P89LPC906/907/908 使用指南

1. 概述....4

管脚配置....4
产品对照....5
管脚描述－P89LPC906....9
管脚描述－P89LPC907....10
管脚描述－P89LPC908....11
特殊功能寄存器....11
存储器结构....18

2. 时钟....19

增强型 CPU....19
时钟定义....19
CPU 时钟(OSCCLK)....19
低频振荡器选项－P89LPC906....19
中频振荡器选项－P89LPC906....19
高频振荡器选项－P89LPC906....19
振荡器选项选择－P89LPC906....20
时钟输出－P89LPC906....20
片内 RC 振荡器选项....20
看门狗振荡器选项....20
外部时钟输入选项－P89LPC906....20
CPU 时钟（CCLK）唤醒延迟....21
CPU 时钟（CCLK）调整：DIVM 寄存器....21
低功耗选择（P89LPC906）....21

3. 中断....22

中断优先级结构....23
外部中断输入....24
外部中断管脚干扰抑制....24

4. I/O 口....25

I/O 口配置....25
准双向口输出配置....25
开漏输出配置....26
仅为输入配置....26
推挽输出配置....26
P0 口模拟功能....27

5. 定时器 0 和 1....28

模式 0.....29
模式 2.....30
模式 3.....30
模式 6－P89LPC907.....30
定时器溢出触发输出－P89LPC907.....32

6. 实时时钟/系统定时器.....33

实时时钟源.....33
RTCS 1-0 的更改.....35
实时时钟中断/唤醒.....35
影响实时时钟的复位源.....35

7. 电源监控功能.....35

掉电检测.....35
上电检测.....36
节电模式.....37

8. UART（P89LPC907, P89LPC908）.....39

模式 0.....39
模式 1.....39
模式 2.....39
模式 3.....39
SFR 的位置.....39
波特率发生器及选择.....40
更新 BRGR1 和 BRGR0 寄存器.....40
帧错误.....41
间隔检测.....41
更多关于 UART 模式 0 的信息.....43
更多关于 UART 模式 1 的信息.....44
更多关于模式 2 和模式 3 的信息.....44
模式 2 和 3 中 SM2=1 时的帧错误和 RI.....45
间隔检测.....45
双缓冲.....45
不同模式中的双缓冲.....45
双缓冲使能时的发送中断（模式 1，2 和 3）.....46
双缓冲中的第 9 位（位 8）数据（模式 1，2 和 3）.....46
多机通信.....47
自动地址识别.....47

9. 复位.....48

上电复位后代码的执行.....49

10. 模拟比较器.....50

比较器配置.....50
内部参考电压.....51

比较器中断.....51
比较器和节电模式.....52
比较器配置举例.....52

11. 键盘中断（KBI） 52

12. 看门狗定时器.....54

看门狗功能.....54
看门狗清零序列.....55
定时器模式中的看门狗定时器.....57
掉电操作.....58
看门狗时钟源.....58
不使用外部振荡器的掉电周期唤醒.....58

13. 附加特性.....58

软件复位.....59
双数据指针.....59

14. FLASH 程序存储器 59

概述.....59
特性.....59
IAP-LITE 简介.....60
FLASH 作为数据存储器使用 60
其它 FLASH 单元的访问.....62
擦除-编程其它 FLASH 单元 63
读其它 FLASH 单元.....63
用户配置字节.....65
用户保密字节.....66
引导向量.....67
引导状态.....68

15. 指令集.....68

16. 修订记录.....71

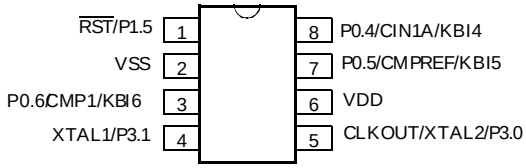
1. 概述

P89LPC906/907/908 是一款单片封装的微控制器，适合于许多要求高集成度、低成本的场合，可以满足多方面的性能要求。P89LPC906/907/908 采用了高性能的处理器结构，指令执行速度 6 倍于标准 80C51 器件。P89LPC906/907/908 集成了许多系统级的功能，这样可大大减少元件的数目和电路板面积并降低系统的成本。

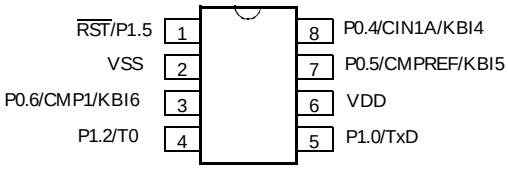
管脚配置

8 脚封装

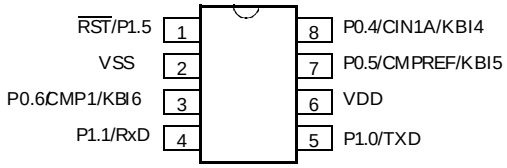
P89LPC906



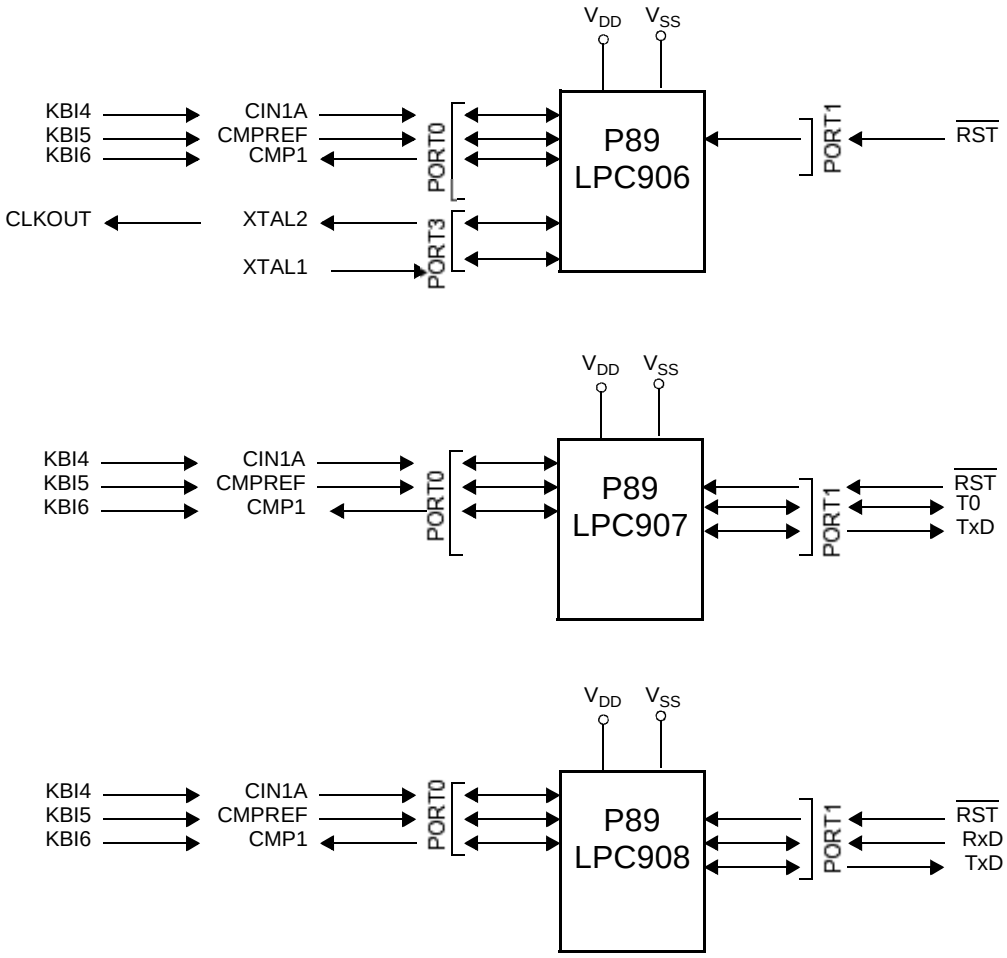
P89LPC907



P89LPC908



逻辑符号

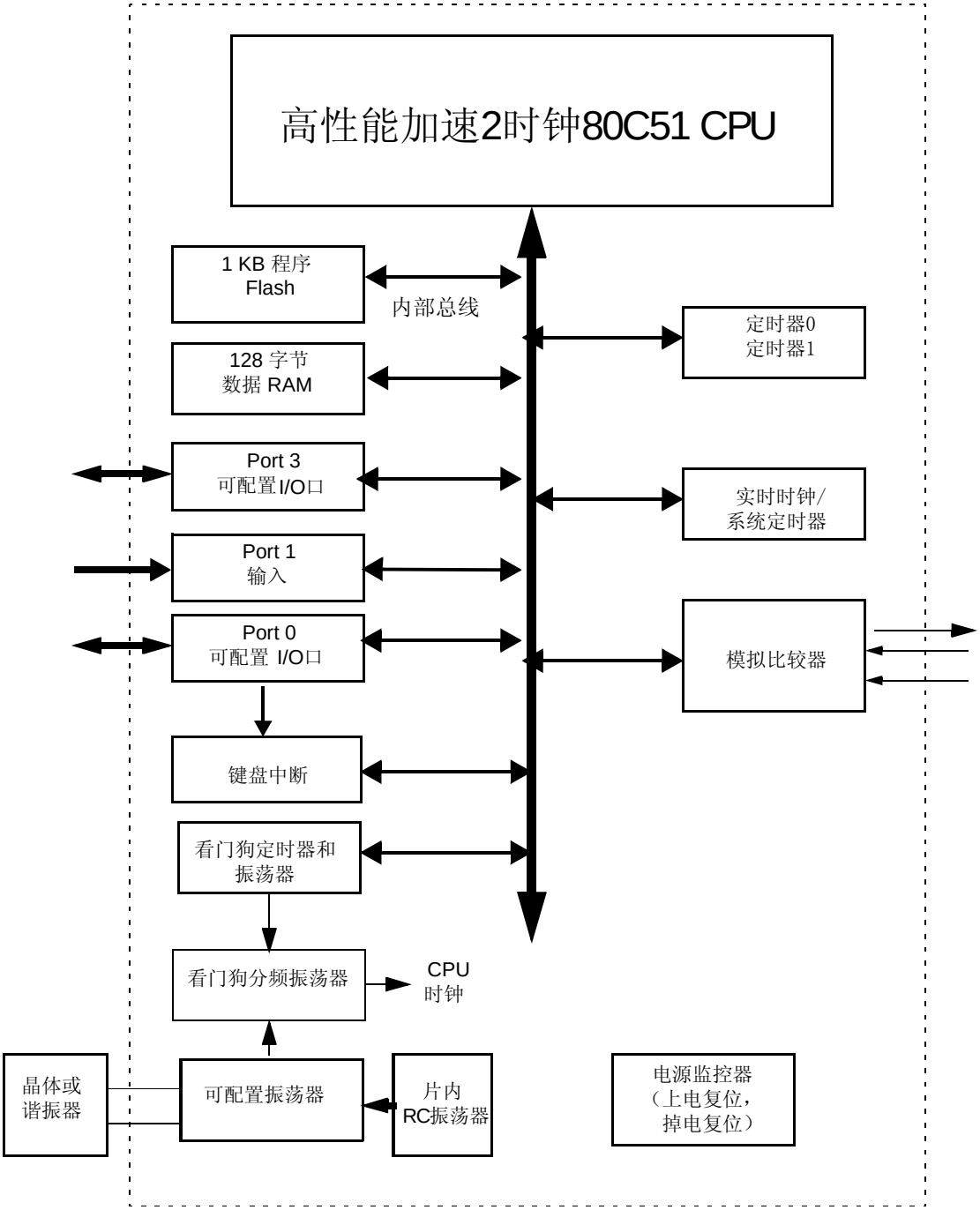


产品对照

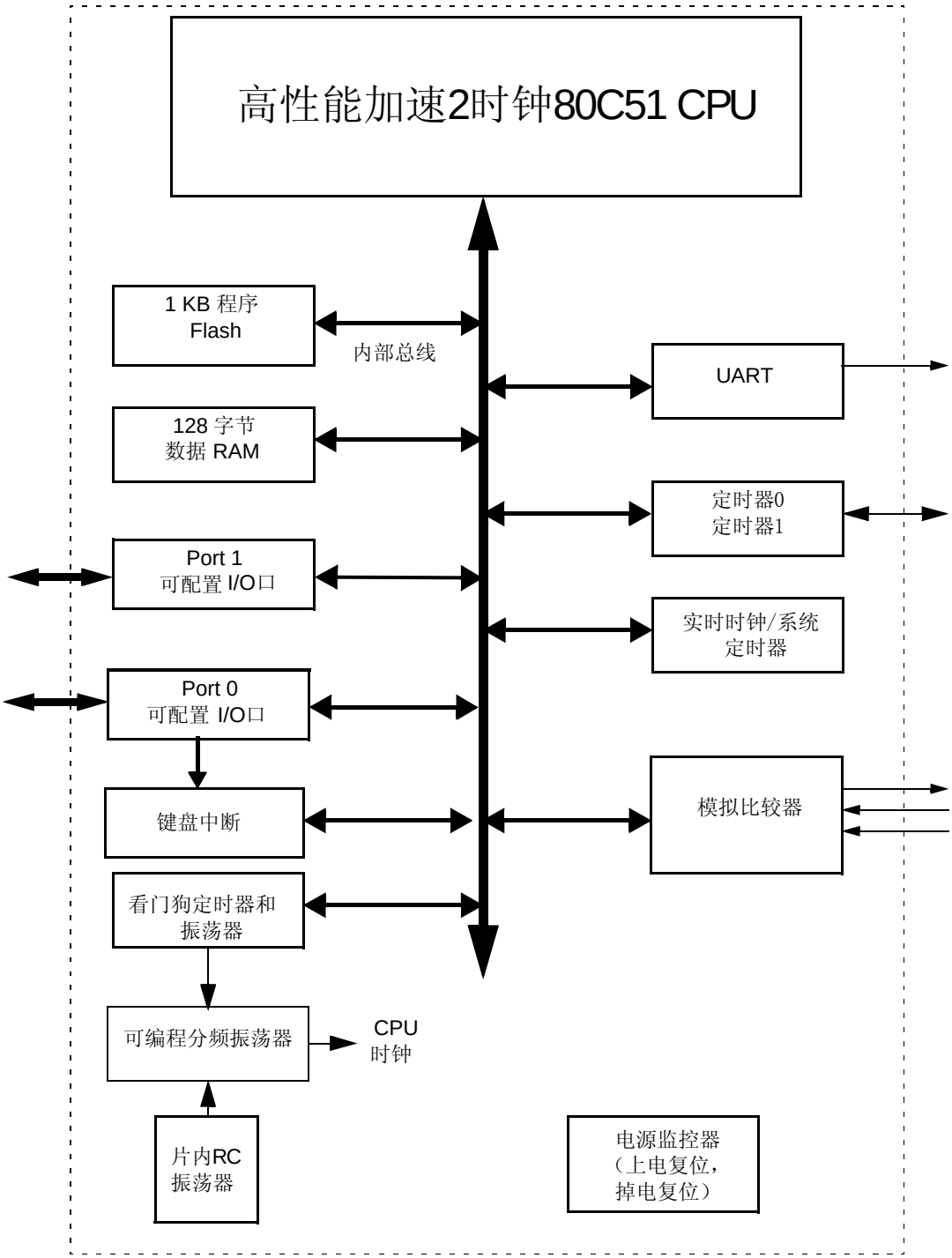
下图列出了这三种器件的不同点。

器件型号	外部晶振管脚	CLKOUT 输出	T0 PWM 输出	模拟比较器	UART	
					TxD	RxD
P89LPC906	x	x	—	x	—	—
P89LPC907	—	—	x	x	x	—
P89LPC908	—	—	—	x	x	x

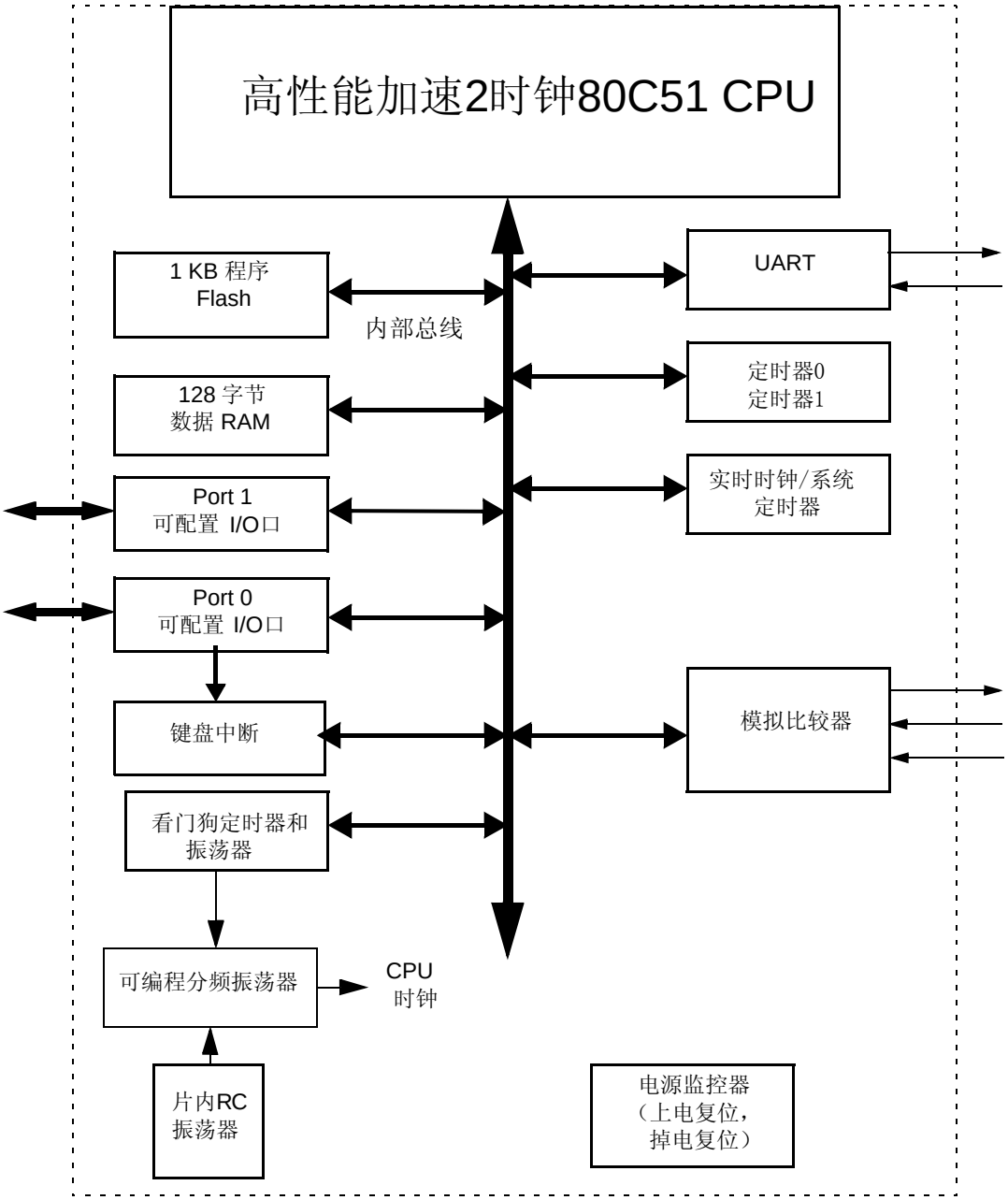
功能框图－P89LPC906



功能框图－P89LPC907



功能框图－P89LPC908



管脚描述－P89LPC906

助记符	管脚号	类型	名称及功能描述
P0.4-P0.6	3,7,8	I/O	**PORT0**：P0 是一个可由用户定义输出类型的 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”。 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：
	8	I/O I I	**P0.4** P0 口位 4 **CIN1A** 比较器正向输入 A **KBI4** 键盘输入 4
	7	I/O I I	**P0.5** P0 口位 5 **CMPREF** 比较器参考输入（负） **KBI5** 键盘输入 5
	3	I/O O I	**P0.6** P0 口位 6 **CMP1** 比较器 1 输出 **KBI6** 键盘输入 6
P1.5	1	I I	**P1.5** P1 口位 5（仅为输入） **$\overline{\text{RST}}$** 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在电路编程模式。
P3.0~P3.1	4,5	I/O	**PORT3**：P3 口是一个可由用户定义输出类型的 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”。 所有管脚都具有施密特触发输入。 P3 口还可提供如下特殊功能：
	5	I/O O O	**P3.0** P3 口位 0。 **XTAL2** 振荡放大器输出（由 Flash 配置选择为晶体振荡器时） **CLKOUT** 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）。
	4	I/O I	**P3.1** P3 口位 1。 **XTAL1** 振荡电路和内部时钟发生器输入（通过 Flash 配置选择）。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_{SS}	2	I	**地**: 0V 参考点
V_{DD}	6	I	**电源**：正常操作模式、空闲模式和掉电模式时的电源。

管脚描述－P89LPC907

助记符	管脚号	类型	名称及功能描述
P0.4-P0.6	3,7,8	I/O	**PORT0**：P0 是一个可由用户定义输出类型的 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”。 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：
	8	I/O I I	**P0.4** P0 口位 4 **CIN1A** 比较器正向输入 **KBI4** 键盘输入 4
	7	I/O I I	**P0.5** P0 口位 5 **CMPREF** 比较器参考输入（负） **KBI5** 键盘输入 5
	3	I/O O I	**P0.6** P0口位6 **CMP1** 比较器1输出 **KBI6** 键盘输入6
P1.0-P1.5	1,4,5		**PORT1**： P1 口是一个可由用户定义输出类型的 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”。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：
	5	I/O O	**P1.0** P1 口位 0 **TxD** 串口发送
	4	I/O I/O	**P1.2** P1 口位 2 **T0** 定时器 0 外部时钟输入，翻转输出，PWM 输出
	1	I I	**P1.5** P1 口位 5（仅为输入） **$\overline{\text{RST}}$** 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在电路编程模式。
V_{SS}	2	I	**地**: 0V 参考点
V_{DD}	6	I	**电源**：正常操作模式、空闲模式和掉电模式时的电源。

管脚描述－P89LPC908

助记符	管脚号	类型	名称及功能描述
P0.4-P0.6	3,7,8	I/O	**PORT0**：P0 是一个可由用户定义输出类型的 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”。 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：
	8	I/O I I	**P0.4** P0 口位 4 **CIN1A** 比较器正向输入 **KBI4** 键盘输入 4
	7	I/O I I	**P0.5** P0 口位 5 **CMPREF** 比较器参考输入（负） **KBI5** 键盘输入 5
	3	I/O O I	**P0.6** P0口位6 **CMP1** 比较器1输出 **KBI6** 键盘输入6
P1.0-P1.5	1,4,5		**PORT1**： P1 口是一个可由用户定义输出类型的 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”。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：
	5	I/O O	**P1.0** P1 口位 0 **TXD** 串口发送
	4	I/O I	**P1.1** P1 口位 1 **RXD** 串口接收
	1	I I	**P1.5** P1 口位 5（仅为输入） **$\overline{\text{RST}}$** 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在电路编程模式。
V_{SS}	2	I	**地**: 0V 参考点
V_{DD}	6	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（SFR）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：
   －除非特别说明，‘-’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 1 特殊功能寄存器－P89LPC906

名称	定义	地址	位功能和位地址 MSB							LSB	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	-	-	ENT0	SRST	0	-	DPS	00H[1]
			F7	F6	F5	F4	F3	F2	F1	F0	
B*	B 寄存器	F0H									00H
CMP1#	比较器 1 控制	ACH	-	-	CE1	-	CN1	OE1	CO1	CMF1	00H[1]
DIVM#	CPU 时钟分频控制	95H									00H
DPTR	数据指针（2 字节）										
DPH	指针高字节	83H									00H
DPL	指针低字节	82H									00H
FMADRH#	编程 Flash 地址高字节	E7H									00H
FMADRL#	编程 Flash 地址低字节	E6H									00H
FMCON#	编程 Flash 控制（读）	E4H	BUSY	-	-	-	HVA	HVE	SV	OI	70H
	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
IEN0*	中断使能 0	A8H	EA	EWDRT	EBO	-	ET1	-	ET0	-	00H
			EF	EE	ED	EC	EB	EA	E9	E8	
IEN1*#	中断使能 1	E8H	-	-	-	-	-	EC	EKBI	-	00H[1]
			BF	BE	BD	BC	BBB	BA	B9	B8	
IP0*	中断优先级 0	B8H	-	PWDRT	PBO	-	PT1	-	PT0	-	00H[1]
IP0H#	中断优先级 0 高字节	B7H	-	PWDRTH	PBOH	-	PT1H	-	PT0H	-	00H[1]
			FF	FE	FD	FC	FB	FA	F9	F8	
IP1*#	中断优先级 1	F8H	-	-	-	-	-	PC	PKBI	-	00H[1]
IP1H#	中断优先级 1 高字节	F7H	-	-	-	-	-	PCH	PKBIH	-	00H[1]
KBCON#	键盘控制寄存器	94H	-	-	-	-	-	-	PATN_SEL	KBIF	00H[1]
KBMASK#	键盘中断屏蔽	86H									00H
KBPATN#	键盘模式	93H									FFH
			87	86	85	84	83	82	81	80	
P0*	P0 口	80H		CMP1/KB6	CMPREF/KB5	CIN1A/KB4		-			注 1
			97	96	95	94	93	92	91	90	
P1*	P1 口	90H			$\overline{\text{RST}}$			-			
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	P3 口	B0H							XTAL1	XTAL2	注 1
P0M1#	0 口输出模式选择 1	84H		P0M1.6	P0M1.5	P0M1.4		-			FFH

续上表

名称	定义	地址	位功能和位地址 MSB LSB								复位值
			97	96	95	94	93	92	91	90	
P0M2#	0 口输出模式选择 2	85H		P0M2.6	P0M2.5	P0M2.4		-			00H
P1M1#	1 口输出模式选择 1	91H			P1M1.5			-			FFH[1]
P1M2#	1 口输出模式选择 2	92H			P1M2.5			-			00H[1]
P3M1#	3 口输出模式选择 1	B1H							P3M1.1	P3M1.0	03H[1]
P3M2#	3 口输出模式选择 2	B2H							P3M2.1	P3M2.0	00H[1]
PCON#	电源控制寄存器	87H	-	-	BOPD	BOI	GF1	GF0	PMOD1	PMOD0	00H
PCONA#	电源控制寄存器 A	B5H	RTCPD	-	VCPD	-	-	-	-	-	00H[1]
			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#	0 口数字输入禁能	F6H	-	-	PT0AD.5	PT0AD.4	-	-	-	-	00H
RSTSRC#	复位源寄存器	DFH	-	-	BOF	POF	-	R_WD	R_SF	R_EX	注 2
RTCCON#	实时时钟控制	D1H	RTCF	RTCS1	RTCS0	-	-	-	ERTC	RTCEN	60H[1,5]
RTCH#	实时时钟高字节	D2H									00H[5]
RTCL#	实时时钟低字节	D3H									00H[5]
SP	堆栈指针	81H									07H
TAMOD#	定时器 0 附加模式	8FH	-	-	-	-	-	-	-	T0M2	00H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	定时器 0 和 1 控制	88H	TF1	TR1	TF0	TR0	-	-	-	-	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 模式	89H	-	-	T1M1	T1M0	-	-	T0M1	T0M0	00H
TRIM#	内部振荡调整寄存器	96H	-	ENCLK	TRIM.5	TRIM.4	TRIM.3	TRIM.2	TRIM.1	TRIM.0	注 4,5
WDCON#	看门狗控制寄存器	A7H	PRE2	PRE1	PRE0	-	-	WDRUN	WDTOF	WDCLK	注 3,5
WDL#	看门狗装载	C1H									FFH
WFEED1#	看门狗清零 1	C2H									
WFEED2#	看门狗清零 2	C3H									

表 2 特殊功能寄存器－P89LPC907

名称	定义	地址	位功能和位地址 MSB LSB								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	-	-	-	-	SRST	0	-	DPS	00H[1]
			F7	F6	F5	F4	F3	F2	F1	F0	
B*	B 寄存器	F0H									00H
BRGR0#§	波特率发生器低字节	BEH									00H

续上表

名称	定义	地址	位功能和位地址 MSB							LSB	复位值
			E7	E6	E5	E4	E3	E2	E1	E0	
BRGR1#§	波特率发生器高字节	BFH									00H
BRGCON#	波特率发生器控制	BDH	-	-	-	-	-	-	SBRGS	BRGEN	00H
CMP1#	比较器 1 控制	ACH	-	-	CE1	-	CN1	OE1	CO1	CMF1	00H[1]
DIVM#	CPU 时钟分频控制	95H									00H
DPTR	数据指针（2 字节）										
DPH	指针高字节	83H									00H
DPL	指针低字节	82H									00H
FMADRH#	编程 Flash 地址高字节	E7H									00H
FMADRL#	编程 Flash 地址低字节	E6H									00H
FMCON#	编程 Flash 控制（读）	E4H	BUSY	-	-	-	HVA	HVE	SV	OI	70H
	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
IEN0*	中断使能 0	A8H	EA	EWDRT	EBO	ES	ET1	-	ET0	-	00H
			EF	EE	ED	EC	EB	EA	E9	E8	
IEN1*#	中断使能 1	E8H	-	EST	-	-	-	EC	EKBI	-	00H[1]
			BF	BE	BD	BC	BBB	BA	B9	B8	
IP0*	中断优先级 0	B8H	-	PWDRT	PBO	PS	PT1	-	PT0	-	00H[1]
IP0H#	中断优先级 0 高字节	B7H	-	PWDRTH	PBOH	PSH	PT1H	-	PT0H	-	00H[1]
			FF	FE	FD	FC	FB	FA	F9	F8	
IP1*#	中断优先级 1	F8H	-	PST	-	-	-	PC	PKBI	-	00H[1]
IP1H#	中断优先级 1 高字节	F7H	-	PSTH	-	-	-	PCH	PKBIH	-	00H[1]
KBCON#	键盘控制寄存器	94H	-	-	-	-	-	-	PATN_SEL	KBIF	00H[1]
KBMASK#	键盘中断屏蔽	86H									00H
KBPATN#	键盘模式	93H									FFH
			87	86	85	84	83	82	81	80	
P0*	P0 口	80H		CMP1/KB6	CMPREF/KB5	CIN1A/KB4		KB2		KB0	注 1
			97	96	95	94	93	92	91	90	
P1*	P1 口	90H			$\overline{\text{RST}}$			T0		TxD	
			B7	B6	B5	B4	B3	B2	B1	B0	
P0M1#	0 口输出模式选择 1	84H		P0M1.6	P0M1.5	P0M1.4		P0M1.2		P0M1.0	FFH
P0M2#	0 口输出模式选择 2	85H		P0M2.6	P0M2.5	P0M2.4		P0M2.2		P0M2.0	00H

续上表

名称	定义	地址	位功能和位地址 MSB							LSB	复位值
			B7	B6	B5	B4	B3	B2	B1	B0	
P1M1#	1 口输出模式选择 1	91H			P1M1.5			P1M1.2		P1M1.0	FFH[1]
P1M2#	1 口输出模式选择 2	92H			P1M2.5			P1M2.2		P1M2.0	00H[1]
PCON#	电源控制寄存器	87H	SMOD1	SMOD0	BOPD	BOI	GF1	GF0	PMOD1	PMOD0	00H
PCONA#	电源控制寄存器 A	B5H	RTCPD	-	VCPD	-	-	-	SPD	-	00H[1]
			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#	0 口数字输入禁能	F6H	-	-	PT0AD.5	PT0AD.4	-	-	-	-	00H
RSTSRC#	复位源寄存器	DFH	-	-	BOF	POF	-	R_WD	R_SF	R_EX	注 2
RTCCON#	实时时钟控制	D1H	RTCF	RTCS1	RTCS0	-	-	-	ERTC	RTCEN	60H[1,5]
RTCH#	实时时钟高字节	D2H									00H[5]
RTCL#	实时时钟低字节	D3H									00H[5]
SBUF	串口数据缓冲寄存器	99H									xxH
SCON*	串行口控制	98H	SM0	SM1	SM2		TB8		TI		00H
SSTAT#	串行口扩展状态	BAH	DBMOD	INTLO	CIDIS	DBISEL					00H
SP	堆栈指针	81H									07H
TAMOD#	定时器 0 辅助模式	8FH	-	-	-	-	-	-	-	T0M2	00H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	定时器 0 和 1 控制	88H	TF1	TR1	TF0	TR0	-	-	-	-	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 模式	89H	-	-	T1M1	T1M0	-	-	T0M1	T0M0	00H
TRIM#	内部振荡调整寄存器	96H	-	-	TRIM.5	TRIM.4	TRIM.3	TRIM.2	TRIM.1	TRIM.0	注 4,5
WDCON#	看门狗控制寄存器	A7H	PRE2	PRE1	PRE0	-	-	WDRUN	WDTOF	WDCLK	注 3,5
WDL#	看门狗装载	C1H									FFH
WFEED1#	看门狗清零 1	C2H									
WFEED2#	看门狗清零 2	C3H									

表 3 特殊功能寄存器－P89LPC908

名称	定义	地址	位功能和位地址 MSB							LSB	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	-	EBRR	-	-	SRST	0	-	DPS	00H[1]
			F7	F6	F5	F4	F3	F2	F1	F0	
B*	B 寄存器	F0H									00H
BRGR0#§	波特率发生器低字节	BEH									00H
BRGR1#§	波特率发生器高字节	BFH									00H
BRGCON#	波特率发生器控制	BDH	-	-	-	-	-	-	SBRGS	BRGEN	00H
CMP1#	比较器 1 控制	ACH	-	-	CE1	-	CN1	OE1	CO1	CMF1	00H[1]
DIVM#	CPU 时钟分频控制	95H									00H
DPTR	数据指针（2 字节）										
DPH	指针高字节	83H									00H
DPL	指针低字节	82H									00H
FMADRH#	编程 Flash 地址高字节	E7H									00H
FMADRL#	编程 Flash 地址低字节	E6H									00H
FMCON#	编程 Flash 控制（读）	E4H	BUSY	-	-	-	HVA	HVE	SV	OI	70H
	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
IEN0*	中断使能 0	A8H	EA	EWDRT	EBO	ES/ESR	ET1	-	ET0	-	00H
			EF	EE	ED	EC	EB	EA	E9	E8	
IEN1*#	中断使能 1	E8H	-	EST	-	-	-	EC	EKBI	-	00H[1]
			BF	BE	BD	BC	BBB	BA	B9	B8	
IP0*	中断优先级 0	B8H	-	PWDRT	PBO	PS/PSR	PT1	-	PT0	-	00H[1]
IP0H#	中断优先级 0 高字节	B7H	-	PWDRTH	PBOH	PSH/PSRH	PT1H	-	PT0H	-	00H[1]
			FF	FE	FD	FC	FB	FA	F9	F8	
IP1*#	中断优先级 1	F8H	-	PST	-	-	-	PC	PKBI	-	00H[1]
IP1H#	中断优先级 1 高字节	F7H	-	PSTH	-	-	-	PCH	PKBIH	-	00H[1]
KBCON#	键盘控制寄存器	94H	-	-	-	-	-	-	PATN_SEL	KBIF	00H[1]
KBMASK#	键盘中断屏蔽	86H									00H
KBPATN#	键盘模式	93H									FFH

续上表

名称	定义	地址	位功能和位地址 MSB							LSB	复位值
			87	86	85	84	83	82	81	80	
P0*	P0 口	80H		CMP1/ KB6	CMPREF/ KB5	CIN1A/ KB4		KB2		-	注 1
			97	96	95	94	93	92	91	90	
P1*	P1 口	90H			$\overline{RST}$			-	RxD	TxD	
P0M1#	0 口输出模式选择 1	84H		P0M1.6	P0M1.5	P0M1.4		P0M1.2		-	FFH
P0M2#	0 口输出模式选择 2	85H		P0M2.6	P0M2.5	P0M2.4		P0M2.2		-	00H
P1M1#	1 口输出模式选择 1	91H			P1M1.5			-	P1M1.1	P1M1.0	FFH[1]
P1M2#	1 口输出模式选择 2	92H			P1M2.5			-	P1M2.1	P1M2.0	00H[1]
PCON#	电源控制寄存器	87H	SMOD1	SMOD0	BOPD	BOI	GF1	GF0	PMOD1	PMOD0	00H
PCONA#	电源控制寄存器 A	B5H	RTCPD	-	VCPD	-	-	-	SPD	-	00H[1]
			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#	0 口数字输入禁能	F6H	-	-	PT0AD.5	PT0AD.4	-	PT0AD.2	-	-	00H
RSTSRC#	复位源寄存器	DFH	-	-	BOF	POF	R_BK	R_WD	R_SF	R_EX	注 2
RTCCON#	实时时钟控制	D1H	RTCF	RTCS1	RTCS0	-	-	-	ERTC	RTCEN	60H[1,5]
RTCH#	实时时钟高字节	D2H									00H[5]
RTCL#	实时时钟低字节	D3H									00H[5]
SADDR#	串口地址寄存器	A9H									00H
SADEN#	串口地址使能	B9H									00H
SBUF	串口数据缓冲区	99H									xxH
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	串行口控制	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SSTAT#	串行口扩展状态	BAH	DBMOD	INTLO	CIDIS	DBISEL	FE	BR	OE	STINT	00H
SP	堆栈指针	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	定时器 0 和 1 控制	88H	TF1	TR1	TF0	TR0	-	-	-	-	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 模式	89H	-	-	T1M1	T1M0	-	-	T0M1	T0M0	00H
TRIM#	内部振荡调整寄存器	96H	-	-	TRIM.5	TRIM.4	TRIM.3	TRIM.2	TRIM.1	TRIM.0	注 4,5
WDCON#	看门狗控制寄存器	A7H	PRE2	PRE1	PRE0	-	-	WDRUN	WDTOF	WDCLK	注 3,5
WDL#	看门狗装载	C1H									FFH

续上表

名称	定义	地址	位功能和位地址 MSB LSB								复位值
			8F	8E	8D	8C	8B	8A	89	88	
WFEED1#	看门狗清零 1	C2H									
WFEED2#	看门狗清零 2	C3H									

注：

* SFR 可位寻址。

\# 由 80C51 SFR 修改或增加所得。

\- 保留位，必须写入‘0’。

§ 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。

SFR 中的无效位（标有“-”的位）总是为“X”（未知）。除非特别指明，否则不要向这些位写入 1，因为它们可能在将来扩展用于其它功能。这些位的复位值均为 0，读出来的值不可知。

1. 上电复位后所有的 I/O 口都仅为输入（高阻）状态。
2. RSTSRC 寄存器反映 P89LPC906/907/908 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 111001x1，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。其它复位不会引起 TRIM 寄存器的初始化。
5. 上电复位是唯一影响这些特殊功能寄存器的复位。

存储器结构

P89LPC906/907/908 存储空间分配如图 1.1 所示。

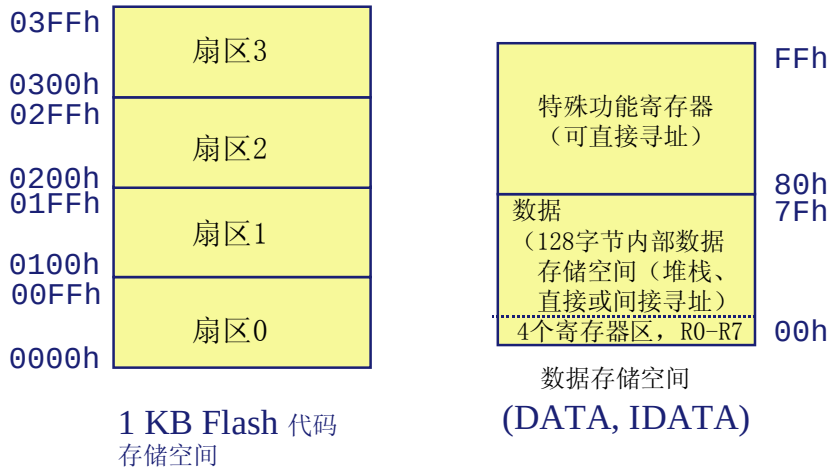


图 1.1 P89LPC906/907/908 存储空间分配

P89LPC906/907/908 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。

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。

CODE 1KB 字节代码存储空间。作为程序执行部分和通过 MOVC 指令访问。

2. 时钟

增强型 CPU

P89LPC906/907/908 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。

时钟定义

P89LPC906/907/908 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 2.3,图 2.4），也可降低到较低的频率（见“CPU 时钟（CCLK）调整：DIVM 寄存器”）。

  注：f_{OSC} 定义成 OSCCLK 的频率。
- XCLK－晶体振荡器的输出（P89LPC906）
- CCLK－CPU 时钟。
- PCLK－用于不同外围器件的时钟，为 CCLK/2。

CPU 时钟(OSCCLK)

P89LPC906 提供几个可由用户选择的振荡器选项。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。

P89LPC907 和 P89LPC908 可选择片内看门狗振荡器和片内 RC 振荡器作为 CPU 的时钟源。

低频振荡器选项－P89LPC906

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－P89LPC906

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－P89LPC906

此选项支持 4MHz～12MHz 外部晶振，同时也支持陶瓷谐振器。如果 CCLK 是 8MHz 或更低，CLKLP SFR 位（AUXR1.7）可设置成‘1’来降低功耗。复位时，CLKLP 为‘0’，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。

振荡器必须配置成以下方式：
－低频振荡器
－中频振荡器
－高频振荡器

*为限制振荡器驱动电平可串联电阻，这对于低频振荡器尤其重要。

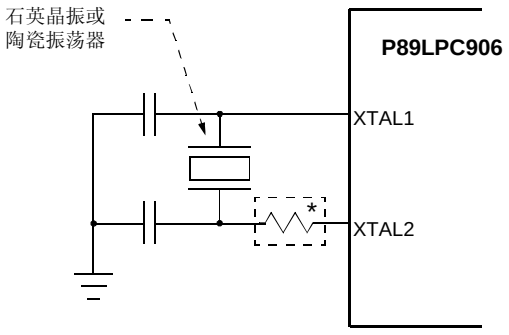


图 2.1 使用晶振－P89LPC906

振荡器选项选择－P89LPC906

通过 UCFG1 的 FOSC2:0 或 RTCCON 的 RTCS1:0 来选择振荡器。如果通过设置 FOSC2:0 位选择了内部 RC 振荡器或 WDT 振荡器作为 OSCCLK 时钟源，则 RTCS1:0 位只可选择晶体振荡器。反之，由 FOSC2:0 选择晶体振荡器。见表 6.1 和表 6.2。

时钟输出－P89LPC906

P89LPC906 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。要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）并且没有使用晶振作为实时时钟的时钟源。这样可使外部器件与 P89LPC906 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：复位时，TRIM SFR 初始化为出厂的预编程值。因此当用户需要置位或清零 ENCLK 位时，应当使 TRIM 寄存器 5:0 位的值保持不变。实现方法如下：可先读出 TRIM 的内容（例如读入 ACC），然后修改 TRIM.6 后再写入 TRIM；或者，使用 ANL 或 ORL 指令将 TRIM.6 位置位或清零来实现。振荡器的频率随着 TRIM 值的增加而降低。

片内 RC 振荡器选项

P89LPC906/907/908 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，±1%。（注：初始值误差小于 1%，请参考数据手册的温度范围内的操作。）。用户程序可修改 TRIM 寄存器将 RC 振荡器调整为其它频率，振荡器的频率随着 TRIM 值的增加而降低。

如果 CCLK 是 8MHz 或更低，CLKLP SFR 位（AUXR1.7）可设置成‘1’来降低功耗。复位时，CLKLP 为‘0’，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。

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－P89LPC906

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。

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：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 脚（P3.0）输出 CCLK/2 的时钟。当 ENCLK=0 时无时钟输出（P89LPC906）。
TRIM.5-0		频率调整值。

注：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 指令来清零或置位 TRIM 的位 6。

图 2.2 片内 RC 振荡器 TRIM 寄存器

CPU 时钟（CCLK）唤醒延迟

P89LPC906/907/908 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。如果时钟源为三个晶振选项中的任意一个（P89LPC906），延迟时间为 992 个 OSCCLK 周期加 60-100us，如果时钟源是内部 RC 振荡器或看门狗振荡器，延迟时间为 224 个 OSCCLK 周期加 60-100us。

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 510 分频来提供 CCLK。可通过下式得出 CCLK 的频率：

CCLK 频率＝f_{OSC}/(2N)

此处，f_{OSC} 为 OSCCLK 频率。

N 是 DIVM 的值。

N 的取值范围为 0～255，因此，CCLK 的频率范围为 f_{OSC}～f_{OSC}/510（N=0 时,CCLK=f_{OSC}）。

此特性可用于暂时使 CPU 以较低频率工作以降低功耗。通过分频，程序以较低速度运行时，使 CPU 仍保持对事件响应的能力，而不只是对能产生中断的事件（能使 CPU 从空闲模式退出）才响应。这常常会得到比空闲模式更低的功耗。并且比掉电模式少了振荡器起振时间。在程序内 DIVM 的值可随时改变而无需中断程序运行。

低功耗选择（P89LPC906）

P89LPC906 设计最大工作频率为 12MHz(CCLK)。但是如果 CCLK 为 8MHz 或更低，CLKLP 位（AUXR1.7）可置位以降低功耗。此外，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。

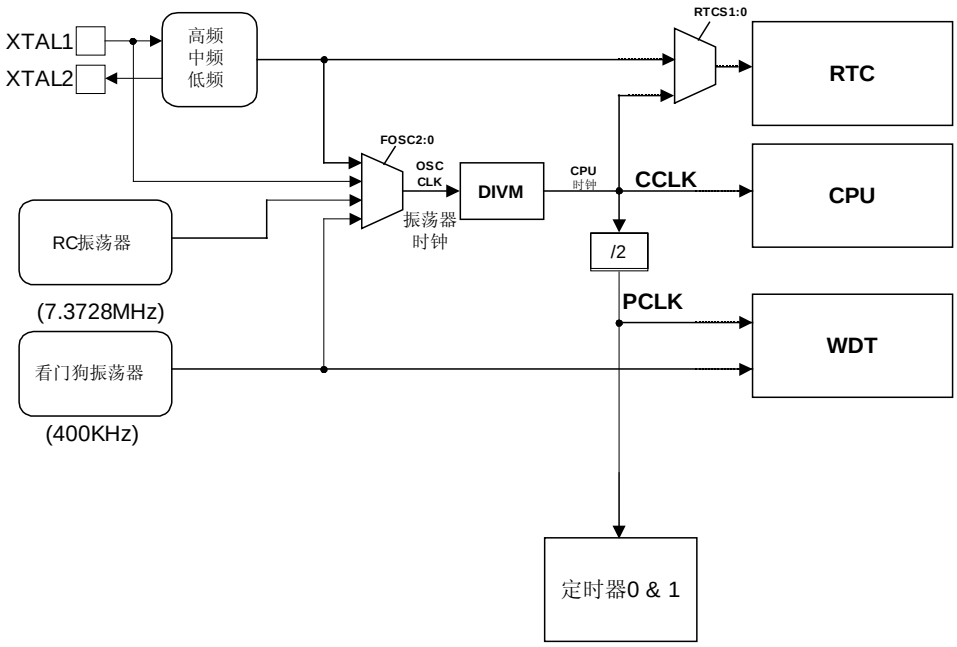


图 2.3 振荡器控制框图－P89LPC906

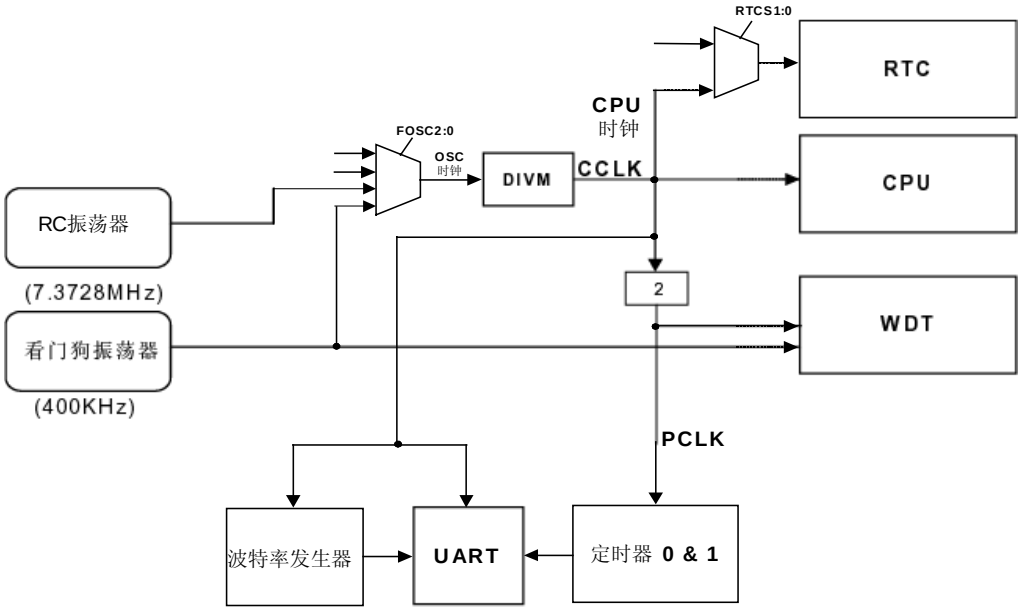


图 2.4 振荡器控制框图－P89LPC907,P89LPC908

3. 中断

P89LPC906/907/908 采用 4 中断优先级结构。这为中断源的处理提供了极大的灵活性。P89LPC906 支持 6 个中断源：定时器 0 和 1，掉电检测，看门狗/实时时钟，键盘和比较器。P89LPC907 支持 7 个中断源：定时器 0 和 1，串口 Tx，掉电检测，看门狗/实时时钟，键盘以及比较器 1 和 2。P89LPC908 支持 9 个中断源：定时器 0 和 1，串口 Tx，串口 Rx，组合串口 Rx/Tx，掉电检测，看门狗/实时时钟，键盘以及定时器 1 和 2。

任何一个中断源均可通过对中断使能寄存器 IEN0 或 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中断优先级寄存器 IP0，

IP0H，IP1，IP1H 中相应位来实现。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。当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 3.2 对中断源、标志位、向量地址、使能位、优先级位、仲裁队列和每个中断是否可将 CPU 从掉电模式唤醒等方面进行了总结。

中断优先级结构

P89LPC906/907/908 包含 4 个与 4 中断优先级结构相关的寄存器：IP0,IP0H,IP1,IP1H。每个中断占用寄存器的两位，Ipx 和 IpxH（x＝0,1），如表 3.1 所示。

表 3.1 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 3.2 中断总结－P89LPC906

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
定时器 0 中断	TF0	000Bh	ET0(IEN0.1)	IP0H.1,IP0.1	3	No
定时器 1 中断	TF1	001Bh	ET1(IEN0.3)	IP0H.3,IP0.3	5	No
掉电检测	BOF	002Bh	EBO(IEN0.5)	IP0H.5,IP0.5	1	Yes
看门狗定时器/实时时钟	WDOVF/RTCF	0053h	EWDRT(IEN0.6)	IP0H.6,IP0.6	2	Yes
KBI 中断	KBIF	003Bh	EKBI(IEN1.1)	IP1H.1,IP1.1	4	Yes
比较器中断	CMF	0043h	EC(IEN1.2)	IP1H.2,IP1.2	6	Yes

表 3.3 中断总结－P89LPC907,P89LPC908

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
定时器 0 中断	TF0	000Bh	ET0(IEN0.1)	IP0H.1,IP0.1	3	No
定时器 1 中断	TF1	001Bh	ET1(IEN0.3)	IP0H.3,IP0.3	5	No
串口 Tx 和 Rx[1,3]	TI&RI	0023h	ES/ESR (IEN0.4)	IP0H.4,IP0.4	8	No
串口 Rx[1,3]	RI					
掉电检测	BOF	002Bh	EBO(IEN0.5)	IP0H.5,IP0.5	1	Yes
看门狗定时器/实时时钟	WDOVF/RTCF	0053h	EWDRT (IEN0.6)	IP0H.6,IP0.6	2	Yes
KBI 中断	KBIF	003Bh	EKBI(IEN1.1)	IP1H.1,IP1.1	4	Yes
比较器中断	CMF	0043h	EC(IEN1.2)	IP1H.2,IP1.2	6	Yes
串口 Tx[2]	TI	006Bh	EST(IEN1.6)	IP1H.6,IP1.6	7	No

续上表

1. SSTAT.5=0 选择串口（UART）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，见下面的注 3）。 2. 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。尽管 P89LPC907 不含 RxD 管脚，但它仍可实现该功能，允许切换到 Tx 中断向量。 3. 如果 SSTAT.0=1，串口的附加标志位 FE,BR,OE 将导致产生该中断。

外部中断输入

P89LPC906/907/908 有一个键盘中断（见“键盘中断（KBI）”）功能，可用作外部中断输入。如果 P89LPC906/907/908 处于掉电或空闲状态时键盘中断使能，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

P89LPC906/907/908 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 P89LPC906/907/908 数据手册的 AC 电气特性）。

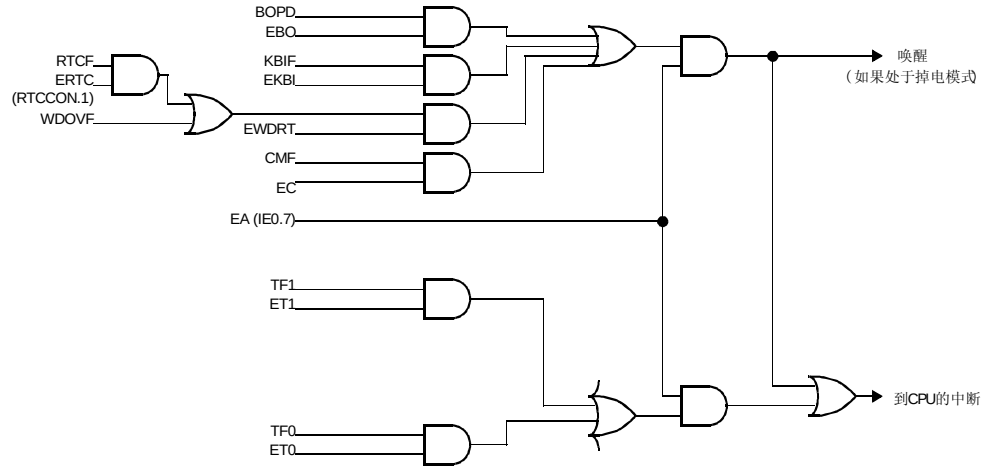


图 3.1 中断源，中断使能，掉电唤醒中断源－P89LPC906

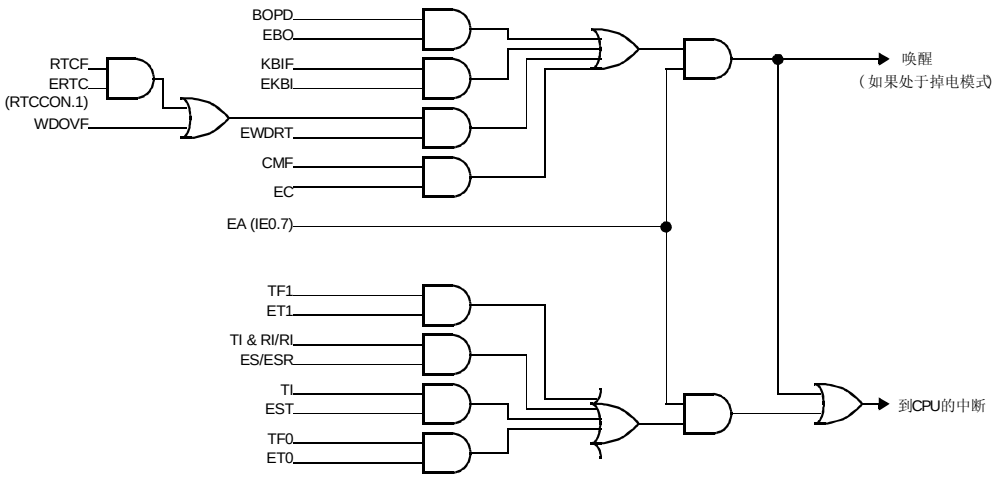


图 3.2 中断源，中断使能，掉电唤醒中断源－P89LPC907,P89LPC908

4. I/O 口

P89LPC906/907/908 有 3～6 个 I/O 口。I/O 口的具体数目取决于所选择的时钟和复位方式。

表 4.1 可用的 I/O 口数目

时钟源	复位选项	I/O 口数目
		8 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）	6
	使用外部复位脚 $\overline{\text{RST}}$	5
外部时钟输入 （P89LPC906）	无外部复位（上电时除外）	5
	使用外部复位脚 $\overline{\text{RST}}$	4
低/中/高频振荡器 （外部晶振或谐振器） （P89LPC906）	无外部复位（上电时除外）	4
	使用外部复位脚 $\overline{\text{RST}}$	3

I/O 口配置

除了 1 个口外，P89LPC906/907/908 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 4.2 所示。四种输出类型分别为：准双向口(标准 80C51 输出模式)，推挽，开漏输出或仅为输入功能。每个口配置 2 个控制寄存器控制每个管脚输出类型。P1.5（ $\overline{\text{RST}}$ ）只能作为输入口，无法进行配置。

表 4.2 口输出方式设定

PxM1.y	PxM2.y	口输出模式
0	0	准双向模式
0	1	推挽
1	0	仅为输入（高阻抗）
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，可适应不同的需要。

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低到高的转换。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 4.1 所示。

P89LPC906/907/908 为 3V 器件，但大部分管脚可承受 5V 电压（XTAL1 和 XTAL2 除外）。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 P89LPC906/907/908 数据手册中 AC 电气特性的干扰滤波器部分的内容）。

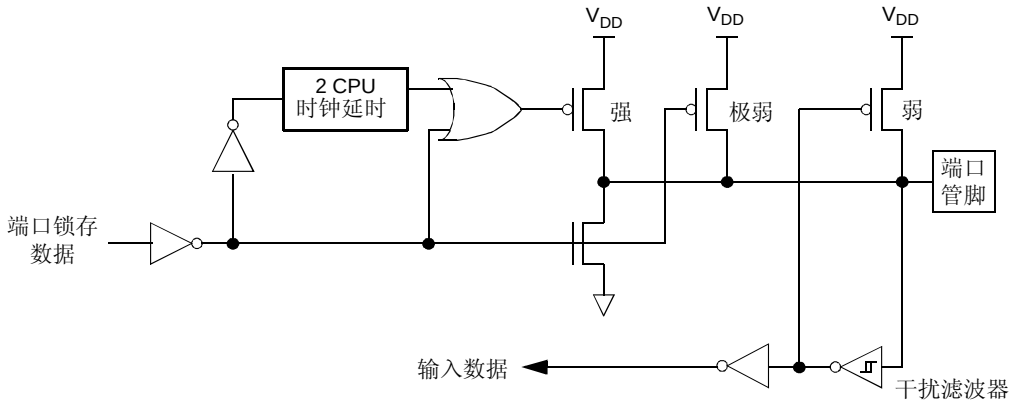


图 4.1 准双向输出

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而仅驱动下拉晶体管。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}。这种方式的下拉与准双向口相同。

开漏输出配置如图 4.2 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 P89LPC906/907/908 数据手册中 AC 电气特性的干扰滤波器部分的内容）。

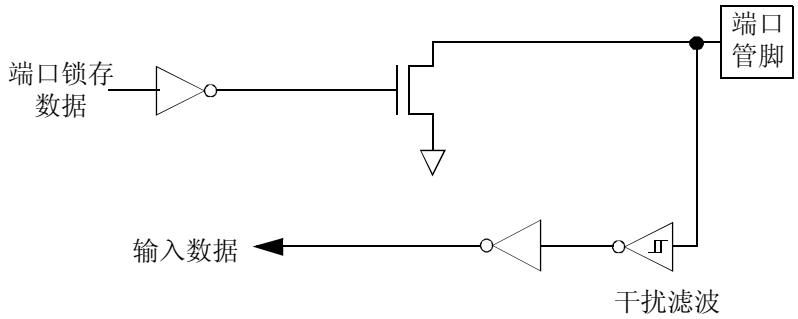


图 4.2 开漏输出

仅为输入配置

该配置如图 4.3 所示。它带有一个施密特触发输入口以及一个干扰抑制电路。

（请参考 P89LPC906/907/908 数据手册中 AC 电气特性的干扰滤波器部分的内容）。

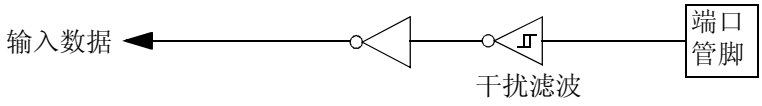


图 4.3 仅为输入

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驱动电流的情况。

推挽输出配置如图 4.4 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 P89LPC906/907/908 数据手册中 AC 电气特性的干扰滤波器部分的内容）。

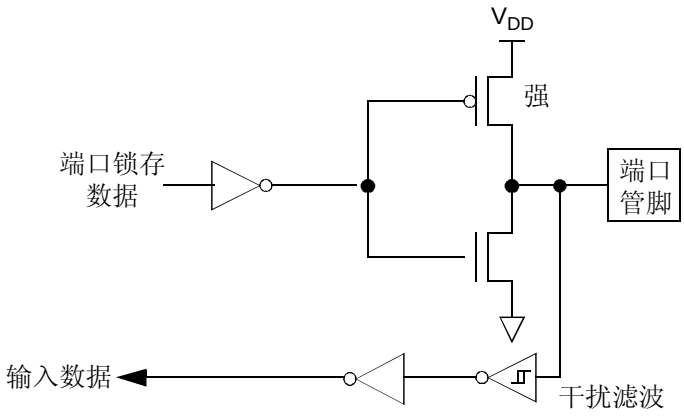


图 4.4 推挽输出

P0 口模拟功能

P89LPC906/907/908 集成了 1 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禁止数字输入和输出功能。

通过将端口设置成仅为输入（在 I/O 口配置中描述）模式来禁止数字信号输出（见表 4.2）。

P0 口的数字输入可通过 PT0AD 寄存器来禁止。复位后，PT0AD 的各位默认为 0 以使能管脚的数字功能。

表 4.3 端口输出配置－P89LPC906

口	配置寄存器位		可选功能	注
	PxM1.y	PxM2.y		
P0.4	P0M1.4	P0M2.4	KBI4,CIN1A	参考“P0 口模拟功能”来使用模拟输入（CIN1A 和 CMPREF）。
P0.5	P0M1.5	P0M2.5	KBI5,CMPREF	
P0.6	P0M1.6	P0M2.6	KBI6, CMP1	
P1.5	不可配置		$\overline{\text{RST}}$	仅为输入。通过用户配置位 RPD（UCFG1.6）选择作为通用输入口或复位输入口 $\overline{\text{RST}}$ 。 上电时总是作为复位输入口。
P3.0	P3M1.0	P3M2.0	XTAL2, CLKOUT	
P3.1	P3M1.1	P3M2.1	XTAL1	

表 4.4 端口输出配置－P89LPC907

口	配置寄存器位		可选功能	注
	PxM1.y	PxM2.y		
P0.4	P0M1.4	P0M2.4	KBI4,CIN1A	参考“P0 口模拟功能”来使用模拟输入（CIN1A 和 CMPREF）。
P0.5	P0M1.5	P0M2.5	KBI5,CMPREF	
P0.6	P0M1.6	P0M2.6	KBI6, CMP1	
P1.0	P1M1.0	P1M2.0	TxD	
P1.2	P1M1.2	P1M2.2	T0	
P1.5	不可配置		$\overline{\text{RST}}$	仅为输入。通过用户配置位 RPD（UCFG1.6）选择作为通用输入口或复位输入口。 上电时总是作为复位输入口。

表 4.5 端口输出配置－P89LPC908

口	配置寄存器位		可选功能	注
	PxM1.y	PxM2.y		
P0.4	P0M1.4	P0M2.4	KBI4,CIN1A	参考“P0 口模拟功能”来使用模拟输入（CIN1A 和 CMPREF）。
P0.5	P0M1.5	P0M2.5	KBI5,CMPREF	
P0.6	P0M1.6	P0M2.6	KBI6, CMP1	
P1.0	P1M1.0	P1M2.0	TxD	
P1.1	P1M1.1	P1M2.1	RxD	
P1.5	不可配置		$\overline{\text{RST}}$	仅为输入。通过用户配置位 RPD（UCFG1.6）选择作为通用输入口或复位输入口。 上电时总是作为复位输入口。

表 4.6：附加端口特性

上电后所有的管脚仅为输入模式。**请注意此配置不同于** LPC76x。

- 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。
- 每个 P89LPC906/907/908 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。请参考 P89LPC906/907/908 数据手册。

P89LPC906/907/908 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。

5. 定时器 0 和 1

P89LPC906/907/908 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。P89LPC907 的定时器 0 可配置为定时器或事件计数器（见图 5.1），另外增加了定时器溢出时 T0 脚自动翻转的功能选项。P89LPC907 的定时器 1 和 P89LPC906 和 P89LPC908 的定时器 0、定时器 1 只可用作定时器。

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 上每发生一次 1 到 0 的跳变时加 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 T0C/$\overline{\text{T}}$ 选择。P89LPC906 和 P89LPC908 的定时器 0 和定时器 1 以及 P89LPC907 的定时器 1 有 4 种工作模式（模式 0,1,2,3），由 TMOD 的两位（TnM1，TnM0）决定。模式 0、1、2 和 3 对于两个定时是一样的，模式 3 则不同。操作模式在后面详细描述。除此之外，P89LPC907 的定时器 0 还可工作在模式 6，该模式由 TAMOD 的 T0M2 模式位来决定。

TMOD 地址：89H

不可位寻址	7	6	5	4	3	2	1	0
复位源：任何复位	-	-	T1M1	T1M0	-	T0C/$\overline{T}$	T0M1	T0M0

复位值：00000000B

位	符号	功能
TMOD.7	-	保留。
TMOD.6	-	保留。
TMOD.5,4	T1M1,T1M0	定时器 1 的工作模式选择器。这些位用来决定定时器 1 的工作模式（见图 5-2）。
TMOD.3	-	保留。
TMOD.2	T0C/$\overline{T}$	定时器 0 的定时或计数功能选择。该位清零时定时器 0 用作定时功能（从 CCLK 输入脉冲）（P89LPC907）；在 P89LPC906 或 P89LPC908 中，该位必须写 0。
TMOD.1,0	T0M1,T0M0	定时器 0 的工作模式选择器。这两位用来决定定时器 0 的工作模式（见图 5-2）。在 P89LPC907 中，这两位和 TAMOD 寄存器的 T0M2 位一起来决定定时器 0 的工作模式（见图 5-2）。

图 5.1 定时器/计数器方式控制寄存器（TMOD）

TAMOD-P89LPC907 地址：8FH

不可位寻址	7	6	5	4	3	2	1	0
复位源：任何复位	-	-	-	-	-	-	-	T0M2

复位值：xxx0xxx0B

位	符号	功能
TAMOD.7-1	-	保留将来之用。用户程序不要将其置 1。
TMOD.0	T0M2	定时器 0 的工作模式选择位 2。它与 TMOD 寄存器的 T0M1 和 T0M0 位一起来决定定时器 0 的工作模式（P89LPC907）。
	TnM2-TnM0	定时器模式
	000	8048 定时器，“TLn”用作 5 位预分频器。（模式 0）
	001	16 位定时器/计数器。“THn”和“TLn”级联；无预分频器。（模式 1）
	010	8 位自动重装定时器/计数器。溢出时 THn 的内容装入 TLn。（模式 2）
	011	该模式下定时器 0 用作 2 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 控制位来控制。TH0 只可用作一个 8 位的定时器，由定时器 1 的控制位来控制（见文中描述）。该模式下定时器 1 关闭。（模式 3）
	100	保留。用户不要配置成该模式。
	101	保留。用户不要配置成该模式。
	110	PWM 模式（见“模式 6－P89LPC907”）。
	111	保留。用户不要配置成该模式。

图 5.2 定时器/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，即带 32 分频-预分频器的 8 位计数器。如图 5.4 所示。

在此模式中，定时器寄存器配置为 13 位寄存器。当计数值由全 1 变为全 0 时，置位定时器中断标志 TFn。当 TRn＝1，使能定时器的计数输入。TRn 是特殊功能寄存器 TCON（图 5.3）中的控制位。

该 13 位寄存器由 THn 的 8 位和 TLn 的低 5 位组成。TLn 的高 3 位不定，可忽略。置位运行标志（TRn）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 5.4。

模式 1 除了使用的是 16 位的寄存器（THn 和 TLn）外，其它与模式 0 相同，见图 5.5。

模式 2

在此模式中，定时器寄存器作为可自动重装的 8 位计数器（TLn），如图 5.6 所示。TLn 溢出不仅置位 TFn，而且将 THn 的值重新装入 TLn（THn 的值由软件预置）。重装时 THn 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 中定时器 1 停止工作。等效于使 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。定时器 0 模式 3 的逻辑如图 5-7 所示。TL0 使用定时器 0 的控制位： TR0 和 TF0。TH0 只用作定时功能（计数机器周期），占用定时器 1 的 TR1 和 TF1。因此，该模式下 TH0 控制“定时器 1”的中断。

模式 3 用于需要一个额外的 8 位定时器的场合。

注：当定时器 0 工作在模式 3 时，定时器 1 可通过进入/退出其自身的模式 3 来起动/关闭。它仍可用作串口的波特率发生器（P89LPC907,P89LPC908），或应用在不需要中断的任何场合。

模式 6－P89LPC907

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 5.8）。除以下几点外，其结构与模式 2 相同：

- TF0 由硬件置位和清零；
- TF0 的低电平宽度为 TH0 的值，必须在 1～254 之间，并且：
- TF0 的高电平宽度为 256－TH0。
- 向 TH0 装入 00H 使 T0 管脚强制为高电平，向 TH0 装入 FFH 使 T0 管脚强制为低电平。

注：TF0 上升沿时中断使能，并且 TF0 可象其它模式一样由软件清零。

TCON 地址：88H

不可位寻址

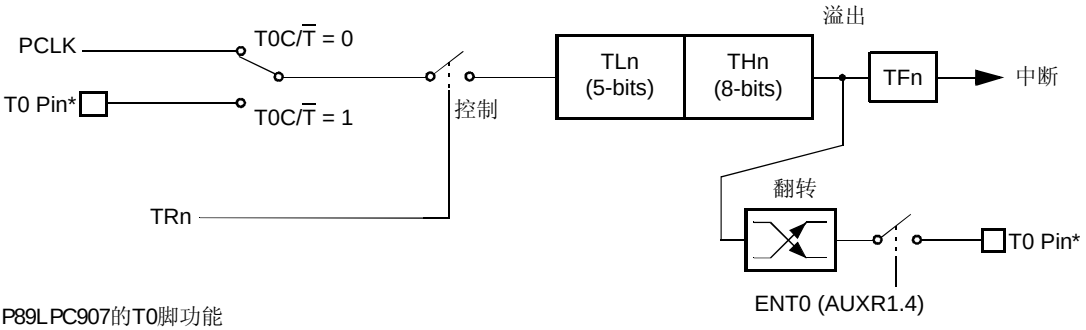
复位源：任何复位

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	-	-	-	-

复位值：00000000B

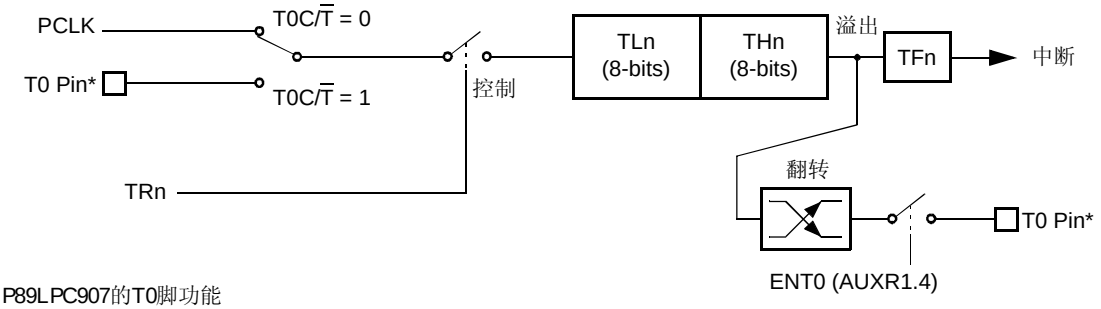
位	符号	功能
TCON.7	TF1	定时器 1 溢出标志。定时器溢出时由硬件置位。中断执行时硬件清零，或由软件清零。
TCON.6	TR1	定时器 1 运行控制位。通过软件将其置位/清零来起动/关闭定时器/计数器 1。
TCON.5	TF0	定时器 0 溢出标志。定时器/计数器溢出时由硬件置位。中断执行时硬件清零，或由软件清零（硬件清零时，模式 6 不同，见以上有关模式 6 的描述）。
TCON.4	TR0	定时器 0 运行控制位。通过软件将其置位/清零来起动/关闭定时器/计数器 0。
TCON.3	-	保留将来之用。用户不要将其置 1。
TCON.2	-	保留将来之用。用户不要将其置 1。
TCON.1	-	保留将来之用。用户不要将其置 1。
TCON.0	-	保留将来之用。用户不要将其置 1。

图 5.3 定时器/计数器控制寄存器（TCON）



* P89LPC907的T0脚功能

图 5.4 定时器/计数器 0 或 1 的工作模式 0（13 位计数器）



* P89LPC907的T0脚功能

图 5.5 定时器/计数器 0 或 1 的工作模式 1（16 位计数器）

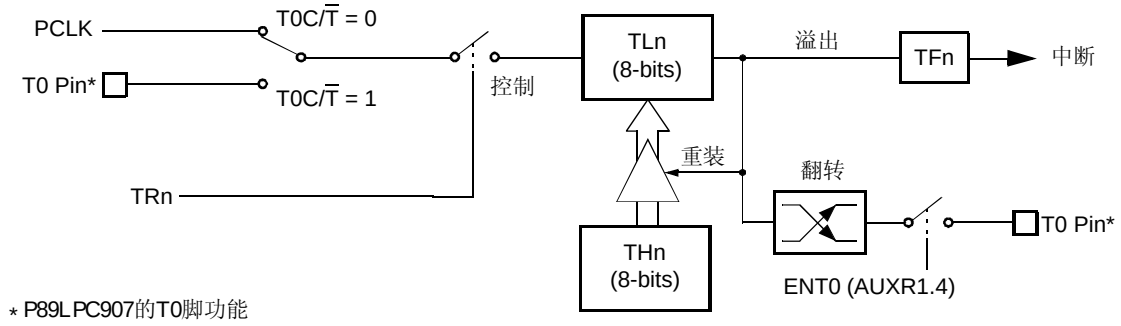


图 5.6 定时器/计数器 0 或 1 的工作模式 2（8 位自动重装）

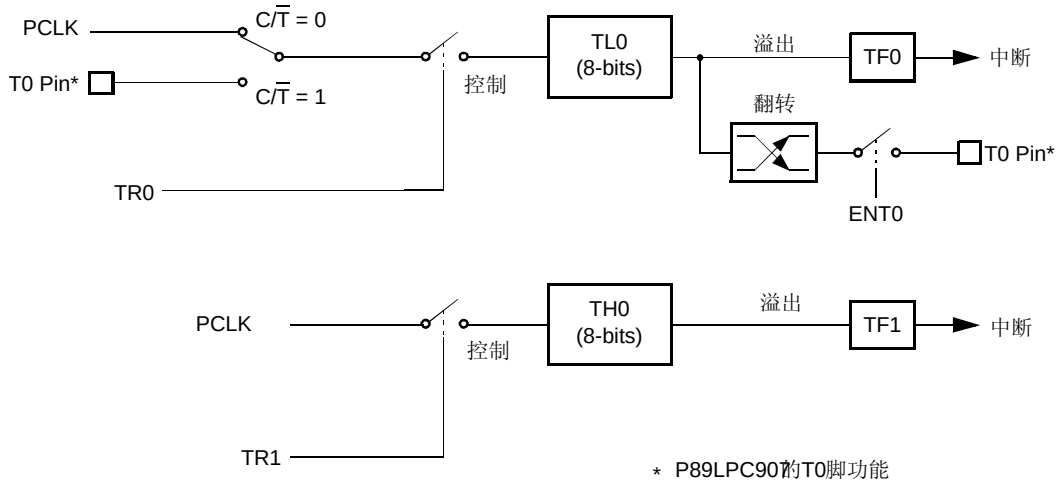


图 5.7 定时器/计数器 0 的工作模式 3（2 个 8 位计数器）

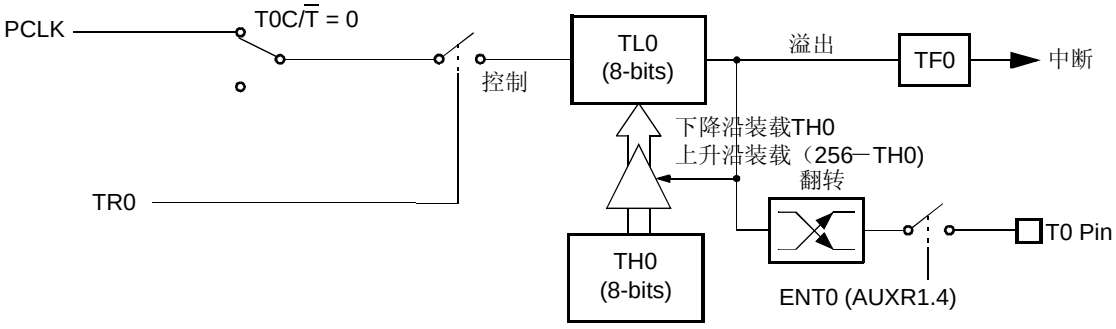


图 5.8 定时器/计数器 0 或 1 的工作模式 6（PWM 自动重装）,P89LPC907

定时器溢出触发输出－P89LPC907

定时器 0 可配置为发生定时器溢出时自动触发 T0 端口输出。此功能通过配置 AUXR1 寄存器的 ENT0 控制位来使能。打开该模式后，在首次定时器溢出之前端口的输出为逻辑 1。为了实现该功能，T0C/$\overline{T}$ 必须清零来选择 PCLK 作为定时器的时钟源。

6. 实时时钟/系统定时器

P89LPC906/907/908 具有一个简单的实时时钟/系统定时器。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 6.2）。实时时钟是一个 23 位的倒计数器。

实时时钟源

P89LPC906 该计数器的时钟源可以是 CCLK 或者 XTAL1-2 振荡器 （XCLK）。P89LPC907 和 P89LPC908 该计数器的时钟源是 CCLK。请参考“时钟”部分的图 2.3“振荡器控制方框图－P89LPC906”。CCLK 可使用 XTAL1-2 振荡器、内部 RC 振荡器或看门狗振荡器作为时钟源。如果 XTAL1-2 振荡器用于产生 CCLK，RTC 就使用 XTAL1-2 振荡器的输出或 CCLK 作为时钟源。可能的时钟组合如下面各表所示。

RTC 有 3 个相关的 SFR：

- 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）。

实时时钟/系统定时器通过置位 RTCEN（RTCCON.0）使能。实时时钟是一个 23 位的倒计数器（当 RTCEN=0 时初始化成全‘0’），它由 1 个 7 位的预分频器和 1 个 16 位的可装载倒计数器组成。当 RTCEN 写入‘1’时，计数器先装载（RTCH，RTCL,’1111111’），然后向下计数。当它为全‘0’时，计数器再次重装（RTCH,RTCL,’1111111’），并建立 RTCF（RTCCON.7）标志。

在实时时钟重装过程中，任何对 RTCH 和 RTCL 的写操作将导致重装失败。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。

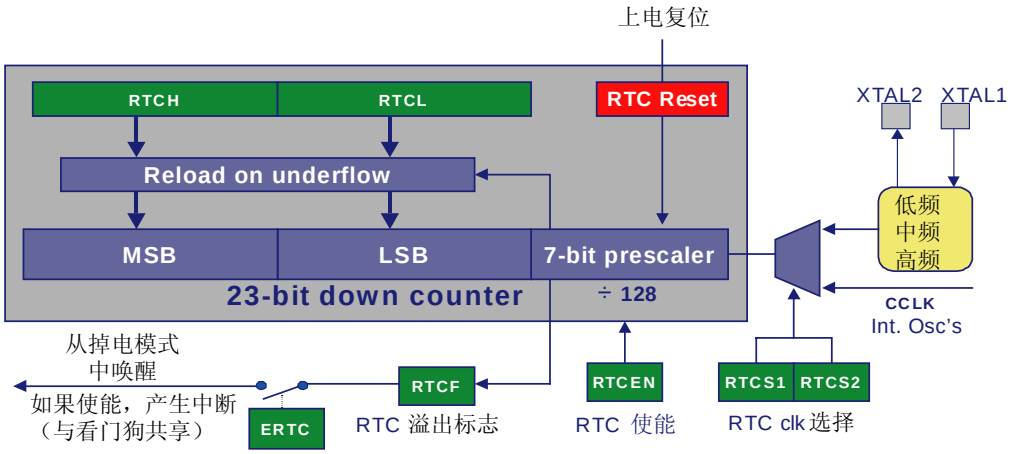


图 6.1 实时时钟/系统定时器框图

表 6.1 实时时钟/系统定时器时钟源－P89LPC906

FOSC2 (UCFG1.2)	FOSC1 (UCFG1.1)	FOSC0 (UCFG1.0)	RTCS1:0	CCLK 频率	RTC 时钟频率
0	0	0	00	高频晶振/DIVM	高频晶振（XCLK）
			01		
			10		
			11		高频晶振/DIVM（CCLK）

续上表

FOSC2 (UCFG1.2)	FOSC1 (UCFG1.1)	FOSC0 (UCFG1.0)	RTCS1:0	CCLK 频率	RTC 时钟频率
0	0	1	00	中频晶振/DIVM	中频晶振（XCLK）
			01		
			10		
			11		中频晶振/DIVM（CCLK）
0	1	0	00	低频晶振/DIVM	低频晶振（XCLK）
			01		
			10		
			11		低频晶振/DIVM（CCLK）
0	1	1	00	RC 振荡器/DIVM	高频晶振（XCLK）
			01		中频晶振（XCLK）
			10		低频晶振（XCLK）
			11		RC 振荡器/DIVM（CCLK）
1	0	0	00	WDT 振荡器/DIVM	高频晶振（XCLK）
			01		中频晶振（XCLK）
			10		低频晶振（XCLK）
			11		WDT 振荡器/DIVM（CCLK）
1	0	1	xx	未定义	
1	1	0			
1	1	1	00	外部时钟/DIVM	外部时钟（XCLK）
			01		
			10		
			11		外部时钟/DIVM（CCLK）

表 6.2 实时时钟/系统定时器时钟源－P89LPC907,P89LPC908

FOSC2 (UCFG1.2)	FOSC1 (UCFG1.1)	FOSC0 (UCFG1.0)	RTCS1:0	CCLK 频率	RTC 时钟频率
0	0	0	x	未定义	
0	0	1			
0	1	0			
0	1	1	00	RC 振荡器/DIVM	未定义
			01		
			10		
			11		RC 振荡器/DIVM（CCLK）
1	0	0	00	WDT 振荡器/DIVM	未定义
			01		
			10		
			11		WDT 振荡器/DIVM（CCLK）
1	0	1	xx	未定义	
1	1	0			
1	1	1			

RTCS 1-0 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。

影响实时时钟的复位源

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 6.1,6.2）
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	实时时钟使能。如果该位为 1，实时时钟使能。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 6.2 RTCCON 寄存器

7. 电源监控功能

P89LPC906/907/908 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。

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。也可通过置位 BOI(PCON.4)和 EBO(IEN0.5)来产生中断。

掉电检测的使能和禁止通过配置 BOPD(PCON.5)、位域 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，不论 PMOD1-0 和 BOPD 取值如何，掉电检测被禁止。如果 BOE 处于编程状态，由 PMOD1-0 和 BOPD 来决定掉电检测的使能和禁能。PMOD1-0 用来选择低功耗模式。当 PMOD1-0=‘11’，掉电检测电路停止工作以降低功耗。BOPD 默认为‘0’，只要 BOE 被编程，

掉电检测在上电时被使能。

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 电气特性）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 P89LPC906/907/908 器件的电源电压可以低于 2.7V，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 被编程，PMOD1-0≠‘11’，BOPD=0），不管复位或中断是否被使能，检测到掉电时都将使 BOF(RSTSRC.5)置位。BOF 一直保持置位直到通过软件写入‘0’将其清除。注：如果 BOE 未编程，BOF 就无意义。如果 BOE 被编程，内部复位时，BOF 和上电标志（POF-RSTSRC.4）一同被置位。

若要正确检测到掉电，V_{DD} 上升和下降时间必须符合一定规格。请参考 P89LPC906/907/908 的数据手册。

表 7.1 掉电检测选项

<table>
<tr><th>BOE
(UCFG1.5)</th><th>PMOD1-0
(PCON.1-0)</th><th>BOPD
(PCON.5)</th><th>BOI
(PCON.4)</th><th>EBO
(IEN0.5)</th><th>EA
(IEN0.7)</th><th>描述</th></tr>
<tr><td>0（擦除）</td><td>xx</td><td>x</td><td>x</td><td>x</td><td>x</td><td rowspan="2">掉电检测禁止。V_{DD} 操作电压范围为 2.4-3.6V。</td></tr>
<tr><td rowspan="6">1(已编程)</td><td>11
（完全掉电）</td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
<tr><td rowspan="5">≠11
(完全掉电外的其它模式)</td><td>1
(掉电检测关闭)</td><td>x</td><td>x</td><td>x</td><td>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。</td></tr>
<tr><td rowspan="4">0
(掉电检测有效)</td><td>0
(掉电检测产生复位)</td><td>x</td><td>x</td><td>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。</td></tr>
<tr><td rowspan="3">1
(掉电检测产生中断)</td><td>1
(使能掉电中断)</td><td>1
(全局中断使能)</td><td>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。</td></tr>
<tr><td>0</td><td>x</td><td rowspan="2">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 V_{DD} 跌落到掉电检测点时置位。BOF 可通过写入“0”清零。</td></tr>
<tr><td>x</td><td>0</td></tr>
</table>

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（RSTSRC.4）标志置位。POF 将会一直保持置位状态，直到通过软件将其清零。注：如果 BOE（UCFG1.5）被编程，POF 置位时 BOF（RSTSRC.5）置位。如果 BOE 未被编程，BOF 就没有意义。

节电模式

P89LPC906/907/908 支持 3 种不同的节电模式，由 PCON.1-0 决定（见表 7.2）。

表 7.2 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	掉电模式：掉电模式将振荡器停振以使功耗最小。 P89LPC906/907/908 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－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）。 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 V_{RAM}。这样将 RAM 内容保存为进入掉电模式时的状态。SFR 内容在 V_{DD} 低于 V_{RAM} 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 V_{DD} 必须上升到操作电压范围之内。 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： ● 掉电检测 ● 看门狗定时器（当 WDCLK(WDCON.0)为“1”时） ● 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） ● 实时时钟/系统定时器(及所使用的振荡器电路，RTCPD(PCONA.7)为“1”时除外)
1	1	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。注：此时不能产生掉电复位或中断。电压比较器中断和掉电中断都不可用作唤醒源。内部 RC 振荡器被禁能，除非 RC 振荡器用作系统时钟并且 RTC 被使能。 下面是所支持的唤醒源： ● 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器件。 ● 键盘中断 ● 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） 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。

PCON 地址：87H

不可位寻址

复位源：任何复位

7	6	5	4	3	2	1	0
SMOD1	SMOD0	BOPD	BOI	GF1	GF0	PMOD1	PMOD0

复位值：00000000B

位	符号	功能
PCON.7	SMOD1	当利用定时器 1 来产生波特率时，该位为串口（UART）双倍波特率位。为 1 时，UART 的波特率为定时器 1 的溢出速率。为 0 时，UART 的波特率为定时器 1 溢出速率/2。P89LPC907,P89LPC908（见图 8.2）
PCON.6	SMOD0	帧错误位置（P89LPC908）： －为 0 时，SCON.7 作为 UART 的 SM0 位 －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 8.3 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。）
PCON4	BOI	掉电检测中断使能。为 1 时，检测到掉电时产生中断；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“低功耗模式”）。

图 7.1 电源控制寄存器（PCON）

PCONA 地址：B5H

不可位寻址

复位源：任何复位

7	6	5	4	3	2	1	0
RTCPD	-	VCPD	-	-	-	SPD	-

复位值：00000000B

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	-	保留将来之用。
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	-	保留将来之用
PCONA.2	-	保留将来之用
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（P89LPC907,P89LPC908）。
PCONA.0	-	保留将来之用

图 7.2 电源控制寄存器（PCONA）

8. UART（P89LPC907, P89LPC908）

P89LPC907 和 P89LPC908 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。P89LPC906/907/908 还带有一个独立的波特率发生器。波特率可以选择由 CCLK（由一个常数分频），定时器 1 溢出或者独立的波特率发生器产生。

P89LPC907 不含 RxD 管脚，因此本节所描述的接收功能与它无关。P89LPC907 作为发送器运用在需要发送串行数据的场合。发送功能通常用于为调试过程提供信息。

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 种工作模式：

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 CCLK/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”）。

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 CCLK 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（逻辑 1）。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”）。

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 8.1 用于 UART 的 SFR 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

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（见图 8.2）。需要注意的是，如果SMOD1(PCON.7)清零，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。

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禁止（BRGCON 寄存器的 BRGEN 位为‘0’）时写入。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 8.2 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	CCLK/16
0	1	0	0	CCLK/(256－TH1)64
		1	0	CCLK/(256－TH1)32
		X	1	CCLK/((BRGR1,BRGR0)+16)
1	0	0	X	CCLK/32
		1	X	CCLK/16
1	1	0	0	CCLK/(256－TH1)64
		1	0	CCLK/(256－TH1)32
		X	1	CCLK/((BRGR1,BRGR0)+16)

BRGCON 地址： BDH

不可位寻址

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SBRGS	BRGEN

复位源：任何复位

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 8.2）
BRGCON.0	BRGEN	波特率发生器使能。只有当BRGEN为0时，才可对寄存器BRGR1和BRGR0进行写操作。

图 8.1 BRGCON 寄存器

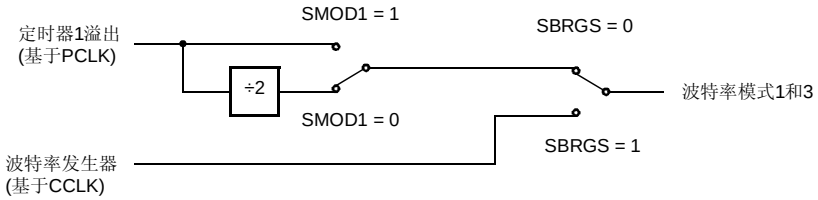


图 8.2 UART 波特率的产生（模式 1，3）

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。一旦检测到间隔条件，UART 将进入空闲模式并一直保持，直至接收到停止位。间隔检测还可通过置位 EBRR 位（AUXR1.6）对器件复位。

间隔检测复位将强制使程序计数器的高字节为引导向量值，低字节为 00h。该地址作为程序执行的首地址。

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：00000000B

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：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 8-2)
	1 0	2：9 位 UART　　CCLK/32 或 CCLK/16
	1 1	3：9 位 UART　　可变(见表 8-2)
SCON.5	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.4	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.3	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.2	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.1	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。
SCON.0	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（位 8）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 8.3　串口控制寄存器(SCON)

SSTAT 地址： BAH

不可位寻址

7	6	5	4	3	2	1	0
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。该位置位时使能双缓冲。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。注意：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	组合中断禁止。该位置位时使用各自独立的 Rx 和 Tx 中断。该位清零时使用 0 选择 Tx/Rx 组合中断（与传统 80C51 UART 相同）。该位复位为 0 时选择组合中断。
SSTAT.4	DBISEL	双缓冲发送中断选择－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=0）或结束（INTLO=1）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	间隔检测标志。当连续检测到 11 个位都为低电平时表明检测到一个间隔。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。该位置位时，FE, BR 或 OE 可产生中断。这些中断（向量地址 0023h）与 RI（CIDIS=1 时）或组合 TI/RI（CIDIS=0）共用中断。该位清零时，FE,BR,OE 不产生中断。(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。）注：如果 EBRR(AUXR1.6)置位，BR 可导致间隔检测复位。

图 8.4 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 8.5。

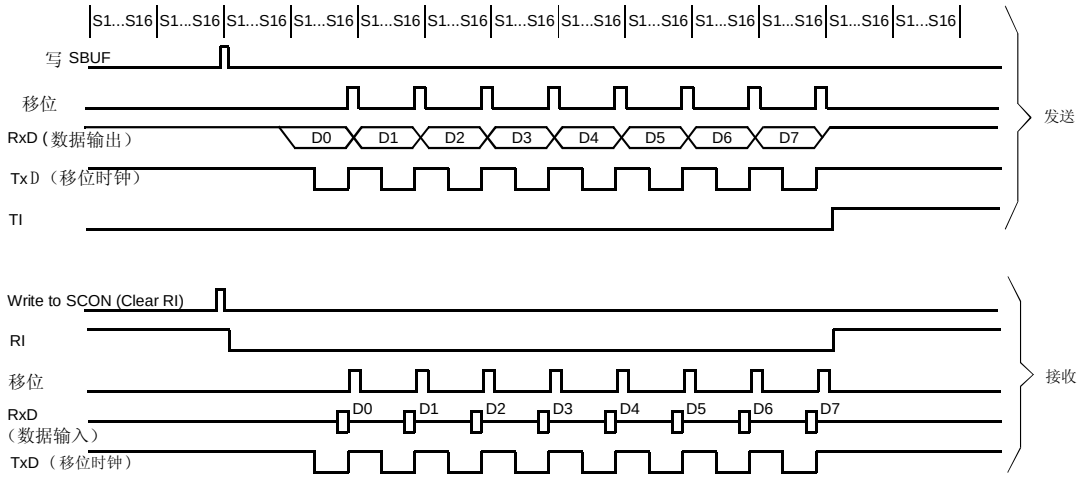


图 8.5 串口模式 0（双缓冲必须禁止）

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。每个位时间被分成 16 个计数状态。在第 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，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。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。

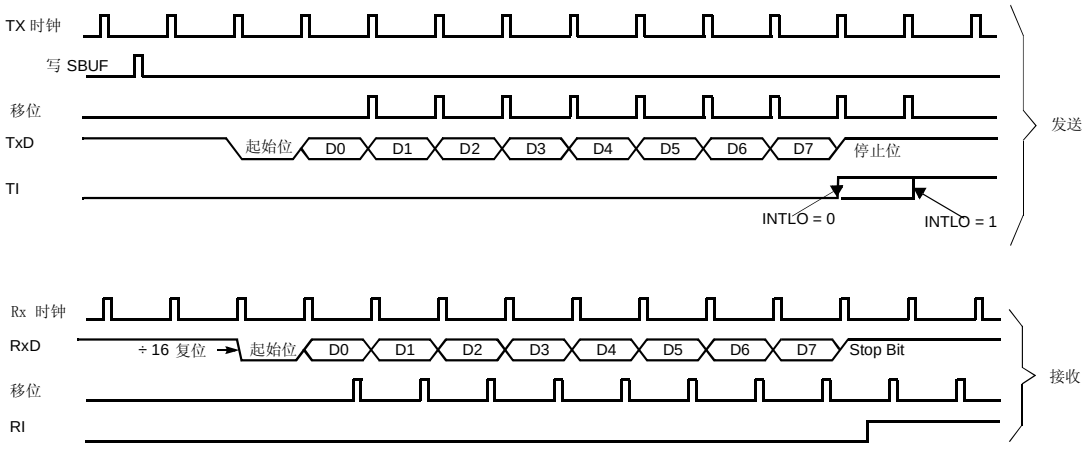


图 8.6 串口模式 1（只显示单发送缓冲的状况）

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，RI 不被置位。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。

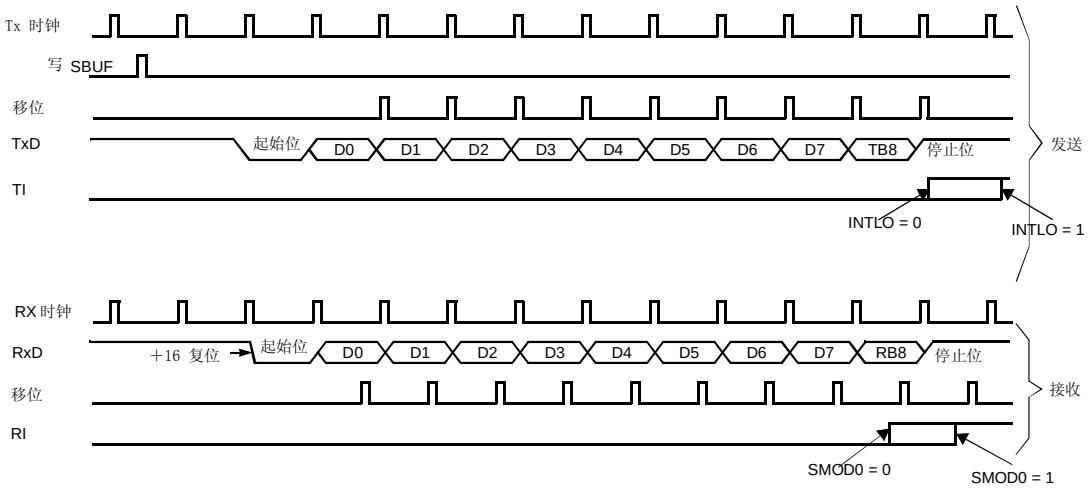


图 8.7 串口模式 2 或 3（只显示单发送缓冲的状况）

模式 2 和 3 中 SM2=1 时的帧错误和 RI

如果在模式 2 和 3 中 SM2=1，RI 和 FE 的状态如下表所示。

表 8.3 模式 2 和 3 中 SM2=1 时 FE 和 RI 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 8-7 相似，SMOD0=0，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 8-7 相似，SMOD0=1，RI 在停止位时发生	发生在停止位

间隔检测

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。间隔检测在状态寄存器（SSTAT）中报告。对于模式 1，间隔包括起始位，8 个数据位和 2 个停止位。对于模式 2 和模式 3，间隔包括起始位，9 个数据位和 1 个停止位。间隔检测位通过软件或复位清零。如果 UART 使能，并且 EBRR 位（AUXR1.6）被置位时发生间隔检测，可复位器件。

双缓冲

UART 具有一个发送双缓冲器，这就允许在发送第一个字符的时候向 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相邻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,即 SSTAT.7=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，这样就会出现无法判断 Tx 中断是否已经产生，UART 无法得知是否有后续数据等待发送的不确定状况。

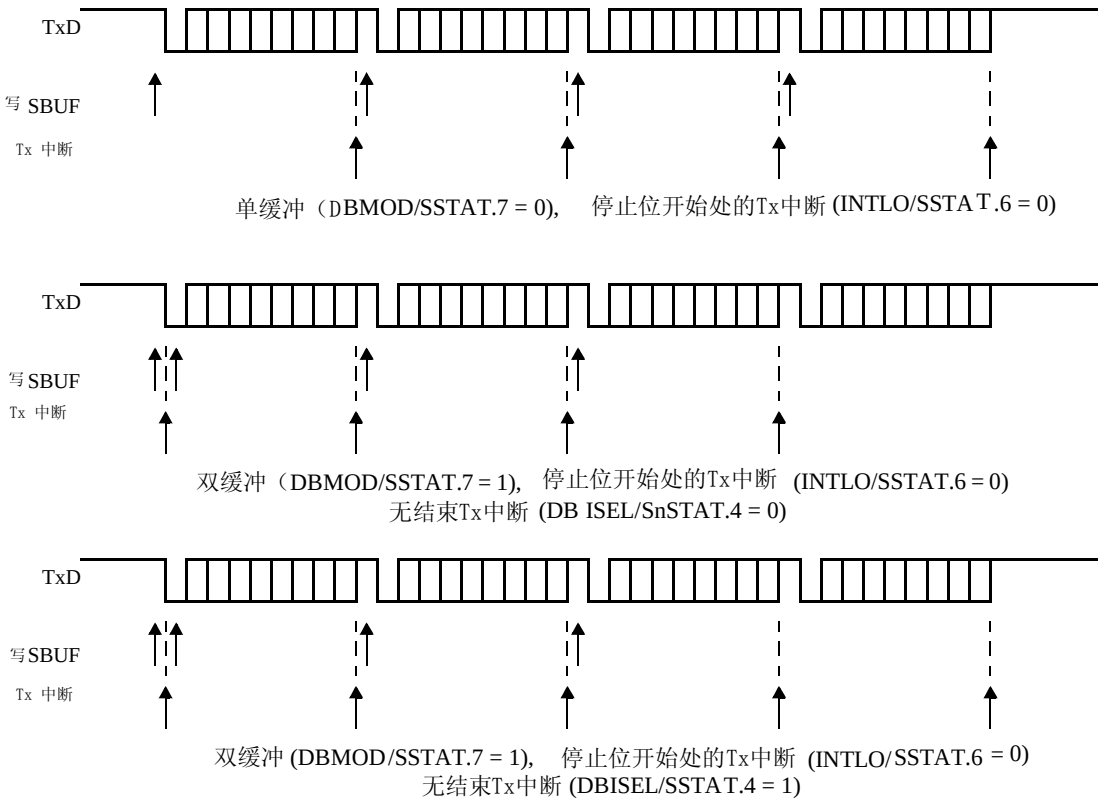


图 8.8 带双缓冲的发送和不带双缓冲的发送

双缓冲中的第 9 位（位 8）数据（模式 1，2 和 3）

如果双缓冲被禁止（DBMOD，即 SSTAT.7=0），对 TB8 的写操作可以在写入 SBUF 之前或之后进行，

只要在 TB8 被移出之前将其更新即可。在 TB8 移出（通过 Tx 中断指示）之前不要改变其值。

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：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，这样就会出现无法判断 Tx 中断是否已经产生，UART 无法得知是否有后续数据等待发送的不确定状况。

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 1 的位 0 必须为 1 来排除从机 0，从机 1 唯一的地址应当为 1100 0001。通过给定一个位 0＝0（从机 0）和位 1＝0（从机 1）的地址可同时选择两个从机，从而取地址 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=1111 1001
给定地址=1100 0XX0

从机 1 SADDR=1110 0000
SADEN=1111 1010
给定地址=1110 0X0X

从机 2 SADDR=1110 0000
SADEN=1111 1100
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 和 0101 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。

9. 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，P1.5 用作一个输入管脚。

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。在上电过程中，连接到该管脚的外部电路不应将其拉低，否则将使器件一直处于复位状态。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。

注：为了确保上电复位的产生，电源周期内，电源使用前必须有一个 V_{DD} 下降到低于 V_{POR} 的过程（见数据手册的“DC 电气特性”）。

复位可由下列复位源引起（见图 9.1）：

- 外部复位管脚（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 间隔字符检测复位（P89LPC908）

每一个复位源在复位寄存器 RSTSRC 中都有一个对应的标志。用户可读取该寄存器以判断最近的复位源是哪一个。这些标志位可通过软件写入“0”清零。可以有多于一个的标志位置位：

- 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，之前置位的标志位不会受到影响

上电复位后代码的执行

P89LPC906/907/908 包含 2 个特殊的 Flash 单元：引导向量和引导状态位。复位后，P89LPC906/907/908 检查引导状态位的内容。如果引导状态位为 0，上电后程序从 0000H 单元开始执行，0000H 单元也通常是用户应用程序的起始地址。如果引导状态位为 1，将引导向量的内容作为执行地址的高字节，低字节设为 00H。出厂默认设置为 00H。UART 间隔检测复位（P89LPC908）的效果等同于非零的引导状态位。

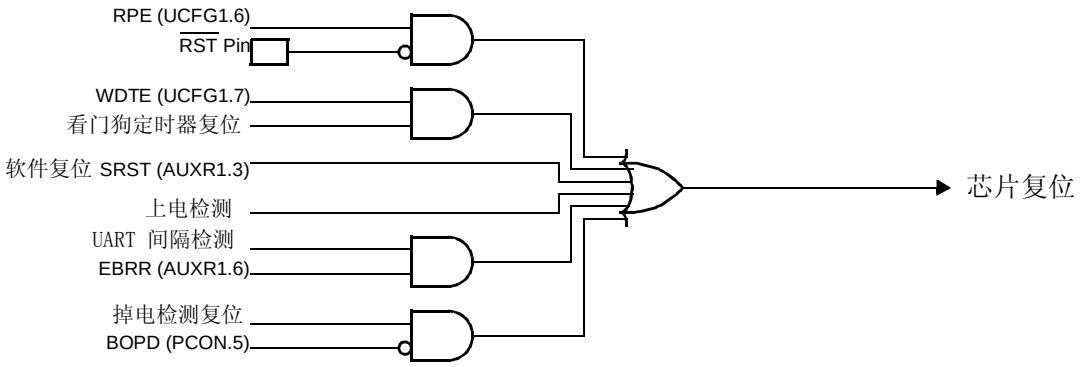


图 9.1 复位方框图

RSTSRC 地址：DFH

不可位寻址	7	6	5	4	3	2	1	0
复位源：上电复位	-	-	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。（P89LPC908）
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 9.2 复位源寄存器

10. 模拟比较器

P89LPC906/907/908 有 1 个模拟比较器。当正向输入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。

比较器连接方式如图 10-2。比较器的最低工作电压为 V_{DD}=2.4V。

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。

比较器配置

每个比较器控制寄存器 CMP1 如图 10.1 所示。比较器可能的配置如图 10.3 所示。

CMP1 地址：ACH

不可位寻址	7	6	5	4	3	2	1	0
复位源：任何复位	—	—	CE1	-	CN1	OE1	CO1	CMF1

复位值：xx000000B

位	符号	功能
CMP.7,6	—	保留将来之用。
CMP.5	CE1	比较器使能位。当置位时，对应的比较器使能。CE1 置位 10 微秒后比较器输出有效值。
CMP.4	—	保留将来之用。
CMP.3	CN1	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。
CMP.2	OE1	输出使能，为‘1’时，如果比较器使能（CE1＝1），比较结果输出连接到 CMP1 管脚。此输出和 CPU 时钟不同步。
CMP.1	CO1	比较器输出，和 CPU 同步以允许软件进行读取。
CMP.0	CMF1	比较器中断标志。当比较器输出 CO1 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 10.1 比较器控制寄存器（CMP1）

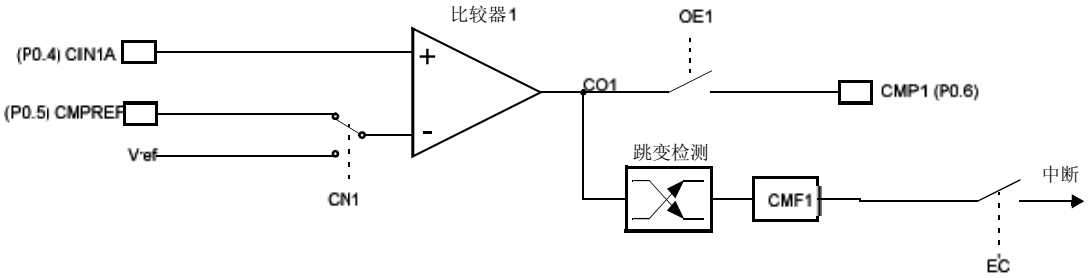


图 10.2 比较器输入和输出的连接

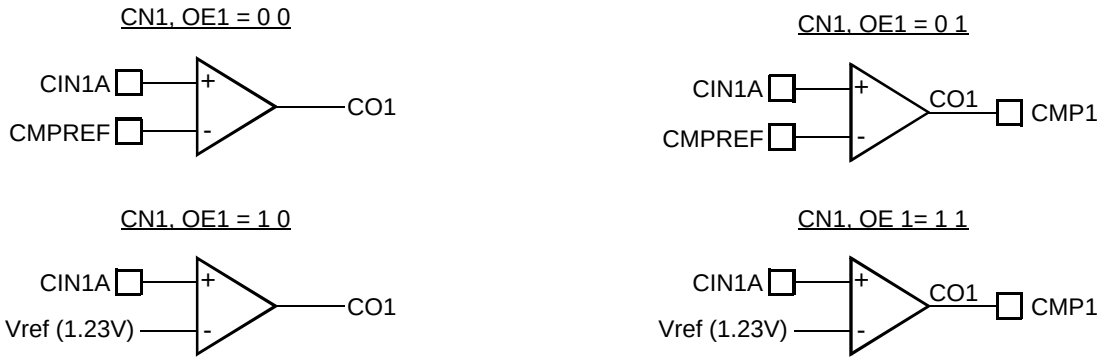


图 10.3 比较器配置

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。请参考数据手册。

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 CMF1。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。如果 IEN1 寄存器的中断使能位 EC 置位且通过配置 IEN0 寄存器的 EA 位使中断系统使能，可产生中断。

当比较器被禁止时，比较器的输出 COx 变为高电平。如果比较器在输出为低电平时被禁止，比较器的输出就由低变高，这样将使比较器标志 CMFx 置位。如果比较器中断被使能，将产生中断。因此，用户必须在比较器被禁止前关闭比较器中断，或者，在比较器被禁止后清除比较器标志 CMFx。

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的功耗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。若要降低功耗，用户可通过置位 PCONA.5 来禁止比较器，或将器件设置为完全掉电模式。

比较器配置举例

下面是一段比较器初始化的程序。比较器 1 配置如下： CMPREF 为输入，CMP 为比较器输出，比较器输出改变时产生中断。

```
CMPINIT:
    MOV     PT0AD,#030h     ；禁止 CIN,CMPREF 的数字输入功能，使能其模拟功能
    ANL     P0M2,#0CFh      ；禁止 CIN,CMPREF 的数字输出功能，
    ORL     P0M1,#030h      ；用于模拟功能：CIN, CMPREF
    MOV     CMP1,#024h      ；开启比较器，并进行如下设置：
                            ；－CMPREF 为反向输入。
                            ；－比较结果输出到 CMP。
    CALL    delay10us       ；比较器启动后至少 10us 方可使用
    ANL     CMP1,#0FEh      ；清除比较器的中断标志。
    SETB    EC              ；使能比较器中断。中断优先级不变。
    SETB    EA              ；使能中断系统（如果有必要）
    RET                     ；返回调用处。
```

比较器中断服务程序返回前必须先清除中断标志（此例中为 CMF1）。

11. 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。如果通过置位 IEN1 寄存器的 EKBI 位和 EA 位将中断使能，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。

为了将键盘中断设置为类似 87LPC76x 的 KBI 功能，用户必须设置 KBPATN=0FFH 和 PATN_SEL=0（不相等），这样由 KBMASK 寄存器使能的任何连接到 P0 口管脚按键都将使硬件置位 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.6	KBPATN.5	KBPATN.4	-	-	-	-

复位值：11111111B

位	符号	功能
KBPATN.6,5,4	—	模式位 6,5,4。

图 11.1 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 11.2 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.6	KBMASK.5	KBMASK.4	-	-	-	-

复位值：00000000B

位	符号	功能
KBMASK.7	—	保留。
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.3:0	—	保留。

注：为了使 KBMASK 寄存器的设置有效，键盘中断必须被使能。
KBMASK.7,KBMASK.3,KBMASK.2,KBMASK.1 和 KBMASK.0 通常被写入‘0’。

图 11.3 键盘中断屏蔽寄存器（KBM）

12. 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复位实现复位。

看门狗功能

WDCON 和 UCFG1 寄存器控制着 WDT 的开/关、WDT 的时钟源、预分频值和溢出时是否使能 WDT 使器件复位。另外，WDT 还含有一种安全机制，可利用 UCFG1 的配置值通过 IAP 或商业编程器来强制使能 WDT。

如果 WDTE 位（UCFG1.7）被置位，使能 WDT 溢出时对器件的复位。复位后，不管 WDTE 位是何种状态，WDT 将继续运行。

通过 WDRUN 位（WDCON.2）的置位和清零来分别起动和关闭 WDT。复位后该位被置位，WDT 保持运行状态。每次写 WDCON 操作后都跟随一个清零序列（见后面的“清零序列”）。用户可以通过 WDCON 的其它位来选择 WDT 的时钟源和预分频值。

当未使能看门狗溢出对器件的复位时，如果需要，WDT 可工作在“定时器模式”并被使能产生中断（IEN0.6）。

看门狗安全使能位 WDSE（UCFG1.4）和 WDTE 一起，可在上电时使看门狗强制进入某种工作状态。详情请参考表 12.1。

表 12.1 看门狗定时器配置

WDTE (UCFG1.7)	WDSE (UCFG1.4)	功能
0	X	看门狗复位禁止。定时器可作为内部定时器并可用于产生中断。此时，WDSE 的值无影响。
1	0	看门狗复位使能。用户通过设置 WDCLK 来选择时钟源。
1	1	看门狗复位使能，还具有以下安全特性： 1. WDCLK 强制为 1（使用看门狗振荡器） 2. WDCON 和 WDL 寄存器只可写一次 3. WDRUN 强制为 1 并且不能通过软件清除。

图 12.3 所示为看门狗定时器工作在看门狗模式。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（递减）。预分频器的时钟源可选择 PCLK 或看门狗振荡器，由 WDCON 寄存器的 WDCLK 位决定。（注：时钟源的切换不会立即生效－见“看门狗时钟源”。）

当看门狗发生向下溢出且复位被使能时，看门狗将请求执行复位。当看门狗复位使能时，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。

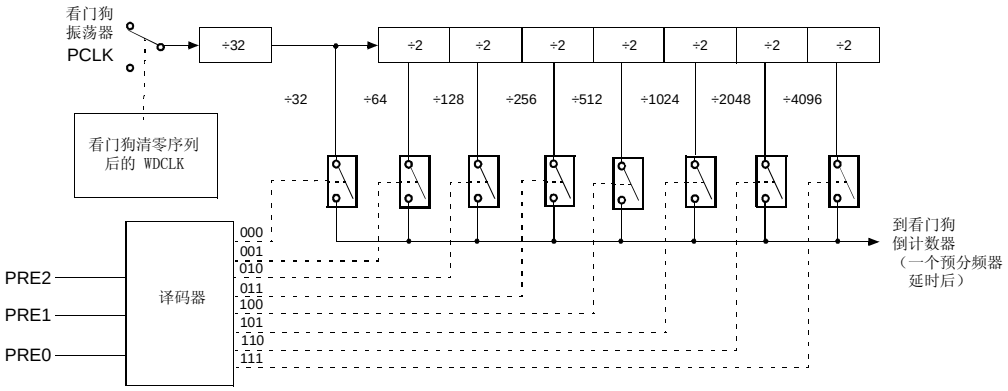


图 12.1 看门狗预分频器

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 12.3）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，看门狗定时器必须在定时溢出前进行清零（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 P89LPC906/907/908 中断系统已经使能且有可能在看门狗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 位倒计数器，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 WD_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 位倒计数器时需要执行清零序列。

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-PRE0	时钟预分频器节拍选择，详见表 12.2。
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（看门狗运行）。
WDCON.1	WDTOF	看门狗溢出标志。该位在 8 位倒计数器溢出时置位。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“掉电操作”）。（注：如果 WDTE 和 WDSE 都置位，该位被强制为 1。）详见“看门狗时钟源”。

图 12.2 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：

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

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：

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

tclks 的最大个数为：

$tclks=(2^{(5+7)})(255+1)+1=1,048,577$

下表所示为 P89LPC906/907/908 看门狗溢出值的取样。

表 12.2 P89LPC906/907/908 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	8,193	20.5ms	1.37 ms
001	0	65	162.5μs	10.8μs
	255	16,385	41.0ms	2.73 ms
010	0	129	322.5μs	21.5μs
	255	32,769	81.9ms	5.46 ms
011	0	257	642.5μs	42.8μs
	255	65,537	163.8ms	10.9 ms
100	0	513	1.28ms	85.5μs
	255	131,073	327.7ms	21.8 ms

续上表

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)
101	0	1,025	2.56ms	170.8μs
	255	262,145	655.4ms	43.7 ms
110	0	2,049	5.12ms	341.5μs
	255	524,289	1.31s	87.4 ms
111	0	4097	10.2ms	682.8μs
	255	1,048,577	2.62s	174.8 ms

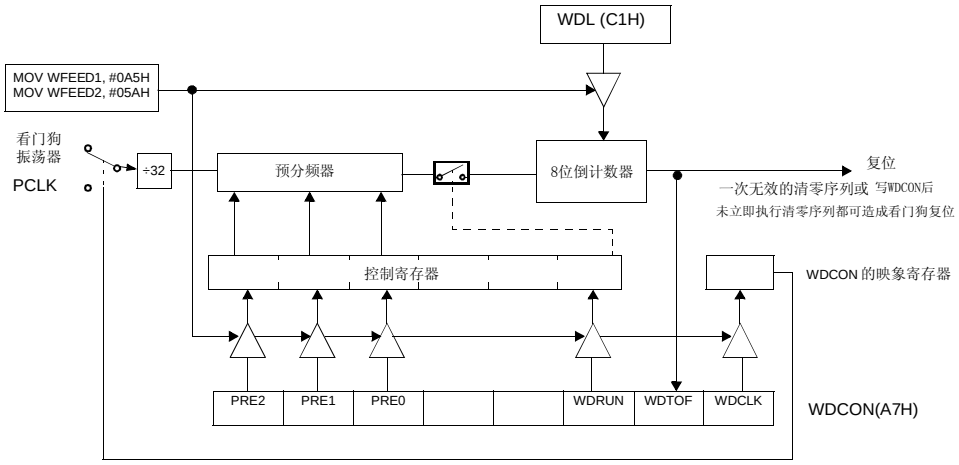


图 12.3 看门狗模式中的看门狗定时器（WDTE=1）

定时器模式中的看门狗定时器

图 12.4 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。定时器溢出将置位 WDTOF 位。如果 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。看门狗溢出时，WDL 的内容重新装入倒计数器并且倒计数器立即重新开始计数。

溢出前，将 WDL 装入倒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。

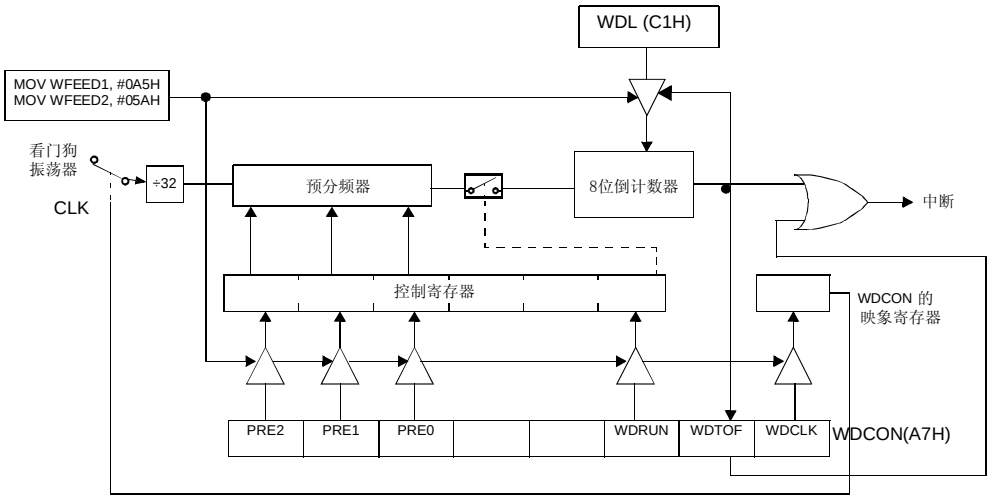


图 12.4 定时器模式中的看门狗定时器（WDTE=0）

掉电操作

掉电模式下，只要 WDT 振荡器用作 WDT 的时钟源，WDT 振荡器将继续运行，消耗大约 50uA 的电流。如果选择 PCLK 作为 WDT 的时钟源，WDT 振荡器将和其它器件一同进入掉电模式（见“看门狗时钟源”）。掉电模式中，PCLK 停止运行，导致看门狗被禁止。

看门狗时钟源

看门狗定时器系统片内包含一个 400kHz 的振荡器。通过配置看门狗控制寄存器 WDCON 的 WDCLK 位来选择看门狗振荡器或 PCLK（参考图 12.1）作为看门狗的时钟源。如果看门狗特性被使能，为了防止看门狗溢出时复位 CPU，必须通过软件将其清零。

在改变 WDCLK(WDCON.0)之后，对时钟源的切换不会立即生效。如图 12.3 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。

由于清零序列结束后预分频器立刻开始计数，切换时钟会造成预分频器的计数误差。此误差为 2 个旧时钟周期加 2 个新时钟周期。

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。

不使用外部振荡器的掉电周期唤醒

在不使用外部振荡器的条件下，周期唤醒所需功耗取决于用来产生唤醒信号的内部振荡器的功耗。内部 RC 振荡器可用作实时时钟的时钟源，它消耗的电流大约为 300uA。如果用 WDT 代替 RC 振荡器来产生中断，消耗的电流可降至 50uA。WDT 溢出时可将器件唤醒。

13. 附加特性

AUXR1 寄存器包含了几个特殊用途的控制位，这些位分别与几种芯片功能相关联。如图 13.1 所示。

AUXR1 地址：A2H

不可位寻址

复位源：任何复位

7	6	5	4	3	2	1	0
CLKLP	EBRR	-	-	SRST	0	-	DPS

复位值：000000x0B

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时钟频率高达 12MHz 的操作（P89LPC906）。
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（P89LPC908）。在 P89LPC906 和 P89LPC907 中，对 AUXR1 寄存器执行写操作时，该位被写入‘0’。
AUXR1.5	-	保留。
AUXR1.4	-	保留。
AUXR1.3	SRST	软件复位。当 SRST 通过软件置位时，P89LPC906/907/908 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 AUXR1 加 1 来使 DPS 位翻转而不影响寄存器的其它位。
AUXR1.1	–	未用。允许设置成“1”。
AUXR1.0	DPS	数据指针选择。选择两个数据指针中的一个。

图 13.1 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，使处理器彻底复位。当第 3 位为 1 的数据被写入 AUXR1 时，所有 SFR 将被初始化，程序重新由 0000 处开始执行。因此，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

CPU 可利用双数据指针（DPTR）来指向某条指令的确定地址。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 数据指针加 1
- JMP @A+DPTR 跳转到 DPTR 的相对间接地址
- MOV DPTR,#data16 数据指针装入一个 16 位的常数
- MOVC A,@A+DPTR 将 DPTR 相对地址的代码字节移入累加器
- MOVX A,@DPTR 将 DPTR 指向的数据存储单元的数据字节移入累加器
- MOVX @DPTR,A 将累加器的数据字节移入 DPTR 指向的数据存储单元

而且任何读取或操纵 DPH 和 DPL 寄存器（当前 DPTR 的该字节和低字节）的指令都受 DPS 位设置的影响。MOVX 指令在 P89LPC906/907/908 中的应用受到限制，因为它们没有外部数据总线。但是，它们可用来访问 Flash 配置信息（见“Flash 配置”）。

AUXR1 的位 2 永远都是逻辑 0。这样 DPS 位可以简单地通过将 AUXR1 加 1 来翻转（由此来切换数据指针），而不会改变寄存器的其它位。

14. Flash 程序存储器

概述

P89LPC906/907/908 Flash 存储器提供电路中的电擦除和编程。Flash 可以字节为单位进行读取和写入。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。存储单元的设计优化了擦除和编程结构。P89LPC906/907/908 使用 V_{DD} 电压来执行编程/擦除算法。另外，利用商业编程器的串行编程为实现在电路编程提供一个简单接口。P89LPC906/907/908 Flash 存储器甚至在经过 100, 000 次擦除和编程之后仍然能可靠地保存存储器的内容。

特性

- IAP-Lite 允许代码存储器的单个或多个字节用于数据存储。
- 可在整个操作电压范围内执行编程和擦除。
- 使用 IAP-Lite 进行读/编程/擦除。
- 任意 Flash 编程/擦除时间小于 2ms（4ms 擦除/编程）。
- 使用工业标准的商用编程器进行在电路串行编程。
- 可对每一个 Flash 扇区进行编程加密。
- 每个字节至少可执行 100,000 次擦除/编程。
- 256 字节的扇区长度，16 字节的页规格。
- 数据至少可保存 10 年。

IAP-Lite 简介

P89LPC906/907/908 支持 IAP-Lite 编程和擦除功能。未加密扇区中的任何字节都可通过 MOVC 指令访问，因此，未加密扇区可用作非易失性数据存储器。另外，用户还可访问其它 flash 单元，诸如 UCFG1、引导扇区、状态位、加密字节和标识字节。访问上述 flash 单元的方法与访问用户程序存储器稍微有些不同。

Flash 作为数据存储器使用

IAP-Lite 提供了一种程序擦除功能，通过一次操作完成一页内一个或多个字节的擦除和编程而不影响该页的其它字节。IAP-Lite 功能在微控制器的固件控制下得以实现，通过使用 4 个 SFR 和 1 个内部 16 字节“页寄存器”来完成对未加密扇区的擦除和编程。这些 SFR 为：

- FMCON（Flash 控制寄存器）。读时用作状态寄存器；写时用作命令寄存器。注：写入命令时状态位被清零。
- FMDATA（Flash 数据寄存器）。存放要装入页寄存器的数据。
- FMADRL，FMADRH（Flash 存储器地址低字节，Flash 存储器地址高字节）。用来指示页寄存器的字节地址或用户程序存储器的页。

页寄存器包括 16 个字节，每个字节都有一个更新标志。当将一个 LOAD 命令写入 FMCON 时，页寄存器和所有更新标志都被清零。当 FMDATA 被写入数据时，写入的值存放在 FMADRL 低 6 位指向的页寄存器单元。同时，相应单元的更新标志置位。FMADRL 自动增加到下一个单元。在页寄存器的最后一个字节被写入后，FMADRL 重新指向页寄存器的第一个字节，但不影响 FMADRL[7:4]的值。此时，不能继续向页寄存器写入数据了。通过在写入 FMDATA 前改变 FMADRL 的内容可将任何字节单元装入页寄存器。但是，每发布一个 LOAD 命令，页寄存器的每个单元只能被写入一次。不要试图对一个页寄存器单元执行多次写操作。

FMADRH 和 FMADRL[7:4]用来选择执行擦除-编程功能的程序存储器页。当向 FMCON 写入擦除-编程命令后，页寄存器中被更新单元对应在程序存储器页的单元的所有内容被清除，页寄存器中相应单元的内容被编程到程序存储器页中。只有用户程序阵列中装入页寄存器的字节才可被擦除和编程，用户程序存储器的其它字节不受影响。

向 FMCON 写入擦除-编程命令（68H）将起动擦除-编程过程并使 CPU 进入编程-空闲状态。CPU 会一直保持这种空闲状态直到擦除-编程周期结束或被一个中断终止。从编程-空闲状态退出后，FMCON 的内容是该周期的状态信息。

如果在擦除/编程周期内有中断产生，擦除/编程操作被终止，FMCON 的 OI 标志（操作被中断）置位。如果某一应用允许在擦除-编程过程中产生中断，可通过用户程序在每次擦除-编程操作后检查 OI 标志（FMCON.0）来判断操作是否被终止。一旦擦除-编程操作被终止，用户程序需要重复整个过程来起动页寄存器的装载。

不管有多少个字节装入页寄存器，擦除-编程周期的时间为 4ms。

程序存储器中一个字节（或多个字节）的擦除-编程操作步骤如下：

- 向 FMCON 写入 LOAD 命令（00H）。LOAD 命令将清除页寄存器的所有单元及相应的更新标志。
- 将页寄存器内的地址写入 FMADRL。由于 FMADRL[5:0]用来装载页寄存器地址，FMADRH 和 FMADRL[7:4]用来装载擦除-编程命令，所以，用户此时可写入页寄存器内的字节单元（FMADRL[3:0]）和程序存储器页地址（FMADRH 和 FMADRL[7:4]）。
- 将要编程的数据写入 FMDATA。这可让 FMADRL 增加，使其指向页寄存器的下一个字节。
- 如果需要的话，将要编程的下个字节的地址写入 FMADRL（如果是连续字节就不需要，因为 FMADRL 是自动递增的）。所有被编程的字节必须在同一页中。

- 将要编程的下一字节数据写入 FMDATA。
- 重复 FMADRL 和/或 FMDATA 的写操作，直至所有要编程的字节都被装入页寄存器。
- 如果地址单元不属于之前写入的页寄存器地址 FMADRL[3:0]之列，将用户程序存储器的页地址写入 FMADRH 和 FMADRL[7:4]。
- 向 FMCON 写入擦除-编程命令（68H），起动擦除-编程周期。
- 读 FMCON 来检查状态。如果操作被终止，重新通过 LOAD 命令来起动操作。

装载页寄存器和执行一次擦除/编程操作的汇编程序如图 14.2 所示。类似的 C 语言程序如图 14.3 所示。

FMCON 地址：E4h

不可位寻址	7	6	5	4	3	2	1	0
复位源：任何复位	-	-	-	-	HVA	HVE	SV	OI
复位值								

位	符号	功能
FMCON.7-4	-	保留。
FMCON.3	HVA	高电压终止。当在编程或擦除周期内检测到中断或掉电以及在编程或擦除周期开始时掉电检测被禁止时置位该位。
FMCON.2	HVE	高电压错误。高电压发生器出错时该位置位。
FMCON.1	SV	安全出错。试图对加密扇区或页进行编程、擦除或 CRC 校验时该位置位。
FMCON.0	OI	操作被中断。由于中断或复位而使擦除-编程终止时该位置位。

图 14.1 Flash 存储器控制寄存器

```
; * 输入：                                    *
; *       R3 = 编程字节数（字节）              *
; *       R4 = 页地址 MSB（字节）              *
; *       R5 = 页地址 LSB（字节）              *
; *       R7 = RAM 数据缓冲区指针（字节）      *
; * 输出：                                    *
; *       R7 = 状态（字节）                    *
; *       C = 无错时清零，有错时置位           *
LOAD    EQU     00H
EP      EQU     68H
PGM_USER:
        MOV     FMCON,#LOAD             ; 装载命令，清除页寄存器
        MOV     FMADRH,R4               ; 写入高地址
        MOV     FMADRL,R5               ; 写入低地址
        MOV     A,R7                    ;
        MOV     R0,A                    ; 指针装入 R0
LOAD_PAGE:
        MOV     FMDAT,@R0               ; 数据写入页寄存器
        INC     R0                      ; 指向下一个字节
        DJNZ    R3,LOAD_PAGE            ; 一直操作直至编程字节数为零
        MOV     FMCON,#EP               ; 否则擦除&编程页

        MOV     R7,FMCON                ; 备份返回的状态
```

```
        MOV     A,R7                    ；读状态
        ANL     A,#0FH                  ；保存低四位
        JNZ     BAD                     ；
        CLR     C                       ；如果操作无误，清除错误标志
        RET                             ；返回
BAD:
        SETB    C                       ；设置错误标志
        RET                             ；返回
```

图 14.2 擦除/编程（部分）页的汇编程序

```
unsigned  char  idata  dbytes [16];                    //数据缓冲区
unsigned  char  Fm_stat;                               //状态结果
bit  PGM_USER (unsigned  char , unsigned  char );
bit  prog_fail;

void  main ( )
{
        prog_fail = PGM_USER ( 0x1F ,0xC0 );
}

bit  PGM_USER (unsigned  char  page_hi , unsigned  char  page_lo )
    {
        # define  LOAD          0x00        // 清除页寄存器，使能装载
        # define  EP            0x68        // 擦除&编程页
        unsigned  char          i;          // 循环次数
      FMCON = LOAD;                         // 装载命令，清除页寄存器
      FMADRH = page_hi;         //
      FMADRL = page_lo;         // 页地址写入地址寄存器
      for  ( i= 0 ; i <16 ; i=i+1 )
          {
            FMDATA  =  dbytes [i] ;
          }
      FMCON = EP;               // 擦除&编程页命令
      Fm_stat = FMCON;          // 读结果状态
      if  ( ( Fm_stat & 0xF ) ! =0 )    prog_fail = 1 ; else  prog_fail = 0;
      return (prog_fail);
      }
```

图 14.3 擦除/编程（部分）页的 C 语言程序

其它 flash 单元的访问

除用户程序阵列外，用户固件还可访问其它 flash 单元，诸如 UCFG1、引导向量、状态位、加密字节和标识字节。这些单元的访问方法与用户程序存储器稍微有些不同。标识字节只可读，加密字节只能在某些特定条件下擦除。

IAP-Lite 功能在微控制器的固件控制下得以实现，通过使用 4 个 SFR 可轻松完成编程、擦除或读操作。

这些 SFR 为：

- FMCON（Flash 控制寄存器）。读时用作状态寄存器；写时用作命令寄存器。注：写入命令时状态位被清零。
- FMDATA（Flash 数据寄存器）。将接收的数据装入 flash 单元或读出 flash 单元的数据。
- FMADRL（Flash 存储器地址低字节）。用来指示 flash 单元。

可访问的 flash 单元及其地址如表 14.1 所示。

表 14.1 通过 IAP-Lite 来访问的 Flash 单元

单元	地址	描述
UCFG1	00h	用户配置字节 1
引导向量	02h	引导向量
状态位	03h	状态位字节
加密字节 0	08h	加密字节，扇区 0
加密字节 1	09h	加密字节，扇区 1
加密字节 2	0Ah	加密字节，扇区 2
加密字节 3	0Bh	加密字节，扇区 3
Mfgr ld	10h	标识字节，厂商 ID
ID_1	11h	标识字节，ID 1
ID_2	12h	标识字节，ID 2

擦除-编程其它 flash 单元

擦除-编程周期需要 4ms 的时间，其操作步骤如下：

- 将 flash 单元的地址写入 FMADRL。
- 将 CONF 命令（6CH）写入 FMCON。
- 将编程数据写入 FMDATA。
- 读 FMCON 来检查状态。如果操作被终止，重复执行。

向 FMDATA 写入编程数据将起动擦除-编程过程并使 CPU 进入编程-空闲状态。CPU 会一直保持这种空闲状态直到擦除-编程周期结束或被一个中断终止。从编程-空闲状态退出后，FMCON 的内容是该周期的状态信息。

如果在擦除/编程周期内有中断产生，擦除/编程操作被终止，FMCON 的 OI 标志（操作被中断）置位。如果某一应用允许在擦除-编程过程中产生中断，可通过用户程序在每次擦除-编程操作后检查 OI 标志（FMCON.0）来判断操作是否被终止。一旦擦除-编程操作终止，用户程序要重复整个过程。

读其它 flash 单元

通过以下步骤来读取 flash 单元的内容：

- 将 flash 单元的地址写入 FMADRL。
- 将 CONF 命令（6CH）写入 FMCON。
- 将数据从 FMDATA 中读出。

整个读操作可在一个机器周期内完成，CPU 无需进入空闲状态。读操作可被中断。但是，不必要进行状态检查。

一个 flash 单元擦除/编程操作的汇编语言程序如图 14-4 所示。类似的 C 语言程序如图 14.5 所示。读取 flash 单元的 C 语言程序如图 14.6 所示。

```
；* 输入：                                    ；
；*      R5   =   写入的数据（字节）          ；
；*      R7   =   单元地址（字节）            ；
；* 输出：                                    ；
；*      None

CONF    EQU    6CH

WR_ELEM:
        MOV     FMADRL,R7                   ；  写地址
        MOV     FMCON,#CONF                 ；  装载CONF命令
        MOV     FMDAT,R5                    ；  写数据
        MOV     R7,FMCON                    ；  备份返回的状态
        MOV     A,R7                        ；  读状态
        ANL     A,#0FH                      ；  保存低四位
        JNZ     BAD                         ；  判断操作是否有误
        CLR     C                           ；  如果操作无误，清除错误标志
        RET                                 ；  返回
BAD:
        SETB    C                           ；  如果操作出错，置位错误标志
        RET                                 ；  返回
```

图 14.4 擦除/编程 flash 单元的汇编程序

```
unsigned   char    Fm_stat;                    // 状态结果

bit   PGM_EL ( unsigned   char , unsigned   char );
bit   prog_fail;
void   main ( )
{
            prog_fail = PGM_EL ( 0x02 , 0x1C );
}
bit   PGM_EL (unsigned   char   el_addr , unsigned   char   el_data )
        {
                #define   CONF      0x6C        // 访问 Flash 单元
        FMADRL = el_addr ;                      // 将单元地址写入地址寄存器
        FMCON = CONF;                   // 装载命令，清除页寄存器
        FMDATA = el_data;               // 写入数据并起动操作周期
        Fm_stat = FMCON;                // 读结果状态
        if ( (Fm_stat & 0x0F) !=0 )   prog_fail =1;   else   prog_fail = 0;
        return (prog_fail);
        }
```

图 14.5 擦除/编程 flash 单元的 C 语言程序

```
# include   < REG921 . H>

unsigned   char   READ_EL (unsigned   char)；
unsigned   char   GET_EL;
void   main ( )
{
            GET_EL = READ_EL (0x02) ;
}
unsigned   char   READ_EL (unsigned   char   el_addr)
      {
            #define   CONF       0x6C              // 访问 Flash 单元
            unsigned   char   el_data;             // 读出的单元数据存放的位置
       FMADRL   = el_addr;                  // 将单元地址写入地址寄存器
       FMCON   = CONF;                     // 写入访问 Flash 单元的命令
       el_data   = FMDATA;                    // 读单元数据
       return ( el_data ) ;
       }
```

图 14.6 读 flash 单元的 C 语言程序

用户配置字节

P89LPC906/907/908 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，如图 14.7 所示。

UCFG1 地址：xxxxh

	7	6	5	4	3	2	1	0
未编程值：63h	WDTE	RPE	BOE	WDSE	-	FOSC2	FOSC1	FOSC0

位	符号	功能
UCFG1.7	WDTE	看门狗定时器复位使能位，置位时使能看门狗定时器的复位，清零时禁止看门狗定时器的复位。定时器仍可用于产生中断。详见表 13.1。
UCFG1.6	RPE	复位脚使能，当置 1 时，使能 P1.5 脚的复位功能。清零时，P1.5 可作为输入口使用。注：在上电过程中，RPE 选项无效，该管脚总是作为复位输入管脚。在复位之后，该管脚执行 RPE 位所定义的功能。只有上电复位才能暂时使 RPE 位的设定无效。其它复位源无法影响 RPE 位的设定。
UCFG1.5	BOE	掉电检测使能（见“掉电检测”）
UCFG1.4	WDSE	看门狗安全使能位。详见表 12.1。
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。其它信息详见“低功耗选择（P89LPC906）”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。当通过配置 FOSC2:0 选择内部 RC 或看门狗振荡器时，晶振配置由 RTCCON 控制。见表 6.1 和表 6.2。**注：P89LPC906 可使用外部时钟输入和晶振来获得 CPU 频率。**
	FOSC2-FOSC0	振荡器配置
	1 1 1	从 XTAL1 输入的外部时钟
	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%
	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

UCFG1 的默认设置如下：看门狗复位禁止、复位脚使能、掉电检测使能和使用内部 RC 振荡器。

图 14.7 Flash 用户配置字节 1（UCFG1）

用户保密字节

P89LPC906/907/908 有 4 个用户扇区保密字节（SEC0,…,SEC3），每个保密字节对应一个扇区，位分配如下：

SECx 地址：xxxxh

7	6	5	4	3	2	1	0
—	—	—	—	—	EDISx	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	EDISx	**禁止对扇区 x 的擦除。**禁止 IAP 模式下扇区 x 的擦除。通过编程，该位和扇区 x 只可使用商用编程器接口执行‘整体’擦除命令进行擦除。在 IAP 模式中，不能将该位和扇区 x 擦除。
SECx.1	SPEDISx	**禁止对扇区 x 的擦除和编程**。禁止对全部或部分扇区 x 进行编程或擦除。该位和扇区 x 可通过扇区擦除命令（IAP 或商用编程器）或‘整体’擦除命令（商用编程器）进行擦除。
SECx.0	MOVCDISx	**MOVC 禁能**。禁止对扇区 x 的 MOVC 操作。任何对 MOVC 指令受保护的扇区进行 MOVC 访问都将返回无效数据。只有当扇区 x 被擦除时才能将该位擦除。

图 14.8 用户扇区保密字节（SEC0, …, SEC3）

表 14.2 加密位的作用

EDISx	SPEDISx	MOVCDISx	编程影响
0	0	0	无。
0	0	1	指定扇区进行 CRC 计算时置位安全出错标志；如果 MOVCDISx 位置位，安全出错标志置位来进行整体 CRC 计算；操作周期终止；存储器内容不变；CRC 无效；执行编程/擦除命令不会造成安全出错。
0	1	X	执行编程命令或擦除页命令时置位安全出错标志；操作周期终止；存储器内容不变；允许扇区擦除和整体擦除。
1	X	X	执行编程或擦除命令时置位安全出错标志；操作周期终止；存储器内容不变；允许整体擦除。

引导向量

BOOTVEC 地址：xxxxh

7	6	5	4	3	2	1	0
-	-	-	BOOTV4	BOOTV3	BOOTV2	BOOTV1	BOOTV0

出厂默认值：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，P89LPC906/907/908 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“上电复位后的代码执行”）。

图 14.9 引导向量（BOOTVEC）

引导状态

BOOTSTAT 地址：xxxxh

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	BSB

出厂默认值：00h

位	符号	功能
BOOTSTAT.7-1	-	保留(保持其为未编程状态 0)
BOOTSTAT.0	BSB	引导状态位。如果该位编程为 1，P89LPC906/907/908 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“上电复位后的代码执行”）。

图 14.10 引导状态（BOOTSTAT）

15. 指令集

表 15.1 指令集小结

助记符	描述	字节	周期	Hex 码
	算术指令			
ADD A,Rn	寄存器的内容加到 A	1	1	28-2F
ADD A,dir	直接寻址字节的内容加到 A	2	1	25
ADD A,@Ri	间接寻址存储器单元的内容加到 A	1	1	26-27
ADD A,#data	立即数加到 A	2	1	24
ADDC A,Rn	寄存器的内容带进位位加到 A	1	1	38-3F
ADDC A,dir	直接寻址单元的内容带进位位加到 A	2	1	35
ADDC A,@Ri	间接寻址存储单元的内容带进位位加到 A	1	1	36-37
ADDC A,#data	立即数带进位位加到 A	2	1	34
SUBB A,Rn	从寄存器 A 中减去寄存器的内容，可借位	1	1	98-9F
SUBB A,dir	从寄存器 A 中减去直接寻址单元的内容，可借位	2	1	95
SUBB A,@Ri	从寄存器 A 中减去间接寻址存储单元的内容，可借位	1	1	96-97
SUBB A,#data	从寄存器 A 中减去立即数，可借位	2	1	94
INC A	寄存器 A 的值加 1	1	1	04
INC Rn	寄存器的值加 1	1	1	08-0F
INC dir	直接寻址单元的内容加 1	2	1	05
INC @Ri	间接寻址存储单元的内容加 1	1	1	06-07
DEC A	寄存器 A 的值减 1	1	1	14
DEC Rn	寄存器的值减 1	1	1	18-1F
DEC dir	直接寻址单元的内容减 1	2	1	15
DEC @Ri	间接寻址存储单元的内容减 1	1	1	16-17
INC DPTR	数据指针加 1	1	2	A3
MUL AB	A 和 B 相乘，乘积放在 A 中	1	4	A4
DIV AB	A 除以 B，商放在 A 中	1	4	84
DA A	A 的内容十进制调整	1	1	D4

续上表

助记符	描述	字节	周期	Hex 码
	逻辑指令			
ANL A,Rn	寄存器与 A 相与，结果放在 A 中	1	1	58-5F
ANL A,dir	直接寻址单元与 A 相与，结果放在 A 中	2	1	55
ANL A,@Ri	间接寻址存储单元与 A 相与，结果放在 A 中	1	1	56-57
ANL A,#data	立即数与 A 相与，结果放在 A 中	2	1	54
ANL dir,A	直接寻址单元与 A 相与，结果放在直接寻址单元中	2	1	52
ANL dir,#data	立即数与直接寻址单元相与，结果放在直接寻址单元中	3	2	53
ORL A,Rn	寄存器与 A 相或，结果放在 A 中	1	1	48-4F
ORL A,dir	直接寻址单元与 A 相或，结果放在 A 中	2	1	45
ORL A,@Ri	间接寻址存储单元与 A 相或，结果放在 A 中	1	1	46-47
ORL A,#data	立即数与 A 相或，结果放在 A 中	2	1	44
ORL dir,A	直接寻址单元与 A 相或，结果放在直接寻址单元中	2	1	42
ORL dir,#data	立即数与直接寻址单元相或，结果放在直接寻址单元中	3	2	43
XRL A,Rn	寄存器与 A 相异或，结果放在 A 中	1	1	68-6F
XRL A,dir	直接寻址单元与 A 相异或，结果放在 A 中	2	1	65
XRL A,@Ri	间接寻址存储单元与 A 相异或，结果放在 A 中	1	1	66-67
XRL A,#data	立即数与 A 相异或，结果放在 A 中	2	1	64
XRL dir,A	直接寻址单元与 A 相异或，结果放在直接寻址单元中	2	1	62
XRL dir,#data	立即数与直接寻址单元相异或，结果放在直接寻址单元中	3	2	63
CLR A	A 清零	1	1	E4
CPL A	A 取反	1	1	F4
SWAP A	A 的内容高低半字节交换	1	1	C4
RL A	A 的内容循环左移	1	1	23
RLC A	A 的内容带进位位循环左移	1	1	33
RR A	A 的内容循环右移	1	1	03
RRC A	A 的内容带进位位循环右移	1	1	13
	数据传送			
MOV A,Rn	寄存器的数据传送给 A	1	1	E8-EF
MOV A,dir	直接寻址单元的内容传送给 A	2	1	E5
MOV A,@Ri	间接寻址存储单元的内容传送给 A	1	1	E6-E7
MOV A,#data	立即数传送给 A	2	1	74
MOV Rn,A	A 的内容传送给寄存器	1	1	F8-FF
MOV Rn,dir	直接寻址单元的内容传送给直接寻址字节	2	2	A8-AF
MOV Rn,#data	立即数传送给寄存器	2	1	78-7F
MOV dir,A	A 的内容传送给直接寻址单元	2	1	F5
MOV dir,Rn	寄存器的内容传送给直接寻址单元	2	2	88-8F
MOV dir,dir	直接寻址单元的内容传送给直接寻址单元	3	2	85
MOV dir,@Ri	间接存储单元的内容传送给直接寻址单元	2	2	86-87
MOV dir,#data	立即数传送给直接寻址单元	3	2	75
MOV @Ri,A	A 的内容传送给间接寻址存储单元	1	1	F6-F7
MOV @Ri,dir	直接寻址单元的内容传送给间接寻址存储单元	2	2	A6-A7

续上表

助记符	描述	字节	周期	Hex 码
	数据传送			
MOV @Ri,#data	立即数传送给间接寻址存储单元	2	1	76-77
MOV DPTR,#data	立即数传动给数据指针	3	2	90
MOVC A,@A+DPTR	DPTR 相对地址单元的代码字节传送给 A	1	2	93
MOVC A,@A+PC	PC 相对地址单元的代码字节传送给 A	1	2	94
MOVX A,@Ri	外部数据（A8）传送给 A	1	2	E2-E3
MOVX A,@DPTR	外部数据（A16）传送给 A	1	2	E0
MOVX @Ri,A	A 的内容传送到外部数据单元（A8）	1	2	F2-F3
MOVX @DPTR,A	A 的内容传送到外部数据单元（A16）	1	2	F0
PUSH dir	直接寻址单元的内容压栈	2	2	C0
POP dir	直接寻址单元的内容出栈	2	2	D0
XCH A,Rn	交换 A 和寄存器的内容	1	1	C8-CF
XCH A,dir	交换 A 和直接寻址单元的内容	2	1	C5
XCH A,@Ri	交换 A 和间接寻址存储单元的内容	1	1	C6-C7
XCHD A,@Ri	交换 A 和间接寻址存储单元半字节的内容	1	1	D6-D7
	位处理指令			
助记符	描述	字节	周期	Hex 码
CLR C	进位位清零	1	1	C3
CLR bit	直接寻址位清零	2	1	C2
SETB C	进位位置位	1	1	D3
SETB bit	直接寻址位置位	2	1	D2
CPL C	进位位取反	1	1	B3
CPL bit	直接寻址位取反	2	1	B2
ANL C,bit	直接寻址位与进位位相与，结果放在进位位中	2	2	82
ANL C,/bit	直接寻址位取反后与进位位相与，结果放在进位位中	2	2	B0
ORL C,bit	直接寻址位与进位位相或，结果放在进位位中	2	2	72
ORL C,/bit	直接寻址位取反后与进位位相或，结果放在进位位中	2	2	A0
MOV C,bit	直接寻址位的值传送给进位位	2	1	A2
MOV bit,C	进位位传送给直接寻址位	2	2	92
	转移指令			
ACALL addr11	子程序的绝对调用	2	2	116F1
LCALL addr16	长调用到子程序	3	2	12
RET	子程序返回	1	2	22
RETI	中断返回	1	2	32
AJMP addr11	无条件绝对跳转	2	2	016E1
LJMP addr16	无条件长跳转	3	2	02
SJMP rel	短跳转（相对地址）	2	2	80
JC rel	进位位＝1 时跳转	2	2	40
JNC rel	进位位＝0 时跳转	2	2	50

续上表

助记符	描述	字节	周期	Hex 码
	转移指令			
JB bit,rel	直接寻址位＝1 时跳转	3	2	20
JNB bit,rel	直接寻址位＝0 时跳转	3	2	30
JBC bit,rel	直接寻址位＝1 时跳转，然后清零	3	2	10
JMP @A+DPTR	跳转到 DPTR 相对地址指向的地址	1	2	73
JZ rel	累加器＝0 时跳转	2	2	60
JNZ rel	累加器≠0 时跳转	2	2	70
CJNE A,dir,rel	比较 A 和直接寻址单元的内容，不等时转移到相对地址	3	2	B5
CJNE A,#d,rel	比较 A 和立即数，不等时转移到相对地址	3	2	B4
CJNE Rn,#d,rel	比较寄存器和立即数，不等时转移到相对地址	3	2	B8-BF
CJNE @Ri,#d,rel	比较间接寻址单元和立即数，不等时转移到相对地址	3	2	B6-B7
DJNZ Rn,rel	寄存器减 1，不为零时转移到相对地址	2	2	D8-DF
DJNZ dir,rel	直接寻址单元减 1，不为零时转移到相对地址	3	2	D5
	其它指令			
NOP	空操作	1	1	00

16. 修订记录

2003 年 12 月 8 日

第一版