

P87LPC778 OTP 单片机数据手册

1. 概述

P87LPC778 是一款 20 脚单片封装的微控制器，主要用在要求高集成度、低成本的情况，可以满足多方面的性能要求。作为 PHILIPS 小型封装系列中的一员，P87LPC778 包含一个 4 路 8 位的 A/D 转换器、可编程振荡器（可配置为高速和低速晶振或 RC 振荡方式）、宽范围的工作电压、可编程 I/O 口输出配置、可选择施密特触发输入、LED 驱动输出和一个内置的看门狗定时器。P87LPC778 的基于 80C51 加速处理器的结构使其指令执行速度为标准 80C51 器件的两倍。

2. 特性

- 当操作频率为 20MHz 时，除乘法和除法指令外，加速 80C51 CPU 的指令执行时间为 300~600ns。
- 用于数字功能时，工作电压范围为 2.7V~5.5V。
- 4 路 10 位的脉宽调制器。
- 4 路 8 位的 A/D 转换器，20MHz 的晶振频率下转换时间为 9.3us。
- I²C 通信端口和全双工 UART。
- 内部振荡器 2.5%。选择使用内部振荡器时无需外接振荡器件。
- 2 个模拟比较器。
- 8 个键盘中断输入，另外附加 2 个外部中断输入。
- 看门狗定时器具有独立的片内振荡器，无需外接元件。看门狗溢出时间有 8 种选择。
- 20 脚 TSSOP 封装。

3. 订购信息

表 1 订购信息

货品号	封装			
	名称	描述	温度	版本
P87LPC778FDH	TSSOP20	塑料缩小型封装；20 脚；本体宽度 4.4mm	-40℃~+85℃	SOT360-1

4. 功能框图

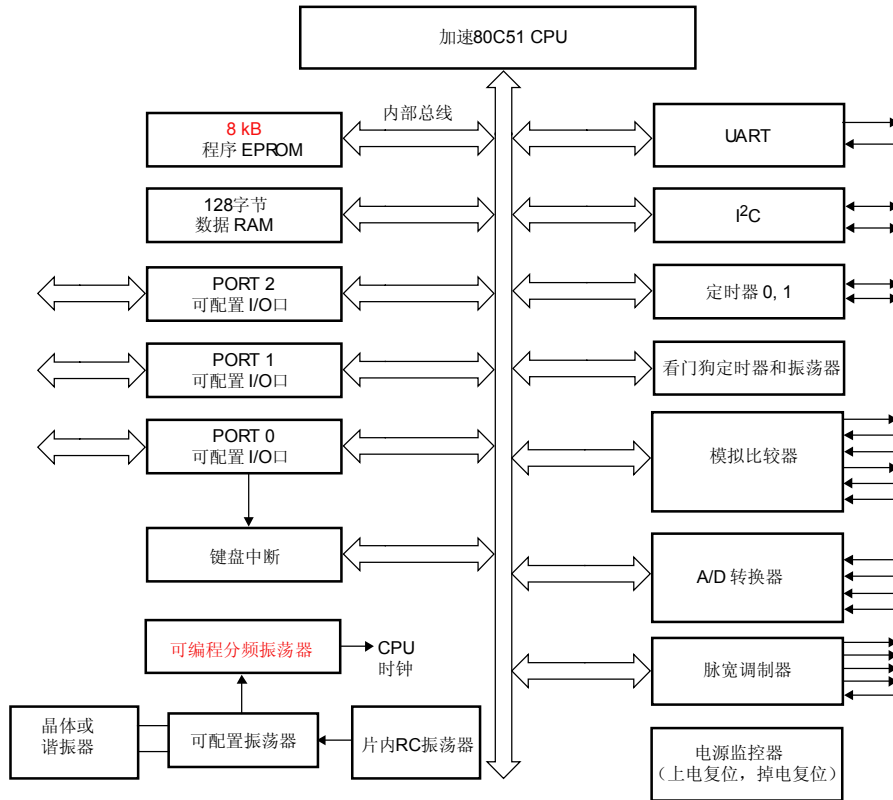
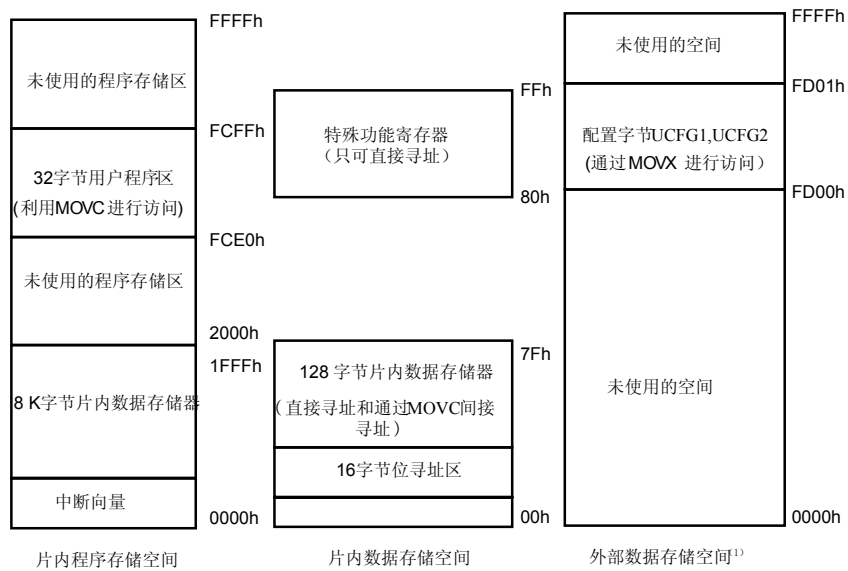


图 1 方框图



(1) P87LPC778 不支持对外部数据存储器的访问。但是，如果用户配置字节存放在外部数据存储器中，可利用 MOVX 指令将其读出。

图 2 存储器分布图

5. 管脚

5.1 管脚配置

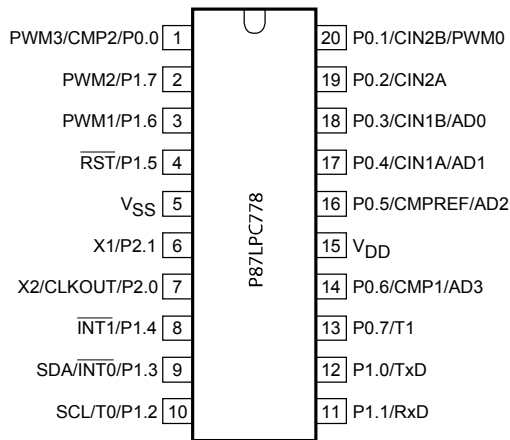


图 3 20 脚 DIP 和 SO

5.2 管脚描述

表 2 管脚描述

符号	管脚	类型	描述
P0.0~P0.7	1,20-16, 14,13	I/O	P0 口 ：P0 口是一个可由用户配置输出类型的 8 位 I/O 口。P0 口锁存器在准双向模式中配置，由 UCFG1 配置字节的 PRHI 位来决定复位时写入 ‘0’ 还是 ‘1’。P0 口可用作输入或输出，取决于所选的端口配置。每个 I/O 口可单独配置。详情参考第 8.9 节 “I/O 口” 和表 67 “DC 电气特性”。 P0 口还具有键盘中断输入功能。 P0 口提供了以下特殊功能：
P0.0	1	O	CMP2 —比较器 2 输出。
		O	PWM3 —PWM 输出 3。
P0.1	20	I	CIN2B —比较器 2 正向输入 B。
		O	PWM0 —PWM 输出 0。
P0.2	19	I	CIN2A —比较器 2 正向输入 A。
P0.3	18	I	CIN1B —比较器 1 正向输入 B。
		I	AD0 —A/D 通道 0 输入。
P0.4	17	I	CIN1A —比较器 1 正向输入 A。
		I	AD1 —A/D 通道 1 输入。
P0.5	16	I	CMPREF —比较器参考（负向）输入。
		I	AD2 —A/D 通道 2 输入。
P0.6	14	O	CMP1 —比较器 1 输出。
		I	AD3 —A/D 通道 3 输入。
P0.7	13	I/O	T1 —定时器/计数器 1 外部计数输入或溢出输出。

续上表...

符号	管脚	类型	描述
P1.0~P1.7	12-8,4,3, 2	I/O	P1 口 : 除下面特别说明的 3 个管脚外, P1 口其它管脚的输出类型可由用户进行配置。P1 口锁存器在准双向模式中配置, 由 UCFG1 配置字节的 PRHI 位来决定复位时写入 ‘0’ 还是 ‘1’。P1 口可用作输入或输出, 取决于所选的端口配置。每个 I/O 口可单独配置。详情参考第 8.9 节 “I/O 口” 和表 67 “DC 电气特性”。 P1 口提供了以下特殊功能:
P1.0	12	O	TxD —串口输出。
P1.1	11	I	RxD —串口输入。
P1.2	10	I/O	T0 —定时器/计数器 0 外部计数输入或溢出输出。
		I/O	SCL —I ² C 总线串行时钟输入/输出。为了符合 I ² C 总线规范, P1.2 配置成开漏输出模式。
P1.3	9	I	INT0 —外部中断 0 输入。
		I/O	SDA —I ² C 总线串行数据输入/输出。为了符合 I ² C 总线规范, P1.3 配置成开漏输出模式。
P1.4	8	I	INT1 —外部中断 1 输入。
P1.5	4	I	RST —外部复位输入 (可通过 EPROM 配置来选择)。一旦该管脚上出现低电平, 微控制器复位、I/O 口和外围器件进入默认状态、处理器从地址 0 开始执行。当该管脚用作 I/O 口时, P1.5 仅为施密特输入。
P1.6	3	O	P1.6 —P1 口位 6。
		O	PWM1 —PWM 输出 1。
P1.7	2	O	P1.7 —P1 口位 7。
		O	PWM2 —PWM 输出 2。
P2.0~P2.1	7,6	I/O	P2 口 : P2 口是一个可由用户配置输出类型的 2 位 I/O 口。P2 口锁存器在准双向模式中配置, 由 UCFG1 配置字节的 PRHI 位来决定复位时写入 ‘0’ 还是 ‘1’。P2 口可用作输入或输出, 取决于所选的端口配置。每个 I/O 口可单独配置。详情参考第 8.9 节 “I/O 口” 和表 67 “DC 电气特性”。 P2 口提供以下特殊功能:
P2.0	7	O	X2 —振荡放大器的输出 (通过 EPROM 配置选择晶体振荡器时)。
		O	CLKOUT —如果通过相应的 SFR 位使能并且使用内部 RC 振荡或外部时钟输入时, CPU 时钟被 6 分频后输出。
P2.1	6	I	X1 —振荡电路和内部时钟发生电路的输入 (通过 EPROM 配置选择时)。
V _{SS}	5	I	地 : 0V 参考点。
V _{DD}	15	I	电源 : 正常工作模式、空闲模式和掉电模式的电源。

6. 逻辑符号

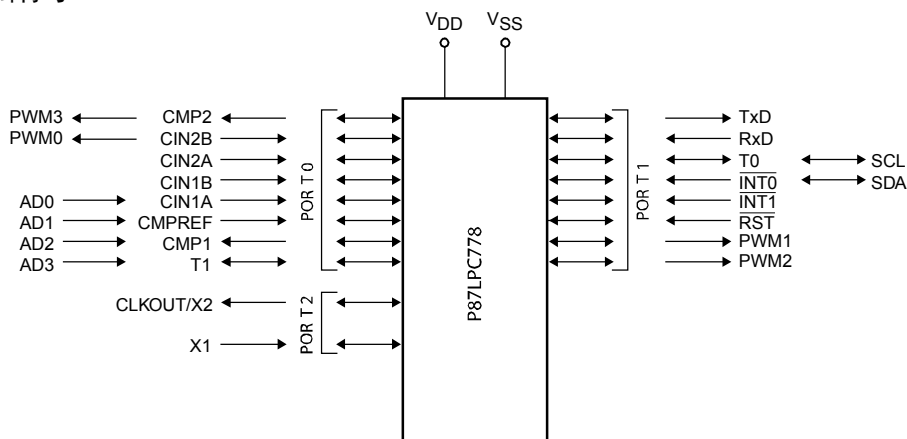


图 4 逻辑符号

7. 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：

- 用户不要试图访问任何未定义的 SFR 地址。
- 对任何已定义的 SFR 的访问必须符合 SFR 的功能。
- 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。

表 3 特殊功能寄存器

*表示可以位寻址

名称	描述	地址	位功能和地址								复位值 Hex
			MSB							LSB	
		位地址	E7	E6	E5	E4	E3	E2	E1	E0	
ACC*	累加器	E0h									00H
		位地址	C7	C6	C5	C4	C3	C2	C1	C0	
ADCON*	A/D 控制	C0h	ENADC			ADC1	ADCS	RCCLK	AADR1	AADR0	02H
AUXR1	辅助功能寄存器	A2h	KBF	BOD	BOI	LPEP	SRST	0	-	DPS	02h
		位地址	F7	F6	F5	F4	F3	F2	F1	F0	
B*	B 寄存器	F0h									00H
CMP1	比较器 1 控制	ACh	-	-	CE1	CP1	CN1	OE1	CO1	CMF1	00H
CMP2	比较器 2 控制	ADh	-	-	CE2	CP2	CN2	OE2	CO2	CMF2	00H
CNSW0	PWM 计数器映象 寄存器 0	D1h	CNSW7	CNSW6	CNSW5	CNSW4	CNSW3	CNSW2	CNSW1	CNSW0	FFH
CNSW1	PWM 计数器映象 寄存器 1	D2h	-	-	-	-	-	-	CNSW9	CNSW8	FFH
CPSW0	PWM 比较映象寄 寄存器 0	D3h	CPSW07	CPSW06	CPSW05	CPSW04	CPSW03	CPSW02	CPSW01	CPSW00	00H

续上表...

名称	描述	地址	位功能和地址								复位值
			MSB				LSB				Hex
CPSW1	PWM 比较映象寄存器 1	D4h	CPSW17	CPSW16	CPSW15	CPSW14	CPSW13	CPSW12	CPSW11	CPSW10	00H
CPSW2	PWM 比较映象寄存器 2	D5h	CPSW27	CPSW26	CPSW25	CPSW24	CPSW23	CPSW22	CPSW21	CPSW20	00H
CPSW3	PWM 比较映象寄存器 3	D6h	CPSW37	CPSW36	CPSW35	CPSW34	CPSW33	CPSW32	CPSW31	CPSW30	00H
CPSW4	PWM 比较映象寄存器 4	D7h	CPSW47	CPSW46	CPSW45	CPSW44	CPSW43	CPSW42	CPSW41	CPSW40	00H
DIVM	CPU 时钟分频	95h									00H
DPTR	数据指针 (2 字节)										
DPH	数据指针高字节	83h									00H
DPL	数据指针低字节	82h									00H
		位地址	CF	CE	CD	CC	CB	CA	C9	C8	
I2CFG*	I ² C 总线配置	C8h/RD	SLAVEN	MASTRQ	0	TIRUN	-	-	CT1	CT0	00H
		C8h WR	SLAVEN	MASTRQ	CLRTI	TIRUN	-	-	CT1	CT0	
		位地址	DF	DE	DD	DC	DB	DA	D9	D8	
I2CON*	I ² C 总线控制	D8h/RD	RDAT	ATN	DRDY	ARL	STR	STP	MASTER	-	80h
		D8h WR	CXA	IDLE	CDR	CARL	CSTR	CSTP	XSTR	XSTP	
I2DAT	I ² C 总线数据	D9h/RD	RDAT	0	0	0	0	0	0	0	80h
		D9h WR	XDAT	×	×	×	×	×	×	×	
		位地址	AF	AE	AD	AC	AB	AA	A9	A8	
IEN0*	中断使能 0	A8h	EA	EWD	EBO	ES	ET1	EX1	ET0	EX0	00H
		位地址	EF	EE	ED	EC	EB	EA	E9	E8	
IEN1*	中断使能 1	E8h	ETI	-	EC1	EAD	-	EC2	EKB	EI2	00H
		位地址	BF	BE	BD	BC	BB	BA	B9	B8	
IP0*	中断优先级 0	B8h	-	PWD	PBO	PS	PT1	PX1	PT0	PX0	00H
IP0H	中断优先级 0 高字节	B7h	-	PWDH	PBOH	PSH	PT1H	PX1H	PT0H	PX0H	00H
		位地址	FF	FE	FD	FC	FB	FA	F9	F8	
IP1*	中断优先级 1	F8h	PTI	-	PC1	PAD	-	PC2	PKB	PI2	00H
IP1H	中断优先级 1 高字节	F7h	PTIH	-	PC1H	PADH	-	PC2H	PKBH	PI2H	00H
		位地址									
KBI	键盘中断	86h									00H
		位地址	87	86	85	84	83	82	81	80	
P0*	P0 口	80h	T1	CMP1	CMPREF	CIN1A	CIN1B	CIN2A	CIN2B	CMP2	[1]
		位地址	97	96	95	94	93	92	91	90	
P1*	P1 口	90h	DAC0	DAC1	\overline{RST}	INT1	INT0	T0	RxD	TxD	[1]
		位地址	A7	A6	A5	A4	A3	A2	A1	A0	
P2*	P2 口	A0h	-	-	-	-	-	-	X1	X2	[1]
P0M1	P0 口输出模式 1	84h	P0M1.7	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0	00H

续上表...

名称	描述	地址	位功能和地址								复位值
			MSB				LSB				Hex
P0M2	P0 口输出模式 2	85h	P0M2.7	P0M2.6	P0M2.5	P0M2.4	P0M2.3	P0M2.2	P0M2.1	P0M2.0	00H
P1M1	P1 口输出模式 1	91h	P1M1.7	P1M1.6	-	P1M1.4	-	-	P1M1.1	P1M1.0	00H
P1M2	P1 口输出模式 2	92h	P1M2.7	P1M2.6	-	P1M2.4	-	-	P1M2.1	P1M2.0	00H
P2M1	P2 口输出模式 1	A4h	P2S	P1S	P0S	ENCLK	ENT1	ENT0	P2M1.1	P2M1.0	00H
P2M2	P2 口输出模式 2	A5h	-	-	-	-	-	-	P2M2.1	P2M2.0	00H
PCON	功率控制	87h	SMOD1	SMOD0	BOF	POF	GF1	GF0	PD	IDL	[2]
		位地址	D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	程序状态字	D0h	CY	AC	F0	RS1	RS0	OV	F1	P	00H
PT0AD	P0 口数字输入禁止	F6h									00H
		位地址	9F	9E	9D	9C	9B	9A	99	98	
SCON*	串口控制	98h	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SBUF	串口数据缓冲	99h									xxH
SADDR	串口地址	A9h									00H
SADEN	串口地址使能	B9h									00H
SP	堆栈指针	81h									07H
		位地址	8F	8E	8D	8C	8B	8A	89	88	
TCON*	定时器 0 和 1 控制	88h	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TH0	定时器 0 高字节	8Ch									00H
TH1	定时器 1 高字节	8Dh									00H
TL0	定时器 0 低字节	8Ah									00H
TL1	定时器 1 低字节	8Bh									00H
TMOD	定时器 0 和 1 方式寄存器	89h	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
WDCON	看门狗控制	A7h	-	-	WDOVF	WDRUN	WDCLK	WDS2	WDS1	WDS0	[3]
WDRST	看门狗复位	A6h									xxH

[1] I/O 口复位后的值由配置字节 UCFG1 的位 PRHI 决定。

[2] PCON 的复位值为 xxBOF POF-0000b。复位不会影响 BOF 和 POF 标志。POF 标志 上电时由硬件置位。BOF 标志在掉电复位/中断和上电时置位。

[3] 看门狗复位时 WDCON 的复位值为 xx01 0000b。如果看门狗使能，其它复位时 WDCON 的复位值为 xx01 0000b；如果看门狗禁能，其它复位时 WDCON 的复位值为 xx00 0000b。

8. 功能描述

注：详细功能描述请参考 P87LPC778 使用指南。

8.1 增强型 CPU

P87LPC778 采用增强型 80C51 CPU，其运行速度是标准 80C51 器件的 2 倍。这就意味着 P87LPC778 运行在 5MHz 时的性能和标准 80C51 器件运行在 10MHz 时的性能相同。每个机器周期由 6 个振荡周期组成，大多数指令的执行时间为 6 或 12 个时钟。用户也可选择恢复标准 80C51 的时序，这样，每个机器就包含 12 个振荡周期。

在后面各章节中，‘CPU 时钟’都是指控制内部指令执行的时钟。有别于器件采用标准 80C51 时序（通过 CLKR 配置位）或时钟被分频（通过 DIVM 寄存器）时的外部时钟。详见第 8.10 节“振荡器”。

8.2 模拟功能

P87LPC778 内部集成了 1 个 A/D 转换器和 2 个模拟比较器。为了得到最佳模拟性能和降低功耗，用作模拟功能的管脚的数字输出和数字输入功能必须被禁止。

数字输出功能可通过将管脚输出设置成仅为输入模式（高阻）来禁止，见第 8.9 节“I/O 口”。

P0 口的数字输入通过 PT0AD 寄存器来禁止。PT0AD 寄存器的每一位对应着 P0 口的一个管脚，置位相应的位就可禁止管脚的数字输入。口线的数字输入功能被禁止后，任何指令读出端口都得到 0。

8.3 A/D 转换器

P87LPC778 集成了一个 4 路、8 位的 A/D 转换器。P0 口的 4 个管脚复用为 A/D 输入。由于器件的 I/O 口数量有限，A/D 转换器将处理器的电源脚 V_{DD} 和 V_{SS} 用作电源脚和参考电压脚。A/D 转换器的工作电压也降低到 3.0V (V_{DD})。

A/D 转换器由 4 路模拟开关和 8 位逐次逼近式 ADC 组成。A/D 转换器通过使用一个比率电位器来保证 DAC 的单调性。

特殊功能寄存器 ADCON 用来控制 A/D 转换器，详见表 4 和 5。A/D 转换器先通过置位 ENADC 位使能，稳定运行 10us 后再开始转换。开始 A/D 转换前，通过位 AADR1 和 AADR0 来选择一个模拟输入脚。在 A/D 转换过程中不能改变这两位值。

A/D 转换通过置位 ADCS 来启动，该位的置位状态在转换过程中一直保持不变。转换结束后，ADCS 位清零，ADCI 位置位。当位 ADCI 置位时，如果系统中断使能，A/D 中断使能（通过 IE1 寄存器的 EAD 位来实现）且 A/D 中断的优先级最高，则产生中断。

转换结束后，结果保存到寄存器 DAC0 中，这也是 ADC 的结果。该结果将一直保持不变直至启动下次转换。下次转换启动前，ADCI 位必须通过软件清零。A/D 通道选择和置位 ADCS 以启动下一次 A/D 转换可使用同一条指令，但不能与清除 ADCI 共用一条指令。

A/D 转换的连接图见表 5。

理想的 A/D 转换结果可通过下式计算得出：

$$\text{Result} = (V_{IN} - V_{SS}) \times \frac{255}{V_{DD} - V_{SS}} \quad (\text{结果四舍五入为整数}) \quad (1)$$

表 4 ADCON—A/D 控制寄存器位分配（地址：C0h）

可位寻址；复位值：00H

位	7	6	5	4	3	2	1	0
符号	ENADC	—	—	ADCI	ADCS	RCCLK	AAADR1	AAADR0

表 5 ADCON—A/D 控制寄存器位描述（地址：C0h）

位	符号	描述
7	ENADC	当 ENADC=1 时，A/D 转换器使能，10us 后启动一次转换。当 ADCS 或 ADCI 为‘1’时不能清除 ENADC 位。
6,5	-	保留为将来之用。用户程序不要将其置 1。
4	ADCI	A/D 转换结束/中断标志。该标志在 A/D 转换结束后置位。如果中断使能且优先级足够高，该位置位时将产生中断。该位必须通过软件清零。

续上表...

位	符号	描述
3	ADCS	A/D 转换启动。通过软件置位该位来启动所选 A/D 输入的转换。ADCS 在 A/D 转换过程中一直保持置位状态，转换结束后自动清零。当 ADCS 或 ADCI 为 1，不启动新的转换。见表 6。
2	RCCLK	当 RCCLK=0 时，CPU 时钟用作 A/D 时钟。当 RCCLK=1 时，内部 RC 振荡器用作 A/D 时钟。该位在 ADCS 和 ADCI 为 0 时可写。
1,0	AADR1,0	AADR1 和 AADR0 用来选择 A/D 转换通道。这两位只能在 ADCS 和 ADCI 为 0 时写入。见表 7。

表 6 ADCON—ADCI,ADCS A/D 状态

ADCI,ADCS	A/D 状态
00	A/D 空闲，可以启动转换
01	A/D 忙，不能启动转换
10	A/D 转换结束。在启动一次新转换前，ADCI 必须清零。
11	A/D 转换结束。在启动一次新转换前，ADCI 必须清零。该状态只出现在 A/D 转换结束后的一个机器周期内。

表 7 ADCON—AADR1,AADR0 A/D 输入选择

AADR1,AADR0	A/D 输入选择
00	AD0 (P0.3)
01	AD1 (P0.4)
10	AD2 (P0.5)
11	AD3 (P0.6)

8.4 A/D 时序

A/D 可选择两种时钟方式。默认的 A/D 时钟源是 CPU 时钟。当 CPU 时钟用作 A/D 时钟源时，完成一次 A/D 转换需要 31 个机器周期。A/D 可能工作在 20MHz 的最大 CPU 时钟频率下，从而得到 9.3us 的转换时间。此时，可通过下式来计算 A/D 转换时间： $186\mu\text{s}/\text{CPU 时钟频率 (MHz)}$ 。为了保证 A/D 转换的精度，CPU 时钟频率不得低于 1MHz。

片内 RC 振荡器也可用作 A/D 时钟源，即使 RC 振荡器不用作 CPU 时钟。这可通过置位 ADCON 的 RCCLK 位来实现。片内 RC 振荡器用作 A/D 时钟源有几个优点：第一，在较低的 CPU 时钟频率下可加快 A/D 转换速度。第二，CPU 可运行在 1MHz 以下，但不影响 A/D 转换精度。第三，可使用掉电模式来完全关闭 CPU、振荡器以及其它外围功能，从而获得最大可能的 A/D 转换精度。

当 A/D 以 RC 振荡器为时钟源而 CPU 使用其它时钟源时，需要 3 或 4 个机器周期来同步 A/D 操作。所需的时间从最小 3 个机器周期（以 CPU 时钟频率计算）+108 个 RC 时钟~最大 4 个机器周期（以 CPU 时钟频率计算）+112 个 RC 时钟。

不同的 CPU 时钟频率所对应的 A/D 转换时间如表 8 所示。表 8 中列出了（RCCLK=1）使用 RC 时钟频率 6MHz 时的最大时间，标称时间取理想的 6MHz RC 时钟频率和平均 3.5 个机器周期（以 CPU 时钟频率计算）。

表 8 A/D 转换时间

CPU 时钟频率	RCCLK=0	RCCLK=1		
		最小	标称	最大
32kHz	NA	563.4us	659us	757us
1MHz	186us	32.4us	39.3us	48.9us
4MHz	46.5us	18.9us	23.6us	30.1us
11.0592MHz	16.8us	16us	20.2us	27.1us
12MHz	15.5us	15.9us	20.1us	26.9us
16MHz	11.6us	15.5us	19.7us	26.4us
20MHz	9.3us	15.3us	19.4us	26.1us

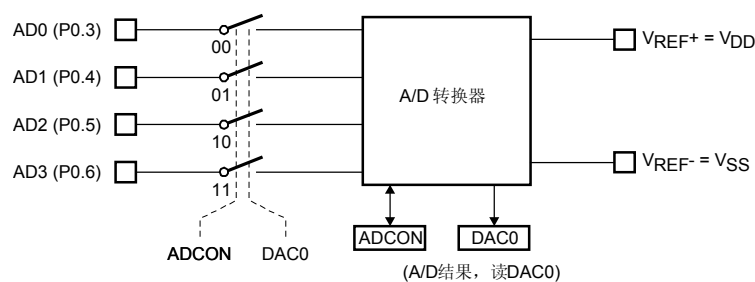


图 5 A/D 转换器连接图

8.4.1 A/D 工作在掉电和空闲模式

当使用 CPU 时钟作为 A/D 时钟源时，空闲模式可节省功耗和/或减少转换过程中的噪声。如果 A/D 中断有效，转换结束时 CPU 操作恢复且空闲模式自动终止。空闲模式下，CPU 本身的噪声被消除，但是振荡器和正在运行的其它片内元件产生的噪声仍然存在。

当 A/D 使用片内 RC 振荡器作为时钟源时 (RCCLK=1)，CPU 进入掉电模式。掉电模式下，可消除大部分片内噪声源从而获得最大的 A/D 转换精度。

当使用 CPU 时钟作为 A/D 时钟源时 (RCCLK=0)，进入掉电模式后 A/D 转换终止但不唤醒 CPU。当 A/D 转换恢复时，DAC0 的内容无效。

当 A/D 转换启动时，为了获得最大精度的转换结果，必须在 2 个机器周期内激活掉电或空闲模式。这 2 个机器周期是以 CPU 时钟频率计算的。当在掉电或空闲模式下使用 A/D 时，注意在 A/D 转换完成之前，CPU 不应被其它中断唤醒。唤醒掉电模式和空闲模式的原因是不同的。

A/D 转换精度还受到应用中其它处产生的噪声、电源噪声和电源调节噪声的影响。由于 P87LPC778 的电源端还用作 A/D 的参考电压和电源，因此电源对 A/D 读取精度有着非常直接的影响。在非掉电模式下使用 A/D 转换时，通过 CLKR 或 DIVM 将时钟分频会对 A/D 转换精度有不利的影响。

8.4.2 A/D 转换例程

第一部分给出了使用 A/D 时对 I/O 口进行配置。程序中将用于 A/D 转换的四个口都设置成可用的 A/D 通道。用作模拟功能的 I/O 口配置见 8.2 节“模拟功能”。

；对用于 A/D 转换的 I/O 口进行配置，不影响其它管脚。

```
mov PT0AD,#78h ; 禁止 A/D 输入管脚的数字输入功能
```

```
anl P0M2,#87h ; 禁止 A/D 输入管脚的数字输出功能
```

```
orl P0M1,#78h ; 禁止 A/D 输入管脚的数字输出功能
```

下面程序使用了 A/D 中断。程序 ADStart 使用累加器中提供的 A/D 通道开始 A/D 转换。未检查通道编号的有效性。A/D 必须提前使能以使其有足够的时间稳定下来。

中断处理程序读取转换结果并将其存入存储器地址 ADResult。启动转换前必须使能中断。

; 启动 A/D 转换。

ADStart:

```
orl   ADCON,A       ; 添加新的通道编号
setb  ADCS          ; 启动 A/D 转换
; orl   PCON,#01h   ; CPU 在此进入空闲模式
; orl   PCON,#02h   ; 如果 RCCLK=1, CPU 在此进入掉电模式
ret
```

; A/D 中断处理程序。

ADInt:

```
push  ACC           ; 保存累加器
mov   A,DAC0        ; 读 DAC0 寄存器, 得到 A/D 结果
mov   ADResult,A    ; 保存结果
clr   ADCI          ; 清 A/D 结束标志
anl   ADCON,#0fch   ; 清 A/D 通道编号
pop   ACC           ; 恢复累加器
reti
```

下面是查询方式的 A/D 转换。启动累加器提供的 A/D 通道的转换。未检查通道编号的有效性。A/D 必须提前使能以使其有足够的时间稳定下来。转换结果存入累加器。

ADRead:

```
orl   ADCON,A       ; 添加新的通道编号
setb  ADCS          ; 启动 A/D 转换
```

ADChk:

```
jnb   ADCI,ADChk    ; 等待 ADCI 置位
mov   A,DAC0        ; 获取 A/D 转换结果
clr   ADCI          ; 清 A/D 结束标志
anl   ADCON,#0fch   ; 清 A/D 通道编号
ret
```

8.5 模拟比较器

P87LPC778 集成了 2 个模拟比较器。输入和输出选项允许比较器配置成不同模式。当正向输入（两个可选管脚之一）大于反向输入（选择一个外部管脚或内部参考电压）时，输出为 ‘1’（可从寄存器读出和/或输出到管脚）。反之则输出为 ‘0’。输出变化时每个比较器可配置为产生中断。

8.5.1 比较器配置

每个比较器对应一个控制寄存器，比较器 1 对应 CMP1，比较器 2 对应 CMP2。两个控制寄存器完全相同，见表 9 和 10。

两个比较器的连接见图 6。每个比较器有 8 种可能的配置，由各自对应的 CMPn 寄存器的控制位 CPn，CNn 和 OEn 来决定，见图 7。比较器工作电压可低至 3.0V。

当比较器首次使能时，比较器输出和中断标志 10us 后有效。这段时间内比较器中断禁能，中断标志在中断使能前必须清除以防止立即响应中断。

表 9 CMPn—比较器控制寄存器 CMP1 和 CMP2 位分配 (CMP1: ACh, CMP2: ADh)
不可位寻址; 复位值: 00H

位	7	6	5	4	3	2	1	0
符号	-	-	CEn	CPn	CNn	OEn	COn	CMFn

表 10 CMPn—比较器控制寄存器 CMP1 和 CMP2 位描述 (CMP1: ACh, CMP2: ADh)

位	符号	描述
7,6	-	保留为将来之用。用户不要将其置 1。
5	CEn	比较器使能。该位通过软件置位时, 相应的比较器使能。CEn 首次置位 10us 后, 比较器输出有效。
4	CPn	比较器正向输入选择。该位为 '0' 时, CINnA 用作比较器正向输入。该位为 '1' 时, CINnB 用作比较器正向输入。
3	CNn	比较器反向输入选择。该位为 '0' 时, 比较器参考管脚 CMPREF 用作反向比较器输入。该位为 '1' 时, 内部比较器参考电压 Vref 用作反向输入。
2	OEn	输出使能。该位为 '1' 时, 如果比较器使能 (CEn=1), 比较器输出连接到 CMPn。该输出与 CPU 时钟不同步。
1	COn	比较器输出, 和 CPU 时钟同步, 允许通过软件读出, 比较器禁能 (CEn=0) 时被清除。
0	CMFn	比较器中断标志。当比较器输出 COn 变化时, 该位由硬件置位。如果中断使能并且优先级足够高, 该位置位可产生中断。比较器禁能 (CEn=0) 时通过软件清零。

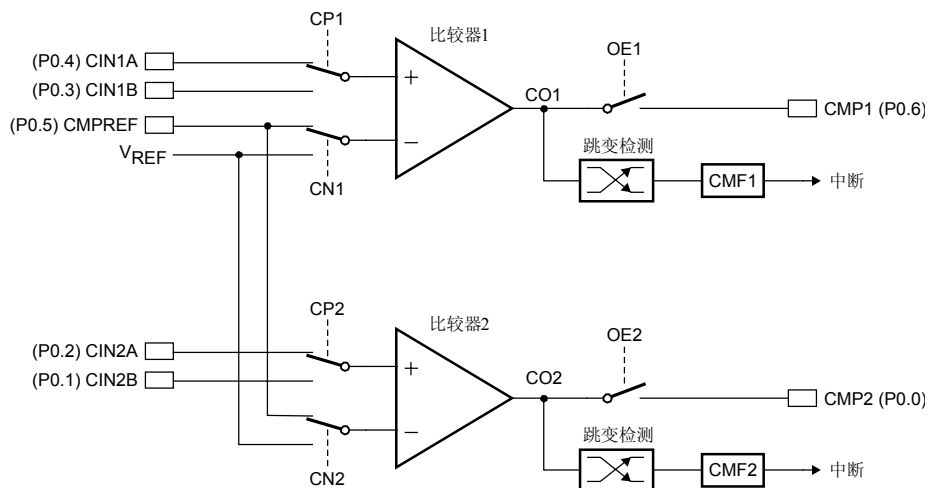


图 6 比较器输入和输出连接

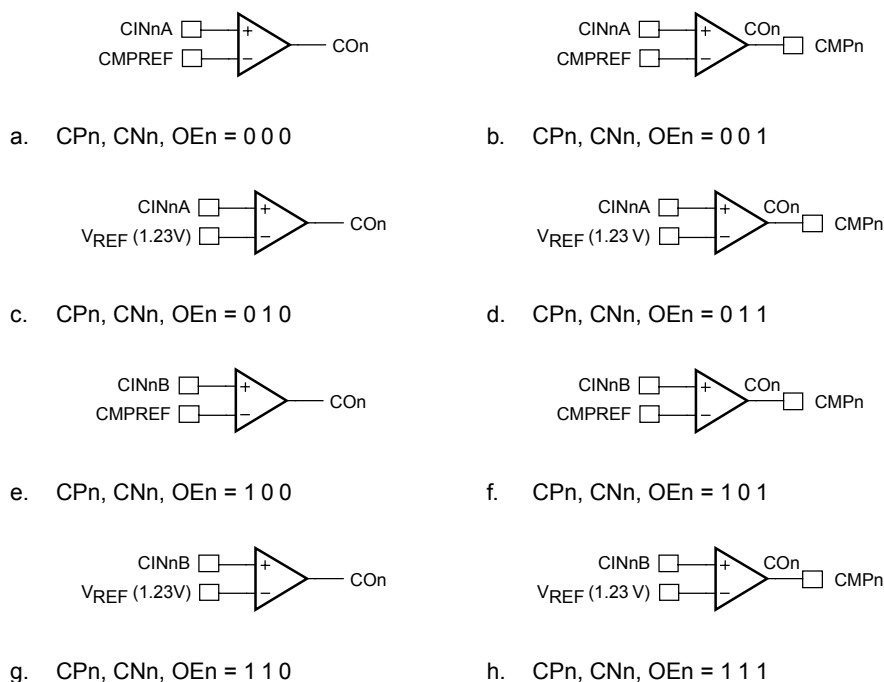


图 7 比较器配置

8.5.2 内部参考电压

当只使用一个比较器输入管脚时，由内部参考电压发生器来提供默认参考电压，其值为 $V_{ref}=1.28V \pm 10\%$ 。

8.5.3 比较器中断

每个比较器都对应一个中断标志 $CMFn$ （配置寄存器中的一位）。比较器输出状态变化时 $CMFn$ 标志置位。该标志可通过软件查询或产生一个中断。当 $IEN1$ 寄存器的 ECn 使能位置位和中断系统通过 $IEN0$ 寄存器的 EA 位被使能时，产生中断。

8.5.4 比较器和节电模式

在掉电模式或空闲模式下，两个比较器或其中之一均可继续保持使能状态。在节电模式下，比较器将继续工作。当比较器中断使能时，比较器输出发生改变时将会产生一个中断并将处理器唤醒。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，打开准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，必须将比较器的功耗考虑在内。

8.5.5 比较器配置举例

下面是一段比较器 1 的初始化程序。比较器 1 使用 $CIN1A$ 和 $CMPREF$ 输入，比较器输出到 $CMP1$ 脚，当比较器输出改变时产生中断。

```
CmpInit: b
mov    PT0AD,#30h    ; 禁止 CIN1A,CMPREF 管脚的数字输入功能，使能比较器功能
anl    P0M2,#0CFh    ; 禁止 CIN1A,CMPREF 管脚的数字输出功能，使能比较器功能
orl    P0M1,#30h     ;
```

```

mov    CMP1,#024h      ; 开启比较器 1, 进行如下设置:
                        ; -CIN1A 为正向输入
                        ; -CMPREF 为反向输入
                        ; -比较结果输出到 CMP1 脚

call   delay10us       ; 比较器起动 10us 后方可使用
anl    CMP1,#0FEh      ; 清比较器 1 中断标志
setb   EC1              ; 使能比较器 1 中断,
                        ; 保持当前优先级

setb   EA               ; 使能中断系统 (如果有必要)
ret                                          ; 返回调用处
    
```

在中断返回前必须清除中断标志 (此处为 CMF1)。

8.6 脉宽调制器

P87LPC778 含有 4 个脉宽调制 (PWM) 通道, 可产生长度和间隔可编程的脉冲。

PWM0 从 P0.1 输出, PWM1 从 P1.6 输出, PWM2 从 P1.7 输出, PWM3 从 P0.0 输出。

器件复位后, 每个 PWM 通道的输出由 UCFG1.5 (PRHI) 位的状态来反映。PRHI 为 0 时 PWM 输出低电平, PRHI 为 1 时 PWM 输出高电平。

当 PRHI 为 0 时, 必须先向用作 PWM 输出的 I/O 口写入 1, 这样它们才能反映内部 PWM 的输出。

PWM 的方框图见图 8。

PWM 连续输出之间的间隔由 10 位的递减计数器来控制, 该计数器对内部微控制器时钟进行计数。

当 UCFG1 寄存器的位 3 为 ‘1’ (6 时钟模式) 时, 微处理器时钟 (即 PWM 计数器时钟) 和时钟源拥有相同的频率。

$$f_{CPWM} = f_{OSC} \quad (2)$$

当 UCFG1 寄存器的位 3 为 ‘0’ (12 时钟模式) 时, 微处理器时钟和 PWM 计数器时钟的频率为时钟源的一半:

$$f_{CPWM} = \frac{f_{osc}}{2} \quad (3)$$

当计数器溢出时, 重新装入用户定义的值。这种结构允许用户将 PWM 频率设置成微控制器时钟频率的任意整数分之一。PWM 的重复频率由下式得出:

$$f_{PWM} = \frac{f_{CPWM}}{(CNSW + 1)} \quad (4)$$

此处, CNSW 是寄存器 CNSW0 和 CNSW1 组合而成的一个 10 位数的值, 见下表。

‘映象’ 一词并非是指写入控制计数器寄存器的实际值, 而是写入到保存寄存器中的值。

在用户程序中, 置位 XFER 位, 等到计数器下次溢出时将下面保存寄存器的值装入包含实际重装值的寄存器中。见下面的描述。

表 11 CNSW0—计数器映象寄存器 0 位分配 (地址: 0D1H)

位	7	6	5	4	3	2	1	0
符号	CNSW7	CNSW6	CNSW5	CNSW4	CNSW3	CNSW2	CNSW1	CNSW0

表 12 CNSW1—计数器映象寄存器 1 位分配 (地址: 0D2H)

位	7	6	5	4	3	2	1	0
符号	未用	未用	未用	未用	未用	未用	CNSW9	CNSW8

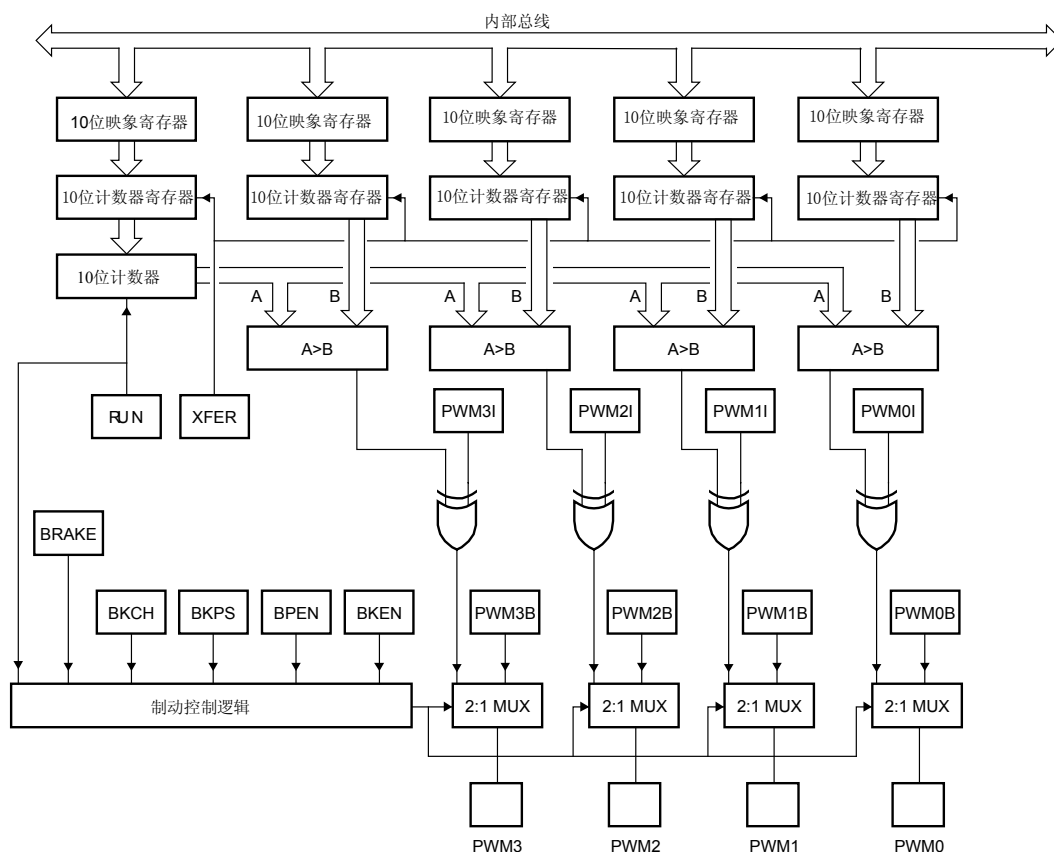


图 8 PWM 方框图

每个 PWM 输出脉冲的宽度由对应比较映像寄存器 CPSW0~CPSW4、CPSW0-3 (位 0-7) 和 CPSW4 (位 8-9) 的值决定。计数器溢出时 PWM 输出强制为高电平。此状态将保持到下次溢出前, 然后变低。PWMn 输出高电平期间的微控制器时钟脉冲数为:

$$t_{HI} = (CNSW - CPSW_n + 1)$$

如果比较值大于计数器重装值, PWM 输出总是为高电平。另外还存在两种特殊情况。第一, 比较值为全零, 000, PWM 输出总是为高电平。第二, 比较值为全 1, 3FF, PWM 输出总是为电平。此时, 比较值仍被装入映像寄存器, 在用户程序控制下将保存寄存器的值写入实际的比较寄存器。在下面描述的寄存器中, ‘CPSW’ 后的数字用来指示 PWM 输出, 即 CPSW0 控制 PWM0 的宽度、CPSW1 控制 PWM1 的宽度, 等等。‘CPSW’ 后面的第二个数字, 如 CPSW0.0, 用来指示比较值的位, 即 CPSW0.0 表示值装入 PWM0 比较寄存器的位 0。

表 13 CPSW0—比较映像寄存器 0 位分配 (地址: 0D3H)

可位寻址; 复位值: 00H

位	7	6	5	4	3	2	1	0
符号	CPSW07	CPSW06	CPSW05	CPSW04	CPSW03	CPSW02	CPSW01	CPSW00

表 14 CPSW1—比较映像寄存器 1 位分配 (地址: 0D4H)

可位寻址; 复位值: 00H

位	7	6	5	4	3	2	1	0
符号	CPSW17	CPSW16	CPSW15	CPSW14	CPSW13	CPSW12	CPSW11	CPSW10

表 15 CPSW2—比较映象寄存器 0 位分配 (地址: 0D5H)

可位寻址; 复位值: 00H

位	7	6	5	4	3	2	1	0
符号	CPSW27	CPSW26	CPSW25	CPSW24	CPSW23	CPSW22	CPSW21	CPSW20

表 16 CPSW3—比较映象寄存器 0 位分配 (地址: 0D6H)

可位寻址; 复位值: 00H

位	7	6	5	4	3	2	1	0
符号	CPSW37	CPSW36	CPSW35	CPSW34	CPSW33	CPSW32	CPSW31	CPSW30

表 17 CPSW4—比较映象寄存器 0 位分配 (地址: 0D7H)

可位寻址; 复位值: 00H

位	7	6	5	4	3	2	1	0
符号	CPSW39	CPSW38	CPSW29	CPSW28	CPSW19	CPSW18	CPSW09	CPSW08

PWM 模块的所有功能都在 PWMCON0 寄存器的控制下完成。大多数控制位的操作都非常简单。

例如, 每个输出都对应一个翻转位, 使 PWM 的输出与它的未翻转输出值相反。

由传送位 (PWMCON0.6) 来控制将数据从映象寄存器传送到控制寄存器。传送位置位后, 数据在计数器首次溢出时开始传送。

用户也可通过传送位 (PWMCON.6) 对溢出导致的数据传送进行监测。当数据开始传送时, PWM 逻辑将自动复位传送位。

运行位 PWMCON0.7 允许 PWM 的计数器处于运行或空闲状态。

从映象寄存器到工作寄存器的数据传送仅发生在计数器产生下溢时。这种状况需要用户程序来观察后面的预防措施。

如果 XFER 置位而 RUN 未使能, 对 PWMCON1 进行写操作将永不会发生传送。

如果接着将 RUN 置位而 XFER 未置位, 将得不到更新的比较值和计数器值。

如果 XFER 和 RUN 都置位, 并且在下溢之前写 PWMCON0 使 RUN 置位而 XFER 未置位, 则永不会发生传送。比较和计数器值将再次使用试图更新之前的值。

如上文所述, 通过查询传送位可以确定何时发生传送。除非有强制的原因使其执行别的方式, 否则我们建议用户在写 PWMCON0 时将运行位 (PWMCON0.7) 和传送位 (PWMCON0.6) 置位。当运行位 (PWMCON0.7) 清零时, PWM 输出呈现运行位清零前的状态。通常该状态是不可知的。

为了使 PWM 输出在运行位清除时处于已知状态, 比较寄存器可写入 ‘始终为 1’ 或 ‘始终为 0’, 这样, 当计数器停止时, 输出就可得到期望的状态。在这之后, 置位 PWMCON0 的传送位和运行位, 然后通过查询 PWMCON0 来确认传送何时发生。一旦传送开始, 运行位被清零。输出将保持到运行位清零前的状态。

表 18 PWMCON0—PWM 控制寄存器 0 位分配 (地址: 0DAH)

复位值: 00H

位	7	6	5	4	3	2	1	0
符号	RUN	XFER	PWM3I	PWM2I	-	PWM1I	PWM0I	-

表 19 PWMCON0—PWM 控制寄存器 0 位描述（地址：0DAH）

位	符号	值	描述
7	RUN	0	计数器停止并装入预置值。如果制动使能，PWMx 输出等于对应 PWMxB 位（PWMCON1[3:0]）的值。如果制动未使能，PWMx 输出等于比较后的值。
		1	计数器运行。
6	XFER	0	计数器运行。
		1	计数器和比较器未连接到有效的寄存器。
5	PWM3I	0	PWM3 输出不翻转。从周期开始到比较之前 PWM3 输出 ‘1’，之后输出 ‘0’。
		1	PWM3 输出翻转。从周期开始到比较之前 PWM3 输出 ‘0’，之后输出 ‘1’。
4	PWM2I	0	PWM2 输出不翻转。从周期开始到比较之前 PWM2 输出 ‘1’，之后输出 ‘0’。
		1	PWM2 输出翻转。从周期开始到比较之前 PWM2 输出 ‘0’，之后输出 ‘1’。
2	PWM1I	0	PWM1 输出不翻转。从周期开始到比较之前 PWM1 输出 ‘1’，之后输出 ‘0’。
		1	PWM1 输出翻转。从周期开始到比较之前 PWM1 输出 ‘0’，之后输出 ‘1’。
1	PWM0I	0	PWM0 输出不翻转。从周期开始到比较之前 PWM0 输出 ‘1’，之后输出 ‘0’。
		1	PWM0 输出翻转。从周期开始到比较之前 PWM0 输出 ‘0’，之后输出 ‘1’。

8.6.1 PWM 制动功能

通常当 PWM 制动使能时，4 个 PWM 输出将强制进入用户设定的状态，即 PWMCON1 位 0~3 选择的状态。

正如 PWMCON1 寄存器所描述的，如果 PWMCON1.4 为 1，制动使能由 PWMCON1.7 (BKCH) 和 PWMCON1.5 (BPEN) 控制。

BKCH 和 BKEN 为 0 时制动使能。当 PWMCON1.7 为 1 且 RUN 位 (PWMCON0.7) 为 0 时，制动使能。当 PWMCON1.6 为 1 且制动脚 (P0.2) 的极性与之相同时，制动使能。制动使能时 RUN 位 (PWMCON0.7) 自动清零。不允许 PWMCON1.7 和 PWMCON1.5 同时为 1。

由于制动脚使能将自动清零 RUN 位，因此用户程序可通过查询该位来判定制动脚何时触发制动。

制动信号释放后，PWM 脚继续保持 PWMCON1[3:0] 的制动状态。因为 RUN 位清零后，PWM 脚将保持制动状态直到 PWM 重新启动。

PWMCON1 寄存器详见表 20，21 和 22。

表 20 PWMCON1—PWM 控制寄存器 1 位分配（地址：C8H）

复位值：00H

位	7	6	5	4	3	2	1	0
符号	BKCH	BKPS	BPEN	BKEN	PWM3B	PWM2B	PWM1B	PWM0B

表 21 PWMCON1—PWM 控制寄存器 1 位描述（地址：C8H）

位	符号	值	描述
7	BKCH	-	见下表 22。
6	BKPS	0	如果 P0.2（制动脚）为低电平时，‘制动’有效。
		1	如果 P0.2（制动脚）为高电平时，‘制动’有效。
5	BPEN		见下表 22。
4	BKEN	0	‘制动’禁能。
		1	‘制动’使能（见下表 22）。
3	PWM3B	0	当制动使能时，PWM3 为低电平。
		1	当制动使能时，PWM3 为高电平。

续上表...

位	符号	值	描述
2	PWM2B	0	当制动使能时, PWM2 为低电平。
		1	当制动使能时, PWM2 为高电平。
1	PWM1B	0	当制动使能时, PWM1 为低电平。
		1	当制动使能时, PWM1 为高电平。
0	PWM0B	0	当制动使能时, PWM0 为低电平。
		1	当制动使能时, PWM0 为高电平。

表 22 PWMCON1 制动条件

BPEN	BKCH	制动条件
0	0	设置软件制动
0	1	PWM 不运行时打开 (制动脚无效)
1	0	制动使能时打开 (PWM 运行无效)
1	1	清除软件制动

8.7 I²C 总线接口

I²C 总线通过两条线 (SDA 和 SCL) 来实现连接到总线上的器件之间的数据传输。总线有以下特性:

- 主机和从机之间的双向数据传输。
- 从机串行寻址 (无需另外接线)。
- 每传输完一个字节产生应答。
- 多主总线。
- 多主机同时传输时进行仲裁避免总线数据冲突。

I²C 总线子系统的硬件简化了 I²C 总线的驱动软件。除了必须的仲裁、帧错误检查、时钟扩展和总线超时定时器外, I²C 总线还包含一个一位的接口。这个接口通过循环查询或中断来与软件同步。详情请参考应用笔记 AN422 第 4 章 ‘8XC751 微控制器用作 I²C 总线主控制器’ 中有关 87C77X 的 I²C 总线接口和驱动程序描述。

在 I²C 总线操作中有 6 个重要的时间段, 由定时器 I 设定:

- 器件用作主机时 SCL 的最小高电平时间。
- 器件用作主机时 SCL 的最小低电平时间。这个时间段对于象 P87LPC778 这样只包含一位硬件接口的器件并不重要, 因为 SCL 低电平时间会一直延长到软件响应 I²C 总线标志。软件响应时间通常等于或大于最小低电平时间。只要软件响应时间小于 (最小高电平时间+最小低电平时间), 定时器 I 就能够满足最小时间的要求。
- 停止条件中 SCL 高到 SDA 高的最小时间。
- I²C 总线停止条件和起始条件之间 SDA 高到 SDA 低的最小时间 (4.7ms, 见 I²C 总线规范)。
- 起始条件中 SDA 低到 SCL 低的最小时间。
- I²C 传输数据帧过程中 SCL 跳变的最大时间。数据帧在起始条件和停止条件之间进行传输。这段时间用来检测软件响应的产生和外部 I²C 总线事件的出现。SCL ‘强制为低’ 用来指示错误的主机或从机; SCL ‘强制为高’ 用来指示错误的设备或 I²C 总线上的噪声导致所有主机退出 I²C 总线仲裁的情况。

前 5 个时间段为 4.7ms (见 I²C 总线规范), 由定时器 I 低 3 位确定。定时器 I 对 87LPC778 CPU 时钟进行计数。定时器 I 有 4 个预置值, 可以优化不同振荡频率的时序。低频时, 软件响应时间增加, I²C 总线性能降低。见特殊功能寄存器 I2CFG 预分频值 (CT0, CT1) 的描述。

SCL 跳变的最大时间很重要, 但是其实际值却并不严格。10 位定时器 I 用来计数最大时间, 当 I²C 使能

时, SCL 脚信号变化时将定时器 I 清零。在 I²C 两个数据帧传输之间 (即下一个起始条件之前出现的复位或停止条件) 定时器不运行。当计数器 I 运行时, SCL 脚信号变化 1020 到 1023 个机器周期后定时器溢出。定时器溢出使 I²C 接口产生硬件复位。如果由于软件响应时间不够而造成总线挂起, P87LPC778 复位, 释放 SCL, 但仍允许总线其它器件继续工作。

8.7.1 I²C 总线中断

如果 I²C 总线中断使能 (EA 和 EI2C 都为 1), 当由于起始条件、终止条件、仲裁丢失或数据准备就绪等 (参考下面 ATN 的描述) 而使 ATN 标志置位时, 产生 I²C 总线中断。但实际上, 这样做降低了 I²C 总线接口的工作效率, 因为 I²C 总线中断服务程序必须从上百个条件中分辨出满足置位 ATN 标志的条件。而且, 由于 I²C 总线的运行速度很高, 为了等待 I²C 事件, 软件必须提高代码执行速度。

通常, I²C 总线中断只用来指示空闲从器件的起始条件或空闲主器件的停止条件 (如果它等待使用 I²C 总线)。这可通过使能上述条件的 I²C 中断来实现。

表 23 I2CON—I²C 总线控制寄存器位分配 (地址: D8H)

位寻址^[1]; 复位值: 81H

位	7	6	5	4	3	2	1	0
符号 (R)	RDAT	ATN	DRDY	ARL	STR	STP	MASTER	-
符号 (W)	CXA	IDLE	CDR	CARL	CSTR	CSTP	XSTR	XSTP

[1] 尽管 80C51 系列提供了位寻址功能, 但不能使用 SETB、CLR、CPL、MOV (bit) 或 JBC 指令来改变 I2CON 寄存器的值。这是因为对于这个寄存器的读和写操作意义不同。但可通过 JB 和 JNB 对 I2CON 进行位测试。

表 24 I2CON—I²C 总线控制寄存器位描述 (地址: D8H)

位	符号	访问	描述
7	RDAT	R	最后接收到的数据位。
	CXA	W	清除发送激活标志。
6	ATN	R	如果 DRDY、ARL、STR 或 STP 中的任何一个标志为 1, ATN=1。
	IDLE	W	I ² C 总线从器件模式中, 向该位写入 '1' 可使 I ² C 总线硬件将总线忽略, 直至总线被再次需要。
5	DRDY	R	数据就绪标志, 在 SCL 上升沿时置位。
	CDR	W	向该位写入 '1' 来清除 DRDY 标志。
4	ARL	R	仲裁丢失标志, 发送模式下仲裁失败时置位。
	CARL	W	向该位写入 '1' 来清除 ARL 标志。
3	STR	R	起始标志。主机或非空闲从机检测到起始条件时置位。
	CSTR	W	向该位写入 '1' 来清除 STR 标志。
2	STP	R	停止标志。主机或非空闲从机检测到停止条件时置位。
	CSTP	W	向该位写入 '1' 来清除 STP 标志。
1	MASTER	R	指示器件当前是否用作总线主机。
	XSTR	W	向该位写入 '1' 来产生一个重复起始条件。
0	-	R	未定义。
	XSTP	W	向该位写入 '1' 来产生一个停止条件。

8.7.2 读 I2CON

RDAT—SDA 的数据在 SCL 的上升沿捕获到 RDAT。I2DAT 也包含一个 RDAT 位 (低 7 位为 0)。从 RDAT 读取数据和从 I2DAT 中读取数据的不同是读 I2DAT 会将 DRDY 位清零, 使 I²C 总线可以继续处理下

一位数据。通常，接收字节的前 7 位从 I2DAT 中读出，第 8 位从 I2CON 寄存器的 RDAT 中读出。然后，写 I2DAT 来发送应答位和清除 DRDY。

ATN—当 DRDY、ARL、STR 和 STP 中的一个或多个为 ‘1’ 时 ATN 为 1。因此，通过测试 ATN 位可确定是否从等待状态中退出 I²C 总线服务程序。

DRDY—除空闲从机外，DRDY（和 ATN）在 SCL 的上升沿置位。DRDY 可通过置位 CDR 或读、写 I2DAT 寄存器来清零。SCL 的下一个低电平时间将延伸到程序响应（清除 DRDY）为止。

8.7.3 检测 ATN 和 DRDY

当程序检测到 ATN=1 时，将继续检测 DRDY。当 DRDY=1 时且接收最后一位数据时，器件捕获 I2DAT 或 I2CON 中的 RDAT 位。接着，如果要发送下一位数据，将数据写入 I2DAT。无论如何，器件必须清除 DRDY 再返回继续监测 ATN。注意：如果 ARL、STR 和 STP 中的任何一位置位，清除 DRDY 不会使 SCL 释放成高电平，I²C 总线也不能继续处理下一位数据。如果程序检测到 ATN=1 和 DRDY=0，应当继续检测 ARL、STR 和 STP。

ARL—当器件的发送激活标志被置位但在总线仲裁中失败时，ARL 为 1。ARL=1 时器件的发送激活标志清零。以下 4 种情况下 ARL 被置位：

1. 如果程序发送 ‘1’ 或重复起始条件时另一个器件发送 ‘0’ 或停止条件，在 SCL 的上升沿 SDA 的数据为 ‘0’。（如果其它器件发送停止条件，ARL 置位后 STP 也立即被置位。）
2. 当程序发送 ‘1’ 时，另一个器件发送重复起始条件，并使 SDA 在 SCL 变低之前先变低。（此时 ARL 和 STR 同时为 1。）
3. 主器件模式下，当程序发送重复起始条件时，另一个器件发送 ‘1’，并使 SCL 在 SDA 变低前先变低。
4. 主器件模式下，由于其它器件发送 ‘0’ 而使程序无法发送停止条件。

STR—当检测到非空闲从机或主机发送 I²C 总线起始条件时 STR 位置位。（当空闲从机由于起始位有效而被激活时不置位 STR 位；此时从机不会执行任何有效操作，直至 DRDY 在 SCL 上升沿置位。）

STP—当检测到非空闲从机或主机发送 I²C 总线停止条件时 STP 位置位。（空闲从机的停止条件不会置位 STP 位。）

MASTER—当器件用作 I²C 总线的主机时 ‘MASTER’ 位为 1。当 MASTRQ 为 ‘1’ 且总线空闲时 MASTER 置位。（即当由于复位或 ‘定时器 I’ 溢出而未接收到起始位或在上一个起始位后又接收到一个停止位时）。ARL 置位或 MASTRQ 软件写入 0 后 MASTER 位清零，再置位 XSTP。

8.7.4 写 I2CON

通常，对于 I²C 总线信息的每一位，都是等待 ATN=1 的一个服务子程序。根据 DRDY、ARL、STR、STP 的状态和位在信息中的当前位置，器件向 I2CON 的一位或多位写入数据，或读/写 I2DAT 寄存器。

CXA—向 ‘CXA’ 写入 ‘1’ 来清除发送激活状态（也可通过读 I2DAT 寄存器来清除发送激活状态）。

8.7.5 关于发送激活

通过写 I2DAT 寄存器或置位 I2CON 的位 XSTR 或 XSTP 来置位发送激活标志。当发送激活标志置位时，I²C 总线接口只将 SDA 线拉低且 ARL 位只能置位。发送激活标志通过读 I2DAT 寄存器或向 I2CON 的 CXA 写入 1 来清除。ARL 为 1 时发送激活标志自动清除。

IDLE—向 ‘IDLE’ 写入 1 使从机的 I²C 总线硬件忽略 I²C 总线活动直到产生下一个起始条件（但如果 MASTRQ 为 ‘1’，停止条件将使器件变为主机）。

CDR—向 ‘CDR’ 写入 1 来清除 DRDY（DRDY 也可通过读/写 I2DAT 寄存器来清除）。

CARL—向 ‘CARL’ 写入 1 来清除 ARL 位。

CSTR—向 ‘CSTR’ 写入 1 来清除 STR 位。

CSTP—向‘CSTP’写入1来清除STP位。注意：如果DRDY、ARL、STR或STP中的一位或多位为1，SCL的低电平时间将延伸到服务程序响应将其清零为止。

XSTR—仅当器件用作主机时，通过向‘XSTR’和‘CDR’写入1来触发I²C总线硬件发送一个重复起始条件。注意XSTR不能也不应该用来发送‘初始’（非重复）起始条件；初始起始条件是由I²C总线硬件自动发送的。使XSTR=1也包含了向I2DAT的XDAT写入1的作用；它将置位发送激活标志并在SCL低电平时间内将SDA释放为高电平。SCL变为高电平后，I²C总线硬件等待一段适当的最小时间后拉低SDA来产生起始条件。

XSTP—仅当器件用作主机时，通过向‘XSTP’和‘CDR’写入1来触发I²C总线硬件发送一个停止条件。如果没有更多的初始化信息，服务程序应当清除I2CFG的MASTRQ位，再向XSTP写入1。使XSTP=1也包含了向I2DAT的XDAT写入0的作用；它将置位发送激活标志并在SCL低电平时间内拉低SDA。SCL变为高电平后，I²C总线硬件等待一段适当的最小时间后拉高SDA来产生停止条件。

表 25 I2DAT—I²C总线数据寄存器位分配（地址：D9H）

不可位寻址；复位值：xxH

位	7	6	5	4	3	2	1	0
符号 (R)	RDAT	-	-	-	-	-	-	-
符号 (W)	XDAT	-	-	-	-	-	-	-

表 26 I2DAT—I²C总线数据寄存器位描述（地址：D9H）

位	符号	访问	描述
7	RDAT	R	最后接收到的数据位，每个SCL上升沿从SDA线上捕获而得。读I2DAT也可清零DRDY和发送激活状态标志。
	XDAT	W	要发送的下一位数据。写I2DAT也可清零DRDY和置位发送激活状态标志。
6~0	-	-	保留为将来之用。用户程序不要将其置1。

8.7.6 关于软件响应时间

由于P87LPC778可工作在20MHz的频率下，I²C总线接口可优化成高速操作，因此I²C总线服务程序很可能响应DRDY（在SCL上升沿时置位）并在SCL重新变为低电平前写I2DAT。如果XDAT位的数据直接被SDA使用，则违反了I²C总线规范。但是，编程者不必担心这种可能性的发生，因为只有在SCL为低电平时XDAT才能直接被SDA使用。

反过来，包含I²C总线服务子程序的程序可能会花费很长时间来响应DRDY。典型的情况是其它外围功能中断使能时，I²C总线程序根据信息传输过程中查询的标志位来执行。如果产生中断，I²C总线服务程序的响应时间会延长。编程者不必太担心，因为I²C总线硬件将会相应地延伸SCL低电平时间直到服务程序响应。响应的唯一限制就是等待时间不能超过定时器I的溢出时间。

表 27 I2CFG—I²C总线配置寄存器位分配（地址：C8H）

不可位寻址；复位值：00H

位	7	6	5	4	3	2	1	0
符号	SLAVEN	MASTRQ	CLRTI	TIRUN	-	-	CT1	CT0

表 28 I2CFG—I²C总线配置寄存器位描述（地址：C8H）

位	符号	描述
7	SLAVEN	从机使能。该位为1时使能I ² C总线系统的从机功能。如果SLAVEN和MASTRQ为0，I ² C总线硬件禁能。复位或I ² C总线定时器溢出时该位清零。

续上表...

位	符号	描述
6	MASTRQ	主机请求。通过向该位写入 1 来向 I ² C 总线请求用作主机。如果该位由 0 变为 1 时正在传输数据，请求操作延迟到检测到停止条件后。发送起始条件并置位 DRDY（这样使得 ATN=1 和产生一个 I ² C 中断）。如果主机要释放 I ² C 总线的控制权，它可以置位 I2CON 的 XSTP 位。MASTRQ 通过 I ² C 总线定时器溢出清零。
5	CLRTI	如果该位写入 1，则定时器 I 溢出标志被清零。该位读出时为 0。
4	TIRUN	如果该位写入 1，则启动定时器 I 运行；如果该位写入 0，终止或清除定时器 I。该位和 SLAVEN、MASTRQ、MASTER 一起决定了表 29 所示的工作模式。
3,2	-	保留为将来之用。用户程序不应将其置 1。
1,0	CT1,CT0	这两位可编程用作 CPU 时钟频率，在器件用作 I ² C 总线主机时用来优化 SCL 的最小高电平和低电平时间。这两位确定的时间值控制着 SCL 的时间参数以及停止和起始条件的时序。

CT1 和 CT0 对应的值如表 30 所示。将实际的振荡器频率与表中最大 CPU 时钟频率 f_{osc} 相比较，从而得到允许 I²C 总线以最大速率运行的特定振荡频率。在表中，CT1 和 CT0 的值位于第一列，CPU 时钟最大频率大于或等于实际频率。表 30 还列出了 CT1/CT0 不同组合对应的机器周期数。可以通过下式计算得出 SCL 实际的最小高电平和低电平时间：

$$SCL \text{ 最小高电平/低电平时间 (us)} = \frac{6 * \text{最小时间计数}}{CPU \text{ 时钟 (MHz)}} \quad (5)$$

例如，在 8MHz 的频率下，如果 CT1/CT0 设置为 10，则可得到 SCL 最小高电平和低电平时间为 5.25us。表 30 还列出了每组 CT1/CT0 值对应的定时器 I 的溢出周期（用机器周期表示）。由于最小 SCL 高电平和低电平时间测试方法的不同，定时器 I 的溢出周期也不同。在 I²C 总线接口工作过程中，定时器 I 的值在 SCL 跳变时预置，由 CT1/CT0 决定。必须对定时器 I 的预置值进行选择，以便定时器 I 计数值达到 008（定时器 I 的实际预置值为（8—机器周期计数））时已经过了最小 SCL 高电平或低电平时间。

表 29 TIRUN 和 SLAVEN、MASTRQ、MASTER 之间的关系

SLAVEN, MASTRQ, MASTER	TIRUN	工作模式
全为 0	0	I ² C 总线接口禁能。定时器 I 清零，停止运行。这可以看成是复位后的状态。如果一个 I ² C 总线应用想要在某段时间内忽略 I ² C 总线活动，就可将 SLAVEN,MASTRQ,MASTER 全部清零。
全为 0	1	I ² C 总线接口禁能。
不全为 0	0	I ² C 总线接口使能。定时器 I 的低 3 位运行计数，高位不变，产生一个最小时间，无法检测 I ² C 总线的‘挂起’状态。该配置用在低速 I ² C 总线操作中。
不全为 0	1	I ² C 总线接口使能。定时器 I 在 I ² C 总线传输数据帧过程中运行，SCL 跳变、起始和停止条件产生时将其清零。该配置用在正常的 I ² C 总线操作中。

表 30 CT1,CT0 值

CT1,CT0	最小时间计数 (机器周期)	CPU 最大时钟 (100MHz 的 I ² C 总线)	溢出周期 (机器周期)
10	7	8.4MHz	1023
01	6	7.2MHz	1022
00	5	6.0MHz	1021
11	4	4.8MHz	1020

8.8 中断

P87LPC778 采用 4 中断优先级结构。这为 P87LPC778 的中断源的处理提供了极大的灵活性。P87LPC778 支持 13 个中断源。

每个中断源可通过置位或清零寄存器 IEN0 或 IEN1 中相应的位实现单独使能或禁能。IEN0 寄存器还包含一个全局禁能位 EA，它可禁止所有中断。

每个中断源可编程为 4 个中断优先级之一，通过置位或清零 IP0，IP0H，IP1 和 IP1H 的相应位来实现。正在执行的中断服务程序可响应更高优先级中断，但不响应同优先级或低优先级的中断。最高优先级中断不会被任何其它中断源中断。因此，当两个不同优先级的中断源同时申请中断时，响应高优先级中断。

如果多个同优先级的中断源同时申请中断，通过一个内部查询程序确定首先响应哪一个中断请求，这被称之为仲裁队列。注意：仲裁队列只用来处理同优先级中断源申请中断的情况。

表 31 对中断源、标志位、向量地址、使能位、优先级位、仲裁队列和中断是否能将 CPU 从掉电模式唤醒进行了小结。

表 31 中断小结

描述	中断标志位	向量地址	中断使能位	中断优先级	仲裁队列	掉电唤醒
外部中断 0	IE0	0003h	EX0(IEN0.0)	IP0H.0,IP0.0	1 (最高)	能
定时器 0 中断	TF0	000Bh	ET0(IEN0.1)	IP0H.1,IP0.1	4	不能
外部中断 1	IE1	0013h	EX1(IEN0.2)	IP0H.2,IP0.2	7	能
定时器 1 中断	TF1	001Bh	ET1(IEN0.3)	IP0H.3,IP0.3	10	不能
串口 Tx 和 Rx	TI&RI	0023h	ES(IEN0.4)	IP0H.4,IP0.4	12	不能
掉电检测	BOD	002Bh	EBO(IEN0.5)	IP0H.5,IP0.5	2	能
I ² C 总线中断	ATN	0033h	EI2(IEN1.0)	IP1H.0,IP1.0	5	不能
KBI 中断	KBF	003Bh	EKB(IEN1.1)	IP1H.1,IP1.1	8	能
比较器 2 中断	CMF2	0043h	EC2(IEN1.2)	IP1H.2,IP1.2	11	能
看门狗定时器	WDOVF	0053h	EWD(IEN0.6)	IP0H.6,IP0.6	3	能
A/D 转换器	ADCI	005Bh	EAD(IEN1.4)	IP1H.4,IP1.4	6	能
比较器 1 中断	CMF1	0063h	EC1(IEN1.5)	IP1H.5,IP1.5	9	能
定时器 I 中断	-	0073h	ETI(IEN1.7)	IP1H.7,IP1.7	13 (最低)	不能

8.8.1 外部中断输入

P87LPC778 有 2 个中断输入和键盘中断功能。键盘中断在后面章节中单独描述。两个中断输入与标准 80C51 微控制器的中断输入完全相同。

外部中断通过置位或清零 TCON 寄存器的位 IT1 或 IT0 编程为电平触发或边沿触发。如果 ITn=0，外部中断 n 由 \overline{INTn} 脚的低电平来触发。如果 ITn=1，外部中断 n 为边沿触发。边沿触发模式下，对 \overline{INTn} 脚进行连续采样，如果采样到一个周期为高电平下一个周期为低电平，TCON 中的中断请求标志 IEn 置位，产生中断请求。

由于外部中断管脚一个机器周期采样一次，因此，为了得到正确的采样结果，中断输入的高或低电平

必须保持至少 6 个 CPU 时钟。如果外部中断为边沿触发，外部中断源必须先后使请求管脚的高电平和低电平分别至少保持 1 个机器周期，以保证边沿能被检测到，进而置位中断请求标志 IEn。当调用中断服务时，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有效，直至产生中断请求。如果外部中断在中断服务执行完成后仍然有效，将产生下次中断。当中断为电平触发时不需要清除中断标志 IEn，因为中断只跟中断输入脚的电平有关。

如果 P87LPC778 处于掉电或空闲模式时外部中断使能，中断将唤醒处理器，使其恢复正常运行。详情请参考第 8.12 节“节电模式”。

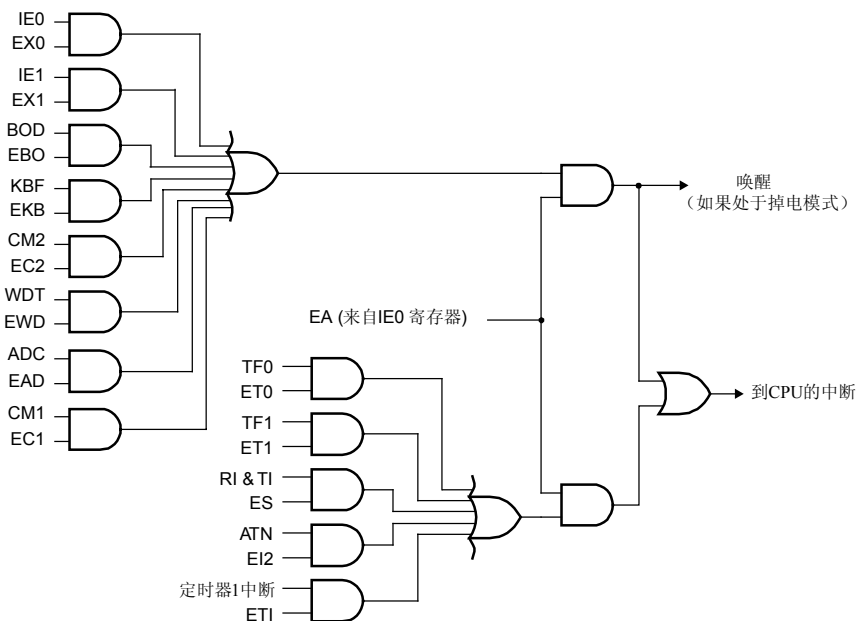


图 9 中断源，中断使能和掉电唤醒源

8.9 I/O 口

P87LPC778 有 3 个 I/O 口，P0 口、P1 口和 P2 口。具体的 I/O 口数目取决于所选的振荡器和复位方式。当使用两个管脚的外部振荡器和外部复位电路时，P87LPC778 至少有 15 个可用的 I/O 口。如果使用片内振荡器和片内复位，可使用的 I/O 口高达 18 个。

除 3 个 I/O 口外，P87LPC778 的其它 I/O 均可由软件配置成 4 中输出类型之一，见表 32。这 4 种输出类型为双向（标准 80C51 端口输出）、推挽、开漏和仅为输入模式。每个端口的输出类型由相应的 2 个配置寄存器进行选择。

表 32 I/O 口输出配置设定

PxM1.y	PxM2.y	I/O 口输出模式
0	0	准双向
0	1	推挽
1	0	仅为输入（高阻）
1	1	开漏

8.9.1 准双向输出配置

准双向模式是标准 80C51 及其大多数衍生器件 I/O 口的输出模式，它也是 P87LPC778 I/O 口默认的输出模式。该模式下，I/O 口不需要重新配置就可直接用作输入和输出功能。这是因为当口线输出为 1 时驱动能力很弱，外部器件可以将其拉低。当管脚输出为低时，它的驱动能力很强可吸收很大的电流。准双向口

除了有三个上拉晶体管适应不同的需要外，和开漏输出有点相似。

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

第二个晶体管称为‘弱’上拉，当口线锁存器为 1 且管脚本身也为 1 时打开。该上拉提供拉电流使准双向口输出为‘1’。如果管脚输出为 1 时由外部器件下拉到低时，弱上拉关闭，只有极弱上拉维持开状态。这时为了将管脚强拉为低，外部器件必须有足够大的灌电流能力使管脚上的电压降到它的输入门槛电压以下。

第三个晶体管称为‘强’上拉。当口线锁存由 0 变为 1 时，该上拉用来加快准双向口从低到高的转变。当发生这种情况时，强上拉打开约 2 个 CPU 时钟以使管脚能够迅速上拉到高电平，然后再关闭。

准双向口配置见图 10。

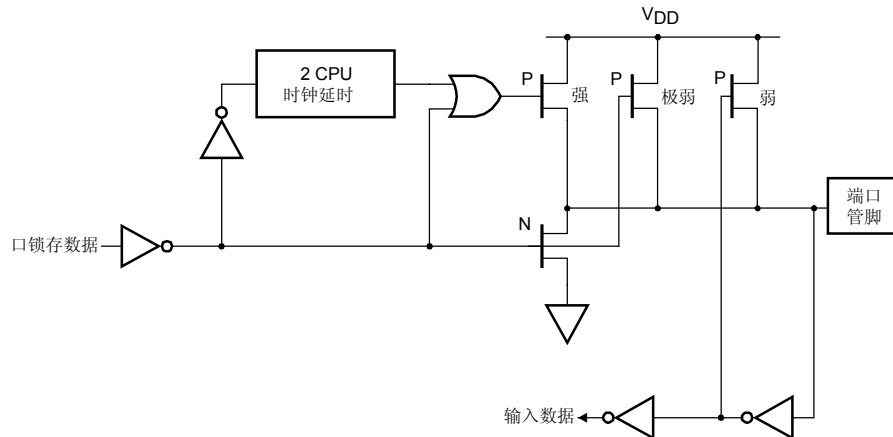


图 10 准双向口输出

8.9.2 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，只驱动下拉晶体管。当用作一个逻辑输出时，这种配置必须有外部上拉，一般通过电阻外接到 V_{DD} 。该模式中的下拉与准双向模式相同。

开漏输出配置见图 11。

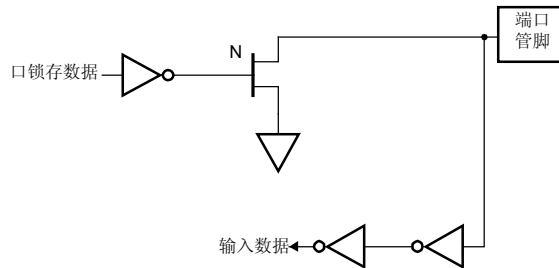


图 11 开漏输出

8.9.3 推挽输出配置

推挽输出的下拉结构与开漏和准双向口输出相同。但当口线锁存器为 1 时，它可提供一个持续的强上拉。推挽模式一般用于需要更大拉电流的情况。

推挽输出配置见图 12。

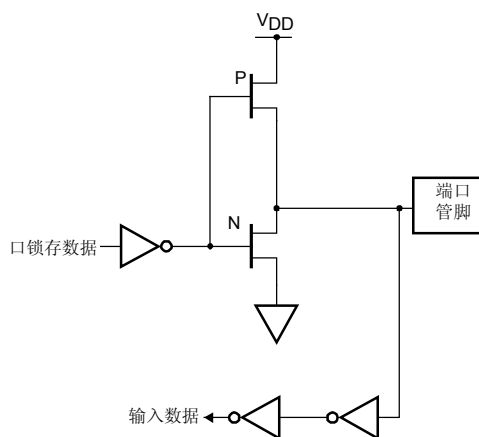


图 12 推挽输出

P1.2、P1.3 和 P1.5 不能选择输出方式。P1.2 和 P1.3 永远为开漏输出。向其对应的口锁存器写入 1 时可用作输入。当 P87LPC778 使用内部复位而不使用外部复位输入 \overline{RST} 时，P1.5 可用作施密特触发输入。

另外，当 P87LPC778 使用晶体振荡器时，P2.0 和 P2.1 不能用作 I/O 口。详见第 8.10 节“振荡器”。

复位后的 I/O 管脚值由 UCFG1 寄存器的 PRHI 位决定。I/O 管脚可根据应用的需要配置成复位后为高或低。当复位后口线设置为高时，这些口线为准双向口，不能输出大电流。

每个 P87LPC778 输出口都可提供 20mA 的吸入电流驱动 LED。但是所有口的输出电流总和不能超过规定的最大电流 80 mA。

P87LPC778 各个端口的电平转换速度可以控制，这就可避免因电平转换过快而导致的噪声。转换时间由出厂时设定，上升时间和下降时间大约为 10ns 时间。

当 P2.0 和 P2.1 用作其它功能时，P2M1 中的控制位无法控制选择其输出模式。寄存器的各位可启用 I/O 口的施密特输入、选择定时器 0 和定时器 1 的输出、或在器件使用内部 RC 振荡器或外部时钟输入时用来使能时钟输出。后两项功能见第 8.14 节“定时器/计数器”和第 8.10 节“振荡器”。上述功能的使能位如表 33 和表 34 所示。

每个 P87LPC778 的 I/O 口可选择为 TTL 电平输入或施密特触发输入。用一个位即可对其选择。对于 P1.2，P1.3 和 P1.5 只能选择为施密特输入方式。

表 33 P2M1—P2 口方式寄存器 1 位分配（地址：A4h）

不可位寻址；复位值：00H

位	7	6	5	4	3	2	1	0
符号	P2S	P1S	P0S	ENCLK	ENT1	ENT0	(P2M1.1)	(P2M1.0)

表 34 P2M1—P2 口方式寄存器 1 位描述（地址：A4h）

位	符号	描述
7	P2S	当 P2S=1 时，P2 口施密特触发输入使能。
6	P1S	当 P1S=1 时，P1 口施密特触发输入使能。
5	P0S	当 P0S=1 时，P0 口施密特触发输入使能。
4	ENCLK	当 ENCLK 置位且 P87LPC778 使用内部 RC 振荡器时，X2 脚（P2.0）的时钟输出使能。详情请参考第 8.10 节“振荡器”。
3	ENT1	该位置位时，定时器 1 溢出将触发 P0.7。输出频率为定时器 1 溢出速率的一半。详情请参考第 8.14 节“定时器/计数器”。
2	ENT0	该位置位时，定时器 0 溢出将触发 P1.2。输出频率为定时器 0 溢出速率的一半。详情请参考第 8.14 节“定时器/计数器”。
1,0	-	这两位和 P2M2 寄存器中的相应的位一道控制着 P2.1 和 P2.0 的输出配置，见表 32。

8.9.4 键盘中断 (KBI)

键盘中断功能用于在键盘的键或连接到 P87LPC778 特殊管脚的按键被按下时产生中断, 如图 13 所示。该中断可将 CPU 从空闲或掉电模式下唤醒。此特性尤其适用于便携式且使用电池供电的系统。

P87LPC778 允许部分或全部 P0 口使能用作键盘中断输入, 通过置位 KBI 寄存器中的相应位来实现, 如表 35 和 36 所示。KBI 中断功能激活后, 只要任何一个使能管脚被拉低, AUXR1 寄存器的键盘中断标志 (KBF) 置位, 产生中断 (如果使能)。注意: KBF 必须通过软件清零。

由于人对时间的分辨能力有限以及键开关机械延时的存在, 使用 KBI 特性时通常要通过在中断服务程序中查询 P0 口来确定按下的是哪个键, 尽管这样可能会将 CPU 从掉电模式唤醒。详情请参考第 8.12 节“节电模式”。

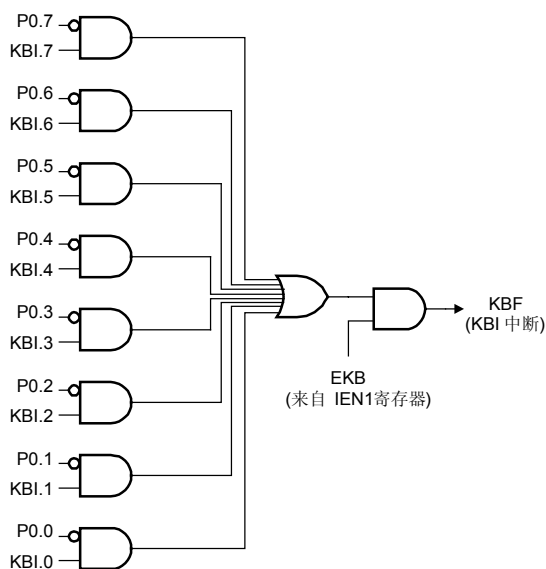


图 13 键盘中断

表 35 KBI—键盘中断寄存器位分配 (地址: 86H)

不可位寻址; 复位值: 00H

位	7	6	5	4	3	2	1	0
符号	KBI.7	KBI.6	KBI.5	KBI.4	KBI.3	KBI.2	KBI.1	KBI.0

表 36 KBI—键盘中断寄存器位描述 (地址: 86H)

位	符号	描述
7	-	该位置位来使能 P0.7 的键盘中断功能。
6	-	该位置位来使能 P0.6 的键盘中断功能。
5	-	该位置位来使能 P0.5 的键盘中断功能。
4	-	该位置位来使能 P0.4 的键盘中断功能。
3	-	该位置位来使能 P0.3 的键盘中断功能。
2	-	该位置位来使能 P0.2 的键盘中断功能。
1	-	该位置位来使能 P0.1 的键盘中断功能。
0	-	该位置位来使能 P0.0 的键盘中断功能。

8.10 振荡器

P87LPC778 提供了几种振荡器选项, 允许根据需从高精度到低成本范围内进行优化。这些选项在编程 EPROM 时配置。支持的基本振荡器类型包括低频、中频和高频晶振, 频率范围为 20kHz~20MHz, 也

支持陶瓷振荡器和片内 RC 振荡器。

8.10.1 低频振荡器选项

此选项支持 20kHz~100kHz 的外部晶体。

表 37 选择低频振荡器选项时推荐使用的振荡器电容

振荡器频率	$V_{DD}=2.7\sim 4.5V$			$V_{DD}=4.5V\sim 5.5V$		
	下限	理想值	上限	下限	理想值	上限
20kHz	15pF	15pF	33pF	33pF	33pF	47pF
32kHz	15pF	15pF	33pF	33pF	33pF	47pF
100kHz	15pF	15pF	33pF	15pF	15pF	33pF

8.10.2 中频振荡器选项

此选项支持 100kHz~4MHz 的外部晶体。也支持陶瓷振荡器。

表 38 选择中频振荡器选项时推荐使用的振荡器电容

振荡器频率	$V_{DD}=2.7\sim 4.5V$			$V_{DD}=4.5V\sim 5.5V$		
	下限	理想值	上限	下限	理想值	上限
100kHz	33pF	33pF	47pF	33pF	33pF	47pF
1MHz	15pF	15pF	33pF	15pF	22pF	47pF
4MHz	15pF	15pF	33pF	15pF	15pF	33pF

8.10.3 高频振荡器选项

此选项支持 4MHz~20MHz 的外部晶体。也支持陶瓷振荡器。

表 39 选择高频振荡器选项时推荐使用的振荡器电容

振荡器频率	$V_{DD}=2.7\sim 4.5V$			$V_{DD}=4.5V\sim 5.5V$		
	下限	理想值	上限	下限	理想值	上限
4MHz	15pF	33pF	47pF	15pF	33pF	68pF
8MHz	15pF	15pF	33pF	15pF	33pF	47pF
16MHz	-	-	-	15pF	15pF	33pF
20MHz	-	-	-	15pF	15pF	33pF

8.10.4 片内 RC 振荡器选项

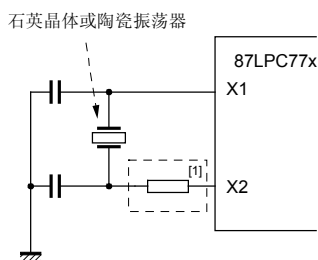
片内 RC 振荡器的典型频率是 6MHz，可通过 DIVM 寄存器分频获取更低的频率。如果使用片内 RC 振荡器，使能 X2/P2.0 管脚的时钟输出。

8.10.5 外部时钟输入选项

此配置中，外部时钟从 X1/P2.1 脚输入，用作处理器时钟。当 V_{DD} 大于 4.5V 时，频率可从 0Hz~20MHz。当 V_{DD} 小于 4.5V 时，频率可达到 10MHz。当使用外部时钟输入模式时，X2/P2.0 脚可用作标准 I/O 口或外部时钟输出。

8.10.6 时钟输出

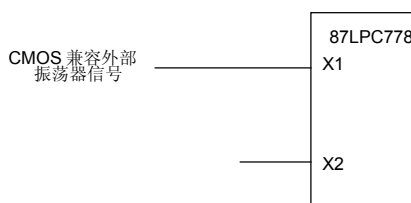
当使用片内 RC 振荡器或外部时钟输入时，P87LPC778 支持时钟输出功能。这就使得外部器件可与 P87LPC778 同步。如果 P2M1 寄存器的 ENCLK 位置位，无论器件是否处于空闲状态，只要片内振荡器运行，时钟信号就从 X2/CLKOUT 输出。时钟输出频率为 CPU 时钟的 1/6。如果空闲模式不需要时钟输出，可将时钟关闭来节省功率。选择外部时钟输入选项时也可使能时钟输出。



振荡器必须配置成以下模式：低频、中频或高频振荡器。

(1) 串联一个电阻来限制晶体驱动电平。这在低频振荡中尤其重要（见文中描述）。

图 14 使用晶振



振荡器必须配置成外部时钟输入模式。通过置位 P2M1 寄存器的位 ENCLK 来使时钟信号从 X2 脚输出。

图 15 使用外部时钟输入

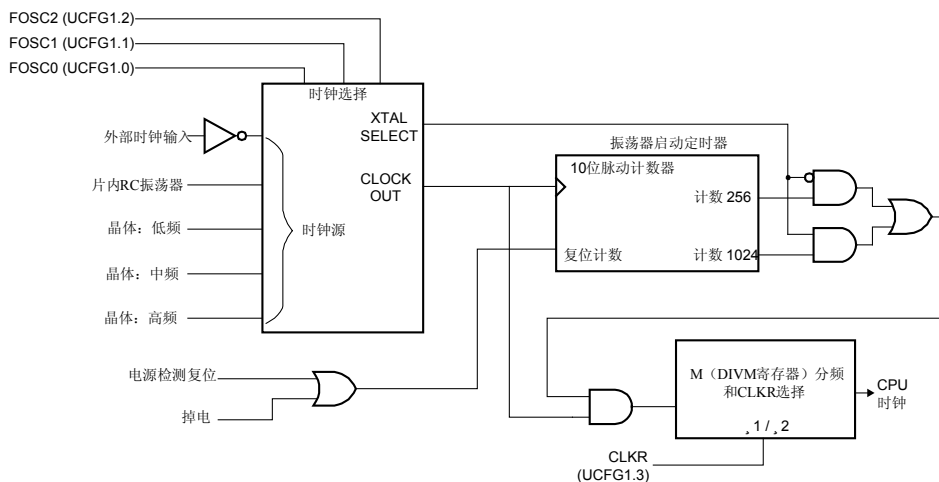


图 16 振荡器控制方框图

8.10.7 CPU 时钟调整：CLKR 和 DIVM

为了向下兼容, CLKR 配置位通过 2 分频 CPU 时钟将 P87LPC778 指令和外围时序设置成与标准 80C51 时序相匹配。默认 P87LPC778 每个机器周期包含 6 个 CPU 时钟, 而标准 80C51 器件每个机器周期包含 12 个 CPU 时钟。该分频值还可用于外围时序, 允许 80C51 代码分别以振荡器频率和/或定时器溢出频率运行。CLKR 位位于 EPROM 配置寄存器 UCFG1, 详见第 8.18 节“EPROM 特性”。

另外, 也可通过编程分频器对振荡器进行分频得到 CPU 时钟, 由 DIVM 寄存器来控制。如果 DIVM 寄存器的值为 0 (默认), CPU 以振荡器或 2 分频的振荡器为时钟源, 由上述的 CLKR 功能决定。

当 DIVM 的值 N 为 1~255 时, CPU 时钟被 $2 \times (N+1)$ 分频。时钟分频值为 4~512。这就使得 CPU 可暂时以较低的频率运行以减少功耗, 类似于空闲模式。时钟分频后, 除响应中断事件 (例如, 使 CPU 退出空闲模式的事件) 外, CPU 仍然保持着对其它事件的响应能力, 这就比掉电模式少了振荡器启动时间。DIVM

的值可在程序中随时改变而不中断程序的执行。

8.11 电源监控功能

P87LPC778 集成了电源监控功能，用来防止初始上电时的错误操作和正常工作时的断电或掉电。通过两个硬件功能来实现：上电检测和掉电检测。

8.11.1 掉电检测

掉电检测用于防止在电源降至某个特定电压值时处理器的错误操作。掉电检测的默认动作是复位处理器，或通过置位 AUXR1 寄存器的 BOI 位（AUXR1.5）来产生中断。

P87LPC778 支持两个掉电电平：2.5V 或 3.8V。当 V_{DD} 低于所选掉电电压时，掉电检测器触发并保持到 V_{DD} 上升到掉电检测电压之上。当掉电检测产生一次处理器复位，只要 V_{DD} 低于掉电电压，处理器就一直保持复位状态。当 V_{DD} 从大于掉电电压降到小于掉电电压时产生掉电检测中断。如果中断系统和 BOI 中断被使能（通过设置 IEN0 的 EA 和 EBO 位），中断被响应。

如果掉电检测被激活，置位 PCON 寄存器的 BOF 标志使得处理器的复位源可通过软件进行设定。BOF 标志由软件清零，清零前一直保持有效。

为了正确激活掉电检测， V_{DD} 下降时间不能快于 50mV/us。为了器件的正确复位， V_{DD} 恢复时的上升时间不能快于 2mV/us。

掉电电压（2.5V 或 3.8V）通过 EPROM 配置寄存器 UCFG1 的 BOV 位来选择。当 BOV=1 时，掉电检测电压为 2.5V。当 BOV=0 时，掉电检测电压为 3.8V。

如果在应用中不需要掉电检测功能，可以将其关闭来节省功率。掉电检测通过置位 AUXR1 寄存器的控制位 BOD（AUXR1.6）来禁能。

8.11.2 上电检测

上电检测的功能与掉电检测类似，它设计成在电源初始上电、电源电压上升到掉电检测电压之前工作。上电检测激活后，PCON 寄存器的 POF 标志置位来指示初始上电条件。POF 标志的状态一直保持，直到通过软件清零。

8.12 节电模式

P87LPC778 支持空闲和掉电两种节电模式。

8.12.1 空闲模式

空闲模式下外围器件继续工作，产生中断时可激活处理器。任何使能的中断源或复位源都可终止空闲模式。将 PCON 寄存器的 IDL 位置位可进入空闲模式（见表 41 和 42）。

8.12.2 掉电模式

掉电模式下关闭振荡器可最大限度地降低功耗。通过置位 PCON 寄存器（见表 41 和 42）的 PD 位就可进入掉电模式。

复位或任何一个中断源（见表 40）都可使处理器退出掉电模式。前提条件是中断被使能且优先级高于正在处理的中断。

掉电模式下，电源电压可能降低至 RAM 保持有效电压 V_{RAM} 。这就使得 RAM 保存为进入掉电模式时的内容。 V_{DD} 低于 V_{RAM} 后 SFR 的内容不受保护，因此建议此时通过复位来唤醒处理器。 V_{DD} 必须在退出掉电模式之前上升到工作电压范围内。由于看门狗定时器具有独立的振荡器，如果它在掉电模式下继续工作，溢出时可将处理器复位。

注：如果掉电检测复位使能， V_{DD} 低于掉电电压时处理器将被复位。如果掉电检测配置成中断并使能，

当 V_{DD} 低于掉电电压时，它可将处理器从掉电模式唤醒。

当处理器从掉电模式唤醒后，振荡器立即启动并在振荡器运行稳定后开始执行程序。如果使用外部晶振，振荡器稳定时间为起振后 1024 个 CPU 时钟。如果使用内部 RC 或外部时钟输入，振荡器稳定时间为起振后 256 个时钟。

表 40 中断源

唤醒源	条件
外部中断 0 或 1	相应的中断必须使能。
键盘中断	键盘中断特性必须使能和正确设置。相应的中断必须使能。
比较器 1 或 2	比较器必须使能和正确设置。相应的中断必须使能。
看门狗定时器复位	看门狗定时器必须通过设置 UCFG1 EPROM 配置字节的 WDTE 位来使能。
看门狗定时器中断	必须清零 UCFG1 EPROM 配置字节的 WDTE 位。相应的中断必须使能。
掉电检测复位	AUXR1 的 BOD 位必须清零（掉电检测使能）。AUXR1 的 BOI 位必须清零（掉电中断禁能）。
掉电检测中断	AUXR1 的 BOD 位必须清零（掉电检测使能）。AUXR1 的 BOI 位必须置位（掉电中断使能）。相应的中断必须使能。
复位输入	外部复位输入必须使能。
A/D 转换器	内部 RC 时钟（RCCLK=1）用作 A/D 转换时钟，使其可工作在掉电模式。A/D 必须使能和正确设置。相应的中断必须使能。

由于某些芯片功能可在掉电模式下继续工作和消耗电流，因此大大增加了掉电模式的总功耗。这部分功能包括掉电检测、看门狗定时器和比较器。

8.12.3 低电压 EPROM 操作

EPROM 阵列包含一些模拟电路，这些电路 V_{DD} 小于 4V 时需要， V_{DD} 大于 4V 时不需要。如果 LPEP 位（AUXR1.4）通过软件置位，这部分模拟电路被断开以节省功耗。LPEP 只能通过上电复位清除，因此它只在 V_{DD} 小于 4V 的应用中被置位。

表 41 PCON—功率控制寄存器位分配（地址：87H）

不可位寻址；复位值：上电复位时为 30H，掉电复位时为 20H，其它复位源复位时为 00H。

位	7	6	5	4	3	2	1	0
符号	SMOD1	SMOD0	BOF	POF	GF1	GF0	PD	IDL

表 42 PCON—功率控制寄存器位描述（地址：87H）

位	符号	描述
7	SMOD1	该位置位时，模式 1、2 和 3 的 UART 波特率加倍。
6	SMOD0	该位用来选择 SCON 寄存器位 7 的功能。该位为 0 时，SCON.7 是 SM0 位；该位为 1 时，SCON.7 是 FE（帧错误）标志。相关信息见表 48 和 49。
5	BOF	掉电标志。上电、掉电复位或中断产生时该位自动置位，也可在上电时置位。该位通过软件清零。相关信息请参考第 8.11 节“电源监控功能”。
4	POF	上电标志。上电复位时该位自动置位。该位通过软件清零。相关信息请参考第 8.11 节“电源监控功能”。
3	GF1	通用标志 1。可通过用户软件读出或写入，不影响任何操作。
2	GF0	通用标志 0。可通过用户软件读出或写入，不影响任何操作。
1	PD	掉电控制位。该位置位时掉电模式激活。掉电模式终止时该位清零（见文中描述）。
0	IDL	空闲模式控制位。该位置位时空闲模式激活，空闲模式终止时该位清零（见文中描述）。

8.13 复位

P87LPC778 有一个可配置成外部复位的低有效的复位输入脚。另外，复位还可配置成内部复位方式，用于在初始上电时使器件复位。由于看门狗定时器具有独立的片内振荡器，它可用作振荡器失效检测。

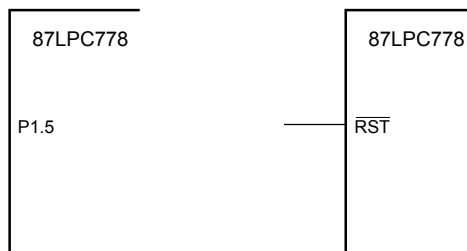


图 17 典型的外部复位电路

通过清零 EPROM 配置寄存器 UCFG1 的位 RPD 来禁止外部复位输入和使能内部复位的产生。有关 EPROM 配置的描述见第 8.18 节“EPROM 特性”。

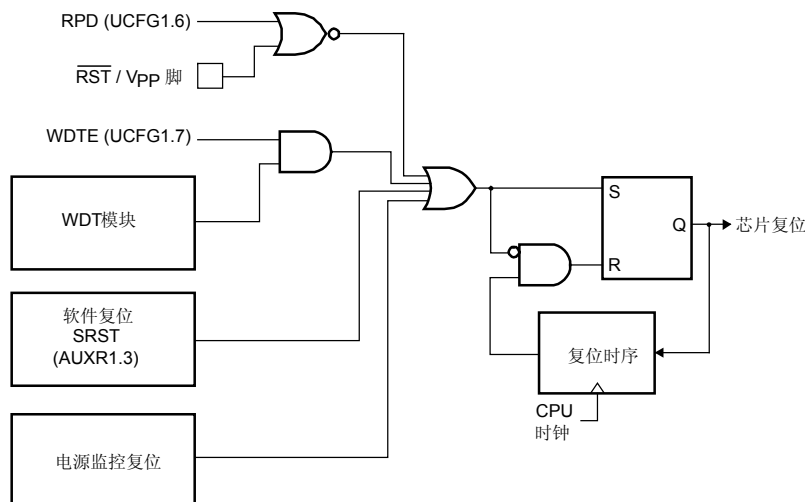


图 18 复位源方框图

8.14 定时器/计数器

P87LPC778 有 2 个通用定时器/计数器，与标准 80C51 定时器 0 及定时器 1 兼容。这 2 个定时器均可配置为定时器或事件计数器（见表 43 和 44）。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，每经过一个机器周期，寄存器的值加 1。因此，我们可以将其看成是对机器周期进行计数。由于一个机器周期由 6 个 CPU 时钟周期组成，所以定时器的计数频率为 CPU 时钟频率的 1/6。有关 CPU 时钟的描述详见第 8.1 节“增强型 CPU”。

用作‘计数’功能时，外部输入脚 T0 或 T1 每产生一次 1 到 0 的跳变，寄存器的值加 1。使用该功能时，外部输入每个机器周期被采样一次。当管脚某一周期采样为高而下一周期采样为低时，计数值加 1。寄存器值在检测到跳变后的下一个周期被更新。由于检测下降沿跳变需要 2 个机器周期（12 个 CPU 时钟），所以最大计数频率为 1/6 CPU 时钟。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。

‘定时器’或‘计数器’功能由特殊功能寄存器 TMOD 的控制位 C/T 来选择。除此之外，定时器 0 和定时器 1 还有 4 种工作模式，取决于 TMOD 的位 (M1,M0) 的值。两个定时器/计数器的模式 0、1 和 2 相同，模式 3 不同。这 4 种工作模式见后面的描述。

表 43 TMOD—定时器/计数器方式控制寄存器位分配（地址：89H）

不可位寻址；复位值：00H

位	7	6	5	4	3	2	1	0
符号	GATE	C/\bar{T}	M1	M0	GATE	C/\bar{T}	M1	M0

表 44 TMOD—定时器/计数器方式控制寄存器位描述（地址：89H）

位	符号	描述
7	GATE	定时器 1 的门控位。该位置位时，只有当 $\overline{INT1}$ 脚为高电平且 TR1 控制位置位时，定时器/计数器 1 才使能。该位清零时，TR1 控制位置位时定时器 1 使能。
6	C/\bar{T}	定时器 1 的定时器或计数器选择器。该位为 0 时定时器 1 用作定时器（计数内部系统时钟）；该位为 1 时定时器 1 用作计数器（计数 T1 输入脚的脉冲）。
5,4	M1,M0	定时器 1 工作模式选择（见下表）。
3	GATE	定时器 0 的门控位。该位置位时，只有当 $\overline{INT0}$ 脚为高电平且 TR0 控制位置位时，定时器/计数器 0 才使能。该位清零时，TR0 控制位置位时定时器 0 使能。
2	C/\bar{T}	定时器 0 的定时器或计数器选择器。该位为 0 时定时器 0 用作定时器（计数内部系统时钟）；该位为 1 时定时器 0 用作计数器（计数 T0 输入脚的脉冲）。
1,0	M1,M0	定时器 0 工作模式选择（见下表）。

表 45 M0,M1 定时器工作模式

M1,M0	定时器工作模式
00	8048 定时器，‘TLn’ 用作 5 位的预分频器。
01	16 位定时器/计数器。‘THn’ 和 ‘TLn’ 级联；无预分频器。
10	8 位自动重装定时器/计数器。定时器/计数器溢出时 THn 的值装入 TLn。
11	定时器 0 用作两个 8 位的定时器/计数器。TL0 用作一个 8 位的定时器/计数器，由标准定时器 0 的控制位来控制。TH0 只能用作 8 位的定时器，由定时器 1 的控制位进行控制（见文中描述）。该模式下定时器 1 关闭。

8.14.1 模式 0

定时器工作在模式 0 类似于一个 8048 定时器，即带 32 分频的预分频器的 8 位计数器。模式 0 的操作见图 19。

该模式下，定时器配置成一个 13 位的寄存器。当计数值从全 1 变为全 0 时，定时器中断标志 TF_n 置位。当 TR_n=1，GATE=0 或 $\overline{INTn}=1$ 时，定时器计数输入使能。（置位 GATE 可使外部输入 \overline{INTn} 对定时器进行控制，以便于实现脉宽的测量）。TR_n 是特殊功能寄存器 TCON 的控制位（表 46 和 47）。GATE 位位于 TMOD 寄存器。

13 位的寄存器由 8 位 TH_n 和低 5 位 TL_n 组成。TL_n 高 3 位的值不确定，可以忽略。置位运行标志 (TR_n) 不会清零寄存器。

定时器 0 和定时器 1 的模式 0 操作相同，见图 19。有两个不同的 GATE 位，一个用于定时器 1 (TMOD.7)，一个用于定时器 0 (TMOD.3)。

8.14.2 模式 1

除定时器寄存器为 16 位外，模式 1 的模式 0 的操作相同。见图 20。

8.14.3 模式 2

该模式下，定时器寄存器配置成自动重装的 8 位计数器 (TL1)，如图 21 所示。TL_n 的溢出不仅置位

TFn, 而且将 THn 的值装入 TLn, THn 的值由软件预置。重装时 THn 的内容不变。定时器 0 和定时器 1 的模式 2 相同。

8.1.4.4 模式 3

该模式下, 定时器 1 关闭, 等效于 TR1=0。

该模式下, TL0 和 TH0 用作两个独立的 8 位计数器。定时器 0 的模式 3 逻辑如图 22 所示。TL0 使用定时器 0 的控制位: C/T、GATE、TR0、INT0 和 TF0。TH0 只能用作定时器功能 (计数器周期), 使用定时器 1 的 TR1 和 TF1。这样, TH0 就控制着 ‘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 工作在模式 3 下时, P87LPC778 可以看作有 3 个定时器/计数器。当定时器 0 工作在模式 3 下时, 定时器 1 可通过进入或退出模式 3 来开/关。它仍可用于串口的波特率发生器, 或应用在任何不需要中断的场合。

表 46 TCON—定时器/计数器控制寄存器位分配 (地址: 88H)

可位寻址; 复位值: 00H

位	7	6	5	4	3	2	1	0
符号	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

表 47 TCON—定时器/计数器控制寄存器位描述 (地址: 88H)

位	符号	描述
7	TF1	定时器 1 溢出标志。定时器/计数器 1 溢出时由硬件置位。中断处理时由硬件清零, 也可通过软件清零。
6	TR1	定时器 1 运行控制位。由软件置位/清零该位来开/关定时器/计数器 1。
5	TF0	定时器 0 溢出标志。定时器/计数器 0 溢出时由硬件置位。中断处理时由硬件清零, 也可通过软件清零。
4	TR0	定时器 0 运行控制位。由软件置位/清零该位来开/关定时器/计数器 0。
3	IE1	中断 1 边沿标志。检测到外部中断 1 跳变沿时由硬件置位。中断处理时由硬件清零, 也可通过软件清零。
2	IT1	中断 1 类型控制位。由软件置位/清零该位来选择外部中断以下降沿/低电平方式触发。
1	IE0	中断 0 边沿标志。检测到外部中断 0 跳变沿时由硬件置位。中断处理时由硬件清零, 也可通过软件清零。
0	IT0	中断 0 类型控制位。由软件置位/清零该位来选择外部中断以下降沿/低电平方式触发。

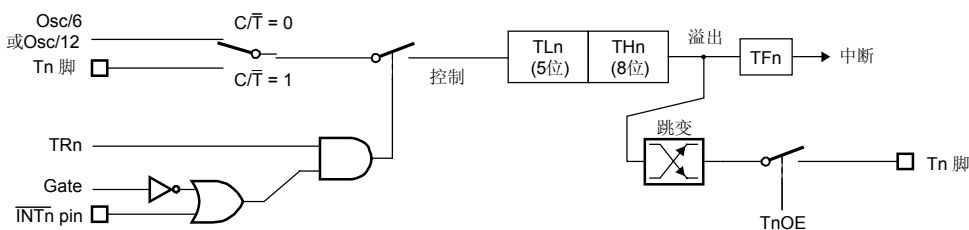


图 19 定时器/计数器 0 或 1 的模式 0 (13 位计数器)

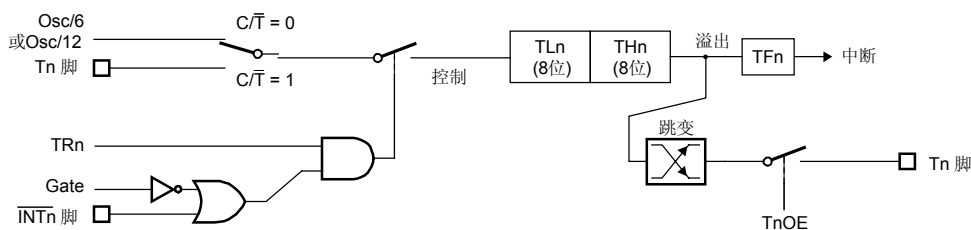


图 20 定时器/计数器 0 或 1 的模式 1 (16 位计数器)

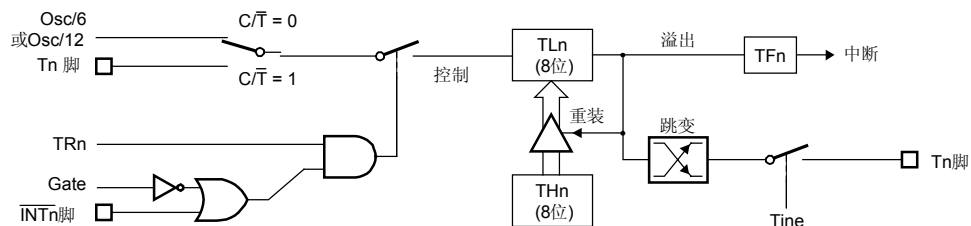


图 21 定时器/计数器 0 或 1 的模式 2 (8 位自动重装)

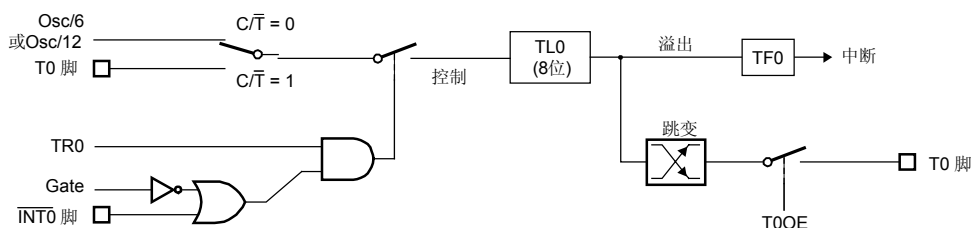
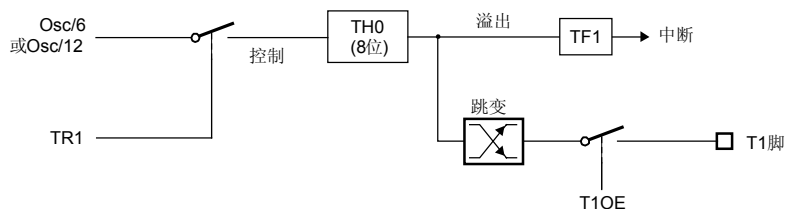


图 22 定时器/计数器 0 的模式 3 (2 个 8 位计数器)



8.14.5 定时器溢出触发输出

定时器 0 和 1 可配置为溢出时自动触发端口输出。T0、T1 的计数输入和定时器触发输出占用相同的管脚。此功能由 P2M1 寄存器的位 ENT0 和 ENT1 使能，供定时器 0 和定时器 1 应用。此模式打开时，在首次溢出前端口输出为高。

8.15 UART

P87LPC778 包含一个增强型 80C51 UART。模式 1 和模式 3 中，UART 的波特率由定时器 1 产生，模式 0 和 2 中的波特率固定。因为 P87LPC778 的 CPU 时钟与标准 80C51 器件不同，所以两者波特率的计算方法也有所不同。标准 80C51 UART 的增强特性包括帧错误检测和自动地址识别。

串行端口是全双工的，可以同时接收和发送数据。串口的接收带缓冲功能，可在读出 SBUF 寄存器中的字节前启动第二个字节的接收。但是，如果第二个字节接收完毕后第一个字节还未读出，则第一个字节丢失。串口的接收和发送寄存器都通过特殊功能寄存器 SBUF 访问。写 SBUF 相当于装载发送寄存器，读

SBUF 相当于访问一个逻辑上独立的接收寄存器。

串口有 4 种工作模式。

8.15.1 模式 0

串行数据通过 RxD 进出, TxD 输出移位时钟。发送或接收 8 位数据, LSB 在前。波特率固定为 1/6 CPU 时钟频率。

8.15.2 模式 1

TxD 脚发送, RxD 脚接收, 每次传输 10 位数据: 1 个起始位 (逻辑 0)、8 个数据位 (LSB 在前) 和 1 个停止位 (逻辑 1)。接收数据时, 停止位存放到特殊功能寄存器 SCON 的 RB8。波特率可变, 由定时器 1 的溢出率决定。

8.15.3 模式 2

TxD 脚发送, RxD 脚接收, 每次传输 11 位数据: 起始位 (逻辑 0)、8 个数据位 (LSB 在前)、1 个可编程第 9 位数据和 1 个停止位 (逻辑 1)。发送数据时, 第 9 个数据位 (SCON 的 TB8) 编程为 0 或 1。或者, 例如, 将奇偶位 (PSW 的 P) 写 TB8。接收数据时, 第 9 个数据位移入 SCON 的 RB8, 忽略停止位。波特率可编程为 1/16 或 1/32 CPU 时钟频率, 由 PCON 的 SMOD1 位决定。

8.15.4 模式 3

TxD 脚发送, RxD 脚接收, 每次传输 11 位数据: 1 个起始位 (逻辑 0)、8 个数据位 (LSB 在前)、1 个可编程第 9 位数据和 1 个停止位 (逻辑 1)。实际上, 除波特率外, 模式 3 和模式 2 的操作完全相同。模式 3 的波特率可变, 由定时器 1 的溢出率决定。

4 种工作模式中, 使用一个以 SBUF 为目标寄存器的指令来启动发送过程。模式 0 的接收过程通过 RI=0 和 REN=1 来启动。如果 REN=1, 接收到起始位后就可启动其它模式的接收过程。

8.15.5 串口控制寄存器 (SCON)

特殊功能寄存器 SCON 是串口控制和状态寄存器, 如表 48 和 49 所示。该寄存器包括: 工作模式选择位、发送和接收的第 9 位数据 (TB8 和 RB8) 以及串口中断位 (TI 和 RI)。

帧错误位 (FE) 用来检测接收到的数据流中的停止位是否丢失。FE 位与 SM0 位共用 SCON 的第 7 位, 该位何时用作何种功能由 PCON 寄存器的 SMOD0 位来决定。如果 SMOD0=0, SCON.7 为 SM0 位。如果 SMOD0=1, SCON.7 为 FE 位。一旦置位, FE 位将保持不变直到通过软件清零。这样就可实现字符串的帧错误检测, 而不需要对单个字符进行逐个检测。

表 48 SCON—串口控制寄存器位分配 (地址: 98H)

可位寻址; 复位值: 00H

位	7	6	5	4	3	2	1	0
符号	FE/SM0	SM1	SM2	REN	TB8	RB8	TI	RI

表 49 SCON—串口控制寄存器位描述 (地址: 98H)

位	符号	描述
7	FE	帧错误。检测到无效的停止位时由 UART 接收器将其置位。要访问该位, PCON 寄存器的 SMOD0 位必须为 1。见下面 SM0 位的描述。
	SM0	和 SM1 一起来定义串口的工作模式。要访问该位, PCON 寄存器的 SMOD0 位必须为 0。见上面 FE 位的描述。
6	SM1	和 SM0 一起来定义串口的工作模式 (见下面的表 50)。

续上表...

位	符号	描述
5	SM2	使能模式 2 和 3 的多机通信特性。模式 2 和 3 中，当 SM2=1 时，如果接收到的第 9 位数据 (RB8) 为 0，不激活 RI 位。模式 1 中，当 SM2=1 时，如果未接收到有效停止位，则不激活 RI。模式 0 中 SM2 为 0。
4	REN	使能串行接收。该位通过软件置位或清零。REN=1 时，允许接收；REN=0 时禁止接收。
3	TB8	模式 2 和 3 发送的第 9 位数据。根据需要由软件置位或清零。
2	RB8	模式 2 和 3 接收到的第 9 位数据。模式 1 中，如果 SM2=0，RB8 是接收到的停止位。模式 0 中不使用 RB8。
1	TI	发送中断标志。模式 0 中，发送完第 8 位数据时由硬件置位；其它模式的串行发送过程中，开始发送停止位时由硬件置位。该位必须通过软件清零。
0	RI	接收中断标志。模式 0 中，接收完第 8 位数据时由硬件置位；其它模式的串行接收过程（例外的情况见 SM2 的描述）中，在接收到停止位的中间时刻由硬件置位。该位必须通过软件清零。

表 50 SM0,SM1 串口模式

SM0,SM1	UART 模式	波特率
00	0: 移位寄存器模式	CPU 时钟/6
01	1: 8 位 UART	可变 (见文中描述)
10	2: 9 位 UART	CPU 时钟/32 或 CPU 时钟/16
11	3: 9 位 UART	可变 (见文中描述)

8.15.6 波特率

模式 0 的波特率固定: 模式 0 波特率 = CPU 时钟 / 6。模式 2 的波特率由特殊功能寄存器 PCON 的 SMOD1 决定。当 SMOD1=0 (复位时的值) 时，波特率为 CPU 时钟频率的 1/32。当 SMOD1=1 时，波特率为 CPU 时钟频率的 1/16。

$$\text{模式 2 波特率} = \frac{1 + \text{SMOD1}}{32} \times \text{CPU 时钟频率} \quad (6)$$

8.15.7 利用定时器 1 产生波特率

当定时器 1 用作波特率发生器时，模式 1 和 3 的波特率由定时器 1 的溢出速率和 SMOD1 的值决定。这时，定时器 1 的中断必须禁止。定时器本身可配置成‘定时器’或‘计数器’和工作在 3 种工作模式中的任何一种。在大多数典型应用中，定时器 1 用作定时器，工作在自动重装模式 (TMOD 的高半字节 = 0010b)。这种情况下的波特率可由下式计算得出：

$$\text{模式 1, 3 波特率} = \frac{\text{CPU 时钟频率} / 192 (\text{SMOD1} = 1 \text{ 时为 } 96)}{256 - (\text{TH1})} \quad (7)$$

表 51 和表 52 列出了各种常用的波特率以及它们如何利用定时器 1 得到。

表 51 波特率、定时器值和 SMOD1=0 时的 CPU 时钟频率

定时器计数值	波特率					
	2400	4800	9600	19.2k	38.4k	57.6k
-1	0.4608	0.9216	*1.8432	*3.6864	*7.3728	*11.0592
-2	0.9216	1.8432	*3.6864	*7.3728	*14.7456	
-3	1.3824	2.7648	5.5296	*11.0592	-	-

续上表...

定时器计数 值	波特率					
	2400	4800	9600	19.2k	38.4k	57.6k
-4	*1.8432	*3.6864	*7.3728	*14.7456	-	-
-5	2.3040	4.6080	9.2160	*18.4320	-	-
-6	2.7648	5.5296	*11.0592	-	-	-
-7	3.2256	6.4512	12.9024	-	-	-
-8	*3.6864	*7.3728	*14.7456	-	-	-
-9	4.1472	8.2944	16.5888	-	-	-
-10	4.6080	9.2160	*18.4320	-	-	-

表 52 波特率、定时器值、SMOD1=1 时的 CPU 时钟频率

定时器值	波特率						
	2400	4800	9600	19.2k	38.4k	57.6k	115.2k
-1	0.2304	0.4608	0.9216	*1.8432	*3.6864	5.5296	*11.0592
-2	0.4608	0.9216	*1.8432	*3.6864	*7.3728	*11.0592	-
-3	0.6912	1.3824	2.7648	5.5296	*11.0592	16.5888	-
-4	0.9216	*1.8432	*3.6864	*7.3728	*14.7456	-	-
-5	1.1520	2.3040	4.6080	9.2160	*18.4320	-	-
-6	1.3824	2.7648	5.5296	*11.0592	-	-	-
-7	1.6128	3.2256	6.4512	12.9024	-	-	-
-8	*1.8432	*3.6864	*7.3728	*14.7456	-	-	-
-9	2.0736	4.1472	8.2944	16.5888	-	-	-
-10	2.3040	4.6080	9.2160	*18.4320	-	-	-
-11	2.5344	5.0688	10.1376	-	-	-	-
-12	2.7648	5.5296	*11.0592	-	-	-	-
-13	2.9952	5.9904	11.9808	-	-	-	-
-14	3.2256	6.4512	12.9024	-	-	-	-
-15	3.4560	6.9120	13.8240	-	-	-	-
-16	*3.6864	*7.3728	*14.7456	-	-	-	-
-17	3.9168	7.8336	15.6672	-	-	-	-
-18	4.1472	8.2944	16.5888	-	-	-	-
-19	4.3776	8.7552	17.5104	-	-	-	-
-20	4.6080	9.2160	*18.4320	-	-	-	-
-21	4.8384	9.6768	19.3536	-	-	-	-

[1] 表 51 和表 52 适用于 UART 的模式 1 和 3 (波特率可变模式), 给出了 2400~115.2 范围内的标准波特率对应的 CPU 时钟频率 (以 MHz 为单位)。

[2] 表 51 给出的是 PCON 寄存器的 SMOD1 位为 0 (复位后的默认值) 时的定时器设置和 CPU 时钟频率, 而表 52 的值是在 SMOD1 位=1 的条件下得到的。

[3] 表中列出的 20MHz 以内的 CPU 时钟频率产生的波特率范围: 9600~115.2k。其它 CPU 时钟频率值只能得到更低的波特率 (表中未列出)。

[4] 表中各行带*标识的量表明它是可以直接得到的标准的晶振和陶瓷振荡器频率, 不需要特别定制。

8.15.8 更多关于 UART 模式 0 的信息

串行数据从 RxD 进出。TxD 输出移位时钟。发送或接收 8 位数据 (LSB 在前)。波特率固定为 CPU 时钟频率/6。图 23 所示为模式 0 时串口的功能简图和时序。

发送过程是由执行一条以 SBUF 为目的寄存器的指令启动的。在 S6P2 时刻出现的‘写 SBUF’信号把 1 装入发送移位寄存器的第 9 位, 同时通知 Tx 控制模块开始发送数据。内部时序中, ‘写 SBUF’和 SEND 激活之间相差一个完整的机器周期。

由 SEND 来使能移位寄存器的内容从 P1.1 输出和 P1.0 的移位时钟输出功能。移位时钟在每个机器周期的 S3、S4 和 S5 内为低电平, 在 S6、S1 和 S2 内为高电平。在 SEND 有效的每个机器周期的 S6P2 时刻, 发送移位寄存器的内容向右移一位。

数据向右移出, 左边补 0, 当数据字节的 MSB 位移到移位寄存器的输出位置时, 其左边是装入“1”的第 9 位, 再左的内容均为 0。此时通知 TX 控制器作最后一次移位, 然后禁能 SEND 端并置位 TI。这都发生在‘写 SBUF’后第 10 个机器周期的 S1P1 时刻。接收过程通过置位 REN 和清零 R1 来启动。在下个机器周期的 S6P2 时刻, RX 控制单元向接收移位寄存器写入 11111110 并在下个时钟信号到来时激活 RECEIVE。

由 RECEIVE 来使能 P1.0 的移位时钟输出功能。移位时钟在每个机器周期的 S3P1 和 S6P1 时刻跳变。在 RECEIVE 有效的每一机器周期的 S6P2 时刻, 接收移位寄存器内容向左移一位。从右移位进来的值是该机器周期 S5P2 时从 P1.1 脚上采样得来的。

数据从右边移入时, 左边移出为“1”。当最初写入最右端的“0”移至移位寄存器的最左端时, 通知 RX 控制器做最后一次移位后装入 SBUF。在写入 SCON 清除 R1 后第 10 个机器周期, RECEIVE 端被清除且置位 RI。

8.15.9 更多关于 UART 模式 1 的信息

TxD 脚发送, RxD 脚接收, 每次传输 10 位: 1 个起始位 (0), 8 个数据位 (LSB 在前) 和 1 个停止位 (1)。接收时, 停止位存入 SCON 的 RB8。该模式下, P87LPC778 的波特率由定时器 1 的溢出速率决定。图 24 所示为模式 1 时串口的功能简图和发送接收时序。

发送过程是由执行一条以 SBUF 为目的寄存器的指令启动的。“写 SBUF”信号还把 1 装入发送移位寄存器的第 9 位, 同时通知 Tx 控制单元进行发送。实际上发送过程开始于 16 分频计数器下次翻转后的那个机器周期的 S1P1 时刻。(这样, 每位的发送时序与 16 分频计数器同步, 而并不与“写 SBUF”信号同步)。

发送以激活 SEND 端开始, 向 TxD 发送一起始位。一位 (时间) 以后 DATA 端有效, 使输出移位寄存器中数据得以送至 TxD。再过一位, 产生第一个移位脉冲。

数据向右移出, 左边补 0, 当数据字节的 MSB 位移到移位寄存器的输出位置时, 其左边是装入“1”的第 9 位, 再左的内容均为 0。此时通知 TX 控制器作最后一次移位, 然后禁能 SEND 端并置位 TI。这都发生在“写 SBUF”后 16 分频计数器的第 10 次翻转时。

接收在 RxD 端检测到负跳变时启动, 为此 MCU 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, 同时将 1FFH 写入输入移位寄存器。复位 16 分频计数器确保计时器翻转时位与输入数据位时间同步。

计数器的 16 个状态将每个位时间分为 16 份。在第 7、8、9 状态时, 位检测器对 RxD 端的值采样。三个采样值中取多数 (至少 2 个) 作为读入值, 这样可以抑制噪声。如果所接收的第一位不为 0, 接收电路被复位, 等待另一个负跳变的到来。这用来防止错误的起始位。如果起始位有效, 则被移入输入移位寄存器, 并开始接收这一帧中的其它位。

当数据位逐一由右边移入时, “1”从左边被移出。当起始位移到最左边时 (模式 1 为 9 位寄存器), 通知接收控制器进行最后一次移位, 装载 SBUF 及 RB8 并置位 RI。仅当最后一位移位脉冲产生时同时满足下述 2 个条件: ①RI=0, ②SM2=0 或接收到的停止位=1, 才会装载 SBUF 和 RB8, 并且置位 RI。

上述两个条件任一不满足, 所接收到的数据帧就会丢失, 不再恢复。两者都满足时, 停止位就进入

RB8, 8 位数据进入 SBUF, RI=1。这时, 无论上述条件满足与否, 接收控制单元都会重新等待 RxD 的负跳变。

8.15.10 更多关于 UART 模式 2 和模式 3 的信息

TxD 脚发送, RxD 脚接收, 每次传输 11 位: 1 个起始位 (0), 8 个数据位 (LSB 在前), 1 个可编程第 9 位数据和 1 个停止位 (1)。发送时, 第 9 位数据 (TB8) 可置为 0 或 1。接收时, 第 9 位存入 SCON 的 RB8。模式 2 时波特率可编程选为 CPU 时钟频率的 1/16 或 1/32。模式 3 时可由定时器 1 获取可变的波特率。

图 25 和 26 所示为模式 2、3 时串行口的功能简图。接收部分与模式 1 相同。发送部分仅在发送移位寄存器第 9 位时和模式 1 有所不同。

发送过程是由执行一条以 SBUF 为目的寄存器的指令启动的。“写 SBUF”同时将 TB8 装入发送移位寄存器的第 9 位, 并通知发送控制器进行一次发送。发送过程在 16 分频计数器下一次翻转后机器周期的 S1P1 时刻启动。(这样, 每位的发送时序与 16 分频计数器同步, 并不与“写 SBUF”信号同步)。

发送以激活 SEND 端开始, 向 TxD 发送一个起始位。一位 (时间) 以后 DATA 端有效, 使输出移位寄存器中数据得以送至 TxD。再过一位, 产生第一个移位脉冲。第一个移位时钟将“1” (停止位) 送入移位寄存器的第 9 位, 此后每次移位只把 0 送入第 9 位, 所以当数据位向右移出时, “0”从左边移入。当 TB8 移至输出位置上时, 它左边就是停止位, 其余位均为零。此时将通知发送控制器作最后一次移位, 然后禁用 SEND 并置位 TI。这都发生在“写 SBUF”后第 11 次计数器翻转时。

接收在 RxD 端检测到负跳变时启动, 为此 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, 同时将 1FFH 写入输入移位寄存器。

在每一位的第 7、8、9 状态时, 位检测器对 RxD 端值进行采样。三个采样值取多数 (至少 2 次) 为输入值。如果接收的第一位不为 0, 接收电路复位, 等待下一个负跳变的到来。如果起始位有效, 则被移入输入移位寄存器, 并开始接收这一帧中的其它位。

数据位从右边移入, “1”从左边移出。当起始位移至寄存器 (模式 2 和 3 为 9 位寄存器) 的最左端时, 通知接收控制器进行最后一次移位, 装载 SBUF 及 RB8 并置位 RI。

仅当产生最后一位移位脉冲时同时满足下列 2 个条件: ①RI=0, ②SM2=0 或接收到的第 9 位数据为 1 时, 才装载 SBUF 和 RB8, 并置位 RI。

上述两个条件任一不满足, 接收到的数据帧丢失, 不再恢复, RI 仍为 0。当两者都满足时, 第 9 位数据位就装入 RB8, 前 8 位数据则装入 SBUF, 一位时间后, 无论上述条件满足与否, 单元都会重新等待 RxD 端负跳变的到来。

8.15.11 多机通信

UART 模式 2 及模式 3 有一个特别应用—多机通信。在模式 2 和模式 3 下, 发送或接收 9 位数据。接收时第 9 位存入 RB8。UART 可编程为: 接收到停止位时, 仅当 RB8=1 时串口中断才有效。可通过置位 SCON 内 SM2 位来选择这一特性。下述为多机系统使用这一特性的一种方法。

当主机要向多个从机之一发送数据块时, 首先发送出目标从机的地址。地址字节和数据字节的第 9 位不同, 地址字节的第 9 位为 1, 而数据字节为 0, SM2=1 时, 数据字节不会使各从机中断, 而地址字节可以中断所有从机, 因此每个从机都要检查接收到的数据来判断是否被寻址。被寻址的从机清除 SM2 位以准备接收后面的数据。未被寻址的从机的 SM2 位仍为 1, 不理睬后面的数据, 继续各自的工作。

模式 0 时 SM2 无效, 模式 1 时 SM2 用来检测停止位是否有效, 尽管这可通过帧错误标志来实现。在模式 1 的接收过程中, 如果 SM2=1, 那么只有接收到有效的结束位才可产生接收中断。

8.15.12 地址自动识别

地址自动识别是使 UART 通过硬件比较从串行数据位中识别出某一特定地址。这样就不必花费大量时

间去处理由串行口输入的串行地址。该特性通过置位 SCON 的 SM2 位来使能。在 9 位 UART 模式（模式 2 和模式 3）中，如果接收到的内容包含“给定”地址或“广播”地址，接收中断标志（RI）将自动置位。在 9 位模式下第 9 个信息位必须为 1，用来指示接收到的是地址而非数据。

使用地址自动识别特性时，主机通过调用给定从机地址选择与一个（或多个）从机通信。使用广播地址时，所有从机都被联系。在此使用了两个特殊功能寄存器，SADDR 表示从机地址，SADEN 表示地址屏蔽。SADEN 用于定义 SADDR 内哪几位需使用而哪几位不予考虑。SADEN 屏蔽可以与 SADDR 逻辑“与”得出给定地址，用来对每一从机进行寻址。下面举例说明该方法的通用性：

表 53 从机 0/1 举例

例 1	例 2
从机 0 SADDR=1100 0000 SADEN= <u>1111 1101</u> 给定地址=1100 00X0	从机 1 SADDR=1100 0000 SADEN= <u>1111 1110</u> 给定地址=1100 000X

例中 SADDR 相同，而 SADEN 不同以区分两个从机。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 的位 1 必须为 0，从机 0 只能取独有的地址 1100 0010 以区别。由于从机 0 的位 0 必须为 1，从机 1 只能取独有的地址 1100 0001 以区别。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：

表 54 从机 0/1/2 举例

例 1	例 2	例 3
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 可选通从机 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效。大多数情况下，无效位被设定为 1，这样，“广播”地址为 FFH。复位时 SADDR 和 SADEN 均为 00H，此时给定地址与广播地址均无效。这样有效地禁止了自动寻址模式，微处理器只能使用标准的 UART 通讯模式。

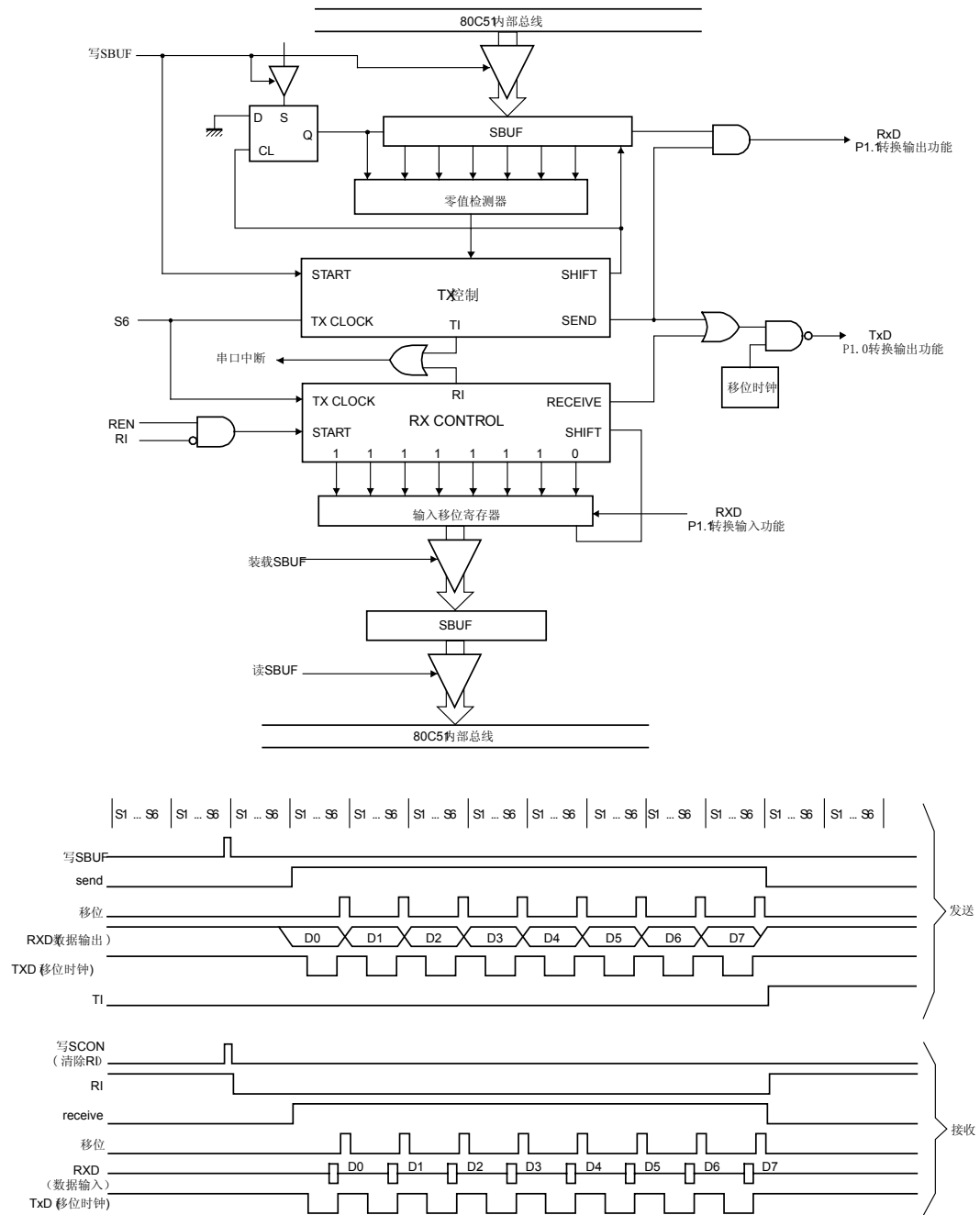


图 23 串口模式 0

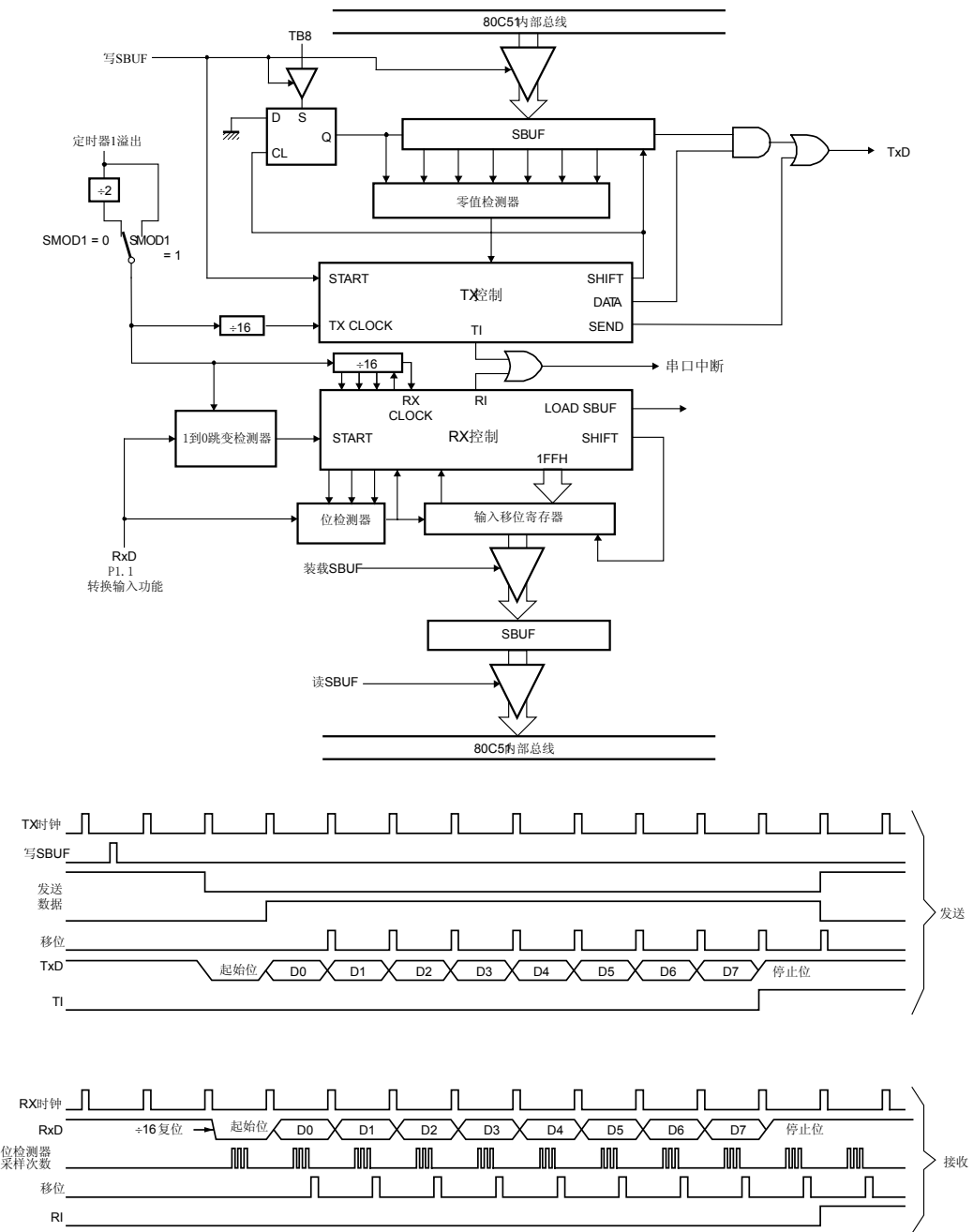


图 24 串口模式 1

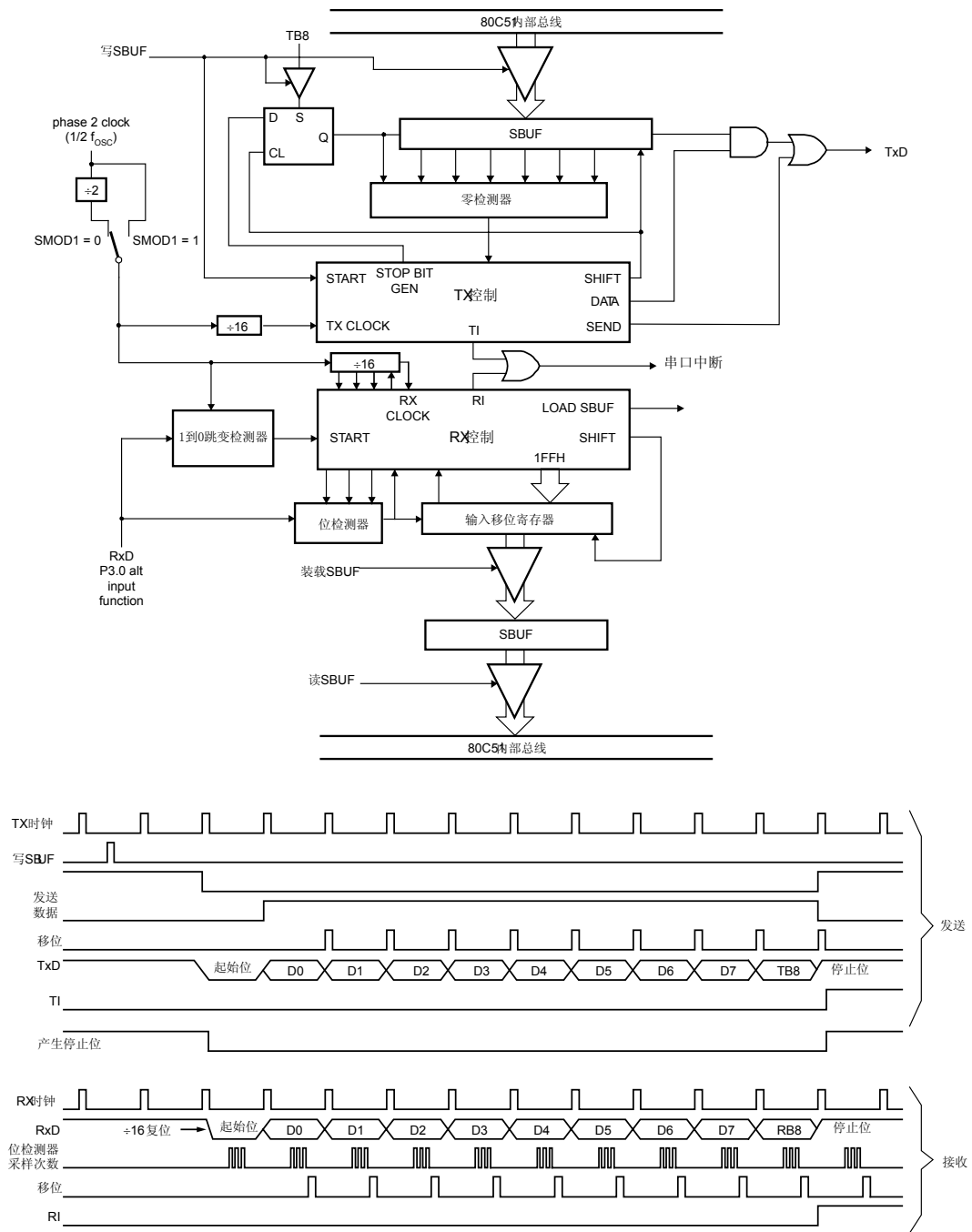


图 25 串口模式 2

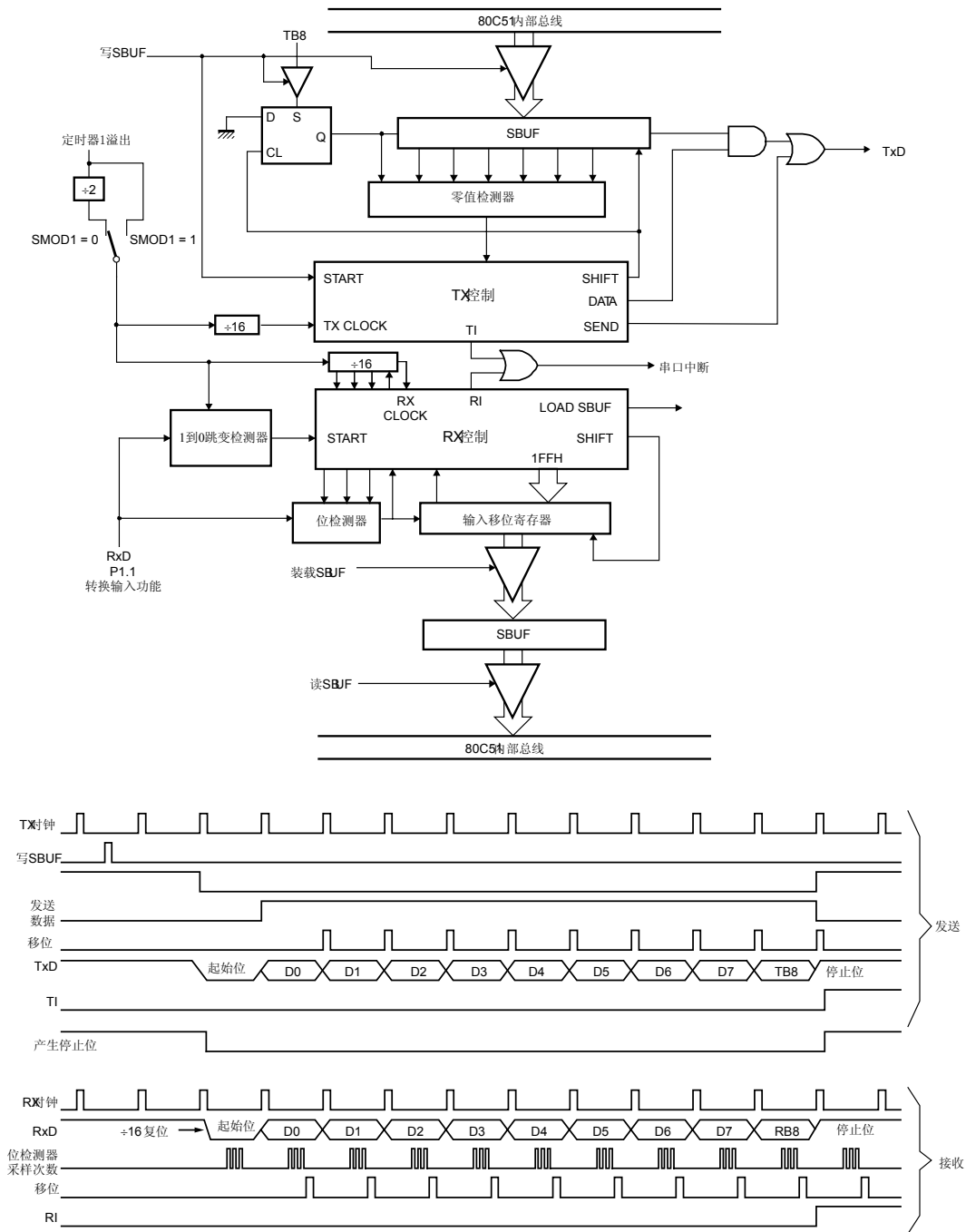


图 26 串口模式 3

8.16 看门狗定时器

当看门狗定时器通过 WDTE 配置位使能时，它工作在完全独立的片内振荡器下，保证了最大程度的可靠性。当看门狗特性使能时，定时器必须定时地通过软件喂狗来防止其溢出时复位 CPU，定时器不能关闭。当看门狗定时器禁能时（通过设置 UCFG1 配置寄存器的 WDTE 位），它可用作一个间隔定时器，并可产生中断。看门狗定时器如图 27 所示。

看门狗的溢出时间有 8 种选择，标称时间范围从 16ms~2.1s。独立的看门狗 RC 振荡器的频率误差为 ±37%。溢出时间选择和其它控制位见表 55 和 56。当看门狗功能使能时，芯片初始化时执行一次 WDCON

寄存器写操作来设置看门狗溢出值。建议初始化 WDCON 寄存器时，先执行喂狗时序，再写 WDCON 寄存器的配置位 WDS[2:0]。使用这种方法，看门狗初始化可在启动后的 10ms 内进行，而不需要在初始化完成之前等待看门狗溢出。

由于看门狗定时器振荡器是完全独立于 CPU 使用的外部振荡电路的内置振荡器，它实质上是用作振荡器失效检测功能。如果看门狗特性使能和 CPU 振荡器由于某种原因失效，看门狗定时器溢出并复位 CPU。

当看门狗功能使能时，定时器会因为其它复位源造成的芯片复位而暂时无效，例如上电复位、掉电复位或外部复位等。

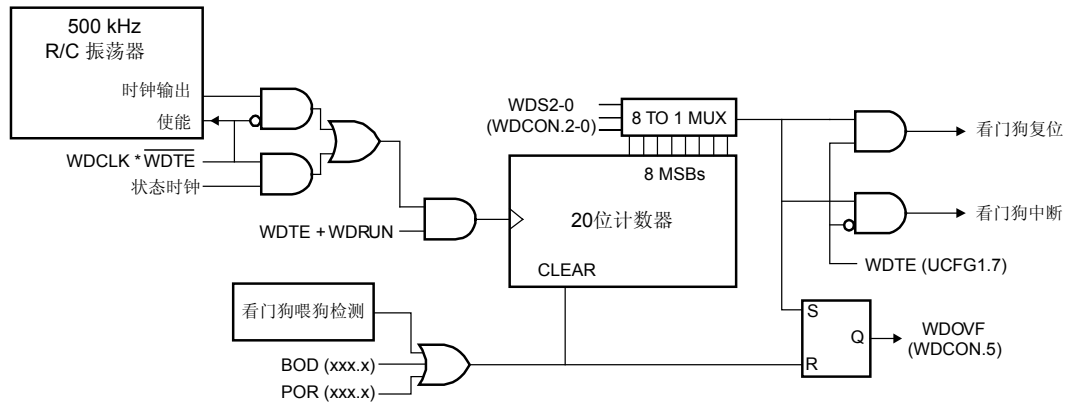


图 27 看门狗定时器的方框图

8.16.1 看门狗喂狗时序

如果看门狗定时器正在运行，溢出前必须执行喂狗时序来防止产生芯片复位。看门狗喂狗时序就是先后向 WDRST 寄存器写入 1Eh 和 E1h。下面是一个操作实例：

WDFeed:

```

mov   WDRST,#1eh   ; 看门狗喂狗时序第一步
mov   WDRST,#0e1h  ; 看门狗喂狗时序第二步
    
```

这两条写 WDRST 的指令不必连续。看门狗定时器不会立即响应错误的喂狗时序，如果正确的喂狗时序未在看门狗定时器溢出前出现，看门狗定时器溢出。

芯片复位后，用户程序只有有限的时间执行喂狗时序或改变溢出时间。如果应用中使用较低的 CPU 时钟频率，则在看门狗溢出前能执行的指令数很少。

8.16.2 看门狗复位

如果产生看门狗复位，内部复位的有效期约为 1us。如果 CPU 时钟正在运行，看门狗复位后立即开始执行程序。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运行稳定后恢复程序的执行。

表 55 WDCON—看门狗定时器控制寄存器位分配（地址：A7H）

不可位寻址；复位值：30H（看门狗复位）、10H（其它复位源，看门狗通过 WDTE 配置位使能）、00H（其它复位源，看门狗通过 WDTE 配置位禁能）

位	7	6	5	4	3	2	1	0
符号	-	-	WDOVF	WDRUN	WDCLK	WDS2	WDS1	WDS0

表 56 WDCON—看门狗定时器控制寄存器位描述（地址：A7H）

位	符号	描述
7,6	-	保留为将来之用。用户程序不能将其置 1。
5	WDVOF	看门狗定时器溢出标志。看门狗复位或定时器溢出时置位。喂狗时清零。
4	WDRUN	看门狗运行控制。WDRUN=1 时启动看门狗定时器；WDRUN=0 时停止看门狗定时器。当 WDTE 配置位=1 时该位被强制置位（看门狗运行）。
3	WDCLK	看门狗时钟选择。WDCLK=1 时看门狗定时器时钟频率为 CPU 时钟的 1/6；WDCLK=0 时看门狗定时器时钟为看门狗 RC 振荡器。当 WDTE 配置位=1 时该位被强制清零（使用看门狗 RC 振荡器）。
2~0	WDS[2:0]	看门狗溢出率选择。

表 57 看门狗溢出时钟选择

WDS[2:0]	溢出时钟	最小时间	标称时间	最大时间
000	8,192	10ms	16ms	23ms
001	16,384	20ms	32ms	45ms
010	32,768	41ms	65ms	90ms
011	65,536	82ms	131ms	180ms
100	131,072	165ms	262ms	360ms
101	262,144	330ms	524ms	719ms
110	524,288	660ms	1.05s	1.44s
111	1,048,576	1.3s	2.1s	2.9s

8.17 附加特性

AUXR1 寄存器包含几个与芯片功能相关的特别控制位。见表 58 和 59。

表 58 AUXR1—AUXR1 寄存器位分配（地址：A2H）

不可位寻址；复位值：00H

位	7	6	5	4	3	2	1	0
符号	KBF	BOD	BOI	LPEP	SRST	0	-	DPS

表 59 ARXR1—AUXR1 寄存器位描述（地址：A2H）

位	符号	描述
7	KBF	键盘中断标志。任意一个使能用作键盘中断功能的管脚为低电平时置位。该位必须通过软件清零。
6	BOD	掉电禁能。该位置位时，关闭掉电检测来以节省功耗。详见第 8.11 节“电源检测功能”。
5	BOI	掉电中断。置位时禁止掉电检测产生芯片复位，但允许掉电检测用作中断。详见第 8.11 节“电源监控功能”。
4	LPEP	低功耗 EPROM 控制位。用在低电压系统中以节省功耗。该位由软件置位，只能通过上电或掉电复位清零。详见第 8.12 节“节电模式”。
3	SRST	软件复位。该位通过软件置位时，象硬件复位一样复位 P87LPC778。
2	0	该位为硬件 0。允许通过 AUXR1 加 1 使 DPS 位翻转，寄存器的其它位不受影响。
1	-	保留为将来之用。用户程序不要将其置 1。
0	DPS	数据指针选择。由用户程序选择 2 个数据指针中的一个使用。详见文中描述。

8.17.1 软件复位

AUXR1 的 SRST 位允许通过软件将处理器完全复位，就象发生外部复位或看门狗复位一样。如果一个位 3 为 ‘1’ 的值写入 AUXR1，所有 SFR 被赋初值，程序从地址 0000 开始执行。因此，写 AUXR1 要特别小心以防产生意外复位。

8.17.2 双数据指针

双数据指针 (DPTR) 增加了一种处理器指向特定指令地址的方法。AUXR1 寄存器的 DPS 位选择两个数据指针中的一个。未选中的 DPTR 软件不予接受，除非 DPS 位变化。

受数据指针选择影响的指令有：

- INC DPTR: 数据指针加 1
- JMP @A+DPTR: 跳转到相对 DPTR 值的间接地址
- MOV DPTR,#data16: 将一个 16 位的常数写入数据指针
- MOVC A,@A+DPTR: 将 DPTR 相对地址的程序字节移入累加器
- MOVX A,@DPTR: 将 DPTR 指向的数据存储器单元的内容移入累加器
- MOVX @DPTR,A: 将累加器的内容移入 DPTR 指向的数据存储器

任何读或修改 DPH 和 DPL (DPTR 的高字节和低字节) 寄存器的指令都受 DPS 位的影响。由于 P87LPC778 没有外部数据总线，因此 MOVX 指令的使用受到限制，但是它们可以用来访问 EPROM 配置信息 (见第 8.18 节 “EPROM 特性”)。

AUXR1 的位 2 永远为 0。因此，只需通过 AUXR1 加 1 就可改变 DPS 位的值 (切换数据指针)，而寄存器的其它位不受影响。

8.18 EPROM 特性

对 P87LPC778 EPROM 的编程通过串行编程的方法来实现。进入编程模式后，命令、地址和数据通过两个管脚传输。串行编程可以很方便地实现应用板上 P87LPC778 的在电路编程。

P87LPC778 包含可读的 3 个标识字节，EPROM 编程系统利用它们来识别器件。标识字节指明 P87LPC778 是由 PHILIPS 生产的。标识字节位于 FC30h、FC31h 和 FC60h 地址，用户程序可利用 DPTR 寄存器寻址的 MOVC 指令将其读出。

利用 MOVC 指令也可访问 FCE0h~FCFFh 的特定用户代码区。‘用户代码’区的编程方法与主代码 EPROM 相同，用来保存序列号、生产日期或其它应用信息。

8.18.1 系统配置字节

P87LPC778 的大量用户配置特性必须在上电时定义，开始执行程序后就不可进行设置了。这些特性通过编程 2 个 EPROM 字节来配置，与编程 EPROM 程序空间的方法相同。两个配置字节 UCFG1 和 UCFG2 的内容见表 60、61、63 和 64。两个配置字节的值可通过 MOVX 指令读出，其地址如表中所示。

表 60 UCFG1—EPROM 系统配置字节 1 寄存器位分配 (地址: FD00H)

未编程的值: FFH

位	7	6	5	4	3	2	1	0
符号	WDTE	RPD	PRHI	BOV	CLKR	FOSC2	FOSC1	FOSC0

表 61 UCFG1—EPROM 系统配置字节 1 寄存器位描述 (地址: FD00H)

位	符号	描述
7	WDTE	看门狗定时器使能。该位编程为 0 时，看门狗定时器禁能，但定时器仍用来产生中断。

续上表...

位	符号	描述
6	RPD	复位管脚禁能。该位编程为 0 时，禁能 P1.5 的复位功能，使其用作仅为输入的端口。
5	PRHI	口复位高。该位为 1 时，口复位为高电平。该位为 0 时，口复位为低电平。
4	BOV	掉电电压选择。该位为 1 时，掉电检测电压为 2.5V。该位为 0 时，掉电检测电压为 3.8V。见第 8.11 节“电源监控功能”的描述。
3	CLKR	时钟频率选择。该位为 0 时，CPU 时钟被 2 分频。使得 P87LPC778 同标准 80C51 器件一样，一个机器周期包含 12 个 CPU 时钟。为了向下兼容，该分频值也可用于外围时序。
2~0	FOSC[2:0]	CPU 振荡器类型选择。见第 8.10 节“振荡器”。下表中未列出的组合供未来使用。

表 62 FOSC2~FOSC0 振荡器配置

FOSC[2:0]	振荡器配置
111	X1 脚输入的外部时钟（默认）
011	内部 RC 振荡器，6MHz
010	低频晶振，20kHz~100kHz
001	中频晶振或谐振器，100kHz~4MHz
000	高频晶振或谐振器，4MHz~20MHz

表 63 UCFG2—EPROM 系统配置字节 2 寄存器位分配（地址：FD01H）

未编程的值：FFH

位	7	6	5	4	3	2	1	0
符号	SB2	SB1	-	-	-	-	-	-

表 64 UCFG2—EPROM 系统配置字节 2 寄存器位描述（地址：FD01H）

位	符号	描述
7,6	SB2,SB1	EPROM 保密位。见表 65。
5~0	-	保留为将来之用。

8.18.2 保密位

当两个保密位都未编程时，可对 EPROM 的代码进行校验。当只有保密位 1 被编程时，禁止编程 EPROM。这时，只有保密位 2 仍可被编程。当两个保密位都被编程时，禁止校验 EPROM。

表 65 EPROM 安全位

SB2	SB1	保密描述
1	1	两个保密位都未被编程。程序不加密。可编程和校验 EPROM。
1	0	只有保密位 1 被编程。禁止编程 EPROM。保密位 2 仍可被编程。
0	1	只有保密位 2 被编程。不支持该组合。
0	0	两个保密位都被编程。禁止校验和编程 EPROM。

9. 极限参数

表 66 极限参数

遵循最大绝对额定系统规范 (IEC60134)。

符号	参数	条件	最小	最大	单位
T _{amb(bias)}	工作环境温度		-55	+85	°C
T _{stg}	储存温度		-65	+150	°C
V _{RST}	$\overline{\text{RST}}$ /V _{pp} 到 V _{SS} 的电压		-	+11.0	V
V _n	任意脚对 V _{SS} 电压		-0.5	V _{DD} +0.5	V
I _{OL(I/O)}	每个 I/O 口的低电平输出电流		-	50	mA
I _{OH(I/O)}	每个 I/O 口的高电平输出电流		-	-50	mA
I _{OL(tot)(max)}	输出最大总 I _{OL}		-	200	mA
I _{OH(tot)(max)}	输出最大总 I _{OH}		-	-200	mA
P _{tot(pack)}	封装的功率损耗	基于封装的热传递, 并非器件的功耗		1.5	W

[1] 器件在超过表 66 “极限参数” 工作可能会造成永久性的损坏。这里只列出了一些极限值, 并未涉及在这些极限值或其它条件下 (除在表 67 “DC 电气特性” 和表 69 “AC 电气特性” 中所描述的之外) 的器件功能操作。

[2] 本产品带有保护器件内部的电路设计, 以避免超负荷的损坏性影响。但是建议不要在超过极限值的情况下工作。

[3] 参数在操作温度范围内是有效的, 除非另有规定。所有的电压都是相对 V_{SS} 而言的, 除非另有说明。

10. 静态特性

表 67 DC 电气特性

除非特别说明, V_{DD}=2.7V~5.5V T_{amb}=-40°C~+85°C, 工业级

符号	参数	条件	最小	典型 ¹	最大	单位
I _{DD}	电源电流, 工作模式	5.0V; 20MHz ¹⁰	-	15	25	mA
		3.0V; 10MHz ¹⁰	-	4	7	mA
I _{ID}	电源电流, 空闲模式	5.0V; 20MHz ¹⁰	-	6	10	mA
		3.0V; 10MHz ¹⁰	-	2	4	mA
I _{PD}	电源电流, 掉电模式	5.0V ¹⁰	-	1	10	μA
		3.0V ¹⁰	-	1	5	μA
V _{RAM}	RAM 保持电压		1.5	-	-	V
V _{IL}	输入低电压(TTL 输入)	4.5V < V _{DD} < 5.5V	-0.5	-	0.2V _{DD} -0.1	V
V _{IL1}	负门槛电压(施密特输入)		-0.5	-	0.3 V _{DD}	V
V _{IH}	输入高电压(TTL 输入)		0.2V _{DD} +0.9	-	V _{DD} +0.5	V
V _{IH1}	正门槛电压(施密特输入)		0.7 V _{DD}	-	V _{DD} +0.5	V
V _{hys}	滞后电压		-	0.2V _{DD}	-	V
V _{OL}	输出低电压, 所有端口 ^{4,8}	I _{OL} =3.2mA, V _{DD} =4.5V	-	-	0.4	V
V _{OL1}	输出低电压, 所有端口 ^{4,8}	I _{OL} =20mA, V _{DD} =4.5V	-	-	1.0	V
V _{OH}	输出高电压, 所有端口 ²	I _{OH} =-30μA, V _{DD} =4.5V	V _{DD} -0.7	-	-	V
V _{OH1}	输出高电压, 所有端口 ³	I _{OH} =-1.0mA, V _{DD} =4.5V	V _{DD} -0.7	-	-	V
C _{IO}	输入/输出管脚电容 ⁹		-	-	15	pF

续上表...

符号	参数	条件	最小	典型 ¹	最大	单位
I_{IL}	逻辑 0 输入电流, 所有端口 ⁷	$V_{IN}=0.4V$	-	-	-50	μF
I_{LI}	输入漏电流, 所有端口 ⁶	$V_{IN}=V_{IL}$ 或 V_{IH}	-	-	± 2	μA
I_{TL}	逻辑 1 到 0 跳变电流, 所有端口 ^{2,5}	$V_{IN}=2.0V$ at $V_{DD}=5.5V$	-150		-650	μA
R_{RST}	内部复位上拉电阻		40	-	225	$K \Omega$
V_{BOLOW}	BOV=1 时掉电电压 ¹¹		2.35	-	2.69	V
V_{BOHI}	BOV=0 时掉电电压 ¹²		3.45	3.8	3.99	V
V_{REF}	参考电压		1.11	1.26	1.41	V

[1] 不能保证得到典型的标称值。表中所列值在室温, 电压 5V 下测得。

[2] 带弱上拉的准双向口模式 (用于所有带上拉的口)。请勿用于开漏管脚。

[3] 推挽模式的口。请勿用于开漏管脚。

[4] 除了高阻模式外的所有输出模式。

[5] 准双向口模式和外部驱动 1 变 0 时的管脚的跳变电流。当 V_{IN} 大约为 2V 时, 该电流最大。

[6] 在高阻模式下测得。该参数在低温下未作测试但可得到保证。

[7] 在准双向模式下测得。

[8] 在稳态 (非瞬态) 条件下, I_{OL} 必须受到以下限制:

- a) 每个管脚最大 I_{OL} : 20mA
- b) 所有输出管脚的最大 I_{OL} : 80mA
- c) 所有管脚的最大 I_{OH} : 5mA

如果 I_{OL} 超过测试条件, V_{OL} 可能超过相应的规格。管脚不能保证吸收大于上表所列的电流。

[9] 管脚电容由其特性得到, 但未作测试。

[10] I_{DD} , I_{ID} 和 I_{PD} 的规格在以下条件下测得: 使用外部时钟源, 关闭比较器、掉电检测和看门狗定时器。
 $V_{DD}=3V$, $LPEP=1$ 。

[11] 器件可在 $V_{DD} \geq 2.7V$ 时启动操作。在 $f_{OSC} \leq 10MHz$ 时, 在处于掉电电压点时可保证指令的继续正确运行。在 $V_{DD} < 2.7V$ 时则不能保证初始上电操作。

[12] 器件可在 $V_{DD} \geq 4.0V$ 时启动操作。在 $f_{OSC} \leq 20MHz$ 时, 在处于掉电电压点时可保证指令的继续正确运行。在 $V_{DD} < 4.0$ 和 $f_{OSC} > 10MHz$ 时则不能保证初始上电操作。

表 68 A/D 转换器 DC 电气特性

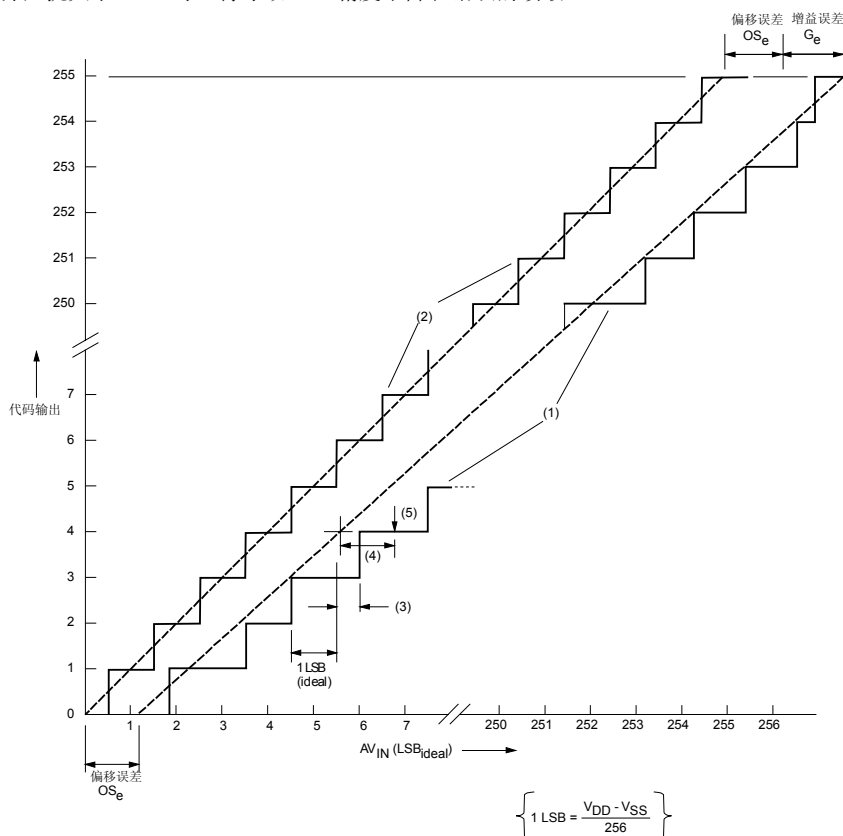
除非特别说明, $V_{DD}=2.7V \sim 5.5V$ $T_{amb}=-40^{\circ}C \sim +85^{\circ}C$, 工业级

符号	参数	条件	最小	最大	单位
AV_{IN}	模拟输入电压		$V_{SS}-0.2$	$V_{DD}+0.2$	V
C_{IA}	模拟输入电容			15	pF
DLe	微分非线性 ^{1,2,3}			± 1	LSB
ILe	积分非线性 ^{1,4}			± 1	LSB
OSe	偏移误差 ^{1,5}			± 1	LSB
Ge	增益误差 ^{1,6}			± 0.4	%
Ae	绝对电压误差 ^{1,7}			± 1	LSB
M_{CTC}	通道间匹配			± 1	LSB
Ct	输入口间窜扰 ⁸	0~100kHz		-60	dB

续上表...

符号	参数	条件	最小	最大	单位
-	输入转换速率			100	V/ms
-	输入阻抗			10	kΩ

- [1] 条件: $V_{SS}=0V$; $V_{DD}=5.12V$ 。
- [2] A/D 是单调的, 无丢失的代码。
- [3] 微分非线性(DLe)是指实际的阶宽和理想的阶宽之间的差。见图 28。
- [4] 积分非线性(ILe)是指在调整了增益和偏移误差之后实际台阶的中心与理想的转换曲线的峰值差。见图 28。
- [5] 偏移误差(OSe)是指符合实际转换曲线的直线(在除去增益误差后)与符合理想转换曲线的直线之间的绝对差。见图 28。
- [6] 增益误差(Ge)是指符合实际转换曲线的直线(在除去偏移误差后)与符合理想转换曲线的直线之间的相对差。见图 28。
- [7] 绝对电压误差(Ae)是指未校正 ADC 的实际转换曲线和理想转换曲线台阶中心之间的最大差。
- [8] 当模拟和数字信号同时输入 A/D 管脚时应当考虑。
- [9] 当输入电压变化比转换速率快时, 将导致错误的读取。
- [10] 信号源阻抗大于 10kΩ时, 将导致 A/D 精度下降和错误的读取。



- (1)实际转换曲线
- (2)理想转换曲线
- (3)微分非线性误差(DLe)
- (4)积分非线性误差(ILe)
- (5)实际转换曲线的台阶中心

图 28 A/D 转换特性

11. 动态特性

表 69 AC 电气特性

除非特别说明, $V_{DD}=2.7V\sim 5.5V$ $V_{SS}=0V$ 。^{1,2,3} $T_{amb}=-40^{\circ}C\sim +85^{\circ}C$, 工业级

符号	图例	参数	最小	最大	单位	
外部时钟						
f_c	30	振荡频率($V_{DD}=4.0V\sim 5.5V$)	0	20	MHz	
f_c	30	振荡频率($V_{DD}=2.7V\sim 5.5V$)	0	10	MHz	
t_c	30	时钟周期和 CPU 时序	$1/f_c$	-	ns	
t_{CLCX}	30	时钟低电平时间 ^[1]	$f_{osc}=20MHz$	20	-	ns
t_{CLCX}	30		$f_{osc}=10MHz$	40	-	ns
t_{CHCX}	30	时钟高电平时间 ^[1]	$f_{osc}=20MHz$	20	-	ns
t_{CHCX}	30		$f_{osc}=10MHz$	40	-	ns
内部 RC 振荡器						
f_{CCAL}		片内振荡器校准 ^[2]	$f_{RCOSC}=6MHz$	-1	+1	%
f_{CTOL}		片内振荡器容许误差 ^{[3][4]}	$f_{RCOSC}=6MHz$	-2.5	+2.5	%
f_{CTOLI}		片内振荡器容许误差 ^[3]	$f_{RCOSC}=6MHz$	-25	+25	%
移位寄存器						
t_{XLXL}	29	串行口时钟周期	$6t_c$	-	ns	
t_{QVXH}	29	输出数据建立到时钟上升沿	$5t_c-133$	-	ns	
t_{XHGX}	29	输出数据在时钟上升沿后保持	$1t_c-80$	-	ns	
t_{XHDX}	29	输入数据建立到时钟上升沿	-	$5t_c-133$	ns	
t_{XHDX}	29	输入数据在时钟上升沿后保持	0	-	ns	

[1] 仅应用于外部时钟源, 而不是连接到 X1 和 X2 脚的晶体。

[2] 在 $V_{DD}=0.5$ 和室温下测得。

[3] 这些参数根据特性得到, 未经测试。

[4] 该参数只在 $0^{\circ}C\sim +70^{\circ}C$ 温度范围内存在。

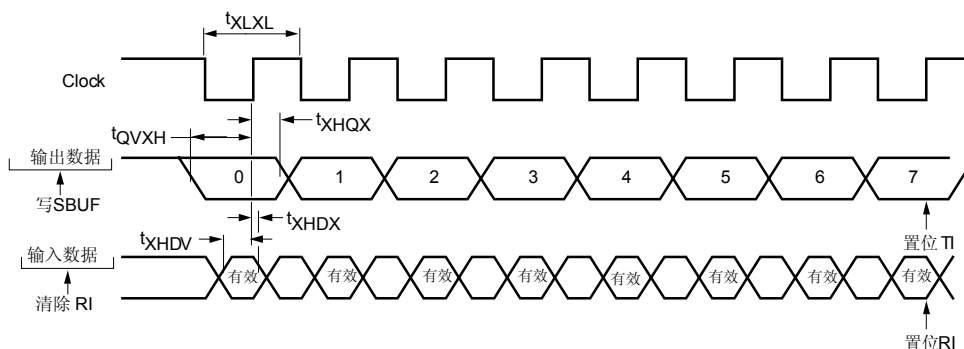


图 29 移位寄存器模式时序

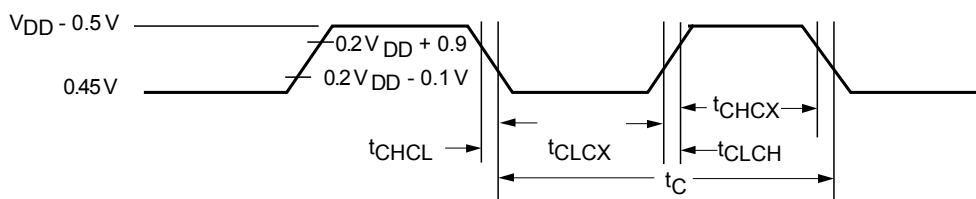


图 30 外部时钟时序