HC18P23xL

数据手册

8 位 LCD 型 OTP 单片机

目录

1 产品简述

HC18P23xL是一颗采用高速低功耗CMOS工艺设计开发的8位高性能精简指令单片机，内部8K×16位一次性编程ROM(OTP-ROM)，512×8位的数据寄存器（RAM），6组双向I/O口，三个Timer定时器/计数器，两个CCP模块，3组6路可编程带死区控制的固定相位PWM，一个AD模块，支持一路UART及SPI通信，一个8×32（4×32）的液晶显示驱动模块，一个16通道的12位模数转换器，多个系统时钟，四种系统工作模式以及多个中断源。这款单片机可以广泛应用于带有显示功能的游戏摇杆、定时器、遥控器等产品。

1.1 特性

- CPU 特性
  - 36条高性能精简指令
  - 8K×16位的OTP程序存储器
  - 512×8位的数据存储器
  - 8级堆栈缓存器
  - 2T/4T时钟模式
  - 立即、直接和两组间接寻址模式
  - 16位RDT查表
- I/O 口
  - 6组双向I/O口：PORTA，PORTB，PORTC，PORTD，PORTE，PORTF
  - 最多48个双向I/O口，其中有40个可用作液晶驱动口
  - 高灌/高拉电流能力，可直接驱动LED（IOH/IOL：16mA/25mA@ 4.4V/0.6V）
  - 最多47个可编程弱上拉/下拉口（PA、PB、PC、PD、PE、PF）
  - 所有IO驱动电流可配置（IOH/IOL：5mA/8mA@ 4.4V/0.6V）
  - PORTF大灌电流、与LCD COM口复用(IOH/IOL：17mA/65mA@ 4.4V/0.6V)
  - 最多48个具有唤醒功能的电平变化中断端口（PA、PB、PC、PD、PE、PF）
- 三个 Timer 定时器/计时器
  - Timer0：带有8位预分频器的8位定时器/计数器
  - Timer1：带有预分频器的16位定时器/计数器
  - Timer2：带有8位周期寄存器的8位定时器
- 两个CCP模块
  - 16位捕捉、16位比较、最高12位PWM
- 3组6路可编程带死区控制的固定相位PWM
  其中1组12bit、2组8bit
- 液晶显示驱动
  - 最大驱动液晶点阵8×32（4×32）
  - 1/3、1/4、1/5、1/8 Duty，1/2、1/3、1/4Bias
- BOR复位系统
  - BOR2.0V/2.4V/3.6V
- UART 模块
- SPI 模块
- AD 模数转换器
  - 12位转换分辨率
  - 最多16个模拟输入通道（15个外部ADC输入，1个内部1/4VDD检测）
  - 内部参考电压(VDD、4V、3V、2V)
  - 外部参考电压
- 双系统时钟
  - 高频系统时钟
    - 高频晶体振荡器：最高 20MHz
    - 内部 RC 振荡器：高达 32MHz
  - 低频系统时钟
    - 低频晶体振荡器： 32.768KHz
    - 低频 RC 振荡器： 32K（5V 典型值）
- 系统工作模式
  - 高频模式
  - 低频模式
  - 休眠模式
  - 绿色模式
- 中断源
  - 定时器中断：Timer0、Timer1和Timer2
  - INT0外部中断

- 所有IO电平变化中断
- CCP1/CCP2中断
- ADC中断
- UART中断
- SPI中断
- PWM中断

■ 复位

- 上电复位（POR）
- 外部复位（MCLRB Reset）
- 欠压复位（BOR）
- 看门狗定时器复位（WDT Reset）

✓ 选型表

产品型号	ROM	RAM	堆栈	定时器	I/O	PWM	LCD	ADC	UART /SPI	PKG
HC18P232L	8K*16	512*8	8	3	14	6	4*6	9+1	1	SOP16
HC18P233L	8K*16	512*8	8	3	22	6	6*10	10+1	1	SSOP24
HC18P234L	8K*16	512*8	8	3	26	8	8*12	10+1	1	SOP28
HC18P235L	8K*16	512*8	8	3	46	8	4*30/ 8*30	13+1	1	LQFP48
HC18P235L	8K*16	512*8	8	3	48	8	4*32/ 8*32	15+1	1	Dice

使用注意事项：

1、 在系统对功耗要求较高时 ADC 推荐使用 2M 及以下采样时钟；

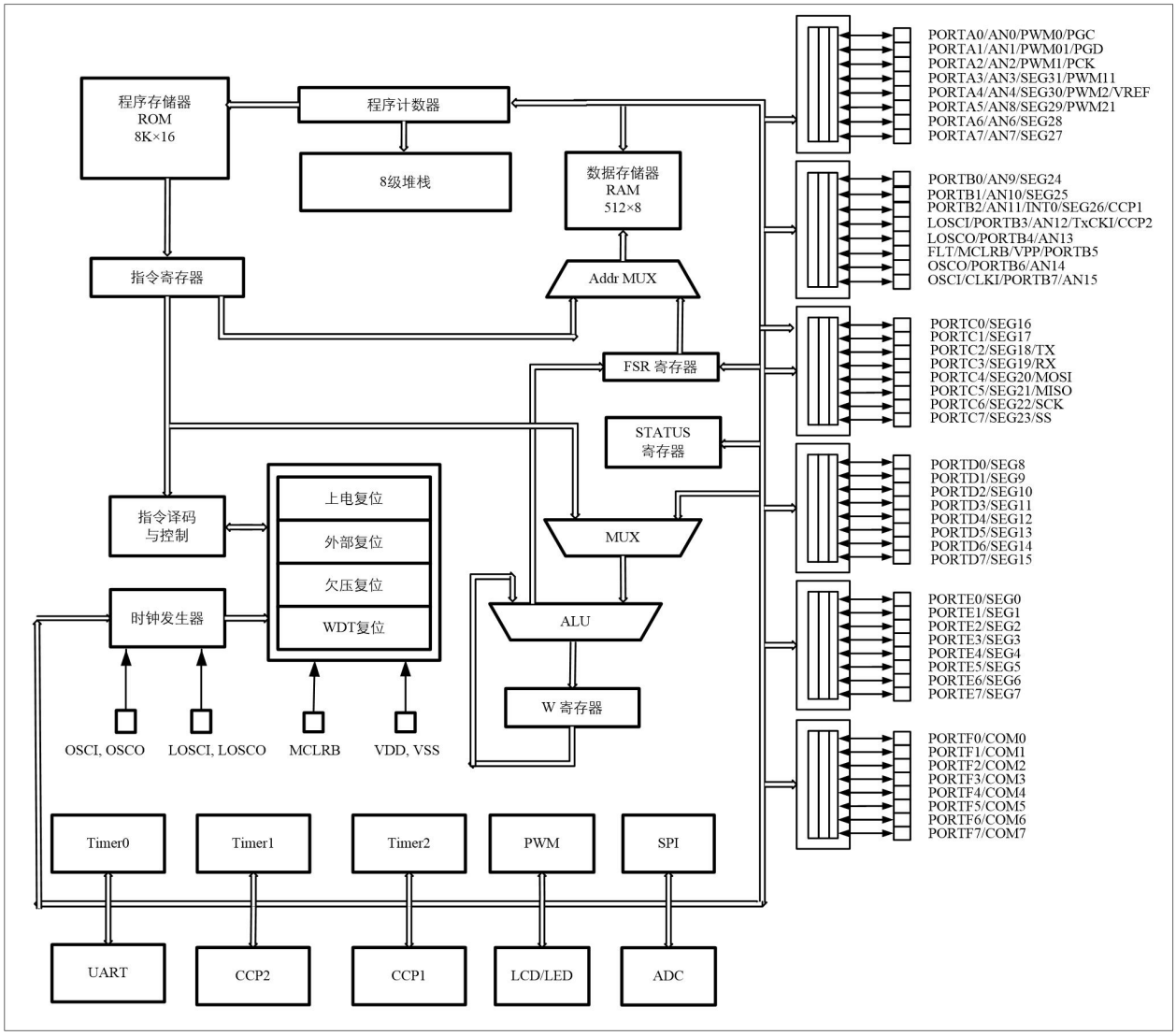
2、 为提高 ADC 检测精度，建议在 VDD 和 GND 之间并一个电容（104 电容即可）。

3、 使用 LCD 时，相应的 COM 口及 SEG 口需禁止上拉电阻及下拉电阻功能，且不能外接 VDD 及 GND，防止 LCD 波形异常。

4、 使用 LCD 时，需根据 LCD Pannel 尺寸选择合适的偏置电阻，避免由于驱动不够造成显示效果不佳。

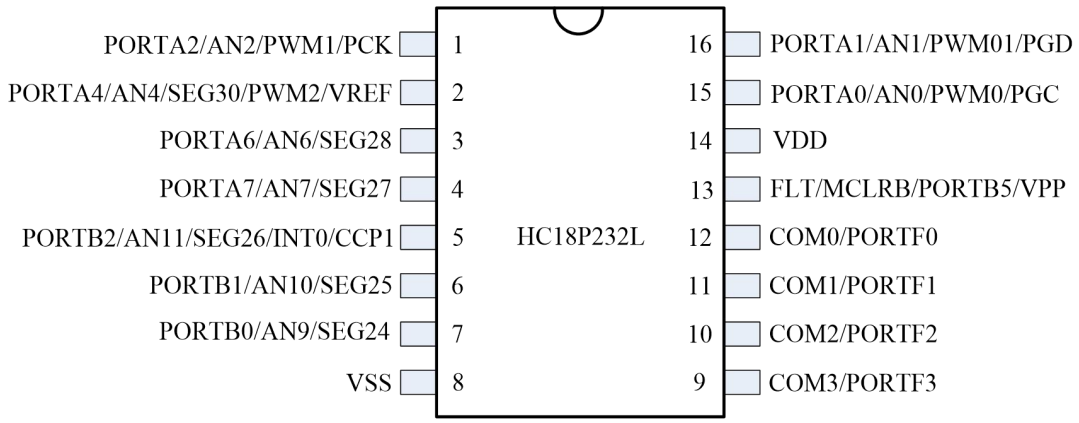
5、 在使用 1/3Duty 时，COM3（PORTF3）端口为 NC，请勿使用。

1.2 系统框图

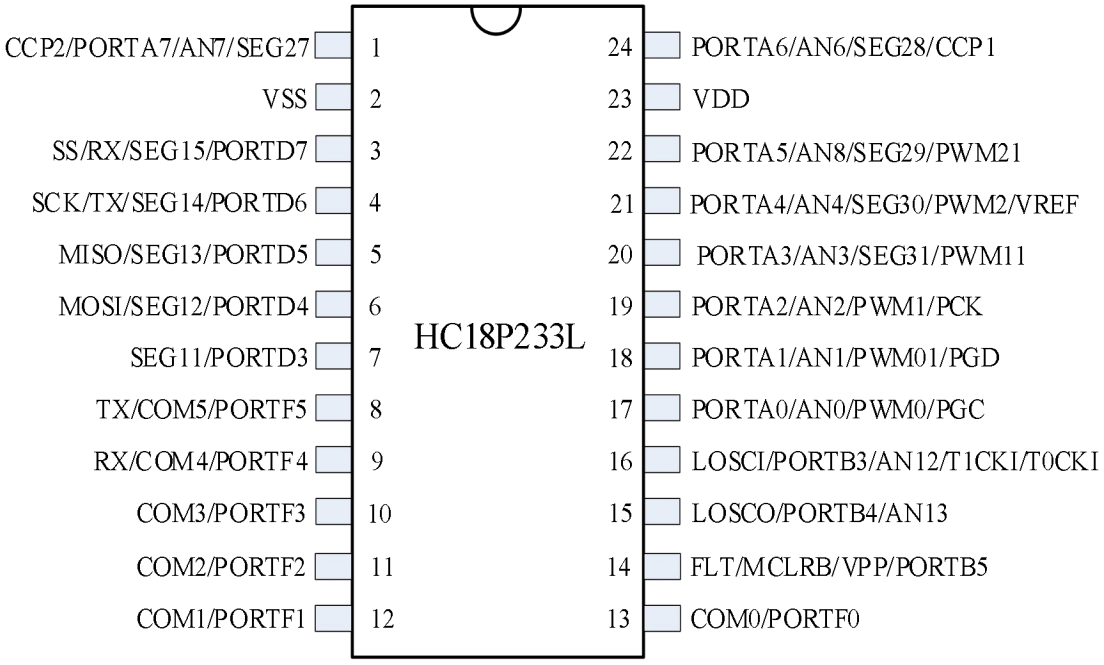

程序存储器
ROM
8K×16
程序计数器
8级堆栈
数据存储器
RAM
512×8
指令寄存器
Addr MUX
FSR 寄存器
STATUS
寄存器
指令译码
与控制
上电复位
外部复位
欠压复位
WDT复位
MUX
时钟发生器
ALU
W 寄存器
OSCI, OSCO
LOSCI, LOSCO
MCLRB
VDD, VSS
Timer0
Timer1
Timer2
PWM
SPI
UART
CCP2
CCP1
LCD/LED
ADC
PORTA0/AN0/PWM0/PGC
PORTA1/AN1/PWM01/PGD
PORTA2/AN2/PWM1/PCK
PORTA3/AN3/SEG31/PWM11
PORTA4/AN4/SEG30/PWM2/VREF
PORTA5/AN8/SEG29/PWM21
PORTA6/AN6/SEG28
PORTA7/AN7/SEG27
PORTB0/AN9/SEG24
PORTB1/AN10/SEG25
PORTB2/AN11/INT0/SEG26/CCP1
LOSCI/PORTB3/AN12/TxCKI/CCP2
LOSCO/PORTB4/AN13
FLT/MCLRB/VPP/PORTB5
OSCO/PORTB6/AN14
OSCI/CLKI/PORTB7/AN15
PORTC0/SEG16
PORTC1/SEG17
PORTC2/SEG18/TX
PORTC3/SEG19/RX
PORTC4/SEG20/MOSI
PORTC5/SEG21/MISO
PORTC6/SEG22/SCK
PORTC7/SEG23/SS
PORTD0/SEG8
PORTD1/SEG9
PORTD2/SEG10
PORTD3/SEG11
PORTD4/SEG12
PORTD5/SEG13
PORTD6/SEG14
PORTD7/SEG15
PORTE0/SEG0
PORTE1/SEG1
PORTE2/SEG2
PORTE3/SEG3
PORTE4/SEG4
PORTE5/SEG5
PORTE6/SEG6
PORTE7/SEG7
PORTF0/COM0
PORTF1/COM1
PORTF2/COM2
PORTF3/COM3
PORTF4/COM4
PORTF5/COM5
PORTF6/COM6
PORTF7/COM7

1.3 引脚图

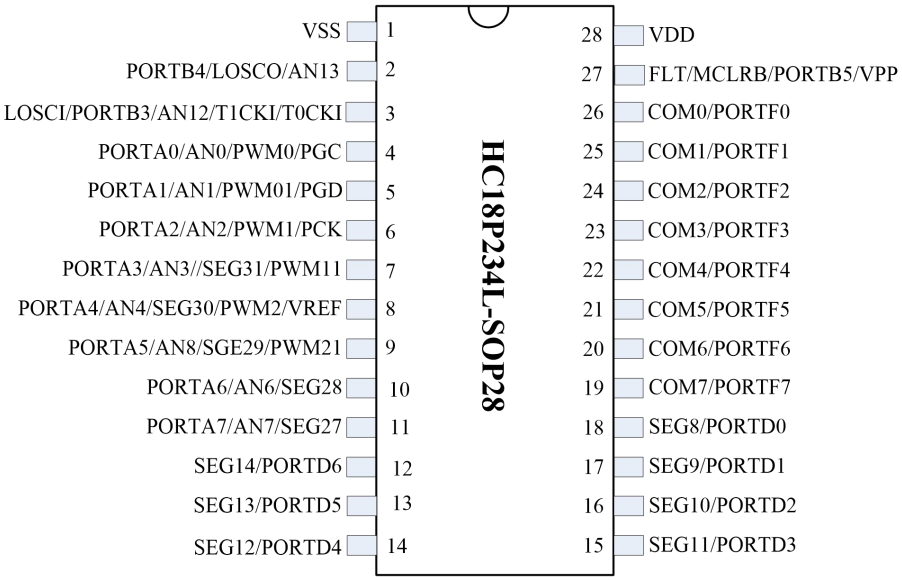
1.3.1 HC18P232L 引脚图



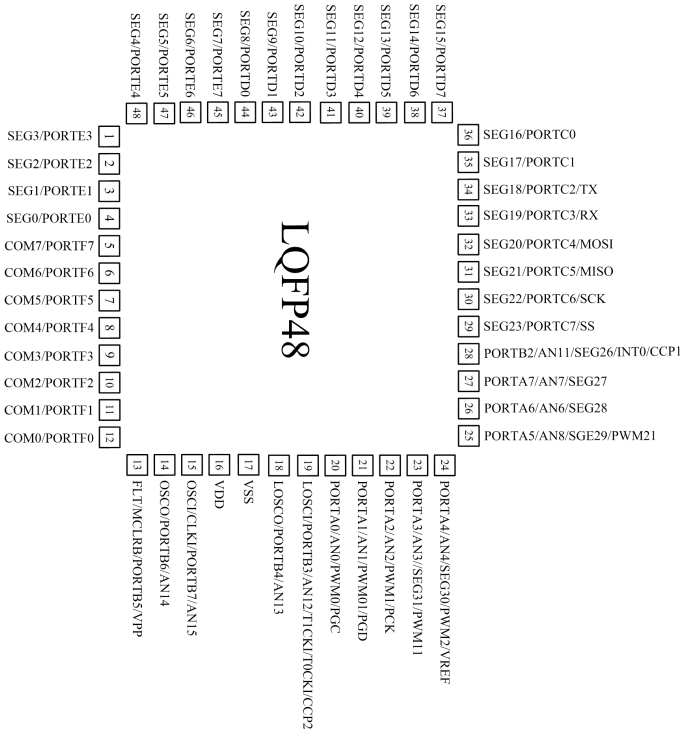
1.3.2 HC18P233L 引脚图



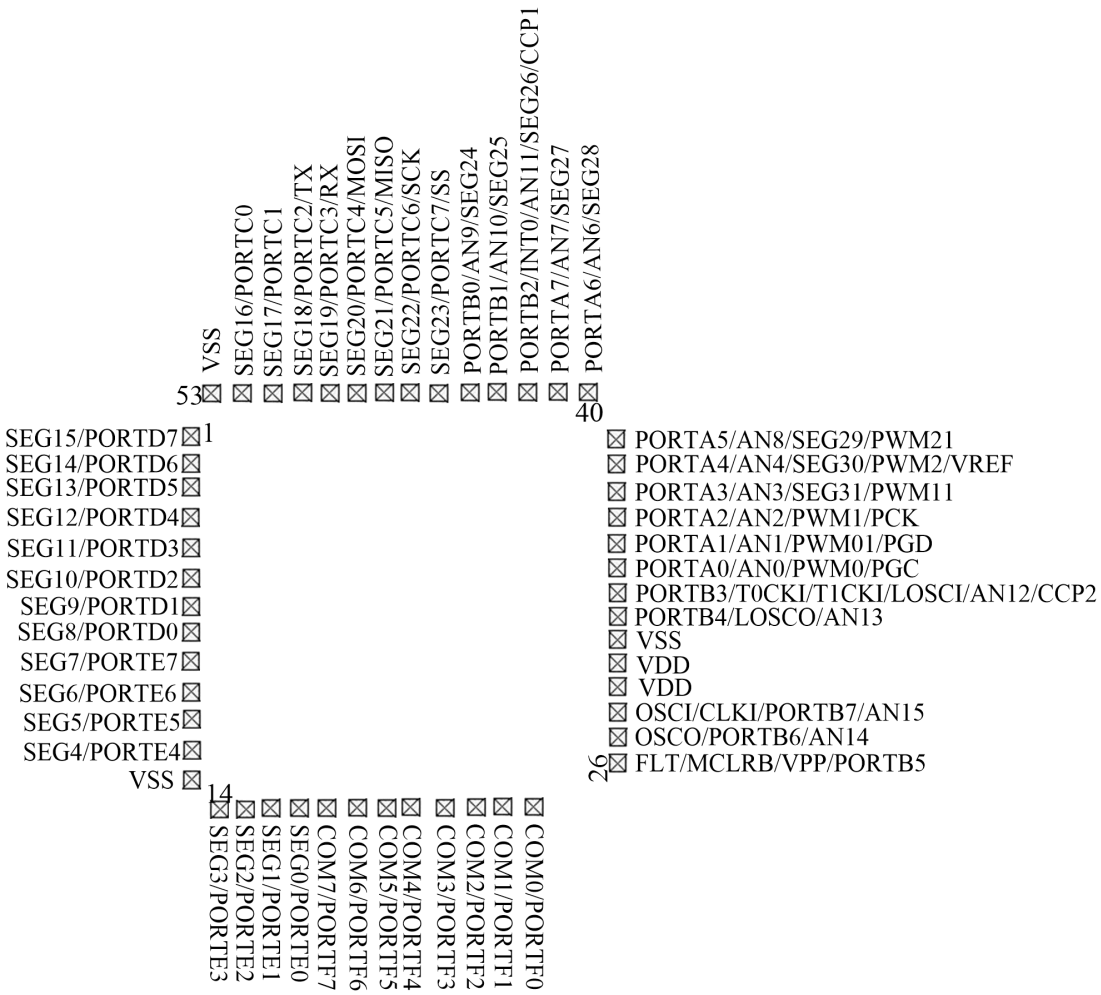
1.3.3 HC18P234L 引脚图



1.3.4 HC18P235L 引脚图



1.3.5 HC18P235L_DICE 引脚图



1.4 引脚说明

LQFP48	名称	类型	说明
1	SEG3 PORTE3	AN I/O	SEG3输出口 输入/输出口，带可编程上下拉，端口电平变化中断
2	SEG2 PORTE2	AN I/O	SEG2输出口 输入/输出口，带可编程上下拉，端口电平变化中断
3	SEG1 PORTE1	AN I/O	SEG1输出口 输入/输出口，带可编程上下拉，端口电平变化中断
4	SEG0 PORTE0	AN I/O	SEG0输出口 输入/输出口，带可编程上下拉，端口电平变化中断
5	COM7 PORTF7	AN I/O	COM7输出口 输入/输出口，带可编程上下拉，端口电平变化中断
6	COM6 PORTF6	AN I/O	COM6输出口 输入/输出口，带可编程上下拉，端口电平变化中断
7	COM5 PORTF5	AN I/O	COM5输出口 输入/输出口，带可编程上下拉，端口电平变化中断
8	COM4 PORTF4	AN I/O	COM4输出口 输入/输出口，带可编程上下拉，端口电平变化中断
9	COM3 PORTF3	AN I/O	COM3输出口 输入/输出口，带可编程上下拉，端口电平变化中断
10	COM2 PORTF2	AN I/O	COM2输出口 输入/输出口，带可编程上下拉，端口电平变化中断
11	COM1 PORTF1	AN I/O	COM1输出口 输入/输出口，带可编程上下拉，端口电平变化中断
12	COM0 PORTF0	AN I/O	COM0输出口 输入/输出口，带可编程上下拉，端口电平变化中断
13	MCLRB VPP PORTB5 FLT	I P I/O I	复位输入口，低电平有效 编程高压电源输入 输入/输出口，只能输出0，端口电平变化中断 PWM故障检测输入口
14	OSCO PORTB6 AN14	AN I/O AN	高频晶体振荡器输输出口 输入/输出口，带可编程上下拉，端口电平变化中断 AD通道14输入口
15	OSCI CLKI PORTB7 AN15	AN I I/O AN	高频晶体振荡器输入口 外部时钟输入口 输入/输出口，带可编程上下拉，端口电平变化中断 AD通道15输入口
16	VDD	P	电源输入
17	VSS	P	电源地
18	PORTB4 LOSCO AN13	I/O AN AN	输入/输出口，带可编程上下拉，端口电平变化中断 低频晶体振荡器输出口 AD通道13输入口
19	PORTB3 T0CKI	I/O I	输入/输出口，带可编程上下拉，端口电平变化中断 Timer0外部时钟输入口（施密特触发器）

	T1CKI	I	Timer1外部时钟输入口（施密特触发器）
	LOSCI	AN	低频晶体振荡器输入口
	AN12	AN	AD通道12输入口
	CCP2	I/O	CCP2输入/输出口。
20	PORTA0	I/O	输入/输出口，带可编程上下拉，端口电平变化中断
	AN0	AN	AD通道0输入口
	PWM0	O	12位PWM0输出口
	PGC	I	编程时钟输入口
21	PORTA1	I/O	输入/输出口，带可编程上下拉，端口电平变化中断
	AN1	AN	AD通道1输入口
	PWM01	O	与PWM0有固定相位关系的12位PWM输出
	PGD	I/O	编程数据输入/输出口
22	PORTA2	I/O	输入/输出口，带可编程上下拉，端口电平变化中断
	AN2	AN	AD通道2输入口
	PWM1	O	8位PWM1输出口
	PCK	O	测试模式下内部RC输出口
23	PORTA3	I/O	输入/输出口，带可编程上下拉，端口电平变化中断
	AN3	AN	AD通道3输入口
	SEG31	AN	SEG31输出口
	PWM11	O	与PWM1有固定相位关系的8位PWM输出
24	PORTA4	I/O	输入/输出口，带可编程上下拉，端口电平变化中断
	AN4	AN	AD通道4输入口
	SEG30	AN	SEG30输出口
	PWM2	O	8位PWM2输出口
	VREF	AN	ADC参考电压输入口
25	PORTA5	I/O	输入/输出口，带可编程上下拉，端口电平变化中断
	AN8	AN	AD通道8输入口
	SEG29	AN	SEG29输出口
	PWM21	0	与PWM2有固定相位关系的8位PWM输出
26	PORTA6	I/O	输入/输出口，带可编程上下拉，端口电平变化中断
	SEG28	AN	SEG28输出口
	AN6	AN	AD通道6输入口
27	PORTA7	I/O	输入/输出口，带可编程上下拉，端口电平变化中断
	SEG27	AN	SEG27输出口
	AN7	AN	AD通道7输入口
28	PORTB2	I/O	输入/输出口，带可编程上下拉，端口电平变化中断
	SEG26	AN	SEG26输出口
	AN11	AN	AD通道11输入口
	CCP1	IO	CCP1输入/输出口。
29	SEG23	AN	SEG23输出口
	PORTC7	I/O	输入/输出口，带可编程上下拉，端口电平变化中断
	SS	I/O	SPI从机片选口
30	SEG22	AN	SEG22输出口
	PORTC6	I/O	输入/输出口，带可编程上下拉，端口电平变化中断
	SCK	I/O	SPI时钟口
31	SEG21	AN	SEG21输出口

	PORTC5 MISO	I/O I/O	输入/输出口，带可编程上下拉，端口电平变化中断 SPI数据输出口，主机的输入和从机的输出
32	SEG20 PORTC4 MOSI	AN I/O I/O	SEG20输出口 输入/输出口，带可编程上下拉，端口电平变化中断 SPI数据输入口，主机的输出和从机的输入
33	SEG19 PORTC3 RX	AN I/O I	SEG19输出口 输入/输出口，带可编程上下拉，端口电平变化中断 UART接收数据端口
34	SEG18 PORTC2 TX	AN I/O O	SEG18输出口 输入/输出口，带可编程上下拉，端口电平变化中断 UART发送数据端口
35	SEG17 PORTC1	AN I/O	SEG17输出口 输入/输出口，带可编程上下拉，端口电平变化中断
36	SEG16 PORTC0	AN I/O	SEG16输出口 输入/输出口，带可编程上下拉，端口电平变化中断
37	SEG15 PORTD7	AN I/O	SEG15输出口 输入/输出口，带可编程上下拉，端口电平变化中断
38	SEG14 PORTD6	AN I/O	SEG14输出口 输入/输出口，带可编程上下拉，端口电平变化中断
39	SEG13 PORTD5	AN I/O	SEG13输出口 输入/输出口，带可编程上下拉，端口电平变化中断
40	SEG12 PORTD4	AN I/O	SEG12输出口 输入/输出口，带可编程上下拉，端口电平变化中断
41	SEG11 PORTD3	AN I/O	SEG11输出口 输入/输出口，带可编程上下拉，端口电平变化中断
42	SEG10 PORTD2	AN I/O	SEG10输出口 输入/输出口，带可编程上下拉，端口电平变化中断
43	SEG9 PORTD1	AN I/O	SEG9输出口 输入/输出口，带可编程上下拉，端口电平变化中断
44	SEG8 PORTD0	AN I/O	SEG8输出口 输入/输出口，带可编程上下拉，端口电平变化中断
45	SEG7 PORTE7	AN I/O	SEG7输出口 输入/输出口，带可编程上下拉，端口电平变化中断
46	SEG6 PORTE6	AN I/O	SEG6输出口 输入/输出口，带可编程上下拉，端口电平变化中断
47	SEG5 PORTE5	AN I/O	SEG5输出口 输入/输出口，带可编程上下拉，端口电平变化中断
48	SEG4 PORTE4	AN I/O	SEG4输出口 输入/输出口，带可编程上下拉，端口电平变化中断

注：I＝ 输入 O＝ 输出 I/O＝ 输入/ 输出 P＝ 电源 AN＝ 模拟输入输出

1.5 引脚电路

图 1-1：PORTA[7:5][3:0]的等效电路

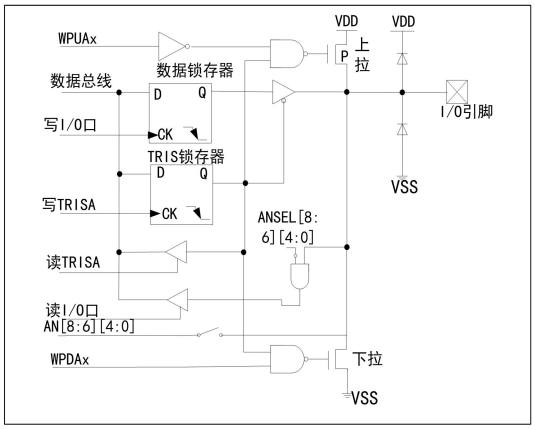


图1-2：PORTA4的等效电路

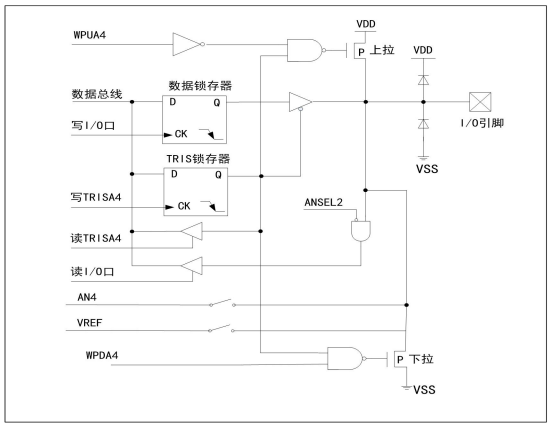


图1-3：PORTB[7:6][4:0]的等效电路

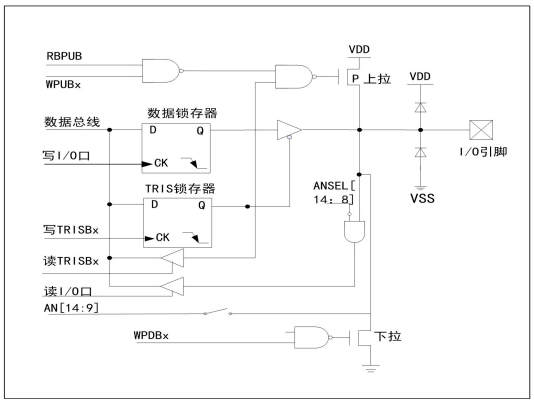


图1-4：PORTB5口的等效电路

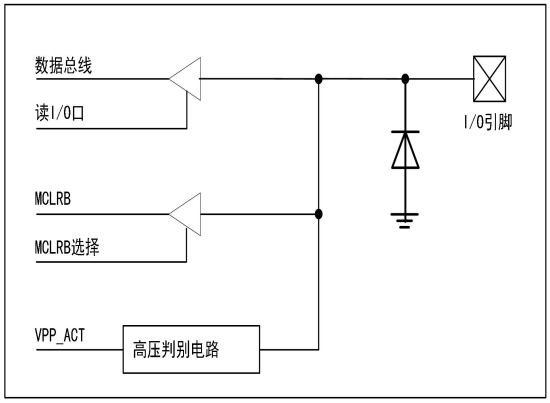


图 1-5：PORTF[7:0]口的等效电路

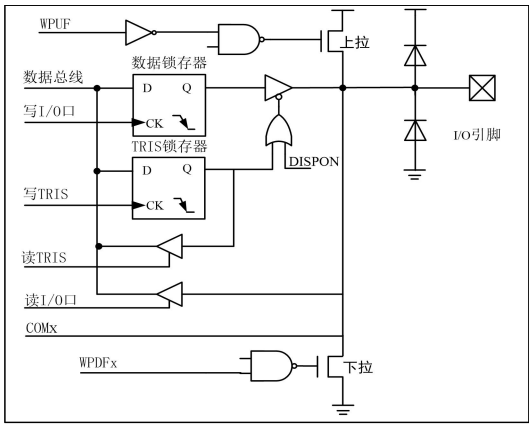
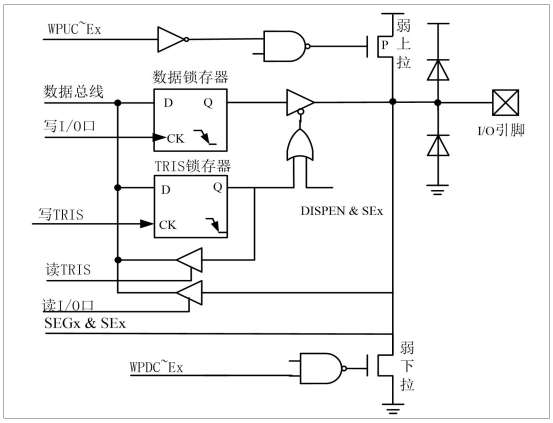


图 1- 6：PORTC、PORTD、PORTE 口等效电路



2 中央处理器（CPU）

HC18P23xL CPU内核包括：

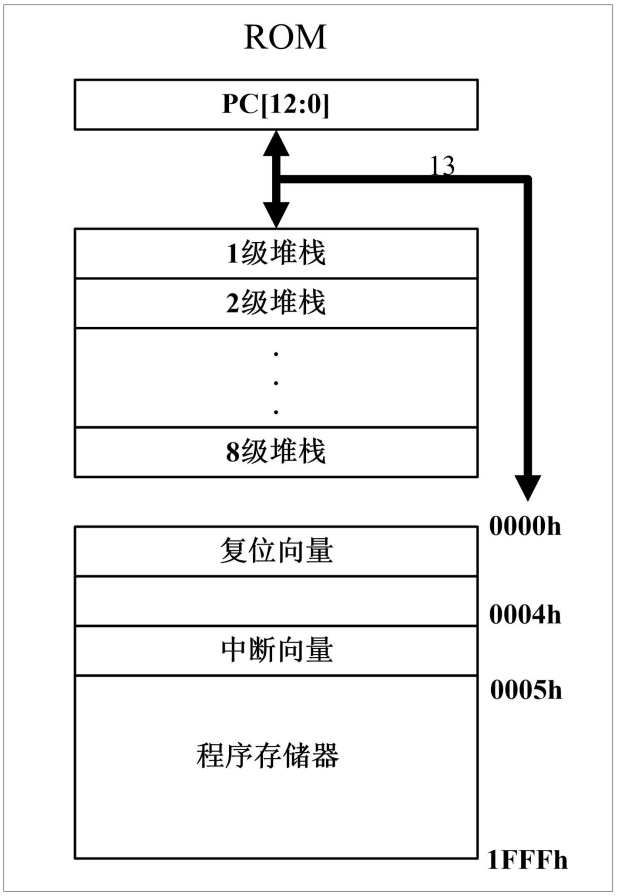
- 2T/4T 时钟模式
- 8 级堆栈
- 程序存储器
- 寻址方式
- 数据存储器

2.1 存储器

2.1.1 程序存储器（OTP-ROM）

HC18P23xL具有8K×16位的程序存储器，下图给出了程序存储器的映射。访问超出物理地址以外的单元时，会导致返回到地址最低单元。

复位向量是0000h，中断向量是0004h。



2.1.1.1 复位向量（0000h）

复位向量为0000h

- **上电复位（POR=0，BOR=X，TO=1）**

- **低电压复位（POR=1，BOR=0，TO=1）**
- **看门狗复位（POR=1，BOR=1，TO=0）**
- **外部复位（POR=1，BOR=1，TO=1）**

发生上述任一种复位后，程序将从0000h处重新开始执行，系统寄存器也都将恢复为默认值。根据PCON寄存器中的POR，BOR标志及STATUS 寄存器中的TO标志位的内容可以判断系统复位方式。下面一段程序演示了如何定义ROM中的复位向量。

- **例：定义复位向量**

```
                ORG         0000H       ;复位向量
                GOTO        MAIN        ;跳转到用户程序
                ...
                ORG         400H        ;用户程序起始
MAIN:           ...
                ...
                END                     ;用户程序结束
```

- **例：复位源判断**

```
                ORG         0000H
                GOTO        RST_JUGE
                ...
RST_JUGE:
                BCF         STATUS,RP0  ;Bank0
                BTFSS       PCON,POR
                GOTO        ISPOR       ;POR标志为0，判定为上电复位
                BTFSS       PCON,BOR
                GOTO        ISBOR       ;POR=1，BOR=0，判定为低电压复位
                BTFSS       STATUS,TO
                GOTO        ISWDTR      ;POR=1，BOR=1，TO=0，判定为WDT复位
EXT_RST:        ...                     ;POR=1，BOR=1，TO=1，判定为外部复位
                ...
ISPOR:          BSF         PCON,POR    ;上电复位处理程序
                ...
ISBOR:          BSF         PCON,BOR    ;低电压复位处理程序
                ...
ISWDTR:         CLRWDT                  ;TO标志置1，WDT复位处理程序
                ...                     ;其他程序，注意处理Bank
```

2.1.1.2 中断向量（0004h）

中断向量地址为0004h。一旦有中断响应，程序计数器PC的当前值就会存入堆栈缓存器并跳转到0004H 开始执行中断服务程序。中断服务子程序中需要对相应状态寄存器进行适当的断点保护和恢复。下面的示例程序说明了如何编写中断服务程序。

- **例：中断子程序的编写**

```
                BTFSS       STATUS,RP0
                GOTO        Bank0_Inter     ;判断进中断前在Bank0还是Bank1
Bank1_Inter:                                ;Bank1处理程序
                MOVWF       W_TEMP1         ;保护W寄存器
                SWAPF       STATUS,W
                MOVWF       STATUS_TEMP1    ;保护STATUS寄存器
                MOVF        PCLATH,W
                MOVWF       PCLATH_TEMP1    ;保护PCLATH寄存器
                BCF         STATUS，RP0
                BSF         Bank_Flag
                GOTO        Interrput_Sev
Bank0_Inter:
                MOVWF       W_TEMP0
                SWAPF       STATUS,W
                MOVWF       STATUS_TEMP0
                MOVF        PCLATH,W
                MOVWF       PCLATH_TEMP0
                BCF         Bank_Flag
                GOTO        Interrput_Sev

Interrput_Sev:
                BCF         STATUS，RP0
                BTFSC       INTCON,INTF
                GOTO        ISR_INT         ;发生INT0中断
                BTFSC       INTCON,T0IF
                GOTO        ISR_T0          ;发生T0中断
                ...
                ...
Exit_Int:
                BCF         STATUS，RP0
                BTFSS       Bank_Flag
                GOTO        Bank0_Exit
Bank1_Exit:
                BSF         STATUS，RP0
                MOVF        PCLATH_TEMP1 ,W
                MOVWF       PCLATH          ;恢复PCLATH
                SWAPF       STATUS_TEMP1,W
                MOVWF       STATUS          ;恢复STATUS
                SWAPF       W_TEMP1,F
                SWAPF       W_TEMP1,W       ;恢复W
                RETFIE                      ;退出中断

Bank0_Exit:
                BCF         STATUS，RP0
```

```
MOVF    PCLATH_TEMP0 ,W
MOVWF   PCLATH          ;恢复PCLATH
SWAPF   STATUS_TEMP0,W
MOVWF   STATUS          ;恢复STATUS
SWAPF   W_TEMP0,F
SWAPF   W_TEMP0,W       ;恢复W
RETFIE                  ;退出中断
```

对于编写中断服务程序，需要以下几个要点需注意：

1. 中断入口地址为 0x04，响应中断后，程序自动跳转到 0x04 开始执行；
2. 中断服务程序需首先对相应的寄存器进行保护；
3. 保存系统寄存器时注意 Bank，如示例代码中，分别定义一组 RAM 保存进入中断前 Bank 的状态；
4. 中断服务子程序返回前对保护的寄存器进行恢复，注意恢复顺序，对 W 必须使用 SWAPF；
5. 程序中使能两个以上的中断源时，程序需对发生中断的中断源进行判断，从而执行相应的服务程序。
6. 需要软件清零对应的中断标志；
7. RETFIE 指令将自动使能 GIE，请勿在中断服务子程序中用其它指令使能 GIE，以免造成中断响应混乱。

2.1.1.3 查表

方式一：

利用 ADDWF PCL，F 和 RETLW 指令实现数据表，因为以 PCL 为目的操作数的运算将改变程序指针（PC）值，其具体操作为 PC 的低 8 位为 ALU 的运算结果，PC 的高 5 位将从 PC 高位缓冲器 PCLATH 中获得。如下是数据表实现的一个例子。

- **例：数据查表**

```
    ...
    MOVLW   HIGH  TAB1    ;获得数据表地址高8位（内部宏指令）
    MOVWF   PCLATH        ;表地址高位赋给PCLATH
    MOVF    TABBUF,W      ;获得表数据偏移量，调用前赋值。
    CALL    TAB1          ;调用数据表
    ...

    ORG     100H
TAB1：
    ADDWF   PCL,F         ;表头运算
    RETLW   DATA0_TAB1    ;W=0对应数据
    RETLW   DATA1_TAB1    ;W=1对应数据
    RETLW   DATA2_TAB1    ;W=2对应数据
    ...
    RETLW   DATAFE_TAB1   ;W=0xFE对应数据
```

对于数据查表的编程，需注意：

1. 数据表宽度：8 位；
2. 数据表无法直接跨页访问，单页可实现最大长度：255；
3. 当 PCL 与 W 的加运算有进位时，进位将被舍弃数据表溢出，将造成查表混乱；故表头尽量放在数据页前端，以免数据表溢出；
4. TABBUF 的值不得大于表长，否则将造成运行混乱。

➢ **例：跳转表**

跳转表能够实现多地址跳转功能。由于 PCL 和 W 的值相加即可得到新的 PCL，同时 PCH 从 PCLATH 中载入，因此，可以通过对 PCL 加上不同的 W 值来实现多地址跳转，可参考以下范例。

```
                …
                ORG             0100H
                MOVLW           HIGH    TAB2    ;获得跳转表地址高位（内部宏指令）
                MOVWF           PCLATH
                MOVF            TABBUF,W
TAB2：          ADDWF           PCL,F
                GOTO            LABLE0_TAB2     ;TABBUF =0,跳转  LABLE0_TAB2
                GOTO            LABLE1_TAB2     ;以下类推
                GOTO            LABLE2_TAB2
                GOTO            LABLE3_TAB2
```

注：

如上跳转表，有 4 个跳转分支，TABBUF 的合法范围为 0x00～0x03。

方式二：

可以通过以下5个特殊功能寄存器对ROM区中的数据进行查找。

PMCON

PMDATL

PMDATH

PMADRL

PMADRH

寄存器 PMADRH 指向 ROM 区数据地址的高字节（bit8~bit15），寄存器 PMADRL 指向 ROM 区数据地址的低字节（bit0~bit7）。将 PMCON 寄存器的 RDON 位置 1 启动读操作，使用两条指令来读数据，RDON 位置 1 后的二条指令被自动忽略，建议用户 RDON 位置 1 后的两条指令为 NOP。执行完读操作后，所查找的数据保存在 PMDATH:PMDATL 寄存器。

➢ **例：查找ROM 地址为“TABLE”的值**

```
                BCF             STATUS,RP0      ;BANK0
                MOVF            TABLE _ADDR_H, W
                MOVWF           PMADRH          ;设置TABLE地址高字节
                MOVF            TABLE _ADDR_L, W
                MOVWF           PMADRL          ;设置TABLE地址低字节
                BSF             PMCON, RDON     ;开始读
```

```
                NOP
                NOP                             ;等待两条指令
                MOVF        PMDATL, W
                MOVWF       TABLE _DATAL  ;TABLE _DATAL= TABLE地址数据低字节
                MOVF        PMDATH, W
                MOVWF       TABLE _DATAH  ;TABLE _DATAH= TABLE地址数据高字节
                ...
                ...
TABLE:          DW          1234H           ;定义数据表（16 位）数据。
                DW          F178H
                DW          2123H
```

9Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMCON	-	-	-	-	-	-	-	RDON
R/W	-	-	-	-	-	-	-	R/W
POR的值	-	-	-	-	-	-	-	0

bit [0] **RDON**：读控制位

0 =不启动ROM存储器读操作

1 =启动ROM读操作（由硬件清零RDON；软件只能将RDON位置1，但不能清零）

9Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMDATL	PMD7	PMD6	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	x	x	x	x	x	x

9Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMDATH	PMD15	PMD14	PMD13	PMD12	PMD11	PMD10	PMD9	PMD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	x	x	x	x	x	x

PMDx[15:0]：ROM存储器读操作后， PMADRH:PMADRL 指向地址的数据

9Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMADRL	PMA7	PMA6	PMA5	PMA4	PMA3	PMA2	PMA1	PMA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	x	x	x	x	x	x

9Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMADRH	PMA15	PMA14	PMA13	PMA12	PMA11	PMA10	PMA9	PMA8
R/W	R/W	- R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	x	x	x	x	x	x

PMAx[15:0]：ROM存储器地址

2.1.2 芯片配置选择表

HC18P23xL提供的配置字可以对多个系统模块做配置选择，详细配置选择见下面表格。系统还提供了一个4×16bit的器件ID供用户存储校验或其他代码标识号。**在程序运行过程中不能访问这些存储单元，但可在编程烧录/校验时对它们进行读写。**

编译选项	内容	功能说明
高频系统时钟选择(OSCHM)	高频晶体振荡器	高频晶体振荡器 OSCI/OSCO 作为高频晶体振荡器输入/输出口。
	内部 RC 振荡器	内部 16M RC 振荡器。
低频系统时钟选择(OSCLM)	32K WDT 振荡器	内部 32K WDT 振荡器。
	定时器 1 振荡器	低频晶体振荡器，32.768KHz，LOSCI/LOSCO 为低频晶体振荡器输入/输出口。
高频内部 RC 振荡器频率选择(ROSC)	16MHz	内部高频 RC 振荡器频率配置为 16MHz。
	8MHz	内部高频 RC 振荡器频率配置为 8MHz。
	4MHz	内部高频 RC 振荡器频率配置为 4MHz。
	2MHz	内部高频 RC 振荡器频率配置为 2MHz。
	1MHz	内部高频 RC 振荡器频率配置为 1MHz。
	500KHz	内部高频 RC 振荡器频率配置为 500KHz。
WDT 功能使能(WDTEN)	使能 WDT 功能	使能看门狗定时器功能。
	屏蔽 WDT 功能	屏蔽看门狗定时器功能。
外部复位使能(MCLREN)	使能外部复位	使能外部复位引脚。
	屏蔽，做输入	屏蔽外部复位引脚，外部复位引脚做输入功能。
加密使能位(CP0)	加密	使能用户程序区 CODE 加密功能。
	不加密	屏蔽用户程序区 CODE 加密功能。
启动时钟选择(SPDS)	高频系统时钟	系统选择高频系统时钟作为启动时钟。
	低频系统时钟	系统选择低频系统时钟作为启动时钟。
ADC 使能	打开 ADC 功能	使能 ADC 功能。
	禁止 ADC 功能	禁止 ADC 功能。
PORSEL 选择	18ms	上电复位时间选择为 18ms。
	4.5ms	上电复位时间选择为 4.5ms。
SMTENB	IO 口施密特使能	使能 IO 口施密特功能。
	IO 口施密特禁止	禁止 IO 口施密特功能。
BORSEL	BOR3.6V	当系统电压低于 3.6V 时，系统复位。
	BOR2.4V	当系统电压低于 2.4V 时，系统复位。
	BOR2.0V	当系统电压低于 2.0V 时，系统复位。
时钟模式选择(FCPUT)	4T	4T 模式，1 个指令周期有 4 个系统时钟周期组成。
	2T	2T 模式，1 个指令周期有 2 个系统时钟周期组成。

选择芯片配置字注意事项：

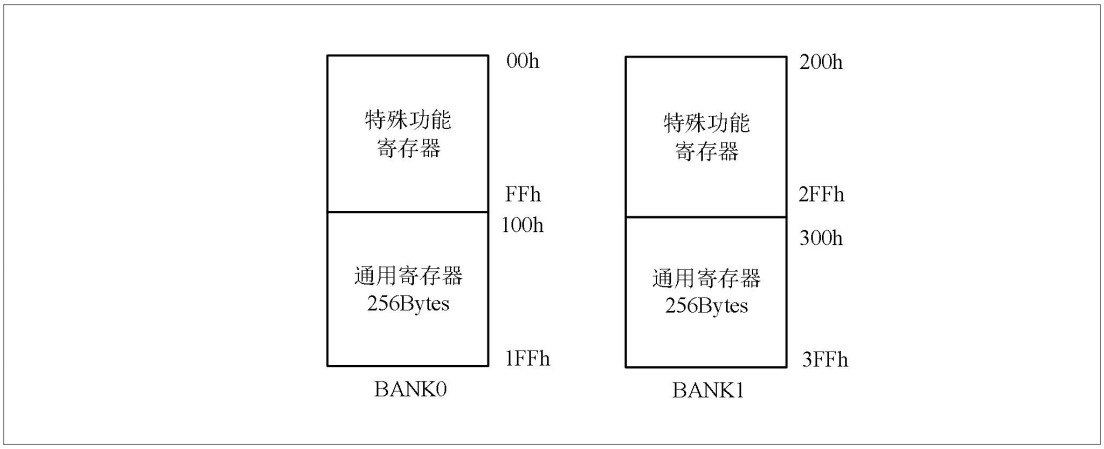
1. 在系统允许的情况下，尽量选用较低系统时钟频率，有利于降低系统功耗和提升系统电磁兼容性；
2. 时钟模式选择为 2T 时，PWM 模块的最大分辨率降低到 9 位；
3. 强干扰情况下，建议开启 WDT 功能。

2.1.3 通用数据寄存器（RAM）

HC18P23xL共有512个通用寄存器（GPR）和182个特殊功能寄存器（SFR），分在2个存储区Bank0和Bank1，每个存储区的低256个地址单元保留为特殊功能寄存器，RP0是存储区的选择位。

CORE Register 00-09h&200-209h

00h&200h	INDF0	0	INDF0	间接寻址 0 寄存器（不是实际存在的物理寄存器）						
01h&201h	INDF1	1	INDF1	间接寻址 1 寄存器（不是实际存在的物理寄存器）						
02h&202h	PCL	2	PCL	程序计数器（PC）低字节						
03h&203h	STATUS	3	STATUS			RP0	TO	PD	Z	DC	C
04h&204h	FSR0L	4	FSR0L	间接寻址 0 地址低位指针						
05h&205h	FSR0H	5	FSR0H	间接寻址 0 地址高位指针						
06h&206h	FSR1L	6	FSR1L	间接寻址 1 地址低位指针						
07h&207h	FSR1H	7	FSR1H	间接寻址 1 地址高位指针						
08h&208h	PCLATH	8	PCLATH				程序计数器高 5 位			
09h&209h	INTCON	9	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF



2.1.4 特殊功能寄存器（SFR）

2.1.4.1 特殊功能寄存器列表

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位初值
BANK0										
010h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111
011h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111
012h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111
013h	TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111
014h	TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	1111 1111
015h	TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	1111 1111
01Ch	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	0000 0000
01Dh	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	0000 0000
01Eh	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	0000 0000
01Fh	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	0000 0000
020h	PORTE	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	0000 0000

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位初值
021h	PORTF	PORTF7	PORTF6	PORTF5	PORTF4	PORTF3	PORTF2	PORTF1	PORTF0	0000 0000
028h	WPUA	WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	1111 1111
029h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	1111 1111
02Ah	WPUC	WPUC7	WPUC6	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	1111 1111
02Bh	WPUD	WPUD7	WPUD6	WPUD5	WPUD4	WPUD3	WPUD2	WPUD1	WPUD0	1111 1111
02Ch	WPUE	WPUE7	WPUE6	WPUE5	WPUE4	WPUE3	WPUE2	WPUE1	WPUE0	1111 1111
02Dh	WPUF	WPUF7	WPUF6	WPUF5	WPUF4	WPUF3	WPUF2	WPUF1	WPUF0	1111 1111
034h	WPDA	WPDA7	WPDA6	-	WPDA4	WPDA3	WPDA2	WPDA1	WPDA0	11-1 1111
035h	WPDB	WPDB7	WPDB6	WPDB5	WPDB4	WPDB3	WPDB2	WPDB1	WPDB0	1111 1111
036h	WPDC	WPDC7	WPDC6	WPDC5	WPDC4	WPDC3	WPDC2	WPDC1	WPDC0	1111 1111
037h	WPDD	WPDD7	WPDD6	WPDD5	WPDD4	WPDD3	WPDD2	WPDD1	WPDD0	1111 1111
038h	WPDE	WPDE7	WPDE6	WPDE5	WPDE4	WPDE3	WPDE2	WPDE1	WPDE0	1111 1111
039h	WPDF	WPDF	WPDF7	WPDF6	WPDF5	WPDF4	WPDF3	WPDF2	WPDF1	1111 1111
040h	IOCA	IOCA7	IOCA6	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	0000 0000
041h	IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0	0000 0000
042h	IOCC	IOCC7	IOCC6	IOCC5	IOCC4	IOCC3	IOCC2	IOCC1	IOCC0	0000 0000
043h	IOCD	IOCD7	IOCD6	IOCD5	IOCD4	IOCD3	IOCD2	IOCD1	IOCD0	0000 0000
044h	IOCE	IOCE7	IOCE6	IOCE5	IOCE4	IOCE3	IOCE2	IOCE1	IOCE0	0000 0000
045h	IOCF	IOCF7	IOCF6	IOCF5	IOCF4	IOCF3	IOCF2	IOCF1	IOCF0	0000 0000
04Ch	PORCTR	-	-	-	CCPCT	SPPCT1	SPPCT0	UAPCT1	UAPCT0	---0 0000
04Dh	LAEN	LAEN7	LAEN6	LAEN5	LAEN4	LAEN3	LAEN2	LAEN1	LAEN0	0000 0000
04Eh	LBEN	LBEN7	LBEN6	LBEN5	LBEN4	LBEN3	LBEN2	LBEN1	LBEN0	0000 0000
04Fh	LCEN	LCEN7	LCEN6	LCEN5	LCEN4	LCEN3	LCEN2	LCEN1	LCEN0	0000 0000
050h	LDEN	LDEN7	LDEN6	LDEN5	LDEN4	LDEN3	LDEN2	LDEN1	LDEN0	0000 0000
051h	LEEN	LEEN7	LEEN6	LEEN5	LEEN4	LEEN3	LEEN2	LEEN1	LEEN0	0000 0000
052h	LFEN	LFEN7	LFEN6	LFEN5	LFEN4	LFEN3	LFEN2	LFEN1	LFEN0	0000 0000
054h	PIR1	-	ADIF	-	-	-	CCP1IF	T2IF	T1IF	-0-- -000
055h	PIR2	PWM2IF	PWM1IF	PWM0IF	-	RXIF	TXIF	SPIF	CCP2IF	000- 0000
056h	PIR3	-	-	-	RFIF	REIF	RDIF	RCIF	RAIF	---0 0000
058h	T1L	Timer1 计数寄存器低字节								xxxx xxxx
059h	T1H	Timer1 计数寄存器高字节								xxxx xxxx
05Ah	T1CON	T1CS1	T1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	-	T1ON	0000 00-0
05Bh	T0	Timer0 计数寄存器								xxxx xxxx
05Ch	T2	Timer2 计数寄存器								xxxx xxxx
05Dh	PR2	Timer2 周期寄存器								0000 0000
05Eh	T2CON	-	T2CKPS3	T2CKPS2	T2CKPS1	T2CKPS0	T2ON	-	-	-000 00--
05Fh	PR1L	Timer1 周期寄存器低字节								xxxx xxxx
060h	PR1CON	PWM1T1	PWM1T0	PWM2T1	PWM2T0	T1CKPS3	T1CKPS2	PWMPR1	PR1EN	0000 0000
070h	PIE1	-	ADIE	-	-	-	CCP1IE	T2IE	T1IE	-0-- -000

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位初值
071h	PIE2	PWM2IE	PWM1IE	PWM0IE	-		UARTIE	SPIE	CCP2IE	000- -000
072h	PIE3	-	-	-	RFIE	REIE	RDIE	RCIE	RAIE	---0 0000
078h	OPTION	RBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
079h	PCON	LVD2EN	LVD1EN	-	WDTENS	LVD2F	LVD1F	POR	BOR	00-1 qqqq
07Ah	OSCCON	T0OSCEN	-	-	-	-	-	HXEN	SCS	0--- --0q
080h	CCPR2L	CCP2 寄存器低字节								xxxx xxxx
081h	CCPR2H	CCP2 寄存器高字节								xxxx xxxx
082h	CCP2CON	-	-	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000
083h	CCPR1L	CCP1 寄存器低字节								0000 0000
084h	CCPR1H	CCP1 寄存器高字节								0000 0000
085h	CCP1CON	-	-	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000
08Ch	ANSELL	ANSEL7	ANSEL6	-	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0	11-1 1111
08Dh	ANSELH	ANSEL15	ANSEL14	ANSEL13	ANSEL12	ANSEL11	ANSEL10	ANSEL9	ANSEL8	1111 1111
092h	ADRESL	ADC 结果寄存器低字节								0000 0000
093h	ADRESH	ADC 结果寄存器高字节								0000 0000
094h	ADCON0	VHS1	VHS0	CHS3	CHS2	CHS1	CHS0	ADON	ADEN	0000 0000
095h	ADCON1	ADFM	ADCS2	ADCS1	ADCS0	-	-	-	ADREF	0000 ---0
096h	ADCLK	-	-	-	-	-	ADCLK2	ADCLK1	ADCLK0	---- -000
09Ah	PMDATL	程序存储器读数据寄存器的低字节								0000 0000
09Bh	PMDATH	程序存储器读数据寄存器的高字节								0000 0000
09Ch	PMADRL	程序存储器读地址寄存器的低字节								0000 0000
09Dh	PMADRH	程序存储器读地址寄存器的高字节								0000 0000
09Eh	PMCON	-	-	-	-	-	-	-	RDON	---- ---0
BANK1										
230h	SPSTAT	-	-	-	-	-	RXOV	MODF	WCOL	---- -000
231h	SPCTL	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	0000 0000
232h	SPDAT	SPI 数据寄存器								0000 0000
23Ah	BRT	波特率发生器寄存器								0000 0000
23Bh	AUXR	-	UARTEN	UARTM0	BRTR	BRTX12	S1BRS	SMOD	SMOD0	-000 0000
23Ch	SCON	SM0/FE	SM1	SM2	REN	TR8	RB8	RXWK	-	0000 000-
23Dh	SBUF	串行口缓冲寄存器								0000 0000
23Eh	SADEN	从机地址寄存器								0000 -000
23Fh	SADDR	从机地址掩码寄存器								00-0 0000
250h	PWM2DT	DT2.7	DT2.6	DT2.5	DT2.4	DT2.3	DT2.2	DT2.1	DT2.0	0000 0000
251h	PWM2D	PD2.7	PD2.6	PD2.5	PD2.4	PD2.3	PD2.2	PD2.1	PD2.0	0000 0000
252h	PWM2P	PP2.7	PP2.6	PP2.5	PP2.4	PP2.3	PP2.2	PP2.1	PP2.0	0000 0000
253h	PWM2C	-	-	-	-	PWM2S1	PWM2S0	CK21	CK20	---- 0000
254h	PWM1DT	DT1.7	DT1.6	DT1.5	DT1.4	DT1.3	DT1.2	DT1.1	DT1.0	0000 0000
255h	PWM1D	PD1.7	PD1.6	PD1.5	PD1.4	PD1.3	PD1.2	PD1.1	PD1.0	0000 0000

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位初值
256h	PWM1P	PP1.7	PP1.6	PP1.5	PP1.4	PP1.3	PP1.2	PP1.1	PP1.0	0000 0000
257h	PWM1C	-	-	-	-	PWM1S1	PWM1S0	CK11	CK10	---- 0000
258h	PWM0DT	DT0.7	DT0.6	DT0.5	DT0.4	DT0.3	DT0.2	DT0.1	DT0.0	0000 0000
259h	PWM0DL	PD0.7	PD0.6	PD0.5	PD0.4	PD0.3	PD0.2	PD0.1	PD0.0	0000 0000
25Ah	PWM0DH	-	-	-	-	PD0.11	PD0.10	PD0.9	PD0.8	---- 0000
25Bh	PWM0PL	PP0.7	PP0.6	PP0.5	PP0.4	PP0.3	PP0.2	PP0.1	PP0.0	0000 0000
25Ch	PWM0PH	-	-	-	-	PP0.11	PP0.10	PP0.9	PP0.8	---- 0000
25Dh	PWM0C	-	-	FLTS	FLTC	PWM0S1	PWM0S0	CK01	CK00	--00 0000
25Eh	PWMEN	-	EFLT	EPWM21	EPWM11	EPWM01	EPWM2	EPWM1	EPWM0	-000 0000
25Fh	FLTM	-	-	FLT2M1	FLT2M0	FLT1M1	FLT1M0	FLT0M1	FLT0M0	--00 0000
2B0h	LCDCON	BIASCT1	BIASCT0	DUTCT	-	CS1	CS0	RLCD1	RLCD0	000- 0000
2B1h	LCDPS	-	-	-	-	LP3	LP2	LP1	LP0	---- 0000
2B2h	DISPCTR	-	-	-	-	-	SLPENB	DISPON	DISPCT	---- -000
2B3h	LCDLVD	-	-	-	-	LCDVD3	LCDVD2	LCDVD1	LCDVD0	---- 0000
2B6h	SEGSE0	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	0000 0000
2B7h	SEGSE1	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	0000 0000
2B8h	SEGSE2	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16	0000 0000
2B9h	SEGSE3	SE31	SE30	SE29	SE28	SE27	SE26	SE25	SE24	0000 0000
2C0h	SEGDATA0	SEG0/C7	SEG0/C6	SEG0/C5	SEG0/C4	SEG0/C3	SEG0/C2	SEG0/C1	SEG0/C0	xxxx xxxx
2C1h	SEGDATA1	SEG1/C7	SEG1/C6	SEG1/C5	SEG1/C4	SEG1/C3	SEG1/C2	SEG1/C1	SEG1/C0	xxxx xxxx
2C2h	SEGDATA2	SEG2/C7	SEG2/C6	SEG2/C5	SEG2/C4	SEG2/C3	SEG2/C2	SEG2/C1	SEG2/C0	xxxx xxxx
2C3h	SEGDATA3	SEG3/C7	SEG3/C6	SEG3/C5	SEG3/C4	SEG3/C3	SEG3/C2	SEG3/C1	SEG3/C0	xxxx xxxx
2C4h	SEGDATA4	SEG4/C7	SEG4/C6	SEG4/C5	SEG4/C4	SEG4/C3	SEG4/C2	SEG4/C1	SEG4/C0	xxxx xxxx
2C5h	SEGDATA5	SEG5/C7	SEG5/C6	SEG5/C5	SEG5/C4	SEG5/C3	SEG5/C2	SEG5/C1	SEG5/C0	xxxx xxxx
2C6h	SEGDATA6	SEG6/C7	SEG6/C6	SEG6/C5	SEG6/C4	SEG6/C3	SEG6/C2	SEG6/C1	SEG6/C0	xxxx xxxx
2C7h	SEGDATA7	SEG7/C7	SEG7/C6	SEG7/C5	SEG7/C4	SEG7/C3	SEG7/C2	SEG7/C1	SEG7/C0	xxxx xxxx
2C8h	SEGDATA8	SEG8/C7	SEG8/C6	SEG8/C5	SEG8/C4	SEG8/C3	SEG8/C2	SEG8/C1	SEG8/C0	xxxx xxxx
2C9h	SEGDATA9	SEG9/C7	SEG9/C6	SEG9/C5	SEG9/C4	SEG9/C3	SEG9/C2	SEG9/C1	SEG9/C0	xxxx xxxx
2CAh	SEGDATA10	SEG10/C7	SEG10/C6	SEG10/C5	SEG10/C4	SEG10/C3	SEG10/C2	SEG10/C1	SEG10/C0	xxxx xxxx
2CBh	SEGDATA11	SEG11/C7	SEG11/C6	SEG11/C5	SEG11/C4	SEG11/C3	SEG11/C2	SEG11/C1	SEG11/C0	xxxx xxxx
2CCh	SEGDATA12	SEG12/C7	SEG12/C6	SEG12/C5	SEG12/C4	SEG12/C3	SEG12/C2	SEG12/C1	SEG12/C0	xxxx xxxx
2CDh	SEGDATA13	SEG13/C7	SEG13/C6	SEG13/C5	SEG13/C4	SEG13/C3	SEG13/C2	SEG13/C1	SEG13/C0	xxxx xxxx
2CEh	SEGDATA14	SEG14/C7	SEG14/C6	SEG14/C5	SEG14/C4	SEG14/C3	SEG14/C2	SEG14/C1	SEG14/C0	xxxx xxxx
2CFh	SEGDATA15	SEG15/C7	SEG15/C6	SEG15/C5	SEG15/C4	SEG15/C3	SEG15/C2	SEG15/C1	SEG15/C0	xxxx xxxx
2D0h	SEGDATA16	SEG16/C7	SEG16/C6	SEG16/C5	SEG16/C4	SEG16/C3	SEG16/C2	SEG16/C1	SEG16/C0	xxxx xxxx
2D1h	SEGDATA17	SEG17/C7	SEG17/C6	SEG17/C5	SEG17/C4	SEG17/C3	SEG17/C2	SEG17/C1	SEG17/C0	xxxx xxxx
2D2h	SEGDATA18	SEG18/C7	SEG18/C6	SEG18/C5	SEG18/C4	SEG18/C3	SEG18/C2	SEG18/C1	SEG18/C0	xxxx xxxx
2D3h	SEGDATA19	SEG19/C7	SEG19/C6	SEG19/C5	SEG19/C4	SEG19/C3	SEG19/C2	SEG19/C1	SEG19/C0	xxxx xxxx
2D4h	SEGDATA20	SEG20/C7	SEG20/C6	SEG20/C5	SEG20/C4	SEG20/C3	SEG20/C2	SEG20/C1	SEG20/C0	xxxx xxxx

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	xxxx xxxx
2D5h	SEGDATA21	SEG21/C7	SEG21/C6	SEG21/C5	SEG21/C4	SEG21/C3	SEG21/C2	SEG21/C1	SEG21/C0	xxxx xxxx
2D6h	SEGDATA22	SEG22/C7	SEG22/C6	SEG22/C5	SEG22/C4	SEG22/C3	SEG22/C2	SEG22/C1	SEG22/C0	xxxx xxxx
2D7h	SEGDATA23	SEG23/C7	SEG23/C6	SEG23/C5	SEG23/C4	SEG23/C3	SEG23/C2	SEG23/C1	SEG23/C0	xxxx xxxx
2D8h	SEGDATA24	SEG24/C7	SEG24/C6	SEG24/C5	SEG24/C4	SEG24/C3	SEG24/C2	SEG24/C1	SEG24/C0	xxxx xxxx
2D9h	SEGDATA25	SEG25/C7	SEG25/C6	SEG25/C5	SEG25/C6	SEG25/C3	SEG25/C2	SEG25/C1	SEG25/C0	xxxx xxxx
2DAh	SEGDATA26	SEG26/C7	SEG26/C6	SEG26/C5	SEG26/C4	SEG26/C3	SEG26/C2	SEG26/C1	SEG26/C0	xxxx xxxx
2DBh	SEGDATA27	SEG27/C7	SEG27/C6	SEG27/C5	SEG27/C4	SEG27/C3	SEG27/C2	SEG27/C1	SEG27/C0	xxxx xxxx
2DCh	SEGDATA28	SEG28/C7	SEG28/C6	SEG28/C5	SEG28/C4	SEG28/C3	SEG28/C2	SEG28/C1	SEG28/C0	xxxx xxxx
2DDh	SEGDATA29	SEG29/C7	SEG29/C6	SEG29/C5	SEG29/C4	SEG29/C3	SEG29/C2	SEG29/C1	SEG29/C0	xxxx xxxx
2DEh	SEGDATA30	SEG30/C7	SEG30/C6	SEG30/C5	SEG30/C4	SEG30/C3	SEG30/C2	SEG30/C1	SEG30/C0	xxxx xxxx
2DFh	SEGDATA31	SEG31/C7	SEG31/C6	SEG31/C5	SEG31/C4	SEG31/C3	SEG31/C2	SEG31/C1	SEG31/C0	xxxx xxxx

注： x = 未知， u = 不变， q = 取值视条件而定， — = 未实现

2.1.4.2 累加器

8 位数据寄存器W用来执行ALU与数据存储器之间数据的传送操作。如果操作结果为零（Z）或有进位产生（C或DC），程序状态寄存器STATUS中相应位会发生变化。

W 并不在RAM中，因此不可以用直接寻址和间接寻址模式对其进行读写。

- 例：W寄存器的读写操作

立即数写入W寄存器操作：

```
MOVLW       H'0FF'        ;送十六进制数
MOVLW       D'10'         ;送十进制数
MOVLW       B'11110000'   ;送二进制数
```

将W寄存器的数据写入数据寄存器BUF中：

```
MOVWF       BUF
```

将数据寄存器BUF中的数读入W寄存器：

```
MOVF        BUF,W
```

将W寄存器的数据与BUF中的数据加法运算后，结果存入BUF中：

```
ADDWF       BUF,F
```

2.1.4.3 INDFx寄存器

INDF0、INDF1 寄存器不是实际存在的寄存器，寻址 INDF0、INDF1 将实现间接寻址。

2.1.4.4 程序计数器

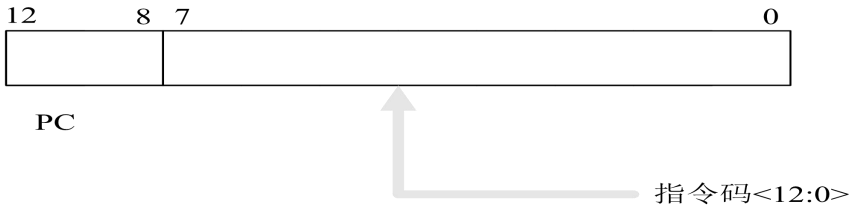
程序计数器（PC）为13位宽，低字节来自可读写的PCL寄存器，高字节（PC[12:8]）不可读写，可通过PCLATH 寄存器间接写入。

02h&202h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

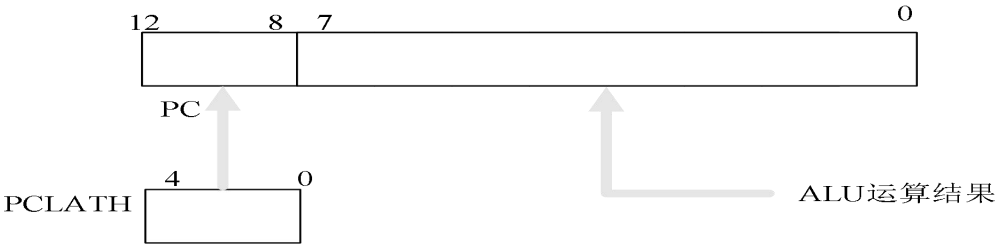
08h&208h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCLATH	-	-	-	PCH12	PCH11	PCH10	PCH9	PCH8
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
POR的值	-	-	-	0	0	0	0	0

程序计数器顺序指令运行时，每个指令周期程序自动加1，以下三种情况将使程序计数器重新装载。

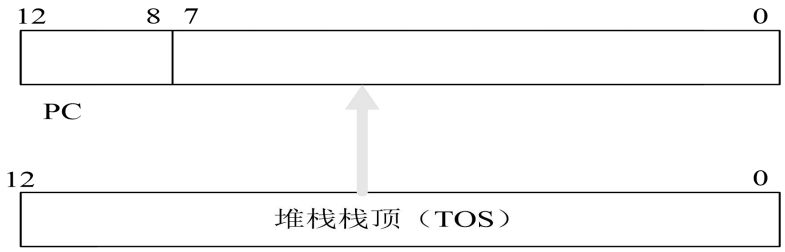
分支指令（GOTO/CALL）：



以PCL作为目的操作数的指令：



子程序返回指令（RETURN/RETLW/RETFIE）：



2.1.4.5 STATUS寄存器

STATUS寄存器包含ALU的算术状态、复位状态和寄存器的存储区选择位。

03&203h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	-	-	RP0	TO	PD	Z	DC	C
R/W	-	-	R/W	R	R	R/W	R/W	R/W
POR的值	-	-	0	1	1	x	x	x

Bit [5] **RP0**：BANK选择位

1 = Bank1

0 = Bank0

Bit [4] **TO**：超时位

1 = 上电、执行了CLRWDT指令或SLEEP指令

0 = 发生了WDT溢出

Bit [3] **PD**：掉电位

1 = 上电或执行了CLRWDT指令

0 = 执行了SLEEP指令

Bit [2] **Z：** 结果为零位

1 = 算术或逻辑运算的结果为零

0 = 算术或逻辑运算的结果不为零

Bit [1] **DC：** 半进位/借位位

1 = 加法运算时低四位有进位/减法运算时没有向高四位借位

0 = 加法运算时低四位没有进位/减法运算时有向高四位借位

Bit [0] **C：** 进位/借位位

1 = 加法运算时有进位/减法运算时没有借位发生/移位后移出逻辑1

0 = 加法运算时没有进位/减法运算时有借位发生/移位后移出逻辑0

2.1.4.6 PCON寄存器

079h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	LVD2EN	LVD1EN	-	WDTENS	LVD2F	LVD1F	POR	BOR
R/W	R/W	R/W	-	R/W	R	R	R/W	R/W
POR的值	0	0	-	1	q	q	q	q

注： - = 未实现，q = 取值视条件而定

Bit [7] **LVD2EN**:3.6V低电压检测使能位

1 = 使能3.6V低电压检测

0 = 禁止3.6V低电压检测

Bit [6] **LVD1EN**:2.4V低电压检测使能位

1 = 使能2.4V低电压检测

0 = 禁止2.4V低电压检测

Bit [4] **WDTENS**:看门狗软件使能位（需要配置字中使能WDT，该位使能时才有效）

1 = 软件使能看门狗

0 = 软件禁止看门狗

Bit [3] **LVD2F**:3.6V低电压检测标志位

1 = 电压低于3.6V

0 = 电压高于3.6V

Bit [2] **LVD1F**:2.4V低电压检测标志位

1 = 电压低于2.4V

0 = 电压高于2.4V

Bit [1] **POR**:上电复位状态位

1 = 非上电复位

0 = 上电复位（需要软件置1）

Bit [0] **BOR**:欠压复位状态位

1 = 未发生欠压复位

0 = 发生了欠压复位（需要软件置1）

注：

1、在芯片执行 SLEEP 指令前，软件关闭看门狗定时器，可以节省休眠或绿色模式下芯片功耗；

2、LVD2EN 和 LVD1EN 检测电平值仅作为设计参考，不能用作芯片工作电压值得精确检测。

2.2 寻址模式

HC18P23xL 共有三种寻址方式：立即寻址、直接寻址和间接寻址模式

2.2.1 立即寻址

立即数参与运算的寻址方式

- 例：立即寻址

```
        ADDLW   06h     ; W 的内容加 6，结果放入 W
```

2.2.2 直接寻址

寄存器参与运算的寻址方式

- 例：直接寻址

```
        MOVWF   OPTION  ; W 的内容装入 OPTION
```

2.2.3 间接寻址

由指针 FSR（FSRxL/FSRxH）指向的寄存器参与运算的寻址方式。INDFx 寄存器不是物理寄存器，对 INDFx 寄存器操作可以实现间接寻址

- 例：利用间接寻址对 100h～1FFh，300h~3FFh 通用数据存储器进行清零

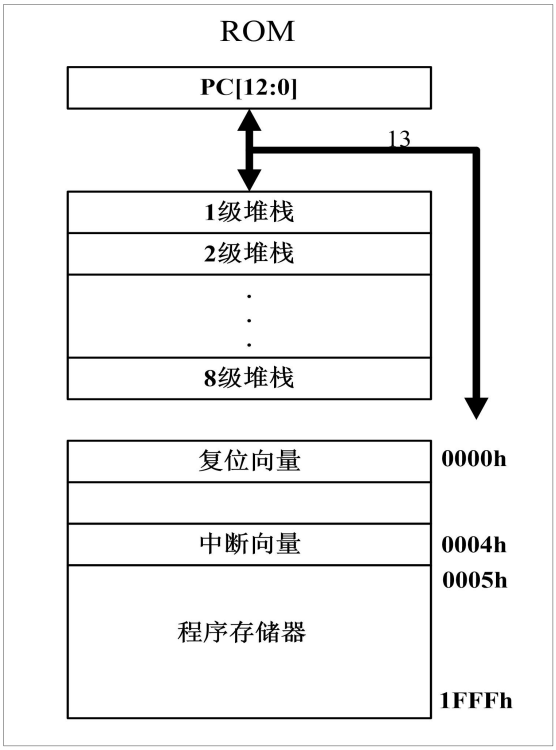
```
        MOVLW   00h         ;清零 0x100~0x1FF
        MOVWF   FSR0L
        MOVLW   0x01
        MOVWF   FSR0H       ; FSR 指向 100h 地址
NEXTBYTE:
        CLRF    INDF0       ;对 FSR 指向的数据存储器清零
        INCFSZ  FSR0L,F     ;FSR0L+1，指向下一个地址
                            ;注意这里的边界值为欲操作 RAM 最大地址+ 1
                            ;利用间接寻址，注意意外指向特殊寄存器的情况
        GOTO    NEXTBYTE    ;FSR0L 的值加一不溢出，循环清零下一个地址

        MOVLW   00h         ;清零 0x300~0x3FF
        MOVWF   FSR1L       ;FSR 指向 300h 地址
        MOVLW   0x03
        MOVWF   FSR1H
NEXTBYTE_1:
        CLRF    INDF1       ;对 FSR 指向的数据存储器清零
        INCFSZ  FSR1L,F     ;FSR1L，指向下一个地址
                            ;注意这里的边界值为欲操作 RAM 最大地址+1
                            ;利用间接寻址，注意意外指向特殊寄存器的情况
        GOTO    NEXTBYTE_1  ;FSR1L 的值加不溢出，循环清零下一个地址

CONTINUE:
        …                   ;完成清零操作
```

2.3 堆栈

HC18P23xL 具有一个 8 级深度的硬件堆栈。当执行 CALL 指令或由于中断导致程序跳转时，PC 值会被压入堆栈;当执行 RETURN、RETLW 或 RETFIE 指令时，PC 值从堆栈弹出。



注：

压栈级数请勿超过 8 级，超过 8 级压栈将导致堆栈溢出，溢出后堆栈指针循环，新的压栈将覆盖原堆栈内容。

3 复位

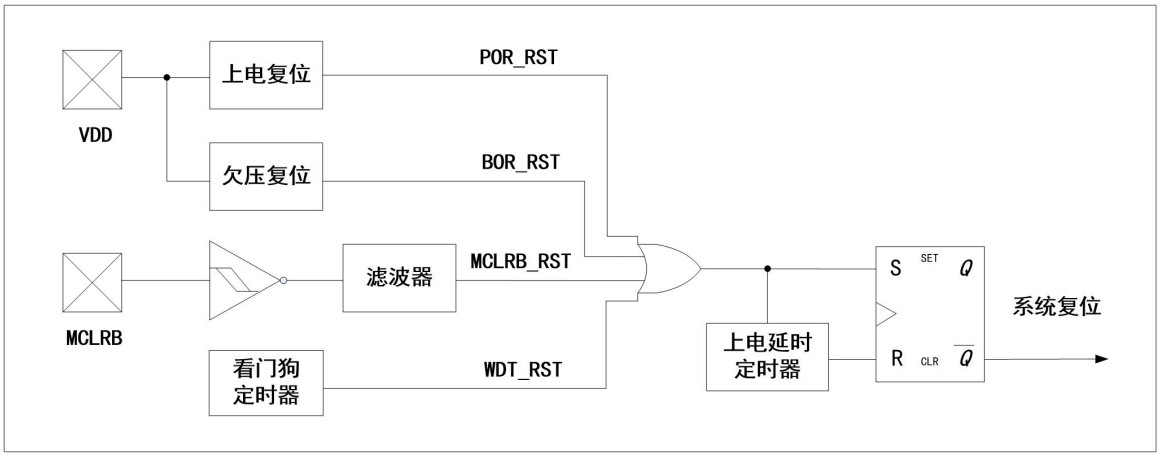
3.1 概述

HC18P23xL 共有四种复位方式：

- 上电复位（POR）
- 外部复位（MCLRB Reset）
- 欠压复位（BOR）
- 看门狗定时器复位（WDT Reset）

当上述任何一种复位产生时，系统进入复位状态，所有的特殊功能寄存器被初始化，程序停止运行，同时程序计数器（PC）清零。经过上电延时定时器延时后，系统结束复位状态，程序从 0000h 地址开始执行。STATUS 寄存器的 bit4（TO 位）及 PCON 寄存器的 bit0（BOR 位）、bit1（POR 位）显示系统复位状态信息，可通过 3 个标志位判断复位来源，从而控制系统的运行路径。

复位电路示意图



特殊功能寄存器复位状态

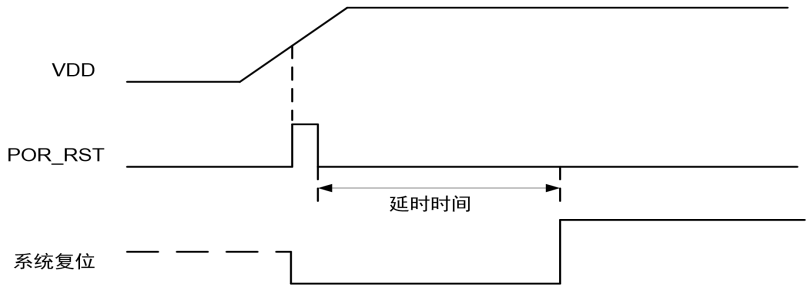
TO	POR	BOR	复位方式	说明
1	0	x	上电复位	电源上电。
u	u	0	欠压复位	电源电压低于BOR电压点。
u	u	u	外部复位	外部复位管脚低电平。
0	u	u	看门狗定时器复位	运行模式下，看门狗定时器溢出。

复位方式	STATUS寄存器	PCON寄存器
上电复位	0001 1xxx	00-1 qq00
正常工作模式下的外部复位	0001 1xxx	00-1 qq0u
休眠模式下的外部复位	0001 0uuu	00-1 qquu
欠压复位	0001 0uuu	00-1 qqu0
看门狗定时器复位	0000 1uuu	00-1 qquu

注： u = 不变， x = 未知， - = 未使用，q = 取值视条件而定

3.2 上电复位

系统上电过程中，VDD 达到系统正常工作电压之前，上电复位电路产生内部复位信号，可通过查询 PCON 寄存器来判断是否发生上电复位。VDD 最大上升时间 T_{VDD} 必须满足规格要求。任何一种复位方式都需要一定的响应时间，系统提供完善的复位流程以保证复位动作的顺利进行。对于不同类型的振荡器，完成复位所需要的时间也不同。因此，VDD 的上升速度和不同晶振的起振时间都不固定。RC 振荡器的起振时间最短，晶体振荡器的起振时间则较长。在用户的使用过程中，应考虑系统对上电复位时间的要求。



注：关于上电复位，请注意以下几点：

1、 VDD 上电必须从 0V 开始，若 VDD 有残留电压，POR_RST 信号无法稳定产生；

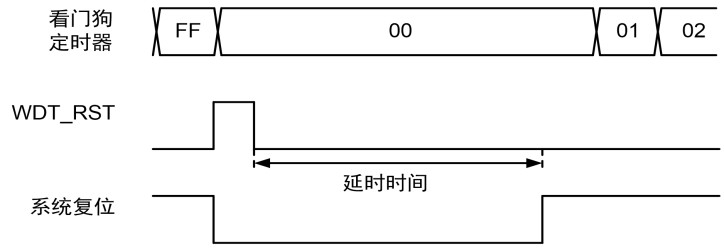
2、 VDD 上电斜率必须满足大于 500mV/ms，否则 POR_RST 信号可能无法产生；

3、 上电复位延时的复位延时时间在配置字里选择，两个档位（18ms/4.5ms）。

3.3 看门狗定时器复位

在高频和低频模式下，看门狗定时器溢出会产生WDT复位；在绿色和休眠模式下，看门狗定时器溢出将唤醒SLEEP并使其返回高频或低频模式，程序从SLEEP指令下一条开始执行。WDT定时器配置字和WDTENS都为1时，才能使能看门狗定时器。

看门狗复位示意图：



注：关于看门狗复位使用时，请注意以下几点：

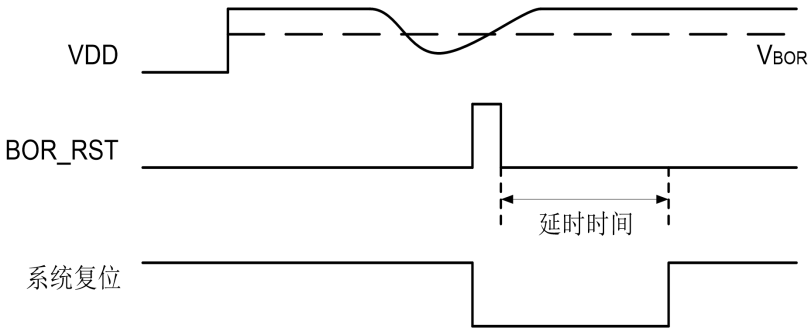
1、 看门狗的使能逻辑：看门狗使能 = 看门狗配置字使能 & 看门狗软件使能（WDTENS=1）；
2、 不能在中断程序中对看门狗进行清零，否则无法监控主程序跑飞情况；
3、 程序中应该只在主程序中有一次清看门狗的动作，这种架构能够最大限度的发挥看门狗的保护功能；
4、 看门狗复位的延时时间为 2.2ms/1.1ms；
5、 使用时注意：不论哪种方式复位后，看门狗软件使能位（WDTENS）的值为 1。

3.4 欠压复位

3.4.1 欠压复位的产生

当VDD电压下降到V_{BOR}以下，且持续时间满足，系统产生欠压复位。

欠压复位示意图：



低电压复位（BOR）是单片机内置的掉电复位保护装置，当VDD 跌落并低于BOR 检测电压值时，BOR被触发，系统复位。不同的单片机有不同的BOR 检测电平。因此采用BOR 依赖于系统要求和环境状况。如果电源跌落剧烈，远低于BOR 触发点，BOR 能够起到保护作用，让系统正常复位；如果电源电压跌落不是很剧烈，仅仅是接近BOR 触发点而造成的系统错误，则BOR 就不能起到保护作用让系统复位。

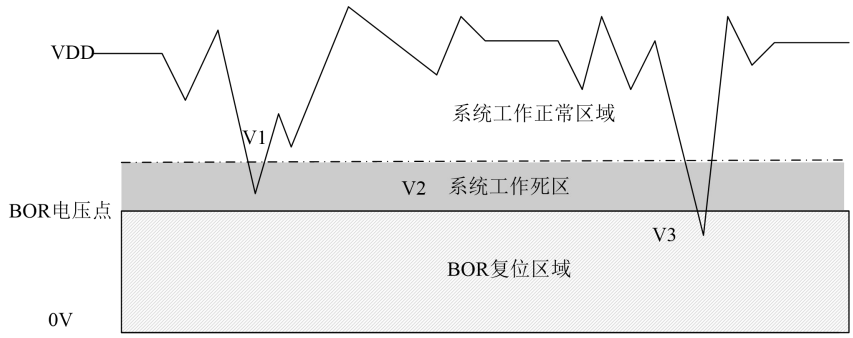
为避免电源较大的抖动，HC18P23xL采取必要的电源抖动处理电路或其他保护电路，防止电源抖动超过1.0V，导致芯片工作异常。

HC18P23xL通过配置字BOR编译选项控制选择低电压检测档位，请客户在使用时根据情况选择合适的BOR电压。

BOR档位：BOR3.6V/2.4V/2.0V

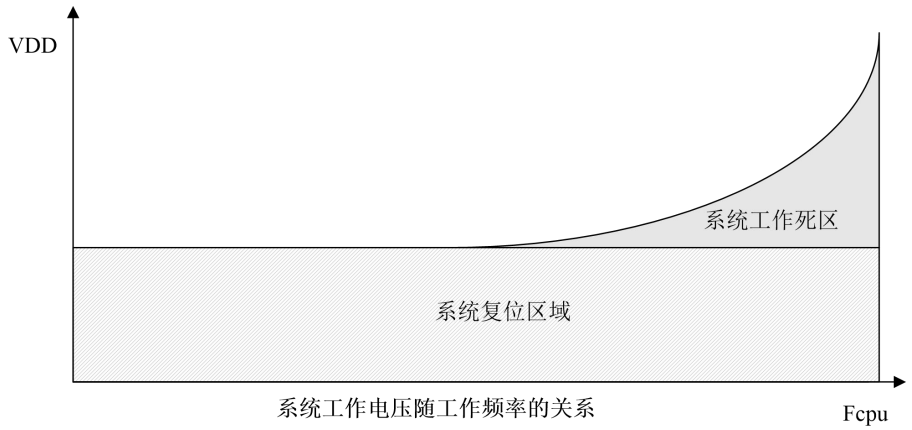
3.4.2 工作死区

电压跌落可能会进入系统死区。系统死区意味着电源不能满足系统的最小工作电压要求。下图是一个典型的掉电复位示意图。图中，VDD受到严重的干扰，电压值降得非常低。虚线以上区域系统正常工作，在虚线以下的区域内，系统进入未知的工作状态，这个区域称作死区。当VDD跌至V1 时，系统仍处于正常状态；当VDD跌至V2时，系统进入死区，系统工作在死区时，可能导致程序的运行紊乱;当电压跌至V3，且低于BOR电压点，系统可正常复位，处于BOR电压点的时间过短，系统仍无法正常产生欠压复位信号，可能导致程序的运行紊乱。



3.4.3 工作死区与工作频率的关系

工作死区电压与工作速度相关，如下图示意了死区与工作频率的关系：



系统工作电压随工作频率的关系

3.4.4 死区防护

对于死区防护，有以下几点建议：

- 合理使用看门狗复位电路
- 降低系统的工作频率
- 合理采用外部复位电路（电压偏移复位电路、外部 IC 复位）

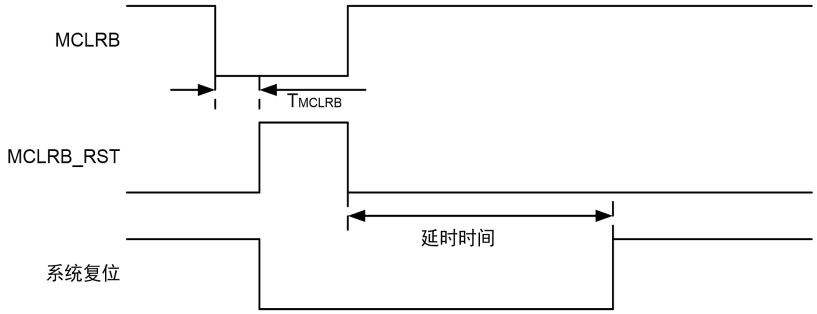
注：

二极管 RC 复位电路电压偏移复位电路、外部 IC 复位防止系统进入死区。

3.5 外部复位

当外部复位端口 MCLRB 输入一个持续时间超过 T_{MCLRB} 的低电平时，产生外部复位。MCLRB 选择配置字（编译选项）为 1，MCLRB 口为外部复位输入口。

外部复位示意图：



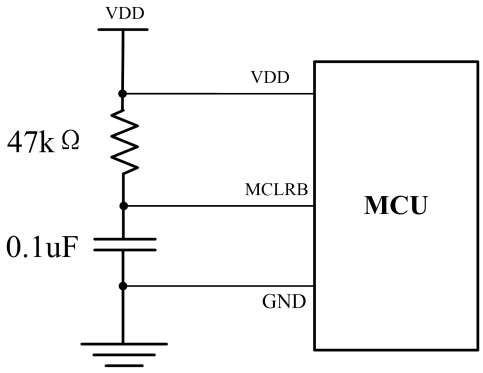
注：

TMCLRB 需大于 200μs（典型值）；外部复位延时时间为 2.2ms/1.1ms。

3.5.1 外部 RC 复位电路

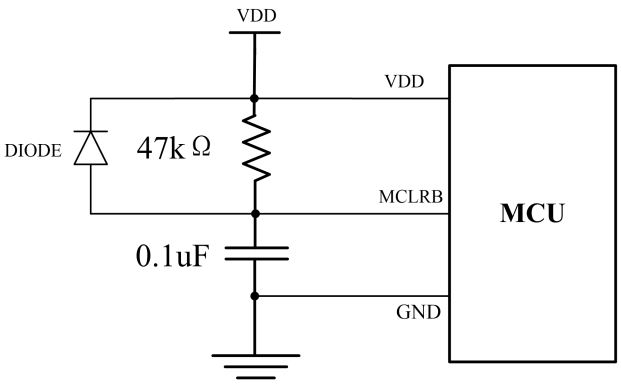
由电阻和电容组成的基本RC复位电路，它在系统上电的过程中能够为复位引脚提供一个缓慢上升的复位信号。这个复位信号的上升速度低于VDD的上电速度，为系统提供合理的复位时序，当复位引脚检测到高电平时，系统复位结束，进入正常工作状态。

如下图：



3.5.2 二极管 RC 复位电路

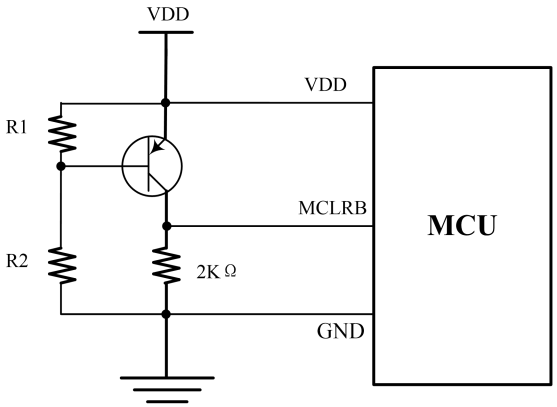
在基本RC复位电路上增加一个二极管（DIODE），对于电源异常情况，二极管正向导通使电容快速放电并与VDD保持一致，避免复位引脚持续高电平，系统无法正常复位。



3.5.3 电压偏置复位电路

电压偏置复位电路是一种简单的电压检测复位电路，调整电压检测点，可以解决系统死区问题。电路中，R1和R2构成分压电路，当R1和R2的分压值高于三极管的开启电压时，三极管集电极输出高电平，单片机正常工作；当R1和R2的分压值低于三极管的开启电压时，集电极输出低电平，MCU复位。

对于不同应用需求，选择适当的分压电阻。分压电阻R1和R2在电路中要耗电，此处的功耗必须计入整个系统的功耗中。



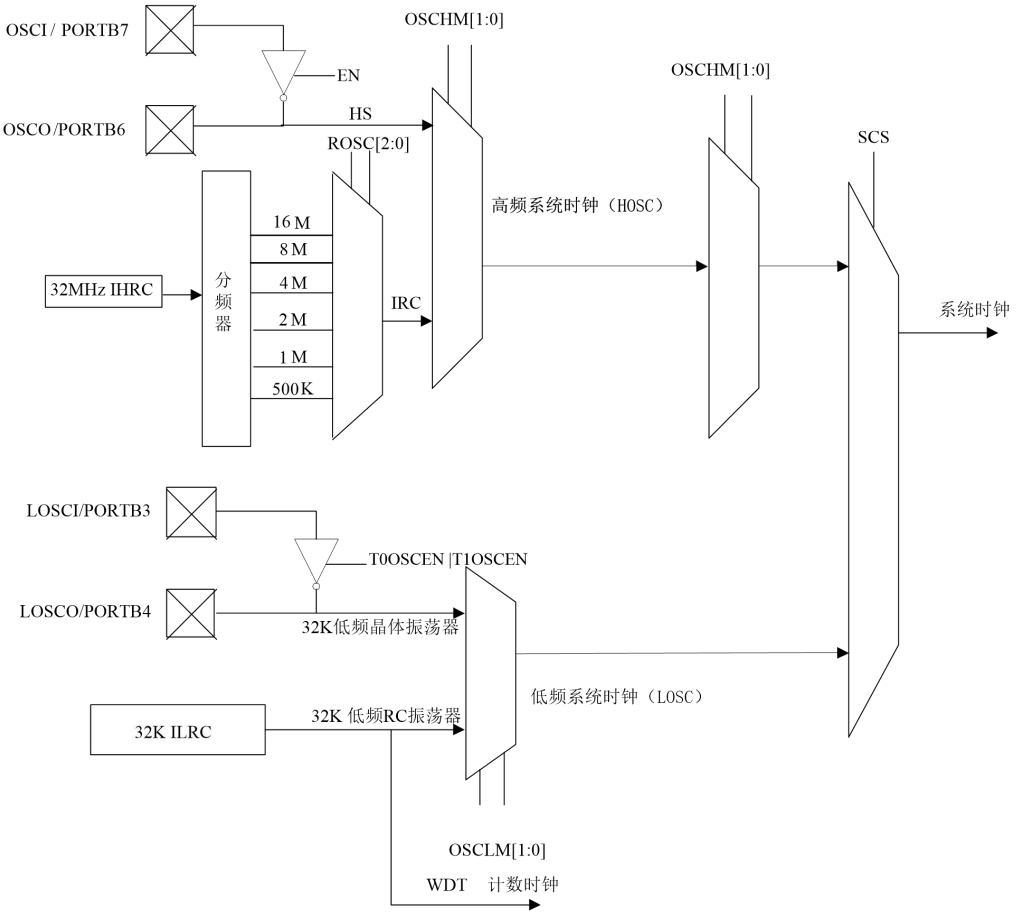
4 系统时钟

4.1 概述

HC18P23xL内带双时钟系统：高频时钟和低频时钟。高频时钟的时钟源由高频晶振或内部RC振荡电路（IRC 32MHz）提供。低频时钟的时钟源则由低频晶振或内部低速RC振荡电路（RC 32KHz@5V）提供。两种时钟都可作为系统时钟源Fosc。OSCCON寄存器的SCS位控制高频时钟和低频时钟之间切换。

- 高频模式：Fcpu = Fsys / N，N = 2或4，时钟模式选择决定N的值。
- 低频模式：Fcpu = Fsys / N，N = 2或4，时钟模式选择决定N的值。

4.2 时钟框图



- OSCHM[1:0]：高速系统时钟选择配置字
- OSCLM[1:0]：低速系统时钟选择配置字
- ROSC[2:0]：高速内部RC振荡器频率选择配置字
- Fosc：时钟源频率
- Fsys：系统时钟频率
- Fcpu：指令时钟频率

4.3 系统高频时钟

系统高频时钟有两种选择，通过OSCHM[1:0]高频系统时钟选择配置字来控制。

高频系统时钟选择配置字：

OSCHM[1:0]	说明
00	内部 RC 振荡器（IRC），OSCI/OSCO 作为普通 IO 口。
01	高频晶体振荡器（HS），OSCI/OSCO 作为高频晶体振荡器输入/输出口。

4.3.1 内部高频 RC 振荡器

配置字OSCHM[1:0]和ROSC[2:0]控制单片机的内置RC高速时钟。OSCHM[1:0]若选择“00”，则内置RC振荡器作为系统时钟源，OSCI/OSCO作为通用I/O口。

内置RC高频时钟有16M/8M/4M/2M/1M /500K六种选择。

高频内部RC振荡器频率选择配置字

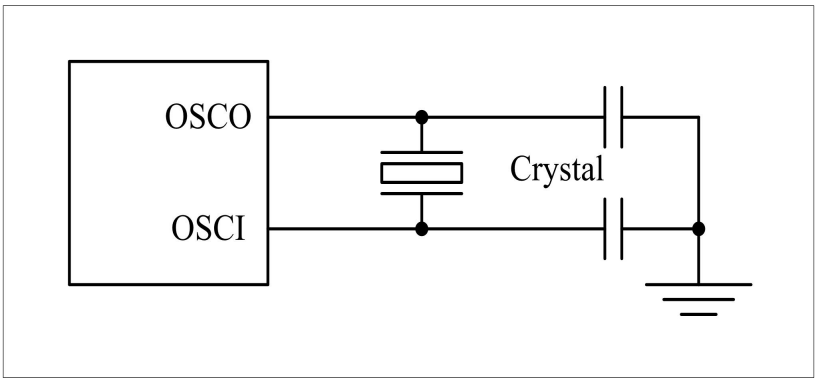
ROSC[2:0]	说明
110	内部RC振荡器频率选择16MHz
101	内部RC振荡器频率选择8MHz
100	内部RC振荡器频率选择4MHz
011	内部RC振荡器频率选择2MHz
010	内部RC振荡器频率选择1MHz
001	内部RC振荡器频率选择500KHz

4.3.2 外部高频时钟

外部高频时钟，由配置字 OSCHM 控制具体模式的选择

- 高频晶体振荡器： 最高 20MHz

高频晶体振荡器的频率为1MHz~20MHz，推荐的典型值为4MHz、8MHz和16MHz，电容推荐值为20pF。



注：

OSCI 和 OSCO 引脚与振荡器和起振电容之间距离 10mm 以内。

4.4 系统低频时钟

高频时钟有两种选择，通过低频时钟选择配置字来选择。

- 低频晶体振荡器： 32.768KHz
- 低频 RC 振荡器： 32KHz（5V 典型值）

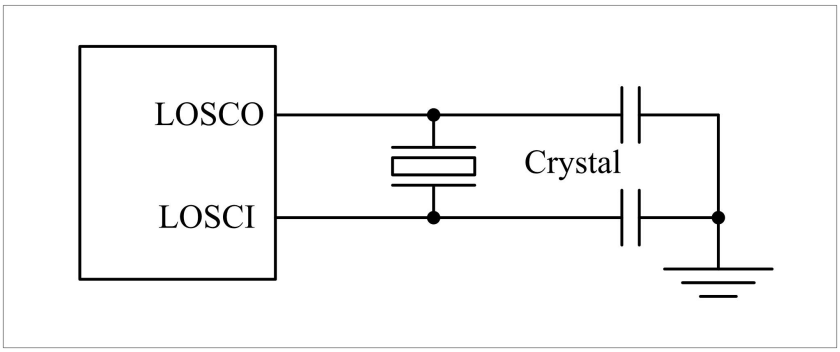
低频系统时钟选择配置字

OSCLM[1:0]	说明
00	低频 RC 振荡器，32KHz，LOSCI/LOSCO 作为输入/输出口
01	低频晶体振荡器，32.768KHz，LOSCI/LOSCO 作为低频晶体振荡器输入/输出口

4.4.1 低频晶体振荡器

低频晶体振荡器的频率为32.768KHz，电容推荐值为20pF。

低频晶体振荡器电路



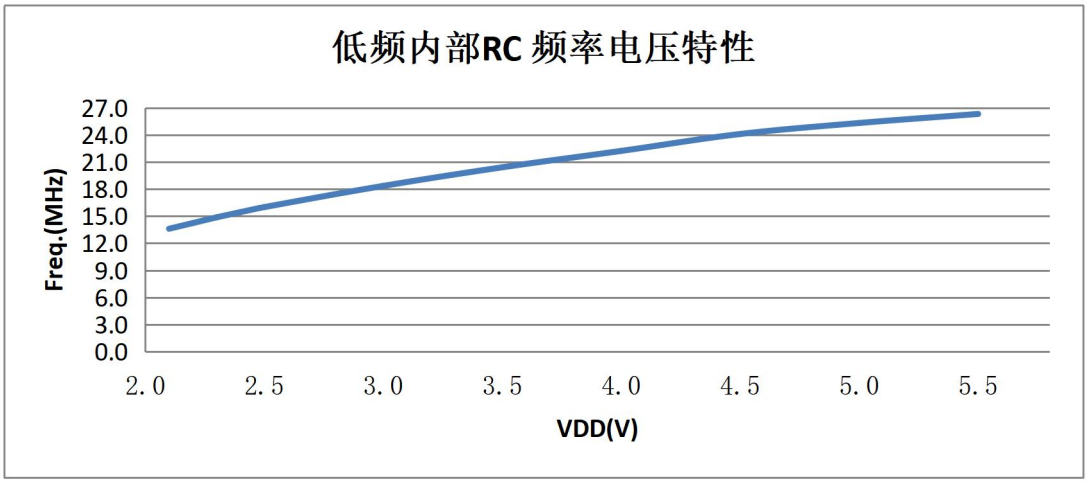
系统工作在绿色模式下，可以使能低频晶体振荡器。

注：
外部高频晶振接 OSCO、OSCI 端口，外部低频晶振接 LOSCO、LOSCI 端口。

4.4.2 低频 RC 振荡器

系统低频时钟源也可采用RC振荡电路。低频内部RC振荡电路的输出频率受系统电压和环境温度的影响较大，通常为5V时输出32KHz（典型值）。

输出频率与工作电压之间的关系如下图所示：



注：

低频内部 RC 时钟也用作看门狗定时器的时钟。

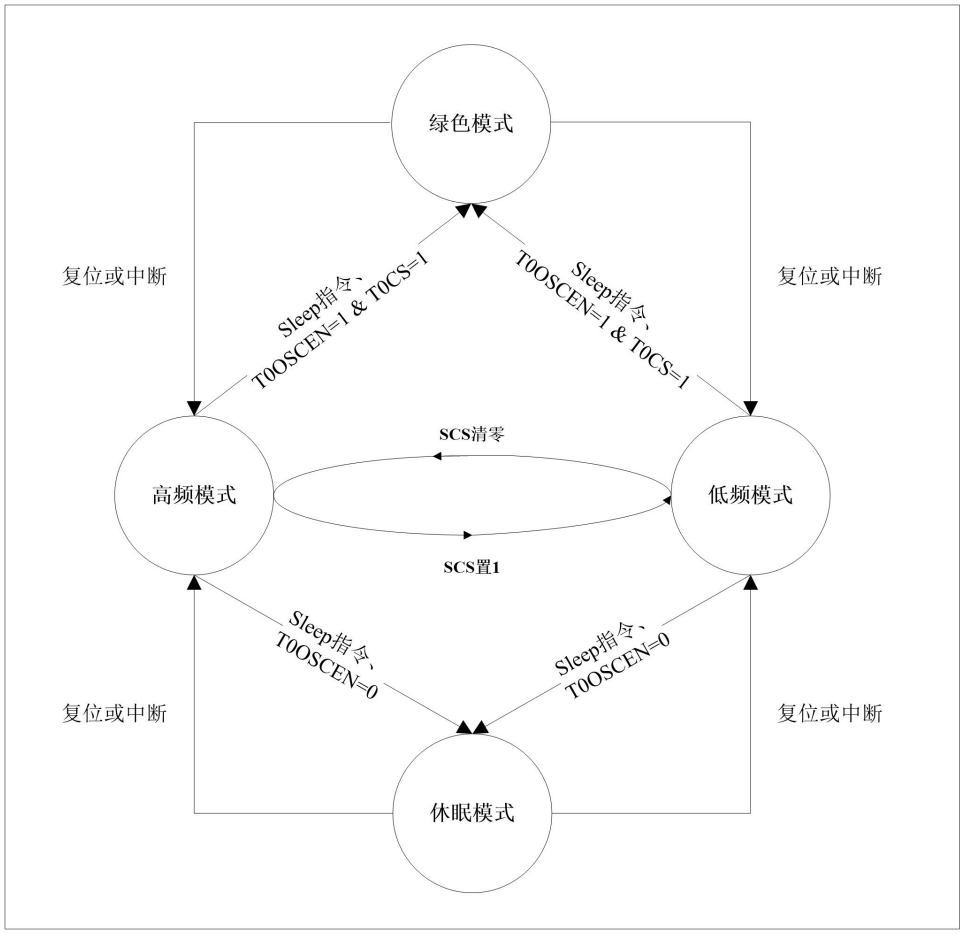
5 系统工作模式

5.1 概述

HC18P23xL可在以下四种工作模式之间进行切换：

- 高频模式
- 低频模式
- 休眠模式
- 绿色模式

系统复位后，工作于高频模式还是低频模式，由系统配置字决定。程序运行过程中，可以通过设置SCS 位使系统在高频和低频模式之间切换。



注：

1. 从休眠或绿色模式唤醒，中断使能的情况则进入相应中断，否则执行下一句；
2. 外部复位和 Timer2 中断不能唤醒休眠或绿色模式；
3. 进入休眠或绿色模式前，关闭 WDT 可降低功耗。

各种模式下振荡器模块及Timer0/Timer1的工作状态表

模块	高频模式	低频模式	绿色模式	休眠模式
高频振荡器	运行	由HXEN决定	由HXEN决定	关闭
低频振荡器	运行	运行	运行	关闭
Timer0	运行	运行	定时唤醒模式下运行	计数器模式下运行
Timer1	运行	运行	异步定时唤醒模式下运行	异步计数器模式下运行

5.2 模式切换举例

➢ 例：高频/低频模式切换到睡眠模式。

```
BCF     STATUS,RP0      ;Bank0
BCF     OSCCON,T0OSCEN
SLEEP
```

➢ 例：高频模式切换到低频模式.

```
BCF     STATUS,RP0      ;Bank0
BSF     OSCCON,SCS      ;SCS = 1，系统进入低频模式
```

➢ 例：从低频模式切换到高频模式。

```
BCF     STATUS,RP0      ;Bank0
BCF     OSCCON,SCS      ;SCS = 0，系统进入高频模式
```

➢ 例：从高频/低频模式切换到绿色模式

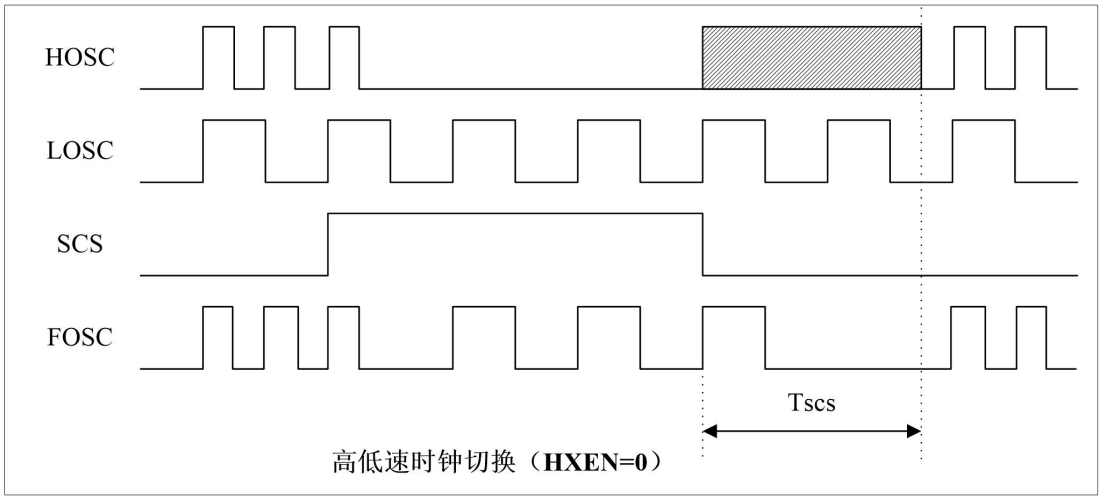
```
; T0定时器定时唤醒
BCF     STATUS,RP0      ;Bank0
MOVLW   0x05
MOVWF   OPTION
BSF     OPTION,T0CS
BSF     OSCCON, T0OSCEN
BSF     INTCON,T0IE     ;使能T0定时器。
BSF     INTCON,GIE
CLRF    T0
SLEEP
```

➢ 例：从高频/低频模式切换到绿色模式。

```
;T0定时器定时唤醒，OSCLM=01，低频晶体振荡器为32.768KHz，定时唤醒时间为0.5s。
BCF     STATUS,RP0      ;Bank0
MOVLW   0x05
MOVWF   OPTION
BSF     OPTION,T0CS
BSF     OSCCON,T0OSCEN
BCF     INTCON,T0IF
BSF     INTCON,T0IE     ;使能T0定时器
BSF     INTCON,GIE
CLRF    T0
```

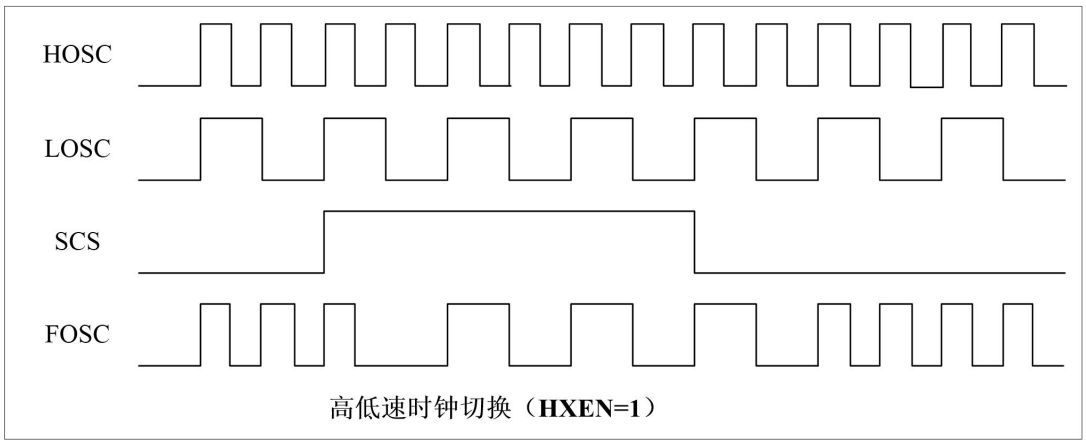
```
RTC_MODE:
        SLEEP
        BCF     STATUS,RP0      ;Bank0
        BCF     INTCON,T0IF     ;0.5s时间到
        …
        GOTO    RTC_MODE
```

5.3 高低频模式切换



高低速时钟切换（**HXEN=0**）

高低频切换时序：



高低速时钟切换（**HXEN=1**）

时钟切换时间（Tscs）计算：

Tscs = 高频振荡器起振时间 + 高频振荡器稳定时间

不同类型高频振荡器的稳定时间表

振荡器类型	高频振荡器稳定时间
高频晶体振荡器	1024 Clock
外部/内部 RC 振荡器	16 Clock

5.4 唤醒时间

系统进入休眠模式后，系统时钟停止运行。外部中断把系统从休眠模式下唤醒时，系统需要等待振荡器起振定时器（OST）定时结束，以使振荡电路进入稳定工作状态，等待的这一段时间称为唤醒时间。

唤醒时间结束后，系统进入高频或低频模式。

唤醒时间的计算如下：

唤醒时间 = 起振时间 + OST定时时间

同类型振荡器OST定时时间表

振荡器类型	OST 定时时间
高/低频晶体振荡器	1024 Clock
外/内部 RC 振荡器	16 Clock
低频 RC 振荡器	4 Clock

注：系统进入绿色模式后，低频时钟正常运行。外部或内部中断将系统从绿色模式中唤醒不需要唤醒时间。

5.5 OSCCON 寄存器

07Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCCON	T0OSCEN	-	-	-	-	-	HXEN	SCS
R/W	R/W	-	-	-	-	-	R/W	R/W
POR的值	0	-	-	-	-	-	0	q

注： x = 未知， — = 未实现， q = 取值视条件而定

bit [7] **T0OSCEN**：低频振荡器使能位

1 = 在低速或绿色模式下使能内部32K WDT振荡器

0 = 在低速或绿色模式下禁止内部32K WDT振荡器

bit [1] **HXEN**：高频振荡器使能位

1 = 在低速或绿色模式下使能高频振荡器

0 = 在低速或绿色模式下禁止高频振荡器

bit [0] **SCS**：高低频模式选择位

1 = 系统时钟选择为低频系统时钟

0 = 系统时钟选择为高频系统时钟

6 中断

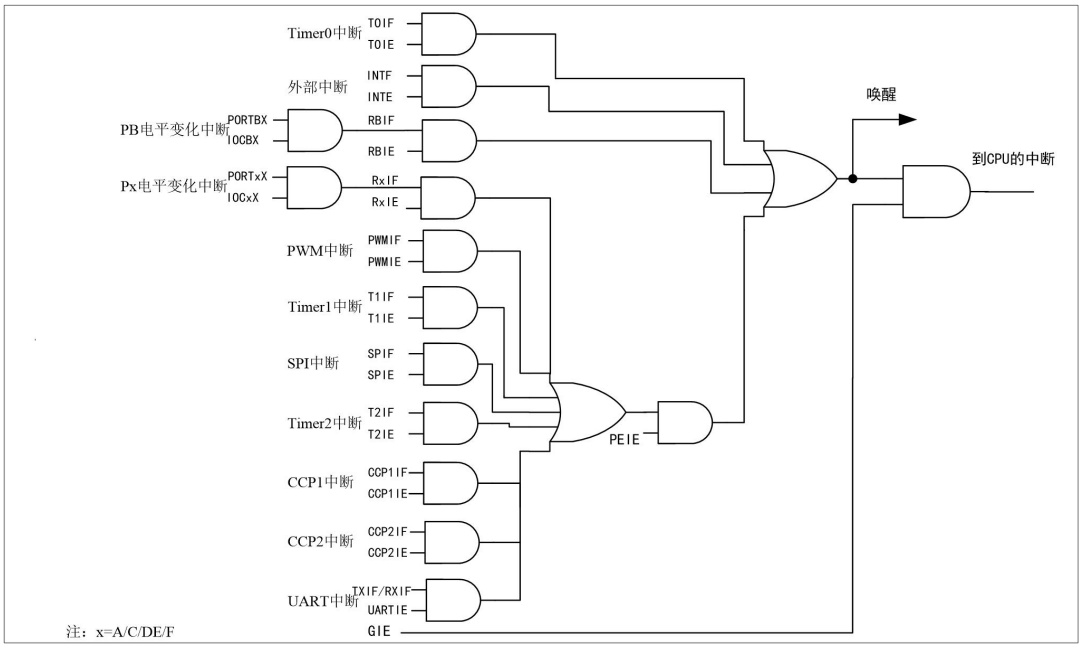
HC18P23xL中断源：

- Timer0定时器中断
- INT0外部中断
- PORT口电平变化中断
- Timer1定时器中断
- Timer2定时器中断
- CCP1中断
- CCP2中断
- AD中断
- UART中断
- SPI中断
- PWM中断

系统产生中断时，程序计数器（PC）值压入堆栈，程序跳转至0004h，进入中断服务程序。当程序运行到RETFIE指令时，系统退出中断服务程序，程序计数器值出栈，系统执行PC+1地址对应的指令。

为避免误进入中断，在使能中断和退出中断服务程序之前，必须清除相应中断标志位。在中断服务程序中不需要软件使能GIE，以免造成程序紊乱。

中断示意图



6.1 内核中断

使能内核中断必须将GIE和相应中断的使能位置1，使能PORTB电平变化中断还需要将相应端口配置为输入并且IOCB的相应位置1。INT0外部中断和PORTB电平变化中断可以唤醒SLEEP，Timer0中断在计数器模式和定时唤醒模式下可以唤醒SLEEP。

09h、209h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	x

注： x = 未知

bit [7] **GIE**：全局中断使能位

1 = 使能所有未屏蔽的中断

0 = 禁止所有中断

bit [5] **T0IE**：Timer0溢出中断使能位

1 = 使能Timer0中断

0 = 禁止Timer0中断

bit [4] **INTE**： INT0外部中断使能位

1 = 使能INT0外部中断

0 = 禁止INT0外部中断

bit [3] **RBIE**：PORTB电平变化中断使能位

1 = 使能PORTB电平变化中断

0 = 禁止PORTB电平变化中断

bit [2] **T0IF**：Timer0溢出中断标志位，Timer0计数寄存器在FFh至00h时产生溢出信号

1 = Timer0计数寄存器溢出（必须由软件清零）

0 = Timer0计数寄存器未溢出

bit [1] **INTF**：INT0外部中断标志位

1 = 发生INT0外部中断（必须由软件清零）

0 = 未发生INT0外部中断

bit [0] **RBIF**：PORTB电平变化中断标志位

1 = PORTB[7:0]中至少有一个口的电平状态发生了改变（必须由软件清零）

0 = PORTB[7:0]电平状态没有变化

078h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	-	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	1	1	1	1	1	1	1

bit [6] **INTEDG**：触发INT0外部中断的边沿选择位

1 = INT0引脚上升沿触发中断

0 = INT0引脚下降沿触发中断

6.2 外设中断

使能外设中断必须将GIE和PEIE置1，同时将相应中断的使能位置1。Timer1门控事件中断可以唤醒SLEEP，Timer1中断在异步计数器模式和异步定时唤醒模式下可以唤醒SLEEP，CCPx中断在捕捉模式下可以唤醒SLEEP。

09h、209h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	x

bit [7] **GIE**：全局中断使能位
1 = 使能所有未屏蔽的中断
0 = 禁止所有中断

bit [6] **PEIE**：外设中断使能位
1 = 使能所有未屏蔽的外设中断
0 = 禁止所有外设中断

070h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE1	RBPUB	ADIE	-	-	-	CCP1IE	T2IE	T1IE
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
POR的值	1	0	-	-	-	0	0	0

bit [6] **ADIE**：ADC中断使能位
1 = 使能ADC中断
0 = 禁止ADC中断

bit [2] **CCP1IE**：CCP1中断使能位
1 = 使能CCP1中断
0 = 禁止CCP1中断

bit [1] **T2IE**：Timer2计数寄存器与PR2匹配中断使能位
1 = 使能Timer2匹配中断
0 = 禁止Timer2匹配中断

bit [0] **T1IE**：Timer1溢出中断使能位
1 = 使能Timer1溢出中断
0 = 禁止Timer1溢出中断

071h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE2	PWM2IE	PWM1IE	PWM0IE	-	-	UARTIE	SPIE	CCP2IE
R/W	R/W	R/W	R/W	-	-	R/W	R/W	R/W
POR的值	0	0	0	-	-	0	0	0

bit [7:5] **PWMxIE**：PWM中断使能位
1 = 使能PWM中断
0 = 禁止PWM中断

bit [2] **UARTIE**：UART中断使能位
1 = 使能UART中断
0 = 禁止UART中断

bit [1] **SPIE**： SPI中断使能位

1 = 使能SPI中断

0 = 禁止SPI中断

bit [0] **CCP2IE**：CCP2中断使能位

1 = 使能CCP2中断

0 = 禁止CCP2中断

072h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE3	-	-	-	RFIE	REIE	RDIE	RCIE	RAIE
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
POR的值	-	-	-	0	0	0	0	0

bit[4:0] **RxIE**： PORTx电平变化中断使能位（x=A/C/D/E/F）

1 = 使能PORTx电平变化中断

0 = 禁止PORTx电平变化中断

054h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR1	-	ADIF	-	-	-	CCP1IF	T2IF	T1IF
R/W	-	R/W	-	-	-	R/W	R/W	R/W
POR的值	-	0	-	-	-	0	0	0

bit [6] **ADIF**：AD中断标志位

1 =ADC转换已完成（必须由软件清零）

0 = ADC转换未完成或尚未开始

bit [2] **CCP1IF**：CCP1中断标志位

捕捉模式：

1 = 发生了捕捉事件（必须用软件清零）

0 = 未发生捕捉事件

比较模式：

1 = 发生了比较事件（必须用软件清零）

0 = 未发生比较事件

PWM 模式：

在此模式下未使用

bit [1] **T2IF**：Timer2计数寄存器与PR2匹配中断标志位

1 = Timer2发生匹配（必须用软件清零）

0 = Timer2未发生匹配

bit[0] **T1IF**：Timer1溢出中断标志位，Timer1计数寄存器在FFFFh至0000h时产生溢出信号

1 = Timer1计数寄存器溢出（必须由软件清零）

0 = Timer1计数寄存器未溢出

055h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR2	PWM2IF	PWM1IF	PWM0IF	-	RXIF	TXIF	SPIF	CCP2IF
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
POR的值	0	0	0	-	0	0	0	0

bit [7:5] **PWMxIF**：PWM中断标志位

1 = PWM中断产生中断（必须由软件清零）

0 = PWM中断未产生中断

bit [3] **RXIF**：UART接收标志位

1 = UART接收中断产生中断（必须由软件清零）

0 = UART接收中断未产生中断

bit [2] **TXIF**：UART发送中断标志位

1 = UART发送中断产生中断（必须由软件清零）

0 = UART发送中断未产生中断

bit [1] **SPIF**：SPI中断标志位

1 = SPI中断产生中断（必须由软件清零）

0 = SPI中断未产生中断

bit [0] **CCP2IF**：CCP2中断标志位

捕捉模式：

1 = 发生了捕捉事件（必须用软件清零）

0 = 未发生捕捉事件

比较模式：

1 = 发生了比较事件（必须用软件清零）

0 = 未发生比较事件

PWM 模式：

在此模式下未使用

056h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR3	-	-	-	RFIF	REIF	RDIF	RCIF	RAIF
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
POR的值	-	-	-	0	0	0	0	0

bit [4:0] **RxIF**：PORTx电平变化中断标志位(x=A/C/D/E/F)

1 = PORTx[7:0]中至少有一个口的电平状态发生了改变（必须由软件清零）

0 = PORTx[7:0]电平状态没有变化

6.3 GIE 全局中断

只有当全局中断控制位GIE置“1”的时候程序才能响应中断请求。一旦有中断发生，程序计数器入栈，程序转至中断向量地址（ORG 0004H），堆栈层数加1。

例：设置全局中断控制位（GIE）

```
BSF     INTCON,GIE      ;使能 GIE。
```

注：在所有中断中，GIE 处于关闭状态。

6.4 中断保护

有中断请求发生并被响应后，程序转至0004H执行中断服务程序。

中断服务程序开始执行时，需保存W寄存器、STATUS寄存器、PCLATH寄存器的内容;结束中断服务程序时，恢复PCLATH寄存器、STATUS寄存器、W寄存器的数值，注意顺序。

注：

1.为了使保存系统寄存器的 RAM，可进行特殊操作，保证保存相应的系统寄存器；

2.在退出中断时，由于需要先恢复 STATUS，再使用 MOVF 指令恢复 W，可能会改变 STATUS，因此必须使用 SWAPF 指令恢复 W。注意在中断中共有两句 SWAPF 指令。

➢ 例：对W、PCLATH 和STATUS 进行入栈保护。

```
                ORG     0000H
                GOTO    START
                ORG     0004H
                GOTO    INT_SERVICE
                ORG     0010H
START:
                …
INT_SERVICE:
                MOVWF   W_TEMP              ;保存 W
                SWAPF   STATUS,W
                MOVWF   STATUS_TEMP         ;保存 STATUS
                MOVF    PCLATH, W
                MOVWF   PCLATH_TEMP         ;保存 PCLATH
                CLRF    STATUS              ;切换到BANK0
                …
                MOVF    PCLATH_TEMP ,W
                MOVWF   PCLATH              ;恢复PCLATH
                SWAPF   STATUS_TEMP,W
                MOVWF   STATUS              ;恢复STATUS
                SWAPF   W_TEMP,F
                SWAPF   W_TEMP,W            ;恢复W
                RETFIE                      ;退出中断
                …
                END
```

6.5 Timer0 定时器中断

T0 溢出时，无论 T0IE 处于何种状态，T0IF 都会置 1。若 T0IE 和 T0IF 都置 1，**且 GIE 使能**，系统就会响应 TIMER0 的中断；若 T0IE = 0，则无论 T0IF 是否置 1，系统都不会响应 TIMER0 中断。

➢ 例：T0 中断请求设置。

```
                MOVLW   0xC5
                BCF     STATUS,RP0          ;Bank0
```

```
                MOVWF   OPTION          ;T0时钟＝Fcpu / 64
                MOVLW   0x40            ;T0初始值＝64
                BCF     STATUS,RP0
                MOVWF   T0
                BSF     INTCON,T0IE     ;置T0中断使能标志
                BCF     INTCON,T0IF     ;清T0中断标志
                BSF     INTCON,GIE      ;使能GIE
```

➢ 例：T0 中断服务程序

```
                ORG     0004H
                GOTO    INT_SERVICE
INT_SERVICE:
                MOVWF   W_TEMP          ;保存W
                SWAPF   STATUS, W
                MOVWF   STATUS_TEMP     ;保存STATUS
                MOVF    PCLATH, W
                MOVWF   PCLATH_TEMP     ;保存PCLATH
                BCF     STATUS,RP0      ;BANK0
                BTFSS   INTCON,T0IF     ;检查是否有T0中断请求标志
                GOTO    EXIT_INT        ;T0IF = 0，退出中断
T0ISR:
                BCF     INTCON,T0IF     ;清T0IF
                MOVLW   0x40
                MOVWF   T0              ;重置T0值
                …                       ;T0中断程序

EXIT_INT:
                MOVF    PCLATH_TEMP ,W
                MOVWF   PCLATH          ;恢复PCLATH
                SWAPF   STATUS_TEMP,W
                MOVWF   STATUS          ;恢复STATUS
                SWAPF   W_TEMP, F
                SWAPF   W_TEMP,W        ;恢复W
                RETFIE                  ;退出中断
```

6.6 INT0 外部中断

INT0 被触发，则无论INTE 处于何种状态，INTF 都会被置“1”。如果INTF＝1 且INTE＝1，**GIE使能**，系统响应该中断;如果INTF＝1 而INTE＝0，系统并不会执行中断服务。在处理多中断时尤其需要注意。

078h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	RBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

bit [6] **INTEDG**： 触发INT0外部中断的边沿选择位

1 = INT0引脚上升沿触发中断

0 = INT0 引脚下降沿触发中断

➢ 例：INT0 中断请求设置，电平触发。

```
        BCF     STATUS,RP0      ;Bank0
        BSF     OPTION,INTEG    ;INT0 置为上升沿触发
        BCF     INTCON,INTF     ;INT0 中断请求标志清零
        BSF     INTCON,INTE     ;使能INT0 中断
        BSF     INTCON,GIE      ;使能GIE
```

➢ 例：INT0 中断。

```
        ORG     0004H ;
        GOTO    INT_SERVICE
INT_SERVICE:
        …                       ;保存STATUS、W 和PCLATH
        BTFSS   INTCON,INTF     ;检测INTF
        GOTO    EXIT_INT        ;INTF= 0，退出中断
        BCF     INTCON,INTF     ; INTF清零
        …                       ;INT0 中断服务程序
        …
EXIT_INT:
        …                       ;恢复STATUS、W和PCLATH
        RETFIE                  ;退出中断
```

6.7 PORT 电平变化中断

PORTx电平变化中断时，则无论RxIE处于何种状态，RxIF都会被置“1”。如果RxIF＝1 且RxIE＝1，**GIE使能**，系统响应该中断;如果RxIF＝1 而RxIE＝0，系统并不会执行中断服务。

电平变化中断必须将PORTx端口设为输入，并将寄存器IOCx对应位置“1”。

040h~045h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOCx	IOCx7	IOCx6	IOCx5	IOCx4	IOCx3	IOCx2	IOCx1	IOCx0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

IOCx[7:0]: PORTx[7:0]电平变化中断使能控制位。

0 = 该端口禁止电平变化中断;

1 = 该端口使能电平变化中断。

x = A、B、C、D、E、F

注：1.如要允许 PORTB 口电平变化中断必须将 IOCB 的对应端口的位置 1；

2.PORTB 电平变化中断中，在清零 RBIF 之前必须执行 PORTB 端口读操作；

3.PORTA/C/D/E/F 需使能 PEIE。

➢ 例：PORTB1 电平变化中断请求设置。

```
        MOVLW   0x02
        BCF     STATUS,RP0          ;BANK0
        IORWF   TRISB,F             ;PORTB1 端口为输入
        MOVLW   0x02
        IORWF   IOCB,F              ;使能PORTB1端口为电平变化中断
        MOVF    PORTB, W            ;读PORTB 口
        BCF     INTCON,RBIF         ; PROTB 中断请求标志清零
        BSF     INTCON,RBIE         ;使能PROTB 中断
        BSF     INTCON,GIE          ;使能GIE
```

➢ 例：PORTB 中断。

```
        ORG     0004H
        GOTO    INT_SERVICE
INT_SERVICE:
        …                           ;保存STATUS、W 和PCLATH
        BCF     STATUS,RP0          ;BANK0
        BTFSS   INTCON,RBIF         ;检测RBIF
        GOTO    EXIT_INT            ;RBIF = 0，退出中断
        MOVF    PORTB,W             ;读PORTB 端口
        BCF     INTCON,RBIF         ; RBIF 清零
        …                           ; PORTB 电平变化中断服务程序
        …
EXIT_INT:
        …                           ;恢复STATUS、W 和PCLATH
        RETFIE                      ;退出中断
```

➢ 例：PORTB 中断唤醒SLEEP。

```
        BCF     STATUS,RP0          ;BANK0
        MOVLW   0x02
        IORWF   TRISB,F             ;PORTB1 端口为输入
        MOVLW   0x02
        IORWF   IOCB,F              ;使能PORTB1端口为电平变化中断
        MOVF    PORTB,W             ;读PORTB口
        BCF     INTCON,RBIF         ;PROTB中断请求标志清零
        BSF     INTCON,RBIE         ;使能PROTB中断
        SLEEP
        BCF     INTCON,RBIE
        MOVF    PORTB,W             ;读PORTB端口
        …                           ;其他程序
```

注：1.如要允许 PORTx 口电平变化中断必须将 IOCx 的对应端口的位置 1；
2.PORTx 电平变化中断中，在清零 RxIF 之前必须执行 PORTx 端口读操作。

6.8 Timer1 中断

T1溢出时，无论T1IE 处于何种状态，T1IF都会置“1”。若T1IE 和T1IF 都置“1”，**且PEIE、GIE均使能**，系统就会响应TIMER1的中断;若T1IE = 0，则无论T1IF 是否置“1”，系统都不会响应TIMER1中断。

➢ 例：TIMER1工作于异步计数模式，并中断唤醒SLEEP

```
                MOVLW   0xA4
                BCF     STATUS,RP0          ;BANK0
                MOVWF   T1CON               ;T1时钟源为T1CKI;分频比为1:4;异步计数器模式
                MOVLW   nnH
                MOVWF   T1H
                MOVLW   nnH
                MOVWF   T1L                 ;Timer1赋初值
                MOVLW   0xC0
                MOVWF   INTCON              ;使能外设中断
                BSF     PIE1,T1IE
                BCF     STATUS,RP0
                BCF     PIR1,T1IF
                BSF     T1CON,T1ON
                BCF     OSCCON,T0OSCEN      ;禁止低频晶体振荡器
                SLEEP                       ;进入SLEEP

T1INT_SERVICE:
                …                           ;保存STATUS、W 和PCLATH
                BCF     STATUS,RP0          ; BANK0
                BTFSS   PIR1,T1IF           ;检测T1IF
                GOTO    EXIT_INT            ;T1IF = 0，退出中断
                BCF     PIR1,T1IF           ;T1IF 清零。
                …                           ;TIMER1 中断服务程序
                …
EXIT_INT:
                …                           ;恢复STATUS、W和PCLATH
                RETFIE                      ;退出中断
```

6.9 Timer2 定时器中断

当T2的值和PR2的值相同时，TIMER2中断被触发，则无论T2IE 处于何种状态，T2IF 都会被置“1”。如果T2IF＝1 且T2IE＝1，**且PEIE、GIE均使能**，系统响应该中断;如果T2IF＝1 而T2IE＝0，系统并不会执行中断服务。

➢ 例：TIMER2 中断请求设置

```
                BCF     STATUS,RP0          ;BANK0
                MOVLW   0xFF
```

```
                MOVWF   PR2             ;设置T2 周期
                MOVLW   0x04
                MOVWF   T2CON           ;设置分频比
                CLRF    T2
                BSF     PIE1,T2IE       ;使能TIMER2 中断
                BSF     INTCON,GIE
                BSF     T2CON,T2ON      ;使能TIMER2
```

➢ 例：TIMER2 中断。

```
                ORG     0004H
                GOTO    T2INT_SERVICE
T2INT_SERVICE:
                …                       ;保存STATUS、W 和PCLATH。
                BCF     STATUS,RP0      ;BANK0
                BTFSS   PIR1,T2IF       ;检测T2IF
                GOTO    EXIT_INT        ;T2IF = 0，退出中断
                BCF     PIR1,T2IF       ;T2IF 清零
                …                       ;TIMER2 中断服务程序
                …
EXIT_INT:
                …                       ;恢复STATUS、W和PCLATH
                RETFIE                  ;退出中断
```

6.10 AD 中断

当 ADC 完成，ADON 被硬件清零，无论 ADIE 处于何种状态，与此同时 ADIF 被置“1”。若 ADIE、ADIF 为“1”，**且 PEIE、GIE 均使能**，系统就会相应 ADC 中断；若 ADIE = 0，则无论 ADIF 是否置“1”，系统都不会响应 ADC 中断。

6.11 CCP 中断

当发生 CCPx 中断时，无论 CCPxIE 处于何种状态，CCPxIF 被置“1”。若 CCPxIE、CCPxIF 为“1”，**且 PEIE、GIE 均使能**，系统就会相应 CCPx 中断；若 CCPxIE = 0，则无论 CCPxIF 是否置“1”，系统都不会响应 CCPx 中断。

6.12 UART 中断

当 UART 接收、发送完成后，无论 UARTIE 处于何种状态，TXIF、RXIF 都被置“1”。若 UARTIE、TXIF、RXIF 为“1”，**且 PEIE、GIE 均使能**，系统就会相应产生 UART 中断。

6.13 SPI 中断

当 SPI 接收、发送完成后，无论 SPIE 处于何种状态，SPIF 都被置“1”。若 SPIE、SPIF 为“1”，**且 PEIE、GIE 均使能**，系统就会相应产生 SPI 中断。

6.14 PWM 中断

当 PWMx 周期计数器溢出，无论 PWMxIE 处于何种状态，PWMxIF 都被置“1”。若 PWMxIE 为“1”，且 **PEIE、GIE 均使能**，系统就会相应产生 PWM 中断。

6.15 多中断操作

在同一时刻，系统中可能出现多个中断请求。此时，用户必须根据系统的要求对各中断进行优先权的设置。中断请求标志IF由中断事件触发，当IF处于有效值“1”时，系统并不一定会响应该中断。各中断触发事件如下表所示：

中断	有效触发
T0IF	T0溢出
INTF	由INTEDG控制
RBIF	PORTB电平变化
RxIF	其他PORT口电平变化中断
T2IF	T2的值和PR2相同
TXIF/RXIF	UART发生发送接收事件
SPIF	SPI发生发送接收事件
PWMxIF	PWMx周期计数溢出中断

多个中断同时发生时，需要注意的是：首先，必须预先设定好各中断的优先权；其次，利用IE和IF控制系统是否响应该中断。在程序中，必须对中断控制位和中断请求标志进行检测。

➢ 例：多中断条件下检测中断请求。

```
                ORG     0004H
                GOTO    INT_SERVICE
INT_SERVICE:
                …                               ;保存STATUS、W和PCLATH
INT0CHK:                                        ;检查是否有INT0中断请求
                BCF     STATUS,RP0              ;BANK0
                BTFSS   INTCON,INTE             ;检查是否使能INT0中断
                GOTO    INT0CHK                 ;跳到下一个中断
                BTFSC   INTCON,INTF             ;检查是否有INT0中断请求
                GOTO    INT0                    ;进入INT0  中断
INTT0CHK:                                       ;检查是否有T0  中断请求
                BTFSS   INTCON,T0IE             ;检查是否使能T0中断
                GOTO    INTT2CHK                ;跳到下一个中断
                BTFSC   INTCON,T0IF             ;检查是否有T0中断请求
                GOTO    INTT0                   ;进入T0  中断
```

```
INTT2CHK:                                   ;检查是否有T2中断请求
            BCF     STATUS,RP0              ;BANK0
            BTFSS   PIE1,T2IE               ;检查是否使能T2中断
            GOTO    INTRBCHK                ;跳到下一个中断。
            BCF     STATUS,RP0              ;BANK0
            BTFSC   PIR1,T2IF               ;检查是否有T2中断请求
            GOTO    INTT2                   ;进入T2中断
INTRBCHK:
            BTFSS   INTCON,RBIE             ;检查是否使能PB电平变化中断
            GOTO    INT_EXIT                ;跳到中断结束
            BTFSC   INTCON,RBIF             ;检查是否有PB电平变化中断请求
            GOTO    INTRB                   ;进入PB电平变化中断
INTT2:
            BCF     PIR1,T2IF
            …                               ;T2中断处理程序
            GOTO    INT_EXIT

INT_EXIT:
            …                               ;恢复STATUS、W和PCLATH
            RETFIE                          ;退出中断
```

7 I/O口

HC18P23xL共有六组双向端口：

- PORTA口
- PORTB口
- PORTC口
- PORTD口
- PORTE口
- PORTF口

7.1 I/O 口模式

010h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

011h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

012h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

013h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

014h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

015h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

TRISx[7:0]：PORTx[7:0]的输入输出控制位

1 =输入状态

0 =输出状态

注：1. PORTB5 可设置为开漏输出。

08Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELL	ANSEL7	ANSEL6	-	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	-	1	1	1	1	1

08Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELH	ANSEL15	ANSEL14	ANSEL13	ANSEL12	ANSEL11	ANSEL10	ANSEL9	ANSEL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

ANSEL[15:0]：A/D引脚数模控制位

1：模拟模式，作为模拟信号口，仅可作为AD通道的模拟输入

0：数字模式，作为数字输入或输出口

注： ANSEL 上电初始值为 B'xxx1 1111'，即作为模拟输入。无论是否应用到 AD，均需要在上电后,对 IO 操作之前按需配置，否则 IO 口可能无法受控于对应的端口寄存器，状态将不确定。

ANSEL[4:0]对应 AN4~AN0（PA4~PA0），ANSEL[7:6]对应 AN7、AN6（PA7、PA6）;

ANSEL[8]对应 AN8（即 PA5）;

ANSEL[13:9]对应 AN13~AN9（PB4~PB0），ANSEL[15:14]对应 AN15、AN14（PB7、PB6）。

7.2 I/O 口上拉控制寄存器

028~02Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WPUx	WPUx7	WPUx6	WPUx5	WPUx4	WPUx3	WPUx2	WPUx1	WPUx0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

x = A、B、C、D、E、F

WPUx[7:0]：PORTx[7:0]的上拉使能位

1 = 上拉禁止

0 = 上拉使能

078h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	RBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

bit [7] **RBPUB**：PORTB上拉使能位

1 = PORTB上拉由WPUB决定

0 = 使能PORTB上拉(此时无论WPUB为何值PORTB都上拉)

注：1. 注意此处上拉控制寄存器逻辑，0 为使能，1 为禁止；

2. I/O 禁止悬空状态，输入状态需设定内部上拉电阻。

7.3 I/O 口下拉控制寄存器

034h~039h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WPDx	WPDx7	WPDx6	WPDx5	WPDx4	WPDx3	WPDx2	WPDx1	WPDx0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

x＝A、B、C、D、E、F

WPDx[7:0]：PORTx[7:0]的下拉使能位

1＝下拉禁止

0＝下拉使能

注：1. 注意此处下拉控制寄存器逻辑，0 为使能，1 为禁止；
2. 当端口设置为输出时，下拉无效；
3. 当下拉打开时，上拉无效。

7.4 PORT 驱动控制寄存器

04Dh~051h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LxEN	LxEN7	LxEN6	LxEN5	LxEN4	LxEN3	LxEN2	LxEN1	LxEN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

x=A/B/C/D/E/F

LxEN[7:0]：LxEN[7:0]PORT口驱动电流控制位

0＝ 驱动电流不变

1＝1/3源电流及漏电流

7.5 I/O 口数据寄存器

01Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	x	x	x	x	x	x

01Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	x	x	x	x	x	x

01Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	x	x	x	x	x	x

01Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	x	x	x	x	x	x

020h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTE	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	x	x	x	x	x	x

021h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTF	PORTF7	PORTF6	PORTF5	PORTF4	PORTF3	PORTF2	PORTF1	PORTF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	x	x	x	x	x	x	x	x

- 例：PORTA0做AD输入，PORTA[4:1]做输入，PORTB口输出0XFF

```
        ……                          ;其他初始化
        BCF     STATUS,RP0          ;Bank0
        MOVLW   B'00000001'         ;PA0为模拟信号口，其他为数字信号口
        MOVWF   ANSELL
        CLRF    ANSELH              ;PB为数字口
        MOVLW   0xFF
        MOVWF   TRISA               ;PA口作为输入口
        CLRF    PORTA               ;清空PORTA
        MOVLW   0x00
        MOVWF   TRISB               ;设置PB口为输出
        MOVLW   0xFF
        MOVWF   PORTB               :PB口输出高电平
TEST1：                             ;下面访问PORTA作为输入检测得到的数据
        BTFSS   PORTA,1             ;测试PORTA1输入电平
        GOTO    TEST1               ;PORTA1检测到低电平
        GOTO    TEST2               ;PORTA1检测到高电平
TEST2：
        ……                          ;其他程序
```

7.6 管脚配置寄存器

04Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORCTR	-	-	-	CCPCT	SPPCT1	SPPCT0	UAPCT1	UAPCT0
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W

POR的值	-	-	-	0	0	0	0	0

bit[1:0] **UAPCT[1:0]** UART 管脚配置位

00 = RX 配置在 PORTC3，TX 配置在 PORTC2（默认）

01 = RX 配置在 PORTF4，TX 配置在 PORTF5

10 = RX 配置在 PORTB0，TX 配置在 PORTB1

11 = RX 配置在 PORTD7，TX 配置在 PORTD6

bit[3:2] **SPPCT[1:0]** SPI 管脚配置位

00 = SS/SCK/MISO/MOSI 配置在 PORTC7/ PORTC6/ PORTC5/ PORTC4（默认）

01 = SS/SCK/MISO/MOSI 配置在 PORTF4/PORTF5/PORTF6/PORTF7

10 = SS/SCK/MISO/MOSI 配置在 PORTB0/ PORTB1/ PORTB2/ PORTB3

11 = SS/SCK/MISO/MOSI 配置在 PORTD7/ PORTD6/ PORTD5/ PORTD4

bit[4] **CCPCT** CCP 管脚配置位

0 = CCP1/CCP2 管脚配置在 PORTB2/PORTB3（默认）

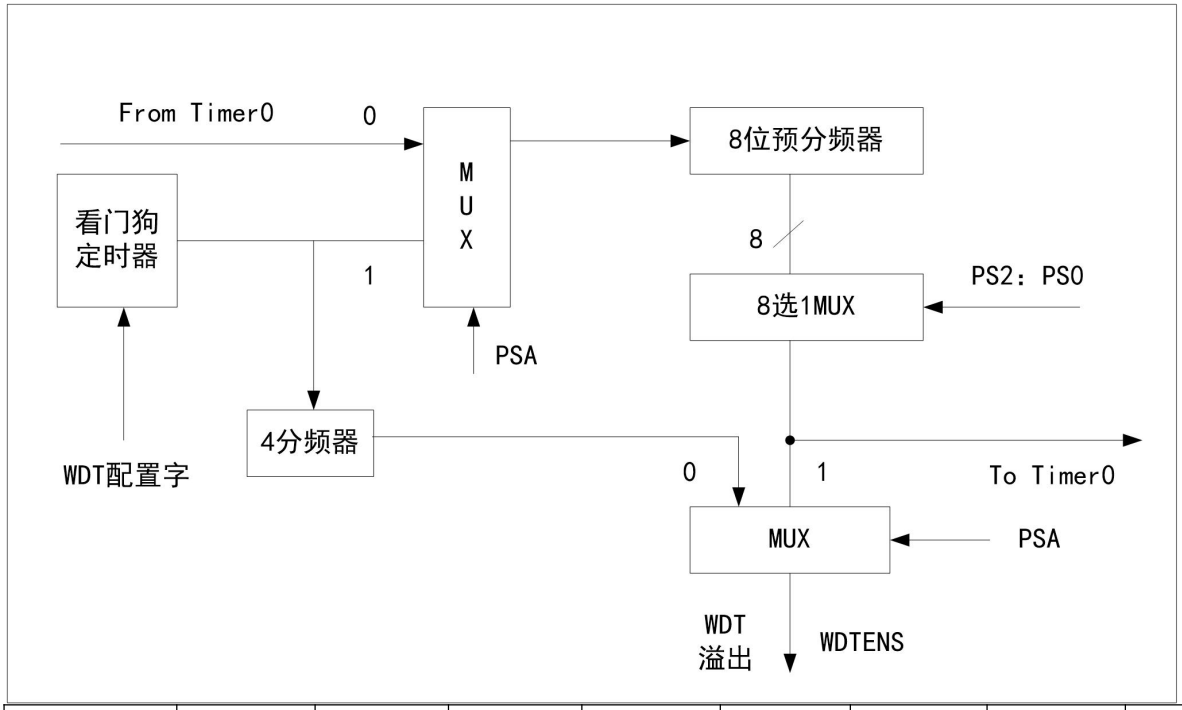
1 = CCP1/CCP2 管脚配置在 PORTA6/PORTA7

注：管脚复用功能优先级：SPI > UART > CCP > IO。

8 定时器/计数器

8.1 看门狗定时器

HC18P23xL的看门狗定时器与Timer0定时器/计数器共用一个预分频器。当PSA为0时，看门狗定时器每72ms（典型值）产生一个溢出信号；当PSA为1时，WDT溢出时间由预分频器OPTION[2:0]设置决定，具体请参考8.2节Timer0定时器/计数器。



079h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	LVD2EN	LVD1EN	-	WDTENS	LVD2F	LVD1F	POR	BOR
R/W	R/W	R/W	-	R/W	R	R	R/W	R/W
POR的值	0	0	-	1	q	q	q	q

bit [4] **WDTENS**：硬件看门狗软件使能位（需配置字使能看门狗，否则该位无效）

1 = 软件使能硬件看门狗定时器

0 = 软件屏蔽硬件看门狗定时器

注：看门狗的使能逻辑 看门狗使能 = 芯片配置字使能（WDTEN） & 软件使能（WDTENS）。

当系统处于休眠或绿色模式，看门狗定时器溢出将唤醒SLEEP并使其返回高频或低频模式，程序从SLEEP指令下一条开始执行。

注：

1. 对看门狗清零之前，检查I/O口的状态和RAM 的内容可增强程序的可靠性；
2. 不能在中断中对看门狗清零，否则无法侦测到主程序跑飞的状况；
3. 程序中应该只在主程序中有一次清看门狗的动作，这种架构能够最大限度的发挥看门狗的保护功能。

➢ 例：看门狗在主程序中的应用

MAIN：

```
BCF     STATUS,RP0      ; Bank 0
BSF     PCON,WDTENS     ;软件使能WDT
…                       ;检查IO状态是否正确
…                       ;检查RAM是否正确
GOTO    ERR             ;检查IO/RAM出错，进入出错处理程序
CLRWDT                  ;在整个程序中，仅有一条清狗指令
…
CALL    SUB1
CALL    SUB2
…
GOTO    MAIN
```

➢ 例：在休眠状态下，屏蔽看门狗功能，可以节省系统功耗

```
…
BCF     STATUS,RP0      ; Bank 0
BCF     PCON,WDTENS     ;软件屏蔽看门狗功能
BCF     OSCCON,T0OSCEN  ;禁止低频晶体振荡器
SLEEP                   ;进入休眠模式
BSF     PCON,WDTENS     ;唤醒后,重新使能看门狗功能
…
```

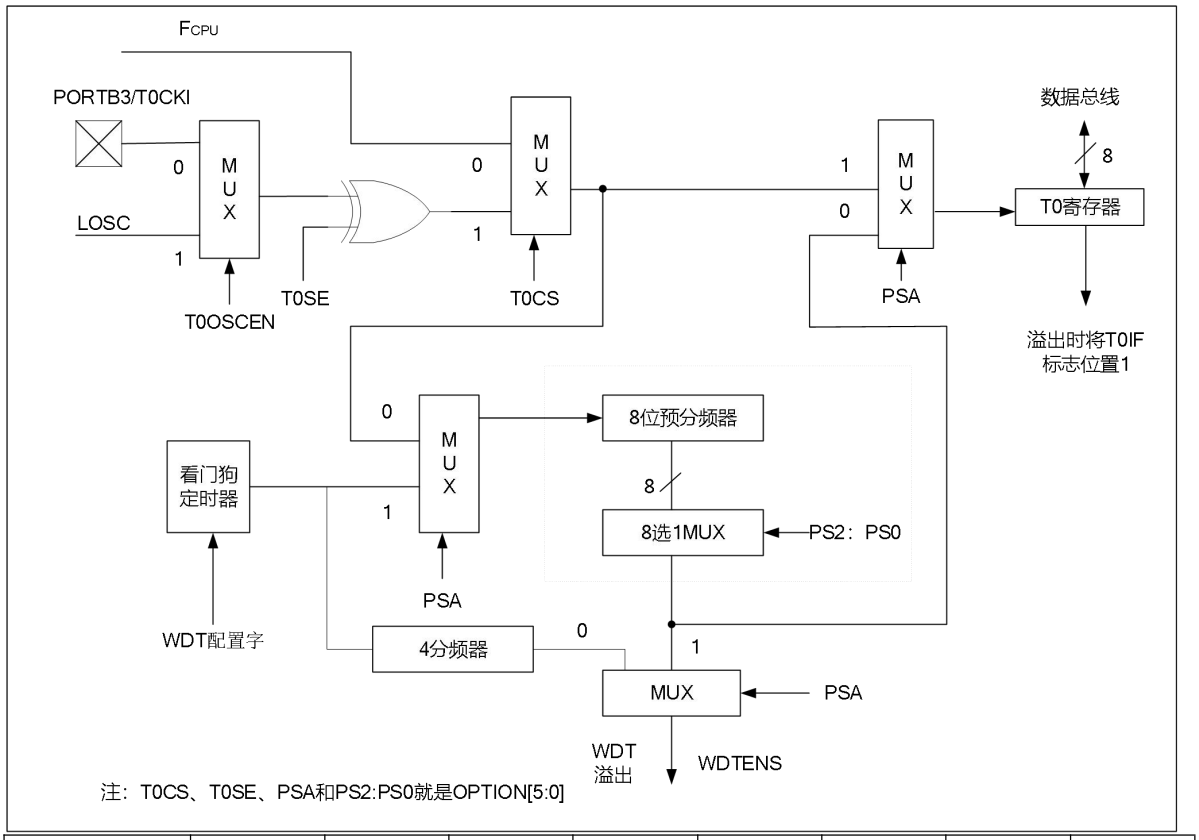
➢ 例：对看门狗定时器操作，看门狗定时器使能和清零

```
BCF     STATUS,RP0      ; Bank 0
BSF     PCON,WDTENS     ;使能看门狗
CLRWDT                  ;看门狗定时器清零
…
```

8.2 Timer0 定时器/计数器

Timer0 定时器/计数器模块具有如下功能：

- 8 位可编程定时器
- 外部事件计数器
- 绿色模式定时唤醒



07Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCCON	T0OSCEN	-	-	-	-	-	HXEN	SCS
R/W	R/W	-	-	-	-	-	R/W	R/W
POR的值	0	-	-	-	-	-	0	q

078h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	RBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

看门狗定时器与Timer0定时器/计数器共用一个预分频器，当PSA=1预分频器分配给WDT时，Timer0在所选中时钟源的每个周期递增；当PSA=0预分频器分配给Timer0时，Timer0根据PS[2:0]值选择的预分频时钟递增。

Timer0的预分频器不可寻址，当预分频器分配给Timer0时，对Timer0计数寄存器的写操作可以对预分频器清零。

1、Timer0预分频比选择

PS[2:0]	Timer0预分频比	WDT预分频比	WDT溢出时间（典型值）
000	1 : 2	1：1	18ms
001	1 : 4	1：2	36 ms
010	1 : 8	1：4	72ms
011	1 : 16	1：8	144ms
100	1 : 32	1：16	288ms
101	1 : 64	1：32	576ms

110	1 : 128	1：64	1.152s
111	1 : 256	1：128	2.304s

2、Timer0 工作模式选择

T0CS	T0OSCEN	T0SE	Timer0工作状态
0	x	x	定时器模式，计数时钟 FCPU， 休眠和绿色模式下停止
1	0	0	计数器模式，计数时钟 T0CKI，上升沿计数 休眠模式下工作，溢出中断可唤醒 SLEEP
1	0	1	计数器模式，计数时钟 T0CKI，下降沿计数 休眠模式下工作，溢出中断可唤醒 SLEEP
1	1	0	定时唤醒模式，计数时钟LOSC，上升沿计数 绿色模式下工作，溢出中断可唤醒SLEEP
1	1	1	定时唤醒模式，计数时钟LOSC，下降沿计数 绿色模式下工作，溢出中断可唤醒SLEEP

注：
Timer0 工作模式的选择需符合上表描述，选择除上表以外情况可能会造成程序运行混乱，请谨慎操作。

- 例：Timer0工作于定时器模式，计数时钟为Fcpu，T0计满到FF后溢出进入中断

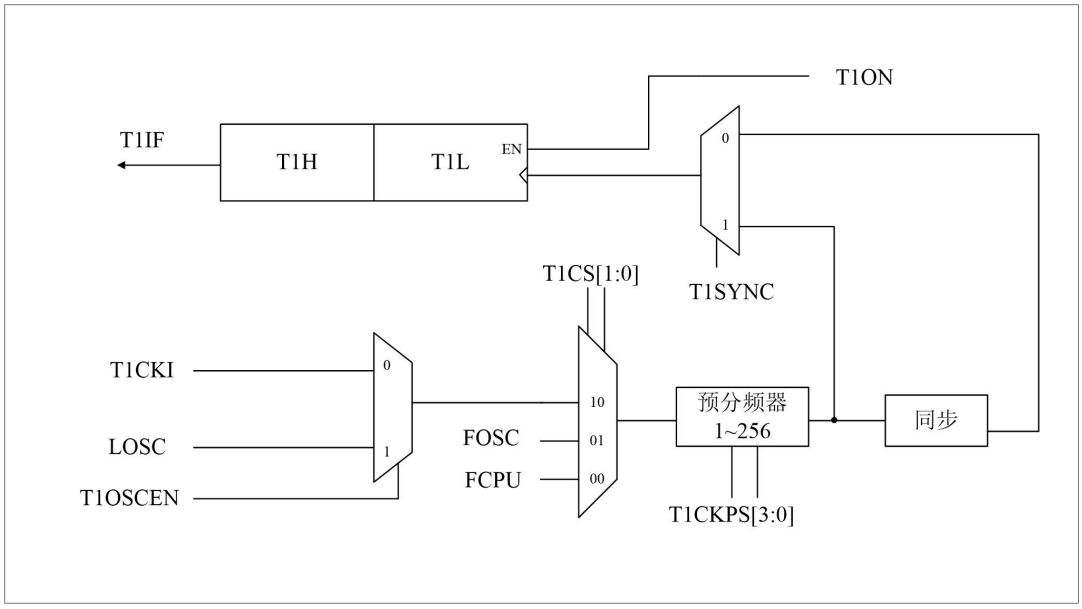
```
        MOVLW   0x01
        BCF     STATUS,RP0          ;Bank0
        MOVWF   OPTION              ;定时器模式，分频比为1:4
        MOVLW   0x00
        MOVWF   T0                  ;T0赋初值
        BSF     INTCON,T0IE
        BCF     INTCON,T0IF
        BSF     INTCON,GIE

T0INT_SERVICE:
        …                           ;保存STATUS、W 和PCLATH
        BCF     STATUS,RP0          ;Bank0
        BTFSS   INTCON,T0IF         ;检测T0IF
        GOTO    EXIT_INT            ;T0IF = 0，退出中断
        BCF     INTCON,T0IF         ;T0IF 清零
        …                           ;TIMER0 中断服务程序
        …
EXIT_INT:
        …                           ;恢复STATUS、W和PCLATH
        RETFIE                      ;退出中断
```

8.3 Timer1 定时器/计数器

Timer1 定时器/计数器模块具有如下功能：

- 16 位可编程定时器
- 外部事件计数器，可编程选择同步、异步功能
- 绿色模式定时唤醒



8.3.1 Timer1 控制寄存器

05Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CON	T1CS1	T1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	-	T1ON
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W
POR的值	0	0	0	0	0	0	-	0

Timer1时钟源选择

T1CS1	T1CS0	T1OSCEN	时钟源
0	0	x	指令时钟（F_{CPU}）
0	1	x	系统时钟（F_{sys}）
1	0	0	T1CKI引脚上的外部时钟
1	0	1	低频系统时钟

注：

Timer1 时钟源的选择需符合上表描述，选择除上表以外情况会造成程序运行不正常，请谨慎操作。

Timer1输入时钟预分频比选择

T1CKPS[1:0]	Timer1 预分频比
00	1 : 1
01	1 : 2
10	1 : 4

11	1：8

Timer1的预分频器不可寻址，可以通过对Timer1计数寄存器写操作将预分频器清零。

Timer1工作模式选择

T1ON	T1CS[1:0]	T1OSCEN	T1SYNC	Timer1工作模式
1	00	x	x	定时器模式，休眠和绿色模式下停止
1	01	x	x	定时器模式，休眠和绿色模式下停止
1	10	0	0	同步计数器模式，休眠模式下停止
1	10	0	1	异步计数器模式，休眠模式下工作，溢出中断可唤醒SLEEP
1	10	1	0	同步定时唤醒模式，绿色模式下停止，溢出中断不能唤醒SLEEP
1	10	1	1	异步定时唤醒模式，绿色模式下工作，溢出中断可唤醒SLEEP

注：

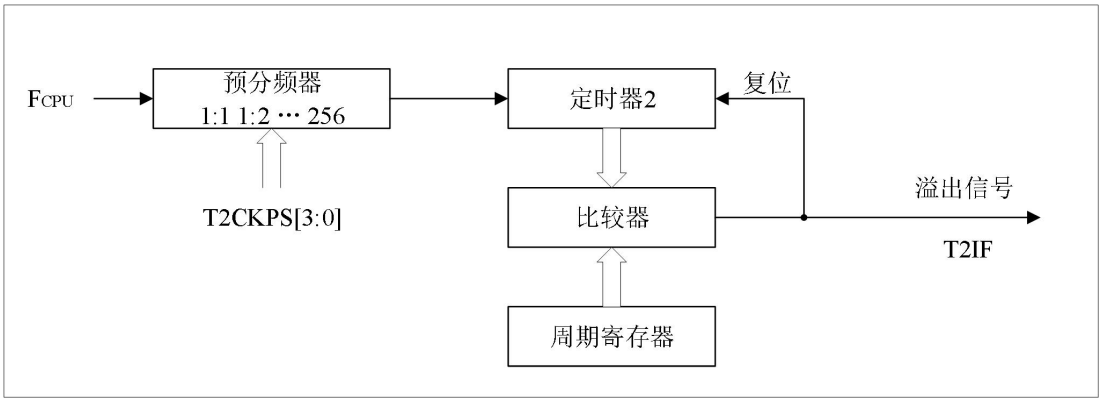
1、T1 为 16 位计时器，在溢出中断重新赋值时应先 T1H，后 T1L，避免 T1L 在操作中的进位被覆盖；清空时则应先 T1L 后 T1H，避免 T1L 进位意外进入 T1H 造成清空失败；

2、Timer1 工作于同步计数器模式和同步定时唤醒模式时，不能唤醒 SLEEP 或绿色模式；

3、Timer1 工作模式的选择需符合上表描述，选择除上表以外情况可能会造成程序运行混乱，请谨慎操作。

8.4 Timer2 定时器

Timer2定时器具有8位预分频器和8位周期寄存器（PR2），Timer2定时器的输入时钟为指令时钟FCPU，输入时钟通过预分频器产生Timer2计数时钟，当计数到与周期寄存器（PR2）的值相同时，在下一指令周期产生Timer2溢出信号，可根据实际需要选择不同的预分频比及设置周期寄存器的值，产生不同溢出时间。



8.4.1 Timer2 定时器

05Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CON	-	T2CKPS3	T2CKPS2	T2CKPS1	T2CKPS0	T2ON	-	-
R/W	-	R/W	R/W	R/W	R/W	R/W	-	-
POR的值	-	0	0	0	0	0	-	-

bit [2] **T2ON**： Timer2模块使能位

1 = 使能Timer2模块

0 = 禁止Timer2模块

Timer2具有一个8位可编程预分频器，关闭Timer2模块和对Timer2计数寄存器或T2CON寄存器写操作都将对预分频器清零。

T2CKPS[3:0]	Timer2 预分频比
0000	1 : 1
0001	1 : 2
0010	1 : 4
0011	1：8
0100	1：16
0101	1：32
0110	1：64
0111	1：128
1xxx	1：256

8.4.2 Timer2 计数寄存器

05Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2	Timer2计数寄存器							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

8.4.3 Timer2 周期寄存器

05Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PR2	Timer2周期寄存器							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	1	1	1	1	1	1	1	1

Timer2 定时器的输入时钟为指令时钟 FCPU，输入时钟通过预分频器产生 Timer2 计数信号，当计数到与周期寄存器（PR2）的值相同时产生 Timer2 溢出信号。

Timer2 溢出时间 = （PR2 + 1）× 预分频比/Fcpu。

8.5 CCP 模块

HC18P23xL具有2个独立的CCP模块CCP1和CCP2，每个CCP模块具有三种模式：

- 捕捉
- 比较
- PWM

CCP模块的时基由Timer1和Timer2提供。

1、CCP模块的时基

CCP模式	时钟源
捕捉	Timer1
比较	Timer1
PWM	Timer2/Timer1

2、CCPxCON寄存器

082h、085h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCPxCON	-	-	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	-	0	0	0	0	0	0

bit [5:4] **DCxB[1:0]**：PWM占空比最低有效位

捕捉模式：未使用

比较模式：未使用

PWM模式：PWM占空比的低2位，高8位是CCPRxL寄存器

bit [3:0] **CCPxM[3:0]**：CCPx模式选择位

0000 = 捕捉/比较/PWM关闭（复位CCP模块）

0001 = 未使用（保留）

0010 = 比较模式，匹配时输出翻转电平（PIRx寄存器的CCPxIF位置1）

0011 = 未使用（保留）

0100 = 捕捉模式，每个下降沿

0101 = 捕捉模式，每个上升沿

0110 = 捕捉模式，每4个上升沿

0111 = 捕捉模式，每16个上升沿

1000 = 比较模式，匹配时输出高电平（PIRx寄存器的CCPxIF位置1）

1001 = 比较模式，匹配时输出低电平（PIRx寄存器的CCPxIF位置1）

1010 = 比较模式，匹配时仅产生软件中断（PIRx寄存器的CCPxIF位置1，CCPx引脚不受影响）

1011 = 比较模式，触发特殊事件（PIRx寄存器的CCPxIF位置1，Timer1计数寄存器复位， CCPx引脚不受影响。）

11xx = PWM模式

8.5.1 捕捉模式

在输入捕捉模式，适合用于测量引脚输入周期性方波信号的周期、频率和占空比等，也适合用于测量引脚输入的非周期性矩形方波脉冲信号的宽度、到达时刻或消失时刻等参数。

当CCPx模块工作于捕捉模式时，一旦有下列事件在引脚CCPx上发生，CCPRx寄存器立即捕捉下这

一时刻的Timer1计数值：

- 每个下降沿
- 每个上升沿
- 每 4 个上升沿
- 每 16 个上升沿

CCPxCON寄存器的CCPxM[3:0]设置的是预分频器，关闭CCP模块或者CCP模块不在捕捉模式，预分频计数器将会被清零。为避免错误中断，可在改变预分频比前通过清零 CCPxCON寄存器来关闭CCP模块。

在捕捉模式下，Timer1必须运行在定时器模式或同步计数器模式。

使用注意：

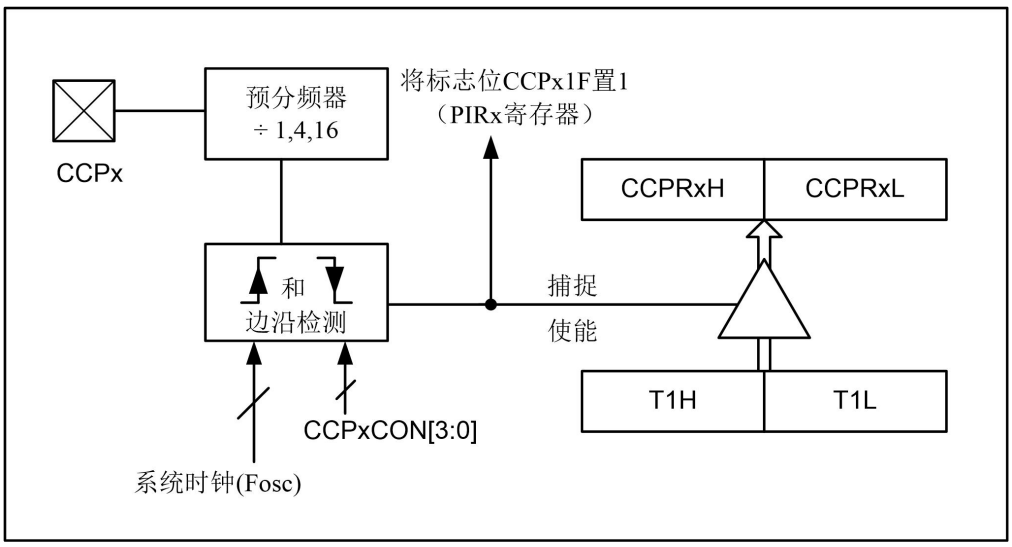
1、 在捕捉模式下，CCPx引脚必须由相应的方向控制器设定为输入方式；

2、 CCPxCON寄存器的CCPxM[3:0]设置的是预分频器，关闭CCP模块或者CCP模块不在捕捉模式，预分频计数器将会被清零。为避免错误中断，可在改变预分频比前通过清零 CCPxCON寄存器来关闭CCP模块。如果需要中途改变预分频器的分频比，建议使用以下程序片段：

```
BCF     STATUS,RP0      ;BANK0
CLRF    CCPxCON         ;关闭CCPx模块
MOVLW   NEW_CAPT_PS     ;选取新的分频比（1:1、1:4、1:16）
MOVWF   CCPxCON         ;赋予CCPxCON寄存器，并打开CCPx模块
```

3、 当一个捕捉事件发生后，硬件自动将CCPx的中断标志位CCPxIF置1，表示产生了一次CCPx捕捉中断。CCPxIF位必须用软件重新清零。当CCPRx寄存器中的值还未被程序读取，而又发生了另一个新的捕捉事件时，原先的值将被新的值覆盖掉；

4、在捕捉模式下，Timer1必须运行在定时器模式或同步计数器模式。



CCPx引脚上发生变化时，CCPRxH:CCPRxL捕捉Timer1计数寄存器的16位值，PIRx寄存器的中断标志位CCPxIF被置1。如果在CCPRxH和CCPRxL寄存器的值被读出之前又发生另一次捕捉，那么原来的捕捉值会被新捕捉值覆盖。

8.5.2 比较模式

输出比较模式，适用于从引脚上输出不同宽度的矩形正脉冲、负脉冲、延时驱动信号、可控硅驱动信号、步进电机驱动信号等。

在比较模式下，CCPRxH:CCPRxL寄存器对成为了Timer1的周期寄存器，一旦Timer1计数寄存器对与CCPRxH和CCPRxL寄存器对发生匹配， Timer1计数寄存器对在Timer1时钟的下一个上升沿复位， CCPx模块根据CCPxM[3:0] 控制位的配置进行相应操作：

- CCPx 引脚输出翻转电平
- CCPx 引脚输出高电平
- CCPx 引脚输出低电平
- 仅产生软件中断
- 产生特殊事件触发信号

所有比较模式都能产生CCP中断。

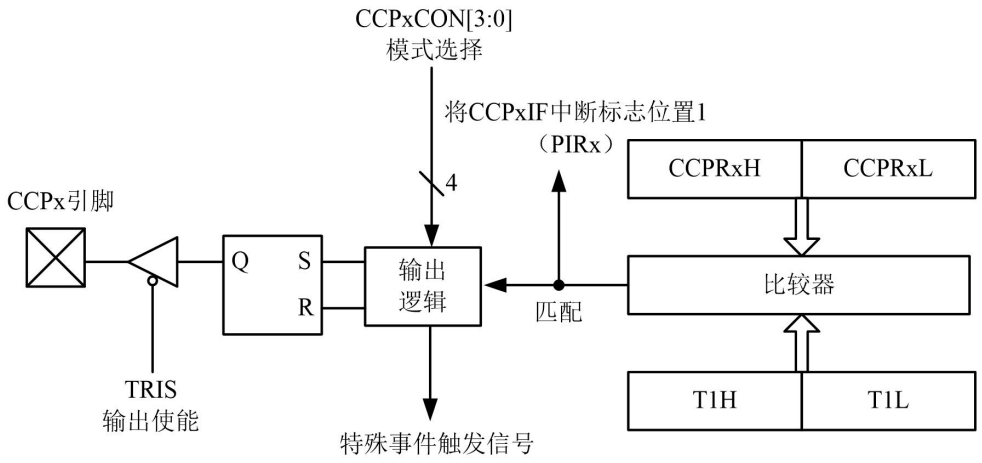
使用注意：

1、 当选择产生特殊事件触发信号时，如果ADC被使能，则启动一次ADC转换(仅限于CCP1)。在此模式下CCPx模块不会对CCPx引脚进行控制；

2、 在比较模式下，CCPx引脚必须由相应寄存器设定为输出模式，以便作为比较器的输出端使用；

3、 应该注意的是，如果对控制寄存器CCPxCON进行重新赋值，将会迫使CCPx引脚输出一个默认的低电平，而这并非是正常的比较输出结果；

4、 在比较模式下，Timer1必须运行在定时器模式或同步计数器模式下。



特殊事件触发信号将：

- 清零T1H和T1L寄存器。
- 不会将PIR1寄存器的中断标志位T1IF置1。
- 将ADON位置1以启动ADC转换（仅限CCP1）

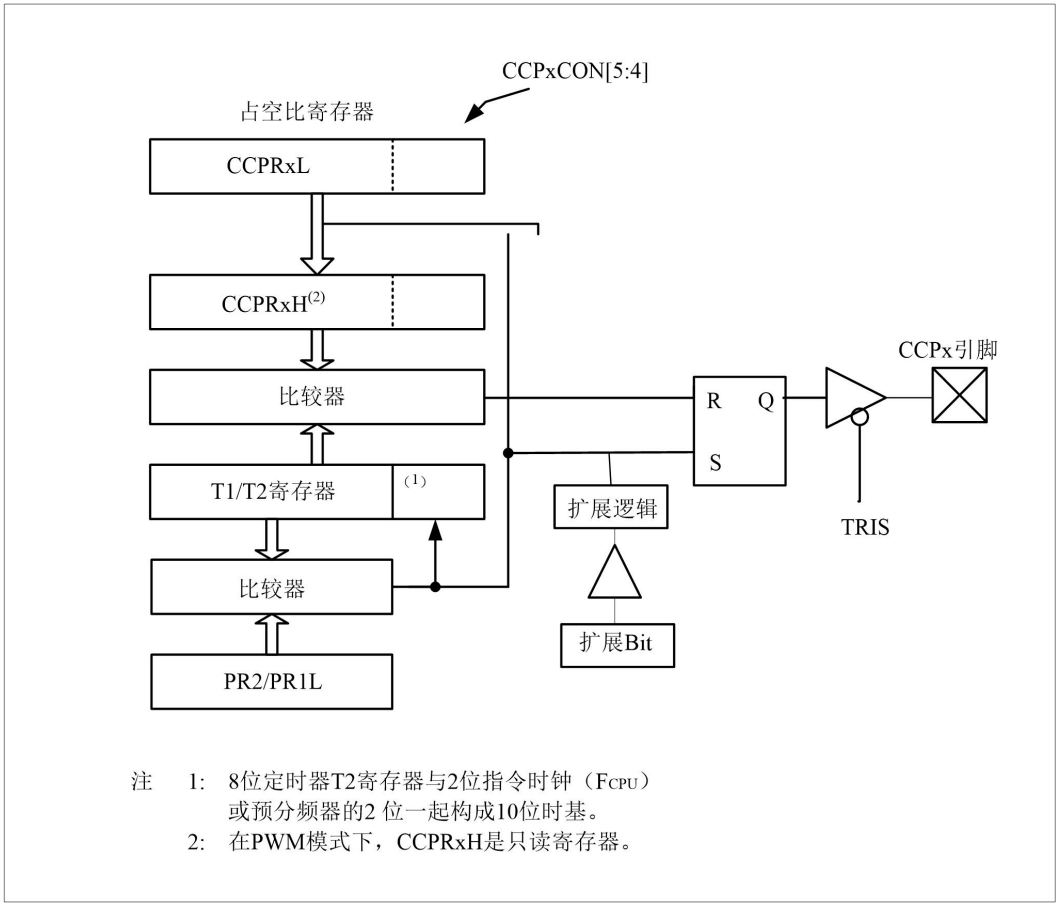
8.5.3 PWM 模式

脉宽调制，PWM(pulse width modulation)输出工作模式，适用于从引脚上输出脉冲宽度随时可调的PWM 信号。例如，实现直流电动机调速、简易 D/A 转换器、步进电机的变频控制等。

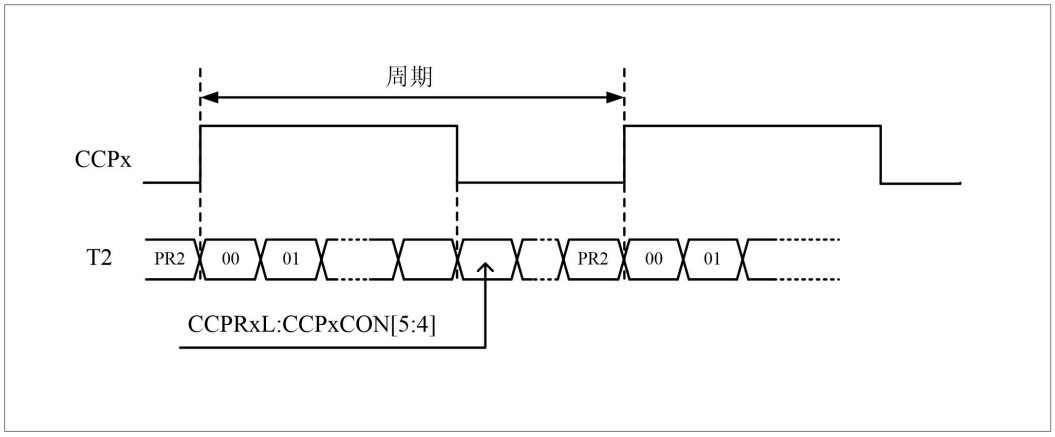
8.5.3.1 PWM不选择扩展

一、 PWM 时钟源为 Timer2

在PWM模式下，当Timer2计数寄存器中的值与PR2寄存器中的值发生匹配时，在下一个计数时钟Timer2计数寄存器被清零，CCPx引脚被置1（如果PWM占空比为0%，CCPx引脚将不会被置1），PWM占空比值从CCPRxL锁存到CCPRxH（在Timer2计数寄存器中的值与PR2寄存器中的值发生匹配前，占空比值不会被锁存到CCPRxH中）。



注 1: 8位定时器T2寄存器与2位指令时钟（F_{CPU}）或预分频器的2 位一起构成10位时基。
2: 在PWM模式下，CCPRxH是只读寄存器。



✓ **以下公式中，当芯片配置为 4T 模式时，n=1;当芯片配置为 2T 模式时，n=1/2**

PWM 周期：

$$\text{PWM 周期} = [(PR2) + 1] \bullet 4 \bullet n \bullet Tsys$$

（Timer2 预分频值）

注： Tsys = 1/Fsys

PWM脉冲宽度：

$$\text{脉冲宽度} = (CCPRxL:CCPxCON[5:4]) \bullet n \cdot Tsys \cdot (\text{Timer2 预分频值})$$

注： Tsys = 1/Fsys

如果脉冲宽度值比周期长，则指定的PWM引脚将保持不变。

PWM占空比：

$$\text{占空比} = \frac{(CCPRxL:CCPxCON<5:4>)}{4(PR2 + 1)}$$

当时钟模式选择配置字选择为2T时，PWM占空比由CCPRxL寄存器和CCPxCON寄存器的DCxB[1]位决定。

PWM分辨率：

$$\text{分辨率} = \frac{Log[4(PR2 + 1)]}{Log(2)} \text{ 位}$$

分辨率是PR2寄存器值的函数，当PR2为255时PWM最大分辨率为10位（时钟模式选择配置字选择为2T时，PWM最大分辨率为9位）。

PWM使用：

1. 设置相关TRISB位为1，禁止CCPx引脚输出
2. 设置PWM周期，输入PR2寄存器值
3. 设置CCPxCON寄存器，将CCP模块配置为PWM模式
4. 设置PWM占空比，输入CCPRxL寄存器和CCPxCON[5:4]寄存器值
5. 配置和启动Timer2
   - 将PIR1寄存器的T2IF中断标志位清零
   - 设置T2CON寄存器的T2CKPS位，选择Timer2预分频
   - 将T2CON寄存器的T2ON置1，使能Timer2
6. 设置PWM输出
   - 等待Timer2溢出，PIR1寄存器的T2IF位置1
   - 将TRISB2或TRISB3(TRISA6或TRISA7)位清零，让CCPx引脚输出

7.如何关闭PWM输出

- 将TRISB2或TRISB3(TRISA6或TRISA7)位置1，设引脚输入
- 设置CCPxCON寄存器的CCPxCON[3:0]设为0000，关PWM功能，CCPx用作I/O口
- 根据需要,设PB2或PB3(PA6或PA7)输出为高电平或者低电平

➢ 例：配置PWM输出38K驱动方波，采用4M/4T模式。

```
        ORG     0000H           ;复位向量
        GOTO    MAIN
        ORG     0004H           ;中断
        ……
        RETFIE

MAIN:
        BCF     STATUS,RP0      ;BANK0
        BCF     PCON,WDTENS     ;DISABLE WDT
        MOVLW   0x00
        MOVWF   PORCTR          ;设置端口映射为PB2
        CLRF    OPTION          ;使用高频时钟
        BSF     TRISB,2         ;CCPx口设为输入

        MOVLW   D'25'
        MOVWF   PR2             ;设置PR2为26μs

        MOVLW   H'0D'
        MOVWF   CCPR1L          ;占空比高8位
        MOVLW   B'00001100'     ;选PWM模块，填写占空比低两位
        MOVWF   CCP1CON         ;脉冲宽度13μs

        BCF     PIR1,T2IF
        CLRF    T2CON           ;分频比1：1
        BSF     T2CON,T2ON      ;开T2
        BSF     PIE1,T2IE

        BTFSS   PIR1,T2IF       ;等待T2溢出
        GOTO    $-1
        BCF     PIR1,T2IF       ;清除中断标志
        BCF     PIE1,T2IE
        BCF     TRISB,2         ;配置完成，PB2将持续输出38K
```

二、 PWM2时钟源为Timer1

05Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PR1L	Timer1周期寄存器低字节							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR值	0	0	0	0	0	0	0	0

060h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PR1CON	PWM1T1	PWM1T0	PWM2T1	PWM2T0	T1CKPS3	T1CKPS2	PWMPR1	PR1EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR值	0	0	0	0	0	0	0	0

bit[0] **PR1EN**：Timer1可设周期使能位

1= Timer1为可设周期的8位Timer

0= Timer1为16位Timer

bit[1] **PWMPR1**：Timer1提供PWM时钟源使能位

1=CCP2的PWM的时钟源由8位可设周期的Timer1提供（当PR1EN有效时，该位可用）

0=CCP2的PWM的时钟源由Timer2提供

bit[3:2] **T1CKPS**[3:2]:8位可设周期Timer1的分频比控制位的高两位

T1CKPS[3:0]	Timer1 预分频比
0000	1 : 1
0001	1 : 2
0010	1 : 4
0011	1 : 8
0100	1 : 16
0101	1 : 32
0110	1 : 64
0111	1 : 128
1xxx	1 : 256

使用注意：

1. 将寄存器PR1CON[bit1~bit0]置1，使PWM2的时钟源选择位可设周期的8位Timer1，T1L必须赋值为0X00，其他使用操作可参考由Timer2提供时钟源的PWM2的使用。

2. PWM2的时钟源选择T1的Fsys时，配置字OPTION的时钟模式应选择4T，不要选择2T。否则会出现占空比异常的情况。

8.5.3.2 PWM选择扩展

当PWM选择扩展周期时，4 个调制周期为一个输出周期，此时PWM最高可扩展到12位。也就是说当选择扩展PWM模式时，PWM将为4个波形一个周期，此时不论你在哪一个波形输出时改变PWM的数据存储器都将在下一个周期才生效。

扩展周期控制寄存器PR1CON

060h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PR1CON	PWM1T1	PWM1T0	PWM2T1	PWM2T0	T1CKPS3	T1CKPS2	PWMPR1	PR1EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR值	0	0	0	0	0	0	0	0

bit[7:6] **PWM1T[1:0]**：PWM1的扩展周期选择位

00： 无扩展周期

01： 扩展周期为1

10： 扩展周期为1、2

11： 扩展周期为1、2、3

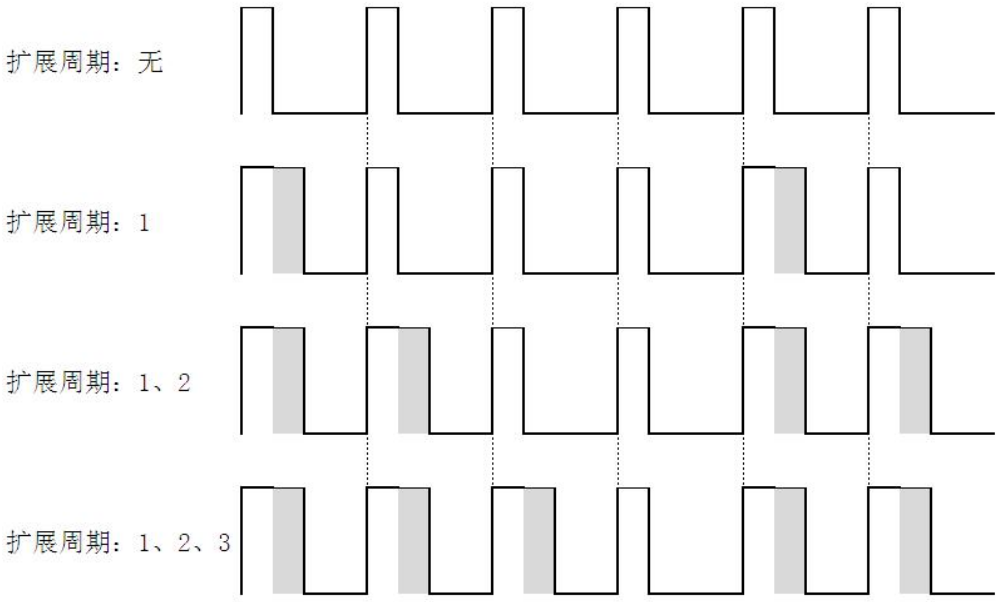
bit[5:4] **PWM2T[1:0]**：PWM2的扩展周期选择位

00： 无扩展周期

01： 扩展周期为1

10： 扩展周期为1、2

11： 扩展周期为1、2、3



9 PWM模块

9.1 概述

- 3 组带死区互补 PWM 或 6 路独立 PWM 输出
- 提供每个 PWM 周期溢出中断，但中断共用同一向量入口
- 输出极性可选择
- 提供出错侦测功能可紧急关闭 PWM 输出
- PWM 工作时钟源可设定时钟分频比
- PWM 可做定时器/计数器使用

HC18P23xL 集成了一个 12 位 PWM 模块 PWM0 和两个 8 位 PWM1/PWM2，三个模块各有一个计数器，PWM0 的计数器由 EPWM0 或 EPWM01 来控制，只要它们中的任何一个使能，都可以启动计数器，计数器的时钟源通过 PWM0C 控制寄存器里的 CK0 来选择。

如果 EPWM0 使能，而没有通过功能引脚映射寄存器进行 PWM0 的映射，这样也不会从芯片管脚上输出 PWM0，这时候 PWM0 的计数器可以当一个定时器来使用，当计数器溢出时，如果中断允许也会产生 PWM 中断。

如果 EFLT 置 1，PWM0/PWM1/PWM2 输出和其互补输出可由 FLT 引脚输入信号变化自动关闭。一旦检测到 FLT 引脚输入有效电平，PWM 输出会立即关闭，但 PWM 内部计数器仍在继续运行，这样方便在 FLT 引脚错误去除后继续 PWM 输出。在 FLT 输入信号有效期间，FLTS 位无法清除。只有当 FLT 输入信号消失后，才能软件清除 FLTS 状态位，此时 PWM 恢复正常输出。

PWM 定时器也为 PWM0/1/2 提供 3 个中断源，在每个 PWM 周期都会产生中断。它们有不同的控制位和标志位。这样用户可以实现每个 PWM 周期中更改下一次循环的周期或占空比。

9.2 PWM 相关寄存器

25Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMEN	-	EFLT	EPWM21	EPWM11	EPWM01	EPWM2	EPWM1	EPWM0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	0	0	0	0	0	0	0

bit [6] **EFLT**：FLT引脚配置位

1 = PWM故障检测输入引脚

0 =普通I/O口或其他功能

bit [5] **EPWM21**：PWM21输出控制位

1 = PWM21输出允许

0 = PWM21 输出禁止，用作 I/O 功能

bit [4] **EPWM11**：PWM11输出控制位

1 = PWM11输出允许

0 = PWM11输出禁止，用作I/O功能

bit [3] **EPWM01**：PWM01输出控制位

1 = PWM01输出允许

0 = PWM01 输出禁止，用作 I/O 功能

bit [2] **EPWM2**：PWM2输出控制位

1 = PWM2输出允许

0 = PWM2输出禁止，用作I/O功能

bit [1] **EPWM1**：PWM1输出控制位

1 = PWM1输出允许

0 = PWM1输出禁止，用作I/O功能

bit [0] **EPWM0**：PWM0输出控制位

1 = PWM0输出允许

0 = PWM0 输出禁止，用作 I/O 功能

当PWMEN清零后，PWM输出立即关闭。

FLT 端口主要用于检测异常信号，快速关闭 PWM 输出，FLT 探测到故障后，由硬件执行使 PWM 输出关闭，所以当故障发生后，它可以快速响应，使得 PWM 输出无效以保护连接 PWM 的大功率器件。当使能 FLT 引脚故障检测功能后，端口禁止任何上下拉电阻功能。

如果 EFLT 位清零，则表示 FLT 端口对 PWM 定时器输出控制无效。

25Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FLTM	-	-	FLT2M1	FLT2M0	FLT1M1	FLT1M0	FLT0M1	FLT0M0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	-	0	0	0	0	0	0

输出故障时，PWM 口输出状态：

FLTxM[1:0]：PWMx 口 FLT 故障时后，端口输出状态

00 = PWMx 输出高电平

01 = PWMx0 输出高电平，PWMx1 输出低电平

10 = PWMx0 输出低电平，PWMx1 输出高电平

11 = PWMx 输出低电平

9.3 PWM0 模块

9.3.1 PWM0 控制寄存器

HC18P23xL 包含一个 12 位 PWM 模块。PWM 模块可以产生周期和占空比分别可调的脉宽调制波形。PWMC 寄存器用于控制 PWM 模块的时钟，PWMPH/L 寄存器用于控制 PWM 输出波形的周期。PWMDH/L 寄存器用于控制 PWM 模块输出波形的占空比。

在 PWM 输出允许器件可以修改这三个寄存器，但在下一个 PWM 周期修改才会起作用。

25Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0C	-	-	FLTS	FLTC	PWM0S1	PWM0S0	CK01	CK00
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	-	0	0	0	0	0	0

bit [5] **FLTS**：FLT状态位

1 =PWM输出关闭，硬件置1

0 = PWM 正常状态，软件清零

bit [4] **FLTC**：FLT引脚配置位

1 = FLT为高电平时，PWM输出关闭

0 = FLT为低电平时，PWM输出关闭

bit [3:2] **PWM0S[1:0]**：PWM0和PWM01占空比输出模式选择位

11 = PWM0和PWM01均为低有效

10 = PWM0为低有效，PWM01为高有效

01 = PWM0为高有效，PWM01为低有效

00 = PWM0和PWM01均为高有效

bit [1:0] **CK0[1:0]**：PWM1输出控制位

11 = Fosc/16

10 = Fosc/8

01 = Fosc/4

00 = Fosc/2

注：Fosc = 32MHz

注意：

1、 PWM0C寄存器中的FLTS和FLTC位控制PWM0/1/2定时器，而寄存器中的PWM0S,CK0[1:0]只能影响12位PWM定时器。

2、 PWM输出关闭时，PWM0/1/2和PWM01/11/21输出固定电平：

PWMxS[1:0] =00，PWMx和PWMx1均输出固定低电平；

PWMxS[1:0] =01，PWMx输出固定低电平，PWMx1输出固定高电平；

PWMxS[1:0] =10，PWMx输出固定高电平，PWMx1输出固定低电平；

PWMxS[1:0] =11，PWMx和PWMx1均输出固定高电平；

其中x=0/1

3、一旦检测到FLT引脚输入有效电平，PWM输出会立即关闭，但PWM内部计数器仍在继续运行。这样方便在FLT引脚错误去除后继续PWM输出。

4、在FLT输入信号有效期间，FLTS位无法清除。只有当FLT输入信号消失后，才能软件清除FLTS状态位，此时PWM恢复正常输出。

9.3.2 PWM0 周期寄存器

12 位 PWM 周期控制寄存器的高 4 位

25Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0PH	-	-	-	-	PP0.11	PP0.10	PP0.9	PP0.8
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR的值	-	-	-	-	0	0	0	0

12 位 PWM 周期控制寄存器的低 8 位

25Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0PL	PP0.7	PP0.6	PP0.5	PP0.4	PP0.3	PP0.2	PP0.1	PP0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

PWM0输出周期=[PWM0PH:PWM0PL] ×PWM时钟源

当[PWM0PH:PWM0PL]=0x00时：

如果PWM0S[1:0] =00，不管PWM占空比为多少，PWM0输出低电平，PWM01输出低电平。

如果PWM0S[1:0] =01，不管PWM占空比为多少，PWM0输出低电平，PWM01输出高电平。

如果PWM0S[1:0] =10，不管PWM占空比为多少，PWM0输出高电平，PWM01输出低电平。

如果PWM0S[1:0] =11，不管PWM占空比为多少，PWM0输出高电平，PWM01输出高电平。

9.3.3 PWM0 占空比寄存器

12 位 PWM 脉宽控制寄存器的高 4 位

25Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DH	-	-	-	-	PD0.11	PD0.10	PD0.9	PD0.8
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR的值	-	-	-	-	0	0	0	0

12 位 PWM 脉宽控制寄存器的低 8 位

259h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DL	PD0.7	PD0.6	PD0.5	PD0.4	PD0.3	PD0.2	PD0.1	PD0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

PWM输出占空比= [PWM0DH:PWM0DL] X PWM时钟源

当[PWM0PH:PWM0PL]≤[PWM0DLH:PWM0DL]，

如果PWM0S[1:0] =00，PWM0输出高电平，PWM01输出高电平

如果PWM0S[1:0] =01，PWM0输出高电平，PWM01输出低电平

如果PWM0S[1:0] =10，PWM0输出低电平，PWM01输出高电平

如果PWM0S[1:0] =11，PWM0输出低电平，PWM01输出低电平

使用注意事项：

1、 选择PWM模块时钟源。

2、 设置PWM周期/占空比，先设置低位，再设置高位。注意，即使高位数值不变，也要重写一次，否则，低位的修改无效。

3、 通过设置PWM控制寄存器(PWM0C)的PWM0Sx位选择PWM输出模式（高电平/低电平有效）。

4、 通过设置PWM控制寄存器(PWMEN)的EPWM0或EPWM01来允许PWM上桥或下桥输出。

5、 如果PWM周期或者占空比需要改变，操作流程如同步骤2和步骤3说明。修改后的重载计数器的值在下一个周期开始有效。

6、 注意不要对PWM关键寄存器进行误操作。

9.4 PWM1 模块

9.4.1 PWM1 控制寄存器

257h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1C	-	-	-	-	PWM1S1	PWM1S0	CK11	CK10
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR的值	-	-	-	-	0	0	0	0

bit [3:2] **PWM1S[1:0]**：PWM1和PWM11占空比输出模式选择位

11 = PWM1和PWM11均为低有效

10 = PWM1为低有效，PWM11为高有效

01 = PWM1为高有效，PWM11为低有效

00 = PWM1和PWM11均为高有效

bit [1:0] **CK1[1:0]**：PWM1输出控制位

11 = Fosc/16

10 = Fosc/8

01 = Fosc/4

00 = Fosc/2

注：Fosc = 32MHz

9.4.2 PWM1 周期寄存器

周期控制寄存器

256h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1P	PP1.7	PP1.6	PP1.5	PP1.4	PP1.3	PP1.2	PP1.1	PP1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

PWM1输出周期=[PWM1P] X PWM时钟源

当[PWM1P]=0x00时：

如果PWM1S[1:0] =00，不管PWM占空比为多少，PWM1输出低电平，PWM11输出低电平。

如果PWM1S[1:0] =01，不管PWM占空比为多少，PWM1输出低电平，PWM11输出高电平。

如果PWM1S[1:0] =10，不管PWM占空比为多少，PWM1输出高电平，PWM11输出低电平。

如果PWM1S[1:0] =11，不管PWM占空比为多少，PWM1输出高电平，PWM11输出高电平。

9.4.3 PWM1 占空比寄存器

脉宽控制寄存器

255h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1D	PD1.7	PD1.6	PD1.5	PD1.4	PD1.3	PD1.2	PD1.1	PD1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

PWM输出占空比= [PWM0D] X PWM时钟源

当[PWM1P]≤[PWM1D]，

如果PWM1S[1:0] =00，PWM1输出高电平，PWM11输出高电平

如果PWM1S[1:0] =01，PWM1输出高电平，PWM11输出低电平
如果PWM1S[1:0] =10，PWM1输出低电平，PWM11输出高电平
如果PWM1S[1:0] =11，PWM1输出低电平，PWM11输出低电平

使用注意事项：

1、 选择PWM模块时钟源。
2、 设置PWM周期/占空比，先设置低位，再设置高位。注意，即使高位数值不变，也要重写一次，否则，低位的修改无效。
3、 通过设置PWM控制寄存器(PWM1C)的PWM1Sx位选择PWM输出模式（高电平/低电平有效）。
4、 通过设置PWM控制寄存器(PWMEN)的EPWM1或EPWM01来允许PWM上桥或下桥输出。
5、 如果PWM周期或者占空比需要改变，操作流程如同步骤2和步骤3说明。修改后的重载计数器的值在下一个周期开始有效。
6、 注意不要对PWM关键寄存器进行误操作。

9.5 PWM2 模块

9.5.1 PWM2 控制寄存器

253h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2C	-	-	-	-	PWM2S1	PWM2S0	CK21	CK20
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR的值	-	-	-	-	0	0	0	0

bit [3:2] **PWM2S[1:0]**：PWM2和PWM21占空比输出模式选择位
11 = PWM1和PWM11均为低有效
10 = PWM2为低有效，PWM21为高有效
01 = PWM2为高有效，PWM21为低有效
00 = PWM2和PWM21均为高有效

bit [1:0] **CK2[1:0]**：PWM2输出控制位
11 = Fosc/16
10 = Fosc/8
01 = Fosc/4
00 = Fosc/2
注：Fosc = RC32MHz

9.5.2 PWM2 周期寄存器

周期控制寄存器

252h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2P	PP2.7	PP2.6	PP2.5	PP2.4	PP2.3	PP2.2	PP2.1	PP2.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

PWM2输出周期=[PWM2P] ×PWM时钟源

当[PWM2P]=0x00时：

如果PWM2S[1:0] =00，不管PWM占空比为多少，PWM2输出低电平，PWM21输出低电平。
如果PWM2S[1:0] =01，不管PWM占空比为多少，PWM2输出低电平，PWM21输出高电平。
如果PWM2S[1:0] =10，不管PWM占空比为多少，PWM2输出高电平，PWM21输出低电平。
如果PWM2S[1:0] =11，不管PWM占空比为多少，PWM2输出高电平，PWM21输出高电平。

9.5.3 PWM2 占空比寄存器

脉宽控制寄存器

251h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2D	PD2.7	PD2.6	PD2.5	PD2.4	PD2.3	PD2.2	PD2.1	PD2.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

PWM2输出占空比= [PWM0D] × PWM时钟源

当[PWM2P]≤[PWM2D]，

如果PWM2S[1:0] =00，PWM2输出高电平，PWM21输出高电平
如果PWM2S[1:0] =01，PWM2输出高电平，PWM21输出低电平
如果PWM2S[1:0] =10，PWM2输出低电平，PWM21输出高电平
如果PWM2S[1:0] =11，PWM2输出低电平，PWM21输出低电平

使用注意事项：

1、 选择PWM模块时钟源。
2、 设置PWM周期/占空比，先设置低位，再设置高位。注意，即使高位数值不变，也要重写一次，否则，低位的修改无效。
3、 通过设置PWM控制寄存器(PWM2C)的PWM2Sx位选择PWM输出模式（高电平/低电平有效）。
4、 通过设置PWM控制寄存器(PWMEN)的EPWM2或EPWM21来允许PWM上桥或下桥输出。
5、 如果PWM周期或者占空比需要改变，操作流程如同步骤2和步骤3说明。修改后的重载计数器的值在下一个周期开始有效。
6、 注意不要对PWM关键寄存器进行误操作。

9.6 死区时间

一般的，当没有插入死区时间时，PWM01/11/21 输出波形与 PWM0/1/2 输出波形成固定相位关系。当 PWM 控制寄存器中 EPWM01/11/21 位置 1 时，PWM01/11/21 的输出波形硬件自动产生。

注意：

1、 当 PWM0/1/2 被禁止，如果 PWM01/11/21 被允许，则 PWM01/11/21 仍然会有信号输出。
2、 如果 EFLT 置位，当 FLT 端口有效时，PWM01/11/21 和 PWM0/1/2 都输出固定电平:

PWMxS[1:0] =00，PWMx和PWMx1均输出固定低电平；
PWMxS[1:0] =01，PWMx输出固定低电平，PWMx1输出固定高电平；
PWMxS[1:0] =10，PWMx输出固定高电平，PWMx1输出固定低电平；
PWMxS[1:0] =11，PWMx和PWMx1均输出固定高电平；

其中x=0/1

HC18P23xL PWM 提供死区时间控制功能。通过写 PWM01/11/21 死区时间控制寄存器，在 PWM0/1/2 和 PWM01/11/21 之间产生死区时间。PWM0/1/2 和 PWM01/11/21 的周期相同。

注意：

1、 修改死区时间前，应禁止 PWM 输出。
2、 在 PWMn 和 PWMn1 为互补波形输出时，为了产生死区时间，请确保（PWMn 占空比寄存器值 x 分频系数（2,4,8,16）＞PWMn1 的死区时间（n=0，1,2）控制。否则，PWM01/11/21 当 PWMnS[1:0]=10 时输出高电平，PWMnS[1:0]=01 时输出低电平。
3、 PWMDT 寄存器用于控制死区时间，它的时基为振荡器时钟，而周期和占空比的时基由 TnCKn[1:0]（n=0,1,2）控制，最小为 2 个振荡器时钟。

PWM0 死区时间控制寄存器

258h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DT	DT07	DT06	DT05	DT04	DT03	DT02	DT01	DT00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

PWM1 死区时间控制寄存器

254h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1DT	DT17	DT16	DT15	DT14	DT13	DT12	DT11	DT10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

PWM2 死区时间控制寄存器

250h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2DT	DT27	DT26	DT25	DT24	DT23	DT22	DT21	DT20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

PWMn（n=0,1,2）的死区时间为 PWMnDT× Tosc

注：Tosc = 1/Fosc = 1/32MHz

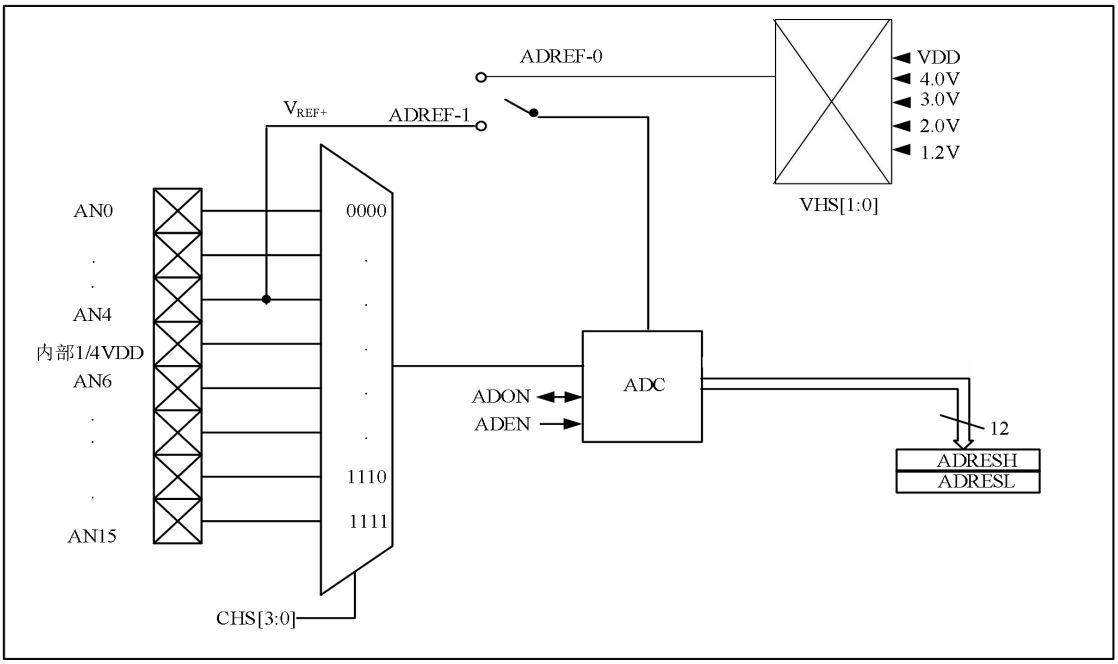
使用说明：当使用 PWM 模块时，系统时钟必须选择为内部 RC 模式。

10 模数转换（ADC)

10.1 ADC 概述

HC18P23xL具有一个12位转换分辨率的模数转换器，共有15个外部模拟输入通道，1个内部电池检测通道。

ADC的等效电路：



10.2 A/D 寄存器

08Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELL	ANSEL7	ANSEL6	-	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR值	1	1	-	1	1	1	1	1

08Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELH	ANSEL15	ANSEL14	ANSEL13	ANSEL12	ANSEL11	ANSEL10	ANSEL9	ANSEL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR值	1	1	1	1	1	1	1	1

ANSEL[15:0]：A/D引脚数模控制位

1：模拟模式，作为模拟信号口，仅可作为AD通道的模拟输入。

0：数字模式，作为数字输入或输出口。

注：1、该寄存器上电初始值为 B'1111 1111'，即作为模拟输入。无论是否应用到 AD，均需要在上电后对 IO 操作之前按需配置，否则 IO 口可能无法受控于对应的端口寄存器，状态将不确定。

2、ANSEL[4:0]对应 AN4~AN0（PA4~PA0），ANSEL[7:6]对应 AN7、AN6（PA7、PA6）；

ANSEL[8]对应 AN8（即 PA5）；

ANSEL[13:9]对应 AN13~AN9（PB4~PB0），ANSEL[15:14]对应 AN15、AN14（PB7、PB6）。

10.3 A/D 控制寄存器

094h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCON0	VHS1	VHS0	CHS3	CHS2	CHS1	CHS0	ADON	ADEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

bit[7:6] **VHS[1:0]**：ADC内部参考电压选择位

00 = 内部2V

01 = 内部3V

10 = 内部4V

11 = VDD

bit[5:2] **CHS[3:0]**：ADC外部通道选择位

CHS[3:0] = x（x=0…….15），表示当前检测通道位ANx，如CHS[3:0] = 0011，表示当前检测通道位为外部通道AN3；外部通道除设置CHS[3:0]外，还需设置相应管脚为模拟输入。

当CHS[3:0] = 0101时，表示当前检测通道位为AN5—内部1/4VDD通道

bit[1] **ADON**：ADC转换启动控制位

0 = 转换结束后，硬件自动清0，在转换过程中，软件清0将终止转换

1 = 启动转换

bit[0] **ADEN**：ADC模块使能位

0 = 关闭ADC模块

1 = 使能ADC模块

注：在使能ADC模块后，建议延时20us在启动ADC转换，以保证ADC模块稳定；

在SLEEP模式下，建议关闭ADC模块，以降低待机功耗。

095h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCON1	ADFM	ADCS2	ADCS1	ADCS0	-	-	-	ADREF
R/W	R/W	W	W	W	-	-	-	R/W
POR的值	0	0	0	0	-	-	-	0

bit[7] **ADFM**：ADC转换数据长度控制位

0 = ADC转换结果为12位数据，数据格式：ADRESH[7:0]:ADRESL[7:4]

1 = ADC转换结果为10位数据，数据格式：ADRESH[1:0]:ADRESL[7:0]

bit[6:4] **ADCS[2:0]**：ADC时钟选择位

000 = Fsys

001 = Fsys/2

010 = Fsys/4

011 = Fsys/8

100 = Fsys/16

101 = Fsys/32

110 = Fsys/64

111 = FRC

bit[3:1] **保留位**

bit[0] **ADREF**：参考电压选择位

0 = 选择内部参考电压，参考电压由VHS[1:0]控制

1 = 选择外部参考电压（外部参考电压输出口），内部参考电压无效

注：

使用外部参考电压(ADREF=1)时，如果 Vref 脚输出或输入高电平会导致功耗偏大。在 sleep 模式下为保证系统的低功耗，请关闭外部参考电压(ADREF=0)。

096h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCLK	-	-	-	-	-	ADCLK2	ADCLK1	ADCLK0
R/W	-	-	-	-	-	R/W	R/W	R/W
POR的值	-	-	-	-	-	0	0	0

bit[2:0] **ADCLK[2:0]**：ADC时钟选择位

000 = Fsys

001 = Fsys/2

010 = Fsys/4

注意：

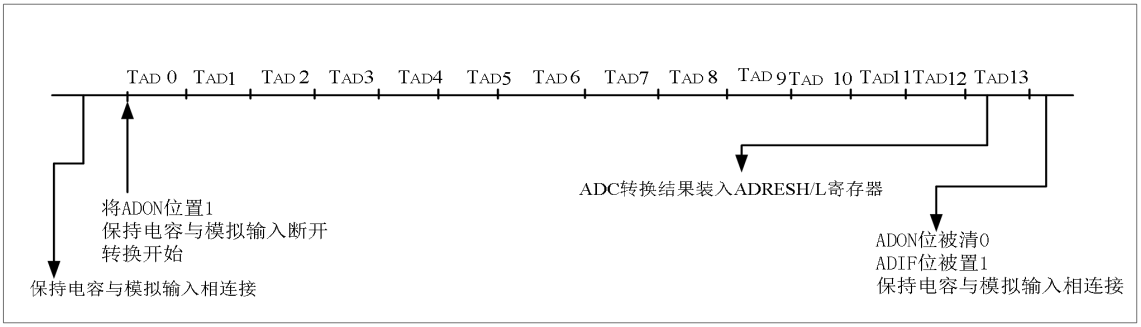
1、AN5为内部1/4VDD输入通道，外部没有输入引脚。可作为电池系统的电池检测器；

2、ADC 所采集数据，当选择存格式为左对齐时，ADC 精度只能为 12 位，高 8 位存放在 ADRESH 寄存器中，低 4 位存放在 ADRESL 寄存器高 4 位上。当选择存放格式为右对齐时，ADC 精度只能为 10 位，高 2 位存放在 ADRESH 的低 2 位上，低 8 位存放在 ADRESL 上。

10.4 AD 转换时间

ADC转换一位数据所需的时间定义为TAD，转换一次完整的12位数据需要14个TAD。为确保ADC正确转换，必须满足适当的TAD时间。

模数转换TAD 周期



ADC转换时间(TAD)与工作频率关系表

ADC 转换时间（TAD）	系统频率（Fsys）:4MHz

ADC 时钟源	ADCS[2:0]	典型值
Fsys	000	4us
Fsys /2	001	8us
Fsys /4	010	16us
Fsys /8	011	32us
Fsys /16	100	64us
Fsys /32	101	128us
Fsys /64	110	256us
FRC	111	视 RC 的值而定

ADCLK软件配置表

ADCLK[2:0]：AD时钟选择位（在确定系统频率后，选择不同的AD时钟分频对应的频率。）

000:AD转换频率在4Mhz及以上

001：AD转换频率为1M

010：AD转换频率为2M

011/1XX：AD转换频率为1M以下

系统频率/AD时钟源选择对应的ADC时钟选择为对应关系如下表所示：

Fsys（系统频率）	ADCS[2:0] AD 分频	ADCLK AD 时钟
32M	011	000
	100	010
	101	001
	11x	011/1 xx
16M	010	000
	011	010
	100	001
	101/11x	011/1 xx
8M	010	010
	011	001
	1xx	011/1 xx
4M	001	010
	010	001
	011/1xx	011/1xx
2M	000	010
	001	001
	01x/1xx	011/1 xx
1M	000	001
	001/0xx/1xx	011/1 xx
500K	xxx	011/1 xx

软件配置ADC转换时钟使用注意说明：

1、 Fsys为系统时钟，Fosc为RC时钟，Fcpu为指令时钟。如用户在OPTION中选择4M/2T，对应Fosc=8M，Fsys=4M，Fcpu=2M。

2、 为保证ADC转换精度，在不同ADC转换时钟下需配置不同的ADC转换时钟个数：

ADC转换时钟	ADCKCS(ADCKCS[2:0])	说明
2 Mhz	010	当ADC转换时钟频率为2Mhz时，需将ADCKCS[2:0]配置成010
1 Mhz	001	当ADC转换时钟频率为1Mhz时，需将ADCKCS[2:0]配置成001
＜1 Mhz	011/1xx	当ADC转换时钟频率小于1 Mhz时，需将ADCKCS[2:0]配置成011/1xx（ADCKCS[2:0]上电默认为111）

3、 为了加快ADC转换速度，建议选用较快的时钟源（ADC转换时钟不能超过4MHz）。

10.5 ADC 使用

1. 配置端口：
   - 设置 TRISA 寄存器禁止引脚输出
   - 设置 ANSEL 寄存器配置引脚为模拟输入
2. 配置ADC模块：
   - 选择 ADC 转换时钟，设置 ADCS[2:0]、ADCKCS[2:0]
   - 选择 ADC 参考电压，设置 ADREF、ADCON0
   - 选择 ADC 输入通道，设置 CHS[3:0]
   - 使能 ADC 模块，设置 ADEN
3. 配置ADC中断（可选）：
   - 清零 ADC 中断标志
   - 使能 ADC 中断
   - 使能外设中断
   - 使能全局中断
4. 等待所需采集时间
5. 设置ADON为1 启动一次ADC转换
6. 通过以下方式之一等待ADC转换完成：
   - 查询 ADON 位
   - 等待 ADC 中断（已使能中断）
7. 读取ADC结果
8. 清零ADC中断标志（如果已使能中断则需要）

➢ 例：配置AD，结果保留在Bank0的NTCADHIGH、NTCADLOW中。

```
                …                                   ;其他程序
AD_TEST:
                BCF     STATUS,RP0          ;Bank0
                BSF     TRISA,0             ;设置AD口为输入
                BCF     STATUS,RP0          ;Bank0
                MOVLW   B'01010000'         ;INNER REF Fsys/32 ADRESH[7:0]
```

```
        ADRESL[7,6]
            MOVWF   ADCON1
            MOVLW   B'00000011'
            MOVWF   ADCLK
                                        ;配置AD通道
            BCF     STATUS,RP0          ;Bank0
            MOVLW   B'00000001'
            MOVWF   ANSELL              ;PA0作为模拟输入
            CLRF    ANSELH
            BCF     STATUS,RP0          ;Bank0
            BCF     ADCON0,CHS2
            BCF     ADCON0,CHS1
            BCF     ADCON0,CHS0
            BCF     ADCON0,VHS1
            BCF     ADCON0,VHS0         ;参考电压为内部VDD
            NOP                         ;延时
            NOP
            BSF     ADCON0,ADEN         ;使能ADC
            CALL    DELAY_1             ;延时，用户可自行完成
            BSF     ADCON0,ADON         ;开始一次转换
AD_TEST_WAIT：
            BTFSC   ADCON0,ADON         ;等待转换完成
            GOTO    AD_TEST_WAIT
                                        ;转换完成，保存结果
            MOVF    ADRESH,W            ;LOAD THE AD HIGH 8 BITS TO W
            MOVWF   NTCADHIGH           ;客户应用时注意BANK
            MOVF    ADRESL,W            ;LOAD THE AD LOW 8 BITS TO W
            MOVWF   NTCADLOW
```

注意：

1、使能ADEN后（不是使能ADON），系统必须延迟一定的时间（视外部输入信号而定）等待ADC电路稳定；

2、睡眠或绿色模式下，禁止ADC并将AD参考电压设为非内部VDD以降低功耗；

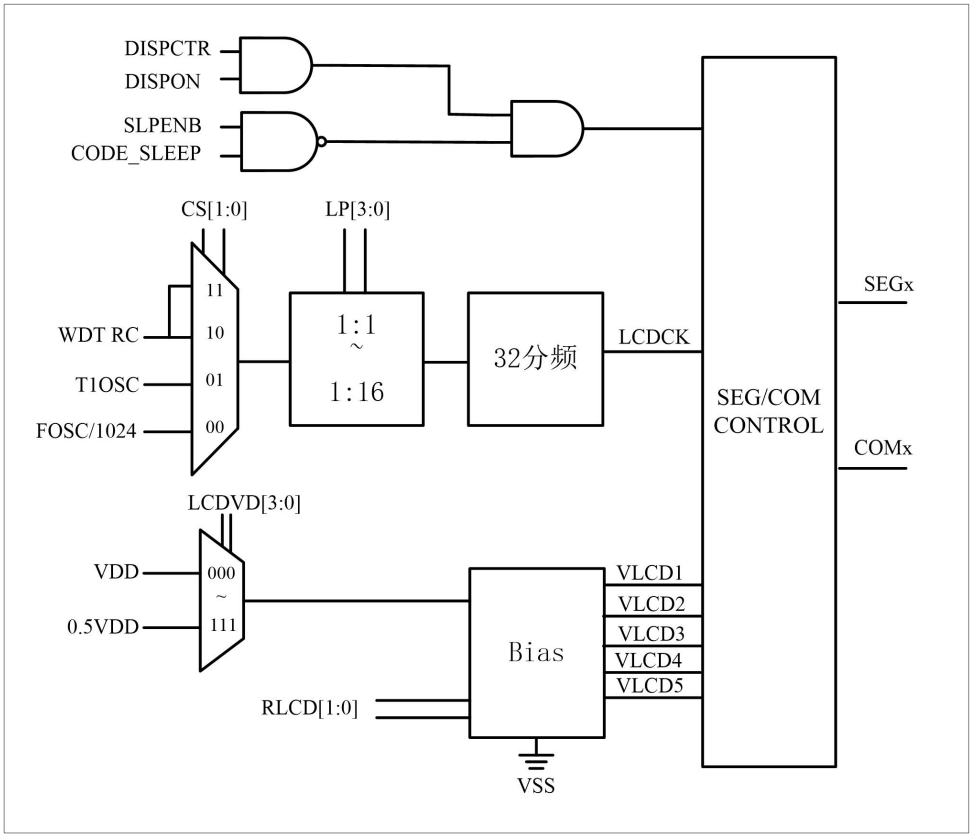
3、为保证ADC转换精度，芯片VDD电压应高于所选ADC内部参考电压（4V/3V/2V）0.7V以上。

11 液晶显示驱动(LCD)

11.1 概述

HC18P23xL具有一个8×32的液晶驱动模块，而且所有的LCD驱动口都可以用作输入输出口。

在SLEEP模式下默认LCD显示，如需在SLEEP模式下关闭LCD，则进入休眠前关闭LCD模块（DISPON置0），退出休眠后再打开LCD模块（DISPON置1）。



注：

1. 帧频计算：$F=F_{LCD}\times N/(L\times 32)$；（L为LP分频，N为1/4duty或1/8duty）；

2. 1/4Duty帧频范围（16Hz~250Hz）；1/8Duty帧频范围（8~125Hz）；

3. 在选择1/4duty时，建议使用1:4分频，在1/8duty时，使用1:2分频。

11.2 LCD 相关寄存器

11.2.1 LCD 选择寄存器

2B2h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DISPCTR	-	-	-	-	-	SLPENB	DISPON	DISPCT
R/W	-	-	-	-	-	R/W	R/W	R/W
POR的值	-	-	-	-	-	0	0	0

bit [2] **SLPENB**: SLEEP下LCD使能位

0 = 使能，在SLEEP模式下，LCD正常使用

1 = 禁止，在SLEEP模式下，LCD关闭

bit [1] **DISPON**：LCD显示使能位

0 = 禁止LCD/LED驱动

1 = 使能LCD/LED驱动

Bit[2] **DISPCT**：显示选择位

0 = 选择LCD驱动模块，LED驱动模块禁止

1 = 选择LED驱动模块，LCD驱动模块禁止

2B0h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCDCON	BIASCT1	BIASCT0	DUTCT1	DUTCT0	CS1	CS0	RLCD1	RLCD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

bit [7:6] **BIASCT[1:0]:** LCD偏置控制位

11= 1/4 bias

10 = 1/3 bias

01 = 1/2bias（1/4Duty）

00 = 保留

注：在选择01 = 1/2bias时只能驱动TN型LCD Pannel。

bit [5] **DUTCT[1：0]:** LCD占空比控制位

00 = 1/8duty（COM0~COM7）

01 = 1/5duty（COM0~COM4）（COM5-COM7作普通IO）

10 = 1/4duty（COM0~COM3）（COM4-COM7作普通IO）

11 = 1/3duty（COM0~COM2）（COM4-COM7作普通IO，COM3为NC不能用作普通IO）

当DUTCT = 0时，COM4~COM7作为LCD COM口；当DUTCT = 1时，COM4~COM7为普通IO。

bit [3:2] **CS[1:0]**：时钟源选择位

00 = FOSC/1024

01 = T1OSC

1x = WDT RC（在SLEEP模式下LCD显示，这时即使配置字关闭WDT，WDT也会强制打开）

bit [1:0] **RLCD[1:0]**：分压电阻选择位

00: R1 = R2 = R3 = 270kΩ

01: R1 = R2 = R3 = 90kΩ

10: R1 = R2 = R3 = 30kΩ

11: R1 = R2 = R3 = 2.5kΩ

注：请根据LCD Pannel尺寸选择合适的偏置电阻，避免由于驱动不够造成显示效果不佳的现象。Pannel较小时可选用270K，较大时可选用2.5K。

11.2.2 LCD 对比度控制位

2B3h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCDLVD	-	RSEL	SPDSEL1	SPDSEL0	LCDVD3	LCDVD2	LCDVD1	LCDVD0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	0	0	0	0	0	0	0

bit [3:0] **LCDVD:**对比度控制位

1xxx = 50% VDD（**8/16VDD**）

0111 = 56.25% VDD（**9/16VDD**）

0110 =62.5%VDD（**10/16VDD**）

0101 = 68.75% VDD（**11/16VDD**）

0100 = 75% VDD（**12/16VDD**）

0011 = 81.25% VDD（**13/16VDD**）

0010 = 87.5% VDD（**14/16VDD**）

0001 = 93.75% VDD（**15/16VDD**）

0000 = VDD（**16/16VDD**）

bit [5:4] **SPDSEL[1:0]:**加速时间选择位

00 = 1/64* F_{LCD}

01 = 2/64* F_{LCD}

10 = 4/64* F_{LCD}

11 = 8/64* F_{LCD}

bit [6] **RSEL:**加速电阻选择位

0 = 42.5KΩ

1 = 2.5KΩ

11.2.3 LCD 预分频寄存器

LCD预分频时钟选择寄存器

2B1h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCDPS	-	-	-	-	LP3	LP2	LP1	LP0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR的值	-	-	-	-	0	0	0	0

bit [3:0] **LP[3:0]**：LCD预分频比选择位

1111 = 1:16
1110 = 1:15
1101 = 1:14
1100 = 1:13
1011 = 1:12
1010 = 1:11
1001 = 1:10
1000 = 1:9
0111 = 1:8
0110 = 1:7
0101 = 1:6
0100 = 1:5
0011 = 1:4
0010 = 1:3
0001 = 1:2
0000 = 1:1

11.2.4 LCD SEG 输出控制寄存器

LCD SEG驱动口输出控制寄存器

2B6h ~2B9h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SEGSE0	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0
SEGSE1	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8
SEGSE2	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16
SEGSE3	SE31	SE30	SE29	SE28	SE27	SE26	SE25	SE24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

bit [7:0] **SEx**：SEGx的LCD使能位

1 = 使能引脚的SEGx功能

0 = 使能引脚的I/O功能

11.2.5 LCD RAM

LCD显示像素RAM映射地址

2C0h ~2DFh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SEGDATA0	**SEG0/C7**	**SEG0/C6**	**SEG0/C5**	**SEG0/C4**	**SEG0/C3**	**SEG0/C2**	**SEG0/C1**	**SEG0/C0**
SEGDATA1	**SEG1/C7**	**SEG1/C6**	**SEG1/C5**	**SEG1/C4**	**SEG1/C3**	**SEG1/C2**	**SEG1/C1**	**SEG1/C0**
SEGDATA2	**SEG2/C7**	**SEG2/C6**	**SEG2/C5**	**SEG2/C4**	**SEG2/C3**	**SEG2/C2**	**SEG2/C1**	**SEG2/C0**
SEGDATA3	**SEG3/C7**	**SEG3/C6**	**SEG3/C5**	**SEG3/C4**	**SEG3/C3**	**SEG3/C2**	**SEG3/C1**	**SEG3/C0**
SEGDATA4	**SEG4/C7**	**SEG4/C6**	**SEG4/C5**	**SEG4/C4**	**SEG4/C3**	**SEG4/C2**	**SEG4/C1**	**SEG4/C0**
SEGDATA5	**SEG5/C7**	**SEG5/C6**	**SEG5/C5**	**SEG5/C4**	**SEG5/C3**	**SEG5/C2**	**SEG5/C1**	**SEG5/C0**
SEGDATA6	**SEG6/C7**	**SEG6/C6**	**SEG6/C5**	**SEG6/C4**	**SEG6/C3**	**SEG6/C2**	**SEG6/C1**	**SEG6/C0**
SEGDATA7	**SEG7/C7**	**SEG7/C6**	**SEG7/C5**	**SEG7/C4**	**SEG7/C3**	**SEG7/C2**	**SEG7/C1**	**SEG7/C0**
SEGDATA8	**SEG8/C7**	**SEG8/C6**	**SEG8/C5**	**SEG8/C4**	**SEG8/C3**	**SEG8/C2**	**SEG8/C1**	**SEG8/C0**
SEGDATA9	**SEG9/C7**	**SEG9/C6**	**SEG9/C5**	**SEG9/C4**	**SEG9/C3**	**SEG9/C2**	**SEG9/C1**	**SEG9/C0**
SEGDATA10	**SEG10/C7**	**SEG10/C6**	**SEG10/C5**	**SEG10/C4**	**SEG10/C3**	**SEG10/C2**	**SEG10/C1**	**SEG10/C0**
SEGDATA11	**SEG11/C7**	**SEG11/C6**	**SEG11/C5**	**SEG11/C4**	**SEG11/C3**	**SEG11/C2**	**SEG11/C1**	**SEG11/C0**
SEGDATA12	**SEG12/C7**	**SEG12/C6**	**SEG12/C5**	**SEG12/C4**	**SEG12/C3**	**SEG12/C2**	**SEG12/C1**	**SEG12/C0**
SEGDATA13	**SEG13/C7**	**SEG13/C6**	**SEG13/C5**	**SEG13/C4**	**SEG13/C3**	**SEG13/C2**	**SEG13/C1**	**SEG13/C0**
SEGDATA14	**SEG14/C7**	**SEG14/C6**	**SEG14/C5**	**SEG14/C4**	**SEG14/C3**	**SEG14/C2**	**SEG14/C1**	**SEG14/C0**
SEGDATA15	**SEG15/C7**	**SEG15/C6**	**SEG15/C5**	**SEG15/C4**	**SEG15/C3**	**SEG15/C2**	**SEG15/C1**	**SEG15/C0**
SEGDATA16	**SEG16/C7**	**SEG16/C6**	**SEG16/C5**	**SEG16/C4**	**SEG16/C3**	**SEG16/C2**	**SEG16/C1**	**SEG16/C0**
SEGDATA17	**SEG17/C7**	**SEG17/C6**	**SEG17/C5**	**SEG17/C4**	**SEG17/C3**	**SEG17/C2**	**SEG17/C1**	**SEG17/C0**
SEGDATA18	**SEG18/C7**	**SEG18/C6**	**SEG18/C5**	**SEG18/C4**	**SEG18/C3**	**SEG18/C2**	**SEG18/C1**	**SEG18/C0**
SEGDATA19	**SEG19/C7**	**SEG19/C6**	**SEG19/C5**	**SEG19/C4**	**SEG19/C3**	**SEG19/C2**	**SEG19/C1**	**SEG19/C0**
SEGDATA20	**SEG20/C7**	**SEG20/C6**	**SEG20/C5**	**SEG20/C4**	**SEG20/C3**	**SEG20/C2**	**SEG20/C1**	**SEG20/C0**
SEGDATA21	**SEG21/C7**	**SEG21/C6**	**SEG21/C5**	**SEG21/C4**	**SEG21/C3**	**SEG21/C2**	**SEG21/C1**	**SEG21/C0**
SEGDATA22	**SEG22/C7**	**SEG22/C6**	**SEG22/C5**	**SEG22/C4**	**SEG22/C3**	**SEG22/C2**	**SEG22/C1**	**SEG22/C0**
SEGDATA23	**SEG23/C7**	**SEG23/C6**	**SEG23/C5**	**SEG23/C4**	**SEG23/C3**	**SEG23/C2**	**SEG23/C1**	**SEG23/C0**
SEGDATA24	**SEG24/C7**	**SEG24/C6**	**SEG24/C5**	**SEG24/C4**	**SEG24/C3**	**SEG24/C2**	**SEG24/C1**	**SEG24/C0**
SEGDATA25	**SEG25/C7**	**SEG25/C6**	**SEG25/C5**	**SEG25/C6**	**SEG25/C3**	**SEG25/C2**	**SEG25/C1**	**SEG25/C0**
SEGDATA26	**SEG26/C7**	**SEG26/C6**	**SEG26/C5**	**SEG26/C4**	**SEG26/C3**	**SEG26/C2**	**SEG26/C1**	**SEG26/C0**
SEGDATA27	**SEG27/C7**	**SEG27/C6**	**SEG27/C5**	**SEG27/C4**	**SEG27/C3**	**SEG27/C2**	**SEG27/C1**	**SEG27/C0**
SEGDATA28	**SEG28/C7**	**SEG28/C6**	**SEG28/C5**	**SEG28/C4**	**SEG28/C3**	**SEG28/C2**	**SEG28/C1**	**SEG28/C0**
SEGDATA29	**SEG29/C7**	**SEG29/C6**	**SEG29/C5**	**SEG29/C4**	**SEG29/C3**	**SEG29/C2**	**SEG29/C1**	**SEG29/C0**
SEGDATA30	**SEG30/C7**	**SEG30/C6**	**SEG30/C5**	**SEG30/C4**	**SEG30/C3**	**SEG30/C2**	**SEG30/C1**	**SEG30/C0**
SEGDATA31	**SEG31/C7**	**SEG31/C6**	**SEG31/C5**	**SEG31/C4**	**SEG31/C3**	**SEG31/C2**	**SEG31/C1**	**SEG31/C0**
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	X	X	X	X	X	X	X	X

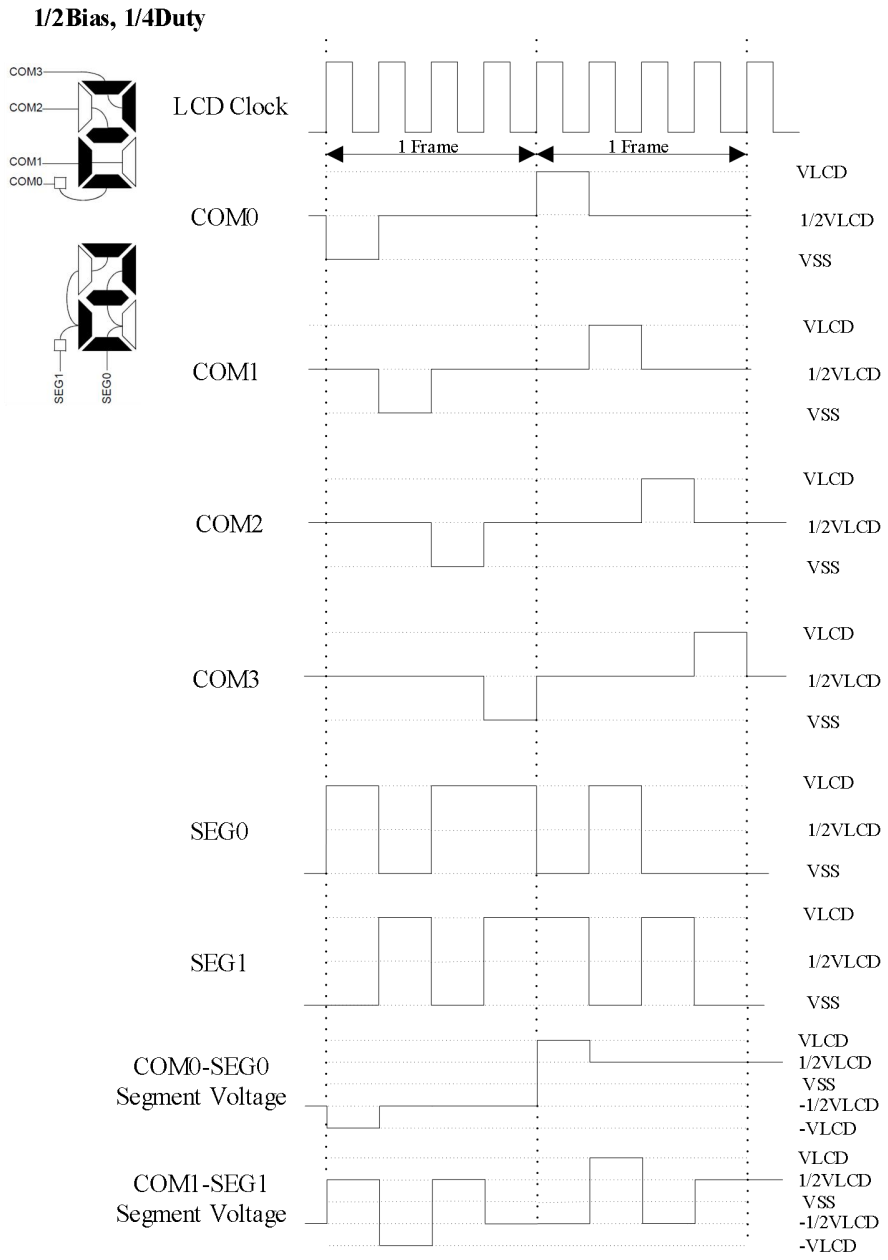
bit [7:0] **SEGx/Cy**：SEGx、COMy像素点亮控制位

0 = 点亮像素（不透明）

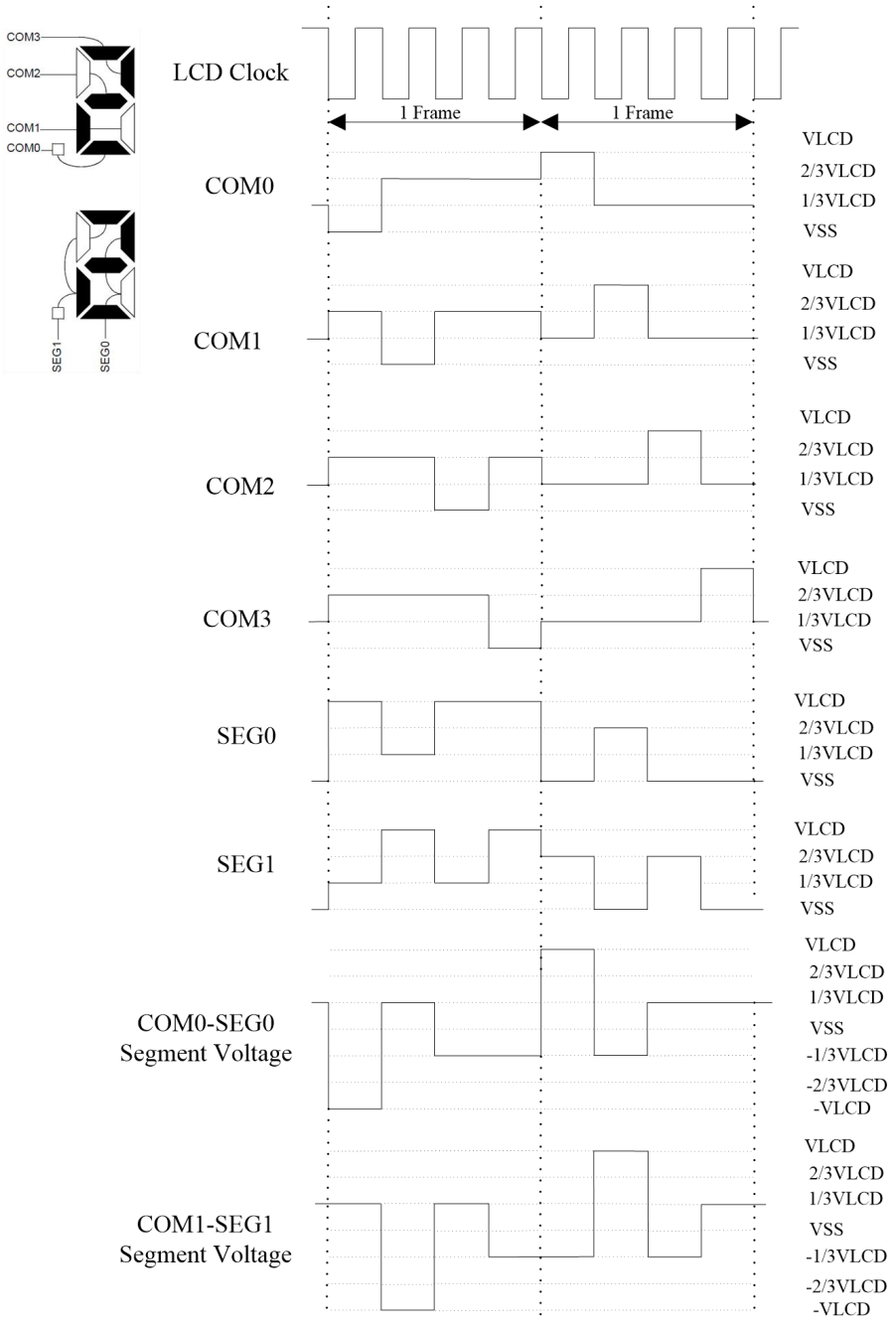
1 = 不点亮象素（透明）

使用说明：HC18P23xL亮度控制为0有效，即0点亮像素，1不点亮，在应用中请注意。

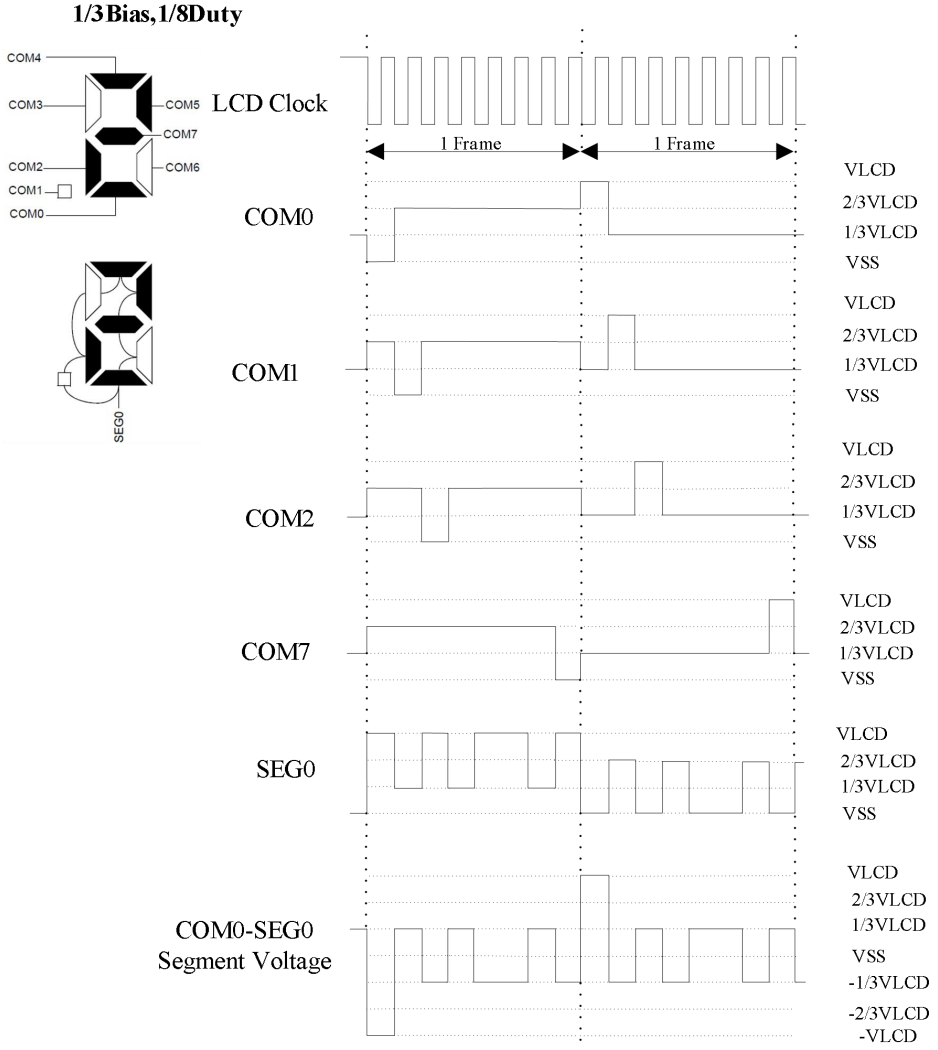
11.3 LCD 波形时序图



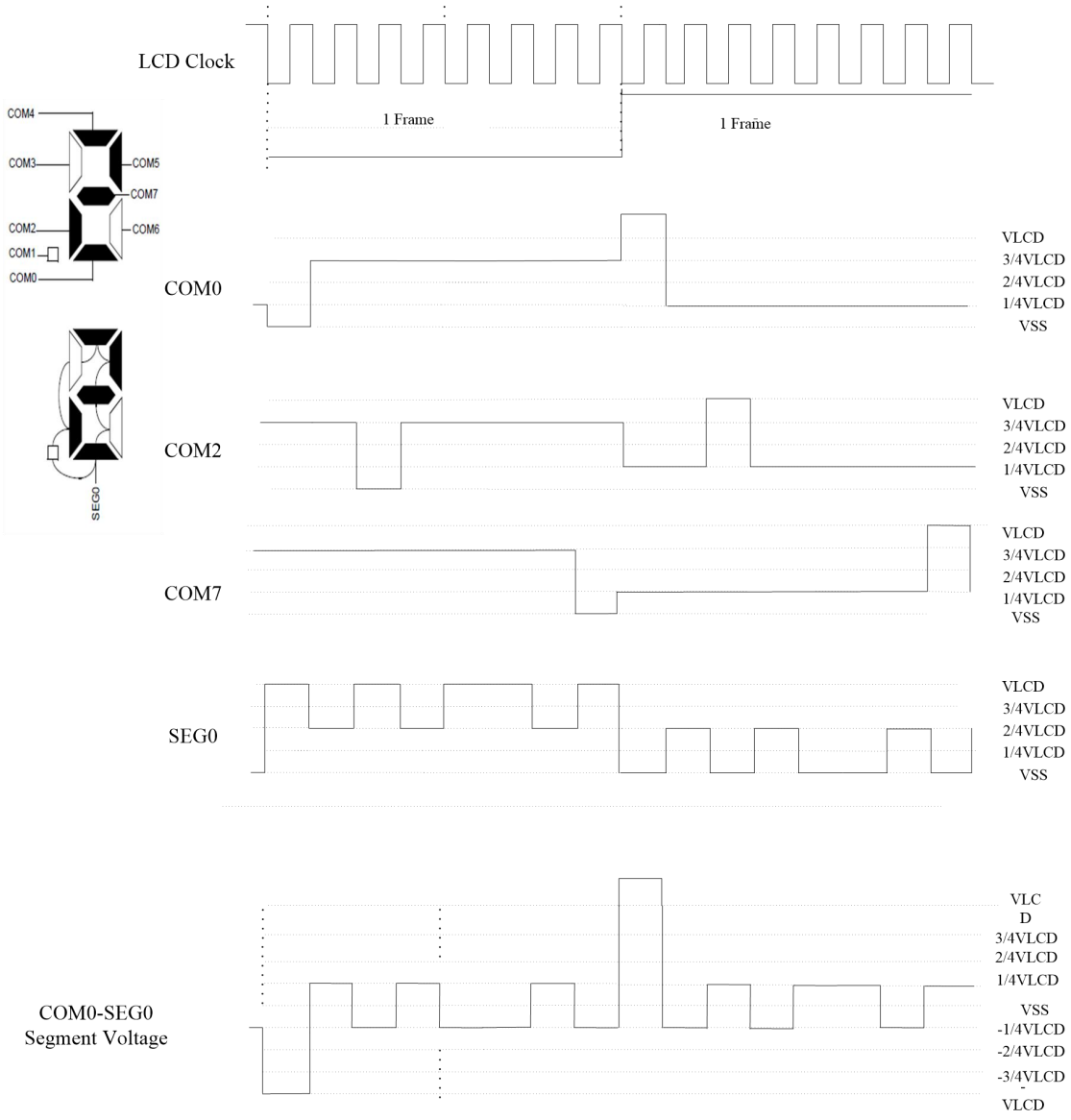
1/3Bias, 1/4Duty



1/3 Bias, 1/8 Duty



1/4Bais, 1/8Duty



12 串行口通信

12.1 概述

HC18P23xL 具有 1 个采用 UART(Universal Asychronous Receiver/Transmitter)工作方式的全双工串行通信接口。串行口由 2 个数据缓冲器、一个移位寄存器、一个串行控制寄存器和一个波特率发生器等组成。串行口的数据缓冲器由 2 个互相独立的接收、发送缓冲器构成，可以同时发送和接收数据。发送缓冲器只能写入而不能读出，接收缓冲器只能读出而不能写入，因而两个缓冲器可以共用一个地址码。串行口的两个缓冲器共用的地址码是 23Dh。两个缓冲器统称串行通信特殊功能寄存器 SBUF。

HC18P23xL的串行口都有4种工作方式，其中两种方式的波特率是可变的，另两种是固定的，以供不同应用场合选用。用户可用软件设置不同的波特率和选择不同的工作方式。主机可通过查询或中断方式对接收/发送进行程序处理，使用十分灵活。

串行口对应的硬件部分是TxD和RxD引脚。

12.2 UART 相关寄存器

12.2.1 串行口的控制寄存器

23Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCON	SM0/FE	SM1	SM2	REN	TB8	RB8	RXWK	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
POR的值	0	0	0	0	0	x	0	-

bit [7:6] **SM0/SM1**：串口工作方式选择位

SM0	SM1	工作方式	功能说明	波特率
0	0	方式0	同步移位串行方式：移位寄存器	当UX6 = 0时，波特率是F_{CPU} /12 当UX6 = 1时，波特率是F_{CPU} /2
0	1	方式1	8位UART，波特率可变	BRT独立波特率发生器的溢出率/16
1	0	方式2	9位UART	$(2^{SMOD}/64) \times F_{CPU}$
1	1	方式3	9位UART，波特率可变	BRT独立波特率发生器的溢出率/16

bit [7] **FE**：帧错误检测位

1 =有侦错误，硬件置1

0 =无侦错误或软件清零

bit [5] **SM2**：多机通信使能控制位(第九位“1”校验器)

1 =在方式 1 下，允许停止位确认检验，只有有效的停止位“1”才能置位 RXIF

在方式2和3下，只有地址字节（第9位=“1”）才能置位RXIF

0 =在方式 1 下，禁止停止位确认检验，任何停止位都会置位 RXIF

在方式 2 和 3 下，任何字节都会置位 RXIF

bit [4] **REN**：串行接收使能控制位

1 =允许串行接收

0 =禁止串行接收

bit [3] **TB8**：方式2/方式3时，为要发送的第9位数据，由软件置1或清零

bit [2] **RB8**：方式2/方式3时，为要发送的第9位数据，作为奇偶校验位或者地址帧/数据帧的标志位

bit [1] **RXWK**：接收中断唤醒使能位

1 =允许RXD下降沿置RXIF，允许RXD唤醒SLEEP；

0 =允许RXD下降沿置RXIF，禁止RXD唤醒SLEEP；

12.2.2 串行口数据缓冲寄存器

23Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SBUF								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

bit [7:0] **SBUF[7:0]** 串口缓冲寄存器，写为需要发送的数据，读为接收到的数据

串行口1缓冲寄存器(SBUF)的地址是23Dh，实际是2个缓冲器，写SBUF的操作完成待发送数据的加载，读SBUF的操作可获得已接收到的数据。两个操作分别对应两个不同的寄存器，1个是只写寄存器，1个是只读寄存器。

串行通道内设有数据寄存器。在所有的串行通信方式中，在写入SBUF信号的控制下，把数据装入相同的9位移位寄存器，前面8位为数据字节，其最低位为移位寄存器的输出位。根据不同的工作方式会自动将“ 1” 或TB8的值装入移位寄存器的第9位,并进行发送。

串行通道的接收寄存器是一个输入移位寄存器。在方式0时它的字长为8位，其他方式时为9位。当一帧接收完毕，移位寄存器中的数据字节装入串行数据缓冲器SBUF中，其第9位则装入SCON寄存器中的RB8位。如果由于SM2使得已接收到的数据无效时，RB8和SBUF中内容不变。

由于接收通道内设有输入移位寄存器和SBUF缓冲器，从而能使一帧接收完将数据由移位寄存器装入SBUF后，可立即开始接收下一帧信息，主机应在该帧接收结束前从SBUF缓冲器中将数据取走，否则前一帧数据将丢失。SBUF以并行方式送往内部数据总线。

12.2.3 辅助寄存器

23Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AUXR	-	UARTEN	UARTM0	BRTR	BRTX12	S1BRS	SMOD	SMOD0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	0	0	0	0	x	0	0

bit [6] **UARTEN**：UART管脚控制位

1 =使能UART管脚功能

0 =禁止UART管脚功能，做普通IO口

bit [5] **UARTM0**：串口模式0的通信速度设置位

1 = UART串口模式0 2分频

0 = UART 串口模式 0 12 分频

bit [4] **BRTR**：独立波特率发生器运行控制位

1 =允许独立波特率发生器运行

0 =不允许独立波特率发生器运行

bit [3] **BRTX12**：独立波特率发生器计数控制位

1 =独立波特率发生器每1个时钟计数一次

0 =独立波特率发生器每 12 个时钟计数一次

bit [2] **S1BRS**：串口（UART）的波特率发生器控制位

1 =选择独立波特率发生器作为串口（UART）波特率发生器

0 =未用

bit [1] **SMOD**：波特率选择位

1 =串行通信方式1、2、3的波特率加倍

0 =串行通信方式1、2、3的波特率不加倍

bit [0] **SMOD0**：帧错误检测有效控制位

1 = SCON寄存器中的 SM0/FE应用于FE（帧错误检测）功能

0 = SCON寄存器中的 SM0/FE应用于SM0功能，和SM1一起指定串行口的工作方式

12.2.4 独立波特率发生器寄存器

23Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BRT								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	x	0	0

独立波特率发生器寄存器BRT用于保存重装时间常数。

12.2.5 自动地址识别寄存器

23Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SADEN								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	x	0	0

Bit[7:0] SADEN[7:0]：从机地址寄存器

23Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SADDR								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	x	0	0

bit[7:0] **SADDR[7:0]**：从机地址掩码寄存器

12.3 工作方式

UART有4种工作方式，在四种方式中，任何将SBUF作为目标寄存器的写操作都会启动发送。在方

式0中由条件RXIF= 0和REN = 1初始化接收。这会在TXD引脚上产生一个时钟信号，然后在RXD引脚上移出8位数据。在其它方式中由输入的起始位初始化接收（如果RXIF = 0和REN = 1）。外部发送器通信以发送起始位开始。在发送之前TXD引脚必须被设置为输出高电平。

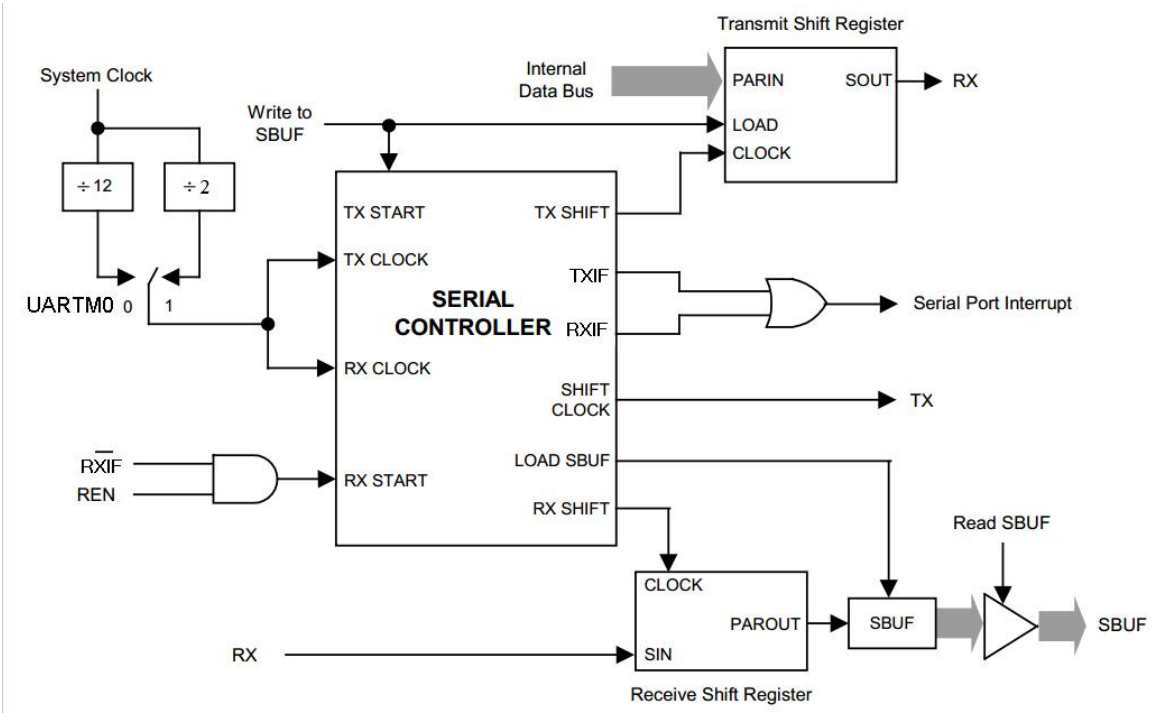
SM0	SM1	工作方式	类型	波特率
0	0	方式0	同步	波特率是F_{CPU} /12×6^{UX6}
0	1	方式1	异步	BRT独立波特率发生器的溢出率/16
1	0	方式2	异步	(2^{SMOD} /64) ×F_{CPU}
1	1	方式3	异步	BRT独立波特率发生器的溢出率/16

12.3.1 方式 0：同步半双工通讯

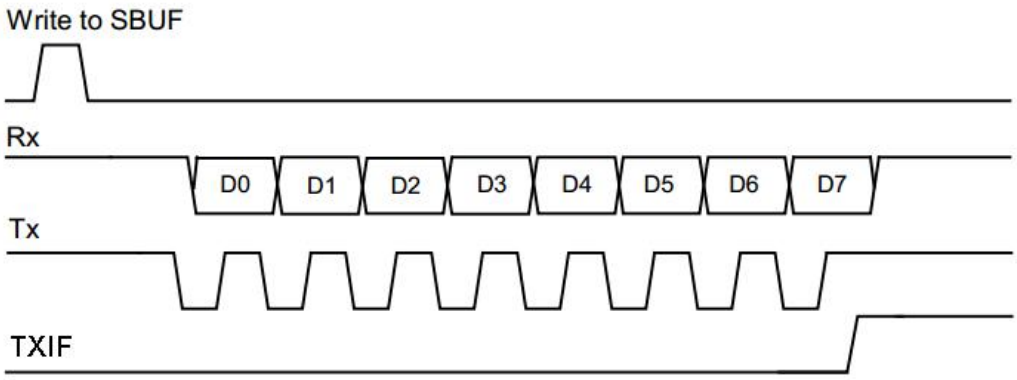
方式0支持与外部设备的同步通信，在RX引脚上收发串行数据，TX引脚发送移位时钟。HC18P23xL提供TX引脚上的移位时钟，因此这种方式是串行通信的半双工方式。在这个方式中，每帧收发8位，低位先接收或发送。

通过置UARTM0位为0或1，波特率固定为F_{CPU}的1/12或1/2。当UARTM0位等于0时，串行端口以F_{CPU}的1/12运行，当UARTM0位等于1时，串行端口以F_{CPU}的1/2运行。

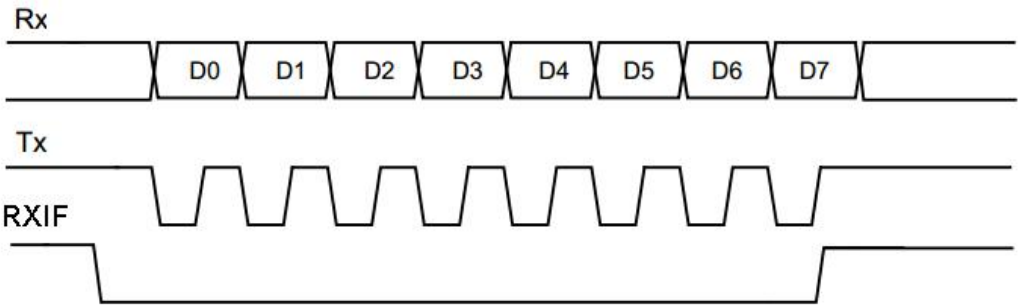
功能块框图如下图所示，数据通过 RX 引脚移入和移出串行端口，移位时钟由 TX 引脚输出。



任何将 SBUF 作为目标寄存器的写操作都会启动发送。下一个系统时钟 TX 控制块开始发送。数据转换发生在移位时钟的下降沿，移位寄存器的内容逐次从左往右移位，空位置 0。当移位寄存器中的所有 8 位都发送后，TX 控制模块停止发送操作，然后在下一个系统时钟的上升沿将 TXIF 位置 1。

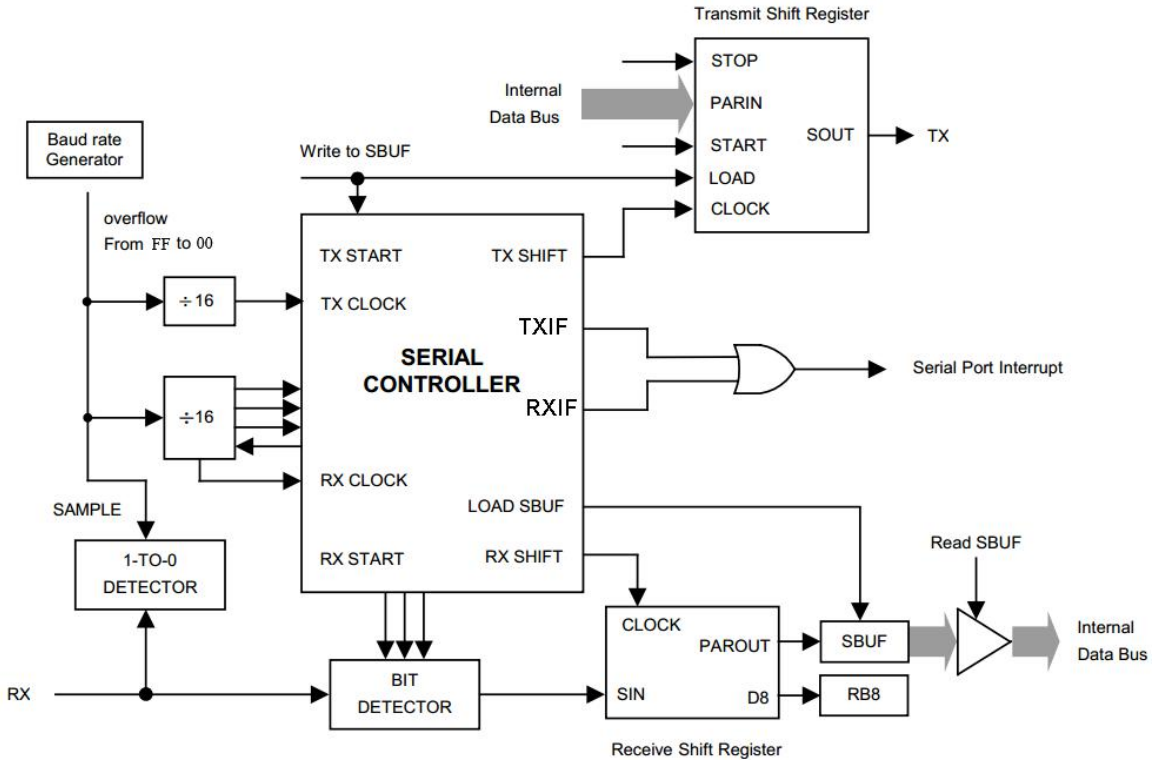


REN 位置 1 和 RXIF 位清零初始化接收。下一个系统时钟启动接收，在移位时钟的上升沿锁存数据，接收转换寄存器的内容逐次向左移位。当所有 8 位数据都移到移位寄存器中后，RX 控制块停止接收，在下一个系统时钟的上升沿 RXIF 置位，直到被软件清零才允许下一次接收。

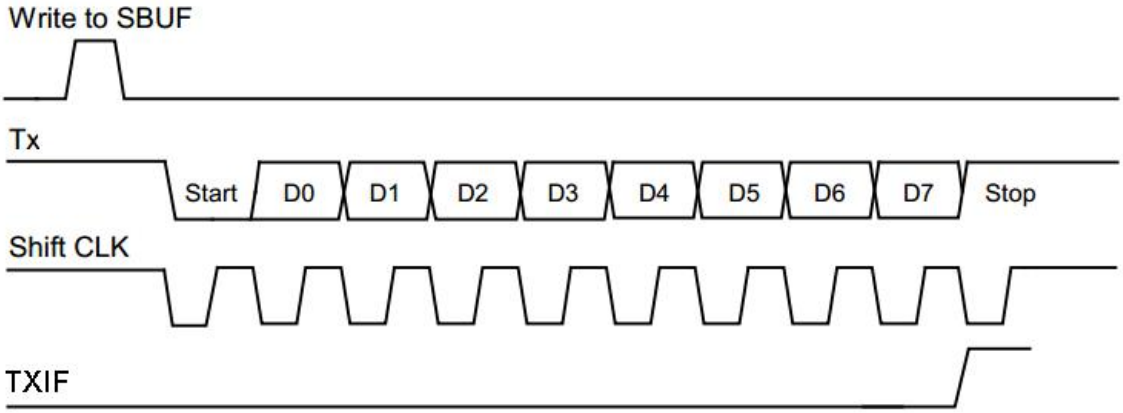


12.3.2 方式 1：8 位 UART，可变波特率，异步全双工

方式 1 提供 10 位全双工异步通信，10 位由一个起始位（逻辑 0），8 个数据位（低位在前）和一个停止位（逻辑 1）组成。在接收时，这 8 个数据位存储在 SBUF 中而停止位储存在 RB8 中。方式 1 中的波特率固定为自带波特率发生器溢出率的 1/16。功能块框图如下图所示。



任何将 SBUF 作为目标寄存器的写操作都会启动发送，实际上发送是从 16 分频计数器中的下一次跳变之后的系统时钟开始的，因此位时间与 16 分频计数器是同步的，与对 SBUF 的写操作不同步。起始位首先在 TX 引脚上移出，然后是 8 位数据位。在发送移位寄存器中的所有 8 位数据都发送完后，停止位在 TX 引脚上移出，在停止位发出的同时 TXIF 标志置位。

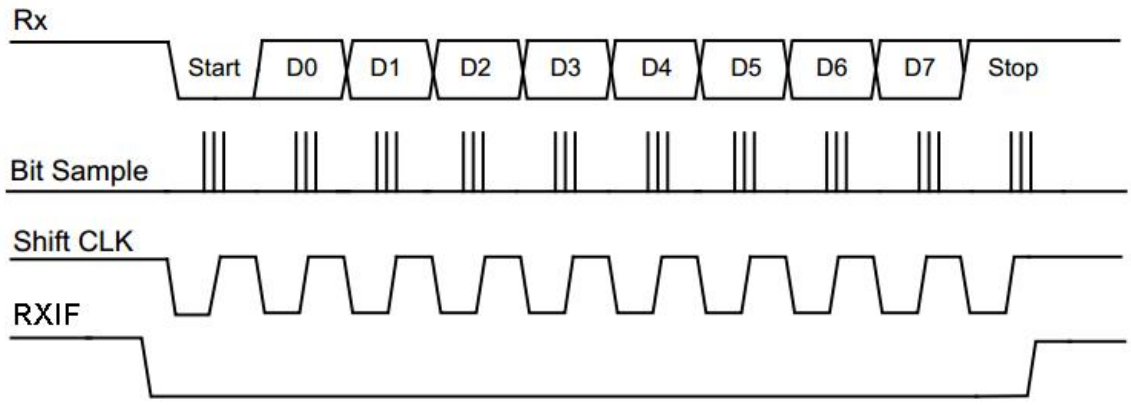


只有REN置1时才允许接收。当RX引脚检测到下降沿时串行口开始接收串行数据。为此，CPU对RX不断采样，采样速率为波特率的16倍。当检测下降沿时，16分频计数器立即复位，这有助于16分频计数器与RX引脚上的串行数据位同步。16分频计数器把每一位的时间分为16个状态，在第7、8、9状态时，位检测器对RX端的电平进行采样。为抑制噪声，在这3个状态采样中至少有2次采样值一致数据才被接收。如果所接收的第一位不是0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待RX引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。8个数据位和1个停止位（包含错误的停止位，详细见寄存器SM2位说明）移入之后，移位寄存器的内容和停止位(包含错误的停止位)被分别装入SBUF和RB8中，RXIF置1，但必须满足下列条件：

(1) RXIF = 0

(2) SM2 = 0或者接收的停止位= 1

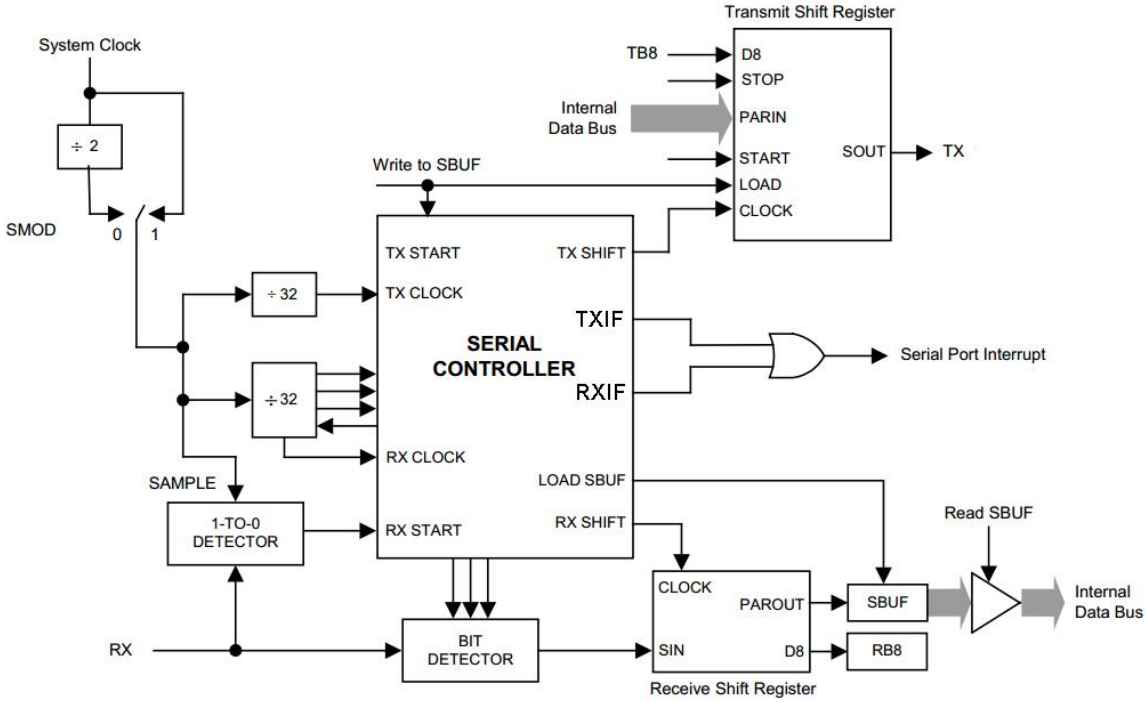
如果这些条件被满足，那么停止位（包含错误的停止位）装入 RB8，8 个数据位装入 SBUF，RXIF 被置位。否则接收的帧会丢失。这时，接收器将重新去探测 RX 端是否另一个下降沿。用户必须用软件清零 RXIF，然后才能再次接收。



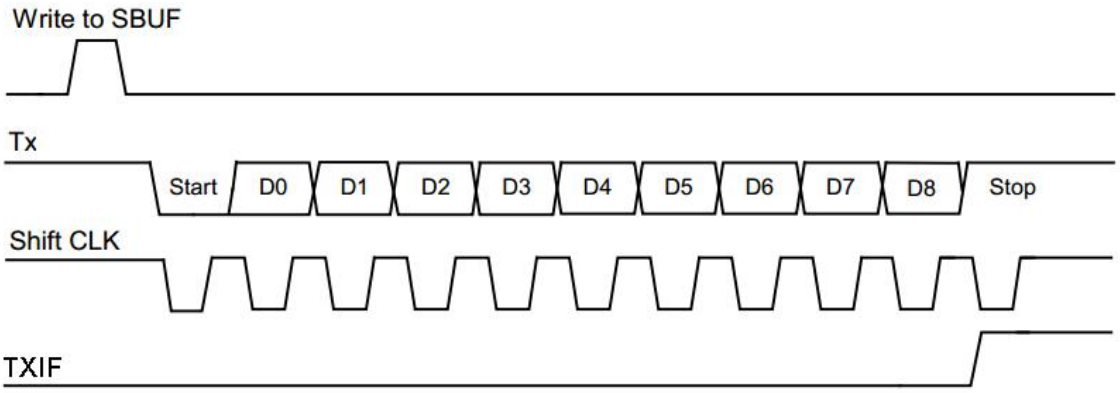
12.3.3 方式 2：9 位 UART，固定波特率，异步全双工

这个方式使用异步全双工通信中的 11 位。一帧由一个起始位（逻辑 0），8 个数据位（低位在前），一个可编程的第 9 数据位和一个停止位（逻辑 1）组成。方式 2 支持多机通信和硬件地址识别（详见多机通讯章节）。在数据传送时，第 9 数据位（TB8 位）可以写 0 或 1。当接收到数据时，第 9 数据位移

入 RB8 而停止位不保存。SMOD 位选择波特率为系统工作频率的 1/32 或 1/64。功能块框图如下所示。



任何将 SBUF 作为目标寄存器的写操作都会启动发送，同时也将 TB8 载入到发送移位寄存器的第 9 位中。实际上发送是从 16 分频计数器中的下一次跳变之后的系统时钟开始的，因此位时间与 16 分频计数器是同步的，与对 SBUF 的写操作不同步。起始位首先在 TX 引脚上移出，然后是 9 位数据。在发送转换寄存器中的所有 9 位数据都发送完后，停止位在 TX 引脚上移出，在停止位开始发送时 TXIF 标志置位。



只有REN置位时才允许接收。当RX引脚检测到下降沿时串行口开始接收串行数据。为此，CPU对RX不断采样，采样速率为波特率的16倍。当检测下降沿时，16分频计数器立即复位。这有助于16分频计数器与RX引脚上的串行数据位同步。16分频计数器把每一位的时间分为16个状态，在第7、8、9状态时，位检测器对RX端的电平进行采样。为抑制噪声，在这3个状态采样中至少有2次采样值一致数据才被接收。如果所接收的第一位不是0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待RX引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。9个数据位和1个停止位移入之后，移位寄存器的内容被分别装入SBUF和RB8中，RXIF置1，但必须满足下列条件：

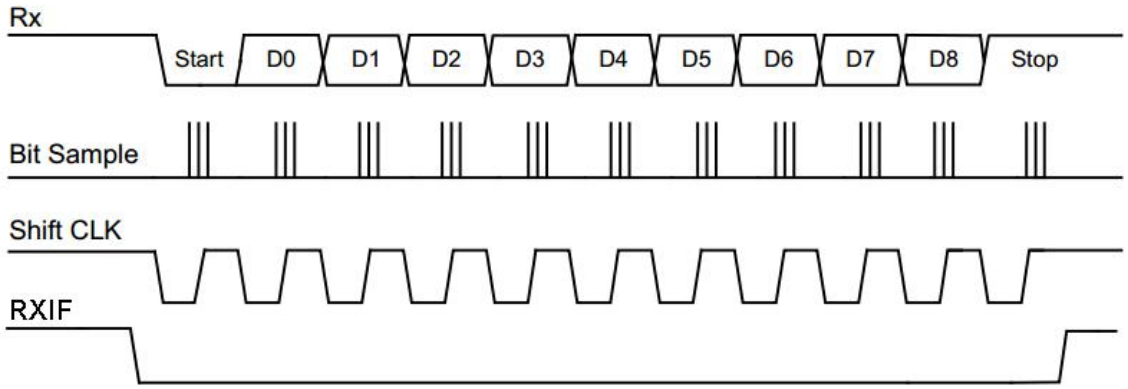
(1) RXIF = 0

(2) SM2 = 0或者接收的第9位= 1，且接收的字节符合约定从机地址

如果这些条件被满足，那么第9位移入RB8，8位数据移入SBUF，RXIF被置位。否则接收的数据帧

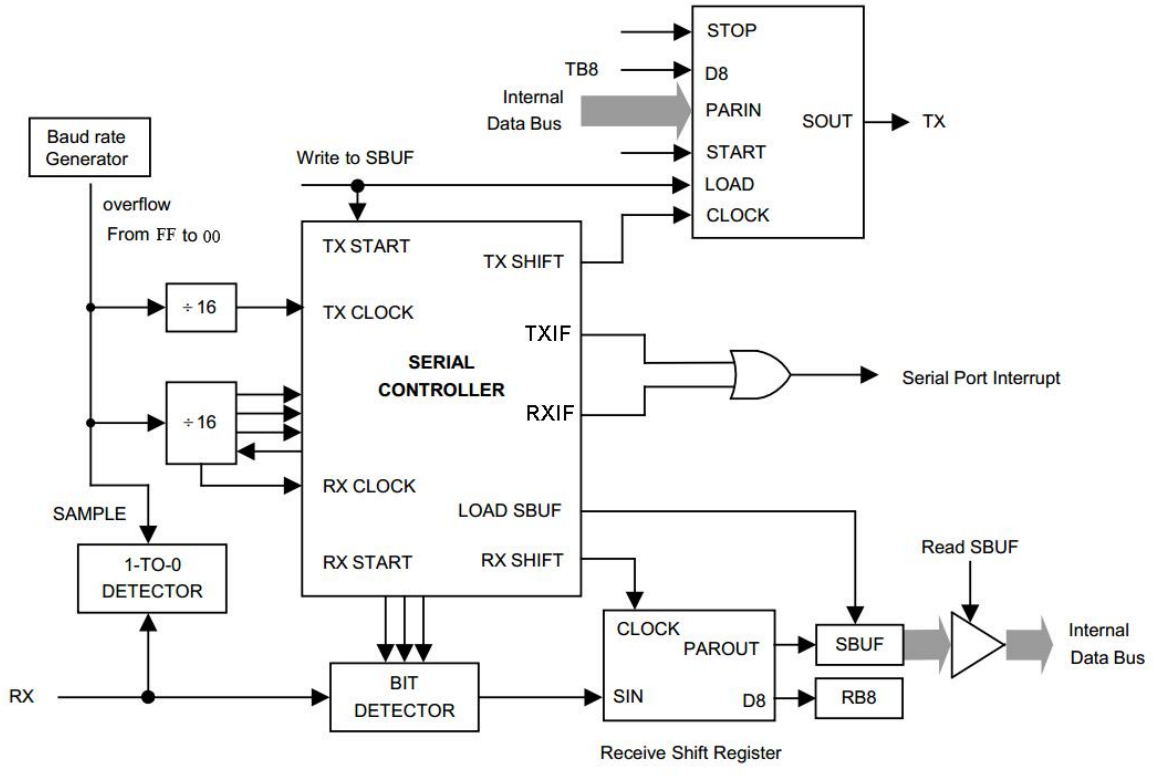
会丢失。

在停止位的当中，接收器回到寻找 RX 引脚上的另一个下降沿。用户必须用软件清除 RXIF，然后才能再次接收。



12.3.4 方式 3：9 位 UART，可变波特率，异步全双工

方式 3 使用方式 2 的传输协议以及方式 1 的波特率产生方式。



12.4 波特率发生器

UART 自带一个波特率发生器，它实质上就是一个 8 位递增计数器。

在方式0中，波特率可编程为系统时钟的1/12或1/2，由UARTM0位决定。当UARTM0为0时，串行端口在F_{CPU}的1/12下运行。当UX6为1时，串行端口在F_{CPU}的1/2下运行。

在方式2中，波特率固定为系统时钟的1/32或1/64，由SMOD位中决定。当SMOD位为0时，UART以F_{CPU}的1/64运行。当SMOD位为1时，UART以F_{CPU}的1/32运行。

$$Baud = 2^{SMOD} \times (\frac{F_{CPU}}{64})$$

在方式1和方式3中，波特率公式如下：

$$Baud = \frac{F_{CPU}}{16 \times (256-SBRT)}$$

下表为常用CPU频率与常用波特率所对应的波特率发生器的重载值：

常用 波特率	F_{CPU}		
	2MHz	4MHz	8MHz
1200	98	30	-
2400	CC	98	30
4800	E6	CC	98
9600	F3	E6	CC

12.5 多机通信

12.5.1 软件地址识别

方式2和方式3具有适用于多机通讯功能。在这两个方式下，接收的是9位数据，第9位移入RB8中，之后是停止位。可以这样设定UART：当接收到停止位，且RB8 = 1时，串行口中断有效（请求标志RXIF置位）。此时置位SM2位，UART工作在多机通讯模式。

在多机通讯系统中，按如下所述来使用这一功能。当主机要发送一数据块给几个从机中的一个时，先发送一地址字节，以寻址目标从机。地址字节与数据字节可用第9数据位来区别，地址字节的第9位为1，数据字节的第9位为0。

如果从机SM2为1，则不会响应数据字节中断。地址字节可以使所有从机产生中断，每一个从机都检查所接收到的地址字节，以判别本机是不是目标从机。被寻到的从机对SM2位执行清零操作，并准备接收即将到来的数据字节。当接收完毕时，从机再一次将SM2置位。没有被寻址的从机，则保持SM2位为1，不响应数据字节。

注：在方式1中，SM2用来检测停止位是否有效，如果SM2 = 1，接收中断不会响应直到接收到一个有效的停止位。

12.5.2 自动（硬件）地址识别

在方式2和方式3中，SM2置位，UART运行状态如下：接收到停止位，RB8的第9位为1（地址字节），且接收到的数据字节符合UART的从机地址，UART产生一个中断。从机将SM2清零，接收后续数据字节。

第9位为1表明该字节是地址而非数据。当主机要发送一组数据给几个从机中的一个时，必须先发送目标从机地址。所有从机等待接收地址字节，为了确保仅在接收地址字节时产生中断，SM2位必须置位。自动地址识别的特点是只有地址匹配的从机才能产生中断，硬件完成地址比较。

中断产生后，地址匹配的从机清零SM2，继续接收数据字节。地址不匹配的从机不受影响，将继续等待接收和它匹配的地址字节。全部信息接收完毕后，地址匹配的从机应该再次把SM2置位，忽略所有传送的非地址字节，直到接收到下一个地址字节。

使用自动地址识别功能时，主机可以通过调用给定的从机地址选择与一个或多个从机通信。主机使用广播地址可以寻址所有从机。有两个特殊功能寄存器，从机地址（SADDR）和地址屏蔽（SADEN）。从机地址是一个 8 位的字节，存于 SADDR 寄存器中。SADEN 用于定义 SADDR 各位的有效与否，如

果 SADEN 中某一位为 0，则 SADDR 中相应位被忽略，如果 SADEN 中某一位置位，则 SADDR 中相应位将用于产生约定地址。这可以使用户在不改变 SADDR 寄存器中的从机地址的情况下灵活地寻址多个从机。

	从机1	从机2
SADDR	10100100	10100111
SADEN	11111010	11111001
约定地址	10100x0x	10100xx1
广播地址	1111111x	11111111

从机1和从机2的约定地址最低位是不同的。从机1忽略了最低位，而从机2的最低位是1。因此只与从机1通讯时，主机必须发送最低位为0的地址（10100000）。类似地，从机1的第1位为0，从机2的第1位被忽略。因此，只与从机2通讯时，主机必须发送第1位为1的地址（10100011）。如果主机需要同时与两从机通讯，则第0位为1，第1位为0，第2位被两从机都忽略，两个不同的地址用于选定两个从机（1010 0001和1010 0101）。

主机可以通过广播地址与所有从机同时通讯。这个地址等于SADDR和SADEN的位或，结果中的0表示该位被忽略。多数情况下，广播地址为0xFF，该地址可被所有从机应答。

系统复位后，SADDR和SADEN两个寄存器初始化为0，这两个结果设定了约定地址和广播地址为xxxxxxxx（所有位都被忽略）。这有效地去除了多从机通讯的特性，禁止了自动寻址方式。这样的UART将对任何地址都产生应答，兼容了不支持自动地址识别的8051控制器。用户可以按照上面提到的方法实现软件地址识别的多机通讯。

12.6 帧出错检测

错误标志位被置位后，只能通过软件清零，尽管后续接收的帧没有任何错误也不会自动清零。

如果检测到一个无效（低）停止位，那么帧出错位（FE位）置1。

13 串行外部设备接口SPI

此系列单片机还提供另一种高速串行通信接口SPI接口。SPI是一种全双工、高速、同步的通信总线，有两种操作模式：主模式和从模式。在主模式中支持高达4 Mbps的速率(CPU采用更高主频时，则可更高。从模式时速度无法太快，Fsys/8以内较好)，还具有传输完成标志和写冲突标志保护。

13.1 SPI 相关寄存器

13.1.1 SPI 控制寄存器

231h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPCTL	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	x	0	0

bit [7] **SSIG**：SS引脚忽略控制位

1 = MSTR确定器件为主机还是从机

0 = SS脚用于确定器件为主机还是从机，SS脚可作为I/O使用

bit [6] **SPEN**：SPI使能位

1 = SPI使能

0 = SPI被禁止，所有SPI引脚都作为I/O使用

bit [5] **DORD**：PWM21输出控制位

1 =数据字的LSB（最低位）最先发送

0 =数据字的 MSB (最高位) 最先发送

bit [4] **MSTR**：主/从模式选择位

1 = 主机

0 = 从机

bit [3] **CPOL**：SPI时钟极性

1 = SPICLK空闲时为高电平。SPICLK的前时钟沿为下降沿而后沿为上升沿

0 = SPICLK 空闲时为低电平。SPICLK 的前时钟沿为上升沿而后沿为下降沿

bit [2] **CPHA**：SPI时钟相位选择

1 =数据在SPICLK的前时钟沿驱动，并在后时钟沿采样

0 =数据在SS为低（SSIG=0）时被驱动，在SPICLK的后时钟沿被改变，并在前时钟沿被采样。（SSIG=1时的操作未定义）

bit [1:0] **SPR[1:0]**：SPI时钟速率选择控制位

11 = Fcpu/128

10 = Fcpu/64

01 = Fcpu/16

00 = Fcpu/4

13.1.2 SPI 状态寄存器

054h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR2	PWM2IF	PWM1IF	PWM0IF	-	RXIF	TXIF	SPIF	CCP2IF
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
POR的值	0	0	0	-	0	0	0	0

bit [1] **SPIF**：SPI传输完成标志

1 = 串行传输完成时，硬件置1，软件清零，当SPI处于主模式且SSIG=0时，如果SS为输入并被驱动为高电平，SPIF也将置位,表示“模式改变”

0 = 串行传输未完成。

230h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPSTAT	-	-	-	-	-	RXOV	MODF	WCOL
R/W	-	-	-	-	-	R/W	R/W	R/W
POR的值	-	-	-	-	-	0	0	0

bit [2] **RXOV**：接收溢出标志

1 = 发生接收溢出，软件清零

0 = 未发生接收溢出。

bit [1] **MODF**：模式改变错误标志

1 = 发生模式改变错误，软件清零

0 = 未发生模式改变错误。

bit [0] **WCOL**：SPI写冲突标志

1 = 在数据传输的过程中如果对SPI数据寄存器SPDAT执行写操作，WCOL将置1，软件清零

0 = 未发生写冲突。

13.1.3 SPI 数据寄存器

232h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPDAT								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	x	0	0

bit [7:0] **SPDAT[7:0]**：SPI 数据寄存器

13.2 SPI 信号描述

主输出从输入（MOSI）：该信号连接主设备和一个从设备，数据通过 MOSI 从主设备串行传送到从设备，主设备输出，从设备输入。

主输入从输出（MISO）：该信号连接主设备和一个从设备。数据通过 MISO 从从设备串行传入到主设备，从设备输出，主设备输入。若该设备为从设备且未被选时，从设备的 MISO 引脚处于高阻状态。

串行时钟（SCK）：该信号用作控制 MOSI 和 MISO 线上输入输出数据的同步移动，每 8 个时钟周期 MOSI 和 MISO 线上传送一个字节，如果从设备未被选中，SCK 信号将被此设备忽略。注意：只有主设备才能产生 SCK 信号。

从设备选择引脚（$\overline{SS}$）： 每个从属外围设备由一个从选择引脚$\overline{SS}$选择，当引脚信号为低电平时，表明该从设备被选中。主设备可以通过软件控制连接于从设备$\overline{SS}$引脚的端口电平选择每个从设备，很明显，

只有一个主设备可以驱动通讯网络。为了防止 MISO 总线冲突，同一时间只允许一个从设备与主设备通讯。在主设备模式中，$\overline{SS}$引脚状态关联 SPI 状态寄存器 SPSTAT 中 MODF 标志位以防止多个主设备驱动 MOSI 和 SCK。

下列情况，$\overline{SS}$引脚可以作为普通端口或其它功能使用：

（1） 设备作为主设备，SPI 控制寄存器 SPCTL 寄存器的 SSIG 位置 1。这种配置仅仅存在于通讯网络中只有一个主设备的情况，因此，SPI 状态寄存器 SPSTA 中 MODF 标志位不会被置 1。

（2） 设备配置为从设备，SPI 控制寄存器 SPCTL 的 CPHA 位和 SSIG 位置 1。这种配置情况存在于只有一个主设备一个从设备的通讯网络中，因此，设备总是被选中的，主设备也不需要控制从设备的$\overline{SS}$引脚选择其作为通讯目标。

从设备的$\overline{SS}$引脚被使能时，其它主设备可通过使该引脚维持低电平，从而选中该从设备。为防止 MISO 总线冲突，原则上不允许两个及以上的从设备被选中。

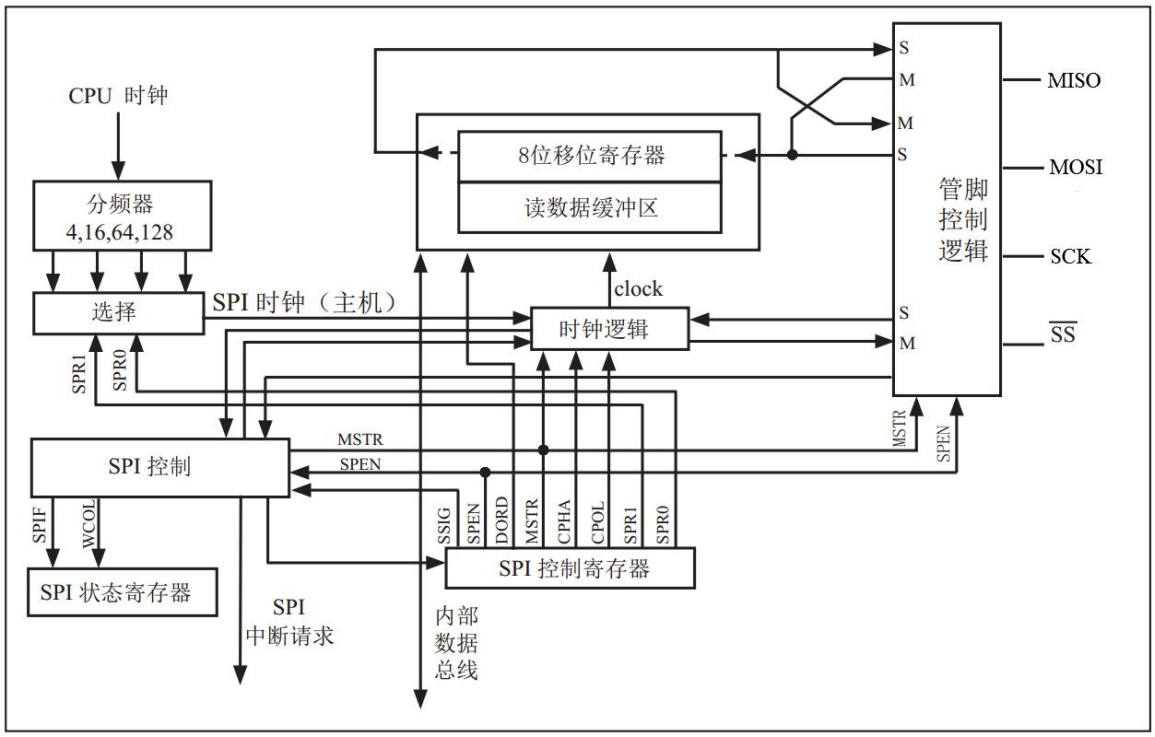
主设备的$\overline{SS}$引脚被使能时，若$\overline{SS}$被拉低将置模式错误标志 MODF（可中断），且 MSTR 位也将被清零，从而使该设备强制切换成从设备。

当 MSTR = 0（从模式）及 CPHA = 0 时，SSIG 必须为 0，因为此时数据传送需要$\overline{SS}$引脚配合，才能完成多数据传送。

13.3 SPI 时钟速率

在主模式下，SPI 的速率有 4 级选择，分别是内部时钟的 4、16、64 或 128 分频，可通过 SPCTL 寄存器的 SPR[1:0]位进行选择。

13.4 SPI 功能框图

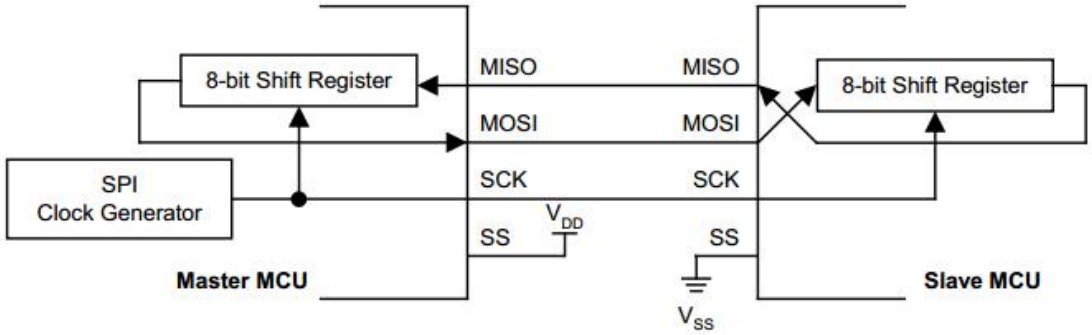


13.5 SPI 工作模式

SPI 可配置为主模式或从模式中的一种。SPI 模块的配置和初始化通过设置相关寄存器来完成。进一步设置相关寄存器即可完成数据传送。

在 SPI 通讯期间，数据同步地被串行的移进移出，串行时钟线（SCK）使两条串行数据线（MOSI&MISO）上数据的移动和采样保持同步。从设备选择线（$\overline{SS}$）可以独立地选择从属设备；如果从设备没有被选中，则不能参与 SPI 总线上的活动。

当 SPI 主设备通过 MOSI 线传送数据到从设备时，从设备通过 MISO 线发送数据到主设备作为相应，从而实现在同一时钟下数据发送与接收的同步全双工传输。发送移位寄存器和接收寄存器使用相同的 SFR 地址，对 SPI 数据寄存器 SPDAT 进行写操作将写入发送移位寄存器，对 SPDAT 寄存器进行读操作将获得接收移位寄存器的数据。注：写入的数据不会影响到需要读出的数据。



主模式

(1) 模式启动

SPI 主设备控制控制 SPI 总线上的所有数据传送的启动。一个 SPI 总线中只允许一个主设备可以启动传送。

(2) 发送

在 SPI 主模式下，写一个字节数据到 SPI 数据寄存器 SPDAT，数据将会写入发送移位缓冲器。如果发送移位寄存器中已经存在一个数据或正在传送一个数据，那么主 SPI 将产生一个 WCOL 信号以表明写入太快。但是发送移位寄存器中的数据不会受到影响，发送也不会中断。

(3) 接收

当主设备通过 MOSI 线传送数据到从设备时，同时对应的从设备也可以通过 MISO 线将其发送移位寄存器的数据传送给主设备的接收移位寄存器，实现全双工操作。故 SPIF 标志置 1 即表示数据发送完成也表示数据接收完成。本 SPI 模块接收为双缓冲器，即数据可以在 SPIF 置 1 后读出，但必需在下一字节数据接收完成前读出，否则将置接收溢出标志 RXOV，如果发生接收溢出，则后面的数据将不会被移入接收寄存器，接收溢出时，SPIF 可正常置 1。

从模式

(1) 模式启动

将 MSTR 置 0（若$\overline{SS}$被使能则必需拉低）时，设备处于从模式下运行，数据传送过程中设备模式不能改变（$\overline{SS}$引脚必需维持低电平），否则数据传送将失败（SPIF 不会被置 1）。

(2) 发送

SPI 从设备下不能启动数据传送，所以 SPI 从设备必需在主设备开始一次新的数据传送之前将要传送给主设备的数据写入发送移位寄存器。若发送前未写入数据到发送移位寄存器，从设备将传送数据“0x00”给主设备。若写入数据时发送移位寄存器已经存在数据（或发生在传送过程中），那么 SPI 从设

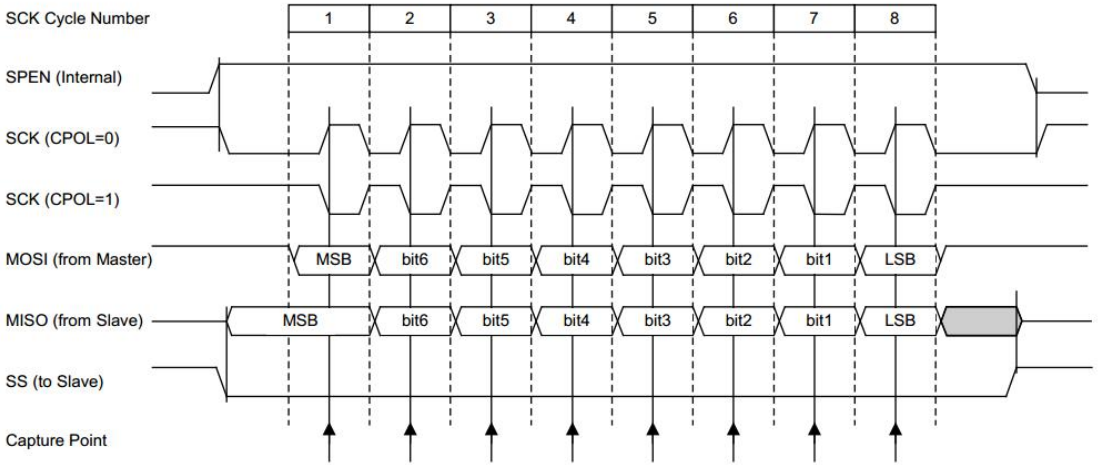
备的 WCOL 标志位将置 1，表示发生写 SPDAT 冲突。但是移位寄存器的数据不受影响，传送也不会被中断，传送完成 SPIF 将被置 1。

(3) 接收

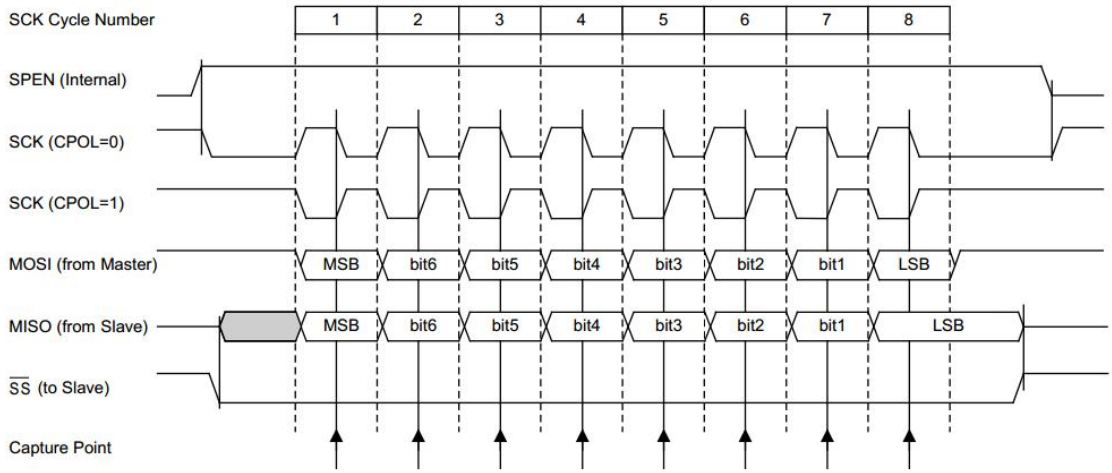
从模式下，按照主设备控制的 SCK 信号，数据通过 MOSI 引进移入，当计数器计数 SCK 边缘数到 8 时，表示一个字节数据接收完毕，SPIF 将置 1，数据可以通过此时读取 SPDAT 寄存器获得，但必需在下一数据接收完成前被读出，否则将置接收溢出标志 RXOV，如果发生接收溢出，则后面的数据将不会被移入接收寄存器，接收溢出时，SPIF 可正常置 1。

13.6 SPI 传送形式

通过软件设置寄存器的 CPOL 位和 CPHA 位，用户可以选择 SPI 时钟极性和相位的四种组合方式。CPOL 位定义时钟的极性，即空闲时的电平状态。CPHA 位定义时钟相位，即定义允许数据移位采样的时钟边沿。在通信的两个主从设备中，时钟极性相位设置应当保持一致。

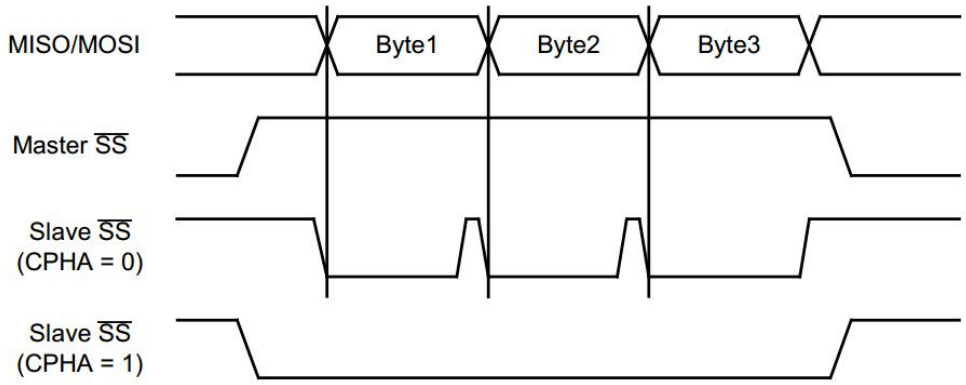


如果 CPHA = 0；数据在 SCK 的第一沿就被捕获，所以从设备必需在 SCK 的第一个沿之前就准备好数据，因此，$\overline{SS}$引脚的下降沿从设备就开始数据。$\overline{SS}$引脚在每次传送完一个字节后必需拉高，在发送下一字节之前重新又被拉低，故 CPHA = 0 时，SSIG 位无效，即$\overline{SS}$脚被强制使能。



如果 CPHA = 1，主设备在 SCK 的第一个沿将数据输出到 MOSI 线上，从设备把 SCK 的第一个沿作为开始发送信号。用户必需在第一个 SCK 的前 2 个沿内完成对 SPDAT 完成写操作。传送过程中彼此模式不能改变，否则数据发送接收将失败，模式被改变的寄存器数据（发送数据）及状态（接收为空）不

变。这种数据传送形式为单一主从设备间通信的首先形式。



13.7 SPI 出错检测

在数据未发送或发送期间继续对 SPDAT 做写入操作会引起写冲突，WCOL 位会被置 1，但发送不会终止。需软件清零

13.8 SPI 配置对照表

SPEN	SSIG	$\overline{SS}$	MSTR	主或从模式	MISO	MOSI	SCK	备注
0	x	I/O	x	SPI功能禁止	I/O	I/O	I/O	SPI禁止
1	0	0	0	从机模式	输出	输入	输入	选择从机
1	0	1	0	从机模式未被选中	高阻	输入	输入	未被选中。MISO为高阻，以避免总线冲突
1→0	0	0	1→0	关闭SPI	输出	输入	输入	SS配置为输入，SSIG为0。如果SS被驱动为低电平。则被选择作为从机。此时MSTR将清零，并置模式错误标志MODF，可用于请求中断。
1	0	1	1	主（空闲）	输入	高阻	高阻	当主机空闲时MOSI和SCK为高阻态以避免总线冲突。用户必须将SCK上拉或下拉（根据CPOL的取值）以避免SCK出现悬浮状态。
				主（激活）		输出	输出	作为主机激活时，MOSI和SCK为推挽输出。
1	1	I/O	0	从	输出	输入	输入	CPHA不能为0
1	1	I/O	1	主	输入	输出	输出	-

14 指令表

Field	指令格式		描述	C	DC	Z	周期
移动	MOVWF	F	F←W	-	-	-	1
	MOVF	F, D	D←F(D =0 时为 W, D =1 时为 F)	-	-	√	1
	MOVLW	k	W←k	-	-	-	1
算术	ADDWF	F, D	D←W+F(D =0 时为 W, D =1 时为 F)	√	√	√	1
	ADDLW	k	W←W+k	√	√	√	1
	SUBWF	F, D	D←F-W(D =0 时为 W, D =1 时为 F)	√	√	√	1
	SUBLW	k	W←k -W(D =0 时为 W, D =1 时为 F)	√	√	√	1
	DAW	-	W 寄存器值进行 BCD 调整	√	√	-	1
	INCF	F, D	D←F+1(D =0 时为 W, D =1 时为 F)	-	-	√	1
	DECF	F, D	D←F-1(D =0 时为 W, D =1 时为 F)	-	-	√	1
逻辑	ANDWF	F, D	D←W 与 F(D =0 时为 W, D =1 时为 F)	-	-	√	1
	ANDLW	k	W←W 与 k	-	-	√	1
	IORWF	F, D	D←W 或 F(D =0 时为 W, D =1 时为 F)	-	-	√	1
	IORLW	k	W←W 或 k	-	-	√	1
	XORWF	F, D	D←W 异或 F(D =0 时为 W, D =1 时为 F)	-	-	√	1
	XORLW	k	W←W 异或 k	-	-	√	1
	COMF	F, D	D←F 取反(D =0 时为 W, D =1 时为 F)	-	-	√	1
处理	SWAPF	F, D	D[7:4,3:0]←F[3:0,7:4] (D =0 时为 W, D =1 时为 F)	-	-	-	1
	RRF	F, D	D←F 带进位右移(D =0 时为 W, D =1 时为 F)	√	-	-	1
	RLF	F, D	D←F 带进位左移(D =0 时为 W, D =1 时为 F)	√	-	-	1
	CLRW	-	W←0	-	-	√	1
	CLRF	F	F←0	-	-	√	1
	CLRWDT	-	清零看门狗定时器，影响 TO，PD 位	-	-	-	1
	BCF	F, d	F[d]←0(0≤d≤7)	-	-	-	1
	BSF	F, d	F[d]←1(0≤d≤7)	-	-	-	1
分支	INCFSZ	F, D	D←F+1(D =0 时为 W, D =1 时为 F)，如果 D=0 则跳过下一句	-	-	-	1(2)
	DECFSZ	F, D	D←F-1(D =0 时为 W, D =1 时为 F)，如果 D=0 则跳过下一句	-	-	-	1(2)
	BTFSC	F, d	如果 F[d]=0(0≤d≤7)则跳过下一句	-	-	-	1(2)
	BTFSS	F, d	如果 F[d]=1(0≤d≤7)则跳过下一句	-	-	-	1(2)
	GOTO	k	无条件跳转	-	-	-	2
	CALL	k	调用子程序	-	-	-	2
其他	RETURN	-	从子程序返回	-	-	-	2
	RETFIE	-	从中断返回，并置位 GIE	-	-	-	2
	RETLW	k	W←k，带参数返回	-	-	-	2
	NOP	-	空操作	-	-	-	1
	SLEEP	-	进入待机模式，影响 TO，PD 位	-	-	-	1

15 电性参数

15.1 极限参数

储存温度……-50°C~125°C
工作温度……-40°C~85°C
电源供应电压……VSS-0.3V~VSS+6.0V
端口输入电压……VSS-0.3V~VDD+0.3V

15.2 直流特性

符号	参数	测试条件		最小值	典型值	最大值	单位
		VDD	条件（常温 25°C）				
VDD	工作电压	—	Fsys = 8MHz，2T	1.8	—	5.5	V
		—	Fsys = 16MHz，2T	2.6	—	5.5	V
IDD1	工作电流	3V	Fsys = 16MHz，4T，LCD 关闭，高频模式， WDT 禁止，无负载	—	1.70	—	mA
		5V		—	3.00	—	mA
IDD2	工作电流	3V	Fsys = 8MHz，4T，LCD 关闭，高频模式， WDT 禁止，无负载	—	1.20	—	mA
		5V		—	2.20	—	mA
IDD3	工作电流	3V	Fsys = 32KHz，4T，LCD 关闭，低频模式， WDT 禁止，无负载	—	3.0	—	μA
		5V		—	10.0	—	μA
IDD4	工作电流	3V	Fsys = 32KHz，4T，LCD 关闭，绿色模式， WDT 禁止，无负载	—	2.0	—	μA
		5V		—	7.0	—	μA
Isb1	静态电流	3V	LCD 关闭，休眠模式，WDT 使能，无负载	—	2.0	—	μA
		5V		—	7.0	—	μA
Isb2	静态电流	3V	LCD 关闭，休眠模式，外部低频晶振计时，无负载	—	2.0	—	μA
		5V		—	5.2	—	μA
Isb3	静态电流	3V	LCD 打开（270K 偏置电阻），休眠模式，WDT 禁止，无负载	—	5	—	μA
		5V		—	10	—	μA
Isb3	静态电流	3V	LCD 打开（270K 偏置电阻），休眠模式 WDT 禁止 全部点亮	—	14	—	μA
		5V		—	36	—	μA
Isb4	静态电流	3V	LCD 关闭，休眠模式，WDT 禁止，无负载	—	—	1	μA
		5V		—	—	1	μA
V_{IL1}	输入低电平	—	输入口	VSS	—	0.3VDD	V
V_{IH1}	输入高电平	—	输入口	0.7VDD	—	VDD	V
V_{IL2}	输入低电平	—	施密特输入口	VSS	—	0.2VDD	V
V_{IH2}	输入高电平	—	施密特输入口	0.8VDD	—	VDD	V
V_{BOR1}	低电压复位 2.0V	—	—	—	2.0	—	V
V_{BOR2}	低电压复位 2.4V	—	—	—	2.4	—	V
V_{BOR3}	低电压复位 3.6V	—	—	—	3.6	—	V

V_{LVD0}	低电压标志	—	—	—	2.4	—	V
V_{LVD1}	低电压标志	—	—	—	3.6	—	V
I_{OL1}	PORTF(LFEN=0)	3V	VOL=VSS+0.6V	—	30	—	mA
		5V		—	65	—	mA
I_{OH1}	PORTF(LFEN=0)	3V	VOH=VDD-0.6V	—	11	—	mA
		5V		—	17	—	mA
I_{OL2}	PORTF(LFEN=1)	3V	VOL=VSS+0.6V	—	9	—	mA
		5V		—	20	—	mA
I_{OH2}	PORTF(LFEN=1)	3V	VOH=VDD-0.6V		4	—	mA
		5V		—	6.5	—	mA
I_{OL3}	PORTx(LFEN=0)	3V	VOL=VSS+0.6V	—	18	—	mA
		5V		—	28	—	mA
I_{OH3}	PORTx(LFEN=0)	3V	VOH=VDD-0.6V	—	9.0	—	mA
		5V		—	17	—	mA
I_{OL4}	PORTx(LFEN=1)	3V	VOL=VSS+0.6V	—	5.0	—	mA
		5V		—	8.0	—	mA
I_{OH4}	PORTx(LFEN=1)	3V	VOH=VDD-0.6V	—	3.5	—	mA
		5V		—	5.5	—	mA
R_{PH}	内部上拉电阻	5V	可编程上拉电阻	—	100	—	kΩ
Rpd	内部下拉电阻	5V	可编程下拉电阻	—	100	—	kΩ
Vad	ADC 输入电压	—	—	VSS	—	V_{REF}	V
DNL	差分非线性误差	5V	AD 时钟频率 2MHz	—	±1	—	LSB
INL	积分非线性误差	5V	AD 时钟频率 2MHz	—	±1	—	LSB
Iadc	ADC 工作电流	3V	AD 时钟频率 2MHz	—	0.3	—	mA
		5V		—	0.5	—	mA

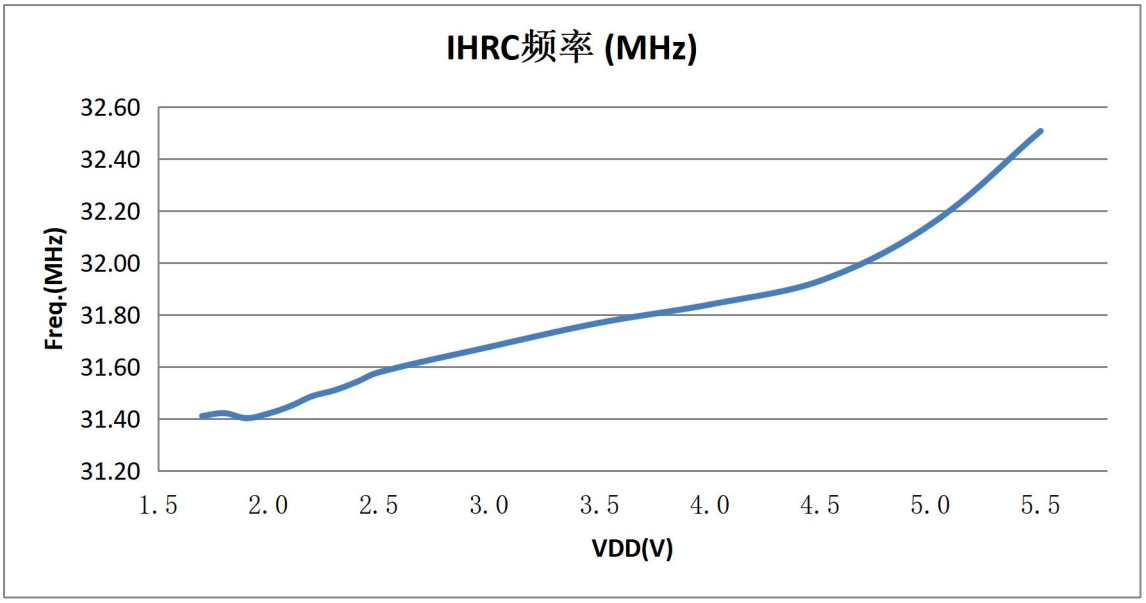
注：表格中 x = A、B、C、D、E 五组 PORT 口。

15.3 交流特性

符号	参数	测试条件		最小值	典型值	最大值	单位
		VDD	条件（常温 25°C）				
F_{SYS}	系统时钟	—	1.8V~5.5V	—	8	—	MHz
		—	2.6V~5.5V	—	16	—	MHz
F_{RCH}	高频内部 RC 振荡器	5V	—	—	32	—	MHz
F_{RCL}	低频内部 RC 振荡器	5V	—	—	32	—	KHz
F_{OSH}	外部高频晶振	—	2.6~5.5V	1	—	20	MHz
F_{OSL}	外部低频晶振	—	2.0~5.5V	—	32.768	—	KHz
T_{VDD}	VDD 上升时间	5V	—	—	—	100	ms
T_{BOR}	欠压复位响应时间	5V	—	100	—	—	ns
T_{WDT}	看门狗溢出时间	5V	使用预分频 1:1	—	18	—	ms
			不使用预分频器	—	72	—	ms

TMCLRB	复位脉冲时间	5V	—	200	—	—	μs

15.4 电气特性曲线图



16 开发工具

16.1 OTP 烧录器（PM18-5.0）

- HC-PM18-5.0：支持HC18系列MCU大批量的脱机烧录

注：
详情请参考 **HC-PM18** 用户手册。如有更新请到网站下载最新资料。
http://www.holychip.cn

16.2 HC-IDE

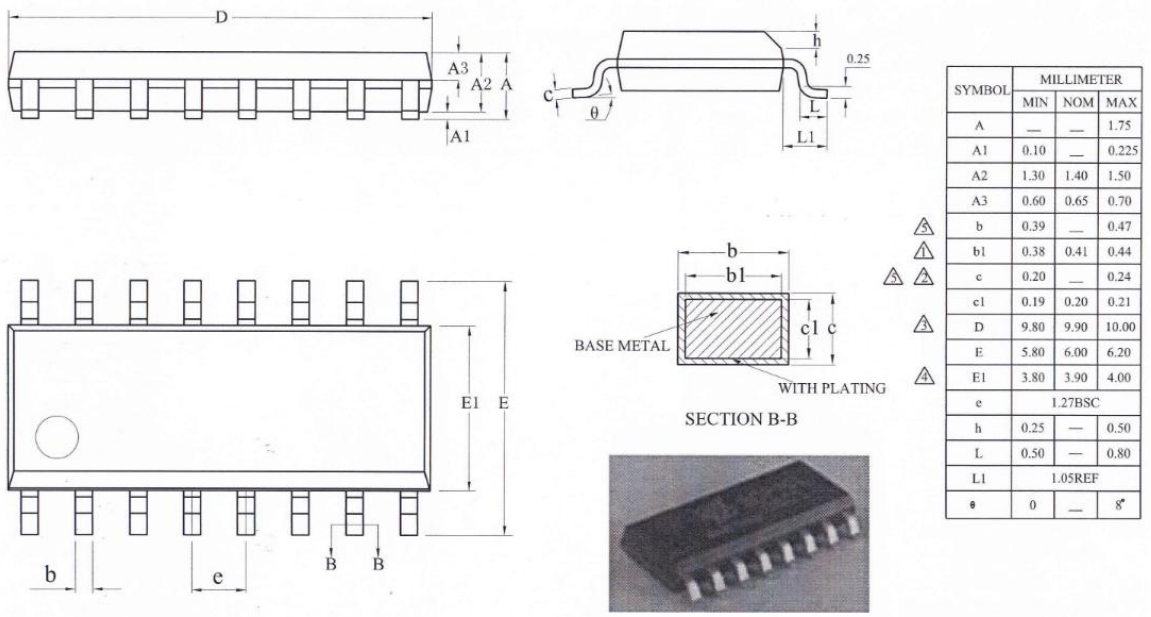
Holychip 8位单片机的集成开发环境HC-IDE包括编译器。

- HC-IDE：HC-IDE V3.0.5.0(支持汇编/C语言)

注：
1、 详情请参考 HC-IDE 用户手册。
2、 IDE 版本请关注芯圣官网：http://www.holychip.cn/

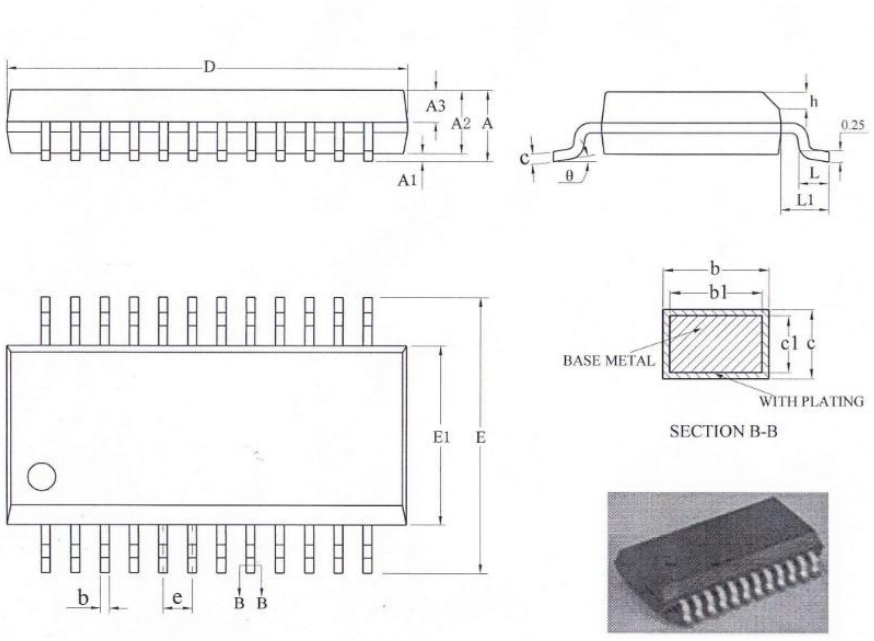
17 封装尺寸

17.1 SOP16



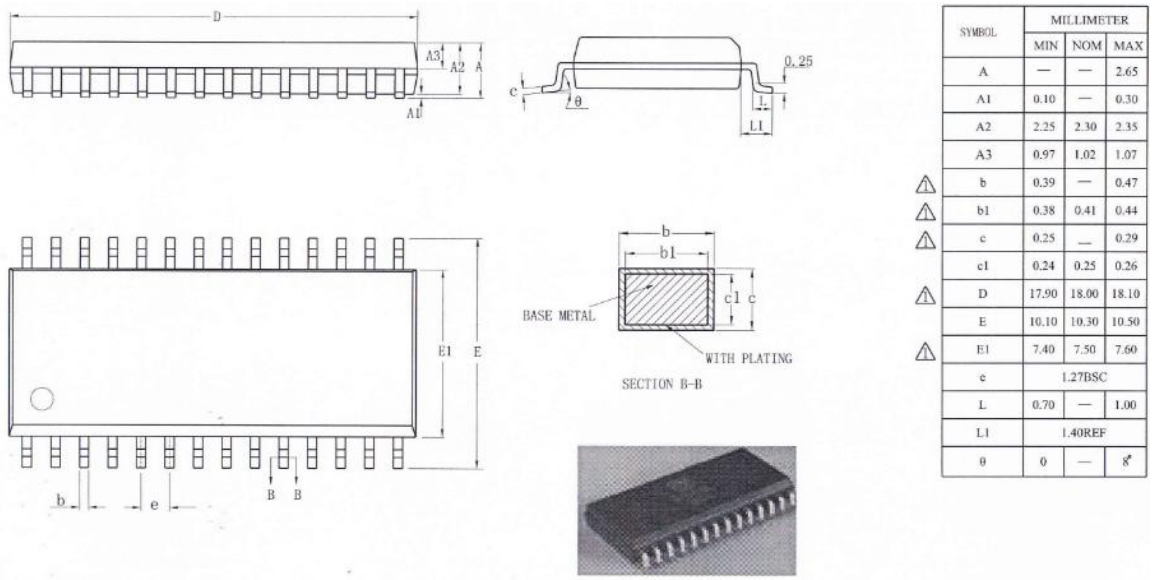
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.10	—	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
5 b	0.39	—	0.47
1 b1	0.38	0.41	0.44
2 3 c	0.20	—	0.24
c1	0.19	0.20	0.21
3 D	9.80	9.90	10.00
E	5.80	6.00	6.20
4 E1	3.80	3.90	4.00
e	1.27BSC		
h	0.25	—	0.50
L	0.50	—	0.80
L1	1.05REF		
θ	0	—	8°

17.2 SSOP24



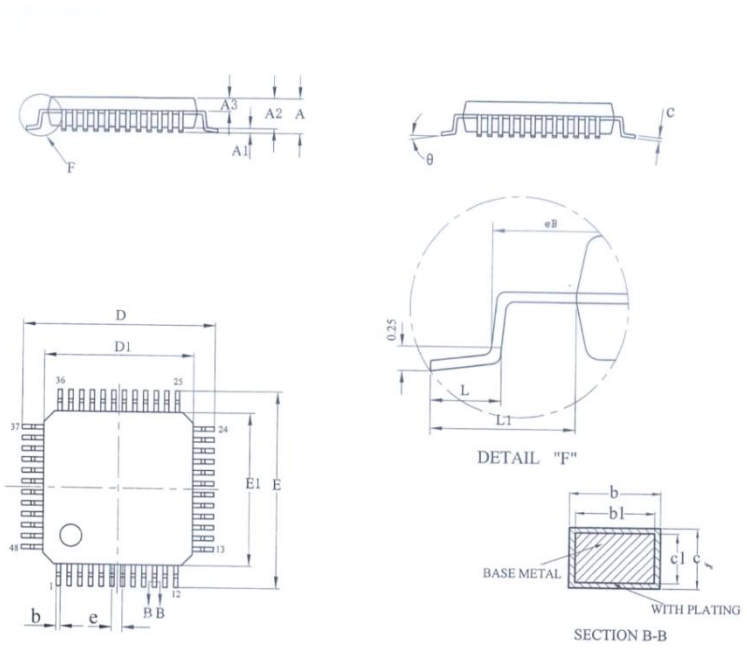
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.10	0.15	0.25
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.23	—	0.31
b1	0.22	0.25	0.28
c	0.20	—	0.24
c1	0.19	0.20	0.21
D	8.55	8.65	8.75
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	0.635BSC		
h	0.30	—	0.50
L	0.50	—	0.80
L1	1.05REF		
θ	0	—	8°

17.3 SOP28



	SYMBOL	MILLIMETER		
		MIN	NOM	MAX
	A	—	—	2.65
	A1	0.10	—	0.30
	A2	2.25	2.30	2.35
	A3	0.97	1.02	1.07
⚠	b	0.39	—	0.47
⚠	b1	0.38	0.41	0.44
⚠	c	0.25	—	0.29
	c1	0.24	0.25	0.26
⚠	D	17.90	18.00	18.10
	E	10.10	10.30	10.50
⚠	E1	7.40	7.50	7.60
	e	1.27BSC		
	L	0.70	—	1.00
	L1	1.40REF		
	θ	0	—	8°

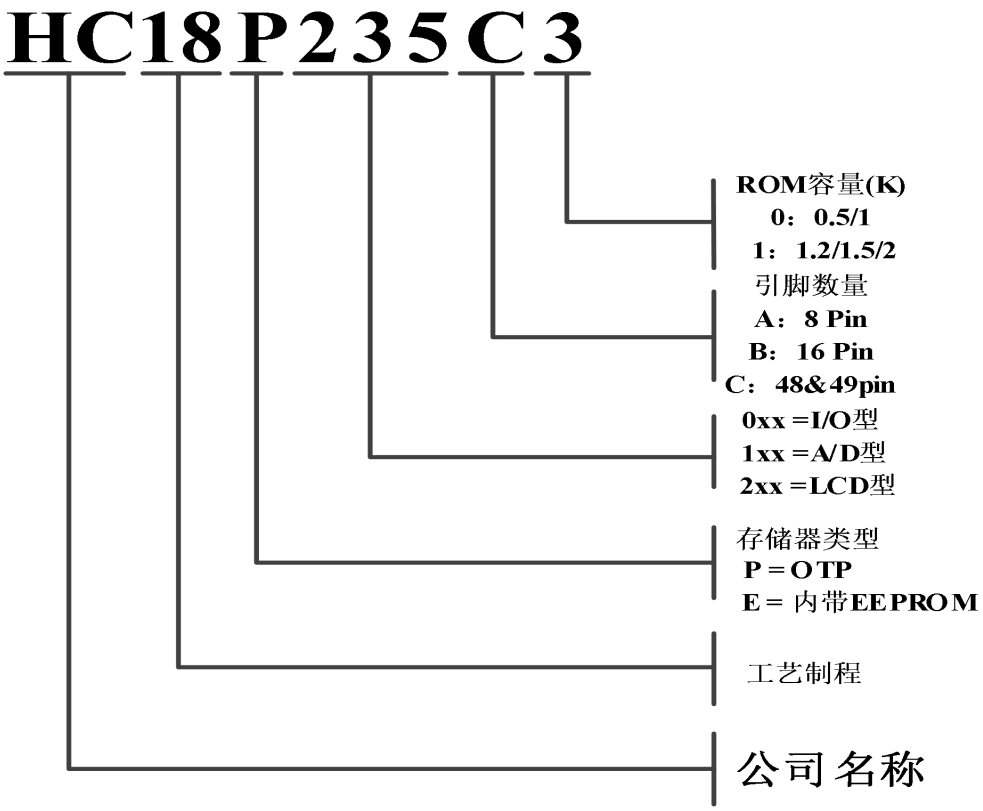
17.4 LQFP48



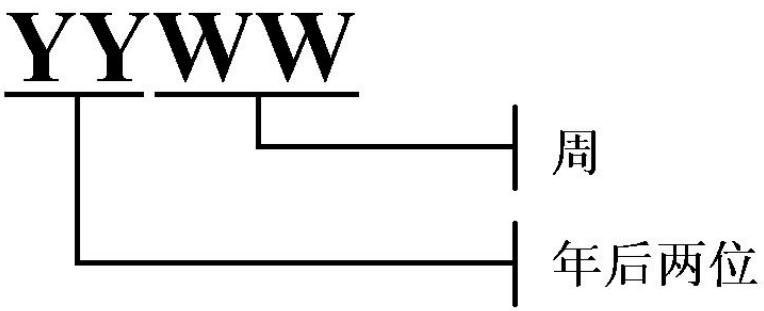
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.19	—	0.27
b1	0.18	0.20	0.23
c	0.13	—	0.18
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
eB	8.10	—	8.25
e	0.50BSC		
L	0.45	—	0.75
L1	1.00BSC		
θ	0	—	7°

18 芯片正印命名规则

18.1 芯片型号说明（第一行）



18.2 日期码规则（第二行）



18.3 生产批号（第三行）

例：**FA126026A**

19 修改记录

版本	日期	描述
Ver1.00	2020-11	第一版
Ver1.01	2021-09-18	P72，CCP-PWM 使用 T1 时基，T1L 必须清零
Ver1.01	2021-09-18	P72，CCP-PWM 使用 T1 时基时钟源 Fsys，时钟模式必须为 4T

芯圣电子

2020年11月