

BL08P122

数据手册

16引脚8位
IO型OTP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.....	8
2 中央处理器（CPU）.....	9
2.1 存储器.....	9
2.1.1 程序存储器（OTP ROM）.....	9
2.1.2 通用数据存储器（RAM）.....	12
2.1.3 特殊功能寄存器（SFR）.....	13
2.1.4 芯片配置选择.....	16
2.2 寻址模式.....	17
2.2.1 立即寻址.....	17
2.2.2 直接寻址.....	17
2.2.3 间接寻址.....	17
2.2.4 堆栈.....	18
3 系统时钟.....	19
3.1 概述.....	19
3.2 系统高频时钟.....	20
3.2.1 内部高频 RC.....	20
3.2.2 外部高频晶体振荡器.....	20
3.3 系统低频时钟.....	21
3.3.1 内部低频 RC 振荡器.....	21
3.3.2 外部低频晶体振荡器.....	21
4 复位.....	22
4.1 概述.....	22
4.2 上电复位.....	23
4.3 WDT 复位.....	24
4.4 欠压复位.....	24
4.4.1 欠压复位的产生.....	24
4.4.2 工作死区.....	25
4.4.3 工作死区与工作频率的关系.....	25
4.4.4 死区防护.....	25
5 系统工作模式.....	26
5.1 概述.....	26
5.2 高频模式.....	28
5.3 低频模式.....	28

5.4 绿色模式.....	28
5.5 休眠模式.....	29
5.6 不同时钟源下模式选择.....	29
5.7 唤醒时间.....	30
6 中断.....	31
6.1 概述.....	31
6.2 中断请求和标志寄存器.....	31
6.3 GIE 全局中断.....	33
6.4 中断保护.....	33
6.5 INT0/1 中断.....	33
6.6 TIMER0 中断.....	34
6.7 端口电平变化中断.....	34
6.8 TIMER1/2 中断.....	34
6.9 LVD 中断.....	35
7 I/O 端口.....	36
7.1 I/O 端口模式.....	36
7.2 I/O 上拉模式.....	37
7.3 I/O 下拉模式.....	37
7.4 I/O 端口数据寄存器.....	38
8 定时器.....	39
8.1 看门狗定时器.....	39
8.2 TIMER0 定时/计数器.....	40
8.3 TIMER1 定时/计数器.....	42
8.3.1 功能概述.....	42
8.3.2 T1 使用操作说明.....	42
8.3.3 T1 相关寄存器.....	43
8.4 TIMER2 定时/计数器.....	45
8.4.1 功能概述.....	45
8.4.2 T2 使用操作说明.....	45
8.4.3 T2 相关寄存器.....	46
8.5 TIMER1 和 TIMER2 互补带死区的 PWM 软件实现方式.....	49
9 低压检测 (LVD).....	50
9.1 LVD 控制寄存器.....	50
9.2 LVD 使用.....	51
10 指令表.....	54
11 电气特性.....	55
12 开发工具.....	57
12.1 OTP 烧录器.....	57
12.2 BL-IDE.....	57

13 封装信息	58
13.1 SOP16.....	58
13.2 SOP14.....	59
13.3 SOP8.....	60

1 产品简介

BL08P122 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 位一次性编程 ROM(OTP-ROM)，128×8 位的数据寄存器 (RAM)，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、LVD 可以检测 VDD 的 1/4 电压值。

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: 2K*14位
 - 数据存储器 (RAM) 空间: 128*8位
- ◆ **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: PORTB<0>, PORTB<1>可设置触发边沿
- ◆ **BOR**
 - 1.8V/2.0V/2.7V/3.6V
- ◆ **LVD**
 - LVD检测VDD的1/4电压值
- ◆ **中断**
 - 定时器中断: Timer0和Timer1、Timer2
 - INT0、INT1外部中断、LVD中断
 - 端口电平变化中断
- ◆ **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
- ◆ **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
 - SOP16
 - SOP14
 - SOP8
 - ◆ 最低工作电压
 - 2.0V~5.5V (Fsys= 4MHz)
 - 2.0V~5.5V (Fsys= 8MHz)

1.2 引脚图

PORTA4	1	16	PORTA3
PORTA6	2	15	PORTA2/PCK/VREF/TM1PWM
PGD/TM2PWM/PORTA1	3	14	PORTB7/OSCO
PGC/TM1PWM/PORTA0	4	13	PORTB6/OSCI
VDD	5	12	GND
T2CKI/TM2PWM/PORTA7	6	11	PORTB2/T0CKI/TM1PWM
VPP/PORTB5	7	10	PORTB1/INT1/TM2PWM
TM2PWM/PORTB3	8	9	PORTB0/INT0/TM1PWM/T1CKI

PORTA6	1	14	PORTA2/PCK/VREF/TM1PWM
PGD/TM2PWM/PORTA1	2	13	PORTB7/OSCO
PGC/TM1PWM/PORTA0	3	12	PORTB6/OSCI
VDD	4	11	GND
T2CKI/TM2PWM/PORTA7	5	10	PORTB2/T0CKI/TM1PWM
VPP/PORTB5	6	9	PORTB1/INT1/TM2PWM
TM2PWM/PORTB3	7	8	PORTB0/INT0/TM1PWM/T1CKI

VDD	1	8	GND
T2CKI/TM2PWM/PORTA7	2	7	PGC/TM1PWM/PORTA0
TM2PWM/PORTB3	3	6	PGD/TM2PWM/PORTA1
VPP/PORTB5	4	5	PCK/VREF/TM1PWM/PORTA2

1.3引脚描述

脚位	名称	类型	说明
1	PORTA4	I/O	输入/输出口，带可编程上拉电阻，端口电平变化中断。
2	PORTA6	I/O	输入/输出口，带可编程上拉电阻，端口电平变化中断。
3	PORTA1 PGD TM2PWM	I/O I/O 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。 Timer2 PWM输出口。
4	PORTA0 PGC TM1PWM	I/O I/O 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。 Timer1 PWM输出口。
5	VDD	P	电源
6	PORTA7 T2CKI TM2PWM	I/O I O	输入/输出口，带可编程上拉电阻，端口电平变化中断。 T2时钟输入。 Timer2 PWM输出口。
7	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。
8	PORTB3 TM2PWM	I/O AN	输入/输出口，带可编程上下拉电阻，端口电平变化中断。 Timer2 PWM输出口。
9	PORTB0 INT0 T1CKI TM1PWM	I/O I O 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0输入 T1时钟输入。 Timer1 PWM输出口
10	PORTB1 INT1 TM2PWM	I/O I O	输入/输出口，带可编程上拉电阻，端口电平变化中断。 外部中断1输入 Timer2 PWM输出口。
11	PORTB2 TM1PWM T0CKI	I/O O 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。 T0时钟输入
12	GND	P	地。
13	PORTB6 OSCO	I/O AN	输入/输出口，带可编程上下拉电阻，端口电平变化中断。 高/低频晶体振荡器输出口。
14	PORTB7 OSCI	I/O 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。
15	PORTA2 PCK VREF TM1PWM	I/O O P O	输入/输出口，带可编程上拉电阻，端口电平变化中断。 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。
16	PORTA3	I/O	输入/输出口，带可编程上拉电阻，端口电平变化中断。

注: I = 输入 O = 输出 I/O = 输入/ 输出 P = 电源

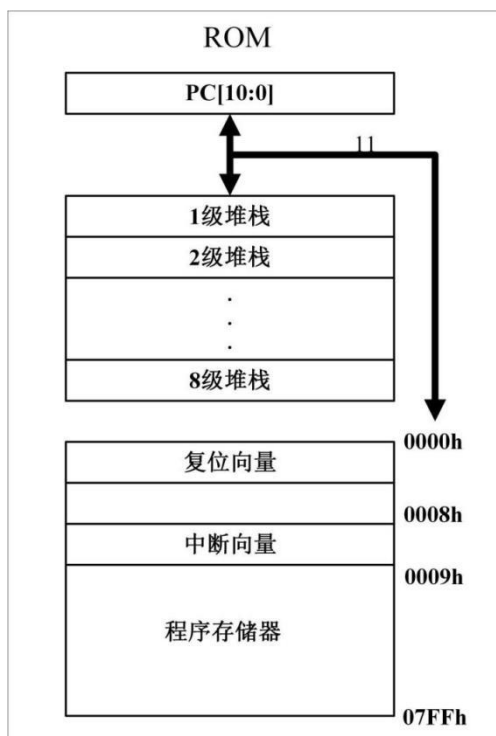
2 中央处理器（CPU）

2.1 存储器

2.1.1 程序存储器（OTP ROM）

BL08P122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 复位向量（0000H）

- 上电复位（POR=1，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1，判定为上电复位
    BTFSS    T1CR_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      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:  CLRWDTR          ;TO标志置1,WDT复位处理程序
    ...                          ;其他程序，注意处理BANK

```

2.1.1.2 中断向量（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。

2.1.1.3 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 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
...
ORG     100H
TAB1:
    ADDWF    PCL,F        ;表头运算
    RETLW    DATA0_TAB1  ;W=0对应数据
    RETLW    DATA1_TAB1  ;W=1对应数据
    RETLW    DATA2_TAB1  ;W=2对应数据
    ...
    RETLW    DATAFE_TAB1 ;W=0XFE对应数据

```

对于数据查表的编程，需注意：

1. 数据表宽度：8 位
2. 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。

➤ 例：跳转表

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

...

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

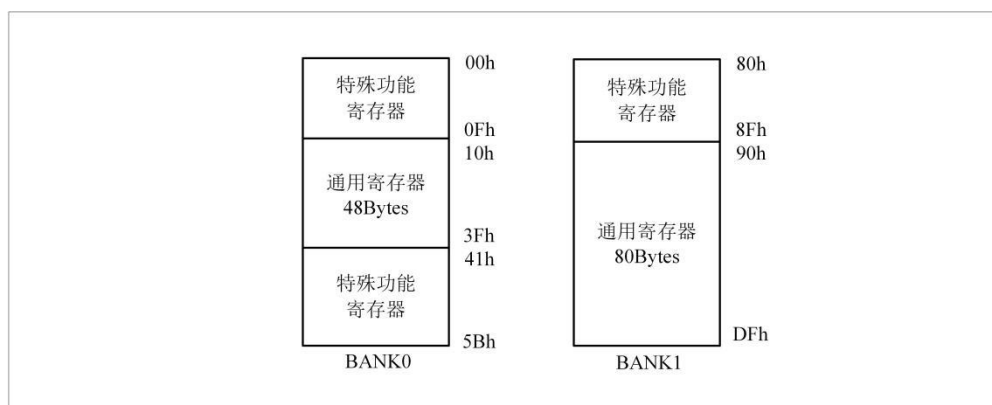
注：

如上跳转表，有 4 个跳转分支，TABBUF 的合法范围为 0X00~0X03

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	TOIF	INT0E	PBIE	TOIE

2.13 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		PORTB3	PORTB2	PORTB1	PORTB0
07h	IOCA	IOCA7	IOCA6		IOCA4	IOCA3	IOCA2	IOCA1	IOCA0
08h	-	-	-	-	-	-	-	-	-
09h	IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	IOCB0
0Ah	PCLATH	-	-	-	-	-	程序计数器高 3 位缓存器		
0Bh	PDCON	PDB7	PDB6	PDB5	PDB4	PDB3	PDB2	PDB1	PDB0
0Ch	-	-	-	-	-	-	-	-	-
0Dh	PBLON	PHB7	PHB6	PHB5	PHB4	PHB3	PHB2	PHB1	PHB0
0Eh	INTECON	GIE	-	INT0F	PBIF	T0IF	INT0E	PBIE	T0IE
0Fh	PIR	-	INT1EDG	INT1E	INT1F	LVDIE	LVDIF	PAIE	PAIF
41h	OPTION	WDTEN	INT0EDG	T0CS	T0SE	PSA	PS2	PS1	PS0
42h	PDCON1	PDA7	PDA6		PDA4	PDA3	PDA2	PDA1	PDA0
43h	PBLON1	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
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	NSELL	NSEL7	NSEL6	-	NSEL4	NSEL3	NSEL2	NSEL1	NSEL0
57h	NSELH	NSEL15	NSEL14	-	-	NSEL11	NSEL10	NSEL9	NSEL8
58h	LVDRESL	LVD 结果寄存器低字节				保留位			
59h	LVDRESH	LVD 结果寄存器高字节							
5Ah	LVDCON	-	-	CHS3	CHS2	CHS1	CHS0	LVDON	LVDEN
5Bh	LVDCON1	-	-	-	-	VHS2	VHS1	VHS0	-

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-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

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

BL08P122 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

BL08P122 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

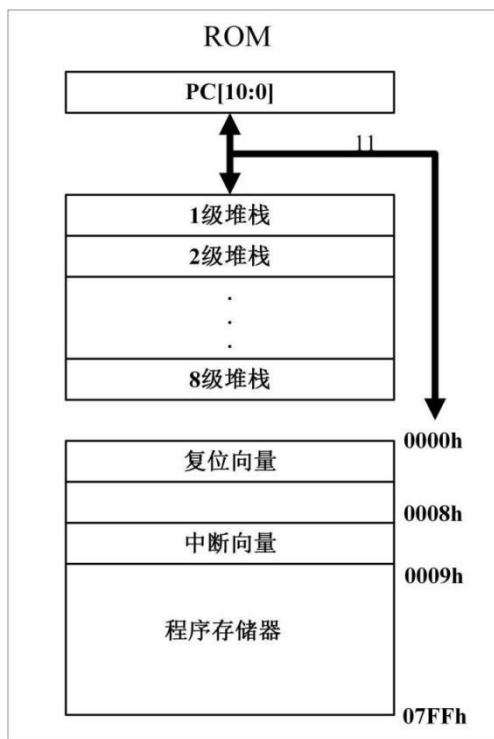
BTFSS Z

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

BL08P122 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。

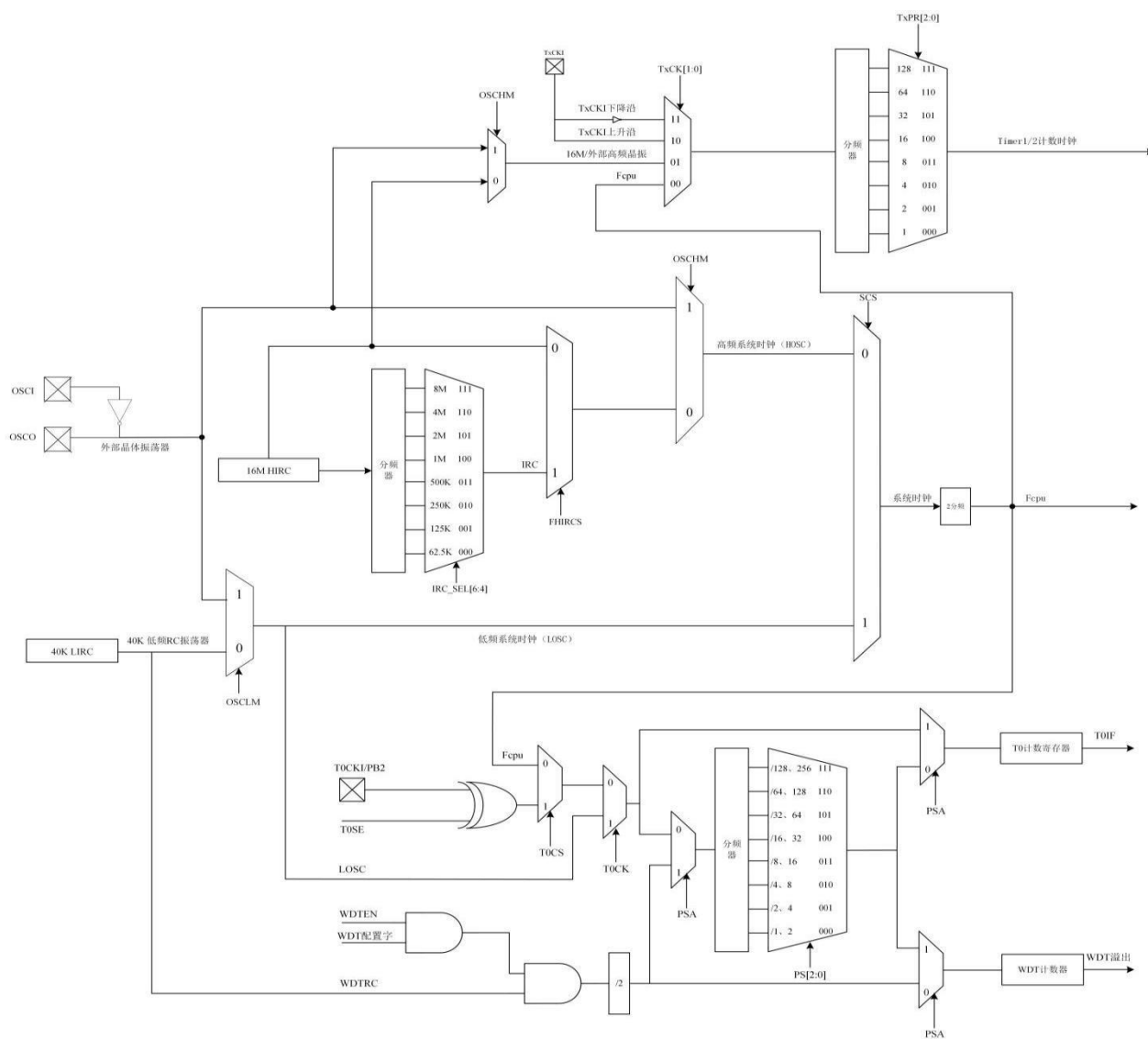


注：

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



注: OSCHM和OSCLM不能同时为1

BL08P122 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: $F_{cpu} = F_{sys} / 2$;

低频模式: $F_{cpu} = F_{sys} / 2$;

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

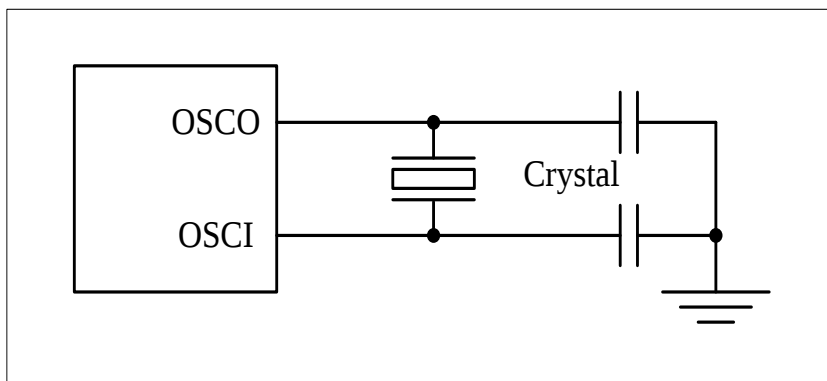
IRC_SEL[6:4]	说明
111	内部RC振荡器频率选择8MHz
110	内部RC振荡器频率选择4MHz
101	内部RC振荡器频率选择2MHz
100	内部RC振荡器频率选择1MHz
011	内部RC振荡器频率选择500KHz
010	内部RC振荡器频率选择250KHz
001	内部RC振荡器频率选择125KHz
000	内部RC振荡器频率选择62.5KHz

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

BL08P122 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）

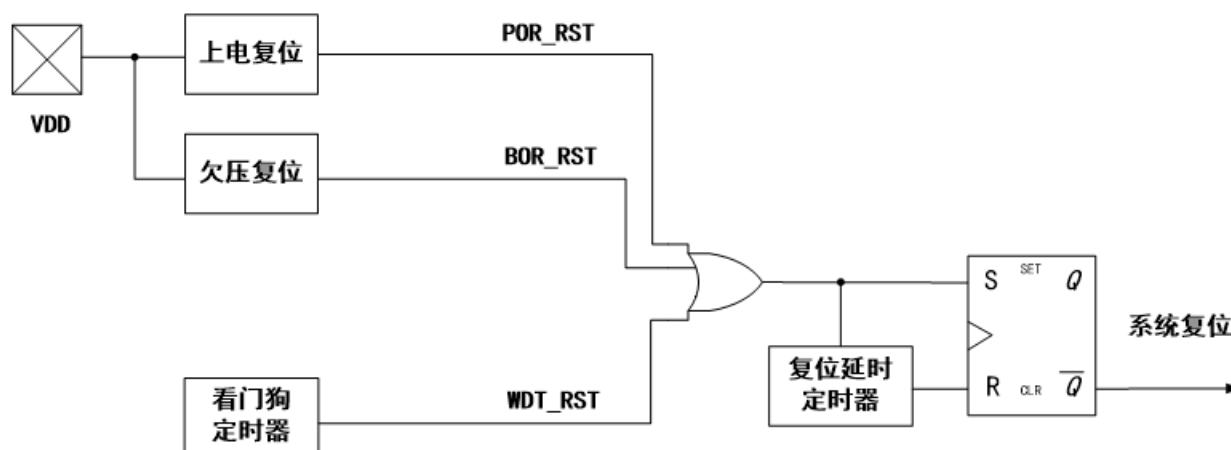
当上述任何一种复位产生时，系统进入复位状态，所有的特殊功能寄存器被初始化，程序停止运行，同时程序计数器（PC）清零。经过上电延时定时器延时后，系统结束复位状态，程序从 000h 地址开始执行。STATUS 寄存器的 Bit4（TO 位）及 T1CR_AUXR 寄存器的 Bit3（BOR 位）、Bit4（POR 位）显示系统复位状态信息，可通过这 3 个标志位判断复位来源，从而控制系统的运行路径。

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

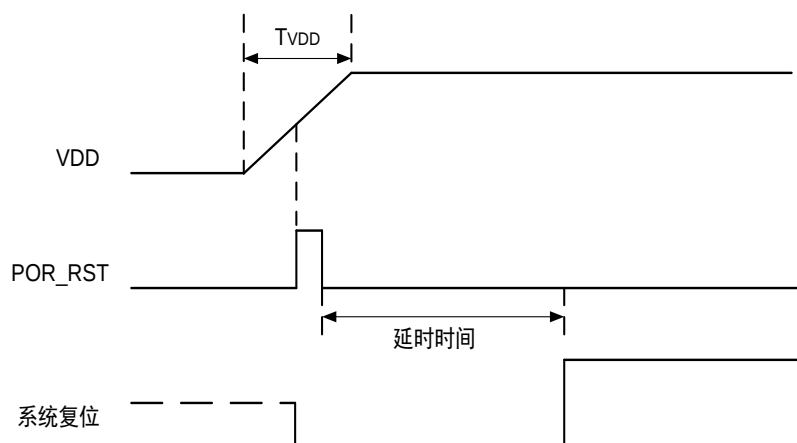
4.2 上电复位

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：



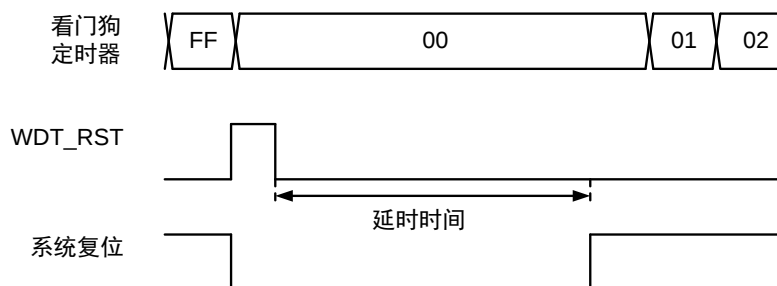
关于上电复位，请注意以下几点：

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：

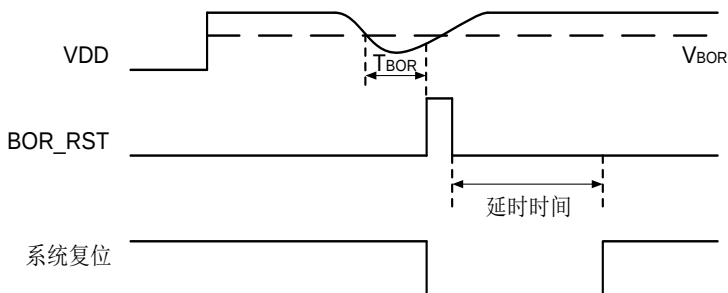
1. 主程序中有一次清看门狗的动作，这种架构能够最大限度的发挥看门狗的保护功能。看门狗的使能逻辑：看门狗使能 = 看门狗配置字使能 & 看门狗软件使能（WDTEN=1）。
2. 不建议在中断程序中对看门狗进行清零，否则无法监控主程序跑飞情况。

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：

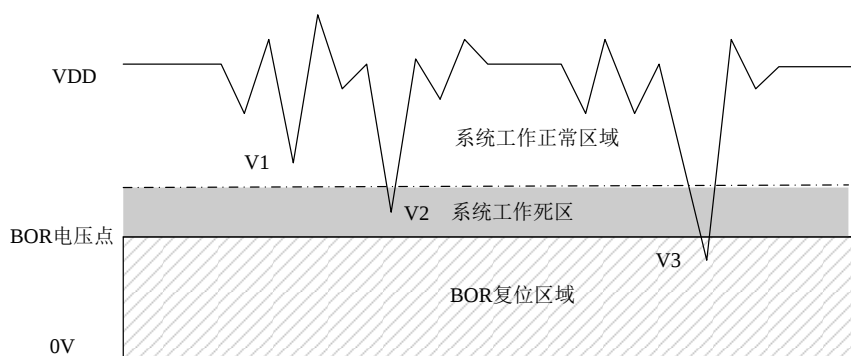


注： T_{BOR} 需大于 200ns，否则电压跌落时可能不产生欠压复位信号。

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。

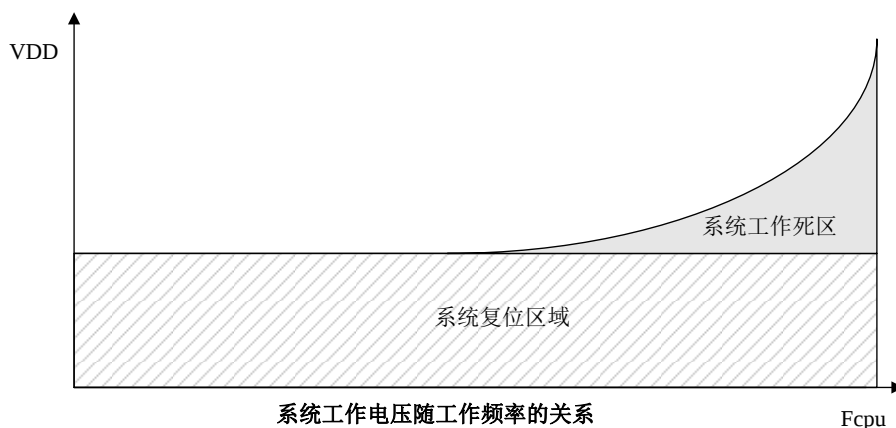
4.4.2 工作死区

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

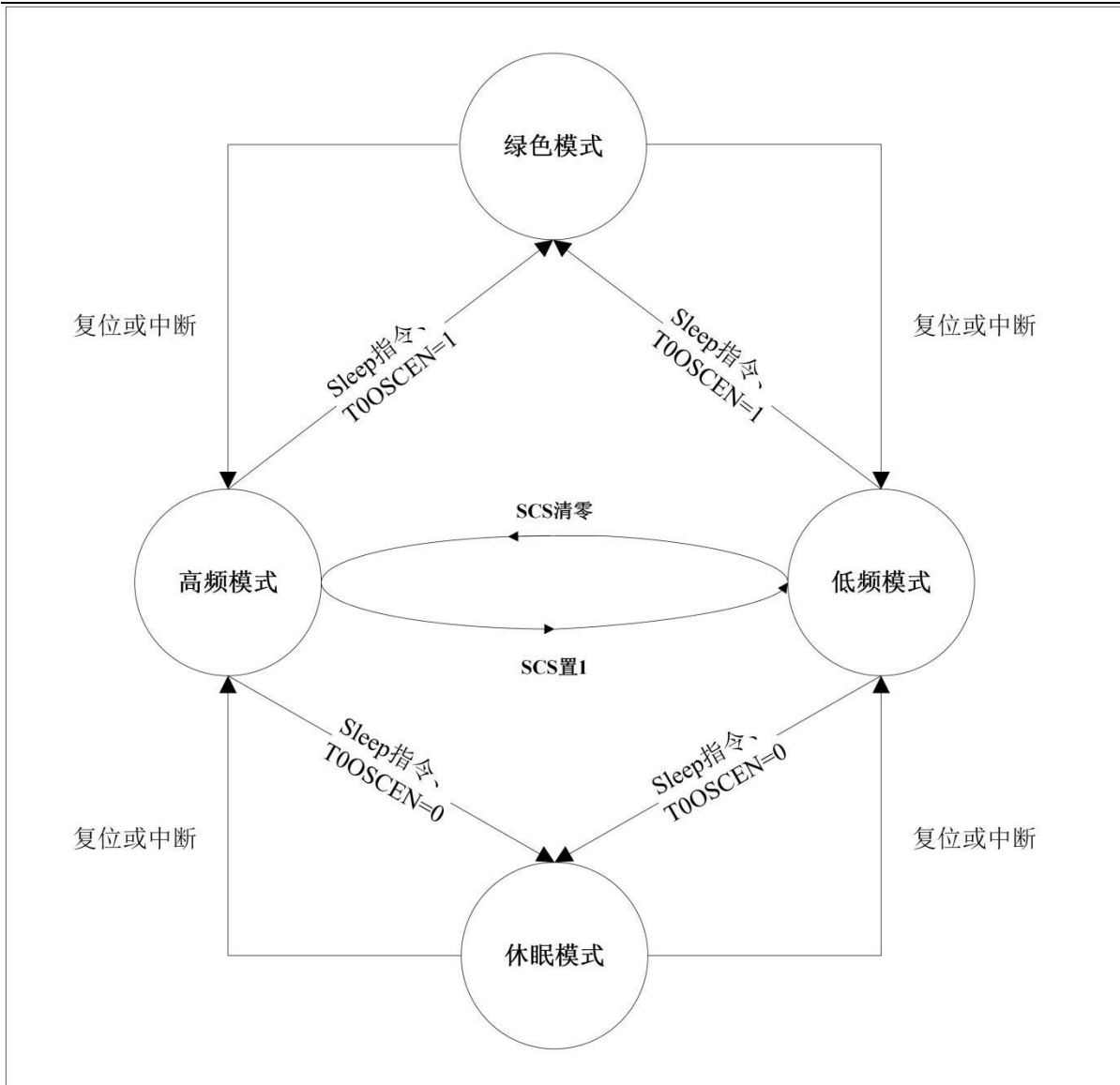
5 系统工作模式

5.1 概述

BL08P122 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：

- 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：

- 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。

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

注：在使用外部低频晶振作时钟源时，若使用端口电平变化中断做唤醒，需维持端口变化状态至少 200ms（如低电平转低高电平，需保持高电平至少 200ms）。

当进入低功耗模式（1ua）后再 sleep 唤醒后，开启 BOR（BOREN）前需要先开启 LVDEN=1 并等待 50us 左右再开启 BOR（BOREN）。

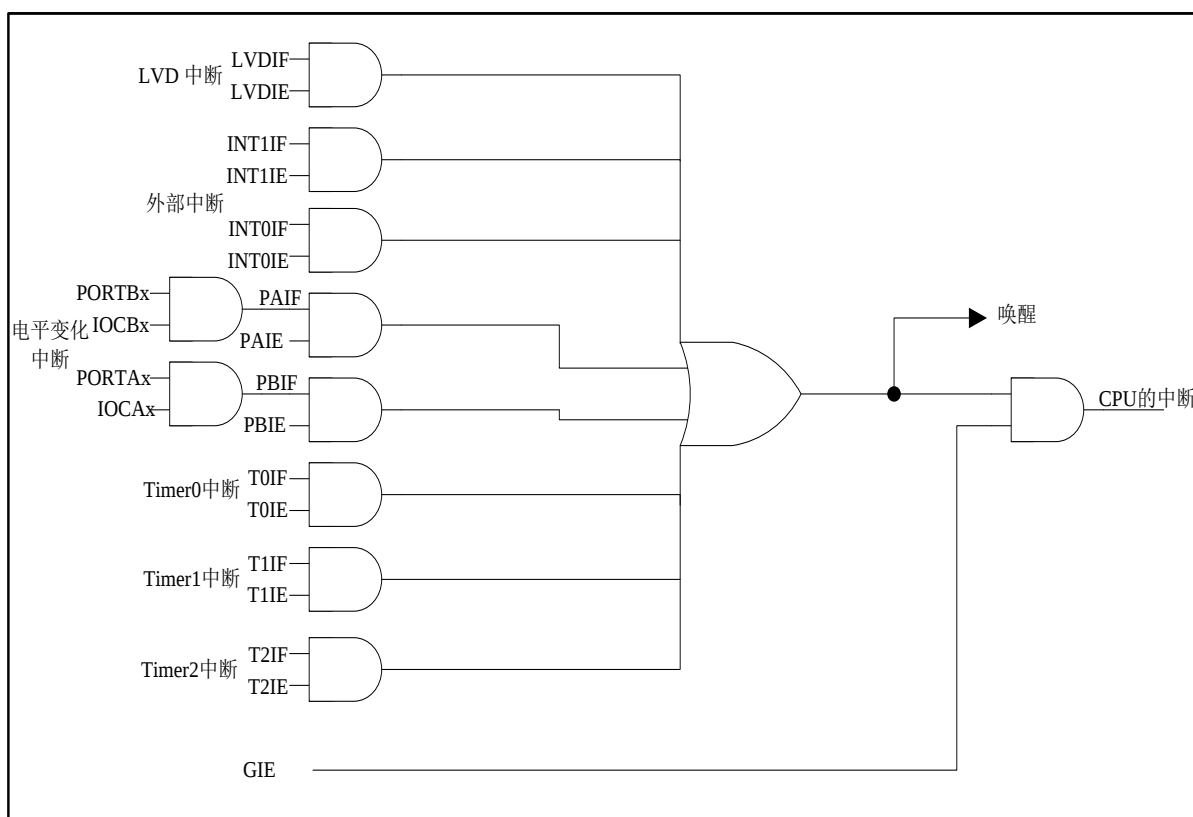
例：

```

...
BCF    T1CR_AUXR, 2    ;关闭 BOR
NOP
NOP
NOP
SLEEP                                ;休眠
NOP
NOP
NOP
...                                ;唤醒
BSF    LVDEN            ;打开 LVDEN（如果是高频模式进休眠则不需要这条语句）
CALL   Delay50us        ;延时 50us
BSF    T1CR_AUXR, 2    ;打开 BOR
...

```

6 中断



6.1 概述

BL08P122 提供 3 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、LVD 中断。

系统从高频或低频模式进入睡眠模式时，INT0/1 外部中断、端口电平变化中断、LVD 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。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	LVDIE	LVDIF	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 [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

4lh	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	INT1E	INT1F	LVDIE	LVDIF	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	IOCA6	-	IOCA4	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

6.9LVD 中断

当 LVD 完成检测 VDD, LVDON 被硬件清零, 无论 LVDIE 处于何种状态, 与此同时 LVDIF 被置“1”。若 LVDIE、LVDIF 为“1”, 且 GIE 使能, 系统就会相应 LVD 中断; 若 LVDIE = 0, 则无论 LVDIF 是否置“1”, 系统都不会响应 LVD 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	LVDIE	LVDIF	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] LVDIE: LVD中断使能位

1 = 使能 LVD 中断

0 = 禁止 LVD 中断

Bit [2] LVDIF: LVD中断标志位

1 =LVD 检测已完成 (必须由软件清 0)

0 =LVD 检测未完成或尚未开始

7 I/O端口

7.1 I/O 端口模式

PORTA 控制寄存器

056h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NSELL	NSEL7	NSEL6	-	NSEL4	NSEL3	NSEL2	NSEL1	NSEL0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR 值	1	1	-	1	1	1	1	1

NSEL[4:0]: PORTA4~PORTA0 引脚控制位

- 1: 高阻模式。
- 0: 数字模式, 作为数字输入或输出口。

NSEL[7:6]: PORTA7~PORTA6 引脚控制位

- 1: 高阻模式。
- 0: 数字模式, 作为数字输入或输出口。

PORTB 控制寄存器

057h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NSELH	NSEL15	NSEL14	-	-	NSEL11	NSEL10	NSEL9	NSEL8
R/W	R/W	R/W	-	-	R/W	R/W	R/W	R/W
POR 值	1	1	-	-	1	1	1	1

NSEL[3:0]: PORTB3~PORTB0 引脚控制位

- 1: 高阻模式。
- 0: 数字模式, 作为数字输入或输出口。

NSEL[7:6]: PORTB7~PORTB6 引脚控制位

- 1: 高阻模式。
- 0: 数字模式, 作为数字输入或输出口。

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。

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。

7.2 I/O 上拉模式

PBLON 寄存器

0Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBLON	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

PBLON1 寄存器

43h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBLON1	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

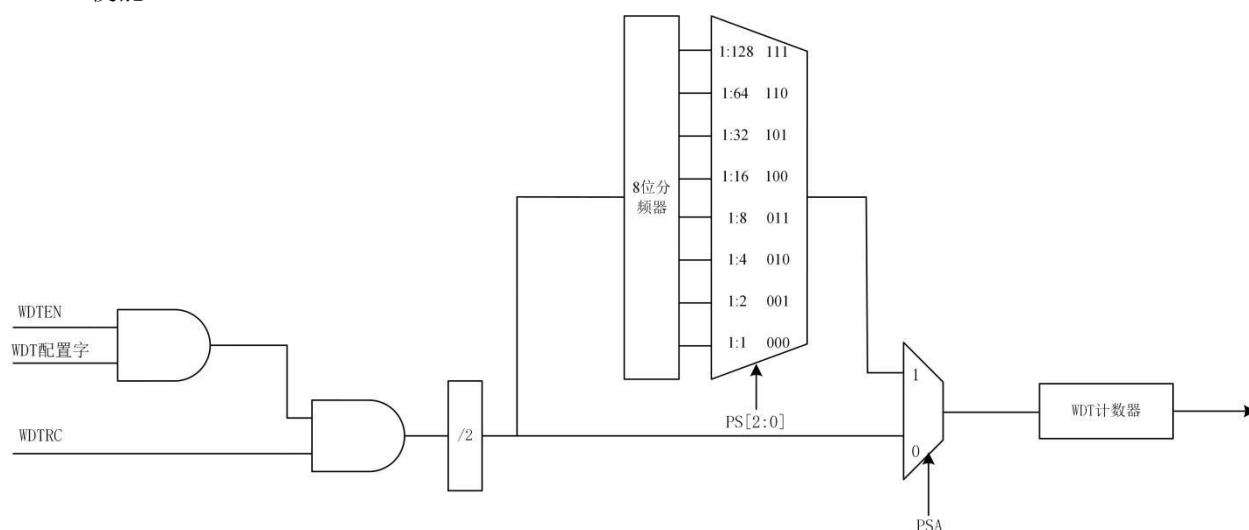
注：端口方向寄存器设为输出时，此时读端口操作为读数据寄存器的值；
端口方向寄存器设为输入时，此时读端口操作为读端口的输入电平状态。

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 WDTE 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*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	WDTE	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 WDTE 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

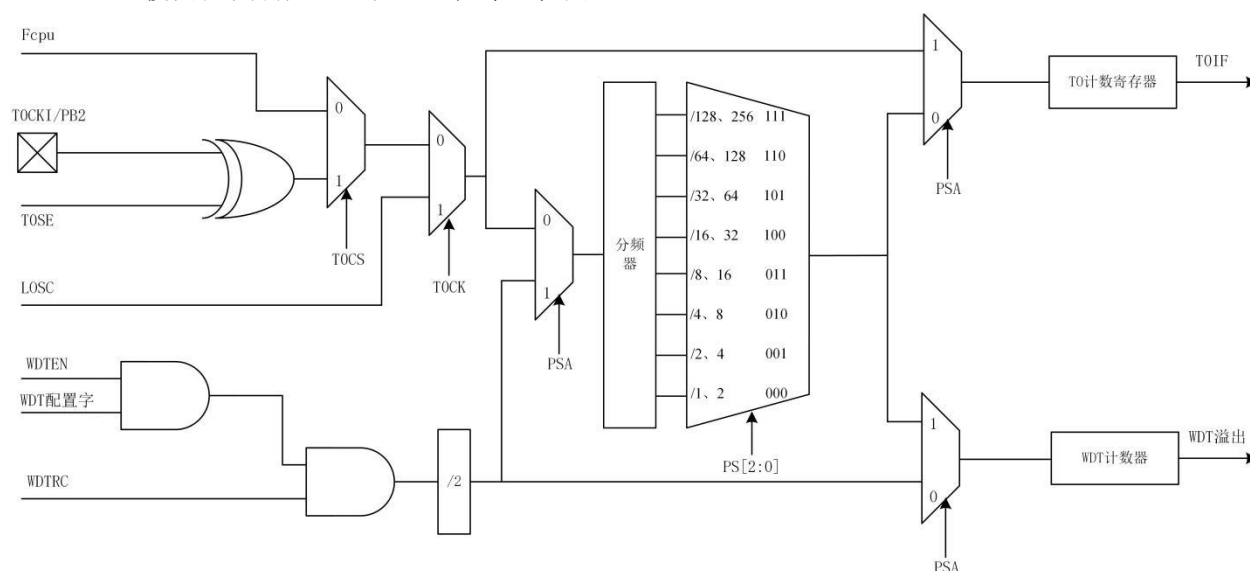
T0 的计数时钟可选择 cpu 时钟 Fcpu、外部管脚 T0CKI、系统低频时钟 LOSC。

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/ (所选时钟源频率/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

T0OSCEN	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

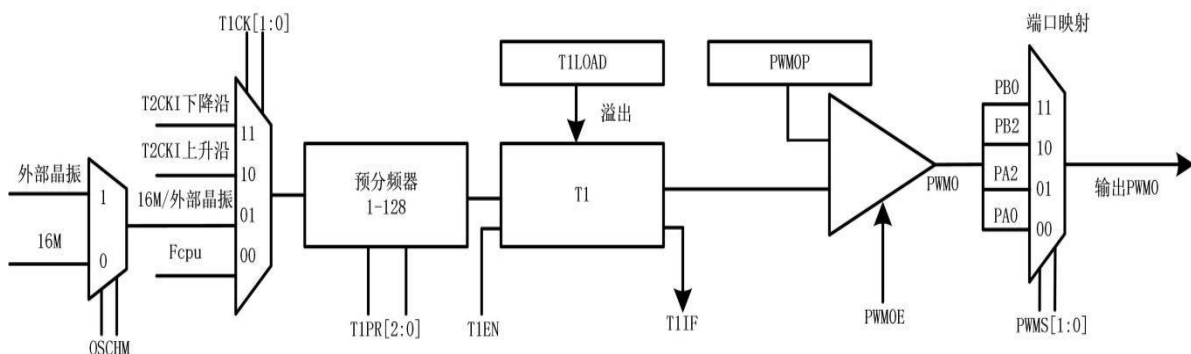
1、Timer0时钟源选择为外部时钟源T0CKI或外部低频晶振时，具有唤醒功能。

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 PWM0E=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。

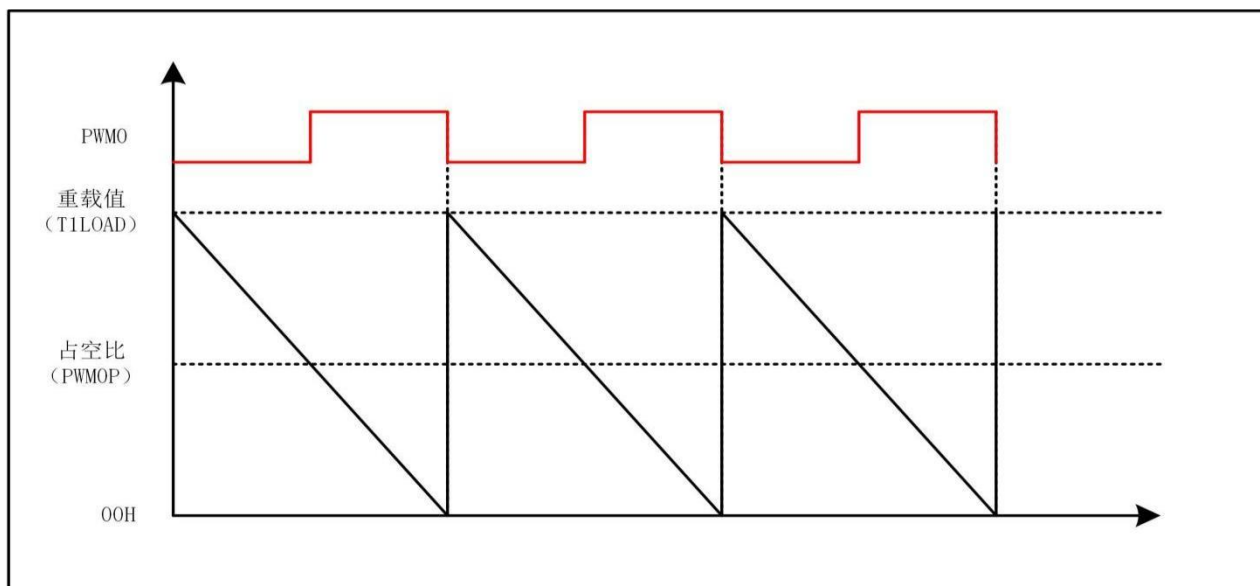
PWM0 占空比的计算如下：

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	T0OSC EN	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] T0OSCEN 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/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 TICKI 作为计数时钟时，TICKI 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
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

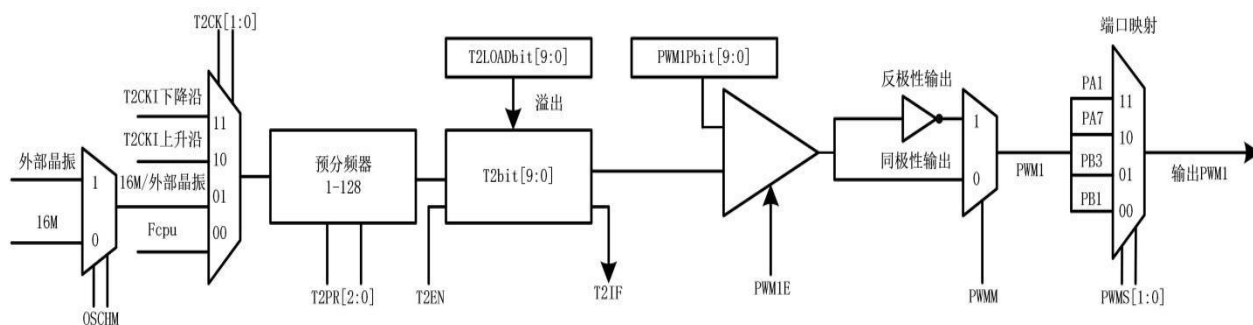
注意：

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

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：

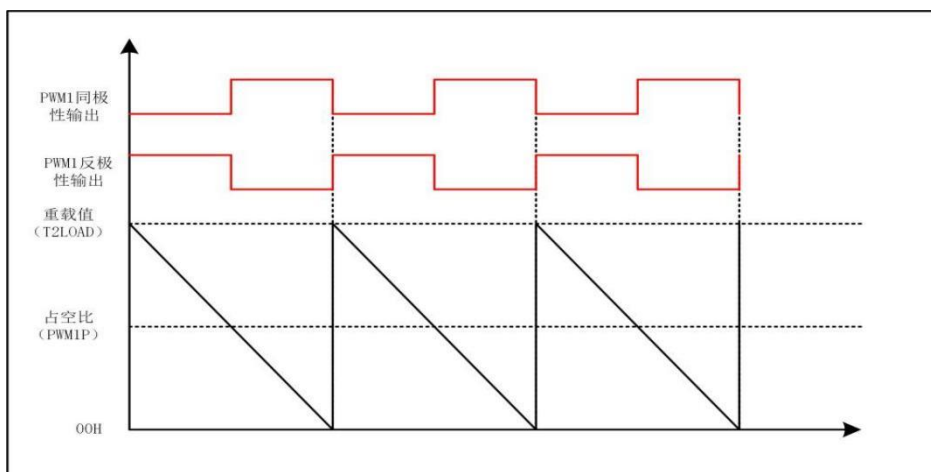
当 PWM1E=1 时，将输出 PWM 波形，当 T2 计数到与 PWM1P 相等时，PWM1 输出置 1；当 T2 计数溢出时，PWM1 输出清 0。PWM1 占空比的计算如下：

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：

1. T2CK[1:0]=01B 时,T2 时钟源为内部固定 16M 或外部晶振（最高 4M）

2. 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；

3. 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）。

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= $T_{death} - T_{cpu} = 9\mu s$ 
NOP
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

```

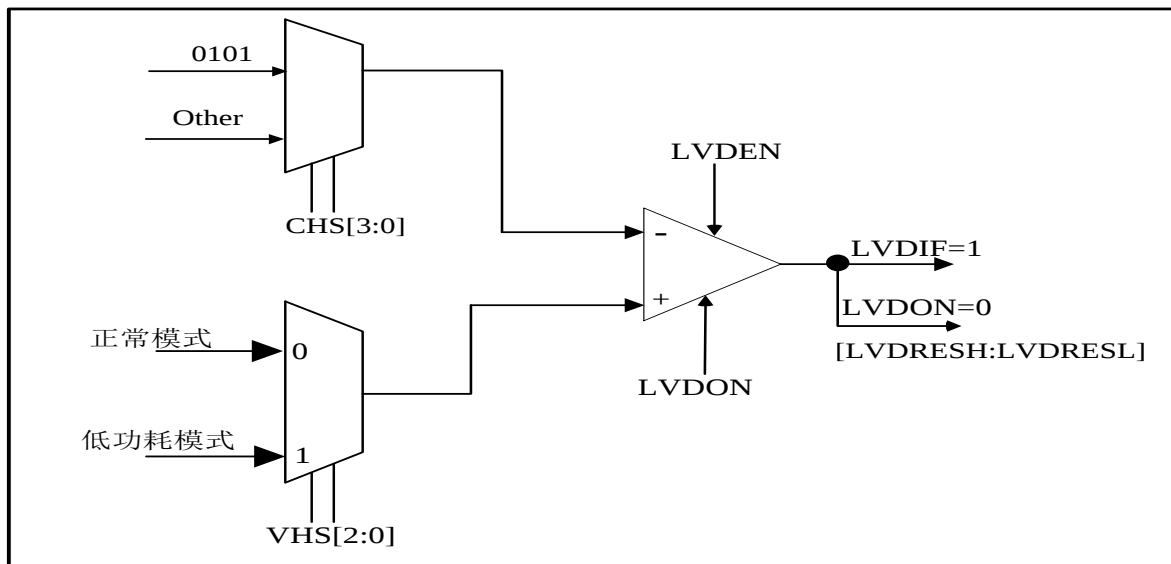
```

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 低压检测 (LVD)

BL08P122具有一个12位的检测VDD功能的LVD模块，可以直接通过读取[LVDRESH:LVDRESL]寄存器获取当前VDD/4电压值，可以通过配置LVDIE使能进入LVD中断。



9.1 LVD 控制寄存器

LVDCON寄存器

05Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LVDCON	-	-	CHS3	CHS2	CHS1	CHS0	LVDON	LVDEN
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]

0101 : 检测 VDD

其他: 休眠模式下选择此配置, 降低功耗

Bit[1]: LVDON 启动 LVD 测量使能位

1=启动一次 LVD 测试

0=LVD 测量完成后, 硬件自动清零

Bit[0]: LVDEN LVD 使能位

1=使能 LVD

0=屏蔽 LVD

LVDCON1寄存器

05Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LVDCON1	-	-	-	-	VHS2	VHS1	VHS0	-
R/W	R	R	R	R	R/W	R/W	R/W	R
POR的值	0	0	0	0	0	0	0	0

Bit[3:1]: VHS[2:0] 功耗配置位

000: 正常工作模式

001~011: 低功耗模式

100~111：保留位

注：

- LVD 使能检测 VDD 时，需要配置为正常工作模式。
- 低功耗模式影响检测精度导致检测误差较大，进休眠模式时配置成低功耗模式降低功耗

LVDRESH

059h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LVDRESH	LVD 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

LVDRESL

058h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LVDRESL	LVD 结果寄存器低字节				保留位			
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

1、LVD 精度为 12 位，高 8 位存放在 LVDRESH 寄存器中，低 4 位存放在 LVDRESL 寄存器高 4 位上。

2、[LVDRESH:LVDRESL]一次检测结果为 VDD 的 1/4。

9.2LVD 使用

1. 配置LVD模块

- 选择 LVD 检测，设置 CHS[3:0]=0101，选择检测 VDD
- 配置 LVDCON1，设置 VHS[2:0]=000
- 使能 LVD 模块，设置 LVDEN=1
- 等待 LVD 模块建立稳定时间，建议软件延时 100uS 以上

2. 配置LVD中断（不进中断可忽略此步骤）

- 清零 LVDIF 中断标志
- 使能 LVDIE 中断
- 使能全局中断

3. 启动检测

- 配置 LVDON=1，启动一次 LVD 检测

4. 通过以下方式之一等待LVD检测完成

- 查询 LVDON=0（LVD 测量完成后，硬件自动清零）
- 查询 LVDIF=1（LVD 检测已完成，由软件清 0）或等待 LVD 中断（已使能中断）

5. 读取LVD检测结果

- 查询[LVDRESH:LVDRESL]=B “xxxxxxx xxxx0000”
- 为了保证检测的稳定性，建议 3~5 步骤重复 5 次左右取平均值

注：

1、LVDRESH 数据高 8 位，LVDRESL 数据低 4 位。

2、[LVDRESH:LVDRESL]一次检测结果为 VDD 的 1/4。

6. 清零LVD中断标志（如果已使能中断则需要）

➤ 例：配置LVD检测，结果保留在[LVDRESH: LVDRESL]中。

```
LVDVH    RES    1           ;转存LVD结果高字节自定义
LVDVL    RES    1           ;转存LVD结果低字节自定义
        .....             ;其他程序

LVD_TEST:
        MOVLW    B'00010100' ;设置检测VDD
        MOVWF    LVDCON
        CLRF     LVDCON1      ;LVD正常工作模式
        BSF      LVDCON,LVDEN ;使能LVD模块
        CALL     DELAY_100us  ;延时100us，用户可自行完成
        BSF      LVDCON,LVDON ;开始一次检测

LVD_TEST_WAIT:
        BTFSC    LVDCON,LVDON ;等待检测完成
        GOTO     LVD_TEST_WAIT
        ;检测完成，保存结果
        MOVF     LVDRESH,W    ;读取LVD结果寄存器高字节
        MOVWF    LVDVH        ;转存LVD结果高字节
        MOVF     LVDRESL,W    ;读取LVD结果寄存器低字节
        MOVWF    LVDVL        ;转存LVD结果低字节
```

➤ 例：带LVD检测，进Sleep配置。

```
        .....             ;其他程序

LVD检测           ;LVD检测启动
        .....             ;其他程序

        GOTO     Sleep_Test

Sleep_Test:
        .....             ;其他程序
        BSF      VHS0       ;设置低功耗模式
        CLRF     LVDCON     ;关LVD，不检测VDD
        .....             ;其他程序
        BCF      T1CR_AUXR, 2 ;关闭 BOR
        NOP
        NOP
        NOP
        SLEEP          ;休眠
        NOP
        NOP
        NOP
        ...           ;唤醒
        BSF      LVDEN      ;打开 LVDEN（如果是高频模式进休眠则不需
        要这条语句）
        CALL     Delay50us  ;延时 50us
```

BSF T1CR_AUXR, 2 ;打开 BOR

10 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
CLRW	-	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	-
BTFSC	f,d	检测 f 中的 d 位, 为 0 则跳过	1(2)	-
BTFSS	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	-

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）				
VDD	工作电压	—	FCPU = 8MHz	2.0	-	5.5	V
			FCPU = 4MHz	2.0	-	5.5	
			FCPU = 2MHz	2.0	-	5.5	
			FCPU = 1MHz	1.8	-	5.5	
			FCPU = 500KHz	1.8	-	5.5	
			FCPU = 32KHz	1.8	-	5.5	
IDD1	工作电流	3V	FCPU = 8MHz, 高频模式	-	1.6	-	mA
		5V	WDT 禁止, 无负载	-	2.7	-	mA
IDD2	工作电流	3V	FCPU = 4MHz, 高频模式	-	0.8	-	mA
		5V	WDT 禁止, 无负载	-	1.5	-	mA
IDD3	工作电流	3V	FCPU = 2MHz, 高频模式	-	0.5	-	mA
		5V	WDT 禁止, 无负载	-	0.8	-	mA
IDD4	工作电流	3V	FCPU = 20KHz, 低频模式	-	32	-	μA
		5V	WDT 禁止, 无负载	-	85	-	μA
IDD5	工作电流	3V	FCPU = 20KHz, 绿色模式	-	19	-	μA
		5V	WDT 禁止, BOR 使能, 无负载	-	28	-	μA
Isb1	静态电流	3V	休眠模式, WDT 使能, 无负载	-	2.7	-	μA
		5V	BOR 禁止	-	7	-	μA
Isb2	静态电流	3V	休眠模式, WDT 使能, 无负载	-	19	-	μA
		5V	BOR 使能	-	28	-	μA
Isb3	静态电流	3V	休眠模式, WDT 禁止, 无负载	-	0.5	-	μA
		5V	BOR 禁止, LVD 关闭, LVD 低功耗模式	-	1	-	μA
Isb4	静态电流	3V	休眠模式, WDT 禁止, 无负载	-	16	-	μA
		5V	BOR 使能	-	22	-	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	19	22	24	mA

IOH1	输出拉电流	5V	输出口, $V_{out}=V_{DD}-0.5V$	11	16	18	mA
RPH1	内部上拉电阻	5V	可编程上拉电阻	-	120	-	k Ω
RPD	内部下拉电阻	5V	可编程下拉电阻	-	120	-	k Ω
VBOR	低电压复位	—	-	-0.1	所选 BOR	+0.1	V
VLVD	低电压检测	—	-	-0.1	检测 $V_{DD}/4$	+0.1	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 $V_{DD}=5V$ 、常温 $25^{\circ}C$ 。

◆ 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, $V_{DD}=5V$	-	-	10	μs
内部 RC40K 启动时间	Tset2	常温, $V_{DD}=5V$	-	-	150	μs
内部高频 RC 频率精度	FIRC1	$V_{DD}=1.8V\sim 5.5V, 25^{\circ}C$	16(1-1%)	16	16(1+1 %)	MHz
内部高频 RC 频率精度	FIRC2	$V_{DD}=5.0V, -40^{\circ}C \sim +85^{\circ}C$	16(1-2.5%)	16	16(1+2.5%)	MHz
内部低频 RC 频率精度	FWRC1	$V_{DD}=1.8V\sim 5.5V, 25^{\circ}C$	-50%	40	+50%	KHz
内部低频 RC 频率精度	FWRC2	$V_{DD}=5.0V, -40^{\circ}C \sim +85^{\circ}C$	-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

◆ 其他特性：

- 1、ESD (HBM) : CLASS 3A ($\geq 4000V$)
- 2、ESD (MM) : CLASS 2 ($\geq 200V$)
- 3、Latch_up: CLASS I

12 开发工具

12.1 OTP 烧录器

- PM18：支持 BL08 系列 MCU 大批量的脱机烧录。

注：

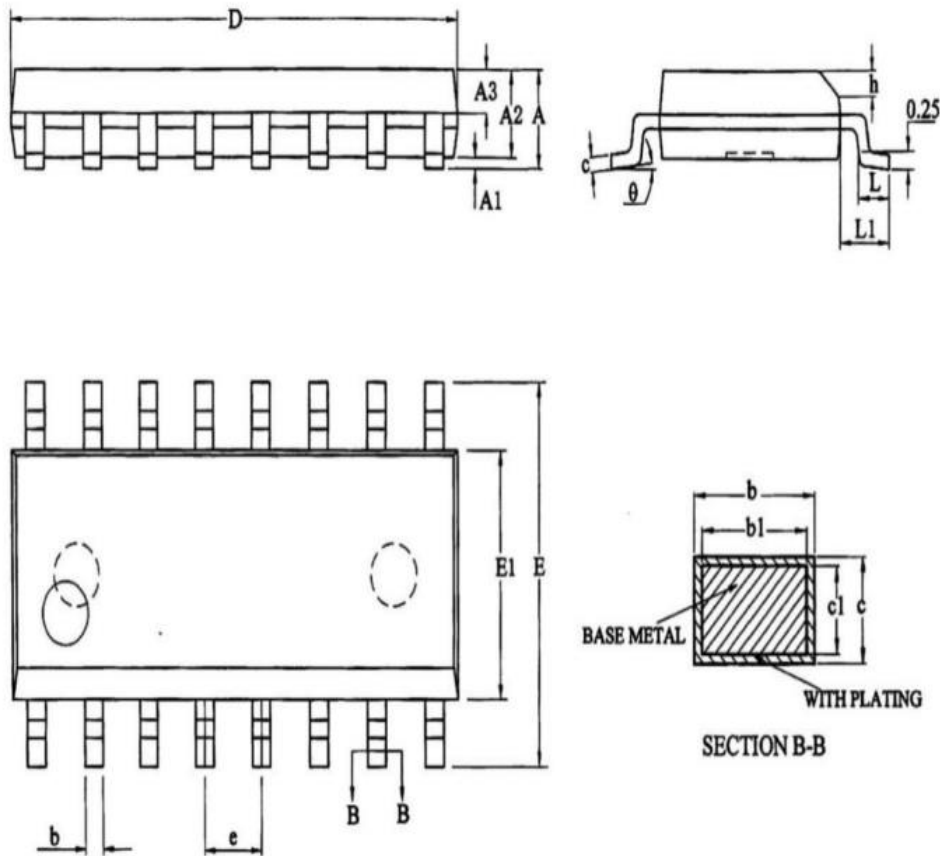
详情请参考 BL-PM18 用户手册。

12.2 BL-IDE

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

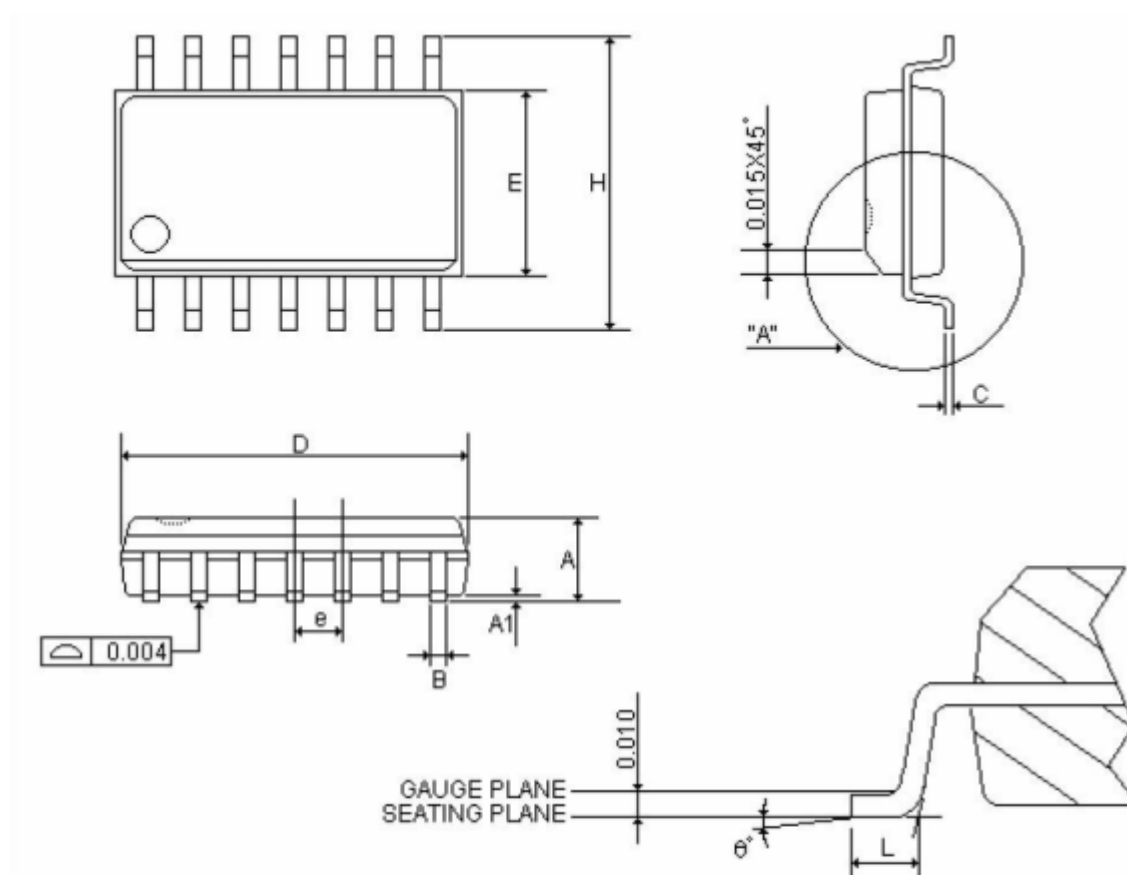
13封装信息

13.1 SOP16



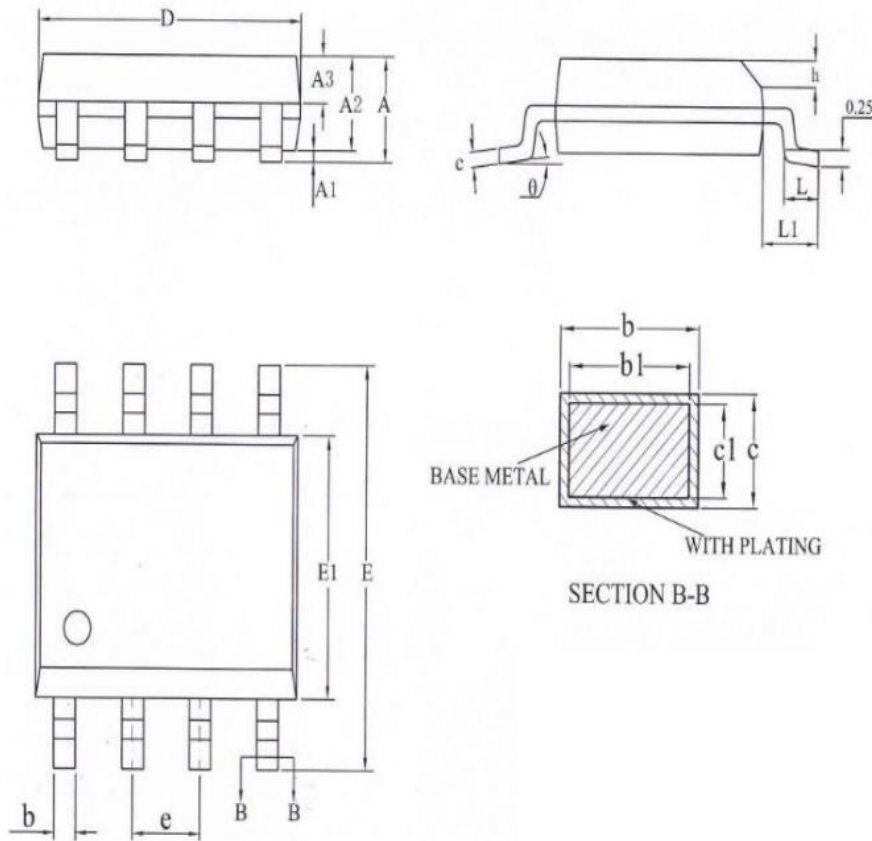
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.05	—	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	—	0.48
b1	0.38	0.41	0.43
c	0.21	—	0.26
c1	0.19	0.20	0.21
D	9.70	9.90	10.10
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	1.27BSC		
h	0.25	—	0.50
L	0.50	—	0.80
L1	1.05BSC		
θ	0	—	8°

13.2 SOP14



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

13.3 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°

BOLING 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。BOLING 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，BOLING 的产品不是专门设计来应用于外科植入、生命维持和任何 BOLING 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 BOLING 的产品用于上述领域，即使这些是由 BOLING 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 BOLING 及其雇员、子公司、分支机构和销售商与上述事宜无关。

波领科技

2024 年 04 月