HC18P110A0/B0

数据手册

16/14/8 引脚 8 位
ADC 型 OTP 单片机

目录

1 产品简述

HC18P110A0/B0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，内部有1K*14位一次性编程ROM(OTP-ROM)，128*8位的数据寄存器（RAM），两个双向I/O口，三个Timer定时器/计数器，两个CCP模块，一个AD模块，多个系统时钟，四种系统工作模式，一个8通道的12位模数转换器以及多个中断源。这款单片机可以广泛应用于测量、马达控制、工业控制、家电类产品等。

1.1 特性

- **存储器**
  程序存储器（OTP ROM）空间：1K*14 位
  数据存储器（RAM）空间：128*8 位
- **8 级堆栈缓冲器**
- **PAD 配置**
  输入输出双向端口：
  PORTA、PORTB（PORTB5 只能输出低）
  内置上拉电阻端口：
  PORTA、PORTB
  具有唤醒功能的电平变化中断端口：
  PORTB，可通过 IOCB 独立配置
  具有唤醒功能的外部中断引脚：
  PORTB0，可设置触发边沿
- **高灌（TYP：25mA）/高拉（TYP：16mA）电流能力，可直接驱动 LED**
- **低电压复位系统（BOR）**
  BOR1.5V/1.8V/2.0V/2.2V/2.4V/2.7V/3.6V
- **6 个中断源**
  定时器中断：Timer0、Timer1和Timer2
  INT0外部中断
  PORTB电平变化中断
  ADC中断
  Timer1门控中断
  CCP1/CCP2中断
- **强大的指令系统**
  机器周期由 2 或 4 个时钟周期组成（2T/4T）
  大部分指令仅需一个机器周期
  跳转指令 GOTO 可在整个 ROM 区执行
  调用指令 CALL 可在整个 ROM 区执行
- ESD：4KV
- **封装形式**
  SOP8/14/16
- **AD模数转换器**
  12位转换分辨率
  最多8个模拟输入通道（7个外部ADC输入，1个内部电池检测）
  内部参考电压（VDD、4V、3V、2V）；外部参考电压
  ADC时钟源：Fosc/1/2/4/8/16/32/64，FRC；
- **三个 Timer 定时器/计时器**
  Timer0：带8位预分频器的8位定时器/计数器
  Timer1：16位定时器/计数器，四种门控模式
  Timer2：带8位周期寄存器的8位定时器
- **两个CCP模块**
  16位捕捉，最大分辨率12.5 ns
  16位比较，最大分辨率200 ns
- **一个最高 12 位分辨率脉宽调制模块（PWM）**
- **看门狗定时器**
  时钟源由内部低频 RC 振荡器提供
  溢出时间软件可设
- **双时钟系统**
  高速时钟：晶体振荡器模式，高达 20MHz
  外部 RC 模式
  内部 RC 模式，高达 8MHz
  低速时钟：晶体振荡器模式，32768Hz
  内部 RC 模式，32KHz@5V
- **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）

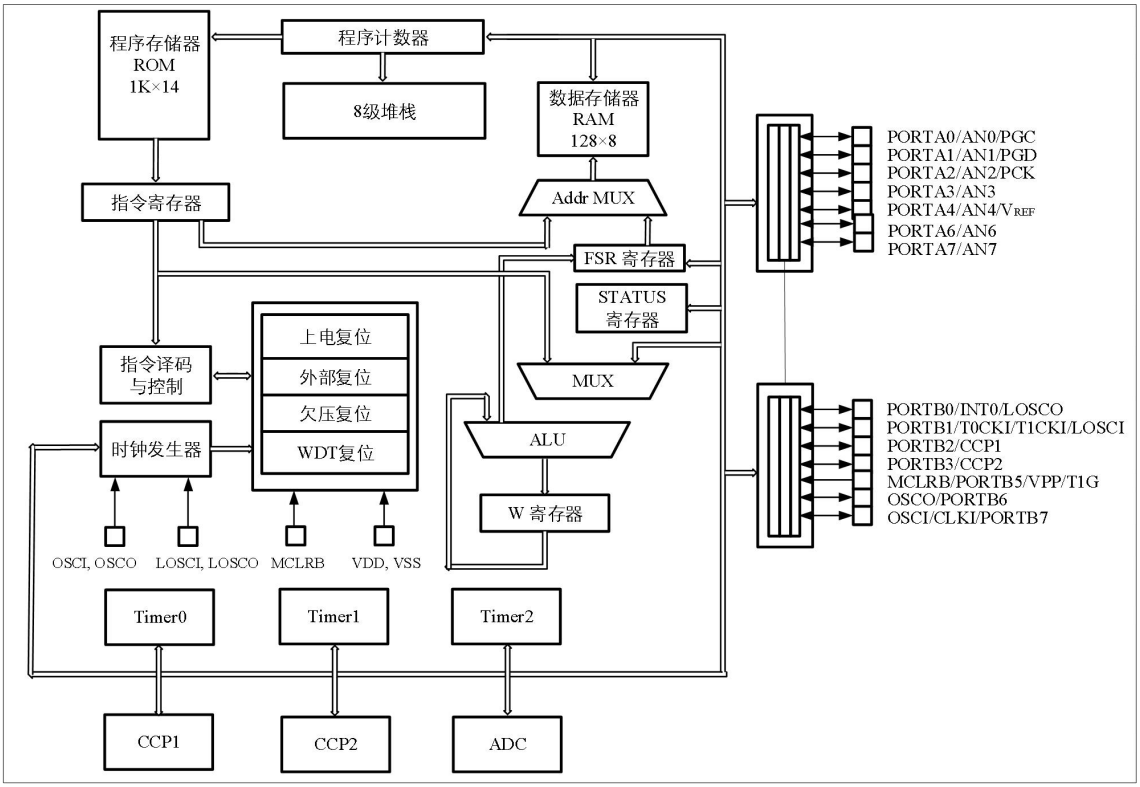
✓ 选型表

产品型号	ROM	RAM	堆栈	定时器	I/O	CCP	Wake up port	ADC	封装形式
HC18P110A0	1K*14	128*8	8	3	5+1	1	2	12*(4+1)	SOP8
HC18P110B0	1K*14	128*8	8	3	11+1	2	7	12*(5+1)	SOP14
HC18P111B0	1K*14	128*8	8	3	13+1	2	7	12*(7+1)	SOP16

芯片使用注意事项:

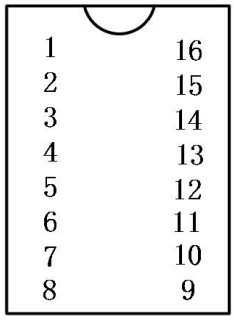
PORTB5 端口为开漏输出端口，方向寄存器 TRISB[5]控制端口输入/输出模式。

1.2 系统框图



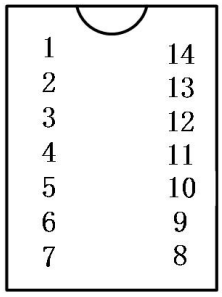
1.3 引脚图

HC18P110B0_16PIN



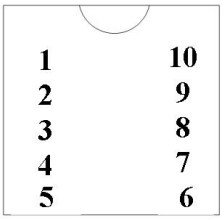
VDD	1	16	VSS
OSCI/CLKI/PORTB7	2	15	PORTA0/AN0/PGC
OSCO/PORTB6	3	14	PORTA1/AN1/PGD
T1G/MCLRB/PORTB5/VPP	4	13	PORTA2/AN2/PCK
PORTB3/CCP2	5	12	PORTA3/AN3
PORTB2/CCP1	6	11	PORTA4/AN4/V_{REF}
PORTB1/T0CKI/T1CKI/LOSCI	7	10	PORTA6/AN6
PORTB0/INT0/LOSCO	8	9	PORTA7/AN7

HC18P110B0_14PIN



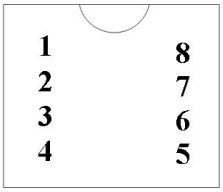
VDD	1	14	VSS
OSCI/CLKI/PORTB7	2	13	PORTA0/AN0/PGC
OSCO/PORTB6	3	12	PORTA1/AN1/PGD
T1G/MCLRB/PORTB5/VPP	4	11	PORTA2/AN2/PCK
PORTB3/CCP2	5	10	PORTA3/AN3
PORTB2/CCP1	6	9	PORTA4/AN4/V_{REF}
PORTB1/T0CKI/T1CKI/LOSCI	7	8	PORTB0/INT0/LOSCO

HC18P110A0_10PIN



VDD	1	10	VSS
OSCO/PORTB6	2	9	PORTA0/AN0/PGC
T1G/MCLRB/PORTB5/VPP	3	8	PORTA1/AN1/PGD
PORTB3/CCP2	4	7	PORTA2/AN2/PCK
PORTB2/CCP1	5	6	PORTA4/AN4/Vref

HC18P110A0_8PIN



VDD	1	8	VSS
MCLRB/VPP/PORTB5	2	7	PORTA0/AN0/PGC
CCP1/PORTB2	3	6	PORTA1/AN1/PGD
Vref/AN4/PORTA4	4	5	PORTA2/AN2/PCK

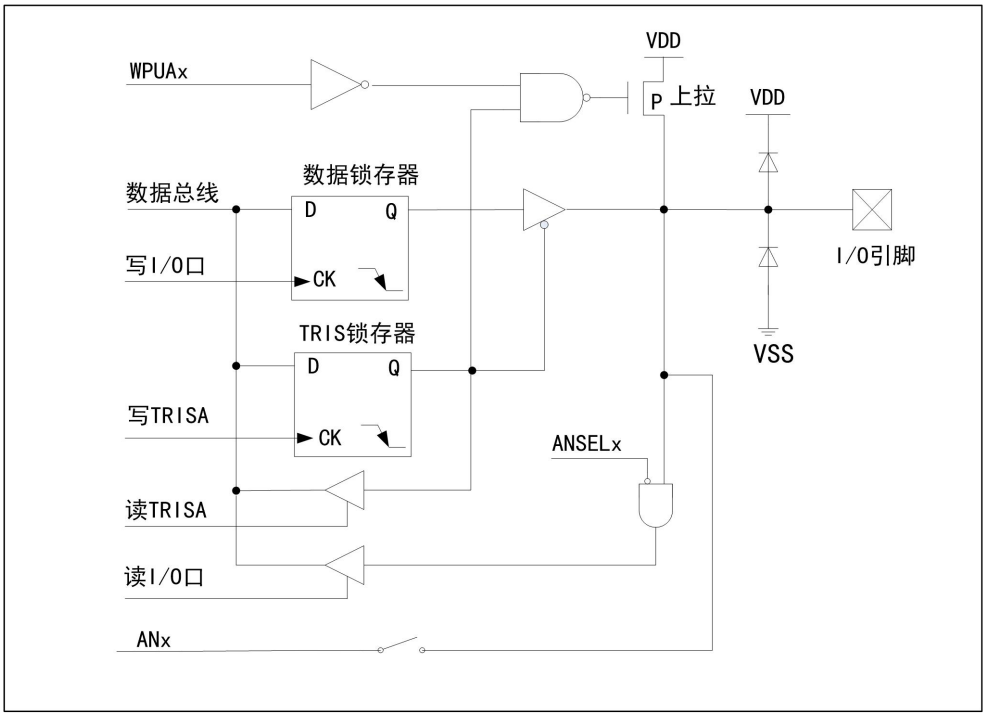
1.4 引脚描述

16PIN 脚位	名称	类型	说明
1	VDD	P	电源输入。
2	PORTB7 OSCI CLKI	I/O I I	输入/输出口，带可编程上拉电阻，端口电平变化中断。 高频晶体振荡器输入口。 外部时钟输入口。
3	PORTB6 OSCO	I/O O	输入/输出口，带可编程上拉电阻，端口电平变化中断。 高频晶体振荡器输出口。
4	PORTB5 MCLRB VPP T1G	I I P I	输入/开漏输出口，端口电平变化中断。 复位输入口，低电平有效。 编程高压电源输入。 输入口，门控信号输入。
5	PORTB3 CCP2	I/O I/O	输入/输出口，带可编程上拉电阻，端口电平变化中断。 捕捉/比较/PWM2输入/输出口。
6	PORTB2 CCP1	I/O I/O	输入/输出口，带可编程上拉电阻，端口电平变化中断。 捕捉/比较/PWM1输入/输出口。
7	PORTB1 T0CKI T1CKI LOSCI	I/O I I I	输入/输出口，带可编程上拉电阻，端口电平变化中断。 Timer0外部时钟输入口（施密特触发器）。 Timer1外部时钟输入口（施密特触发器）。 低频晶体振荡器输入口。
8	PORTB0 INT0 LOSCO	I/O I O	输入/输出口，带可编程上拉电阻，端口电平变化中断。 外部中断输入口（施密特触发器）。 低频晶体振荡器输出口。
9	PORTA7	I/O	输入/输出口，带可编程上拉电阻。
10	PORTA6	I/O	输入/输出口，带可编程上拉电阻。
11	PORTA4 AN4 V_{REF}	I/O AN AN	输入/输出口，带可编程上拉电阻。 ADC通道4输入口。 ADC参考电压输入口。
12	PORTA3 AN3	I/O AN	输入/输出口，带可编程上拉电阻。 ADC通道3输入口。
13	PORTA2 AN2 PCK	I/O AN O	输入/输出口，带可编程上拉电阻。 ADC通道2输入口。 编程内部RC输出口。
14	PORTA1 AN1 PGD	I/O AN I/O	输入/输出口，带可编程上拉电阻。 ADC通道1输入口。 编程数据输入/输出口。
15	PORTA0 AN0 PGC	I/O AN I	输入/输出口，带可编程上拉电阻。 ADC通道0输入口。 编程时钟输入口。
16	VSS	P	电源地。

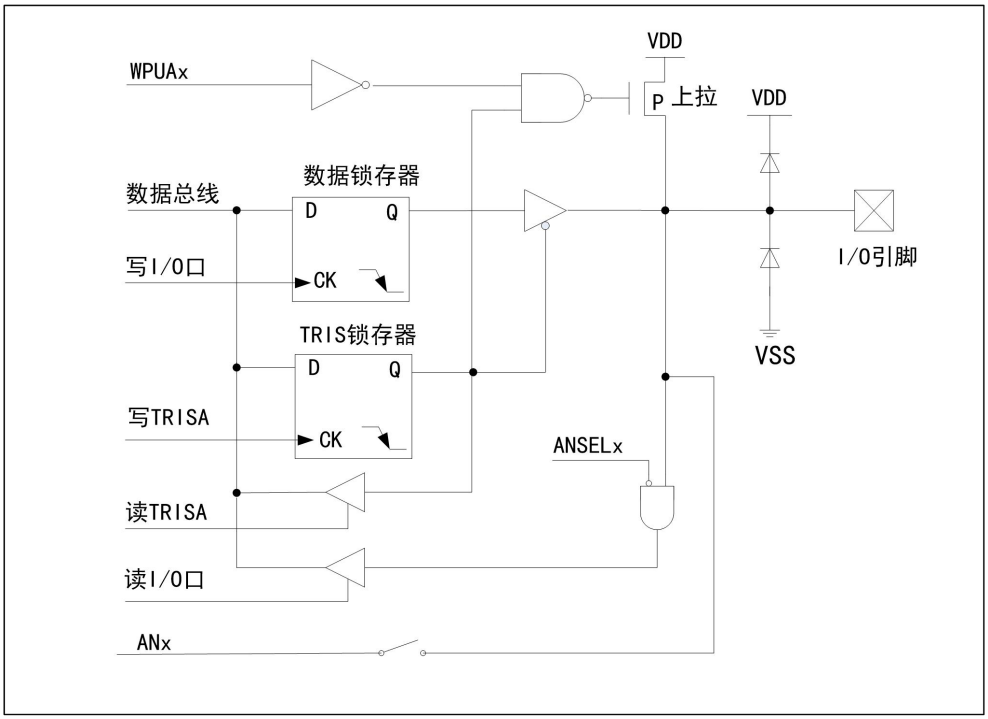
注： I = 输入　　O = 输出　　I/O = 输入/ 输出　　P = 电源　　AN = 模拟输入

1.5 引脚电路

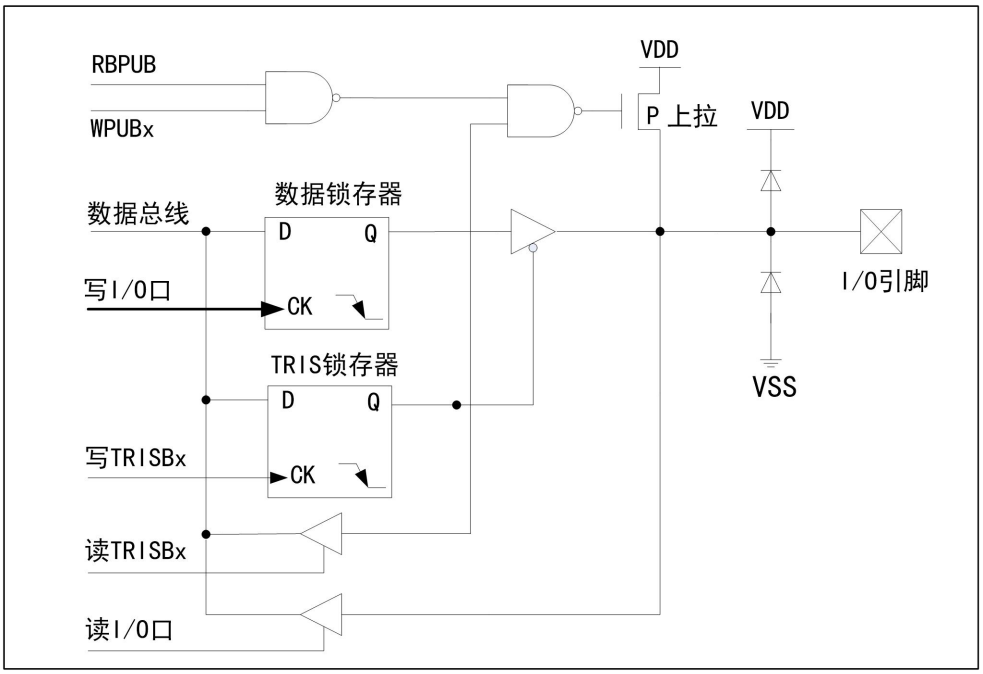
◆ PORTA [3:0]口的等效电路：



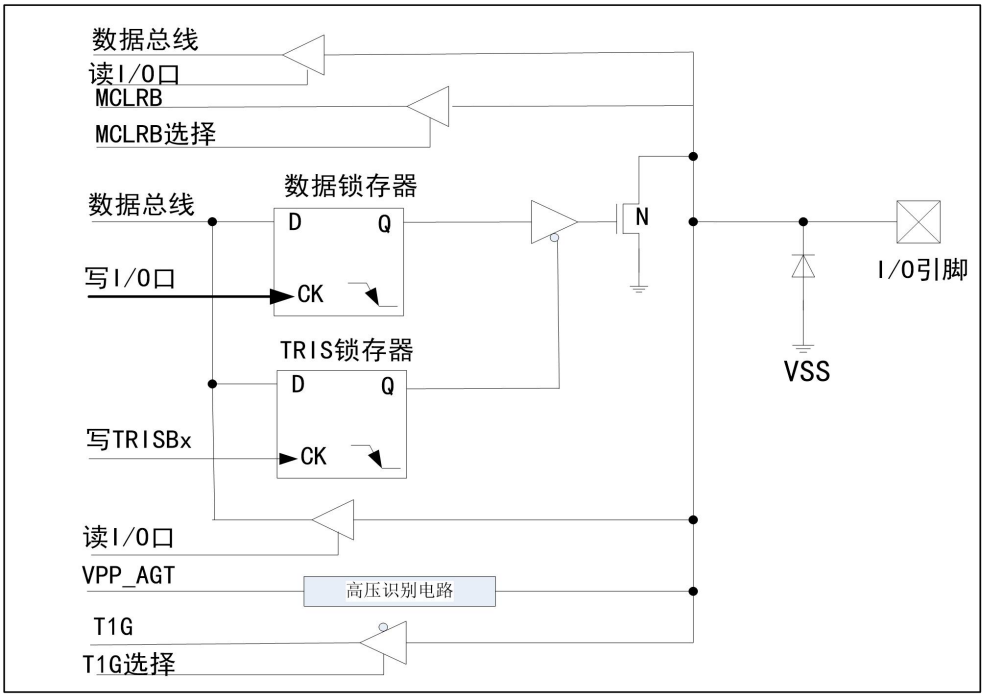
◆ PORTA4口的等效电路：



◆ PORTB[7:6][3:0]口的等效电路：



◆ PORTB5 口的等效电路：

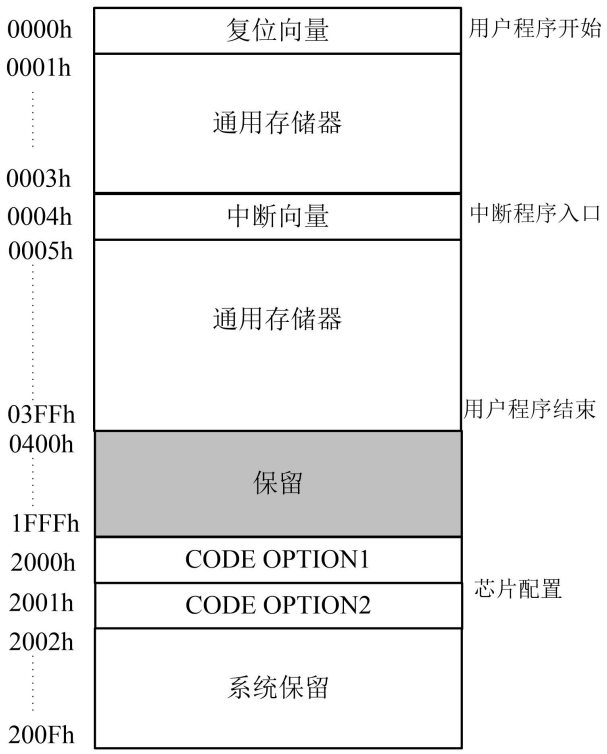


2 中央处理器（CPU）

2.1 存储器

2.1.1 程序存储器（OTP ROM）

✓ 用户程序空间：1K



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     100H      ;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:
                BSF         STATUS,RP0  ;BANK1
                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根据程序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**

```
W_TEMP          EQU         0X70
STATUS_TEMP     EQU         0X71
PCLATH_TEMP     EQU         0X72

                ...
                ORG         0004H
                MOVWF       W_TEMP          ;保护W寄存器
                SWAPF       STATUS,W
                MOVWF       STATUS_TEMP     ;保护STATUS寄存器
                MOVF        PCLATH,W
                MOVWF       PCLATH_TEMP     ;保护PCLATH寄存器
                CLRF        STATUS          ;切换到BANK0

                BTFSC       INTCON,INTF
```

```
                GOTO        ISR_INT0        ;发生INT0中断
                BTFSC       INTCON,T0
                GOTO        ISR_T0          ;发生TMR0溢出中断
                ...
                ...

INT_EXIT:       MOVF        PCLATH_TEMP,W
                MOVWF       PCLATH          ;恢复PCLATH寄存器
                SWAPF       STATUS_TEMP,W
                MOVWF       STATUS          ;恢复STATUS寄存器
                SWAPF       W_TEMP,F
                SWAPF       W_TEMP,W        ;恢复W寄存器
                RETFIE                      ;中断处理服务子程序返回
ISR_INT0:
                BCF         INTCON,INTF     ;外部中断处理
                ...
                GOTO        INT_EXIT
ISR_T0:
                BCF         INTCON,T0IF     ;TMR0中断处理
                ...
                GOTO        INT_EXIT
```

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、 保存系统寄存器时使用到的 RAM 建议定义在所有 BANK 均映射的位置。
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、 需要软件清空对应的中断标志。
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 2 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~bit13），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 PMDATLH:PMDATL 寄存器。

- **例：查找ROM 地址为“TABLE”的值**

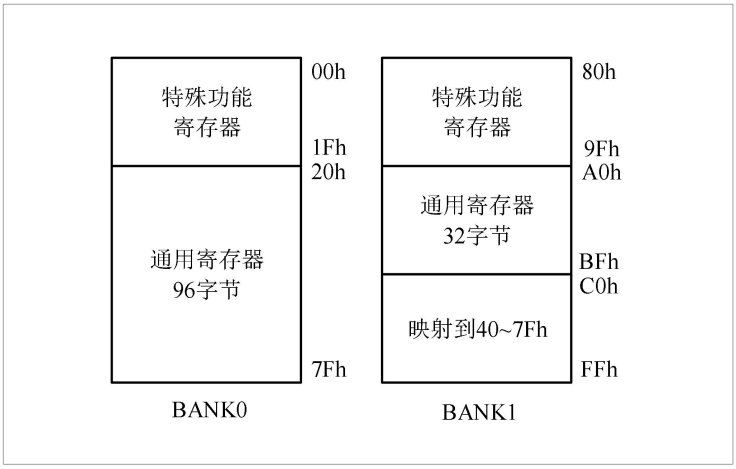
```
        BCF     STATUS,RP0          ;BANK0
        MOVF    TABLE _ADDR_H, W
        BSF     STATUS,RP0          ;BANK1
        MOVWF   PMADRH              ;设置TABLE地址高字节
        BCF     STATUS,RP0          ;BANK0
        MOVF    TABLE _ADDR_L, W
        BSF     STATUS,RP0           ;BANK1
        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
        BCF     STATUS,RP0          ;BANK0
        MOVWF   TABLE _DATA_L       ;TABLE _DATA_L= TABLE地址数据低字节
        BSF     STATUS,RP0          ;BANK1
        MOVF    PMDATH, W
        BCF     STATUS,RP0          ;BANK0
        MOVWF   TABLE _DATA_H       ;TABLE _DATA_H= TABLE
        ...
        ...

TABLE   DW      1234H               ;定义数据表（14 位）数据
        DW      3178H
        DW      2123H
```

2.1.2 通用数据存储器（RAM）

HC18P110A0/B0共有128个通用寄存器（GPR）和47个特殊功能寄存器（SFR），分在两个存储区Bank0和Bank1，每个存储区的低32个地址单元保留为特殊功能寄存器，RP0是存储区的选择位。

✓ ***RAM***



注：

将变量定义在0X40~0X7F，则可以在所有BANK访问，中断时保存系统寄存器的RAM建议定义在此。

2.1.3 特殊功能寄存器（SFR）

2.1.3.1 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
BANK0									
00h	INDF	间接寻址寄存器（不是实际存在的物理寄存器）							
01h	T0	Timer0 计数寄存器							
02h	PCL	程序计数器（PC）低字节							
03h	STATUS	-	-	RP0	TO	PD	Z	DC	C
04h	FSR	间接寻址地址指针							
05h	PORTA	PORTA7	PORTA6	-	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0
06h	PORTB	PORTB7	PORTB6	PORTB5	-	PORTB3	PORTB2	PORTB1	PORTB0
0Ah	PCLATH	-	-	-	-	-	-	程序计数器高 2 位	
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0Ch	PIR1	T1GIF	ADIF	-	-	-	CCP1IF	T2IF	T1IF
0Dh	PIR2	-	-	-	-	-	-	-	CCP2IF
0Eh	T1L	Timer1 计数寄存器低字节							
0Fh	T1H	Timer1 计数寄存器高字节							
10h	T1CON	T1CS1	T1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	-	T1ON
11h	T2	Timer2 计数寄存器							
12h	T2CON	-	T2CKPS3	T2CKPS2	T2CKPS1	T2CKPS0	T2ON	-	-
13h	WPUB	WPUB7	WPUB6	-	-	WPUB3	WPUB2	WPUB1	WPUB0
14h	IOCB	IOCB7	IOCB6	IOCB5	-	IOCB3	IOCB2	IOCB1	IOCB0
15h	CCPR1L	CCP1 寄存器低字节							
16h	CCPR1H	CCP1 寄存器高字节							
17h	CCP1CON	-	-	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0

18h	PR1L	Timer1 周期寄存器低字节							
19h	PR1CON	PWM1T1	PWM1T0	PWM2T1	PWM2T0	T1CKPS3	T1CKPS2	PWMPR1	PR1EN
1Bh	CCPR2L	CCP2 寄存器低字节							
1Ch	CCPR2H	CCP2 寄存器高字节							
1Dh	CCP2CON	-	-	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0
1Eh	ADRESH	ADC 结果寄存器高字节							
1Fh	ADCON0	-	VHS1	VHS2	CHS2	CHS1	CHS0	ADON	ADEN
BANK1									
80h	INDF	间接寻址寄存器（不是实际存在的物理寄存器）							
81h	OPTION	RBPUB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
82h	PCL	程序计数器（PC）低字节							
83h	STATUS	-	-	RP0	TO	PD	Z	DC	C
84h	FSR	间接寻址地址指针							
85h	TRISA	TRISA7	TRISA6	-	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
86h	TRISB	TRISB7	TRISB6	TRISB5	-	TRISB3	TRISB2	TRISB1	TRISB0
87h	ADCKCS						ADCKCS2	ADCKCS1	ADCKCS0
8Ah	PCLATH	-	-	-	-	-	-	程序计数器高 2 位	
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
8Ch	PIE1	T1GIE	ADIE	-	-	-	CCP1IE	T2IE	T1IE
8Dh	PIE2	-	-	-	-	-	-	-	CCP2IE
8Eh	PCON	LVD2EN	LVD1EN	-	WDTENS	LVD2F	LVD1F	POR	BOR
8Fh	T1GCON	T1GEN	T1GPOL	T1GTM	T1GSPM	T1GON	T1GVAL	T1GSS1	T1GSS0
90h	OSCCON	T0OSCEN	-	-	-	-	-	HXEN	SCS
92h	PR2	Timer2 周期寄存器							
93h	WPUA	WPUA7	WPUA6	-	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
94h	PMDATL	程序存储器读数据寄存器的低字节							
95h	PMDATH	-	-	程序存储器读数据寄存器的高字节					
96h	PMADRL	程序存储器读地址寄存器的低字节							
97h	PMADRH	-	-	-	-	程序存储器读地址寄存器的高字节			
98h	PMCON	-	-	-	-	-	-	-	RDON
9Dh	ANSEL	ANSEL7	ANSEL6	-	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0
9Eh	ADRESL	ADC 结果寄存器低字节							
9Fh	ADCON1	ADFM	ADCS2	ADCS1	ADCS0	-	-	-	ADREF

注： x = 未知， u = 保持与复位前不变， q = 取值视条件而定， - = 未实现

✓ *特殊寄存器复位初值*

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BANK0									
00h	INDF	x	x	x	x	x	x	x	x
01h	T0	x	x	x	x	x	x	x	x
02h	PCL	0	0	0	0	0	0	0	0

03h	STATUS	0	0	0	1	1	x	x	x
04h	FSR	x	x	x	x	x	x	x	x
05h	PORTA	x	x	-	x	x	x	x	x
06h	PORTB	x	x	x	-	x	x	x	x
0Ah	PCLATH	-	-	-	-	-	-	0	0
0Bh	INTCON	0	0	0	0	0	0	0	x
0Ch	PIR1	0	0	-	-	-	0	0	0
0Dh	PIR2	-	-	-	-	-	-	-	0
0Eh	T1L	x	x	x	x	x	x	x	x
0Fh	T1H	x	x	x	x	x	x	x	x
10h	T1CON	0	0	0	0	0	0	-	0
11h	T2	0	0	0	0	0	0	0	0
12h	T2CON	-	0	0	0	0	0	-	-
13h	WPUB	1	1	-	-	1	1	1	1
14h	IOCB	0	0	0	-	0	0	0	0
15h	CCPR1L	x	x	x	x	x	x	x	x
16h	CCPR1H	x	x	x	x	x	x	x	x
17h	CCP1CON	-	-	0	0	0	0	0	0
18h	PR1L	0	0	0	0	0	0	0	0
19h	PR1CON	0	0	0	0	0	0	0	0
1Bh	CCPR2L	x	x	x	x	x	x	x	x
1Ch	CCPR2H	x	x	x	x	x	x	x	x
1Dh	CCP2CON	-	-	0	0	0	0	0	0
1Eh	ADRESH	x	x	x	x	x	x	x	x
1Fh	ADCON0	-	-	0	0	0	0	0	0
BANK1									
80h	INDF	x	x	x	x	x	x	x	x
81h	OPTION	1	1	1	1	1	1	1	1
82h	PCL	0	0	0	0	0	0	0	0
83h	STATUS	0	0	0	1	1	x	x	x
84h	FSR	x	x	x	x	x	x	x	x
85h	TRISA	1	1	-	1	1	1	1	1
86h	TRISB	1	1	1	-	1	1	1	1
87h	ADCKS	-	-	-	-	-	1	1	1
8Ah	PCLATH	-	-	-	-	-	-	0	0
8Bh	INTCON	0	0	0	0	0	0	0	x
8Ch	PIE1	0	0	-	-	-	0	0	0
8Dh	PIE2	-	-	-	-	-	-	-	0
8Eh	PCON	0	0	-	1	q	q	q	q
8Fh	T1GCON	0	0	0	0	0	x	0	0
90h	OSCCON	0	-	-	-	-	-	0	q

92h	**PR2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
93h	**WPUA**	**1**	**1**	**-**	**1**	**1**	**1**	**1**	**1**
94h	**PMDATL**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
95h	**PMDATH**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
96h	**PMADRL**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
97h	**PMADRH**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
98h	**PMCON**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
9Dh	**ANSEL**	**1**	**1**	**-**	**1**	**1**	**1**	**1**	**1**
9Eh	**ADRESL**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
9Fh	**ADCON1**	**0**	**0**	**0**	**0**	**-**	**-**	**-**	**0**

注： x= 未知， u= 保持与复位前不变，， q= 取值视条件而定， -= 未实现

2.1.3.2 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
```

2.1.3.3 INDF寄存器

INDF寄存器不是实际存在的寄存器，寻址INDF将实现间接寻址。

2.1.3.4 程序计数器

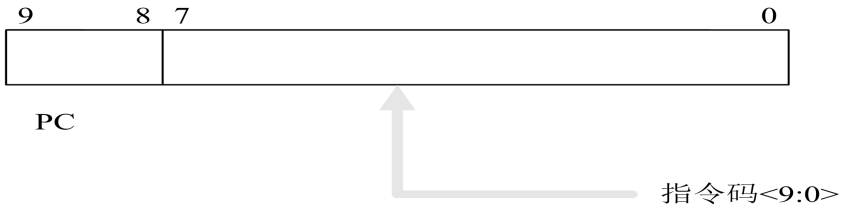
程序计数器（PC）为10位宽，低字节来自可读写的PCL寄存器，高字节（PC[9:8]）不可读写，可通过PCLATH寄存器间接写入。如果对PCL进行赋值，PCLATH也不会改变。

02h	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

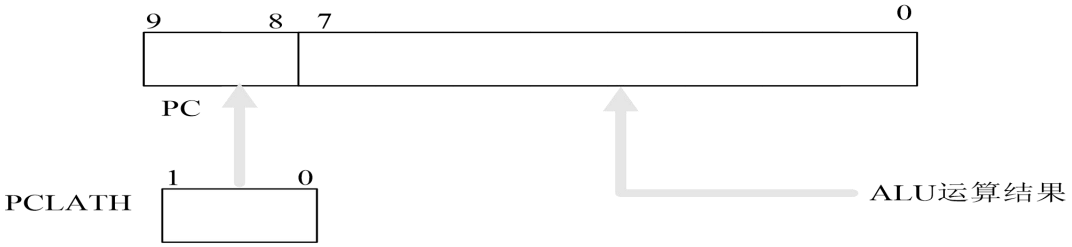
0Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCLATH	-	-	-	-	-	-	PCH9	PCH8
R/W	-	-	-	-	-	-	R/W	R/W
POR的值	-	-	-	-	-	-	0	0

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。

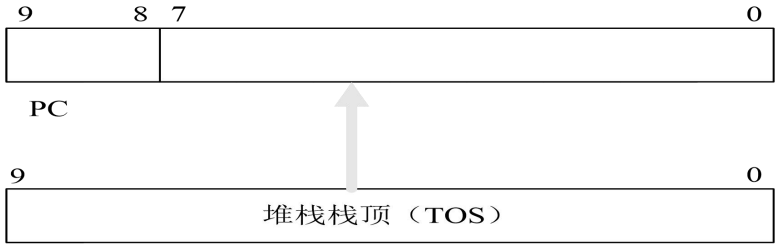
分支指令（GOTO/CALL）：



以PCL作为目的操作数的指令：



子程序返回指令（RETURN/RETLW/RETFIE）：



2.1.3.5 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。

03h、83h	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	0	0	1	1	x	x	x

注： x = 未知， - = 未实现

Bit [7:6] **保持置0，意外操作可能引起程序失控**

Bit [5] **RP0**:寄存器存储区选择位（**用于直接寻址**）

0 = Bank0（00h-7Fh）

1 = Bank1（80h-FFh）

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

2.1.3.6 PCON寄存器

8Eh	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**:3.7V低电压检测使能位

1 = 使能3.7V低电压检测

0 = 禁止3.7V低电压检测

Bit [6] **LVD1EN**:2.5V低电压检测使能位

1 = 使能2.5V低电压检测

0 = 禁止2.5V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**:3.7V低电压检测标志位

1 = 电压低于3.7V

0 = 电压高于3.7V

bit 2 **LVD1F**:2.5V低电压检测标志位

1 = 电压低于2.5V

0 = 电压高于2.5V

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

HC18P110A0/B0 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 指向的寄存器参与运算的寻址方式。INDF 寄存器不是物理寄存器，对 INDF 寄存器操作可以实现间接寻址

- 例：利用间接寻址对 20h～7Fh，A0h~0FFh 通用数据存储器进行清零

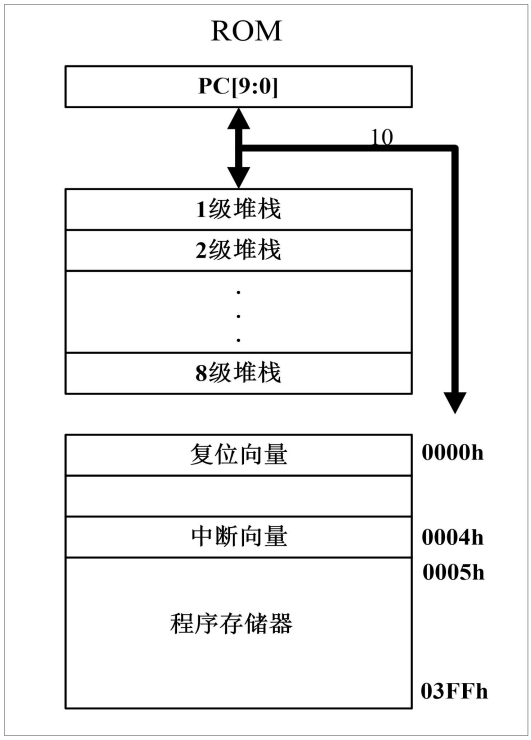
```
        MOVLW   20h     ;清零 0X20~0X7F
        MOVWF   FSR     ;FSR 指向 20h 地址
NEXTBYTE:
        CLRF    INDF    ;对 FSR 指向的数据存储器清零
        INCF    FSR,F   ;FSR + 1,指向下一个地址
        MOVLW   80h
        XORWF   FSR,W
        BTFSS   STATUS,Z
        GOTO    NEXTBYTE
```

```
                MOVLW   0A0h        ;清零 0XA0~0XBF
                                    ;0C0h~0FFh 映射到 BANK1，无需再清
                MOVWF   FSR         ;FSR 指向 A0h 地址
NEXTBYTE_1:
                CLRF    INDF        ;对 FSR 指向的数据存储器清零
                INCF    FSR,F       ;FSR + 1,指向下一个地址
                MOVLW   0C0h
                XORWF   FSR,W
                BTFSS   STATUS,Z
                GOTO    NEXTBYTE_1

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

2.3 堆栈

HC18P110A0/B0 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

HC18P110A0/B0 共有四种复位方式：

- 上电复位（POR）
- 外部复位（MCLRB Reset，仅在外部复位引脚处于使能状态）
- 欠压复位（BOR）
- 看门狗定时器复位（WDT Reset）

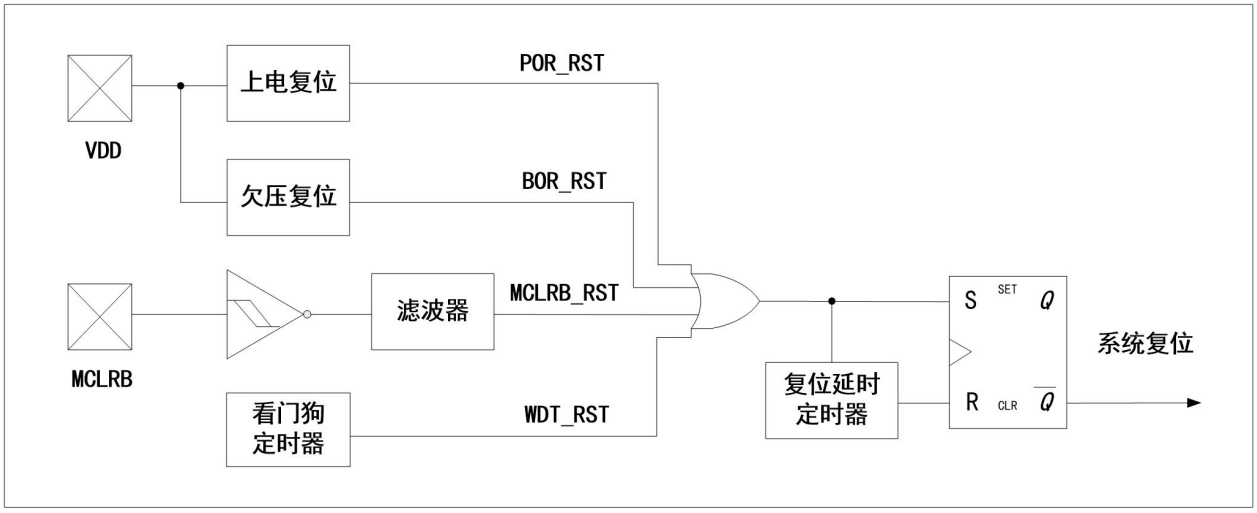
当上述任何一种复位产生时，系统进入复位状态，所有的特殊功能寄存器被初始化，程序停止运行，同时程序计数器（PC）清零。经过上电延时定时器延时后，系统结束复位状态，程序从 0000h 地址开始执行。STATUS 寄存器的 BIT4（TO 位）及 PCON 寄存器的 BIT0（BOR 位）、BIT1（POR 位）显示系统复位状态信息，可通过 3 个标志位判断复位来源，从而控制系统的运行路径。

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。

注： u＝ 保持与复位前不变， x＝ 未知

复位电路示意图：



复位延时定时器在复位信号结束后，提供一定时间的延时

说明	复位延时定时器时间（典型值）
上电复位	18ms/4.5ms（配置字选择）
欠压复位	18ms/4.5ms（配置字选择）
外部复位	2.2ms/1.1ms（配置字选择）
看门狗定时器复位	2.2ms/1.1ms（配置字选择）

特殊功能寄存器复位状态：

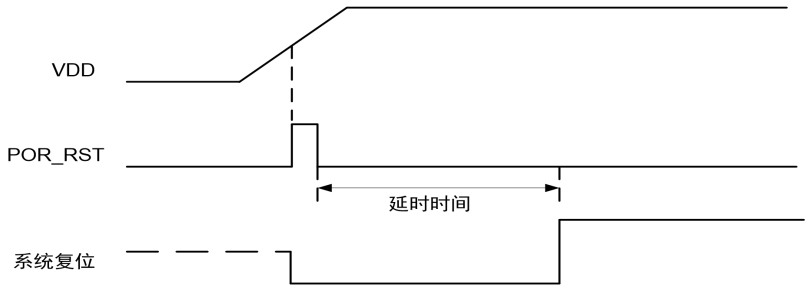
复位方式	STATUS寄存器	PCON寄存器
上电复位	0001 1xxx	00-1 qq00
正常工作模式下的外部复位	0001 1xxx	00-1 qq0u
休眠模式下的外部复位	0001 0uuu	00-1 qquu
欠压复位	0001 0uuu	00-1 qqu0
看门狗定时器复位	0000 1uuu	00-1 qquu

注： u＝ 保持与复位前不变， 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 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 震荡器的起振时间最短，晶体震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。

上电复位示意图：



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。

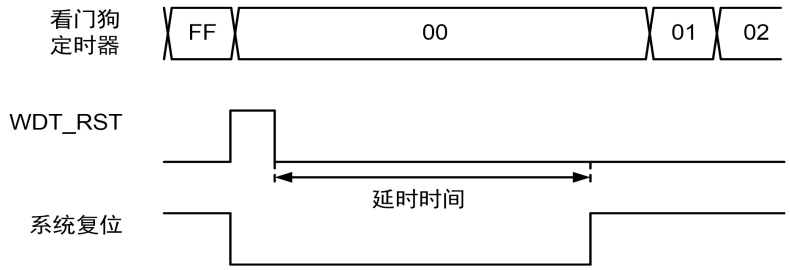
2、 VDD 上电斜率必须满足大于 500mV/ms，否则 POR_RST 信号可能无法产生。

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：

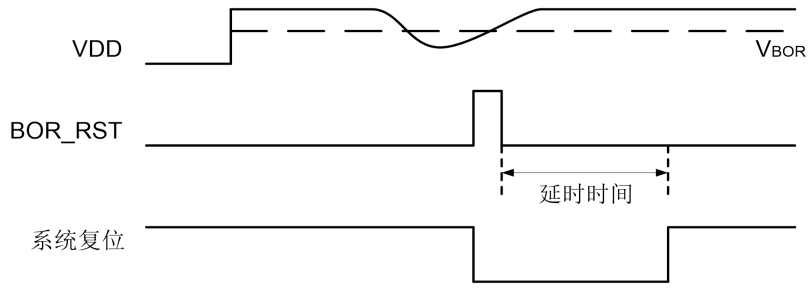
1、 看门狗的使能逻辑：看门狗使能 = 看门狗配置字使能 & 看门狗软件使能（WDTENS=1）。
2、 不能在中断程序中对看门狗进行清零，否则无法监控主程序跑飞情况。
3、 程序中应该只在主程序中有一次清看门狗的动作，这种架构能够最大限度的发挥看门狗的保护功能。
4、 看门狗复位的延时时间为 2.2ms/1.1ms。
5、 使用时注意：不论哪种方式复位后，看门狗软件使能位（WDTENS）的值为 1。

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：



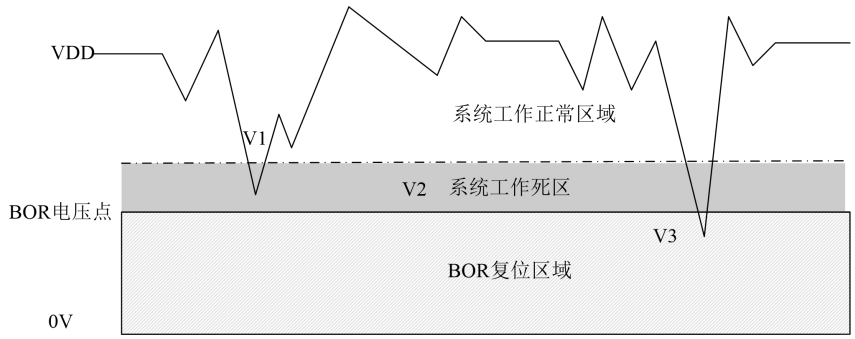
低电压复位（BOR）是单片机内置的掉电复位保护装置，当VDD 跌落并低于BOR 检测电压值时，BOR被触发，系统复位。不同的单片机有不同的BOR 检测电平，BOR 检测电平值仅为个电压点，并不能覆盖所有死区范围。因此采用BOR 依赖于系统要求和环境状况。如果电源跌落剧烈，远低于BOR 触发点，BOR 能够起到保护作用，让系统正常复位；如果电源电压跌落不是很剧烈，仅仅是接近BOR 触发点而造成的系统错误，则BOR 就不能起到保护作用让系统复位。

HC18P110A0/B0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：BOR3.7V/2.8V/2.5V/2.2V/2.0V/1.8V/1.5V

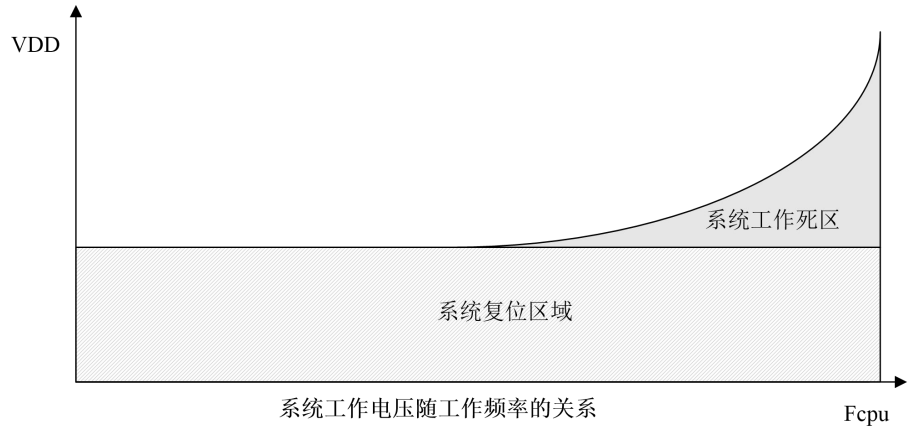
3.4.2 工作死区

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）

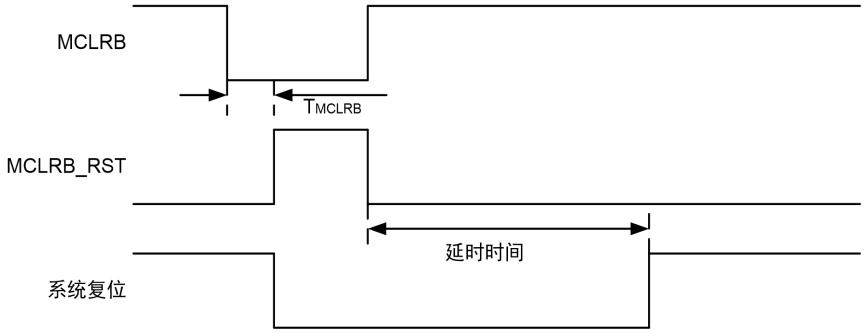
注：

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 **TMCLRB** 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：



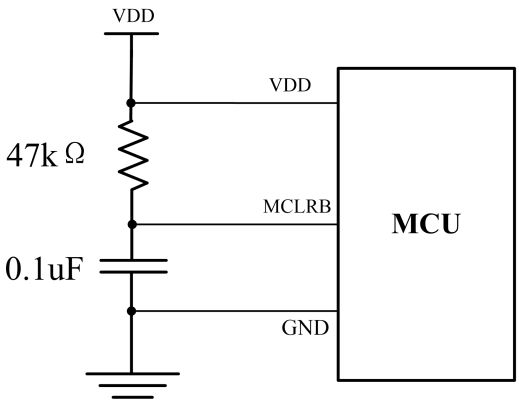
注：

TMCLRB 需大于 200uS（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

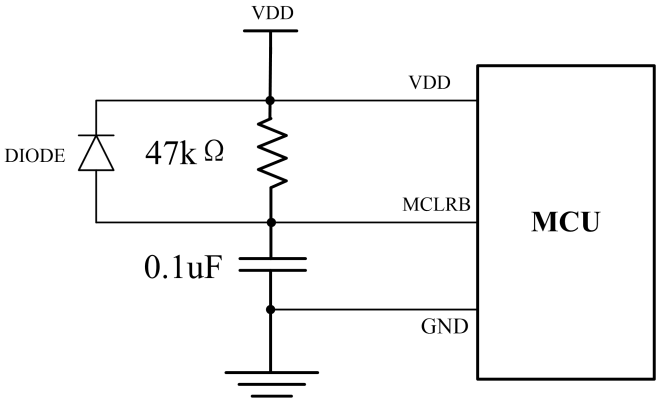
由电阻和电容组成的基本RC 复位电路，它在系统上电的过程中能够为复位引脚提供一个缓慢上升的复位信号。这个复位信号的上升速度低于VDD 的上电速度，为系统提供合理的复位时序，当复位引脚检测到高电平时，系统复位结束，进入正常工作状态。

如下图：



3.5.2 二极管 RC 复位电路

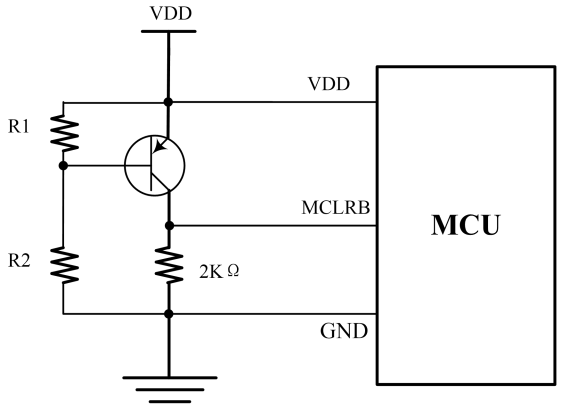
在基本RC复位电路上增加一个二极管（DIODE），对于电源异常情况，二极管正向导通使电容快速放电并与VDD保持一致，避免复位引脚持续高电平，系统无法正常复位。



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。



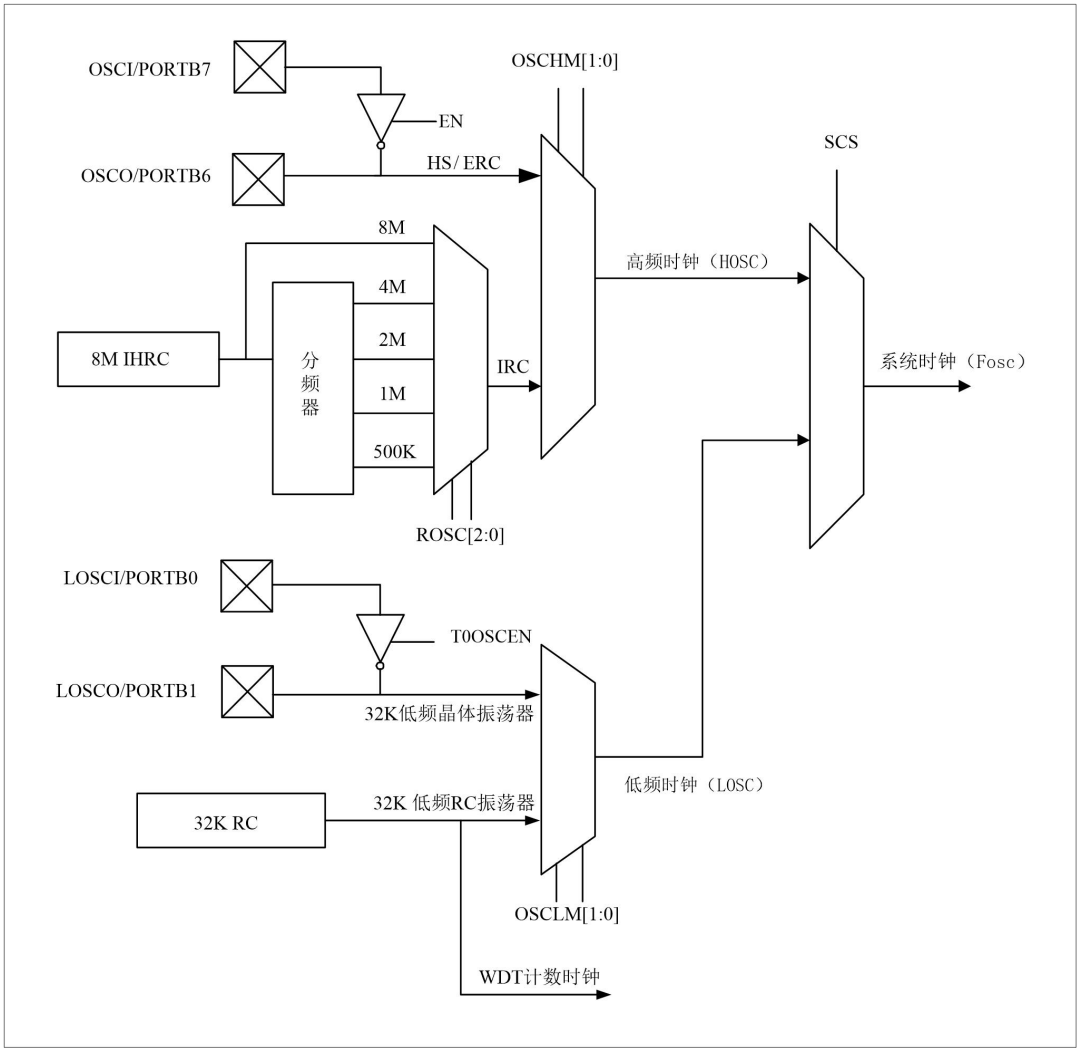
4 系统时钟

4.1 概述

HC18P110A0/B0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8MHz RC 振荡电路（IRC 8MHz）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：系统时钟频率

4.3 系统高频时钟

系统高频时钟有三种选择，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。 外部时钟输入，OSCI 作为外部时钟输入口，OSCO 作为外部时钟输出口。
10	外部 RC 振荡器（ERC），OSCI 作为外部 RC 震荡输入口。

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8M/4M/2M/1M /500K 五种选择。

高频内部RC振荡器频率选择配置字：

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

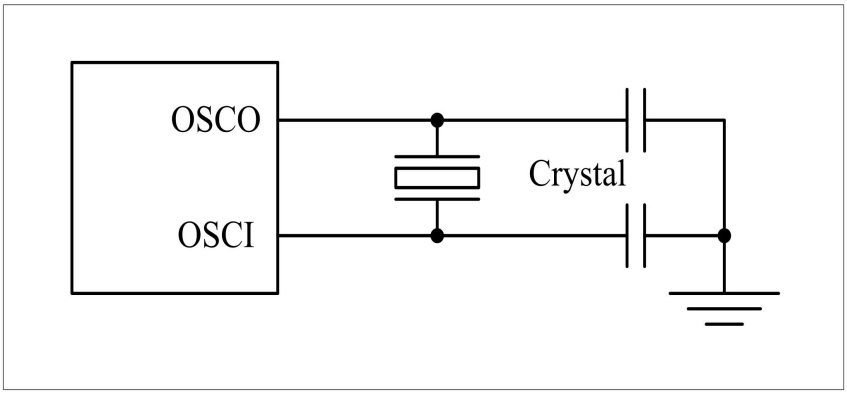
4.3.2 外部高频时钟

外部高频时钟共三种模式，由配置字 OSCHM 控制具体模式的选择：

- 高频晶体振荡器：最高 20MHz
- 外部 RC 振荡器
- 外部时钟输入

4.3.2.1 高频晶体振荡器

高频晶体振荡器的频率为400K~20MHz，推荐的典型值为4MHz、8MHz和16MHz，电容推荐值为20pF。



注：

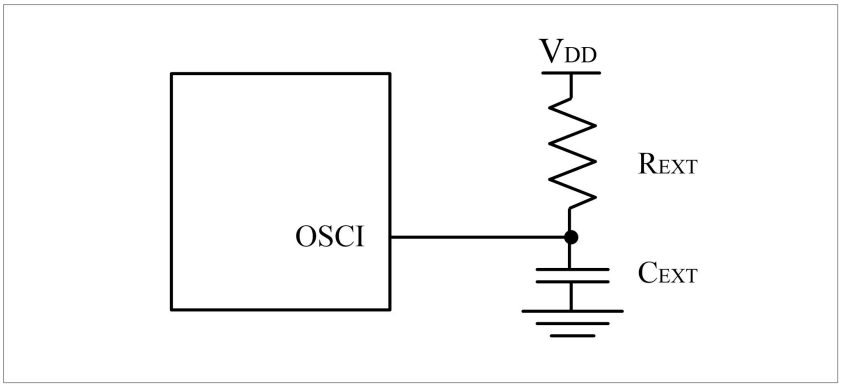
OSCI 和 OSCO 引脚与振荡器和起振电容之间距离 10mm 以内。

4.3.2.2 外部RC振荡器

外部RC振荡器的频率随电源电压、电阻（R_{EXT}）和电容（C_{EXT}）以及工作温度变化而变化。如果高频时钟选择为外部RC振荡器，OSCO引脚为通用I/O口。

建议值： 3 kΩ ≤ R_{EXT} ≤ 100 kΩ

C_{EXT} > 20pF



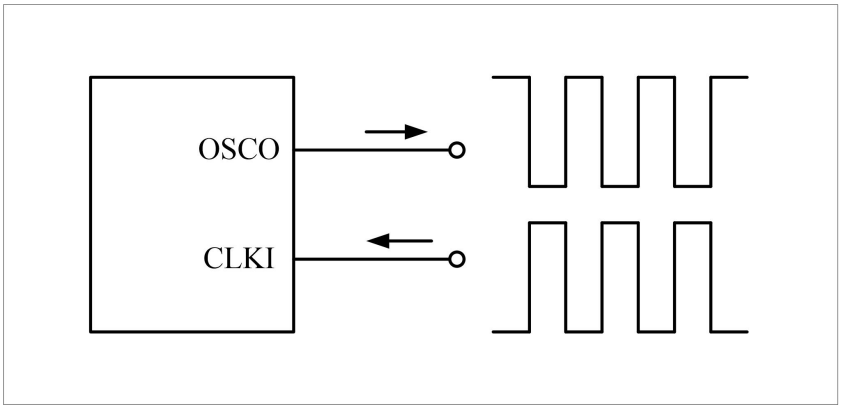
注：

Rext 和 Cext 尽量靠近单片机的 OSCI 引脚（10mm 以内）。

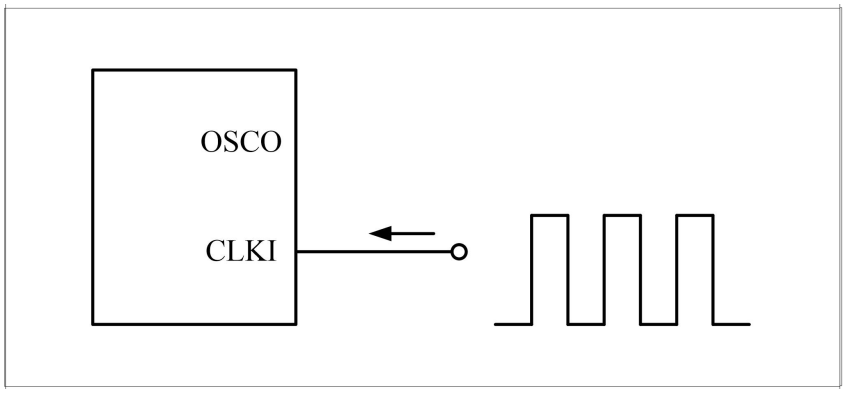
4.3.2.3 外部时钟源

单片机可选择外部时钟信号作为系统时钟，由配置字OSCHM控制，从OSCI 脚送入。

如果配置字OSCHM选择为高频晶体振荡器，外部时钟由CLKI引脚输入，OSCO引脚为CLKI的反向输出。



如果配置字OSCHM选择为外部RC振荡器，外部时钟由CLKI引脚输入，OSCO引脚为通用I/O口。



注：
外部振荡电路中的 GND 必须尽可能的接近单片机的 VSS 端口。

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： 32K（5V 典型值）

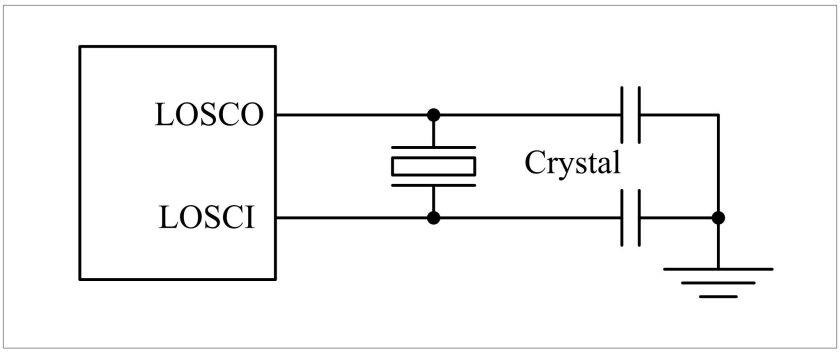
低频系统时钟选择配置字：

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：



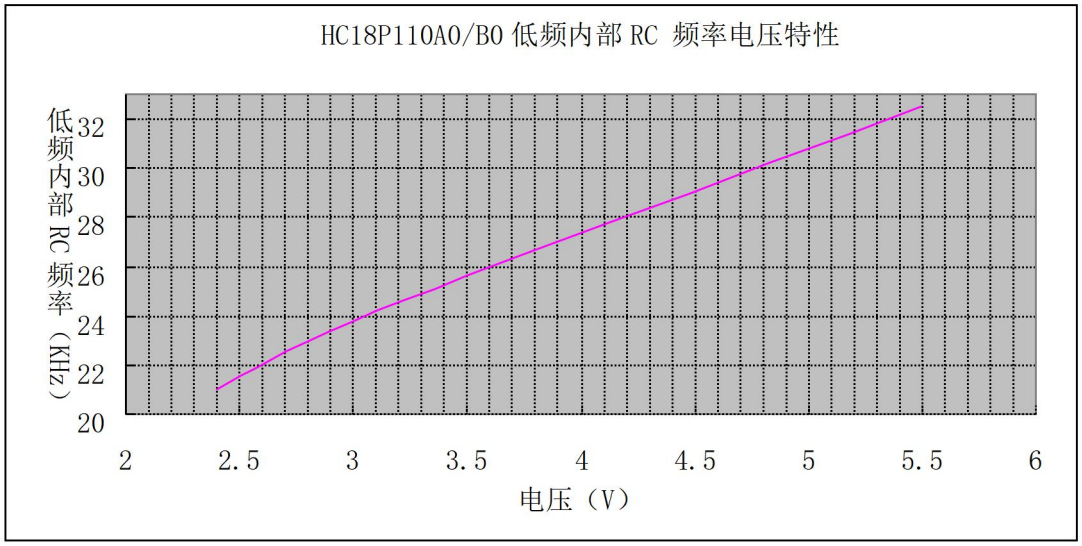
系统工作在绿色模式下，可以使能低频晶体振荡器。

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。

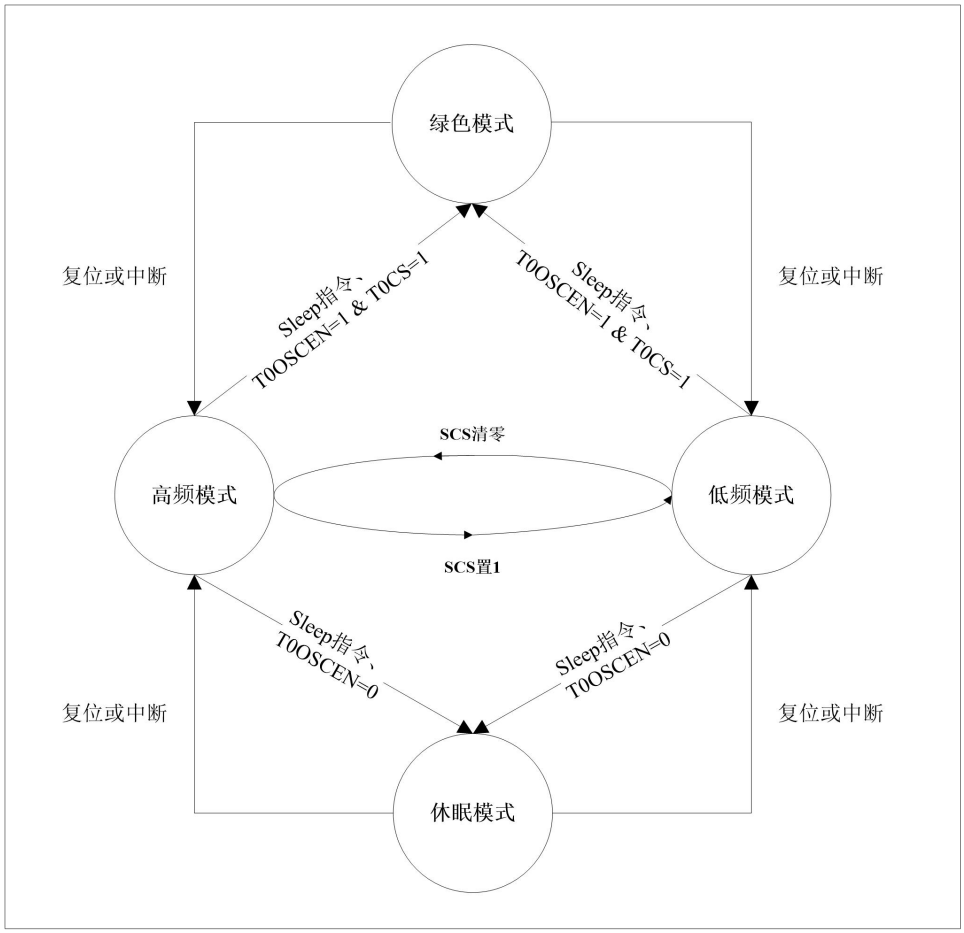
5 系统工作模式

5.1 概述

HC18P110A0/B0可在如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、 系统不论从低速或者高速模式进入绿色模式，唤醒后，返回到高速模式。

3、 外部复位和 Timer2 中断不能唤醒休眠或绿色模式。

4、 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

5.2 模式切换举例

- 例：高频/低频模式切换到睡眠模式。

```
BSF     STATUS,RP0
BCF     OSCCON,T0OSCEN
SLEEP
```

- 例：高频模式切换到低频模式.

```
BSF     STATUS,RP0          ;BANK1
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。

```
BSF     STATUS,RP0          ;BANK1
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

```
MOVLW   0X05
BSF     STATUS,RP0          ;BANK1
MOVWF   OPTION
BSF     OPTION,T0CS
BSF     OSCCON, T0OSCEN
BCF     STATUS,RP0          ;BANK0
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。

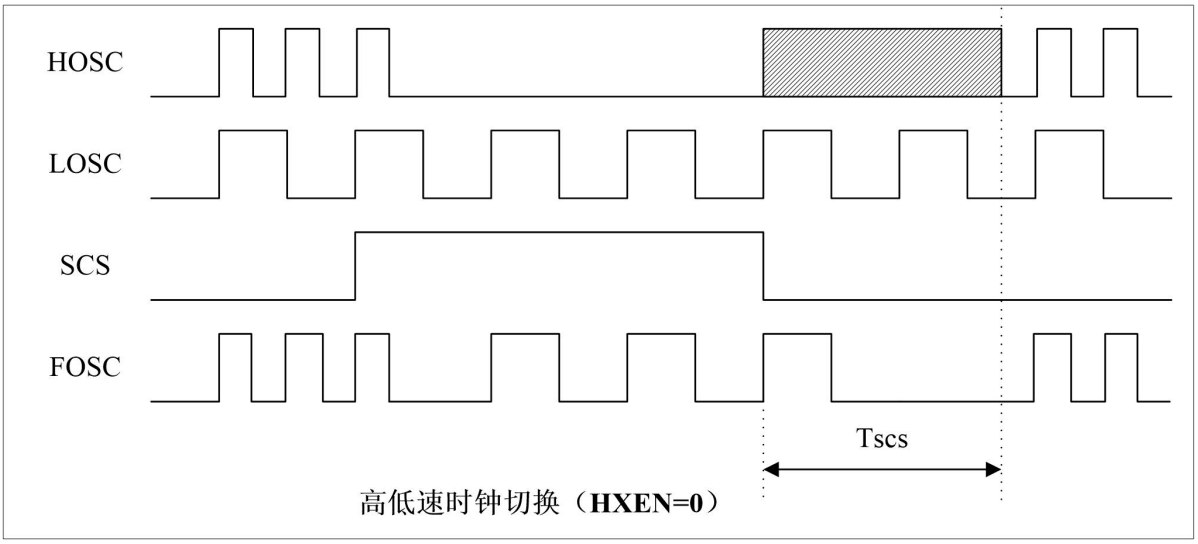
```
MOVLW   0X05
BSF     STATUS,RP0          ;BANK1
MOVWF   OPTION
BSF     OPTION,T0CS
BSF     OSCCON,T0OSCEN
BCF     OSCCON,T0IF
BCF     STATUS,RP0          ;BANK0
BSF     INTCON,T0IE         ;使能T0定时器
BSF     INTCON,GIE
```

```
        CLRF    T0

RTC_MODE:
        SLEEP
        BCF     STATUS,RP0      ;BANK0
        BCF     INTCON,T0IF     ;0.5s时间到
        …
        GOTO    RTC_MODE
```

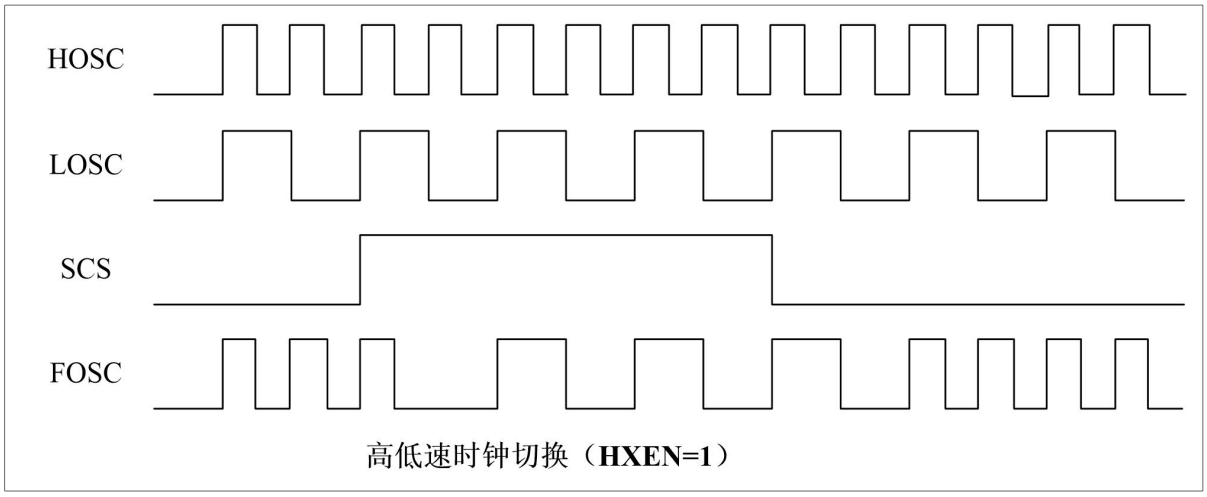
5.3 高低频时钟切换

高低频切换时序：



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

不同类型振荡器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

90h	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

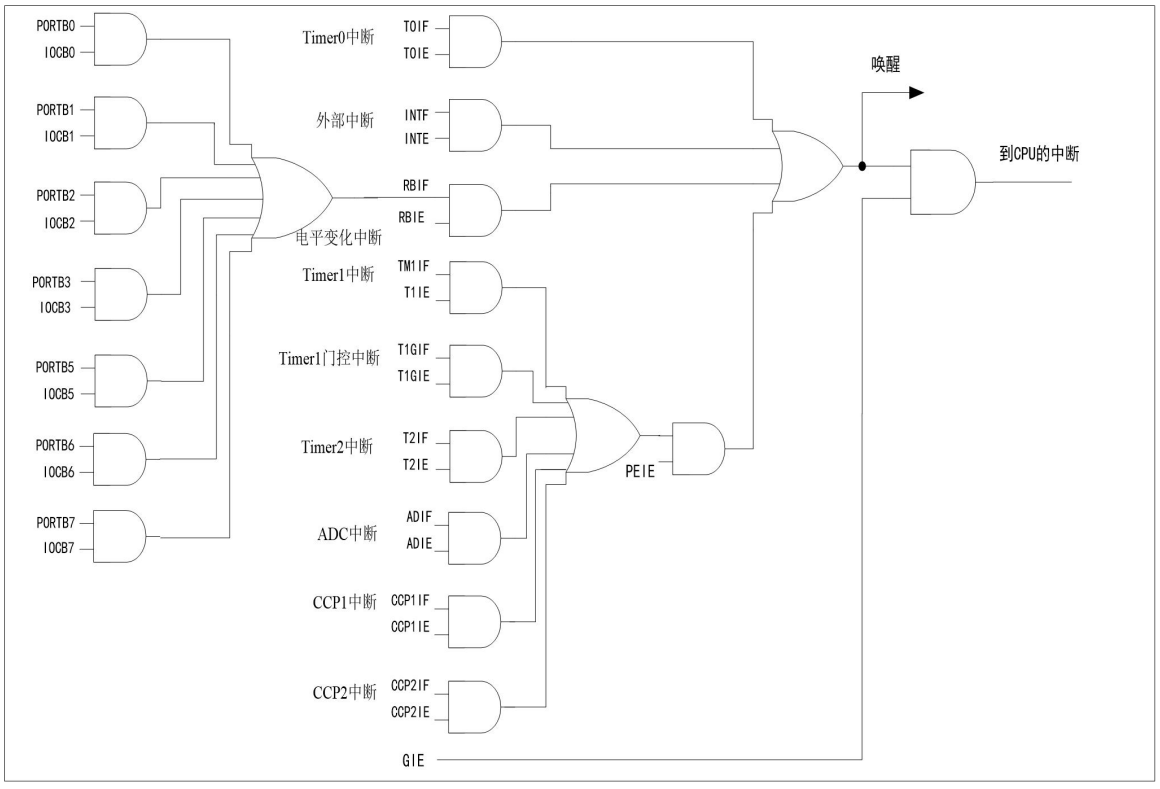
HC18P110A0/B0共有九个中断源:

- Timer0定时器中断
- INT0外部中断
- PORTB电平变化中断
- Timer1定时器中断
- Timer2定时器中断
- ADC中断
- Timer1门控中断
- CCP1中断
- CCP2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。在中断内部，GIE被除能，避免中断嵌套，程序运行到RETFIE指令时，系统退出中断服务程序并自动使能GIE，程序计数器值出栈，系统执行PC+1地址对应的指令。中断不会自动清除中断标志，需要软件清除。

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。

0Bh、8Bh	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0）

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0）

0 = 未发生INT0外部中断

Bit [0] **RBIF**:PORTB电平变化中断标志位

1 = PORTB[7:5] &PORTB[3:0]中至少有一个端口的电平状态发生了改变（必须由软件清0）

0 = PORTB[7:5] &PORTB[3:0]电平状态没有变化

81h	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

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。ADC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。

注：

PEIE 控制的外设中断包括 T1、T2、T2G、AD、CCPx 中断，欲开启此 4 类中断，除允许相应的 IE，还需要使能 PEIE 及 GIE。

0Bh、8Bh	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

8Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE1	T1GIE	ADIE	-	-	-	CCP1IE	T2IE	T1IE
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
POR的值	0	0	-	-	-	0	0	0

Bit [7] **T1GIE**:Timer1门控中断使能位

1 = 使能Timer1门控中断

0 = 禁止Timer1门控中断

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

8Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIE2	-	-	-	-	-	-	-	CCP2IE
R/W	-	-	-	-	-	-	-	R/W
POR的值	-	-	-	-	-	-	-	0

Bit [0] **CCP2IE**:CCP2中断使能位

1 = 使能CCP2中断

0 = 禁止CCP2中断

0Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR1	T1GIF	ADIF	-	-	-	CCP1IF	T2IF	T1IF
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
POR的值	0	0	-	-	-	0	0	0

Bit [7] **T1GIF**:Timer1门控中断标志位

1 = Timer1门控产生中断

0 = Timer1门控未产生中断

Bit [6] **ADIF**:ADC中断标志位

1 = AD转换已完成（必须用软件清零）

0 = AD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0）

0 = Timer1计数寄存器未溢出

0Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR2	-	-	-	-	-	-	-	CCP2IF
R/W	-	-	-	-	-	-	-	R/W
POR的值	-	-	-	-	-	-	-	0

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

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 都必须处于使能状态。

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、多 BANK 的 IC 中，为了使保存系统寄存器的 RAM 可以在所有 BANK 访问，建议将这些 RAM 指定在所有 BANK 均映射的地址，该 IC 为 0X40~0X7F。

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_TEM ,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
        BSF     STATUS,RP0          ;BANK1
        MOVWF   OPTION              ;T0时钟＝Fcpu / 64
        MOVLW   0X40                ;T0初始值＝64H
        BCF     STATUS,RP0          ;BANK0
        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_TEM ,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。

81h	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。

```
        BSF     STATUS,RP0          ;BANK1
        BSF     OPTION,INTEG        ;INT0置为上升沿触发
        BCF     STATUS,RP0          ;BANK0
        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
        BCF     STATUS,RP0          ;BANK0
        BTFSS   INTCON,INTF         ;检测T0IF
        GOTO    EXIT_INT            ;T0IF＝0，退出中断
        BCF     INTCON,INTF         ;T0IF清零
        …                           ;INT0中断服务程序
        …
EXIT_INT:
        …                           ;恢复STATUS、W和PCLATH
        RETFIE                      ;退出中断
```

6.7 PORTB 电平变化中断

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

电平变化中断必须将PORTB端口设为输入，并将寄存器IOCB对应位置1。

16h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOCB	IOCB7	IOCB6	IOCB5	-	IOCB3	IOCB2	IOCB1	IOCB0

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

IOCB[7:0]: PORTBx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

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。

➢ 例：PORTB1 电平变化中断请求设置。

```
        MOVLW   0X02
        BSF     STATUS,RP0          ;BANK1
        IORWF   TRISB,F             ;PORTB1端口为输入
        MOVLW   0X02
        BCF     STATUS,RP0          ;BANK0
        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。

```
        MOVLW   0X02
        BSF     STATUS,RP0          ;BANK1
        IORWF   TRISB,F             ;PORTB1端口为输入
        MOVLW   0X02
        BCF     STATUS,RP0          ;BANK0
```

```
        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
        BCF     STATUS,RP0      ;BANK0
        MOVF    PORTB,W         ;读PORTB端口
        …                       ;其他程序
```

注：

PORTB 电平变化唤醒，需在 SLEEP 指令后执行 PORTB 端口读操作。

6.8 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

```
        MOVLW   0XFF
        BSF     STATUS,RP0      ;BANK1
        MOVWF   PR2             ;设置T2周期
        MOVLW   0X04
        BCF     STATUS,RP0      ;BANK0
        MOVWF   T2CON           ;设置分频比
        CLRF    T2
        BSF     STATUS,RP0      ;BANK1
        BSF     PIE1,T2IE       ;使能TIMER2中断
        BCF     STATUS,RP0      ;BANK0
        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
```

6.9 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
        MOVLW   nnH
        MOVWF   T1H
        MOVLW   nnH
        MOVWF   T1L                     ;Timer1赋初值
        MOVLW   0XC0
        MOVWF   INTCON                  ;使能外设中断
        BSF     STATUS,RP0              ;BANK1
        BSF     PIE1,T1IE
        BCF     STATUS,RP0
        BCF     PIR1,T1IF
        BSF     T1CON,T1ON
        BSF     STATUS,RP0
        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
```

6.10 TIMER1 门控中断

当发生T1门控中断时，无论T1GIE 处于何种状态，T1GIF都会置1。若T1GIE 和T1GIF 都置1，

且PEIE、GIE均使能，系统就会响应TIMER1门控中断；若T1GIE = 0，则无论T1GIF 是否置1，系统都不会响应TIMER1门控中断。

6.11 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。

6.12 CCP1 中断和 CCP2 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。

6.13 多中断操作举例

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
T2IF	T2的值和PR2相同

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
            BSF     STATUS,RP0              ;BANK1
            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
            GOTO    INT_EXIT                ;跳到中断结束
            BTFSC   INTCON,RBIF
            GOTO    INTRB                   ;进入PORTB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

HC18P110A0/B0共有两个I/O端口：

- PORTA口
- PORTB口

7.1 I/O 口模式

85h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISA	TRISA7	TRISA6	-	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
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

86h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISB	TRISB7	TRISB6	TRISB5	-	TRISB3	TRISB2	TRISB1	TRISB0
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 [7:0] **PORTx[7:0]**的输入输出控制位

1 =输入状态

0 =输出状态

9Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSEL	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

Bit [7:5]: [4:0] ：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：

ANEL 上电初始值为 B’ 11x1 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。

7.2 I/O 口上拉电阻

93h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WPUA	WPUA7	WPUA6	-	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
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

13h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WPUB	WPUB7	WPUB6	-	-	WPUB3	WPUB2	WPUB1	WPUB0
R/W	R/W	R/W	-	-	R/W	R/W	R/W	R/W
POR的值	1	1	-	-	1	1	1	1

WPUx[7:0] PORTx[7:0]的上拉使能位

1 = 上拉禁止

0 = 上拉使能

81h	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、 注意此处为 PORTB 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 口数据寄存器

05h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTA	PORTA7	PORTA6	-	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0
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

06h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTB	PORTB7	PORTB6	PORTB5	-	PORTB3	PORTB2	PORTB1	PORTB0
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

注：

PORTB5 端口为 OD 端口，可正常输出低电平。无法正常输出高电平，需要外接上拉电阻！

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

```
……                              ;其他初始化
BSF     STATUS,RP0              ;BANK1
MOVLW   B'00000001'             ;PA0为模拟信号口，其他为数字信号口
MOVWF   ANSEL
MOVLW   H'FF'
MOVWF   TRISA                   ;PA口作为输入口
```

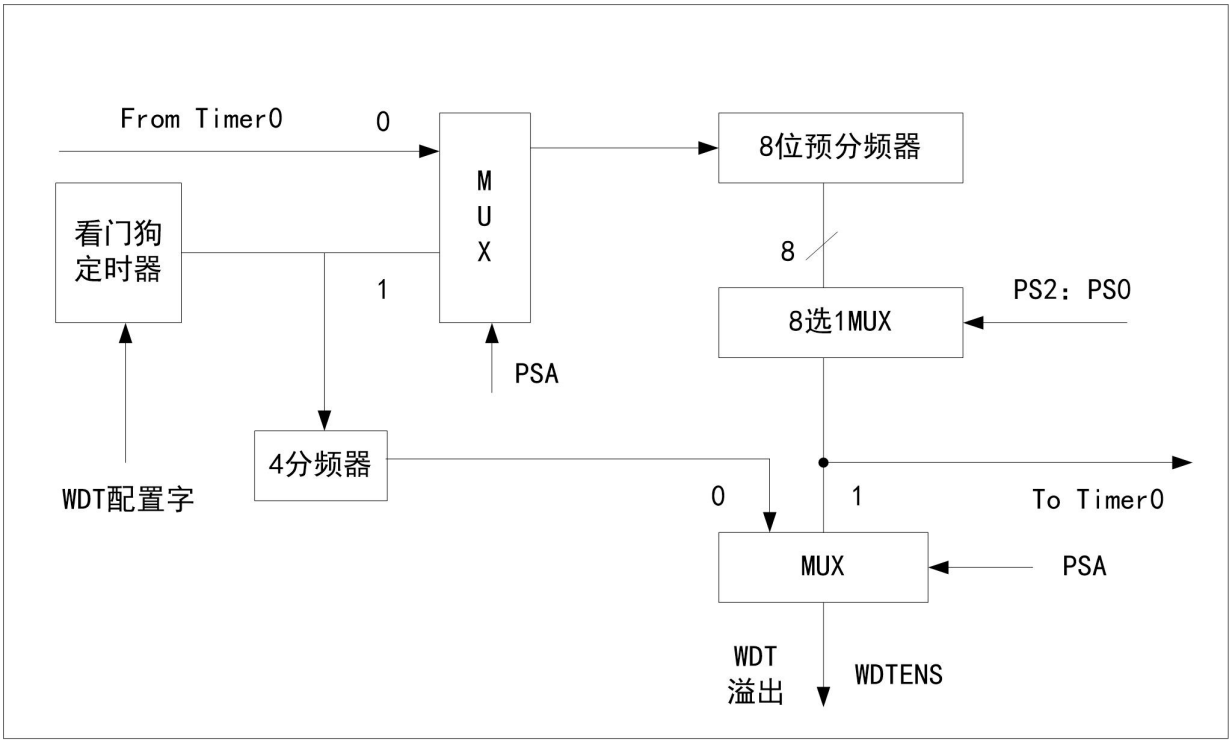
```
        BCF     STATUS,RP0      ;BANK0
        CLRF    PORTA           ;清空PORTA
        BSF     STATUS,RP0      ;BANK1
        MOVLW   H’00’
        MOVWF   TRISB           ;设置PB口为输出
        BCF     STATUS,RP0      ;BANK0
        MOVLW   H’FF’
        MOVWF   PORTB           :PB口输出高电平
TEST1：                         ;下面访问PORTA作为输入检测得到的数据
        BCF     STATUS,RP0      ;BANK0
        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
```

8 定时器

8.1 看门狗定时器 WDT

HC18P110A0/B0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Timer0定时器/计数器。

看门狗定时器和预分频器框图：



8Eh	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：

```
BSF     STATUS,RP0      ;BANK1
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
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

```
…
BSF     STATUS,RP0      ;BANK1
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

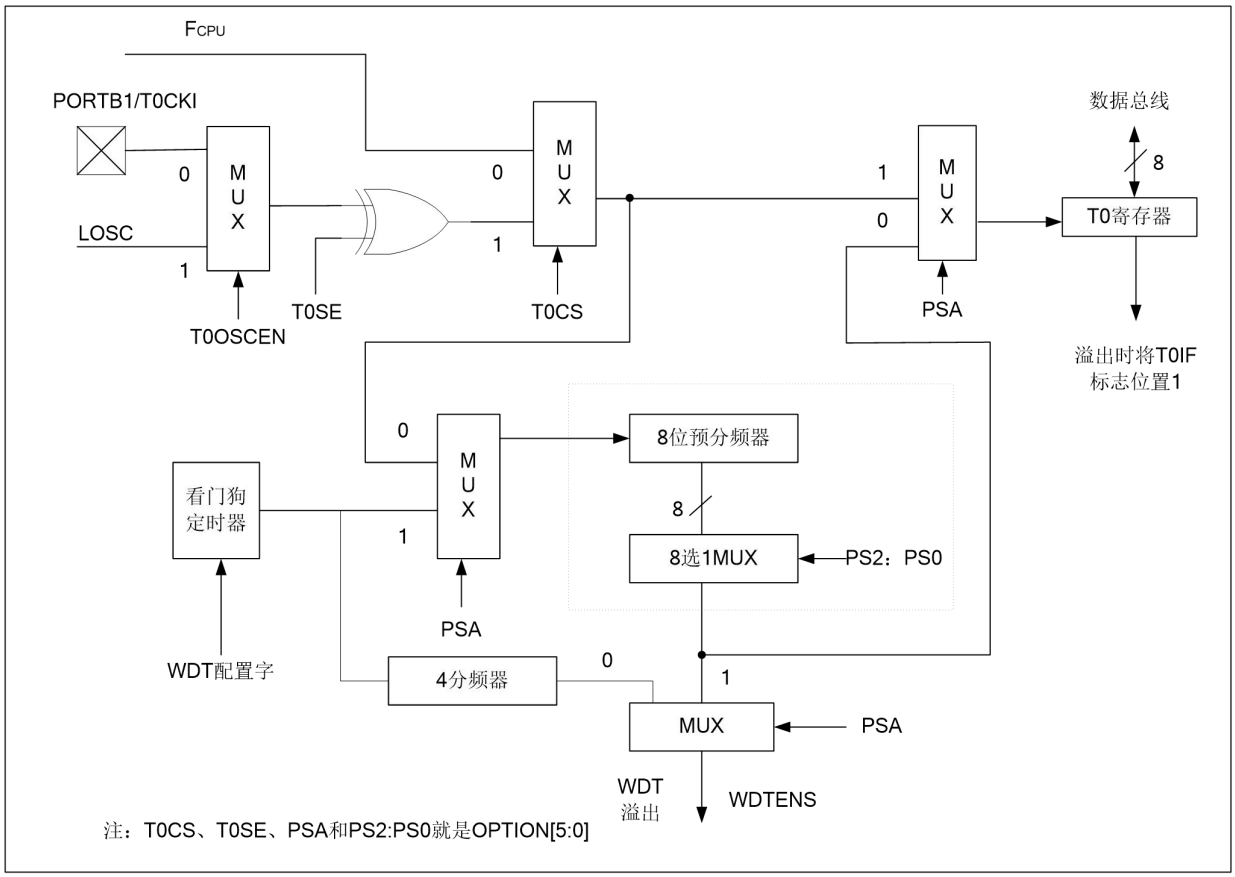
```
BSF     STATUS,RP0      ;BANK1
BSF     PCON,WDTENS     ;使能看门狗
CLRWDT                  ;看门狗定时器清零
…
```

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

Timer0模块和预分频器（与WDT共享）框图：



90h	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

81h	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]的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。

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	1152ms
111	1:256	1:128	2304ms

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
        BSF     STATUS,RP0          ;BANK1
        MOVWF   OPTION              ;定时器模式，分频比为1:4
        MOVLW   0X00
        BCF     STATUS,RP0          ;BANK0
        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
```

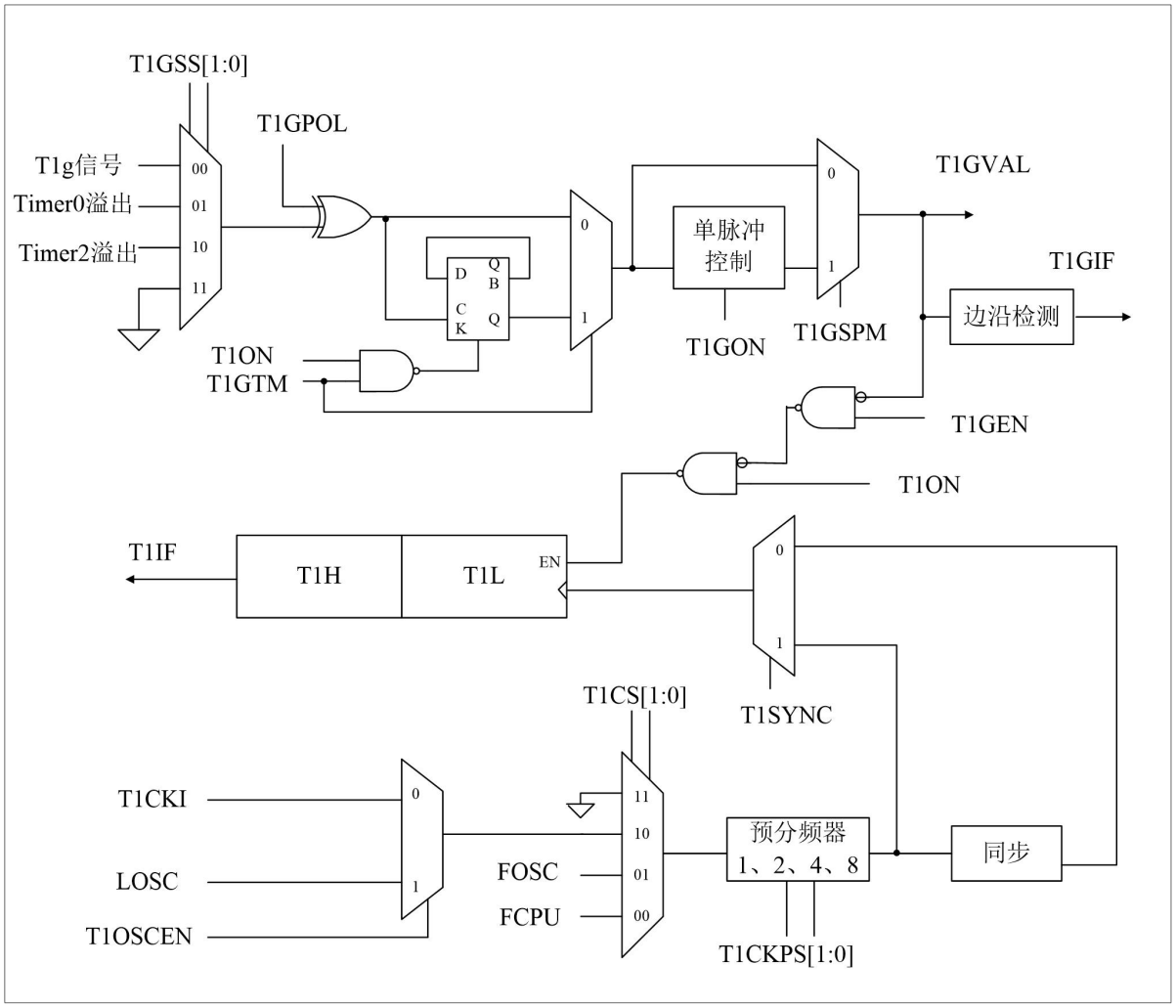
```
        …                                   ;TIMER0中断服务程序
        …
EXIT_INT:
        …                                   ;恢复STATUS、W和PCLATH
        RETFIE                              ;退出中断
```

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，可编程选择同步或异步功能
- 绿色模式定时唤醒
- Timer1 门控

Timer1模块框图：



8.3.1 TIMER1 控制寄存器

10h	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（FCPU）
0	1	x	系统时钟（Fsys）
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0。

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

8.3.2 TIMER1 门控寄存器

8Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1GCON	T1GEN	T1GPOL	T1GTM	T1GSPM	T1GON	T1GVAL	T1GSS1	T1GSS0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
POR的值	0	0	0	0	0	x	0	0

Timer1模式选择：

T1ON	T1GEN	Timer1工作模式
0	x	关闭
1	0	定时器/计数器模式
1	1	门控模式

Timer1门控源和门控极性选择：

T1GSS[1:0]	T1GPOL	Timer1门控源
01	0	Timer0溢出信号
01	1	Timer0溢出信号取反
10	0	Timer2溢出信号
10	1	Timer2溢出信号取反
00	0	门控引脚信号
00	1	门控引脚信号取反
11	x	系统保留

Timer1 门控模式选择：

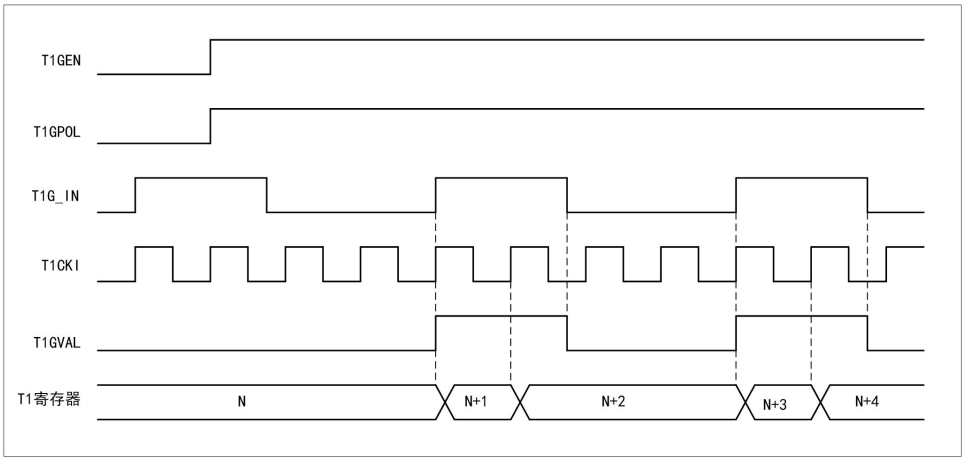
T1GTM	T1GSPM	门控模式
0	0	门控脉宽模式
0	1	门控单脉宽模式
1	0	门控周期模式
1	1	门控单周期模式

在门控模式下，通过寄存器T1CON选择时钟源及分频比，通过寄存器T1GCON控制选择门控模式、门控源及门控极性。其中门控寄存器中：T1GVAL表示Timer1门控的当前状态;T1GON是Timer1门控单脉宽和单周期模式采集使能和状态位，T1GON为1表示采集就绪，正在等待一个边沿，当门控输入信号有效时，Timer1开始计数，输入信号无效时，Timer1保持当前计数不变。T1GON为0表示Timer1采集已经结束或尚未开始;当T1GSPM 清零时，T1GON位会自动清零。

模式一：Timer1门控脉宽模式

Timer1门控脉宽模式使能时，可测量Timer1门控信号持续脉宽的长度。Timer1门控输入有效时，Timer1将在Timer1时钟源的上升沿递增。Timer1门控输入无效时，不会发生递增，Timer1将保持当前计数。

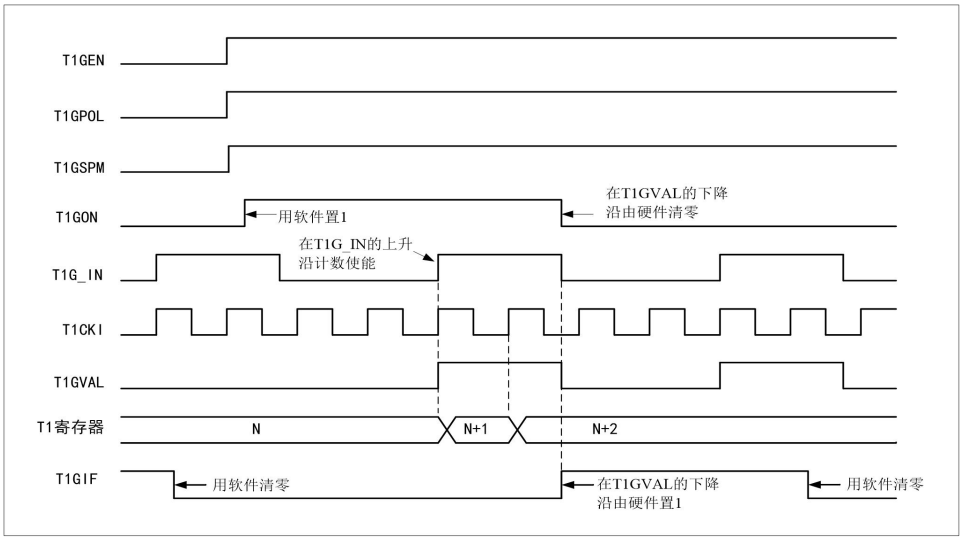
时序详细信息参见下图：



模式二：Timer1门控单脉宽模式

Timer1门控单脉宽模式使能时，可测量Timer1门控输入信号一个脉宽的长度。

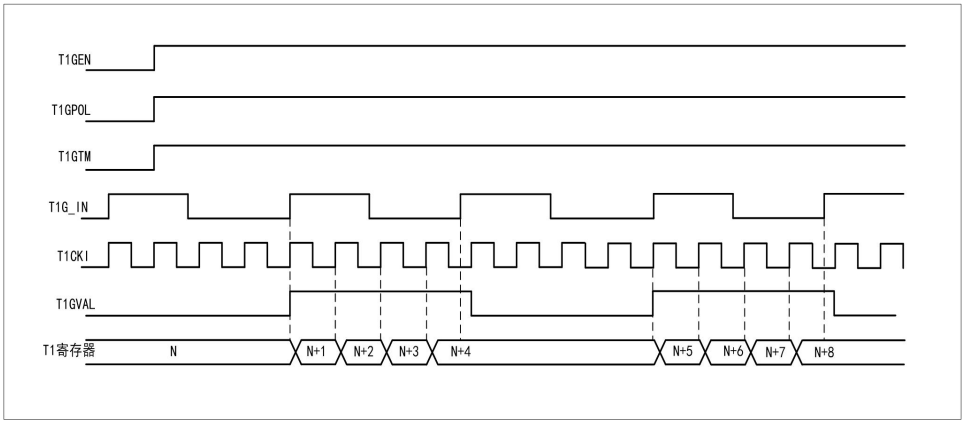
时序详细信息参见下图：



模式三：Timer1门控周期模式

Timer1门控周期模式使能时，可测量Timer1门控输入有效信号的整个周期的长度。门控设置完成后，门控输入信号的第一个上升沿开始计数，到第二个上升沿停止计数。门控输入信号的第三个上升沿开始累加计数，至第四个上升沿停止计数，以此类推。

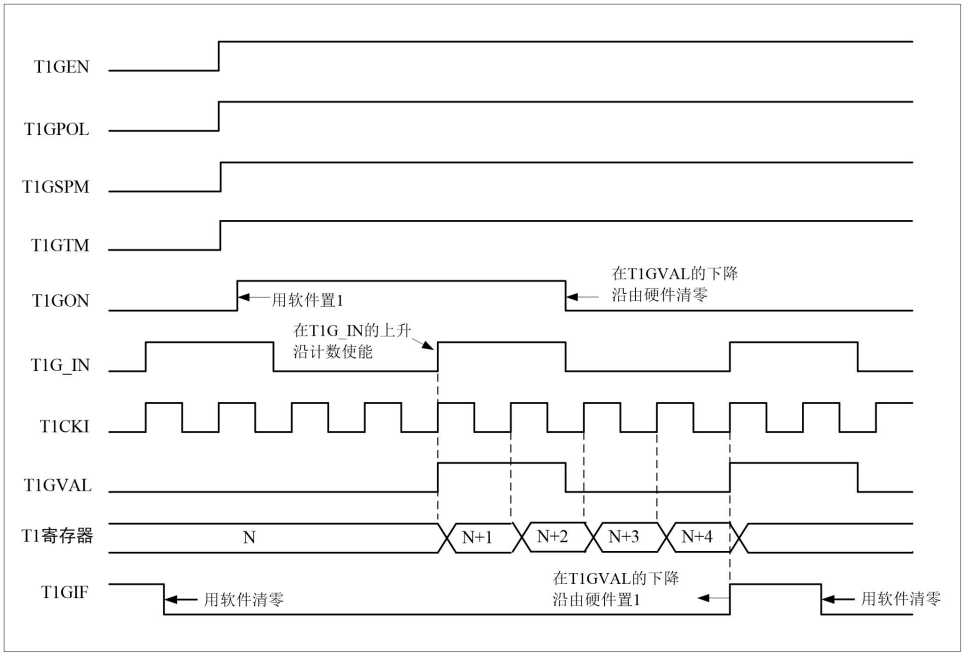
时序详细信息参见下图：



模式四：Timer1门控单周期模式

Timer1门控单周期模式使能时，可测量Timer1门控输入信号一个周期的长度。

时序详细信息参见下图：



➢ 例：测量某一单脉宽的长度

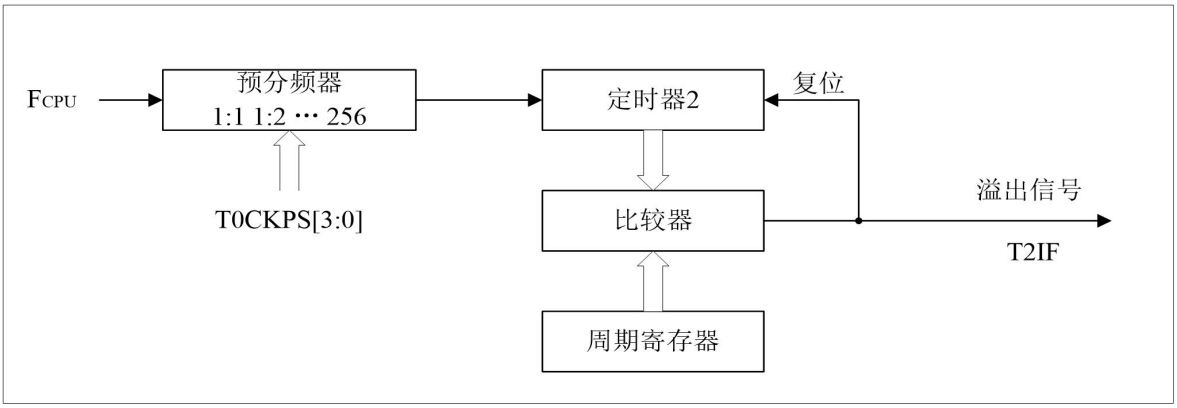
```
BSF     STATUS,RP0          ;BANK1
CLRF    INTCON              ;禁止门控中断
BCF     STATUS,RP0          ;BANK0
BSF     T1CON,T1CS1
BCF     T1CON,T1CS0
BCF     T1CON,T1OSCEN       ;Timer1时钟源为T1CKI
BCF     T1CON,T1CKPS1
BCF     T1CON,T1CKPS0       ;分频比为1:1
```

```
BSF     STATUS,RP0          ;BANK1
BSF     T1GCON,T1GPOL
BCF     T1GCON,T1GSS1
BCF     T1GCON,T1GSS0       ;门控引脚信号取反
BCF     T1GCON,T1GTM
BSF     T1GCON,T1GSPM       ;单脉宽模式
BCF     STATUS,RP0          ;BANK0
CLRF    T1L
CLRF    T1H
BSF     T1CON,T1ON
BCF     PIR1,T1GIF          ;清除门控中断标志位
BSF     STATUS,RP0          ;BANK1
BSF     T1GCON,T1GEN        ;门控使能
BSF     T1GCON,T1GON        ;门控打开
BCF     STATUS,RP0          ;BANK0
BTFSS   PIR1,T1GIF
GOTO    $-1
MOVF    T1H,0               ;发生一次门控中断时间
MOVWF   T1H_TEMP
MOVF    T1L,0
MOVWF   T1L_TEMP            ;脉宽长度存至T1H_TEMP,T1L_TEMP
```

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。

Timer2模块框图：



Timer2 溢出时间 = （PR2 + 1）* 预分频比/Fcpu

12h	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

11h	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

92h	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。

8.5 CCP 模块

HC18P110A0/B0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

17h、1Dh	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，如果ADC模块被使能，启动一次ADC转换(仅限于CCP1)，CCPx引脚不受影响。）

11xx = PWM模式

8.5.1 捕捉模式

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 TMR1 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

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。

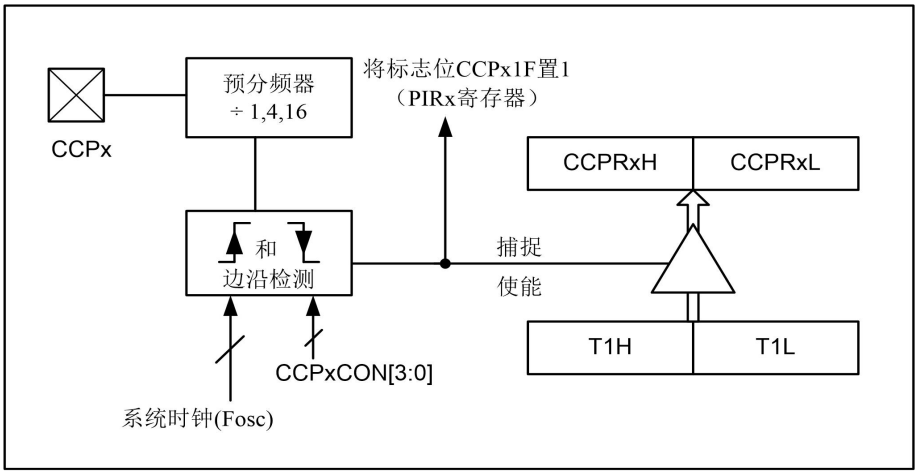
2、 CCPxCON寄存器的CCPxM[3:0]设置的是预分频器，关闭CCP模块或者CCP模块不在捕捉模式，预分频计数器将会被清零。为避免错误中断，可在改变预分频比前通过清零 CCPxCON寄存器来关闭CCP模块。如果需要中途改变预分频器的分频比，建议使用以下程序片段：

```
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。

捕捉模式工作原理图：



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寄存器对会不断地与Timer1的周期寄存器中的累加值作比较，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：

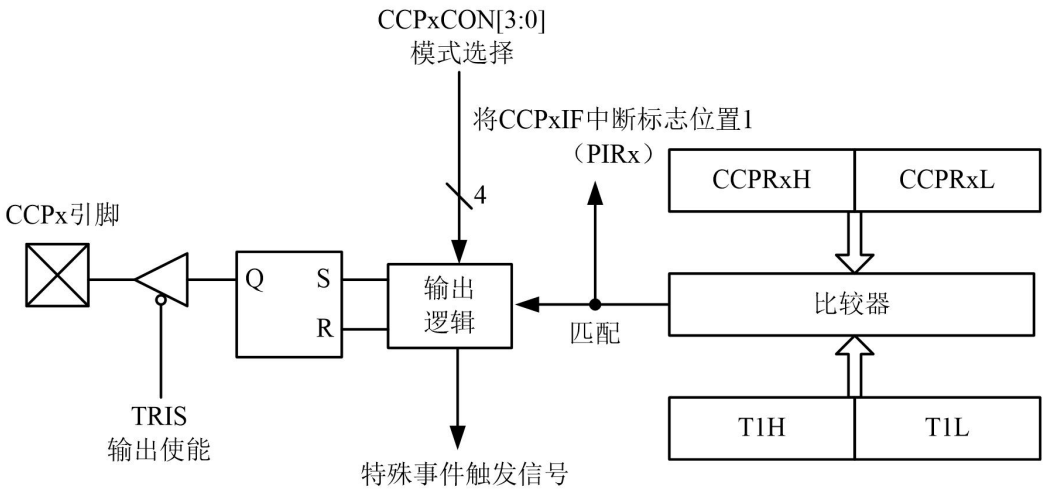
1、当选择产生特殊事件触发信号时，如果ADC被使能，则启动一次ADC转换(仅限于CCP1)。在此模式下CCPx模块不会对CCPx引脚进行控制。

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。

比较模式工作原理图：



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

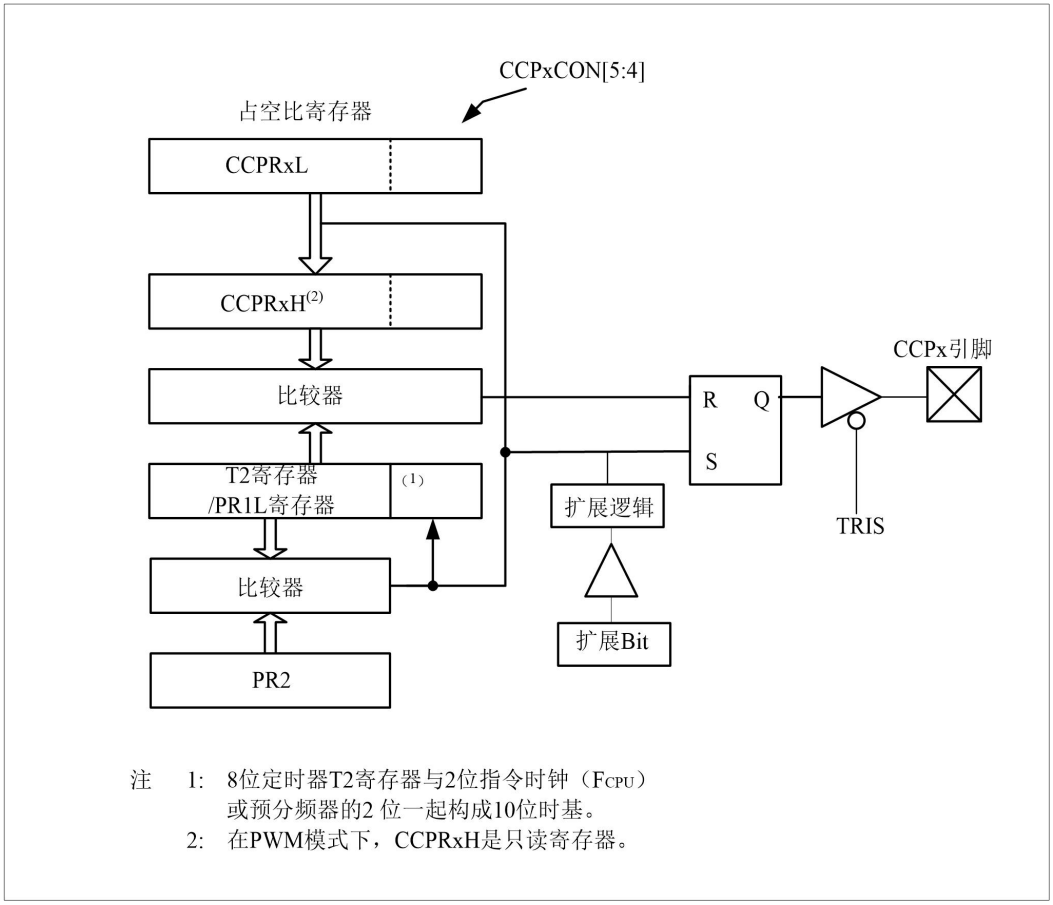
脉宽调制，PWM(pulse width modulation)输出工作模式，适用于从引脚上输出脉冲宽度随时可调的 PWM 信号。例如，实现直流电动机调速、简易 D/A 转换器、步进电机的变频控制等。

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）。

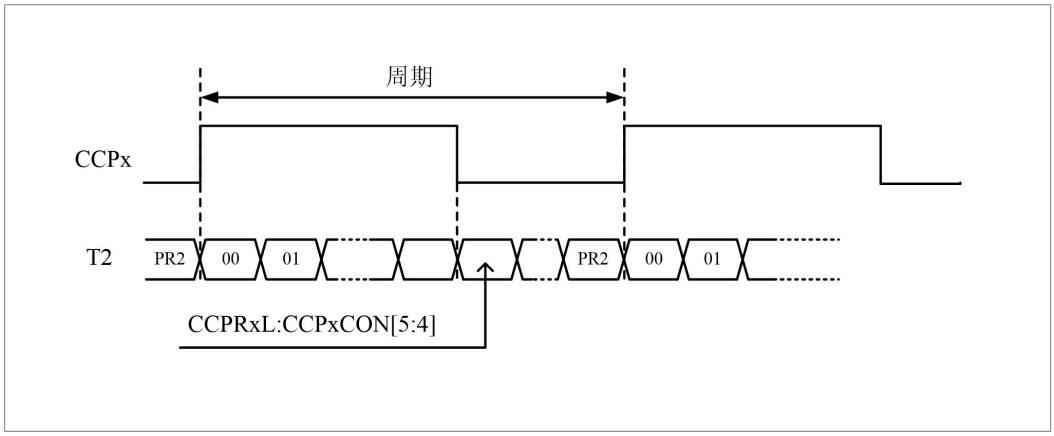
PWM模式工作原理图：



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。

PWM波形图：



✓ **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：

$$\text{PWM 周期} = [(PR2) + 1] \bullet 4 \bullet n \bullet Tosc \bullet (\text{Timer2 预分频值})$$

注： Tosc = 1/Fosc

PWM脉冲宽度：

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

注： Tosc = 1/Fosc

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位清零，让CCPx引脚输出

7.如何关闭PWM输出
   - 将TRISB2或TRISB3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:
        BSF     STATUS,RP0      ;BANK1
        BCF     PCON,WDTENS     ;DISABLE WDT
        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26US

        BCF     STATUS,RP0      ; BANK0
        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13US

        BCF     PIR1,T2IF
        CLRF    T2CON           ;分频比1：1
        BSF     T2CON,T2ON      ;开T2
        BSF     STATUS,RP0
        BSF     PIE1,T2IE

        BTFSS   PIR1,T2IF       ;等待T2溢出
        GOTO    $-1
        BCF     PIR1,T2IF       ;清除中断标志
        BSF     STATUS,RP0      ;BANK1
        BCF     PIE1,T2IE
        BCF     TRISB,2         ; PB2将持续输出38K
```

二、 PWM2时钟源为Timer1

18h	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

19h	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

19h	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[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

Bit[7:6] **PWM1T[1:0]**:PWM1的扩展周期选择位

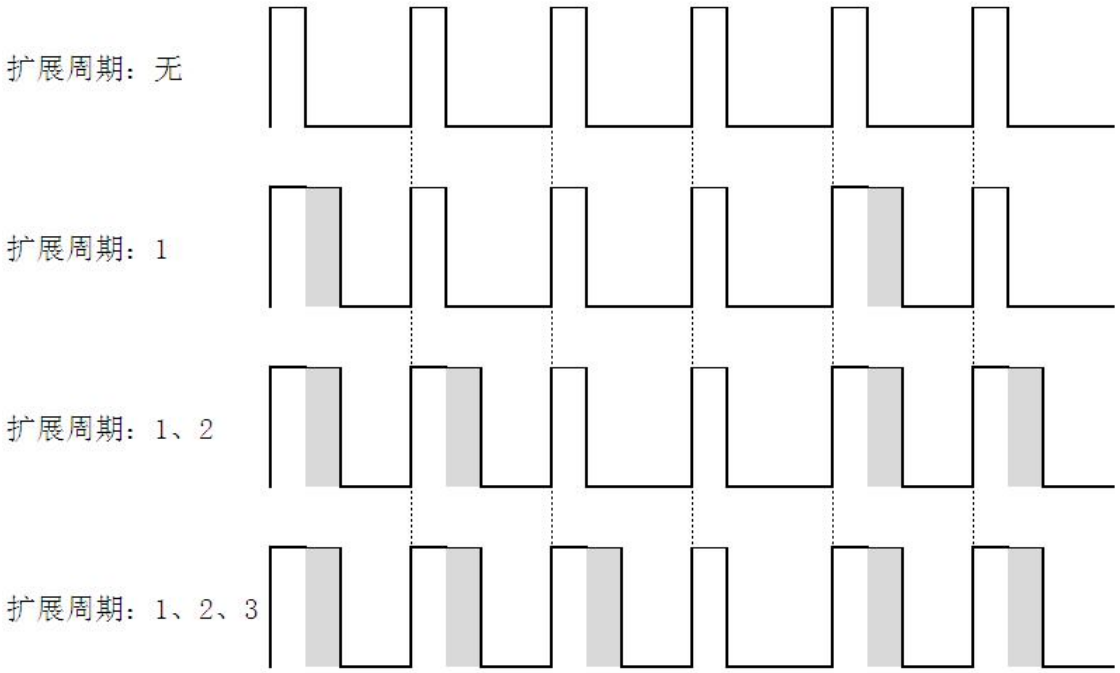
00: 无扩展周期

01: 扩展周期为1

10: 扩展周期为1、2

11: 扩展周期为1、2、3

PWM扩展示意如下：

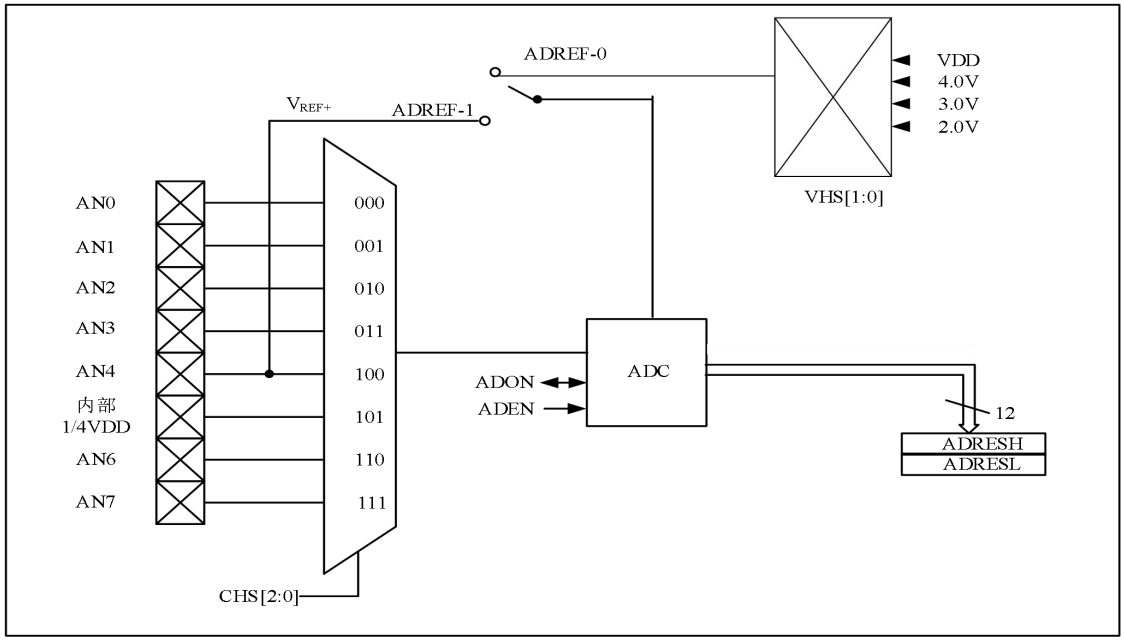


9 模数转换器 (ADC)

9.1 ADC 概述

HC18P110A0/B0具有一个12位转换分辨率的模数转换器，共有7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：



9.2 ADC 寄存器

9Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSEL	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

Bit [7-5]　未实现

Bit [4-0]　**ANSEL[7:6] ,[4:0]**：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：该寄存器上电初始值为 B’11xx 111x’，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。

1Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCON0	-	VHS1	VHS0	CHS2	CHS1	CHS0	ADON	ADEN
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

9Fh	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	R/W	R/W	R/W	-	-	-	R/W
POR的值	0	0	0	0	-	-	-	0

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)。

87h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCKCS	-	-	-	-	-	ADCKCS2	ADCKCS1	ADCKCS0
R/W	-	-	-	-	-	R/W	R/W	R/W
POR的值	-	-	-	-	-	1	1	1

ADC模拟通道选择

CHS [2:0]	模拟通道
000	AN0
001	AN1
010	AN2
011	AN3
100	AN4
101	AN5
110	AN6
111	AN7

ADC参考电压选择

ADREF	VHS[1:0]	参考电压
0	00	内部2.0V
0	01	内部3.0V
0	10	内部4.0V
0	11	内部VDD
1	xx	外部参考电压

ADC数据存放格式选择

ADFM	数据格式
0	ADRESH[7:0]:ADRESL[7:4]
1	ADRESH[1:0]:ADRESL[7:0]

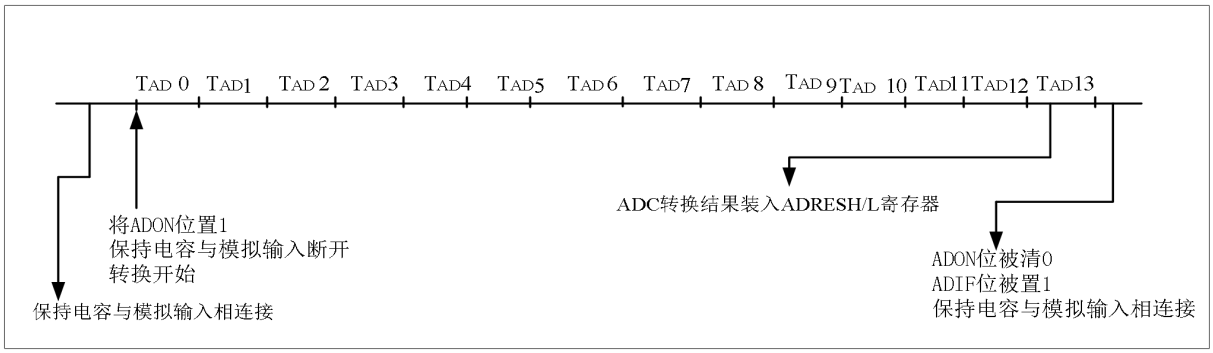
注意：

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。

9.3 ADC 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	2us
Fsys /2	001	4us
Fsys /4	010	8us
Fsys /8	011	16us
Fsys /16	100	32us
Fsys /32	101	64us
Fsys /64	110	128us

ADCKCS软件配置表：

Fsys	ADCS[2:0]	ADCKCS[2:0]
8M	011	001/010
	100	011/1xx
	101	011/1xx
	110	011/1xx
	111	011/1xx
4M	010	001/010
	011	011/1xx
	100	011/1xx
	101	011/1xx
	110	011/1xx
	111	011/1xx
2M	001	001/010
	010	011/1xx
	011	011/1xx
	100	011/1xx
	101	011/1xx
	110	011/1xx
	111	011/1xx
1M	000	001/010

	001	011/1xx
	010	011/1xx
	011	011/1xx
	100	011/1xx
	101	011/1xx
	110	011/1xx
	111	011/1xx
500K	000	011/1xx
	001	011/1xx
	010	011/1xx
	011	011/1xx
	100	011/1xx
	101	011/1xx
	110	011/1xx
	111	011/1xx
250K	000	011/1xx
	001	011/1xx
	010	011/1xx
	011	011/1xx
	100	011/1xx
	101	011/1xx
	110	011/1xx
	111	011/1xx

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
≥4Mhz	000	当ADC转换时钟频率高于或等于4 Mhz时，需将ADCKCS[2:0]配置成000
2 Mhz 或1 Mhz	001/010	当ADC转换时钟频率为2Mhz或1Mhz时，需将ADCKCS[2:0]配置成001/010
＜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1MHz）。

4. ADC连续采集时，两次采集之间须延时至少50微秒。即前一次采集完成，ADON硬件清零，须延时一段时间，再置一ADON，开启下一次采集。

9.4 ADC 操作

ADEN位置1将使能ADC模块，ADON位置1将启动一次ADC转换。ADC转换完成，ADON位硬件清零，ADIF中断标志位置1，ADRESH/ADRESL寄存器值被更新。如果必须在转换完成前终止转换，可用软件将ADON位清零，ADRESH/ ADRESL寄存器将保持前次ADC转换的结果。

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]
- 选择 ADC 参考电压，设置 ADREF、ADCON0
- 选择 ADC 输入通道，设置 CHS[2:0]
- 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）

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。

10 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

11 电性参数

11.1 极限参数

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

11.2 直流特性

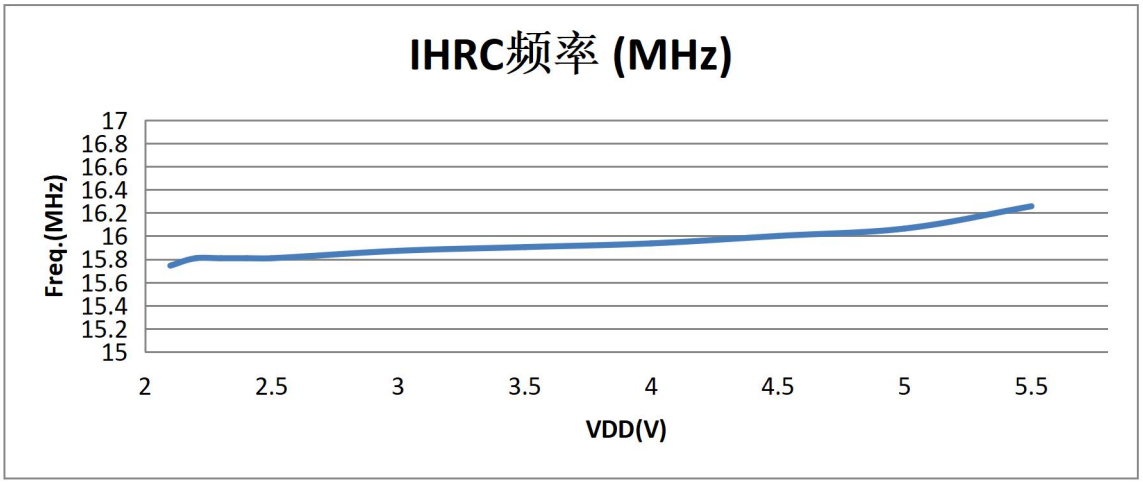
符号	参数	测试条件		最小值	典型值	最大值	单位
		VDD	条件（常温 25℃）				
VDD	工作电压	—	Fosc = 8MHz	2.4	—	5.5	V
		—	Fosc = 16MHz	4.5	—	5.5	V
IDD1	工作电流	3V	Fosc = 8MHz，4T，ADC 关闭，高频模式， WDT 禁止，无负载	—	1	—	mA
		5V		—	2	—	mA
IDD2	工作电流	3V	Fosc = 4MHz，4T，ADC 关闭，高频模式， WDT 禁止，无负载	—	0.6	—	mA
		5V		—	1.2	—	mA
IDD3	工作电流	3V	Fosc = 32KHz，4T，ADC 关闭，低频模式， WDT 禁止，无负载	—	5	—	uA
		5V		—	12	—	uA
IDD4	工作电流	3V	Fosc = 32KHz，4T，ADC 关闭，绿色模式， WDT 禁止，无负载	—	4	—	uA
		5V		—	12	—	uA
Isb1	静态电流	3V	ADC 关闭，休眠模式，WDT 使能，无负载	—	3	—	uA
		5V		—	9	—	uA
Isb2	静态电流	3V	ADC 关闭，休眠模式，WDT 禁止，无负载	—	—	1	uA
		5V		—	—	1	uA
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_{BOR}	低电压复位	—	—	—	—	—	V
V_{LVD1}	低电压标志	—	—	—	2.5	—	V
V_{LVD2}	低电压标志	—	—	—	3.7	—	V
I_{OL}	输出灌电流	5V	输出口，Vout=VSS+0.6V	—	25	—	mA
I_{OH}	输出拉电流	5V	输出口，Vout=VDD-0.6V	—	16	—	mA
R_{PH}	内部上拉电阻	5V	可编程上拉电阻	—	36	—	kΩ
V_{AD}	ADC 输入电压	—	—	VSS	—	V_{REF}	V
DNL	非线性误差	5V	T_{AD}=2us	—	+/-1	—	LSB
INL	线性误差	5V	T_{AD}=2us	—	+/-1	—	LSB
I_{ADC}	ADC 工作电流	3V	—	—	0.15	—	mA
		5V		—	0.5	—	mA

11.3 交流特性

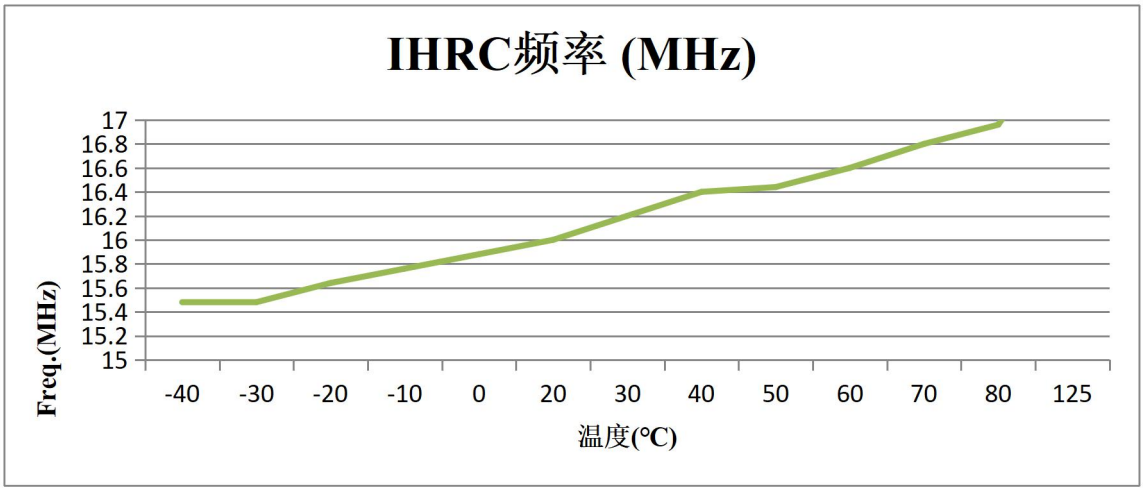
符号	参数	测试条件		最小值	典型值	最大值	单位
		VDD	条件（常温 25℃）				
Fosc	系统时钟	—	2.4V~4.5V	—	—	8	MHz
		—	4.5V~5.5V	—	—	20	MHz
FRCH	高频内部 RC 振荡器	5V	—	—	8	—	MHz
FRCL	低频内部 RC 振荡器	5V	—	14	32	—	KHz
FRCH	高频内部 RC 振荡器偏差		1.7V~5.5V			0.6%	
FRCL	低频内部 RC 振荡器偏差		1.7V~5.5V			8%	
FRCH	高频内部 RC 振荡器偏差		-40℃~85℃			2.7%	
FRCL	低频内部 RC 振荡器偏差		-40℃~85℃			20.66%	
TWDT	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	—	—	us

11.4 电气特性曲线图

测试条件：25 度



测试条件：5.0V



12 芯片配置字

芯片配置	配置选择	说明
POR时间	18ms	上电复位时间设置为18ms
	4.5ms	上电复位时间设置为4.5ms
时钟模式	4T	1个指令周期由4个内部RC振荡器时钟组成
	2T	1个时钟周期由2个内部RC振荡器时钟组成
高低频启动	高频启动	系统启动时钟选择高频时钟
	低频启动	系统启动时钟选择低频时钟
ADC功能使能	屏蔽ADC功能	关闭ADC模块
	使能ADC功能	使能ADC模块
低频系统时钟	定时器1振荡器	低频系统时钟选择外部低频晶振
	32K WDT振荡器	低频系统时钟选择内部WDT RC时钟
高频内部RC频率	8MHz	内部RC振荡器频率为8MHz
	4MHz	内部RC振荡器频率为4MHz
	2MHz	内部RC振荡器频率为2MHz
	1MHz	内部RC振荡器频率为1MHz
	500KHz	内部RC振荡器频率为500KHz
加密功能使能	不加密	屏蔽代码加密功能
	加密	使能代码加密功能
外部复位使能	屏蔽，做输入	屏蔽外部复位功能，PORTB5/MCLRB作为输入管脚
	使能外部复位	使能外部复位功能，PORTB5/MCLRB作为外部复位管脚
WDT功能使能	屏蔽WDT	屏蔽芯片内嵌硬件看门狗功能
	使能WDT	使能芯片内嵌硬件看门狗功能（仍可通过软件屏蔽）
高频系统时钟	高频晶体振荡器	高频系统时钟选择外部高频晶振
	内部8M振荡器	高频系统时钟选择内部8M振荡器
	外部RC振荡器	高频系统时钟选择外部RC振荡器
端口施密特使能	使能施密特	使能输入端口施密特功能
	屏蔽施密特	屏蔽输入端口施密特功能
BOR电压	2.0V	复位电压设置在2.0V
	2.4V	复位电压设置在2.4V
	3.6V	复位电压设置在3.6V
	NONE	关闭欠压复位

13 开发工具

13.1 OTP 烧录器（HC-PM5.0）

- PM18：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

13.2 HC-IDE

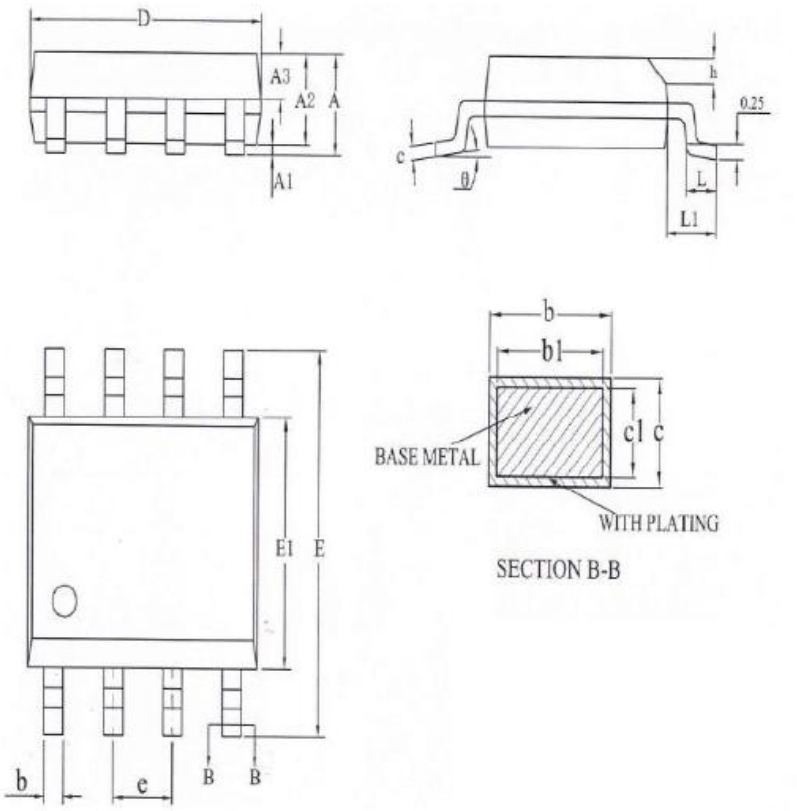
Holychip 8位单片机的集成开发环境HC-IDE包括编译器、HC-ICE18联机烧录、HC-PM18下载烧录软件。

- HC-IDE：V3.0.5.0
- ICD18 Programmer：V4.0
- PM18 Programmer：V5.0.5.0

> 注：
> 详情请参考 **HC-IDE** 用户手册。如有更新请到网站下载最新资料。

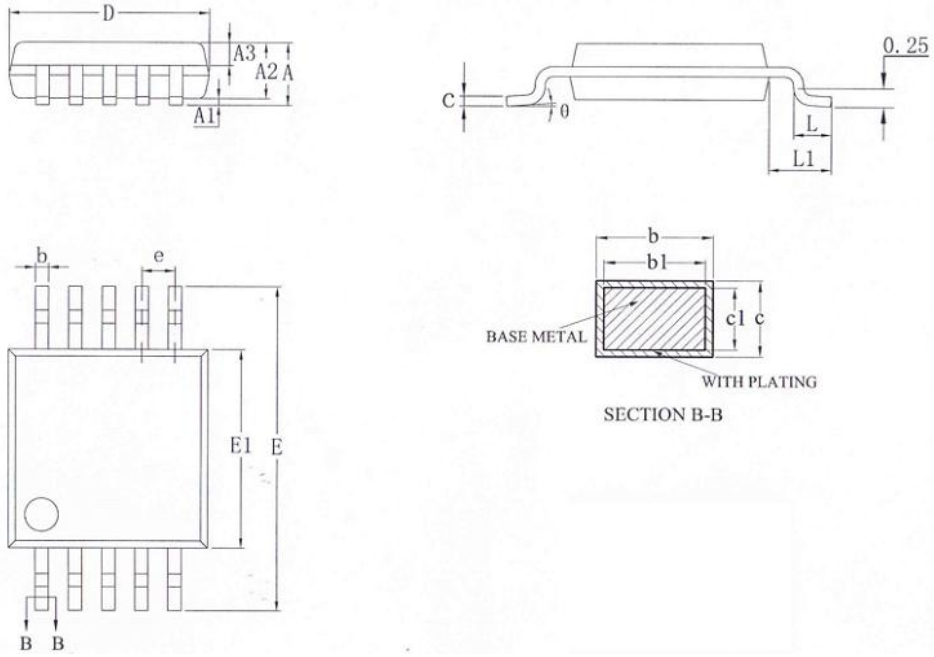
14 封装尺寸

14.1 SOP8



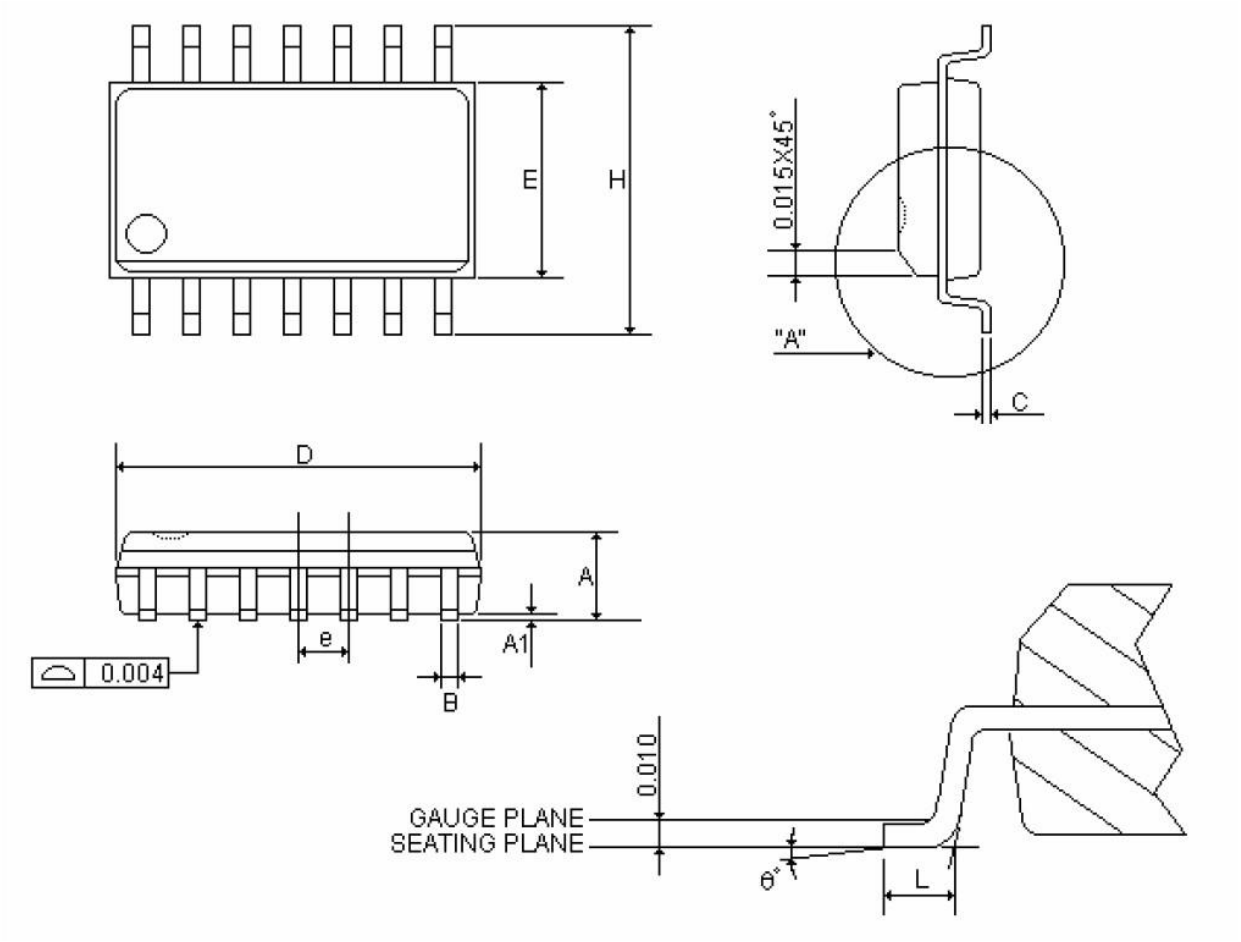
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.10	—	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
⚠1 b	0.39	—	0.47
⚠1 b1	0.38	0.41	0.44
⚠1 c	0.20	—	0.24
c1	0.19	0.20	0.21
⚠1 D	4.80	4.90	5.00
E	5.80	6.00	6.20
⚠1 E1	3.80	3.90	4.00
e	1.27BSC		
h	0.25	—	0.50
L	0.50	—	0.80
L1	1.05REF		
θ	0	—	8°

14.2 MSOP10



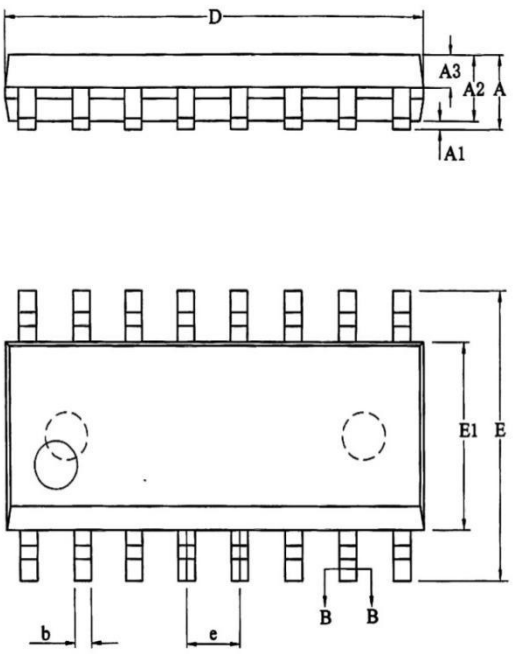
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.10
A1	0.05	—	0.15
A2	0.75	0.85	0.95
A3	0.30	0.35	0.40
b	0.18	—	0.26
b1	0.17	0.20	0.23
c	0.15	—	0.19
c1	0.14	0.15	0.16
D	2.90	3.00	3.10
E	4.70	4.90	5.10
E1	2.90	3.00	3.10
e	0.50BSC		
L	0.40	—	0.70
L1	0.95REF		
θ	0	—	8°

14.3 SOP14



SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	0.058	0.064	0.068	1.4732	1.6256	1.7272
A1	0.004	-	0.010	0.1016	-	0.254
B	0.013	0.016	0.020	0.3302	0.4064	0.508
C	0.0075	0.008	0.0098	0.1905	0.2032	0.2490
D	0.336	0.341	0.344	8.5344	8.6614	8.7376
E	0.150	0.154	0.157	3.81	3.9116	3.9878
e	-	0.050	-	-	1.27	-
H	0.228	0.236	0.244	5.7912	5.9944	6.1976
L	0.015	0.025	0.050	0.381	0.635	1.27
θ°	0°	-	8°	0°	-	8°

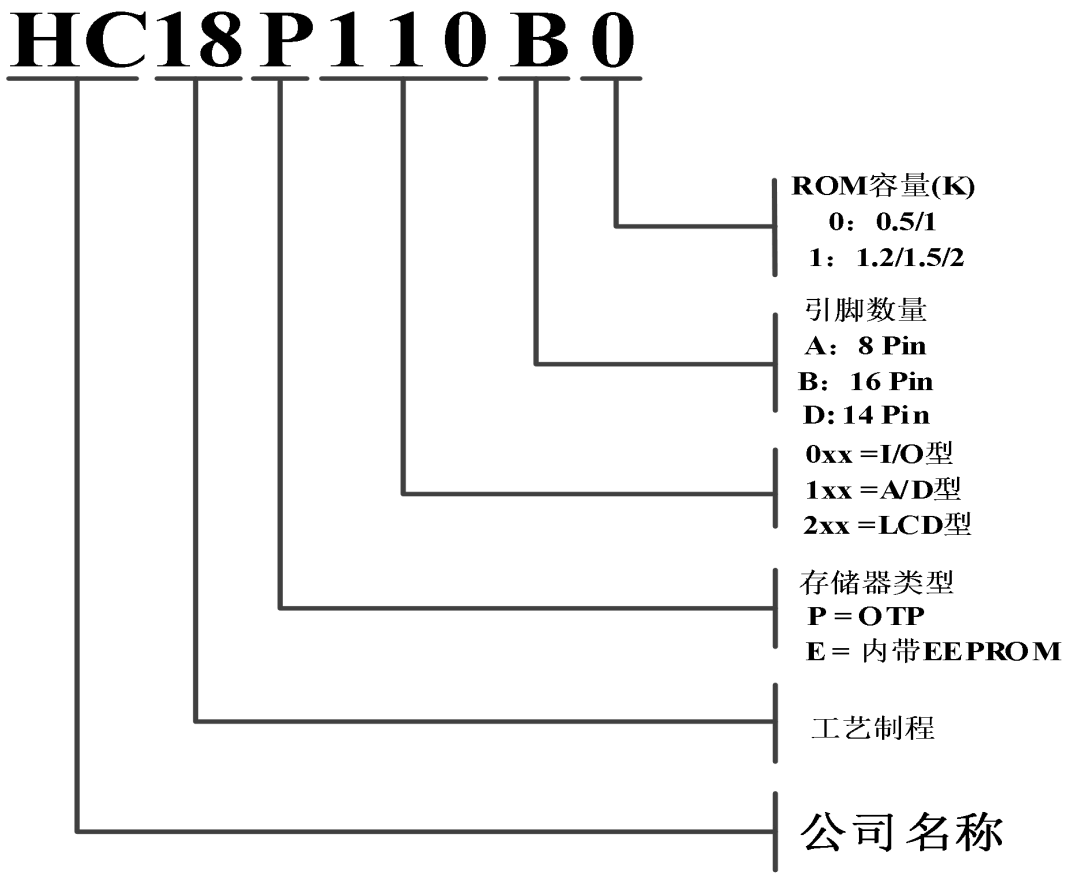
14.4 SOP16



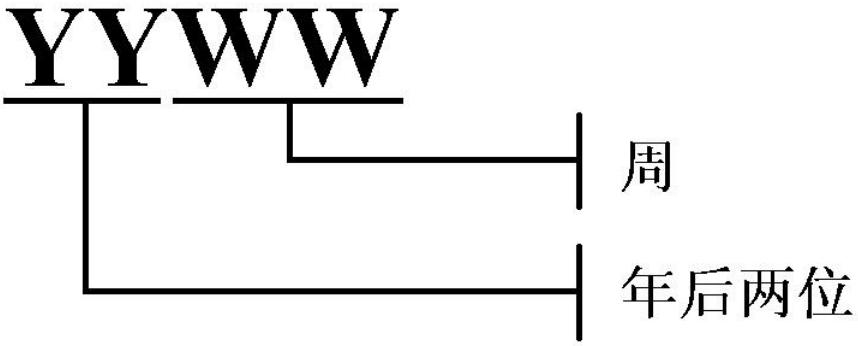
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.05	—	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	—	0.48
b1	0.38	0.41	0.43
c	0.21	—	0.26
c1	0.19	0.20	0.21
D	9.70	9.90	10.10
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	1.27BSC		
h	0.25	—	0.50
L	0.50	—	0.80
L1	1.05BSC		
θ	0	—	8°

15 芯片正印命名规则

15.1 芯片型号说明（第一行）



15.2 日期码规则（第二行）



15.3 生产批号（第三行）

例：FA126026A

16 修改记录

版本	日期	描述
Ver1.00	2020-11	第一版
Ver1.01	2021-09-18	P67，CCP-PWM 使用 T1 时基，T1L 必须清零
Ver1.01	2021-09-18	P67，CCP-PWM 使用 T1 时基时钟源 Fsys，时钟模式必须为 4T
Ver1.01	2021-09-18	P72，两次 ADC 采集之间须延时至少 13 微秒

芯圣电子

2020 年 11 月