

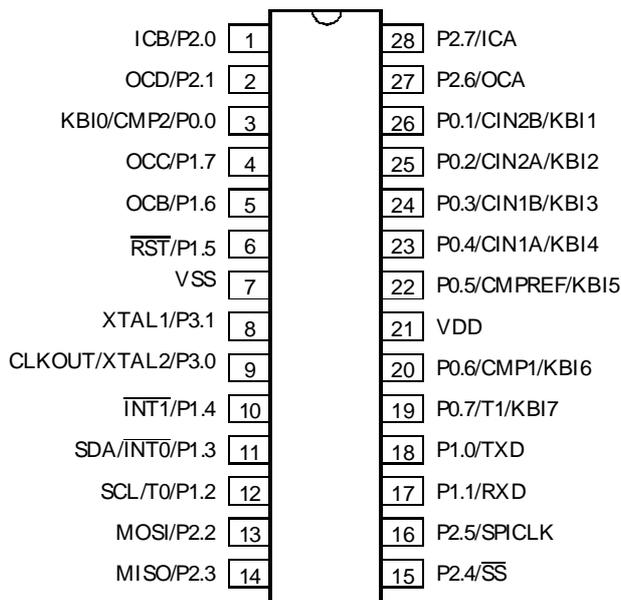
## P89LPC932 Flash 单片机使用指南

### 概述

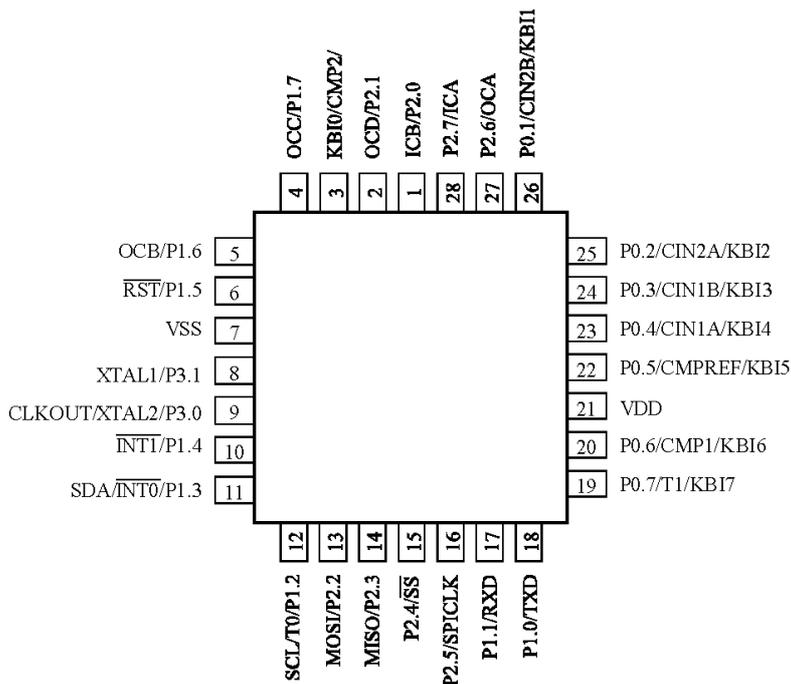
LPC932 是一款单片封装的微控制器，适合于许多要求高集成度、低成本的情况。可以满足多方面的性能要求。LPC932 采用了高性能的处理器结构，指令执行时间只需 2 到 4 个时钟周期。6 倍于标准 80C51 器件。LPC932 集成了许多系统级的功能，这样可大大减少元件的数目、电路板面积以及系统的成本。

### 管脚配置

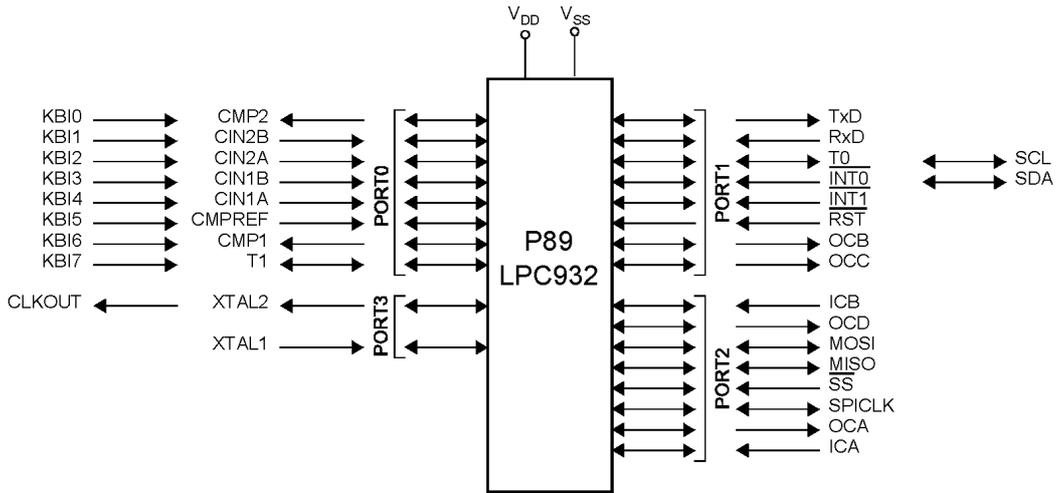
#### 28 脚 TSSOP 封装



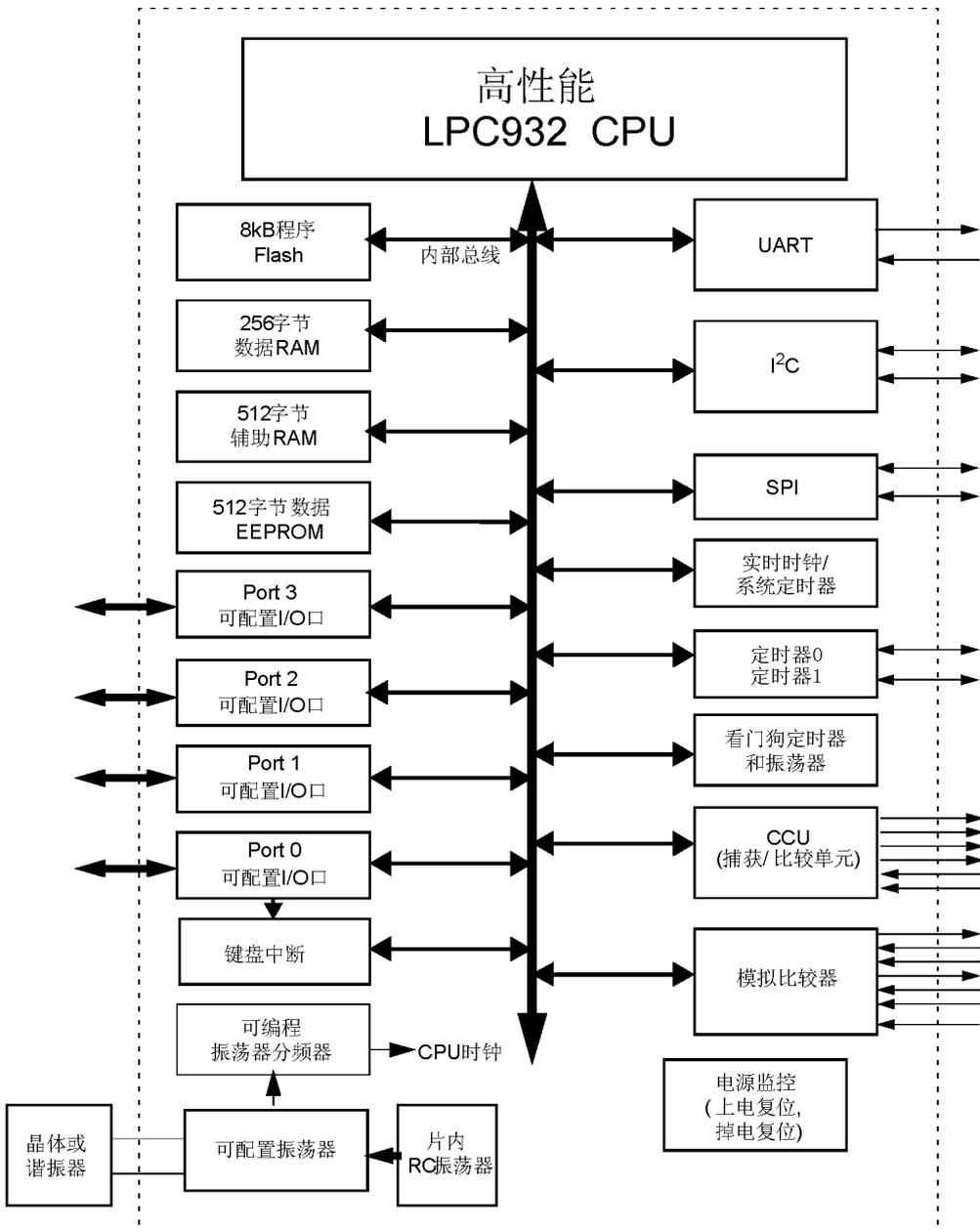
#### 28 脚 PLCC 封装



逻辑符号



功能框图



管脚描述

符号	管脚号 (28脚)	类型	名称及功能描述
P0.0~P0.7	3,26,25,24,23,22,20,19	I/O	<b>PORT0</b> : P0 是一个可由用户定义输出类型的 8 位 I/O 口, 在上电复位时, P0 锁存器配置为内部上拉禁止的仅为输入模式。P0 口由口配置寄存器设定为输出或输入模式, 每一个管脚均可单独设定。详细请参考 I/O 口配置和 DC 电气特性部分。P0 口具有键盘输入中断功能。 所有管脚都具有施密特触发输入。 P0 口还可提供如下特殊功能:
	3	I/O O I	<b>P0.0</b> P0 口位 0 <b>CMP2</b> 比较器 2 输出 <b>KBI0</b> 键盘输入 0
	26	I/O I I	<b>P0.1</b> P0 口位 1 <b>CIN2B</b> 比较器 2 正向输入 B <b>KBI1</b> 键盘输入 1
	25	I/O I I	<b>P0.2</b> P0 口位 2 <b>CIN2A</b> 比较器 2 正向输入 A <b>KBI2</b> 键盘输入 2
	24	I/O I I	<b>P0.3</b> P0 口位 3 <b>CIN1B</b> 比较器 1 正向输入 B <b>KBI3</b> 键盘输入 3
	23	I/O I I	<b>P0.4</b> P0 口位 4 <b>CIN1A</b> 比较器 1 正向输入 A <b>KBI4</b> 键盘输入 4
	22	I/O I I	<b>P0.5</b> P0 口位 5 <b>CMPREF</b> 比较器参考 (反向) 输入 <b>KBI5</b> 键盘输入 5
	20	I/O I I	<b>P0.6</b> P0 口位 6 <b>CMP1</b> 比较器 1 输出 <b>KBI6</b> 键盘输入 6
	19	I/O I/O I	<b>P0.7</b> P0 口位 7 <b>T1</b> 定时/计数器 1 外部计数输入或溢出输出 <b>KBI7</b> 键盘输入 7
P1.0~P1.7	18,17,12,11,10,6,5,4	I/O (P1.0- P1.4, P1.6- P1.7) I (P1.5)	<b>PORT1</b> : 除了下面说明的三个管脚外, P1 是一个可由用户定义输出类型的 8 位 I/O 口。在上电复位时, P1 锁存器配置为内部上拉禁止的仅为输入模式。P1 口由口配置寄存器设定为输出或输入模式, 每一位均可单独设定。详细请参考 I/O 口配置和 DC 电气特性部分。P1.2-P1.3 作为输出时为开漏。P1.5 为仅为输入模式。所有管脚都具有施密特触发输入。 P1 口还可提供如下特殊功能:
	18	I/O O	<b>P1.0</b> P1 口位 0 <b>TxD</b> 串行口输出
	17	I/O I	<b>P1.1</b> P1 口位 1 <b>RxD</b> 串行口输入
	12	I/O I/O I/O	<b>P1.2</b> P1 口位 2 <b>T0</b> 定时/计数器 0 外部计数输入或溢出输出 (作为输出时为开漏) <b>SCL</b> I <sup>2</sup> C 串行时钟输入/输出
	11	I I I/O	<b>P1.3</b> P1 口位 3 (作为输出时为开漏) <b>INT0</b> 外部中断 0 输入 <b>SDA</b> I <sup>2</sup> C 串行数据输入/输出
	10	I I	<b>P1.4</b> P1 口位 4 <b>INT1</b> 外部中断 1 输入。
	6	I I	<b>P1.5</b> P1 口位 5 (仅为输入) <b>RST</b> 上电时作为外部复位输入(通过 UCFG1 选择)。作为复位管脚时, 输入的低电平会使芯片复位, I/O 口和外围功能进入默认状态, 处理器从地址 0 开始执行。另外该管脚还可用于在上电时强制进入 ISP 模式。
	5	I/O I	<b>P1.6</b> P1 口位 6 <b>OCB</b> 输出比较 B

	4	I/O I	<b>P1.7</b> P1 口位 7 <b>OCC</b> 输出比较 C
P2.0~P2.7	1,2,13, 14,15, 16,27, 28	I/O	<b>PORT2:</b> P2 是一个可由用户定义输出类型的 8 位 I/O 口, 在上电复位时, P2 锁存器配置为内部上拉禁止的仅为输入模式。P2 口由口配置寄存器设定为输出或输入模式, 每一位均可单独设定。详细请参考 I/O 口配置和 DC 特性部分。该端口在 20 脚封装中是不存在的, 并自动配置为输出以降低功耗。不要使能这些管脚的特殊功能。 所有管脚都具有施密特触发输入。 P2 口还可提供如下特殊功能:
	1	I/O I	<b>P2.0</b> P2 口位 0 <b>ICB</b> 输入捕获 B
	2	I/O O	<b>P2.1</b> P2 口位 1 <b>OCD</b> 输出比较 D
	13	I/O I/O	<b>P2.2</b> P2 口位 2 <b>MOSI</b> SPI 主机输出/从机输入。当配置为主机时, 该管脚为输出; 当配置为从机时, 该管脚为输入。
	14	I/O I/O	<b>P2.3</b> P2 口位 3 <b>MISO</b> SPI 主机输入/从机输出。当配置为主机时, 该管脚为输入; 当配置为从机时, 该管脚为输出。
	15	I/O I	<b>P2.4</b> P2 口位 4 <b>SS</b> SPI 从机选择
	16	I/O I/O	<b>P2.5</b> P2 口位 5 <b>SPICLK</b> SPI 时钟。当配置为主机时, 该管脚为输出; 当配置为从机时, 该管脚为输入 (20 脚封装中没有这些功能)
	27	I/O O	<b>P2.6</b> P2 口位 6 <b>OCA</b> 输出比较 A
	28	I/O I	<b>P2.7</b> P2 口位 7 <b>ICA</b> 输入捕获 A
P3.0~P3.1	9,8	I/O	<b>PORT3:</b> P3 是一个可由用户定义输出类型的 2 位 I/O 口, 在上电复位时, P3 锁存器配置为内部上拉禁止的仅为输入模式。P3 口由口配置寄存器设定为输出或输入模式, 每一个管脚均可单独设定。详细请参考 I/O 口配置和 DC 电气特性部分。 所有管脚都具有施密特触发输入。 P0 口还可提供如下特殊功能:
	9	I/O O O	<b>P3.0</b> P3 口位 0 <b>XTAL2</b> 振荡放大器输出 (通过 Flash 配置为晶体振荡器时) <b>CLKOUT</b> 通过使能 SFR 位 (ENCLK-TRIM.6) 将 CPU 时钟 2 分频后输出 (当 CPU 时钟为内部 RC 振荡器、看门狗振荡器或外部时钟输入时可用, 但当 XTAL1/XTAL2 用于产生实时时钟/系统定时器的时钟源时除外)。
	8	I/O I	<b>P3.1</b> P3 口位 1 <b>XTAL1</b> 振荡电路和内部时钟发生器电路输入 (通过 Flash 选择)。如果使用内部 RC 振荡器或看门狗振荡器作为 CPU 时钟源并且 XTAL1/XTAL2 不用于产生实时时钟/系统定时器的时钟时, 可作为 I/O 口使用。
V <sub>SS</sub>	7	I	地: 0V 参考点
V <sub>DD</sub>	21	I	电源: 正常操作模式、空闲模式和掉电模式时的电源。

### 特殊功能寄存器

注: 对特殊功能寄存器的访问必须遵循以下方式:

1. 用户不要试图访问任何未经定义的 SFR 地址。
2. 对任何已定义的 SFR 的访问必须符合 SFR 的功能。
3. 标注为 ‘-’, ‘0’ 或 ‘1’ 的 SFR 位只能以如下方式读或写:
  - ‘-’ 必须写入 0, 但当读出时不返回任何确定的值 (即使向其写入 0)。这是一个保留位, 作为将来功能扩展之用。
  - ‘0’ 必须写入 0, 并且当读出时返回 0。
  - ‘1’ 必须写入 1, 并且当读出时返回 1。

特殊功能寄存器

名称	定义	地址	位功能和位地址								复位值
			E7	E6	E5	E4	E3	E2	E1	E0	
ACC*	累加器	E0H									00H
AUXR1#	辅助功能寄存器	A2H	CLKLP	EBRR	ENT1	ENT0	SRST	0	-	DPS	00H <sup>1</sup>
B*	B 寄存器	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
BRGR0#§	波特率发生器低字节	BEH									
BRGR1#§	波特率发生器高字节	BFH									
BRGCON#	波特率发生器控制	BDH	-	-	-	-	-	-	SBRGS	BRGEN	00H <sup>6</sup>
CCCR A#	捕获比较 A 控制	EAH	ICECA2	ICECA1	ICECA0	ICESA	ICNFA	FCOA	OCMA1	OCMA0	00H
CCCR B#	捕获比较 B 控制	EBH	ICECB2	ICECB1	ICECB0	ICESB	ICNFB	FCOB	OCMB1	OCMB0	00H
CCCR C#	捕获比较 C 控制	ECH	-	-	-	-	-	FCOC	OCMC1	OCMC0	00H
CCCR D#	捕获比较 D 控制	EDH	-	-	-	-	-	FCOD	OCMD1	OCMD0	00H
CMP1#	比较器 1 控制	ACH	-	-	CE1	CP1	CN1	OE1	CO1	CMF1	00H <sup>1</sup>
CMP2#	比较器 2 控制	ADH	-	-	CE2	CP2	CN2	OE2	CO2	CMF2	00H <sup>1</sup>
DEECON#	数据 EEPROM 控制	F1H	EEIF	HVERR	ECTL1	ECTL0	-	-	-	EADR8	0EH
DEEDAT#	数据 EEPROM 数据	F2H									00H
DEEADR#	数据 EEPROM 地址	F3H									00H
DIVM#	CPU 时钟分频控制	95H									00H
DPTR	数据指针 (双字节)										
DPH	指针高字节	83H									00H
DPL	指针低字节	82H									00H
FMADR H#	编程 Flash 地址高字节	E7H									00H
FMADR L#	编程 Flash 地址低字节	E6H									00H
FMCON	编程 Flash 控制 (读)	E4H	BUSY	-	-	-	HVA	HVE	SV	OI	70H
FMDATA#	编程 Flash 控制 (写)		FMCMD.7	FMCMD.6	FMCMD.5	FMCMD.4	FMCMD.3	FMCMD.2	FMCMD.1	FMCMD.0	
FMDATA#	编程 Flash 数据	E5H									00H
I2ADR#	I <sup>2</sup> C 从地址寄存器	DBH	I2ADR.6	I2ADR.5	I2ADR.4	I2ADR.3	I2ADR.2	I2ADR.1	I2ADR.0	GC	00H
I2CON*#	I <sup>2</sup> C 控制寄存器	D8H	DF	DE	DD	DC	DB	DA	D9	D8	
I2DAT#	I <sup>2</sup> C 数据寄存器	DAH	-	I2EN	STA	STO	SI	AA	-	CRSEL	00H
I2SCL H#	串行时钟发生器/SCL 占空比寄存器高字节	DDH									00H
I2SCL L#	串行时钟发生器/SCL 占空比寄存器低字节	DCH									00H
I2STAT#	I <sup>2</sup> C 状态寄存器	D9H	STA.4	STA.3	STA.2	STA.1	STA.0	0	0	0	F8H
ICRAH#	输入捕获 A 高字节	ABH									00H
ICRAL#	输入捕获 A 低字节	AAH									00H
ICRBH#	输入捕获 B 高字节	AFH									00H
ICRBL#	输入捕获 B 低字节	AEH									00H
IEN0*	中断使能 0	A8H	AF	AE	AD	AC	AB	AA	A9	A8	
IEN1*#	中断使能 1	E8H	EA	EWDRT	EBO	ES/ESR	ET1	EX1	ET0	EX0	00H
IP0*	中断优先级 0	B8H	EF	EE	ED	EC	EB	EA	E9	E8	
IP0H#	中断优先级 0 高字节	B7H	EIEE	EST	-	ECCU	ESPI	EC	EKBI	EI2C	00H <sup>1</sup>
IP1*#	中断优先级 1	F8H	BF	BE	BD	BC	BBB	BA	B9	B8	
IP1H#	中断优先级 1 高字节	F7H	-	PWDRT	PBO	PS/PSR	PT1	PX1	PT0	PX0	00H <sup>1</sup>
KBCON#	键盘中断	94H	-	PWDRTH	PBOH	PSH/PSRH	PT1H	PX1H	PT0H	PX0H	00H <sup>1</sup>
KBMASK#	键盘中断屏蔽	86H	FF	FE	FD	FC	FB	FA	F9	F8	
KBPATN#	键盘模式	93H	PIEE	PST	-	PCCU	PSPI	PC	PKBI	PI2C	00H <sup>1</sup>
OCRAH#	输出比较 A 高字节	EFH	PIEEH	PSTH	-	PCCUH	PSPIH	PCH	PKBIH	PI2CH	00H <sup>1</sup>
OCRAL#	输出比较 A 低字节	EEB	-	-	-	-	-	-	PATN_SEL	KBIF	00H <sup>1</sup>
OCRBH#	输出比较 B 高字节	FBH									00H
OCRBL#	输出比较 B 低字节	FAH									00H
OCRCH#	输出比较 C 高字节	FDH									00H
OCRCL#	输出比较 C 低字节	FCH									00H
OCRDH#	输出比较 D 高字节	FFH									00H
OCRDL#	输出比较 D 低字节	FEH									00H
P0*	P0 口	80H	87	86	85	84	83	82	81	80	注 1
P1*	P1 口	90H	T1/KB7	CMP1/KB6	CMPREF/KB5	CIN1A/KB4	CIN1B/KB3	CIN2A/KB2	CIN2B/KB1	CMP2/KB0	
P2*	P2 口	A0H	97	96	95	94	93	92	91	90	注 1
			OCC	OCB	RST	INT1	INT0/SDA	T0/SCL	RxD	TxD	
			A7	A6	A5	A4	A3	A2	A1	A0	
			ICA	OCA	SPICLK	SS	MISO	MOSI	OCD	ICB	注 1

		B7	B6	B5	B4	B3	B2	B1	B0		
P3*	P3 口	-	-	-	-	-	-	XTAL1	XTAL2	注 1	
P0M1#	0 口输出模式选择 1	P0M1.7	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0	FFH	
P0M2#	0 口输出模式选择 2	P0M2.7	P0M2.6	P0M2.5	P0M2.4	P0M2.3	P0M2.2	P0M2.1	P0M2.0	00H	
P1M1#	1 口输出模式选择 1	P1M1.7	P1M1.6	-	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0	FFH <sup>1</sup>	
P1M2#	1 口输出模式选择 2	P1M2.7	P1M2.6	-	P1M2.4	P1M2.3	P1M2.2	P1M2.1	P1M2.0	00H <sup>1</sup>	
P2M1#	2 口输出模式选择 1	P2M1.7	P2M1.6	P2M1.5	P2M1.4	P2M1.3	P2M1.2	P2M1.1	P2M1.0	FFH <sup>1</sup>	
P2M2#	2 口输出模式选择 2	P2M2.7	P2M2.6	P2M2.5	P2M2.4	P2M2.3	P2M2.2	P2M2.1	P2M2.0	00H <sup>1</sup>	
P3M1#	3 口输出模式选择 1	-	-	-	-	-	-	P3M1.1	P3M1.0	03H <sup>1</sup>	
P3M2#	3 口输出模式选择 2	-	-	-	-	-	-	P3M2.1	P3M2.0	00H <sup>1</sup>	
PCON#	电源控制寄存器	SMOD1	SMOD0	BOPD	BOI	GF1	GF0	PMOD1	PMOD0	00H	
PCONA#	电源控制寄存器 A	RTCPD	DEEPD	VCPD		I2PD	SPPD	SPD	CCUPD	00H <sup>1</sup>	
PSW*	程序状态字	D7	D6	D5	D4	D3	D2	D1	D0		
		CY	AC	F0	RS1	RS0	OV	F1	P	00H	
PT0AD#	0 口数字输入禁能	-	-	PT0AD.5	PT0AD.4	PT0AD.3	PT0AD.2	PT0AD.1	-	00H	
RSTSRC#	复位源寄存器	-	-	BOF	POF	R_BK	R_WD	R_SF	R_EX	注 2	
RTCCON#	实时时钟控制	RTCF	RTCS1	RTCS0	-	-	-	ERTC	RTCEN	60H <sup>1,5</sup>	
RTCH#	实时时钟高字节									00H <sup>5</sup>	
RTCL#	实时时钟低字节									00H <sup>5</sup>	
SADDR#	串口地址寄存器									00H	
SADEN#	串口地址使能									00H	
SBUF	串口数据缓冲区									xxH	
SCON*	串行口控制	9F	9E	9D	9C	9B	9A	99	98		
		SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H	
SSTAT#	串行口扩展状态	DBMOD	INTLO	CIDIS	DBISEL	FE	BR	OE	STINT	00H	
SP	堆栈指针									07H	
SPCTL#	SPI 控制寄存器	SSIG-	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	04H	
SPSTAT#	SPI 状态寄存器	SPIF	WCOL	-	-	-	-	-	-	00H	
SPDAT#	SPI 数据寄存器									00H	
TAMOD#	定时器 0/1 附加模式	-	-	-	T1M2	-	-	-	T0M2	00H	
		8F	8E	8D	8C	8B	8A	89	88		
TCON*	定时器 0/1 控制	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H	
TCR20*#	CCU 控制寄存器 1	PLEEN	HLTRN	HLTEN	ALTCDD	ALTAB	TDIR2	TMOD21	TMOD20	00H	
TCR21#	CCU 控制寄存器 2	TCOU2	-	-	-	PLLDV.3	PLLDV.2	PLLDV.1	PLLDV.0	00H	
TH0	定时器 0 高字节									00H	
TH1	定时器 1 高字节									00H	
TH2#	CCU 定时器高字节									00H	
TICR2#	CCU 中断控制	TOIE2	TOCIE2D	TOCIE2C	TOCIE2B	TOCIE2A	-	TICIE2B	TICIE2A	00H	
TIFR2#	CCU 中断标志	TOIF2	TOCF2D	TOCF2C	TOCF2B	TOCF2A	-	TICF2B	TICF2A	00H	
TISE2#	CCU 中断状态编码	-	-	-	-	-	ENCINT.2	ENTINT.1	ENTINT.0	00H	
TL0	定时器 0 低字节									00H	
TL1	定时器 1 低字节									00H	
TL2#	CCU 定时器低字节									00H	
TMOD	定时器 0/1 模式	T1GATE	T1C/T	T1M1	T1M0	T0GATE	T0C/T	T0M1	T0M0	00H	
TOR2H#	CCU 重装高字节	CFH									00H
TOR2L#	CCU 重装低字节	CEH									00H
TPCR2H#	预分频器控制高字节	CBH									00H
TPCR2L#	预分频器控制低字节	CAH	TPCR2L.7	TPCR2L.6	TPCR2L.5	TPCR2L.4	TPCR2L.3	TPCR2L.2	TPCR2L.1	TPCR2L.0	00H
TRIM#	内部振荡调整寄存器	-	ENCLK	TRIM.5	TRIM.4	TRIM.3	TRIM.2	TRIM.1	TRIM.0	注 4	
WDCON#	看门狗控制寄存器	PRE2	PRE1	PRE0	-	-	WDRUN	WDTOF	WDCLK	注 3,5	
WDL#	看门狗装载									FFH	
WFEED1#	看门狗清零 1										
WFEED2#	看门狗清零 2										

注:

\* 特殊寄存器可位寻址。

# 从 80C51 特殊功能寄存器修改而来或新增加的寄存器。

§ 只有当 BRGCON 中的 BRGEN 为 0 时, 才可对 BRGR1 和 BRGR0 进行写操作。当 BRGEN=1 时, 写入其中任意一个, 结果将是不可预知的。

% 标为“-”的位总是为“X”(未知)。不要向这些位写入 1, 因为它们可能在将来扩展用于其它功能。这些位的位值均为 0。

1. 上电复位后所有的 I/O 口都为仅为输入(高阻)状态。

2. RSTSRC 寄存器反映 LPC932 复位的原因。在上电复位时, 所有复位源标志都清零(POF 和 BOF 除外)。上电复位值

为 xx110000。

3. 看门狗复位时 WDCON 复位值为 111001x1b, 即 PRE2-PRE0 全 1, WDRUN=1, WDCLK=1。看门狗复位时, WDTOF=1; 上电复位时, WDTOF=0。其它复位不会影响 WDTOF。
4. 上电复位时, TRIM 寄存器初始化为出厂时的配置。
5. 上电复位是唯一影响这些特殊功能寄存器的复位。

## 功能描述

### 增强型 CPU

P89LPC932 采用增强型 80C51 CPU, 其运行速度是标准 80C51 的 6 倍。一个机器周期由 2 个 CPU 时钟周期组成, 大多数指令执行时间为 1 到 2 个机器周期。

## 时钟

### 时钟定义

LPC932 的几个内部时钟定义如下:

- OSCCLK—输入到 DIVM 分频器的时钟。OSCCLK 可选择 4 个时钟源之一 (见图 3), 也可分频到较低的频率 (见 DIVM 寄存器一节)。注: fosc 定义为 OSSCLK 频率
- CCLK—CPU 时钟; 时钟分频器的输出。每个机器周期包含 2 个 CCLK 周期, 大多数指令执行时间为 1 到 2 个机器周期 (2 到 4 个 CCLK 周期)。
- RCCLK—内部 7.373MHz RC 振荡器输出。
- PCLK—用于不同外围器件的时钟, 为 CCLK/2。

### 振荡器时钟(OSCCLK)

LPC932 提供几个可由用户选择的振荡器选项来产生 CPU 时钟。这样就满足了从高精度到低成本的不同需求。这些选项在对 Flash 进行编程时配置, 包括片内看门狗振荡器、片内 RC 振荡器、使用外部晶振的振荡器或外部时钟源。晶振可选择低、中或高频晶振, 频率范围为 20KHz 到 12MHz。

#### 低频振荡器选项

此选项支持 20KHz~100KHz 的外部晶振, 同时也支持陶瓷谐振器。

#### 中频振荡器选项

此选项支持 100KHz~4MHz 的外部晶振, 同时也支持陶瓷谐振器。

#### 高频振荡器选项

此选项支持 4MHz~12MHz 外部晶振, 同时也支持陶瓷谐振器。

### 时钟输出

LPC932 支持可由用户选择的时钟输出功能。当不使用晶振时, 可从 XTAL2/CLKOUT 输出时钟。实现该功能的前提是已选择另外的时钟源 (片内 RC 振荡器、看门狗振荡器或 X1 脚输入的外部时钟) 并且未使用晶振作为实时时钟的时钟源。这样可使外部器件与 LPC932 同步。时钟输出的使能通过置位 TRIM 寄存器中的 ENCLK 位实现。

该时钟输出的频率为 CCLK/2。如果在空闲模式中不需要输出时钟, 那么可在进入空闲模式之前将该功能关闭以降低功耗。注: 在复位时, TRIM 寄存器初始化为出厂的预编程值。

当用户置位或清零 ENCLK 位时, 应当保持 TRIM[5:0]的内容不变。可以通过读出 TRIM 的内容 (例如读入 ACC), 然后修改 TRIM.6 后再写入 TRIM。另外也可使用 ANL 或 ORL 指令置位或清零 ENCLK(TRIM.6)。

### 片内 RC 振荡器选项

LPC932 具有一个 6 位 TRIM 寄存器，可对 RC 振荡器的频率进行调整。在复位时，TRIM 的值初始化为出厂时预编程值以将振荡器频率调整为 7.373MHz±2.5%。用户程序可修改 TRIM 寄存器将 RC 振荡器调整为其它频率。

TRIM	地址: 96H
不可位寻址	7 6 5 4 3 2 1 0
复位源: 上电	— ENCLK TRIM.5 TRIM.4 TRIM.3 TRIM.2 TRIM.1 TRIM.0
复位值: 上电复位时,	ENCLK=0, TRIM.5-0 装入出厂编程值
位	符号 功能
TRIM.7	— 保留
TRIM.6	ENCLK 当 ENCLK=1 且未使用晶振时, XTAL2 (P3.0) 上输出频率为 CCLK/2 的时钟。 当 ENCLK=0 时, 无时钟输出。
TRIM.5-0	RC 振荡器频率调整值
注: 在复位时, TRIM 寄存器初始化为出厂的预编程值。当用户需要置位或清零 ENCLK 位时, 应当保持 TRIM[5:0]的内容不变。可以通过读出 TRIM 的内容 (例如读入 ACC), 然后修改 TRIM.6 后再写入 TRIM。另外也可使用 ANL 或 ORL 指令置位或清零 ENCLK(TRIM.6)。	

图 1 片内 RC 振荡器 TRIM 寄存器

### 看门狗振荡器选项

看门狗具有一个独立的振荡器，其频率为 400KHz。在不需要使用高频振荡器时，可使用该振荡器降低功耗。

### 外部时钟输入选项

在此配置中，提供 CPU 时钟的外部时钟源从 XTAL1/P3.1 脚输入。频率可从 0Hz 到 12MHz。XTAL2/P3.0 脚可作为标准 I/O 口或者时钟输出。

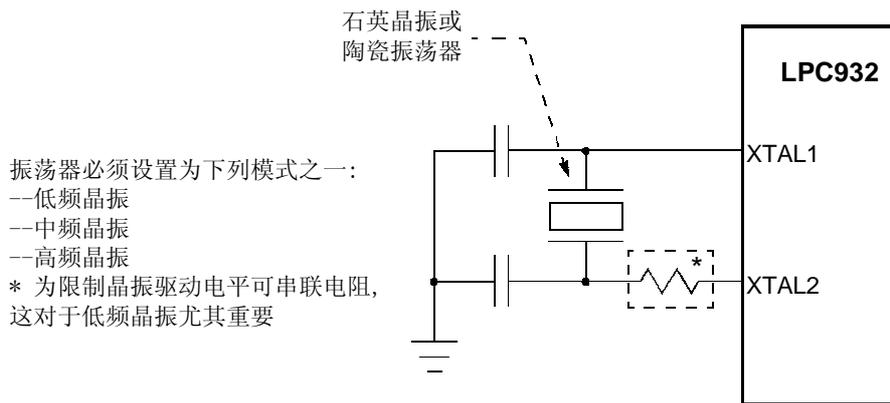


图 2 晶振的使用

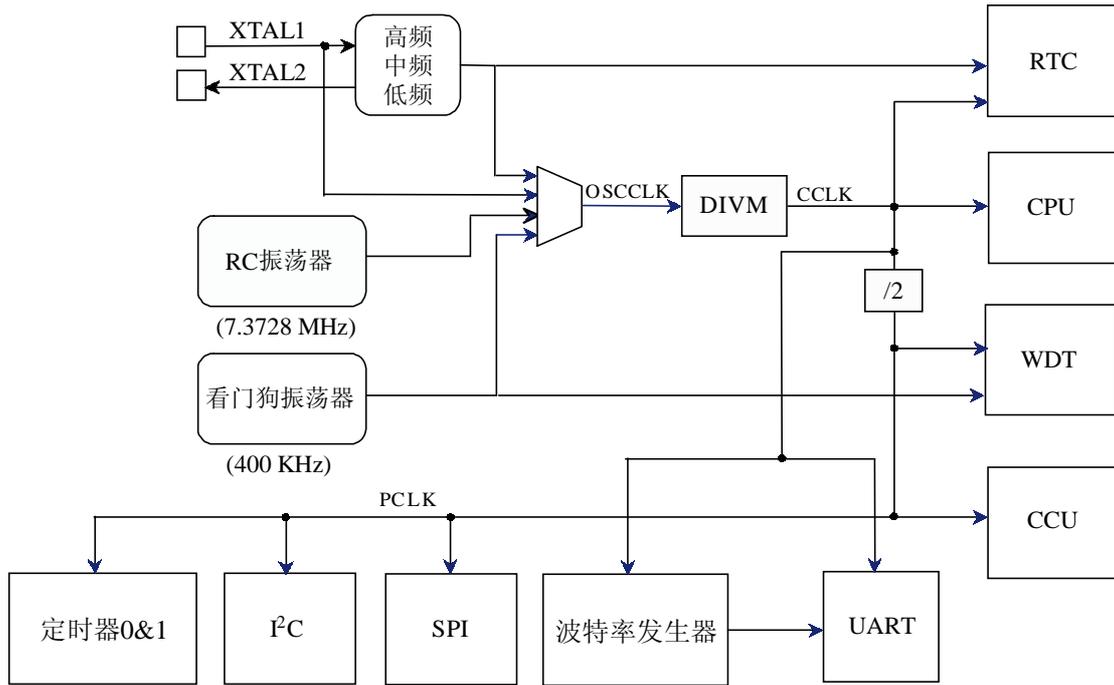


图3 振荡器控制框图

#### 振荡器时钟（OSCCLK）唤醒延迟

LPC932 具有一个内部唤醒定时器，可使时钟延迟直到稳定下来，其延迟时间取决于使用的时钟源。如果时钟源为 3 个晶振选项中的任意一个（低、中或高频），延迟时间为 992 个 OSCCLK 周期。如果时钟源为内部 RC 振荡器、看门狗振荡器或外部时钟，延迟时间为 224 个 OSCCLK 周期。

#### CPU 时钟（CCLK）调整：DIVM 寄存器

OSCCLK 频率可通过配置分频寄存器 DIVM 进行分频来提供 CCLK。OSCCLK 被一个整数所除得到 CCLK 频率。见下面的公式：

$$\text{CCLK 频率} = \text{fosc} / (\text{N} + 1)$$

此处：fosc 为 OSCCLK 频率，N 为 DIVM 值

由于 N 的范围为 0~255，CCLK 频率可从 fosc 到 fosc/256。

此特性可使 CPU 暂时以较低频率工作以降低功耗。通过将时钟分频，程序以较低速度运行时，使 CPU 仍保持对事件响应的能力，而不是只能响应能产生中断的事件（使 CPU 从空闲模式退出的事件）。这常常会得到比空闲模式更低的功耗。和掉电模式相比则少了振荡器起振时间。在程序内 DIVM 的值可随时改变而不会中断程序的运行。

#### 低功耗选择

LPC932 设计的最大工作频率为 12MHz (CCLK)。但是如果 CCLK 为 8MHz 或更低，则可置位 CLKLP 位 (AUXR1.7) 以降低功耗。在任何一次复位后，CLKLP 都为 0 以允许实现最高性能。如果 CCLK 运行在 8MHz 或更低的频率时，该位可以在软件当中置位。

#### 存储器结构

LPC932 的存储器结构如图 4 所示：

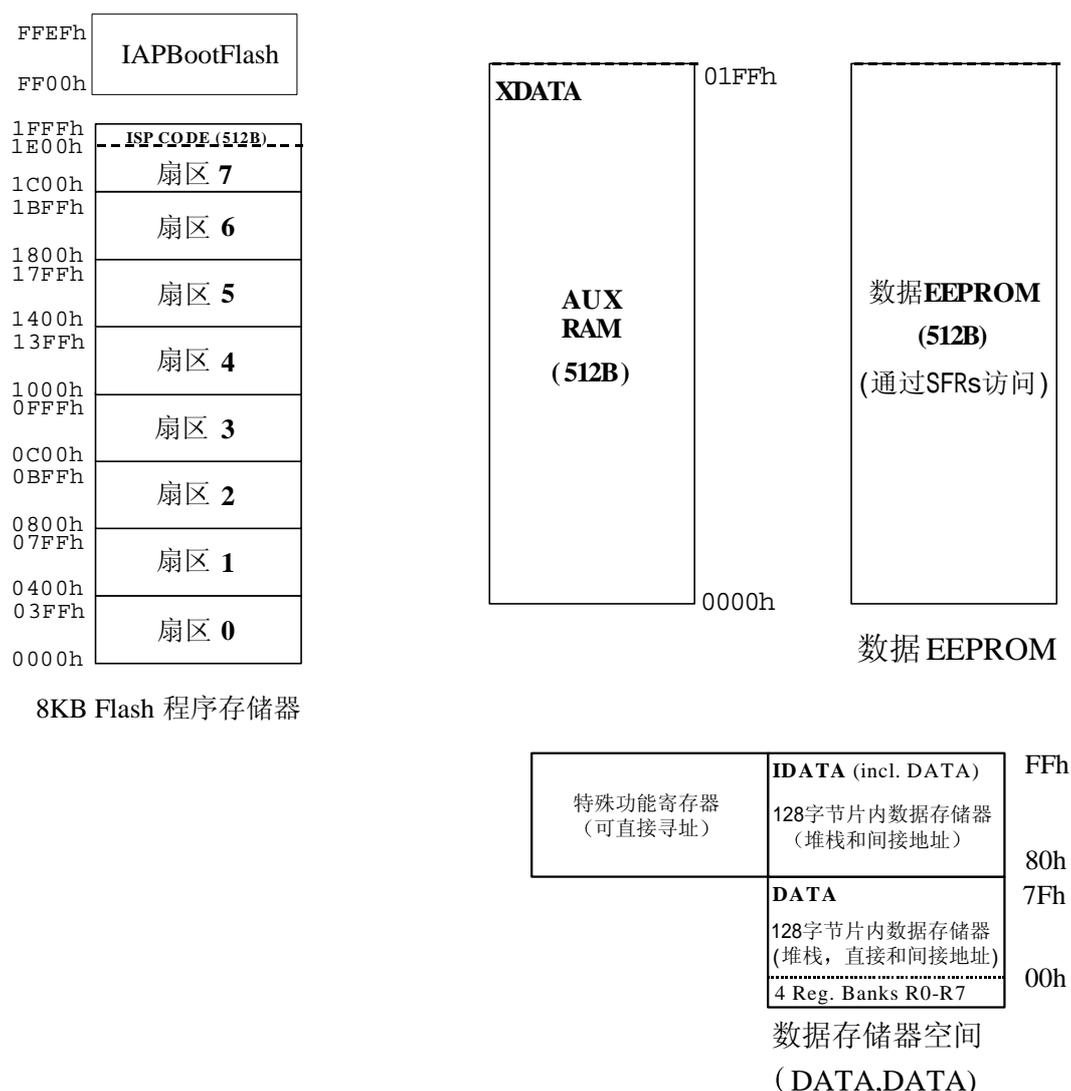


图4 LPC932 存储器结构

LPC932 的不同存储空间如下所示:

- DATA** 128 字节内部数据存储空间 (00h..7Fh)。可使用除 MOVX 和 MOVC 之外的指令直接或间接寻址。此空间可作为全部或部分堆栈空间。
- IDATA** 间接数据。256 字节内部数据存储空间 (00h..FFh)。可使用除 MOVX 和 MOVC 之外的指令进行间接寻址。全部或部分堆栈位于此空间。该区域包括了 DATA 区域和其之上的 128 字节间接 RAM。
- SFR** 特殊功能寄存器。选择的 CPU 寄存器和外设控制及状态控制寄存器, 只能通过直接寻址访问。
- XDATA** 外部数据或附加 RAM。与标准 80C51 64K 字节存储空间完全相同。通过 MOVX 指令使用 DPTR, R0 或 R1 寻址。该空间的全部或部分可在片内实现。LPC932 集成了 512 字节片内 XDATA 存储器。
- CODE** 64K 字节代码存储空间。作为程序执行部分和通过 MOVC 指令访问。LPC932 有 8K 字节片内代码存储器。

LPC932 还具有 512 字节片内数据 EEPROM 存储器, 通过 SFR 进行访问 (详见数据 EEPROM 一节)。

### 数据 RAM 分配

768 字节片内 RAM 分配如下:

表 1 片内数据 RAM

类型	数据 RAM	规格 (字节)
DATA	可直接或间接寻址	128
IDATA	可间接寻址	256
XDATA	附加的片内存储器, 使用 MOVX 指令进行访问	512

### 中断

LPC932 采用 4 中断优先级结构。这为 P89LPC932 的 15 个中断源的处理提供了极大的灵活性。

任何一个中断源均可通过对 IEN0 和 IEN1 中相应的位置位或清零, 实现单独使能或禁能。IEN0 中还包含了一个全局使能位 EA, 它可使能所有的中断。

每个中断源都可被单独设置为四个中断优先级之一, 分别通过清零或置位 IP0, IP0H, IP1, IP1H 中相应位来实现 (00--最低, 11--最高)。一个中断服务程序可响应更高级的中断, 但不能响应同优先级或低级中断。最高级中断服务程序不响应其它任何中断。如果两个不同中断优先级的中断源同时申请中断时, 响应较高优先级的中断申请。

如果两个同优先级的中断源同时申请中断。通过一个内部查询顺序序列确定首先响应哪一个中断请求, 这叫做仲裁队列。注: 仲裁队列只用来处理相同优先级别中断源同时申请中断的情况。表 3 汇集了中断源, 中断标志, 向量地址, 使能位、优先级别、仲裁队列顺序和是否中断可将 CPU 从掉电方式中唤醒。

### 中断优先级结构

有 4 个特殊功能寄存器与中断优先级相关, 它们分别是: IP0, IP0H, IP1, IP1H。每个中断都由两个位 IPx 和 IPxH (x=0, 1) 控制, 这样可将其分配到 4 个优先级中的任意一个。

表 2 中断优先级

优先级位		中断优先级
IPxH	IPx	
0	0	0 级(最低优先级)
0	1	1 级
1	0	2 级
1	1	3 级(最高优先级)

表 3 中断汇总

描述	中断标志位	中断向量	中断使能位	中断优先级	仲裁顺序	掉电唤醒
外部中断 0	IE0	0003H	EX0(IEN0.0)	IP0H.0,IP0.0	1(最高)	√
定时器 0 中断	TF0	000BH	ET0(IEN0.1)	IP0H.1,IP0.1	4	
外部中断 1	IE1	0013H	EX1(IEN0.2)	IP0H.2,IP0.2	7	√
定时器 1 中断	TF1	001BH	ET1(IEN0.3)	IP0H.3,IP0.3	10	
串口 Tx、Rx <sup>1,4</sup>	TI&RI	0023H	ES/ESR (IEN0.4)	IP0H.4,IP0.4	13	√
串口 Rx <sup>1,4</sup>	RI					
掉电检测	BOF	002BH	EBO(IEN0.5)	IP0H.5,IP0.5	2	
看门狗定时器/ 实时时钟	WDOVF/ RTCF	0053H	EWDRT(IEN0.6)	IP0H.6,IP0.6	3	√
I <sup>2</sup> C 中断	SI	0033H	EI2C(IEN1.0)	IP1H.0,IP1.1	5	
KBI 中断	KBIF	003BH	EKBI(IEN1.1)	IP1H.1,IP1.1	8	√
比较器 1/2 中断	CMF1/CMF2	0043H	EC(IEN1.2)	IP1H.2,IP1.2	11	√
SPI 中断	SPIF	004BH	ESPI(IEN1.3)	IP1H.3,IP1.3	14	
捕获/比较单元 <sup>2</sup>	注 2	005BH	ECCU(IEN1.4)	IP1H.4,IP1.4	6	
保留		0063H	(IEN1.5)	IP1H.5,IP0.5	9	√
串口 Tx <sup>3</sup>	TI	006BH	EST(IEN1.6)	IP1H.6,IP1.6	12	
数据 EEPROM 写 完成	EEPROM	0073H	EIEE(IEN1.7)	IP1H.7,IP1.7	15(最低)	

注:

1. SSTAT.5=0 选择串口 Tx 和 Rx 的中断组合；SSTAT.5=1 只选择串口 Rx 中断（Tx 中断将会不同）。
2. CCU 中断具有多个中断源。TIFR2 寄存器中的任何一个中断源都可导致产生 CCU 中断。
3. 当且仅当 SSTAT.5=1 时，该中断作为串口（UART）的 Tx 中断，否则该中断被禁止。
4. 如果 SSTAT.0=1，串口附加的标志位 FE,BR,OE 将导致产生该中断。

### 外部中断输入

LPC932 有 2 个外部中断输入以及键盘中断功能。这两个外部中断输入和标准 80C51 微控制器相同。通过清零或置位 TCON 寄存器中的 IT1 或 IT0，可将外部中断设为电平触发或边沿触发。如果 ITn=0，外部中断 n 通过  $\overline{INTn}$  脚的低电平触发。如果 ITn=1，外部中断 n 为边沿触发。该模式下，对  $\overline{INTn}$  脚连续采样，如果在一个周期为高电平而下一个周期为低电平，中断请求标志 IEn 将置位。这样就产生了一个中断请求。

由于外部中断脚每个机器周期采样一次，输入高或低电平应当保持至少 1 个机器周期以确保能够被正确采样到。如果外部中断为边沿触发，外部中断源应当将中断脚至少保持 1 个机器周期高电平然后至少保持 1 个机器周期低电平。这样就确保了边沿能够被检测到以使 IEn 置位。当调用中断服务程序后，CPU 自动将 IEn 清零。

如果外部中断为电平触发，外部中断源必须一直保持请求有效，直到产生所请求的中断为止。如果中断服务完成后而外部中断仍旧维持，这将产生下一次中断。当中断为电平触发时不必清除中断标志 IEn，因为中断只和输入口电平有关。

如果 LPC932 处于掉电或空闲状态时外部中断使能，中断将唤醒处理器使其恢复运行。详见节电模式部分。

### 外部中断管脚干扰抑制

LPC932 的大多数管脚都有干扰抑制电路，可以滤除窄脉冲的干扰（请参考 AC 特性）。管脚 SDA/ $\overline{INT0}$ /P1.3 和 SCL/T0/P1.2 没有干扰抑制电路。也就是说， $\overline{INT1}$  有干扰抑制电路，而  $\overline{INT0}$  没有干扰抑制电路。

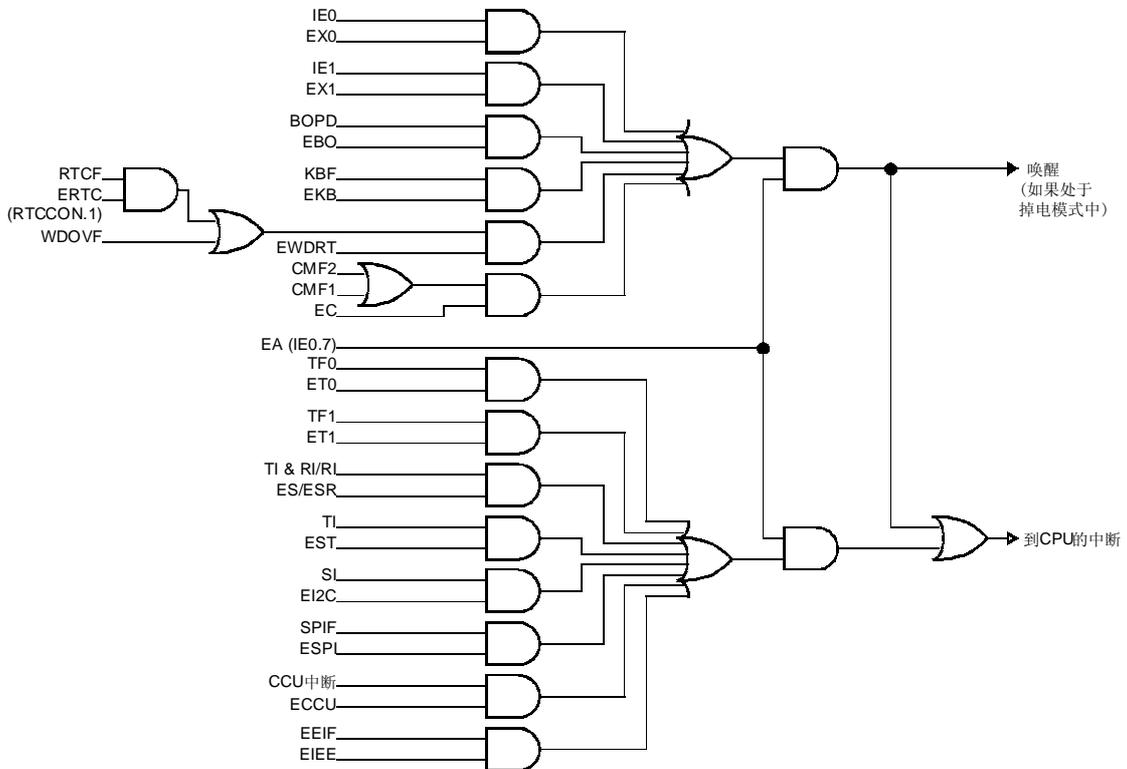


图 5 中断源，中断使能，掉电唤醒中断源

## I/O 口

LPC932 有 4 个 I/O 口, P0、P1、P2 和 P3。P0、P1 和 P2 为 8 位 I/O 口而 P3 为 2 位 I/O 口。I/O 口的具体数目取决于所选择的振荡和复位方式 (见表 4)。

表 4 可用的 I/O 口数目

时钟源	复位选项	I/O 口数目
		28 脚封装
片内振荡器或看门狗振荡器	无外部复位 (上电时除外)	26
	使用外部复位脚	25
外部时钟输入	无外部复位 (上电时除外)	25
	使用外部复位脚	24
低/中/高速振荡器 (外部晶振或谐振器)	无外部复位 (上电时除外)	24
	使用外部复位脚	23

## I/O 口配置

除了 3 个口以外, LPC932 其他所有的 I/O 口均可由软件配置成 4 种输出类型之一, 如表 5 所示。四种输出类型分别为: 准双向口(标准 8051 输出模式), 推挽, 开漏输出或仅为输入 (高阻)。每个口配置 2 个控制寄存器控制每个管脚的输出类型。P1.5(RST)只能作为输入口, 无法进行配置。P1.2(SCL/T0)和 P1.3(SDA/INT0)只能配置为输入口或开漏口。

表 5 口输出方式设定

PxM1.Y	PxM2.Y	口输出模式
0	0	准双向口
0	1	推挽
1	0	仅为输入 (高阻)
1	1	开漏

## 准双向口输出配置

准双向口输出类型可用作输出和输入功能而不需重新配置口线输出状态。这是因为当口线输出为逻辑高电平时驱动能力很弱, 允许外部装置将其拉低。当管脚输出为低时, 它的驱动能力很强, 可吸收相当大的电流。准双向口有三个上拉晶体管以适应不同的需要。

在三个上拉晶体管中, 有一个“极弱上拉”, 当口线锁存为逻辑“1”时打开, 当引脚悬空时, 这个极弱的上拉源产生很弱的上拉电流将引脚上拉为高电平。

第二个上拉晶体管称为“弱”上拉, 当口寄存器为 1 且管脚本身也为 1 时打开。此上拉提供拉电流使准双向口输出为‘1’。如果一个管脚输出为‘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, 为了把这个管脚强拉为低, 外部装置必须有足够的灌电流能力使管脚上的电压降到门槛电压以下。

第三个上拉晶体管称为“强上拉”。当口线锁存器由 0 到 1 跳变时, 这个上拉用来加快准双向口由逻辑 0 到逻辑 1 的转换。当发生这种情况时, 强上拉打开约 2 个机器周期以使管脚能够迅速上拉到高电平。

准双向口输出如图 6 所示。

尽管 LPC932 为 3V 器件, 但管脚可承受 5V 电压。在准双向口模式中, 如果用户在管脚加上 5V 电压, 将会有电流从管脚流向  $V_{DD}$ , 这将导致额外的功率消耗。因此, 建议不要在准双向口模式中向管脚施加 5V 电压。

准双向口带有一个施密特触发输入以及一个干扰抑制电路 (请参考 89LPC932 数据手册的 AC 特性部分)。

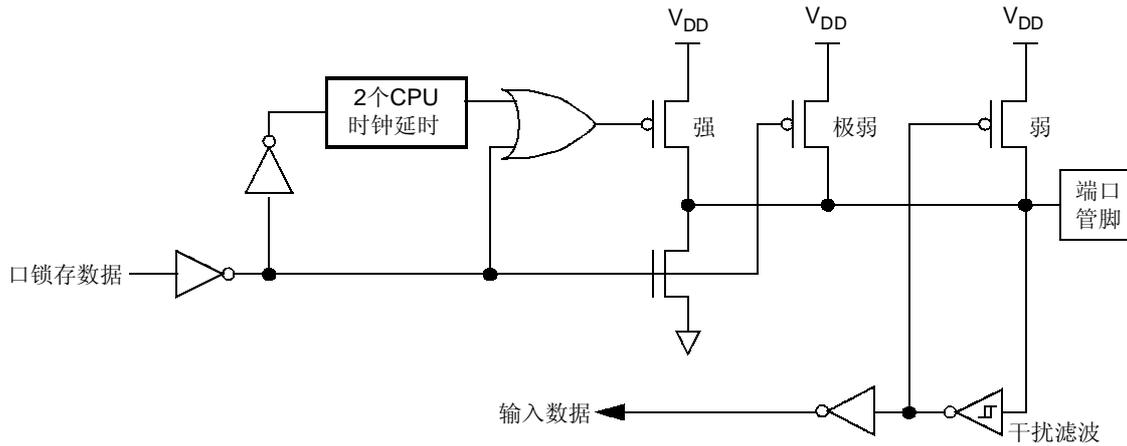


图 6 准双向口输出

### 开漏输出配置

当口线锁存器为 '0' 时，开漏输出关闭所有的上拉晶体管。作为一个逻辑输出时，这种配置方式必须有外部上拉，一般通过电阻外接到  $V_{DD}$ 。这种方式的下拉和准双向口相同。开漏输出的配置如图 7 所示。

开漏端口带有一个施密特触发输入以及一个干扰抑制电路（请参考 89LPC932 数据手册的 AC 特性部分）。

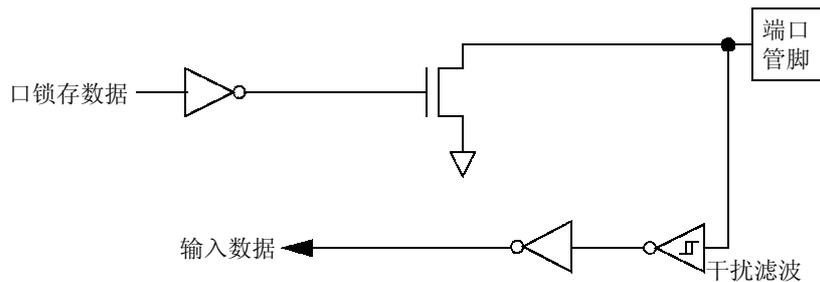


图 7 开漏输出

### 仅为输入配置

输入口配置如图 8 所示。

输入口带有一个施密特触发输入以及一个干扰抑制电路（请参考 89LPC932 数据手册的 AC 特性部分）。

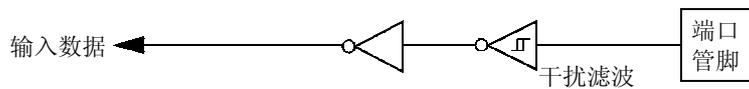


图 8 仅为输入

### 推挽输出配置

推挽输出配置的下拉结构和开漏输出以及准双向口的下拉结构相同，但当锁存器为 '1' 时提供持续的强上拉。推挽模式一般用于需要更大拉电流的情况。

推挽管脚配置如图 9 所示。

推挽管脚带有一个施密特触发输入以及一个干扰抑制电路（请参考 89LPC932 数据手册的 AC 特性部分）。

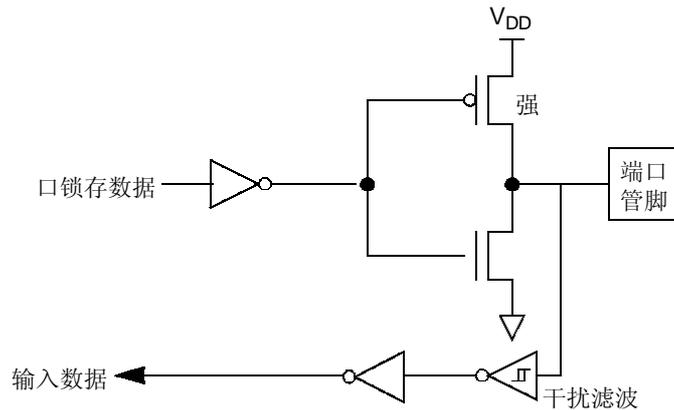


图9 推挽输出

### P0 口模拟功能

P89LPC932 集成了两个模拟比较器。为了得到最佳的模拟性能并降低功耗，用于模拟功能的管脚必须同时禁止数字输入和输出功能。将端口设置成仅为输入（高阻抗）时禁止数字信号输出，见表 5。

P0 口的数字输入可通过 PT0AD 寄存器禁止。该寄存器的位 1...5 分别对应 P0 口管脚 P0.1..P0.5。置位 PT0AD 中的位禁止相应管脚的数字输入。通过指令对禁止数字输入的管脚相应的位进行访问时，读出总是为 0。复位后，PT0AD1:5 默认为 0 以启用数字功能。

表 6 端口输出配置

口	配置位		可选功能	注
	PxM1.y	PxM2.y		
P0.0	P0M1.0	P0M2.0	KBI0,CMP2	详见“P0 口模拟功能”
P0.1	P0M1.1	P0M2.1	KBI1,CIN2B	
P0.2	P0M1.2	P0M2.2	KBI2,CIN2A	
P0.3	P0M1.3	P0M2.3	KBI3,CIN1B	
P0.4	P0M1.4	P0M2.4	KBI4,CIN1A	
P0.5	P0M1.5	P0M2.5	KBI5,CMPREF	
P0.6	P0M1.6	P0M2.6	KBI6,CMP1	
P0.7	P0M1.7	P0M2.7	KBI7,T1	
P1.0	P1M1.0	P1M2.0	TxD	
P1.1	P1M1.1	P1M2.1	RxD	
P1.2	P1M1.2	P1M2.2	T0,SCL	仅为输入或开漏
P1.3	P1M1.3	P1M2.3	$\overline{INT0}$ ,SDA	仅为输入或开漏
P1.4	P1M1.4	P1M2.4	$\overline{INT1}$	
P1.5	不可配置		RST	仅为输入。通过用户配置位 RPD (UCFG1.6) 选择作为通用输入或复位输入口。上电时总是作为复位输入口。
P1.6	P1M1.6	P1M2.6	OCB	
P1.7	P1M1.7	P1M2.7	OCC	
P2.0	P2M1.0	P2M2.0	ICB	
P2.1	P2M1.1	P2M2.1	OCD	
P2.2	P2M1.2	P2M2.2	MOSI	
P2.3	P2M1.3	P2M2.3	MISO	
P2.4	P2M1.4	P2M2.4	SS	
P2.5	P2M1.5	P2M2.5	SPICLK	
P2.6	P2M1.6	P2M2.6	OCA	
P2.7	P2M1.7	P2M2.7	ICA	
P3.0	P3M1.0	P3M2.0	XTAL2,CLKOUT	
P3.1	P3M1.1	P3M2.1	XTAL1	

### 附加端口特性

上电后所有的管脚都为高阻态（仅为输入模式）。请注意此配置不同于 LPC76x 系列器件。

- 上电之后，除 P1.5 之外，所有口都可由软件进行配置。
- P1.5 只可用于输入功能。P1.2 和 P1.3 可配置为仅为输入或开漏。

每个 LPC932 输出口都可提供灌电流驱动 LED。但是所有口的输出电流总和不能超过规定的最大电流。请查阅器件手册得到详细的规格。

LPC932 所有端口的电平转换速度都可以控制，这就可避免因电平转换过快而导致的噪声。转换速度在出厂时设定为大约 10ns 的上升时间和下降时间。

### 电源监控功能

LPC932 内含电源监控功能用于防止初始上电及掉电时的错误操作。这是通过两个硬件功能完成的：上电检测及掉电检测（Brownout Detect）。

#### 掉电检测

掉电检测功能可用于检测电源电压是否降至某一特定值以下。掉电检测的默认操作是使处理器复位。但也可通过置位 AUXR1 寄存器内 BOI(PCON.4)位和 EBO(IEN0.5)位配置为产生一个中断。

使能和禁止掉电检测通过 BOPD(PCON.5)位，PMOD1-0(PCON.1-0)和用户配置位 BOE(UCFG1.5)来实现。如果 BOE 处于未编程状态，那么不管 PMOD1-0 和 BOPD 的状态如何，掉电检测都被禁止。如果 BOE 为已编程状态，PMOD1-0 和 BOE 用于决定是否使能掉电检测。PMOD1-0 用于选择节电模式。如果 PMOD1-0=11，掉电检测被禁止。BOPD 默认为 0，指示当 BOE 已编程时，在上电后使能掉电检测。

如果掉电检测使能，操作电压的范围为 2.7V-3.6V。当  $V_{DD}$  低于掉电电压  $V_{BO}$ （见 DC 特性）时产生掉电条件，并在  $V_{DD}$  上升超过  $V_{BO}$  时取消。如果掉电检测被禁止，操作电压范围为 2.4-3.6V。如果 LPC932 器件的电源电压可以低于 2.7，BOE 应当保持未编程状态，这样器件可在 2.4V 时工作。否则持续的掉电复位将使器件无法工作。

如果掉电检测使能（BOE 已编程，PMOD1-0≠11,BOPD=0），那么不管使能的是掉电复位还是掉电中断，BOF（RSTSRC.5）都在检测到掉电时置位。BOF 一直保持置位状态，除非通过用户软件写入“0”将其清零。需要注意的是，如果 BOE 未编程，BOF 的状态没有任何意义。如果 BOE 已编程并且发生了初始的上电，BOF 将和上电标志 POF（RSTSRC.4）一起置位。

若要正确检测到掉电， $V_{DD}$  上升和下降速度必须符合一定规格。请参阅数据手册。

表 7 掉电检测选项

BOE (UCFG1.5)	PMOD1-0 (PCON.1-0)	BOPD (PCON.5)	BOI (PCON.4)	EBO (PCON.4)	EA (IEN0.5)	描述
未编程	XX	X	X	X	X	掉电检测禁止。 $V_{DD}$ 操作电压范围为 2.4-3.6V。
已编程	≠11	1	X	X	X	掉电检测禁止。 $V_{DD}$ 操作电压范围为 2.4-3.6V。但在上电时 BOPD 默认为 0。
		0	0	1	1	掉电复位使能。 $V_{DD}$ 操作范围为 2.7V-3.6V。掉电检测复位时，BOF（RSTSRC.5）将置位用于指示复位源。BOF 可通过写入“0”清零。
				1	1	掉电中断使能。 $V_{DD}$ 操作范围为 2.7-3.6V。掉电检测中断时，BOF（RSTSRC.5）将置位。BOF 可通过写入“0”清零。
				0	X	掉电复位和中断都被禁止。 $V_{DD}$ 操作范围为 2.4-3.6V。但 BOF（RSTSRC.5）将在电源电压跌落到掉电检测点时置位。BOF 可通过写入“0”清零。
			1	X	0	

### 上电检测

上电检测功能类似于掉电检测，但设计成在电源初始上电后，上升到掉电检测门槛电平之前工作。当检测到初始上电时，POF 标志置位。POF 将会一直保持置位状态，直到通过软件向该位写入“0”才会将其清零。需要注意的是，BOE (UCFG1.5) 如果已编程，BOF (RSTSRC.5) 将在 POF 置位时置位。如果 BOE 未编程，BOF 是没有意义的。

### 节电模式

LPC932 支持 3 种不同的节电模式。可通过 SFR 位 PCON1-0 进行选择—（见表 8）。

表 8 节电模式

PMOD1 (PCON. 1)	PMOD0 (PCON. 0)	描述
0	0	正常模式(默认)—非节电模式
0	1	空闲模式。空闲模式下片内外围功能继续工作，允许其在产生中断时激活处理器。任何一个使能的中断或复位均可结束空闲模式。
1	0	<p>掉电模式：掉电模式将振荡器停振以使功耗最小。LPC932 可通过任何复位或特定的中断退出掉电模式。这些特定的中断包括—外部中断 INT0/INT1、掉电中断、键盘、实时时钟（系统定时器）、看门狗以及比较器触发。通过复位唤醒处理器时必须先使能相应的复位。通过中断唤醒处理器时，必须先使能相应的中断并置位 EA (IEN0.7)。</p> <p>在掉电模式中，内部 RC 振荡器被禁止，除非选择 RC 振荡器作为系统时钟并使能 RTC。在掉电模式中，电源电压可以降低到 RAM 保持电压 <math>V_{RAM}</math>。这样将 RAM 内容保存为进入掉电模式时的状态。SFR 内容在 <math>V_{DD}</math> 低于 <math>V_{RAM}</math> 时不受保护。因此这种情况下建议通过复位唤醒处理器。在退出掉电模式前 <math>V_{DD}</math> 必须上升到操作电压范围之内。</p> <p>当处理器从掉电模式中唤醒时，它立即启动振荡器并且当振荡器稳定下来时开始执行程序。振荡器的稳定时间是指 1024 个 CPU 时钟计数（使用其中一种晶振配置时）或 256 个时钟（使用内部 RC 或外部时钟输入配置时）。在掉电模式中某些功能继续工作并消耗电流。这样就增加了掉电时的整体功耗。这些功能包括：</p> <ul style="list-style-type: none"> <li>• 掉电检测</li> <li>• 看门狗定时器（当 WDCLK(WDCON.0)为“1”时）</li> <li>• 比较器（注：比较器可通过置位 PCONA.5 和禁止比较器单独实现掉电）</li> <li>• 实时时钟/系统定时器(及其所使用的振荡器电路，RTCPD(PCONA.7)为“1”时除外)</li> </ul>
1	1	<p>完全掉电模式：和掉电模式的区别在于：完全掉电模式下掉电检测电路和电压比较器都被关闭以节省功耗。下面是所支持的唤醒源：</p> <ul style="list-style-type: none"> <li>• 看门狗定时器(当 WDCLK(WDCON.0)为“1”时)。可以产生中断或复位来唤醒 CPU。</li> <li>• 外部中断 INT0/INT1</li> <li>• 键盘中断</li> <li>• 实时时钟/系统定时器</li> </ul> <p>注：在掉电模式下使用内部 RC 振荡器作为 RTC 时钟源会增加相当大的功耗。当实时时钟在掉电模式下运行时，使用外部低频时钟可实现较低的功耗。</p>

PCON 地址: 87H								
不可位寻址	7	6	5	4	3	2	1	0
复位源: 任何复位	SMOD1	SMOD0	BOPD	BOI	GF1	GF0	PMOD1	PMOD0
复位值: 0000000B								
位	符号	功能						
PCON.7	SMOD1	串口 (UART) 双倍波特率位。为 0 时, UART 的波特率为输入频率 (T1 定时器或波特率发生器, 由 BRGCON 决定) 除以 2。为 1 时, UART 波特率为输入频率 (T1 定时器或波特率发生器)。						
PCON.6	SMOD0	帧错误位置: 一为 0 时, SCON.7 作为 UART 的 SM0 位 一为 1 时, SCON.7 作为 UART 的帧错误(FE)标志。 该位还决定 UART 接收中断 RI 发生的位置 (详见图 38 关于 RI 的描述)						
PCON.5	BOPD	掉电检测关闭。为 1 时, 掉电检测功能处于掉电状态, 即关闭状态。为 0 时, 掉电检测使能 (注: 在任何编程或擦除命令执行之前, BOPD 必须为 0。否则这些命令不会执行。)						
PCON.4	BOI	掉电检测中断使能。为 1 时, 检测到掉电时产生中断而不是复位; 为 0 时产生复位。						
PCON.3	GF1	通用标志 1, 可通过软件读写。但对器件的操作并无影响。						
PCON.2	GF0	通用标志 0, 可通过软件读写。但对器件的操作并无影响。						
PCON.1-0 PMOD1-PMOD0 低功耗模式 (见表 8)。								

图 10 电源控制寄存器 (PCON)

PCONA 地址: 85H								
不可位寻址	7	6	5	4	3	2	1	0
复位源: 任何复位	RTCPD	DEEPPD	VCPD	-	I2PD	SPPD	SPD	CCUPD
复位值: 0000000B								
位	符号	功能						
PCONA.7	RTCPD	实时时钟掉电: 该位为 1 时, 用于实时时钟的内部时钟被禁止。						
PCONA.6	DEEPPD	数据 EEPROM 掉电: 该位为 1 时, 数据 EEPROM 掉电。注: 如果处于掉电或完全掉电模式下, 不管该位的状态如何, 数据 EEPROM 都会进入掉电状态。						
PCONA.5	VCPD	模拟电压比较器掉电: 该位为 1 时, 电压比较器掉电。用户必须在置位该位之前禁止电压比较器。						
PCONA.4	-	保留将来之用						
PCONA.3	I2PD	I <sup>2</sup> C 掉电: 该位为 1 时, I <sup>2</sup> C 的内部时钟被禁止。注: 如果处于掉电或完全掉电模式下, 不管该位的状态如何, I <sup>2</sup> C 时钟都将会被禁止。						
PCONA.2	SPPD	SPI 掉电: 该位为 1 时, SPI 的内部时钟被禁止。注: 如果处于掉电或完全掉电模式下, 不管该位的状态如何, SPI 内部时钟都将会被禁止。						
PCONA.1	SPD	串口 (UART) 掉电: 该位为 1 时, UART 的内部时钟被禁止。注: 如果处于掉电或完全掉电模式下, 不管该位的状态如何, UART 内部时钟都将会被禁止。						
PCONA.0	CCUPD	比较/捕获单元 (CCU) 掉电: 该位为 1 时, CCU 的内部时钟被禁止。注: 如果处于掉电或完全掉电模式下, 不管该位的状态如何, CCU 的内部时钟都将会被禁止。 (注: FCFG1 中的 CCUDIS 位优先于该位。如果 CCUDIS=1, CCU 被关闭) 注: 掉电检测的掉电控制位为 PCON.5。						

图 11 电源控制寄存器 A

### 复位

P1.5 $\overline{\text{RST}}$  管脚可作为低有效复位输入或数字输入口。当 UCFG1 寄存器中的位 RPE(复位管脚使能)置位时, 使能外部复位输入功能。当清零时, 复位脚可作为一个输入管脚。

注: 在上电过程中, RPE 选择无效, 该管脚总是作为外部复位输入。在上电完成之后, 该管脚可根据

RPE 位的状态作为外部复位输入或数字输入口。只有上电复位会暂时使 RPE 的设定失效，其它复位源无法影响 RPE 位的设定。

复位可由下列复位源引起（见图 12）：

- 外部复位脚（上电或通过 UCFG1 配置为使用外部复位）
- 上电检测
- 掉电检测
- 看门狗定时器
- 软件复位
- UART 间隔检测复位

每一个复位源在复位寄存器中都有一个对应的标志。用户可读取该寄存器以判断导致最近一次复位的复位源是哪一个。这些标志位可通过软件写入“0”清零。可以有多个的标志位位置：

- 上电复位时，POF 和 BOF 都置位，而其它标志位清零
- 对于其它的复位，任何在复位之前置位的标志位不会受到复位的影响

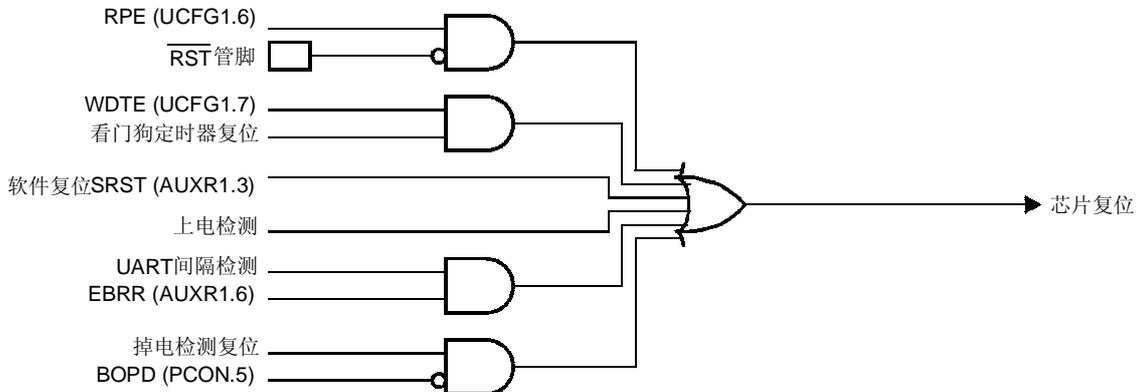


图 12 复位源方框图

RSTSRC 地址：DFH								
不可位寻址								
复位源：上电复位	7	6	5	4	3	2	1	0
	—	—	BOF	POF	R_BK	R_WD	R_SF	R_EX
复位值：xx110000B（这是上电复位值，其它复位源将置位相应的位）								
位	符号	功能						
RSTSRC.7-6	—	保留将来之用。用户程序不要将其置 1。						
RSTSRC.5	BOF	掉电检测标志。当掉电检测激活时，该位置位。它将保持置位直到通过软件写入 0 将其清零。（注：上电复位时，POF 和该位置位，而其它标志位都清零）						
RSTSRC.4	POF	上电检测标志。当上电检测激活时，POF 置位以指示初始上电的状态。POF 标志将保持置位直到软件写入 0 将其清零。（注：上电复位时，BOF 和该位置位，而其它标志位都清零）						
RSTSRC.3	R_BK	间隔字符检测复位。当检测到间隔并且 EBRR(AUXR1.6)置位时，将产生系统复位并将该位置位。通过软件写入 0 或上电复位清零。						
RSTSRC.2	R_WD	看门狗定时器复位标志。通过软件写入 0 或上电复位清零（注：UCFG1.7 必须为 1）						
RSTSRC.1	R_SF	软件复位标志。通过软件写入 0 或上电复位清零。						
RSTSRC.0	R_EX	外部复位标志。当该位为 1 时，它指示复位源为外部管脚复位。该标志位可通过软件写入 0 或上电复位清零。如果 RST 在上电复位结束后仍然保持低电平，R_EX 将置位。						

图 13 复位源寄存器

### 复位向量

在复位之后，LPC932 将从地址 0000h 或引导 (Boot) 地址处取指令。将引导向量作为地址高字节，00h 作为地址低字节即构成了引导地址。如果发生 UART 间隔复位或者非易失性引导状态位 (BOOTSTAT.0) =1，或者器件被强制进入 ISP 模式，器件将会使用引导向量作为复位向量。否则，指令将从地址 0000h 处开始执行。

### 定时器 / 计数器 0 和 1

LPC932 有两个通用定时/计数器，与标准 80C51 定时器 0 及定时器 1 兼容。这两个定时器均可配置为定时器或事件计数器 (见图 14)。另外增加了定时器 0/1 溢出时 T0/T1 脚自动翻转的功能选项。

用作“定时器”功能时，每经过一个 PCLK，寄存器值加 1。

用作“计数器”功能时，寄存器在对应的外部输入管脚 T0/T1 的每一个下降沿加 1。使用该功能时，外部输入每个机器周期被采样一次。当某一周期时管脚状态采样为高而下一周期采样为低时，计数器加 1。寄存器值在检测到跳变的下一个机器周期被更新。由于检测下降沿跳变需要花费两个机器周期 (4 个 CPU 时钟) 的时间，所以计数频率最大值为 CPU 时钟频率的 1/4。对外部输入信号的占空比并无限制，但为了保证给定的电平信号在其改变之前至少被采样一次，信号必须至少保持一个完整的机器周期。

“定时”或“计数”功能通过特殊功能寄存器 TMOD 中的控制位 TnC/T $\bar{T}$  (n=0 和 1 分别表示定时器 0 和 1) 选择。定时器 0 和定时器 1 有 5 种工作模式 (模式 0, 1, 2, 3 和 6)，这几种工作模式通过 TMOD 中的 TnM1, TnM0 和 TAMOD 中的 TnM2 进行选择。模式 0、1、2 和 6 对于两个定时/计数器是一样的。模式 3 则不同。操作模式在后面详细描述。

TMOD 地址: 89H									
不可位寻址	7      6      5      4      3      2      1      0								
复位源: 任何复位	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>T1GATE</td> <td>T1C<math>\bar{T}</math></td> <td>T1M1</td> <td>T1M0</td> <td>T0GATE</td> <td>T0C<math>\bar{T}</math></td> <td>T0M1</td> <td>T0M0</td> </tr> </table>	T1GATE	T1C $\bar{T}$	T1M1	T1M0	T0GATE	T0C $\bar{T}$	T0M1	T0M0
T1GATE	T1C $\bar{T}$	T1M1	T1M0	T0GATE	T0C $\bar{T}$	T0M1	T0M0		
复位值: 0000000B									
位	符号      功能								
TMOD.7	T1GATE      定时器 1 的选通控制。该位置位时，只有当 $\overline{INT1}$ 脚为高电平并且 TR1 控制位置位时才使能定时/计数器 1。清零时，置位 TR1 即可使能定时器/计数器 1。								
TMOD.6	T1C $\bar{T}$ 定时器 1 运行控制位。通过软件置位/清零将定时器 1 打开/关闭。								
TMOD.5,4	T1M1, T1M0      定时器 1 模式选择。这两个位和 T1M2 (TAMOD.4) 用于确定定时器 1 的模式 (见图 15)。								
TMOD.3	T0GATE      用于定时器 0，置位时只有在 $\overline{INT0}$ 脚置高及 TR0 控制位置位时才可打开定时器/计数器 0。清零时，置位 TR0 即可打开定时器/计数器 0。								
TMOD.2	T0C $\bar{T}$ 用于定时器 0，控制定时器 0 用作定时器或计数器，清零则用作定时器，置位用作计数器。								
TMOD.1,0	T0M1, T0M0      定时器 0 模式选择。这两个位和 T0M2 (TAMOD.0) 用于确定定时器 0 的模式 (见图 15)。								

图 14 定时/计数器模式控制寄存器 (TMOD)

TAMOD 地址: 8FH								
不可位寻址	7	6	5	4	3	2	1	0
复位源: 任何复位	-	-	-	T1M2	-	-	-	T0M2
复位值: xxx0xxx0B								
位	符号	功能						
TAMOD.7-5	—	保留将来之用。用户程序不要将其置 1。						
TAMOD.4	T1M2	定时器 1 模式选择位 2。和 T1M1,T1M0 一起确定定时器 1 的模式。						
TAMOD.3-1	—	保留将来之用。用户程序不要将其置 1。						
TAMOD.0	T0M2	定时器 0 模式选择位 2, 和 T0M1,T0M0 一起确定定时器 0 的模式。						
	TnM2-TnM0	定时器模式						
	000	8048 定时器。“TLn” 用作 5 位预分频器						
	001	16 位定时器/计数器, THn 和 TLn 级联。无预分频器。						
	010	8 位自动重载定时器, 当 TLn 溢出时将 THn 存放的值装入 TLn。						
	011	定时器 0 此时作为双 8 位定时/计数器。TL0 作为一个 8 位定时器/计数器, 通过标准定时器 0 控制位控制。TH0 仅作为一个 8 位定时器, 由定时器 1 控制位控制, 在该模式下定时器 1 关闭。						
	100	保留将来之用。用户不要配置为该模式。						
	101	保留将来之用。用户不要配置为该模式。						
	110	PWM 模式 (见“模式 6”)。						
	111	保留将来之用。用户不要配置为该模式。						

图 15 定时/计数器附加模式控制寄存器(TAMOD)

### 模式 0

将定时器设置成模式 0 时类似 8048 定时器, 即 8 位计数器带 32 分频的预分频器。图 17 所示为模式 0 工作方式。

在此模式中, 定时器寄存器配置为 13 位寄存器。当计数从全为“1”翻转为全为“0”时, 置位定时器中断标志 TF<sub>n</sub>。当 TR<sub>n</sub>=1 并且 GATE=0 或  $\overline{INTn}=1$  时, 定时器 0 计数输入使能。(置位 GATE 可使外部中断输入  $\overline{INTn}$  对定时器进行控制, 以便于实现脉宽的测量)。TR<sub>n</sub> 是 TCON 寄存器中的控制位, GATE 位于 TMOD 寄存器中。

该 13 位寄存器包含 TH<sub>n</sub> 的 8 位及 TL<sub>n</sub> 的低 5 位, TL<sub>n</sub> 的高 3 位不定, 可将其忽略。置位运行标志(TR<sub>n</sub>)不会清零该寄存器。

定时器 0 及定时器 1 在模式 0 中的操作相同。见图 17。

### 模式 1

模式 1 除了使用了 TH<sub>n</sub> 及 TL<sub>n</sub> 全部 16 位外, 其它与模式 0 相同。见图 18。

### 模式 2

在此模式中, 定时器寄存器作为可自动重载的 8 位计数器 (TL<sub>n</sub>), 如图 19 所示, TL<sub>n</sub> 溢出不仅置位 TF<sub>n</sub>, 而且将 TH<sub>n</sub> 内容重新装入 TL<sub>n</sub>, TH<sub>n</sub> 内容必须由软件预置。重装时 TH<sub>n</sub> 内容不变。定时器 0 及定时

器 1 在模式 2 中的操作相同。

### 模式 3

在模式 3 时定时器 1 关闭，这等效于 TR1=0。

在此模式中，定时器 0 的 TL0 及 TH0 作为两个独立的 8 位计数器。图 20 所示为模式 3 时定时器 0 的逻辑图。TL0 占用定时器 0 的控制位：T0C/T, T0GATE, TR0,  $\overline{\text{INT0}}$  及 TF0。TH0 限定为定时器功能（对机器周期计数），并占用定时器 1 的 TR1 及 TF1。此时 TH0 控制“定时器 1”中断。

模式 3 用于需要一个额外的 8 位定时器的场合。定时器 0 工作于模式 3 时，LPC932 可以看作有 3 个定时器/计数器。注：当定时器 0 工作于模式 3 时，定时器 1 可通过进入/退出它自己的模式 3 实现打开/关闭。它仍可用作串行端口的波特率发生器，或者应用于任何不要求中断的场合。

### 模式 6

在该模式中，定时器可以改变为一个具有 256 个定时器时钟周期的 PWM（见图 21）。它的结构和模式 2 相似，但有以下几点除外：

- TF<sub>n</sub> (n=0 和 1 分别表示定时器 0 和 1) 通过硬件置位和清零；
- TF<sub>n</sub> 低电平宽度为 TH<sub>n</sub> 的值，必须介于 1 到 254 之间，并且；
- TF<sub>n</sub> 高电平周期总是为 256-TH<sub>n</sub>。
- 向 TH<sub>n</sub> 装入 00h 将使 Tx 脚强制为高电平。向 TH<sub>n</sub> 装入 0FFh 将使 Tx 脚强制为低电平。

注：在 TF<sub>n</sub> 上升沿时中断仍然使能，并且 TF<sub>n</sub> 仍然可以像其它任何模式一样通过软件清零。

TCON 地址: 88H									
可位寻址		7	6	5	4	3	2	1	0
复位源: 任何复位		TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
复位值: 00000000B									
位	符号	功能							
TCON.7	TF1	定时器 1 溢出标志。在定时/计数器溢出时由硬件置位。中断处理时由硬件清零。或通过软件清零（模式 6 除外，见上面一节）。							
TCON.6	TR1	定时器 1 运行控制位。由软件置位/清零以打开/关闭定时器 1。							
TCON.5	TF0	定时器 0 溢出标志。在定时/计数器溢出时由硬件置位。中断向量指向中断服务程序时，该标志由硬件清零或通过软件清零（模式 6 除外，见上面一节）。							
TCON.4	TR0	定时器 0 运行控制位。由软件置位/清零以打开/关闭定时器 0。							
TCON.3	IE1	中断 1 边沿标志。当检测到外部中断 1 边沿时由硬件置位该标志。中断处理时由硬件清零，或通过软件清零。							
TCON.2	IT1	中断 1 类型控制位，由软件置位/清零以选择外部中断以下降沿/低电平方式触发。							
TCON.1	IE0	中断 0 边沿标志。当检测到外部中断 1 边沿时由硬件置位该标志。中断处理时由硬件清零，或通过软件清零。							
TCON.0	IT0	中断 0 类型控制位。由软件置位/清零以选择外部中断以下降沿/低电平方式触发。							

图 16 定时器/计数器控制寄存器 (TCON)

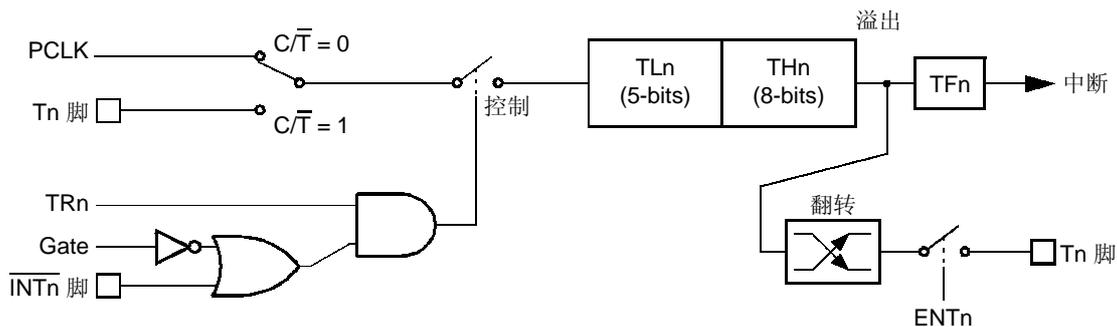


图 17 模式 0 时的定时/计数器 0 或 1 (13 位计数器)

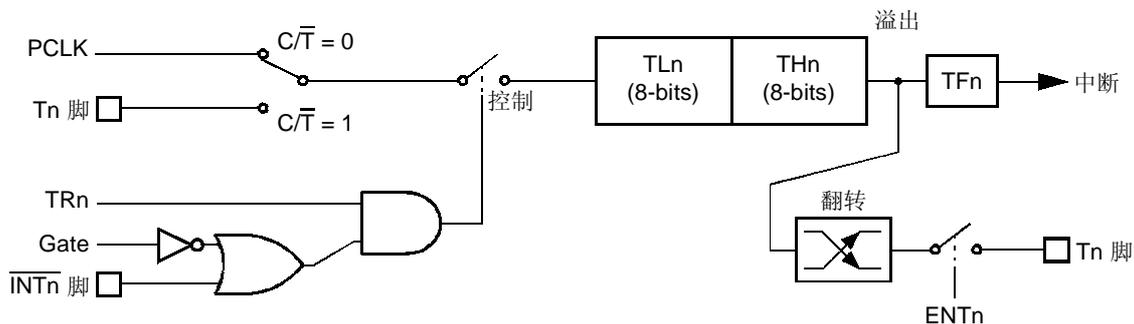


图 18 模式 1 时的定时/计数器 0 或 1 (16 位定时/计数器)

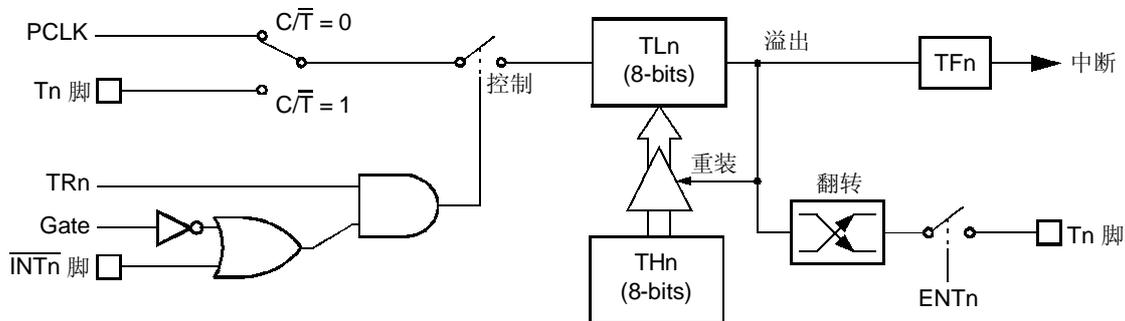


图 19 模式 2 时的定时/计数器 0 或 1 (8 位自动重装)

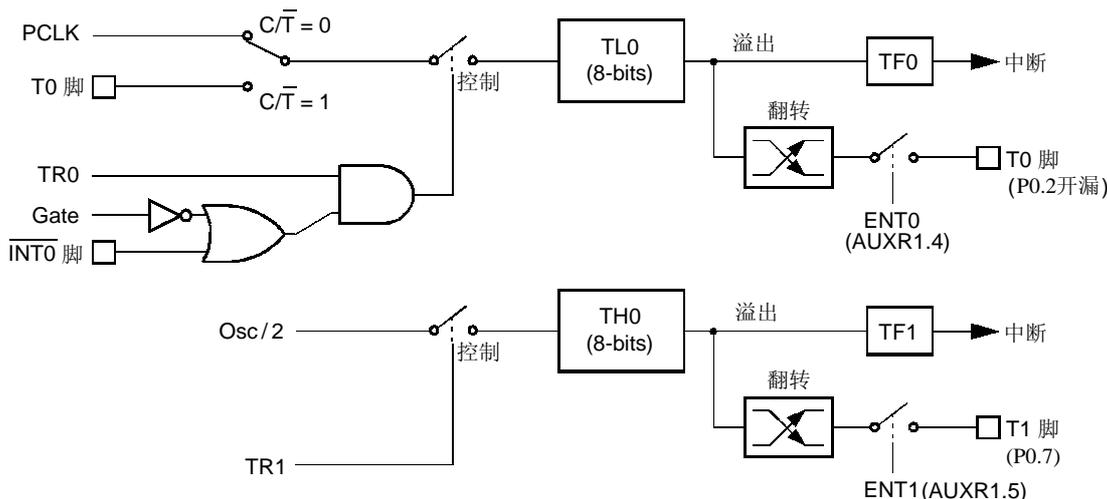


图 20 模式 3 时的定时器 0 (双 8 位计数器)

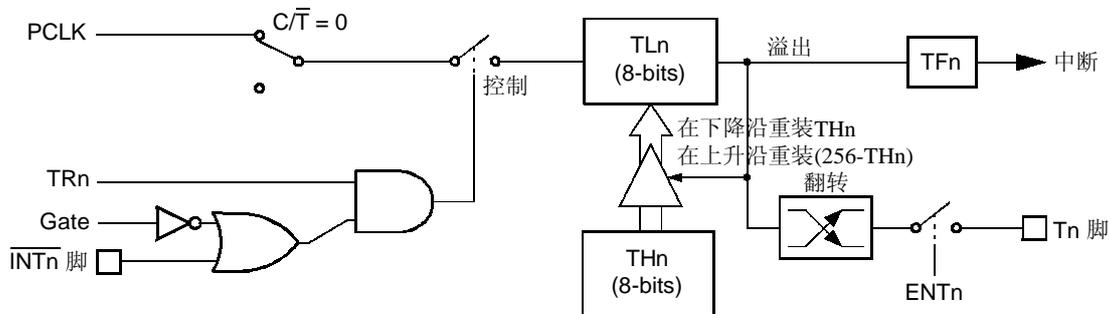


图 21 模式 6 时的定时器 0 或 1 (PWM 自动重装)

### 定时器溢出触发输出

定时器 0 和 1 可配置为发生定时器溢出时自动触发端口输出。T0/T1 的计数输入和 PWM 输出与定时器触发输出占用相同的管脚。此功能通过 AUXR1 寄存器中的控制位 ENT0 和 ENT1 分别使能定时器 0 和 1。该模式打开时，在首次定时器溢出前端口的输出为逻辑 1。为了使该模式生效，必须清零 C/T 位以选择 PCLK 作为定时器的时钟源。

### 实时时钟/系统定时器

LPC932 具有一个简单的实时时钟。它允许用户在器件其它部分掉电时能够继续运行一个精确的定时器。实时时钟可以作为一个中断或一个唤醒源 (见图 5)。实时时钟是一个 23 位倒计数器。该定时器的时钟源可以是 CPU 时钟 (CCLK) 或者 XTAL1-2 振荡器 (前提是 XTAL1-2 振荡器不作为 CPU 的时钟源)。如果 XTAL1-2 振荡器作为 CPU 时钟源，不管 RTCCON 寄存器中的 RTCS1:0 状态如何，实时时钟 (RTC) 都会使用 CCLK 作为它的时钟源。RTC 有 3 个相关的寄存器：

- RTCCON—实时时钟控制
- RTCH—实时时钟计数器重装高字节 (位 22—15)
- RTCL—实时时钟计数器重装低字节 (位 14—7)

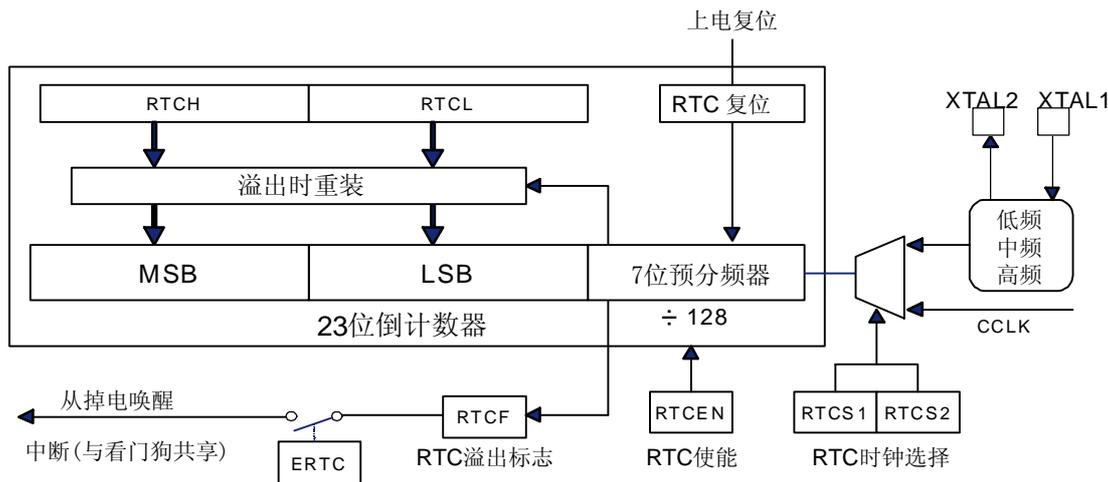


图 22 实时时钟/系统定时器框图

实时时钟/系统定时器可通过置位 `RTCEN(RTCCON.0)` 使能。实时时钟是一个 23 位倒计数器（当 `RTCEN=0` 时，初始化为全 0），包括一个 7 位预分频器和一个 16 位可重装的倒计数器。当 `RTCEN` 为 1 时，计数器首先装入 (`RTCH,RTCL,'1111111'`) 然后向下计数。当它为全零时，计数器会再次装入 (`RTCH,RTCL,'1111111'`)，而标志 `RTCF(RTCCON.7)` 将会置位。

在实时时钟重装过程中，任何对 `RTCH` 和 `RTCL` 的写操作都会导致计数器重装失败。当前计数终止时，`RTCH` 和 `RTCL` 的内容将会装入计数器并开始进行新的计数。通过清零 `RTCEN` 然后再将其置位，可强制计数器立即重装。

### 实时时钟源

如果将内部 RC 振荡器或内部看门狗定时器作为 CPU 时钟（内部或外部振荡器模式），那么 `RTCS1-0(RTCCON.6-5)` 用于选择 RTC 的时钟源。如果使用外部晶体振荡器或 `XTAL1` 的外部时钟输入作为 CPU 时钟，RTC 将会使用 `CCLK` 作为它的时钟源。

表 9 实时时钟/系统定时器时钟源

RTCS1 (RTCCON.6)	RTCS0 (RTCCON.5)	FOSC2 (UCFG1.2)	FOSC1 (UCFG1.1)	FOSC0 (UCFG1.0)	RTC 时钟源	CPU 时钟源
x	x	0	0	0	CCLK	高频晶振
x	x	0	0	1	CCLK	中频晶振
x	x	0	1	0	CCLK	低频晶振
0	0	0	1	1	高频晶振	内部 RC 振荡器
0	1				中频晶振	
1	0				低频晶振	
1	1				CCLK	
0	0	1	0	0	高频晶振	看门狗振荡器
0	1				中频晶振	
1	0				低频晶振	
1	1				CCLK	
x	x	1	0	1	未定义	未定义
x	x	1	1	0	未定义	未定义
x	x	1	1	1	CCLK	外部时钟输入

### RTCS1-0 的更改

当 RTC 当前使能（即 `RTCCON.0=1`）时，`RTCS1-0` 不能被改变。置位 `RTCEN` 和更改 `RTCS1-0` 可以在

一条对 RTCCON 的写指令内完成。但如果 RTCEN=1，在更改 RTCS1-0 之前必须先清零 RTCEN。

### 实时时钟中断/唤醒

如果 ERTC(RTCCON.1), EWDRT(IEN0.6)和 EA(IEN0.7)置位为 1，RTCF 可作为一个中断源。该中断向量和看门狗定时器的中断向量相同。还可配置为器件的一个唤醒源。

### 复位源对实时时钟的影响

只有上电复位会将实时时钟及其相关的 SFR 复位为默认状态。

RTCCON 地址: D1H	
不可位寻址	7 6 5 4 3 2 1 0
复位源: 上电复位	RTCF RTCS1 RTCS0 - - - ERTC RTCEN
复位值: 011xxx00B	
位	符号 功能
RTCCON.7	RTCF 实时时钟标志。当 23 位实时时钟到达计数值 0 时，该位置位。可通过软件清零。
RTCCON.6-5	RTCS1-0 实时时钟源选择（见表 9）
RTCCON.4-2	- 保留将来之用。用户程序请勿将其置 1。
RTCCON.1	ERTC 实时时钟中断使能。实时时钟和看门狗定时器共用一个中断源。注：如果用户配置位 WDTE(UCFG1.7)为 0，看门狗定时器可使能为产生中断。用户可读取 RTCF(RTCCON.7)位以确定中断是否由实时时钟产生。
RTCCON.0	RTCEN 0：禁止实时时钟；1：使能实时时钟。如果该位为 0，实时时钟将被禁止。注：该位不会使实时时钟进入掉电状态。不管 RTCEN 的状态如何，RTCPD(PCONA.7)为 1 时会使实时时钟进入掉电，并且禁止该模块。

图 23 RTCCON 寄存器

### 捕获/比较单元 (CCU)

该单元有以下特性：

- 16 位定时器，并可在溢出时重装 16 位值
- 时钟 (CCUCLK) 可选，可将时钟源以 1 到 1024 间的任何整数进行分频。
- 4 个比较/PWM 输出，可选择极性
- 对称/非对称 PWM 选择
- 2 路比较输入，带有事件计数器和数字噪声滤波器
- 7 个中断（1 个溢出，2 个捕获，4 个比较）使用同一个中断向量
- 通过映像寄存器可实现安全的 16 位读/写

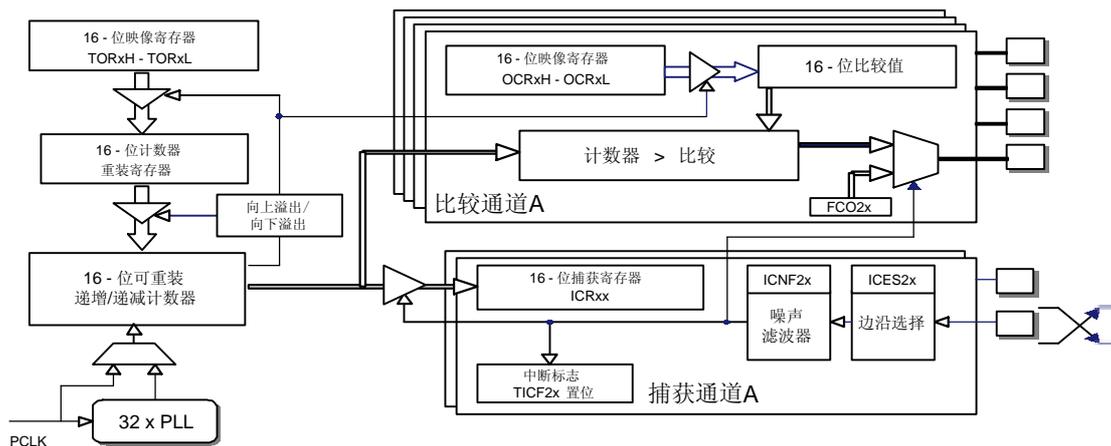


图 24 捕获比较单元 (CCU) 框图

### CCU 时钟 (CCUCLK)

CCU 的时钟为 CCUCLK。它可以是基本定时器模式中的 PCLK，或 PLL 输出（见图 24）。PLL 使用的时钟源为 0.5MHz 到 1MHz，32 倍频后产生 PWM 模式（不对称或对称）中 16MHz 到 32MHz 的 CCUCLK。PLL 包含一个 4 位分频器（TCR21 寄存器中的 PLLDV3:0）将 PCLK 分频为 0.5MHz 到 1MHz。

### CCU 时钟预分频

CCUCLK 可通过预分频器进一步分频。该预分频器是一个 10 位自由运行的计数器，在溢出时可编程重新装载。将一个值写入预分频器时将导致预分频器重新启动。

### 基本定时器操作

该定时器是一个自由运行的，可递增/递减的计数器。它的步距由预分频器决定。定时器的启动通过置位 CCU 控制寄存器 0（TCR20）中的 CCU 模式选择位 TMOD21 和 TMOD20 实现。详见 TCR20 寄存器的描述（图 26）。

CCU 方向控制位 TDIR2 决定计数的方向。TDIR2=0：递增计数，TDIR2=1，递减计数。如果在计数器运行时改变定时器的计数方向，那么在写入 TDIR2 之后的 CCUCLK 周期内，计数序列发生翻转。定时器在任何时候都可进行读或写。新写入的值将在预分频器溢出时生效。定时器通过两个寄存器 TL2（低字节）和 TH2（高字节）进行访问。第三个 16 位 SFR，TOR2H:TOR2L，决定溢出后的重装值。TL2, TH2 和 TOR2H, TOR2L 在复位后都为 0。

**递增计数：**当定时器内容为 FFFFH 时，在下一个 CCUCLK 周期计数器的值将会被设置为 TOR2H:TOR2L 的内容。

**递减计数：**当定时器的内容为 0000H 时，在下一个 CCUCLK 周期计数器的值将会被设置为 TOR2H:TOR2L 的内容。在执行重装 CCUCLK 周期内，CCU 中断标志寄存器（TIFR2）中的 CCU 定时器溢出中断标志（TOIF2）将会置位。如果此时 IE 寄存器中的 EA 和 IEN1 寄存器中的 ECCU 位都置位，程序将指向溢出中断的向量。用户必须通过软件写入逻辑 0 将该位清零。

向 TOR2H 和 TOR2L 写入重装值时，写入的值送入两个 8 位映像寄存器。为了将映像寄存器中的值锁存到 TOR2H 和 TOR2L，用户必须将 CCU 定时器控制寄存器中的比较/溢出更新位 TCOU2 置位。该位的功能取决于定时器运行在 PWM 模式还是基本定时器模式。在基本定时器模式中，置位 TCOU2 将会导致立即执行锁存，并且读取 TCOU2 的值将一直得到 0；在 PWM 模式中，置位 TCOU2 将使映像寄存器的内容在

下次 CCU 定时器溢出时更新。当锁存被挂起时，TCOU2 的值读为 1；当锁存被执行后，TCOU2 的值将变为 0。TCOU2 还控制输出比较寄存器 OCR2A,OCR2B 和 OCR2C 的锁存。

当写入定时器高字节 TH2 时，写入的内容送到一个映像寄存器。对 TL2 进行写操作时，在 TL2 的内容更新的同时，映像寄存器的值送入定时器的高字节。如果一个对 TL2 的写操作之后跟着另一个对 TL2 的写操作，中间没有对 TH2 的写操作时，TH2 的值将直接送入定时器的高字节。

如果将 16 位 CCU 定时器作为一个 8 位定时器，用户可将 FFh (用于递增计数) 或 00h (用于递减计数) 写入 TH2。当写 TL2 时，FFh 或 00h 将会装入 CCU 定时器。在 8 位定时器操作时，用户不需要重新写 TH2。除非计数方向发生改变。

当读取定时器时，必须先读取 TL2。读 TL2 时，定时器高字节的内容在同一个 PCLK 周期内传送到映像寄存器。当读取 TH2 时，实际上是读取映像寄存器的内容。如果一个对 TL2 的读操作之后跟着另一个对 TL2 的读操作，而中间没有对 TH2 的读操作时，定时器的高字节的值将直接送入 TH2。

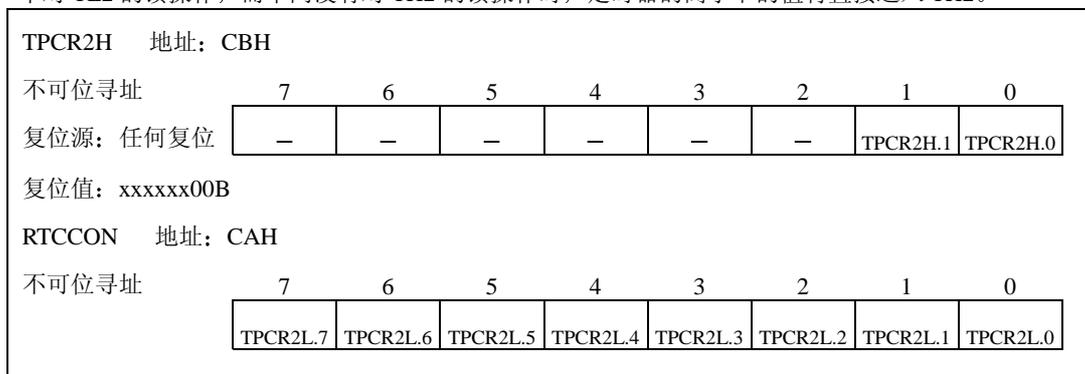


图 25 CCU 预分频器控制寄存器

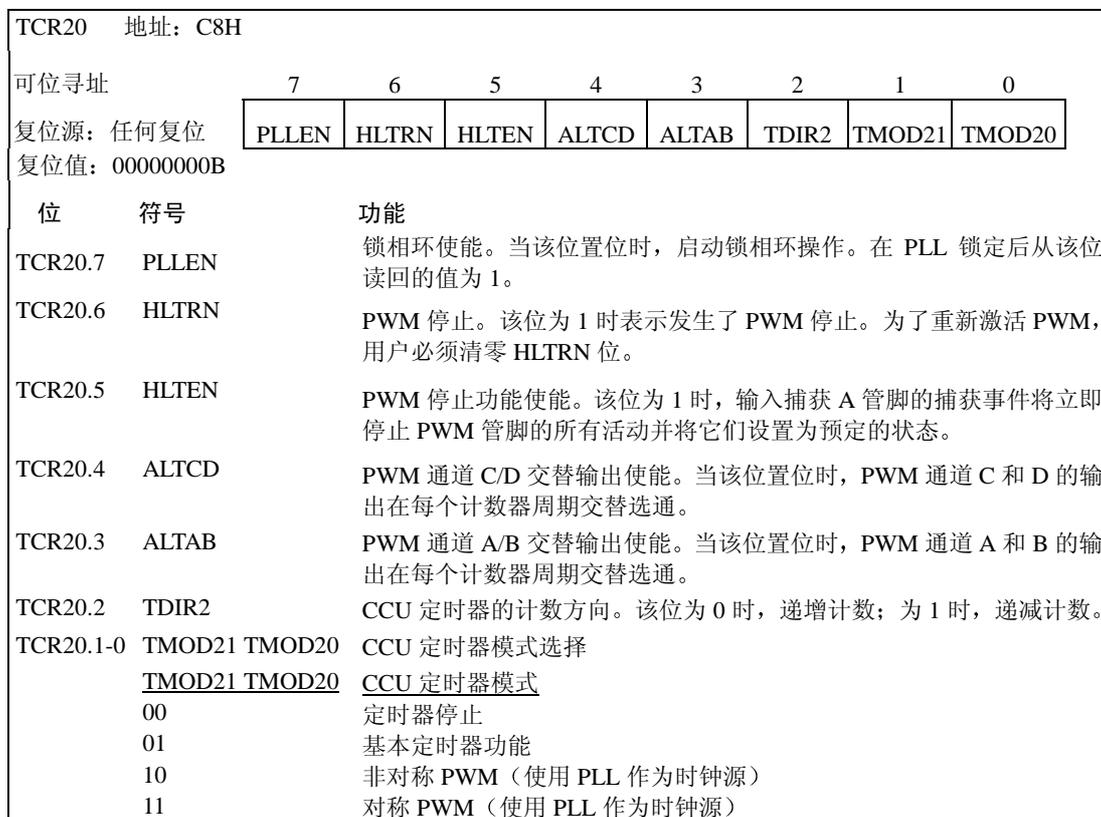


图 26 CCU 控制寄存器 0

### 输出比较

四个输出比较通道 A,B,C 和 D 分别通过 4 个 16 位寄存器 OCRAH:OCRAL, OCRBH:OCRBL, OCRCH:OCRCL 和 OCRDH:OCRDL 进行控制。每个输出比较通道在进行操作之前都必须使能。设置捕获比较 x 控制寄存器—CCCRx(x=A,B,C,D)中的 OCMx1:0 位可使能不同的输出比较类型。当使能一个比较通道时，用户必须将相应的 I/O 口设置为所需要的输出类型（注：作为比较通道输出时，管脚 P2.6, P1.6, P1.7, P2.1 必须置为 1）。当 TH2:TL2 的内容和 OCRxH:OCRxL 相匹配时，TIFR2 中的定时器输出比较中断标志 TOCFx 置位。该位的置位发生在出现比较的下一个 CCUCCLK 周期。如果 EA 和定时器输出比较中断使能位 TOCIE2x（位于 TICR2 寄存器中）以及 IEN1 中的 ECCU 位都已置位，程序计数器将指向相应的中断向量。用户可通过软件写入“0”将其手动清零。

OC CRx 中的两个位，输出比较 x 模式位 OCMx1 和 OCMx0 对匹配发生时所采取的动作进行选择。即使中断被禁止，所使能的比较动作仍然会发生。

为了保证比较输出的动作，比较值必须在 CCU 定时器的计数范围之内。

当比较通道使能时，I/O 口（必须配置为输出）将会连接到一个由比较逻辑所控制的内部锁存。该锁存的值在复位后为 0，并且只能通过调用一个强制的比较来改变。强制比较的产生是通过置位 OC CRx 中的强制比较 x 输出位 FCOx 实现的。当 OCMx1/OCMx0 对 I/O 口进行设置后，向该位写入 1 将使相应的 I/O 口产生一个跳变，但不会产生中断。在基本定时器操作模式中，读取 FCOx 位总是为 0（注：该位在 PWM 模式中有不同的功能）。当输出比较管脚使能并连接到比较锁存，比较管脚的状态保持不变，直到发生比较事件或强制比较为止。

CCR20 地址: CCCRA: EAH, CCCRB: EBH, CCCRC: ECH, CCCRD: EDH								
不可位寻址	7	6	5	4	3	2	1	0
复位源: 任何复位	ICECx2	ICECx1	ICECx0	ICESx	ICNFx	FCOx	OCMx1	OCMx0
复位值: 00000000B								
位	符号	功能						
CCCRx.7	ICECx2	捕获延迟设定位 2。详见表 10						
CCCRx.6	ICECx1	捕获延迟设定位 2。详见表 10						
CCCRx.5	ICECx0	捕获延迟设定位 2。详见表 10						
CCCRx.4	ICESx	输入捕获 x 边沿选择位。为 0 时，下降沿触发捕获；为 1 时，上升沿触发捕获。						
CCCRx.3	ICNFx	输入捕获 x 噪声滤波器使能位。为 1 时，捕获逻辑需要检测到同一个值的 4 个连续采样以识别捕获事件的边沿。输入每两个 CCLK 周期被采样一次，与定时器速度无关。						
CCCRx.2	FCOx	强制比较 x 输出位。置位时，调用一个强制的比较						
CCCRx.1-0	OCMx1,OCMx0	输出比较 x 模式。见表 11						

图 27 捕获比较控制寄存器

当用户改变输出比较值时，写入 OCRH2x 和 OCRL2x 的值传送到两个 8 位映像寄存器中。为了将映像寄存器的内容锁存到捕获比较寄存器中，用户必须向 CCU 定时器比较/溢出更新位 TCOU2（位于 CCU 控制寄存器 1—TCR1 中）写入 1。该位的功能取决于定时器运行在 PWM 模式还是基本定时器模式。在基本定时器模式中，向 TCOU2 写入 1 将会导致立即执行锁存，并且读取 TCOU2 的值将一直得到 0；在 PWM 模式中，置位 TCOU2 将使映像寄存器的内容在下次 CCU 定时器溢出时更新。当锁存被挂起时，TCOU2 的值读数为 1；当锁存被执行后，TCOU2 的值将变为 0。TCOU2 还控制所有输出比较寄存器和定时器溢出重装寄存器 TOR2 的锁存。

### 输入捕获

输入捕获总是处于使能状态。每次在两个输入捕获管脚产生捕获事件时，定时器的内容都会传送到相应的 16 位输入捕获寄存器 ICRAH:ICRAL 或 ICRBH:ICRBL 中。捕获事件由 CCCR<sub>x</sub> 寄存器中的输入捕获边沿选择位 ICES<sub>x</sub> (x 为 A 或者 B) 进行定义。用户必须将相关的 I/O 口配置为输入状态，以实现外部事件触发捕获。

在捕获输入端可使能采样噪声滤波器。当输入捕获噪声滤波器 ICNF<sub>x</sub> 位置位时，捕获逻辑需要检测到同一个值的 4 个连续采样以识别捕获事件的边沿。输入值每两个 CCLK 周期被采样一次，与定时器速度无关。

通过一个事件计数器可以设置在延迟一定数目的捕获事件之后触发捕获。寄存器 CCCR<sub>x</sub> 寄存器中的三个位 ICEC<sub>x2</sub>, ICEC<sub>x1</sub> 和 ICEC<sub>x0</sub> 用于决定捕获逻辑在触发捕获之前所监视到的捕获事件数目。

当检测到捕获事件时，定时器输入捕获 x (x 为 A 或 B) 中断标志—TICF<sub>2x</sub> (TIFR2.1 或 TIFR2.0) 置位。如果 EA 和定时器输入捕获 x 使能位—TICIE<sub>2x</sub> (TICR2.1 或 TICR2.0) 以及 IEN1 中的 ECCU 位此时都已置位，程序计数器将指向相应的中断向量。中断标志必须通过写入“0”将其手动清零。

当读取输入捕获寄存器时必须先读取 ICR<sub>xL</sub>。当读 ICR<sub>xL</sub> 时，捕获寄存器高字节的内容传送到映像寄存器。当读取 ICR<sub>xH</sub> 时，实际读取的是映像寄存器的内容。(如果一个对 ICR<sub>xL</sub> 的读操作之后跟着另一个对 ICR<sub>xL</sub> 的读操作，而中间没有对 ICR<sub>xH</sub> 的读操作时，捕获寄存器高字节的新值(来自后一次对 ICR<sub>xL</sub> 的读操作)将位于映像寄存器中。)

表 10 用于输入捕获的事件延迟计数器

ICEC <sub>x2</sub> (CCCR <sub>x</sub> .7)	ICEC <sub>x1</sub> (CCCR <sub>x</sub> .6)	ICEC <sub>x0</sub> (CCCR <sub>x</sub> .5)	延迟 (边沿的数目)
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	7
1	1	1	15

### PWM 操作

PWM 操作有两种主要的模式：对称和非对称。这两种模式通过向 TMOD21:TMOD20 写入 10H 或 11H 进行选择。见“基本定时器操作”一节。

在非对称 PWM 操作中，CCU 定时器按照递减计数模式操作，与 TDIR2 的设定无关。该情况下，TDIR2 的值读出总是为 1。

在对称模式中，定时器交替递增/递减计数并且 TDIR2 的值无效。该模式与基本定时器操作的主要区别在于比较模块的操作。该模块在 PWM 模式中用于 PWM 波形的产生。表 11 所示为 PWM 模式中比较管脚的状态。

用户必须将输出比较管脚配置为输出状态以使能 PWM 输出。使用基本定时器操作时，当 PWM (比较) 管脚连接到比较逻辑，它们的逻辑状态保持不变。但由于 FCO 位用于保持停止的值，因此只有一个比较事件才能改变管脚的状态。

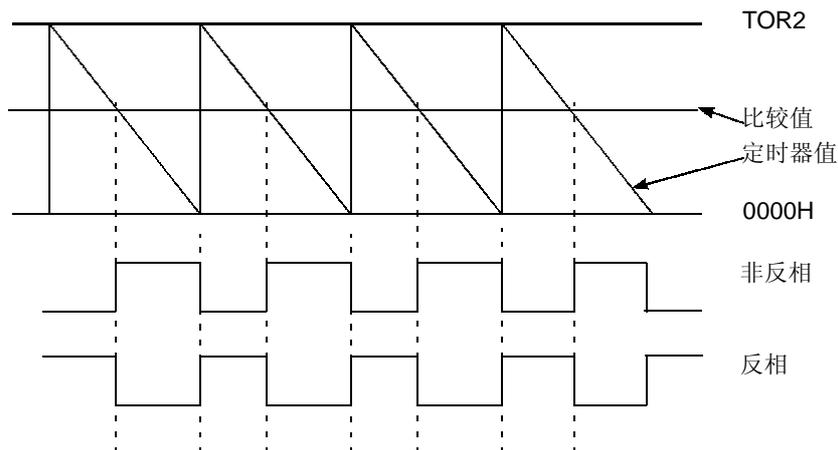


图 28 非对称 PWM，递减计数

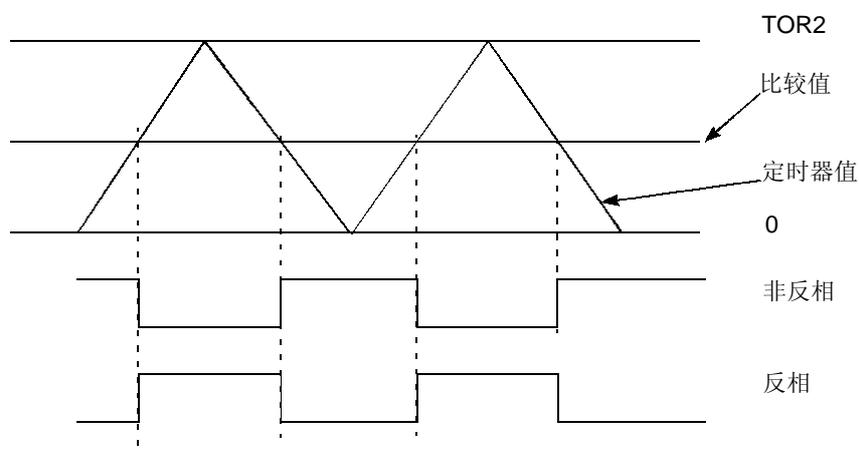


图 29 对称 PWM

当计数器在顶端改变方向时，CCU 定时器溢出中断标志置位。如果 TOR 包含 01FFH，CCU 定时器将会计数：..., 01FEH, 01FFH, 01FEH, ...。该标志在 TOR 变为 TOR-1 之后的计数器周期内置位。

当定时器在底部改变方向时，定时器将会计数：..., 0001H, 0000H, 0001H, ...。CCU 定时器溢出中断标志在从 0001H 变为 0000H 之后的 CCUCLK 周期内置位。

TCR20 中的 TDIR2 位的状态反映了当前计数的方向。在对称模式中对该位的写操作无效。

#### 交替输出模式

在非对称模式中，用户可以将 PWM 通道 A/B 和 C/D 编程为交替输出对用于桥驱动控制。通过置位 TCR20 中的 ALTAB 或 ALTCD 位，这些 PWM 通道的输出就在每个计数器周期交替选通。如图 30 所示：

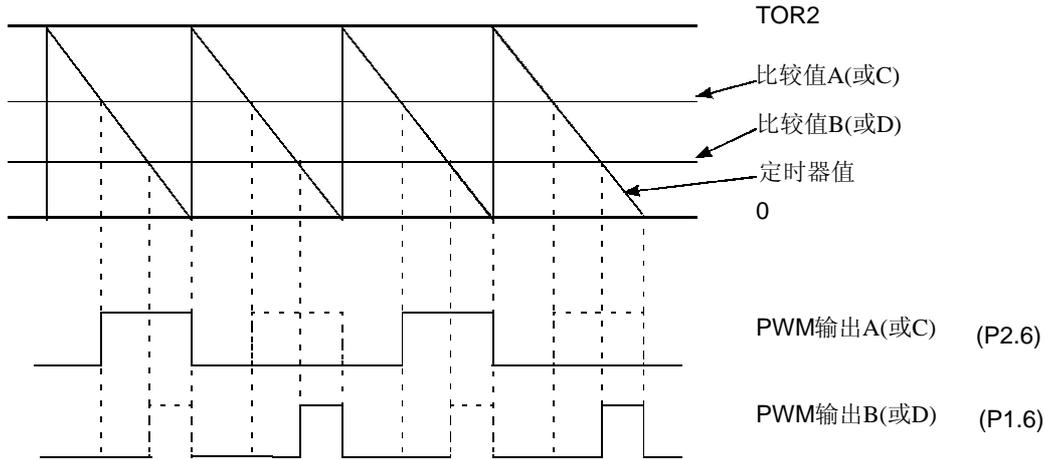


图 30 交替输出模式

表 11 输出比较管脚状态

OCMx1 <sup>1</sup> (CCCRx.1)	OCMx0 <sup>1</sup> (CCCRx.0)	输出比较管脚状态		
		基本定时器模式	非对称 PWM	对称 PWM
0	0	输出比较禁止。这是上电后默认的状态，管脚配置为输入		
0	1	当比较处于操作中时置位。比较发生匹配时 <sup>2</sup> 清零。	非反转的 PWM。在比较匹配时 <sup>2</sup> 置位；在 CCU 定时器向下溢出时 <sup>2</sup> 清零	非反转的 PWM。递增计数发生比较匹配时 <sup>2</sup> 清零；递减计数发生匹配时 <sup>2</sup> 置位。
1	0		反转的 PWM。比较匹配时 <sup>2</sup> 清零；CCU 定时器向下溢出时 <sup>2</sup> 置位。	反转的 PWM。递增计数发生比较匹配时 <sup>2</sup> 置位；递减计数发生匹配时 <sup>2</sup> 清零。
1	1	在比较匹配时 <sup>2</sup> 翻转		

注：1. x=A, B, C, D      2. 表示事件发生之后的 CCUCLK 周期中

#### 同步的 PWM 寄存器更新

对 OCRx 寄存器进行写操作时，一个内建的机制可以确保写入的值不会在 PWM 脉冲的中间更新。否则导致产生一个奇数长度的脉冲。和基本定时器操作模式一样，当写入寄存器时，写入的值放入两个映像寄存器。当值已写入后，将 TCOU2 置位会导致映像寄存器的内容在下次 CCU 定时器溢出时被更新。如果 OCRxH 和/或 OCRxL 在它们的值更新之前被读出，那么读出的是最近一次写入的值。

#### 停止

PWM 的停止功能通过置位 TCR20 中的 HLTEN 位使能。当停止功能使能时，输入捕获 A 管脚使能的捕获事件将立即停止 PWM 脚上所有的活动并将它们设置成由 FCOx 位所定义的预设状态。在 PWM 模式中，CCCRx 寄存器中的 FCOx 位保持停止时管脚所强制进入的值。该设定值可以被读回。即使该附加功能使能，捕获功能和中断仍然执行正常的操作。当 PWM 单元停止时，定时器仍然正常运行。TCR20 中的 HLTRN 位将置位以指示停止的发生。为了重新激活 PWM，用户必须清零 HLTRN 位。通过置位 HLTRN，用户可强制 PWM 单元进入停止状态。

#### PLL 的操作

PWM 模块还包括一个 PLL（锁相环）用于产生介于 16MHz 到 32MHz 的 CCUCLK 频率。在该频率下，PWM 模块提供超声 PWM 频率。10 位分辨率规定了晶振的频率为 1MHz 或更高。PLL 的输入信号为 0.5—1MHz，而输出信号为输入信号频率的 32 倍。该输出信号作为定时器的时钟。用户必须将 PCLK 分频，分频系数为 1—16。该分频器位于寄存器 TCR21 中。PLL 频率可表达如下：

PLL 频率 = PCLK / (N+1)

此处：N 为 PLLDV3:0 的值

由于 N 的范围为 0~15，CCLK 频率的范围可以从 PCLK 到 PCLK/16。

TCR21 地址：F9H								
不可位寻址	7	6	5	4	3	2	1	0
复位源：任何复位	TCOU2	-	-	-	PLLDV.3	PLLDV.2	PLLDV.1	PLLDV.0
复位值：0xxx0000B								
位	符号	功能						
TCR21.7	TCOU2	在基本定时器模式中，向 TCOU2 写入 1 将将会导致立即执行锁存，并且读取 TCOU2 的值将一直得到 0；在 PWM 模式中，置位 TCOU2 将使映像寄存器的内容在下次 CCU 定时器溢出时更新。当锁存被挂起时，TCOU2 的值读出为 1；当锁存被执行后，TCOU2 的值将变为 0。TCOU2 还控制所有输出比较寄存器 OCRAx, OCRBx 和 OCRCx 的锁存。						
TCR21.6-4	-	保留将来之用。用户程序不要将其置 1。						
TCR21.3-0	PLLDV.3-0	PLL 分频器						

图 31 CCU 控制寄存器 1

置位 TCR20 中的 PLEN 位启动 PLL。置位 PLEN 后，直到 PLL 锁定时 PLEN 的读出值为 1。此时，PWM 单元准备就绪，定时器也可使能。推荐使用下面的启动顺序：

1. 在启动定时器之前设置 PWM 模块。
2. 计算正确的分频系数，以使 PLL 接收 500KHz 到 1MHz 的输入时钟信号。将此正确的值写入 PLLDV
3. 置位 PLEN。等待该位读出为 1。
4. 将一个值写入 TMOD21, TMOD20 启动定时器

当定时器由 PLL 驱动时，定时器异步于微控制器的其余部分。有下面一些限制：

- 在异步模式下，用户最后不要对定时器进行读或写。结果将是不可预知的。
- 中断和标志不同步。对事件的识别将会延迟一些 CCLK 周期（对于中断和读操作）

#### CCU 中断结构

CCU 有 7 个独立的中断源：定时器溢出，输入捕获部件 A/B 的捕获输入事件以及输出比较部件 A 到 D 的比较匹配事件。CCU 中断服务程序使用同一个中断向量，如果中断同时发生，将会出现一个难题。为了解决这种情况，在 TIFR2 寄存器中设置了这 7 个中断的优先级编码功能（每一位都与 TIFR2 中相应的中断使能位相与）。固定的优先级顺序如下按照从高到低排列：

- TOIF2
- TICF2A
- TICF2B
- TOCF2A
- TOCF2B
- TOCF2C
- TOCF2D

当 TIFR2 寄存器中的任何一个中断标志置位时，在 CCU 定时器中断状态编码寄存器 TISE2 中的 3 个优先级编码器输出位（见图 32）可用。需要注意的是，为了产生中断，单个中断源的使能位和 CCU 全局中断使能位(EA)都必须置位。

CCU 中断服务程序可以这样处理:

1. 从 TISE2 寄存器读出优先级编码值以确定要处理的中断源。
2. 在相应当前事件（最高优先级）之后，将 TIFR2 寄存器中相应的中断标志位清零
3. 读 TISE2 寄存器。如果优先级编码中断源为“000”，表示所有的 CCU 中断都已处理完，执行中断返回，否则，返回到第 2 步处理下一个中断。

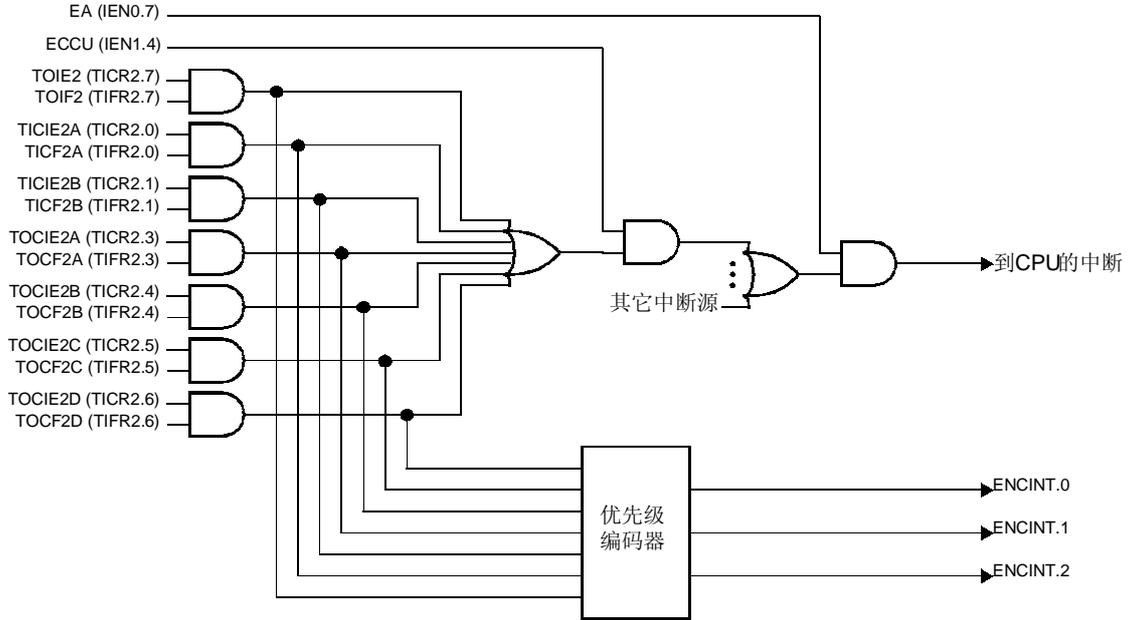


图 32 捕获/比较单元中断

TISE2 地址: DEH									
不可位寻址		7	6	5	4	3	2	1	0
复位源: 任何复位		-	-	-	-	-	ENCINT.2	ENCINT.1	ENCINT.0
复位值: xxxxx000B									
位	符号	功能							
TISE2.7-3	-	保留将来之用。用户程序不要将其置 1。							
TISE2.2	ENCINT.2-0	CCU 中断编码输出。当多个中断发生时，CCU 中断标志寄存器 (TIFR2) 中的多个中断标志置位。可以读出编码器的输出值以确定所要服务的中断。当处理完一个中断时，用户必须清零相应的中断标志位。详见 TIFR2 寄存器的描述。							
	<u>ENCINT.2-0</u>	中断源							
	000	无中断挂起							
	001	输出比较事件 D 中断 (最低优先级)							
	010	输出比较事件 C 中断							
	011	输出比较事件 B 中断							
	100	输出比较事件 A 中断							
	101	输入捕获事件 B 中断							
	110	输入捕获事件 A 中断							

图 33 CCU 中断状态编码寄存器

TIFR2 地址: E9H							
不可位寻址	7	6	5	4	3	2	1 0
复位源: 任何复位	TOIF2	TOCF2D	TOCF2C	TOCF2B	TOCF2A	-	TICF2B TICF2B
复位值: 00000x00B							
位	符号	功能					
TIFR2.7	TOIF2	CCU 定时器溢出中断标志位。CCU 定时器溢出时由硬件置位。由软件清零。					
TIFR2.6	TOCF2D	输出比较通道 D 中断标志位。当 TH2:TL2 与 OCRHD:OCRLD 的内容匹配时, 由硬件置位。要产生该中断必须使能比较通道 D。如果 IEN0 中的 EA, IEN1 中的 ECUU 以及 TOCIE2D 都置位, 程序计数器将会指向对应的中断向量。通过软件清零该标志。					
TIFR2.5	TOCF2C	输出比较通道 C 中断标志位。当 TH2:TL2 与 OCRHC:OCRLC 的内容匹配时, 由硬件置位。要产生该中断必须使能比较通道 C。如果 IEN0 中的 EA, IEN1 中的 ECUU 以及 TOCIE2C 都置位, 程序计数器将会指向对应的中断向量。通过软件清零该标志。					
TIFR2.4	TOCF2B	输出比较通道 B 中断标志位。当 TH2:TL2 与 OCRHB:OCRLB 的内容匹配时, 由硬件置位。要产生该中断必须使能比较通道 B。如果 IEN0 中的 EA, IEN1 中的 ECUU 以及 TOCIE2B 都置位, 程序计数器将会指向对应的中断向量。通过软件清零该标志。					
TIFR2.3	TOCF2A	输出比较通道 A 中断标志位。当 TH2:TL2 与 OCRHA:OCRLA 的内容匹配时, 由硬件置位。要产生该中断必须使能比较通道 A。如果 IEN0 中的 EA, IEN1 中的 ECUU 以及 TOCIE2A 都置位, 程序计数器将会指向对应的中断向量。通过软件清零该标志。					
TIFR2.2	-	保留将来之用。用户程序不要将其置 1。					
TIFR2.1	TICF2B	输入捕获通道 B 中断标志位。当检测到输入捕获事件时, 由硬件置位。通过软件清零该标志。					
TIFR2.0	TICF2A	输入捕获通道 A 中断标志位。当检测到输入捕获事件时, 由硬件置位。通过软件清零该标志。					

图 34 CCU 中断标志寄存器

TICR2 地址: E9H							
不可位寻址	7	6	5	4	3	2	1 0
复位源: 任何复位	TOIE2	TOCIE2D	TOCIE2C	TOCIE2B	TOCIE2A	-	TICIE2B TICIE2B
复位值: 0000x00B							
位	符号	功能					
TICR2.7	TOIE2	CCU 定时器溢出中断使能位。					
TICR2.6	TOCIE2D	输出比较通道 D 中断使能位。如果 EA 和该位都置位, 当比较通道使能且 TH2:TL2 与 OCRHD:OCRLD 的内容匹配时, 程序计数器将会指向对应的中断向量。					
TICR2.5	TOCIE2C	输出比较通道 C 中断使能位。如果 EA 和该位都置位, 当比较通道使能且 TH2:TL2 与 OCRHC:OCRLC 的内容匹配时, 程序计数器将会指向对应的中断向量。					
TICR2.4	TOCIE2B	输出比较通道 B 中断使能位。如果 EA 和该位都置位, 当比较通道使能且 TH2:TL2 与 OCRHB:OCRLB 的内容匹配时, 程序计数器将会指向对应的中断向量。					
TICR2.3	TOCIE2A	输出比较通道 A 中断使能位。如果 EA 和该位都置位, 当比较通道使能且 TH2:TL2 与 OCRHA:OCRLA 的内容匹配时, 程序计数器将会指向对应的中断向量。					
TICR2.2	-	保留将来之用。用户程序不要将其置 1。					
TICR2.1	TICIE2B	输入捕获通道 B 中断使能位。如果 EA 和该位都置位, 当检测到输入捕获事件时, 程序计数器将会指向对应的中断向量。					
TICR2.0	TICIE2A	输入捕获通道 A 中断使能位。如果 EA 和该位都置位, 当检测到输入捕获事件时, 程序计数器将会指向对应的中断向量。					

图 35 CCU 中断控制寄存器

## UART

LPC932 具有一个增强型的 UART。它和传统的 80C51 UART 兼容, 但有一点除外, 即定时器 2 的溢出不能用于产生波特率。LPC932 还带有一个独立的波特率发生器。波特率可以选择由振荡器 (由一个常数频率), 定时器 1 溢出或者独立的波特率发生器产生。除了产生波特率以外, 在标准 80C51 UART 基础上还增加了帧错误检测、间隔检测、自动地址识别、可选的双缓冲以及几个中断选项。

UART 具有 4 种操作模式:

### 模式 0

串行数据通过 RxD 进出。TxD 输出移位时钟。每次发送或接收都为 8 位, LSB (最低位) 在前。波特率固定为 CPU 时钟频率的 1/16。

### 模式 1

TxD 脚发送, RxD 脚接收, 每次数据为 10 位: 1 个起始位 (逻辑 0), 8 个数据位 (LSB 在前) 以及 1 个停止位 (逻辑 1)。当接收数据时, 停止位保存在 SCON 中的 RB8。该模式的波特率可变, 由定时器 1 溢出速率或波特率发生器决定 (详见“波特率发生器及其选择”一节)。

### 模式 2

TxD 脚发送, RxD 脚接收, 每次数据为 11 位: 1 个起始位 (逻辑 0), 8 个数据位 (LSB 在前), 一个可编程第 9 位数据及 1 个停止位 (逻辑 1)。发送数据时, 第 9 个数据位 (SCON 中的 TB8 位) 可置为 0 或 1。例如可将奇偶位 (PSW 内 P 位) 放入 TB8。接收时, 第 9 位数据存入 SCON 的 RB8 位, 而停止位不会

被保存。波特率可编程为 CPU 时钟频率的 1/16 或 1/32，由 PCON 内 SMOD1 位决定。

### 模式 3

TxD 脚发送，RxD 脚接收，每次数据为 11 位：1 个起始位（逻辑 0），8 个数据位（LSB 在前），1 个可编程的第 9 位数据及 1 个停止位。实际上，模式 3 除了波特率外其它均与模式 2 相同。模式 3 的波特率可变并由定时器 1 溢出率或波特率发生器决定（详见“波特率发生器及其选择”一节）。

在上述 4 种模式中，发送过程是以一条写 SBUF 作为目标寄存器的指令开始的。在模式 0 中接收过程通过设置 R1=0 及 REN=1 启动，在其它模式中，如果 REN=1 则通过接收起始位来启动。

### SFR 的位置

UART 的 SFR 位于以下位置：

表 12 用于 UART 的 SFR/扩充 SFR 的位置

寄存器	描述	SFR 位置
PCON	电源控制	87H
SCON	串口 (UART) 控制	98H
SBUF	串口 (UART) 数据缓冲器	99H
SADDR	串口 (UART) 地址	A9H
SADEN	串口 (UART) 地址使能	B9H
SSTAT	串口 (UART) 状态	BAH
BRGR1	波特率发生器速率高字节	BFH
BRGR0	波特率发生器速率低字节	BEH
BRGCON	波特率发生器控制	BDH

### 波特率发生器及选择

LPC932 的增强型 UART 具有一个独立的波特率发生器。波特率取决于对 BRGR1 和 BRGR0 预先编程的值。UART 也可使用定时器 1 或者波特率发生器的输出（由 BRGCON.2-1 决定）。需要注意的是，如果 SMOD1(PCON.7)置位，定时器 T1 被 2 分频。独立的波特率发生器使用 CCLK 作为时钟源（见图 37）

### 更新 BRGR1 和 BRGR0 寄存器

波特率寄存器 BRGR1 和 BRGR0 只能在波特率发生器被禁止时 (BRGCON 寄存器中的 BRGEN 位为 0) 才能写入。这样避免了向波特率发生器装入临时的值（警告：当 BRGEN=1 时写入 BRGR0 或 BRGR1，结果将是不可预知的）。

表 13 UART 的波特率产生

SCON.7 (SM0)	SCON.6 (SM1)	PCON.7 (SMOD1)	BRGCON.1 (SBRGS)	接收/发送波特率
0	0	X	X	fosc/16
0	1	0	0	T1 溢出速率/32
		1	0	T1 溢出速率/16
		X	1	fosc/((BRGR1, BRGR0)+16)
1	0	0	X	fosc/32
		1	X	fosc/16
1	1	0	0	T1 溢出速率/32
		1	0	T1 溢出速率/16
		X	1	fosc/((BRGR1, BRGR0)+16)

BRGCON 地址: BDH	
不可位寻址	7 6 5 4 3 2 1 0
复位源: 任何复位	- - - - - SBRGS BRGEN
复位值: xxxxxx00B	
位	符号 功能
BRGCON.7-2	- 保留将来之用。用户程序不要将其置 1。
BRGCON.1	SBRGS 选择波特率发生器用于产生 UART 模式 1 和 3 的波特率 (见表 13)
BRGCON.0	BRGEN 0: 禁止波特率发生器; 1: 使能波特率发生器。只有当 BRGEN 为 0 时, 才可对寄存器 BRGR1 和 BRGR0 进行写操作。

图 36 BRGCON 寄存器

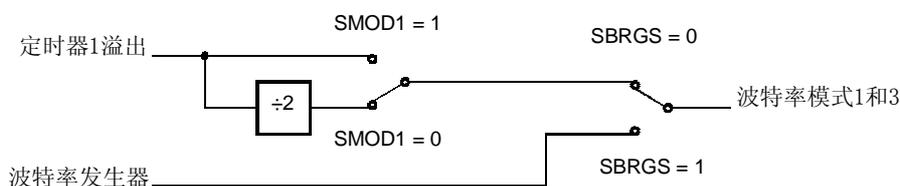


图 37 UART 波特率的产生 (模式 1, 3)

### 帧错误

当检测到停止位为 0 时产生帧错误。帧错误在状态寄存器 SSTAT 中报告。此外, 如果 SMOD0(PCON.6) 为 1, SCON.7 单独作为帧错误位。如果 SMOD0 为 0, SCON.7 作为 SM0。建议在 SMOD0 为 0 时对 SM0 和 SM1(SCON.7-6)进行设置。

### 间隔检测

间隔检测在状态寄存器 SSTAT 中报告。当连续检测到 11 个位都为低电平位时, 则认为检测到一个间隔。由于一个间隔条件同样满足帧错误条件, 因此检测到间隔时也会报告帧错误。当检测到一个间隔条件时, UART 将进入空闲状态并一直保持到接收到一个停止位为止。间隔检测可用于对器件复位并通过置位 EBRR(AUXR1.6)使器件进入 ISP 模式。

SCON 地址: 98H									
可位寻址		7	6	5	4	3	2	1	0
复位源: 任何复位		SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI
复位值: 00000000									
位	符号	功能							
SCON.7	SM0/FE	该位的用途由 PCON 寄存器中的 SMOD0 决定。如果 SMOD0=0, 该位作为 SM0, 和 SM1 一起定义串口模式; 如果 SMOD0=1, 该位作为 FE(帧错误)。当检测到一个无效的停止位时, FE 被接收器置位。一旦置位后, 该位不能由有效帧清零, 只能通过软件清零。(注: UART 模式位 SM0 和 SM1 应当在 SMOD0=0 时进行设置——在任何复位后默认的状态)							
SCON.6	SM1	和 SM0 定义串行口操作模式 (见下表)							
	SM0, SM1	UART 模式	波特率						
	0 0	0: 同步移位寄存器	CCLK/16 (复位后的默认模式)						
	0 1	1: 8 位 UART	可变(见表 13)						
	1 0	2: 9 位 UART	CCLK/32 或 CCLK/16						
SCON.5	1 1	3: 9 位 UART	可变(见表 13)						
SCON.4	SM2	使能模式 2 和 3 中的多机通信功能。在模式 2 或 3 中, 如果 SM2=1 而且接收到的第 9 位数据 (RB8) 为 0 时, 则 RI 不会被激活。在模式 0 中, SM2 应当为 0。在模式 1 中, SM2 必须为 0。							
SCON.3	REN	使能串行接收。由软件置位以使能接收。软件清零则禁止接收。							
SCON.2	TB8	模式 2 和 3 中将要发送的第 9 位数据, 可以根据需要由软件置位或清零。							
SCON.1	RB8	模式 2 和 3 中接收的第 9 位数据, 在模式 1 中 (SM2 必须为 0), RB8 是接收到的停止位。在模式 0 中, RB8 未定义。							
SCON.0	TI	发送中断标志。模式 0 中, 在第 8 位数据发送结束时由硬件置位。在其它任何串行发送模式中, 在发送停止位 (见 SSTAT 寄存器中 INTLO 位的描述) 时由硬件置位。必须通过软件清零。							
	RI	接收中断标志, 模式 0 中, 第 8 位接收完成时由硬件置位。在模式 1 中, 在接收停止位的中间时刻由硬件置位。在模式 2 或 3 中, 如果 SMOD0=0, 在接近第 9 位的中间时刻置位; 如果 SMOD0=1, 在接近停止位的中间时刻置位 (例外见 SM2-SCON.5)。必须通过软件清零。							

图 38 串口控制寄存器(SCON)

SSTAT 地址: BAH		7	6	5	4	3	2	1	0
不可位寻址		DBMOD	INTLO	CIDIS	DBISEL	FE	BR	OE	STINT
复位源: 任何复位									
复位值: 00000000B									
位	符号	功能							
SSTAT.7	DBMOD	双缓冲模式。0: 双缓冲禁止; 1: 双缓冲使能。UART 模式 0 时必须为 0。为了和现有的 80C51 器件兼容, 该位复位时为 0 禁止双缓冲。							
SSTAT.6	INTLO	发送产生中断的位置。0: 在停止位的开始产生 Tx 中断; 1: 在停止位的结束产生 Tx 中断。模式 0 时必须为 0。在使用单缓冲时, 如果在停止位的结束产生 Tx 中断, 在下一个起始位之前可能存在一个间隙。							
SSTAT.5	CIDIS	0: 组合的 Tx/Rx 中断; 1: 各自独立的 Rx 和 Tx 中断。该位复位时为 0 选择组合中断。							
SSTAT.4	DBISEL	双缓冲发送中断选择一只在双缓冲使能时使用。该位在双缓冲使能时控制中断产生的次数。当该位置位时, 每向 SBUF 写入一个字符就产生一个发送中断, 另外在最后一个字符的停止位开始 (INTLO=1) 或结束 (INTLO=0) 时再产生一次发送中断 (即缓冲区已空)。最后这次中断可用于指示所有发送操作已完成。当该位为 0 时, 每向 SBUF 写入一个字符只产生一次发送中断。另外在双缓冲被禁止时, 该位必须为 0。 注: 除了第一个写入的字符 (SBUF 为空时) 之外, 所有发送中断的位置都由 INTLO 决定。当写入第一个字符时, 在 SBUF 被写入后立即产生发送中断。							
SSTAT.3	FE	当接收器在帧结束时没有收到有效的停止位时, 该标志置位。通过软件清零。							
SSTAT.2	BR	如果接收到的一个字符所有位 (包括停止位) 都为 0, 间隔检测标志置位。在模式 1 中的第 8 位和模式 2, 3 的第 9 位时启动间隔检测。间隔检测独立于 UART 工作, 并提供可供用户查询的间隔检测状态位。该位通过软件清零。							
SSTAT.1	OE	当缓冲器仍然为满时 (在软件读出缓冲器中的前一个字符之前), 如果接收缓冲器接收到一个新的字符 (也就是说, 当接收完一个新字节的第 8 位时 RI 仍然是置位的状态), 那么超限错误标志 OE 将置位。该位通过软件清零。							
SSTAT.0	STINT	状态中断使能: 0: FE, BR, OE 不能引起任何中断。(注: FE, BR 或 OE 位通常和 RI 一起使用, 不管 STINT 的状态如何, RI 都将产生中断。) 1: FE, BR 和 OE 可产生中断。CIDIS=1 时, 与 RI 共用中断; CIDIS=0 时, 中断与 TI/RI 的组合共用。 注: 如果 EBRR(AUXR1.6)置位, BR 可导致间隔检测复位。							

图 39 串口状态寄存器 (SSTAT)

#### 更多关于 UART 模式 0 的信息

在模式 0 中, 执行写 SBUF 的指令将会启动发送。在发送结束时, TI(SCON.1)置位, 该位必须由软件清零。在该模式中必须禁止双缓冲。

通过清零 RI(SCON.0)启动接收。开始同步串行传送并且在传送结束时 RI 将再次置位。当 RI 清零时, 开始下一个字符的接收, 相关时序见图 40。

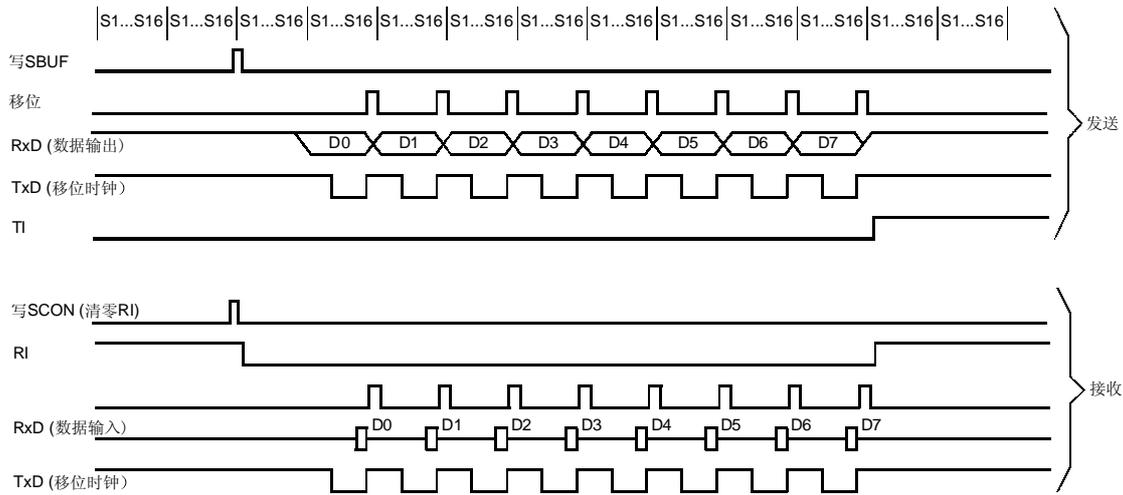


图 40 串口模式 0 (双缓冲必须禁止)

**更多关于 UART 模式 1 的信息**

接收在检测到 RxD 端电平负跳变时启动，CPU 对 RxD 不断采样，采速率为波特率的 16 倍。当检测到负跳变时，16 分频计数器立即复位以将计时器的翻转值调整到与输入数据位时间的边界一致。

计数器的 16 个状态将每个位时间分为 16 份。在每个位时间的第 7、8、9 计数状态时，位检测器对 RxD 端的值进行采样。取值为三个采样值中取多数（至少 2 个）作为读入值，这样做是为了抑制噪声。如果在第一个位时间所接收的位不为 0，说明它不是一帧数据的起始位，接收电路复位并等待另一个负跳变的到来。这样可以防止错误的起始位。如果起始位被证明是有效的，则被移入输入移位寄存器，并开始接收这一帧剩余的位。

当且仅当产生最后一位移位脉冲时满足下列条件：RI=0 以及 SM2=0 或接收到的停止位=1，数据才会被装入 SBUF 和 RB8，并置位 RI。

上述两个条件中的任何一个不满足，所接收到的数据帧都会丢失，并且无法恢复。两个条件都满足时，停止位就进入 RB8，而 8 位数据则进入 SBUF，并且 RI 置位。

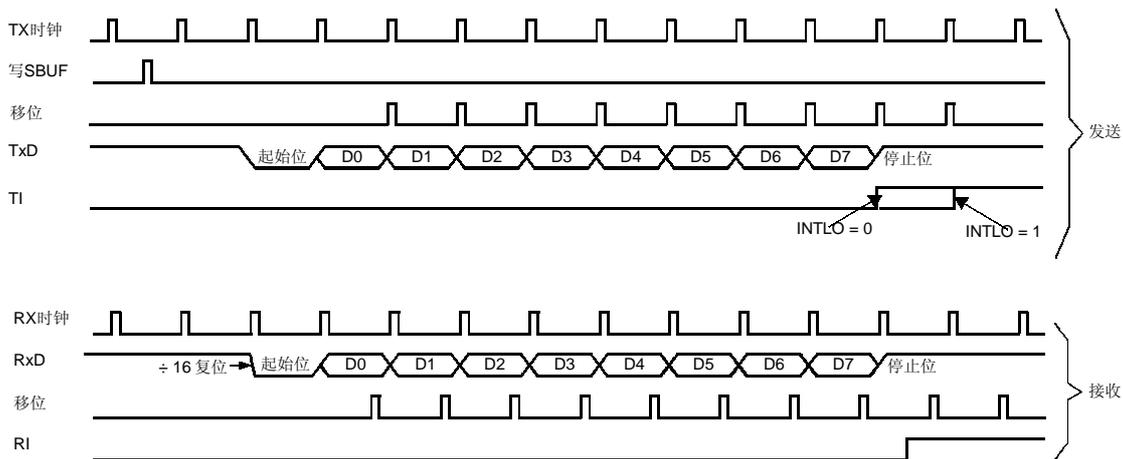


图 41 串口模式 1 (只显示单发送缓冲的状况)

**更多关于模式 2 和模式 3 的信息**

接收的工作方式与模式 1 相同。

当且仅当产生最后一位移位脉冲时满足下列条件：(a) RI=0，以及(b) SM2=0 或接收到的第 9 位数据=1，数据才会被装入 SBUF 和 RB8，并置位 RI。

上述两个条件中的任何一个不满足，所接收到的数据帧都会丢失，并且无法恢复。两个条件都满足时，接收到的第 9 位数据就进入 RB8，而前 8 位数据则进入 SBUF，并且 RI 置位。

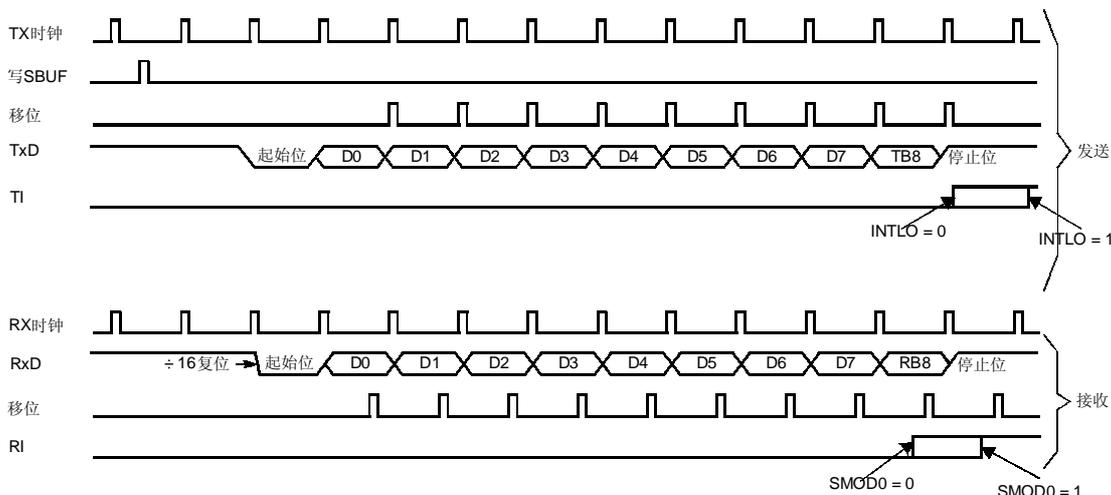


图 42 串口模式 2 或 3（只显示单发送缓冲的状况）

如果在模式 2 和 3 中 SM2=1，RI 和 FE 的状态如下表所示。

表 14 模式 2 和 3 中 SM2=1 时 RI 和 FE 的状态

模式	PCON.6 (SMOD0)	RB8	RI	FE
2	0	0	当 RB8=0 时无 RI	发生在停止位
		1	和图 42 相似，RI 在 RB8 时发生，FE 的前一位	发生在停止位
3	1	0	当 RB8=0 时无 RI	不会发生
		1	和图 42 相似，RI 在 RB8 时发生，FE 的前一位	发生在停止位

### 双缓冲

UART 具有一个发送双缓冲器，这就允许第一个字符正在发送的时候向 SBUF 写入第二个字符。只要下个字符在前一个字符的起始位和停止位之间写入 SBUF，那么发送的字符串中两个字符之间就只有一个停止位。

双缓冲可以被禁止。当禁止时 (DBMOD=0)，UART 和传统的 80C51 UART 兼容。如果使能该功能，UART 允许在前一个数据移位发送的过程中向 SBUF 写入新数据。

### 不同模式中的双缓冲

只有在模式 1，2 和 3 中才可以使能双缓冲。当处于模式 0 时，必须禁止双缓冲 (DBMOD=0)。

双缓冲使能时的发送中断 (模式 1，2 和 3)

与传统的 UART 不同的是，当使能双缓冲时，Tx 中断发生在双缓冲器准备接收新数据的时候。发送的过程是这样的 (假设为 8 个数据位)：

1. 双缓冲器初始化为空。
2. CPU 将数据写入 SBUF
3. SBUF 的数据装入移位寄存器并立即产生一个 Tx 中断。
4. 如果有后续的数据，执行第 6 步，否则执行第 5 步。
5. 如果没有后续的数据，那么：
  - 如果 DBISEL=0，不再产生任何中断。

- 如果 DBISEL=1 且 INTLO=0, 在移位寄存器中当前数据（也是最后的数据）停止位的开始产生 Tx 中断。
- 如果 DBISEL=1 且 INTLO=1, 在移位寄存器中当前数据（也是最后的数据）停止位的结束产生 Tx 中断。

6. 如果有后续的数据, CPU 再次将数据写入 SBUF。那么:

- 如果 INTLO=0, 装入新数据并在移位寄存器中当前数据停止位的开始产生 Tx 中断。
- 如果 INTLO=1, 装入新数据并在移位寄存器中当前数据停止位的结束产生 Tx 中断。

跳到第 3 步。

注: 在发送上一个数据的停止位时, 如果 DBISEL=1, 而且 CPU 正将数据写入 SBUF, 这样就会出现一个不确定的状况。即 UART 是否知道还有后续的数据。

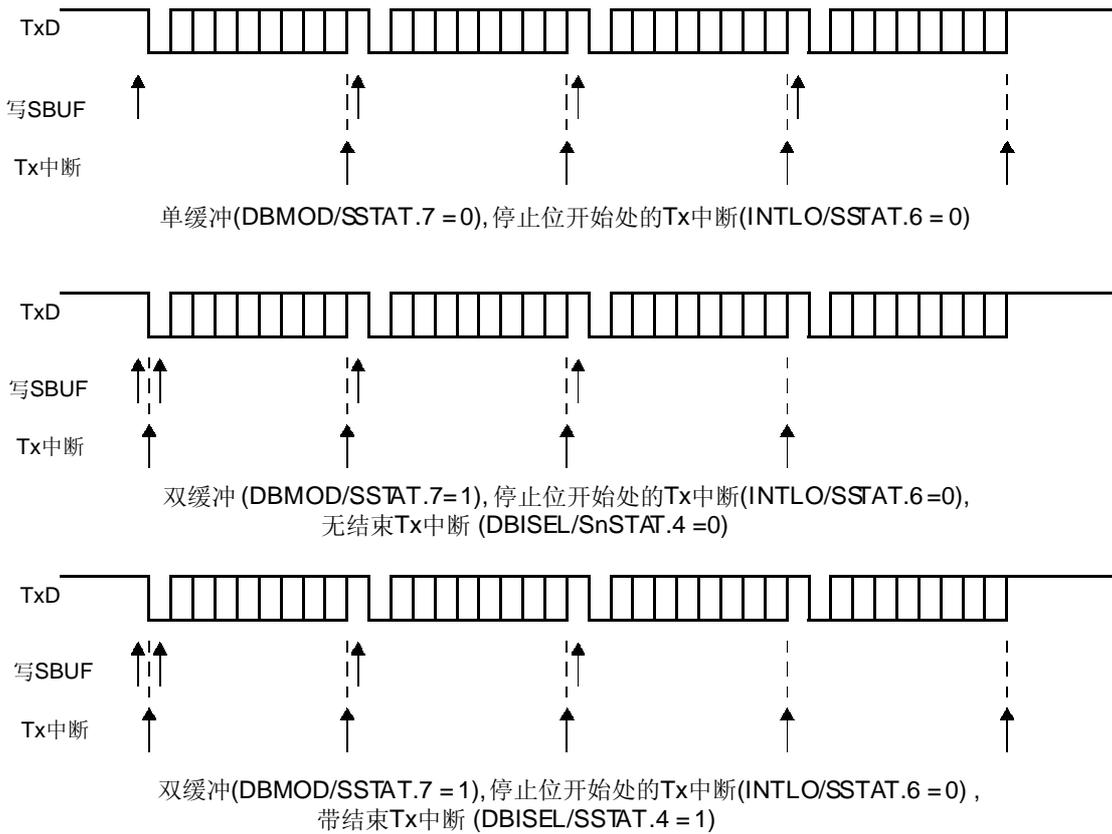


图 43 带双缓冲的发送和不带双缓冲的发送

#### 双缓冲中的第 9 位数据（模式 1, 2 和 3）

如果双缓冲被禁止 (DBMOD=0), 对 TB8 的写操作可以在写入 SBUF 之前或之后进行, 只要在第 9 位数据被移出之前将其更新即可。在该移出（通过 Tx 中断指示）之前不要改变 TB8。

如果双缓冲使能, TB8 必须在写 SBUF 之前更新, 因为 TB8 将和 SBUF 的数据一起双缓冲。上一节所描述的发送操作就变成下面这样:

1. 双缓冲器初始化为空。
2. CPU 对 TB8 进行写操作。
3. CPU 将数据写入 SBUF。
4. SBUF/TB8 的数据装入移位寄存器并立即产生一个 Tx 中断。
5. 如果没有后续的数据, 执行第 7 步, 否则执行第 6 步。
6. 如果没有后续的数据, 那么:

- 如果 DBISEL=0, 不再产生任何中断。
- 如果 DBISEL=1 且 INTLO=0, 在移位寄存器中当前数据 (也是最后的数据) 停止位发送的开始产生 Tx 中断。
- 如果 DBISEL=1 且 INTLO=1, 在移位寄存器中当前数据 (也是最后的数据) 停止位发送的结束产生 Tx 中断。
- 7. 如果有后续的数据, CPU 再次写 TB8.
- 8. CPU 再次将数据写入 SBUF. 那么:
  - 如果 INTLO=0, 装入新数据并在移位寄存器中当前数据停止位发送的开始产生 Tx 中断。
  - 如果 INTLO=1, 装入新数据并在移位寄存器中当前数据停止位发送的结束产生 Tx 中断。

跳到第 4 步。

注: 如果 DBISEL=1, 而且当 CPU 写入 SBUF 的同时, 上一个数据的停止位移出, 就会出现一个不确定的状况。

### 多机通信

UART 模式 2 及模式 3 有一个专门的应用领域即多机通信。在这些模式时, 发送及接收均为 9 位数据。接收时第 9 位数据存入 RB8。UART 可编程为: 接收到停止位时, 仅当 RB8=1 时串口中断才激活。可通过置位 SCON 内 SM2 位来使能这一特性。下面讲述多机系统使用该特性的一种方法:

当主机需要发送一数据块给某一台从机时, 首先发送一个地址字节以识别目标从机。地址字节与数据字节的区别在于第 9 位数据, 地址字节的第 9 位为 1, 而数据字节的第 9 位为 0。SM2=1 时, 数据字节不会使从机产生中断, 而地址字节则会使所有从机都产生中断, 这样每个从机可以检查接收到的字节并判断是否被寻址。被寻址的从机将清零 SM2 位以准备接收随后的数据字节。未被寻址的从机的 SM2 位仍为 1, 这样就忽略随后的数据继续各自工作。

注: 在模式 0 中 SM2 无效, 在模式 1 中 SM2 必须为 0。

### 自动地址识别

自动地址识别是这样一种特性, 它允许 UART 使用硬件进行比较, 从串行数据流中识别出特定的地址。这样就不必花费大量软件资源去检查每一个从串口输入的串行地址。将 SCON 内 SM2 置位可使能该特性。在 9 位 UART 模式 (模式 2 和模式 3) 下, 如果接收的字节中包含“给定”地址或“广播”地址, 接收中断标志 (RI) 将自动置位。在 9 位模式下要求第 9 个信息位为 1 以表明该信息内容是地址而非数据。

使用自动地址识别特性时, 主机通过调用给定的从机地址选择与一个 (或多个) 从机通信。使用广播地址可以联系所有的从机。有两个特殊功能寄存器用来定义从机地址: SADDR 和地址屏蔽 SADEN。SADEN 用于定义 SADDR 内哪几位需要使用而哪几位是无关位。SADEN 可以与 SADDR 逻辑“与”得出给定的地址, 主机使用该地址对每一从机进行寻址。使用给定地址可以识别多个从机而排除另外的从机。下面的例子就显示了该方案的通用性:

```
从机 0      SADDR=1100 0000
             SADEN=1111 1101
             给定地址=1100 00X0
```

```
从机 1      SADDR=1100 0000
             SADEN=1111 1110
             给定地址=1100 000X
```

上例中 SADDR 相同, 而 SADEN 的数据用于区分两个从机。从机 0 要求位 0 为 0 而忽略位 1。从机 1 则要求位 1 为 0 而忽略位 0。由于从机 1 要求位 1 必须为 0, 从机 0 唯一的地址应当是 1100 0010。由于从机 0 的位 0 必须为 1, 从机 1 唯一的地址应当为 1100 0001。而取地址 1100 0000 时两从机都可被寻址。

下例所示为选择从机 1、2 而不选从机 0:

从机 0	SADDR=1100 0000 SADEN= <u>1111 1001</u> 给定地址=1100 0XX0
从机 1	SADDR=1110 0000 SADEN= <u>1111 1010</u> 给定地址=1100 0X0X
从机 2	SADDR=1110 0000 SADEN= <u>1111 1100</u> 给定地址=1110 00XX

上述三个从地址只有低 3 位不同。从机 0 要求位 0=0, 它可通过 1110 0110 单独寻址; 从机 1 要求位 1=0, 可通过 1110 0101 单独寻址; 从机 2 要求位 2 为 0, 可通过 1110 0011 单独寻址。由于必须使地址字节的第 2 位为“1”以屏蔽从机 2, 因此使用地址 1110 0100 可选择从机 0 和 1 同时屏蔽从机 2。将 SADDR 和 SADEN 相“或”后产生每个从机的“广播”地址, 结果为零的位视为无关位。大多数情况下, 无关位被解释为 1, 这样, “广播”地址为 FFH。复位时 SADDR 和 SADEN 均为 00H, 此时产生了一个所有位都是无关位的给定地址, 即所有位都“无关”的广播地址。这样有效地禁止了自动寻址模式, 允许微处理器使用不带有上述特性的标准 UART 驱动器。

### I<sup>2</sup>C 总线接口

I<sup>2</sup>C 总线用两条线 (SDA 和 SCL) 在总线和器件之间传递信息。总线的主要特性如下:

- 主机和从机之间为双向数据传送
- 多主机总线 (无中央主机)
- 多主机同时传送时进行仲裁避免总线上数据冲突
- 串行时钟同步使得不同位速率的器件可以通过一条串行总线进行通信
- 串行时钟同步可作为握手机制, 用于挂起和恢复串行传输
- I<sup>2</sup>C 总线可用于测试和诊断

典型的 I<sup>2</sup>C 总线配置如图 44 所示。根据方向位 (R/W) 状态的不同, I<sup>2</sup>C 总线上具有两种不同类型的数据传输:

- 数据从主发送器传送到从接收器。主机发送的第一个字节为从地址, 接下来为数据字节。从机每接收一个字节就返回一个应答位。
- 数据从从发送器传送到主接收器。第一个字节 (从地址) 由主机发送, 然后从机返回一个应答位。接下来是从机向主机发送的数据字节。主机每接收一个字节会返回一个应答位, 但最后一个字节除外。接收完最后一个字节时返回一个“非应答”。主机产生所有的串行时钟脉冲以及起始和停止条件。传输以一个停止条件或一个重复的起始条件结束。由于重复的起始条件是下一次传输的开始, 因此 I<sup>2</sup>C 总线不会被释放。

LPC932 器件提供字节方式的 I<sup>2</sup>C 接口。有 4 种操作模式可供选择: 主发送器模式、主接收器模式、从发送器模式和从接收器模式。下面将会有详细的描述。

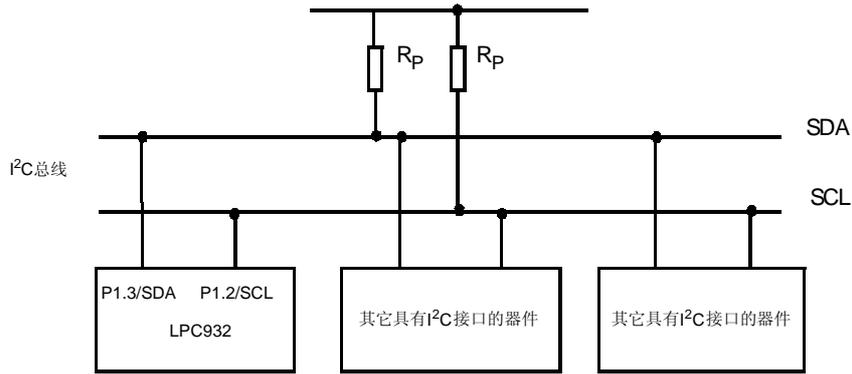


图 44 I<sup>2</sup>C 总线配置

LPC932 CPU 和 I<sup>2</sup>C 总线通过下面 6 个特殊功能寄存器实现接口: I2CON (I<sup>2</sup>C 控制寄存器)、I2DAT (数据寄存器)、I2STAT (状态寄存器)、I2ADR (地址寄存器)、I2SCLH (SCL 占空比寄存器高字节) 和 I2SCLL (SCL 占空比寄存器低字节)。

### I<sup>2</sup>C 数据寄存器

I2DAT 寄存器包含要发送的数据或刚接收到的数据。当该 8 位直接寻址寄存器没有处理移位数据时, CPU 可对其进行读写。这意味着用户只能在 SI 置位时对 I2DAT 进行访问。I2DAT 中的数据在 SI 置位时一直保持不变。I2DAT 中的数据总是从右向左移位: 发送的第一个位是最高位 (Bit7), 而在接收完一个字节后, 接收到的第一个数据位位于 I2DAT 的最高位 (MSB)。

I2DAT	地址: DAh							
不可位寻址	7	6	5	4	3	2	1	0
复位源: 任何复位	I2DAT.7	I2DAT.6	I2DAT.5	I2DAT.4	I2DAT.3	I2DAT.2	I2DAT.1	I2DAT.0
复位值: 0000000B								

图 45 I<sup>2</sup>C 数据寄存器

### I<sup>2</sup>C 从地址寄存器

CPU 可对 I2ADR 寄存器进行读或写操作。该寄存器仅在 I<sup>2</sup>C 处于从模式时才使用。在主模式中, 该寄存器的内容无效。I2ADR 的最低位 (LSB) 为通用调用位。当该位置位时对通用调用地址 (00H) 进行识别。

I2ADR	地址: DBH							
不可位寻址	7	6	5	4	3	2	1	0
复位源: 任何复位	I2ADR.6	I2ADR.5	I2ADR.4	I2ADR.3	I2ADR.2	I2ADR.1	I2ADR.0	GC
复位值: 0000000B								
位	符号	功能						
I2ADR.7, 1	I2ADR.6, 0	7 位自身从地址。当处于主模式时, 该寄存器的内容无效。						
I2ADR.0	GC	通用调用位。置位时, 对通用调用地址进行识别, 否则忽略。						

图 46 I<sup>2</sup>C 从地址寄存器

### I<sup>2</sup>C 控制寄存器

CPU 可对该寄存器进行读或写操作。其中有两个位受硬件影响: SI 和 STO 位。SI 位由硬件置位而 STO 位由硬件清零。

当 I<sup>2</sup>C 处于主模式时, CRSEL 决定 SCL 的频率, 在从模式中, 该位被忽略并自动与主 I<sup>2</sup>C 器件的时钟频率 (最大为 400kHz) 同步。当 CRSEL=1 时, I<sup>2</sup>C 接口将定时器 1 的溢出速率的 1/2 作为 I<sup>2</sup>C 时钟频率。

定时器 1 在 8 位自动重装模式（模式 2）中由用户编程。

$$I^2C \text{ 数据传输速率} = \text{定时器溢出速率} / 2 = PCLK / (2 \times (256 - \text{重装值}))$$

如果 fosc=12MHz，重装值为 0~255，则 I<sup>2</sup>C 数据传输速率范围为 11.72~3000Kbit/s。

当 CRSEL=0 时，I<sup>2</sup>C 接口使用内部时钟发生器（其频率由 I2SCLL 和 I2SCLH 寄存器的值决定）作为时钟源。占空比并一定要为 50%。

STA 位为起始标志。该位的置位将使 I<sup>2</sup>C 接口进入主模式，并在进入主模式后发送一个起始条件和重复起始条件。

STO 位为停止标志。该位的置位将使主模式中的 I<sup>2</sup>C 接口发送一个停止条件，或使从模式中的 I<sup>2</sup>C 接口从一个错误状态中恢复。

当 STA 和 STO 都置位时，如果是在主模式中，将向 I<sup>2</sup>C 总线发送一个停止条件，然后再发送一个起始条件；如果是在从模式中，将产生一个内部停止条件，但不发送到总线。

I2CON 地址: D8H		7	6	5	4	3	2	1	0
可位寻址		—	I2EN	STA	STO	SI	AA	—	CRSEL
复位源: 任何复位		—	I2EN	STA	STO	SI	AA	—	CRSEL
复位值: x0000x0B		—	I2EN	STA	STO	SI	AA	—	CRSEL
位	符号	功能							
I2CON.7	—	保留将来之用。用户程序不要将其置 1。							
I2CON.6	I2EN	<b>I<sup>2</sup>C 接口使能</b> 该位置位时，使能 I <sup>2</sup> C 接口；该位为 0 时，I <sup>2</sup> C 功能被禁止。							
I2CON.5	STA	<b>起始标志</b> STA=1: I <sup>2</sup> C 进入主模式，检测 I <sup>2</sup> C 总线并当总线空闲时产生一个起始条件。如果总线忙，则等待一个停止条件（释放总线）并在延迟半个内部时钟发生器周期后发送一个起始条件。当 I <sup>2</sup> C 接口已经处于主模式中并发送或接收了一些数据时，会发送一个重复的起始条件。STA 可在任何时候置位，当 I <sup>2</sup> C 接口处于可寻址的从模式时，STA 也可以置位。 STA=0: 不会产生起始或重复起始条件。							
I2CON.4	STO	<b>停止标志</b> STO=1: 在主模式中，向 I <sup>2</sup> C 总线发送一个停止条件。I <sup>2</sup> C 总线检测到停止条件时将自动清零该位。在从模式中，该位的置位可使总线从错误状态中恢复。该情况下，不向总线发送停止条件。硬件的表现就好像是接收到一个停止条件并切换到不可寻址的从接收器模式。STO 标志由硬件自动清零。							
I2CON.3	SI	<b>I<sup>2</sup>C 中断标志</b> 当进入 25 种可能的 I <sup>2</sup> C 状态中的任何一个后，该位置位。当 EA, EI2C(IEN1.0)以及 SI 置位时，产生一个中断请求。SI 必须通过软件清零。							
I2CON.2	AA	<b>声明应答标志</b> 当该位置位时，下面的任意条件之一将产生一个应答： <ul style="list-style-type: none"> <li>• 接收到自身的从地址</li> <li>• 接收到通用调用地址（GC 置位时）</li> <li>• 当 I<sup>2</sup>C 接口处于主接收器模式时，接收到一个数据字节</li> <li>• 当 I<sup>2</sup>C 接口处于可寻址的从接收器模式时，接收到一个数据字节</li> </ul> 当该位清零时，下列情况下将返回一个非应答标志（SDA 上的高电平）： <ul style="list-style-type: none"> <li>• 当 I<sup>2</sup>C 接口处于主接收器模式时，接收到一个数据字节</li> <li>• 当 I<sup>2</sup>C 接口处于可寻址的从接收器模式时，接收到一个数据字节</li> </ul>							
I2CON.1	—	保留将来之用。用户程序不要将其置 1。							
I2CON.0	CRSEL	<b>SCL 时钟选择</b> CRSEL=1: 定时器 1 溢出产生 SCL；CRSEL=0: 使用内部 SCL 发生器。							

图 47 I<sup>2</sup>C 控制寄存器

### I<sup>2</sup>C 状态寄存器

I2STAT 是一个 8 位只读寄存器。它包含了 I<sup>2</sup>C 接口的状态代码。最低 3 位总是为 0。I<sup>2</sup>C 一共有 26 种可能的状态。当代码为 F8H 时表示无可用的相关状态信息，SI 也不会置位。所有其它 25 种状态码都对应一个已定义的 I<sup>2</sup>C 状态。进入任何一个状态时，SI 都会置位。详见表 16 到 19。

I2STAT 地址: D9H								
不可位寻址	7	6	5	4	3	2	1	0
复位源: 任何复位	STA.4	STA.3	STA.2	STA.1	STA.0	0	0	0
复位值: 11111000B								
位	符号	功能						
I2STAT.7-3	STA.4-0	I <sup>2</sup> C 的状态代码						
I2STAT.2-0	—	这 3 位未用并总是为 0。						

图 48 I<sup>2</sup>C 状态寄存器

### I<sup>2</sup>C SCL 占空比寄存器 I2SCLH 和 I2SCLL

当设置 CRSEL=0 以选择内部 SCL 发生器作为 I<sup>2</sup>C 接口的时钟源时，用户必须对 I2SCLL 和 I2SCLH 进行设置以选择合适的数据传输速率。I2SCLH 定义 SCL 高电平的 PCLK 周期数，I2SCLL 定义 SCL 低电平的 PCLK 周期数。频率由下面的公式决定：

$$\text{位速率} = f_{\text{PCLK}} / (2 \times (\text{I2SCLH} + \text{I2SCLL}))$$

此处， $f_{\text{PCLK}}$  表示 PCLK 的频率。

I2SCLL 和 I2SCLH 的值不一定要相同，用户可通过设置这两个寄存器得到不同的 SCL 占空比。但寄存器的值必须确保 I<sup>2</sup>C 数据传输速率范围为 0~400kHz。因此 I2SCLL 和 I2SCLH 的值有一些限制，建议两个寄存器的取值都应大于 3 个 PCLK 周期。

表 15 I<sup>2</sup>C 时钟速率选择

I2SCLL + I2SCLH	CRSEL	位数据速率 (Kbits/s)				
		7.373MHz	3.6865MHz	1.8433MHz	12MHz	6MHz
6	0	-	307	154	-	-
7	0	-	263	132	-	-
8	0	-	230	115	-	375
9	0	-	205	102	-	333
10	0	369	184	92	-	300
15	0	246	123	61	400	200
25	0	147	74	37	240	120
30	0	123	61	31	200	100
50	0	74	37	18	120	60
60	0	61	31	15	100	50
100	0	37	18	9	60	30
150	0	25	12	6	40	20
200	0	18	9	5	30	15
-	1	3.6-922Kbps 模式 2 中的 定时器 1	1.8-461Kbps 模式 2 中的 定时器 1	0.9-230Kbps 模式 2 中的 定时器 1	5.86-1500Kbps 模式 2 中的定 时器 1	2.93-750Kbps 模式 2 中的定 时器 1

### I<sup>2</sup>C 操作模式

#### 主发送器模式

在该模式中，数据从主机发送到从机。在进入主发送器模式之前，I2CON 必须按照下表进行初始化：

I2CON(D8h)	7	6	5	4	3	2	1	0
	—	I2EN	STA	STO	SI	AA	—	CRSEL
	—	1	0	0	0	x	—	位速率

CRSEL 定义串行位速率。I2EN 必须置位以启用 I<sup>2</sup>C 功能。如果 AA 位为 0，而另一个器件成为总线的主控器时，I<sup>2</sup>C 将不会对它自身的从地址或通用调用地址产生应答。换句话说，如果 AA 复位，I<sup>2</sup>C 不能进

入从模式。STA, STO 和 SI 必须设置为 0。

第一个发送的数据包含接收器件的从地址（7 位）和数据方向位。在此模式下，数据方向位（R/W）应当为 0 表示执行写操作。因此第一个发送的字节为 SLA+W。数据的发送每次为 8 位。每发送完一个字节，都接收到一个应答位。起始和停止条件用于指示串行传输的起始和结束。

通过置位 STA 进入 I<sup>2</sup>C 主发送器模式。I<sup>2</sup>C 逻辑在总线空闲后立即发送一个起始条件。当发送完起始条件后，SI 置位。此时状态寄存器（I2STAT）中的状态代码应当为 08H。该状态代码用于指向一个中断服务程序。该中断程序将从地址和数据方向位（SLA+W）装入 I2DAT。SI 位必须在数据传输能够继续进行之前清零。

当从地址和方向位已发送且接收到应答位之后，SI 位再次置位，并且对于主模式，可能的状态代码为 18H, 20H 或 38H，如果从模式使能（AA=1），可能的状态代码为 68H, 78H 或 B0H。每个状态代码对应的执行动作见表 16。

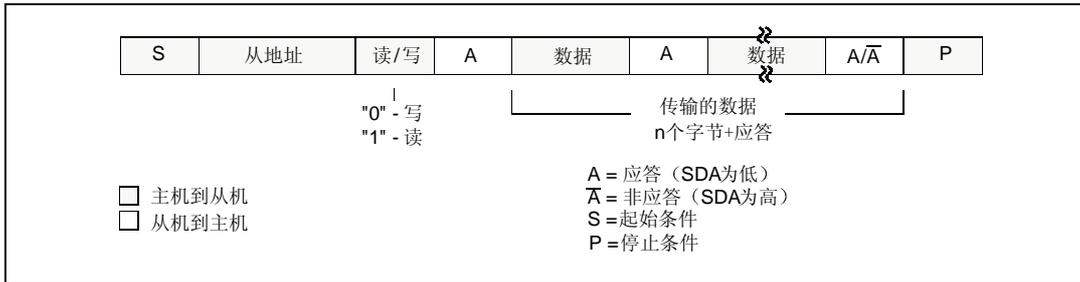


图 49 主发送器模式中的格式

### 主接收器模式

在主接收器模式中，数据字节接收自从发送器。传输的初始化与主发送器模式相同。发送完起始条件后，中断服务程序必须将 7 位从地址和数据方向位（SLA+R）装入 I<sup>2</sup>C 数据寄存器（I2DAT）。SI 位必须在数据传输能够继续进行之前清零。

当从地址和方向位已发送且接收到应答位之后，SI 置位而状态寄存器将显示状态代码。对于主模式，可能的状态代码为 40H, 48H 或 38H，对于从模式，可能的状态代码为 68H, 78H 或 B0H。详见表 17。

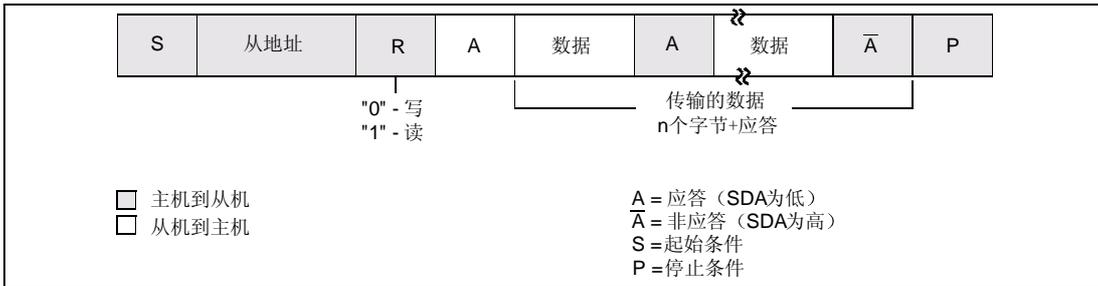


图 50 主接收器模式的格式

在一个重复的起始条件之后，I<sup>2</sup>C 可以切换到主发送器模式。

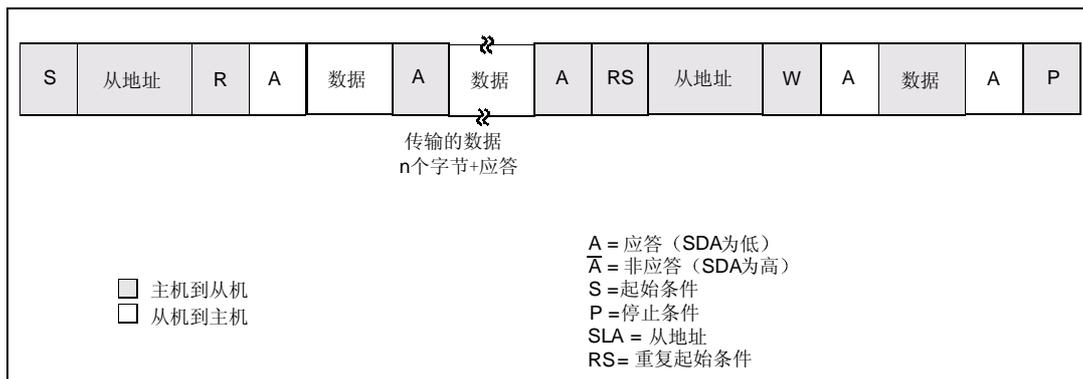


图 51 在发送重复起始条件后，主接收器切换到主发送器

### 从接收器模式

在从接收器模式中，从主发送器接收数据字节。要初始化从接收器模式，用户必须将从地址写入从地址寄存器 (I2ADR) 并按照下表配置 I<sup>2</sup>C 控制寄存器 (I2CON):

	7	6	5	4	3	2	1	0
I2CON(D8h)	—	I2EN	STA	STO	SI	AA	—	CRSEL
	—	1	0	0	0	1	—	—

CRSEL 不用于从模式。I2EN 必须置位以启用 I<sup>2</sup>C 功能。AA 位必须置位以使 I<sup>2</sup>C 应答自身的从地址或通用调用地址。STA, STO 和 SI 设置为 0。

当 I2ADR 和 I2CON 完成初始化时，I<sup>2</sup>C 等待直到它被自身的从地址加上数据方向位 0(W)寻址。如果数据方向位为 1(R)，I<sup>2</sup>C 将进入从发送器模式。在接收到自身从地址和 W 位后，SI 置位并可从 I2STAT 中读出有效的状态代码。状态代码及其对应的执行动作见表 18。

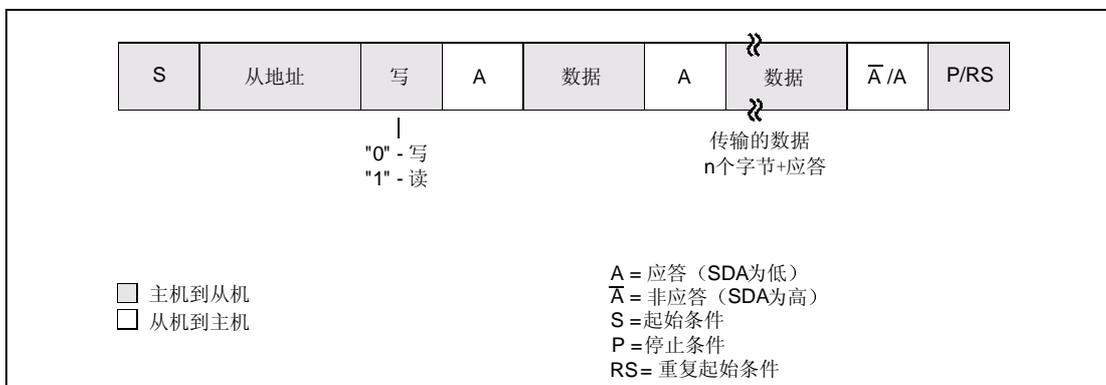


图 52 从接收器模式的格式

### 从发送器模式

第一个字节的接收和处理与从接收器模式相同。但在该模式中，方向位指示传输的方向掉转。串行数据通过 P1.3/SDA 发送而串行时钟通过 P1.2/SCL 输入。在串行传输的开始和结束对起始和停止条件进行识别。在一个给定的应用中，I<sup>2</sup>C 可以为主模式也可以为从模式。在从模式中，I<sup>2</sup>C 寻找它自身的从地址和通用调用地址。如果检测到其中一个地址，将产生中断请求。当微控制器希望成为总线主机时，硬件在进入主模式前一直等待，直到总线释放。这样就不会中断一个可能的从机动作。如果在主模式中总线仲裁丢失，I<sup>2</sup>C 将立即切换到从模式并能在同一个串行传输中检测自身的从地址。

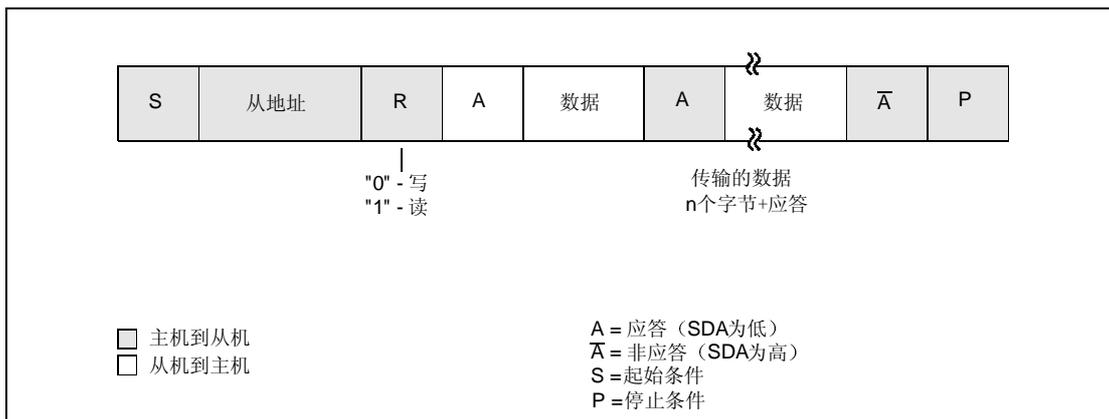


图 53 从发送器模式的格式

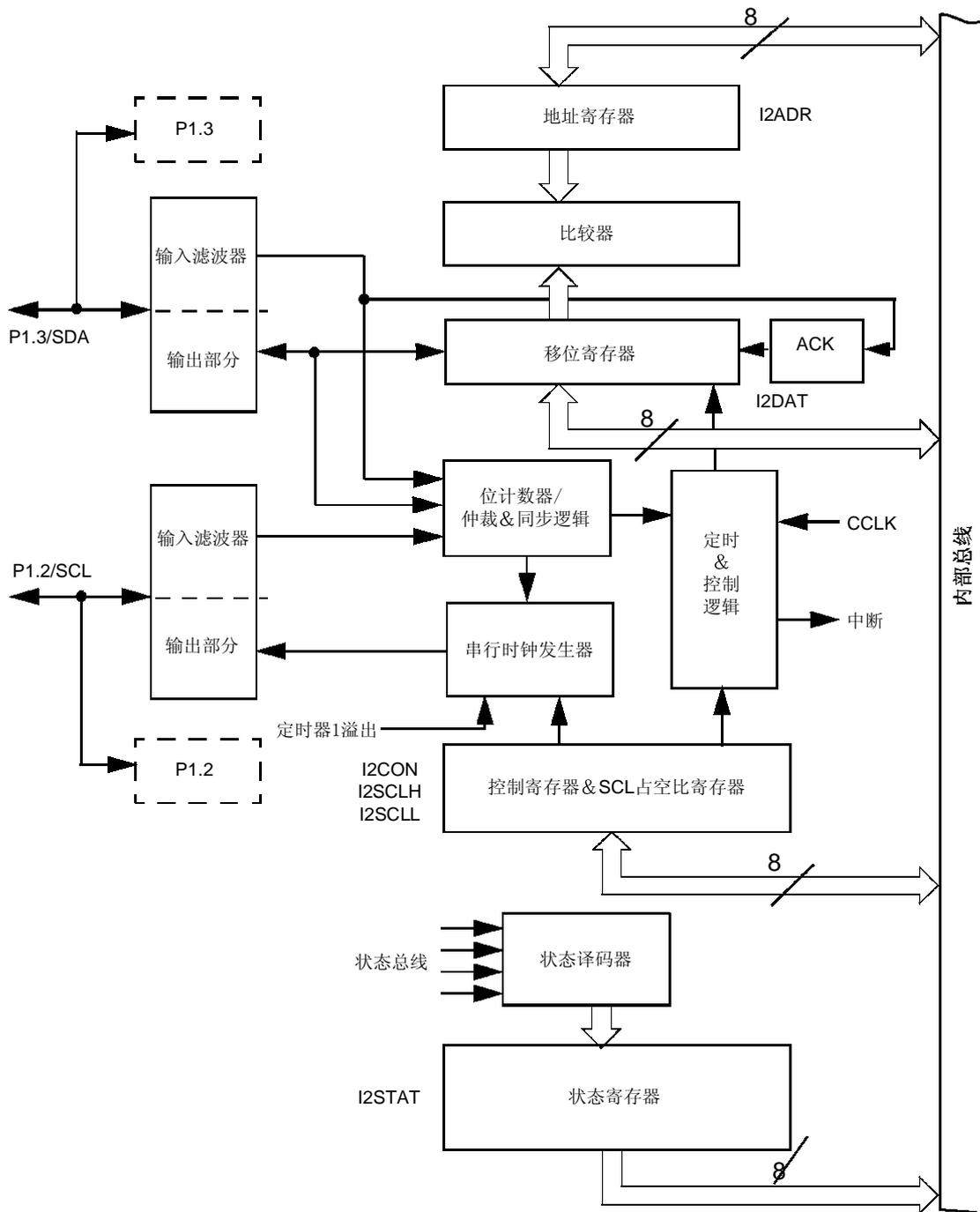


图 54 I<sup>2</sup>C 总线串行接口框图

表 16 主发送器模式

状态代码 (I2STAT)	I <sup>2</sup> C 总线硬件状态	应用软件的响应					I <sup>2</sup> C 硬件执行的下一个动作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
08H	已发送起始条件	装入 SLA+W	x	0	0	x	将发送 SLA+W, 接收 ACK 位
10H	已发送重复起始条件	装入 SLA+W 或 装入 SLA+R	x	0	0	x	同上; 将发送 SLA+W, I <sup>2</sup> C 将切换到主接收器模式
18H	已发送 SLA+W; 已接收 ACK	装入数据字节	0	0	0	x	将发送数据字节, 接收 ACK 位
		无 I2DAT 动作	1	0	0	x	将发送重复起始条件
		无 I2DAT 动作	0	1	0	x	将发送停止条件; STO 标志将复位
		无 I2DAT 动作	1	1	0	x	将发送停止条件, 然后发送起始条件; STO 标志将复位
20H	已发送 SLA+W; 已接收非 ACK	装入数据字节	0	0	0	x	将发送数据字节, 接收 ACK 位
		无 I2DAT 动作	1	0	0	x	将发送重复起始条件
		无 I2DAT 动作	0	1	0	x	将发送停止条件; STO 标志将复位
		无 I2DAT 动作	1	1	0	x	将发送停止条件, 然后发送起始条件; STO 标志将复位
28H	已发送 I2DAT 中的数 据字节; 已接收 ACK	装入数据字节	0	0	0	x	将发送数据字节, 接收 ACK 位
		无 I2DAT 动作	1	0	0	x	将发送重复起始条件
		无 I2DAT 动作	0	1	0	x	将发送停止条件; STO 标志将复位
		无 I2DAT 动作	1	1	0	x	将发送停止条件, 然后发送起始条件; STO 标志将复位
30H	已发送 I2DAT 中的数 据字节; 已接收非 ACK	装入数据字节	0	0	0	x	将发送数据字节, 接收 ACK 位
		无 I2DAT 动作	1	0	0	x	将发送重复起始条件
		无 I2DAT 动作	0	1	0	x	将发送停止条件; STO 标志将复位
		无 I2DAT 动作	1	1	0	x	将发送停止条件, 然后发送起始条件; STO 标志将复位
38H	在 SLA+R/W 或数据 字节中丢失仲裁	无 I2DAT 动作	0	0	0	x	I <sup>2</sup> C 总线将被释放; 进入不可寻址从模式
		无 I2DAT 动作	1	0	0	x	当总线变为空闲时发送起始条件

表 17 主接收器模式

状态代码 (I2STAT)	I <sup>2</sup> C 总线硬件状态	应用软件的响应					I <sup>2</sup> C 硬件执行的下一个动作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
08H	已发送起始条件	装入 SLA+W	x	0	0	x	将发送 SLA+R, 接收 ACK 位
10H	已发送重复起始条件	装入 SLA+R 装入 SLA+W	x	0	0	x	同上 将发送 SLA+W, I <sup>2</sup> C 将切换到主发送器模式
38H	在非 ACK 位中丢失 仲裁	无 I2DAT 动作	0	0	0	x	I <sup>2</sup> C 总线将被释放; I <sup>2</sup> C 将进入从模式
		无 I2DAT 动作	1	0	0	x	当总线恢复空闲后发送起始条件
40H	已发送 SLA+R; 已接 收 ACK	无 I2DAT 动作	0	0	0	0	将接收数据字节; 返回非 ACK 位
		无 I2DAT 动作	0	0	0	1	将接收数据字节; 返回 ACK 位
48H	已发送 SLA+R; 已接 收非 ACK	无 I2DAT 动作	1	0	0	x	将发送重复起始条件
		无 I2DAT 动作	0	1	0	x	将发送停止条件; STO 标志将复位
		无 I2DAT 动作	1	1	0	x	将发送停止条件, 然后发送起始条件; STO 标志将复位
50H	已接收数据字节; 已 返回非 ACK	读数据字节	0	0	0	0	将接收数据字节, 返回非 ACK 位
		读数据字节	0	0	0	1	将接收数据字节; 返回 ACK 位
58H	已接收数据字节; 已 返回 ACK	读数据字节	1	0	0	x	将发送重复起始条件
		读数据字节	0	1	0	x	将发送停止条件; STO 标志将复位
		读数据字节	1	1	0	x	将发送停止条件, 然后发送起始条件; STO 标志将复位

表 18 从接收器模式

状态代码 (I2STAT)	I <sup>2</sup> C 总线硬件状态	应用软件的响应					I <sup>2</sup> C 硬件执行的下一个动作
		读/写 I2DAT	写 I2CON				
			STA	STO	SI	AA	
60H	已接收自身 SLA+W; 已返回 ACK	无 I2DAT 动作	x	0	0	0	将接收数据字节并返回非 ACK 位
		无 I2DAT 动作	x	0	0	1	将接收数据字节并返回 ACK 位
68H	主控器时在 SLA+W 中丢失仲裁; 已接收 自身 SLA+W, 已返回 ACK	无 I2DAT 动作	x	0	0	0	将接收数据字节并返回非 ACK 位
		无 I2DAT 动作	x	0	0	1	将接收数据字节并返回 ACK 位
70H	已接收通用调用地址 (00H); 已返回 ACK	无 I2DAT 动作	x	0	0	0	将接收数据字节并返回非 ACK 位
		无 I2DAT 动作	x	0	0	1	将接收数据字节并返回 ACK 位
78H	主控器时在 SLA+R/W 中丢失仲 裁; 已接收通用调用 地址; 已返回 ACK	无 I2DAT 动作	x	0	0	0	将接收数据字节并返回非 ACK 位
		无 I2DAT 动作	x	0	0	1	将接收数据字节并返回 ACK 位
80H	前一次寻址使用自身 从地址; 已接收数据 字节; 已返回 ACK	读数据字节	x	0	0	0	将接收数据字节并返回非 ACK 位
		读数据字节	x	0	0	1	将接收数据字节并返回 ACK 位

88H	前一次寻址使用自身从地址; 已接收数据字节; 已返回非 ACK	读数据字节 读数据字节 读数据字节 读数据字节	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	切换到不可寻址 SLV 模式; 不识别自身 SLA 或通用调用地址 切换到不可寻址 SLV 模式; 识别自身 SLA; 如果 S1ADR.0=1, 将识别通用调用地址 切换到不可寻址 SLV 模式; 不识别自身 SLA 或通用调用地址; 当总线空闲后发送起始条件 切换到不可寻址 SLV 模式; 识别自身 SLA; 如果 S1ADR.0=1, 将识别通用调用地址; 当总线空闲后发送起始条件
90H	前一次寻址使用通用调用; 已接收数据字节; 已返回 ACK	读数据字节 读数据字节	x x	0 0	0 0	0 1	将接收数据字节并返回非 ACK 位 将接收数据字节并返回 ACK 位
98H	前一次寻址使用通用调用; 已接收数据字节; 已返回非 ACK	读数据字节 读数据字节或 读数据字节或 读数据字节	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	切换到不可寻址 SLV 模式; 不识别自身 SLA 或通用调用地址 切换到不可寻址 SLV 模式; 识别自身 SLA; 如果 S1ADR.0=1, 将识别通用调用地址 切换到不可寻址 SLV 模式; 不识别自身 SLA 或通用调用地址; 当总线空闲后发送起始条件 切换到不可寻址 SLV 模式; 识别自身 SLA; 如果 S1ADR.0=1, 将识别通用调用地址; 当总线空闲后发送起始条件
A0H	当使用 SLV/REC 或 SLV/TRX 静态寻址时, 接收到停止条件或重复的起始条件	无 I2DAT 动作或 无 I2DAT 动作或 无 I2DAT 动作或 无 I2DAT 动作	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	切换到不可寻址 SLV 模式; 不识别自身 SLA 或通用调用地址 切换到不可寻址 SLV 模式; 识别自身 SLA; 如果 S1ADR.0=1, 将识别通用调用地址 切换到不可寻址 SLV 模式; 不识别自身 SLA 或通用调用地址; 当总线空闲后发送起始条件 切换到不可寻址 SLV 模式; 识别自身 SLA; 如果 S1ADR.0=1, 将识别通用调用地址; 当总线空闲后发送起始条件

表 19 从发送器模式

状态代码 (I2STAT)	I <sup>2</sup> C 总线硬件状态	应用软件的响应				I <sup>2</sup> C 硬件执行的下一个动作	
		读/写 I2DAT	写 I2CON				
			STA	STO	SI		AA
A8H	已接收自身 SLA+R; 已返回 ACK	装入数据字节或装入数据字节	x x	0 0	0 0	0 1	将发送最后的数据字节并接收 ACK 位 将发送数据字节并接收 ACK 位
B0H	主控器时在 SLA+R/W 中丢失仲裁; 已接收自身 SLA+R, 已返回 ACK	装入数据字节或装入数据字节	x x	0 0	0 0	0 1	将发送最后的数据字节并接收 ACK 位 将发送数据字节并接收 ACK 位
B8H	已发送 I2DAT 中数据字节; 已返回 ACK	装入数据字节或装入数据字节	x x	0 0	0 0	0 1	将发送最后的数据字节并接收 ACK 位 将发送数据字节并接收 ACK 位
C0H	已发送 I2DAT 中数据字节; 已返回非 ACK	无 I2DAT 动作或 无 I2DAT 动作或 无 I2DAT 动作或 无 I2DAT 动作	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	切换到不可寻址 SLV 模式; 不识别自身 SLA 或通用调用地址 切换到不可寻址 SLV 模式; 识别自身 SLA; 如果 S1ADR.0=1, 将识别通用调用地址 切换到不可寻址 SLV 模式; 不识别自身 SLA 或通用调用地址; 当总线空闲后发送起始条件 切换到不可寻址 SLV 模式; 识别自身 SLA; 如果 S1ADR.0=1, 将识别通用调用地址; 当总线空闲后发送起始条件
C8H	已发送 I2DAT 中最后的数据字节 (AA=0); 已返回 ACK	无 I2DAT 动作或 无 I2DAT 动作或 无 I2DAT 动作或 无 I2DAT 动作	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	切换到不可寻址 SLV 模式; 不识别自身 SLA 或通用调用地址 切换到不可寻址 SLV 模式; 识别自身 SLA; 如果 S1ADR.0=1, 将识别通用调用地址 切换到不可寻址 SLV 模式; 不识别自身 SLA 或通用调用地址; 当总线空闲后发送起始条件 切换到不可寻址 SLV 模式; 识别自身 SLA; 如果 S1ADR.0=1, 将识别通用调用地址; 当总线空闲后发送起始条件
F8H	无可用相关信息; SI=0	无 I2DAT 动作		无 I2DAT 动作			等待或进行当前的传输
00H	在 MST 或选择的从模式中, 由于非法的起始或停止条件, 使总线发生错误。当干扰导致 I <sup>2</sup> C 进入一个未定义的状态时, 也可产生状态 00H	无 I2DAT 动作	0	1	0	x	在 MST 或寻址 SLV 模式中只有内部硬件受影响。在所有情况下, 总线被释放, 而 I <sup>2</sup> C 切换到不可寻址 SLV 模式。STO 复位。

要了解更多有关 I<sup>2</sup>C 接口的内容, 请参阅 I<sup>2</sup>C 规范。

## 串行外围接口 (SPI)

LPC932 还提供另一种高速串行通信接口——SPI 接口。SPI 是一种全双工、高速、同步的通信总线，有两种操作模式：主模式和从模式。在主模式和从模式中均支持高达 3Mbit/s 的速率。还具有传输完成标志和写冲突标志保护。

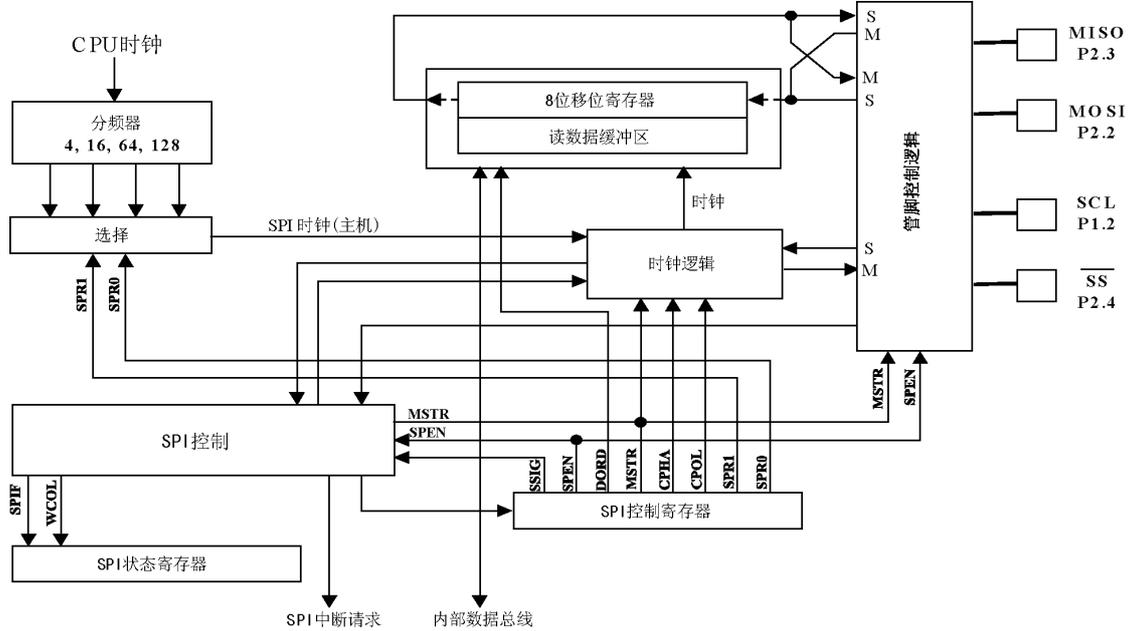


图 55 SPI 方框图

SPI 接口有 4 个管脚：SPICLK, MOSI, MISO 和 SS：

- SPICLK, MOSI 和 MISO 通常和两个或更多 SPI 器件连接在一起。数据通过 MOSI 从主机传送到从机，通过 MISO 从从机传送到主机。SPICLK 信号在主模式时为输出，在从模式时为输入。如果 SPI 系统被禁止，即 SPEN(SPCTL.6)=0(复位值)，这些管脚都可作为 I/O 口使用。
- $\overline{SS}$  为从机选择管脚。在典型的配置中，SPI 主机使用 I/O 口选择一个 SPI 器件作为当前的从机。SPI 从器件通过其  $\overline{SS}$  脚确定是否被选择。如果满足下面的条件之一， $\overline{SS}$  就被忽略：
  - 如果 SPI 系统被禁止，即 SPEN(SPCTL.6)=0(复位值)
  - 如果 SPI 配置为主机，即 MSTR(SPCTL.4)=1，并且 P2.4 配置为输出（通过 P2M1.4 和 P2M2.4）
  - 如果  $\overline{SS}$  脚被忽略，即 SSIG(SPCTL.7)位=1，该脚配置用于 I/O 口功能。

注：即使 SPI 被配置为主机（MSTR=1），它仍然可以通过拉低  $\overline{SS}$  脚配置为从机（如果 P2.4 配置为输入且 SSIG=0）。要能使该特性，应当置位 SPIF(SPSTAT.7)（见“通过 SS 改变模式”一节）。

典型连接如图 59~61 所示。

SPCTL 地址: E2H		
不可位寻址		7      6      5      4      3      2      1      0
复位源: 任何复位		SSIG   SPEN   DORD   MSTR   CPOL   CPHA   SPR1   SPR0
复位值: 00000100B		
位	符号	功能
SPCTL.7	SSIG	$\overline{SS}$ 忽略。如果 SSIG=0, $\overline{SS}$ 脚用于确定器件为主机还是从机。如果 SSIG=1, MSTR 确定器件为主机还是从机。 $\overline{SS}$ 脚可作为 I/O 口使用 (见表 20)。
SPCTL.6	SPEN	SPI 使能。当 SPEN=1 时, SPI 使能; SPEN=0 时, SPI 被禁止, 所有 SPI 管脚都作为 I/O 口。
SPCTL.5	DORD	SPI 数据顺序: 1: 数据字的 LSB 最先发送; 0: 数据字的 MSB 最先发送。
SPCTL.4	MSTR	主/从模式选择 (见表 20)。
SPCTL.3	CPOL	SPI 时钟极性 (见图 62~65): 1: SPICLK 空闲时为高电平。SPICLK 的前时钟沿为下降沿而后沿为上升沿。 0: SPICLK 空闲时为低电平。SPICLK 的前时钟沿为上升沿而后沿为下降沿。
SPCTL.2	CPHA	SPI 时钟相位选择 (见图 62~65): 1: 数据在 SPICLK 的前时钟沿驱动, 并在后时钟沿采样。 0: 数据在 SS 为低时被驱动, 在 SPICLK 的后时钟沿被改变, 并在前时钟沿被采样。(注: SSIG=1 时的操作未定义)
SPCTL.1-0	SPR1-SPR0	SPI 时钟速率
	00	CCLK/4
	01	CCLK /16
	10	CCLK /64
	11	CCLK /128

图 56 SPI 控制寄存器

SPSTAT 地址: E1H		
不可位寻址		7      6      5      4      3      2      1      0
复位源: 任何复位		SPIF   WCOL   -      -      -      -      -      -
复位值: 00xxxxxB		
位	符号	功能
SPSTAT.7	SPIF	SPI 传输完成标志。当一次串行传输完成时, SPIF 置位, 并当 ESPI(IEN1.3)和 EA 都置位时产生中断。当 SPI 处于主模式且 SSIG=0 时, 如果 $\overline{SS}$ 为输入并被驱动为低电平, SPIF 也将置位 (见“通过 $\overline{SS}$ 改变模式”一节)。SPIF 标志通过软件向其写入“1”清零。
SPSTAT.6	WCOL	SPI 写冲突标志。在数据传输的过程中如果对 SPI 数据寄存器 SPDAT 执行写操作, WCOL 将置位 (见“写冲突”一节)。SPIF 标志通过软件向其写入“1”清零。
SPSTAT.5-0	—	保留将来之用。用户程序不要将其置 1。

图 57 SPI 状态寄存器定义

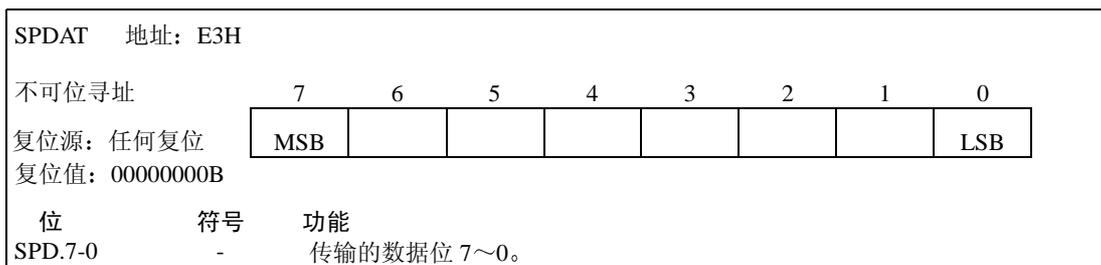


图 58 SPI 数据寄存器

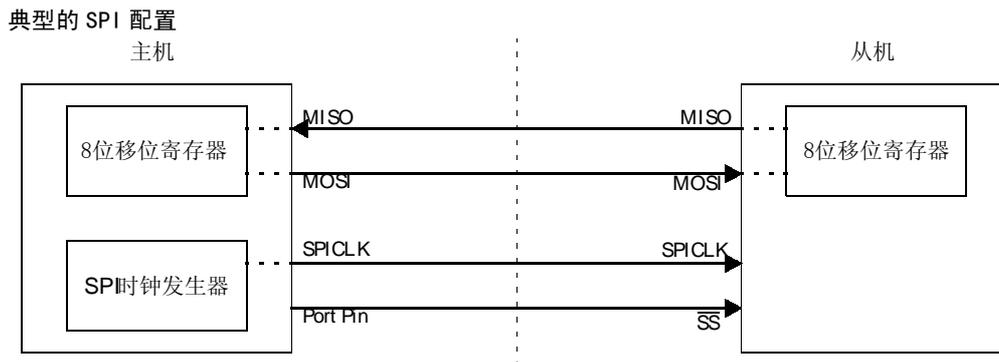


图 59 SPI 单主机单从机配置

在图 59 中，从机的 SSIG(SPCTL.7)为 0， $\overline{SS}$  用于选择从机。SPI 主机可使用任何端口（包括 P2.4/ $\overline{SS}$ ）来驱动  $\overline{SS}$  脚。

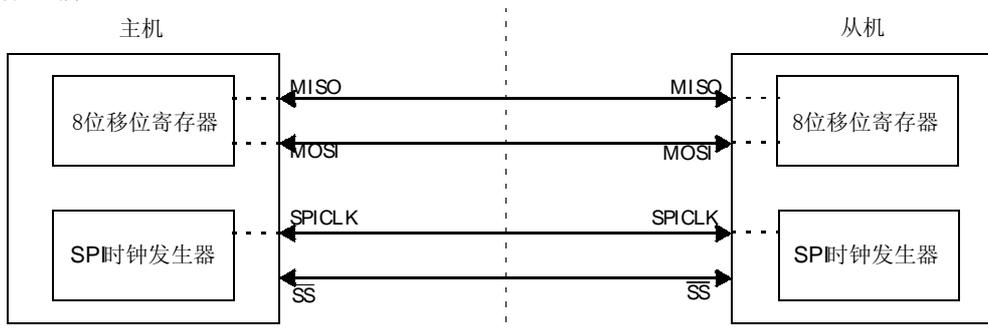


图 60 SPI 双器件配置（两器件可互为主从）

图 60 所示为两个器件互为主从的情况。当没有发生 SPI 操作时，两个器件都可配置为主机 (MSTR=1)，将 SSIG 清零并将 P2.4(SS)配置为准双向模式。当其中一个器件启动传输时，它可将 P2.4 配置为输出并驱动为低电平，这样就强制另一个器件变为从机（见“通过  $\overline{SS}$  改变模式”一节）。

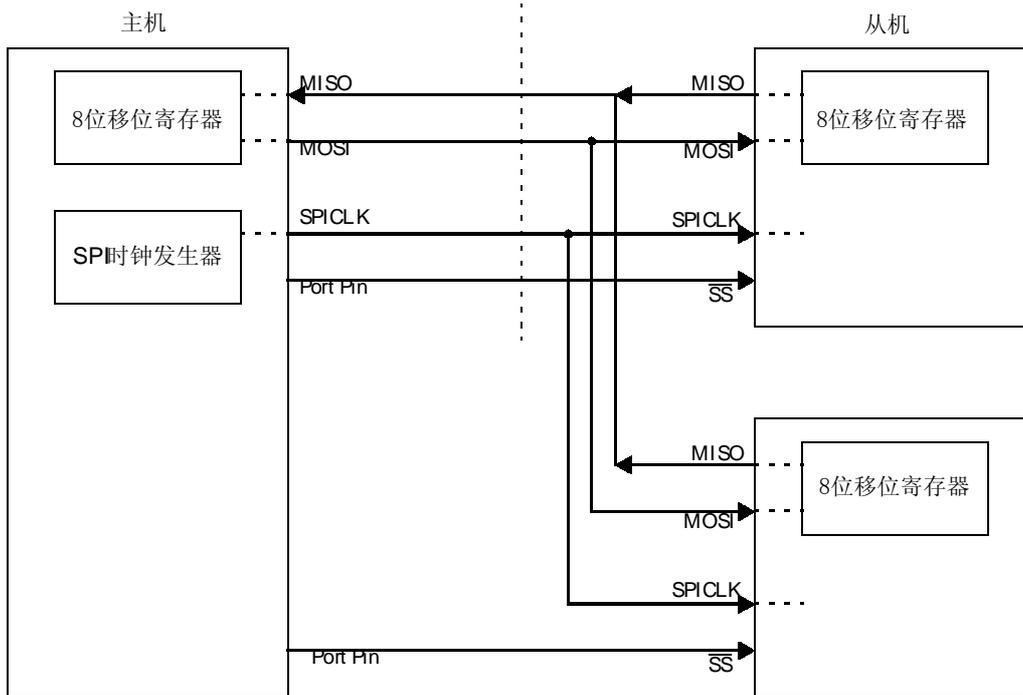


图 61 SPI 单主机多从机配置

在图 61 中，从机的 SSIG(SPCTL.7)为 0，从机通过对应的  $\overline{SS}$  信号被选中。SPI 主机可使用任何端口（包括 P2.4/SS）来驱动  $\overline{SS}$  脚。

### 对 SPI 进行配置

表 20 所示为主/从模式的配置以及模式的使用和传输方向。

表 20 SPI 主从选择

SPEN (SPCTL.6)	SSIG (SPCTL.7)	P2M2.4	SS 脚	MSTR (SPCTL.4)	主或从模式	MISO	MOSI	SPICLK	备注
0	X	X	P2.4 <sup>1</sup>	X	SPI 禁止	P2.3 <sup>1</sup>	P2.2 <sup>1</sup>	P2.5 <sup>1</sup>	SPI 禁止。P2.2,P2.3,P2.4,P2.5 作为 I/O 口使用
1	0	X	0	0	从	输出	输入	输入	选择作为从机
1	0	X	1	0	从	高阻	输入	输入	未被选中。MISO 为高阻态以避免争夺总线
1	0	0	0	1(>0) <sup>2</sup>	从	输出	输入	输入	P2.4/SS 配置为输入或准双向口。SSIG 为 0。如果 SS 被驱动为低电平，则被选择作为从机。当 SS 变为低电平时，MSTR 将清零。
1	0	0	1	1	主	输入	高阻	高阻	MOSI 和 SPICLK 为高阻态以避免争夺总线。 注：用户必须将 SPICLK 上拉或下拉（根据 CPOL-SPCTL.3 的取值）以避免出现悬浮状态。
1	0	1	X	1	主	输入	输出	输出	MOSI 和 SPICLK 为推挽输出
1	1	X	P2.4 <sup>1</sup>	0	从	输出	输入	输入	
1	1	X	P2.4 <sup>1</sup>	1	主	输入	输出	输出	

注：1. 作为普通 I/O 口使用

2. 当 SS 处于输入模式时被驱动为低电平且 SSIG=0 时，MSTR 位自动清零。

### 作为从机时的额外注意事项

当 CPHA=0 时，SSIG 必须为 0，SS 脚必须取反并且在每个连续的串行字节之间重新设置为高电平。如果 SPDAT 寄存器在 SS 有效（低电平）时执行写操作，那么将导致一个写冲突错误。SPHA=0 且 SSIG=0 时的操作未定义。

当 CPHA=1 时，SSIG 可以置位。如果 SSIG=0，SS 脚可在连续传输之间保持低有效（即一直固定为低电平）。这种方式有时适用于具有单固定主机和单从机驱动 MISO 数据线的系统。

### 作为主机时的额外注意事项

在 SPI 中，传输总是由主机启动的。如果 SPI 使能 (SPEN=1) 并选择作为主机，主机对 SPI 数据寄存器的写操作将启动 SPI 时钟发生器和数据的传输。在数据写入 SPDAT 之后的半个到一个 SPI 位时间后，数据将出现在 MOSI 脚。

需要注意的是，主机可以通过将对应器件的 SS 脚驱动为低电平实现与之通信。写入主机 SPDAT 寄存器的数据从 MOSI 脚移出发送到从机的 MOSI 脚。同时从机 SPDAT 寄存器的数据从 MISO 脚移出发送到主机的 MISO 脚。

传输完一个字节后，SPI 时钟发生器停止，传输完成标志 SPIF 置位并产生一个中断（如果 SPI 中断使能）。主机和从机 CPU 的两个移位寄存器可以看作是一个 16 循环移位寄存器。当数据从主机移位传送到从机的同时，数据也以相反的方向移入。这意味着在一个移位周期中，主机和从机的数据相互交换。

### 通过 SS 改变模式

如果 SPEN=1, SSIG=0 且 MSTR=1, SPI 使能为主机模式。SS 脚可配置为输入或准双向模式。这种情况下，另外一个主机可将该脚驱动为低电平，从而将该器件选择为 SPI 从机并向其发送数据。为了避免争夺总线，SPI 系统执行以下动作：

1. MSTR 清零并且 CPU 变为从机。这样 SPI 就变成从机。MOSI 和 SPICLK 强制变为输入模式，而 MISO 则变为输出模式。

2. SPIF 标志置位。如果 SPI 中断使能，则产生 SPI 中断。

用户软件必须一直对 MSTR 位进行检测，如果该位被一个从机选择所清零而用户想继续将 SPI 作为主机，这时就必须重新置位 MSTR，否则就进入从机模式。

### 写冲突

SPI 在发送时为单缓冲，在接收时为双缓冲。这样在前一次发送尚未完成之前，不能将新的数据写入移位寄存器。当发送过程中对数据寄存器进行写操作时，WCOL 位将置位以指示数据冲突。这种情况下，当前发送的数据继续发送，而新写入的数据将丢失。

当对主机或从机进行写冲突检测时，主机发生写冲突的情况是很罕见的，因为主机拥有数据传输的完全控制权。但从机有可能发生，因为当主机启动传输时，从机无法进行控制。

接收数据时，接收到的数据传送到一个并行读数据缓冲区，这样将释放移位寄存器以进行下一个数据的接收。但必须在下个字符完全移入之前从数据寄存器中读出接收到的数据，否则，前一个接收数据将丢失。

WCOL 可通过软件向其写入“1”清零。

### 数据模式

时钟相位位(CPHA)允许用户设置采样和改变数据的时钟边沿。时钟极性位 CPOL 允许用户设置时钟极性。图 62~65 所示为时钟相位位 CPHA 的不同设定。

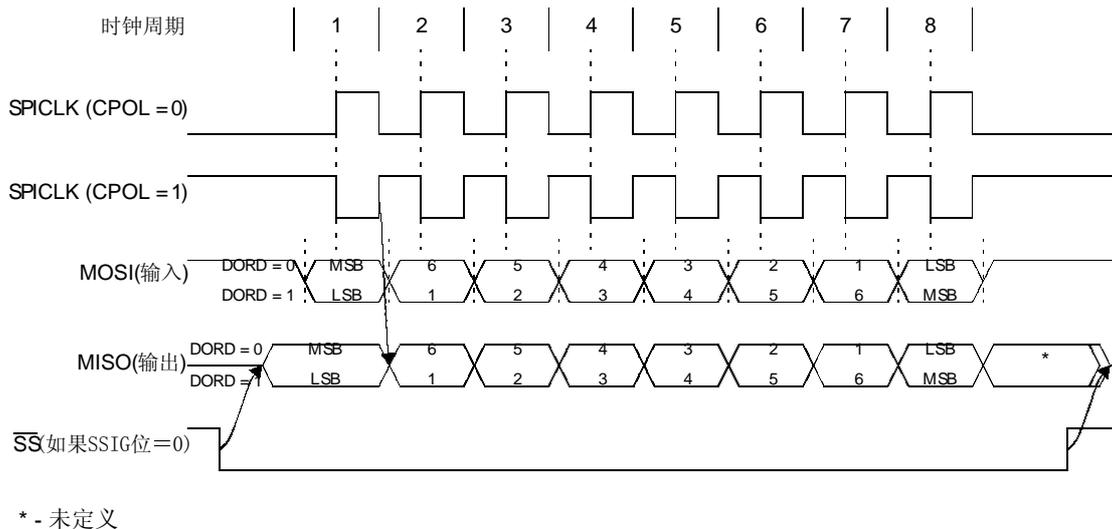


图 62 SPI 从机传输格式 (CPHA=0)

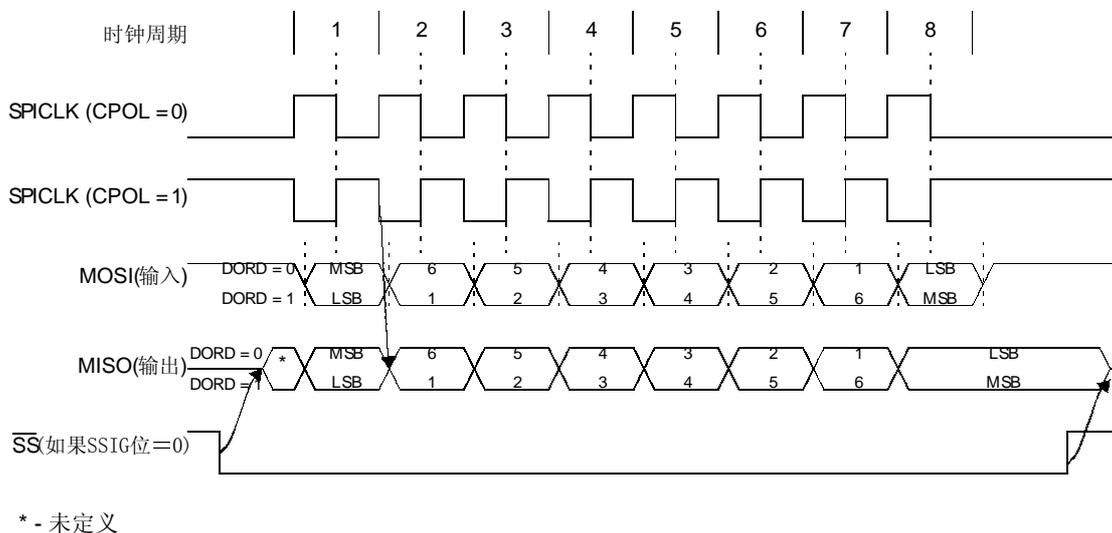


图 63 SPI 从机传输格式 (CPHA=1)

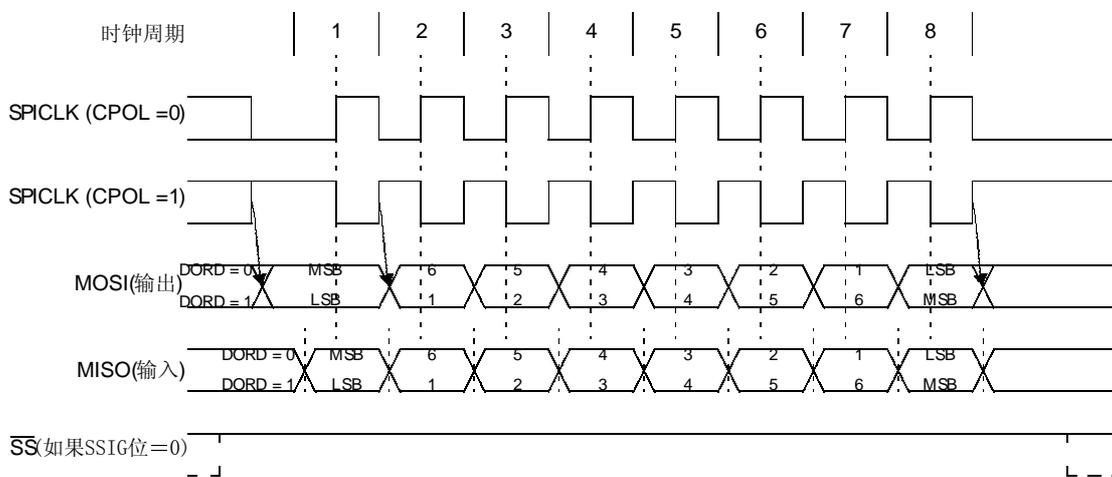


图 64 SPI 主机传输格式 (CPHA=0)

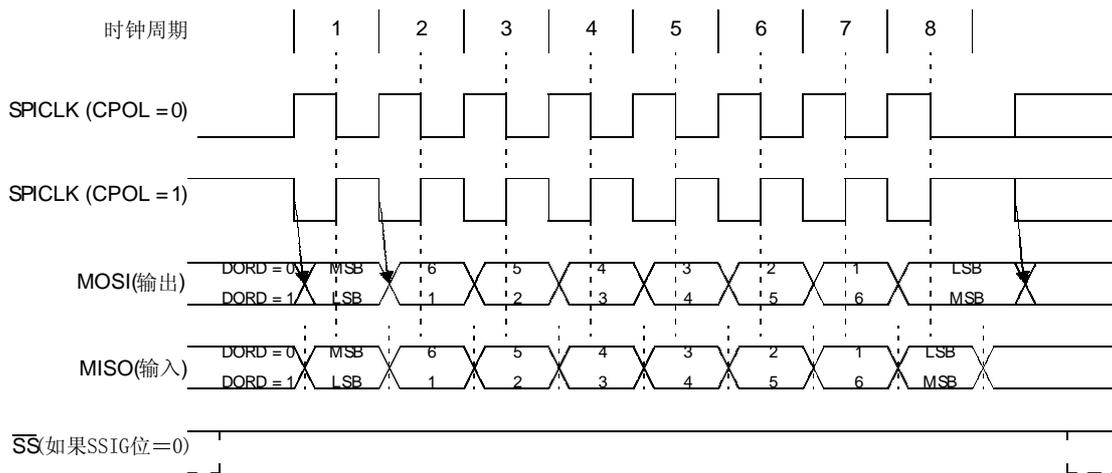


图 65 SPI 主机传输格式 (CPHA=1)

### SPI 时钟预分频器选择

SPI 时钟预分频器选择是通过 SPCTL 寄存器中的 SPR1-SPR0 位实现的 (见图 56)。

### 模拟比较器

LPC932 有 2 个模拟比较器, 输入和输出选项允许将比较器配置成为不同模式。当正向输入 (二个可选择脚之一) 电压大于反向输入时 (可选择外部管脚输入或内部参考电压), 输出信号为 “1” (可从寄存器读出和/或输出到管脚), 反之则输出为 “0”。每个比较器都可配置为当输出发生改变时产生中断。

### 比较器的配置

每个比较器都有各自的控制寄存器。比较器 1 为 CMP1, 比较器 2 为 CMP2。两个控制寄存器完全相同, 如图 66 所示。

两个比较器总的连接方式如图 67 所示。每个比较器可实现 8 种可能的配置方式, 这由对应的 CMPn 寄存器中的控制位 CPn、CNn 和 OEn 决定, 如图 68 所示。

当每个比较器刚被使能时, 比较器输出和中断标志需要 10 微秒的稳定时间, 在这段时间里, 相应的比较器中断不应使能, 并且在使能中断以前必须清零相应的比较器中断标志, 以避免立即响应中断服务。

CMPn	地址: CMP1:ACH	CMP2:ADH						
不可位寻址	7	6	5	4	3	2	1	0
复位源: 任何复位	—	—	CE1	CP1	CN1	OE1	CO1	CMF1
复位值: xx000000B								
位	符号	功能						
CMPn.7,6	—	保留将来之用, 用户程序请勿将其置‘1’。						
CMPn.5	CEn	比较器使能位。当置位时, 对应的比较器使能。CEn 置位 10 微秒后比较器输出有效值。						
CMPn.4	CPn	比较器正向输入选择, 为 ‘0’ 时选择 CINnA 作为正向输入, 为 ‘1’ 时选择 CINnB 作为正向输入。						
CMPn.3	CNn	比较器反向输入选择, 为 ‘0’ 时选择 CMPREF 作为比较器反向输入, 为 ‘1’ 时选择内部比较器参考电压 Vref 作为比较器反向输入。						
CMP1.2	OEn	输出使能, 为 ‘1’ 时, 比较结果输出连接到 CMPn 管脚。此输出和 CPU 时钟不同步。						
CMPn.1	COn	比较器输出, 和 CPU 同步以允许软件进行读取。						
CMPn.0	CMFn	比较器中断标志。当比较器输出 COn 状态改变时由硬件置位。使能比较器中断时, 该位置位可产生硬件中断。通过软件清零。						

图 66 比较器控制寄存器 (CMP1 和 CMP2)

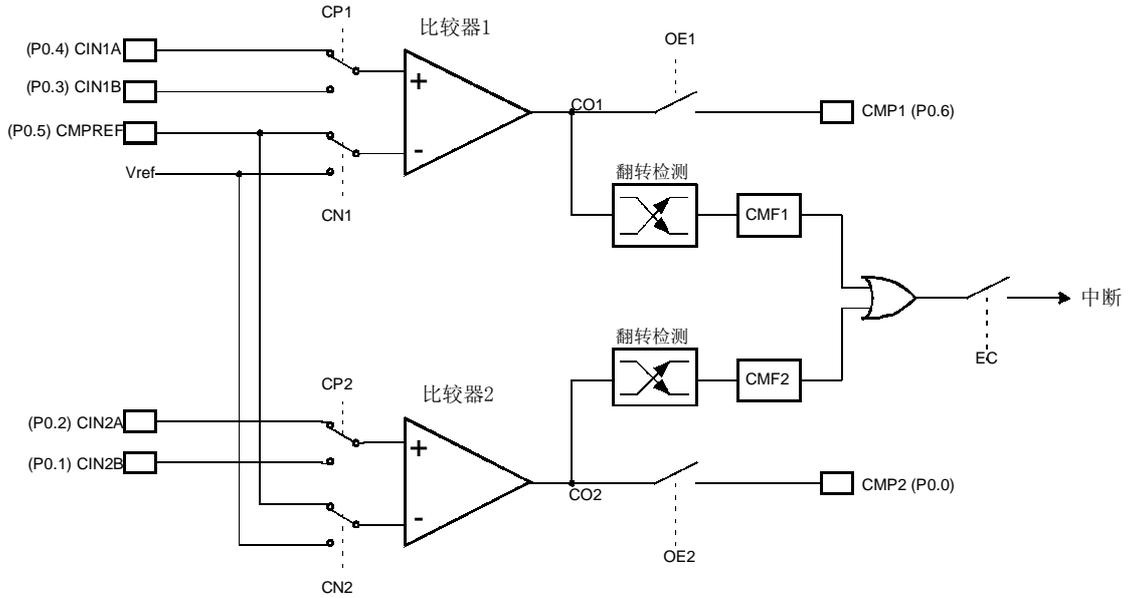


图 67 比较器输入和输出的连接

<p><u>CPn, CNn, OEn = 0 0 0</u></p>	<p><u>CPn, CNn, OEn = 0 0 1</u></p>
<p><u>CPn, CNn, OEn = 0 1 0</u></p>	<p><u>CPn, CNn, OEn = 0 1 1</u></p>
<p><u>CPn, CNn, OEn = 1 0 0</u></p>	<p><u>CPn, CNn, OEn = 1 0 1</u></p>
<p><u>CPn, CNn, OEn = 1 1 0</u></p>	<p><u>CPn, CNn, OEn = 1 1 1</u></p>

图 68 比较器配置

#### 内部参考电压

当使用单个比较器输入管脚时，内部参考电压发生器可提供一个默认的参考电压。参考电压的值请参阅数据手册。

#### 比较器中断

每个比较器配置寄存器中都有一个比较器中断标志位  $CMFn$ 。当比较器输出状态改变时中断标志位置位，此标志位可通过软件查询或用于产生一个中断。两个比较器使用同一个中断向量。当  $IEN1$  中的  $EC$  位置位且  $IEN0$  中的  $EA$  位置位时系统将会产生中断。如果两个比较器都使能中断，在进入中断服务程序之后，用户读取中断标志以确定是哪一个比较器产生的中断。

### 比较器和节电模式

在掉电模式或空闲模式下，两个比较器或其中一个可以继续保持使能状态。但在完全掉电模式中，两个比较器都被自动禁止。

当比较器中断使能时（完全掉电模式除外），比较器输出发生改变时将会产生一个中断并将 CPU 唤醒。当比较器输出到管脚使能时，此管脚应该配置为推挽输出模式以便在掉电工作模式下获得较快的开关速度。这样做是因为当振荡器停止后，打开双向口不会产生正常情况下的短时强上拉。

比较器在掉电或空闲状态下所消耗的电流和正常操作模式下相同。当系统功耗是一个重要的指标时，就必须将比较器的功耗考虑在内。为了使功耗降低，用户可通过禁止比较器并置位 PCONA.5 使比较器掉电，或简单地将器件设置为完全掉电模式。

### 比较器配置举例

下面是一段初始化比较器的程序。比较器 1 配置成为 CIN1A 和 CMPREF 输入，比较器结果输出到 CMP1 脚，当比较器输出结果发生改变时产生中断。

CMPINIT:

```

MOV    PT0AD,#30H    ; 禁止 CIN1A 和 CMPREF 上的数字输入功能，使能比较器功能
ANL    P0M2,#0CFH    ; 禁止 CIN1A 和 CMPREF 上的数字输出功能，使能比较器功能
ORL    P0M1,#30H     ; 开启比较器 1 并进行如下设置
MOV    CMP1,#24H     ; 一正向输入为 CIN1A
                          ; 一选择 CMPREF 脚作为参考电压
                          ; 一比较结果输出到 CMP1
CALL   DELAY10US    ; 比较器启动至少 10 微秒后方可使用
ANL    CMP1,#0FEH    ; 清除比较器 1 的中断标志
SETB   EC1           ; 使能比较器 1 中断，保持当前中断优先权
SETB   EA            ; 开中断
RET                                ; 返回调用处
    
```

在中断返回前必须清零中断标志位（该例中为 CMF1）。

### 键盘中断（KBI）

键盘中断功能主要用于当 P0 口等于或不等于特定的模式时产生一个中断。该功能可用于总线地址识别或对键盘的识别。用户可通过 SFR 将端口配置为不同的用途。

有 3 个特殊功能寄存器用于该功能。键盘中断屏蔽寄存器（KBMASK）用于定义连接到 P0 口的使能触发中断的输入管脚。键盘模式寄存器（KBPATN）用于定义与 P0 口值相比较的模式。当键盘中断功能有效且条件匹配时，键盘中断控制寄存器（KBCON）中的键盘中断标志（KBIF）置位。如果中断使能，则会产生一个中断。键盘中断控制寄存器（KBCON）中的 PATN\_SEL 位用于比较时定义等于或不等于。

为了将其设置为类似 87LPC76x 的 KBI 功能，用户必须设置 KBPATH=0FFH 和 PATN\_SEL=1（不相等），这样任何连接到 P0 口管脚（由 KBMASK 寄存器使能）的按键在按下时都将使硬件置位 KBIF 并产生中断（如果中断使能）。中断可用于将 CPU 从空闲模式或掉电模式中唤醒。此特性尤其适合便携式且使用电池供电的系统，因为这些系统需要对功耗进行管理同时又要方便用户使用。

为了置位中断标志并导致中断产生，P0 口的模式的保持时间必须长于 6 个 CCLK。

KBPATN 地址: 93H								
不可位寻址	7	6	5	4	3	2	1	0
复位源: 任何复位	KBPATN.7	KBPATN.6	KBPATN.5	KBPATN.4	KBPATN.3	KBPATN.2	KBPATN.1	KBPATN.0
复位值: 11111111B								
位	符号	功能						
KBPATN.7-0	—	模式位 7~0。						

图 69 键盘模式寄存器

KBCON 地址: 94H								
不可位寻址	7	6	5	4	3	2	1	0
复位源: 任何复位	—	—	—	—	—	—	PATN_SEL	KBIF
复位值: xxxxxx00B								
位	符号	功能						
KBCON.7-2	—	保留						
KBCON.1	PATN_SEL	模式匹配极性选择。该位置位时, P0 口必须等于用户在 KBPATN 中定义的模式才能产生中断。清零时, P0 口必须不等于 KBPATN 寄存器的值时才能产生中断。						
KBCON.0	KBIF	键盘中断标志。当 P0 匹配了用户在 KBPATN, KBMASK 和 PATN_SEL 中定义的条件时, 该位置位。需要通过软件向其写入 0 清零。						

图 70 键盘控制寄存器

KBMASK 地址: 86H								
不可位寻址	7	6	5	4	3	2	1	0
复位源: 任何复位	KBMASK.7	KBMASK.6	KBMASK.5	KBMASK.4	KBMASK.3	KBMASK.2	KBMASK.1	KBMASK.0
复位值: 00000000B								
位	符号	功能						
KBMASK.7	—	该位置位时, 使能 P0.7 作为键盘中断源						
KBMASK.6	—	该位置位时, 使能 P0.6 作为键盘中断源						
KBMASK.5	—	该位置位时, 使能 P0.5 作为键盘中断源						
KBMASK.4	—	该位置位时, 使能 P0.4 作为键盘中断源						
KBMASK.3	—	该位置位时, 使能 P0.3 作为键盘中断源						
KBMASK.2	—	该位置位时, 使能 P0.2 作为键盘中断源						
KBMASK.1	—	该位置位时, 使能 P0.1 作为键盘中断源						
KBMASK.0	—	该位置位时, 使能 P0.0 作为键盘中断源						

图 71 键盘中断屏蔽寄存器 (KBM)

### 看门狗定时器

看门狗定时器子系统可通过复位使系统从错误的操作中恢复。当软件没能在定时器溢出之前将其清零, 看门狗定时器就会产生复位。看门狗定时器只能通过上电实现复位。

### 看门狗功能

看门狗定时器带有一个片内 400KHz 振荡器。当看门狗使能时, 可以通过配置看门狗控制寄存器 WDCON 中的 WDCLK 位, 选择看门狗振荡器或者 PCLK(见图 72)作为时钟源。当看门狗特性使能时, 定时器必须通过软件定时清零以防止其溢出导致 CPU 复位。

有 4 个 SFR 用于看门狗功能: 看门狗控制寄存器 (WDCON)、看门狗装载寄存器 (WDL)、看门狗清零寄存器 WFEED1 和 WFEED2。

另外还有 Flash 配置字节 UCFG1 中的两个位与看门狗配置相关: 看门狗定时器使能位 (WDTE) 和看门狗安全使能位 (WDSE)。下面的表格详细讲述了这两个位的配置。

表 21 看门狗定时器配置

WDTE (UCFG1.7)	WDSE (UCFG1.4)	功能
0	x	看门狗被禁止。定时器可作为一个内部定时器并可用于产生中断。WDSE 无效。
1	0	看门狗使能。用户可对 WDCLK 进行设置以选择时钟源。
1	1	看门狗使能。还具有以下的安全特性： 1. WDCLK 被强制为 1（使用看门狗振荡器） 2. WDCON 和 WDL 寄存器只可写入一次。 3. WDRUN 被强制为 1。

图 74 所示为看门狗定时器在看门狗模式下工作。它包含一个可编程的 13 位预分频器和一个 8 位倒计数器。该倒计数器以预分频器的节拍作为时钟源。预分频器的时钟源可选择 PCLK 或看门狗振荡器（通过 WDCON 寄存器中的 WDCLK 选择）。（注：对时钟源的切换将不会立即生效——见“切换看门狗的时钟源”一节。）

当看门狗被禁止时，它还可作为一个间隔定时器并可产生中断。

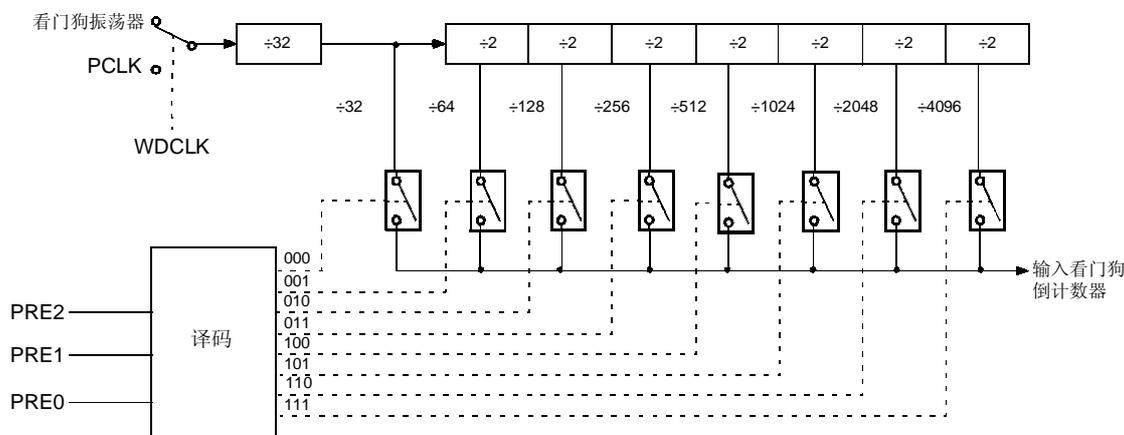


图 72 看门狗预分频器

当看门狗使能且发生向下溢出时，看门狗将请求执行复位。当看门狗使能时，在写入 WDL 或 WDCON 后必须执行一个看门狗清零序列以使新的值生效。

如果发生看门狗复位，内部复位至少保持一个看门狗时钟周期（PCLK 或看门狗振荡器时钟）有效。如果 CCLK 仍然运行，代码将在复位周期之后立即开始执行。如果处理器处于掉电模式，看门狗复位将启动振荡器并在振荡器稳定之后恢复代码的执行。

### 看门狗清零序列

看门狗定时器控制寄存器和 8 位倒计数器(图 73)并不直接由用户装载。用户对 WDCON 和 WDL 进行写操作。在清零序列结束时，WDCON 和 WDL 寄存器的值装入控制寄存器和 8 位倒计数器。在清零序列之前，任何写入这两个寄存器的新值都将无效。为了避免看门狗复位，看门狗定时器必须定时进行清零（通过一个叫做“清零序列”的特殊软件时序）。

要将看门狗清零，必须按顺序执行两条指令。在这两条写指令之间允许读出 SFR 但不允许写操作。这两条指令就是将 A5H 送入 WFEED1 寄存器和将 5AH 送入 WFEED2 寄存器。不正确的清零序列会立即导致看门狗复位。下面是一个清零序列操作实例：

```

CLR    EA                ; 禁止中断
MOV    WFEED1,#0A5h     ; 执行清零第一部分
MOV    WFEED2,#05Ah     ; 执行清零第二部分
SETB   EA                ; 使能中断
    
```

该序列是假设 LPC932 中断系统已经使能且有可能在清零的过程中产生中断请求。如果允许中断响应并且该中断服务程序中包含对任意 SFR 的写操作，这种情况将会触发看门狗复位。如果在清零序列中不会

发生中断，可以取消禁止和重新使能中断的指令。

在看门狗模式中（WDTE=1），写入 WDCON 后必须立即执行一次清零序列以使 WDL 装入 8 位计数器，WDCON 装入映像寄存器。如果不这样操作将立即导致看门狗复位。例如设置 WDRUN=1：

```

MOV    ACC,WDCON        ; 取出 WDCON 的值
SETB   ACC.2            ; 设置 RUN=1
MOV    WDL,#0FFh       ; 装入 8 位倒计数器的新值
CLR    EA               ; 禁止中断
MOV    WDCON,ACC        ; 写回到 WDCON（在看门狗使能后，必须立即执行清零序列）
MOV    WFEED1,#0A5h    ; 执行清零第一部分
MOV    WFEED2,#05Ah    ; 执行清零第二部分
SETB   EA               ; 使能中断
    
```

在定时器模式中（WDTE=0），WDCON 每 CCLK 周期被装入控制寄存器一次（不要求清零序列），但在溢出发生之前，将 WDL 寄存器值装入 8 位计数器时需要执行清零序列。

看门狗控制寄存器（WDCON）

WDCON 地址：A7H									
不可位寻址		7	6	5	4	3	2	1	0
复位源：见下面的复位值		PRE2	PRE1	PRE0	-	-	WDRUN	WDTOF	WDCLK
复位值：111xx1?1B（注：WDCON.7,6,5,2,0 在任何复位时都置 1，WDCON.1 在上电复位时清零，在看门狗复位时置位，不受其它任何复位影响。）									
位	符号	功能							
WDCON.7-5	PRE2-PRE1	时钟预分频器节拍选择，详见表 22。							
WDCON.4-3	-	保留将来之用，用户程序请勿将其置 1。							
WDCON.2	WDRUN	看门狗运行控制。看门狗定时器在 WDRUN=1 时启动并在 WDRUN=0 时停止。该位在 WDTE 和 WDSE 都置位时被强制设置为 1。							
WDCON.1	WDTOF	看门狗溢出标志。该位在 8 位计数器溢出时置位。在看门狗模式中，一个清零序列将清零该位。还可通过软件向其写入 0 将其清零。							
WDCON.0	WDCLK	看门狗输入时钟选择。当该位为 1 时，选择看门狗振荡器作为时钟源；为 0 时，选择 PCLK。（如果 CPU 处于掉电状态，当 WDCLK=0 时，看门狗被禁止，见“WDCLK=0 和 CPU 掉电”一节）。（注：如果 WDTE 和 WDSE 都置位，该位被强制为 1。）详见“切换看门狗时钟源”一节。							

图 73 看门狗定时器控制寄存器

看门狗定时器在溢出前的时钟个数可由下面的等式得到：

$$tclks = (2^{(5+PRE)} + 1)(WDL + 1)$$

其中，

- PRE 为预分频器的值（PRE2-PRE0），范围为 0 到 7。
- WDL 为看门狗装载寄存器的值，范围为 0 到 255。

因此 tclks 的最小个数为：

$$tclks = (2^{(5+0)} + 1)(0 + 1) = 33$$

tclks 的最大个数为：

$$tclks = (2^{(5+0)} + 1)(255 + 1) = 1,048,832$$

下表所示为 LPC932 看门狗溢出值的取样。

表 22 LPC932 看门狗溢出值取样

PRE2-PRE0	WDL (十进制)	溢出周期 (看门狗时钟)	看门狗时钟源	
			400KHz 看门狗振荡器时钟 (正常)	12MHz CCLK(6MHz CCLK/2 看门狗时钟)
000	0	33	82.5µs	5.50µs
	255	8193	20.5ms	1.37 ms
001	0	65	162.5µs	10.8µs
	255	16385	41.0ms	2.73 ms
010	0	129	322.5µs	21.5µs
	255	32769	81.9ms	5.46 ms
011	0	257	642.5µs	42.8µs
	255	65537	163.8ms	10.9 ms
100	0	513	1.28ms	85.5µs
	255	131073	327.7ms	21.8 ms
101	0	1025	2.56ms	170.8µs
	255	262145	655.4ms	43.7 ms
110	0	2049	5.12ms	341.5µs
	255	524289	1.31s	87.4 ms
111	0	4097	10.2ms	682.8µs
	255	1048577	2.62s	174.8 ms

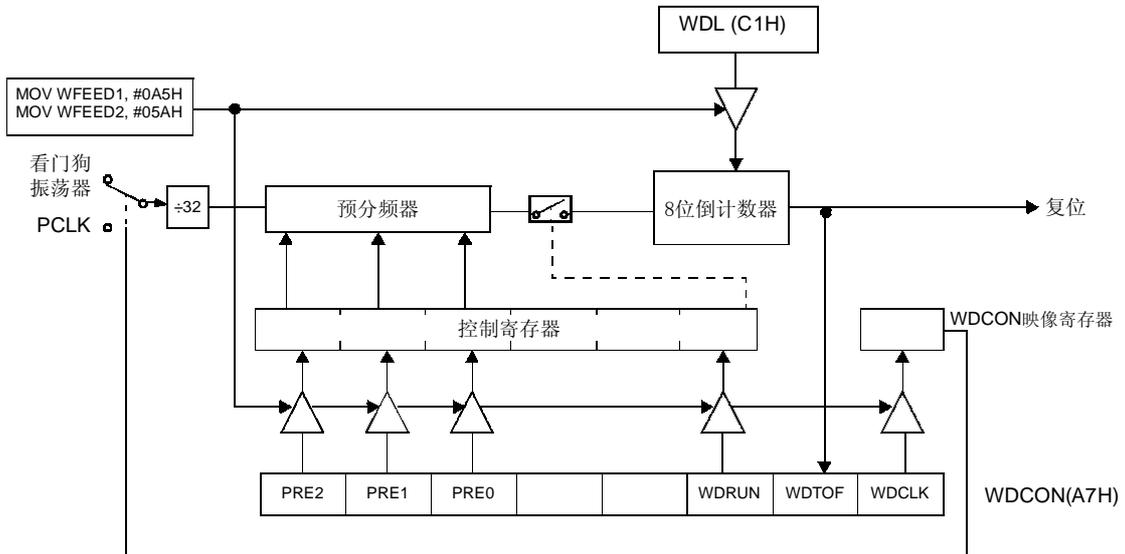


图 74 看门狗模式中的看门狗定时器 (WDTE=1)

定时器模式中的看门狗定时器

图 75 所示为定时器模式中的看门狗定时器。在该模式中，WDCON 的任何改变都将在一个看门狗时钟周期后写入映像寄存器。当定时器超时，产生一次看门狗溢出。超时同时置位 WDTOF 位。如果 EWDRT(IEN0.6)置位，看门狗的溢出将导致中断的产生。WDTOF 通过软件向其写入 0 清零。当发生超时的的时候，WDL 的内容重新装入计数器并且计数器立即重新开始计数。

在超时发生之前，将 WDL 装入计数器仍然需要执行清零序列。错误的清零序列在该模式中将被忽略。

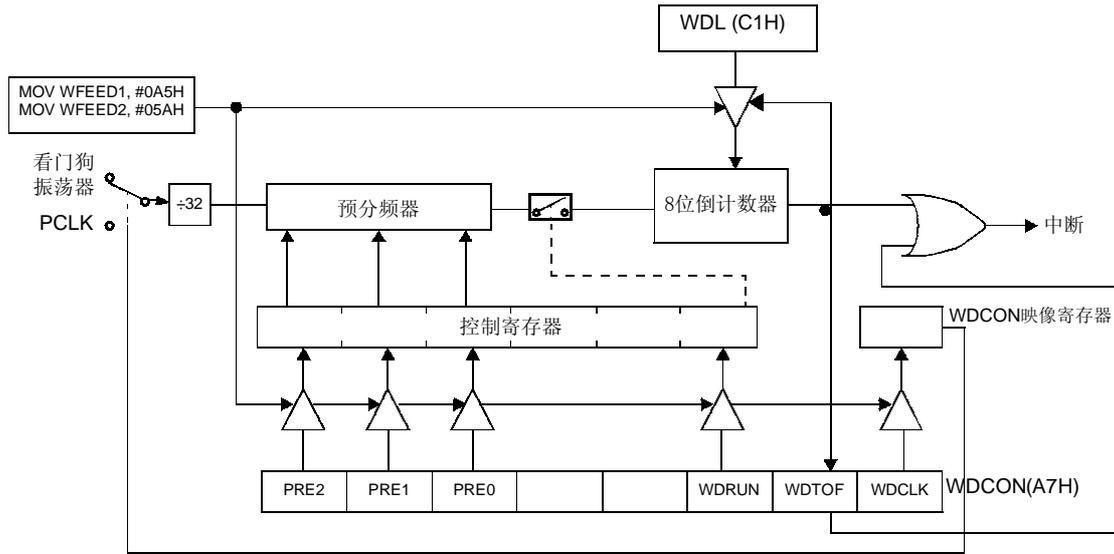


图 75 定时器模式中的看门狗定时器 (WDTE=0)

#### WDCLK=0 和 CPU 掉电

如果 WDCLK=0, PCLK 将作为看门狗的时钟。如果 CPU 处于掉电模式中, PCLK 停止运行, 因此看门狗被禁止。

#### 切换看门狗时钟源

在改变 WDCLK(WDCON.0)之后, 对时钟源的切换不会立即生效。如图 74 所示, 要在一个清零序列之后才将选择装入。此外, 由于时钟同步逻辑的关系, 在放弃旧的时钟源之前还需要经过两个旧的时钟周期。然后需要两个新的时钟周期以使新的时钟有效。

注: 当切换时钟时, 很重要的一点就是, 在完成清零序列后将旧的时钟再保持 2 个时钟周期。否则, 当旧的时钟源禁止时看门狗也被禁止。例如, 假设 PCLK(WCLK=0)为当前的时钟源, 在将 WCLK 设置为 1 后, 程序在清零序列完成后至少应当等待 2 个 PCLK(4 个 CCLK)才能进入掉电模式。否则, 当 CCLK 关闭时, 看门狗也会被禁止。看门狗振荡器永远也不可能成为时钟源, 除非 CCLK 再次打开。

#### 辅助功能

AUXR1 寄存器包含了几个特殊用途的控制位组成, 这些位分别与几种芯片功能相关联。AUXR1 在图 76 中详述。

AUXR1 地址: A2H								
不可位寻址								
复位源: 任何复位	7	6	5	4	3	2	1	0
复位值: 00000x0B	CLKLP	EBRR	ENT1	ENT0	SRST	0	-	DPS
位	符号	功能						
AUXR1.7	CLKLP	时钟低电压选择。置位时降低时钟电路的功耗。当时钟频率为 8MHz 或以下时可置位该位。复位后该位清零支持最大时钟频率 12MHz。						
AUXR1.6	EBRR	UART 间隔检测复位使能。如果该位为 1, UART 间隔检测将导致芯片复位						
AUXR1.5	ENT1	当该位置位时, 定时器 1 的溢出将使 P0.7 发生翻转。因此输出频率为定时器 1 溢出速率的 1/2。详见“定时/计数器”章节。						
AUXR1.4	ENT0	当该位置位时, 定时器 0 的溢出将使 P1.2 发生翻转。因此输出频率为定时器 0 溢出速率的 1/2。详见“定时/计数器”章节。						
AUXR1.3	SRST	软件复位。当 SRST 通过软件置位时, P89LPC932 就像硬件复位一样复位。						
AUXR1.2	0	该位包含一个硬件 0。这样就允许通过将 DPTR 加 1 切换 DPS 位而不影响寄存器的其它位。						
AUXR1.1	-	未用						
AUXR1.0	DPS	数据指针选择。选择两个数据指针中的一个。						

图 76 AUXR1 寄存器

### 软件复位

AUXR1 寄存器的 SRST 位使软件能像发生外部复位或看门狗复位一样彻底复位。如果写 AUXR1.3 位的值为“1”，所有的 SFRS 将初始化并且从地址 0000 开始执行。写入 AUXR1 时，务必小心以防止产生意外的软件复位。

### 双数据指针

双数据指针 (DPTR) 增加了处理器指向确定地址的方式。AUXR1 寄存器的 DPS 位选择两个数据指针中的一个。软件无法访问当前未被选择的 DPTR，除非 DPS 位状态改变。

受数据指针选择影响的指令有：

- INC DPTR
- JMP @A+DPTR
- MOV DPTR,#data16
- MOVC A,@A+DPTR
- MOVX A,@DPTR
- MOVX @DPTR,A

而且任何读取或操纵 DPH 和 DPL 寄存器(DPTR 的高字节和低字节)的指令均受 DPS 位设置的影响。MOVX 指令在 LPC932 中的应用收到限制，因为它没有外部数据总线。但是，它们可用于访问 Flash 配置信息（见 Flash 配置一节）或附加数据存储（XDATA）。

AUXR1 的第 2 位的逻辑电平永远都为“0”，这样 DPS 位可以简单地通过将 AUXR1 加 1 进行切换（因此对数据指针进行切换）而不会更改寄存器的其它位。

### 数据 EEPROM

LPC932 拥有 512 个字节的片内数据 EEPROM。它们可用于保存配置参数。数据 EEPROM 由 SFR 控制，可字节读、字节写以及可擦除（通过行填充和块填充）。用户通过 3 个 SFR 和 1 个中断对其进行读、写和填充：

- 地址寄存器 (DEEADR) 用于地址位 7-0（位 8 位于 DEECON 寄存器）
- 控制寄存器 (DEECON) 用于地址位 8、设置操作模式以及状态位（见图 77）
- 数据寄存器 (DEEDAT) 用于写入或读出数据 EEPROM

DEECON 地址: F1H									
不可位寻址		7	6	5	4	3	2	1	0
复位源: 任何复位		EEIF	HVERR	ECTL.1	ECTL.0	-	-	-	EADR8
复位值: 0x00xxx0B									
位	符号	功能							
DEECON.7	EEIF	数据 EEPROM 中断标志。当读或写操作完成时该位置位。由软件复位。							
DEECON.6	-	保留将来之用。用户程序请勿将其置 1。							
DEECON.5-4	ECTL.1,ECTL.0	操作模式选择							
	ECTL.1,ECTL.0	操作模式							
	00	字节读/写模式							
	10	行 (64 字节) 填充							
	11	块 (512 字节) 填充							
DEECON3-1	-	保留将来之用。用户程序请勿将其置 1。							
DEECON.0	EADR8	数据 EEPROM 地址最高位。EADR7-0 位于 DEEADR 中。							

图 77 数据 EEPROM 控制寄存器

字节模式: 在该模式中，一次可以读和写一个字节数据。数据存放在 DEEDAT 寄存器而地址在 DEEADR 寄存器中。

行填充: 在该模式中, 被寻址的行 (64 字节, 地址 DEEADR.5-0 被忽略) 按照 DEEDAT 的格式填充。要将整个行都擦除为 00h 或将整个行编程为 FFh, 需要在进行行填充之前将 00h 或 FFh 写入 DEEDAT。

块填充: 在该模式中, 所有 512 个字节都按照 DEEDAT 的格式填充。为了将整个块都擦除为 00h 或将整个块编程为 FFh, 需要在进行块填充之前将 00h 或 FFh 写入 DEEDAT。使用该命令之前 EADR8 必须为 1。

在任何模式中, 操作完成之后硬件都会置位 EEIF 位。如果 IEN1.7 和 EA 都置位, 将产生中断请求。EEIF 位通过软件清零。

#### 数据 EEPROM 读

字节的读取可通过查询或中断进行:

1. 将 DEECON 中的 ECTL1-0(DEECON.5-4) 写为 00 并将正确的地址第 8 位写入 EADR (注: 如果正确的值已写入 DEECON, 就不再需要对该寄存器进行写操作)
2. 不需要对 DEEDAT 进行写操作, 将地址位 7~0 写入 DEEADR。
3. 如果 EIEE(IEN1.7) 和 EA(IEN0.7) 都为 1, 等待中断, 然后读取/查询 EEIF(DEECON.7) 位直到它变为 1。如果 EIEE 或 EA 为 0, 中断被禁止, 就只能使用查询的方式。
4. 从 DEEDAT 寄存器读出 EEPROM 数据。

注: 如果在写 DEEADR 之前(如果 DEECON.5-4=00)写 DEEDAT, 将开始执行对 EEPROM 的写操作。用户必须注意在读操作时避免出现这样的情况。

#### 数据 EEPROM 写

字节的写入可通过查询或中断进行:

1. 将 DEECON 中的 ECTL1-0(DEECON.5-4) 写为 00 并将正确的地址第 8 位写入 EADR (注: 如果正确的值已写入 DEECON, 就不再需要对该寄存器进行写操作)
2. 将数据写入 DEEDAT
3. 将地址位 7~0 写入 DEEADR。
4. 如果 EIEE(IEN1.7) 和 EA(IEN0.7) 都为 1, 等待中断, 然后读取/查询 EEIF(DEECON.7) 位直到它变为 1。如果 EIEE 或 EA 为 0, 中断被禁止, 就只能使用查询的方式。当 EEIF 为 1 时, 操作完成, 数据被写入。

由于在写入 DEEDAT 寄存器后写 DEEADR 寄存器将立即执行写操作 (如果 DEECON.5-4=00), 因此用户必须在写 DEEDAT 寄存器的时候要非常小心。强烈建议用户在写 DEEDAT 寄存器之前禁止中断并在所有写操作执行完毕之后重新使能中断。例如:

```
CLR    EA            ; 禁止中断
MOV    DEEDAT,@R0   ; 写入数据格式
MOV    DEEADR,@R1   ; 写入地址
SETB   EA            ; 使能中断,如果 IEN1.7(EEIE)位置位,则等待中断并查询 DEECON.7(EEIF)
```

#### 硬件复位

在任何硬件复位 (包括看门狗和系统定时器复位) 时, “记忆” DEEDAT 写操作的状态机将被初始化。如果在硬件复位发生在一个对 DEEDAT 的写操作之后, 那么对 DEEADR 寄存器的写操作将导致一次对 EEPROM 的读操作。

#### 对 DEEDAT 寄存器的多次写操作

如果在写入 DEEADR 寄存器之前对 DEEDAT 寄存器执行了多次写入, 那么最后写入 DEEDAT 的数据将会写入到相应地址。

### 写 DEECON 和 DEEDAT 寄存器的顺序

如果 DEECON.5-4=00, 那么对 DEEDAT 寄存器的写操作将被认为是有效的 (即触发状态机开始执行写操作)。如果这些模式位已经为 00 并且地址位 8 也正确, 那么在写入 DEEDAT 寄存器之前就不需要写 DEECON 寄存器。

### 数据 EEPROM 行填充

使用预先确定的数据格式填充一行 (64 个字节)。可通过查询或中断进行:

1. 将 DEECON 中的 ECTL1-0(DEECON.5-4)写为 10 并将正确的地址第 8 位写入 EADR (注: 如果正确的值已写入 DEECON, 就不再需要对该寄存器进行写操作)
2. 将填充格式写入 DEEDAT (如果正确的值已经写入 DEEDAT, 就不需要对该寄存器进行写操作)。
3. 将地址位 7~0 写入 DEEADR。注: 地址 5-0 忽略。
4. 如果 EIEE(IEN1.7)和 EA(IEN0.7)都为 1, 等待中断, 然后读取/查询 EEIF(DEECON.7)位直到它变为 1。如果 EIEE 或 EA 为 0, 中断被禁止, 就只能使用查询的方式。当 EEIF 为 1 时, 操作完成, 整个行以 DEEDAT 的格式填充。

### 数据 EEPROM 块填充

使用预先确定的数据格式填充整个 EEPROM 数据块。可通过查询或中断进行:

1. 将 DEECON 中的 ECTL1-0(DEECON.5-4)写为 11, EADR8=1。
2. 将填充格式写入 DEEDAT。
3. 将任意地址写入 DEEADR。注: 整个地址都被忽略。
4. 如果 EIEE(IEN1.7)和 EA(IEN0.7)都为 1, 等待中断, 然后读取/查询 EEIF(DEECON.7)位直到它变为 1。如果 EIEE 或 EA 为 0, 中断被禁止, 就只能使用查询的方式。当 EEIF 为 1 时, 操作完成。

## FLASH 程序存储器

### 概述

LPC932 Flash 存储器提供电路中的电擦除和编程。Flash 可以字节为单位读取或写入。芯片擦除功能可实现整个程序存储器的擦除。扇区和页擦除功能可擦除任意的 Flash 扇区 (1KB) 或页 (64 字节)。可提供在系统编程和标准的并行编程。片内产生的擦除和写入时序为用户提供了友好的编程接口。LPC932 Flash 存储器甚至在经过 10,000 次擦除和编程之后仍然能可靠地保存存储器的内容。存储单元的设计使擦除和编程结构最优化。LPC932 使用  $V_{DD}$  电压来执行编程和擦除算法。

### 特性

- 内部固化的引导 ROM, 包含了一个低级的在应用编程(IAP)子程序
- 用户可调用这些程序来实现在应用中编程(IAP)。
- 默认的装载程序可通过串口进行在系统编程。该程序位于程序存储器空间的顶端。
- Boot 向量允许用户将 Flash 装载代码放入 Flash 存储器内的任何位置。这种配置为用户提供了应用的灵活性。
- 可在整个操作电压范围内执行编程和擦除
- 使用 ISP/IAP 进行读/编程/擦除
- 任意编程/擦除时间小于 2ms
- 使用工业标准的商用编程器进行并行编程
- 在系统编程
- 可对每一个 Flash 扇区进行编程加密
- 每个字节至少可执行 10,000 次擦除/编程

- 数据至少可保存 10 年

## LPC932 的 ISP&IAP 性能

### Flash 的结构

LPC932 包含 8 个 1KB 扇区的 Flash 程序存储器。每个扇区可进一步分成 64 字节的页。除了扇区擦除和页擦除外，还包含一个 64 字节页寄存器，它可实现给定页 1 到 64 字节的同时编程，这彻底降低了整个编程的时间。在应用中编程 (IAP) 接口可使用户在最终应用程序中对用户代码进行擦除和重新编程。此外，还支持对用户可编程字节 (包括 UCFG1、状态字节和引导向量) 的擦除和重新编程。芯片出厂后，用户代码空间的高 512 字节包含了一个串行在系统编程 (ISP) 子程序，允许器件在电路中通过串行口进行编程。

### Flash 的编程和擦除

有三种方法可实现对 Flash 的编程或擦除。第一，在最终用户应用程序中通过一个公共入口调用低级程序对 Flash 进行编程和擦除。第二，调用片内 ISP 装载程序，该装载程序调用引导 ROM 中的低级子程序 (与 IAP 编程共用)。第三，使用支持该器件的商用编程器进行并行编程或擦除。该器件不提供对代码内容的直接校验。而是提供一个扇区或页的 32 位 CRC 结果。

### 引导 ROM

当微控制器对自身的 Flash 存储器进行编程时，所有操作的细节都由固化在 256 字节引导 ROM 中的代码进行处理。引导 ROM 与 Flash 存储器是各自独立的。用户程序使用适当的参数调用引导 ROM 中的公共入口即可实现所需要的操作。引导 ROM 的操作包括以下内容：擦除程序扇区，擦除页，编程页，CRC，编程保密位等等。引导 ROM 占用 FF00 到 FFFFH 的程序存储器的地址。因此不与用户程序代码空间冲突。

### 上电复位代码的执行

LPC932 包含两个特殊的 Flash 寄存器：引导向量和状态字节 (注意：它们不在 SFR 中而在 FLASH 空间里)。在复位信号的下降沿，LPC932 检查状态字节中的内容。如果为 0，则转去 0000H 地址开始执行程序。这是用户应用代码的正常起始地址。如果状态字节不为 0，则将引导向量的值作为程序计数器的高字节，低字节固定为 00H。工厂的默认设定为 01EH，对应为默认的 ISP 装载程序地址 01E00H。该引导装载程序在工厂已经预先编程并可被用户擦除。用户如果需要使用这部分代码，就必须小心以避免将 1C00H 到 1FFFH 的 1KB 扇区擦除。使用页擦除功能可擦除从 1C00H 到 1DFFH 的 8 个 64 字节的页。如果需要的话，可以将引导向量设置到用户的引导装载程序入口地址。

### 引导装载程序的硬件激活

在上电时，通过强制器件进入 ISP 模式也可执行引导装载程序 (见图 78)。在上电时，复位脚拉低，并在 VDD 上升到正常工作电压之后使 RST 脚保持一段固定时间长度的低电平，之后是 3 个下降沿脉冲。少于 3 个或多于 3 个脉冲都不会使器件进入 ISP 模式。时序的规格可参阅器件手册。

这和非零状态字节的效果是一样的。这样就可以在正常时执行用户代码，但又可以手动强制进入 ISP 操作。如果改变引导向量出厂时默认的设定 1EH，将不再指向工厂预先编程的 ISP 引导装载程序。如果发生这种情况，改变引导向量的唯一方法就是使用并行编程。并行编程提供用户应用程序所不包含的定制的装载程序，以实现对引导向量和状态字节的擦除和编程。在对 Flash 进行编程后，状态字节可编程为 0，以允许用户应用代码从地址 0000H 开始执行。

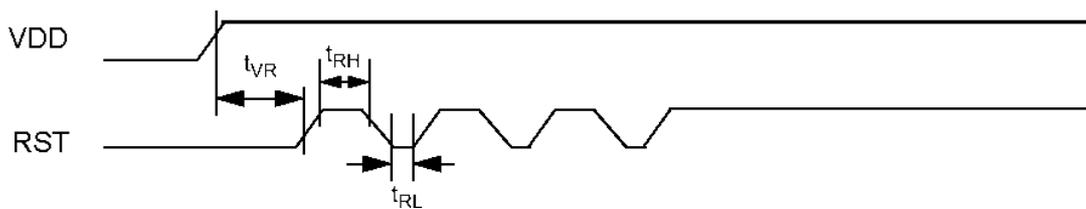


图 78 强制进入 ISP 模式

**在系统编程(ISP)**

ISP 编程执行时不需要将微控制器从系统中移出。在系统编程特性包含了一系列内部的硬件资源，与内部固件相结合可实现通过串口对 LPC932 的远程编程。固件由 Philips 提供并嵌入到每一个 LPC932 芯片当中。Philips 的在系统编程(ISP)特性使嵌入式应用中的在电路编程变得可行，并最大限度减小了额外的元件开销和电路板面积。ISP 功能使用芯片的 5 个管脚。只需要一个小的连接器就可实现通过外部电路使用该特性的应用接口。

**使用在系统编程(ISP)**

ISP 特性可以适应用户应用中较宽的范围的波特率而不依赖于振荡器的频率。它还可以适应较宽范围的振荡器频率。这是通过对一个接收到的字符中单个位进行位时间测量来实现的。该信息可用于设定定时器值从而对波特率进行编程。为此 ISP 特性要求主机必须向 LPC932 发送一个初始字符（大写的 U）以使芯片对波特率进行设定。ISP 固件提供对接收字节的自动回送。一旦波特率的设定完成之后，ISP 固件将只接受 Intel-Hex 格式的记录。Intel Hex 格式的记录包含了用于表示十六进制数的 ASCII 字符。汇总如下：

: NNAAAARRDD..DDCC<crLf>

在 Intel Hex 格式中，‘NN’表示记录中数据字节的个数。LPC932 最多可接受 64 (40H) 个数据字节。‘AAAA’字符串表示记录中首字节的地址。如果记录中的字节数为 0，通常该区域设置为 ‘0000’。‘RR’表示记录的类型：‘00’表示数据类型，‘01’表示文件结束标识。在该应用中，增加额外的记录格式以指示 ISP 应用的命令或数据。数据字节的最大数目限制为 64（十进制）。ISP 命令汇总表 23。LPC932 收到记录后，将数据保存并计算校验和。在整个记录接收完毕后才执行由记录类型所指示的操作。如果校验和发生错误，LPC932 将从串口发出 ‘X’ 指示校验和出错。如果校验结果与记录相符，将会执行指定的命令操作。在大多数情况下，从串口发出 ‘.’ 表示正确接收。

表 23 在系统编程 (ISP) Hex 记录格式

记录类型	命令/数据 功能
00	编程用户代码数据储存器页 :nnaaaa00dd... ddcc 其中： nn = 编程的字节数 (hex) aaaa = 页地址 dd... dd = 数据字节 cc = 校验和 例: 1000000000102030405006070809cc
01	文件结束，读版本 Id :00xxxx01cc 其中： xxxx = 必须的字段，但其值“无关” cc = 校验和 例: 00000001cc

02	<p>混合写功能 :02xxxx02ssddcc</p> <p>其中: xxxx = 必须的字段, 但其值“无关”          ss = 子功能代码          dd = 数据          cc = 校验和</p> <p>子功能代码:          00 = UCFG1          01 = 保留          02 = 引导向量          03 = 状态字节          04 = 保留          05 = 保留          06 = 保留          07 = 保留          08 = 保密字节 0          09 = 保密字节 1          0A = 保密字节 2          0B = 保密字节 3          0C = 保密字节 4          0D = 保密字节 5          0E = 保密字节 6          0F = 保密字节 7</p> <p>例: 020000020347cc</p>
03	<p>混合读功能 :01xxxx03sscc</p> <p>其中:          xxxx = 必须的字段, 但其值“无关”          ss = 子功能代码          cc = 校验和</p> <p>子功能代码:          00 = UCFG1          01 = 保留          02 = 引导向量          03 = 状态字节          04 = 保留          05 = 保留          06 = 保留          07 = 保留          08 = 保密字节 0          09 = 保密字节 1          0A = 保密字节 2          0B = 保密字节 3          0C = 保密字节 4          0D = 保密字节 5          0E = 保密字节 6          0F = 保密字节 7          10 = 厂商 ID          11 = 器件 ID          12 = 派生器件 ID</p> <p>例: 0100000312cc</p>

04	擦除扇区/页 : 03xxxx04ssaaaacc 其中 xxxx = 必须的字段, 但其值“无关” aaaa = 扇区/页地址 ss = 00 擦除扇区 = 01 擦除页 cc = 校验和 例: 03000004010000F8
05	读扇区 CRC : 01xxxx05aacc 其中 xxxx = 必须的字段, 但其值“无关” aa = 扇区地址高字节 cc = 校验和 例 :0100000504F6cc
06	读全局 CRC : 00xxxx06cc 其中 xxxx = 必须的字段, 但其值“无关” cc = 校验和 例 :00000006cc
07	直接装载波特率 : 02xxxx07HLLcc 其中 xxxx = 必须的字段, 但其值“无关” HH = 定时器高字节 LL = 定时器低字节 cc = 校验和 例: 02000007FFFcc
08	整片擦除 : 00xxxx08cc 其中 xxxx = 必须的字段, 但其值“无关” cc = 校验和 例 :00000008cc

### 在应用中编程的方法

通过一个应用程序对几个 IAP 子程序的调用, 可实现有选择地对 Flash 扇区、页、保密位、配置字节和器件 ID 进行擦除和编程。所有的调用都是通过一个公共的接口 PGM\_MTP 来实现的。在调用位于地址 FF00H 的 PGM\_MTP 之前, 通过设置微控制器的寄存器可选择编程的功能。IAP 调用见表 24。

表 24 IAP 功能调用

IAP 功能	IAP 调用参数
编程用户代码页	输入参数: ACC = 00h R3 = 编程字节数 R4 = 页地址 (高字节) R5 = 页地址 (低字节) R7 = RAM 中数据缓冲区的指针 返回参数 R7 = 状态 进位位 (C) = 置位表示错误, 清零表示无错误
读版本 ID	输入参数: ACC = 01h 返回参数 R7 = IAP 代码版本 ID

混合写	<p>输入参数:                  ACC = 02h                  R5 = 写入的数据                  R7 = 寄存器地址                      00 = UCFG1                      01 = 保留                      02 = 引导向量                      03 = 状态字节                      04 = 保留                      05 = 保留                      06 = 保留                      07 = 保留                      08 = 保密字节 0                      09 = 保密字节 1                      0A = 保密字节 2                      0B = 保密字节 3                      0C = 保密字节 4                      0D = 保密字节 5                      0E = 保密字节 6                      0F = 保密字节 7</p> <p>返回参数                  R7 = 状态                  进位位 (C) = 置位表示错误, 清零表示无错误</p>
混合读	<p>输入参数:                  ACC = 03h                  R7 = 寄存器地址                      00 = UCFG1                      01 = 保留                      02 = 引导向量                      03 = 状态字节                      04 = 保留                      05 = 保留                      06 = 保留                      07 = 保留                      08 = 保密字节 0                      09 = 保密字节 1                      0A = 保密字节 2                      0B = 保密字节 3                      0C = 保密字节 4                      0D = 保密字节 5                      0E = 保密字节 6                      0F = 保密字节 7</p> <p>返回参数                  R7 = 若无错误则为寄存器数据, 否则为错误状态                  进位位 (C) = 置位表示错误, 清零表示无错误</p>
擦除扇区/页	<p>输入参数:                  ACC = 04h                  R7 = 70h(擦除页)或 71H(擦除扇区)                  R4 = 扇区/页地址 (高字节)                  R5 = 扇区/页地址 (低字节)</p> <p>返回参数                  R7 = 状态                  进位位 (C) = 置位表示错误, 清零表示无错误</p>

读扇区 CRC	输入参数: ACC = 05h R7 = 扇区地址 返回参数: R4 = CRC 位 31:24 R5 = CRC 位 23:16 R6 = CRC 位 15:8 R7 = CRC 位 7:0(若无错误) R7 = 错误状态(如果错误) 进位位(C) = 置位表示错误, 清零表示无错误
读全局 CRC	输入参数: ACC = 06h 返回参数 R4 = CRC 位 31:24 R5 = CRC 位 23:16 R6 = CRC 位 15:8 R7 = CRC 位 7:0(若无错误) R7 = 错误状态(如果错误) 进位位(C) = 置位表示错误, 清零表示无错误
读用户代码	输入参数: ACC = 07h R4 = 地址(高字节) R5 = 地址(低字节) 返回参数 R7 = 数据

#### 用户配置字节

LPC932 的众多用户可配置的特性必须在上电时定义, 开始执行程序后便不可设置了。这些特性是通过配置 Flash 字节 UCFG1 实现(见图 79)。

UCFG1 地址: xxxh		7	6	5	4	3	2	1	0
未编程值: 00h		WDTE	RPE	BOE	WDSE	-	FOSC2	FOSC1	FOSC0
位	符号	功能							
UCFG1.7	WDTE	看门狗定时器使能位, 当置 0 时, 则关闭看门狗定时器, 定时器可用来作产生中断, 反之, 看门狗定时器使能。							
UCFG1.6	RPE	复位脚使能, 当编程(置 1)时, 使能 P1.5 的复位功能。							
UCFG1.5	BOE	掉电检测使能(见“掉电检测”一节)							
UCFG1.4	WDSE	看门狗使能位。详见表 21。							
UCFG1.3	-	保留(保持其为未编程状态 0)							
UCFG1.2-0	FOSC2-FOSC0	CPU 振荡器类型选择。详见“时钟”一节。除了下面给出的组合外, 其它值不能使用。它们被保留供将来使用。							
	<u>FOSC2-FOSC0</u>	<u>振荡器配置</u>							
	1 1 1	从 XTAL1 输入的外部时钟(为未编程时默认的设置)							
	1 0 0	看门狗振荡器, 400KHz (+20%, -30% 误差)							
	0 1 1	内部 RC 振荡器 7.373MHz (±2.5% 误差)							
	0 1 0	低频晶振, 20KHz 至 100KHz							
	0 0 1	中频晶振或谐振器, 100KHz 至 4MHz							
	0 0 0	高频晶振或谐振器, 4MHz 至 12MHz							

图 79 Flash 用户配置字节 1 (UCFG1)

#### 用户扇区保密字节

LPC932 具有 8 个用户扇区保密字节 (SEC0, ..., SEC7), 每个字节都对应一个扇区并具有下面的位分配:

SECx 地址: xxxxh									
		7	6	5	4	3	2	1	0
		-	-	-	-	-	GEDISx	SPEDISx	MOVCDISx
未编程值: 00h									
位	符号	功能							
SECx.7-3	-	保留(应使其保持为未编程状态 0)							
SECx.2	GEDISx	禁止 ISP/IAP 对扇区 x 进行整体擦除。当通过编程接口执行整体擦除命令时, 该位和扇区 x 都被擦除。在 ISP/IAP 模式中, 不能通过整体擦除命令将该位和扇区 x 擦除。要擦除该位, 器件必须退出应用程序并通过编程器进行擦除。							
SECx.1	SPEDISx	禁止对全部或部分扇区 x 进行编程或擦除。当在 ISP/IAP 模式中或通过编程接口执行整体擦除命令时, 该位和扇区 x 被擦除。							
SECx.0	MOVCDISx	禁止对扇区 x 的 MOVC 操作。任何对受保护的扇区进行访问的 MOVC 指令都将返回无效数据。只有当扇区 x 被擦除时才能将该位擦除。							

图 80 用户扇区保密字节 (SEC0, ..., SEC7)

引导向量

BOOTVEC 地址: xxxxh									
		7	6	5	4	3	2	1	0
		-	-	-	BOOTV4	BOOTV3	BOOTV2	BOOTV1	BOOTV0
未编程值: 00h									
位	符号	功能							
BOOTVEC.7-5	-	保留(保持其为未编程状态 0)							
BOOTSTAT4.0	-	引导向量。如果选择引导向量作为复位地址, LPC932 将从引导向量处 (BOOTVEC 作为地址高字节, 00h 作为地址低字节) 开始执行程序 (见“复位向量”一节)。							

图 81 引导向量 (BOOTVEC)

引导状态

BOOTSTAT 地址: xxxxh									
		7	6	5	4	3	2	1	0
		-	-	-	-	-	-	-	BVS
未编程值: 00h									
位	符号	功能							
BOOTSTAT.7-1	-	保留(保持其为未编程状态 0)							
BOOTSTAT.0	BVS	引导向量选择。如果该位编程为 1, LPC932 将一直从引导向量处 (BOOTVEC 作为地址高字节, 00h 作为地址低字节) 开始执行程序 (见“复位向量”一节)。							

图 82 引导状态 (BOOTSTAT)

## P89LPC932 勘误

典型的 P89LPC932 器件具有下面的顶部标识:

```
P89C932x  x
xxxxxxx  xx
xxYYWW R
```

第三行最后一个字母“R”用来识别对器件的修正。这次勘误包括了对下列 P89LPC932 的修正:

修正标识	备注
'C'	

YY 表示制造的年份，WW 表示该器件是在当年的第几个星期制造的。

### 功能性错误

**CCU. 1: CCU 处于对称模式，同时使用了 CCU 预分频器**

#### 介绍

CCU 是捕获比较单元，它具有产生对称 PWM 信号的选项。10 位预分频器可用于调整 CCU 定时器的时钟频率。

#### 问题

当 CCU 处于对称模式时使用了 10 位预分频器（寄存器 TRCR2），那么 CCU 中断将不起作用，即通用 CCU 中断标志 ECCU 不会置位。这样重装寄存器 TOR2 和输出比较寄存器 OCRx 的内容将无法更新。

#### 解决办法

在对称模式中，不要将预分频器 TRCR 设置为除 0 以外的值。

**CCU. 2: 基本定时器模式中的 CCU 输出比较**

#### 问题

当处于输出比较状态模式“01”和“10”时，管脚无输出信号。

#### 解决办法

暂无

**CCU. 3: CCU 基本定时器模式中的强制比较位 FCOx**

#### 问题

强制比较位 FCOx 无效。

#### 解决办法

暂无

**PWRDWN. 1: 进入掉电模式**

#### 介绍

掉电模式通过将 CPU、外围功能和晶振关闭使功耗降至最低。

#### 问题

当选择高频晶振和外部时钟源时，AUXR1 寄存器中的 CLKLP 位必须在进入掉电模式之前置位，这样才能使功耗降到最低。

#### 解决办法

在进入掉电模式之前将 DIVM 寄存器设置为 1 使时钟频率减半。然后置位 CLKLP 位可实现最低功耗。

**TMRO/1. 1: 空闲模式中的定时器 0/1 输出翻转**

### 介绍

定时器 0/1 在溢出时可触发管脚翻转。空闲模式通过停止 CPU 运行来降低功耗，但所有外围功能都保持运行。

### 问题

当进入空闲模式时，定时器输出管脚翻转不起作用。

解决办法

暂无

ISP. 1: 通过 Reset 管脚上的 3 个脉冲进入 ISP 模式

### 介绍

在系统编程 (ISP) 可用于对 Flash 存储器进行重新编程。

### 问题

在上电复位后，通过 Reset 管脚上的 3 个脉冲无法正确进入 ISP 模式。

解决办法

可通过 AUXR1 寄存器使能间隔检测功能来进入 ISP 模式。当 UART 检测到一个间隔信号，部件将进入 ISP 模式。

### 电气和时序规格错误

f<sub>CLK, 1</sub>: 2.4V < VDD < 2.7V 时的最大操作频率

### 介绍

P89LPC932 的操作范围为 2.4V~3.6V 和 0~12MHz。

### 问题

P89LPC932 在高于内部 RC 振荡器频率 7.37MHz 运行时，操作电压必须高于 2.7V。频率在 7.37MHz 以下时，P89LPC932 的操作电压可低至 2.4V。

解决办法

暂无