

HC18M121E2

数据手册

14引脚8位
ADC型MTP单片机

目录

目录.....	2
1产品简介	5
1.1 功能特性	5
1.2 引脚图	7
1.3 引脚描述	7
2中央处理器（CPU）	9
2.1 存储器	9
2.1.1 程序存储器（MTPROM）	9
2.1.2 通用数据存储器（RAM）	16
2.1.3 特殊功能寄存器（SFR）	17
2.1.4 芯片配置选择	20
2.2 寻址模式	21
2.2.1 立即寻址	21
2.2.2 直接寻址	21
2.2.3 间接寻址	21
2.2.4 堆栈	22
3系统时钟	23
3.1 概述	23
3.2 系统高频时钟	24
3.2.1 内部高频 RC	24
3.2.2 外部高频晶体振荡器	24
3.3 系统低频时钟	25
3.3.1 内部低频 RC 振荡器	25
3.3.2 外部低频晶体振荡器	25
4复位	26
4.1 概述	26
4.2 上电复位	27
4.3 WDT 复位	28
4.4 欠压复位	28
4.4.1 欠压复位的产生	28
4.4.2 工作死区	29
4.4.3 工作死区与工作频率的关系	29
4.4.4 死区防护	29
5系统工作模式	30
5.1 概述	30
5.2 高频模式	31
5.3 低频模式	31
5.4 绿色模式	31

5.5 休眠模式	32
5.6 不同时钟源下模式选择	32
5.7 唤醒时间	33
6 中断	34
6.1 概述	34
6.2 中断请求和标志寄存器	34
6.3 GIE 全局中断	36
6.4 中断保护	36
6.5 INT0/1 中断	36
6.6 TIMER0 中断	37
6.7 端口电平变化中断	37
6.8 TIMER1/2 中断	37
6.9 ADC 中断	38
7 I/O 端口	39
7.1 I/O 端口模式	39
7.2 I/O 上拉模式	39
7.3 I/O 下拉模式	40
7.4 I/O 端口数据寄存器	40
8 定时器	41
8.1 看门狗定时器	41
8.2 TIMER0 定时/计数器	42
8.3 TIMER1 定时/计数器	44
8.3.1 功能概述	44
8.3.2 T1 使用操作说明	44
8.3.3 T1 相关寄存器	45
8.4 TIMER2 定时/计数器	47
8.4.1 功能概述	47
8.4.2 T2 使用操作说明	47
8.4.3 T2 相关寄存器	48
8.5 TIMER1 和 TIMER2 互补带死区的 PWM 软件实现方式	51
9 模数转换 (ADC)	52
9.1 A/D 引脚控制寄存器	52
9.2 A/D 控制寄存器	53
9.3 ADC 使用	55
10 LVD	57
11 EEPROM	59
11.1 控制操作	59
11.1.1 复位	60
11.1.2 起始/停止命令	60
11.1.3 时钟及数据传输	60

11.1.4 控制字节	61
11.1.5 应答	61
11.1.6 等待模式	61
11.2 写操作	62
11.2.1 字节写	62
11.2.2 页写	62
11.2.3 应答轮询	63
11.3 读操作	64
11.3.1 当前地址读	64
11.3.2 自由地址读	64
11.3.3 连续地址读	65
12 指令表	66
13 电气特性	67
14 开发工具	69
14.1 MTP 烧录器（HC-PM18 PRO）	69
14.2 HC-IDE	69
15 封装信息	70
16 数据手册版本修正记录	71

1 产品简介

HC18M121E2 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 2K×14 位多次可编程 ROM(MTP-ROM)，128×8 位的数据寄存器（RAM），支持 2Kbit 的 EEROM，2 组双向 I/O 口，三个 Timer 定时器/计数器，两个 Timer 控制的 PWM 模块。

一个 12 通道的 12 位模数转换器，多个系统时钟，四种系统工作模式以及多个中断源。

这款单片机可以广泛应用于小家电类、数码电量类、电动车码表等产品。

1.1 功能特性

◆ 存储器配置

- 程序存储器（MTP ROM）空间：2K*14位
- 数据存储器（RAM）空间：128*8位
- 串行EEPROM空间：256byte*8（2K）

◆ 强大的指令系统

- 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
- 8级堆栈缓存器

◆ I/O 引脚配置

- 最多14个IO口均具有可编程的上下拉
- 2组双向I/O口：PORTA，PORTB
- 具有唤醒功能的电平变化中断端口：PORTA，PORTB
- 具有唤醒功能的外部中断引脚：PORTB0（INT0），PORTB1（INT1）可设置触发边沿

◆ BOR

- 1.8V/2.0V/2.7V/3.6V

◆ LVD

- 多级电压检测
- 可编程设置检测VDD

◆ 中断

- 定时器中断：Timer0和Timer1，Timer2
- INT0，INT1外部中断
- 端口电平变化中断
- ADC中断

◆ 定时器

- 看门狗计数器（WDT）
- Timer0：带有8位预分频器的8位定时器/计数器
- Timer1：带有预分频器的1个PWM功能的8位定时器
- Timer2：带有预分频器的1个PWM功能的10位定时器

◆ 系统时钟

- 内建高精度16MHz RC时钟
- 内建40KHz低频RC时钟
- 高频晶体振荡器：最高4MHz
- 低频晶体振荡器：32.768KHz

◆ ADC

- 12位转换分辨率
- 最多12个模拟输入通道（11个外部ADC输入，1个内部1/4VDD检测）
- 内部参考电压(VDD、4V、3V、2V、1.3V)和外部参考电压

◆ 工作模式

- 高频模式
- 低频模式

- 休眠模式
- 绿色模式
- ◆ 复位
 - 上电复位
 - BOR欠压复位
 - WDT溢出复位
- ◆ 封装
 - SOP14
- ◆ 最低工作电压
 - 2.0V~5.5V (Fcpu= 4MHz)
 - 2.7V~5.5V (Fcpu= 8MHz)

1.2 引脚图

HC18M121E2_SOP14			
VDD	1	14	GND
OSCI/AN14/PORTB6	2	13	PORTA2/AN2/PCK/V_{REF}/TM1PWM
OSCO/AN15/PORTB7	3	12	PORTA3/AN3
T2CKI/TM2PWM/AN7/PORTA7	4	11	PORTA1/AN1/PGD/TM2PWM
VPP/PORTB5	5	10	PORTA0/AN0/PGC/TM1PWM
TM2PWM/AN11/PORTB3	6	9	PORTB0/AN8/INT0/TM1PWM/T1CKI
T0CKI/TM1PWM/AN10/PORTB2	7	8	PORTB1/AN9/INT1/TM2PWM

1.3 引脚描述

脚位	名称	类型	说明
1	VDD	P	电源
2	PORTB6 OSCI AN14	I/O AN AN	输入/输出口，带可编程上下拉电阻，端口电平变化中断。 高/低频晶体振荡器输入口。 ADC通道14输入口。
3	PORTB7 AN15 OSCO	I/O AN AN	输入/输出口，带可编程上下拉电阻，端口电平变化中断。 ADC通道15输入口。 高/低频晶体振荡器输出口。
4	PORTA7 TM2PWM AN7 T2CKI	I/O O AN I	输入/输出口，带可编程上拉电阻，端口电平变化中断。 Timer2 PWM输出口。 ADC通道7输入口。 T2时钟输入
5	PORTB5 VPP	I P	输入/输出口，带可编程上下拉电阻，端口电平变化中断。 编程高压电源输入。
6	PORTB3 AN11 TM2PWM	I/O AN O	输入/输出口，带可编程上下拉电阻，端口电平变化中断。 ADC通道11输入口。 Timer2 PWM输出口。
7	PORTB2 TM1PWM AN10 T0CKI	I/O O AN I	输入/输出口，带可编程上拉电阻，端口电平变化中断。 Timer1 PWM输出口。 ADC通道10输入口。 T0时钟输入
8	PORTB1 INT1 AN9 TM2PWM	I/O I AN 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1输入口 ADC通道9输入口。 Timer2 PWM输出口。
9	PORTB0 AN8 INT0 T1CKI TM1PWM	I/O AN I O O	输入/输出口，带可编程上拉电阻，端口电平变化中断。 ADC通道8输入口。 外部中断0输入 T1时钟输入。 Timer1 PWM输出口
10	PORTA0	I/O	输入/输出口，带可编程上拉电阻，端口电平变化中断。

	AN0 PGC TM1PWM	AN I/O O	ADC通道0输入口。 编程数据输入/输出口。 Timer1 PWM输出口。
11	PORTA1 PGD AN1 TM2PWM	I/O I/O AN O	输入/输出口，带可编程上拉电阻，端口电平变化中断。 编程数据输入/输出口。 ADC通道1输入口。 Timer2 PWM输出口。
12	PORTA3 AN3	I/O AN	输入/输出口，带可编程上拉电阻，端口电平变化中断。 ADC通道3输入口。
13	PORTA2 AN2 PCK VREF TM1PWM	I/O AN O P O	输入/输出口，带可编程上拉电阻，端口电平变化中断。 ADC通道2输入口。 内部高频RC振荡频率输出口。 外部参考电压输入口。 Timer1 PWM输出口。
14	GND	P	地。
-	PORTA4 (EEPROM_SCL)	I/O	串行时钟输入。同步数据传输，上升沿数据写入，下降沿数据读出。 PORTA4 输出控制高低电平，控制EEPROM_SCL串行时钟。
-	PORTA6 (EEPROM_SDA)	I/O	串行地址和数据的输入/输出。 双向串行数据传输脚，漏极开路，需要开内部上拉电阻。

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

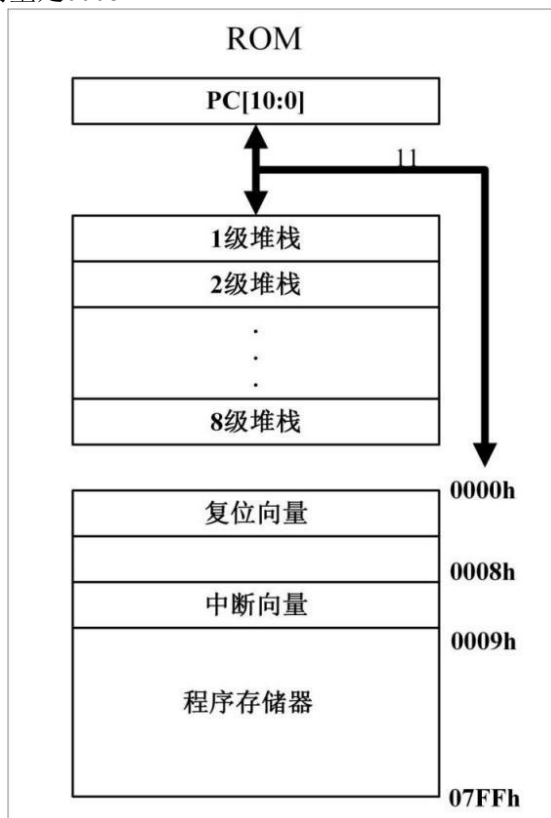
2 中央处理器（CPU）

2.1 存储器

2.1.1 程序存储器（MTP ROM）

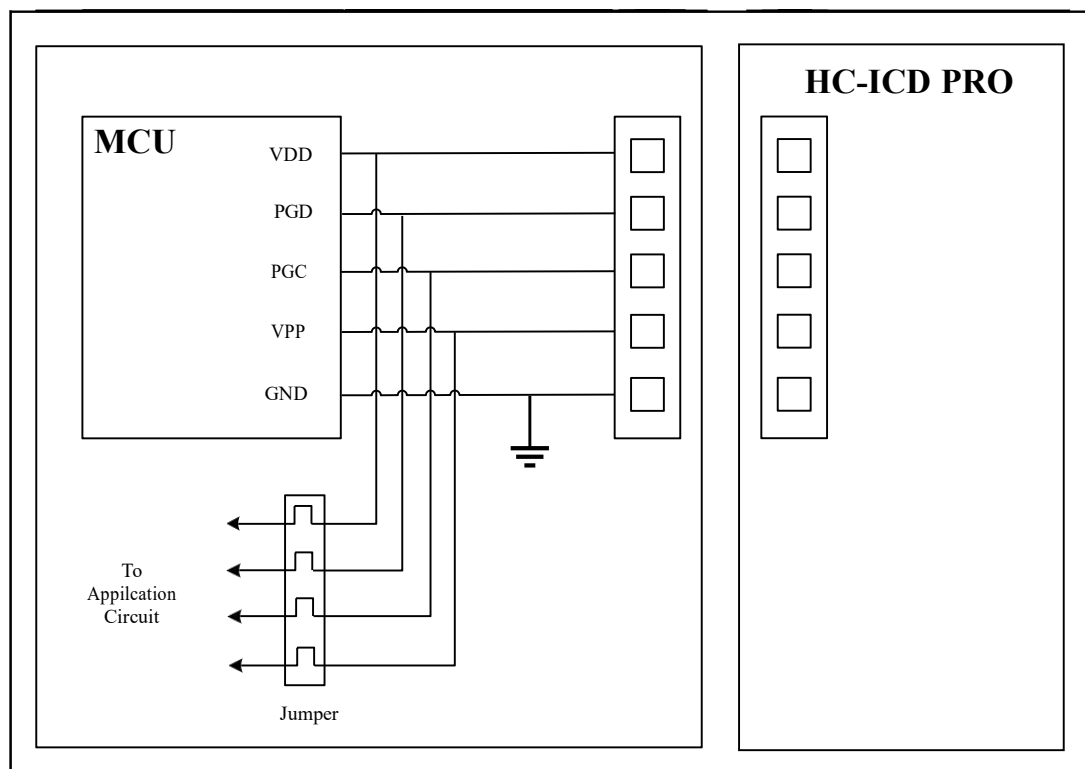
HC18M121E2具有2K×14位的程序存储器，下图给出了程序存储器的映射。访问超出物理地址以外的单元时，会导致返回到地址最低单元。

复位向量是0000H，中断向量是0008H。



2.1.1.1 MTP ICP 操作

用户可以通过 HC-ICD Pro 仿真器的双线方式对 MCU 进行仿真和编程，当 MCU 已经焊在用户板上后，如果用户采用上电复位的方式，只需要连接五根线（VDD、GND、PGC、PGD、VPP），用户系统必须断电，由仿真器提供电源。



注意事项： 仿真时，通讯需要用到 VPP、PGD、PGC 端口，其中 PGD、PGC 两个端口不能当成普通 IO 口使用。

另外 VPP 端口，仿真下载完程序后，此时可以断开 MCU 和仿真器之间的 VPP 连接，断开后 VPP 可用作普通 IO 口。对于工具，为了满足 VPP 仿真时可以作为普通 IO 口，下载完程序后，VPP 需要下电。重新进测试模式时，VDD 需要先下电再重新上电。单步指令和全速运行指令下，timer 可以计数，否则 timer 将会停止计数。

2.1.1.2 复位向量（0000H）

- 上电复位（POR=1，BOR=X，TO=1）
- 低电压复位（POR=1，BOR=1，TO=1）
- 看门狗复位（POR=1，BOR=1，TO=0）

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T1CR_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          400H        ;用户程序起始
MAIN:
...
...
END              ;用户程序结束
    
```

- 例：复位源判断

```

ORG          0000H
GOTO         RST_JUGE

...

RST_JUGE:
    
```

```

BTFSS    T1CR_AUXR,POR
GOTO     ISPOR      ;POR 标志为0, 判定为上电复位
BTFSS    T1CR_AUXR,BOR
GOTO     ISBOR      ;POR=1,BOR=1,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:
...

ISPOR:
BSF      T1CR_AUXR,POR    ;上电复位处理程序

ISBOR:
BSF      T1CR_AUXR,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

```

2.1.1.3 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

```

ORG      0000H
GOTO     MAIN
ORG      0008H
GOTO     INT_SERVICE

MAIN:
...

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
...
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

```

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 0X08，响应中断后，程序自动跳转到 0X08 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，注意恢复顺序，对 W 必须使用 SWAPF。
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. 需要软件清空对应的中断标志。
6. 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。

2.1.1.4 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 3 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
...
TAB1:
    ORG      100H
    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. 数据表宽度为 8 位。
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；故表头尽量放在数据页前端，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。
4. CPU 运行在 8M 下，无法使用 MTP 的查表，4M 及以下才能使用。

➤ 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  ;TABBUF =0,跳转 LABEL0_TAB2
    GOTO     LABEL1_TAB2  ;以下类推
    GOTO     LABEL2_TAB2
    GOTO     LABEL3_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（Bit 8~Bit 15），寄存器 PMADRL 指向 ROM 区数据地址的低字节（Bit 0~Bit 7）。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

```

MOVF    TABLE_ADDR_H, W
MOVWF   PMADRH           ;设置TABLE地址高字节
MOVF    TABLE_ADDR_L, W
MOVWF   PMADRL           ;设置TABLE地址低字节
BSF     PMCON, RDON      ;开始读
NOP
NOP                       ;等待两条指令
MOVF    PMDATL, W
MOVWF   TABLE_DATA_L   ;TABLE_DATA_L= TABLE地址数据低字节
MOVF    PMDATH, W
MOVWF   TABLE_DATA_H   ;TABLE_DATA_H= TABLE地址数据高字节
...
...
TABLE   DW      1234H      ;定义数据表（16位）数据。
        DW      F178H
        DW      2123H

```

2.1.1.4.1 读 ROM 使能寄存器 PMCON

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-	-	-	-	-	-	-	RDON

位编号	位符号	说明
7-1	-	保留位
0	RDON	读控制位 0: 不启动 ROM 存储器读操作 1: 启动 ROM 读操作 (由硬件清零 RDON; 软件只能将 RDON 位置 1, 但不能清零)

2.1.1.4.2 读 ROM 数据寄存器 PMDATL、PMDATH

PMDATL

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	PMDATL							

位编号	位符号	说明
7-0	PMDATL	读 ROM 数据寄存器低 8 位

PMDATH

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	PMDATH							

位编号	位符号	说明
7-0	PMDATH	读 ROM 数据寄存器高 8 位

2.1.1.4.3 读 ROM 地址寄存器 PMADRL、PMADRH

PMADRL

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	PMADRL							

位编号	位符号	说明
7-0	PMADRL	读 ROM 地址寄存器低 8 位

PMADRH

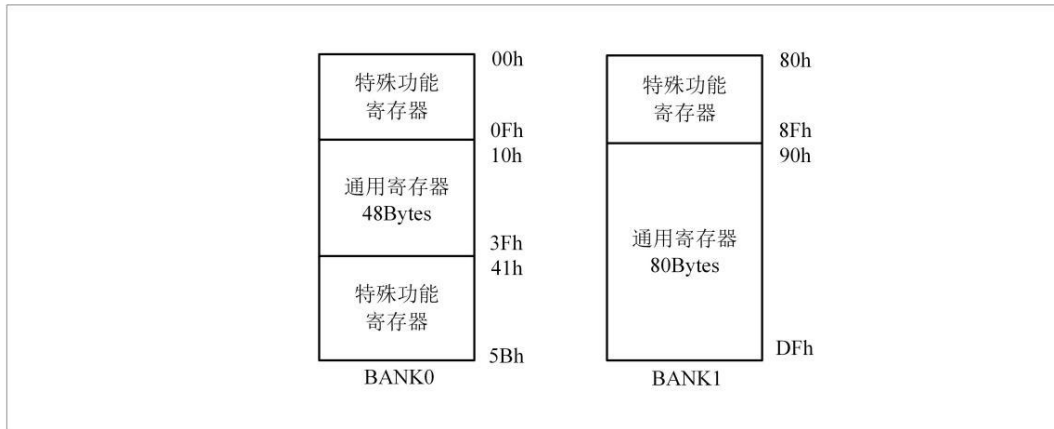
位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	PMADRH							

位编号	位符号	说明
7-0	PMADRH	读 ROM 地址寄存器高 8 位

2.1.2 通用数据存储器（RAM）

共有 128 个通用寄存器（GPR），分在 Bank0/Bank1 存储区。

数据存储器映射：



00h&80h	INDF	间接寻址寄存器（不是实际存在的物理寄存器）							
02h&82h	PCL	程序计数器（PC）低字节							
03h&83h	STATUS	RST	-	RP0	TO	PD	Z	DC	C
04h&84h	FSR	数据指针寄存器							
0Ah&8Ah	PCLATH						程序计数器高 3 位缓存器		
0Eh&8Eh	INTECON	GIE	-	INT0F	PBIF	T0IF	INT0E	PBIE	T0IE

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	-	RP0	TO	PD	Z	DC	C
04h	FSR	数据指针寄存器							
05h	PORTA	PORTA7	PORTA6		PORTA4	PORTA3	PORTA2	PORTA1	PORTA0
06h	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
07h	IOCA	IOCA7	-		-	IOCA3	IOCA2	IOCA1	IOCA0
08h	LVDSCR	LVDF	-	-	LVDSEL3	LVDSEL2	LVDSEL1	LVDSEL0	LVDEN
09h	IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0
0Ah	PCLATH	-	-	-	-	-	程序计数器高 3 位寄存器		
0Bh	PDCON	PDB7	PDB6	PDB5	PDB4	PDB3	PDB2	PDB1	PDB0
0Ch	-	-	-	-	-	-	-	-	-
0Dh	PHCON	PHB7	PHB6	PHB5	PHB4	PHB3	PHB2	PHB1	PHB0
0Eh	INTECON	GIE	-	INT0F	PBIF	T0IF	INT0E	PBIE	T0IE
0Fh	PIR	-	INT1ED G	INT1E	INT1F	ADIE	ADIF	PAIE	PAIF
40h	PMCON	-	-	-	-	-	-	MTP_CLEN	RD_FLAG
41h	OPTION	WDTEN	INT0ED G	T0CS	T0SE	PSA	PS2	PS1	PS0
42h	PDCON1	PDA7	PDA6		PDA4	PDA3	PDA2	PDA1	PDA0
43h	PHCON1	PHA7	PHA6		PHA4	PHA3	PHA2	PHA1	PHA0
45h	TRISA	TRISA7	TRISA6		TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
46h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
47h	PMADRL	-	-	-	-	-	-	-	-
48h	PMADRH	-	-	-	-	-	-	-	-
49h	PMDATL	-	-	-	-	-	-	-	-
4Ah	PMDATH	-	-	-	-	-	-	-	-
4Bh	T1CR_AUXR	T0CK	T2IE	T2IF	POR	BOR	BOREN	T1IE	T1IF
4Ch	T1CON	T1EN	PWM0E	T0OSCEN	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	T2CON	T2EN	PWM1E	SCS	T2CK1	T2CK0	T2PR2	T2PR1	T2PR0
51h	TMR2H	PWMS1	PWMS0	TIMER2_9	TIMER2_8	T2LOAD_9	T2LOAD_8	PWM1_9	PWM1_8
52h	T2	Timer2 [7:0]计数寄存器							
53h	T2LOAD	Timer2[7:0]重载寄存器							
54h	PWM1P	PWM1[7:0]占空比控制寄存器							
55h	PWMSET	-	-	-	-	-	-	-	PWMM
56h	ANSELL	ANSEL7	ANSEL6	-	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0
57h	ANSELH	ANSEL15	ANSEL14	-	-	ANSEL11	ANSEL10	ANSEL9	ANSEL8
58h	ADRESL	ADC 结果寄存器低字节							

59h	ADRESH	ADC 结果寄存器高字节							
5Ah	ADCON0			CHS3	CHS2	CHS1	CHS0	ADON	ADEN
5Bh	ADCON1	-	ADCS2	ADCS1	ADCS0	VHS2	VHS0	VHS0	ADREF

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	数据指针寄存器							
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

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	0	0	0	0	0	0	0

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 11 位）→ PC

子程序返回指令 RETRUN/RETLW/RETFIE：TOS（堆栈栈顶）→ PC

ADDWF PCL, F

PCLATH[2:0], ALU[7:0]（ALU 运算结果）→ PC

其它 PCL 作为目的操作数指令

PCLATH[2:0], 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	-	RP0	TO	PD	Z	DC	C
R/W	R/W	-	R/W	R	R	R/W	R/W	R/W
POR 的值	0	-	0	1	1	x	x	x

Bit [7] RST: 唤醒源标志

1 = 芯片通过 PORTx 变化唤醒(复位/SLEEP 指令)

0 = 芯片通过其它复位唤醒

Bit [5] RP0: BANK 切换位

1 = 切换到 BANK1

0 = 切换到 BANK0

Bit [4] TO: 超时位

1 = 上电、执行了 CLRWDWT 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 CLRWDWT 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T1CR_AUXR

4Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CR-UXR	T0CK	T2IE	T2IF	POR	BOR	BOREN	T1IE	T1IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	0	0	0	q	q	1	0	0

注: q = 取值视条件而定

Bit [7] T0CK: T0 时钟选择位

0 = T0 计数时钟由 T0CS 决定

1 = T0 以外部低频振荡器作为计数时钟

Bit [6] T2IE: T2 溢出中断使能位

0 = 不使能 T2 中断

1 = 使能 T2 中断

Bit [5] T2IF: T2 溢出中断标志位

0 = T2 未溢出

1 = T2 溢出

- Bit [4] POR: 上电复位状态位
0 = 非上电复位
1 = 发生了上电复位 (需要软件置 0)
- Bit [3] BOR: 欠压复位状态位
0 = 未发生欠压复位
1 = 发生了欠压复位 (需要软件置 0)
- Bit [2] BOREN: 欠压复位使能
1 = 使能欠压复位
0 = 禁止欠压复位
- Bit [1] T1IE: T1 溢出中断使能位
0 = 不使能 T1 中断
1 = 使能 T1 中断
- Bit [0] T1IF: T1 溢出中断标志位
0 = T1 未溢出
1 = T1 溢出

2.1.4 芯片配置选择

HC18M121E2 配置表

芯片配置	配置选择	说明
BOR 电压	1.8V	复位电压设置为 1.8V
	2.0V	复位电压设置为 2.0V
	2.7V	复位电压设置为 2.7V
	3.6V	复位电压设置为 3.6V
WDT 溢出时间	TWDT0	TWDT(no Prescaler)=14.4ms
	TWDT1	TWDT(no Prescaler)=3.6ms
	TWDT2	TWDT(no Prescaler)=230.4ms
	TWDT3	TWDT(no Prescaler)=57.6ms
WDTE	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)
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
输入管脚施密特	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
振荡器模式选择	内部低频 RC 振荡器: 40KHz	
	内部高频 RC 振荡器: 16MHz	
	外部低频晶体振荡器: 32KHz	
	外部高频晶体振荡器: 4M	
内部高频 RC 频率选择 16M	非 16M, 由高频内部 RC 频率决定	
	固定选择 16M	
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
	250KHz	内部 RC 振荡器频率为 250KHz
	125KHz	内部 RC 振荡器频率为 125KHz
	62.5KHz	内部 RC 振荡器频率为 62.5KHz

2.2 寻址模式

HC18M121E2 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 0X10~0X3F 通用数据存储器进行清零

BCF RP0 ;BANK0

MOVLW 0X10 ;清零 0X10~0X3F

MOVWF FSR

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 0X3F ;注意这里的边界值为欲操作 RAM 最大地址 + 1

XORWF FSR,W ;利用间接寻址，注意意外指向特殊寄存器的情况

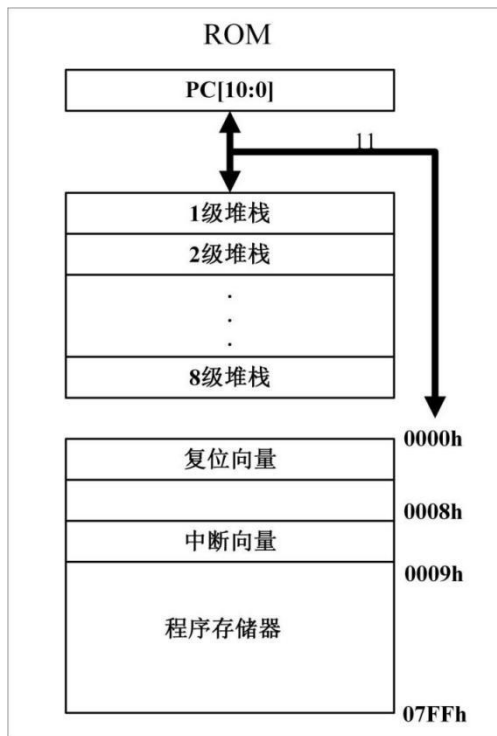
BTFSS Z

GOTO NEXTBYTE ;FSR 的值小于 3Fh,循环清零下一个地址

CONTINUE: ;完成清零操作

2.2.4 堆栈

HC18M121E2 具有一个 8 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。

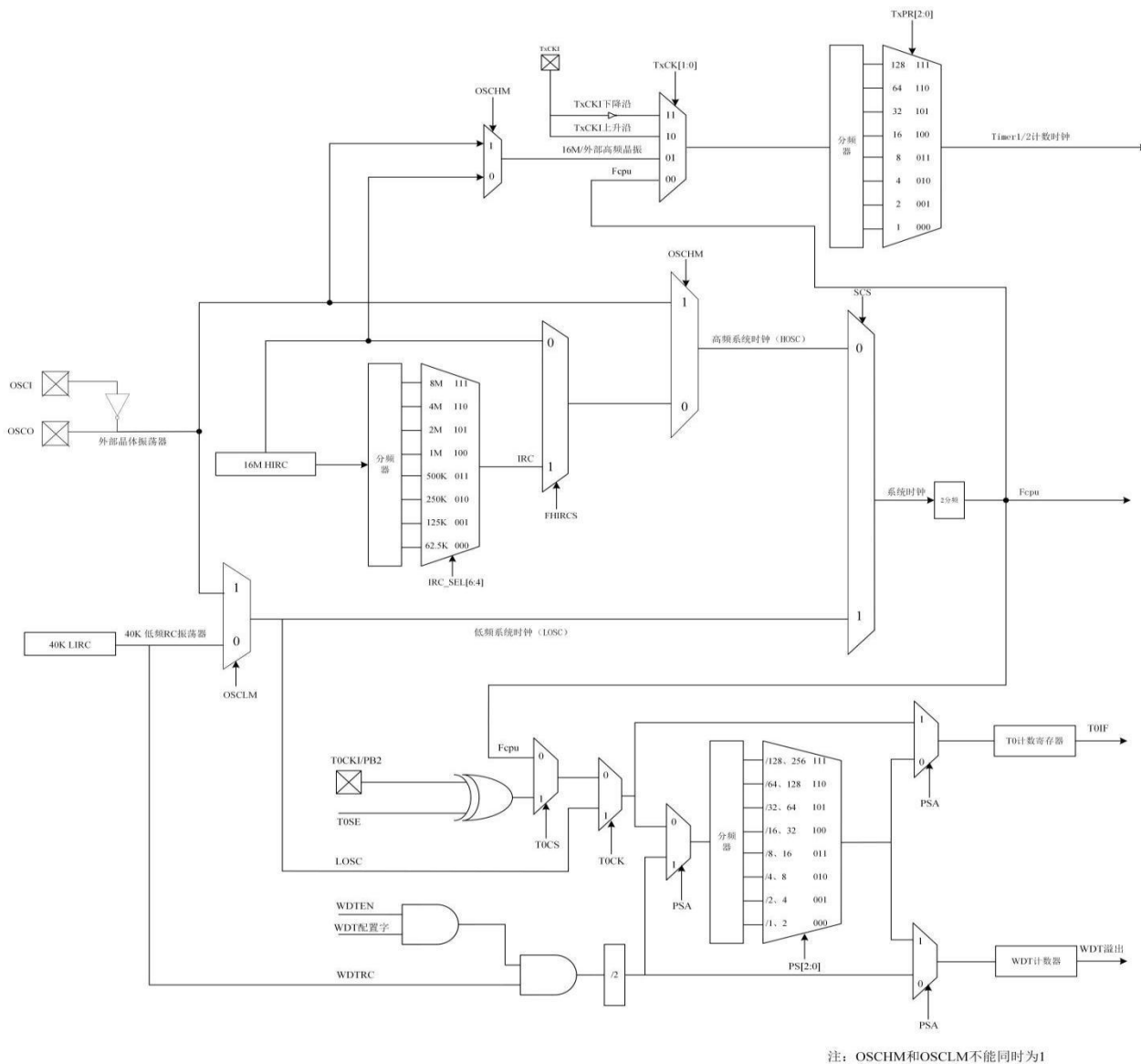


注：

压栈级数请勿超过 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 系统时钟

3.1 概述



HC18M121E2 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（RC 16MHz）提供。低频时钟的时钟源则由内部低频 RC 振荡电路（RC 40KHz@5V）提供。两种时钟都可作为系统时钟源 Fosc。

高频模式：Fcpu = Fsys / 2；

低频模式：Fcpu = Fsys / 2；

也可选用外部晶体振荡器作为系统时钟源外部晶体振荡器共两种模式，由配置字 OSCHM/OSCLM 控制具体模式的选择

高频晶体振荡器：最高 4MHz

低频晶体振荡器：32.768KHz

3.2 系统高频时钟

系统高频时钟为内部高频 RC，或外部高频晶体振荡器最高 4M。

当选用内部高频 16M 时，需在电源两端加一个 100nf 的电容，防止时钟抖动较大，造成误差

3.2.1 内部高频 RC

内置高频 RC 振荡器有 16MHz、8MHz、4MHz、2MHz、1MHz、500KHz、250KHz、125KHz、62.5KHz 九种可选。

高频内部RC振荡器频率选择配置字

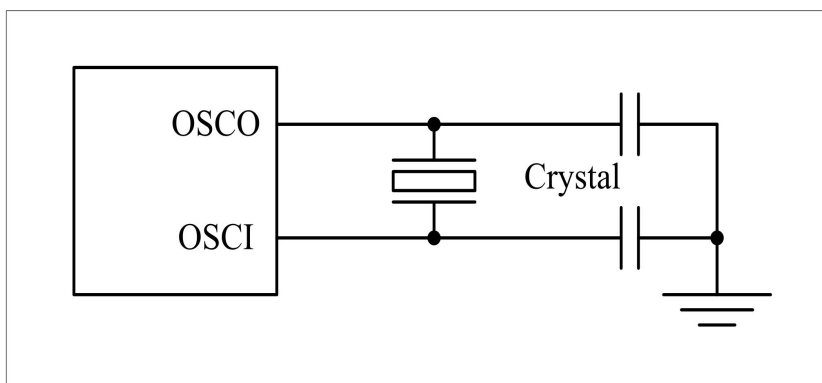
IRC_SEL[6:4]	说明
111	内部RC振荡器频率选择8MHz
110	内部RC振荡器频率选择4MHz
101	内部RC振荡器频率选择2MHz
100	内部RC振荡器频率选择1MHz
011	内部RC振荡器频率选择500KHz
010	内部RC振荡器频率选择250KHz
001	内部RC振荡器频率选择125KHz
000	内部RC振荡器频率选择62.5KHz

16M 时钟单独选择，由配置字的 FHIRCS 决定。

FHIRCS	说明
1	非 16M，由高频内部 RC 频率决定
0	固定选择 16M

3.2.2 外部高频晶体振荡器

可选高频晶体振荡器的频率最高为4MHz，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越近越好。

3.3 系统低频时钟

系统低频时钟为内部低频 RC40KHz，或外部低频晶体振荡器 32.768KHz。

3.3.1 内部低频 RC 振荡器

内部低频 RC 振荡器的频率为 40KHz，除可供 WDT 使用外，也可以提供给系统使用。

3.3.2 外部低频晶体振荡器

低频晶体振荡器的频率为32.768KHz，电容推荐值为20pF。

系统工作在绿色模式下，可以使能低频晶体振荡器。

注：

外部晶体振荡器均使用 OSC0、OSCI 端口。

4 复位

4.1 概述

HC18M121E2 共有三种复位方式：

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

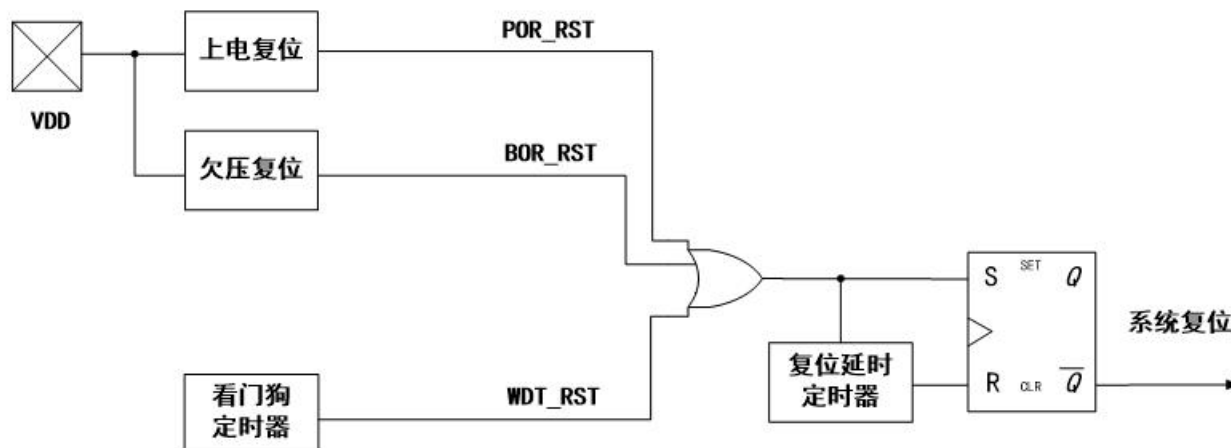
当上述任何一种复位产生时，系统进入复位状态，所有的特殊功能寄存器被初始化，程序停止运行，同时程序计数器（PC）清零。经过上电延时定时器延时后，系统结束复位状态，程序从 000h 地址开始执行。STATUS 寄存器的 Bit4（TO 位）及 T1CR_AUXR 寄存器的 Bit3（BOR 位）、Bit4（POR 位）显示系统复位状态信息，可通过这 3 个标志位判断复位来源，从而控制系统的运行路径。

特殊功能寄存器复位状态：

TO	POR	BOR	复位方式	说明
1	1	x	上电复位	电源上电
u	u	1	欠压复位	电源电压低于 LVR 电压点
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	2ms
欠压复位	2ms
看门狗定时器复位	OPTION 选择

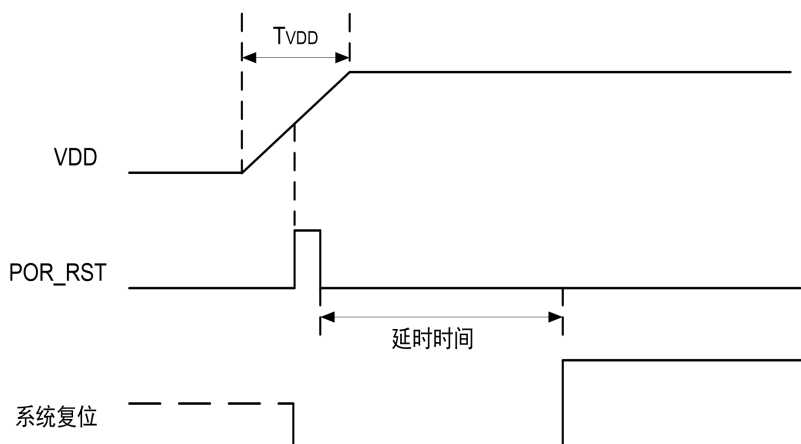
4.2 上电复位

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 Bit 4（TO 位）及 T1CR_AUXR 寄存器的 Bit 3（BOR 位）、Bit 4（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：



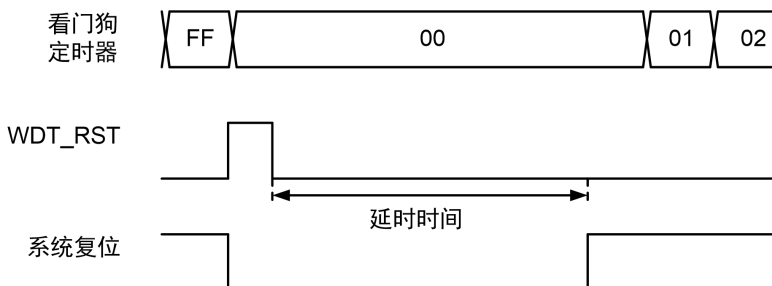
关于上电复位，请注意以下几点：

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。
3. 在应用中，应避免端口有负压产生，否则会导致 POR 复位。

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：

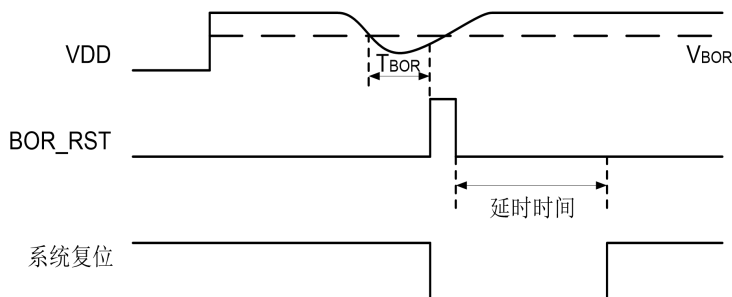
1. 主程序中有一次清看门狗的动作，这种架构能够最大限度的发挥看门狗的保护功能。看门狗的使能逻辑：看门狗使能 = 看门狗配置字使能 & 看门狗软件使能（WDTEN=1）。
2. 不建议在中断程序中对看门狗进行清零，否则无法监控主程序跑飞情况。

4.4 欠压复位

4.4.1 欠压复位的产生

当 VDD 电压下降到 VBOR 以下，且持续时间超过 TBOR 时，系统产生欠压复位。

欠压复位示意图：



注：

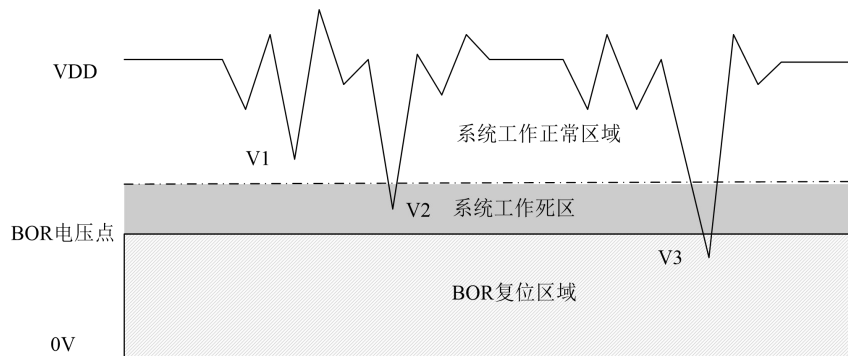
T_{BOR} 需大于 200ns，否则电压跌落时可能不产生欠压复位信号。

特别说明：电压检测电路有一定的迟滞特性，迟滞电压为 5%V 左右。即当 VDD 电压下降到所选

BOR 电压档位时 BOR 复位有效，而 VDD 电压需要上升到 BOR 档位电压 5%V 时 BOR 复位才会解除。

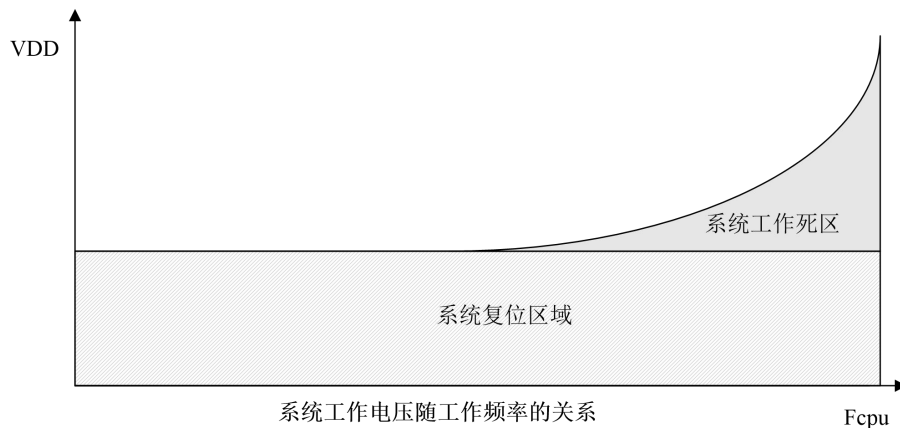
4.4.2 工作死区

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 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

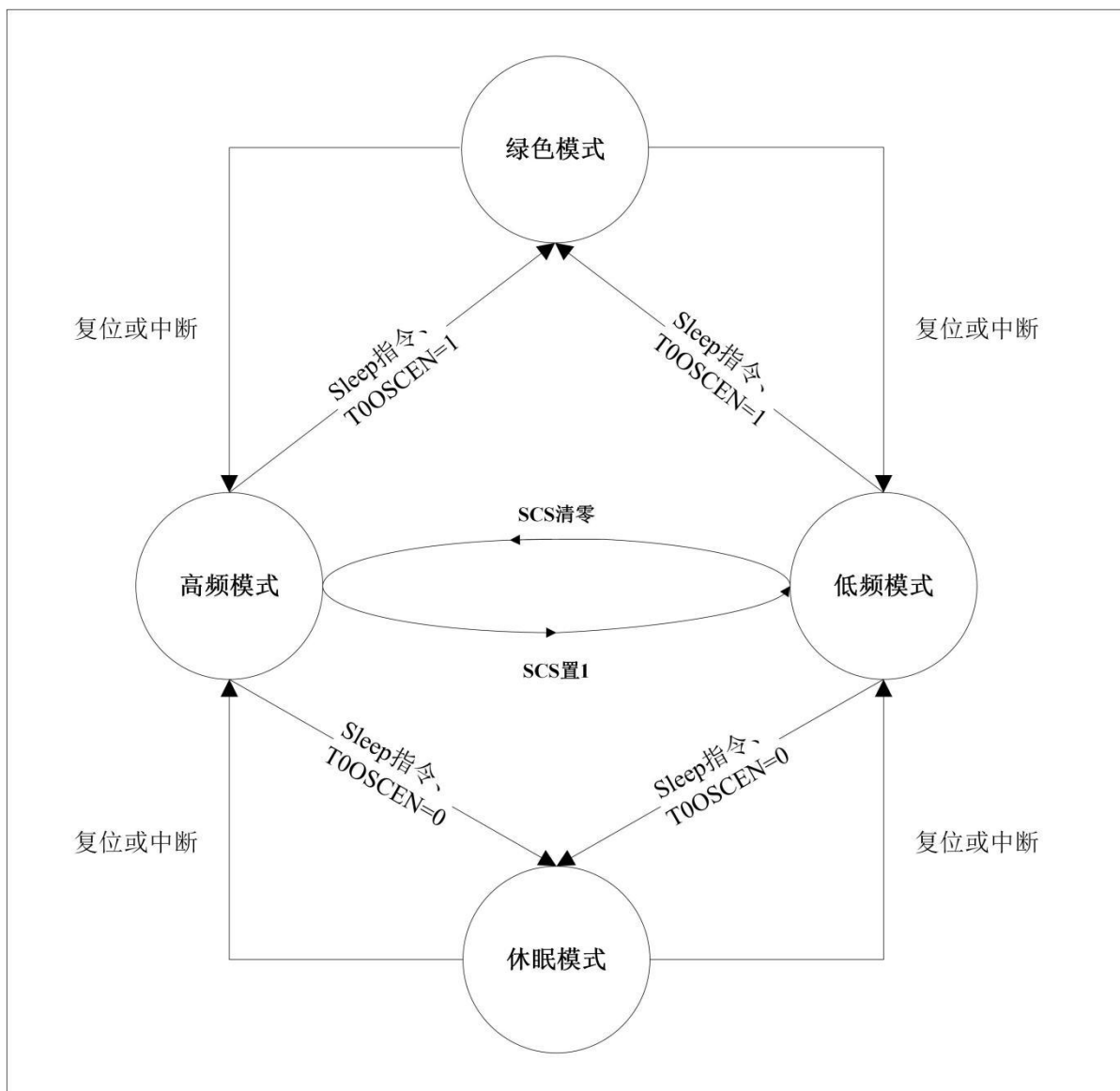
5 系统工作模式

5.1 概述

HC18M121E2 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，由系统配置字决定。程序运行过程中，可以通过设置 SCS 位使系统在高频和低频模式之间切换。



5.2 高频模式

将 SCS 清零，系统时钟切换到高频时钟。
此时系统时钟选择为高频时钟。

5.3 低频模式

将 SCS 置 1，系统时钟切换到低频时钟。
此时系统时钟选择为低频时钟。

5.4 绿色模式

SLEEP 指令可使 MCU 进入绿色模式（T0OSCEN=1），同时对 MCU 会产生以下影响：
系统主时钟的振荡器停止振荡，Timer0 保持工作（此时 Timer0 时钟源为 LOSC）
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 退出绿色模式：
如果使能了 ADC 中断，转换完成时将唤醒 SLEEP。
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
定时器 1 外部计数溢出发生
定时器 2 外部计数溢出发生
任何形式的系统复位发生

绿色模式下，系统除 Timer0 工作，几乎停止了所有的操作。

注：

- 1、进入绿色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，即使系统进入绿色模式，WDT 定时器仍会工作，但在绿色模式下 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。

5.5 休眠模式

SLEEP 指令可使 MCU 进入休眠模式 (T0OSCEN=0)，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：

- 如果使能了 ADC 中断，转换完成时将唤醒 SLEEP。
- 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 外部计数溢出发生
- 定时器 1 外部计数溢出发生
- 定时器 2 外部计数溢出发生
- 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。
- 3、进 SLEEP 模式时，ADC 采集通道请勿选择内部 1/4VDD(AN5)。

5.6 不同时钟源下模式选择

1. 时钟源选择为：内部高频/内部低频；OSCHM:OSCLM=0:0

	休眠模式	绿色模式	低频模式	高频模式
内部高频	关闭	关闭	关闭	打开
内部低频	关闭	打开	打开	打开

2. 时钟源选择为：外部高频/内部低频；OSCHM:OSCLM=1:0

	休眠模式	绿色模式	低频模式	高频模式
外部高频	关闭	关闭	关闭	打开
内部低频	关闭	打开	打开	打开

3. 时钟源选择为：内部高频/外部低频；OSCHM:OSCLM=0:1

	休眠模式	绿色模式	低频模式	高频模式
内部高频	关闭	关闭	关闭	打开
外部低频	关闭	打开	打开	打开

5.7 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
内部高频 RC 振荡器	16 Clock
内部低频 RC 振荡器	16 Clock
高频晶体振荡器	16 Clock
低频晶体振荡器	16 Clock

注：

clock 指分频后的时钟（烧录时 option 中选的时钟）

内部高频唤醒时间为 VBG 模块稳定时间+16 个计数周期（内部高频分频后的系统时钟来计数，唤醒时间最长情况是 VDD=2V 系统时钟选最低频 计算唤醒时间在 600us 内，

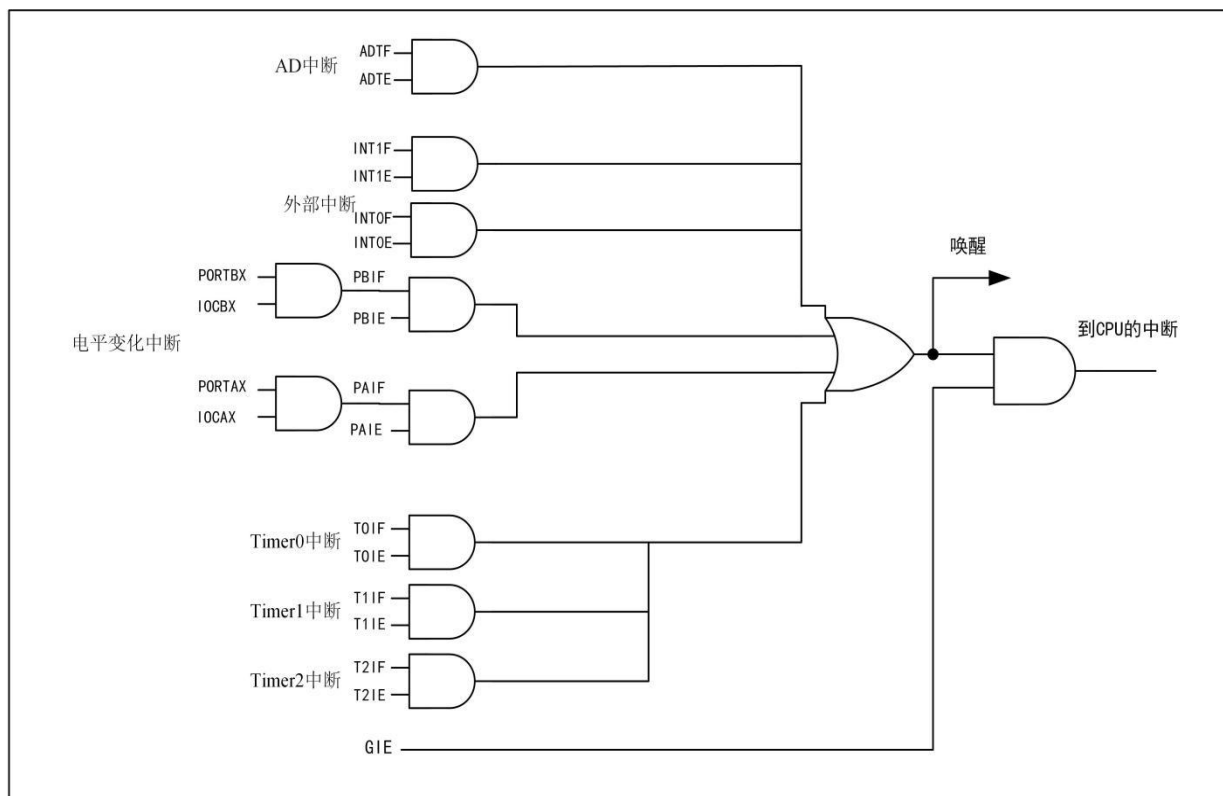
唤醒时间最快情况是 VDD=5 系统时钟选最高频 计算唤醒时间在 35us 内），

内部低频的唤醒时间为 RC40K 的启动时间+16 个低频计数周期。

外部高频唤醒时间为高频晶振起振时间（5ms 内，仿真值）+16 个外部高频计数周期（4us）。

外部低频唤醒时间为低频晶振起振时间（400ms 内，仿真值）+16 个外部低频计数周期（500us）。

6 中断



6.1 概述

HC18M121E2 提供 4 个中断源：Timer0/1/2 定时器中断、INT0/1 外部中断、PORTB/A 端口电平变化中断、AD 中断。

系统从高频或低频模式进入睡眠模式时，AD 中断、INT0/1 外部中断、端口电平变化中断以及 Timer0/1/2 在外部 T0CKI/T1CKI/T2CKI 作为计数时钟时的中断可以将单片机唤醒。

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。

6.2 中断请求和标志寄存器

INTECON 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 INTECON 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 INTECON 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	-	INT0F	PBIF	T0IF	INT0E	PBIE	T0IE
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	0	-	1	0	0	0	0	0

- Bit[7] GIE: 中断总使能
1 = 使能所有中断
0 = 屏蔽所有中断
- Bit [5] INT0F: INT0 中断标志位
1 = INT0 产生外部中断(必须由软件清零)
0 = INT0 未产生外部中断
- Bit [4] 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(必须由软件清零)
0 = PORTB 未产生端口电平变化中断
- Bit [3] T0IF: Timer0 溢出中断使能位
1 = Timer0 产生 Timer0 溢出中断(必须由软件清零)
0 = Timer0 未产生 Timer0 溢出中断
- Bit [2] INT0E: INT0 中断使能位
1 = 使能 INT0 外部中断
0 = 屏蔽 INT0 外部中断
- Bit [1] PBIE: PORTB 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

PIR 中存放 INT1 中断、PORTA 电平变化中断、AD 中断请求标志。一旦有中断请求发生，则 PIR 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 PIR 的状态，程序判断是否有中断发生，并执行相应的中断服务。

PIR 寄存器

0Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR	-	INT1EDG	INT1E	INT1F	ADIE	ADIF	PAIE	PAIF
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	0	0	1	0	0	0	0

- Bit [6] INT1EDG: INT1 中断边沿选择
1 = INT1 上升沿中断
0 = INT1 下降沿中断
- Bit [5] INT1E: INT1 中断使能位
1 = 使能 INT1 外部中断
0 = 屏蔽 INT1 外部中断
- Bit [4] INT1F: INT1 中断标志位
1 = INT1 产生外部中断(必须由软件清零)
0 = INT1 未产生外部中断
- Bit [3] ADIE: ADC 中断使能位
1 = 使能 ADC 中断
0 = 禁止 ADC 中断
- Bit [2] ADIF: AD 中断标志位
1 = AD 转换已完成 (必须由软件清 0)

0=AD 转换未完成或尚未开始

Bit [1] PAIE: PORTA 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] PAIF: PORTA 端口电平变化中断标志位

1 = PORTA 产生端口电平变化中断(必须由软件清零)

0 = PORTA 未产生端口电平变化中断

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

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 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，恢复 W 寄存器、PCLATH 寄存器和 STATUS 寄存器的数值。

6.5 INT0/1 中断

INT0 被触发，则无论 INT0E 处于何种状态，INT0F 都会被置“1”。如果 INT0F=1 且 INT0E=1，系统响应该中断；如果 INT0F=1 而 INT0E=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。

INT1 被触发，则无论 INT1E 处于何种状态，INT1F 都会被置“1”。如果 INT1F=1 且 INT1E=1，系统响应该中断；如果 INT1F=1 而 INT1E=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。

OPTION 寄存器

41h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	-	INT0EDG	T0CS	T0SE	PSA	PS2	PS1	PS0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	-	0	1	1	1	1	1	1

Bit [6] INT0EDG: INT0 中断边沿选择

1 = INT0 上升沿中断

0 = INT0 下降沿中断

PIR 寄存器

0Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR	-	INT1EDG	INT1IE	INT1IF	ADIE	ADIF	PAIE	PAIF
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	0	0	1	0	0	0	0

Bit [6] INT1EDG: INT1 中断边沿选择

1 = INT1 上升沿中断

0 = INT1 下降沿中断

6.6 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”，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。

6.7 端口电平变化中断

PORTB(A)电平变化中断时，则无论 PB(A)IE 处于何种状态，相应 PB(A)IF 都会被置“1”。

如果 PB(A)IF=1 且 PB(A)IE=1，系统响应该中断；如果 PB(A)IF=1 而 PB(A)IE=0，系统并不会执行中断服务。

电平变化中断必须将 PORTB(A)端口设为输入，并将寄存器 IOCB(A)对应位置“1”。

注：PORTB(A)端口变化中断共用中断使能控制信号 PB(A)IE。

IOCA 寄存器

07h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOCA	IOCA7	-	-	-	IOCA3	IOCA2	IOCA1	IOCA0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR 的值	0	0	-	0	0	0	0	0

Bit [7:0] IOCAx: PORTAx 变化中断使能

1 = 使能 PORTAx 端口变化中断/唤醒功能

0 = 屏蔽 PORTAx 端口变化中断/唤醒功能

IOCB 寄存器

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

6.8 Timer1/2 中断

当 T1/2 计数溢出时，Timer1/2 中断被触发，则无论 T1/2IE 处于何种状态，T1/2IF 都会被置“1”。如果 T1/2IF=1 且 T1/2IE=1，系统响应该中断；如果 T1/2IF=1 而 T1/2IE=0，系统并不会执行中断服务。

T1CR_AUXR 寄存器

4Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CR_AUXR	T0CK	T2IE	T2IF	POR	BOR	BOREN	T1IE	T1IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	0	0	0	q	q	1	0	0

Bit [6] T2IE: Timer2 溢出中断使能位

1 = 使能 Timer2 溢出中断

0 = 屏蔽 Timer2 溢出中断

Bit [5] T2IF: Timer2 溢出中断使能位

1 = Timer2 产生 Timer2 溢出中断(必须由软件清零)

0 = Timer2 未产生 Timer2 溢出中断

Bit [1] T1IE: Timer1 溢出中断使能位

1 = 使能 Timer1 溢出中断

0 = 屏蔽 Timer1 溢出中断

Bit [0] T1IF: Timer1 溢出中断使能位

1 = Timer1 产生 Timer1 溢出中断(必须由软件清零)

0 = Timer1 未产生 Timer1 溢出中断

6.9 ADC 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 GIE 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。

PIR 寄存器

0Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIR	-	INT1EDG	INT1E	INT1F	ADIE	ADIF	PAIE	PAIF
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	-	0	0	1	0	0	0	0

Bit [3] ADIE: ADC 中断使能位

1 = 使能 ADC 中断

0 = 禁止 ADC 中断

Bit [2] ADIF: AD 中断标志位

1 = AD 转换已完成 (必须由软件清 0)

0 = AD 转换未完成或尚未开始

7 I/O端口

7.1 I/O 端口模式

PORTA 端口方向寄存器

45h	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

Bit[7:0]:TRISAx

0: TRISAx 对应端口为输出

1: TRISAx 对应端口为输入

注: 有关 PORTA 的相关控制寄存器, Bit5 可写可读但无实际作用。

PA4 连接 E2--SCL 引脚, PA6 连接 E2--SDA 引脚。

PORTB 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	-	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]:TRISBx

0: TRISBx 对应端口为输出

1: TRISBx 对应端口为输入

注: 有关 PORTB 的相关控制寄存器, Bit4 可写可读但无实际作用。

注:

- 1、端口方向寄存器设为输出时, 此时读端口操作作为读数据寄存器的值。
- 2、端口方向寄存器设为输入时, 此时读端口操作作为读端口的输入电平状态。

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	-	PHB3	PHB2	PHB1	PHB0
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] 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

PHCON1 寄存器

43h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PHCON1	PHA7	PHA6	-	PHA4	PHA3	PHA2	PHA1	PHA0
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] PHAx: PORTAx 上拉控制

1 = 屏蔽 PORTAx 输入上拉功能

0 = 使能 PORTAx 输入上拉功能

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	PDB7	PDB6	PDB5	-	PDB3	PDB2	PDB1	PDB0
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] 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

PDCON1 寄存器

42h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDCON1	PDA7	PDA6	-	PDA4	PDA3	PDA2	PDA1	PDA0
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] PDAx: PORTAx 下拉控制

1 = 屏蔽 PORTAx 输入下拉功能

0 = 使能 PORTAx 输入下拉功能

7.4 I/O 端口数据寄存器

PORTA 端口数据寄存器

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

PORTB 端口数据寄存器

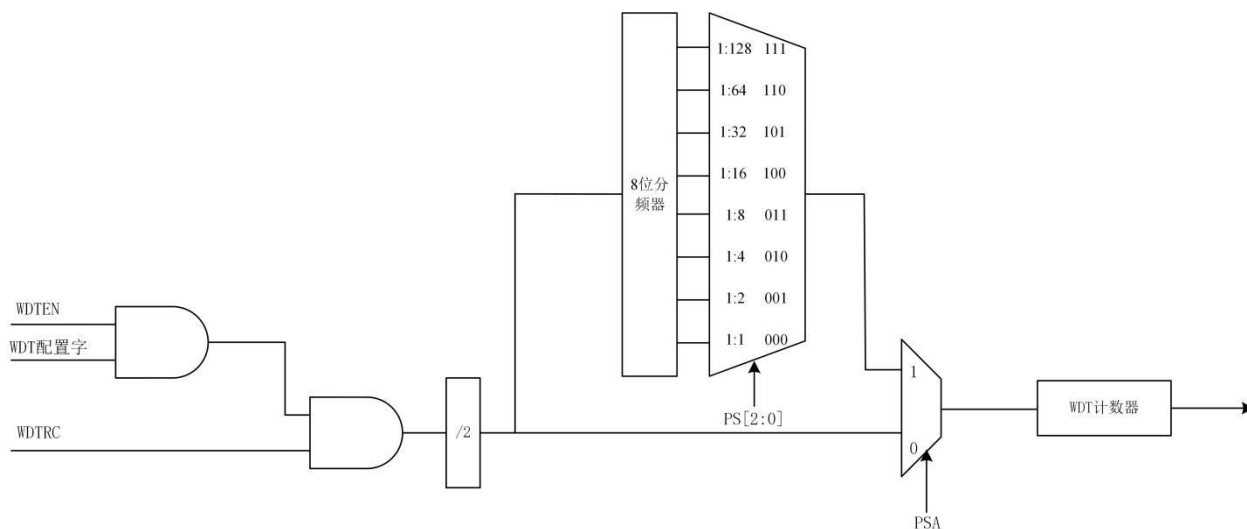
06h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTB	PORTB7	PORTB6	PORTB5	-	PORTB3	PORTB2	PORTB1	PORTB0
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
POR 的值	x	x	x	-	x	x	x	x

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 振荡器是否开启由 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 3.6ms~230.4ms。

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 14.4ms，软件进行 4 分频，则溢出时间为 14.4*4=57.6ms。

OPTION 寄存器

41h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	WDTEN	INT0EDG	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	0	0	1	1	1	1	1	1

Bit 7 WDTE: 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 并使其返回高频或低频模式，程序从 SLEEP 指令下一条开始执行。

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
- 绿色模式定时唤醒

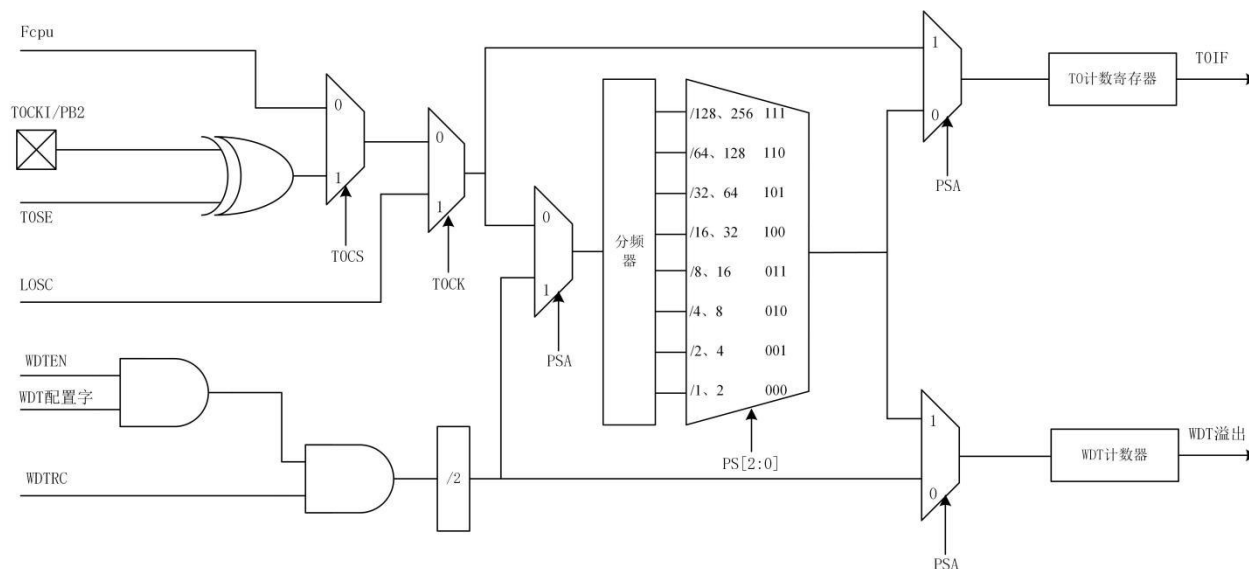
T0 的计数时钟可选择 cpu 时钟 Fcpu、外部管脚 T0CKI、系统低频时钟 LOSC。

预分频器为定时器 T0 与 WDT 定时器共用；

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 的计数周期 = $(256 - T0[7:0]) * (1 / (\text{所选时钟源频率} / \text{分频数}))$

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	WDTEN	INT0EDG	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	0	0	1	1	1	1	1	1

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 外部 T0CKI 计数沿选择

1 = T0CKI 下降沿计数
0 = T0CKI 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，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]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 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

Timer0 工作模式选择

T0SCEN	T0CK	T0CS	T0SE	Timer0 工作状态
x	0	0	x	定时器模式，计数时钟 FCPU，休眠和绿色模式下停止
x	0	1	0	计数器模式，计数时钟 T0CKI，上升沿计数休眠/绿色模式下工作，溢出中断可唤醒 SLEEP
x	0	1	1	计数器模式，计数时钟 T0CKI，下降沿计数休眠/绿色模式下工作，溢出中断可唤醒 SLEEP
1	1	x	x	定时唤醒模式，计数时钟 LOSC 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。

T1CR_AUXR 寄存器

4Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CR_AUXR	T0CK	T2IE	T2IF	POR	BOR	BOREN	T1IE	T1IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	0	0	0	q	q	1	0	0

Bit 7 T0CK：T0 时钟选择

1 = T0 计数时钟为 LOSC

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

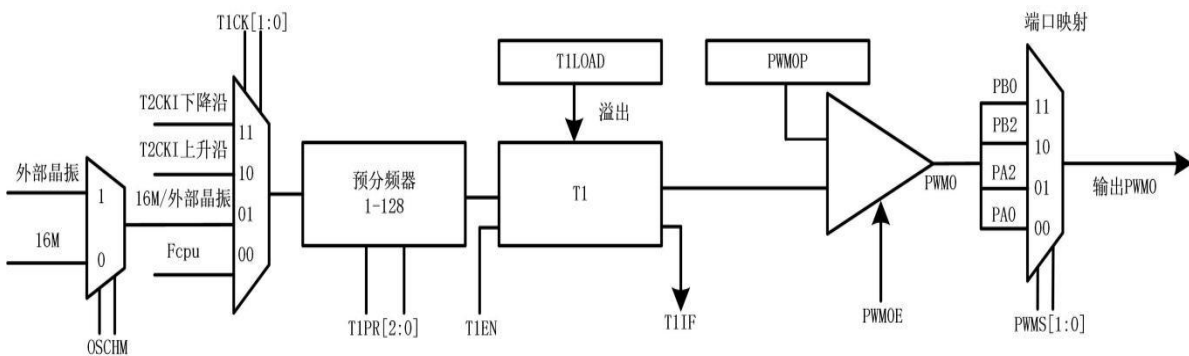
注：关于 Timer0 的时钟源选择，需注意：

- 1、Timer0 时钟源选择为外部时钟源 T0CKI 或外部低频晶振时，具有唤醒功能。
- 2、Timer0 配置为 RTC 模式时，T0 以外部低频振荡器作为计数时钟。

8.3 Timer1 定时/计数器

T1 是一个递减计数器，它的值可以读写，当计数从 00 减到 FF 时，产生 T1 溢出信号，将中断标志位 T1IF 置 1。

Timer1 模块框图



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）
- 溢出中断功能
- 溢出唤醒功能

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，PWMOP 的值写入缓冲器 PWMOP BUFER 用于新的占空比波形生成。

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 PWMOE=1 时，将输出 PWM 波形，当 T1 计数到与 PWMOP 相等时，PWM0 输出置 1；当 T1 计数溢出时，PWM0 输出清 0。

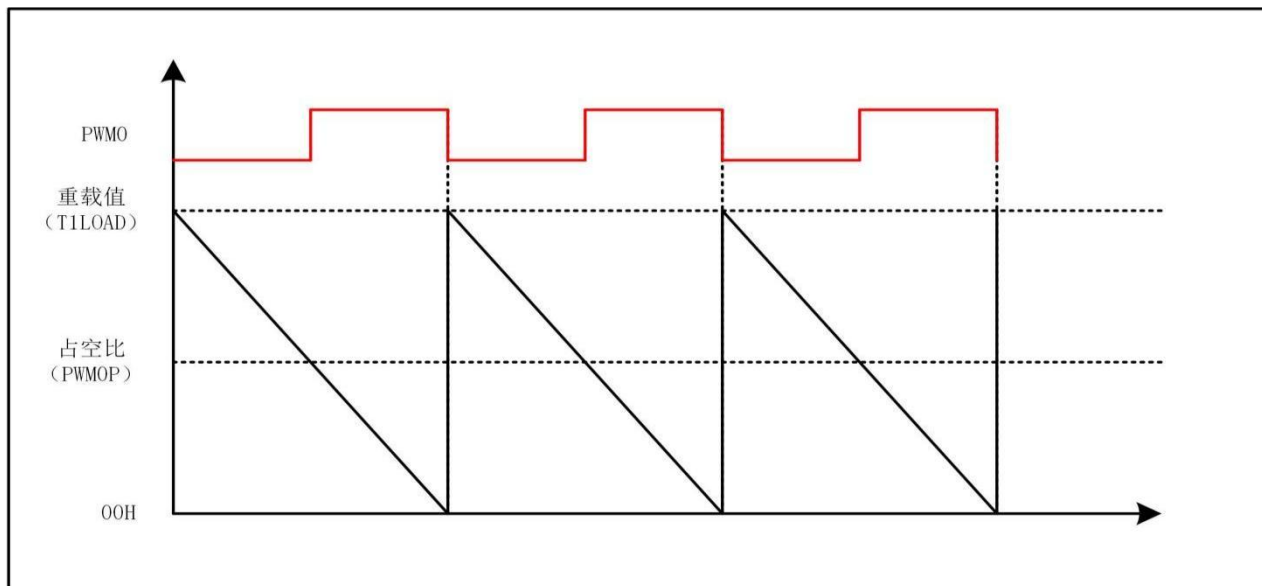
PWM0 占空比的计算如下：

PWM0 高电平时间 = (PWMOP) * 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 = (PWMOP / (T1LOAD+1))

Timer1 的 PWM0 波形如图：



8.3.3 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	T0SCEN	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，对应端口作为 I/O 口

1: 允许 PWM0 输出，对应端口输出 PWM0 信号

注：PB2、PB0、PA2、PA0 都可作为 PWM0 的输出口，它们由 PWMS[1:0] 决定。

Bit [5] T0SCEN 绿色模式或休眠模式选择位（使能后，T0 可工作在定时器模式下）

0: SLEEP 后进入休眠模式

1: SLEEP 后进入绿色模式

Bit [4:3] T1CK[1:0]: T1 时钟源选择

T1CK[1:0]	T1 时钟源
00	Fcpu
01	16M/外部晶振
10	T1CKI 上升沿
11	T1CKI 下降沿

注意：

1、T1CK[1:0]=01B 时，T1 时钟源为内部固定 16M 或外部晶振（最高 4M）

2、系统高频时钟源选择内部高频时，T1 时钟源为固定 16M；选择外部高频晶振时，T1 时钟源为外部高频晶振（最高 4M）。系统工作模式为低频模式时不支持该模式。

例：

当时钟源选择内部高频/内部低频时，高频模式下 T1 时钟源为固定 16M；
当时钟源选择外部高频/内部低频时，高频模式下 T1 时钟源为外部高频晶振（最高 4M）；
当时钟源选择内部高频/外部低频时，高频模式下 T1 时钟源为固定 16M；

3、当 TIMER1 选择 T1CKI 作为计数时钟时，T1CKI 口由硬件设为施密特端口

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

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
TILOAD	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

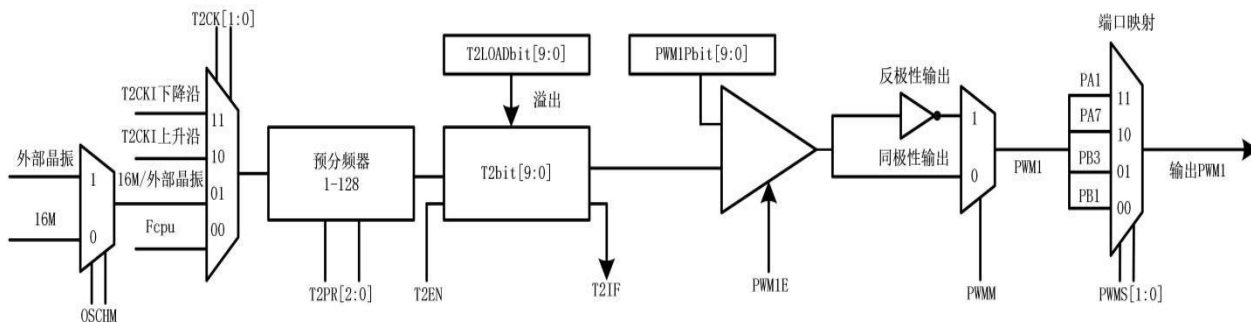
注意：

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

8.4 Timer2 定时/计数器

T2 是一个递减计数器，它的值可以读写，当计数 00 减到 FF 时，产生 T2 溢出信号，将中断标志位 T2IF 置 1。

Timer2 模块框图



8.4.1 功能概述

定时/计数器 T2 10bit 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）
- 可通过纯软件方式实现 Timer1 和 Timer2 互补带死区的 PWM 功能(8bit)。
- 溢出中断功能
- 溢出唤醒功能

8.4.2 T2 使用操作说明

T2CK[1:0] 可选择 T2 的时钟源，T2PR[2:0] 可选择 T2 的预分频比，所选中的时钟源通过预分频器后产生 T2 的时钟。

当 T2 递减到 0 时，此时产生 T2 溢出中断请求标志 T2IF 置 1，重载寄存器值自动置入 T2，PWM1P 的值写入缓冲器 PWM1P BUFER 用于新的占空比波形生成。

通过 T2PR[2:0] 可选择时钟源的分频比，可选择范围为 1~128 分频，对 T2 的写操作将使预分频器清零，分频比保持不变。

PWM1 操作说明：

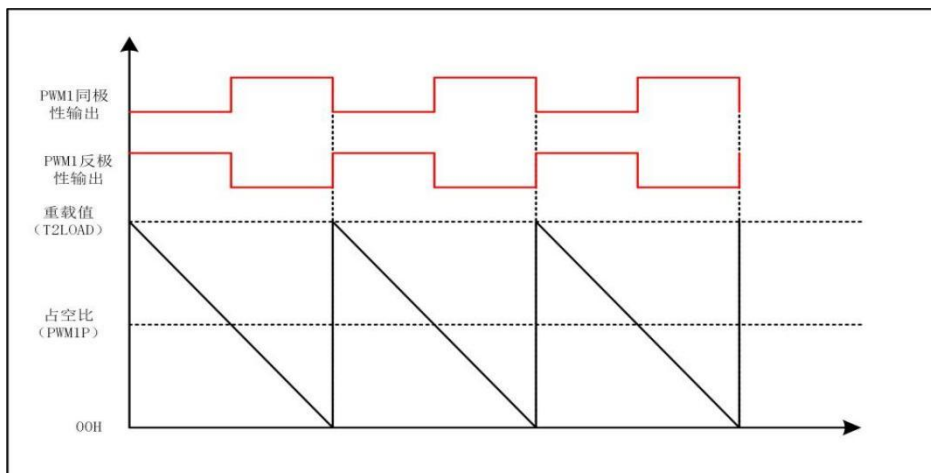
当 PWM1E=1 时，将输出 PWM 波形，当 T2 计数到与 PWM1P 相等时，PWM1 输出置 1；当 T2 计数溢出时，PWM1 输出清 0。PWM1 占空比的计算如下：

PWM1 高电平时间 = (PWM1P) * T2 计数时钟周期

PWM1 周期 (T2 的溢出周期) = (T2LOAD+1) * T2 的计数时钟周期

PWM1 占空比 = (PWM1P / (T2LOAD+1))

Timer2 的 PWM1 波形如下:



8.4.3 T2 相关寄存器

Timer2 控制寄存器

50h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CON	T2EN	PWM1E	SCS	T2CK1	T2CK0	T2PR2	T2PR1	T2PR0
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] T2EN: T2 使能控制

- 0: 关闭 T2
- 1: 启动 T2

Bit [6] PWM1E: PWM1 选择

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

注: PB3、PB1、PA7、PA1 都可作为 PWM1 的输出口, 它们由 PWMS[1:0]决定。

Bit [5] SCS: 高/低频模式选择位

- 0: 系统时钟切换为高频系统时钟
- 1: 系统时钟切换为低频系统时钟

Bit [4:3] T2CK[1:0]: T2 时钟源选择

T2CK[1:0]	T2 时钟源
00	Fcpu
01	16M/外部晶振
10	T2CKI 上升沿
11	T2CKI 下降沿

注意:

- T2CK[1:0]=01B 时, T2 时钟源为内部固定 16M/外部晶振 (最高 4M)
- 高频时钟源选择内部高频时, T2 时钟源为固定 16M; 选择外部高频晶振时, T2 时钟源为外部高频晶振 (最高 4M)。系统工作模式为低频模式时不支持该模式。

例:

当时钟源选择内部高频/内部低频时, 高频模式下 T2 时钟源为固定 16M;
当时钟源选择外部高频/内部低频时, 高频模式下 T2 时钟源为外部高频晶振 (最高 4M);
当时钟源选择内部高频/外部低频时, 高频模式下 T2 时钟源为固定 16M;

- 当 TIMER2 选择 T2CKI 作为计数时钟时, T2CKI 口由硬件设为施密特端口

Bit [2:0] T2PR[2:0]: T2 预分频倍数选择

T2PR[2:0]	Timer2 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

TMR2H 寄存器

51h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMR2H	PWMS1	PWMS0	TIMER2_9	TIMER2_8	T2LOAD_9	T2LOAD_8	PWM1_9	PWM1_8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	0	0	1	1	1	1	0	0

Bit[7:6]: PWMS[1:0]: PWM 管脚映射寄存器

- 00: PB0 作为 TM1PWM 管脚, PB1 作为 TM2PWM
- 01: PB2 作为 TM1PWM 管脚, PB3 作为 TM2PWM
- 10: PA2 作为 TM1PWM 管脚, PA7 作为 TM2PWM
- 11: PA0 作为 TM1PWM 管脚, PA1 作为 TM2PWM

Bit[5:4]: T1MER2[9:8]: Timer2 定时计数寄存器[9:8]

Bit[3:2]: T2LOAD[9:8]: Timer2 重载寄存器[9:8]

Bit[1:0]: PWM1[9:8]: PWM1P 占空比寄存器[9:8]

T2 计数寄存器

52h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2	Timer2 定时计数寄存器[7:0]							
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] Timer2 的值

注: 更新 T2 寄存器时, 应先更新 TMR2H[5:4], 再更新 T2 寄存器

此外, 更新 TMR2H 寄存器时, 请使用整体赋值操作, 不要使用位操作, 否则会出现赋值失败。

T2 重载寄存器

53h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2LOAD	Timer2 重载寄存器[7:0]							
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 Timer2 的重载值

注: 更新 T2LOAD 寄存器时, 应先更新 TMR2H[3:2], 再更新 T2LOAD 寄存器

注意:

当 T2EN=0 时, 写 T2LOAD 将自动加载到 T2 中; 当 T2EN=1 时, 写 T2LOAD 不会加载 T2 中, 在 T2 溢出时自动加载到 T2 中。

PWM1P 占空比寄存器

54h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1P	PWM1P 占空比寄存器							
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

PWMSET 寄存器

55h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMSET	-	-	-	-	-	-	-	PWMM
R/W	-	-	-	-	-	-	-	R/W
POR 的值	-	-	-	-	-	-	-	0

Bit[0] PWMM : PWM1 极性输出选择

- 0: PWM1 同极性输出
- 1: PWM1 反极性输出

注:

在使用 PWM0 和 PWM1 互补带死区输出时, 要把 PWMM 置 1, 使 PWM1 反极性输出;
并且 Timer1 和 Timer2 要选择一样的时钟源, 且周期一致。

由于靠软件实现, 故死区时间主要依赖软件和使用方式, 具体使用方式如下说明:

建议的使用方式:

- 1、T1、T2 都使用 F_{cpu} 且不分频。
- 2、T1、T2 都使用比 F_{cpu} 快的时钟源, 误差小于一个 T_{cpu} (指令周期)。

不建议的使用方式 (误差较大):

- 1、T1、T2 都使用 F_{cpu} 且进行分频。
- 2、T1、T2 都使用 F_{cpu} 慢的时钟源, 越慢误差越大

计算公式及使用方法 (计算时, 进制需保持一致):

T_{death} (预设的死区时长)

$T1LOAD = T2LOAD$ (周期一致)

$PWM0P = (T1LOAD + 1) \times \text{预设的占空比}$

$PWM1P = PWM0P - 2 \times (T_{death} / T_{cpu})$

先使能占空比小的 Timer, 软件延时 $T_{death} - T_{cpu}$ (尽可能用 NOP 指令延时实现) 后再使能另外一个占空比较大一些的 Timer。

8.5 Timer1 和 Timer2 互补带死区的 PWM 软件实现方式

参考例程如下，各个参数均可调整。

设时钟为 2M/2T，即 $F_{cpu}=1M$ ， $T_{cpu}=1\mu s$ 。需要调制一对带死区互补的 PWM 波形，周期 256 μs ，PWM0 占空比 50%，死区时间 $T_{death} = 10\mu s$ 。

```
DELAY:                                //延时时间 5us+2us+2us=Tdeath - Tcpu = 9us
NOP
NOP
NOP
NOP
NOP
RETURN
```

```
CLRF    T1
CLRF    T1LOAD    //初始化
MOVLW   0X40
MOVWF   T1CON      //使能 PWM0 输出//Timer1 分频比 1: 1
```

```
CLRF    TMR2H
CLRF    T2
CLRF    T2LOAD
CLRF    TMR2H      //初始化，高位必须整体赋值，清零
MOVLW   0X40
MOVWF   T2CON      //使能 PWM1 输出//Timer2 分频比 1: 1
```

```
MOVLW   0XFF
MOVWF   T1LOAD      //设PWM0的周期为100us
MOVLW   0X80        //设置PWM0占空比寄存器
MOVWF   PWM0P       //设PWM0
```

```
CLRF    TMR2H      //先设置Timer2高两位配置
MOVLW   0XFF
MOVWF   T2LOAD      //设PWM1的周期为100us
MOVLW   0X6C        //设置PWM1占空比寄存器
MOVWF   PWM1P       //设PWM1
```

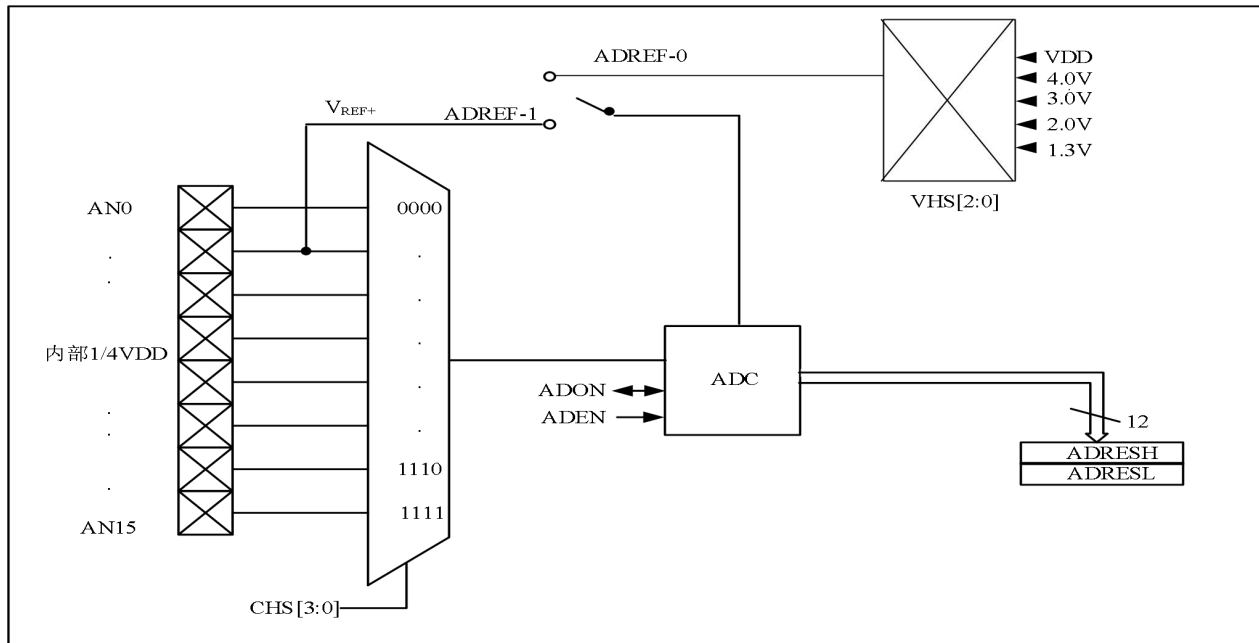
```
BSF     T1CON,T2EN  //开T2,先开占空比小的，再开占空比大的
CALL    DEALY       //延时（错位互补）
BSF     T2CON,T1EN  //开T1
```

配置完成，PWM0/PWM1输出波形

9 模数转换 (ADC)

HC18M121E2具有一个12位转换分辨率的模数转换器，共有11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：



9.1 A/D 引脚控制寄存器

ANSEL[15:0]寄存器

056h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELL	ANSEL7	-	-	-	ANSEL3	ANSEL2	ANSEL1	ANSEL0
R/W	R/W	-	-	R/W	R/W	R/W	R/W	R/W
POR 值	1	-	-	1	1	1	1	1

057h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELH	ANSEL15	ANSEL14	-	-	ANSEL11	ANSEL10	ANSEL9	ANSEL8
R/W	R/W	R/W	-	-	R/W	R/W	R/W	R/W
POR 值	1	1	-	-	1	1	1	1

ANSEL[15:0]: A/D 引脚数模控制位

- 1: 模拟模式，作为模拟信号口，仅可作为 AD 通道的模拟输入。
- 0: 数字模式，作为数字输入或输出口。

注：

ANSEL上电初始值为B ‘1111 1111’，即作为模拟输入。

无论是否应用到AD，均需要在上电后，对IO操作之前按需配置，否则IO口可能无法受控于对应的端口寄存器，状态将不确定。

ANSEL[3:0]对应AN3~AN0 (PA3~PA0)，ANSEL[7]对应AN7 (PA7)

ANSEL[11:8]对应AN11~AN8 (PB3~PB0)，ANSEL[15:14]对应AN15、AN14 (PB7、PB6)

9.2 A/D 控制寄存器

ADCON0寄存器

05Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCON0	-	-	CHS3	CHS2	CHS1	CHS0	ADON	ADEN
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	-	0	0	0	0	0	0

Bit[5:2]: CHS[3:0] AD 通道选择位

Bit[1]: ADON 开始 AD 转换使能位

1=开始一次 AD 转换

0=AD 转换完成后, 硬件自动清零

Bit[0]: ADEN AD 使能位

1=使能 ADC

0=屏蔽 ADC

ADCON1寄存器

05Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCON1	-	ADCS2	ADCS1	ADCS0	VHS2	VHS1	VHS0	ADREF
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	0	0	0	0	0	0	0

Bit[6:4]: ADCS[2:0] ADC 时钟选择位

Bit[3:1]: VHS[2:0] ADC 内部参考电压选择位

Bit[0]: ADREF ADC 外部参考电压选择位

1=外部参考电压

0=内部参考电压

ADC模拟通道选择

CHS [3:0]	模拟通道
0000	AN0
0001	AN1
0010	AN2
0011	AN3
0100	-
0101	内部 1/4VDD(AN5)
0110	-
0111	AN7
1000	AN8
1001	AN9
1010	AN10
1011	AN11
1100	-
1101	-
1110	AN14
1111	AN15

ADC参考电压选择

ADREF	VHS[2:0]	参考电压
0	000	内部VDD

0	001	内部4.0V
0	010	内部3.0V
0	011	内部2.0V
0	1xx	内部1.3V
1	xxx	外部参考电压

ADRESH

059h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADRESH	ADC 结果寄存器高字节							
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

ADRESL

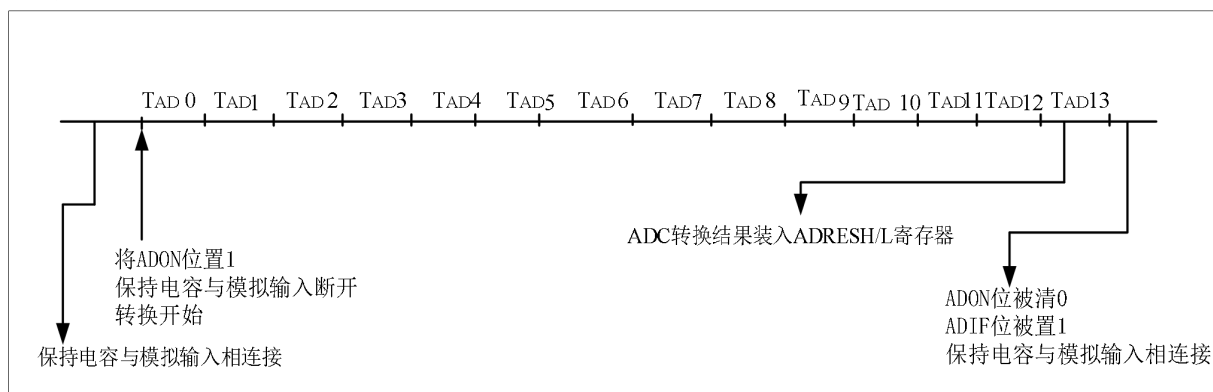
058h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADRESL	ADC 结果寄存器低字节							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	0	0	0	0	0	0	0	0

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 精度为 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。

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)		cpu 频率 (Fcpu) :2MHz
ADC 时钟源	ADCS[2:0]	典型值
Fcpu	000	8us
Fcpu /2	001	16us
Fcpu/4	010	32us
Fcpu /8	011	64us
Fcpu /16	100	128us
Fcpu /32	101	256us
Fcpu /64	110	512us
FRC	111	32us

选择FRC时钟源后，ADC需等待一个指令周期后才能启动转换操作，这使得可以执行SLEEP指令，以降低转换期间的系统噪声。如果使能了ADC中断，转换完成时将唤醒SLEEP。如果禁止了ADC中断，尽管ADEN位仍保持为1，转换完成后ADC模块将关闭。

ADC时钟源不是FRC时，尽管ADEN位仍保持为1，SLEEP指令会导致当前转换中止，ADC模块关闭。

除了选择FRC时钟源，改变系统时钟频率均会改变ADC的时钟频率，从而影响ADC转换时间。

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。

注：

1、ADC的运行时钟最高为2M，所以用户需根据Fcpu的频率来设置分频系数（ADCS[2:0]）。

2、当Fcpu高于1 MHz时，仅当在休眠和绿色模式下进行转换时才推荐使用FRC时钟源。FRC就是内部IRC的分频，固定是500K。

9.3 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[3:0]
- 使能 ADC 模块，设置 ADEN

3. 配置ADC中断（可选）：

- 清零 ADC 中断标志
- 使能 ADC 中断
- 使能外设中断
- 使能全局中断

4. 等待所需采集时间

5. 设置ADON为1 启动一次ADC转换

6. 通过以下方式之一等待ADC转换完成：

- 查询 ADON 位
- 等待 ADC 中断（已使能中断）

7. 读取ADC结果

8. 清零ADC中断标志（如果已使能中断则需要）

➤ 例：配置AD，结果保留在BANK0的NTCADHIGH、NTCADLOW中。

```

.....;其他程序
AD_TEST:
    BSF      TRISA,0      ;设置AD口为输入
    MOVLW    B'01010000'  ;INNER REF Fsys/32 ADRESH[7:0] ADRESL[7,6]
    MOVWF    ADCON1
    
```

```

                                ;配置AD通道
MOVLW    B'00000001'
MOVWF    ANSELL                ;PA0作为模拟输入
CLRF     ANSELH
BCF      ADCON0,CHS2
BCF      ADCON0,CHS1
BCF      ADCON0,CHS0
BCF      ADCON1,VHS2
BCF      ADCON1,VHS1
BCF      ADCON1,VHS0          ;参考电压为内部VDD
NOP
NOP                                ;延时
BSF      ADCON0,ADEN          ;使能ADC
CALL     DELAY_1              ;延时，用户可自行完成
BSF      ADCON0,ADON          ;开始一次转换
AD_TEST_WAIT:
BTFSK    ADCON0,ADON          ;等待转换完成
GOTO     AD_TEST_WAIT
                                ;转换完成，保存结果
MOVF     ADRESH,W              ;LOAD THE AD HIGH 8 BITS TO W
MOVWF    NTCADHIGH            ;客户应用时注意BANK
MOVF     ADRESL,W              ;LOAD THE AD LOW 8 BITS TO W
MOVWF    NTCADLOW             ;客户应用时注意BANK

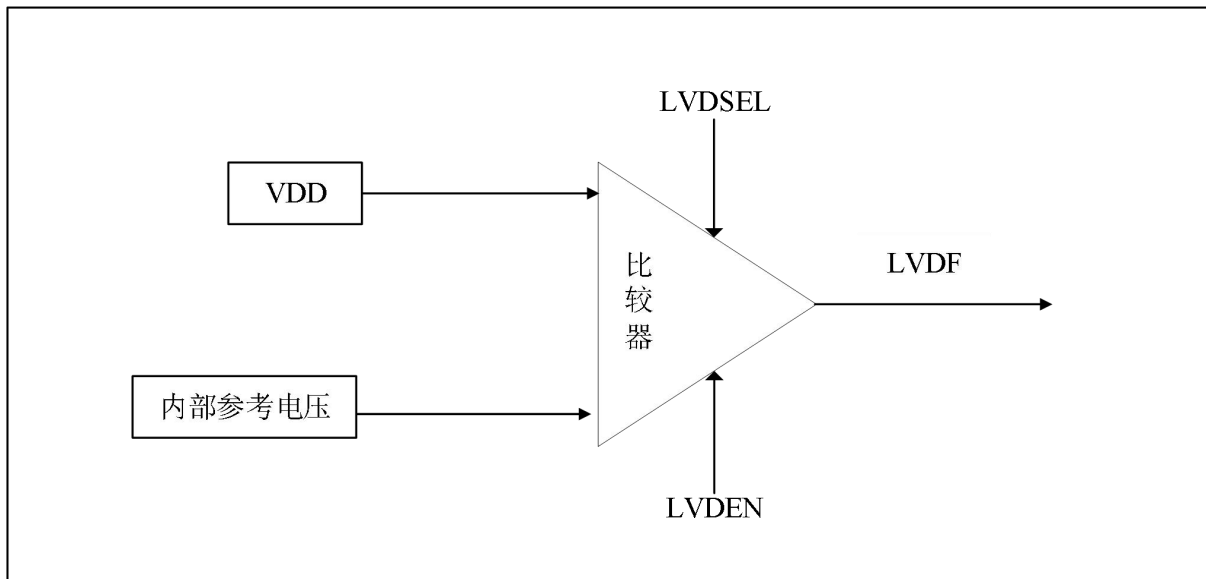
```

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，将ADC参考电压设为非内部VDD以降低功耗。
- 3、进SLEEP模式时，ADC采集通道请勿选择内部1/4VDD(AN5)。

10 LVD

LVD/CMP 工作原理框图:



芯片内置低电压检测模块 LVD，可通过寄存器位 LVDEN 开启，通过 LVDSEL 选择电压检测量值。
当 VDD 电压低于电压检测量值时检测状态标志位 LVDF 将被置 1

LVD控制寄存器

08h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LVDCR	LVDF	-	-	LVDSEL3	LVDSEL2	LVDSEL1	LVDSEL0	LVDEN
R/W	R	-	-	R/W	R/W	R/W	R/W	R/W
POR 的值	0	-	-	0	0	0	0	0

Bit [7]LVDF: LVD 检测标志位

0: VDD 电压高于电压检测量值

1: VDD 电压低于电压检测量值

Bit [4:1] LVDSEL[3:0]: VDD 电压检测档位

0000: 1.8V

0001: 2.0V

0010: 2.1V

0011: 2.2V

0100: 2.4V

0101: 2.5V

0110: 2.6V

0111: 2.7V

1000: 2.8V

1001: 2.9V

1010: 3.0V

1011: 3.2V

1100: 3.3V

1101: 3.6V

1110: 4.0V

1111: 4.2V

Bit [0] **LVDEN**: LVD 使能位

1: 开启 LVD

0: 关闭 LVD

11 EEPROM

HC18M121E2 内部集成有 2048 位的串行可擦除只读存储器（EEPROM），内部组织为 256 个字节，每个字节 8 位；擦写寿命达到 100 万次以上，数据保护时间长达 100 年。

通过 PORTA4/PORTA6/ 端口对 EEPROM 进行读写操作。

管脚对于如下表：

端口	EEPROM
PORTA4 (EEPROM_SCL)	串行时钟输入。同步数据传输，上升沿数据写入，下降沿数据读出。 通过 PORTA4 输出控制高低电平，控制 EEPROM_SCL 串行时钟。
PORTA6 (EEPROM_SDA)	串行地址和数据的输入/输出。 双向串行数据传输脚，漏极开路，PORTA6 需要 PORTA6 的内部上拉电阻打开。

1. **PORTA6:** EEPROM 的串行地址和数据的输入/输出口。

EEPROM 此管脚内部为开漏，在读取 EEPROM 的数据时，需要把 PORTA6 的内部上拉电阻打开。

2. **PORTA4:** EEPROM 的串行时钟输入。同步数据传输，上升沿数据写入，下降沿数据读出。

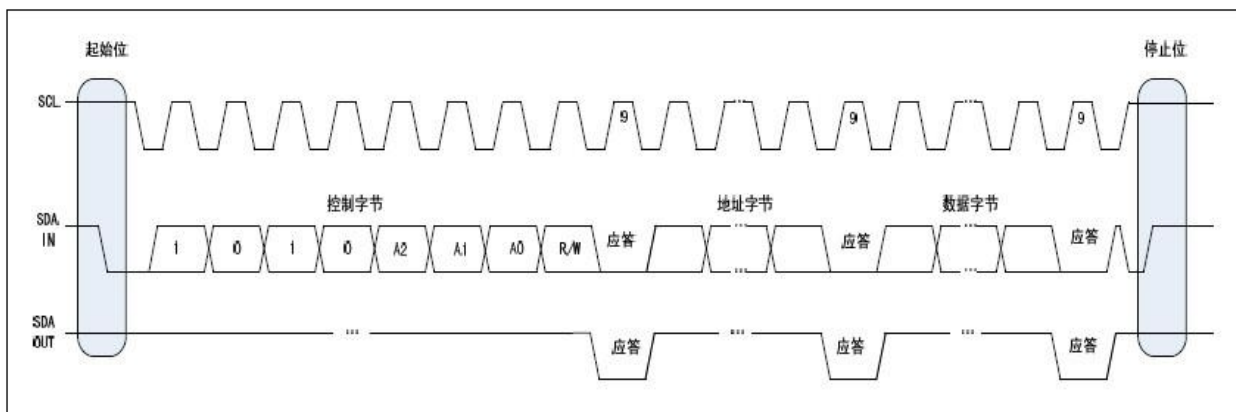
PORTA[4]、PORTA[6]端口仅作为读取或者存储 EEPROM 的数据使用，因此没有引到芯片外部。不能做其他功能应用。

EEPROM 器件操作共分三个部分：

- 控制操作
- 写操作
- 读操作

11.1 控制操作

控制操作时序如图：



EEPROM 器件控制操作共分六个部分：

- 复位
- 起始/停止命令
- 时钟及数据传输
- 控制字节
- 应答
- 等待模式

11.1.1 复位

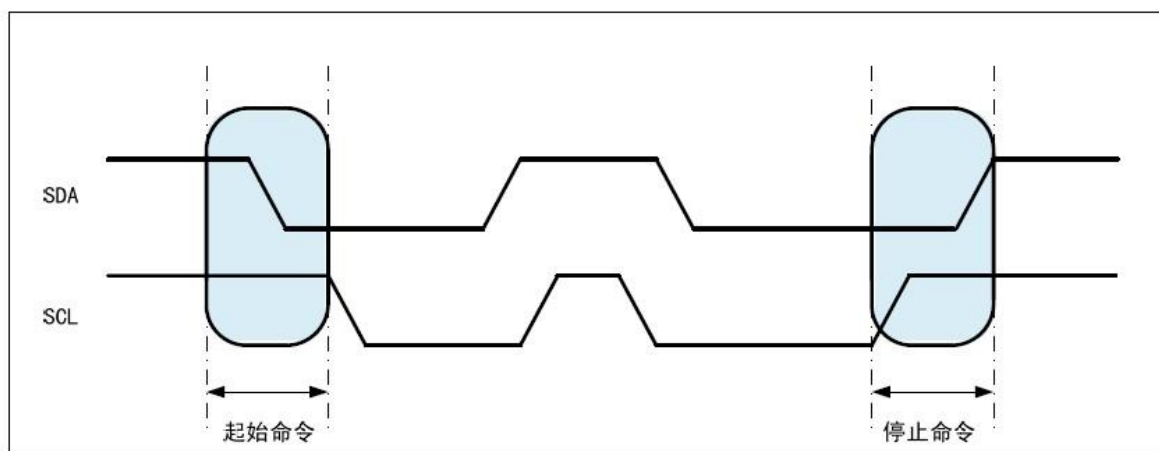
器件在协议中断、断电或系统复位后，可以通过以下步骤复位：

1. 连续输入9个时钟；
2. 在每个时钟周期中确保当SCL为高时SDA也为高；
3. 建立一个起始条件。

11.1.2 起始/停止命令

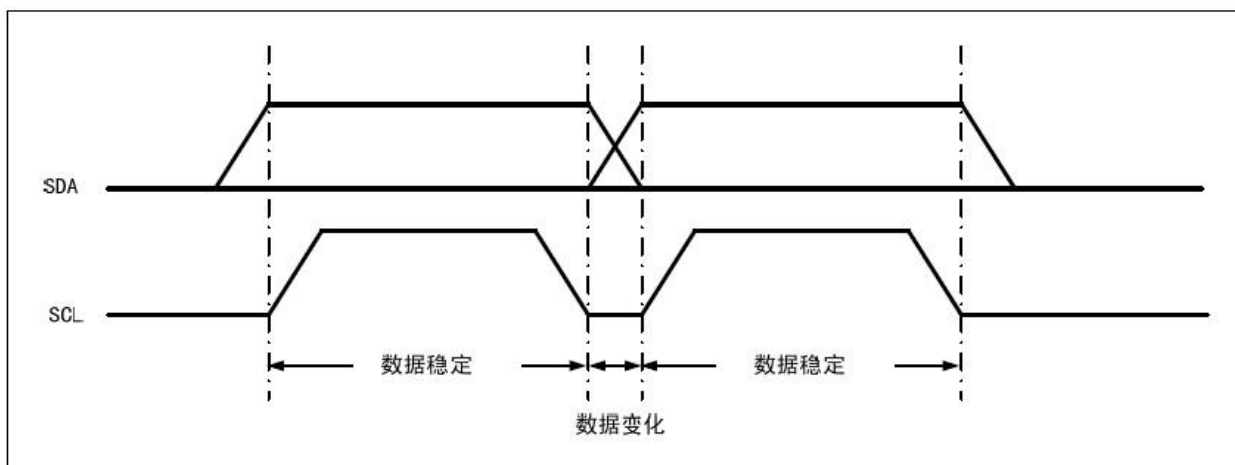
当 SCL 为高，SDA 由高到低的变化被视为起始命令，必须以起始命令作为任何一次读/写操作命令的开始（参考如下图）。

当 SCL 为高，SDA 由低到高的变化被视为停止命令，在一个读操作后，停止命令会使器件进入等待态低功耗模式（参考如下图）。



11.1.3 时钟及数据传输

SDA 引脚通常被外围器件拉高。SDA 引脚的数据应在 SCL 为低时变化（参考如下图）；当数据在 SCL 为高时变化，将视为上文所述的一个起始或停止命令。



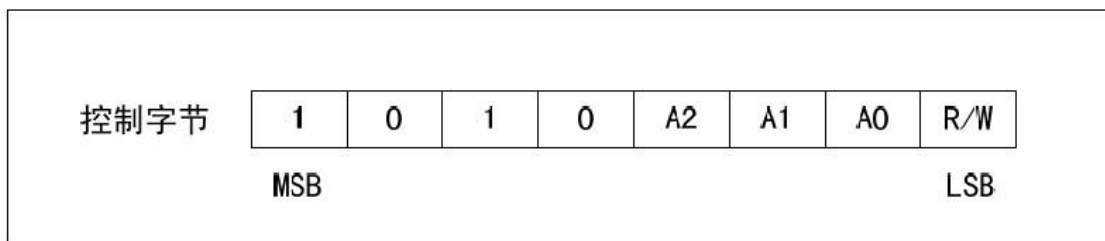
11.1.4 控制字节

器件在接到起始命令后，需要一个 8 位的控制字节来启动一次读/写操作（参考如下图）。

控制字节的前 4 位是一个固定的 1、0 数据，如图所示。后 3 位是 A2、A1 和 A0 引脚所对应的器件地址位，这 3 位必须与相应器件地址引脚的逻辑电平保持一致，才能正常操作。

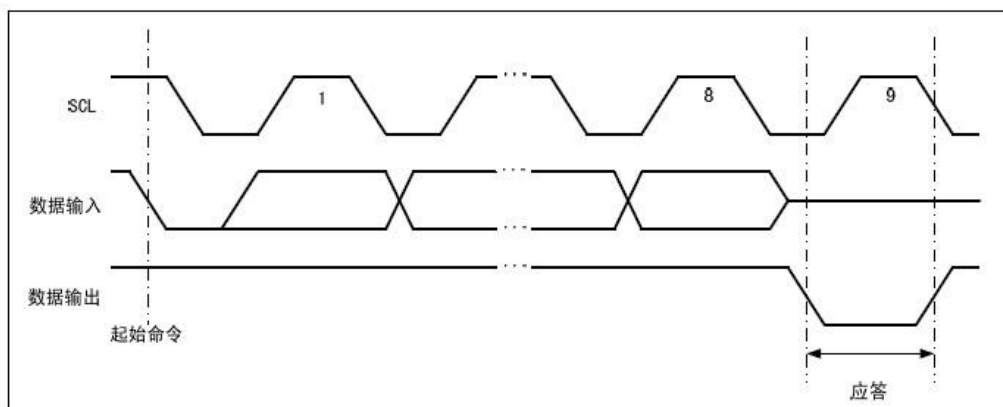
HC18M121E2 器件地址位固定设置为“000”（A2~A0）。

控制字节的第 8 位是读/写选择位，该位为高则启动读操作，为低则启动写操作。如果器件地址正确匹配，器件将应答一个“0”；否则，芯片将返回等待模式。



11.1.5 应答

所有的地址和数据字节都是以 8 位为一组串行输入和输出的。每当主控器件接收到一组 8 位的数据后，应当在第 9 个时钟周期向器件返回一个应答信号。



11.1.6 等待模式

EEPROM 特有一个低功耗的等待模式，可以通过以下方法进入该模式：

1. 上电；
2. 收到停止位并且结束所有的内部操作后。

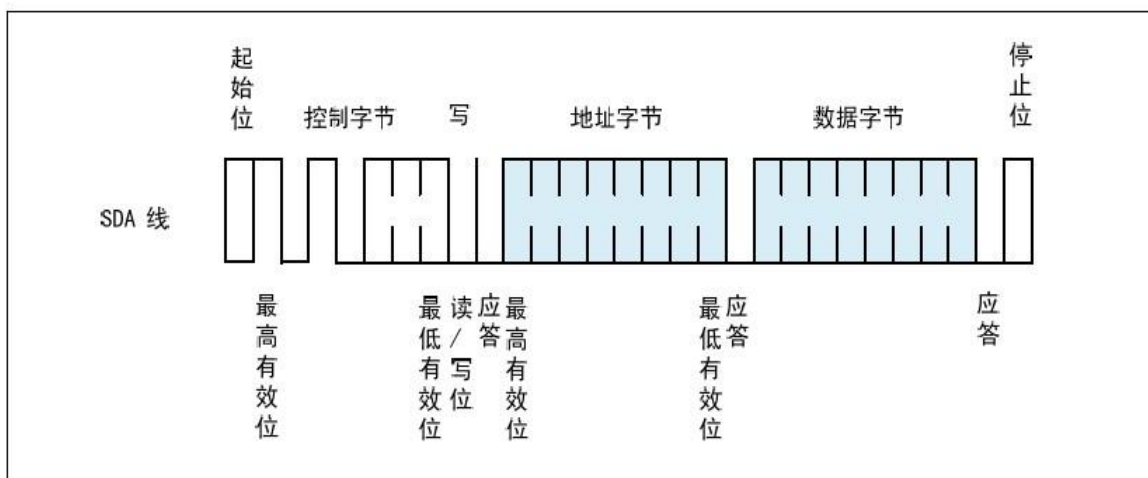
11.2 写操作

EEPROM器件共分两种写方式：

- 字节写
- 页写

11.2.1 字节写

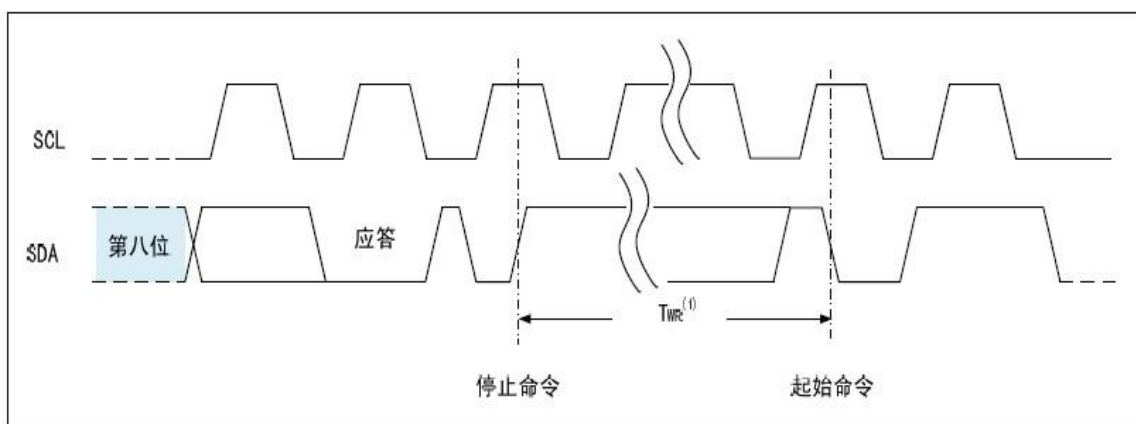
在输入器件地址并得到器件应答后，需要一个 8 位数据地址和一个 8 位数据来进行写操作；器件收到数据地址并再次返回应答信号后，时钟将前 8 位数据送入器件；接收到这 8 位数据后，器件返回应答信号，并且主控器件在收到停止命令后结束写操作。



此时器件进入内部定时的写周期（非易失性寄存器的写时间）， T_{WR} 。所有的输入操作在该写周期内均无效，而且只有在写周期结束后，器件才会对操作指令做出应答。（参考如下图）

SCL：串行时钟输入

SDA：串行数据输入/输出



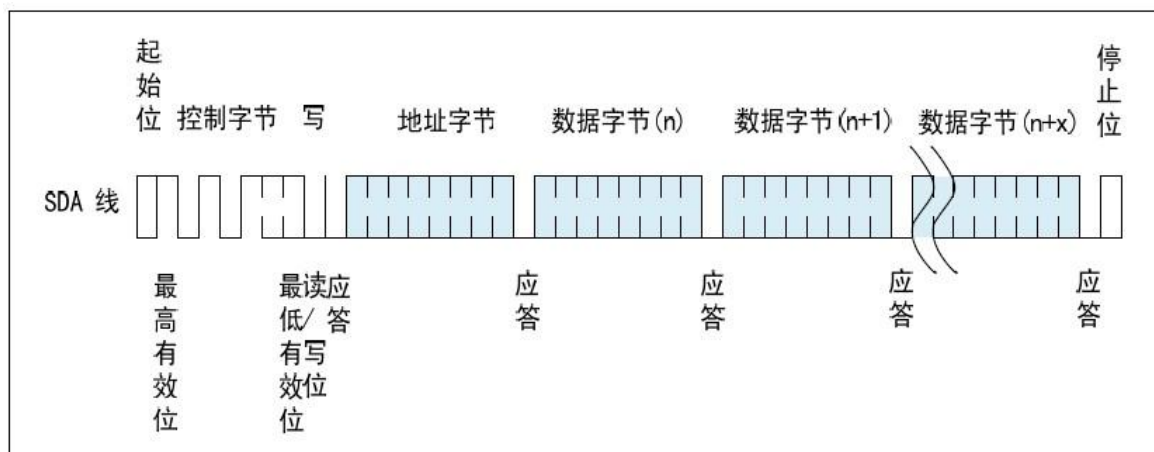
注：写周期 T_{WR} 是指一个写序列最后一个有效停止命令到内部擦/写周期结束的时间。

11.2.2 页写

器件能够实现 8 字节页写。

页写操作与字节写操作的启动方式相同。不同的是在时钟送入第一组数据并得到器件应答后，主控器件不是发出停止命令，而是继续发送其余七组数据。每收到一组数据器件都会返回应答信号“0”，主控器件必须以停止命令来结束页写操作（参考如下图）。

页写操作每接收一组数据，数据地址的低三位会在内部自动递加、数据地址的高五位不会变化，保持存储器的页地址不变。当内部产生的数据地址达到页边界时，数据地址将会翻转，接下来的数据的写入地址将被置为同一页的最小地址，先写入的数据将被覆盖。



11.2.3 应答轮询

一旦主控器件启动内部定时写周期并且器件输入被禁止，便可进行应答轮询。该过程包括发送一个带有器件地址的起始命令。读/写位由需要进行的操作决定。当内部写周期结束，器件返回应答信号后，主控器件即可执行下一个读/写命令。

11.3 读操作

EEPROM器件共分三种读方式：

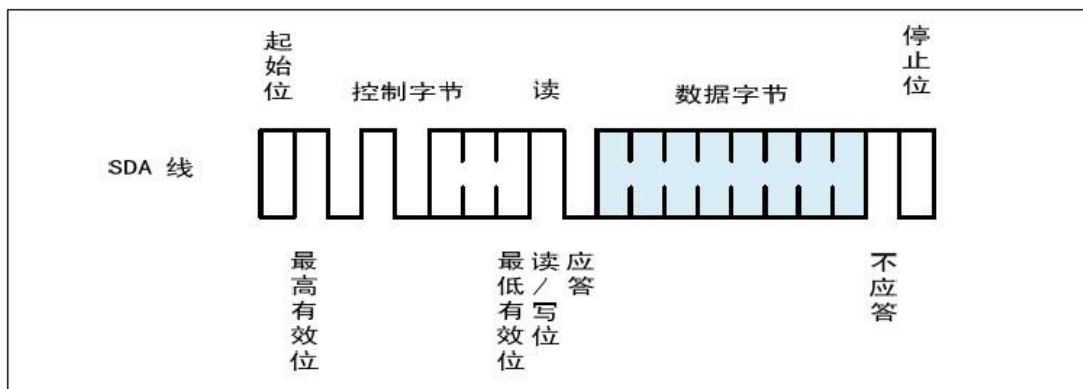
- 当前地址读
- 自由地址读
- 连续地址读

11.3.1 当前地址读

内部数据地址计数器保留最后一次访问的地址，并自动加 1。只要芯片处于上电状态，这个地址在操作运行器件始终有效。

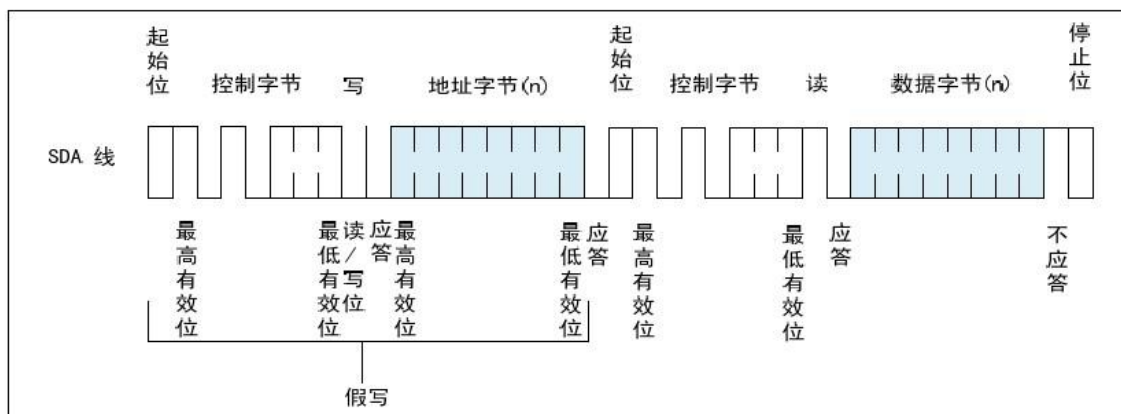
在读操作中，如果从存储器的最后一页的最后一个字节开始读，则读下一个字节时地址将会翻转到整个器件的最小地址；在写操作中，如果从当前页面的最后一个字节开始写，则写下一个字节时地址将会翻转到同一页内的最小地址。

一旦时钟将读/写位为高的器件地址送入，并得到器件应答后，就会串行输出当前地址的数据。主控器件不对器件返回应答信号，而是产生一个紧随的停止命令。（参考如下图）



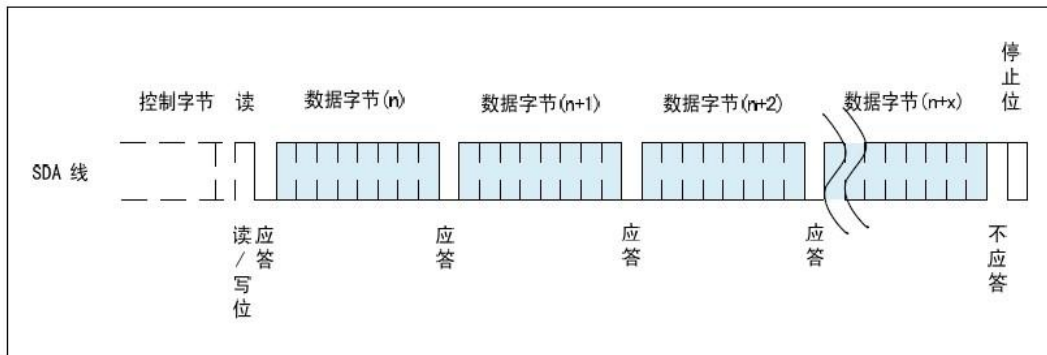
11.3.2 自由地址读

自由读需要通过假的字节写操作来获得数据地址。一旦器件地址和数据地址字节被时钟送入并得到器件的应答后，主控器件必须产生另一个起始命令。主控器件通过发送一个读/写选择位为高的器件地址来开启一次当前地址读。器件对器件地址做出应答后由时钟串行输出数据。主控器件不对数据传输返回应答信号“0”，而是产生一个紧随的停止命令。（参考如下图）



11.3.3 连续地址读

连续读由一个当前地址读或自由读启动。主控制器件收到一组数据后应当返回应答信号。器件每接收到一个应答信号，数据地址将被自动加 1，并且将串行输出下一组数据。当器件达到存储器的最大地址时，数据地址将翻转到最小地址，并且继续进行连续读操作。主控制器件不发出应答信号，而是产生一个紧随的停止命令来结束连续读操作。（参考如下图）



12 指令表

助记符	操作数	说明	周期数	受影响的状态位
ADDWF	f,d	W 和 f 相加	1	C, DC, Z
ADCWF	f,d	F+W+C	1	C, DC, Z
ADDLW	k	将立即数和 W 相加	1	C, DC, Z
SUBWF	f,d	f 减去 W	1	C, DC, Z
SBCWF	f,d	f-W-C	1	C, DC, Z
SUBLW	k	立即数减去 W	1	C, DC, Z
DAW	-	W 寄存器值进行 BCD 调整	1	C, DC
DSW		W 寄存器减法 BCD 调整		C, DC
ANDWF	f,d	W 和 f 作逻辑与运算	1	Z
ANDLW	k	立即数和 W 作逻辑与运算	1	Z
IORWF	f,d	W 和 f 作逻辑或运算	1	Z
IORLW	k	立即数和 W 作逻辑或运算	1	Z
XORWF	f,d	W 和 f 作逻辑异或运算	1	Z
XORLW	k	立即数和 W 作逻辑异或运算	1	Z
COMF	f,d	f 取反	1	Z
CLRWF	-	将 W 清零	1	Z
CLRF	f	将 f 清零	1	Z
INCF	f,d	f 加 1	1	Z
INCFSZ	f,d	f 加 1, 为 0 则跳过	1(2)	-
DECF	f,d	f 减 1	1	Z
DECFSZ	f,d	f 减 1, 为 0 则跳过	1(2)	-
BCF	f,d	将 f 中的 d 位清 0	1	-
BSF	f,d	将 f 中的 d 位置 1	1	-
BTFSF	f,d	检测 f 中的 d 位, 为 0 则跳过	1(2)	-
BTFSF	f,d	检测 f 中的 d 位, 为 1 则跳过	1(2)	-
MOVWF	f	将 W 的内容传送到 f	1	-
MOVF	f,d	将 f 的内容送到目标寄存器	1	Z
MOVLW	k	将立即数 k 传送到 W	1	-
RLF	f,d	对 f 执行带进位的循环左移	1	C
RRF	f,d	对 f 执行带进位的循环右移	1	C
SWAPF	f,d	将 f 的两个半字节进行交换	1	-
CALL	k	调用子程序	2	-
GOTO	k	无条件跳转	2	-
RETFIE	-	从中断返回	2	GIE
RETURN	-	从子程序返回	2	-
RETLW	k	返回时将立即数传送到 W	2	-
CLRWDI	-	清零看门狗定时器	1	TO, PD
SLEEP	-	进入待机模式	1	TO, PD
NOP	-	空操作	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	—	FCPU = 8MHz	2.7	-	5.5	V
			FCPU = 4MHz	2.0	-	5.5	
			FCPU = 2MHz	2.0	-	5.5	
			FCPU = 1MHz	2.0	-	5.5	
			FCPU = 500KHz	2.0	-	5.5	
			FCPU = 32KHz	2.0	-	5.5	
IDD1	工作电流	3V	FCPU = 8MHz, 高频模式	-	1.3	-	mA
		5V	WDT 禁止, 无负载	-	2.0	-	mA
IDD2	工作电流	3V	FCPU = 4MHz, 高频模式	-	0.8	-	mA
		5V	WDT 禁止, 无负载	-	1.2	-	mA
IDD3	工作电流	3V	FCPU = 2MHz, 高频模式	-	0.5	-	mA
		5V	WDT 禁止, 无负载	-	0.7	-	mA
IDD4	工作电流	3V	FCPU = 20KHz, 低频模式	-	9	-	μA
		5V	WDT 禁止, 无负载	-	15	-	μA
IDD5	工作电流	3V	FCPU = 20KHz, 绿色模式	-	-	-	μA
		5V	WDT 禁止, BOR 使能, 无负载	-	30	-	μA
Isb1	静态电流	3V	休眠模式, WDT 使能, 无负载, BOR 禁止	-	-	-	μA
		5V	休眠模式, WDT 使能, 无负载, BOR 禁止	-	5	-	μA
Isb2	静态电流	3V	休眠模式, WDT 使能, 无负载, BOR 使能	-	-	-	μA
		5V	休眠模式, WDT 使能, 无负载, BOR 使能	-	20	-	μA
Isb3	静态电流	3V	休眠模式, WDT 禁止, 无负载, BOR 禁止	-	-	1	μA
		5V	休眠模式, WDT 禁止, 无负载, BOR 禁止	-	-	1	μA
Isb4	静态电流	3V	休眠模式, WDT 禁止, 无负载, BOR 使能	-	-	-	μA
		5V	休眠模式, WDT 禁止, 无负载, BOR 使能	-	19	-	μA
ILC	端口输入漏电流	3V	端口输入模式, VIN=VDD 或 GND	-1	0	1	μA
		5V	端口输入模式, VIN=VDD 或 GND	-1	0	1	μA
VIL1	输入低电平	5V	非施密特输入口		0.5VDD		V
VIH1	输入高电平	5V			0.5VDD		V
VIL2	输入低电平	5V	施密特输入口		0.3VDD		V
VIH2	输入高电平	5V			0.7VDD		V
IOL1	输出灌电流	5V	输出口, Vout=VSS+0.5V		27		mA
IOH1	输出拉电流	5V	输出口, Vout=VDD-0.5V		15		mA
RPH1	内部上拉电阻	5V	可编程上拉电阻	-	120	-	kΩ

RPD	内部下拉电阻	5V	可编程下拉电阻	-	120	-	kΩ
VBO R	低电压复位	—	-	-5%	所选 BOR	+5%	V
VPO R	上电复位电压	—	-		1.2		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	Tset1	常温，VDD=5V	-	-	10	μs
内部 RC40K 启动时间	Tset2	常温，VDD=5V	-	-	150	μs
内部高频 RC 频率精度	FIRC1	VDD=1.8V~5.5V, 25°C	16(-1.0%)	16	16(1.5%)	MHz
内部高频 RC 频率精度	FIRC2	VDD=5.0V, -40°C ~+85°C	16(-1.0%)	16	16(1.5%)	MHz
内部低频 RC 频率精度	FWRC1	VDD=1.8V~5.5V, 25°C	-50%	40	+50%	KHz
内部低频 RC 频率精度	FWRC2	VDD=5.0V, -40°C ~+85°C	-50%	40	+50%	KHz
外部高频晶振	FOSH	2.0~5.5V	—	—	4	MHz
外部低频晶振	FOSL	2.0~5.5V	—	32.768	—	KHz
CPU 上电工作时间	TEST3	常温，VDD=5V，最快	200	250	300	μs

◆ 其他特性：

- ESD (HBM) : CLASS 3A (≥4000V)
- ESD (MM) : CLASS 2 (≥200V)
- Latch_up: CLASS I

◆ ADC 特性：

参数	符号	条件	最小值	典型值	最大值	单位
供电电压	VAD	-	2	5.0	5.5	V
精度	NR	$GND \leq VAIN \leq V_{ref}$	-	-	12	bit
ADC 输入电压	VAIN	-	GND	-	Vref	V
ADC 输入电阻	RAIN	VAIN=5V	2	-	-	MΩ
模拟电压源推荐阻抗	ZAIN	-	-	-	10	kΩ
ADC 转换电流	IAD	ADC 模块打开，VDD=5.0V	-	0.6	1	mA
ADC 输入电流	IADIN	VDD=5.0V	-	-	10	μA
微分非线性误差	DLE	VDD=5.0V, -40°C~+85°C	-2	-	+2	LSB
积分非线性误差 (1MHz 转换频率)	ILE1	VDD=5.0V, Vref=1.3V, 25°C	-5	-	+2	LSB
		VDD=5.0V, Vref=2V, 25°C	-5	-	+2	
		VDD=5.0V, Vref=3V, 25°C	-4	-	+2	
		VDD=5.0V, Vref=4V, 25°C	-3	-	+2	
		VDD=5.0V, Vref=VDD, 25°C	-2	-	+2	
		VDD=5.0V, Vref=外参, 25°C	-2	-	+2	
积分非线性误差 (1MHz 转换频率)	ILE2	VDD=5.0V, Vref=1.3V, -40°C	-1	-	+3	LSB
		VDD=5.0V, Vref=2V, -40°C	-1	-	+3	
		VDD=5.0V, Vref=3V, -40°C	-7	-	+2	
		VDD=5.0V, Vref=4V, -40°C	-4	-	+3	
		VDD=5.0V, Vref=VDD, -40°C	-2	-	+2	
积分非线性误差 (1MHz 转换频率)	ILE3	VDD=5.0V, Vref=1.3V, +85°C	-12	-	+2	LSB
		VDD=5.0V, Vref=2V, +85°C	-12	-	+2	
		VDD=5.0V, Vref=3V, +85°C	-9	-	+2	
		VDD=5.0V, Vref=4V, +85°C	-7	-	+2	
		VDD=5.0V, Vref=VDD, +85°C	-2	-	+3	
满刻度误差	EF	VDD=5.0V	-5	-	+5	LSB
偏移量误差	EZ	VDD=5.0V	-3	-	+5	LSB
总绝对误差	EAD	VDD=5.0V	-5	-	+5	LSB

14 开发工具

14.1 MTP 烧录器（HC-PM18 PRO）

- PM18 PRO：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。

14.2 HC-IDE

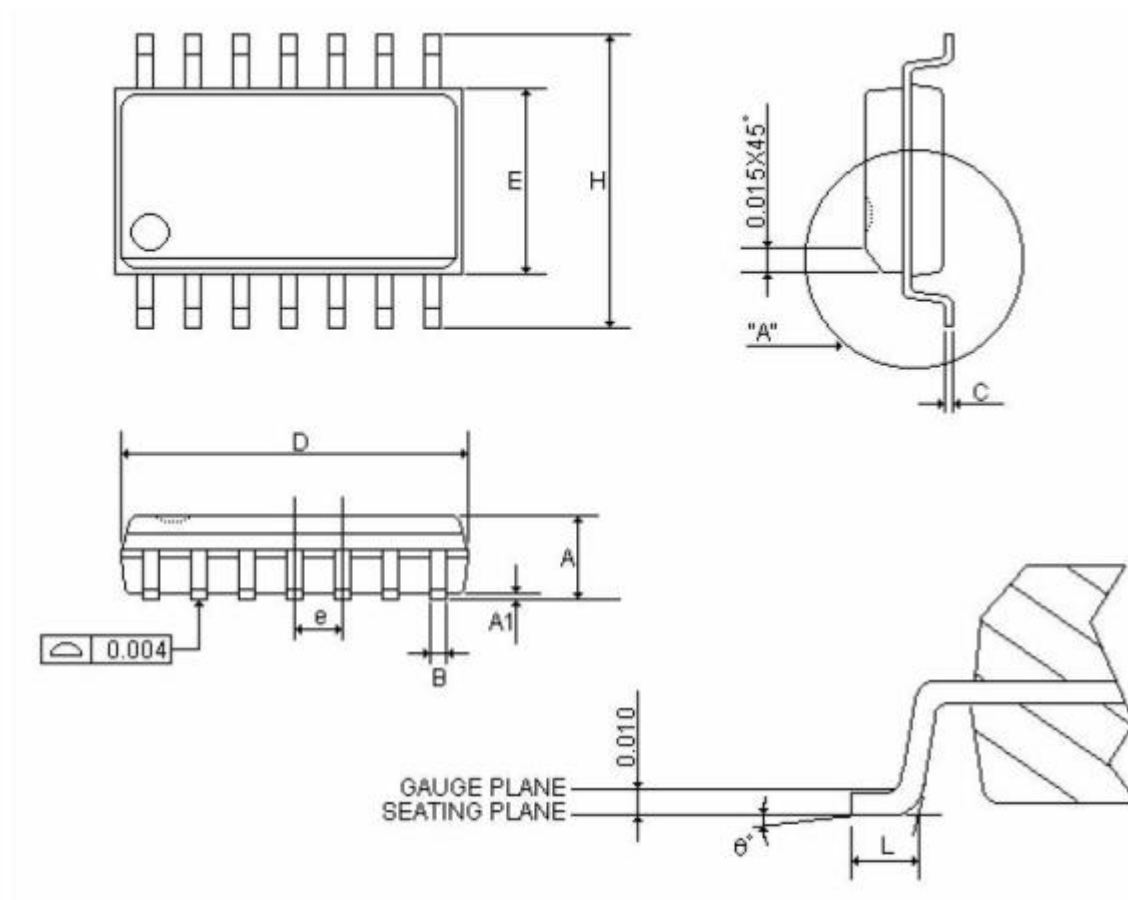
Holychip 8 位单片机的集成开发环境 HC-IDE 包括编译器、HC-PM18 Pro 下载烧录软件。

- HC-IDE：HC-IDE V3.0.x.x(支持汇编/C 语言)

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/>

15 封装信息



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°

16 数据手册版本修正记录

版本	日期	描述
Ver1.00	2024-02-22	初版
Ver1.01	2024-07-09	P36, OPTION寄存器错误修改; P54, ADRESE/ADRESH地址错误修改; 9.3章节, 例子中配置内参VDD有误, 修改相应代码。 4.2章节, 添加端口引入负压会产生复位的注意事项。

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

