

(Version 1.33: 2011-08-26)

## 目录

一、简介.....	3
二、特征.....	3
三、管脚分配.....	5
封装脚位图.....	5
管脚描述.....	6
PAD 图.....	9
衬底接 GND.....	9
四、存储器.....	10
4.1 程序存储器.....	10
4.2 数据存储器.....	11
4.3 特殊功能寄存器.....	11
4.4 特殊功能寄存器表.....	12
4.5 特殊功能寄存器描述.....	14
4.5.1 与系统相关特殊功能寄存器描述.....	14
4.5.2 与中断及看门狗相关特殊功能寄存器描述.....	18
4.5.3 与 IO 口相关特殊功能寄存器描述.....	22
4.5.4 与定时器 0 相关特殊功能寄存器描述.....	32
4.5.5 与定时器 1 及 PWP 相关特殊功能寄存器描述.....	34
4.5.6 与 PWM 相关特殊功能寄存器描述.....	36
4.5.7 与 SPI 相关特殊功能寄存器描述.....	40
4.5.8 与 I2C 相关特殊功能寄存器描述.....	43
4.5.9 与 ADC 相关特殊功能寄存器描述.....	46
4.5.10 与 VREF 相关特殊功能寄存器描述.....	49
五、功能描述.....	51
5.1 TCC/WDT 预分频器.....	51
5.2 I/O 端口.....	51
5.3 定时器.....	52
5.4 复位和唤醒.....	53
5.5 中断.....	54
5.6 振荡器.....	55
5.7 SPI.....	57
5.7.1 SPI 功能描述.....	59
5.7.2 SPI 模式的时序图.....	61
5.8 模数转换器 (A/D) 模块.....	63
5.9 I2C 模块.....	64
5.9.1 从动模式.....	64
5.9.2 主控模式.....	66
六、绝对最大范围.....	66
七、电气特性.....	66
八、封装尺寸图.....	68
SOP20.....	68
SOP14.....	69

SOP16.....	70
附：版本记录.....	71

# 一、简介

TX416是一个具有低功耗和高速CMOS工艺的8位微处理器。它的操作核心由RISC类体系结构实现。有一个脉冲宽度调制输出和SPI的通讯接口。同时也增强了单片机的其它内部特性，如暂停、唤醒功能、振荡器选择和可编程分频器等，增加了单片机的使用灵活度，而这些特性也同时保证实际应用时只需要最少的外部器件，进而降低了整个产品的成本。此单片机广泛被应用在红外遥控发射器、马达驱动、工业控制、消费性产品和子系统控制器等场合。

采用 16 位精简指令集并拥有高达 256 字节的 RAM, 程序可方便的访问控制器内部的 4096 \* 16 bits ROM 空间及 RAM 空间而不需要执行繁琐的换页功能。采用特殊功能寄存器与 RAM 统一寻址的方式。

TX416是属于一次可编程(One-Time Programmable, OTP)单片机，配合使用我们提供的程序开发工具，可简单有效的更新程序，给设计者提供了快速有效的开发途径。

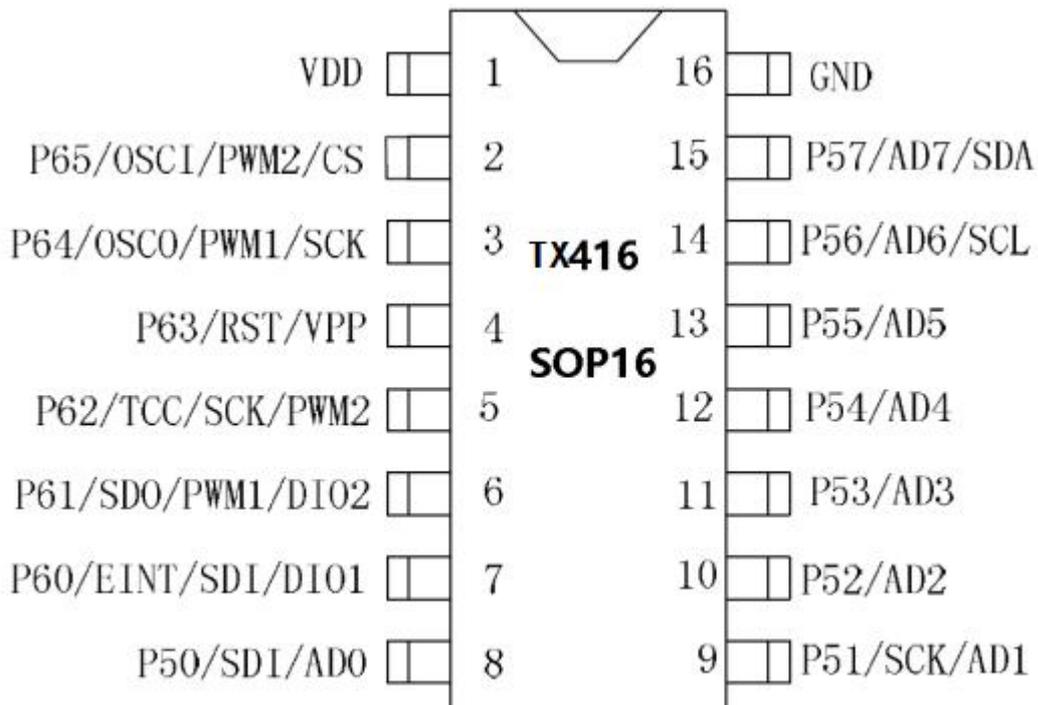
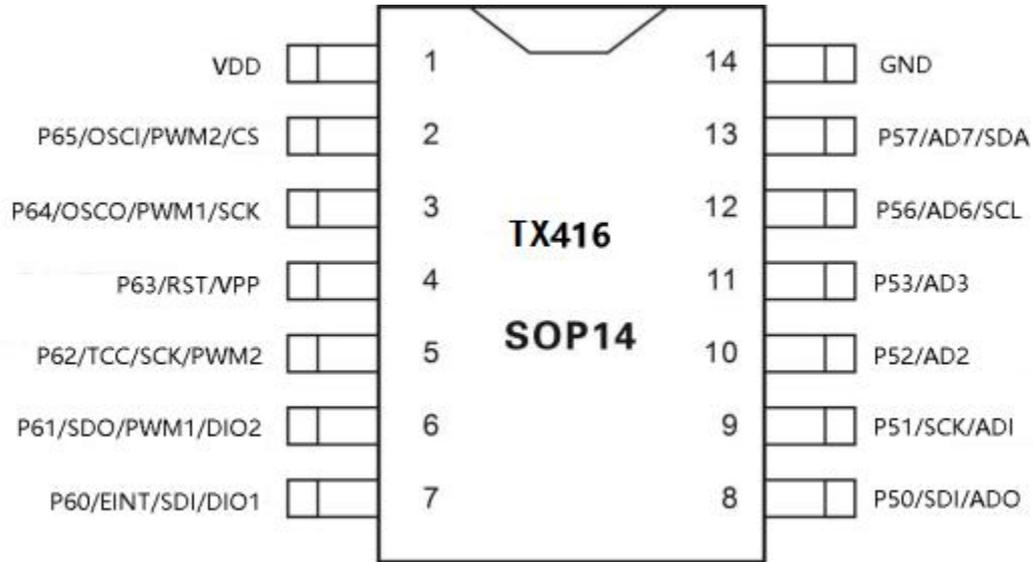
# 二、特征

- 高性能、低功耗的8位微处理器
- 先进的RISC结构
  - 58条指令 – 大多数指令执行时间为单个指令周期（振荡频率/2）
- 外设特点
  - 一个具有预分频器及中断功能和时钟源选择与触发边沿选择的 8 位定时器
  - 一个具有预分频器及中断功能的 8 位脉宽比较器
  - 一个具有独立振荡器的内部看门狗
  - 一个与 IO 口复用的外部中断输入
  - 芯片内置电阻电容振荡器
  - 烧录电路接口及程序代码保护功能
  - 低电压侦测（LVD）功能
  - 低电压复位（LVR）特性
  - 所有 IO 具有唤醒功能
  - 2 通道 8bits PWM/PTMR, 可以设定串接成 16bits PWM/PTMR.
  - 1 个 SPI 接口
  - 12 个可编程选择上拉电阻的 IO 口
  - 7 个可编程选择集电极开路的 IO 口
  - 12 个可编程选择下拉电阻的 IO 口
  - 1 个可编程 Reference level 输出.
  - 8-CH(or 9-CH) 10Bit ADC.
- 特殊的处理器特点

- 上电复位及掉电检测
  - 片内经过标定的 RC 振荡器
  - 片内/片外有 11 个硬件中断源
    1. 外部中断(/INT)。
    2. TCC 溢出中断。
    3. Timer1 比较器匹配中断。
    4. IO 口改变中断（从睡眠模式唤醒）
    5. SPI 中断。
    6. WDT timeout 中断。
    7. ADC 转换完成中断。
    8. Low-voltage detection 中断。
    9. PTMR1/PWM1 比较器匹配中断。
    10. PTMR2/PWM2 比较器匹配中断。
    11. I2C 中断。
  - 8 级堆栈
  - 两种睡眠模式，暂停与唤醒特性可以节省功耗
  - 指令为 16 位宽，对所有寄存器、RAM 的访问都不需要分页；程序 ROM 也不需要分页
  - 表格读取功能
  - 直接和间接数据寻址模式
  - 位操作指令
  - 编程选择振荡的起振时间
- 存储器
- 4K \* 16bit 程序存储器（OTPROM）
  - 256 \* 8bit 数据存储器
  - 128 \* 8bit 特殊功能寄存器
- IO 和封装
- 16 个可用 IO（或 16 个可用 IO 加 1 个 AD 输入）
- 适用的温度范围： -40 ~ 80°C
- 工作电压范围： 2.3 ~ 5.5V
- 工作频率范围
- i. 晶体类型：在 5V 时 DC~20MHz, 在 3V 时 DC~20MHz, 在 2.3V 时, DC~16MHz.
  - ii. 内部 RC 类型：有 8MHz、4MHz、1MHz、455KHz 等可以选。
- TX416 25°C 时的功耗：
1. 正常模式：在 5V/4MHz 时，小于 1.5mA  
在 3V/32768 时，典型 25uA
  2. 睡眠模式：小于 2uA

### 三、管脚分配

#### 封装脚位图



## 管脚描述

引脚名称	引脚号	I/O/P类型	缓冲器类型	功能
P50 /SDI/AD0 P50 SDI AD0		I/O I I	ST/TTL ST AN	数字 I/O 引脚, 可软件设为下拉,IO 口改变从睡眠模式唤醒 可以通过烧录成 SPI 模式的数据输入引脚 模拟通道输入 0
P51 /SCK/AD1 P51 SCK AD1		I/O I/O I	ST/TTL ST AN	数字 I/O 引脚, 可软件设为下拉,IO 口改变从睡眠模式唤醒 可以通过烧录成 SPI 模式的 CLK 引脚 模拟通道输入 1
P52 /AD2 P52 AD2		I/O I	ST/TTL AN	数字 I/O 引脚, 可软件设为下拉,IO 口改变从睡眠模式唤醒 模拟通道输入 2
P53 /AD3 P53 AD3		I/O I	ST/TTL AN	数字 I/O 引脚, 可软件设为下拉,IO 口改变从睡眠模式唤醒 模拟通道输入 3
P54/AD4 P54 AD4		I/O I	ST/TTL AN	数字 I/O 引脚, 可软件设为上下拉,IO 口改变从睡眠模式唤醒 模拟通道输入 4
P55/AD5 P55 AD5		I/O I	ST/TTL AN	数字 I/O 引脚, 可软件设为上下拉,IO 口改变从睡眠模式唤醒 模拟通道输入 5
P56/AD6/SCL P56 AD6 SCL		I/O I I/O	ST/TTL AN ST	数字 I/O 引脚, 可软件设为上下拉,IO 口改变从睡眠模式唤醒 模拟通道输入 6 I2C 的时钟脚
P57/AD7/SDA P57 AD7 SDA		I/O I I/O	ST/TTL AN ST	数字 I/O 引脚, 可软件设为上下拉,IO 口改变从睡眠模式唤醒 模拟通道输入 7 I2C 的数据脚
Vref/AD8 Vref AD8		I/O I	AN AN	参考电压输入脚或内部参考电压输出脚 模拟通道输入 8
P60/EINT/SDI P60 EINT		I/O I	ST/TTL ST	数字 I/O 引脚,集电极开路输出与内部上下拉,IO 口改变从睡眠模式唤醒 外部中断脚,下降沿触发

SDI *TDIO1		I I/O	ST/TTL ST	可以通过烧录成 SPI 模式的数据输入引脚 DIO1（烧录数据脚）
P61/SDO/PWM1 P61  SDO PWM1 *TDIO2		I/O  O O I/O	ST/TTL   ST	数字 I/O 引脚,集电极开路输出与内部上下拉,IO 口改变从睡眠模式唤醒 SPI 模式的数据输出引脚 可以通过烧录选择将 PWM1 的输出定义在这里 DIO2（烧录数据脚）
P62/TCC/SCK/PWM2 P62  TCC SCK PWM2		I/O  I I/O O	ST/TTL  ST ST  O	数字 I/O 引脚,集电极开路输出与内部上下拉,IO 口改变从睡眠模式唤醒 外部时钟与计数输入脚 可以通过烧录成 SPI 模式的 CLK 引脚 可以通过烧录选择将 PWM2 的输出定义在这里
P63/RST P63  RST *VPP		I/O  I	ST/TTL  ST	数字 I/O 引脚,集电极开路输出与内部上下拉, IO 口改变从睡眠模式唤醒 当选择复位脚时为复位功能,低电平复位 OTP 烧录时,此脚是烧录脚 VPP, 7.3V 高压
P64/OSCO/PWM1 P64  OSCO PWM1 *TSCK		I/O  O O I	ST/TTL   ST	数字 I/O 引脚,集电极开路输出与内部上拉,IO 口改变从睡眠模式唤醒 振荡输出脚 可以通过烧录选择将 PWM1 的输出定义在这里 SCK（烧录 CLK 脚）
P65/OSCI/PWM2 P65  OSCI PWM2 *TCS		I/O  I O I	ST/TTL  ST  ST	数字 I/O 引脚,集电极开路输出与内部上拉,IO 口改变从睡眠模式唤醒 振荡输入脚 可以通过烧录选择将 PWM2 的输出定义在这里 CS（烧录 CS 脚）
P66/SSB/SCL/PWM1 P66  SCL SSB PWM1		I/O  I/O I O	ST/TTL  ST ST  O	数字 I/O 引脚,集电极开路输出与内部上拉,IO 口改变从睡眠模式唤醒 I2C 的时钟脚 SPI Slave 输入模式的 /CS 引脚 可以通过烧录选择将 PWM1 的输出定义在这里
P67 /SDA/PWM2 P67  SDA PWM2		I/O  I/O O	ST/TTL  ST	数字 I/O 引脚,集电极开路输出与内部上拉,IO 口改变从睡眠模式唤醒 I2C 的数据脚 可以通过烧录选择将 PWM2 的输出定义在这里
VDD		P		电源输入, 烧录脚
VDDL		P		内部电源输出脚

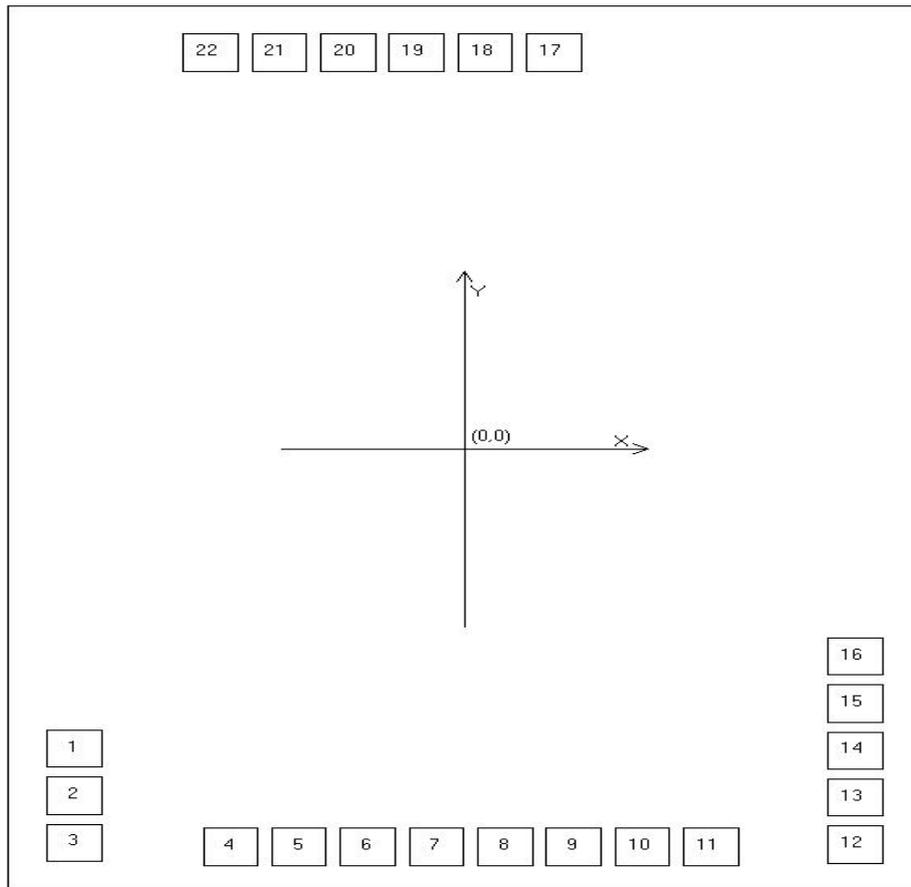
VSS		P		接地端, 烧录脚
AVDD		P		模拟电源输入
AVSS		P		模拟接地端

图注: I = 输入      O = 输出      I/O = 输入/输出      P = 电源  
 - = 未使用      TTL = TTL 输入      ST = 施密特输入      AN = 模拟信号

注:1. 实际管脚顺序以下面 PAD 图为准

## PAD 图

NO.	PAD NAME	NO.	PAD NAME
1	P62/TCC/SCK/PWM2	12	P57/AD7/SDA
2	P61/SDO/PWM1	13	VSS
3	P60/EINT/SDI	14	AVSS
4	VREF/AD8	15	AVDD
5	P50/SDI/AD0	16	VDD
6	P51/SCK/AD1	17	VDDL
7	P52/AD2	18	P65/OSCI/PWM2
8	P53/AD3	19	P64/OSCO/PWM1
9	P54/AD4	20	P63/RST
10	P55/AD5	21	P66/SSB/SCL/PWM1
11	P56/AD6/SCL	22	P67 /SDA/PWM2



衬底接 GND

## 四、存储器

本节讲述 TX416 的存储器。TX416 具有两个主要的存储器空间：数据存储器空间和程序存储器空间。此外，TX416 还有寄存器区对 IC 外设进行特定的操作。数据存储器空间及寄存器空间统一编址，并与程序存储器空间分开。

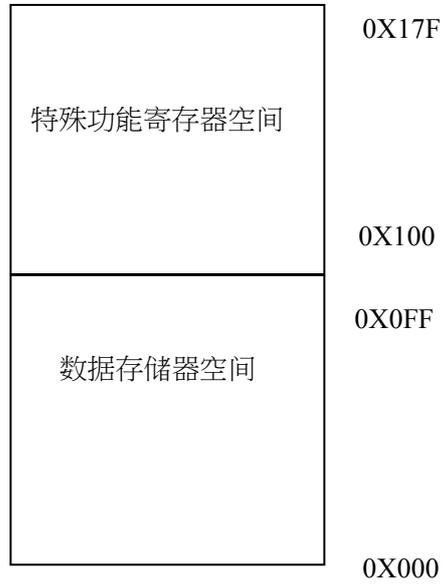
### 4.1 程序存储器

4K \* 16bit 的 OTP ROM 空间，复位矢量地址为 0X000，硬件中断矢量地址为 0X003~0X01F。

复位入口地址	0X000
Low-voltage detection 中断入口地址	0X003
外部 INT 中断入口地址	0X005
IO 口改变唤醒中断入口地址	0X007
定时器 TCC 中断入口地址	0X009
看门狗 WDT 溢出中断入口地址	0X00B
AD 完成中断入口地址	0X00D
PWP (TIMER1) 中断入口地址	0X011
PTMR1/PWM1 中断入口地址	0X013
PTMR2/PWM2 中断入口地址	0X015
SPI 中断入口地址	0X017
I2C 中断入口地址	0X01B
程序存储器空间	0X020
	0XFFF

## 4.2 数据存储器

256 \* 8bit 的 RAM 空间， 有两种寻址方式：直接寻址及通过 INDF 间接寻址。



## 4.3 特殊功能寄存器

128 \* 8bit 的特殊功能寄存器空间采用非连续编址的形式。对于没有对应寄存器的地址进行读写操作，结果未定义。

特殊功能寄存器只能采用直接寻址方式。

## 4.4 特殊功能寄存器表

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INDF	100h	用 FSR 的内容寻址数据存储单元来寻址该地址单元（非实际存在的寄存器）							
TCC	101h	Timer0 模块的寄存器							
PCL	102h	程序计数器（PC）的低字节							
STATUS	103h	RST	GP1	GP0	T	P	Z	DC	C
FSR	104h	间接数据存储地址指针							
P5D	105h	P5 数据锁存器							
P6D	106h	P6 数据锁存器							
P5S	115h	读取 P5 的引脚电平							
P6S	116h	读取 P6 的引脚电平							
WUCON6	120h	WU67	WU66	WU65	WU64	WU63	WU62	WU61	WU60
WUCON5	121h	WU57	WU56	WU55	WU54	WU53	WU52	WU51	WU50
CONT	122h	TS1	/INT	TS0	TE	PAB	PSR2	PSR1	PSR0
OPEN_DRAIN	123h	OD67	OD66	OD65	OD64		OD62	OD61	OD60
IOC5	125h	P5 数据方向寄存器							
IOC6	126h	P6 数据方向寄存器							
PULL_P5H	12Ah	/PH57	/PH56	/PH55	/PH54	/PD57	/PD56	/PD55	/PD54
PULL_DOWN	12Bh	/PD63	/PD62	/PD61	/PD60	/PD53	/PD52	/PD51	/PD50
TICON	12Ch						TM1E	TM1P1	TM1P0
PULL_HIGH	12Dh	/PH67	/PH66	/PH65	/PH64	/PH63	/PH62	/PH61	/PH60
WDTCON	12Eh	EIS	EI_ES	WDTE	SLPC	WDT_CKS			
INTIE	12Fh	WDT_IE			ICIE	TM1IE	SPIIE	EXIE	TCIE
PWM1_DC	130h	PWM1 占空比寄存器							
PWM2_DC	131h	PWM2 占空比寄存器							
PWM1_PR	132h	PWM1 周期寄存器							
PWM2_PR	133h	PWM2 周期寄存器							
PWMCON	134h	PEN2	PEN1	PS2[2:0]			PS1[2:0]		
PTMRCON	135h	PTM1E	PTM1IE	PTM1IF	PTM2E	PTM2IE	PTM2IF	PO_INV	PTM12CA
SPIRB	13Ah	SPI receive buffer							
SPIWB	13Bh	SPI transmit buffer							
SPIS	13Ch	PHA	AU_SSE			SDO_OD	SCK_OD		RBF
SPIC	13Dh	CES	SPIE	SRO	SSE	ORD	SBR2	SBR1	SBR0
TMR1	13Eh	Timer1 模块的寄存器							
PWP	13Fh	脉宽设置寄存器							
ADCON1	140h	ADC	ADEN	AD_IE	AD_IF	CHS3	CHS2	CHS1	CHS0
AD_LSB	141h	DAT1	DAT0						
AD_MSB	142h	DAT9	DAT8	DAT7	DAT6	DAT5	DAT4	DAT3	DAT2
ADCON2	143h	SH	VREF_S				ADPS2	ADPS1	ADPS0

名称	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CCON	144h	I2C_E	RSTA	STA/STO	MS	ACK_TX	RW	BUSY	ARB_LOST
I2CSTA	145h	RX_E	RX_F	TX_E	TX_F	STO_E	STO_F	STA_F	FULL
I2CDAT	146h	I2CD7	I2CD6	I2CD5	I2CD4	I2CD3	I2CD2	I2CD1	I2CD0
I2CSAD	147h	I2CSA6	I2CSA5	I2CSA4	I2CSA3	I2CSA2	I2CSA1	I2CSA0	GC_E
PAL	14Bh	PA[6]	PA[5]	PA[4]	PA[3]	PA[2]	PA[1]	PA[0]	PA_SEL
PAH	14Ch			PA[12]	PA[11]	PA[10]	PA[9]	PA[8]	PA[7]
REFCON	154h	REF_EN	BIAS_R						
REFDAT1	155h				RDB4	RDB3	RDB2	RDB1	RDB0
REFDAT2	156h				RU4	RU3	RU2	RU1	RU0
IO_SINK2	171h								P5H_XS
IO_FUNC	172h	F_I2C[1]	F_I2C[0]		F_CKO[1]	F_CKO[0]	F_SPI	F_PWM[1]	F_PWM[0]
IO_BUF2	173h								P5H_ST
IO_SINK	174h	P523_XS	P501_XS	P67_XS	P66_XS	P645_XS	P623_XS	P61_XS	P60_XS
IO_BUF	175h	P523_ST	P501_ST	P67_ST	P66_ST	P645_ST	P623_ST	P61_ST	P60_ST
ACHCON	17Ah	P57AE	P56AE	P55AE	P54AE	P53AE	P52AE	P51AE	P50AE
GCKCON	17Ch	G_SYS	G_GIO	G_TMR1	G_XINST		G_SPI	G_ANA	G_PWM
LVRCON	17Dh	LVR_ENB	LV_F	LVD_IE	FT_CK1	FT_CK0	LVS2	LVS1	LVS0
R3F(INTIF)	17Fh				ICIF	TM1IF	SPIIF	EXIF	TCIF

## 4.5 特殊功能寄存器描述

说明: ·R: 可读, W: 可写, C: 可清零  
 ·RESET: 复位状态  
 ·U: 表示无变化  
 ·x: 表示不确定

### 4.5.1 与系统相关特殊功能寄存器描述

#### A(累加器)

内部数据传输, 或者指令操作数保持。它是一个不可寻址寄存器。

#### INDF(间接寻址寄存器)(地址:0x100)

INDF 寄存器不是实际存在的寄存器。它用于间接地址的指针。任何对 INDF 进行操作的指令, 实际上是存取由 RAM 选择寄存器 FSR 所指定的 RAM 内容。

#### PCL(程序计数器)和堆栈(地址:0x102)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0
R/W/C	R/W							
RESET	1	1	1	1	1	1	1	1
WDT_out	1	1	1	1	1	1	1	1
Wake_up	U	U	U	U	U	U	U	U

PC 和硬件堆栈为 16 位宽。

产生 4096×16 位的片内 OTP ROM 地址获取指令代码。

复位后 PC 所有位均设置为‘1’。

- “ADD R2, A”允许相对地址被装入当前 PC, 同时 PC 的高 8 位被清零。
- 任何有可能改变 PCL 的值的指令(例如: ”ADD PCL, A”, “MOV PCL, A”, “BC PCL, 6”, ...)(“TBL”除外)都将导致 PC 的高 8 位被清零。因此, 跳转限制在编程页的前 256 个程序存储空间。
- “TBL”允许相对地址被加载到当前 PC(PCL + A→PCL), PC 的高 8 位的值不变。因此, 跳转可以在编程页的任意 256 个程序存储空间。
- 除了改变 PCL 内容指令需要一个以上指令周期外(fclk/2), 其余指令只要一个指令周期。

#### STATUS(状态寄存器) (地址:0x103)

状态寄存器包含 ALU 算术结果的状态和复位状态。

与其它寄存器相同, 状态寄存器可以作为任何指令的目的操作数。如果状态寄存器作为一条指令的目的操作数, 而这条指令又影响了 Z、DC 或 C 标志, 那么就不允许对这 3 位进行写操作。这些位的位置或清零取决于器件的逻辑模块。此外, T 和 P 位是不能进行写操作的, 所以当执行一条把状态寄存器作为目的操作数的指令时, 其结果可能会与预想的不同。

因此, 如果想改变状态寄存器的内容, 建议使用位操作指令 BC、BS 半字节交换指令 SWAP 和传送指令 MOV R, A, 因为这些指令不影响 Z、C 或 DC 标志位。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	RST	GP1	GP0	T	P	Z	DC	C
R/W/C	R/W							
RESET	0	0	0	1	1	0	0	0
WDT_out	0	0	0	0	U	0	0	0
Wake_up	U	U	U	U	U	U	U	U

Bit7(RST) 复位类型

0: 睡眠后从复位脚唤醒;

1: 睡眠后从 IO 口改变唤醒。

Bit6(GP1) 通用读写位。

Bit5(GP0) 通用读写位。

Bit4(T) 超时标志位

0: 发生了 WDT 超时;

1: 上电复位或执行 WDTC 指令、SLEP 指令后被置 1。

Bit3(P) 掉电标志位

0: 执行了 SLEP 指令;

1: 上电复位或执行 WDTC 指令。

Bit2(Z) 零标志位

0: 算术或者逻辑运算的结果不为零;

1: 算术或者逻辑运算的结果为零。

Bit1(DC) 辅助进位/借位位 (ADD, SUB 指令)

0: 未发生执行结果的低 4 位向高 4 位进位;

1: 发生了执行结果的低 4 位向高 4 位进位。

Bit0(C) 进位/借位位 (ADD, SUB 指令)

0: 未发生执行结果向高位进位;

1: 发生了执行结果向高位进位。

注: 对于循环 (RLC、RRC) 指令, 该位装载源寄存器的最高位或最低位。

#### FSR(RAM 选择寄存器) (地址:0x104)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	D7	D6	D5	D4	D3	D2	D1	D0
R/W/C	R/W							
RESET	0	0	0	0	0	0	0	0
WDT_out	0	0	0	0	0	0	0	0
Wake_up	U	U	U	U	U	U	U	U

Bits 0~7 用于在间址寻址下选择寄存器(地址: 00~FF)。

如果没有使用间址寻址方式, R4 可以用作一个 8 位通用读/写寄存器。

**PAL (表格指针低 8 位) (地址: 0x14B)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	A7	A6	A5	A4	A3	A2	A1	A0
R/W/C	R/W							
RESET	X	X	X	X	X	X	X	X
WDT_out	X	X	X	X	X	X	X	X
Wake_up	U	U	U	U	U	U	U	U

**PAH (表格指针高 8 位) (地址: 0x14C)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol				A12	A11	A10	A9	A8
R/W/C				R/W	R/W	R/W	R/W	R/W
RESET				X	X	X	X	X
WDT_out				X	X	X	X	X
Wake_up				U	U	U	U	U

此查表方法可在程序存储器的任意空间使用，不受任何限制，表格指针必须先行设定，其方式是将表格的低 7 位地址放在表格指针寄存器 PAL[7:1]中。PAL[0]用来表示表格的高低字节，PAL[0]为 0 是低字节，PAL[0]为 1 是高字节。将表格的高 6 位地址放在表格指针寄存器 PAH[5:0]，这个寄存器存放表格较高的 6 位地址。在设定完表格指针后，表格数据可以使用“MOVC”指令从当前程序所在的存储器中来查表读取。

例：

CODE\_TAB:

```

MOV     PAL, A           ; 将偏移量送入低 8 位指针
MOV     A, @(CODE_TAB_DB >> 7) ; 取表格指针地址的高 6
                                   ; 位地址
MOV     PAH, A          ; 并送入表格指针的高位地址
MOV     A, @((CODE_TAB_DB << 1) & 0X00FF) ; 取表格指针地址
                                   ; 的低 7 位送到 PAL[7:1]中
ADD     PAL, A          ; 加偏移量并送入低 8 位指针中
MOV     A, @0           ; 判断是否有进位
JBC     STATUS, C       ; 有进位就加 1
MOV     A, @1           ; 没有进位就加 0
ADD     PAH, A          ;
MOVC    A, @PAL         ; 读数查表
RET

```

CODE\_TAB\_DB: ; 低位 8 位数据在前

```

DB     0X014, 0X00F    ;K1,K2
DB     0X00B, 0X00C    ;K3,K4
DB     0X00E, 0X016    ;K5,K6
DB     0X000, 0X001    ;K7,K8

```

**GCKCON (时钟控制寄存器) (地址:0x17C)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	G_SYS	G_GIO	G_TMR1	G_XINST		G_SPI	G_ANA	G_PWM
R/W/C	R/W	R/W	R/W	R/W		R/W	R/W	R/W
RESET	1	1	1	1		1	1	1
WDT_out	1	1	1	1		1	1	1
Wake_up	U	U	U	U		U	U	U

Bit7(G\_SYS) 与系统相关的特殊寄存器开关

可控制的特殊寄存器是： IO\_SINK, IO\_IBUF, IO\_BUF2, ACHCON 和 LVRCON.

1: 使能

0: 关闭

Bit6(G\_GIO) 与 IO 口相关的寄存器开关

可控制的特殊寄存器是： WUCON6, WUCON5, PULL\_P5H, OPEN\_DRAIN, PULL\_DOWN, PULL\_HIGH 和 WDTCON(Bit7, 6, 3, 2, 1, 0).

1: 使能

0: 关闭

Bit5(G\_TMR1) 与 Timer1 相关的寄存器开关

可控制的特殊寄存器是： T1CON, TMR1 和 PWP 以及 I2C

1: 使能

0: 关闭

Bit4(G\_XINST) 扩展指令与相关寄存器的开关

可以控制的指令有： MOVC, PUSH, POP, RET, RETI, CALL, INT.

可控制的特殊寄存器是： PAH, PAL 和 Stack5, Stack6, Stack7, Stack8.

1: 使能

0: 关闭

Bit2(G\_SPI) 与 SPI 相关的寄存器开关

可控制的特殊寄存器是： SPIRB, SPIWB, SPIS 和 SPIC.

1: 使能

0: 关闭

Bit1(G\_ANA) 与 AD 相关的寄存器开关

可控制的特殊寄存器是： ADCON, AD\_LSB, AD\_MSB, ADCON2, REFCON, REFDAT1, REFDAT2.

1: 使能

0: 关闭

Bit0(G\_PWM) 与 PWM 相关的寄存器开关

可控制的特殊寄存器是： PWM1\_DC, PWM2\_DC, PWM1\_PR, PWM2\_PR 和 PWMCON.

1: 使能

0: 关闭

**LVRCON (LVR 控制寄存器) (地址:0x17D)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	LVR_ENB	LVD_F	LVD_IE	FT_CK1	FT_CK0	LVS2	LVS1	LVS0
R/W/C	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	1	1	0
WDT_out	0	U	0	0	0	1	1	0
Wake_up	U	U	U	U	U	U	U	U

Bit7(LVR\_ENB) 低电压复位使能

- 0: 使能低电复位
- 1: 禁止低电压复位

Bit6(LVD\_F) 低电压检测标志位

- 0: 没有低于设定的电压.
- 1: 检测到有低于设定的电压

Bit5(LVD\_IE) 低电压检测中断使能位

- 0: 检测到低电压不中断
- 1: 检测到有低电压中断

Bit4-3(FT\_CK) 低电压复位时钟源

- 00 : 没有应用
- 01 :  $F_{14K}$ , 触发脉宽 > 220us
- 10 :  $F_{14K} / 2$ , 触发脉宽 > 480us
- 11 :  $F_{14K} / 4$ , 触发脉宽 > 960us

Bit2-0(LVS) 低电压复位的检测电压选择

- 000 : 2.0v
- Others : 没有应用

**4.5.2 与中断及看门狗相关特殊功能寄存器描述****WDTCON (WDT 控制寄存器) (地址:0x12E)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	EIS	EI_ES	WDTE	SLPC	WDT_CKS			
R/W/C	R/W	R/W	R/W	R/W	R/W			
RESET	0	0	1	1	0			
WDT_out	0	0	1	1	U			
Wake_up	U	U	1	1	U			

Bit7(EIS) 外部中断选择脚

- 0: 为 P60 通用 IO 口
- 1: 为外部中断脚

Bit6(EI\_ES) 外部中断触发选择

- 0: 外部中断下降沿触发
- 1: 外部中断上升沿触发

Bit5(WDTE) WDT 使能位, 只有在 OPTION CODE 中的 ENWDT 位为 1 时才起作用

- 0: WDT 禁止
- 1: WDT 允许

WDT 控制位。WDTE 位仅在代码选项位 ENWDT 为“0”时无效。也就是说, 如果 ENWDT 位为“0”, 不管 WDTE 位如何设置, WDT 都是无效。WDTE 位仅在代码选项位 ENWDT 为“1”时有效。如果 ENWDT 代码选项位为“1”, 则 WDT 由 WDTE 位设置为无效/有效。0=WDT 无效, 1=WDT 有效。WDTE 位是可读可写的。

Bit4(SLPC) SLEEP2 模式控制, 当写 0 时进入 SLEEP2 模式。

Bit3(WDT\_CKS) WDT 时钟源选择

- 0: 内部 14K RC 振荡为时钟源 (即在没有分频情况下为 18mS 左右溢出)
- 1: 从 OSCO 上的输出做时钟源 (只有在用内部 RC 时才可以)

#### INTIE (中断允许寄存器) (地址:0x12F)

INTIE 寄存器是可读写的寄存器, 包含 TMR0 溢出、INT 下降沿外部中断、及 TMR1 匹配中断等各种使能控制位。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	WDT_IE			ICIE	TM1IE	SPIIE	EXIE	TCIE
R/W/C	R/W			R/W	R/W	R/W	R/W	R/W
RESET	0			0	0	0	0	0
WDT_out	0			0	0	0	0	0
Wake_up	U			U	U	U	U	U

Bit7(WDT\_IE) 看门狗溢出中断允许

- 0: 看门狗溢出中断禁止
- 1: 看门狗溢出中断允许

Bit4(ICIE) IO 口改变中断允许

- 0: IO 口改变中断禁止
- 1: IO 口改变中断允许

Bit3(TM1IE) TIMER1 中断允许

- 0: TIMER1 和 PWP 匹配中断禁止
- 1: TIMER1 和 PWP 匹配中断允许

Bit2(SPIIE) SPI 中断允许

- 0: SPI 禁止中断
- 1: SPI 允许中断

Bit1(EXIE) 外部中断允许

- 0: 外部中断禁止
- 1: 外部中断允许

Bit0(TCIE) TCC 中断允许

- 0: TCC 溢出中断禁止
- 1: TCC 溢出中断允许

#### ISR (INTIF: 中断标志寄存器) (地址:0x17F)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	WDT_IF			ICIF	TM1IF	SPIIF	EXIF	TCIF
R/W/C	R/W			R/W	R/W	R/W	R/W	R/W
RESET	0			0	0	0	0	0
WDT_out	0			0	0	0	0	0
Wake_up	U			U	U	U	U	U

读此寄存器的结果为 IOCF 和 ISR 的逻辑与。

Bit7(WDT\_IF) WDT 溢出中断标志位

- 0: 没有 WDT 溢出中断
- 1: 有 WDT 溢出中断

Bit4(ICIF) IO 口改变中断标志

- 0: IO 口改变没有中断
- 1: IO 口改变有中断

Bit3(TMP1IF) TIMER1 中断标志

- 0: TIMER1 没有中断
- 1: TIMER1 有中断

Bit2(SPIIF) SPI 中断标志

- 0: SPI 没有中断
- 1: SPI 有中断

Bit1(EXIF) 外部中断标志

- 0: 外部没有中断
- 1: 外部有中断

Bit0(TCIF) TCC 中断标志

- 0: TCC 没有中断
- 1: TCC 有中断

- 下列例子P60 为外部中断脚，每进一次外部中断(P60high 变low)时改变P50(1、0 交替)。

```

7          ORG      0X0000
8          JMP      _START
9          ORG      0X0005          //外部INT中断入口地址
10         MOV      A, @0X01
11         XOR      P5, A
12         BC       INTIF, 1          //标志清除
13         RETI
14
15         ORG      0X0100
16 _START:
17         MOV      A, @00000000b    //P50为输出脚
18         MOV      IOC5, A
19         MOV      A, @00000001b    //P60为输入脚
20         MOV      IOC6, A
21         MOV      A, @11111110b    //P60电平拉高
22         MOV      PULL_HIGH, A
23
24         MOV      A, @0XB0          //P60为外部中断脚, 外部中断下降延触发
25         MOV      WDTCON, A
26
27         MOV      A, @0X02          //外部中断允许
28         MOV      INTE, A
29         ENI                       //中断开启
30 ;-----
31 _SCANKEY:
32         NOP
33         JMP      $-1

```

- 下列例子P60 为唤醒脚,进入睡眠时(Sleep2)使用P60 唤醒,唤醒后执行下一行使P50 改变(1、0 交替)。

```

7          ORG      0X0000
8          JMP      _START
9          ORG      0X0007          //IO口改变唤醒中断入口地址
10         MOV      A, @0X01
11         XOR      P5, A
12         BC       INTIF, 1          //标志清除
13         RETI
14
15         ORG      0X0100
16 _START:
17         MOV      A, @00000000b    //P50为输出脚
18         MOV      IOC5, A
19         MOV      A, @00000001b    //P60为输入脚
20         MOV      IOC6, A
21         MOV      A, @11111110b    //P60电平拉高
22         MOV      PULL_HIGH, A
23
24         MOV      A, @0X10          //WDT禁止
25         MOV      WDTCON, A
26
27         MOV      A, @0X10          //IO口改变中断允许
28         MOV      INTE, A
29         BS       WUCON6, 0        //P60输入改变唤醒使能
30         ENI          //中断开启
31 ;;-----
32 _SCANKEY:
33         BC       WDTCON, 4          //进SLEEP2睡眠, 等P60唤醒后执行下一行
34         MOV      A, @0X01
35         XOR      P5, A
36         JMP      $-2

```

### 4.5.3 与 IO 口相关特殊功能寄存器描述

#### P5 D(端口 5 寄存器) (地址:0x105)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	P57	P56	P55	P54	P53	P52	P51	P50
R/W/C	R/W							
RESET	0	0	0	0	0	0	0	0
WDT_out	0	0	0	0	0	0	0	0
Wake_up	U	U	U	U	U	U	U	U

P5 口的输出寄存器

#### P6D (端口 6 寄存器) (地址:0x106)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	P67	P66	P65	P64	P63	P62	P61	P60
R/W/C	R/W							
RESET	0	0	0	0	0	0	0	0

WDT_out	0	0	0	0	0	0	0	0
Wake_up	U	U	U	U	U	U	U	U

P6 口的输出寄存器

#### P5S (端口 5 状态) (地址:0x115)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	P57	P56	P55	P54	P53	P52	P51	P50
R/W/C	R	R	R	R	R	R	R	R

P5 口的读寄存器

#### P6S (端口 6 状态) (地址:0x116)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	P67	P66	P65	P64	P63	P62	P61	P60
R/W/C	R	R	R	R	R	R	R	R

P6 口的读寄存器

#### WUCON6 (P6 唤醒控制寄存器) (地址:0x120)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	WU67	WU66	WU65	WU64	WU63	WU62	WU61	WU60
R/W/C	R/W							
RESET	0	0	0	0	0	0	0	0
WDT_out	0	0	0	0	0	0	0	0
Wake_up	U	U	U	U	U	U	U	U

IO 口输入改变唤醒控制位

Bit7(WU67) P67 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

Bit6(WU66) P66 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

Bit5(WU65) P65 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

Bit4(WU64) P64 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

Bit3(WU63) P63 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

Bit2(WU62) P62 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

Bit1(WU61) P61 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

Bit0(WU60) P60 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

### WUCON5 (P5 唤醒控制寄存器) (地址:0x121)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	WU57	WU56	WU55	WU54	WU53	WU52	WU51	WU50
R/W/C	R/W							
RESET	0	0	0	0	0	0	0	0
WDT_out	0	0	0	0	0	0	0	0
Wake_up	U	U	U	U	U	U	U	U

IO 口输入改变唤醒控制位

Bit7(WU57) P57 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

Bit6(WU56) P56 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

Bit5(WU55) P55 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

Bit4(WU54) P54 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

Bit3(WU53) P53 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

Bit2(WU52) P52 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

Bit1(WU51) P51 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

Bit0(WU50) P50 口输入改变唤醒控制位

0: 禁止唤醒

1: 使能唤醒

#### IOC5 (P5 端口控制寄存器) (地址:0x125)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	P57	P56	P55	P54	P53	P52	P51	P50
R/W/C	R/W							
RESET	1	1	1	1	1	1	1	1
WDT_out	1	1	1	1	1	1	1	1
Wake_up	U	U	U	U	U	U	U	U

“1”设置相关 I/O 管脚为输入脚;

”0”设置相关 I/O 管脚为输出脚。

#### IOC6 (P6 端口控制寄存器) (地址:0x126)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	P67	P66	P65	P64	P63	P62	P61	P60
R/W/C	R/W							
RESET	1	1	1	1	1	1	1	1
WDT_out	1	1	1	1	1	1	1	1
Wake_up	U	U	U	U	U	U	U	U

“1”设置相关 I/O 管脚为输入脚;

”0”设置相关 I/O 管脚为输出脚。

#### PULL\_P5H (P57~P54 上拉下拉控制寄存器) (地址:0x12A)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	/PH57	/PH56	/PH55	/PH54	/PD57	/PD56	/PD55	/PD54
R/W/C	R/W							
RESET	1	1	1	1	1	1	1	1
WDT_out	1	1	1	1	1	1	1	1
Wake_up	U	U	U	U	U	U	U	U

上拉或下拉使能，分别控制各个端口的下拉电阻是否使能，为 0 使能。

Bit7(/PH57) P57 上拉控制位

0: 打开 P57 的上拉;

1: 关闭 P57 的上拉。

Bit6(/PH56) P56 上拉控制位

0: 打开 P56 的上拉;

1: 关闭 P56 的上拉。

Bit5(/PH55) P55 上拉控制位

0: 打开 P55 的上拉;

1: 关闭 P55 的上拉。

Bit4(/PH54) P54 上拉控制位

0: 打开 P54 的上拉;

1: 关闭 P54 的上拉。

Bit3(/PD57) P57 下拉控制位

0: 打开 P57 的下拉;

1: 关闭 P57 的下拉。

Bit2(/PD56) P56 下拉控制位

0: 打开 P56 的下拉;

1: 关闭 P56 的下拉。

Bit1(/PD55) P55 下拉控制位

0: 打开 P55 的下拉;

1: 关闭 P55 的下拉。

Bit0(/PD54) P54 下拉控制位

0: 打开 P54 的下拉;

1: 关闭 P54 的下拉。

#### **PULL\_DOWN (下拉控制寄存器) (地址:0x12B)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	/PD63	/PD62	/PD61	/PD60	/PD53	/PD52	/PD51	/PD50
R/W/C	R/W							
RESET	1	1	1	1	1	1	1	1
WDT_out	1	1	1	1	1	1	1	1
Wake_up	U	U	U	U	U	U	U	U

下拉使能，分别控制各个端口的下拉电阻是否使能，为 0 使能。

Bit7(/PD63) P63 下拉控制位

0: 打开 P63 的下拉;

1: 关闭 P63 的下拉。

Bit6(/PD62) P62 下拉控制位

0: 打开 P62 的下拉;

1: 关闭 P62 的下拉。

Bit5(/PD61) P61 下拉控制位

0: 打开 P61 的下拉;

1: 关闭 P61 的下拉。

Bit4(/PD60) P60 下拉控制位

- 0: 打开 P60 的下拉;
- 1: 关闭 P60 的下拉。

Bit3(/PD53) P53 下拉控制位

- 0: 打开 P53 的下拉;
- 1: 关闭 P53 的下拉。

Bit2(/PD52) P52 下拉控制位

- 0: 打开 P52 的下拉;
- 1: 关闭 P52 的下拉。

Bit1(/PD51) P51 下拉控制位

- 0: 打开 P51 的下拉;
- 1: 关闭 P51 的下拉。

Bit0(/PD50) P50 下拉控制位

- 0: 打开 P50 的下拉;
- 1: 关闭 P50 的下拉。

#### OPEN\_DRAIN (集电极开路控制寄存器) (地址:0x123)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	OD67	OD66	OD65	OD64	-	OD62	OD61	OD60
R/W/C	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0
WDT_out	0	0	0	0	0	0	0	0
Wake_up	U	U	U	U	U	U	U	U

集电极开路使能，分别控制各个端口的集电极开路是否使能，为 1 使能。

Bit7(OD67) P67 集电极开路控制位

- 0: 关闭 P67 的集电极开路;
- 1: 打开 P67 的集电极开路。

Bit6(OD66) P66 集电极开路控制位

- 0: 关闭 P66 的集电极开路;
- 1: 打开 P66 的集电极开路。

Bit5(OD65) P65 集电极开路控制位

- 0: 关闭 P65 的集电极开路;
- 1: 打开 P65 的集电极开路。

Bit4(OD64) P64 集电极开路控制位

- 0: 关闭 P64 的集电极开路;
- 1: 打开 P64 的集电极开路。

Bit2(OD62) P62 集电极开路控制位

0: 关闭 P62 的集电极开路;

1: 打开 P62 的集电极开路。

Bit1(OD61) P61 集电极开路控制位

0: 关闭 P61 的集电极开路;

1: 打开 P61 的集电极开路。

Bit0(OD60) P60 集电极开路控制位

0: 关闭 P60 的集电极开路;

1: 打开 P60 的集电极开路。

#### **PULL\_HIGH (上拉控制寄存器) (地址:0x12D)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	/PH67	/PH66	/PH65	/PH64	/PH63	/PH62	/PH61	/PH60
R/W/C	R/W							
RESET	1	1	1	1	1	1	1	1
WDT_out	1	1	1	1	1	1	1	1
Wake_up	U	U	U	U	U	U	U	U

上拉使能，分别控制各个端口的上拉电阻是否使能，为 0 使能。

Bit7(/PH67) P67 上拉控制位

0: 打开 P67 的上拉;

1: 关闭 P67 的上拉。

Bit6(/PH66) P66 上拉控制位

0: 打开 P66 的上拉;

1: 关闭 P66 的上拉。

Bit5(/PH65) P65 上拉控制位

0: 打开 P65 的上拉;

1: 关闭 P65 的上拉。

Bit4(/PH64) P64 上拉控制位

0: 打开 P64 的上拉;

1: 关闭 P64 的上拉。

Bit3(/PH63) P63 上拉控制位

0: 打开 P63 的上拉;

1: 关闭 P63 的上拉。

Bit2(/PH62) P62 上拉控制位

0: 打开 P62 的上拉;

1: 关闭 P62 的上拉。

Bit1(PH61) P61 上拉控制位

0: 打开 P61 的上拉;

1: 关闭 P61 的上拉。

Bit0(PH60) P60 上拉控制位

0: 打开 P60 的上拉;

1: 关闭 P60 的上拉。

#### IO\_SINK2 (IO Buffer 设定寄存器) (地址:0x171)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol								P5H_XS
R/W/C								R/W
RESET								0
WDT_out								0
Wake_up								U

Bit0(P5H\_XS) 当 P57~P54 口设成输出, 使能该 IO 口 Sink 能力.

0: 禁止加强 IO 口的 Sink 能力

1: 使能加强 IO 口的 Sink 能力

#### IO\_FUNC (IO Function 设定寄存器) (地址:0x172)

	Bit 7-6	Bit 5	Bit 4-3	Bit 2	Bit 1-0
Symbol	F_I2C[1:0]		F_CKO [1:0]	F_SPI	F_PWM[1:0]
R/W/C	R/W		R/W	R/W	R/W
RESET	00		00	0	11
WDT_out	00		00	0	11
Wake_up	U		U	U	U

Bit7-6(F\_I2C) 设定 I2C 脚位

00: SCL = P66; SDA = P67

01: 保留

10: SCL = P56; SDA = P57

11: 保留

Bit4-3(F\_CKO) 设定时钟除频输出于 P66.

00: 没有输出

01: Fosc

10: Fosc / 2

11: Fosc / 4

Bit2 (F\_SPI) 选择 SPI SCK/SDI 脚位

0: SCK = P62 ; SDI = P60.

1: SCK = P51 ; SDI = P50.

Bit1-0(F\_PWM) 选择 PWM 输出的脚位

00: PWM1 = P61; PWM2 = P62

01: PWM1 = P64; PWM2 = P65

10: PWM1 = P66; PWM2 = P67

11: 保留

### IO\_BUF2 (IO Buffer 设定寄存器) (地址:0x173)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol								P5H_ST
R/W/C								R/W
RESET								0
WDT_out								0
Wake_up								U

Bit0(P5H\_ST) 设定 P57~P54 buffer

0: CMOS 输入

1: 施密特输入, 在输出时会串一个 330 左右的电阻再输出

### IO\_SINK (IO SINK Capability 设定寄存器) (地址:0x174)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	P523_XS	P501_XS	P67_XS	P66_XS	P645_XS	P623_XS	P61_XS	P60_XS
R/W/C	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0	0	0	0	0
WDT_out	0	0	0	0	0	0	0	0
Wake_up	U	U	U	U	U	U	U	U

Bit7(P523\_XS) 当 P52, P53 口设成输出, 使能该 IO 口 Sink 能力.

0: 禁止

1: 使能

Bit6(P501\_XS) 当 P50, P51 口设成输出, 使能该 IO 口 Sink 能力.

0: 禁止

1: 使能

Bit5(P67\_XS) 当 P67 口设成输出, 使能该 IO 口 Sink 能力.

0: 禁止

1: 使能

Bit4(P66\_XS) 当 P66 口设成输出, 使能该 IO 口 Sink 能力.

0: 禁止

1: 使能

Bit3(P645\_XS) 当 P64, P65 口设成输出, 使能该 IO 口 Sink 能力.

0: 禁止

1: 使能

Bit2(P623\_XS) 当 P62, P63 口设成输出, 使能该 IO 口 Sink 能力.

0: 禁止

1: 使能

Bit1(P61\_XS) 当 P61 口设成输出, 使能该 IO 口 Sink 能力.

0: 禁止

1: 使能

Bit0(P60\_XS) 当 P60 口设成输出, 使能该 IO 口 Sink 能力.

0: 禁止

1: 使能

### IO\_BUF (IO Buffer 设定寄存器) (地址:0x175)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	P523_ST	P501_ST	P67_ST	P66_ST	P645_ST	P623_ST	P61_ST	P60_ST
R/W/C	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	1	1	1	1	1	1	1	1
WDT_out	1	1	1	1	1	1	1	1
Wake_up	U	U	U	U	U	U	U	U

Bit7(P523\_ST) 设定 P52, P53 buffer

0: CMOS 输入

1: 施密特输入, 在输出时会串一个 330 左右的电阻再输出

Bit6(P501\_ST) 设定 P50, P51 buffer

0: CMOS 输入

1: 施密特输入, 在输出时会串一个 330 左右的电阻再输出

Bit5(P67\_ST) 设定 P67 buffer

0: CMOS 输入

1: 施密特输入, 在输出时会串一个 330 左右的电阻再输出

Bit4(P66\_ST) 设定 P66 buffer

0: CMOS 输入

1: 施密特输入, 在输出时会串一个 330 左右的电阻再输出

Bit3(P645\_ST) 设定 P64, P65 buffer

0: CMOS 输入

1: 施密特输入, 在输出时会串一个 330 左右的电阻再输出

Bit2(P623\_ST) 设定 P62, P63 buffer

0: CMOS 输入

1: 施密特输入, 在输出时会串一个 330 左右的电阻再输出

Bit1(P61\_ST) 设定 P61 buffer

0: CMOS 输入

1: 施密特输入，在输出时会串一个 330 左右的电阻再输出

Bit0(P60\_ST) 设定 P60 buffer

0: CMOS 输入

1: 施密特输入，在输出时会串一个 330 左右的电阻再输出

#### 4.5.4 与定时器 0 相关特殊功能寄存器描述

CONT(控制寄存器) (地址:0x122)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	TS1	/INT	TS0	TE	PAB	PSR2	PSR1	PSR0
R/W/C	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
RESET	1	0	1	1	1	1	1	1
WDT_out	1	0	1	1	1	1	1	1
Wake_up	U	U	U	U	U	U	U	U

Bit6(/INT) 中断使能位，Read Only。

0: 由指令 DISI 设置中断屏蔽；

1: 由指令 ENI/RETI 设置中断使能。

Bit7-Bit5(TS1-TS0) TCC 时钟源的选择位。

11: 内部指令周期做时钟源；

00: 由外部 TCC 脚来做时钟源。

01: OSCO 周期做时钟源；(When CKSRC = RCOSC)

10: reserved.

Bit4(TE) 外部 TCC 触发源选择位

0: 外部 TCC 上升沿计数；

1: 外部 TCC 下降沿计数。

Bit3(PAB)预分频器分配位。

0: 预分频器分配给 TCC；

1: 预分频器分配给 WDT，当预分频器分配给 WDT 后，TCC 为 1:1 分频。

Bit0(PSR0) ~ Bit2(PSR2) TCC/WDT 预分频位

PSR2	PSR1	PSR0	TCC Rate	WDT Rate
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

## TCC(定时寄存器) (地址:0x101)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	D7	D6	D5	D4	D3	D2	D1	D0
R/W/C	R/W							
RESET	0	0	0	0	0	0	0	0
WDT_out	0	0	0	0	0	0	0	0
Wake_up	U	U	U	U	U	U	U	U

此寄存器为 8 位定时器，可由程序进行读/写操作。它用于对内部时钟计数，并拥有可编程的预分器（最大 256 分频）。

- 以下例子为 TCC 从 0x20 开始，数到 0xFF 后进中断改变 P50 状态(1、0 交替)，中断的频率为 11.1kHz。

```

7  ORG  0x0000
8  JMP  _START
9  ORG  0x0009      //TCC中断入口地址
10 JMP  _INT_TCC
11
12 _INT_TCC:
13  MOV  A, @0X20
14  MOV  TMR0, A    //TCC重设 从0x20开始，若无重设则从0开始
15  MOV  A, @0x01
16  XOR  P5, A
17  CLR  INTIF     //TCC中断标志清除
18  RETI
19
20 ORG  0x0100
21 _START:
22  CLR  IOC5     //p50为输出脚
23  CLR  P5
24
25  MOV  A, @0X20 //TCC从0x20开始
26  MOV  TMR0, A
27  BC   WDTCON, 5 // WDT close
28  MOV  A, @0XA0
29  MOV  CONT, A  //内部指令周期作为时钟源，预分频分配给TCC，预分频=1: 2
30
31  BS   INTE, 0  //TCC中断允许
32  ENI           //中断开启
33  NOP
34  JMP  $-1

```



$F_{osc} = 10M$  ,  $F_{sys} = 10M/2 = 5M$

预分频 = 1:2

$0x20 = 32$  ,  $256-32 = 224$

TCC 中断时间 =  $224 \times (1/5M) \times 2 = 89.6\mu s$

#### 4.5.5 与定时器 1 及 PWP 相关特殊功能寄存器描述

##### T1CON (Timer1 控制寄存器) (地址:0x12C)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol						TM1E	TM1P1	TM1P0
R/W/C						R/W	R/W	R/W
RESET						0	0	0
WDT_out						0	0	0
Wake_up						U	U	U

Bit2(TM1E): TIMER1 是否使能

0: 禁止

1: 允许

Bit1~Bit0(TM1P1-0): TIMER1 预分频比选择 (对 Fsys 进行预分频)

TM1P1	TM1P0	分频
0	0	1:2
0	1	1:8
1	0	1:16
1	1	1:32

##### TMR1 (TIMER1 寄存器) (地址:0x13E)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	TMR17	TMR16	TMR15	TMR14	TMR13	TMR12	TMR11	TMR10
R/W/C	R/W							
RESET	0	0	0	0	0	0	0	0
WDT_out	0	0	0	0	0	0	0	0
Wake_up	U	U	U	U	U	U	U	U

该寄存器中为 TIMER1 计数值, 当计到和 PWP 相等时重新从 00 开始。

##### PWP (脉宽预置寄存器) (地址:0x13F)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	PWP7	PWP6	PWP5	PWP4	PWP3	PWP2	PWP1	PWP0
R/W/C	R/W							
RESET	1	1	1	1	1	1	1	1
WDT_out	1	1	1	1	1	1	1	1
Wake_up	U	U	U	U	U	U	U	U

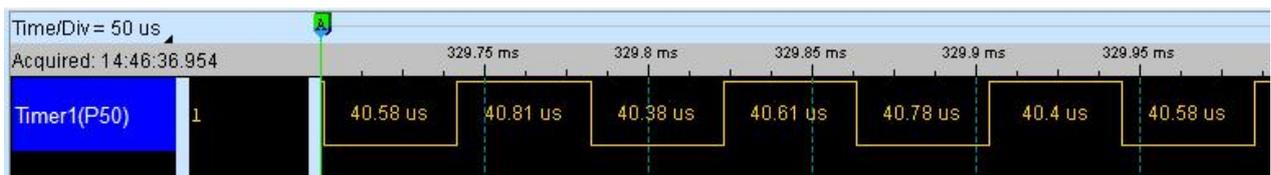
PWP7~ PWP0 设置成与波特率时钟期望宽度。

- 以下例子为 TIMER1 从0 开始，数到100 后进中断改变P50 状态(1 、0 交替)，中断的频率为 25khz。

```

8  ORG  0x0000
9  JMP  _START
10 ORG  0x0011      //PWP (TIMER1) 中断入口地址
11 JMP  _INT_TIMER1
12
13 _INT_TIMER1:
14  MOV  A, @0x01
15  XOR  P5, A
16  BC   INTIF, 3   //TIMER1中断标志清除
17  RETI
18
19 ORG  0x0100
20 _START:
21  CLR  IOC5
22  CLR  P5
23
24  CLR  STATUS
25  CLR  FSR
26  CLR  T1CON      //Timer1控制寄存器
27  CLR  TMR1
28  CLR  PWP
29  CLR  INTIF
30
31  MOV  A, @0X10   //Watch dog off
32  MOV  WDTCON, A
33  MOV  A, @0X08
34  MOV  INTE, A    // Bit3(TM1IE)TIMER1中断允许
35  MOV  A, @100
36  MOV  PWP,A      //PWP设100, 100后进中断
37
38  MOV  A, @0x04
39  MOV  T1CON, A   //time1 enable,分频1:2
40
41  ENI             //中断开启
42  NOP
43  JMP  $-1

```



$F_{osc} = 10M$  ,  $F_{sys} = 10M/2 = 5M$

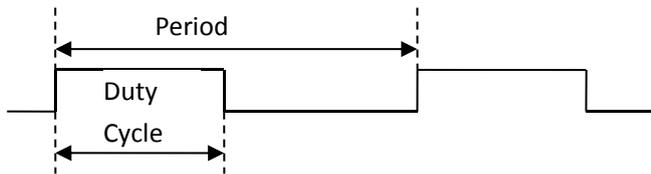
预分频 = 1:2

TIMER1 中断时间 =  $100( (1/5M) * 2) = 40us$

## 4.5.6 与 PWM 相关特殊功能寄存器描述

### PWM1\_DC (PWM1 占空比寄存器) (地址:0x130)

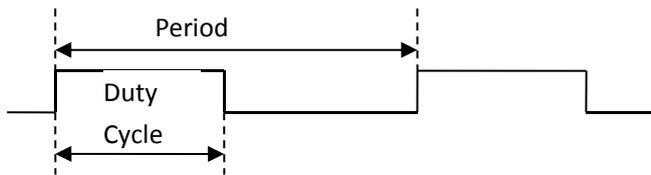
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	DAT7	DAT6	DAT5	DAT4	DAT3	DAT2	DAT1	DAT0
R/W/C	R/W							
RESET	0	0	0	0	0	0	0	0
WDT_out	0	0	0	0	0	0	0	0
Wake_up	U	U	U	U	U	U	U	U



$$\text{Duty Cycle} = (\text{PWM1\_DC}/\text{PWM1\_PR}) * 100\%$$

### PWM2\_DC (PWM2 占空比寄存器) (地址:0x131)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	DAT7	DAT6	DAT5	DAT4	DAT3	DAT2	DAT1	DAT0
R/W/C	R/W							
RESET	0	0	0	0	0	0	0	0
WDT_out	0	0	0	0	0	0	0	0
Wake_up	U	U	U	U	U	U	U	U



$$\text{Duty Cycle} = (\text{PWM2\_DC}/\text{PWM2\_PR}) * 100\%$$

### PWM1\_PR (PWM1 周期寄存器) (地址:0x132)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	DAT7	DAT6	DAT5	DAT4	DAT3	DAT2	DAT1	DAT0
R/W/C	R/W							
RESET	1	1	1	1	1	1	1	1
WDT_out	1	1	1	1	1	1	1	1
Wake_up	U	U	U	U	U	U	U	U

### PWM2\_PR (PWM2 周期寄存器) (地址:0x133)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	DAT7	DAT6	DAT5	DAT4	DAT3	DAT2	DAT1	DAT0
R/W/C	R/W							
RESET	1	1	1	1	1	1	1	1
WDT_out	1	1	1	1	1	1	1	1
Wake_up	U	U	U	U	U	U	U	U

**PWMCON (PWM 控制寄存器) (地址:0x134)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	PEN2	PEN1	PS2[2:0]		PS1[2:0]			
R/W/C	R/W	R/W	R/W		R/W			
RESET	0	0	000		000			
WDT_out	0	0	000		000			
Wake_up	U	U	U		U			

Bit7(PEN2) PWM2 输出使能位。

0: 禁止 PWM2 输出

1: 允许 PWM2 输出

Bit6(PEN1) PWM1 输出使能位。

0: 禁止 PWM1 输出

1: 允许 PWM1 输出

Bit[5:3](PS2[2:0]) PWM2 时钟预分频 (注: 是对系统频率再分频, 而不是振荡频率)

Bit[2:0](PS1[2:0]) PWM1 时钟预分频 (注: 是对系统频率再分频, 而不是振荡频率)

注意: PWM 的使能, 还需要将 PTMR 打开, 否则输出不了 PWM。

PS[2]	PS[1]	PS[0]	Clock (Hz)	Period
0	0	0	Fsys/1	
0	0	1	Fsys/2	
0	1	0	Fsys/4	
0	1	1	Fsys/8	
1	0	0	Fsys/16	
1	0	1	Fsys/32	
1	1	0	Fsys/64	
1	1	1	Fsys/128	

PWM 的计算:

PWM 频率 = (振荡频率 / PWM 时钟分频) / PWM\_PR 的值

占空比 = (PWM\_DC / PWM\_PR) \* 100%

例如: 振荡频率 = 455KHz 预分频为 0 即 1 分频(假设 Fsys = Fosc / 2)

PWM\_PR = 6 PWM\_DC = 2

PWM 频率 = 455K / 2 / 6 = 37.9KHz

占空比 = 2 / 6 \* 100% = 33.3%

**PTMRCON (PTMR1/PTMR2 控制寄存器) (地址:0x135)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	PTM1E	PTM1IE	PTM1IF	PTM2E	PTM2IE	PTM2IF	PO_INV	PTM12CA
R/W/C	R/W	R/W	R/C	R/W	R/W	R/C	R/W	R/W
RESET	0	0	0	0	0	0	0	0
WDT_out	0	0	0	0	0	0	0	0
Wake_up	U	U	U	U	U	U	U	U

Bit7(PTM1E): PTIMER1 是否使能

0: 禁止

1: 使能

Bit6(PTM1IE): PTIMER1 中斷是否使能

0: 禁止中断

1: 允许中断

Bit5(PTM1IF): PTIMER1 中断标志位

0: 中断没有发生

1: 有中断发生,

当在分立模式下, PTMR1=PWM1\_PR 时产生中断标志

当在层叠模式下, PTMR16= (PWM2\_DC, PWM1\_DC) 时产生中断标志

Bit4(PTM2E): PTIMER2 是否使能

0: 禁止

1: 使能

Bit3(PTM2IE): PTIMER2 中斷是否使能

0: 禁止中断

1: 允许中断

Bit2(PTM2IF): PTIMER2 中断标志位

0: 没有中断发生

1: 有中断发生

当在分立模式下, PTMR2=PWM2\_PR 时产生中断标志

当在层叠模式下, PTMR16= (PWM2\_PR, PWM1\_PR) 时产生中断标志

Bit1(PO\_INV): PWM 输出是高有效还是低有效选择

0: 平时为低, PWM 输出高有效

1: 平时为高, PWM 输出低有效

Bit0(PTM12CAS): PTIMER1 和 PTIMER2 8 位模式与 16 位模式选择

0: 独立的 8 位模式

1: 层叠的 16 位模式

(1) 16bits PWM 的周期寄存器 MSB=PWM2\_PR, LSB=PWM1\_PR.

- (2) 16bits PWM 的占空比寄存器 MSB=PWM2\_DC, LSB=PWM1\_DC.
- (3) 16bits PWM, Period (PWM16\_PR) MSB=PWM2\_PR, LSB=PWM1\_PR.  
Duty (PWM16\_DC) MSB=PWM2\_DC, LSB=PWM1\_DC.  
PWM16 频率= (振荡频率 / PWM1 时钟分频) / PWM16\_PR 的值  
占空比= (PWM16\_DC / PWM16\_PR) \* 100%  
若系统设置为二分频, 16bits PWM 频率范围  
最快 : PWM16 频率= $F_{osc} / 2 / (256^2 - 1) = F_{osc} / 131070$ ,  
Duty=0, 1/65535, 2/65535, ..., 65535/65535  
最慢 : PWM16 频率= $F_{osc} / 256 / (256^2 - 1) = F_{osc} / 16776960$ ,  
Duty=0, 1/65535, 2/65535, ..., 65535/65535
- (4) (PWM2\_PR, PWM1\_PR) >= (PWM2\_DC, PWM1\_DC)

- 以输出PWM1 站控比20%, PWM2 站控比50% 为例

```

6 _START:
7 MOV A, @0X00
8 MOV IOC6, A ;设置IO口 P6D所有口为输出口
9
10 MOV A, @20
11 MOV PWM1DC, A
12 MOV A, @100
13 MOV PWM1PR, A ;PWM1占空比= (20/100) *100%=20%
14
15 MOV A, @50
16 MOV PWM2DC, A
17 MOV A, @100
18 MOV PWM2PR, A ;PWM2占空比= (50/100) *100%=50%
19
20 MOV A, @0X00
21 MOV IO_FUNC, A ;设置PWM输出脚 PWM1=P61 PWM2=P62
22
23 MOV A, @0XC9
24 MOV PWM1CON, A ;PWM1 2输出使能, 时钟预分频=FSYS/2
25
26 MOV A, @0X90
27 MOV PTMR1CON, A ;PTIMER1 2 使能 (务必开启, 开启后PWM才能动作)

```



$F_{osc} = 10M$

$F_{sys} = 10M/2 = 5M$

预分频 =  $5M/2 = 2.5M$

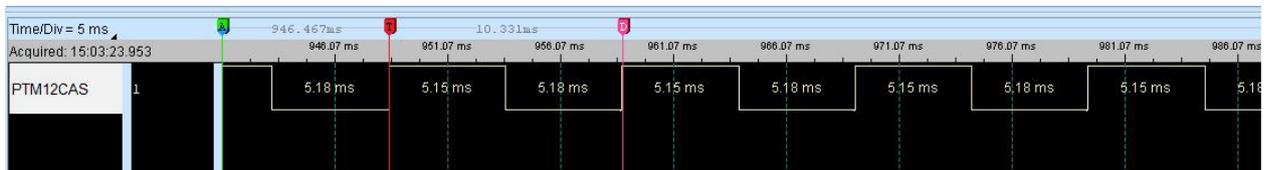
PWM 频率 =  $(2.5M/100) = 25kHz (40us)$

- 下述例子为PTM12CAS 16 模式(PWM2 输出)

```

6 _START:
7 MOV A, @0X00
8 MOV IOC6, A ;设置IO口 P6D所有口为输出
9
10 MOV A, @20
11 MOV PWM1DC, A
12 MOV A, @100
13 MOV PWM1PR, A ;PWM1占空比= (20/100) *100%=20%
14
15 MOV A, @50
16 MOV PWM2DC, A
17 MOV A, @100
18 MOV PWM2PR, A ;PWM2占空比= (50/100) *100%=50%
19
20 MOV A, @0X00
21 MOV IO_FUNC, A ;设置PWM输出脚 PWM1=P61 PWM2=P62
22
23 MOV A, @0XC9
24 MOV PWM1CON, A ;PWM1 2输出使能, 时钟预分频=FSYS/2
25
26 MOV A, @0X91
27 MOV PTMR1CON, A ;PTIMER1 2 使能(务必开启, 开启后PWM才能动作)
28 ;PTM12CAS 16位模式(由PWM2输出)

```



$PWM\ 16DC = (MSB=PWM2\_DC = 50, LSB=PWM1\_DC = 20) = 0x3214 = 12820$  PWM  
 $16\ PR = (MSB=PWM2\_PR = 100, LSB=PWM1\_PR = 100) = 0x6464 = 25700$  占空比=  
 $(12820 / 25700) * 100\% = 49\%$

$F_{osc} = 10M, F_{sys} = 10M/2 = 5M$

预分频 =  $5M/2 = 2.5M$

PTM12CAS 频率 =  $(2.5M/25700) = 97.27Hz(10.28ms)$

## 4.5.7 与 SPI 相关特殊功能寄存器描述

**SPIRB (SPI 接收数据寄存器) (地址:0x13A)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	DAT7	DAT6	DAT5	DAT4	DAT3	DAT2	DAT1	DAT0
R/W/C	R/W							

RESET	U	U	U	U	U	U	U	U
WDT_out	U	U	U	U	U	U	U	U
Wake_up	U	U	U	U	U	U	U	U

**SPIWB (SPI 发送数据寄存器) (地址:0x13B)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	DAT7	DAT6	DAT5	DAT4	DAT3	DAT2	DAT1	DAT0
R/W/C	R/W							
RESET	U	U	U	U	U	U	U	U
WDT_out	U	U	U	U	U	U	U	U
Wake_up	U	U	U	U	U	U	U	U

**SPIS (SPI 状态寄存器) (地址:0x13C)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	PHA	AU_SSE			SDO_OD	SCK_OK		RBF
R/W/C	R/W	R/W			R/W	R/W		R/W
RESET	1	0			0	0		0
WDT_out	1	0			0	0		0
Wake_up	U	U			U	U		U

Bit7(PHA) : 時鐘相位選擇  
与 SPIC 的 BIT7 联合使能

Bit6(AU\_SSE): 不必設置 SSE, 資料將自動傳送及接收.  
主模式下, 一旦给 SPIWB 写数, 就自动启动 SPI 发送和接收  
从模式下, 一旦有 SCK 过来, 就自动启动 SPI 发送和接收  
0: 禁止自动发送和接收  
1: 使能自动发送和接收

Bit 3 (SDO\_OD): SDO 集电极开路控制位  
0 : SDO禁止集电极开路  
1 : SDO使能集电极开路

Bit 2 (SCK\_OD): SCK集电极开路控制位  
0 : SCK禁止集电极开路  
1 : SCK使能集电极开路

Bit 0 (RBF): 读缓冲区满标志位  
0 : 接收没有完成, 读缓冲区没有满  
1 : 接收完成, 读缓冲区满了

**SPIC (SPI 控制寄存器) (地址:0x13D)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	CES	SPIE	SRO	SSE	ORD	SBRS2	SBRS1	SBRS0
R/W/C	R/W							

RESET	0	0	0	0	0	0	0	0
WDT_out	0	0	0	0	0	0	0	0
Wake_up	U	U	U	U	U	U	U	U

Bit7(CES): 时钟边沿选择

当 SPIS. 7 (PHA) =1 时

CES=1: 下降沿移出数据, 上升沿移入数据 (SCK 默认为 1)

CES=0: 上升沿移出数据, 下降沿移入数据 (SCK 默认为 0)

当 SPIS. 7 (PHA) =0 时

CES=1: 上升沿移出数据, 下降沿移入数据 (SCK 默认为 1)

CES=0: 下降沿移出数据, 上升沿移入数据 (SCK 默认为 0)

Bit6(SPIE): SPI 允许位

0: 禁止 SPI

1: 允许 SPI

Bit5(SRO): SPI 缓冲溢出标志位

0: 没有溢出

1: 表示在新的数据装入缓冲时上次数据未读出, 仅在 slave 模式有效

Bit4(SSE): SPI 移位允许

0: 移位完成, 由硬件清 0

1: SPI 开始移出数据, 由软件置 1

Bit3(ORD): 数据传输顺序

0: MSB 优先

1: LSB 优先

Bit2~Bit0(SBRS2~SBRS0): SPI 波特率选择

SBRS2	SBRS1	SBRS0	MODE	波特率
0	0	0	Master	Fsys/1
0	0	1	Master	Fsys/2
0	1	0	Master	Fsys/4
0	1	1	Master	Fsys/8
1	0	0	Master	Fsys/16
1	0	1	Slave	/SS enable
1	1	0	Slave	/SS disable
1	1	1	Master	TMR1/2

使用SPI 传送一笔数据(0x55), 再读取传送的数据

```

10 _START:
11 CLR IOC5
12 MOV A, @0X89 ;P60为输入脚
13 MOV IOC6, A
14
15 BC IO_FUNC, 2 ;SCK = P62, SDI = P60, SDO = P61
16
17 CLR SPIWB
18 CLR SPIRB
19
20 MOV A, @0X80 ;PHA = 1
21 MOV SPIST, A
22 MOV A, @0XC4 ;SPI允许,波特率=Fsys/16,下降沿移出数据,上升沿移入数据(SCK default '1')
23 MOV SPICON, A
24
25 BS P5, 3 ;P53 = CE ,CE default 在高位
26 BS P6, 2 ;SCK default 在高位
27
28 _MASTER_WRITE:
29 JBS P6S, 7
30 JMP $-1
31 MOV A, @0X55
32 MOV SPIWB, A ;SPIWB=SPI transmit buffer, 0x55
33 BC P5, CE ;CE(=3) pull down SPI starts
34 BS SPICON, SSE ;SSE(=4): SPI移位允许, 设为1 SPI开始移出数据, 移完后由硬件清0
35 JBC SPICON, SSE ;判断位移是否完成
36 JMP $-1
37
38 _MASTER_READ:
39 MOV A, SPIRB ;读取SPIRB的值
40 XOR A, @0X55
41 JBS STATUS, Z ;判断是否等于0X55
42 JMP _ERROR
43 BS P5, CE ;CE回到高位
44 JMP _MASTER_WRITE
45
46 _ERROR:
47 BS P5, 0
48 BC P5, 0
49 JMP $-2

```

## 4.5.8 与 I2C 相关特殊功能寄存器描述

### I2CCON (I2C 控制寄存器) (地址: 0x144)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	I2C_E	RSTA	STA/STO	MS	ACK_TX	RW	BUSY	ARB_LOST
R/W/C	R/W	R/W	R/W	R/W	R/W	R	R	R/W
RESET	0	0	0	0	0	0	0	1
WDT_out	X	X	X	X	X	X	X	X
Wake_up	U	U	U	U	U	U	U	U

Bit7(I2C\_E): I2C 使能位

- 0: 禁止 I2C 功能
- 1: 使能 I2C 功能

Bit6(RSTA): 发送‘RE-START’ 位 (仅主模式)

从 0 变到 1: 发送‘RE-START’ 位

Bit5(STA/STO): 发送 Start/Stop 位 (仅主模式)

从 0 变到 1: 发送 ‘START’ 位

从 1 变到 0: 发送 ‘STOP’ 位

Bit4(MS): 主从模式选择位

1: 主模式

0: 从模式

Bit3(ACK\_TX): ACK 回答位

0: 回覆 ACK

1: 回覆 NACK

Bit2(RW): 读写标志位

0: 主模式下为写, 从模式下为读

1: 从模式下为写, 主模式下为读

Bit1(BUSY): I2C 忙状态位

0: 空闲

1: 忙, 当检测到 ‘START’ 时置 1, 当检测到 ‘STOP’ 时清 0

Bit0(ARB\_LOST): I2C 失败标志

0: 由软件清零

1: 当 I2C 有错误时, 由硬件置 1

### I2CSTA (I2C 状态寄存器 ) (地址: 0x145)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	RX_E	RX_F	TX_E	TX_F	STO_E	STO_F	STA_F	FULL
R/W/C	R/W	R/C	R/W	R/C	R/W	R/C	R/C	R
RESET	0	0	0	0	0	0	0	0
WDT_out	X	X	X	X	X	X	X	X
Wake_up	U	U	U	U	U	U	U	U

Bit7(RX\_E): I2C 接收中断使能位

0: 禁止 I2C 接收中断

1: 使能 I2C 接收中断

Bit6(RX\_F): I2C 接收完成标志位

0: 由软件清 0

1: I2C 接收完一个 Byte, 由硬件置 1

Bit5(TX\_E): I2C 发送中断使能位

0: 禁止 I2C 发送中断

1: 使能 I2C 发送中断

Bit4(TX\_F): I2C 发送完成标志位

- 0: 由软件清 0
- 1: I2C 发送完一个 Byte, 由硬件置 1

Bit3(STO\_E): I2C 收到 ‘STOP’ 中断使能位

- 0: 禁止 I2C 收到 ‘STOP’ 中断
- 1: 使能 I2C 收到 ‘STOP’ 中断

Bit2(STO\_F): I2C 收到 ‘STOP’ 的标志位

- 0: 由软件清 0
- 1: 当收到 ‘STOP’后, 由硬件置 1

Bit1(STA\_F): 收到 ‘START’ 或 ‘Re-START’ 位的标志位

- 0: 由软件清 0
- 1: 当收以 ‘STOP’ 或 ‘Re-START’位, 由硬件置 1

Bit0(FULL): I2C 数据缓冲区是否为空

- 0: 为空, 在发送完成后或接收后有读缓冲区, 由硬件清 0
- 1: 为满, 在发送前有数据写入或收到数据后, 由硬件置 1

#### I2CDAT (I2C 数据寄存器 ) (地址: 0x146)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	I2CD7	I2CD6	I2CD5	I2CD4	I2CD3	I2CD2	I2CD1	I2CD0
R/W/C	R/W							
RESET	0	0	0	0	0	0	0	0
WDT_out	X	X	X	X	X	X	X	X
Wake_up	U	U	U	U	U	U	U	U

I2CD7-I2CD0: I2C 接收发送数据寄存器

#### I2CSAD (I2C 从模式地址寄存器) (地址: 0x147)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	I2CSA6	I2CSA5	I2CSA4	I2CSA3	I2CSA2	I2CSA1	I2CSA0	GC_E
R/W/C	R/W	R/W						
RESET	0	0	0	0	0	0	0	0
WDT_out	X	X	X	X	X	X	X	X
Wake_up	U	U	U	U	U	U	U	U

Bit0(GC\_E): 从模式地址寄存器使能位

- 0: 禁止 (即不使用地址包)
- 1: 使能 (即使用地址包)

Bit7-Bit1(I2CSA6-0):

从模式下: I2C 的地址= I2CSA6-I2CD0

主模式下: I2C 的速率为, SCL 的时钟频率=  $F_{sys}/(I2CSAD+3)*2$ , I2CSAD 不能为 0

## 4.5.9 与 ADC 相关特殊功能寄存器描述

### ADCON1 (ADC 控制寄存器) (地址: 0x140)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	ADC	ADEN	AD_IE	AD_IF	CHS3	CHS2	CHS1	CHS0
R/W/C	R/W							
RESET	0	0	0	0	0	0	0	0
WDT_out	X	X	X	X	X	X	X	X
Wake_up	U	U	U	U	U	U	U	U

Bit7(ADC): AD 转换标志位

0: AD 转换完成, 由硬件清 0

1: 允许 AD 转换, 由软件置 1

Bit6 (ADEN): AD 使能位

0: 禁止 AD

1: 使能 AD

Bit5(AD\_IE): AD 中断使能位

0: 禁止 AD 完成中断

1: 允许 AD 完成中断

Bit4(AD\_IF): AD 中断标志位

0: 没有中断, 由软件清 0

1: AD 完成产生中断标志, 由硬件置 1

Bit3-0(CHS3-CHS0): AD 通道选择(0~8)

0000: AD 通道 0, P50

0001: AD 通道 1, P51

0010: AD 通道 2, P52

0011: AD 通道 3, P53

0100: AD 通道 4, P54

0101: AD 通道 5, P55

0110: AD 通道 6, P56

0111: AD 通道 7, P57

1111: AD 通道 8, VREF

### AD\_LSB (ADC LSB 数据寄存器) (地址: 0x141)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	DAT1	DAT0						
R/W/C	R	R						
RESET	X	X						

WDT_out	X	X						
Wake_up	U	U						

**AD\_MSB (ADC MSB 数据寄存器) (地址: 0x142)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	DAT9	DAT8	DAT7	DAT6	DAT5	DAT4	DAT3	DAT2
R/W/C	R	R	R	R	R	R	R	R
RESET	X	X	X	X	X	X	X	X
WDT_out	X	X	X	X	X	X	X	X
Wake_up	U	U	U	U	U	U	U	U

**ADCON2 (ADC 控制寄存器 2) (地址: 0x143)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	SH	VREF_S				ADPS2	ADPS1	ADPS0
R/W/C	R/W	R/W				R/W	R/W	R/W
RESET	0	0				0	0	0
WDT_out	X	X				X	X	X
Wake_up	U	U				U	U	U

Bit7(SH): AD 采样与保持使能位

- 0: 使能
- 1: 禁止

Bit6(VREF\_S): AD 参考源的选择

- 0: 使用 AVDD 的电压做 AD 电压参考源
- 1: 使用 VREF 脚上的电压做 AD 电压参考源

Bit2-0 (ADPS2-ADPS0): ADC 时钟源

ADPS2-ADS0	Clock (Hz)
000	Fsys/2
001	Fsys/4
010	Fsys/6
011	Fsys/8
100	Fsys/10
101	Fsys/12
110	Fsys/14
111	Fsys/16

**ACHCON (模拟通道使能控制寄存器) (地址:0x17A)**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	P57AE	P56AE	P55AE	P54AE	P53AE	P52AE	P51AE	P50AE
R/W/C	R/W							
RESET	0	0	0	0	0	0	0	0
WDT_out	0	0	0	0	0	0	0	0

---

Wake_up	U	U	U	U	U	U	U	U
---------	---	---	---	---	---	---	---	---

Bit7(P57AE) P57 模拟通道与数字通道选择

0: 数字通道

1: 模拟通道

Bit6(P56AE) P56 模拟通道与数字通道选择

0: 数字通道

1: 模拟通道

Bit5(P55AE) P55 模拟通道与数字通道选择

0: 数字通道

1: 模拟通道

Bit4(P54AE) P54 模拟通道与数字通道选择

0: 数字通道

1: 模拟通道

Bit3(P53AE) P53 模拟通道与数字通道选择

0: 数字通道

1: 模拟通道

Bit2(P52AE) P52 模拟通道与数字通道选择

0: 数字通道

1: 模拟通道

Bit1(P51AE) P51 模拟通道与数字通道选择

0: 数字通道

1: 模拟通道

Bit0(P50AE) P50 模拟通道与数字通道选择

0: 数字通道

1: 模拟通道

使用AVDD 脚上的电压做AD 的参考电压源

```

11 _START:
12 MOV A, @00000010B
13 MOV IOC5, A ;设置I/O管脚 ,P51 设置 input pin floating
14
15 MOV A, @0X41 ;AD模块使能,P5.1 AD Input
16 MOV ADCON1, A
17
18 MOV A, @0x03 ;使用AVDD脚上的电压做AD的参考电压源, AD模块时钟频率=Fsys/8
19 MOV ADCON2, A
20
21 BS ACHCON, 1 ;P5.1 模拟通道输入
22
23 _AD_RUN:
24 BS ADCON1, 7 ; AD转换开启
25 BC ADCON1, 7 ;AD转换关闭,为了限制只转一笔数据
26 JBS ADCON2, 4 ;判断AD转换完成标志
27 JMP $-1
28
29 MOV A, ADMSB ;储存AD转换后的高8位值到 TEMP_MSB
30 MOV TEMP_MSB, A
31
32 MOV A, ADLSB ;储存AD转换后的低4位值到 TEMP_LSB
33 MOV TEMP_LSB, A
34 ...
35 JMP _AD_RUN ;转下一笔资料

```

#### 4.5.10 与 VREF 相关特殊功能寄存器描述

##### REFCON (内部参考电压控制寄存器) (地址:0x154)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol	REF_EN	BIAS_R						
R/W/C	R/W	W						
RESET	0	0						
WDT_out	0	0						
Wake_up	U	U						

Bit7(REF\_EN) 内部参考电压使能位

- 0: 当选择用 REF 脚做 AD 参考电压源时, 用 REF 脚上接的外部电压做 AD 参考电压源
- 1: 当选择用 REF 脚做 AD 参考电压源时, 用 REF 脚内部产生的电压做 AD 参考电压源

Bit6(BIAS\_R) 内部反馈电阻是否使能

- 0: 80K 内部反馈电阻
- 1: 10K 内部反馈电阻

注意: Bit6 不可以读

##### REFDAT1 (VReference 设定寄存器) (地址:0x155)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol				RDB4	RDB3	RDB2	RDB1	RDB0

R/W/C				R/W	R/W	R/W	R/W	R/W
RESET				0	0	0	0	0
WDT_out				0	0	0	0	0
Wake_up				U	U	U	U	U

Bit4-0(RDB4-0) Resistor RDB of reference circuit, the corresponding R value is 320K, 160K, 80K, 40K, 20K.

0: the corresponding R is enabled;

1: otherwise.

#### REFDAT2 (VReference 微调寄存器) (地址:0x156)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Symbol				RU4	RU3	RU2	RU1	RU0
R/W/C				R/W	R/W	R/W	R/W	R/W
RESET				0	0	0	0	0
WDT_out				0	0	0	0	0
Wake_up				U	U	U	U	U

Bit6-0(RU6-0)Resistor RU of reference circuit, the corresponding R value is 60K, 30K, 15K, 8K, 4K.

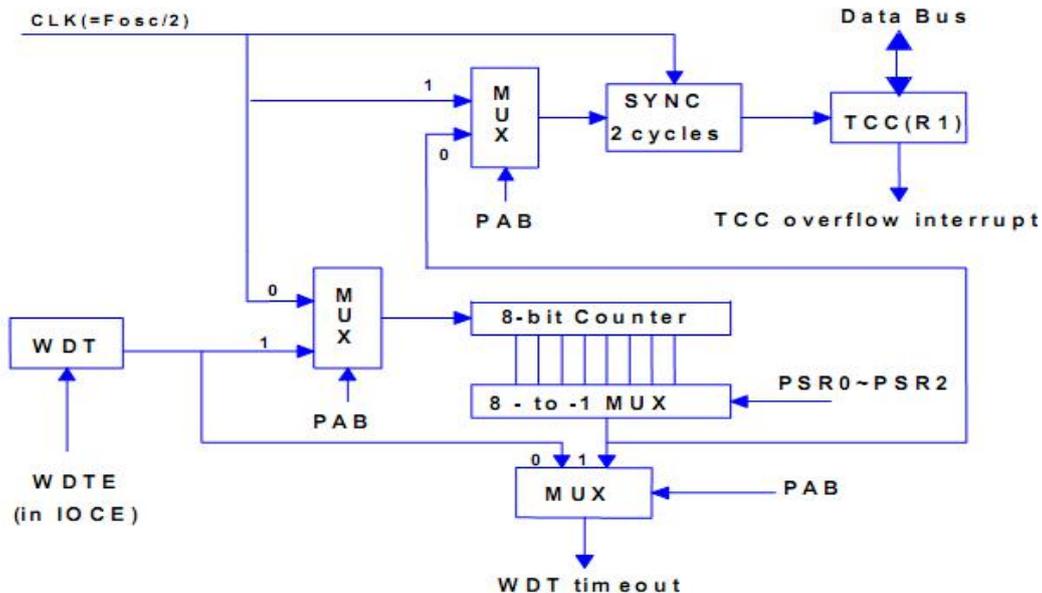
1: the corresponding R is enabled;

0: otherwise.

## 五、功能描述

### 5.1 TCC/WDT 预分频器

有一个 8bit 计数器可以作为 TCC 或 WDT 的分频器。这个预分频器在同一时刻只能提供给 TCC 或 WDT 之一使用，CONT 寄存器的 PAB 位用于决定预分频器通过每次写入 TCC 的指令清除。当分配给 WDT 模式时，WDT 和预分频器由指令 WDTC 和 SLEP 进行清 0。



TCC 和 WDT 的模块图

R1(TCC)是一个 8bit 定时器/计数器。TCC 在每个指令周期(无预分频)加 1。

WDT 是一个自由运行的片内 RC 振荡器。即使是振荡器关闭（例如：在休眠模式），WDT 仍然保持运行。在正常操作或者休眠模式，WDT 溢出（如果有效）将导致复位。在正常模式下，WDT 在任何时候都可以通过软件编程设置为无效或有效（如果代码选项位 ENWDT 为“1”）。参考 IOCE 寄存器的 WDTE 位。没有预分频器时，WDT 的时间输出周期大约是 18MS。

### 5.2 I/O 端口

I/O 寄存器组，端口 5~端口 6，是双向三态 I/O 端口。

P50~P53 通过软件控制可以具有内部下拉。

P54~P57 和 P60~P63 通过软件控制可以具有内部上下拉。

P64~P67 通过软件控制可以具有内部上拉。

P50~P57 和 P60~P67 通过软件控制可以具有唤醒功能。

通过编程控制 I/O 控制寄存器组 (IOC5~IOC6)，I/O 端口可以被定义为“输入”或“输出”管脚。I/O 寄存器组和 I/O 控制寄存器组都是可读可写的。下图描述了 I/O 接口电路。注意在读取 I/O 端口时输入和输出管脚的读取路径是不同的。

注意：VDD=5V，振荡器起振时间为 3.5ms±30%

VDD=3V, 振荡器起振时间为  $3.5\text{ms}\pm 30\%$

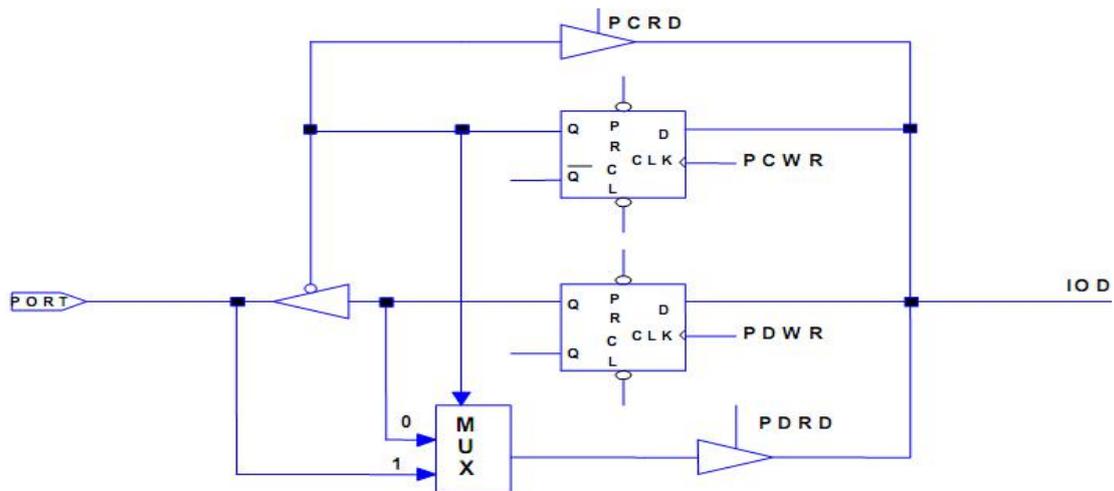


图 7 (a) I/O 端口电路和 I/O 控制寄存器

## 5.3 定时器

### 概述

定时器 1(TMR1)是一个 8 位时钟计数器, 它带一个可编程的分频器。TMR1 可读写, 任何复位情况均清 0。在使用中, 为了降低功耗, 可以通过设置 TMR1EN[T1CON<2>]位为 0 关闭 TMR1。

### 功能描述

**预分频器:** 由 T1P1 和 T1P02 (T1CON<1,0>) 决定分频率为 1:2、1:8、1:16 或 1:32 (是对  $F_{\text{sys}}$  进行预分频)。它可以在写入 TMR1, T1CON 或任何复位的情况下被清 0。

**PWP:** 脉冲宽度预置寄存器。预先写入波特率时钟期望宽度。

**TMR1:** 定时器 1 寄存器, TMR1 加 1 直到它与 PWP 值相等, 然后清为 0。

**比较器:** PWP 和 TMR1 相等改变输出状态, 同时将置 TMR1IF 为 1。

### 可编程相关寄存器组

#### TMR1 的相关状态/数据寄存器

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMR1/RE	TMR17	TMR16	TMR15	TMR14	TMR13	TMR12	TMR11	TMR10
PWP/RF	PWP7	PWP6	PWP5	PWP4	PWP3	PWP2	PWP1	PWP0
T1CON/IOCC	0	0	0	0	0	TM1E	TM1P1	TM1P0

TMR1: 定时器寄存器, TMR17~TMR10 加 1 直到与 PWP 相同, 然后 TMR1 清 0

PWP: 脉冲宽度预置寄存器。PWP7~PWP0 为预先写入波特率时钟期望宽度值

T1CON: TIMER1 控制寄存器

TM1E (BIT2): Timer1 允许位

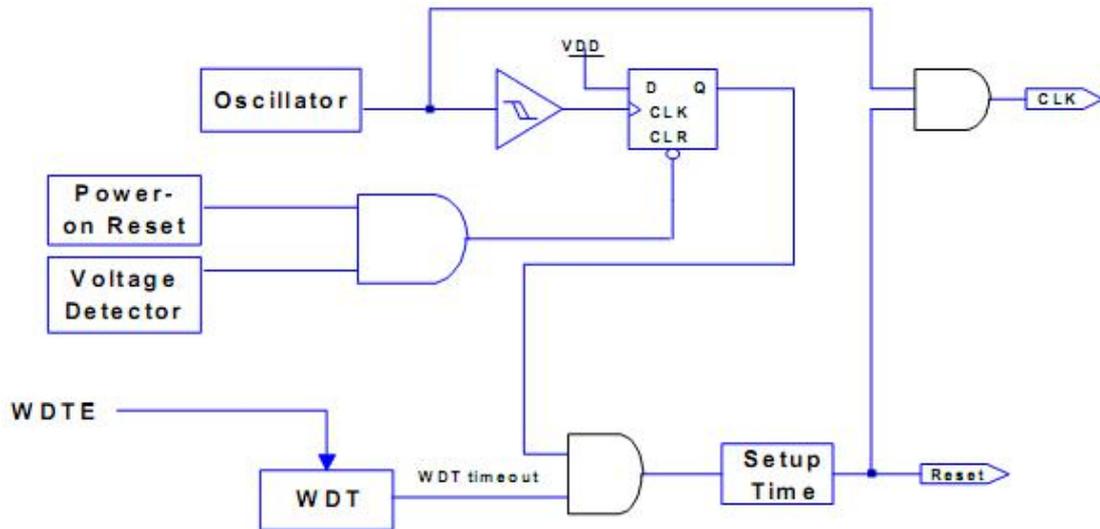
TM1P1 和 TM1P0 (BIT1~0): 为 Timer1 对 F<sub>sys</sub> 的预分频位

TM1P1	TM1P0	Prescaler Rate
0	0	1:2
0	1	1:8
1	0	1:16
1	1	1:32

## 5.4 复位和唤醒

复位的原因有:

- (1) 上电复位
- (2) WDT 溢出 (如果有效)



复位电路框图

在检测到复位信号后, 设备将保持一个大约 3.5ms (复位) 再加在 IRC 时为 300uS 的起振时间, 高速晶振时为 9mS 的起振时间, 低速晶振时为 576mS 的起振时间的复位状态。一旦复位发生, 以下功能将被执行:

- 振荡器正在运行或者开始运行。
- 程序计数器(R2)设置为全“1”。
- 当电源打开, R3 的 5~6 位和 R4 的高 2 位被清 0。
- 所有 I/O 端口的管脚被设置为输入模式 (高阻抗状态)
- WDT 和预分频器清 0。
- 如果代码选项 ENWDT 为“1”, WDT 有效。
- CONT 寄存器的 BIT3、6 清 0, IOCE 寄存器的 BIT0、4~5 设置为“1”。

注意: 振荡器起振时间

IRC 时为 300uS±30%,  
 高速晶振时为 9mS±30%,  
 低速晶振为 576mS±30%

执行 SLEEP 指令（命名为 SLEEP1 模式）可以进入休眠模式（省电模式）。进入休眠模式时，WDT（如果有效）被清除但仍然保持运行。WDT 溢出（如果有效）就唤醒并复位。R3 的 T、P 标志位可判断复位（唤醒）的原因。

#### 复位时状态寄存器 T、P 位状态

以下溢出情况，将会引起复位：1.电源上电。2.WDT 溢出。

表 复位后半 T 和 P 值

Reset Type	T	P
Power on	1	1
WDT during Operation mode	0	P
WDT wake-up during SLEEP1 mode	0	0
Wake-Up on pin change during SLEEP2 mode		

\*p :复位前值

表 不同事件对 T、P 影响

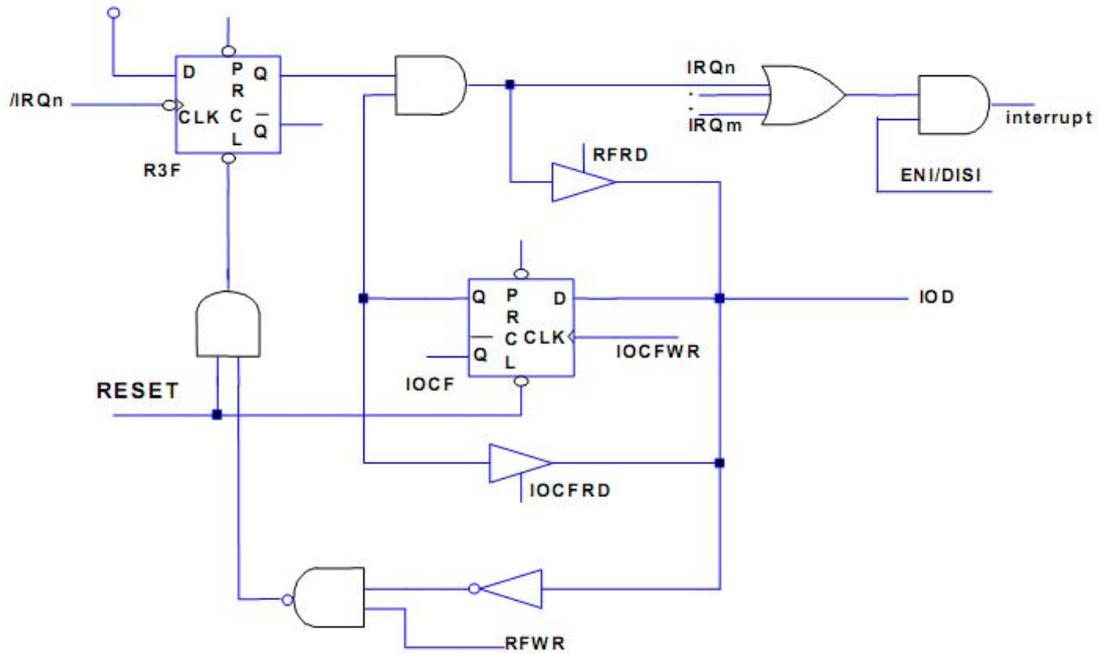
Event	T	P
Power on	1	1
WDTC instruction	1	1
WDT time-out	0	*P
SLEEP instruction	1	0
Wake-Up on pin change during SLEEP2 mode	P	P

## 5.5 中断

TX416 有以下中断：

1. 外部中断(/INT)。
2. TCC 溢出中断。
3. Timer1 比较器匹配中断。
4. IO 口改变中断（从睡眠模式唤醒）
5. SPI 中断。
6. WDT timeout 中断。
7. ADC 轉換完成中断。
8. Low-voltage detection 中断。
9. PTMR1/PWM1 比较器匹配中断。
10. PTMR2/PWM2 比较器匹配中断。
11. I2C 中断。

R3F 是中断状态寄存器，它记录中断标志位的中断请求。IOCF 是中断屏蔽寄存器。全局中断由指令 ENI 设置有效，由 DISI 指令设置为无效。当中断产生时（当允许时），它将使下一条指令从分别从不同的中断入口地址处开始执行。在执行中断服务子程序时，通过查询 R3F 寄存器的标志决定中断源。在离开中断服务子程序之前，应该通过软件清除中断标志位并且使能中断以免重复中断。中断状态寄存器（R3F）中的标志位的置 1 与它的屏蔽位的状态以及 ENI 指令的执行无关。注意，读 R3F 内容是 R3F 与 IOCF 的逻辑与的输出。RETI 指令离开中断子程序并且使全局中断（ENI 指令的执行）有效。



中断输入电路

## 5.6 振荡器

### 1. 振荡模式

TX416 能在五种振荡模式下运行。它们是高频晶体振荡模式 (HX)，陶瓷振荡模式 (XT)，低频晶体振荡模式 (LX)，内部 RC 振荡模式 (IRC)。

### 2. 晶体振荡/陶瓷振荡(XTAL)

如下所示：TX416 能通过 OSCI 引脚输入外部时钟信号，在许多场合，引脚 OSCI 和引脚 OSCO 与晶体或陶瓷振荡器相连。表 5.1 给出了 C1 和 C2 推荐值。因为各个谐振器特性不同，用户应根据其规格选择 C1 和 C2。对低频晶振（32768）必须并接一个 1M 的电阻。

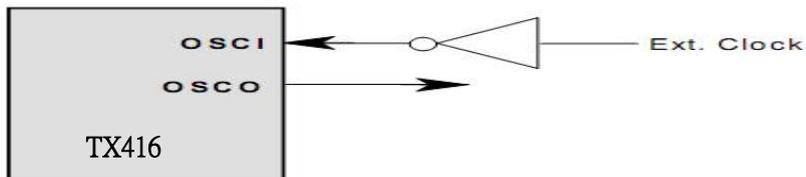


图 外部时钟脉冲输入电路

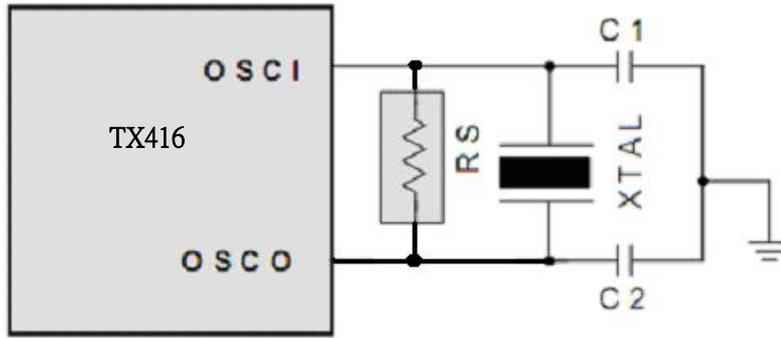


图 晶体/谐振电路图（并联电阻）

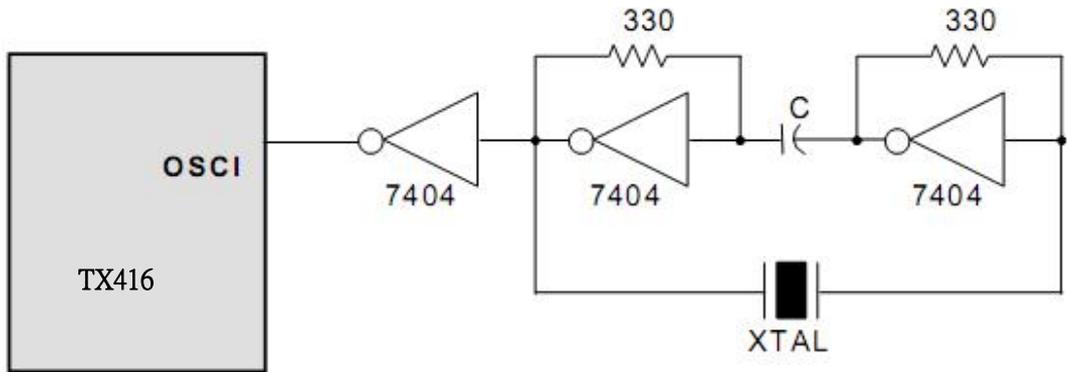


图 晶振/谐振电路图（串联谐振模式）

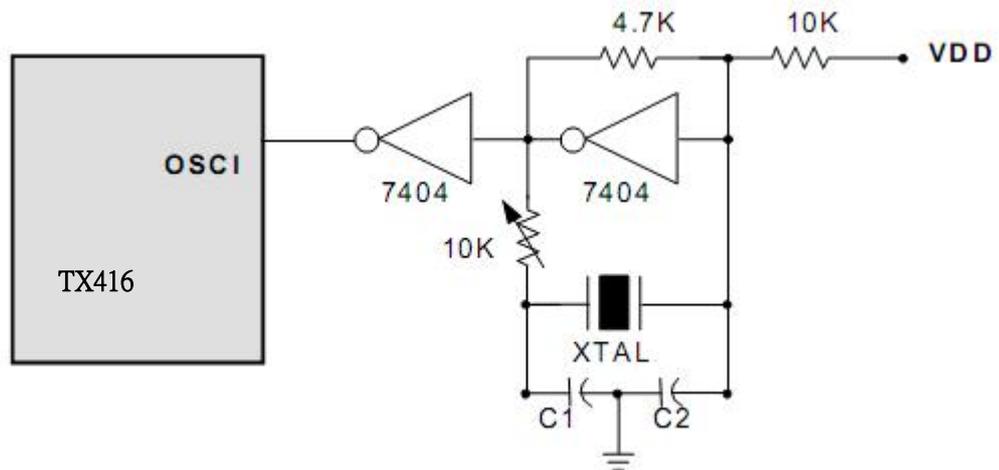


图 晶振/谐振电路图（并联谐振模式）

表 5.1 晶振/谐振电容选择

Oscillator Type	Frequency Mode	Frequency	C1(pF)	C2(pF)
Crystal Oscillator	LXT	32.768KHz	25	15
		100KHz	25	25
		200KHz	25	25
	HXT	455KHz	1000	1000
		1.0MHz	15~30	15~30

		2.0MHz	15	15
		4.0MHz	15	15

### 3. 内部 RC 振荡模式

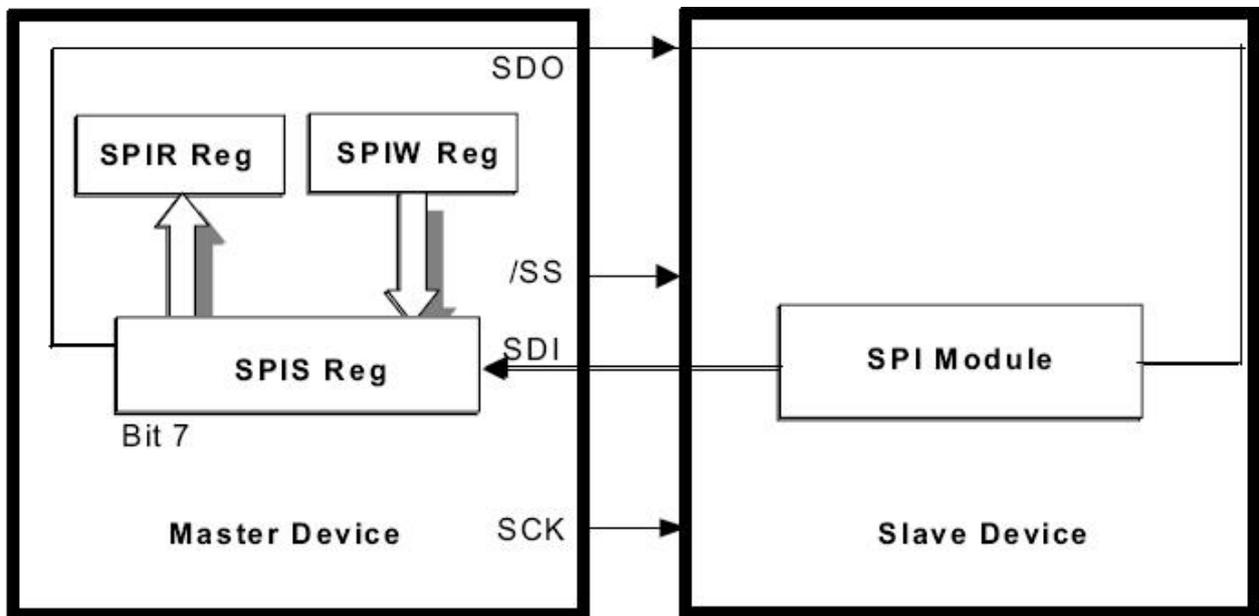
选择内部 RC 振荡模式时，可在烧录时选择 3M、4M、6M、8M、10M、12M、20M 等频率

## 5.7 SPI

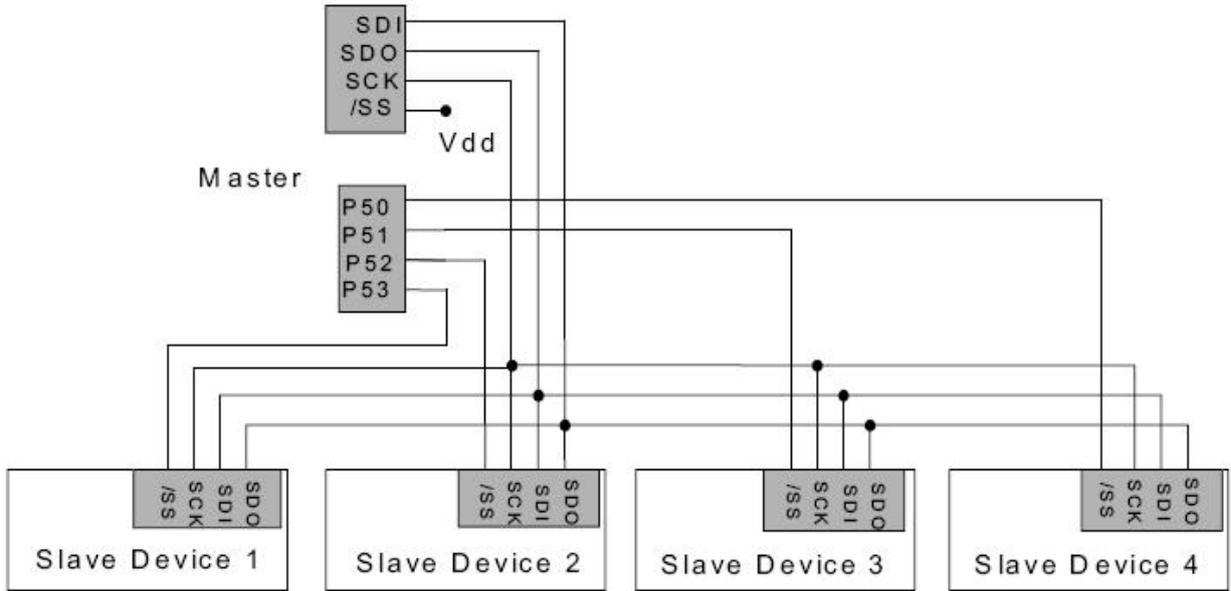
下图描述了 TX416 如何通过 SPI 模块与其它设备进行通信。如果 TX416 是主控制器，它通过 SCK 传送时钟信号。一对 8 位数据在同时进行传送和接收。但是，如果 TX416 被定义为从设备，它的 SCK 就被编程作为一个输入。数据将根据时钟速率和选择边沿连续发送。

特征：

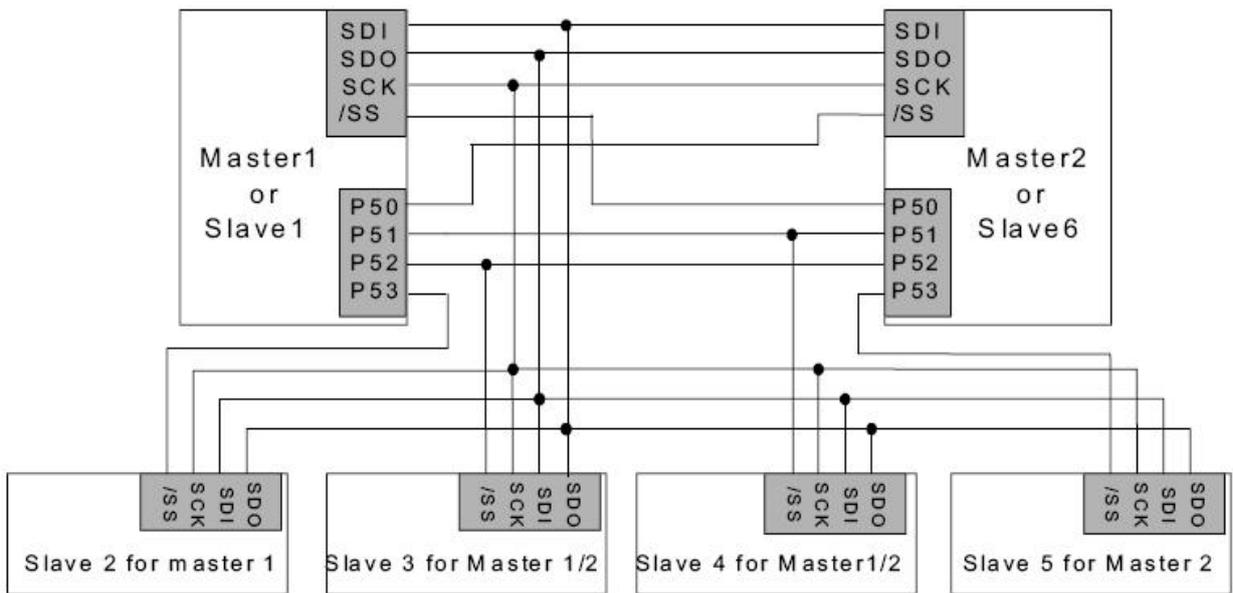
- 1) 可用于主模式或从模式
- 2) 三线或四线同步通信，即全双工。
- 3) 可编程的通信波特率。
- 4) 可编程时钟极性。
- 5) 读缓冲区满，产生中断标志。
- 6) 位传输频率最大达到 8MHz。



SPI 主/从通信



单一主模式和多种从模式的 SPI 配置



多种主模式和多种从模式的 SPI 配置



**P66//SS:** 从选择（选项）。此管脚（/SS）在从模式时可能需要

**RBF:** 由缓冲区满检测器置 1，由软件清 0

**RBFI:** 由缓冲区满检测器置 1，由软件清 0（当开中断时）

**缓冲区满检测器:** 当 8 位移位完成时设置为 1

**SSE:** 将数据装入 SPIS 寄存器，然后开始移位

**SPIS 寄存器:** 字节移出或移入。高位在前。SPIS 寄存器和 SPIW 寄存器同时被装载。一旦数据被写入，SPIS 就开始传送/接收。当 8 位数据移位完成后，接收到的数据将移到 SPIR 寄存器中。RBF（读缓冲区满）标志。

**SPIR 寄存器:** 读缓冲区。此缓冲区拒绝任何写入直到 8 位移位完成。

**SBRS2~SBRS0:** 对时钟频率/速率和时钟源编程。

**时钟选择:** 选择内部时钟或外部时钟作为移位时钟。

**边沿选择:** 通过编程 CES 位选择适当的时钟边沿。

#### 信号和管脚描述

四个管脚 SDI、SDO、SCK 和/SS 的详细描述如下：

##### SDI/P60:

串行数据输入。

串行接收数据。

如果没有被选择，应定义为高阻抗。

主从器件的锁存时钟分频率和时钟边沿应设置为相同。

接收到的字节将更新上一次收到字节。

当 SPI 操作完成时，RBF 位将被置 1。

##### SDO/P61:

串行数据输出。

串行传送数据。

主从器件相同时钟分频率和时钟。

接收到的字节将更新上一次传送字节。

当 SPI 操作完成时，CES 位将被清 0。

##### SCK/P62:

串行时钟。

由主设备产生。

SDI 管脚和 SDO 管脚的数据通信同步。

CES 位用于选择通信边沿。

SBRS0~SBRS2 位用于决定通信波特率。

CES 位、SBRS0~SBRS2 位在从模式下无用。

##### /SS/P66:

从模式选择，低电平有效。

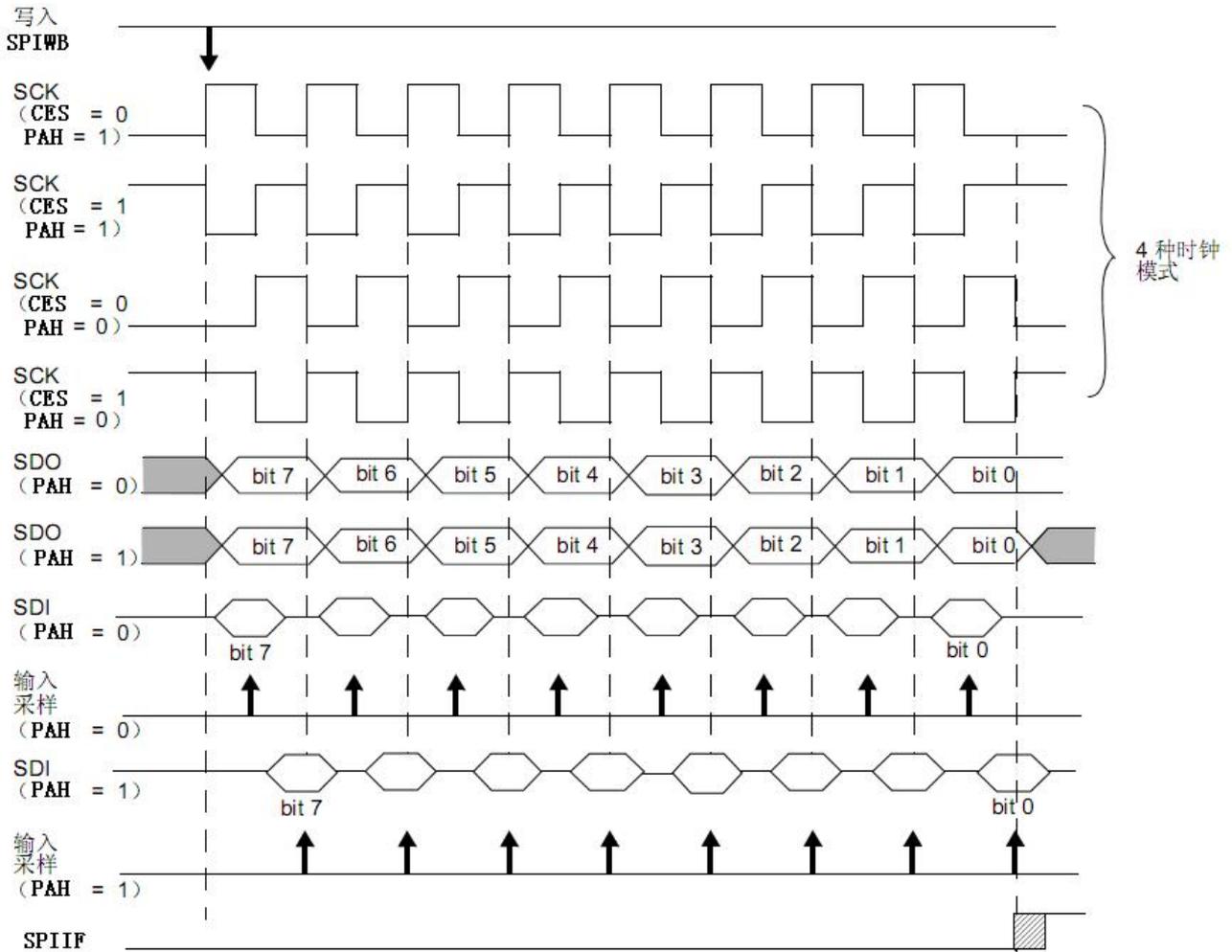
由主设备产生，让从设备接收数据。

在 SCK 的第一个周期之前变为低并且保持到最后一个周期（第 8 个）结束。

当/SS 为高电平时忽略 SDI 和 SDO 管脚的数据，因为 SDO 不被驱动。

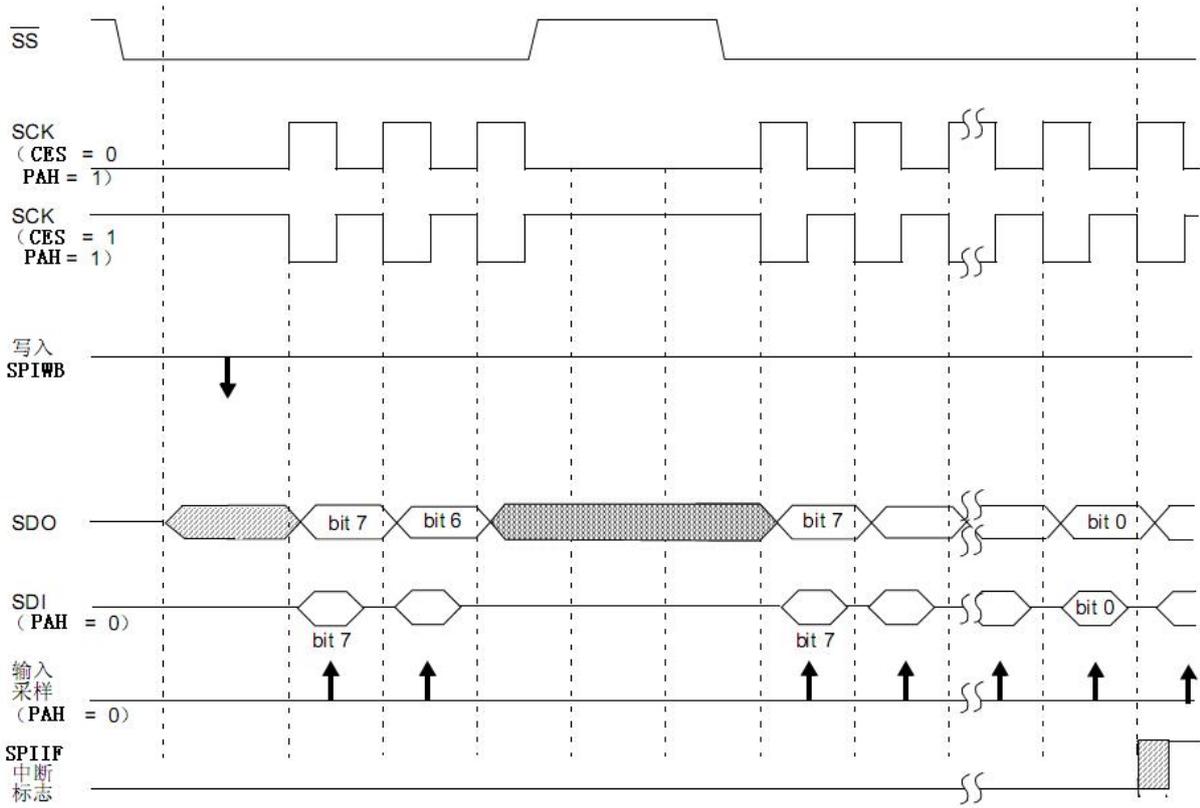
## 5.7.2 SPI 模式的时序图

主控模式：

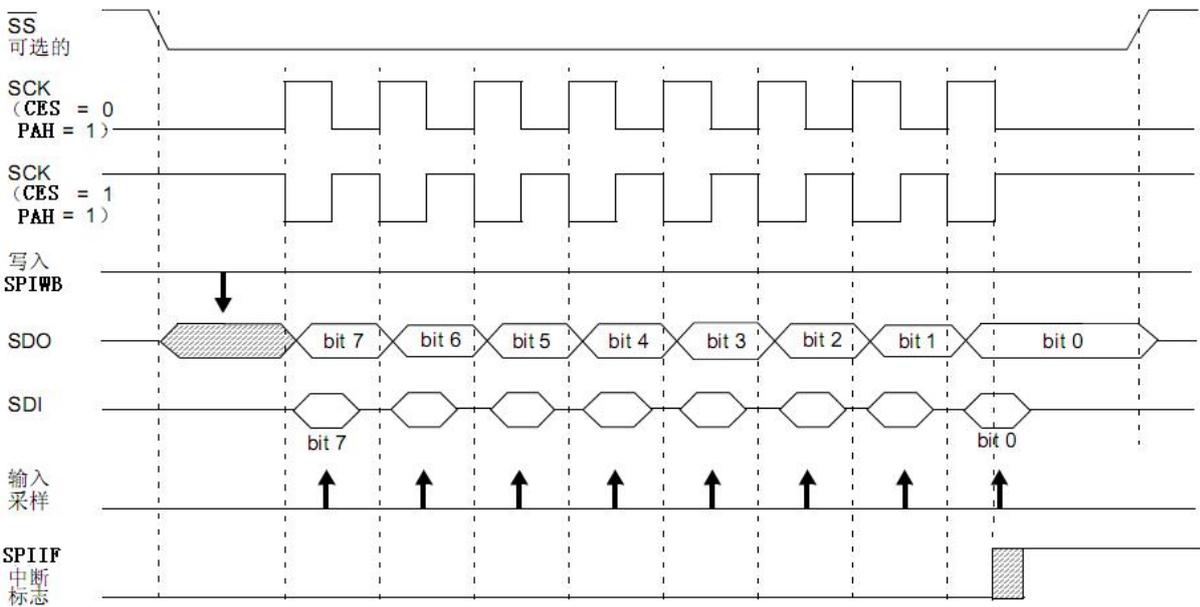


从动模式：

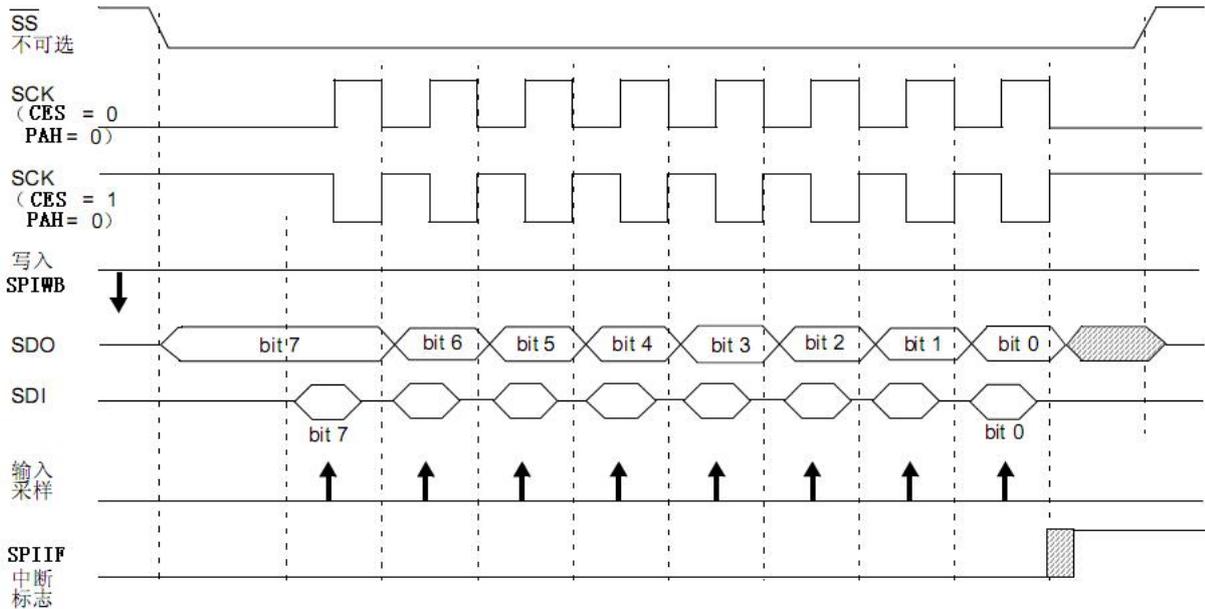
从动模式同步时序图



SPI 从动模式 (PAH=1) 时序图



SPI 从动模式 (PAH=0) 时序图



## 5.8 模数转换器 (A/D) 模块

TX416 的模数 (A/D) 转换器模块有 9 个模拟输入通道。

A/D 转换器将模拟输入信号转换成对应的 10 位数字量。

A/D 转换器的一个新特性是增加了可编程采集时间。该特性使用户可选择新的转换通道并立即置位 ADC。当 ADC 被置位后，在转换实际开始前，选择的通道已开根据设定的采集时间采样。这减少了采集周期所需的固件开销。

该模块有 4 个寄存器：

A/D 转换结果高字节寄存器 (AD\_MSB)

A/D 转换结果低字节寄存器 (AD\_LSB)

A/D 控制寄存器 (ADCON)

A/D 控制寄存器 2 (ADCON2)

ADCON 寄存器用于配置 A/D 模块的端口引脚功能和参考电压源及启动 AD，ADCON2 用于配置可编程采样时间。

每个与 A/D 转换器有关的端口引脚都可以配置成模拟输入或数字 I/O 引脚。AD\_MSB 和 AD\_LSB 寄存器包含 A/D 转换的结果。A/D 转换完成时，结果装入 AD\_MSB/AD\_LSB 寄存器，ADC 位(在 ADCON 中)被清零。

在上电复位时 AD\_MSB: AD\_LSB 寄存器中的值不变，上电复位后 AD\_MSB: AD\_LSB 寄存器中值不确定。当根据需要设置好 A/D 转换模块之后，在开始转换之前必须获得 A/D 转换的输入通道。该模拟输入通道的相应 IOC 位必须被设置为输入。欲确定采样时间，参见“A/D 采集时间要求”。在采样时间到达后，A/D 转换即可开始。采集时间可以设置为介于 ADC 置位到开始转换之前发生。

应按照以下步骤进行 A/D 转换：

1. 设置 A/D 转换模块：

选择 A/D 输入通道 (ADCON, Bit 3~0)

选择 A/D 转换采样时间 (ADCON2, Bit 2~0)

采样频率最佳为：1M ~ 1.5M Hz

打开 A/D 模块时钟 (GCKCON, Bit 4)

打开 A/D 转换模块和 Analog Input Channel (ADCON, Bit 6、5)

## 2、启动转换：

将 ADC 位置 1 (ADCON, Bit 7)。

## 3、等待 A/D 转换完成，可以通过以下方法来判断：

软件查询 ADC 位的状态是否为 0。

## 4、读 A/D 结果寄存器 (AD\_MSB: AD\_LSB)。

## 5、要进行下一次 A/D 转换，根据要求转入步骤 1 或步骤 2。

## 5.9 I2C 模块

I2C 模式下能实现全部从动功能，且硬件支持启动位和停止位中断，以便于固件实现主控功能。I2C 模式实现标准模式规范以及 7 位寻址。

有两个引脚用于数据传输：SCL 引脚作为时钟线，而 SDA 引脚作为数据线。

通过将 I2C 使能位 I2C\_E (I2CCON<7>) 置 1 以使能 I2C 模块的功能。

SSP 模块有 5 个寄存器用于 I2C 操作，这 5 个寄存器是：

- I2C 控制寄存器 (I2CCON)
- I2C 状态寄存器 (I2CSTA)
- 串行接收 /发送缓冲器 (I2CDAT)
- I2C 地址寄存器 (I2CSAD)
- I2C 端口选择寄存器 (IO\_FUNC) 选择在哪个口上有 I2C 功能

I2CCON 寄存器用于控制 I2C 的工作。可通过设置 I2CCON 的 BIT4 来选择是主模式还是从模式。

### 5.9.1 从动模式

在从动模式下，SCL 引脚和 SDA 引脚必须被配置为输入。当地址匹配或在地址匹配后发送的数据被接收时，硬件会自动产生一个应答 (ACK) 脉冲，并把当时接收到的值装入 I2CDAT 寄存器。

某些条件会使 I2C 模块不发出此 ACK 脉冲。这些条件包括 (之一或全部)：

a) 在接收到数据前，缓冲器满标志位 RX\_F(I2CSTA<6>) 置 1。

b) 在接收到数据前，溢出标志位 FULL (I2CSTA<0>) 置 1。

#### 5.9.1.1 寻址

一旦 I2C 模块被使能，它就会等待启动条件发生。启动条件发生后，8 位数据被移入 I2CDAT 寄存器。在时钟 (SCL) 线的上升沿采样所有的输入位。在第 8 个时钟 (SCL) 脉冲的下降沿寄存器 SSPSR<7:1> 的值会和 I2CSAD 地址寄存器的值比较。如果地址匹配，并且 RX\_F 和 FULL 都被清零，会发生下列事件：

a) 接收寄存器的值被装入 I2CDAT 寄存器。

b) 缓冲器满标志位 RX\_F 被置 1。

- c) 产生 ACK 脉冲。
- d) 在第 9 个 SCL 脉冲的下降沿，I2C 中断标志位被置 1（如果允许中断，则产生中断）。

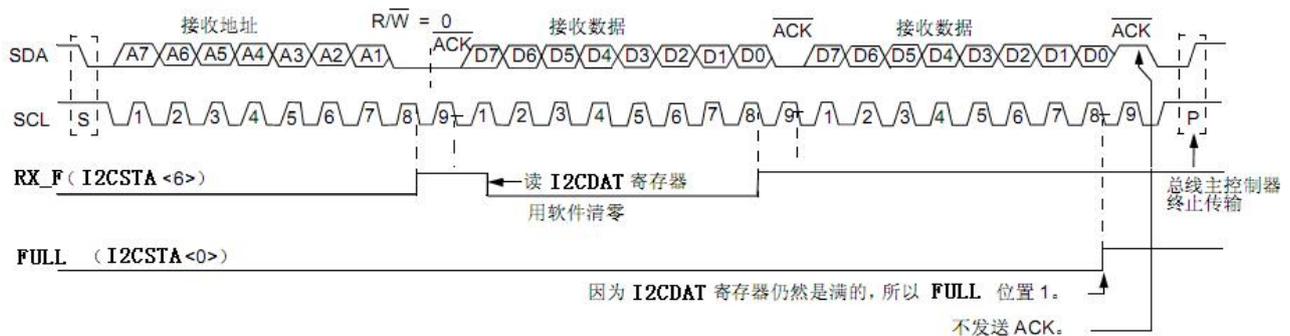
### 5.9.1.2 接收

当地址字节的 R/W 位清零并发生地址匹配时，I2CSTA 寄存器中的 R/W 位清零。接收到的地址被装入 I2CDAT 寄存器。

当发生地址字节溢出时，则不会产生应答脉冲（ACK）。溢出条件是指 FULL 位（I2CSTA<0>）置 1，或者 RX\_F 位（I2CSTA<6>）置 1。这是一个由于用户固件导致的错误状态。

每个数据传输字节都会产生 I2C 中断。标志位 RX\_F 必须用软件清零。通过 I2CSTA 寄存器可以确定该字节的状态。

#### I2C™ 接收时序（7 位地址）

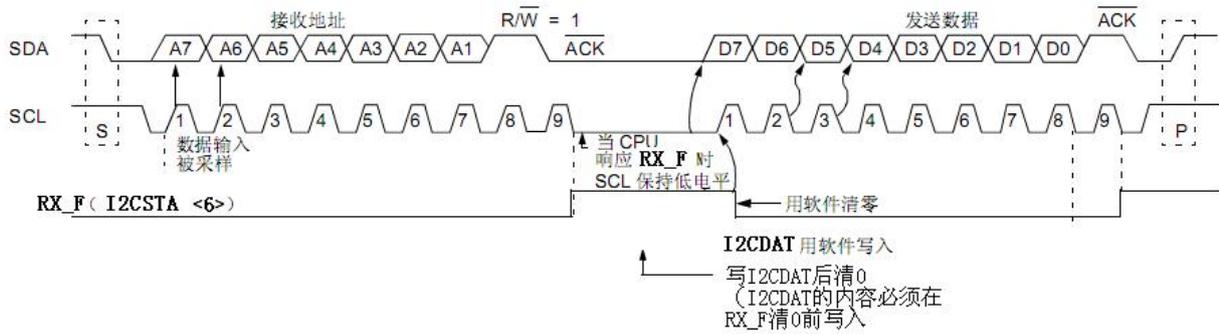


### 5.9.1.3 发送

当输入地址字节的 R/W 位置 1 并发生地址匹配时，I2CSTA 寄存器的 R/W 位被置 1。接收到的地址被装入 I2CDAT 寄存器。ACK 脉冲在第 9 位上发送，SCL 引脚保持低电平。发送数据必须被装入 I2CDAT 寄存器。然后，应该通过将 RX\_F 位（I2CSTA<6>）清 0 来使能 SCL 引脚。主控制器必须在发出另一个时钟脉冲前监视 SCL 引脚。从控制器可以通过延长时钟低电平时间不与主控制器同步。8 个数据位在 SCL 输入的下降沿被移出。这可以确保在 SCL 为高电平期间 SDA 信号是有效的。

每个数据传输字节都会产生 I2C 中断。标志位 TX\_F 必须用软件清零，I2CSTA 寄存器用于确定字节的状态。标志位 TX\_F 在第 9 个时钟脉冲的下降沿被置 1。对于从发送器，来自主接收器的 ACK 脉冲将在第 9 个 SCL 输入脉冲的上升沿被锁存。若 SDA 线为高电平（无 ACK 应答信号），则表示数据传输已完成。在这种情况下，如果从控制器锁存了 ACK，将复位从动逻辑（复位 I2CSTA 寄存器），同时从控制器监视下一个启动位的出现。如果 SDA 线为低电平（ACK），则必须将下一个要发送的数据装入 I2CDAT 寄存器。然后，通过将 TX\_F 位（I2CSTA<4>）清 0 使能 SCL 引脚。

#### I2C™ 发送时序（7 位地址）



## 5.9.2 主控模式

主控模式通过固件在检测到启动条件和停止条件时产生中断来工作。停止（P）位和启动（S）位在复位时或禁止 I2C 模块时清零。停止（P）位和启动（S）位会根据启动和停止条件翻转。当 P 位置 1 时，可以获得 I2C 总线的控制权；否则，P 位和 S 位都清零，总线处于空闲状态。

下列事件会使 I2C 中断标志位置 1（如果允许 I2C 中断，则产生中断）：

- 启动条件
- 停止条件
- 发送 / 接收到数据传输字节

当同时使能主控模式和从动模式时，需要使用软件区分中断源。

# 六、绝对最大范围

(所有电压参考 GND)

项目	符号	额定值	单位
耐压范围	$V_{DD}$	0 ~ 6.5	V
输入/输出电压	$V_I / V_O$	GND-0.3~VDD+0.3	V
工作温度	$T_{DD}$	-40 ~ 85	°C
存放温度	$T_{ST}$	-50 ~ 125	°C

# 七、电气特性

交流电气特性 ( $T_A=0^{\circ}\text{C} \sim 70^{\circ}\text{C}$ ,  $V_{DD}=3\text{V} \& 5\text{V}$ )

符号	参数	条件	最小值	典型值	最大值	单位
Dclk	振荡输入占空比	-	45	50	55	%
Tins	指令周期		100	-	DC	nS

Ttcc	TCC 输入周期	-	$(T_{ins}+20)/N^*$	-	-	nS
Tdrh	复位保持时间		3.5-30%	3.5	3.5+30%	mS
Trst	复位脚的脉冲宽度		2000	-	-	nS
Twdt	看门狗周期时间		17.6-30%	17.6	17.6+30%	mS
Thold	输入电平保持时间			20		nS
Tdelay	输出延时时间	20pF 的电容负载		50		nS

注意：N=预分器的倍数

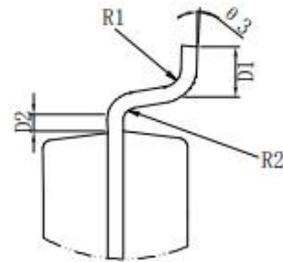
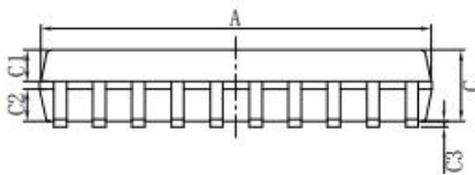
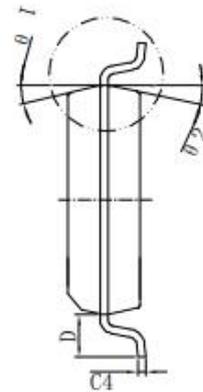
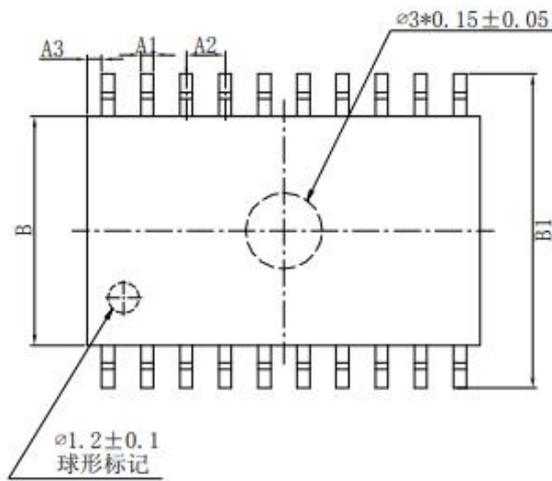
### 直流电气特性 ( $T_A=0^{\circ}\text{C} \sim 70^{\circ}\text{C}, V_{DD}=3\text{V} \& 5\text{V}, \text{LDO} = 3.0\text{V}$ )

符号	参数	条件	最小值	典型值	最大值	单位
FXT	晶振 VDD=2.3V		DC	-	4.0	MHz
	晶振 VDD=3.0V		DC	-	8.0	MHz
	晶振 VDD=5.0V		DC	-	20.0	MHz
IIL	输入漏电流	VIN=VSS、VDD	-	0	±1	uA
VIH1	输入高电平 (VDD=5V)	Ports 5、6	2.6	-	-	V
VIL1	输入低电平 (VDD=5V)	Ports 5、6	-	-	1.4	V
VIH2	输入高电平 (VDD=3V)	Ports 5、6	1.8	-	-	V
VIL2	输入低电平 (VDD=3V)	Ports 5、6	-	-	0.9	V
VOH	输出高电平 (VDD=3V)	IO_BUF=0, IOH=6mA	2.7	-	-	V
		IO_BUF=1, IOH=6mA	2.7	-	-	V
	输出高电平 (VDD=5V)	IO_BUF=0, IOH=10mA	4.5	-	-	V
		IO_BUF=1, IOH=10mA	4.5	-	-	V
VOL	输出低电平 (VDD=3V)	IO_SINK=0, IO_BUF=0, IOL=7mA	-	-	0.3	V
		IO_SINK=0, IO_BUF=1, IOL=7mA	-	-	0.3	V
	输出低电平 (VDD=5V)	IO_SINK=0, IO_BUF=0, IOL=14mA	-	-	0.5	V
		IO_SINK=0, IO_BUF=1, IOL=15mA	-	-	0.5	V
	输出低电平 (VDD=3V)	IO_SINK=1, IOL=14mA	-	-	0.3	V
	输出低电平 (VDD=5V)	IO_SINK=1, IOL=30mA	-	-	0.5	V
IPH	上拉电阻电流	上拉电阻打开, 输入接 GND	50	118	240	uA
IPD	下拉电阻电流	下拉电阻打开, 输入接 VDD	20	31	120	uA
ISB1	睡眠电流	输入接 VDD, 输出悬空, WDT=DIS	-	-	3	uA
ISB2	睡眠电流	输入接 VDD, 输出悬空, WDT=EN	-	-	6	uA
ICC1	工作电流(VDD=3.0V)	Fosc=32768Hz, 输出悬空, WDT=DIS	-	34	50	uA
ICC2	工作电流(VDD=3.0V)	Fosc=32768Hz, 输出悬空, WDT=EN	-	41	55	uA
ICC3	工作电流(VDD=5.0V)	Fosc=4MHz, 输出悬空	-	1.1	2.0	mA
ICC4	工作电流(VDD=5.0V)	Fosc=10MHz, 输出悬空	-		4.0	mA

# 八. 封装尺寸图

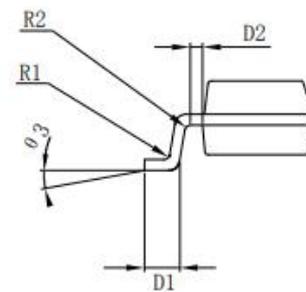
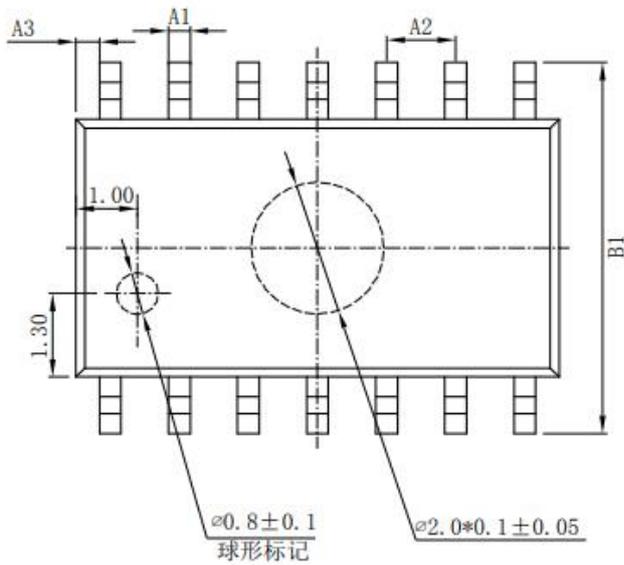
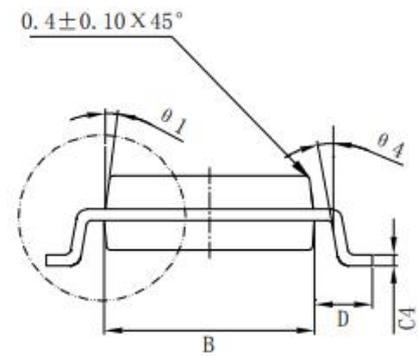
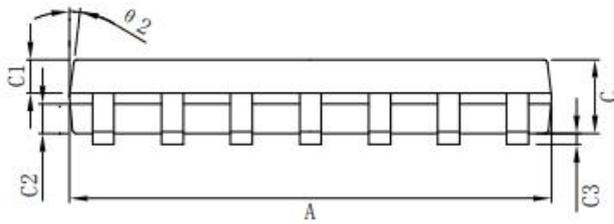
## SOP20

标注	尺寸	最小 (mm)	最大 (mm)	标注	尺寸	最小 (mm)	最大 (mm)
A		12.60	12.90	C4		0.246	0.262
A1		0.381	0.431	D		1.353	1.453
A2		1.24	1.30	D1		0.764	0.964
A3		0.45	0.46	D2		0.18TYP	
B		7.40	7.60	R1		0.30TYP	
B1		10.206	10.406	R2		0.20TYP	
C		2.15	2.30	$\theta 1$		12° TYP4	
C1		0.938	1.038	$\theta 2$		12° TYP4	
C2		0.938	1.038	$\theta 3$		0° ~ 8°	
C3		0.145	0.205				



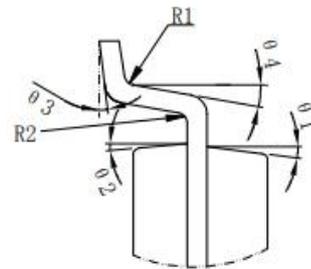
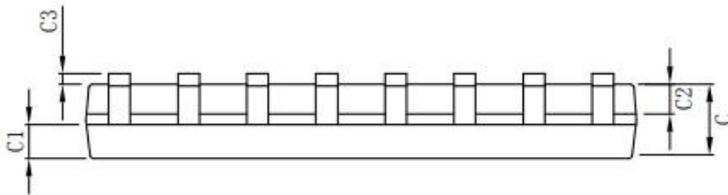
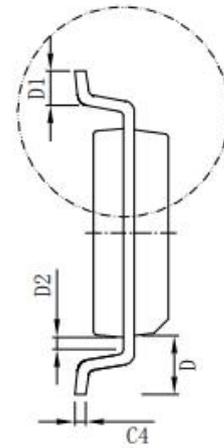
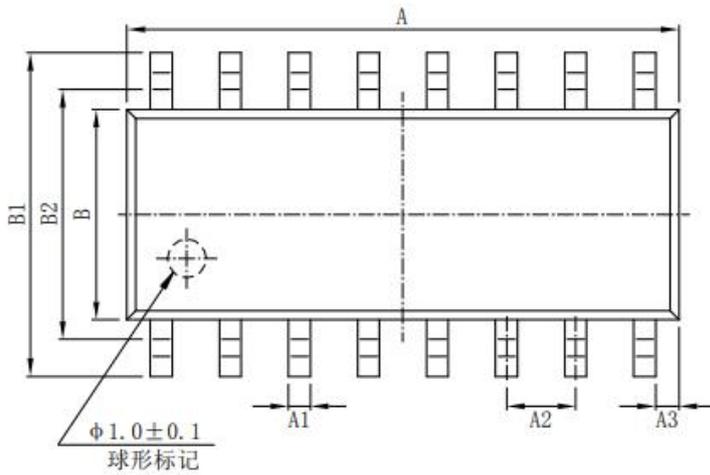
## SOP14

标注	尺寸	最小(mm)	最大(mm)	标注	尺寸	最小(mm)	最大(mm)
A		8.55	8.75	C4		0.193	0.213
A1		0.356	0.456	D		0.95	1.15
A2		1.27TYP		D1		0.40	0.70
A3		0.312TYP		D2		0.20TYP	
B		3.80	4.00	R1		0.20TYP	
B1		5.80	6.20	R2		0.20TYP	
C		1.40	1.60	$\theta 1$		8° ~ 12° TYP4	
C1		0.60	0.70	$\theta 2$		8° ~ 12° TYP4	
C2		0.55	0.65	$\theta 3$		0° ~ 8°	
C3		0.05	0.25	$\theta 4$		4° ~ 12°	



## SOP16

标注	尺寸	最小(mm)	最大(mm)	标注	尺寸	最小(mm)	最大(mm)
A		9.80	10.00	C4		0.203	0.233
A1		0.356	0.456	D		1.05TYP	
A2		1.27TYP		D1		0.40	0.70
A3		0.302TYP		D2		0.15	0.25
B		3.85	3.95	R1		0.20TYP	
B1		5.84	6.24	R2		0.20TYP	
B2		5.00TYP		$\theta 1$		8° ~ 12° TYP4	
C		1.40	1.60	$\theta 2$		8° ~ 12° TYP4	
C1		0.61	0.71	$\theta 3$		0° ~ 8°	
C2		0.54	0.64	$\theta 4$		4° ~ 12°	
C3		0.05	0.25				



**注意:**

1. 以上信息如有更新，将不另做通知，请用户在使用前先确定手中的资料是否为最新版本。
2. 对于错误或不恰当操作所导致的后果，我们将不承担责任。

**附：版本记录**

版别	日期	更新内容	页次	备注
<b>V1.00</b>	2010-10-08	正式版本		
<b>V1.01</b>	2010-10-25	修正 P11 与 P35 的错误		
<b>V1.02</b>	2010-11-4	修正 ADCON2, BIT6 与 VREFCON, BIT7 的说明	P30, P34	
V1.03	2010-11-26	修正错误		
V1.04	2011-1-5	修改 P64 只能做输出用，不能做输入用		
V1.1	2011-1-12	修正 ADCON2.7 的描述	P30	
V1.2	2011-2-17	修改适用温度范围		
V1.31	2011-7-7	增加封装尺寸图		
V1.32	2011-7-20	修改为 C 版规格，有将 P64 改为双向口，修改 REFCON 寄存器及上电复位时间与起振时间。		
V1.33	2011-8-26	修改为 D 版规格，将上电复位时间修改为 3.5ms。		