



1.基本特性

ROM : 2K x 14 位

RAM : 72 x 8 位

堆栈 : 2 级

高级模式与一般模式

I/O 口 : 20 个 I/O

定时器/计数器 : 8 位 x1 (TMR0)

预分频器 : 8 位

Watchdog 定时器 : 在芯片上基于内部 RC 振荡器的 watchdog , 最短周期是 20ms; 用户
可通过使用预分频器使 watchdog 溢出周期达到 2.56s。

上电复位& Watchdog 定时器溢出复位

复位定时器 : 20 mS (5V)

四种外部振荡模式 : RC,LP,NT,HS 晶振

工作电压 : 2.2V 5.5V

指令集 : 78 个

唤醒 : Port B (PB₇~ PB₀)

复位向量 : 7FFH



2. 脚位定义

RTCC	1		28	RESETB
VDD	2		27	OSC ₁
NC	3		26	OSC ₂
VSS	4		25	PC ₇
NC	5		24	PC ₆
PA ₀	6		23	PC ₅
PA ₁	7		22	PC ₄
PA ₂	8		21	PC ₃
PA ₃	9		20	PC ₂
PB ₀	10		19	PC ₁
PB ₁	11		18	PC ₀
PB ₂	12		17	PB ₇
PB ₃	13		16	PB ₆
PB ₄	14		15	PB ₅

封装方式：DIP、SOP

VSS	1		28	RESETB
RTCC	2		27	OSC ₁
VDD	3		26	OSC ₂
VDD	4		25	PC ₇
PA ₀	5		24	PC ₆
PA ₁	6		23	PC ₅
PA ₂	7		22	PC ₄
PA ₃	8		21	PC ₃
PB ₀	9		20	PC ₂
PB ₁	10		19	PC ₁
PB ₂	11		18	PC ₀
PB ₃	12		17	PB ₇
PB ₄	13		16	PB ₆
VSS	14		15	PB ₅

封装方式：SSOP



脚位描述

脚位名称	I/O	描述
RTCC	I	外部时钟输入到 TMR0 计数器
PA ₃₋₀	I/O	I/O 口
PB ₇₋₀	I/O	I/O 口 & 唤醒 (输入模式)
PC ₇₋₀	I/O	I/O 口
RESETB	I	系统复位信号输入
OSC ₁	I	振荡器输入
OSC ₂	O	振荡器输出
VDD	P	电源输入
VSS	P	接地输入

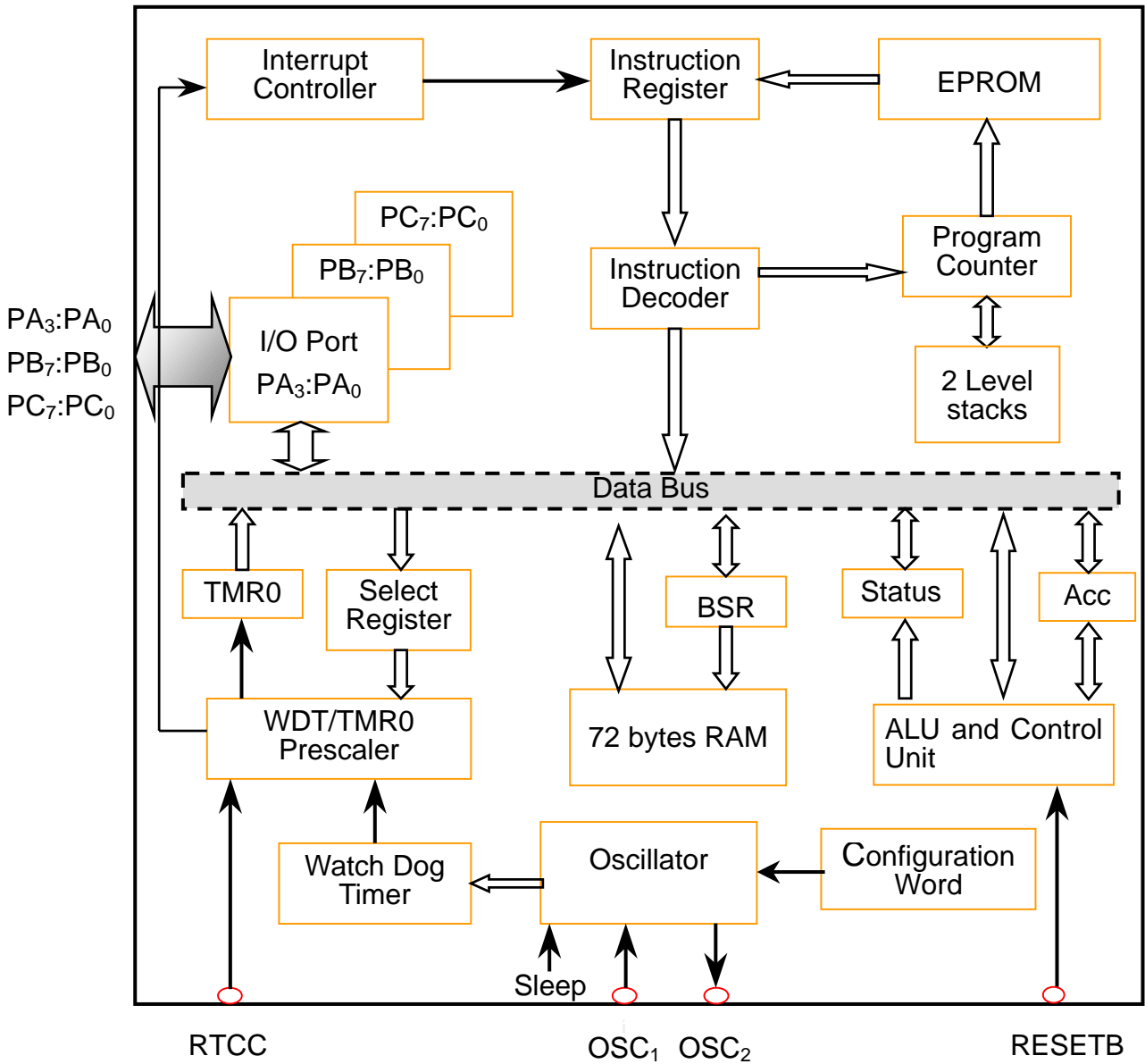
I: 输入; O: 输出; I/O: Bi-方向; P: 电源

3. 控制寄存器

Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG (Instruction)					TYPE	CPT	WDTE	FOSC ₁	FOSC ₀
SELECT				SUR ₀	EDGE ₀	PSA	PS ₂	PS ₁	PS ₀
IAR	\$00		A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
TMR0	\$01	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
PC	\$02	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
STATUS	\$03		SA ₁	SA ₀	\overline{TO}	\overline{PD}	Z	DC	C
BSR	\$04		D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
I/O Port _A	\$05					PA ₃	PA ₂	PA ₁	PA ₀
I/O Port _B	\$06	PB ₇	PB ₆	PB ₅	PB ₄	PB ₃	PB ₂	PB ₁	PB ₀
I/O Port _C	\$07	PC ₇	PC ₆	PC ₅	PC ₄	PC ₃	PC ₂	PC ₁	PC ₀



4. 系统图表





5. 存储器映像

TM58PC20 存储器分为程序存储器和数据存储器。

5.1 程序存储器

可直接寻址的同一页上只有 512 个字。其它的程序存储器可以通过设置状态寄存器的 bit6~bit5 寻址。指令执行顺序是由程序计数器控制的，程序计数器自动加 1。然而顺序也能被 skip, call, goto 指令改变，或者通过移动数据到程序计数器改变顺序。

TM58PC20有一个可访问2K空间的11位程序计数器。如果访问地址超过2K，地址仍会映像到2K存储器，也就是2K + M将会映像到M。在复位向量位置加一个NOP指令会导致在地址000h处重新启动。ROM结构如图5-1所示。

000H Page 0 1FFH
200H Page 1 3FFH
400H Page 2 5FFH
600H Page 3 7FEH
7FFH Reset vector

图 5-1 ROM 结构



5.2 Configuration 存储器

configuration word 位于 800H , 包含振荡选择、WDT 允许、编码保护、运行类型选择。

Bit	Symbol	Description			
		Bit1	Bit0	OSC Type	Resonance Frequency
1~0	FOSC ₁ ~FOSC ₀	0	0	LP (low speed)	32~200K hz
		0	1	NT (Normal speed)	200K~10M hz
		1	0	HS (high speed)	10~20M hz
		1	1	RC	32K ~ 6M hz
2	WDTE	WDTE: Watchdog enable/disable control 1: WDT enable 0: WDT disable			
3	CPT	CPT: Code Protection bit 1: OFF 0: ON			
4	TYPE	TYPE: Select operating mode 1: Advanced mode 0: General mode			

图 5-2 Configuration Word



5.3 数据存储器

数据存储器是由专用功能寄存器和通用存储器组成的。

TM58PC20提供72个通过bank选择机制存取数据的通用寄存器。专用功能寄存器包含程序计数器（PC）、定时器（TMR0）寄存器、状态寄存器、bank选择寄存器、I/O定向寄存器。此外，TM58PC20有3个辅助寄存器：间接寻址寄存器（IAR）、选择寄存器（Select）、I/O定向寄存器（IODIR）。寄存器映像如图5-3所示。

	Bank0	Bank1	Bank2	Bank3
00h	IAR	Map back to address in Bank0		
01h	TMR0			
02h	PC			
03h	STATUS			
04h	BSR			
05h	PORTA			
06h	PORTB			
07h	PORTC			
08h~0fh	General Purpose Register			
8+16*4=72	General Purpose Register 10-1F	General Purpose Register 30-3F	General Purpose Register 50-5F	General Purpose Register 70-7F

图 5-3 寄存器映像

- A. 间接寻址寄存器不是物理寄存器，而是以间接寻址来辅助bank选择寄存器。访问间接寻址寄存器的任何指令实际上是映像到bank选择寄存器所指的地址。因为间接寻址寄存器不是物理线路，用户读取间接寻址寄存器本身（BSR=00H）将总是返回数据00h。写间接寻址寄存器本身等同于NOP指令。
- B. 选择寄存器用来控制WDT和TMR0。在数据存储中它不分配一个唯一的地址，而通过选择指令只设置控制位，也就是说，它是只写寄存器。累加器的内容将会通过执行选择指令被送到可选寄存器。如果程序未设置选择寄存器，默认值是3FH。如图5-4所示说明如何设置选择寄存器。



Bit	Symbol	Description				
		PS ₂	PS ₁	PS ₀	TMR0 rate	WDT rate
2~0	PS ₂ ~PS ₀	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
3	PSA	PSA: 预分频器分配位 1: 将预分频器分配给 WDT 0: 将预分频器分配给TMR0				
4	EDGE ₀	EDGE ₀ : TMR0源信号边沿控制位 1: 外部时钟信号从高电平到低电平时, 定时器加1 0: 外部时钟信号由低电平到高电平时, 定时器加1				
5	SUR ₀	SUR ₀ : TMR0时钟源位 1: RTCC输入 0: (系统时钟)/4 或内部指令周期				

图 5-4 选择寄存器

C. I/O方向控制寄存器与选择寄存器相似，也是只写寄存器。设置一个I/O作为输入，相应的方向控制位必须是高位，同样地零代表输出。任何方向控制位可用IODIR指令作为输入或输出单独地被编程。如果寄存器不被编程，所有的I/O口将总保持输入模式。

● 程序计数器是11位二进制计数器，除了在下方的条件下，每个指令周期计数器都加1。

1. call, goto, lgoto and lcall：标志将移到程序计数器。

2. retla and ret：堆栈顶端值将弹出到程序计数器。

当程序计数器到下一页时自动加 1。值得注意的是：状态寄存器中页选位不会同时被改变。除非页选位在程序中已更新，否则 GOTO, CALL 或 MOVAM 02H 将会返回到原来的页。为了减少程序的复杂性，TM58PC20 提供 2 个指令子程序调用和分步处理，即 LCALL, LGOTO。LCALL 和 LGOTO 可寻址 ROM 任何地址，但不必设定页选位。CALL 和 GOTO 的操作分别是 8 位和 9 位，因此需要特殊位（即页选位）寻址整个存储器。然而 LCALL 和 GOTO 有 11 位易于寻址 ROM 所有空间的操作数。



-
- TMR0是8位二进制计数器/定时器，该寄存器通过RTCC脚的外部信号边沿变化或内部指令周期加1。它具有如下特性。
 - A. 可读可写
 - B. 2个内部时钟同步
 - C. 可通过设置选择寄存器使用可编程预分频器。其他详细说明在下一章会有具体描述。
 - 状态寄存器包含页选位、超时位、掉电位和ALU状态。值得注意的是 \overline{TO} 和 \overline{PD} 是受硬件控制的，而不会被程序改变。



位	符号	描述		
0	C	进位位与借位位		
		加指令	减指令	
		1: 从 MSB 进位 0: 无进位	1: 无借位 ^(注1) 0: 从 MSB借位	
1	DC	四位进位位与四位借位位		
		加指令	减指令	
		1: 从低四位进位 0: 无低四位进位	1: 无低四位借位 0: 从低四位借位	
2	Z	零标志位： 1：逻辑操作结果是0 0：逻辑操作结果非0		
3	\overline{PD}	掉电标志位： ^(注2) 1：上电后或执行CLRWD指令后，结果为1。 0：执行SLEEP指令后，结果为0。		
4	\overline{TO}	溢出标志位： 1: 上电后或执行CLRWD或SLEEP指令后，结果为1 0: WDT时间溢出时，结果为0		
6~5	SA ₁ ~SA ₀	PA1	PA2	页位置
		0	0	Page 0 (000H~1FFH)
		0	1	Page 1 (200H~3FFH)
		1	0	Page 2 (400H~5FFH)
		1	1	Page 3 (600H~7FFH)

图 5-5 状态寄存器

注1: 减指令的执行是通过与减数的二进制补码相加来实现的，C = 1代表结果是正数。C位与借位之间的关系如图5-5-1所示。



B0H - 50H										50H - B0H									
	C	B7	B6	B5	B4	B3	B2	B1	B0		C	B7	B6	B5	B4	B3	B2	B1	B0
+		1	0	1	1	0	0	0	0	+		0	1	0	1	0	0	0	0
=	1	0	1	1	0	0	0	0	0	=	0	1	0	1	0	0	0	0	0

图 5-5-1

注 2： \overline{TO} 和 \overline{PD} 位是低有效，可决定复位的不同原因。图 5-5-2 举例说明了不同复位后 \overline{TO} 和 \overline{PD} 的值。

\overline{TO}	\overline{PD}	复位结果
0	0	从睡眠模式 WDT 超时
0	1	从一般模式 WDT 超时
1	0	从睡眠模式 RESETB 输入低复位
1	1	上电复位
不变	不变	从一般模式 RESETB 输入低复位

图 5-5-2

- bank 选择寄存器与间接寻址寄存器间接地访问数据存储器。直接寻址必须根据 bank 选择寄存器存取 bank1 ~ bank3，因为在一般模式下只有 5 位地址操作数。Bank 选择寄存器的 bit6~bit5 被用来选择指定的存储器 bank。地址 20H~2FH，40H~4FH，60H~6FH 是不可存取的，这些地址将映像到 00H~0FH (bank0)，地址映像如图 5-6 所示。

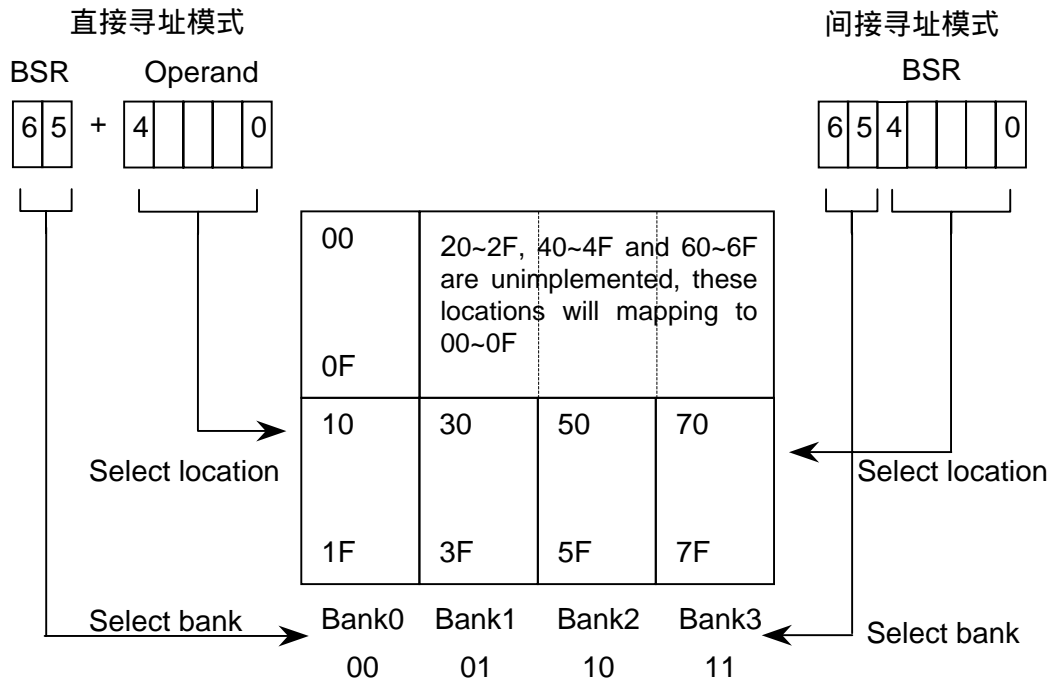


图 5-6 直接寻址与间接寻址映像

- Port A~C 是可编程的 I/O 口。值得注意的是，即使 I/O 脚是输出模式，读 I/O 指令也总是读 I/O 脚。复位时所有 I/O 脚都被置成输入模式，直到 IO 定向寄存器被改变。



6. 功能描述

6.1 TMR0 与 Watchdog 定时器

图 6-1 显示 TMR0/WDT 预分频器的方框图。如图所示，预分频器寄存器可以是 TMR0 的预分频器或是 WDT 的后分频器。

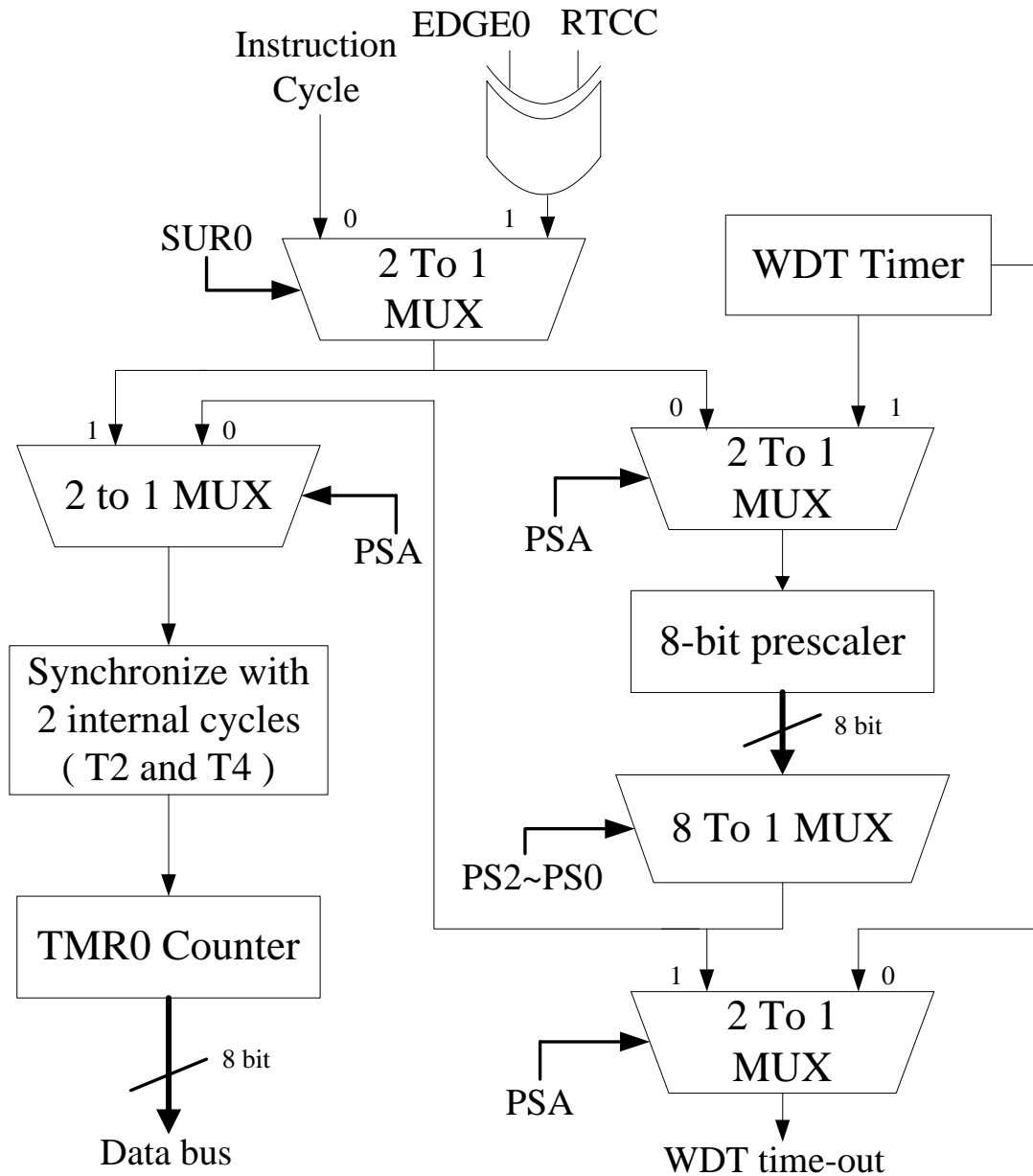


图 6-1 TMR0/WDT 预分频器方框图



TMR0 是一个 8 位定时器/计数器。TMR0 的时钟源可以来自指令时钟或外部时钟。

- A. 选择指令时钟，选择寄存器的 SUR0 位应该清零。当未使用预分频器时，TMR0 将会在每个指令周期加 1。
- B. 选择外部时钟，选择寄存器的 SUR0 位应该置 1。在该模式，TMR0 根据 EDGE0 来决定在外部时钟的上升沿或下降沿加 1。当 TMR0 选择外部时钟时，应当注意外部时钟将与内部时钟同步进行。TM58PC20 通过内部时钟的 T2 和 T4 采样来同步外部时钟。如果外部脉冲小于 2 个内部时钟周期，那么脉冲可能会被忽略。也就是说，外部时钟必须至少在 2 个内部时钟周期保持稳定状态。

WDT 计数器是一个 8 位二进制计数器，WDT 时钟源是由一个独立的不需要任何外部时钟的 RC 振荡器芯片提供的。因此即使芯片已进入睡眠状态 WDT 仍继续计数。若 WDT 超时，系统将重激活并将超时标志位(状态寄存器的 bit4) 清 0。WDT 超时时间会随温度、电源电压的变化而变化，并且可通过设置预分频器来改变超时时间。通过设置 PS2~PS0 为“111”，最大分频率可达 1:128。

通过 PSA 位预分频器可分配到 TMR0 或 WDT。WDT 或 TMR0 不能同时使用预分频器。当预分频器从 TMR0 变成 WDT，或从 WDT 变成 TMR0 时，执行下面的例子（2-3）。下面的例子可以避免超时复位。

```

Clrwdt
Clrm      TMR0; clear prescaler & TMR0
Movla    B'00xx1111'
Select
Clrwdt
Movla    B'00xx1xxx'; set prescaler to
desired
    
```

例 2 预分频器从 TMR0 变成 WDT

```

Clrwdt ; clear prescaler & WDT
Movla  B'00xx0xxx'
Select ; set prescaler to TMR0 with
      ; new rate
    
```

例 3 预分频器从 WDT 变成 TMR0



当预分频器被分配到 WDT，“CLRWDT”和“SLEEP”指令将会清除预分频器和 WDT。当预分频器被分配到 TMR0，任何写到 TMR0 的指令都会清除预分频器。

6.2 复位

当满足下列任意一个条件时，TM58PC20都可能复位：

- (1) 上电
- (2) RESETB脚输入一个负脉冲
- (3) WDT超时复位 (如果允许).

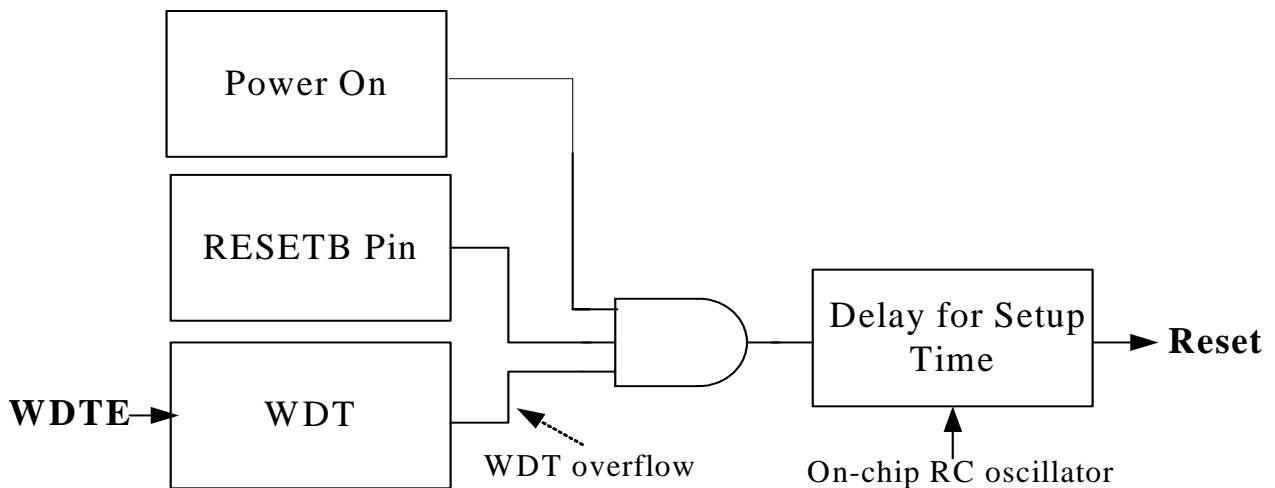


图 6-2 复位控制器

如图6-2所示，三个复位条件被列出。一般来说，我们称第一种情况为冷复位。冷复位的时间对于低速晶振和RC振荡器来说可能太短了，它们需要比激活时间长得多的时间。

注：激活时间大约20ms，激活时间随电源电压、process、温度变化而变化。

后两种情况被称为热复位。不同的复位对寄存器和随机 RAM 的影响也是不同的。 \overline{TO} 和 \overline{PD} 用来决定复位的类型。这两种复位的关系如图 6-3 所示。



Address	Name	Cold Reset	Warm Reset
N/A	Accumulator	xxxx xxxx	pppp pppp
N/A	IODIR	1111 1111	1111 1111
N/A	Select	--11 1111	--11 1111
00h	IAR	---- ----	---- ----
01h	TMR0	xxxx xxxx	pppp pppp
02h	PC	111 1111 1111	111 1111 1111
03h	STATUS	0001 1xxx	000? ?ppp ¹
04h	BSR	1xxx xxxx	1ppp pppp
05h	PORTA	0000 xxxx	0000 pppp
06h	PORTB	xxxx xxxx	pppp pppp
07h	PORTC	xxxx xxxx	pppp pppp
	General Purpose RAM	Xxxx xxxx	Pppp pppp

6-3 复位条件

X: 未知的; P: 原来的数值; ?: 依据不同复位条件;



6.3 高级模式

在高级模式，提供了唤醒功能。当输入脚电压变化时，芯片可以从睡眠模式唤醒。因此在进入睡眠模式前需要读出输入电压。

去抖动时间是接受键的第二次按键之前必须经过的时间间隔。用户可以利用延迟程序设置时间间隔（见例 1）。

按键抖动

```
After_wakeup
;-----
int_nt1          ;; filter out key begin bounce
    btmsc rb,0
    lgoto int_nt1
int_loop1        ;; filter out key end bounce
    call delay   ;; worse case 30ms
    btmsc rb,0
    lgoto int_loop1

    call delay_routine ;; such as 30ms
    btmsc rb,0
    lgoto int_loop1
;-----
```

例 1 按键去抖动



7. 指令集

Mnemonic Operands	Instruction Code (Advance)	Cycles	Status Affected	OP-code
ADDAM M, m	(M)+(acc) (M)	1	C, DC, Z	10 0101 1MMM MMMM
ADDAM M, a	(M)+(acc) (acc)	1	C, DC, Z	10 0101 0MMM MMMM
ANDAM M, m	(M) . (acc) (M)	1	Z	10 0100 1MMM MMMM
ANDAM M, a	(M) . (acc) (acc)	1	Z	10 0100 0MMM MMMM
ANDLA I	Literal . (acc) (acc)	1	Z	11 1001 iiiii iiiii
BCM M, b0	Clear bit0 of (M)	1	None	00 1100 0MMM MMMM
BCM M, b1	Clear bit1 of (M)	1	None	00 1100 1MMM MMMM
BCM M, b2	Clear bit2 of (M)	1	None	00 1101 0MMM MMMM
BCM M, b3	Clear bit3 of (M)	1	None	00 1101 1MMM MMMM
BCM M, b4	Clear bit4 of (M)	1	None	00 1110 0MMM MMMM
BCM M, b5	Clear bit5 of (M)	1	None	00 1110 1MMM MMMM
BCM M, b6	Clear bit6 of (M)	1	None	00 1111 0MMM MMMM
BCM M, b7	Clear bit7 of (M)	1	None	00 1111 1MMM MMMM
BSM M, b0	Set bit0 of (M)	1	None	00 1000 0MMM MMMM
BSM M, b1	Set bit1 of (M)	1	None	00 1000 1MMM MMMM
BSM M, b2	Set bit2 of (M)	1	None	00 1001 0MMM MMMM
BSM M, b3	Set bit3 of (M)	1	None	00 1001 1MMM MMMM
BSM M, b4	Set bit4 of (M)	1	None	00 1010 0MMM MMMM
BSM M, b5	Set bit5 of (M)	1	None	00 1010 1MMM MMMM
BSM M, b6	Set bit6 of (M)	1	None	00 1011 0MMM MMMM
BSM M, b7	Set bit7 of (M)	1	None	00 1011 1MMM MMMM
BTMSC M, b0	If bit0 of (M) = 0, skip next instruction	1 + (skip)	None	00 0100 0MMM MMMM
BTMSC M, b1	If bit1 of (M) = 0, skip next instruction	1 + (skip)	None	00 0100 1MMM MMMM
BTMSC M, b2	If bit2 of (M) = 0, skip next instruction	1 + (skip)	None	00 0101 0MMM MMMM
BTMSC M, b3	If bit3 of (M) = 0, skip next instruction	1 + (skip)	None	00 0101 1MMM MMMM
BTMSC M, b4	If bit4 of (M) = 0, skip next instruction	1 + (skip)	None	00 0110 0MMM MMMM
BTMSC M, b5	If bit5 of (M) = 0, skip next instruction	1 + (skip)	None	00 0110 1MMM MMMM
BTMSC M, b6	If bit6 of (M) = 0, skip next instruction	1 + (skip)	None	00 0111 0MMM MMMM



BTMSC M, b7	If bit7 of (M) = 0, skip next instruction	1 + (skip)	None	00 0111 1MMM MMMM
BTMSS M, b0	If bit0 of (M) = 1, skip next instruction	1 + (skip)	None	00 0000 0MMM MMMM
BTMSS M, b1	If bit1 of (M) = 1, skip next instruction	1 + (skip)	None	00 0000 1MMM MMMM
BTMSS M, b2	If bit2 of (M) = 1, skip next instruction	1 + (skip)	None	00 0001 0MMM MMMM
BTMSS M, b3	If bit3 of (M) = 1, skip next instruction	1 + (skip)	None	00 0001 1MMM MMMM
BTMSS M, b4	If bit4 of (M) = 1, skip next instruction	1 + (skip)	None	00 0010 0MMM MMMM
BTMSS M, b5	If bit5 of (M) = 1, skip next instruction	1 + (skip)	None	00 0010 1MMM MMMM
BTMSS M, b6	If bit6 of (M) = 1, skip next instruction	1 + (skip)	None	00 0011 0MMM MMMM
BTMSS M, b7	If bit7 of (M) = 1, skip next instruction	1 + (skip)	None	00 0011 1MMM MMMM
CALL I	Call subroutine	2	None	11 0110 iiiiiiii
CLRA	Clear accumulator	1	Z	10 0001 0000 0000
CLRM M	Clear memory M	1	Z	10 0001 1MMM MMMM
CLRWDT	Clear watch-dog register	1	TO, PO	10 0000 0000 0001
COMM M, m	~(M) (M)	1	Z	10 0010 1MMM MMMM
COMM M, a	~(M) (acc)	1	Z	10 0010 0MMM MMMM
DECM M, m	Decrement M to M	1	Z	10 0110 1MMM MMMM
DECM M, a	(M) - 1 (acc)	1	Z	10 0110 0MMM MMMM
DECMSZ M, m	(M) - 1 (M), skip if (M) = 0	1 + (skip)	None	10 0111 1MMM MMMM
DECMSZ M, a	(M) - 1 (acc), skip if (M) = 0	1 + (skip)	None	10 0111 0MMM MMMM
GOTO I	Goto branch	2	None	11 101i iiiiiiii
INCM M, m	(M) + 1 (M)	1	Z	10 1000 1MMM MMMM
INCM M, a	(M) + 1 (acc)	1	Z	10 1000 0MMM MMMM
INCMSZ M, m	(M) + 1 (M), skip if (M) = 0	1 + (skip)	None	10 1001 1MMM MMMM
INCMSZ M, a	(M) + 1 (acc), skip if (M) = 0	1 + (skip)	None	10 1001 0MMM MMMM
IODIR M	Set i/o direction	1	None	10 0000 0000 0MMM
IORAM M, m	(M) ior (acc) (M)	1	Z	10 1111 1MMM MMMM
IORAM M, a	(M) ior (acc) (acc)	1	Z	10 1111 0MMM MMMM
IORLA I	Literal ior (acc) (acc)	1	Z	11 0011 iiiiiiii



LCALL I	Call subroutine. However, LCALL can addressing 2K address	2	None	01 0iii iiiiii
LGOTO I	Go branch to any address	2	None	01 1iii iiiiii
MOVAM m	Move data form acc to memory	1	None	10 0000 1MMM MMMM
MOVLA I	Move literal to accumulator	1	None	11 0001 iiiiii
MOVM M, m	(M) (M)	1	Z	10 0011 1MMM MMMM
MOVM M, a	(M) (acc)	1	Z	10 0011 0MMM MMMM
NOP	No operation	1	None	10 0000 0000 0000
RET	Return	2	None	11 1111 0111 1111
RETLA I	Return and move literal to accumulator	2	None	11 1100 iiiiii
RLM M, m	Rotate left from m to itself	1	C	10 1100 1MMM MMMM
RLM M, a	Rotate left from m to acc	1	C	10 1100 0MMM MMMM
RRM M, m	Rotate right from m to itself	1	C	10 1110 1MMM MMMM
RRM M, a	Rotate right from m to acc	1	C	10 1110 0MMM MMMM
SELECT	Set select register	1	None	10 0000 0000 0010
SLEEP	Enter sleep (saving) mode	1	TO, PO	10 0000 0000 0011
SUBAM M, m	(M)-(acc) (M)	1	C, DC, Z	10 1010 1MMM MMMM
SUBAM M, a	(M) -(acc) (acc)	1	C, DC, Z	10 1010 0MMM MMMM
SWAPM M, m	Swap data from m to itself	1	None	10 1101 1MMM MMMM
SWAPM M, a	Swap data from m to acc	1	None	10 1101 0MMM MMMM
XORAM M, m	(M) xor (acc) (M)	1	Z	10 1011 1MMM MMMM
XORAM M, a	(M) xor (acc) (acc)	1	Z	10 1011 0MMM MMMM
XORLA I	Literal xor (acc) (acc)	1	Z	11 1000 iiiiii



8. 电气特性

8.1 绝对最大额定值

电源电压 $V_{SS}-0.3V$ to $V_{SS}+5.5V$ 存储温度 -50 to 125

输入电压 ... $V_{SS}-0.3V$ to $V_{DD}+0.3V$ 工作温度... -0 to 70

8.2 直流电特性

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		VDD	Conditions				
VDD	Operating Voltage	---		2.2		5.5	V
V_{IH}	Input High Voltage	5V	I/O Port	2		VDD	V
V_{IL}	Input Low Voltage	5V	I/O Port			0.8	V
I_{DD1}	Standby Current	5V	WDT disable		1		uA
			WDT enable		10		
I_{IL}	Input Leakage Current	5V	Vin=VDD, VSS		1		uA
I_{OH}	I/O Port Driving Current	5V	Voh=4.5V		9		mA
			Voh=4V		17		
			Voh=3.5V		23		
I_{OL}	I/O Port Sink Current	5V	Vol=0.5V		20		mA
			Vol=01V		35		
			Vol=1.5V		50		



8.3 交流电特性

Symbol	Parameter	Test Conditions		Min	Typ	Max	Unit
		VDD	Conditions				
f_{sys1}	System Clock	5V	LP Crystal mode	32		200	Khz
		3V		32		200	
f_{sys1}	System Clock	5V	NT Crystal mode	0.2		10	Mhz
		3V		0.2		10	
f_{sys3}	System Clock	5V	HS Crystal mode	10		20	Mhz
		3V					
f_{sys4}	System Clock	5V	RC mode			6	Mhz
		3V				6	
T_{wdt}	Watchdog Timer	5V			20		mS
		3V			30		
T_{rht}	Reset Hold Time	5V			20		mS
		3V			30		