

## 目录

目录.....	1
1 总体描述 .....	5
2 主要特色 .....	5
3 管脚定义 .....	6
3.1 管脚配置 .....	6
3.2 管脚定义 .....	7
4 内部方框图 .....	9
5 FLASH ROM 和 SRAM 结构.....	10
5.1 flash rom .....	10
5.2 Code Option 区域（用户烧写设置） .....	11
5.3 sram.....	11
6 特殊功能寄存器(SFR) .....	13
6.1 SFR 映像 .....	13
6.2 SFR 说明 .....	13
7 电源、复位和时钟 .....	15
7.1 电源电路 .....	15
7.2 上电复位过程 .....	15
7.3 复位方式 .....	16
7.3.1 外部 RST 复位 .....	16
7.3.2 低电压复位 LVR.....	16
7.3.3 上电复位 POR.....	17
7.3.4 看门狗复位 WDT .....	17
7.3.5 复位初始状态 .....	18
7.4 时钟电路 .....	19
7.5 省电模式 STOP.....	19
8 中央处理单元 CPU 及指令系统.....	20
8.1 CPU .....	20

8.2 寻址方式 .....	20
8.2.1 立即寻址 .....	20
8.2.2 直接寻址 .....	20
8.2.3 间接寻址 .....	20
8.2.4 寄存器寻址 .....	20
8.2.5 相对寻址 .....	20
8.2.6 变址寻址 .....	20
8.2.7 位寻址 .....	21
8.3 指令系统.....	21
<b>9 INTERRUPT 中断 .....</b>	<b>23</b>
9.1 中断源、向量.....	23
9.2 中断结构图.....	25
9.3 中断优先级 .....	25
9.4 中断处理流程.....	26
9.5 中断相关 SFR 寄存器.....	26
<b>10 定时器 TIMER0、TIMER1 .....</b>	<b>28</b>
10.1 T0 和 T1 相关特殊功能寄存器 .....	28
10.2 T0 工作模式.....	30
10.3 T1 工作模式.....	31
<b>11 PWM .....</b>	<b>33</b>
11.1 PWM 结构框图.....	34
11.2 PWM 相关 SFR 寄存器.....	35
11.3 PWM 波形及用法.....	37
11.4 两路互补带死区 PWM 实现方法.....	39
11.5 PWM 频率高精度步进调整实现方法.....	42
<b>12 GP I/O .....</b>	<b>43</b>
12.1 GPIO 结构图 .....	44
12.2 I/O 端口相关寄存器 .....	46
12.3 I/O 端口复用 .....	47

<b>13 TOUCH KEY SENSOR</b> .....	<b>48</b>
13.1 Touch Key 原理.....	48
13.2 Touch Key 相关寄存器.....	48
13.3 Touch Key 使用方法.....	52
<b>14 软件 UART</b> .....	<b>53</b>
<b>15 串行接口 (SIF)</b> .....	<b>53</b>
15.1 SIF 相关 SFR 寄存器.....	55
15.2 SIF 应用注意事项.....	60
<b>16 模数转换 ADC</b> .....	<b>61</b>
16.1 ADC 相关寄存器.....	61
16.2 ADC 转换步骤.....	63
<b>17 IAP 操作</b> .....	<b>63</b>
17.1 IAP 操作相关寄存器.....	63
17.2 IAP 操作流程.....	64
17.3 IAP 读写范例程序.....	64
<b>18 电气特性</b> .....	<b>66</b>
18.1 极限参数.....	66
18.2 推荐工作条件.....	66
18.3 直流电气特性.....	66
18.4 交流电气特性 (VDD = 2.5V ~ 5.5V, TA = 25°C, 除非另有说明).....	67
18.5 ADC 电气特性 (TA = 25°C, 除非另有说明).....	67
18.6 ADC 实测曲线图.....	68
<b>19 订购信息</b> .....	<b>69</b>
<b>20 封装信息</b> .....	<b>70</b>

---

21 规格更改记录 .....	71
-----------------	----

## 1 总体描述

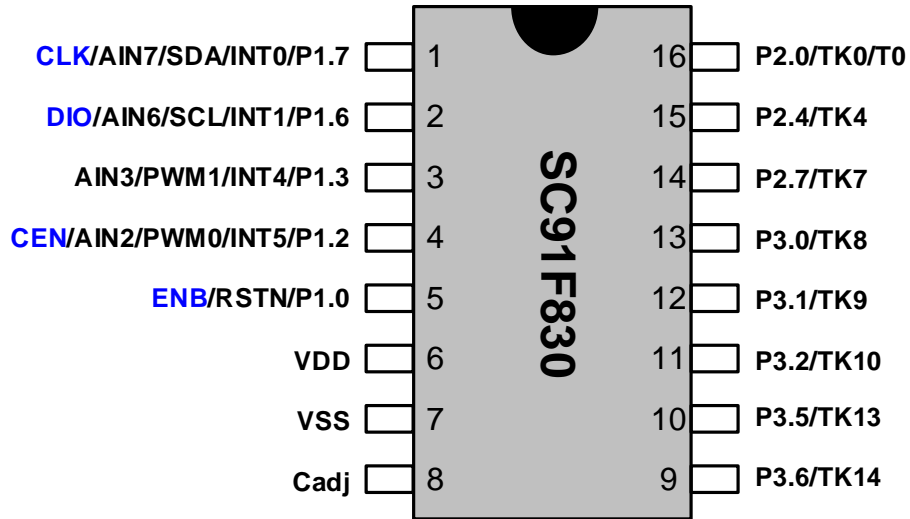
SC91F830 是一颗内置电容型触摸按键功能的加强型超快速 1T 8051 工业级 Flash 微控制器，指令系统完全兼容传统 8051 产品系列。SC91F830 内部集成有最多 8 路触摸按键电路，其它资源还包括：8KB Flash ROM（内部 256Byte 可作为 EEPROM）、512B SRAM、最多 13 个 GP I/O(包含 8 个大电流驱动)、2 个 16 位定时器、1 路类 IIC 的串行通讯接口 SIF、最多 4 路 10 位高精度 ADC、最多 4 路外部中断口（其中 1 路是双沿中断）、2 路 8 位 PWM、内部 1%高精度 16M Hz 振荡器、软件 UART 等资源。为提高可靠性及简化客户电路，SC91F830 内部也集成有 4 级可选电压 LVR、2.4V 基准 ADC 参考电压、WDT 等高可靠电源电路。SC91F830 具有非常优异的抗干扰性能和抗 EMI 能力，非常适合应用于电磁炉、抽油烟机、消毒柜、电饭煲、电压力锅、面包机等各种小家电和卫浴、灯具等工业控制和消费应用领域。

## 2 主要特色

- 工作电压：2.4V~5.5V
- 工作温度：-40 ~ 85 度
- 封装：SOP16L
- 内核：超快速的 1T 8051
- 存储器：8KB Flash ROM（MOVC 禁止寻址 0000H~00FFH），其中包含 256B 可 IAP  
512B SRAM
- 系统时钟：内建 16M Hz 振荡器
  - IC 工作的系统时钟,可通过编程器选择设定为 4MHz
  - 频率误差: 跨越 (4.5V~5.5V) 及 (-20 度, 85 度) 应用环境, 不超过  $\pm 1\%$
  - 用户可通过修改 SFR 在工作中改变系统时钟
- 低电压复位 (LVR) :
  - 复位电压有 4 级可选：分别是: 3.65V、3.50V、2.60V、2.45V
  - 缺省值为用户烧写 Code Option 所选值
- Flash 烧写：4 线串口烧写接口
- 中断 (INT) :
  - TIMER0, TIMER1, INT0/1/4/5, ADC, PWM、SIF、TKS 共 10 个中断源
  - INT0/4/5 为 3 个中断向量入口，下降沿触发
  - INT1 为单独的中断向量入口，可设上升沿、下降沿、双沿中断
  - 两级中断优先级可设
- 数字外围：
  - 最多 8 路 Touch Key 电路
  - 13 个双向强输出 GP I/O，4 种模式可设（P1/P2 口有大灌电流能力）
  - 16 位 WDT，可选时钟分频比
  - 2 个标准 80C51 16 位定时器 TIMER0 及 TIMER1
  - 2 路共用周期、单独可调占空比的 8 位 PWM
  - 1 路类 IIC 的串行通讯接口 SIF
  - 软件 UART(最高 56000 波特率@16M)
- 模拟外围：
  - 4 路 10 位 ADC
    - 1) 内建基准的 2.4V 参考电压
    - 2) ADC 的参考电压有 2 种选择, 分别是 VDD 和 内部 2.4V
    - 3) 可设 ADC 转换完成中断
- 省电模式：
  - STOP MODE (也称 POWER DOWN MODE)
  - 可由 INT0/1/4/5 或者外部 RSTN 唤醒 STOP MODE

### 3 管脚定义

#### 3.1 管脚配置



SC91F830管脚配置图

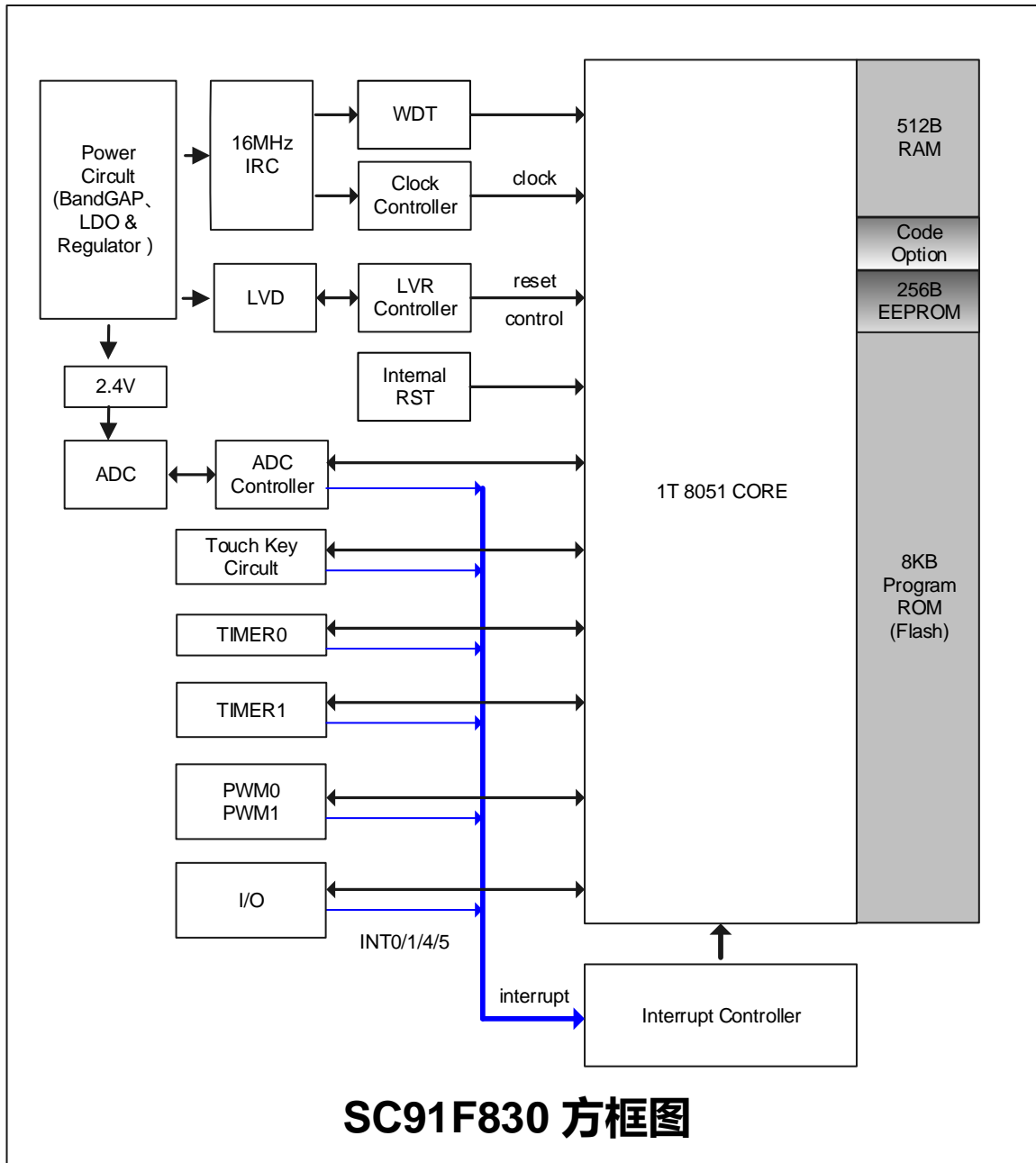
### 3.2 管脚定义

管脚编号	管脚名称	管脚类型	功能说明
1	<b>CLK/AIN7/SDA/INT0/P1.7</b>	I/O	1) P1.7 : GPIO P1.7, 大灌电流口 2) INT0 : 外部中断 INT0 3) SDA : SIF 的 DATA 口 4) AIN7 : ADC 输入通道 7 <b>CLK: 编程口 CLK</b>
2	<b>DIO/AIN6/SCL/INT1/P1.6</b>	I/O	1) P1.6 : GPIO P1.6, 大灌电流口 2) <b>INT1</b> : 双沿外部中断 INT1, 可设为上升、下降、双沿中断 3) SCL : SIF 的 CLK 口 4) AIN6 : ADC 输入通道 6 <b>DIO: 编程口 DIO</b>
3	<b>AIN3/PWM1/INT4/P1.3</b>	I/O	1) P1.3 : GPIO P1.3, 大灌电流口 2) INT4 : 外部中断 INT4 3) PWM1 : PWM1 的输出口 4) AIN3 : ADC 输入通道 3
4	<b>CEN/AIN2/PWM0/INT5/P1.2</b>	I/O	1) P1.2 : GPIO P1.2, 大灌电流口 2) INT5 : 外部中断 INT5 3) PWM0 : PWM0 的输出口 4) AIN2 : ADC 输入通道 2 <b>CEN: 编程口 CEN</b>
5	<b>ENB/RSTN/P1.0</b>	I/O	1) RST : RESET 管脚(Default), 低电平使能。用户电路不能在上电时强制拉低(上电复位时, 系统默认为 RST, 复位后可通过设置 SFR (RSTCFG) 取消 RESET 功能并将此 Pin 设为 IO。 2) P1.0 : GPIO P1.0 <b>ENB: 编程口 ENB</b>
6	<b>VDD</b>	Power	电源
7	<b>VSS</b>	Power	接地
8	<b>Cadj</b>	I	Touch Key 灵敏度调节电容 建议接 1.0~10nF 低温漂涤纶电容 电容大小取决于用户 Layout 及触摸面板介质的材质和厚度, 用户通过内部 TK 参数合理设置后 TKCNT 能在 1000~4000 为宜 ( <u>相当于每个按键单次扫描时长为 62.5~250us@16Mhz</u> 再加上放电时长的设置 4~32us)。

<b>9</b>	<b>TK14/P3.6</b>	I/O	1) P3.6 : GPIO P3.6 2) TK14 : TK 的通道 14
<b>10</b>	<b>TK13/P3.5</b>	I/O	1) P3.5 : GPIO P3.5 2) TK13 : TK 的通道 13
<b>11</b>	<b>TK10/P3.2</b>	I/O	1) P3.2 : GPIO P3.2 2) TK10 : TK 的通道 10
<b>12</b>	<b>TK9/P3.1</b>	I/O	1) P3.1 : GPIO P3.1 2) TK9 : TK 的通道 9
<b>13</b>	<b>TK8/P3.0</b>	I/O	1) P3.0 : GPIO P3.0 2) TK8 : TK 的通道 8
<b>14</b>	<b>TK7/P2.7</b>	I/O	1) P2.7 : GPIO P2.7 (大灌电流口) 2) TK7 : TK 的通道 7
<b>15</b>	<b>TK4/P2.4</b>	I/O	1) P2.4 : GPIO P2.4 (大灌电流口) 2) TK4 : TK 的通道 4
<b>16</b>	<b>TK0/P2.0</b>	I/O	1) P2.0 : GPIO P2.0 (大灌电流口) 2) TK0 : TK 的通道 0

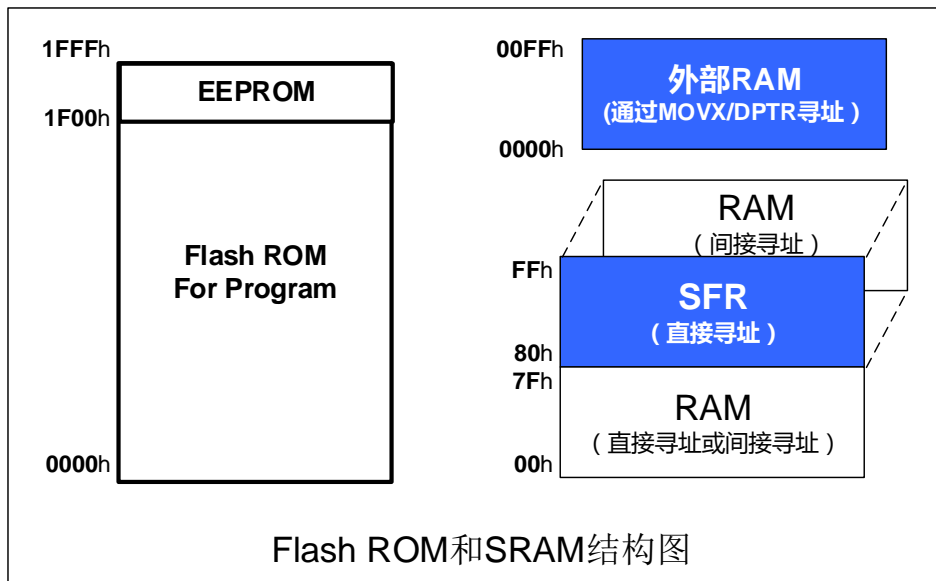


## 4 内部方框图



## 5 FLASH ROM 和 SRAM 结构

SC91F830 的 Flash ROM 和 SRAM 结构如下:

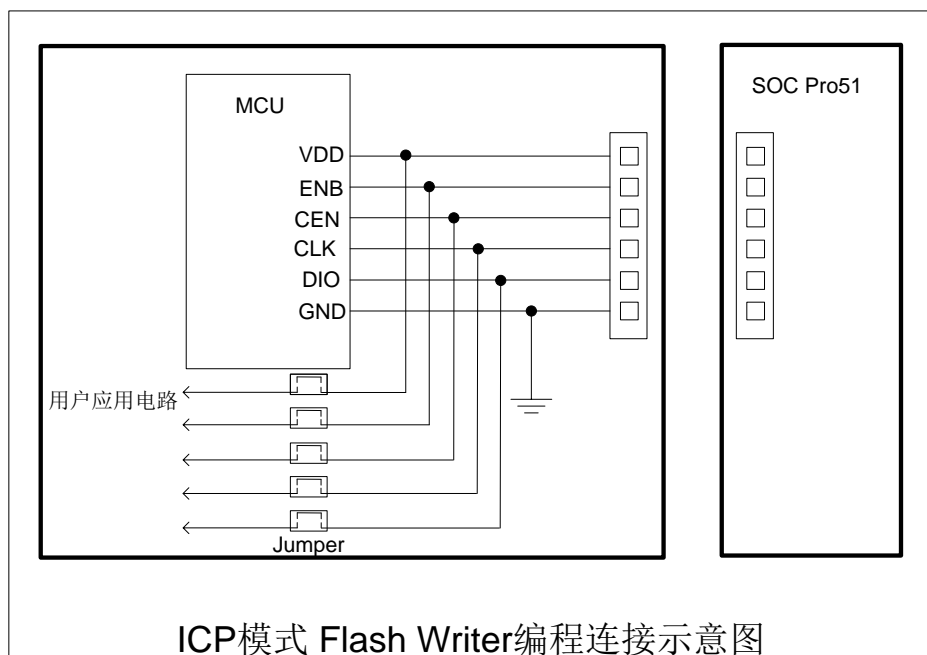


### 5.1 FLASH ROM

SC91F830 有 8KB 的 Flash ROM, ROM 地址为 0000H~1FFFFH, 其中地址为 1F00H~1FFFFH 的 256Byte Flash 可以作为 EEPROM 使用 (即支持用户在程式中擦写, 详细操作请看 IAP 章节)。此 8KB Flash ROM 可反复擦写 10 万次, 可通过 SinOneChip 提供的专用 ICP 烧写器(SOC Pro51/DPT51)来进行编程及擦除。地址为 0000H~00FFH 地址的 256B 区间 MOVX 指令不可寻址。

SC91F830 的 8KB Flash ROM 能提供查空 BLANK、编程 PROGRAM、校验 VERIFY 和擦除 ERASE (此 IC 写入新的程序和数据不需要擦除) 功能, 但不提供读取 READ 的功能。

SC91F830 的 Flash ROM 通过 Pin5 (ENB)、Pin4 (CEN)、Pin1 (CLK)、Pin2 (DIO)、VDD、VSS 来进行编程, 具体连接关系如下:



## 5.2 CODE OPTION 区域（用户烧写设置）

SC91F830 内部有单独的一块 Flash 区域用于保存客户的上电初始值设置，此区域称为 Code Option 区域。用户在烧写 IC 时将此部分代码写入 IC 内部，IC 在复位初始化时，就会将此设置调入 SFR 作为初始设置。

IFB	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
<b>IFB1</b>	-	-	-	DISLVR	LVRS[3:0]			
<b>IFB2</b>	-	-	-	ENWDT	-	Vrefs	-	IRCFS

IFB1 编号	符号	说明
4	<b>DISLVR</b>	LVR 开关 0: LVR 有效 1: LVR 无效
3~0	<b>LVRS [3:0]</b>	LVR 电压选择控制 1011: 3.65V 复位 1010: 3.50V 复位 0101: 2.60V 复位 0100: 2.45V 复位 此电压点的值为常温值，实际值随温度会发生一些变化(约±0.1V@-40~85℃)，具体表现为温度越高 LVR 电压点会下降，温度越低 LVR 电压点会抬高。

IFB2 编号	符号	说明
4	<b>ENWDT</b>	WDT 开关 0: WDT 无效 1: WDT 有效（但 IC 在执行 IAP 过程中 WDT 停止计数）
2	<b>Vrefs</b>	ADC 参考电压选择 0: 内部 VDD 1: 内部精准的 2.4V
0	<b>IRCFS</b>	IRC 频率选择控制 0: IRC 频率为 16MHz 1: IRC 频率为 4MHz

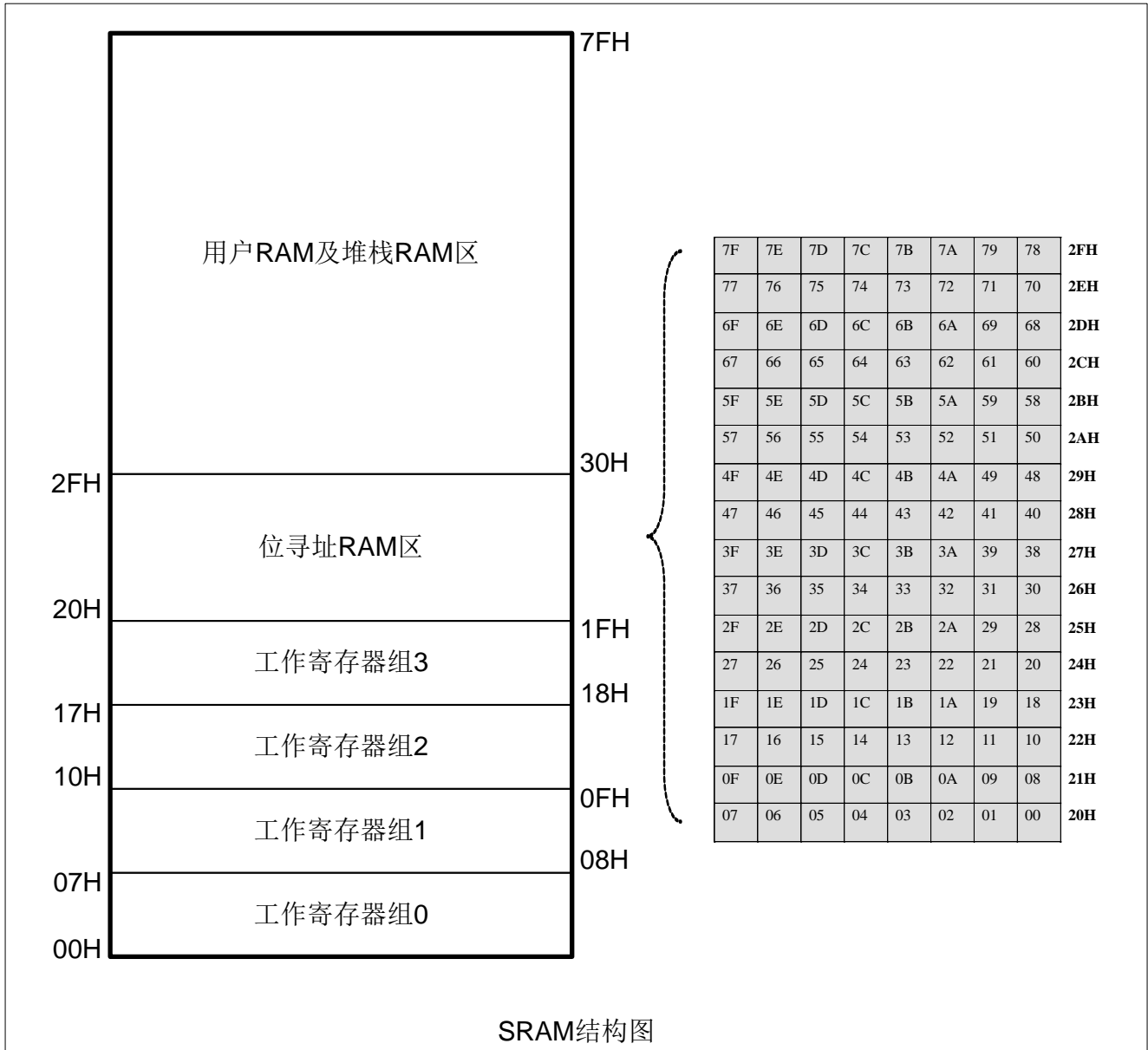
## 5.3 SRAM

SC91F830 单片机内部集成了 512B 的 SRAM，供用户使用，内部 SRAM 地址范围为 00H~FFH。其中高 128B（地址 80H~FFH）只能间接寻址，低 128B（地址 00H~7FH）可直接寻址也可间接寻址。

特殊功能寄存器 SFR 的地址也是 80H~FFH。但 SFR 同高 128B SRAM 的区别是：SFR 寄存器是直接寻址，而高 128B SRAM 只能是间接寻址。

### 内部 256B SRAM:

内部 SRAM 区可分为三部分：①工作寄存器组 0~3，地址 00H~1FH，程序状态字寄存器 PSW 中的 RS0、RS1 组合决定了当前使用的工作寄存器，使用工作寄存器组 0~3 可加快运算的速度；②位寻址区 20H~2FH，此区域用户可用作普通 RAM 也可用作按位寻址 RAM；按位寻址时，位的地址为 00H~7FH，（此地址按位编地址，不同于通用 SRAM 按字节编地址），程序中可由指令区分；③用户 RAM 和堆栈区，SC91F830 复位过后，8 位的堆栈指针指向堆栈区，用户一般会在初始化程序时设置初值，建议设置在 60H~7FH 的单元区间。



**外部 256B SRAM:**

外部 256B SRAM 支持传统的访问外部 RAM 方法。使用 MOVX A, @Ri 或 MOVX @Ri, A 来访问外部低位 256 字节 RAM; 也可以用 MOVX A, @DPTR 或 MOVX @DPTR, A 来访问外部 256 字节 RAM。

## 6 特殊功能寄存器(SFR)

### 6.1 SFR 映像

SC91F830 系列有一些特殊功能寄存器，我们称为 SFR。这些 SFR 寄存器的地址位于 80H~FFH，有些可以位寻址，有些不能位寻址。能够进行位寻址操作的寄存器的地址末位数都是“0”或“8”，这些寄存器在需要改变单个位的数值时非常方便。所有的 SFR 特殊功能寄存器都必须使用直接寻址方式寻址。

SC91F830 的特殊功能寄存器名称及地址如下表：

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8h	PWMCR	PWMPRD	PWMDTY1	PWMDTY0	PWMCFG	-	-	-
F0h	B	-	-	-			RSTCFG	-
E8h	TKCR	-	IAPKEY		IAPADL	IAPDAT	IAPCTL	IRCC
E0h	ACC	-	-	-	-	-	-	-
D8h	SIFSTA	-	-	-	-	-	-	-
D0h	PSW	-	-	-	SIFCFG	SIFCTL	SIFTXD	SIFRXD
C8h	-	WDTCR	-	-	-	-	-	-
C0h	-		-	-	ADCCFG	ADCCR	ADCVH	ADCVL
B8h	IP	-	-	-	-	-	-	-
B0h	P3	P3CFG1	P3CFG0		EXIE	EXIP		-
A8h	IE	-	-	-	-	-	-	-
A0h	P2	P2CFG1	P2CFG0	-	-	-		-
98h	-	-	TKCFG1	TKCFG2	TKCFG3	TKSTA	TKPVH	TKPVL
90h	P1	P1CFG1	P1CFG0	P1IT	-	-	TKCNTH	TKCNTL
88h	TCON	TMOD	TL0	TL1	TH0	TH1	TMCON	-
80h		SP	DPL	DPH	-	-	-	PCON
	可位寻址	不可位寻址						

说明：

1.SFR 寄存器中空的部分代表没有此寄存器 RAM，不建议用户使用。

2.SFR 中的 F7H、FFH 为系统配置使用的特殊功能寄存器，用户使用可能会导致系统异常，用户在初始化系统时，不能对此 2 个寄存器进行清零或其它操作。

### 6.2 SFR 说明

特殊功能寄存器 SFR 的具体解释说明如下：

符号	地址	说明	7	6	5	4	3	2	1	0	上电初始值
SP	81H	堆栈指针	SP[7:0]								00000111b
DPL	82H	DPTR 数据指针低位	DPL[7:0]								00000000b
DPH	83H	DPTR 数据指针高位	DPH[7:0]								00000000b
PCON	87H	电源管理控制寄存器	-	-	-	-	-	-	STOP	-	xxxxxx0xb
TCON	88H	定时器控制寄存器	TF1	TR1	TF0	TR0	-	-	-	-	0000xxxxb
TMOD	89H	定时器工作模式寄存器	GATE1	C/T1	M11	M01	GATE0	C/T0	M10	M00	00000000b
TL0	8AH	定时器 0 低 8 位	TL0[7:0]								00000000b
TL1	8BH	定时器 1 低 8 位	TL1[7:0]								00000000b
TH0	8CH	定时器 0 高 8 位	TH0[7:0]								00000000b
TH1	8DH	定时器 1 高 8 位	TH1[7:0]								00000000b
TMCON	8EH	定时器频率控制寄存器	-	-	-	-	-	-	T1FD	T0FD	xxxxxx00b

P1	90H	P1 口数据寄存器	P1.7	P1.6	-	-	P1.3	P1.2	-	P1.0	11xx11x1b
P1CFG1	91H	P1 口模式设定寄存器	P17M[1:0]		P16M[1:0]		-		-		1010xxxxb
P1CFG0	92H	P1 口模式设定寄存器	P13M[1:0]		P12M[1:0]		-		P10M[1:0]		1010xx10b
P1IT	93H	INT1 中断类型寄存器	-	-	-	-	-	-	INT1ES[1:0]		xxxxxx00b
TKCNTH	96H	触摸按键计数器高位寄存器	-	TKCNT[14:8]							x0000000b
TKCNTL	97H	触摸按键计数器低位寄存器	TKCNT[7:0]							00000000b	
TKCFG1	9AH	触摸按键配置寄存器 1	-	CMPFLT[2:0]		TKIOS	-	DISCTS[1:0]			x0000x00b
TKCFG2	9BH	触摸按键配置寄存器 2	FRANDS[1:0]		-	-	CTIME[3:0]				00xx0000b
TKCFG3	9CH	触摸按键配置寄存器 3	-	-	SVSS[2:0]		CMPVREFS[2:0]				xx000000b
TKSTA	9DH	触摸按键状态寄存器	TMOU	-	-	-	DCMPO	ACMPO	SACMPO	TKIF	0xxx0000b
TKPVH	9EH	触摸按键比较值高位寄存器	-	TKPV[14:8]							x0000000b
TKPVL	9FH	触摸按键比较值低位寄存器	TKPV[7:0]							00000000b	
P2	A0H	P2 口数据寄存器	P2.7	-	-	P2.4	-	-	-	P2.0	1xx1xxx1b
P2CFG1	A1H	P2 口模式设定寄存器	P27M[1:0]		-		-		P24M[1:0]		00xxxx00b
P2CFG0	A2H	P2 口模式设定寄存器	-		-		-		P20M[1:0]		xxxxxx00b
IE	A8H	中断控制寄存器	EA	EADC	EPWM	ESIF	ET1	-	ET0	ETK	00000000b
P3	B0H	P3 口数据寄存器	-	P3.6	P3.5	-	-	P3.2	P3.1	P3.0	x11xx111b
P3CFG1	B1H	P3 口模式设定寄存器	-		P36M[1:0]		P35M[1:0]		-		xx0000xxb
P3CFG0	B2H	P3 口模式设定寄存器	-		P32M[1:0]		P31M[1:0]		P30M[1:0]		xx000000b
EXIE	B4H	外部中断使能控制寄存器	-	-	EINT5	EINT4	-	-	EINT1	EINT0	xx00xx00b
EXIP	B5H	外部中断优先级设定寄存器	-	-	IPEX5	IPEX4	-	-	IPEX1	IPEX0	xx00xx00b
IP	B8H	中断优先级控制寄存器	-	IPADC	IPPWM	IPSIF	IPT1	-	IPT0	IPTK	x0000x00b
ADCCFG	C4H	P1 口 ADC 配置寄存器	P17AIN7	P16AIN6	-	-	P13AIN3	P12AIN2	-	VREFS	00xx00x0b
ADCCR	C5H	ADC 控制寄存器	ADCEN	ADCS	LOWSP	EOC	-	ADCIS[2:0]			0000x000b
ADCVH	C6H	ADC 结果寄存器, 储存转化后的结果 ADCV[9:2]	ADCV[9:2]							10000000b	
ADCVL	C7H	ADC 结果寄存器, 储存转化后的结果 ADCV[1:0]	-	-	-	-	-	-	ADCV[1:0]		xxxxxx00b
WDTCR	C9H	WDT 控制寄存器	ENWDT	-	-	CLRWDT	-	-	WDTCKS[1:0]		0xx0xx00b
PSW	D0H	程序状态字寄存器	CY	AC	F0	RS1	RS0	OV	-	P	000000x0b
SIFCFG	D4H	SIF 配置寄存器	ENSI	INVI	-	-	SIMOD[2:0]		ACKO		00xx0000b
SIFCTL	D5H	SIF 控制寄存器	-	-	-	-	-	-	MCMMD[1:0]		xxxxxx00b
SIFTXD	D6H	SIF 发送数据寄存器	SIFTXD[7:0]							00000000b	
SIFRXD	D7H	SIF 接收数据寄存器	SIFRXD[7:0]							xxxxxxx0b	
SIFSTA	D8H	SIF 状态寄存器	RTNACK	-	-	-	STPIF	TXIF	RXIF	STRIF	0xxx0000b
ACC	E0H	累加器	ACC[7:0]							00000000b	
TKCR	E8H	触摸按键控制寄存器	ENTKS	TRIG	-	CHRVS	TKAINS[3:0]				00x00000b
IAPKEY	EAH	IAP 保护寄存器	IAPKEY[7:0]							00000000b	
IAPADL	ECH	IAP 地址低 8 位寄存器, 高位恒定为 1	IAPADR[7:0]							11111111b	
IAPDAT	EDH	IAP 数据寄存器	IAPDAT[7:0]							11111111b	
IAPCTL	EEH	IAP 控制寄存器	-	-	-	-	PAYTIMES[1:0]		CMD[1:0]		xxxx0000b
IRCC	EFH	IRC 频率改变寄存器	-	IRCC[6:0]							xnnnnnnb
B	F0H	B 寄存器	-							00000000b	
RSTCFG	F6H	复位设置寄存器 设定门限电压及复位使能	-	-	DISRST	DISLVR	LVR[3:0]				xx0nnnnb
PWMCR	F8H	PWM 控制寄存器	ENPWM	PWMIF	-	-	DTY18	ENPWM10	DTY08	ENPWM00	00xx0000b
PWMPRD	F9H	PWM 周期设置寄存器	PWMPRD[7:0]							11111111b	
PWMDTY1	FAH	PWM1 高电平设置寄存器	PWMDTY1[7:0]							00000000b	
PWMDTY0	FBH	PWM0 高电平设置寄存器	PWMDTY0[7:0]							00000000b	
PWMCFG	FCH	PWM 设置寄存器	-	-	INV1	INV0	-	PWMCKS[2:0]			xx00x000b

8051 CPU 内核常用特殊功能寄存器介绍:

### 1. 程序计数器 PC

程序计数器 PC 不属于 SFR 寄存器。PC 有 16 位，是用来控制指令执行顺序的寄存器。单片机上电或者复位后，PC 值为 0000H,也即是说单片机程序从 0000H 地址开始执行程序。

### 2. 累加器 ACC (E0H)

累加器 ACC 是 8051 内核单片机的最常用的寄存器之一，指令系统中采用 A 作为助记符。常用来存放参加计算或者逻辑运算的操作数及结果。

### 3. B 寄存器(F0H)

B 寄存器在乘除法运算中必须与累加器 A 配合使用。乘法指令 MUL A, B 把累加器 A 和寄存器 B 中的 8 位无符号数相乘，所得的 16 位乘积的低位字节放在 A 中，高位字节放在 B 中。除法指令 DIV A, B 是用 A 除以 B，整数商放在 A 中，余数放在 B 中。寄存器 B 还可以作为通用的暂存寄存器使用。

### 4. 堆栈指针 SP(81H)

堆栈指针是一个 8 位的专用寄存器，它指示出堆栈顶部在通用 RAM 中的位置。单片机复位后，SP 初始值为 07H，即堆栈会从 08H 开始向上增加。08H~1FH 为工作寄存器组 1~3，最好将 SP 值修改为 60H~7FH 的区间为宜。

### 5. PSW(D0h) 程序状态字寄存器

位编号	7	6	5	4	3	2	1	0
符号	CY	AC	F0	RS1	RS0	OV	-	P
上电初始值	0	0	0	0	0	0	x	0

位编号	位符号	说明															
7	<b>CY</b>	标志位 1: 加法运算最高位有进位，或者减法运算最高位有借位时 0: 加法运算最高位无进位，或者减法运算最高位无借位时															
6	<b>AC</b>	进位辅助标志位（可在 BCD 码加减法运算时方便调整） 1: 加法运算时在 bit3 位有进位，或减法运算在 bit3 位有借位时 0: 无借位、进位															
5	<b>F0</b>	用户标志位															
4~3	<b>RS1、RS0</b>	工作寄存器组选择位: <table border="1" style="margin-left: 20px;"> <tr> <th>RS1</th> <th>RS0</th> <th>当前使用的工作寄存器组 0~3</th> </tr> <tr> <td>0</td> <td>0</td> <td>组 0 (00H~07H)</td> </tr> <tr> <td>0</td> <td>1</td> <td>组 1 (08H~0FH)</td> </tr> <tr> <td>1</td> <td>0</td> <td>组 2 (10H~17H)</td> </tr> <tr> <td>1</td> <td>1</td> <td>组 3 (18H~1FH)</td> </tr> </table>	RS1	RS0	当前使用的工作寄存器组 0~3	0	0	组 0 (00H~07H)	0	1	组 1 (08H~0FH)	1	0	组 2 (10H~17H)	1	1	组 3 (18H~1FH)
RS1	RS0	当前使用的工作寄存器组 0~3															
0	0	组 0 (00H~07H)															
0	1	组 1 (08H~0FH)															
1	0	组 2 (10H~17H)															
1	1	组 3 (18H~1FH)															
2	<b>OV</b>	溢出标志位															
0	<b>P</b>	奇偶标志位。此标志位为累加器 ACC 中 1 的个数的奇偶值。 1: ACC 中 1 的个数为奇数 0: ACC 中 1 的个数为偶数（包括 0 个）															
1	保留位	保留位															

### 6. 数据指针 DPTR (82H、83H)

数据指针 DPTR 是一个 16 位的专用寄存器，由低 8 位 DPL (82H) 和高 8 位 DPH (83H) 组成。DPTR 是以传统 8051 内核单片机中唯一可以直接进行 16 位操作的寄存器，也可以分别对 DPL 和 DPH 按 Byte 进行操作。

## 7 电源、复位和时钟

### 7.1 电源电路

SC91F830 内建了一个经调校过的精准 2.4V 电压，可用作 ADC 内部参考电压。用户可在 ADC 章节查找具体设置内容。

### 7.2 上电复位过程

SC91F830 上电后，在客户端软件执行前，会经过以下的过程：

- ◆ 复位阶段
- ◆ 调入信息阶段
- ◆ 正常操作阶段

#### 复位阶段

是指 SC91F830 会一直处于复位的情况，直到供应给 SC91F830 的电压高过某一电压，内部才开始有效的 Clock。复位阶段的时间长短和外部电源的上升速度有关，外部电源一定要高过用户编程 IC 时选定的 LVR 电压后，复位阶段才会完成。

#### 调入信息阶段

在 SC91F830 内部有一个预热计数器。在复位阶段期间，此预热计数器一直被清为 0，直到电压过了最高的 LVR 门坎后，该预热计数器开始计数。当内部的预热计数器计数到一定数目后，每隔一定数量个 IRC clock 就会从 Flash ROM 中的 IFB（包含 Code Option）读出一个 byte 数据存放到内部系统寄存器中。直到预热完成后，该复位信号才会结束。

#### 正常操作阶段

结束调入信息阶段后，SC91F830 开始从 Flash 中读取指令代码即进入正常操作阶段。这时的 LVR 电压值是用户写入 Code Option 的设置值。

### 7.3 复位方式

SC91F830 有 4 种复位方式：①外部 RST 复位②低电压复位 LVR③上电复位 POR④看门狗 WDT 复位。

#### 7.3.1 外部 RST 复位

外部 RST 复位就是从外部 RST 给 SC91F830 一定宽度的复位脉冲信号，来实现 SC91F830 的复位。

RST/P1.0 管脚在上电时作为复位管脚使用，用户可以在复位结束后通过软件来将其修改为 P1.0 使用。修改方法参考下面 LVR 部分关于 RSTCFG（F6H）的使用说明。

#### 7.3.2 低电压复位 LVR

SC91F830 内建了一个低电压复位电路。而复位的门限电压有 4 种选择，缺省值 Default 是用户写入的 Option 值，共有 4 种电压选择 3.65V、3.50V、2.60V、2.45V。

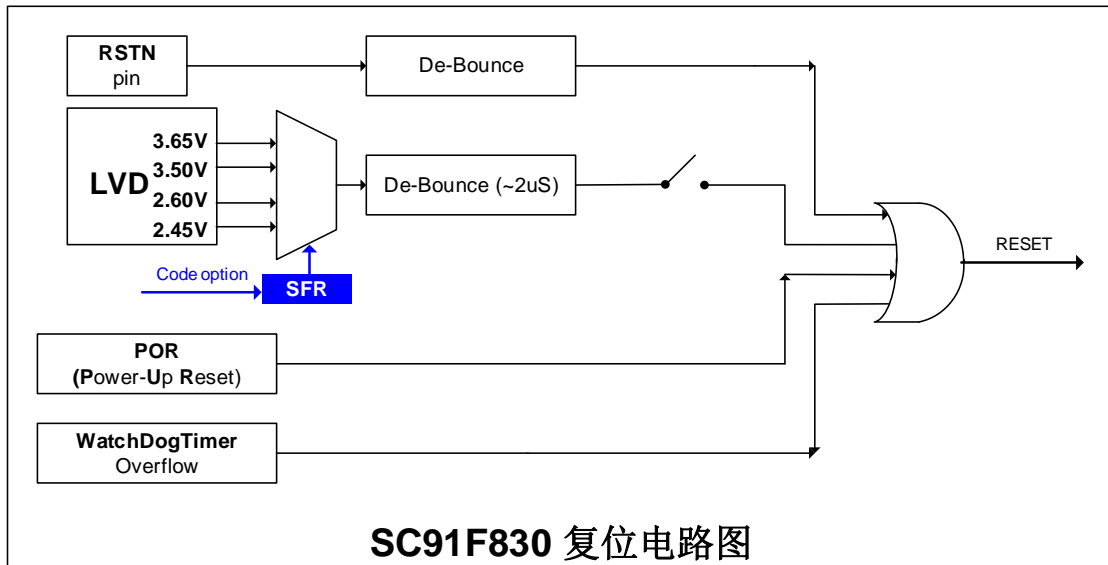
##### RSTCFG (F6h)复位设置寄存器（读/写）

位编号	7	6	5	4	3	2	1	0
符号	-	-	DISRST	DISLVR	LVRS[3:0]			
读/写	-	-	读/写	读/写	读/写			
上电初始值	x	x	0	n	n	n	n	n

位编号	位符号	说明
7,6	保留位	保留位
5	DISRST	IO/RST 复位切换控制 0：P1.0 当复位脚使用 1：P1.0 当正常的 I/O 管脚使用
4	DISLVR	LVR 使能设置 0：LVR 正常使用 1：LVR 无效
3~0	LVRS [3:0]	LVR 电压选择控制 1011：3.65V 复位 1010：3.50V 复位 0101：2.60V 复位 0100：2.45V 复位



SC91F830 的复位部分电路结构图如下：

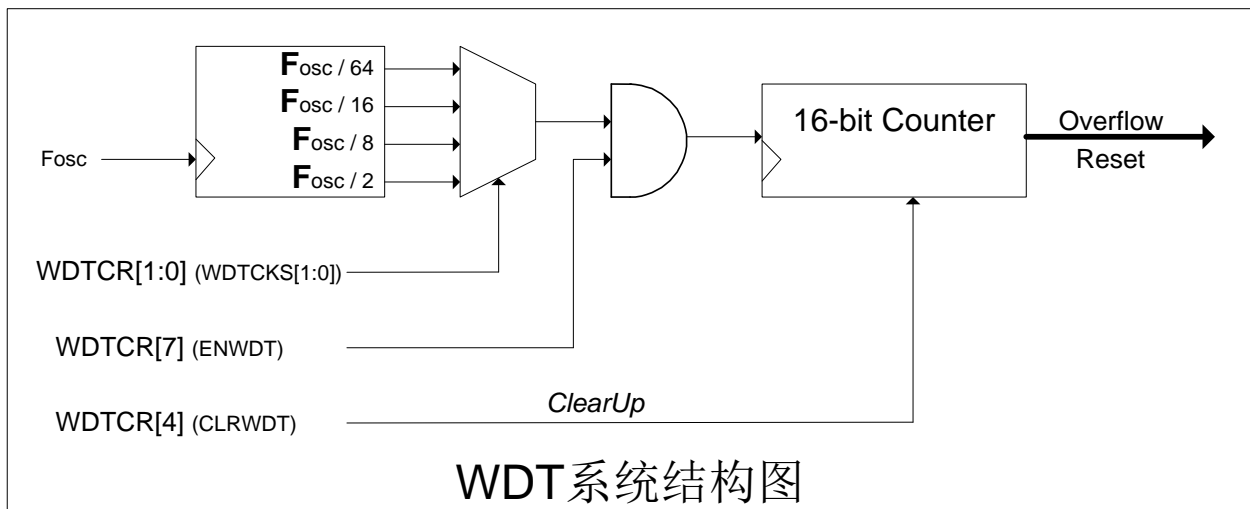


### 7.3.3 上电复位 POR

SC91F830 内部有上电复位电路，当电源电压 VDD 上升到复位电压点时，系统自动复位。

### 7.3.4 看门狗复位 WDT

SC91F830 有一个 16 位的 WDT，其时钟源为内部的 16MHz 振荡器。其系统结构如下图所示：



**WDTCR (C9h) 看门狗控制寄存器（读/写）**

位编号	7	6	5	4	3	2	1	0
符号	ENWDT	-	-	CLRWDT	-	-	WDTCKS[1:0]	
读/写	读/写	-	-	读/写	-	-	读/写	
上电初始值	0	x	x	0	x	x	0	0

位编号	位符号	说明
7	ENWDT	WDT 开关（此位由系统将用户 Code Option 所设的值调入） 1: WDT 开始工作 0: WDT 关闭
6,5,3,2	保留位	保留位
4	CLRWDT	WDT 清“0”位（写 1 有效） 1: WDT 计数器从 0 开始计数

1,0	WDTCKS [1:0]	此位由系统硬件自动置 0			
		看门狗时钟选择（表中按照 16Mhz 系统频率给出 WDT 的参考时间，如使用其他时钟的用户，请同比例计算）			
		WDTCKS.1	WDTCKS.0	WDT 时钟频率	WDT 溢出时间 (@16Mhz)
		0	0	Fosc/64	262.144ms
		0	1	Fosc/16	65.536ms
1	0	Fosc/8	32.768ms		
1	1	Fosc/2	8.192ms		

### 7.3.5 复位初始状态

当 SC91F830 处于复位状态时，多数寄存器会回到其初始状态。看门狗 WDT 处于关闭的状态，PORT 口寄存器为 FFh。程序计数器 PC 初始值为 0000h，堆栈指针 SP 初始值为 07h。“热启动”的 Reset（如 WDT、LVR、软件复位等）不会影响到 SRAM，SRAM 值始终是复位前的值。SRAM 内容的丢失会发生在电源电压低到 RAM 无法保存为止。

SFR 寄存器的上电复位初始值如下表：

SFR 名称	初始值	SFR 名称	初始值
ACC	0000000b	ADCCR	0000x000b
B	0000000b	ADCVH	1000000b
PSW	00000x0b	ADCVL	xxxxxx00b
SP	00000111b	WDTCR	nxx0xx00b
DPL	0000000b	IAPKEY	0000000b
DPH	0000000b	IAPADL	11111111b
PCON	xxxxxx0xb	IAPDAT	11111111b
IE	0000000b	IAPCTL	xxxx0000b
IP	x0000000b	IRCC	xnnnnnnnb
P1	11xx11x1b	RSTCFG	xx0nnnnnb
P2	1xx1xxx1b	PWMCR	00xx0000b
P3	x11xx111b	PWMPRD	11111111b
P1CFG0	1010xx10b	PWMDTY1	0000000b
P1CFG1	1010xxxxb	PWMDTY0	0000000b
P2CFG0	xxxxxx00b	PWMCFG	xx00x000b
P2CFG1	00xxxx00b	SIFCFG	00xx0000b
P3CFG0	xx000000b	SIFCTL	xxxxxx00b
P3CFG1	xx0000xxb	SIFTXD	0000000b
TCON	0000xxxxb	SIFRXD	xxxxxxxxxb
TMOD	0000000b	SIFSTA	0xxx0000b
TH0	0000000b	TKCNTH	x000000b
TL0	0000000b	TKCNTL	0000000b
TH1	0000000b	TKPVH	x000000b
TL1	0000000b	TKPVL	0000000b
TMCON	xxxxxx00b	TKCFG1	x0000x00b
EXIE	xx00xx00b	TKCFG2	00xx0000b
EXIP	xx00xx00b	TKCFG3	xx000000b
PIIT	xxxxxx00b	TKSTA	0xxx0000b
ADCCFG	00xx00x0b	TKCR	00x00000b

## 7.4 时钟电路

SC91F830 内建了一个振荡频率可调的高精度 IRC 作为系统时钟，出厂时被精确地调校至 16MHz@5V/25℃，用户可以通过编程器的 Code Option 将系统时钟设置为 16MHz 和 4MHz 使用。调校过程是过滤掉制程上的偏差对精度所造成的影响。此 IRC 受工作的环境温度和工作电压影响会有一些的漂移，对于压漂（4.5V~5.5V）以及(-20℃~85℃)的温漂一般状况会在 ±1% 以内。

SC91F830 有一个特殊的功能：用户可修改 SFR 的值实现 IRC 频率在一定范围的调整（大约±10%）。

### IRCC (EFH) 系统时钟改变寄存器（读/写）

位编号	7	6	5	4	3	2	1	0
符号	-	IRCC[6:0]						
读/写	-	读/写						
上电初始值	-	n	n	n	n	n	n	n

位编号	位符号	说明																				
6~0	IRCC[6:0]	<p><b>IRC 频率改变寄存器</b></p> <p>IRCC[6:0]在上电后的值 IRCC[s]确保 IRC 可准确工作在 16/4MHz(根据用户 Code Option 的选择)，此数值的初始值每颗 IC 都可能会有差异。用户可通过修改此寄存器的值实现 IRC 工作频率的改变。</p> <p>初始值为 IRCC[s]，此时 IC 工作在 16/4MHz，IRCC[6:0]每改变 1 则 IRC 频率改变约 62.5kHz。</p> <p>IRCC[6:0]和 IRC 输出频率的关系如下：</p> <table border="1"> <thead> <tr> <th>IRCC[6:0]值</th> <th>IRC 实际输出频率（16M 为例）</th> </tr> </thead> <tbody> <tr> <td>IRCC[s]-n</td> <td>(16000-n*62.5)kHz</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>IRCC[s]-2</td> <td>16000-125=15875kHz</td> </tr> <tr> <td>IRCC[s]-1</td> <td>16000-62.5=15937.5kHz</td> </tr> <tr> <td>IRCC[s]</td> <td>16000kHz</td> </tr> <tr> <td>IRCC[s]+1</td> <td>16000+62.5=16062.5kHz</td> </tr> <tr> <td>IRCC[s]+2</td> <td>16000+125=16125kHz</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>IRCC[s]+n</td> <td>(16000+n*62.5)kHz</td> </tr> </tbody> </table> <p>说明：</p> <ol style="list-style-type: none"> <li>1. IC 每次上电后 IRCC[6:0]的值都是 IRC 工作在最接近 16/4MHz 的 IRCC[s]；用户可借助 EEPROM 在每次上电后修正 IRCC 的值以让 IRC 工作在用户需要的频率；</li> <li>2. 为保证 IC 工作可靠，IRC 最高工作频率尽量勿超过 16MHz 的 10% 即 17.6M Hz；</li> <li>3. IRCC[6:0]的值如果超过范围，则 IRC 只会选择这 7 个 Bit 的值运行，比如 IRCC[6:0]如果赋值 80h 则效果等同于 00h；</li> <li>4. 请用户确认 IRC 频率的改变不会影响其它功能；</li> </ol>	IRCC[6:0]值	IRC 实际输出频率（16M 为例）	IRCC[s]-n	(16000-n*62.5)kHz	...	...	IRCC[s]-2	16000-125=15875kHz	IRCC[s]-1	16000-62.5=15937.5kHz	IRCC[s]	16000kHz	IRCC[s]+1	16000+62.5=16062.5kHz	IRCC[s]+2	16000+125=16125kHz	...	...	IRCC[s]+n	(16000+n*62.5)kHz
IRCC[6:0]值	IRC 实际输出频率（16M 为例）																					
IRCC[s]-n	(16000-n*62.5)kHz																					
...	...																					
IRCC[s]-2	16000-125=15875kHz																					
IRCC[s]-1	16000-62.5=15937.5kHz																					
IRCC[s]	16000kHz																					
IRCC[s]+1	16000+62.5=16062.5kHz																					
IRCC[s]+2	16000+125=16125kHz																					
...	...																					
IRCC[s]+n	(16000+n*62.5)kHz																					
7	保留位	保留位																				

## 7.5 省电模式 STOP

SC91F830 提供了一个特殊功能寄存器 PCON。只要对该寄存器 PCON.1 写入 1，内部的晶振就会停止，进到 STOP 模式，达到省电功能。在 STOP 模式下，用户可以通过外部中断 INT0~INT7 把 SC91F830 唤醒，也可以通过外部复位将 STOP 唤醒。

**PCON (87h) 电源控制寄存器 (只写、\*不可读\*)**

位编号	7	6	5	4	3	2	1	0
符号	-	-	-	-	-	-	STOP	-
读/写	-	-	-	-	-	-	只写	-
上电初始值	x	x	x	x	x	x	0	x
位编号	位符号		说明					
1	STOP		STOP 模式控制 0: 正常操作模式 1: 节能模式, 内部晶振停止工作					

## 8 中央处理单元 CPU 及指令系统

### 8.1 CPU

SC91F830 所用的 CPU 是一个超快速的 1T 标准 8051 内核, 其指令完全兼容传统 8051 内核单片机。

### 8.2 寻址方式

SC91F830 的 1T 8051 CPU 指令的寻址方式有: ①立即寻址②直接寻址③间接寻址④寄存器寻址⑤相对寻址⑥变址寻址⑦位寻址

#### 8.2.1 立即寻址

立即寻址也称为立即数, 它是在指令操作数中直接给出参加运算的操作数, 指令举例如下:

MOV A, #50H (这条指令是将立即数 50H 送到累加器 A 中)

#### 8.2.2 直接寻址

在直接寻址方式中, 指令操作数域给出的是参加运算操作数的地址。直接寻址方式只能用来表示特殊功能寄存器、内部数据寄存器和位地址空间。其中特殊功能寄存器和位地址空间只能用直接寻址方式访问。举例如下:

ANL 50H, #91H (表示 50H 单元中的数与立即数 91H 相“与”, 结果存放在 50H 单元中。其中 50H 为直接地址, 表示内部数据寄存器 RAM 中的一个单元。)

#### 8.2.3 间接寻址

间接寻址采用 R0 或 R1 前添加“@”符号来表示。假设 R1 中的数据是 40H, 内部数据存储 40H 单元的数据为 55H, 则指令为

MOV A, @R1 (把数据 55H 传送至累加器 A)。

#### 8.2.4 寄存器寻址

寄存器寻址时对选定的工作寄存器 R7~R0、累加器 A、通用寄存器 B、地址寄存器和进位 C 中的数进行操作。其中寄存器 R7~R0 由指令码的低 3 位表示, ACC、B、DPTR 及进位位 C 隐含在指令码中。因此, 寄存器寻址也包含一种隐含寻址方式。寄存器工作区的选择由程序状态字寄存器 PSW 中的 RS1、RS0 来决定。指令操作数指定的寄存器均指当前工作区的寄存器。

INC R0 是指 (R0) +1→R0

#### 8.2.5 相对寻址

相对寻址是将程序计数器 PC 中的当前值与指令第二字节给出的数相加, 其结果作为转移指令的转移地址。转移地址也成为转移目的地址, PC 中的当前值成为基地址, 指令第二字节给出的数成为偏移量。由于目的地址是相对于 PC 中的基地址而言, 所以这种寻址方式成为相对寻址。偏移量为带符号的数, 所能表示的范围为+127~-128。这种寻址方式主要用于转移指令。

JC \$+50H

表示若进位位 C 为 0, 则程序计数器 PC 中的内容不改变, 即不转移。若进位位 C 为 1, 则以 PC 中的当前值及基地址, 加上偏移量 50H 后所得到的结果作为该转移指令的目的地址。

#### 8.2.6 变址寻址

在变址寻址方式中, 指令操作数制定一个存放变址基址的变址寄存器。变址寻址时, 偏移量与变址基值相加, 其结果作为操作数的地址。变址寄存器有程序计数器 PC 和地址寄存器 DPTR。

MOVC A, @A+DPTR

表示累加器 A 为偏移量寄存器, 其内容与地址寄存器 DPTR 中的内容相加, 其结果作为操作数的地址, 取出该单元中的数送入累加器 A 中。

### 8.2.7 位寻址

位寻址是指对一些可进行位操作的内部数据存储器 RAM 和特殊功能寄存器进行位操作时的寻址方式。在进行位操作时，借助于进位位 C 作为位操作累加器，指令操作数直接给出该位的地址，然后根据操作码的性质对该位进行位操作。位地址与字节直接寻址中的字节地址编码方式完全一样，主要由操作指令的性质加以区分，使用时应特别注意。

MOV C, 20H (将地址为 20H 的位操作寄存器值送入进位位 C 中。)

### 8.3 指令系统

1T 8051 指令系统

助记符	功能说明	字节	周期
<b>算术操作指令</b>			
ADD A, Rn	寄存器内容加到累加器 A	1	1
ADD A, direct	直接地址单元中的数据加到累加器 A	2	2
ADD A, @Ri	间接 RAM 中的数据加到累加器 A	1	2
ADD A, #data	立即数加到累加器 A	2	2
ADDC A, Rn	寄存器带进位加到累加器	1	1
ADDC A, direct	直接地址单元的内容带进位加到累加器	2	2
ADDC A, @Ri	间接 RAM 内容带进位加到累加器	1	2
ADDC A, #data	立即数带进位加到累加器	2	2
SUBB A, Rn	累加器带借位减寄存器内容	1	1
SUBB A, direct	累加器带借位减直接地址单元的内容	2	2
SUBB A, @Ri	累加器带借位减间接 RAM 中的内容	1	2
SUBB A, #data	累加器带借位减立即数	2	2
INC A	累加器加 1	1	1
INC Rn	寄存器加 1	1	2
INC direct	直接地址单元加 1	2	3
INC @Ri	间接 RAM 单元加 1	1	3
DEC A	累加器减 1	1	1
DEC Rn	寄存器减 1	1	2
DEC direct	直接地址单元减 1	1	3
DEC @Ri	间接 RAM 单元减 1	2	3
INC DPTR	地址寄存器 DPTR 加 1	1	1
MUL AB	A 乘以 B	1	2
DIV AB	A 除以 B	1	6
DA A	累加器十进制调整	1	3
<b>逻辑操作指令</b>			
ANL A, Rn	累加器与寄存器相“与”	1	1
ANL A, direct	累加器与直接地址单元相“与”	2	2
ANL A, @Ri	累加器与间接 RAM 单元相“与”	1	2
ANL A, #data	累加器与立即数相“与”	2	2
ANL direct, A	直接地址单元与累加器相“与”	2	3
ANL direct, #data	直接地址单元与立即数相“与”	3	3
ORL A, Rn	累加器与寄存器相“或”	1	1
ORL A, direct	累加器与直接地址单元相“或”	2	2
ORL A, @Ri	累加器与间接 RAM 单元相“或”	1	2
ORL A, #data	累加器与立即数相“或”	2	2
ORL direct, A	直接地址单元与累加器相“或”	2	3
ORL direct, #data	直接地址单元与立即数相“或”	3	3
XRL A, Rn	累加器与寄存器相“异或”	1	1
XRL A, direct	累加器与直接地址单元相“异或”	2	2
XRL A, @Ri	累加器与间接地址单元相“异或”	1	2
XRL A, #data	累加器与立即数相“异或”	2	2
XRL direct, A	直接地址单元与累加器相“异或”	2	3
XRL direct, #data	直接地址单元与立即数相“异或”	3	3

CLR A	累加器清“0”	1	1
CPL A	累加器求反	1	1
RL A	累加器循环左移	1	1
RLC A	累加器带进位位循环左移	1	1
RR A	累加器循环右移	1	1
RRC A	累加器带进位位循环右移	1	1
SWAP A	累加器内高低半字节交换	1	1
<b>布尔变量操作指令</b>			
CLR C	清 0 进位位	1	1
CLR bit	清 0 直接地址位	2	3
SETB C	进位位置 1	1	1
SETB bit	直接地址位置 1	2	3
CPL C	进位位求反	1	1
CPL bit	直接地址位求反	2	3
ANL C, bit	进位位和直接地址为相“与”	2	2
ANL C, bit	进位位和直接地址位的反码相“与”	2	2
ORL C, bit	进位位和直接地址位相“或”	2	2
ORL C, bit	进位位和直接地址位反码相“或”	2	2
MOV C, bit	直接地址位送入进位位	2	2
MOV bit, C	进位位送入直接地址位	2	3
JC rel	进位位为 1 则转移	2	3
JNC rel	进位位为 0 则转移	2	3
JB bit, rel	直接地址位为 1 则转移	3	5
JNB bit, rel	直接地址位为 0 则转移	3	5
JBC bit, rel	直接地址位为 1 则转移, 该位清 0	3	5
<b>数据传送类指令</b>			
MOV A, Rn	寄存器内容送入累加器	1	1
MOV A, direct	直接地址单元中的数据送入累加器	2	2
MOV A, @Ri	间接 RAM 中的数据送入累加器	1	2
MOV A, #data	立即数送入累加器	2	2
MOV Rn, A	累加器内容送入寄存器	1	1
MOV Rn, direct	直接地址单元中的数据送入寄存器	2	3
MOV Rn, #data	立即数送入寄存器	2	2
MOV direct, A	累加器内容送入直接地址单元	2	2
MOV direct, Rn	寄存器内容送入直接地址单元	2	2
MOV direct1, direct2	直接地址单元中的数据送入另一个直接地址单元	3	3
MOV direct, @Ri	间接 RAM 中的数据送入直接地址单元	2	3
MOV direct, #data	立即数送入直接地址单元	3	3
MOV @Ri, A	累加器内容送间接 RAM 单元	1	2
MOV @Ri, direct	直接地址单元数据送入间接 RAM 单元	2	3
MOV @Ri, #data	立即数送入间接 RAM 单元	2	2
MOV DPTR, #data16	16 位立即数送入 DPTR	3	3
MOVC A, @A+DPTR	以 DPTR 为基地址变址寻址单元中的数据送入累加器	1	5
MOVC A, @A+PC	以 PC 为基地址变址寻址单元中的数据送入累加器	1	4
MOVX A, @Ri	逻辑上在外部的片内扩展 RAM (8 位地址), 送入累加器	1	3
MOVX @Ri, A	累加器送入逻辑上在外部的片内扩展 RAM (8 位地址)	1	4
MOVX A, @DPTR	逻辑上在外部的片内扩展 RAM (16 位地址), 送入累加器	1	2
MOVX @DPTR, A	累加器送入逻辑上在外部的片内扩展 RAM (16 位地址)	1	3
PUSH direct	直接地址单元中的数据压入堆栈	2	3
POP direct	栈底数据弹出送入直接地址单元	2	2
XCH A, Rn	寄存器与累加器交换	1	2
XCH A, direct	直接地址单元与累加器交换	2	3
XCH A, @Ri	间接 RAM 与累加器交换	1	3
XCHD A, @Ri	间接 RAM 的低半字节与累加器交换	1	3

控制转移类指令			
ACALL address11	绝对（短）调用子程序	2	4
LCALL address16	长调用子程序	3	4
RET	子程序返回	1	4
RETI	中断返回	1	4
AJMP address11	绝对（短）转移	2	3
LJMP address16	长转移	3	4
SJMP rel	相对转移	2	3
JMP @A+DPTR	相对于 DPTR 的间接转移	1	5
JZ rel	累加器为 0 转移	2	4
JNZ rel	累加器非 0 转移	2	4
CJNE A, direct, rel	累加器与直接地址单元比较，不相等则转移	3	5
CJNE A, #data, rel	累加器与立即数比较，不相等则转移	3	4
CJNE Rn, #data, rel	寄存器与立即数比较，不相等则转移	3	4
CJNE @Ri, #data, rel	间接 RAM 单元与立即数比较，不相等则转移	3	5
DJNZ Rn, rel	寄存器减 1，非 0 转移	2	4
DJNZ direct, rel	直接地址单元减 1，非 0 转移	3	5
NOP	空操作	1	1

SC91F830 的 MOVC 指令禁止寻址 0000~00FFH 地址，具体使用说明请参考《赛元 MCU 应用注意事项》

## 9 INTERRUPT 中断

SC91F830 单片机提供了 11 个中断源：Timer0、Timer1、PWM、ADC、SIF、TK、INT0、INT1、INT4、INT5。这 10 个中断源分为 2 个中断优先级，并可以单独分别设置为高优先级或者低优先级。每个中断分别有独立的优先级设置位、中断标志、中断向量和使能位，总的使能位 EA 可以实现所有中断的打开或者关闭。

### 9.1 中断源、向量

SC91F830 的中断源、中断向量、及相关控制位列表如下：

中断源	中断发生时间	中断标志	中断使能控制	中断优先级控制	中断向量	查询优先级	中断号 (C51)	标志清除方式	能否唤醒 STOP
TK	Touch Key 完成	TKSTA[0] TKIF	IE[0] (ETK)	IP[0]	0003H	1 (高)	0	H/W Auto	不能
Timer0	Timer0 溢出	TCON[5] (TF0)	IE[1] (ET0)	IP[1]	000BH	2	1	H/W Auto	不能
Timer1	Timer1 溢出	TCON[7] (TF1)	IE[3] (ET1)	IP[3]	001BH	3	3	H/W Auto	不能
SIF	SIF 中断	STPIF/RXIF/ TXIF/STRIF	IE[4]	IP[4]	0023H	4	4	必须用户清除	不能
PWM	PWM 溢出	PWMCR[7] (PWMIF)	IE[5] (EPWM)	IP[5]	002BH	5	5	必须用户清除	不能
ADC	ADC 转换完成	ADCCR[4] (EOC/ADCIF)	IE[6] (EADC)	IP[6]	0033H	6	6	必须用户清除	不能
INT0	下降沿	隐藏式	EXIE[0]	EXIP[0]	003BH	7	7	H/W Auto	能
INT1	下降沿 上升沿 双沿	隐藏式	EXIE[1]	EXIP[1]	0043H	8	8	H/W Auto	能
INT4	下降沿	隐藏式	EXIE[4]	EXIP[4]	005BH	9	11	H/W Auto	能
INT5	下降沿	隐藏式	EXIE[5]	EXIP[5]	0063H	10	12	H/W Auto	能

在 EA=1 及各中断使能控制为 1 的情况下，各中断发生情况如下：

定时器中断：Timer0 和 Timer1 溢出时会产生中断并将中断标志 TF0 和 TF1 置为“1”，当单片机执行该定时器中断时，中断标志 TF0 和 TF1 会被硬件自动清“0”。

PWM 中断：当 PWM 计数器溢出时(也就是说：计数器数到超过 PWMPRD 时)，PWMIF 位(PWM Interrupt Flag)会被硬件自动置“1”，PWM 中断产生。在 PWM 中断发生后，硬件并不会自动清除 PWMIF 位，此 bit 必须由使用者的软件负责清除。

ADC 中断：ADC 中断的发生时间为 ADC 转换完成时，其中断标志就是 ADC 转换结束标志 EOC/ADCIF

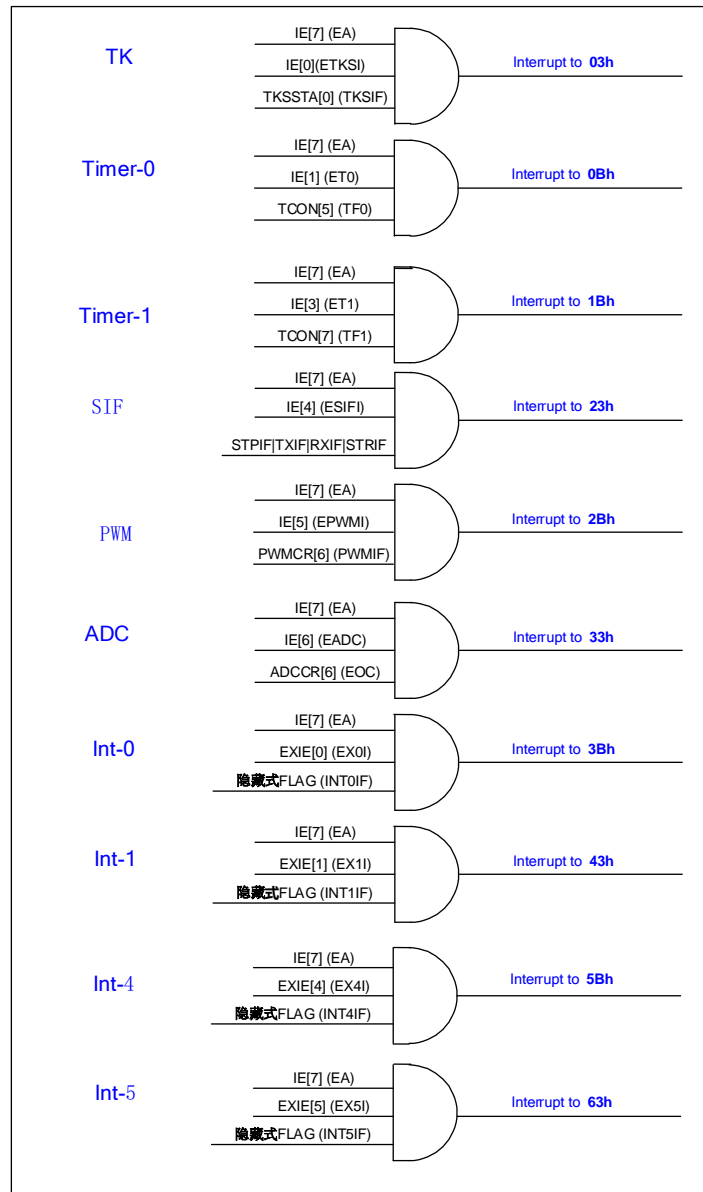
(ADCCR.4)。当使用者设定 ADCS 开始转换后, EOC 会被硬件自动清除为“0”;当转换完成后, EOC 会被硬件自动置为“1”。使用者在 ADC 中断发生之后, 进入中断服务程序时, 必须用软件去清除它。

外部中断 INT<sub>x</sub>(x=0/1/4/5): 外部中断 INT0/1/4/5 有单独的中断向量, 当外部中断口有中断条件发生时, 外部中断就发生了。这 8 个外部中断标志是系统隐藏式的, 不需要用户做处理, 硬件会自动清除。其中 INT0/4/5 的外部中断仅下降沿触发, 无需用户设置; INT1 为初始值为单下降沿的外部中断, 如果用户需要双沿或者上升沿中断, 可通过设置 SFR (P1IT) 来实现。用户可通过 EXIP 寄存器来设置每个中断的优先级级别。外部中断 INT0/1/4/5 还可以唤醒单片机的 STOP。



## 9.2 中断结构图

SC91F830 的中断结构如下图所示：



SC91F830中断结构和向量

## 9.3 中断优先级

SC91F830 单片机的中断具有两个中断优先级，这些中断源请求可编程为高优先级中断或者低优先级中断，即可实现两级中断服务程序的嵌套。一个正在执行的低优先级中断能被高优先级中断请求所中断，但不能被另一个同一优先级的中断请求所中断，一直执行到结束，遇到返回指令 **RETI**，返回主程序后再执行一条指令才能响应新的中断请求。

也就是说

- ① 低优先级中断可被高优先级中断请求所中断，反之不能；
- ② 任何一种中断，在响应过程中，不能被同一优先级的中断请求所中断。

中断查询顺序：SC91F830 单片机的同一优先级中断，如果同时来几个中断，则中断响应的优先顺序同 C51

中的中断查询号相同，即查询号小的会优先响应，查询号大的会慢响应。

## 9.4 中断处理流程

当一个中断产生并且被 CPU 响应，则主程序运行被中断，将执行下述操作

- 1, 当前正在执行的指令执行完;
- 2, PC 值被压入堆栈, 保护现场;
- 3, 中断向量地址载入程序计数器 PC;
- 4, 执行相应的中断服务程序;
- 5, 中断服务程序结束并 RETI;
- 6, 将 PC 值退栈, 并返回执行中断前的程序。

在此过程中，系统不会立即执行其它同一优先级的中断，但会保留所发生的中断请求，在当前中断处理结束后，转去执行新的中断请求。

## 9.5 中断相关 SFR 寄存器

### IE (A8h) 中断使能寄存器 (读/写)

位编号	7	6	5	4	3	2	1	0
符号	EA	EADC	EPWM	ESIF	ET1	-	ET0	ETK
读/写	读/写	读/写	读/写	读/写	读/写	-	读/写	读/写
上电初始值	0	0	0	0	0	x	0	0

位编号	位符号	说明
7	EA	中断使能的总控制 0: 关闭所有的中断 1: 打开所有的中断
6	EADC	ADC 中断使能控制 0: 关闭 ADC 中断 1: 允许 ADC 转换完成时产生中断
5	EPWM	PWM 中断使能控制 0: 关闭 PWM 中断 1: 允许 PWM 计数溢出 (数到 PWMPRD) 时产生中断
4	ESIF	SIF 中断使能控制 0: 关闭 SIF 中断 1: 允许 SIF 的所有中断
3	ET1	Timer1 中断使能控制 0: 关闭 TIMER1 中断 1: 允许 TIMER1 中断
1	ET0	Timer0 中断使能控制 0: 关闭 TIMER0 中断 1: 允许 TIMER0 中断
0	ETK	Touch Key 中断使能控制 0: 关闭 Touch Key 中断 1: 打开 Touch Key 中断

### IP (B8h) 中断优先级寄存器 (读/写)

位编号	7	6	5	4	3	2	1	0
符号	-	IPADC	IPPWM	IPSIF	IPT1	-	IPT0	IPTK
读/写	-	读/写	读/写	读/写	读/写	-	读/写	读/写
上电初始值	x	0	0	0	0	x	0	0
位编号	位符号	说明						
6	IPADC	ADC 中断优先级选择 0: ADC 中断优先级为低 1: ADC 中断优先级为高						

5	<b>IPPWM</b>	PWM 中断优先权选择 0: PWM 中断优先权为低 1: PWM 中断优先权为高
4	<b>IPSIF</b>	SIF 中断优先权选择 0: SIF 中断优先权为低 1: SIF 中断优先权为高
3	<b>IPT1</b>	Timer1 中断优先权选择 0: Timer1 中断优先权为低 1: Timer1 中断优先权为高
1	<b>IPT0</b>	Timer0 中断优先权选择 0: Timer0 中断优先权为低 1: Timer0 中断优先权为高
0	<b>IPTK</b>	Touch Key 中断优先权选择 0: Touch Key 中断优先权为低 1: Touch Key 中断优先权为高
7	<b>保留位</b>	保留位

**EXIE (B4h) 外部中断使能寄存器(读/写)**

位编号	7	6	5	4	3	2	1	0
符号	-	-	EINT5	EINT4	-	-	EINT1	EINT0
读/写	-	-	读/写	读/写	-	-	读/写	读/写
上电初始值	x	x	0	0	x	x	0	0

位编号	位符号	说明
5~4 1~0	<b>EINTx</b> (x=0/1/4/5)	外部中断使能控制 0: 关闭外部中断 INTx(x=0/1/4/5) 的中断 1: 允许外部中断 INTx(x=0/1/4/5) 发生中断
7,6,3,2	<b>保留位</b>	保留位

**EXIP (B5h) 外部中断优先权寄存器(读/写)**

位编号	7	6	5	4	3	2	1	0
符号	-	-	IPEX5	IPEX4	-	-	IPEX1	IPEX0
读/写	-	-	读/写	读/写	-	-	读/写	读/写
上电初始值	x	x	0	0	x	x	0	0

位编号	位符号	说明
5~4,1~0	<b>IPEXn</b> (n=0/1/4/5)	外部中断优先权选择 0: 外部中断 INTn(n=0/1/4/5) 的中断优先级是“低” 1: 外部中断 INTn(n=0/1/4/5) 的中断优先级是“高”

**P1IT (93h) INT1 外部中断类型寄存器(读/写)**

位编号	7	6	5	4	3	2	1	0
符号	-	-					INT1ES[1:0]	
读/写	-	-					读/写	读/写
上电初始值	x	x	x	x	x	x	0	0

位编号	位符号	说明
1~0	<b>INT1ES[1:0]</b>	INT1 Edge Selction ,外部中断类型选择 00: 下降沿中断 01: 无中断

		10: 双沿中断 11: 上升沿中断
7~2	保留位	保留位

## 10 定时器 TIMER0、TIMER1

SC91F830 单片机内部有两个 16 位定时器/计数器分别称为 T0 和 T1，它们具有计数方式和定时方式两种工作模式。特殊功能寄存器 TMOD 中有一个控制位 C/Tx 来选择 T0 和 T1 是定时器还是计数器。它们本质上都是一个加法计数器，只是计数的来源不同。定时器的来源为系统时钟或者其分频时钟，但计数器的来源为外部管脚的输入脉冲。GATEx 和 TRx 是 T0 和 T1 在定时器/计数器模式计数的开关控制，只有在 GATEx=0 且 TRx=1 的时候，T0 和 T1 才会被打开计数。

计数器模式下，P2.0/T0 管脚上的每一个脉冲，T0 的计数值分别增加 1。

定时器模式下，可通过特殊功能寄存器 TMCON 来选择 T0 和 T1 的计数来源是 Fosc/12 或 Fosc/4。

定时器/计数器 T0 有 4 种工作模式，定时器/计数器 T1 有 3 种工作模式（模式三不存在）：

- ①模式 0：13 位定时器/计数器模式
- ②模式 1：16 位定时器/计数器模式
- ③模式 2：8 位自动重载模式
- ④模式 3：两个 8 位定时器/计数器模式。

在上述模式中，T0 和 T1 的模式 0、1、2 都相同，模式 3 不同。

### 10.1 T0 和 T1 相关特殊功能寄存器

符号	地址	说明	7	6	5	4	3	2	1	0	Reset 值
<b>TCON</b>	<b>88H</b>	定时器控制寄存器	TF1	TR1	TF0	TR0	-	-	-	-	0000xxxxb
<b>TMOD</b>	<b>89H</b>	定时器工作模式寄存器	GATE1	C/T1	M11	M01	GATE0	C/T0	M10	M00	00000000b
<b>TL0</b>	<b>8AH</b>	定时器 0 低 8 位									00000000b
<b>TL1</b>	<b>8BH</b>	定时器 1 低 8 位									00000000b
<b>TH0</b>	<b>8CH</b>	定时器 0 高 8 位									00000000b
<b>TH1</b>	<b>8DH</b>	定时器 1 高 8 位									00000000b
<b>TMCON</b>	<b>8EH</b>	定时器频率控制寄存器	-	-	-	-	-	-	T1FD	T0FD	xxxxxx00b

各寄存器的解释说明如下：

#### TCON (88h) 定时器控制寄存器

位编号	7	6	5	4	3	2	1	0
符号	TF1	TR1	TF0	TR0	-	-	-	-
读/写	读/写	读/写	读/写	读/写	-	-	-	-
上电初始值	0	0	0	0	x	x	x	x

位编号	位符号	说明
7	<b>TF1</b>	T1 溢出中断请求标志。T1 产生溢出，发生中断时，硬件将 TF1 置为“1”，申请中断，CPU 响应时，硬件清“0”。
6	<b>TR1</b>	定时器 T1 的运行控制位。此位由软件置 1 和清 0。当 GATE1 TMOD[7]=0,TR1=1 时，允许 T1 开始计数。TR1=0 时禁止 T1 计数。
5	<b>TF0</b>	T0 溢出中断请求标志。T0 产生溢出，发生中断时，硬件将 TF0 置为“1”，申请中断，CPU 响应时，硬件清“0”。
4	<b>TR0</b>	定时器 T0 的运行控制位。此位由软件置位和清 0。当 GATE0 TMOD[3]=0,TR0=1 时，允许 T0 开始计数。TR0=0 时禁止 T0 计数。
3~0	保留位	保留位

#### TMOD (89h) 定时器工作模式寄存器（读/写）

位编号	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---

符号	GATE1	C/T1	M11	M01	GATE0	C/T0	M10	M00
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
上电初始值	0	0	0	0	0	0	0	0
	T1				T0			

位编号	位符号	说明
7	<b>GATE1</b>	TMOD[7]控制定时器 1，置 0 且 TR1 置 1 时才打开 T1
6	<b>C/T1</b>	TMOD[6]控制定时器 1 0: 定时器，T1 计数来源于 Fosc 分频 1: 保留
5,4	<b>M11,M01</b>	定时器/计数器 1 模式选择 0 0 : 13 位定时器/计数器，TL1 高 3 位无效 0 1 : 16 位定时器/计数器，TL1 和 TH1 全 1 0 : 8 位自动重载定时器，溢出时将 TH1 存放的值自动重装入 TL1 1 1 : 定时器/计数器 1 无效（停止计数）
3	<b>GATE0</b>	TMOD[3]控制定时器 0，置 0 且 TR0 置 1 时才打开 T0
2	<b>C/T0</b>	TMOD[2]控制定时器 0 0: 定时器，T0 计数来源于 Fosc 分频 1: 计数器，T0 计数来源于外部管脚 T0/P2.0
1,0	<b>M10,M00</b>	定时器/计数器 0 模式选择 0 0 : 13 位定时器/计数器，TL0 高 3 位无效 0 1 : 16 位定时器/计数器，TL0 和 TH0 全 1 0 : 8 位自动重载定时器，溢出时将 TH0 存放的值自动重装入 TL0 1 1 : 定时器 0 此时作为双 8 位定时器/计数器。TL0 作为一个 8 位定时器/计数器，通过标准定时器 0 的控制位控制；TH0 仅作为一个 8 位定时器，由定时器 1 的控制位控制。

TMOD 寄存器中 TMOD[0]~TMOD[3]是设置 T0 的工作模式；TMOD[4]~TMOD[7]是设置 T1 的工作模式。

定时器和计数器 Tx 功能由特殊功能寄存器 TMOD 的控制位 C/Tx 来选择，M0x 和 M1x 都是用来选择 Tx 的工作模式。GATEx 和 TRx 作为 T0 和 T1 的开关控制，只有在 GATEx=0 且 TRx=1 是 T0 和 T1 才打开。

#### TMCON (8Eh) 定时器频率控制寄存器（读/写）

位编号	7	6	5	4	3	2	1	0
符号	-	-	-	-	-	-	T1FD	T0FD
读/写	-	-	-	-	-	-	读/写	读/写
上电初始值	x	x	x	x	x	x	0	0

位编号	位符号	说明
1	<b>T1FD</b>	T1 输入频率选择控制 0: T1 频率源自于 Fosc/12 1: T1 频率源自于 Fosc/4
0	<b>T0FD</b>	T0 输入频率选择控制 0: T0 频率源自于 Fosc/12 1: T0 频率源自于 Fosc/4
7~2	保留位	保留位

#### IE (A8h) 中断使能寄存器（读/写）

位编号	7	6	5	4	3	2	1	0
符号	EA	EADC	ESIF	EPWM	ET1	-	ET0	ETK
读/写	读/写	读/写	读/写	读/写	读/写	-	读/写	读/写

上电初始值	0	0	0	0	0	x	0	0
-------	---	---	---	---	---	---	---	---

位编号	位符号	说明
3	<b>ET1</b>	Timer1 中断使能控制 0: 关闭 TIMER1 中断 1: 允许 TIMER1 中断
1	<b>ET0</b>	Timer0 中断使能控制 0: 关闭 TIMER0 中断 1: 允许 TIMER0 中断

**IP (B8h) 中断优先级寄存器 (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	-	IPADC	IPSIF	EPWM	IPT1	-	IPT0	ETK
读/写	-	读/写	读/写	读/写	读/写	-	读/写	读/写
上电初始值	x	0	0	0	0	x	0	0

位编号	位符号	说明
3	<b>IPT1</b>	Timer1 中断优先权 0: 设定 Timer 1 的中断优先权是“低” 1: 设定 Timer 1 的中断优先权是“高”
1	<b>IPT0</b>	Timer0 中断优先权 0: 设定 Timer 0 的中断优先权是“低” 1: 设定 Timer 0 的中断优先权是“高”

## 10.2 T0 工作模式

通过对寄存器 TMOD 中的 M10、M00 (TMOD[1]、TMOD[0]) 的设置, 定时器/计数器 0 可实现 4 种不同的工作模式。

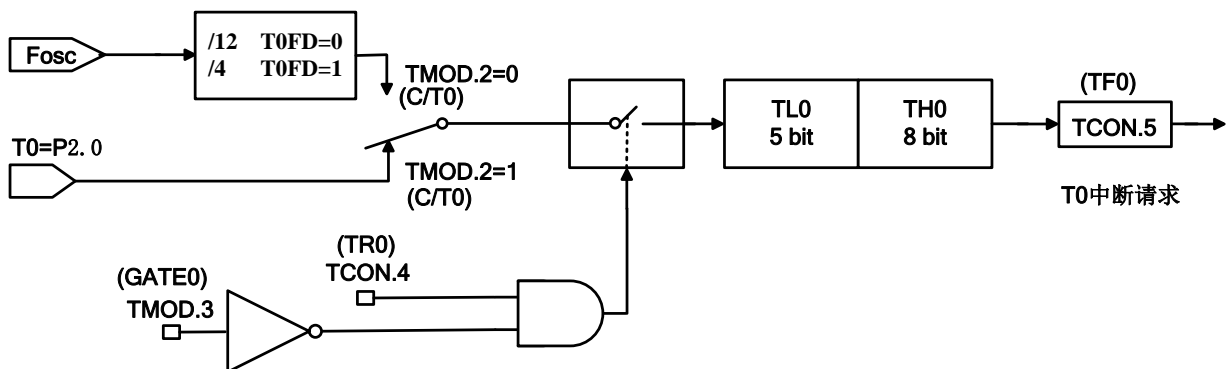
### 工作模式 0: 13 位计数器/定时器。

TH0 寄存器存放 13 位计数器/定时器的高 8 位 (TH0.7~TH0.0), TL0 存放低 5 位 (TL0.4~TL0.0)。TL0 的高三位 (TL0.7~TL0.5) 是不确定值, 读取时应被忽略掉。当 13 位定时器/计数器递增溢出时, 系统会将定时器溢出标志 TF0 置 1。如果定时器 0 中断被允许, 将会产生一个中断。

C/T0 位选择计数器/定时器的时钟输入源。如果 C/T0=1, 定时器 0 输入脚 T0 (P2.0) 的电平从高到低的变化, 会使定时器 0 数据寄存器加 1。如果 C/T0=0, 选择系统时钟的分频为定时器 0 的时钟源。

当 GATE0=0, TR0 置 1 打开定时器 T0。TR0 置 1 并不强行复位定时器, 意味着如果 TR0 置 1, 定时器寄存器将从上次 TR0 清 0 时的值开始计数。所以, 在允许定时器之前, 应该设定定时器寄存器的初始值。

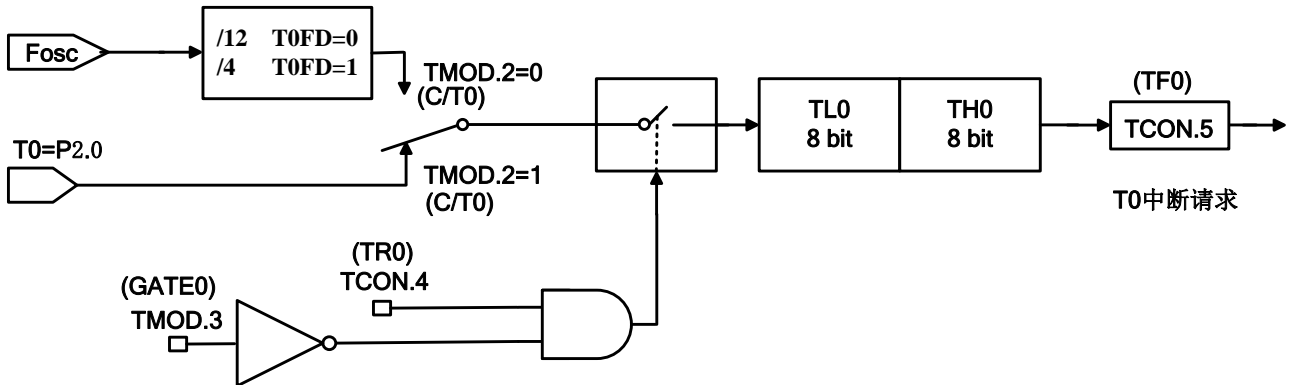
当作为定时器应用时, 可配置 T0FD 来选择时钟源的分频比例。



定时器/计数器工作模式0: 13位定时器/计数器

### 工作模式 1: 16 位计数器/定时器

除了使用 16 位（TL0 的 8 位数据全部有效）计数器/定时器之外，模式 1 和模式 0 的运行方式相同。打开和配置计数器/定时器方式也相同。



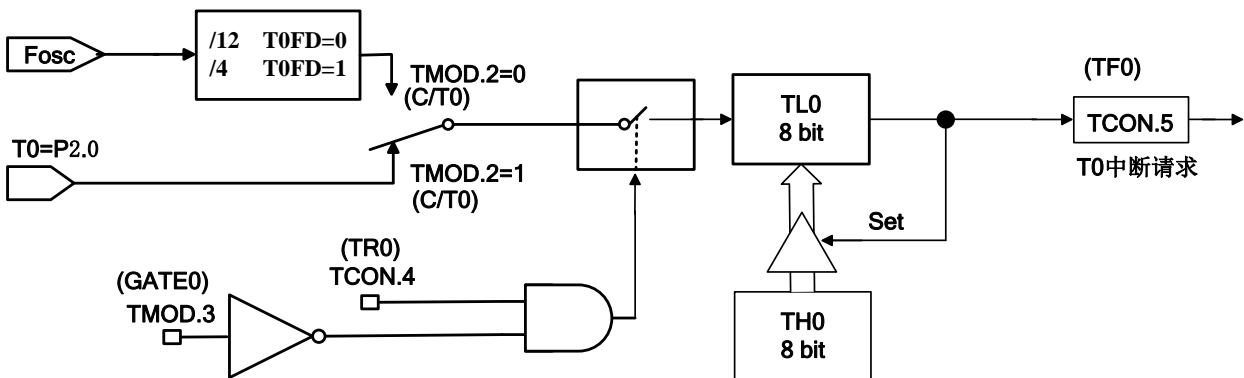
**定时器/计数器工作模式1: 16位定时器/计数器**

### 工作模式 2: 8 位自动重载计数器/定时器

在工作模式 2 中，定时器 0 是 8 位自动重载计数器/定时器。TL0 存放计数值，TH0 存放重载值。当在 TL0 中的计数器溢出至 0x00 时，定时器溢出标志 TF0 被置 1，寄存器 TH0 的值被重载入寄存器 TL0 中。如果定时器中断使能，当 TF0 置 1 时将产生一个中断，但在 TH0 中的重载值不会改变。在允许定时器正确计数开始之前，TL0 必须初始化为所需要的值。

除了自动重载功能外，工作模式 2 中的计数器/定时器的使能和配置方式同模式 0 和 1 是相同的。

当作为定时器应用时，可配置寄存器 TMCON.0(T0FD)来选择定时器时钟源被系统时钟 Fosc 分频的比例。



**定时器/计数器工作模式2: 自动重载的8位定时器/计数器**

### 工作模式 3: 两个 8 位计数器/定时器（仅限于定时器 0）

在工作模式 3 中，定时器 0 用作两个独立的 8 位计数器/定时器，分别由 TL0 和 TH0 控制。TL0 通过定时器 0 的控制位（在 TCON 中）和状态位（在 TMOD 中）：TR0、C/T0、GATE0、TF0 控制。定时器 0 可通过 T0 的 TMOD.2(C/T0)来选择是定时器模式还是计数器模式。

TH0 通过定时器 1 的控制 TCON 来设置相关的控制，但 TH0 仅被限定为定时器模式，无法通过 TMOD.1(T1FD)来设定为计数器模式。TH0 由定时器控制位 TR1 的控制使能，需设定 TR1=1。当发生溢出及产生中断时，TF1 会置 1，并按 T1 发生中断来进行相应的处理。

在 T0 被设为工作模式 3 时，TH0 定时器占用了 T1 的中断资源及 TCON 中寄存器，T1 的 16 位计数器会停止计数，相当于“TR1=0”。当采用 TH0 定时器工作时，需设置 TR1=1。

## 10.3 T1 工作模式

通过对寄存器 TMOD 中的 M11、M01 (TMOD[5]、TMOD[4]) 的设置, 定时器/计数器 1 可实现 3 种不同的工作模式。

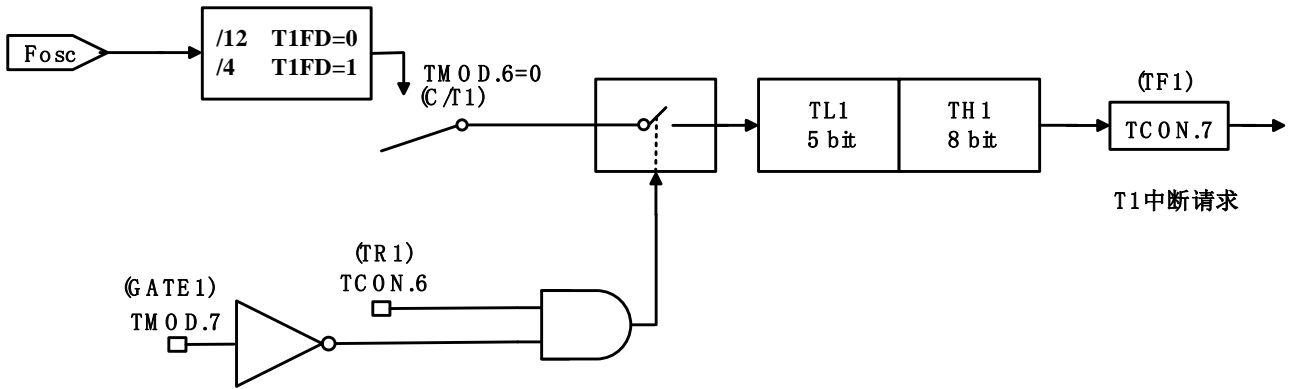
#### 工作模式 0: 13 位计数器/定时器。

TH1 寄存器存放 13 位计数器/定时器的高 8 位 (TH1.7~TH1.0); TL1 存放低 5 位 (TL1.4~TL1.0)。TL1 的高三位 (TL1.7~TL1.5) 是不确定值, 读取时应被忽略掉。当 13 位定时器计数器递增溢出时, 系统会将定时器溢出标志 TF1 置 1。如果定时器 1 中断被允许, 将会产生一个中断。C/T1 位选择计数器/定时器的时钟源。

如果 C/T1=1, 定时器 1 关闭计数功能。如果 C/T1=0, 选择系统时钟的分频为定时器 1 的时钟源。

当 GATE1=0, TR1 置 1 打开定时器。TR1 置 1 并不强行复位定时器, 意味着如果 TR1 置 1, 定时器寄存器将从上次 TR1 清 0 时的值开始计数。所以, 在允许定时器之前, 应该设定定时器寄存器的初始值。

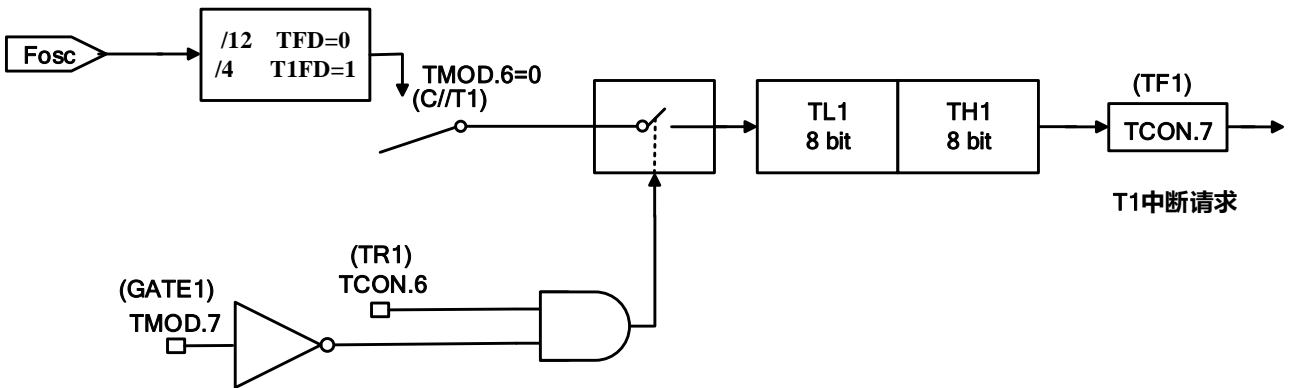
当作为定时器应用时, 可配置 T1FD 来选择时钟源的分频比例。



**定时器/计数器工作模式0: 13 位定时器/计数器**

#### 工作模式 1: 16 位计数器/定时器

除了使用 16 位 (TL1 的 8 位数据全部有效) 计数器/定时器之外, 模式 1 和模式 0 的运行方式相同。打开和配置计数器/定时器方式也相同。



**定时器/计数器工作模式1: 16 位定时器/计数器**

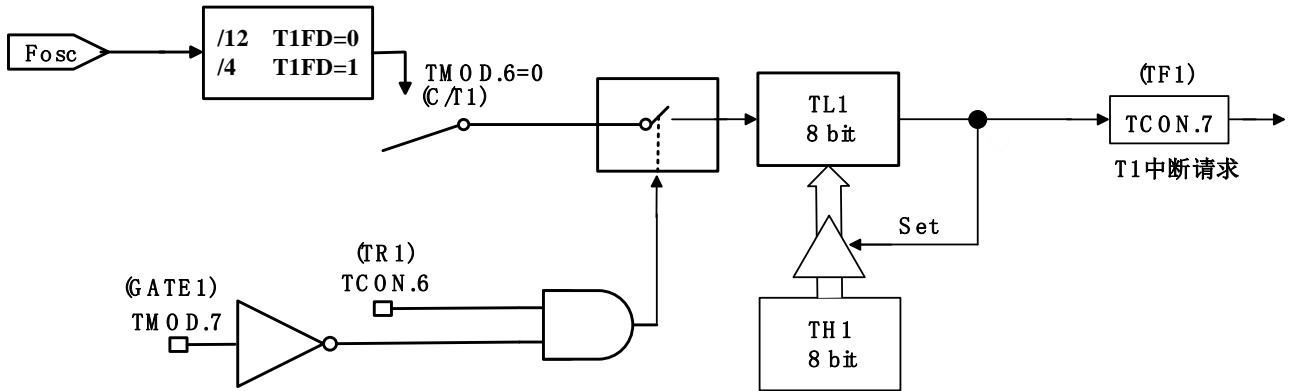
#### 工作模式 2: 8 位自动重载计数器/计数器

在工作模式 2 中, 定时器 1 是 8 位自动重载计数器/定时器。TL1 存放计数值, TH1 存放重载值。当在 TL1 中的计数器溢出至 0x00 时, 定时器溢出标志 TF1 被置 1, 寄存器 TH1 的值被重载入寄存器 TL1 中。如果定时器中断使能, 当 TF1 置 1 时将产生一个中断, 但在 TH1 中的重载值不会改变。在允许定时器正确计数开始之前, TL1 必须初始化为所需要的值。

除了自动重载功能外, 工作模式 2 中的计数器/定时器的使能和配置方式同方式 0 和 1 是相同的。

当作为定时器应用时, 可配置寄存器 TMCON.4(T1FD)来选择定时器时钟源被系统时钟 Fosc 分频的比例。





定时器/计数器工作模式2: 自动重载的8位定时器/计数器

## 11 PWM

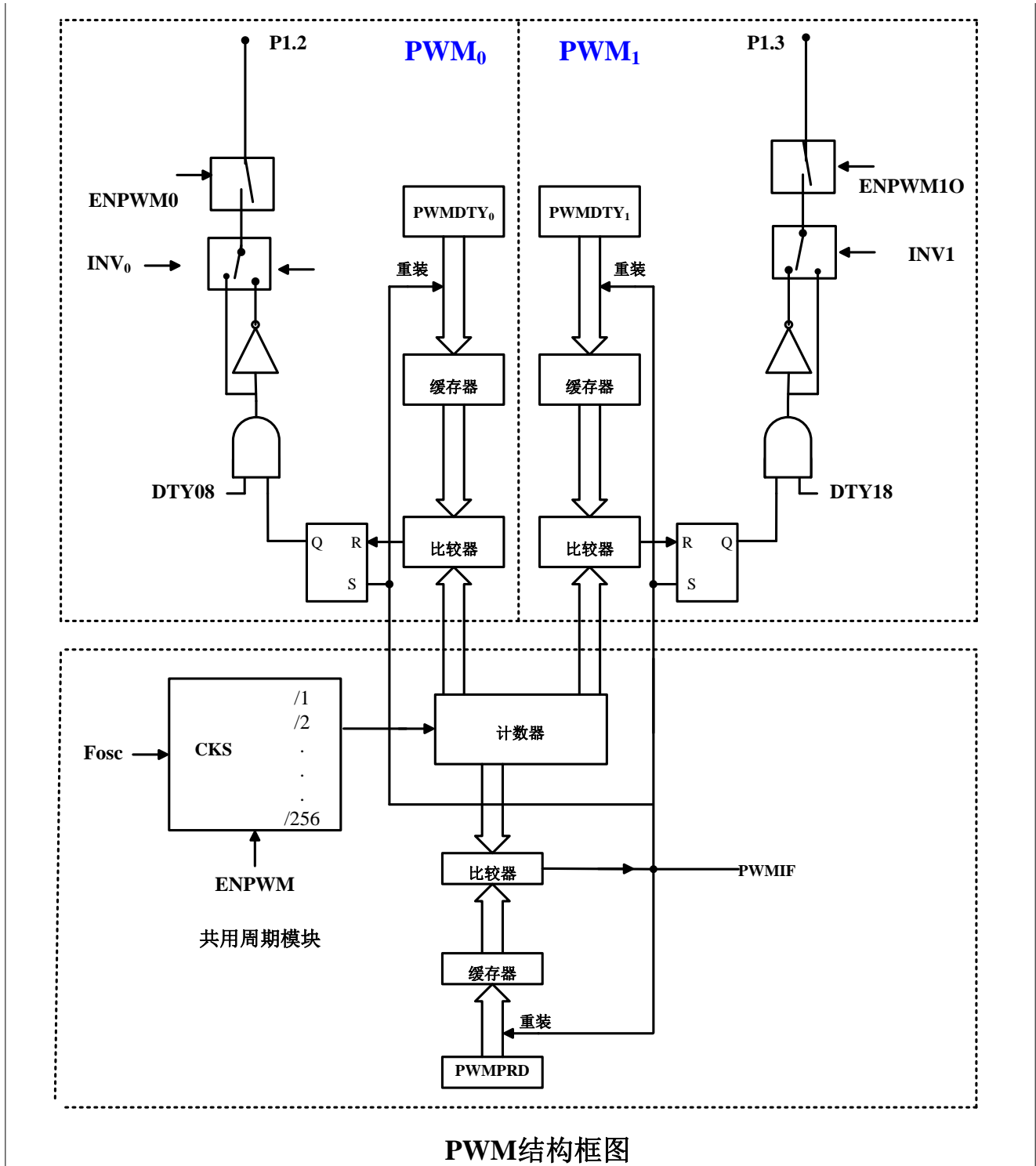
SC91F830 提供了一个独立的计数器, 它可以支持两路的 PWM 输出: PWM0 及 PWM1。

SC91F830 的 PWM 具有的功能为:

- ① 8 位 PWM 精度
- ② PWM0/PWM1 周期相同, 但占空比可单独设置
- ③ 输出可设置正反向
- ④ 提供 1 个 PWM 溢出的中断。
- ⑤ 注意: 程式从 IO 切换到 PWM 输出后, IO 不会自动切换到强推挽输出 PWM, 而是保持切换之前的 IO 模式设置。如果用户需要强推挽 PWM, 需要用户先将 IO 设置为强推挽模式

SC91F830 的 PWM 可支持周期及占空比的调整, 寄存器 PWMCR 控制 PWM0 和 PWM1 相关设置, PWMCFG 设置 PWM 计数器计数时钟源及输出电平, PWMPRD 设置两个 PWM 共同的周期, PWMDTY0 和 PWMDTY1 分别控制 PWM0 和 PWM1 的占空比。

### 11.1 PWM 结构框图



## 11.2 PWM 相关 SFR 寄存器

**PWMCR (F8h) PWM 控制寄存器 (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	ENPWM	PWMIF	-	-	DTY18	ENPWM1O	DTY08	ENPWM0O
读/写	读/写	读/写	-	-	读/写	读/写	读/写	读/写
上电初始值	0	0	x	x	0	0	0	0

位编号	位符号	说明
7	<b>ENPWM</b>	PWM 模块开关控制(Enable PWM) 1: 允许 Clock 进到 PWM 单元, 开始 PWM 的工作 0: 关闭 PWM 单元的工作, 主要是为了省电
6	<b>PWMIF</b>	PWM 中断请求标志位(PWM Interrupt Flag) 当 PWM 计数器溢出时(也就是说: 数到超过 PWMPRD 时), 此位会被硬件自动设定成 1。如果此时 IE[5] (EPWM) 也是被设定成 1, PWM 的中断产生。在 PWM 中断发生后, 硬件并不会自动清除此位, 此位必须由使用者的软件负责清除。
3	<b>DTY18</b>	强制 PWM1 固定输出高 (Force PWM1 as HIGH) 1: 强制把 PWM1 的输出固定为 1 0: PWM1 的输出由 PWM 计数器以及 PWMDTY1 来决定
2	<b>ENPWM1O</b>	PWM1 功能开关 1: 允许 PWM1 输出到 P1.3 0: PWM1 不输出到 P1.3
1	<b>DTY08</b>	强制 PWM0 固定输出高 (Force PWM0 as HIGH) 1: 强制把 PWM0 的输出固定为 1 0: PWM0 的输出由 PWM 计数器以及 PWMDTY0 来决定
0	<b>ENPWM0O</b>	PWM0 功能开关 1: 允许 PWM0 输出到 P1.2 0: PWM0 不输出到 P1.2
5,4	<b>保留位</b>	保留位

SFR **PWMPRD[7:0]** 是两路 PWM 共享的周期设置控制器。每当 PWM 计数器数到 **PWMPRD[7:0]** 预先设置的值时, 下一个 PWM CLK 到来时该计数器会跳数到 00h, 也就是说 PWM1 及 PWM0 的周期都是  $(PWMPRD[7:0] + 1) * PWM$  时钟。

**PWMPRD (F9h) PWM 周期设置寄存器 (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	<b>PWMPRD[7:0]</b>							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
上电初始值	1	1	1	1	1	1	1	1

位编号	位符号	说明
7~0	<b>PWMPRD[7:0]</b>	PWM0 和 PWM1 共用的周期设置; 此数值代表 PWM0 和 PWM1 输出波形的 (周期 - 1); 也就是说 PWM 输出的周期值为 $(PWMPRD[7:0] + 1) * PWM$ 时钟;

PWM 计数器的计数时间可由 **PWMCFG[2:0]** 所控制, 分别可以选择不同个数的系统时钟去计数一个单位 (pre-scalar selector), 即选择 PWM 计数器时钟源被系统时钟  $F_{osc}$  分频的分频比。PWM1 和 PWM0 还可以被 **PWMCFG[5:4]** 中的 INV1 和 INV0 来选择, PWM 输出是否反向。

**PWMCFG (FCh) PWM 设置寄存器 (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	-	-	INV1	INV0	-	CKS[2:0]		

读/写	-	-	读/写	读/写	-	读/写		
上电初始值	x	x	0	0	x	0	0	0

位编号	位符号	说明
5	<b>INV1</b>	PWM1 输出反向控制(INVerse PWM1 Output) 1: 把 PWM1 的输出反向 0: PWM1 的输出不反向
4	<b>INV0</b>	PWM0 输出反向控制(INVerse PWM0 Output) 1: 把 PWM0 的输出反向 0: PWM0 的输出不反向
2~0	<b>CKS</b>	PWM 时钟源选择(PWM ClocK source Selector) 000: Fosc 001: Fosc/2 010: Fosc/4 011: Fosc/8 100: Fosc/32 101: Fosc/64 110: Fosc/128 111: Fosc/256
7,6,3	保留位	保留位

**PWMDTY1 (FAh) PWM1 高电平设置寄存器 (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	<b>PWMDTY1[7:0]</b>							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
上电初始值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7~0	<b>PWMDTY1[7:0]</b>	PWM1 占空比长度设置; PWM1 的高电平宽度是 (PWMDTY1[7:0])个 PWM 时钟

**PWMDTY0 (FBh) PWM0 高电平设置寄存器 (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	<b>PWMDTY0[7:0]</b>							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
上电初始值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7~0	<b>PWMDTY0[7:0]</b>	PWM0 占空比长度设置; PWM0 的高电平宽度是 (PWMDTY0[7:0])个 PWM 时钟

**IE (A8h) 中断使能寄存器 (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	EA	EADC	EPWM	ESIF	ET1	-	ET0	ETK
读/写	读/写	读/写	读/写	读/写	读/写	-	读/写	读/写
上电初始值	0	0	0	0	0	x	0	0

位编号	位符号	说明
5	<b>EPWM</b>	PWM 中断使能控制

		0: 关闭 PWM 中断 1: 允许 PWM 计数器溢出时产生中断
--	--	--------------------------------------

**IP (B8h) 中断优先级寄存器(读/写)**

位编号	7	6	5	4	3	2	1	0
符号	-	IPADC	IPPWM	IPSIF	IPT1	-	IPT0	IPTK
读/写	-	读/写	读/写	读/写	读/写	-	读/写	读/写
上电初始值	x	0	0	0	0	x	0	0

位编号	位符号	说明
5	<b>IPPWM</b>	PWM 中断优先级选择 0: 设定 PWM 的中断优先级是“低” 1: 设定 PWM 的中断优先级是“高”

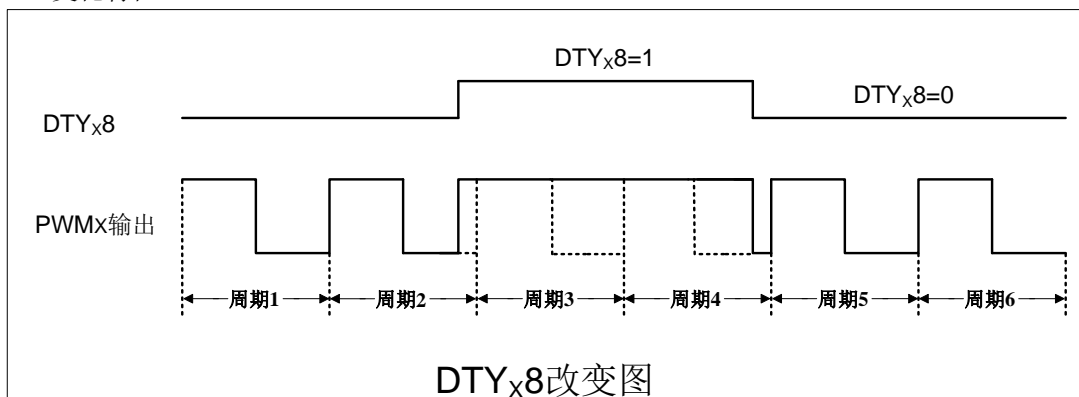
注意事项:

1. ENPWM 位能控制 PWM 模块是否工作。
2. ENPWMxO 位能选择 PWMx 口作为 GPIO 还是作为 PWMx 输出。
3. EPWM(IE.4)位能控制 PWM 是否被允许产生中断。
4. 如果 ENPWM 置 1, PWM 模块被打开, 但 PWMxO=0,两个 PWM 输出被关闭并作为 GPIO 口。此时 PWM 模块可以作为一个 8 位 Timer 使用, 此时 EPWM(IE.4)被置 1, PWM 仍然会产生中断。
5. 两个 PWM 共用周期, 溢出时产生 PWM 中断是同一中断向量。

### 11.3 PWM 波形及用法

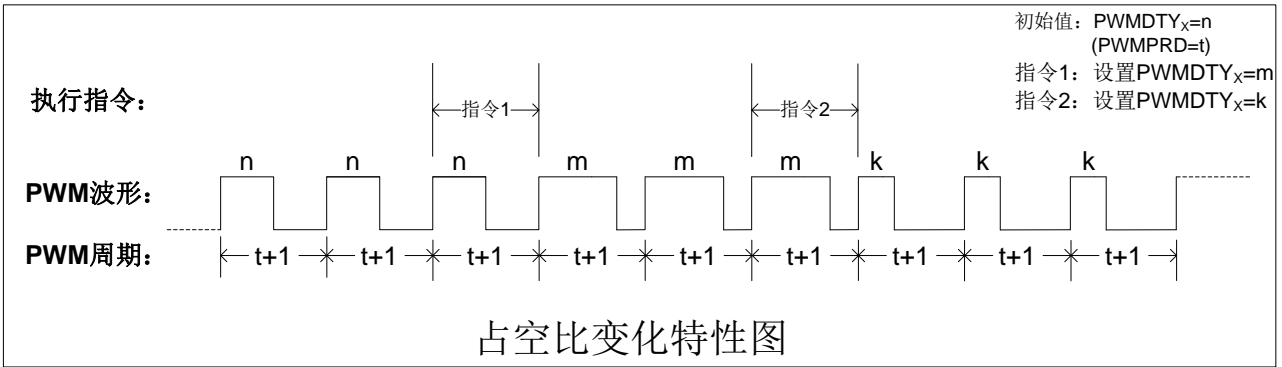
各 SFR 参数改变对 PWM 波形影响如下所述:

①DTYx8 变化特性



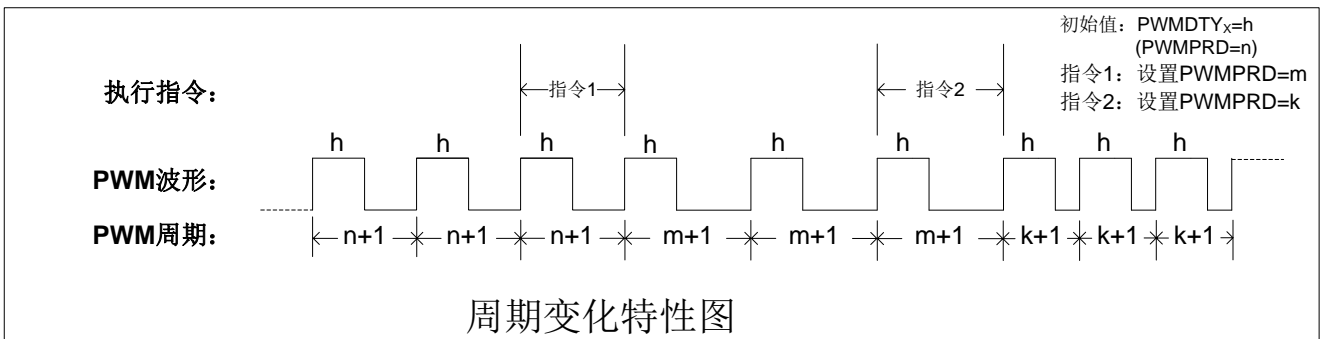
当 PWMx 输出波形时, 若 DTYx8(PWMCR.1/ PWMCR.3)改变, PWMx 波形会立即改变。如上图所示, 在周期 2 中使 DTYx8 置 1, PWMx 会立即响应, 固定输出高; 在周期 4 某处 DTYx8 清 0, PWMx 立即响应, 取消固定输出。

②占空比变化特性



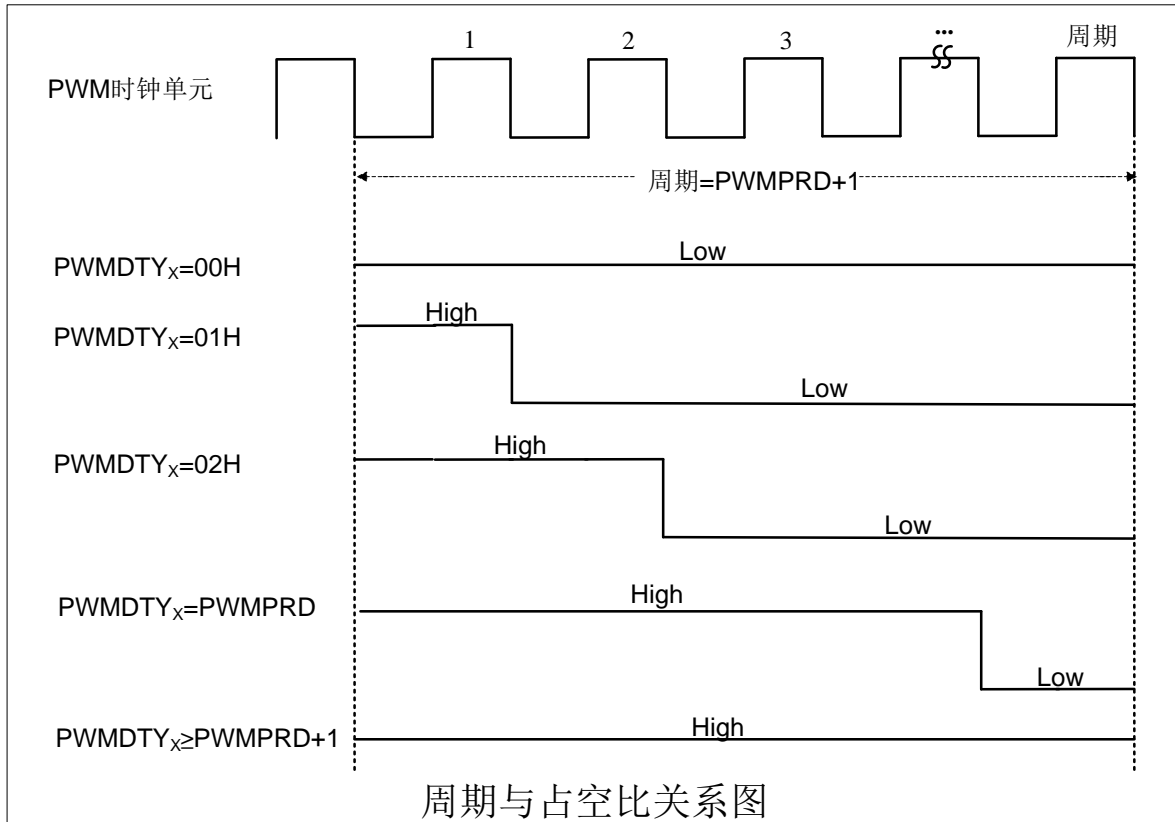
当 PWMx 输出波形时，若需改变占空比，可通过改变高电平设置寄存器(PWMDTYx)的值实现。但需要注意，更改 PWMDTYx 的值，占空比不会立即改变，而是等待本周期结束，在下一个周期发生改变。相关波形输出如上图所示。

### ③周期变化特性



当 PWMx 输出波形时，若需改变周期，可通过改变周期设置寄存器 PWMPRD 的值实现。同改变占空比一样，更改 PWMPRD 的值，周期不会立即改变，而是等待本周期结束，在下一个周期改变，参考上图所示。

### ④周期和占空比的关系



周期和占空比的关系如上图所示。该结果的前提是  $PWMx(x=0, 1)$  输出反向控制(INVx)初始为 0，若需得到相反结果，可置  $PWMCFG.5(INV1)$ 和  $PWMCFG.4(INV0)$ 为 1。需要注意  $INVx$  的变化特性也  $DTYx8$  相同，更改则立即响应。

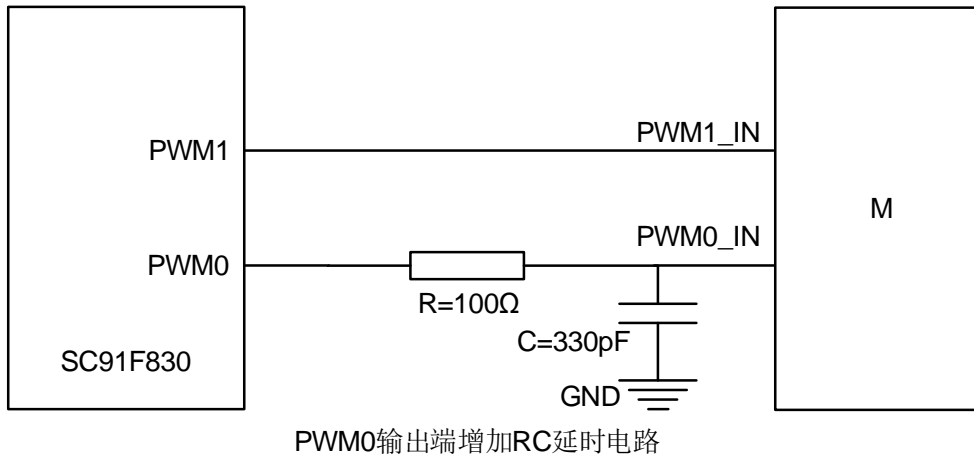
### 11.4 两路互补带死区 PWM 实现方法

大功率电机、变频器的末端都是由大功率管、IGBT（绝缘栅极型功率管）等元件组成的 H 桥。每个桥的上半桥和下半桥是绝对不能同时导通的。但由于 IGBT 等功率器件都存在一定的结电容，会形成器件导通关断的延迟现象，导致某个半桥元件在应该关断时没有关断，烧毁功率元件。

为了使 H 桥或半 H 桥的上下管不会因为开关速度问题发生同时导通，需要在电平翻转时插入一个时间间隔，这个时间间隔就是 PWM 输出时的死区。

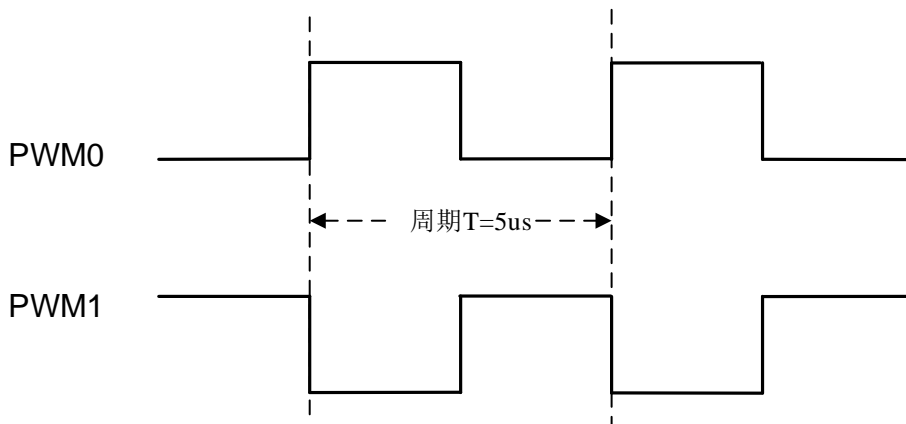
SC91F830 可以通过两路独立的 PWM0 和 PWM1 输出互补带死区的 PWM 波形。

下面以输出两路频率为 200kHz，占空比约为 50%的互补 PWM 方波，并产生一个不小于 5nS 的死区为例说明具体的实现方法（IC 工作频率 16MHz，VDD 为 5V，外围器件 M 的  $V_{IH}$  为 0.7VDD,  $V_{IL}$  为 0.3VDD）。



①输出两路互补的，频率为 **200kHz**，占空比 **1:2** 的 PWM 方波：

```
void PWMinit(void)
{
    PWMCFG = 0x20;           // PWM0 输出到 P1.2, PWM1 反向输出到 P1.3
                            // 选择 PWM 时钟源为 fosc = 16MHz
    PWMPRD = 79;            // PWM 周期 T=(79+1)*(1/16M)=5us, 200kHz
    PWMDTY0 = 40;           // PWM0 的 Duty 设为 1:2
    PWMDTY1 = 40;           // PWM1 的 Duty 设为 1:2
    PWMCR = 0x85;           // 打开 PWM 电源, 使能 PWM0 和 PWM1
}
```



PWM0及PWM1输出波形图

②修改 **PWMDTY0** 调节 **PWM0** 占空比，形成后死区延时  $t_1$

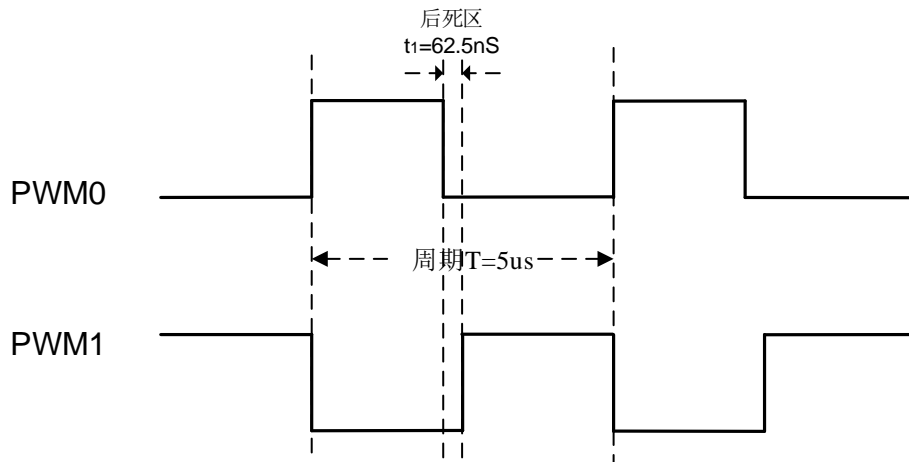
```
PWMDTY0 = 39;           // PWM0 的 Duty 设为 39:80 形成后死区延时
```

PWM 的时钟源为 16MHz，则每一个 Duty 可调节：

$$\frac{1}{16M} = 62.5ns$$



PWM0 由 40 变为 39，PWM0 输出高电平的时间减少了 62.5ns。



PWM0及PWM1输出波形图

③在 PWM0 上加上 RC 电路，其中  $R=100\Omega$ ， $C=330pF$

则有：

$$\tau = 100\Omega \times 330pF = 33ns$$

上升沿延时：

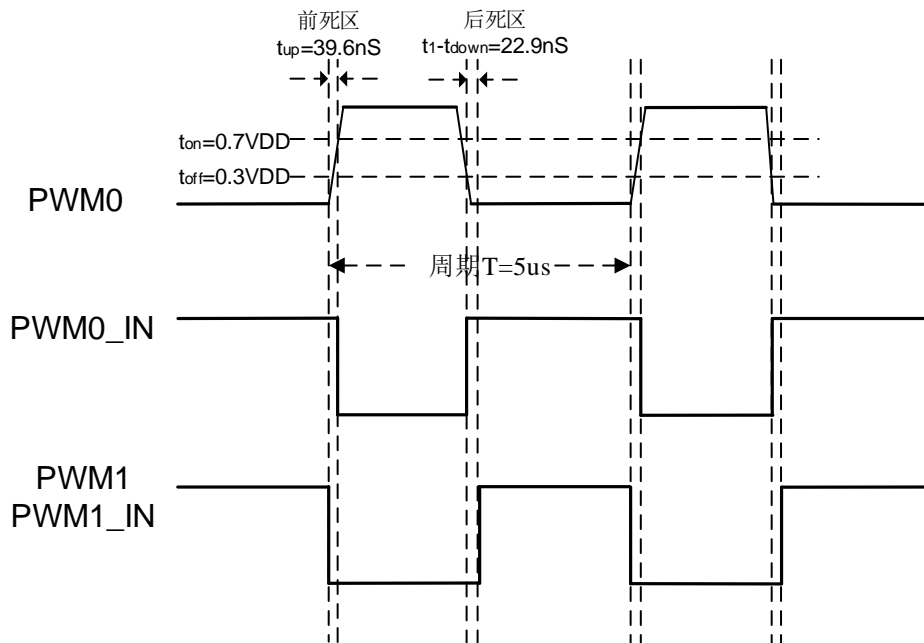
$$\begin{aligned} t_{up} &= \tau \times \ln\left(\frac{5V}{5V - 0.7 \times 5V}\right) \\ &= 33ns \times 1.2 = 39.6ns \end{aligned}$$

下降沿延时：

$$\begin{aligned} t_{down} &= \tau \times \ln\left(\frac{5V}{0.3 \times 5V}\right) \\ &= 33ns \times 1.2 = 39.6ns \end{aligned}$$

此时后死区延时变为：

$$t_1 - t_{down} = 62.5ns - 39.6ns = 22.9ns$$



PWM0及PWM1输出波形图

由此得到两路前死区延时约 39.6ns，后死区延时约 22.9ns 的互补 PWM 方波。

#### 死区延时的取值

死区时间大，系统工作更加可靠，但是会引起输出波形失真并降低工作效率；死区时间小，则效率高，但是会降低可靠性。一般功率器件的输入端死区延时都控制在几十 ns 或者根据管子特性作调整。

## 11.5 PWM 频率高精度步进调整实现方法

SC91F830 可以输出最高达 8MHz 的 PWM 频率，但 PWM 频率较高时，通过改变 PWMPRD 无法实现较高精度 PWM 步进调整。也就是说：PWM 频率越高，则 PWMPRD 值越小，对 PWMPRD 加减 1 带来的 PWM 输出频率变化越大，PWM 输出的步进精度就会较差。

SC91F830 提供一种通过改变 IRC 频率来实现高速 PWM 频率高精度步进调整的方法，步进精度小于 0.4%。用户可通过修改 IRCC 寄存器的值改变 IRC 和 PWM 的频率输出。请用户确认 IRC 频率的改变不会影响其它功能。

#### IRCC (EFh) 系统时钟改变寄存器（读/写）

位编号	7	6	5	4	3	2	1	0
符号	-	IRCC[6:0]						
读/写	-	读/写						
上电初始值	-	n	n	n	n	n	n	n

IRC 频率输出、PWM 频率输出 和 IRCC [6:0]的关系如下：

IRCC[6:0]值	IRC 实际输出频率（16M 为例）	PWMPRD 值	PWM 频率
IRCC[s]-n	$(16000-n*62.5)$ kHz	P	$(16000-n*62.5)/(P+1)$ kHz
...	...	P	...
IRCC[s]-2	$16000-125=15875$ kHz	P	$15875/(P+1)$ kHz
IRCC[s]-1	$16000-62.5=15937.5$ kHz	P	$15937.5/(P+1)$ kHz
IRCC[s]	16000kHz	P	$16000/(P+1)$ kHz
IRCC[s]+1	$16000+62.5=16062.5$ kHz	P	$16062.5/(P+1)$ kHz
IRCC[s]+2	$16000+125=16125$ kHz	P	$16125/(P+1)$ kHz

...	...	P	...
IRCC[s]+n	(16000+n*62.5)kHz	P	(16000+n*62.5)/(P+1)kHz

注：IRC 出厂设定的 16MHz 和步进的 62.5kHz，每颗 IC 会略有偏差。

以常用的 1.7M/2.4M/3MHz 几个 PWM 中心频率为例，说明用户 PWM 参数选择、IRC 工作频率和 PWM 频率输出的关系如下：

参数/PWM 频率	1.7MHz±5%	2.4MHz±5%		3MHz±5%
PWMPRD 最佳值	8	6	5	4
IRC 工作频率范围	15.3MHz±5% (14.54MHz ~ 16.07MHz)	16.8MHz±5% (最优)	14.4MHz±5% (次优)	15MHz±5% (14.25MHz ~ 15.75MHz)
IRCC 偏移量	-24 ~ 1	-1 ~ 27	-38 ~ -14	-27 ~ -4
IRCC[6:0] 值的有效范围	0~127	0~127 超界的 IC 请改变 PWMPRD=5	0~127	0~127

说明：在 PWM 中心频率为 1.7MHz 和 3MHz 的 ±5% 范围内，都能找到最优的 PWMPRD 参数，确保 IC 可以调到精准的频率输出；在 PWM 中心频率为 2.4MHz 的 ±5% 范围内，PWMPRD 最优值为 6，非常少量的调整不到最佳频率的 IC（如果 IRCC[6:0]修正后的值超过 127 仍然无法得到最佳的 PWM 输出频率值时），说明 IRCC[s] 偏大，可通过选择改变 PWMPRD=5，将 IRCC 值负向调整就一定可以实现 PWM 频率的精准输出。

下表以 PWMPRD=6 为例说明中心频率为 2.4MHz 的 PWM 频率步进输出频率和参数的关系，可以看出 PWM 频率在 2.4MHz 的 ±5% 范围内可实现 0.9kHz 左右的步进改变，步进调整精度约为 0.9/24000=0.38%。

十进制 IRCC 偏移量 (IRCC[6:0]-RCC[s])	0	1	...	11	12	13	14	15	...	25	26
IRC 频率值 (kHz)	16000	16062.5	...	16687.5	16750	16812.5	16875	16937	...	17562.5	17625
PWM 周期 (PWMPRD+1)	7	7	...	7	7	7	7	7	...	7	7
PWM 频率输出(kHz)	2285.71	2294.64	...	2383.93	2392.86	2401.79	2410.71	2419.64	...	2508.93	2517.86

用户可进行以 2.4MHz 为中心频率的 PWM 步进调整，具体做法步骤为：

1. 读取 IRCC[6:0] 上电初始值 IRCC[s]；
2. 设置相关的 PWM 寄存器，特别提出 PWMPRD+1 的值选择 7 较合适（PWMPRD=6）；
3. 根据所需要的 PWM 频率，将对应的 IRCC[6:0] 值（IRCC[s]+偏移量）写入 IRCC 寄存器并打开 PWM 即可得到相应的 PWM 频率输出。
4. 如果对应的 IRCC[6:0] 的值超过 0~127 的范围（只可能是大于 127，说明 IRCC[s] 偏大），则请将 PWMPRD+1 的值调整到 6 后，再计算此时所需要的 IRCC 偏移量及对应的 IRCC[6:0] 的值，通过将 IRCC[6:0] 的值负向调整，此时一定可以得到用户希望的 PWM 频率输出（此时 IRC 频率降低较多，请特别留意）。

用户如要得到非 2.4MHz 的其它频率也可用相同的方法通过先确定 PWMPRD，再改变 IRCC 偏移量实现。如果来实现更高精度的调整，用户可以通过同时改变 PWMPRD 和 IRCC 两个参数来得到更高精度的 PWM 步进频率，如有以上需求，请与赛元工程师联系。

## 12 GP I/O

SC91F830 提供了最多 13 个 GPIO 端口，此 13 个 IO 同其他功能复用，其中 P1 和 P2 这 8 个 IO 口的灌电流驱动能力约是其它 IO 口的 2 倍，可用来作为 LED 显示的 COM 驱动。SC91F830 的 I/O 口和标准 8051 的 I/O 口一样，是带强推挽输出的双向 IO 口，有四种 IO 模式可以选择：准双向 IO 模式、强推挽输出模式、高阻仅输入、N 型开漏输出模式。P1 口上电的缺省模式为纯输入模式。

**准双向 IO 结构：**也就是说，当对一个 I/O 口写“0”时，它有很强 (>15mA) 的推低 (Sink) 能力，并且使用者此时应该把该 I/O 口视为输出“Output”，但是如果对该 I/O 口写“1”后，该 I/O 口会有短暂的强上拉 (两个 clock 周期)，之后就一直以弱上拉的状态保持该 I/O 口为高输出，而此时允许使用者从外部输入信号，盖过该弱上拉。简单说，写“0”代表强的“Output”，写“1”代表“Input”。

**强推挽结构：**若是有特别的强推挽输出需要，使用者可以设定对应的寄存器 PxCFGn，让 I/O 口输出很强的“1”，

而非“允许被盖过的 1”，且具有较强的电流驱动能力。

高阻仅输入结构：高阻态，仅作为口输入使用。

N 型开漏输出结构：IC 内部的上拉电阻断开，需要外接上拉电阻。

## 12.1 GPIO 结构图

### 1. 准双向模式 (Quasi-Bi)

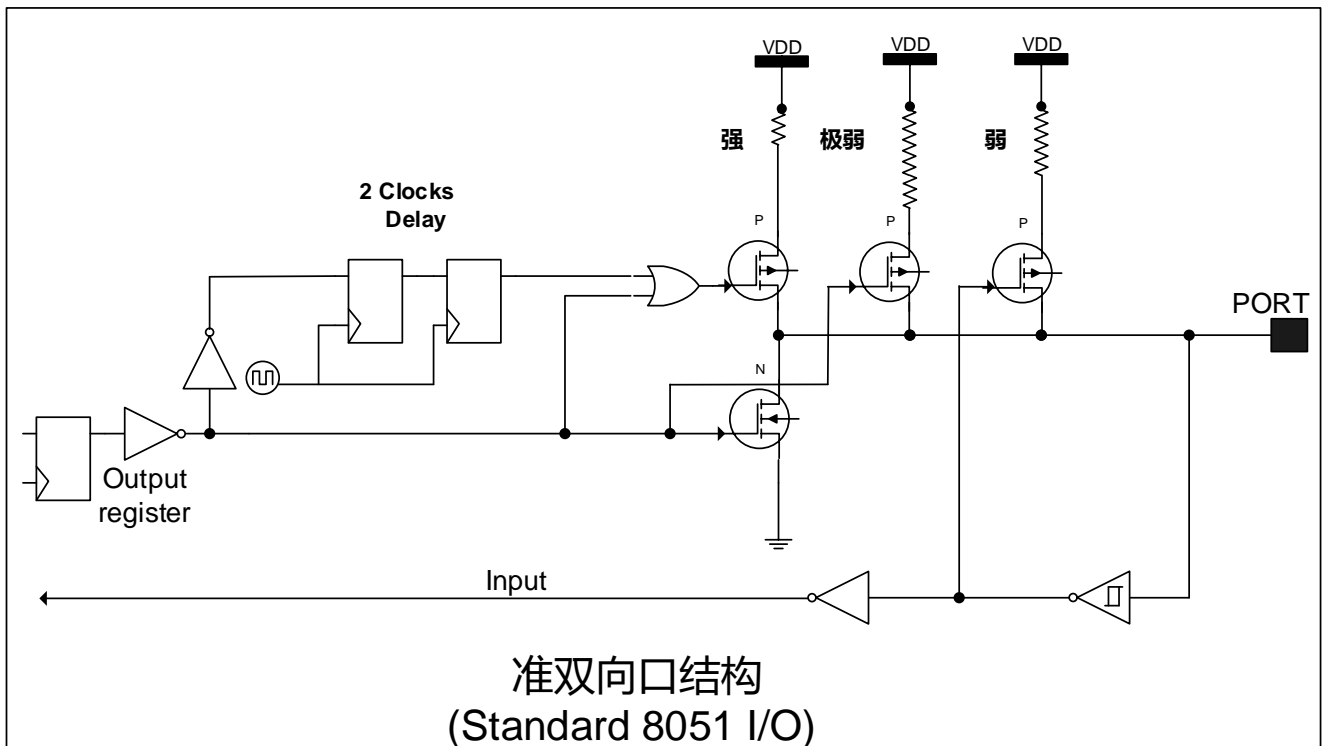
准双向口有 3 个上拉 MOS 管以适应不同的需要，分别称为“弱 (Weak) 上拉”、“极弱 (Very weak) 上拉”和“强 (Strong) 上拉”。

在 3 个上拉 MOS 管中，有 1 个上拉 MOS 管称为“弱上拉”，当口线寄存器为 1 且引脚本身也为 1 时打开。此上拉提供基本驱动电流使准双向口输出为 1。如果 1 个引脚输出为 1 而由外部装置下拉到低时，弱上拉关闭而“极弱上拉”维持开状态，为了把这个引脚强拉为低，外部装置必须有足够的灌电流能力使引脚上的电压降到门槛电压以下。

第 2 个上拉 MOS 管称为“极弱上拉”，当口线锁存为 1 时打开。当引脚悬空时，这个极弱的上拉源产生很弱的电流将引脚上拉为高电平。

第 3 个上拉 MOS 管称为“强上拉”，当口线锁存器由 0 跳变为 1 时，这个上拉用来加快准双向口由逻辑 0 到逻辑 1 转换。当发生这种情况时，强上拉打开约 2 个机器周期以使引脚能迅速地上拉到高电平。

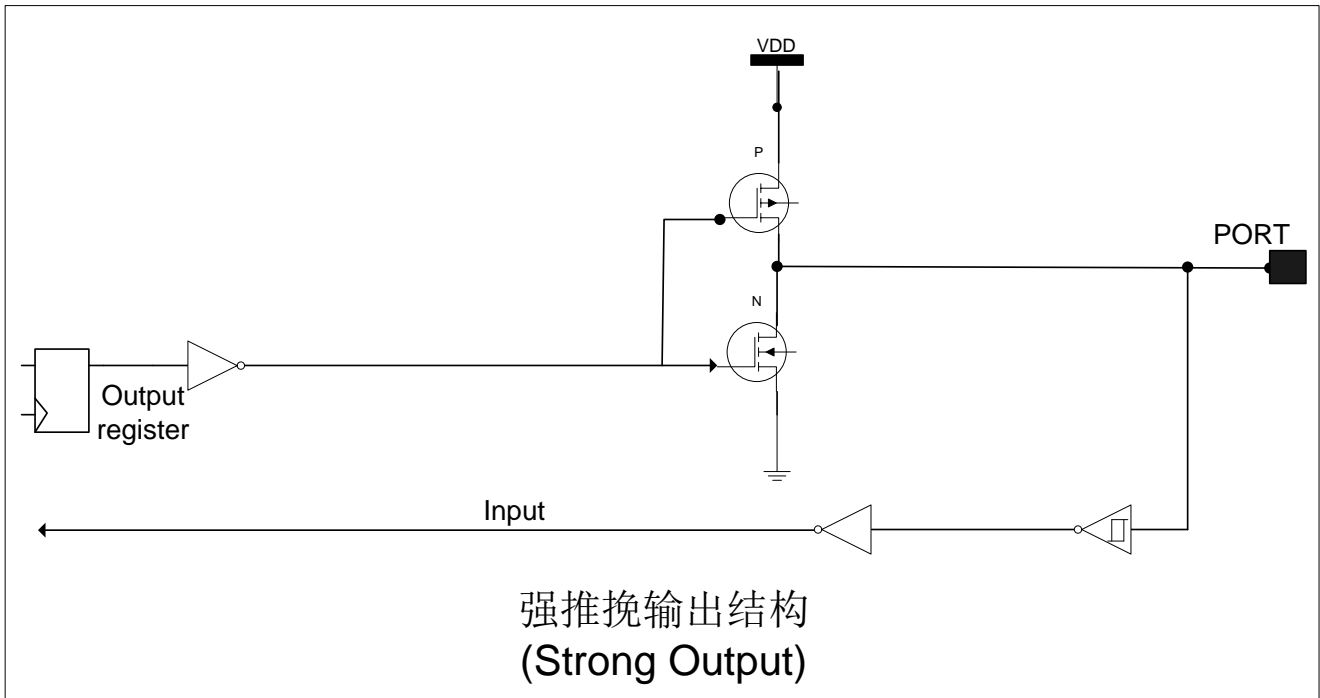
准双向模式的端口结构示意图如下：



## 2. 强推挽输出模式

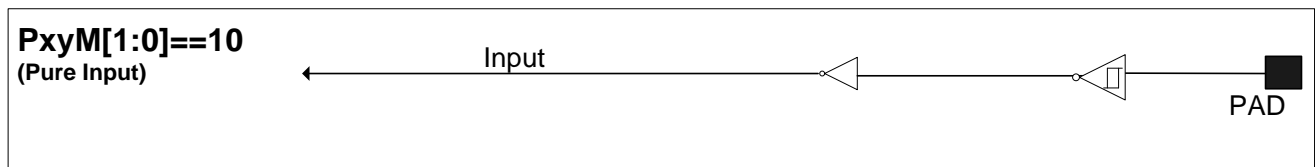
强推挽输出配置的下拉结构与开漏输出与准双向口的下拉结构相同，但当锁存器为 1 时能提供持续的强上拉，即能够提供持续的大电流驱动（大于 15mA）的输出高。

强推挽输出模式的端口结构示意图如下：



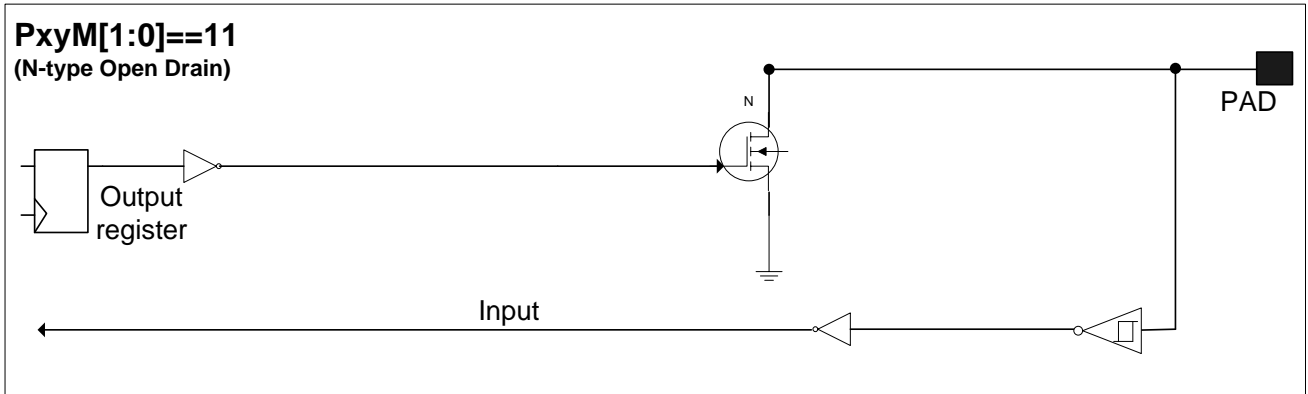
## 3. 仅输入模式 (Input only) 高阻

此种模式仅有输入，没有输出能力。仅输入模式的端口结构示意图如下所示：



## 4. 开漏输出模式 (Open Drain)

此种模式没有输出高的能力。如果需要输出高，用户必须外接上拉电阻。此时外加引脚的电压不能超过 VDD+0.3V。开漏输出模式的端口结构示意图如下：



开漏输出结构  
(Open drain)

## 12.2 I/O 端口相关寄存器

### P1CFG1 (91h) P1 口模式配置寄存器(读/写)

位编号	7	6	5	4	3	2	1	0
符号	P17M[1:0]		P16M[1:0]		-		-	
读/写	读/写	读/写	读/写	读/写	-	-	-	-
上电初始值	1	0	1	0	x	x	x	x

### P1CFG0 (92h) P1 口模式配置寄存器(读/写)

位编号	7	6	5	4	3	2	1	0
符号	P13M[1:0]		P12M[1:0]		-		P10M[1:0]	
读/写	读/写	读/写	读/写	读/写	-	-	读/写	读/写
上电初始值	1	0	1	0	x	x	1	0

### P2CFG1 (A1h) P2 口模式配置寄存器(读/写)

位编号	7	6	5	4	3	2	1	0
符号	P27M[1:0]		-		-		P24M[1:0]	
读/写	读/写	读/写	-	-	-	-	读/写	读/写
上电初始值	0	0	x	x	x	x	0	0

### P2CFG0 (A2h) P2 口模式配置寄存器(读/写)

位编号	7	6	5	4	3	2	1	0
符号	-		-		-		P20M[1:0]	
读/写	-	-	-	-	-	-	读/写	读/写
上电初始值	x	x	x	x	x	x	0	0

### P3CFG1 (B1h) P3 口模式配置寄存器(读/写)

位编号	7	6	5	4	3	2	1	0
符号	-		P36M[1:0]		P35M[1:0]		-	
读/写	-	-	读/写	读/写	读/写	读/写	-	-
上电初始值	x	x	0	0	0	0	x	x

### P3CFG0 (B2h) P3 口模式配置寄存器(读/写)

位编号	7	6	5	4	3	2	1	0
符号	-		P32M[1:0]		P31M[1:0]		P30M[1:0]	
读/写	-	-	读/写	读/写	读/写	读/写	读/写	读/写

上电初始值	x	x	0	0	0	0	0	0
-------	---	---	---	---	---	---	---	---

位编号	位符号	说明
7,6,3,2,0	<b>P1xM[1:0]</b> (x=0/2/3/6/7)	P1 口模式配置 00: P1x 的特性同标准 8051 MCU 的 I/O 口, 即是准双向 IO 口 01: 强制 P1x 口 有很强的输出高驱动能力 10: P1x 为高阻仅输入模式 (上电初始值) 11: P1x 为 N 型开漏模式
7,4,0	<b>P2xM[1:0]</b> (x=0/4/7)	P2 口模式配置 00: P2x 的特性同标准 8051 MCU 的 I/O 口, 即是准双向 IO 口 01: 强制 P2x 口 有很强的输出高驱动能力 10: P2x 为高阻仅输入模式 11: P2x 为 N 型开漏模式
6,5,2,1,0	<b>P3xM[1:0]</b> (x=0/1/2/5/6)	P3 口模式配置 00: P3x 的特性同标准 8051 MCU 的 I/O 口, 即是准双向 IO 口 01: 强制 P3x 口 有很强的输出高驱动能力 10: P3x 为高阻仅输入模式 11: P3x 为 N 型开漏模式

**P1 (90h) P1 口数据寄存器(读/写)**

位编号	7	6	5	4	3	2	1	0
符号	P1.7	P1.6	-	-	P1.3	P1.2	-	P1.0
读/写	读/写	读/写	-	-	读/写	读/写	-	读/写
上电初始值	1	1	x	x	1	1	x	1

**P2 (A0h) P2 口数据寄存器(读/写)**

位编号	7	6	5	4	3	2	1	0
符号	P2.7	-	-	P2.4	-	-	-	P2.0
读/写	读/写	-	-	读/写	-	-	-	读/写
上电初始值	1	x	x	1	x	x	x	1

**P3(B0h) P3 口数据寄存器(读/写)**

位编号	7	6	5	4	3	2	1	0
符号	-	P3.6	P3.5	-	-	P3.2	P3.1	P3.0
读/写	-	读/写	读/写	-	-	读/写	读/写	读/写
上电初始值	x	1	1	x	x	1	1	1

位编号	位符号	说明
7,6,3,2,0	<b>P1.x</b> (x=0/2/3/6/7)	P1 口锁存寄存器数据
7,4,0	<b>P2.x</b> (x=0/4/7)	P2 口锁存寄存器数据
6,5,2,1,0	<b>P3.x</b> (x=0/1/2/5/6)	P3 口锁存寄存器数据

## 12.3 I/O 端口复用

(略)

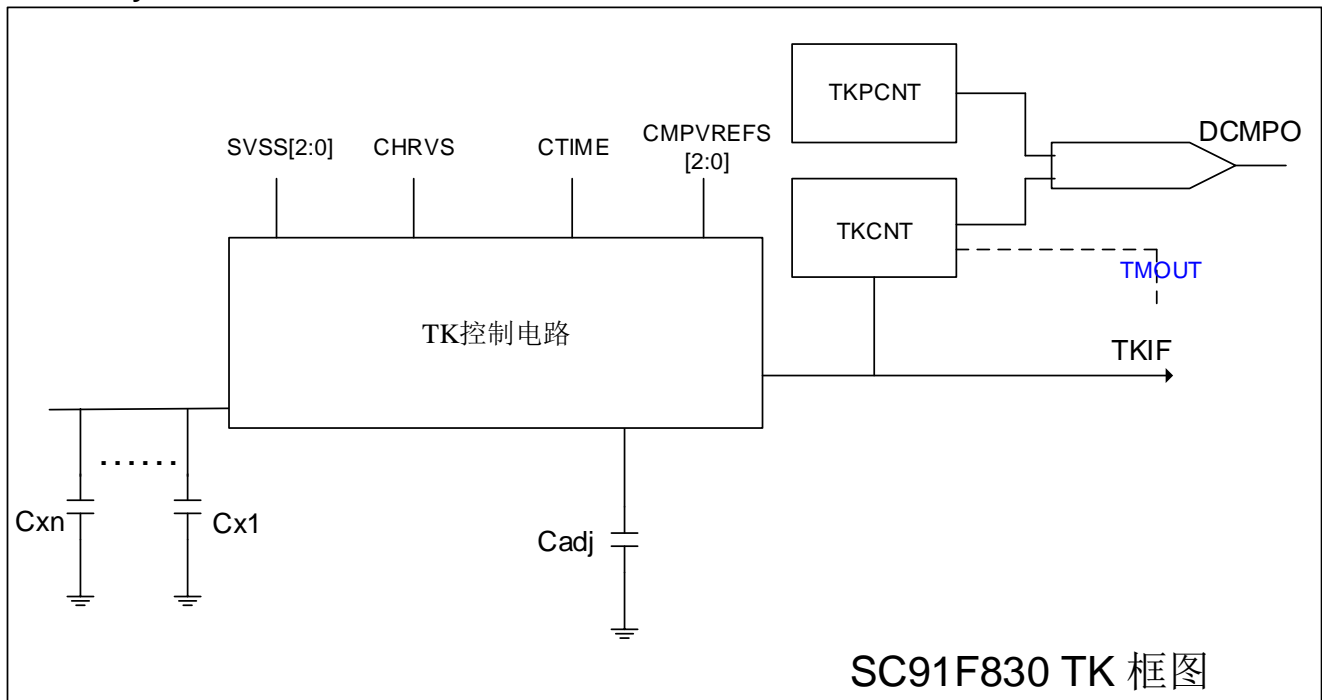
## 13 TOUCH KEY SENSOR

SC91F830 内建一个 10 通道的电容性触摸按键 Touch Key Sensor。

- ① 最多 10 个通道的 Touch Key;
- ② 具有很强的抗干扰能力;
- ③ 具有灵活的灵敏度调节能力;
- ④ 可通过 EMI 测试;

### 13.1 TOUCH KEY 原理

Touch Key 电路架构图:



Touch Key 原理:

- 1, 稳压电源 (或 VDD) 按 CTIME 设置的频率向外部的分布电容 Cxn 和基准电容 Cadj 充放电。
- 2, Cadj 电压会逐步上升至比较器的翻转电压后, 经过滤波电路后给出信号。
- 3, TKCNT 会记录从充电开始到看到信号的时间。
- 4, 如果有手触摸到按键时, Cxn 的值会增加, 则 Cadj 电压上升的时间会减小, 也就是 TKCNT 的值会减小。
- 5, 用户便可以通过 TKCNT 的扫描值来确认是否有触摸按键被按下。

### 13.2 TOUCH KEY 相关寄存器

**TKCR (E8h) Touch Key 控制寄存器 (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	ENTKS	TRIG	-	CHRVS	TKAINS[3:0]			
读/写	读/写	读/写	-	读/写	读/写	读/写	读/写	读/写
上电初始值	0	0	x	0	0	0	0	0

位编号	位符号	说明
7	ENTKS	Touch Key 电路电源开关 0: 关闭 Touch Key 的电源



		1: 打开 Touch Key 的电源
6	<b>TRIG</b>	<b>Touch Key 触发开关</b> 写入“1”完成一次 1 个按键的 Touch Key 扫描（1 个充放电周期）
4	<b>CHRVS</b>	<b>充电电源选择</b> 0: 选择内建的稳压电源作为充电电压源 1: 选择 VDD 作为充电电源
3~0	<b>TKIS[3:0]</b>	<b>触摸按键通道选择</b> 0000: TK0 0001: 保留 0010: 保留 0011: 保留 0100: TK4 0101: 保留 0110: 保留 0111: TK7 1000: TK8 1001: TK9 1010: TK10 1011: 保留 1100: 保留 1101: TK13 1110: TK14 1111: 保留 使用说明: 1, 当 ENTKS=1, 且 TRIG=1 时, TKIS[3:0]被选中的通道才会被设置为触摸按键模式, 当 TK 扫描完成后, 系统会将此 IO 自动设置为 IO 之前的状态; 2, 同一时间只能有一个通道被选择为触摸按键模式; 3, 要想让 TK 正常工作, 必须先 ENTKS=1, 然后再 TRIG=1; 在同一指令内将此两位同时置 1, 因 TK 电源还未打开, TK 将不能正常工作。

**TKCNTH (96h) Touch Key 计数器数值寄存器高 7 位（只读）**

位编号	7	6	5	4	3	2	1	0
符号	-	TKCNT[14:8]						
读/写	-	读	读	读	读	读	读	读
上电初始值	x	0	0	0	0	0	0	0

**TKCNTL (97h) Touch Key 计数器数值寄存器低 8 位（读/写）**

位编号	7	6	5	4	3	2	1	0
符号	TKCNT[7:0]							
读/写	读	读	读	读	读	读	读	读
上电初始值	0	0	0	0	0	0	0	0
位编号	位符号	说明						
6~0	<b>TKCNT[8:14]</b>	<b>Touch Key 计数器数值寄存器高 7 位</b> 记录电荷转移时间长度计数器的高 7 位数值						
7~0	<b>TKCNT[7:0]</b>	<b>Touch Key 计数器数值寄存器低 8 位</b> 记录电荷转移时间长度计数器的低 8 位数值						

**TKCFG1 (9Ah) Touch Key 配置寄存器 1（读/写）**

位编号	7	6	5	4	3	2	1	0
符号	-	CMPFLT[2:0]			TKIOS	-	DISCTS[1:0]	
读/写	-	读/写	读/写	读/写	读/写	-	读/写	读/写
上电初始值	x	0	0	0	0	x	0	0

位编号	位符号	说明
6~4	<b>CMPFLT[2:0]</b>	<b>Cadj 电容电压比较器滤波长度选择</b> 000: 无滤波 001: 滤波长度为 2 个系统时钟 (约 125ns) 010: 滤波长度为 4 个系统时钟 (约 250ns) 011: 滤波长度为 8 个系统时钟 (约 500ns) 100: 滤波长度为 12 个系统时钟 (约 750ns) 101: 滤波长度为 16 个系统时钟 (约 1us) 110: 滤波长度为 24 个系统时钟 (约 1.5us) 111: 滤波长度为 32 个系统时钟 (约 2us) 设置说明: 此部分选项建议跟 <b>CTIME</b> 联动设置, 滤波长度的时钟个数最好为 <b>CTIME[0:3]</b> 的 1.1~1.5 倍, 最长不能超过 2 倍。比如: <b>CTIME=8</b> , 建议设为 12; <b>CTIME=10</b> , 建议设为 16; <b>CTIME=6</b> , 建议设为 8.
3	<b>TKIOS</b>	<b>TK 通道完成 TK 扫描后 Cadj 和 IO 是否连通的设置</b> 1: TK 扫描完成后, CMP 翻转, Cadj 同 IO 切断连接 0: TK 扫描完成后, CMP 翻转, Cadj 同 IO 保持连接 设置说明: 鉴于 IO 状态和电平很复杂, 建议此位通常置 1
1~0	<b>DISCTS</b>	<b>基准电容放电时间长度选择</b> 00: 用 64 个系统时钟(约 4us)长度将 Cadj 放电至 0V 01: 用 128 个系统时钟(约 8us)长度将 Cadj 放电至 0V 10: 用 256 个系统时钟(约 16us)长度将 Cadj 放电至 0V 11: 用 512 个系统时钟(约 32us)长度将 Cadj 放电至 0V

**TKCFG2 (9Bh) Touch Key 配置寄存器 2 (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	FRANDS[1:0]-		-	-	CTIME[3:0]			
读/写	读/写	读/写	-	-	读/写	读/写	读/写	读/写
上电初始值	0	0	x	x	0	0	0	0

位编号	位符号	说明
7~6	<b>FRANDS[1:0]</b>	<b>电荷转移频率跳频设定</b> 00/11: 不做跳频 01: 小幅度跳频 (约±10%范围跳频@2Mhz 充电频率) 10: 大幅度跳频 (约±20%范围跳频@2Mhz 充电频率)
3~0	<b>CTIME[3:0]</b>	<b>Cadj 电容单次充电时长选择</b> 单次从稳压电压源到 Cx 充电和 Cx 到 Cadj 充电的时长设置为: <b>CTIME[3:0]*2*系统时钟周期长度。</b> 比如: 如果设置 <b>CTIME[3:0]</b> 为 8, 则充电长度为 8*2*62.5ns=1000ns; 也就是说: 相当于电荷转移的频率为 8/8 也就是 1Mhz (1/1000ns)。
<b>使用说明: FTRNDS[1:0]为 01 时, CTIME 必须为 0010~1110 之间的任意值;</b> <b>FTRNDS[1:0]为 10 时, CTIME 必须为 0011~1101 之间的任意值;</b>		

**TKCFG3 (9Ch) Touch Key 配置寄存器 3 (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	-	-	SVSS[2:0]			CMPVREFS[2:0]		
读/写	-	-	读/写	读/写	读/写	读/写	读/写	读/写
上电初始值	x	x	0	0	0	0	0	0

位编号	位符号	说明
3~2	SVSS[2:0]	<b>稳压源电压设置选择</b> 000: 充电稳压源电压为 2.00V 001: 充电稳压源电压为 2.25V 010: 充电稳压源电压为 2.50V 011: 充电稳压源电压为 2.75V 100: 充电稳压源电压为 3.00V 101: 充电稳压源电压为 3.33V 110: 充电稳压源电压为 3.67V 111: 充电稳压源电压为 4.00V
1~0	CMPVREFS[2:0]	<b>比较器参考电压选择</b> 000: 设定 VREF 为 1.22V 001: 设定 VREF 为 1.33V 010: 设定 VREF 为 1.39V 011: 设定 VREF 为 1.50V 100: 设定 VREF 为 1.61V 101: 设定 VREF 为 1.72V 110: 设定 VREF 为 1.83V 111: 设定 VREF 为 2.00V

**使用说明: 为保证使用正常, 请保证 SVSS 的电压至少要比 CMPVREF 高 1.5V 以上。**

**TKSTA (9Dh) Touch Key 状态寄存器 (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	TMOUT	-	-	-	DCMPO	ACMPO	SACMPO	TKIF
读/写	读/写	-	-	-	读	读	读	读/写
上电初始值	0	x	x	x	0	0	0	0

位编号	位符号	说明
7	TMOUT	<b>Cadj 电压 CMP 翻转超时</b> 如果 TKCNT[14:0]溢出后, 比较器的输出仍然没有从低翻转到高, 则会停止对 Cadj 的充电, 并将此 TMOUT 标志置为 1, 并将 TKIF 置 1 后进入 TK 中断。用户可在中断程序中查询 TMOUT 的状态来区分 TKIF 的来源。 此时发生异常情况, 请用户修改 TK 部分的参数设置使其回归到正常值, 或者用户硬件系统发生了不可恢复的故障。
3	DCMPO	<b>数字比较器输出状态</b> 0: TKCNT 值小于用户设定值 TKPV 1: TKCNT 值大于用户设定值 TKPV
2	ACMPO	<b>模拟比较器 (Cadj 电压比较器) 输出状态</b> 0: Cadj 电压低于 CMPVREFS 设定值 1: Cadj 电压高于 CMPVREFS 设定值
1	SACMPO	<b>系统时钟采样后的模拟比较器 (Cadj 电压比较器) 输出状态</b>
0	TKIF	<b>Touch Key 中断标志</b> 在每次模拟比较器翻转后, 此标志被置为 1, 同时产生 Touch Key 中

		断。当中断被响应后，硬件会将此标志自动清除掉，也可以由软件清除。
--	--	----------------------------------

**TKPVH (9Eh) 触摸按键对比值寄存器高 7 位 (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	TKPV[14:8]							
读/写	-	读/写	读/写	读/写	读/写	读/写	读/写	读/写
上电初始值	x	0	0	0	0	0	0	0

**TKPVL (9Fh) 触摸按键对比值寄存器低 8 位 (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	TKPV[7:0]							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
上电初始值	0	0	0	0	0	0	0	0

位编号	位符号	说明
6~0	<b>TKPV[8:14]</b>	<b>数字比较器负端预设值寄存器高 7 位</b> 用户可以直接将电荷转移的值和此预设值作比较，而不需要经过运算。 如果 $TKCNT \geq TKPV$ 表示有 TK，同时 DCMPO 输出为 1； 如果 $TKCNT < TKPV$ 则表示没有 TK，同时 DCMPO 输出为 0；
7~0	<b>TKPV[7:0]</b>	<b>数字比较器负端预设值寄存器低 8 位</b> 用户可以直接将电荷转移的值和此预设值作比较，而不需要经过运算。

### 13.3 TOUCH KEY 使用方法

**一次 TK 采样程序流程：**

1. 选择 TK 按键的通道，设置  $TKAINS[3:0]$ ，选择当前的输入通道为 TK 的输入通道（其它的通道仍然为 IO，并且保持之前的状态），将  $TKIOS$  置为 1；
2.  $ENTKS$  置 1，打开 Touch Key 电源；
3. 设置参数： $CMPVREFS$ 、 $SVSS$ 、 $FTRANS$ 、 $CTIMES$ 、 $DISCTS$ 、 $CMPFLTS$ ；打开 TK 中断
4.  $TRIG$  设置为 1（此时  $TKAINS[3:0]$  设置的通道成为 TK 按键的输入通道，其它 TK 仍然为 IO，并保持原始状态），TK 电路开始放电和充电，并采样相关充放电完成时间；
5.  $TKIF$  产生（如果  $TMOUT$  为 0，则  $TKCNT$  的值为真实值；如果  $TMOUT$  为 1，则  $TKCNT$  值不可信，请重新从第三步开始操作）。
6. TK 中断响应，并将 TK 中断关闭，等待下一次设置及扫描。

## 14 软件 UART

SC91F830 内置高速的系统时钟，可以方便的使用少量的硬件资源，通过软件模拟 UART 的操作时序来实现 UART 的收发功能。赛元提供软件 UART 的标准程序模块供用户使用，能以最高 56000/19200 波特率稳定工作在 16/4MHz 系统时钟下。

软件 UART 所需硬件资源：

- 半双工模式（分时收/发）：一路外部中断和一路定时器；
- 全双工模式（同时收/发）：一路外部中断和二路定时器；

用户使用软件 UART 程序模块的操作步骤如下：

1. 配置所选用的 RXD 和 TXD 对应引脚的 I/O 模式：设置 RXD 为纯输入模式，TXD 为强推挽模式，RXD 选择有外部中断的引脚；
2. 根据实际使用的波特率配置定时器模式和重装值，配置 RXD 对应的外部中断为下降沿触发，配置中断为高优先级，打开外部中断，关定时器中断。
3. 软件 UART 接收
  - 1) 当 RXD 引脚外部中断发生时，在中断服务程序中关闭外部中断，打开定时中断；
  - 2) 进入定时中断服务程序，检测 RXD 引脚电平，判断起始位；
  - 3) 接收 RXD 引脚的 8 位数据；
  - 4) 判断停止位；
  - 5) 接收完成，打开外部中断，关闭定时器，准备接收下一个字节；
  - 6) 保存接收到的数据，重复步骤 1-5；
4. 软件 UART 发送：
  - 1) 打开定时器；
  - 2) 进入定时中断服务程序，发送起始位；
  - 3) 发送 8 位数据；
  - 4) 发送停止位，关闭定时器；
  - 5) 准备下一字节数据，重复步骤 1-4；

软件 UART 通信数据格式为：1 位起始位+8 位数据位+1 位停止位

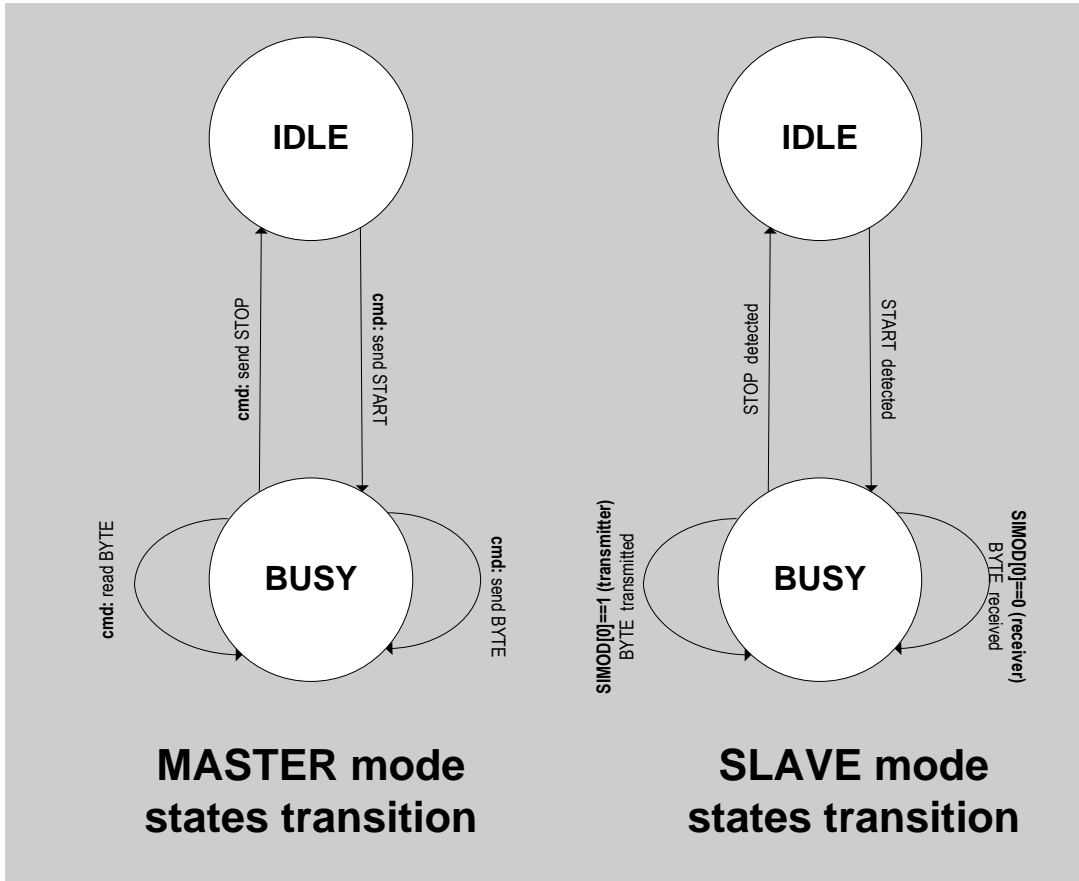
注意：由于软件 UART 的发送是在中断中完成，启动发送后直到发送完成之前不要再次发送数据、不要更改发送缓冲区数据，也不要进行 IAP 操作，以防数据错乱和更改。

赛元提供软件 UART 的程序模块及使用例程，可在[赛元网站](#)下载使用，详细使用方法请参考《赛元软件 UART 应用指南》。

## 15 串行接口（SIF）

SC91F830 内建了一个简单的串口通讯接口 Serial Interface(SIF)，其电气特性类似 I<sup>2</sup>C bus，该 SIF 可以有 MASTER mode 以及 SLAVE mode 的两种选择，但是其 SLAVE mode 无法完全兼容一个具有标准 I<sup>2</sup>C Interface 的器件。

SC91F830 的 SIF 可以有两种操作模式：MASTER mode 和 SLAVE mode。在 MASTER mode 下，其操作主要必须由用户下命令完成，例如：SEND-START, SEND-BYTE, READ-BYTE, 以及 SEND-STOP。而在 SLAVE mode 下，则是采用被动方式，靠硬件的中断机制来收取 START, BYTE, 以及 STOP 信号。



## 15.1 SIF 相关 SFR 寄存器

符号	地址	说明	7	6	5	4	3	2	1	0	Reset 值
SIFCFG	D4H	SIF 配置寄存器	ENSI	INVI			SIMOD[2:0]			ACKO	00xx0000b
SIFCTL	D5H	SIF 控制寄存器	-	-	-	-	-		MCMD[1:0]		xxxxxx00b
SIFTXD	D6H	SIF 发送数据寄存器	SIFTXD[7:0]								00000000b
SIFRXD	D7H	SIF 接收数据寄存器	SIFRXD[7:0]								xxxxxxxxb
SIFSTA	D8H	SIF 状态寄存器	RTNACK	-	-	-	STPIF	TXIF	RXIF	STRIF	0xxx0000b

**SIFCFG (D4h) SIF 配置寄存器**

位编号	7	6	5	4	3	2	1	0
符号	ENSI	INVI	-	-	SIMOD[2:0]			ACKO
读/写	读/写	读/写	-	-	读/写	读/写	读/写	读/写
上电初始值	0	0	x	x	0	0	0	0

位编号	位符号	说明
7	ENSI	SIF 开关控制寄存器 0: P1.6 及 P1.7 当作正常 I/O 使用, 其状态由 P1CFG1 及 P1 寄存器决定, 内建的 SIF 处于 IDLE 状态, 所有 SIF 中断标志被清除为 0, 所有的 中断条件侦测停止。 1: 启动 SIF 的功能, 让 SC91F830 的 SIF 处于工作状态。 必须注意的是: SDA 及 SCL 延续启动前的状态! 所以在设定此位前, 应该要先把 SFR: P1CFG0/1 中的 P16M[1:0] 及 P17M[1:0] 都设成 00b, 并且对 P1.6 及 P1.7 都写 1, 以让 P1.7(SDA) 及 P1.6(SCL) 处于 “weak pull-up” 及 “input” 状态, 符合 “类 I <sup>2</sup> C” 的需求。
6	INVI	INVI (INVerse Input), SIF 电平反向输出 0: SC91F830 SIF 所取用的 SDA 直接取用 P1.7 的输入, SCL 直接取用 P1.6 的输入。 1: SC91F830 SIF 用的 SDA 取用 P1.7 的反向, SIF 用的 SCL 取用 P1.6 的反向。
3	SIMOD [2]	MASTER/SLAVE 模式选择 0: MASTER 模式 1: SLAVE 模式
2	SIMOD [1]	ACK 应答选择 0: 有 ACK 应答位 1: 无 ACK 应答位
1	SIMOD [0]	SLAVE 模式发送/接收状态选择 0: SLAVE 模式接收状态 1: SLAVE 模式发送状态
0	ACKO	<b>ACK 信号位 ACKO:</b> 只要 SIF 处于 8+1 bit 的通讯模式下, 当接收到 8-bit 数据后, 所要回复送到 SDA 的第 9bit。此 bit 只对 SIMOD[1]==0 时有用途。
5,4	保留位	保留位

### SIMODE[2:0]设置说明

**SIF 模式设置 (选择 MASTER 模式或者 SLAVE 模式, 选择有无 ACK 应答位)**

SIMODE[2:0]			Mode
主从模式选择位 (MASTER/SLAVE)	应答位选择 (ACK)	发送/接收选择位 (RX/TX)	
0	0	x	MASTER 模式有 ACK 应答位
0	1	x	MASTER 模式无 ACK 应答位
1	0	0	SLAVE 模式有 ACK 应答接收状态
1	0	1	SLAVE 模式有 ACK 应答发送状态
1	1	0	SLAVE 模式无 ACK 接收状态
1	1	1	SLAVE 模式无 ACK 发送状态

00x: SIF 当做 MASTER 使用, 并且采用 8+1 bits(ACK/NACK) 通讯。于发出 START 信号后, 发出 STOP 信号前, 强占住 SCL 的控制权 (强推挽)。

时序图详见 “SIFCTL (D5h) SIF 控制寄存器” 的说明

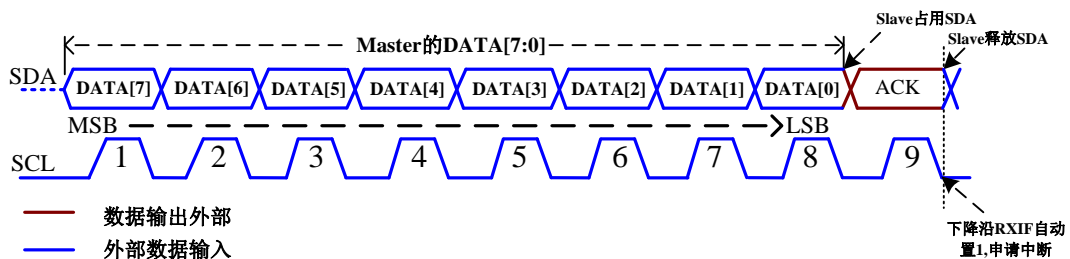
01x: SIF 当做 MASTER 使用, 并且采用 8 bits 通讯(无 ACK/NACK)。于发出 START 信号后, 发出 STOP 信号前, 强占住 SCL 的控制权 (强推挽)

时序图详见 “SIFCTL (D5h) SIF 控制寄存器” 的说明

100: → 此 SIF 当做 SLAVE 使用

→ 采用 8+1 bits(ACKO) 通讯, 随时被动的侦测来自主机的 START 信号和 STOP 信号

→ 此时 SIF 只能接收数据 (pure “receiver”), 且每收满一个 BYTE 就会置位中断标志 “RXIF”。

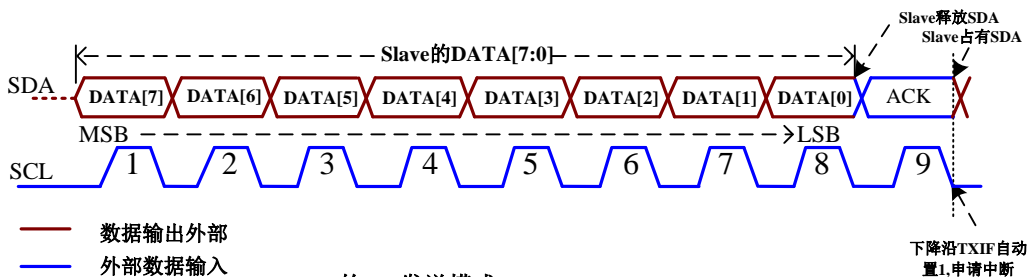


SLAVE 的 8+1 接收模式: SIMODE[2:0]=100

101: → 此 SIF 当做 SLAVE 使用

→ 采用 8+1 bits(set ACKI) 通讯, SLAVE 接收到 START 信号后, 强占 SDA 线, 直到用户将 SLAVE 切换为 SLAVE 模式接收状态 (pure “receiver”) 前, 都不能侦测到主机发出的 STOP 信号。

→ 此时 SIF 只能传送数据 (pure “transmitter”), 且每送出一个 BYTE 就会置位中断标志 “TXIF”。



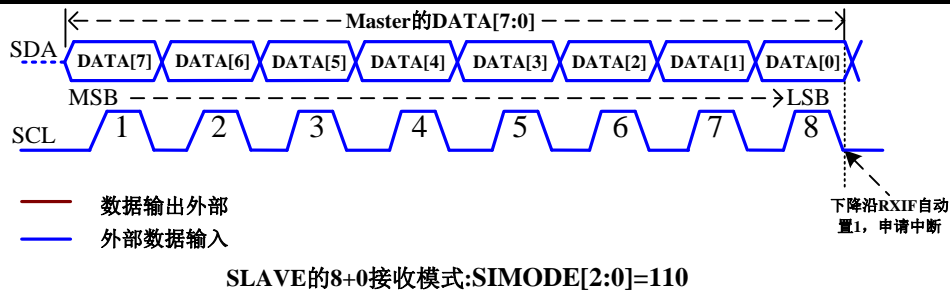
SLAVE 的 8+1 发送模式: SIMODE[2:0]=101

110: → 此 SIF 当做 SLAVE 使用

→ 采用 8 bits (无 ACKO) 通讯, 随时被动的侦测来自主机的 START 信号和 STOP 信号

→ 此时 SIF 只能接收数据 (pure “receiver”), 且每收满一个 BYTE 就会置位中断标志 “RXIF”。

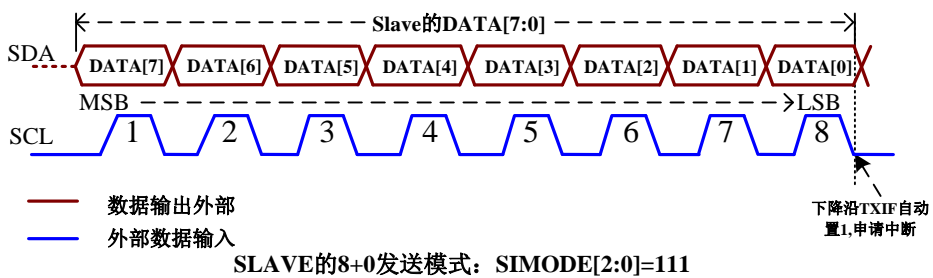




111: →此 SIF 当做 SLAVE 使用

→采用 8bits(无 ACKI) 通讯, SLAVE 接收到 START 信号后, 强占 SDA 线, 直到用户将 SLAVE 切换为 SLAVE 模式接收状态(pure “receiver”)前, 都不能侦测到主机发出的 STOP 信号。

→此时 SIF 只能传送数据 (pure “transmitter”), 且每送出一个 BYTE 就会置位中断标志 “TXIF”。



### SIFCTL (D5h) SIF 控制寄存器 (MASTER 模式)

位编号	7	6	5	4	3	2	1	0
符号	-	-	-	-	-	-	MCMD[1:0]	
读/写	-	-	-	-	-	-	读/写	读/写
上电初始值	x	x	x	x	x	x	0	0

### MCMD[1:0]说明

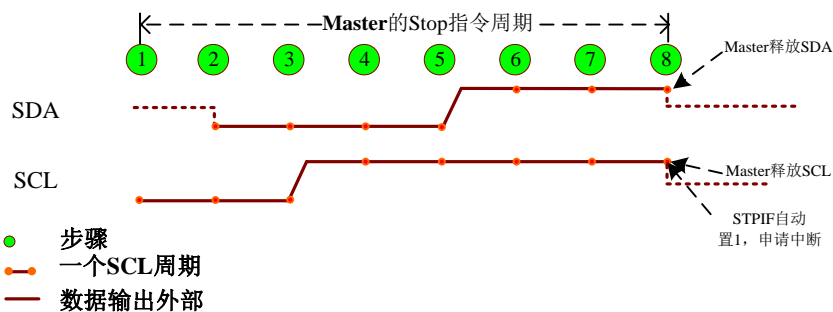
#### MCMD[1:0] SIF 命令寄存器

当 SIF 处于 MASTER mode (SMOD[2]==0)时, 对此寄存器写入一定的值, 对应要 SIF 发出一定的动作。此寄存器存的值并不重要, 不会触发对 SIF bus 的动作, 重要的是对它 “写入的动作” 及数值才会触动对 SIF bus 的动作。

00 : send STOP event

对 MCMD[1:0] 写入 00, 代表要 SC91F830 送出一个 STOP Frame。此命令只对 SC91F830 处于 Master Mode 并且处于 “BUSY” state 有用。

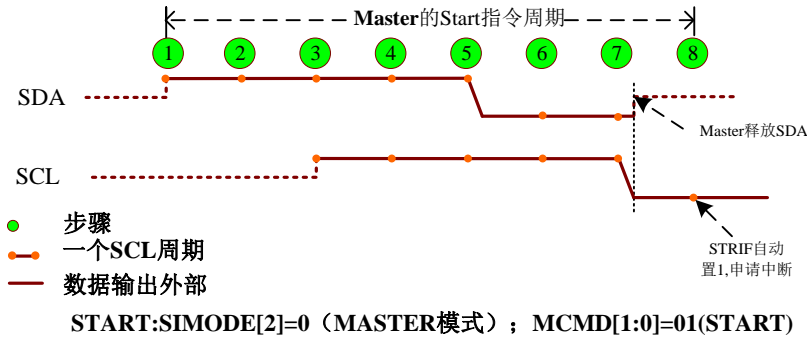
在发出 STOP event 后, SC91F830 会释放 SCL 以及 SDA 的控制权, 让 SIF 回复到 “IDLE” state, 同时也会设定 interrupt flag “STPIF” 为 “1”。



**01: Send START event**

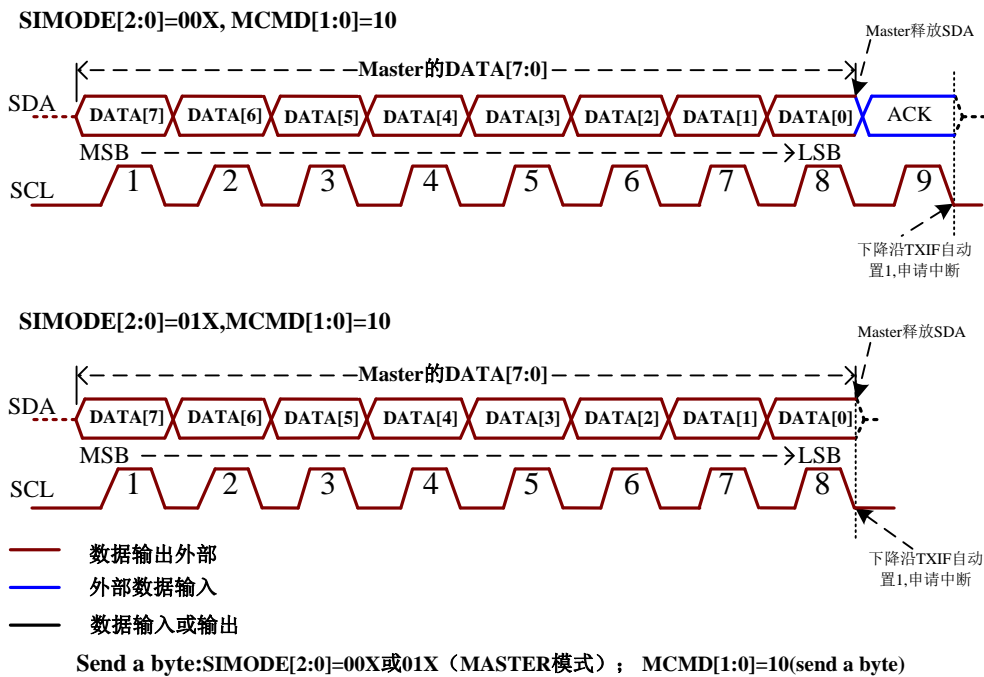
对 MCMD[1:0] 写入 01, 代表要 SC91F830 送出一个 START event。此命令只对 SC91F830 处于 MASTER Mode 并且处于 “IDLE” state。

在发出 START event 后, SC91F830 会强占着 SCL(with driving cap), 但是会释放 SDA(no driving cap), 同时也会设定 interrupt flag “STRIF” 为 “1”。


**10: Send a byte**

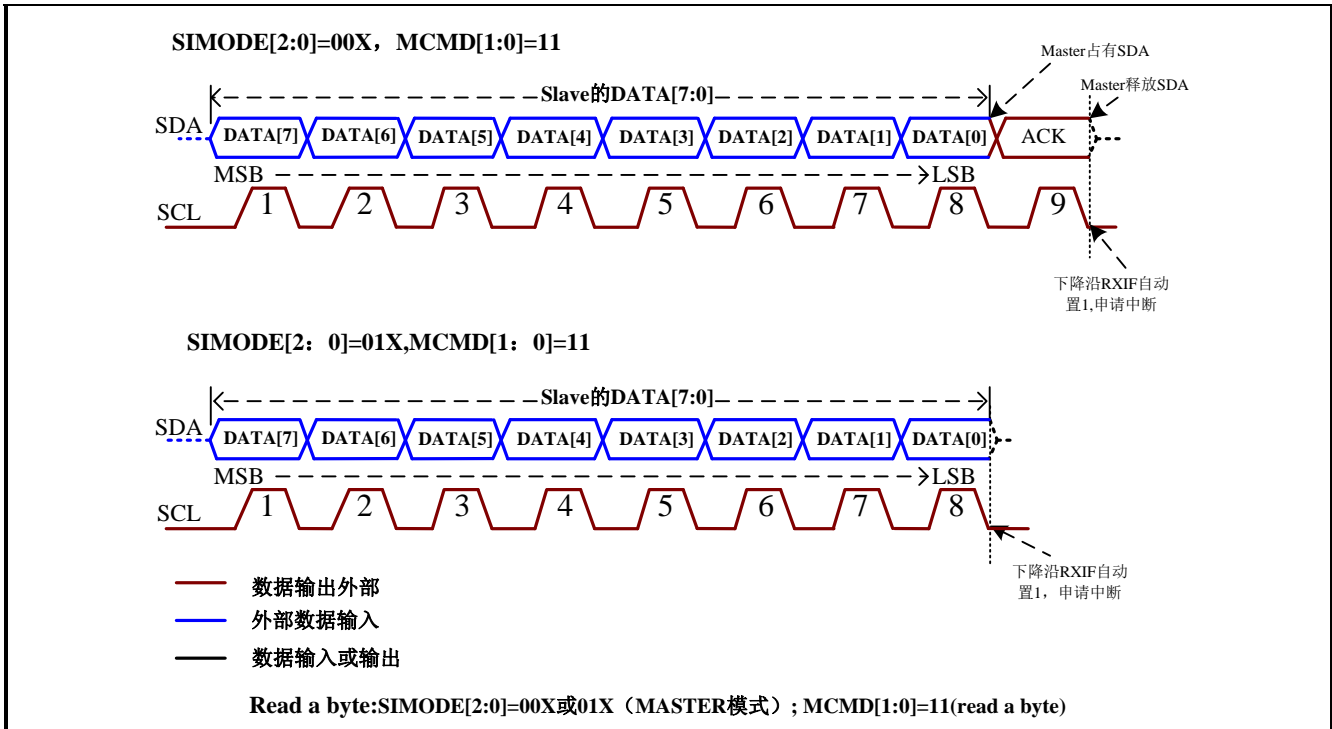
对 MCMD[1:0] 写入 10, 会让 SC91F830 从 SIF 发出一个 BYTE。依照 SIMOD[1] 的状况, 可能会再接收一个 ACK bit 或是就此结束。

在命令结束后, SC91F830 还是会强占 SCL, 但是会释放 SDA, , 同时也会设定 interrupt flag “TXIF” 为 “1”。


**11: Read a byte**

对 MCMD[1:0] 写入 11, 就会让 SC91F830 从 SCL 发出 8 个 clocks, 并且从 SDA 读回一个 BYTE。依照 SIMOD[1] 的状况, 可能会回复一个 ACK bit 到 SDA 或是就此结束。

在命令结束后, SC91F830 还是会强占 SCL, 但是会释放 SDA, 同时也会设定 interrupt flag “RXIF” 为 “1”。


**SIFSTA (D8h) SIF 状态寄存器**

位编号	7	6	5	4	3	2	1	0
符号	RTNACK	-	-	-	STPIF	TXIF	RXIF	STRIF
读/写	读/写	-	-	-	读/写	读/写	读/写	读/写
上电初始值	0	x	x	x	0	0	0	0

位编号	位符号	说明
7	<b>RTNACK</b>	<b>ACK 返回信号位 RTNACK</b> 不管 SC91F830 处于 MASTER 或是 SLAVE mode, 只要 SIMOD[1]==0, 在对 SIF 送出一个 BYTE 之后, 总会再从 SIF 收到一个 ACK bit; 此 Returned ACK bit, 就会被放在此寄存器。
3	<b>STPIF</b>	<b>SIF STOP 信号中断请求标志 STPIF</b> 如果 SC91F830 处于 MASTER mode (SIMOD[2]==0), 在发完 Send STOP 命令后, 此 bit 会被硬件设定成 1; 如果 SC91F830 处于 SLAVE mode, 在检测到 I <sup>2</sup> C Bus 上有 STOP event 时, 此 bit 也会被设定成 1; 此 bit 是一个 Interrupt flag, 可对 CPU 提出中断请求; 此 bit 必须由 User 自行以 CPU instruction 清除。
2	<b>TXIF</b>	<b>SIF 发送完成中断标志 TXIF</b> 在完成 Send a Byte 命令后, SC91F830 会于 SCL 的 neg-edge 设定此 bit 为 1. 此 bit 是一个 Interrupt flag, 可对 CPU 提出中断请求; 此 bit 必须由 User 自行以 CPU instruction 清除。
1	<b>RXIF</b>	<b>SIF 接收完成中断标志 RXIF</b> 在完成 Read a Byte 命令后, SC91F830 会于 SCL 的 neg-edge 设定此 bit 为 1. 此 bit 是一个 Interrupt flag, 可对 CPU 提出中断请求; 此 bit 必须由 User 自行以 CPU instruction 清除。
0	<b>STRIF</b>	<b>SIF START 信号中断标志 STRIF</b> 如果 SC91F830 处于 MASTER mode (SIMOD[2]==0), 在发完 Send START 命令后, 此 bit 会被硬件设定成 1; 如果 SC91F830 处于 SLAVE mode, 在检测到 I <sup>2</sup> C Bus 上有 START event 时, 此 bit 也会被设定成 1; 此 bit 是一个 Interrupt flag, 可对 CPU 提出中断请求; 此 bit 必须由 User 自行以 CPU instruction 清除。

6,5,4	保留位	保留位
-------	-----	-----

**SIFTXD (D6h) SIF 发送数据寄存器**

位编号	7	6	5	4	3	2	1	0
符号	SIFTXD[7:0]							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
上电初始值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7~0	SIFTXD	SIF 数据位 (发送)

**SIFRXD (D7h) SIF 接收数据寄存器**

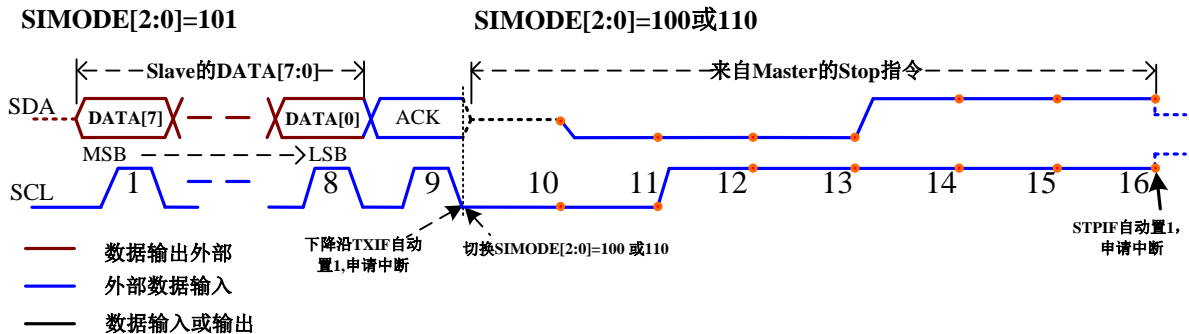
位编号	7	6	5	4	3	2	1	0
符号	SIFRXD[7:0]							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
上电初始值	x	x	x	x	x	x	x	x

位编号	位符号	说明
7~0	SIFRXD	SIF 数据位 (接收)

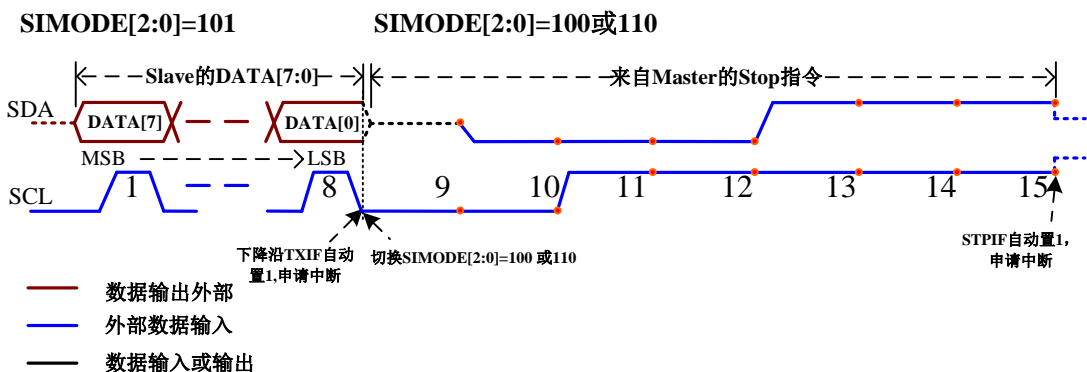
## 15.2 SIF 应用注意事项

### 1. SLAVE 模式发送状态→SLAVE 模式接收 STOP

SLAVE 模式发送状态下, SLAVE 一直占有 SDA 总线, 根据 MASTER 的 SCL 信号发送数据, 无法接收 MASTER 发来的 STOP 信号; 当用户需要接收 MASTER 的 STOP 信号时, 需在 SLAVE 的 1 BYTE 数据发送完成时, 将 SLAVE 设置为接收模式(SLAVE 释放 SDA 总线)。



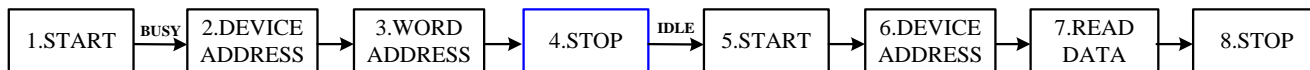
SLAVE模式发送状态转接收状态 (with ACK)



SLAVE模式发送状态转接收状态 (without ACK)

## 2. MASTER 模式与标准 I<sup>2</sup>C 设备进行随机读取操作

SIF 在 MASTER 模式与标准 I<sup>2</sup>C 设备进行随机读取操作时，用户需在 SIF 发送第 2 个 START 信号前，发送一个 STOP 信号，使 SIF 进入 IDLE 模式，才能发送 START 信号，具体流程如下：



# 16 模数转换 ADC

SC91F830 内建一个 10-bit 4 通道的高精度逐次逼近型 ADC，此 ADC 和 IO 口的其它功能复用。

ADC 的参考电压可以有 2 种选择：

- ①是 VDD 管脚（即直接是内部的 VDD）；
- ②是内部 Regulator 输出的参考电压精准的 2.4V；

## 16.1 ADC 相关寄存器

**ADCCFG (C4h)ADC 参考电压配置寄存器（读/写）**

位编号	7	6	5	4	3	2	1	0
符号	P17AIN7	P16AIN6	-	-	P13AIN3	P12AIN2	-	VREFS
读/写	读/写	读/写	-	-	读/写	读/写	-	读/写
上电初始值	0	0	x	x	0	0	x	n

位编号	位符号	说明
7,6,3,2	<b>P1xAINx</b> (x=2/3/6/7)	<b>P1 口 ADC 设置寄存器</b> 0: 设定 P1xAINx 为 IO 口 1: 设定 P1xAINx 为 ADC 输入，并自动将上拉电阻移除。
0	<b>VREFS</b>	<b>参考电压选择(初始值从 Code Option 调入，用户可修改设置)</b> 0: 设定 VREF 为 VDD 1: 设定 VREF 为内部准确的 2.4V

**ADCCR (C5h)ADC 转换控制寄存器**

位编号	7	6	5	4	3	2	1	0
符号	ADCEN	ADCS	ADCKS	EOC/ ADCIF	-	ADCIS[2:0]		
读/写	读/写	读/写	读/写	读/写	-	读/写	读/写	读/写
上电初始值	0	0	0	0	x	0	0	0

位编号	位符号	说明
7	<b>ADCEN</b>	启动 ADC 的电源 <b>0:</b> 关闭 ADC 模块电源 <b>1:</b> 开启 ADC 模块电源
6	<b>ADCS</b>	<b>ADC 开始触发控制 (ADC Start)</b> 对此 bit 写“1”，开始做一次 ADC 的转换，即该位只是 ADC 转换的触发信号。此位只可写入 1 有效。
5	<b>ADCKS</b>	<b>ADC 时钟频率选择(ADC Clocks Selector)</b> 0: 设定 ADC 所使用的 clock 频率为 Fosc 1: 设定 ADC 所使用的 clock 频率为 Fosc/6 ADC 转化需要 89 个 ADC CLOCK 完成
4	<b>EOC /ADCIF</b>	<b>转换完成/ADC 中断请求标志(End Of Conversion / ADC Interrupt Flag)</b> 0: 转换尚未完成 1: ADC 转换完成。需用户软件清除

		ADC 转换完成标志 EOC: 当使用者设定 ADCS 开始转换后, 此位会被硬件自动清除为 0; 当转换完成后, 此位会被硬件自动置为 1; ADC 中断请求标志 ADCIF: 此位同时也当作是 ADC 中断的中断请求标志, 如果用户使能 ADC 中断, 那么在 ADC 的中断发生后, 用户必须用软件清除此位。
2~0	<b>ADCIS[2:0]</b>	<b>ADC 输入通道选择(ADC Input Selector)</b> 000: 保留 001: 保留 010: 选用 P1.2 当作 ADC 的输入 011: 选用 P1.3 当作 ADC 的输入 100: 保留 101: 保留 110: 选用 P1.6 当作 ADC 的输入 111: 选用 P1.7 当作 ADC 的输入
3	保留位	保留位

**ADC VH (C6h) ADC 转换数值寄存器 (高 8 位) (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	ADCV[9:2]							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
上电初始值	1	0	0	0	0	0	0	0

**ADC VL (C7h) ADC 转换数值寄存器 (低 2 位) (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	-	-	-	-	-	-	ADCV[1:0]	
读/写	-	-	-	-	-	-	读/写	读/写
上电初始值	x	x	x	x	x	x	0	0

位编号	位符号	说明
7~0	<b>ADCV[9:2]</b>	ADC 转换值的高 8 位数值
1~0	<b>ADCV[1:0]</b>	ADC 转换值的低 2 位数值

**IE (A8h) 中断使能寄存器 (读/写)**

位编号	7	6	5	4	3	2	1	0
符号	EA	EADC	EPWM	EPSIF	ET1	-	ET0	ETK
读/写	读/写	读/写	读/写	读/写	读/写	-	读/写	读/写
上电初始值	0	0	0	0	0	x	0	0

位编号	位符号	说明
6	<b>EADC</b>	ADC 中断使能控制 0: 不允许 EOC/ADCIF 产生中断 1: 允许 EOC/ADCIF 产生 中断

**IP (B8h) 中断优先级寄存器(读/写)**

位编号	7	6	5	4	3	2	1	0
符号	-	IPADC	IPPWM	IPSIF	IPT1	-	IPT0	IPTK
读/写	-	读/写	读/写	读/写	读/写	-	读/写	读/写
上电初始值	x	0	0	0	0	x	0	0

位编号	位符号	说明
-----	-----	----

6	IPADC	ADC 中断优先权选择 0: 设定 ADC 的中断优先级是“低” 1: 设定 ADC 的中断优先级是“高”
---	-------	---

## 16.2 ADC 转换步骤

用户实际进行 ADC 转换所需要的操作步骤如下:

- ① 设定 ADC 输入管脚; (设定 P1xAInx 对应的位为 ADC 输入, 通常 ADC 管脚会预先固定)
- ② 设定 ADC 参考电压 Vref, 设定 ADC 转换所用的频率
- ③ 开启 ADC 模块电源;
- ④ 选择 ADC 输入通道; (设置 ADCIS 位, 选择 ADC 输入通道)
- ⑤ 启动 ADCS, 转换开始
- ⑥ 等待 EOC/ADCIF=1, 如果 ADC 中断使能, 则 ADC 中断会产生, 用户需要软件清 0 EOC/ADCIF 标志
- ⑦ 从 ADCVH、ADCVL 获得 10 位数据, 先高位后低位, 一次转换完成
- ⑧ 如不转换输入通道, 则重 5~7 的步骤, 进行下一次转换

注意事项: 在设定 IE[6](EADC)前, 使用者最好用软件先清除 EOC/ADCIF, 并且在 ADC 中断服务程序执行完时, 也清除该 EOC/ADCIF, 以避免不断的产生 ADC 中断。

## 17 IAP 操作

SC91F830 内部有 256B Flash 可以进行 In Application Programming (IAP) 操作, 即允许用户程序动态的把数据写入内部的 Flash, 即作为 EEPROM 使用。

用户使用 IAP 时, 只能把数据写入内部 8K Flash ROM 的最后 256 Bytes (1F00H ~ 1FFFH)。

### 17.1 IAP 操作相关寄存器

IAP 相关 SFR 寄存器说明:

符号	地址	说明	7	6	5	4	3	2	1	0	Reset 值
IAPKEY	EAH	IAP 保护寄存器	IAPKEY[7:0]								00000000b
IAPADL	ECH	IAP 地址低位	IAPADR[7:0]								11111111b
IAPDAT	EDH	IAP 写入/读出资料	IAPDAT[7:0]								11111111b
IAPCTL	EEH	IAP 命令	-	-	-	-	PAYTIMES [1:0]		CMD[1:0]		xxxx0000b

IAPKEY (EAH) IAP 保护寄存器 (读/写)

位编号	7	6	5	4	3	2	1	0
符号	IAPKEY[7:0]							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
上电初始值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7~0	IAPKEY[7:0]	打开 IAP 功能及 IAP 操作时限设置 写入一个非零值 n, 则代表意思为: ①打开 IAP 功能; ②n 个系统时钟后如果接收不到 IAP 写入命令, 则 IAP 功能被重新关闭;

IAPADL (ECH) IAP 写入地址低 8 位寄存器

位编号	7	6	5	4	3	2	1	0
符号	IAPADR[7:0]							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
上电初始值	1	1	1	1	1	1	1	1

位编号	位符号	说明
7~0	IAPADR[7:0]	IAP 写入地址的低 8 位

IAPDAT (EDH) IAP 数据寄存器

位编号	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---

符号	IAPDAT[7:0]							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
上电初始值	1	1	1	1	1	1	1	1

位编号	位符号	说明
7~0	<b>IAPDAT</b>	IAP 写入的数据

**IAPCTL (EEH) IAP 控制寄存器**

位编号	7	6	5	4	3	2	1	0
符号	-	-	-	-	PAYTIMES[1:0]		CMD[1:0]	
读/写	-	-	-	-	读/写	读/写	读/写	读/写
上电初始值	x	x	x	x	0	0	0	0

位编号	位符号	说明
3~2	<b>PAYTIMES[1:0]</b>	IAP 写入操作时, CPU Hold Time 时间长度设定 00: 设定 CPU HOLD TIME 4mS@16Mhz/4Mhz 01: 设定 CPU HOLD TIME 2mS@16Mhz/4Mhz 10: 设定 CPU HOLD TIME 1mS@16Mhz/4Mhz 11: 保留 说明: CPU Hold 的是 PC 指针, 其他功能模块继续工作; 中断标志会被保存, 并在 Hold 结束后进入中断, 但多次的中断只能保留最后一次。 选择建议: VDD 在 2.7V~5.5V, 可选择 10 VDD 在 2.4~5.5V, 可选择 01 或者 00
1~0	<b>CMD[1:0]</b>	IAP 写入操作命令 00 := (保留) 01 := (保留) 10 := 写入 11 := (保留)

## 17.2 IAP 操作流程

SC91F830 的 IAP 写入流程如下:

- ① 写入 IAPDAT[7:0] (准备好 IAP 写入的数据);
- ② 写入 {IAPADR[11:8], IAPADR[7:0]} (准备好 IAP 操作的目标地址, IAPADR[11:8]固定为 1);
- ③ 写入 IAPKEY[7:0] 写入一个非 0 的值  $n$  (打开 IAP 保护, 且在  $n$  个系统时钟内没收到写入命令 IAP 会被关闭);
- ④ 写入 IAPCTL[3:0] (设定 CPU Hold 时间, 写入 CMD[1:0]为 1、0, CPU Hold 并启动 IAP 写入);
- ⑤ IAP 写入结束, CPU 继续后续操作;

注意事项: 利用 MOV<sub>C</sub> 指令, 用户可以读取此部分 IAP 写入的数据。

## 17.3 IAP 读写范例程序

```
#include "intrins.h"
unsigned char code *POINT=0x1F00;
unsigned char DATA1,ADDR1;
```

**IAP 写操作 C 的 Demo 程序:**

```
IAPDAT=DATA1;           //送数据 DATA1 到 IAP 数据寄存器
IAPADL=ADDR1;           //写入地址值 ADDR1
IAPKEY=0xf0;            //此值可根据实际调整; 需保证本条指令执行后到对 IAPCTL 赋值前,
                        //时间间隔需小于 240 (0xf0) 个系统时钟, 否则 IAP 功能关闭;
                        //开启中断时要特别注意
```



```
IAPCTL=0x0a;           //执行 IAP 写入操作, 1ms@16M/4M;  
_nop_();                //等待(至少需要 1 个_nop_())  
_nop_();  
_nop_();  
_nop_();
```

**IAP 读操作 C 的 Demo 程序:**

```
DATA1=*(POINT+ADDR1); //读取 ADDR1 的值到 DATA1
```

**IAP 读操作汇编的 Demo 程序:**

```
MOV DPTR, #1F00H;      //DPTR 赋初值  
MOV A, ADDR1;          //地址值送 A  
MOVC A, @A+DPTR;      //读取 ADDR1 的值到 A
```

## 18 电气特性

### 18.1 极限参数

符号	参数	最小值	最大值	UNIT
VDD/VSS	直流供电电压	-0.3	5.5	V
Voltage ON any Pin	任一脚输入/输出电压	-0.3	VDD+0.3	V
TA	工作环境温度	-40	85	°C
TSTG	储存温度	-55	125	°C

### 18.2 推荐工作条件

符号	参数	最小值	最大值	UNIT
VDD	工作电压	2.4	5.5	V
TA	工作环境温度	-40	85	°C

### 18.3 直流电气特性

(VDD = 5V, TA = +25°C, 除非另有说明)

符号	参数	最小值	典型值	最大值	单位	测试条件
<b>电流</b>						
Iop1	工作电流		8.0	10	mA	IRC=16MHz
Iop2	工作电流		3.5	5.0	mA	IRC=4MHz
Ipd	待机电流 (Power Down 模式)	-	0.1	1.0	uA	IRC=16M/4M Hz
Itk1	Toukey 工作电流		0.5	1	mA	IRC=16MHz
<b>IO 口特性</b>						
V <sub>IH</sub>	输入高电压	0.7VDD	-	VDD+0.5	V	
V <sub>IL</sub>	输入低电压	-0.5	-	0.3VDD	V	
V <sub>IH,RSTN</sub>	输入高电压, RSTN 脚	2.0		VDD	V	
V <sub>IL,RSTN</sub>	输入低电压, RSTN 脚	-0.2		1.5	V	
IOL1	灌电流 P3		12		mA	V <sub>Pin</sub> =0.4V
IOL2	灌电流 P3		23		mA	V <sub>Pin</sub> =0.8V
IOL3	灌电流 P3		34		mA	V <sub>Pin</sub> =1.5V
IOL4	灌电流 P1/P2		25		mA	V <sub>Pin</sub> =0.4V
IOL5	灌电流 P1/P2		45		mA	V <sub>Pin</sub> =0.8V
IOL6	灌电流 P1/P2		67		mA	V <sub>Pin</sub> =1.5V
IOH1	输出高的电流 (准双向口模式) P3		40		uA	V <sub>pin</sub> =4.7V
IOH2	输出高的电流 (强推挽模式) P1/P2/P3		16		mA	V <sub>pin</sub> =4.3V
IOH3	输出高的电流 (强推挽模式) P1/P2/P3		7		mA	V <sub>pin</sub> =4.7V
IOH4	输出高的电流 (准双向口模式) P1/P2		60		uA	V <sub>pin</sub> =4.7V
做为 ADC 参考电压的内部基准 2.4V						
VDD24	内部基准 2.4V 电压输出	2.37	2.40	2.45	V	TA=-40~85°C

(VDD = 3.3V, TA = +25°C, 除非另有说明)

符号	参数	最小值	典型值	最大值	单位	测试条件
<b>电流</b>						

Iop3	工作电流		7.8	10	mA	IRC=16MHz
Iop4	工作电流		3.5	5.0	mA	IRC=4MHz
Ipd2	待机电流 (Power Down 模式)	-	0.1	1	uA	IRC=16M/4M Hz
Itk2	Toukey 工作电流		0.4	1	mA	IRC=16MHz
IO 口特性						
V <sub>IH</sub>	输入高电压	0.7VDD	-	VDD+0.5	V	
V <sub>IL</sub>	输入低电压	-0.5	-	0.3VDD	V	
V <sub>IH,RSTN</sub>	输入高电压, RSTN 脚	2.0		VDD	V	
V <sub>IL,RSTN</sub>	输入低电压, RSTN 脚	-0.2		1.5	V	
IOL7	灌电流 P3		9		mA	V <sub>Pin</sub> =0.4V
IOL8	灌电流 P3		15		mA	V <sub>Pin</sub> =0.8V
IOL9	灌电流 P1/P2		18		mA	V <sub>Pin</sub> =0.4V
IOL10	灌电流 P1/P2		31		mA	V <sub>Pin</sub> =0.8V
IOH5	输出高的电流 (准双向口模式) P3		26		uA	V <sub>pin</sub> =3.0V
IOH6	输出高的电流 (强推挽模式) P1/P2/P3		5		mA	V <sub>pin</sub> =3.0V
IOH7	输出高的电流 (准双向口模式) P1/P2		40		uA	V <sub>pin</sub> =3.0V
做为 ADC 参考电压的内部基准 2.4V						
VDD24	内部基准 2.4V 电压输出	2.37	2.40	2.45	V	T <sub>A</sub> =-40~85°C

#### 18.4 交流电气特性 (VDD = 2.5V ~ 5.5V, TA = 25°C, 除非另有说明)

符号	参数	最小值	典型值	最大值	单位	测试条件
Tosc	振荡器起振时间		5	20	us	IRC=16MHz VDD=5V
Treset	复位脉冲宽度	18			us	低电平有效
FIRC	RC 振荡稳定性	15.84	16	16.16	MHz	VDD=4.5~5.5 V T <sub>A</sub> =-20~85 °C

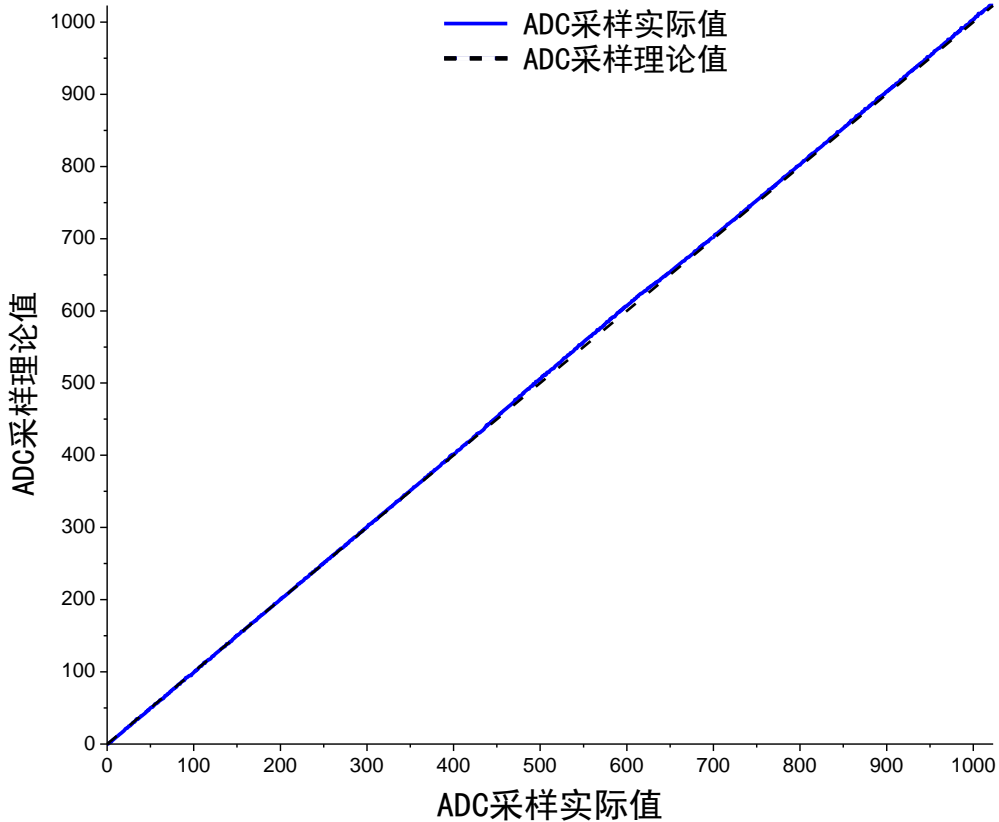
#### 18.5 ADC 电气特性 (TA = 25°C, 除非另有说明)

符号	参数	最小值	典型值	最大值	单位	测试条件
VAD	供电电压	2.4	5.0	5.5V	V	
NR	精度		10		bit	GND ≤ VAIN ≤ VREF
VAIN	ADC 输入电压	GND		VDD	V	
RAIN	ADC 输入电阻	5			MΩ	V <sub>IN</sub> =5V
Rref	Vref 输入阻抗		13.5		KΩ	
ZAIN	模拟电压源推荐阻抗			10	KΩ	
IADC1	ADC 转换电流 1		1.0		mA	ADC 模块打开 VDD=5V
IADC2	ADC 转换电流 2		0.5		mA	ADC 模块打开 VDD=3.3V
DNL	微分非线性误差		±1	±1.5	LSB	VDD=5V
INL	积分非线性误差		±3	±5	LSB	VDD=5V
EAD	总绝对误差		±3	±6	LSB	VDD=5V
TADC	ADC 转换时间					90 个 ADC CLKs

说明：ADC 曲线的线性度及一致性好，其数据的偏差值如下图发生在 2.5V（1/2Vref）的-0.4V~+0.7V（2.1V~3.2V 区间，方向及误差值稳定在+3~-+8LSB，用户可在程式中对 ADC 数值进行修正，修正后的 ADC 精度能在±2LSB 以内，即实际净精度在 9 位以上。

## 18.6 ADC 实测曲线图

ADC 实测曲线图如下：（测试条件为：5V, ADC CLK 频率选择为 Fosc/6, ADC 输入通道对地接 100pF 电容）

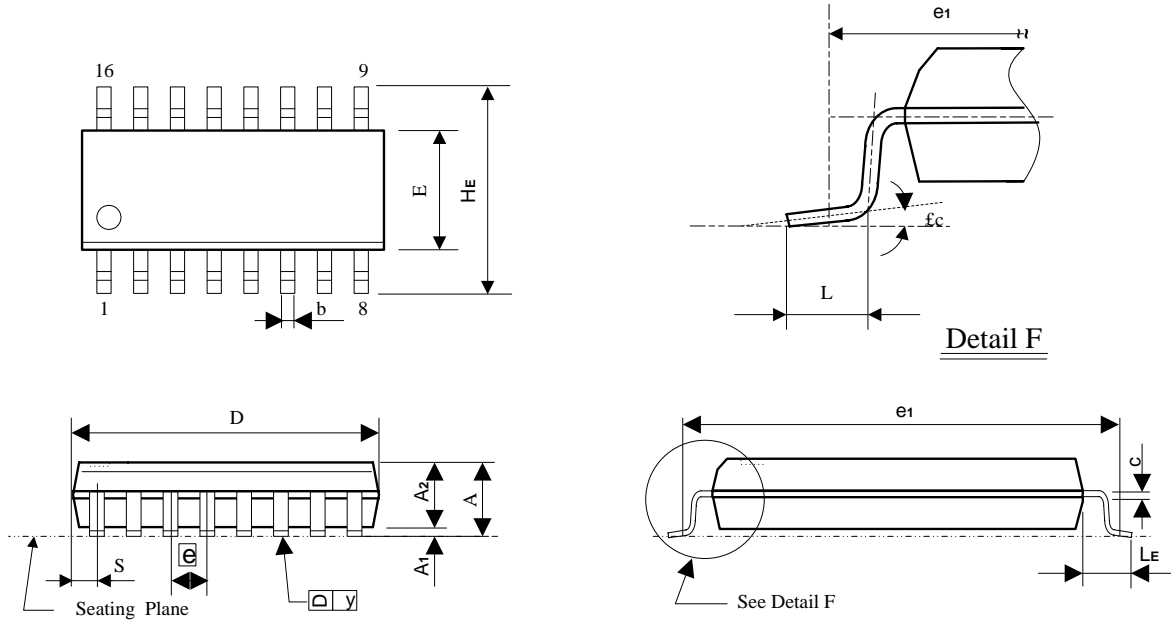


## 19 订购信息

产品编号	封装	包装
SC91F830M16U	SOP16L	管装

## 20 封装信息

SOP 16L(150mil) 外形尺寸 单位:毫米



符号	mm(毫米)		
	最小	正常	最大
A	1.500	1.600	1.700
A1	0.100	0.150	0.200
A2	1.400	1.450	1.500
b	0.356	0.406	0.456
C	-	0.203	-
D	9.700	9.900	10.100
E	3.800	3.900	4.000
HE	5.900	6.000	6.100
$\text{⓪}$	1.27(BSC)		
L	0.600	0.660	0.700
LE	0.950	1.050	1.150
$\theta$	0°	-	8°

## 21 规格更改记录

版本	记录	日期
V1.0	初版	2012年8月