第 12 章 I/O 端口

目录

本章包括下列主题：

12.1 简介

通用 I/O 引脚可被认为是最简单的外设。它们使 PIC32MX 单片机能够监视和控制其他器件。为了增加器件的灵活性和功能性，一些引脚需要与其他功能复用。这些功能取决于器件上所具有的外设部件。一般来说，当某个外设正在工作时，其相应的引脚就不能被用作通用 I/O 引脚。

下面列出了该模块的部分主要特性：

- 可单独使能 / 禁止输出引脚的漏极开路
- 可单独使能 / 禁止输入引脚的上拉
- 可监视选定输入并在检测到不匹配条件时产生中断
- 可在 CPU SLEEP （休眠）和 IDLE （空闲）模式下继续工作
- 可使用 CLR、 SET 和 INV 寄存器进行快速位操作

图12-1给出了典型I/O端口结构的框图。该框图显示了许多可以在I/O引脚上进行复用的外设功能。

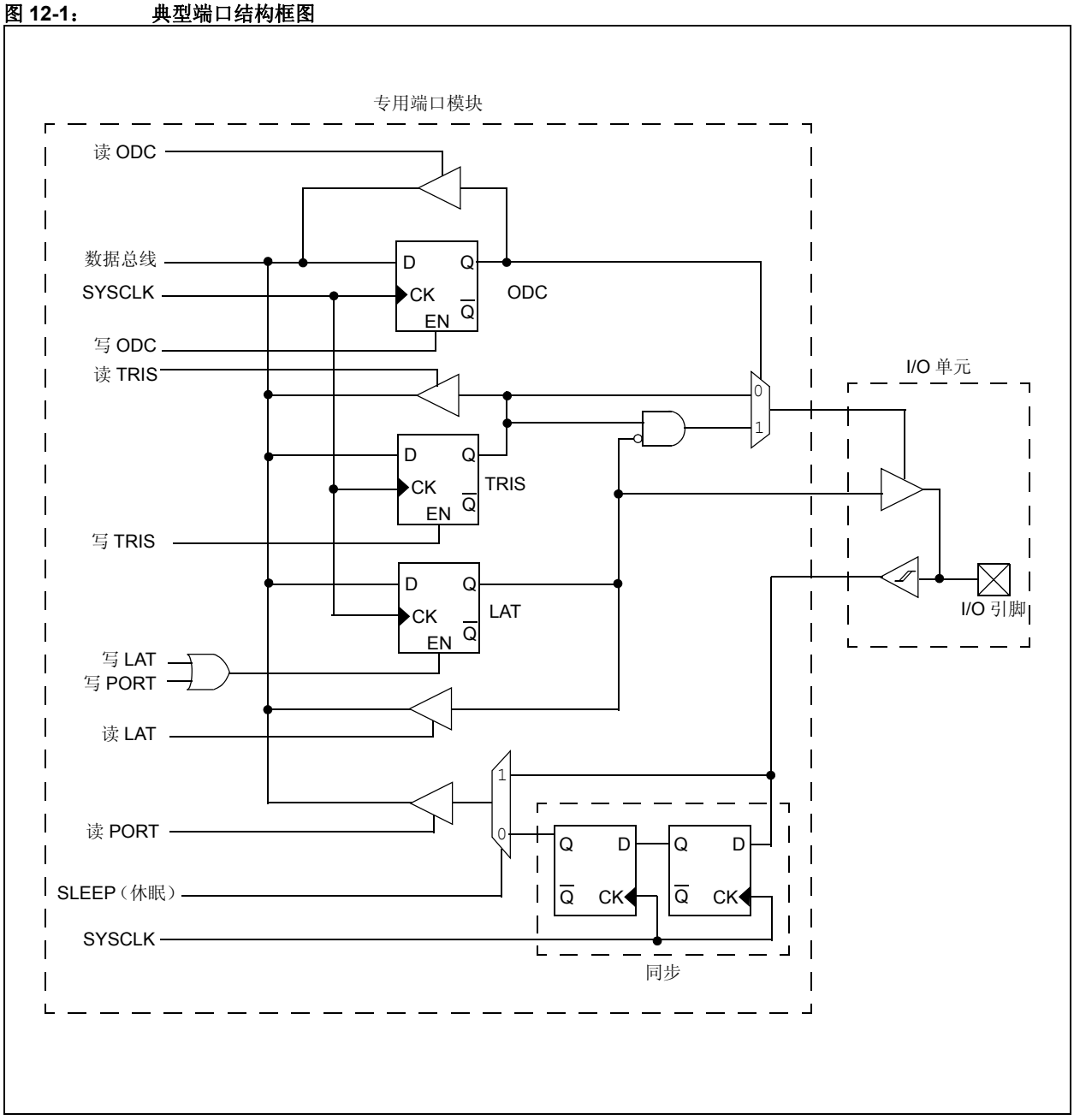
I/O 端口模块包含以下特殊功能寄存器 （Special Function Register， SFR）：

- TRISx：模块 “x” 的数据方向寄存器
- PORTx：模块 “x” 的端口寄存器
- LATx：模块 “x” 的锁存寄存器
- ODCx：模块 “x” 的漏极开路控制寄存器
- CNCON：电平变化中断控制寄存器
- CNEN：输入电平变化通知中断允许寄存器
- CNPUE：输入电平变化通知上拉使能寄存器

I/O 端口模块还具有以下用于中断控制的相关位：

- INT 寄存器 IEC1：中断允许控制寄存器 1 中的 CN 事件中断允许控制位 （CNIE）。
- INT 寄存器 IFS1：中断标志状态寄存器 1 中的 CN 事件中断标志状态位 （CNIF）。
- INT 寄存器 IPC6：中断优先级控制寄存器 6 中的中断优先级控制位 （CNIP<2:0>）。

图 12-1：　　典型端口结构框图



12.2 控制寄存器

在读写任意 I/O 端口之前，应该先为应用正确配置所需的引脚。每个 I/O 端口都有 3 个与端口操作直接关联的寄存器：TRIS、PORT 和 LAT。每个 I/O 端口引脚在这些寄存器中都具有相应的位。根据 PIC32MX 器件型号，最多会有 7 个 I/O 端口可用。在本章中，字母“x”表示任意或所有端口模块。例如，“TRISx”表示 TRISA、TRISB 和 TRISC 等。如果任何位及其关联的数据和控制寄存器对于特定器件是无效的，则会被禁止，并读为零。

注：	端口和可用 I/O 引脚的总数将取决于不同的器件。在一个给定的器件中，并非端口控制寄存器中的所有位都可用。更多详细信息，请参见具体器件数据手册。

12.2.1 TRIS （三态）寄存器

TRIS 寄存器用于配置通过端口 I/O 引脚的数据方向。 TRIS 寄存器位决定 PORT I/O 引脚是输入还是输出。

- TRIS 位设为 1 时，相应的 I/O 端口引脚配置为输入。
- TRIS 位设为 0 时，相应的 I/O 端口引脚配置为输出。
- 读 TRIS 寄存器时，将会读取最后写入 TRIS 寄存器的值。
- 上电复位之后，所有 I/O 端口引脚都定义为输入。

12.2.2 PORT 寄存器

PORT 寄存器用于访问 （读取） I/O 引脚。

- 写 PORT 寄存器时，数据将会写入相应的 LAT 寄存器 （PORT 数据锁存器）。那些配置为输出的 I/O 端口引脚会被更新。
- 写 PORT 寄存器实际上等同于写 LAT 寄存器。
- 读 PORT 寄存器时，将会读取施加到端口 I/O 引脚的同步信号。

12.2.3 LAT 寄存器

LAT 寄存器 （PORT 数据锁存器）用于保存写入端口 I/O 引脚的数据。

- 写 LAT 寄存器时，会将数据锁存到相应的端口 I/O 引脚。那些配置为输出的 I/O 端口引脚会被更新。
- 读 LAT 寄存器时，将会读取 PORT 数据锁存器中保存的数据，而不是从端口 I/O 引脚读取数据。

12.2.4 SET、 CLR 和 INV I/O 端口寄存器

除 TRIS、 PORT 和 LAT 基址寄存器外，每个端口模块还有关联的 SET （置 1）、 CLR （清零）和 INV （取反）寄存器，用于提供原子级位操作并允许更快速的 I/O 引脚操作。正如寄存器名称所示，向 SET、 CLR 或 INV 寄存器写入值会有效地执行其名称所示的操作，但只会修改相应的基址寄存器中指定为 1 的位。不会修改指定为 0 的位。

- 向 TRISASET 寄存器写入 0x0001，只会将基址寄存器 TRISA 中的 bit 0 置 1
- 向 PORTDCLR 寄存器写入 0x0020，只会将基址寄存器 PORTD 中的 bit 5 清零
- 向 LATCINV 寄存器写入 0x9000，只会将基址寄存器 LATC 中的 bit 15 和 bit 12 取反

读取 SET、CLR 和 INV 寄存器会返回未定义的值。要查看对 SET、CLR 或 INV 寄存器执行写操作后的效果，必须读取基址寄存器。

SET、CLR 和 INV 寄存器不限于 TRIS、PORT 和 LAT 寄存器。其他 I/O 端口模块寄存器 ODC、CNEN 和 CNPUE 也具有这些位操作寄存器。

翻转 I/O 引脚的典型方法需要用软件对 PORT 寄存器执行读 - 修改 - 写操作。例如，读 PORTx 寄存器、设置掩码并修改所需的输出位，将结果值回写到 PORTx 寄存器中。该方法容易受读 - 修改 - 写问题影响：读取端口值之后，端口值可能会在修改数据回写之前更改，从而更改先前状态。该方法需要的指令数也更多。

```
PORTA ^= 0x0001;
```

更高效的原子级方法是使用 PORTxINV 寄存器。写 PORTxINV 寄存器实际上会对目标基址寄存器执行读 - 修改 - 写操作，等效于上面介绍的软件操作，但它是在硬件中执行的。要使用该方法翻转 I/O 引脚，可以向 PORTxINV 寄存器的相应位中写入 1。该操作将会读取 PORTx 寄存器，仅将指定为 1 的那些位取反，并将结果值写入 LATx 寄存器，从而在单个原子级指令周期中翻转相应的 I/O 引脚。

```
PORTAINV = 0x0001;
```


在 TRISx 中的 SET、 CLR 和 INV 寄存器行为

- 向 TRISxSET 寄存器中写入值时，将会读取 TRISx 基址寄存器，将指定为 1 的所有位置 1，并将修改值回写到 TRISx 基址寄存器中。
- 向 TRISxCLR 寄存器中写入值时，将会读取 TRISx 基址寄存器，将指定为 1 的所有位清零，并将修改值回写到 TRISx 基址寄存器中。
- 向 TRISxINV 寄存器中写入值时，将会读取 TRISx 基址寄存器，将指定为 1 的所有位取反，并将修改值回写到 TRISx 基址寄存器中。
- 不会修改指定为 0 的任何位。

在 PORTx 中的 SET、 CLR 和 INV 寄存器行为

- 向 PORTxSET 寄存器中写入值时，将会读取 PORTx 基址寄存器，将指定为 1 的所有位置 1，并将修改值回写到 LATx 基址寄存器中。那些配置为输出的 I/O 端口引脚会被更新。
- 向 PORTxCLR 寄存器中写入值时，将会读取 PORTx 基址寄存器，将指定为 1 的所有位清零，并将修改值回写到 LATx 基址寄存器中。那些配置为输出的 I/O 端口引脚会被更新。
- 向 PORTxINV 寄存器中写入值时，将会读取 PORTx 基址寄存器，将指定为 1 的所有位取反，并将修改值回写到 LATx 基址寄存器中。那些配置为输出的 I/O 端口引脚会被更新。
- 不会修改指定为 0 的任何位。

在 LATx 中的 SET、 CLR 和 INV 寄存器行为

- 向 LATxSET 寄存器中写入值时，将会读取 LATx 基址寄存器，将指定为 1 的所有位置 1，并将修改值回写到 LATx 基址寄存器中。那些配置为输出的 I/O 端口引脚会被更新。
- 向 LATxCLR 寄存器中写入值时，将会读取 LATx 基址寄存器，将指定为 1 的所有位清零，并将修改值回写到 LATx 基址寄存器中。那些配置为输出的 I/O 端口引脚会被更新。
- 向 LATxINV 寄存器中写入值时，将会读取 LATx 基址寄存器，将指定为 1 的所有位取反，并将修改值回写到 LATx 基址寄存器中。那些配置为输出的 I/O 端口引脚会被更新。
- 不会修改指定为 0 的任何位。

12.2.5 ODC 寄存器

每个 I/O 引脚都可被单独配置为常规数字输出或漏极开路输出。这是由与每个 I/O 引脚相关的漏极开路控制寄存器 ODCx 控制的。如果 I/O 引脚的 ODC 位为 1，则该引脚作为漏极开路输出引脚。如果 I/O 引脚的 ODC 位为 0，则该引脚被配置为常规数字输出引脚 （ODC 位仅对输出引脚有效）。发生复位后， ODCx 寄存器的所有位的状态均设为 0。

漏极开路特性可通过外部上拉电阻，在所需的任意引脚 （仅用作数字功能）上产生高于 VDD 的输出电压。所允许的最大漏极开路电压与最大 VIH 规范相同。ODC 寄存器在所有 I/O 模式下均有效，使得即使有外设正在控制引脚，也可以产生漏极开路输出。虽然用户可以通过操作对应的 LAT 和 TRIS 位来达到相同的效果，但该过程不允许外设在漏极开路模式下工作 （I^2C™ 引脚的默认操作除外）。因为 I^2C 引脚已经是漏极开路引脚，所以 ODCx 设置不会影响 I^2C 引脚。同样，在 JTAG 扫描单元插入到 ODCx 逻辑和 I/O 之间时， ODCx 设置也不会影响 JTAG 输出特性。

12.2.6 CN 控制寄存器

有几个 I/O 引脚可单独配置为在输入引脚上检测到电平变化时产生中断。有 3 个与 CN （电平变化通知）模块相关的控制寄存器。CNCON 控制寄存器用于使能或禁止 CN 模块。CNEN 寄存器包含 CNENx 控制位，其中 “x”表示 CN 输入引脚的编号。 CNPUE 寄存器包含 CNPUEx 控制位。每个 CN 引脚都连接一个上拉设备，可通过 CNPUEx 控制位使能或禁止该设备。上拉设备充当连接到该引脚的电流源，并且当连接按钮或键盘设备时无需外部电阻。关于 CN 上拉设备的电流规范，请参见具体器件数据手册的 **“电气特性”**章节。

下表简要汇总了所有与 I/O 端口相关的寄存器。该汇总表之后列出了相应的寄存器，并且每个寄存器均附有详细的说明。

表 12-1： I/O 端口 SFR 汇总

名称		Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
TRISx	31:0	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	TRISx<15:8>							
	7:0	TRISx<7:0>							
TRISxCLR	31:0	写入时会将 TRISx 中的选定位清零，读取时获得的值未定义							
TRISxSET	31:0	写入时会将 TRISx 中的选定位置 1，读取时获得的值未定义							
TRISxINV	31:0	写入时会将 TRISx 中的选定位取反，读取时获得的值未定义							
PORTx	31:0	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	PORTx<15:8>							
	7:0	PORTx<7:0>							
PORTxCLR	31:0	写入时会将 LATx 中的选定位清零，读取时获得的值未定义							
PORTxSET	31:0	写入时会将 LATx 中的选定位置 1，读取时获得的值未定义							
PORTxINV	31:0	写入时会将 LATx 中的选定位取反，读取时获得的值未定义							
LATx	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	LATx<15:8>							
	7:0	LATx<7:0>							
LATxCLR	31:0	写入时会将 LATx 中的选定位清零，读取时获得的值未定义							
LATxSET	31:0	写入时会将 LATx 中的选定位置 1，读取时获得的值未定义							
LATxINV	31:0	写入时会将 LATx 中的选定位取反，读取时获得的值未定义							
ODCx	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	ODCx<5:8>							
	7:0	ODCx<7:0>							
ODCxCLR	31:0	写入时会将 ODCx 中的选定位清零，读取时获得的值未定义							
ODCxSET	31:0	写入时会将 ODCx 中的选定位置 1，读取时获得的值未定义							
ODCxINV	31:0	写入时会将 ODCx 中的选定位取反，读取时获得的值未定义							

表 12-1： I/O 端口 SFR 汇总 （续）

名称		Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
CNCON	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	ON	FRZ	SIDL	—	—	—	—	—
	7:0	—	—	—	—	—	—	—	—
CNCONCLR	31:0	写入时会将 CNCON 中的选定位清零，读取时获得的值未定义							
CNCONSET	31:0	写入时会将 CNCON 中的选定位置 1，读取时获得的值未定义							
CNCONINV	31:0	写入时会将 CNCON 中的选定位取反，读取时获得的值未定义							
CNEN	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	CNEN<21:16>					
	15:8	CNEN<15:8>							
	7:0	CNEN<7:0>							
CNENCLR	31:0	写入时会将 CNEN 中的选定位清零，读取时获得的值未定义							
CNENSET	31:0	写入时会将 CNEN 中的选定位置 1，读取时获得的值未定义							
CNENINV	31:0	写入时会将 CNEN 中的选定位取反，读取时获得的值未定义							
CNPUE	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	CNPUE<21:16>					
	15:8	CNPUE<15:8>							
	7:0	CNPUE<7:0>							
CNPUECLR	31:0	写入时会将 CNPUE 中的选定位清零，读取时获得的值未定义							
CNPUESET	31:0	写入时会将 CNPUE 中的选定位置 1，读取时获得的值未定义							
CNPUEINV	31:0	写入时会将 CNPUE 中的选定位取反，读取时获得的值未定义							
IEC1	31:24	—	—	—	—	—	—	USBIE	FCEIE
	23:16	—	—	—	—	DMA3IE	DMA2IE	DMA1IE	DMA0IE
	15:8	RTCCIE	FSCMIE	I2C2MIE	I2C2SIE	I2C2BIE	U2TXIE	U2RXIE	U2EIE
	7:0	SPI2RXIE	SPI2TXIE	SPI2EIE	CMP2IE	CMP1IE	PMPIE	AD1IE	CNIE
IFS1	31:24	—	—	—	—	—	—	USBIF	FCEIF
	23:16	—	—	—	—	DMA3IF	DMA2IF	DMA1IF	DMA0IF
	15:8	RTCCIF	FSCMIF	I2C2MIF	I2C2SIF	I2C2BIF	U2TXIF	U2RXIF	U2EIF
	7:0	SPI2RXIF	SPI2TXIF	SPI2EIF	CMP2IF	CMP1IF	PMPIF	AD1IF	CNIF
IPC6	31:24	—	—	—	AD1IP<2:0>			AD1IS<1:0>	
	23:16	—	—	—	CNIP<2:0>			CNIS<1:0>	
	15:8	—	—	—	I2C1IP<2:0>			I2C1IS<1:0>	
	7:0	—	—	—	U1IP<2:0>			U1IS<1:0>	

寄存器 12-1： TRISx：TRIS 寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

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

图注：			
R = 可读位	W = 可写位	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 31-16 **保留**：写入 0；忽略读操作

bit 15-0 **TRISx<15:0>**：TRIS 寄存器位
1 = 相应端口引脚为 “输入”
0 = 相应端口引脚为 “输出”

寄存器 12-2： TRISxCLR：TRIS 清零寄存器

写入时会将 TRISx 中的选定位清零，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 TRISx 中的选定位清零**

在一个或多个位中写入 1 会将 TRISx 寄存器中的相应位清零，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例：TRISxCLR = 0x00008001 时，会将 TRISx 寄存器中的 bit 15 和 bit 0 清零。

寄存器 12-3： TRISxSET：TRIS 置 1 寄存器

写入时会将 TRISx 中的选定位置 1，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 TRISx 中的选定位置 1**

在一个或多个位中写入 1 会将 TRISx 寄存器中的相应位置 1，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例：TRISxSET = 0x00008001 时，会将 TRISx 寄存器中的 bit 15 和 bit 0 置 1。

寄存器 12-4： TRISxINV：TRIS 取反寄存器

写入时会将 TRISx 中的选定位取反，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 TRIS 中的选定位取反**

在一个或多个位中写入 1 会将 TRISx 寄存器中的相应位取反，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例：TRISxINV = 0x00008001 时，会将 TRISx 寄存器中的 bit 15 和 bit 0 取反。

寄存器 12-5： PORTx：PORT 寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

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

图注：			
R = 可读位	W = 可写位	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 31-16 **保留：**写入 0；忽略读操作

bit 15-0 **PORTx<15:0>：**PORT 寄存器位
读 = 端口引脚上的值
写 = 写入 LATx 寄存器、端口锁存器和 I/O 引脚的值

寄存器 12-6： PORTxCLR：PORT 清零寄存器

写入时会将 LATx 中的选定位清零，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 LATx 中的选定位清零**

在一个或多个位中写入 1 会将 LATx 寄存器中的相应位清零，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例：PORTxCLR = 0x00008001 时，会将 LATx 寄存器中的 bit 15 和 bit 0 清零。

寄存器 12-7： PORTxSET：PORT 置 1 寄存器

写入时会将 LATx 中的选定位置 1，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 LATx 中的选定位置 1**

在一个或多个位中写入 1 会将 LATx 寄存器中的相应位置 1，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例：PORTxSET = 0x00008001 时，会将 LATx 寄存器中的 bit 15 和 bit 0 置 1。

寄存器 12-8： PORTxINV：PORT 取反寄存器

写入时会将 LATx 中的选定位取反，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 LATx 中的选定位取反**

在一个或多个位中写入 1 会将 LATx 寄存器中的相应位取反，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例：PORTxINV = 0x00008001 时，会将 LATx 寄存器中的 bit 15 和 bit 0 取反。

寄存器 12-9： LATx：LAT 寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

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

图注：			
R = 可读位	W = 可写位	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 31-16 **保留：**写入 0；忽略读操作

bit 15-0 **LATx<15:0>：**LAT 寄存器位

读 = 端口锁存器中的值，而不是 I/O 引脚上的值
写 = 写入端口锁存器和 I/O 引脚的值

寄存器 12-10： LATxCLR：LAT 清零寄存器

写入时会将 LATx 中的选定位清零，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 LATx 中的选定位清零**

在一个或多个位中写入 1 会将 LATx 寄存器中的相应位清零，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： `LATxCLR = 0x00008001` 时，会将 LATx 寄存器中的 bit 15 和 bit 0 清零。

寄存器 12-11： LATxSET：LAT 置 1 寄存器

写入时会将 LATx 中的选定位置 1，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 LATx 中的选定位置 1**

在一个或多个位中写入 1 会将 LATx 寄存器中的相应位置 1，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： `LATxSET = 0x00008001` 时，会将 LATx 寄存器中的 bit 15 和 bit 0 置 1。

寄存器 12-12： LATxINV：LAT 取反寄存器

写入时会将 LATx 中的选定位取反，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 LATx 中的选定位取反**

在一个或多个位中写入 1 会将 LATx 寄存器中的相应位取反，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： `LATxINV = 0x00008001` 时，会将 LATx 寄存器中的 bit 15 和 bit 0 取反。

寄存器 12-13： ODCx：漏极开路配置寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

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	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 31-16 **保留：**写入 0；忽略读操作

bit 15-0 **ODCx<15:0>：**ODCx 寄存器位
如果端口引脚配置为输出 （相应的 TRISx 位 = 0）
1 = 使能端口引脚漏极开路输出
0 = 禁止端口引脚漏极开路输出
如果端口引脚配置为输入，则 ODCx 位不起任何作用。

寄存器 12-14： ODCxCLR：漏极开路配置清零寄存器

写入时会将 ODCx 中的选定位清零，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 ODCx 中的选定位清零**

在一个或多个位中写入 1 会将 ODCx 寄存器中的相应位清零，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： `ODCxCLR = 0x00008001` 时，会将 ODCx 寄存器中的 bit 15 和 bit 0 清零。

寄存器 12-15： ODCxSET：漏极开路配置置 1 寄存器

写入时会将 ODCx 中的选定位置 1，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 ODCx 中的选定位置 1**

在一个或多个位中写入 1 会将 ODCx 寄存器中的相应位置 1，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： `ODCxSET = 0x00008001` 时，会将 ODCx 寄存器中的 bit 15 和 bit 0 置 1。

寄存器 12-16： ODCxINV：漏极开路配置取反寄存器

写入时会将 ODCx 中的选定位取反，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 ODCx 中的选定位取反**

在一个或多个位中写入 1 会将 ODCx 寄存器中的相应位取反，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： `ODCxINV = 0x00008001` 时，会将 ODCx 寄存器中的 bit 15 和 bit 0 取反。

寄存器 12-17： CNCON：电平变化中断控制寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	R/W-0	R/W-0	r-x	r-x	r-x	r-x	r-x
ON	FRZ	SIDL	—	—	—	—	—
bit 15							bit 8

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 31-16 **保留：**写入 0；忽略读操作

bit 15 **ON：**电平变化通知模块使能位
1 = 使能 CN 模块
0 = 禁止 CN 模块

注： 使用 1:1 PBCLK 分频比时，在清零模块 ON 位的指令之后，用户的软件不应立即在 SYSCLK 周期中读 / 写外设的 SFR。

bit 14 **FRZ：**调试异常模式冻结位
1 = 在 CPU 处于调试异常模式时停止工作
0 = 在 CPU 处于调试异常模式时继续工作

注： FRZ 仅在调试异常模式下可写，在正常模式下强制为 0。

bit 13 **SIDL：**IDLE （空闲）模式停止位
1 = 当器件进入 IDLE （空闲）模式时，模块停止工作
0 = 在 IDLE （空闲）模式下模块继续工作

bit 12-0 **保留：**写入 0；忽略读操作

寄存器 12-18： CNCONCLR：电平变化中断控制清零寄存器

写入时会将 CNCON 中的选定位清零，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 CNCON 中的选定位清零**

在一个或多个位中写入 1 会将 CNCON 寄存器中的相应位清零，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： CNCONCLR = 0x00008000 时，会将 CNCON 寄存器中的 bit 15 清零。

寄存器 12-19： CNCONSET：电平变化中断控制置 1 寄存器

写入时会将 CNCON 中的选定位置 1，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 CNCON 中的选定位置 1**

在一个或多个位中写入 1 会将 CNCON 寄存器中的相应位置 1，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： CNCONSET = 0x00008000 时，会将 CNCON 寄存器中的 bit 15 置 1。

寄存器 12-20： CNCONINV：电平变化中断控制取反寄存器

写入时会将 CNCON 中的选定位取反，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 CNCON 中的选定位取反**

在一个或多个位中写入 1 会将 CNCON 寄存器中的相应位取反，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： CNCONINV = 0x00008000 时，会将 CNCON 寄存器中的 bit 15 取反。

寄存器 12-21： CNEN：输入电平变化通知中断允许寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CNEN21	CNEN20	CNEN19	CNEN18	CNEN17	CNEN16
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNEN15	CNEN14	CNEN13	CNEN12	CNEN11	CNEN10	CNEN9	CNEN8
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
CNEN7	CNEN6	CNEN5	CNEN4	CNEN3	CNEN2	CNEN1	CNEN0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 31-22 **保留：**写入 0；忽略读操作

bit 21-0 **CNEN<21:0>：**CNEN 寄存器位

如果端口引脚配置为输入（相应的 TRISx 位 = 1）
1 = 使能端口引脚输入电平变化通知
0 = 禁止端口引脚输入电平变化通知
如果端口引脚配置为输出，则 CNENx 位不起任何作用。

寄存器 12-22： CNENCLR：输入电平变化通知中断允许寄存器清零寄存器

写入时会将 CNEN 中的选定位清零，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 CNEN 中的选定位清零**

在一个或多个位中写入 1 会将 CNEN 寄存器中的相应位清零，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： CNENCLR = 0x00008001 时，会将 CNEN 寄存器中的 bit 15 和 bit 0 清零。

寄存器 12-23： CNENSET：输入电平变化通知中断允许寄存器置 1 寄存器

写入时会将 CNEN 中的选定位置 1，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 CNEN 中的选定位置 1**

在一个或多个位中写入 1 会将 CNEN 寄存器中的相应位置 1，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： CNENSET = 0x00008001 时，会将 CNEN 寄存器中的 bit 15 和 bit 0 置 1。

寄存器 12-24： CNENINV：输入电平变化通知中断允许寄存器取反寄存器

写入时会将 CNEN 中的选定位取反，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 CNEN 中的选定位取反**

在一个或多个位中写入 1 会将 CNEN 寄存器中的相应位取反，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： CNENINV = 0x00008001 时，会将 CNEN 寄存器中的 bit 15 和 bit 0 取反。

寄存器 12-25： CNPUE：输入电平变化通知上拉使能寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CNPUE21	CNPUE20	CNPUE19	CNPUE18	CNPUE17	CNPUE16
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNPUE15	CNPUE14	CNPUE13	CNPUE12	CNPUE11	CNPUE10	CNPUE9	CNPUE8
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
CNPUE7	CNPUE6	CNPUE5	CNPUE4	CNPUE3	CNPUE2	CNPUE1	CNPUE0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 31-22 **保留：**写入 0；忽略读操作

bit 21-0 **CNPUE<21:0>：**CNPUE 寄存器位

如果端口引脚配置为输入（相应的 TRISx 位 = 1）

1 = 使能端口引脚上拉

0 = 禁止端口引脚上拉

如果端口引脚配置为输出，则应禁止相应的 CNPUEx 位。

寄存器 12-26： CNPUECLR：电平变化中断上拉使能清零寄存器

写入时会将 CNPUE 中的选定位清零，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 CNPUE 中的选定位清零**

在一个或多个位中写入 1 会将 CNPUE 寄存器中的相应位清零，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： `CNPUECLR = 0x00008001` 时，会将 CNPUE 寄存器中的 bit 15 和 bit 0 清零。

寄存器 12-27： CNPUESET：电平变化中断上拉使能置 1 寄存器

写入时会将 CNPUE 中的选定位置 1，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 CNPUE 中的选定位置 1**

在一个或多个位中写入 1 会将 CNPUE 寄存器中的相应位置 1，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： `CNPUESET = 0x00008001` 时，会将 CNPUE 寄存器中的 bit 15 和 bit 0 置 1。

寄存器 12-28： CNPUEINV：电平变化中断上拉使能取反寄存器

写入时会将 CNPUE 中的选定位取反，读取时获得的值未定义	
bit 31	bit 0

bit 31-0 **将 CNPUE 中的选定位取反**

在一个或多个位中写入 1 会将 CNPUE 寄存器中的相应位取反，但不会影响未实现位或只读位。写入 0 不会影响该寄存器。

示例： `CNPUEINV = 0x00008001` 时，会将 CNPUE 寄存器中的 bit 15 和 bit 0 取反。

寄存器 12-29： IEC1：中断允许控制寄存器 1(1)

r-x	r-x	r-x	r-x	r-x	r-x	R/W-0	R/W-0
—	—	—	—	—	—	USBIE	FCEIE
bit 31							bit 24

r-x	r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	DMA3IE	DMA2IE	DMA1IE	DMA0IE
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RTCCIE	FSCMIE	I2C2MIE	I2C2SIE	I2C2BIE	U2TXIE	U2RXIE	U2EIE
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
SPI2RXIE	SPI2TXIE	SPI2EIE	CMP2IE	CMP1IE	PMPIE	AD1IE	CNIE
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 0 **CNIE：**电平变化通知中断允许位
1 = 允许中断
0 = 禁止中断

注 1： 该中断寄存器中的阴影位名称用于控制其他 PIC32MX 外设，与 I/O 端口电平变化通知无关。

寄存器 12-30： IFS1：中断标志状态控制寄存器 1(1)

r-x	r-x	r-x	r-x	r-x	r-x	R/W-0	R/W-0
—	—	—	—	—	—	USBIF	FCEIF
bit 31							bit 24

r-x	r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	DMA3IF	DMA2IF	DMA1IF	DMA0IF
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RTCCIF	FSCMIF	I2C2MIF	I2C2SIF	I2C2BIF	U2TXIF	U2RXIF	U2EIF
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
SPI2RXIF	SPI2TXIF	SPI2EIF	CMP2IF	CMP1IF	PMPIF	AD1IF	CNIF
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 0 **CNIE**：电平变化通知中断允许位
1 = 允许中断
0 = 禁止中断

注 1：该中断寄存器中的阴影位名称用于控制其他 PIC32MX 外设，与 I/O 端口电平变化通知无关。

寄存器 12-31： IPC6：中断优先级控制寄存器 6(1)

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	AD1IP<2:0>			AD1IS<1:0>	
bit 31							bit 24

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CNIP<2:0>			CNIS<1:0>	
bit 23							bit 16

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	I2C1IP<2:0>			I2C1IS<1:0>	
bit 15							bit 8

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	U1IP<2:0>			U1IS<1:0>	
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 20-18 **CNIP<2:0>**：电平变化通知中断优先级位
111 = 中断优先级为 7
110 = 中断优先级为 6
101 = 中断优先级为 5
100 = 中断优先级为 4
011 = 中断优先级为 3
010 = 中断优先级为 2
001 = 中断优先级为 1
000 = 禁止中断

bit 17-16 **CNIS<1:0>**：电平变化通知中断子优先级位
11 = 中断子优先级为 3
10 = 中断子优先级为 2
01 = 中断子优先级为 1
00 = 中断子优先级为 0

注 1：该中断寄存器中的阴影位名称用于控制其他 PIC32MX 外设，与 I/O 端口电平变化通知无关。

12.3 工作模式

12.3.1 数字输入

可通过将相应的 TRIS 寄存器位设为 1 来将引脚配置为数字输入。配置为输入时，引脚是施密特触发器。多个数字引脚还可与模拟输入功能复用，且在上电复位时默认为模拟输入。将AD1PCFG 寄存器中的相应位设为 1 可将该引脚使能为数字引脚。

注：	关于输入缓冲器类型的更多详细信息，请参见具体器件数据手册。为了将与 10 位模数转换器（Analog-to-Digital，A/D）模块复用的引脚用作数字 I/O，AD1PCFG 寄存器中的相应位必须设为 1，即使关闭 A/D 模块时也需如此。

12.3.2 模拟输入

某些引脚可配置为模拟输入，供 A/D 和比较器模块使用。将 AD1PCFG 寄存器中的相应位设为 0 可将该引脚使能为模拟输入引脚，与相应引脚的 TRIS 寄存器设置无关。

12.3.3 数字输出

可通过将相应的 TRIS 寄存器位设为 0 来将引脚配置为数字输出。配置为数字输出时，这些引脚是 CMOS 驱动器，或可通过将 ODC 寄存器中的相应位置 1 来配置为漏极开路输出。

12.3.4 模拟输出

某些引脚可配置为模拟输出，例如供比较器模块使用的 CVREF 输出电压。如果比较器模块配置为提供输出，就会在该引脚上输出模拟电压，与相应引脚的 TRIS 寄存器设置无关。

注：	关于 A/D 和比较器模块使用的更多详细信息，请参见具体器件数据手册。

12.3.5 漏极开路配置

除 PORT、 LAT 和 TRIS 寄存器用于数据控制外，每个被配置为数字输出的端口引脚也可在有效驱动输出和漏极开路输出之间进行选择。这是由与每个端口相关的漏极开路控制寄存器 ODCx 控制的。从上电复位开始，当某个 I/O 引脚配置为数字输出时，其输出默认为有效驱动输出。将 ODCx 寄存器中的某个位设为 1 时，相应引脚将配置为漏极开路输出。

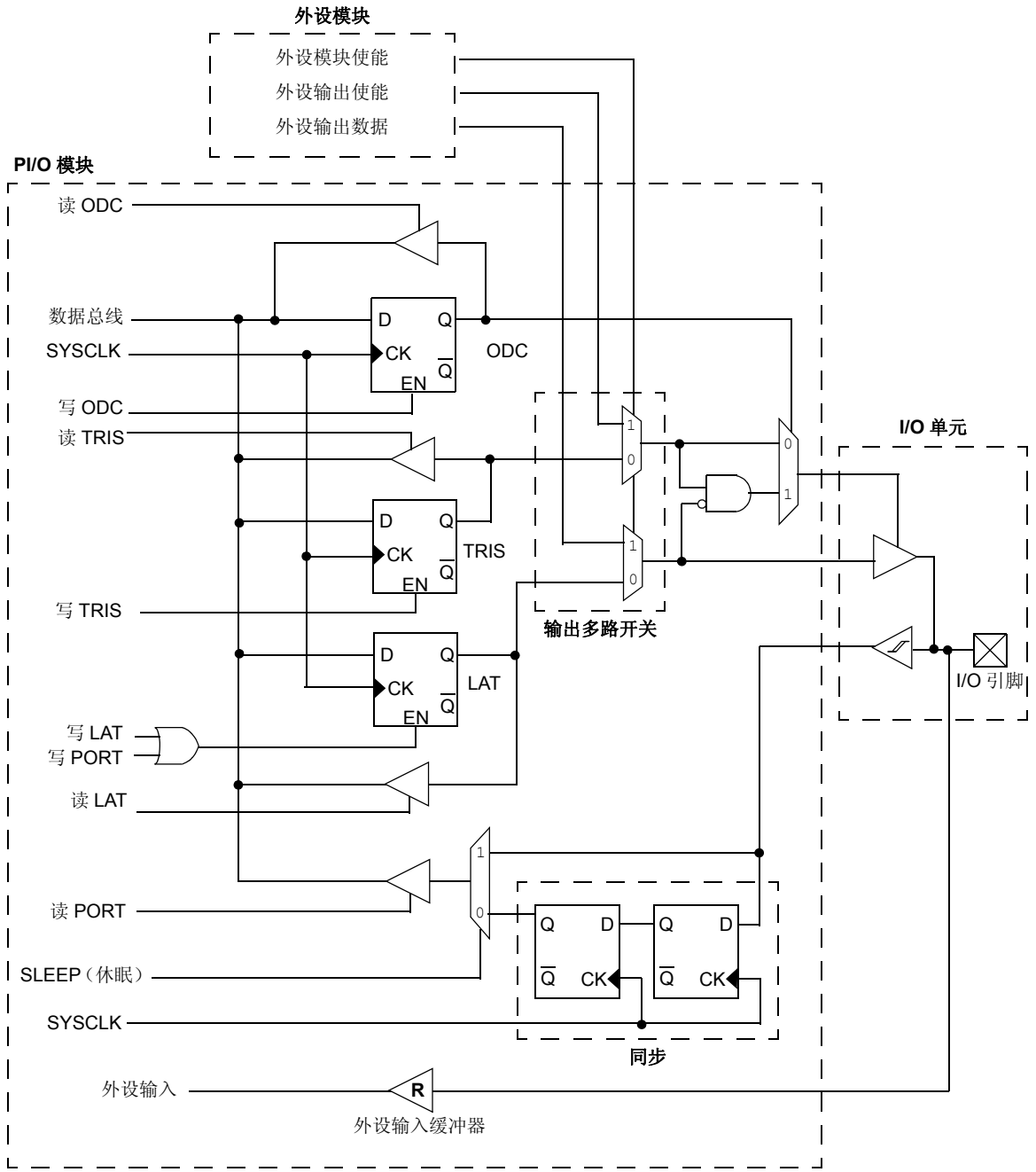
漏极开路特性允许通过使用外部上拉电阻，在所需的任意仅用作数字功能的引脚上产生高于 VDD（如 5V）的输出电压。所允许的最大漏极开路电压与最大 VIH 规范相同。

12.3.6 外设复用

许多引脚还支持一个或多个外设模块。当配置为外设操作时，引脚可能不能用作通用输入或输出。在许多情况下，虽然一些外设会改写 TRIS 配置，引脚仍必须配置为输入或输出。图 12-2 所示为端口如何与其他外设共用，以及端口所连接的相关 I/O 引脚。对于有些 PIC32MX 器件，每个 I/O 引脚可能要复用多个外设功能。外设功能的优先级取决于具体产品数据手册的引脚图中引脚说明的顺序。

请注意，引脚输出可以通过TRISx寄存器位进行控制，或者在某些情况下，通过外设本身进行控制。

图 12-2：　典型共用端口结构框图



注：　该框图是共用端口 / 外设结构的大致表示，仅用作图示。任意特定端口 / 外设组合的实际结构可能会与此处显示的不同。

R = 外设输入缓冲器类型可能会有不同。关于外设的详细信息，请参见 PIC32MX 数据手册。

12.3.6.1 复用数字输入外设

以下情况是复用数字输入外设的特性：

- 外设不控制 TRISx 寄存器。
  一些外设要求引脚配置为输入，配置方法是将相应的 TRISx 位设为 1。
- 外设输入路径与 I/O 输入路径无关，并使用依赖于外设的输入缓冲器。
- PORTx 寄存器数据输入路径不受影响，并且能够读取引脚值。

12.3.6.2 复用数字输出外设

以下情况是复用数字输出外设的特性：

- 外设控制输出数据。
  一些外设要求引脚配置为输出，配置方法是将相应的 TRISx 位设为 0。
- 如果外设引脚具有自动三态功能 （例如 PWM 输出），则外设可以将引脚置为三态。
- 引脚输出驱动器类型会受外设影响 （例如驱动能力和转换率等）。
- PORTx 寄存器输出数据不起任何作用。

12.3.6.3 复用数字双向外设

以下情况是复用数字双向外设的特性：

- 外设可以自动将引脚配置为输出，但不能配置为输入。
  一些外设要求引脚配置为输入，配置方法是将相应的 TRISx 位设为 1。
- 外设控制输出数据。
- 引脚输出驱动器类型会受外设影响 （例如驱动能力和转换率等）。
- PORTx 寄存器数据路径不受影响，并且能够读取引脚值。
- PORTx 寄存器输出数据不起任何作用。

12.3.6.4 复用模拟输入外设

以下情况是复用模拟输入外设的特性：

所有数字端口输入缓冲器都被禁止， PORTx 寄存器读为 0，以防止瞬态开路电流。

12.3.6.5 复用模拟输出外设

以下情况是复用模拟输出外设的特性：

- 所有数字端口输入缓冲器都被禁止， PORTx 寄存器读为 0，以防止瞬态开路电流。
- 不论相关的 TRISx 设置如何，模拟输出均由引脚驱动。

注：	为了将与 A/D 模块复用的引脚用作数字 I/O，AD1PCFG 寄存器中的相应位必须设为 1，即使关闭 A/D 模块时也需如此。

12.3.6.6 软件输入引脚控制

分配给 I/O 引脚的一些功能可能是那些不控制引脚输出驱动器的输入功能。这类外设的一个示例就是输入捕捉模块。如果使用相应的 TRIS 控制位将与输入捕捉相关的 I/O 引脚配置为输出引脚，则用户可以通过其相应的 LAT 寄存器手动影响输入捕捉引脚的状态。这种做法在有些情况下很有用，尤其适用于在没有外部信号连接到输入引脚时进行测试。

如图 12-2 所示，外设多路开关的结构将决定外设输入引脚是否可以通过 PORT 寄存器由软件进行控制。当使能外设功能时，图中所示的概念上的外设会断开 I/O 引脚上的端口数据连接。

一般来说，以下外设允许通过 LAT 寄存器手动控制其输入引脚：

- 外部中断引脚
- 定时器时钟输入引脚
- 输入捕捉引脚
- PWM 故障引脚

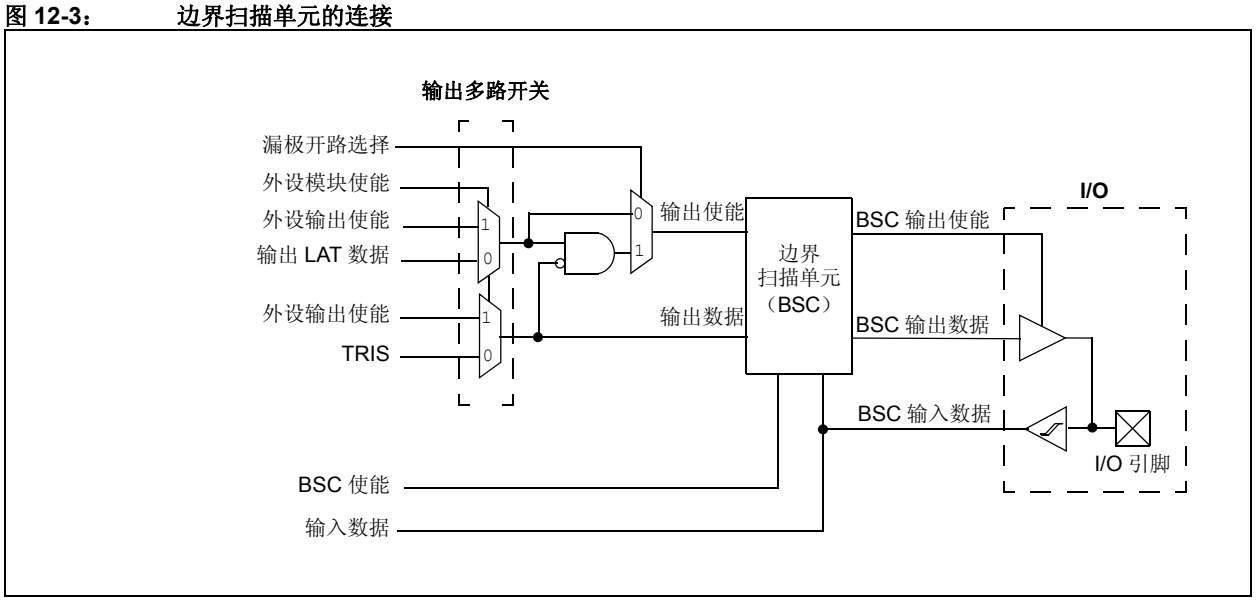
大多数串行通信外设在使能时将完全控制 I/O 引脚，因此不能通过相应的 PORT 寄存器影响与该外设相关的输入引脚。这些外设包括以下模块：

- SPI
- UART
- I^2C

12.3.7 边界扫描单元的连接

PIC32MX 器件支持 JTAG 边界扫描。边界扫描单元（Boundary Scan Cell，BSC）位于内部 I/O 逻辑电路和 I/O 引脚之间，如图 12-3 所示。大多数 I/O 引脚都具有边界扫描单元，但 JTAG 引脚没有。对于正常的 I/O 操作， BSC 被禁止，从而被旁路：BSC 的输出使能输入直接连接到 BSC 输出使能，而 BSC 的输出数据输入直接连接到 BSC 输出数据。电源引脚（VDD、VSS 和 VCAP/VDDCORE）和 JTAG 引脚（TCK、 TDI、 TDO 和 TMS）不具有 BSC。

图 12-3： 边界扫描单元的连接



12.3.8 端口说明

关于可用 I/O 端口和外设复用细节的说明，请参见具体器件数据手册。

12.3.9 电平变化通知引脚

电平变化通知（Change Notification， CN）引脚使 PIC32MX 器件能够向处理器发出中断请求，以响应选定输入引脚上的状态变化（相应的 TRISx 位必须为 1）。可以选择（使能）最多 22 个输入引脚产生 CN 中断。可用的 CN 输入引脚总数取决于所选的 PIC32MX 器件。更多详细信息，请参见具体器件数据手册。

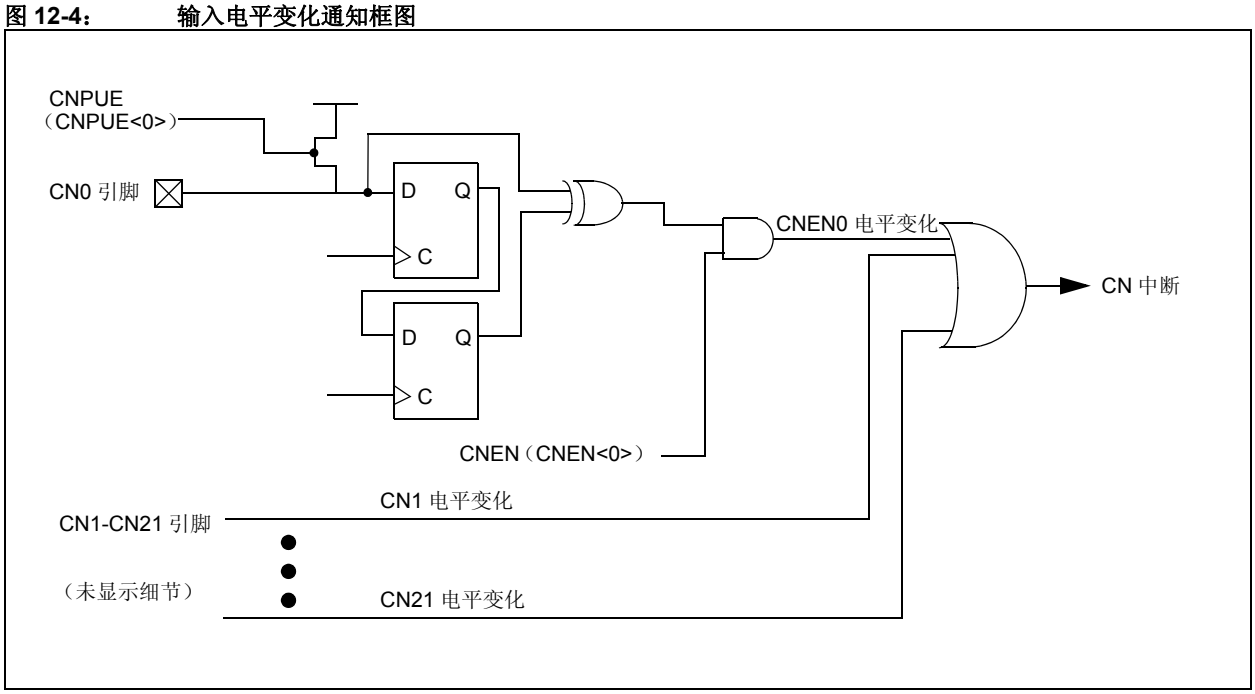
将使能引脚的值与指定 PORT 寄存器在上一次读操作期间采样的值进行比较。如果引脚值不同于上一次读取的值，则会产生不匹配条件。不匹配条件会在任意已使能的输入引脚上发生。对这些不匹配条件进行逻辑或运算，从而提供单个电平变化中断信号。对于使能的引脚，将在每个内部系统时钟周期（SYSCLK）进行采样。

每个 CN 引脚都有一个与之相连的上拉电路。上拉电路充当连接到该引脚的电流源，当连接了按钮或键盘设备时，不再需要使用外部电阻。可使用包含每个 CN 引脚控制位的 CNPUE 寄存器分别使能各个上拉电路。将 CNPUE 寄存器的任一位置 1 均可使能相应引脚的上拉功能。

注：	只要端口引脚被配置为数字输出， CN 引脚的上拉电路应始终被禁止。

图 12-4 给出了 CN 硬件的基本功能。

图 12-4： 输入电平变化通知框图



12.3.9.1 CN 配置和操作

CN 引脚配置如下：

1. 禁止 CPU 中断。
2. 将所需的 CN I/O 引脚设置为输入，方法是将相应的 TRISx 寄存器位设为 1。
   注：如果 I/O 引脚与某个模拟外设共用，则可能需要将相应的 AD1PCFG 寄存器位设为 1，以确保 I/O 引脚为数字输入。
3. 使能 CN 模块， ON （CNCON<15>） = 1。
4. 使能各个 CN 输入引脚，使能可选上拉。
5. 读取相应的 PORT 寄存器，以清除 CN 输入引脚上的不匹配条件。
6. 配置 CN 中断优先级 CNIP<2:0> 和子优先级 CNIS<1:0>。
7. 清零 CN 中断标志， CNIF = 0。
8. 允许 CN 中断， CNIE = 1。
9. 允许 CPU 中断。

当 CN 中断发生时，用户应读取与该 CN 引脚相关的 PORT 寄存器。这样做将清除不匹配条件，并设置 CN 逻辑以检测下一次引脚电平变化。可以将当前的端口值与上一次 CN 中断时或初始化期间得到的端口读取值比较，来确定发生了电平变化的引脚。

CN 引脚具有最小输入脉冲宽度规范。如需了解更多信息，请参见具体器件数据手册的**“电气特性”**章节。

12.4 中断

12.4.1 中断配置

CN 模块具有专用的中断标志位 CNIF 和相应的中断允许 / 屏蔽位 CNIE。这些位用于决定中断源和使能 / 禁止各个中断源。

CNIE 位用于定义在相应的 CNIF 置 1 时，中断控制器的行为。当 CNIE 位清零时，中断控制器模块不会为事件产生 CPU 中断。如果 CNIE 位置 1，则中断控制器模块会在相应的 CNIF 位置 1 时向 CPU 产生中断 （受以下段落中概述的优先级和子优先级制约）。

处理特定中断的用户软件程序需要负责在服务程序完成之前清零相应的中断标志位。

CN 模块的优先级可以使用 CNIP<2:0> 位设置。该优先级定义了中断源将分配到的优先级组。优先级组值的范围为 7 （最高优先级）到 0 （不产生中断）。较高优先级组中的中断会抢占正在处理、但优先级较低的中断。

子优先级位用于设置中断源在优先级组中的优先级。子优先级 CNIS<1:0> 值的范围为 3（最高优先级）到 0 （最低优先级）。处于相同优先级组，但具有更高子优先级值的中断会抢占子优先级较低、但正在进行的中断。

优先级组和子优先级位让多个中断源可以共用相同的优先级和子优先级。如果在该配置下同时发生若干个中断，则中断源在优先级 / 子优先级组对中的自然顺序将决定所产生的中断。自然优先级基于中断源的向量编号。向量编号越小，中断的自然优先级就越高。在当前中断的中断标志清零之后，所有不按照自然顺序执行的中断会基于优先级、子优先级和自然顺序产生相应的中断。

产生允许的中断之后，CPU 会跳转到为该中断分配的向量处。该中断的向量编号与自然顺序编号相同。然后，CPU 会在向量地址处开始执行代码。该向量地址处的用户代码应执行特定于应用程序的操作、清零 CNIF 中断标志，然后退出。关于向量地址表的详细信息和关于中断的更多信息，请参见**第 8 章 “中断”**（DS61108）。

表 12-2： 电平变化通知中断向量 （EBASE = 0x8000:0000）

中断	向量 / 自然顺序	IRQ 编号	向量地址 IntCtl.VS = 0x01	向量地址 IntCtl.VS = 0x02	向量地址 IntCtl.VS = 0x04	向量地址 IntCtl.VS = 0x08	向量地址 IntCtl.VS = 0x10
CN	23	23	8000 04E0	8000 07C0	8000 0D80	8000 1900	8000 3000

表 12-3： 优先级和子优先级分配示例

中断	优先级组	子优先级	向量 / 自然顺序
CN	7	3	23

例 12-1： 电平变化通知配置示例

```
/*
    The following code example illustrates a Change Notice interrupt configuration for pins
    CN1(PORTC.RC13), CN4(PORTB.RB2) and CN18(PORTF.RF5).
*/

    /* NOTE:disable vector interrupts prior to configuration */

    CNCON = 0x8000;        // Enable Change Notice module
    CNEN =  0x00040012;    // Enable individual CN pins CN1, CN4 and CN18
    CNPUE = 0x00040012;    // Enable weak pull ups for pins CN1, CN4 and CN18

    /* read port(s) to clear mismatch on change notice pins */
    PORTB;
    PORTC;
    PORTF;

    IPC6SET = 0x00140000;  // Set priority level=5
    IPC6SET = 0x00030000;  // Set Subpriority level=3
                           // Could have also done this in single
                           // operation by assigning IPC6SET = 0x00170000

    IFS1CLR  = 0x0001;     // Clear the interrupt flag status bit
    IEC1SET  = 0x0001;     // Enable Change Notice interrupts

    /* re-enable vector interrupts after configuration */
```

例 12-2： 电平变化通知 ISR 代码示例

```
/*
    The following code example demonstrates a simple interrupt service routine for CN
    interrupts.The user's code at this vector can perform any application specific
    operations.The user's code must read the CN corresponding PORT registers to clear the
    mismatch conditions before clearing the CN interrupt status flag.Finally, the CN
    interrupt status flag must be cleared before exiting.
*/
void __ISR(_CHANGE_NOTICE_VECTOR, ipl5 ChangeNoticeHandler(void)
{
    ... perform application specific operations in response to the interrupt

    readB = PORTB          // Read PORTB to clear CN4 mismatch condition
    readC = PORTC          // Read PORTC to clear CN1 mismatch condition
    readF = PORTF          // Read PORTF to clear CN18 mismatch condition
    ...
    IFS1CLR  = 0x0001;     // Be sure to clear the CN interrupt status
                           // flag before exiting the service routine.
{
```

注： CN ISR 代码示例显示的是 MPLAB® C32 C 编译器的特定语法。关于对 ISR 的支持，请参见编译器手册。

12.5 节能和调试模式下的操作

注：	在本手册中，对于特定模块中使用的功耗模式和器件使用的功耗模式进行了区分；例如，比较器的 Sleep （休眠）模式和 CPU 的 SLEEP （休眠）模式。为了指示所期望功耗模式的类型，模块功耗模式使用大写字母加小写字母 （Sleep， Idle，Debug）（休眠、空闲和调试）来表示，器件功耗模式使用全大写字母 （SLEEP，IDLE， DEBUG）（休眠、空闲和调试）来表示。

12.5.1 SLEEP （休眠）模式下的 I/O 端口操作

在器件进入 SLEEP（休眠）模式时，系统时钟会被禁止；但是，CN 模块会继续工作。如果使能的 CN 引脚之一改变了状态，则状态位 CNIF （IFS1<0>）将被置 1。如果 CNIE 位 （IEC1<0>）置 1，并且它的优先级大于当前 CPU 优先级，则器件会从 SLEEP （休眠）或 IDLE （空闲）模式唤醒，并执行 CN 中断服务程序。

如果为 CN 中断分配的优先级小于等于当前 CPU 优先级，则不会唤醒 CPU，器件将进入 IDLE （空闲）模式。

12.5.2 IDLE （空闲）模式下的 I/O 端口操作

当器件进入 IDLE（空闲）模式时，系统时钟源继续保持工作。SIDL 位（CNCON<13>）用于选择在 IDLE （空闲）模式下模块是停止还是继续工作。

- 如果 SIDL = 1，则在 IDLE（空闲）模式下，模块会继续对输入 CN I/O 引脚进行采样，但同步会被禁止。
- 如果 SIDL = 0，则在 IDLE（空闲）模式下，模块会继续进行同步和对输入 CN I/O 引脚进行采样。

12.5.3 DEBUG （调试）模式下的 I/O 端口操作

FRZ 位 （CNCON<14>）决定 CPU 在 DEBUG （调试）模式下执行调试异常代码 （即，应用程序暂停）时， CN 模块是继续运行还是停止。

- 如果 FRZ = 0，则在 DEBUG （调试）模式下，即使应用程序暂停，模块也会继续工作。
- 如果 FRZ = 1，且应用程序在 DEBUG （调试）模式下暂停，则模块将停止工作，并且不更改 CN 模块的状态。在 CPU 继续开始执行代码之后，模块将继续工作。

注：	只有 CPU 在调试异常模式下执行时，FRZ 位才可读写。在所有其他模式下，FRZ 位读为 0。如果 FRZ 位在 DEBUG （调试）模式期间发生改变，则只有退出当前调试异常模式并重新进入该模式之后，新值才会生效。在调试异常模式期间，在进入 Debug （调试）模式时 FRZ 位会读取外设状态。

12.6 各种复位的影响

12.6.1 器件复位

在发生器件复位时，所有 TRIS、LAT、PORT、ODC、CNEN、CNPUE 和 CNCON 寄存器会被强制设为它们的复位状态。

12.6.2 上电复位

在发生上电复位时，所有 TRIS、LAT、PORT、ODC、CNEN、CNPUE 和 CNCON 寄存器会被强制设为它们的复位状态。

12.6.3 看门狗复位

在发生看门狗复位时，所有 TRIS、LAT、PORT、ODC、CNEN、CNPUE 和 CNCON 寄存器不变。

12.7 I/O 端口应用

例 12-3： 代码示例

```
/*
 The following code example illustrates configuring
 RB0, RB1 as analog (default) inputs, RB2 as a digital
 input, RB3 as digital output and RB4 as digital output
 with open-drain enabled using SET, CLR atomic SFR registers.*/

 AD1PCFGCLR = 0x0003;       // RB0, RB1 = analog pins
 TRISBSET = 0x0003;         // RB0, RB1 = inputs

 AD1PCFGSET = 0x000C;       // RB2, RB3 = digital pins
 TRISBSET =  0x0004;        // RB2 = input
 TRISBCLR =  0x0018;        // RB3, RB4 = outputs

 ODCBSET = 0x0010;          // RB4 open-drain enabled

/*
 The following code example illustrates same configuration
 above using Base SFR registers directly.*/

 AD1PCFG = 0x001C;          // RB0, RB1 = analog pins; RB2, RB3, RB4 = digital pins
 TRISB =  0x0007;           // RB0, RB1, RB2 = inputs; RB3, RB4 = outputs

 ODCB = 0x0010;             // RB4 open-drain enabled
```

12.8 I/O 引脚控制

表 12-4 汇总了 I/O 引脚模式设置。

表 12-4： I/O 引脚配置

数字引脚控制的必需设置							
模式或引脚用法	引脚类型	缓冲器类型	TRIS 位	ODC 位	CNEN 位	CNPUE 位 (1)	AD1PCFG 位
输入	IN	ST	1	—	—	—	1
CN	IN	ST	1	—	1	1	1
输出	OUT	CMOS	0	0	—	—	1
漏极开路	OUT	OPEN	0	1	—	—	1

模拟引脚控制的必需设置							
模式或引脚用法	引脚类型	缓冲器类型	TRIS 位	ODC 位	CNEN 位	CNPUE 位 (1)	AD1PCFG 位
ANx 输入	IN	A	1	—	—	—	0
CV 输出	OUT	A	—	—	—	—	0

JTAG 引脚控制的必需设置 (2)							
模式或引脚用法	引脚类型	缓冲器类型	TRIS 位	ODC 位	CNEN 位	CNPUE 位 (1)	AD1PCFG 位
TCK	IN	ST	—	—	—	—	—
TDI	IN	ST	—	—	—	—	—
TMS	IN	ST	—	—	—	—	—
TDO	OUT	CMOS	—	—	—	—	—

ICSP 引脚控制的必需设置 (3)							
模式或引脚用法	引脚类型	缓冲器类型	TRIS 位	ODC 位	CNEN 位	CNPUE 位 (1)	AD1PCFG 位
PGC	IN	ST	—	—	—	—	—
	OUT	CMOS	—	—	—	—	—
PGD	IN	ST	—	—	—	—	—
	OUT	CMOS	—	—	—	—	—

图注： CMOS = CMOS 兼容输入或输出
ST = 带 CMOS 电平的施密特触发器输入
I = 输入
O = 输出

注 1： CN 使能上拉位是可选的。

2： 当使能 JTAG，并选择相应的 DEBUG （调试）模式时，将会自动设置 JTAG 模块的引脚控制。无需任何用户配置。

3： ICSP™ 模块的引脚控制在进入 ICSP 模式时自动设置。无需任何用户配置。

12.9 设计技巧

问 1： *我应如何配置未用的 I/O 引脚？*

答 1： 未用 I/O 引脚可以用软件设置为输出（相应的 TRIS 位 = 0）并驱动为低电平（相应的 LAT 位 = 0）。

问 2： ***PIC32MX 的 I/O 引脚是否可以与 5V 设备连接？***

答 2： 可以，但存在一些限制。 PIC32MX 的 I/O 引脚配置为输入时，可承受 5V 电压，这意味着引脚可承受最高为 5V 的输入。 I/O 引脚配置为输出时，只能驱动与 PIC32MX V_{DD} 引脚所提供电压相同的电压，该电压限制为 3.6V。根据 5V 设备的输入引脚设计，要正确地读为逻辑“高电平”信号，该电压可能不够高。关于连接不同逻辑电平系列的详细讨论，请参见“*Microchip 3V Tips 'n Tricks*”（DS41285）指南。

12.10 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 PIC32MX 器件系列而编写的，但其概念是相近的，通过适当修改并受到一定限制即可使用。当前与 I/O 端口相关的应用笔记有：

标题	应用笔记编号
Implementing Wake-up on Key Stroke	AN552

注： 如需获取更多 PIC32MX 系列器件的应用笔记和代码示例，请访问 Microchip 网站（www.microchip.com）。

12.11 版本历史

版本 A （2007 年 8 月）

这是本文档的初始版本。

版本 B （2007 年 10 月）

更新了文档 （删除了 “机密” 状态）。

版本 C （2008 年 4 月）

将状态修改为 “初稿”；将 U-0 修改为 r-x ；修改了寄存器 12-13 ；修改了图 12-1 和 12-2。

版本 D （2008 年 5 月）

修改了寄存器 12-17，为 FRZ 增加了注释；为寄存器 12-19、 12-30 和 12-31 增加了注释；修改了例12-1和12-2；将保留位从“保持为”更改为“写入”；为ON位（CNCON寄存器）增加了注释。

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。

ISBN: 978-1-60932-097-3

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

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2002 认证。公司在 PIC® MCU 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品方面的质量体系流程均符合 ISO/TS-16949:2002。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

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://support.microchip.com
网址：www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

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

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

克里夫兰 Cleveland
Independence, OH
Tel: 216-447-0464

Fax: 216-447-0643

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

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

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

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

圣克拉拉 Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

加拿大多伦多 Toronto
Mississauga, Ontario,
Canada
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200

Fax: 852-2401-3431

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

中国 – 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 – 重庆
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

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

中国 – 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

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

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

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

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

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

中国 – 西安
Tel: 86-29-8833-7252

Fax: 86-29-8833-7256

中国 – 厦门
Tel: 86-592-238-8138

Fax: 86-592-238-8130

中国 – 珠海
Tel: 86-756-321-0040
Fax: 86-756-321-0049

台湾地区 – 高雄
Tel: 886-7-536-4818

Fax: 886-7-536-4803

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

亚太地区

台湾地区 – 新竹
Tel: 886-3-6578-300
Fax: 886-3-6578-370

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

印度 India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

印度 India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

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

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

韩国 Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

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

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

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

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

欧洲

奥地利 Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 Denmark-Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

荷兰 Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

西班牙 Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

英国 UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

01/05/10