目录

1.总体描述

SC91F731 是一颗加强型的超快速 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。SC91F731 内部集成有 8KB Flash ROM（包含 256Byte EEPROM）、256B SRAM、最多 18GP I/O（包含 8 个大电流 LED 驱动口）、5 路外部中断、2 个 16 位定时器、8 路 10 位高精度 ADC、1 路 8 位可分时输出至 2 路的 PWM、1 路 12 位 Buzzer、1 个串口通讯接口（SIF）、1 个 32K Base timer、内部高精度 16M/8M Hz 振荡器等资源。为提高可靠性及简化客户电路，SC91F731 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。SC91F731 可广泛应用于各种大小家电、工业控制等应用领域。

2. 主要特色

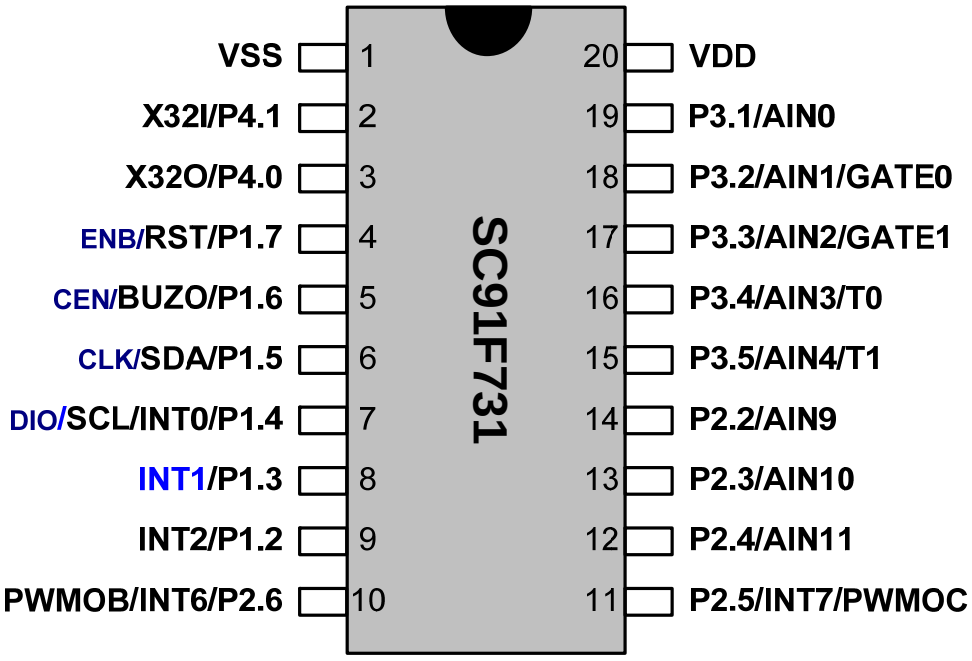
- 工作电压: 3.6V~5.5V
- 工作温度: -40 ~ 85 度
- 封装: SOP20、DIP20
- 内核: 超快速的 1T 8051
- 存储器: 8KB Flash ROM（MOVC 禁止寻址 0000~00FFH 的 256B）, 256B SRAM
- 系统时钟:

  内建 16M/8M Hz 振荡器
  - IC 工作的系统时钟,可通过编程器选择设定
  - 频率误差: 跨越 (4.5V~5.5V) 及 (-40 度, 85 度) 应用环境, 不超过 ±2%

  外接 32K 晶振
  - 不作为 IC 的系统时钟
  - 仅作为 Base Timer 来精确计时使用
  - 可以唤醒 STOP
  - 可通过 Code Option 取消此晶振，并作为 GPIO 使用
- 低电压复位（LVR）:
  - 复位电压有 4 级可选; 分别是: 3.5V、3.7V、3.9V、4.1V
  - Default 为用户烧写 Code Option 时的选择值
- Flash 烧写: 4 线串口烧写接口
- 中断（INT）:
  - TIMER0,TIMER1, X32K , PWM, SIF,ADC,INT0~2,,INT6~7 共 11 个中断源
  - INT0、INT2、INT6~7 外部中断为 4 个中断向量入口，仅下降沿触发
  - 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:
  - 18 个 4 种模式可设的强输出 GP I/O（包含 8 个大电流 LED 驱动）
  - 16 位 WDT，可选时钟分频比
  - 2 个标准 80C51 16 位定时器 TIMER0 及 TIMER1
  - 1 个串行通讯接口 SIF，可设 Slave 和 Master Mode
  - 1 路周期/占空比可调的 8 位 PWM ,可分时输出至 2 个不同管脚
  - 1 路 12 位 Buzzer
  - 1 路来自于 32K 晶振的准确 Base Timer
- 模拟外围:
  - **8 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 以及 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 INT0~2，INT6~7 唤醒 STOP MODE
  - 可由外部 32K 振荡产生的定时中断唤醒 STOP MODE

3. 管脚定义

3.1 管脚配置



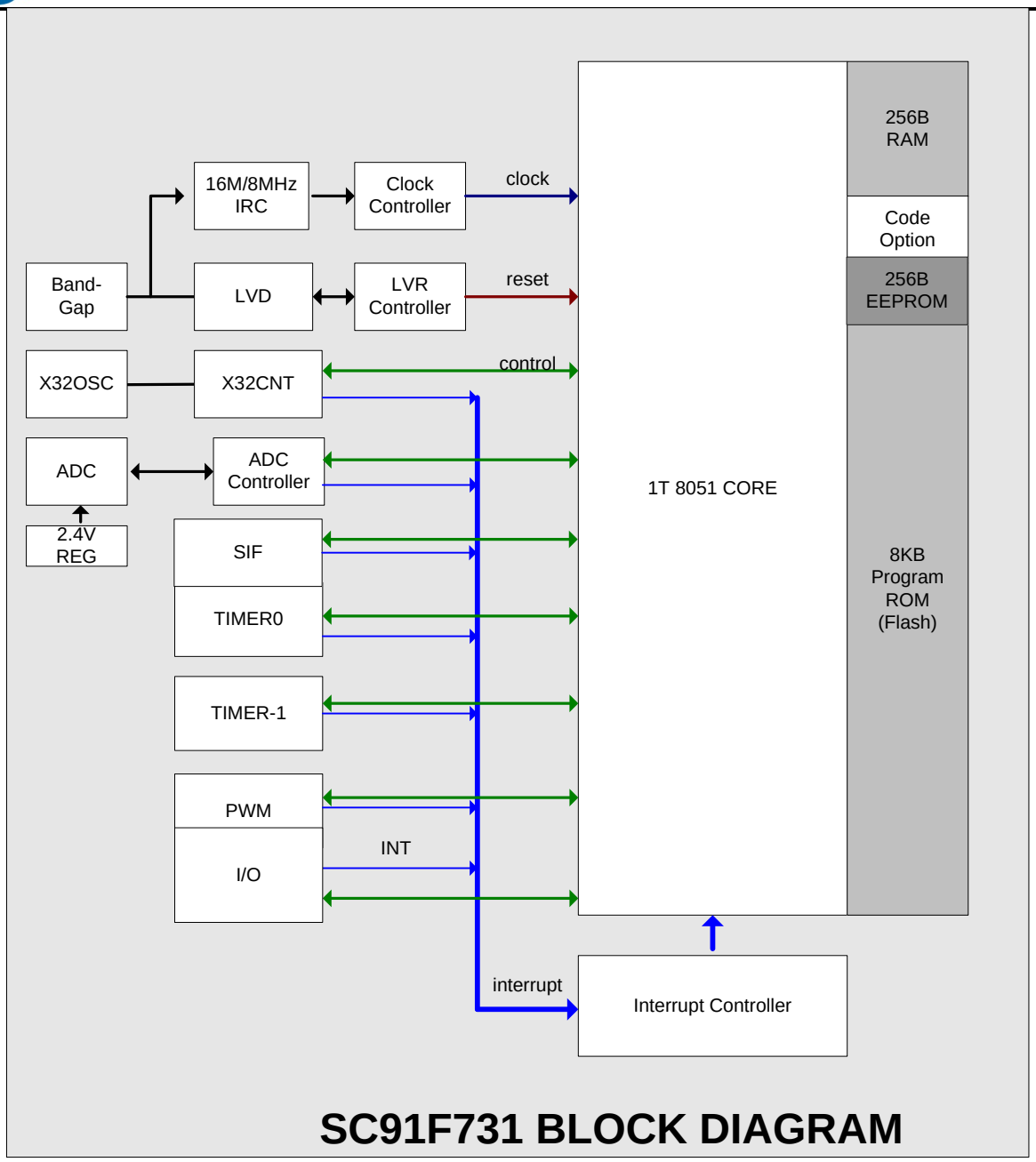
SC91F731管脚配置图

3.2 管脚定义

管脚编号（20Pin）	管脚名称	管脚类型	功能说明
1	VSS	Power	接地
2	P4.1/X32I	I/O	1) P4.1: GPIO P4.1（大电流 LED 驱动） 2) 外部 32K Crystal 输入脚
3	P4.0/X32O	I/O	1) P4.0 : GPIO P4.0（大电流 LED 驱动） 2) 外部 32K Crystal 输出脚
4	P1.7/RST	I/O	1) P1.7: GPIO P1.7（大电流 LED 驱动） 2) 外部复位管脚 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。 3) 烧写管脚
5	P1.6/BUZO	I/O	1) P1.6 : GPIO P1.6（大电流 LED 驱动） 2) BUZO: Buzzer 的输出 3) 烧写管脚
6	P1.5/SDA	I/O	1) P1.5: GPIO P1.5（大电流 LED 驱动） 2) SDA: 串口通讯 SIF 的 SDA 3) 烧写管脚
7	INT0/P1.4/SCL	I/O	1) INT0: 外部中断 0 2) P1.4: GPIO P1.4（大电流 LED 驱动） 3) SCL: 串口通讯 SIF 的 SCL 4) 烧写管脚
8	INT1/P1.3	I/O	1) INT1 : 外部中断 1（可通过 SFR INT1IT 来设置上升沿、下降沿或者双沿中断，做过零检测功能时建议使用此口） 2) P1.3 : GPIO P1.3（大电流 LED 驱动）
9	INT2/P1.2	I/O	1) INT2 : 外部中断 2 2) P1.2: GPIO P1.2（大电流 LED 驱动）
10	INT6/P2.6/PWMOB	I/O	1) INT6: 外部中断 6 2) P2.6 GPIO P2.6 3) PWMOB: PWM 输出 B 路
11	INT7/P2.5/PWMOC	I/O	1) INT7: 外部中断 7 2) P2.5

			GPIO P2.5 3) PWMOC: PWM 输出 C 路
12	**P2.4/AIN11**	I/O	1) P2.4 : GPIO P2.4 2) AIN11 : ADC 输入通道 11
13	**P2.3/AIN10**	I/O	1) P2.3 : GPIO P2.3 2) AIN10 : ADC 输入通道 10
14	**P2.2/AIN9**	I/O	1) P2.2 : GPIO P2.2 2) AIN9 : ADC 输入通道 9
15	**P3.5/AIN4**	I/O	1) P3.5 : GPIO P3.5 2) AIN4 : ADC 输入通道 4
16	**P3.4/AIN3**	I/O	1) P3.4 : GPIO P3.4 2) AIN3 : ADC 输入通道 3
17	**P3.3/AIN2**	I/O	1) P3.3 : GPIO P3.3 2) AIN2 : ADC 输入通道 2
18	**P3.2/AIN1**	I/O	1) P3.2 : GPIO P3.2 2) AIN1 : ADC 输入通道 1
19	**P3.1/AIN0**	I/O	1) P3.1 : GPIO P3.1 2) AIN0 : ADC 输入通道 0
20	**VDD**	Power	电源 3.6V – 5.5V

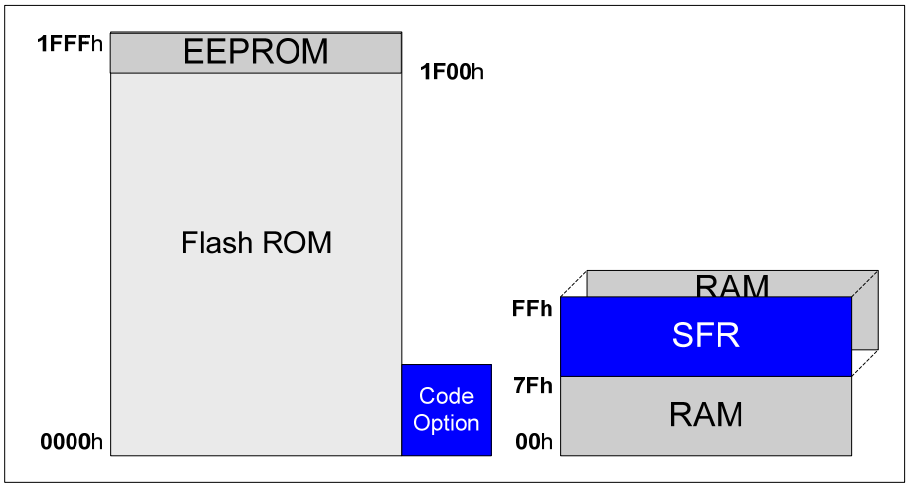
4. 内部方框图



SC91F731 BLOCK DIAGRAM

5. FLASH ROM和SRAM结构

SC91F731 的 Flash ROM 和 SRAM 结构如下：



Code Option 客户代码区

5.1 FLASH ROM

SC91F731 有 8KB 的 Flash ROM，ROM 地址为 0000H~1FFFH，此 8KB Flash ROM 可反复擦写 10 万次。0000H~1EFFH 的 7.75KB 区间用户可以用作应用程序空间，最高地址的 256B 空间即 1F00H~1FFFH 地址，用户可用作 EEPROM 使用，即通常所说的 IAP 功能，IC 内部有命令可直接修改此部分内容。用户不使用 EEPROM，则整体的 8KB 都可以作为应用程序空间使用。此 8KB FLASH ROM 的 7.75KB Flash 和 0.25KB EEPROM 可通过 SinOneChip 提供的专用 ICP 烧写器(SOC Pro51/DPT51)来进行编程及擦除。低位地址的 256B 即 0000H~00FFH 区间的 ROM 部分不能通过 MOVC 指令寻址。

SC91F731 的 8KB Flash ROM 中地址最高位的 4 个 Byte 中（也是 IAP 地址的最高位），存储有实际 IRC 相对于 16M/8MHZ 的偏差值、内部实际参考电压相对于 2.4V 的偏差值。方便用户调用来修正 IRC 和 2.4V，可实现非常精准的计时和测量，以消除每颗 IC 之间的偏差。值得说明的是，此部分内容的丢失不会影响到 IC 的正常运行及 IRC、2.4V 的精度，只是用户无法进行更加精准的修正而已。如用户有使用此 4Bytes 的值，请注意在程序中进行 IAP 操作时勿对此 4 个 Bytes 地址进行 IAP 的写操作。

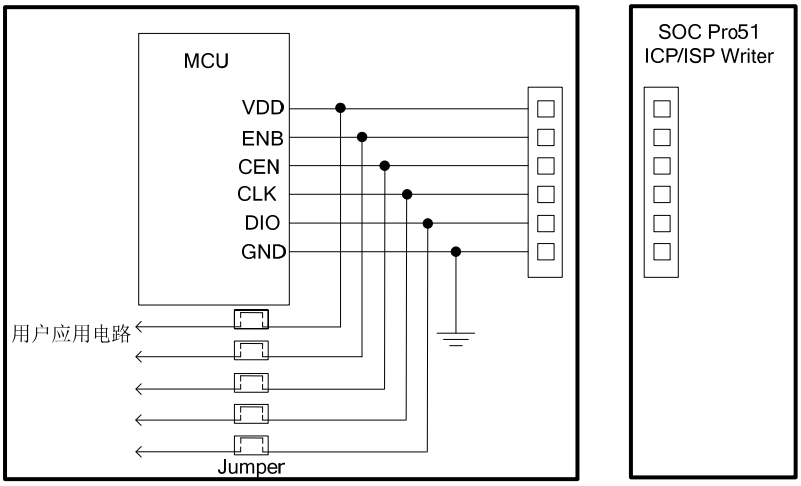
此 4Byte 内容的格式及所代表的意义如下：

<table>
<tr><th>地址</th><th>内部数值
16 进制（10 进制）</th><th>代表的含义</th></tr>
<tr><td>1FFF</td><td></td><td>地址 1FFE 内容的取反，用户可用来校验使用；</td></tr>
<tr><td>1FFE</td><td>00H（0）
至
FFH（255）</td><td>内部 IRC 实际值相对于 16M/8M Hz 的偏差值（精确至 1‰）；
<table>
<tr><th>符号位</th><th>内部的值</th><th>代表的含义</th></tr>
<tr><td>Bit7</td><td>Bit7~Bit0</td><td></td></tr>
<tr><td rowspan="3">1
(代表
负偏)</td><td>94H</td><td>16M/8M(1-20‰)Hz</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>81H</td><td>16M/8M(1-1‰)Hz</td></tr>
<tr><td></td><td>00H</td><td>精准的 16M/8M Hz</td></tr>
<tr><td rowspan="3">0
(代表
正偏)</td><td>14H</td><td>16M/8M(1+20‰)Hz</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>01H</td><td>16M/8M(1+1‰)Hz</td></tr>
</table></td></tr>
<tr><td>1FFD</td><td></td><td>地址 1FFC 内容的取反，用户可用来校验使用；</td></tr>
<tr><td>1FFC</td><td>00H（0）</td><td>内部参考电压实际值相对于 2.400V 的偏差值（精确至 1mv）；</td></tr>
</table>

	至 FFH（255）	符号位 Bit7 / 内部的值 Bit7~Bit0 / 代表的含义 1（代表负偏）：FFH 2.273V（2.400V-0.127V）；...；81H 2.399V（2.400V-0.001V） 00H 精准的 2.400V 0（代表正偏）：7FH 2.527V（2.400V+0.127V）；...；01H 2.401V（2.400V+0.001V）

SC91F731 的 8KB Flash ROM 可提供整区域（全部 8KB）或者 7.75KB（0000H~1EFFH）和 0.25KB（1F00H~1FFFH）分别单独的查空 BLANK、编程 PROGRAM、校验 VERIFY 和擦除 ERASE 功能，但不提供读取 READ 的功能。

SC91F731 的 Flash ROM 通过 Pin4（ENB）、Pin5（CEN）、Pin6（CLK）、Pin7（DIO）、VDD、VSS 来进行编程，具体连接关系如下：



ICP模式 Flash Writer编程连接示意图

5.2 非挥发性规划区(INFORMATION BLOCK EEPROM)

SC91F731 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
IFB1	VrefS[1:0]		USE32K	ENWDT		DISLVR	LVRS[1:0]	
IFB2	8MHz							

IFB1 编号	符号	说明
7,6	**VrefS[1:0]**	ADC 参考电压选择 00：选择 VDD 为 ADC 参考电压 01：选择内部 2.4V 为 ADC 参考电压 10：保留 11：保留
5	**USE32K**	外部 32K 使用设定 0：不使用外部 32K 振荡器，P4.0、P4.1 作为 GPIO 1：使用外部的 32K 振荡器，P4.1、P4.0 分别作为 32K OSCI 和 32K

		OSCO 使用
4	**ENWDT**	WDT 开关 0：WDT 无效 1：WDT 有效（但 IC 在执行 IAP 时 WDT 停止计数）
2	**DISLVR**	LVR 开关 0：LVR 有效 1：LVR 无效
1,0	**LVRS [1:0]**	LVR 电压选择控制 00： 4.1V 复位 01： 3.9V 复位 10： 3.7V 复位 11： 3.5V 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
7	**8MHz**	系统时钟选择 0：16MHz 1：8MHz（出厂默认值）

5.3 SRAM

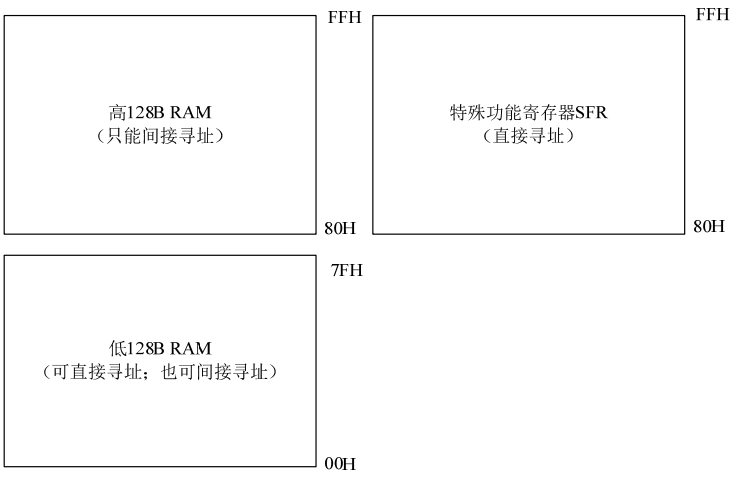
SC91F731 单片机内部集成了 256B 的 SRAM，供用户使用，地址范围为 00H~FFH。其中高 128B（地址 80H~FFH）只能间接寻址，低 128B（地址 00H~7FH）可直接寻址也可间接寻址。

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。

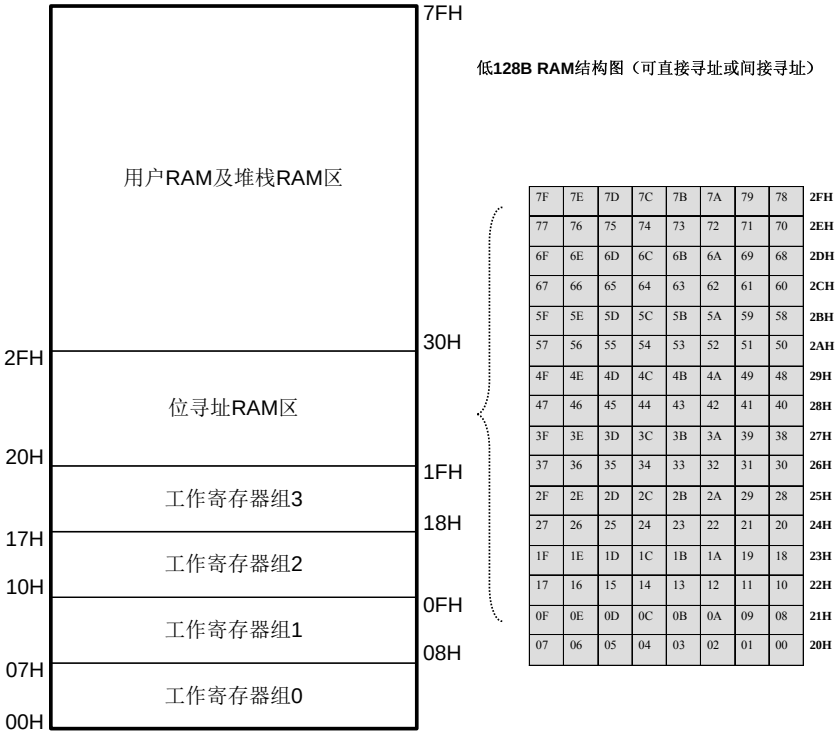
低 128B SRAM 区可分为三部分：①工作寄存器组 0~3，地址 00H~1FH,程序状态字寄存器 PSW 中的 RS0、RS1 组合决定了当前使用的工作寄存器，使用工作寄存器组 0~3 可加快运算的速度；②位寻址区 20H~2FH，此区域用户可以用作普通 RAM 也可用作按位寻址 RAM；按位寻址时，位的地址为 00H~7FH，（此地址按位编地址，不同于通用 SRAM 按字节编地址），程序中可由指令区分；③用户 RAM 区

SC91F731 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 80H 以后的单元区间。

256B RAM结构图



低 128B RAM 结构如下：



6. 特殊功能寄存器(SFR)

6.1 SFR 映像

SC91F731 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。

SC91F731 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		PWMDTY	PWMCFG			保留

F0H	B						RSTCFG	保留
E8H			IAPKEY		IAPADL	IAPDAT	IAPCTL	保留
E0H	ACC							
D8H	SIFSTA							
D0H	PSW				SIFCFG	SIFCTL	SIFTXD	SIFRXD
C8H		WDTCR	X32CTL					
C0H	P4		P4CFG0	BUZTGPH	BUZTGPL	ADCCR	ADCVH	ADCVL
B8H	IP					ADCCFG0	ADCCFG1	
B0H	P3	P3CFG1	P3CFG0	保留	EXIP	EXIE		
A8H	IE							
A0H	P2	P2CFG1	P2CFG0					
98H								
90H	P1	P1CFG1	P1CFG0	INT1IT				
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 EFH、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 3 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	功能	7	6	5	4	3	2	1	0	初始值
SP	**81H**	堆栈指针	SP[7:0]								*00000111*
DPL	**82H**	数据指针低位	DPL[7:0]								*00000000*
DPH	**83H**	数据指针高位	DPH[7:0]								*00000000*
PCON	**87H**	电源管理寄存器	-	-	-	-	-	-	STOP	-	*xxxxxx0x*
TCON	**88H**	定时器控制寄存器	TF1	TR1	TF0	TR0	-	-	-	-	*00000000*
TMOD	**89H**	定时器模式寄存器	GATE1	C/T1	M11	M01	GATE0	C/T0	M10	M00	*00000000*
TL0	**8AH**	定时器 0 低 8 位	TL0[7:0]								*00000000*
TL1	**8BH**	定时器 1 低 8 位	TL1[7:0]								*00000000*
TH0	**8CH**	定时器 0 高 8 位	TH0[7:0]								*00000000*
TH1	**8DH**	定时器 1 高 8 位	TH1[7:0]								*00000000*
TMCON	**8EH**	定时器时钟选择寄存器	-	-	-	-	-	-	T1M	T0M	*xxxxxxx00*
P1	**90H**	P1 口数据寄存器	P17	P16	P15	P14	P13	P12	-	-	*111111xx*
P1CFG1	**91H**	P1 Configuration - 1	P17M[1:0]		P16M[1:0]		P15M[1:0]		P14M[1:0]		*00000000*
P1CFG0	**92H**	P1 Configuration - 0	P13M[1:0]		P12M[1:0]		-		-		*0000xxxx*
INT1IT	**93H**	INT1 中断类型寄存器	-	-	-	-	-	-	INT1ES[1:0]		*xxxxxx00*
P2	**A0H**	P2 口数据寄存器	-	P26	P25	P24	P23	P22	-	-	*x11111xx*
P2CFG1	**A1H**	P2 Configuration - 1	-		P26M[1:0]		P25M[1:0]		P24M[1:0]		*xx000000*
P2CFG0	**A2H**	P2 Configuration - 0	P23M[1:0]		P22M[1:0]		-		-		*0000xxxx*
IE	**A8H**	中断使能寄存器	EA	EADC	ESIF	EPWM	ET1	EX32K	ET0	-	*0000000x*
P3	**B0H**	P3 口数据寄存器	-	-	P35	P34	P33	P32	P31	-	*xx11111x*
P3CFG1	**B1H**	P3 Configuration - 1	-		-		P35M[1:0]		P34M[1:0]		*xxxx0000*
P3CFG0	**B2H**	P3 Configuration - 0	P33M[1:0]		P32M[1:0]		P31M[1:0]		-		*000000xx*
EXIP	**B4H**	外部中断优先权设置	IPEX7	IPEX6	-	-	-	IPEX2	IPEX1	IPEX0	*00xxx000*
EXIE	**B5H**	外部中断使能设置	EINT7	EINT6	-	-	-	EINT2	EINT1	EINT0	*00xxx000*
IP	**B8H**	中断优先权寄存器	-	IPADC	IPSIF	IPPWM	IPT1	IPX32K	IPT0	-	*x000000x*
ADCCFG0	**BDH**	P2 口配置寄存器	VREFS[1:0]		-	P24AIN11	P23AIN10	P22AIN9	-	-	*nnx000xx*
ADCCFG	**BEH**	P3 口配置寄存器	-	-	P35AIN4	P34AIN3	P33AIN2	P32AIN1	P31AIN0	-	*xx00000x*

1											
P4	C0H	P4 口数据寄存器	-	-	-	-	-	-	P41	P40	xxxxxx11
P4CFG0	C2H	P4 Configuration - 0	-	-	-	-	P41M[1:0]		P40M[1:0]		xxxx1010
BUZTGPH	C3H	BUZZER 控制寄存器高位	ENBUZ	-	-	-	BUZTGP[11:8]				0xxx1111
BUZTGPL	C4H	BUZZER 控制寄存器低位	BUZTGP[7:0]								11111111
ADCCR	C5H	ADC 控制寄存器	ENADC	ADCS	LOWSP	EOC	ADCIS[3:0]				00000000
ADCVH	C6H	ADC 转换结果高字节	ADCV[9:2]								00000000
ADCVL	C7H	ADC 转换结果低字节	-	-	-	-	-	-	ADCV[1:0]		xxxxxx00
WDTCR	C9H	看门狗控制寄存器	ENWDT	-	-	CLRWDT	-	-	WDTCKS[1:0]		nxx0xx00
X32CTL	CAH	32K BaseTimer 控制寄存器	ENX32	FE	-	X32IF	-	-	X32IFS[1:0]		00x0xx00
PSW	D0H	程序状态字寄存器	CY	AC	F0	RS1	RS0	OV	-	P	000000x0
SIFCFG	D4H	SIF 设置寄存器	ENSI	INVI	-	-	SIMOD[2:0]			ACKO	00xx0000
SIFCTL	D5H	SIF 控制寄存器	-	-	-	-	-	-	MCMD[1:0]		xxxxx000
SIFTXD	D6H	SIF 发送数据寄存器	SIFTXD[7:0]								00000000
SIFRXD	D7H	SIF 接收数据寄存器	SIFRXD[7:0]								xxxxxxxx
SIFSTA	D6H	SIF 状态寄存器	RTNACK	-	-	-	STPIF	TXIF	RXIF	STRIF	0xxx0000
ACC	E0H	累加器 ACC	ACC[7:0]								00000000
IAPKEY	EAH	IAP 保护门	IAPKEY[7:0]								00000000
IAPADL	ECH	IAP 地址低位	IAPADR[7:0]								11111111
IAPDAT	EDH	IAP 写入/读出 资料	IAPDAT[7:0]								11111111
IAPCTL	EEH	IAP 命令	-	-	-	-	PAYTIMES[1:0]		CMD[1:0]		xxxx0000
B	F0H	B 寄存器	B[7:0]								00000000
RSTCFG	F6H	复位脚配置寄存器	-	-	-	-	DISRST	DISLVR	LVRS[1:0]		xxxx0nnn
PWMCR	F8H	PWM 控制寄存器	ENPWM	PWMIF	-	-	-	DTY8	PWMOS[1:0]		00xxx000
PWMPRD	F9H	PWM 周期寄存器	PWMPRD[7:0]								11111111
PWMDTY	FBH	PWM 占空比寄存器	PWMDTY[7:0]								00000000
PWMCFG	FCH	PWM 配置寄存器	-	-	-	INV	-	CKS[2:0]			xxx0x000

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 R1~R3，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	**CY**	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	**AC**	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	**F0**	用户标志位
4~3	**RS1、RS0**	工作寄存器组选择位: RS1 | RS0 | 当前使用的工作寄存器组 0~3 0 | 0 | 工作寄存器组 0 （00H~07H） 0 | 1 | 工作寄存器组 1 （08H~0FH） 1 | 0 | 工作寄存器组 2 （10H~17H） 1 | 1 | 工作寄存器组 3 （18H~1FH）
2 0	**OV** **P**	溢出标志位 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

SC91F731 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。

内部精准 2.4V 的值被保存在 8K Flash ROM 地址为 1FFCH 的 Byte 中，也就是存放在供用户可使用的 EEPROM 中。用户可以在使用时通过 MOVC 调用内部的准确值，以便于软件修正因每颗电压偏差带来的影响。所以，用户使用中应避免软件的 IAP 操作去更改此部分的内容。保存的格式及意义请参考 Flash ROM 部分的说明。

7.2 上电复位过程

SC91F731 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 SC91F731 会一直处于复位的情况, 直到供应给 SC91F731 的电压高达一定程度后, 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可选择最低的 LVR 电压后 (3.5V)，复位阶段才会完成。

调入信息阶段

在 SC91F731 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，直到电压过了最低的 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（Code Option）读出一个 byte 数据存放到内部系统寄存器中。直到预热计数器到达 1023 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, SC91F731 开始从 Flash 中读取指令代码即进入正常操作阶段。

7.3 复位方式

SC91F731 有 5 种复位方式：①外部 RST 复位②低电压复位 LVR③上电复位 POR④软件复位⑤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 SC91F731 一定宽度的复位脉冲信号，来实现 SC91F731 的复位。

RST/P1.7 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 P17 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

SC91F731 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 4.1V, 另外也可以选择 3.9V、3.7V、3.5V。

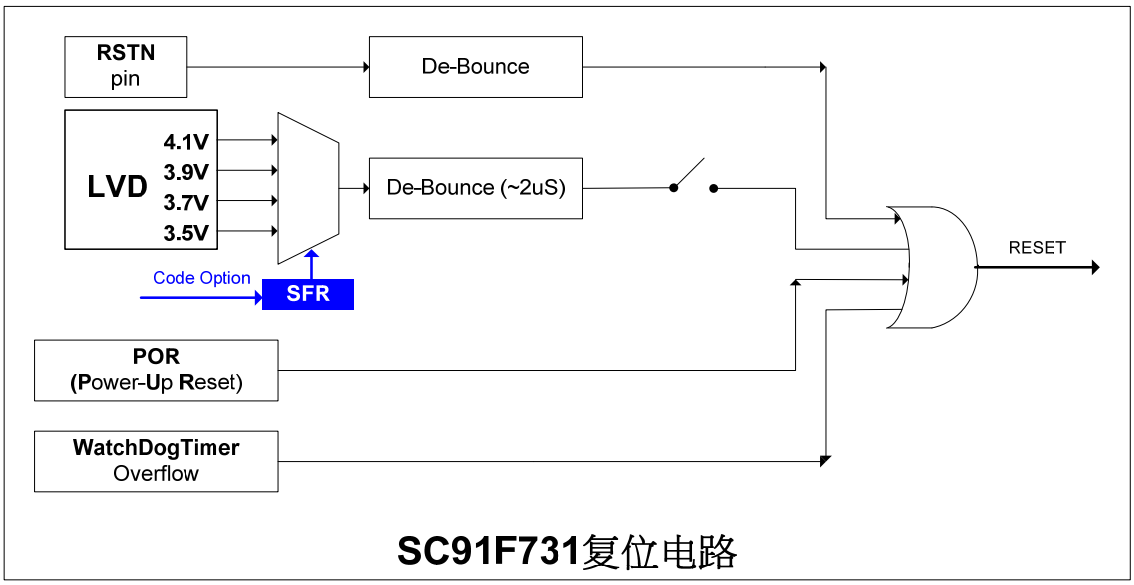
RSTCFG (F6h)复位设置寄存器（读/写）

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

位编号	位符号	说明
7~4	**保留位**	保留位
3	**DISRST**	IO/RST 复位切换控制 0：P1.7 当复位脚使用 1：P1.7 当正常的 I/O 管脚使用
2	**DISLVR**	LVR 使能设置 0：LVR 正常使用 1：LVR 无效

1,0	**LVRS [1:0]**	LVR 电压选择控制 00： 4.1V 复位 01： 3.9V 复位 10： 3.7V 复位 11： 3.5V 复位

SC91F731 位部分电路结构图如下：



SC91F731复位电路

7.3.3 上电复位POR

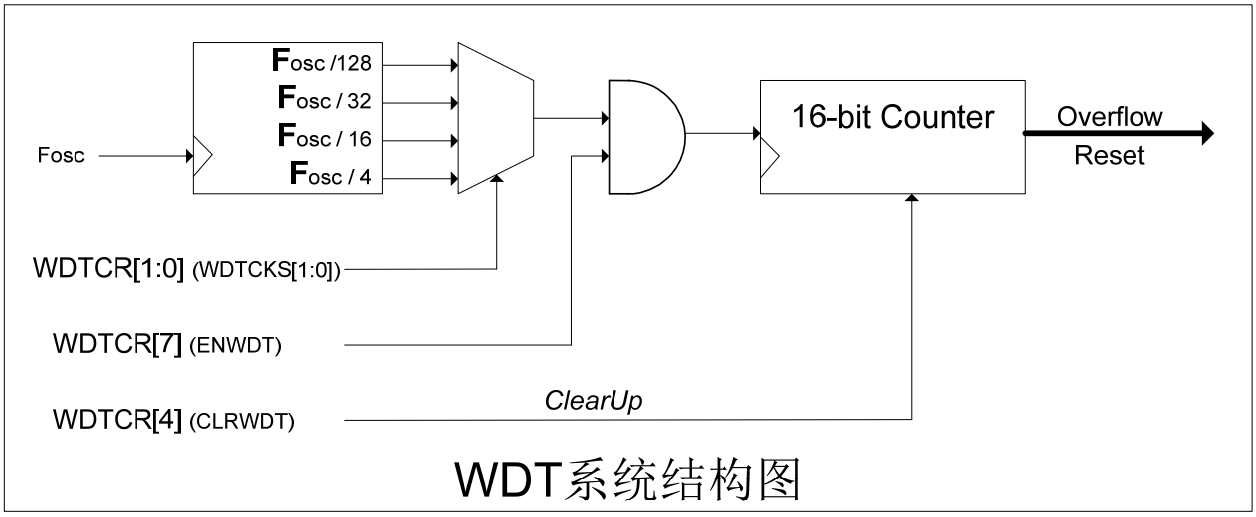
SC91F731 内部有上电复位电路，当电源电压 VDD 上升到复位电压点时，系统自动复位。

7.3.4 软件复位

SC91F731 提供一种特别的复位方式，以供用户在特殊场合使用。软件复位方法：先将 RST/P1.7 管脚设置为 P1.7，然后将 P1.7 设置为准双向/强推挽/开漏输出模式并输出低电平，最后将 RST/P1.7 管脚设置为 RST，这时会使系统复位。

7.3.5 看门狗复位WDT

SC91F731 有一个 16 位的 WDT，其时钟源为内部的 16M/8M Hz 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	n	x	x	0	x	x	0	0

位编号	位符号	说明
7	**ENWDT**	WDT 开关（复位后，系统从 Code Option 调入此寄存器） 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 此位由系统硬件自动置 0
1,0	**WDTCKS [1:0]**	看门狗时钟选择

WDTCKS.1	WDTCKS.0	WDT 时钟频率	WDT 溢出时间
0	0	Fosc/128	524.288ms@16MHz 1.048S@8MHz
0	1	Fosc/32	131.072ms@16MHz 262.144ms@8MHz
1	0	Fosc/16	65.536ms@16MHz 131.072ms@8MHz
1	1	Fosc/4	16.384ms@16MHz 32.768ms@8MHz

7.3.6 复位初始状态

当 SC91F731 处于复位状态时，多数寄存器会回到其初始状态。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
SP	00000111	IP	x000000x
DPL	00000000	ADCCFG0	nnx00000
DPH	00000000	ADCCFG1	0000000x
PCON	xxxxxx0x	P4	xxxxxx11
TCON	0000xxxx	P4CFG0	xxxx1010
TMOD	00000000	WDTCR	nxx0xx00
TL0	00000000	X32CTL	00x0xx00
TL1	00000000	BUZTGPH	0xxx1111
TH0	00000000	BUZTGPL	11111111
TH1	00000000	ADCCR	00000000
TMCON	00000000	ADCVH	00000000
P1	11111111	ADCVL	xxxxxx00
P1CFG1	00000000	PSW	000000x0
P1CFG0	00000000	ACC	00000000
P2	11111111	IAPKEY	00000000
P2CFG1	00000000	IAPADL	11111111
P2CFG0	00000000	IAPDAT	11111111
IE	0000000x	IAPCTL	xxxx0000
P3	11111111	B	00000000
P3CFG1	00000000	RSTCFG	xxxx0nnn
P3CFG0	00000000	PWMCR	00xxxxxx
EXIP	00000000	PWMPRD	11111111
EXIE	00000000	PWMDTY	00000000
INT1IT	xxxxxx00	PWMCFG	x0xx0000

7.4 内部时钟电路 16M/8M Hz IRC

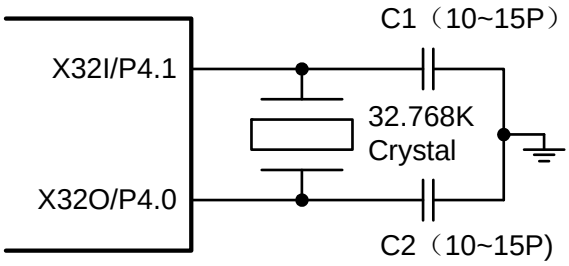
SC91F731 内建了一个振荡频率可调的高精度 IRC 作为系统时钟， 出厂时被精确地调校至 8MHz@5V/25℃, 用户可通过编程器修改至 16MHz 使用。 调校过程是过滤掉制程上的偏差对精度所造成的影响, 并把 16M/8M Hz 时所对应的参数写到内部 FLASH MEMORY 中 ,同时也会被保存至 1FFE 地址的 Flash ROM（也是 IAP Flash）中，供用户进行精确地偏差校准。

此 IRC 受工作的环境温度和工作电压影响会有一定的漂移。对于压漂（4.5V~5.5V）以及(-40℃~85℃)的温漂一般状况会在 2% 以内。

7.5 外部 32K CRYSTAL和BASE TIMER 控制

SC91F731 内建一个可外接 32.768K Hz Crystal 的振荡电路，该振荡器内部直接连接一个 17-bit 的 Base Timer, 可以把 CPU 从 STOP mode 唤醒, 并且产生 Interrupt。此 Base Timer 可用来准确计时，但不参与系统时钟。此用法可使用户在 Stop mode 下运行实时时钟、基准计时，即可以非常省电的进行计时。

P4.0/P4.1 作为 32K Base Timer 使用的接法电路如下：



P4.1/P4.0 作为 IO 使用或者作为外部 32K OSC 使用，只能由用户在编程器写入 Code 时的 Code Option 位 USEX32 来选择。

P4.1/P4.0 作为 IO 使用或者作为外部 32K OSC 使用，由编程器写入 Code Option 位选择。

Code Option：

位编号	7	6	5	4	3	2	1	0
符号	Vrefs[1:0]		USEX32	ENWDT	-	DISLVR	LVRS[1:0]	

位编号	位符号	说明
7	**USEX32**	外部 32K 振荡器开关 0：外部 32K 振荡器无效，P4.0/P4.1 作为 IO 使用(P4.0/P4.1 初始设置为输入模式) 1：外部 32K 振荡器有效，P4.0/P4.1 分别作为 X32OSCO/X32OSCI 使用

如用户选择外接 32K 振荡器，用户可在 SC91F731 软件中启动和关闭外部 32K 振荡器，并可选择 32K 中断的频率（0.25s/0.5s/1s/2s 四选一），其相关的 SFR 寄存器如下：

X32CTL (C2h) 32K BaseTimer 控制寄存器（读/写）

位编号	7	6	5	4	3	2	1	0
符号	ENX32	FE	-	X32IF	-	-	X32IFS[1:0]	
读/写	读/写	读/写	-	读/写	-	-	读/写	
上电初始值	0	0	x	0	x	x	0	0

位编号	位符号	说明
7	**ENX32**	32K OSC 启动控制 0：32K Hz 的 IRC 不启动 1：启动 32K Hz 的 IRC（需 IFB 的 USEX32K 设置为 1，才有效） 注意: 当设定 ENX32 为 1 后, 32KHz 的 Crystal Oscillator 可能需要耗费 10ms~25ms 的时间才能真正启动。 而当设定 ENX32 为 0 时, 内部的 Base Timer 会被清为 0。 所以， 第一次的 Base Timer 中断或许会慢一点, 但是只要 ENX32 固定在 1 的情况下, 第二次以后的中断就会非常

		准确。
6	**FE**	快速起振使能 Fast Enable. 这个 bit 如果被设定成 1, 32K Crystal Oscillator 会较快起振, 但是耗电也会比较大, 通常用于外部有比较大的负载的情况。 用户可以于非 STOP mode 的情况下, 把 ENX32 以及 FE 设定成 1 让 32K Crystal 起振, 当侦测到 X32IF 时, 再单独把 FE 设定成 0, 然后再进到 STOP mode。
4	**X32IF**	32K Base Timer 中断申请标志 当 CPU 接受 Base Timer 的中断后, 此标志位会被硬件自动清除。 用户也可以用软件清除.
1,0	**X32IFS[1:0]**	32K Base Timer 中断频率选择（Interrupt Frequency Selection） 00：每 0.25 秒产生一个 interrupt 01：每 0.5 秒产生一个 interrupt 10：每 1 秒产生一个 interrupt 11：每 2 秒产生一个 interrupt
5,3,2	**保留位**	保留位

使用注意事项：

- 如果 IFB 中的 USEX32 不是被设定成 1, 那么所有对 X32CTL 的写入都会失效 !
- 如果 IFB 中的 USEX32 被设定成 1, 那么对 X32CTL 的任何写入动作都会把同时把 32K Crystal Oscilator 的内部 Wakeup Counter 清为 0。
- 要改变 X32IFS[1:0] 前, 必须把 IE[2] 及 X32CTL[7] (ENX32) 设定成 0, 否则可能发生当机的情况。
- 在 32K Crystal Oscillator 起振后需将 FE 关闭，否则会影响 32K Crystal Oscillator 的频率

7.6 省电模式 STOP

SC91F731 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 16M/8M Hz 晶振就会停止, 进到 STOP 模式 , 达到省电功能。

在 STOP 模式下， 使用者可以通过外部中断 INT0~INT2，INT6~7 把 SC91F731 唤醒，用户也可用外部 32K 的 Base Timer 中断定时唤醒 STOP。

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 16M/8M Hz 晶振停止工作

8 中央处理单元CPU 及指令系统

8.1 CPU

SC91F731 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

SC91F731 系列的 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,内部数据存储器 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
算术操作指令			
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
逻辑操作指令			
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
布尔变量操作指令			
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
数据传送类指令			
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

LJML 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
SC91F731 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

SC91F731 单片机提供了 11 个中断源： Timer0、Timer1、X32K、PWM、SIF、ADC、INT0、INT1、INT2、INT6、INT7。这 11 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

SC91F731 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	中断优先权控制	中断向量	查询优先级	中断号（C51）	标志清除方式	能否唤醒 STOP
Timer0	Timer0 溢出	TCON[5] (TF0)	IE[1] (ET0)	IP[1]	000BH	1（高）	1	H/W Auto	不能
X32K	32K Base Timer 溢出	X32CTL[4] (X32IF)	IE[2] (EX32K)	IP[2]	0013H	2	2	H/W Auto	能
Timer1	Timer1 溢出	TCON[7] (TF1)	IE[3] (ET1)	IP[3]	001BH	3	3	H/W Auto	不能
PWM	PWM 溢出	PWMCR[6] (PWMIF)	IE[4] (EPWM)	IP[4]	0023H	4	4	必须用户清除	不能
SIF	SIF 相关的命令完成	STPIF RXIF TXIP STRIF	IE[5] (ESIF)	IP[5]	002BH	5	5	必须用户清除	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
INT2	下降沿	隐藏式	EXIE[2]	EXIP[2]	004BH	9	9	H/W Auto	能
INT6	下降沿	隐藏式	EXIE[6]	EXIP[6]	006BH	13	13	H/W Auto	能
INT7	下降沿	隐藏式	EXIE[7]	EXIP[7]	0073H	14（低）	14	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”。

SIF 中断：SIF 命令完成后后产生相应的中断标志 STRIF、TXIF、RXIF、STPIF 置为“1”。 使用者在 SIF 中断发生之后，进入中断服务程序时，必须用软件去清除它。

X32K 中断：32K Base Timer 在用户设置对应的溢出时间后发生溢出并将中断标志 X32IF 置为“1”，当单片机执行该 X32K 中断时，中断标志 X32IF 会被硬件自动清“0”。

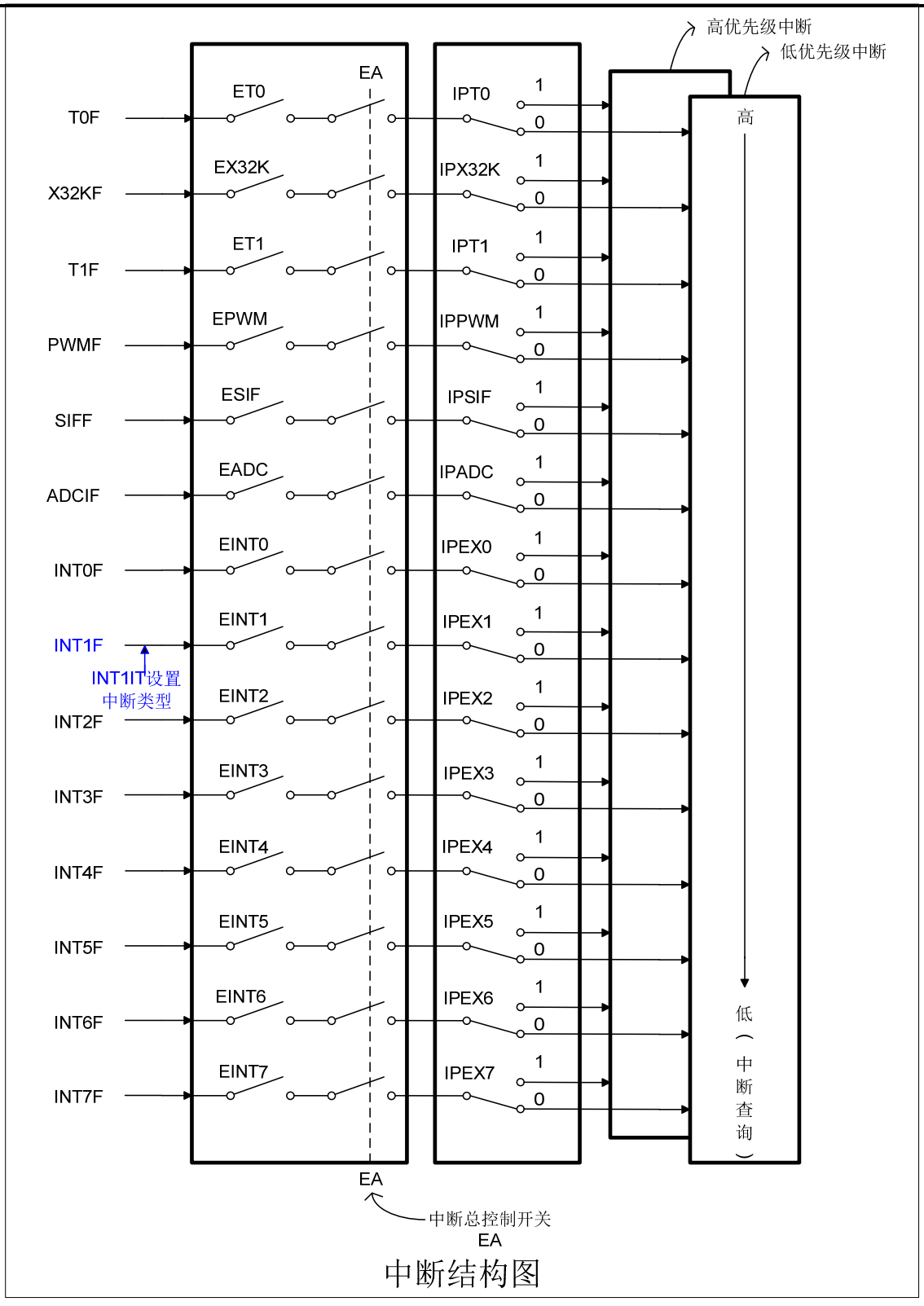
PWM 中断：当 PWM 计数器溢出时(也就是说: 计数器数到超过 PWMPRD 时), PWMIF 位(PWM Interrupt Flag)会被硬件自动置“1”，PWM 中断产生。在 PWM 中断发生后, 硬件并不会自动清除 PWMIF 位, 此 bit 必须由使用者的软件负责清除。

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 INTx(x=0、1、2、6、7)：外部中断 INT0~2、INT6~7 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 5 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、INT2、INT6~7 的外部中断仅下降沿触发，无需用户设置；INT1 是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 （INT1IT）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 INT0~2、INT6~7 还可以唤醒单片机的 STOP。

9.2 中断结构图

SC91F731 的中断结构如下图所示：



中断结构图

9.3 中断优先级

SC91F731 单片机的中断具有两个中断优先级，这些中断源的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：SC91F731 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	ESIF	EPWM	ET1	EX32K	ET0	-
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	x

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	**ESIF**	SIF 中断使能 0：关闭 SIF 中断 1：允许 SIF 中断
4	**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
3	**ET1**	Timer1 中断使能控制 0：关闭 TIMER1 中断 1：允许 TIMER1 中断
2	**EX32K**	32K Base Timer 中断使能控制 0：关闭 32K 中断 1：打开 32K 中断
1	**ET0**	Timer0 中断使能控制 0：关闭 TIMER0 中断 1：允许 TIMER0 中断
0	**保留位**	保留位

IP (B8h) 中断优先权寄存器(读/写)

位编号	7	6	5	4	3	2	1	0
符号	-	IPADC	IPSIF	IPPWM	IPT1	IPX32K	IPT0	-
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	x

位编号	位符号	说明
6	**IPADC**	ADC 中断优先权选择 0：ADC 中断优先权为低 1：ADC 中断优先权为高
5	**IPSIF**	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
4	**IPPWM**	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
3	**IPT1**	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
2	**IPX32K**	32K Base Timer 中断优先权控制 0：32K 中断优先权为低 1：32K 中断优先权为高
1	**IPT0**	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
7,0	**保留位**	保留位

EXIE (B5h) 外部中断使能寄存器(读/写)

位编号	7	6	5	4	3	2	1	0
符号	EINT7	EINT6	-	-	-	EINT2	EINT1	EINT0
读/写	读/写	读/写	-	-	-	读/写	读/写	读/写
上电初始值	0	0	x	x	x	0	0	0

位编号	位符号	说明
7~0	**EINTx** **(x=0、1、2、6、7)**	外部中断使能控制（EXIE.3~5 为保留位，请勿写 1） 0： 关闭外部中断 INTx 的中断 1： 允许外部中断 INTx 发生中断

EXIP (B4h) 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	IPEX7	IPEX6	-	-	-	IPEX2	IPEX1	IPEX0
读/写	读/写	读/写	读	读	读	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	**IPEXn** **(n=0、1、2、6、7)**	外部中断优先权选择(EXIP.3~5 为保留位，请勿写 1) 0 : 外部中断 INTn 的中断优先级是 “低” 1: 外部中断 INTn 的中断优先级是 “高”

INT1IT (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	**INT1ES[1:0]**	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**	

10. 定时器TIMER0 、TIMER1

SC91F731 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，P3.4/T0 和 P3.5/T1 管脚上的每一个脉冲，T0 和 T1 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
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									00000000b
TL1	**8BH**	定时器 1 低 8 位									00000000b
TH0	**8CH**	定时器 0 高 8 位									00000000b
TH1	**8DH**	定时器 1 高 8 位									00000000b
TMCON	**8EH**	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	**TF1**	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	**TR1**	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	**TF0**	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	**TR0**	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	**GATE1**	TMOD[7]控制定时器 1，置 0 且 TR1 置 1 时才打开 T1
6	**C/T1**	TMOD[6]控制定时器 1 0：定时器，T1 计数来源于 Fosc 分频 1：计数器，T1 计数来源于外部管脚 T1/P3.5
5,4	**M11,M01**	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	**GATE0**	TMOD[3]控制定时器 0，置 0 且 TR0 置 1 时才打开 T0
2	**C/T0**	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/P3.4
1,0	**M10,M00**	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	**T1FD**	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	**T0FD**	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	EX32	ET0	-
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	x

位编号	位符号	说明
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

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	IPPWM	IPT1	IPX32	IPT0	-
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	x

位编号	位符号	说明
3	**IPT1**	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	**IPT0**	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 。

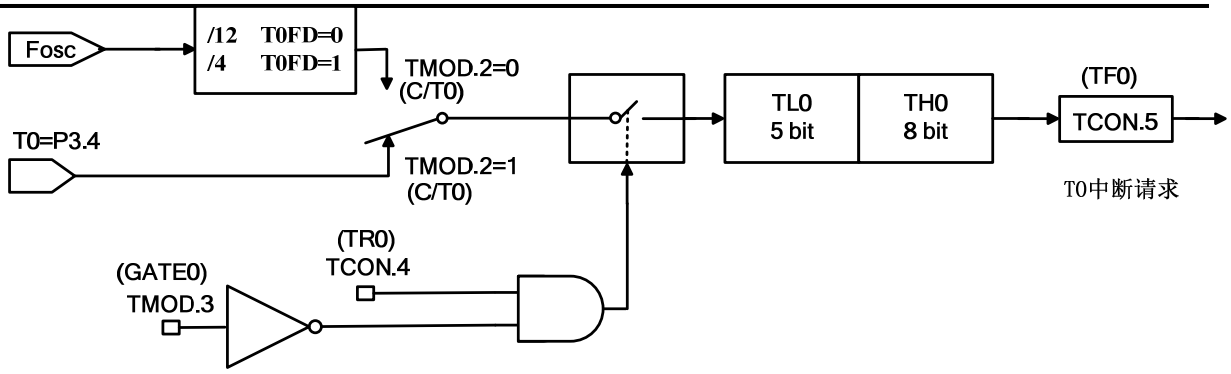
工作模式 0: 13 位计数器/定时器。

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（P3.4）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。

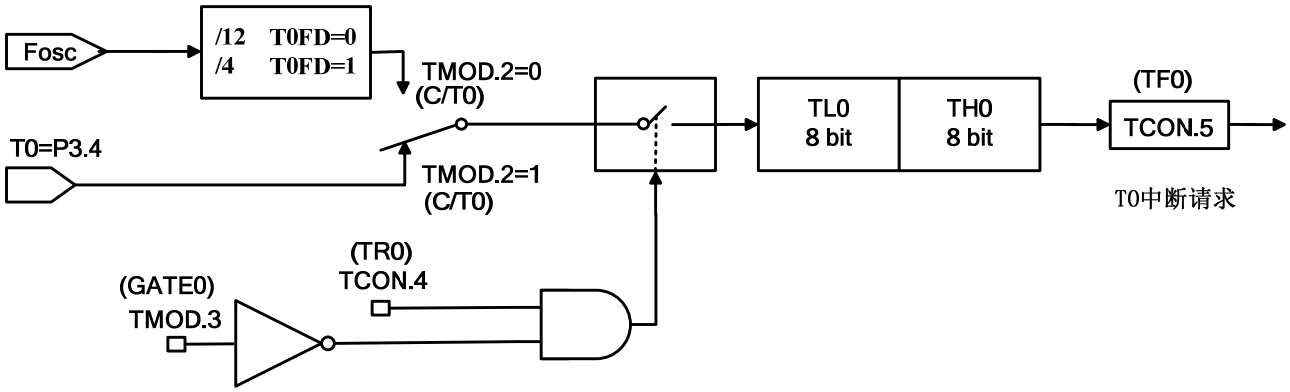
当作为定时器应用时，可配置 T0FD 来选择时钟源的分频比例。



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。



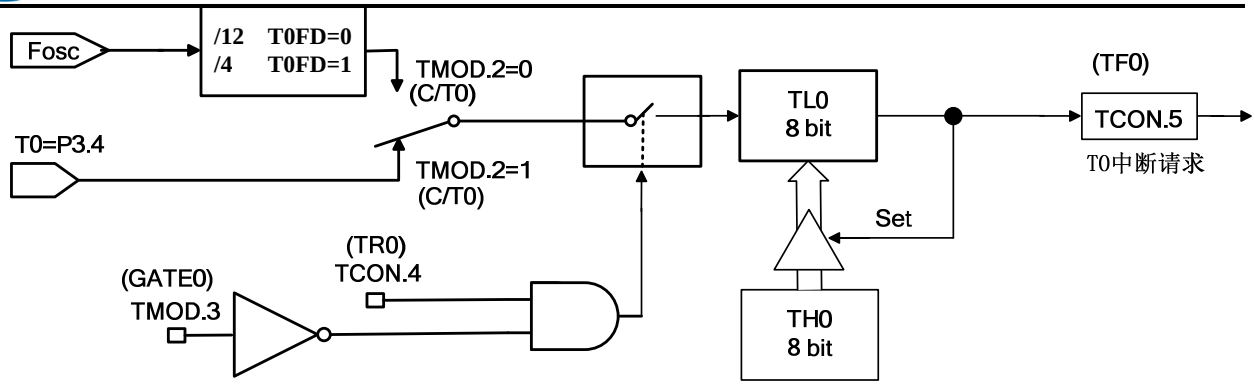
定时器/计数器工作模式1：16位定时器/计数器

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。



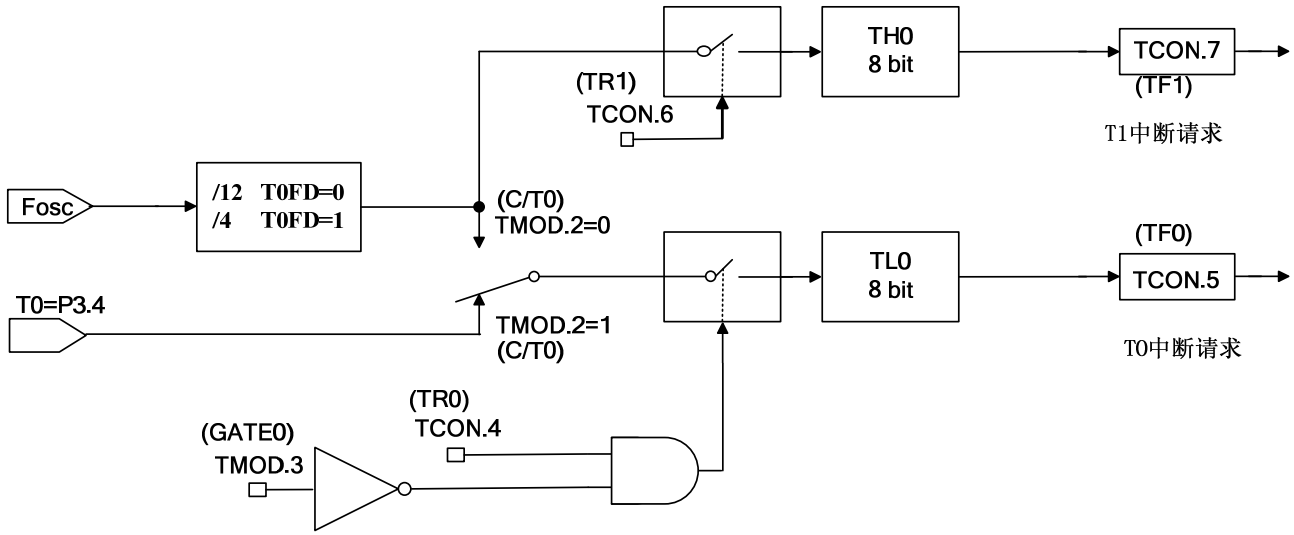
定时器/计数器工作模式2: 自动重载的8位定时器/计数器

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（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。当发生溢出及产生中断时，会按 T1 发生中断来进行相应的处理。

在 T0 被设为工作模式 3 时，T1 的 16 位计数器/定时器会停止计数，相当于 TR1=0，仅 T0 作为两个 8 位计数器使用，但 TH0 的高 8 位占用了 T1 的中断资源及 TCON 中寄存器的位置。



定时器/计数器工作模式3: 2个8位定时器/计数器

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 。

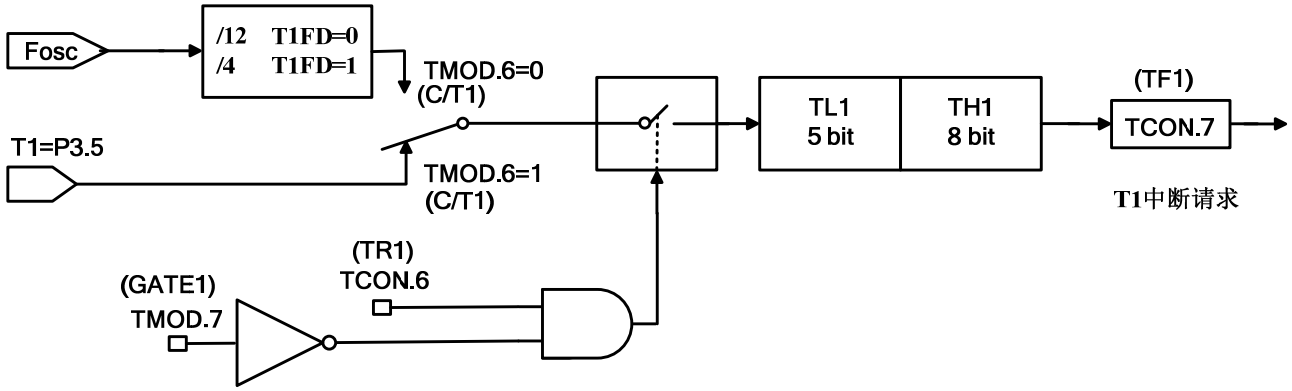
工作模式 0: 13 位计数器/定时器。

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 输入脚 T1（P3.5）的电平从高到低的变化，会使定时器 1 数据寄存器加 1。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。

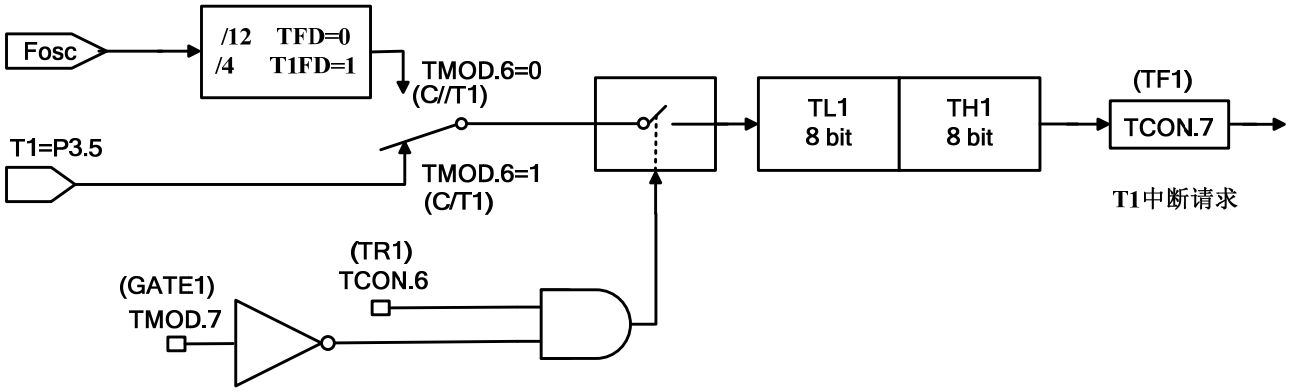
当作为定时器应用时，可配置 T1FD 来选择时钟源的分频比例。



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。



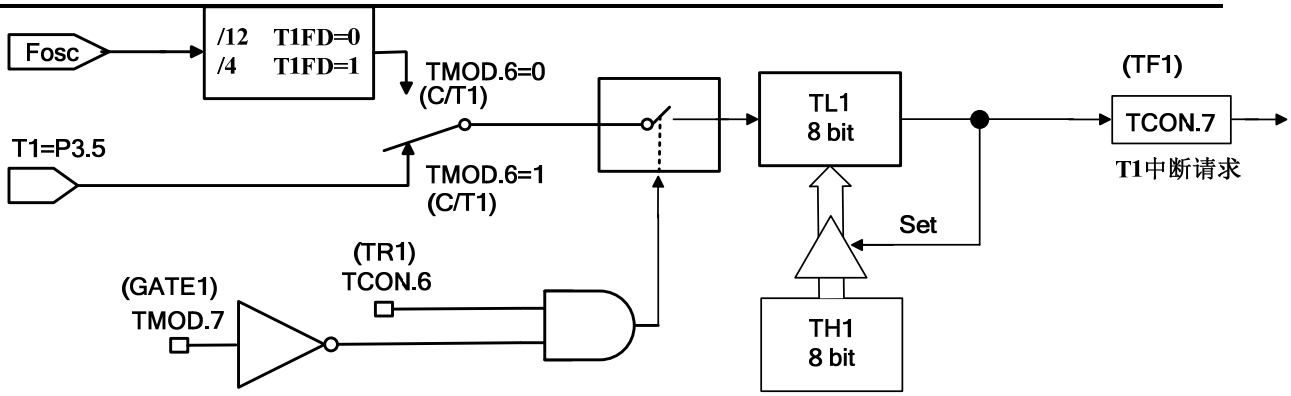
定时器/计数器工作模式1：16位定时器/计数器

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

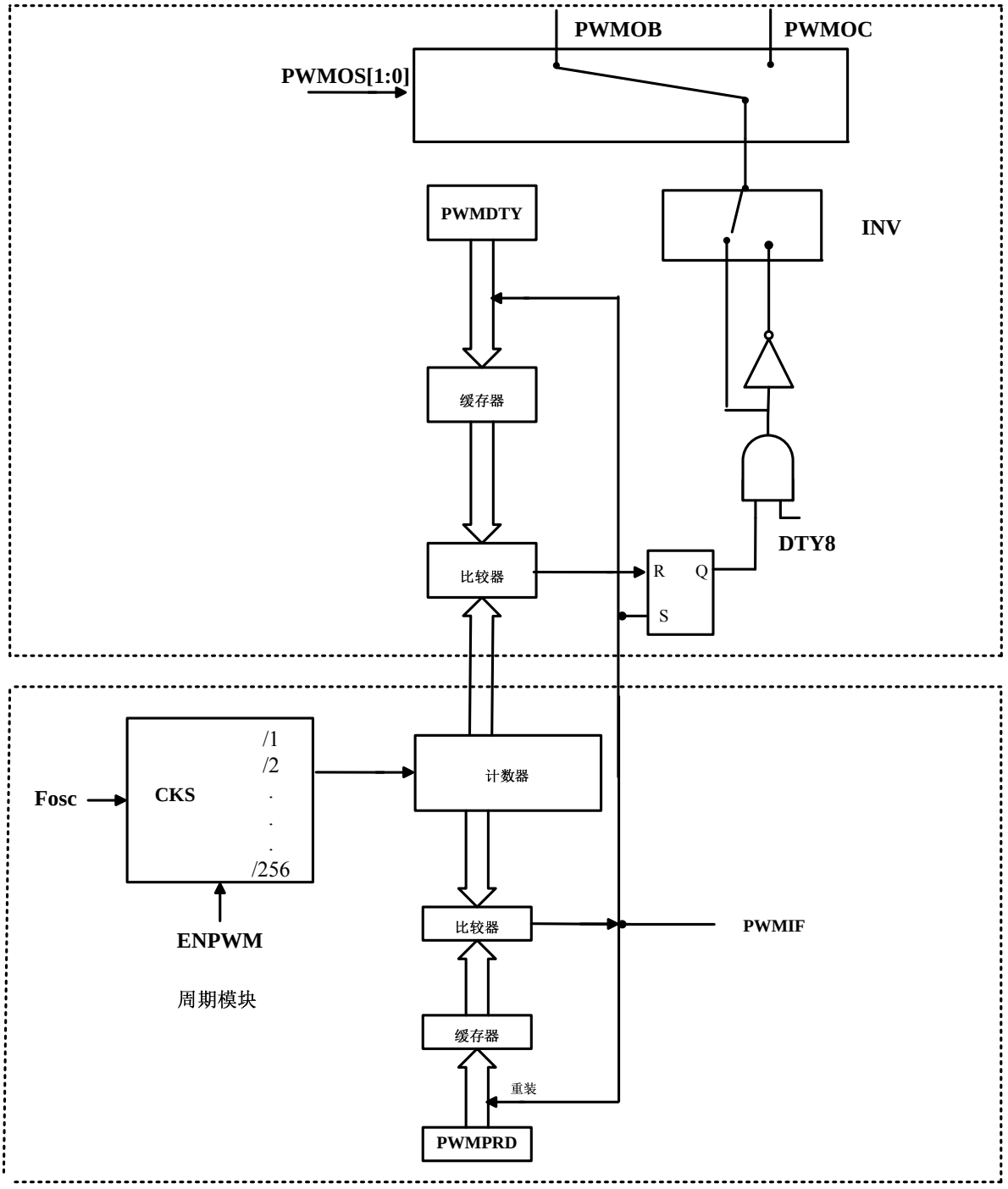
11. PWM

SC91F731 提供了一个独立的计数器, 它可以作为 PWM 输出。

SC91F731 的 PWM 具有的功能为①8 位 PWM 精度②占空比可设置③输出可设置正反向④提供 1 个 PWM 溢出的中断⑤可分时输出至 PWMOB、PWMOC。

SC91F731 的 PWM 可支持周期及占空比的调整，寄存器 PWMCR 控制 PWM 的相关设置，PWMCFG 设置 PWM 计数器计数时钟源及输出电平，PWMPRD 设置 PWM 的周期，PWMDTY 控制 PWM 的占空比。

11.1 PWM结构框图



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	-	-	-	DTY8	PWMOS[1:0]	
读/写	读/写	读/写			-	读/写	读/写	读/写
上电初始值	0	0	x	x	x	0	0	0

位编号	位符号	说明
7	**ENPWM**	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	**PWMIF**	PWM 中断请求标志位(**PWM I**nterrupt **F**lag) 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[4] (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。
2	**DTY8**	强制 PWM 固定输出高（Force PWM as HIGH) 1：强制把 PWM 的输出固定为 1 0：PWM 的输出由 PWM 计数器以及 PWMDTY 来决定
1,0	**PWMOS[1:0]**	PWM 输出通道选择（其中 P2.7 管脚未引出） 00：PWM 无 输出， P26, P25 都当作正常的 GPIO 01：保留 10：PWM 从 P2.6 输出，P2.6 PIN 当做 PWM 的输出 11：PWM 从 P2.5 输出，P2.5 PIN 当做 PWM 的输出
5,4,3	**保留位**	保留位

SFR **PWMPRD[7:0]** 是 PWM 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 00_h, 也就是说 PWM 的周期是 (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	**PWMPRD[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	**PWMPRD[7:0]**	PWM 的周期设置; 此数值代表 PWM 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 Fosc 分频的分频比。PWM 还可以被 PWMCFG[4]中的 INV 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	-	-	-	INV	-	CKS[2:0]		
读/写	-	-	-	读/写	-	读/写	读/写	读/写
上电初始值	x	x	x	0	x	0	0	0

位编号	位符号	说明
4	**INV**	PWM 输出反向控制(**INV**erse PWM Output) 1 ： 把 PWM 的输出反向 0 ： PWM 的输出不反向
2~0	**CKS[2:0]**	PWM 时钟源选择(PWM **C**loc**K** source **S**elector) 000：Fosc 001：Fosc/2 010：Fosc/4

		011：Fosc/8 100：Fosc/32 101：Fosc/64 110：Fosc/128 111：Fosc/256
7,6,5,3	**保留位**	保留位

PWMDTY (FBh) PWM 高电平设置寄存器（读/写）

位编号	7	6	5	4	3	2	1	0
符号	**PWMDTY[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	**PWMDTY[7:0]**	PWM 占空比长度设置； PWM 的高电平宽度 是 (PWMDTY[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	ESIF	EPWM	ET1	EX32	ET0	-
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	x

位编号	位符号	说明
4	**EPWM**	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) 中断优先权寄存器(读/写)

位编号	7	6	5	4	3	2	1	0
符号	-	IPADC	IPSIF	IPPWM	IPT1	IPX32	IPT0	-
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	x

位编号	位符号	说明
4	**IPPWM**	PWM 中断优先权选择 0：设定 PWM 的中断优先级是 “低” 1：设定 PWM 的中断优先级是 “高”

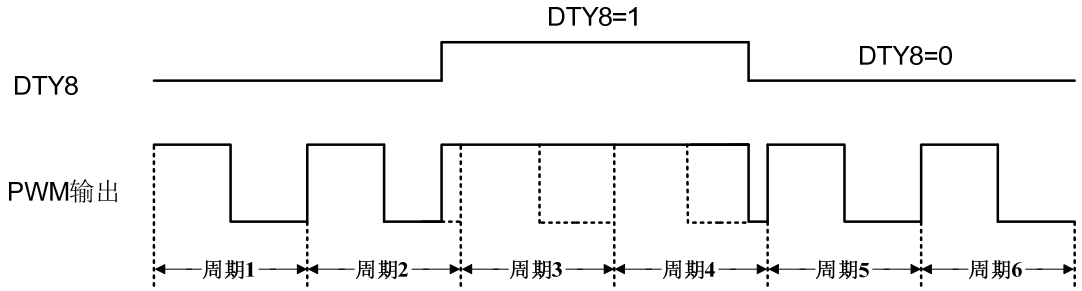
注意事项：

1， ENPWM 位能控制 PWM 模块是否工作。
2， ENPWMO 位能选择 PWM 口作为 GPIO 还是作为 PWM 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 PWMO=0, 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。

11.3 PWM波形及用法

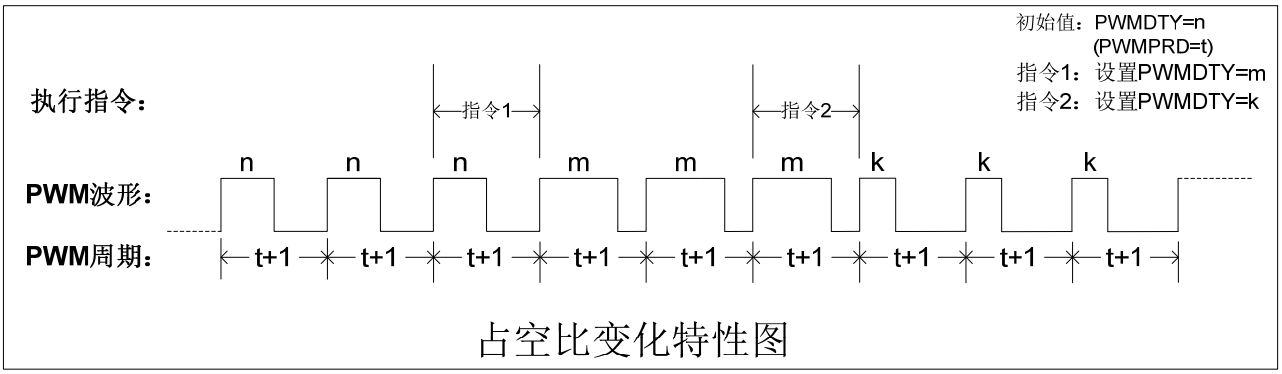
各 SFR 参数改变对 PWM 波形影响如下所述：

①DTY8 改变特性



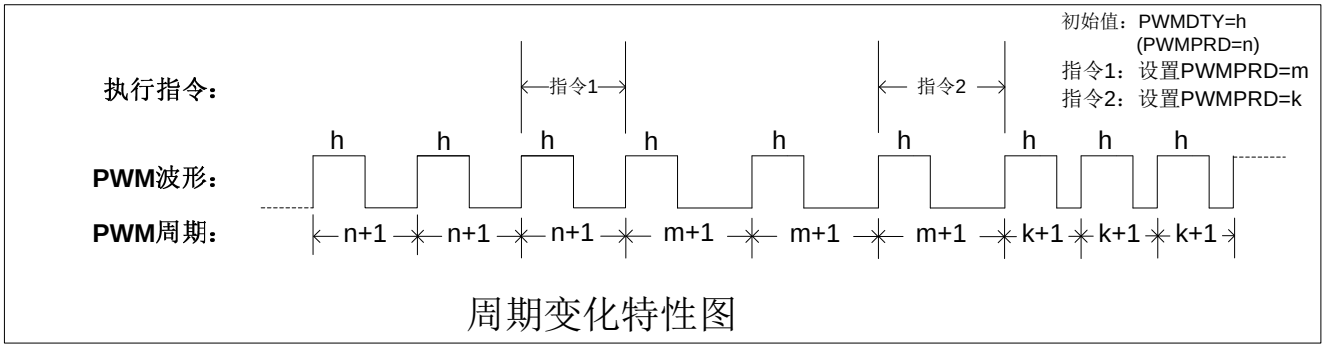
当 PWM 输出波形时，若 DTY8(PWMCR.1)改变，PWM 波形会立即改变。如上图所示，在周期 2 中使 DTY8 置 1，PWM 会立即响应，固定输出高；在周期 4 某处 DTY8 清 0，PWM 立即响应，取消固定输出。

②占空比变化特性



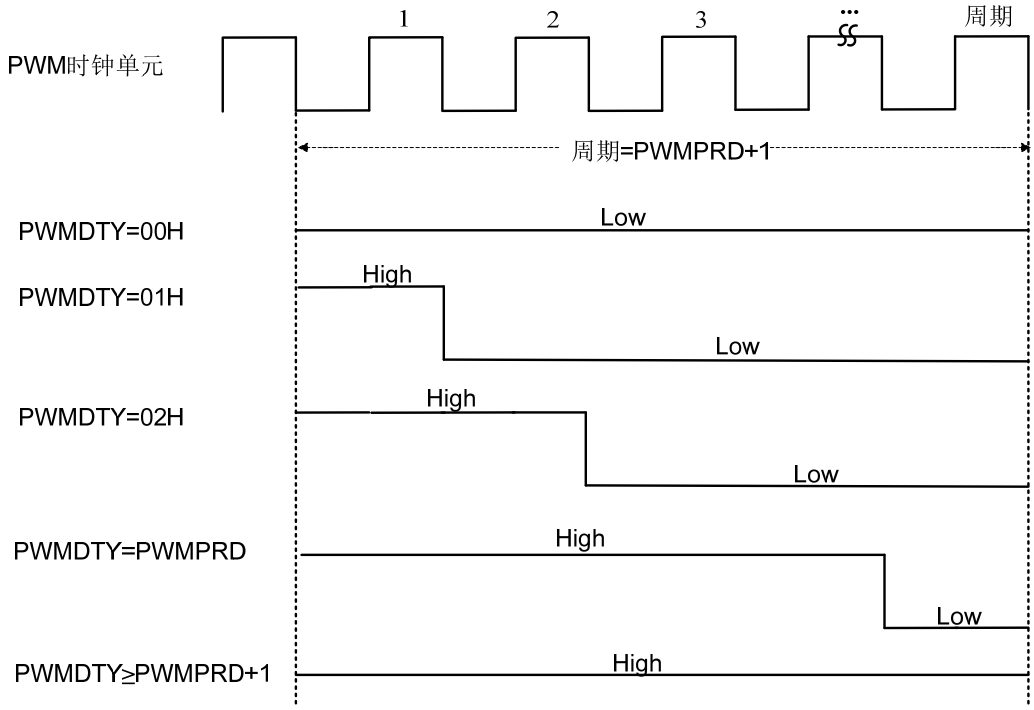
当 PWM 输出波形时，若需改变占空比，可通过改变高电平设置寄存器(PWMDTY)的值实现。但需要注意，更改 PWMDTY 的值，占空比不会立即改变，而是等待本周期结束，在下个周期发生改变。相关波形输出如上图所示。

③周期变化特性



当 PWM 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，在下个周期改变，参考上图所示。

③ 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 PWM 输出反向控制 INV 初始为 0，若需得到相反结果，可置 PWMCFG.4(INV)为 1。需要注意 INV 的变化特性也 DTY8 相同，更改则立即响应。

12 BUZZER

SC91F731 内建一个频率可调整的蜂鸣器输出 BUZZER，该输出同 P1.6 共用管脚。蜂鸣器的输出频率来源于系统时钟，用户可通过改变 BUZTGP[11:0]来改变蜂鸣器计数器设置，从而改变输出的频率。输出频率的可调范围为 2KHz ~ 8MHz@16MHz 或者 1KHz ~ 4MHz@8MHz，占空比固定是 50%。分频比例由一个 12 位的寄存器 BUZTGP[11:0]控制，BUZZER 输出频率的计算方法为：16M(8M)Hz / (2 * {BUZTGP[11:0]+1 })

BUZTGPH 蜂鸣器反转输出计数器高位（读/写）

位编号	7	6	5	4	3	2	1	0
符号	ENBUZ	-	-	-	BUZTGP[11:8]			
读/写	-	-	-	-	读/写	读/写	读/写	读/写
上电初始值	0	x	x	x	1	1	1	1

BUZTGPL 蜂鸣器反转输出计数器低位（读/写）

位编号	7	6	5	4	3	2	1	0
符号	BUZTGP[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	**ENBUZ**	BUZZER 和 P1.6 口切换控制 0：P1.6 当做正常的 I/O 使用 1：启动 Buzzer 功能, 并且设定 P1.6 为 Buzzer 的强推挽输出
3~0	**BUZTGP[11:8]**	BUZZER 输出频率控制高 4 位
7~0	**BUZTGP[7:0]**	BUZZER 输出频率控制低 8 位

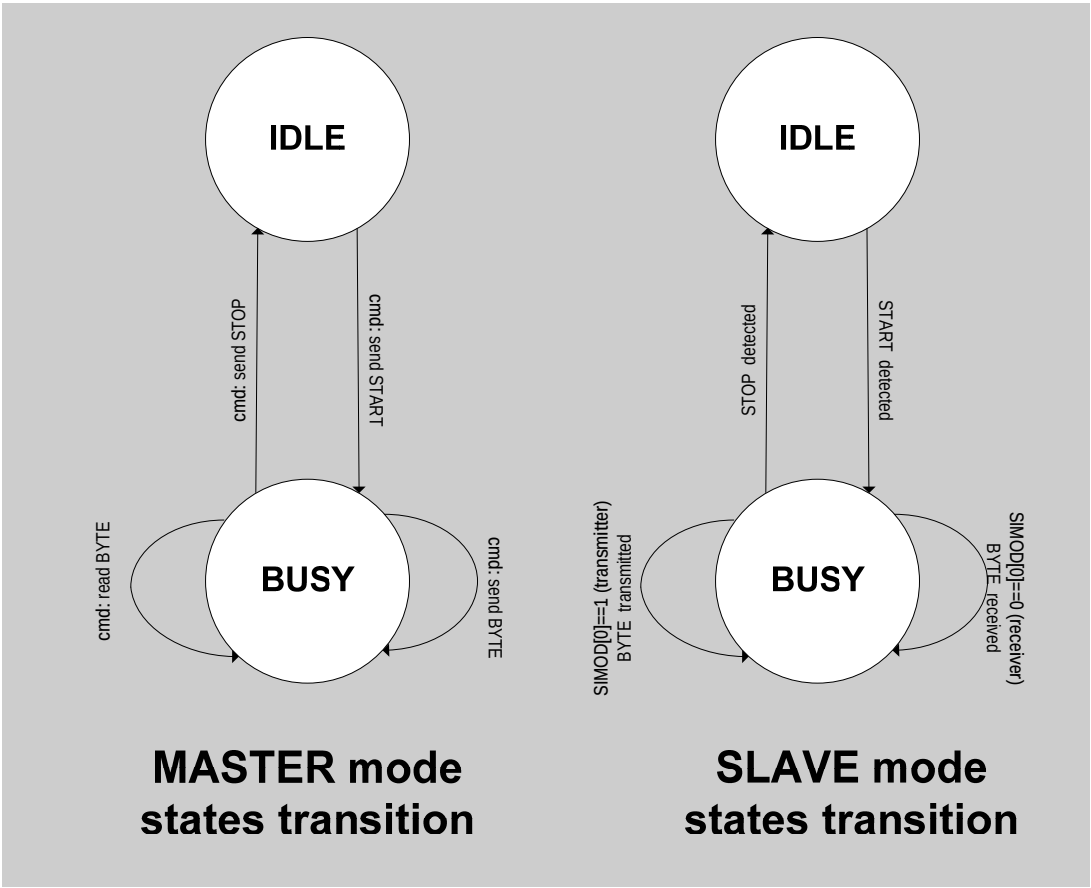
说明：当 ENBUZ 被清除/维持在 0 时，BUZTGP[11:0]保持原设定值，内建的 12-bit Counter 被清除为 000H;此时 P1.6 也不会有 BUZZER 的输出，而是 维持 GPIO 的功能；在 ENBUZ 被设定成 1 的情况下，内建的 12-bit Counter 保持计数，当该 Counter 计数到 {BUZTGP[11:0] }时，P1.6 就会反转(Toggle)输出，同时该 12-bit 的

Counter 也会自动被清除为 000H。也就是说，在 BUZZER 不需要输出的时候，ENBUZ 应被置 0，P1.6 为 GPIO，请用户谨慎设置 P1.6 口的平时状态。

13 串行接口（SIF）

SC91F731 内建了一个简单的串口通讯接口 Serial Interface(SIF), 其电气特性类似 **I²C** 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 **I²C** Interface 的器件。

SC91F731 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。



13.1 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：P1.5 及 P1.4 当作正常 I/O 使用, 其状态由 P1CFG 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 SC91F731 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： P1CFG1 中的 P15M[1:0] 及 P14M[1:0] 都设成 00b, 并且对 P1.5 及 P1.4 都写 1, 以让 P1.5(SDA) 及 P1.4(SCL) 处于 “weak pull-up” 及 “input” 状态, 符合 “类 I^2C” 的需求。
6	**INVI**	INVI (INVerse Input)，SIF 电平反向输出 0：SC91F731 SIF 所取用的 SDA 直接取用 P1.5 的输入， SCL 直接取用 P1.4 的输入。 1：SC91F731 SIF 用的 SDA 取用 P1.5 的反向，SIF 用的 SCL 取用 P1.4 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**	**ACK 信号位 ACKO：** 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

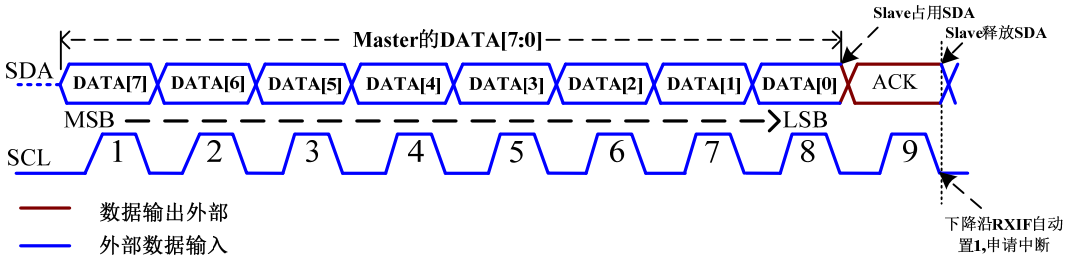
00x：SIF 当做 MASTER 使用, 并且采用 8+1 bits(ACK/NACK) 通讯。 于发出 START 信号后, 发出 STOP 信号前, 强占住 SCL 的控制权 (强推挽)。

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)

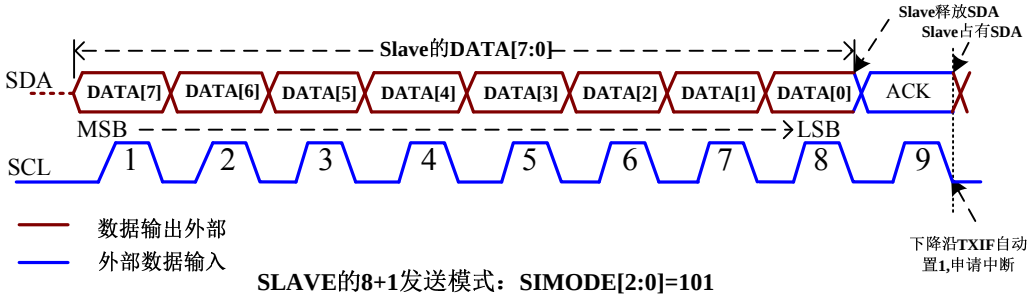
时序图详见“SIFCTL (D5h) SIF 控制寄存器”的说明

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”。



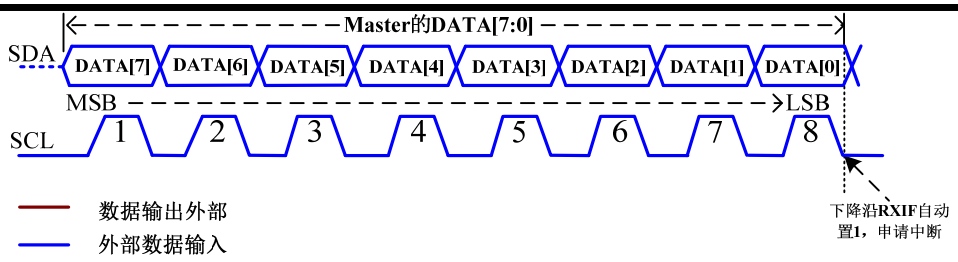
SLAVE的8+1接收模式:SIMODE[2:0]=100

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”。

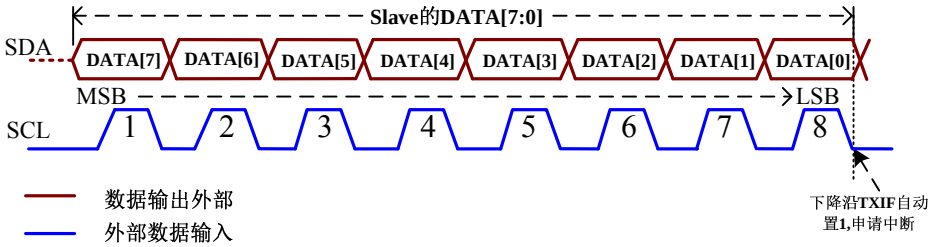


SLAVE的8+0接收模式:SIMODE[2:0]=110

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”。



SLAVE的8+0发送模式：SIMODE[2:0]=111

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

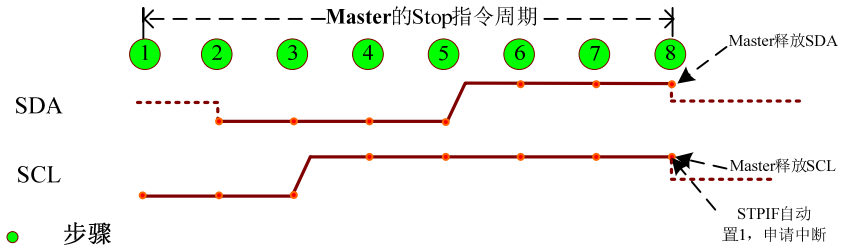
MCMD[1:0] SIF 命令寄存器

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 SC91F731 送出一个 STOP Frame。 此命令只对 SC91F731 处于 Master Mode 并且处于 ”BUSY” state 有用。

在发出 STOP event 后, SC91F731 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” 。

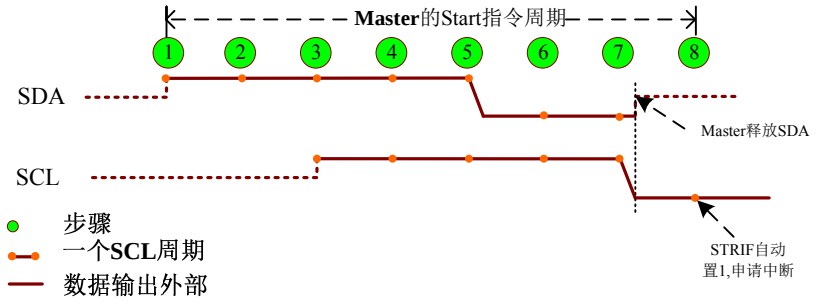


STOP:SIMODE[2]=0（MASTER模式）；MCMD[1:0]=00(STOP)

01：Send START event

对 MCMD[1:0] 写入 01, 代表要 SC91F731 送出一个 START event。此命令只对 SC91F731 处于 MASTER Mode 并且处于 “IDLE” state。

在发出 START event 后, SC91F731 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”。



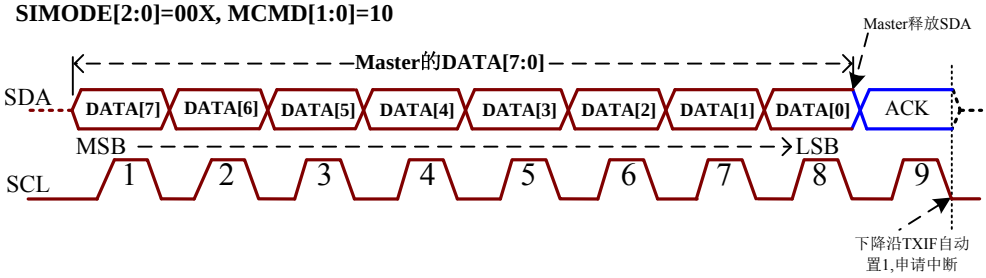
START:SIMODE[2]=0（MASTER模式）；MCMD[1:0]=01(START)

10 ： Send a byte

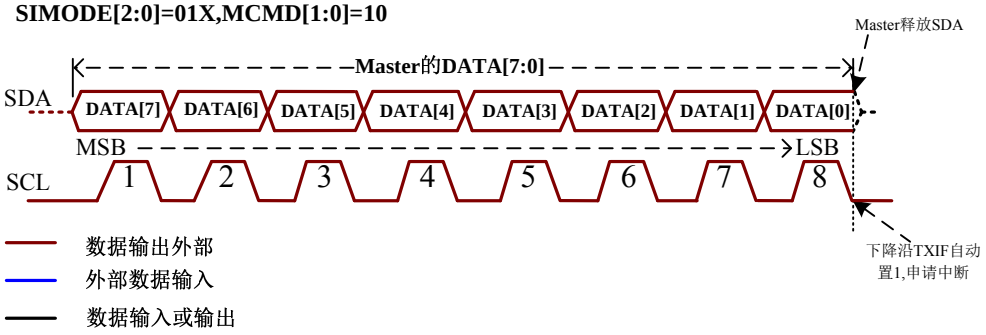
对 MCMD[1:0] 写入 10, 会让 SC91F731 从 SIF 发出一个 BYTE。 依照 SIMOD[1] 的状况, 可能会再接收一个 ACK bit 或是就此结束。

在命令结束后, SC91F731 还是会强占 SCL, 但是会释放 SDA, , 同时也会设定 interrupt flag “TXIF” 为 “1”。

SIMODE[2:0]=00X, MCMD[1:0]=10



SIMODE[2:0]=01X,MCMD[1:0]=10



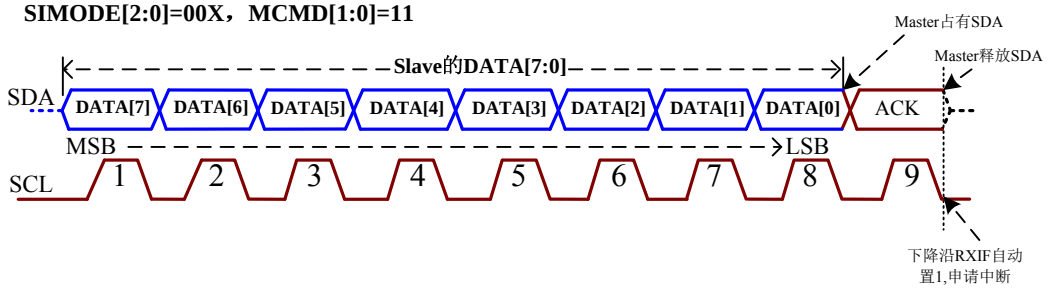
Send a byte:SIMODE[2:0]=00X或01X（MASTER模式）； MCMD[1:0]=10(send a byte)

11 ： Read a byte

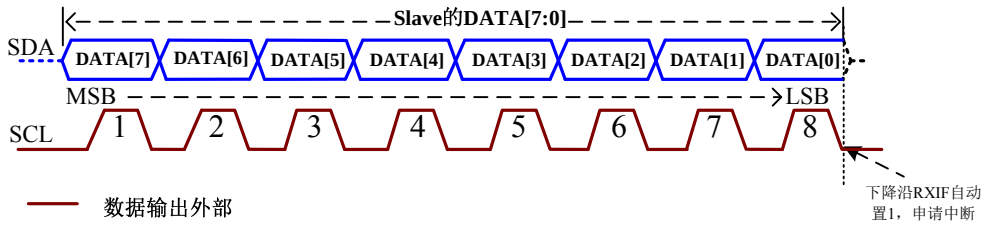
对 MCMD[1:0] 写入 11, 就会让 SC91F731 从 SCL 发出 8 个 clocks, 并且从 SDA 读回一个 BYTE。 依照 SIMOD[1] 的状况, 可能会回复一个 ACK bit 到 SDA 或是就此结束。

在命令结束后, SC91F731 还是会强占 SCL, 但是会释放 SDA, 同时也会设定 interrupt flag “RXIF” 为 “1”。

SIMODE[2:0]=00X，MCMD[1:0]=11



SIMODE[2：0]=01X,MCMD[1：0]=11



Read a byte:SIMODE[2:0]=00X或01X（MASTER模式）; MCMD[1:0]=11(read a byte)

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	**RTNACK**	**ACK 返回信号位 RTNACK** 不管 SC91F731 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	**STPIF**	**SIF STOP 信号中断请求标志 STPIF** 如果 SC91F731 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 SC91F731 处于 SLAVE mode, 在侦测到 I^2C 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	**TXIF**	**SIF 发送完成中断标志 TXIF** 在完成 Send a Byte 命令后, SC91F731 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	**RXIF**	**SIF 接收完成中断标志 RXIF** 在完成 Read a Byte 命令后, SC91F731 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	**STRIF**	**SIF START 信号中断标志 STRIF** 如果 SC91F731 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 SC91F731 处于 SLAVE mode, 在侦测到 I^2C 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

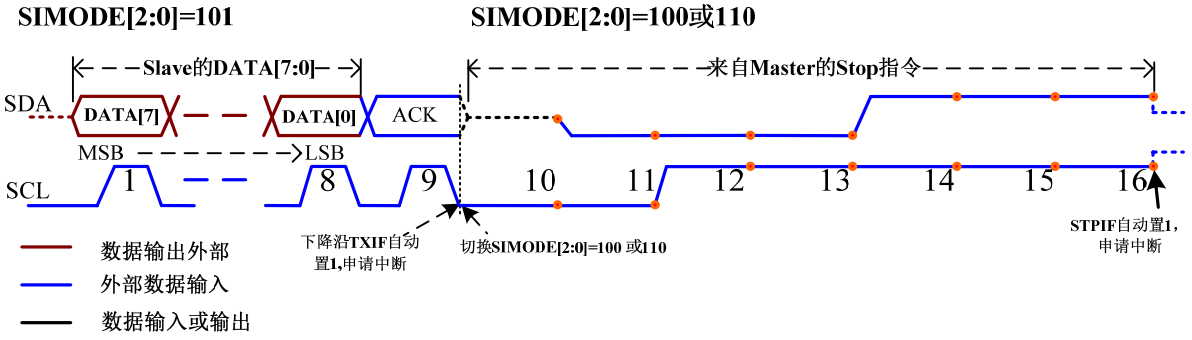
位编号	7	6	5	4	3	2	1	0
符号	**SIFRXD[7:0]**							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
上电初始值	x	x	x	x	x	x	x	x

位编号	位符号	说明
7~0	**SIFRXD**	**SIF 数据位（接收）**

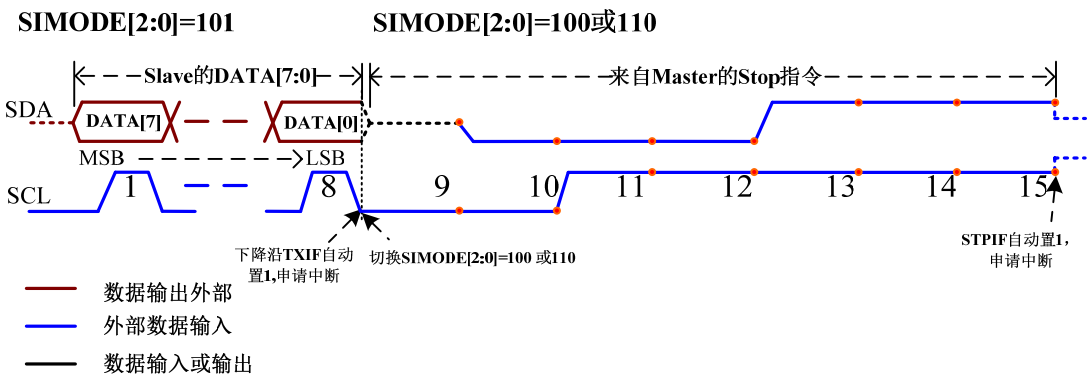
13.2 SIF应用注意事项

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)。



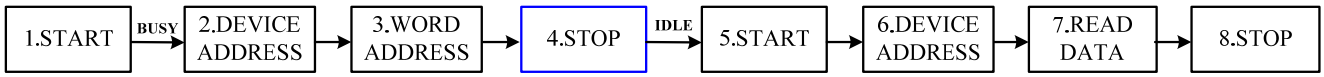
SLAVE模式发送状态转接收状态（with ACK）



SLAVE模式发送状态转接收状态（without ACK）

2. MASTER 模式与标准 I²C 设备进行随机读取操作

SIF 在 MASTER 模式与标准 I²C 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：



14 GP I/O

SC91F731 提供了最多 18 个 GPIO 端口，此 18 个 IO 同其他功能复用。其中 P4.0、P4.1、P1.2~P1.7 的 8 个 IO 具有 35mA 的 Sink 能力，可用来作为 LED 及数码管的 COM 驱动。SC91F731 的 I/O 口和标准 8051 的 I/O 口一样，是带强推挽输出的准双向 IO 口，通过设置 PxCFG 寄存器，有四种 IO 模式可以选择：①准双向 IO 模式②强推挽输出模式③仅输入模式④开漏输出模式。

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况保持该 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, 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。

仅输入结构：高阻态，仅作为口输入使用。

开漏输出结构：IC 内部的上拉电阻断开，需要外接上拉电阻。

14.1 GPIO结构图

1. 准双向模式（Quasi-Bi）

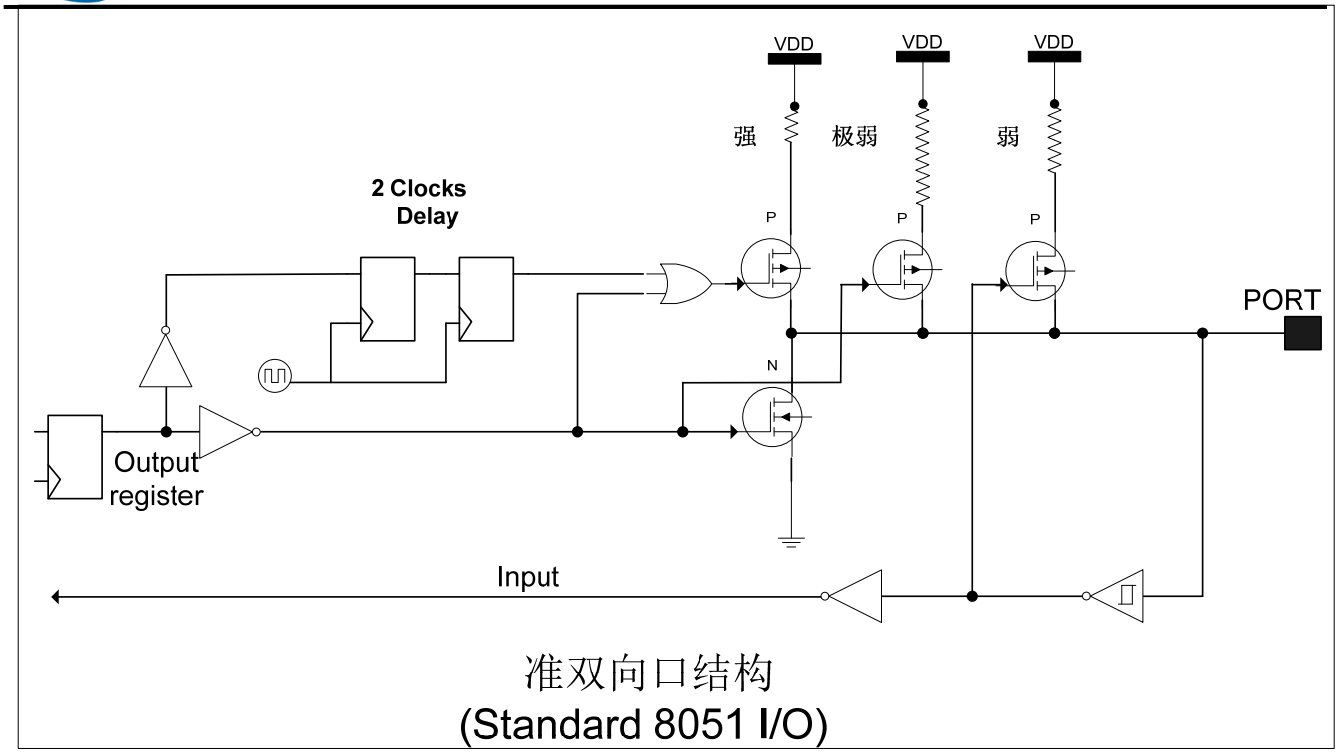
准双向口有 3 个上拉的 MOS 管以适应不同的需要，分别称为“弱（Weak）上拉”、“极弱（Very weak）上拉”和“强（Strong）上拉”。

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：

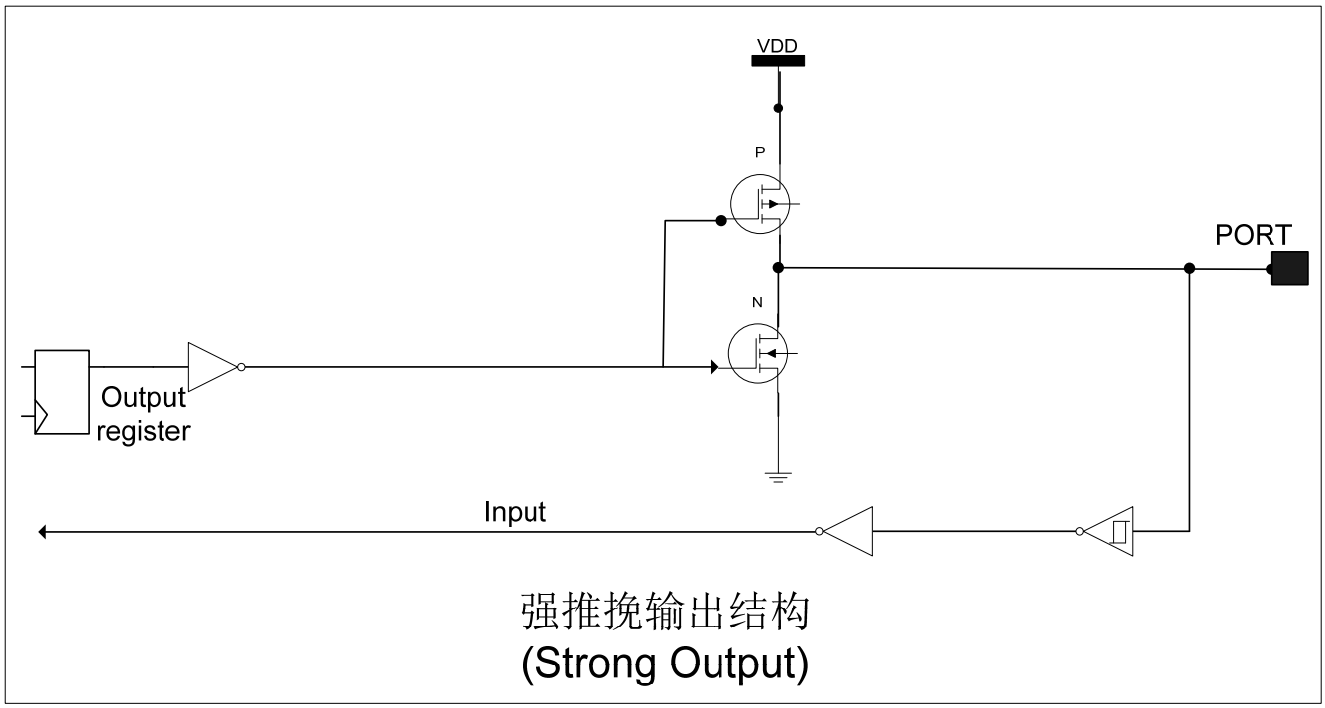


准双向口结构
(Standard 8051 I/O)

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。

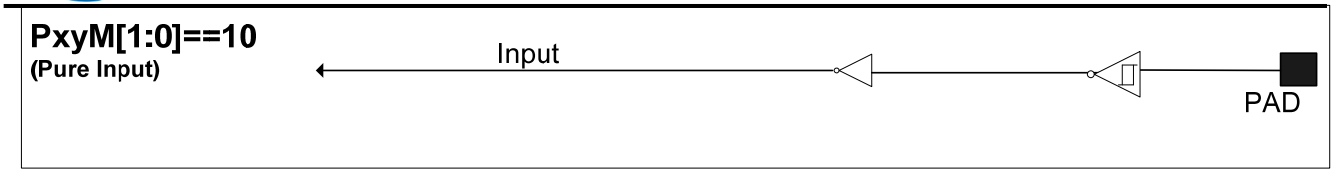
强推挽输出模式的端口结构示意图如下：



强推挽输出结构
(Strong Output)

3，仅输入模式（Input only）高阻

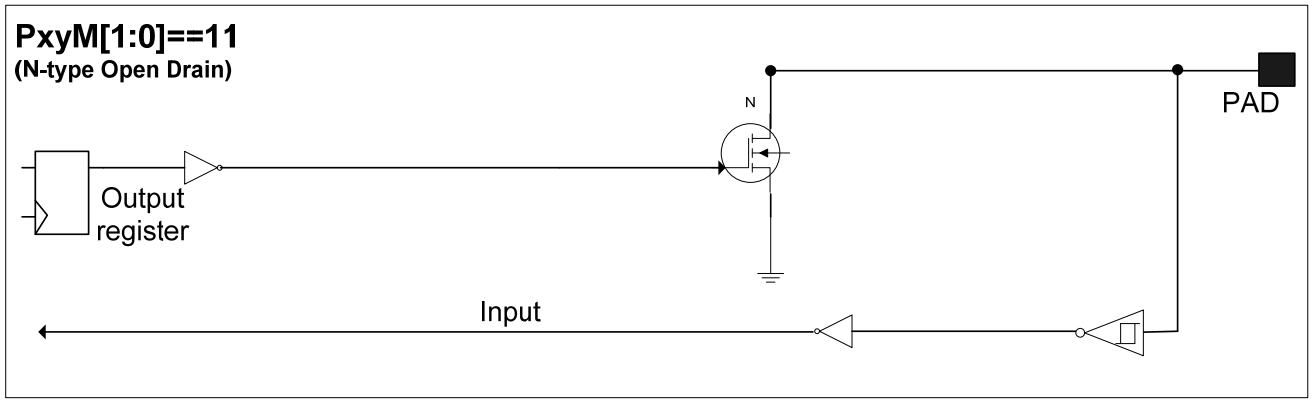
此种模式仅有输入，没有输出能力。仅输入模式的端口结构示意图如下所示：



仅输入结构
(Input only)

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)

14.2 I/O 端口相关寄存器

P1CFG1 (91h) P1 口模式控制寄存器 1(读/写)

位编号	7	6	5	4	3	2	1	0
符号	P17M[1:0]		P16M[1:0]		P15M[1:0]		P14M[1:0]	
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

P1CFG0 (92h) P1 口模式控制寄存器 0(读/写)

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

P2CFG1 (A1h) P2 口模式控制寄存器 1(读/写)

位编号	7	6	5	4	3	2	1	0
符号	-		P26M[1:0]		P25M[1:0]		P24M[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

P2CFG0 (A2h) P2 口模式控制寄存器 0(读/写)

位编号	7	6	5	4	3	2	1	0

符号	P23M[1:0]		P22M[1:0]		-		-	
读/写	读/写	读/写	读/写	读/写	-	-	-	-
上电初始值	0	0	0	0	x	x	x	x

P3CFG1 (B1h) P3 口模式控制寄存器 1(读/写)

位编号	7	6	5	4	3	2	1	0
符号	-		-		P35M[1:0]		P34M[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

P3CFG0 (B2h) P3 口模式控制寄存器 0(读/写)

位编号	7	6	5	4	3	2	1	0
符号	P33M[1:0]		P32M[1:0]		P31M[1:0]		-	
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	x	x

P4CFG0 (C2h) P4 口模式控制寄存器 0(读/写)

位编号	7	6	5	4	3	2	1	0
符号	-	-	-	-	P40M[1:0]		P40M[1:0]	
读/写	-	-	-	-	读/写	读/写	读/写	读/写
上电初始值	x	x	x	x	1	0	1	0

位编号	位符号	说明
7~0	**PxyM1 : PxyM0** **（x=1、2、3、4）** **(y=0~7)**	Pxy 口的模式控制寄存器 PxyM1 和对应的 PxyM0 配对使用，设置该 Pxy 口的模式： PxyM[1:0] 00：准双向口模式 01：强推挽输出模式 10：仅输入模式 11：开漏输出模式

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	P1.5	P1.4	P1.3	P1.2	-	-
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	x	x

P2 (A0h) P2 口数据寄存器(读/写)

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

P3(B0h) P3 口数据寄存器(读/写)

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

P4 (C0h) P4 口数据寄存器(读/写)

位编号	7	6	5	4	3	2	1	0
符号	-	-	-	-	-	-	P4.1	P4.0
读/写	-	-	-	-	-	-	读/写	读/写
上电初始值	x	x	x	x	x	x	1	1

位编号	位符号	说明
7~0	**P1.x** **（x=2~7）**	P1 口锁存寄存器数据
7~0	**P2.x** **(x=2~6)**	P2 口锁存寄存器数据
7~0	**P3.x** **(x=1~5)**	P3 口锁存寄存器数据
1~0	**P4.x** **(x=0,1)**	P4 口锁存寄存器数据

14.3 GPIO的特别说明

P1、P2、P3 口在 IC 内部为完整的 8 个 IO，但在 SC91F731 中有些 IO 没有被引出（P1.0、P1.1、P2.0、P2.1、P2.7、P3.0、P3.6 和 P3.7）。用户根据使用的情况来设置相关 IO 的状态（建议保留初始化的原始状态不变，或者设为 Mode3 纯输入模式，可以实现最低功耗）。

14.4 I/O端口复用

(略)

15. 模数转换ADC

SC91F731 内建一个 10-bit 8 通道的高精度逐次逼近型 ADC。

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；

15.1 ADC相关寄存器

ADCCFG0 (BDh)P2 口 ADC 配置寄存器 （读/写）

位编号	7	6	5	4	3	2	1	0
符号	VREFS[1:0]		-	P24AIN11	P23AIN10	P22AIN9	-	-
读/写	读/写	读/写	-	读/写	读/写	读/写	-	-
上电初始值	n	n	x	0	0	0	x	x

ADCCFG1 (BEh)P3 口 ADC 功能配置寄存器 （读/写）

位编号	7	6	5	4	3	2	1	0
符号	-	-	P35AIN4	P34AIN3	P33AIN2	P32AIN1	P31AIN0	-
读/写	-	-	读/写	读/写	读/写	读/写	读/写	-
上电初始值	x	x	0	0	0	0	0	x

位编号	位符号	说明
7~6	**VREFS[1:0]**	**参考电压选择 (上电时由 Code Option 调入，用户可在程式中修改)** 00：设定 VREF 为 VDD 01：设定 VREF 为 内部准确的 2.4V 10：保留 11：保留
4~0	**P2yAINn**	P2 口切换为 ADC 输入 0：P2.y 为 GPIO

		1：P2.y 作为 ADC 的 Channel n 输入
7~1	**P3yAINn**	P3 口切换为 ADC 输入 0：P3.y 为 GPIO 1：P3.y 作为 ADC 的 Channel n 输入

ADCCR (C5h)ADC 转换控制寄存器

位编号	7	6	5	4	3	2	1	0
符号	ENADC	ADCS	LOWSP	EOC	ADCIS[3:0]			
读/写	读/写	写 1	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	**ENADC**	启动 ADC 的电源 **0**：关闭 ADC 模块电源 1：开启 ADC 模块电源
6	**ADCS**	**ADC 开始触发控制（ADC Start）** 对此 bit 写 “1”, 开始做 1 次 ADC 的转换，也就是只是 ADC 转换的触发信号。此位只可写入 1 有效。
5	**LOWSP**	**ADC 时钟频率选择** 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	**EOC /ADCIF**	**转换完成/ADC 中断请求标志(*End Of Conversion / ADC Interrupt Flag*)** 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。
3~0	**ADCIS[3:0]**	**ADCIS[3:0] ADC 输入通道选择(*ADC Input Selector*)** 0000 := 选用 P3.1 当作 ADC 的 Input. 0001 := 选用 P3.2 当作 ADC 的 Input. 0010 := 选用 P3.3 当作 ADC 的 Input. 0011 := 选用 P3.4 当作 ADC 的 Input. 0100 := 选用 P3.5 当作 ADC 的 Input. 1001 := 选用 P2.2 当作 ADC 的 Input. 1010 := 选用 P2.3 当作 ADC 的 Input. 1011 := 选用 P2.4 当作 ADC 的 Input.

ADCVH (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

ADCVL (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	**ADCV[9:2]**	ADC 转换值的高 8 位数值
2~0	**ADCV[1:0]**	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	ESIF	EPWM	ET1	EX32K	ET0	-
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	x

位编号	位符号	说明
6	**EADC**	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) 中断优先权寄存器(读/写)

位编号	7	6	5	4	3	2	1	0
符号	-	IPADC	IPSIF	IPPWM	IPT1	IPX32K	IPT0	-
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	x

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”

15.2 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 设定对应管脚为 ADC 输入（设定 **P2yAINn/ P3yAINn** 对应位为 ADC 输入，通常 ADC 管脚会先固定）;
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。

16. IAP(IN APPLICATION PROGRAMMING)

SC91F731 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。SC91F731 目前只支持 8M Hz 主频下的 IAP 操作，16MHz 应用下可将此部分空间作为程序空间使用。

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

16.1 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 地指低位	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	**IAPDAT**	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	**PAYTIMES[1:0]**	IAP 写入操作时，CPU Hold Time 时间长度设定 00：设定 CPU HOLD TIME 为 8ms@8MHZ 01：设定 CPU HOLD TIME 为 4ms@8MHZ 10：设定 CPU HOLD TIME 为 2ms@8MHZ 11：保留

		说明：CPU Hold 的仅仅是 PC 指针，其它功能模块正常工作。中断标志会被记录，并在 Hold 结束后响应中断。但对应同一中断所发生的多次中断，仅能响应最后一个中断。
1~0	**CMD[1:0]**	IAP 写入操作命令 00 := (保留) 01 := (保留) 10 := 写入 11 := (保留)

16.2 IAP操作流程

SC91F731 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[12:8], IAPADR[7:0]}（准备好 IAP 操作的目标地址，IAPADR[12:8]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 10，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 MOVC 指令，用户可以访问此部分 DATA 的内容。

16.3 IAP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：

```
#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=0x06;           //执行 IAP 写入操作, 4ms@8M
_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
```

17. 电气特性

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	任一管脚输入/输出电压	-0.3	VDD+0.3	V
TA	工作环境温度	-40	85	℃
TSTG	储存温度	-55	125	℃

推荐工作条件

符号	参数	最小值	最大值	UNIT
VDD	工作电压	3.6	5.5	V
TA	工作环境温度	-40	85	℃

直流电气特性（ VDD = 3.6V ~ 5.5V, TA = +25℃,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
电流						
Iop1	工作电流		5	20	mA	IRC=16MHz VDD=5V
Iop2	工作电流		4	20	mA	IRC=8MHz VDD=5V
Ipd	待机电流（Power Down 模式）	-	0.1	1	uA	IRC=16MHz VDD=5V 所有 IO 设置为准双向模式
Imax	VSS、GND 所能承受的最大电流值			100	mA	VDD=5V
IO 口特性						
VIH	输入高电压	0.7VDD	-	VDD+0.5	V	
VIL	输入低电压	-0.5	-	0.35VDD	V	
VIH,RST	输入高电压, RST 脚	2.0		VDD	V	
VIL,RST	输入低电压, RST 脚	-0.2		1.5	V	
IOL1	输入灌电流 P2/P3/P1.0/P1.1		20		mA	VDD=5V Vpin=0.8V
IOL2	输入灌电流 P1.2~P1.7/P4		35		mA	VDD=5V Vpin=0.8V
IOL3	输入灌电流 P2/P3/P1.0/P1.1		10		mA	VDD=5V Vpin=0.4V
IOL4	输入灌电流 P1.2~P1.7/P4		20		mA	VDD=5V Vpin=0.4V
IOH1	输出高电流（准双向口模式）P1/P2/P3/P4		50		uA	VDD=5V Vpin=4.7V
IOH2	输出高电流（强推挽模式）P1/P2/P3/P4		10		mA	VDD=5V Vpin=4.3V
IOH3	输出高电流（强推挽模式）P1/P2/P3		5		mA	VDD=5V Vpin=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	2.4	2.43	V	TA=-40~85℃

交流电气特性（ VDD = 3.6V ~ 5.5V, TA = 25℃, 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

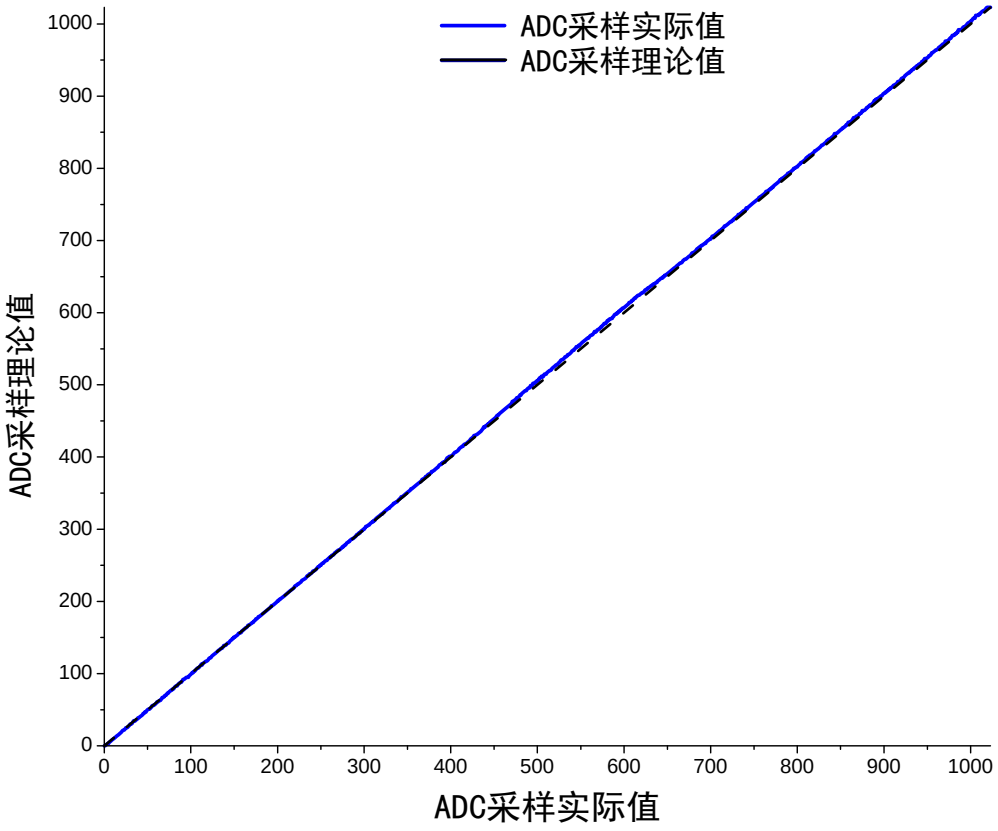
Tosc	振荡器起振时间		5	20	us	IRC=16MHz VDD=5V
T32kosc	32K 振荡器起振时间		0.8	2	s	VDD=5V 特定振荡器
Treset	复位脉冲宽度	64			us	
FIRC	RC 振荡稳定性	15.70	16.00	16.30	MHZ	VDD=5V TA=-40~85℃

ADC 电气特性（TA = 25℃, 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	3.6	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	VIN=5V
Rref	Vref 输入阻抗		13.5		KΩ	
ZAIN	模拟电压源推荐阻抗			10	KΩ	
IADC	ADC 转换电流		1.0		mA	ADC 模块打开 VDD=5V
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	±5	LSB	VDD=5V
TADC	ADC 转换时间	89 个 ADC CLKs				VDD=5V 10bit 精度

说明：

1. **ADC 的偏差如下图基本发生在 2.5V（1/2 Vref）±0.4V 区间，方向及误差值很稳定在 -1~+5LSB；**
2. **可通过程式修正 ADC 所测值，修正后 ADC 精度能在±2LSB 以内，即实际净精度在 9 位以上；**
3. **ADC 曲线的线性度及一致性好；**

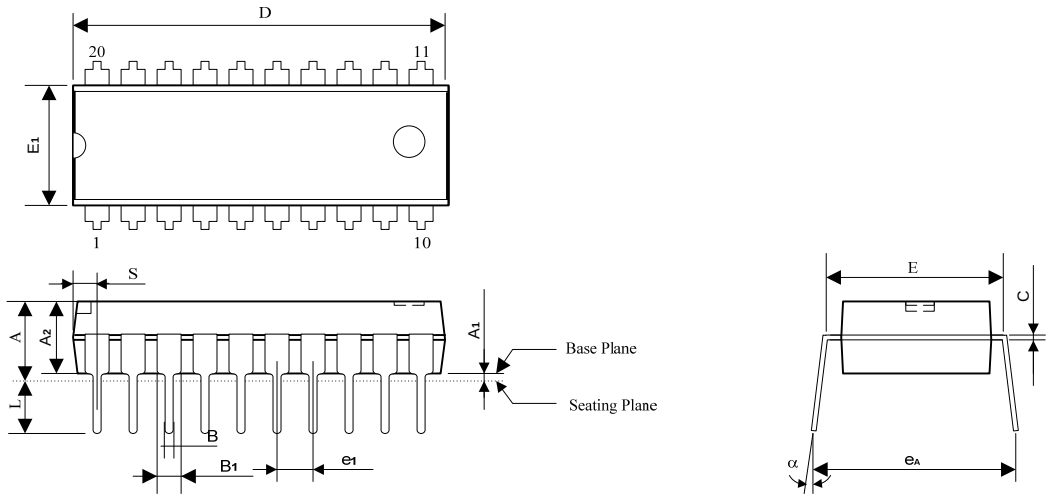

ADC采样实际值
ADC采样理论值
ADC采样理论值
ADC采样实际值
0
100
200
300
400
500
600
700
800
900
1000

18 订购信息

产品编号	封装	包装
SC91F731/020DU	DIP20	管装
SC91F731M/020MU	SOP20	管装

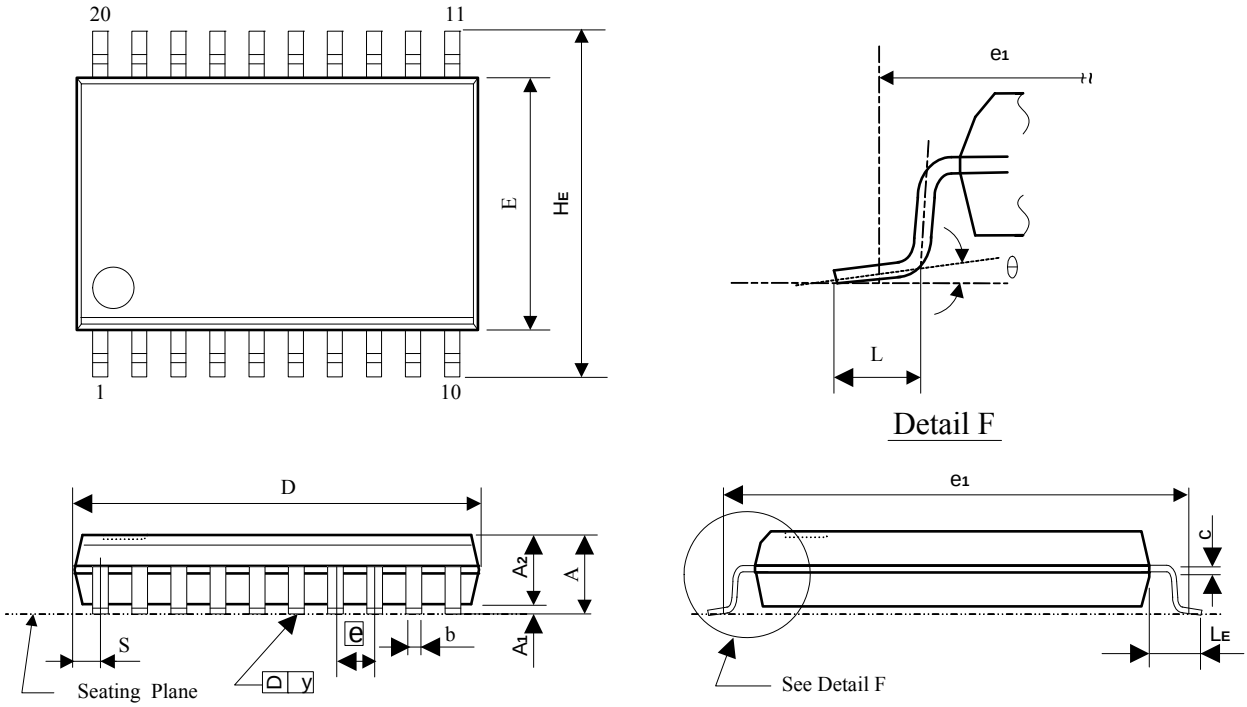
19 封装信息

P-DIP 20L （300mil）外形尺寸　　单位: 毫米



符号	mm(毫米)		
	最小	正常	最大
A	3.60	3.80	4.00
A1	0.51	-	-
A2	3.20	3.30	3.40
B	0.44	-	0.53
B1	1.52(BSC)		
C	0.25	-	0.31
D	26.03	26.23	26.43
E1	6.35	6.55	6.75
e1	2.54(BSC)		
L	3.00	-	-
E	7.62(BSC)		
eA	7.62	-	9.30

SOP 20L(300mil) 外形尺寸　　单位:毫米



符号	mm(毫米)		
	最小	正常	最大
A	2.465	2.515	2.565
A1	0.100	0.150	0.200
A2	2.100	2.300	2.500
b	0.356	0.406	0.456
C	0.254(BSC)		
D	12.500	12.700	12.900
E	7.400	7.450	7.500
HE	10.206	10.306	10.406
e	1.27(BSC)		
L	0.800	0.864	0.900
LE	1.303	1.403	1.503
θ	0°	-	10°
S	0.660(BSC)		

20 规格更改记录

版本	记录	日期
V1.3	**修改软件复位功能描述** **修改 DIP20 及 SOP20 封装外形尺寸**	2012 年 6 月
V1.2	**修正 BUZZER 部分的错误** **修改 SIF 部分说明** **开放系统时钟 8M Hz 应用（包含 IAP 操作）** **禁止 16M Hz IAP 操作** **开放 IRC、2.4V 精确值** **修改 IAP 范例程序**	2012 年 3 月
V1.1	**增加 P1.2~P1.7、P4 口的大电流驱动** **修改 32K 定时中断的时间设定** **修改 IAP 范例程序**	2011 年 10 月
V1.0	**初版**	2011 年 5 月