

HC15P013A0

数据手册

8引脚8位
I/O型OTP单片机

目录

1 产品简介	4
1.1 功能特性	4
1.2 引脚图	6
1.3 引脚描述	6
2 中央处理器（CPU）	8
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）	11
2.1.4 芯片配置选择	14
2.2 寻址模式	14
2.2.1 立即寻址	15
2.2.2 直接寻址	15
2.2.3 间接寻址	15
2.3 堆栈	15
3 系统时钟	16
3.1 概述	16
3.2 时钟框图	16
3.3 系统高频时钟	17
3.3.4 内部高频 RC	17
3.4 系统低频时钟	17
3.3.5 内部低频 RC 振荡器	17
4 复位	18
4.1 概述	18
4.2 上电复位	18
4.3 WDT 复位	19
4.4 欠压复位	20
4.4.1 欠压复位的产生	20
4.4.2 工作死区	20
4.4.3 工作死区与工作频率的关系	21
4.4.4 死区防护	21
5 系统工作模式	22
5.1 概述	22
5.2 休眠模式	22
5.3 唤醒时间	23
6 中断	24
6.1 概述	24
6.2 中断请求和标志寄存器	24
6.3 GIE 全局中断	25
6.4 中断保护	25
6.5 INT0 中断	26

6.6 TIMER0 中断	27
6.7 端口电平变化中断	28
6.8 TIMER1 中断	29
7 I/O 端口	31
7.1 I/O 端口模式	31
7.2 I/O 上拉模式	31
7.3 I/O 下拉模式	32
7.4 I/O 开漏模式	32
7.5 I/O 端口数据寄存器	33
8 定时器	34
8.1 看门狗定时器	34
8.2 TIMER0 定时/计数器	36
8.3 TIMER1 定时/计数器	38
8.3.1 功能概述	38
8.3.2 T1 使用操作说明	39
8.3.3 T1 相关寄存器	39
9 LVD	41
10 指令表	43
11 电气特性	44
12 开发工具	46
12.1 OTP 烧录器 (HC-PM18 4.0)	46
12.2 HC-IDE	46
13 封装信息	47
13.1 SOP8	47
14 数据手册版本修正记录	48

1 产品简介

HC15P013A0 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，内有 1K*14 位一次性可编程 ROM（OTP-ROM），48*8 位的数据存储器（RAM），一组双向 I/O 口，2 个 8 位定时器/计数器，1 路 PWM，多级 LVD 检测。这款单片机可以广泛应用于简单控制和小家电等产品。

1.1 功能特性

- ◆ **存储器配置**
 - 程序存储器（OTP ROM）空间：1K*14位
 - 数据存储器（RAM）空间：48*8位
- ◆ **强大的指令系统**
 - 时钟系统可设（2T）
 - 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
 - 输入输出双向端口：PORTB<5:0>
 - 具有唤醒功能的电平变化中断端口：PORTB，可通过IOCB独立配置
 - 具有唤醒功能的外部中断引脚：PORTB<0>，可设置触发边沿
- ◆ **BOR**
 - 4 级低电压复位
- ◆ **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
 - INT0外部中断
 - 端口电平变化中断
- ◆ **定时器**
 - 看门狗计数器（WDT）
 - 1 个8位定时器
 - 1 个PWM功能的8位定时器
- ◆ **系统时钟**
 - 内建高精度8MHz RC时钟
 - 内建60KHz低频RC时钟
- ◆ **工作模式**
 - 高频模式
 - 低频模式
 - 休眠模式（可开启/关闭BOR）
- ◆ **复位**
 - 上电复位
 - BOR欠压复位

- WDT溢出复位

- ◆ 封装

- DIP8/SOP8

- SOT23-6

使用注意事项:

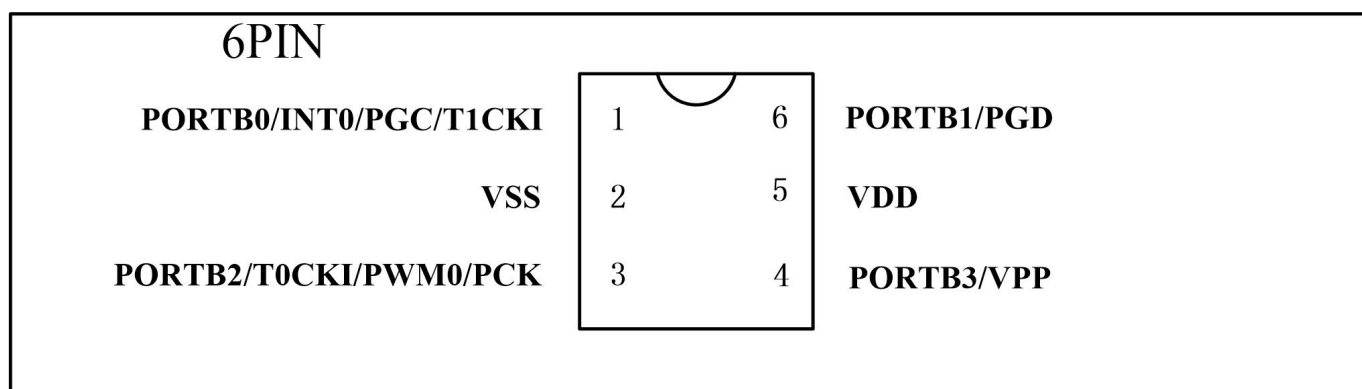
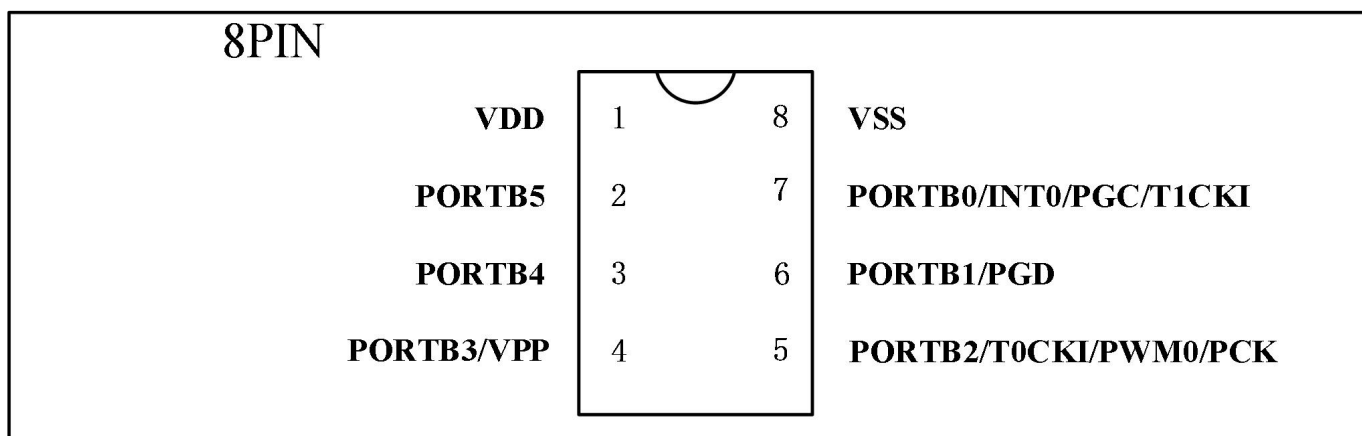
1、BOR 2.0v 以下，主频不能选择 8M/2T

2、VDD 在 1.3v 左右及以上时，如果 VPP 大于 VDD2.5V,芯片会进入测试模式，所以应用上请避免 VDD 和 VPP 同时上电，VDD 先上电芯片工作后，再上 VPP 不会进入测试模式。

3. 若芯片配置为高频，进入 sleep 唤醒后，如果需要打开 Bor 或者 LVD，需要延时 100us 后再打开；若芯片配置为低频，进入 sleep 后唤醒后不能再开启 Bor 和 LVD 功能，开启会导致芯片复位。

4. 芯片 eft 能力较弱，避免在高干扰环境下使用。

1.2 引脚图



1.3 引脚描述

脚位	名称	类型	说明
1	VDD	P	电源输入
2	PORTB5	I/O	输入/输出口，带可编程上/下拉电阻，可设置为开漏输出
3	PORTB4	I/O	输入/输出口，带可编程上拉/下拉电阻，可设置为开漏输出
4	PORTB3 VPP	I/O P	输入口，带可编程上/下拉电阻，开漏输出 编程高压电源输入
5	PORTB2 T0CKI PCK PWM0	I/O I O O	输入/输出口，带可编程上/下拉电阻，可设置为开漏输出 Timer0 外部计数时钟输入口 工具校准输出口 PWM0 输出

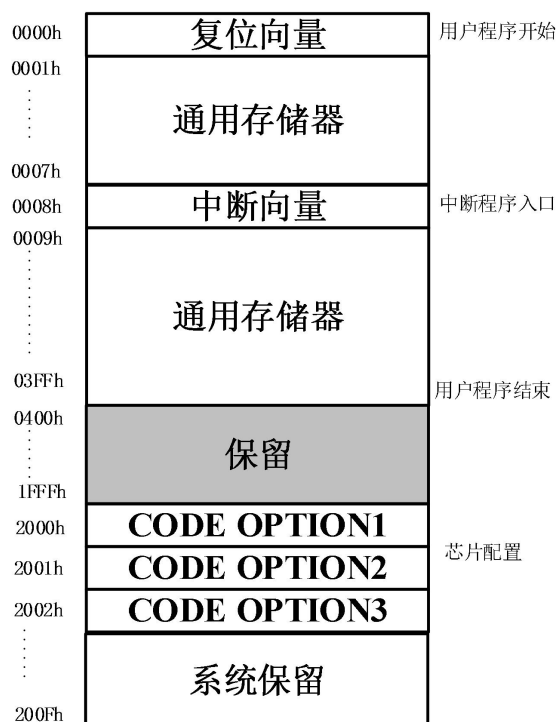
6	PORTB1 PGD	I/O I/O	输入/输出口，带可编程上拉电阻/下拉电阻，可设置为开漏输出 编程数据口
7	PORTB0 INT0 PGC TICKI	I/O I I I	输入/输出口，带可编程上拉电阻/下拉电阻，可设置为开漏输出 外部中断输入口 编程时钟输入口 T1 时钟输入
8	VSS	P	电源地

注: I = 输入 O = 输出 I/O = 输入/ 输出 P = 电源

2 中央处理器（CPU）

2.1 存储器

2.1.1 程序存储器（OTP ROM）



2.1.1.1 复位向量（0000h）

复位向量为0000h

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

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    AUXR,POR
    GOTO     ISPOR                      ;POR标志为0，判定为上电复位
    BTFSS    AUXR,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:
    ..
ISPOR:
    BSF      AUXR,POR                  ;上电复位处理程序
    ...
ISBOR:
    BSF      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
    ...

```

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      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 48 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

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	-	-	TO	PD	Z	DC	C
04h	FSR	数据指针寄存器							
06h	PORTB	-	-	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
08h	LVDCLR	LVDIF	-	-	LVDSEL3	LVDSEL2	LVDSEL1	LVDSEL0	LVDEN

09h	IOCB	-	-	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0
0Ah	PCLATH	-	-	-	-	-	-	程序计数器高2位缓存器	
0Bh	PDCON	-	-	PDB5	PDB4	PDB3	PDB2	PDB1	PDB0
0Ch	ODCON	-	-	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0
0Dh	PHCON	-	-	PHB5	PHB4	PHB3	PHB2	PHB1	PHB0
0Eh	INTECON	GIE	-	INTF	PBIF	T0IF	INTE	PBIE	T0IE
41h	OPTION	WDTEN	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
46h	TRISB	-	-	TRISB 5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB 0
4Bh	TICR_AUX R	T0CK	-	-	POR	BOR	BOREN	T1IE	T1IF
4Ch	TICON	T1EN	PWM0E	-	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							

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	据指针寄存器							
R/W	R	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 10 位宽，低字节来自可读写的 PCL 寄存器，高字节（PC[9:8]）不可读写，可通过 PCLATH 寄存器间接写入。如果对 PCL 进行赋值，PCLATH 也不会改变。

程序计数器高2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2位	
R/W	-	-	-	-	-	-	R/W	R/W
POR的值	-	-	-	-	-	-	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[9:0]（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[9:8], ALU[7:0] (ALU 运算结果) → PC

➤ 其它 PCL 作为目的操作数指令

F-MCU: PCLATH[9:8], 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	-	-	TO	PD	Z	DC	C
R/W	R/W	-	-	R	R	R/W	R/W	R/W
POR的值	0	-	-	1	1	x	x	x

Bit [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 [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-AUXR	T0CK			POR	BOR	BOREN	T1IE	T1IF
R/W	R/W			R/W	R/W	R/W	R/W	R/W
POR的值	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 [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

HC15P013A0 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
时钟模式	2T	1个时钟周期由2个内部RC振荡器时钟组成(固定为2T)
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: 60KHz	
	内部高频RC振荡器: 8MHz	
	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

HC15P013A0 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。

2.2.2 直接寻址

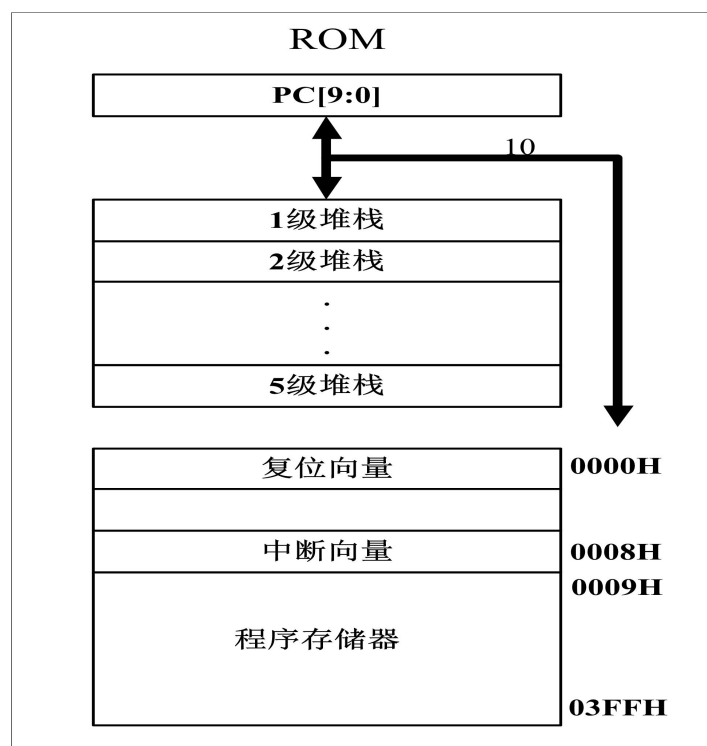
寄存器参与运算的寻址方式。

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。

2.3 堆栈

HC15P013A0 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。

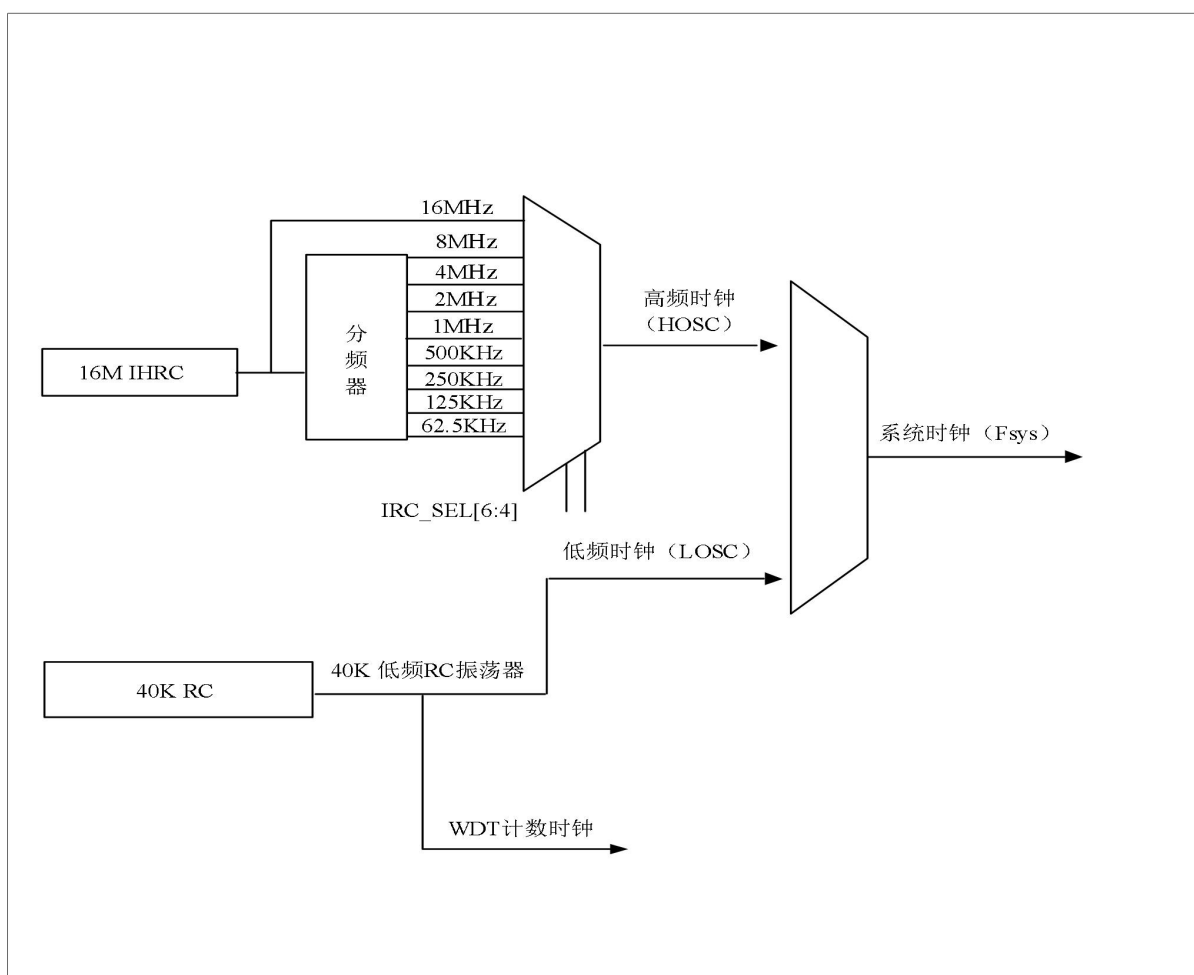
3 系统时钟

3.1 概述

HC15P013A0 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 8MHz RC 振荡电路（RC 8MHz）提供。低频时钟的时钟源则由内部低频 RC 振荡电路（RC 60KHz@5V）提供。两种时钟都可作为系统时钟源 F_{osc} 。

- 高频模式： $F_{cpu} = F_{sys} / 2$
- 低频模式： $F_{cpu} = F_{sys} / 2$

3.2 时钟框图



- F_{osc} : 时钟源频率
- F_{sys} : 系统时钟频率
- F_{cpu} : 指令时钟频率

3.3 系统高频时钟

系统高频时钟为内部高频 8M RC。

3.3.1 内部高频 RC

内置高频 RC 振荡器有、8MHz、4MHz、2MHz、1MHz、500KHz、250KHz、125KHz、62.5KHz 八种可选。

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），其中烧录器校准值写到 OPTION 的 0x2001 地址的低 6 位。

3.4 系统低频时钟

系统低频时钟为内部低频 RC60KHz。

3.3.2 内部低频 RC 振荡器

内部低频 RC 振荡器的频率为 60KHz，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 60KHz（典型值）。

4 复位

4.1 概述

HC15P013A0 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）

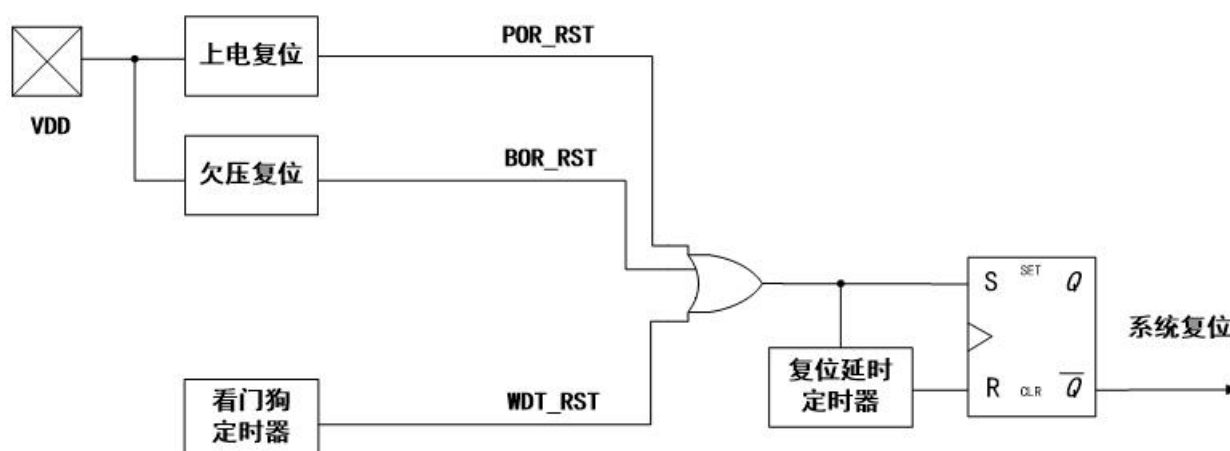
当上述任何一种复位产生时，系统进入复位状态，所有的特殊功能寄存器被初始化，程序停止运行，同时程序计数器（PC）清零。经过上电延时定时器延时后，系统结束复位状态，程序从 000h 地址开始执行。STATUS 寄存器的 Bit4（TO 位）及 T1CR_AUXR 寄存器的 Bit3（BOR 位）、Bit4（POR 位）显示系统复位状态信息，可通过这 3 个标志位判断复位来源，从而控制系统的运行路径。

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

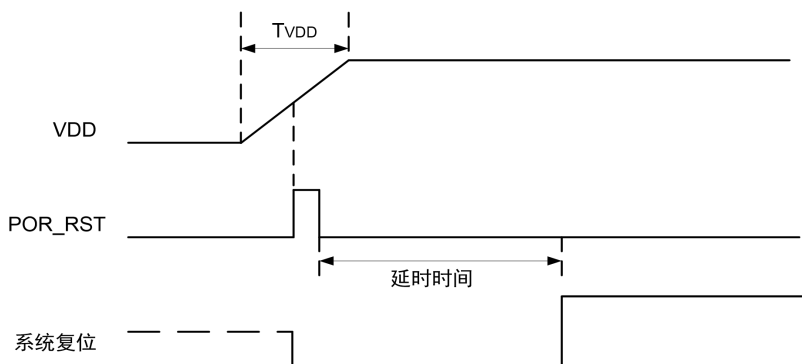
复位方式	复位延时定时器时间（典型值）	
上电复位		OPTION 选择
欠压复位		OPTION 选择
看门狗定时器复位		OPTION 选择

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 T1CR_AUXR 寄存器的 Bit3 (BOR 位)、Bit4 (POR 位) 来判断是否发生上电复位。VDD 最大上升时间 TVDD 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:



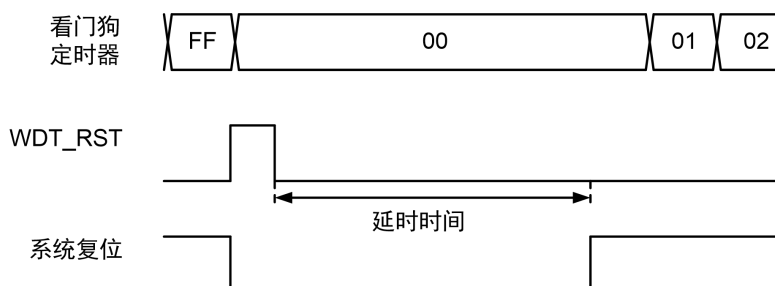
关于上电复位, 请注意以下几点:

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, 才能使用看门狗定时器。

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：

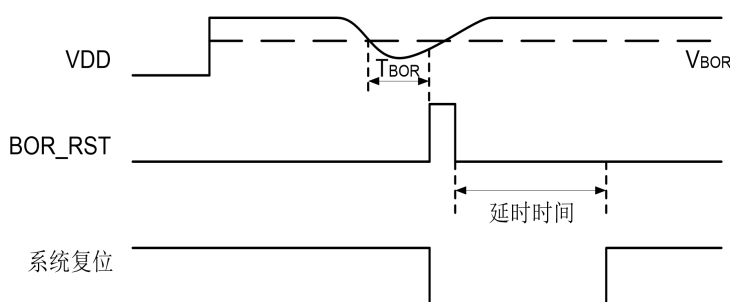
1. 主程序中有一次清看门狗的动作，这种架构能够最大限度的发挥看门狗的保护功能。看门狗的使能逻辑：看门狗使能 = 看门狗配置字使能 & 看门狗软件使能（WDTEN=1）。
2. 不建议在中断程序中对看门狗进行清零，否则无法监控主程序跑飞情况。

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：

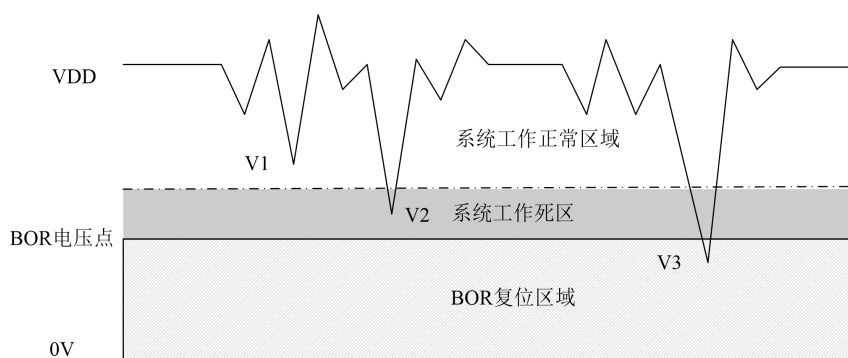


注： T_{BOR} 需大于 200ns，否则电压跌落时可能不产生欠压复位信号。

特别说明：电压检测电路有一定的迟滞特性，迟滞电压为 0.1V 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+0.1V 时 BOR 复位才会解除。

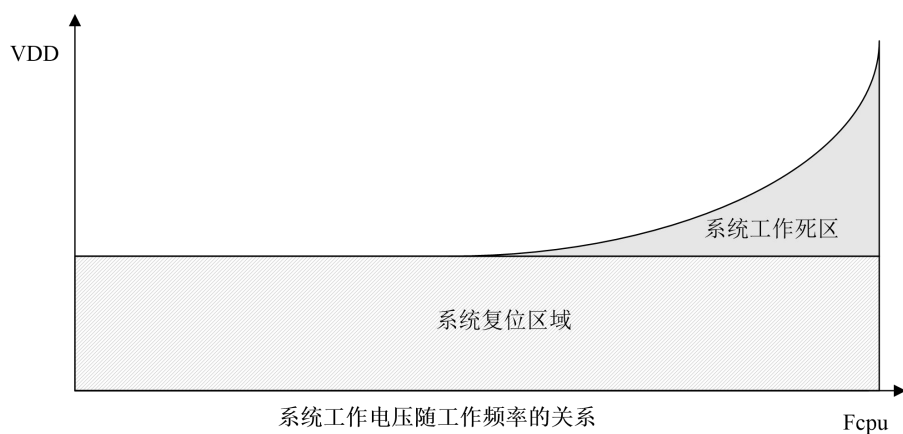
4.4.2 工作死区

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

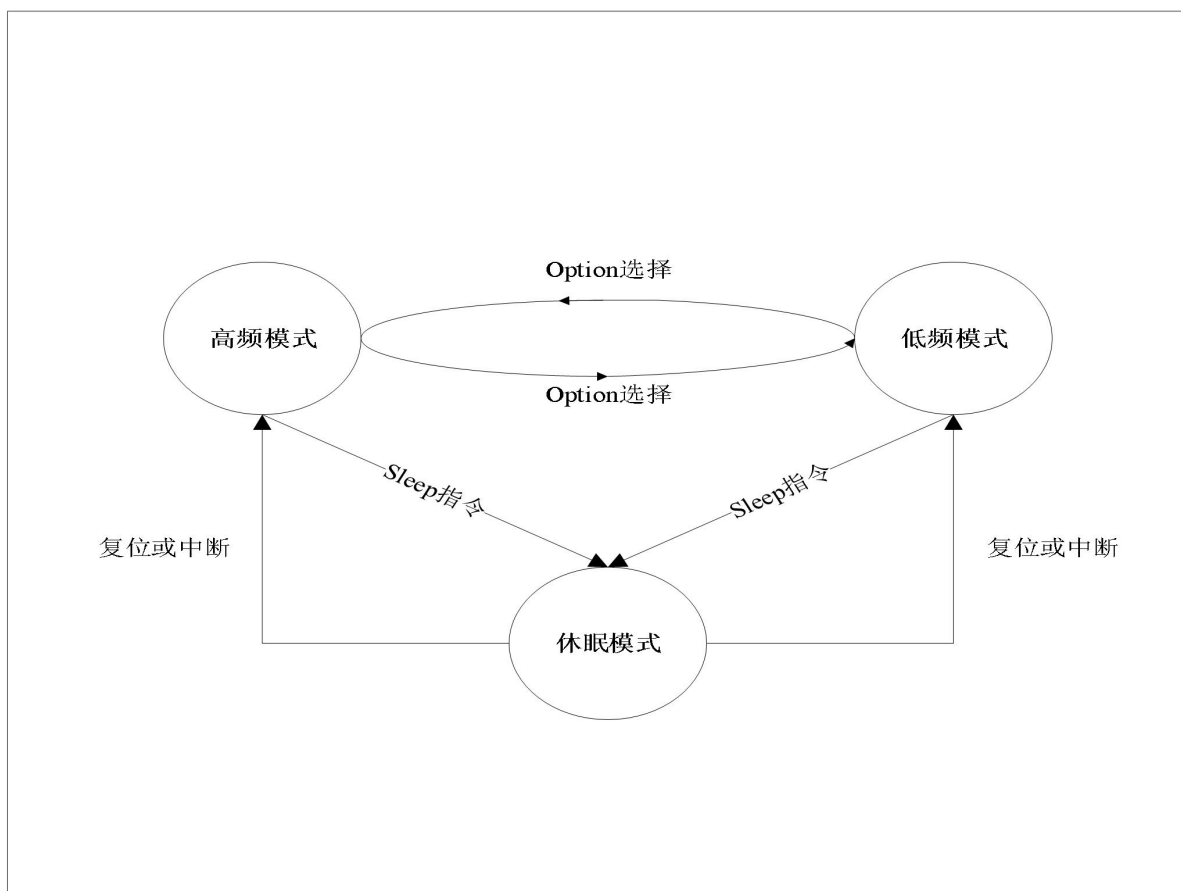
5 系统工作模式

5.1 概述

HC15P013A0 可在如下三种工作模式之间进行切换：

- 高频模式
- 低频模式
- 休眠模式

系统复位后，工作于高频模式还是低频模式，由配置字决定。



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
 - 定时器1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。

5.3 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	4 Clock

6 中断

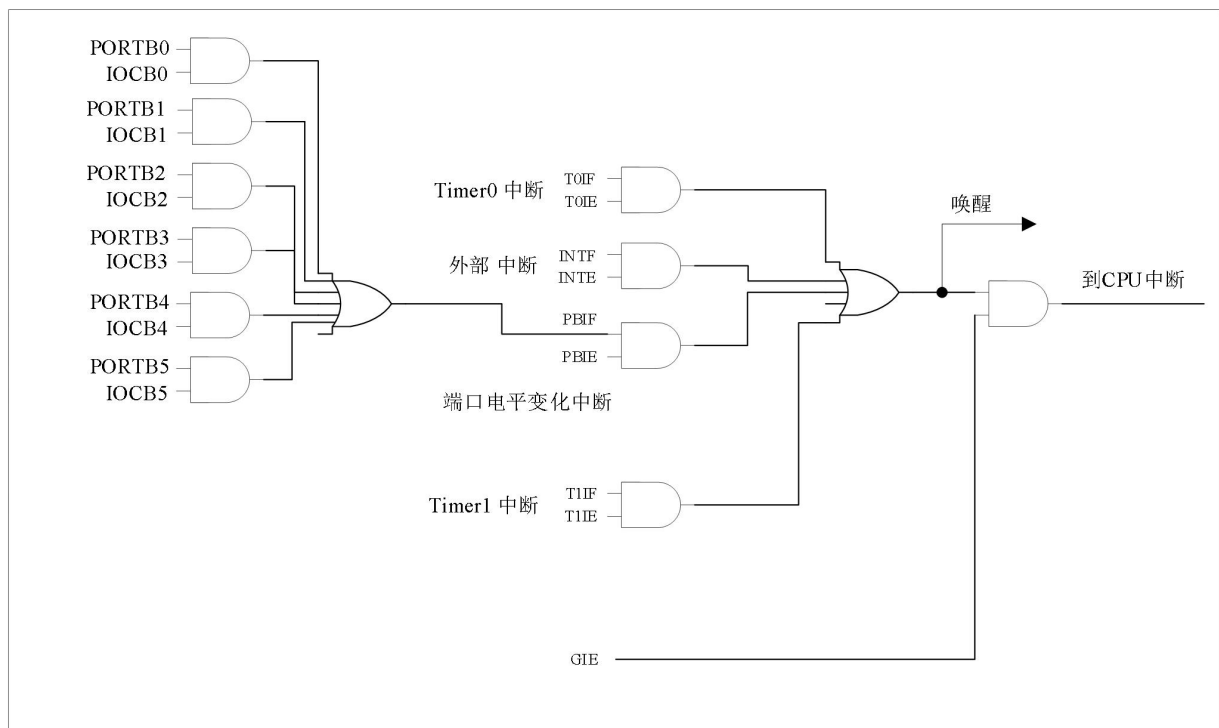
6.1 概述

HC15P013A0 提供 4 个中断源：

- Timer0 定时器中断
- INT0 外部中断
- 端口电平变化中断
- Timer1 定时器中断

系统从高频或低频模式进入睡眠模式时，INT0 外部中断、端口电平变化中断和 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：



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	-	INTF	PBIF	TOIF	INTE	PBIE	TOIE

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] **GIE**: 中断总使能

1 = 使能所有中断

0 = 屏蔽所有中断

Bit [5] **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 [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

0 = PORTB 未产生端口电平变化中断

Bit [3] **TOIF**: 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

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] **TOIE**: Timer0 溢出中断使能位

1 = 使能 Timer0 溢出中断

0 = 屏蔽 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

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。

➤ 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

```

6.5 INT0 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。

OPTION 寄存器

41h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	WDTEN	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**: INT0 中断边沿选择

1 = INT0 上升沿中断

0 = INT0 下降沿中断

➤ 例：INT0 中断请求设置，电平触发。

```

    BSF      OPTION,INTEDG    ;INT0置为上升沿触发
    BCF      INTECON,INTF     ;INT0中断请求标志清零
    BSF      INTECON,INTE     ;使能INT0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
        BTFSS    INTECON,INTF     ;检测INT0IF
        GOTO     EXIT_INT         ;INT0IF = 0, 退出中断
        BCF      INTECON,INTF     ;INT0IF清零
        ...                      ;INT0中断服务程序

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

```

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。

INTECON寄存器

0Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTECON	GIE	-	INTF	PBIF	T0IF	INTE	PBIE	T0IE
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 [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

Bit [3] **T0IF:** Timer0 溢出中断标志位

1 = Timer0 产生 Timer0 溢出中断

0 = Timer0 未产生 Timer0 溢出中断

➤ 例：T0中断请求设置。

```

        BCF      T0CR,T0CK        ;T0计数时钟由T0CS决定
        MOVLW    0X15
        MOVWF    OPTION           ;T0时钟 = Fcpu / 64
        MOVLW    0X40             ;T0初始值 = 64D
        MOVWF    T0
        BSF      INTECON,T0IE     ;置T0中断使能标志
        BCF      INTECON,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PCLATH

```

T0ISR:

```

BTFSS    INTECON,T0IF    ;检查是否有T0中断请求标志
GOTO     EXIT_INT        ;T0IF = 0, 退出中断
BCF      INTECON,T0IF    ;清T0IF
MOVLW    0X40
MOVWF    T0              ;重置T0值
...      ; T0中断程序

```

EXIT_INT:

```

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

```

6.7 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。

IOCB 寄存器

09h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOCB	-	-	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	-	-	0	0	0	0	0	0

Bit [7:0] **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     INTECON,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   INTECON,PBIF       ;检测PBIF
        GOTO    EXIT_INT           ;PBIF = 0，退出中断
        MOVF    PORTB,W           ;读PORTB端口
        BCF     INTECON,PBIF       ;PBIF清零
        ...                          ;PORTB电平变化中断服务程序
        ...
EXIT_INT:
        ...                          ;恢复STATUS、W和PCLATH
        RETFIE                      ;退出中断

```

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     INTECON,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,否则进入中断
        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
T1CR_AUXR	T0CK	-	-	POR	BOR	BOREN	T1IE	T1IF
R/W	R/W	-	-	R/W	R/W	R/W	R/W	R/W
POR 的值	0	-	-	q	q	1	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      T1CR_AUXR,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
BTFSS    T1CR_AUXR,T1IF   ;检测T1IF
GOTO     EXIT_INT         ;T1IF = 0，退出中断
BCF      T1CR_AUXR,T1IF   ;T1IF 清零
...      ;TIMER1中断服务程序
...

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

```

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	-	-	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	-	-	1	1	1	1	1	1

Bit [5:0] **TRISBx**: PORTBx 模式控制位

1 = 输入模式

0 = 输出模式

➤ 例: I/O 模式选择。

```

MOV LW    0XFF           ;所有端口设为输入模式
MOV WRF   TRISB

CLRF      TRISB           ;所有端口设为输出模式

BSF       TRISB,2         ;PORTB2设为输入模式
BCF       TRISB,2         ;PORTB2设为输出模式

```

注：以上方向控制当相应位为 0 时，对应端口为输出；为 1 时，相应端口为输入。

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	-	-	PHB5	PHB4	PHB3	PHB2	PHB1	PHB0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	-	-	1	1	1	1	1	1

Bit [5:0] **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。

```

CLRF      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	-	-	PDB5	PDB4	PDB3	PDB2	PDB1	PDB0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	-	-	1	1	1	1	1	1

Bit [5:0] **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

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 ;使能所有下拉
BSF PDCON,2 ;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	-	-	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	-	-	0	0	0	0	0	0

Bit [5:0] **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

BCF ODCON,2 ;禁止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	-	-	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR的值	-	-	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清0
```

8 定时器

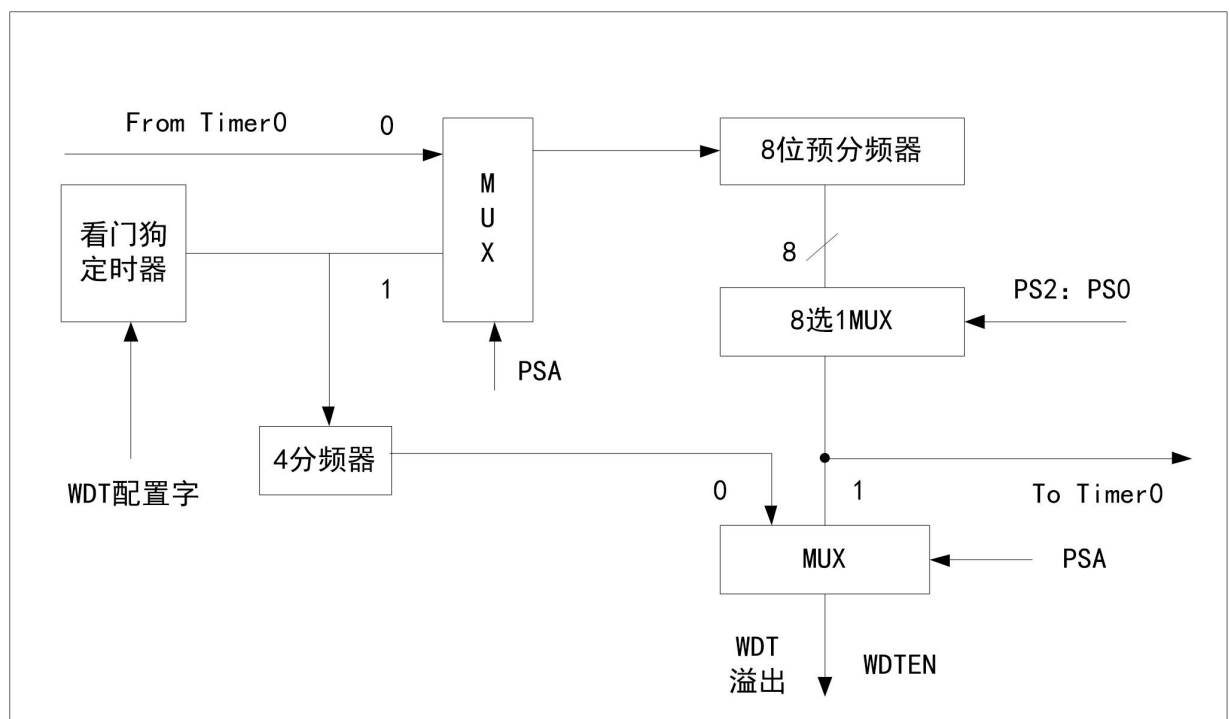
8.1 看门狗定时器

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 振荡器是否开启由 OPTION 中的 WDTE 和软件的 WDTEN 位共同决定。只有 WDTE 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 $14.4 \times 4 = 57.6\text{ms}$ 。

看门狗定时器和预分频器框图



OPTION 寄存器

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

--	--	--	--	--	--	--	--	--

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，否则无法检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 41h,7 ; 软件使能 WDT

...

CLRWDT ; 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      OPTION,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
CLRWDT   ;在整个程序中，仅有一条清狗指令
...
CALL     SUB1
CALL     SUB2
...
GOTO     MAIN

```

➤ 例：在休眠状态下，屏蔽看门狗功能，可以节省系统功耗。

```

...
BCF      OPTION,WDTEN    ;软件屏蔽看门狗功能
SLEEP   ;进入休眠模式
NOP
BSF      OPTION,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

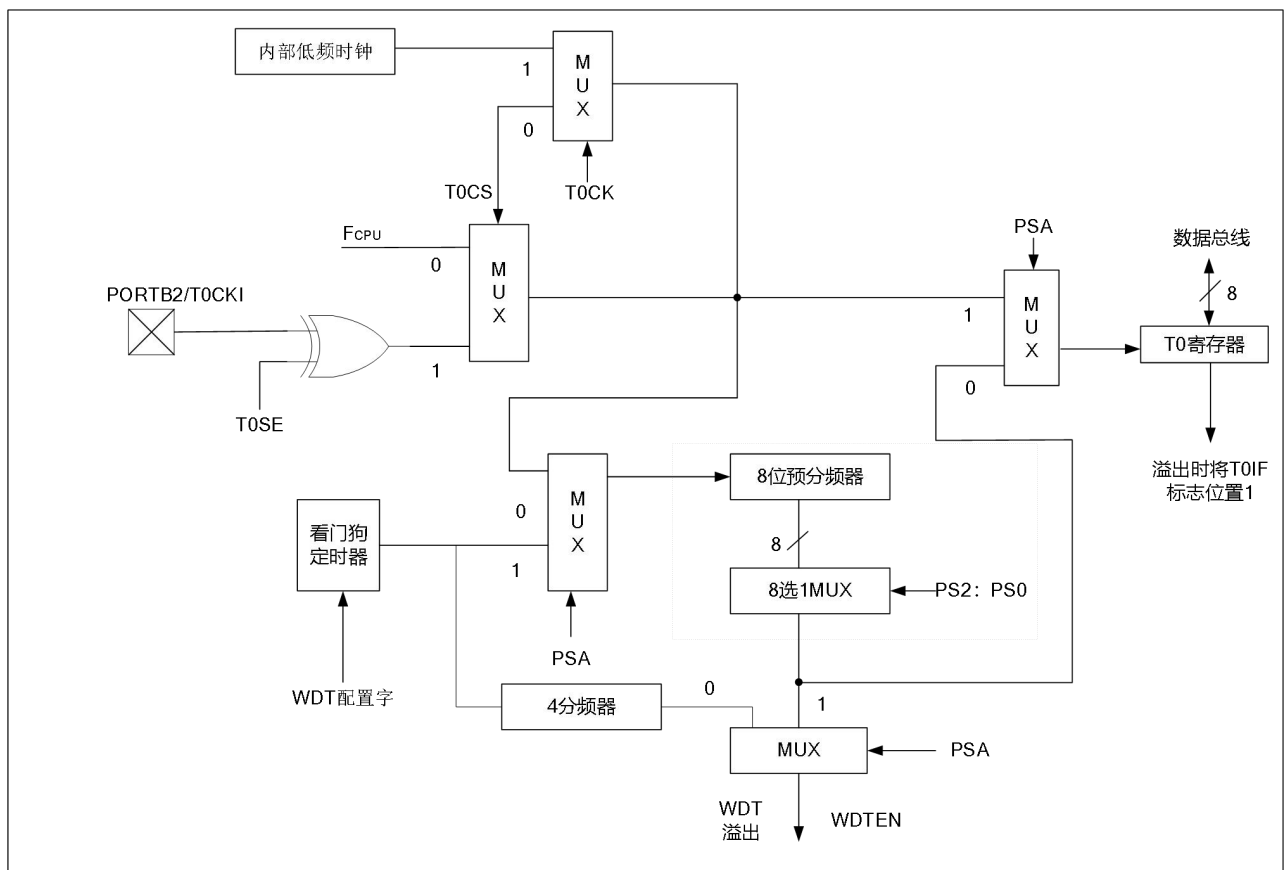
定时器 Timer0 由 8 位计数器 T0，控制寄存器 OPTION 组成。

T0 的计数时钟来自系统时钟 F_{CPU} 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： $T0 = \text{分频数} / F_{CPU}$ 。

Timer0 模块和预分频器（与 WDT 共享）框图



OPTION 寄存器

4lh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	WDTEN	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR 的值	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 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, 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

T0CR寄存器

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

Bit[7] **T0CK**: T0 时钟选择

1 = T0 以内部低频时钟作为计数时钟

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。

1. 注: 关于 Timer0 的时钟源选择, 需注意
2. 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 $F_{cpu} = 4\text{MHz}/4 = 1\text{MHz}$; 系统选择高频时钟 4MHz, 2T 模式, 则 $F_{cpu} = 4\text{MHz}/2 = 2\text{MHz}$ 。
3. 2、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。
4. 3、Timer0 配置为 RTC 模式时, T0 以外部低频振荡器作为计数时钟。

➤ 例: 利用Timer0定时器产生1ms定时, 配置字选择2MHz系统时钟, 2T模式。

...

	BCF	T1CR_AUXR,T0CK	;T0计数时钟由T0CS决定
	MOVLW	0X11	
	MOVWF	OPTION	;T0CS = 0,PSA=0,PS0=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	INTECON,T0IF	
	BSF	INTECON,T0IE	;使能Timer0中断
	BSF	INTECON,GIE	;使能中断功能
	...		
ISR_T0			;Timer0 中断处理程序
	BCF	INTECON,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	

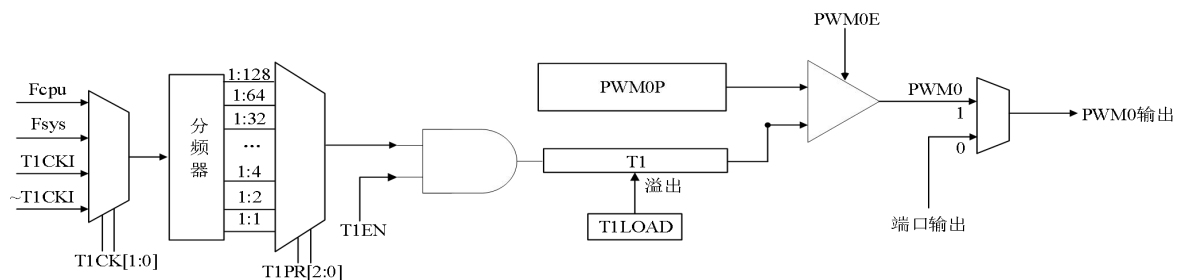
8.3 Timer1 定时/计数器

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）
- 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。

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 PWM0E=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))

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	-	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，PORTB2 端口作为 I/O 口

1: 允许 PWM0 输出，PORTB2 端口输出 PWM0 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Fosc=8M，Fsys=8M，Fcpu=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。

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

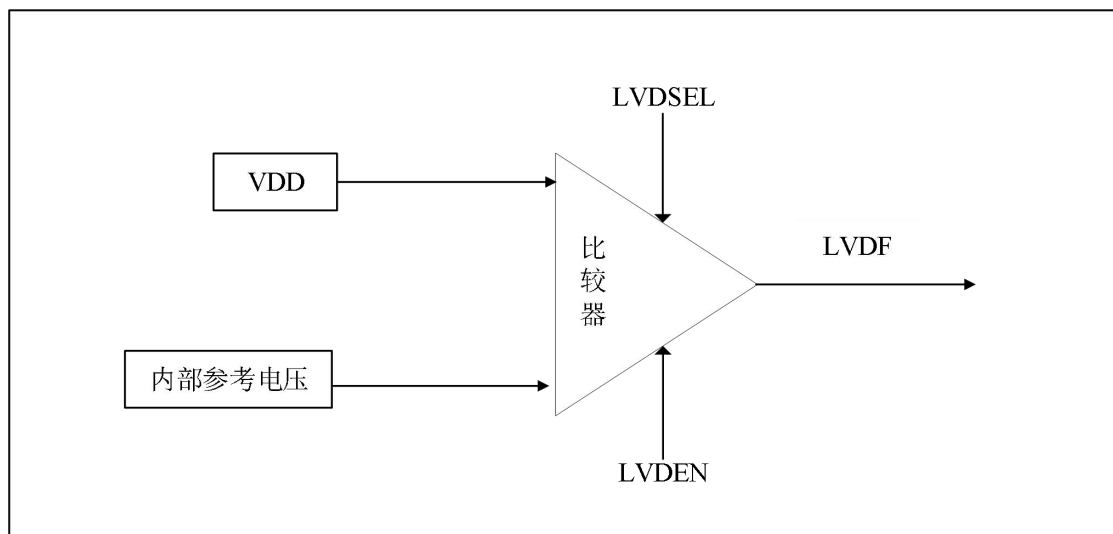
Bit [7:0] 用于设置 PWM0 的高电平时间

注意:

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

9 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 选择电压检测测量值。

当 VDD 电压低于电压检测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 电压高于电压检测测量值

1: VDD 电压低于电压检测测量值

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] **LV DEN**: LVD 使能位

1: 开启 LVD

0: 关闭 LVD

10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
	CLRWD T -	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
	NOP -	空操作	-	-	-	1
	SLEEP -	进入待机模式, 影响 TO, PD 位	-	-	-	1

11 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）				
			FCPU = 0~4MHz	2.0	—	5.5	
			FCPU = 0~2MHz	1.5	—	5.5	
			FCPU = 0~1MHz	1.5	—	5.5	
			FCPU = 0~455KHz	1.5	—	5.5	
			FCPU = 0~60KHz	1.5	—	5.5	
IDD1	工作电流	3V	FCPU = 8MHz, 2T, WDT 禁止, 无负载	—	1.0	—	mA
		5V		—	2.0	—	mA
IDD2	工作电流	3V	FCPU = 4MHz, 2T WDT 禁止, 无负载	—	0.8	—	mA
		5V		—	1.5	—	mA
IDD3	工作电流	3V	FCPU = 2MHz, 2T WDT 禁止, 无负载	—	0.5	—	mA
		5V		—	0.9	—	mA
IDD4	工作电流	3V	FCPU = 60KHz, 2T, WDT 禁止, 无负载	—	7	—	μA
		5V		—	10	—	μA
IDD5	工作电流	—	VDD<BOR 档位	—	—	1	μA
Isb1	静态电流	3V	休眠模式, WDT 使能, 无负载	—	5	—	μA
		5V		—	15	—	μA
Isb2	静态电流	3V	休眠模式, WDT 禁止, 无负载	—	—	1	μA
		5V		—	—	1	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		0.3VDD		V
VIH1	输入高电平	5V			0.3VDD		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	19	21	24	mA
IOH1	输出拉电流	5V	输出口, Vout=VDD-0.5V	13	15	18	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
VBOR	低电压复位	—	-	-0.15v	所选 BOR	+0.15v	V
LVD	低电压检测	—	-	-0.15v	所选 LVD	+0.15v	V
VPOR	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 RC8M 启动时间	Tset1	常温, VDD=5V	5	6	8	μs
内部 RC32K 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	8(1-2%)	8	8(1+2 %)	MHz
内部高频 RC 频率精度	FIRC2	VDD=5.0V, -40°C~+85°C	8(1-3.5%)	8	8(1+3.5%)	MHz
内部低频 RC 频率精度	FWRC1	VDD=1.8V~5.5V, 25°C	-50%	60	+50%	KHz
内部低频 RC 频率精度	FWRC2	VDD=5.0V, -40°C~+85°C	-50%	60	+50%	KHz

◆ 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 CMP	T=-40°C~85°C	2.0	-	5.5	V
工作电流	I CMP		-	10		μA
输入失调电压	Voffset			10	20	mV
输入共模电压	Vcom		0	-	VDD-0.7	V
输出回滞电压	Vhys		5	12	30	mV
响应时间	T RESP			100	500	ns
比较器模式改变稳定时间	CPmc		-15%	2.5	7.5	us
比较器电流消耗	CPcs	VDD=3.3.V		20		ua

12 开发工具

12.1 OTP 烧录器（HC-PM18 4.0）

- PM18 4.0：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。

12.2 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。

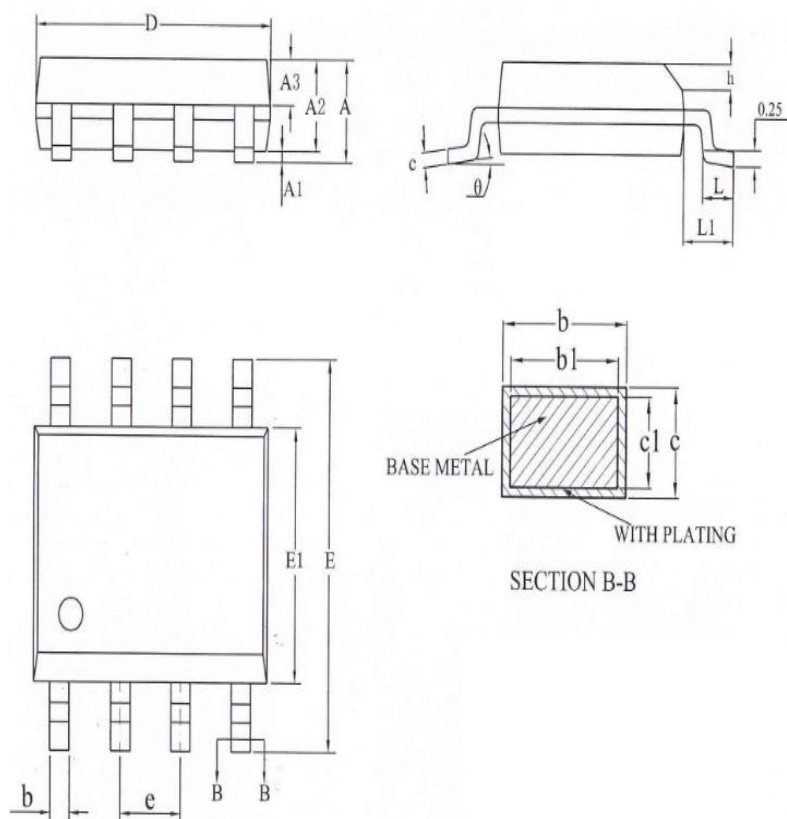
- HC-IDE：V2.10
- HC-IDE：HC-IDE V3.0.x.x(支持汇编/C 语言)

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/>

13封装信息

13.1 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°

14 数据手册版本修正记录

版本	日期	描述
Ver1.00	2022-10-25	初版
Ver1.01	2023-06-09	规格书删掉16M时钟
Ver1.02	2023-06-30	增加VPP和VDD避免同时上电的注意事项；增加Bor误复位注意事项
Ver1.03	2023-08-01	删除复位脉冲时间相关描述
Ver1.04	2023-09-11	增加eft使用注意事项

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

2022 年 10 月