL	÷6（0101）
20 MHz	HSPLL和ECPLL	÷5（0100）
16 MHz	HSPLL和ECPLL	÷4（0011）
12 MHz	HSPLL和ECPLL	÷3（0010）
8 MHz	ECPLL、XTPLL和FRCPLL	÷2（0001）
4 MHz	ECPLL、XTPLL和FRCPLL	÷1（0000）

图9-2：　PLL模块



注 1：　该多路开关受COSC<2:0>位（当依靠PLL运行时）或NOSC<2:0>位（准备切换至PLL时）控制。

9.6 参考时钟输出

除了某些振荡器模式中使用的CLKO输出（Fosc/2）外，还可对PIC24FJ256GA705系列器件中的器件时钟进行配置，以为端口引脚提供参考时钟输出信号。该功能在所有振荡器配置中都可用，允许用户选择更大范围的时钟分频比来驱动应用中的外部器件。CLKO由配置位OSCIOFCN使能，与REFO参考时钟无关。REFO可映射到具有已映射输出功能的任一I/O引脚。更多信息，请参见表11-7。

该参考时钟输出由REFOCONL、REFOCONH和REFOTRIML寄存器控制。将ROEN位（REFOCONL<15>）置1可在REFO引脚上提供时钟信号。RODIV<14:0>位（REFOCONH<14:0>）允许选择不同的时钟分频比选项。ROTRIM<0:8>位（REFOTRIML<7:15>）允许用户将一个小数加到RODIVx值中。ROSWEN位（REFOCONL<9>）指示时钟分频器已成功切换。为切换分频器或微调REFO频率，用户应等到该位清零。向ROTRIMx和RODIVx写入更新值，将ROSWEN位置1，然后等到此位清零，之后可认为REFO时钟有效。

ROSEL<3:0>位（REFOCONL<3:0>）决定用于参考时钟输出的时钟源。ROSLP位（REFOCONL<11>）决定当器件处于休眠模式时REFO上的参考时钟源是否可用。

要在休眠模式下使用参考时钟输出，ROSLP位必须置1，同时必须使能ROSELx位选择的时钟，使其可以在休眠模式下工作（如果可能）。任何时钟切换期间清零ROSELx位都会使参考时钟输出频率随着系统时钟的变化而变化。ROOUT位使能/禁止REFO引脚上的参考时钟输出。

ROACTIVE位（REFOCONL<8>）指示模块处于工作状态；可以通过禁止模块（将ROEN设为0）将该位清零。ROACTIVE位指示模块处于工作状态时，用户不得更改参考时钟源或调整微调值或分频值。为避免出现毛刺，在ROACTIVE位为1之前，用户不应禁止模块。

PLLSS配置位（FOSC<4>）在清零时，可用于通过PLL生成REFO时钟，此时钟与系统时钟无关。PLL不能用于主时钟链。例如，如果系统时钟使用8 MHz的FRC，则PLL可使用此FRC作为输入并生成32 MHz的REFO输出（PLL 4x模式）。

9.7 辅助振荡器

9.7.1 基本SOSC工作

PIC24FJ256GA705系列器件不必将SOSCEN位置1来使用辅助振荡器。需要SOSC的任何模块（例如RTCC或Timer1）在需要时钟信号时将自动开启SOSC。但是，SOSC的起振时间较长（需要1秒）。为了避免外设启动时产生延迟，可使用SOSCEN位手动启动SOSC。

要使用辅助振荡器，SOSCSEL位（FOSC<3>）必须置1。将SOSCSEL位编程为0可将SOSC引脚配置为数字模式，从而使能引脚的数字I/O功能。

9.7.2 晶振选择

用于SOSC的32.768 kHz晶振必须具有以下规格，以便在SOSC处于高功耗模式（默认）时正确启动并以合适的频率运行：

- 12.5 pF负载电容
- 1.0 pF并联电容
- ESR的典型值为35k-50k；最大值为70k

此外，两个外部晶振负载电容应为18-22 pF，具体大小将基于PCB板布线。电容应为C0G，其容差为5%且额定电压为25V或更高。

SOSC的精度和占空比可以在REFO引脚上进行测量，建议占空比为40-60%，建议精度为±0.65 Hz。

9.7.3 低功耗SOSC工作

辅助振荡器可根据器件配置的要求工作在两种不同的功耗级别下。在低功耗模式下，振荡器工作在低驱动强度、低功耗状态下。默认情况下，振荡器使用高驱动强度，因而需要更多的功耗。低功耗模式由配置位SOSCHP（FDEVOPT1<3>）选择。由于此模式具有低驱动强度，使得SOSC对噪声更加敏感，从而需要较长的起振时间。该模式可与低负载电容晶振（6-9 pF）搭配使用来降低RTCC中的休眠电流。使用低功耗模式时，必须小心设计和布置SOSC电路，以确保振荡器正确启动并振荡。需要谨慎控制PCB布线问题和杂散电容等因素，以便使晶振正常工作。

寄存器9-6： REFOCONL：参考振荡器控制寄存器低位字

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-0
ROEN	—	ROSIDL	ROOUT	ROSLP	—	ROSWEN	ROACTIVE
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	ROSEL3	ROSEL2	ROSEL1	ROSEL0
bit 7							bit 0

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

bit 15 **ROEN：**参考振荡器输出使能位
1 = 使能参考振荡器模块
0 = 禁止参考振荡器

bit 14 **未实现：**读为0

bit 13 **ROSIDL：**REFO空闲模式停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 在空闲模式下模块继续工作

bit 12 **ROOUT：**参考时钟输出使能位
1 = 参考时钟在REFO引脚上输出
0 = 参考时钟不在REFO引脚上输出

bit 11 **ROSLP：**休眠模式下参考振荡器输出停止位
1 = 休眠模式下参考振荡器继续工作
0 = 休眠模式下禁止参考振荡器

bit 10 **未实现：**读为0

bit 9 **ROSWEN：**参考时钟RODIVx/ROTRIMx切换使能位
1 = 切换时钟分频器；当前正在进行时钟分频器切换
0 = 已完成时钟分频器切换

bit 8 **ROACTIVE：**参考时钟请求状态位
1 = 参考时钟已激活（用户不应更改REFO设置）
0 = 参考时钟未激活（用户可更新REFO设置）

bit 7-4 **未实现：**读为0

bit 3-0 **ROSEL<3:0>：**参考时钟源选择位
1111-1001 = 保留
1000 = REFI引脚
0111 = 保留
0110 = PLL
0101 = SOSC
0100 = LPRC
0011 = FRC
0010 = POSC
0001 = 系统时钟（FOSC/2）
0000 = FOSC

寄存器9-7： REFOCONH：参考振荡器控制寄存器高位字

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	RODIV<14:8>						
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RODIV<7:0>							
bit 7							bit 0

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

bit 15 **未实现：** 读为0

bit 14-0 **RODIV<14:0>：** 参考时钟分频比位
指定参考时钟的1/2周期（参考时钟源时钟数）
（例如：输出周期 = [参考时钟源 * 2] * RODIV<14:0>；此公式不适用于RODIV<14:0> = 0）。
111111111111111 = REFO时钟为基本时钟频率的65,534（32,767 * 2）分频
111111111111110 = REFO时钟为基本时钟频率的65,532（32,766 * 2）分频
•
•
•
000000000000011 = REFO时钟为基本时钟频率的6（3 * 2）分频
000000000000010 = REFO时钟为基本时钟频率的4（2 * 2）分频
000000000000001 = REFO时钟为基本时钟频率的2（1 * 2）分频
000000000000000 = REFO时钟与基本时钟的频率相同（不分频）

寄存器9-8： **REFOTRIML**：参考振荡器微调寄存器低位字

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROTRIM<0:7>							
bit 15							bit 8

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
ROTRIM8	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15-7 **ROTRIM<0:8>**：REFO微调位
这些位为REFO时钟的1/2周期提供加到RODIVx值中的小数。
000000000 = 0/512（将0.0分频值加到RODIVx值中）
000000001 = 1/512（将0.001953125分频值加到RODIVx值中）
000000010 = 2/512（将0.00390625分频值加到RODIVx值中）
•
•
•
100000000 = 256/512（将0.5000分频值加到RODIVx值中）
•
•
•
111111110 = 510/512（将0.99609375分频值加到RODIVx值中）
111111111 = 511/512（将0.998046875分频值加到RODIVx值中）

bit 6-0 **未实现**：读为0

10.0 节能特性

> **注：** 本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的“节能特性”（DS39698），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

PIC24FJ256GA705系列器件提供了管理功耗的功能，该功能是通过选择性地管理CPU和外设的时钟来实现的。一般来说，较低的时钟频率和减少时钟所驱动电路的数目可降低功耗。所有PIC24F器件均可通过以下4种方式管理功耗：

- 时钟频率
- 基于指令的休眠模式和空闲模式
- 软件控制的打盹模式
- 用软件有选择地进行外设控制

可以组合使用这些方法，从而在保证关键应用特性（如对于时序敏感的通信）的情况下有选择地调节应用的功耗。

10.1 时钟频率和时钟切换

PIC24F器件的时钟频率范围较宽，用户可根据应用需要进行选择。如果未锁定系统时钟配置，用户只需更改NOSC<2:0>位即可选择低功耗或高精度振荡器。在工作期间更改系统时钟的过程以及相应的限制，已在第9.0节“振荡器配置”中进行了详细的讨论。

10.2 基于指令的节能模式

PIC24F器件有两种特殊的节能模式，通过执行特殊的PWRSAV指令可以进入这两种模式。休眠模式下时钟停止工作并暂停所有代码执行；空闲模式下CPU暂停工作并暂停代码执行，但是允许外设模块继续工作。PWRSAV指令的汇编语法如例10-1所示。

MPLAB XC16 C编译器具有“内置”函数，可实现节能模式，如下所示：

```
Idle();     // places part in Idle
Sleep();    // places part in Sleep
```

在被允许的中断产生、WDT超时或器件复位时，器件会退出休眠和空闲模式。器件退出这两种模式的过程称为“唤醒”。

> **注：** SLEEP_MODE和IDLE_MODE是在所选器件的汇编包含文件中定义的常量。

10.2.1 休眠模式

休眠模式具有以下特性：

- 系统时钟源关闭。如果使用了片上振荡器，也要关闭它。
- 如果没有I/O引脚拉电流，则器件电流消耗将降至最低。
- 由于禁止了系统时钟源，所以故障保护时钟监视器在休眠模式下不工作。
- 如果WDT被使能，则LPRC时钟将在休眠模式下继续运行。
- 如果WDT被使能，则它在进入休眠模式之前被自动清零。
- 某些器件功能或外设可在休眠模式下继续工作。这包括I/O端口上的输入电平变化通知（Input Change Notification，ICN）或使用外部时钟输入的外设等。任何需要系统时钟源工作的外设都会在休眠模式下被禁止。

当发生以下任何事件时，器件将从休眠模式唤醒：

- 产生任何被单独允许的中断
- 任何形式的器件复位
- WDT超时

从休眠模式唤醒时，处理器将使用在进入休眠模式时的有效时钟源重新启动。

例10-1： PWRSAV指令语法

```
PWRSAV    #SLEEP_MODE          ; Put the device into SLEEP mode
PWRSAV    #IDLE_MODE           ; Put the device into IDLE mode
```

10.2.2 空闲模式

空闲模式具有以下特性：

- CPU停止执行指令。
- WDT被自动清零。
- 系统时钟源保持工作状态。默认情况下，所有外设模块将继续使用系统时钟源正常工作，也可以有选择地禁止它们（见**第10.4节“选择性外设模块控制”**）。
- 如果WDT或FSCM被使能，则LPRC也将保持工作状态。

当发生以下任何事件时，器件将从空闲模式唤醒：

- 产生任何已被单独允许的中断。
- 任何器件复位。
- WDT超时。

从空闲模式唤醒时，重新为CPU提供时钟，且立即从`PWRSAV`指令之后的下一条指令或ISR中的第一条指令开始执行。

10.2.3 在节能指令执行期间的中断

在执行`PWRSAV`指令时产生的任何中断都将延迟到进入休眠或空闲模式后才起作用，并导致器件从休眠或空闲模式唤醒。

10.2.4 低压保持稳压器

PIC24FJ256GA705系列器件集成了另一个片内稳压器，旨在为标称值为1.2V的一些单片机功能供电。此稳压器允许原本处于非活动状态的功能（如数据RAM和WDT）保持在节能模式，或者保持在与其他情况相比功耗更低的状态。

保持休眠模式下所使用的功耗要低于标准休眠模式下的功耗，但恢复和开始执行所用的时间也较长。在退出保持休眠模式时，额外需要10-15 μs（典型值）来将V_{CAP}从1.2V充电到1.8V并开始执行指令。

VREGS位可控制退出休眠模式（常规和保持）的速度，但消耗的功耗较大。使能稳压器带隙会增大电流，但会将从休眠模式恢复的时间缩短约10 μs。

低压保持稳压器仅在调用休眠模式时可用。它由$\overline{\text{LPCFG}}$配置位（FPOR<2>）控制，在固件中由RETEN位（RCON<12>）控制。必须将$\overline{\text{LPCFG}}$编程为`0`并将RETEN位置1，才能使能该稳压器。

10.2.5 退出低压保持休眠模式

所有用于退出标准休眠模式的方法均适用于退出保持休眠模式（$\overline{\text{MCLR}}$和INT0等）。不过，为使稳压器从1.8V（工作模式）切换到1.2V（保持模式），在执行保持休眠模式到可以退出保持休眠模式之间使用一个硬件“锁定定时器”。

在“锁定时间”内，POR或$\overline{\text{MCLR}}$是退出保持休眠模式的惟一方法。被置为有效的中断（例如INT0）在“锁定时间”内会被屏蔽。然后，锁定定时器会设置一个从器件进入保持休眠模式到可从保持休眠模式退出的最小间隔。在锁定期间中断不“保持在待处理状态”；它们会被屏蔽，并且为了在锁定时间结束后退出休眠模式，退出源在经过锁定时间后必须置为有效。

锁定定时器源自LPRC时钟，其频率容差较大（未经调整）。

锁定时间有以下两种情况：

- 如果LPRC在保持休眠模式下未运行，则锁定时间为2个LPRC周期 + LPRC唤醒时间
- 如果LPRC在保持休眠模式下在运行，则锁定时间为1个LPRC周期

有关LPRC时序的信息，请参见“交流电气规范”中的表32-20和表32-21。

10.2.6 低功耗休眠模式汇总

RETEN位和VREGS位（RCON<12,8>）可实现四种不同的休眠模式，具体因唤醒时间和功耗而异。表10-1对这些模式进行了汇总。有关电流消耗和唤醒时间的具体信息，请参见**第32.0节“电气特性”**。

表10-1： 低功耗休眠模式

RETEN	VREGS	模式	相对功耗（1 = 最低）
0	0	休眠	3
0	1	快速唤醒	4
1	0	保持休眠	1
1	1	快速保持	2

10.3 打盹模式

通常，更改时钟速度和使用某种节能模式是降低功耗的首选策略。在某些情况下可能不可行。例如，某些应用可能必须保持不间断的同步通信，即便在它不执行任何其他操作时也不例外。降低系统时钟速度可能会带来通信错误，而使用节能模式可能会完全停止通信。

打盹模式是另一种简单有效的节能方法，它可以在器件仍然执行代码的情况下降低功耗。在此模式下，系统时钟以相同的时钟源和相同的速度继续工作。外设模块时钟速度保持不变，但CPU时钟的速度降低了。保持这两个时钟域同步，可以保持外设访问SFR的能力，同时CPU以较慢的速度执行代码。

通过将DOZEN位（CLKDIV<11>）置1使能打盹模式。外设与内核的时钟速度之比是由DOZE<2:0>位（CLKDIV<14:12>）决定的。有8种可能的配置，从1:1到1:256，其中1:1是默认设置。

在事件驱动的应用中，还可使用打盹模式有选择地降低功耗。这样就可以实现不间断地运行对时序敏感的功能（如同步通信），而CPU保持空闲等待事件调用中断服务程序。通过将ROI位（CLKDIV<15>）置1，可以使器件在产生中断时自动返回到全速CPU工作模式。默认情况下，中断事件对打盹模式工作没有影响。

10.4 选择性外设模块控制

空闲和打盹模式允许用户通过降低CPU时钟速度或停止CPU时钟大幅降低功耗。然而，外设模块的时钟仍然保持运行，因此会有功耗产生。可能在有些情况下应用需要这些模式无法提供的功能，比如将绝大部分能源分配给CPU处理工作，而只为外设提供最低的功耗。

PIC24F器件允许有选择地禁止外设模块，从而降低或消除它们的功耗，以此满足上述需求。这可以通过两个控制位来实现：

- 外设使能位，通常称为“XXXEN”，位于模块主控制SFR中。
- 外设模块禁止（PMD）位，通常称为“XXXMD”，位于某个PMD控制寄存器中。

这两位在使能或禁止相关模块时具有相似的功能。将模块的PMD位置1会禁止该模块的所有时钟源，从而将其功耗降至绝对最低。在此状态下，与此外设相关的控制和状态寄存器也会被禁止，所以无法写这些寄存器且读取值无效。许多外设模块都有一个对应的PMD位。

而通过清零某个模块的XXXEN位将会禁止其功能，但是仍然允许对其寄存器进行读写操作。这会降低功耗，但是没有使用PMD位所降低的程度高。大多数外设模块都有一个使能位；但输入捕捉、输出比较和RTCC模块除外。

要节省更多的功耗，也可在器件进入空闲模式时有选择地禁止外设模块。使用通用名格式为“XXXIDL”的控制位可以执行此操作。默认情况下，可以在空闲模式下工作的所有模块都可以执行此操作。使用“在空闲模式下禁止”功能可进一步降低空闲模式下的功耗，在功耗极其重要的应用中，更大程度地节省功耗。

表 10-2： 外设模块禁止寄存器汇总

寄存器	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
PMD1	—	—	T3MD	T2MD	T1MD	—	—	—	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	—	ADCMD	0000
PMD2	—	—	—	—	—	IC3MD	IC2MD	IC1MD	—	—	—	—	—	OC3MD	OC2MD	OC1MD	0000
PMD3	—	—	—	—	—	CMPMD	RTCCMD	PMPMD	CRCMD	—	—	—	—	—	I2C2MD	—	0000
PMD4	—	—	—	—	—	—	—	—	—	—	—	—	REFOMD	CTMUMD	LVDMD	—	0000
PMD5	—	—	—	—	—	—	—	—	—	—	—	—	CCP4MD	CCP3MD	CCP2MD	CCP1MD	0000
PMD6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPI3MD	0000
PMD7	—	—	—	—	—	—	—	—	—	—	DMA1MD	DMA0MD	—	—	—	—	0000
PMD8	—	—	—	—	—	—	—	—	—	—	—	—	CLC2MD	CLC1MD	—	—	0000

图注： — = 未实现，读为 0。复位值以十六进制显示。

寄存器10-1： PMD1：外设模块禁止寄存器1

U-0	U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
—	—	T3MD	T2MD	T1MD	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	—	ADC1MD
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13 **T3MD：**Timer3模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 12 **T2MD：**Timer2模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 11 **T1MD：**Timer1模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 10-8 **未实现：**读为0

bit 7 **I2C1MD：**I2C1模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 6 **U2MD：**UART2模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 5 **U1MD：**UART1模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 4 **SPI2MD：**SPI2模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 3 **SPI1MD：**SPI1模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 2-1 **未实现：**读为0

bit 0 **ADC1MD：**A/D转换器模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

寄存器10-2： PMD2：外设模块禁止寄存器2

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	IC3MD	IC2MD	IC1MD
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	OC3MD	OC2MD	OC1MD
bit 7							bit 0

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

bit 15-11 **未实现：**读为0

bit 10 **IC3MD：**输入捕捉3模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 9 **IC2MD：**输入捕捉2模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 8 **IC1MD：**输入捕捉1模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 7-3 **未实现：**读为0

bit 2 **OC3MD：**输出比较3模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 1 **OC2MD：**输出比较2模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 0 **OC1MD：**输出比较1模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

寄存器10-3： PMD3：外设模块禁止寄存器3

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CMPMD	RTCCMD	PMPMD
bit 15							bit 8

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0
CRCMD	—	—	—	—	—	I2C2MD	—
bit 7							bit 0

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

bit 15-11 **未实现：**读为0

bit 10 **CMPMD：**三比较器模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 9 **RTCCMD：**RTCC模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 8 **PMPMD：**增强型并行主端口禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 7 **CRCMD：**CRC模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 6-2 **未实现：**读为0

bit 1 **I2C2MD：**I2C2模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 0 **未实现：**读为0

寄存器10-4： PMD4：外设模块禁止寄存器4

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	REFOMD	CTMUMD	LVDMD	—
bit 7							bit 0

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

bit 15-4 **未实现：**读为0

bit 3 **REFOMD：**参考输出时钟禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 2 **CTMUMD：**CTMU模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 1 **LVDMD：**高/低压检测模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 0 **未实现：**读为0

寄存器10-5： PMD5：外设模块禁止寄存器5

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	CCP4MD	CCP3MD	CCP2MD	CCP1MD
bit 7							bit 0

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

bit 15-4 **未实现：**读为0

bit 3 **CCP4MD：**MCCP4模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 2 **CCP3MD：**MCCP3模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 1 **CCP2MD：**MCCP2模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 0 **CCP1MD：**MCCP1模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

寄存器10-6： PMD6：外设模块禁止寄存器6

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	SPI3MD
bit 7							bit 0

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

bit 15-1 **未实现：**读为0

bit 0 **SPI3MD：**SPI3模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

寄存器10-7： PMD7：外设模块禁止寄存器7

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	—	DMA1MD	DMA0MD	—	—	—	—
bit 7							bit 0

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

bit 15-6 **未实现：**读为0

bit 5 **DMA1MD：**DMA1控制器（通道4到7）禁止位
1 = 禁止控制器
0 = 使能控制器电源和时钟源

bit 4 **DMA0MD：**DMA0控制器（通道0到3）禁止位
1 = 禁止控制器
0 = 使能控制器电源和时钟源

bit 3-0 **未实现：**读为0

寄存器10-8： PMD8：外设模块禁止寄存器8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0
—	—	—	—	CLC2MD	CLC1MD	—	—
bit 7							bit 0

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

bit 15-4 **未实现：**读为0

bit 3 **CLC2MD：**CLC2模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 2 **CLC1MD：**CLC1模块禁止位
1 = 禁止模块
0 = 使能模块电源和时钟源

bit 1-0 **未实现：**读为0

注：

11.0 I/O端口

注：	本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》的**"带外设引脚选择（PPS）功能的I/O端口"**（DS39711），该文档可从Microchip 网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

所有器件引脚（除了V_{DD}、V_{SS}、$\overline{MCLR}$和OSCI/CLKI以外）均由外设和并行I/O（PIO）端口共用。所有I/O输入端口都为施密特触发器（ST）输入，提高了抗噪声能力。

11.1 并行I/O（PIO）端口

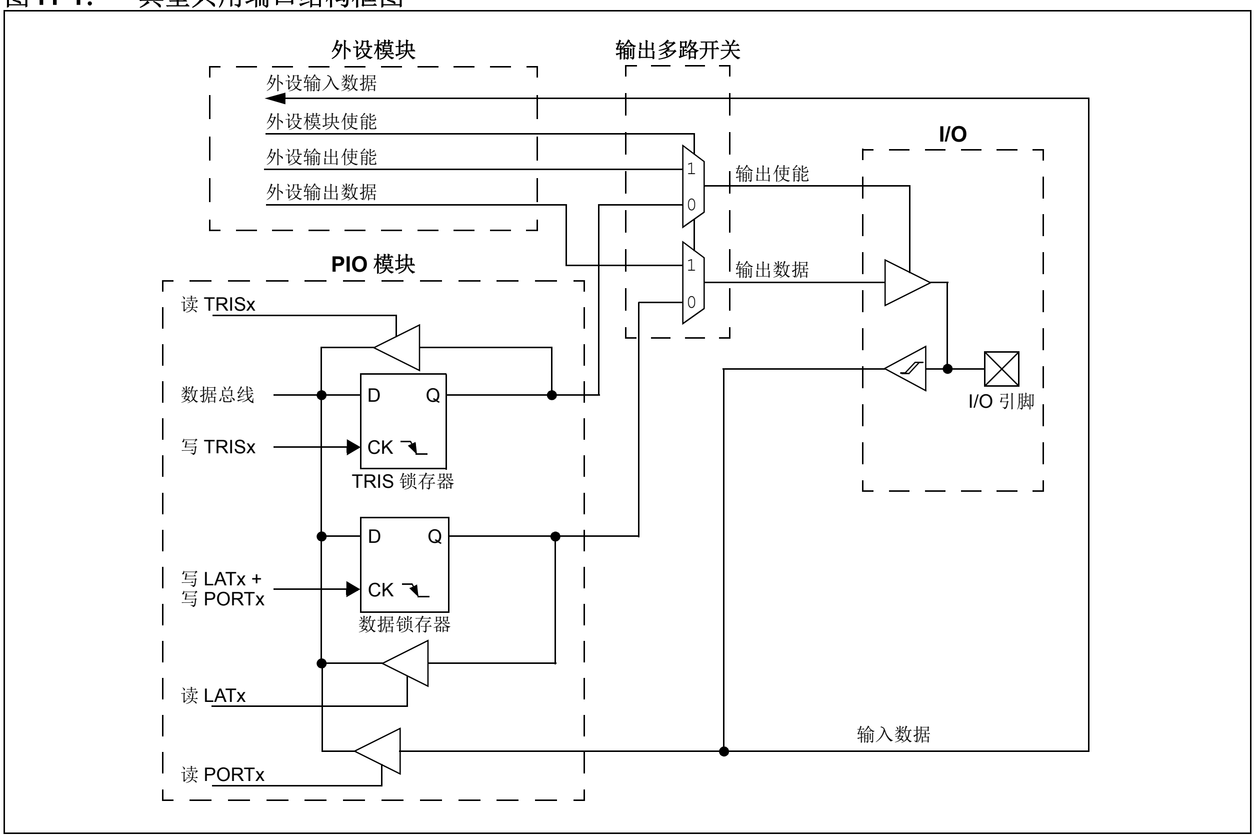
与某个外设共用一个引脚的并行I/O端口通常服从于该外设。外设的输出缓冲器数据和控制信号提供给一对多路开关。这对多路开关用于选择I/O引脚的输出数据和控制信号是来自外设还是相应的端口。该逻辑电路同时会阻止"环通"（loop through，即一个端口的数字输出可以驱动共用同一个引脚的外设的输入）。图11-1显示了端口是如何与其他外设复用的以及与这些外设连接的相关I/O引脚。

当外设使能，并且外设正在驱动相关引脚时，禁止将该引脚作为通用输出引脚。可以读该I/O引脚，但并行端口位的输出驱动器将被禁止。如果使能某外设但该外设没有驱动相应的引脚，则该引脚可由端口驱动。

所有端口引脚都有三个与端口引脚作为数字I/O时的工作直接相关的寄存器，和一个与端口引脚作为模拟输入时的工作相关的寄存器。数据方向寄存器（TRISx）决定引脚是输入还是输出。如果数据方向位为1，则引脚为输入。复位后，所有端口引脚均定义为输入。读输出锁存寄存器（LATx）时，读到的是锁存器中的值；写锁存器时，写入的是锁存器。但读取PORTx寄存器时，读到的是端口引脚的值；而写入端口引脚时，写入的是相应的锁存器。

对于特定器件无效的任何位及其相关的数据和控制寄存器都将被禁止。这意味着对应的LATx和TRISx寄存器以及端口引脚都将读为零。表11-3到表11-5列出了各器件型号可用的ANSELx位和端口。当端口引脚与另一个定义为仅输入的外设或功能复用时，它将被视为专用端口，因为没有任何其他的竞争输入源。

图11-1： 典型共用端口结构框图



11.1.1 I/O端口写/读时序

端口方向改变或端口写操作与同一端口的读操作之间需要间隔一个指令周期。通常该指令是NOP指令。

11.1.2 漏极开路配置

除PORTx、LATx和TRISx寄存器用于数据控制外，各端口引脚也可单独配置为数字输出或漏极开路输出。这是由与每个端口相关的漏极开路控制寄存器ODCx控制的。将其中的任何位置1即可将相应的引脚配置为漏极开路输出。

这种漏极开路特性允许通过使用外部上拉电阻在任何只能用作数字功能的引脚上产生高于VDD（如5V）的输出电压。允许的最大漏极开路电压与最大VIH规范相同。

11.2 配置模拟端口引脚（ANSx）

ANSx和TRISx寄存器用于控制具有模拟功能的引脚的操作。每个具有模拟功能的端口引脚都与某个ANSx位相关，该位决定了引脚功能是模拟的还是数字的。有关不同ANSx和TRISx位设置时引脚的详细操作，请参见表11-1。

当读取PORTx寄存器时，所有配置为模拟输入通道的引脚均读为零（低电平）。

11.2.1 模拟输入引脚和电压注意事项

用作器件输入的引脚可承受的电压与该引脚的输入功能有关。大多数输入引脚能够承受最高5.5V的直流电压，这是数字逻辑电路的典型电压值。但是，几个引脚只能承受最高为VDD的电压。应始终避免使这些引脚上的电压超过VDD。

表11-2总结了不同的电压容差。更多详细信息，请参见第32.0节“电气特性”。

表11-1： 配置I/O引脚的模拟/数字功能

引脚功能	ANSx设置	TRISx设置	备注
模拟输入	1	1	建议保持ANSx = 1。
模拟输出	1	1	建议保持ANSx = 1。
数字输入	0	1	固件必须在将引脚配置为数字输入后等待至少一个指令周期，然后才能读取有效输入值。
数字输出	0	0	请确保禁止该引脚上的模拟输出功能（如果具备的话）。

表11-2： 端口或引脚可承受的输入电压说明

端口或引脚	可承受的输入电压	说明
PORTB<11:10,8:5>	5.5V	可承受高于VDD的输入电压；这对于大部分标准逻辑有用。
PORTC<9:6>		
PORTA<14:7,4:0>	VDD	只能承受VDD输入电压。
PORTB<15:12,9,4:0>		
PORTC<5:0>		

表**11-3：** **PORTA**引脚和**ANSELx**可用性

器件	**PORTA I/O**引脚															
	RA15	**RA14**	**RA13**	**RA12**	**RA11**	**RA10**	**RA9**	**RA8**	**RA7**	**RA6**	**RA5**	**RA4**	**RA3**	**RA2**	**RA1**	**RA0**
PIC24FJXXXGA705	—	X	X	X	X	X	X	X	X	—	—	X	X	X	X	X
PIC24FJXXXGA704	—	—	—	—	—	X	X	X	X	—	—	X	X	X	X	X
PIC24FJXXXGA702	—	—	—	—	—	—	—	—	—	—	—	X	X	X	X	X
有ANSELA位	—	—	—	—	—	—	—	—	—	—	—	—	X	X	X	X

表**11-4：** **PORTB**引脚和**ANSELx**可用性

器件	**PORTB I/O**引脚															
	RB15	**RB14**	**RB13**	**RB12**	**RB11**	**RB10**	**RB9**	**RB8**	**RB7**	**RB6**	**RB5**	**RB4**	**RB3**	**RB2**	**RB1**	**RB0**
PIC24FJXXXGA705	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
PIC24FJXXXGA704	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
PIC24FJXXXGA702	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
有ANSELB位	X	X	X	X	—	—	X	—	—	—	—	—	X	X	X	X

表**11-5：** **PORTC**引脚和**ANSELx**可用性

器件	**PORTC I/O**引脚															
	RC15	**RC14**	**RC13**	**RC12**	**RC11**	**RC10**	**RC9**	**RC8**	**RC7**	**RC6**	**RC5**	**RC4**	**RC3**	**RC2**	**RC1**	**RC0**
PIC24FJXXXGA705	—	—	—	—	—	—	X	X	X	X	X	X	X	X	X	X
PIC24FJXXXGA704	—	—	—	—	—	—	X	X	X	X	X	X	X	X	X	X
PIC24FJXXXGA702	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
有ANSELC位	—	—	—	—	—	—	—	—	—	—	—	—	X	X	X	X

11.3 电平变化中断（IOC）

I/O端口的电平变化中断功能允许PIC24FJ256GA705系列器件在选定输入引脚的电平状态变化（Change-Of-State，COS）时向处理器发出中断请求。即便是在时钟被禁止的休眠模式下，该特性也可检测到输入电平状态变化。

将针对某个引脚的IOCPx和/或IOCNx寄存器位置1，可以使能该引脚电平变化中断功能。例如，PORTC具有名称为IOCPC和IOCNC的寄存器可用于这些功能。将IOCPx寄存器位设置为值1允许在从低电平跳变为高电平时产生中断，而将IOCNx寄存器位设置为值1则允许在从高电平跳变为低电平时产生中断。将这两个寄存器位均设置为值1则允许在上述两种情况下均产生中断（例如，引脚上的一个脉冲会产生两个中断）。为检测到任意IOC，全局IOC中断允许位（IEC1<3>）必须置1，PADCON<15>位必须置1（IOCON），相关ISFx标志必须清零。

针对某个引脚产生中断请求时，相应的状态标志（IOCFx寄存器位）将置1，表示该引脚的电平发生了变化。IOCFx寄存器位在通过写入零来清零之前始终保持为1。给定端口中的任一IOCFx标志位置1时，IOCSTAT寄存器中相应的IOCPxF位将置1。该标志指示在给定端口的某个位上检测到了电平状态变化。所有IOCFx<15:0>位均清零时IOCPxF标志将清零。

通过读-修改-写操作向一个或多个位写入零，可将多个单独状态标志清零。如果在读-修改-写序列期间对某个引脚状态位进行清零时检测到该引脚上出现了另一个边沿，则相关电平变化标志仍会在读-修改-写序列结束时置1。

用户应使用例11-1中所示的指令序列（或等效序列）来将电平变化中断状态寄存器清零。

该序列结束时，W0寄存器中与检测到电平发生变化的端口引脚对应的位均将清零。这样，所有引脚电平状态变化的指示都不会丢失。

电平变化中断具有异步和实时特性，因而检测到电平变化时，在端口引脚上读取的值可能不会指示端口的状态，因为在清零中断标志和读取端口的间隔内可能会发生另外一次电平变化。如果用户应用允许，这种情况由用户代码进行处理。为将此间隔时间降至最小，建议使任何修改IOCFx寄存器的代码在中断处理程序中运行或在禁止中断的条件下运行。

每个电平变化中断（Interrupt-on-Change，IOC）引脚都有一个与之相连的弱上拉和弱下拉电路。弱上拉电路提供连接到该引脚的电流源，而弱下拉电路提供连接到该引脚的电流汇。这样的话，当连接按钮或键盘设备时，便不再需要使用外部电阻。

使用IOCPUx寄存器可单独使能上拉电路，使用IOCPDx寄存器可单独使能下拉电路。每个IOC引脚都为其上拉电路和下拉电路提供了独立的控制位。将某个控制位置1可使能其对应引脚的弱上拉或弱下拉电路。

> **注：** 当引脚被配置为数字输出时，引脚上的上拉和下拉电路应始终被禁止。

例11-1： IOC状态读取/清零（汇编语言）

```
MOV    0xFFFF, W0    ; Initial mask value 0xFFFF -> W0
XOR    IOCFx, W0     ; W0 has '1' for each bit set in IOCFx
AND    IOCFx         ; IOCFx & W0 ->IOCFx
```

例11-2： 端口读/写（汇编语言）

```
MOV    0xFF00, W0    ; Configure PORTB<15:8> as inputs
MOV    W0, TRISB     ; and PORTB<7:0> as outputs
NOP                  ; Delay 1 cycle
BTSS   PORTB, #13    ; Next Instruction
```

例11-3： 端口读/写（C语言）

```
TRISB = 0xFF00;              // Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
Nop();                       // Delay 1 cycle
If (PORTBbits.RB13){ };      // Next Instruction
```


11.4 I/O端口控制寄存器

寄存器11-1： PADCON：端口配置寄存器

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
IOCON	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	PMPTTL
bit 7							bit 0

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

bit 15 **IOCON**：电平变化中断使能位
1 = 使能电平变化中断功能
0 = 禁止电平变化中断功能

bit 14-1 **未实现**：读为0

bit 0 **PMPTTL**：PMP端口类型位
1 = PMP端口引脚采用TTL电平
0 = PMP端口引脚采用施密特触发器

寄存器11-2： IOCSTAT：电平变化中断状态寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/HS/HC-0	R/HS/HC-0	R/HS/HC-0
—	—	—	—	—	IOCPCF	IOCPBF	IOCPAF
bit 7							bit 0

图注：	HS = 硬件置1位	HC = 硬件清零位	
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15-3 **未实现**：读为0

bit 2 **IOCPCF**：电平变化中断PORTC标志位
1 = 在使能了IOC的PORTC引脚上检测到电平变化
0 = 未检测到电平变化，或者用户清除了所有检测到的电平变化

bit 1 **IOCPBF**：电平变化中断PORTB标志位
1 = 在使能了IOC的PORTB引脚上检测到电平变化
0 = 未检测到电平变化，或者用户清除了所有检测到的电平变化

bit 0 **IOCPAF**：电平变化中断PORTA标志位
1 = 在使能了IOC的PORTA引脚上检测到电平变化
0 = 未检测到电平变化，或者用户清除了所有检测到的电平变化

寄存器11-3： TRISx：PORTx输出使能寄存器[1]

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISx<15:8>							
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISx<7:0>							
bit 7							bit 0

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

bit 15-0 **TRISx<15:0>**：PORTx输出使能位
1 = PORTx[n]引脚不输出LATx[n]
0 = PORTx[n]引脚输出LATx[n]

注 1： 有关该寄存器各个位的可用性，请参见表11-3、表11-4和表11-5。

寄存器11-4： PORTx：PORTx输入数据寄存器[1]

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PORTx<15:8>							
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PORTx<7:0>							
bit 7							bit 0

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

bit 15-0 **PORTx<15:0>**：PORTx数据输入值位

注 1： 有关该寄存器各个位的可用性，请参见表11-3、表11-4和表11-5。

寄存器11-5： LATx：PORTx输出数据寄存器[1]

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
LATx<15:8>							
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
LATx<7:0>							
bit 7							bit 0

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

bit 15-0 **LATx<15:0>**：PORTx数据输出值位

注 1： 有关该寄存器各个位的可用性，请参见表11-3、表11-4和表11-5。

寄存器11-6： ODCx：PORTx漏极开路使能寄存器[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ODCx<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ODCx<7:0>							
bit 7							bit 0

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

bit 15-0 **ODCx<15:0>**：PORTx漏极开路使能位
1 = 使能PORTx引脚漏极开路
0 = 禁止PORTx引脚漏极开路

注 1： 有关该寄存器各个位的可用性，请参见表11-3、表11-4和表11-5。

寄存器11-7： ANSELx：PORTx模拟选择寄存器[1]

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANSELx<15:8>							
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANSELx<7:0>							
bit 7							bit 0

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

bit 15-0 **ANSELx<15:0>**：PORTx模拟选择位
1 = 使能PORTx[n]引脚模拟输入，禁止PORTx[n]引脚数字输入
0 = 禁止PORTx[n]引脚模拟输入，使能PORTx[n]引脚数字输入

注 1： 有关该寄存器各个位的可用性，请参见表11-3、表11-4和表11-5。

寄存器11-8： IOCPx：电平变化中断上升沿x寄存器[1,2,3]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
IOCPx<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IOCPx<7:0>							
bit 7							bit 0

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

bit 15-0 **IOCPx<15:0>**：电平变化中断上升沿x使能位
1 = 允许IOCx引脚上升沿电平变化中断；相关状态位和中断标志将在检测到上升沿时置1
0 = 禁止IOCx引脚上升沿电平变化中断

注 **1**： 将IOCPx和IOCNx均置1将同时允许IOCx引脚上升沿和下降沿电平变化中断，将这两个寄存器均清零会禁止该功能。
2： 在模块已使能（IOCON = 1）时更改该寄存器的值可能会导致虚假IOC事件。更改该寄存器的值时，相应的中断必须忽略、清零（使用IOCFx）或屏蔽（在中断控制器内），或者，该模块必须禁止（IOCON = 0）。
3： 有关该寄存器各个位的可用性，请参见表11-3、表11-4和表11-5。

寄存器11-9： IOCNx：电平变化中断下降沿x寄存器[1,2,3]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
IOCNx<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IOCNx<7:0>							
bit 7							bit 0

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

bit 15-0 **IOCNx<15:0>**：电平变化中断下降沿x使能位
1 = 允许IOCx引脚下降沿电平变化中断；相关状态位和中断标志将在检测到下降沿时置1
0 = 禁止IOCx引脚下降沿电平变化中断

注 **1**： 将IOCPx和IOCNx均置1将同时使能IOCx引脚上升沿和下降沿电平变化中断，将这两个寄存器均清零会禁止该功能。
2： 在模块已使能（IOCON = 1）时更改该寄存器的值可能会导致虚假IOC事件。更改该寄存器的值时，相应的中断必须忽略、清零（使用IOCFx）或屏蔽（在中断控制器内），或者，该模块必须禁止（IOCON = 0）。
3： 有关该寄存器各个位的可用性，请参见表11-3、表11-4和表11-5。

寄存器11-10： IOCFx：电平变化中断标志x寄存器[1,2]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
IOCFx<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IOCFx<7:0>							
bit 7							bit 0

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

bit 15-0 **IOCFx<15:0>**：电平变化中断标志x位
1 = 检测到相关引脚上发生了电平变化；当IOCPx = 1且在IOCx引脚上检测到上升沿时，或当IOCNx = 1且在IOCx引脚上检测到下降沿时置1。
0 = 未检测到电平变化，或者用户清除了检测到的电平变化

注 1： 不能通过软件写操作将IOCFx寄存器位置1（因为这将需要增加有效逻辑）。要测试IOC中断，建议先使能一个或多个GPIO引脚的IOC功能，再使用相应的LATx寄存器位触发IOC中断。
2： 有关该寄存器各个位的可用性，请参见表11-3、表11-4和表11-5。

寄存器11-11： IOCPUx：电平变化中断上拉使能x寄存器[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
IOCPUx<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IOCPUx<7:0>							
bit 7							bit 0

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

bit 15-0 **IOCPUx<15:0>**：电平变化中断上拉使能x位
1 = 使能上拉
0 = 禁止上拉

注 1： 有关该寄存器各个位的可用性，请参见表11-3、表11-4和表11-5。

寄存器11-12： **IOCPDx**：电平变化中断下拉使能**x**寄存器[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
IOCPDx<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IOCPDx<7:0>							
bit 7							bit 0

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

bit 15-0 **IOCPDx<15:0>**：电平变化中断下拉使能x位
1 = 使能下拉
0 = 禁止下拉

注 1：有关该寄存器各个位的可用性，请参见表11-3、表11-4和表11-5。

11.5 外设引脚选择（PPS）

通用器件的主要挑战是提供尽可能多的外设功能，同时将其与I/O引脚功能的冲突降到最小。在需要多个外设复用一个引脚的应用中，要在应用程序代码中进行变通比较困难，换句话说彻底重新设计可能是惟一的选择。

外设引脚选择（PPS）功能提供了这些选择的替代方法，使得用户可以在较宽的I/O引脚范围内选择和配置外设功能。通过增加特定器件上可用的引脚配置选项，用户可以让单片机更好地适合他们的整个应用，而不必通过修改应用来适应器件。

外设引脚选择功能通过固定数量的数字I/O引脚进行操作。用户可将任一数字外设的输入和/或输出独立映射到这些I/O引脚之一。外设引脚选择通过软件执行，通常不需要对器件重新编程。一旦建立外设映射，就同时包含了硬件保护，以防止对外设映射的意外或错误更改。

11.5.1 可用的引脚

外设引脚选择功能可在最多44个引脚的范围内使用；可用的引脚数取决于特定器件及其引脚数。支持外设引脚选择功能的引脚在其完整引脚名称中有“RPn”或“RPIn”标识，其中“n”指的是可重映射的引脚的编号。“RP”用于指定支持可重映射输入和输出功能的引脚，而“RPI”指定仅支持可重映射输入功能的引脚。

PIC24FJ256GA705系列器件支持的可重映射输入/输出引脚数大于可重映射仅输入引脚数。在此器件系列中，最多有33个可重映射输入/输出引脚，具体取决于所选特定器件的引脚数。这些引脚的编号为RP0到RP28以及RPI29到RPI32。

表1-1汇总了各封装的引脚排列选项。

11.5.2 可用的外设

外设引脚选择管理的外设都是仅数字外设， 包括通用串行通信（UART和SPI）、通用定时器时钟输入、定时器相关外设（输入捕捉和输出比较）以及外部中断输入。由于比较器模块的输出为离散数字信号，因此也包括在内。

PPS不适用于以下外设：

- I^2C（输入和输出）
- 输入电平变化通知
- EPMP信号（输入和输出）
- 模拟（输入和输出）
- INT0

有引脚选择和无引脚选择的外设的最大区别是有引脚选择的外设与默认I/O引脚之间无关联。必须始终在使用外设前将其分配给特定的I/O引脚。相反，假设无引脚选择的外设处于激活状态且未与其他外设发生冲突，则此外设可始终通过默认引脚使用。

11.5.2.1 外设引脚选择功能优先级

引脚可选的外设输出（例如，输出比较和UART发送）的优先级高于引脚（例如EPMP和端口I/O）上的通用数字功能。专用数字输出的优先级高于同一引脚上的PPS输出。本数据手册的引脚框图按照优先级顺序列出了外设输出。请参见这些引脚框图以了解关于某个特定引脚的优先级信息。

与具有固定外设的PIC24F器件不同，可选引脚的外设输入从不具有该引脚的所有权。引脚的输出缓冲器由该引脚的TRISx位设置或固定外设控制。如果引脚配置为数字模式，则PPS输入正常工作。如果使能了同一引脚上的模拟功能，则禁止PPS输入。

11.5.3 控制外设引脚选择

PPS功能通过两组特殊功能寄存器（SFR）进行控制：一组用于映射外设输入，一组用于映射输出。由于输入和输出是单独控制的，因此特定外设的输入和输出（若该外设都有）均可无限制地施加到任何可选的功能引脚上。

根据映射的是输入还是输出，有两种方法可处理外设与外设可选引脚之间的关联。

11.5.3.1 输入映射

外设引脚选择选项的输入根据外设进行映射；即，与外设相关的控制寄存器指示要映射到的引脚。RPINRx寄存器用于配置外设输入映射（见寄存器11-13至寄存器11-31）。

各寄存器均包含一组或两组6位位域，每组都与一个可选引脚的外设相关。使用适当的6位值对指定外设的位域进行编程，会将具有此值的RPn/RPIn引脚映射到该外设。对于任何给定器件，任何位域值的有效范围对应于此器件所支持的外设引脚选择的最大数目。

表11-6： 可选输入源（将输入映射到功能）[(1)]

输入名称	功能名称	寄存器	功能映射位
输出比较触发1	OCTRIG1	RPINR0<5:0>	OCTRIG1R<5:0>
外部中断1	INT1	RPINR0<13:8>	INT1R<5:0>
外部中断2	INT2	RPINR1<5:0>	INT2R<5:0>
外部中断3	INT3	RPINR1<13:8>	INT3R<5:0>
外部中断4	INT4	RPINR2<5:0>	INT4R<5:0>
输出比较触发2	OCTRIG2	RPINR2<13:8>	OCTRIG2R<5:0>
Timer2外部时钟	T2CK	RPINR3<5:0>	T2CKR<5:0>
Timer3外部时钟	T3CK	RPINR3<13:8>	T3CKR<5:0>
输入捕捉1	ICM1	RPINR5<5:0>	ICM1R<5:0>
输入捕捉2	ICM2	RPINR5<13:8>	ICM2R<5:0>
输入捕捉3	ICM3	RPINR6<5:0>	ICM3R<5:0>
输入捕捉4	ICM4	RPINR6<13:8>	ICM4R<5:0>
输入捕捉1	IC1	RPINR7<5:0>	IC1R<5:0>
输入捕捉2	IC2	RPINR7<13:8>	IC2R<5:0>
输入捕捉3	IC3	RPINR8<5:0>	IC3R<5:0>
输出比较故障A	OCFA	RPINR11<5:0>	OCFAR<5:0>
输出比较故障B	OCFB	RPINR11<13:8>	OCFBR<5:0>
CCP时钟输入A	TCKIA	RPINR12<5:0>	TCKIAR<5:0>
CCP时钟输入B	TCKIB	RPINR12<13:8>	TCKIBR<5:0>
UART1接收	U1RX	RPINR18<5:0>	U1RXR<5:0>
UART1允许发送	$\overline{\text{U1CTS}}$	RPINR18<13:8>	U1CTSR<5:0>
UART2接收	U2RX	RPINR19<5:0>	U2RXR<5:0>
UART2允许发送	$\overline{\text{U2CTS}}$	RPINR19<13:8>	U2CTSR<5:0>
SPI1数据输入	SDI1	RPINR20<5:0>	SDI1R<5:0>
SPI1时钟输入	SCK1IN	RPINR20<13:8>	SCK1R<5:0>
SPI1从选择输入	SS1IN	RPINR21<5:0>	SS1R<5:0>
SPI2数据输入	SDI2	RPINR22<5:0>	SDI2R<5:0>
SPI2时钟输入	SCK2IN	RPINR22<13:8>	SCK2R<5:0>
SPI2从选择输入	SS2IN	RPINR23<5:0>	SS2R<5:0>
通用定时器外部时钟	TxCK	RPINR23<13:8>	TXCKR<5:0>
CLC输入A	CLCINA	RPINR25<5:0>	CLCINAR<5:0>
CLC输入B	CLCINB	RPINR25<13:8>	CLCINBR<5:0>
SPI3数据输入	SDI3	RPINR28<5:0>	SDI3R<5:0>
SPI3时钟输入	SCK3IN	RPINR28<13:8>	SCK3R<5:0>
SPI3从选择输入	SS3IN	RPINR29<5:0>	SS3R<5:0>

注 1： 除非另外声明，否则所有输入均使用施密特触发器（ST）输入缓冲器。

11.5.3.2 输出映射

与输入不同，外设引脚选择选项的输出根据引脚进行映射。这种情况下，与特定引脚相关的控制寄存器指示要映射的外设输出。RPORx寄存器用于控制输出映射。每个寄存器均包含两组6位位域；各位域都与一个RPn引脚相关（见寄存器11-32至寄存器11-46）。位域值对应一个外设，该外设的输出映射到引脚（见表11-7）。

由于采用的映射技术，输出映射的外设列表中还包含一个空值000000。此值允许任何给定引脚始终与所有可选引脚的外设的输出断开。

表11-7： 可选输出源（将功能映射到输出）

输出功能编号	功能	输出名称
0	无（引脚已禁止）	—
1	C1OUT	比较器1的输出
2	C2OUT	比较器2的输出
3	U1TX	UART1发送
4	$\overline{\text{U1RTS}}$	UART1请求发送
5	U2TX	UART2发送
6	$\overline{\text{U2RTS}}$	UART2请求发送
7	SDO1	SPI1数据输出
8	SCK1OUT	SPI1时钟输出
9	SS1OUT	SPI1从选择输出
10	SDO2	SPI2数据输出
11	SCK2OUT	SPI2时钟输出
12	SS2OUT	SPI2从选择输出
13	OC1	输出比较1
14	OC2	输出比较2
15	OC3	输出比较3
16	OCM2A	CCP2A输出比较
17	OCM2B	CCP2B输出比较
18	OCM3A	CCP3A输出比较
19	OCM3B	CCP3B输出比较
20	OCM4A	CCP4A输出比较
21	OCM4B	CCP4B输出比较
22	保留	—
23	SDO3	SPI3数据输出
24	SCK3OUT	SPI3时钟输出
25	SS3OUT	SPI3从选择输出
26	C3OUT	比较器3输出
27	PWRGT	RTCC电源控制
28	REFO	参考时钟输出
29	CLC1OUT	CLC1输出
30	CLC2OUT	CLC2输出
31	RTCC	RTCC时钟输出

11.5.3.3 映射限制

外设引脚选择的控制模式相当灵活。除阻止由两个配置为相同功能输入的物理引脚或两个配置为同一引脚的功能输出引起的信号冲突的系统模块外，没有硬件强制执行的锁定。此灵活性可扩展到允许一个输入驱动多个外设或一个功能输出驱动多个输出引脚。

11.5.3.4 PIC24FJ256GA705系列器件的映射例外

表11-8总结了可用的可重映射引脚的差异。

当开发使用可重映射引脚的应用时，用户还应记住以下事项：

- 对于RPINRx寄存器，对应于特定器件的未实现引脚的位组合被视为无效；对应的模块将不具有映射到它的输入。
- 对于RPORx寄存器，对应未实现引脚的位域也未实现；写入这些位域不起作用。

11.5.4 控制配置更改

由于可在运行时更改外设重映射，因此需要对外设重映射设置一些限制条件以防止意外更改配置。PIC24F器件有以下三种用于阻止更改外设映射的功能：

- 控制寄存器锁定序列
- 连续状态监视
- 配置位重映射锁定

11.5.4.1 控制寄存器锁定

正常工作状态下，不允许写RPINRx和RPORx寄存器。尝试的写操作看似正常执行，但寄存器的内容并没有发生变化。要更改这些寄存器的内容，寄存器必须用硬件解锁。寄存器锁定由IOLOCK位（OSCCON<6>）控制。将IOLOCK置1将阻止写入控制寄存器；而将IOLOCK清零将允许写入。

要置1或清零IOLOCK，必须执行以下指定命令序列：

1. 将46h写入OSCCON<7:0>。
2. 将57h写入OSCCON<7:0>。
3. 通过一次操作清零（或置1）IOLOCK。

与振荡器LOCK位的类似序列不同，IOLOCK在更改前一直保持一种状态。这允许使用一个解锁序列对所有外设引脚选择进行配置，然后对所有控制寄存器进行更新，最后用第二个锁定序列锁定。

11.5.4.2 连续状态监视

除了阻止直接写入外，RPINRx和RPORx寄存器的内容还由影子寄存器通过硬件不停地进行监视。如果任何寄存器发生了不希望的更改（例如，由ESD或其他外部事件导致单元值发生误更改），则将触发配置不匹配复位。

11.5.4.3 配置位引脚选择锁定

为了进一步确保安全，可配置器件以阻止对RPINRx和RPORx寄存器执行多次写会话。IOL1WAY（FOSC<5>）配置位会阻止IOLOCK位在置1后再被清零。若IOLOCK保持置1状态，寄存器解锁过程将不会执行，且不能写入外设引脚选择控制寄存器。清零该位并重新使能外设重映射的惟一方法是执行器件复位。

默认（未编程）状态下，IOL1WAY置1，限制用户只能进行一次写会话。编程IOL1WAY允许用户（通过正确地使用解锁序列）对外设引脚选择寄存器进行不受限制的访问。

表11-8： PIC24FJ256GA705系列器件的可重映射引脚的例外情况

器件	RPn引脚（I/O）		RPIn引脚	
	总数	未实现	总数	未实现
PIC24FJXXXGA705	29	—	4	—
PIC24FJXXXGA704	29	—	0	RPI29-32
PIC24FJXXXGA702	18	RP16-25	0	RPI29-32

11.5.5 外设引脚选择注意事项

控制外设引脚选择的功能需要注意应用设计中几个容易被忽视的事项。对于仅可用作可重映射外设的几个常用外设尤其如此。

主要问题是在器件默认（复位）状态下外设引脚选择功能在默认引脚上不可用。由于所有RPINRx寄存器复位为`111111`且所有RPORx寄存器复位为`000000`，因此所有外设引脚选择输入均连接到Vss，而所有外设引脚选择输出均断开连接。

这种情况需要用户在执行任何其他应用代码前使用正确的外设配置初始化器件。由于IOLOCK位复位为解锁状态，所以不必在器件退出复位后执行解锁序列。但是，考虑到应用的安全性，最好在写入控制寄存器之后将IOLOCK位置1并锁定配置。

由于解锁序列对时序有严格要求，因此必须与更改振荡器配置的方式一样，用汇编语言编程。如果大部分应用程序是用C语言或其他高级语言编写的，则应通过编写嵌入汇编执行解锁序列。

选择配置需要查看所有外设引脚选择及其引脚分配，尤其是那些未在应用中使用的引脚。在所有情况下，未用的可选引脚的外设应被完全禁止。未用外设的输入应分配给未用的RPn/RPIn引脚功能。具有未用的RPn功能的I/O引脚应配置为空外设输出。

将外设分配给特定引脚的操作不能自动执行引脚I/O电路的任何其他配置。理论上，也就是说将可选引脚的输出添加到某个引脚意味着在驱动输出时可能会无意中驱动现有的外设输入。用户必须熟悉共用一个可重映射引脚的其他固定外设的行为，了解何时使能或禁止这些外设。为了安全起见，共用同一个引脚的固定数字外设在不用时应禁止。

遵照这些方针，对特定外设的可重映射的引脚进行配置并不会自动使能该功能。必须特别配置外设实现特定的操作并使能外设，如同将外设连接到固定引脚时一样。此操作在应用程序代码中的位置（紧跟在器件复位和外设配置之后或在主应用程序内）取决于外设及其在应用中的使用。

最后一个注意事项是，外设引脚选择功能既不改写模拟输入也不会将具有模拟功能的引脚重新配置为数字I/O。若某个引脚在器件复位时配置为模拟输入，则在使用该引脚的外设引脚选择功能时，必须明确将此引脚重新配置为数字I/O。

例11-4所示为使用UART1实现带有流控制的双向通信的配置。使用了以下输入和输出功能：

- 输入功能：U1RX和$\overline{\text{U1CTS}}$
- 输出功能：U1TX和$\overline{\text{U1RTS}}$

例11-4： 配置UART1输入和输出功能

```
// Unlock Registers
asm volatile    ("MOV   #OSCCON, w1   \n"
                 "MOV   #0x46, w2     \n"
                 "MOV   #0x57, w3     \n"
                 "MOV.b w2, [w1]      \n"
                 "MOV.b w3, [w1]      \n"
                 "BCLR  OSCCON, #6")  ;

// or use XC16 built-in macro:
// __builtin_write_OSCCONL(OSCCON & 0xbf);

// Configure Input Functions (Table 11-6)
   // Assign U1RX To Pin RP0
   RPINR18bits.U1RXR = 0;

   // Assign U1CTS To Pin RP1
   RPINR18bits.U1CTSR = 1;

// Configure Output Functions (Table 11-7)
   // Assign U1TX To Pin RP2
   RPOR1bits.RP2R = 3;

   // Assign U1RTS To Pin RP3
   RPOR1bits.RP3R = 4;

// Lock Registers
asm volatile   ("MOV   #OSCCON, w1   \n"
                "MOV   #0x46, w2     \n"
                "MOV   #0x57, w3     \n"
                "MOV.b w2, [w1]      \n"
                "MOV.b w3, [w1]      \n"
                "BSET  OSCCON, #6")  ;

// or use XC16 built-in macro:
// __builtin_write_OSCCONL(OSCCON | 0x40);
```

11.5.6 外设引脚选择寄存器

PIC24FJ256GA705系列器件共实现了34个寄存器用于配置可重映射的外设：

- 输入可重映射外设寄存器（19）
- 输出可重映射外设寄存器（15）

注： 仅在IOLOCK（OSCCON<6>）= 0时才能更改输入和输出寄存器的值。请参见第11.5.4.1节“控制寄存器锁定”了解特定的命令序列。

寄存器11-13： RPINR0：外设引脚选择输入寄存器0

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCTRIG1R5	OCTRIG1R4	OCTRIG1R3	OCTRIG1R2	OCTRIG1R1	OCTRIG1R0
bit 7							bit 0

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

bit 15-14	未实现：读为0
bit 13-8	**INT1R<5:0>**：将外部中断1（INT1）分配给相应的RPn或RPIn引脚的位
bit 7-6	未实现：读为0
bit 5-0	**OCTRIG1R<5:0>**：将输出比较触发1分配给相应的RPn或RPIn引脚的位

寄存器11-14： RPINR1：外设引脚选择输入寄存器1

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0
bit 7							bit 0

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

bit 15-14	未实现：读为0
bit 13-8	**INT3R<5:0>**：将外部中断3（INT3）分配给相应的RPn或RPIn引脚的位
bit 7-6	未实现：读为0
bit 5-0	**INT2R<5:0>**：将外部中断2（INT2）分配给相应的RPn或RPIn引脚的位

寄存器11-15： RPINR2：外设引脚选择输入寄存器2

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCTRIG2R5	OCTRIG2R4	OCTRIG2R3	OCTRIG2R2	OCTRIG2R1	OCTRIG2R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT4R5	INT4R4	INT4R3	INT4R2	INT4R1	INT4R0
bit 7							bit 0

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

bit 15-14 未实现：读为0

bit 13-8 **OCTRIG2R<5:0>**：将输出比较触发2分配给相应的RPn或RPIn引脚的位

bit 7-6 未实现：读为0

bit 5-0 **INT4R<5:0>**：将外部中断4（INT4）分配给相应的RPn或RPIn引脚的位

寄存器11-16： RPINR3：外设引脚选择输入寄存器3

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T3CKR5	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T2CKR5	T2CKR4	T2CKR3	T2CKR2	T2CKR1	T2CKR0
bit 7							bit 0

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

bit 15-14 未实现：读为0

bit 13-8 **T3CKR<5:0>**：将Timer3时钟分配给相应的RPn或RPIn引脚的位

bit 7-6 未实现：读为0

bit 5-0 **T2CKR<5:0>**：将Timer2时钟分配给相应的RPn或RPIn引脚的位

寄存器11-17： RPINR5：外设引脚选择输入寄存器5

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	ICM2R5	ICM2R4	ICM2R3	ICM2R2	ICM2R1	ICM2R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	ICM1R5	ICM1R4	ICM1R3	ICM1R2	ICM1R1	ICM1R0
bit 7							bit 0

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

bit 15-14 未实现：读为0

bit 13-8 **ICM2R<5:0>**：输入捕捉模式2位

bit 7-6 未实现：读为0

bit 5-0 **ICM1R<5:0>**：输入捕捉模式1位

寄存器11-18： RPINR6：外设引脚选择输入寄存器6

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	ICM4R5	ICM4R4	ICM4R3	ICM4R2	ICM4R1	ICM4R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	ICM3R5	ICM3R4	ICM3R3	ICM3R2	ICM3R1	ICM3R0
bit 7							bit 0

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

bit 15-14 未实现：读为0

bit 13-8 **ICM4R<5:0>**：输入捕捉模式4位

bit 7-6 未实现：读为0

bit 5-0 **ICM3R<5:0>**：输入捕捉模式3位

寄存器11-19： RPINR7：外设引脚选择输入寄存器7

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC2R5	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC1R5	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0
bit 7							bit 0

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

bit 15-14	**未实现：**读为0
bit 13-8	**IC2R<5:0>：**将输入捕捉2（IC2）分配给相应的RPn或RPIn引脚的位
bit 7-6	**未实现：**读为0
bit 5-0	**IC1R<5:0>：**将输入捕捉1（IC1）分配给相应的RPn或RPIn引脚的位

寄存器11-20： RPINR8：外设引脚选择输入寄存器8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC3R5	IC3R4	IC3R3	IC3R2	IC3R1	IC3R0
bit 7							bit 0

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

bit 15-6	**未实现：**读为0
bit 5-0	**IC3R<5:0>：**将输入捕捉3（IC3）分配给相应的RPn或RPIn引脚的位

寄存器11-21： RPINR11：外设引脚选择输入寄存器11

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-8 **OCFBR<5:0>：**将输出比较故障B（OCFB）分配给相应的RPn或RPIn引脚的位

bit 7-6 **未实现：**读为0

bit 5-0 **OCFAR<5:0>：**将输出比较故障A（OCFA）分配给相应的RPn或RPIn引脚的位

寄存器11-22： RPINR12：外设引脚选择输入寄存器12

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	TCKIBR5	TCKIBR4	TCKIBR3	TCKIBR2	TCKIBR1	TCKIBR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	TCKIAR5	TCKIAR4	TCKIAR3	TCKIAR2	TCKIAR1	TCKIAR0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-8 **TCKIBR<5:0>：**将MCCP/SCCP时钟输入B分配给相应的RPn或RPIn引脚的位

bit 7-6 **未实现：**读为0

bit 5-0 **TCKIAR<5:0>：**将MCCP/SCCP时钟输入A分配给相应的RPn或RPIn引脚的位

寄存器11-23： **RPINR18：外设引脚选择输入寄存器18**

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0
bit 7							bit 0

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

bit 15-14 未实现：读为0

bit 13-8 **U1CTSR<5:0>**：将UART1允许发送（$\overline{\text{U1CTS}}$）分配给相应的RPn或RPIn引脚的位

bit 7-6 未实现：读为0

bit 5-0 **U1RXR<5:0>**：将UART1接收（U1RX）分配给相应的RPn或RPIn引脚的位

寄存器11-24： **RPINR19：外设引脚选择输入寄存器19**

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U2RXR5	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0
bit 7							bit 0

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

bit 15-14 未实现：读为0

bit 13-8 **U2CTSR<5:0>**：将UART2允许发送（$\overline{\text{U2CTS}}$）分配给相应的RPn或RPIn引脚的位

bit 7-6 未实现：读为0

bit 5-0 **U2RXR<5:0>**：将UART2接收（U2RX）分配给相应的RPn或RPIn引脚的位

寄存器11-25： RPINR20：外设引脚选择输入寄存器20

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SDI1R5	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0
bit 7							bit 0

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

bit 15-14 **未实现**：读为0

bit 13-8 **SCK1R<5:0>**：将SPI1时钟输入（SCK1IN）分配给相应的RPn或RPIn引脚的位

bit 7-6 **未实现**：读为0

bit 5-0 **SDI1R<5:0>**：将SPI1数据输入（SDI1）分配给相应的RPn或RPIn引脚的位

寄存器11-26： RPINR21：外设引脚选择输入寄存器21

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0
bit 7							bit 0

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

bit 15-6 **未实现**：读为0

bit 5-0 **SS1R<5:0>**：将SPI1从选择输入（SS1IN）分配给相应的RPn或RPIn引脚的位

寄存器11-27： RPINR22：外设引脚选择输入寄存器22

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0
bit 7							bit 0

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

bit 15-14　**未实现：**读为0

bit 13-8　**SCK2R<5:0>：**将SPI2时钟输入（SCK2IN）分配给相应的RPn或RPIn引脚的位

bit 7-6　**未实现：**读为0

bit 5-0　**SDI2R<5:0>：**将SPI2数据输入（SDI2）分配给相应的RPn或RPIn引脚的位

寄存器11-28： RPINR23：外设引脚选择输入寄存器23

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	TXCKR5	TXCKR4	TXCKR3	TXCKR2	TXCKR1	TXCKR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SS2R5	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0
bit 7							bit 0

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

bit 15-14　**未实现：**读为0

bit 13-8　**TXCKR<5:0>：**将通用定时器外部输入（TxCK）分配给相应的RPn或RPIn引脚的位

bit 7-6　**未实现：**读为0

bit 5-0　**SS2R<5:0>：**将SPI2从选择输入（SS2IN）分配给相应的RPn或RPIn引脚的位

寄存器11-29： RPINR25：外设引脚选择输入寄存器25

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	CLCINBR5	CLCINBR4	CLCINBR3	CLCINBR2	CLCINBR1	CLCINBR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	CLCINAR5	CLCINAR4	CLCINAR3	CLCINAR2	CLCINAR1	CLCINAR0
bit 7							bit 0

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

bit 15-14 未实现：读为0

bit 13-8 **CLCINBR<5:0>**：将CLC输入B分配给相应的RPn或RPIn引脚的位

bit 7-6 未实现：读为0

bit 5-0 **CLCINAR<5:0>**：将CLC输入A分配给相应的RPn或RPIn引脚的位

寄存器11-30： RPINR28：外设引脚选择输入寄存器28

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK3R5	SCK3R4	SCK3R3	SCK3R2	SCK3R1	SCK3R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SDI3R5	SDI3R4	SDI3R3	SDI3R2	SDI3R1	SDI3R0
bit 7							bit 0

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

bit 15-14 未实现：读为0

bit 13-8 **SCK3R<5:0>**：将SPI3时钟输入（SCK3IN）分配给相应的RPn或RPIn引脚的位

bit 7-6 未实现：读为0

bit 5-0 **SDI3R<5:0>**：将SPI3数据输入（SDI3）分配给相应的RPn或RPIn引脚的位

寄存器11-31： RPINR29：外设引脚选择输入寄存器29

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SS3R5	SS3R4	SS3R3	SS3R2	SS3R1	SS3R0
bit 7							bit 0

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

bit 15-6 **未实现：**读为0

bit 5-0 **SS3R<5:0>：**将SPI3从选择输入（SS3IN）分配给相应的RPn或RPIn引脚的位

寄存器11-32： RPOR0：外设引脚选择输出寄存器0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP1R5	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP0R5	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-8 **RP1R<5:0>：**RP1输出引脚映射位
将外设输出编号n分配给引脚RP1（请参见表11-7了解外设功能编号）。

bit 7-6 **未实现：**读为0

bit 5-0 **RP0R<5:0>：**RP0输出引脚映射位
将外设输出编号n分配给引脚RP0（请参见表11-7了解外设功能编号）。

寄存器11-33： RPOR1：外设引脚选择输出寄存器1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP3R5	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP2R5	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-8 **RP3R<5:0>：**RP3输出引脚映射位
将外设输出编号n分配给引脚RP3（请参见表11-7了解外设功能编号）。

bit 7-6 **未实现：**读为0

bit 5-0 **RP2R<5:0>：**RP2输出引脚映射位
将外设输出编号n分配给引脚RP2（请参见表11-7了解外设功能编号）。

寄存器11-34： RPOR2：外设引脚选择输出寄存器2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP5R5	RP5R4	RP5R3	RP5R2	RP5R1	RP5R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP4R5	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-8 **RP5R<5:0>：**RP5输出引脚映射位
将外设输出编号n分配给引脚RP5（请参见表11-7了解外设功能编号）。

bit 7-6 **未实现：**读为0

bit 5-0 **RP4R<5:0>：**RP4输出引脚映射位
将外设输出编号n分配给引脚RP4（请参见表11-7了解外设功能编号）。

寄存器11-35： RPOR3：外设引脚选择输出寄存器3

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP7R5	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP6R5	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-8 **RP7R<5:0>：**RP7输出引脚映射位
将外设输出编号n分配给引脚RP7（请参见表11-7了解外设功能编号）。

bit 7-6 **未实现：**读为0

bit 5-0 **RP6R<5:0>：**RP6输出引脚映射位
将外设输出编号n分配给引脚RP6（请参见表11-7了解外设功能编号）。

寄存器11-36： RPOR4：外设引脚选择输出寄存器4

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP9R5	RP9R4	RP9R3	RP9R2	RP9R1	RP9R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP8R5	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0
bit 7							bit 0

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

bit 15-14 **未实现**：读为0

bit 13-8 **RP9R<5:0>**：RP9输出引脚映射位
将外设输出编号n分配给引脚RP9（请参见表11-7了解外设功能编号）。

bit 7-6 **未实现**：读为0

bit 5-0 **RP8R<5:0>**：RP8输出引脚映射位
将外设输出编号n分配给引脚RP8（请参见表11-7了解外设功能编号）。

寄存器11-37： RPOR5：外设引脚选择输出寄存器5

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP11R5	RP11R4	RP11R3	RP11R2	RP11R1	RP11R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP10R5	RP10R4	RP10R3	RP10R2	RP10R1	RP10R0
bit 7							bit 0

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

bit 15-14 **未实现**：读为0

bit 13-8 **RP11R<5:0>**：RP11输出引脚映射位
将外设输出编号n分配给引脚RP11（请参见表11-7了解外设功能编号）。

bit 7-6 **未实现**：读为0

bit 5-0 **RP10R<5:0>**：RP10输出引脚映射位
将外设输出编号n分配给引脚RP10（请参见表11-7了解外设功能编号）。

寄存器11-38： RPOR6：外设引脚选择输出寄存器6

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP13R5	RP13R4	RP13R3	RP13R2	RP13R1	RP13R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP12R5	RP12R4	RP12R3	RP12R2	RP12R1	RP12R0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-8 **RP13R<5:0>：**RP13输出引脚映射位
将外设输出编号n分配给引脚RP13（请参见表11-7了解外设功能编号）。

bit 7-6 **未实现：**读为0

bit 5-0 **RP12R<5:0>：**RP12输出引脚映射位
将外设输出编号n分配给引脚RP12（请参见表11-7了解外设功能编号）。

寄存器11-39： RPOR7：外设引脚选择输出寄存器7

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP15R5	RP15R4	RP15R3	RP15R2	RP15R1	RP15R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP14R5	RP14R4	RP14R3	RP14R2	RP14R1	RP14R0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-8 **RP15R<5:0>：**RP15输出引脚映射位
将外设输出编号n分配给引脚RP15（请参见表11-7了解外设功能编号）。

bit 7-6 **未实现：**读为0

bit 5-0 **RP14R<5:0>：**RP14输出引脚映射位
将外设输出编号n分配给引脚RP14（请参见表11-7了解外设功能编号）。

寄存器11-40： RPOR8：外设引脚选择输出寄存器8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP17R5	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP16R5	RP16R4	RP16R3	RP16R2	RP16R1	RP16R0
bit 7							bit 0

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

bit 15-14 未实现：读为0

bit 13-8 **RP17R<5:0>**：RP17输出引脚映射位
将外设输出编号n分配给引脚RP17（请参见表11-7了解外设功能编号）。

bit 7-6 未实现：读为0

bit 5-0 **RP16R<5:0>**：RP16输出引脚映射位
将外设输出编号n分配给引脚RP16（请参见表11-7了解外设功能编号）。

寄存器11-41： RPOR9：外设引脚选择输出寄存器9

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP19R5	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP18R5	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0
bit 7							bit 0

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

bit 15-14 未实现：读为0

bit 13-8 **RP19R<5:0>**：RP19输出引脚映射位
将外设输出编号n分配给引脚RP19（请参见表11-7了解外设功能编号）。

bit 7-6 未实现：读为0

bit 5-0 **RP18R<5:0>**：RP18输出引脚映射位
将外设输出编号n分配给引脚RP18（请参见表11-7了解外设功能编号）。

寄存器11-42： RPOR10：外设引脚选择输出寄存器10

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP21R5	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP20R5	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-8 **RP21R<5:0>：**RP21输出引脚映射位
将外设输出编号n分配给引脚RP21（请参见表11-7了解外设功能编号）。

bit 7-6 **未实现：**读为0

bit 5-0 **RP20R<5:0>：**RP20输出引脚映射位
将外设输出编号n分配给引脚RP20（请参见表11-7了解外设功能编号）。

寄存器11-43： RPOR11：外设引脚选择输出寄存器11

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP23R5	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP22R5	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-8 **RP23R<5:0>：**RP23输出引脚映射位
将外设输出编号n分配给引脚RP23（请参见表11-7了解外设功能编号）。

bit 7-6 **未实现：**读为0

bit 5-0 **RP22R<5:0>：**RP22输出引脚映射位
将外设输出编号n分配给引脚RP22（请参见表11-7了解外设功能编号）。

寄存器11-44： RPOR12：外设引脚选择输出寄存器12

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP25R5	RP25R4	RP25R3	RP25R2	RP25R1	RP25R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP24R5	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-8 **RP25R<5:0>：**RP25输出引脚映射位
将外设输出编号n分配给引脚RP25（请参见表11-7了解外设功能编号）。

bit 7-6 **未实现：**读为0

bit 5-0 **RP24R<5:0>：**RP24输出引脚映射位
将外设输出编号n分配给引脚RP24（请参见表11-7了解外设功能编号）。

寄存器11-45： RPOR13：外设引脚选择输出寄存器13

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP27R5	RP27R4	RP27R3	RP27R2	RP27R1	RP27R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP26R5	RP26R4	RP26R3	RP26R2	RP26R1	RP26R0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-8 **RP27R<5:0>：**RP27输出引脚映射位
将外设输出编号n分配给引脚RP27（请参见表11-7了解外设功能编号）。

bit 7-6 **未实现：**读为0

bit 5-0 **RP26R<5:0>：**RP26输出引脚映射位
将外设输出编号n分配给引脚RP26（请参见表11-7了解外设功能编号）。

寄存器11-46： RPOR14：外设引脚选择输出寄存器14

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP28R5	RP28R4	RP28R3	RP28R2	RP28R1	RP28R0
bit 7							bit 0

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

bit 15-6 **未实现：**读为0

bit 5-0 **RP28R<5:0>：**RP28输出引脚映射位
将外设输出编号n分配给引脚RP28（请参见表11-7了解外设功能编号）。

12.0 TIMER1

> **注：** 本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的**"定时器"**（DS39704），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

Timer1模块是一个16位定时器，可作为实时时钟（RTC）的时间计数器，或作为独立运行的间隔定时器/计数器。Timer1可在以下三种模式下工作：

- 16位定时器
- 16位同步计数器
- 16位异步计数器

而且，Timer1还支持下列特性：

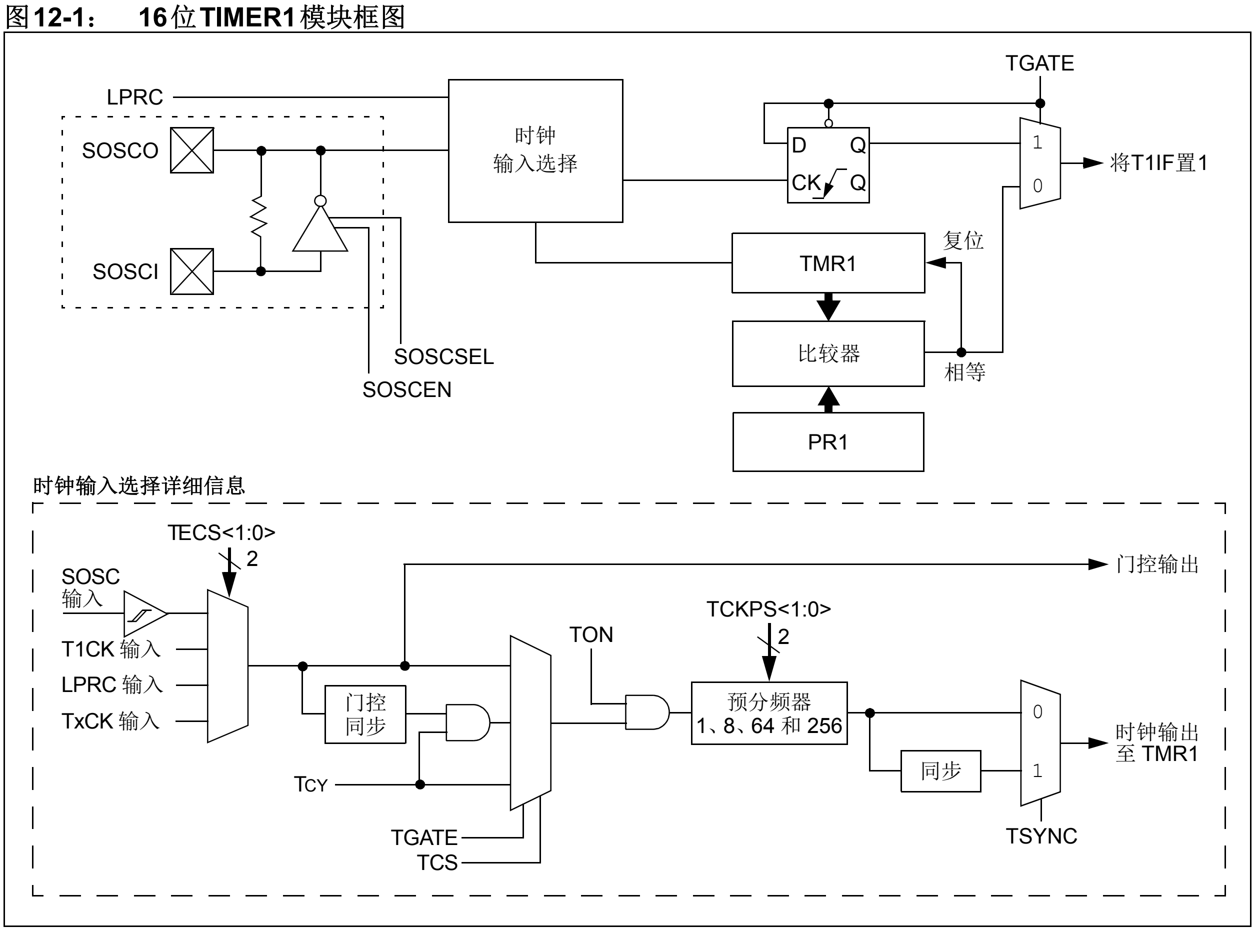
- 定时器门控操作
- 可选择的预分频比设置
- CPU空闲和休眠模式下的定时器操作
- 在16位周期寄存器匹配或外部门控信号的下降沿产生中断

图12-1给出了16位定时器模块的框图。

配置Timer1的步骤：

1. 将TON位置1（= 1）。
2. 使用TCKPS<1:0>位选择定时器预分频比。
3. 使用TCS、TECS<1:0>和TGATE位设置时钟和门控模式。
4. 将TSYNC位置1或清零以配置同步或异步操作。
5. 将定时器周期值装入PR1寄存器。
6. 如果需要中断，则将中断允许位T1IE置1。使用优先级位T1IP<2:0>来设置中断的优先级。

图12-1： 16位TIMER1模块框图



寄存器12-1： T1CON：TIMER1控制寄存器[1]

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
TON	—	TSIDL	—	—	—	TECS1	TECS0
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—
bit 7							bit 0

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

bit 15 **TON**：Timer1使能位
1 = 启动16位Timer1
0 = 停止16位Timer1

bit 14 **未实现**：读为0

bit 13 **TSIDL**：Timer1空闲模式停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 在空闲模式下模块继续工作

bit 12-10 **未实现**：读为0

bit 9-8 **TECS<1:0>**：Timer1扩展时钟源选择位（当TCS = 1时选择）
11 = 通用定时器（TxCK）外部输入
10 = LPRC振荡器
01 = T1CK外部时钟输入
00 = SOSC

bit 7 **未实现**：读为0

bit 6 **TGATE**：Timer1门控时间累加使能位
当TCS = 1时：
该位被忽略。
当TCS = 0时：
1 = 使能门控时间累加
0 = 禁止门控时间累加

bit 5-4 **TCKPS<1:0>**：Timer1输入时钟预分频比选择位
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3 **未实现**：读为0

bit 2 **TSYNC**：Timer1外部时钟输入同步选择位
当TCS = 1时：
1 = 同步外部时钟输入
0 = 不同步外部时钟输入
当TCS = 0时：
该位被忽略。

bit 1 **TCS**：Timer1时钟源选择位
1 = 由定时器选择扩展时钟
0 = 内部时钟（Fosc/2）

bit 0 **未实现**：读为0

注 1： 定时器运行（TON = 1）时改变T1CON的值会导致定时器预分频计数器复位，不推荐这么做。

13.0 TIMER2/3

注：	本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的**“定时器”**（DS39704），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

Timer2/3模块为32位定时器，也可被配置为具有可选工作模式的独立16位定时器。

作为32位定时器，Timer2/3具有以下三种工作模式：

- 两个独立的16位定时器，可实现所有16位工作模式（异步计数器模式除外）
- 单个32位定时器
- 单个32位同步计数器

这些定时器还支持以下功能：

- 定时器门控操作
- 可选择的预分频比设置
- 空闲和休眠模式期间定时器工作
- 在32位周期寄存器匹配时产生中断
- A/D事件触发信号（32位模式的Timer2/3和16位模式的Timer3）

所有16位定时器都能单独用作同步定时器或计数器。它们也可以提供上面列出的功能，但A/D事件触发信号除外。该触发信号只能在32位模式的Timer2/3和16位模式的Timer3上实现。通过设置T2CON和T3CON寄存器中的相应位来确定工作模式和使能特性。寄存器13-1给出了T2CON的一般形式；寄存器13-2给出了T3CON的一般形式。

对于32位定时器/计数器操作，Timer2是32位定时器的最低有效字；Timer3则是32位定时器的最高有效字。

注：	对于32位操作，T3CON中的控制位将被忽略。设置和控制只使用T2CON中的控制位。32位定时器模块采用Timer2的时钟和门控输入，但中断由Timer3中断标志产生。

要将Timer2/3配置为32位工作模式：

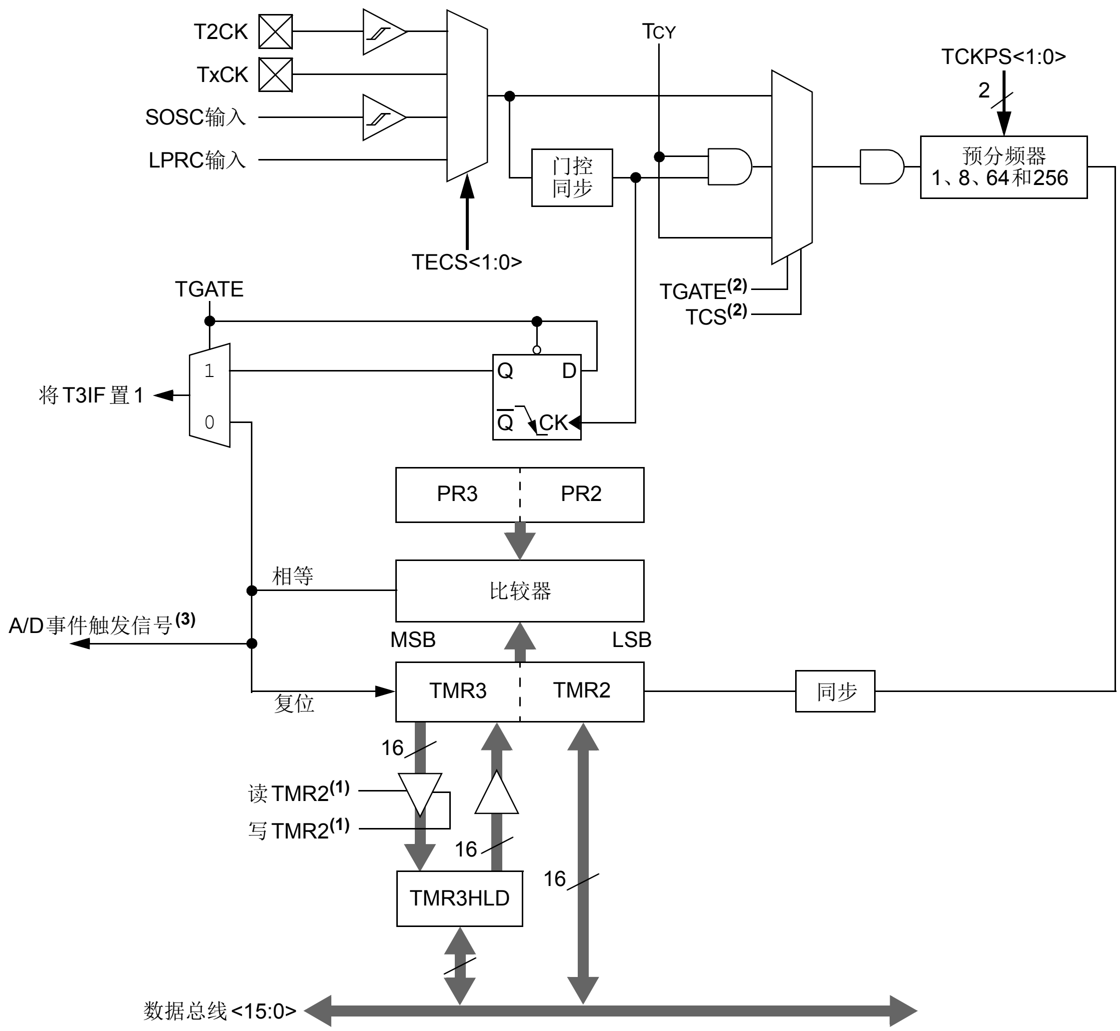
1. 将T32位置1（T2CON<3> = 1）。
2. 使用TCKPS<1:0>位选择Timer2的预分频比。
3. 使用TCS和TGATE位设置时钟和门控模式。如果TCS被设置为外部时钟，则必须将RPINRx（TyCK）配置给可用的RPn/RPIn引脚。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。
4. 装载定时器周期值。PR3将存放该值的最高有效字（msw），而PR2则存放最低有效字（lsw）。
5. 如果需要中断，则将中断允许位T3IE置1。使用优先级位T3IP<2:0>来设置中断的优先级。请注意，虽然Timer2控制定时器，但中断却来自于Timer3中断。
6. 将TON位置1（= 1）。

任意时刻的定时器值都将存储在寄存器对TMR<3:2>中。TMR3始终存放计数值的最高有效字，而TMR2则存放最低有效字。

要将任何定时器配置为独立的16位工作模式：

1. 将T32位（T2CON<3>）清零。
2. 使用TCKPS<1:0>位选择定时器预分频比。
3. 使用TCS和TGATE位设置时钟和门控模式。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。
4. 将定时器周期值装入PRx寄存器。
5. 如果需要中断，则将中断允许位TxIE置1。使用优先级位TxIP<2:0>来设置中断的优先级。
6. 将TON位置1（TxCON<15> = 1）。

图13-1： TIMER2/3（32位）框图

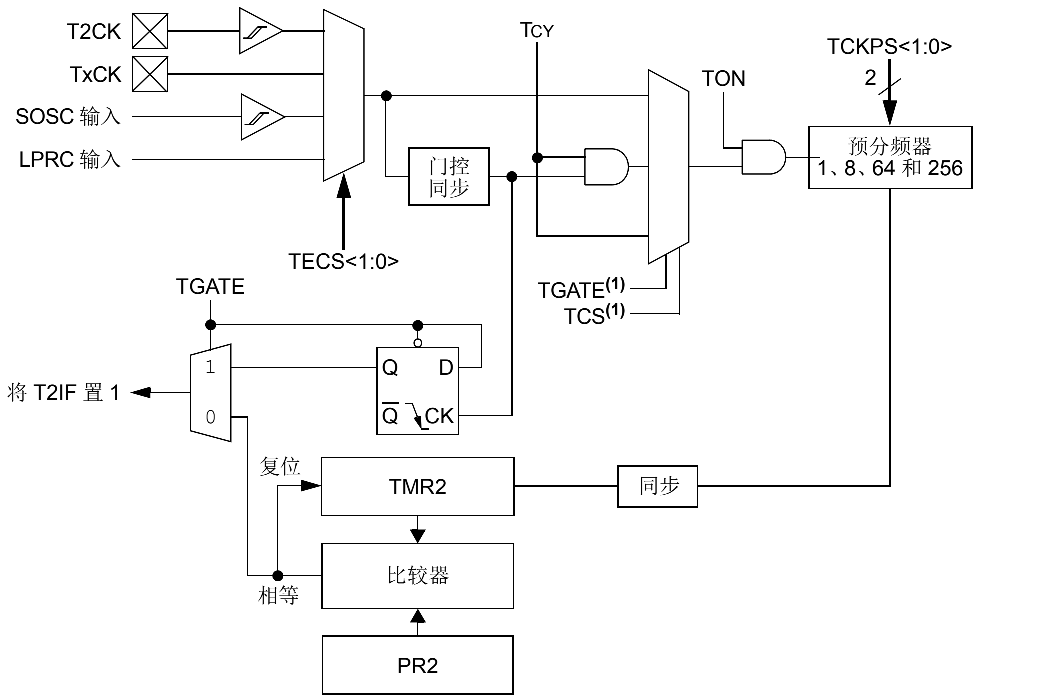


注 1：必须将32位定时器配置位T32置1才能使定时器工作在32位定时器/计数器模式下。所有控制位都在T2CON寄存器中。

2：在使用定时器时钟输入之前，必须将其分配给可用的RPn/RPIn引脚。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。

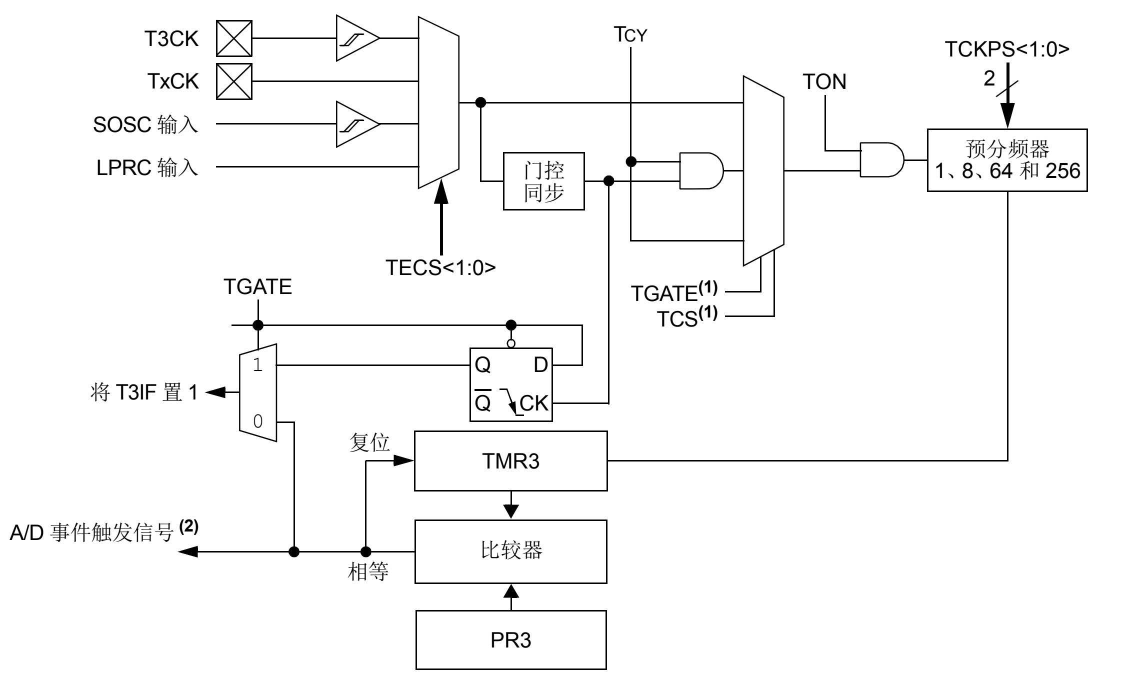
3：只有32位模式的Timer2/3和16位模式的Timer3可以产生A/D事件触发信号。

图 13-2： TIMER2（16 位同步）框图



注 1： 在使用定时器时钟输入之前，必须将其分配给可用的 RPn/RPIn 引脚。更多信息，请参见第 11.5 节 “外设引脚选择（PPS）”。

图 13-3： TIMER3（16 位异步）框图



注 1： 在使用定时器时钟输入之前，必须将其分配给可用的 RPn/RPIn 引脚。更多信息，请参见第 11.5 节 “外设引脚选择（PPS）”。
2： 只有 Timer3 可以产生 A/D 事件触发信号。

寄存器13-1： TxCON：TIMER2控制寄存器[1]

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
TON	—	TSIDL	—	—	—	TECS1[2]	TECS0[2]
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	T32[3]	—	TCS[2]	—
bit 7							bit 0

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

bit 15 **TON**：Timerx使能位
当TxCON<3> = 1时：
1 = 启动32位Timerx/y
0 = 停止32位Timerx/y
当TxCON<3> = 0时：
1 = 启动16位Timerx
0 = 停止16位Timerx

bit 14 **未实现**：读为0

bit 13 **TSIDL**：Timerx空闲模式停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 在空闲模式下模块继续工作

bit 12-10 **未实现**：读为0

bit 9-8 **TECS<1:0>**：Timerx扩展时钟源选择位（当TCS = 1时选择）[2]
当TCS = 1时：
11 = 通用定时器（TxCK）外部输入
10 = LPRC振荡器
01 = TyCK外部时钟输入
00 = SOSC
当TCS = 0时：
忽略这两位；定时器时钟由内部系统时钟（Fosc/2）提供。

bit 7 **未实现**：读为0

bit 6 **TGATE**：Timerx门控时间累加使能位
当TCS = 1时：
该位被忽略。
当TCS = 0时：
1 = 使能门控时间累加
0 = 禁止门控时间累加

bit 5-4 **TCKPS<1:0>**：Timerx输入时钟预分频比选择位
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

注 1： 定时器运行（TON = 1）时改变TxCON的值会导致定时器预分频计数器复位，不推荐这么做。
2： 如果TCS = 1且TECS<1:0> = x1，则必须将所选外部定时器输入（TxCK或TyCK）配置给可用的RPn/RPIn引脚。更多信息，请参见第11.5节“外设引脚选择（PPS）”。
3： 在32位模式下，T3CON控制位不影响32位定时器的工作。

寄存器13-1： TxCON：TIMER2控制寄存器[1]（续）

bit 3 **T32**：32位定时器模式选择位[3]

1 = Timerx和Timery形成一个32位定时器

0 = Timerx和Timery用作两个16位定时器

在32位模式下，T3CON控制位不影响32位定时器的工作。

bit 2 **未实现**：读为0

bit 1 **TCS**：Timerx时钟源选择位[2]

1 = 定时器时钟源由TECS<1:0>选择

0 = 内部时钟（Fosc/2）

bit 0 **未实现**：读为0

注 1： 定时器运行（TON = 1）时改变TxCON的值会导致定时器预分频计数器复位，不推荐这么做。

2： 如果TCS = 1且TECS<1:0> = x1，则必须将所选外部定时器输入（TxCK或TyCK）配置给可用的RPn/RPIn引脚。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。

3： 在32位模式下，T3CON控制位不影响32位定时器的工作。

寄存器13-2： TyCON：TIMER3控制寄存器[1]

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
TON[2]	—	TSIDL[2]	—	—	—	TECS1[2,3]	TECS0[2,3]
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE[2]	TCKPS1[2]	TCKPS0[2]	—	—	TCS[2,3]	—
bit 7							bit 0

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

bit 15 **TON**：Timery使能位[2]
1 = 启动16位Timery
0 = 停止16位Timery

bit 14 **未实现**：读为0

bit 13 **TSIDL**：Timery空闲模式停止位[2]
1 = 当器件进入空闲模式时，模块停止工作
0 = 在空闲模式下模块继续工作

bit 12-10 **未实现**：读为0

bit 9-8 **TECS<1:0>**：Timery扩展时钟源选择位（当TCS = 1时选择）[2,3]
11 = 通用定时器（TxCK）外部输入
10 = LPRC振荡器
01 = TyCK外部时钟输入
00 = SOSC

bit 7 **未实现**：读为0

bit 6 **TGATE**：Timery门控时间累加使能位[2]
当TCS = 1时：
该位被忽略。
当TCS = 0时：
1 = 使能门控时间累加
0 = 禁止门控时间累加

bit 5-4 **TCKPS<1:0>**：Timery输入时钟预分频比选择位[2]
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3-2 **未实现**：读为0

bit 1 **TCS**：Timery时钟源选择位[2,3]
1 = 来自TyCK引脚的外部时钟（上升沿）
0 = 内部时钟（Fosc/2）

bit 0 **未实现**：读为0

注 1： 定时器运行（TON = 1）时改变TyCON的值会导致定时器预分频计数器复位，不推荐这么做。
2： 当使能32位操作（T2CON<3> = 1）时，该位对Timery的操作没有影响；所有定时器功能都通过T2CON进行设置。
3： 如果TCS = 1且TECS<1:0> = x1，则必须将所选外部定时器输入（TyCK）配置给可用的RPn/RPIn引脚。更多信息，请参见第11.5节“外设引脚选择（PPS）”。

14.0 带专用定时器的输入捕捉

注：	本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的**“带专用定时器的输入捕捉”**（DS70000352），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

PIC24FJ256GA705系列中的器件包含3个独立的输入捕捉模块。每个模块都为捕捉外部脉冲事件和产生中断提供了各种配置和工作选项。

输入捕捉模块的主要特性包括：

- 可通过级联两个相邻的模块，采用硬件方式配置为所有32位工作模式
- 输出比较操作有同步和触发两种模式，最多具有31个用户可选的同步/触发源
- 一个4级深的FIFO缓冲区，用于捕捉并保存多个事件的定时器值
- 中断产生可配置
- 每个模块可使用最多6个时钟源来驱动独立的内部16位计数器

此模块可通过两个寄存器进行控制：ICxCON1（寄存器14-1）和ICxCON2（寄存器14-2）。模块的一般框图如图14-1所示。

14.1 一般工作模式

14.1.1 同步和触发模式

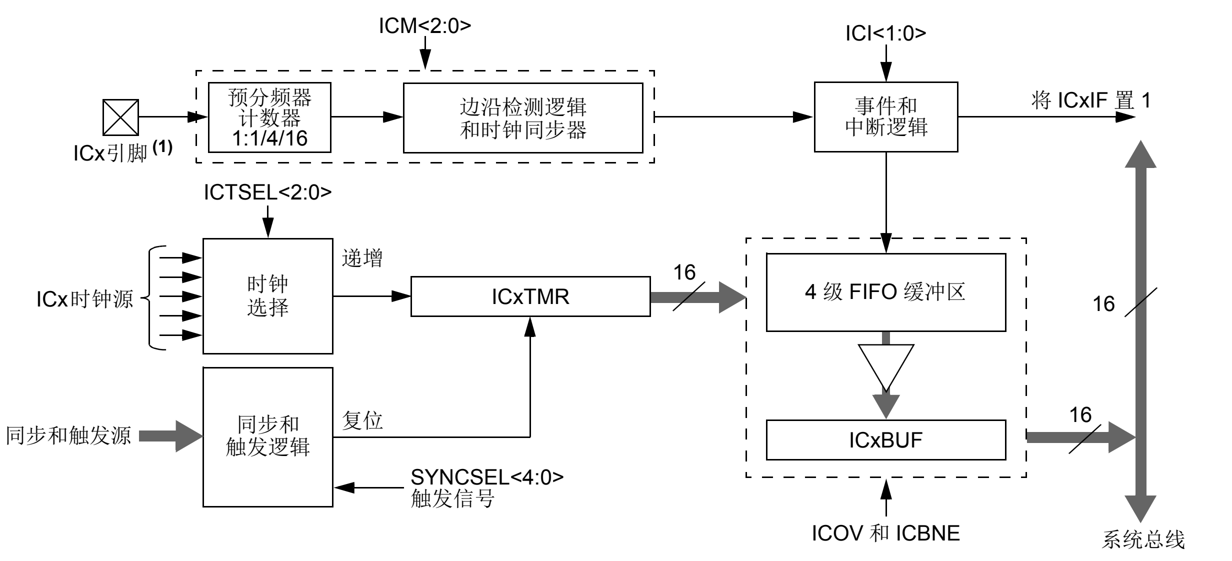
输入捕捉模块工作在自由运行模式下时，内部16位计数器ICxTMR连续递增计数，并在每次溢出时从FFFFh折回0000h。其周期与所选的外部时钟源同步。发生捕捉事件时，将内部计数器的当前16位值写入FIFO缓冲区。

在同步模式下，一旦使能了所选的时钟源，模块就开始捕捉ICx引脚上的事件。只要所选的同步源发生事件，内部计数器就会复位。在触发模式下，模块等待来自其他内部模块的同步事件发生后，才允许内部计数器运行。

通过将SYNCSEL<4:0>位（ICxCON2<4:0>）设置为00000并清零ICTRIG位（ICxCON2<7>）选择标准的自由运行工作模式。当SYNCSELx位设置为除00000以外的其他值时，将选择同步和触发模式。ICTRIG位选择同步或触发模式；置1该位将选择触发工作模式。在两种模式下，SYNCSELx位用于决定同步/触发源。

当SYNCSELx位设置为00000且ICTRIG置1时，模块工作在软件触发模式下。在此情况下，可通过手动将TRIGSTAT位（ICxCON2<6>）置1来启动捕捉操作。

图14-1： 输入捕捉x框图



注 1： ICx输入必须在使用前分配给可用的RPn/RPIn引脚。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。

14.1.2 级联（32位）模式

默认情况下，每个模块都可以使用自己的16位定时器独立工作。要提高分辨率，可将相邻的偶数编号的模块和奇数编号的模块配置为一个32位模块。（例如，模块1和2配对，模块3和4配对等）。奇数编号的模块（ICx）提供32位寄存器对中的低16位，而偶数编号的模块（ICy）提供高16位。ICx寄存器的折回操作会导致其相应的ICy寄存器递增。

通过将两个模块的IC32位（ICxCON2<8>）都置1用硬件配置级联操作。

14.2 捕捉操作

输入捕捉模块可配置为在ICx信号的上升沿或所有跳变沿处捕捉定时器值并产生中断。捕捉可配置在所有上升沿上发生，或仅在某些上升沿（例如每4个或16个上升沿）发生。可单独配置中断以在每个事件或一组事件发生时产生中断。

将模块设置为捕捉操作：

1. 将ICx输入配置为可用的外设引脚选择引脚之一。
2. 如果要使用同步模式，应在继续前禁止同步源。
3. 确保ICBNE位（ICxCON1<3>）清零之前已通过读ICxBUF从FIFO中除去了所有以前的数据。
4. 设置SYNCSELx位（ICxCON2<4:0>）以选择所需的同步/触发源。
5. 设置ICTSELx位（ICxCON1<12:10>）以选择所需的时钟源。
6. 设置ICIx位（ICxCON1<6:5>）以选择所需的中断频率。
7. 选择同步或触发工作模式：
   a) 检查SYNCSELx位是否设置为00000。
   b) 对于同步模式，清零ICTRIG位（ICxCON2<7>）。
   c) 对于触发模式，将ICTRIG位置1并清零TRIGSTAT位（ICxCON2<6>）。
8. 设置ICMx位（ICxCON1<2:0>）以得到所需的工作模式。
9. 使能所选的同步/触发源。

对于32位级联操作，设置过程稍有不同：

1. 将两个模块的IC32位（ICyCON2<8>和ICxCON2<8>）置1，首先使能偶数编号的模块。这可以确保两个模块一起开始工作。
2. 分别设置两个模块的ICTSELx和SYNCSELx位以选择相同的同步/触发源和时基源。首先设置偶数编号的模块，然后设置奇数编号的模块。两个模块必须使用相同的ICTSELx和SYNCSELx位设置。
3. 清零偶数编号模块的ICTRIG位（ICyCON2<7>）。这可以强制该模块与奇数编号的模块运行在同步模式下（不管其触发模式设置如何）。
4. 使用奇数编号的模块的ICIx位（ICxCON1<6:5>）设置所需的中断频率。
5. 使用奇数编号的模块的ICTRIG位（ICxCON2<7>）配置触发或同步工作模式。

注：	对于同步工作模式，最后一个步骤是使能同步源。使能同步源之前，两个输入捕捉模块都保持复位状态。

6. 使用奇数编号的模块的ICMx位（ICxCON1<2:0>）设置所需的捕捉模式。

使能了时基和同步/触发源之后，模块准备就绪，可以捕捉事件。ICBNE位（ICxCON1<3>）置1时，FIFO中至少有一个捕捉值。从FIFO读取输入捕捉值，直到ICBNE清零。

对于32位操作，读取ICxBUF和ICyBUF以得到整个32位定时器值（ICxBUF用于lsw，ICyBUF用于msw）。当奇数编号的模块的ICBNE位（ICxCON1<3>）置1时，FIFO缓冲区中至少有一个捕捉值。继续读缓冲寄存器直到ICBNE清零（由硬件自动执行）。

寄存器14-1： ICxCON1：输入捕捉x控制寄存器1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R-0, HSC	R-0, HSC	R/W-0	R/W-0	R/W-0
—	ICI1	ICI0	ICOV	ICBNE	ICM2[(1)]	ICM1[(1)]	ICM0[(1)]
bit 7							bit 0

图注：	HSC = 硬件置1/清零位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为0

bit 13 **ICSIDL：**输入捕捉x空闲模式停止控制位
1 = 在CPU空闲模式下输入捕捉x停止工作
0 = 在CPU空闲模式下输入捕捉x继续工作

bit 12-10 **ICTSEL<2:0>：**输入捕捉x定时器选择位
111 = 系统时钟（Fosc/2）
110 = 保留
101 = 保留
100 = Timer1
011 = 保留
010 = 保留
001 = Timer2
000 = Timer3

bit 9-7 **未实现：**读为0

bit 6-5 **ICI<1:0>：**输入捕捉x每次中断的捕捉次数选择位
11 = 每4次捕捉事件产生一次中断
10 = 每3次捕捉事件产生一次中断
01 = 每2次捕捉事件产生一次中断
00 = 每次捕捉事件产生一次中断

bit 4 **ICOV：**输入捕捉x溢出状态标志位（只读）
1 = 发生了输入捕捉x溢出
0 = 未发生输入捕捉x溢出

bit 3 **ICBNE：**输入捕捉x缓冲区非空状态位（只读）
1 = 输入捕捉x缓冲区非空，至少可以再读一个捕捉值
0 = 输入捕捉x缓冲区为空

bit 2-0 **ICM<2:0>：**输入捕捉x模式选择位[(1)]
111 = 中断模式：只有当器件处于休眠或空闲模式时输入捕捉x才可作为中断引脚（只检测上升沿，所有其他控制位都不适用）
110 = 未使用（禁止模块）
101 = 预分频捕捉模式：每16个上升沿捕捉一次
100 = 预分频捕捉模式：每4个上升沿捕捉一次
011 = 简单捕捉模式：每个上升沿捕捉一次
010 = 简单捕捉模式：每个下降沿捕捉一次
001 = 边沿检测捕捉模式：每个边沿（上升沿和下降沿）捕捉一次；ICI<1:0>位不控制此模式下中断的产生
000 = 输入捕捉x模块关闭

注 1： ICx输入也必须配置给可用的RPn/RPIn引脚。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。

寄存器14-2： ICxCON2：输入捕捉x控制寄存器2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	IC32
bit 15							bit 8

R/W-0	R/W-0, HS	U-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-1
ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7							bit 0

图注：	HS = 硬件置1位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15-9 **未实现**：读为0

bit 8 **IC32**：级联两个输入捕捉模块使能位（32位操作）
1 = ICx和ICy级联作为32位模块使用（两个模块中的此位都必须置1）
0 = ICx作为16位模块独立工作

bit 7 **ICTRIG**：输入捕捉x同步/触发选择位
1 = 用SYNCSELx位指定的源触发ICx
0 = 使ICx与SYNCSELx位指定的源同步

bit 6 **TRIGSTAT**：定时器触发状态位
1 = 定时器源被触发且正在运行（该位可用硬件置1，也可用软件置1）
0 = 定时器源未被触发，保持清零状态

bit 5 **未实现**：读为0

注 **1**： 这些输入仅用作触发源，不能用作同步源。
2： 请勿通过选择该模式将输入捕捉x模块作为自己的触发源使用。

寄存器14-2： ICxCON2：输入捕捉x控制寄存器2 （续）

bit 4-0 **SYNCSEL<4:0>**：同步/触发源选择位

11111 = 未使用
11110 = 未使用
11101 = 未使用
11100 = CTMU触发[(1)]
11011 = A/D中断[(1)]
11010 = CMP3触发[(1)]
11001 = CMP2触发[(1)]
11000 = CMP1触发[(1)]
10111 = 未使用
10110 = MCCP4 IC/OC中断
10101 = MCCP3 IC/OC中断
10100 = MCCP2 IC/OC中断
10011 = MCCP1 IC/OC中断
10010 = IC3中断[(2)]
10001 = IC2中断[(2)]
10000 = IC1中断[(2)]
01111 = 未使用
01110 = 未使用
01101 = Timer3匹配事件
01100 = Timer2匹配事件
01011 = Timer1匹配事件
01010 = 未使用
01001 = 未使用
01000 = 未使用
00111 = MCCP4同步/触发输出
00110 = MCCP3同步/触发输出
00101 = MCCP2同步/触发输出
00100 = MCCP1同步/触发输出
00011 = OC3同步/触发输出
00010 = OC2同步/触发输出
00001 = OC1同步/触发输出
00000 = 关闭

注 1： 这些输入仅用作触发源，不能用作同步源。
2： 请勿通过选择该模式将输入捕捉x模块作为自己的触发源使用。

注：

15.0 带专用定时器的输出比较

> **注：** 本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的**“带专用定时器的输出比较”**（DS70005159），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

PIC24FJ256GA705系列中的所有器件均具有3个独立的输出比较模块。每个模块都为发生器件内部事件时产生连续脉冲信号提供了各种配置和工作选项，还可产生脉宽调制（Pulse-Width Modulated，PWM）波形以驱动功率应用。

输出比较模块的主要特性包括：

- 可通过级联两个相邻的模块，采用硬件方式配置为所有的32 位工作模式
- 输出比较操作有同步和触发两种模式，最多具有31个用户可选的同步/触发源
- 两个单独的周期寄存器（一个主寄存器OCxR和一个辅助寄存器OCxRS），可更加灵活地产生宽度可变的脉冲
- 可配置为在发生输出事件时产生单脉冲信号或连续脉冲信号，或产生连续的PWM波形
- 每个模块可使用最多 6 个时钟源来驱动独立的内部16位计数器

15.1 一般工作模式

15.1.1 同步和触发模式

输出比较模块工作在自由运行模式下时，内部16位计数器OCxTMR连续递增计数，并在每次溢出时从0xFFFF折回0x0000。其周期与所选的外部时钟源同步。每当内部计数器与其中一个周期寄存器发生匹配时，产生比较或PWM事件。

在同步模式下，一旦使能了所选的时钟源，模块就开始执行比较或PWM操作。只要所选的同步源发生事件，模块的内部计数器就会复位。在触发模式下，模块等待来自其他内部模块的同步事件发生后，才允许该计数器运行。

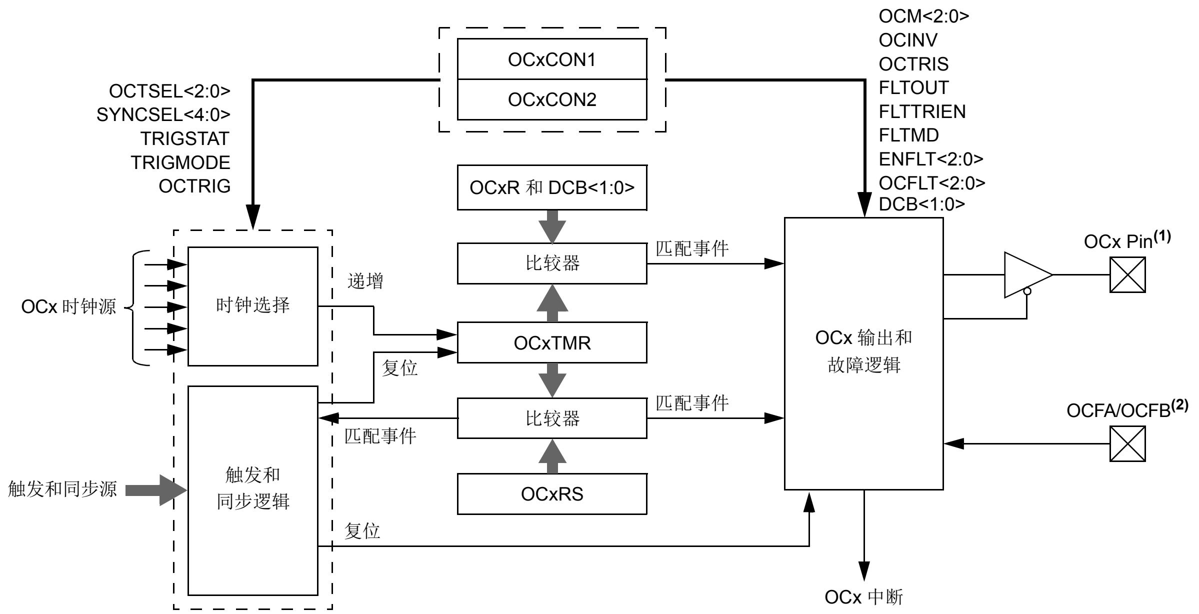
默认选择自由运行模式，或者在 SYNCSEL<4:0> 位（OCxCON2<4:0>）设置为00000时，也可选择自由运行模式。当SYNCSELx位设置为除00000以外的其他值时，将选择同步或触发模式。OCTRIG位（OCxCON2<7>）选择同步或触发模式；置1该位将选择触发工作模式。在两种模式下，SYNCSELx位用于决定同步/触发源。

15.1.2 级联（32位）模式

默认情况下，每个模块使用自己的16位定时器和占空比寄存器独立运行。要提高分辨率，可将相邻的偶数编号的模块和奇数编号的模块配置为一个32位模块。（例如，模块1和2配对，模块3和4配对等）。奇数编号的模块（OCx）提供32位寄存器对中的低16位，而偶数编号的模块（OCy）提供高16位。OCx寄存器的折回操作会导致其相应的OCy寄存器递增。

通过将两个模块的OC32位（OCxCON2<8>）都置1用硬件配置级联操作。有关级联的更多详细信息，请参见《dsPIC33/PIC24系列参考手册》的**“带专用定时器的输出比较”**（DS70005159）。

图15-1： 输出比较x框图（16位模式）



注 1： OCx输出必须在使用前分配给可用的RPn引脚。更多信息，请参见第11.5节“外设引脚选择（PPS）”。

2： OCFA/OCFB故障输入必须在使用前分配给可用的RPn/RPIn引脚。更多信息，请参见第11.5节“外设引脚选择（PPS）”。

15.2 比较操作

在比较模式下（图15-1），输出比较模块可配置为产生单脉冲信号或产生连续脉冲信号。它还可在发生每个定时器事件时重复翻转输出引脚。

要将模块设置为比较操作：

1. 如果您正在使用的OCx模块上具有外设引脚选择引脚，则将OCx输出配置为其中一个可用外设引脚选择引脚。否则，配置专用OCx输出引脚。
2. 计算OCxR和（对于双比较模式）OCxRS占空比寄存器的所需值：
   a) 确定指令时钟周期。考虑定时器源的外部时钟（如果使用）频率和定时器预分频比设置。
   b) 计算从定时器起始值（0000h）到输出脉冲的上升沿所需的时间。
   c) 根据所需的脉冲宽度和到脉冲上升沿的时间计算出现脉冲下降沿的时间。
3. 将上升沿的值写入OCxR，下降沿的值写入OCxRS。
4. 将定时器周期寄存器PRy的值设置为大于或等于OCxRS中的值。
5. 设置OCM<2:0>位以选择合适的比较操作（= 0xx）。
6. 如需触发工作模式，将OCTRIG置1以使能触发模式。置1或清零TRIGMODE可配置触发工作模式，置1或清零TRIGSTAT可选择硬件或软件触发。如需同步模式，清零OCTRIG。
7. 设置SYNCSEL<4:0>位配置触发或同步源。如需定时器工作在自由运行模式下，请将SYNCSELx位设置为00000（无同步/触发源）。
8. 使用OCTSEL<2:0>位选择时基源。如有必要，将选定定时器的TON位置1，使能比较时基进行计数。使能了时基后，将立即启动同步工作模式；发生触发源事件后，启动触发工作模式。

对于32位级联操作，还包含以下必需步骤：

1. 将两个寄存器的OC32位（OCyCON2<8>和OCxCON2<8>）置1。首先使能偶数编号的模块，以确保两个模块一起开始工作。
2. 清零偶数编号模块的OCTRIG位（OCyCON2<7>），使其在同步模式下运行。
3. 为OCy配置所需的输出和故障设置。
4. 通过清零OCTRIS位强制OCx的输出引脚为输出状态。
5. 如需触发工作模式，请使用OCTRIG（OCxCON2<7>）、TRIGMODE（OCxCON1<3>）和SYNCSEL<4:0>（OCxCON2<4:0>）位配置触发选项。
6. 首先为OCy配置所需的比较或PWM工作模式（OCM<2:0>），然后再为OCx配置。

根据选择的输出模式，模块保持OCx引脚处于默认状态，并在OCxR中的值与定时器中的值匹配时强制翻转该引脚的状态。在双比较模式下，当定时器中的值与OCxRS中的值匹配时，OCx被强制返回默认状态。在单比较模式下发生定时器值与OCxR的匹配、以及在双比较模式下每次定时器值与OCxRS发生匹配后，OCxIF中断标志都会置1。

单脉冲事件仅发生一次，但是可通过简单地重写OCxCON1寄存器的值使其重复发生。连续脉冲事件在终止之前可无限次地发生。

15.3 脉宽调制（PWM）模式

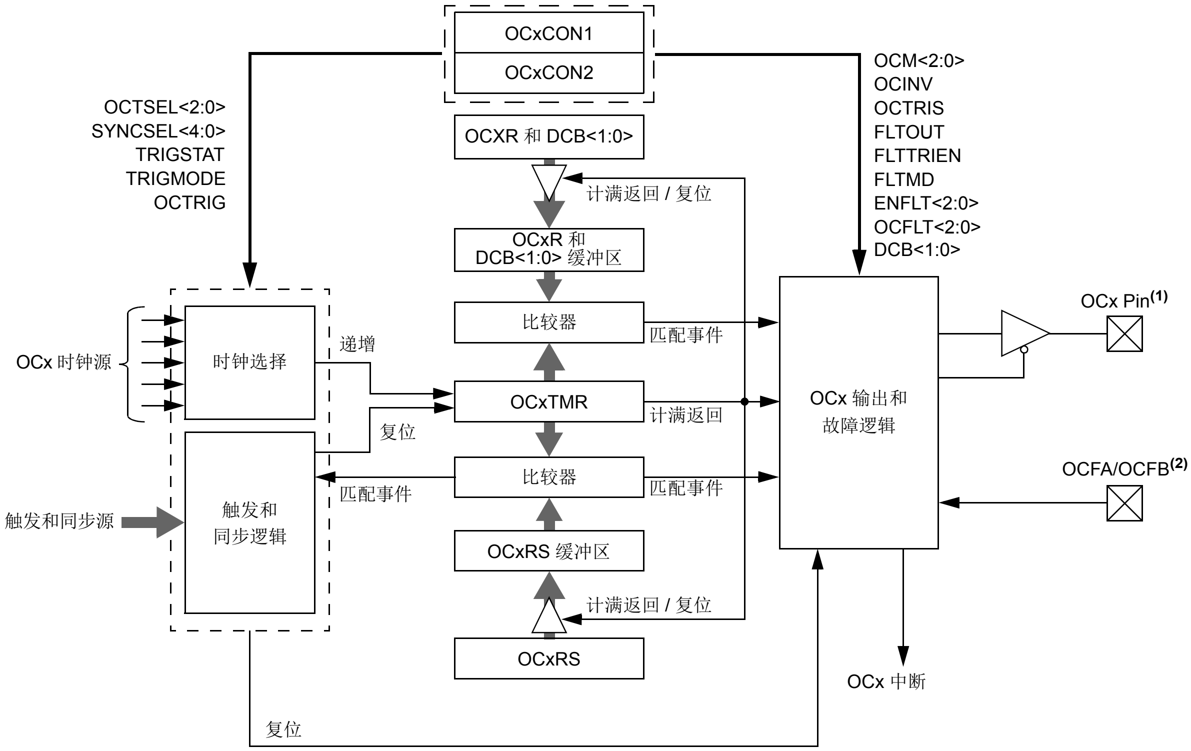
在PWM模式下，输出比较模块可配置为产生边沿对齐或中心对齐的脉冲波形。所有PWM操作都是双缓冲的（缓冲寄存器是模块内部的寄存器，没有映射到SFR空间）。

将输出比较模块配置为PWM工作模式：

1. 如果您正在使用的OCx模块上具有外设引脚选择引脚，则将OC输出配置为其中一个可用外设引脚选择引脚。否则，配置专用OCx输出引脚。
2. 计算所需的占空比并将计算结果装入OCxR寄存器。
3. 计算所需的周期并将计算结果装入OCxRS寄存器。
4. 选择当前的OCx作为同步源，方法是将`0x1F`写入SYNCSEL<4:0>位（OCxCON2<4:0>），将`0`写入OCTRIG位（OCxCON2<7>）。
5. 通过写OCTSEL<2:0>位（OCxCON1<12:10>）选择时钟源。
6. 如果需要的话，允许定时器和输出比较模块中断。使用PWM故障引脚时需要输出比较中断。
7. 通过OCM<2:0>（OCxCON1<2:0>）位选择所需的PWM模式。
8. 通过使用ENFLT<2:0>位可以使能适当的故障输入，如寄存器15-1中所述。
9. 如果定时器被选为时钟源，请设置选定定时器预分频值。选定定时器的预分频器输出将用作OCx定时器的时钟输入，而不是选定定时器的输出。

注：	本外设包含的输入和输出功能可能需要通过外设引脚选择功能进行配置。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。

图15-2： 输出比较x框图（双缓冲，16位PWM模式）



注 1： OCx输出必须在使用前分配给可用的RPn引脚。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。
2： OCFA/OCFB故障输入必须在使用前分配给可用的RPn/RPIn引脚。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。

15.3.1 PWM周期

PWM周期可通过写入PRy（定时器周期寄存器）来指定。可使用公式15-1来计算PWM周期。

公式15-1： **计算PWM周期**[1]

$$\text{PWM 周期} = [(\text{PRy}) + 1 \bullet T_{CY} \bullet (\text{定时器预分频值})$$

其中：

$$\text{PWM 频率} = 1/[\text{PWM 周期}]$$

注 1： 基于TCY = TOSC * 2；禁止打盹模式和PLL。

注：	若PRy的值为N，则会使PWM周期为N + 1个时基计数周期。例如：写入PRy寄存器的值为7将产生由8个时基周期组成的PWM周期。

15.3.2 PWM占空比

PWM占空比是通过写OCxRS和OCxR寄存器指定的。在任何时间都可以写入OCxRS和OCxR寄存器，但是在PRy和TMRy发生匹配（即周期完成）前不会锁存占空比值。这可以为PWM占空比提供双重缓冲，对于PWM的无毛刺操作是极其重要的。

以下是PWM占空比的部分重要边界参数：

- 如果OCxR、OCxRS和PRy装入0000h，那么OCx引脚将保持低电平（占空比为0%）。
- 如果OCxRS大于PRy，则引脚将保持高电平（占空比为100%）。

请参见例15-1了解PWM模式时序的详细信息。表15-1和表15-2给出了当器件分别工作在4 MIPS和10 MIPS时，所对应的PWM频率和分辨率的示例。

公式15-2： 计算最大PWM分辨率[1]

$$\text{最大 PWM 分辨率（位）} = \frac{\log_{10}\left(\dfrac{F_{CY}}{F_{PWM} \cdot (\text{定时器预分频值})}\right)}{\log_{10}(2)} \text{ 位}$$

注 1： 基于FCY = FOSC/2；禁止打盹模式和PLL。

例15-1： 计算PWM周期和占空比[1]

1. 在所需的PWM频率为52.08 kHz、FOSC = 32 MHz并带PLL（32 MHz器件时钟速率）且Timer2预分频比为1:1的条件下，计算定时器周期寄存器的值。

   $T_{CY} = 2 \cdot T_{OSC} = 62.5$ ns

   PWM周期 = 1/PWM频率 = 1/52.08 kHz = 19.2 µS

   PWM周期 = (PR2 + 1) • TCY • (Timer2 预分频值)

   19.2 µS = (PR2 + 1) • 62.5 ns • 1

   PR2 = 306

2. 在PWM频率为52.08 kHz且器件的时钟速率为32 MHz时，计算占空比的最大分辨率：

   PWM分辨率 = $\log_{10}(F_{CY}/F_{PWM})/\log_{10}2)$ 位

   = $(\log_{10}(16\text{ MHz}/52.08\text{ kHz})/\log_{10}2)$ 位

   = 8.3位

注 1： 基于TCY = 2 * TOSC；禁止打盹模式和PLL。

表15-1： 器件工作在4 MIPS（FCY = 4 MHz）时的PWM频率和分辨率示例[1]

PWM频率	7.6 Hz	61 Hz	122 Hz	977 Hz	3.9 kHz	31.3 kHz	125 kHz
定时器预分频比	8	1	1	1	1	1	1
周期寄存器值	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
分辨率（位）	16	16	15	12	10	7	5

注 1： 基于FCY = FOSC/2；禁止打盹模式和PLL。

表15-2： 器件工作在16 MIPS（FCY = 16 MHz）时的PWM频率和分辨率示例[1]

PWM频率	30.5 Hz	244 Hz	488 Hz	3.9 kHz	15.6 kHz	125 kHz	500 kHz
定时器预分频比	8	1	1	1	1	1	1
周期寄存器值	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
分辨率（位）	16	16	15	12	10	7	5

注 1： 基于FCY = FOSC/2；禁止打盹模式和PLL。

寄存器15-1： **OCxCON1：输出比较x控制寄存器1**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2[2]	ENFLT1[2]
bit 15							bit 8

R/W-0	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0	R/W-0	R/W-0	R/W-0
ENFLT0[2]	OCFLT2[2,3]	OCFLT1[2,4]	OCFLT0[2,4]	TRIGMODE	OCM2[1]	OCM1[1]	OCM0[1]
bit 7							bit 0

图注：	HSC = 硬件置1/清零位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15-14 **未实现：**读为0

bit 13 **OCSIDL：**输出比较x空闲模式停止控制位
1 = 输出比较x在CPU空闲模式下停止
0 = 输出比较x在CPU空闲模式下继续工作

bit 12-10 **OCTSEL<2:0>：**输出比较x定时器选择位
111 = 外设时钟（FCY）
110 = 保留
101 = 保留
100 = Timer1时钟（仅支持同步时钟）
011 = 未实现
010 = 未实现
001 = Timer3时钟
000 = Timer2时钟

bit 9 **ENFLT2：**故障输入2使能位[2]
1 = 使能故障2（比较器1/2/3输出）[3]
0 = 禁止故障2

bit 8 **ENFLT1：**故障输入1使能位[2]
1 = 使能故障1（OCFB引脚）[4]
0 = 禁止故障1

bit 7 **ENFLT0：**故障输入0使能位[2]
1 = 使能故障0（OCFA引脚）[4]
0 = 禁止故障0

bit 6 **OCFLT2：**输出比较x PWM故障2（比较器1/2/3）条件状态位[2,3]
1 = 已发生PWM故障2
0 = 未发生PWM故障2

bit 5 **OCFLT1：**输出比较x PWM故障1（OCFB引脚）条件状态位[2,4]
1 = 已发生PWM故障1
0 = 未发生PWM故障1

bit 4 **OCFLT0：**PWM故障0（OCFA引脚）条件状态位[2,4]
1 = 已发生PWM故障0
0 = 未发生PWM故障0

注 1： OCx输出也必须配置给可用的RPn引脚。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。
2： 当OCM<2:0> = 111或110时，故障输入使能位和故障状态位有效。
3： 比较器1输出控制OC1-OC3通道。
4： OCFA/OCFB故障输入也必须配置给可用的RPn/RPIn引脚。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。

寄存器15-1： OCxCON1：输出比较x控制寄存器1（续）

bit 3 **TRIGMODE**：触发状态模式选择位
1 = 当OCxRS = OCxTMR时清零TRIGSTAT（OCxCON2<6>），或由软件清零TRIGSTAT位
0 = TRIGSTAT只能由软件清零

bit 2-0 **OCM<2:0>**：输出比较x模式选择位(1)
111 = OCx处于中心对齐PWM模式(2)
110 = OCx处于边沿对齐PWM模式(2)
101 = 双比较连续脉冲模式：初始化OCx引脚为低电平；当OCxTMR与OCxR和OCxRS交替匹配时连续翻转OCx的状态
100 = 双比较单脉冲模式：初始化OCx引脚为低电平；在一个周期内当OCxTMR分别与OCxR和OCxRS匹配时翻转OCx的状态
011 = 单比较连续脉冲模式：比较事件连续翻转OCx引脚
010 = 单比较单脉冲模式：初始化OCx引脚为高电平；比较事件强制OCx引脚为低电平
001 = 单比较单脉冲模式：初始化OCx引脚为低电平；比较事件强制OCx引脚为高电平
000 = 禁止输出比较通道

注 1： OCx输出也必须配置给可用的RPn引脚。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。
2： 当OCM<2:0> = 111或110时，故障输入使能位和故障状态位有效。
3： 比较器1输出控制OC1-OC3通道。
4： OCFA/OCFB故障输入也必须配置给可用的RPn/RPIn引脚。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。

寄存器15-2： OCxCON2：输出比较x控制寄存器2

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1(3)	DCB0(3)	OC32
bit 15							bit 8

R/W-0	R/W-0, HS	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0
OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7							bit 0

图注：	HS = 硬件置1位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15 **FLTMD**：故障模式选择位
1 = 故障模式保持，直到消除故障源并且相应的OCFLT0位由软件清零
0 = 故障模式保持，直到消除故障源并且启动了新的PWM周

bit 14 **FLTOUT**：故障输出位
1 = 发生故障时PWM输出被驱动为高电平
0 = 发生故障时PWM输出被驱动为低电平

bit 13 **FLTTRIEN**：故障输出状态选择位
1 = 发生故障条件时引脚被强制为输出状态
0 = 引脚I/O状态不受故障影响

bit 12 **OCINV**：OCMP反相位
1 = OCx输出反相
0 = OCx输出不反相

bit 11 **未实现**：读为0

bit 10-9 **DCB<1:0>**：PWM占空比最低有效位(3)
11 = OCx下降沿延迟3/4指令周期
10 = OCx下降沿延迟1/2指令周期
01 = OCx下降沿延迟1/4指令周期
00 = OCx 下降沿在指令周期一开始就出现

bit 8 **OC32**：级联两个输出比较模块使能位（32位操作）
1 = 使能级联模块操作
0 = 禁止级联模块操作

bit 7 **OCTRIG**：OCx触发/同步选择位
1 = 用SYNCSELx位指定的源触发OCx
0 = OCx与SYNCSELx位指定的源同步

bit 6 **TRIGSTAT**：定时器触发状态位
1 = 定时器源已触发并正在运行
0 = 定时器源未被触发，保持清零状态

bit 5 **OCTRIS**：OCx输出引脚方向选择位
1 = OCx引脚为三态
0 = OCx引脚上连接了输出比较外设x

注 1： 不要通过以下方式将输出比较x模块作为自己的触发源：选择这种模式或其他同等SYNCSELx设置。
2： 这些输入仅用作触发源，不能用作同步源。
3： DCB<1:0>位只有在PWM模式（OCM<2:0>（OCxCON1<2:0>）= 111、110）下才是双重缓冲的。

寄存器15-2： OCxCON2：输出比较x控制寄存器2（续）

bit 4-0 **SYNCSEL<4:0>**：触发/同步源选择位

`11111` = OCx同步输出[(1)]
`11110` = OCTRIG1引脚
`11101` = OCTRIG2引脚
`11100` = CTMU触发[(2)]
`11011` = A/D中断[(2)]
`11010` = CMP3触发[(2)]
`11001` = CMP2触发[(2)]
`11000` = CMP1触发[(2)]
`10111` = 未使用
`10110` = MCCP4 IC/OC中断
`10101` = MCCP3 IC/OC中断
`10100` = MCCP2 IC/OC中断
`10011` = MCCP1 IC/OC中断
`10010` = IC3中断[(2)]
`10001` = IC2中断[(2)]
`10000` = IC1中断[(2)]
`01111` = 未使用
`01110` = 未使用
`01101` = Timer3匹配事件
`01100` = Timer2匹配事件（默认）
`01011` = Timer1匹配事件
`01010` = 未使用
`01001` = 未使用
`01000` = 未使用
`00111` = MCCP4同步/触发输出
`00110` = MCCP3同步/触发输出
`00101` = MCCP2同步/触发输出
`00100` = MCCP1同步/触发输出
`00011` = 未使用
`00010` = OC3同步/触发输出[(1)]
`00001` = OC1同步/触发输出[(1)]
`00000` = 关闭，自由运行模式，无同步且在FFFFh处计满返回

注 1： 不要通过以下方式将输出比较x模块作为自己的触发源：选择这种模式或其他同等SYNCSELx设置。
2： 这些输入仅用作触发源，不能用作同步源。
3： DCB<1:0>位只有在PWM模式（OCM<2:0>（OCxCON1<2:0>）= `111`、`110`）下才是双重缓冲的。

注：

16.0 捕捉/比较/PWM/定时器模块（MCCP）

> **注：** 本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的**"捕捉/比较/PWM/定时器（MCCP和SCCP）"**（DS33035），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

PIC24FJ256GA705系列器件包含几个捕捉/比较/PWM/定时器基本模块，这些模块提供了先前PIC24F器件的三种不同外设功能。该模块可在以下三种主要模式下工作：

- 通用定时器
- 输入捕捉
- 输出比较/PWM

该系列器件的MCCP模块具有4种实例。MCCP1提供最多6路输出和一系列丰富的电源控制功能，而MCCP2-MCCP4支持2路输出。

在任何时候，MCCPx模块只能在三种主要模式之一下工作。除非模块重新配置为新模式，否则其他模式不可用。

图16-1给出了该模块的概念框图。所有三种模式共用一个时基发生器和一个公共定时器寄存器对（CCPxTMRH/L）；其他共用硬件组件在特定模式需要时增加。

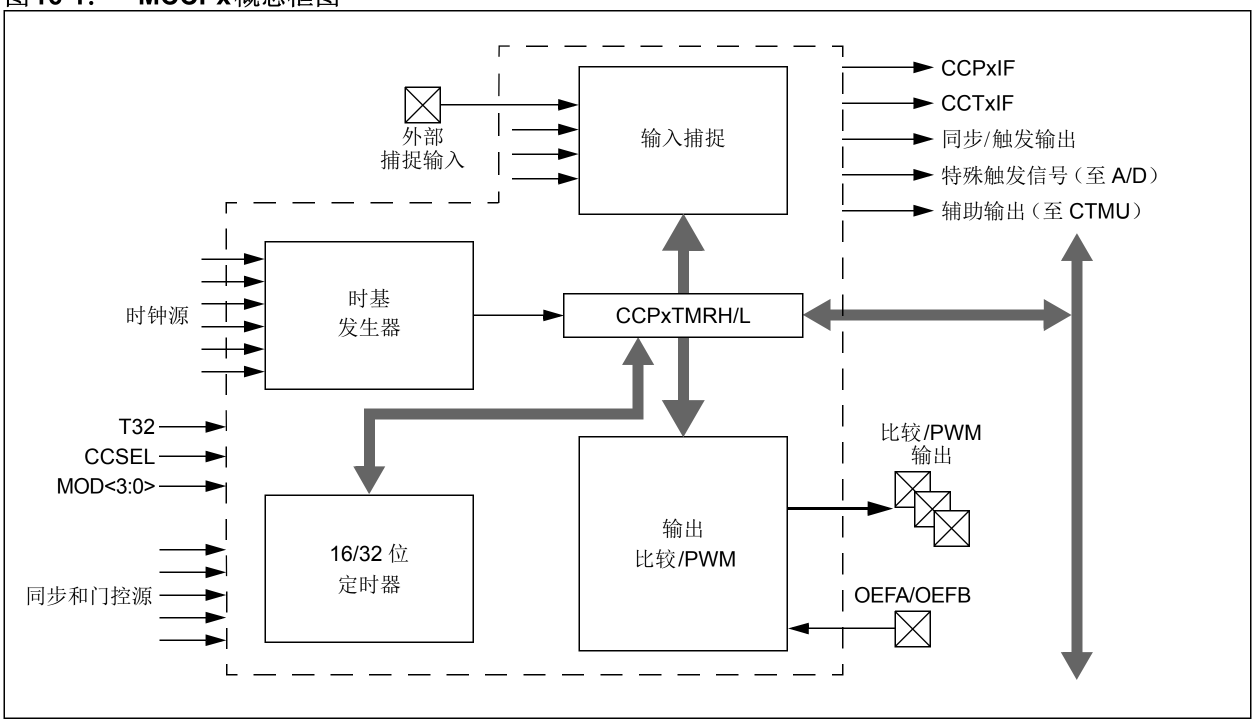
每个模块总共具有8个控制和状态寄存器：

- CCPxCON1L（寄存器16-1）
- CCPxCON1H（寄存器16-2）
- CCPxCON2L（寄存器16-3）
- CCPxCON2H（寄存器16-4）
- CCPxCON3L（寄存器16-5）
- CCPxCON3H（寄存器16-6）
- CCPxSTATL（寄存器16-7）
- CCPxSTATH（寄存器16-8）

每个模块还包括8个缓冲区/计数器寄存器，用作定时器值寄存器或数据存放缓冲区：

- CCPxTMRH/CCPxTMRL（定时器高/低位字计数器）
- CCPxPRH/CCPxPRL（定时器周期高/低位字）
- CCPxRAH/CCPxRAL（主输出比较数据缓冲区）
- CCPxRBH/CCPxRBL（辅助输出比较数据缓冲区）
- CCPxBUFH/CCPxBUFL（输入捕捉高/低位字缓冲区）

图16-1： MCCPx概念框图

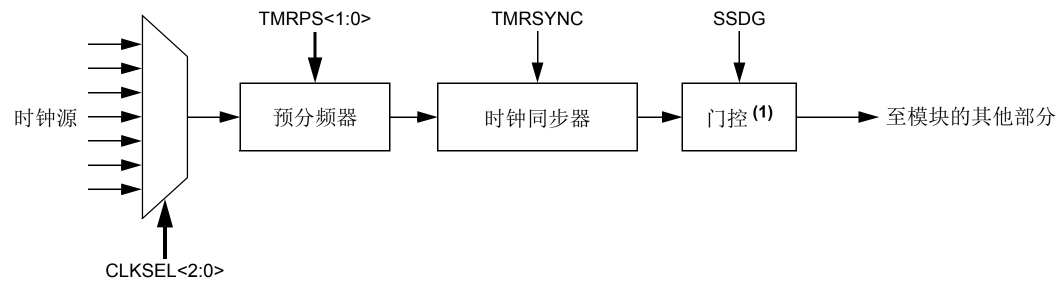


16.1 时基发生器

定时器时钟发生器（Timer Clock Generator，TCG）可使用单片机上已提供的时钟信号之一，生成用作模块内部时基的时钟。它在模块的三种主要模式中用作模块的时间参考。图16-2给出了内部时基的图示。

有8个输入可供时钟发生器使用，这些输入使用CLKSEL<2:0>位（CCPxCON1L<10:8>）进行选择。可用的时钟源包括FRC与LPRC、辅助振荡器和TCLKI外部时钟输入。系统时钟是默认的时钟源（CLKSEL<2:0> = 000）。在PIC24FJ256GA705系列器件上，MCCPx模块的时钟源必须与系统时钟同步。因此，选择时钟源时，可能存在时钟输入时序限制或模块工作限制。

图16-2： 定时器时钟发生器



注 1： 门控仅在定时器模式下可用。

16.2 通用定时器

当CCSEL = 0且MOD<3:0> = 0000时，选择定时器模式。定时器可以作为一个32位定时器或双16位定时器工作，具体取决于T32位的设置（表16-1）。

表16-1： 定时器工作模式

T32（CCPxCON1L<5>）	工作模式
0	双定时器模式（16位）
1	定时器模式（32位）

双16位定时器模式提供简单的定时器功能，具有两个独立的16位定时器/计数器。主定时器使用CCPxTMRL和CCPxPRL寄存器。只有主定时器可以与器件上的其他模块进行交互。它可以产生由其他MCCPx模块使用的MCCPx同步输出信号。此外，它还可以使用由其他模块产生的SYNC<4:0>位信号。

辅助定时器使用CCPxTMRH和CCPxPRH寄存器。它仅可用作调度CPU事件的定期中断源。它不会产生类似于主时基的输出同步/触发信号。在双定时器模式下，定时器周期高位字寄存器CCPxPRH可产生MCCPx比较事件（CCPxIF），供器件上的许多其他模块使用。

32位定时器模式将CCPxTMRL和CCPxTMRH寄存器一起使用，作为单个32位定时器。当CCPxTMRL发生溢出时，CCPxTMRH将递增1。在需要跟踪长时间周期时，该模式可以提供简单的定时器功能。请注意，应先将T32位（CCPxCON1L<5>）置1，然后再通过写入CCPxTMRL或CCPxPRH寄存器来初始化32位定时器。

16.2.1 同步和触发操作

在16位和32位模式下，定时器还可以进行同步操作或触发操作。两者均使用SYNC<4:0>位（CCPxCON1H<4:0>）来确定输入信号源。不同之处在于信号如何影响定时器。

在同步操作中，当通过SYNC<4:0>选择的输入置为有效时，定时器会发生复位或清零。定时器将立即重新从零开始计数，除非由于某种其他原因而受阻。每当TRIGEN位（CCPxCON1H<7>）清零时，就将使用同步操作。SYNC<4:0>位可以具有除11111之外的任何值。

在触发操作中，定时器会一直保持复位状态，直到通过SYNC<4:0>选择的输入置为有效为止；发生该事件时，定时器会开始计数。每当TRIGEN位置1时，就将使用触发操作。在触发模式下，只要CCPTRIG位（CCPxSTATL<7>）置1，定时器就会在触发事件后继续运行。要清零CCPTRIG位，必须通过将TRCLR位（CCPxSTATL<5>）置1来清除触发事件，复位定时器，并将其保持为零，直到发生另一个触发事件。在PIC24FJ256GA705系列器件上，只有系统时钟为时基源（CLKSEL<2:0> = 000）时，才能使用触发工作模式。

图16-3： 双16位定时器模式

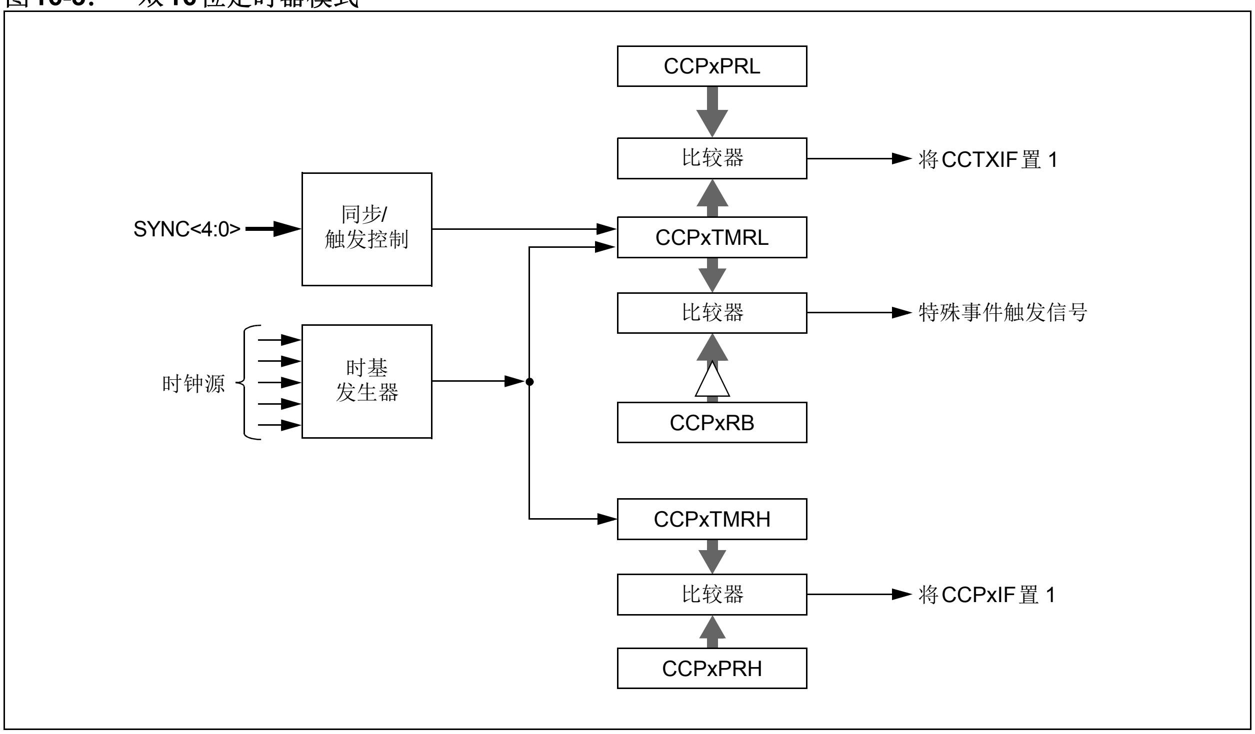
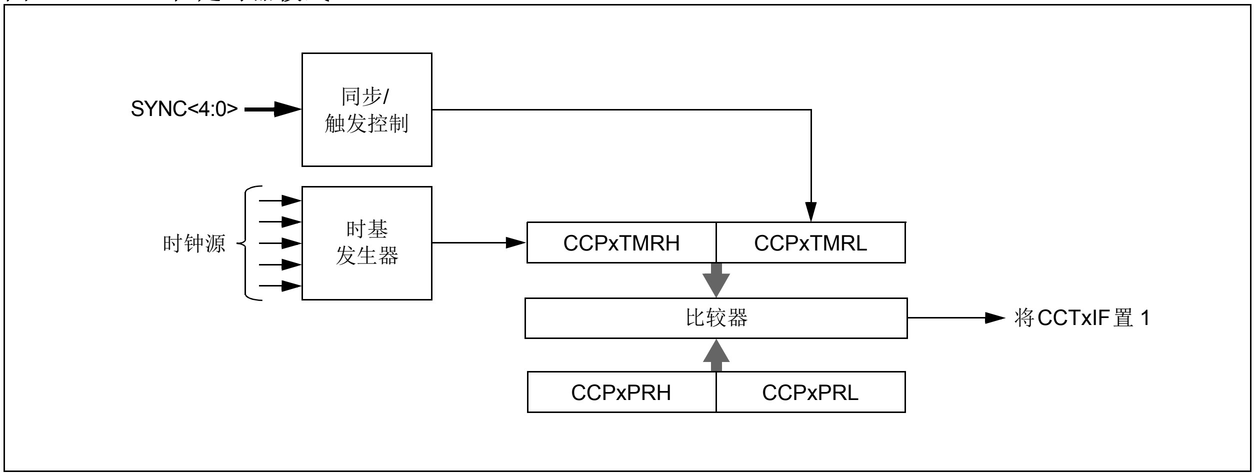


图16-4： 32位定时器模式



16.3 输出比较模式

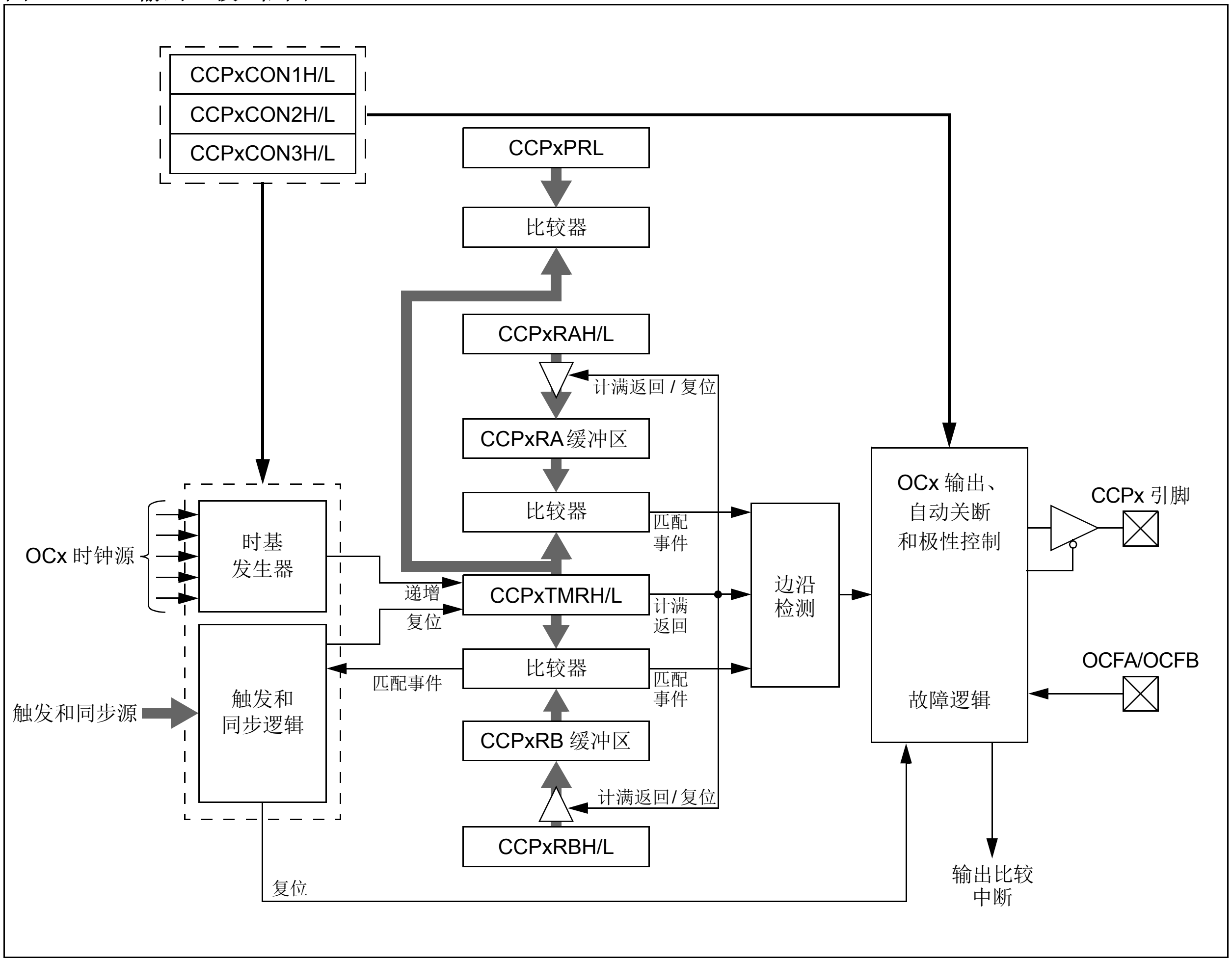
输出比较模式会将定时器寄存器值与一个或两个比较寄存器（取决于其工作模式）的值进行比较。在发生比较匹配事件时，输出比较x模块可以产生单个输出跳变或一连串输出脉冲。类似于大多数PIC MCU外设，输出比较x模块也可以在发生比较匹配事件时产生中断。

表16-2显示了输出比较模式下可用的不同模式。

表16-2： 输出比较/PWM模式

MOD<3:0>（CCPxCON1L<3:0>）	**T32（CCPxCON1L<5>）**	工作模式	
0001	0	在比较匹配时输出高电平（16位）	单边沿模式
0001	1	在比较匹配时输出高电平（32位）	
0010	0	在比较匹配时输出低电平（16位）	
0010	1	在比较匹配时输出低电平（32位）	
0011	0	在比较匹配时输出翻转（16位）	
0011	1	在比较匹配时输出翻转（32位）	
0100	0	双边沿比较（16位）	双边沿模式
0101	0	双边沿比较（16位缓冲）	PWM模式
0110	0	中心对齐脉冲（16位缓冲）	中心PWM模式
0111	0	可变频率脉冲（16位）	
1111	0	外部输入源模式（16位）	

图16-5： 输出比较x框图



16.4 输入捕捉模式

输入捕捉模式用于在输入引脚上或其他内部触发源上有事件发生时，捕捉来自独立时基的定时器值。输入捕捉功能在需要进行频率（时间周期）和脉冲测量的应用中很有用。图16-6给出了输入捕捉模式的简化框图。

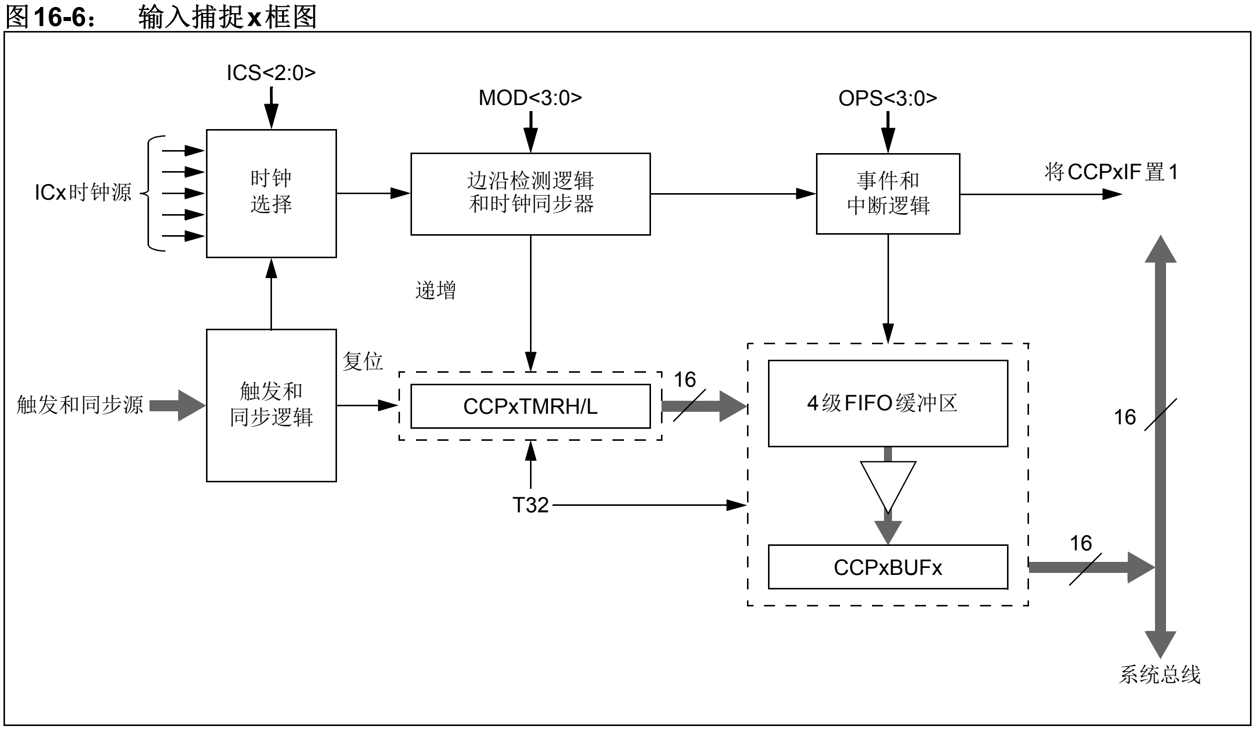
输入捕捉模式使用专用的16/32位同步递增定时器来实现捕捉功能。当发生捕捉事件时，定时器值将写入FIFO。内部值可以使用CCPxTMRH/L寄存器读取（具有同步延时）。

要使用输入捕捉模式，必须将CCSEL位（CCPxCON1L<4>）置1。T32和MOD<3:0>位用于选择适当的捕捉模式，如表16-3所示。

表16-3： 输入捕捉模式

MOD<3:0>（CCPxCON1L<3:0>）	T32（CCPxCON1L<5>）	工作模式
0000	0	边沿检测（16位捕捉）
0000	1	边沿检测（32位捕捉）
0001	0	每个上升沿（16位捕捉）
0001	1	每个上升沿（32位捕捉）
0010	0	每个下降沿（16位捕捉）
0010	1	每个下降沿（32位捕捉）
0011	0	每个上升沿/下降沿（16位捕捉）
0011	1	每个上升沿/下降沿（32位捕捉）
0100	0	每4个上升沿（16位捕捉）
0100	1	每4个上升沿（32位捕捉）
0101	0	每16个上升沿（16位捕捉）
0101	1	每16个上升沿（32位捕捉）

图16-6： 输入捕捉x框图



16.5 辅助输出

MCCPx模块具有一个辅助输出，使其他外设可以访问内部模块信号。辅助输出用于连接到其他MCCPx模块，或者其他数字外设，从而提供以下类型的功能：

- 时基同步
- 外设触发和时钟输入
- 信号门控

输出信号的类型使用AUXOUT<1:0>控制位（CCPxCON2H<4:3>）选择。此外，输出信号的类型还取决于模块的工作模式。

在PIC24FJ256GA705系列器件上，只有CTMU放电触发可以访问辅助输出信号。

表16-4： 辅助输出

AUXOUT<1:0>	CCSEL	MOD<3:0>	备注	信号说明
00	x	xxxx	禁止辅助输出	无输出
01	0	0000	时基模式	时基周期复位或计满返回
10				特殊事件触发输出
11				无输出
01	0	0001 至 1111	输出比较模式	时基周期复位或计满返回
10				输出比较事件信号
11				输出比较信号
01	1	xxxx	输入捕捉模式	时基周期复位或计满返回
10				反映ICDIS位的值
11				输入捕捉事件信号

寄存器16-1：　CCPxCON1L：CCPx控制1低位字寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CCPON	—	CCPSIDL	CCPSLP	TMRSYNC	CLKSEL2	CLKSEL1	CLKSEL0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TMRPS1	TMRPS0	T32	CCSEL	MOD3	MOD2	MOD1	MOD0
bit 7							bit 0

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

bit 15　**CCPON**：CCPx模块使能位
1 = 使用由MOD<3:0>控制位指定的工作模式使能模块
0 = 禁止模块

bit 14　**未实现**：读为0

bit 13　**CCPSIDL**：CCPx空闲模式停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 在空闲模式下模块继续工作

bit 12　**CCPSLP**：CCPx休眠模式使能位
1 = 在休眠模式下模块继续工作
0 = 在休眠模式下模块不工作

bit 11　**TMRSYNC**：时基时钟同步位
1 = 模块时基时钟与内部系统时钟同步；时序限制适用
0 = 模块时基时钟与内部系统时钟不同步

bit 10-8　**CLKSEL<2:0>**：CCPx时基时钟选择位
111 = TCKIA引脚
110 = TCKIB引脚
101 = PLL时钟
100 = 2x外设时钟
010 = SOSC时钟
001 = 参考时钟输出
000 = 系统时钟

对于MCCP1：
011 = CLC1输出
对于MCCP2：
011 = CLC2输出

bit 7-6　**TMRPS<1:0>**：时基预分频比选择位
11 = 1:64预分频比
10 = 1:16预分频比
01 = 1:4预分频比
00 = 1:1预分频比

bit 5　**T32**：32位时基选择位
1 = 对于定时器使用32位时基，单边沿输出比较或输入捕捉功能
0 = 对于定时器使用16位时基，单边沿输出比较或输入捕捉功能

bit 4　**CCSEL**：捕捉/比较模式选择位
1 = 输入捕捉外设
0 = 输出比较/PWM/定时器外设（确切功能由MOD<3:0>位选择）

寄存器16-1： CCPxCON1L：CCPx控制1低位字寄存器（续）

bit 3-0 **MOD<3:0>**：CCPx模式选择位

CCSEL = 1（输入捕捉模式）时：
1xxx = 保留
011x = 保留
0101 = 每16个上升沿捕捉一次
0100 = 每4个上升沿捕捉一次
0011 = 每个上升沿和下降沿各捕捉一次
0010 = 每个下降沿捕捉一次
0001 = 每个上升沿捕捉一次
0000 = 每个上升沿和下降沿各捕捉一次（边沿检测模式）

CCSEL = 0（输出比较/定时器模式）时：
1111 = 外部输入模式：禁止脉冲发生器，时钟源通过ICS<2:0>选择
1110 = 保留
110x = 保留
10xx = 保留
0111 = 可变频率脉冲模式
0110 = 中心对齐脉冲比较模式，进行缓冲
0101 = 双边沿比较模式，进行缓冲
0100 = 双边沿比较模式
0011 = 16位/32位单边沿模式，在发生比较匹配时翻转输出
0010 = 16位/32位单边沿模式，在发生比较匹配时将输出驱动为低电平
0001 = 16位/32位单边沿模式，在发生比较匹配时将输出驱动为高电平
0000 = 16位/32位定时器模式，禁止输出功能

寄存器16-2： CCPxCON1H：CCPx控制1高位字寄存器

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
OPSSRC[1]	RTRGEN[2]	—	—	OPS3[3]	OPS2[3]	OPS1[3]	OPS0[3]
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TRIGEN	ONESHOT	ALTSYNC	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
bit 7							bit 0

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

bit 15 **OPSSRC：**输出后分频器源选择位[1]
1 = 输出后分频器对模块触发输出事件进行分频
0 = 输出后分频器对时基中断事件进行分频

bit 14 **RTRGEN：**重新触发使能位[2]
1 = 当TRIGEN位 = 1时可以重新触发时基
0 = 当TRIGEN位 = 1时不能重新触发时基

bit 13-12 **未实现：**读为0

bit 11-8 **OPS3<3:0>：**CCPx中断输出后分频比选择位[3]
1111 = 在发生第16次时基周期匹配时产生中断
1110 = 在发生第15次时基周期匹配时产生中断
• • •
0100 = 在发生第5次时基周期匹配时产生中断
0011 = 在发生第4次时基周期匹配或第4次输入捕捉事件时产生中断
0010 = 在发生第3次时基周期匹配或第3次输入捕捉事件时产生中断
0001 = 在发生第2次时基周期匹配或第2次输入捕捉事件时产生中断
0000 = 在发生时基周期匹配或输入捕捉事件时产生中断

bit 7 **TRIGEN：**CCPx触发使能位
1 = 使能时基的触发操作
0 = 禁止时基的触发操作

bit 6 **ONESHOT：**单脉冲模式使能位
1 = 使能单脉冲触发模式；触发模式持续时间由OSCNT<2:0>设置
0 = 禁止单脉冲触发模式

bit 5 **ALTSYNC：**CCPx时钟选择位
1 = 使用备用信号作为模块同步输出信号
0 = 模块同步输出信号为时基复位/计满返回事件

bit 4-0 **SYNC<4:0>：**CCPx同步源选择位
关于同步源输入的定义，请参见表16-5。

注 1： 该控制位在输入捕捉模式下没有任何作用。
2： 当TRIGEN = 0时，该控制位没有任何作用。
3： 对于输入捕捉模式，输出后分频比设置为1:5到1:16（0100-1111）会导致FIFO缓冲区溢出。

表16-5： 同步源

SYNC<4:0>	同步源
11111	无；定时器在发生CCPxPR匹配或值达到FFFFh时计满返回
11110	保留
11101	保留
11100	CTMU触发
11011	A/D开始转换
11010	CMP3触发
11001	CMP2触发
11000	CMP1触发
10111	保留
10110	保留
10101	保留
10100	保留
10011	保留
10010	保留
10001	CLC2输出
10000	CLC1输出
01111	保留
01110	保留
01101	保留
01100	保留
01011	INT2引脚
01010	INT1引脚
01001	INT0引脚
01000	保留
00111	保留
00110	保留
00101	MCCP4同步输出
00100	MCCP3同步输出
00011	MCCP2同步输出
00010	MCCP1同步输出
00001	MCCPx同步输出(1)
00000	MCCPx定时器同步输出(1)

注 1： 连接到CCP1时为CCP1，连接到CCP2时为CCP2，依此类推。

寄存器16-3： CCPxCON2L：CCPx控制2低位字寄存器

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0
PWMRSEN	ASDGM	—	SSDG	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ASDG7	ASDG6	ASDG5	ASDG4	ASDG3	ASDG2	ASDG1	ASDG0
bit 7							bit 0

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

bit 15 **PWMRSEN**：CCPx PWM重启使能位
1 = 在关断输入结束之后，ASEVT位在下一个PWM周期开始时自动清零
0 = 必须用软件清零ASEVT位，才会恢复输出引脚上的PWM活动

bit 14 **ASDGM**：CCPx自动关断门控模式使能位
1 = 等到下一次时基复位或计满返回时才发生关断
0 = 立即发生关断事件

bit 13 **未实现**：读为0

bit 12 **SSDG**：CCPx软件关断/门控控制位
1 = 手动强制自动关断、定时器时钟门控或输入捕捉信号门控事件（ASDGM位的设置仍然适用）
0 = 模块正常工作

bit 11-8 **未实现**：读为0

bit 7-0 **ASDG<7:0>**：CCPx自动关断/门控源使能位
1 = 使能ASDGx源n（关于自动关断/门控源，请参见表16-6）
0 = 禁止ASDGx源n

表16-6： 自动关断源

ASDG<7:0>	自动关断源			
	MCCP1	MCCP2	MCCP3	MCCP4
1xxx xxxx	OCFB			
x1xx xxxx	OCFA			
xx1x xxxx	CLC1	CLC2	未使用	
xxx1 xxxx	未使用			
xxxx 1xxx	未使用			
xxxx x1xx	CMP3输出			
xxxx xx1x	CMP2输出			
xxxx xxx1	CMP1输出			

寄存器16-4： CCPxCON2H：CCPx控制2高位字寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
OENSYNC	—	OCFEN	OCEEN	OCDEN	OCCEN	OCBEN	OCAEN
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ICGSM1	ICGSM0	—	AUXOUT1	AUXOUT0	ICS2	ICS1	ICS0
bit 7							bit 0

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

bit 15 **OENSYNC**：输出使能同步位
1 = 由输出使能位产生的更新在下一次时基复位或计满返回时发生
0 = 由输出使能位产生的更新立即发生

bit 14 **未实现**：读为0

bit 13-8 **OCxEN**：输出使能/转向控制位
1 = OCMx引脚由CCPx模块控制，并产生输出比较或PWM信号
0 = OCMx引脚不由CCPx模块控制；引脚可供端口逻辑或引脚上复用的另一个外设使用

bit 7-6 **ICGSM<1:0>**：输入捕捉门控源模式控制位
11 = 保留
10 = 单脉冲模式：来自门控源的下降沿禁止以后的捕捉事件（ICDIS = 1）
01 = 单脉冲模式：来自门控源的上升沿使能以后的捕捉事件（ICDIS = 0）
00 = 电平敏感模式：来自门控源的高电平将使能以后的捕捉事件；低电平将禁止以后的捕捉事件

bit 5 **未实现**：读为0

bit 4-3 **AUXOUT<1:0>**：辅助输出信号事件选择位
11 = 输入捕捉或输出比较事件；在定时器模式下无信号
10 = 信号输出由模块工作模式定义（见表16-4）
01 = 时基计满返回事件（所有模式）
00 = 禁止

bit 2-0 **ICS<2:0>**：输入捕捉时钟源选择位
111 = 保留
110 = 保留
101 = CLC2输出
100 = CLC1输出
011 = 比较器3输出
010 = 比较器2输出
001 = 比较器1输出
000 = 输入捕捉x（ICMx）I/O引脚

寄存器16-5： **CCPxCON3L：CCPx控制3低位字寄存器[1]**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DT<5:0>					
bit 7							bit 0

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

bit 15-6 **未实现：**读为0

bit 5-0 **DT<5:0>：**CCPx死区选择位[1]

111111 = 在互补输出信号之间插入63个死区延时周期
111110 = 在互补输出信号之间插入62个死区延时周期
...
000010 = 在互补输出信号之间插入2个死区延时周期
000001 = 在互补输出信号之间插入1个死区延时周期
000000 = 禁止死区逻辑

注 1： 该寄存器仅在MCCP1模块中实现。

寄存器16-6： CCPxCON3H：CCPx控制3高位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
OETRIG	OSCNT2	OSCNT1	OSCNT0	—	OUTM2[1]	OUTM1[1]	OUTM0[1]
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	POLACE	POLBDF[1]	PSSACE1	PSSACE0	PSSBDF1[1]	PSSBDF0[1]
bit 7							bit 0

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

bit 15 **OETRIG**：CCPx死区选择位
1 = 对于触发模式（TRIGEN = 1）：只有在触发之后，模块才驱动使能的输出引脚
0 = 正常输出引脚操作

bit 14-12 **OSCNT<2:0>**：单脉冲事件计数位
111 = 将单脉冲事件延长7个时基周期（总共8个时基周期）
110 = 将单脉冲事件延长6个时基周期（总共7个时基周期）
101 = 将单脉冲事件延长5个时基周期（总共6个时基周期）
100 = 将单脉冲事件延长4个时基周期（总共5个时基周期）
011 = 将单脉冲事件延长3个时基周期（总共4个时基周期）
010 = 将单脉冲事件延长2个时基周期（总共3个时基周期）
001 = 将单脉冲事件延长1个时基周期（总共2个时基周期）
000 = 不延长单脉冲触发事件

bit 11 **未实现**：读为0

bit 10-8 **OUTM<2:0>**：PWMx输出模式控制位[1]
111 = 保留
110 = 输出扫描模式
101 = 有刷直流输出模式，正向
100 = 有刷直流输出模式，反向
011 = 保留
010 = 半桥输出模式
001 = 推挽输出模式
000 = 可转向单输出模式

bit 7-6 **未实现**：读为0

bit 5 **POLACE**：CCPx输出引脚、OCMxA、OCMxC和OCMxE极性控制位
1 = 输出引脚极性为低电平有效
0 = 输出引脚极性为高电平有效

bit 4 **POLBDF**：CCPx输出引脚、OCMxB、OCMxD和OCMxF极性控制位[1]
1 = 输出引脚极性为低电平有效
0 = 输出引脚极性为高电平有效

bit 3-2 **PSSACE<1:0>**：PWMx输出引脚、OCMxA、OCMxC和OCMxE关断状态控制位
11 = 引脚在发生关断事件时驱动为有效
10 = 引脚在发生关断事件时驱动为无效
0x = 引脚在发生关断事件时为三态

bit 1-0 **PSSBDF<1:0>**：PWMx输出引脚、OCMxB、OCMxD和OCMxF关断状态控制位[1]
11 = 引脚在发生关断事件时驱动为有效
10 = 引脚在发生关断事件时驱动为无效
0x = 引脚在发生关断事件时处于高阻态

注 1： 这些位仅在MCCP1模块中实现。

寄存器16-7： CCPxSTATL：CCPx状态寄存器低位字

U-0	U-0	U-0	U-0	U-0	W-0	U-0	U-0
—	—	—	—	—	ICGARM	—	—
bit 15							bit 8

R-0	W1-0	W1-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
CCPTRIG	TRSET	TRCLR	ASEVT	SCEVT	ICDIS	ICOV	ICBNE
bit 7							bit 0

图注：	C = 可清零位	W = 可写位	
R = 可读位	W1 = 仅可写1位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15-11 **未实现：**读为0

bit 10 **ICGARM：**输入捕捉门控激活位
向该位写入1会激活输入捕捉x模块，在ICGSM<1:0> = 01或10时产生单脉冲门控事件；读为0。

bit 9-8 **未实现：**读为0

bit 7 **CCPTRIG：**CCPx触发状态位
1 = 定时器已触发并正在运行
0 = 定时器未触发并保持复位状态

bit 6 **TRSET：**CCPx触发置1请求位
当TRIGEN = 1时，向该位写入1可触发定时器（该位总是读为0）。

bit 5 **TRCLR：**CCPx触发清零请求位
当TRIGEN = 1时，向该位写入1可取消定时器触发（该位总是读为0）。

bit 4 **ASEVT：**CCPx自动关断事件状态/控制位
1 = 关断事件正在进行；CCPx输出处于关断状态
0 = CCPx输出正常工作

bit 3 **SCEVT：**单边沿比较事件状态位
1 = 已发生单边沿比较事件
0 = 未发生单边沿比较事件

bit 2 **ICDIS：**输入捕捉x禁止位
1 = 输入捕捉x引脚（ICMx）上的事件不产生捕捉事件
0 = 输入捕捉x引脚上的事件产生捕捉事件

bit 1 **ICOV：**输入捕捉x缓冲区溢出状态位
1 = 输入捕捉x FIFO缓冲区已溢出
0 = 输入捕捉x FIFO缓冲区未溢出

bit 0 **ICBNE：**输入捕捉x缓冲区状态位
1 = 输入捕捉x缓冲区中有数据
0 = 输入捕捉x缓冲区为空

寄存器16-8： **CCPxSTATH：CCPx状态寄存器高位字**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R-0	R-0	R-0	R-0	R-0
—	—	—	PRLWIP	TMRHWIP	TMRLWIP	RBWIP	RAWIP
bit 7							bit 0

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

bit 15-5 **未实现：**读为0

bit 4 **PRLWIP：**CCPxPRL写进行状态位
1 = 正在使用缓冲内容更新CCPxPRL寄存器
0 = 未在更新CCPxPRL寄存器

bit 3 **TMRHWIP：**CCPxTMRH写进行状态位
1 = 正在使用缓冲内容更新CCPxTMRH寄存器
0 = 未在更新CCPxTMRH寄存器

bit 2 **TMRLWIP：**CCPxTMRL写进行状态位
1 = 正在使用缓冲内容更新CCPxTMRL寄存器
0 = 未在更新CCPxTMRL寄存器

bit 1 **RBWIP：**CCPxRB写进行状态位
1 = 正在使用缓冲内容更新CCPxRB寄存器
0 = 未在更新CCPxRB寄存器

bit 0 **RAWIP：**CCPxRA写进行状态位
1 = 正在使用缓冲内容更新CCPxRA寄存器
0 = 未在更新CCPxRA寄存器

注：

17.0 串行外设接口（SPI）

> **注：** 本数据手册总结了PIC24FJ256GA705系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24系列参考手册》的 **“串行外设接口（SPI）”**（DS70005185），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

串行外设接口（Serial Peripheral Interface，SPI）模块是用于同其他外设或单片机器件进行通信的同步串行接口。这些外设可以是串行EEPROM、移位寄存器、显示驱动器和A/D转换器等。SPI模块与Motorola® SPI和SIOP接口兼容。PIC24FJ256GA705系列中的所有器件都包含三个SPI模块。

该模块可在两种缓冲模式下工作。在标准缓冲模式下，通过单个串行缓冲区移位数据。在增强型缓冲模式下，通过FIFO缓冲区移位数据。FIFO深度取决于所配置的模式。

该模块可以发送和接收从2到32位的可变长度数据。

> **注：** 无论是在标准还是增强型缓冲模式下，都不要对SPIxBUF寄存器执行读-修改-写操作（如位操作指令）。

工作在主模式或从模式下时，该模块还支持基本的帧SPI协议。共支持四种帧SPI配置。

该模块还支持音频模式。有4种不同的音频模式可供使用。

- I^{2}S模式
- 左对齐模式
- 右对齐模式
- PCM/DSP模式

在其中每种模式下，串行时钟是自由运行的，而音频数据总是在传输。

如果在两个器件之间发生音频协议数据传输，则通常一个器件是主器件，另一个是从器件。但是，音频数据可以在两个从器件之间传输。由于音频协议需要自由运行的时钟，所以主器件可以是第三方控制器。在两种情况下，主器件都会产生两个自由运行的时钟：SCKx和LRC（左右声道时钟/$\overline{\text{SSx}}$/FSYNC）。

SPI串行接口由以下四个引脚组成：

- SDIx：串行数据输入
- SDOx：串行数据输出
- SCKx：移位时钟输入或输出
- $\overline{\text{SSx}}$：低电平有效从选择或帧同步I/O脉冲

SPI模块可以被配置为使用2个、3个或4个引脚工作。在3引脚模式下，不使用$\overline{\text{SSx}}$。在2引脚模式下，不使用SDOx和$\overline{\text{SSx}}$。

SPI模块可以产生3个反映在数据通信期间所发生事件的中断。它可以产生以下类型的中断：

1. 接收中断，通过SPIxRXIF来指示。在以下条件下会发生该事件：
   - 接收水印中断
   - SPIROV = 1
   - SPIRBF = 1
   - SPIRBE = 1

   前提是在SPIxIMSKL/H中使能了相应的屏蔽位。
2. 发送中断，通过SPIxTXIF来指示。在以下条件下会发生该事件：
   - 发送水印中断
   - SPITUR = 1
   - SPITBF = 1
   - SPITBE = 1

   前提是在SPIxIMSKL/H中使能了相应的屏蔽位。
3. 通用中断，通过SPIxIF来指示。在以下条件下会发生该事件：
   - FRMERR = 1
   - SPIBUSY = 1
   - SRMT = 1

   前提是在SPIxIMSKL/H中使能了相应的屏蔽位。

增强型缓冲模式下的模块框图如图17-1所示。

> **注：** 在本章中，SPI模块统称为SPIx，或分别称为SPI1、SPI2或SPI3。特殊功能寄存器也使用类似的符号表示。例如，SPIxCON1和SPIxCON2表示这3个SPI模块中任何一个的控制寄存器。

17.1 主模式操作

要将SPIx模块设置为主模式操作，请执行以下步骤：

1. 在相应IECx寄存器中禁示SPIx中断。
2. 通过清零SPIEN位停止并复位SPIx模块。
3. 清除接收缓冲区。
4. 如果使用标准缓冲模式，则将ENHBUF位（SPIxCON1L<0>）清零；如果使用增强型缓冲模式，则将该位置1。
5. 如果不使用SPIx中断，请跳过此步骤。否则，请执行以下附加步骤：
   a) 清除相应IFSx寄存器中的SPIx中断标志/事件。
   b) 向相应IPCx寄存器中写入SPIx中断优先级和次优先级位。
   c) 将相应IECx寄存器中的SPIx中断允许位置1。
6. 写入波特率寄存器SPIxBRGL。
7. 清零SPIROV位（SPIxSTATL<6>）。
8. 设置MSTEN（SPIxCON1L<5>）= 1并将所需设置写入SPIxCON1L寄存器。
9. 通过将SPIEN位（SPIxCON1L<15>）置1使能SPI操作。
10. 将待发送数据写入SPIxBUFL和SPIxBUFH寄存器。数据一写入SPIxBUFL/H寄存器，发送（和接收）就会立即开始。

17.2 从模式操作

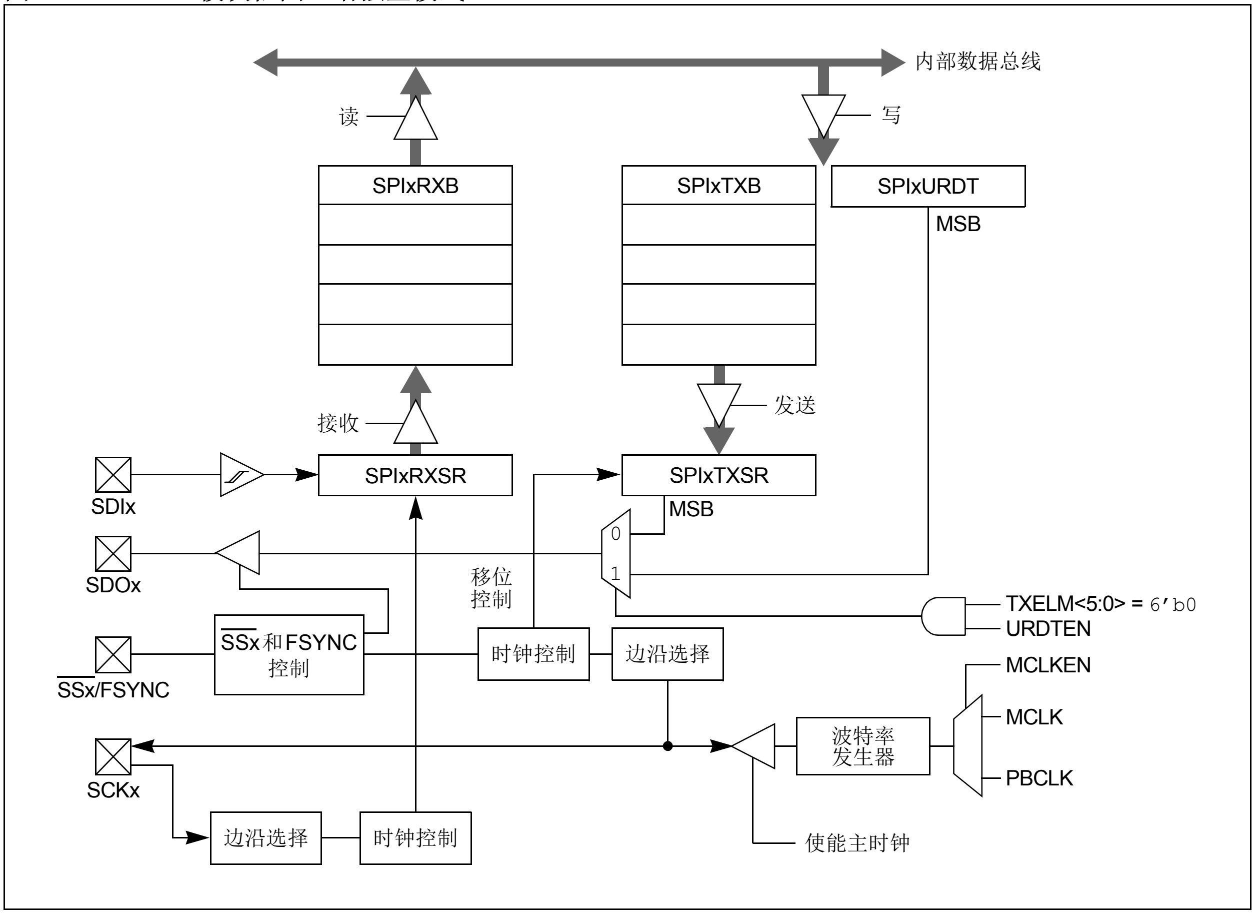
要将SPIx模块设置为从模式操作，请执行以下步骤：

1. 如果使用中断，则在相应IECx寄存器中禁示SPIx中断。
2. 通过清零SPIEN位停止并复位SPIx模块。
3. 清除接收缓冲区。
4. 如果使用标准缓冲模式，则将ENHBUF位（SPIxCON1L<0>）清零；如果使用增强型缓冲模式，则将该位置1。
5. 如果使用中断，则执行以下附加步骤：
   a) 清除相应IFSx寄存器中的SPIx中断标志/事件。
   b) 向相应IPCx寄存器中写入SPIx中断优先级和次优先级位。
   c) 将相应IECx寄存器中的SPIx中断允许位置1。
6. 清零SPIROV位（SPIxSTATL<6>）。
7. 设置MSTEN（SPIxCON1L<5>）= 0，并将所需设置写入SPIxCON1L寄存器。
8. 通过将SPIEN位（SPIxCON1L<15>）置1使能SPI操作。
9. 主器件一提供串行时钟，发送（和接收）就会立即开始。

从模式下提供以下附加功能：

- 从选择同步：
  $\overline{\text{SSx}}$引脚允许器件工作于同步从模式。如果将SSEN位（SPIxCON1L<7>）置1，则仅当将$\overline{\text{SSx}}$引脚驱动为低电平状态时，才会在从模式下使能发送和接收。为了使$\overline{\text{SSx}}$引脚用作输入，不能驱动端口输出或其他外设输出。如果将SSEN位置1且$\overline{\text{SSx}}$引脚驱动为高电平，则即使模块在发送过程中，SDOx引脚也不会再被驱动并将处于三态。下次使用SPIxTXB寄存器中存放的数据将$\overline{\text{SSx}}$引脚驱动为低电平时，将再次尝试中止的发送。如果未将SSEN位置1，则$\overline{\text{SSx}}$引脚不会影响从模式下的模块操作。
- SPITBE状态标志操作：
  在从模式操作中，SPITBE位（SPIxSTATL<3>）具有不同的功能。下面介绍了从模式操作各种设置的SPITBE功能：
  - 如果清零SSEN（SPIxCON1L<7>），则当用户代码加载SPIxBUF时，将清零SPITBE位。当模块将SPIxTXB中数据传送到SPIxTXSR时，该位置1。这与主模式下的SPITBE位功能类似。
  - 如果将SSEN置1，则当用户代码加载SPIxBUF时，将清零SPITBE位。不过，仅当SPIx模块完成数据传输时，该位才置1。当$\overline{\text{SSx}}$引脚变为高电平时，将中止发送并且稍后可能重试。这样，每个数据字都保存在SPIxTXB中，直到所有位都发送到接收器。

图17-1： SPIx模块框图（增强型模式）



17.3 音频模式操作

要将SPIx模块初始化为音频模式，请按照以下步骤将其初始化为主/从模式，同时将AUDEN位（SPIxCON1H<15>）置1。在主模式+音频模式下：

- 只要SPIEN位（SPIxCON1L<15>）= 1，此模式就会使能器件生成SCKx和LRC脉冲。
- 在主模式下，SPIx模块在任何情况下都将连续生成LRC和SCKx，无论是否发送数据。
- SPIx模块在一个SCKx周期内输出LRC和SCKx的前沿，并且串行数据连续移入和移出，即使TX FIFO为空时也是如此。

在从模式+音频模式下：

- 只要SPIEN位（SPIxCON1L<15>）= 1，此模式就会使能器件接收SCKx和LRC脉冲。
- SPIx模块在SDOx上输出零，但直到模块接收到LRC（即左声道的前沿）才会移出或移入数据（SDIx）。
- 模块接收到LRC的前沿之后，如果DISSDI（SPIxCON1L<4>）= 0，则开始接收数据，并且串行数据将连续移出，即使TX FIFO为空时也是如此。

17.4 SPI控制寄存器

寄存器17-1： SPIxCON1L：SPIx控制寄存器1低位字

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPIEN	—	SPISIDL	DISSDO	MODE32[1,4]	MODE16[1,4]	SMP	CKE[1]
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSEN[2]	CKP	MSTEN	DISSDI	DISSCK	MCLKEN[3]	SPIFE	ENHBUF
bit 7							bit 0

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

bit 15 **SPIEN：**SPIx使能位
1 = 使能模块
0 = 关闭并复位模块，禁止时钟，禁止中断事件产生，允许SFR修改

bit 14 **未实现：**读为0

bit 13 **SPISIDL：**SPIx空闲模式停止位
1 = 在CPU空闲模式下停止工作
0 = 在CPU空闲模式下继续工作

bit 12 **DISSDO：**禁止SDOx输出端口位
1 = 模块不使用SDOx引脚；引脚由端口功能控制
0 = SDOx引脚由模块控制

bit 11-10 **MODE<32,16>：**串行字长度位[1,4]

AUDEN = 0：

MODE32	MODE16	通信
1	x	32位
0	1	16位
0	0	8位

AUDEN = 1：

MODE32	MODE16	通信
1	1	24位数据，32位FIFO，32位声道/64位帧
1	0	32位数据，32位FIFO，32位声道/64位帧
0	1	16位数据，16位FIFO，32位声道/64位帧
0	0	16位数据，16位FIFO，16位声道/32位帧

bit 9 **SMP：**SPIx数据输入采样阶段位
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
从模式：
总是在数据输出时间的中间采样输入数据，而不管SMP设置如何。

注 1： 当AUDEN = 1时，该模块按CKE = 0时的情况工作，不论其实际值如何。
2： 当FRMEN = 1时，不使用SSEN。
3： 只有SPIEN位 = 0时，才能写入MCLKEN。
4： 该声道对于DSP/PCM模式没有意义，因为LRC会遵循FRMSYPW位。

寄存器17-1： SPIxCON1L：SPIx控制寄存器1低位字（续）

bit 8 **CKE**：SPIx时钟边沿选择位[1]
1 = 时钟状态从有效转换到空闲时发送
0 = 时钟状态从空闲转换到有效时发送

bit 7 **SSEN**：从选择使能位（从模式）[2]
1 = 宏在从模式下使用$\overline{\text{SSx}}$引脚；$\overline{\text{SSx}}$引脚用作从选择输入
0 = 宏不使用$\overline{\text{SSx}}$引脚（$\overline{\text{SSx}}$引脚将由端口I/O控制）

bit 6 **CKP**：SPIx时钟极性选择位
1 = 空闲状态时时钟信号为高电平；工作状态时为低电平
0 = 空闲状态时时钟信号为低电平；工作状态时为高电平

bit 5 **MSTEN**：主模式使能位
1 = 主模式
0 = 从模式

bit 4 **DISSDI**：禁止SDIx输入端口位
1 = 模块不使用SDIx引脚；引脚由端口功能控制
0 = SDIx引脚由模块控制

bit 3 **DISSCK**：禁止SCKx输出端口位
1 = 模块不使用SCKx引脚；引脚由端口功能控制
0 = SCKx引脚由模块控制

bit 2 **MCLKEN**：主时钟使能位[3]
1 = BRG使用MCLK
0 = BRG使用PBCLK

bit 1 **SPIFE**：帧同步脉冲边沿选择位
1 = 帧同步脉冲（从空闲转换到有效的边沿）与第一个位时钟同时出现
0 = 帧同步脉冲（从空闲转换到有效的边沿）在第一个位时钟之前出现

bit 0 **ENHBUF**：增强型缓冲模式使能位
1 = 使能增强型缓冲模式
0 = 禁止增强型缓冲模式

注 **1**： 当AUDEN = 1时，该模块按CKE = 0时的情况工作，不论其实际值如何。
2： 当FRMEN = 1时，不使用SSEN。
3： 只有SPIEN位 = 0时，才能写入MCLKEN。
4： 该声道对于DSP/PCM模式没有意义，因为LRC会遵循FRMSYPW位。

寄存器17-2： SPIxCON1H：SPIx控制寄存器1高位字

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
AUDEN[1]	SPISGNEXT	IGNROV	IGNTUR	AUDMONO[2]	URDTEN[3]	AUDMOD1[4]	AUDMOD0[4]
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FRMEN	FRMSYNC	FRMPOL	MSSEN	FRMSYPW	FRMCNT2	FRMCNT1	FRMCNT0
bit 7							bit 0

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

bit 15 **AUDEN**：音频编解码器支持使能位[1]
1 = 使能音频协议；MSTEN控制SCKx和帧（也称为LRC）的方向，并且该模块按FRMEN = 1、FRMSYNC = MSTEN、FRMCNT<2:0> = 001和SMP = 0时的情况工作，不论它们的实际值如何
0 = 禁止音频协议

bit 14 **SPISGNEXT**：SPIx符号扩展RX FIFO读取数据使能位
1 = 对接收FIFO中的数据进行符号扩展
0 = 不对接收FIFO中的数据进行符号扩展

bit 13 **IGNROV**：忽略接收溢出位
1 = 接收溢出（ROV）不是关键错误；在ROV期间，FIFO中的数据不会被接收数据覆盖
0 = ROV是关键错误，会停止SPI操作

bit 12 **IGNTUR**：忽略发送数据不足位
1 = 发送不足（TUR）不是关键错误，发送由URDTEN指示的数据，直到SPIxTXB不为空
0 = TUR是关键错误，会停止SPI操作

bit 11 **AUDMONO**：音频数据格式发送位[2]
1 = 音频数据为单声道（即每个数据字在左右声道同时发送）
0 = 音频数据为立体声

bit 10 **URDTEN**：发送数据不足数据使能位[3]
1 = 在发送数据不足条件期间发送SPIxURDTL/H寄存器的数据
0 = 在发送数据不足条件期间发送最后接收到的数据

bit 9-8 **AUDMOD<1:0>**：音频协议模式选择位[4]
11 = PCM/DSP模式
10 = 右对齐模式：该模块按SPIFE = 1时的情况工作，不论其实际值如何
01 = 左对齐模式：该模块按SPIFE = 1时的情况工作，不论其实际值如何
00 = I^2S模式：该模块按SPIFE = 0时的情况工作，不论其实际值如何

bit 7 **FRMEN**：帧SPIx支持位
1 = 使能帧SPIx支持（$\overline{\text{SSx}}$引脚用作FSYNC输入/输出）
0 = 禁止帧SPIx支持

注 1：只有SPIEN位 = 0时，才能写入AUDEN。
2：只有SPIEN位 = 0时才能写入AUDMONO，并且该位仅在AUDEN = 1时有效。
3：只有IGNTUR = 1时，URDTEN才有效。
4：只有SPIEN位 = 0时才能写入AUDMOD<1:0>位，并且该位仅在AUDEN = 1时有效。不处于PCM/DSP模式时，该模块按FRMSYPW = 1时的情况工作，不论其实际值如何。

寄存器17-2： SPIxCON1H：SPIx控制寄存器1高位字（续）

bit 6 **FRMSYNC**：帧同步脉冲方向控制位
1 = 帧同步脉冲输入（从器件）
0 = 帧同步脉冲输出（主器件）

bit 5 **FRMPOL**：帧同步/从选择极性位
1 = 帧同步脉冲/从选择为高电平有效
0 = 帧同步脉冲/从选择为低电平有效

bit 4 **MSSEN**：主模式从选择使能位
1 = 使能SPIx从选择支持，其极性由FRMPOL决定（在主模式下在发送期间自动驱动 $\overline{\text{SSx}}$ 引脚）
0 = 禁止SPIx从选择支持（$\overline{\text{SSx}}$ 引脚将由端口IO控制）

bit 3 **FRMSYPW**：帧同步脉冲宽度位
1 = 帧同步脉冲为1个串行字长宽（由MODE<32,16>/WLENGTH<4:0>定义）
0 = 帧同步脉冲为1个时钟（SCK）宽

bit 2-0 **FRMCNT<2:0>**：帧同步脉冲计数器位
控制每个同步脉冲发送的串行字数。
111 = 保留
110 = 保留
101 = 每32个串行字产生一个帧同步脉冲
100 = 每16个串行字产生一个帧同步脉冲
011 = 每8个串行字产生一个帧同步脉冲
010 = 每4个串行字产生一个帧同步脉冲
001 = 每2个串行字产生一个帧同步脉冲（由音频协议使用的值）
000 = 每个串行字产生一个帧同步脉冲

注 1： 只有SPIEN位 = 0时，才能写入AUDEN。
2： 只有SPIEN位 = 0时才能写入AUDMONO，并且该位仅在AUDEN = 1时有效。
3： 只有IGNTUR = 1时，URDTEN才有效。
4： 只有SPIEN位 = 0时才能写入AUDMOD<1:0>位，并且该位仅在AUDEN = 1时有效。不处于PCM/DSP模式时，该模块按FRMSYPW = 1时的情况工作，不论其实际值如何。

寄存器17-3： **SPIxCON2L：SPIx控制寄存器2低位字**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	WLENGTH<4:0>[1,2]				
bit 7							bit 0

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

bit 15-5 **未实现：**读为0

bit 4-0 **WLENGTH<4:0>：**可变字长位[1,2]

11111 = 32位数据
11110 = 31位数据
11101 = 30位数据
11100 = 29位数据
11011 = 28位数据
11010 = 27位数据
11001 = 26位数据
11000 = 25位数据
10111 = 24位数据
10110 = 23位数据
10101 = 22位数据
10100 = 21位数据
10011 = 20位数据
10010 = 19位数据
10001 = 18位数据
10000 = 17位数据
01111 = 16位数据
01110 = 15位数据
01101 = 14位数据
01100 = 13位数据
01011 = 12位数据
01010 = 11位数据
01001 = 10位数据
01000 = 9位数据
00111 = 8位数据
00110 = 7位数据
00101 = 6位数据
00100 = 5位数据
00011 = 4位数据
00010 = 3位数据
00001 = 2位数据
00000 = 请参见SPIxCON1L<11:10>中的MODE<32,16>位

注 1： 只有AUDEN = 0时，这些位才有效。
2： 通过更改这些位来改变长度并不会影响发送/接收FIFO的深度。

寄存器17-4： **SPIxSTATL：SPIx状态寄存器低位字**

U-0	U-0	U-0	R/C-0, HS	R-0, HSC	U-0	U-0	R-0, HSC
—	—	—	FRMERR	SPIBUSY	—	—	SPITUR[1]
bit 15							bit 8

R-0, HSC	R/C-0, HS	R-1, HSC	U-0	R-1, HSC	U-0	R-0, HSC	R-0, HSC
SRMT	SPIROV	SPIRBE	—	SPITBE	—	SPITBF	SPIRBF
bit 7							bit 0

图注：	C = 可清零位	HS = 硬件置1位	x = 未知
R = 可读位	W = 可写位	0 = 清零	HSC = 硬件置1/清零位
-n = POR时的值	1 = 置1	U = 未实现位，读为0	

bit 15-13 **未实现：**读为0

bit 12 **FRMERR：**SPIx帧错误状态位
1 = 检测到帧错误
0 = 未检测到帧错误

bit 11 **SPIBUSY：**SPIx活动状态位
1 = 模块当前正忙于处理一些事务
0 = 没有正在进行的事务（在读取时）

bit 10-9 **未实现：**读为0

bit 8 **SPITUR：**SPIx发送数据不足状态位[1]
1 = 发送缓冲区遇到发送数据不足条件
0 = 发送缓冲区不存在发送数据不足条件

bit 7 **SRMT：**移位寄存器空状态位
1 = 没有正在进行或待处理的事务（即，SPIxTXB和SPIxTXSR都不包含要发送的数据）
0 = 有正在进行或待处理的事务

bit 6 **SPIROV：**SPIx接收溢出状态位
1 = 在SPIxRXB已满时，完全接收到新的字节/半字/字
0 = 未溢出

bit 5 **SPIRBE：**SPIx 接收缓冲区空状态位
1 = 接收缓冲区为空
0 = 接收缓冲区不为空
<u>标准缓冲模式：</u>
当读SPIxBUF来读SPIxRXB时，该位由硬件自动置1。当SPIx将数据从SPIxRXSR传输到SPIxRXB时，该位由硬件自动清零。
<u>增强型缓冲模式：</u>
指示RXELM<5:0> = 6'b000000。

bit 4 **未实现：**读为0

bit 3 **SPITBE：**SPIx发送缓冲区空状态位
1 = SPIxTXB为空
0 = SPIxTXB不为空
<u>标准缓冲模式：</u>
当SPIx将数据从SPIxTXB传输到SPIxTXSR时，该位由硬件自动置1。当写SPIxBUF来装入SPIxTXB时，该位由硬件自动清零。
<u>增强型缓冲模式：</u>
指示TXELM<5:0> = 6'b000000。

注 1： SPIEN = 0时，SPITUR会清零。当IGNTUR = 1时，SPITUR会提供发送数据不足条件的动态状态，但不会停止接收/发送操作，不需要用软件清零。

寄存器17-4： SPIxSTATL：SPIx状态寄存器低位字（续）

bit 2 **未实现：**读为0

bit 1 **SPITBF：**SPIx发送缓冲区满状态位
1 = SPIxTXB已满
0 = SPIxTXB未满
标准缓冲模式：
当写SPIxBUF来装入SPIxTXB时，该位由硬件自动置1。当SPIx将数据从SPIxTXB传输到SPIxTXSR时，该位由硬件自动清零。
增强型缓冲模式：
指示TXELM<5:0> = 6'b111111。

bit 0 **SPIRBF：**SPIx接收缓冲区满状态位
1 = SPIxRXB已满
0 = SPIxRXB未满
标准缓冲模式：
当SPIx将数据从SPIxRXSR传输到SPIxRXB时，该位由硬件自动置1。当读SPIxBUF来读SPIxRXB时，该位由硬件自动清零。
增强型缓冲模式：
指示RXELM<5:0> = 6'b111111。

注 1： SPIEN = 0时，SPITUR会清零。当IGNTUR = 1时，SPITUR会提供发送数据不足条件的动态状态，但不会停止接收/发送操作，不需要用软件清零。

寄存器17-5：　SPIxSTATH：SPIx状态寄存器高位字

U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
—	—	RXELM5[3]	RXELM4[2]	RXELM3[1]	RXELM2	RXELM1	RXELM0
bit 15							bit 8

U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
—	—	TXELM5[3]	TXELM4[2]	TXELM3[1]	TXELM2	TXELM1	TXELM0
bit 7							bit 0

图注：	HSC = 硬件置1/清零位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15-14　**未实现：**读为0

bit 13-8　**RXELM<5:0>：**接收缓冲区元素计数位（在增强型缓冲模式下有效）[1,2,3]

bit 7-6　**未实现：**读为0

bit 5-0　**TXELM<5:0>：**发送缓冲区元素计数位（在增强型缓冲模式下有效）[1,2,3]

注 **1：** 只有FIFODEPTH = 8或更高值时，才存在RXELM3和TXELM3位。

2： 只有FIFODEPTH = 16或更高值时，才存在RXELM4和TXELM4位。

3： 只有FIFODEPTH = 32时，才存在RXELM5和TXELM5位。

寄存器17-6： SPIxBUFL：SPIx缓冲区寄存器低位字

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATA<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATA<7:0>							
bit 7							bit 0

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

bit 15-0 **DATA<15:0>**：SPIx FIFO数据位

当MODE<32,16>或WLENGTH<4:0>位选择16到9位数据时，SPIx仅使用DATA<15:0>。当MODE<32,16>或WLENGTH<4:0>位选择8到2位数据时，SPIx仅使用DATA<7:0>。

寄存器17-7： SPIxBUFH：SPIx缓冲区寄存器高位字

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATA<31:24>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATA<23:16>							
bit 7							bit 0

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

bit 15-0 **DATA<31:16>**：SPIx FIFO数据位

当MODE<32,16>或WLENGTH<4:0>位选择32到25位数据时，SPIx使用DATA<31:16>。当MODE<32,16>或WLENGTH<4:0>位选择24到17位数据时，SPIx仅使用DATA<23:16>。

寄存器17-8： SPIxBRGL：SPIx波特率发生器寄存器低位字

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	BRG<12:8>[(1)]				
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRG<7:0>[(1)]							
bit 7							bit 0

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

bit 15-13 **未实现：**读为0

bit 12-0 **BRG<12:0>：**SPIx波特率发生器分频比位[(1)]

注 1： 当SPIEN = 1时，更改BRG值会导致未定义的行为。

寄存器17-9： **SPIxIMSKL：SPIx中断屏蔽寄存器低位字**

U-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	—	—	FRMERREN	BUSYEN	—	—	SPITUREN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0
SRMTEN	SPIROVEN	SPIRBEN	—	SPITBEN	—	SPITBFEN	SPIRBFEN
bit 7							bit 0

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

bit 15-13 **未实现：**读为0

bit 12 **FRMERREN：**允许通过FRMERR产生中断事件位
1 = 帧错误产生中断事件
0 = 帧错误不产生中断事件

bit 11 **BUSYEN：**允许通过SPIBUSY产生中断事件位
1 = SPIBUSY产生中断事件
0 = SPIBUSY不产生中断事件

bit 10-9 **未实现：**读为0

bit 8 **SPITUREN：**允许通过SPITUR产生中断事件位
1 = 发送数据不足（TUR）产生中断事件
0 = 发送数据不足不产生中断事件

bit 7 **SRMTEN：**允许通过SRMT产生中断事件位
1 = 移位寄存器空（SRMT）产生中断事件
0 = 移位寄存器空不产生中断事件

bit 6 **SPIROVEN：**允许通过SPIROV产生中断事件位
1 = SPIx接收溢出产生中断事件
0 = SPIx接收溢出不产生中断事件

bit 5 **SPIRBEN：**允许通过SPIRBE产生中断事件位
1 = SPIx接收缓冲区为空产生中断事件
0 = SPIx接收缓冲区为空不产生中断事件

bit 4 **未实现：**读为0

bit 3 **SPITBEN：**允许通过SPITBE产生中断事件位
1 = SPIx发送缓冲区为空产生中断事件
0 = SPIx发送缓冲区为空不产生中断事件

bit 2 **未实现：**读为0

bit 1 **SPITBFEN：**允许通过SPITBF产生中断事件位
1 = SPIx发送缓冲区已满产生中断事件
0 = SPIx发送缓冲区已满不产生中断事件

bit 0 **SPIRBFEN：**允许通过SPIRBF产生中断事件位
1 = SPIx接收缓冲区已满产生中断事件
0 = SPIx接收缓冲区已满不产生中断事件

寄存器17-10： SPIxIMSKH：SPIx中断屏蔽寄存器高位字

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RXWIEN	—	RXMSK5[1]	RXMSK4[1,4]	RXMSK3[1,3]	RXMSK2[1,2]	RXMSK1[1]	RXMSK0[1]
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TXWIEN	—	TXMSK5[1]	TXMSK4[1,4]	TXMSK3[1,3]	TXMSK2[1,2]	TXMSK1[1]	TXMSK0[1]
bit 7							bit 0

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

bit 15 **RXWIEN**：接收水印中断允许位
1 = RXMSK<5:0> ≤ RXELM<5:0>时触发接收缓冲区元素水印中断
0 = 禁止接收缓冲区元素水印中断

bit 14 **未实现**：读为0

bit 13-8 **RXMSK<5:0>**：接收缓冲区屏蔽位[1,2,3,4]
接收屏蔽位；与RXWIEN位配合使用。

bit 7 **TXWIEN**：发送水印中断允许位
1 = TXMSK<5:0> = TXELM<5:0>时触发发送缓冲区元素水印中断
0 = 禁止发送缓冲区元素水印中断

bit 6 **未实现**：读为0

bit 5-0 **TXMSK<5:0>**：发送缓冲区屏蔽位[1,2,3,4]
发送屏蔽位；与TXWIEN位配合使用。

注 **1**： 高于FIFODEPTH的屏蔽值是无效的。在这种情况下，模块对于任何值都不会触发匹配。
2： 只有FIFODEPTH = 8或更高值时，才存在RXMSK2和TXMSK2位。
3： 只有FIFODEPTH = 16或更高值时，才存在RXMSK3和TXMSK3位。
4： 只有FIFODEPTH = 32时，才存在RXMSK4和TXMSK4位。

寄存器17-11： **SPIxURDTL：SPIx数据不足数据寄存器低位字**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
URDATA<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
URDATA<7:0>							
bit 7							bit 0

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

bit 15-0 **URDATA<15:0>**：SPIx数据不足数据位
仅当URDTEN = 1时才使用这些位。发生发送数据不足条件时，该寄存器保存要发送的数据。
当MODE<32,16>或WLENGTH<4:0>位选择16到9位数据时，SPIx仅使用URDATA<15:0>。当MODE<32,16>或WLENGTH<4:0>位选择8到2位数据时，SPIx仅使用URDATA<7:0>。

寄存器17-12： **SPIxURDTH：SPIx数据不足数据寄存器高位字**

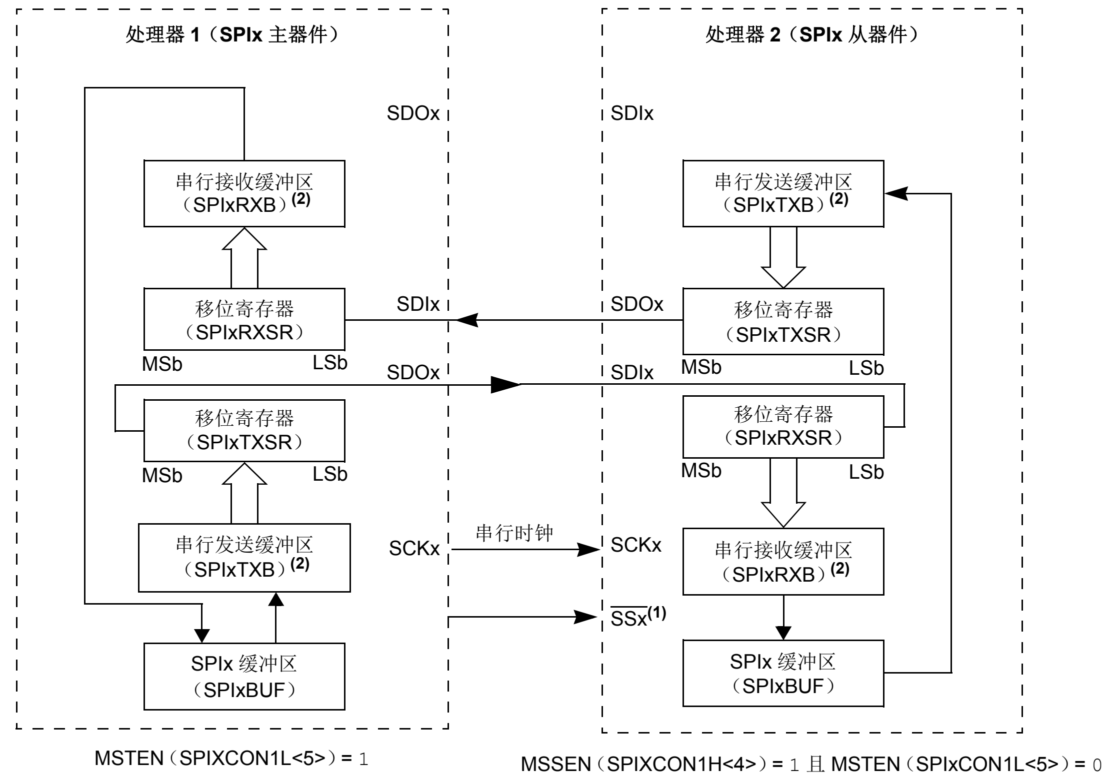
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
URDATA<31:24>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
URDATA<23:16>							
bit 7							bit 0

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

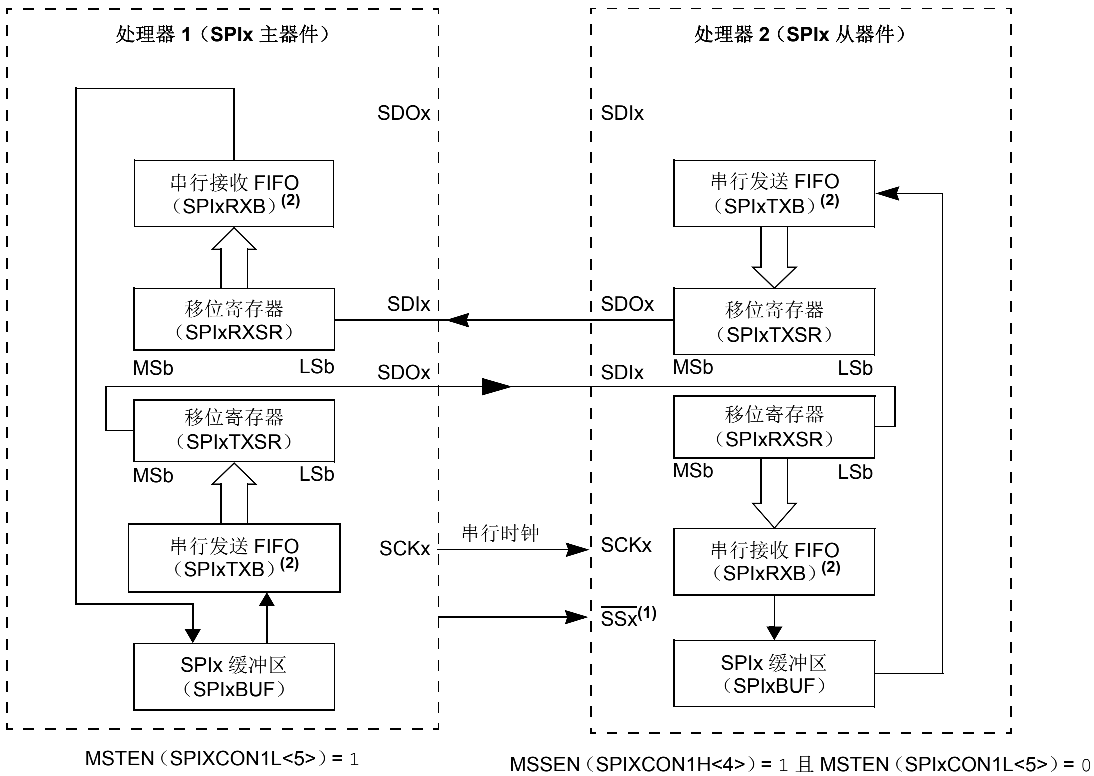
bit 15-0 **URDATA<31:16>**：SPIx数据不足数据位
仅当URDTEN = 1时才使用这些位。发生发送数据不足条件时，该寄存器保存要发送的数据。
当MODE<32,16>或WLENGTH<4:0>位选择32到25位数据时，SPIx仅使用URDATA<31:16>。当MODE<32,16>或WLENGTH<4:0>位选择24到17位数据时，SPIx仅使用URDATA<23:16>。

图17-2： SPIx主/从器件连接（标准模式）



注 1： 可选择在从模式下使用SSx引脚。
2： 用户必须向SPIxBUF写入发送数据，从SPIxBUF读取接收数据。SPIxTXB和SPIxRXB寄存器通过存储器映射到SPIxBUF。

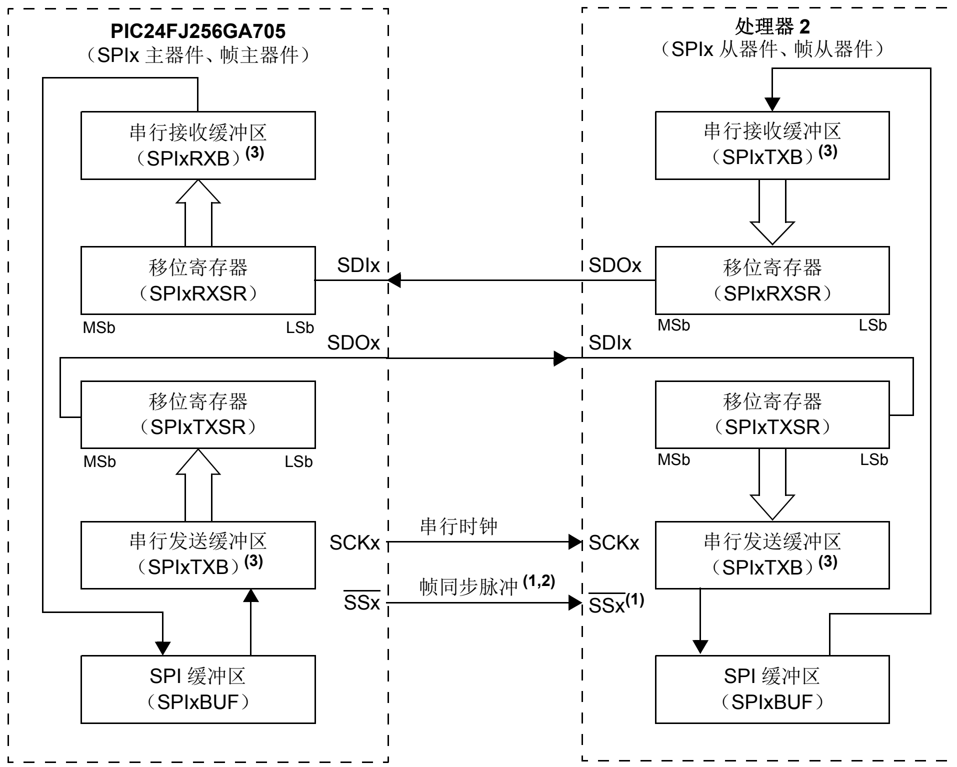
图17-3： SPIx主/从器件连接（增强型缓冲模式）



注 1： 可选择在从模式下使用$\overline{\text{SSx}}$引脚。

2： 用户必须向SPIxBUF写入发送数据，从SPIxBUF读取接收数据。SPIxTXB和SPIxRXB寄存器通过存储器映射到SPIxBUF。

图17-4：　SPIx主器件、帧主器件连接图



注 1：　在帧SPI模式下，$\overline{\text{SSx}}$引脚用于发送/接收帧同步脉冲。

2：　帧SPI模式需要使用全部四个引脚（即，无法选择使用$\overline{\text{SSx}}$引脚）。

3：　SPIxTXB和SPIxRXB寄存器通过存储器映射到SPIxBUF寄存器。

图17-5： SPIx主器件、帧从器件连接图

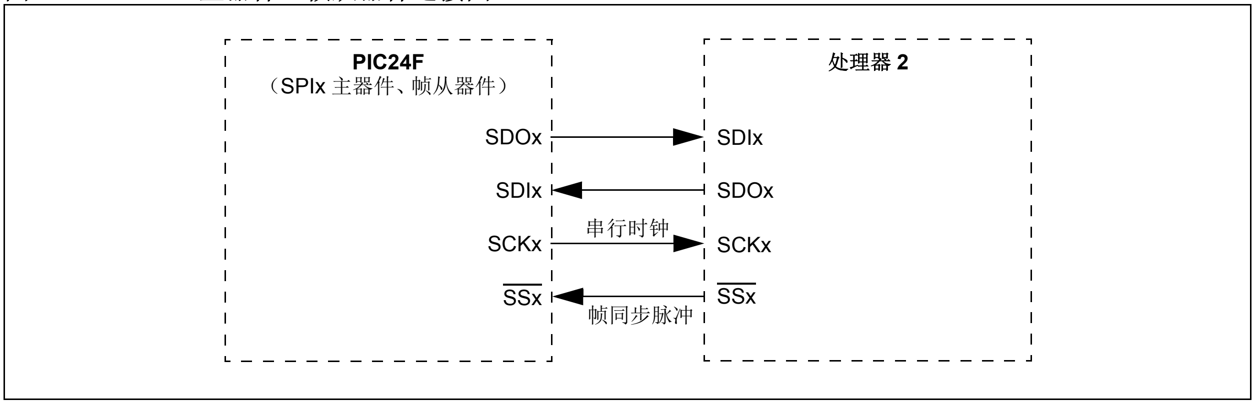


图17-6： SPIx从器件、帧主器件连接图

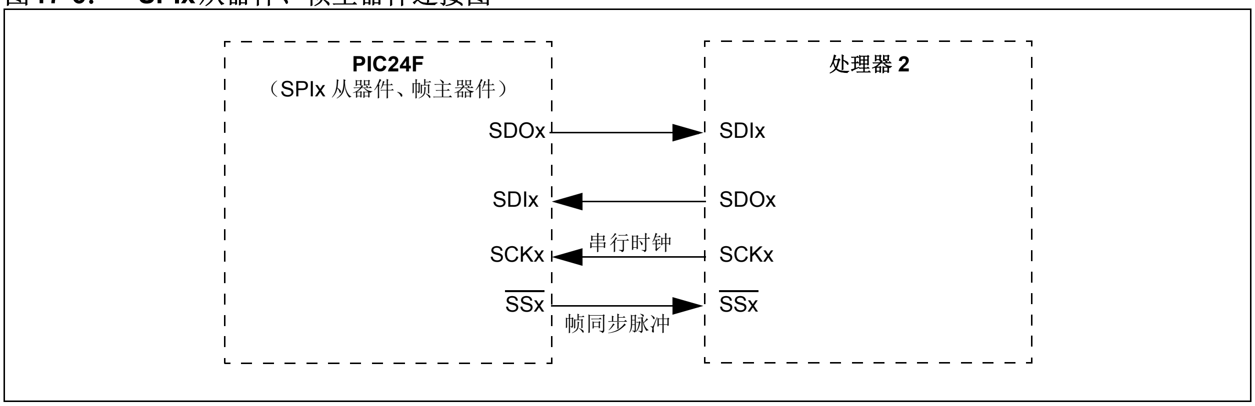
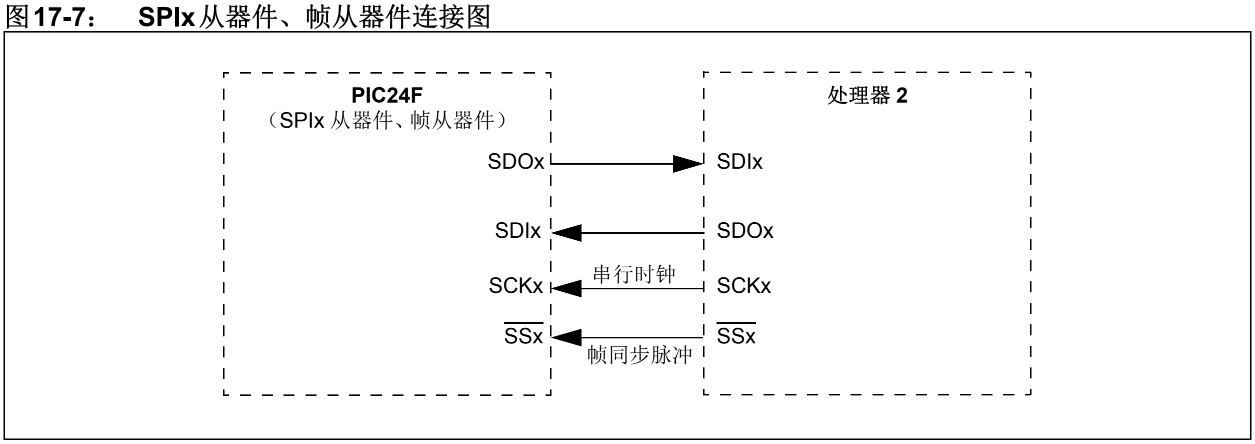


图17-7： SPIx从器件、帧从器件连接图



公式17-1： 器件速度与SPIx时钟速度之间的关系

$$\text{波特率} = \frac{F_{PB}}{(2 * (SPIxBRG + 1))}$$

其中：

F_{PB}为外设总线时钟频率。

18.0 I²C

> 注： 本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的“**I²C**”（DS70000195），该文档可从Microchip网站（www.microchip.com）获得。本数据手册中的信息取代了FRM中的信息。

I²C模块是用来与其他外设或单片机器件通信的串行接口。这些外设可以是串行EEPROM、显示驱动器和A/D转换器等。

I²C模块支持以下特性：

- 独立的主/从逻辑
- 7位和10位器件地址
- I²C协议中定义的广播呼叫地址
- 时钟延长，可为处理器提供用于响应从器件数据请求的延时
- 支持100 kHz和400 kHz两种总线规范
- 可配置的地址掩码
- 多主器件模式以防止仲裁时报文丢失
- 总线重发器模式，允许作为从器件接收来自所有地址的所有报文
- 自动SCL

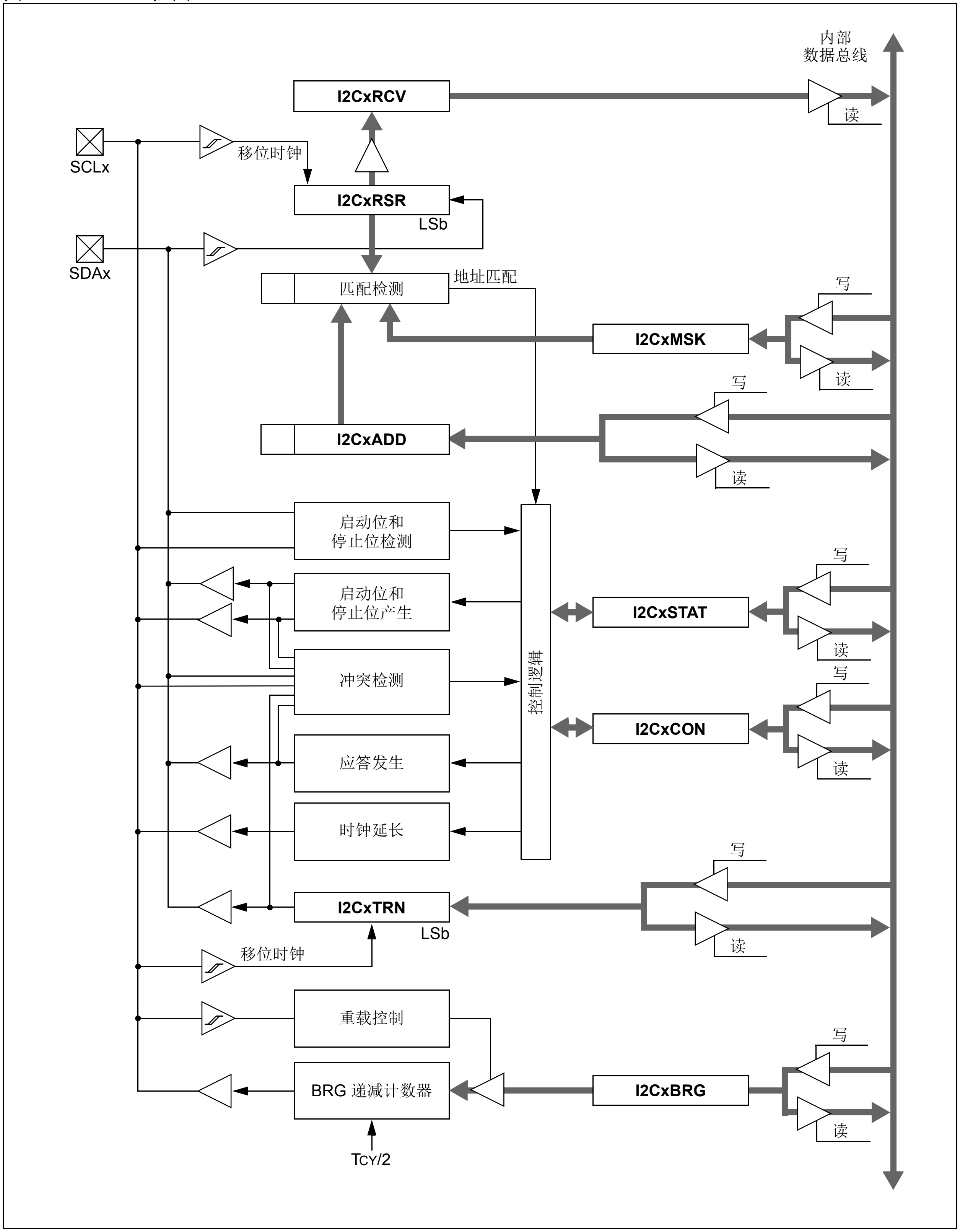
图18-1给出了该模块的框图。

18.1 作为主器件在单主器件环境中通信

在主模式下发送报文的细节取决于要与之进行通信的器件的通信协议。通常，事件序列如下：

1. 在SDAx和SCLx上发送一个启动条件。
2. 发送带有写操作指示的I²C器件地址字节到从器件。
3. 等待并验证从器件的应答。
4. 发送第一个数据字节（有时是命令）到从器件。
5. 等待并验证从器件的应答。
6. 发送串行存储器地址低字节到从器件。
7. 重复步骤4和5，直至发送完所有数据字节。
8. 在SDAx和SCLx上发送一个重复启动条件。
9. 发送带有读操作指示的器件地址字节到从器件。
10. 等待并验证从器件的应答。
11. 使能主器件接收以接收串行存储器数据。
12. 在数据字节接收完毕时，产生ACK或NACK条件。
13. 在SDAx和SCLx上产生一个停止条件。

图18-1： I2Cx框图


内部
数据总线
I2CxRCV
读
SCLx
移位时钟
I2CxRSR
LSb
SDAx
匹配检测
地址匹配
写
I2CxMSK
读
写
I2CxADD
读
启动位和
停止位检测
启动位和
停止位产生
I2CxSTAT
写
读
冲突检测
控制逻辑
写
I2CxCON
读
应答发生
时钟延长
写
I2CxTRN
LSb
读
移位时钟
重载控制
写
BRG 递减计数器
I2CxBRG
读
TCY/2

18.2 设置用作总线主器件时的波特率

使用公式18-1计算波特率发生器的重载值。

公式18-1： 计算波特率重载值[1,2,3]

$$F_{SCL} = \frac{F_{CY}}{(I2CxBRG + 2) * 2}$$

或：

$$I2CxBRG = \left[\frac{F_{CY}}{(F_{SCL} * 2)} - 2\right]$$

注 1： 基于FCY = FOSC/2；禁止打盹模式和PLL。
2： 这些时钟速率值仅供参考。实际的时钟速率与各个系统级参数有关。应在其实际应用中测量实际的时钟速率。
3： 禁止BRG值0和1。

18.3 从地址掩码

I2CxMSK寄存器（寄存器18-4）指定了7位和10位寻址模式下某些“无关”地址位位置。将I2CxMSK寄存器中某个位置1（= 1），不论相应的地址位是0还是1，从模块都会做出响应。例如，当将I2CxMSK设置为0010000000时，从模块将检测两个地址0000000000和0010000000。

为了使能地址掩码，必须通过将STRICT位（I2CxCONL<11>）清零来禁止智能外设管理接口（Intelligent Peripheral Management Interface，IPMI）。

> 注： 为了符合 I^2C 协议所作的更改，表18-2中是一些保留地址，工作在从模式下的I2C外设不会应答这些地址。这同样适用于包含任意这些地址的所有地址掩码设置。

表18-1： I2Cx时钟速率[1,2]

所需的系统FSCL	FCY	I2CxBRG值		实际FSCL
		（十进制）	（十六进制）	
100 kHz	16 MHz	78	4E	100 kHz
100 kHz	8 MHz	38	26	100 kHz
100 kHz	4 MHz	18	12	100 kHz
400 kHz	16 MHz	18	12	400 kHz
400 kHz	8 MHz	8	8	400 kHz
400 kHz	4 MHz	3	3	400 kHz
1 MHz	16 MHz	6	6	1.000 MHz
1 MHz	8 MHz	2	2	1.000 MHz

注 1： 基于FCY = FOSC/2；禁止打盹模式和PLL。
2： 这些时钟速率值仅供参考。实际的时钟速率与各个系统级参数有关。应在其实际应用中测量实际的时钟速率。

表18-2： I2Cx保留地址[1]

从地址	R/$\overline{W}$位	说明
0000 000	0	广播呼叫地址[2]
0000 000	1	启动字节
0000 001	x	Cbus地址
0000 01x	x	保留
0000 1xx	x	HS模式主代码
1111 0xx	x	10位从地址高字节[3]
1111 1xx	x	保留

注 1： 上述地址位永远不会导致地址匹配，无论地址掩码的设置如何。
2： 仅当GCEN = 1时，才会应答该地址。
3： 仅在10位寻址模式下才会与地址的高字节匹配。

寄存器18-1： I2CxCONL：I2Cx控制寄存器低位字

R/W-0	U-0	HC, R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL[(1)]	STRICT	A10M	DISSLW	SMEN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	HC, R/W-0	HC, R/W-0	HC, R/W-0	HC, R/W-0	HC, R/W-0
GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

图注：	HC = 硬件清零位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15 **I2CEN**：I2Cx使能位（只能用软件写入）
1 = 使能I2Cx模块并将SDAx和SCLx引脚配置为串行端口引脚
0 = 禁止I2Cx模块；所有I^2C引脚均由端口功能控制

bit 14 **未实现**：读为0

bit 13 **I2CSIDL**：I2Cx空闲模式停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 在空闲模式下模块继续工作

bit 12 **SCLREL**：SCLx释放控制位（仅在I^2C从模式下）[(1)]
模块复位以及（I2CEN = 0）设置SCLREL = 1。
如果STREN = 0：[(2)]
1 = 释放时钟
0 = 强制时钟为低电平（时钟延长）
如果STREN = 1：
1 = 释放时钟
0 = 保持时钟为低电平（时钟延长）；用户可以将该位设定为0，在下一次SCLx为低电平时进行时钟延长

bit 11 **STRICT**：I2Cx严格保留地址规则使能位
1 = 强制执行严格保留寻址（有关保留地址的信息，请参见表18-2）
在从模式下：器件不对保留地址空间进行响应，对于属于该类别的地址产生NACK。
在主模式下：允许器件生成位于保留地址空间的地址。
0 = 对保留寻址产生应答
在从模式下：器件会对属于保留地址空间的地址进行响应。当与任意保留地址匹配时，器件将产生ACK。
在主模式下：保留。

bit 10 **A10M**：10位从器件地址标志位
1 = I2CxADD为10位从器件地址
0 = I2CxADD为7位从器件地址

bit 9 **DISSLW**：压摆率控制禁止位
1 = 标准速度模式下禁止压摆率控制（100 kHz，对于1 MHz模式也禁止）
0 = 高速模式下使能压摆率控制（400 kHz）

注 1： 在从器件发送开始时自动清零；在从器件接收结束时自动清零。用户软件必须在写入发送缓冲区和将SCLREL位置1之间提供一段延时。该延时必须大于从器件发送的最小建立时间，如**第32.0节“电气特性”**中所述。
2： 在从器件发送开始时自动清零。

寄存器18-1： I2CxCONL：I2Cx控制寄存器低位字（续）

bit 8 **SMEN**：SMBus输入电平使能位
1 = 使能输入逻辑以使阈值符合SMBus规范
0 = 禁止SMBus特定输入

bit 7 **GCEN**：广播呼叫使能位（仅在I^2C从模式下）
1 = 允许在I2CxRSR中接收到广播呼叫地址时产生中断；已使能模块接收
0 = 禁止广播呼叫地址

bit 6 **STREN**：SCLx时钟延长使能位
仅在I^2C从模式下；与SCLREL位配合使用。
1 = 使能时钟延长
0 = 禁止时钟延长

bit 5 **ACKDT**：应答数据位
在I^2C主模式下，在主器件接收模式期间。当用户在接收结束时发出一个应答序列时发送的值。
在I^2C从模式下，当AHEN = 1或DHEN = 1时。当从器件在地址或数据接收结束时发出一个应答序列时，发送该值。
1 = 发送NACK
0 = 发送ACK

bit 4 **ACKEN**：应答序列使能位
仅在I^2C主模式下；在主器件接收模式期间适用。
1 = 在SDAx和SCLx引脚上发出应答序列，并发送ACKDT数据位
0 = 应答序列空闲

bit 3 **RCEN**：接收使能位（仅在I^2C主模式下）
1 = 使能I^2C接收模式；在8位接收数据字节结束时由硬件自动清零
0 = 接收序列不在进行中

bit 2 **PEN**：停止条件使能位（仅在I^2C主模式下）
1 = 在SDAx和SCLx引脚上发出停止条件
0 = 停止条件空闲

bit 1 **RSEN**：重复启动条件使能位（仅在I^2C主模式下）
1 = 在SDAx和SCLx引脚上发出重复启动条件
0 = 重复启动条件空闲

bit 0 **SEN**：启动条件使能位（仅在I^2C主模式下）
1 = 在SDAx和SCLx引脚上发出启动条件
0 = 启动条件空闲

注 1：在从器件发送开始时自动清零；在从器件接收结束时自动清零。用户软件必须在写入发送缓冲区和将SCLREL位置1之间提供一段延时。该延时必须大于从器件发送的最小建立时间，如**第32.0节“电气特性”**中所述。

2：在从器件发送开始时自动清零。

寄存器18-2： I2CxCONH：I2Cx控制寄存器高位字

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	PCIE	SCIE	BOEN	SDAHT[(1)]	SBCDE	AHEN	DHEN
bit 7							bit 0

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

bit 15-7 **未实现：**读为0

bit 6 **PCIE：**停止条件中断允许位（仅在I^2C从模式下）
1 = 允许在检测到停止条件时产生中断
0 = 禁止停止条件检测中断

bit 5 **SCIE：**启动条件中断允许位（仅在I^2C从模式下）
1 = 允许在检测到启动或重复启动条件时产生中断
0 = 禁止启动条件检测中断

bit 4 **BOEN：**缓冲区改写使能位（仅在I^2C从模式下）
1 = 仅当RBF位 = 0时，在接收到地址/数据字节时，更新I2CxRCV且产生ACK信号，忽略I2COV位的状态
0 = 仅当I2COV位清零时更新I2CxRCV

bit 3 **SDAHT：**SDAx保持时间选择位[(1)]
1 = 在SCLx的下降沿之后，在SDAx上最少有300 ns的保持时间
0 = 在SCLx的下降沿之后，在SDAx上最少有100 ns的保持时间

bit 2 **SBCDE：**从模式总线冲突检测使能位（仅在I^2C从模式下）
如果在SCLx的上升沿，在模块输出高电平状态时采样到SDAx为低电平，则BCL位会置1，总线会变为空闲状态。此检测模式仅在数据和ACK发送序列期间有效。
1 = 允许从模式总线冲突中断
0 = 禁止从模式总线冲突中断

bit 1 **AHEN：**地址保持使能位（仅在I^2C从模式下）
1 = 在所接收匹配地址字节的第8个SCLx下降沿之后，SCLREL位（I2CxCONL<12>）将清零，SCLx将保持低电平
0 = 禁止地址保持

bit 0 **DHEN：**数据保持使能位（仅在I^2C从模式下）
1 = 在接收数据字节的SCLx的第8个下降沿之后，从硬件清零SCLREL位（I2CxCONL<12>）且SCLx保持低电平。
0 = 禁止数据保持

注 1： 必须将该位设置为0才能进行1 MHz操作。

寄存器18-3： I2CxSTAT：I2Cx状态寄存器

HSC, R-0	HSC, R-0	HSC, R-0	U-0	U-0	HSC, R/C-0	HSC, R-0	HSC, R-0
ACKSTAT	TRSTAT	ACKTIM	—	—	BCL	GCSTAT	ADD10
bit 15							bit 8

HS, R/C-0	HS, R/C-0	HSC, R-0	HSC, R/C-0	HSC, R/C-0	HSC, R-0	HSC, R-0	HSC, R-0
IWCOL	I2COV	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	RBF	TBF
bit 7							bit 0

图注：	C = 可清零位	HS = 硬件置1位	0 = 清零
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	HSC = 硬件置1/清零位	

bit 15 **ACKSTAT**：应答状态位（在所有主模式和从模式下更新）
1 = 未收到来自从器件的应答
0 = 收到来自从器件的应答

bit 14 **TRSTAT**：发送状态位（作为I^{2}C主器件工作时，适用于主器件发送操作）
1 = 主器件正在进行发送（8位 + ACK）
0 = 主器件不在进行发送

bit 13 **ACKTIM**：应答时间状态位（仅在I^{2}C从模式下有效）
1 = 指示I^{2}C总线处于应答序列中，在SCLx时钟的第8个下降沿置1
0 = 不处于应答序列，在SCLx时钟的第9个上升沿清零

bit 12-11 **未实现**：读为0

bit 10 **BCL**：总线冲突检测位（主/从模式；在禁止I^{2}C模块（I2CEN = 0）时清零）
1 = 主器件或从器件发送操作期间检测到了总线冲突
0 = 未检测到总线冲突

bit 9 **GCSTAT**：广播呼叫状态位（在检测到停止条件后清零）
1 = 接收到广播呼叫地址
0 = 未接收到广播呼叫地址

bit 8 **ADD10**：10位地址状态位（在检测到停止条件后清零）
1 = 10位地址匹配
0 = 10位地址不匹配

bit 7 **IWCOL**：I2Cx写冲突检测位
1 = 因为I^{2}C模块忙，尝试写I2CxTRN寄存器的操作失败；必须用软件清零
0 = 未发生冲突

bit 6 **I2COV**：I2Cx接收溢出标志位
1 = 在I2CxRCV寄存器仍存放先前字节时接收到字节；在发送模式下，I2COV是“无关”位，必须用软件清零
0 = 无溢出

bit 5 **D/$\overline{\text{A}}$**：数据/地址位（作为I^{2}C从器件工作时）
1 = 指示上次接收的字节为数据
0 = 指示上次接收或发送的字节是地址

bit 4 **P**：I2Cx停止位
在检测到启动、复位或停止条件时更新；在禁止I^{2}C模块（I2CEN = 0）时清零。
1 = 指示上次检测到停止位
0 = 上次未检测到停止位

寄存器18-3： I2CxSTAT：I2Cx状态寄存器（续）

bit 3 **S**：I2Cx启动位
在检测到启动、复位或停止条件时更新；在禁止I^2C模块（I2CEN = 0）时清零。
1 = 指示上次检测到启动（或重复启动）位
0 = 上次未检测到启动（或重复启动）位

bit 2 **R/$\overline{W}$**：读/$\overline{写}$信息位（作为I^2C从器件工作时）
1 = 读：表示数据自从器件输出
0 = 写：表示数据输入到从器件

bit 1 **RBF**：接收缓冲区满状态位
1 = 接收完成，I2CxRCV已满
0 = 接收未完成，I2CxRCV为空

bit 0 **TBF**：发送缓冲区满状态位
1 = 正在进行发送，I2CxTRN已满（8位数据）
0 = 发送完成，I2CxTRN为空

寄存器18-4： I2CxMSK：I2Cx从模式地址掩码寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	MSK<9:8>	
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MSK<7:0>							
bit 7							bit 0

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

bit 15-10 **未实现**：读为0

bit 9-0 **MSK<9:0>**：地址中bit x的I2Cx掩码选择位
1 = 使能输入报文地址中bit x的掩码；在此位置上不需要位匹配
0 = 禁止bit x的掩码；在此位置上需要位匹配

19.0 通用异步收发器（UART）

> **注：** 本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的**“UART”**（DS39708），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

通用异步收发器（Universal Asynchronous Receiver Transmitter，UART）模块是PIC24F系列器件提供的串行I/O模块之一。UART是可以与外设（如个人计算机、LIN/J2602、RS-232和RS-485接口）通信的全双工异步系统。该模块还通过$\overline{\text{UxCTS}}$和$\overline{\text{UxRTS}}$引脚支持硬件流控制选项。UART模块包含IrDA编码器/解码器单元。

PIC24FJ256GA705系列器件配有两个UART模块，称为UART1和UART2。

UARTx模块的主要特性有：

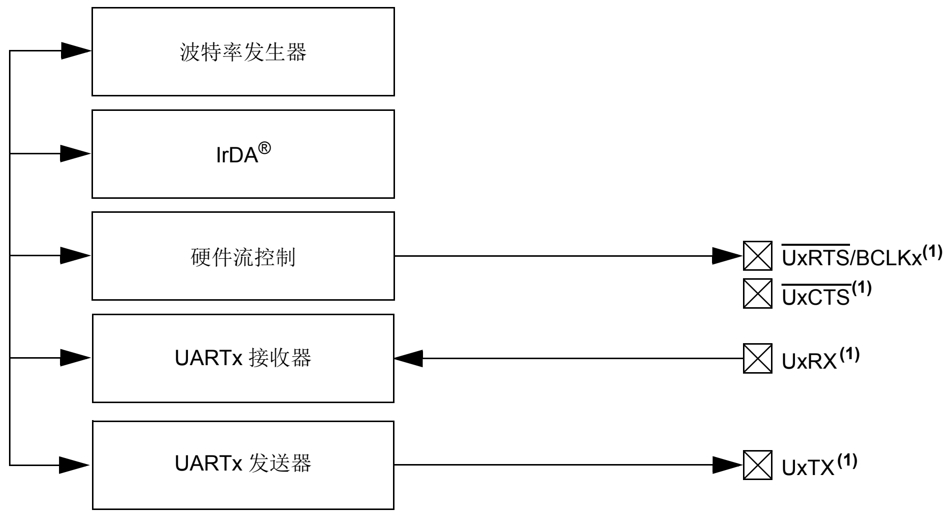
- 通过UxTX和UxRX引脚进行全双工8位或9位数据传输
- 偶校验、奇校验或无奇偶校验选项（对于8位数据）
- 一个或两个停止位
- 通过$\overline{\text{UxCTS}}$和$\overline{\text{UxRTS}}$引脚支持硬件流控制选项
- 完全集成的波特率发生器，具有16位预分频器
- 在16x模式下以16 MIPS速率工作时，波特率范围从最高1 Mbps到最低15 Hz
- 在4x模式下以16 MIPS速率工作时，波特率范围从最高4 Mbps到最低61 Hz
- 4级深先进先出（First-In-First-Out，FIFO）发送数据缓冲区
- 4级深FIFO接收数据缓冲区
- 奇偶校验错误、帧错误和缓冲区溢出错误检测
- 支持带地址检测功能的9位模式（第9位 = 1）
- 独立的发送和接收中断
- 用于诊断支持的环回模式
- 用于发送和接收线的极性控制
- 支持同步字符和间隔字符
- 支持自动波特率检测
- IrDA编码器和解码器逻辑
- 包含DMA支持
- 用于IrDA支持的16倍波特率时钟输出

图19-1给出了UARTx模块的简化框图。UARTx模块由以下重要硬件组成：

- 波特率发生器
- 异步发送器
- 异步接收器

> **注：** 在本章中，与特定UART模块相关的寄存器名称和位名称的引用一般都采用以“x”代替特定模块编号的方式。因此，“UxSTA”可能指UART1或UART2的状态寄存器。

图19-1： UARTx简化框图



注 1： UART1和UART2输入和输出必须在使用之前分配给可用的RPn/RPIn引脚。更多信息，请参见第11.5节“外设引脚选择（PPS）”。

19.1 UARTx波特率发生器（BRG）

UARTx模块包含一个专用的16位波特率发生器。UxBRG寄存器控制一个自由运行的16位定时器的周期。公式19-1给出了BRGH = 0时计算波特率的公式。

公式19-1： BRGH = 0时的UARTx波特率[1,2]

$$\text{波特率} = \frac{F_{CY}}{16 \cdot (UxBRG + 1)}$$

$$UxBRG = \frac{F_{CY}}{16 \cdot \text{波特率}} - 1$$

注 1： FCY表示指令周期时钟频率（FOSC/2）。
2： 基于FCY = FOSC/2；禁止打盹模式和PLL。

例19-1给出了下列条件下的波特率误差计算：

- FCY = 4 MHz
- 目标波特率 = 9600

可能的最大波特率（BRGH = 0）是FCY/16（当UxBRG = 0时），可能的最小波特率是FCY/(16 * 65536)。

公式19-2给出了BRGH = 1时计算波特率的公式。

公式19-2： BRGH = 1时的UARTx波特率[1,2]

$$\text{波特率} = \frac{F_{CY}}{4 \cdot (UxBRG + 1)}$$

$$UxBRG = \frac{F_{CY}}{4 \cdot \text{波特率}} - 1$$

注 1： FCY表示指令周期时钟频率。
2： 基于FCY = FOSC/2；禁止打盹模式和PLL。

可能的最大波特率（BRGH = 1）是FCY/4（当UxBRG = 0时），可能的最小波特率是FCY/(4 * 65536)。

向UxBRG寄存器中写入新值会导致BRG定时器复位（清零）。这保证了BRG在产生新的波特率之前不需要等待定时器溢出。

例19-1： 波特率误差计算（BRGH = 0）[1]

目标波特率 = FCY/(16 (UxBRG + 1))

求解UxBRG值：

UxBRG = ((FCY/目标波特率)/16) – 1
UxBRG = ((4000000/9600)/16) – 1
UxBRG = 25

计算波特率 = 4000000/(16 (25 + 1))
= 9615

误差 = (计算波特率 – 目标波特率)/目标波特率
= (9615 – 9600)/9600
= 0.16%

注 1： 基于FCY = FOSC/2；禁止打盹模式和PLL。

19.2 8位数据模式下的发送

1. 设置UARTx：
    a) 将适当的值写入数据位、奇偶校验位和停止位。
    b) 将适当的波特率值写入UxBRG寄存器。
    c) 设置发送和接收中断允许位和中断优先级位。
2. 使能UARTx。
3. 将UTXEN位置1（置1后两个周期内产生发送中断）。
4. 将数据字节写入UxTXREG字的低字节。该数据字节将被立即传送给发送移位寄存器（TSR）且在波特率时钟的下一个上升沿开始移出串行比特流。
5. 另外，数据字节也可在UTXEN = 0时被传送，且随后用户可将UTXEN置1。由于波特率时钟是从清零状态启动的，所以这能实现立即开始发送串行比特流。
6. 将根据中断控制位UTXISEL<1:0>的设置产生发送中断。

19.3 9位数据模式下的发送

1. 设置UARTx（如第19.2节“8位数据模式下的发送”所述）。
2. 使能UARTx。
3. 将UTXEN位置1（产生发送中断）。
4. 只将16位值写入UxTXREG。
5. 向UxTXREG写入一个字可触发9位数据向TSR传送。串行比特流将会在波特率时钟的第一个上升沿开始移出。
6. 将根据控制位UTXISELx的设置产生发送中断。

19.4 间隔和同步字符发送序列

下面的序列将发送一个由间隔字符和其后的自动波特率同步字节组成的报文帧头。

1. 将UARTx配置为所需的模式。
2. 将UTXEN和UTXBRK置1以设置间隔字符。
3. 将一个“虚拟”字符装入UxTXREG寄存器中以启动发送（该值会被忽略）。
4. 向UxTXREG写入“55h”，这会将同步字符装入发送FIFO中。
5. 当间隔字符发送完成后，由硬件将UTXBRK位复位。然后开始发送同步字符。

19.5 8位或9位数据模式下的接收

1. 设置UARTx（如第19.2节“8位数据模式下的发送”所述）。
2. 通过将URXEN位（UxSTA<12>）置1来使能UARTx。
3. 当接收到一个或多个数据字符时，将会根据中断控制位URXISEL<1:0>的设置产生接收中断。
4. 读OERR位以确定是否发生了溢出错误。必须用软件将OERR位复位。
5. 读UxRXREG。

读取UxRXREG字符的行为会将下一个字符传送到接收FIFO的顶部，其中包含一组新的PERR和FERR值。

19.6 $\overline{\text{UxCTS}}$和$\overline{\text{UxRTS}}$控制引脚的操作

UARTx允许发送（$\overline{\text{UxCTS}}$）和UARTx请求发送（$\overline{\text{UxRTS}}$）是两个与UARTx模块有关由硬件控制的引脚。这两个引脚允许UARTx运行在单工模式和流控制模式下。这两个引脚控制数据终端设备（Data Terminal Equipment，DTE）之间的数据发送和接收。通过UxMODE寄存器的UEN<1:0>位来配置这两个引脚。

19.7 红外支持

UARTx模块提供两种红外UART支持：一种是IrDA时钟输出，支持外部IrDA编码器和解码器（支持传统模块）；另一种是完全实现的IrDA编码器和解码器。注意，因为IrDA模式需要16倍的波特率时钟，所以它们只能工作在BRGH位（UxMODE<3>）为0时。

19.7.1 支持外部IrDA的IrDA时钟输出

为了支持外部IrDA编码器和解码器，可将BCLKx引脚（与$\overline{\text{UxRTS}}$引脚相同）配置为产生16倍的波特率时钟。当UEN<1:0> = 11时，如果使能UARTx模块，BCLKx引脚将输出16倍的波特率时钟；它可以用于支持IrDA编解码器芯片。

19.7.2 内置IrDA编码器和解码器

UARTx模块在其内部完全实现了IrDA编码器和解码器。内置IrDA编码器和解码器的功能可通过IREN位（UxMODE<12>）来使能。当使能（IREN = 1）时，接收引脚（UxRX）可作为红外接收器的输入引脚。发送引脚（UxTX）可作为红外发送器的输出引脚。

寄存器19-1： UxMODE：UARTx模式寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
UARTEN[1]	—	USIDL	IREN[2]	RTSMD	—	UEN1	UEN0
bit 15							bit 8

R/W-0, HC	R/W-0	R/W-0, HC	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ABAUD	URXINV	BRGH	PDSEL1	PDSEL0	STSEL
bit 7							bit 0

图注：	HC = 硬件清零位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15 **UARTEN：**UARTx使能位[1]
1 = 使能UARTx；UARTx根据UEN<1:0>的定义控制所有UARTx引脚
0 = 禁止UARTx；由端口锁存器控制所有UARTx引脚；UARTx的功耗最小

bit 14 **未实现：**读为0

bit 13 **USIDL：**UARTx空闲模式停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 在空闲模式下模块继续工作

bit 12 **IREN：**IrDA编码器和解码器使能位[2]
1 = 使能IrDA编码器和解码器
0 = 禁止IrDA编码器和解码器

bit 11 **RTSMD：**$\overline{\text{UxRTS}}$引脚模式选择位
1 = $\overline{\text{UxRTS}}$引脚处于单工模式
0 = $\overline{\text{UxRTS}}$引脚处于流控制模式

bit 10 **未实现：**读为0

bit 9-8 **UEN<1:0>：**UARTx使能位
11 = 使能并使用UxTX、UxRX和BCLKx引脚；$\overline{\text{UxCTS}}$引脚由端口锁存器控制
10 = 使能并使用UxTX、UxRX、$\overline{\text{UxCTS}}$和$\overline{\text{UxRTS}}$引脚
01 = 使能并使用UxTX、UxRX和$\overline{\text{UxRTS}}$引脚；$\overline{\text{UxCTS}}$引脚由端口锁存器控制
00 = 使能并使用UxTX和UxRX引脚；$\overline{\text{UxCTS}}$和$\overline{\text{UxRTS}}$/BCLKx引脚由端口锁存器控制

bit 7 **WAKE：**在休眠模式下检测到启动位时唤醒的使能位
1 = UARTx将继续采样UxRX引脚；在出现下降沿时产生中断；在随后的上升沿该位由硬件清零
0 = 不使能唤醒

bit 6 **LPBACK：**UARTx环回模式选择位
1 = 使能环回模式
0 = 禁止环回模式

bit 5 **ABAUD：**自动波特率使能位
1 = 使能在下一个字符传输过程中对波特率进行测量——需要接收到同步字段（55h）；完成时由硬件清零
0 = 禁止波特率测量或测量已完成

bit 4 **URXINV：**UARTx接收极性翻转位
1 = UxRX的空闲状态为0
0 = UxRX的空闲状态为1

注 1： 如果UARTEN = 1，外设的输入和输出必须配置给可用的RPn/RPIn引脚。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。
2： 该功能仅在16倍BRG模式（BRGH = 0）下可用。

寄存器19-1： UxMODE：UARTx模式寄存器（续）

bit 3 **BRGH**：高波特率使能位
1 = 高速模式（每个位周期内产生4个BRG时钟信号）
0 = 标准模式（每个位周期内产生16个BRG时钟信号）

bit 2-1 **PDSEL<1:0>**：奇偶校验和数据选择位
11 = 9位数据，无奇偶校验
10 = 8位数据，奇校验
01 = 8位数据，偶校验
00 = 8位数据，无奇偶校验

bit 0 **STSEL**：停止位选择位
1 = 2个停止位
0 = 1个停止位

注 1： 如果UARTEN = 1，外设的输入和输出必须配置给可用的RPn/RPIn引脚。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。
2： 该功能仅在16倍BRG模式（BRGH = 0）下可用。

寄存器19-2： UxSTA：UARTx状态和控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0, HC	R/W-0	R-0, HSC	R-1, HSC
UTXISEL1	UTXINV[(1)]	UTXISEL0	URXEN	UTXBRK	UTXEN[(2)]	UTXBF	TRMT
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-1, HSC	R-0, HSC	R-0, HSC	R/C-0, HS	R-0, HSC
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

图注：	C = 可清零位	HSC = 硬件置1/清零位	
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知
HS = 硬件置1位	HC = 硬件清零位		

bit 15,13 **UTXISEL<1:0>**：UARTx发送中断模式选择位
11 = 保留；不要使用
10 = 当一个字符被传送到发送移位寄存器（TSR）导致发送缓冲区为空时，产生中断
01 = 当最后一个字符被移出发送移位寄存器，所有发送操作执行完毕时产生中断
00 = 当一个字符被传输到发送移位寄存器（这意味着发送缓冲区中至少有一个字符）时产生中断

bit 14 **UTXINV**：UARTx IrDA编码器发送极性翻转位[(1)]
IREN = 0：
1 = UxTX的空闲状态为0
0 = UxTX的空闲状态为1
IREN = 1：
1 = UxTX的空闲状态为1
0 = UxTX的空闲状态为0

bit 12 **URXEN**：UARTx接收使能位
1 = 使能接收，UARTx控制UxRX引脚
0 = 禁止接收，端口控制UxRX引脚

bit 11 **UTXBRK**：UARTx发送间隔字符位
1 = 在下次发送时发送同步间隔字符——启动位，后跟12个0位，然后是停止位；完成时由硬件清零
0 = 禁止或已完成同步间隔字符的发送

bit 10 **UTXEN**：UARTx发送使能位[(2)]
1 = 使能发送，UARTx控制UxTX引脚
0 = 禁止发送，中止所有等待的发送，缓冲区复位；由端口控制UxTX引脚

bit 9 **UTXBF**：UARTx发送缓冲区满状态位（只读）
1 = 发送缓冲区已满
0 = 发送缓冲区未满，至少还可写入一个字符

bit 8 **TRMT**：发送移位寄存器空位（只读）
1 = 发送移位寄存器为空，同时发送缓冲区为空（上一次发送已完成）
0 = 发送移位寄存器非空，发送在进行中或在发送缓冲区中排队

注 1： 仅当使能了IrDA编码器（IREN = 1）时，该位的值才影响模块的发送属性。
2： 如果UARTEN = 1，外设的输入和输出必须配置给可用的RPn/RPIn引脚。更多信息，请参见第11.5节“外设引脚选择（PPS）”。

寄存器19-2： **UxSTA：UARTx状态和控制寄存器（续）**

bit 7-6 **URXISEL<1:0>**：UARTx接收中断模式选择位
11 = 当RSR传输使接收缓冲区为满时（即，有4个数据字符），中断标志位置1
10 = 当RSR传输使接收缓冲区为3/4满时（即，有3个数据字符），中断标志位置1
0x = 当接收到任一字符且将字符从RSR传输到接收缓冲区时，中断标志位置1；接收缓冲区有一个或多个字符

bit 5 **ADDEN**：地址字符检测位（接收到数据的bit 8 = 1）
1 = 使能地址检测模式（如果没有选择9位模式，这个控制位将无效）
0 = 禁止地址检测模式

bit 4 **RIDLE**：接收器空闲位（只读）
1 = 接收器空闲
0 = 接收器工作

bit 3 **PERR**：奇偶校验错误状态位（只读）
1 = 检测到当前字符（在接收FIFO顶部的字符）的奇偶校验错误
0 = 没有检测到奇偶校验错误

bit 2 **FERR**：帧错误状态位（只读）
1 = 检测到当前字符（在接收FIFO顶部的字符）的帧错误
0 = 未检测到帧错误

bit 1 **OERR**：接收缓冲区溢出错误状态位（清零/只读）
1 = 接收缓冲区已溢出
0 = 接收缓冲区没有溢出。清零先前置1的OERR位（1 → 0的跳变）会使接收缓冲区和RSR复位为空状态

bit 0 **URXDA**：UARTx接收缓冲区中有数据位（只读）
1 = 接收缓冲区中有数据，有至少一个字符可被读取
0 = 接收缓冲区为空

注 **1**： 仅当使能了IrDA编码器（IREN = 1）时，该位的值才影响模块的发送属性。
2： 如果UARTEN = 1，外设的输入和输出必须配置给可用的RPn/RPIn引脚。更多信息，请参见**第11.5节“外设引脚选择（PPS）”**。

寄存器19-3： **UxRXREG：UARTx接收寄存器（通常只读）**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0
—	—	—	—	—	—	—	UxRXREG8
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
UxRXREG<7:0>							
bit 7							bit 0

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

bit 15-9 **未实现：**读为0

bit 8-0 **UxRXREG<8:0>：**已接收字符的数据位

寄存器19-4： **UxTXREG：UARTx发送寄存器（通常只写）**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	W-x
—	—	—	—	—	—	—	UxTXREG8
bit 15							bit 8

W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
UxTXREG<7:0>							
bit 7							bit 0

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

bit 15-9 **未实现：**读为0

bit 8-0 **UxTXREG<8:0>：**已发送字符的数据位

寄存器19-5： UxBRG：UARTx波特率发生器寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRG<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRG<7:0>							
bit 7							bit 0

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

bit 15-0 **BRG<15:0>**：波特率分频比位

寄存器19-6： UxADMD：UARTx地址检测和匹配寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADMMASK7	ADMMASK6	ADMMASK5	ADMMASK4	ADMMASK3	ADMMASK2	ADMMASK1	ADMMASK0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADMADDR7	ADMADDR6	ADMADDR5	ADMADDR4	ADMADDR3	ADMADDR2	ADMADDR1	ADMADDR0
bit 7							bit 0

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

bit 15-8 **ADMMASK<7:0>**：ADMADDR<7:0>（UxADMD<7:0>）掩码位

对于ADMMASKx：
1 = 使用ADMADDRx来检测地址匹配
0 = 不使用ADMADDRx来检测地址匹配

bit 7-0 **ADMADDR<7:0>**：地址检测任务减轻负载位

与ADMMASK<7:0>位（UxADMD<15:8>）位配合使用，用于在地址检测模式期间减轻处理器检测地址字符的任务。

20.0 增强型并行主端口（EPMP）

注：	本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》的**“增强型并行主端口（EPMP）”**（DS39730），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

增强型并行主端口（EPMP）模块提供一个4位（仅限主模式）或8位（主模式和从模式）并行数据总线接口，可与存储器、FIFO和LCD控制器等片外模块以及其他单片机进行通信。此模块可充当通信总线上的主器件或从器件。

对于EPMP主模式，所有外部地址均映射到内部扩展数据空间（EDS）。这通过为每条片选分配EDS区域，然后将片选分配到特定外部资源（如存储器或外部控制器）来实现。该区域不应再被分配给其他器件资源，如RAM或SFR。要对外部资源执行读或写操作，CPU只需在为EPMP分配的地址范围内执行读或写操作。

EPMP模块的主要特性包括：

- 扩展数据空间（EDS）接口支持直接从CPU访问
- 最多10条可编程地址线
- 最多2条片选线
- 最多2条应答线（每条片选线对应一条应答线）
- 4位或8位宽数据总线
- 可编程选通选项（每条片选线）：
  - 独立的读和写选通，或；
  - 带使能选通的读/$\overline{写}$选通
- 可编程地址/数据复用
- 可编程地址等待状态
- 可编程数据等待状态（每条片选线）
- 可编程的控制信号极性（每条片选线）
- 支持传统的并行从端口
- 支持增强型并行从端口：
  - 地址支持
  - 4字节深自动递增缓冲区

该系列中只有高引脚数封装实现EPMP。28引脚器件上不提供EPMP功能。

20.1 可在不同模式下寻址的存储器

可由器件寻址的存储空间取决于地址/数据复用选择；它的大小从1K到2 MB不等。有关不同的存储器可寻址模式，请参见表20-1。

20.2 PMDOUT1和PMDOUT2寄存器

EPMP数据输出1和数据输出2寄存器仅在从模式下使用。这两个寄存器用作输出数据的缓冲区。

20.3 PMDIN1和PMDIN2寄存器

在从模式下使用EPMP数据输入1和数据输入2寄存器来缓冲输入数据。这两个寄存器保存随时钟异步输入的数据。在主模式下，PMDIN1是输入数据的保持寄存器。

表20-1： 按器件引脚数列出的EPMP特性差异

<table>
<tr><th>数据端口大小</th><th>PMA<9:8></th><th>PMA<7:0></th><th>PMD<7:4></th><th>PMD<3:0></th><th>可访问的存储器</th></tr>
<tr><td colspan="6">非复用地址（ADRMUX<1:0> = 00）</td></tr>
<tr><td>8位（PTSZ<1:0> = 00）</td><td>Addr<9:8></td><td>Addr<7:0></td><td colspan="2">数据</td><td>1K</td></tr>
<tr><td>4位（PTSZ<1:0> = 01）</td><td>Addr<9:8></td><td>Addr<7:0></td><td>—</td><td>数据</td><td>1K</td></tr>
<tr><td colspan="6">1个地址段（ADRMUX<1:0> = 01）</td></tr>
<tr><td>8位（PTSZ<1:0> = 00）</td><td>—</td><td>PMALL</td><td colspan="2">Addr<7:0>数据</td><td>1K</td></tr>
<tr><td rowspan="2">4位（PTSZ<1:0> = 01）</td><td rowspan="2">Addr<9:8></td><td rowspan="2">PMALL</td><td>Addr<7:4></td><td>Addr<3:0></td><td rowspan="2">1K</td></tr>
<tr><td>—</td><td>数据（1）</td></tr>
<tr><td colspan="6">2个地址段（ADRMUX<1:0> = 10）</td></tr>
<tr><td rowspan="3">8位（PTSZ<1:0> = 00）</td><td rowspan="3">—</td><td>PMALL</td><td colspan="2">Addr<7:0></td><td rowspan="3">64K</td></tr>
<tr><td>PMALH</td><td colspan="2">Addr<15:8></td></tr>
<tr><td>—</td><td colspan="2">数据</td></tr>
<tr><td rowspan="3">4位（PTSZ<1:0> = 01）</td><td rowspan="3">Addr<9:8></td><td>PMALL</td><td colspan="2">Addr<3:0></td><td rowspan="3">1K</td></tr>
<tr><td>PMALH</td><td colspan="2">Addr<7:4></td></tr>
<tr><td>—</td><td colspan="2">数据</td></tr>
<tr><td colspan="6">3个地址段（ADRMUX<1:0> = 11）</td></tr>
<tr><td rowspan="4">8位（PTSZ<1:0> = 00）</td><td rowspan="4">—</td><td>PMALL</td><td colspan="2">Addr<7:0></td><td rowspan="4">2 MB</td></tr>
<tr><td>PMALH</td><td colspan="2">Addr<15:8></td></tr>
<tr><td>PMALU</td><td colspan="2">Addr<22:16></td></tr>
<tr><td>—</td><td colspan="2">数据</td></tr>
<tr><td rowspan="4">4位（PTSZ<1:0> = 01）</td><td rowspan="4">Addr<13:12></td><td>PMALL</td><td colspan="2">Addr<3:0></td><td rowspan="4">16K</td></tr>
<tr><td>PMALH</td><td colspan="2">Addr<7:4></td></tr>
<tr><td>PMALU</td><td colspan="2">Addr<11:8></td></tr>
<tr><td>—</td><td colspan="2">数据</td></tr>
</table>

表**20-2**： 增强型并行主端口引脚说明

引脚名称 （备用功能）	类型	说明
PMA<22:16>	O	地址总线位<22:16>
PMA14	O	地址总线位14
	I/O	数据总线位14（带复用寻址的16位端口）
（PMCS1）	O	片选1（备用位置）
PMA<13:8>	O	地址总线位<13:8>
	I/O	数据总线位<13:8>（带复用寻址的16位端口）
PMA<7:3>	O	地址总线位<7:3>
PMA2	O	地址总线位2
（PMALU）	O	复用地址的地址锁存器最高字节选通端
PMA1	I/O	地址总线位1
（PMALH）	O	复用地址的地址锁存器高字节选通端
PMA0	I/O	地址总线位0
（PMALL）	O	复用地址的地址锁存器低字节选通端
PMD<15:8>	I/O	数据总线位<15:8>（非复用寻址）
PMD<7:4>	I/O	数据总线位<7:4>
	O	地址总线位<7:4>（带1段复用寻址的4位端口）
PMD<3:0>	I/O	数据总线位<3:0>
PMCS1	O	片选1
PMCS2	O	片选2
PMWR	I/O	写选通[1]
（PMENB）	I/O	使能信号[1]
PMRD	I/O	读选通[1]
（PMRD/$\overline{\text{PMWR}}$）	I/O	读/$\overline{\text{写}}$信号[1]
PMBE1	O	字节指示
PMBE0	O	半字节或字节指示
PMACK1	I	应答信号1
PMACK2	I	应答信号2

注 **1**： 信号功能取决于MODE<1:0>和SM位（PMCON1<9:8>和PMCSxCF<8>）的设置。

寄存器20-1：　PMCON1：EPMP控制寄存器1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	—	MODE1	MODE0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CSF1	CSF0	ALP	ALMODE	—	BUSKEEP	IRQM1	IRQM0
bit 7							bit 0

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

bit 15　**PMPEN：**并行主端口使能位
1 = 使能EPMP
0 = 禁止EPMP

bit 14　**未实现：**读为0

bit 13　**PSIDL：**并行主端口空闲模式停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 在空闲模式下模块继续工作

bit 12-11　**ADRMUX<1:0>：**地址/数据复用选择位
11 = 低地址位使用3个地址段与数据位复用
10 = 低地址位使用2个地址段与数据位复用
01 = 低地址位使用1个地址段与数据位复用
00 = 地址和数据使用独立的引脚

bit 10　**未实现：**读为0

bit 9-8　**MODE<1:0>：**并行端口模式选择位
11 = 主模式
10 = 增强型PSP；使用的引脚为PMRD、PMWR、PMCS、PMD<7:0>和PMA<1:0>
01 = 缓冲的PSP；使用的引脚为PMRD、PMWR、PMCS和PMD<7:0>
00 = 传统并行从端口；使用的引脚为PMRD、PMWR、PMCS和PMD<7:0>

bit 7-6　**CSF<1:0>：**片选功能位
11 = 保留
10 = PMA14用于片选1
01 = 保留
00 = PMCS2用于片选2，PMCS1用于片选1

bit 5　**ALP：**地址锁存器极性位
1 = 高电平有效（PMALL、PMALH和PMALU）
0 = 低电平有效（$\overline{\text{PMALL}}$、$\overline{\text{PMALH}}$和$\overline{\text{PMALU}}$）

bit 4　**ALMODE：**地址锁存器选通模式位
1 = 使能“智能”地址选通（仅在当前访问将导致锁存地址不同于先前地址时，才存在各地址段）
0 = 禁止“智能”地址选通

bit 3　**未实现：**读为0

bit 2　**BUSKEEP：**总线保持位
1 = 当未驱动数据总线时，保持其最后的值
0 = 当未驱动数据总线时，其处于高阻态

bit 1-0　**IRQM<1:0>：**中断请求模式位
11 = 当读取读缓冲区3或写入写缓冲区3时产生中断（缓冲的PSP模式），或当PMA<1:0> = 11时执行读或写操作时产生中断（仅限可寻址PSP模式）
10 = 保留
01 = 在读/写周期结束时产生中断
00 = 不产生中断

寄存器20-2：　PMCON2：EPMP控制寄存器2

R-0, HSC	U-0	R/C-0, HS	R/C-0, HS	U-0	U-0	U-0	U-0
BUSY	—	ERROR	TIMEOUT	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RADDR23[1]	RADDR22[1]	RADDR21[1]	RADDR20[1]	RADDR19[1]	RADDR18[1]	RADDR17[1]	RADDR16[1]
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知
C = 可清零位	HS = 硬件置1位	HSC = 硬件置1/清零位	

bit 15　**BUSY：**忙位（仅用于主模式）
　1 = 端口忙
　0 = 端口不忙

bit 14　**未实现：**读为0

bit 13　**ERROR：**错误位
　1 = 事务错误（请求非法事务）
　0 = 事务成功完成

bit 12　**TIMEOUT：**超时位
　1 = 事务超时
　0 = 事务成功完成

bit 11-8　**未实现：**读为0

bit 7-0　**RADDR<23:16>：**并行主端口保留地址空间位[1]

注　**1：**　如果RADDR<23:16> = 00000000，则片选2的最后一个EDS地址将为FFFFFFh。

寄存器20-3： PMCON3：EPMP控制寄存器3

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
PTWREN	PTRDEN	PTBE1EN	PTBE0EN	—	AWAITM1	AWAITM0	AWAITE
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15 **PTWREN**：写/使能选通端口使能位
1 = 使能PMWR/PMENB端口
0 = 禁止PMWR/PMENB端口

bit 14 **PTRDEN**：读/写选通端口使能位
1 = 使能PMRD/$\overline{\text{PMWR}}$端口
0 = 禁止PMRD/$\overline{\text{PMWR}}$端口

bit 13 **PTBE1EN**：高半字节/字节使能端口使能位
1 = 使能PMBE1端口
0 = 禁止PMBE1端口

bit 12 **PTBE0EN**：低半字节/字节使能端口使能位
1 = 使能PMBE0端口
0 = 禁止PMBE0端口

bit 11 **未实现**：读为0

bit 10-9 **AWAITM<1:0>**：地址锁存器选通等待状态位
11 = 等待3½个TCY
10 = 等待2½个TCY
01 = 等待1½个TCY
00 = 等待½个TCY

bit 8 **AWAITE**：地址锁存器选通后地址保持等待状态位
1 = 等待1¼个TCY
0 = 等待¼个TCY

bit 7-0 **未实现**：读为0

寄存器20-4： PMCON4：EPMP控制寄存器4

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	PTEN14	PTEN<13:8>					
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN<7:3>					PTEN<2:0>		
bit 7							bit 0

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

bit 15 **未实现：**读为0

bit 14 **PTEN14：**PMA14端口使能位
1 = PMA14用作地址线14或片选1
0 = PMA14用作端口I/O

bit 13-3 **PTEN<13:3>：**EPMP地址端口使能位
1 = PMA<13:3>用作EPMP地址线
0 = PMA<13:3>用作端口I/O

bit 2-0 **PTEN<2:0>：**PMALU/PMALH/PMALL选通使能位
1 = PMA<2:0>用作地址线或地址锁存器选通
0 = PMA<2:0>用作端口I/O

寄存器20-5： PMCSxCF：EPMP片选x配置寄存器

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CSDIS	CSP	CSPTEN	BEP	—	WRSP	RDSP	SM
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
ACKP	PTSZ1	PTSZ0	—	—	—	—	—
bit 7							bit 0

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

bit 15 **CSDIS**：片选x禁止位
1 = 禁止片选x功能
0 = 使能片选x功能

bit 14 **CSP**：片选x极性位
1 = 高电平有效（PMCSx）
0 = 低电平有效（$\overline{\text{PMCSx}}$）

bit 13 **CSPTEN**：PMCSx端口使能位
1 = 使能PMCSx端口
0 = 禁止PMCSx端口

bit 12 **BEP**：片选x半字节/字节使能极性位
1 = 半字节/字节使能高电平有效（PMBE0和PMBE1）
0 = 半字节/字节使能低电平有效（$\overline{\text{PMBE0}}$和$\overline{\text{PMBE1}}$）

bit 11 **未实现**：读为0

bit 10 **WRSP**：片选x写选通极性位
对于从模式和主模式（SM = 0）：
1 = 写选通高电平有效（PMWR）
0 = 写选通低电平有效（$\overline{\text{PMWR}}$）
对于主模式（SM = 1）：
1 = 使能选通高电平有效（PMENB）
0 = 使能选通低电平有效（$\overline{\text{PMENB}}$）

bit 9 **RDSP**：片选x读选通极性位
对于从模式和主模式（SM = 0）：
1 = 读选通高电平有效（PMRD）
0 = 读选通低电平有效（$\overline{\text{PMRD}}$）
对于主模式（SM = 1）：
1 = 读/写选通高电平有效（PMRD/$\overline{\text{PMWR}}$）
0 = 读/写选通低电平有效（$\overline{\text{PMRD}}$/PMWR）

bit 8 **SM**：片选x选通模式位
1 = 读/写和使能选通（PMRD/$\overline{\text{PMWR}}$和PMENB）
0 = 读和写选通（PMRD和PMWR）

bit 7 **ACKP**：片选x应答极性位
1 = ACK高电平有效（PMACK1）
0 = ACK低电平有效（$\overline{\text{PMACK1}}$）

bit 6-5 **PTSZ<1:0>**：片选x端口大小位
11 = 保留
10 = 保留
01 = 4位端口大小（PMD<3:0>）
00 = 8位端口大小（PMD<7:0>）

bit 4-0 **未实现**：读为0

寄存器20-6： PMCSxBS：EPMP片选x基地址寄存器[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]
BASE<23:16>							
bit 15							bit 8

R/W[1]	U-0	U-0	U-0	R/W[1]	U-0	U-0	U-0
BASE15	—	—	—	BASE11	—	—	—
bit 7							bit 0

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

bit 15-7 **BASE<23:15>**：片选x基地址位[1]

bit 6-4 **未实现**：读为0

bit 3 **BASE11**：片选x基地址位[1]

bit 2-0 **未实现**：读为0

注 **1**： POR时，PMCS1BS的值为0080h，PMCS2BS的值为8080h。

2： 如果对整个PMCS2BS寄存器写入0x0000，则片选1的最后一个EDS地址将为FFFFFFh。在这种情况下，不应使用片选2。PMCS1BS不具有此类功能。

寄存器20-7： PMCSxMD：EPMP片选x模式寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
ACKM1	ACKM0	AMWAIT2	AMWAIT1	AMWAIT0	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DWAITB1	DWAITB0	DWAITM3	DWAITM2	DWAITM1	DWAITM0	DWAITE1	DWAITE0
bit 7							bit 0

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

bit 15-14 **ACKM<1:0>**：片选x应答模式位
11 = 保留
10 = PMACKx用于确定何时完成读/写操作
01 = PMACKx用于确定何时完成读/写操作且超时
（如果DWAITM<3:0> = 0000，则最大超时为255个TCY，否则为DWAITM<3:0>个周期）。
00 = 未使用PMACKx

bit 13-11 **AMWAIT<2:0>**：片选x备用主等待状态位
111 = 等待10个备用主周期
...
001 = 等待4个备用主周期
000 = 等待3个备用主周期

bit 10-8 **未实现**：读为0

bit 7-6 **DWAITB<1:0>**：读/写选通前片选x数据建立等待状态位
11 = 等待3¼个TCY
10 = 等待2¼个TCY
01 = 等待1¼个TCY
00 = 等待¼个TCY

bit 5-2 **DWAITM<3:0>**：片选x数据读/写选通等待状态位
<u>对于写操作：</u>
1111 = 等待15½个TCY
...
0001 = 等待1½个TCY
0000 = 等待½个TCY
<u>对于读操作：</u>
1111 = 等待15¾个TCY
...
0001 = 等待1¾个TCY
0000 = 等待¾个TCY

bit 1-0 **DWAITE<1:0>**：读/写选通后片选x数据保持等待状态位
<u>对于写操作：</u>
11 = 等待3¼个TCY
10 = 等待2¼个TCY
01 = 等待1¼个TCY
00 = 等待¼个TCY
<u>对于读操作：</u>
11 = 等待3个TCY
10 = 等待2个TCY
01 = 等待1个TCY
00 = 等待0个TCY

寄存器20-8： PMSTAT：EPMP状态寄存器（仅从模式）

R-0, HSC	R/W-0, HS	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IBF	IBOV	—	—	IB3F[1]	IB2F[1]	IB1F[1]	IB0F[1]
bit 15							bit 8

R-1, HSC	R/W-0, HS	U-0	U-0	R-1, HSC	R-1, HSC	R-1, HSC	R-1, HSC
OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E
bit 7							bit 0

图注：	HS = 硬件置1位	HSC = 硬件置1/清零位	
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15 **IBF**：输入缓冲区满状态位
1 = 所有可写的输入缓冲寄存器已满
0 = 部分或所有可写的输入缓冲寄存器为空

bit 14 **IBOV**：输入缓冲区上溢状态位
1 = 尝试对已满的输入寄存器执行写操作（必须由软件清零）
0 = 未发生上溢

bit 13-12 **未实现**：读为0

bit 11-8 **IB3F:IB0F**：输入缓冲区x满状态位[1]
1 = 输入缓冲区包含尚未读取的数据（读缓冲区将清零此位）
0 = 输入缓冲区不包含任何未读取的数据

bit 7 **OBE**：输出缓冲区空状态位
1 = 所有可读的输出缓冲寄存器均为空
0 = 部分或所有可读的输出缓冲寄存器已满

bit 6 **OBUF**：输出缓冲区下溢状态位
1 = 对空的输出缓冲寄存器执行了读操作（必须由软件清零）
0 = 未发生下溢

bit 5-4 **未实现**：读为0

bit 3-0 **OB3E:OB0E**：输出缓冲区x状态空位
1 = 输出缓冲区x为空（将数据写入该缓冲区会将该位清零）
0 = 输出缓冲区x包含未发送的数据

注 1： 即使某个位表示缓冲区中的字节，在进行字节读取时与字（字节0和1或字节2和3）对应的位也被清零。

寄存器20-9： **PADCON：焊盘配置控制寄存器**

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
IOCON	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	PMPTTL
bit 7							bit 0

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

bit 15 **IOCON：**用于非PMP功能

bit 14-1 **未实现：**读为0

bit 0 **PMPTTL：**EPMP模块TTL输入缓冲器选择位
1 = EPMP模块输入（PMDx和PMCS1）使用TTL输入缓冲器
0 = EPMP模块输入使用施密特触发器输入缓冲器

21.0 带时间戳功能的实时时钟和日历（RTCC）

> **注：** 本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解实时时钟和日历的更多信息，请参见《dsPIC33/PIC24系列参考手册》的 **“带时间戳的RTCC”**（DS70005193），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

RTCC为用户提供了可校准的实时时钟和日历（RTCC）功能。

RTCC模块的主要特性包括：

- 可选择的时钟源
- 采用24小时格式提供小时、分和秒
- 可看到一个半秒周期
- 提供日历——星期、日期、月和年
- 可配置闹钟每半秒、1秒、10秒、1分、10分、1小时、1天、1周、1月或1年响一次
- 闹钟使用递减计数器进行重复
- 闹钟具有无限重复响铃
- 年份2000至2099闰年纠正
- 采用BCD格式，以减小软件开销
- 为长期电池工作进行了优化
- 32.768 kHz时钟晶振的用户校准/带周期性自动调整的32 kHz INTRC频率
- 小数秒同步
- 校准为每月误差在±2.64秒内
- 校准为最高260 ppm的晶振误差
- 能够定期唤醒外部器件，而无需 CPU 干预（外部电源控制）
- 针对外部电路控制的电源控制输出
- 校准每15秒生效一次
- 用于时间和日期的时间戳捕捉寄存器
- 可编程的预分频器和时钟分频器电路允许使用任何时钟源（最高32 MHz）进行操作，包括32.768 kHz晶振、50/60 Hz电源线时钟、外部实时时钟（RTC）或31.25 kHz LPRC时钟

21.1 RTCC源时钟

RTCC时钟分频器模块将输入的振荡器源转换为精确的1/2和1秒时钟，供RTCC使用。时钟分频器优化为使用3个不同的振荡器源工作：

- 32.768 kHz晶振
- 31 kHz低功耗RC振荡器（LPRC）
- 外部50 Hz或60 Hz电源线频率

模块提供了一个异步预分频器PS<1:0>（RTCCON2L<5:4>），它使RTCC可以使用速度更高的时钟源工作，例如系统时钟。可以选择1:16、1:64或1:256的分频比，允许最高32 MHz的时钟源为RTCC提供时钟。

21.1.1 粗粒度分频

时钟分频器模块具有一个用于对输入时钟频率进行分频的16位计数器。分频比由DIV<15:0>寄存器位（RTCCON2H<15:0>）设置。DIV<15:0>位的值应设定为可产生1/2秒的标称时钟分频器计数周期。

21.1.2 细粒度分频

细粒度分频使用FDIV<4:0>（RTCCON2L<15:11>）位进行设置。提高FDIVx值会延长总的时钟分频器周期。

如果FDIV<4:0> = 00000，细粒度分频电路实际上会被禁止。否则，它会每1/2秒选择性地从时钟分频器输入中去除一个时钟脉冲。该功能允许用户在16秒的固定周期内最多去除31个脉冲，具体取决于FDIVx的值。

公式21-1给出了DIV<15:0>值的计算公式。DIV<15:0>计算结果的小数余数可以用于计算FDIV<4:0>的值。

公式21-1： RTCC时钟分频器输出频率

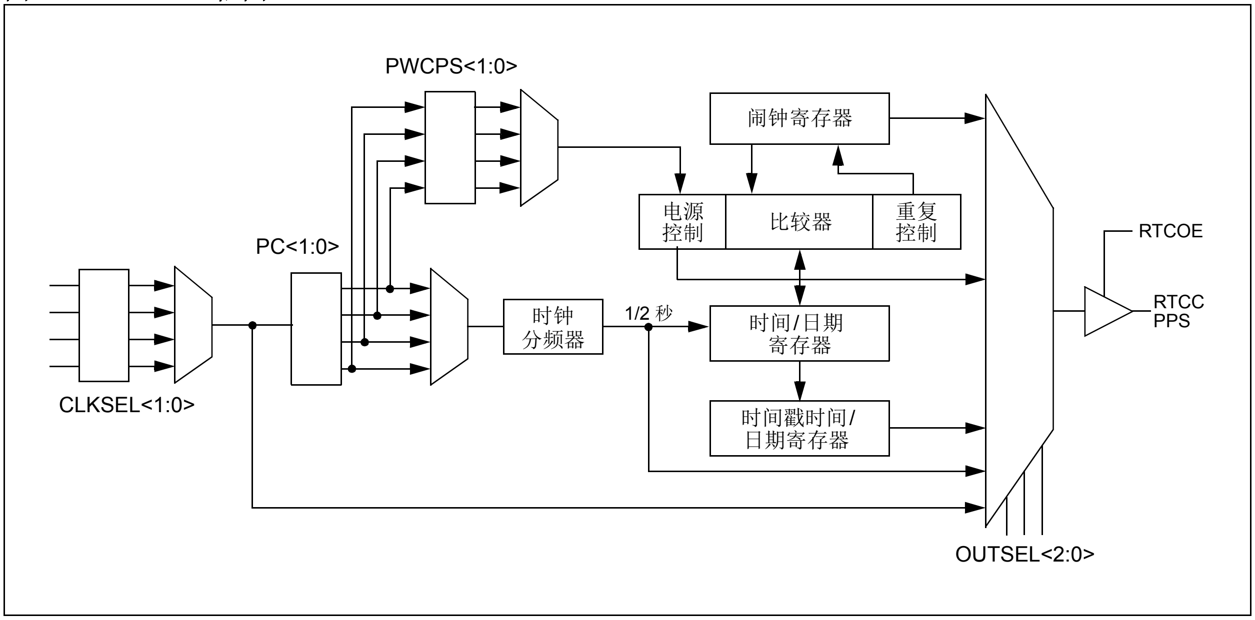
$$F_{OUT} = \frac{F_{IN}}{2 \bullet (PS\text{<}1:0\text{>}\ 预分频比) \bullet (DIV\text{<}15:0\text{>} + 1) + \left(\frac{FDIV\text{<}4:0\text{>}}{32}\right)}$$

DIV<15:0> 值是该计算结果的整数部分：

$$DIV\text{<}15:0\text{>} = \frac{F_{IN}}{2 \bullet (PS\text{<}1:0\text{>}\ 预分频比)} - 1$$

FDIV<4:0>值是DIV<15:0>计算结果的小数部分乘以32。

图21-1： RTCC框图


PWCPS<1:0>
闹钟寄存器
电源
控制
比较器
重复
控制
RTCOE
PC<1:0>
RTCC
PPS
时钟
分频器
1/2 秒
时间/日期
寄存器
CLKSEL<1:0>
时间戳时间/
日期寄存器
OUTSEL<2:0>

21.2 RTCC模块寄存器

RTCC模块寄存器可分为以下四类：

- RTCC控制寄存器
- RTCC值寄存器
- 闹钟值寄存器
- 时间戳寄存器

21.2.1 寄存器映射

以前的RTCC实现使用寄存器指针访问RTCC时间和日期寄存器，以及闹钟时间和日期寄存器。这些寄存器现在映射到存储器，并且可单独寻址。

21.2.2 写锁定

为避免意外更改时间控制或时间值寄存器，WRLOCK位（RTCCON1L1<11>）必须清零（0）。当 WRLOCK位为0时，POR处于默认状态，并会在任何器件复位（POR、BOR和$\overline{\text{MCLR}}$）时清零。建议在日期和时间寄存器正确初始化且RTCEN位（RTCCON1L<15>）置1后将WRLOCK位置1。

只要WRLOCK为1，则尝试对RTCEN位、RTCCON2L/H寄存器或者日期或时间寄存器执行的任何写操作都将被忽略。当WRLOCK为1时，可以更改报警、电源控制和时间戳寄存器。

将WRLOCK位清零时，在向其写入1之后需要一个解锁序列，将两个字节连续写入NVMKEY寄存器。例21-1给出了汇编序列示例。如果WRLOCK已清零，则可以将其置1，而无需使用解锁序列。

> **注：** 为避免意外写入定时器，建议在任何其他时候都保持WRLOCK位（RTCCON1L<11>）清零。对于将WRLOCK位置1的操作，由于在55h/AA序列和WRLOCK置1之间仅允许有一个指令周期的时间；因此，建议按照例21-1中的过程编写代码。

21.2.3 选择RTCC时钟源

可使用RTCCON2L寄存器中的CLKSEL<1:0>位选择RTCC模块的时钟源。这些位设置为00时，辅助振荡器（SOSC）用作参考时钟；这些位为01时，LPRC用作参考时钟。当CLKSEL<1:0> = 10时，外部电源线（50 Hz和60 Hz）用作时钟源。当CLKSEL<1:0> = 11时，系统时钟用作时钟源。

例21-1： 将WRLOCK位置1

```
DISI    #6                      ;disable interrupts for 6 instructions
MOV     #NVKEY, W1
MOV     #0x55, W2                ; first unlock code
MOV     W2, [W1]                 ; write first unlock code
MOV     #0xAA, W3                ; second unlock sequence
MOV     W3, [W1]                 ; write second unlock sequence
BCLR    RTCCON1L, #WRLOCK        ; clear the WRLOCK bit
```

21.3 寄存器

21.3.1 RTCC控制寄存器

寄存器21-1： RTCCON1L：RTCC控制寄存器1（低位字）

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
RTCEN	—	—	—	WRLOCK	PWCEN	PWCPOL	PWCPOE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0
RTCOE	OUTSEL2	OUTSEL1	OUTSEL0	—	—	—	TSAEN
bit 7							bit 0

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

bit 15 **RTCEN：**RTCC使能位
1 = 使能RTCC，并基于选定时钟源进行计数
0 = 不使能RTCC

bit 14-12 **未实现：**读为0

bit 11 **WRLOCK：**RTCC寄存器写锁定
1 = 锁定RTCC寄存器
0 = RTCC可以由用户写入

bit 10 **PWCEN：**电源控制使能位
1 = 使能电源控制
0 = 禁止电源控制

bit 9 **PWCPOL：**电源控制极性位
1 = 电源控制输出为高电平有效
0 = 电源控制输出为低电平有效

bit 8 **PWCPOE：**电源控制输出使能位
1 = 使能电源控制输出引脚
0 = 禁止电源控制输出引脚

bit 7 **RTCOE：**RTCC输出使能位
1 = 使能RTCC输出
0 = 禁止RTCC输出

bit 6-4 **OUTSEL<2:0>：**RTCC输出信号选择位
111 = 未使用
110 = 未使用
101 = 未使用
100 = 时间戳A事件
011 = 电源控制
010 = RTCC输入时钟
001 = 秒时钟
000 = 闹钟事件

bit 3-1 **未实现：**读为0

bit 0 **TSAEN：**时间戳A使能位
1 = 在$\overline{\text{TMPRN}}$引脚上检测到低电平脉冲时产生时间戳事件
0 = 禁止时间戳

寄存器21-2： RTCCON1H：RTCC控制寄存器1（高位字）

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ALRMEN	CHIME	—	—	AMASK3	AMASK2	AMASK1	AMASK0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ALMRPT7	ALMRPT6	ALMRPT5	ALMRPT4	ALMRPT3	ALMRPT2	ALMRPT1	ALMRPT0
bit 7							bit 0

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

bit 15 **ALRMEN：**闹钟使能位
1 = 使能闹钟（只要ALMRPT<7:0> = 00h且CHIME = 0，就会在闹钟事件后自动清零）
0 = 禁止闹钟

bit 14 **CHIME：**响铃使能位
1 = 使能响铃；ALMRPT<7:0>位从00h返回到FFh
0 = 禁止响铃；ALMRPT<7:0>位到达00h就停止

bit 13-12 **未实现：**读为0

bit 11-8 **AMASK<3:0>：**闹钟掩码配置位
0000 = 每半秒
0000 = 每秒
0010 = 每10秒
0011 = 每分钟
0100 = 每10分钟
0101 = 每小时
0110 = 一天一次
0111 = 一周一次
1000 = 一月一次
1001 = 每年一次（除配置为2月29日每4年一次外）
101x = 保留——不要使用
11xx = 保留——不要使用

bit 7-0 **ALMRPT<7:0>：**闹钟重复计数器值位
11111111 = 闹钟将重复255次
•
•
•
00000000 = 闹钟将重复0次
每发生一次闹钟事件，计数器就递减1。只有在CHIME = 1的情况下，计数器才会在递减到00后返回FF。

寄存器21-3： RTCCON2L：RTCC控制寄存器2（低位字）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
FDIV4	FDIV3	FDIV2	FDIV1	FDIV0	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
PWCPS1	PWCPS0	PS1	PS0	—	—	CLKSEL1	CLKSEL0
bit 7							bit 0

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

bit 15-11 **FDIV<4:0>**：小数时钟分频比位
00000 = 没有小数时钟分频
00001 = 每16秒将周期增加1个RTCC输入时钟周期
00010 = 每16秒将周期增加2个RTCC输入时钟周期
•
•
•
11101 = 每16秒将周期增加30个RTCC输入时钟周期
11111 = 每16秒将周期增加31个RTCC输入时钟周期

bit 10-8 **未实现**：读为0

bit 7-6 **PWCPS<1:0>**：电源控制预分频比选择位
00 = 1:1
01 = 1:16
10 = 1:64
11 = 1:256

bit 5-4 **PS<1:0>**：预分频比选择位
00 = 1:1
01 = 1:16
10 = 1:64
11 = 1:256

bit 3-2 **未实现**：读为0

bit 1-0 **CLKSEL<1:0>**：时钟选择位
00 = SOSC
01 = LPRC
10 = PWRLCLK引脚
11 = 系统时钟

21.3.2 RTCVAL寄存器映射

寄存器21-4：　RTCCON2H：RTCC控制寄存器2（高位字）[1]

R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
DIV<15:8>							
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
DIV<7:0>							
bit 7							bit 0

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

bit 15-0　**DIV<15:0>**：时钟分频比位
设置时钟分频器计数器的周期；值应引起标称1/2秒下溢。

注 **1**：　仅当WRLOCK = 1时才允许写入该寄存器。

寄存器21-5： RTCCON3L：RTCC控制寄存器3（低位字）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PWCSAMP7	PWCSAMP6	PWCSAMP5	PWCSAMP4	PWCSAMP3	PWCSAMP2	PWCSAMP1	PWCSAMP0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PWCSTAB7	PWCSTAB6	PWCSTAB5	PWCSTAB4	PWCSTAB3	PWCSTAB2	PWCSTAB1	PWCSTAB0
bit 7							bit 0

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

bit 15-8 **PWCSAMP<7:0>**：电源控制采样窗口定时器位
11111111 = 即使PWCEN = 0时，仍始终使能采样窗口
11111110 = 采样窗口为254个TPWCCLK时钟周期
•
•
•
00000001 = 采样窗口为1个TPWCCLK时钟周期
00000000 = 无采样窗口

bit 7-0 **PWCSTAB<7:0>**：电源控制稳定性窗口定时器位[1]
11111111 = 稳定性窗口为255个TPWCCLK时钟周期
11111110 = 稳定性窗口为254个TPWCCLK时钟周期
•
•
•
00000001 = 稳定性窗口为1个TPWCCLK时钟周期
00000000 = 无稳定性窗口；采样窗口在闹钟事件触发时启动

注 1： 当稳定性窗口定时器超时后，始终会启动采样窗口，除非其初始值为00h。

寄存器21-6： RTCSTATL：RTCC状态寄存器（低位字）

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/C-0	U-0	R/C-0	R-0	R-0	R-0
—	—	ALMEVT	—	TSAEVT[(1)]	SYNC	ALMSYNC	HALFSEC[(2)]
bit 7							bit 0

图注：	C = 可清零位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15-6 **未实现：**读为0

bit 5 **ALMEVT：**闹钟事件位
1 = 发生了闹钟事件
0 = 未发生闹钟事件

bit 4 **未实现：**读为0

bit 3 **TSAEVT：**时间戳A事件位[(1)]
1 = 发生了时间戳事件
0 = 未发生时间戳事件

bit 2 **SYNC：**同步状态位
1 = 可以在软件读取期间更改TIME寄存器
0 = 可以安全读取TIME寄存器

bit 1 **ALMSYNC：**闹钟同步状态位
1 = 不得修改闹钟寄存器（ALMTIME 和 ALMDATE）和闹钟位（AMASK<3:0>），可在软件读取期间更改闹钟控制位（ALRMEN，ALMRPT<7:0>）
0 = 可安全地写入/修改闹钟寄存器和闹钟控制位

bit 0 **HALFSEC：**半秒状态位[(2)]
1 = 一秒的后半周期
0 = 一秒的前半周期

注 1： 用户软件可向该位写入1以启动时间戳A事件；直到TSAEVT读为1，时间戳捕捉才有效。
2： 该位是只读位。当写入SECONE<3:0>位时，该位清零。

21.3.3 RTCC值寄存器

寄存器21-7： TIMEL：RTCC时间寄存器（低位字）

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15	**未实现：**读为0
bit 14-12	**SECTEN<2:0>：**秒的十位数字的BCD值位 包含从0到5的值。
bit 11-8	**SECONE<3:0>：**秒的个位数字的BCD值位 包含从0到9的值。
bit 7-0	**未实现：**读为0

寄存器21-8： TIMEH：RTCC时间寄存器（高位字）

U-0	U-0	R/W-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 7							bit 0

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

bit 15-14	**未实现：**读为0
bit 13-12	**HRTEN<1:0>：**小时的十位数字的BCD值位 包含从0到2的值。
bit 11-8	**HRONE<3:0>：**小时的个位数字的BCD值位 包含从0到9的值。
bit 7	**未实现：**读为0
bit 6-4	**MINTEN<2:0>：**分钟的十位数字的BCD值位 包含从0到5的值。
bit 3-0	**MINONE<3:0>：**分钟的个位数字的BCD值位 包含从0到9的值。

寄存器21-9： DATEL：RTCC日期寄存器（低位字）

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 7							bit 0

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

bit 15-14 未实现：读为0

bit 13-12 **DAYTEN<1:0>**：日的十位数字的BCD值位
包含从0到3的值。

bit 11-8 **DAYONE<3:0>**：日的个位数字的BCD值位
包含从0到9的值。

bit 7-3 未实现：读为0

bit 2-0 **WDAY<2:0>**：星期的个位数字的BCD值位
包含从0到6的值。

寄存器21-10： DATEH：RTCC日期寄存器（高位字）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-x	R/W-x	R/W-x	R/W-x
YRTEN3	YRTEN2	YRTEN1	YRTEN0	YRONE3	YRONE2	YRONE1	YRONE0
bit 15							bit 8

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 7							bit 0

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

bit 15-12 **YRTEN<3:0>**：年的十位数字的BCD值位

bit 11-8 **YRONE<3:0>**：年的个位数字的BCD值位

bit 7-5 未实现：读为0

bit 4 **MTHTEN**：月的十位数字的BCD值位
包含从0到1的值。

bit 3-0 **MTHONE<3:0>**：月的个位数字的BCD值位
包含从0到9的值。

21.3.4 闹钟值寄存器

寄存器21-11： ALMTIMEL：RTCC闹钟时间寄存器（低位字）

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15 未实现：读为0

bit 14-12 **SECTEN<2:0>**：秒的十位数字的BCD值位
包含从0到5的值。

bit 11-8 **SECONE<3:0>**：秒的个位数字的BCD值位
包含从0到9的值。

bit 7-0 未实现：读为0

寄存器21-12： ALMTIMEH：RTCC闹钟时间寄存器（高位字）

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 7							bit 0

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

bit 15-14 未实现：读为0

bit 13-12 **HRTEN<1:0>**：小时的十位数字的BCD值位
包含从0到2的值。

bit 11-8 **HRONE<3:0>**：小时的个位数字的BCD值位
包含从0到9的值。

bit 7 未实现：读为0

bit 6-4 **MINTEN<2:0>**：分钟的十位数字的BCD值位
包含从0到5的值。

bit 3-0 **MINONE<3:0>**：分钟的个位数字的BCD值位
包含从0到9的值。

寄存器21-13： ALMDATEL：RTCC闹钟日期寄存器（低位字）

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-12 **DAYTEN<1:0>：**日的十位数字的BCD值位
包含从0到3的值。

bit 11-8 **DAYONE<3:0>：**日的个位数字的BCD值位
包含从0到9的值。

bit 7-3 **未实现：**读为0

bit 2-0 **WDAY<2:0>：**星期的个位数字的BCD值位
包含从0到6的值。

寄存器21-14： ALMDATEH：RTCC闹钟日期寄存器（高位字）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
YRTEN3	YRTEN2	YRTEN1	YRTEN0	YRONE3	YRONE2	YRONE1	YRONE0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	MTHTEN	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 7							bit 0

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

bit 15-12 **YRTEN<3:0>：**年的十位数字的BCD值位

bit 11-8 **YRONE<3:0>：**年的个位数字的BCD值位

bit 7-5 **未实现：**读为0

bit 4 **MTHTEN：**月的十位数字的BCD值位
包含从0到1的值。

bit 3-0 **MTHONE<3:0>：**月的个位数字的BCD值位
包含从0到9的值。

21.3.5 时间戳寄存器

寄存器21-15： TSATIMEL：RTCC时间戳A时间寄存器（低位字）(1)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15 **未实现：**读为0

bit 14-12 **SECTEN<2:0>：**秒的十位数字的BCD值位
包含从0到5的值。

bit 11-8 **SECONE<3:0>：**秒的个位数字的BCD值位
包含从0到9的值。

bit 7-0 **未实现：**读为0

注 1： 如果TSAEN = 0，bit<15:0>可用于非上电复位（$\overline{\text{MCLR}}$和WDT等）过程中的持久存储。

寄存器21-16： TSATIMEH：RTCC时间戳A时间寄存器（高位字）[1]

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-12 **HRTEN<1:0>：**小时的十位数字的BCD值位
包含从0到2的值。

bit 11-8 **HRONE<3:0>：**小时的个位数字的BCD值位
包含从0到9的值。

bit 7 **未实现：**读为0

bit 6-4 **MINTEN<2:0>：**分钟的十位数字的BCD值位
包含从0到5的值。

bit 3-0 **MINONE<3:0>：**分钟的个位数字的BCD值位
包含从0到9的值。

注 1： 如果TSAEN = 0，bit<15:0>可用于非上电复位（$\overline{\text{MCLR}}$和WDT等）过程中的持久存储。

寄存器21-17： TSADATEL：RTCC时间戳A日期寄存器（低位字）[1]

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-12 **DAYTEN<1:0>：**日的十位数字的BCD值位
包含从0到3的值。

bit 11-8 **DAYONE<3:0>：**日的个位数字的BCD值位
包含从0到9的值。

bit 7-3 **未实现：**读为0

bit 2-0 **WDAY<2:0>：**星期的个位数字的BCD值位
包含从0到6的值。

注 1： 如果TSAEN = 0，bit<15:0>可用于非上电复位（$\overline{\text{MCLR}}$和WDT等）过程中的持久存储。

寄存器21-18： **TSADATEH：RTCC时间戳A日期寄存器（高位字）**[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
YRTEN3	YRTEN2	YRTEN1	YRTEN0	YRONE3	YRONE2	YRONE1	YRONE0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	MTHTEN	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 7							bit 0

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

bit 15-12 **YRTEN<3:0>**：年的十位数字的BCD值位

bit 11-8 **YRONE<3:0>**：年的个位数字的BCD值位

bit 7-5 **未实现**：读为0

bit 4 **MTHTEN**：月的十位数字的BCD值位
包含从0到1的值。

bit 3-0 **MTHONE<2:0>**：月的个位数字的BCD值位
包含从0到9的值。

注 1： 如果TSAEN = 0，bit<15:0>可用于非上电复位（$\overline{\text{MCLR}}$和WDT等）过程中的持久存储。

21.4 校准

21.4.1 时钟源校准

可以校准连接到RTCC的晶振，以便以两种方式提供精确的1秒时钟。首先，通过调整写入DIV<15:0>位的值来对频率进行粗调。然后，可向FDIV<4:0>控制位写入5位值以执行精细时钟分频。

DIVx和FDIVx值组合在一起可形成21位预分频值。如果振荡器源比理想速度略快，则可以增大FDIV<4:0>值，以使RTC频率略有下降。若要使RTC频率显著下降，应增大DIV<15:0>的值。如果振荡器源慢于理想速度，则可通过减小FDIV<4:0>实现微小的校准变化，而较大的校准变化可能需要通过减小DIV<15:0>来实现。

在校准前，用户必须确定晶体的误差。这应使用器件上的其他定时器资源或外部时序参考来完成。用户决定误差值是否包含晶体初始误差（由温度和晶体老化所引起的漂移）。

21.5 闹钟

- 闹钟的时间间隔可配置为从半秒到一年
- 使用ALRMEN位（RTCCON1H<15>）使能
- 可选择一次性闹钟和重复闹钟

21.5.1 配置闹钟

使用ALRMEN位使能闹钟功能。闹钟事件发生后该位清零。只有在ALRMEN = 0时才允许写入闹钟值寄存器。

如图21-2所示，可通过AMASK<3:0>位（RTCCON1H<11:8>）配置闹钟时间间隔。这些位确定要触发闹钟，闹钟的哪些位、多少位必须和时钟值匹配。

也可配置闹钟根据预配置的时间间隔重复。一旦使能了闹钟，其重复次数就会被存储到ALMRPT<7:0>位（RTCCON1H<7:0>）。当ALMRPTx位的值等于00h且CHIME位（RTCCON1H<14>）清零时，禁止重复功能，仅发生单次闹钟。通过将FFh装入ALMRPT<7:0>，可使闹钟重复最多255次。

每发生一次闹钟事件，ALMRPTx位的值都会递减1。一旦该值达到00h，就会发生最后一次闹钟事件，然后ALRMEN位自动清零，关闭闹钟。

如果CHIME位 = 1，闹钟可无限次重复发生。在这种情况下，当ALMRPTx位的值达到00h时，不会禁止闹钟，而是返回FFh并继续无限次地计数。

21.5.2 闹钟中断

每一次闹钟事件都会产生一个中断。该输出与RTCC时钟完全同步，且可用作其他外设的触发时钟。

注：	当使能闹钟（ALRMEN = 1）时，更改除RTCOE位（RTCCON1L<7>）、ALMRPT<7:0>位（RTCCON1H<7:0>）和CHIME位外的任何寄存器位都将导致错误的闹钟事件，从而引起错误的闹钟中断。为避免错误闹钟事件，只能在禁止闹钟（ALRMEN = 0）时更改定时器和闹钟值。

图21-2： 闹钟掩码设置

闹钟掩码设置（AMASK<3:0>）	星期	月	日	小时	分钟	秒
0000 —— 每半秒						
0001 —— 每秒						
0010 —— 每 10 秒						s
0011 —— 每分钟						s s
0100 —— 每 10 分钟					m	s s
0101 —— 每小时					m m	s s
0110 —— 每日				h h	m m	s s
0111 —— 每周	d			h h	m m	s s
1000 —— 每月			d d	h h	m m	s s
1001 —— 每年 (1)		m m	d d	h h	m m	s s

注 1： 每年，除非配置为2月29日。

21.6 电源控制

RTCC包含电源控制功能，允许器件定期唤醒外部器件，等待外部器件稳定后从该器件采样唤醒事件，然后关闭外部器件。这可完全由RTCC自行完成，无需从当前的低功耗模式唤醒。

要使用该功能：

1. 使能RTCC（RTCEN = 1）。
2. 将PWCEN位（RTCCON1L<10>）置1。
3. 将RTCC引脚配置为驱动PWC控制信号（RTCOE = 1且OUTSEL<2:0> = 011）。

可使用PWCPOL位（RTCCON1L<9>）选择PWC控制信号的极性。低电平有效信号或高电平有效信号可以与相应的外部开关配合使用，以接通或关断一个或多个外部器件的电源。低电平有效设置还可以与RTCC引脚的漏极开路设置结合使用，以直接驱动外部器件的接地引脚（通过相应的外部VDD上拉器件），无需使用外部开关。最后，CHIME位应置1，以使能PWC周期性。

一旦RTCC和PWC使能并运行，PWC逻辑将产生一个控制输出和一个采样门控输出。控制输出驱动到RTCC引脚（RTCOE = 1且OUTSEL<2:0> = 011），用于为器件上电或断电，如上所述。

一旦控制输出置为有效，稳定性窗口便会启动，其中外部器件被给予足够的时间来上电和提供稳定的输出。

一旦输出稳定，RTCC便会在采样窗口期间提供采样门控。该采样门控的使用取决于所使用的外部器件，但通常而言，它用于屏蔽来自外部器件的一个或多个唤醒信号。

最后，稳定性和采样窗口在采样窗口到期后关闭，外部器件断电。

21.6.1 电源控制时钟源

稳定性窗口和采样窗口由RTCCON3L寄存器中的PWCSAMPx和PWCSTABx位域（分别为RTCCON3L<15:8>和<7:0>）控制。由于稳定性窗口和采样窗口均通过RTCC时钟定义，其绝对值因PWC时钟基本周期（TPWCCLK）的值而异。例如，使用32.768 kHz SOSC输入时钟将产生1/32768 = 30.518 µs的TPWCCLK。PWCSTABx和PWCSAMPx的8位大小允许窗口大小为0到255个TPWCCLK。PWC时钟的周期也可以通过1:1、1:16、1:64或1:256预分频比调整，具体取决于PWCPS<1:0>位（RTCCON2L<7:6>）。

此外，在确定电源控制操作时，PWCSTABx和PWCSAMPx位域的某些值具有特定的控制意义。如果任一位域为00h，对应的窗口将不起作用。此外，如果PWCSTABx位域为FFh，即使禁止电源控制，稳定性窗口仍将保持活动状态。

21.7 时间戳事件

RTCC包含一组时间戳寄存器，可以在接收到外部输入信号时捕捉时间和日期寄存器值。当TMPRN引脚出现低电平脉冲时，RTCC将触发时间戳事件。

21.7.1 时间戳操作

使用TSAEN位（RTCCON1L<0>）使能时间戳事件输入。当时间戳事件发生时，当前时间和日期值将被存储在TSATIMEL/H和TSADATEL/H寄存器中，TSAEVT状态位（RTCSTATL<3>）将被置1，并且会发生RTCC中断。在用户清除TSAEVT状态位之前，不能发生新的时间戳捕捉事件。

注 1：	当TSAEN = 0时，TSATIMEL/H和TSADATEL/H寄存器对可用于数据存储。所有类型的非上电复位（MCLR和WDT等）过程中，将保持TSATIMEL/H和TSADATEL/H的值。

21.7.2 手动时间戳操作

使能时间戳功能（TSAEN = 1）时，可通过向TSAEVT位写入1将当前时间和日期捕捉到TSATIMEL/H和TSADATEL/H寄存器中。该写操作不会将TSAEVT位置1，而是启动时间戳捕捉。捕捉操作完成时，TSAEVT位将置1。用户必须轮询TSAEVT位以确定捕捉操作何时完成。

在读取时间戳寄存器之后，应将TSAEVT位清零以实现进一步的硬件或软件时间戳捕捉事件。

22.0 32位可编程循环冗余校验（CRC）发生器

注：	本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的 **“32位可编程循环冗余校验（CRC）”**（DS30009729），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

32位可编程CRC发生器为各种网络和安全应用提供快速产生校验和的硬件实现方法。它提供以下特性：

- 用户可编程的CRC多项式公式，最多32位
- 可编程移位方向（小尾数法或大尾数法格式）
- 独立的数据和多项式长度
- 可配置中断输出
- 数据FIFO

CRC发生器的简化框图如图22-1所示。图22-2中给出了CRC移位引擎的简化形式。

图22-1： CRC框图

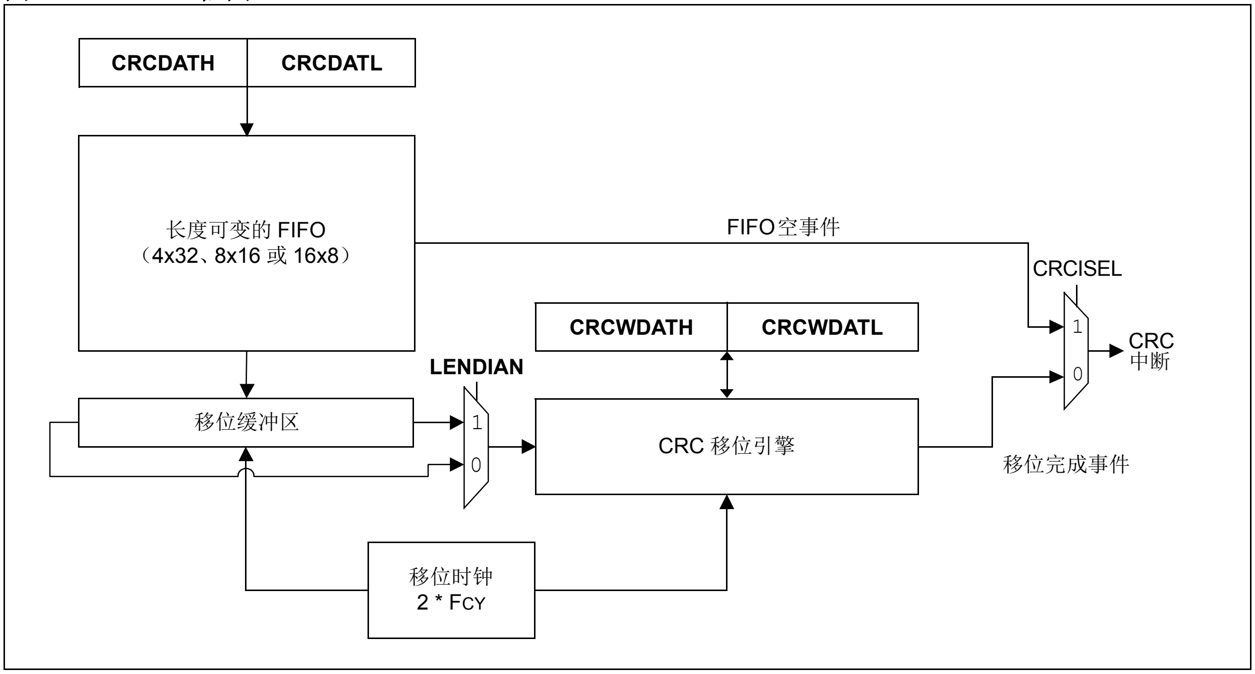
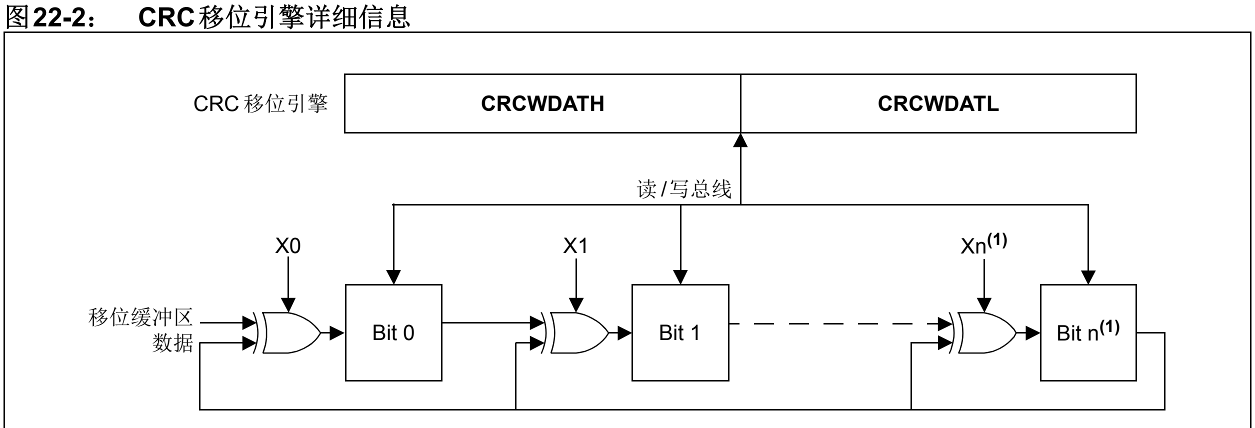


图22-2： CRC移位引擎详细信息



注 1： n = PLEN<4:1> + 1。

22.1 用户接口

22.1.1 多项式接口

可使用最多32位，将CRC模块编程为最多产生32阶的CRC多项式。

通过PLEN<4:0>位（CRCCON2<4:0>）选择反映公式中最高次幂的多项式长度。

CRCXORL和CRCXORH寄存器控制公式中包含的幂项。将某个特定位置1将会在公式中包含相应的幂项。在功能上相当于与CRC引擎的相应位进行异或操作。清零该位会禁止此异或操作。

例如，考虑两个CRC多项式，一个是16位公式，另一个是32位公式。

公式22-1： 16位和32位CRC多项式

X16 + X12 + X5 + 1

和

X32+X26 + X23 + X22 + X16 + X12 + X11 + X10 + X8 + X7 + X5 + X4 + X2 + X + 1

要将这些多项式编程到CRC发生器中，请按表22-1中设置寄存器位。

注意，请将相应位置1以表示公式中将使用相应的项（例如X26和X23）。因为公式的第0位始终参与逻辑异或运算，因此X0位是无关位。对于长度为32的多项式，总是假定将使用第32位。因此，存在X<31:1>位，没有第32位。

22.1.2 数据接口

CRC模块具有可缓存各种数据宽度的FIFO。可使用DWIDTH<4:0>位（CRCCON2<12:8>）将输入数据宽度配置为1到32位之间的任何值。当数据宽度大于15时，FIFO为4字深。当DWIDTHx值在8至15之间时，FIFO为8字深。当DWIDTHx值小于8时，FIFO为16字深。

首先将CRC计算所需的数据写入FIFO。即使数据宽度小于8，可被写入FIFO的最小数据元素也是1个字节。例如，如果DWIDTHx位为5，则数据大小为DWIDTH<4:0> + 1或6。数据作为整个字节写入，CRC模块忽略两个未使用的高位。

数据一旦被写入CRCDAT寄存器的最高有效位（即由数据宽度定义的最高有效位），VWORD<4:0>位（CRCCON1<12:8>）的值将递增1。例如，如果DWIDTHx位为24，则写入CRCDATH的bit 7时，VWORDx位递增1。因此，CRCDATL必须始终先于CRCDATH写入。

当CRCGO位（CRCCON1<4>）置1且VWORDx位的值大于零时，CRC引擎开始移入数据。

每个字都被从FIFO复制到缓冲寄存器（这会导致VWORDx位递减1）。然后从缓冲区中移出数据。CRC引擎会以每指令周期两个位的速率继续移入数据直到VWORDx位达到0。因此对于给定的数据宽度，每个字完成计算所需的指令周期数为数据宽度的一半。例如，完成一个32位数据字的CRC计算需要16个周期。

当VWORDx位达到所配置DWIDTHx位的最大值（4、8或16）时，CRCFUL位（CRCCON1<7>）置1。当VWORDx位达到零时，CRCMPT位（CRCCON1<6>）置1。只要CRCEN为0，FIFO就为空且将VWORD<4:0>位设置为00000。

在写入CRCWDAT之后，必须至少经过一个指令周期才可以读VWORDx位。

表22-1： CRC设置示例（16位和32位多项式）

CRC控制位	位值	
	16位多项式	32位多项式
PLEN<4:0>	01111	11111
X<31:16>	0000 0000 0000 0001	0000 0100 1100 0001
X<15:1>	0001 0000 0010 000	0001 1101 1011 011

22.1.3 数据移位方向

LENDIAN位（CRCCON1<3>）用于控制移位方向。默认情况下，CRC通过引擎从最高有效位开始移入数据。将LENDIAN置1（= 1）可导致CRC从最低有效位开始移入数据。此设置可使CRC与各种通信机制更好地集成，并省去了用软件使位反序的开销。注意，这只是改变了将数据移入引擎的方向。CRC计算的结果仍然是正常的CRC结果，不是反序的CRC结果。

22.1.4 中断操作

用户可以配置该模块在以下两种条件下产生中断。

如果CRCISEL为0，当VWORD<4:0>位的值从1跳变为0时产生中断。如果CRCISEL为1，在CRC操作完成且模块将CRCGO位置0时产生中断。手动将CRCGO置0不会产生中断。请注意，产生中断时，CRC 计算尚未完成。产生中断后，模块将仍需要(PLENx + 1)/2个时钟周期，才会完成CRC计算。

22.1.5 典型操作

要使用模块进行典型CRC计算，请执行以下步骤：

1. 将CRCEN位置1以使能模块。
2. 配置模块以实现所需的操作：
   a) 使用CRCXOR寄存器和PLEN<4:0>位对所需多项式编程。
   b) 使用DWIDTH<4:0>和LENDIAN位配置数据宽度和移位方向。
3. 将CRCGO位置1以启动计算。
4. 通过写入CRCWDAT寄存器设置所需的CRC非直接初始值。
5. 在空间可用时，通过写入CRCDAT寄存器将所有数据加载到FIFO（在加载下一个数据之前，CRCFUL位必须为零）。
6. 等待至数据FIFO为空（CRCMPT位置1）。
7. 读取结果：
   如果数据宽度（DWIDTH<4:0>位）大于多项式长度（PLEN<4:0>位）：
   a) 等待(DWIDTH<4:0> + 1)/2个指令周期以确保移位缓冲区的移位已完成。
   b) 将数据宽度更改为多项式长度（DWIDTH<4:0> = PLEN<4:0>）。
   c) 将一个虚设数据字写入CRCDAT寄存器。
   d) 等待2个指令周期以将FIFO的数据移至移位缓冲区，并等待(PLEN<4:0> + 1)/2个指令周期来移出结果。

   或者，如果数据宽度（DWIDTH<4:0>位）小于多项式长度（PLEN<4:0>位）：
   1. 当所有移位完成时，清零CRC中断选择位（CRCISEL = 0）来获取中断。清零CRC中断标志。将虚设数据写入CRCDAT寄存器并等待至将CRC中断标志置1。
   2. 从CRCWDAT寄存器读取最终的CRC结果。
   3. 恢复数据宽度（DWIDTH<4:0>位）以进行进一步计算（可选）。如果数据宽度（DWIDTH<4:0>位）等于或小于多项式长度（PLEN<4:0>位）：
      a) 当所有移位完成时，清零CRC中断选择位（CRCISEL = 0）来获取中断。
      b) 通过设置CRCGO = 0来暂停计算。
      c) 清零CRC中断标志。
      d) 写入总数据长度与CRCDAT寄存器中多项式长度相等的虚设数据。
      e) 通过设置CRCGO = 1恢复计算。
      f) 等待至CRC中断标志置1。
      g) 从CRCWDAT寄存器读取最终的CRC结果。

有8个寄存器用于控制可编程CRC操作：

- CRCCON1
- CRCCON2
- CRCXORL
- CRCXORH
- CRCDATL
- CRCDATH
- CRCWDATL
- CRCWDATH

CRCCON1和CRCCON2寄存器（寄存器22-1和寄存器22-2）用于控制模块的操作和配置各种设置。

CRCXOR寄存器（寄存器22-3和寄存器22-4）用于选择CRC公式中要使用的多项式项。CRCDAT和CRCWDAT寄存器是分别用作双字输入数据和CRC处理输出的缓冲区的寄存器对。

寄存器22-1： **CRCCON1：CRC控制1寄存器**

R/W-0	U-0	R/W-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
CRCEN	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0
bit 15							bit 8

R-0, HSC	R-1, HSC	R/W-0	R/W-0, HC	R/W-0	U-0	U-0	U-0
CRCFUL	CRCMPT	CRCISEL	CRCGO	LENDIAN	—	—	—
bit 7							bit 0

图注：	HC = 硬件清零位	HSC = 硬件置1/清零位	
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15 **CRCEN**：CRC使能位
1 = 使能模块
0 = 禁止模块；所有状态机、指针和CRCWDAT/CRCDAT寄存器都复位；其他SFR不复位

bit 14 **未实现**：读为0

bit 13 **CSIDL**：空闲模式CRC停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 在空闲模式下模块继续工作

bit 12-8 **VWORD<4:0>**：CRC指针值位
指出FIFO中有效字的个数。当PLEN<4:0> ≥ 7时，最大值为8；当PLEN<4:0> ≤ 7时，最大值为16。

bit 7 **CRCFUL**：CRC FIFO满位
1 = FIFO为满
0 = FIFO未满

bit 6 **CRCMPT**：CRC FIFO空位
1 = FIFO为空
0 = FIFO非空

bit 5 **CRCISEL**：CRC中断选择位
1 = FIFO空时产生中断；仍通过CRC移位最后的数据字
0 = 移位完成时产生中断且结果已计算好

bit 4 **CRCGO**：启动CRC位
1 = 启动CRC串行移位器
0 = 关闭CRC串行移位器

bit 3 **LENDIAN**：数据移位方向选择位
1 = 数据字从最低有效位开始移入CRC（小尾数法格式）
0 = 数据字从最高有效位开始移入CRC（大尾数法格式）

bit 2-0 **未实现**：读为0

寄存器22-2： **CRCCON2：CRC控制寄存器2**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DWIDTH4	DWIDTH3	DWIDTH2	DWIDTH1	DWIDTH0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	PLEN4	PLEN3	PLEN2	PLEN1	PLEN0
bit 7							bit 0

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

bit 15-13 **未实现：**读为0

bit 12-8 **DWIDTH<4:0>：**CRC数据字宽度配置位
配置数据字的宽度（数据字宽度-1）。

bit 7-5 **未实现：**读为0

bit 4-0 **PLEN<4:0>：**多项式长度配置位
配置多项式的长度（多项式长度-1）。

寄存器22-3： **CRCXORL：CRC多项式异或操作寄存器（低位字）**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
X<7:1>							—
bit 7							bit 0

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

bit 15-1 **X<15:1>**：多项式的项x^n的异或操作使能位

bit 0 **未实现**：读为0

寄存器22-4： **CRCXORH：CRC多项式异或操作寄存器（高位字）**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X<31:24>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X<23:16>							
bit 7							bit 0

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

bit 15-0 **X<31:16>**：多项式的项x^n的异或操作使能位

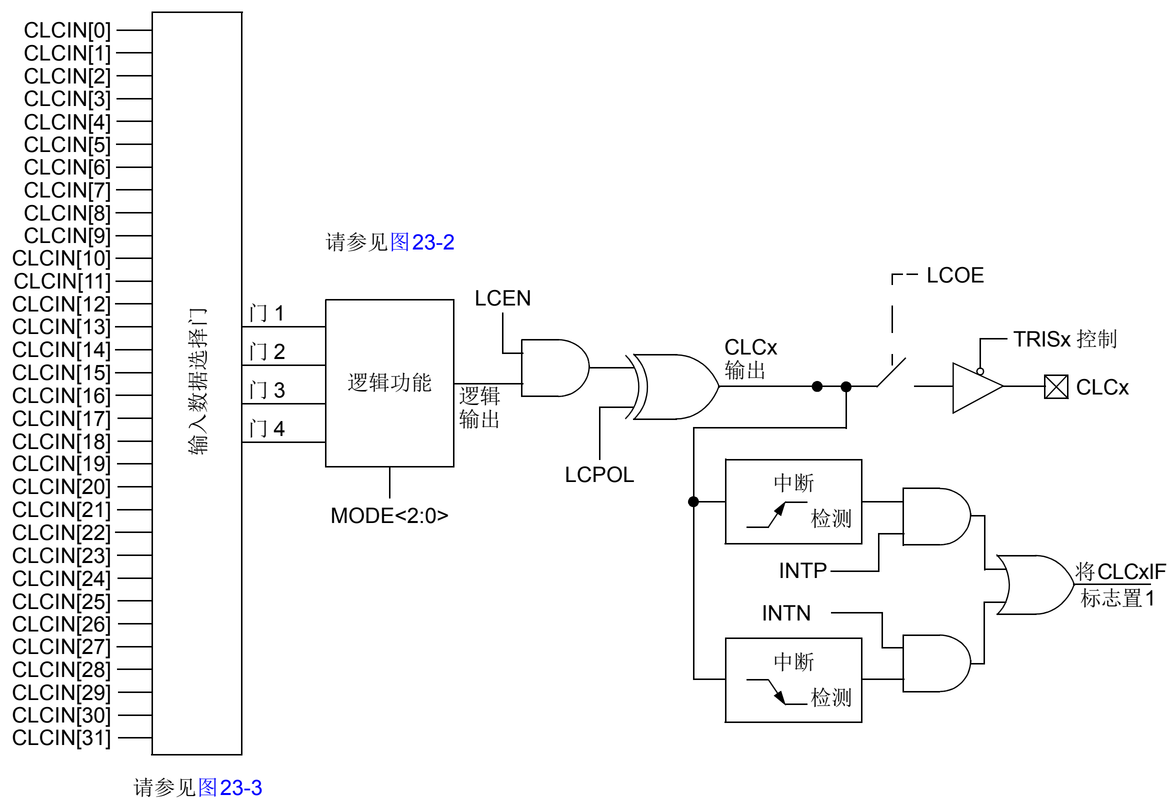
23.0 可配置逻辑单元（CLC）发生器

注：	本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的“**可配置逻辑单元（CLC）**”（DS33949），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

可配置逻辑单元（CLC）模块允许用户将信号组合指定为逻辑功能的输入以及允许用户使用逻辑输出来控制其他外设或I/O引脚。另外，CLC模块的操作不受软件执行限制并支持大量的输出设计，这为嵌入式设计提供了更大的灵活性与更多的可能性。

选定的逻辑功能具有4个输入门。这4个输入门使用4个数据源选择多路开关从最多32个信号的信号池中选择输入数据源。图23-1给出了模块的概要图。图23-3显示了数据源多路开关和逻辑输入门连接的详细信息。

图23-1： CLCx模块



注： 本图中显示的所有寄存器位均可在CLCxCONL寄存器中找到。

图 23-2： CLCx 逻辑功能组合选项

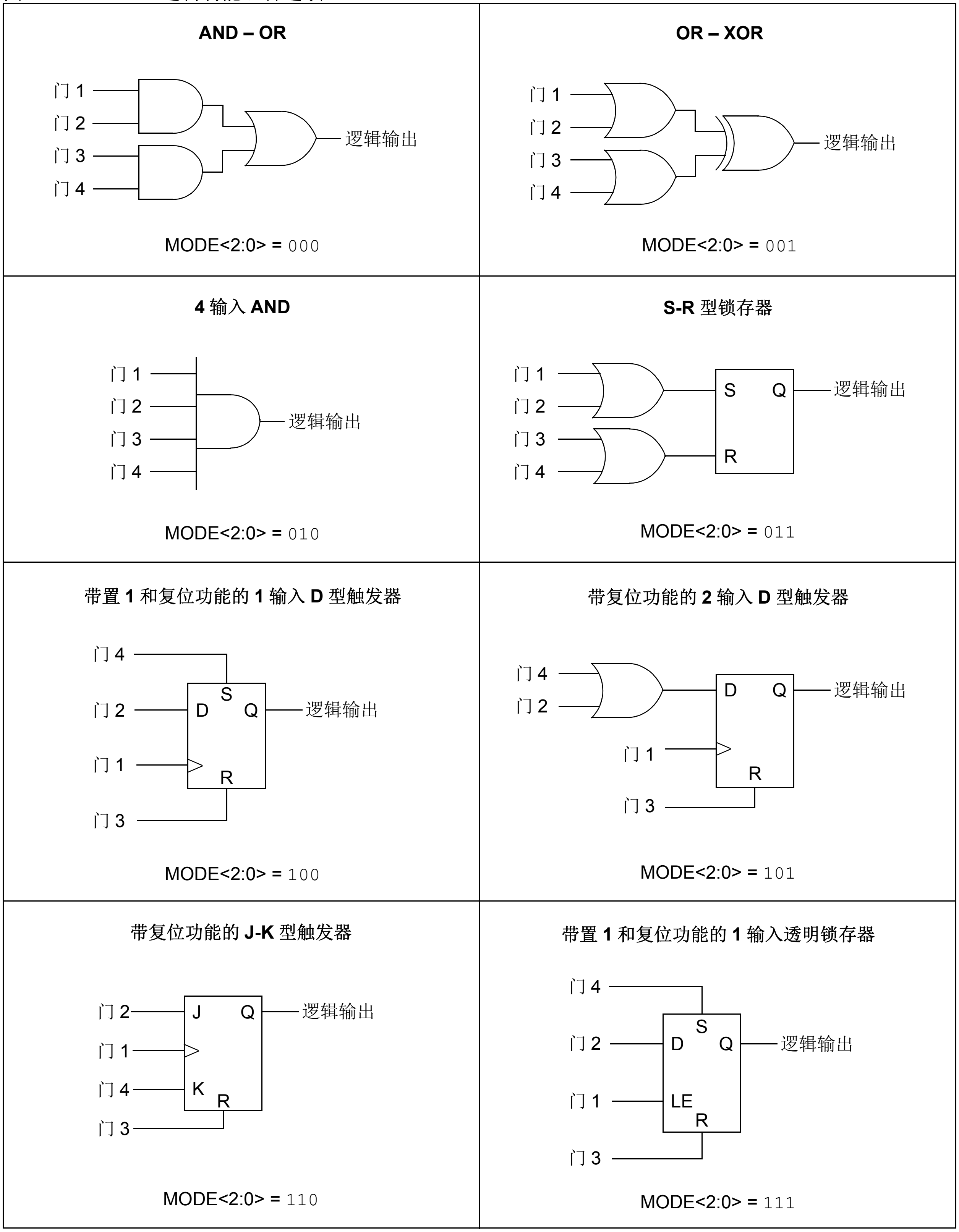
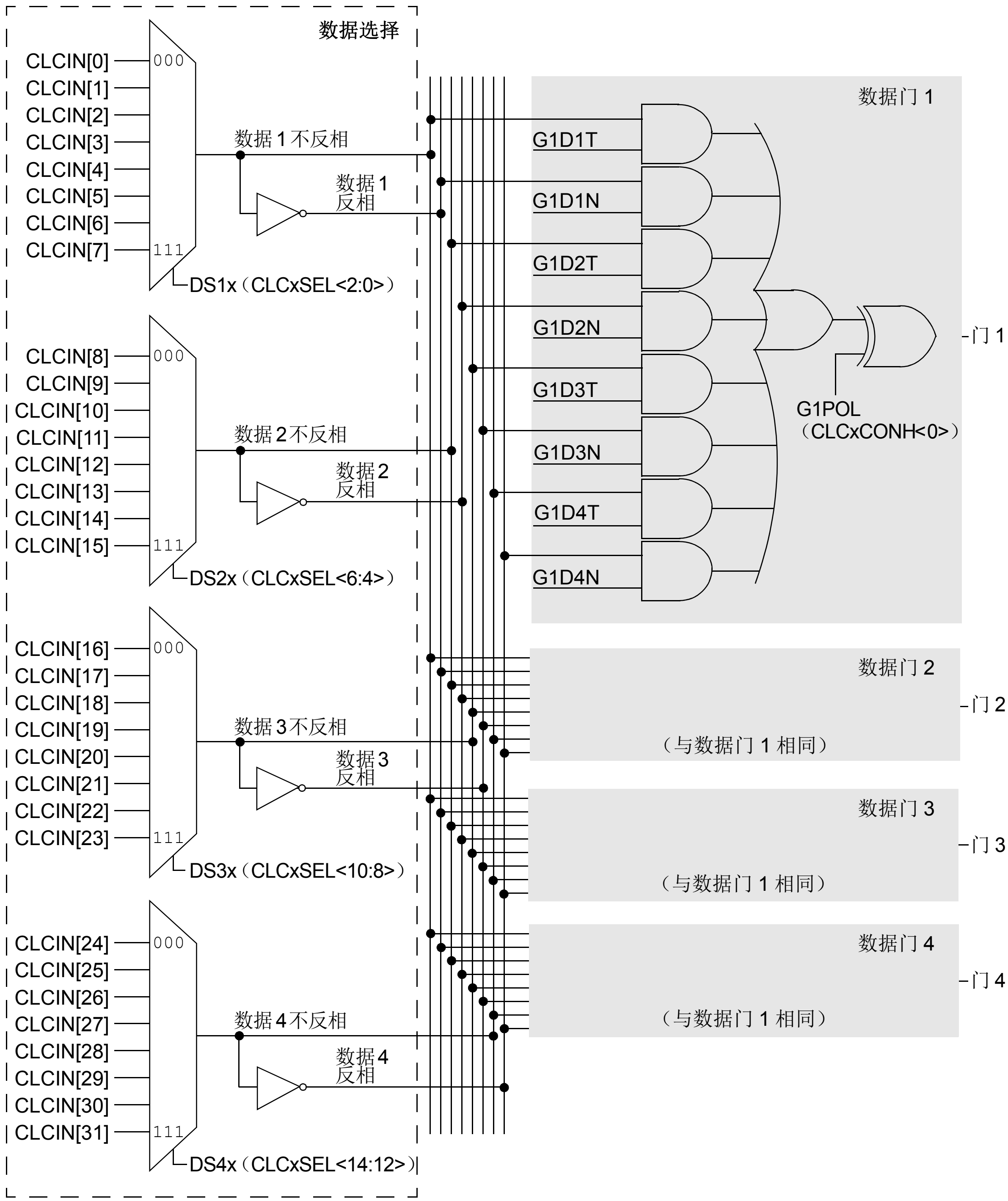


图23-3：　CLCx输入源选择框图



注：　在上电时，所有控制都是未定义的。

23.1 控制寄存器

CLCx模块由以下寄存器控制：

- CLCxCONL
- CLCxCONH
- CLCxSEL
- CLCxGLSL
- CLCxGLSH

CLCx控制寄存器（CLCxCONL和CLCxCONH）用于使能模块和允许中断、控制输出使能位、选择输出极性和选择逻辑功能。此外，CLCx控制寄存器不仅允许用户控制单元输出的逻辑极性，而且允许控制一些中间变量的逻辑极性。

CLCx输入多路开关选择寄存器（CLCxSEL）允许用户使用4个数据输入选择多路开关最多选择4个数据输入源。每个多路开关具有8个可用的数据源。

CLCx门逻辑输入选择寄存器（CLCxGLSL和CLCxGLSH）允许用户选择使用来自每个选择多路开关的哪些输出作为逻辑单元输入门的输入。每个数据源多路开关会同时输出其输出的真值和取反形式。这8个信号全部都会被使能，通过逻辑单元输入门一起进行逻辑或运算。

寄存器23-1： CLCxCONL：CLCx控制寄存器（低位字）

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0
LCEN	—	—	—	INTP	INTN	—	—
bit 15							bit 8

R/W-0	R-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
LCOE	LCOUT	LCPOL	—	—	MODE2	MODE1	MODE0
bit 7							bit 0

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

bit 15 **LCEN：**CLCx使能位
1 = 使能CLCx，并对输入信号进行混合
0 = 禁止CLCx，并且其输出为逻辑零

bit 14-12 **未实现：**读为0

bit 11 **INTP：**CLCx正边沿中断允许位
1 = LCOUT上出现上升沿时，将产生中断
0 = 不产生中断

bit 10 **INTN：**CLCx负边沿中断允许位
1 = LCOUT上出现下降沿时，将产生中断
0 = 不产生中断

bit 9-8 **未实现：**读为0

bit 7 **LCOE：**CLCx端口使能位
1 = 使能CLCx端口引脚输出
0 = 禁止CLCx端口引脚输出

bit 6 **LCOUT：**CLCx数据输出状态位
1 = CLCx输出高电平
0 = CLCx输出低电平

bit 5 **LCPOL：**CLCx输出极性控制位
1 = 模块的输出反相
0 = 模块的输出不反相

bit 4-3 **未实现：**读为0

寄存器23-1： CLCxCONL：CLCx控制寄存器（低位字）（续）

bit 2-0 **MODE<2:0>**：CLCx模式位
111 = 单元是带置1和复位功能的1输入透明锁存器
110 = 单元是带复位功能的JK型触发器
101 = 单元是带复位功能的2输入D型触发器
100 = 单元是带置1和复位功能的1输入D型触发器
011 = 单元是SR锁存器
010 = 单元是4输入AND逻辑
001 = 单元是OR-XOR逻辑
000 = 单元是AND-OR逻辑

寄存器23-2： CLCxCONH：CLCx控制寄存器（高位字）

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	G4POL	G3POL	G2POL	G1POL
bit 7							bit 0

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

bit 15-4 **未实现**：读为0

bit 3 **G4POL**：门4极性控制位
1 = 通道4逻辑的输出在施加到逻辑单元时反相
0 = 通道4逻辑的输出不反相

bit 2 **G3POL**：门3极性控制位
1 = 通道3逻辑的输出在施加到逻辑单元时反相
0 = 通道3逻辑的输出不反相

bit 1 **G2POL**：门2极性控制位
1 = 通道2逻辑的输出在施加到逻辑单元时反相
0 = 通道2逻辑的输出不反相

bit 0 **G1POL**：门1极性控制位
1 = 通道1逻辑的输出在施加到逻辑单元时反相
0 = 通道1逻辑的输出不反相

寄存器23-3： CLCxSEL：CLCx输入多路开关选择寄存器

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	DS4<2:0>			—	DS3<2:0>		
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	DS2<2:0>			—	DS1<2:0>		
bit 7							bit 0

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

bit 15 **未实现：**读为0

bit 14-12 **DS4<2:0>：**数据选择多路开关4信号选择位
111 = MCCP3比较事件中断标志（CCP3IF）
110 = MCCP1比较事件中断标志（CCP1IF）
101 = 未实现
100 = CTMU A/D触发信号
011 = 对应于CLCx模块的SPIx输入（SDIx）（见表23-1）
010 = 比较器3输出
001 = 特定于模块的CLCx输出（见表23-1）
000 = CLCINB I/O引脚

bit 11 **未实现：**读为0

bit 10-8 **DS3<2:0>：**数据选择多路开关3信号选择位
111 = MCCP3比较事件中断标志（CCP3IF）
110 = MCCP2比较事件中断标志（CCP2IF）
101 = DMA通道1中断
100 = 对应于CLCx模块的UARTx RX输出（见表23-1）
011 = 对应于CLCx模块的SPIx输出（SDOx）（见表23-1）
010 = 比较器2输出
001 = CLCx输出（见表23-1）
000 = CLCINA I/O引脚

bit 7 **未实现：**读为0

bit 6-4 **DS2<2:0>：**数据选择多路开关2信号选择位
111 = MCCP2比较事件中断标志（CCP2IF）
110 = MCCP1比较事件中断标志（CCP1IF）
101 = DMA通道0中断
100 = A/D转换完成中断
011 = 对应于CLCx模块的UARTx TX输入（见表23-1）
010 = 比较器1输出
001 = CLCx输出（见表23-1）
000 = CLCINB I/O引脚

bit 3 **未实现：**读为0

bit 2-0 **DS1<2:0>：**数据选择多路开关1信号选择位
111 = Timer3匹配事件
110 = Timer2匹配事件
101 = 未实现
100 = REFO输出
011 = INTRC/LPRC时钟源
010 = SOSC时钟源
001 = 系统时钟（T_{CY}）
000 = CLCINA I/O引脚

表23-1： 特定于模块的输入数据源

位域值		输入源	
		CLC1	CLC2
DS4<2:0>	011	SDI1	SDI2
	001	CLC2输出	CLC1输出
DS3<2:0>	100	U1RX	U2RX
	011	SDO1	SDO2
	001	CLC1输出	CLC2输出
DS2<2:0>	011	U1TX	U2TX
	001	CLC2输出	CLC1输出

寄存器23-4： **CLCxGLSL：CLCx门逻辑输入选择低位字寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N
bit 7							bit 0

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

bit 15 **G2D4T**：门2数据源4真值使能位
1 = 为门2使能数据源4信号
0 = 为门2禁止数据源4信号

bit 14 **G2D4N**：门2数据源4取反使能位
1 = 为门2使能数据源4反相信号
0 = 为门2禁止数据源4反相信号

bit 13 **G2D3T**：门2数据源3真值使能位
1 = 为门2使能数据源3信号
0 = 为门2禁止数据源3信号

bit 12 **G2D3N**：门2数据源3取反使能位
1 = 为门2使能数据源3反相信号
0 = 为门2禁止数据源3反相信号

bit 11 **G2D2T**：门2数据源2真值使能位
1 = 为门2使能数据源2信号
0 = 为门2禁止数据源2信号

bit 10 **G2D2N**：门2数据源2取反使能位
1 = 为门2使能数据源2反相信号
0 = 为门2禁止数据源2反相信号

bit 9 **G2D1T**：门2数据源1真值使能位
1 = 为门2使能数据源1信号
0 = 为门2禁止数据源1信号

寄存器23-4： CLCxGLSL：CLCx门逻辑输入选择低位字寄存器（续）

bit 8	**G2D1N**：门2数据源1取反使能位 1 = 为门2使能数据源1反相信号 0 = 为门2禁止数据源1反相信号
bit 7	**G1D4T**：门1数据源4真值使能位 1 = 为门1使能数据源4信号 0 = 为门1禁止数据源4信号
bit 6	**G1D4N**：门1数据源4取反使能位 1 = 为门1使能数据源4反相信号 0 = 为门1禁止数据源4反相信号
bit 5	**G1D3T**：门1数据源3真值使能位 1 = 为门1使能数据源3信号 0 = 为门1禁止数据源3信号
bit 4	**G1D3N**：门1数据源3取反使能位 1 = 为门1使能数据源3反相信号 0 = 为门1禁止数据源3反相信号
bit 3	**G1D2T**：门1数据源2真值使能位 1 = 为门1使能数据源2信号 0 = 为门1禁止数据源2信号
bit 2	**G1D2N**：门1数据源2取反使能位 1 = 为门1使能数据源2反相信号 0 = 为门1禁止数据源2反相信号
bit 1	**G1D1T**：门1数据源1真值使能位 1 = 为门1使能数据源1信号 0 = 为门1禁止数据源1信号
bit 0	**G1D1N**：门1数据源1取反使能位 1 = 为门1使能数据源1反相信号 0 = 为门1禁止数据源1反相信号

寄存器23-5： **CLCxGLSH：CLCx门逻辑输入选择高位字寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N
bit 7							bit 0

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

bit 15 **G4D4T：**门4数据源4真值使能位
1 = 为门4使能数据源4信号
0 = 为门4禁止数据源4信号

bit 14 **G4D4N：**门4数据源4取反使能位
1 = 为门4使能数据源4反相信号
0 = 为门4禁止数据源4反相信号

bit 13 **G4D3T：**门4数据源3真值使能位
1 = 为门4使能数据源3信号
0 = 为门4禁止数据源3信号

bit 12 **G4D3N：**门4数据源3取反使能位
1 = 为门4使能数据源3反相信号
0 = 为门4禁止数据源3反相信号

bit 11 **G4D2T：**门4数据源2真值使能位
1 = 为门4使能数据源2信号
0 = 为门4禁止数据源2信号

bit 10 **G4D2N：**门4数据源2取反使能位
1 = 为门4使能数据源2反相信号
0 = 为门4禁止数据源2反相信号

bit 9 **G4D1T：**门4数据源1真值使能位
1 = 为门4使能数据源1信号
0 = 为门4禁止数据源1信号

bit 8 **G4D1N：**门4数据源1取反使能位
1 = 为门4使能数据源1反相信号
0 = 为门4禁止数据源1反相信号

bit 7 **G3D4T：**门3数据源4真值使能位
1 = 为门3使能数据源4信号
0 = 为门3禁止数据源4信号

bit 6 **G3D4N：**门3数据源4取反使能位
1 = 为门3使能数据源4反相信号
0 = 为门3禁止数据源4反相信号

bit 5 **G3D3T：**门3数据源3真值使能位
1 = 为门3使能数据源3信号
0 = 为门3禁止数据源3信号

bit 4 **G3D3N：**门3数据源3取反使能位
1 = 为门3使能数据源3反相信号
0 = 为门3禁止数据源3反相信号

寄存器23-5： **CLCxGLSH：CLCx门逻辑输入选择高位字寄存器（续）**

bit 3 **G3D2T：**门3数据源2真值使能位
1 = 为门3使能数据源2信号
0 = 为门3禁止数据源2信号

bit 2 **G3D2N：**门3数据源2取反使能位
1 = 为门3使能数据源2反相信号
0 = 为门3禁止数据源2反相信号

bit 1 **G3D1T：**门3数据源1真值使能位
1 = 为门3使能数据源1信号
0 = 为门3禁止数据源1信号

bit 0 **G3D1N：**门3数据源1取反使能位
1 = 为门3使能数据源1反相信号
0 = 为门3禁止数据源1反相信号

24.0 带阈值检测功能的12位A/D转换器

> **注：** 本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解有关12位A/D转换器的更多信息，请参见《dsPIC33/PIC24系列参考手册》的 **"带阈值检测功能的12位A/D转换器"**（DS39739），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

A/D转换器具有以下主要特性：

- 逐次逼近寄存器（Successive Approximation Register，SAR）转换
- 可选10位或12位（默认）转换分辨率
- 转换速度最高可达200 ksps（12位）
- 最多19个模拟输入通道（内部和外部）
- 多个内部参考电压输入通道
- 外部参考电压输入引脚
- 单极型差分采样/保持（S/H）放大器
- 自动阈值扫描和比较操作，用于预测转换结果
- 可选择转换触发源
- 固定长度（每通道一个字）的可配置转换结果缓冲区
- 4个结果对齐选项
- 中断产生可配置
- 采用间接地址生成的增强型DMA操作
- 在CPU休眠和空闲模式期间工作

12位A/D转换器模块是早期PIC24器件中提供的10位模块的改进版本。它是逐次逼近寄存器（SAR）转换器，相比之前所作的改进有：12位分辨率，提供广泛的自动采样选项，与其他模拟模块更紧密地集成，以及具有可配置的结果缓冲区。

该模块还包含了一个特有的阈值检测功能，让模块自身可以基于转换结果做出简单决定，通过外设间接寻址（Peripheral Indirect Addressing，PIA）增强DMA控制器的操作。

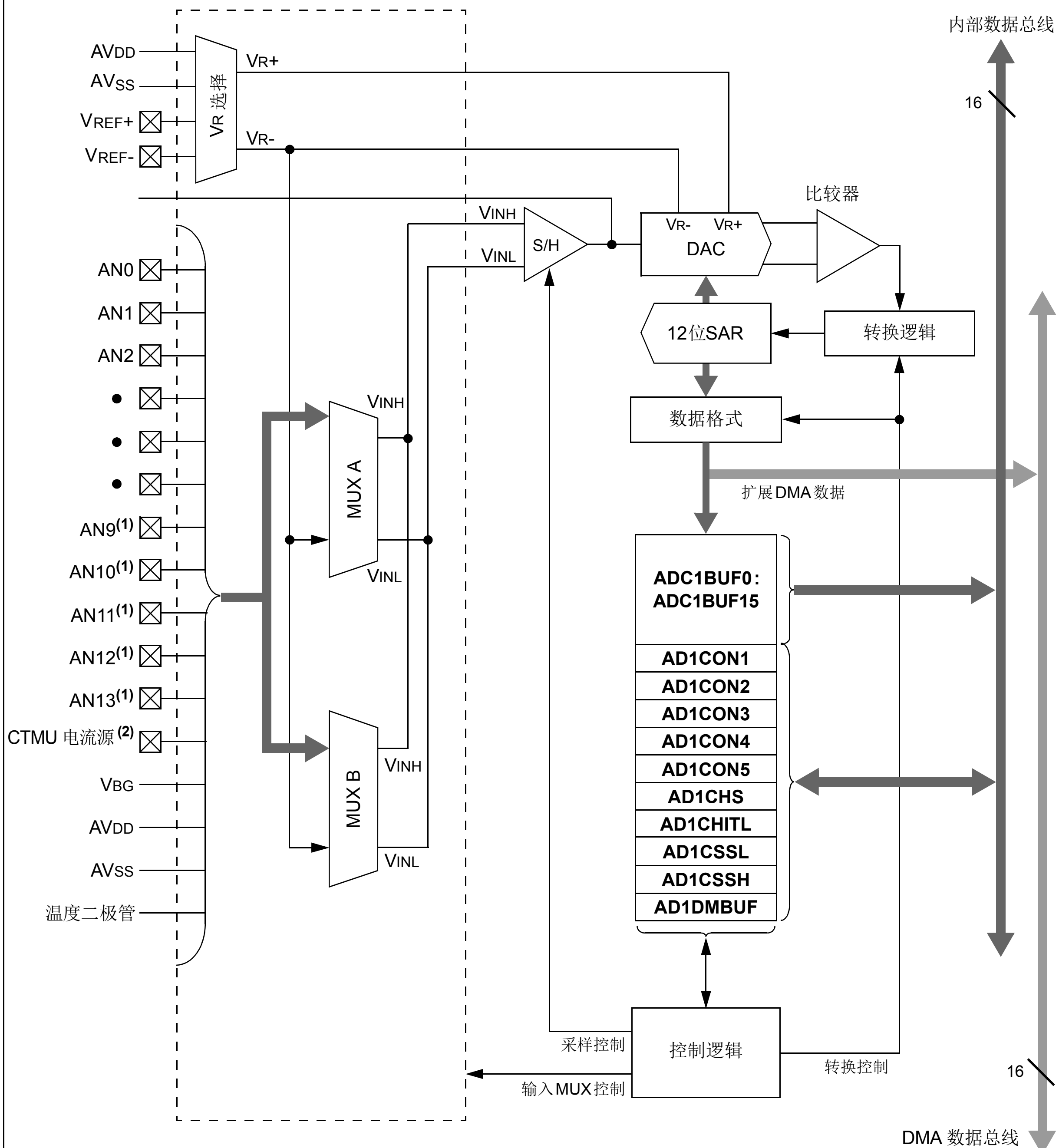
图24-1给出了该模块的简化框图。

24.1 基本操作

要执行标准A/D转换：

1. 配置模块：
   a) 将ANSx寄存器的相应位置1以将端口引脚配置为模拟输入（详细信息请参见**第11.2节"配置模拟端口引脚（ANSx）"**）。
   b) 选择能满足模拟输入预期范围的参考电压源（AD1CON2<15:13>）。
   c) 为每条通道选择正负多路开关输入（AD1CHS<15:0>）。
   d) 选择模拟转换时钟以便使期望的数据速率与处理器时钟匹配（AD1CON3<7:0>）。
   e) 选择适当的采样/转换序列（AD1CON1<7:4>和AD1CON3<12:8>）。
   f) 对于通道A扫描操作，选择要包括的正通道（AD1CSSH和AD1CSSL寄存器）。
   g) 选择转换结果在缓冲区（AD1CON1<9:8>和AD1CON5寄存器）中的存储方式。
   h) 选择中断发生的频率（AD1CON2<5:2>）。
   i) 启动A/D模块（AD1CON1<15>）。
2. 配置A/D中断（如果需要）：
   a) 清零AD1IF位（IFS0<13>）。
   b) 允许AD1IE中断（IEC0<13>）。
   c) 选择A/D中断优先级（IPC3<6:4>）。
3. 如果模块配置为手动采样，将SAMP位（AD1CON1<1>）置1开始采样。

图24-1： 12位A/D转换器框图（PIC24FJ256GA705系列）



注 1： Anx引脚是否可用取决于封装。

2： 当SAMP = 1且TGEN = 0时，CTMU电流源连接至所选ANx引脚。有关详细信息，请参见**第27.0节“充电时间测量单元（CTMU）”**。

24.2 扩展DMA操作

除所有12位A/D转换器都具备的标准功能外，PIC24FJ256GA705系列器件还实现了有限的DMA功能扩展。此项扩展添加了与器件的DMA控制器配合工作的功能，在模块内置缓冲区的基础上扩展了A/D模块的数据存储能力。

DMA功能扩展由DMAEN位（AD1CON1<11>）控制；将此位置1可使能该功能。DMABM位（AD1CON1<12>）配置DMA功能的工作方式。

24.2.1 扩展缓冲区模式

在扩展缓冲区模式（DMABM = 1）下，A/D数据缓冲寄存器以及来自所有编号大于13的通道的数据均将映射到用户指定的数据RAM区域。这使用户能够从数据存储器读取编号大于13的通道（自身没有存储器映射A/D缓冲单元）的转换结果。

为此，必须将DMA配置为外设间接寻址模式并使DMA目标地址指向缓冲区的起始处。必须设置DMA计数，以便在达到所需转换次数后产生中断。

在扩展缓冲区模式下，A/D控制位的功能将与非DMA模式类似。BUFREGEN位仍将用于选择FIFO模式和通道对齐模式，但是目标FIFO中的字数将由DMA模式下的SMPI<4:0>位决定。在FIFO模式下，BUFM位仍会将输出FIFO分为两组（每组13个结果，SMPIx位应进行相应设置），而且BUFS位仍将指示哪一组结果将被写入以及哪一组结果可读取。

24.2.2 PIA模式

DMABM = 0时，A/D模块配置为与DMA控制器协同进行外设间接寻址（PIA）模式操作。在此模式下，A/D模块生成一个11位间接地址（Indirect Address，IA）。这与DMA控制器中的目标地址进行逻辑或运算，以定义A/D转换数据的存储位置。

在PIA模式下，缓冲区空间创建为一系列连续的小缓冲区（每个模拟通道一个）。通道缓冲区的大小确定可容纳多少条模拟通道。通过DMABL<2:0>位（AD1CON4<2:0>）选择缓冲区的大小。每个缓冲区大小的选择范围是1个字到128个字。每条通道都分配一个这种大小的缓冲区，无论通道是否真正具有转换数据。

通过将通道缓冲区内的基地址与三到五个位（取决于缓冲区大小）相连接来创建IA以标识通道。基地址的宽度范围是零到七位，具体取决于缓冲区大小。该地址右侧填充0，以在数据空间中保持地址对齐。然后根据需要在连接而成的通道和基地址位左侧填充零补全11位IA。

写操作期间使用SMPIx位（AD1CON2<6:2>）将IA配置为自动递增将写入每个模拟输入的子缓冲区的通道。

和任何具有DMA功能的模块的PIA操作一样，DMADSTn寄存器中的目标基地址必须适当进行掩码以适应IA。表24-1给出了完整地址的形成方式。请注意，地址掩码根据缓冲区大小的不同而不同。由于存在掩码要求，特定缓冲区大小可能无法使用某些地址范围。用户应验证DMA基地址是否与所选缓冲区大小兼容。

图24-2显示了地址中各个部分对数据存储器中缓冲单元的定义方式。此例中，模块为32个4字缓冲区“分配”256字节的数据RAM（1000h至1100h）。但这并不是硬性分配，而且不能阻止这些单元用于其他用途。例如在此例中，如果模拟通道1、3和8正在进行采样和转换，转换数据只会写入通道缓冲区（分别从1008h、1018h和1040h开始）。PIA缓冲区空间中的未使用部分可用于其他用途。用户应负责跟踪记录缓冲单元并防止数据覆盖。

24.3 寄存器

12位A/D转换器通过总计13个寄存器进行控制：

- AD1CON1至AD1CON5（寄存器24-1至寄存器24-5）
- AD1CHS（寄存器24-6）
- ANCFG（寄存器24-7）
- AD1CHITL（寄存器24-8）
- AD1CSSH和AD1CSSL（寄存器24-9和寄存器24-10）
- AD1CTMENH和AD1CTMENL（寄存器24-11和寄存器24-12）
- AD1DMBUF（未显示）——扩展缓冲区模式的16位转换缓冲区

表24-1： **PIA模式下的间接地址生成**

DMABL<2:0>	每条通道的缓冲区大小（字）	生成的偏移地址（低11位）	可用输入通道	允许的DMADSTn地址
000	1	000 00cc ccc0	32	xxxx xxxx xx00 0000
001	2	000 0ccc ccn0	32	xxxx xxxx x000 0000
010	4	000 cccc cnn0	32	xxxx xxxx 0000 0000
011	8	00c cccc nnn0	32	xxxx xxx0 0000 0000
100	16	0cc cccn nnn0	32	xxxx xx00 0000 0000
101	32	ccc ccnn nnn0	32	xxxx x000 0000 0000
110	64	ccc cnnn nnn0	16	xxxx x000 0000 0000
111	128	ccc nnnn nnn0	8	xxxx x000 0000 0000

图注： ccc = 通道编号（三到五位），n = 缓冲区基地址（零到七位），
x = 用户可为基地址定义的DMADSTn范围，0 = IA的DMADSTn掩码位

24.4 实现最佳A/D转换器性能

为了在保持精度的前提下获得最短的总转换时间（称作“吞吐量”），必须考虑多项因素。下面将详细介绍这些因素。

- AVDD相关性——如果AVDD电源 < 2.7V，则电荷泵使能位（PUMPEN，AD1CON3<13>）应置1。输入通道多路开关的电阻随AVDD变化（AVDD越低，内部开关电阻就越高）。电荷泵应提供较高的内部AVDD来保持尽可能低的开关电阻。
- TAD相关性——ADC时序由TAD（而非TCYC）驱动。正确选择TAD时间是获得最佳ADC吞吐量的关键所在。必须注意的是，总ADC吞吐量并不仅仅是SAR的“转换时间”，而是转换时间、采样时间以及内部同步逻辑的附加TAD延时三者的组合。
- TCYC和TAD之间的关系——TCYC和TAD之间不存在固定的1:1时序关系。从根本上说，最快的吞吐量由TAD（最小值）设置，而非TCYC。TAD时间由ADCS<7:0>位设置为TCYC的整数倍，具体倍数可编程。根据表32-25，TAD（最小值）时间大于专用ADC RC时钟发生器的4 MHz周期。因此，TAD必须为2 TCYC，以使用RC时钟获得最快的吞吐量。TAD（最小值）是3.597 MHz（而非4 MHz）的倍数。为了尽可能提高运行速度，TCYC必须为TAD（最小值）的倍数，因为ADCSx的值为整数。例如，如果使用标准的14.31818 MHz“彩色脉冲串”晶振，则TCYC为279.4 ns，与TAD（最小值）非常接近且ADC吞吐量达到最佳水平。以16 MHz运行实际上会降低吞吐量，因为250 ns的TCYC违反了TAD（最小值），TAD必须为500 ns。
- 源电阻（RS）驱动相关性——某些传感器具有较高的输出阻抗（> 2.5 kΩ）。如果RS较高，则将需要较长的采样时间以通过电阻路径对S/H电容充电（见图25-3）。最差的情况是AVSS到AVDD的满量程阶跃电压，采样电容在AVSS处。电容时间常量为（RS + RIC + RSS）（CHOLD），采样时间至少需要为6个时间常量（首选8个）。由于ADC逻辑时序以TAD为基准，因此在所有条件下，采样时间（TAD）都必须足以对CHOLD进行充电/放电。请勿假定一个TAD对于采样时间即足够，达到应用所需的精度可能需要更长的时间。CHOLD值为40 pF。

  关闭采样开关时，ADC输入引脚上存在少量的电荷。如果RS较高，则会产生1 LSB以上的直流误差。为了获得最佳结果，建议保持RS < 50Ω。此外，也可通过增加采样时间来减小误差（RS值为2 kΩ时，需要3 µs的采样时间来消除误差）。
- 计算吞吐量——ADC的吞吐量基于TAD。吞吐量的计算公式如下：

  吞吐量 = 1/(采样时间 + SAR 转换时间 + 时钟同步时间)

  其中：

  采样时间是针对应用计算出的TAD周期数。SAR转换时间是12 TAD（对于10位转换）和14 TAD（对于12位转换）。时钟同步时间是2.5 TAD（最差情况）。

  示例：对于12位ADC吞吐量，如果使用FRC = 8 MHz且采样时间为1 TAD，则使用8 MHz FRC意味着TCYC = 250 ns且要求：TAD = 2 TCYC = 500 ns。因此，吞吐量为：

  *吞吐量 = 1/(500 ns) + (14 * 500 ns) + (2.5 * 500 ns) = 114.28 KS/s*

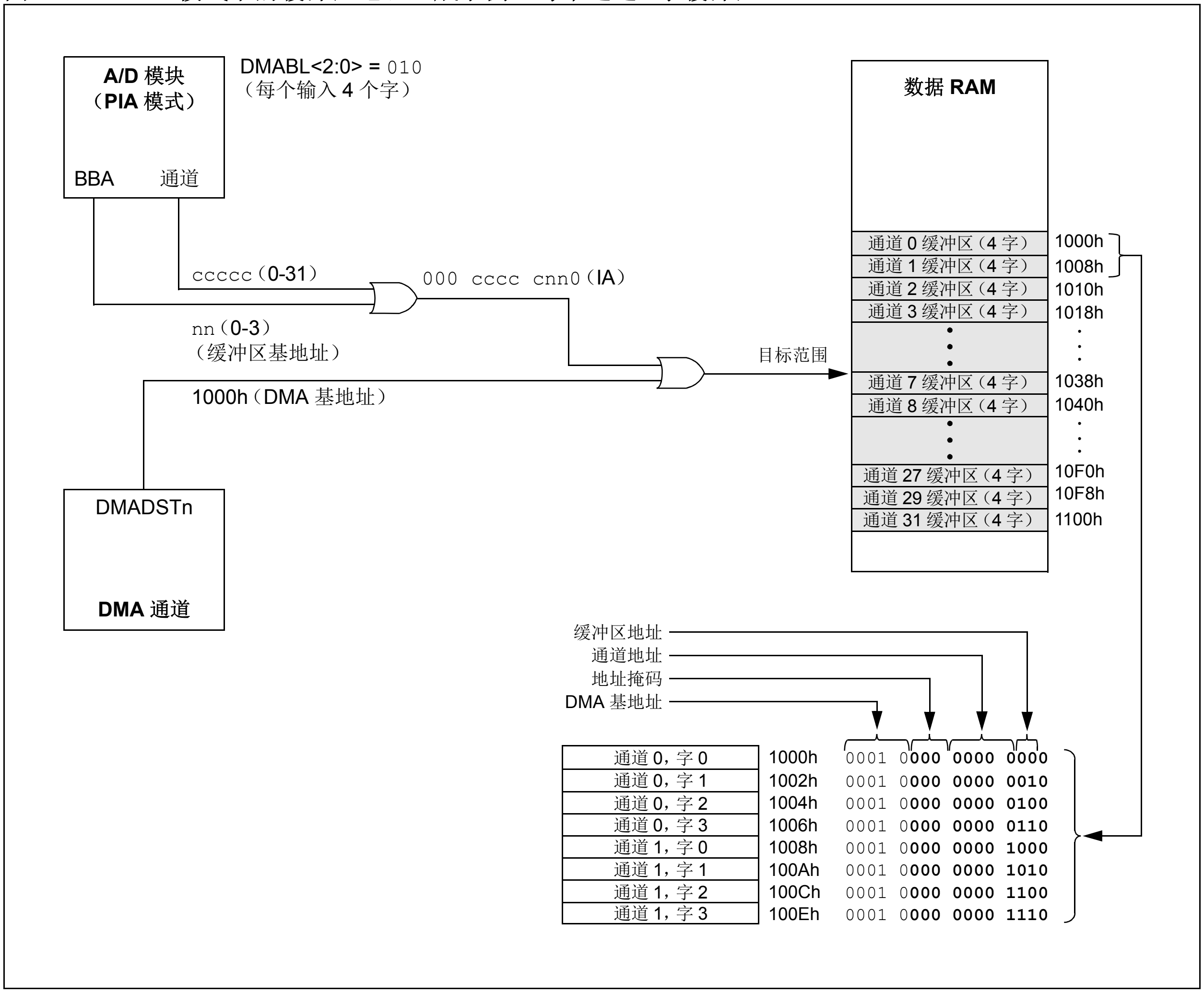
  请注意，时钟同步延时可能只有1.5 TAD，对应可产生121 Ks/s的吞吐量，但也会由于时序关系是异步的且未指定而无法得到保证。最差情况下，应使用2.5 TAD的时序来计算吞吐量。

  示例：某个传感器具有20 kΩ的输出阻抗。如果AVDD为3.0，则所需的最大采样时间将由如下公式确定：

  $$\text{采样时间} = 6 * (R_S + R_{IC} + R_{SS}) * C_{HOLD}$$
  $$= 6 * (20K + 250 + 350) * 40\ pF$$
  $$= 4.95\ \mu s$$

如果TAD = 500 ns，则需要4.95 µs/500 ns = 10 TAD的采样时间（对应于传感器输出上的满量程阶跃电压）。由于AVDD高于2.7V，因此RSS为350Ω。

图24-2： PIA模式下的缓冲区地址生成示例（每个通道4字缓冲区）



寄存器24-1： AD1CON1：A/D控制寄存器1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADON	—	ADSIDL	DMABM[(1)]	DMAEN	MODE12	FORM1	FORM0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0, HSC	R-0, HSC
SSRC3	SSRC2	SSRC1	SSRC0	—	ASAM	SAMP	DONE
bit 7							bit 0

图注：	C = 可清零位	U = 未实现位，读为0	
R = 可读位	W = 可写位	HSC = 硬件置1/清零位	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15 **ADON**：A/D工作模式位
1 = A/D转换器正在工作
0 = A/D转换器关闭

bit 14 **未实现**：读为0

bit 13 **ADSIDL**：A/D空闲模式停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 在空闲模式下模块继续工作

bit 12 **DMABM**：扩展DMA缓冲区模式选择位[(1)]
1 = 扩展缓冲区模式：缓冲区地址由DMADSTn寄存器定义
0 = PIA模式：缓冲区地址由DMA控制器和AD1CON4<2:0>定义

bit 11 **DMAEN**：扩展DMA/缓冲区使能位
1 = 使能扩展DMA和缓冲区功能
0 = 禁止扩展功能

bit 10 **MODE12**：12位A/D工作模式位
1 = 12位A/D工作
0 = 10位A/D工作

bit 9-8 **FORM<1:0>**：数据输出格式位（见以下格式）
11 = 小数结果，有符号，左对齐
10 = 绝对小数结果，无符号，左对齐
01 = 十进制结果，有符号，右对齐
00 = 绝对十进制结果，无符号，右对齐

bit 7-4 **SSRC<3:0>**：采样时钟源选择位
0000 = SAMP由软件清零
0001 = INT0
0010 = Timer3
0100 = CTMU触发信号
0101 = Timer1（在休眠模式期间不触发）
0110 = Timer1（在休眠模式期间可能触发）
0111 = 自动转换模式

bit 3 **未实现**：读为0

bit 2 **ASAM**：A/D采样自动启动位
1 = 最后一次转换结束后立即开始采样；SAMP位自动置1
0 = 手动将SAMP位置1时开始采样

注 1： 只有在扩展DMA和缓冲区功能可用（DMAEN = 1）时才可使用该位。

寄存器24-1： AD1CON1：A/D控制寄存器1（续）

bit 1	**SAMP**：A/D采样使能位 1 = A/D采样/保持放大器正在采样 0 = A/D采样/保持放大器正在保持
bit 0	**DONE**：A/D转换状态位 1 = A/D转换已完成 0 = A/D转换尚未开始或在进行

注 1： 只有在扩展DMA和缓冲区功能可用（DMAEN = 1）时才可使用该位。

寄存器24-2： AD1CON2：A/D控制寄存器2

R/W-0	R/W-0	R/W-0	r-0	R/W-0	R/W-0	U-0	U-0
PVCFG1	PVCFG0	NVCFG0	—	BUFREGEN	CSCNA	—	—
bit 15							bit 8

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUFS	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS
bit 7							bit 0

图注：	r = 保留位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15-14 **PVCFG<1:0>**：A/D转换器正参考电压配置位
1x = 未实现，不要使用
01 = 外部VREF+
00 = AVDD

bit 13 **NVCFG0**：A/D转换器负参考电压配置位
1 = 外部VREF-
0 = AVSS

bit 12 **保留**：保持为0

bit 11 **BUFREGEN**：A/D缓冲寄存器使能位
1 = 转换结果装入根据所转换通道确定的缓冲单元
0 = A/D结果缓冲区视为FIFO缓冲区

bit 10 **CSCNA**：采样多路开关A期间CH0+的扫描输入选择位
1 = 扫描输入
0 = 不扫描输入

bit 9-8 **未实现**：读为0

bit 7 **BUFS**：缓冲区填充状态位
当DMAEN = 1且DMABM = 1时：
1 = A/D当前正从[缓冲区起始地址 + (缓冲区大小/2)]到[缓冲区起始地址 + (缓冲区大小 – 1)]填充目标缓冲区。用户应访问位于[缓冲区起始地址]到[缓冲区起始地址 + (缓冲区大小/2) – 1]的数据。
0 = A/D当前正从[缓冲区起始地址]到[缓冲区起始地址 + (缓冲区大小/2) – 1]填充目标缓冲区。用户应访问位于[缓冲区起始地址 + (缓冲区大小/2)]到[缓冲区起始地址 + (缓冲区大小 – 1)]的数据。
当DMAEN = 0时：
1 = A/D当前正在填充ADC1BUF13-ADC1BUF25，用户应访问ADC1BUF0-ADC1BUF12中的数据。
0 = A/D当前正在填充ADC1BUF0-ADC1BUF12，用户应访问ADC1BUF13-ADC1BUF25中的数据。

寄存器24-2： AD1CON2：A/D控制寄存器2（续）

bit 6-2 **SMPI<4:0>**：中断采样速率/DMA递增速率选择位

当DMAEN = 1且DMABM = 0时：

11111 = 完成第32次采样/转换操作后递增DMA地址
11110 = 完成第31次采样/转换操作后递增DMA地址
•
•
•
00001 = 完成第2次采样/转换操作后递增DMA地址
00000 = 完成每次采样/转换操作后递增DMA地址

当DMAEN = 1且DMABM = 1时：

11111 = 完成第32次采样/转换操作后复位DMA偏移
11110 = 完成第31次采样/转换操作后复位DMA偏移
•
•
•
00001 = 完成第2次采样/转换操作后复位DMA偏移
00000 = 完成每次采样/转换操作后复位DMA偏移

当DMAEN = 0时：

11111 = 每完成32次采样转换时产生中断
11110 = 每完成31次采样转换时产生中断
•
•
•
00001 = 每完成2次采样转换时产生中断
00000 = 每完成1次采样转换时产生中断

bit 1 **BUFM**：缓冲区填充模式选择位

1 = 在第一次中断时从ADC1BUF0开始填充缓冲区，而在下一次中断时从ADC1BUF13开始填充缓冲区
0 = 总是从ADC1BUF0开始填充缓冲区

bit 0 **ALTS**：交替输入采样模式选择位

1 = 在第一次采样时使用采样多路开关A选择的输入通道，而在下一次采样时使用采样多路开关B选择的输入通道
0 = 总是使用采样多路开关A选择的输入通道

寄存器24-3：　AD1CON3：A/D控制寄存器3

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC[1]	EXTSAM	PUMPEN[2]	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
bit 7							bit 0

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

bit 15　**ADRC：**A/D转换时钟源位[1]
1 = 专用ADC RC时钟发生器（4 MHz标称值）。
0 = 时钟由系统时钟产生

bit 14　**EXTSAM：**扩展采样时间位
1 = A/D在SAMP = 0之后仍然在采样
0 = A/D完成采样

bit 13　**PUMPEN：**电荷泵使能位[2]
1 = 使能开关的电荷泵
0 = 禁止开关的电荷泵

bit 12-8　**SAMC<4:0>：**自动采样时间选择位
11111 = 31 T_{AD}
• • •
00001 = 1 T_{AD}
00000 = 0 T_{AD}

bit 7-0　**ADCS<7:0>：**A/D转换时钟选择位
11111111 = 256 • T_{CY} = T_{AD}
• • •
00000001 = 2 • T_{CY} = T_{AD}
00000000 = T_{CY} = T_{AD}

注 1： 选择内部ADC RC时钟时，要求ADCSx为1或更大值。当ADRC = 1将违反T_{AD}（最小值）规范时，设置ADCSx = 0。

2： 如果AV_{DD} < 2.7V，则用户应使能电荷泵。如果禁止电荷泵，则由于MUX内阻的增大，将需要更长的采样时间。

寄存器24-4：　AD1CON4：A/D控制寄存器4

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DMABL<2:0>(1)		
bit 7							bit 0

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

bit 15-3　**未实现：**读为0

bit 2-0　**DMABL<2:0>：**DMA缓冲区大小选择位(1)
111 = 为每个模拟输入分配128字缓冲区
110 = 为每个模拟输入分配64字缓冲区
101 = 为每个模拟输入分配32字缓冲区
100 = 为每个模拟输入分配16字缓冲区
011 = 为每个模拟输入分配8字缓冲区
010 = 为每个模拟输入分配4字缓冲区
001 = 为每个模拟输入分配2字缓冲区
000 = 为每个模拟输入分配1字缓冲区

注 1： 仅当AD1CON1<11> = 1且AD1CON1<12> = 0时使用DMABL<2:0>位；否则忽略其值。

寄存器24-5： AD1CON5：A/D控制寄存器5

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
ASEN	LPEN	CTMREQ	BGREQ	—	—	ASINT1	ASINT0
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	WM1	WM0	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 = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15 **ASEN**：自动扫描使能位
1 = 使能自动扫描
0 = 禁止自动扫描

bit 14 **LPEN**：低功耗使能位
1 = 扫描后使能低功耗
0 = 扫描后使能全功耗

bit 13 **CTMREQ**：CTMU请求位
1 = 当A/D使能并处于活动状态时使能CTMU
0 = 不通过A/D使能CTMU

bit 12 **BGREQ**：带隙请求位
1 = 当A/D使能并处于活动状态时使能带隙
0 = 不通过A/D使能带隙

bit 11-10 **未实现**：读为0

bit 9-8 **ASINT<1:0>**：自动扫描（阈值检测）中断模式位
11 = 在阈值检测序列完成且发生有效比较后产生中断
10 = 在发生有效比较后产生中断
01 = 在阈值检测序列完成后产生中断
00 = 不产生中断

bit 7-4 **未实现**：读为0

bit 3-2 **WM<1:0>**：写模式位
11 = 保留
10 = 仅自动比较（不保存转换结果，但在发生由CMx和ASINTx位定义的有效匹配时产生中断）
01 = 转换并保存（当发生由CMx位定义的匹配时，转换结果保存到由寄存器位决定的存储单元中）
00 = 传统操作（转换数据保存到由缓冲寄存器位决定的存储单元中）

bit 1-0 **CM<1:0>**：比较模式位
11 = 窗外模式：如果转换结果超出相应缓冲区对定义的窗口，则发生有效匹配
10 = 窗内模式：如果转换结果处于相应缓冲区对定义的窗口内，则发生有效匹配
01 = 大于模式：如果结果大于相应缓冲寄存器中的值，则发生有效匹配
00 = 小于模式：如果结果小于相应缓冲寄存器中的值，则发生有效匹配

寄存器24-6： **AD1CHS：A/D采样选择寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NB2	CH0NB1	CH0NB0	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NA2	CH0NA1	CH0NA0	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0
bit 7							bit 0

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

bit 15-13 **CH0NB<2:0>：**采样多路开关B的通道0的反相输入选择位
1xx = 未实现
01x = 未实现
001 = 未实现
000 = AVSS

bit 12-8 **CH0SB<4:0>：**采样多路开关B的通道0的同相输入选择位
11110 = AVDD(1)
11101 = AVSS(1)
11100 = 带隙参考（VBG）(1)
10000-11011 = 保留
01111 = 未连接外部通道（用于CTMU）
01110 = 未连接外部通道（用于CTMU温度传感器）
01101 = AN13
01100 = AN12
01011 = AN11
01010 = AN10
01001 = AN9
01000 = AN8
00111 = AN7
00110 = AN6
00101 = AN5
00100 = AN4
00011 = AN3
00010 = AN2
00001 = AN1
00000 = AN0

bit 7-5 **CH0NA<2:0>：**采样多路开关A的通道0的反相输入选择位
与CHONB<2:0>定义相同。

bit 4-0 **CH0SA<4:0>：**采样多路开关A的通道0的同相输入选择位
与CHOSB<4:0>定义相同。

注 1： 这些输入通道没有相应的存储器映射结果缓冲区。

寄存器24-7： ANCFG：A/D带隙参考配置寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	VBGEN3[(1)]	VBGEN2[(1)]	VBGEN1[(1)]
bit 7							bit 0

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

bit 15-3 **未实现：**读为0

bit 2 **VBGEN3：**A/D带隙参考使能位[(1)]
1 = 使能带隙参考
0 = 禁止带隙参考

bit 1 **VBGEN2：**CTMU和比较器带隙参考使能位[(1)]
1 = 使能带隙参考
0 = 禁止带隙参考

bit 0 **VBGEN1：**VREG、BOR、HLVD、FRC、NVM和A/D升压带隙参考使能位[(1)]
1 = 使能带隙参考
0 = 禁止带隙参考

注 1： 当模块请求带隙参考电压时，将在一段短暂的启动时间后自动使能该参考电压。用户可在使能请求带隙参考电压的模块之前，使用ANCFG寄存器手动使能带隙参考电压，以避免这段启动时间（约1 ms）。

寄存器24-8： AD1CHITL：A/D扫描比较命中寄存器（低位字）

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHH<13:8>(1)					
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHH<7:0>							
bit 7							bit 0

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

bit 15-14 **未实现：**读为0

bit 13-0 **CHH<13:0>：**A/D比较命中位(1)

如果CM<1:0> = 11：
1 = A/D结果缓冲区n已写入数据或已发生匹配
0 = A/D结果缓冲区n尚未写入数据

对于CM<1:0>的所有其他值：
1 = A/D结果通道n上发生了匹配
0 = A/D结果通道n上未发生匹配

注 1： 28引脚器件上未实现CHH<13:10>位。

寄存器24-9： AD1CSSH：A/D输入扫描选择寄存器（高位字）

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CSS<28:24>				
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15-13 **未实现：** 读为0

bit 12-8 **CSS<28:24>：** A/D输入扫描选择位
1 = 输入扫描时包含相应的通道
0 = 输入扫描时跳过通道

bit 7-0 **未实现：** 读为0

寄存器24-10： AD1CSSL：A/D输入扫描选择寄存器（低位字）

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSS<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSS<7:0>							
bit 7							bit 0

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

bit 15-0 **CSS<15:0>：** A/D输入扫描选择位
1 = 输入扫描时包含相应的通道
0 = 输入扫描时跳过通道

寄存器24-11： AD1CTMENH：A/D CTMU使能寄存器（高位字）

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
—	CTMEN<30:28>			—	—	CTMEN<25:24>	
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMEN<23:16>(1)							
bit 7							bit 0

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

bit 15 **未实现：**读为0

bit 14-12 **CTMEN<30:28>：**转换期间CTMU使能位
1 = 在转换期间使能CTMU并连接到选定的通道
0 = CTMU不连接到该通道

bit 11-10 **未实现：**读为0

bit 9-0 **CTMEN<25:16>：**转换期间CTMU使能位(1)
1 = 在转换期间使能CTMU并连接到选定的通道
0 = CTMU不连接到该通道

注 1： 64引脚器件上不具有CTMEN<23:16>位。

寄存器24-12： AD1CTMENL：A/D CTMU使能寄存器（低位字）

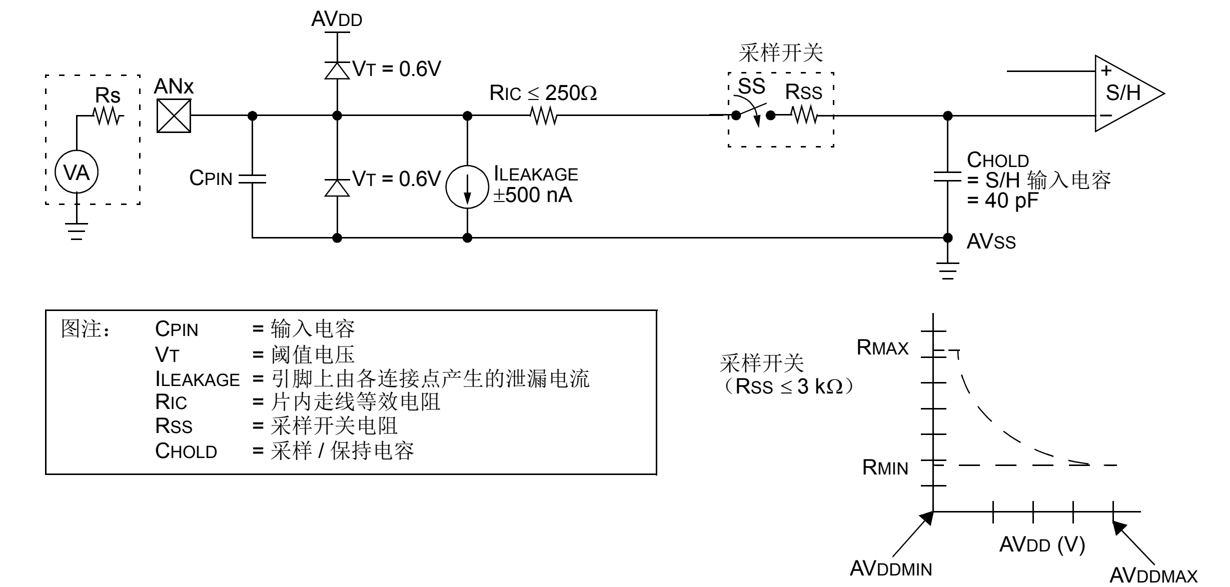
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMEN<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMEN<7:0>							
bit 7							bit 0

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

bit 15-0 **CTMEN<15:0>：**转换期间CTMU使能位
1 = 在转换期间使能CTMU并连接到选定的通道
0 = CTMU不连接到该通道

图24-3： 12位A/D转换器模拟输入模型



注：C_{PIN}值取决于器件封装，未经测试。如果$R_S \le 2.5\ k\Omega$，可忽略C_{PIN}的影响。

公式24-1： A/D转换时钟周期

$$T_{AD} = T_{CY}(ADCS + 1)$$

$$ADCS = \frac{T_{AD}}{T_{CY}} - 1$$

注：基于 $T_{CY} = 2/F_{OSC}$；打盹模式和 PLL 被禁止。

图24-4：　12位A/D传递函数

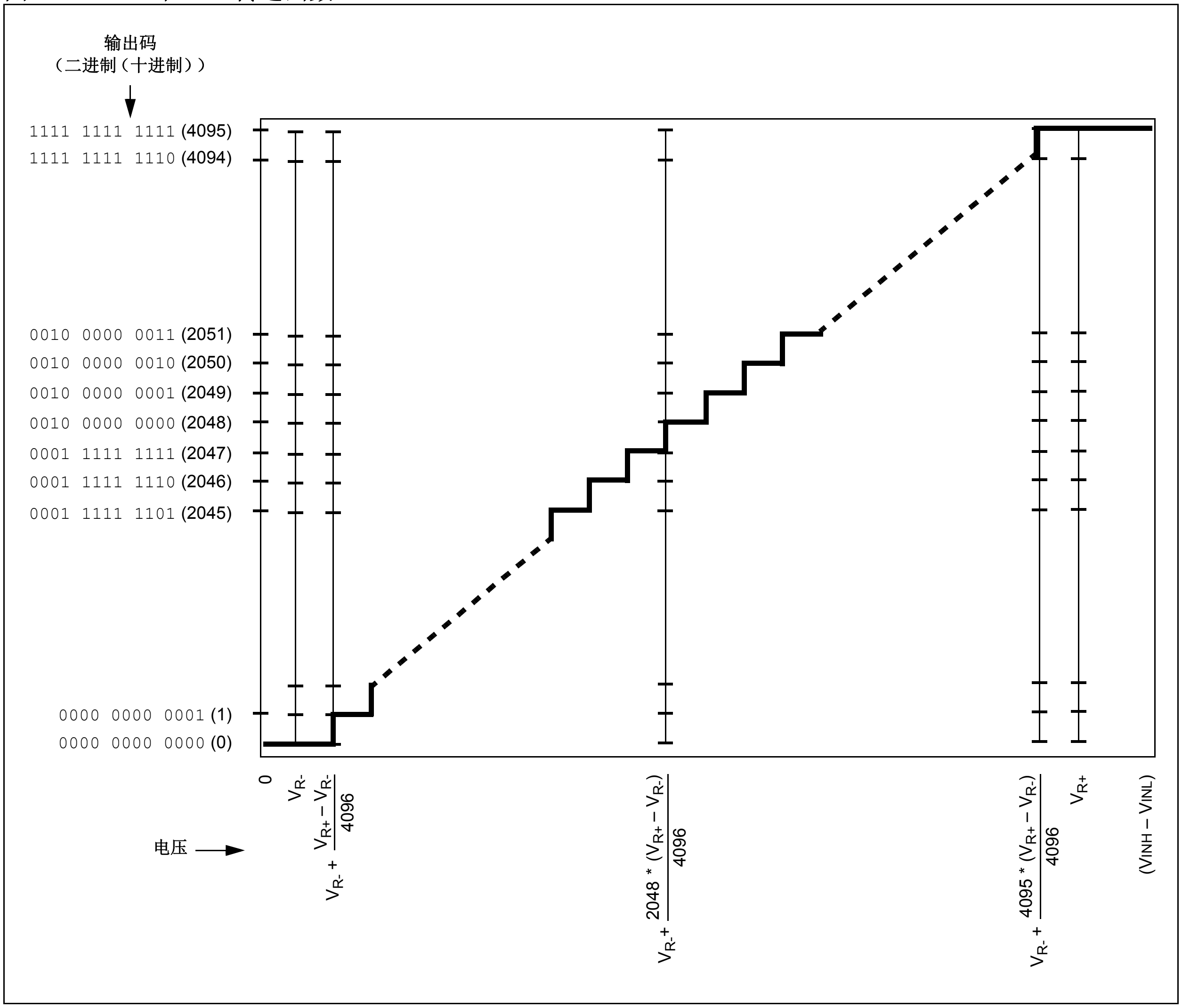
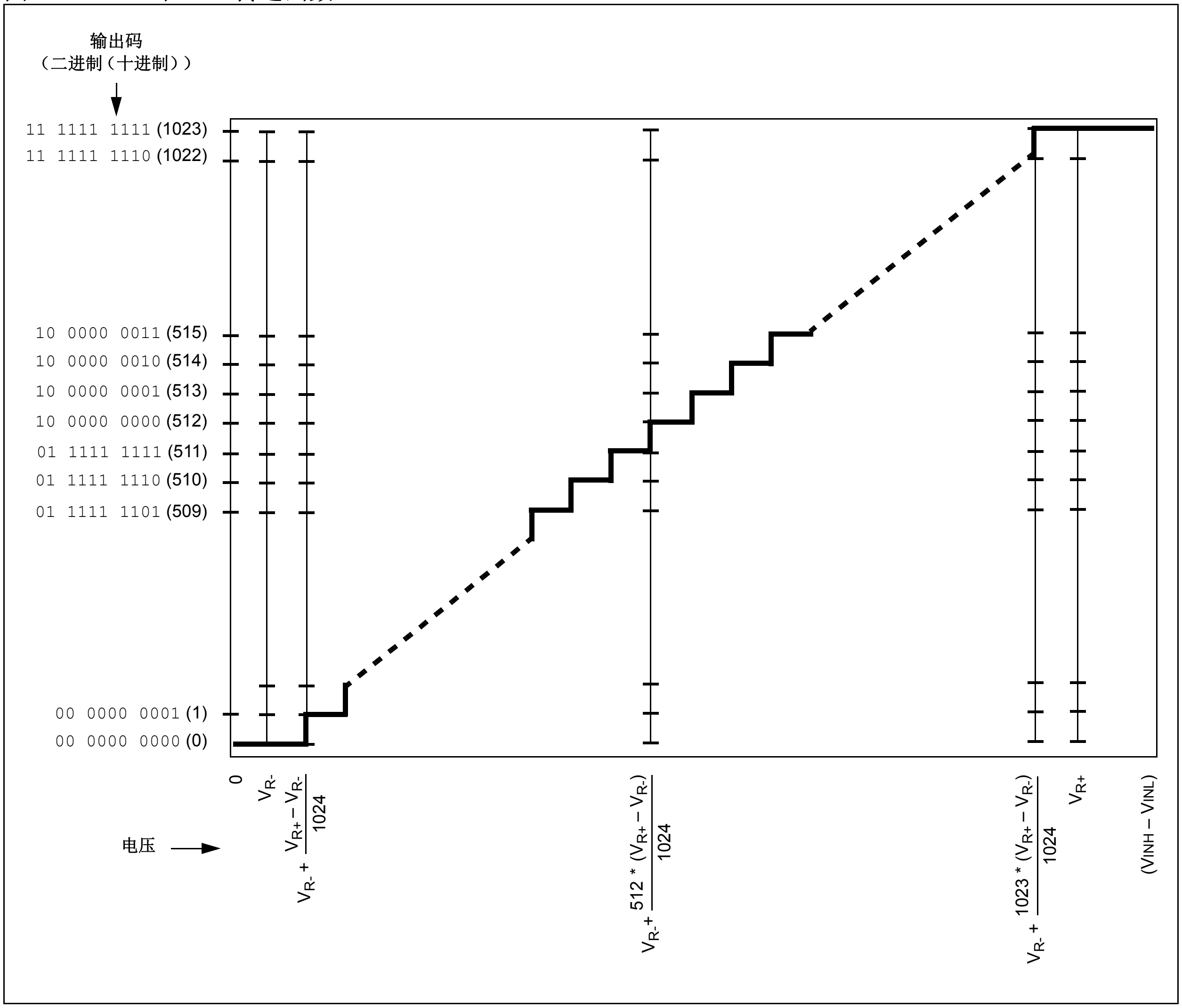


图24-5： 10位A/D传递函数



注：

25.0 三比较器模块

> **注：** 本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的**“可扩展的比较器模块”**（DS39734），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

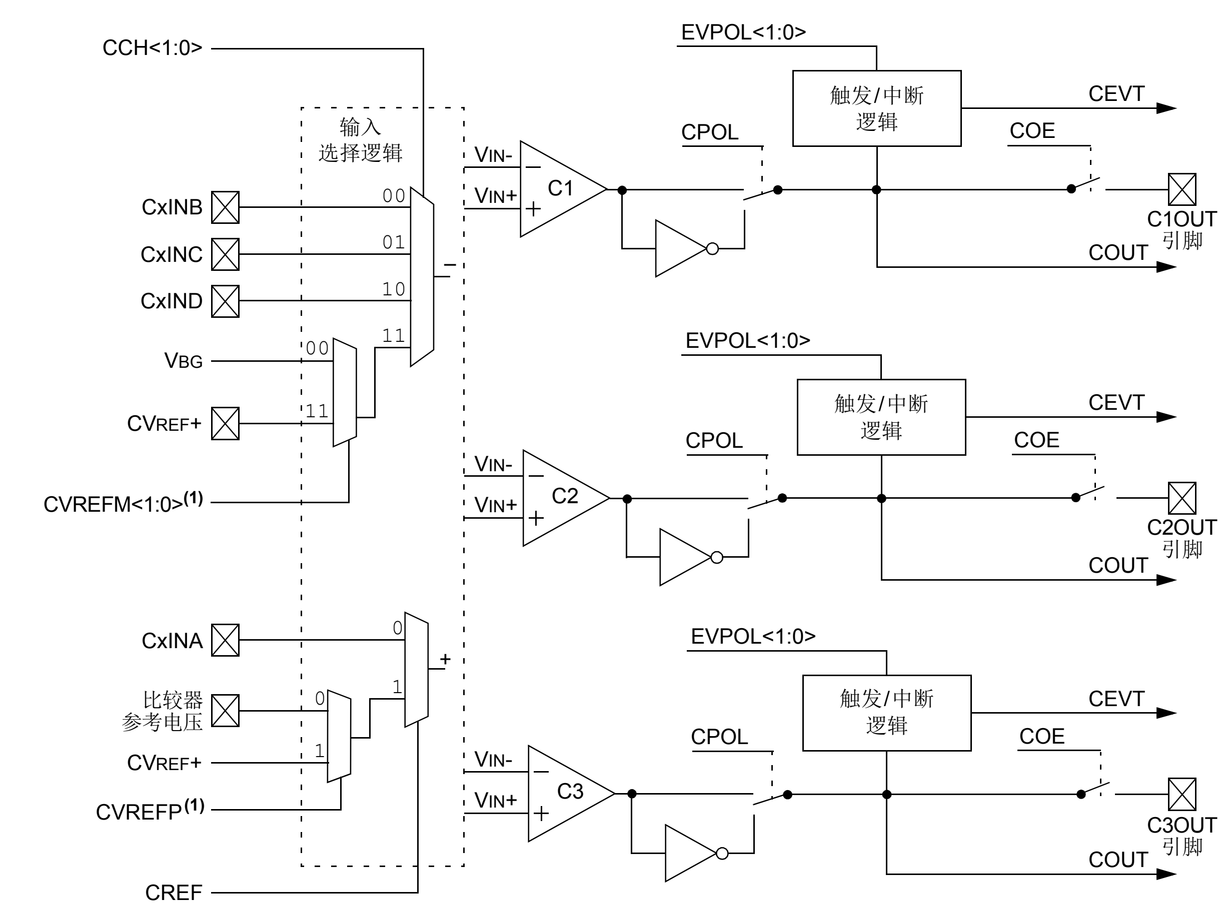
三比较器模块提供三个双输入比较器。比较器的输入可配置为使用五个外部模拟输入（CxINA、CxINB、CxINC、CxIND和CVREF+）之一，参考电压输入可以来自其中一个内部带隙参考或比较器参考电压发生器（VBG和CVREF）。

比较器输出可直接连接到CxOUT引脚。当对应的COE位等于1时，I/O引脚逻辑使得引脚输出不同步的比较器输出信号。

模块的简化框图如图25-1所示。图25-2到图25-4中给出了各种可能的比较器配置图。

每个比较器都有自己的控制寄存器CMxCON（寄存器25-1），用于使能和配置其操作。所有三个比较器的输出和事件状态都在CMSTAT寄存器（寄存器25-2）中给出。

图25-1：　三比较器模块框图



注　1：　有关该位的详细信息，请参见CVRCON寄存器（寄存器26-1）。

图25-2： CREF = 0时的各个比较器配置

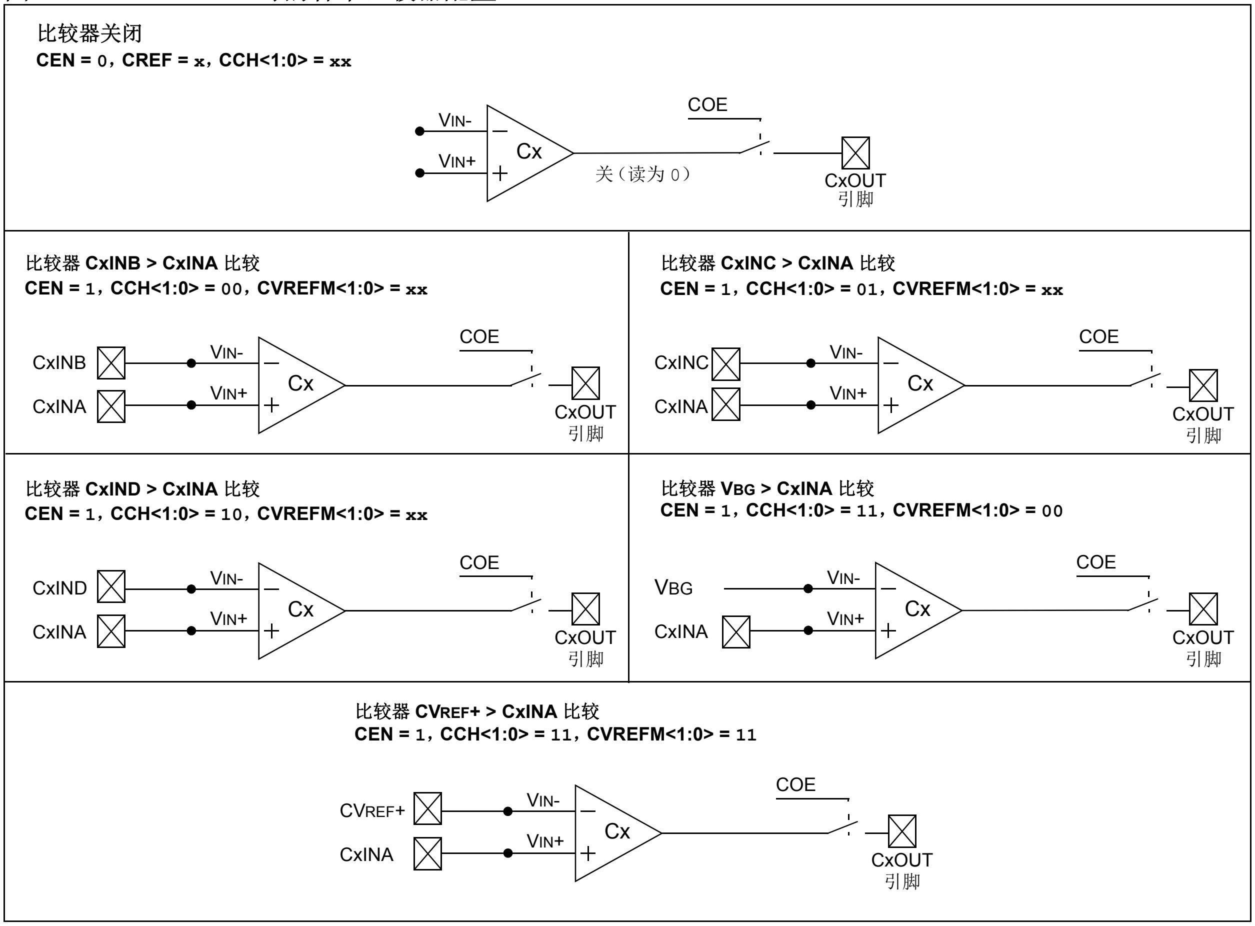


图25-3： CREF = 1且CVREFP = 0时的各个比较器配置

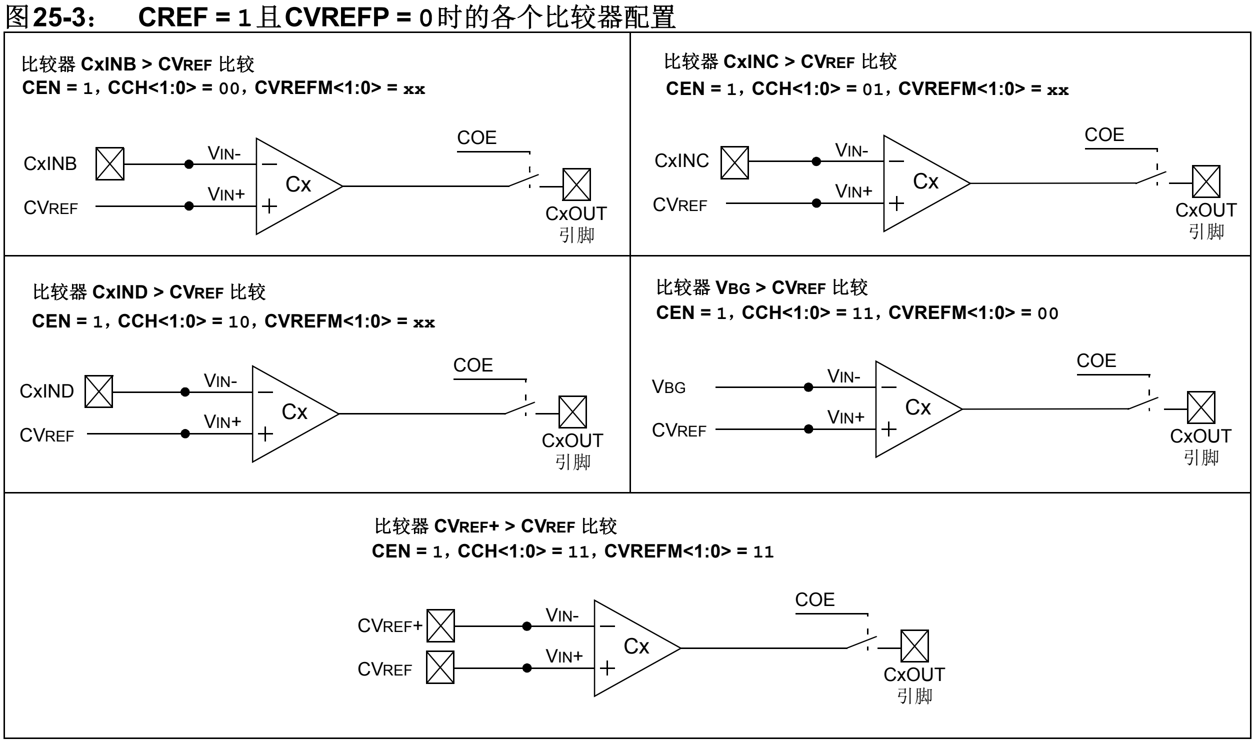
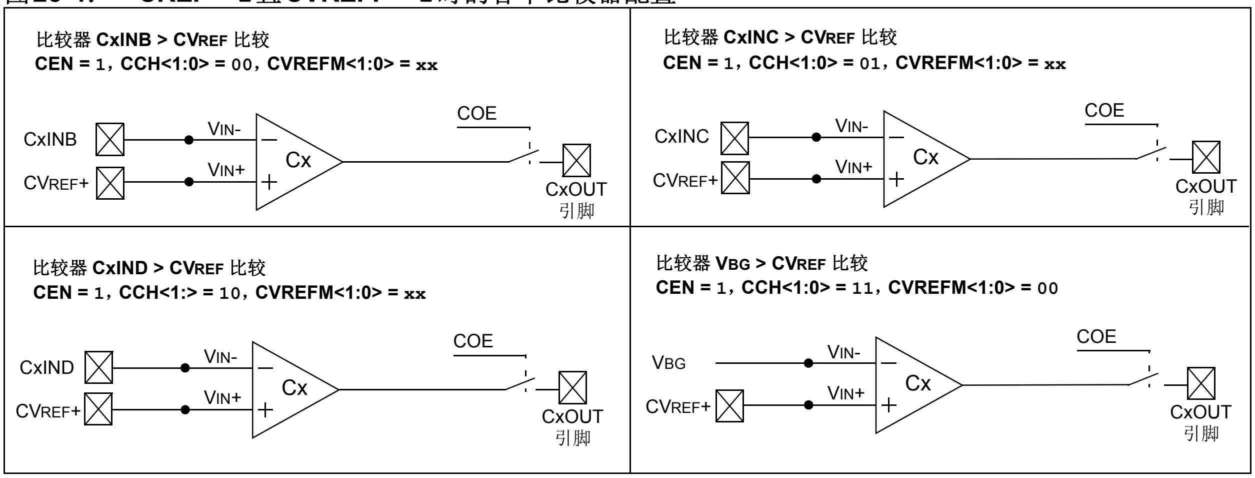


图25-4： CREF = 1且CVREFP = 1时的各个比较器配置



寄存器25-1：　　CMxCON：比较器x控制寄存器（比较器1至3）

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0, HS	R-0, HSC
CEN	COE	CPOL	—	—	—	CEVT	COUT
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0
bit 7							bit 0

图注：	HS = 硬件置1位	HSC = 硬件置1/清零位	
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15　**CEN：**比较器使能位
1 = 使能比较器
0 = 禁止比较器

bit 14　**COE：**比较器输出使能位
1 = 在CxOUT引脚上提供比较器输出
0 = 比较器输出仅供内部使用

bit 13　**CPOL：**比较器输出极性选择位
1 = 比较器输出反相
0 = 比较器输出不反相

bit 12-10　**未实现：**读为0

bit 9　**CEVT：**比较器事件位
1 = 发生由EVPOL<1:0>定义的比较器事件；该位清零之前禁止后续触发和中断
0 = 未产生比较器事件

bit 8　**COUT：**比较器输出位
<u>当CPOL = 0时：</u>
1 = VIN+ > VIN-
0 = VIN+ < VIN-
<u>当CPOL = 1时：</u>
1 = VIN+ < VIN-
0 = VIN+ > VIN-

bit 7-6　**EVPOL<1:0>：**触发/事件/中断极性选择位
11 = 比较器输出的任何变化导致发生触发/事件/中断（当CEVT = 0时）
10 = 比较器输出的以下跳变导致发生触发/事件/中断：
<u>如果CPOL = 0（极性不反相）：</u>
仅从高电平到低电平的跳变。
<u>如果CPOL = 1（极性反相）：</u>
仅从低电平到高电平的跳变。
01 = 比较器输出的以下跳变导致发生触发/事件/中断：
<u>如果CPOL = 0（极性不反相）：</u>
仅从低电平到高电平的跳变。
<u>如果CPOL = 1（极性反相）：</u>
仅从高电平到低电平的跳变。
00 = 禁止发生触发/事件/中断

bit 5　**未实现：**读为0

寄存器25-1： CMxCON：比较器x控制寄存器（比较器1至3）（续）

bit 4 **CREF：**比较器参考电压选择位（同相输入）
1 = 同相输入连接到内部CVREF电压
0 = 同相输入连接到CxINA引脚

bit 3-2 **未实现：**读为0

bit 1-0 **CCH<1:0>：**比较器通道选择位
11 = 比较器反相输入连接到CVRCON寄存器中的CVREFM<1:0>位指定的内部可选参考电压
10 = 比较器反相输入连接到CxIND引脚
01 = 比较器反相输入连接到CxINC引脚
00 = 比较器反相输入连接到CxINB引脚

寄存器25-2： CMSTAT：比较器模块状态寄存器

R/W-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
CMIDL	—	—	—	—	C3EVT	C2EVT	C1EVT
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
—	—	—	—	—	C3OUT	C2OUT	C1OUT
bit 7							bit 0

图注：	HSC = 硬件置1/清零位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15 **CMIDL：**比较器空闲模式停止位
1 = 当器件进入空闲模式时，所有比较器停止工作
0 = 所有使能的比较器在空闲模式下继续工作

bit 14-11 **未实现：**读为0

bit 10 **C3EVT：**比较器3事件状态位（只读）
显示比较器3的当前事件状态（CM3CON<9>）。

bit 9 **C2EVT：**比较器2事件状态位（只读）
显示比较器2的当前事件状态（CM2CON<9>）。

bit 8 **C1EVT：**比较器1事件状态位（只读）
显示比较器1的当前事件状态（CM1CON<9>）。

bit 7-3 **未实现：**读为0

bit 2 **C3OUT：**比较器3输出状态位（只读）
显示比较器3的当前输出（CM3CON<8>）。

bit 1 **C2OUT：**比较器2输出状态位（只读）
显示比较器2的当前输出（CM2CON<8>）。

bit 0 **C1OUT：**比较器1输出状态位（只读）
显示比较器1的当前输出（CM1CON<8>）。

注：

26.0 比较器参考电压

> **注：** 本数据手册总结了该 PIC24F 系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》的 **“双比较器模块”**（DS39710），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM 中的信息。

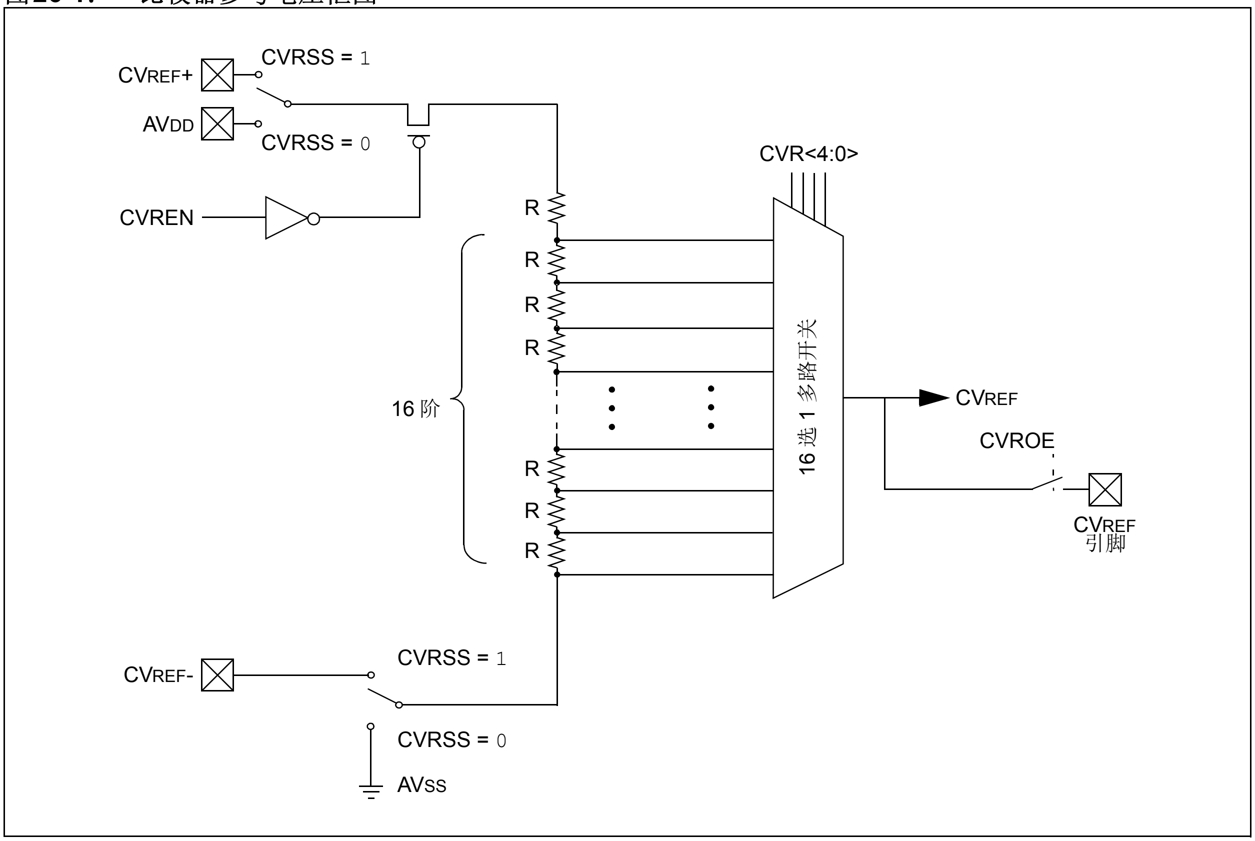
26.1 配置比较器参考电压

参考电压模块由CVRCON 寄存器（寄存器 26-1）控制，它能提供两种范围的输出电压，每种范围都具有 16 种不同的电压。这两种电压范围的主要区别在于 CVREF 值选择位（CVR<4:0>）所选的步长不同，其中一个范围提供更高的分辨率。

比较器参考电压模块的电压源可以来自VDD和VSS或外部VREF+和VREF-。电压源通过CVRSS位（CVRCON<5>）选择。

在改变CVREF输出值时，必须考虑到比较器参考电压的稳定时间。

图26-1： 比较器参考电压框图



寄存器26-1： CVRCON：比较器参考电压控制寄存器

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CVREFP	CVREFM1	CVREFM0
bit 15							bit 8

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	CVRSS	CVR4	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 = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15-11 **未实现**：读为0

bit 10 **CVREFP**：比较器参考电压选择位（仅当CREF为1时有效）
1 = CVREF+用作比较器的参考电压
0 = 此模块内的CVR<4:0>（5位DAC）为比较器提供参考电压

bit 9-8 **CVREFM<1:0>**：比较器带隙参考源选择位（仅当CCH<1:0> = 11时有效）
00 = 提供带隙电压作为比较器的输入
01 = 保留
10 = 保留
11 = 提供CVREF+作为比较器的输入

bit 7 **CVREN**：比较器参考电压使能位
1 = CVREF电路上电
0 = CVREF电路掉电

bit 6 **CVROE**：比较器VREF输出使能位
1 = CVREF电压输出到CVREF引脚
0 = CVREF电压与CVREF引脚断开

bit 5 **CVRSS**：比较器VREF源选择位
1 = 比较器参考源CVRSRC = CVREF+ – CVREF-
0 = 比较器参考源CVRSRC = AVDD – AVSS

bit 4-0 **CVR<4:0>**：比较器VREF值选择位（0 ≤ CVR<4:0> ≤ 31）
CVRSS = 1时：
CVREF = (CVREF-) + (CVR<4:0>/32) • (CVREF+ – CVREF-)
CVRSS = 0时：
CVREF = (AVSS) + (CVR<4:0>/32) • (AVDD – AVSS)

27.0 充电时间测量单元（CTMU）

> **注：** 本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。有关充电时间测量单元的更多信息，请参见《dsPIC33/PIC24系列参考手册》的**"带阈值检测功能的充电时间测量单元（CTMU）及CTMU操作"**（DS30009743），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

充电时间测量单元（Charge Time Measurement Unit，CTMU）是一个灵活的模拟模块，它提供充电测量、脉冲源之间的精确时间差测量以及异步脉冲生成功能。它的主要特性包括：

- 十三个外部边沿输入触发源
- 每个边沿源的极性控制
- 边沿顺序控制
- 边沿电平或边沿跳变响应控制
- 时间测量分辨率为1纳秒
- 适合电容测量的精确电流源

CTMU可与其他片上模拟模块一起，用于精确测量时间、电容、电容的相对变化，或生成独立于系统时钟的输出脉冲。CTMU模块是与电容式触摸传感器接口的理想选择。

CTMU可通过三个寄存器进行控制：CTMUCON1L、CTMUCON1H和CTMUCON2L。CTMUCON1L用于使能模块、控制CTMU的工作模式、控制边沿顺序、选择电流源的电流范围以及微调电流。CTMUCON1H用于控制边沿源选择和边沿源极性选择。CTMUCON2L寄存器用于选择电流放电源。

27.1 测量电容

CTMU模块通过在两个单独的输入通道上产生输出脉冲（脉冲宽度等于边沿事件之间的时间间隔）来测量电容。这两个输入通道上的脉冲边沿事件源有四种选择：两个内部外设模块（OC1和Timer1）和最多13个外部引脚（CTEDG1到CTEDG13）。该脉冲和该模块的高精度电流源一起使用，可根据以下关系计算电容：

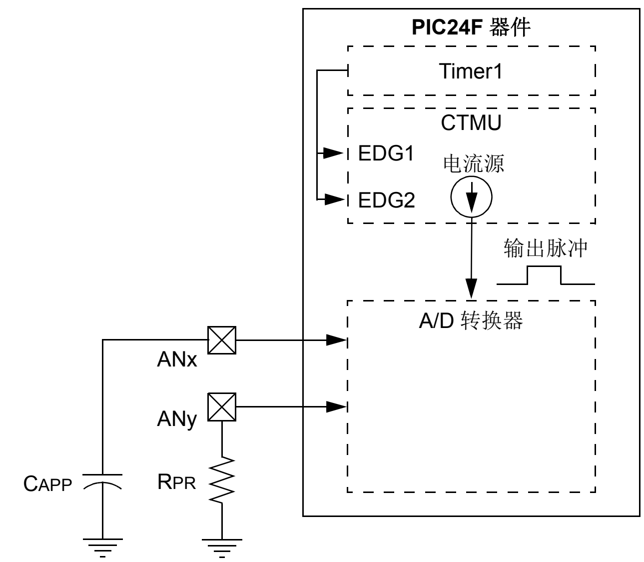
公式27-1：

$$I = C \bullet \frac{dV}{dT}$$

如需测量电容，A/D转换器在CTMU输出的脉冲信号之后对它的其中一个输入通道采样外部电容（C_{APP}）。由另一个A/D通道上的高精度电阻（R_{PR}）提供电流源校准。脉冲信号结束后，转换器确定电容上的电压。电容的实际计算在应用中通过软件执行。

图27-1给出了电容测量使用的外部连接以及此应用中CTMU和A/D模块的关系。此示例还展示了来自Timer1的边沿事件，但也可以选择使用内部边沿源的其他配置。有关使用CTMU模块测量电容和时间的详细说明，请参见《dsPIC33/PIC24系列参考手册》的**"带阈值检测功能的充电时间测量单元（CTMU）及CTMU操作"**（DS30009743）。

图27-1： 电容测量的典型连接和内部配置



27.2 测量时间/将电流源连接至A/D输入引脚

对脉冲宽度的时间测量也可类似地执行，使用A/D模块的内部电容（CAD）和高精度电阻校准电流。图27-2显示了时间测量使用的外部连接以及此应用中CTMU和A/D模块的关系。此示例还显示了来自外部CTEDx引脚的边沿事件，但也可以选择使用内部边沿源的其他配置。

通过将TGEN位（CTMUCON1L<12>）清零使能该模式。在采样开关后，电流源连接至A/D的输入。因此，A/D位SAMP必须置1，以便通过通道选择MUX将电流源连接至所需引脚。

27.3 脉冲生成和延时

CTMU模块也可生成边沿与器件的系统时钟异步的输出脉冲。更明确地说，它可以产生具有可编程延时的脉冲，自边沿事件输入至模块后经过这段延时，脉冲才会输出。

当通过将TGEN位（CTMUCON1<12>）置1将模块配置为脉冲生成延时时，内部电流源连接到比较器2的B输入。将电容（CDELAY）连接到比较器2的引脚C2INB，且比较器参考电压CVREF连接到C2INA。CVREF随后被配置为特定跳变点。当检测到边沿事件时，模块开始对CDELAY充电。当CDELAY充电到超过CVREF跳变点时，在CTPLS上输出脉冲信号。脉冲延时的时间长度由CDELAY和CVREF跳变点的值决定。

图27-3给出了脉冲生成的外部连接，以及所需的不同模拟模块之间的关系。CTED1显示为输入脉冲源时，有其他选项可用。关于使用CTMU模块生成脉冲的详细说明，请参见《dsPIC33/PIC24系列参考手册》。

图27-2： 时间测量（TGEN = 0）的典型连接和内部配置

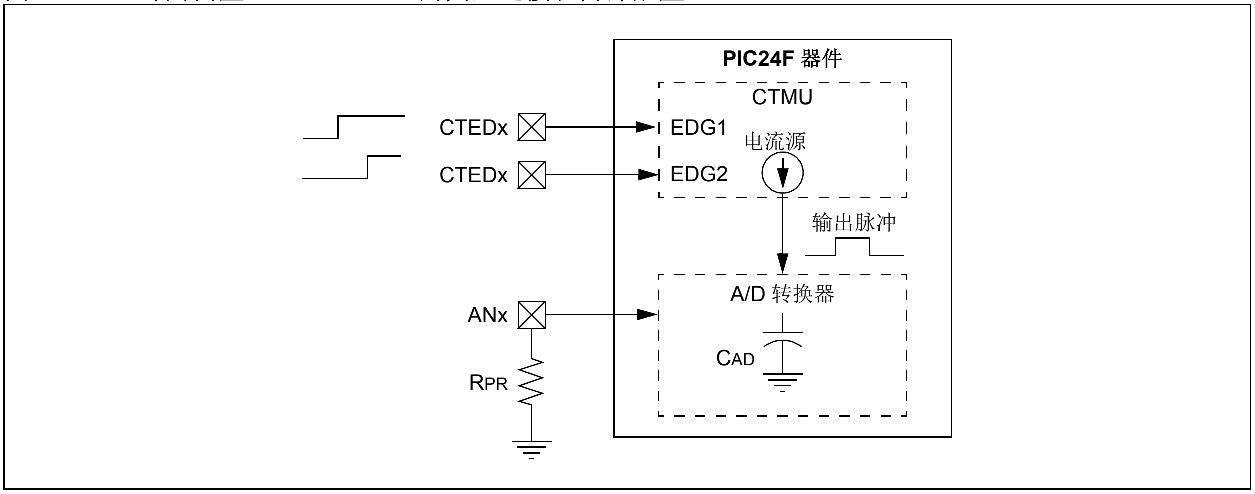
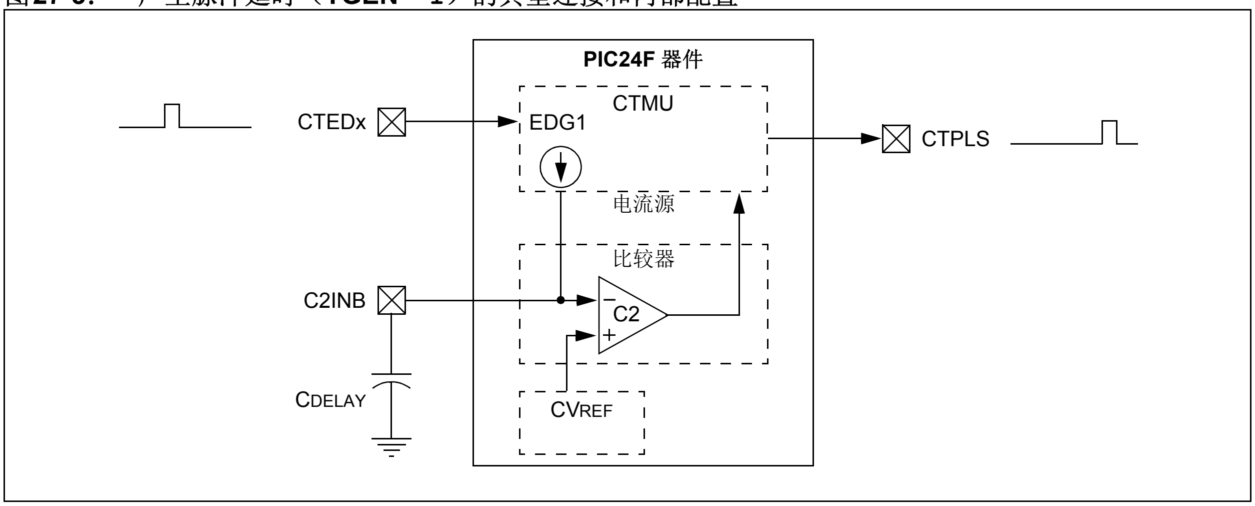


图27-3： 产生脉冲延时（TGEN = 1）的典型连接和内部配置



27.4 测量裸片温度

CTMU可配置为使用A/D通过专用的A/D通道24来测量裸片温度。执行以下步骤来测量二极管电压：

- 内部电流源必须设置为5.5 μA（IRNG<1:0> = 0x2）或55 μA（IRNG<1:0> = 0x3）。
- 为将电流源连接至二极管，EDG1STAT和EDG2STAT位必须相等（均为0或1）。
- CTMREQ位（AD1CON5<13>）必须设为1。
- A/D通道选择位必须为24（0x18），使用单端测量。

二极管的电压随温度变化而变化，如下图（图27-4）所示。请注意，下图会因所选电流源的幅值而异。

在-40°C到+100°C的范围内，斜率几乎呈线性，温度计算公式如下：

公式27-2：

对于5.5 μA电流源：

$$Tdie = \frac{710\ mV - Vdiode}{1.8}$$

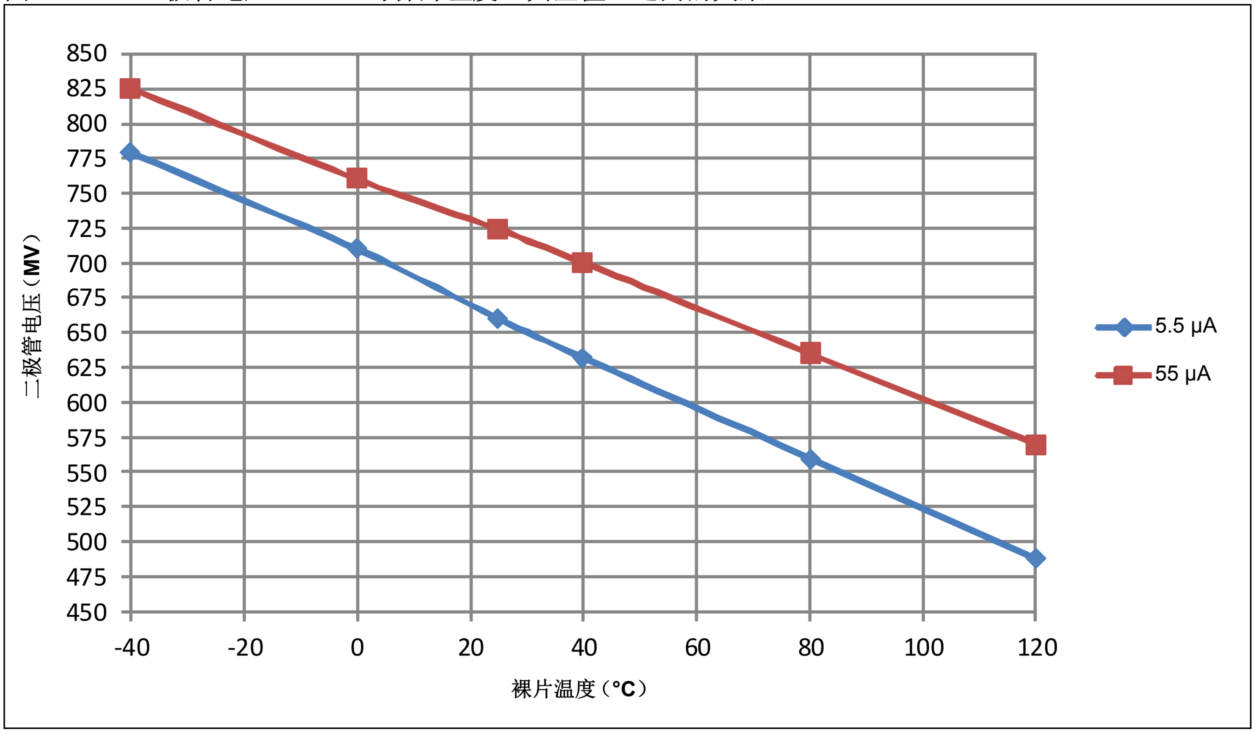
其中，*Vdiode*以mV为单位，*Tdie*以°C为单位

对于55 μA电流源：

$$Tdie = \frac{760\ mV - Vdiode}{1.55}$$

其中，*Vdiode*以mV为单位，*Tdie*以°C为单位

图27-4： 二极管电压（mV）与裸片温度（典型值）之间的关系



寄存器27-1： CTMUCON1L：CTMU控制寄存器1的低位字

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0
bit 7							bit 0

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

bit 15 **CTMUEN：**CTMU使能位
1 = 使能模块
0 = 禁止模块

bit 14 **未实现：**读为0

bit 13 **CTMUSIDL：**CTMU空闲模式停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 在空闲模式下模块继续工作

bit 12 **TGEN：**延时生成使能位
1 = 使能边沿延时生成并将电流源连接至比较器引脚
0 = 禁止边沿延时生成并将电流源连接至所选A/D输入引脚

bit 11 **EDGEN：**边沿使能位
1 = 未阻止边沿
0 = 阻止边沿

bit 10 **EDGSEQEN：**边沿顺序使能位
1 = 边沿1事件必须在边沿2事件发生前发生
0 = 无需边沿顺序

bit 9 **IDISSEN：**模拟电流源控制位
1 = 模拟电流源输出接地
0 = 模拟电流源输出未接地

bit 8 **CTTRIG：**CTMU触发信号控制位
1 = 使能触发信号输出
0 = 禁止触发信号输出

bit 7-2 **ITRIM<5:0>：**电流源微调位
011111 = 对标称电流的最大正向调整
011110
•
•
•
000001 = 对标称电流的最小正向调整
000000 = IRNG<1:0>指定的标称电流输出
111111 = 对标称电流的最小负向调整
•
•
•
100010
100001 = 对标称电流的最大负向调整

寄存器27-1：　CTMUCON1L：CTMU控制寄存器1的低位字（续）

bit 1-0　**IRNG<1:0>**：电流源范围选择位

如果IRNGH = 0：
11 = 55 μA
10 = 5.5 μA
01 = 550 nA
00 = 550 μA

如果IRNGH = 1：
11 = 保留
10 = 保留
01 = 2.2 mA
00 = 550 μA

寄存器27-2： CTMUCON1H：CTMU控制寄存器1的高位字

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EDG1MOD	EDG1POL	EDG1SEL3	EDG1SEL2	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
EDG2MOD	EDG2POL	EDG2SEL3	EDG2SEL2	EDG2SEL1	EDG2SEL0	—	IRNGH
bit 7							bit 0

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

bit 15 **EDG1MOD**：边沿1边沿敏感选择位
1 = 输入是边沿敏感的
0 = 输入是电平敏感的

bit 14 **EDG1POL**：边沿1极性选择位
1 = 边沿1设定为正边沿响应
0 = 边沿1设定为负边沿响应

bit 13-10 **EDG1SEL<3:0>**：边沿1源选择位
1111 = CMP C3OUT
1110 = CMP C2OUT
1101 = CMP C1OUT
1100 = IC3中断
1011 = IC2中断
1010 = IC1中断
1001 = CTED8引脚
1000 = CTED7引脚
0111 = CTED6引脚
0110 = CTED5引脚
0101 = CTED4引脚
0100 = CTED3引脚
0011 = CTED1引脚
0010 = CTED2引脚
0001 = OC1
0000 = Timer1匹配

bit 9 **EDG2STAT**：边沿2状态位
指示边沿2的状态并可写入以控制电流源。
1 = 已发生边沿2事件
0 = 未发生边沿2事件

bit 8 **EDG1STAT**：边沿1状态位
指示边沿1的状态并可写入以控制电流源。
1 = 已发生边沿1事件
0 = 未发生边沿1事件

bit 7 **EDG2MOD**：边沿2边沿敏感选择位
1 = 输入是边沿敏感的
0 = 输入是电平敏感的

bit 6 **EDG2POL**：边沿2极性选择位
1 = 边沿2设定为正边沿响应
0 = 边沿2设定为负边沿响应

寄存器27-2： CTMUCON1H：CTMU控制寄存器1的高位字（续）

bit 5-2	**EDG2SEL<3:0>**：边沿2源选择位 1111 = CMP C3OUT 1110 = CMP C2OUT 1101 = CMP C1OUT 1100 = 外设时钟 1011 = IC3中断 1010 = IC2中断 1001 = IC1中断 1000 = CTED13引脚 0111 = CTED12引脚 0110 = CTED11引脚 0101 = CTED10引脚 0100 = CTED9引脚 0011 = CTED1引脚 0010 = CTED2引脚 0001 = OC1 0000 = Timer1匹配
bit 1	**未实现**：读为0
bit 0	**IRNGH**：高电流范围选择位 1 = 使用高电流范围（550 μA-2.2 mA） 0 = 使用低电流范围（550 nA-50 μA） 电流输出通过CTMUCON1L寄存器中的IRNG<1:0>位进行设置。

寄存器27-3： **CTMUCON2L：CTMU控制寄存器2的低位字**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	IRSTEN	—	DSCHS2	DSCHS1	DSCHS0
bit 7							bit 0

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

bit 15-5 **未实现：**读为0

bit 4 **IRSTEN：**CTMU电流源复位使能位
1 = DSCHS<2:0>位或IDISSEN控制位选择的信号将复位CTMU边沿检测逻辑
0 = CTMU边沿检测逻辑必须由软件复位

bit 3 **未实现：**读为0

bit 2-0 **DSCHS<2:0>：**放电源选择位
111 = CLC2输出
110 = CLC1输出
101 = 禁止
100 = A/D转换结束
011 = MCCP3辅助输出
010 = MCCP2辅助输出
001 = MCCP1辅助输出
000 = 禁止

注：

28.0 高/低压检测（HLVD）

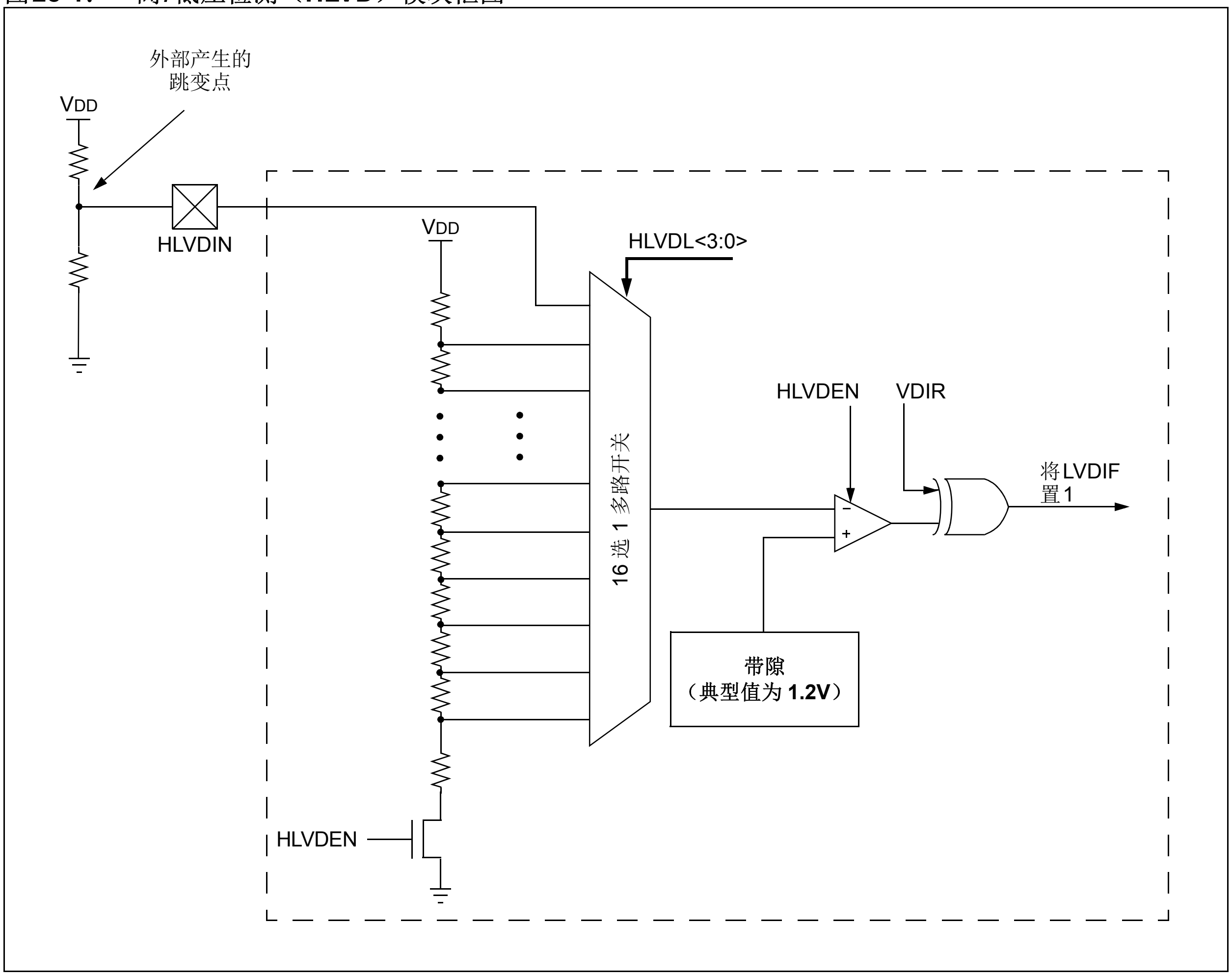
注：	本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。有关高/低压检测的更多信息，请参见《dsPIC33/PIC24 系列参考手册》的 **“高度集成的可编程高/低电压检测（HLVD）模块”**（DS39725），该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。

高/低压检测（High/Low-Voltage Detect，HLVD）模块是一个可编程的电路，它允许用户指定器件的电压跳变点和变化方向。

如果器件电压在变化方向相对于该跳变点发生了偏离，就会将中断标志位置1。如果允许了中断，程序将跳转到中断向量地址处执行，然后由软件响应该中断。在发生POR或BOR事件时，LVDIF标志可能会置1。固件应在应用程序首次使用该中断标志之前将其清零，即使已禁止中断也该如此。

HLVD控制寄存器（见寄存器28-1）完全控制HLVD模块的操作。用户可通过软件控制该寄存器将电路“关闭”，从而使器件的电流消耗降至最低。

图28-1： 高/低压检测（HLVD）模块框图



寄存器28-1： HLVDCON：高/低压检测控制寄存器

R/W-0	U-0	R/W-0	U-0	R/W-0	HS, HC, R-0	HS, HC, R-0	HS, HC, R-0
HLVDEN	—	LSIDL	—	VDIR	BGVST	IRVST	LVDEVT[2]
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	HLVDL3	HLVDL2	HLVDL1	HLVDL0
bit 7							bit 0

图注：	HS = 硬件置1位	HC = 硬件清零位	
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 15 **HLVDEN：**高/低压检测功能使能位
1 = 使能HLVD
0 = 禁止HLVD

bit 14 **未实现：**读为0

bit 13 **LSIDL：**HLVD空闲模式停止位
1 = 当器件进入空闲模式时，模块停止工作
0 = 在空闲模式下模块继续工作

bit 12 **未实现：**读为0

bit 11 **VDIR：**电压变化方向选择位
1 = 当电压等于或超过跳变点（HLVDL<3:0>）时，事件发生
0 = 当电压等于或低于跳变点（HLVDL<3:0>）时，事件发生

bit 10 **BGVST：**带隙电压稳定标志位
1 = 表示带隙电压稳定
0 = 表示带隙电压不稳定

bit 9 **IRVST：**内部参考电压稳定标志位
1 = 内部参考电压稳定；高压检测逻辑在检测到指定的电压范围时产生中断标志
0 = 内部参考电压不稳定；高压检测逻辑在检测到指定的电压范围时不会产生中断标志，并且HLVD中断不被允许

bit 8 **LVDEVT：**低压事件状态位[2]
1 = 在当前指令周期内，LVD事件为真
0 = 在当前指令周期内，LVD事件不为真

bit 7-4 **未实现：**读为0

bit 3-0 **HLVDL<3:0>：**高/低压检测限制位
1111 = 使用外部模拟输入（输入来自于HLVDIN引脚）
1110 = 跳变点1[1]
1101 = 跳变点2[1]
1100 = 跳变点3[1]
•
•
•
0100 = 跳变点11[1]
00xx = 未使用

注 1： 有关实际跳变点，请参见第32.0节“电气特性”。
2： 除非LVDEVT = 0，否则LVDIF标志不能由软件清零。必须对电压进行监视，以确保HLVD条件（通过VDIR和HLVDL<3:0>设置）不会生效。

29.0 特殊功能

> **注：** 本数据手册总结了该PIC24F系列器件的功能。但是不应把本手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》的以下章节，该文档可从Microchip网站（www.microchip.com）下载。本数据手册中的信息取代了FRM中的信息。
> - **“看门狗定时器（WDT）”**（DS39697）
> - **“高级器件集成”**（DS39719）
> - **“编程和诊断”**（DS39716）

PIC24FJ256GA705系列器件具有几项特殊功能旨在最大限度地提高应用的灵活性和可靠性，并通过减少外部元件将成本降至最低。提供的特殊功能包括：

- 灵活的配置
- 看门狗定时器（WDT）
- 代码保护
- JTAG边界扫描接口
- 在线串行编程
- 在线仿真

29.1 配置位

配置位存储在已实现的程序存储器的最后一页。可将这些位置1或清零以选择不同的器件配置。有两种配置位类型：系统操作位和代码保护位。系统操作位确定系统级元件的上电设置，如振荡器和看门狗定时器。代码保护位防止读写程序存储器。

29.1.1 配置PIC24FJ256GA705系列器件的注意事项

在PIC24FJ256GA705系列器件中，配置字节以易失性存储方式实现。这就意味着在器件每次上电时都必须对配置数据进行编程。配置数据存储在片上程序存储空间顶部的3个字中，这些字被称为闪存配置字。表29-1给出了它们的具体位置。器件复位时，配置数据会被自动从闪存配置字装入相应的配置寄存器中。

> **注：** 所有类型的器件复位都会重新装入配置数据。

当为这些器件创建应用程序时，用户应始终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到该地址。

程序存储器中所有闪存配置字的最高字节应始终为`0000 0000`。这样当这些存储单元在极少情况下被意外执行时，会被当作一条`NOP`指令。由于并未在相应的存储单元中实现这些配置位，因此向这些存储单元写`0`不会影响器件工作。

表29-1： 配置字地址

配置寄存器	PIC24FJ256GA70X	PIC24FJ128GA70X	PIC24FJ64GA70X
FSEC	02AF00h	015F00h	00AF00h
FBSLIM	02AF10h	015F10h	00AF10h
FSIGN	02AF14h	015F14h	00AF14h
FOSCSEL	02AF18h	015F18h	00AF18h
FOSC	02AF1Ch	015F1Ch	00AF1Ch
FWDT	02AF20h	015F20h	00AF20h
FPOR	02AF24h	015F24h	00AF24h
FICD	02AF28h	015F28h	00AF28h
FDEVOPT1	02AF2Ch	015F2Ch	00AF2Ch

寄存器29-1： FSEC配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
AIVTDIS	—	—	—	CSS2	CSS1	CSS0	CWRP
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
GSS1	GSS0	GWRP	—	BSEN	BSS1	BSS0	BWRP
bit 7							bit 0

图注：	PO = 一次编程位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 23-16 **未实现：**读为1

bit 15 **AIVTDIS：**备用中断向量表禁止位
1 = 禁止AIVT；INTCON2<8>（AIVTEN）位不可用
0 = 使能AIVT；INTCON2<8>（AIVTEN）位可用

bit 14-12 **未实现：**读为1

bit 11-9 **CSS<2:0>：**配置段（CS）代码保护级别位
111 = 无保护（CWRP写保护除外）
110 = 标准安全性
10x = 增强安全性
0xx = 高安全性

bit 8 **CWRP：**配置段程序写保护位
1 = 配置段不受写保护
0 = 配置段受写保护

bit 7-6 **GSS<1:0>：**通用段（GS）代码保护级别位
11 = 无保护（GWRP写保护除外）
10 = 标准安全性
0x = 高安全性

bit 5 **GWRP：**通用段程序写保护位
1 = 通用段不受写保护
0 = 通用段受写保护

bit 4 **未实现：**读为1

bit 3 **BSEN：**引导段（BS）控制位
1 = 未使能引导段
0 = 引导段大小由BSLIM<12:0>决定

bit 2-1 **BSS<1:0>：**引导段代码保护级别位
11 = 无保护（BWRP写保护除外）
10 = 标准安全性
0x = 高安全性

bit 0 **BWRP：**引导段程序写保护位
1 = 引导段可以写入
0 = 引导段受写保护

寄存器29-2： FBSLIM配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
—	—	—	BSLIM<12:8>				
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
BSLIM<7:0>							
bit 7							bit 0

图注：	PO = 一次编程位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 23-13 **未实现：**读为1

bit 12-0 **BSLIM<12:0>：**有效引导段代码闪存页地址限制（补码）位
该位域包含最后一个有效的引导段页 + 1（即，GS的第一页地址）。该值存储为页地址的补码，这样编程额外的0只会增加BS的大小。如果BSLIM<12:0>设为全1（未编程时的默认值），则有效引导段大小为零。

寄存器29-3： FSIGN配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

r-0	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 7							bit 0

图注：	PO = 一次编程位	r = 保留位	
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 23-16 **未实现：**读为1

bit 15 **保留：**保持为0

bit 14-0 **未实现：**读为1

寄存器29-4： FOSCSEL配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	U-1	r-0	r-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
IESO	PLLMODE3	PLLMODE2	PLLMODE1	PLLMODE0	FNOSC2	FNOSC1	FNOSC0
bit 7							bit 0

图注：	PO = 一次编程位	r = 保留位	
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 23-10 **未实现：**读为1

bit 9-8 **保留：**保持为0

bit 7 **IESO：**双速振荡器启动使能位
1 = 使用FRC启动器件，然后自动切换到就绪的用户选择的振荡器
0 = 使用用户选择的振荡器源启动器件

bit 6-3 **PLLMODE<3:0>：**倍频器选择位
1111 = 未使用PLL（PLLEN位不可用）
1110 = 选择了8x PLL
1101 = 选择了6x PLL
1100 = 选择了4x PLL
0111 = 选择了96 MHz PLL（输入频率 = 48 MHz）
0110 = 选择了96 MHz PLL（输入频率 = 32 MHz）
0101 = 选择了96 MHz PLL（输入频率 = 24 MHz）
0100 = 选择了96 MHz PLL（输入频率 = 20 MHz）
0011 = 选择了96 MHz PLL（输入频率 = 16 MHz）
0010 = 选择了96 MHz PLL（输入频率 = 12 MHz）
0001 = 选择了96 MHz PLL（输入频率 = 8 MHz）
0000 = 选择了96 MHz PLL（输入频率 = 4 MHz）

bit 2-0 **FNOSC<2:0>：**振荡器选择位
111 = 带分频器的振荡器（OSCFDIV）
110 = 保留
101 = 低功耗RC振荡器（LPRC）
100 = 辅助振荡器（SOSC）
011 = 带PLL的主振荡器（XTPLL、HSPLL和ECPLL）
010 = 主振荡器（XT、HS和EC）
001 = 带有PLL的快速RC振荡器（FRCPLL）
000 = 快速RC振荡器（FRC）

寄存器29-5： FOSC配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
FCKSM1	FCKSM0	IOL1WAY	PLLSS	SOSCSEL	OSCIOFCN	POSCMD1	POSCMD0
bit 7							bit 0

图注：	PO = 一次编程位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 23-8 **未实现：**读为1

bit 7-6 **FCKSM<1:0>：**时钟切换和监视器选择位
1x = 禁止时钟切换和故障保护时钟监视器
01 = 使能时钟切换，禁止故障保护时钟监视器
00 = 使能时钟切换和故障保护时钟监视器

bit 5 **IOL1WAY：**外设引脚选择配置位
1 = IOLOCK位只能置1一次（需要解锁序列）。
0 = IOLOCK位可根据需要置1和清零（需要解锁序列）

bit 4 **PLLSS：**PLL辅助选择配置位
该配置位只有在PLL未由系统使用（即，未被选作系统时钟源的一部分）时才有效。用于生成独立的REFO时钟输出。
1 = PLL由主振荡器提供时钟
0 = PLL由片上快速RC（Fast RC，FRC）振荡器提供时钟

bit 3 **SOSCSEL：**SOSC选择配置位
1 = 晶振（SOSCI/SOSCO）模式
0 = 数字（SOSCI）外部提供的时钟模式

bit 2 **OSCIOFCN：**CLKO使能配置位
1 = CLKO输出信号在OSCO引脚上有效（主振荡器被禁止或配置为EC模式时）
0 = 禁止CLKO输出

bit 1-0 **POSCMD<1:0>：**主振荡器配置位
11 = 禁止主振荡器模式
10 = 选择HS振荡器模式（10 MHz-32 MHz）
01 = 选择XT振荡器模式（1.5 MHz-10 MHz）
00 = 选择外部时钟模式

寄存器29-6： FWDT配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	R/PO-1	R/PO-1	U-1	R/PO-1	U-1	R/PO-1	R/PO-1
—	WDTCLK1	WDTCLK0	—	WDTCMX	—	WDTWIN1	WDTWIN0
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
WINDIS	FWDTEN1	FWDTEN0	FWPSA	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7							bit 0

图注：	PO = 一次编程位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 23-15 **未实现：**读为1

bit 14-13 **WDTCLK<1:0>：**看门狗定时器时钟选择位（WDTCMX = 1时）
11 = 始终使用LPRC
10 = WINDIS = 0、系统时钟不为LPRC以及器件未处于休眠模式时使用FRC；否则使用LPRC
01 = 始终使用SOSC
00 = 系统时钟不为LPRC以及器件未处于休眠模式时使用外设时钟；否则使用LPRC

bit 12 **未实现：**读为1

bit 11 **WDTCMX：**WDT时钟MUX控制位
1 = 使能WDT时钟MUX，WDT时钟由WDTCLK<1:0>选择
0 = WDT时钟为LPRC

bit 10 **未实现：**读为1

bit 9-8 **WDTWIN<1:0>：**看门狗定时器窗口宽度位
11 = WDT窗口为WDT周期的25%
10 = WDT窗口为WDT周期的37.5%
01 = WDT窗口为WDT周期的50%
00 = WDT窗口为WDT周期的75%

bit 7 **WINDIS：**窗口看门狗定时器禁止位
1 = 禁止窗口WDT
0 = 使能窗口WDT

bit 6-5 **FWDTEN<1:0>：**看门狗定时器使能位
11 = 使能WDT
10 = 禁止WDT（通过SWDTEN位实现控制）
01 = 仅在器件处于工作模式时使能WDT，在器件处于休眠模式时禁止WDT；禁止SWDTEN位
00 = 禁止WDT和SWDTEN

bit 4 **FWPSA：**看门狗定时器预分频比位
1 = WDT预分频比为1:128
0 = WDT预分频比为1:32

寄存器29-6： **FWDT配置寄存器（续）**

bit 3-0 **WDTPS<3:0>**：看门狗定时器后分频比选择位
`1111` = 1:32,768
`1110` = 1:16,384
`1101` = 1:8,192
`1100` = 1:4096
`1011` = 1:2048
`1010` = 1:1024
`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

寄存器29-7： FPOR配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

U-1	U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
—	—	—	—	DNVPEN	$\overline{\text{LPCFG}}$	BOREN1	BOREN0
bit 7							bit 0

图注：	PO = 一次编程位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 23-4 **未实现：**读为1

bit 3 **DNVPEN：**降压保护使能位
1 = BOR无效时使能降压保护
0 = BOR无效时禁止降压保护

bit 2 **$\overline{\text{LPCFG}}$：**低功耗稳压器控制位
1 = 保持功能不可用
0 = 在休眠期间保持功能可用且由RETEN控制

bit 1-0 **BOREN<1:0>：**欠压复位使能位
11 = 通过硬件使能欠压复位；禁止SBOREN位
10 = 欠压复位仅在器件处于工作模式时使能，在器件处于休眠模式时则被禁止；禁止SBOREN位
01 = 通过SBOREN位设置控制欠压复位
00 = 通过硬件禁止欠压复位；禁止SBOREN位

寄存器29-8： FICD配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

r-1	U-1	R/PO-1	U-1	U-1	U-1	R/PO-1	R/PO-1
—	—	JTAGEN	—	—	—	ICS1	ICS0
bit 7							bit 0

图注：	PO = 一次编程位	r = 保留位	
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 23-8 **未实现：**读为1

bit 7 **保留：**保持为1

bit 6 **未实现：**读为1

bit 5 **JTAGEN：**JTAG端口使能位
1 = 使能JTAG端口
0 = 禁止JTAG端口

bit 4-2 **未实现：**读为1

bit 1-0 **ICS<1:0>：**ICD通信通道选择位
11 = 通过PGC1/PGD1进行通信
10 = 通过PGC2/PGD2进行通信
01 = 通过PGC3/PGD3进行通信
00 = 保留；不要使用

寄存器29-9： FDEVOPT1配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	U-1
—	—	—	ALTI2C1	SOSCHP	TMPRPIN	ALTCMPI	—
bit 7							bit 0

图注：	PO = 一次编程位		
R = 可读位	W = 可写位	U = 未实现位，读为0	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

bit 23-5 **未实现：**读为1

bit 4 **ALTI2C1：**备用I2C1位
1 = RB9和RB8上的SDA1和SCL1
0 = RB5和RB6上的ASDA1和ASCL1

bit 3 **SOSCHP：**SOSC高功耗使能位（仅当SOSCSEL = 1时有效）
1 = 使能SOSC高功耗模式
0 = 使能SOSC低功耗模式（更多信息，请参见**第9.7.3节“低功耗SOSC工作”**）

bit 2 **TMPRPIN：**篡改引脚使能位
1 = 禁止$\overline{\text{TMPRN}}$引脚功能（RB9）
0 = 使能TMPRN引脚功能

bit 1 **ALTCMPI：**备用比较器输入使能位
1 = C1INC、C2INC和C3INC位于标准引脚位置上
0 = C1INC、C2INC和C3INC位于RB9[(1)]上

bit 0 **未实现：**读为1

注 1： RB9用于多种功能，但仅允许一个用例。

表29-2： PIC24FJ内核器件ID寄存器

地址	名称	Bit															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FF0000h	DEVID	FAMID<7:0>								DEV<7:0>							
FF0002h	DEVREV	—												REV<3:0>			

表29-3： 器件ID位域说明

位域	寄存器	说明
FAMID<7:0>	DEVID	编码器件的系列ID； FAMID = 0x75。
DEV<7:0>	DEVID	编码器件的独立ID。
REV<3:0>	DEVREV	编码器件的连续（数字）版本标识符。

表29-4： PIC24FJ256GA705系列器件ID

器件	DEVID
PIC24FJ64GA705	07
PIC24FJ128GA705	0B
PIC24FJ256GA705	0F
PIC24FJ64GA704	05
PIC24FJ128GA704	09
PIC24FJ256GA704	0D
PIC24FJ64GA702	06
PIC24FJ128GA702	0A
PIC24FJ256GA702	0E

29.2 器件惟一标识符（UDID）

所有PIC24FJ256GA705系列器件在最终制造期间都单独编码了器件惟一标识符（Unique Device Identifier，UDID）。UDID不能通过批量擦除命令或任何其他用户可访问的方式擦除。凭借该功能，可以在需要时跟踪应用中Microchip器件的制造信息。应用制造商也可将该功能用于任意个可能需要惟一标识的应用场合，例如：

- 跟踪器件
- 惟一序列号
- 惟一安全密钥

UDID包含五个24位程序字。这些字段组合在一起构成一个120位的惟一标识符。

UDID存储在器件配置空间中0x800F00和0x800F08之间的五个只读存储单元中。表29-5列出了标识符字的地址并显示了其中的内容。

表29-5： UDID地址

UDID	地址	说明
UDID1	0x800F00	UDID字1
UDID2	0x800F02	UDID字2
UDID3	0x800F04	UDID字3
UDID4	0x800F06	UDID字4
UDID5	0x800F08	UDID字5

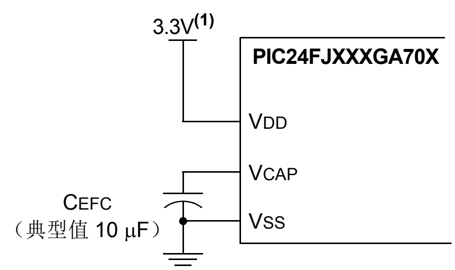
29.3 片上稳压器

所有PIC24FJ256GA705系列器件均使用标称值为1.8V的电压为其内核数字逻辑供电。对于需要工作在一个更高的典型电压值（如3.3V）的设计来讲，这可能会带来问题。为了简化系统设计，PIC24FJ256GA705系列中的所有器件均包含一个片上稳压器，可使器件内核逻辑通过VDD工作。

此稳压器始终使能。它通过范围从约2.1V的VDD一直到器件的VDDMAX为数字内核逻辑提供恒定电压（标称值1.8V）。该稳压器无法将VDD电压升高。为防止电压降得对于稳压器过低时产生“欠压”条件，此时会发生欠压复位。稳压器输出跟随 VDD，通常比此电压小300 mV。

必须在VCAP引脚连接低ESR电容（如陶瓷电容）（图29-1）。这有助于维持稳压器的稳定性。**第32.1节“直流特性”**中提供了该滤波电容（CEFC）的推荐值。

图29-1： 片上稳压器的连接



注 1： 这为典型的工作电压值。请参见**第32.0节“电气特性”**了解VDD的全部工作范围。

29.3.1 片上稳压器和POR

稳压器需要约10 μs的时间来产生输出。在这段称为TVREG的时间内，禁止代码执行。每次器件在掉电（包括休眠模式）后恢复工作时都需要TVREG。TVREG由VREGS位（RCON<8>）和WDTWIN<1:0>配置位（FWDT<9:8>）的状态决定。有关TVREG的更多信息，请参见**第32.0节“电气特性”**。

注：	更多信息，请参见**第32.0节“电气特性”**。本数据手册中的信息取代了FRM中的信息。

29.3.2 稳压器待机模式

除IDD/IPD外，片上稳压器总是会额外消耗一个小的电流，器件工作在休眠模式下也是如此，尽管此时内核数字逻辑并不需要耗电。为了在功耗关键的应用中节省更多的功耗，设置稳压器在器件进入休眠模式时自动进入待机模式。VREGS（RCON<8>）位控制此特性。清零VREGS位，使能待机模式。稳压器需要等待TVREG时间才能从待机模式唤醒。

29.3.3 低压保持稳压器

在休眠模式下，PIC24FJ256GA705系列器件可使用单独的低功耗、低压保持稳压器来为关键电路供电。当所有其他数字内核逻辑掉电时，此稳压器（在标称电压1.2V下工作）保持对数据RAM和RTCC供电。有关低压保持稳压器的详细信息，请参见**第10.2.4节“低压保持稳压器”**。

29.4 看门狗定时器（WDT）

对于PIC24FJ256GA705系列器件，WDT由LPRC振荡器、辅助振荡器（SOSC）或系统定时器驱动。当器件处于休眠模式时，将使用LPRC振荡器。当使能WDT时，时钟源也将使能。

由LPRC提供的WDT时钟源的频率标称值为31 kHz。该信号输入给可配置为5位（32分频）或7位（128分频）工作的预分频器。分频比由FWPSA配置位设置。对于31 kHz的输入，预分频器生成的标称WDT超时周期（T_{WDT}），5位模式下为1 ms，7位模式下为4 ms。

分频比可变的后分频器对WDT预分频器的输出进行分频，以获得范围较宽的超时周期。后分频比由WDTPS<3:0>配置位（FWDT<3:0>）控制，这些配置位总共允许选择16种设置，从1:1至1:32,768。使用预分频器和后分频器，可以使超时周期的范围扩展到1 ms至131秒。

WDT、预分频器和后分频器在以下条件下复位：

- 任何器件复位时
- 在完成时钟切换时，无论时钟切换是由软件（即，在更改NOSCx位后将OSWEN位置1）引起还是由硬件（即，故障保护时钟监视器）引起
- 执行`PWRSAV`指令时（即，进入休眠或空闲模式）
- 当器件退出休眠或空闲模式恢复正常工作时
- 当在正常执行过程中执行`CLRWDT`指令时

如果使能WDT，它将在休眠或空闲模式下继续运行。当发生WDT超时时，将唤醒器件并且代码将从执行`PWRSAV`指令处继续执行。器件被唤醒后，需要用软件将相应的SLEEP或IDLE位（RCON<3:2>）清零。

WDT标志位WDTO（RCON<4>）不会在WDT超时后自动清零。要检测后续的WDT事件，必须用软件将该标志位清零。

注：	执行`CLRWDT`和`PWRSAV`指令会将预分频器和后分频器的计数值清零。

29.4.1 窗口操作

看门狗定时器具有可选的固定窗口工作模式。在此窗口模式下，`CLRWDT`指令只能在编程的WDT周期的后1/4周期复位WDT。在该窗口前执行`CLRWDT`指令会导致WDT复位，这与WDT超时类似。

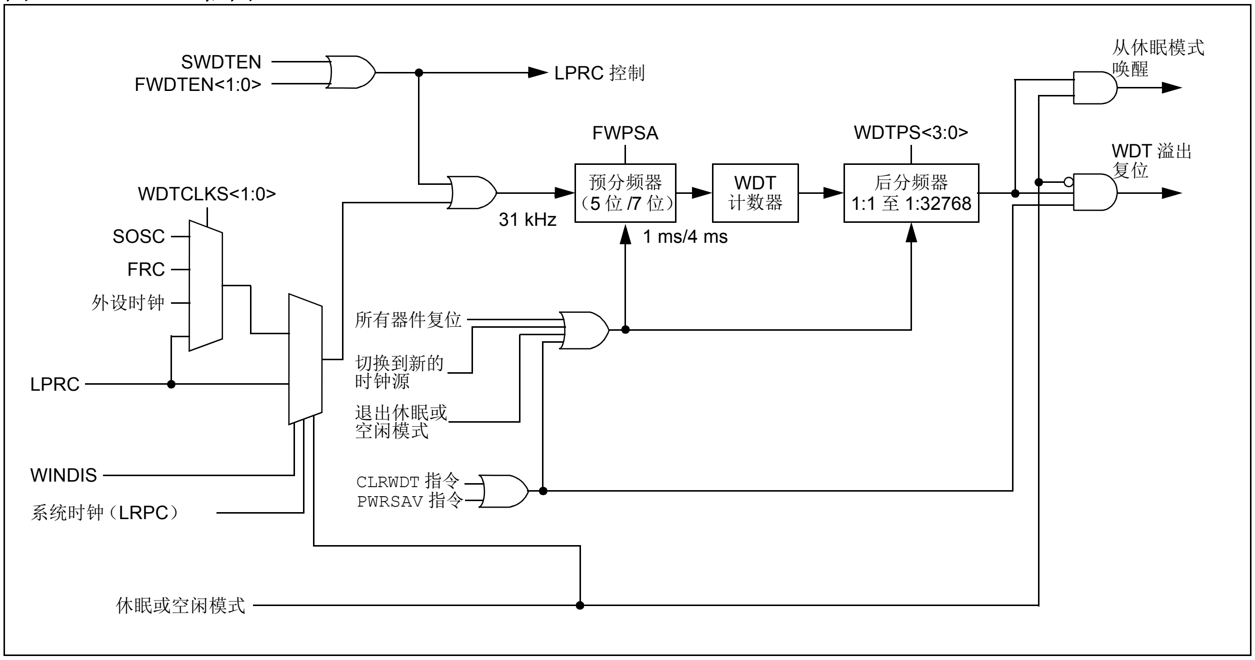
将WINDIS配置位（FWDT<7>）编程为`0`可使能WDT窗口模式。

29.4.2 控制寄存器

通过FWDTEN<1:0>配置位（FWDT<6:5>）使能或禁止WDT。当配置位FWDTEN<1:0> = `11`时，始终使能WDT。

当配置位FWDTEN<1:0> = `10`时，可以选择使用软件来控制WDT。当FWDTEN<1:0> = `00`时，始终禁止看门狗定时器。通过将SWDTEN控制位（RCON<5>）置1来用软件使能WDT。任何器件复位都会导致SWDTEN控制位清零。软件WDT选项允许用户在关键代码段使能WDT并在非关键代码段禁止WDT，以最大限度地降低功耗。

图29-2：　WDT框图


SWDTEN
FWDTEN<1:0>
LPRC 控制
从休眠模式
唤醒
FWPSA
WDTPS<3:0>
WDT 溢出
复位
WDTCLKS<1:0>
预分频器
（5 位 /7 位）
WDT
计数器
后分频器
1:1 至 1:32768
31 kHz
1 ms/4 ms
SOSC
FRC
外设时钟
LPRC
所有器件复位
切换到新的
时钟源
退出休眠或
空闲模式
WINDIS
CLRWDT 指令
PWRSAV 指令
系统时钟（LRPC）
休眠或空闲模式

29.5 程序校验和代码保护

PIC24FJ256GA705系列器件提供支持通用段（GS）安全和引导段（BS）安全的CodeGuard™安全性的基本实现。该功能有助于保护各方的知识产权。

> **注：** 有关安全性使用、配置和操作的更多信息，请参见《dsPIC33/PIC24系列参考手册》的**“CodeGuard™中级安全性”**（DS70005182）。

29.6 JTAG接口

PIC24FJ256GA705系列器件实现了JTAG接口，该接口支持边界扫描器件测试。

29.7 在线串行编程

PIC24FJ256GA705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根线即可完成这一操作，其中时钟线（PGCx）和数据线（PGDx）各一根，其余3根分别是电源线（VDD）、接地线（VSS）和$\overline{\text{MCLR}}$。这允许用户在生产电路板时使用未编程器件，仅在产品交付之前才对单片机进行编程，从而可以使用最新版本的固件或者定制固件进行编程。

29.8 用户OTP存储器

PIC24FJ256GA705系列器件提供256字节的可一次性编程（OTP）存储器，其地址范围为801700h至8017FEh。该存储器可用于持久存储针对特定应用的信息，这些信息不会因对器件进行重新编程而被擦除。这包括许多类型的信息，例如（但不限于）：

- 应用程序校验和
- 代码版本信息
- 产品信息
- 序列号
- 系统制造日期
- 制造批号

OTP存储器不能通过程序执行（即，`TBLWT`指令）写入；只能在器件编程期间写入。全片擦除不会清空数据。

> **注：** OTP存储区域内的数据不能多次编程。

29.9 在线调试器

这一功能允许结合MPLAB IDE进行一些简单的调试。通过PGCx（仿真/调试时钟）和PGDx（仿真/调试数据）引脚控制调试功能。

要使用器件的在线调试功能，在设计中必须实现对$\overline{\text{MCLR}}$、VDD、VSS和PGCx/PGDx引脚对（由ICS<1:0>配置位指定）的正确ICSP连接。此外，当使能该功能时，某些资源就不能用于一般用途了。这些资源包括数据RAM的前80个字节和两个I/O引脚。

30.0 开发支持

一系列软件及硬件开发工具对 PIC® 单片机（MCU）和 dsPIC® 数字信号控制器（DSC）提供支持：

- 集成开发环境
  - MPLAB® X IDE 软件
- 编译器 / 汇编器 / 链接器
  - MPLAB XC 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 X SIM 软件模拟器
- 仿真器
  - MPLAB REAL ICE™ 在线仿真器
- 在线调试器 / 编程器
  - MPLAB ICD 3
  - PICkit™ 3
- 器件编程器
  - 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
- 第三方开发工具

30.1 MPLAB X 集成开发环境软件

MPLAB X IDE 是适用于 Microchip 和第三方软硬件开发工具统一的通用图形用户界面，可以在 Windows®、Linux 和 Mac OS® X 上运行。MPLAB X IDE 是一款全新的 IDE，它基于 NetBeans IDE，包含许多免费的软件组件和插件，适用于高性能的应用程序开发和调试。通过这一无缝交互的用户界面，在不同工具之间的迁移以及从软件模拟器到硬件调试和编程工具的升级都变得极为简便。

MPLAB X IDE 具有完善的项目管理、可视化的调用图、可配置的观察窗口以及包含代码补全功能和上下文菜单的功能丰富编辑器，因此对于新用户来说非常灵活和友好。 MPLAB X IDE 支持对多个项目使用多个工具和同时调试，因此也完全可以满足经验丰富用户的需求。

功能丰富的编辑器：

- 彩色高亮显示语法
- 智能代码补全功能，在输入代码时提供建议和提示
- 基于用户定义规则，代码自动格式化
- 即时解析

用户友好的可定制界面：

- 完全可定制界面：工具栏、工具栏图标、窗口和窗口放置等
- 调用图窗口

基于项目的工作空间：

- 多个项目
- 多个工具
- 多种配置
- 同时调试会话

文件历史和错误跟踪：

- 本地文件历史功能
- 内建对 Bugzilla 缺陷跟踪系统的支持

30.2 MPLAB XC 编译器

MPLAB XC编译器是适用于Microchip所有8位、16位和32位MCU以及DSC器件的完全ANSI C编译器。这些编译器提供强大的集成功能以及出色的代码优化功能，且易于使用。MPLAB XC 编译器可在 Windows、Linux 或 Mac OS X 上运行。

为方便进行源代码级调试，编译器提供了已针对 MPLAB X IDE 优化的调试信息。

MPLAB XC 编译器的免费版支持所有器件和命令，没有时间或存储容量限制，且为大多数应用程序提供了充分的代码优化。

MPLAB XC 编译器包含汇编器、链接器和实用程序。汇编器生成可重定位目标文件，然后通过链接器将生成的可重定位目标文件与其他可重定位目标文件或归档文件归档或链接在一起，进而生成可执行文件。MPLAB XC 编译器使用汇编器来生成目标文件。汇编器具有如下突出特性：

- 支持全部器件指令集
- 支持定点和浮点数据
- 命令行接口
- 丰富的伪指令集
- 灵活的宏语言
- 与 MPLAB X IDE 兼容

30.3 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 X 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

30.4 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK 目标链接器组合由 MPASM 汇编器生成的可重定位目标文件。通过使用链接器脚本中的伪指令，它还可链接预编译库中的可重定位目标文件。

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

30.5 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和PIC32 MCU以及dsPIC DSC器件从符号汇编语言生成可重定位机器码。MPLAB XC 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
- 支持定点和浮点数据
- 命令行接口
- 丰富的指令集
- 与 MPLAB X IDE 兼容

30.6 MPLAB X SIM 软件模拟器

MPLAB X 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 X SIM 软件模拟器完全支持使用 MPLAB XC 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。

30.7 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 器件推出的新一代高速仿真器。结合 MPLAB X IDE 易于使用且功能强大的图形用户界面，该仿真器可对所有 8 位、16 位和 32 位 MCU 及 DSC 器件进行调试和编程。

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（RJ-11）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 X IDE 下载将来版本的固件，对该仿真器进行现场升级。在同类仿真器中，MPLAB REAL ICE 的优势十分明显：全速仿真、运行时变量观察、跟踪分析、复杂断点、逻辑探针、耐用的探针接口及较长（长达 3 米）的互连电缆。

30.8 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 的闪存 DSC 和 MCU 器件。结合 MPLAB X IDE 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（RJ-11）与目标板相连。MPLAB ICD 3 支持所有 MPLAB ICD 2 连接器。

30.9 PICkit 3 在线调试器 / 编程器

结合 MPLAB X IDE 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 调试连接器（RJ-11）（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（In-Circuit Serial Programming™， ICSP™）。

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）。

30.10 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 MCU 和 dsPIC DSC 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。

30.11 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）。

30.12 第三方开发工具

Microchip 还提供一些来自第三方供应商的优秀开发工具。这些工具均经过精心挑选，功能独特，物有所值。

- SoftLog 和 CCS 等公司提供的器件编程器和量产编程器
- Gimpel 和 Trace Systems 等公司提供的软件工具
- Saleae 和 Total Phase 等公司提供的协议分析器
- MikroElektronika、Digilent® 和 Olimex 等公司提供的演示板
- EZ Web Lynx、WIZnet 和 IPLogika® 等公司提供的嵌入式以太网解决方案

31.0 指令集汇总

注：	本章是PIC24F指令集架构（ISA）的简要汇总，并不能作为详尽的参考资料使用。

PIC24F指令集与以前的PIC MCU指令集相比，添加了许多增强功能，并保持了易于从以前的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。只有三条指令需要两个程序存储单元。

每条单字指令都是一个24位字，由一个8位的操作码（指明指令类型）和一个或多个操作数（指定指令具体操作）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：

- 针对字或字节的操作
- 针对位的操作
- 立即数操作
- 控制操作

表31-1给出了在说明指令时要用到的通用符号。表31-2中的PIC24F指令集汇总列出了所有指令及每条指令影响的状态标志。

大多数针对字或字节的W寄存器指令（包括桶形移位指令）有三个操作数：

- 第一个源操作数，通常是寄存器Wb，不带任何地址修改量
- 第二个源操作数，通常是寄存器Ws，带或不带地址修改量
- 保存结果的目标寄存器，通常是寄存器Wd，带或不带地址修改量

不过，针对字或字节的文件寄存器指令只有两个操作数：

- 文件寄存器，由f值指定
- 目标寄存器，可以是文件寄存器f或W0寄存器（用WREG表示）

大多数位操作类指令（包括简单的循环/移位指令）有两个操作数：

- W寄存器（带或不带地址修改量）或文件寄存器（由Ws或f的值指定）
- W寄存器或文件寄存器中的位（由一个立即数值指定，或者由寄存器Wb的内容间接指定）

涉及数据传送的立即数指令，可能使用下列操作数：

- 要被装入到W寄存器或文件寄存器中的立即数（由k指定）
- 要装入立即数的W寄存器或文件寄存器（由Wb或f指定）

然而，涉及算术或逻辑运算的立即数指令，使用如下的操作数：

- 第一个源操作数是寄存器Wb，不带任何地址修改量
- 第二个源操作数是立即数
- 操作结果的目标寄存器（仅在与第一个源操作数不同时）通常是寄存器Wd（带或不带地址修改量）

控制操作类指令可使用以下操作数：

- 程序存储器地址
- 表读和表写指令的模式

除某些双字指令外大多数指令都是单字指令，双字指令用48位来提供所需信息，在第二个字中，高8位全为0。如果指令自身将第二个字当作一条指令来执行的话，它将作为一条NOP指令来执行。

执行大部分单字指令都只需要一个指令周期，除非条件测试结果为真或者指令执行改变了程序计数器的值。对于上述两种特殊情况，指令执行需要两个指令周期，在额外的指令周期中执行NOP指令。值得注意的特殊指令有：BRA（无条件/计算转移指令）、间接CALL/GOTO指令、所有表读和表写指令以及RETURN/RETFIE指令，这些指令都是单字指令，但执行需要2或3个指令周期。

某些涉及到跳过后续指令的指令，如果要执行跳过的话，可能需要两个或三个周期，这取决于被跳过的指令是单字还是双字指令。此外，双字传送需要两个周期。双字指令执行需要两个指令周期。

表31-1： 操作码说明中使用的符号

字段	说明
#text	表示由text定义的立即数
(text)	表示text的内容
[text]	表示由text寻址的存储单元
{ }	可选字段或操作
<n:m>	寄存器位域
.b	字节模式选择
.d	双字模式选择
.S	影子寄存器选择
.w	字模式选择（默认）
bit4	4位位选择字段（用于字寻址指令）∈ {0...15}
C、DC、N、OV和Z	MCU状态位：进位、半进位、负、溢出和全零标志位
Expr	绝对地址、标号或表达式（由链接器解析）
f	文件寄存器地址 ∈ {0000h...1FFFh}
lit1	1位无符号立即数 ∈ {0,1}
lit4	4位无符号立即数 ∈ {0...15}
lit5	5位无符号立即数 ∈ {0...31}
lit8	8位无符号立即数 ∈ {0...255}
lit10	10位无符号立即数 ∈ {0...255}（对于字节模式）或{0...1023}（对于字模式）
lit14	14位无符号立即数 ∈ {0...16383}
lit16	16位无符号立即数 ∈ {0...65535}
lit23	23位无符号立即数 ∈ {0...8388607}；LSb必须为0
无	该字段无需内容，可为空
PC	程序计数器
Slit10	10位有符号立即数 ∈ {-512...511}
Slit16	16位有符号立即数 ∈ {-32768...32767}
Slit6	6位有符号立即数 ∈ {-16...16}
Wb	基本W寄存器 ∈ {W0..W15}
Wd	目标W寄存器 ∈ {Wd, [Wd], [Wd++], [Wd--], [++Wd], [--Wd]}
Wdo	目标W寄存器 ∈ {Wnd, [Wnd], [Wnd++], [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb]}
Wm,Wn	被除数和除数工作寄存器对（直接寻址）
Wn	16个工作寄存器之一 ∈ {W0..W15}
Wnd	16个目标工作寄存器之一 ∈ {W0..W15}
Wns	16个源工作寄存器之一 ∈ {W0..W15}
WREG	W0（文件寄存器指令中使用的工作寄存器）
Ws	源W寄存器 ∈ { Ws, [Ws], [Ws++], [Ws--], [++Ws], [--Ws] }
Wso	源W寄存器 ∈ { Wns, [Wns], [Wns++], [Wns--], [++Wns], [--Wns], [Wns+Wb] }

表31-2： 指令集概述

汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
ADD	ADD f	f = f + WREG	1	1	C、DC、N、OV和Z
	ADD f,WREG	WREG = f + WREG	1	1	C、DC、N、OV和Z
	ADD #lit10,Wn	Wd = lit10 + Wd	1	1	C、DC、N、OV和Z
	ADD Wb,Ws,Wd	Wd = Wb + Ws	1	1	C、DC、N、OV和Z
	ADD Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C、DC、N、OV和Z
ADDC	ADDC f	f = f + WREG + (C)	1	1	C、DC、N、OV和Z
	ADDC f,WREG	WREG = f + WREG + (C)	1	1	C、DC、N、OV和Z
	ADDC #lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C、DC、N、OV和Z
	ADDC Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C、DC、N、OV和Z
	ADDC Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C、DC、N、OV和Z
AND	AND f	f = f .AND.WREG	1	1	N和Z
	AND f,WREG	WREG = f .AND. WREG	1	1	N和Z
	AND #lit10,Wn	Wd = lit10 .AND. Wd	1	1	N和Z
	AND Wb,Ws,Wd	Wd = Wb .AND. Ws	1	1	N和Z
	AND Wb,#lit5,Wd	Wd = Wb .AND. lit5	1	1	N和Z
ASR	ASR f	f = 算术右移f	1	1	C、N、OV和Z
	ASR f,WREG	WREG = 算术右移f	1	1	C、N、OV和Z
	ASR Ws,Wd	Wd = 算术右移Ws	1	1	C、N、OV和Z
	ASR Wb,Wns,Wnd	Wnd = 将Wb算术右移Wns位	1	1	N和Z
	ASR Wb,#lit5,Wnd	Wnd = 将Wb算术右移lit5位	1	1	N和Z
BCLR	BCLR f,#bit4	将f中的指定位清零	1	1	无
	BCLR Ws,#bit4	将Ws中的指定位清零	1	1	无
BRA	BRA C,Expr	如果有进位则跳转	1	1（2）	无
	BRA GE,Expr	如果大于或等于则跳转	1	1（2）	无
	BRA GEU,Expr	如果无符号大于或等于则跳转	1	1（2）	无
	BRA GT,Expr	如果大于则跳转	1	1（2）	无
	BRA GTU,Expr	如果无符号大于则跳转	1	1（2）	无
	BRA LE,Expr	如果小于或等于则跳转	1	1（2）	无
	BRA LEU,Expr	如果无符号小于或等于则跳转	1	1（2）	无
	BRA LT,Expr	如果小于则跳转	1	1（2）	无
	BRA LTU,Expr	如果无符号小于则跳转	1	1（2）	无
	BRA N,Expr	如果为负则跳转	1	1（2）	无
	BRA NC,Expr	如果没有进位则跳转	1	1（2）	无
	BRA NN,Expr	如果不为负则跳转	1	1（2）	无
	BRA NOV,Expr	如果未溢出则跳转	1	1（2）	无
	BRA NZ,Expr	如果不为零则跳转	1	1（2）	无
	BRA OV,Expr	如果溢出则跳转	1	1（2）	无
	BRA Expr	无条件跳转	1	2	无
	BRA Z,Expr	如果为零则跳转	1	1（2）	无
	BRA Wn	计算跳转	1	2	无
BSET	BSET f,#bit4	将f中的指定位置1	1	1	无
	BSET Ws,#bit4	将Ws中的指定位置1	1	1	无
BSW	BSW.C Ws,Wb	将C位内容写入Ws<Wb>	1	1	无
	BSW.Z Ws,Wb	将Z位内容取反写入Ws<Wb>	1	1	无
BTG	BTG f,#bit4	将f中的指定位取反	1	1	无
	BTG Ws,#bit4	将Ws中的指定位取反	1	1	无
BTSC	BTSC f,#bit4	对f中的指定位进行测试，如果为零则跳过	1	1（2或3）	无
	BTSC Ws,#bit4	对Ws中的指定位进行测试，如果为零则跳过	1	1（2或3）	无

表31-2： 指令集概述 （续）

汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
BTSS	BTSS f,#bit4	对f中的指定位进行测试，如果为1则跳过	1	1（2或3）	无
	BTSS Ws,#bit4	对Ws中的指定位进行测试，如果为1则跳过	1	1（2或3）	无
BTST	BTST f,#bit4	对f中的指定位进行测试	1	1	Z
	BTST.C Ws,#bit4	对Ws中的指定位进行测试，并将其值存储到C	1	1	C
	BTST.Z Ws,#bit4	对Ws中的指定位进行测试，并将其反码存储到Z	1	1	Z
	BTST.C Ws,Wb	对Ws<Wb>位进行测试，并将其值存储到C	1	1	C
	BTST.Z Ws,Wb	对Ws<Wb>位进行测试，并将其反码存储到Z	1	1	Z
BTSTS	BTSTS f,#bit4	对f中的指定位进行测试，并将f中的该位置1	1	1	Z
	BTSTS.C Ws,#bit4	对Ws中的指定位进行测试，并将其值存储到C，然后将Ws中的该位置1	1	1	C
	BTSTS.Z Ws,#bit4	对Ws中的指定位进行测试，并将其反码存储到Z，然后将Ws中的该位置1	1	1	Z
CALL	CALL lit23	调用子程序	2	2	无
	CALL Wn	间接调用子程序	1	2	无
CLR	CLR f	f = 0x0000	1	1	无
	CLR WREG	WREG = 0x0000	1	1	无
	CLR Ws	Ws = 0x0000	1	1	无
CLRWDT	CLRWDT	将看门狗定时器清零	1	1	WDTO和Sleep
COM	COM f	f = $\overline{\text{f}}$	1	1	N和Z
	COM f,WREG	WREG = $\overline{\text{f}}$	1	1	N和Z
	COM Ws,Wd	Wd = $\overline{\text{Ws}}$	1	1	N和Z
CP	CP f	比较f和WREG	1	1	C、DC、N、OV和Z
	CP Wb,#lit5	比较Wb和lit5	1	1	C、DC、N、OV和Z
	CP Wb,Ws	比较Wb和Ws（Wb – Ws）	1	1	C、DC、N、OV和Z
CP0	CP0 f	比较f和0x0000	1	1	C、DC、N、OV和Z
	CP0 Ws	比较Ws和0x0000	1	1	C、DC、N、OV和Z
CPB	CPB f	带借位比较f和WREG	1	1	C、DC、N、OV和Z
	CPB Wb,#lit5	带借位比较Wb和lit5	1	1	C、DC、N、OV和Z
	CPB Wb,Ws	带借位比较Wb和Ws（Wb – Ws – $\overline{\text{C}}$）	1	1	C、DC、N、OV和Z
CPSEQ	CPSEQ Wb,Wn	比较Wb和Wn，如果相等则跳过	1	1（2或3）	无
CPSGT	CPSGT Wb,Wn	比较Wb和Wn，如果大于则跳过	1	1（2或3）	无
CPSLT	CPSLT Wb,Wn	比较Wb和Wn，如果小于则跳过	1	1（2或3）	无
CPSNE	CPSNE Wb,Wn	比较Wb和Wn，如果不等于则跳过	1	1（2或3）	无
DAW	DAW.B Wn	Wn = 十进制调整Wn	1	1	C
DEC	DEC f	f = f – 1	1	1	C、DC、N、OV和Z
	DEC f,WREG	WREG = f - 1	1	1	C、DC、N、OV和Z
	DEC Ws,Wd	Wd = Ws – 1	1	1	C、DC、N、OV和Z
DEC2	DEC2 f	f = f – 2	1	1	C、DC、N、OV和Z
	DEC2 f,WREG	WREG = f – 2	1	1	C、DC、N、OV和Z
	DEC2 Ws,Wd	Wd = Ws – 2	1	1	C、DC、N、OV和Z
DISI	DISI #lit14	在k个指令周期内禁止中断	1	1	无
DIV	DIV.SW Wm,Wn	有符号16/16位整数除法	1	18	N、Z、C和OV
	DIV.SD Wm,Wn	有符号32/16位整数除法	1	18	N、Z、C和OV
	DIV.UW Wm,Wn	无符号16/16位整数除法	1	18	N、Z、C和OV
	DIV.UD Wm,Wn	无符号32/16位整数除法	1	18	N、Z、C和OV
EXCH	EXCH Wns,Wnd	交换Wns和Wnd的内容	1	1	无
FF1L	FF1L Ws,Wnd	从左边（MSb）开始查找第一个1	1	1	C
FF1R	FF1R Ws,Wnd	从右边（LSb）开始查找第一个1	1	1	C

表31-2： 指令集概述 （续）

汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
GOTO	GOTO Expr	跳转到地址	2	2	无
	GOTO Wn	间接跳转到地址	1	2	无
INC	INC f	f = f + 1	1	1	C、DC、N、OV和Z
	INC f,WREG	WREG = f + 1	1	1	C、DC、N、OV和Z
	INC Ws,Wd	Wd = Ws + 1	1	1	C、DC、N、OV和Z
INC2	INC2 f	f = f + 2	1	1	C、DC、N、OV和Z
	INC2 f,WREG	WREG = f + 2	1	1	C、DC、N、OV和Z
	INC2 Ws,Wd	Wd = Ws + 2	1	1	C、DC、N、OV和Z
IOR	IOR f	f = f .IOR.WREG	1	1	N和Z
	IOR f,WREG	WREG = f .IOR. WREG	1	1	N和Z
	IOR #lit10,Wn	Wd = lit10 .IOR. Wd	1	1	N和Z
	IOR Wb,Ws,Wd	Wd = Wb .IOR. Ws	1	1	N和Z
	IOR Wb,#lit5,Wd	Wd = Wb .IOR. lit5	1	1	N和Z
LNK	LNK #lit14	分配堆栈帧	1	1	无
LSR	LSR f	f = 逻辑右移f	1	1	C、N、OV和Z
	LSR f,WREG	WREG = 逻辑右移f	1	1	C、N、OV和Z
	LSR Ws,Wd	Wd = 逻辑右移Ws	1	1	C、N、OV和Z
	LSR Wb,Wns,Wnd	Wnd = 将Wb逻辑右移Wns位	1	1	N和Z
	LSR Wb,#lit5,Wnd	Wnd = 将Wb逻辑右移lit5位	1	1	N和Z
MOV	MOV f,Wn	将f中的内容送入Wn	1	1	无
	MOV [Wns+Slit10],Wnd	将[Wns+Slit10]中的内容送入Wnd	1	1	无
	MOV f	将f中的内容送入f	1	1	N和Z
	MOV f,WREG	将f中的内容送入WREG	1	1	N和Z
	MOV #lit16,Wn	将16位立即数送入Wn	1	1	无
	MOV.b #lit8,Wn	将8位立即数送入Wn	1	1	无
	MOV Wn,f	将Wn中的内容送入f	1	1	无
	MOV Wns,[Wns+Slit10]	将Wns中的内容送入[Wns+Slit10]	1	1	无
	MOV Wso,Wdo	将Ws中的内容送入Wd	1	1	无
	MOV WREG,f	将WREG中的内容送入f	1	1	N和Z
	MOV.D Wns,Wd	将W(ns):W(ns + 1)中的双字内容送入Wd	1	2	无
	MOV.D Ws,Wnd	将Ws中的双字内容送入W(nd + 1):W(nd)	1	2	无
MUL	MUL.SS Wb,Ws,Wnd	{Wnd+1, Wnd} = 有符号(Wb) * 有符号(Ws)	1	1	无
	MUL.SU Wb,Ws,Wnd	{Wnd+1, Wnd} = 有符号(Wb) * 无符号(Ws)	1	1	无
	MUL.US Wb,Ws,Wnd	{Wnd+1, Wnd} = 无符号(Wb) * 有符号(Ws)	1	1	无
	MUL.UU Wb,Ws,Wnd	{Wnd+1, Wnd} = 无符号(Wb) * 无符号(Ws)	1	1	无
	MUL.SU Wb,#lit5,Wnd	{Wnd+1, Wnd} = 有符号(Wb) * 无符号(lit5)	1	1	无
	MUL.UU Wb,#lit5,Wnd	{Wnd+1, Wnd} = 无符号(Wb) * 无符号(lit5)	1	1	无
	MUL f	W3:W2 = f * WREG	1	1	无
NEG	NEG f	f = $\overline{f}$ + 1	1	1	C、DC、N、OV和Z
	NEG f,WREG	WREG = $\overline{f}$ + 1	1	1	C、DC、N、OV和Z
	NEG Ws,Wd	Wd = $\overline{Ws}$ + 1	1	1	C、DC、N、OV和Z
NOP	NOP	空操作	1	1	无
	NOPR	空操作	1	1	无
POP	POP f	将栈顶（Top-of-Stack，TOS）的内容弹出到f	1	1	无
	POP Wdo	将栈顶（TOS）的内容弹出到Wdo	1	1	无
	POP.D Wnd	将栈顶（TOS）的内容弹出到W(nd):W(nd+1)中	1	2	无
	POP.S	将影子寄存器的内容弹出到主寄存器	1	1	全部
PUSH	PUSH f	将f的内容压入栈顶（TOS）	1	1	无
	PUSH Wso	将Wso的内容压入栈顶（TOS）	1	1	无
	PUSH.D Wns	将W(ns):W(ns + 1)中的内容压入栈顶（TOS）	1	2	无
	PUSH.S	将主寄存器中的内容压入影子寄存器	1	1	无

表 31-2： 指令集概述 （续）

汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
PWRSAV	PWRSAV #lit1	进入休眠或空闲模式	1	1	WDTO 和 Sleep
RCALL	RCALL Expr	相对调用	1	2	无
	RCALL Wn	计算调用	1	2	无
REPEAT	REPEAT #lit14	将下一条指令重复执行 lit14 + 1 次	1	1	无
	REPEAT Wn	将下一条指令重复执行 (Wn) + 1 次	1	1	无
RESET	RESET	软件器件复位	1	1	无
RETFIE	RETFIE	从中断返回	1	3（2）	无
RETLW	RETLW #lit10,Wn	返回并将立即数存入 Wn	1	3（2）	无
RETURN	RETURN	从子程序返回	1	3（2）	无
RLC	RLC f	f = 对 f 执行带进位的循环左移	1	1	C、N 和 Z
	RLC f,WREG	WREG = 对 f 执行带进位的循环左移	1	1	C、N 和 Z
	RLC Ws,Wd	Wd = 对 Ws 执行带进位的循环左移	1	1	C、N 和 Z
RLNC	RLNC f	f = 循环左移 f（不带进位）	1	1	N 和 Z
	RLNC f,WREG	WREG = 循环左移 f（不带进位）	1	1	N 和 Z
	RLNC Ws,Wd	Wd = 循环左移 Ws（不带进位）	1	1	N 和 Z
RRC	RRC f	f = 对 f 执行带进位的循环右移	1	1	C、N 和 Z
	RRC f,WREG	WREG = 对 f 执行带进位的循环右移	1	1	C、N 和 Z
	RRC Ws,Wd	Wd = 对 Ws 执行带进位的循环右移	1	1	C、N 和 Z
RRNC	RRNC f	f = 循环右移 f（不带进位）	1	1	N 和 Z
	RRNC f,WREG	WREG = 循环右移 f（不带进位）	1	1	N 和 Z
	RRNC Ws,Wd	Wd = 循环右移 Ws（不带进位）	1	1	N 和 Z
SE	SE Ws,Wnd	Wnd = 符号扩展后的 Ws	1	1	C、N 和 Z
SETM	SETM f	f = FFFFh	1	1	无
	SETM WREG	WREG = FFFFh	1	1	无
	SETM Ws	Ws = FFFFh	1	1	无
SL	SL f	f = 左移 f	1	1	C、N、OV 和 Z
	SL f,WREG	WREG = 左移 f	1	1	C、N、OV 和 Z
	SL Ws,Wd	Wd = 左移 Ws	1	1	C、N、OV 和 Z
	SL Wb,Wns,Wnd	Wnd = 将 Wb 左移 Wns 位	1	1	N 和 Z
	SL Wb,#lit5,Wnd	Wnd = 将 Wb 左移 lit5 位	1	1	N 和 Z
SUB	SUB f	f = f – WREG	1	1	C、DC、N、OV 和 Z
	SUB f,WREG	WREG = f – WREG	1	1	C、DC、N、OV 和 Z
	SUB #lit10,Wn	Wn = Wn – lit10	1	1	C、DC、N、OV 和 Z
	SUB Wb,Ws,Wd	Wd = Wb – Ws	1	1	C、DC、N、OV 和 Z
	SUB Wb,#lit5,Wd	Wd = Wb – lit5	1	1	C、DC、N、OV 和 Z
SUBB	SUBB f	f = f – WREG – ($\overline{C}$)	1	1	C、DC、N、OV 和 Z
	SUBB f,WREG	WREG = f – WREG – ($\overline{C}$)	1	1	C、DC、N、OV 和 Z
	SUBB #lit10,Wn	Wn = Wn – lit10 – ($\overline{C}$)	1	1	C、DC、N、OV 和 Z
	SUBB Wb,Ws,Wd	Wd = Wb – Ws – ($\overline{C}$)	1	1	C、DC、N、OV 和 Z
	SUBB Wb,#lit5,Wd	Wd = Wb – lit5 – ($\overline{C}$)	1	1	C、DC、N、OV 和 Z
SUBR	SUBR f	f = WREG – f	1	1	C、DC、N、OV 和 Z
	SUBR f,WREG	WREG = WREG – f	1	1	C、DC、N、OV 和 Z
	SUBR Wb,Ws,Wd	Wd = Ws – Wb	1	1	C、DC、N、OV 和 Z
	SUBR Wb,#lit5,Wd	Wd = lit5 – Wb	1	1	C、DC、N、OV 和 Z
SUBBR	SUBBR f	f = WREG – f – ($\overline{C}$)	1	1	C、DC、N、OV 和 Z
	SUBBR f,WREG	WREG = WREG – f – ($\overline{C}$)	1	1	C、DC、N、OV 和 Z
	SUBBR Wb,Ws,Wd	Wd = Ws – Wb – ($\overline{C}$)	1	1	C、DC、N、OV 和 Z
	SUBBR Wb,#lit5,Wd	Wd = lit5 – Wb – ($\overline{C}$)	1	1	C、DC、N、OV 和 Z
SWAP	SWAP.b Wn	Wn = 将 Wn 的两个半字节相交换	1	1	无
	SWAP Wn	Wn = 将 Wn 的两个字节相交换	1	1	无

表31-2： 指令集概述 （续）

汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
TBLRDH	TBLRDH Ws,Wd	将程序存储器中某个单元的bit<23:16>读入Wd<7:0>	1	2	无
TBLRDL	TBLRDL Ws,Wd	将程序存储器中某个单元的bit<15:0>读入Wd	1	2	无
TBLWTH	TBLWTH Ws,Wd	将Ws<7:0>写入程序存储器中某个单元的bit<23:16>	1	2	无
TBLWTL	TBLWTL Ws,Wd	将Ws写入程序存储器中某个单元的bit<15:0>	1	2	无
ULNK	ULNK	释放堆栈帧	1	1	无
XOR	XOR f	f = f .XOR. WREG	1	1	N和Z
	XOR f,WREG	WREG = f .XOR. WREG	1	1	N和Z
	XOR #lit10,Wn	Wd = lit10 .XOR. Wd	1	1	N和Z
	XOR Wb,Ws,Wd	Wd = Wb .XOR. Ws	1	1	N和Z
	XOR Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N和Z
ZE	ZE Ws,Wnd	Wnd = 零扩展后的Ws	1	1	C、Z和N

注：

32.0 电气特性

本章将对PIC24FJ256GA705系列的电气特性进行概括介绍。其余信息在本文档的将来版本中提供。

下面列出了PIC24FJ256GA705系列器件的绝对最大值。器件长时间工作在最大值条件下，其可靠性可能会受到影响。我们建议不要使器件工作在最大值甚至超过最大值的条件下。

绝对最大值(†)

偏置时的环境温度 .... -40°C至+85°C
储存温度 .... -65°C至+150°C
VDD引脚相对于Vss的电压 .... -0.3V至+4.0V
任一通用数字或模拟引脚（非5.5V耐压）相对于Vss的电压 .... -0.3V至（VDD + 0.3V）
任一通用数字或模拟引脚（5.5V耐压，包括$\overline{\text{MCLR}}$）相对于Vss的电压：
　　当VDD = 0V时： .... -0.3V至+4.0V
　　当VDD ≥ 2.0V时： .... -0.3V至+6.0V
AVDD相对于Vss的电压 .... （VDD – 0.3V）至（4.0V和（VDD + 0.3V）中的较小值）
AVss相对于Vss的电压 .... -0.3V至+0.3V
流出Vss引脚的最大电流 .... 300 mA
流入VDD引脚的最大电流（注1） .... 250 mA
任一I/O引脚的最大输出灌电流 .... 25 mA
任一I/O引脚的最大输出拉电流 .... 25 mA
所有端口的最大灌电流 .... 200 mA
所有端口的最大拉电流（注1） .... 200 mA

注 1： 允许的最大电流由器件最大功耗决定（见表32-1）。

† 注：如果器件工作条件超过上述“绝对最大值”，可能对器件造成永久性损坏。上述数值仅是极限参数，我们建议不要使器件工作在最大值甚至超过最大值的条件下。器件长时间工作在最大值条件下，其可靠性可能受到影响。

32.1 直流特性

图32-1： PIC24FJ256GA705系列器件电压—频率关系图（工业级）

注 1： 工作电压下限取2.0V和V_{BOR}（使能BOR时）中的较低值。为实现最佳模拟性能，请在高于2.2V的电压下工作。

表32-1： 热工作条件

参数	符号	最小值	典型值	最大值	单位
PIC24FJ256GA705：					
工作结温范围	T_J	-40	—	+85	°C
工作环境温度范围	T_A	-40	—	+85	°C
功耗： 芯片内部功耗： $P_{INT} = V_{DD} \times (I_{DD} - \Sigma I_{OH})$ I/O引脚功耗： $P_{I/O} = \Sigma (\{V_{DD} - V_{OH}\} \times I_{OH}) + \Sigma (V_{OL} \times I_{OL})$	P_D	$P_{INT} + P_{I/O}$			W
最大允许功耗	P_{DMAX}	$(T_J - T_A)/\theta_{JA}$			W

表32-2： 热封装特性

特性	符号	典型值	最大值	单位	注
封装热阻，6x6 mm 28引脚QFN	θ_{JA}		—	°C/W	（注1）
封装热阻，4x4x0.6 mm 28引脚UQFN	θ_{JA}		—	°C/W	（注1）
封装热阻，7.50 mm 28引脚SOIC	θ_{JA}		—	°C/W	（注1）
封装热阻，5.30 mm 28引脚SSOP	θ_{JA}		—	°C/W	（注1）
封装热阻，300 mil 28引脚SPDIP	θ_{JA}		—	°C/W	（注1）
封装热阻，6x6x0.5 mm 48引脚UQFN	θ_{JA}	33.7	—	°C/W	（注1）
封装热阻，10x10x1 mm 44引脚TQFP	θ_{JA}	28	—	°C/W	（注1）
封装热阻，7x7x1 mm 48引脚TQFP	θ_{JA}	39.3	—	°C/W	（注1）

注 1： 通过封装模拟获得结与环境的热阻值Theta-JA（θ_{JA}）。

表32-3： 直流特性：温度和电压规范

直流特性			标准工作条件： 工作温度	**2.0V**至**3.6V**（除非另外声明） -40°C ≤ TA ≤ +85°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**
工作电压							
DC10	VDD	电源电压	2.0	—	3.6	V	禁止BOR
			VBOR	—	3.6	V	使能BOR
DC12	VDR	**RAM数据保持电压**(1)	取VPORREL和VBOR中的较大值	—	—	V	仅当使能BOR（BOREN = 1）时才使用VBOR
DC16	VPOR	确保内部上电复位信号的**VDD**启动电压	VSS	—	—	V	（**注2**）
DC17A	SVDD	确保内部上电复位信号的**推荐VDD**上升速率	1V/20 ms	—	1V/10 µS	sec	（**注2**和**注4**）
DC17B	VBOR	VDD由高到低跳变时的欠压复位电压	2.0	2.1	2.2	V	（**注3**）

注 **1：** 该电压是在确保RAM内容始终保持的条件下VDD可以降到的下限。

2： 如果不满足VPOR或SVDD参数，或应用遇到缓慢的VDD掉电斜率，建议使能并使用BOR。

3： 在出现上升的VDD上电序列时，从VPORREL或VBOR电压中较高的值开始执行应用固件（BOREN = 1时）。

4： 如果VDD上升时间在该窗之外，则无法在内部复位处理器，并且不会通过参数化测试。

表32-4： 直流特性：工作电流（IDD）

直流特性			标准工作条件： **2.0V至3.6V**（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）			
参数编号	典型值[1]	最大值	单位	工作温度	VDD	条件
工作电流（IDD）[2]						
DC19	230	365	μA	-40°C至+85°C	2.0V	0.5 MIPS， FOSC = 1 MHz
	250	365	μA	-40°C至+85°C	3.3V	
DC20	430	640	μA	-40°C至+85°C	2.0V	1 MIPS， FOSC = 2 MHz
	440	640	μA	-40°C至+85°C	3.3V	
DC23	1.5	2.4	mA	-40°C至+85°C	2.0V	4 MIPS， FOSC = 8 MHz
	1.65	2.4	mA	-40°C至+85°C	3.3V	
DC24	6.1	7.7	mA	-40°C至+85°C	2.0V	16 MIPS， FOSC = 32 MHz
	6.3	7.7	mA	-40°C至+85°C	3.3V	
DC31	43	130	μA	-40°C至+85°C	2.0V	LPRC（15.5 KIPS）， FOSC = 31 kHz
	46	130	μA	-40°C至+85°C	3.3V	
DC32	1.6	2.5	mA	-40°C至+85°C	2.0V	FRC（4 MIPS）， FOSC = 8 MHz
	1.65	2.5	mA	-40°C至+85°C	3.3V	

注 1： 除非另外声明，否则“典型值”栏中的数据均为3.3V和+25°C条件下的值。典型参数仅作为设计参考，未经测试。

2： 所有IDD测量值的测试条件如下：OSC1使用轨到轨的外部方波驱动。所有I/O引脚均配置为输出并驱动为低电平。$\overline{\text{MCLR}}$ = VDD； WDT和FSCM被禁止。CPU、程序存储器和数据存储器处于工作状态。所有外设模块都不工作，也不为其提供时钟（所有已定义的PMDx位均置1）。禁止JTAG接口。

表32-5： 直流特性：空闲电流（IIDLE）

直流特性			标准工作条件： **2.0V至3.6V**（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）			
参数编号	典型值[(1)]	最大值	单位	工作温度	VDD	条件
空闲电流（IIDLE）[(2)]						
DC40	95	400	μA	-40°C至+85°C	2.0V	1 MIPS， FOSC = 2 MHz
	105	400	μA	-40°C至+85°C	3.3V	
DC43	290	1200	μA	-40°C至+85°C	2.0V	4 MIPS， FOSC = 8 MHz
	315	1200	μA	-40°C至+85°C	3.3V	
DC47	1.05	3.7	mA	-40°C至+85°C	2.0V	16 MIPS， FOSC = 32 MHz
	1.16	3.7	mA	-40°C至+85°C	3.3V	
DC50	350	1100	μA	-40°C至+85°C	2.0V	FRC（4 MIPS）， FOSC = 8 MHz
	360	1100	μA	-40°C至+85°C	3.3V	
DC51	29	110	μA	-40°C至+85°C	2.0V	LPRC（15.5 KIPS）， FOSC = 31 kHz
	33	110	μA	-40°C至+85°C	3.3V	

注 1： 除非另外声明，否则“典型值”栏中的数据均为3.3V和+25°C条件下的值。这些参数仅供设计参考，未经测试。

2： 测量基本IIDLE电流时，内核关闭、时钟开启且所有模块均关闭。外设模块禁止SFR寄存器为全1。所有I/O引脚均配置为输出并驱动为低电平。禁止JTAG接口。

表32-6： 直流特性：掉电电流（IPD）

直流特性			标准工作条件： 2.0V至3.6V（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）			
参数编号	典型值[1]	最大值	单位	工作温度	VDD	条件
掉电电流[4,5]						
DC60	2.5	10	μA	-40°C	2.0V	休眠[2]
	3.2	10	μA	+25°C		
	11.5	45	μA	+85°C		
	3.2	10	μA	-40°C	3.3V	
	4.4	10	μA	+25°C		
	12.2	45	μA	+85°C		
DC61	165	—	nA	-40°C	2.0V	低压保持休眠[3]
	190	—	nA	+25°C		
	14.5	—	μA	+85°C		
	220	—	nA	-40°C	3.3V	
	300	—	nA	+25°C		
	15	—	μA	+85°C		

注 1： 除非另外声明，否则“典型值”栏中的数据均为3.3V和+25°C条件下的值。这些参数仅供设计参考，未经测试。

2： 禁止保持低压稳压器；RETEN（RCON<12>）= 0，$\overline{\text{LPCFG}}$（FPOR<2>）= 1。

3： 使能保持低压稳压器；RETEN（RCON<12>）= 1，$\overline{\text{LPCFG}}$（FPOR<2>）= 0。

4： 测量基本IPD电流时，所有外设和时钟均关闭。所有I/O均配置为输出并驱动为低电平。WDT、BOR和JTAG均被禁止。

5： 这些电流均在该系列中存储容量最大的器件上测得。

表32-7：　直流特性：Δ电流（BOR、WDT、HLVD和RTCC）[3]

直流特性			标准工作条件：**2.0V至3.6V**（除非另外声明） 工作温度　-40°C ≤ TA ≤ +85°C（工业级）			
参数编号	典型值[1]	最大值	单位	工作温度	VDD	条件
欠压复位（Δ**BOR**）增加电流[2]						
DC25	3	5	μA	-40°C至+85°C	2.0V	ΔBOR[2]
	4	5	μA	-40°C至+85°C	3.3V	
看门狗定时器（Δ**WDT**）增加电流[2]						
DC71	220	1000	nA	-40°C至+85°C	2.0V	ΔWDT[2]
	300	1000	nA	-40°C至+85°C	3.3V	
高/低压检测（Δ**HLVD**）增加电流[2]						
DC75	1.3	5	μA	-40°C至+85°C	2.0V	ΔHLVD[2]
	1.9	5	μA	-40°C至+85°C	3.3V	
实时时钟和日历（Δ**RTCC**）增加电流[2]						
DC77	2.5	—	μA	-40°C至+85°C	2.0V	ΔRTCC（在低功耗模式下使能SOSC）[2]
	3	—	μA	-40°C至+85°C	3.3V	
DC77A	350	1000	nA	-40°C至+85°C	2.0V	ΔRTCC（使能LPRC）[2]
	400	1000	nA	-40°C至+85°C	3.3V	

注 1：除非另外声明，否则“典型值”栏中的数据均为3.3V和+25°C条件下的值。这些参数仅供设计参考，未经测试。

2：模块使能和运行期间增加的电流。

3：Δ电流为当模块使能时额外消耗的电流。此电流应被加到基本IPD电流。此电流涉及为WDT和RTCC使能的 所选时钟源。

表 32-8： 直流特性：I/O 引脚输入规范

直流特性			标准工作条件： 2.0V 至 3.6V（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）				
参数编号	符号	特性	最小值	典型值[(1)]	最大值	单位	条件
	VIL	输入低电压[(3)]					
DI10		具有 ST 缓冲器的 I/O 引脚	VSS	—	0.2 VDD	V	
DI11		具有 TTL 缓冲器的 I/O 引脚	VSS	—	0.15 VDD	V	
DI15		$\overline{MCLR}$	VSS	—	0.2 VDD	V	
DI16		OSCI（XT 模式）	VSS	—	0.2 VDD	V	
DI17		OSCI（HS 模式）	VSS	—	0.2 VDD	V	
DI18		具有 I^2C 缓冲器的 I/O 引脚	VSS	—	0.3 VDD	V	
DI19		具有 SMBus 缓冲器的 I/O 引脚	VSS	—	0.8	V	使能 SMBus
	VIH	输入高电压[(3)]					
DI20		具有 ST 缓冲器的 I/O 引脚：					
		具有模拟功能，	0.8 VDD	—	VDD	V	
		仅数字	0.8 VDD	—	5.5	V	
DI21		具有 TTL 缓冲器的 I/O 引脚：					
		具有模拟功能，	0.25 VDD + 0.8	—	VDD	V	
		仅数字	0.25 VDD + 0.8	—	5.5	V	
DI25		$\overline{MCLR}$	0.8 VDD	—	VDD	V	
DI26		OSCI（XT 模式）	0.7 VDD	—	VDD	V	
DI27		OSCI（HS 模式）	0.7 VDD	—	VDD	V	
DI28		具有 I^2C 缓冲器的 I/O 引脚：					
		具有模拟功能，	0.7 VDD	—	VDD	V	
		仅数字	0.7 VDD	—	5.5	V	
DI29		具有 SMBus 缓冲器的 I/O 引脚：					
		具有模拟功能，	2.1	—	VDD	V	2.5V ≤ VPIN ≤ VDD
		仅数字	2.1	—	5.5	V	
DI30	ICNPU	**CNx** 上拉电流	150	—	450	µA	VDD = 3.3V，VPIN = VSS
DI30A	ICNPD	**CNx** 下拉电流	230	—	500	µA	VDD = 3.3V，VPIN = VDD
	IIL	输入泄漏电流[(2)]					
DI50		I/O 端口	—	—	±1	µA	VSS ≤ VPIN ≤ VDD，引脚处于高阻态
DI51		模拟输入引脚	—	—	±1	µA	VSS ≤ VPIN ≤ VDD，引脚处于高阻态
DI55		$\overline{MCLR}$	—	—	±1	µA	VSS ≤ VPIN ≤ VDD
DI56		OSCI/CLKI	—	—	±1	µA	VSS ≤ VPIN ≤ VDD，EC、XT 和 HS 模式

注 1： 除非另外声明，否则“典型值”栏中的数据均为 3.3V 和 +25°C 条件下的值。这些参数仅供设计参考，未经测试。

2： 负电流定义为引脚的拉电流。

3： 关于 I/O 引脚缓冲器类型，请参见表 1-1。

表32-9： 直流特性：I/O引脚输出规范

直流特性			标准工作条件： **2.0V**至**3.6V**（除非另外声明） 工作温度 -40°C ≤ T_A ≤ +85°C（工业级）				
参数编号	符号	特性	最小值	典型值[1]	最大值	单位	条件
	V_{OL}	输出低电压					
DO10		I/O端口	—	—	0.4	V	I_{OL} = 6.6 mA，V_{DD} = 3.6V
			—	—	0.8	V	I_{OL} = 18 mA，V_{DD} = 3.6V
			—	—	0.35	V	I_{OL} = 5.0 mA，V_{DD} = 2V
DO16		OSCO/CLKO	—	—	0.18	V	I_{OL} = 6.6 mA，V_{DD} = 3.6V
			—	—	0.2	V	I_{OL} = 5.0 mA，V_{DD} = 2V
	V_{OH}	输出高电压					
DO20		I/O端口	3.4	—	—	V	I_{OH} = -3.0 mA，V_{DD} = 3.6V
			3.25	—	—	V	I_{OH} = -6.0 mA，V_{DD} = 3.6V
			2.8	—	—	V	I_{OH} = -18 mA，V_{DD} = 3.6V
			1.65	—	—	V	I_{OH} = -1.0 mA，V_{DD} = 2V
			1.4	—	—	V	I_{OH} = -3.0 mA，V_{DD} = 2V
DO26		OSCO/CLKO	3.3	—	—	V	I_{OH} = -6.0 mA，V_{DD} = 3.6V
			1.85	—	—	V	I_{OH} = -1.0 mA，V_{DD} = 2V

注 1： 除非另外声明，否则“典型值”栏中的数据均为3.3V和+25°C条件下的值。这些参数仅供设计参考，未经测试。

表32-10： 直流特性：程序存储器

直流特性			标准工作条件： **2.0V**至**3.6V**（除非另外声明） 工作温度 -40°C ≤ T_A ≤ +85°C（工业级）				
参数编号	符号	特性	最小值	典型值[1]	最大值	单位	条件
		闪存程序存储器					
D130	E_P	单元耐擦写能力	10000	—	—	E/W	-40°C至+85°C
D131	V_{PR}	读操作时的V_{DD}	V_{MIN}	—	3.6	V	V_{MIN} = 最小工作电压
D132B		自定时写的V_{DD}	V_{MIN}	—	3.6	V	V_{MIN} = 最小工作电压
D133A	T_{IW}	自定时字写周期时间	—	20	—	µs	
		自定时行写周期时间	—	1.5	—	ms	
D133B	T_{IE}	自定时页擦除时间	20	—	40	ms	
D134	T_{RETD}	特性保持时间	20	—	—	年	假定没有违反其他规范
D135	I_{DDP}	编程时的供电电流	—	5	—	mA	

注 1： 除非另外声明，否则“典型值”栏中的数据均为3.3V和+25°C条件下的值。

表32-11： 内部稳压器规范

工作条件：-40°C < TA < +85°C（除非另外声明）							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
DVR	TVREG	稳压器启动时间	—	10	—	μs	VREGS = 0（任何POR或BOR）
DVR10	VBG	内部带隙参考电压	1.14	1.2	1.26	V	
DVR11	TBG	带隙参考启动时间	—	1	—	ms	
DVR20	VRGOUT	稳压器输出电压	1.6	1.8	2.0	V	VDD > 1.9V
DVR21	CEFC	外部滤波电容值	10	—	—	μF	建议串联电阻 < 3Ω；要求< 5Ω
DVR30	VLVR	低压稳压器输出电压	—	1.2	—	V	RETEN = 1，$\overline{\text{LPCFG}}$ = 0

表32-12： 高/低压检测特性

工作条件：-40°C < TA < +85°C（除非另外声明）								
参数编号	符号	特性		最小值	典型值	最大值	单位	条件
DC18	VHLVD	VDD跳变时的HLVD电压	HLVDL<3:0> = 0100[1]	3.45	—	3.73	V	
			HLVDL<3:0> = 0101	3.25	—	3.58	V	
			HLVDL<3:0> = 0110	2.95	—	3.25	V	
			HLVDL<3:0> = 0111	2.75	—	3.04	V	
			HLVDL<3:0> = 1000	2.65	—	2.92	V	
			HLVDL<3:0> = 1001	2.45	—	2.70	V	
			HLVDL<3:0> = 1010	2.35	—	2.60	V	
			HLVDL<3:0> = 1011	2.25	—	2.49	V	
			HLVDL<3:0> = 1100	2.15	—	2.39	V	
			HLVDL<3:0> = 1101	2.08	—	2.28	V	
			HLVDL<3:0> = 1110	2.00	—	2.15	V	
DC101	VTHL	HLVDIN引脚跳变时的HLVD电压	HLVDL<3:0> = 1111	—	1.20	—	V	
DC105	TONLVD	HLVD模块使能时间		—	5	—	μs	POR或HLVDEN = 1

注 1： HLVD<3:0>值的跳变点（从0000到0011）未实现。

表32-13： 比较器直流规范

工作条件：2.0V < VDD < 3.6V，-40°C < TA < +85°C（除非另外声明）							
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	—	12	50	mV	（注1）
D301	VICM	输入共模电压	0	—	VDD	V	（注1）
D302	CMRR	共模抑制比	55	—	—	dB	（注1）
D306	IQCMP	每个比较器的AVDD静态电流	—	27	—	μA	使能比较器
D307	TRESP	响应时间	—	300	—	ns	（注2）
D308	TMC2OV	比较器模式改变到输出有效的时间	—	—	10	μs	
D309	IDD	工作电源电流	—	30	—	μA	AVDD = 3.3V

注 1： 这些参数为特性值，未经测试。
2： 测量条件如下：一个输入处于VDD/2电压，另一个输入从VSS跳变至VDD，40 mV步长，15 mV过驱。

表32-14： 比较器参考电压直流规范

工作条件：2.0V < VDD < 3.6V，-40°C < TA < +85°C（除非另外声明）							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
VR310	TSET	稳定时间	—	—	10	μs	（注1）
VRD311	CVRAA	绝对精度	-100	—	+100	mV	
VRD312	CVRUR	单位电阻值（R）	—	4.5	—	kΩ	

注 1： 在CVR<4:0>从11111跳变至00000时测量间隔时间。

表32-15： CTMU电流源规范

直流特性			标准工作条件： 2.0V至3.6V（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）					
参数编号	符号	特性	最小值	典型值(1)	最大值	单位	备注	条件
DCT10	IOUT1	CTMU电流源，基本范围	—	550	—	nA	CTMUCON1L<1:0> = 00(2)	2.5V < VDD < VDDMAX
DCT11	IOUT2	CTMU电流源，10x范围	—	5.5	—	μA	CTMUCON1L<1:0> = 01	
DCT12	IOUT3	CTMU电流源，100x范围	—	55	—	μA	CTMUCON1L<1:0> = 10	
DCT13	IOUT4	CTMU电流源，1000x范围	—	550	—	μA	CTMUCON1L<1:0> = 11(2)，CTMUCON1H<0> = 0	
DCT14	IOUT5	CTMU电流源，高范围	—	2.2	—	mA	CTMUCON1L<1:0> = 01，CTMUCON1H<0> = 1	
DCT21	VDELTA1	温度二极管每摄氏度电压变化	—	-1.8	—	mV/°C	电流 = 5.5 μA	
DCT22	VDELTA2	温度二极管每摄氏度电压变化	—	-1.55	—	mV/°C	电流 = 55 μA	
DCT23	VD1	正向电压	—	710	—	mV	温度 = 0ºC，电流 = 5.5 μA	
DCT24	VD2	正向电压	—	760	—	mV	温度 = 0ºC，电流 = 55 μA	

注 1： 电流微调范围的中点为标称值（CTMUCON1L<7:2> = 000000）。
2： 不要对内部温度检测二极管使用此电流范围。

32.2 交流特性和时序参数

本节定义了PIC24FJ256GA705系列的交流特性和时序参数。

表32-16： 温度和电压规范——交流

交流特性	标准工作条件： **2.0V至3.6V**（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级） 工作电压VDD范围如**第32.1节“直流特性”**中所述。

图32-2： 器件时序规范的负载条件

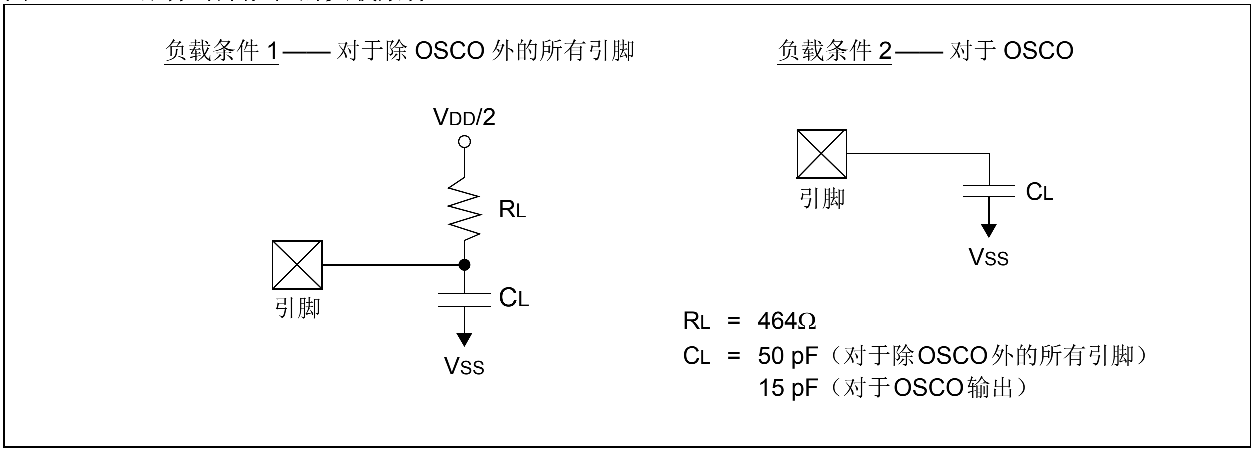


表32-17： 输出引脚上的容性负载要求

参数编号	符号	特性	最小值	典型值(1)	最大值	单位	条件
DO50	COSCO	OSCO/CLKO引脚	—	—	15	pF	当外部时钟用于驱动OSCI时，处于XT和HS模式下
DO56	CIO	所有I/O引脚和OSCO	—	—	50	pF	EC模式
DO58	CB	SCLx和SDAx	—	—	400	pF	在I^2C模式下

注 1： 除非另外声明，否则“典型值”栏中的数据均为3.3V和25°C条件下的值。这些参数仅供设计参考，未经测试。

图32-3： 外部时钟时序

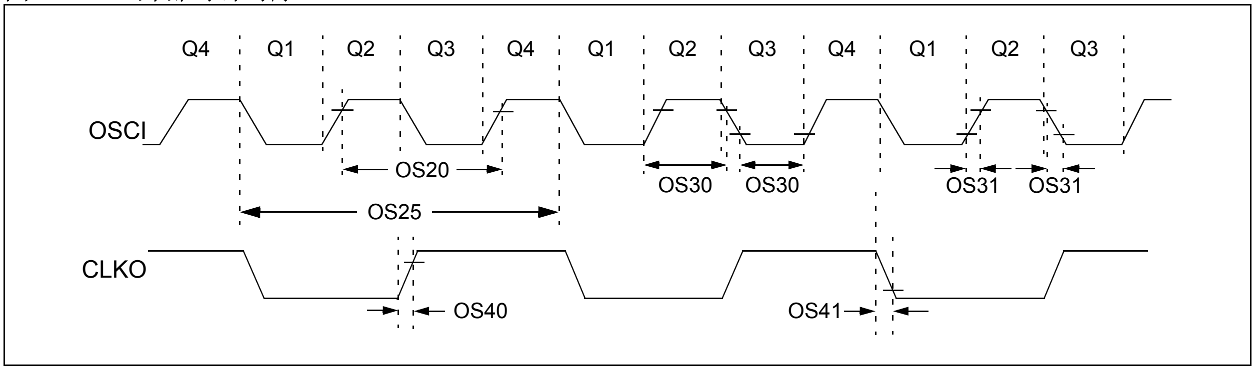


表32-18： 外部时钟时序要求

交流特性			标准工作条件： **2.0V至3.6V**（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）				
参数编号	符号	特性	最小值	典型值(1)	最大值	单位	条件
OS10	FOSC	外部CLKI频率 （外部时钟仅允许运行于EC模式）	DC 4	— —	32 48	MHz MHz	EC ECPLL（注2）
		振荡器频率	3.5 4 10 12 31	— — — — —	10 8 32 24 33	MHz MHz MHz MHz kHz	XT XTPLL HS HSPLL SOSC
OS20	TOSC	TOSC = 1/FOSC	—	—	—	—	关于FOSC的值，请参见参数OS10
OS25	TCY	指令周期(3)	62.5	—	DC	ns	
OS30	TosL，TosH	外部时钟输入（OSCI）高电平或低电平时间	0.45 x TOSC	—	—	ns	EC
OS31	TosR，TosF	外部时钟输入（OSCI）上升或下降时间	—	—	20	ns	EC
OS40	TckR	CLKO上升时间(4)	—	15	30	ns	
OS41	TckF	CLKO下降时间(4)	—	15	30	ns	

注 1： 除非另外声明，否则“典型值”栏中的数据均为3.3V和+25°C条件下的值。这些参数仅供设计参考，未经测试。

2： 表示系统时钟预分频器的输入。PLL分频器和后分频器仍然必须配置成使系统时钟频率不超出图32-1所示的最大频率。

3： 指令周期（TCY）等于输入振荡器时基周期的两倍。所有规定值均基于针对特定振荡器类型，器件在标准工作条件下执行代码时的特性数据。超出这些规定的限定值，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OSCI/CLKI引脚施加了外部时钟。当使用了外部时钟输入时，所有器件的“最大”周期时间限制为“DC”（无时钟）。

4： 测量在EC模式下进行。在OSCO引脚上测量CLKO信号。CLKO在Q1-Q2周期（1/2 TCY）中为低电平，在Q3-Q4周期（1/2 TCY）中为高电平。

表32-19： 锁相环模式的交流规范

交流特性		标准工作条件： 2.0V至3.6V（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）				
符号	特性	最小值	典型值	最大值	单位	条件
FIN	输入频率范围	2	—	24	MHz	
FMIN	来自倍频器的最小输出频率	—	—	16	MHz	4 MHz FIN，4x反馈比， 2 MHz FIN，8x反馈比
FMAX	来自倍频器的最大输出频率	96	—	—	MHz	4 MHz FIN，24x净倍频比， 24 MHz FIN，4x净倍频比
FSLEW	用于确保PLL保持锁定的FIN最大阶跃函数	-4	—	+4	%	FIN全输入范围
TLOCK	VCO的锁定时间	—	—	24	µs	在指定的最小值TREF和一个周期的锁定定时器计数条件下，这是支持的最大VCO锁定时间
JFM8	在PLL输出的任何八个连续周期内倍频器随电压和温度的累积抖动	—	—	±0.12	%	4 MHz FIN，4x反馈比

表32-20： 内部RC精度

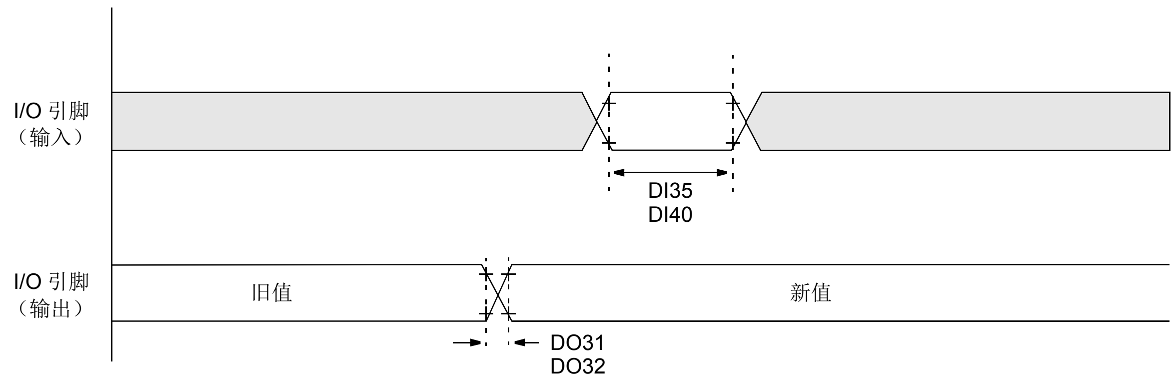
交流特性		标准工作条件： 2.0V至3.6V（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）				
参数编号	特性	最小值	典型值	最大值	单位	条件
F20	频率为8 MHz时的FRC精度	-1.5	+0.15	1.5	%	2.0V ≤ VDD ≤ 3.6V, 0°C ≤ TA ≤ +85°C（注1）
		-2	—	2	%	2.0V ≤ VDD ≤ 3.6V，-40°C ≤ TA ≤ 0°C
F21	频率为31 kHz时的LPRC	-20	—	20	%	VCAP输出电压 = 1.8V
F22	OSCTUN步长	—	0.1	—	%/位	

注 1： 要达到这一精度，单片机封装所受到的物理应力（例如：弯曲PCB）必须保持最小。

表32-21： RC振荡器起振时间

交流特性			标准工作条件： 2.0V至3.6V（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°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
FR0	TFRC	FRC振荡器起振时间	—	15	—	µs	
FR1	TLPRC	低功耗RC振荡器起振时间	—	50	—	µs	

图32-4： CLKO和I/O时序特性



注： 负载条件请参见图32-2。

表32-22： CLKO和I/O时序要求

交流特性			标准工作条件： **2.0V至3.6V**（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）				
参数编号	符号	特性	最小值	典型值(1)	最大值	单位	条件
DO31	TIOR	端口输出上升时间	—	10	25	ns	
DO32	TIOF	端口输出下降时间	—	10	25	ns	
DI35	TINP	INTx引脚高电平或低电平时间（输入）	1	—	—	TCY	
DI40	TRBP	CNx高电平或低电平时间（输入）	1	—	—	TCY	

注 1： 除非另外声明，否则“典型值”栏中的数据均为3.3V和+25°C条件下的值。

表32-23： 复位和欠压复位要求

交流特性			标准工作条件： **2.0V**至**3.6V**（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°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
SY10	TMCL	$\overline{\text{MCLR}}$脉冲宽度（低电平）	2	—	—	μs	
SY12	TPOR	上电复位延时	—	2	—	μs	
SY13	TIOZ	自$\overline{\text{MCLR}}$低电平或看门狗定时器复位起I/O处于高阻态的时间	取（3 TCY + 2）和700中的较小值	—	（3 TCY + 2）	μs	
SY25	TBOR	欠压复位脉冲宽度	1	—	—	μs	VDD ≤ VBOR
SY45	TRST	内部状态复位时间	—	50	—	μs	
SY71	TPM	程序存储器唤醒时间	—	20	—	μs	VREGS = 1时休眠唤醒
			—	1	—	μs	VREGS = 0时休眠唤醒
SY72	TLVR	低压稳压器唤醒时间	—	90	—	μs	VREGS = 1时休眠唤醒
			—	70	—	μs	VREGS = 0时休眠唤醒

表32-24： A/D模块规范

交流特性			标准工作条件： 2.0V至3.6V（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°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
器件电源							
AD01	AVDD	模块电源VDD	取VDD – 0.3和2.2中的较大值	—	取VDD + 0.3和3.6中的较小值	V	
AD02	AVSS	模块电源VSS	VSS – 0.3	—	VSS + 0.3	V	
参考电压输入							
AD05	VREFH	参考电压高电压	AVSS + 1.7	—	AVDD	V	
AD06	VREFL	参考电压低电压	AVSS	—	AVDD – 1.7	V	
AD07	VREF	绝对参考电压	AVSS – 0.3	—	AVDD + 0.3	V	
模拟输入							
AD10	VINH-VINL	满量程输入范围	VREFL	—	VREFH	V	（注2）
AD11	VIN	绝对输入电压	AVSS – 0.3	—	AVDD + 0.3	V	
AD12	VINL	绝对VINL输入电压	AVSS – 0.3	—	AVDD/3	V	
AD13		泄漏电流	—	±1.0	±610	nA	VINL = AVSS = VREFL = 0V，AVDD = VREFH = 3V，源阻抗 = 2.5 kΩ
AD17	RIN	模拟电压源的推荐阻抗	—	—	2.5k	Ω	10位
A/D精度							
AD20B	Nr	分辨率	—	12	—	位	
AD21B	INL	积分非线性误差	—	±1	< ±2	LSb	VINL = AVSS = VREFL = 0V，AVDD = VREFH = 3V
AD22B	DNL	微分非线性误差	—	—	< ±1	LSb	VINL = AVSS = VREFL = 0V，AVDD = VREFH = 3V
AD23B	GERR	增益误差	—	±1	±4	LSb	VINL = AVSS = VREFL = 0V，AVDD = VREFH = 3V
AD24B	EOFF	失调误差	—	±1	±2	LSb	VINL = AVSS = VREFL = 0V，AVDD = VREFH = 3V
AD25B		单调性[1]	—	—	—	—	保证

注 1： A/D转换结果不会因输入电压的增加而减小。

2： 测量是在使用外部VREF+和VREF-作为A/D参考电压的情况下进行的。

表32-25： A/D转换时序要求[1]

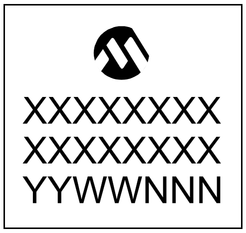
交流特性			标准工作条件： 2.0V至3.6V（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
时钟参数							
AD50	TAD	A/D时钟周期	278	—	—	ns	
AD51	tRC	A/D内部RC振荡器周期	—	250	—	ns	
转换速率							
AD55	tCONV	SAR转换时间，12位模式	—	14	—	TAD	
AD55A		SAR转换时间，10位模式为典型值12 TAD	—	12	—	TAD	
AD56	FCNV	吞吐率	—	—	200	ksps	AVDD > 2.7V[2]
AD57	tSAMP	采样时间	—	1	—	TAD	
时钟同步							
AD61	tPSS	从置1采样位（SAMP）到采样启动的延时	1.5	—	2.5	TAD	

注 1： 因为采样电容最终将释放电荷，因此低于10 kHz的时钟频率可能会影响线性性能，尤其是在温度较高时。
2： 吞吐率基于AD55 + AD57 + AD61和TAD的周期。

33.0 封装信息

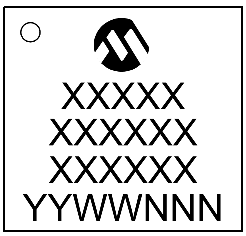
33.1 封装标识信息

28 引脚 QFN（6x6 mm）

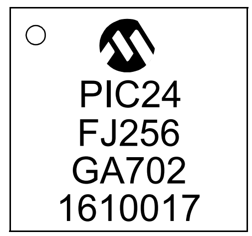


示例

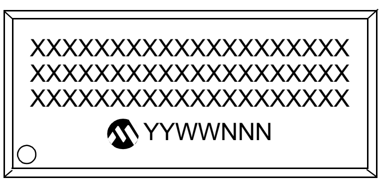
28 引脚 UQFN（4x4x0.6 mm）



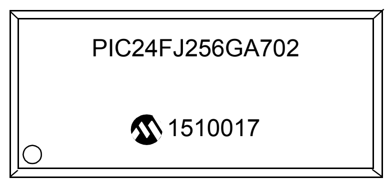
示例



28 引脚 SOIC（7.50 mm）



示例



28 引脚 SSOP（5.30 mm）

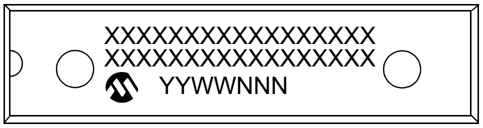
示例

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

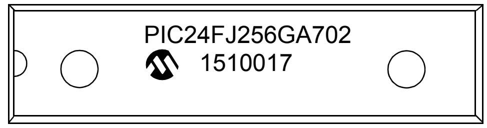
注： Microchip 部件编号如果无法在同一行内完整标注，将换行标出，因此会限制表示客户指定信息的字符数。

33.1 封装标识信息（续）

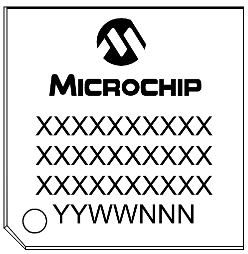
28 引脚 SPDIP（300 mil）



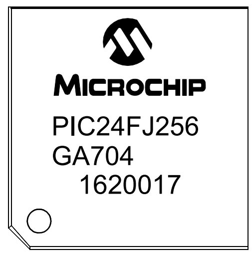
示例



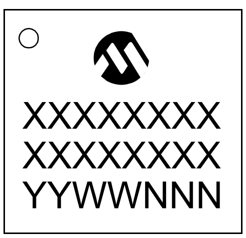
44 引脚 TQFP（10x10x1 mm）



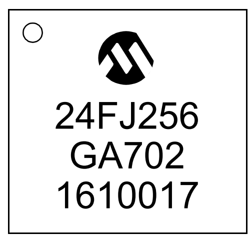
示例



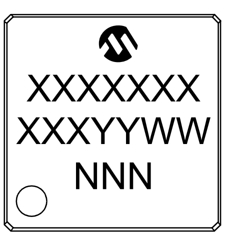
48 引脚 UQFN（6x6 mm）



示例



48 引脚 TQFP（7x7x1.0 mm）



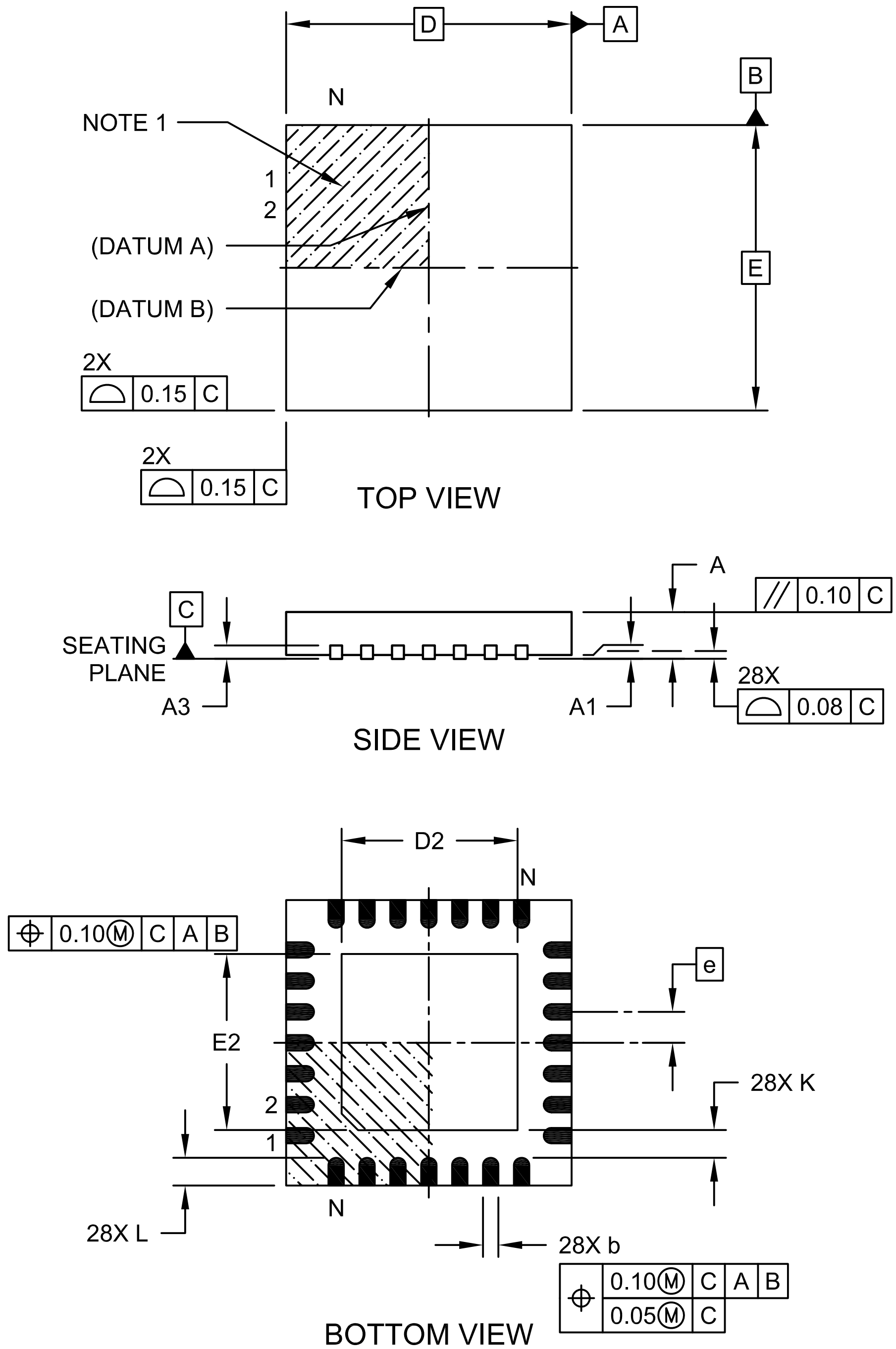
示例

33.2 封装详细信息

以下部分将介绍各种封装的技术细节。

28引脚塑封正方扁平无脚封装（ML）——主体6x6 mm [QFN]，端子长度为0.55 mm

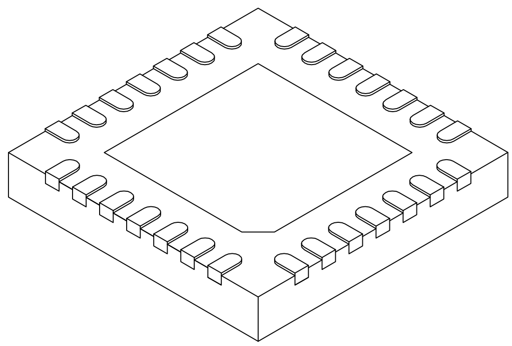
注：	最新封装图请至http://www.microchip.com/packaging查看Microchip封装规范。



Microchip Technology Drawing C04-105C Sheet 1 of 2

28引脚塑封正方扁平无脚封装（ML）——主体6x6 mm [QFN]，端子长度为0.55 mm

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 Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.20 REF	
Overall Width	E		6.00 BSC	
Exposed Pad Width	E2	3.65	3.70	4.20
Overall Length	D		6.00 BSC	
Exposed Pad Length	D2	3.65	3.70	4.20
Terminal Width	b	0.23	0.30	0.35
Terminal Length	L	0.50	0.55	0.70
Terminal-to-Exposed Pad	K	0.20	-	-

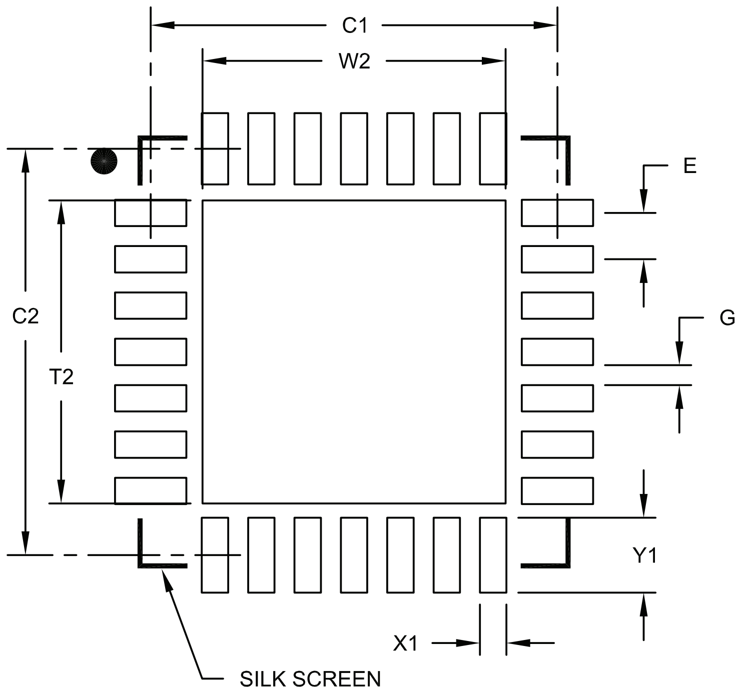
Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. 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 C04-105C Sheet 2 of 2

28引脚塑封正方扁平无脚封装（ML）——主体6x6 mm [QFN]，触点长度为0.55 mm

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		0.65 BSC	
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
Distance Between Pads	G	0.20		

Notes:

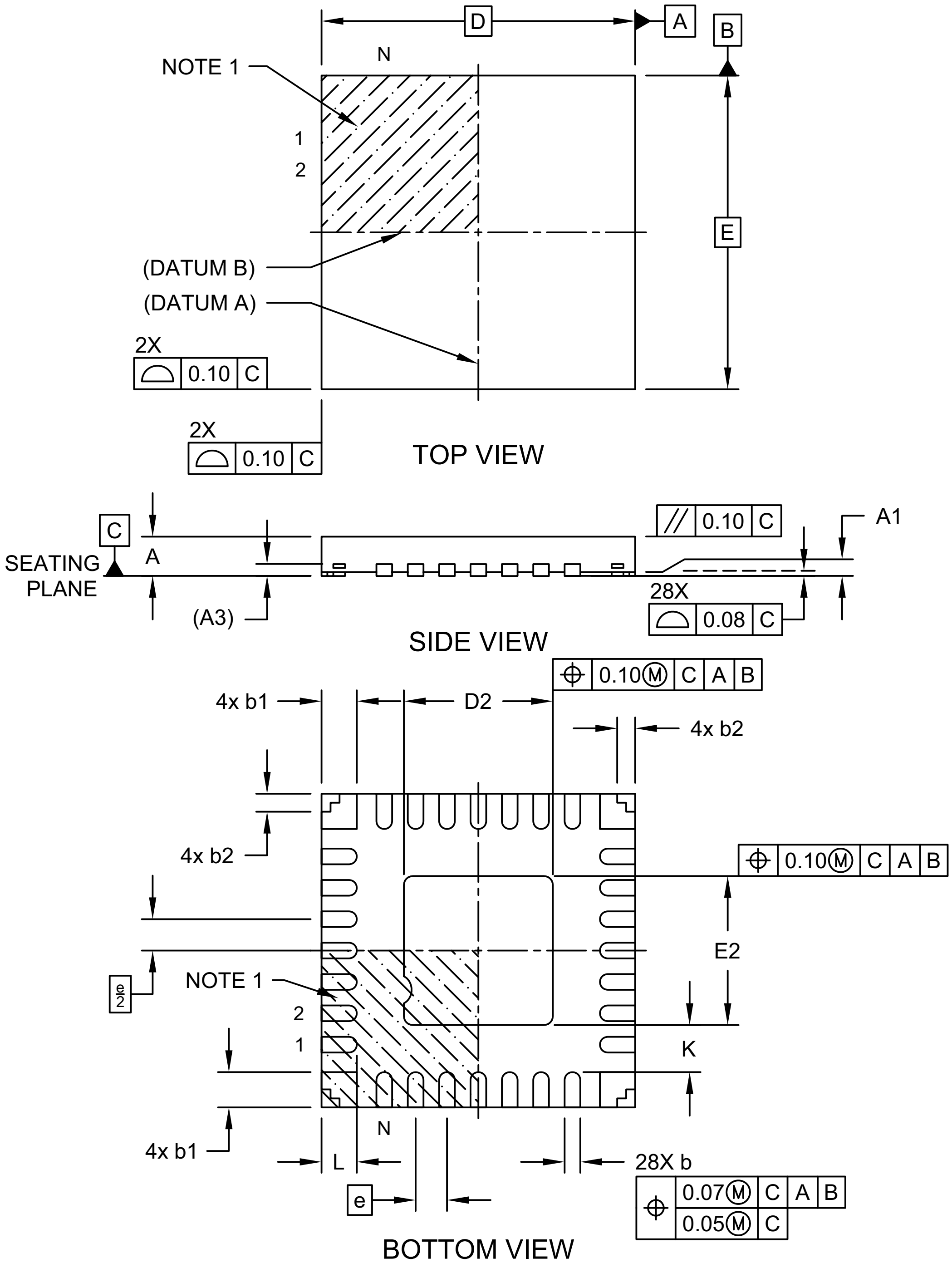
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2105A

28引脚塑封超薄正方扁平无脚封装（M6）——主体4x4x0.6 mm[UQFN]，带角锚点

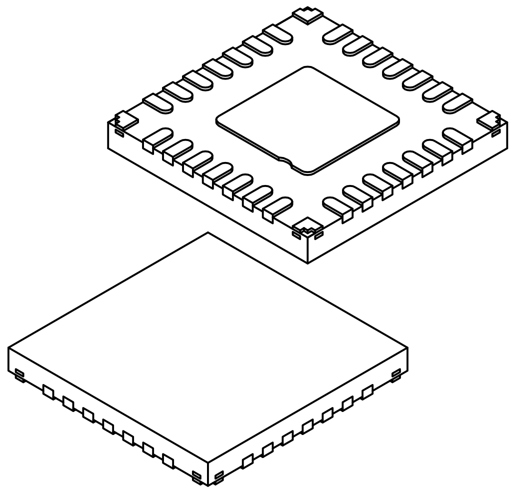
注：	最新封装图请至http://www.microchip.com/packaging查看Microchip封装规范。



Microchip Technology Drawing C04-333-M6 Rev B Sheet 1 of 2

28引脚塑封超薄正方扁平无脚封装（M6）——主体4x4x0.6 mm [UQFN]，带角锚点

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 Pins	N		28	
Pitch	e		0.40 BSC	
Overall Height	A	-	-	0.60
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.152 REF	
Overall Width	E		4.00 BSC	
Exposed Pad Width	E2	1.80	1.90	2.00
Overall Length	D		4.00 BSC	
Exposed Pad Length	D2	1.80	1.90	2.00
Terminal Width	b	0.15	0.20	0.25
Corner Anchor Pad	b1	0.40	0.45	0.50
Corner Pad, Metal Free Zone	b2	0.18	0.23	0.28
Terminal Length	L	0.30	0.45	0.50
Terminal-to-Exposed-Pad	K	-	0.60	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

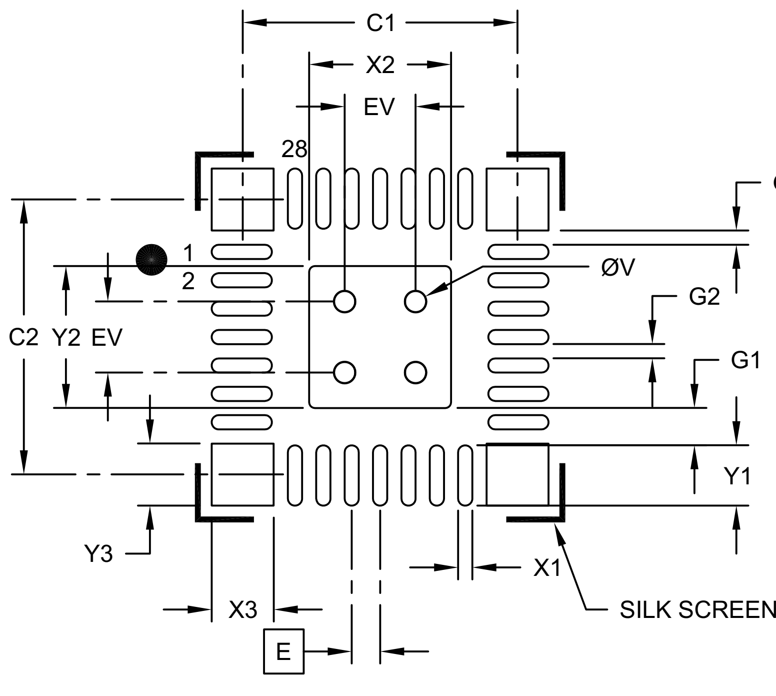
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-333-M6 Rev A Sheet 2 of 2

28引脚塑封超薄正方扁平无脚封装（M6）——主体4x4x0.6 mm [UQFN]，带角锚点

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		0.40 BSC	
Center Pad Width	X2			2.00
Center Pad Length	Y2			2.00
Contact Pad Spacing	C1		3.90	
Contact Pad Spacing	C2		3.90	
Contact Pad Width (X28)	X1			0.20
Contact Pad Length (X28)	Y1			0.85
Contact Pad to Center Pad (X28)	G1		0.52	
Contact Pad to Pad (X24)	G2	0.20		
Contact Pad to Corner Pad (X8)	G3	0.20		
Corner Anchor Width (X4)	X3			0.78
Corner Anchor Length (X4)	Y3			0.78
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

Notes:

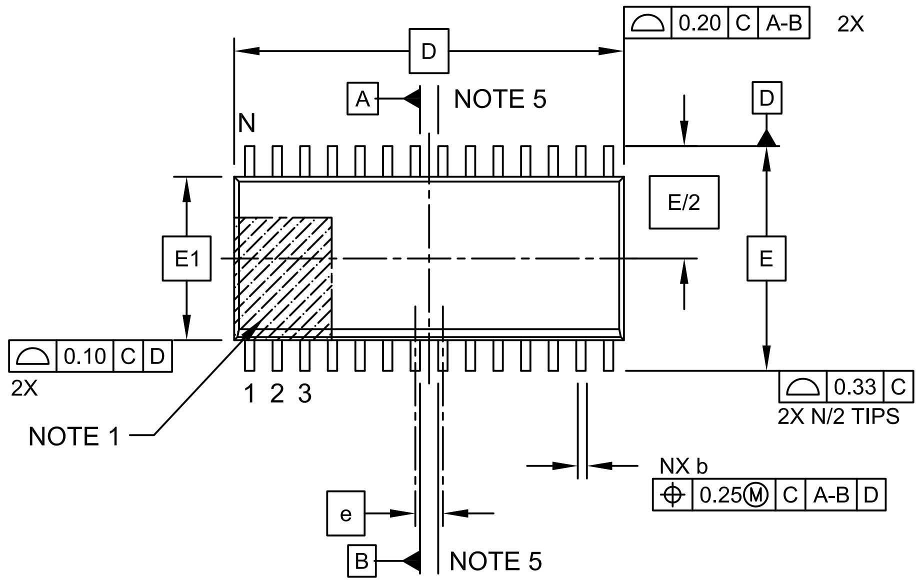
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

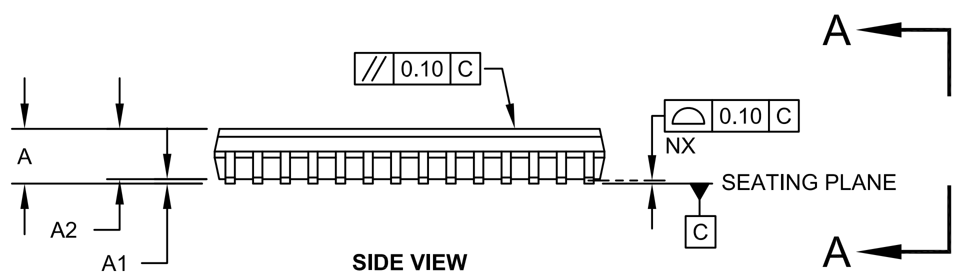
Microchip Technology Drawing C04-2333-M6 Rev B

28引脚塑封宽条小外形封装（SO）——主体7.50 mm [SOIC]

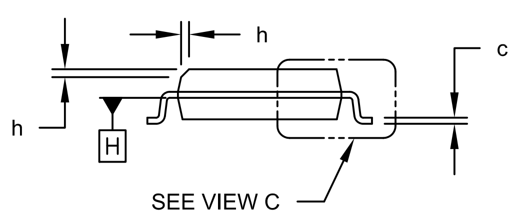
注：	最新封装图请至http://www.microchip.com/packaging查看Microchip封装规范。



TOP VIEW



SIDE VIEW

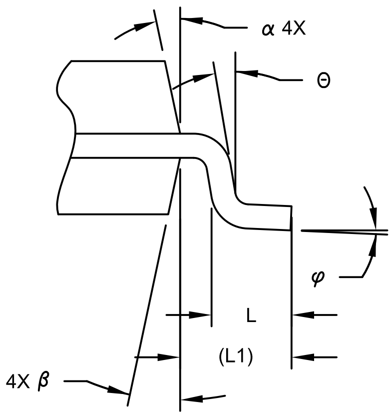


VIEW A-A

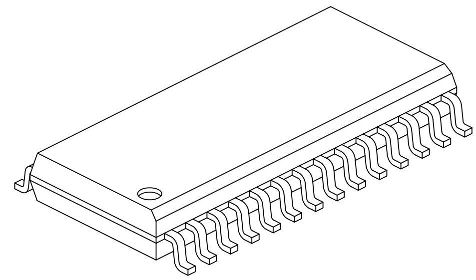
Microchip Technology Drawing C04-052C Sheet 1 of 2

28引脚塑封宽条小外形封装（SO）——主体7.50 mm [SOIC]

注： 最新封装图请至http://www.microchip.com/packaging查看Microchip封装规范。



VIEW C



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		17.90 BSC	
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1		1.40 REF	
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.18	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

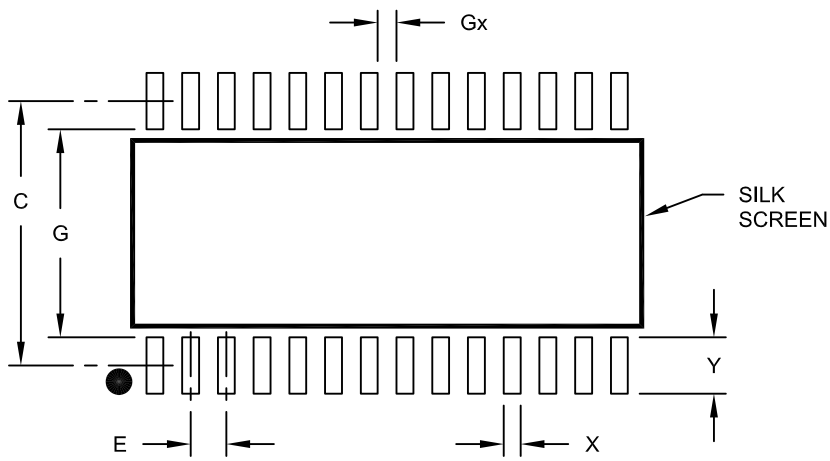
Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. 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.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2

28引脚塑封宽条小外形封装（SO）——主体7.50 mm [SOIC]

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		1.27 BSC	
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

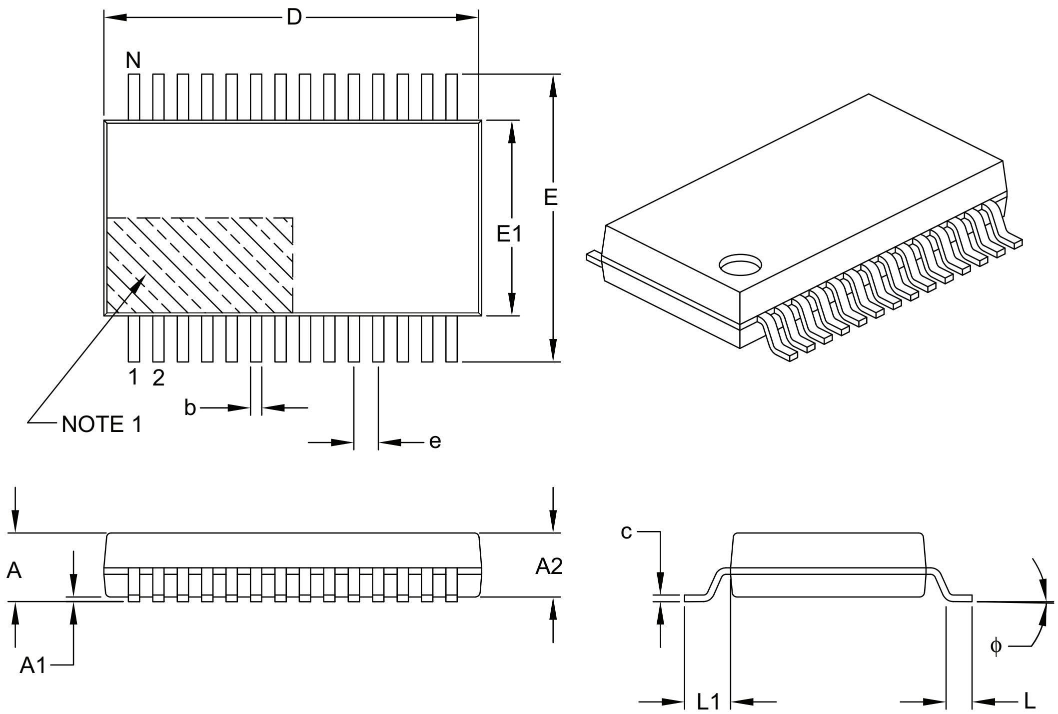
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2052A

28引脚塑封紧缩小外形封装（SS）——主体5.30 mm [SSOP]

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 Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	φ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

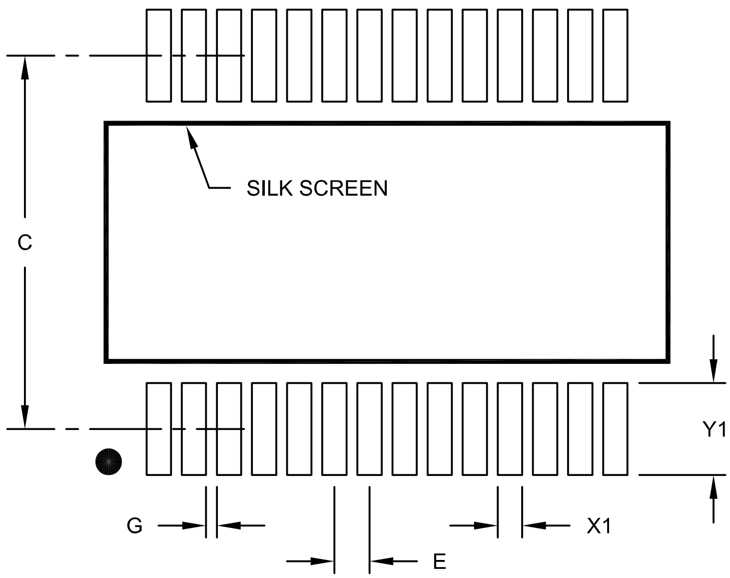
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

28引脚塑封紧缩小外形封装（SS）——主体5.30 mm [SSOP]

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		0.65 BSC	
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

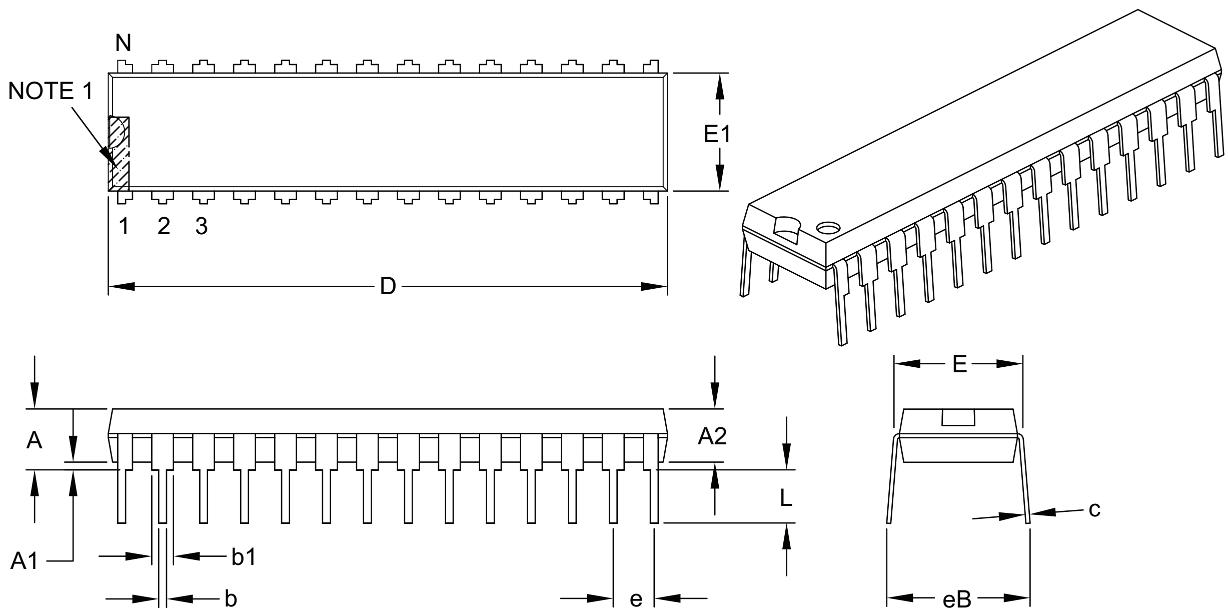
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2073A

28引脚塑封窄型双列直插式封装（SP）——主体300 mil [SPDIP]

注：	最新封装图请至http://www.microchip.com/packaging查看Microchip封装规范。



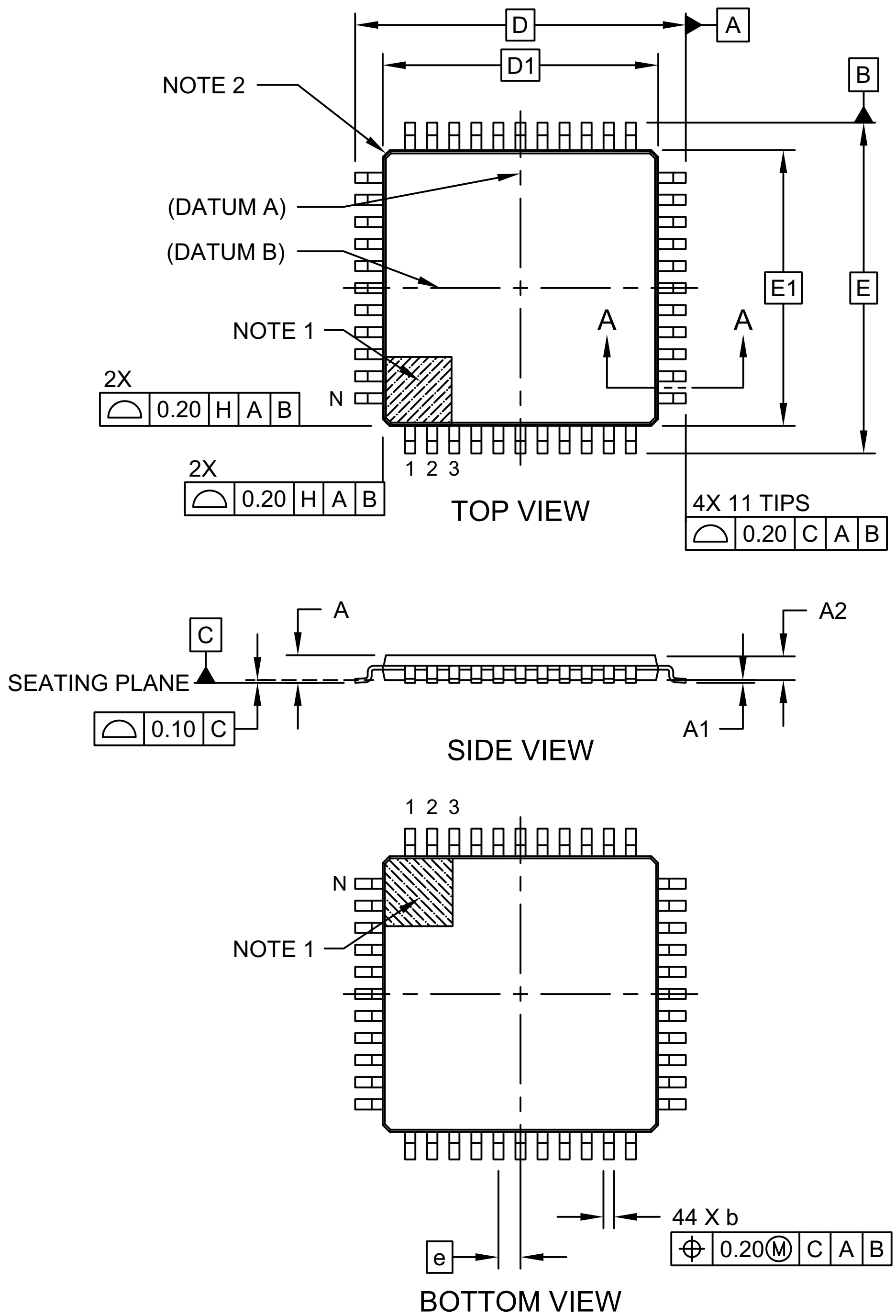
		Units	INCHES	
	Dimension Limits	MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B

44引脚塑封薄型正方扁平封装（PT）——主体10x10x1.0 mm [TQFP]

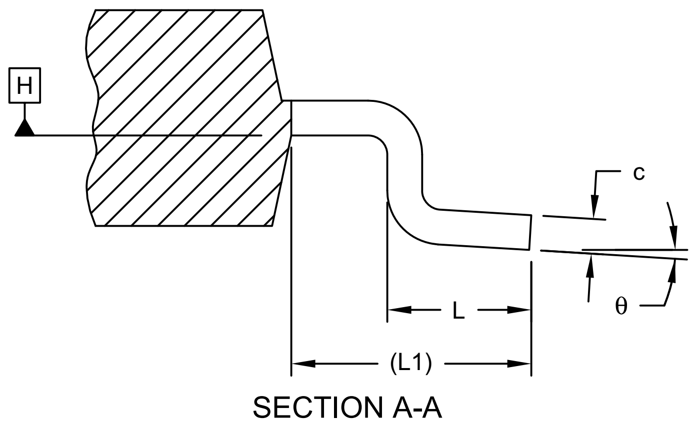
注：	最新封装图请至http://www.microchip.com/packaging查看Microchip封装规范。

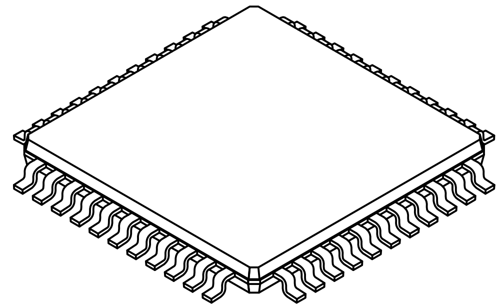


Microchip Technology Drawing C04-076C Sheet 1 of 2

44引脚塑封薄型正方扁平封装（PT）——主体10x10x1.0 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		44	
Lead Pitch	e		0.80 BSC	
Overall Height	A	-	-	1.20
Standoff	A1	0.05	-	0.15
Molded Package Thickness	A2	0.95	1.00	1.05
Overall Width	E		12.00 BSC	
Molded Package Width	E1		10.00 BSC	
Overall Length	D		12.00 BSC	
Molded Package Length	D1		10.00 BSC	
Lead Width	b	0.30	0.37	0.45
Lead Thickness	c	0.09	-	0.20
Lead Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	θ	0°	3.5°	7°

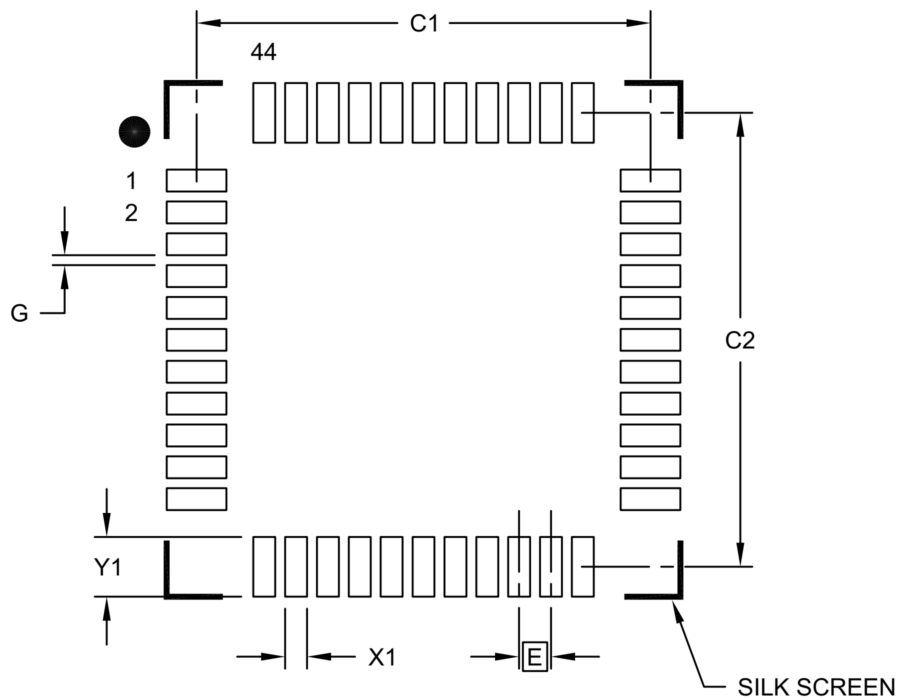
Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Exact shape of each corner is optional.
3. 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 C04-076C Sheet 2 of 2

44引脚塑封薄型正方扁平封装（PT）——主体10x10x1 mm，2.00 mm引脚投影长度 [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	0.80 BSC		
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X44)	X1			0.55
Contact Pad Length (X44)	Y1			1.50
Distance Between Pads	G	0.25		

Notes:

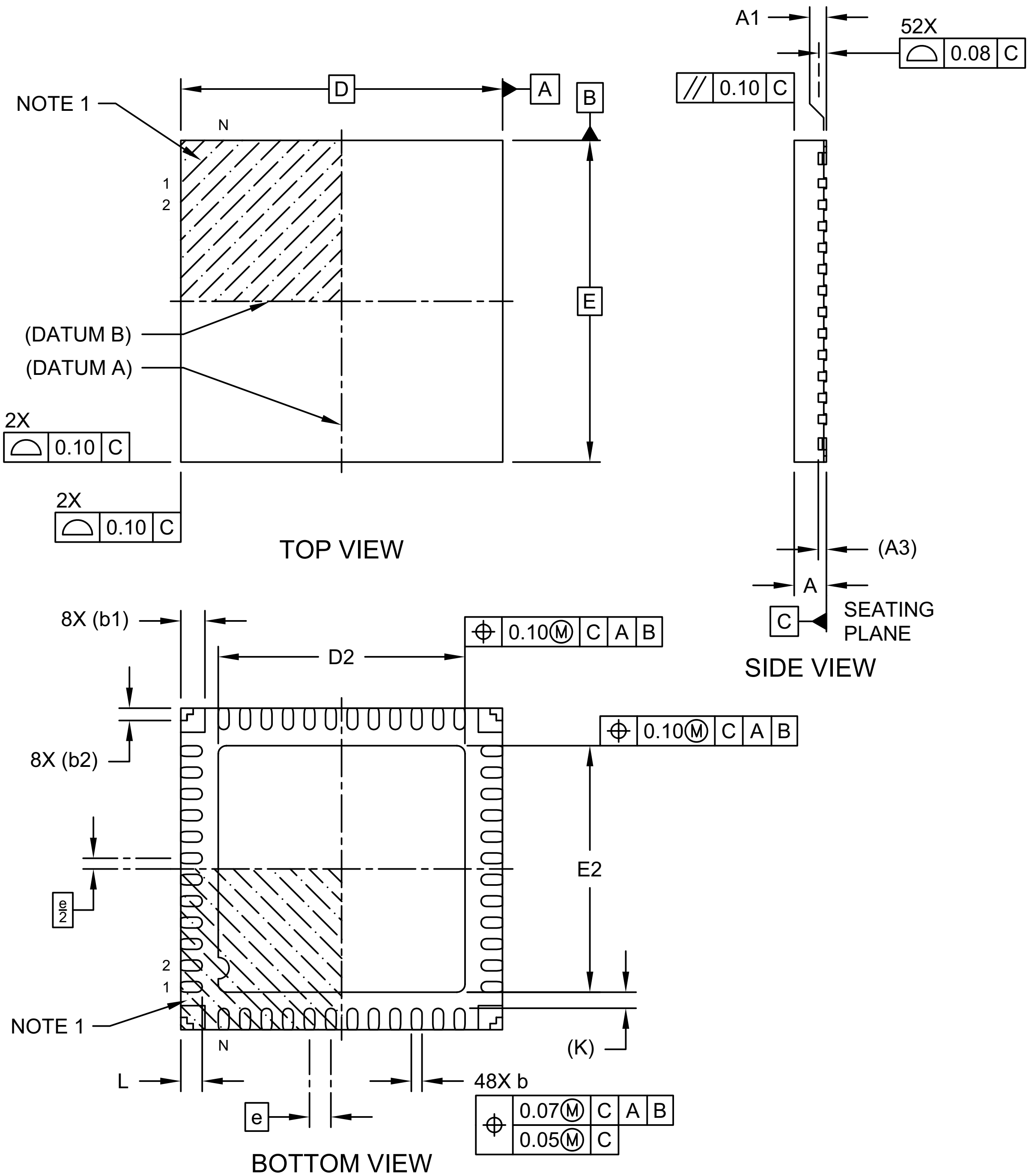
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2076B

48引脚塑封超薄正方扁平无脚封装（M4）——主体6x6 mm [UQFN]，带角锚点和4.6x4.6 mm外露焊盘

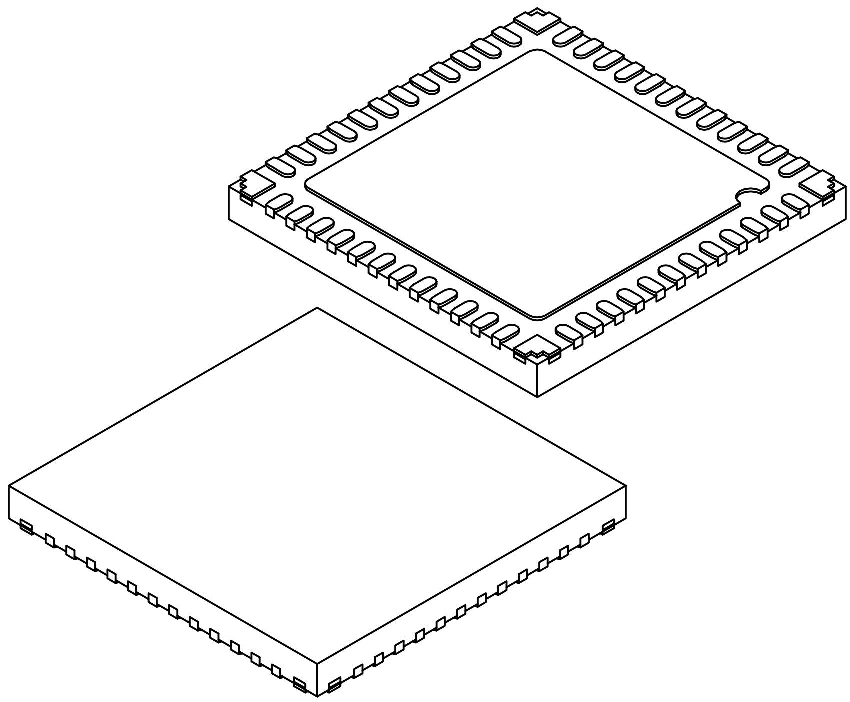
注：	最新封装图请至http://www.microchip.com/packaging查看Microchip封装规范。



Microchip Technology Drawing C04-442A-M4 Sheet 1 of 2

48引脚塑封超薄正方扁平无脚封装（M4）——主体6x6 mm [UQFN]，带角锚点和4.6x4.6 mm外露焊盘

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 Terminals	N		48	
Pitch	e		0.40 BSC	
Overall Height	A	0.50	0.55	0.60
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.15 REF	
Overall Length	D		6.00 BSC	
Exposed Pad Length	D2	4.50	4.60	4.70
Overall Width	E		6.00 BSC	
Exposed Pad Width	E2	4.50	4.60	4.70
Terminal Width	b	0.15	0.20	0.25
Corner Anchor Pad	b1		0.45 REF	
Corner Anchor Pad, Metal-free Zone	b2		0.23 REF	
Terminal Length	L	0.35	0.40	0.45
Terminal-to-Exposed-Pad	K		0.30 REF	

Notes:

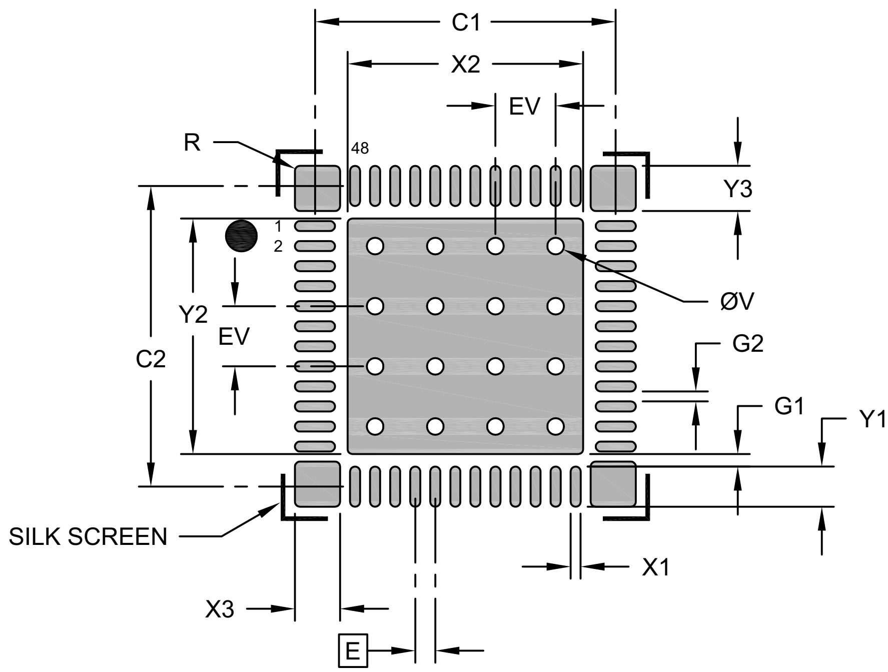
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-442A-M4 Sheet 2 of 2

48引脚塑封超薄正方扁平无脚封装（M4）——主体6x6 mm [UQFN]，带角锚点和4.6x4.6 mm外露焊盘

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	0.40 BSC		
Center Pad Width	X2			4.70
Center Pad Length	Y2			4.70
Contact Pad Spacing	C1		6.00	
Contact Pad Spacing	C2		6.00	
Contact Pad Width (X48)	X1			0.20
Contact Pad Length (X48)	Y1			0.80
Corner Anchor Pad Width (X4)	X3			0.90
Corner Anchor Pad Length (X4)	Y3			0.90
Pad Corner Radius (X 20)	R			0.10
Contact Pad to Center Pad (X48)	G1	0.25		
Contact Pad to Contact Pad	G2	0.20		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

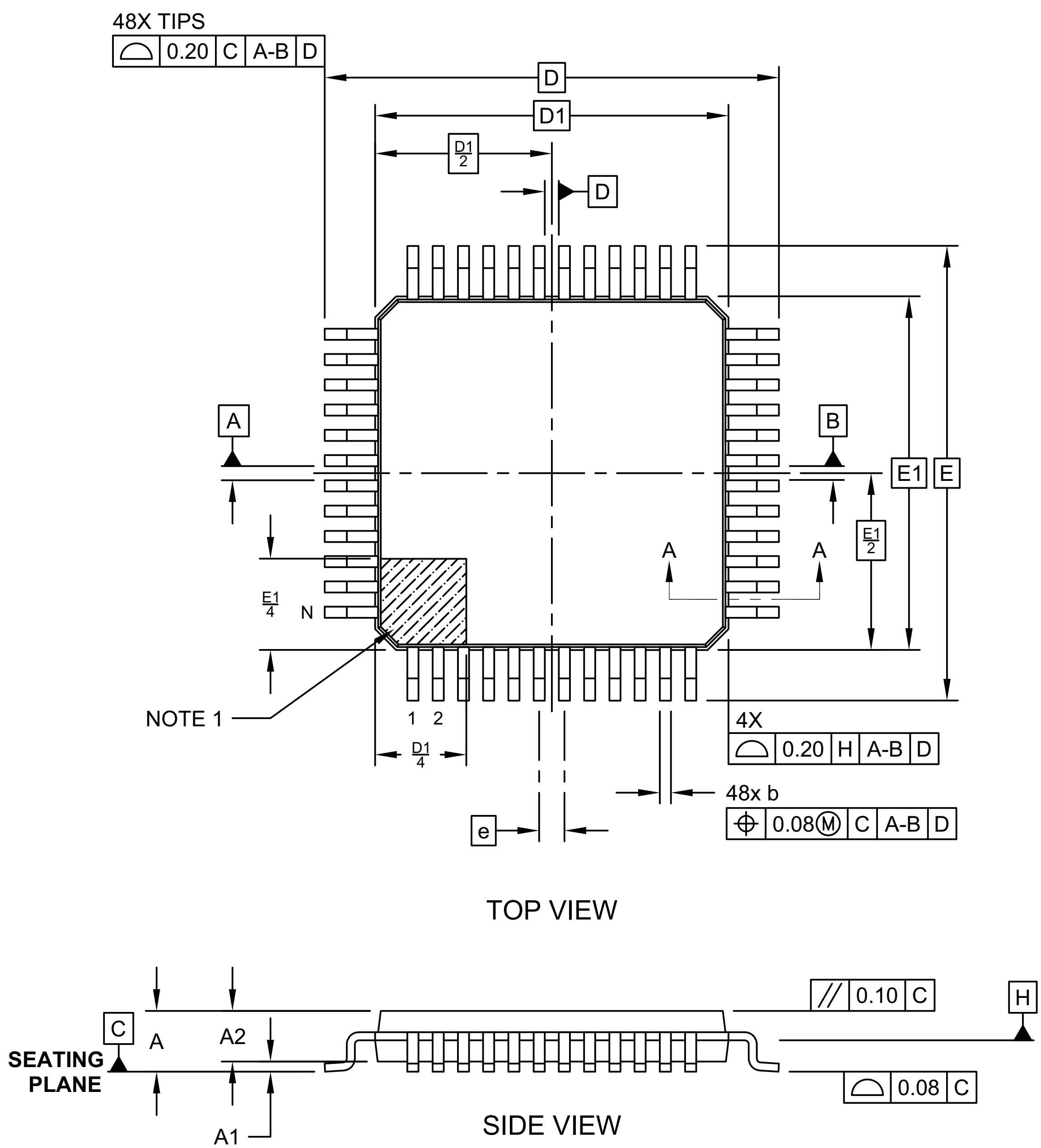
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2442A-M4

48引脚薄型正方扁平封装（PT）——主体7x7x1.0 mm [TQFP]

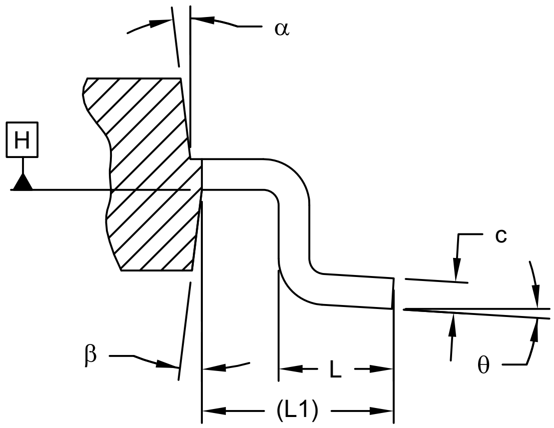
注：	最新封装图请至http://www.microchip.com/packaging查看Microchip封装规范。



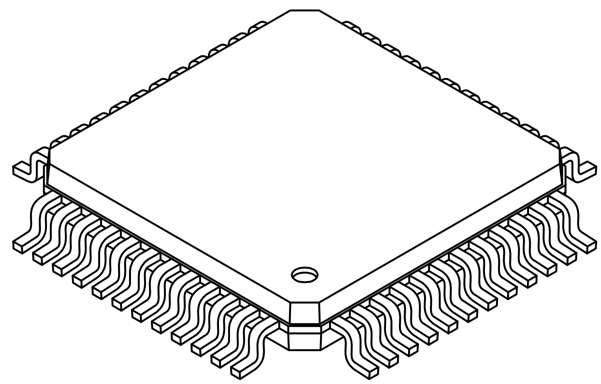
Microchip Technology Drawing C04-300-PT Rev A Sheet 1 of 2

48引脚薄型正方扁平封装（PT）——主体7x7x1.0 mm [TQFP]

注： 最新封装图请至http://www.microchip.com/packaging查看Microchip封装规范。



SECTION A-A



	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Number of Leads	N		48	
Lead Pitch	e		0.50 BSC	
Overall Height	A	-	-	1.20
Standoff	A1	0.05	-	0.15
Molded Package Thickness	A2	0.95	1.00	1.05
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	φ	0°	3.5°	7°
Overall Width	E		9.00 BSC	
Overall Length	D		9.00 BSC	
Molded Package Width	E1		7.00 BSC	
Molded Package Length	D1		7.00 BSC	
Lead Thickness	c	0.09	-	0.16
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:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

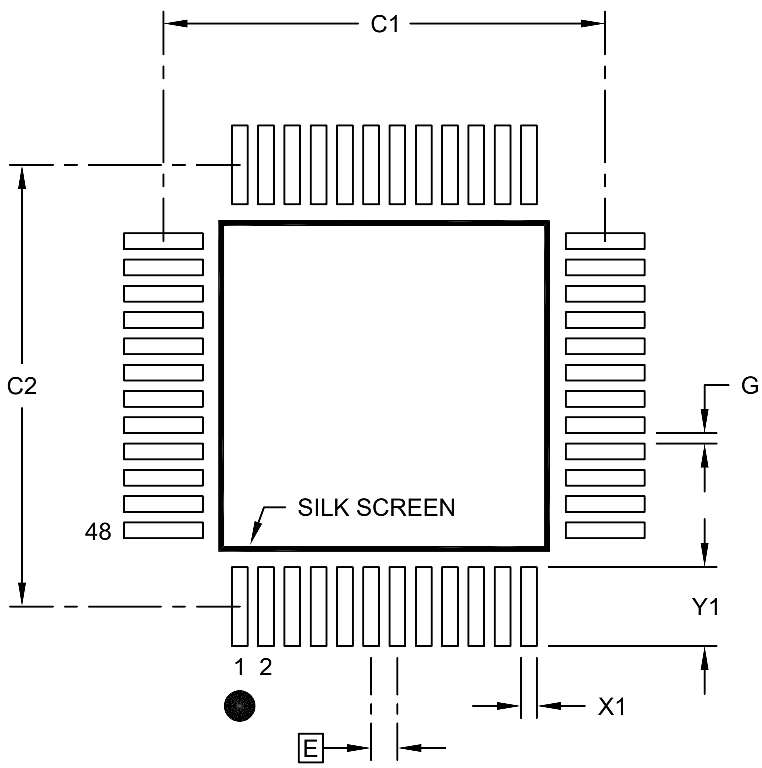
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A-B and D to be determined at center line between leads where leads exit plastic body at datum plane H

Microchip Technology Drawing C04-300-PT Rev A Sheet 2 of 2

48引脚薄型正方扁平封装（PT）——主体7x7x1.0 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		8.40	
Contact Pad Spacing	C2		8.40	
Contact Pad Width (X48)	X1			0.30
Contact Pad Length (X48)	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.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2300-PT Rev A

注：

附录A： 版本历史

版本A（2016年3月）

PIC24FJ256GA705系列器件的初始数据手册。

版本B（2016年10月）

此版本包括以下更新：

- 章节：
  - 删除了**第9.5节 “FRC主动时钟调节”**。
  - 更新了**第32.0节 “电气特性”**中的绝对最大值。
  - 将**第 33.0 节 “封装信息”**中的 48 引脚 QFN（7x7 mm）更改为48引脚UQFN（6x6 mm）。
- 寄存器：
  - 更新了寄存器9-1、寄存器9-3、寄存器9-6、寄存器16-5和寄存器16-6。
- 表：
  - 增加了表11-3、表11-4和表11-5。
  - 更新了第2页上的外设功能表中的GPIO列。
  - 更新了表1、表2、表3、表4、表5、表32-4、表32-5、表32-6、表32-7和表32-25。
- 图：
  - 更新了图9-1。
- 对整篇文档的文本和格式进行了更改。

注：

索引

MICROCHIP网站

Microchip 网站（www.microchip.com）为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的互联网浏览器即可访问。网站提供以下信息：

- **产品支持** —— 数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持** —— 常见问题解答（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。在“支持”（Support）下，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://microchip.com/support 获得网上技术支持。

注：

产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

PIC 24 FJ 256 GA7 05 T - I / PT - XXX

- Microchip 的商标
- 架构
- 闪存系列
- 程序存储器容量（KB）
- 产品组
- 引脚数
- 卷带标志（如果适用）
- 温度范围
- 封装
- 定制编号

架构	24	=	不带DSP的16位改进型哈佛架构
闪存系列	FJ	=	闪存程序存储器
引脚数	02	=	28引脚（QFN、UQFN、SOIC、SSOP和SPDIP）
	04	=	44引脚（TQFP）
	05	=	48引脚（UQFN和TQFP）
温度范围	I	=	-40°C至+85°C（工业级）
封装	ML	=	28引脚（6x6 mm）QFN（塑封正方扁平封装）
	M6	=	28引脚（4x4x0.6 mm）UQFN（超薄正方扁平封装）
	SO	=	28引脚（7.50 mm）SOIC（塑封小外形封装）
	SS	=	28引脚（5.30 mm）SSOP（塑封紧缩小外形封装）
	SP	=	28引脚（300 mil）SPDIP（塑封窄型双列直插式封装）
	PT	=	44引脚（10x10x1 mm）TQFP（薄型正方扁平封装）
	5E	=	48引脚（7x7 mm）UQFN（塑封正方扁平封装）
	PT	=	48引脚（7x7x1 mm）TQFP（薄型正方扁平封装）
定制编号	QTP、SQTP、代码或特殊要求（否则为空白）		
	ES	=	工程样片

示例：

a) PIC24FJ256GA705-I/PT：
PIC24F通用器件，48引脚，工业级温度，
TQFP封装。

b) PIC24FJ256GA702-I/ML：
PIC24F通用器件，28引脚，工业级温度，
QFN封装。

注：

请注意以下有关 Microchip 器件代码保护功能的要点：

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

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

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

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

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-16949:2009 认证。 Microchip 的 PIC® MCU 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。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。

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

商标

Microchip 的名称和徽标组合、 Microchip 徽标、 AnyRate、 AVR、 AVR 徽标、 AVR Freaks、 BeaconThings、 BitCloud、 CryptoMemory、 CryptoRF、 dsPIC、 FlashFlex、 flexPWR、 Heldo、 JukeBlox、 KEELOQ、 KEELOQ 徽标、 Kleer、 LANCheck、 LINK MD、 maXStylus、 maXTouch、 MediaLB、 megaAVR、 MOST、 MOST 徽标、 MPLAB、 OptoLyzer、 PIC、 picoPower、 PICSTART、 PIC32 徽标、 Prochip Designer、 QTouch、 RightTouch、 SAM-BA、 SpyNIC、 SST、 SST 徽标、 SuperFlash、 tinyAVR、 UNI/O 及 XMEGA 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

ClockWorks、 The Embedded Control Solutions Company、 EtherSynch、 Hyper Speed Control、 HyperLight Load、 IntelliMOS、 mTouch、 Precision Edge 和 Quiet-Wire 均为 Microchip Technology Inc. 在美国的注册商标。

Adjacent Key Suppression、 AKS、 Analog-for-the-Digital Age、 Any Capacitor、 AnyIn、 AnyOut、 BodyCom、 chipKIT、 chipKIT 徽标、 CodeGuard、 CryptoAuthentication、 CryptoCompanion、 CryptoController、 dsPICDEM、 dsPICDEM.net、 Dynamic Average Matching、 DAM、 ECAN、 EtherGREEN、 In-Circuit Serial Programming、 ICSP、 Inter-Chip Connectivity、 JitterBlocker、 KleerNet、 KleerNet 徽标、 Mindi、 MiWi、 motorBench、 MPASM、 MPF、 MPLAB Certified 徽标、 MPLIB、 MPLINK、 MultiTRAK、 NetDetach、 Omniscient Code Generation、 PICDEM、 PICDEM.net、 PICkit、 PICtail、 PureSilicon、 QMatrix、 RightTouch 徽标、 REAL ICE、 Ripple Blocker、 SAM-ICE、 Serial Quad I/O、 SMART-I.S.、 SQI、 SuperSwitcher、 SuperSwitcher II、 Total Endurance、 TSHARC、 USBCheck、 VariSense、 ViewSpan、 WiperLock、 Wireless DNA 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

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

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

ISBN：978-1-5224-2417-8

MICROCHIP

10/25/17

全球销售及服务网点

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://www.microchip.com/support
网址：www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

奥斯汀 Austin, TX
Tel: 1-512-257-3370

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

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

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

底特律 Detroit
Novi, MI
Tel: 1-248-848-4000

休斯敦 Houston, TX
Tel: 1-281-894-5983

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453
Tel: 1-317-536-2380

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608
Tel: 1-951-273-7800

罗利 Raleigh, NC
Tel: 1-919-844-7510

纽约 New York, NY
Tel: 1-631-435-6000

圣何塞 San Jose, CA
Tel: 1-408-735-9110
Tel: 1-408-436-4270

加拿大多伦多 Toronto
Tel: 1-905-695-1980
Fax: 1-905-695-2078

亚太地区

中国 - 北京
Tel: 86-10-8569-7000

中国 - 成都
Tel: 86-28-8665-5511

中国 - 重庆
Tel: 86-23-8980-9588

中国 - 东莞
Tel: 86-769-8702-9880

中国 - 广州
Tel: 86-20-8755-8029

中国 - 杭州
Tel: 86-571-8792-8115

中国 - 南京
Tel: 86-25-8473-2460

中国 - 青岛
Tel: 86-532-8502-7355

中国 - 上海
Tel: 86-21-3326-8000

中国 - 沈阳
Tel: 86-24-2334-2829

中国 - 深圳
Tel: 86-755-8864-2200

中国 - 苏州
Tel: 86-186-6233-1526

中国 - 武汉
Tel: 86-27-5980-5300

中国 - 西安
Tel: 86-29-8833-7252

中国 - 厦门
Tel: 86-592-238-8138

中国 - 香港特别行政区
Tel: 852-2943-5100

中国 - 珠海
Tel: 86-756-321-0040

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600

台湾地区 - 新竹
Tel: 886-3-577-8366

亚太地区

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

印度 India - Bangalore
Tel: 91-80-3090-4444

印度 India - New Delhi
Tel: 91-11-4160-8631

印度 India - Pune
Tel: 91-20-4121-0141

日本 Japan - Osaka
Tel: 81-6-6152-7160

日本 Japan - Tokyo
Tel: 81-3-6880-3770

韩国 Korea - Daegu
Tel: 82-53-744-4301

韩国 Korea - Seoul
Tel: 82-2-554-7200

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870

菲律宾 Philippines - Manila
Tel: 63-2-634-9065

新加坡 Singapore
Tel: 65-6334-8870

泰国 Thailand - Bangkok
Tel: 66-2-694-1351

越南 Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

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

芬兰 Finland - Espoo
Tel: 358-9-4520-820

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Garching
Tel: 49-8931-9700

德国 Germany - Haan
Tel: 49-2129-3766400

德国 Germany - Heilbronn
Tel: 49-7131-67-3636

德国 Germany - Karlsruhe
Tel: 49-721-625370

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

德国 Germany - Rosenheim
Tel: 49-8031-354-560

以色列 Israel - Ra'anana
Tel: 972-9-744-7705

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

意大利 Italy - Padova
Tel: 39-049-7625286

荷兰 Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

挪威 Norway - Trondheim
Tel: 47-7289-7561

波兰 Poland - Warsaw
Tel: 48-22-3325737

罗马尼亚 Romania - Bucharest
Tel: 40-21-407-87-50

西班牙 Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

瑞典 Sweden - Gothenberg
Tel: 46-31-704-60-40

瑞典 Sweden - Stockholm
Tel: 46-8-5090-4654

英国 UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820