

# HC18P018A0

数据手册

8 引脚 8 位  
I/O 型 OTP 单片机

# 目录

目录.....	2
<b>1 产品简介.....</b>	<b>4</b>
1.1 功能特性.....	4
1.2 引脚图.....	6
1.3 引脚描述.....	6
<b>2 中央处理器（CPU）.....</b>	<b>8</b>
2.1 存储器.....	8
2.1.1 程序存储器（OTPROM）.....	8
2.1.2 通用数据存储器（RAM）.....	11
2.1.3 特殊功能寄存器（SFR）.....	12
2.1.4 芯片配置选择.....	15
2.2 寻址模式.....	17
2.2.1 立即寻址.....	17
2.2.2 直接寻址.....	17
2.2.3 间接寻址.....	17
2.3 堆栈.....	18
<b>3 系统时钟.....</b>	<b>19</b>
3.1 概述.....	19
3.2 时钟框图.....	19
3.3 系统高频时钟.....	20
3.3.1 内部高频RC.....	20
3.4 系统低频时钟.....	22
3.4.1 内部低频RC 振荡器.....	22
<b>4 复位.....</b>	<b>23</b>
4.1 概述.....	23
4.2 上电复位.....	24
4.3 WDT 复位.....	24
4.4 欠压复位.....	25
4.4.1 欠压复位的产生.....	25
4.4.2 工作死区.....	25
4.4.3 工作死区与工作频率的关系.....	26
4.4.4 死区防护.....	26
4.5 外部复位.....	26
4.5.1 二极管RC 复位电路.....	26
4.5.2 电压偏置复位电路.....	27
<b>5 系统工作模式.....</b>	<b>28</b>
5.1 概述.....	28
5.2 休眠模式.....	29
5.3 模式切换举例.....	29
5.4 高低频时钟切换.....	30
5.5 唤醒时间.....	31
5.6 寄存器OSCCON.....	31
<b>6 中断.....</b>	<b>32</b>
6.1 概述.....	32
6.2 中断请求和标志寄存器.....	33

6.3 GIE 全局中断.....	34
6.4 中断保护.....	34
6.5 Timer0 中断.....	35
6.6 INTO 中断.....	36
6.7 端口电平变化中断.....	37
6.8 Timer1 中断.....	38
6.9 LVD/CMP 中断.....	39
6.10 充电中断.....	41
6.11 过温保护中断.....	41
<b>7 I/O 端口.....</b>	<b>42</b>
7.1 I/O 端口模式.....	42
7.2 I/O 上拉模式.....	42
7.3 I/O 下拉模式.....	43
7.4 I/O 开漏模式.....	43
7.5 I/O 端口数据寄存器.....	44
<b>8 定时器.....</b>	<b>45</b>
8.1 看门狗定时器.....	45
8.2 Timer0 定时/计数器.....	46
8.3 Timer1 定时/计数器.....	49
8.3.1 功能概述.....	49
8.3.2 T1 使用操作说明.....	50
8.3.1 T1 相关寄存器.....	50
<b>9 LVD/CMP.....</b>	<b>54</b>
<b>10 充电控制.....</b>	<b>56</b>
<b>11 大电流口以及内置限流电阻控制.....</b>	<b>57</b>
<b>12 指令表.....</b>	<b>58</b>
<b>13 电气特性.....</b>	<b>60</b>
<b>14 典型应用原理图.....</b>	<b>63</b>
<b>15 开发工具.....</b>	<b>62</b>
15.1 OTP 烧录器 (HC-PM18 4.0) .....	64
15.2 HC-IDE.....	64
<b>16 封装信息.....</b>	<b>65</b>
16.1 SOP8.....	65
<b>17 芯片正印命名规则.....</b>	<b>66</b>
17.1 芯片型号说明 (第一行) .....	66
17.2 日期码规则 (第二行) .....	66
17.3 生产批号 (第三行) .....	66
<b>18 修改记录.....</b>	<b>67</b>

# 1 产品简介

HC18P018A0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，内置大功率1A NMOS管3个，大电流可配置。同时集成充电管理功能，过温保护。内部有1K×14位一次性编程ROM(OTP-ROM)，64×8位的数据寄存器（RAM），1组双向I/O口，2个Timer定时器/计数器，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手电照明等LED功能产品。

## 1.1 功能特性

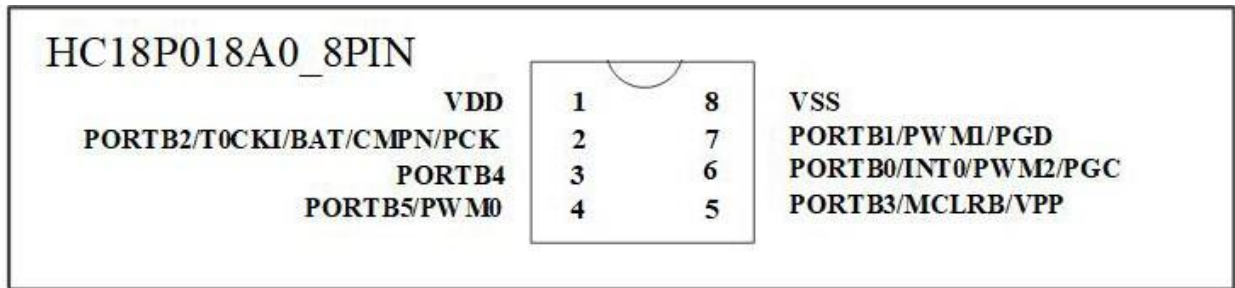
- ◆ **存储器配置**
  - 程序存储器（OTP ROM）空间：1K\*14位/0.5K\*14位
  - 数据存储器（RAM）空间：64\*8位
- ◆ **强大的指令系统**
  - 时钟系统可设（2T/4T）
  - 39条高性能精简指令
  - 大部分指令皆可在一个机器周期完成
  - 支持立即、直接和间接寻址模式
- ◆ **5级堆栈缓冲器**
- ◆ **I/O引脚配置**
  - 所有IO口均具有可编程的上下拉、开漏输出控制
  - 最多6个双向I/O口
  - 单向输入端口：PORTB<3>与复位引脚复用
  - PORTB具有唤醒功能，可通过IOCB独立配置
  - 具有唤醒功能的外部中断引脚：PORTB<0>，可设置触发边沿
  - 3个大灌电流口：PORTB<2:0>，单一电流口最大灌电流1A。
- ◆ **充电模块**
  - 充电模块采用恒流模式，最大充电电流支持400mA。
- ◆ **BOR**
  - 7级低电压复位
  - 系统VDD在未达到BOR点以上时，系统功耗小于1μA
- ◆ **LVD**
  - 16级电压检测
  - 可编程设置检测VDD或LVDIN
- ◆ **中断**
  - 定时器中断：Timer0和Timer1
  - INT0外部中断
  - 端口电平变化中断
  - LVD中断
  - 充电检测中断
  - 过温保护中断
- ◆ **定时器**
  - 看门狗计数器（WDT）
  - 1个8位定时/计数器
  - 1个带有蜂鸣器和3个PWM功能的8位定时器
- ◆ **系统时钟**
  - 内建高精度16MHz RC时钟
  - 内建32KHz低频RC时钟
- ◆ **工作模式**
  - 高频模式

- 低频模式
- 绿色模式
- 休眠模式
- ◆ 复位
  - 上电复位(POR 时间可选, 最小支持 140μs)
  - BOR 欠压复位
  - 外部端口复位
  - WDT 溢出复位
- ◆ 封装
  - SOP8

## ✓ 选型表

产品型号	ROM	RAM	堆栈	定时器	I/O	Wake up port	封装形式
HC18P018A0	1K*14	64*8	5	2	5+1	6	SOP8

## 1.2 引脚图



## 1.3 引脚描述

SOP8	名称	类型	说明
1	VDD	P	电源输入
2	PORTB2 T0CKI CMPN PCK BAT	I/O I/O I I I	输入/输出口，带可编程上下拉，端口电平变化中断，开漏输出 Timer0 外部时钟输入口（施密特触发器） 比较器反相输入端 测试模式下内部 RC 输出口 充电管理端口
3	PORTB4	I/O	输入/输出口，带可编程上下拉，端口电平变化中断，开漏输出，可配置 内置 1K 限流电阻
4	PORTB5 PWM0	I/O O	输入/输出口，带可编程上下拉，端口电平变化中断，开漏输出，灌电流 能力 1A，可配置 PWM0 输出，由 AUXR 寄存器中的 PWMCT 控制
5	PORTB3 VPP MCLR/VPP	I/O I I	输入/输出口，带可编程上下拉，端口电平变化中断，开漏输出 编 程高压输入口 复位输入口，内部上拉电阻自动生效，且为施密特结构
6	PORTB0 INT0 PWM2 PGC	I/O I O I	输入/输出口，带可编程上下拉，端口电平变化中断，开漏输出，灌电流 能力 1A，可配置 INT0 外部中断输入口 PWM2 输出，由 AUXR 寄存器中的 PWMCT 控制 编程时钟输入口
7	PORTB1 PWM1 PGD	I/O O O	输入/输出口，带可编程上下拉，端口电平变化中断，开漏输出，灌电流 能力 1A，可配置 PWM1 输出，由 AUXR 寄存器中的 PWMCT 控制 编 程数据输入/输出口
8	VSS	P	电源地

注 1: I = 输入 O = 输出 I/O = 输入/输出 P = 电源

注 2: 2 脚 (BAT 脚) 作电池输入端时, 必须设置为输入。

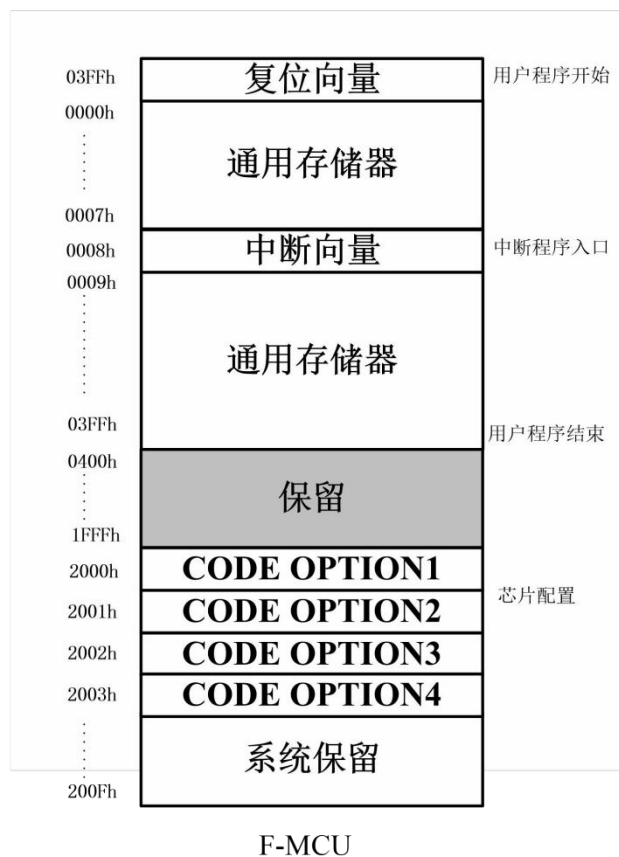
注 3: PB4 为系统时钟输出口, OPTION 配置打开后可输出Fosc。

## 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



**特别说明：** OPTION 选项中都有芯片兼容选择（菲林）。  
当ROM SIZE 选择 0.5K 时注意起始地址。

##### 2.1.1.1 复位向量（0000h/03FFh）

- 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/03FFh 处重新开始执行，系统寄存器也都将恢复为默认值。根据AUXR 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      0040H      ;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:
                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
    
```

### 2.1.1.2 中断向量（0008H）

中断向量地址为 0008H。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 0008H 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      0X20
STATUS_TEMP EQU      0X21
PCLATH_TEMP EQU      0X22
                ...
                ORG      0008H
                MOVWF   W_TEMP      ;保护W 寄存器
                MOVF    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
BTFSC   INTECON,INTF
GOTO    ISR_INT0        ;发生 INT0 中断
BTFSC   INTECON,T0
GOTO    ISR_T0          ;发生 TIMER0 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     INTECON,INTF    ;外部中断处理
...
GOTO    INT_EXIT

ISR_T0:
BCF     INTECON,T0IF    ;TIMER0 中断处理
...
GOTO    INT_EXIT

```

**注:**

对于编写中断服务程序，以下几个要点需注意

1. 中断入口地址为 0008H，响应中断后，程序指针自动跳转到 0008H 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. 中断服务子程序返回前对保护的寄存器进行恢复。
4. 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。
5. 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

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    ;获得数据表地址高位（内部宏指令）
MOVWF   PCLATH        ;表地址高位赋给PCLATH
MOVF    TABBUF,W      ;获得表数据地址
CALL    TAB1          ;调用数据表
...
ORG     0100H
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    DATA0_TAB1    ;W=0 对应数据
...
RETLW    DATAFE_TAB1   ;W=0XFE 对应数据
    
```

**注:**

**对于数据查表的编程，需注意**

1. 数据表数据为 8 位，数据表最大为 255 数据。
2. 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，故表头运算尽量放在数据表页面前端，以免数据表溢出。
3. 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    LABEL0_TAB2    ;W=0, 跳转 LABEL0_TAB2
GOTO    LABEL1_TAB2
GOTO    LABEL2_TAB2
GOTO    LABEL3_TAB2
    
```

**注:**

如上跳转表，有 4 个跳转分支，TABBUF 的选值范围为 0X00~0X03。

## 2.1.2 通用数据存储器（RAM）

共有 64 个通用寄存器（GPR），分在 Bank0 存储区。

地址	寄存器
00H~0FH	SFR
10H~3FH	GPR
40H~5AH	SFR
60H~6FH	GPR

**注:** 其中 07H 地址为 GPR。GP 共 64Bytes，请注意 FSR 高两位初始值。

## 2.1.3 特殊功能寄存器 (SFR)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
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	RST	GP1	GP0	TO	PD	Z	DC	C
04h	FSR	间接寻址地址指针							
06h	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
07h	GPR	通用寄存器							
08h	PCON	WDTEN	EIS	CMPOF	CPNIS3	CPNIS2	CPNIS1	CPNIS0	CMPEN
09h	IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0
0Ah	PCLATH	-	-	-	-	-	-	程序计数器高 2 位缓冲器	
0Bh	PDCON	GP	PDB2	PDB1	PDB0	-	-	-	-
0Ch	ODCON	ODB7	ODB6	ODB5	ODB4	DOB3	ODB2	ODB1	ODB0
0Dh	PHCON	PHB7	PHB6	PHB5	PHB4	PHB3	PHB2	PHB1	PHB0
0Eh	INTECON	GIE	-	CHARE	TMPE	-	INTE	PBIE	TOIE
0Fh	INTFLAG	-	-	CHARF	TMPF	CMPF	INTF	PBIF	TOIF
41h	OPTION	-	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
46h	TRISB	TRISB7	TRISB6	TRISB 5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB 0
48h	PWMCON	PWM0OE	PWM1OE	PWM2OE	PWMCK	PWMMD	PWMINV	PWM1E	PWM2E
49h	PWM1P	PWM1 占空比控制寄存器							
4Ah	PWM2P	PWM2 占空比控制寄存器							
4Bh	T0CR	T0CK	-	-	-	-	-	T1IE	T1IF
4Ch	T1CON	T1EN	PWM0E	BUZ	T1CK1	T1CK0	T1PR2	T1PR1	T1PR0
4Dh	T1	Timer1 计数寄存器							
4Eh	T1LOAD	Timer1 重载寄存器							
4Fh	PWM0P	PWM0 占空比控制寄存器							
50h	OSCCON	T0OSCEN	GP	GP	GP	GP	GP	HXEN	SCS
51h	CMPCR	GP	GP	GP	LVDM	CMPWK	CMPPIE	CMPPIES	CPPIS
52h	PDCON1	PDB7	PDB6	PDB5	PDB4	PDB3	-	-	--
53h	CHARGE0	S7	S6	S5	S4	S3	S2	S1	S0
54h	CHARGE1	CHARGEN	TEMPEN					S9	S8
55h	RAMP	MODESEL	SINK_SEL				SLOP2	SLOP1	SLOP0
59h	AUXR	POR	BOR	GP	GP	GP	RCTRMEN	PWM3CT	PWMCT
5Bh	HIRCTRM	内部高频时钟调整寄存器							

注：大于 40H 地址仅可使用直接寻址模式进行读写操作

### 2.1.3.1 寄存器 INDF

INDF 不是物理寄存器，对INDF 寻址实际上是对FSR 指向的数据存储器地址进行访问，从而实现间接寻址模式。

### 2.1.3.2 寄存器 FSR

间接寻址指针 FSR

04h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSR	-	间接寻址数据指针						
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	1	0	0	0	0	0	0	0

### 2.1.3.3 程序计数器

程序计数器（PC）为 11 位宽，低字节来自可读写的 PCL 寄存器，高字节（PC[10:8]）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。

程序计数器高 3 位

0Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCLATH	-	-	-	-	-	程序计数器高 3 位		
R/W	-	-	-	-	-	R/W	R/W	R/W
POR 的值	-	-	-	-	-	0	0	0

程序计数器低 8 位

02h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCL	程序计数器低 8 位							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

程序存储器指针（PC）的操作模式

- 顺序执行指令：PC+1 → PC
- 分支指令 GOTO/CALL：INST[10:0]（指令码低 10 位）→ PC
- 子程序返回指令RETRUN/RETLW/RETFIE：TOS（堆栈栈顶）→ PC
- ADDWF PCL, F
  - F-MCU: PCLATH[10:8], ALU[7:0]（ALU 运算结果）→ PC
- 其它 PCL 作为目的操作数指令
  - F-MCU: PCLATH[10:8], ALU[7:0] → PC

### 2.1.3.4 寄存器 STATUS

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	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	RST	GP1	GP0	TO	PD	Z	DC	C
R/W	R/W	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

- Bit [7] RST: 唤醒源标志  
 1 = 芯片通过PORTB 变化唤醒(复位/SLEEP 指令)  
 0 = 芯片通过其它复位唤醒
- Bit [6:5] 通用寄存器位
- 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

### 2.1.3.5 寄存器 AUXR

59h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AUXR	POR	BOR	GP	GP	GP	RCTRMEN	GP	PWMCT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	q	q	0	0	0	0	0	0

注：q = 取值视条件而定

- Bit [7] POR: 上电复位状态位  
 1 = 非上电复位  
 0 = 发生了上电复位 (需要软件置 1)
- Bit [6] 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)
- Bit [5:3] GP: 通用功能寄存器位
- Bit [2] RCTRMEN: 内部高频 RC 软件校准使能位  
 1=允许内部高频 RC 软件校准  
 0=禁止内部高频 RC 软件校准
- Bit [1] GP: 通用功能寄存器位

## 2.1.4 芯片配置选择

芯片配置	配置选择	说明
BOR 电压	1.3V (NONE)	复位电压设置为 1.3V
	1.5V	复位电压设置为 1.5V
	1.9V	复位电压设置为 1.9V
	2.2V	复位电压设置为 2.2V
	2.4V	复位电压设置为 2.4V
	2.6V	复位电压设置为 2.6V
	3.6V	复位电压设置为 3.6V
外部复位使能	禁止, 做输入	禁止外部复位功能, PORTB3/MCLR B 作为输入管脚
	使能外部复位	使能外部复位功能, PORTB3/MCLR B 作为外部复位管脚
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
WDT 溢出时间及 POR 时间	TWDT0	PWRT=9ms; TWDT(no Prescaler)=18ms
	TWDT1	PWRT=2.2ms; TWDT(no Prescaler)=4.5ms
	TWDT2	PWRT=144ms; TWDT(no Prescaler)=288ms
	TWDT3	PWRT=36ms; TWDT(no Prescaler)=72ms
	TWDT4	PWRT=140 $\mu$ s; TWDT(no Prescaler)=18ms
	TWDT5	PWRT=140 $\mu$ s; TWDT(no Prescaler)=4.5ms
	TWDT6	PWRT=140 $\mu$ s; TWDT(no Prescaler)=288ms
	TWDT7	PWRT=140 $\mu$ s; TWDT(no Prescaler)=72ms
WDT 功能使能	禁止 WDT	禁止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 (仍可通过软件禁止)
加密功能使能	不加密	禁止代码加密功能
	加密	使能代码加密功能
输入管脚施密特	使能施密特	使能输入端口施密特功能
	禁止施密特	禁止输入端口施密特功能
端口 SMT 翻转电平选择	0.7VDD/0.3DD	选择 0.7VDD/0.3DD
	0.3VDD/0.25VDD	选择 0.3VDD/0.25VDD
端口非SMT 阈值选择	0.25VDD	选择 0.25VDD
	0.5VDD	选择 0.5VDD
IO 输出驱动选择	IOH0/IOL0	IOH/IOL=6mA/21mA
	IOH1/IOL1	IOH/IOL=3.5mA/21mA
	IOH2/IOL2	IOH/IOL=0.9mA/4mA
	IOH3/IOL3	IOH/IOL=16mA/25mA
PORTB3 端口模式选择	PORTB3 位输入/开漏输出口	PORTB3 配置为输入/开漏输出口
	PORTB3 位输入口	PORTB3 仅为输入口
高频振荡器模式选择	内部高频 RC 振荡器	
	外部高频晶体 (1M-20M) 振荡器	
	外部高频晶体 (455KHZ) 振荡器	
系统启动时钟选择	高频启动	
	低频启动	
输出管脚读入	读端口	从芯片管脚读入
	读寄存器	从输出端口寄存器读入

兼容 MCU	F-MCU	兼容F MCU
低频晶振端口选择	PORTB5/PORTB4	低频晶振端口选择PORTB5/PORTB4
	PORTB1/PORTB0	低频晶振端口选择PORTB1/PORTB0
OTP 容量选择位	1K 一次	
	0.5K 第一次	
	0.5K 第二次	
	保留	
封装	6PIN	封装管脚为 6PIN
	8PIN	封装管脚为 8PIN
	14PIN	封装管脚为 14PIN
	16PIN	封装管脚为 16PIN
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
	455KHz	内部RC 振荡器频率为 455MHz
	32KHz	内部RC 振荡器频率为 32MHz
休眠模式下BOR 选择	休眠模式下关闭 BOR	在休眠模式下关闭复位电压
	休眠模式下打开 BOR	在休眠模式下打开复位电压
输出时钟选择	系统时钟输出关闭	
	系统时钟输出打开	
高频内部所选RC振荡器分频	1:1	所选内部高频RC 1 分频
	1:2	所选内部高频RC 2 分频
	1:4	所选内部高频RC 4 分频
	1:8	所选内部高频RC 8 分频
	1:16	所选内部高频RC 16 分频
	1:32	所选内部高频RC 32 分频
	1:64	所选内部高频RC 64 分频
	1:128	所选内部高频RC 128 分频
低频振荡器模式选择	外部低频振荡器	
	32K WDT 振荡器	
内部高频选16MHz	非 16MHz, 由高频内部RC 频率选项决定	内部 RC 振荡器频率由高频内部 RC 频率选项决定
	为 16MHz	内部 RC 振荡器频率为 16MHz



	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
	455KHz	内部RC 振荡器频率为 455KHz
	32KHz	内部RC 振荡器频率为 32KHz
高频内部 RC 分频	1:1	所选内部高频RC 1 分频
	1:2	所选内部高频RC 2 分频
	1:4	所选内部高频RC 4 分频
	1:8	所选内部高频RC 8 分频
	1:16	所选内部高频RC 16 分频
	1:32	所选内部高频RC 32 分频
	1:64	所选内部高频RC 64 分频
	1:128	所选内部高频RC 128 分频
输出管脚读入	读端口	从芯片管脚读入
	读寄存器	从输出端口寄存器读入
兼容 MCU	F-MCU	兼容F MCU
休眠模式下 BOR 选择	休眠模式下关闭BOR	在休眠模式下关闭复位电压
	休眠模式下打开BOR	在休眠模式下打开复位电压
低频晶振端口选择	PORTB5/PORTB4	低频晶振端口选择PORTB5/PORTB4
	PORTB1/PORTB0	低频晶振端口选择PORTB1/PORTB0
输出时钟选择	系统时钟输出关闭	将系统时钟输出口关闭
	系统时钟输出打开	将系统时钟输出口打开
OTP容量选择	1K 一次	1K OTP 烧录一次
	0.5K 第一次	0.5K OTP 烧录第一次
	0.5K 第二次	0.5K OTP 烧录第二次
	保留	
封装	6PIN	
	8PIN	
	14PIN	
	16PIN	

## 2.2 寻址模式

HC18P018A0 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

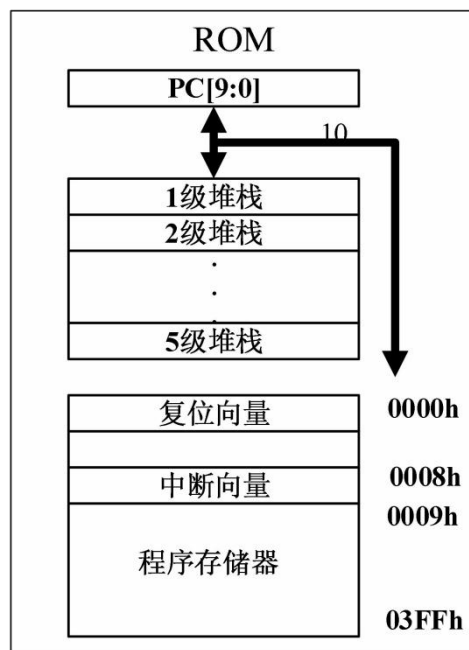
```

MOV LW    20h                ;清零 0X20~0X7F
MOV WF    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
MOV LW    80h
XOR WF    FSR,W
BTFS      STATUS,Z
GOTO      NEXTBYTE
MOV LW    0A0h               ;清零 0XA0~0XBF
                                ;0C0h~0FFh 映射到BANK1, 无需再清
MOV WF    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
MOV LW    0C0h
XOR WF    FSR,W
BTFS      STATUS,Z
GOTO      NEXTBYTE_1

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

### 2.3 堆栈

HC18P018A0 具有一个 5 级深度的硬件堆栈，堆栈指针不能读写。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 5 级，超过 5 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 系统时钟

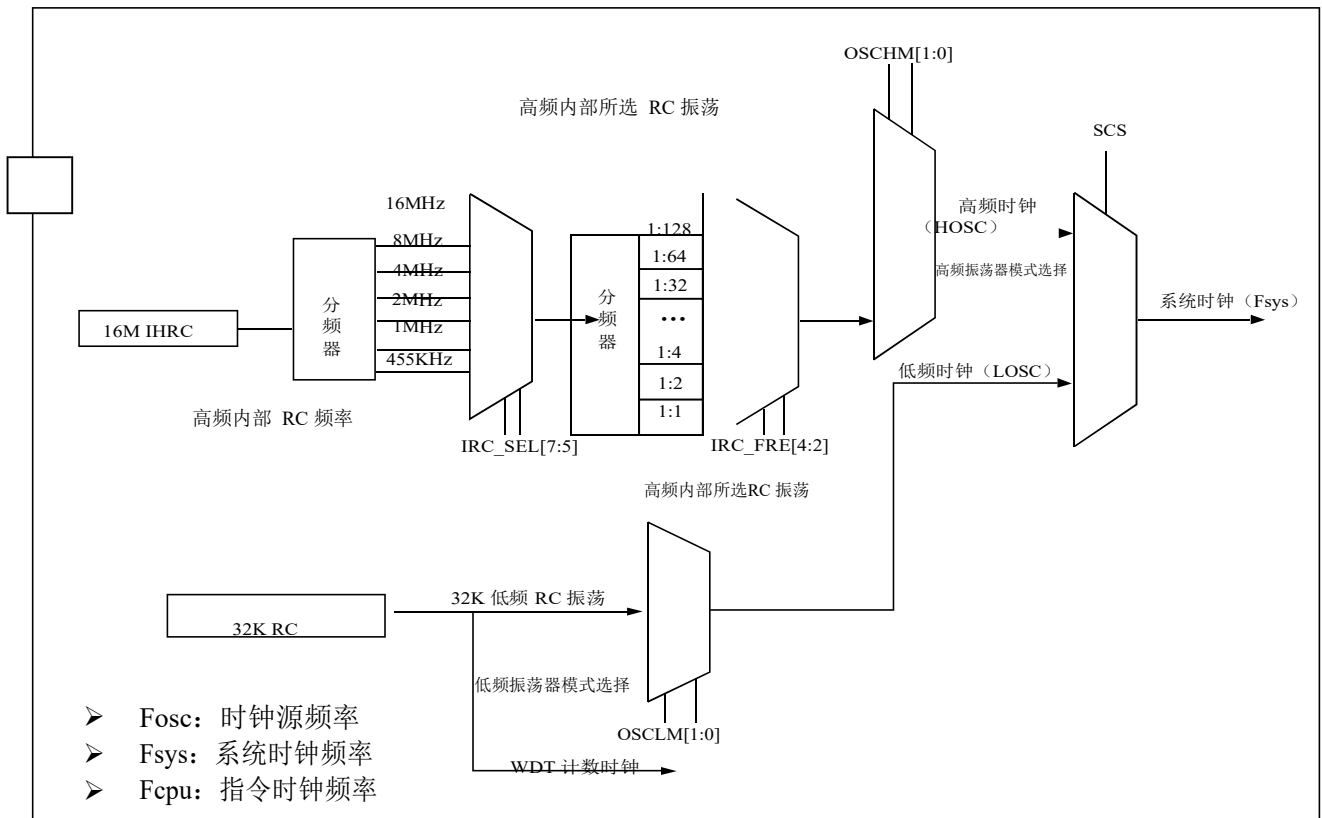
#### 3.1 概述

HC18P018A0 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内部 16MHz RC 振荡电路（IRC 16MHz）提供。低频时钟的时钟源则由内部低频 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。

PB4 为系统时钟输出口，OPTION 配置打开后可输出 Fosc。

#### 3.2 时钟框图



## 3.3 系统高频时钟

系统高频时钟为内部高频RC 振荡器。

### 3.3.1 内部高频RC

内置高频RC 振荡器有 16MHz、8MHz、4MHz、2MHz、1MHz、500KHz 六种可选。由于制造工艺决定了不同芯片的 RC 振荡器频率会有不同，即使每个芯片的 RC 振荡器频率在烧录时已经被烧录器校准到 1% (25°C, VDD=5.0V)，随着 VDD 的变化 RC 振荡器频率也会发生变化，可通过软件操作内部高频时钟调整寄存器进行调整。

寄存器 AUXR

59h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AUXR	POR	BOR	GP	GP	GP	RCTRMEN	GP	PWMCT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	q	q	0	0	0	0	0	0

注： q = 取值视条件而定

Bit [2] RCTRMEN：内部高频 RC 软件校准使能位

1=允许内部高频 RC 软件校准

0=禁止内部高频 RC 软件校准

内部高频 RC 时钟调整寄存器

5Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
HIRCTRM	-	-	内部高频时钟调整寄存器					
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	0	0	q	q	q	q	q	q

注： q = 取值视条件而定

Bit [5:0] 内部高频 RC 振荡器频率调整位（用户必须先使 RCTRMEN=1，频率调整才会生效）

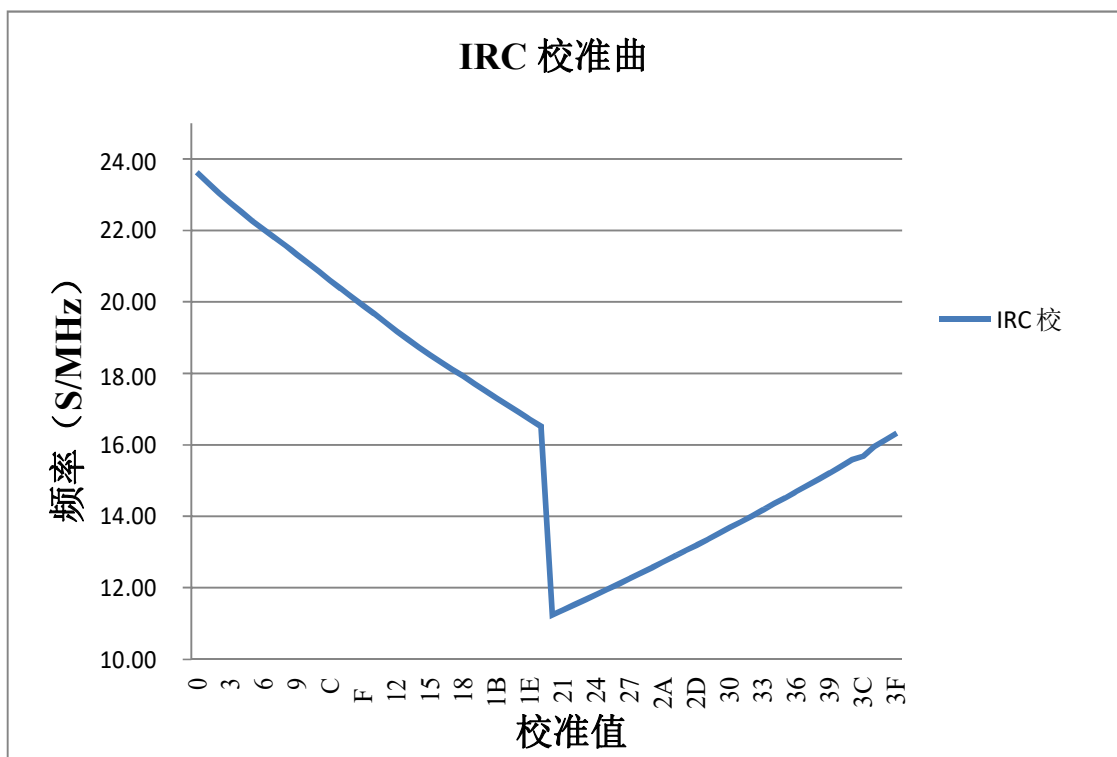
➤ 校准过程：

1. 先读 HIRCTRM，得到烧录器校准该颗芯片后写入的原始校准值。
2. 根据 IRC 校准曲线和 IRC 随 VDD 变化曲线计算需要调整 IRC 的校准值。
3. 然后使能 RCTRMEN，对 HIRCTRM 进行写操作，IRC 调整在下一个指令周期才生效。

➤ 使用方法：

- 1、只有在先使能 RCTRMEN，再写入 HIRCTRM 的情况下，IRC 调整在下一个指令周期才生效。
- 2、当用户有效调整过 IRC 校准值后，即使将 RCTRMEN 禁止，当前 IRC 也会以上一次的校准为准。
- 3、当系统产生复位后，HIRCTRM 的值为烧录器校准该颗芯片时写入的原始校准值。
- 4、系统初次上电复位后，如果 RCTRMEN=0，则 IRC 以原始校准值为当前校准值。

IRC 校准曲线如下图所示：



当 VDD=5V 时，烧录器校准一颗芯片后写入的原始校准值为 0x3A，可以将频率校准到 16MHz。

➤ 举例：

1. 若 VDD 下降到 4V 时 (VDD=4V)，根据 IRC 随 VDD 变化曲线得知频率下降了 0.6MHz，因此目前频率变为 15.4MHz，为保证 VDD 下降不会影响频率的下降，可以先使能 RCTRMEN，根据 IRC 校准曲线，对 HIRCTRM 进行写 0x3E 经行调整。

2. 若 VDD 提升到 5.5V 时 (VDD=5.5V)，根据 IRC 随 VDD 变化曲线得知频率上升了 1.2MHz，因此目前频率变为 16.2MHz，为保证 VDD 提升不会影响频率的提升，可以先使能 RCTRMEN，根据 IRC 校准曲线，对 HIRCTRM 进行写 0x33 经行调整。

注:

1. 烧录器原始寄存器 HIRCTRM 的校准值在 0x00~0x0F 范围, 当写入寄存器的值增大时, 每档位可以降低约 0.25MHz, 反之当写入寄存器的值减小时, 每档位可以提升约 0.25MHz。
2. 烧录器原始寄存器 HIRCTRM 的校准值在 0x10~0x1F 范围, 当写入寄存器的值增大时, 每档位可以降低约 0.20MHz, 反之当写入寄存器的值减小时, 每档位可以提升约 0.20MHz。
3. 烧录器原始寄存器 HIRCTRM 的校准值在 0x20~0x3F 范围, 当写入寄存器的值增大时, 每档位可以提升约 0.15MHz, 反之当写入寄存器的值减小时, 每档位可以降低约 0.15MHz。

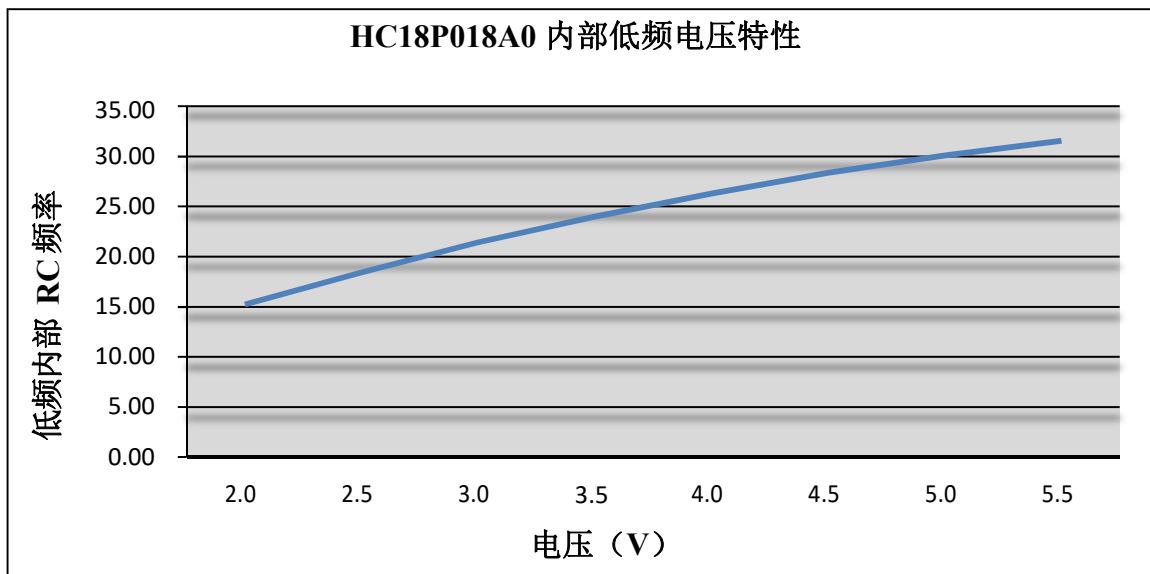
## 3.4 系统低频时钟

低频RC 振荡器: 32KHz (5V 典型值)

### 3.4.1 内部低频RC 振荡器

内部低频 RC 振荡器的频率为 32KHz, 除可供 WDT 使用外, 也可以提供给系统使用。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:



# 4 复位

## 4.1 概述

HC18P018A0 共有四种复位方式：

- 上电复位（POR）
- 外部复位（MCLR Resets，仅在外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）

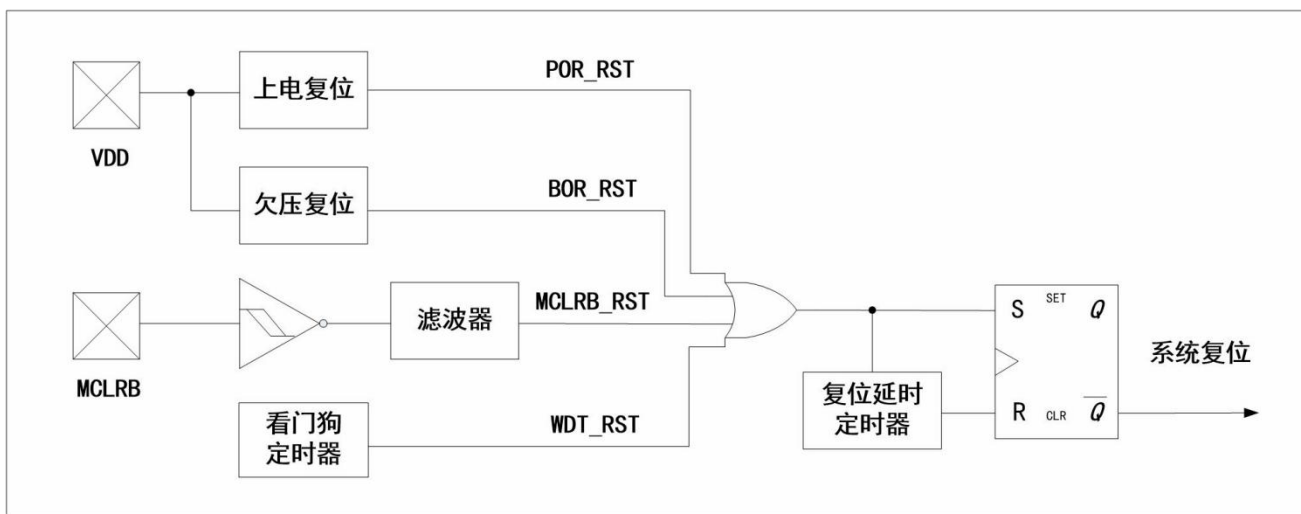
当上述任何一种复位产生时，系统进入复位状态，所有的特殊功能寄存器被初始化，程序停止运行，同时程序计数器（PC）清零。经过上电延时定时器延时后，系统结束复位状态，程序从 000h/3FFh 地址开始执行。STATUS 寄存器的 Bit4（TO 位）及 AUXR 寄存器的 Bit6（BOR 位）、Bit7（POR 位）显示系统复位状态信息，可通过这 3 个标志位判断复位来源，从而控制系统的运行路径。

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 LVR 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：



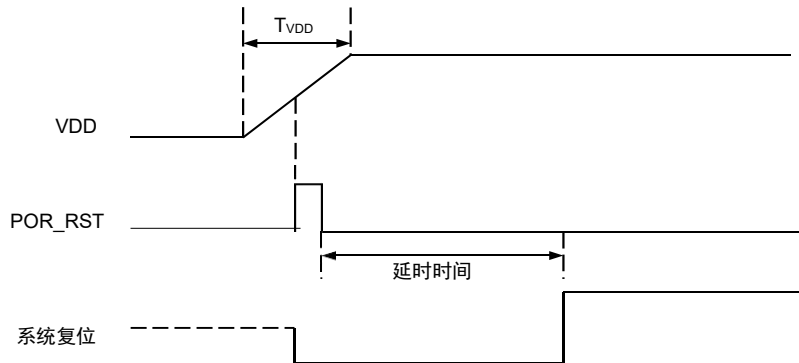
复位延时定时器在复位信号结束后，提供一定时间的延时。

复位方式	复位延时定时器时间（典型值）
上电复位	OPTION 选择
欠压复位	OPTION 选择
外部复位	0.25ms
看门狗定时器复位	OPTION 选择

## 4.2 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 STATUS 寄存器的 Bit4 (TO 位) 及 AUXR 寄存器的 Bit6 (BOR 位)、Bit7 (POR 位) 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。内部高频 RC 振荡器的起振时间最短，外部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。

上电复位示意图：



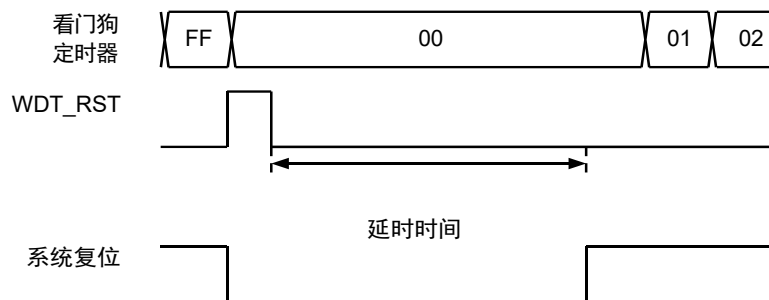
关于上电复位，请注意以下几点：

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。

## 4.3 WDT 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 WDTEN 都为 1 时，才能使能看门狗定时器。

看门狗复位示意图：





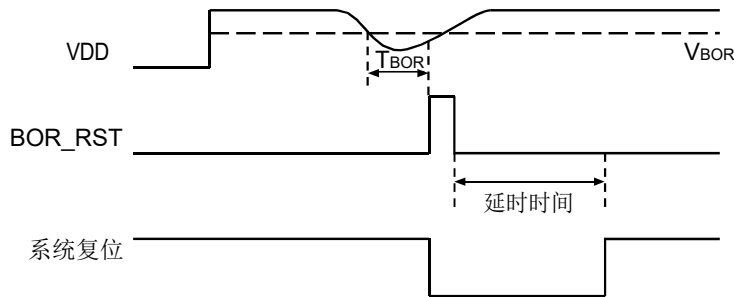
关于看门狗复位使用时，请注意以下几点：

- 1、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。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（WDTEN=1）。
- 2、不建议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、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）。

## 4.4 欠压复位

### 4.4.1 欠压复位的产生

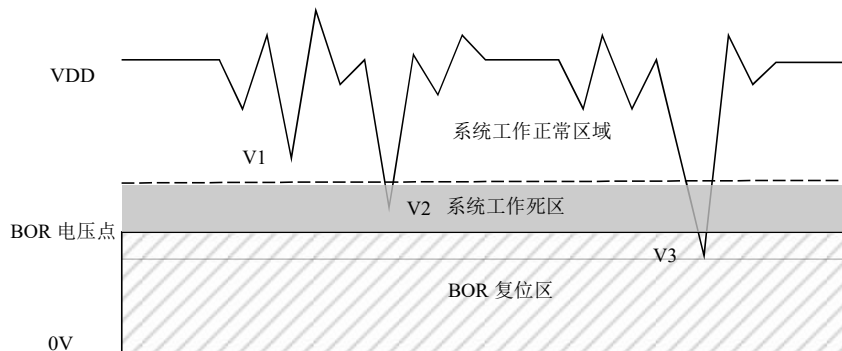
当 VDD 电压下降到  $V_{BOR}$  以下，且持续时间超过  $T_{BOR}$  时，系统产生欠压复位。欠压复位示意图：



注： $T_{BOR}$  需大于 200ns，否则电压跌落时可能不产生欠压复位信号。

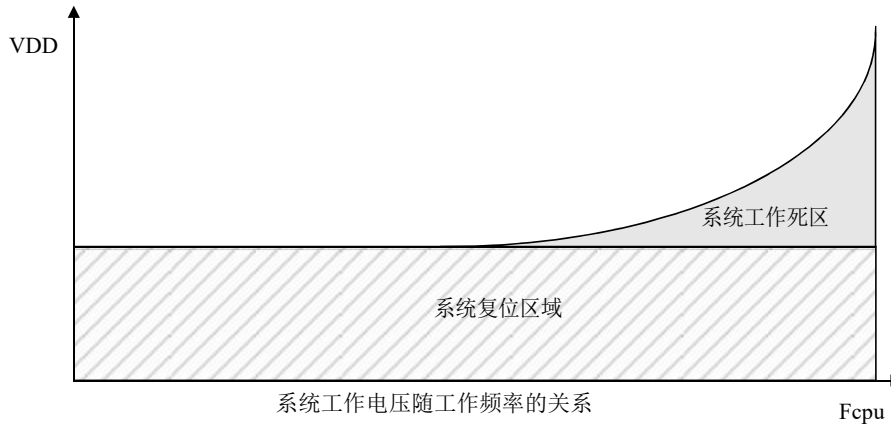
### 4.4.2 工作死区

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，电压值降的非常低。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  $V_1$  时，系统仍处于正常状态；当 VDD 跌至  $V_2$  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  $V_3$ ，且低于 BOR 电压点的时间大于 200ns，系统可正常复位，处于 BOR 电压点的时间小于 200ns，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。



### 4.4.3 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。



### 4.4.4 死区防护

对于死区防护，有以下几点建议：

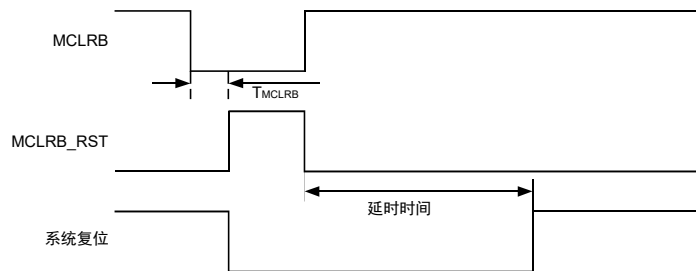
- 合理使用看门狗复位电路
- 降低系统的工作频率
- 合理采用外部复位电路（电压偏移复位电路、外部 IC 复位）

**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。**

## 4.5 外部复位

当外部复位端口 MCLR<sub>B</sub> 使能且输入一个持续时间超过  $T_{MCLR\sub B}$  的低电平时，产生外部复位。

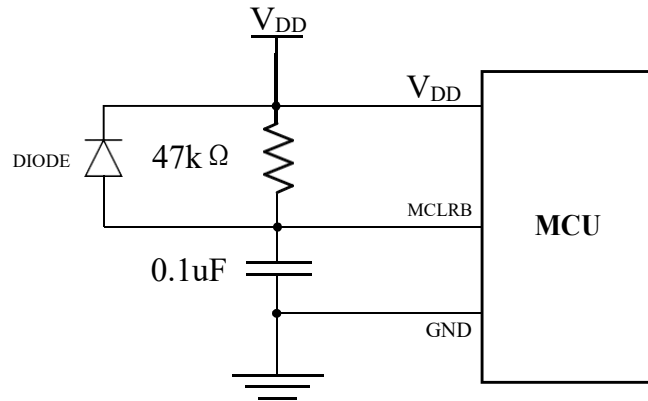
外部复位示意图：



**注： $T_{MCLR\sub B}$  需大于 200 $\mu\text{s}$ （典型值）。**

### 4.5.1 二极管RC 复位电路

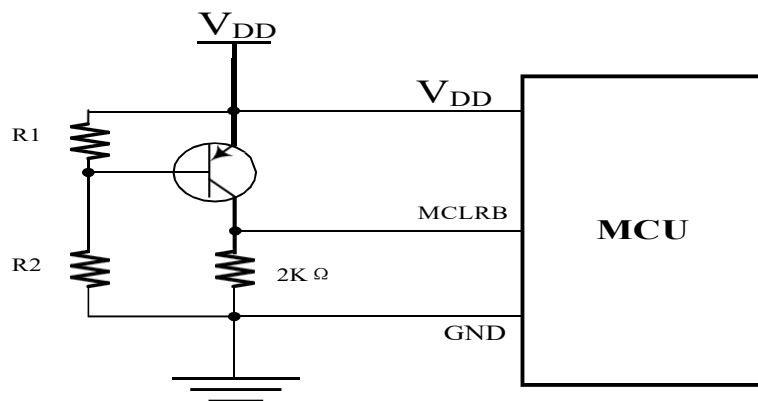
在基本 RC 复位电路上增加一个二极管（DIODE），对于电源异常情况，二极管正向导通使电容快速放电并与VDD保持一致，避免复位引脚持续高电平、系统无法正常复位。



## 4.5.2 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 C 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。



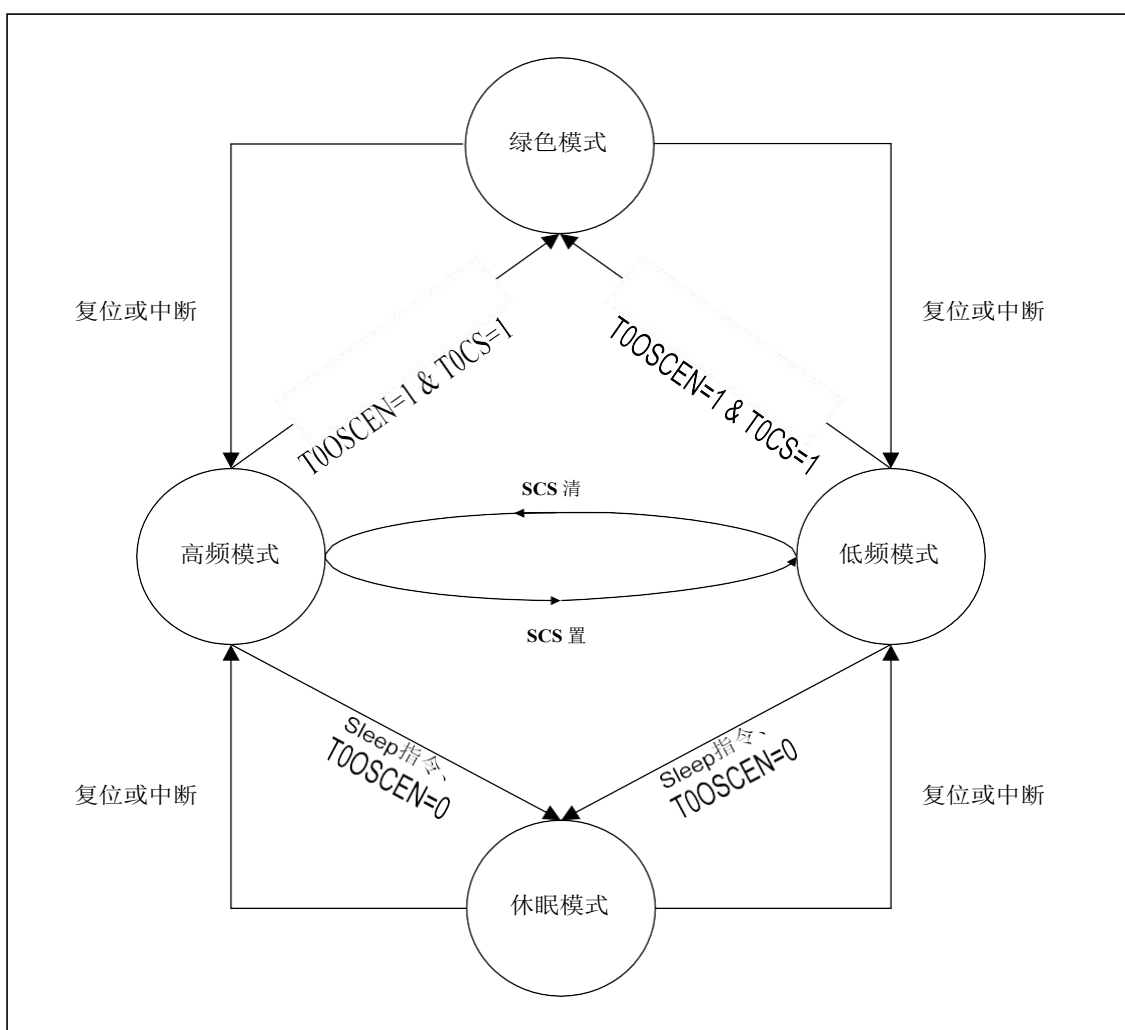
## 5 系统工作模式

### 5.1 概述

HC18P018A0 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。



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 休眠模式

SLEEP 指令可使 MCU 进入休眠模式，同时对 MCU 会产生以下影响：

- 系统主时钟的振荡器停止振荡
- RAM 内容保持不变
- 所有的输入输出端口保持原态不变
- 所有的内部操作全部停止(WDT 不受影响)

以下情况使 MCU 退出休眠模式：

- 有外部中断请求发生
- 有电平变化中断请求发生
- 有 WDT 溢出发生
- 定时器 0 计数溢出发生（RTC 模式开启）
- 定时器 1 外部计数溢出发生
- LVD 的有效检测（在休眠模式下使能LVD 唤醒功能）
- VDD 检测到接入高于 4.4V 的电压, (充电使能位使能)
- 任何形式的系统复位发生 休眠模式下，系统停止了几乎所有的操作，所以整体功耗水平非常低。

注：

1. 进入休眠模式并不会自动打开总中断，但只要有中断请求发生就唤醒系统，如果总中断未打开，系统继续执行下一条指令，否则响应中断服务。
2. 因为 WDT 定时器的时钟源与系统主时钟无关，所以，即使系统进入休眠模式，WDT 定时器仍会工作，但在休眠模式下 WDT 只能产生唤醒信号，并不会产生复位信号。在正常工作下，当 WDT 计数溢出时，芯片复位。

## 5.3 模式切换举例

- 例：高频/低频模式切换到休眠模式。

```
BCF      OSCCON,T0OSCN
SLEEP
```

注：

休眠模式下，只有具有唤醒功能的引脚及复位操作才能将系统唤醒。

- 例：高频模式切换到低频模式。

```
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      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
MOVWF   OPTION
BSF     OPTION,T0CS
BCF     INTFLAG,T0IF
BSF     INTECON,T0IE       ;使能 T0 定时
CLRFB   T0
```

SLEEP

注:

在绿色模式下, T0 具有唤醒功能, 系统唤醒后返回到绿色模式。

➤ 例: 从高频/低频模式切换到绿色模式。  
;T0 定时器定时唤醒, 绿色模式下使能低频振荡器, 定时唤醒时间为 0.5s

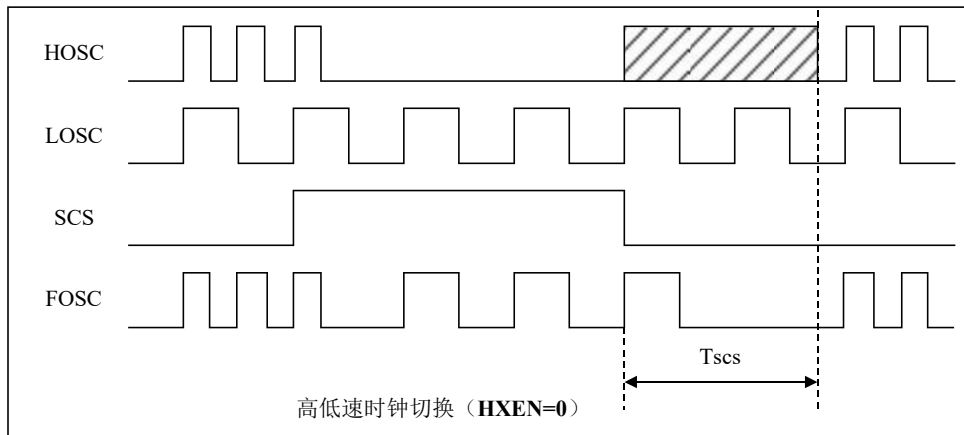
```

MOV LW    0X05
MOV WF    OPTION
BSF       OPTION,T0CS
BSF       OSCCON,T0OSCEN
BCF       INTFLAG,T0IF
BSF       INTECON,T0IE           ;使能 T0 定时
CLRF     T0

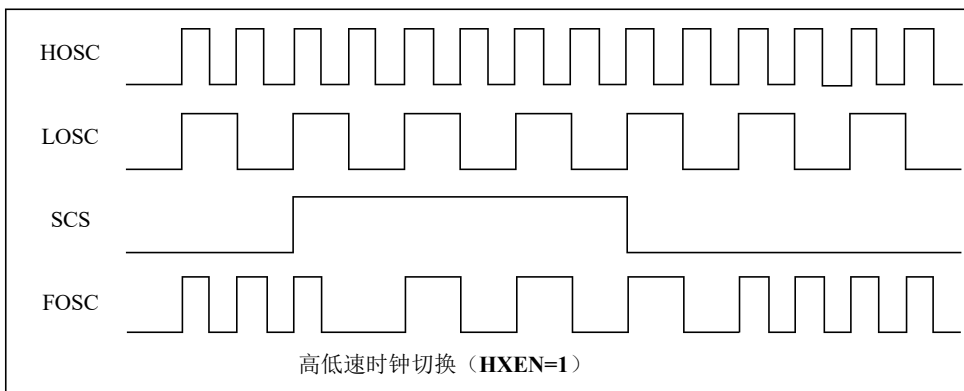
RTC_MODE
SLEEP
BCF       INTFLAG,T0IF           ;0.5s 时间到
...
GOTO     RTC_MODE
    
```

## 5.4 高低频时钟切换

高低频切换时序:



高低频切换时序:



时钟切换时间 (Tscs) 计算:

$$Tscs = \text{高频振荡器起振时间} + \text{高频振荡器稳定时间}$$

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
内部高频RC 振荡器	16 Clock
内部低频RC 振荡器	4 Clock

## 5.5 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，等待的这一段期间称为唤醒时间。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:

$$\text{唤醒时间} = \text{起振时间} + \text{OST 定时时间}$$

不同类型振荡器 OST 定时时间表:

振荡器类型	OST 定时时间
内部高频RC 振荡器	16 Clock
内部低频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。

## 5.6 寄存器 OSCCON

50h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCCON	T0OSCEN	GP	GP	GP	GP	GP	HXEN	SCS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	0	0	0	1	0	0	0	q

注： q = 取值视条件而定

- Bit[7] T0OSCEN: 低频振荡器使能位  
1 = 在低频或绿色模式下使能低频振荡器  
0 = 在低频或绿色模式下禁止低频振荡器
- Bit [6:2] GP: 通用功能寄存器
- Bit[1] HXEN: 高频振荡器使能位  
1 = 在低频或绿色模式下使能高频振荡器  
0 = 在低频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

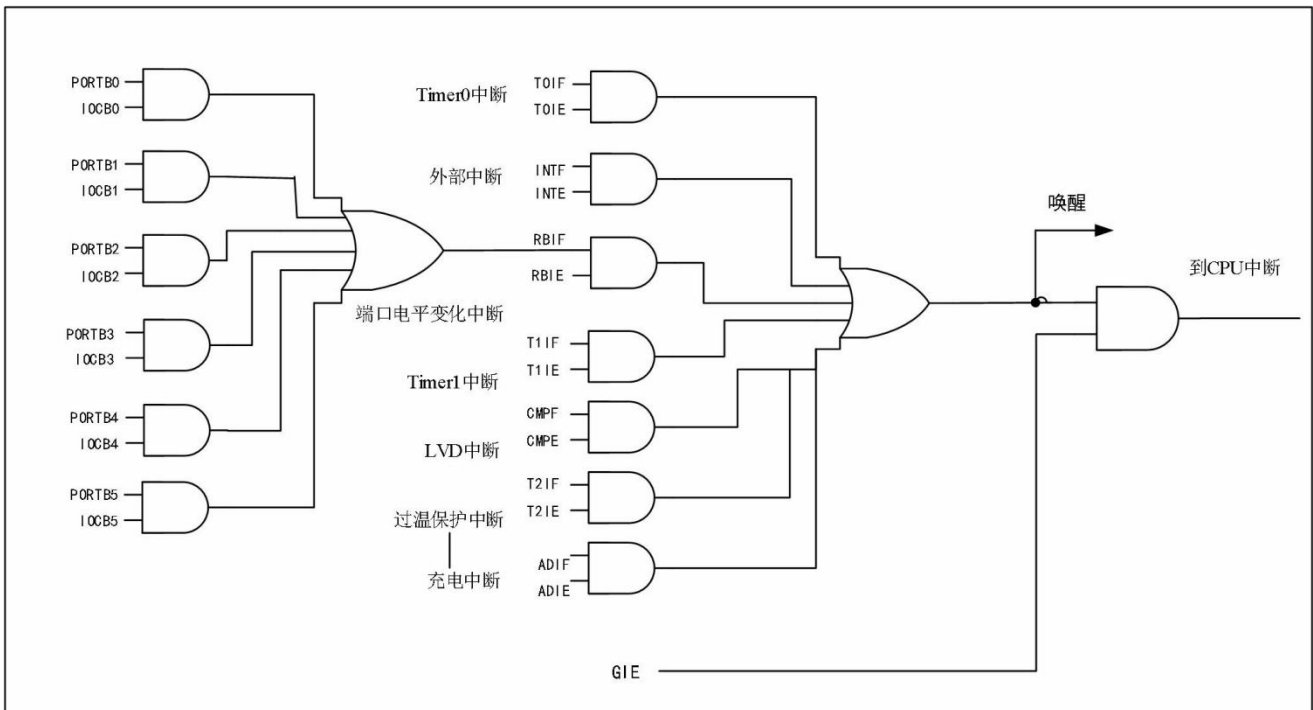
## 6.1 概述

HC18P018A0 提供 7 个中断源：

- Timer0 定时器中断
- INT0 外部中断
- LVD 中断
- 端口电平变化中断
- Timer1 定时器中断
- CHARF 充电中断
- 过温保护中断

系统从高频或低频模式进入休眠模式时，INT0 外部中断、LVD中断、端口电平变化中断、CHARF充电中断和 Timer0/Timer1 中断在计数器模式和定时唤醒模式下可以将单片机唤醒。一旦程序 进入中断，寄存器 INTECON 的位 GIE 将被硬件自动清零以避免再次响应其它中断。系统退出中断后，硬件自动将 GIE 置“1”，以响应下一个中断。

中断示意图：



**注：**

程序响应中断时，GIE 必须处于有效状态。



## 6.2 中断请求和标志寄存器

INTFLAG 中存放 INT0 中断、PORTB 电平变化中断、Timer0 中断请求标志。一旦有中断请求发生，则 INTFLAG 中对应位将被置“1”，该请求被响应后，程序应将该标志位清零。根据 INTFLAG 中中断标志的状态，程序判断是否有中断发生，并执行相应的中断服务。

### INTECON 寄存器

0Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTECON	GIE	-	CHARE	TMPE	-	INTE	PBIE	T0IE
R/W	R/W	R	R/W	R/W	R	R/W	R/W	R/W
POR 的值	0	1	0	0	1	1	1	1

- Bit [7] GIE: 中断总使能  
1 = 使能所有中断  
0 = 屏蔽所有中断
- Bit [5] CHARE: 充电检测中断使能  
1 = 使能充电中断  
0 = 屏蔽充电中断
- Bit [4] TMPE: 温度检测中断使能  
1 = 使能温度检测中断  
0 = 屏蔽温度检测中断
- Bit [2] INTE: 外部中断使能位  
1 = 使能外部中断  
0 = 屏蔽外部中断
- Bit [1] PBIE: 端口电平变化中断使能位  
1 = 使能端口电平变化中断  
0 = 屏蔽端口电平变化中断
- Bit [0] T0IE: Timer0 溢出中断使能位  
1 = 使能 Timer0 溢出中断  
0 = 屏蔽 Timer0 溢出中断

### INTFLAG 寄存器

0Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTFLAG	-	-	CHARF	TMPF	CMPF	INTF	PBIF	T0IF
R/W	R	R	R	R	R	R/W	R/W	R/W
POR 的值	0	0	0	0	0	0	0	0

- Bit [5] CHARF: 充电检测中断标志位  
1 = 充电检测产生中断  
0 = 充电检测未产生外部中断
- Bit [4] TMPF: 温度检测中断标志位  
1 = 温度检测产生中断

- 0 = 温度检测未产生外部中断
- Bit [3] CMPF: 比较器中断标志位
  - 1 = 比较器产生中断
  - 0 = 比较器未产生外部中断
- Bit [2] INTF: 外部中断标志位
  - 1 = INT0 产生外部中断
  - 0 = INT0 未产生外部中断
- Bit [1] PBIF: PORTB 端口电平变化中断标志位
  - 1 = PORTB 产生端口电平变化中断
  - 0 = PORTB 未产生端口电平变化中断
- Bit [0] T0IF: Timer0 溢出中断使能位
  - 1 = Timer0 产生Timer0 溢出中断
  - 0 = Timer0 未产生 Timer0 溢出中断

## 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 0008H）。堆栈层数加 1。

- 例：设置全局中断控制位（GIE）
 

```
BSF      INTECON,GIE      ;使能GIE
```

## 6.4 中断保护

有中断请求发生并被响应后，程序转至 0008H 执行中断子程序。

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

- 例：对W、PCLATH 和 STATUS 进行入栈保护。
 

```
ORG      0000H
GOTO     START
ORG      0008H
GOTO     INT_SERVICE
ORG      0010H

START:
...
INT_SERVICE:
MOVWF   W_TEMP           ;保存
WSWAPF  STATUS,W
MOVWF   STATUS_TEMP      ;保存 STATUS
MOVF    PCLATH,W
MOVWF   PCLATH_TEMP      ;保存PCLATH
...
```

```

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 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。

```

BCF     T0CR,T0CK        ;T0 计数时钟由 T0CS 决
MOV LW  0X15             定
MOVWF   OPTION           ;T0 时钟 = Fcpu / 64
MOV LW  0X40             ;T0 初始值 = 64D
MOVWF   T0
BSF     INTECON,T0IE     ;置 T0 中断使能标志
BCF     INTFLAG,T0IF     ;清 T0 中断标志
BSF     INTECON,GIE      ;使能 GIE

```

➤ 例：T0中断服务程序。

```

ORG     0008H
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

T0ISR:

BTFS    INTFLAG,T0IF     ;检查是否有 T0 中断请求标志
GOTO    EXIT_INT        ;T0IF = 0, 退出中断
BCF     INTFLAG,T0IF     ;清 T0 中断标志
MOV LW  0X40
MOVWF   T0               ;重置 T0 值
...           ; T0 中断程序

EXIT_INT:
MOVF    PCLATH_TEMP,W
MOVWF   PCLATH           ;恢复PCLATH
SWAPE   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 INTO 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，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。

### PCON 寄存器

08h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	WDTEN	EIS	CMPOF	CPNIS3	CPNIS2	CPNIS1	CPNIS0	CMPEN
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
POR 的值	1	0	0	0	0	0	0	0

Bit [6] EIS: INTO 中断引脚功能使能位  
 1 = 使能 PORB0 的 INTO 外部中断功能  
 0 = 屏蔽 PORB0 的 INTO 外部中断功能

### OPTION 寄存器

41h	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	0	0	1	1	1	1	1	1

Bit [6] INTEDG: INTO 中断边沿选择  
 1 = INTO 上升沿中断  
 0 = INTO 引脚下降沿触发中断

➤ 例：INT0 中断请求设置，电平触发。

```

BSF     PCON,EIS    ;PORTB0 的 INTO 外部中断使能
BSF     OPTION,INTEG ;INT0 置为上升沿触发
BCF     INTFLAG,INTF ;INT0 中断请求标志清零
BSF     INTECON,INTE ;使能 INTO 中断
BSF     INTECON,GIE  ;使能 GIE
    
```

➤ 例：INT0 中断。

```

ORG     0008H
GOTO    INT_SERVICE

INT_SERVICE:
...     ;保存 STATUS、W 和 PCLATH
    
```

```

BTSS    INTFLAG,INTF ;检测 TOIF
GOTO    EXIT_INT     ;TOIF = 0, 退出中断
BCF     INTFLAG,INTF ;TOIF 清零
    
```

```

...                               ;INT0中断服务程序
...
EXIT_INT:
...                               ;恢复STATUS、W和PCLATH
RETFIE                           ;退出中断

```

## 6.7 端口电平变化中断

PORTB 电平变化中断时，则无论 PBIE 处于何种状态，相应 PBIF 都会被置“1”。如果 PBIF=1 且 PBIE=1，系统响应应该中断；如果 PBIF=1 而 PBIE=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”。

注意： PORTB 端口变化中断共用中断使能控制信号 PBIE。

09h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0
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] IOCBx: PORTBx 变化中断使能  
 1 = 使能 PORTBx 端口变化中断/唤醒功能  
 0 = 屏蔽 PORTBx 端口变化中断/唤醒功能

➤ 例：PORTB1 电平变化中断请求设置。

```

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      INTFLAG,PBIF ;PROTB 中断请求标志清零
BSF      INTECON, PBIE ;使能 PROTB 中断
BSF      INTECON, GIE ;使能 GIE

```

➤ 例：PORTB 中断。

```

ORG      0008H
GOTO     PORTB_SERVICE

```

```

PORTB_SERVICE:
...                               ;保存 STATUS、W 和 PCLATH
BTFSS   INTFLAG,PBIF ;检测 PBIF
GOTO    EXIT_INT      ;PBIF = 0, 退出中断
MOVWF   PORTB,W       ;读 PORTB 端口
BCF     INTFLAG,PBIF  ;PBIF 清零
...                               ;PORTB 电平变化中断服务程序

```

```

EXIT_INT:      ...
               ...
               RETFIE      ;恢复 STATUS、W 和 PCLATH
               ;退出中断
    
```

**注:**

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

➤ PORTB1 中断唤醒。

```

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     INTFLAG,PBIF ;PROTB 中断请求标志清零
BSF     INTECON,PBIE ;使能 PROTB 中断
SLEEP
BCF     INTECON,PBIE ;如未使能 GIE,直接执行下一句,否则进入中
MOVF    PORTB,W      ;读 PORTB 端口
...
;其他程序
    
```

**注:**

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

## 6.8 Timer1 中断

当 T1 的值和 PR2 的值相同时，Timer1 中断被触发，则无论 T1IE 处于何种状态，T1IF 都会被置“1”。如果 T1IF=1 且 T1IE=1，系统响应该中断；如果 T1IF=1 而 T1IE=0，系统并不会执行中断服务。

4Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0CR	T0CK	-	-	-	-	-	T1IE	T1IF
R/W	R/W	R	R	R	R	R	R/W	R/W
POR 的值	0	0	0	0	0	0	0	0

- Bit[1] T1IE: Timer1 溢出中断标志位  
 1 = 使能 Timer1 溢出中断  
 0 = 禁止 Timer1 溢出中断
- Bit [0] T1IF: Timer1 溢出中断标志位  
 1 = Timer1 计数寄存器溢出  
 0 = Timer1 计数寄存器未溢出

➤ 例: TIMER1 中断请求设置。

```

MOVLW    0XFF
MOVWF    T1
MOVWF    T1LOAD    ;设置T1 周期
MOVLW    0X04
MOVWF    T1CON     ;设置分频比
BSF      T0CR,T1IE ;使能TIMER1 中断
BSF      INTECON,GIE
BSF      T1CON,T1EN ;使能TIMER1
    
```

➤ 例：TIMER1 中断

```

ORG      0008H
GOTO     T1INT_SERVICE

T1INT_SERVICE:
    ... ;保存STATUS、W 和 PCLATH
    BTSS    T0CR,T1IF ;检测T1IF
    GOTO    EXIT_INT  ;T1IF = 0, 退出中断
    BCF     T0CR,T1IF ;T1IF 清零
    ... ;TIMER1 中断服务程序
    ...

EXIT_INT:
    ...
    RETFIE ;退出中断
    
```

## 6.9 LVD/CMP 中断

比较器 CMP 的输出产生下降沿或上升沿时，触发 CMP 中断，中断标志（CMPF）将被置 1，若中断总使能位 GIE 为 1 且 CMP 中断使能位（CMPIE）为 1，则产生 CMP 中断。

### LVD 中断相关寄存器

0Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTFLAG	-	-	CHARF	TMPF	CMPF	INTF	PBIF	TOIF
R/W	R	R	R	R	R/W	R/W	R/W	R/W
POR 的值	0	0	0	0	0	0	0	0

Bit [3] CMPF: 比较器中断标志位  
 1 = 产生 CMP 中断(需软件清零)  
 0 = 无 CMP 中断产生

51h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMPCR	GP	GP	GP	GP	CMPWK	CMPIE	CMPIES	CPPIS
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 [2]    CMPIE: CMP 中断使能位  
0 = 屏蔽 CMP 中断。  
1 = 使能 CMP 中断



## 6.10 充电中断

VDD 检测到接入高于 4.4V 的电压时，触发 CHARE 中断，中断标志（CHARF）将被置 1，若中断总使能位 GIE 为 1 且中断使能位（CHARGEN）为 1，则产生充电中断。该中断可唤醒 SLEEP。该中断标志不可软件清 0。

充电中断相关寄存器：

0Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTFLAG	-	-	CHARF	TMPF	CMPF	INTF	PBIF	T0IF
R/W	R	R	R	R	R/W	R/W	R/W	R/W
POR 的值	0	0	0	0	0	0	0	0

Bit [5] CHARF：充电中断标志位  
1 = 产生充电中断  
0 = 无充电中断产生

54h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CHARGE1	CHARGEN	TEMPEN					S9	S8
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] CHARGEN：充电使能位  
0 = 屏蔽充电  
1 = 使能充电

## 6.11 过温保护中断

芯片温度过高时，触发 过温保护 中断，中断标志（TMPF）将被置 1，若中断总使能位 GIE 为 1 且中断使能位（TEMPE）为 1，则产生过温保护中断。

过温保护中断相关寄存器

0Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTFLAG	-	-	CHARF	TMPF	CMPF	INTF	PBIF	T0IF
R/W	R	R	R	R	R/W	R/W	R/W	R/W
POR 的值	0	0	0	0	0	0	0	0

Bit [4] TEMPF：过温保护中断标志位  
1 = 产生过温保护中断(需软件清零)  
0 = 无过温保护中断产生

54h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CHARGE1	CHARGEN	TEMPEN					S9	S8
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 [6] TEMPEN：过温保护中断使能位  
0 = 屏蔽过温保护中断  
1 = 使能过温保护中断

## 7 I/O 端口

### 7.1 I/O 端口模式

端口方向寄存器

46h	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

Bit [7:0] TRISBx: PORTBx 模式控制位

- 1 = 输入模式
- 0 = 输出模式

特别说明: PORTB3 设为外部复位端口时, 此时端口施密特有效并且上拉电阻有效。

➤ 例: I/O 模式选择。

```

MOVLW    0XFF           ;所有端口设为输入模式
MOVWF    TRISB

CLRF     TRISB          ;所有端口设为输出模式
BCF      TRISB,2        ;PORTB2 设为输出模式
BSF      TRISB,2        ;PORTB2 设为输入模式
    
```

### 7.2 I/O 上拉模式

PHCON 寄存器

0Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PHCON	PHB7	PHB6	PHB5	PHB4	PHB3	PHB2	PHB1	PHB0
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 7:0] PHBx: PORTBx 上拉控制

- 1 = 屏蔽PORTBx 输入上拉功能
- 0 = 使能PORTBx 输入上拉功能

注:

1. 当 I/O 口是输出时, 上拉无效。
2. I/O 禁止浮空状态, 输入状态需设定内部上拉或下拉电阻。

➤ 例: I/O 口的上拉电阻。

```

CLEF     PHCON          ;使能PORTB 上拉
BSF      PHCON,2        ;禁止PORTB2 上拉
    
```

## 7.3 I/O 下拉模式

### PDCON 寄存器

0Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDCON	GP	PDB2	PDB1	PDB0	GP	GP	GP	GP
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] GP: 通用功能寄存器位  
 Bit [6:4] PDBx: PORTBx 下拉控制  
 1 = 屏蔽 PORTBx 输入下拉功能  
 0 = 使能 PORTBx 输入下拉功能  
 Bit [3:0] GP: 通用功能寄存器位

### PDCON1 寄存器

52h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDCON1	PDB7	PDB6	PDB5	PDB4	PDB3	-	GP	GP
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
POR 的值	1	1	1	1	1	1	1	1

- Bit [1:0] GP: 通用功能寄存器位  
 Bit [7:3] PDBx: PORTBx 上拉控制  
 1 = 屏蔽 PORTBx 输入下拉功能  
 0 = 使能 PORTBx 输入下拉功能

#### 注:

1. 当 I/O 口是输出时，下拉无效。
2. 当上拉打开时，下拉无效。

➤ 例: I/O 口的下拉电阻。

```
CLRF      PDCON
CLRF      PDCON1      ;使能所有下拉
BSF      PDCON,6      ;禁止 PORTB2 下拉
```

## 7.4 I/O 开漏模式

### ODCON 寄存器

0Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ODCON	ODB7	ODB6	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0
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] ODBx: PORTBx 开漏控制寄存器  
 1 = PORTBx 使能开漏输出  
 0 = PORTBx 为普通 I/O

➤ 例: I/O 口的开漏模式。

```
MOVLW    0XFF
MOVWF    ODCON      ;PORTB 所有端口设为开漏模式
```

```

MOVLW    0X00
MOVWF    ODCON    ;PORTB 所有端口设为普通I/O 模式
BCF      ODCON,2  ;禁止 PORTB2 开漏模式
BSF      ODCON,2  ;使能PORTB2 开漏模式
    
```

## 7.5 I/O 端口数据寄存器

### PORTB 端口数据寄存器

06h	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/W	R/W	R/W	R/W	R/W	R/W
POR 的值	x	x	x	x	x	x	x	x

- 例：从输入口读取数据。

```

MOVWF    PORTB,W    ;从 PORTB 读数据
    
```

- 例：写数据到输出端。

```

MOVLW    0XFF    ;立即数 0XFF 写入所有输出口
MOVWF    PORTB
    
```

- 例：端口位操作。

```

BSF      PORTB,2    ;PORTB2 置 1
BCF      PORTB,2    ;PORTB2 置 0
    
```

# 8 定时器

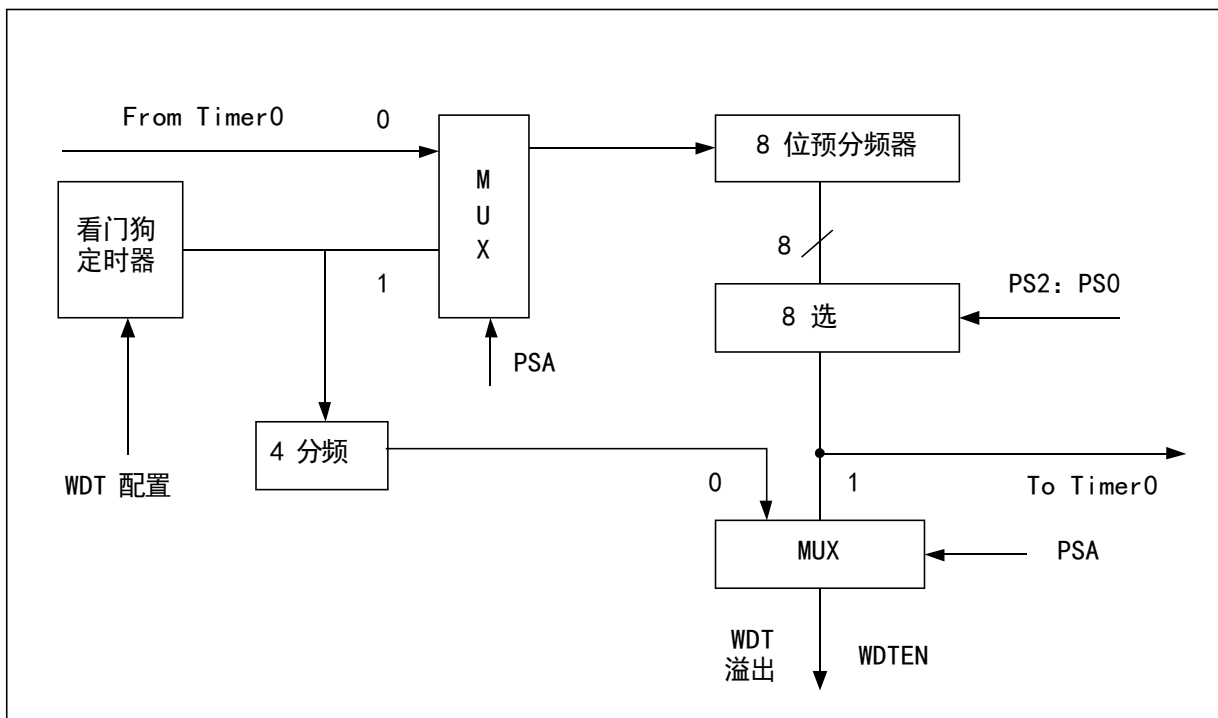
## 8.1 看门狗定时器

WDT 定时器的时钟源于内部低频 RC 振荡器，并可以选择是否经过预分频器。WDT 定时器可以用来产生 WDT 复位或唤醒休眠模式。WDT 定时器是否开启由OPTION 中的 WDTE 和软件的 WDTEN 位共同决定。只有 WDTEN 为 0 时，WDT 定时器被软禁止，为 1 时软使能，若要 WDT 使能还需要 OPTION 的 WDTE 使能。

因为 WDT 定时器的时钟源与系统主时钟无关，所以，即使系统进入休眠模式，WDT 定时器仍会工作，但在休眠模式下 WDT 只能产生唤醒信号，并不会产生复位信号。在正常工作下，当 WDT 计数溢出时，芯片复位。

WDT 的基本溢出时间由 OPTION 的 TWDT 决定，无分频的周期范围是 4.5ms~288ms。WDT 和 T0 共用分频器，当分频器给T0时，WDT为1分频（无分频）；反之当分频器给WDT时 T0为1分频（无分频），由 PSA、PS[2:0]决定。若要更长的时间可对 WDT 进行分频，分频后 WDT 溢出时间为基本溢出时间的分频倍数。例如 OPTION 中 TWDT 选择的基本时间为 18ms，软件进行 4 分频，则溢出时间为  $18*4=72ms$ 。

看门狗定时器和预分频器框图



PCON 寄存器

08h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	WDTEN	EIS	LVDF	LVDSSEL3	LVDSSEL2	LVDSSEL1	LVDSSEL0	LVDEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	1	0	0	0	0	0	0	0

Bit[7] WDTEN: 看门狗使能位  
 1 = 软件使能WDT  
 0 = 软件屏蔽 WDT 功能

看门狗定时器使能需要 WDT 定时器配置字设置使能，并且系统寄存器 WDTEN 位软件置 1。当系统处于休眠模式，看门狗定时器溢出将唤醒 SLEEP 并使其返回高频模式，程序从 SLEEP 指令下一条开始执行

看门狗定时器应用注意事项如下：

- 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；
- 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情况；
- 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：对看门狗定时器操作，看门狗定时器使能和清零。

```
BSF          PCON,7          ;软件使能 WDT
...
CLRWDW      ;看门狗定时器清零
```

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

- 例：看门狗在主程序中的应用。

```
MAIN:
BSF          PCON,WDTEN      ;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 或 RAM 出错，进入出错处理程序
CLRWDW      ;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          PCON,WDTEN      ;软件屏蔽看门狗功能
SLEEP       ;进入休眠模式
NOP
BSF          PCON,WDTEN      ;唤醒后，重新使能看门狗功能
```

## 8.2 Timer0 定时/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
- 溢出中断
- 支持 RTC 模式

定时器 Timer0 由 8 位计数器 T0，控制寄存器 OPTION 组成。

T0 的计数时钟来自系统时钟 Fcpu 或者外部管脚 T0CKI。预分频器为定时器 T0 与 WDT 定时器共用，当 PSA=0 时，预分频器分配给 T0 使用；PSA=1 时，预分频器分配给 WDT 使用。分频系数由 PS[2:0] 决定。

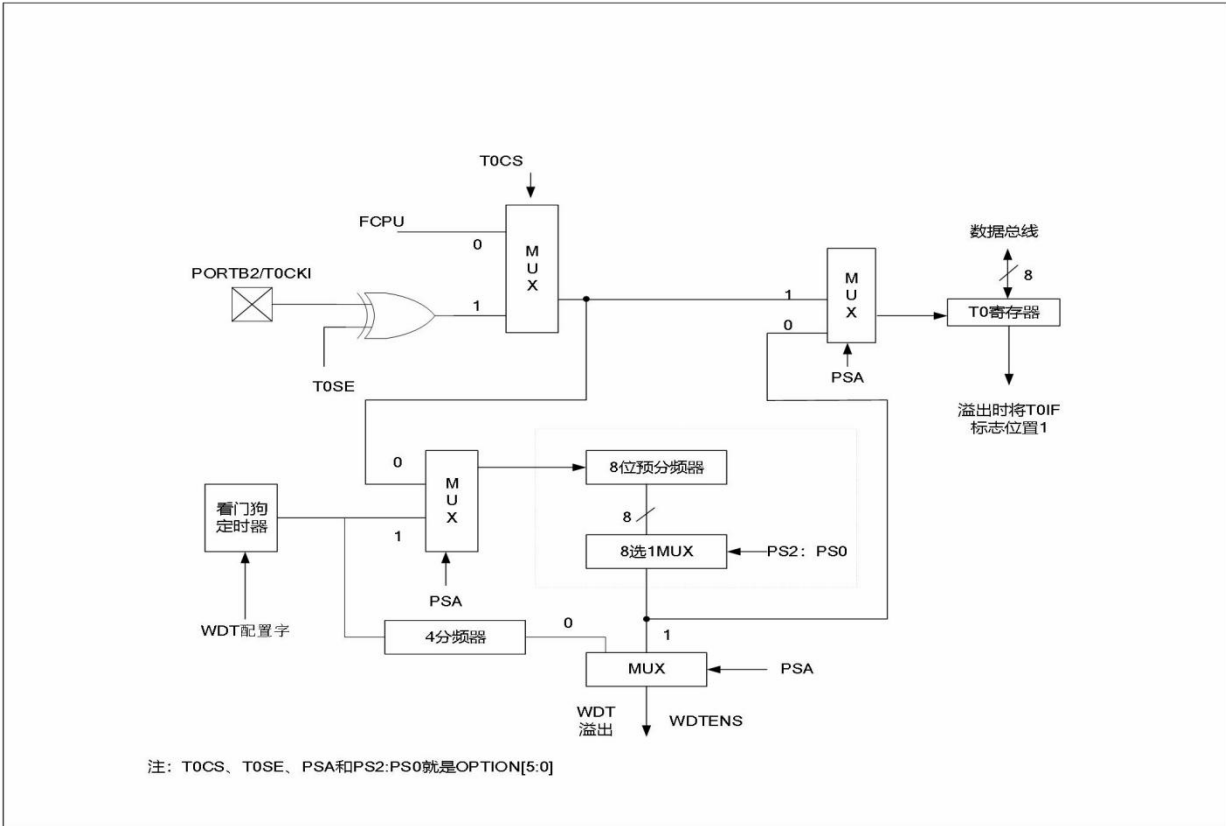
T0 是一个递增计数器，它的值可以读写，当计数到从 FF 溢出到 0 时，产生 T0 溢出信号，将中

断标志位T0IF 置 1。

T0 计数周期公式： $T0 = \text{分频数} / F_{\text{cpu}}$ 。

当时钟源为T0CKI 时，T0 在 SLEEP 状态下继续工作，计数溢出可唤醒 CPU。

Timer0 模块和预分频器（与 WDT 共享）框图



**OPTION 寄存器**

41h	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	0	0	1	1	1	1	1	1

Bit[7] 保留位

Bit[5] T0CS: Timer0 时钟源选择

1 = T0CKI (当Timer0 选择 T0CKI 作为计数时钟时, T0CKI 口由硬件设为施密特端口)

0 = Fcpu

Bit[4] T0SE: Timer0 计数沿选择

1 = 下降沿计数

0 = 上升沿计数

Bit[3] PSA: 预分频分配

1 = WDT

0 = Timer0

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, Timer 0 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 0。

Timer0 预分频比选择

PS[2:0]	Timer0 预分频比	WDT 预分频比
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

T0CR 寄存器

4Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0CR	T0CK	-	-	-	-	-	T1IE	T1IF
R/W	R	R	R	R	R	R	R/W	R/W
POR 的值	0	0	0	0	0	0	0	0

Bit[7] T0CK: T0 时钟选择  
 1 = T0 时钟保留位 (T0 时钟无效)  
 0 = T0 计数时钟由 T0CS 决定

T0 寄存器

01h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0	Timer0 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	x	x	x	x	x	x	x	x

Bit[7:0] T0 的值，用于设定定时时间。

注：

1. Fcpu 即 CPU 的运行速度，若系统选择高频时钟 4MHz，4T 时钟模式，则  $F_{cpu} = 4MHz/4 = 1MHz$ 。系统选择高频时钟 4MHz，2T 模式，则  $F_{cpu} = 4MHz/2 = 2MHz$ 。
2. Timer0 时钟源选择为外部时钟源T0CKI 时，具有唤醒功能。

➤ 例：利用 Timer0 定时器产生 1ms 定时，配置字选择 4MHz 系统时钟，4T 模式。

```

...
BCF      T0CR,T0CK      ;T0 计数时钟由 T0CS 决定
MOVLW   0X11
MOVWF   OPTION         ;T0CS = 0,PSA=0,PS=1,Fcpu = 1MHz
CLRF    T0              ;清除 Timer0 定时器
MOVLW   0X06           ;Timer0 的中断周期
MOVWF   T0             ;Tt0ov = (256-6)*预分频比/Fcpu = 1ms

BCF     INTFLAG,T0IF
    
```



```

BSF      INTECON,T0IE      ;使能 Timer0 中断
BSF      INTECON,GIE      ;使能中断功能
...
ISR_T0
BCF      INTFLAG,T0IF     ;清除 Timer0 中断标志
MOVLW   0X06              ;Timer0 的中断周期
MOVWF   T0                ;Tt0ov = (256-6)*预分频比/Fcpu = 1ms
BSF      T0msFlag         ;1ms 定时标志
GOTO    T0_EXIT
    
```

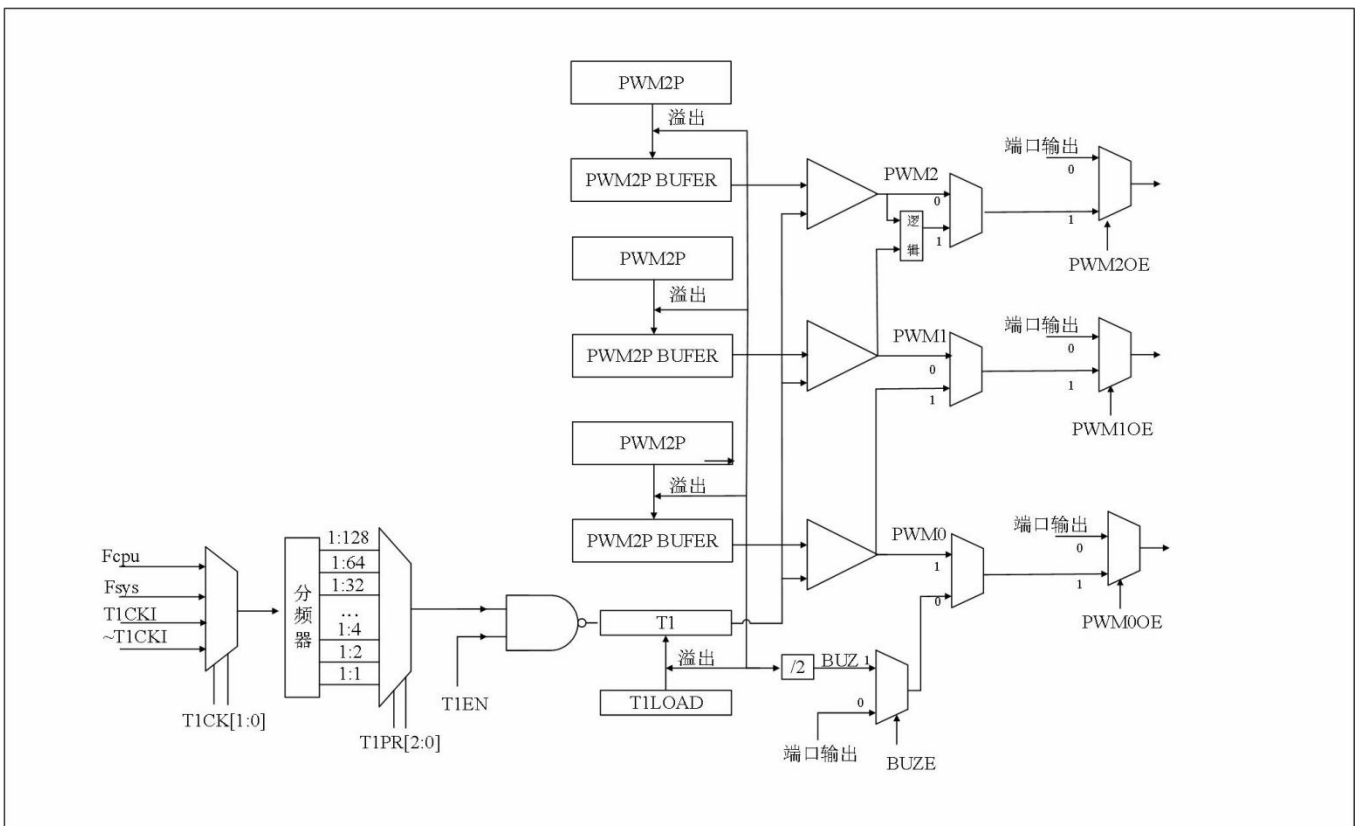
## 8.3 Timer1 定时/计数器

### 8.3.1 功能概述

定时/计数器 T1 包含 1 个可编程预分频器、控制寄存器、重载寄存器及比较寄存器。

- 可通过预分频比设置频率
- 通过重载寄存器设置周期
- 通过比较寄存器设置 PWM 占空比（仅 PWM 模式）
- BUZ 功能
- 溢出中断功能
- 溢出唤醒功能

Timer1 模块框图



### 8.3.2 T1 使用操作说明

T1CK[1:0]可选择T1 的时钟源， T1PR[2:0]可选择T1 的预分频比，所选中的时钟源通过预分频器后产生 T1 的时钟。

当 T1 递减到 0 时，此时产生T1 溢出中断请求标志 T1IF 置 1，重载寄存器值自动置入 T1，PWM0P 的值写入缓冲器 PWM0P BUFER 用于新的占空比波形生成，BUZ 信号反相。

通过 T1PR[2:0]可选择时钟源的分频比，可选择范围为 1~128 分频，对T1 的写操作将使预分频器清零，分频比保持不变。

PWM0 操作说明：

当 PWM0OE=1 时，将输出 PWM 波形，当 T1 计数到与 PWM0P 相等时，PWM0 输出置 1；当 T1 计数溢出时，PWM0 输出清 0。PWM0 占空比的计算如下：

PWM0 高电平时间= (PWM0P) \* T1 计数时钟周期

PWM0 周期 (T1 的溢出周期) = (T1LOAD+1) \*T1 的计数时钟周期

PWM0 占空比= (PWM0P/ (T1LOAD+1) )

当 BUZOE=1 且 PWM0OE=0 时，输出 BUZ 信号，BUZ 信号的输出频率为 T1 溢出频率的 2 分

PWM1 和 PWM2 与 PWM0 共用 T1LOAD 设置周期，通过 PWM0P/PWM1P/PWM2P 分别设置占空比，占空比设置方法同 PWM0。

当 PWMMD=0 时，PORTB1 输出 PWM1 同时 PORTB0 输出 PWM2；当 PWMMD=1 时，PORTB1 输出 PWM0 同时 PORTB0 输出 PWM1^PWM2。

注：

当 PWM0OE、PWM0E 不全部使能时，PWM0 输出当前配置端口的端口数据寄存器状态。PWM1/PWM2 的 PWM0OE、PWM0E 操作，同 PWM0。

### 8.3.1 T1 相关寄存器

Timer1 控制寄存器

4Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CON	T1EN	PWM0E	BUZE	T1CK1	T1CK0	T1PR2	T1PR1	T1PR0
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] T1EN: T1 使能控制

0 = 关闭T1

1 = 启动T1

Bit [6] PWM0E: PWM0 选择

0 = 禁止PWM0 输出，PORTB5 端口作为I/O 口

1 = 允许PWM0 输出 (PWM0OE=0)，PORTB5 端口输出PWM0 信号

Bit [5] BUZE: BUZ 选择

0 = 禁止BUZ 输出，PORTB2 端口作为I/O 口

1 = 允许 BUZ 输出 (PWM0E=0)，PORTB2 端口输出BUZ 信号

Bit [4:3] T1CK[1:0]: T1 时钟源选择

T1CK[1:0]	T1 时钟源
00	Fcpu
01	Fsys
10	T1CKI 上升沿
11	T1CKI 下降沿

Bit [2:0] T1PR[2:0]: T1 预分频倍数选择

T1PR[2:0]	Timer1 预分频比
000	1 : 1
001	1 : 2
010	1 : 4
011	1 : 8
100	1 : 16
101	1 : 32
110	1 : 64
111	1 : 128

**注:**

1. 如果 OPTION 选择 8M/2T, 则  $F_{osc}=16M$ ,  $F_{sys}=8M$ ,  $F_{cpu}=4M$ 。
2. 当 TIMER1 选择 T1CKI 作为计数时钟时, T1CKI 口由硬件设为施密特端口。

#### PWM 控制寄存器

48h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMCON	PWM0OE	PWM1OE	PWM2OE	PWMCK	PWMMD	PWMINV	PWM1E	PWM2E
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] PWM0OE: PWM0 输出选择

- 0 = 允许PWM0/BUZ 输出, PORTB5 端口输出PWM0/BUZ 信号
- 1 = 禁止 PWM0/BUZ 输出, PORTB5 端口作为 I/O 口

Bit [6] PWM1OE: PWM1 输出选择

- 0 = 禁止PWM1 输出, 端口作为 I/O 口
- 1 = 允许PWM1 输出, 端口输出PWM1 信号

Bit [5] PWM2OE: PWM2 输出选择

- 0 = 禁止PWM2 输出, 端口作为 I/O 口
- 1 = 允许PWM2 输出, 端口输出PWM2 信号

Bit [4] PWMCK: T1 时钟倍频选择

- 0 = T1 时钟不倍频
- 1 = T1 时钟倍频(T1PR=000 时有效), 此控制位对所有所选T1 时钟源有效。

Bit [3] PWMMD: PWM 输出选择

- 0 = PORTB1 输出PWM1 同时PORTB0 输出PWM2
- 1 = PORTB1 输出 PWM0 同时 PORTB0 输出 (PWM1 异或 PWM2) 或 (PWM1 同或 PWM2)

Bit [2] PWMINV: PWM 逻辑操作模式选择

- 0 = PWM1 同或 PWM2
- 1 = PWM1 异或 PWM2

Bit [1] PWM1E: PWM1 使能选择

- 0 = 关闭PWM1
- 1 = 开启 PWM1

Bit [0] PWM2E: PWM2 使能选择

- 0 = 关闭PWM2
- 1 = 开启 PWM2

**T1 计数寄存器**

4Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1	Timer1 定时计数寄存器							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	1	1	1	1	1	1	1	1

Bit [7:0] Timer1 的值

**T1 重载寄存器**

4Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1LOAD	Timer1 重载寄存器							
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

Bit [7:0] 用于设置 Timer1 的重载值

**PWM0P 占空比寄存器**

4Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0P	PWM0 占空比设置寄存器							
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

Bit [7:0] 用于设置 PWM0 的高电平时间

**PWM1P 占空比寄存器**

49h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1P	PWM1 占空比设置寄存器							
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

Bit [7:0] 用于设置 PWM1 的高电平时间

**PWM2P 占空比寄存器**

4Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2P	PWM2 占空比设置寄存器							
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

Bit [7:0] 用于设置 PWM2 的高电平时间

**注:**

当 T1EN=0 时, 写 T1LOAD 将自动加载到 T1 中; 当 T1EN=1 时, 写 T1LOAD 不会加载 T1 中, 在 T1 溢出时自动加载到 T1 中。

**PWM 端口控制寄存器 AUXR**

59h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AUXR	POR	BOR	GP	GP	GP	RCTRMEN	GP	PWMCT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	q	q	0	0	0	0	0	0

Bit [0] PWMCT: PWM0/1/2 输出端口配置位

- 0 = PWM0、BUZ 配置在 PORTB5  
PWM1 配置在 PORTB1  
PWM2 配置在 PORTB0
- 1 = PORTB5 作为普通 I/O 口  
PORTB1 作为普通 I/O 口  
PORTB0 作为普通 I/O 口

➤ 例：利用 Timer1 定时器产生 1ms 定时，配置字选择 4MHz 系统时钟，4T 模式。

```

...
MOVLW    0XFA
MOVWF    T1
MOVWF    T1LOAD           ;Tt1ov = (250)*预分频比/Fcpu = 1ms
MOVLW    0X82
MOVWF    T1CON           ;使能 T1 预分频比选择 1:4
MOVLW    0X00
MOVWF    PWMCON
BSF      T0CR,T1IE       ;使能 T1 中断
BSF      INTECON,GIE     ;使能中断功能
...
ISR_T1:
BCF      T0CR,T1IF       ;清除 Timer1 中断标志
BSF      T1msFlag        ;1ms 定时标志
GOTO     T1_EXIT
    
```

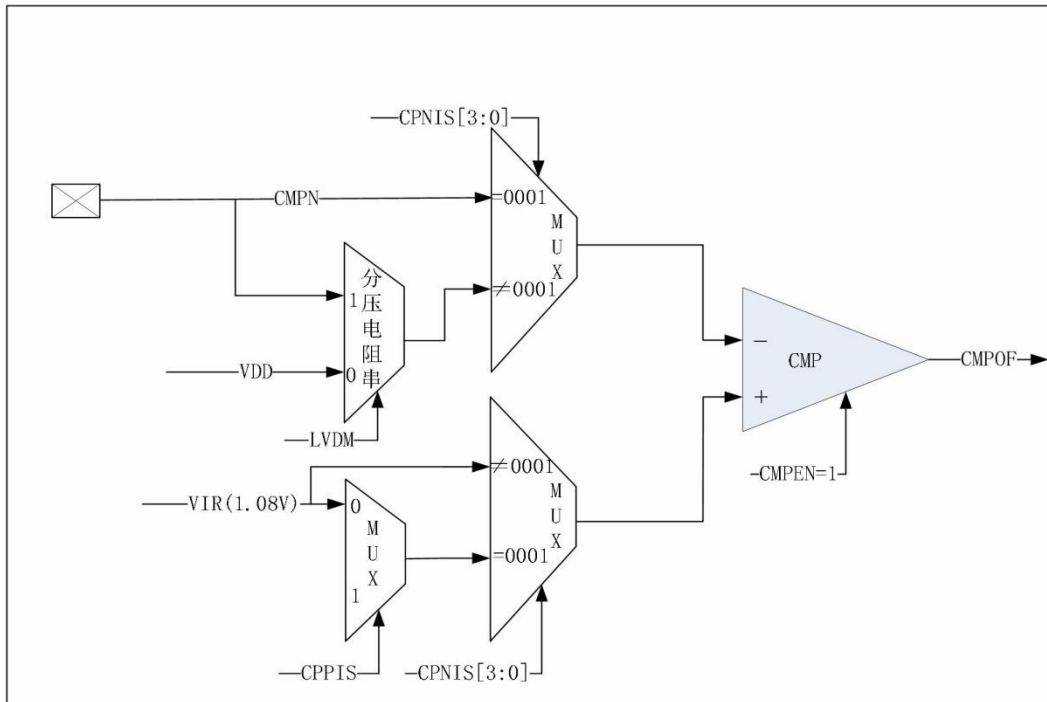
➤ 例：利用 Timer1 定时器产生 PWM0/1/2 周期 1ms，PWM0/1/2 占空比 50%，配置字选择 4MHz 系统时钟，4T 模式

```

...
BCF      AUXR,0          ;PWM0/1/2 输出端口配置位
MOVLW    0XF8
MOVWF    TRISB           ;PORTB0/1/2 输出模式
MOVWF    PORTB           ;PORTB0/1/2 输出低电平
BCF      T0CR,T1IE       ;禁止 T1 中断
BCF      INTECON,GIE     ;禁止中断功能
MOVLW    0XFA
MOVWF    T1
MOVWF    T1LOAD          ;PWM 周期= (T1LOAD+1) *T1 的计数时钟周
MOVLW    0x7D
MOVWF    PWM0P
MOVWF    PWM1P
MOVWF    PWM2P           ;PWM 占空比= (PWM0P/ (T1LOAD+1))
MOVLW    0XC2
MOVWF    T1CON           ;使能 T1 预分频比选择 1:4
MOVLW    0X63
MOVWF    PWMCON         ;允许 PWM0/1/2 输出
    
```

## 9 LVD/CMP

LVD/CMP 工作原理框图:



芯片内置比较器 CMP，包括内部参考电压 VIR（1.08V）电路和 VDD 内部分压电路。可选择端口 CMPN 输入电压的分压信号与 VIR 比较。

通过设置 CMPEN 可以开启或关闭比较器，通过 CPPIS 选择内部参考电压 VIR 作为比较器正输入端的信号源，通过 CPNIS 选择不同的 VDD 分压信号、或 CMPN 输入电压作为比较器负输入端的信号源，通过 CMPOF 读取比较器的输出状态。

开启比较器后选择端口作为输入时，对应端口的数字 I/O 功能将自动关闭。关闭比较器后端口自动恢复数字 I/O 功能。

比较器 CMP 可实现低电压检测功能，即将 VDD 通过电阻产生的多路分压信号与内部参考电压 VIR 进行比较，当选定的分压信号低于 VIR 时，则表示 VDD 电压低于分压信号对应的比较电压，比较器输出将从低电平变为高电平，输出状态位改变并产生 CMP 中断，从而可检测到低电压。

### PCON 寄存器

08h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	WDTEN	EIS	CMPOF	CPNIS3	CPNIS2	CPNIS1	CPNIS0	CMPEN
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
POR 的值	0	0	0	0	0	0	0	0

Bit [5] CMPOF: CMP 输出状态位

0: CMP 负端电压高于正端电压，或 CMP 关闭

1: CMP 负端电压低于正端电压

Bit [4:1] CPNIS[3:0]: CMP 负向端输入选择位（LVD 比较电压选择位）

0000: 1.8V

0001: CMP 端口电压为负向端输入

0010: 2.0V

0011: 2.2V

0100: 2.4V

0101: 2.6V

0110: 2.8V  
 0111: 3.0V  
 1000: 3.2V  
 1001: 3.3V  
 1010: 3.6V  
 1011: 3.7V  
 1100: 3.8V  
 1101: 4.0V  
 1110: 4.1V  
 1111: 4.2V

Bit [0] CMPEN: 比较器 CMP 使能位

1: 开启比较器  
 0: 关闭比较器

#### CMP 控制寄存器

51h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMPCR	GP	GP	GP	LVDM	CMPWK	CMPIE	CMPIES	CPPIS
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 [7:5] GP: 通用功能寄存器位

Bit [4] LVDM: LVD 模式检测(当 CPNIS[3:0] 不等于 0001 时有效)

0: 检测VDD  
 1: 检测端口 (CMPN)

Bit [3] CMPWK: CMP 唤醒使能位

0: 屏蔽CMP 唤醒  
 1: 使能CMP 唤醒

Bit [2] CMPIE: CMP 中断使能位

0: 屏蔽CMP 中断  
 1: 使能CMP 中断

Bit [1] CPIES: CMP 中断触发方式选择位

0: CMP 输出下降沿触发  
 1: CMP 输出上升沿触发

Bit [0] CPPIS: 比较器正相输入端信号选择位 (当 CPNIS[3:0] 等于 0001 时有效)

0: 选择内部参考电压 1.08v  
 1: 未使用

#### 注:

- 1、LVD 使能之后, 有一定的响应时间 (大约 50us 左右), 且 CMPN 脚默认为模拟口。
- 2、多个比较电压连续检测时, 由于比较输出状态位(CMPOF)时只读位, 只能由比较结果置 1/0, 无法软件清除。当 CMPOF=1 时, 默认为芯片处于迟滞响应, 致使当前比较电压检测偏差超过 0.1V。  
 解决方法:  
 1、比较电压检测结束后将 **CMPEN 置 0** 以清除 CMPOF, 待下一个挡位检测设置完成后 **CMPEN 置 1**。  
 2、在多个比较电压检测的应用中, 如果检测比较电压较高的几个电压点, 且应用中电压不会突降到最低比较电压, 需在每个比较电压检测之前添加最低比较电压检测使得负端电压高于正端电压以清除 CMPOF, 消除迟滞。

## 10 充电控制

充电模块采用恒流模式，最大充电电流支持 400mA，涓流充电 40mA。

53h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CHARGE0	S7	S6	S5	S4	S3	S2	S1	S0
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] S7~S0: 充电电流开关控制

1 = 充电电流开关打开，每个开关提供 40mA 充电电流。

0 = 充电电流开关关闭。

54h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CHARGE1	CHARGEN	TEMPEN					S9	S8
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] CHARGEN: 充电使能

1 = 充电使能。

0 = 充电禁止。

Bit [6] TEMPEN: 温度检测使能

1 = 温度检测使能。

0 = 温度检测禁止。

Bit [1:0] S9~S8: 充电电流开关控制

1 = 充电电流开关打开，每个开关提供 40mA 充电电流。

0 = 充电电流开关关闭。



## 11 大电流口以及内置限流电阻控制

HC18P018A0 内置三路灌大电流口，分别配置在 PORTB0、PORTB1、PORTB5 端口，每路端口最大能支持 1A 灌电流。使用大灌电流口时必须将该端口 PWM 使能位置位，否则大电流口无法输出大电流。

55h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RAMP	MODSEL	SINK_SEL				SLOP2	SLOP1	SLOP0
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] MODSEL: 限流电阻使能位  
 1 = 使能内置限流电阻。  
 0 = 禁止内置限流电阻。
- Bit [6] SINK\_SEL: 1A 大电流口使能位  
 1 = 1A 灌电流使能。  
 0 = 灌电流能力由芯片 option 确定。
- Bit [2: 0] SLOP2~0: 端口驱动斜率控制  
 000 = 依次打开关闭 1 路 50mA 输出。  
 001 = 依次打开关闭 2 路 50mA 输出。  
 010 = 依次打开关闭 4 路 50mA 输出。  
 011 = 依次打开关闭 5 路 50mA 输出。  
 100 = 依次打开关闭 10 路 50mA 输出。  
 101 = 依次打开关闭 20 路 50mA 输出。  
 其它=依次打开关闭 20 路 50mA 输出。

### 注:

内置限流电阻，仅PORTB4 端口有此功能，可以使用该端口做LED 指示灯驱动端口。  
 使用大电流口时，要将端口 PWM 使能打开，否则端口无法输出大电流

# 12 指令表

Field	指令格式	描述	C	DC	Z	周期
移动	MOVWF F	$F \leftarrow W$	-	-	-	1
	MOVF F,D	$D \leftarrow F$ (D=0 时为 W, D=1 时为 F)	-	-	√	1
	MOVLW k	$W \leftarrow k$	-	-	-	1
算术	ADDWF F,D	$D \leftarrow W+F$ (D=0 时为 W, D=1 时为 F)	√	√	√	1
	ADCWF F,D	$D \leftarrow W+F+C$ (D=0 时为 W, D=1 时为 F)	√	√	√	1
	ADDLW k	$W \leftarrow W+k$	√	√	√	1
	SUBWF F,D	$D \leftarrow F-W$ (D=0 时为 W, D=1 时为 F)	√	√	√	1
	SBCWF F,D	$D \leftarrow F-W/C$ (D=0 时为 W, D=1 时为 F)	√	√	√	1
	SUBLW k	$W \leftarrow k - W$	√	√	√	1
	DAW -	W 寄存器值进行BCD 调整	√	√	-	1
	INCF F,D	$D \leftarrow F+1$ (D=0 时为 W, D=1 时为 F)	-	-	√	1
DECF F,D	$D \leftarrow F-1$ (D=0 时为 W, D=1 时为 F)	-	-	√	1	
逻辑	ANDWF F,D	$D \leftarrow W$ 与 F (D=0 时为 W, D=1 时为 F)	-	-	√	1
	ANDLW k	$W \leftarrow W$ 与 k	-	-	√	1
	IORWF F,D	$D \leftarrow W$ 或 F (D=0 时为 W, D=1 时为 F)	-	-	√	1
	IORLW k	$W \leftarrow W$ 或 k	-	-	√	1
	XORWF F,D	$D \leftarrow W$ 异或 F (D=0 时为 W, D=1 时为 F)	-	-	√	1
	XORLW k	$W \leftarrow W$ 异或 k	-	-	√	1
	COMF F,D	$D \leftarrow F$ 取反 (D=0 时为 W, D=1 时为 F)	-	-	√	1
处理	SWAPF F,D	$D[7:4,3:0] \leftarrow F[3:0,7:4]$ (D=0 时为 W, D=1 时为 F)	-	-	-	1
	RRF F,D	$D \leftarrow F$ 带进位右移 (D=0 时为 W, D=1 时为 F)	√	-	-	1
	RLF F,D	$D \leftarrow F$ 带进位左移 (D=0 时为 W, D=1 时为 F)	√	-	-	1
	CLRW -	$W \leftarrow 0$	-	-	√	1
	CLRF F	$F \leftarrow 0$	-	-	√	1
	CLRWDT -	清零看门狗定时器, 影响 TO, PD 位	-	-	-	1
	BCF F,d	$F[d] \leftarrow 0$ ( $0 \leq d \leq 7$ )	-	-	-	1
	BSF F,d	$F[d] \leftarrow 1$ ( $0 \leq d \leq 7$ )	-	-	-	1
分支	INCFSZ F,D	$D \leftarrow F+1$ (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	$D \leftarrow F-1$ (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 $F[d]=0$ ( $0 \leq d \leq 7$ ) 则跳过下一句	-	-	-	1(2)
	BTFSS F,d	如果 $F[d]=1$ ( $0 \leq d \leq 7$ ) 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	$W \leftarrow k$ , 带参数返回	-	-	-	2

其他	NOP	-	空操作	-	-	-	1
	SLEEP	-	进入待机模式，影响TO，PD 位	-	-	-	1

# 13 电气特性

## 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
流过 VDD 最大电流.....	100mA
流过 GND 最大电流.....	150mA

## 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	—	F <sub>CPU</sub> = 0~8MHz	2.4	—	5.5	V	
			F <sub>CPU</sub> = 0~4MHz	1.8	—	5.5		
			F <sub>CPU</sub> = 0~2MHz	1.5	—	5.5		
			F <sub>CPU</sub> = 0~1MHz	1.5	—	5.5		
			F <sub>CPU</sub> = 0~455KHz	1.5	—	5.5		
			F <sub>CPU</sub> = 0~32KHz	1.5	—	5.5		
I <sub>DD1</sub>	工作电流	3V	F <sub>CPU</sub> = 8MHz, 2T, WDT 禁止, 无负载	—	1.0	—	mA	
		5V		—	2.0	—	mA	
I <sub>DD2</sub>	工作电流	3V	F <sub>CPU</sub> = 4MHz, 2T WDT 禁止, 无负载	—	0.8	—	mA	
		5V		—	1.5	—	mA	
I <sub>DD3</sub>	工作电流	3V	F <sub>CPU</sub> = 4MHz, 4T WDT 禁止, 无负载	—	0.5	—	mA	
		5V		—	0.9	—	mA	
I <sub>DD4</sub>	工作电流	3V	F <sub>CPU</sub> = 32KHz, 4T, WDT 禁止, 无负载	—	7	—	μA	
		5V		—	10	—	μA	
I <sub>DD5</sub>	工作电流	—	VDD<BOR 档位	—	—	1	μA	
I <sub>sb1</sub>	静态电流	3V	休眠模式, WDT 使能, 无负载	—	5	—	μA	
		5V		—	15	—	μA	
I <sub>sb2</sub>	静态电流	3V	休眠模式, WDT 禁止, 无负载	—	1	1.5	μA	
		5V		—	1	1.5	μA	
I <sub>LC</sub>	端口输入漏电流	3V	端口输入模式, V <sub>IN</sub> =VDD 或 GND	-1	0	1	μA	
		5V	端口输入模式, V <sub>IN</sub> =VDD 或 GND	-1	0	1	μA	
V <sub>IL1</sub>	输入低电平	5V	非施密特输入口(不包括的 PORTB3 端口)	VSS	0.25VDD	0.3VDD	V	
		3V		VSS	0.25VDD	0.3VDD	V	
V <sub>IH1</sub>	输入高电平	5V		0.25VDD	0.3VDD	VDD	V	
		3V		0.25VDD	0.3VDD	VDD	V	
V <sub>IL2</sub>	输入低电平	5V		非施密特输入口, PORTB3 端 口	VSS	0.3VDD	0.5VDD	V
		3V			VSS	0.3VDD	0.5VDD	V
V <sub>IH2</sub>	输入高电平	5V	0.5VDD		0.5VDD	VDD	V	
		3V	0.5VDD		0.5VDD	VDD	V	
I <sub>OL1</sub>	输出灌电流	5V	输出口, V <sub>out</sub> =VSS+0.6V			10		mA
		3V	(不包括PORTB3 口)			-		mA

I <sub>OH1</sub>	输出拉电流	5V	输出口, V <sub>out</sub> =VDD-0.6V		6		mA
		3V			-		mA
I <sub>OL2</sub>	输出灌电流	5V	输出口, V <sub>out</sub> =VSS+0.6V (不包括 PORTB3 口)		30		mA
I <sub>OH2</sub>	输出拉电流	5V	输出口, V <sub>out</sub> =VDD-0.6V		18		mA
I <sub>OL5</sub>	输出灌电流	5V	PORTB3 输出 口,		40		mA
R <sub>PH1</sub>	内部上拉电阻	5V	可编程上拉电阻 (不包括PORTB3)	-	75	-	kΩ
R <sub>PH2</sub>	内部上拉电阻	5V	可编程上拉电阻 (不包括PORTB3)	-	25	-	kΩ
R <sub>PH3</sub>	PORTB3 上拉 电阻 1	5V	可编程上拉电阻	-	200	-	kΩ
	PORTB3 上拉 电阻 2	5V	可编程上拉电阻	-	50	-	kΩ
R <sub>PD</sub>	内部下拉电阻	5V	可编程下拉电阻 (包括PORTB3)	-	125	-	kΩ
R <sub>PD</sub>	内部下拉电阻	5V	内部下拉电阻 (PORTB3)		250		KΩ
V <sub>BOR</sub>	低电压复位	—	-	-10%	所选 BOR	+10%	V
LVD	低电压检测	—	-	-10%	所选LVD	+10%	V
I <sub>LVD</sub>	LVD 功耗				2		uA
V <sub>POR</sub>	上电复位电压	—	-	-10%	1.2	+10%	V

注意：如无另外说明，以上数据测试条件均为VDD=5V、常温 25°C。

➤ AC 特性：

参数	符号	条件	最小值	典型值	最大值	单位
内部 RC16M 启动时间	T <sub>set1</sub>	常温, VDD=5V	-	-	5	μs
内部 RC32K 启动时间	T <sub>set2</sub>	常温, VDD=5V	-	-	150	μs
内部高频RC 频率精度	F <sub>IRC1</sub>	VDD=1.8V~5.5V, 25°C	16(1-1%)	16	16(1+1%)	MHz
内部高频RC 频率精度	F <sub>IRC2</sub>	VDD=5.0V,-40°C~+85°C	16(1-2.5%)	16	16(1+2.5%)	MHz
内部低频RC 频率精度	F <sub>WRC1</sub>	VDD=1.8V~5.5V, 25°C	32(1-1%)	32	32(1-1%)	KHz
内部低频RC 频率精度	F <sub>WRC2</sub>	VDD=5.0V,-40°C~+85°C	32(1-2.5%)	32	32(1-2.5%)	KHz
复位脉冲时间	T <sub>MCLRb</sub>	常温, VDD=5V	200	-	-	μs

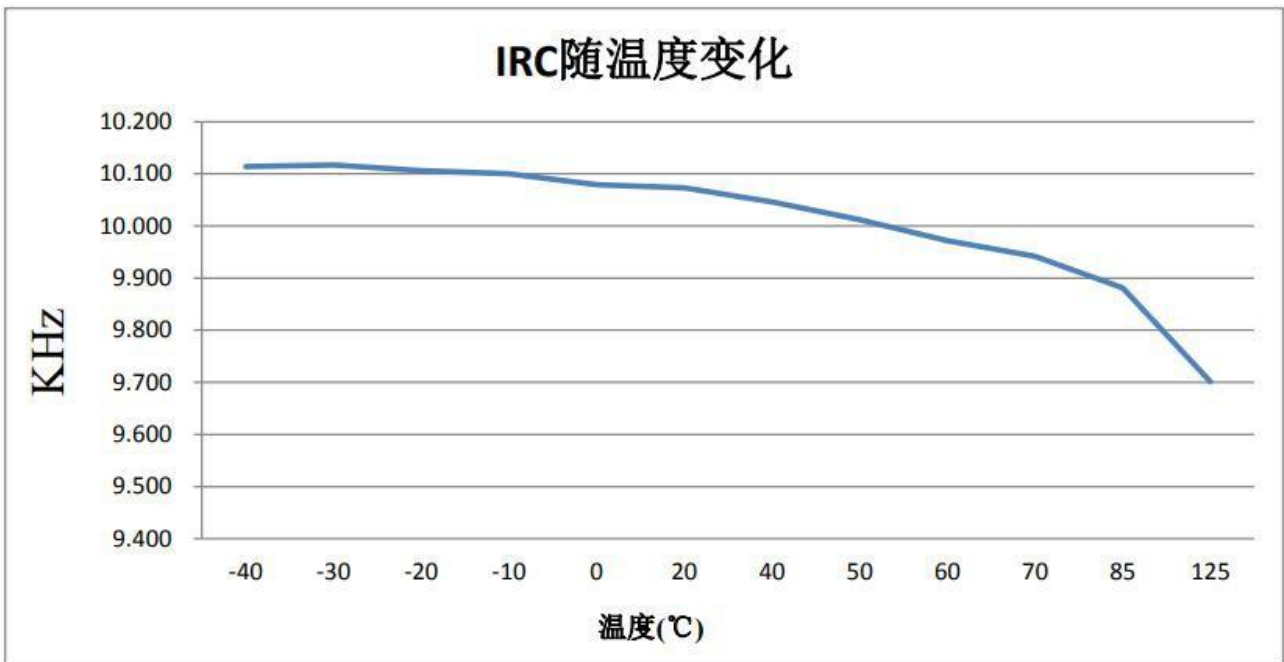
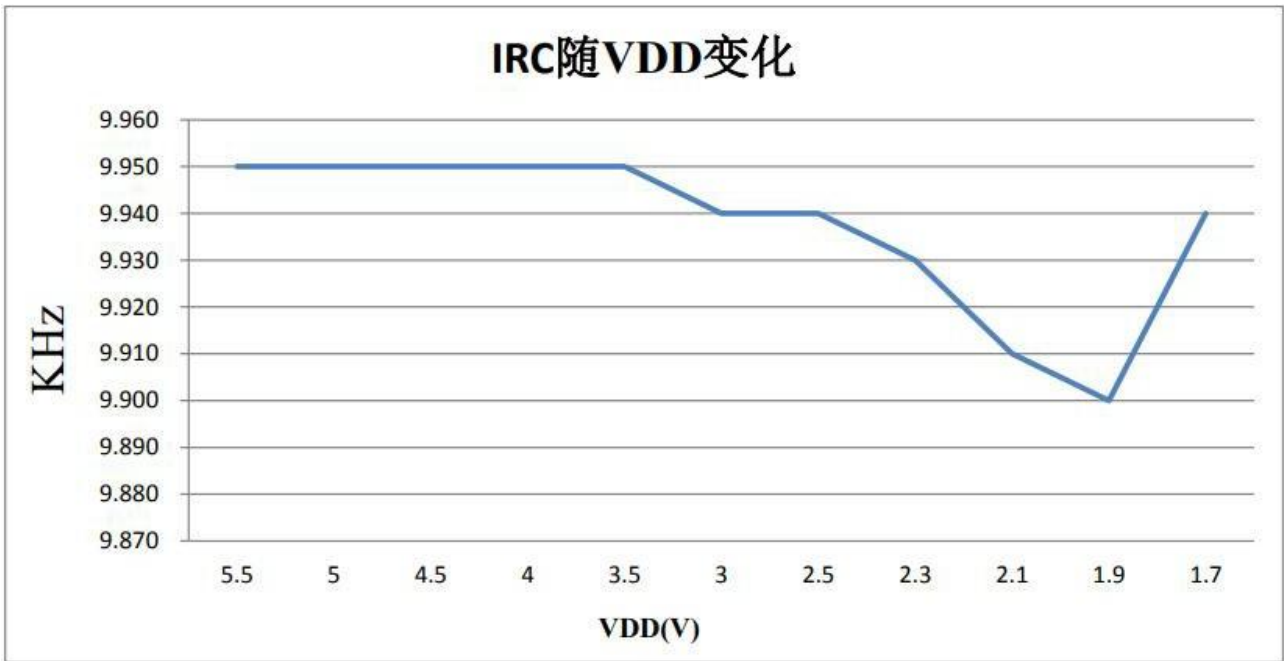
➤ 其他特性：

- 1、ESD (HBM): CLASS 3A (≥4000V)
- 2、ESD (MM) : CLASS 2 (≥200V)
- 3、Latch\_up: CLASS I

CMP 特性参数

VDD=5V, T=25°C

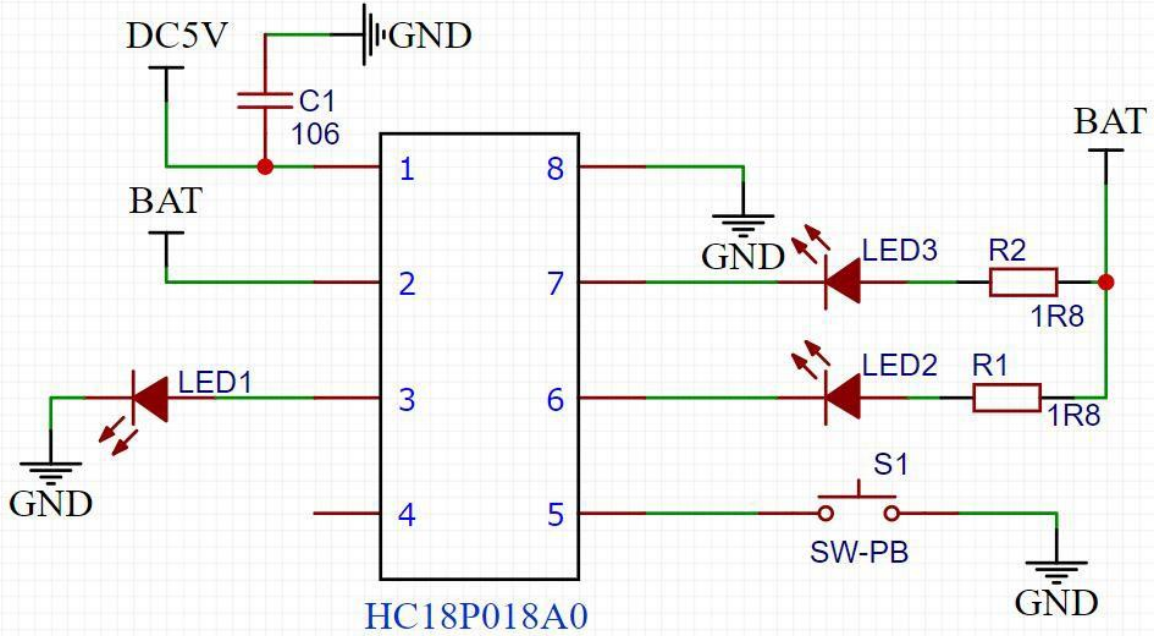
参数	符号	条件	最小值	典型值	最大值	单位
CMP 有效工作电压	V <sub>CMP</sub>	T=-40°C~85°C	2.0	-	5.5	V
工作电流	I <sub>CMP</sub>	T=-40°C~85°C	-	10		μA
输入失调电压	V <sub>offset</sub>	T=-40°C~85°C	-15	-	15	mV
输入共模电压	V <sub>com</sub>	T=-40°C~85°C	0	-	VDD-1.4	V
输出回滞电压	V <sub>hys</sub>	T=-40°C~85°C	5	12	30	mV
响应时间	T <sub>RESP</sub>	T=-40°C~85°C		50		uS
内部参考电压	V <sub>IR</sub>	T=-40°C~85°C	-15%	1.08	+15%	mV



## 14 典型应用原理图

4、6、7 脚为大电流口，可直接驱动电机、发光二极管等大功率器件。

如图：



注：

C1 电容大小根据实际应用调整。

## 15 开发工具

### 15.1 OTP 烧录器 (HC-PM18 5.0)

- 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。

### 15.2 HC-IDE

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

- HC-IDE: V3.0.5.0(支持汇编/C 语言)
- HC-PM18: V5.0.5.0

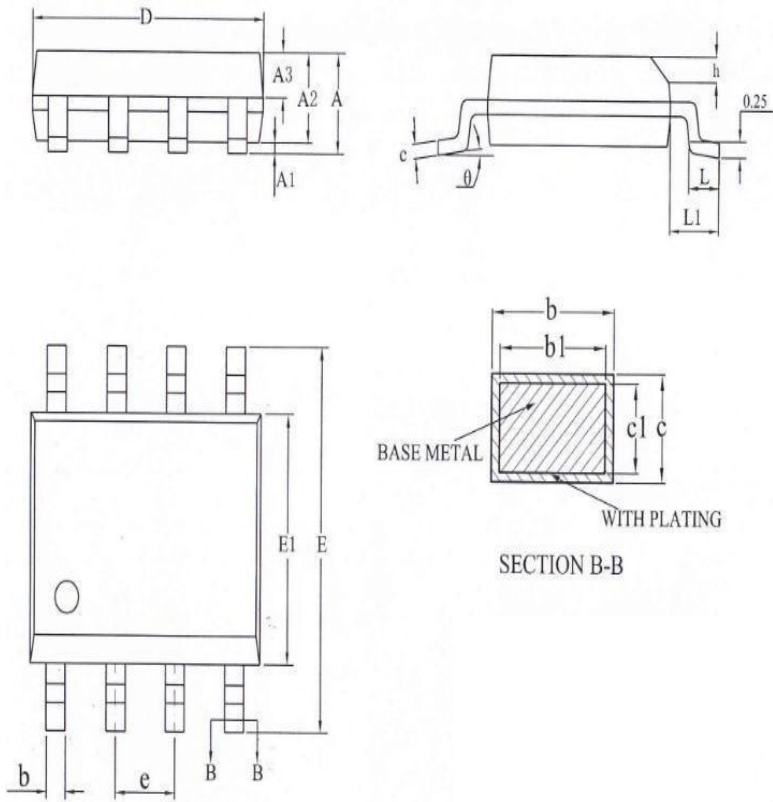
注:

- 1、 详情请参考 HC-IDE 用户手册。
- 2、 IDE 更新请关注芯圣官网: <http://www.holychip.cn/>



# 16 封装信息

## 16.1 SOP8

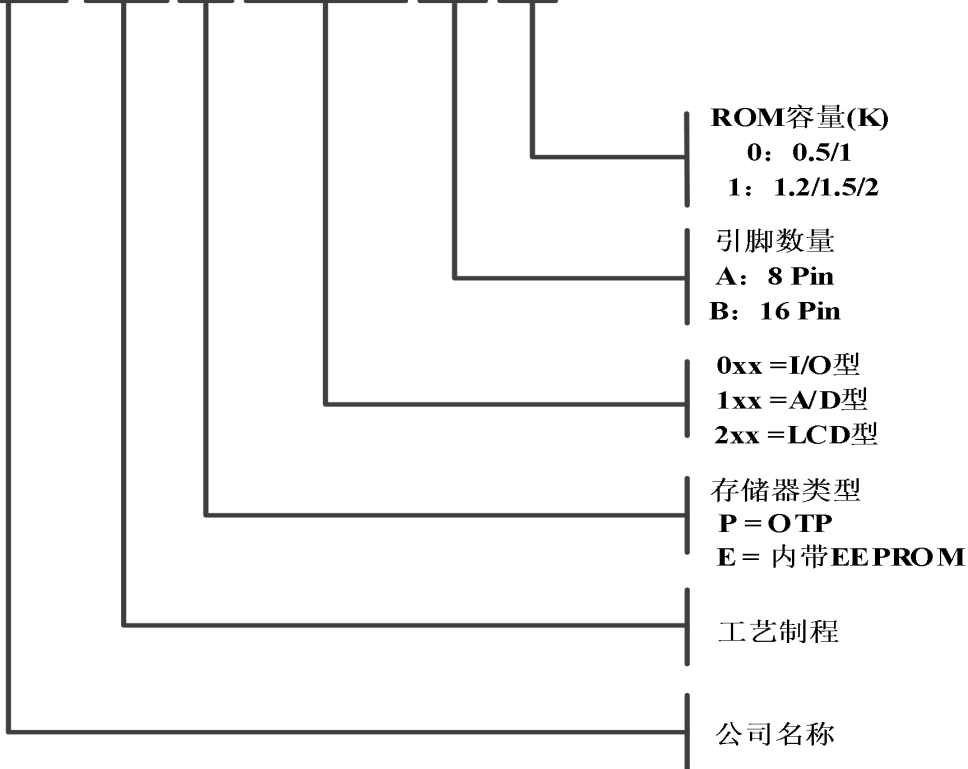


SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.10	—	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	—	0.47
b1	0.38	0.41	0.44
c	0.20	—	0.24
c1	0.19	0.20	0.21
D	4.80	4.90	5.00
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	1.27BSC		
h	0.25	—	0.50
L	0.50	—	0.80
L1	1.05REF		
θ	0	—	8°

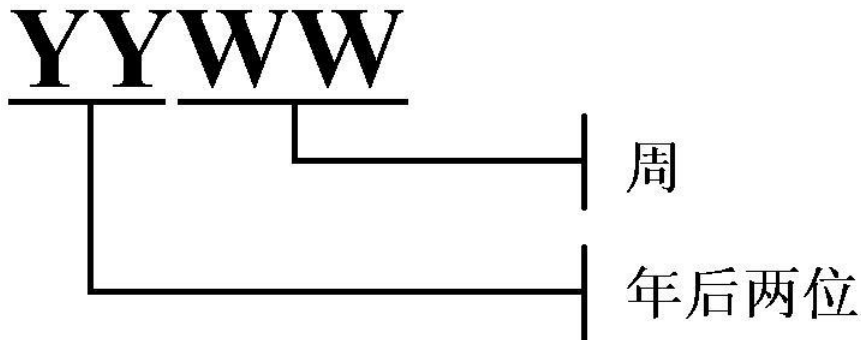
## 17 芯片正印命名规则

### 17.1 芯片型号说明（第一行）

**HC18P018A0**



### 17.2 日期码规则（第二行）



### 17.3 生产批号（第三行）

例: FA126026A

## 18 数据手册版本修正记录

版本	日期	描述
Ver1.00	2021-01-18	初版
Ver1.01	2021-03-15	补充电气特性曲线图
Ver1.02	2022-5-27	1、修改芯片配置字与烧录器 OPTION 对应 2、增加时钟框图描述
Ver1.03	2022-6-27	1、修改芯片配置字与烧录器OPTION 对应 2、删除E-MCU 相关内容
Ver1.04	2022-08-11	1、修改芯片配置字与烧录器OPTION 对应
Ver1.05	2022-9-16	1、删除引脚描述中的CMPP 相关内容 2、删除LVD 中的CMPP 相关内容
Ver1.06	2022-11-16	1、删除封装信息SOP23-6 相关内容
Ver1.07	2023-06-16	1. 修改中断使能位CHARGEN相关描述

HOLYCHIP 公司保留对以下所有产品在可靠性、功能和设计方面的改进作进一步说明的权利。HOLYCHIP 不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，HOLYCHIP 的产品不是专门设计来应用于外科植入、生命维持和任何 HOLYCHIP 产品产生的故障会对个体造成伤害甚至死亡的领域。如果将 HOLYCHIP 的产品用于上述领域，即使这些是由 HOLYCHIP 在产品设计和制造上的疏忽引起的，用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接所产生的律师费用，并且用户保证 HOLYCHIP 及其雇员、子公司、分支机构和销售商与上述事宜无关。

**芯圣电子**

2020 年 01 月