

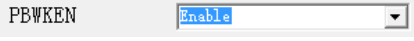
这份文档里包含下列型号

- SM152 : EPROM device (OTP)

## EPROM/ROM-Based 8-Bit Microcontroller Series

### 特色功能:

1、 B3 口可做正常 IO，能输出高低。

2、 兼容义隆选项  ，义隆转换来的程序需要打开该项配置。

## 注意事项:

- 1、中断问题：端口状态变化中断，会有几率丢失中断。
- 2、IO口 latch up 问题：当用 IO 口直接驱动共阳 RGB 七彩灯时，会有灯珠无法点亮情况。  
解决办法：增大IO到RGB灯上的电阻阻值。  
参考阻值：B0 (R)：100欧； B1 (G)：51欧； B2 (B)：10欧。
- 3、频飘问题：5V到2.5V频飘在5%左右。
- 4、建议在芯片电源地之间加一个 104 电容，否则连续上下电有概率导致芯片死机。

KONGSON

## 修正记录

版本号	日期	内容
V10	2016 年 6 月	初版
V20	2017 年 7 月	更新芯片电压频率关系，增加特色功能说明，增加SOT23-6封装脚位
V30	2017 年 9 月	增加注意事项2
V40	2017 年 10 月	增加封装信息

## 功能特性

- 只有42个单字指令
- 除分支指令为两个周期指令以外其余为单周期指令
- PC寻址范围为13-bit
- GOTO指令能跳转到所有的ROM/EPROM地址空间
- 子程序能返回到所有的ROM/EPROM地址空间
- 能处理8位数据
- 5级硬件堆栈
- 运行速度: DC-20 MHz 工作频率  
DC-100 ns 指令周期

型号	管脚#	I/O#	EPROM/ROM空间 (Byte)	RAM (Byte)
SM152	8	6	0.5K	49

- 支持直接与间接数据寻址方式
- 一个带8位预置器的8位定时/计数器 (Timer0)
- 内部上电复位
- 内含一个低电压检测电路供掉电复位使用
- 上电复位计数器 (PWRT) 和振荡启动计数器 (Oscillator Start-up Timer OST)
- 内部振荡器集成了一个看门狗保证了可靠的操作同时软件使能看门狗操作
- 一类双向输入输出I/O口IOB
- 通过编程控制I/O端口的上拉/下拉、开路等状态
- 一个内部计数中断源；两个外部中断源：INT管脚,PortB的输入改变
- 通过INT管脚或者PortB的输入改变来实现睡眠唤醒
- 省电睡眠模式
- 内部有8MHz, 4MHz, 1MHz, 和 455KHzRC振荡器
- 有可靠的保证使得程序代码不被读出。
- 内部RC振荡器
- 提供以下振荡源的选择:
  - IRC: Internal Resistor/Internal Capacitor Oscillator – (内部电阻内部的电容RC振荡器)
- 工作电压范围: 2.0V - 5.5V
  - ≤1MHZ: 2.0V - 5.5V
  - ≤2MHZ: 2.2V - 5.5V
  - ≤4MHZ: 2.6V- 5.5V

## 概叙

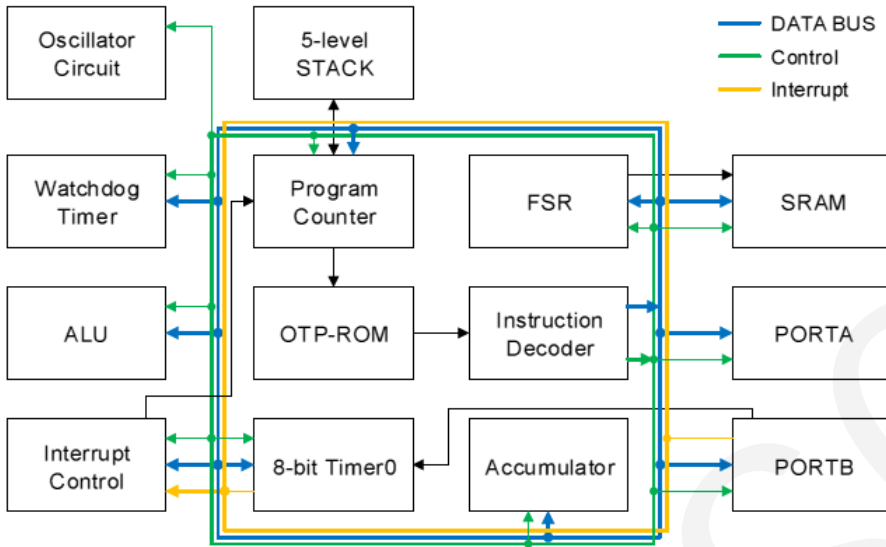
SM152是一款低功耗，高速，高噪声容限，EPROM/ROM基于8位CMOS工艺制造的单片机，采用RISC指令集，共有42条指令，除分支指令为两个周期指令以外其余为单周期指令。这种易用、易记的指令集大大缩短了开发时间。

SM152包含了上电复位(Power-on Reset POR)，掉电复位(Brown-out Reset BOR)，上电复位计数器 (Power-up Reset Timer PWRT)，振荡启动计数器 (Oscillator Start-up Timer OST)，看门狗定时器(Watchdog Timer)，EPROM/ROM, SRAM, 双向三态I/O口，(可以设置为上拉/下拉、开路)，省电睡眠模式，一个带8位预置器的8位定时/计数器，独立中断，睡眠唤醒模式和可靠的代码保护，有两个振荡源可供用户配置选择，包含省电振荡源和低功耗振荡器。

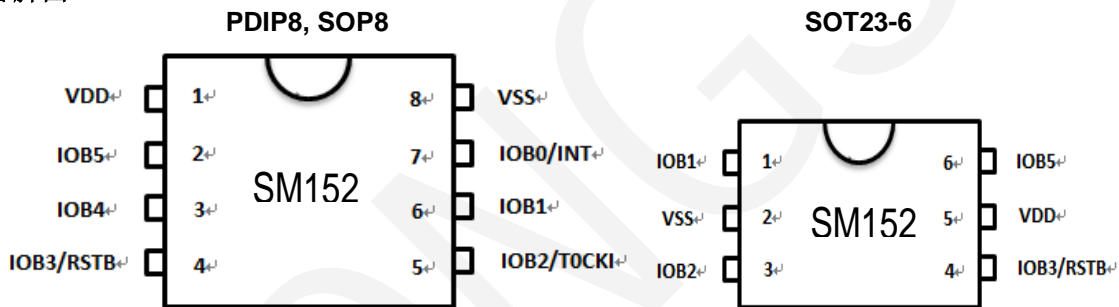
SM152可访问0.5K×13的程序存储空间。

SM152能直接或间接访问寄存器以及数据存储区，所有的特殊功能寄存器分布在数据存储区同时包含特定的程序指针。

### 方块图



### 管脚图



### 管脚功能描述

管脚名称	I/O	说明
IOB0/INT	I/O	双向I/O口同时具有系统唤醒功能 软件设置为上拉/下拉和开路 外部中断输入脚
IOB1	I/O	双向I/O口同时具有系统唤醒功能 软件设置为上拉/下拉和开路
IOB2/T0CKI	I/O	双向I/O口同时具有系统唤醒功能 软件设置为上拉/下拉和开路 外部计数输入脚
IOB3/RSTB	I	<b>开漏输出/输入具有系统唤醒功能</b> 系统复位输入脚. 底电平复位. 设置为复位输入时上拉
IOB4/OSCO	I/O	双向I/O口同时具有系统唤醒功能 (RCOUT 可选择IRC模式) 软件设置为上拉/开路 基于指令周期晶体振荡器输出 (RCOUT 可选择IRC模式)
IOB5	I/O	双向I/O口同时具有系统唤醒功能 (IRC 模式) 软件设置为上拉/开路
Vdd	-	电源

Vss	-	地
-----	---	---

Legend: I=输入, O=输出, I/O=输入/输出

## 1.0 存储器结构

SM152 存储器包含程序存储器和数据存储器。

### 1.1 程序存储器

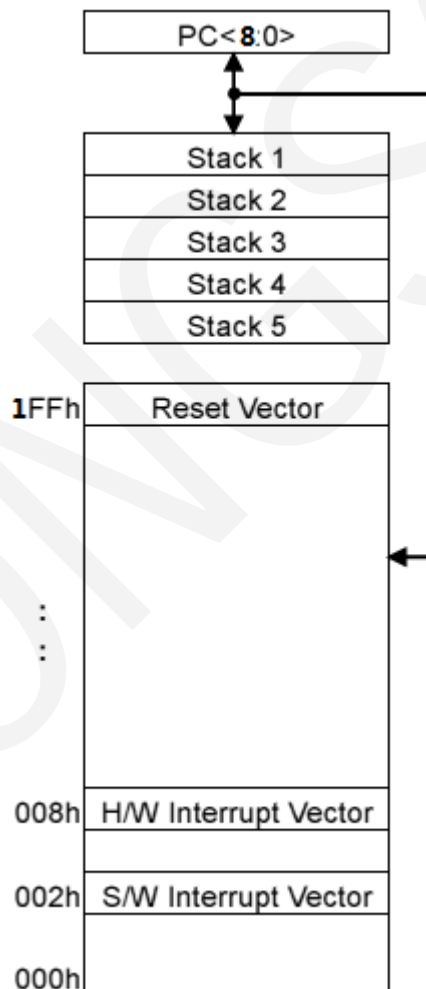
SM152有一个9位PC指针能访问0.5K×13的存储空间。

SM152的复位地址为1FFh。

H/W中断向量地址008h, S/W中断向量地址002h。

SM152的CALL/GOTO能指向在同一个程序页面（一个程序页面为0.5K）的所有存储空间

#### 程序存储器分布图和堆栈结构



SM152

### 1.2 数据存储器

数据存储器包含特殊功能器组和通用寄存器组，所有通用寄存器可以直接寻址或者通过FSR寄存器间接寻址。特殊功能寄存器用来控制CPU或外围功能模块的工作。

表 1.1: SM152寄存器列表

Table 1.1: Registers File Map for LT 503

Address	Description
00h	INDF
01h	TMR0
02h	PCL
03h	STATUS
04h	FSR
05h	N/A
06h	PORTB
07h	General Purpose Register
08h	PCON
09h	WUCON
0Ah	PCHBUF
0Bh	PDCON
0Ch	ODCON
0Dh	PHCON
0Eh	INTEN
0Fh	INTFLAG
10h ~ 3Fh	General Purpose Registers

N/A OPTION

06h IOSTB

表1.2: 通过OPTION 或IOST指令控制的寄存器

地址	说明	B7	B6	B5	B4	B3	B2	B1	B0
N/A (w)	OPTION	*	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
06h (w)	IOSTB	Port B I/O控制寄存器							

表1.3: 寄存器列表

地址	说明	B7	B6	B5	B4	B3	B2	B1	B0
00h (r/w)	INDF	通过FSR 访问数据区(不是一个实际的物理地址)							
01h (r/w)	TMR0	8位定时/计数器							
02h (r/w)	PCL	低8位PC指针							
03h (r/w)	STATUS	RST	GP1	GP0	/TO	/PD	Z	DC	C
04h (r/w)	FSR	*	*	间接地址访问指针 (RAM选择寄存器)					
06h (r/w)	PORTB			IOB5	IOB4	IOB3	IOB2	IOB1	IOB0
07h (r/w)	SRAM	通用寄存器							
08h (r/w)	PCON	WDTE	EIS	LVDTE	*	*	*	*	*
09h (r/w)	WUCON	-	-	WUB5	WUB4	WUB3	WUB2	WUB1	WUB0
0Ah (r/w)	PCHBUF	-	-	-	-	-	-	-	PCH0
0Bh (r/w)	PDCON	*	/PDB2	/PDB1	/PDB0	*	*	*	*
0Ch (r/w)	ODCON	-	-	ODB5	ODB4	-	ODB2	ODB1	ODB0
0Dh (r/w)	PHCON	*	*	/PHB5	/PHB4	*	/PHB2	/PHB1	/PHB0
0Eh (r/w)	INTEN	GIE	*	*	*	*	INTIE	PBIE	T0IE
0Fh (r/w)	INTFLAG	-	-	-	-	-	INTIF	PBIF	T0IF

Legend: - = unimplemented, read as '0', \* = unimplemented, read as '1'

## 2.0 功能介绍

### 2.1 寄存器操作

#### 2.1.1 INDF (间接寻址寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
00h (r/w)	INDF	通过FSR 访问数据区(不是一个实际的物理地址)							

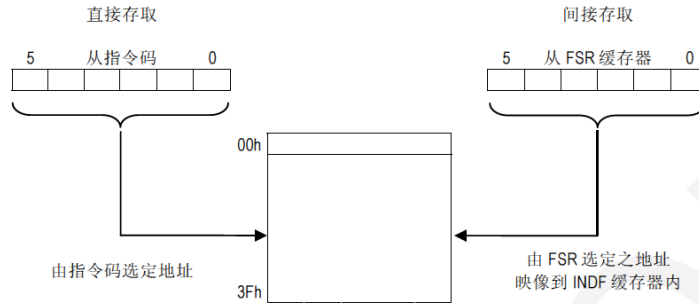
INDF不是一个实际的物理地址，间接寻址时INDF通过RAM选择寄存器（FSR）来访问其所指向的地址。间接寻址读操作直接读地址00h(FSR="0")，间接寻址不能对INDF直接进行写操作（尽管有些状态会发生改变）。

FSR的5-0位可以用来选择64个寄存器（地址：00h ~ 3Fh）。

#### 例 2.1:间接寻址

- 地址38内容为10h
- 地址39内容为0Ah
- 将38写入FSR 中
- 通过A读INDF返回10h
- FSR加1 (@FSR=39h)
- 通过A读INDF返回0A h

图2.1: 直接/间接存取



### 2.1.2 TMR0 (定时/计数器 Time lock/Counter register)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
00h (r/w)	TMR0	8位定时/计数器							

TMR0是一个8位定时/计数器寄存器,Timer0的时钟源可以取值于指令周期或外部实时钟 (T0CKI pin), 使用外部时钟需要设置OPTION的T0CS(T0CS=5)位为1。

使用TMR0的预置器需要设置OPTION的PSA (PSA =3)位为0, 这种模式下TMR0值的改变, 预置器被清零。

### 2.1.3 PCL (Low Bytes of Program Counter) & Stack

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
02h (r/w)	PCL	8位定时/计数器							

SM152的PC指针和堆栈的位数为9位, 堆栈有5级, 低位的PC指针为PCL寄存器,该寄存器时可读写的, 高位的PC指针为PCH寄存器, 该寄存器包含PC<9:8> 位, 该寄存器不能直接读写。PCH寄存器的改变是通过PCHBUF寄存器来实现的。每一条指令执行的时候他的PC指针包含下一条指令的操作地址。指令没有改变PC内容时候、在每一个指令周期PC指针自动加1。

对于GOTO指令有PC<8:0>, PCL 映射成PC<7:0>, PCHBUF不变。

对于CALL指令有PC<8:0>, 下一条指令地址被推进堆栈, PCL 映射成PC<7:0>, PCHBUF不变。

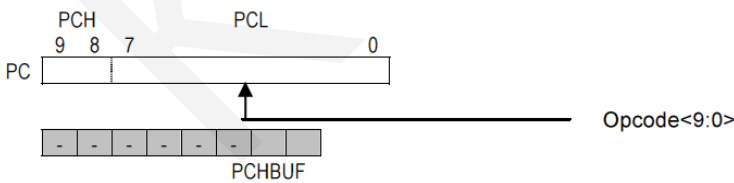
对于RETI, RETFIE, RETURN指令有PC<8:0>, PC的内容更改为出栈信息, PCL 映射成PC<7:0>, PCHBUF不变。

对于其他指令,PCL就是目标信息, PC<7:0>的内容就是指令地址或。不管怎样, PC<8> 来源于 PCHBUF<0> 位 (PCHBUF→ PCH)。

PCHBUF不会改变, 从而PCH不会改变。

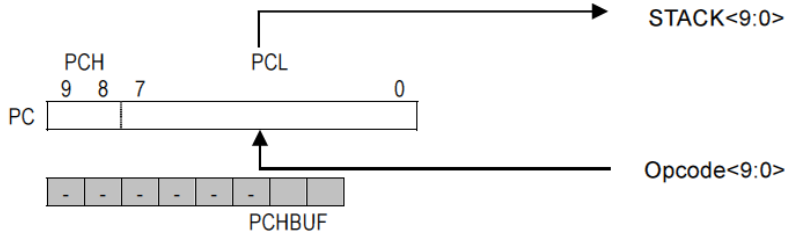
图2.2: 不同的指令调用PC指针跳转方式

#### 1、GOTO指令

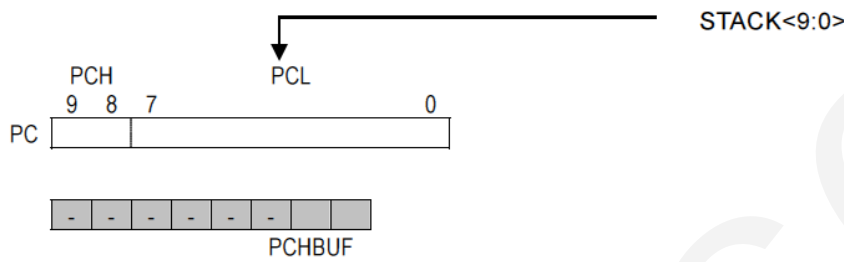


#### 2、CALL指令

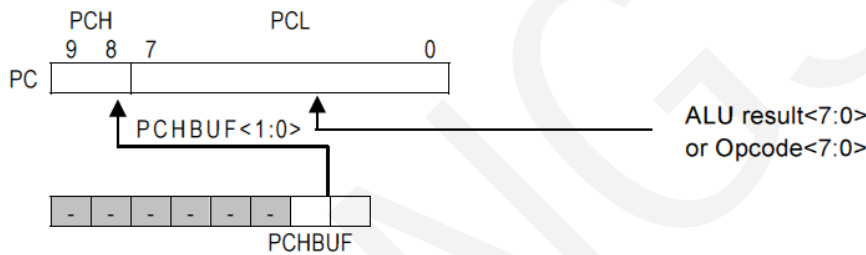




### 3、RETIA, RETFIE, RETURN指令



### 4、以PCL为目的的指令



注释1. PCHBUF只有在PCL内容是目标地址才有效，当PCL是运算结果时候，PCHBUF不起作用。

#### 2.1.4 STATUS (状态字寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
03h (r/w)	STATUS	RST	GP1	GP0	TO	PD	Z	DC	C

状态字寄存器包含运算标志，结果标志。

指令执行以后可能会影响STATUS寄存器的Z、DC、C标志位，则不能直接对这三个标志位进行写操作，这些标志位的设置由MCU的逻辑自动完成。同时，TO和PD位也是不能通过指令直接改变写操作。因此，与STATUS作为目标寄存器的指令后，结果可能会与预期的不同。例如：运行CLRR STATUS将把STATUS的高三位置零和Z标志位置1同时该寄存器的内容如下



C:进位标志

ADDAR, ADDIA

= 1,有进位

= 0,无进位

SUBAR, SUBIA

= 1,无借位

= 0,有借位

注释：减法是通过将2的补第二个操作数的执行。旋转(RRR, RLR)指令，该位装载高或低位源寄存器位。

DC:辅助进位/借位标志.(低四位向高四位进位/借位标志)

ADDAR, ADDIA

- = 1, 底4位有进位
- = 0, 底4位无进位
- SUBAR, SUBIA
- = 1, 底4位无借位
- = 0, 底4位有借位

Z : Zero bit.

- = 1, 算术或逻辑运算结果为“0”时
- = 0, 算术或逻辑运算结果不为“0”时

PD :Power down flag bit.

- = 1, 当系统上电时或执行“CLRWDT”指令后
- = 0, 当执行“SLEEP”指令后

TO :Time overflow flag bit.

- = 1, 当系统上电时或执行“CLRWDT”或SLEEP指令后
- = 0,看门狗定时器溢出

GP1:GP0 :通用寄存器读/写位

RST :定义系统复位类型位.

- = 1, 唤醒SLEEP或Port B脚位变化唤醒SLEEP
- = 0, 其他类型唤醒SLEEP.

### 2.1.5 FSR (间接寻址指针)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
04h (r/w)	FSR	*	*	间接寻址指针					

**Bit5:Bit0** : 用来选择访问间接寻址时目标寄存器地址. 具体描述见2.1.1。

**Bit7:Bit6** : 没有使用。

### 2.1.6 PORTA, PORTB (Port 寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
06h (r/w)	PORTB			IOB5	IOB4	IOB3	IOB2	IOB1	IOB0

读端口(PORTB 寄存器)的状态依赖于该端口是输入/输出模式，写端口是向锁存器写数据。

PORTB是一个6位端口数据寄存器。

### 2.1.7 PCON (电源控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
08h (r/w)	PCON	WDTE	EIS	LVDTE	*	*	*	*	*

**Bit4:Bit0** : Not used. 置“1”。

**LVDTE** : LVDT (低电压检测) 使能位。

- = 0, 关闭 LVDT。
- = 1, 使能 LVDT。

**EIS** : 定义管脚B0/INT功能位

- = 0, IOB0 (双向I/O 口) is selected. 屏蔽了INT功能。
- = 1, INT (外部中断输入脚) , 在这种模式下, PORTB 的IOB0必须置“1”。IOB0作为I/O口输入功能通过硬件屏蔽了, 读取INT管脚信息的与读PORTB.方式相同。

**WDTE** : WDT (watch-dog timer) 使能看门狗定时器

- = 0,关闭WDT。
- = 1,使能WDT。

### 2.1.8 WUCON (Port B输入改变/唤醒控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
09h (r/w)	WUCON	-	-	WUB5	WUB4	WUB3	WUB2	WUB1	WUB0

**WUB0** := 0,禁止IOB0 输入改变/唤醒功能  
=1,使能IOB0 输入改变/唤醒功能

**WUB1** := 0,禁止IOB1 输入改变/唤醒功能  
=1,使能IOB1 输入改变/唤醒功能

**WUB2** := 0,禁止IOB2 输入改变/唤醒功能  
=1,使能IOB2 输入改变/唤醒功能

**WUB3** := 0,禁止IOB3 输入改变/唤醒功能  
=1,使能IOB3 输入改变/唤醒功能

**WUB4** := 0,禁止IOB4 输入改变/唤醒功能  
=1,使能IOB4 输入改变/唤醒功能

**WUB5** := 0,禁止IOB5 输入改变/唤醒功能  
=1,使能IOB5 输入改变/唤醒功能

### 2.1.9 PCHBUF (PC指针高位缓冲区)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Ah (r/w)	PDHBUF								PCH0

**Bit1:Bit0** : 见2.1.3

**Bit7:Bit2** : 没有使用, 置“0”。

### 2.1.10 PDCON (I/O下拉控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Bh (r/w)	PDCON		/PDB2	/PDB1	/PDB0				

**/PDB0** := 0,使能IOB0内部下拉  
=1,禁止IOB0内部下拉

**/PDB1** := 0,使能IOB1内部下拉  
=1,禁止IOB1内部下拉

**/PDB2** := 0,使能IOB2内部下拉  
=1,禁止IOB2内部下拉

**Bit7** : 一般的读/写位

### 2.1.11 ODCON (I/O开路控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Ch (r/w)	ODCON			ODB5	ODB4		ODB2	ODB1	ODB0

**ODB0** := 0,禁止IOB0内部开路  
=1,使能 IOB0内部开路

**ODB1** := 0,禁止IOB1内部开路  
=1,使能 IOB1内部开路

**ODB2** := 0,禁止IOB2内部开路  
=1,使能 IOB2内部开路

**Bit3** : 一般的读/写位

**ODB4** : = 0,禁止IOB4内部开路  
= 1,使能 IOB4内部开路

**ODB5** : = 0,禁止IOB5内部开路  
= 1,使能 IOB5内部开路

### 2.1.12 PHCON (I/O上拉控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Dh (r/w)	PHCON			PHB5	PHB4		PHB2	PHB1	PHB0

**/PHB0** : = 0,使能IOB0内部上拉.  
= 1,禁止IOB0内部上拉

**/PHB1** : = 0,使能IOB1内部上拉.  
= 1,禁止IOB1内部上拉

**/PHB2** : = 0,使能IOB2内部上拉.  
= 1,禁止IOB2内部上拉

**Bit3** : 一般的读/写位

**/PHB4** : = 0,使能IOB4内部上拉.  
= 1,禁止IOB4内部上拉

**/PHB5** : = 0,使能IOB5内部上拉.  
= 1,禁止IOB5内部上拉

### 2.1.13 INTEN (中断屏蔽寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Eh (r/w)	INTEN	GIE	*	*	*	*	INTIE	PBIE	TOIE

**TOIE** : Timer0溢出中断屏蔽位。  
= 0,禁止Timer0溢出中断  
= 1,使能Timer0溢出中断

**PBIE** : Port B输入改变中断屏蔽位  
= 0,禁止Port B输入改变中  
= 1,使能Port B输入改变中

**INTIE** : 外部中断屏蔽位  
= 0,禁止外部中断.  
= 1,使能外部中断

**Bit6:BIT3** :没有使用。置“1”

**GIE** : 中断允许总控位

= 0,禁止所有中断. 对于睡眠唤醒模式的中断事件, MCU将执行SLEEP后的指令。

= 1,使能所有没有屏蔽的中断. 对于睡眠唤醒模式的中断事件, MCU将跳转到中断地址 (008h)。

注释 :在中断事件发生时, GIEB被硬件清零并禁止一切中断, 所以GIE以及与该中断相关的中断屏蔽位需要重开启。RETFIE 为退出中断程序并重新设置GIE =1允许中断。

### 2.1.14 INTFLAG (中断标志寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Fh (r/w)	INTFLAG	-	-	-	-	-	INTIF	PBIF	TOIF

**TOIF** : 溢出中断标志, 发生Timer0溢出中断置1, 软件设置清零

**PBIF** : Port B输入改变中断标志 interrupt flag. Port B输入改变时置1, 软件设置清零

**INTIF** : 外部中断标志. 当管脚INT上升沿/下降沿 (是上升沿/下降沿由 INTEDG 位 (OPTION<6>)决定) 时置1, 软件设置清零

**Bit7:BIT3** : 没有使用, 置0

### 2.1.15 ACC (Accumulator)累加器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
N/A (r/w)	ACC	累加器							

累加器是一个内部数据转化、指令操作和存放操作结果的存储单元, 不能被访问。

### 2.1.16 OPTION Register (选项寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
N/A (w)	OPTION	*	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

通过OPTION 指令访问

在执行OPTION 指令时候,该数据单元由ACC (累加器) 转化为选项寄存器 (OPTION Register)。  
选项寄存器是一个7位只写寄存器, 它的一些控制位主要用来配置与Timer0/WDT 分频器, Timer0, 外部中断选项相关信息。  
除INTEDG位以外其他位是只写并可以置1。

**PS2:PS0** : 分频率选择控制位

PS2:PS0	Timer0 Rate	WDT Rate
0 0 0	1:2	1:1
0 0 1	1:4	1:2
0 1 0	1:8	1:4
0 1 1	1:16	1:8
1 0 0	1:32	1:16
1 0 1	1:64	1:32
1 1 0	1:128	1:64
1 1 1	1:256	1:128

**PSA** : 分频器选择位.

- = 1, WDT (看门狗定时器)
- = 0, TMR0 (Timer0)

**T0SE** : TMR0触发方式控制位

- = 1, T0CKI脚下降沿触发计数
- = 0, T0CKI脚上升沿触发计数

**T0CS** : TMR0 时钟源选择控制位

- = 1, 外部T0CKI脚. 当IOST IOB2 = "0".时, IOB2/T0CKI脚设置为输入
- = 0, internal instruction clock cycle

**INTEDG** : 中断触发方式控制位.

- = 1, 中断触发方式为INT脚上升沿出发
- = 0, 中断触发方式为INT脚下降沿出发

**Bit7** : 没有使用

### 2.1.17 IOSTA & IOSTB (I/O口控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
N/A (w)	IOSTB			IOSTB5	IOSTB4	IOSTB3	IOSTB2	IOSTB1	IOSTB0

通过IOST指令访问

通过指令IOST R (06h)把累加器A的内容加载到I/O控制寄存器, 按位将IOSTB设为1表示该脚为输入 (高阻抗)、设为“0”时表示该脚为输出。

IOST寄存器只写, 系统复位以后设置为输入 (高阻抗)。

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Bh (r/w)	PDCON		/PDB2	/PDB1	/PDB0				

**/PDB0**: = 0,使能IOB 0内部下拉  
=1,禁止IOB0内部下拉

**/PDB1**: = 0,使能IOB1内部下拉  
=1,禁止IOB1内部下拉

**/PDB2**: = 0,使能IOB2内部下拉  
=1,禁止IOB2内部下拉

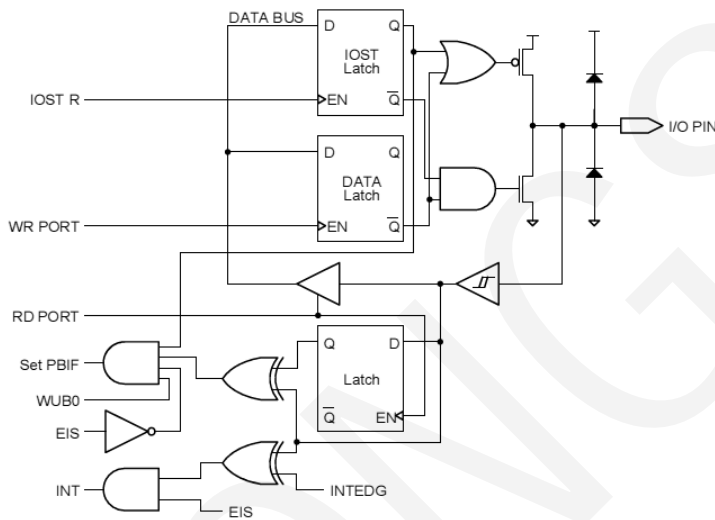
**Bit7** : 一般的读/写位

## 2.2 I/O Ports

port B为双向三态I/O口. Port B 为6脚I/O口. IOB2需要通过选项寄存器 (Option) 的T0CS ((OPTION<5>)) 位控制, 所有的I/O的输入/输出方式.有I/O控制寄存器(IOSTB)设置. IOB<5:4> 和 IOB<2:0>有相应的上拉控制位(PHCON 寄存器)来设置使能内部上拉, 如果设置为输出模式, 内部上拉功能会自动关闭. IOB<2:0>有相应的下拉控制位(PDCON寄存器)来设置使能内部下拉, 如果设置为输出模式, 内部下拉功能会自动关闭. IOB<5:4>和IOB<2:0>有相应的开路控制位(ODCON寄存器)来设置使能开路来设置输出为开路输出. IOB<5:0> 有输入改变中断/唤醒功能.它的每个管脚是否具有该功能通过取决于WUCON寄存器的相应位. 当EIS(PCON<6>)=1 时, IOB0作为外部中断输入脚, 在该模式下IOB0 输入改变中断/唤醒功能被硬件屏蔽, 即使软件已经设置为中断/唤醒功能可用也不可启用该功能. 配置能交替设置I/O口的不同功能, 功能交替设置完以后, 读的I/O的值为0.

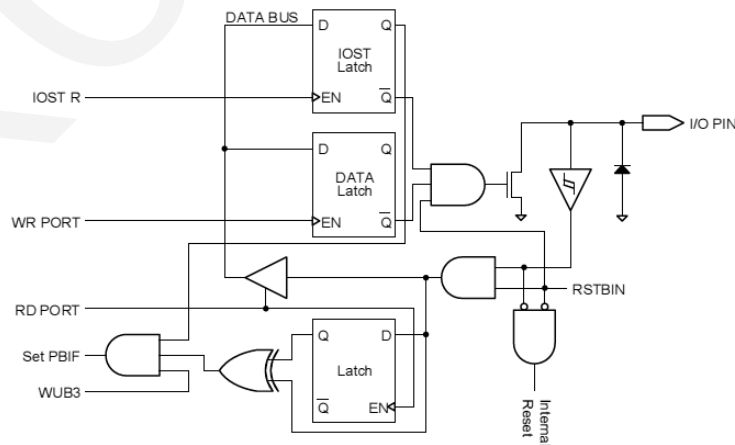
图2.3: I/O 脚的方块图

IOB0/INT:



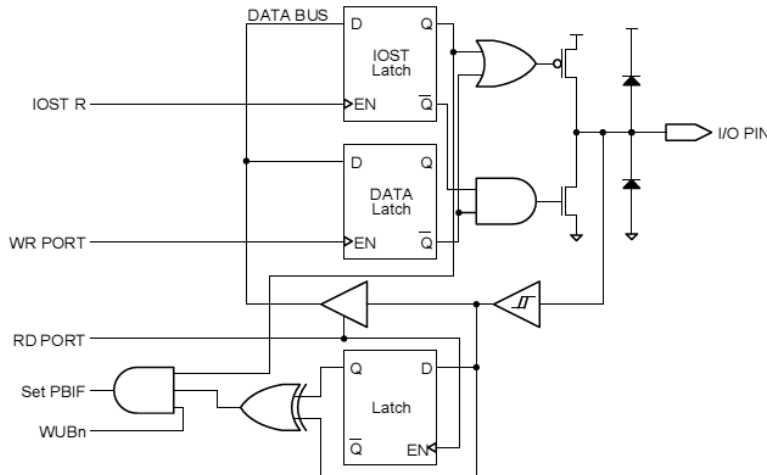
上拉/下拉和漏极开路在图中未显示

IOB3:



电压在这个引脚禁止超过VDD

IOB7 ~ IOB4, IOB2 ~ IOB1:



上拉/下拉和漏极开路在图中未显示

## 2.3 Timer0/WDT & Prescaler 2.3.1 Timer0

Timer0为8位定时/计数器， Timer0 的时钟源可以是内部或外部时钟源(T0CKI pin)

### 2.3.1.1 使用内部时钟: 定时模式

T0CS(OPTION<5>)=0为定时模式， 定时模式在没有预置器的情况下， 定时寄存器每个指令周期自动加1,设置TMR0以后， 定时器将在两个时钟周期以后开始自增。

### 2.3.1.2使用外部时钟: 计数模式

T0CS(OPTION<5>)=1为计数模式， 是选择通过T0CK管脚的上升或下降沿触发Timer0寄存器的增加由T0SE 位(OPTION<4>)决定， 外在时钟要求与内部时钟(Tosc)同步。同步以后， Timer0 实际增加有一个延迟。

在没有预置器的情况下， 外部时钟输入同样也可以作为预置器输出； T0CKI与内部时钟同步时能方便处理在T2 和 T4周期上的预分频. 因此T0CKI为高或低电平必须要保持两个以上时钟周期才有效。

有预置分频时器， 外部时钟输入被异步分频器平分， 这种常用来计算波形。因此： 因此T0CKI的一个波形周期至少4Tosc才能被 预置器平分。

## 2.3.2 看门狗定时器 (WDT)

看门狗定时器 (WDT) 的运行依赖于芯片里的RC振荡器， 无需任何额外电路即能工作。不管时钟OSCI和OSCO管脚是否关闭， 它都能运行， 如在睡眠模式。在一般操作或睡眠模式情况下， 看门狗定时器的溢出都会导致MCU复位同时TO (STATUS<4>)位被清零。 如WDTE 位(PCON<7>)清零. 看门狗定时器不能工作。

在没有预置器时看门狗的溢出为18 ms, 4.5ms, 288ms , 72ms这个时间可以通过SUT<1:0> 设置。

需要看门狗的溢出周期变长可以通过设置OPTION寄存器的看门狗定时器分频大于1:128.,因此最长的看门狗的溢出周期为 36.8 秒。 .

CLRWDWT指令能使WDT和预置器清零,启用看门狗可以防止超时， 如果超时MCU能复位。

SLEEP 指令重置WDT和预置器， 启用看门狗就给机器分派了一个最大睡眠时间。

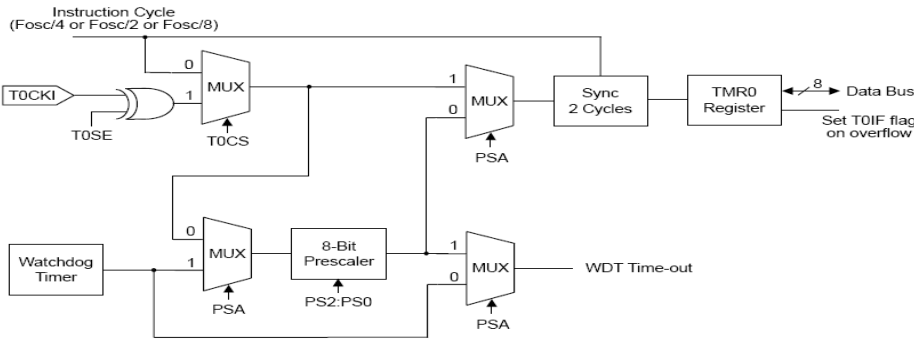
## 2.3.3 Prescaler (预置器)

有一个8位的向下计数器作为Timer0和看门狗定时器(WDT)的预置器。注意该预置器只能分配给Timer0 或 WDT使用， 不能两者同时使用。 PSA 位(OPTION<3>) 决定预置器是指派给Timer0还是WDT. PS<2:0> 位(OPTION<2:0>) 配置分频。当作为Timer0的预置器的时候， TMR0会被预置器清零。 . 当作为WDT的预置器的时候， CLRWDWT 指令会清除预置器内容。预置器不能读写. 机器复位， 预置器各位全为1。

为了避免机器非正常复位， 当Timer0 或 WDT的预置器发生改变的时候， 需要执行CLRWDWT 或 CLRR TMR0 指令， 反之亦然。

图2.4: Block Diagram of The Timer0/WDT Prescaler





## 2.4 中断方式

SM152系统具备有三种中断方式:

1. INT 管脚的外部中断
2. TMR0 溢出中断
3. Port B 输入改变中断 (IOB5:IOB0脚)

INTFLAG为中断标志寄存器, 决定该寄存器机器所发生的中断状态。

中断允许总控位GIE (INTEN<7>), 能使所有中断被开放 (GIE=1) 或屏蔽所有中断(GIE=0), 每中断能否启用决定INTEN 寄存器同时保证GIE=1。

中断发生时GIE 位 (在中断发生前GIE 位和该中断相关的中断屏蔽位置1) 被硬件清零从而禁止进一步中断 (LT503不区分中断优先级), 同时下条指令跳到008h后开始执行。中断标志位在中断允许总控位GIE重新置1的时候需要被软件清零以防止重复中断。一个中断标志位 (PBIF除外的) 会被它的中断事件置1, 而不管与它相关的中断屏蔽位是否启用。通过INTFLAG 和INTEN 的相应中位来判断是否发生中断以及中断类型。当通过INT指令发生软中断时, 下条指令跳到002后开始执行。

### 2.4.1 外部中断

外部中断INT管脚上升沿还是下降沿触发由 INTEDG 位 (OPTION<6>)决定, 当一个有效的跳变发生时标志位INTIF置1,如INTIE位 (INTEN<2>)清零, 该中断被屏蔽。

在睡眠之前INTIE 位已被置1, INT管脚可以作为系统睡眠条件。在睡眠之前GIE位已被置1机器唤醒以后会执行中断服务程序, 否则会运行睡眠以后的下一条指令。

### 2.4.2 Timer0 中断

TMR0发生溢出 (FFh → 00h)时TOIF标志位置1 (INTFLAG<0>). TOIE 位(INTEN<0>)清零, 该中断被屏蔽。

### 2.4.3 Port B 输入改变中断

输入改变中断触发时IOB<5:0> PBIF标志位置1 (INTFLAG<1>). PBIE位(INTEN<1>)清零, 该中断被屏蔽。

在输入改变中断发生之前, 必须读取port B信息

与PortB的管脚相对应的WUBn位 (WUCON<5:0>) i清零或设置为输出或IOB0 脚设置为外部中断输入脚INT 都拥有该功能。PBIE在睡眠之前置1, port B 输入脚改变中断也可以作为睡眠唤醒条件。在睡眠之前GIE位已被置1机器唤醒以后会执行中断服务程序, 否则会运行睡眠以后的下一条指令。

## 2.5 省电模式 (SLEEP)

执行SLEEP 指令以后机器进入省电模式。

执行SLEEP 指令, PD 位清零 (STATUS<3>), TO位置1,看门狗清零同时保持运行状态, 晶体停振。

I/O维持原状

### 2.5.1睡眠唤醒

在睡眠状态下, 单片机能通过以下方式唤醒:

1. RSTB 管脚复位
2. 看门狗复位 (机器设置了看门狗).
3. RB0/INT管脚中断,或PORTB 输入改变中断.

外部的RSTB管脚和看门狗通过设置PD 和TO 位都能使机器复位. PD 和TO 位, PD位置1用于上电复位, 清零用于SLEEP复位, TO 位清零用于看门狗溢出复位。

机器通过中断唤醒, 该中断屏蔽位置1, 中断唤醒不管GIE 是否置1。当GIE位被清零, 机器唤醒以后执行SLEEP指令以后的指令; 当GIE位被置1, 机器唤醒以后跳转到中断复位地址 (008h)。在高频或低频模式机器复位延迟时间为18/4.5/288/72ms (该延迟时间由SUT<1:0>

设置) 加上16个振荡周期。  
在IRC, 机器复位延迟时间为650us。

## 2.6 复位

LT503单片机能通过以下方式复位:

1. 上电复位(POR)
2. 掉电复位(Brown-out Reset BOR)
3. RSTB 管脚复位
4. 看门狗WDT溢出复位

一些寄存器在一些复位条件下没有影响, 在上电和其他一些复位情况下它们的状态是未知的。大多数寄存器会回到复位状态在上电复位, RSTB 管脚复位, 看门狗WDT溢出复位。

对Vdd上升信号检测告之芯片是否加上上电复位脉冲信号。 要使用这个特点, 用户需要把RSTB管脚连接到Vdd。

掉电复位作为一种典型应用主要用在 AC 或重载交换的应用上。

芯片上的低电压检测模块到电压低于一个固定的电压也会使芯片复位, 这样能保证芯片只能在正常电压范围内工作。

RSTB或WDT睡眠唤醒也导致芯片复位, 其复位操作的不会在睡眠之前。

根据不同的复原状态设置对TO和PD位(STATUS<4 : 3>)置1或清零。

### 2.6.1上电复位计数器\_(Power-up Reset Timer PWRT)

上电复位计数器提供一个 18/4.5/288/72ms 延迟时间 (该延迟时间由 SUT<1:0>设置) (或 140us,基于不同的振荡源和复位条件) 在 Power-on Reset (POR), Brown-out Reset (BOR), RSTB Reset 或 看门狗溢出复位。只要 PWRT 在运行, 设备就一直保持的复位状态 Vdd、温度和其他变化而会影响 PWDT 控制的设备延迟时间。

表2.1: PWRT Period

Oscillator Mode	Power-on Reset Brown-out Reset	RSTB Reset WDT time-out Reset
IRC	18/4.5/288/72 ms	140 us

### 2.6.2\_振荡启动计数器(Oscillator Start-up Timer OST)

在PWRT 延迟 (18/4.5/288/72ms) 之后振荡启动计数器会再提供一个64个clock的延迟。这种延迟晶体谐振器能提供稳定的振荡源, 这段时间内只要OST在工作, 设备就一直保持的复位状态。

在 OSCI 信号的振幅到达振荡器输入最大振幅之后, 该计数器只开始增加。

### 2.6.3复位顺序

SM152复位时序如下:

1. 复位锁存器置1, PWRT & OST 清零。
2. 当内部的POR, BOR, RSTB 复位或 WDT溢出复位脉冲加载完成后, PWRT开始计数。
3. PWRT溢出以后, OST开始计数延迟。
4. OST延迟完成以后, 复位锁存器清零最后芯片得到一个复位信号。

在高频或低频振荡模式机器复位延迟时间为18/4.5/288/72ms加上64个振荡周期, 在IRC, ERC振荡模式单机会在Power-on Reset (POR), Brown-out Reset (BOR), 或RSTB复位以后在延迟650us, 看门狗溢出复位后再延迟18/4.5/288/72ms的时间。

图2.5: 复位电路结构图

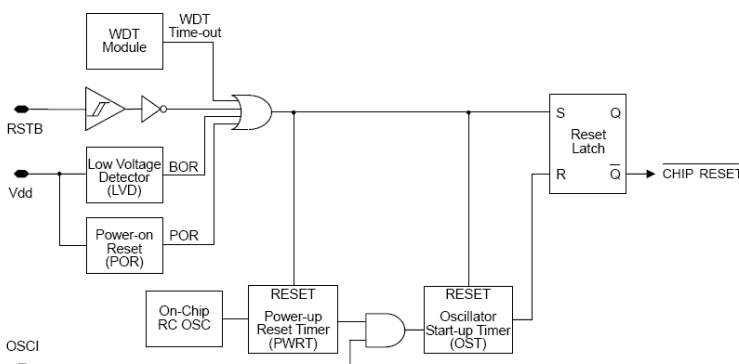


表2.2: 复位以后各个寄存器状态列表

寄存器	地址	上电复位 掉电复位	RSTB复位 WDT 复位
ACC	N/A	xxxx xxxx	uuuu uuuu
OPTION	N/A	-011 1111	-011 1111
IOSTB	N/A	1111 1111	1111 1111
INDF	00h	xxxx xxxx	uuuu uuuu
TMR0	01h	xxxx xxxx	uuuu uuuu
PCL	02h	1111 1111	1111 1111
STATUS	03h	0001 1xxx	000# #uuu
FSR	04h	11xx xxxx	11uu uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu
General Purpose Register	07h	xxxx xxxx	uuuu uuuu
PCON	08h	101- ----	101- ----
WUCON	09h	--00 0000	--00 0000
PCHBUF	0Ah	---- --0	---- --0
PDCON	0Bh	--11 1---	--11 1---
ODCON	0Ch	--00 0000	--00 0000
PHCON	0Dh	--11 1111	--11 1111
INTEN	0Eh	0--- -000	0--- -000
INTFLAG	0Fh	---- -000	---- -000
General Purpose Registers	10 ~ 3Fh	xxxx xxxx	uuuu uuuu

Legend: u = 不变, x = 未知, - = 不起作用, # = 参见下表的值

表2.3: RST/TO/PD 复位和唤醒后的状态

RST	/TO	/PD	复位方式
0	1	1	Power-on Reset
0	1	1	Brown-out reset
0	u	u	RSTB Reset during normal operation
0	1	0	RSTB Reset during SLEEP
0	0	1	WDT Reset during normal operation
0	0	0	WDT Wake-up during SLEEP
1	1	0	Wake-up on pin change during SLEEP

Legend: u = 不变

表: 2.4: TO/PD状态位影响事件

事件	TO	PD
Power-on	1	1
WDT Time-Out	0	u
SLEEP instruction	1	0
CLRWDWT instruction	1	1

Legend: u = 不变

## 2.7 十六进制转化为十进制 (Hexadecimal Convert to Decimal HCD)

SM152具有十进制格式化功能。 当一个寄存器里面的内容需要十进制转化的时候,在执行操作ALU以后必须把结果进行相应的进制转化。 一个数据在处理过程中进行了转化成了十进制,那么所有对这个数进行的操作(包含存放该数据的RAM单元, accumulator (ACC),立即数,以及所要查表信息)都的进行十进制转化,这样的运算结果才正确。

DAA指令能在加法运算完成以后将ACC 里的数据从十六进制转化为十进制重存给ACC

转换操作在例子 2.2 中被说明。

## 例2.2: DAA 转化

Address	Code
NA	#include <8PB53B.ASH>
n	...
n+1	MOVIA 0x90 ;Set immediate data = decimal format number "90" (ACC ← 90h)
n+2	MOVAR 0x30 ;Load immediate data "90" to data memory address 30H
n+3	MOVIA 0x10 ;Set immediate data = decimal format number "10" (ACC ← 10h)
n+4	ADDAR 0x30,A ;Contents of the data memory address 30H and ACC are binary-added ;the result loads to the ACC (ACC ← A0h, C ← 0)
n+5	DAA ;Convert the content of ACC to decimal format, and restored to ACC ;The result in the ACC is "00" and the carry bit C is "1". This represents the ;decimal number "100"
n+6	...

DAS指令能在减法运算完成以后将ACC 里的数据从十六进制转化为十进制重存给ACC 转换操作在例子2.3中被说明

## 例2.3: DAS 转化

Address	Code
NA	#include <8PB53B.ASH>
n	...
n+1	MOVIA 0x10 ;Set immediate data = decimal format number "10" (ACC ← 10h)
n+2	MOVAR 0x30 ;Load immediate data "90" to data memory address 30H
n+3	MOVIA 0x20 ;Set immediate data = decimal format number "20" (ACC ← 20h)
n+4	SUBAR 0x30,A ;Contents of the data memory address 30H and ACC are binary-subtracted ;the result loads to the ACC (ACC ← F0h, C ← 0)
n+5	DAS ;Convert the content of ACC to decimal format, and restored to ACC ;The result in the ACC is "90" and the carry bit C is "0". This represents the ;decimal number "-10"
n+6	...

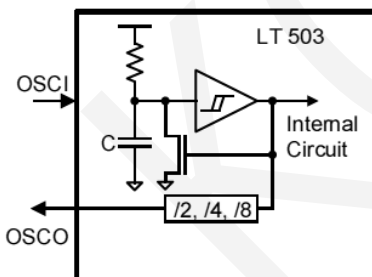
## 2.8 振荡器配置 (Oscillator Configurations)

SM152有一种不同的振荡模式，用户可通过编程Fosc配置位来选择相应的振荡方式：

- IRC:内部电阻内部电容振荡器

使用 IRC/ERIC 振荡模式为成本节省主要使用在定时无须精确场合下的应用，单片机具有 4 种不同的振荡频率，8MHz, 4MHz, 1MHz, and 455KHz, 通过 (RCM<1:0>)来选择一种。

图解2.10: IRC 振荡器模式 (内部R, 内部C 振荡器)



## 2.9 配置选项

表2.4: 配置选项0

位	名称	说明
2, 1, 0	Fosc<2:0>	振荡源选择位 = 1, 1, 1 → IRC mode (外部的RC振荡器) (默认) IOB4/OSCO管脚为取OSCOUT功能 = 0, 1, 1 → IRC mode (internal R & C) IOB4/OSCO管脚为取OSCOUT功能
5, 4, 3	LVDT<2:0>	低电压检测选择位 = 1, 1, 1 → 禁止低电压检测(默认) = 1, 1, 0 → enable, LVDT voltage = 2.0V, 睡眠模式 = 1, 0, 1 → enable, LVDT voltage = 2.0V = 1, 0, 0 → enable, LVDT voltage = 3.6V = 0, 1, 1 → enable, LVDT voltage = 1.8V = 0, 1, 0 → enable, LVDT voltage = 2.2V = 0, 0, 1 → enable, LVDT voltage = 2.4V = 0, 0, 0 → enable, LVDT voltage = 2.6V
7, 6	RCM<1:0>	IRC选择位 = 1, 1 → 4MHz (默认) = 1, 0 → 8MHz = 0, 1 → 1MHz = 0, 0 → 455KHz
10, 9, 8	SUT<2:0>	PWRT & WDT计数周期选择位 (其值必须是分频率的倍数) = 1, 1, 1 → PWRT = WDT prescaler rate = 18ms (default) = 1, 1, 0 → PWRT = WDT prescaler rate = 4.5ms = 1, 0, 1 → PWRT = WDT prescaler rate = 288ms = 1, 0, 0 → PWRT = WDT prescaler rate = 72ms = 0, 1, 1 → PWRT = 650us, WDT prescaler rate = 18ms = 0, 1, 0 → PWRT = 650us, WDT prescaler rate = 4.5ms = 0, 0, 1 → PWRT = 650us, WDT prescaler rate = 288ms = 0, 0, 0 → PWRT = 650us, WDT prescaler rate = 72ms
11	OSCOUT	IRC 模式下IOB4/OSCO功能选择位置 = 1, OSCO (默认) = 0, IOB4
12	RSTBIN	IOB3/RSTB选择位置 = 1, IOB3 (默认) = 0, RSTB

表2.5: 配置选项1

位	名称	说明
0	WDTEN	看门狗使能位 = 1, 使能WDT (默认) = 0, 禁止WDT
1	PROTECT	代码保护选择位 = 1, 1 → 代码不加密EPROM code protection off (默认) = 0, 0 → 代码加密EPROM code protection on
3, 2	OSCD<1:0>	指令运行周期选择位 = 1, 1 → 4个振荡周期 (默认) = 1, 0 → 2个振荡周期 = 0, 1 → 1个振荡周期 = 0, 0 → 8个振荡周期
4	PMOD	省电模式控制位 = 1, 非省电模式 (默认) = 0, 省电模式
5	RDPORT	IO作为输出时,读端口方式控制位 = 1, 从寄存器读 (默认) = 0, 从管脚读
6	SCHMITT	I/O输入缓冲控制位 = 1, 通过Schmitt触发器(默认) = 0, 不通过Schmitt触发器
12 ~ 7	-	没有使用

表2.6: 配置选项2

位	名称	说明
7 ~ 0	CAL<7:0>	IRC 校准选择位

12 ~ 8

-

没有使用

### 3.0 指令集合

操作语法	说明	操作内容	指令周期	影响标志位
<b>BCR</b> R, bit	Clear bit in R	0→R<b>	1	-
<b>BSR</b> R, bit	Set bit in R	1→R<b>	1	-
<b>BTRSC</b> R, bit	Test bit in R, Skip if Clear	Skip if R<b> = 0	1/2 <sup>(1)</sup>	-
<b>BTRSS</b> R, bit	Test bit in R, Skip if Set	Skip if R<b> = 1	1/2 <sup>(1)</sup>	-
<b>NOP</b>	No Operation	No operation	1	-
<b>CLRWDT</b>	Clear Watchdog Timer	00h→WDT, 00h →WDT prescaler	1	TO,PD
<b>OPTION</b>	Load OPTION register	ACC→ OPTION	1	-
<b>SLEEP</b>	Go into power-down mode	00h→ WDT, 00h→ WDT prescaler	1	TO,PD
<b>DAA</b>	Adjust ACC's data format from HEX to DEC after any addition operation	ACC(hex) → ACC(dec)	1	C
<b>DAS</b>	Adjust ACC's data format from HEX to DEC after any subtraction operation	ACC(hex) → ACC(dec)	1	-
<b>INT</b>	S/W interrupt	PC + 1→Top of Stack, 002h→ PC	2	-
<b>RETURN</b>	Return from subroutine	Top of Stack→ PC	2	-
<b>RETFIE</b>	Return from interrupt, set GIE bit	Top of Stack→ PC, 1→ GIE	2	-
<b>CLRA</b>	Clear ACC	00h ACC	1	Z
<b>IOST</b> R	Load IOST register	ACC→ IOST register	1	-
<b>CLRR</b> R	Clear R	00h→ R	1	Z
<b>MOVAR</b> R	Move ACC to R	ACC→ R	1	-
<b>MOVR</b> R, d	Move R	R→ dest	1	Z
<b>DECR</b> R, d	Decrement R	R - 1 →dest	1	Z
<b>DECRSZ</b> R, d	Decrement R, Skip if 0	R - 1 → dest, Skip if result = 0	1/2 <sup>(1)</sup>	-
<b>INCR</b> R, d	Increment R	R + 1 → dest	1	Z
<b>INCRSZ</b> R, d	Increment R, Skip if 0	R + 1 → dest, Skip if result = 0	1/2 <sup>(1)</sup>	-
<b>ADDAR</b> R, d	Add ACC and R	R + ACC→ dest	1	C, DC, Z
<b>SUBAR</b> R, d	Subtract ACC from R	R - ACC→ dest	1	C, DC, Z
<b>ADCAR</b> R, d	Add ACC and R with Carry	R + ACC + C → dest	1	C, DC, Z
<b>SBCAR</b> R, d	Subtract ACC from R with Carry	R + ACC + C→dest	1	C, DC, Z
<b>ANDAR</b> R, d	AND ACC with R	ACC and R→dest	1	Z
<b>IORAR</b> R, d	Inclusive OR ACC with R	ACC or R→ dest	1	Z
<b>XORAR</b> R, d	Exclusive OR ACC with R	R xor ACC→ dest	1	Z
<b>COMR</b> R, d	Complement R	R→dest	1	Z
<b>RLR</b> R, d	Rotate left f through Carry	R<7>→C, R<6:0>→ dest<7:1>, C→ dest<0>	1	C
<b>RRR</b> R, d	Rotate right f through Carry	C ←dest<7>, R<7:1> ← dest<6:0>, R<0>← C	1	C
<b>SWAPR</b> R, d	Swap R	R<3:0> →dest<7:4>, R<7:4> →dest<3:0>	1	-

<b>MOVIA I</b>	Move Immediate to ACC	I → ACC	1	-
<b>ADDIA I</b>	Add ACC and Immediate	I + ACC → ACC	1	C, DC, Z
<b>SUBIA I</b>	Subtract ACC from Immediate	I - ACC → ACC	1	C, DC, Z
<b>ANDIA I</b>	AND Immediate with ACC	ACC and I → ACC	1	Z
<b>IORIA I</b>	OR Immediate with ACC	ACC or I → ACC	1	Z
<b>XORIA I</b>	Exclusive OR Immediate to ACC	ACC xor I → ACC	1	Z
<b>RETIA I</b>	Return, place Immediate in ACC	I ACC, Top of Stack → PC	2	-
<b>CALL I</b>	Call subroutine	PC + 1 → Top of Stack, I → PC	2	-
<b>GOTO I</b>	Unconditional branch	I → PC	2	-

注释: 1. 两周期指令为分支跳转指令

2. bit : Bit 地址为8位寄存器R中的某一位

R : 寄存器地址 (00h to 3Fh)

I : 立即数

ACC : 累加器

d : 目的选择;

=0 (结果存放在ACC)

=1 (结果存放在R)

dest : 目的地

PC : 程序指针

PCHBUF : 高位缓冲程序指针

WDT : 看门狗计数器

GIE : 中断允许总控制位

TO : 计数溢出位

PD : 省电模式选择位

C : 进位/借位标志

DC : 辅助进位/借位标志.(低四位向高四位进位/借位标志)

Z : 零标志

### ADCAR(带进位加法)

### Add ACC and R with Carry

语法

ADCAR R, d

操作数

0 ≤ R ≤ 63

d ∈ [0,1]

操作内容

R + ACC + C → dest

受影响的标志

C, DC, Z

说明

将A寄存器的内含值加上R寄存器的内含值(带进位), 如果‘d’是0结果在ACC中存放。 如果‘d’是‘1’结果在‘R’中存放。

指令执行周期

1

### ADDAR (加法指令)

### ACC and R with Carry

语法

ADDAR R, d

操作数

0 ≤ R ≤ 63

d ∈ [0,1]

操作内容

R + ACC → dest

受影响的标志

C, DC, Z

说明

将A寄存器的内含值加上R寄存器的内含值(不带进位), 如果‘d’是0结果在ACC中存放。 如果‘d’是‘1’结果在‘R’中存放。

指令执行周期

1

### ADDIA

### Add ACC and Immediate

语法	ADDIA I
操作数	$0 \leq I \leq 255$
操作内容	$ACC + I \rightarrow ACC$
受影响的标志	C, DC, Z
说明	将A寄存器的内含值加上立即数‘I’，结果在ACC中存放。
指令执行周期	1
<b>ANDAR</b>	<b>AND ACC and R</b>
语法	ANDAR R, d
操作数	$0 \leq R \leq 63$ $d \in [0,1]$
操作内容	$ACC \text{ and } R \rightarrow dest$
受影响的标志	Z
说明	将A寄存器内含值和R寄存器做相与操作，如果‘d’是0结果在ACC中存放。如果‘d’是‘1’结果在‘R’中存放’。
指令执行周期	1
<b>ANDIA</b>	<b>AND Immediate with ACC</b>
语法	ANDIA I
操作数	$0 \leq I \leq 255$
操作内容	$ACC \text{ and } I \rightarrow dest$
受影响的标志	Z
说明	将A寄存器的内含值与立即数‘I’做相与操作，结果在ACC中存放
指令执行周期	1
<b>BSR</b>	<b>Set Bit in R</b>
语法	BCF R, b
操作数	$0 \leq R \leq 63$ $0 \leq b \leq 7$
操作内容	$1 \rightarrow R\langle b \rangle$
受影响的标志	无
说明	R寄存器的位“b”被设成1
指令执行周期	1
<b>BTRSC</b>	<b>Test Bit in R, Skip if Clear</b>
语法	BTRSC R, b
操作数	$0 \leq R \leq 63$ $0 \leq b \leq 7$
操作内容	当 $R\langle b \rangle = 0$ 跳过后条指令
受影响的标志	无
说明	$R\langle b \rangle = 0$ 跳过后条指令 $R\langle b \rangle = 0$ 时，该指令周期中提取的下条指令被丢弃，并以执行NOP操作来替换这条2周期指令。
指令执行周期	1(2)
<b>BTRSS</b>	<b>Test Bit in R, Skip if Set</b>
语法	BTRSS R, b
操作数	$0 \leq R \leq 63$ $0 \leq b \leq 7$
操作内容	当 $R\langle b \rangle = 1$ 跳过后条指令
受影响的标志	无
说明	$R\langle b \rangle = 1$ 跳过后条指令 $R\langle b \rangle = 1$ 时，该指令周期中提取的下条指令被丢弃，并以执行NOP操作来替换这条2周期指令。
指令执行周期	1(2)
<b>CALL</b>	<b>Subroutine Call</b>



**CALL I**

语法  
操作数  
操作内容  
受影响的标志  
说明  
指令执行周期

CALL I  
 $0 \leq I \leq 1023$   
 PC +1 → Top of Stack;  
 I → PC  
 无  
 子程序调用。首先下一条指令地址(PC+1)进栈。10位立即地址被装入PC指针的位<9 : 0>. CALL是二周期指令。  
 2

**CLRA Clear ACC**

语法  
操作数  
操作内容  
受影响的标志  
说明  
指令执行周期

CLRA  
 无  
 00h → ACC  
 I → Z  
 Z  
 ACC被清零, Z标志为置1  
 1

**CLRR Clear R**

语法  
操作数  
操作内容  
受影响的标志  
说明  
指令执行周期

CLRR R  
 $0 \leq R \leq 63$   
 00h → R  
 I → Z  
 Z  
 R被清零, Z标志为置1  
 1

**CLRWDT Clear Watchdog Timer**

语法  
操作数  
操作内容  
受影响的标志  
说明  
指令执行周期

CLRWDT  
 无  
 00h → WDT;  
 00h → WDT prescaler (已经设置了WDT预置器);  
 1 → TO;  
 1 → PD  
 TO,PD  
 CLRWDT指令重置WDT, 如已经设置了WDT预置器, 也重置WDT预置器; 并把TO,PD位置1  
 1

**COMR Complement R**

语法  
操作数  
操作内容  
受影响的标志  
说明  
指令执行周期

COMR R, d  
 $0 \leq R \leq 63$   
 $d \in [0,1]$   
 R → dest  
 Z  
 将R内含内容取补数, 如果‘d’是0结果在ACC中存放。如果‘d’是1结果在‘R’中存放。  
 1

**DAA Adjust ACC's data format from HEX to DEC**

语法  
操作数  
操作内容  
受影响的标志  
说明  
指令执行周期

DAA  
 无  
 ACC(hex) → ACC(dec)  
 C  
 在有些加法操作以后把ACC内值的十六进制转化十进制,  
 1

**DAS Adjust ACC's data format from HEX to DEC**

语法	DAS
操作数	无
操作内容	ACC(hex) → ACC(dec)
受影响的标志	C
说明	在有些减法操作以后把ACC内值的十六进制转化十进制，
指令执行周期	1
<b>DECR</b>	<b>Decrement R</b>
语法	DECR R, d
操作数	0 ≤ R ≤ 63 d ∈ [0,1]
操作内容	R - 1 → dest
受影响的标志	Z
说明	递减R寄存器的值，如果‘d’是0结果在ACC中存放。如果‘d’是1结果在‘R’中存放。
指令执行周期	1
<b>DECRSZ</b>	<b>Decrement R, Skip if 0</b>
语法	DECRSZ R, d
操作数	0 ≤ R ≤ 63 d ∈ [0,1]
操作内容	R - 1 → dest 如果结果等于0，跳过下条指令
受影响的标志	无
说明	递减R寄存器的值，如果‘d’是0结果在ACC中存放。如果‘d’是1结果在‘R’中存放。 如果结果等于0，该指令周期中提取的下条指令被丢弃，并以执行NOP操作来替换这条2周期指令。
指令执行周期	1(2)
<b>GOTO</b>	<b>Unconditional Branch</b>
语法	GOTO I
操作数	0 ≤ I ≤ 1023
操作内容	I → PC
受影响的标志	无
说明	无条件跳转。10位立即地址被装载入PC指针的位<9 : 0>。GOTO是二周期指令。
指令执行周期	2
<b>INCR</b>	<b>Increment R</b>
语法	INCR R, d
操作数	0 ≤ R ≤ 63 d ∈ [0,1]
操作内容	R + 1 → dest
受影响的标志	Z
说明	将被指定R寄存器的内含值加1，如果‘d’是0结果在ACC中存放。如果‘d’是1结果在‘R’中存放。
指令执行周期	2
<b>INCRSZ</b>	<b>Increment R, Skip if 0</b>
语法	INCRSZ R, d
操作数	0 ≤ R ≤ 63 d ∈ [0,1]
操作内容	R + 1 → dest 如果结果等于0，跳过下条指令
受影响的标志	无
说明	将被指定R寄存器的内含值加1，如果‘d’是0结果在ACC中存放。如果‘d’是1结果在‘R’中存放。 如果结果等于0，该指令周期中提取的下条指令被丢弃，并以执行NOP操作来替换这条2周期指令。
指令执行周期	1(2)
<b>INT</b>	<b>S/W Interrupt</b>

语法	INT
操作数	无
操作内容	PC +1 → Top of Stack; 002h → PC
受影响的标志	无
说明	子程序调用。首先下一条指令地址(PC+1)进栈。10位地址002h被装载入PC指针的位<9 : 0>. CALL是二周期指令。
指令执行周期	2
<hr/>	
<b>IORAR</b>	<b>OR ACC with R</b>
语法	IORAR
操作数	0 ≤ R ≤ 63 d ∈ [0,1]
操作内容	ACC or R → dest
受影响的标志	Z
说明	将A寄存器内含值和R寄存器做或操作, 如果‘d’是0结果在ACC中存放。如果‘d’是1结果在‘R’中存放。
指令执行周期	1
<hr/>	
<b>IORIA</b>	<b>OR Immediate with ACC</b>
语法	IORIA I
操作数	0 ≤ I ≤ 255
操作内容	ACC or I → dest
受影响的标志	Z
说明	将A寄存器的内含值与立即数‘I’做相与操作, 结果在ACC中存放
指令执行周期	1
<hr/>	
<b>IOST</b>	<b>Load IOST Register</b>
语法	IOST R
操作数	R = 5 or 6
操作内容	ACC → IOST register R
受影响的标志	无
说明	将A寄存器的内含值加载到IOST register R中
指令执行周期	1
<hr/>	
<b>MOVAR</b>	<b>Move ACC to R</b>
语法	MOVAR R
操作数	0 ≤ R ≤ 63
操作内容	ACC → R
受影响的标志	无
说明	将数据从ACC传送到R
指令执行周期	1
<hr/>	
<b>MOVIA</b>	<b>Move Immediate to ACC</b>
语法	MOVIA I
操作数	0 ≤ I ≤ 255
操作内容	I → ACC
受影响的标志	无
说明	将立即值载入A寄存器中
指令执行周期	1
<hr/>	
<b>MOVR</b>	<b>Move Immediate to ACC</b>

**语法** MOVR R, d  
**操作数**  $0 \leq R \leq 63$   
 $d \in [0, 1]$   
**操作内容**  $R \rightarrow dest$   
**受影响的标志** 无  
**说明** 将A寄存器内容载入R中，如果‘d’是0结果在ACC中存放。如果‘d’是1结果在‘R’中存放。‘d’为1用来测试该寄存器对标志Z 是否有影响  
**指令执行周期** 1

---

**NOP No Operation**

**语法** NOP  
**操作数** 无  
**操作内容** 无操作  
**受影响的标志** 无  
**说明** 不做任何操作  
**指令执行周期** 1

---

**OPTION Load OPTION Register**

**语法** OPTION  
**操作数** 无  
**操作内容**  $ACC \rightarrow OPTION$   
**受影响的标志** 无  
**说明** 将A寄存器内容载入OPTION中  
**指令执行周期** 1

---

**RETFIE Return from Interrupt, Set ‘GIE’ Bit**

**语法** RETFIE  
**操作数** 无  
**操作内容**  $Top\ of\ Stack \rightarrow PC$   
**受影响的标志** 无  
**说明** 程序计数器载入堆栈返回地址。‘GIE’位被设置到1。这是二周期指令。  
**指令执行周期** 2

---

**RETIA Return with Immediate in ACC**

**语法** RETIA I  
**操作数**  $0 \leq I \leq 255$   
**操作内容**  $I \rightarrow ACC;$   
 $Top\ of\ Stack \rightarrow PC$   
**受影响的标志** 无  
**说明** 程序计数器载入堆栈返回地址，并把立即数送入A中。这是二周期指令。  
**指令执行周期** 2

---

**RETURN Return from Subroutine**

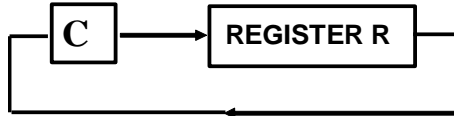
**语法** RETURN  
**操作数** 无  
**操作内容**  $Top\ of\ Stack \rightarrow PC$   
**受影响的标志** 无  
**说明** 程序计数器载入堆栈返回地址。这是二周期指令。  
**指令执行周期** 2

---

**RLR Rotate Left f through Carry**

**语法**  
操作数  
操作内容  
受影响的标志  
说明

RLR R, d  
 $0 \leq R \leq 63$   
 $d \in [0,1]$   
 $R \langle 7 \rangle \rightarrow C$ ;  
 $R \langle 6:0 \rangle \rightarrow \text{dest} \langle 7:1 \rangle$ ;  
 $C \rightarrow \text{dest} \langle 0 \rangle$   
 C  
 R寄存器的内含值又移1-bit，右移时包含C(进位标志)，如下图，结果存放由‘d’决定，如果‘d’是0结果在ACC中存放。如果‘d’是‘1’结果在‘R’中存放’。



指令执行周期  
1

### RRR Rotate Right f through Carry

**语法**  
操作数  
操作内容  
受影响的标志  
说明

RRR R, d  
 $0 \leq R \leq 63$   
 $d \in [0,1]$   
 $C \rightarrow \text{dest} \langle 7 \rangle$ ;  
 $R \langle 7:1 \rangle \rightarrow \text{dest} \langle 6:0 \rangle$ ;  
 $R \langle 0 \rangle \rightarrow C$   
 C  
 R寄存器的内含值又移1-bit，右移时包含C(进位标志)，如下图，结果存放由‘d’决定，如果‘d’是0结果在ACC中存放。如果‘d’是‘1’结果在‘R’中存放’。



指令执行周期  
1

### SLEEP SLEEP

**语法**  
操作数  
操作内容  
受影响的标志  
说明  
指令执行周期

SLEEP  
 无  
 $00h \rightarrow \text{WDT}$ ;  
 $00h \rightarrow \text{WDT prescaler}$ ;  
 $1 \rightarrow \text{TO}$ ;  
 $0 \rightarrow \text{PD}$   
 TO, PD  
 TO位置1。PD位清零，WDT和WDT预置器清零  
 单片机进入睡眠模式  
 1

### SBCAR (带借位加法) Subtract ACC from R with Carry

**语法**  
操作数  
操作内容  
受影响的标志  
说明  
指令执行周期

SBCAR R, d  
 $0 \leq R \leq 63$   
 $d \in [0,1]$   
 $(R - \text{ACC} - C) \rightarrow \text{dest}$   
 C, DC, Z  
 将R寄存器的内含值减去A寄存器的内含值(带借位)，如果‘d’是0结果在ACC中存放。如果‘d’是‘1’结果在‘R’中存放’。  
 1

### SUBAR Subtract ACC from R

语法	SUBAR R, d
操作数	$0 \leq R \leq 63$ $d \in [0,1]$
操作内容	$R - ACC \rightarrow dest$
受影响的标志	C, DC, Z
说明	将R寄存器的内含值减去A寄存器的内含值（不带借位），如果‘d’是0结果在ACC中存放。如果‘d’是‘1’结果在‘R’中存放’。
指令执行周期	1

### SUBIA Subtract ACC from Immediate

语法	SUBIA I
操作数	$0 \leq I \leq 255$
操作内容	$ACC - I \rightarrow ACC$
受影响的标志	C, DC, Z
说明	将A寄存器的内含值减去立即数‘I’，结果在ACC中存放。
指令执行周期	1

### SWAPR Swap nibbles in R

语法	SWAPR R, d
操作数	$0 \leq R \leq 63$ $d \in [0,1]$
操作内容	$R<3:0> \rightarrow dest<7:4>$ ; $R<7:4> \rightarrow dest<3:0>$
受影响的标志	无
说明	将所选定的寄存器，高4位以及低4位，如果‘d’是0结果在ACC中存放。如果‘d’是1结果在‘R’中存放’。
指令执行周期	1

### XORAR Exclusive OR ACC with R

语法	SWAPR R, d
操作数	$0 \leq R \leq 63$ $d \in [0,1]$
操作内容	$ACC \text{ xor } R \rightarrow dest R$
受影响的标志	Z
说明	将A寄存器的值和R寄存器的值XOR在一起，如果‘d’是0结果在ACC中存放。如果‘d’是‘1’结果在‘R’中存放’。
指令执行周期	1

### XORIA Exclusive OR Immediate with ACC

语法	XORIA I
操作数	$0 \leq I \leq 255$
操作内容	$ACC \text{ xor } I \rightarrow ACC$
受影响的标志	Z
说明	将A寄存器的值和立即数‘I’ XOR在一起，结果在ACC中存放。
指令执行周期	1

## 4.0 绝对最大额定值

操作环境温度	0°C到+70°C
存储器额定温度	-65°C到+150°C
DC 电源电压(Vdd)	0V 到+6.0V
输入电压(对地电压 (Vss))	-0.3V 到(Vdd + 0.3)V

## 5.0 操作条件

DC 供电电压	+2.0V到+5.5V
操作温度	0°C到+70°C

\*细节详见 6.1

## 6.0 电气特性

### 6.1 SM152 电气特性

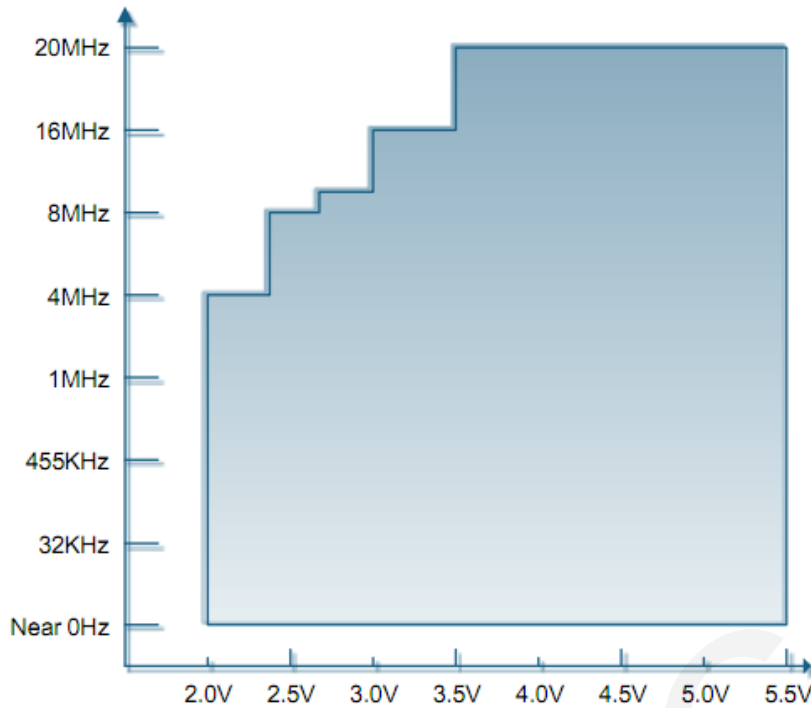
Ta=25°C

电气特性是在四时钟指令周期和 WDT & LVDT 禁用情况下

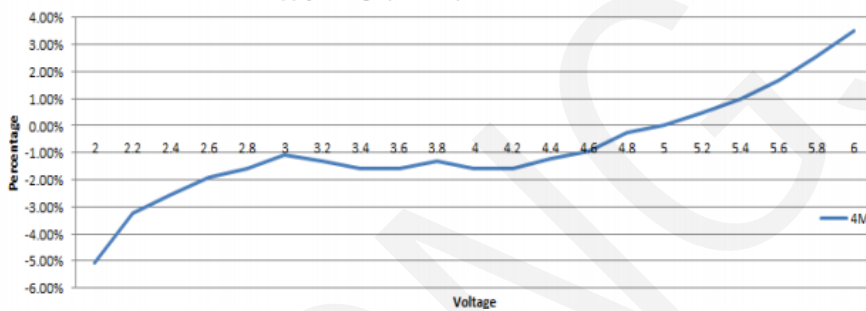
Sym	Description	Conditions	Min.	Typ.	Max.	Unit
Firc	Oscillation range	IRC mode, internal R, Vdd=5V	0.455		8	
		IRC mode, internal R, Vdd=3V	0.455		8	
VIH	Input high voltage	I/O ports, Vdd=5V	2.0			V
		RSTB, T0CKI pins, Vdd=5V	2.0			
		I/O ports, Vdd=3V	1.5			
		RSTB, T0CKI pins, Vdd=3V	1.5			
VIL	Input low voltage	I/O ports, Vdd=5V			1.0	V
		RSTB, T0CKI pins, Vdd=5V			1.0	
		I/O ports, Vdd=3V			0.6	
		RSTB, T0CKI pins, Vdd=3V			0.6	
VOH	Output high voltage	I <sub>OH</sub> =-5.4mA, Vdd=5V	3.6			V
VOL	Output low voltage	I <sub>OL</sub> =8.7mA, Vdd=5V			0.6	V
IPH	Pull-high current	Input pin at Vss, Vdd=5V		-65		uA
IPD	Pull-down current	Input pin at Vdd, Vdd=5V		45		uA
I <sub>WDT</sub>	WDT current	Vdd=5V		9	12	uA
		Vdd=3V		2	4	
T <sub>WDT</sub>	WDT period	Vdd=3V		20.4		mS
		Vdd=4V		17.9		
		Vdd=5V		16.2		
I <sub>LVDT</sub>	LVDT current	Vdd=5V LVDT = 3.6V		30	40	uA
		Vdd=5V LVDT = 2V		23	30	
		Vdd=3V LVDT = 2V		6.8	8.0	
I <sub>SB</sub>	Power down current	Sleep mode, Vdd=5V, WDT enable		20		uA
		Sleep mode, Vdd=5V, WDT disable		3		
		Sleep mode, Vdd=3V, WDT enable		2.5		
		Sleep mode, Vdd=3V, WDT disable		1.1	Sleep	
		F=1MHz				
		F=455KHz				
I <sub>DD</sub>	Operating current	IRC mode, internal R, Vdd=3V, 2 clock instruction				mA
		F=8MHz				
		F=4MHz				
		F=1MHz				
		F=455KHz				

## 6.2 SM152电气特性表

### 6.2.1 操作频率 vs 操作电压 (Ta=25°C)

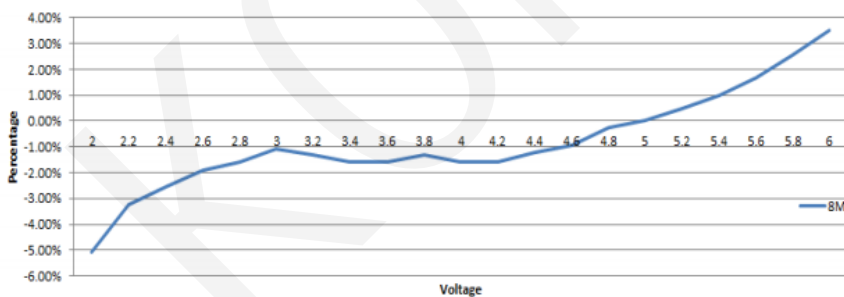


### 6.2.2 内部 4MHz RC vs 供应电压 (Ta=25°C)



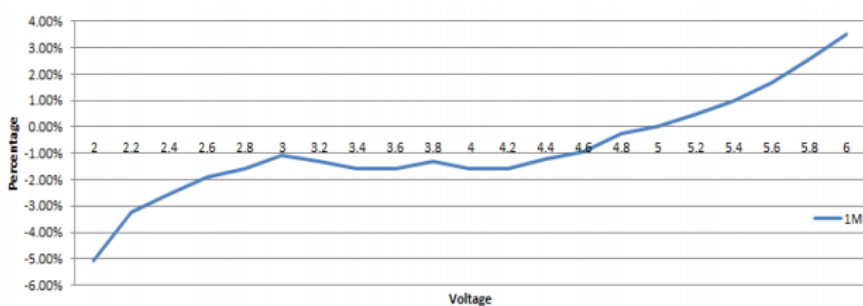
注：曲线仅供设计参考

### 6.2.3 内部 8MHz RC vs 供应电压 (Ta=25°C)



注：曲线仅供设计参考

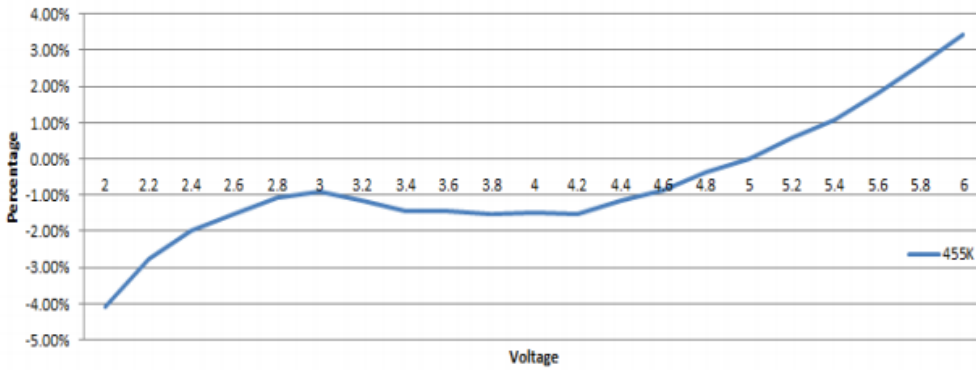
### 6.2.4 内部 1MHz RC vs 供应电压 (Ta=25°C)



注：曲线仅供设计参考

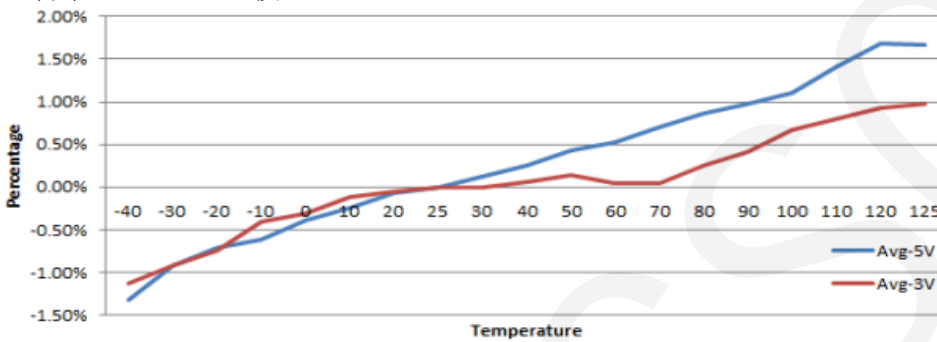


### 6.2.5 内部 455KHz RC vs 供应电压 (Ta=25°C)



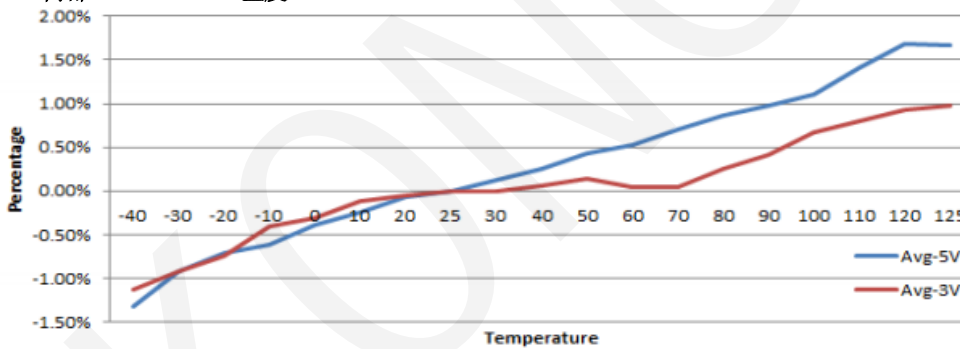
注：曲线仅供设计参考

### 6.2.6 内部 4MHz RC vs 温度



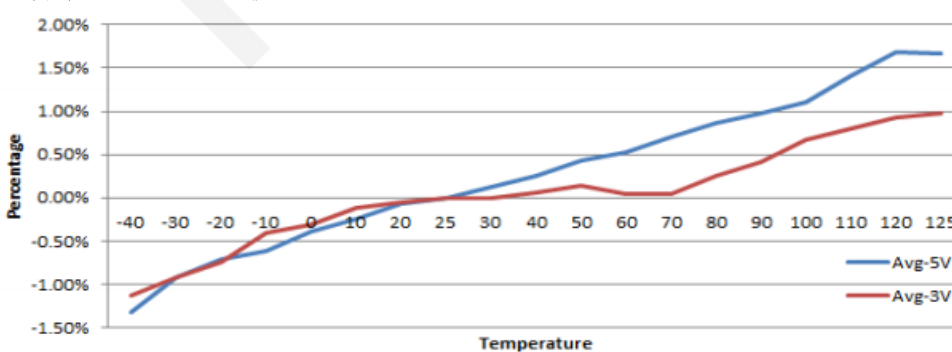
注：曲线仅供设计参考

### 6.2.7 内部 8MHz RC vs 温度



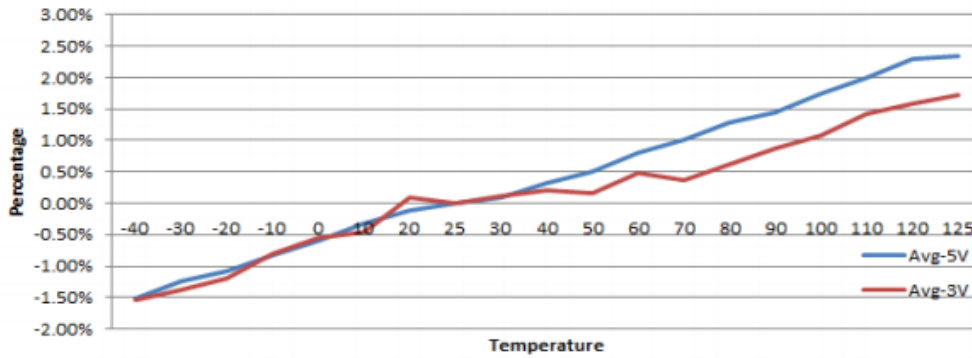
注：曲线仅供设计参考

### 6.2.8 内部 1MHz RC vs 温度



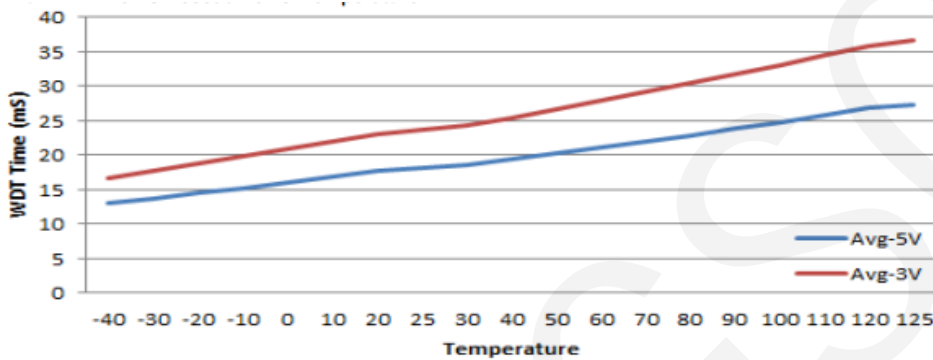
注：曲线仅供设计参考

### 6.2.9 内部 455KH Hz RC vs 温度



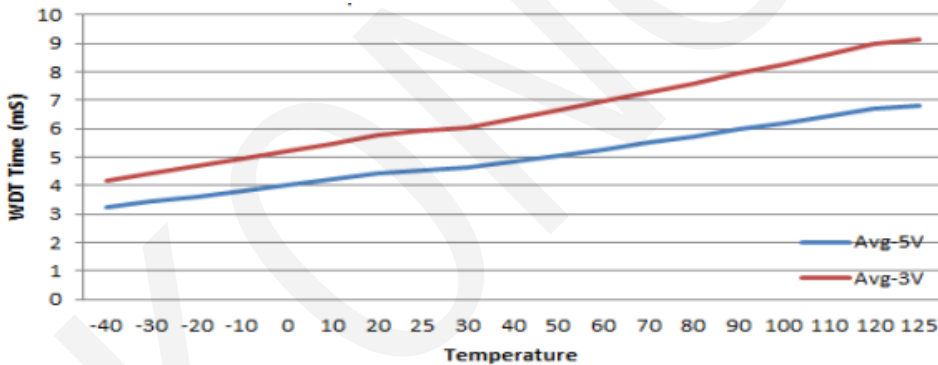
注：曲线仅供设计参考

### 6.2.10 WTD18 毫秒复位时间 vs 温度



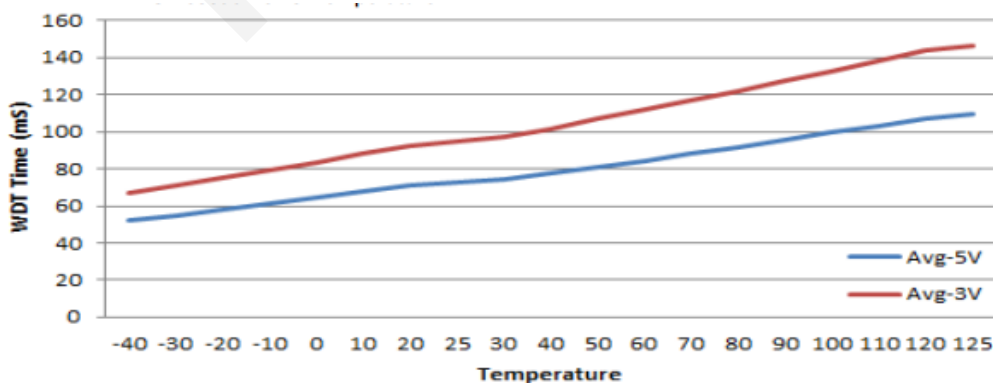
注：曲线仅供设计参考

### 6.2.11 WTD4.5 毫秒复位时间 vs 温度



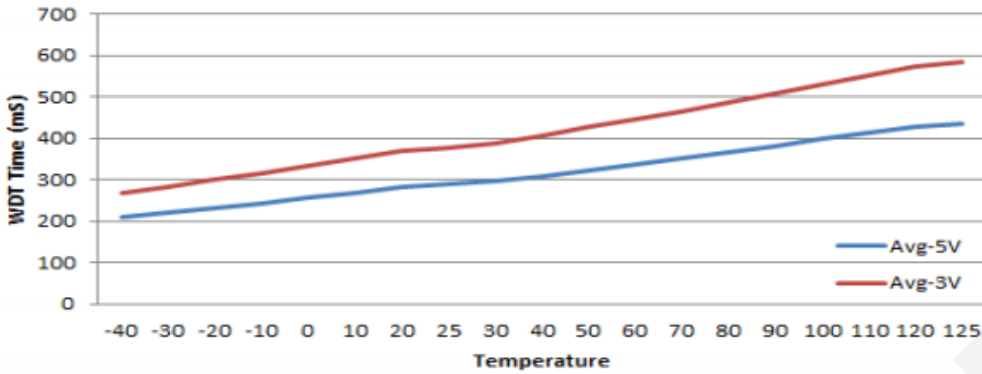
注：曲线仅供设计参考

### 6.2.12 WTD72 毫秒复位时间 vs 温度



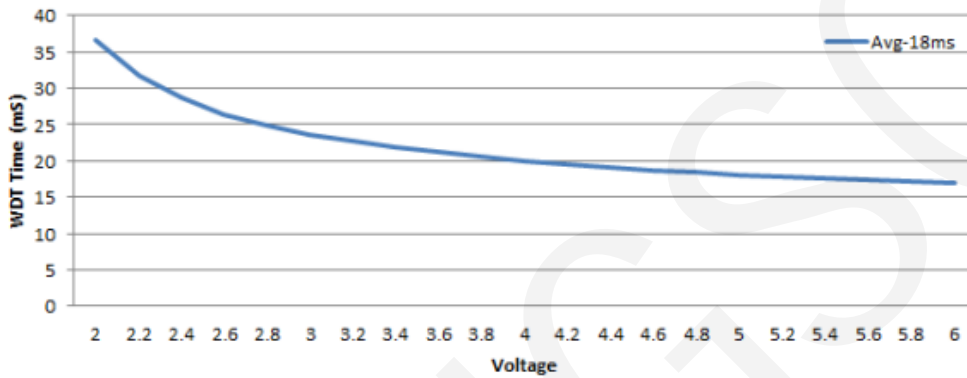
注：曲线仅供设计参考

### 6.2.13 WTD288 毫秒复位时间 vs 温度



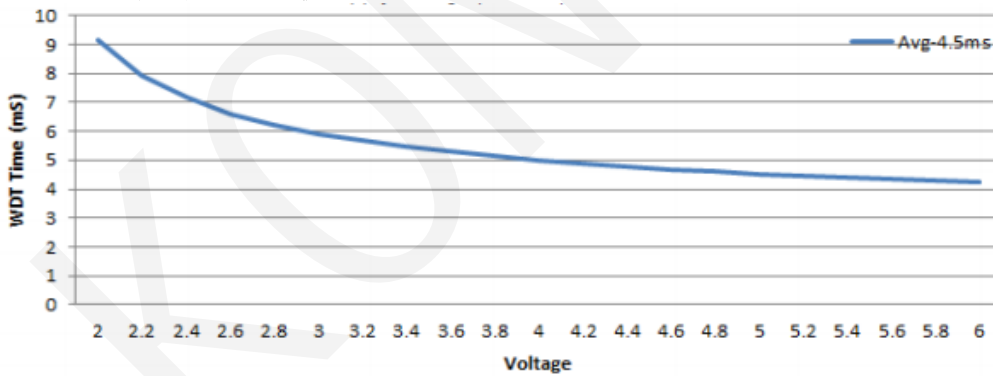
注：曲线仅供设计参考

### 6.2.14 WTD18 毫秒复位时间 vs 供应电压 (Ta=25°C)



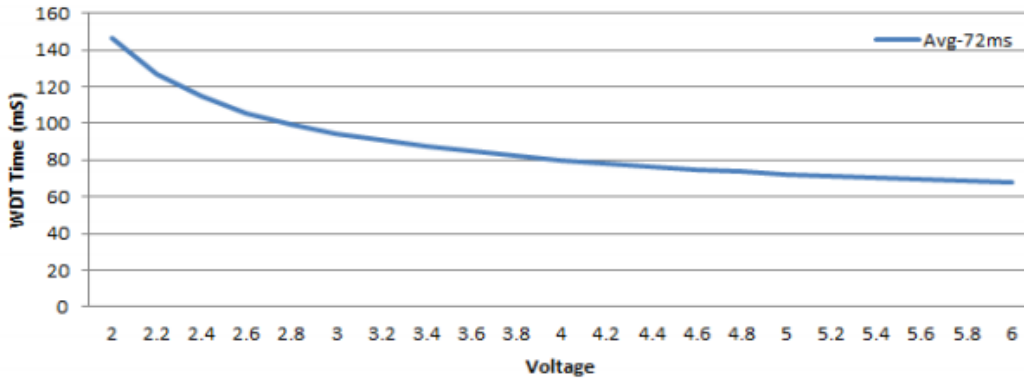
注：曲线仅供设计参考

### 6.2.15 WTD4.5 毫秒复位时间 vs 供应电压 (Ta=25°C)



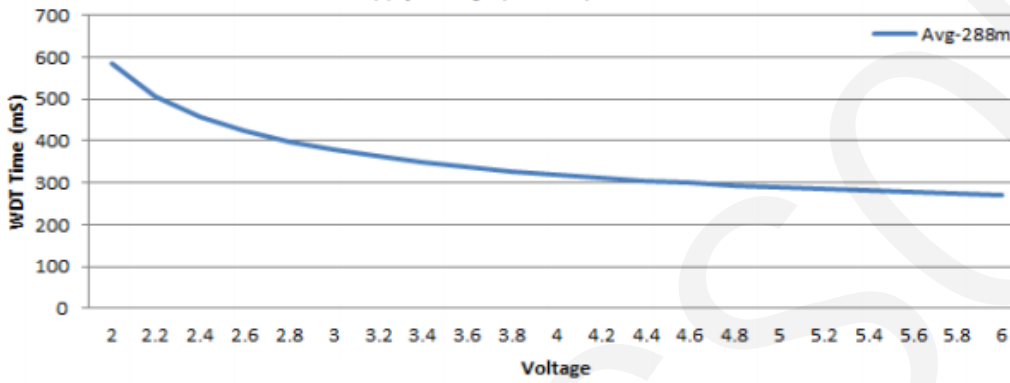
注：曲线仅供设计参考

### 6.2.16 WTD72 毫秒复位时间 vs 供应电压 (Ta=25°C)



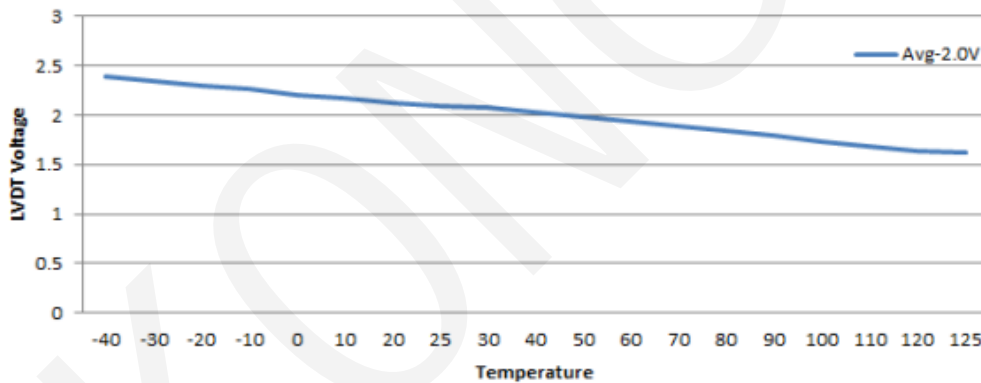
注：曲线仅供设计参考

### 6.2.17 WTD288 毫秒复位时间 vs 供应电压 (Ta=25°C)



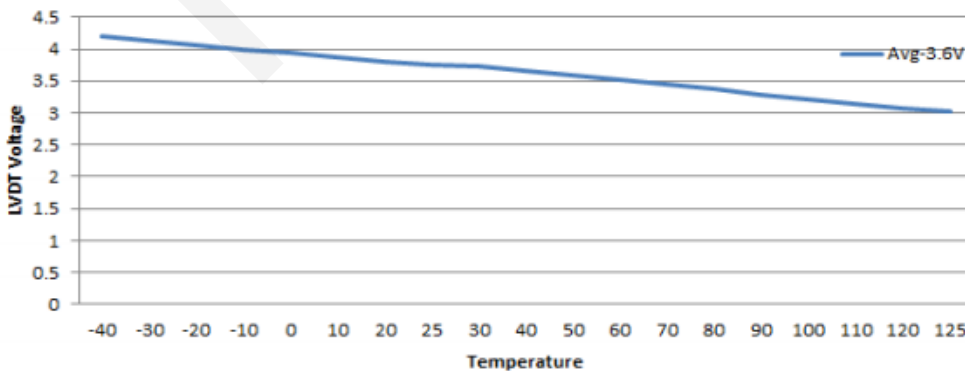
注：曲线仅供设计参考

### 6.2.18 LVDT2.0V vs 温度



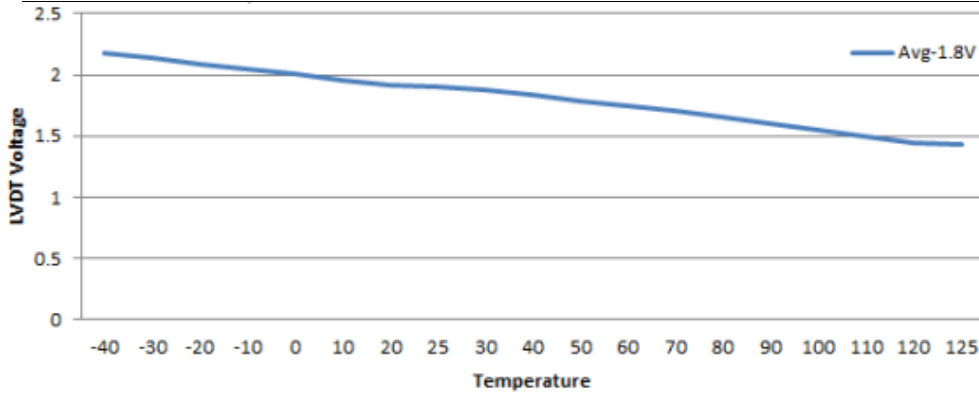
注：曲线仅供设计参考

### 6.2.19 LVDT3.6V vs 温度



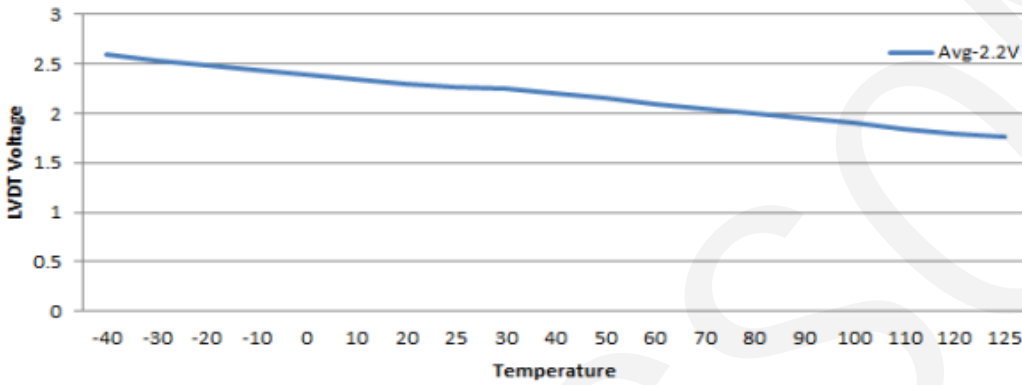
注：曲线仅供设计参考

### 6.2.20 LVDT1.8V vs 温度



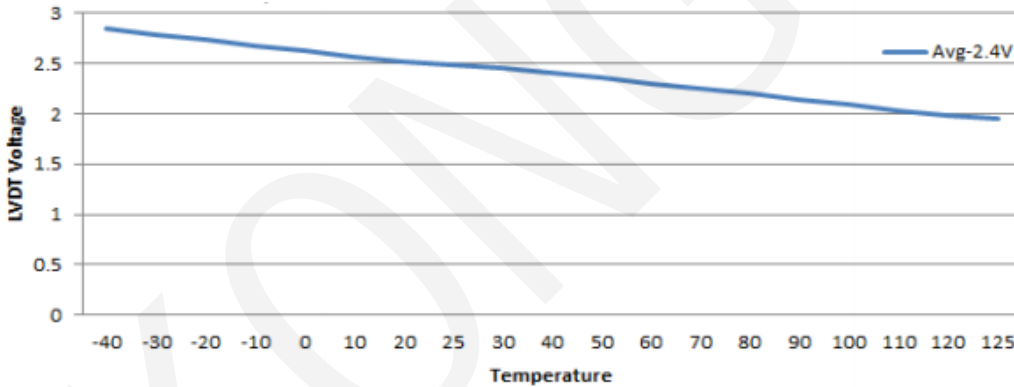
注：曲线仅供设计参考

### 6.2.21 LVDT2.2V vs 温度



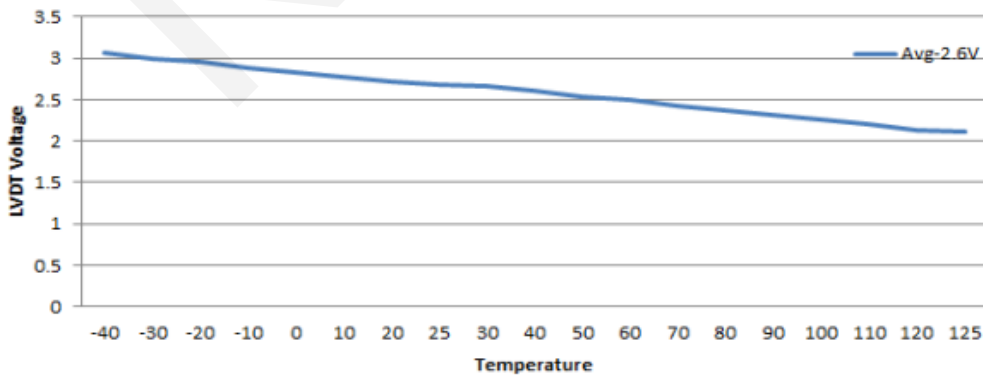
注：曲线仅供设计参考

### 6.2.22 LVDT2.4V vs 温度



注：曲线仅供设计参考

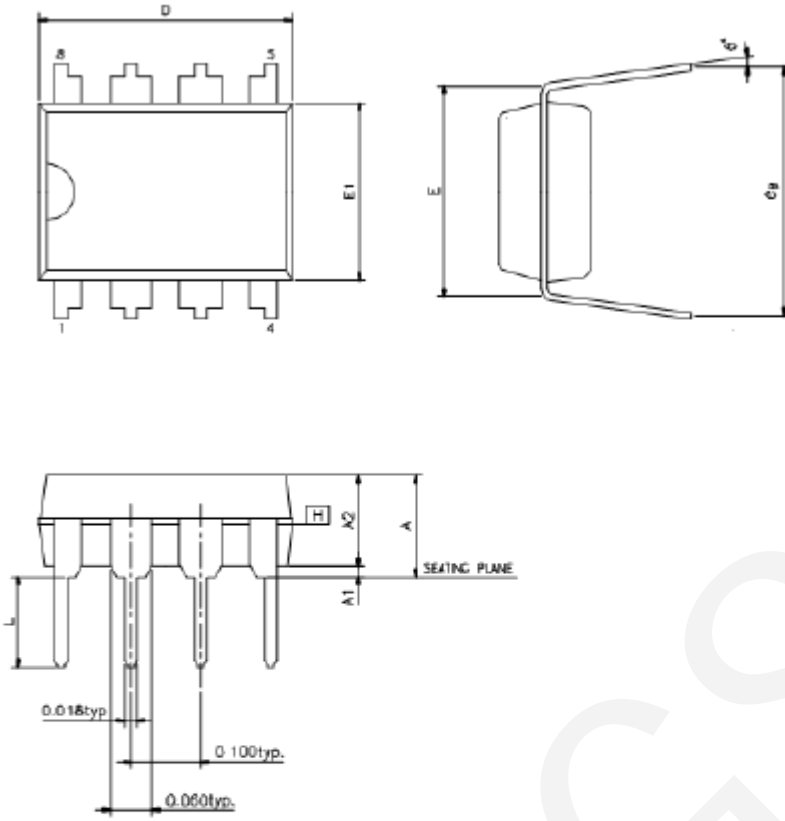
### 6.2.23 LVDT2.6V vs 温度



注：曲线仅供设计参考

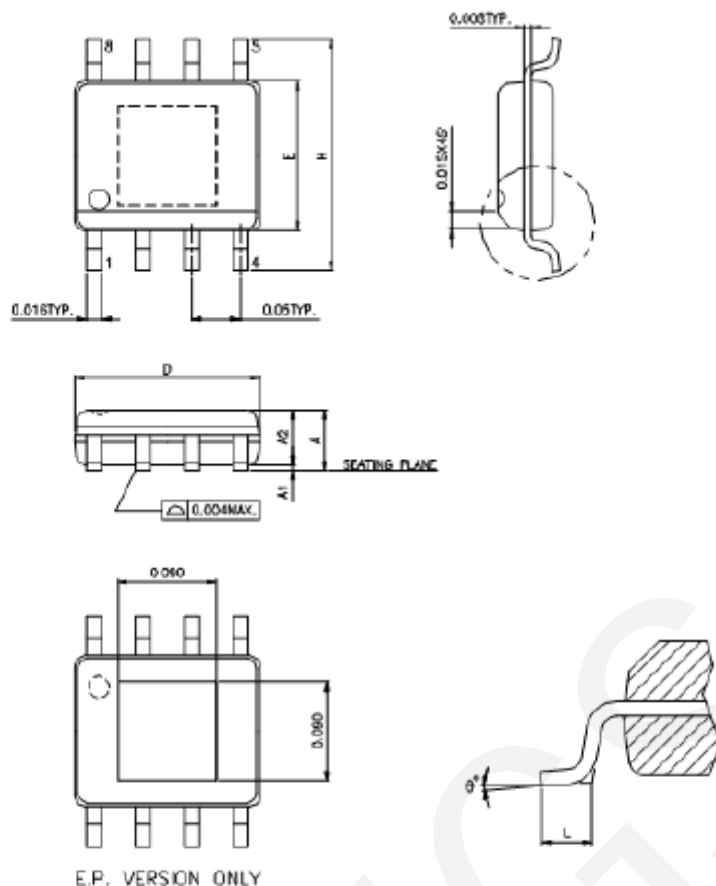
## 7.0 封装尺寸

### 7.1 PDIP 8 PIN



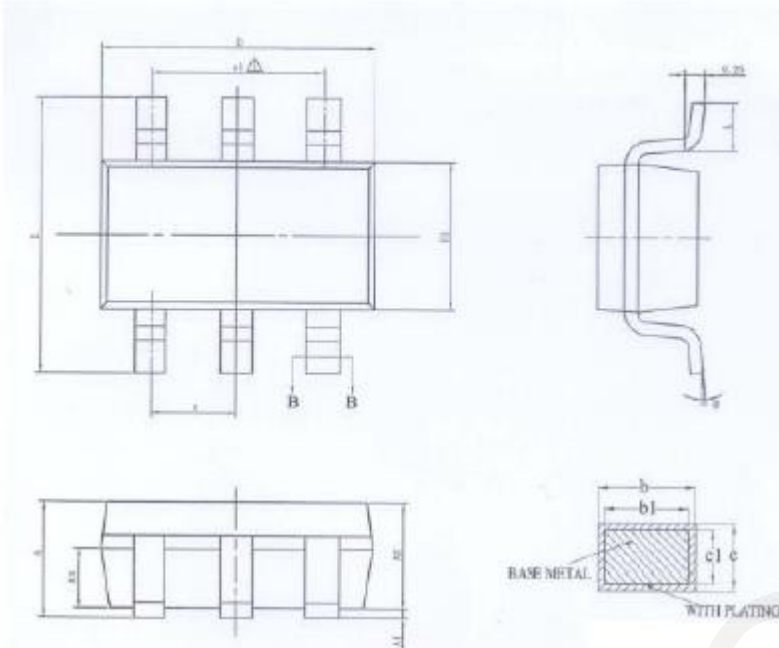
SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	-	-	0.210	-	-	5.334
A1	0.015	-	-	0.381	-	-
A2	0.125	0.130	0.135	3.175	3.302	3.429
D	0.355	0.365	0.400	9.017	9.271	10.16
E	0.300			7.62		
E1	0.245	0.250	0.255	6.223	6.35	6.477
L	0.115	0.130	0.150	2.921	3.302	3.810
eB	0.335	0.355	0.375	8.509	9.017	9.525
θ°	0°	7°	15°	0°	7°	15°

## 7.2 SOP 8 PIN



SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	0.053	-	0.069	1.346	-	1.753
A1	0.004	-	0.010	0.102	-	0.254
A2	-	-	0.059	-	-	1.498
D	0.189	-	0.196	4.800	-	4.978
E	0.150	-	0.157	3.810	-	3.988
H	0.228	-	0.244	5.791	-	6.198
L	0.016	-	0.050	0.406	-	1.270
$\theta^\circ$	0°	-	8°	0°	-	8°

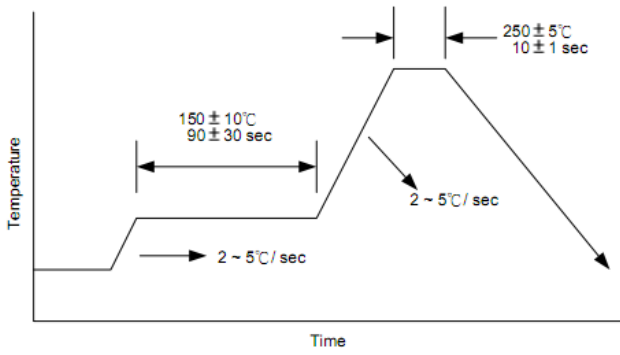
### 7.3 SOT23-6 PIN



SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	-	-	0.053	-	-	1.35
A1	0.002	-	0.006	0.04	-	0.15
A2	0.039	0.043	0.047	1.00	1.10	1.20
A3	0.022	0.026	0.030	0.55	0.65	0.75
b	0.012	-	0.020	0.30	-	0.50
b1	0.012	0.016	0.018	0.30	0.40	0.45
c	0.003	-	0.009	0.08	-	0.22
c1	0.003	0.005	0.008	0.08	0.13	0.20
D	0.107	0.115	0.123	2.72	2.92	3.12
E	0.102	0.110	0.118	2.60	2.80	3.00
E1	0.055	0.063	0.071	1.40	1.60	1.80
e	0.037			0.95		
e1	0.075			1.90		
L	0.012	-	0.024	0.30	-	0.60
θ°	0°	-	8°	0°	-	8°



## 8.0 包装 IR 回流焊接曲线



KONGSON