

TX-I-02X

DATASHEET

TX-I-02X	1
DATASHEET.....	1
1. 特性	6
1.1. 程序存储器.....	7
1.2. 脚位图	9
2. SFR	12
2.1. 地址映射	12
2.1.1. SFR, BANK0	12
2.1.2. SFR, BANK1	13
2.1.3. TMR0, 地址 0x01.....	14
2.1.4. STATUS 寄存器, 地址 0x03, 0x83.....	14
2.1.5. PORTA 寄存器, 地址 0x05.....	16
2.1.6. PORTC 寄存器, 地址 0x07	16
2.1.7. INTCON 寄存器, 地址 0x0B	16
2.1.8. PIR1 寄存器, 地址 0x0C	17
2.1.9. TMR2 寄存器, 地址 0x11	18
2.1.10. T2CON 寄存器, 地址 0x12	18
2.1.11. WDTCON 寄存器, 地址 0x18	19
2.1.12. CMCON0 寄存器, 地址 0x19	20
2.1.13. MSCKCON 寄存器, 地址 0x1B.....	21
2.1.14. SOSCPR 寄存器, 地址 0x1C, 1D.....	22
2.1.15. PR0 寄存器, 地址 0x1A	23
2.1.16. OPTION 寄存器, 地址 0x81.....	23
2.1.17. TRISA 寄存器, 地址 0x85	24
2.1.18. TRISC 寄存器, 地址 0x87	24
2.1.19. PIE1 寄存器, 地址 0x8C	25
2.1.20. OSCCON 寄存器, 地址 0x8F.....	25
2.1.21. PR2 寄存器, 地址 0x92.....	26
2.1.22. WPUA 寄存器, 地址 0x95.....	27
2.1.23. IOCA 寄存器, 地址 0x96.....	27
2.1.24. VRCON 寄存器, 地址 0x99.....	27
2.1.25. PCON 寄存器, 地址 0x8E.....	28
2.1.26. EEDAT 寄存器, 地址 0x9A.....	28
2.1.27. EEADR 寄存器, 地址 0x9B.....	29
2.1.28. EECON1 寄存器, 地址 0x9C	29
2.1.29. EECON2 寄存器, 地址 0x9D	30
2.1.30. 配置寄存器 UCFGx	30

2.1.31. PCL 和 PCLATH.....	32
2.1.32. INDF 和 FSR 寄存器	32
3. 系统时钟源	33
3.1. 时钟源模式.....	33
3.2. 外部时钟模式	34
3.2.1. 振荡器起振定时器 (OST)	34
3.2.2. EC 模式.....	34
3.2.3. LP 和 XT 模式	34
3.3. 内部时钟模式	34
3.3.1. 频率选择位 (IRCF)	35
3.3.2. HFINTOSC 和 LFINTOSC 时钟切换时序	35
3.4. 时钟切换	36
3.4.1. 系统时钟选择 (SCS) 位	36
3.4.2. 振荡器起振超时状态 (OSTS) 位	36
3.5. 双速时钟启动模式	36
3.5.1. 双速启动模式配置	37
3.5.2. 双速启动顺序	37
3.6. 故障保护时钟监控器	37
3.6.1. 故障保护检测	38
3.6.2. 故障保护操作	38
3.6.3. 故障保护条件清除	38
3.6.4. 复位或从休眠中唤醒	39
4. 复位时序	40
4.1. POR 上电复位.....	41
4.2. 外部复位 MCLR	41
4.3. PWRT (上电计时器)	42
4.4. BOR(LVR)低电压复位	42
4.5. 错误指令复位	43
4.6. 超时动作	43
5. BOOT	46
6. 烧录和控制串口	47
7. 看门狗定时器.....	48
7.1. 看门狗	48
8. 定时器 0.....	49
8.1. TIMER 0	49
8.2. TIMER 0 定时器模式.....	49
8.3. TIMER 0 计数器模式.....	50

8.3.1.	软件可配置预分频电路	50
8.3.2.	定时器 0 中断	51
8.3.3.	用外部时钟驱动定时器 0	51
9.	定时器 2.....	52
9.1.	TIMER2 工作原理	52
10.	比较器.....	54
10.1.	比较器概述	54
10.1.1.	模拟输入端的连接	56
10.2.	比较器配置	57
10.3.	比较器控制	59
10.3.1.	模拟输出状态	59
10.3.2.	模拟输出极性	59
10.3.3.	模拟输入开关	59
10.4.	比较器反应时间	59
10.5.	比较器中断	60
10.6.	比较器在睡眠状态下的工作	61
10.7.	比较器在复位状态下的工作	61
10.8.	比较器的参考电压	62
11.	数据 EEPROM.....	64
11.1.	编程数据 EEPROM 步骤	64
11.2.	读数据 EEPROM	65
12.	时钟测量	66
12.1.	慢时钟测量模式	66
13.	中断模式	68
13.1.	INT 中断	68
13.2.	PORTA 电平变化中断	69
13.3.	中断响应	69
13.4.	中断过程中的现场保存	70
14.	睡眠省电模式.....	71
14.1.	唤醒模式	71
14.2.	看门狗唤醒	71
15.	I/O 端口	72
15.1.	PORTA 端口和 TRISA 寄存器	72
15.2.	端口的其他功能	72
15.2.1.	弱上拉	72

15.2.2.	状态变化中断.....	72
15.2.3.	端口描述.....	74
16.	芯片的电气特性	86
16.1.	内置高频振荡器(INTERNAL HIGH FREQUENCY OSC).....	86
16.2.	内置低频振荡器(INTERNAL LOW FREQUENCY OSC).....	86
16.3.	带隙基准源电路(BANDGAP).....	87
16.4.	低压差线性稳压器(LDO).....	87
16.5.	低电压侦测/复位电路 (LVD/LVR)	87
16.6.	上电复位电路 (POR)	88
16.7.	I/O PAD 电路.....	88
16.8.	COMPARATOR 比较器电路.....	88
16.9.	4BIT DAC 电路 (比较器参考 CVREF)	88
16.10.	总体工作电流 (IVDD)	89
17.	指令集列表.....	90
18.	芯片封装信息	92
19.	附录 1, 文档更改历史	100

1. 特性

- PIC16-like 指令集
- 8 层硬件堆栈 x11bit
- 2T 或 4T 指令周期
- 2Kx14b 程序存储空间 (16bytes/page)
- 256x8b 数据 EEPROM (16bytes/page)
- 数据 EEPROM 在应用编程
- 128x8b SRAM
- 1 x 带 8 位预分频的定时器 0
- 1 x 带 8 位预分频的定时器 2
- 带 7 位预分频的 WDT，溢出频率约为 16ms~2048ms
- 上电延迟计数器 PWRT
- 低功耗模式 SLEEP
- 多个唤醒源，INT、端口变化中断、WDT、数据 EEPROM 写完成，等等
- 内置高速 16M RC 振荡器
- 内置低速 32K RC 振荡器
- 支持外部晶振 16M 或 32K，以及外部时钟模式
 - ✧ 时钟缺失检测
 - ✧ 双速启动模式（晶体或外部时钟模式下）
- 内置 2 个高速高精度比较器
 - ✧ 可编程的片上参考电压
 - ✧ 比较结果可直接输出
- 最多 16 个通用 IO，18 根芯片管脚
- 7 个 IO 带独立上拉控制
- 端口变化中断，RA0~RA7
- 支持在系统编程 ICSP
- 支持在线调试
- 程序空间保护
- 工作电压范围：1.8V~ 5.5V
- 最大时钟工作频率：16MHz
 - ✧ F_{SYS}=8MHz: 1.8V~2.7V
 - ✧ F_{SYS}=16MHz: 2.7V~5.5V
- 封装类型：SOP14，SOP18

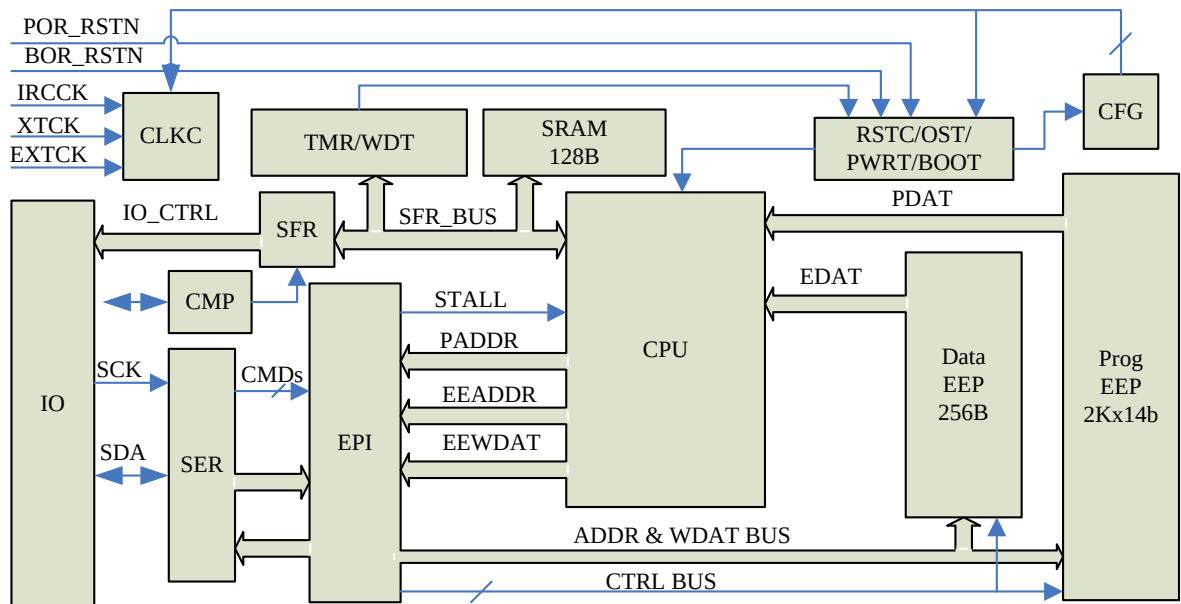


图 1.1 系统功能框图

1. 1. 程序存储器

地址寄存器为 13 位 (0x0000 ~ 0x1FFF)，最多支持 8K 地址空间。程序存储器一共 2Kx14b (0x0000 ~ 0x07FF)，加上 4 个额外的用户配置、工厂配置区，共有 64x14b，它们由 EEPROM 构成。一个 PAGE 是 16 个字，一共有 132 个 PAGE。

其中 0~0x07FF 对主程序区访问，其中未实现部分 0x800~0x1FFF 保留。用户和工厂配置信息区从 0x2000 开始，到 0x203F 结束。

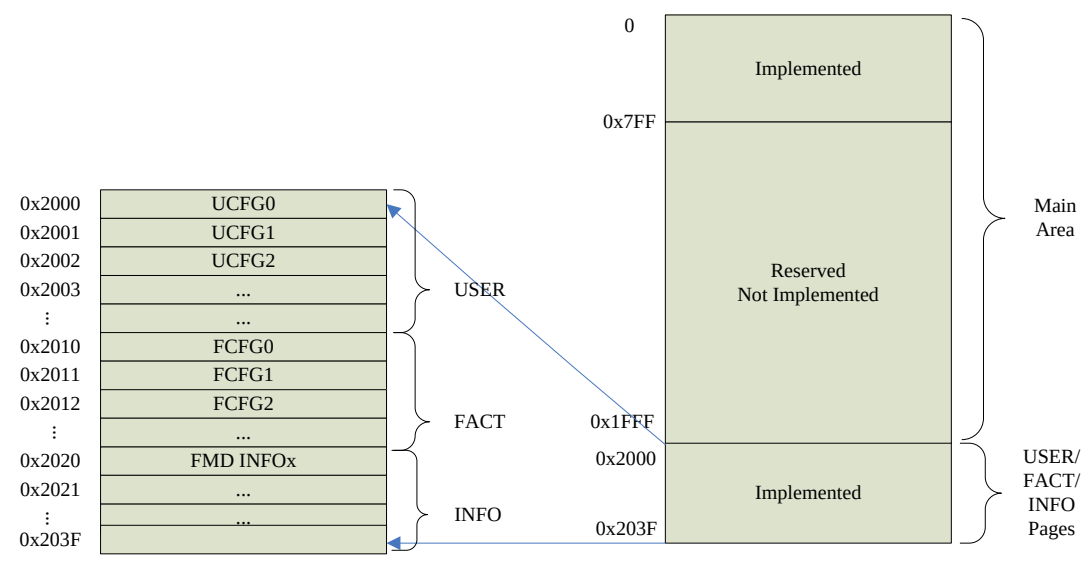


图 1.2 程序空间地址映射

1. 2. 脚位图

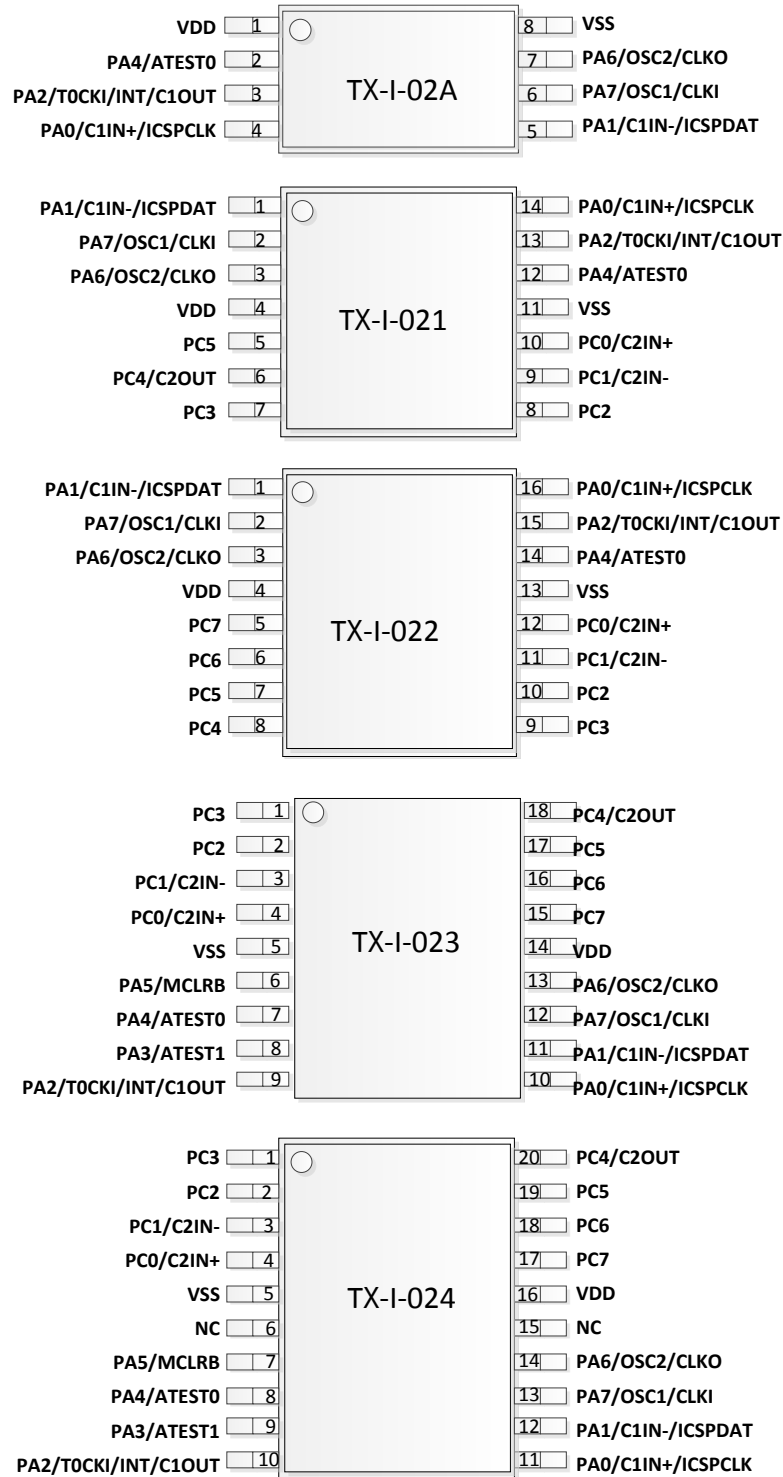


图 1.3 TX-I-02x 所规划 8PIN，14PIN，16PIN，18PIN，20PIN 脚位图

以下为芯片管脚的详细描述：

管脚名	功能名	输入信号 类型	输出信号 类型	具体描述	上下拉
PA0/C1IN+/ICSPCLK	PA0	TTL	CMOS	GPIO with IOC and WPU	可配置上拉
	ICSPCLK	ST	---	Debug/烧录模式串口 clock 信号 (Fmax=6MHz)	
	C1IN+	AN	---	Comparator1 non-inverting input	
PA1/ICSPDAT/C1IN-	PA1	TTL	CMOS	GPIO with IOC and WPU	可配置上拉
	ICSPDAT	TTL	CMOS	Debug/烧录模式串口 data 信号 (Fmax=6MHz)	
	C1IN-	AN	---	Comparator1 inverting input	
PA2/T0CKI/INT/C1OUT	PA2	TTL	CMOS	GPIO with IOC and WPU	可配置上拉
	T0CKI	TTL	---	Timer 0 源头时钟输入 (Fmax=4MHz)	
	INT	TTL	---	外部中断输入	
	C1OUT	---	CMOS	Comparator1 output	
PA3/ATEST1	PA3	TTL	CMOS	GPIO with IOC and WPU	可配置上拉
	ATEST1	TTL	---	模拟测试管脚 1	内部测试用
PA4/ATEST0	PA4	TTL	CMOS	GPIO with IOC and WPU	可配置上拉
	ATEST0	---	---	模拟测试管脚	内部测试用
PA5/MCLR B	PA5	TTL	---	Input only with IOC	可配置上拉
	MCLR B	TTL	---	外部复位输入	MCLR B
PA6/OSC2/CLKO	PA6	TTL	CMOS	GPIO with IOC and WPU	可配置上拉
	CLKO	---	CMOS	测试时钟输出 (Fmax=10MHz)	CLKO
	OSC2	---	XTAL	Crystal/Resonator	OSC2
PA7/OSC1/CLKI	PA7	TTL	CMOS	GPIO with IOC and WPU	可配置上拉
	CLKI	ST	---	External clock input/RC oscillator connection	
	OSC1	XTAL	---	Crystal/Resonator	
PC0/C2IN+	PC0	TTL	CMOS	PORTC I/O	

	C2IN+	AN	---	Comparator2 non-inverting input	
PC1/C2IN-	PC1	TTL	CMOS	PORTC I/O	
	C1IN-	AN	---	Comparator2 inverting input	
PC2	PC2	TTL	CMOS	PORTC I/O	
PC3	PC3	TTL	CMOS	PORTC I/O	
PC4/C2OUT	PC4	TTL	CMOS	PORTC I/O	
	C2OUT	---	CMOS	Comparator2 output	
PC5	PC5	TTL	CMOS	PORTC I/O	
PC6	PC6	TTL	CMOS	PORTC I/O	
PC7	PC7	TTL	CMOS	PORTC I/O	

注意：

1. **IOC**: Interrupt on change, 通用 IO
2. **WPU**: Weak pullup
3. **ST**: 带 CMOS 电平的施密特触发器输入
4. **AN**: 模拟管脚

2. SFR

2.1. 地址映射

2.1.1. SFR, BANK0

ADDR	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR reset
BANK0										
0	INDF	使用 FSR 的内容对数据存储器进行访问（非物理寄存器）								xxxx xxxx
1	TMR0	TIMER0 计数器								xxxx xxxx
2	PCL	程序计数器低 8 位								0000 0000
3	STATUS	-	-	PAGE	/TF	/PF	Z	HC	C	--01 1xxx
4	FSR	间接寻址指针寄存器								xxxx xxxx
5	PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	00x0 0000
6	-	-								---- ----
7	PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	0000 0000
8	-	-								---- ----
9	-	-								---- ----
A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000
B	INTCON	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
C	PIR1	EEIF	CKMEAI F	-	C2IF	C1IF	OSFIF	TMR2IF	-	00-0 000-
D	-	-								---- ----
E	-	-								---- ----
F	-	-								---- ----
10	-	-								---- ----
11	TMR2	TIMER2[7:0] timer2 module register								0000 0000
12	T2CON	-	TOUTPS[3:0]				TMR2ON	T2CKPS[1:0]		-000 0000
13	-	-								---- ----
14	-	-								---- ----
15	-	-								---- ----
16	-	-								---- ----
17	-	-								---- ----
18	WDTCON	-	-	-	WDTPS[3:0]				SWDTEN	---0 1000
19	CMCON0	C2OUT	C1OU T	C2INV	C1INV	CIS	CM[2:0]			0000 0000
1A	PR0	PR0[7:0]								1111 1111

1B	MSCKCON	-	-	-	SLVREN	-	CKMAVG	CKCNTI		---0 -00-
1C	SOSCPRL	SOSCPR [7:0]								1111 1111
1D	SOSCPRH	-	-	-	-	SOSCPR [11:8]				---- 1111
1E	-									---- ----
1F	-									---- ----
20~7F	SRAM BANK0, (96Bytes) 0x00-0x5F									xxxx xxxx

NOTE:

-, 未实现, 保留, 读 0

2.1.2. SFR, BANK1

ADDR	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	POR reset
BANK1										
80	INDF	使用 SFR 的内容对数据存储进行访问（非物理寄存器）								xxxx xxxx
81	OPTION	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
82	PCL	程序计数器低 8 位								0000 0000
83	STATUS	-	-	PAGE	/TF	/PF	Z	HC	C	--01 1xxx
84	FSR	间接寻址指针寄存器								xxxx xxxx
85	TRISA	TRISA[7:6]		-	TRISA[4:0]					11-1 1111
86	-									---- ----
87	TRISC	TRISC[7:0]								1111 1111
88	-									---- ----
89	-									---- ----
8A	PCLATH	-	-	-	程序计数器高 5 位锁存器					---0 0000
8B	INTCON	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF	0000 0000
8C	PIE1	EEIE	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	-	00-0 000-
8D	-									---- ----
8E	PCON	-	-	-	-	-	-	/POR	/BOR	---- --q q
8F	OSCCON	LFMOD	IRCF[2:0]			OSTS	HTS	LTS	SCS	0101 x000
90	-									---- ----
91	-									---- ----
92	PR2	PR2[7:0] timer2 period register								1111 1111
93	-									---- ----
94	-									---- ----
95	WPUA	WPUA[7:6]		-	WPUA[4:0]					11-1 1111
96	IOCA	IOCA[7:0]								0000 0000
97	-									---- ----



NOTE:

注意：

- ### 2.1.3. TMR0, 地址 0x01

Bit	Name	Function
7:0	TMR0[7:0]	Timer 0 计数结果寄存器

2.1.4. STATUS 寄存器, 地址 0x03, 0x83

Bit	7	6	5	4	3	2	1	0
Name	-	-	PAGE	/TF	/PF	Z	HC	C
Reset	-	-	0	1	1	x	x	x
Type	-	-	RW	RO	RO	RW	RW	RW

Bit	Name	Function
7:6	NA	没实现，读 0
5	PAGE	PAGE: Register Bank Select bit (used for direct addressing) 1 = Bank 1 (80h – FFh)
4	/TF	TF: Time-out bit 1 = After power-up, CLRWDT instruction or SLEEP instruction 0 = A WDT time-out occurred
3	/PF	PF: Power-down bit 1 = After power-up or by the CLRWDT instruction 0 = By execution of the SLEEP instruction
2	Z	Z: Zero bit 1 = The result of an arithmetic or logic operation is zero 0 = The result of an arithmetic or logic operation is not zero
1	HC	HC: Digit Carry/Borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions), For Borrow, the polarity is reversed. 1 = A carry-out from the 4 th low-order bit of the result occurred 0 = No carry-out from the 4 th low-order bit of the result
0	C	C: Carry/Borrow bit(1) (ADDWF, ADDLW, SUBLW, SUBWF instructions) 1 = A carry-out from the Most Significant bit of the result occurred 0 = No carry-out from the Most Significant bit of the result occurred

/TF	/PF	条件
1	1	上电或者低电压复位
0	U	WDT 复位
0	0	WDT 唤醒
U	U	正常运行下发生 MCLR 复位
1	0	睡眠状态下发生 MCLR 复位

注意:

1. 和其它寄存器一样，状态寄存器也可以作为任何指令的目标寄存器。如果一条指令影响 Z、HC 或 C 位的指令以状态寄存器作为目标寄存器，将禁止对这三位的写操作，它们只受逻辑结果影响，被置 1 或清 0。因此，当执行一条把状态寄存器作为目标寄存器的指令后，**STATUS** 内容可能和预想的不一致；
2. 建议只使用 **BCR**、**BSR**、**SWAPR** 和 **STR** 指令来改变状态寄存器。

2.1.5. PORTA 寄存器，地址 0x05

Bit	7	6	5	4	3	2	1	0
Name	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RO	RW	RW	RW	RW	RW

Bit	Name	Function
7	PA[7]	PORTA7 数据
6	PA[6]	PORTA6 数据
5	PA[5]	PA5 只有输入功能，不存在相应的数据寄存器
4	PA[4]	PORTA4 数据
3	PA[3]	PORTA3 数据
2	PA[2]	PORTA2 数据
1	PA[1]	PORTA1 数据
0	PA[0]	PORTA0 数据

2.1.6. PORTC 寄存器，地址 0x07

Bit	7	6	5	4	3	2	1	0
Name	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Reset	x	x	x	x	x	x	x	x
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PORTC[7:0]	PORTC 数据寄存器

2.1.7. INTCON 寄存器，地址 0x0B

Bit	7	6	5	4	3	2	1	0
Name	GIE	PEIE	TOIE	INTE	PAIE	TOIF	INTF	PAIF
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
-----	------	----------

7	GIE	GIE: Global Interrupt Enable bit 全局中断使能 1 = Enables all unmasked interrupts 0 = Disables all interrupts
6	PEIE	PEIE: Peripheral Interrupt Enable bit 外设中断使能 1 = Enables all unmasked peripheral interrupts 0 = Disables all peripheral interrupts
5	TOIE	TOIE: Timer0 Overflow Interrupt Enable bit 定时器0溢出中断使能 1 = Enables the Timer0 interrupt 0 = Disables the Timer0 interrupt
4	INTE	INTE: PA2/INT External Interrupt Enable bit 外部中断使能 1 = Enables the PA2/INT external interrupt 0 = Disables the PA2/INT external interrupt
3	PAIE	PAIE: PORTA Change Interrupt Enable bit(1) 1 = Enables the PORTA change interrupt 0 = Disables the PORTA change interrupt
2	TOIF	TOIF: Timer0 Overflow Interrupt Flag bit(2) 定时器0溢出中断标志位 1 = Timer0 register has overflowed (must be cleared in software) 0 = Timer0 register did not overflow
1	INTF	INTF: PA2/INT External Interrupt Flag bit 外部中断标志位 1 = The PA2/INT external interrupt occurred (must be cleared in software) 0 = The PA2/INT external interrupt did not occur
0	PAIF	PAIF: PORTA Change Interrupt Flag bit 1 = When at least one of the PORTA <5:0> pins changed state (must be cleared in software) 0 = None of the PORTA <5:0> pins have changed state

2.1.8. PIR1 寄存器，地址 0x0C

Bit	7	6	5	4	3	2	1	0
Name	EEIF	CKMEAIF	-	C2IF	C1IF	OSFIF	TMR2IF	-
Reset	0	0	-	0	0	0	0	-
Type	RW	RW	-	RW	RW	RW	RW	-

Bit	Name	Function
7	EEIF	EEIF: EE写中断标志位 1 = EE 写操作完成 (必须软件清零) 0 = EE 写操作未完成

6	CKMEAIF	CKMEAIF: 快时钟测量慢时钟操作完成中断标志位 1 = 快时钟测量慢时钟操作完成（必须软件清零） 0 = 快时钟测量慢时钟未完成
5	-	保留位，不要写1
4	C2IF	比较器 2 中断标志位 1 = 比较器 2 输出发生了变化 0 = 比较器 2 输出未发生改变
3	C1IF	比较器 1 中断标志位 1 = 比较器 1 输出发生了变化 0 = 比较器 1 输出未发生改变
2	OSFIF	振荡器故障中断标志位 1 = 系统振荡器发生故障，时钟输入切换为 INTOSC（必须用软件清 0） 0 = 系统时钟运行正常
1	TMR2IF	TMR2IF: Timer2 与PR2比较相等中断标志位 1 = timer2的值等于PR2（必须软件清零） 0 = timer2 的值不等于 PR2
0	-	-

2.1.9. TMR2 寄存器，地址 0x11

Bit	7	6	5	4	3	2	1	0
Name	TMR2[7:0]							
Reset	0000 0000							
Type	RW							

Bit	Name	Function
7:0	TMR2[7:0]	Timer 2 计数结果寄存器

2.1.10. T2CON 寄存器，地址 0x12

Bit	7	6	5	4	3	2	1	0
Name	-	TOUTPS[3:0]				TMR2ON	T2CKPS[1:0]	
Reset	-	0000				0	00	
Type	-	RW				RW	RW	

Bit	Name	Function
7	-	未实现，读 0

6:3	TOUTPS[3:0]	TOUTPS<3:0>: Timer2 Output Postscaler Select bits 定时器2输出后分频比选择 0000 = 1:1 后分频比 0001 = 1:2 后分频比 0010 = 1:3 后分频比 0011 = 1:4 后分频比 0100 = 1:5 后分频比 0101 = 1:6 后分频比 0110 = 1:7 后分频比 0111 = 1:8 后分频比 1000 = 1:9 后分频比 1001 = 1:10 后分频比 1010 = 1:11 后分频比 1011 = 1:12 后分频比 1100 = 1:13 后分频比 1101 = 1:14 后分频比 1110 = 1:15 后分频比 1111 = 1:16 后分频比
2	TMR2ON	TMR2ON: Timer2 On bit 打开定时器2 1 = Timer2 is on 打开 0 = Timer2 is off 关闭
1:0	T2CKPS[1:0]	T2CKPS<1:0>: Timer2 Clock Prescale Select bits 定时器2驱动时钟预分频比选择 00 = Prescaler is 1 01 = Prescaler is 4 1x = Prescaler is 16

2.1.11. WDTCON 寄存器，地址 0x18

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN
Reset	-	-	-	0	1	0	0	0
Type	-	-	-	RW	RW	RW	RW	RW

Bit	Name	Function
7:5	N/A	-
4:1	WDTPS[3:0]	WDTPS<3:0>: 看门狗定时器周期选择: Bit Value = 看门狗定时器驱动时钟之预分频比值 0000 = 1:32

		0001 = 1:64 0010 = 1:128 0011 = 1:256 0100 = 1:512 (复位值) 0101 = 1:1024 0110 = 1:2048 0111 = 1:4096 1000 = 1:8192 1001 = 1:16384 1010 = 1:32768 1011 = 1:65536 1100 = 1:65536 1101 = 1:65536 1110 = 1:65536 1111 = 1:65536
0	SWDTEN	软件打开/关闭看门狗定时器: 1 = 打开 0 = 关闭

2.1.12. CMCON0 寄存器，地址 0x19

Bit	7	6	5	4	3	2	1	0
Name	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
Reset	0	0	0	0	1	0	0	0
Type	RO	RO	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	C2OUT	比较器 2 输出 bit 当 C2INV=0 1: $C2 V_{IN+} > C2 V_{IN-}$ 0: $C2 V_{IN+} < C2 V_{IN-}$ 当 C2INV=1 1: $C2 V_{IN+} < C2 V_{IN-}$ 0: $C2 V_{IN+} > C2 V_{IN-}$
6	C1OUT	比较器 1 输出 bit 当 C1INV=0 1: $C1 V_{IN+} > C1 V_{IN-}$

		0: $C1 V_{IN+} < C1 V_{IN-}$ 当 $C1INV=1$ 1: $C1 V_{IN+} < C1 V_{IN-}$ 0: $C1 V_{IN+} > C1 V_{IN-}$
5	C2INV	比较器 2 输出反向 bit 0 = 不反向 1 = 反向
4	C1INV	比较器 1 输出反向 bit 0 = 不反向 1 = 反向
3	CIS	比较器输入切换 当 $CM[2:0] = 010$ 时, 1 = $C1_{IN+}$ 接 $C1 V_{IN-}$ $C2_{IN+}$ 接 $C2 V_{IN-}$ 0 = $C1_{IN-}$ 接 $C1 V_{IN-}$ $C2_{IN-}$ 接 $C2 V_{IN-}$ 当 $CM[2:0] = 001$ 时, 1 = $C1_{IN+}$ 接 $C1 V_{IN-}$ 0 = $C1_{IN-}$ 接 $C1 V_{IN-}$
2-0	CM[2:0]	比较器模式位: 000 = 比较器关闭, CxIN管脚为模拟IO管脚 001 = 3个输入共用到2个比较器上 010 = 4个输入共用到2个比较器上 011 = 2个共参考比较器 100 = 2个独立比较器 101 = 1个独立比较器 110 = 2个带输出共参考比较器 111 = 比较器关闭, CxIN管脚为数字IO管脚

2.1.13. MSCKCON 寄存器，地址 0x1B

Bit	7	6	5	4	3	2	1	0
MSCKCON	-	-	-	SLVREN	-	CKMAVG	CKCNTI	-
Reset	-	-	-	0	0	0	0	-
Type	-	-	-	RW	RW	RW	RW	-

Bit	Name	Function
7:5	-	保留位，不能写 1

4	SLVREN	软件控制 LVR 使能位 1 = 当 UCFG1<1:0>为 00 时，打开 LVR 0 = 无论 UCFG1<1:0>为何值，禁止 LVR
3	-	保留位，不能写 1
2	CKMAVG	快时钟测量慢时钟周期的测量平均模式 1 = 打开平均模式。（自动测量并累加 4 次） 0 = 关闭平均模式
1	CKCNTI	Clock Count Init -使能快时钟测量慢时钟周期 1 = 使能快时钟测量慢时钟周期 0 = 关闭快时钟测量慢时钟周期 注：这一位在测量完毕后会归零
0	-	保留位，不能写 1

2.1.14. SOSCPR 寄存器，地址 0x1C，1D

SOSCPRL，地址 0x1C

Bit	7	6	5	4	3	2	1	0
Name	SOSCPR[7:0]							
Reset	8'hff							
Type	RW							

SOSCPRH，地址 0x1D

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	SOSCPR[11:8]			
Reset	-	-	-	-	4'hf			
Type	-	-	-	-	RW			

Bit	Name	Function
0x1C: 7:0 0x1D: 3:0	SOSCPR[11:0]	低速振荡器周期（单位：快时钟周期数） 用于慢时钟测量功能

2.1.15. PR0 寄存器，地址 0x1A

Bit	7	6	5	4	3	2	1	0
Name	PR0[7:0]							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	PR0[7:0]	Timer 0 周期(比较)寄存器 (详见 Timer0, timer2 描述章节)

2.1.16. OPTION 寄存器，地址 0x81

Bit	7	6	5	4	3	2	1	0
Name	/PAPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7	/PAPU	PAPU: PORTA Pull-up Enable bit 1 = PORTA pull-ups are disabled 0 = PORTA pull-ups are enabled by individual PORT latch values
6	INTEDG	INTEDG: Interrupt Edge Select bit 1 = Interrupt on rising edge of PA2/INT pin 0 = Interrupt on falling edge of PA2/INT pin
5	T0CS	T0CS: Timer0 Clock Source Select bit 1 = Transition on PA2/T0CKI pin 0 = Internal instruction cycle clock (FOSC/2)
4	T0SE	T0SE: Timer0 Source Edge Select bit 1 = Increment on high-to-low transition on PA2/T0CKI pin 0 = Increment on low-to-high transition on PA2/T0CKI pin
3	PSA	PSA: Prescaler Assignment bit 1 = Prescaler is assigned to the WDT 0 = Prescaler is assigned to the Timer0 module
2	PS2	PS<2:0>: Prescaler Rate Select bits BIT VALUE TIMER0 RATE WDT RATE 000 1 : 2 1 : 1
1	PS1	
0	PS0	

		001	1 : 4	1 : 2
		010	1 : 8	1 : 4
		011	1 : 16	1 : 8
		100	1 : 32	1 : 16
		101	1 : 64	1 : 32
		110	1 : 128	1 : 64
		111	1 : 256	1 : 128

2.1.17. TRISA 寄存器，地址 0x85

Bit	7	6	5	4	3	2	1	0
Name	TRISA[7]	TRISA[6]	TRISA[5]	TRISA[4]	TRISA[3]	TRISA[2]	TRISA[1]	TRISA[0]
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RO	RW	RW	RW	RW	RW

Bit	Name	Function
7:6	TRISA[7:6]	TRISA<7:6> : RA<7:6> Port I/O Tri-State Control bits 1 = Port output driver is disabled 0 = Port output driver is enabled
5	TRISA[5]	TRISA<5> : Input only port Read 1
4:0	TRISA[4:0]	TRISA<4:0> : RA<4:0> Port I/O Tri-State Control bits 1 = Port output driver is disabled 0 = Port output driver is enabled

2.1.18. TRISC 寄存器，地址 0x87

Bit	7	6	5	4	3	2	1	0
Name	TRISC							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	Function
7:0	TRISC[7:0]	PORTC I/O Tri-State Control bits 1 = Port output driver is disabled

		0 = Port output driver is enabled
--	--	-----------------------------------

2.1.19. PIE1 寄存器，地址 0x8C

Bit	7	6	5	4	3	2	1	0
Name	EEIE	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	-
Reset	0	0	-	0	0	0	0	-
TYPE	RW	RW	-	RW	RW	RW	RW	-

Bit	Name	Function
7	EEIE	EEIE: EE写中断使能位 1 = 使能EE 写操作完成中断 0 = 关闭 EE 写操作完成中断
6	CKMEAIE	CKMEAIE: 快时钟测量慢时钟操作完成中断使能位 1 = 使能快时钟测量慢时钟操作完成中断 0 = 关闭快时钟测量慢时钟操作完成中断
5	-	保留位，不要写1
4	C2IE	比较器 2 中断允许位 1 = 允许比较器 2 中断 0 = 禁止比较器 2 中断
3	C1IE	比较器 1 中断允许位 1 = 允许比较器 1 中断 0 = 禁止比较器 1 中断
2	OSFIE	振荡器故障中断允许位 1 = 允许振荡器故障中断 0 = 禁止振荡器故障中断
1	TMR2IE	TMR2IE: Timer2 与PR2比较相等中断使能位 1 = 使能 timer2的值等于PR2 中断 0 = 关闭使能 timer2 的值等于 PR2 中断
0	-	-

2.1.20. OSCCON 寄存器，地址 0x8F

Bit	7	6	5	4	3	2	1	0
Name	LFMOD	IRCF[2]	IRCF[1]	IRCF[0]	OSTS	HTS	LTS	SCS
Reset	0	3'b101			1	0	0	0

TYPE	RW	RW	RO	RO	RO	RW
------	----	----	----	----	----	----

Bit	Name	Function
7	LFMOD	低频内振模式: 1 = 256K 振荡频率模式 0 = 32K 振荡频率模式
6:4	IRCF[2:0]	内部振荡器频率选择 111 =16 MHz 110 =8 MHz 101 =4 MHz(default) 100 =2 MHz 011 =1 MHz 010 =500 kHz 001 =250 kHz 000 =32 kHz (LFINTOSC)
3	OSTS	振荡器起振超时状态位 1 = 器件运行在 FOSC<2:0>指定的外部时钟之下 0 = 器件运行在内部振荡器之下
2	HTS	高速内部时钟状态 1 = HFINTOSC is ready 0 = HFINTOSC is not ready
1	LTS	低速内部时钟状态 1 = LFINTOSC is ready 0 = LFINTOSC is not ready
0	SCS	系统时钟选择位 1 = 系统时钟选择为内部振荡器 0 = 时钟源由 FOSC<2:0>决定

2.1.21. PR2 寄存器，地址 0x92

Bit	7	6	5	4	3	2	1	0
Name	PR2[7:0]							
Reset	1	1	1	1	1	1	1	1
Type	RW	RW	RW		RW	RW	RW	RW

Bit	Name	Function
7:0	PR2[7:0]	Timer 2 周期(比较)寄存器 (详见 Timer2 描述章节)

2.1.22. WPUA 寄存器，地址 0x95

Bit	7	6	5	4	3	2	1	0
Name	WPUA7	WPUA6	WPUA5	WPUA4	WPUA	WPUA2	WPUA1	WPUA0
Reset	1	1	-	1	1	1	1	1
Type	RW	RW	-	RW	RW	RW	RW	RW

Bit	Name	Function
7,6,4,3,2,1,0	WPUAx	Port A 弱上拉使能 1 = 使能 PORT A 端口弱上拉 0 = 断开 PORT A 端口弱上拉
5	NA	PORTA[5]的弱上拉功能由配置寄存器位 MCLRE 决定 MCLRE = 1 使能 PA5 弱上拉 MCLRE = 0 禁止 PA5 弱上拉

2.1.23. IOCA 寄存器，地址 0x96

Bit	7	6	5	4	3	2	1	0
Name	IOCA[7:0]							
Reset	8'h00							
Type	RW							

Bit	Name	Function
7:0	IOCA[7:0]	端口状态触发中断设置 1 = 使能端口状态触发中断 0 = 关闭端口状态触发中断

2.1.24. VRCON 寄存器，地址 0x99

Bit	7	6	5	4	3	2	1	0
Name	VREN	-	VRR	-	VR3	VR2	VR1	VR0
Reset	0	-	0	-	0	0	0	0
Type	RW	-	RW	-	RW	RW	RW	RW

Bit	Name	Function
7	VREN	CVref 使能位 1 = CVref 电路通电 0 = CVref 电路断电，无泄漏电流
6	-	未实现，读 0
5	VRR	CVref 范围选择位 1 = 低电平范围 0 = 高电平范围
4	-	未实现，读 0
3:0	VR<3:0>	CVref 值选择位 VRR=1 时: $CVref = (VR<3:0>/24)*VDD$ VRR=0 时: $CVref = VDD/4 + (VR<3:0>/32)*VDD$

2.1.25. PCON 寄存器，地址 0x8E

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	/POR	/BOR
Reset	-	-	-	-	-	-	q	q
Type	-	-	-	-	-	-	RW	RW

Bit	Name	Function
7:2	-	保留位，读 0
1	/POR	上电复位标志，低有效 0: 发生了上电复位 1: 没发生上电复位或者由软件置 1 /POB 在上电复位后值为 0，此后软件应该将其置 1
0	/BOR	低电压复位标志，低有效 0: 发生了低电压复位 1: 没发生低电压复位或者由软件置 1 /BOR 在上电复位后其值不确定，必须由软件置 1。发生后续复位后，通过查询此位来确定是否低电压复位

2.1.26. EEDAT 寄存器，地址 0x9A

Bit	7	6	5	4	3	2	1	0
Name	EEDAT[7:0]							

Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

2.1.27. EEADR 寄存器，地址 0x9B

Bit	7	6	5	4	3	2	1	0
Name	EEADR[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	RW	RW	RW	RW	RW	RW	RW	RW

2.1.28. EECON1 寄存器，地址 0x9C

Bit	7	6	5	4	3	2	1	0
Name	-	-	WREN3	WREN2	WRERR	WREN1	-	RD
Reset	-	-	0	0	x	0	-	0
Type	-	-	RW	RW	RW	RW	-	WO

Bit	Name	Function
7:6	-	保留位，读 0
5	WREN3	数据 EEPROM 写使能 3 和 WREN2、WREN1 结合使用
4	WREN2	数据 EEPROM 写使能 2 和 WREN3、WREN1 结合使用
3	WRERR	数据 EEPROM 写错误标志位 1: 在 EEPROM 编程周期发生了看门狗或者外部复位，中止 0: 在 EEPROM 编程周期正常完成
2	WREN1	数据 EEPROM 写使能 1 WREN3-1=111: 允许软件对 EEPROM 编程，编程完成后各位自动清 0 WREN3-1=其他值: 禁止软件对 EEPROM 编程
1	-	保留位，读 0
0	RD	数据 EEPROM 读控制位 此位是只写，读永远返回 0 写 1: 启动一次数据 EEPROM 读周期 写 0: 不启动读

2.1.29. EECON2 寄存器，地址 0x9D

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	WR
Reset	-	-	-	-	-	-	-	0
Type	-	-	-	-	-	-	-	RW

Bit	Name	Function
7:1	-	保留位，读 0
0	WR	数据 EEPROM 写控制位 读操作： 1: 数据 EEPROM 编程周期进行中 0: 数据 EEPROM 不处于编程周期 写操作： 1: 启动一次数据 EEPROM 编程周期 0: 无意义

2.1.30. 配置寄存器 UCFGx

软件不能访问 UCFG0、UCFG1、UCFG2，它们只在上电过程由硬件写入。

UCFG0，PROM 地址 0x2000。

Bit	7	6	5	4	3	2	1	0
Name	-	CPB	MCLRE	PWRTEB	WDTE	FOSC<2:0>		

位	名称	描述
Bit[7]	NA	保留位，读 0
Bit[6]	CPB	1: Flash 内容不保护 0: 启动 Flash 内容保护，MCU 能读，串口不能读 注意： 此位只能由 1 改写为 0，而不能由 0 改写为 1。由 0 改写成 1 的唯一方法是进行一次包括 USER_OPT 在内的片擦操作，并且重新上电后 CPB 才变为 1
Bit[5]	MCLRE	1: PA5/MCLR 脚执行 MCLR 功能，是复位脚 0: PA5/MCLR 脚执行 PA5 功能，是数字输入引脚
Bit[4]	PWRTEB	1: PWRT 禁止 0: PWRT 使能
Bit[3]	WDTE	1: WDT 使能，程序不能禁止 0: WDT 禁止，但程序可通过设置 WDTCON 的 SWDTEN 位将 WDT 使能

Bit[2:0]	FOSC[2:0]	000: 32K 晶振模式, PA6/PA7 接低速晶体 001: 20M 晶振模式, PA6/PA7 接高速晶体 010: 外部时钟模式, PA6 为 IO 功能, PA7 接时钟输入 011: INTOSC 模式, PA6 输出系统时钟的 2 分频, PA7 为 IO 引脚 1xx: INTOSCIO 模式, PA6 为 IO 引脚, PA7 为 IO 引脚
----------	-----------	--

UCFG1, PROM 地址 0x2001。

Bit	7	6	5	4	3	2	1	0
Name	-	-	TSEL	FCMEN	IESO	RD_CTRL	LV DEN1	LV DEN0

位	名称	描述
Bit[7:6]	-	保留
Bit[5]	TSEL	指令周期选择位 1: 指令周期为 2T 0: 指令周期为 4T
Bit[4]	FCMEN	时钟故障监视使能 1: 使能时钟故障监视 0: 禁止时钟故障监视
Bit[3]	IESO	双速时钟使能 1: 使能双速时钟模式 0: 禁止双速时钟模式
Bit[2]	RD_CTRL	输出模式时读端口控制 1: 读数据端口返回的 PAD 上的值 0: 读数据端口返回的 Latch 上的值
BIT[1:0]	LV DEN[1:0]	低电压复位选择 00: 低电压复位使能 其它: 禁止低电压复位

UCFG2, PROM 地址 0x2002。

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	LV DS[3:0]			

位	名称	描述	
Bit[7:4]	-	保留位	
Bit[3:0]	LVDS[3:0]	低电压复位阈值选择	
		数值	电压
		0010	1.8V
		0011	2.0V
		0100	2.2V

		0110	2.8V
		其它	保留

2.1.31. PCL 和 PCLATH

程序计数器（PC）为 11 位宽。其低 8 位来自可读写的 PCL 寄存器，高 3 位（PC<10:8>）来自 PCLATH，不能直接读写。只要发生复位，PC 就将被清 0。下图显示了装载 PC 值的两种情形。注意图右边的 LCALL 和 LJUMP 指令，由于指令中的操作码为 11 位，而芯片的 PC 刚好是 11 位，所以这时 PCLATH 并不需要用到。

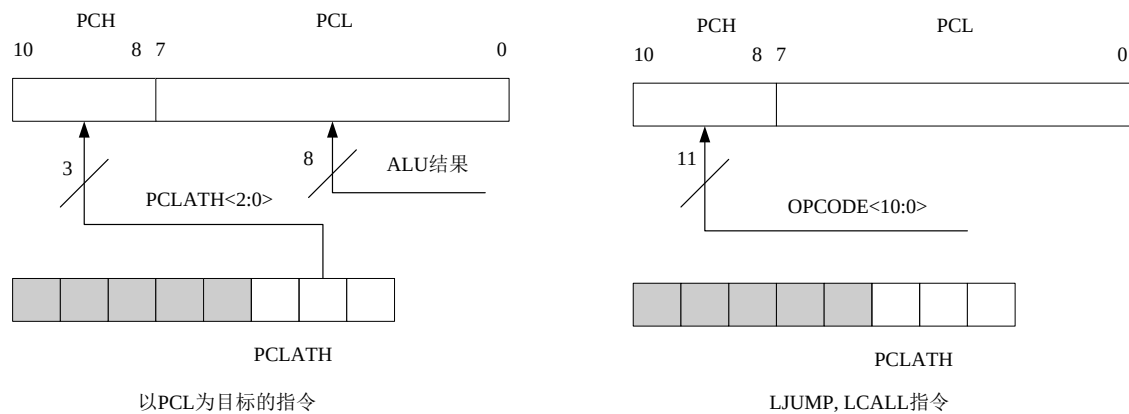


图 2.1.1 不同情况下 PC 的加载

修改 PCL

执行任何以 PCL 寄存器为目标寄存器的指令将同时使程序计数器 PC<10:8> 位被 PCLATH 内容取代。这样可通过将所需的高 3 位写入 PCLATH 寄存器来改变程序计数器的所有内容。

计算 LJUMP 指令是通过向程序计数器加入偏移量（ADDWR PCL）来实现的。通过修改 PCL 寄存器跳转到查找表或程序分支表（计算 LJUMP）时应特别谨慎。假定 PCLATH 设置为表的起始地址，如果表长度大于 255 条指令，或如果存储器地址的低 8 位在表的中间从 0xFF 计满返回到 0x00，那么在每次表起始地址与表内的目标地址之间发生计满返回时，PCLATH 必须递增。

2.1.32. INDF 和 FSR 寄存器

INDF 不是物理存在的寄存器，对 INDF 进行寻址将产生间接寻址，可寻址范围为 0~255。任何使用 INDF 寄存器的指令，实际上是对文件选择寄存器 FSR 所指向的单元进行存取。间接对 INDF 进行读操作将返回 0。间接对 INDF 进行写将导致空操作（可能会影响状态标志位）。

3. 系统时钟源

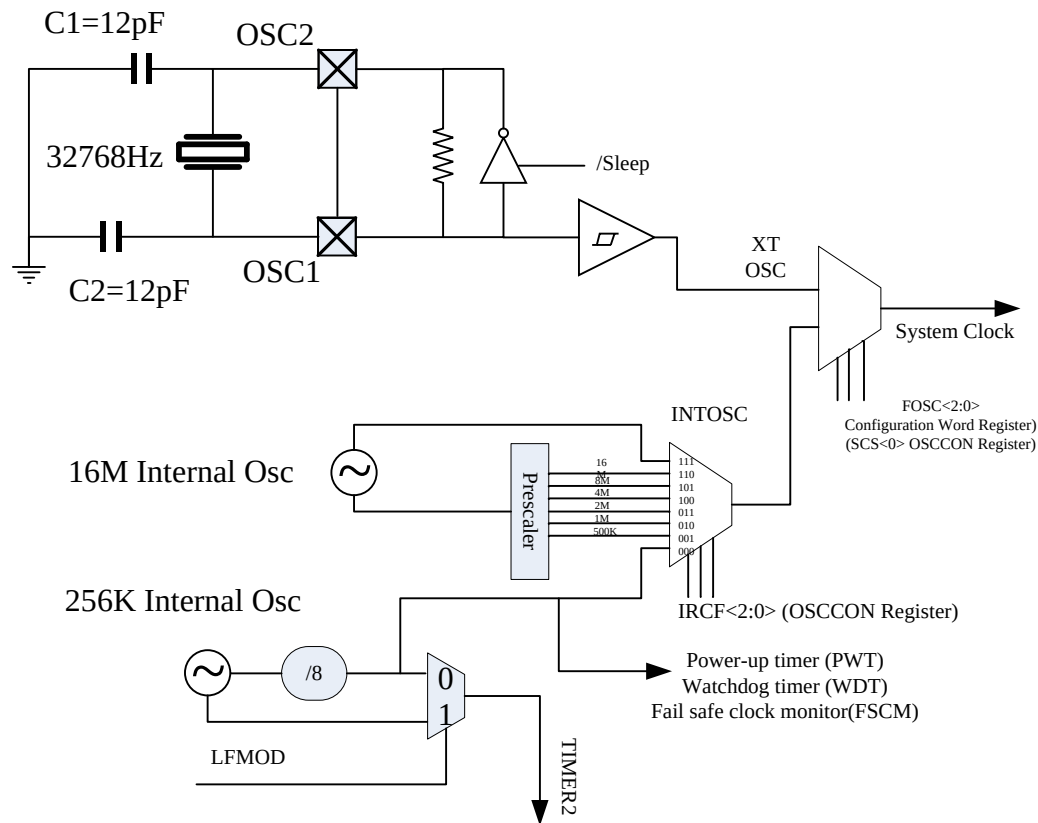


图 3.1 系统时钟源框图

本芯片包含 4 个时钟源：2 个内置振荡器作为各种时钟源，1 个外部晶体振荡器，1 个外部时钟灌入源。内置振荡器包括 1 个内部 16M 高速精准振荡器(HFINTOSC), 1 个内部 32K/256K(LFINTOSC) 低速低功耗振荡器。这些时钟或振荡器结合预分频器可以给系统提供各种频率的时钟源。

系统时钟源的预分频器比例由 OPTION 寄存器里的 IRCF<2:0>位控制。

注意：

看门狗、系统时钟源（IRCF=000）以及 PWRT 统一使用 8 分频之后的输出，即 32KHz，而不管 LFMOD 为何值。

3.1. 时钟源模式

时钟源模式分为外部和内部模式。

- 外部时钟模式依靠外部电路提供时钟源，比如外部时钟 EC 模式，晶体谐振器 XT、LP 模式。
- 内部时钟模式内置于振荡器模块中，振荡器模块有 16MHz 高频振荡器和 32KHz 低频振荡器。可通过 OSCCON 寄存器的系统时钟选择位（SCS）来选择内部或者外部时钟源。

3.2. 外部时钟模式

3.2.1. 振荡器起振定时器（OST）

如果振荡器模块配置为 LP、XT 模式，振荡器起振定时器（OST）将对来自 OSC1 的振荡计数 1024 次。这发生在上电复位（POR）之后以及上电延时定时器（PWRT）延时结束（如果被使能）时，或从休眠中唤醒后。在此期间，程序计数器不递增，程序执行暂停。OST 确保使用石英晶体谐振器或陶瓷谐振器的振荡器电路已经启动并向振荡器模块提供稳定的系统时钟信号。当在时钟源之间切换时，需要一定的延时以使新时钟稳定。

3.2.2. EC 模式

外部时钟模式允许外部产生的逻辑电平作为系统时钟源。工作在此模式下时，外部时钟源连接到 OSC1 输入，OSC2 引脚可用作通用 I/O。

当选取 EC 模式时，振荡器起振定时器（OST）被禁止。因此，上电复位（POR）后或者从休眠中唤醒后的操作不存在延时。MCU 被唤醒后再次启动外部时钟，器件恢复工作，就好像没有停止过一样。

3.2.3. LP 和 XT 模式

LP 和 XT 模式支持连接到 OSC1 和 OSC2 的石英晶体谐振器或陶瓷谐振器的使用。

模式选择内部反相放大器的低或高增益设定，以支持各种谐振器类型及速度。

LP 振荡器模式选择内部反相放大器的最低增益设定。

LP 模式的电流消耗在两种模式中最小。该模式设计仅用于驱动 32.768 kHz 音叉式晶振（钟表晶振）。

XT 振荡器模式选择内部反相放大器的高增益设定。

3.3. 内部时钟模式

振荡器模块有两个独立的内部振荡器，可配置或选取为系统时钟源。

1. HFINTOSC（高频内部振荡器）出厂时已校准，工作频率为 16MHz。
2. LFINTOSC（低频内部振荡器）未经校准，工作频率为 32 kHz。软件对 OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0>进行操作，可选择系统时钟速度。

可通过 OSCCON 寄存器的系统时钟选择（SCS）位，在外部或内部时钟源之间选择系统时钟。

注意：OSCCON 寄存器的 LFMOD 可以选择 LFINTOSC 是 32KHz 或者 256KHz，但看门狗固定使用 32KHz，不管 LFMOD 为何值。

3.3.1. 频率选择位（IRCF）

16MHz HFINTOSC 和 32KHz LFINTOSC 的输出连接到预分频器和多路复用器（见图 3.1）。OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0>用于选择内部振荡器的频率输出。可通过软件选择以下 8 个频率之一：

- 16MHz
- 8MHz
- 4MHz（复位后的缺省值）
- 2MHz
- 1MHz
- 500 kHz
- 250 kHz
- 32 kHz

3.3.2. HFINTOSC 和 LFINTOSC 时钟切换时序

当在 LFINTOSC 和 HFINTOSC 之间切换时，新的振荡器可能为了省电已经关闭（见图 3.2 和图 3.3）。在这种情况下，OSCCON 寄存器的 IRCF 位被修改之后、频率选择生效之前，存在一个延时。OSCCON 寄存器的 LTS 和 HTS 位将反映 LFINTOSC 和 HFINTOSC 振荡器的当前活动状态。频率选择时序如下：

1. OSCCON 寄存器的 IRCF<2:0>位被修改。
2. 如果新时钟是关闭的，开始一个时钟启动延时。
3. 时钟切换电路等待当前时钟下降沿的到来。
4. CLKOUT 保持为低，时钟切换电路等待两个新时钟下降沿的到来。
5. 现在 CLKOUT 连接到新时钟。OSCCON 寄存器的 HTS 和 LTS 位按要求被更新。
6. 时钟切换完成。

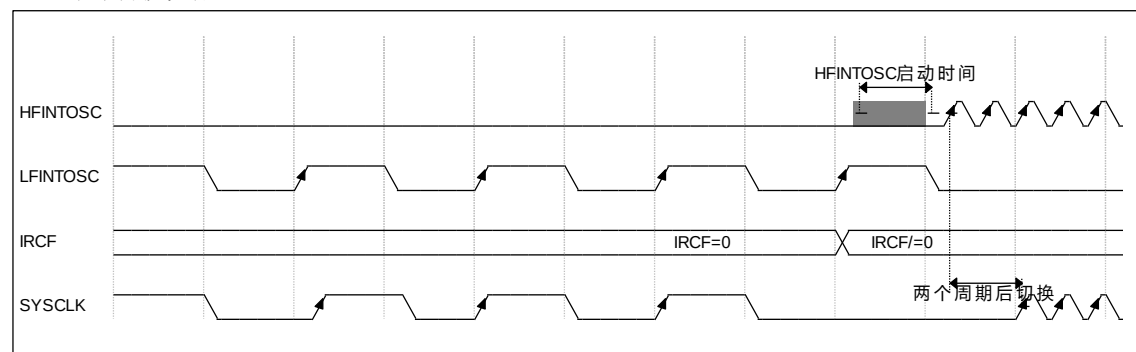


图 3.2 由慢时钟切换到快时钟

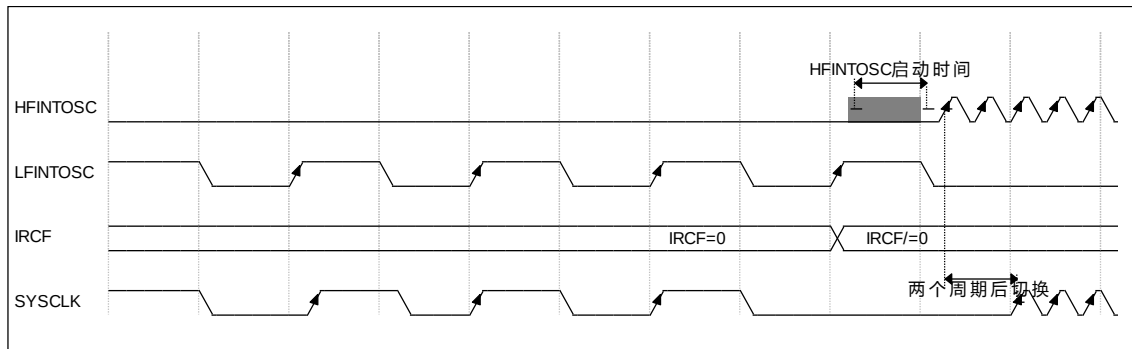


图 3.3 由快时钟切换到慢时钟

3.4. 时钟切换

通过软件对 OSCCON 寄存器的系统时钟选择（SCS）位进行操作，可将系统时钟源在外部和内部时钟源之间切换。

3.4.1. 系统时钟选择（SCS）位

OSCCON 寄存器的系统时钟选择（SCS）位选择用于 CPU 和外设的系统时钟源。

- OSCCON 寄存器的位 SCS = 0 时，系统时钟源由配置字寄存器（UCFG0）中 FOSC<2:0>位的配置决定。
- OSCCON 寄存器的位 SCS = 1 时，根据 OSCCON 寄存器的 IRCF<2:0>位所选的内部振荡器频率选取系统时钟源。复位后，OSCCON 寄存器的 SCS 总是被清零。

注：任何由硬件引起的时钟切换（可能产生自双速启动或故障保护时钟监控器）都不会更新 OSCCON 寄存器的 SCS 位。用户应该监控 OSCCON 寄存器的 OSTS 位以确定当前的系统时钟源。

3.4.2. 振荡器起振超时状态（OSTS）位

OSCCON 寄存器的振荡器起振超时状态（OSTS）位用于指示系统时钟是来自外部时钟源，还是来自内部时钟源。外部时钟源由配置字寄存器（UCFG0）的 FOSC<2:0>定义。OSTS 还特别指明在 LP 或 XT 模式下，振荡器起振定时器（OST）是否已超时。

3.5. 双速时钟启动模式

双速启动模式通过最大限度地缩短外部振荡器起振与代码执行之间的延时，进一步节省了功耗。对于频繁使用休眠模式的应用，双速启动模式将在器件唤醒后除去外部振荡器的起振时间，从而可降

低器件的总体功耗。该模式使得应用能够从休眠中唤醒，将 INTOSC 用作时钟源执行数条指令，然后再返回休眠状态而无需等待主振荡器的稳定。

注：执行 SLEEP 指令将中止振荡器起振时间，并使 OSCCON 寄存器的 OSTS 位保持清零。

当振荡器模块配置为 LP 或 XT 模式时，振荡器起振定时器（OST）使能（见第 3.2.1 节“振荡器起振定时器”）。OST 将暂停程序执行，直到完成 1024 次振荡计数。双速启动模式在 OST 计数时使用内部振荡器进行工作，使代码执行的延时最大限度地缩短。当 OST 计数到 1024 且 OSCCON 寄存器的 OSTS 位置 1 时，程序执行切换至外部振荡器。

3.5.1. 双速启动模式配置

通过以下设定来配置双速启动模式：

- 配置字寄存器（UCFG1）中的位 IESO = 1；内部/外部切换位（使能双速启动模式）。
- OSCCON 寄存器的位 SCS = 0。
- 配置字寄存器（CONFIG）中的 FOSC<2:0>配置为 LP 或 XT 模式。

在下列操作之后，进入双速启动模式：

- 上电复位（POR）且上电延时定时器（PWRT）
- 延时结束（使能时）后，或者从休眠状态唤醒。

如果外部时钟振荡器配置为除 LP 或 XT 模式以外的任一模式，那么双速启动将被禁止。这是因为 POR 后或从休眠中退出时，外部时钟振荡器不需要稳定时间。

3.5.2. 双速启动顺序

1. 从上电复位或休眠中唤醒。
2. 使用内部振荡器以 OSCCON 寄存器的 IRCF<2:0>位设置的频率开始执行指令。
3. OST 使能，计数 1024 个时钟周期。
4. OST 超时，等待内部振荡器下降沿的到来。
5. OSTS 置 1。
6. 系统时钟保持为低，直到新时钟下一个下降沿的到来（LP 或 XT 模式）。
7. 系统时钟切换到外部时钟源。

3.6. 故障保护时钟监控器

故障保护时钟监控器（FSCM）使得器件在出现外部振荡器故障时仍能继续工作。FSCM 能在振荡器起振延时定时器（OST）到期后的任一时刻检测振荡器故障。FSCM 通过将配置字寄存器（UCFG1）中的 FCMEN 位置 1 来使能。FSCM 可用于所有外部振荡模式（LP、XT 和 EC）。



FSCM 模块通过将外部振荡器与 FSCM 采样时钟比较来检测振荡器故障。LFINTOSC 除以 64，就产生了采样时钟。请参见图 3.4。故障检测器内部有一个锁存器。在外部时钟的每个下降沿，锁存器被置 1。在采样时钟的每个上升沿，锁存器被清零。如果采样时钟的整个半周期流逝而主时钟依然未进入低电平，就检测到故障。

当外部时钟出现故障时，FSCM 将器件时钟切换到内部时钟源，并将 PIR1 寄存器的 OSFIF 标志位置 1。如果在 PIR1 寄存器的 OSFIE 位置 1 的同时将该标志位置 1，将产生中断。器件固件随后会采取措施减轻可能由故障时钟所产生的问题。系统时钟将继续来自内部时钟源，直到器件固件成功重启外部振荡器并切换回外部操作。

复位、执行 SLEEP 指令或翻转 OSCCON 寄存器的 SCS 位后，故障保护条件被清除。OSCCON 寄存器的 SCS 位被修改后，OST 将重新启动。OST 运行时，器件继续从 OSCCON 中选定的 INTOSC 进行操作。OST 超时后，故障保护条件被清除，器件将从外部时钟源进行操作。必须先清除故障保护条件，才能清零 OSFIF 标志位。

3.6.4. 复位或从休眠中唤醒

FSCM 设计为能在振荡器起振延时定时器（OST）到期后的任一时刻检测振荡器故障。OST 的使用场合为从休眠状态唤醒后以及任何类型的复位后。OST 不能在 EC 时钟模式下使用，所以一旦复位或唤醒完成，FSCM 就处于激活状态。当 FSCM 被使能时，双速启动也被使能。因此，当 OST 运行时，器件总是处于代码执行阶段。

注：由于振荡器起振时间的范围变化较大，在振荡器起振期间（从复位或休眠中退出时），故障保护电路不处于激活状态。经过一段适当的时间后，用户应检查 OSCCON 寄存器的 OSTS 位，以验证振荡器是否已成功起振以及系统时钟是否切换成功。

4. 复位时序

TX-I-02X 有以下几种不同的复位：

- A) 上电复位 POR
- B) WDT(看门狗)复位 – 在常规运行期间
- C) WDT(看门狗)唤醒 – 在睡眠期间
- D) /MCLR 管脚复位 – 在常规运行期间
- E) /MCLR 管脚复位 – 在睡眠期间
- F) 低电压 (BOR/LVR) 复位
- G) 指令错误复位 (可禁止)

有些寄存器是不被任何复位影响的；这些寄存器的状态在上电复位时是未知的，也不受复位事件影响。大多数其它寄存器都会在以下复位事件时恢复到其“复位状态”：

- 上电复位 POR
- WDT(看门狗)复位 – 在常规运行期间
- WDT(看门狗)复位 – 在睡眠期间
- /MCLR 管脚复位 – 在常规运行期间
- 低电压 (BOR) 复位
- 错误指令复位

WDT(看门狗)睡眠唤醒不会造成和在常规运行状态下 WDT(看门狗)超时所造成的复位。因为睡眠唤醒本身就是一种继续的意思，而不是复位/TO 和/PD 位的设置和清零在不同复位条件下的动作是不同的。具体可参考表 4.1 和 4.2。

/MCLR 管脚背后的电路带有防抖功能，能够滤除一些干扰造成的尖细脉冲信号。

下图为复位电路的总体概述框图。

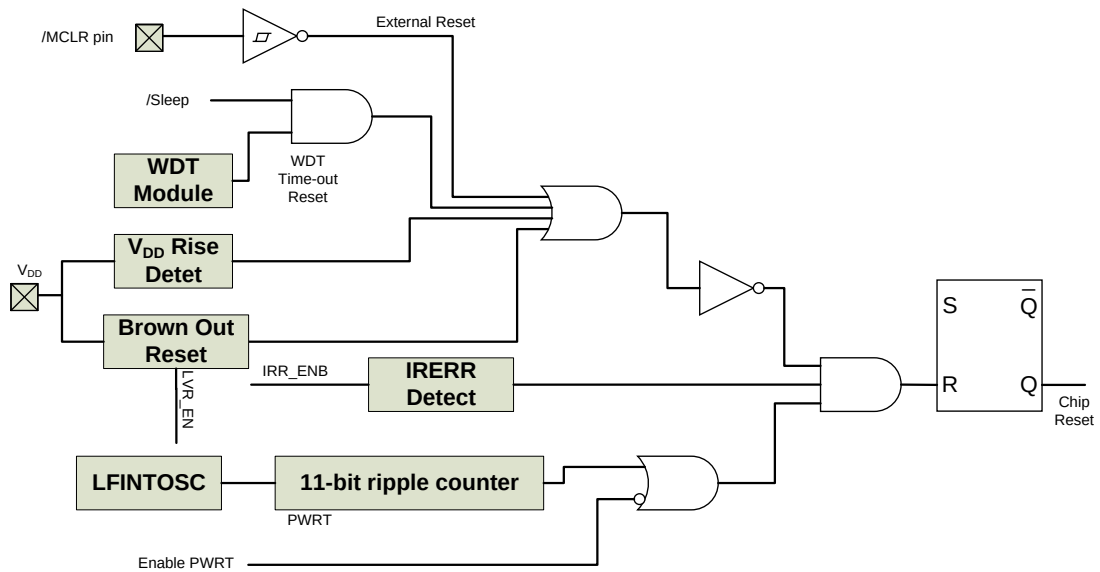


图 4.1 复位功能框图

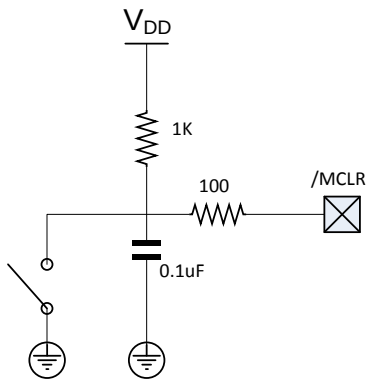
4. 1. POR 上电复位

片上的 POR 电路会将芯片保持在复位状态直到 VDD 电源电压达到足够高，为充分利用片内复位电路的功能，用户可以简单地直接在 VDD 和 /MCLR 之间结一个电阻。这样外部就无需任何 RC 复位电路。不过这要求 VDD 电压上升时间为最大。

上电完成后，系统复位不会立即释放，还要等一个约 4ms 的延时，期间数字电路保持在复位状态。

4. 2. 外部复位 MCLR

需要注意的是，WDT 复位不会把 /MCLR 管脚拉低。在 /MCLR 管脚上施加超过指标的电压（例如 ESD 事件）会造成 /MCLR 复位，而且在管脚上产生超标的大电流，因此我们推荐用户不再直接用一个电阻将 /MCLR 和 VDD 连接起来，而是采用以下电路。



在芯片的 CONFIG OPTION 寄存器（UCFG0）中有一个 MCLRE 使能位，将此位清零会使得复位信号由芯片内部产生。当此位为 1 时，芯片的 PA5/MCLR 脚成为外部复位脚。在这个模式下，/MCLR 脚上有个对 VDD 的弱上拉。

4. 3. PWRT（上电计时器）

PWRT 为上电复位，低电压复位提供一个固定的 64ms（正常情况下）的定时。这个定时器由内部慢时钟驱动。芯片在定时器超时之前都是被保持在复位状态。这段时间能保证 VDD 上升到足够高的电压使得系统能正常工作。PWRT 也可以通过系统 CONFIG 寄存器（UCFG0）来使能。在开启低电压复位功能时，用户应该也打开 PWRT。PWRT 定时是由 VDD 电压超过 VBOR 门限事件启动的。另外需要注意的是，由于由内部慢时钟驱动，定时的实际时间长度是随温度，电压等条件变化而变化的。这个时间不是一个精准参数。

4. 4. BOR (LVR) 低电压复位

低电压复位由 UCFG1<1:0>位来控制。低电压复位就是指当电源电压低于 VBOR 门限电压时所产生的复位。不过当 VDD 电压低于 VBOR 不超过 TBOR 时间时，低电压复位可能不会发生。

VBOR 电压在芯片出货之前需要校准，校准可通过串口写入内部校准寄存器来完成。

如果 BOR（低电压复位）是使能（UCFG1<1:0>=00）的，那么最大 VDD 电压上升时间的要求就不存在。BOR 电路会将芯片控制在复位状态，一直到 VDD 电压达到 VBOR 门限电压以上。

需要注意的是，当 VDD 低于系统能正常工作的门限时，POR 电路并不会产生复位信号。如果要 BOR 电路产生复位信号，VDD 电压必须在 VSS 电平上保持 100uS 以上。

4.5. 错误指令复位

当 CPU 的指令寄存器取指到未定义指令时，系统将进行复位，利用此功能可增加系统的抗干扰能力。

4.6. 超时动作

在上电过程中，芯片内部的超时动作顺序按以下流程执行：

POR 结束后启动 PWRT 计时

由于计时是由 POR 脉冲结束启动的，如果/MCLR 在低电平状态下保持足够长的时间，超时事件就会发生。那么将/MCLR 拉高会让 CPU 立即开始执行。这在测试或者需要多个 MCU 同步的情况下会很有用。

Power Control Register (PCON)

PCON 寄存器里有 2 位指示哪一种复位发生了。Bit0 是/BOR 指示位，其在上电复位是未知态，软件必须将其置 1，然后检查其是否为 0。Bit1 是/POR 指示位，其在上电复位后为 0，软件必须将其置 1。

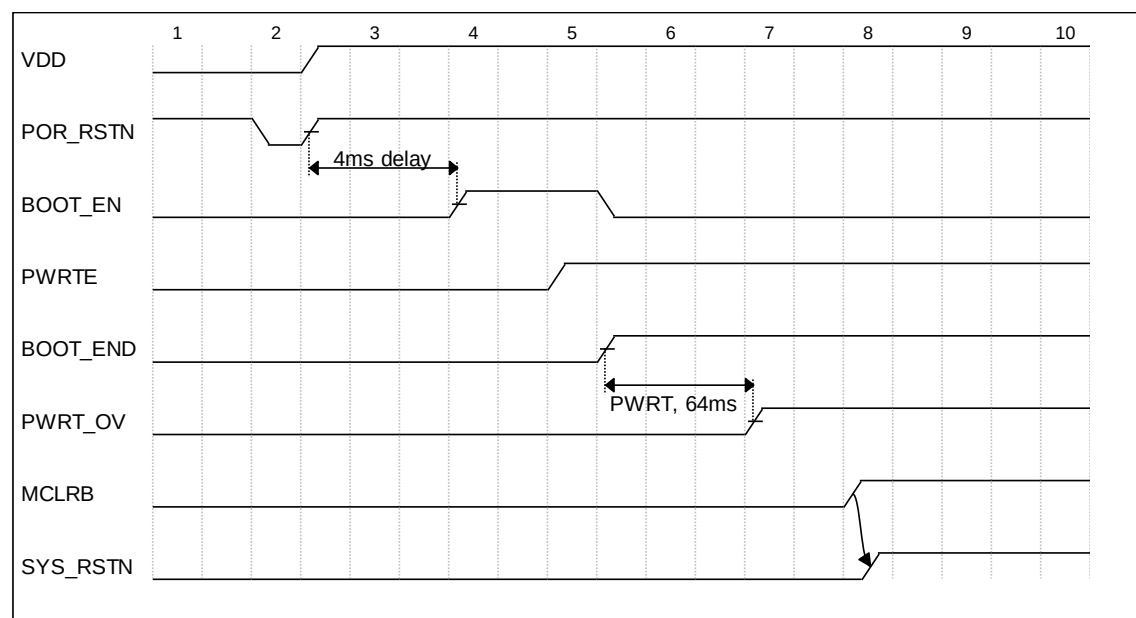


图 4.2 上电复位，使用了 MCLR

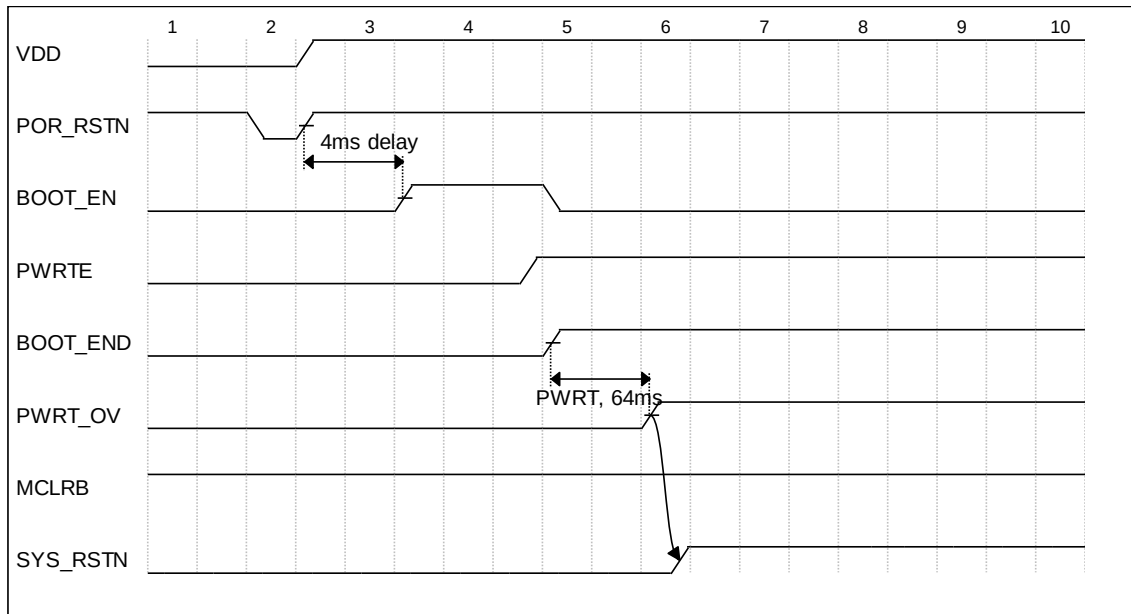


图 4.3 上电复位，没使用 MCLR

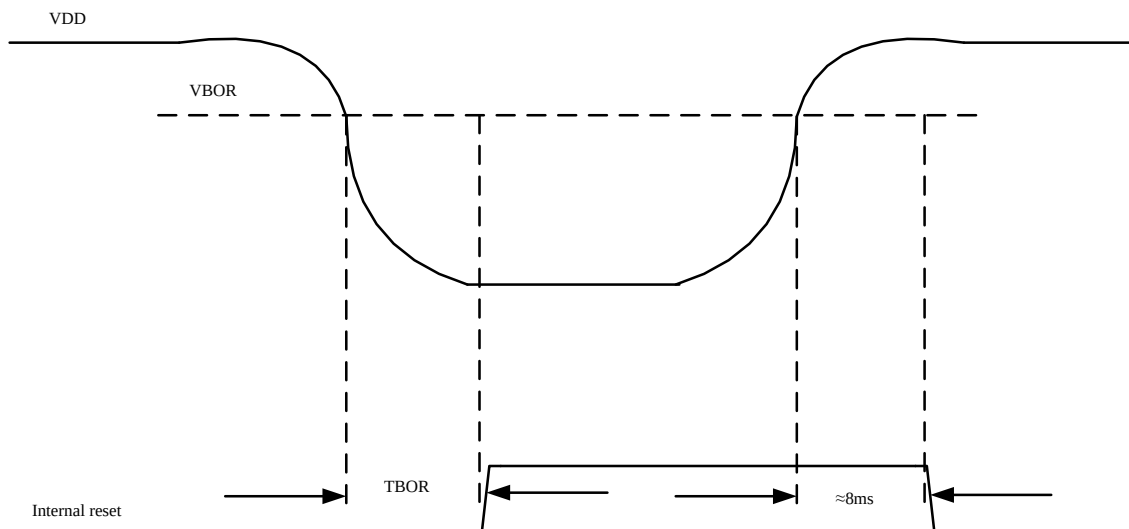


图 4.4 BOR 复位

注意：

1. 上电复位或低电压复位后，并且在 **PWRTEB (UCFG0.4)** 为低时，**PWRT** 有效。它是 **2048 个内部慢时钟周期，约 64ms**；
2. **TBOR** 时间约为 **157us**；
3. 电压恢复正常之后，内部复位不会立即释放，而是要等约为 **4ms** 的时间。

振荡器配置	上电复位	低电压复位	睡眠醒来
-------	------	-------	------

	/PWRTEB=0	/PWRTEB=1	/PWRTEB=0	/PWRTEB=1	
INTOSC	TPWRT	-	TPWRT	-	-

表 4.1 各种情况下的超时

/POR	/BOR	/TO	/PD	条件
0	x	1	1	POR
u	0	1	1	BOR
u	u	0	u	WDT 复位
u	u	0	0	WDT 唤醒
u	u	u	u	常规运行下/MCLR 复位
u	u	1	0	睡眠下/MCLR 复位

表 4.2 STATUS/PCON 位及其意义（u-没变化 x-未知）

5. BOOT

在上电复位或低电压复位之后，插入一个状态，把程序 EEPROM 的 2000H 开始的单元映射到配置寄存器。

系统复位要等 BOOT 结束后才能释放，如图 4.2 和图 4.3 所示，该过程大概需要 17us。

6. 烧录和控制串口

TX-I-02X 具备一个烧录和控制串口。这个接口为用户和开发人员使用此芯片带来方便，具体请参考相关调试文档。

图 7.1 看门狗和定时器 0 框图

在使能了看门狗的情况下，MCU 睡眠时看门狗溢出事件可以作为一个唤醒源，而 MCU 正常工作时作为一个复位源。

条件	看门狗状态
WDTEN 和 SWDTEN 同时为 0	清零
CLRWDT 指令	
进入 SLEEP、退出 SLEEP 时刻	

1. 如果内部慢时钟从 32K 切换到 256K 模式(或反之从 256K 切换到 32K 模式), 都不影响看门狗计时, 因为 WDT 固定使用 32K 时钟源, 见 3.1 小节的时钟框图。

8. 定时器 0

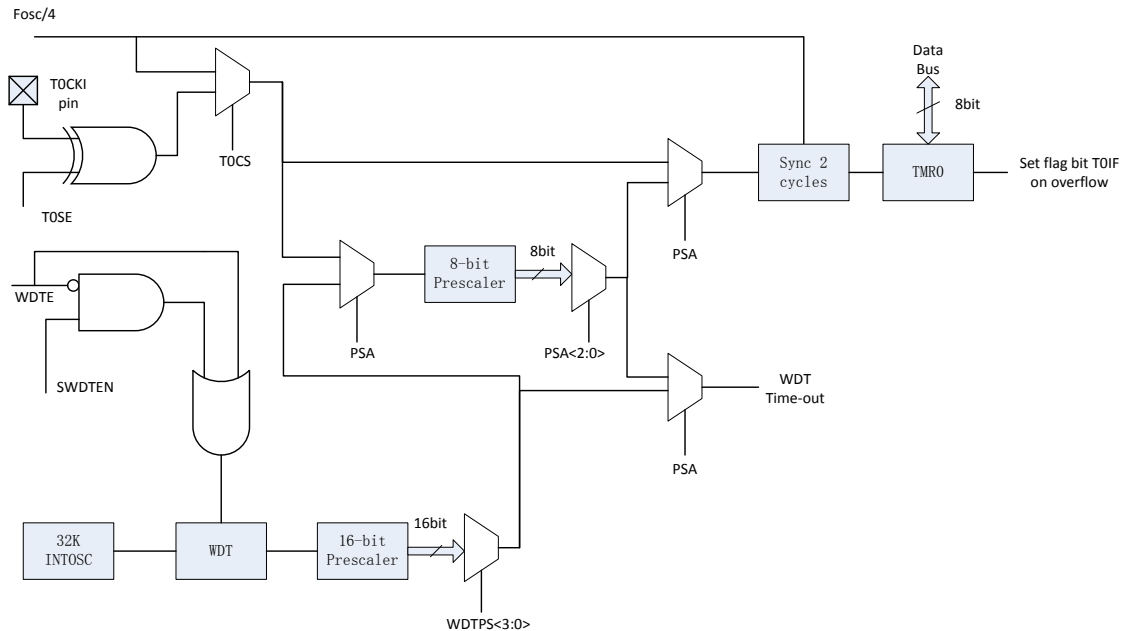


图 8.1 看门狗和定时器 0 框图

8.1. Timer 0

定时器 0 为 8 位，可配置为计数器或定时器使用，当作为外部事件（T0CKI）计数器时，可以配置为上升沿或者下降沿计数。作为定时器时，其计数时钟为系统时钟的 2 分频，即每一指令周期递增一次。

有一个与 WDT 共用的 8 位预分频器，PSA 为 0 时该预分频器分配给定时器 0 使用。

注意：

1. 当改变 PSA 的值时，硬件会自动把预分频器清 0。

8.2. Timer 0 定时器模式

该模式下，定时器 0 在每个指令周期加 1(不带预分频)。软件可以清零 OPTION 寄存器里的 T0CS 位以进入定时器模式。当软件对 TMR0 进行写操作时，定时器在写后面 2 个周期内不会递增。

8.3.Timer 0 计数器模式

该模式下，定时器 0 由每个 T0CKI 管脚的上升沿或下降沿触发加 1(不带预分频)。具体哪一钟沿触发由 OPTION 寄存器里的 T0SE 位决定。软件可以将 OPTION 寄存器里的 T0CS 位置 1 以进入计数器模式。

8.3.1. 软件可配置预分频电路

芯片在定时器 0 和 watchdog 定时器前面有一个分频电路，可以分配给 Timer0 或者 watchdog 定时器用，但二者不能同时使用这个预分频电路。具体分配给 Timer0 还是 watchdog 由 OPTION 寄存器里的 PSA 位决定，PSA 为 0 时，预分频分配给 Timer0 使用。在 Timer0 预分频模式下，总共有 8 个预分频比（1:2 到 1:256）可以通过 OPTION 寄存器里的 PS[2:0]位设置。

预分频电路既不可读也不可写。任何对 TMR0 寄存器的写动作会清零预分频电路。

当预分频电路分配给 watchdog 时，1 条 CLRWDT 指令会清零预分频电路。

在定时器和 watchdog 之间切换预分频电路

由于分频电路可以分配给 Timer0 或者 watchdog 定时器用，在二者之间切换预分频器是有可能导致误复位。

在将预分频电路从分配给 TMR0 切换到分配给 watchdog 时，请务必执行以下指令顺序：

```
BANKSEL TMR0
CLRWDT                                ;Clear WDT
CLRR TMR0                             ;Clear TMR0 and prescaler
BANKSEL OPTION_REG
BSR OPTION_REG,PSA                    ;Select WDT
CLRWDT

LDWI b'11111000'                      ;Mask prescaler bits
ANDWR OPTION_REG,W
IORWI b'00000101'                      ;Set WDT prescaler to 1:32
LDWI OPTION_REG
```

在将预分频电路从分配给 watchdog 切换到分配给 TMR0 时，请务必执行以下指令顺序：

```
CLRWDT                                ;Clear WDT and prescaler
BANKSEL OPTION_REG
LDWI b'11110000'                      ;Mask TMR0 select and prescaler bits
ANDWR OPTION_REG,W
IORWI b'00000011'                      ;Set prescale to 1:16
STR OPTION_REG ;
```

8.3.2. 定时器 0 中断

芯片在定时器 0 从 0xFF 溢出到 0x00 时会置起 T0IF 标志，并产生中断（如果使能了的话）。注意，timer0 中断无法唤醒 CPU 因为在睡眠状态下，定时器是被冻结的。

8.3.3. 用外部时钟驱动定时器 0

在计数其模式下，T0CKI 管脚输入和 Timer0 寄存器之间的同步是由在 Q1, Q2 内部时钟相位采样实现的，所以外部时钟源周期的高电平时间和低电平时间必须满足相关时序要求。

9. 定时器 2

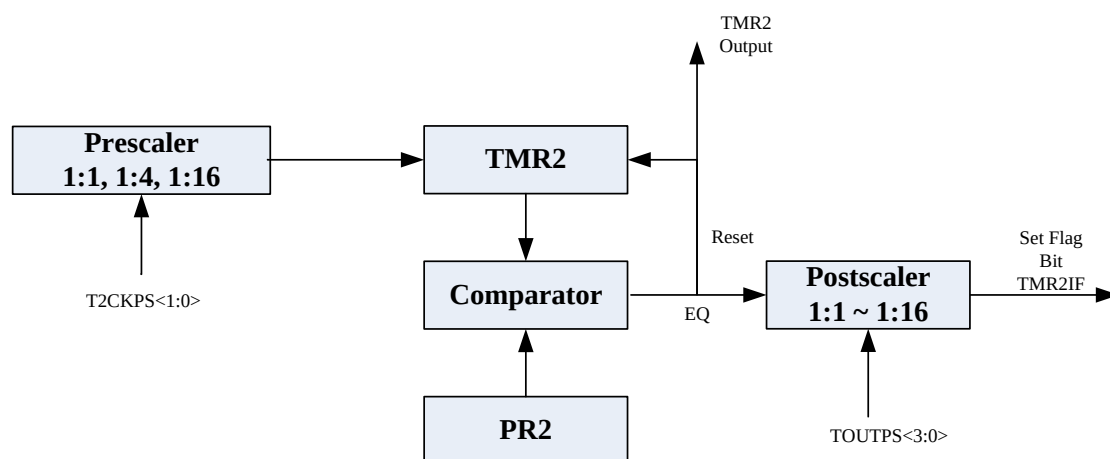


图 8.1 定时器 2 框图

定时器 2 为 8 位定时器包含以下功能：

- 8 位计数寄存器
- 8 位周期寄存器
- TMR2 值等同 PR2 时产生中断
- 1:1, 1:4, 1:16 预分频比
- 1:1~1:16 后分频比

图 8.1 为 Timer2 的整体框图。

9.1. Timer2 工作原理

Timer2 模块的时钟输入是系统指令时钟 (FOSC/2)。该时钟送入 Timer2 预分频器，其预分频比有 1:1、1:4 或 1:16 三种选择。随后预分频器的输出被用于递增 TMR2 寄存器。

TMR2 和 PR2 的值被不断比较以确定何时匹配。TMR2 将从 00h 开始递增直到与 PR2 的值相同。匹配时将发生以下两种情况：

- TMR2 在下一递增周期复位为 00h
- Timer2 后分频比递增

Timer2/PR2 比较器的匹配输出送入 Timer2 后分频器。后分频器的选项范围为 1:1 至 1:16。Timer2 后分频器的输出用于将 PIR1 寄存器的 TMR2IF 中断标志置 1。

TMR2 和 PR2 都是可读写寄存器。在复位时，他们的值分别是 0 和 0xFF。

将 T2CON 寄存器中的 TMR2ON 位置 1 可打开 Timer2，反之将 TMR2ON 位清零关闭 Timer2。

Timer2 预分频器由 T2CON 寄存器的 T2CKPS 位控制。

Timer2 后分频器由 T2CON 寄存器的 TOUTPS 位控制。

预分频和后分频计数器会在写以下寄存器时清零：

- 写 TMR2
- 写 T2CON
- 任何 reset 动作

注：

1. 写 T2CON 并不会清零 TMR2 寄存器。

10. 比较器

片内集成 2 个模拟比较器可以用来比较 2 个模拟电压值而产生数字结果反映模拟电压值的高低关系。本芯片中所包含的比较器有以下特性：

- 输出既可以在内部也可以到片外
- 输出极性可编程
- 输出可作为中断源
- 输出可作为唤醒源
- 输入可挂内部参考电压
- 双比较器
- 多种比较器输入输出连接设定模式
- 可编程参考电压

10.1. 比较器概述

模拟比较器的输入输出信号关系以及其在电路图中的标志如图 10.1 所示。当比较器的模拟输入电压 V_{in+} 小于其模拟输入电压 V_{in-} 时，输出为逻辑低电平，反之则输出逻辑高电平。

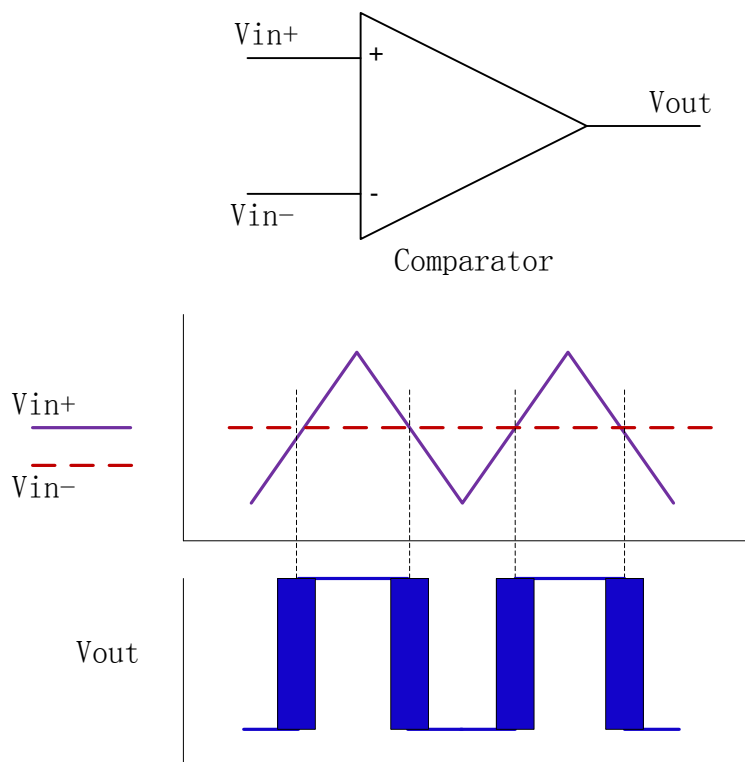
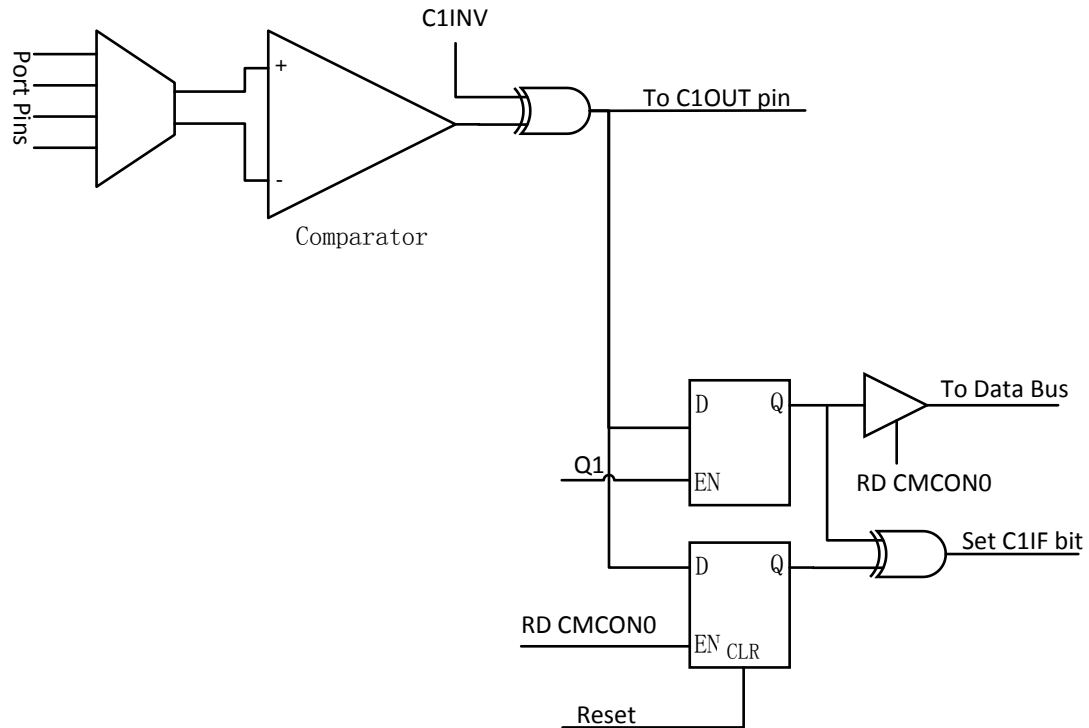


图 10.1 比较器的输入，输出信号特性

当比较器的输入和输出接近相等时，其输出为不定态，因此，为了避免这种不定态，或再某个输入

端有微弱噪声存在，造成比较输入出现不稳定态，一般比较器的设计会保护一个所谓迟滞电压，是比较器的输出态变化会延后输入变化一定的距离，这样基本比较电平稍有波动也不会造成杂散输出信号。

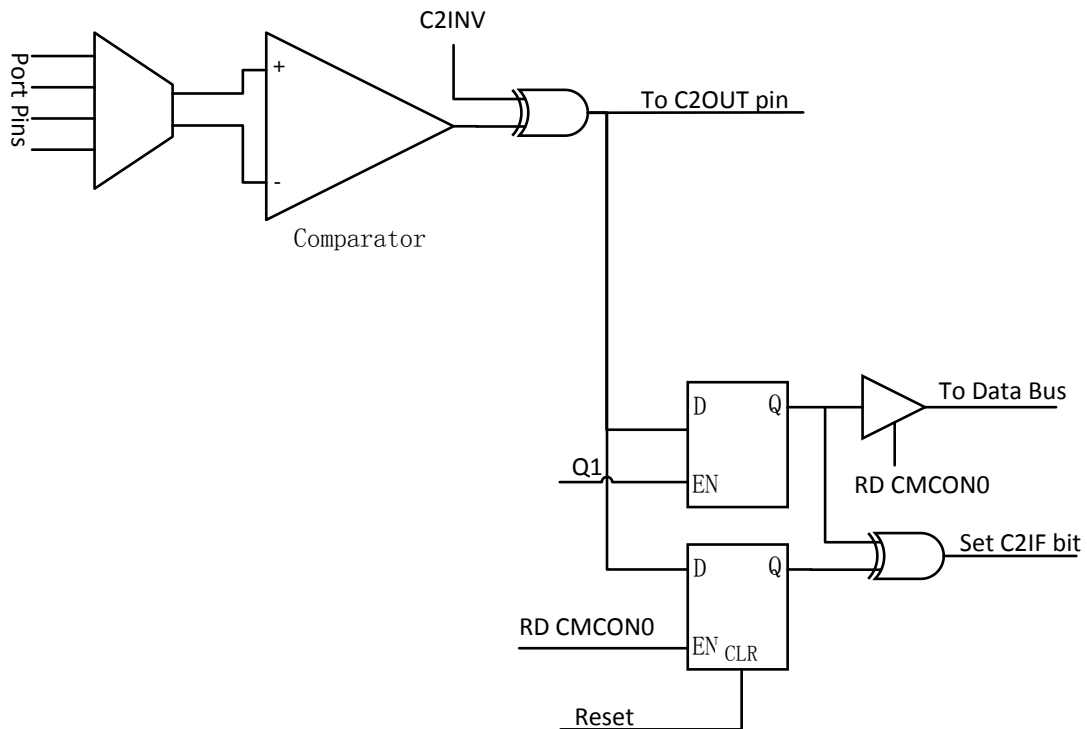
本芯片包含了如下图所示的 2 个比较器，但这 2 个比较器的配置不是独立分开的。



Q1为系统时钟相位。

软件在改变比较器的输入，输出控制寄存器时最好关闭比较器，避免比较器结果错误。

图 10.2 比较器 1 的周边电路



Q1 为系统时钟相位。

软件在改变比较器的输入，输出控制寄存器时最好关闭比较器，避免比较器结果错误。

图 10.3 比较器 2 的周边电路

10.1.1. 模拟输入端的连接

比较器的模拟输入端和同样连接该端口的数字输入共享管脚，需要注意的是在这些管脚上都有对 VDD 和 VSS 的反向保护二极管。如果输入电压偏离这个保护范围以外 0.6V，二极管就会导通到锁闭效应。我们推荐的最大输入源信号的源阻抗不超过 10K 欧姆。另外，如果管脚上外挂电容或齐纳二极管之类的元件，其不应该产生漏电流，不然可能会造成结果不精确。

需要注意的是，当读一个端口寄存器时，如果改管脚被配置为模拟信号管脚，软件会读出 0 值。当管脚被设置为数字输入管脚时，比较器仍然会以为该管脚输入一个模拟信号，并输出相应结果。如果一个管脚被设置为数字输入，而这个管脚上面的实际电压又是一个模拟电平，这可能造成输入缓存电路消耗比说明书上标的更大的电流。

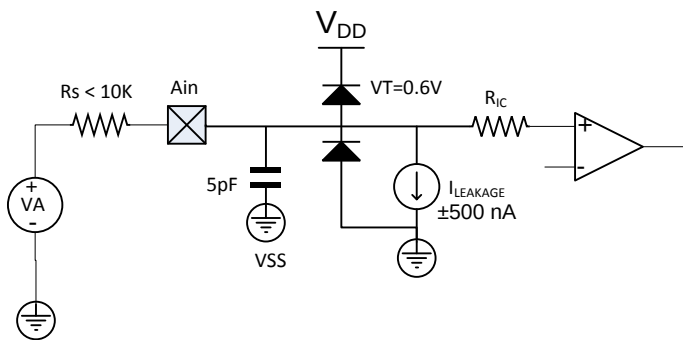


图 10.4 比较器模拟输入模型

10.2. 比较器配置

模拟比较器总共有 8 种配置模式。由 CMCON0 寄存器的 CM<2:0>这 3 位来选择。图 10.5 详细描述了这 8 种选择的具体内容。I/O 信号线的功能也随着这 8 中不同的配置而改变，具体如下：

- 模拟功能 (A)：数字输入缓存被屏蔽
- 数字功能 (D)：比较器数字输出会覆盖管脚上的其他功能
- 正常端口功能 (I/O)：独立于比较器

当端口上标注“A”字样，读的时候无论当前管脚上的状态或 I/O 控制寄存器 TRIS 位的状态都会返回 0 值。用户应该把与被用作模拟输入的管脚相对应的 TRIS 位置为 1 来关闭其数字输出驱动电路。

当端口上标注“D”字样，用户应该将其相应的 TRIS 位职位 0 来打开数字输出驱动电路。

另外，比较器配置切换是应该屏蔽比较器中断以避免不必要的误触发事件。

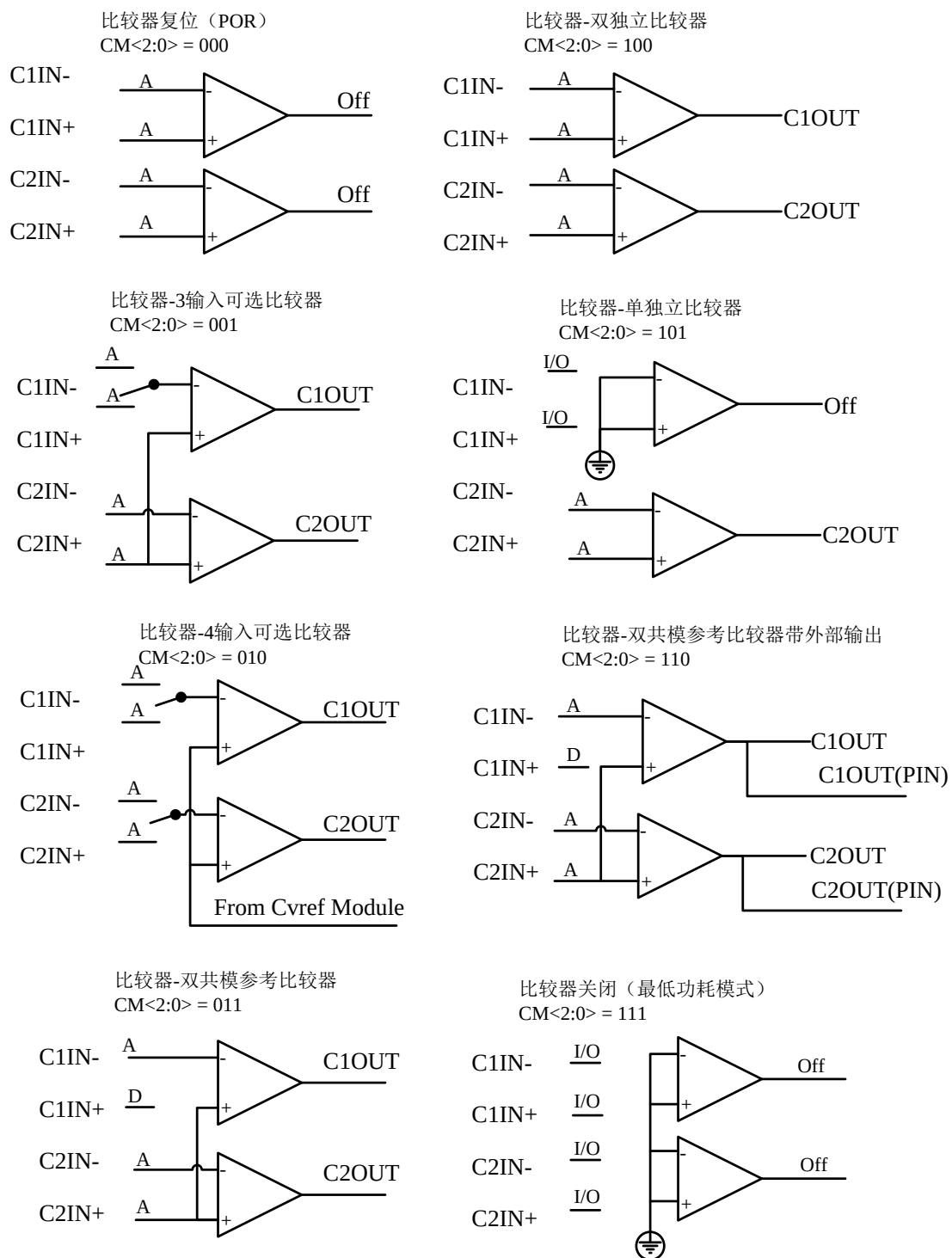


图 10.5 比较器输入，输出配置模式

10.3. 比较器控制

CMCON0 寄存器提供了以下比较器特性的控制功能：

- 模式选择
- 输出状态
- 输出极性
- 输入开关

10.3.1. 模拟输出状态

每个比较器的状态都能通过 CMCON0 寄存器的 CxOUT 位从内部读出。当 CM<2:0>=110 时比较器输出会被连到 CxOUT 管脚。当选择这个模式时，CxOUT 相应的 TRIS 位必须被清零来使能输出驱动电路。

10.3.2. 模拟输出极性

将比较器输出反向就等效于将比较器的输入端交换。比较器输出极性可由 CMCON0 寄存器里的 CxINV 位来控制。将 CxINV 位清零代表比较器输出未被反向。完整的输出，输入条件和极性如下表所列：

Input Condition	CxINV	CxOUT
Vin- > Vin+	0	0
Vin- < Vin+	0	1
Vin- > Vin+	1	1
Vin- < Vin+	1	0

CxOUT 包括寄存器位和实际输出管脚。

10.3.3. 模拟输入开关

比较器的模拟输入负端在以下模式可被切换到 2 个模拟管脚。

- CM<2:0> = 001 (仅比较器 1)
- CM<2:0> = 010 (比较器 1 和 2)

在以上模式下，无论哪根管脚被选做输入，这 2 个管脚都处在模拟模式。CMCON0 寄存器里的 CIS 位控制比较器的输入切换开关。

10.4. 比较器反应时间

模拟比较器输出在输入变化或输入新的参考电压后若干时间后才会变化，这个时间叫做比较器反应

时间。反应时间和参考电压的稳定时间还不是同一个概念。这两者加起来才是比较器的总体反应时间。具体参数，请参考本文档后面的电气参数部分。

10.5. 比较器中断

一旦使能，模拟比较器输出状态的变化就会触发中断。输出状态变化的捕捉用两个锁门器和一个异或门完成。（见图 10-2, 10-3）。当读取 CMCON0 寄存器时，一个锁门器由比较器输出更新。该锁门器保持着状态直到 CMCON0 下一次被读或复位。另一个锁门器在系统时钟的 Q1 期更新。状态失配状态会被一直保存，并使 PIR1 寄存器的 CxIF 位为 1，直到 CMCON0 被读或比较器输出返回到以前的状态。（注，对 CMCON0 寄存器的写动作也会清除失配条件，应为所有的写动作实际上在写之前都包含着一个读动作）。

软件需要自身保存比较器的输出态来决定实际上变化是否有发生。PIR1 寄存器里的 CxIF 位是比较器的中断指示位。这 1 位必须有软件来清零。由于软件可以将这 1 位写成 1，因此软件可以做模拟比较器变化的动作。

比较器的中断发生还需要 INTCON 寄存器中的 PEIE 位和 GIE 位联合搭配工作。如果这其中任何 1 位没有置 1，哪怕 CxIF 位变成 1 了，中断都不会发生。用户可以通过以下动作清除中断：

- A) 对 CMCON0 进行读或写操作，这会结束失配条件
- B) 清除 CxIF 位

持续的失配条件会屏蔽住 CxIF 标志位被清，因此有必要先读一下 CMCON0 寄存器先清除失配条件。

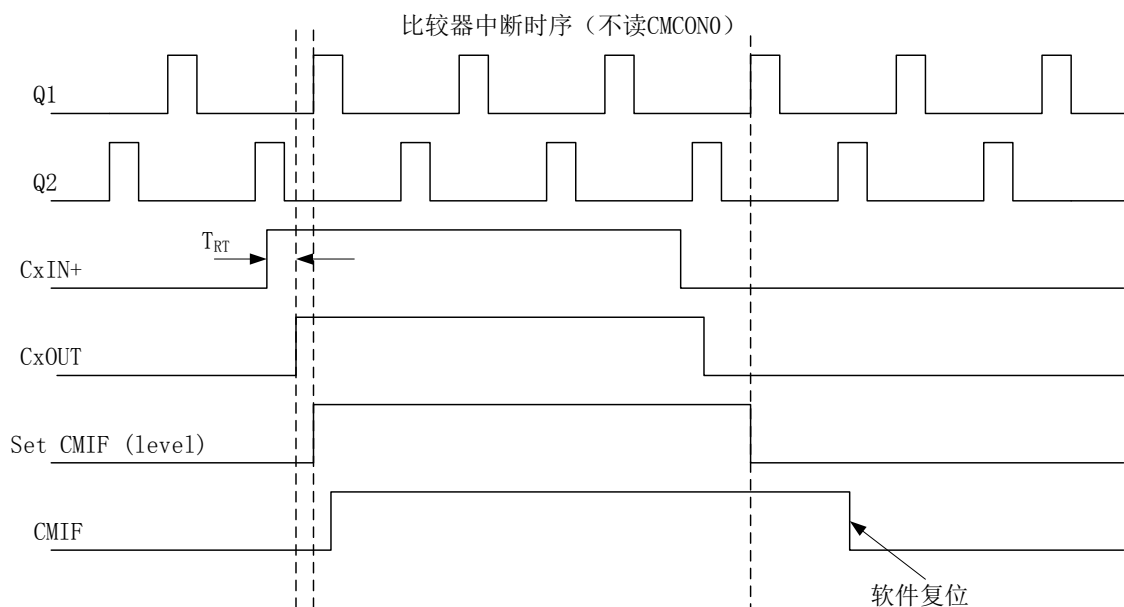


图 10.6 比较器中断时序 1

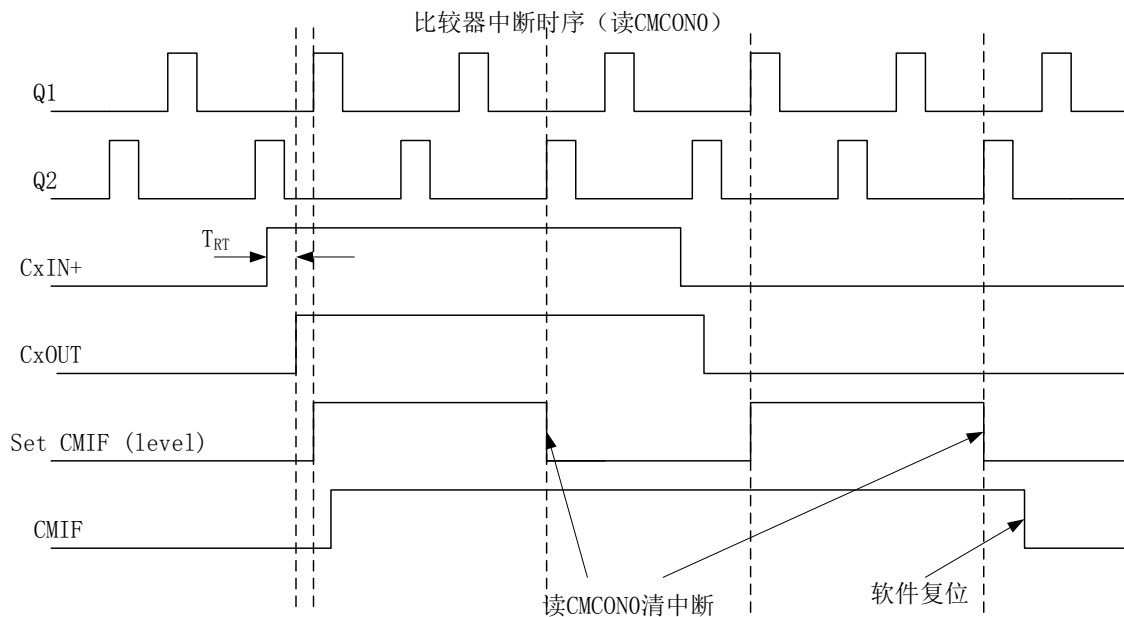


图 10.7 比较器中断时序 2

需要注意的是如果在读 CMCON0 的时候(在系统时钟的 Q2 期), CxOUT 刚好发生变化, 那么 CxIF 标志位有可能不会被置高。

另外, 比较器电路被启动时, 其偏置电路需要 1 μ S 的时间稳定, 在此期间, 比较器的输出是无效的, 软件应该在此期间关闭中断以免产生不必要的误触发。

10.6. 比较器在睡眠状态下的工作

如果用户在使芯片进入睡眠之前使能比较器, 那么比较器会在睡眠状态下继续工作。但此时芯片的功耗会相应增加。如果用户并不使用比较器输出来唤醒芯片, 用户可以在睡眠前关闭比较器。关闭比较器只要在 CMCON0 寄存器中将 CM<2:0>置为 000 或 111 即可; 如果要达到最低功耗, 则应该把 CM<2:0>置为 111, 彻底关闭两个比较器。

如上所述, 比较器的输出可以用来唤醒芯片。如果需要用比较器输出唤醒芯片, 需要在 PIE1 寄存器中将 CxIE 位置 1, 以及 INTCON 寄存器下的 PEIE 位置 1. 如果 INTCON 寄存器的 GIE 位被置 1, 器件就会执行中断服务程序。

10.7. 比较器在复位状态下的工作

芯片复位会强制 CMCON0 和 CMCON1 寄存器进入复位状态, 也就是 CM<2:0>=000 模式。此时, 所有比较器输入都为模拟输入, 比较器本身被关闭以节省功耗。

比较器 5 门控 Timer1

可使用此功能为模拟事件的持续时间或间隔定时。将 CMCON1 寄存器的 T1GSS 位清零将使 Timer1 根据比较器 5 的输出递增。这要求 Timer1 打开且门控使能。

详情请参见第 7 节“[带门控的 Timer1 模块](#)”。

当比较器用作 Timer1 门控源时，建议通过将 C2SYNC 位置 1 将比较器 5 与 Timer1 同步。这将确保 Timer1 在递增时若比较器变化，Timer1 不会错过递增。

比较器 5 输出与 Timer1 同步

通过将 CMCON1 寄存器的 C2SYNC 位置 1 可将比较器 5 的输出与 Timer1 同步。使能时，比较器输出在 Timer1 时钟源的下降沿被锁存。Timer1 使用预分频器时，比较器输出在预分频后被锁存。为了防止竞争情况，比较器输出在 Timer1 时钟源下降沿被锁存，而 Timer1 在时钟源的上升沿递增。更多信息，请参见比较器框图（图 10.6）以及 Timer1 框图（图 7.1）。

10.8. 比较器的参考电压

模拟比较器的输入之一可以被设置成为内部参考电压，该参考电压都有以下特性：

- 独立于比较器的运行
- 两个 16 级电压范围
- 输出低钳位到 VSS
- 输出电压与 VDD 成比例

参考电压的控制由 VRCON 寄存器来控制，如图 10.8。

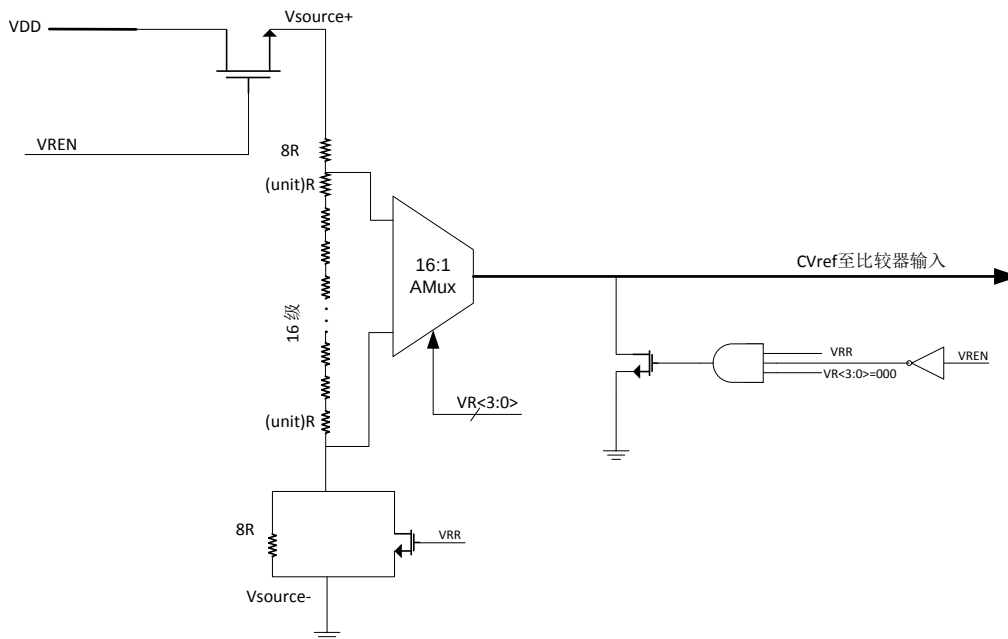


图 10.8 比较器参考电压框图

独立工作

比较器参考电压与比较器配置不相关。将 VRCON 寄存器的 VREN 位将使能参考电压。

输出电压选择

CVREF 参考电压有两种范围，每种均为 16

VRR = 1 (低电压范围):

$$CVREF = (VR<3:0>/24) \times VDD$$

VRR = 0 (高电压范围):

$$CVREF = (VDD/4) + (VR<3:0> \times VDD/32)$$

个电平。范围的选择由 VRCON 寄存器的 VRR 位控制。这 16 个电平由 VRCON 寄存器的 VR<3:0>位设置。

CVREF 输出电压由以下右边公式确定：

如图 10.8 所示，由于模块的构造所限，无法实现 VSS 至 VDD 的满量程。

输出钳位至 VSS

通过将 VRCON 按如下配置可将 CVREF 输出电压设置为 VSS，从而不消耗功率：

- VREN = 0
- VRR = 1
- VR<3:0> = 0000

这使比较器可进行过零检测而不消耗额外的 CVREF 模块电流。

11. 数据 EEPROM

片内集成有 256 个字节的 EEPROM, 通过 EEADR 进行寻址访问。软件可通过 EECON1 和 EECON2 对 EEPROM 进行编程操作, 硬件实现了擦除和编程的自定时功能, 无需软件查询, 节省有限的代码空间, 同时利用此特性, 启动编程周期之后可以进入睡眠模式, 以降低功耗。

数据 EEPROM 在使用（无论是读还是写）之前必须进行以下初始化操作：在未使用到的 EEPROM 某个单元写两次 0xAA，后续程序不要对此单元操作。如：

```
SYSTEM_INIT:
.....
.....
LDWI 0x55
STR EEPROM_ADDR
LDWI 0xAA
STR EEPROM_DATA
LCALL EEPROM_write
LCALL EEPROM_write
.....
```

11.1. 编程数据 EEPROM 步骤

- A. 把 INTCON 的 GIE 位清 0;
- B. 判断 GIE 是否为 1, 是则重复 A 步骤, 否则可以进行下一步;
- C. 往 EEADR 写入目标地址;
- D. 往 EEDAT 写入目标数据;
- E. 把位 WREN3/WREN2/WREN1 全部置 1;
- F. 把位 WR 置 1 (EECON2.0, 此后 WR 会维持高);
- G. 写过程不能改变 WREN3/2/1 的值, 否则编程终止;
- H. 等大概 2ms 之后编程自动完成, WR 自动清 0, WREN3、WREN2、WREN1 清 0;
- I. 如果想再次编程, 重复步骤 C~H 即可;

注意:

1. 编程过程中读操作无效。

11.2. 读数据 EEPROM

要读取数据存储单元，用户必须将地址写入 **EEADR** 寄存器，然后将 **EECON1** 寄存器的控制位 **RD** 置 **1**。在紧接着的下一周期，**EEDAT** 寄存器就被 **EEPROM** 数据写入。因此该数据可由下一条指令读取。**EEDAT** 将保持这个值直到用户下一次从该单元读取或向该单元写入数据时（在写操作过程中）。

下面是读取 **EEPROM** 的一段示例程序：

```
BANKSEL EEADR  
LDWI dest_addr  
STR EEADR  
BSR EECON1, RD  
LDR EEDAT, W
```

12. 时钟测量

12.1. 慢时钟测量模式

此功能可以比较精准的测量内部慢时钟周期。

在此模式下，TIMER2 的预分频、后分频配置自动变为 1:1，组成一个 12 位的定时器，TIMER2 的计数时钟为系统时钟 F_{osc} ，而不是普通模式下的指令时钟 $F_{osc}/2$ 。计数结束后结果自动存到 SOSCPR 寄存器，其单位是系统时钟 F_{osc} 的个数。

操作步骤：

1. 为提高计量精度，建议设置 IRCF 为 111，SCS=1，选择 16M 的系统时钟；
2. 把 T2CON.2 置 1，使能 TIMER2；
3. 如果选择 4 次平均，则把 MSCKCON.2 置 1，否则把它清 0；
4. 置位 MSCKCON.1，开始测量；
5. 测量结束后 MSCKCON.1 自动清 0，中断标志置 1；
6. 可以用查询或中断的方式等待结束；
7. 当查询到中断标志为 1 时读取得到的 SOSCPR 即为最终结果。

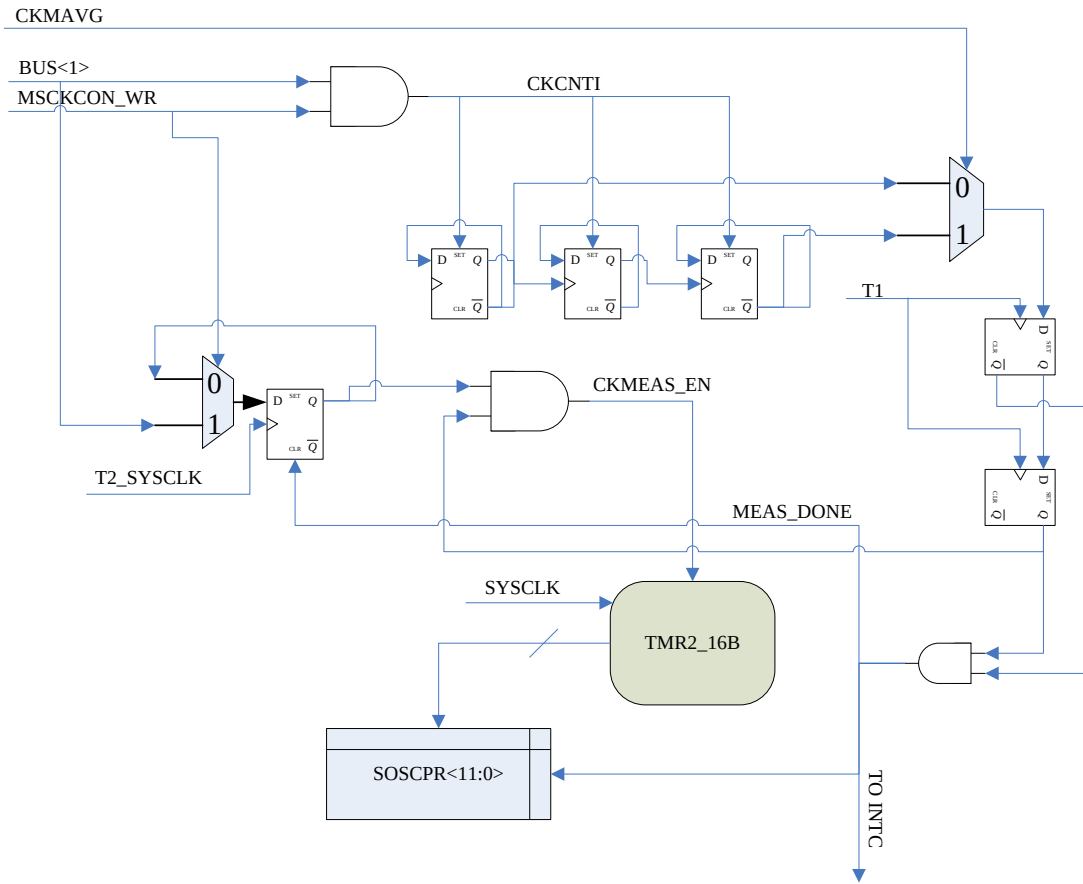


图 13.5 慢时钟测量模式原理框图

13. 中断模式

TX-I-02X 有以下中断源：

- PA2/INT 管脚进来的外部中断
- Timer0 溢出中断
- PORTA 变化中断
- Timer2 比对相等中断
- EEPROM 数据写中断
- 故障保护时钟监控器中断
- 比较器中断

中断控制寄存器（INTCON）和外围中断请求寄存器（PIR1）记录了中端标志位。INTCON 同时也包含全局中断使能位 GIE。

当中断被服务后，以下动作自动发生：

- GIE 被清零，从而关闭中断
- 返回地址被推上堆栈
- 程序指针被加载 0004h 地址

中断返回指令，RETFIE，退出中断函数时同时设置 GIE 位，重新使能未屏蔽的中断。

INTCON 寄存器包含以下中断标志位：

- INT 管脚中断
- PORTA 变化中断
- Timer0 溢出中断

PIR1 中包含着外围中断标志位。PIE1 中包含着其对应的中断使能位。

13.1. INT 中断

INT 引脚上的外部中断是边沿触发的；当 OPTION 寄存器的 INTEDG 位被置 1 时在上升沿触发，而当 INTEDG 位被清零时在下降沿触发。当 INT 引脚上出现有效边沿时，INTCON 寄存器的 INTF 位置 1。可以通过将 INTCON 寄存器的 INTE 控制位清零来禁止该中断。在重新允许该中断前，必须在中断服务程序中先用软件将 INTF 位清零。如果 INTE 位在进入休眠状态前被置 1，则 INT 中断能将处理器从休眠状态唤醒。

注意：

1. 使用 INT 中断时，必须对 ANSEL 和 CM2CON0 寄存器进行初始化，以将模拟通道配置为数字输入。配置为模拟输入的引脚总是读为 0。

13.2. PORTA 电平变化中断

PORTA 输入电平的变化会使 INTCON 寄存器的 PAIF 位置 1。可以通过置 1/清零 INTCON 寄存器的 PAIE 位来使能/禁止该中断。此外，可通过 IOCA 寄存器对该端口的各个引脚进行配置。

注意：

1. 使用 PORTA 电平变化中断时，必须对 ANSEL 和 CM2CON0 寄存器进行初始化，以将模拟通道配置为数字输入。配置为模拟输入的引脚总是读为 0。
2. 初始化电平变化中断时，应先配置为数字输入 IO，把相应的 IOCA 置 1，然后读取一下该 PORTA；
3. 当 IO 电平发生变化时，PAIF 被置 1；
4. 清中断标志位之前应该读取一下 PORTA，然后再对 PAIF 清 0；

13.3. 中断响应

外部中断包括 INT 管脚进来的或者 PORTA 变化中断的延时一般为 1 到 2 个指令周期。具体视中断发生的实际情况而定。

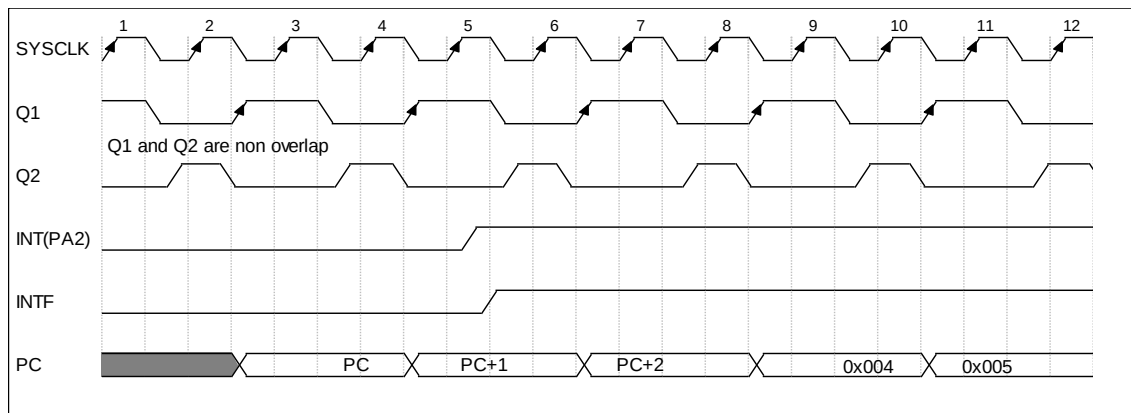


图 14.1 中断响应时序图

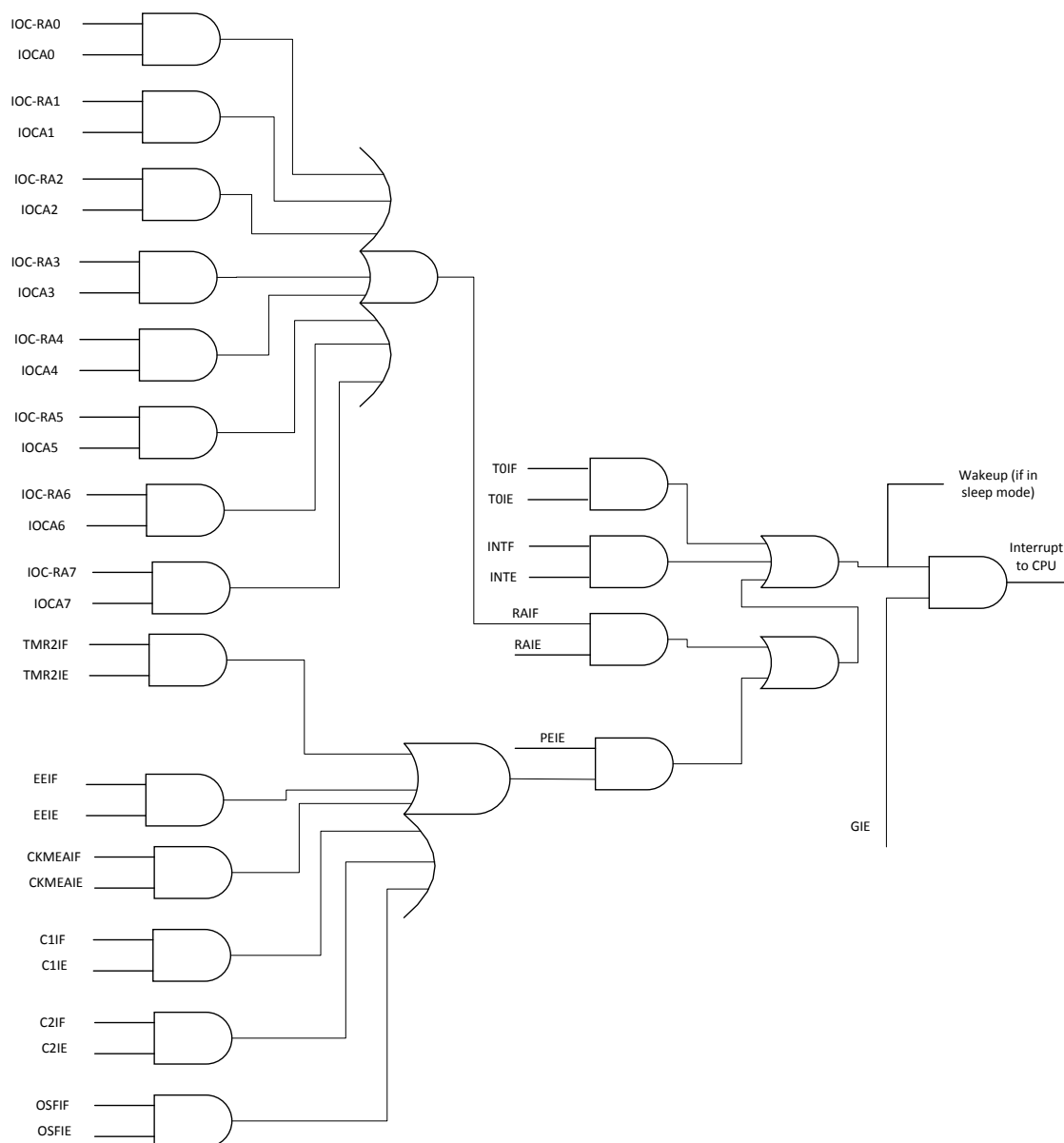


图 14.2 中断产生电路架构框图

13. 4. 中断过程中的现场保存

在中断过程中，只有返回 PC 被自动保存在堆栈上。一般来说，用户可能需要保存重要的寄存器值在堆栈上，例如 W，STATUS 寄存器等。这些必须由软件来完成。临时寄存器 W_TEMP 和 STATUS_TEMP 应该被放置在 GPR 的最后 16byte 里。这 16 个 GPR 落在两个页区间，因此可以稍微节省代码。

14.睡眠省电模式

芯片在执行完 SLEEP 指令后进入睡眠状态。

为了达到最低睡眠功耗，软件应该将所有 IO 置高或低，而且没有外部电路从 IO 耗电。I/O 作为输入的，外部电路应将其拉高或拉低，避免翻转耗电。/MCLR 应该在高电平。

为达到最低功耗，建议配置为晶体模式或者外部时钟模式时，把时钟缺失检测关闭掉，即把 UCFG1 的 FCMEN 位清 0，同时比较器的配置位 CM<2:0>写为 111，以关闭比较器模块。

14.1. 唤醒模式

以下事件可以唤醒芯片：

- /MCLR 管脚上有外部复位
- WDT 超时
- PA2/INT 管脚上有中断，PORTA 变化或其他外围中断

清看门狗指令 CLRWDT、SLEEP（进入睡眠模式）或者从睡眠模式唤醒，都将清除看门狗计数器。

14.2. 看门狗唤醒

看门狗工作在内部慢时钟（32KHz），它是一个 16 位的计数器，和定时器 0 共用一个 8 位的预分频器，使能位位于配置寄存器 UCFG0 的第 3 位，WDTEN，为 1 时表示使能看门狗，为 0 时将由 SWDTEN 位决定使能与否，SWDTEN 位于 WDTCON 寄存器。

清看门狗指令 CLRWDT、SLEEP 会清除看门狗计数器。

在使能了看门狗的情况下，MCU 睡眠时看门狗溢出事件可以作为一个唤醒源，而 MCU 正常工作时作为一个复位源。

15.I/O 端口

本芯片共包含 16 个 GPIO。这些 IO 除了作为普通输入/输出端口以外还通常具备一些与内核周边电路通讯的功能，具体见下。

15.1. PORTA 端口和 TRISA 寄存器

PORTA 是一个 8 位双向端口。与其相应的进出方向寄存器就是 TRISA 寄存器。(不过注意到这里第 5 位没有用到，因为 PORTA[5]为单输入方向端口。在 TRISA 寄存器中将某一位设置为“1”会将该对应 PORTA 端口设置为输入端口(此时，输出驱动电路会被关断)。反之，将某一位设置为“0”会将该对应 PORTA 端口设置为输出端口。在置为输出端口时，输出驱动电路会被打开，输出寄存器里的数据会被放置到输出端口。在 PORTA 上进行读动作时，PORTA 内容会是反映输入端口的状态。在 PORTA 上进行写动作时，PORTA 内容会被写入输出寄存器。所有的写操作都是“读-更改-写”这样一个微流程，即数据被读，然后更改，再写入输出寄存器的过程。当 MCLRE 为 1 时，PORTA[5]读的值为 0，此时它是作为外部复位管脚。

15.2. 端口的其他功能

芯片在 PORTA 的每个端口都有一个状态变化中断选项和弱上拉选项。

15.2.1. 弱上拉

PORTA 的每个端口（除了 PORTA[5]）都有一个可以单独设置的内部弱上拉功能。控制 WPUAx 寄存器里的位就可使能或关断这些弱上拉电路。当 GPIO 被设置为输出时，这些弱上拉电路会被自动关断。弱上拉电路在上电复位期间可以被置为关断。这是由 OPTION 寄存器中的/PAPU 位决定的。PORTA[5]内部也有弱上拉功能，它是在将 PORTA[5]设置为/MCLR 功能时自动使能的。当 PORTA[5]被设置为 GPIO 时，该弱上拉电路被自动关断。

15.2.2. 状态变化中断

PORTA 的每个端口都可以被单独设置成一个中断源(端口状态变化触发中断)。控制 IOCAx 寄存器里的位就可使能或关断这些端口的中断功能。端口状态变化触发中断的功能在上电复位时无效的。

当端口状态变化触发中断的功能被使能时，当前端口电平值会被与上次读动作所读取数据寄存器的旧值作对比。所有错误匹配结果会被或在一起形成中断标志位 INTCON 寄存器中的 PAIF 标志位。

该中断可以将芯片从睡眠状态中唤醒。用户需要在中断服务程序中执行以下程序来清除该标志位：

- A) 对 PORTA 进行一次读或写得动作，这将结束任何错误匹配的状态。
- B) 清零 PAIF 标志位。

错误匹配的条件会一直设置 PAIF 位。对 PORTA 做一次读就可以结束任何错误匹配的状态，使得 PAIF 能被清零。数据寄存器里保持的上一次读的值不会被/MCLR 或低电压复位所影响。只要错误匹配状态存在，PAIF 位就会被置 1。

15.2.3. 端口描述

PORTA 的每个端口都包含着不同的复用功能。其具体功能和控制在这一节里描述。

PORTA[0]

图 16.1 描述了此端口的内部电路结构。PA[0]可以被配置为以下功能端口：

- GPIO
- 调试串口时钟

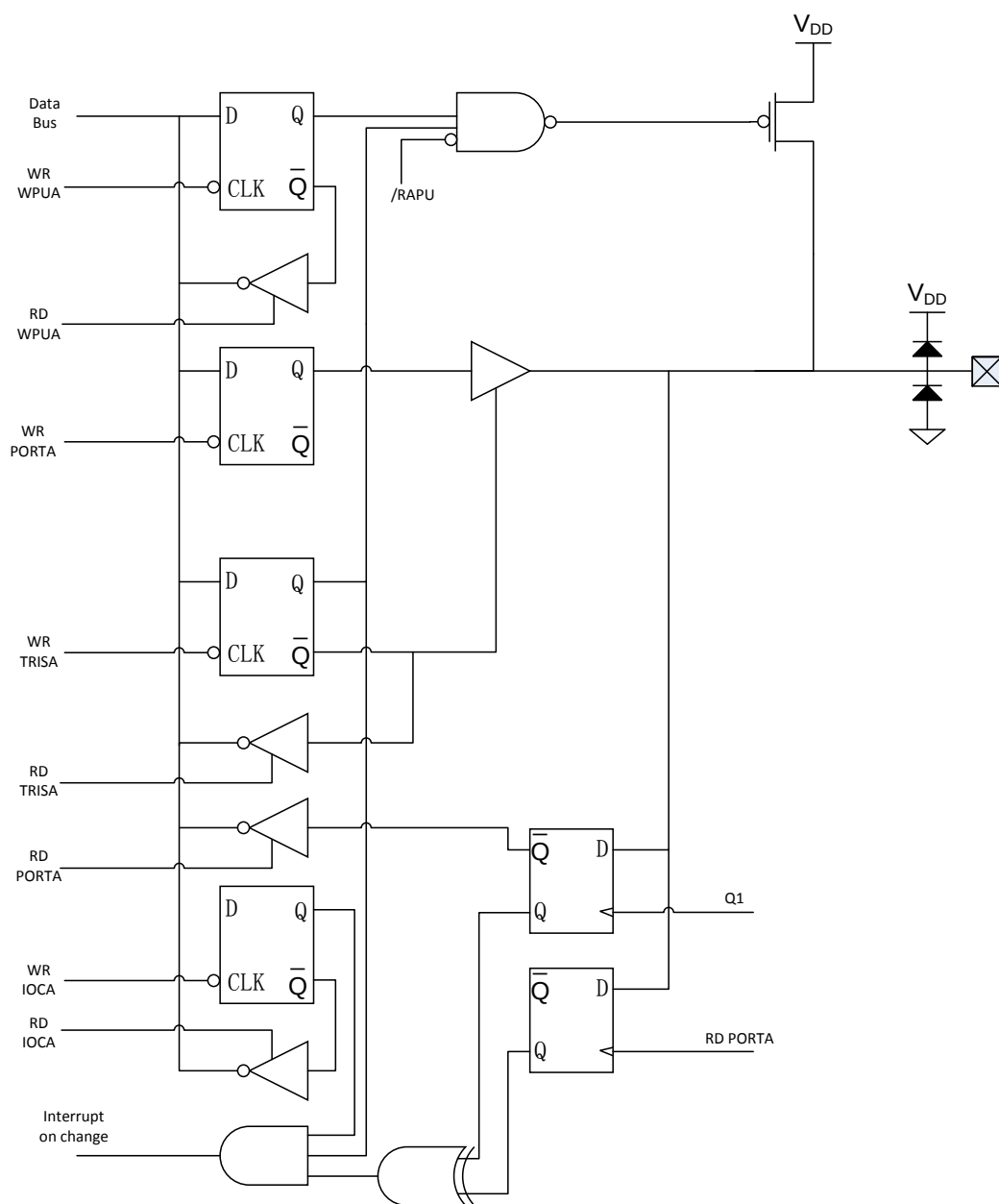


图 16.1 PA0 架构框图

PORTA[1]

图 16.2 描述了此端口的内部电路结构。PA[1]可以被配置为以下功能端口：

- GPIO
- 调试串口数据

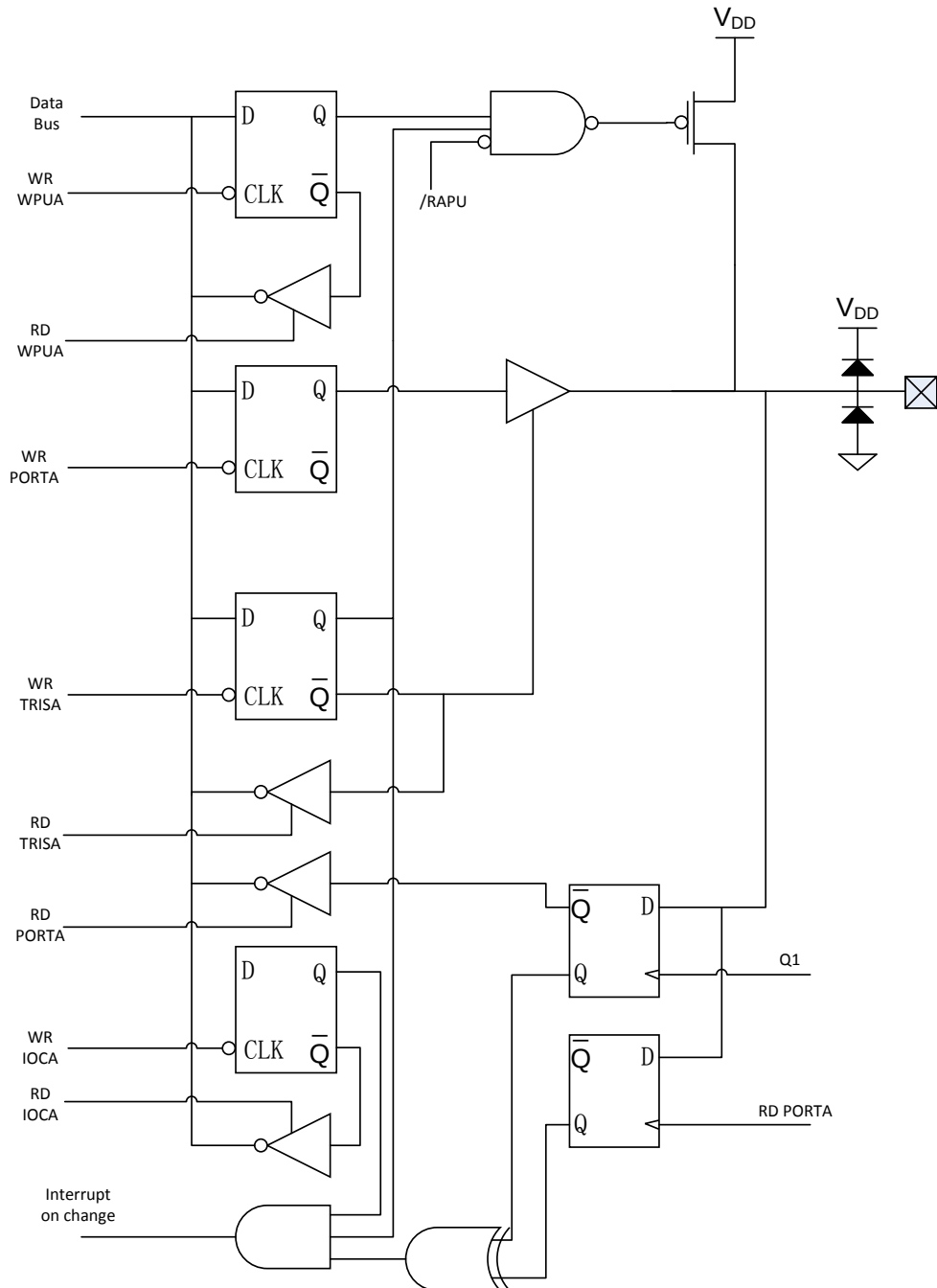


图 16.2 PA1 架构框图

PORTA[2]

图 16.3 描述了此端口的内部电路结构。PA[2]可以被配置为以下功能端口：

- GPIO
- 外部中断输入
- TIMER0 外部时钟源

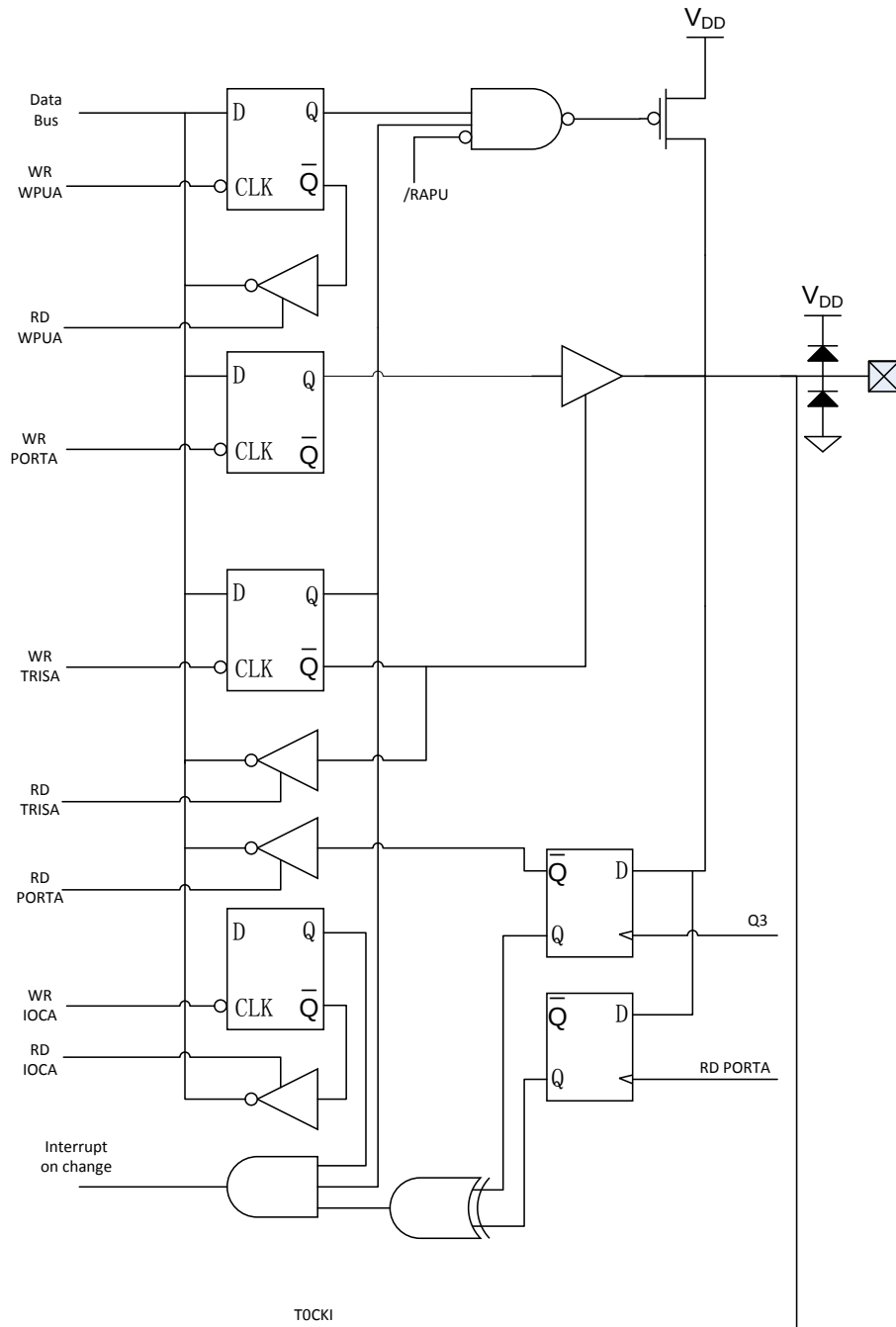


图 16.3 PA2 架构框图

PORTA[3]

图 16.4 描述了此端口的内部电路结构。PA[3]可以被配置为以下功能端口：

- GPIO
- 模拟测试输出

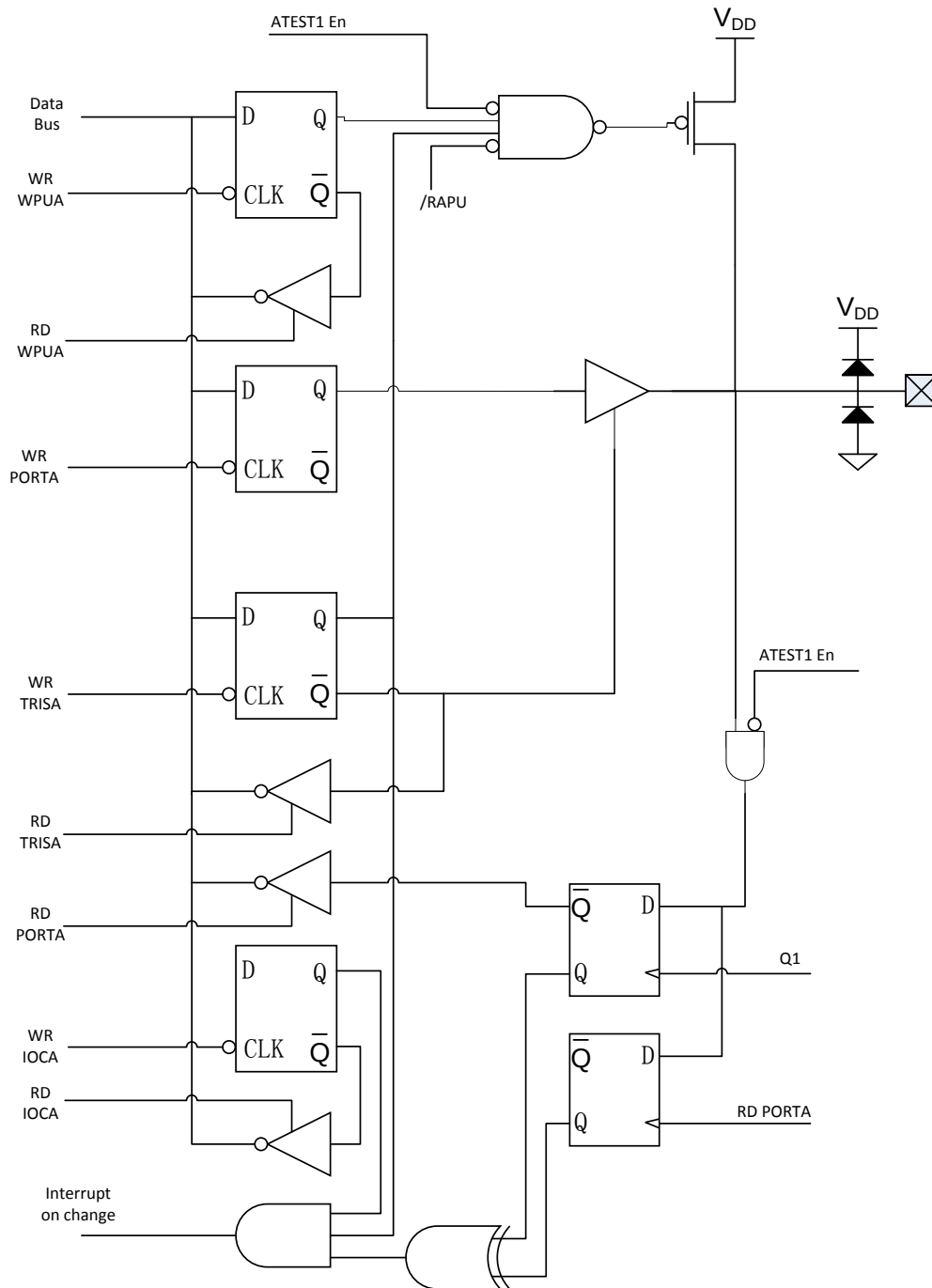


图 16.4 PA3 架构框图

PORTA[4]

图 16.5 描述了此端口的内部电路结构。PA[4]可以被配置为以下功能端口：

- [illegible]

page78

-
- The diagram shows a microcontroller system with the following components and connections:
- Microcontroller:**
 - Data Bus:** Connected to the Data Bus.
 - RD TRISA:** Connected to the Data Bus.
 - RD PORTA:** Connected to the Data Bus.
 - WR IOCA:** Connected to the Data Bus.
 - RD IOCA:** Connected to the Data Bus.
 - Interrupt on change:** Connected to the Data Bus.
 - 16C45 Timer Module:**
 - CLK:** Connected to the Data Bus.
 - Q:** Connected to the Data Bus.
 - Q̄:** Connected to the Data Bus.
 - External Components:**
 - MCLR:** Connected to the Data Bus.
 - Reset:** Connected to the Data Bus.
 - Weak:** Connected to the Data Bus.
 - Q1:** Connected to the Data Bus.
 - RD PORTA:** Connected to the Data Bus.

page79

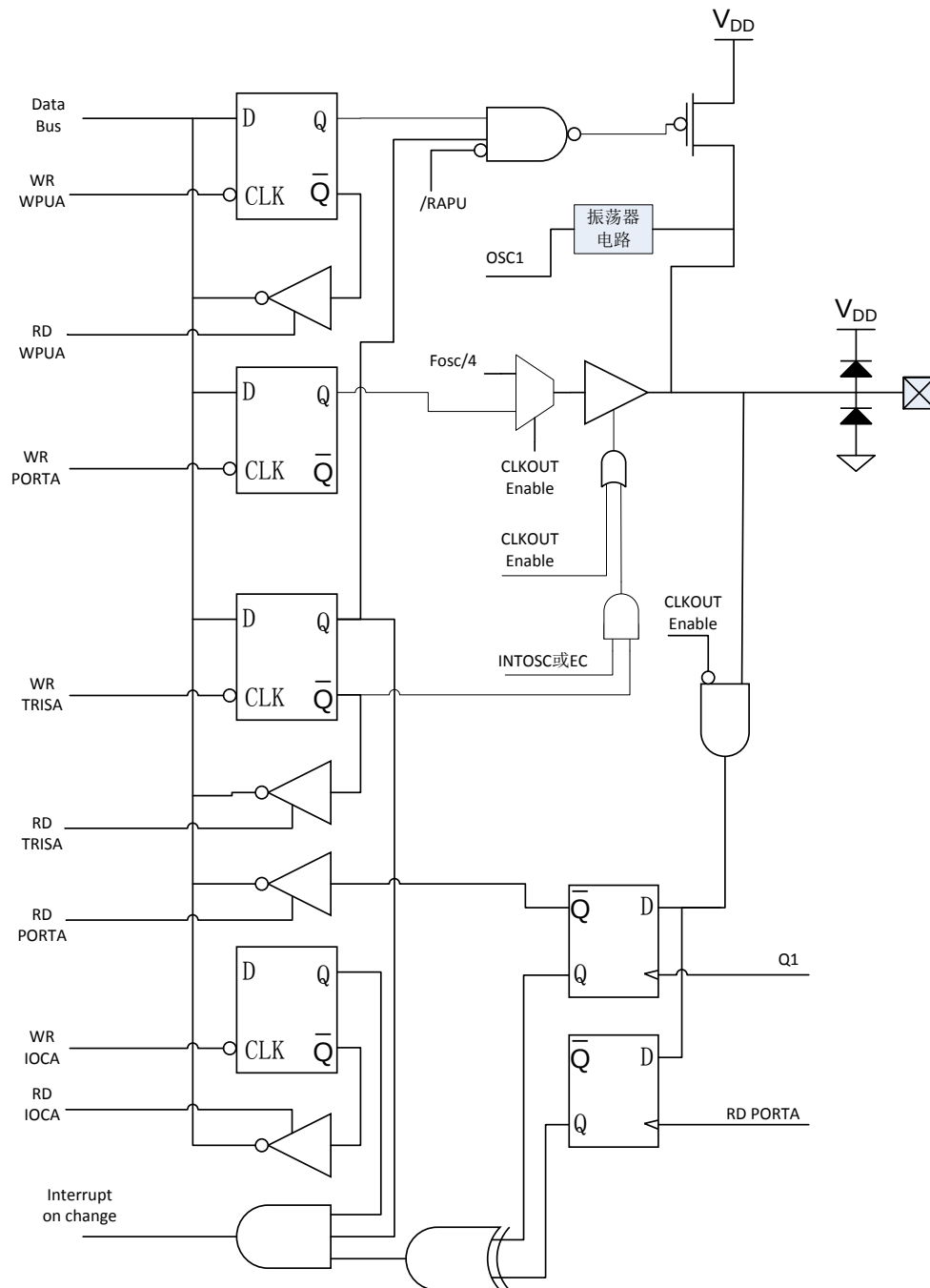


图 16.7 PA[6]架构框图

PORTA[7]

图 16.8 描述了此端口内部电路结构。PA[7]可以配置为以下功能端口：

- GPIO
- 晶振、谐振器连接
- 时钟输入

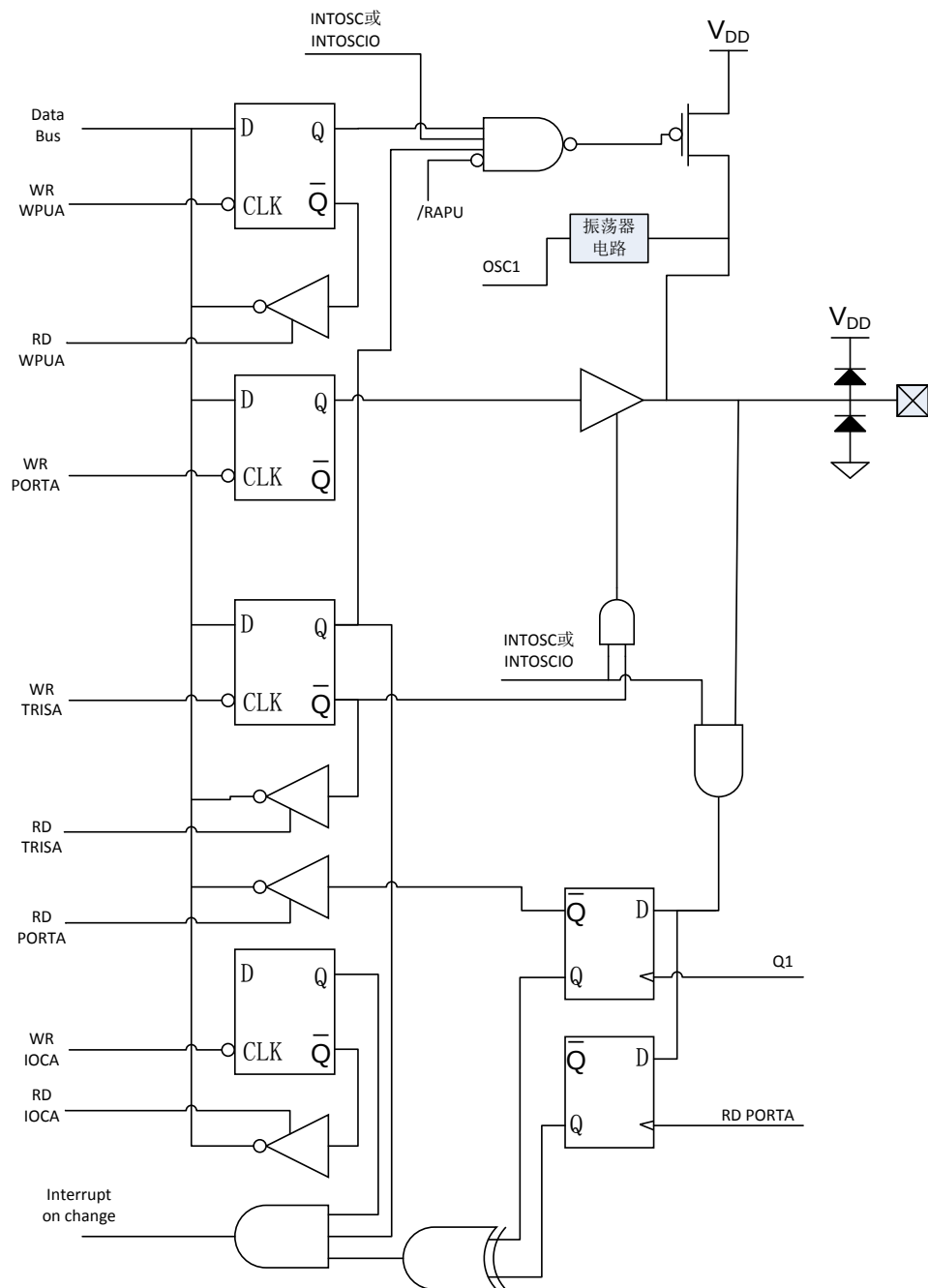


图 16.8 PA[7]架构框图

PORTC[0], [1]

图 16.9 描述了此端口的内部电路结构。PORTC[0]/[1]可以被配置为以下功能端口：

- GPIO
- 比较器输入

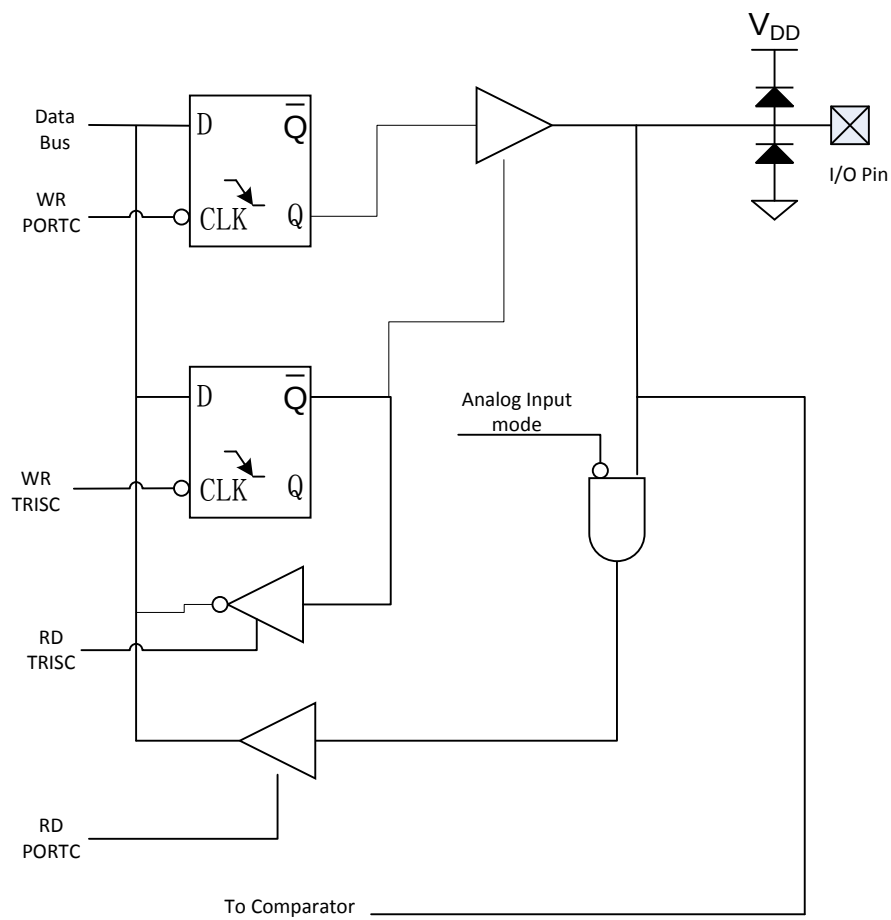


图 16.9 PC0,1 架构框图

PORTC[2], [3]

图 16.10 描述于此端口的内部电路结构。PORTC[2]/[3]可以被配置为以下功能端口：

- GPIO

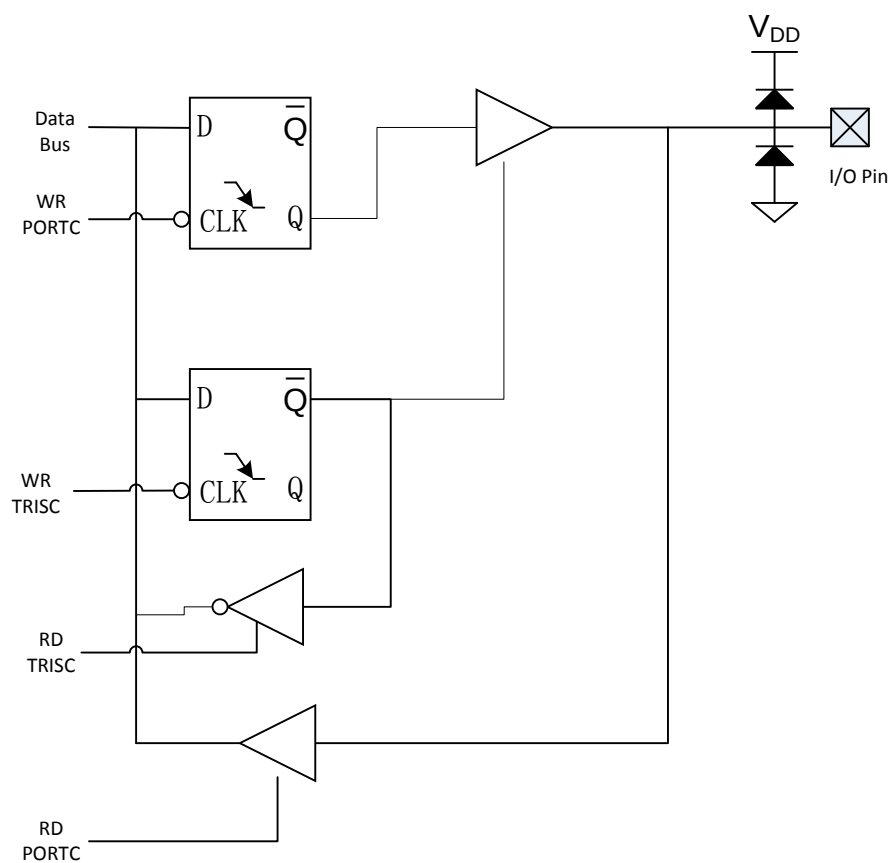


图 16.10 PC2,3 架构框图

PORTC[4]

图 16.11 描述了此端口的内部电路结构。PORTC[4]可以被配置为以下功能端口：

- GPIO

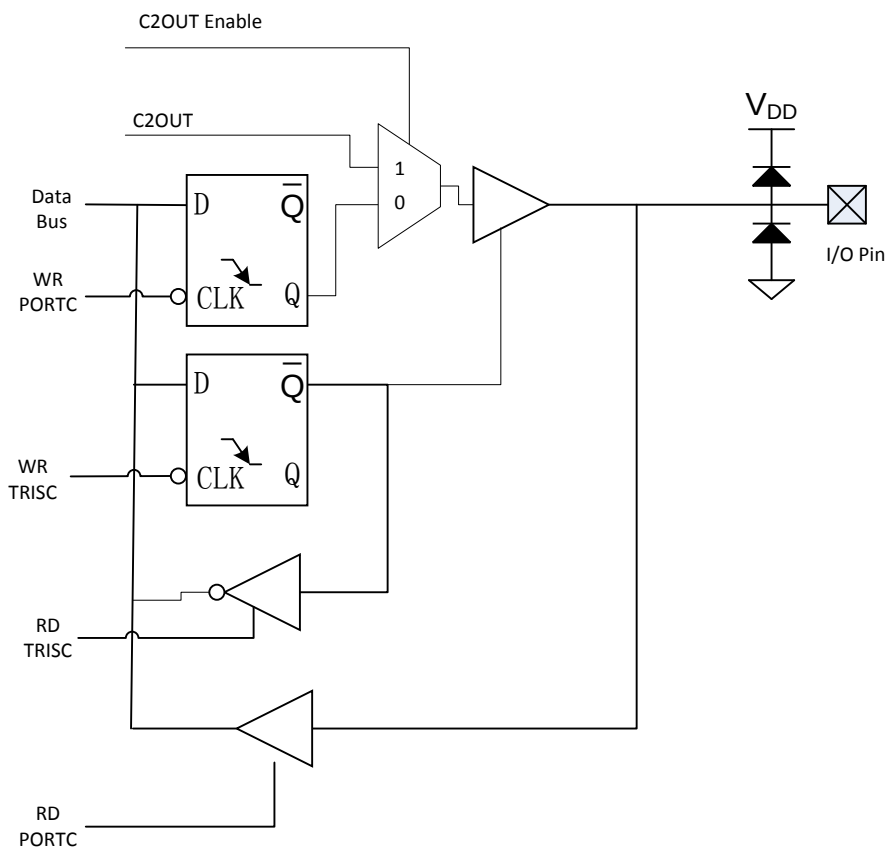


图 16.11 PC4 架构框图

PORTC[5], [6], [7]

图 16.12 描述于此端口的内部电路结构。PORTC[5]/ [6]/ [7]可以被配置为以下功能端口：

- GPIO

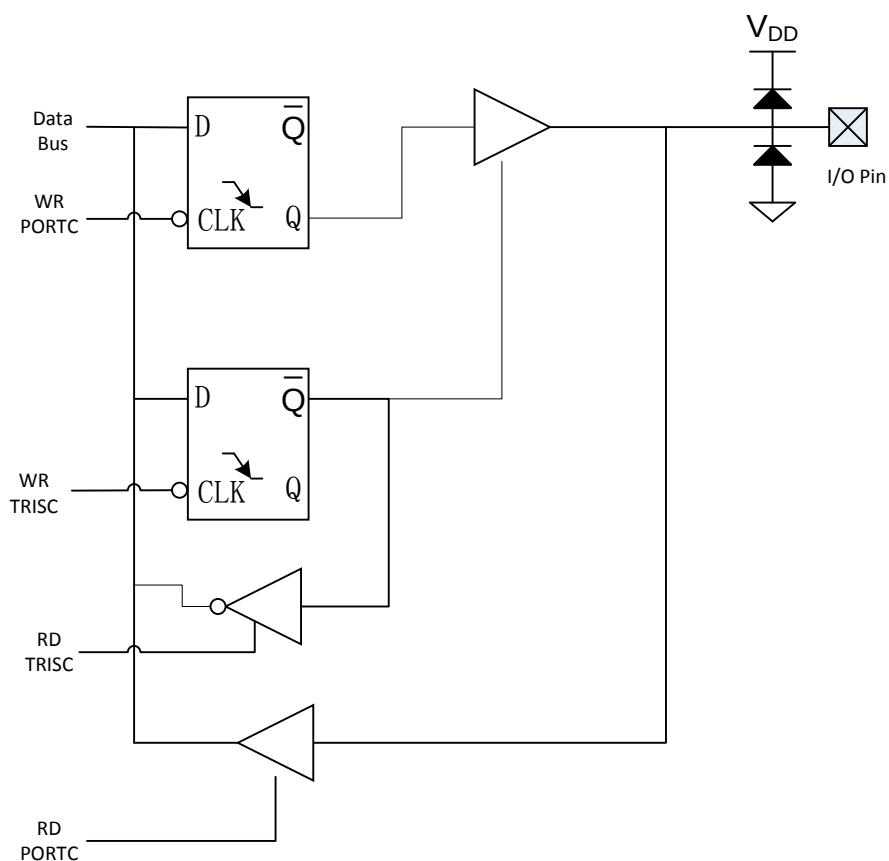


图 16.12 PC5,6,7 架构框图

16. 芯片的电气特性

16.1. 内置高频振荡器 (Internal High Frequency Osc)

电气参数	最小值	典型	最大值	单位	条件/备注
Ivdd 工作电流	Osc: 12 Bgr: 6.1	Osc: 17 Bgr: 9		uA	1.6~5.5v 27°C tt trim set 100000
随温度变化范围		4.2%/100°C			-20~80 °C, tt 3.3v
随电源电压变化范围		+/-3%			2 ~ 5.5V, tt
标准振荡频率	---	15.99	---	MHz	3.3V, 27°C, tt
启动时间	---	2.2	10	uS	
漏电流	---	0.8	2	nA	Disable fosc&bgr
校准范围	---	+/-20%	---		Step 0.625%
	---	+/-30%	---		Step 0.9375%

16.2.

16.3. 内置低频振荡器 (Internal Low Frequency Osc)

此低频振荡器有双模模式，一种模式下振动频率为 32KHz，另一种模式下振动频率为 256KHz。振荡频率模式由 OSCCON 寄存器中的 LFMOD 位控制，0 为 32KHz 模式，1 为 256KHz 模式

电气参数	最小值	典型	最大值	单位	条件/备注
Ivdd 工作电流	0.4	1.1	1.4	uA	1.6~5.5V, tt
随温度变化范围		22.3%/100°C			-20 ~ 80 °C 2.5V
随电源电压变化范围		11.1%/V			2 ~ 5.5V @27°C
标准振荡频率	---	32.3	---	KHz	2.5V, 25°C, TT, 32K mode
	---	258.5	---	KHz	2.5V, 25°C, TT, 256K mode
启动时间	---	4.6	10	uS	2.5V, 25°C, TT
漏电流	---	0.15	1	nA	disable
校准范围	---	---	---		No trim

16. 4. 带隙基准源电路 (Bandgap)

带隙基准电路电气参数如下：

电气参数	最小值	典型	最大值	单位	条件/备注
lvdd 工作电流	6.1	9		uA	1.6~5.5V, tt
随温度变化范围	---	3	47	ppm/°C	-20 ~ 85 °C
输出电压范围	1.2013	1.233	1.269	V	Corner
PSRR		70		dB	tt
启动时间	---	1.2	10	uS	tt
漏电流	---	0.8	2	nA	Disable
输出驱动电流	---	---	---		No resistor loading
Trim 范围	-16%		+8%		-16%, -8%, +8%

16. 5. 低压差线性稳压器 (LDO)

带隙基准电路电气参数如下：

电气参数	最小值	典型	最大值	单位	条件/备注
随电源电压变化范围		1.65		V	VDD>1.7V
		VDD			VDD<1.65V

16. 6. 低电压侦测/复位电路 (LVD/LVR)

电气参数	最小值	典型	最大值	单位	条件/备注
		1.8		V	
		2.0			
		2.2			
		2.8			
LVR delay		125	157	us	

*设计参考值

16.7. 上电复位电路 (POR)

电气参数	最小值	典型	最大值	单位	条件/备注
Ivdd 工作电流		50		nA	3.3V
随温度变化范围	-	1.8V	-		3.3v, tt

16.8. I/O PAD 电路

电气参数	最小值	典型	最大值	单位	条件/备注
VIL	0		0.3*VDD	V	
VIH	0.7*VDD		VDD	V	
源电流		19		mA	5V, 25°C
沉电流		25		mA	5V, 25°C
上拉电阻		28		kΩ	5V
		41.7			3.3V

16.9. Comparator 比较器电路

电气参数	最小值	典型	最大值	单位	条件/备注
工作电流		70		uA	3.3V, tt

16.10. 4bit DAC 电路 (比较器参考 CVREF)

电气参数	最小值	典型	最大值	单位	条件/备注
工作电流				mA	1.8~5.5V , -20°C~85°C
相对精度	—	VDD/16	—	V	同上
绝对精度	—	—	1/2	LSB	同上
单位电阻(unit resistor)	—	5000	—		同上
稳定时间(Settle Time)	—	—	10	uS	00000->11111

16. 11. 总体工作电流 (I_{vdd})

电气参数	最小值	典型	最大值	单位	条件/备注
正常模式		310		uA	3.3V, 2MHz
		130		uA	1.8V, 2MHz
		50		uA	3.3V, 32KHz
		28.5		uA	1.8V, 32KHz
休眠模式 (Sleep, WDT ON)		3		uA	3.3V
休眠模式 (Sleep, WDT OFF)		0.8		uA	3.3V
休眠模式 (Sleep, LVR ON)		15		uA	3.3V

注意:

1. 电流的测试条件为 IO 处于输入模式, IO 外部下拉;
2. 比较器处于关闭状态, CM[2:0]=111。

17.指令集列表

本芯片采用精简指令架构，一共 37 条指令，以下是各指令的描述。

汇编语法	功能	运算	状态位
BCR R, b	Bit clear	0-> R(b)	NONE
BSR R, b	Bit set	1-> R(b)	NONE
BTSC R, b	Bit test, skip if 0	Skip if R(b)=0	NONE
BTSS R, b	Bit test, skip if 1	Skip if R(b)=1	NONE
NOP	No operation	None	NONE
CLRWD	Clear WDT	0-> WDT	/PF, /TF
SLEEP	ENTER SLEEP MODE	0-> WDT, STOP OSC	/PF, /TF
STTMD	Store W TO TMODE	W-> TMODE	NONE
CTLIO R	Control IO direction reg	W-> IODIRr	NONE
STR R(MOVWF)	Store W to reg	W-> R	NONE
LDR R, d(MOVF)	Load reg to d	R-> d	Z
SWAPR R,d	Swap halves reg	[R(0-3)R(4-7)]-> d	NONE
INCR R, d	Increment reg	R+ 1-> d	Z
INCRSZ R, d	Increment reg, skip if 0	R+ 1-> d	NONE
ADDWR R, d	Add W and reg	W+ R-> d	C, HC, Z
SUBWR R, d	Sub W from reg	R- W-> d R+ /W+ 1-> d	C, HC, Z
DECR R, d	Decrement reg	R- 1-> d	Z
DECRSZ R, d	Decrement reg, skip if 0	R- 1-> d	NONE
ANDWR R, d	AND W and reg	R& W-> d	Z
IORWR R, d	Inclu.OR W and reg	W R-> d	Z
XORWR R, d	Exclu.OR W and reg	W^ R-> d	Z
COMR R, d	Complement reg	/R-> d	Z
RRR R, d	Rotate right reg	R(n)-> R(n-1), C-> R(7), R(0)-> C	C
RLR R, d	Rotate left reg	R(n)-> R(n+1), C-> R(0), R(7)-> C	C
CLR	Clear working reg	0-> W	Z
CLRR R	Clear reg	0-> R	Z
RETI	Return from interrupt	Stack-> PC, 1-> GIE	NONE
RET	Return from subroutine	Stack-> PC	NONE

LCALL N	Long CALL subroutine	N-> PC, PC+1-> Stack	NONE
LJUMP N	Long JUMP address	N-> PC	NONE
LDWI I(MOVLW)	Load immediate to W	I-> W	NONE
ANDWI I	AND W and imm	W& I-> W	Z
IORWI I	Inclu.OR W and imm	W I-> W	Z
XORWI I	Exclu.OR W and imm	W^ I-> W	Z
RETW I	Return, place imm to W	Stack-> PC, I-> W	NONE
ADDWI I	Add imm to W	W+I-> W	C, HC, Z
SUBWI I	Subtract W from imm	I-W-> W	C, HC, Z

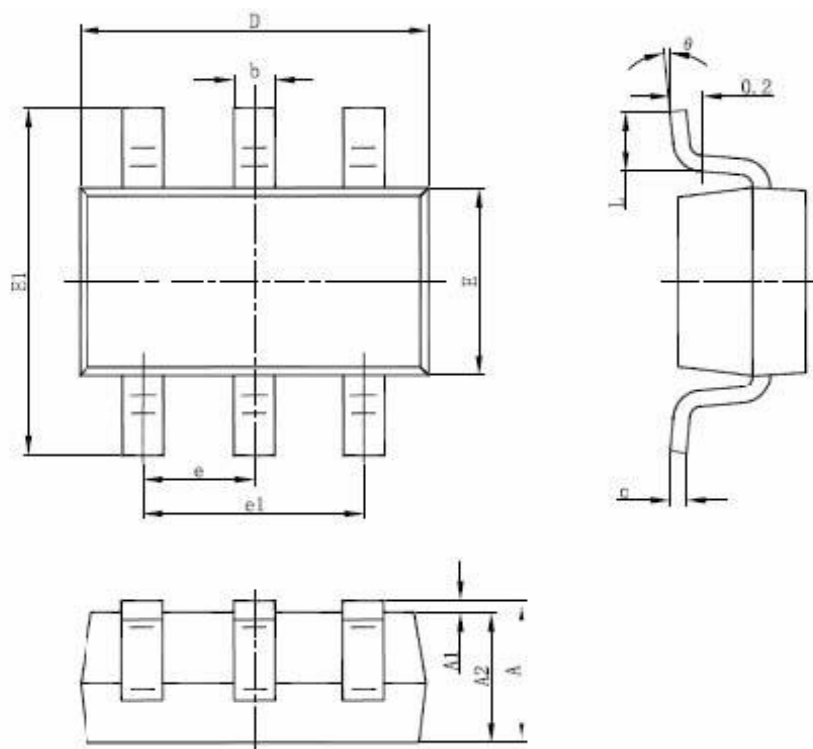
注意:

1. 在 TX-I-02X 芯片里, **TMODE** 寄存器是指 **OPTION**, 即 **STTMD** 指令的操作是把 **W** 存到 **OPTION**;

18. 芯片封装信息

本芯片采用 SOT23-6、SOP8、SOP14、SOP16、SOP18、TSSOP20 封装方式，具体封装尺寸信息如下：

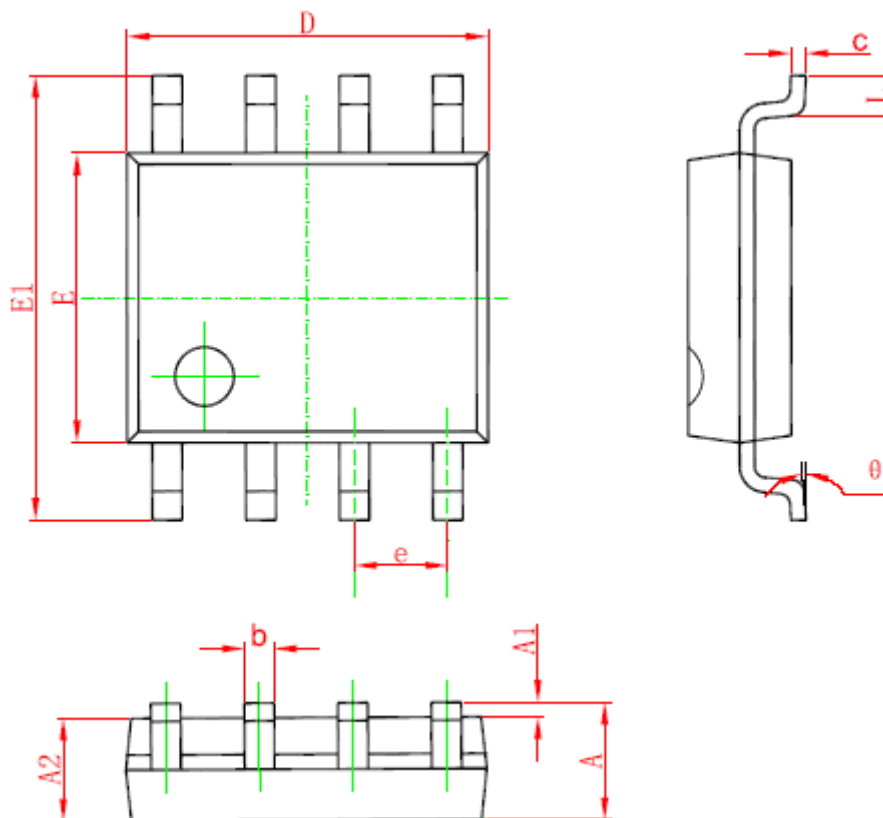
SOT23-6 封装尺寸如下：



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	1.050	1.250	0.041	0.049
A1	0.000	0.100	0.000	0.004
A2	1.050	1.150	0.041	0.045
b	0.300	0.500	0.012	0.020

c	0.100	0.200	0.004	0.008
D	2.820	3.020	0.111	0.119
E	1.500	1.700	0.059	0.067
E1	2.650	2.950	0.104	0.116
e	0.950 (BSC)		0.037 (BSC)	
e1	1.800	2.000		0.079
L	0.300	0.600		0.024
θ	0°	8°	0°	8°

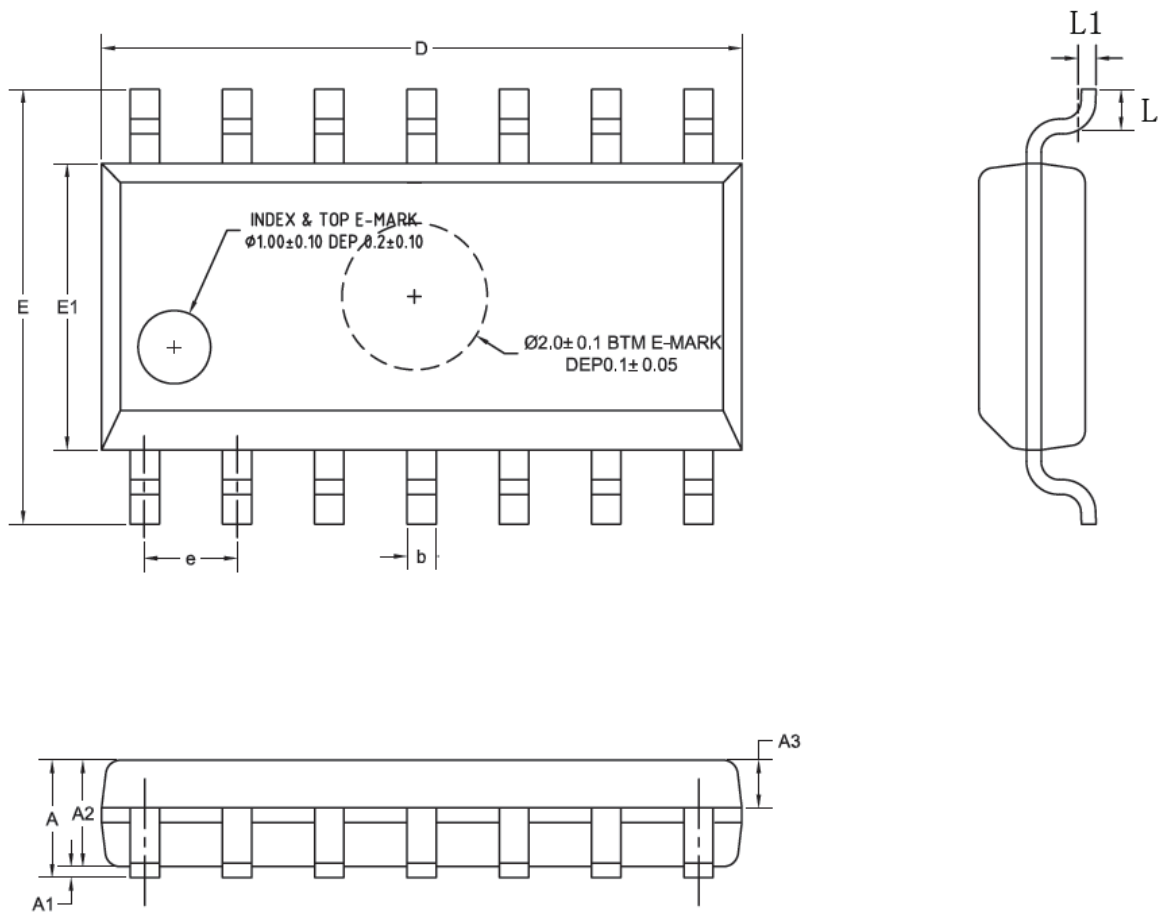
SOP-8 封装尺寸如下:



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	1.350	1.750	0.053	0.069
A1	0.100	0.250	0.004	0.010
A2	1.350	1.550	0.053	0.061
b	0.330	0.510	0.013	0.020

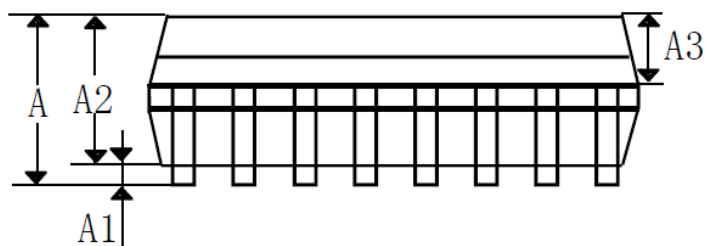
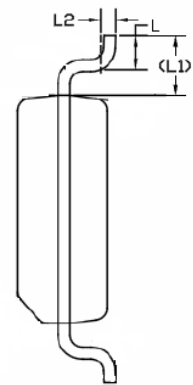
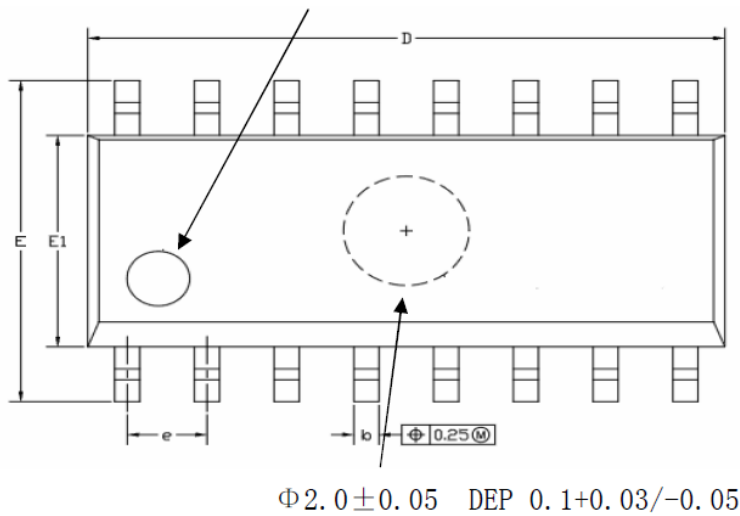
c	0.170	0.250	0.006	0.010
D	4.700	5.100	0.185	0.200
E	3.800	4.000	0.150	0.157
E1	5.800	6.200	0.228	0.244
e	1.270 (BSC)		0.050 (BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°

SOP-14 封装尺寸如下:



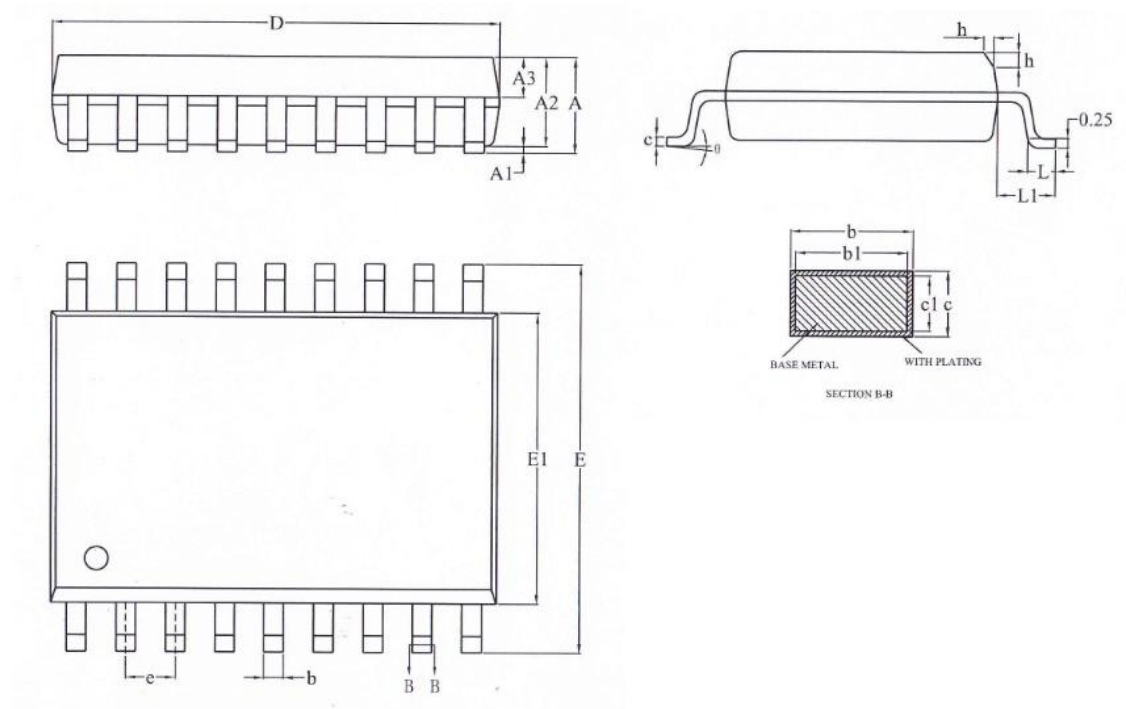
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	1.700	-	0.066
A1	0.100	0.200	0.004	0.008
A2	1.400	1.500	0.054	0.059
A3	0.620	0.680	0.024	0.027
b	0.370	0.420	0.014	0.016
D	8.710	8.910	0.340	0.347
E	5.900	6.100	0.230	0.238
E1	3.800	3.950	0.148	0.154
e	1.270(BSC)		0.050(BSC)	
L	0.500	0.700	0.020	0.027
L1	0.250(BSC)		0.010(BSC)	

SOP-16 封装尺寸如下:



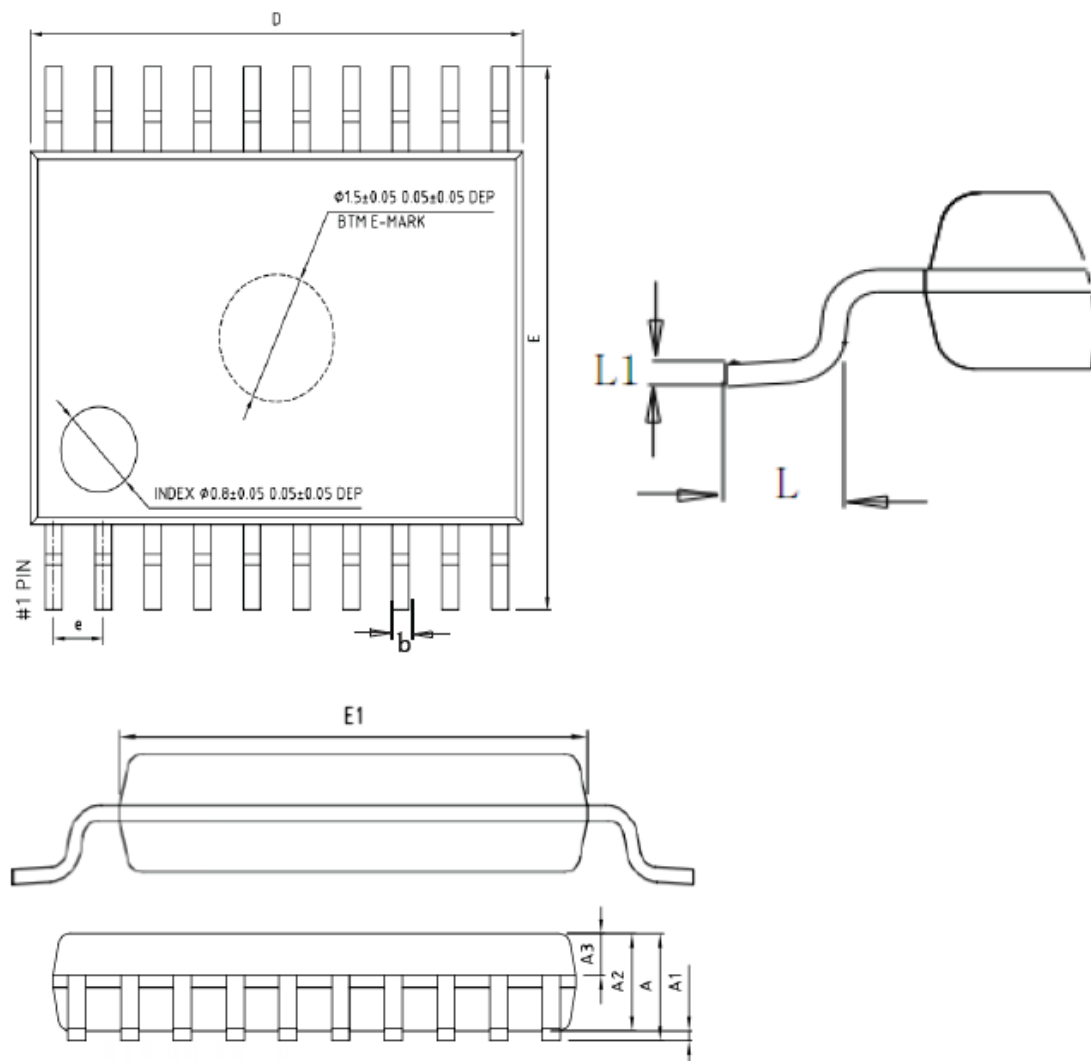
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	1.700	-	0.066
A1	0.100	0.200	0.039	0.008
A2	1.420	1.480	0.554	0.058
A3	0.620	0.680	0.242	0.027
D	9.960	10.160	3.884	0.396
E	5.900	6.100	2.301	0.238
E1	3.870	3.930	1.509	0.153
b	0.370	0.430	0.144	0.017
e	1.240	1.300	0.484	0.051
L	0.500	0.700	0.195	0.027
L1	1.050(REF)		0.041(REF)	
L2	0.250(BSC)		0.010(BSC)	

SOP-18 封装尺寸如下:



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	-	2.650	-	0.103
A1	0.100	0.300	0.004	0.012
A2	2.200	2.400	0.086	0.094
A3	0.970	1.070	0.038	0.042
b	0.350	0.430	0.014	0.017
b1	0.340	0.400	0.013	0.016
c	0.250	0.290	0.010	0.011
c1	0.240	0.260	0.009	0.010
D	11.350	11.550	0.443	0.450
E	10.100	10.500	0.394	0.410
E1	7.400	7.600	0.289	0.296
e	1.270(BSC)		0.050(BSC)	
L	0.70	1.000	0.027	0.039
L1	1.40(REF)		0.055(REF)	
h	0.250	0.750	0.010	0.029
θ	0	8°	0	8°

TSSOP-20 封装尺寸如下:



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	1.070	1.230	0.042	0.048
A1	0.050	0.150	0.002	0.006
A2	0.950	1.050	0.037	0.041
A3	0.411	0.461	0.016	0.018
b	0.200	0.280	0.008	0.011
D	6.400	6.600	0.250	0.257
E	6.300	6.500	0.246	0.254
E1	4.300	4.500	0.168	0.176
e	0.625	0.675	0.024	0.026



TX-I-02X

L	0.550	0.650	0.021	0.025
L1	0.25(BSC)		0.010(BSC)	

19.附录 1，文档更改历史

日期	版本	内容
2015-10-13	1.01	初始版本
2015-10-21	1.02	修改 16.2.3 各 IO 框图，Q3 改为 Q1 修改 BOR 复位描述，更新图 4.4 11.1 的步骤加入 A，“清除 GIE” RAIF 改为 PAIF 更新脚位图的笔误 更新 17.5LVR 的档数：1.8/2.0/2.2/2.8V
2015-11-17	1.03	MSCKCON 的 7~5 位注明“保留位，不能写 1” 更新特性列表，RAPU 改为 PAPU 上电复位时间由 8ms 改为 4ms 修改 2.1.31 小节 添加“INT 中断”和“PORTA 电平变化中断”到 14 章 加入 11.2“DROM 读步骤” 改正 16.2.1“弱上拉”中的 PORTA[3]，改为 PORTA[5] 更新 14pin 的脚位图 更新图 10.5，比较器的配置 RA2 改为 PA2 加入 20pin 脚位图
2015-12-23	1.05	改正一些笔误
2016-1-19	1.06	加入 VIL、VIH，改正上拉电阻阻值 改正“PCL 和 PCLTH”章节的笔误 删掉时钟描述的一个笔误 加入指令集列表
2016-5-12	1.07	修改比较器章节的一些笔误 修改数据 EEPROM 的写流程 扩展“比较器的参考电压”一节 改正 MCLRE 位描述 明确 WDT 固定使用 32K 时钟
2016-7-14	1.08	加入 32KHz 的工作电流
2016-7-28	1.09	加入系统时钟频率及其对应的电压范围 F _{SYS} =8MHz: 1.8V~2.7V: F _{SYS} =16MHz: 2.7V~5.5V
2016-10-10	1.10	修改内部快时钟随 VCC 变化范围，由 0.04%/V 改为 +/-3%
2016-11-1	1.11	更新表格 16.7 和 16.9
2016-11-15	1.12	添加 EEPROM 的初始化操作到第 10 节

