

特性

- 与 MCS51 产品兼容的 8 位单片机
- 增强的 8051 结构
 - 单周期指令
 - 时钟频率达到 20MIPS 运行速率
 - 全静态操作频率: 0Hz 到 20MHz
 - 片内双周期硬件乘法器
 - 128 x 8 内部 1 RAM
 - 4 级中断优先
- 非易失性程序存储
 - 2K 字节系统内部程序 (ISP) Flash 存储器
 - 最少 10,000 可重复擦写
 - 最少 10 年的数据保持时间
 - 32 字节快速编程模式
 - 2 级程序存储器软件加密保护
 - 64 字节用户标识区
 - 2 级程序存储器软件加密保护
- 外围特性
 - 两个 16 位加强定时器、计数器
 - 两个 8 位 PWM 输出
 - 增强型自动地址识别和帧结构错误探测 UART 接口
 - 增强型双缓存发送接受主从式 SPI 接口
 - 带软件复位的可编程看门狗定时器
 - 可选中断和去抖的模拟比较器
 - 8 个常用中断引脚
- 特有属性
 - 双线片上调试接口
 - Power-off 标志位可控布朗输出检测和上电复位
 - 内部 RC 振荡器
 - 低功耗空闲和低电压模式
 - 低电压中断恢复
- I/O 和封装
 - 多达 14 个可编程 I/O 口
 - I/O 可配置的工作方式为准双工、输入、上拉输出和开漏输出
 - I/O 口最大承受电压 5V
 - 16-lead TSSOP/SOIC/PDIP 封装
- 工作条件
 - 电源电压范围 2.4V 到 5.5V
 - 温度范围 -40° C 到 85°

1. 描述

AT89LP216 是一款低功耗、高性能 CMOS 8 位单片机，它有 2k 字节 ISPFlash 存储器。产品生产采用 Atmel 的高密度非易失性存储器技术而且和工业标准 de 的 MCS51 指令集相兼容。AT89LP216 基于一个加强性 CPU 内核，每时钟周期读取单字节指令。在经典 8051 结构中，每次读取需要 6 个时钟周期，使得执行指令需要 12、24 或者 48 个时钟周期。在 AT89LP216CPU 中，指令只需要 1 到 4 个时钟周期就可以达到传统 8051 速度的 6 到 12 倍。70% 的指令字节数与执行的时钟周期数相等，而且其他指令只需要一个额外时钟。在相同功耗下，增强型 CPU 内核可达到 20MIPS，而传统 8051CPU 只能达到 4MIPS。相反地，在相同的工作速率下，新 CPU 内核比传统的 8051 拥有更低的时钟速率和功耗。



带 2K 字节 Flash
的 8 位微控制器

AT89LP216
主要功能

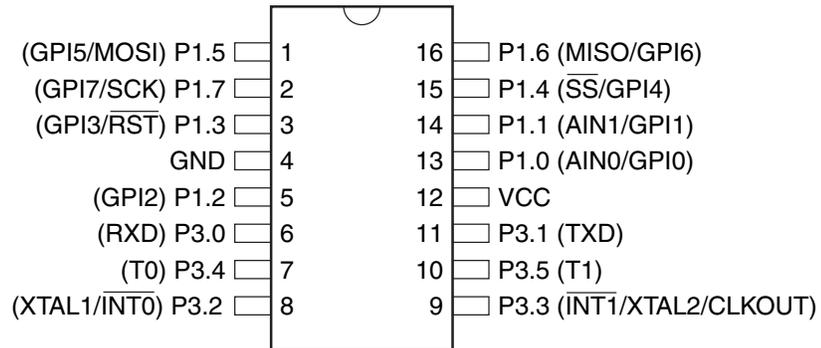
本文是英文数据手册的中文翻译，其目的是方便中国用户的阅读。它无法自动跟随原稿的更新，同时也可能存在翻译上的错误。读者应该以英文原稿为参考以获得更准确的信息。

AT89LP216 也拥有下列标准的特性：2K 字节 ISPFlash 存储器，128 字节 RAM、多达 12 个 I/O 口、2 个 16 位定时器 / 计数器，两 PWM 输出，一个可编程看门狗定时器，一个全双工串口，一个串行外围接口，一个内部 RC 振荡器，片上石英振荡器和一个 4 级、6 矢量中断系统。

AT89LP216 里的两个定时器 / 计数器增加了两个新模式。模式 0 可以被设置为 9 到 16 位的定时器 / 计数器，模式 1 可被设置位 16 位自动装载定时器 / 计数器。此外，定时器 / 计数器可以独立驱动 PWM 输出。AT89LP216 里面的 I/O 口能被独立配置为 4 种工作模式的其中一种。在准双工模式中，I/O 口的工作模式和传统 8051 一样。在输入模式中，接口是三态门。推挽输出模式提供足够的 CMOS 驱动，开漏模式则起到一个下拉的作用。另外，Port1 的所有 8 个引脚可以作为通用中断接口。AT89LP216 的 I/O 口能承受的电压可超出电源电压达到 5.5V。当器件的电源电压为 2.4V 而 I/O 口输入 5.5V 时，所有 I/O 口的反向电流总和不超过 100 μ A。

2. 引脚图

2.1 AT89LP216: 16-lead PDIP/SOIC/TSSOP



3. 引脚描述

Table 3-1. AT89LP216 引脚描述

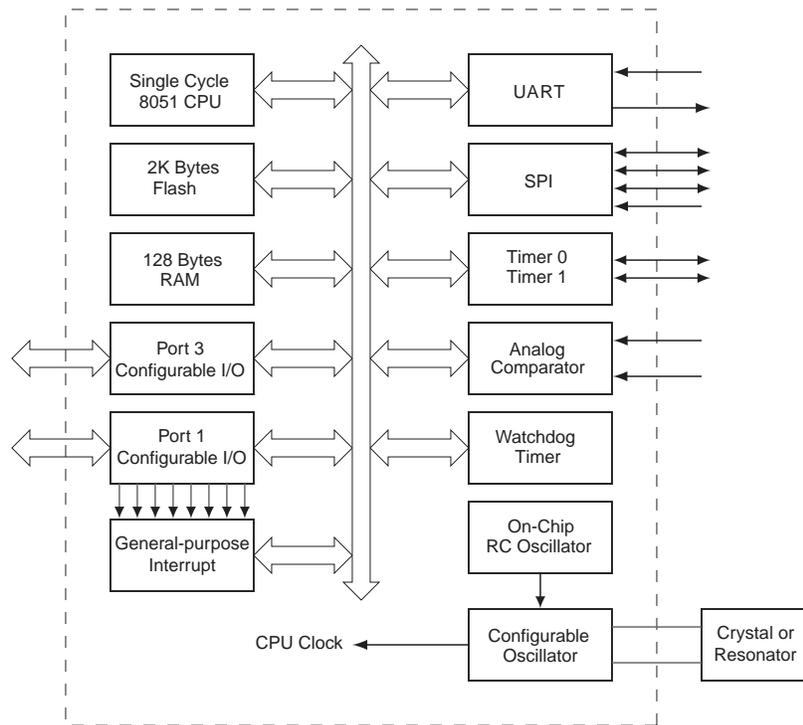
管脚	符号	类型	描述
1	P1.5	I/O I/O I	P1.5: 用户可定义 I/O 口 1 bit 5. MOSI: SPI 主出 / 从入. 当设置为主机时, 这个引脚是输出。当设置为从机时这个引脚为输入。 GPI5: 常用中断输入 5
2	P1.7	I/O I/O I	P1.7: 用户可定义 I/O 口 1 bit 7 SCK: SPI 时钟。当设置为主机时, 这个引脚是输出。当设置为从机的时候该引脚作输入用 GPI7: 常用中断输入 7
3	P1.3	I/O I I I	P1.3: 用户可定义 I/O 口 (如果复位保险被禁止) RST: 外部低电平有效复位输入 (如果复位保险被允许。见第 14 页 “外部复位”) GPI3: 常用中断输入 3 DCL: 当 OCD 有效, 片上调试接口的串行时钟输入
4	GND	I	地
5	P1.2	I/O I	P1.2: 用户可定义 I/O 口 1 bit 2 GPI2: 常用中断输入 2
6	P3.0	I/O I	P3.0: 用户可定义 I/O 口 3 bit 0 RXD: 串口接收输入
7	P3.4	I/O I/O	P3.4: 用户可定义 I/O 口 3 bit 4 TO: 计时器 / 计数器 0 外部输入或 PMW 输出
8	P3.2	I/O I I/O	P3.2: 用户可定义 I/O 口 3 bit 2 XTAL1: 振荡器反向放大器和内部时钟产生电路输入。如果启用内部 RC 振荡器作为时钟源, 则该脚也可被用作输入输出 DDA: 当 OCD 有效时, 作片上调试接口的串行输入输出和使用内部 RC 振荡器作时钟源
9	P3.3	I/O O O I/O	P3.3: 用户可定义 I/O 口 3 bit 3. XTAL2: 振荡器反向放大器输出, 当内部 RC 振荡器被用作时钟源时, 该口可用作输入输出 CLKOUT: 当内部 RC 振荡器被用作时钟源时, 该口被用作内部时钟 2 分频的输出 DDA: 当 OCD 有效时, 作片上调试接口的串行输入输出和使用内部 RC 振荡器作时钟源
10	P3.5	I/O I/O	P3.5: 用户可定义 I/O 口 3 bit 5 T1: 计时器 / 计数器 1 的外部输入或 PWM 的输出
11	P3.1	I/O O	P3.1: 用户可定义 I/O 口 3 bit 1. TXD: 串口传输输出
12	VDD	I	电源供应
13	P1.0	I/O I I	P1.0: 用户可定义 I/O 口 1 bit 0. AIN0: 模拟比较器正向输入端 GPI0: 通用中断输入 0

Table 3-1. AT89LP216 引脚描述

管脚	符号	类型	描述
14	P1.1	I/O I	P1.1: 用户可定义 I/O 口 1 bit 1. AIN1: 模拟比较器反向输入端 GPI1: 通用中断输入 1
15	P1.4	I/O I	P1.4: 用户可定义 I/O 口 1 bit 4. SS: 从 SPI 选择输入端 GPI4: 通用中断输入 4
16	P1.6	I/O I/O 	P1.6: 用户可定义 I/O 口 1 bit 6. MISO: SPI 接口的主机输入 / 从机输出端口。当定义为主机时, 该引脚作输入端, 当定义为从机时该引脚作输出端。 GPI6: 通用中断输入 6

4. 内部结构

Figure 4-1. AT89LP216 内部结构图



5. 与传统 8051 的比较

AT89LP216 是具有增强功能的 51 系列单片机, 它完全兼容于 MCS-51 的指令。另外, 大多数的 SFR 地址, 位定义以及引脚的第二功能都与已有的 Atmel 的 51 系列单片机相同。然而, 鉴于这款单片机的高性能, 某些系统性能与诸如 AT89S52 或 AT89S2051 之类的 51 系列产品有所不同。这些不同之处将在下列章节中详细介绍。

5.1 系统时钟

CPU 的时钟频率与外部 XTAL1 端口的频率相同。此时振荡器不再分频以提供内部时钟，且不支持倍频。

5.2 单周期读取指令的执行

CPU 每时钟周期从存储器中读取一个字节的指令而不是每 6 个时钟周期读取一次。这样就能极大的提升 CPU 的性能。因为，CPU 执行指令不再需要 12 到 48 个指令周期，而只需要 1 到 4 个指令周期。详见第 59 页“指令设置汇总”。

5.3 中断处理

中断控制器在任意一条指令的最后一个时钟周期内获取中断标志位。在下一个到最后一个指令执行的时钟周期中，或在前一条指令的最后一个执行时钟周期内（如果当前指令是一个单周期指令）中断标志位必需置位为有效。

外部中断引脚， $\overline{INT0}$ 和 $\overline{INT1}$ ，每个时钟周期都会被采样一次，而不是每 12 个时钟周期采样一次。更短的指令执行时间加上更快速的中断响应，保证了 CPU 对外部中断的高速响应。

5.4 定时器 / 计数器

定时器 / 计数器的步进增量默认为一个时钟周期一次。传统的 8051 需要 12 个时钟周期才增加一次。对所有的定时器来说，时间基数可被一个比例因子整除，并且可减少步进增量频率。在特殊功能寄存器 CLKREG 的 TPS 标志位中可设置该比例因子（见第 12 页表 9 - 2）。当设置 TPS = 1011B 时，计数器每计一次需 12 个时钟周期。

外部定时器 / 计数器引脚，T0 和 T1，每个时钟周期而不是每 12 个时钟周期被采样一次。这样在计数器模式中计数的频率将更高。

5.5 串口

在模式 0 下，UART 口的波特率是系统时钟频率的一半，而传统 8051 是系统时钟的 1/12。其输出数据仅在串行时钟的上升沿保持稳定。还需注意的是当使用定时器 1 在模式 1 或模式 3 下产生波特率，定时器是以时钟频率计数的，而不是以时钟频率的 1/12 来计数的。要让和 8051 使用相同系统时钟的 AT89LP216 产生同样的波特率，其时钟输出时间必需是原来的 12 倍。定时器 1 在模式 1 下支持 16 位的自动重装，这样能延长时钟输出，从而获得较低的波特率。

5.6 看门狗定时器

相对于传统的 8051 需要 12 个时钟周期才记一次数，AT89LP216 的看门狗定时器每时钟周期计一次数。时间基数可被一个比例因子整除，并且可减少步进增量频率。

5.7 I/O 口

AT89LP216 的 I/O 口可定义成 4 个不同的模式。所有 I/O 口在上电或复位时都默认定义为只输入模式（三态门）。传统的 8051 当上电或复位时都是输出高电平的。如果要使其和 8051 一样，那么必需清除特殊功能寄存器的 P1M0 和 P3M0，使 I/O 口进入准双工模式。用户还可通过禁止三态门用户保险来定义端口为准双工模式。当保险被禁止时，P1M0 和 P3M0 会复位成 00h 而不是 FFh，而且端口被拉成高电平。

5.8 复位

AT89LP216 的复位引脚 RST 相比于 8051 的高电平有效复位它是低电平有效复位的。另外，复位引脚每个时钟周期都采样一次，且至少保持 2 个时钟周期的低电平有效才能被确认复位有效，而不是 24 个时钟周期。

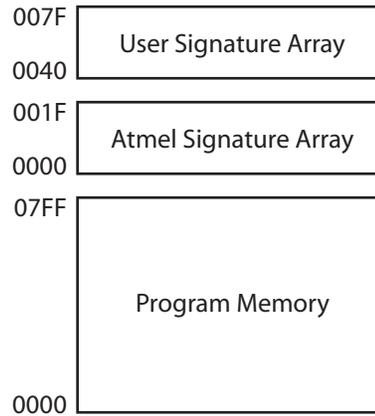
6. 内存结构

AT89LP216 使用的是哈佛结构，它给程序存储器和数据存储器设置了独立的地址空间。程序存储器有支持 64k 字节的直接寻址指令。数据存储器包括 128 字节的内部 RAM 和 128 字节的特殊功能寄存器。AT89LP216 不支持外部数据存储器 and 外部程序存储器。

6.1 程序存储器

AT89LP216 包括 2k 字节的片内 FLASH。FLASH 可重复擦写 10000 次，且数据能保持 10 年。复位和中断向量存放在程序存储器的前 59 个字节中（参考第 19 页，表 12 - 1）。MOVC 指令可寻址 2k 字节的程序存储器。AT89LP216 不支持外部程序存储器。

Figure 6-1. 程序存储器映射



AT89LP216 程序存储器的映射图示于图 6 - 1。除从 0000h 到 07FFh 的 2k 字节程序存储器之外，AT89LP216 还支持 64 字节的在 CPU 只读模式下访问的用户标识区和 32 字节的 ATMEL 标识区。必须对 AUXR1 的 SIGEN 置位才能读标识区。当 SIGEN 为 1，使用 MOVC A, @A+DPTR 指令进入标识区。用户标识区从 0040h 到 007Fh，ATMEL 标识区从 0000h 到 001Fh。SIGEN 需在 MOVC 访问指令存储器前清除。

在出厂前使用器件 ID 来初始化 ATMEL 标识区。用户标识区使用用户鉴别码或常数来初始化。存储在标识区的数据是不安全的。安全位会使写入标识区的操作无效；然而从该区读是始终允许的。

Table 6-1. AUXR1 - 辅助寄存器 1

AUXR1 = A2H	复位值 = XXXX 0XXXB
不可位寻址	

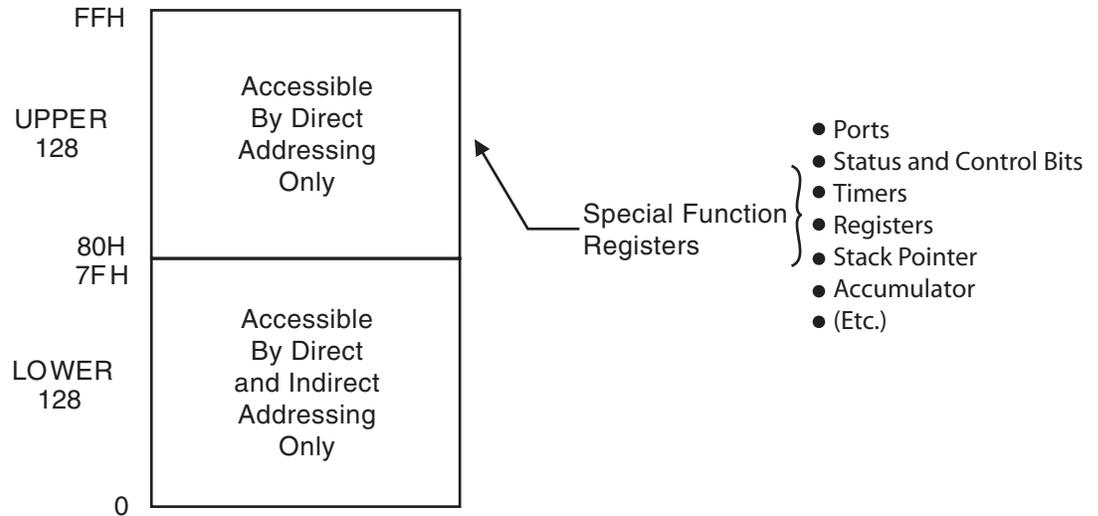
Table 6-1. AUXR1 – 辅助寄存器 1

	–	–	–	–	SIGEN	–	–	–
Bit	7	6	5	4	3	2	1	0

6.2 数据存储器

AT89LP216 包括 128 字节的常用 SRAM 数据存储器 and 128 字节的 I/O 存储器，它们映射在一个 8 位的地址空间中。128 字节的数据存储器可通过直接和间接对低 128 字节地址寻址进行访问。128 字节的 I/O 存储器存放在高 128 字节的地址空间中（图 6-2）。I/O 存储器只可用直接寻址访问，且包括特殊功能寄存器（SFRs）。对高 128 字节的寻址操作会返回无效数据。数据存储器的低 32 字节被分为 4 段分别放入 8 个存储器中。RS0 和 RS1 位（PSW. 3 和 PSW. 4）决定哪个存储区域被使用。使用寄存器寻址的指令只能访问当前区域。AT89LP216 不支持外部数据存储器

Figure 6-2. 数据存储器映射



7. 特殊功能寄存器

特殊功能寄存器示于表 7-1

注：并非所有的地址被占用，未被占用的地址不能进行任何操作。读这些地址通常会返回随机数，写访问这些地址会产生未知后果。用户不应写入这些未列表的区域，因为它们可能会被使用在新产品中开发新功能。

Table 7-1. AT89LP216 SFR 映射和复位值

	8	9	A	B	C	D	E	F	
0F8H									0FFH
0F0H	B 0000 0000								0F7H
0E8H	SPSR 000x x000	SPCR 0000 0000	SPDR xxxx xxxx						0EFH
0E0H	ACC 0000 0000								0E7H
0D8H									0DFH
0D0H	PSW 0000 0000								0D7H
0C8H									0CFH
0C0H			P1M0 ⁽²⁾	P1M1 xx00 0000			P3M0 ⁽²⁾	P3M1 xx00 0000	0C7H
0B8H	IP x000 0000	SADEN 0000 0000							0BFH
0B0H	P3 xx11 1111							IPH x000 0000	0B7H
0A8H	IE 0000 0000	SADDR 0000 0000							0AFH
0A0H			AUXR1 xxxx 0xxx				WDTRST (write-only)	WDTCON 0000 x000	0A7H
98H	SCON 0000 0000	SBUF xxxx xxxx	GPMOD 0000 0000	GPLS 0000 0000	GPIEN 0000 0000	GPIF 0000 0000			9FH
90H	P1 1111 1111	TCONB 0010 0100	RL0 0000 0000	RL1 0000 0000	RH0 0000 0000	RH1 0000 0000		ACSR xx00 0000	97H
88H	TCON 0000 0000	TMOD 0000 0000	TL0 0000 0000	TL1 0000 0000	TH0 0000 0000	TH1 0000 0000		CLKREG 0000 x000	8FH
80H		SP 0000 0111	DPL 0000 0000	DPH 0000 0000				PCON 0000 0000	87H
	0	1	2	3	4	5	6	7	

- Notes: 1. 所有的特殊功能寄存器最左端的列可位寻址。
 2. 当 3 态门保险允许时复位值是 xx11 1111B, 当 3 态门保险不允许时复位值是 xx00 0000B。

8. 增强型 CPU

AT89LP216 使用了一个增强型的 8051CPU, 它的运行速度是标准 8051 的 6 到 12 倍 (或 3 到 6 倍的 X28051) 这种性能提升的原因有两个。第一, 每个时钟周期 CPU 从指令存储器中读取一条单字节指令。第二, CPU 使用一个简单的双线程从而使读取指令和执行指令同时进行。这个基本的线程操作思想使 CPU 工作速率达到 1MIPS 每 MHz。一个简单的实例如图 8 - 1 所示。

MCS-51 指令设置允许指令长度为 1 到 3 字节不等。在“单时钟周期读取单字节指令”系统中，这意味着每条指令的字节数和时钟周期数相等。AT89LP216 的大多数指令遵循下列原则：除少数指令外，指令执行时钟数与指令字节数相等。分支语句和调用语句相当于增加一个计算目标地址的时钟周期，其他复杂指令需要多个周期。更多指令详情请见第 59 页“指令设置汇总”。图例 8-2 和 8-3 是单字节和双字节指令执行过程。

Figure 8-1. 并行指令读取和执行

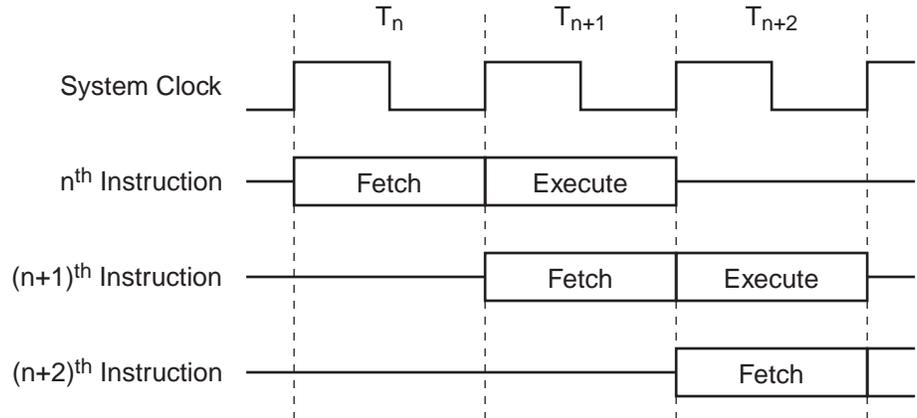


Figure 8-2. 单周期 ALU 操作（例：INC R0）

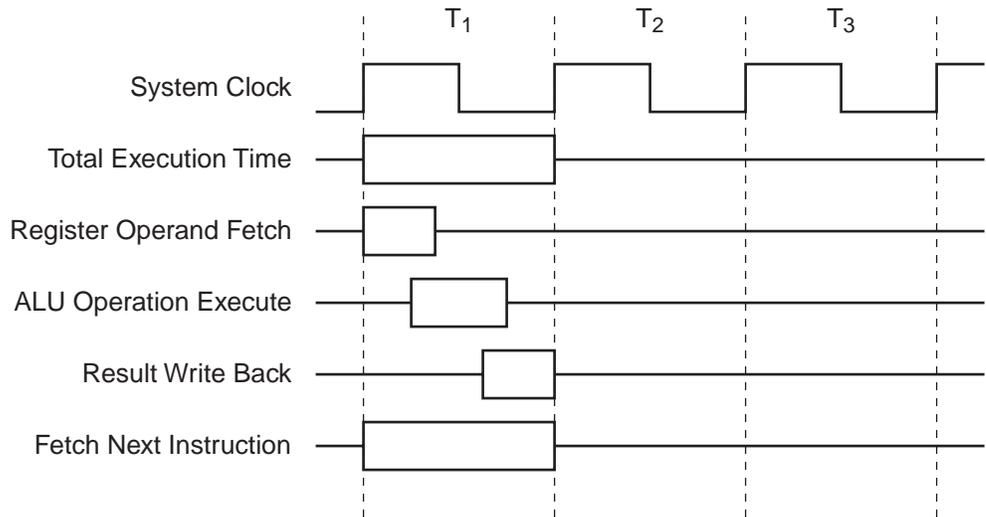
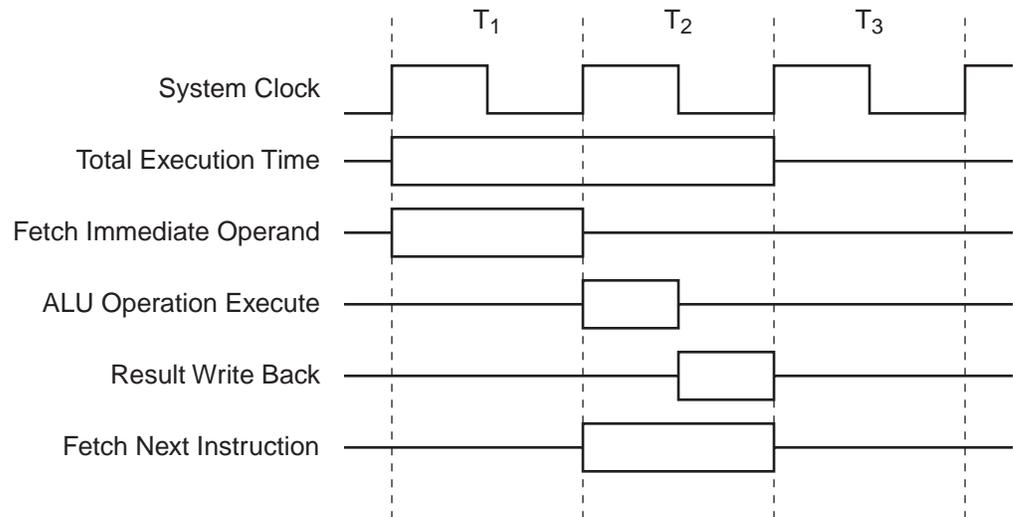


Figure 8-3. 双周期 ALU 操作（例：ADD A, #data）



8.1 部分指令限制

AT89LP216 在 ATMEL 单片机家族中是一个廉价的成员。它拥有 2k 字节的 FLASH 程序存储器。它兼容了 MCS - 51 的架构，同时可用 MCS - 51 的指令对其进行编程设置。然而，某些指令有所不同。所有跳转和分支语句都只能在物理内存（2k 字节）的范围内。这是程序员应该注意的。比如，LJMP 7E0H 是一条有效指令，而 LJMP 900H 则不是。

8.1.1 分枝指令

LCALL, LJMP, ACALL, AJMP, SJMP, and JMP @A+DPTR 等无条件分支指令在物理内存的范围内（000h 到 7FFh）可直接跳转到目的地址。如果超出了物理内存的限制将会得到一个未知的运行结果。CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, and JNZ 等有条件分支语句也必须遵守这个规则。重申一遍，如果超出内存范围进行操作会引起不稳定。在中断的应用中，8051 架构中保护了中断服务地址。

8.1.2 数据存储器，MOVX 相关指令

AT89LP216 拥有 128 字节的内部数据存储器。访问 RAM 过程中若超过 7FH 将会返回一个无效的数据。另外，堆栈深度为 128 字节，也就是 RAM 的大小。堆栈指针不允许超过 7FH。该器件不支持对外部数据存储器 and 外部程序存储器的操作。因此，程序中不应有 MOVX [...] 这样的语句。

即使超过之前所讲的物理内存的范围进行编程，典型的 8051 编译器还是会对其进行编译。所以，用户应该本款产品的特点和一些限制，从而合理的调整程序。

9. 系统时钟

三个可选时钟源中的一个直接产生系统时钟，这可通过配置完成。这三个时钟源分别是外部晶体振荡器、外部时钟信号、片内 RC 振荡器。如表 9-1 所示，时钟源可以通过表中时钟源用户熔丝位

来进行选择，CPU 时钟脉冲不能通过对系统时钟的分频来得到。参阅第 71 页上的“[用户配置熔丝位](#)” on page 68.

Table 9-1. 时钟源设置

时钟源设置位 1	时钟源设置位 0	时钟源选择
0	0	晶体振荡器
0	1	保留
1	0	外部时钟 XTAL1
1	1	内部 8 MHz RC 振荡电路

9.1 晶体振荡器

当被使能后，内部反向振荡器连接在 XTAL1 和 XTAL2 之间作为与外部石英晶体或陶瓷振荡器的连接。当使用晶体振荡器时，P3.2 和 P3.3 将失去输入 / 输出功能。同时，XTAL2 引脚应连接一个缓冲器来驱动板级时钟。

9.2 外部时钟信号

选用外部时钟时，振荡放大器被禁止，而 XTAL1 能被外部时钟信号直接驱动。XTAL2 可以悬空，可以作为 P3.3 通用 I/O 功能使用，或者通过配置作为系统时钟信号的分频输出。

9.3 内部 RC 振荡器

AT89LP216 内部有一个 8.0 MHz \pm 2.5% 的 RC 振荡器。当用作时钟源时，XTAL1 和 XTAL2 可以分别作为通用 I/O P3.2 和 P3.3 使用。XTAL2 也可以作为系统时钟的分频输出。振荡器的频率可以通过设置 RC 调整熔丝位来进行调整（参考第 71 页“[用户配置熔丝位](#)”）。(See “[用户配置熔丝位](#)” on page 68.).

9.4 系统时钟输出

当配置为既不采用外部时钟，也不采用内部 RC 振荡器时，AT89LP216 可以从引脚 XTAL2(P3.3) 输出一个系统分频时钟。这可以通过设置 CLKREG 中的 CLKREG 位来获取输出时钟的参数。其中，CDV[1,0] 决定时钟的分频数。例如，当 COE = “1” 且 CDIV = “00”，P3.3 配置为输出时，可从 P3.3 引脚输出一个频率为 3.950 MHz (\pm 5%) 的时钟信号。

Table 9-2. CLKREG - 时钟控制寄存器

CLKREG = 8FH						复位值 = 0000 0000B		
不可位寻址								
	TPS3	TPS2	TPS1	TPS0	-	CDV1	CDV0	COE
Bit	7	6	5	4	3	2	1	0

符号	功能
TPS3 TPS2 TPS1 TPS0	定时器预选器。用于为定时器 0、定时器 1 和看门狗定时器选择时基信号，是通过 4bit 递减计数器来实现预选功能。当计数到 0 时，就把 TPS 中的预分频值重装载来得到 1 ~ 16 的分频比。在默认值（TPS = 0000B）情况下，每个时钟周期的到来，定时器都会计数。如果要配置为标准 8051 的速率，即 1/12 时钟周期时，必须设置 TPS=1011B。

符号	功能
CDV1 CDV0	分频时钟输出。决定输出时钟信号的频率与系统时钟之间的关系。
	<u>CDIV1</u> <u>CDIV0</u> <u>输出时钟频率</u>
	0 0 f/2
	0 1 f/4
	1 0 f/8
	1 1 f/16
COE	时钟输出使能。使能 COE 可以从引脚 XTAL2 (P3.3) 输出系统时钟信号的一个分频信号。这时，必须选择内部 RC 振荡器或者外部时钟作为系统时钟源。

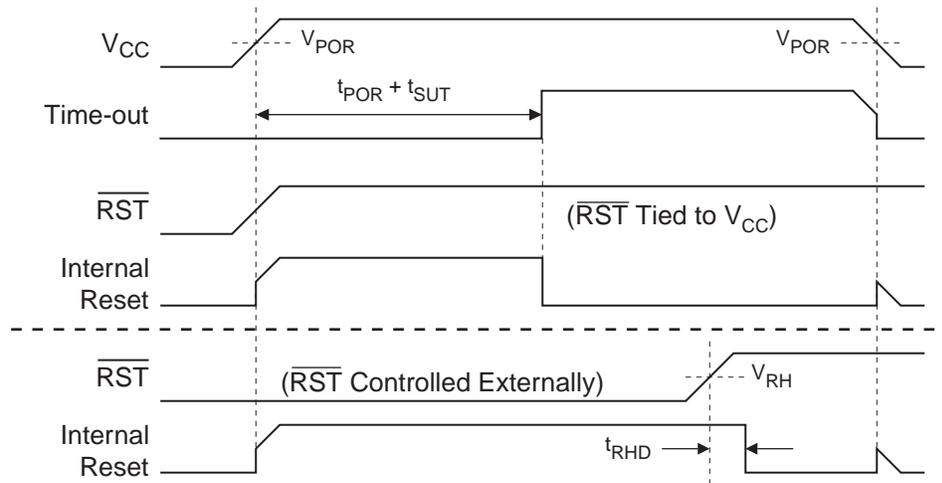
10. 复位

当复位时，所有的 I/O 寄存器都初始化为默认值，输入引脚都为三态，程序从复位矢量地址 0000H 开始执行。AT89LP216 有 5 个复位源：上电复位、欠压复位、外部复位、看门狗复位和软件复位。

10.1 上电复位

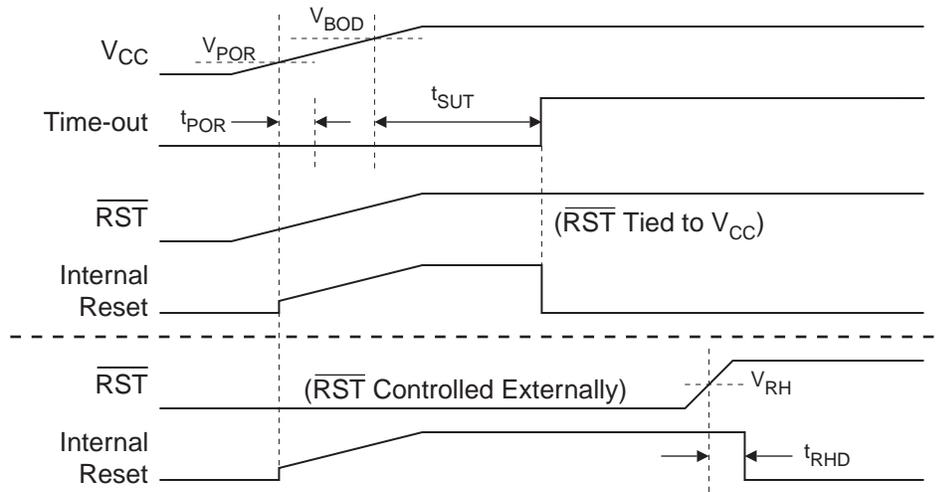
上电复位 (POR) 由片上电压检测电路完成，检测电平一般为 1.4V。当 VCC 低于该电平时，POR 将被激活。因此，POR 电路可用于启动复位或者在没有欠压检测器时检测器件工作电压是否正常而复位芯片。POR 能够保证芯片上电时复位正确。上电复位时序如第 13 页的图 10-1 所示。当 VCC 达到上电复位阈值电压 VPOR 时，初始化序列就被执行，持续时间为 tPOR。初始化序列完成后，启动定时器就决定器件随着 VCC 的上升而持续多长时间 POR。当 VCC 电压低于阈值电压时，POR 在没有任何延时的情况下再次被执行。上电复位（即冷复位）将置位 PCON 中的 POF 标志。另外，当 RST 引脚持续低电平周期大于 Time-out 设定的时间时，芯片内部将产生复位信号。

Figure 10-1. 上电复位时序 (BOD Disabled)



如果同时使能了欠压检测器 (BOD)，这时启动定时器只有在 VCC 达到阈值电压 VBOD 后才会开始计数如图 10-2。然而，如果在初始化时序完成前就发生了欠压，启动定时器必须等待初始化序列完成后才能开始计数。

Figure 10-2. 上电复位 (BOD 开启)



Note: t_{POR} 大约为 $92 \mu s \pm 5\%$

启动定时器的延时可通过配置“启动定时器用户熔丝位”和选择时钟源（表 10-1）来完成。该启动延时必须为 VCC 和时钟源的选择提供足够的建立时间。启动定时熔丝位用于控制芯片在欠压复位或者从内部定时唤醒低功耗模式而所需的芯片启动时间。

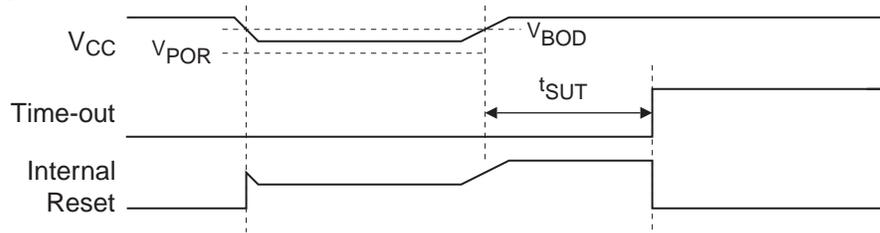
Table 10-1. 启动定时器设置

SUT 熔丝 1	SUT 熔丝 0	时钟源	$t_{SUT} (\pm 5\%) \mu s$
0	0	内部 RC/ 外部时钟信号	16
		晶体振荡器	1024
0	1	内部 RC/ 外部时钟信号	512
		晶体振荡器	2048
1	0	I 内部 RC/ 外部时钟信号	1024
		晶体振荡器	4096
1	1	内部 RC/ 外部时钟信号	4096
		晶体振荡器	16384

10.2 欠压复位

AT89LP216 带有一个片上欠压检测 (BOD) 电路, 当在工作状态时, 用于监视 VCC 并把它与预设定的某个触发阈值进行比较。该 BOD 阈值一般为 2.2V。BOD 的目的是保证系统在全速运行时由于 VCC 的下降而不发生任何错误的执行就能正确地进入复位状态。BOD 时序如图 10-3。当 VCC 低于阈值 V_{BOD} 时, 内部复位马上被激活。当 VCC 大于该 BOD 阈值并维持到 time-out 周期后, 启动定时器就释放内部复位信号 (如表 10-1)。通过设置 BOD 使能熔丝位来使能欠压监测器 (参考第 71 页的“用户配置熔丝位”)。(See “用户配置熔丝位” on page 68.)

Figure 10-3. 欠压检测器复位



10.3 外部

引脚 P1.3/RST 既可以作为低电平有效复位输入，也可以作为通用 I/O P1.3 功能使用。当置位复位使能熔丝位（即置‘1’）时，P1.3 被配置为外部复位输入（参考第 71 页的“用户配置熔丝位”）。当清零时，P1.3 用作输入 / 输出引脚。当配置为复位输入引脚时，该引脚上的电平必须维持至少两个时钟周期的低电平才能触发芯片内部复位。

Note: 在上电时序过程中，熔丝位的设置总是被忽略的，这时该引脚用作复位输入。连接在该引脚上的外部电路也不能在上电时序中就把该引脚拉低，因为该低电平将使器件处于复位状态直到为高电平为止。上电延迟后，该引脚可用于外部复位输入或者通用 I/O，这是由熔丝位的配置决定。只有上电复位优先于复位熔丝位的设定，而其他的复位源都不能优先于复位熔丝位的设定。P1.3/RST 也用作 ISP（在线编程）使能。当外部复位引脚为低电平时，ISP 被使能。当熔丝位配置引脚为通用 I/O 时，只有在上电时拉低引脚 P1.3，芯片才能进入 ISP 模式。

10.4 看门狗复位

当看门狗定时器溢出时，将产生一个持续 16 个时钟周期的内部复位脉冲用于复位芯片。看门狗复位可通过设置 WDTCON 中的 WDTOVF 标志来实现。为了防止看门狗复位芯片，必须在看门狗定时溢出前把看门狗复位时序 1EH/E1H 写入到 WDTRST 中。（参考第 57 页“可编程看门狗定时器”的看门狗定时器的详细操作步骤。） See “可编程看门狗定时器” on page 55.

10.5 软件复位

把软件复位时序写入 5AH/A5H 到寄存器 WDRST 时，CPU 将产生一个维持 16 个时钟周期的内部复位脉冲。置位 WDTCON 中的 SWRST 标志将使能软件复位功能。有关软件复位的更多信息请参考第 58 页“软件复位”。 See “软件复位” on page 55.

11. 省电模式

AT89LP216 支持两种不同的低功耗模式：空闲模式和掉电模式。可通过寄存器 PCON 来选择模式。

11.1 空闲模式

置位 PCON 的 IDL 标志可进入空闲模式。空闲模式下，内部 CPU 时钟信号停止，而 CPU 还处于工作状态，包括 RAM、堆栈指针、程序计数器、程序状态字以及累加器。另外，输入 / 输出引脚保持进入空闲模式时刻的逻辑值。空闲模式下保持外设能运行的目的在于让他们产生中断信号而把 CPU 从空闲模式中唤醒。因此，定时器、串口、SPI 和 GPI 等模块能在空闲模式下照常运行。而比较器和看门狗可设定为使能或禁止。任何使能的中断源或者复位信号都能结束空闲模式。当中断唤醒空闲模式时，中断服务将立即执行，并在 RETI 后的下一条可执行指令将是下一次让器件进入空闲模式的指令。

11.2 掉电模式

置位 PCON 的 PD 标志可进入掉电模式。掉电模式下，振荡器将停止振荡，且 Flash 存储器也进入低功耗状态，来尽量降低功耗。因此，只有上电电路继续消耗功耗。在掉电模式中，供电电压可

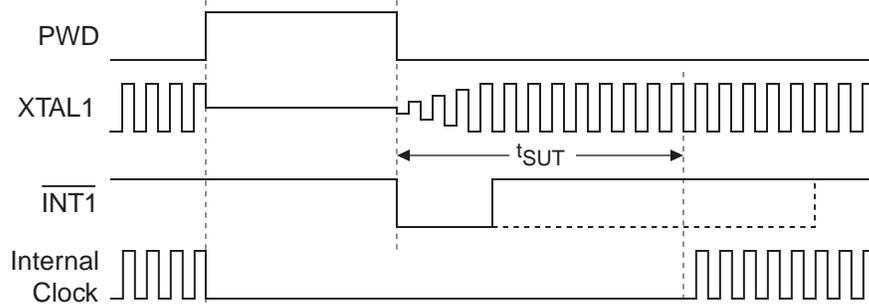
以降低到能持续刷新 RAM 的电平值。这时 VCC 下降后，仍能维持 RAM 中的数据，而 SFR 中的数据却不能保证能维持住。外部复位、上电复位或者某些中断可以使器件从掉电模式中退出。

11.2.1 外部中断唤醒掉电模式

三个外部中断可以配置为唤醒掉电模式。当引脚 XTAL1 或 XTAL2 不用作为晶体振荡器或者外部时钟输入时，可以分别用作外部中断（P3.2）和（P3.3）来退出掉电模式。这时，需要使能外部中断或，并设置为低电平触发中断。当被禁止后，GPI3 也能用来唤醒掉电模式，这时需要使能并设置 GPI3 为低电平检测。

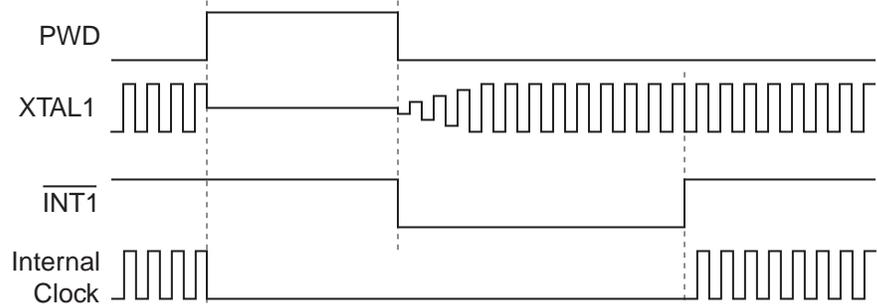
当中断结束掉电模式时，可采用两种唤醒模式。如果 PCON 中的 PWDEX 标志被清零，则如图 11-1 所示，芯片内部产生定时来唤醒芯片，在外部中断引脚电平的下降沿时掉电模式结束，接着振荡器起振，以及内部定时器开始计数，并直到溢出后内部时钟信号才能传送给 CPU 使用，中断服务也是在暂停周期结束后才被执行。配置启动定时熔丝位可控制暂停周期（参考第 14 页上的表 10-1）。此时，外部中断引脚无需维持整个暂停周期的低电平。Table 10-1 on page 13

Figure 11-1. 中断唤醒掉电模式 (PWDEX = 0)



当 PWDEX = “1” 时，唤醒周期是由外部中断信号来控制的。同样，在外部中断引脚电平的下降沿就退出掉电模式，振荡器起振。然而，内部时钟信号只有在中断引脚电平的上升沿后才能被传送给 CPU，如图 11-2。因此，中断引脚必须保持足够长时间的低电平周期使得选用的时钟源稳定后才变为高电平。中断服务程序在跳变为上升沿后就被执行。

Figure 11-2. 中断唤醒掉电模式 (PWDEX = 1)



11.2.2 外部复位唤醒掉电模式

外部复位唤醒掉电模式与 PWDEX = “0” 的中断唤醒情况相似。如图 11-3 所示，在 的下降沿后，掉电模式结束，振荡器起振，以及内部定时器开始计数，并直到溢出后内部时钟信号才能传送给 CPU 使用。配置启动定时熔丝位可控制暂停周期（参考第 14 页上的表 10-1）。如果 在计数溢出前

就跳变为高电平，则在内部时钟重启时产生一个维持 2 个时钟周期的内部复位；否则，器件将保持复位状态直到 变为高电平。

Figure 11-3. 外部复位唤醒掉电模式

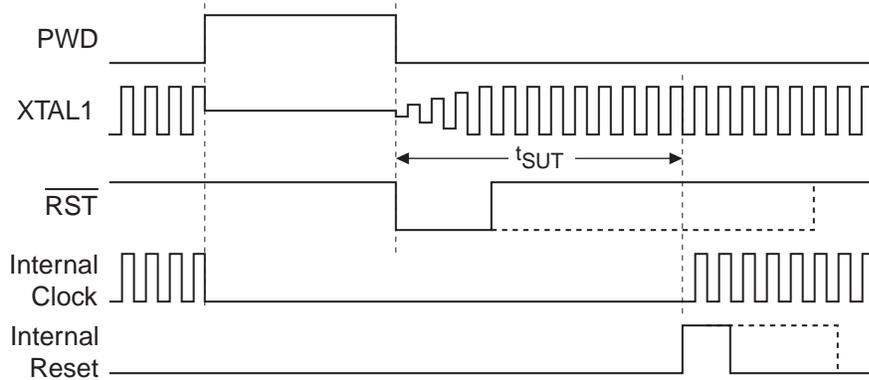


Table 11-1. PCON – 电源控制寄存器

PCON = 87H								复位值 = 000X 0000B
不可位寻址								
Bit	SMOD1	SMOD0	PWDEX	POF	GF1	GF0	PD	IDL
	7	6	5	4	3	2	1	0
符号	功能							
SMOD1	波特率加倍。当置位时，串口模式 1、2 或 3 时波特率加倍。							
SMOD0	帧出错选择。当 SMOD0 = 0，SCON.7 为 SM0；当 SMOD0 = 1，SCON.7 为 FE。注意：只要帧出错，FE 马上被设置，而与 SMOD0 的值无关。							
PWDEX	掉电模式退出。当 PWDEX = 1 时，外部信号控制唤醒；当 PWDEX = 0 时，内部定时计数唤醒。							
POF	断电标志。在上电时置位 POF（即冷复位）。该标志可通过软件置位或清零，而 RST 或 BOD（即热复位）对它不产生影响。							
GF1, GF0	通用标志。							
PD	掉电模式比特位。置位时，激活掉电模式。							
IDL	空闲模式比特位。置位时，激活空闲模式。							

12. 中断

AT89LP216 拥有 7 个中断源：两个外部中断、两个定时中断、一个串口中断、一个通用中断以及一个模拟比较器中断。这些中断源和系统复位一样，在程序空间开始处都有各自的向量地址。可通过单独置位或清零在中断允许寄存器 IE 中相应的比特位，来使能 / 禁止每一个中断源。全局中断使能 / 禁止比特位 EA 也在 IE 寄存器中，它控制所有的中断允许。

每一个中断源（除了模拟比较器中断外）都拥有 4 个优先级，通过置位或清零在中断优先寄存器 IP 和 IPH 中相应的比特位，可以编程为 4 个优先级中的某一等级。而模拟比较器中断固定为最低优先级的中断。处于高优先级的中断服务程序可以中断低优先级的，但不能中断同一优先级级别的或者更高优先级级别的。任何中断源都不能中断优先级最高的中断服务程序的执行。如果在某指令结束时有两个不同级别的中断申请同时发生时，只有优先级高的中断服务被先执行；如果是同一优先级级别的，由于中断矢量地址越低优先级越高，相反中断矢量地址越高其优先级越低，从而内部产生的优先级选择时序就选择高优先级的中断来服务，即先执行中断矢量地址低的那个中断服务程序。注意：只有在同一优先级级别的中断申请同时发生时，才使用优先级选择时序。

通过配置寄存器 TCON 中的 IE0 和 IE1 来产生相应的 IT0 和 IT1 标志，外部中断和可以设置为电平或边沿触发。如果是边沿触发产生中断，则在进入中断服务时硬件清除这些标志；如果是电平触发中断，则外部信号源（而不是片内硬件）控制这些标志。

定时器 0 和定时器 1 由相应的定时 / 计数寄存器中预设溢出值而产生中断（除了定时器 0 的模式 3，因为该模式为波特率发生器）。当进入定时器中断服务程序时，片上的硬件会自动清除中断申请标志。

串口中断由寄存器 SCON 中的 RI 和 TI 标志进行逻辑或运算，以及 SPSR 中的 SPIF 标志决定。当进入中断服务程序时，硬件不清除这些标志，需要用户手工清除。事实上，中断服务的执行就能确定该中断是由 RI、TI 或 SPIF 中的哪一个产生的。因此，在程序中可以进行软件复位相应标志。

通用中断是由寄存器 GPIF 中的八个标志进行逻辑或运算来决定的。在进入中断服务程序时，硬件不清除标志。中断服务程序的执行能确定该中断是由哪一个标志触发，所以在程序中需要软件复位相应标志。如果是电平触发中断，则外部申请源在标志没有被软件清除前是不能进行申请中断服务的。

比较器中断是由寄存器 ACSR 中的 CF 位产生。在进入中断服务程序时，硬件不清除该标志，需要软件清除。

由于 SPI 的中断标志位 SPIF 和在 GPIF 中的通用中断标志位，只能被硬件置位和软件复位。除此之外，大部分的中断标志都可以由软件进行置位或复位，其操作结果与硬件进行置位或复位一致。也就是说，既可以产生中断，也可以取消还未执行的中断。

Table 12-1. 中断向量地址

中断	中断源	地址向量
System Reset	RST or POR or BOD	0000H
External Interrupt 0	IE0	0003H
Timer 0 Overflow	TF0	000BH

Table 12-1. 中断向量地址

中断	中断源	地址向量
外部中断 1	IE1	0013H
定时器 1 溢出	TF1	001BH
串行口中断	RI or TI or SPIF	0023H
通用中断	GPIF	002BH
模拟比较器	CF	0033H

12.1 中断响应时间

中断标志在每个时钟周期由相应的硬件电路置位。中断控制器在执行指令的时钟周期结束时设置这些标志位。如果其中的某个标志在前一个周期就被设置，轮询周期将发现它，假定该中断没有被下面任何一个条件所阻拦，则中断系统将产生 LCALL 指令作为下一条指令来执行相应的中断服务程序：同一优先级的或者更高优先级的中断正在处理；正在执行 RETI 指令或者正在对寄存器 IE、IP 或 IPH 进行写入操作。其中的任何一个都能阻断产生 LCALL 指令而不能进行该中断服务。如果正在执行 RETI 指令，或者对 IE、IP 或 IPH 进行任何的访问，则第二个条件能保证在进行中断服务前将至少执行一条指令。在指令周期结束时，将重复产生轮询周期，而且该轮询值为前一周期的值。如果一个置位了的中断标志被上述其中的一个条件阻断后，则就算后来该条件已经消除了，该中断也将不再执行。换言之，置位的中断标志如果没有被马上服务，则该标志无效。因为每个轮询周期都是重新进行的。

如果申请被激活，且条件也满足，则该中断申请服务产生一个硬件调用函数作为下一条指令来执行，该调用函数占用 4 个周期。因此，从激活中断到开始执行中断申请至少需要 5 个周期。如果中断申请被上述的条件阻断，则所需的响应时间会更长。如果同一优先级或更高优先级的中断正在执行，则等待时间由正在执行的中断服务决定。如果正在执行的指令还未结束，则额外等待时间就不大于 3 个周期，因为最多只能为 4 个周期。如果正在执行 RETI 指令，或者访问 IE 或 IP 时，则额外等待时间就不大于 7 个周期（最大 3 个周期去完成该指令，再外加 4 个周期去完成下一条指令）。因此，在单个中断系统中，一个中断响应时间总为 5 ~ 13 个周期，如图 12-1 和 12-2。

Figure 12-1. 最小中断响应时间

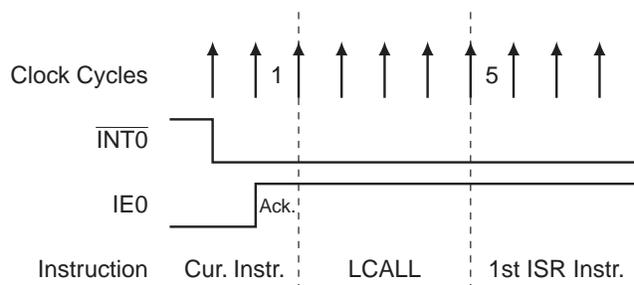


Figure 12-2. 最大中断响应时间

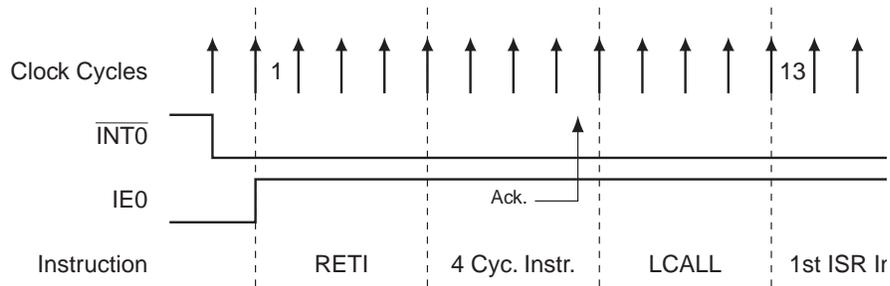


Table 12-2. IE – 中断使能寄存器

IE = A8H		复位值 = 0000 0000B						
可位寻址								
	EA	EC	EGP	ES	ET1	EX1	ET0	EX0
Bit	7	6	5	4	3	2	1	0
符号	功能							
EA	全局使能 / 禁止。当 EA = 0，禁止所有中断源；当 EA = 1，通过置位 / 复位各个中断使能位来允许各个中断源。							
EC	比较器中断使能位							
EGP	通用中断使能位							
ES	串口中断使能位							
ET1	定时器 1 中断使能位							
EX1	外部中断 1 使能位							
ET0	定时器 0 中断使能位							
EX0	外部中断 0 使能位							

Table 12-3. IP – 中断优先级寄存器

IP = B8H		复位值 = X000 0000B						
可位寻址								
	–	–	PGP	PS	PT1	PX1	PT0	PX0
Bit	7	6	5	4	3	2	1	0
符号	功能							
PGP	通用中断优先级低字节							
PS	串口中断优先级低字节							
PT1	定时器 1 中断低字节							
PX1	外部中断 1 低字节							
PT0	定时器 0 中断低字节							
PX0	外部中断 0 低字节							

Table 12-4. IPH - 中断优先级高半位寄存器

IPH = B7H		复位值 = X000 0000B						
不可位寻址								
	—	—	PGH	PSH	PT1H	PX1H	PT0H	PX0H
Bit	7	6	5	4	3	2	1	0

符号	功能
PGH	通用中断优先级高字节
PSH	串口中断优先级高字节
PT1H	定时器 1 中断高字节
PX1H	外部中断 1 高字节
PT0H	定时器 0 中断高字节
PX0H	外部中断 0 高字节

13. I/O 端口

AT89LP216作为通用I/O引脚数一般为9~12个。具体的个数由表13-1中的时钟源与复位源的选择来决定。所有引脚的电平都采用 5V 电平逻辑。因此，即使 VCC 低到 3V，引脚都能上拉或驱动最大到 5.5V。

Table 13-1. I/O 引脚配置表

时钟源	复位源	可用 I/O 数量
外部晶体或振荡电路	采用外部复位	11
	不采用外部复位	12
外部时钟信号	采用外部复位	12
	不采用外部复位	13
内部 RC 振荡器	采用外部复位	13
	不采用外部复位	14

13.1 端口配置

AT89LP216 所有引脚都可以配置为以下 4 种模式中的一种：准双向（标准 8051 端口）、推挽式输出、漏极开路输出、输入模式。如表 13-2，端口模式可通过软件一个一个引脚进行配置，而三态

端口用户熔丝位决定端口引脚的默认状态。如果熔丝位使能，则在器件复位时所有端口引脚都处于输入模式。否则，除了引脚 P1.0 和 P1.1 外，在器件复位时所有引脚都处于准双向状态并被弱上拉。每一个引脚都具有一个施密特触发器，用于提高输入噪声容限。在掉电模式时，除了 P1.3、P3.2 和 P3.3 要用于唤醒器件外，所有输入端的施密特触发器都被禁止因此，在掉电模式时，引脚 P1.3、P3.2 和 P3.3 不应悬空。

Table 13-2. 引脚模式配置表

PxM0.y	PxM1.y	引脚模式
0	0	准双向
0	1	推挽式输出
1	0	输入（高阻抗）
1	1	漏极输出

13.1.1 准双向输出

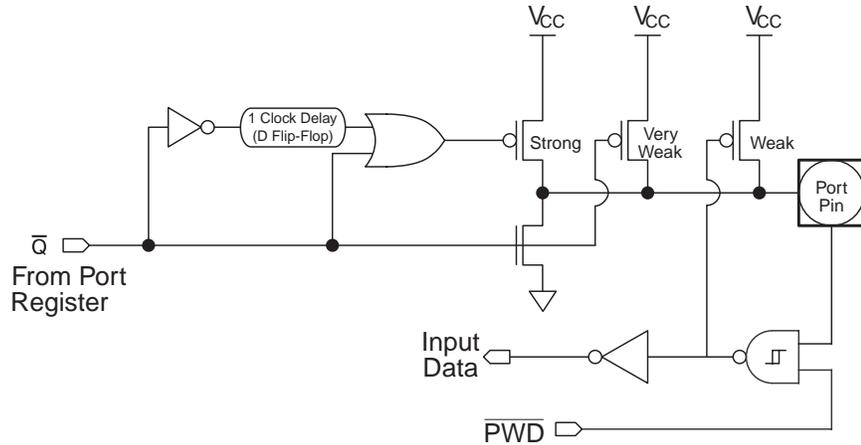
准双向输出模式与标准 8051 端口相似。该模式下，无需配置端口，就既可用于输入，也可以用作输出。因为在输出高电平时是弱驱动的，所以允许外部器件把该引脚拉低；而输出为低电平时是强驱动的，它能吸收大电流。因此，在准双向输出时，用三个上拉晶体管来完成不同需要。

其中一个，称为“非常弱上拉”。只要引脚锁存进逻辑电平“1”，管子打开工作。在引脚悬空时由于产生一个微弱的灌电流而把引脚上拉为高电平状态。

第二个，称为“弱上拉”。只要引脚锁存进逻辑电平“1”且该引脚也处于逻辑电平“1”时，管子打开工作。该上拉管子能提供较大的灌电流从而引脚输出逻辑“1”。如果该引脚被外部器件拉低，该上拉晶体管就关闭，所以只有非常弱的上拉效果。为了在这种情况下能把该引脚拉低，外部器件必须能吸足够大的电流而给弱上拉管子提供过强功率，使得该引脚被拉到低于阈值电压，从而表现为低电平。

第三个，称为“强上拉”。当端口锁存器由逻辑“0”变为“1”时，它能加快准双向端口从低电平变为高电平的瞬态响应。此时，管子打开工作，并在两个时钟周期后很快地将管脚拉高。准双向口的配置如图 13-1。在掉电模式时，引脚 P1.3、P3.2 和 P3.3 的电路不能被禁止（如图 13-3）。

Figure 13-1. 准双向输出



13.1.2 输入模式

输入模式配置如图 13-2，输出驱动是三态的，而输入端包含了一个施密特触发器输入来提高输入噪声容限。在掉电模式时，引脚 P1.3、P3.2 和 P3.3 的电路不能被禁止（如图 13-3）。即使在 VCC 较低的情况下，驱动引脚的电平值可以高达 5.5V。但值得注意的是，施密特触发器的输入有可能被该 VCC 置位。

Figure 13-2. I 输入模式

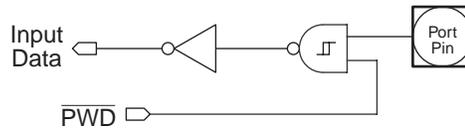
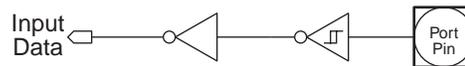


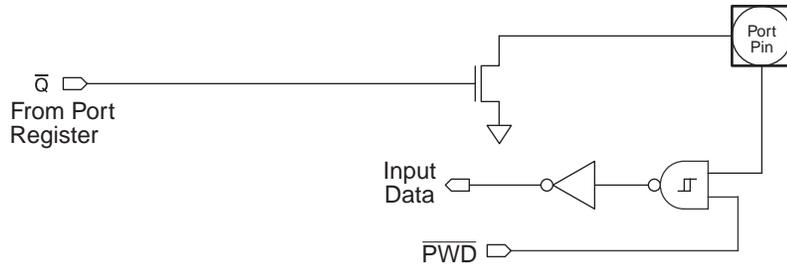
Figure 13-3. 引脚 1.3、P3.2 和 P3.3 的输入模式



13.1.3 漏极输出模式

在漏极输出模式下，当端口锁存进一个逻辑“0”时，端口的上拉晶体管全部关闭，而只有下拉晶体管有效。为了能输出逻辑电平，这时一般使用一个外部上拉电阻连接到 VCC。输出电平时与准双向情况类似。漏极输出的配置如图 13-4。在掉电模式时，引脚 P1.3、P3.2 和 P3.3 的电路不能被禁止（如图 13-3）。即使在 VCC 较低的情况下，驱动引脚的电平值可以高达 5.5V。但值得注意的是，施密特触发器的输入有可能被该 VCC 置位。

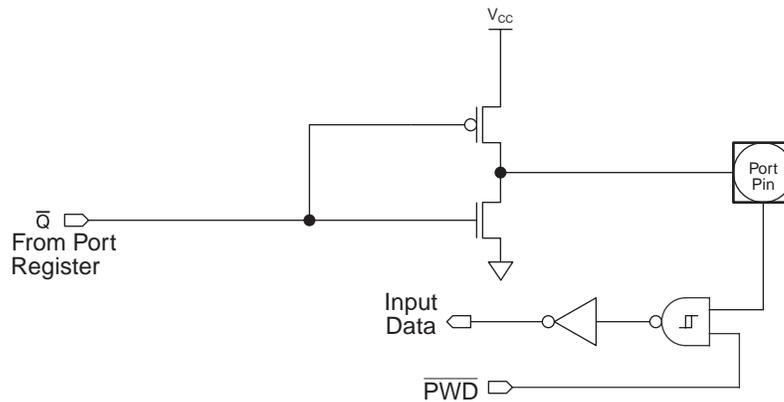
Figure 13-4. 漏极输出模式



13.1.4 推挽式输出模式

推挽式输出的配置具有与漏极输出和准双向输出一致的下拉结构，但在端口锁存进逻辑“1”时它能提供持续强上拉。在需要更多的灌电流从端口输出时，一般采用推挽式输出模式。推挽式输出的端口配置如图 13-5。在掉电模式时，引脚 P1.3、P3.2 和 P3.3 的电路不能被禁止（如图 13-3）。

Figure 13-5. 推挽式输出模式



13.2 端口 1 模拟功能

AT89LP216 组建一台模拟比较器。为了达到最好的模拟性能和最小的能量消耗。用于模拟功能的引脚的数字输入和输出都可以被设置为无效。如 22 页所描述的“端口配制”，当端口引脚被设为数据输入时，数据输出是无效的。只要模拟比较器通过置位 ACSR 中的 CEN 位，引脚 P1.1 和 P1.0 上的数据输入就是无效的。CEN 强制 PWD 在 P1.0 和 P1.1 脚上输入低电平。从而将施密特触发电路置为无效。无论三态门重启之后的状态。P1.0 和 P1.1 引脚始终默认为输入模式。

13.3 端口读 - 修改 - 写

从端口读取数据时，使用哪种指令将决定读到的是端口的状态还是端口寄存器的值。简单的读指令直接访问端口的引脚。读 - 修改 - 写指令，就是读取一个数值，或许进行修改然后写回。这将始终访问端口寄存器。这可能包括位指令比如：CLR 或 SETB，实际上这些指令也要访问整个端口修改其中一位数据，然后写回整个端口。Table 13-3 是一个读 - 修改 - 写指令的详细的表。

Table 13-3. 端口读 - 修改 - 写指令

助记符	指令	范例
ANL	逻辑与	ANL P1, A
ORL	逻辑或	ORL P1, A
XRL	逻辑异或	XRL P1, A

Table 13-3. 端口读 - 修改 - 写指令

助记符	指令	范例
JBC	置位则跳转并清零该位	JBC P3.0, LABEL
CPL	位取反	CPL P3.1
INC	自加	INC P1
DEC	自减	DEC P3
DJNZ	自减, 不等于 0 时跳转	DJNZ P3, LABEL
MOV PX.Y, C	移动进位至端口 x 中的 y 位	MOV P1.0, C
CLR PX.Y	清零端口 x 中的 y 位	CLR P1.1
SETB PX.Y	置位端口 x 中的 y 位	SETB P3.2

13.4 端口复用功能

AT89LP216 大多数的普通 I/O 引脚可以连接多个外设。Table 13-5 列出了端口引脚复用功能。复用功能与处于逻辑与状态下的引脚相联系，为了使一个端口引脚具有复用功能。那个引脚必须置位相应的寄存器。否则这个引脚的输出始终是“0”。此外，在每个引脚被使用之前都必须对他进行置位，来选择适当的输入输出模式，这种模式是它的外设所决定的。

Table 13-4 表示如何配置一个用于复用功能的普通引脚

Table 13-4. 为端口 x 的 y 引脚配制复用功能

PxM0.y	PxM1.y	Px.y	I/O 模式
0	0	1	双向 (内部上拉)
0	1	1	输出
1	0	X	输入
1	1	1	双向 (外部上拉)

Table 13-5. 端口引脚复用功能

端口引脚	配制位		复用功能	备注
	PxM0.y	PxM1.y		
P1.0	P1M0.0	P1M1.0	AIN0	仅输入
			GPI0	
P1.1	P1M0.1	P1M1.1	AIN1	仅输入
			GPI1	
P1.2	P1M0.2	P1M1.2	GPI2	
P1.3	P1M0.3	P1M1.3	GPI3	$\overline{\text{RST}}$ 必须失效

Table 13-5. 端口引脚复用功能

端口引脚	配制位		复用功能	备注
	PxM0.y	PxM1.y		
P1.4	P1M0.4	P1M1.4	SS	
			GPI4	
P1.5	P1M0.5	P1M1.5	MOSI	
			GPI5	
P1.6	P1M0.6	P1M1.6	MISO	
			GPI6	
P1.7	P1M0.7	P1M1.7	SCK	
			GPI7	
P3.0	P3M0.0	P3M1.0	RXD	
P3.1	P3M0.1	P3M1.1	TXD	
P3.2	P3M0.2	P3M1.2	INT0	仅内部 RC 振荡器
P3.3	P3M0.3	P3M1.3	INT1	内部 RC 振荡器或外部时钟源
			CLKOUT	
P3.4	P3M0.4	P3M1.4	T0	
P3.5	P3M0.5	P3M1.5	T1	
P3.6	没有配制		CMPOUT	引脚设定为比较器

14. 增强定时 / 计数器

AT89LP216 有两个 16 位的定时器 / 计数器。定时器 0 和定时器 1。作为定时器每个时钟周期增值。如此，定时器计算每个时钟周期。由于一个时钟周期由一个晶振周期组成。所以计数周期就是时钟周期。使用定时器预测计数器，通过 1 到 16 之间的一个数值 (见 Table 9-2 on page 12)。定时器速率可以被预测。两个定时器使用同一个。

作为一个计数器，寄存器根据输入引脚 T0 或 T1 的电平高低变化而自加。每个时钟周期都对外部输入信号做抽样。当采样信号在一个周期是高电平，在下一个周期是低平时，寄存器数值加 1。在接下来被检测到的转变周期中，新的计数值出现在寄存器里，由于需要两个时钟周期去识别一个高低电平的变化，所以最大的采样频率是 1/2 的振荡器频率。对输入信号的占空比没有限制，但是它至少应该持续一个完整的周期，这样在它变化前至少能被采样到一次。

此外对于定时器 0 和定时器 1 来说，定时或计数的模式有 4 种：变宽度定时器，16 位自动重装定时器，8 位自动重装定时器，和可拆分的定时器特殊功能寄存器 TMOD 里的 C/T 位选择定时器或计数器的功能。TMOD 里的 (M1, M0) 选择操作模式。

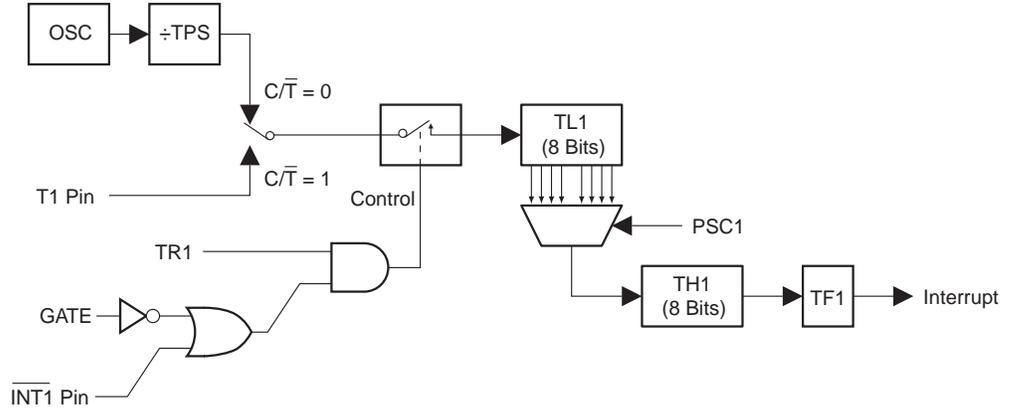
14.1 模式 0 — 可变宽度的定时 / 计数器

在模式 0 情况下，两个定时器都是带有一个 8 位可预分频的计数器。通过设置 TCONB 中的 PSC 位预分频器可以在 1 到 8 位之间变化。供给定时器 9 到 16 位的变化范围，通过默认值把定时器配置为一个 13 位定时器。这与 80c51 里的模式 0 是一样的。当计数器从全“1”到全“0”翻转时，它置位定时器中断标志位 TF1。对定时器而言。当 TR1 = 1 并且 GATE = 0 或者 INT1 = 1 时，计数器的输入被激活。设定 GATE = 1 则允许定时器外部中断 INT1 控制用于脉冲宽度测量在特殊功能寄存器 TCON 中 TR1 是一个控制位。门空位 GATE 在模式控制寄存器 TCON 中 13 位寄存器由 TH1 的 8 位和 TH1 的 5 位构成。TH1 的高 3 位是不变的，应该被忽略。设置的启动标志位 (TR1) 对寄存器不进行清除。

$$\text{Mode 0: Time-out Period} = \frac{256 \times 2^{PSC0+1}}{\text{Oscillator Frequency}} \times (TPS+1)$$

Note: 在模式 0 情况下 RH1/RL1 对定时器 1 而言不是必要的，可以作为临时存储寄存器。

Figure 14-1. 定时 / 计数器 1 模式 0: 可变宽度计数器



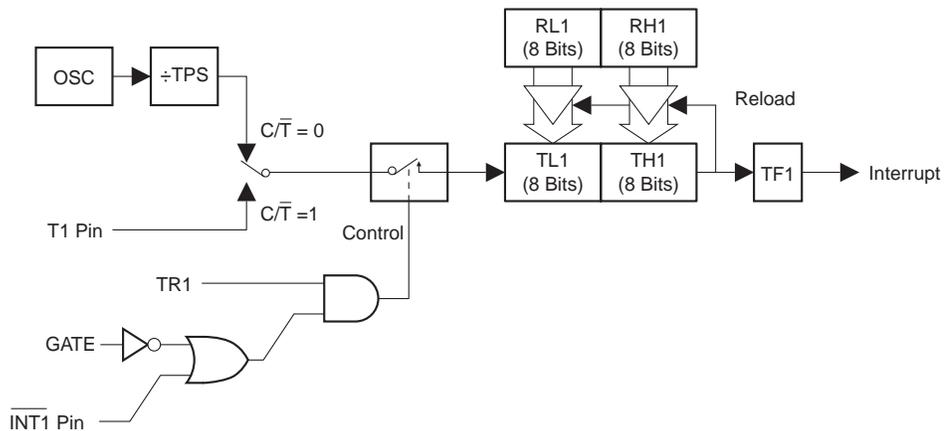
模式 0 情况下。对定时器 0 和定时器 1 的操作是一样的，只是 TR0 和 TF0 还有 $\overline{\text{INT0}}$ 代替了图 Figure 14-1 上相应的定时器 1 的信号。这里有两个不同的门控制位，一个用于定时器 1(TMOD.7) 另外一个用于定时器 0(TMOD.3)。引脚 $\overline{\text{INT0}}$ 和 $\overline{\text{INT1}}$ 分享 XTAL 处的振荡器频率。当用内部 RC 作为系统时钟时，它们仅被用于门控制功能。

14.2 模式 1-16 位自动重装定时 / 计数器

在模式 1 的情况下，配置定时器用于 16 位自动重装。定时器的寄存器运行在 16 位模式下。16 位的重装值储存在 (RH1/RL1) 这两个寄存器中，时钟被用于组合高、低寄存器 (TH1/TL1)。收到时钟脉冲，定时器就把他们相加起来，0000H, 0001H, 0002H 依次类推，溢出出现在 FFFH 到 0000H 的转变时，此时寄存器被装入来自 RH1/RL1 的值。溢出位 TCON 被置位。见 Figure 14-2 重装的值默认是 0000H，提供给全部 16 位的定时器周期，和标准的 8051 相同。模式 1 操作对定时器 0 和定时器 1 是相同的。

$$\text{Mode 1: Time-out Period} = \frac{(65536 - \{RH0, RL0\})}{\text{Oscillator Frequency}} \times (TPS+1)$$

Figure 14-2. 定时 / 计数器 1 模式 1: 16 位自动重装

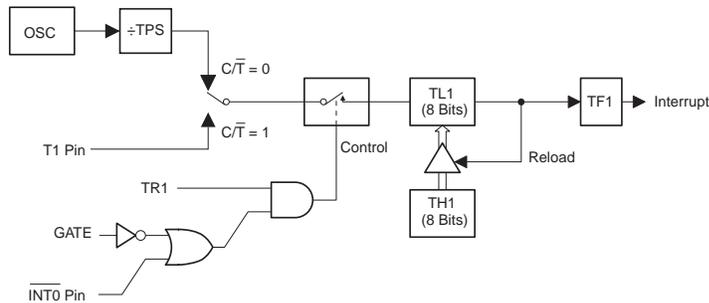


14.3 模式 2 8 位自动重装定时 / 计数器

模式 2 下配置的定时器是一个有自动重装功能的 8 位计数器 (TL1)。如图 Figure 14-3 所示, 从 TL1 溢出不仅需要置位 TF1, 还需要软件设置好 TH1 的值, 重装入 TL1。在装入保持 TH1 不变, 模式 2 操作对定时器 0 和定时器 1 是一样的。

$$\text{Mode 2: Time-out Period} = \frac{(256 - TH0)}{\text{Oscillator Frequency}} \times (TPS + 1)$$

Figure 14-3. 定时器 1 模式 2 8 位自动重装



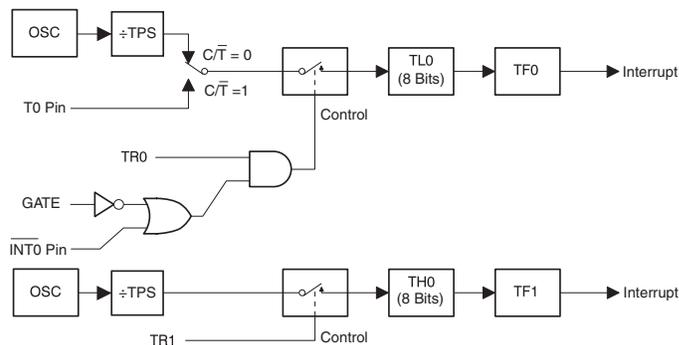
Note: 在模式 2 情况下 RH1/RL1 对定时器 1 而言是没有用的, 可以作为临时寄存器。

14.4 模式 3 可拆分定时器

定时器 1 在模式 3 情况下仅仅保留它的计数功能。效果和设置 TR1 = 0 一样。模式 3 下的定时器 0 设置了 TL0 和 TH0 作为两个单独的计数器。模式 3 下, 定时器 0 如图 Figure 14-4 所示。TL0 使用了定时器 0 的控制位: C/T, GATE, TR0, INT0, 和 TF0。TH0 被锁定成定时器功能, (计数器周期) 接替来自定时器 1 中的 TR1 和 TF1 的功能。因此 TH0 控制定时器 1 的中断, 当定时器 0 处于模式 3 状态时, 定时器 1 仍然遵循它在模式控制寄存器 TMOD 中的设置。

因为需要一个而外的 8 位定时器或计数器, 所以才有了模式 3。定时器 0 在模式 3 下的应用使得 AT89LP216 有了 3 个定时/计数器。当定时器 0 处于模式 3 时, 定时器 1 可以通过转换进入或离开模式 3, 启动或停止。这种情况下, 定时器 1 仍然能被使用, 可以作为波特率发生器不需要任何中断场合都可以使用

Figure 14-4. 定时 / 计数器 0 模式 3: 双 8 位计数器



Note: 在模式 3 情况下 RH1/RL1 对定时器 1 而言是没有用的, 可以作为临时寄存器。

Table 14-1. TCON – 定时 / 计数器控制寄存器

TCON = 88H		复位值 = 0000 0000B						
可位寻址								
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Bit	7	6	5	4	3	2	1	0

符号	功能
TF1	定时器 1 溢出标志位。定时器 / 计数器的溢出由硬件置位。在处理器转向中断程序时由硬件清除。
TR1	定时器 1 启动控制位。由软件设置 / 清除来启动 / 停止定时器 / 计数器的工作。
TF0	定时器 0 溢出标志位。定时器 / 计数器的溢出由硬件置位。在处理器转向中断程序时由硬件清除。
TR0	定时器 0 启动控制位。由软件设置 / 清除来启动 / 停止定时器 / 计数器的工作。
IE1	中断 1 的边缘标志位。当检测到外部中断边缘时由硬件置位。在处理中断时自动清除。
IT1	中断 1 触发方式控制位。由软件设置 / 清除来指定下降沿 / 低电平触发外部中断。
IE0	中断 0 的边缘标志位。当检测到外部中断边缘时由硬件置位。在处理中断时自动清除。
IT0	中断 0 触发方式控制位。由软件设置 / 清除来指定下降沿 / 低电平触发外部中断。

Table 14-2. TMOD: 定时 / 计数器模式控制寄存器

TMOD = 88H				复位值 = 0000 0000B				
不可位寻址								
	GATE	$C\bar{T}$	M1	M0	GATE	$C\bar{T}$	M1	M0
	7	$\bar{6}$	5	4	3	$\bar{2}$	1	0
	定时器 1				定时器 0			
Gate	当置位时通控制。定时器 / 计数器 x 仅在 \overline{INTx} 引脚为高电平并且 TRx 控制引脚被置位时启动。当清除时，无论 TRx 控制位是否被置位定时器都能启动。				定时器 0 门控位			
$C\bar{T}$	定时器或计数器选择位清除时定时器工作（来自外部系统时钟的输入），置位时计数器工作（来自 Tx 输入引脚的输入）。				定时器 0 定时 / 计数选择位			
M1	定时器 1 模式位 1				模式位 0 M1 位			
M0	定时器 1 模式位 0				模式位 0 M0 位			
M1	M0	Mode	操作模式					
0	0	0	可调的 9-16 位定时器模式 8 位定时器 / 计数器。THx 和 TLx 作为 1 - 8 位预分频器。					
0	1	1	可自动再装入的 16 位模式。 16 位定时器 / 计数器 THx 和 TLx 联结；这里没有预分频器。					
1	0	2	8 位自动再装入模式。 可自动再装入的 8 位定时器 / 计数器 THx 保持一个值，TLx TLx 每次溢出后再装入这个值。					
1	1	3	可拆分定时器模式。 (定时器 0) TL0 是一个 8 位定时器 / 计数器，由标准的定时器 0 控制位控制。TH0 是一个 8 位定时器仅由定时器 1 的控制位控制。					
1	1	3	(定时器 1) 定时器 / 计数器 1 停止。					

Timer 寄存器	作用	地址	位寻址
TCON	Control	88H	Yes
TMOD	Mode	89H	No
TL0	Timer 0 low-byte	8AH	No
TL1	Timer 1 low-byte	8BH	No
TH0	Timer 0 high-byte	8CH	No
TH1	Timer 1 high-byte	8DH	No
TCONB	Mode	91H	No
RL0	Timer 0 reload low-byte	92H	No
RL1	Timer 1 reload low-byte	93H	No
RH0	Timer 0 reload high-byte	94H	No
RH1	Timer 1 reload high-byte	95H	No

Table 14-3. TCONB - 定时器 / 计数器 控制寄存器 B

TCONB = 91H								复位值 = 0010 0100B
不可位寻址								
	PWM1EN	PWM0EN	PSC12	PSC11	PSC10	PSC02	PSC01	PSC00
Bit	7	6	5	4	3	2	1	0

符号	功能
PWM1EN	配置定时器 1 用于 T1 (P3.5) 引脚上的脉冲宽度调制输出。
PWM0EN	配置定时器 0 用于 T0 (P3.4) 引脚上的脉冲宽度调制输出。
PSC12 PSC11 PSC10	预分频器用于模式 0 下的定时器 1。TL1 中起作用的位的数量为 PSC1 + 1。重新复位后 PSC1 = 100B，这使得 TL1 的 5 位能与 AT89S2051 中的 13 位模式 0 兼容。
PSC02 PSC01 PSC00	预分频器用于模式 0 下的定时器 0。TL0 中起作用的位的数量为 PSC0 + 1。重新复位后 PSC0 = 100B，这使得 TL0 的 5 位能与 AT89S2051 中的 13 位模式 0 兼容。

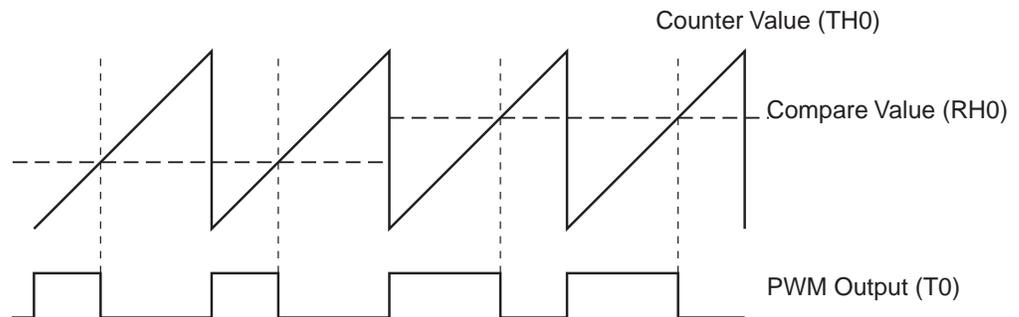
14.5 脉冲宽度调制

在 AT89LP216 上，分别设置 TCONB 中的 PWM0EN 位或 PWM1EN 位，定时器 0 和定时器 1 可独立地配置成 8 位不对称（边缘排列）的脉冲宽度调制器（PWM）。在 PWM 模式下产生的波形在定时器的输入引脚 T0 或 T1 上输出。因此，在 PWM 模式下 C/T 位必须置“0”，并且 T0 (P3.4) 和 T1 (P3.5) 必须设置成输出方式。处于 PWM 模式，定时器溢出标志位和中断将继续工作，定时器 1 仍然可以为异步串行通信口（UART）产生波特率。每个 PWM 信道有四种模式，由模式控制寄存器（TMOD）中的模式位选择。

定时器 0 在 PWM 模式 0 下的波形样例如图 14-5 所示。当 RHO 存储 8 位比较值时，TH0 作为 8 位计数器使用。TH0 为 00H 时 PWM 输出置高电平；TH0 计数达到 RHO 中存储值时，PWM 输出置低电平。

因此脉冲宽度与 RHO 中的值成比例。为了防止短时脉冲波形干扰，向 RHO 写入仅在 TH0 从 FFH 到 00H 溢出时有效。RHO 置 00H 将保持 PWM 输出为低电平。

Figure 14-5. 不对称的脉冲宽度调制



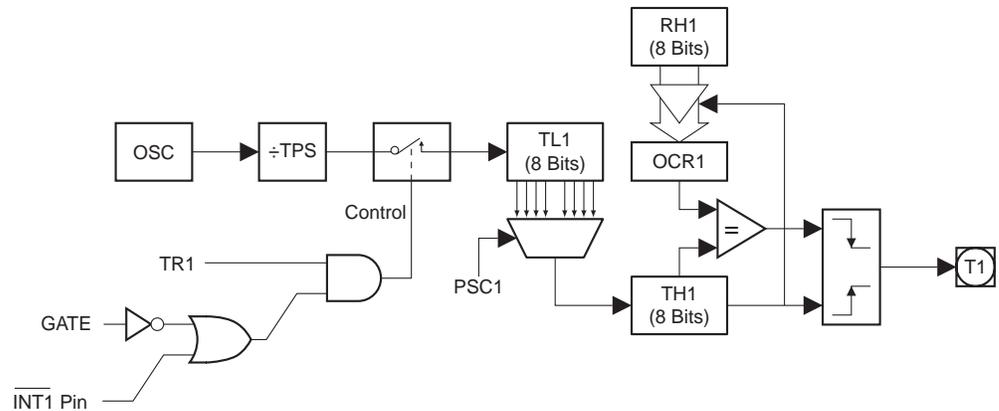
14.5.1 模式 0 - 带有 8 位对数预分频器的 8 位 PWM

在模式 0 下，TLx 作为一个对数预分频器驱动 8 位计数器 THx（见图 14-6）。TCONB 中的 PSCx 位控制预分频器值。在 THx 溢出时，RHx 中的占空比值转给 OCRx，输出置高电平。当 THx 中的计数与 OCRx 匹配时，输出引脚清除为低电平。下面的公式给出了定时器 0 在 PWM 模式 0 下的输出频率和占空比。定时器 1 在 PWM 模式 0 下的情况与定时器 0 相同。

$$\text{Mode 0: } f_{out} = \frac{\text{Oscillator Frequency}}{256 \times 2^{PSC0+1}} \times \frac{1}{TPS+1}$$

$$\text{Mode 0: } \text{Duty Cycle \%} = 100 \times \frac{RH0}{256}$$

Figure 14-6. 定时器 / 计数器 1 PWM 模式 0



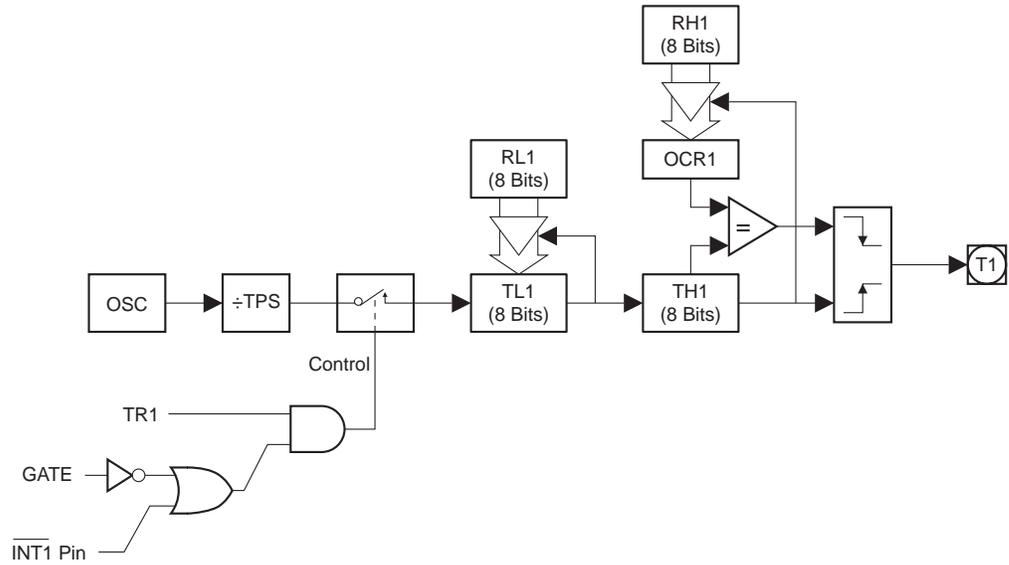
14.5.2 模式 1 - 带有 8 位线性预分频器的 8 位 PWM

在模式 1 下，TLx 供给线性预分频（linear prescaling）一个来自 RLx 的 8 位自动再装入（见图 14-7）。当 TLx 溢出时，TLx 装入 RLx 的值。THx 作为 8 位计数器使用。当 THx 溢出时，RHx 中的占空比值转给 OCRx，输出置高电平。当 THx 中的计数与 OCRx 匹配时，输出引脚清除为低电平。下面的公式给出了定时器 0 在 PWM 模式 1 下的输出频率和占空比。定时器 1 在 PWM 模式 0 下的情况与定时器 0 相同。

$$\text{Mode 1: } f_{out} = \frac{\text{Oscillator Frequency}}{256 \times (256 - RLO)} \times \frac{1}{TPS+1}$$

$$\text{Mode 1: } \text{Duty Cycle \%} = 100 \times \frac{RH0}{256}$$

Figure 14-7. 定时器 / 计数器 1 PWM 模式 1

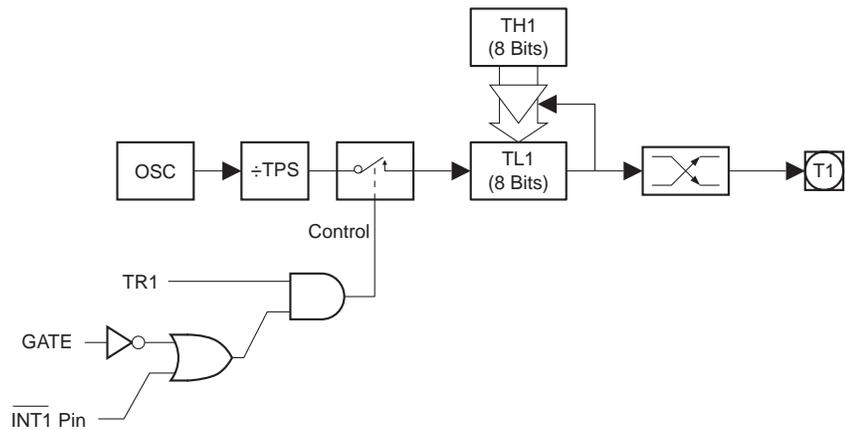


14.5.3 模式 2 - 8 位频率产生

定时器 0 在 PWM 模式 2 中作为一个 8 位自动再装入定时器运行，和正常模式 2 的相同，例外的是输出引脚 T0 在每次 TLO 溢出时被触发（见图 14-8）。定时器 1 在 PWM 模式 2 下和定时器 0 是一样的。PWM 模式 2 可用于输出频率可调的矩形波。THx 充当 8 位计数器。下面公式给出了定时器 0 在 PWM 模式 2 下的输出频率。

$$\text{Mode 2: } f_{out} = \frac{\text{Oscillator Frequency}}{2 \times (256 - TH0)} \times \frac{1}{TPS + 1}$$

Figure 14-8. 定时 / 计数器 1 PWM 模式 2



Note: {RH0 & RL0} / {RH1 & RL1} 对定时器 0 / 定时器 1 而言在 PWM 模式 2 下不是必需的，可以作为暂时存储寄存器使用。

14.5.4 模式 3 - 可拆分的 8 位 PWM

定时器 1 在 PWM 模式 3 下仅仅保持它的计数。效果和设置 TR1 = 0 一样。在某种意义上，定时器 0 在 PWM 模式 3 下设立了 TLO 和 TH0，作为两个分离的 PWM 计数器，与正常模式 3 类似。PWM 模式

3 作用于定时器 0 如图 14-9 所示。在 PWM 模式 3 下只有定时器预分频器可用于改变输出频率。TL0 可以使用定时器 0 的控制位：GATE, TR0, INTO, PWMOEN 和 TF0. TH0 锁定为定时器功能，使用 TR1, PWM1EN 和 TF1. RLO 为 TL0 提供占空比，RH0 为 TH0 提供占空比。

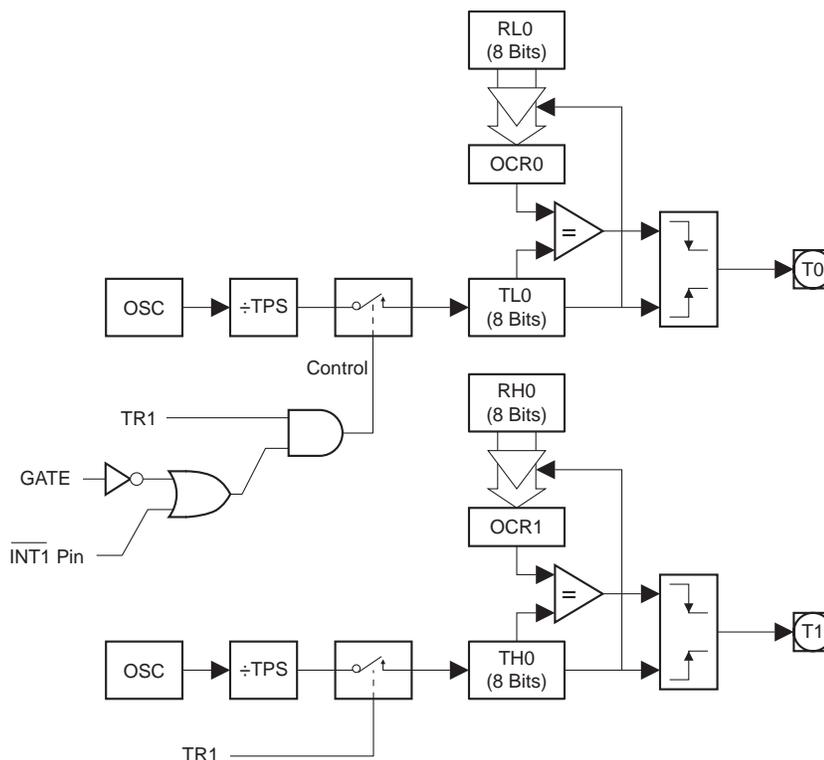
因为应用需要一个单一的 PWM 信道和两个定时器，或两个 PWM 信道和一个额外的定时器或计数器，所以产生了 PWM 模式 3。PWM 模式 3 下的定时器 0，使 AT89LP216 可能出现有三个定时器 / 计数器。当定时器 0 处于模式 3 时，定时器 1 可以通过转换进入和离开自己的模式 3 而被启动和停止。在这种情况下，定时器 1 仍然能使用，连续的端口可作为波特率产生器；任何不需要中断的场合都可以用。下面的公式给出了定时器 0 在 PWM 模式 3 下的输出频率和占空比

$$\text{Mode 3: } f_{out} = \frac{\text{Oscillator Frequency}}{256} \times \frac{1}{TPS+1}$$

$$\text{Mode 3, T0: } \text{Duty Cycle \%} = 100 \times \frac{RLO}{256}$$

$$\text{Mode 3, T1: } \text{Duty Cycle \%} = 100 \times \frac{RH0}{256}$$

Figure 14-9. 定时器 / 计数器 0 PWM 模式 3



15. 外部中断

配置 AT89LP216 使用内部 RC 振荡器，此时 XTAL1 和 XTAL2 可以用作 $\overline{\text{INT0}}$ 和 $\overline{\text{INT1}}$ 的外部中断源。通过将寄存器 TCON 中的 IT1 或 IT0 置位或清空，外部中断可以编程设置为电平激发或跃迁激发。

如果 $IT_x = 0$ ，当检测到 $\overline{INT_x}$ 管脚为低时，外部中断 x 被触发。如果 $IT_x = 1$ ，外部中断为边缘触发。在这种模式下，如果 $\overline{INT_x}$ 管脚的连续采样显示，一个周期为高电平，下一个周期为低电平， $TCON$ 的中断请求标志 IE_x 被置位。然后标志位 IE_x 请求中断。因为外部中断管脚每个时钟周期采样一次，因为输入的高电平或低电平至少应该保持 2 个振荡时期，以确保采样的进行。

如果外部中断是跃迁激活的，外部中断源必须要保持请求管脚高电平至少两个时钟周期，再保持该管脚低电平至少两个时钟周期，以确保该跃迁能够被检测到，使中断请求标志 IE_x 能被置位。如果在边缘触发模式下，调用服务程序时， IE_x 将会由 CPU 自动清空。如果外部中断是水平激活，那么外部源必须保持请求有效，直到真正生成请求中断。因此外部中断源必须在中断服务程序完成之前接触请求，否则将会产生另一个中断。

16. 通用中断

通用中断 (General-purpose Interrupt-GPI) 功能在端口 1 提供 8 个可配置的外部中断。每个端口管脚可以检测高 / 低电平或者正 / 负沿。寄存器 $GPIEN$ 选择使端口 1 的某些位有效，产生一个中断。寄存器 $GPMOD$ 和 $GPLS$ 决定每一个单独管脚的模式。 $GPMOD$ 在电平敏感触发和边缘触发模式中选择一种， $GPLS$ 在高 / 低电平模式和正 / 负沿模式中选择一种。端口 1 的管脚每个时钟周期采样一次。在电平敏感模式下，有效电平必须在两个连续采样中出现，并且在产生中断之前出现。在边缘触发模式下，如果电平值从一个采样转变为下一个采样值，就会检测到跃迁。当一个管脚检测到产生中断的条件，并且该管脚是激活的，那么寄存器 $GPIF$ 的相应标志位被置位。 $GPIF$ 的标志位必须由软件清空。

Table 16-1. $GPMOD$ - 通用中断模式寄存器

$GPMOD = 9AH$								复位值 = 0000 0000B
不可位寻址								
	$GPMOD7$	$GPMOD6$	$GPMOD5$	$GPMOD4$	$GPMOD3$	$GPMOD2$	$GPMOD1$	$GPMOD0$
Bit	7	6	5	4	3	2	1	0
$GPMOD.x$ 0 = 电平敏感中断 P1.x 1 = 边沿触发敏感中断 P1.x								

Table 16-2. $GPLS$ - 通用中断电平选择寄存器

$GPLS = 9BH$								复位值 = 0000 0000B
不可位寻址								
	$GPLS7$	$GPLS6$	$GPLS5$	$GPLS4$	$GPLS3$	$GPLS2$	$GPLS1$	$GPLS0$
Bit	7	6	5	4	3	2	1	0

Table 16-2. GPLS - 通用中断电平选择寄存器

GPMOD.x 0 = P1.x 检测低电平或下降沿
 1 = P1.x 检测高电平或上升沿

Table 16-3. GPIEN - 通用中断使能寄存 器

GPIEN = 9CH

复位值 = 0000 0000B

不可位寻址

	GPIEN7	GPIEN6	GPIEN5	GPIEN4	GPIEN3	GPIEN2	GPIEN1	GPIEN0
Bit	7	6	5	4	3	2	1	0

GPIEN.x 0 = P1.x 中断无效
 1 = P1.x 中断有效

Table 16-4. GPIF - 通用中断标志位寄存器

GPIF = 9DH

复位值 = 0000 0000B

不可位寻址

	GPIF7	GPIF6	GPIF5	GPIF4	GPIF3	GPIF2	GPIF1	GPIF0
Bit	7	6	5	4	3	2	1	0

GPIF.x 0 = P1.x 中断停止
 1 = P1.x 中断激活，必须软件清除

17. 串行接口

AT89LP216 的串行接口符合通用异步收 / 发报机 (UART) 标准。UART 具有以下特点:

- * 全双工操作。
- * 8 个或 9 个数据位元
- * 帧错误检测

- * 自动地址辨认下的多处理器通信模式
- * 利用定时器 1 的波特率产生器
- * 接收缓存满载或传输完成中断

全双工模式的串口，意味着它能同时进行发送和接收。具有接收缓存，表明它能在上一个接收的字节被读出接收寄存器之前开始接收第二个字节（然而，如果第一个字节没有被读出而第二个字节已经完成了接收，那么第一个字节会丢失）。对串行端口接收和发送寄存器的存取都是通过特殊功能寄存器（Special Function Register）SBUF 实现的。写入 SBUF 即加载了发送寄存器，读 SBUF 即访问了物理上独立的接收寄存器。串行端口能够运行在下面四个模式中。

*** 模式 0:** 串行数据通过 RXD 进出。TXD 输出移位时钟。八个数据位元以最低有效位（LSB）在前进行发送 / 接收。波特率固定为震荡频率的一半。

*** 模式 1:** 10 位数据通过 TXD 进行发送，或者通过 RXD 进行接收，其中包括一个起始位（0），8 个数据位元（LSB 在前）以及一个停止位（1）。接收时，停止位进入特殊功能寄存器的 RB8。波特率根据定时器 1 而变化。

*** 模式 2:** 11 位数据通过 TXD 进行发送，或者通过 RXD 进行接收，其中包括一个起始位（0），8 个数据位元（LSB 在前），一个可编程的第 9 位数据，以及一个停止位（1）。发送时，第 9 位数据（SCON 的 TB8）被设置为“0”或“1”。比如，奇偶校验位（P，在 PSW 中）可以被移至 TB8。接收时，第 9 位数据进入特殊功能寄存器 SCON 的 RB8，而停止位被忽略。波特率可以编程设置为震荡频率的 1/16 或 1/32。

*** 模式 3:** 11 位数据通过 TXD 进行发送，或者通过 RXD 进行接收，其中包括一个起始位（0），8 个数据位元（LSB 在前），一个可编程的第 9 位数据，以及一个停止位（1）。实际上，模式 3 除了波特率之外，在其他方面与模式 2 都一致。模式 3 的波特率根据定时器 1 而变化。

这 4 种模式中，通过使用 SBUF 作为目标寄存器的任何指令开始发送。模式 0 中，以 RI=0 和 REN=1 为条件开始接收。在其他模式中，如果 REN=1，则以收到起始位开始接收。

17.1 多处理器间通信

模式 2 和模式 3 提供专用的多处理器间通信功能。在这些模式中，9 个数据位元被接收，其后是一位停止位。而第 9 位进入 RB8。紧接着是停止位。端口可进行编程，这样当停止位被接收时，只有在 RB8=1 时串行端口中断才被激活。这个特性由设置 SCON 的 SM2 位实现。

以下的例子表示了如何使用串行中断实现多处理器间通信。当主处理器必须发送一块数据给从处理器之一时，它首先发送一个地址字节，指明目标从处理器。地址字节与数据字节的不同在于第 9 位在地址字节中为“1”，而在数据字节中为“0”。SM2=1 时，没有从处理器可以被数据字节中断。而地址字节可以中断所有从处理器。每一个从处理器检查收到的字节，看它是否标志了地址。为目标地址的从处理器清空它的 SM2 位，准备接收后面的数据字节。不是目标地址的从处理器将它们的 SM2 位置位，并忽略数据字节。

在模式 0 中，SM2 位没有作用，但是在模式 1 中可以用来检查停止位的合法性。模式 1 的接收过程，如果 SM2=1，接收中断没有被激活，除非接收到有效的停止位。

Table 17-1. SCON - 串行端口控制寄存 ~

SCON Address = 98H						复位值 = 0000 0000B		
不可位寻址								
Bit	SM0/FE	SM1	SM2	REN	TB8	RB8	T1	RI
	7	6	5	4	3	2	1	0
(SMOD0 = 0/1) ⁽¹⁾								

符号	功能																									
FE	帧错误位。当不合法的停止位被检测到时，接受器将该位置 1。FE 位不能被合法帧清 0，必须软件清 0。SMOD0 必须置位才能存取 FE 位。不论 SMOD0 状态如何，FE 都能被置位。																									
SM0	串行端口模式位 0，（SMOD0 必须为 0，以访问 SM0 位）																									
SM1	串口模式位 1																									
	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th style="width: 10%;">SM0</th> <th style="width: 10%;">SM1</th> <th style="width: 10%;">Mode</th> <th style="width: 20%;">描述</th> <th style="width: 20%;">波特率⁽²⁾</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>移位寄存器</td> <td style="text-align: center;">$f_{osc}/2$</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>8-bit UART</td> <td style="text-align: center;">variable (Timer 1)</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">2</td> <td>9-bit UART</td> <td style="text-align: center;">$f_{osc}/32$ or $f_{osc}/16$</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">3</td> <td>9-bit UART</td> <td style="text-align: center;">variable (Timer 1)</td> </tr> </tbody> </table>	SM0	SM1	Mode	描述	波特率 ⁽²⁾	0	0	0	移位寄存器	$f_{osc}/2$	0	1	1	8-bit UART	variable (Timer 1)	1	0	2	9-bit UART	$f_{osc}/32$ or $f_{osc}/16$	1	1	3	9-bit UART	variable (Timer 1)
	SM0	SM1	Mode	描述	波特率 ⁽²⁾																					
	0	0	0	移位寄存器	$f_{osc}/2$																					
	0	1	1	8-bit UART	variable (Timer 1)																					
1	0	2	9-bit UART	$f_{osc}/32$ or $f_{osc}/16$																						
1	1	3	9-bit UART	variable (Timer 1)																						
SM2	启用模式 2 或 3 的自动辨识地址特性。如果 SM2=1，此时除非接收到的第 9 位数据（RB8）是 1，即表明这是一个地址，接收到的字节是指定的或广播的地址，否则不能将 RI 置位。在模式 1 下，如果 SM2=1，那么只有在收到一个有效停止位，并且接收到的字节为指定的或广播的地址时，RI 才被激活。在模式 0 下，SM2 应该为 0。																									
REN	使串行接收有效。通过软件置位使接收有效。通过软件清零使接收无效。																									
TB8	在模式 2，3 中被传输的第 9 位数据。根据设计通过软件置位或清零。																									
RB8	在模式 2，3 中被接收的第 9 位数据。模式 1 中，如果 SM2=0，RB8 就是接收到的停止位。在模式 0 中，没有使用 RB8。																									
TI	发送中断标志。任意串行发送时，在模式 0 中，通过硬件在第 8 位发送结束时置位，或者在其他模式中，在停止位开始发送时置位。必须由软件清零。																									
RI	接收终端标志。任意串行接收时（例外情况见 SM2 说明），在模式 0 中，通过硬件在第 8 位发送结束时置位，或者在其他模式中，在停止位接收时中途置位。必须有软件清零。																									

- Notes: 1. SMOD0 在 PCON.6 中
 2. f_{osc} = 晶体振荡器频率

17.2 波特率

模式 0 中，波特率固定为由下面的公式表示：

$$\text{Mode 0 Baud Rate} = \frac{\text{Oscillator Frequency}}{2}$$

模式 2 的波特率由特殊功能寄存器 PCON.7 的 SMOD1 位的值决定。如果 SMOD1=0（重设的值），波特率为振荡频率的 1/32。如果 SMOD1=1，波特率为振荡频率的 1/16，由下面的公式表示：

$$\text{Mode 2 Baud Rate} = \frac{2^{\text{SMOD1}}}{32} \times (\text{Oscillator Frequency})$$

17.2.1 使用定时器 1 产生波特率

定时器 1 溢出速率决定了模式 1 和模式 3 的波特率。当定时器 1 为波特率发生器时，波特率由定时器 1 的溢出速率和 SMOD1 的值决定，如下面的公式所示：

$$\text{Modes 1, 3 Baud Rate} = \frac{2^{\text{SMOD1}}}{32} \times (\text{Timer 1 Overflow Rate})$$

定时器 1 中断在此处应失效。在它的 3 个运行模式下，定时器本身能够配置为定时器或计数器操作。在最典型的应用中，它被配置为自动重载模式（TMOD 的高 4 位 =0010B）的定时器操作。在这种情况下，波特率由下面的公式指定：

$$\text{Modes 1, 3 Baud Rate} = \frac{2^{\text{SMOD1}}}{32} \times \frac{\text{Oscillator Frequency}}{[256 - (\text{TH1})]} \times \frac{1}{\text{TPS} + 1}$$

程序员可以通过设置定时器运行于 16 位自动重载状态（TMOD 的高 4 位 =0001B），获得非常低的波特率。这种情况下，波特率由下面的公式指定：

$$\text{Modes 1, 3 Baud Rate} = \frac{2^{\text{SMOD1}}}{32} \times \frac{\text{Oscillator Frequency}}{[256 - (\text{RH1}, \text{RL1})]} \times \frac{1}{\text{TPS} + 1}$$

表 17-2 列出通常用到的波特率以及这些波特率如何由定时器 1 得到的。

Table 17-2. 定时器 1 通常波特率 (TPS = 0000B)

波特率	f _{osc} (MHz)	SMOD1	定时器 1		
			C/T	模式	重装值
Mode 0: 1 MHz	2	X	X	X	X
Mode 2: 375K	12	0	X	X	X
62.5K	12	1	0	2	F4H
19.2K	11.059	1	0	2	DCH
9.6K	11.059	0	0	2	DCH
4.8K	11.059	0	0	2	B8H

Table 17-2. 定时器 1 通常波特率 (TPS = 0000B)

波特率	f _{osc} (MHz)	SMOD1	定时器 1		
			C/T	模式	重装值
2.4K	11.059	0	0	2	70H
1.2K	11.059	0	0	1	FEE0H
137.5	11.986	0	0	1	F55CH
110	6	0	0	1	F958H
110	12	0	0	1	F304H

17.3 模式 0 介绍

串行的数据通过 RXD 进出。TXD 输出移位时钟。8 个数据位元，以最低有效位在前进行发送和接收。波特率固定为振荡频率的 1/2。42 页上的图 17-1 显示了模式 0 的串行端口简化功能图和相关的时序。

通过使用 SBUF 作为目标寄存器的任何指令开始发送。同时信号“写入 SBUF”将“1”载入发送移位寄存器的第 9 位，告诉 TX 控制区开始发送。内部定时使一个全工作周期在“写 SBUF”和激活发送 SEND 之间开始并结束。

SEND 将移位寄存器的输出传输至 P3.0 的选择输出功能线，也将移位时钟传输至 P3.1 的选择输出功能线。在移位时钟的下降沿，发送移位寄存器的内容向右移一位。

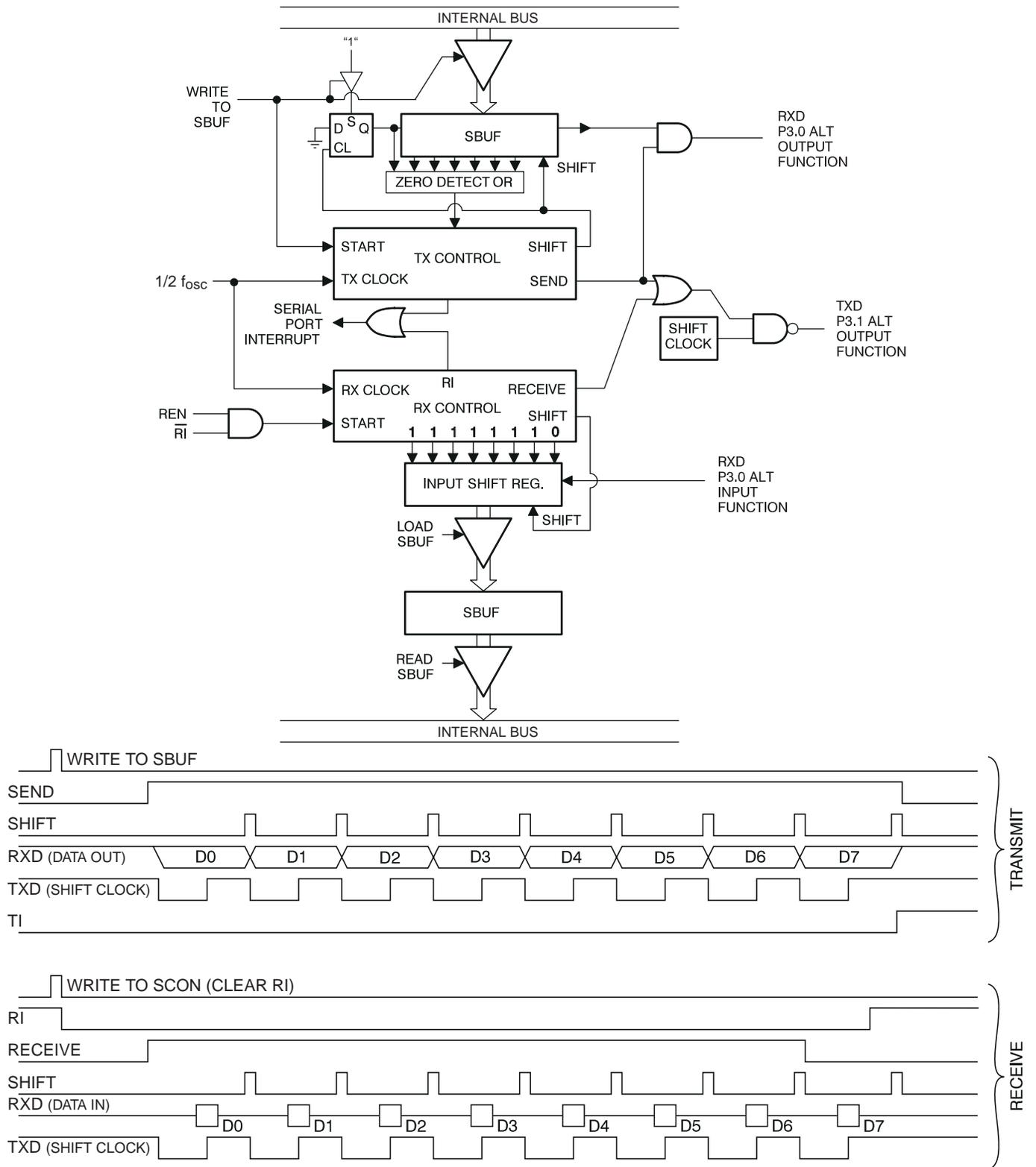
数据右移一位后，最左边填充“0”。当数据字节的最高有效位位于移位寄存器的输出位置时，最初载入第 9 位的“1”位于最高有效位的最左边，并且所有在它左边的位都为“0”。这种情况标志 TX 控制区做最后一次移位，然后接触发送的激活，并将 TI 置位。

当 REN=0, R1=0 时，开始接收过程。在下一个时钟周期，RX 控制单元将位“11111110”写入接收移位寄存器，并在下一个时钟阶段激活接收。

接收 RECEIVE 使移位时钟对 P3.1 的选择输入功能线有效。在移位时钟的下降沿，接收移位寄存器的内容向左移一位。从右边进入的值就是 P3.0 管脚在移位时钟的上升沿进行采样的值。

数据位从右进入时，“1”从最左边移出。当开始载入最右边位置的“0”到达移位寄存器的最左边时，标志着 RX 控制区做最后一次移位，并且载入 SBUF。接下来，将 RECEIVE 清零，将 RI 置位。

Figure 17-1. 串行模式 0



17.4 模式 1 介绍

通过 TXD 进行 10 位数据的发送，通过 RXD 进行 10 位数据的接收。10 位数据包括：1 个起始位（0），8 个数据位元（LSB 在前），以及 1 个停止位（1）。接收时，停止位进入 SCON 的 RB8。AT89LP216 中，波特率由定时器 1 的溢出速率决定。图 17-2 为模式 1 的简化功能图以及发送接收的相关时序。

通过使用 SBUF 作为目标寄存器的任何指令开始发送。信号“写入 SBUF”将一个“1”载入发送移位寄存器的第 9 位，标志 TX 控制单元有一个发送的请求。发送实际开始于紧接着 16 分计数器的下一次翻转机器周期的 S1P1。因此，位的时间是与 16 分计数器同步，而不是与“写入 SBUF”信号同步。

当 SEND 被激活时，开始发送，向 TXD 放入开始位。一个数据位的时间后，数据 DATA 激活，使发送移位寄存器的输出的位到 TXD。之后产生第一个移位脉冲。

随着数据位向右移出，从左边进入的“0”被计时。当数据字的最高有效位位于移位寄存器的输出位置时，最初载入第 9 位的“1”刚好在最高有效位的左边，在它左边的所有位都为“0”。这种情况标志 TX 控制单元将做最后一次移位，然后解除 SEND 的激活并将 TI 置位。这些发生在“写入 SBUF”后 16 分计数器第 10 次翻转时。

RXD 检测到由“1”到“0”的跃迁后，开始接收过程。为了达到这个目的，RXD 以指定波特率的 16 倍的速率进行采样。检测到跃迁后，16 分计数器马上复位，“1FFH”写入输入移位寄存器中。复位 16 分计数器使它的翻转与接收到的位元的次数一致。

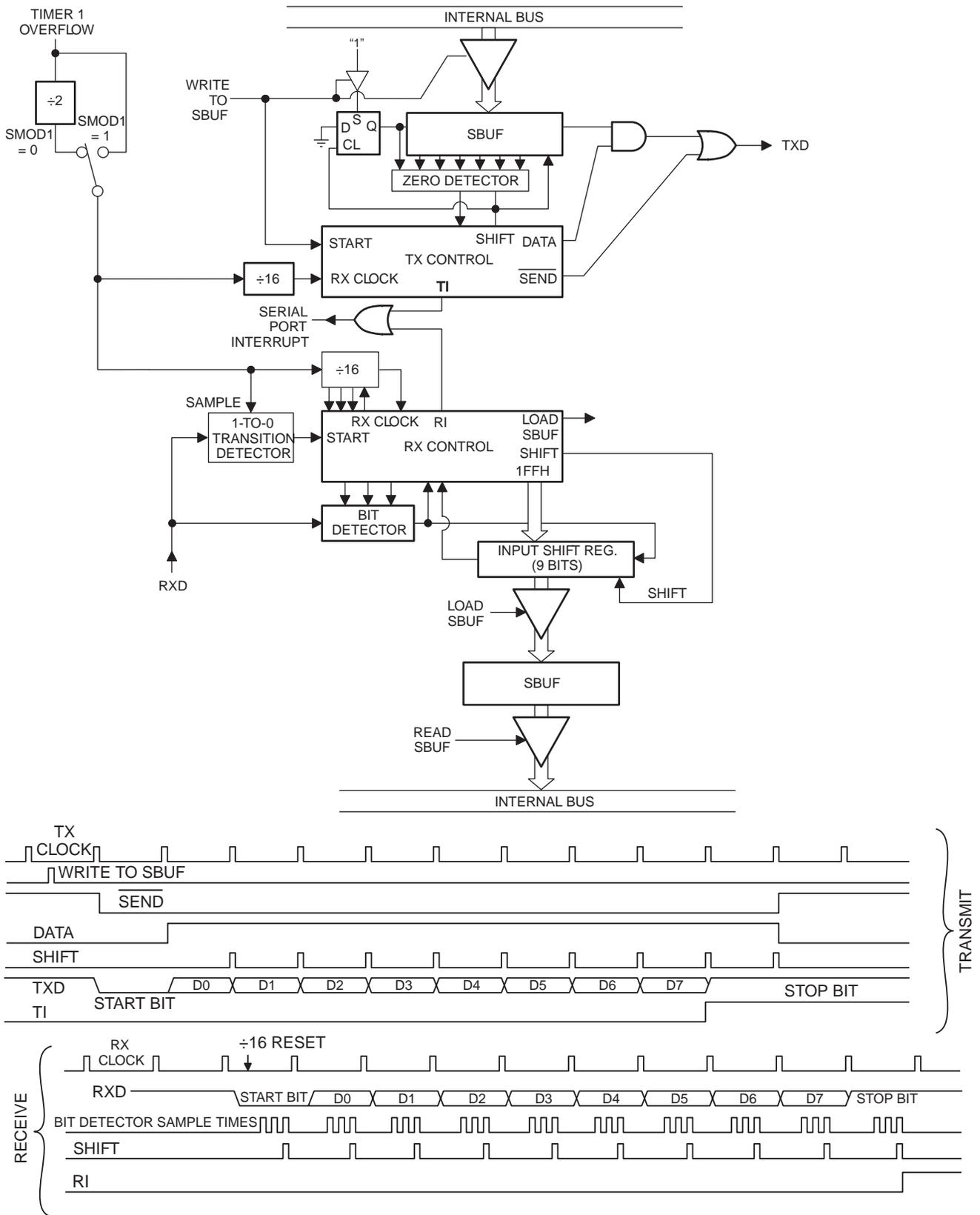
计数器的 16 种状态将每一个位元的时间 16 等分。在每个位元时间的第 7, 8, 9 个计数器状态，位元检测器对 RXD 的值进行采样。公认的值至少存在于 2/3 的采样值中。这样做是为了避免噪音。为了避免错误的位元，如果第一个位元时间接受的值不是 0，接收电路重启，并且该单元继续寻找另一个“1”到“0”的跃迁。如果起始位有效，就将它移至输入移位寄存器，继续接受该帧剩余部分。

随着数据位由右侧进入，“1”从左边移出。当起始位到达移位寄存器（即模式 1 中的 9 位寄存器）最左边位置时，标志着 RX 控制区将做最后一次移位，载入 SBUF 和 SB8，并重启 RI。只有在最后一个移位脉冲产生时，在下列条件都满足的情况下，才会产生载入 SBUF 和 RB8，将 RI 置位的信号。

1. $RI = 0$
2. $SM2 = 0$, 或接收到的停止位 = 1

如果上述两个条件之一不满足，接收到的帧会不可挽回的丢弃。如果两者都满足了，停止位进入 RB8，8 个数据位元进入 SBUF，激活 RI。同时，不管上述条件是否满足，该单元继续寻找 RXD 中“1”到“0”的跃迁。

Figure 17-2. 串行模式 1



17.5 模式 2, 3 介绍

通过 TXD 进行 11 位数据的发送，通过 RXD 进行 11 位数据的接收。11 位数据包括：1 个起始位（0），8 个数据位元（LSB 在前），一个可编程的第 9 位数据位元，以及 1 个停止位（1）。发送时，第 9 位数据位元（TB8）可以被设置为“0”或“1”。接收时，第 9 位数据位元进入 SCON 的 RB8。模式 2 中，波特率可以编程设置为振荡频率的 1/16 或 1/32。模式 3 中，波特率可变，由定时器 1 决定。

图 17-3 和 17-4 显示了模式 2, 3 的串行端口功能图。接收部分与模式 1 完全相同，发送部分与模式 1 的差别只在于发送移位寄存器的第 9 个数据位元。

通过使用 SBUF 作为目标寄存器的任何指令开始发送。信号“写入 SBUF”同样将 TB8 载入发送移位寄存器的第 9 位，标志 TX 控制单元有一个发送的请求。发送实际开始于紧接着 16 分计数器的下一次翻转机器周期的 S1P1。因此，位的时间是与 16 分计数器同步，而不是与“写入 SBUF”信号同步。

当 SEND 被激活时，开始发送，向 TXD 放入起始位。一个数据位的时间后，数据 DATA 激活，使发送移位寄存器的输出的位到 TXD。之后产生第一个移位脉冲。第一次移位记录“1”（停止位）进入移位计时器第 9 位的时间。然后唯一的“0”被计时。因此随着数据位向右移出，从左边进入的“0”被计时。当 TB8 位于移位寄存器的输出位置时，停止位刚好在 TB8 的左边，在停止位的左边都为“0”。这种情况标志 TX 控制单元将做最后一次移位，然后解除 SEND 的激活并将 TI 置位。这些发生在“写入 SBUF”后 16 分计数器第 11 次翻转时。

RXD 检测到由“1”到“0”的跃迁后，开始接收过程。为了达到这个目的，RXD 以指定波特率的 16 倍的速率进行采样。检测到跃迁后，16 分计数器马上复位，“1FFH”写入输入移位寄存器中。

在每个位元时间的第 7, 8, 9 个计数器状态，位元检测器对 RXD 的值进行采样。公认的值至少存在于 2/3 的采样值中。如果第一个位元时间接受的值不是 0，接收电路重启，并且该单元继续寻找另一个“1”到“0”的跃迁。如果起始位有效，就将它移至输入移位寄存器，继续接受该帧剩余部分。

随着数据位由右侧进入，“1”从左边移出。当起始位到达移位寄存器（即模式 2, 3 中的 9 位寄存器）最左边位置时，标志着 RX 控制区将做最后一次移位，载入 SBUF 和 SB8，并重启 RI。只有在最后一个移位脉冲产生时，在下列条件都满足的情况下，才会产生载入 SBUF 和 RB8，将 RI 置位的信号：

1. $RI = 0$
2. $SM2 = 0$ ，或接收到的第 9 个数据位元 = 1

如果上述两个条件之一不满足，接收到的帧会不可挽回的丢弃，并且 RI 不被置位。如果两者都满足了，接收到的第 9 个数据位元进入 RB8，而前 8 个数据位元进入 SBUF。一个位元时间后，不管上述条件是否满足，该单元继续寻找 RXD 输入中“1”到“0”的跃迁。

注意：接收到的停止位的值与 SBUF, RB8 或 RI 均无关。

Figure 17-3. 串行模式 2

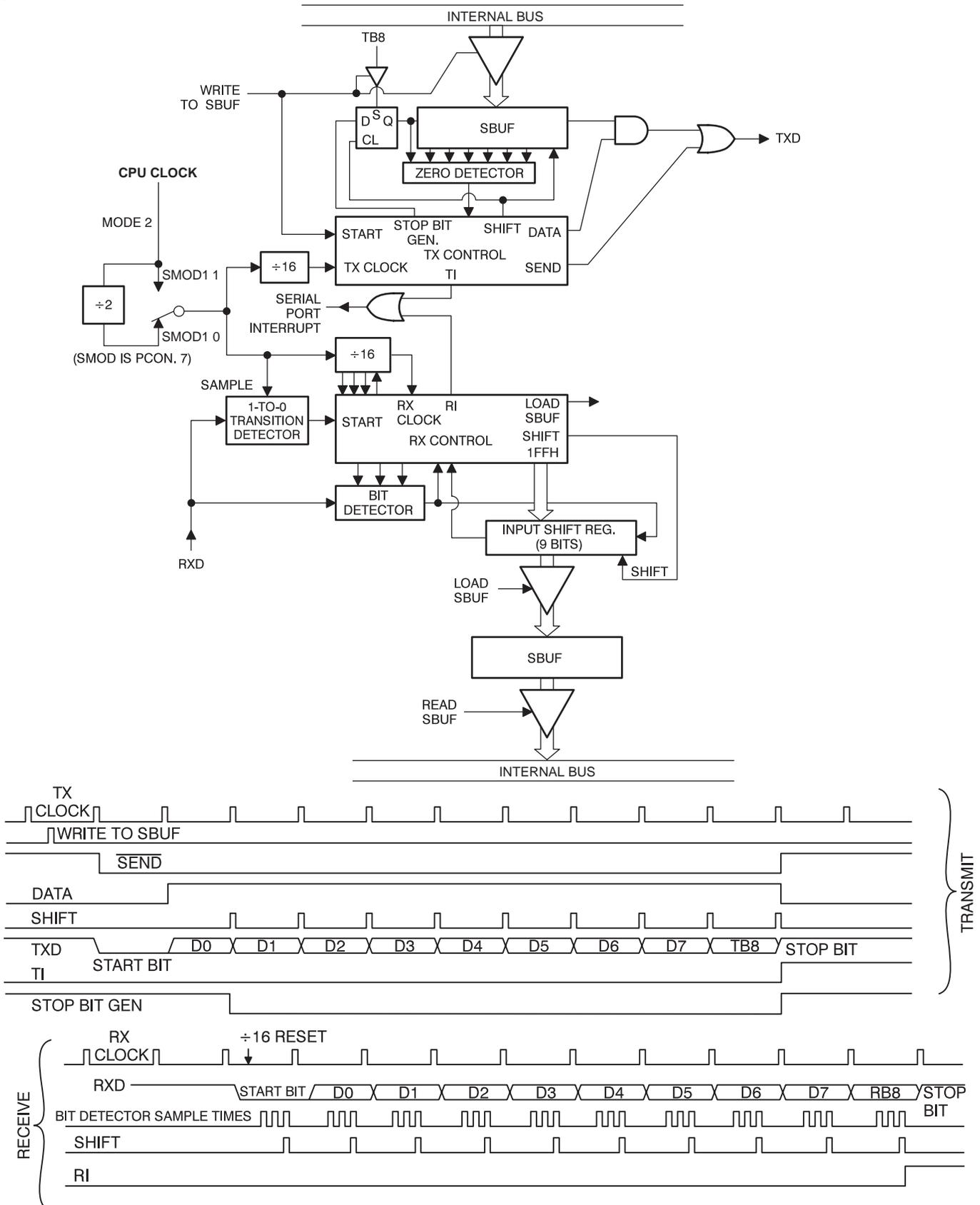
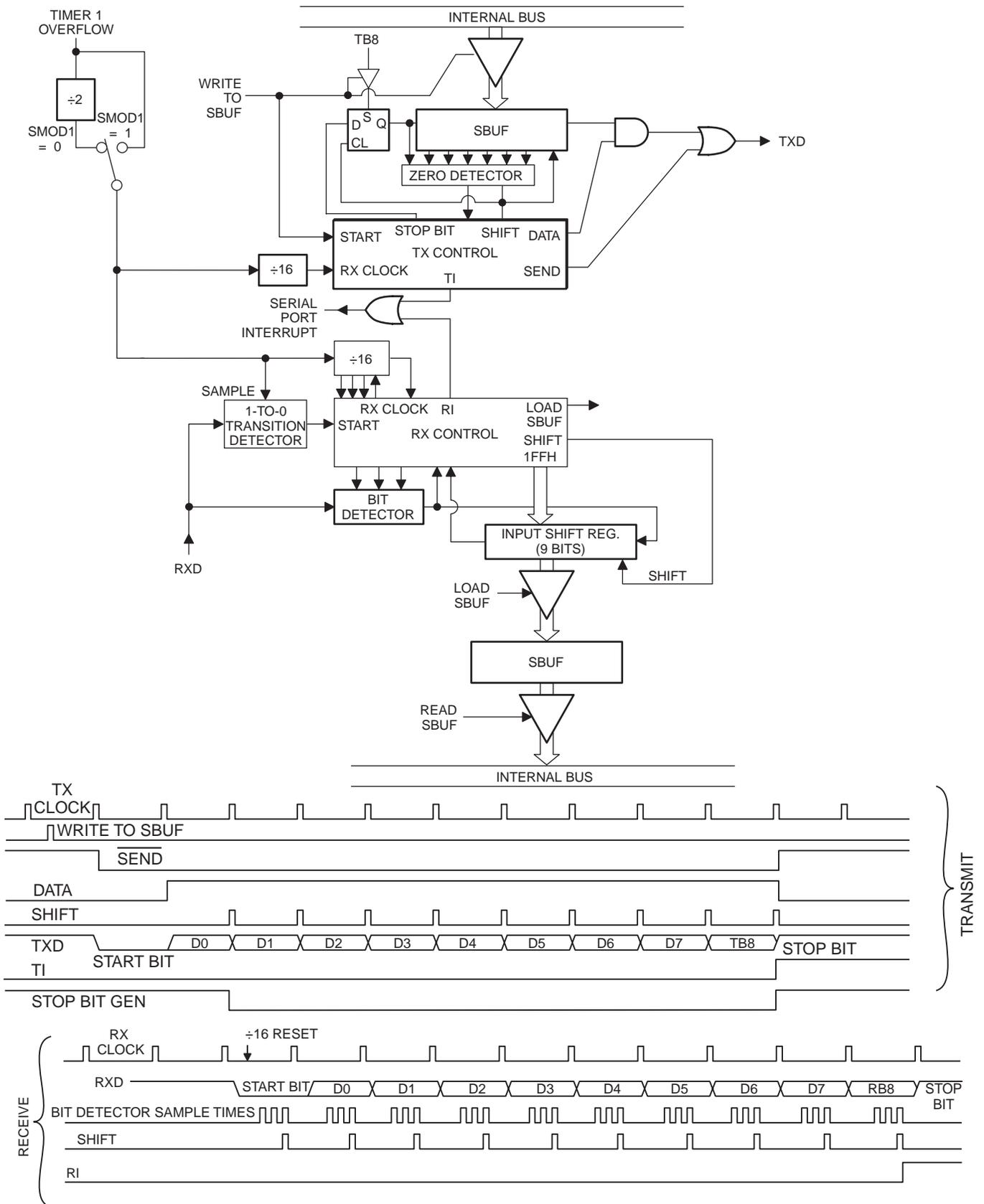


Figure 17-4. 串行模式 3



17.6 帧错误检测

除了所有通常模式外，UART 能提供通过查找丢失的停止位进行帧错误检测，以及自动地址辨认。当用于帧错误检测时，UART 在通信中查找丢失的停止位。丢失的位将使 SCON 寄存器中的 FE 位置位。FE 位与 SM0 共享 SCON.7 位，SCON.7 的功能是由 PCON.6 (SMOD0) 决定的。如果 SMOD0 置位，那么 SCON.7 的功能是 FE。当 SMOD0 清零时，SCON.7 的功能是 SM0。当用作 FE 时，SCON.7 只能由软件清零。

17.7 自动地址识别

自动地址识别具有一种特性，即利用硬件作出的比较使 UART 可以从大量连续字位中识别某些特定地址。这种特性减少了依赖大量软件来检测通过连续端口的每一个连续地址。把 SM2 字位设置在 SCON 就可激活这种特性。在这种第 9 字位 UART 模式中，当接收到包含“已知”地址或“广播”地址的字节时，模式 2 模式 3 和接受中断信号标识将自动激活。这种第 9 字位模式为了表明所接收到的信息是地址而非数据，则要求第 9 个信息字位必须为“1”。

第 8 字位模式也叫模式 1。在这种模式中，如果 SM2 被激活并且接受的信息在第 8 字位后有终端中止的字位，不管是已知地址还是广播地址，接受中断信号标识将会设置为激活。

模式 0 是个备份寄存器模式，同时 SM2 就无任何作用。

利用自动地址识别的特性调用已知分端口的地址或地址群，就可以使主端口有选择性地连接一个或多个分端口。所有的分端口都可以用“广播”地址来连接起来。两种特殊功能的寄存器被用来定义分端口地址 SADDR，和地址掩模 SADEN。SADEN 是用来定义在 SADDR 中哪些字位将被用到，哪些则无任何意义。逻辑上，SADEN 掩模和 SADDR 与门给出已知地址，主端口用已知地址来分配每个分端口的地址。用已知地址可以识别多个分端口并且不和其他的混淆。下面的例子来说明这种配置具有多功能性：

```
分端口 0:           SADDR = 1100 0000
                    SADEN = 1111 1101
                    Given  = 1100 00X0
```

```
分端口 1           SADDR = 1100 0000
                    SADEN = 1111 1110
                    Given  = 1100 000X
```

在上面的例子中，SADDR 是同样的，SADEN 的数据用来区分两个分端口。分端口 0 需要一个“0”在 0 字位而不用考虑 1 字位，分端口 1 需要一个“0”在 1 字位而不用考虑 0 字位。对分端口 0 来说只有唯一的地址 1100 0010，因为分端口 1 在 1 字位是“0”。对分端口 1 来说也只有唯一的地址 1100 0001，因为“1”在 0 字位就不和分端口 0 重复了。用同一个地址即 0 字位 =0（分端口 0）1 字位 =0（分端口 1），两个分端口就可以被同时选择了。因此，两个分端口的地址都可以表示成 1100 0000。

在一台更加复杂的系统中，用下面的方法就可以选择分端口 1 和 2，而同时不包含分端口 0：

```
分端口 0           SADDR = 1100 0000
                    SADEN = 1111 1001
                    Given  = 1100 00X0
```

分端口 1 SADDR = 1110 0000
 SADEN = 1111 1010
 Given = 1110 0X0X

分端口 2 SADDR = 1110 0000
 SADEN = 1111 1100
 Given = 1110 00XX

在上面例子中，三个端口之间的区别在于低端 3 位地址位的不同。端口 0 需要 0 字位 =0，就有唯一的地址 1110 0110。端口 1 需要 1 字位 =0，就有唯一的地址 1110 0101。端口 2 需要 2 字位 =0，就有唯一的地址 1110 0011。要选择端口 0 和 1，就得用地址 1110 0100，因为必须使 2 字位 =1 来排除端口 2。

每一个分端口的广播地址是取 SADDR 和 SADEN 的逻辑 OR 获得的。在这个结果中所有的 0 被忽略。大多数情况下，如果视忽略值为 1，广播地址则为 16 进制的 FF。

重置 SADDR (SFR 地址 0A9H) 和 SADEN (SFR 地址 0B9H) 后，所有地址位归置为“0”。这就产生了包含所有无意义字位的已知地址和包含所有无意义字位的广播地址。并有效地禁止自动寻址模式，使微控制器使用不执行这个特性的标准 80C51 型 UART 驱动器。

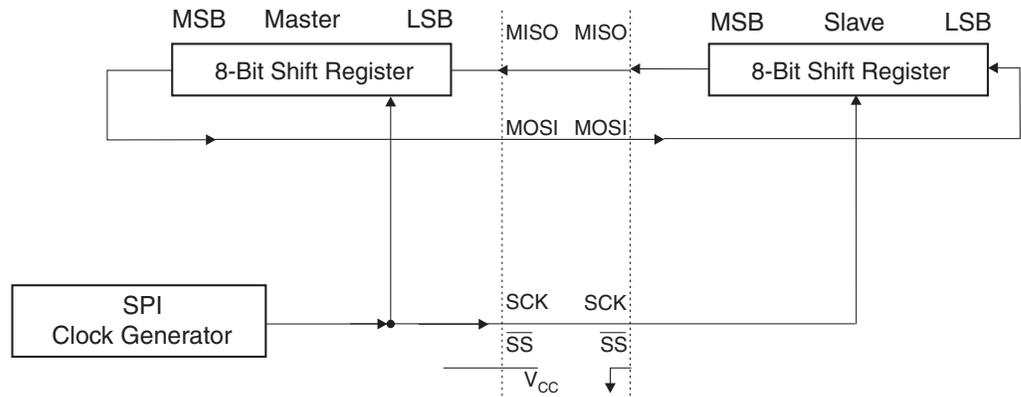
18. 串行外设接口

串行外设接口可使高速同步数据在 AT89LP216 和外设元件或者多个 AT89LP216 元件之间传输。串行外设接口特性如下：

- * 双向，3 线同步数据传输
- * 主、从操作方式
- * 最大字位频率 = fosc/4
- * LSB 或 MSB 优先数据传输
- * 4 级可编程地址位的从模式
- * 传输结束中断标识
- * 存写冲突标识保护
- * 双缓冲接受
- * 双缓冲传输（高级模式才有）
- * 闲置模式唤醒

主、从 CPU 用 SPI 连接的方式如图 18-1 所示。接口的四个引脚分别是主进从出 (MISO)，主出从进 (MOSI)，移位时钟 (SCK) 和从属选择 ()。SCK 引脚在主模式中是时钟输出，而在从模式中是时钟输入。在 SPCR 中 MSTR 位决定了 MISO 和 MOSI 的方向。要注意到 MOSI 连接 MOSI，MISO 连接 MISO。在主模式中， /P1.4 被忽略而只有可能作为一般的输入输出。在从模式中，必须置于低电平来选择单个元件作为从属。当置于高电平时，从属的 SPI 端口无效并且 MOSI/P1.5 能用于一般输出。

Figure 18-1. 主从连接



SPI 有两种操作模式：标准模式（无缓冲存写）和高级模式（缓冲存写）。对主 CPU 的 SPI 数据寄存器的存写触发 SPI 时钟寄存器，然后存写数据从 MOSI 引脚输出再输入到从 CPU 的 MISO 的引脚。当时钟寄存器等待下一个指定波长的满位槽时，传输经过一段初始延时就可触发。转移一个字节后，SPI 时钟寄存器停止，同时设置传输中断位并把接受的字节传输到可读缓冲器中。如果要同时设置 SPI 中断激活位和串口中断激活位，就需要中断（传输）。需要指出的是无论 SPDR 用可写数据寄存器或可读数据寄存器，都取决于访问路径是可读还是可写。在标准模式下，由于可写缓冲器是透明的（并且对 SPDR 的存写路径会访问移位寄存器），所以在进行传输时任何向 SPDR 的存写都将导致和 WCOL 设置的存写冲突。但传输仍将正常完成，不过新字节是无效的所以必须再重新写到 SPDR。

除可写缓冲器能控制传输下一字节外，高级模式类似于标准模式。SPDR 的存写会占用可写缓冲器，并告诉 WCOL 缓冲器已经满荷，以后任何的存写都将覆盖掉缓冲器。当缓冲字节加载到移位寄存器开始传输时，硬件将会清除 WCOL。如果主 SPI 处于闲置状态，比如当前是第一个字节，在加载 SPDR 后，字节传输就会开始并且 WCOL 会被立即清除。在当前字节传输时，下一个字节也有可能被写入到 SPDR。SPDR 中的加载激活标识（LDEN）用来决定传输开始的时间。LDEN 取决于 SPI 传输的前四位字位。每加载下一字节前，主 CPU 应当首先检测 LDEN 的设置和 WCOL 是否被清除。在高级模式下，如果在传输完成后 WCOL 是设置的，比如若可以进行下一字节传输，SPI 会即时把缓存字节加载到移位寄存器，重设 WCOL，进行不中断传输同时重新开始时钟发生器。只要 CPU 能保持最大利用可写缓存，大量的字节就可以在最短时间间隔内传输出去。

Table 18-1. SPCR – SPI 控制寄存器

SPCR Address = E9H		复位值 = 0000 000B						
不可位寻址								
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
Bit	7	6	5	4	3	2	1	0

符号	功能
SPIE	SPI interrupt enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.
SPE	SPI enable. SPI = 1 enables the SPI channel and connects \overline{SS} , MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.
DORD	Data order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.
MSTR	Master/slave select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects slave SPI mode.

符号	功能															
CPOL	Clock polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI clock phase and polarity control.															
CPHA	Clock phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI clock phase and polarity control.															
SPR0 SPR1	<p>SPI clock rate select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, f_{OSC}, is as follows:</p> <table style="margin-left: 20px; border-collapse: collapse;"> <thead> <tr> <th style="border-bottom: 1px solid black;">SPR1</th> <th style="border-bottom: 1px solid black;">SPR0</th> <th style="border-bottom: 1px solid black;">SCK</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>$f_{OSC}/4$</td> </tr> <tr> <td>0</td> <td>1</td> <td>$f_{OSC}/8$</td> </tr> <tr> <td>1</td> <td>0</td> <td>$f_{OSC}/32$</td> </tr> <tr> <td>1</td> <td>1</td> <td>$f_{OSC}/64$</td> </tr> </tbody> </table>	SPR1	SPR0	SCK	0	0	$f_{OSC}/4$	0	1	$f_{OSC}/8$	1	0	$f_{OSC}/32$	1	1	$f_{OSC}/64$
SPR1	SPR0	SCK														
0	0	$f_{OSC}/4$														
0	1	$f_{OSC}/8$														
1	0	$f_{OSC}/32$														
1	1	$f_{OSC}/64$														

- Notes:
1. 激活 SPI 前设置成时钟模式；除 SPE 字位外，设置好所有 SPCR 需要的字位，然后再设置 SPE。
 2. 激活主 SPI 优先于从元件。
 3. 无新加载数据时，主从对应在下个 Tx。

Table 18-2. SPDR – SPI D 数据寄存器

SPDR Address = EAH								复位值 = 00H (after cold reset)
不可位寻址								unchanged (after warm reset)
	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
Bit	7	6	5	4	3	2	1	0

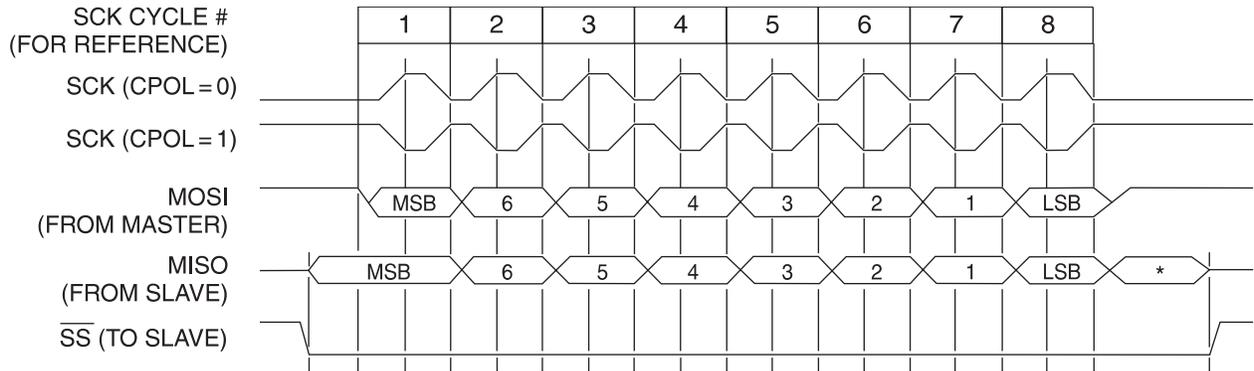
Table 18-3. SPSR – SPI 状态寄存器

SPSR Address = E8H								复位值 = 000X X000B
不可位寻址								
	SPIF	WCOL	LDEN	–	–	SSIG	DISSO	ENH
Bit	7	6	5	4	3	2	1	0

符号	功能
SPIF	SP interrupt flag. When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register followed by reading/writing the SPI data register.
WCOL	When ENH = 0: Write collision flag. The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register followed by reading/writing the SPI data register. When ENH = 1: WCOL works in Enhanced mode as Tx Buffer Full. Writing during WCOL = 1 in enhanced mode will overwrite the waiting data already present in the Tx Buffer. In this mode, WCOL is no longer reset by the SPIF reset but is reset when the write buffer has been unloaded into the serial shift register.
LDEN	Load enable for the Tx buffer in enhanced SPI mode. When ENH is set, it is safe to load the Tx Buffer while LDEN = 1 and WCOL = 0. LDEN is high during bits 0 - 3 and is low during bits 4 - 7 of the SPI serial byte transmission time frame.
SSIG	Slave Select Ignore. If SSIG = 0, the SPI will only operate in slave mode if \overline{SS} (P1.4) is pulled low. When SSIG = 1, the SPI ignores \overline{SS} in slave mode and is active whenever SPE (SPCR.6) is set. P1.4 may be used as a regular I/O pin when SSIG = 1.
DISSO	Disable slave output bit. When set, this bit causes the MISO pin to be tri-stated so more than one slave device can share the same interface with a single master. Normally, the first byte in a transmission could be the slave address and only the selected slave should clear its DISSO bit.
ENH	Enhanced SPI mode select bit. When ENH = 0, SPI is in normal mode, i.e. without write double buffering. When ENH = 1, SPI is in enhanced mode with write double buffering. The Tx buffer shares the same address with the SPDR register.

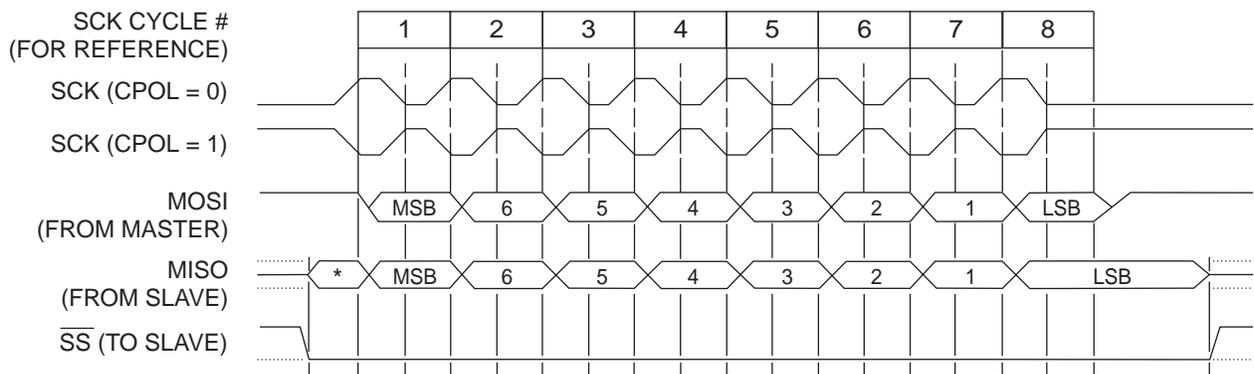
CPHA（时钟状态），CPOL（时钟特性）和 SPR（串行外设时钟比率 = 波特比率）在 SPCR 中控制 SCK 比率和形态。当 SPI 在主模式时，两个 SPR 字位提供四种可能的时钟比率。在从模式下，只要输入来的 SCK 比率不超过最大位比率，SPI 在此比率下运作。根据串行数据的不同，SCK 的相位和特性有四种可能的组合。SPI 数据传输形式表示在图 18-4 和 18-5 中。为避免接口的中断干扰 SCK，CPHA，CPOL 和 SPR 不应该在接口激活时被修改，并且主元件应该先于从元件激活。

Figure 18-4. 在 CPHA=0 时 SPI 传输形式



Note: * 无特殊定义时，MSB 一般只接受数据。

Figure 18-5. * 在 CPHA=1 时 SPI 传输形式



Note: 无特殊定义时，LSB 为前一个传输特性。

19. 模拟比较器

在 AT89LP216 上带有模拟比较器。模拟比较器有下列特性：

- * 比较输出标识和中断
- * 可选择中断条件

- 高电平或低电平
- 上升沿或下降沿
- 输出锁定

* 硬件反跳模式

比较器工作原理是当正输入 AIN0 (P1.0) 大于负输入 AIN0 (P1.1) 时, 输出为逻辑 “1”; 否则输出为逻辑 “0”。在 ACSR 中设置 CEN 字位就可激活比较器。当比较器首先被激活时, 比较器输出和中断标识无法保证能稳定 10us; 此时不能激活相应的比较器中断, 并且在为避免突然中断而触发中断前, 比较器中断标识必须清空。在激活比较器前, 应当把 P1.0 和 P1.1 设置只可输入模式使模拟输入处于三态。第 25 页上图示 “端口 1 模拟功能”

在 ACSR 中设置 CM 位会引起输出值得变化, 这种配置可能导致比较器的中断。只要比较器输出值和 CM 的特定条件相匹配, ACSR 中的比较器中断标识就会被设置。此标识可能会被软件消除也可能产生一个中断然后被软件清除。

19.1 比较器反跳中断

比较器输出样本是每一个时钟周期循环。模拟输入的条件比较器输出极端锁死。尤其是运用速度很慢的模拟输入时更是如此。有三种反跳模式来过滤掉这种干扰杂音。反跳模式中, 比较器用定时器 1 来调整样本时间。当一个相关的传输发生时, 比较器要等到取样输出前的两个计时器溢满时。如果新样本接受期望值, CF 就设置; 否则, 这个过程就无效。通过调节定时器 1 的暂停周期就可调谐过滤器。由于定时器 1 是空转的, 反跳器必须等待两次时间溢满才能保证样本延时至少是一个暂停周期。因此在初始过程后, 中断可发生在第 1 次和第 2 次暂停之间。看图 19-1

闲置模式下比较器是默认在非激活状态下。为使比较器在闲置状态时也能工作, 必须设置 ACSR 中的 CIDL。在 CIDL 设置后, 如果比较器中断在激活状态, 比较器就可唤醒 CPU 从闲置状态到工作状态。关电模式下比较器一直处在非激活状态。

Figure 19-1. 反跳模式中的负边缘

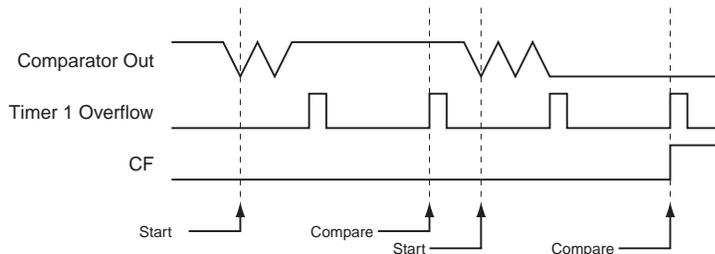


Table 19-1. ACSR- 模拟比较器控制及状态寄存器

ACSR = 97H		复位值 = XXX0 0000B						
不可位寻址								
	-	-	CIDL	CF	CEN	CM3	CM1	CM0
Bit	7	6	5	4	3	2	1	0

符号	功能																																				
CIDL	Comparator Idle Enable. If CIDL = 1 the comparator will continue to operate during Idle mode. If CIDL = 0 the comparator is powered down during Idle mode. The comparator is always shut down during Power-down mode.																																				
CF	Comparator Interrupt Flag. Set when the comparator output meets the conditions specified by the CM [2:0] bits and CEN is set. The flag must be cleared by software. The interrupt may be enabled/disabled by setting/clearing bit 6 of IE.																																				
CEN	Comparator Enable. Set this bit to enable the comparator. Clearing this bit will force the comparator output low and prevent further events from setting CF. When CEN = 1 the analog input pins, P1.0 and P1.1, have their digital inputs disabled.																																				
CM [2:0]	<p>Comparator Interrupt Mode</p> <table border="1"> <thead> <tr> <th>CM2</th> <th>CM1</th> <th>CM0</th> <th>Interrupt Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Negative (Low) level</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Positive edge</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Toggle with debouncing⁽¹⁾</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Positive edge with debouncing⁽¹⁾</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Negative edge</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Toggle</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Negative edge with debouncing⁽¹⁾</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Positive (High) level</td> </tr> </tbody> </table>	CM2	CM1	CM0	Interrupt Mode	0	0	0	Negative (Low) level	0	0	1	Positive edge	0	1	0	Toggle with debouncing ⁽¹⁾	0	1	1	Positive edge with debouncing ⁽¹⁾	1	0	0	Negative edge	1	0	1	Toggle	1	1	0	Negative edge with debouncing ⁽¹⁾	1	1	1	Positive (High) level
CM2	CM1	CM0	Interrupt Mode																																		
0	0	0	Negative (Low) level																																		
0	0	1	Positive edge																																		
0	1	0	Toggle with debouncing ⁽¹⁾																																		
0	1	1	Positive edge with debouncing ⁽¹⁾																																		
1	0	0	Negative edge																																		
1	0	1	Toggle																																		
1	1	0	Negative edge with debouncing ⁽¹⁾																																		
1	1	1	Positive (High) level																																		

Note: 1. Debouncing modes require the use of Timer 1 to generate the sampling delay.

20. 可编程看门狗定时器

在计时器溢满前软件无法支持计时器后，系统会超时并触发系统重启，可编程监控计时器（WDT）就用来防止此类误操作。WDT 计算着 CPU 的时钟循环。一些高字位，PS0, PS1, PS2 在 SFR WDTCN 中用来把监控计时器的时钟周期从 16K 设到 2048K。计时器溢出还可以延长暂停时间（看第 12 页上的表 9-2）。在重启和关电模式下，WDT 是非激活状态。当 WDT 超时延时后，会产生一个内部的 RST 脉冲信号来重启 CPU。WDT 可选择周期见表 20-1。

Table 20-1. Watchdog Timer Time-out Period Selection

WDT Prescaler Bits			Period ⁽¹⁾ (Clock Cycles)
PS2	PS1	PS0	
0	0	0	16K
0	0	1	32K
0	1	0	64K
0	1	1	128K
1	0	0	256K
1	0	1	512K
1	1	0	1024K
1	1	1	2048K

Note: 1. WDT 暂停周期取决于系统时钟频率。

$$\text{Time-out Period} = \frac{2^{(PS+14)}}{\text{Oscillator Frequency}} \times (TPS+1)$$

监控计时器包括一个含 7 位可编程增益的 14 位计时器。把程序 1EH/E1H 写到 WDTRST 寄存器会激活计时器。当 WDT 激活时，WDTCN 里的 WDTEN 字位就会变成“1”。为防止 WDT 在溢出时会重启，必须在暂停周期结束前把监控程序写到 WDTRST。在两次写入指令中间，SFR 可读不可写。指令会把 1EH 移到 WDTRST 寄存器然后 1EH 就会过去。错误的程序会导致监控器的突然重启。可编程程序激活监控计时器的程序如下：

```
MOV WDTRST, #01Eh
MOV WDTRST, #0E1h
```

20.1 软件复位

把软件重启程序 5AH/A5H 写到 WDTRST SFR 就可完成 AT89LP216 的软件重启。不用激活 WDT 重启软件。一次正常的软件重启必须在 WDTCN 中设置 SWRST。但是，如果任何时候有错误程序写到 WDTRST，软件将立即重启并且 SWRST 和 WDTOVF 标识也将设置。在这类情形下，正常的软件重启就要比程序出错引起软件重启好得多。使软件重启的可编程程序如下：

```
MOV WDTRST, #05Ah
MOV WDTRST, #0A5h
```

Table 20-2. WDTCON – 看门狗控制寄存器

WDTCON Address = A7H					复位值 = 0000 X000B			
不可位寻址								
	PS2	PS1	PS0	WDIDLE	–	SWRST	WDTOVF	WDTEN
Bit	7	6	5	4	3	2	1	0
符号	功能							
PS2 PS1 PS0	Prescaler bits for the watchdog timer (WDT). When all three bits are cleared to 0, the watchdog timer has a nominal period of 16K clock cycles. When all three bits are set to 1, the nominal period is 2048K clock cycles.							
WDIDLE	Disable/enable the Watchdog Timer in IDLE mode. When WDIDLE = 0, WDT continues to count in IDLE mode. When WDIDLE = 1, WDT freezes while the device is in IDLE mode.							
SWRST	Software Reset Flag. Set when a software reset is generated by writing the sequence 5AH/A5H to WDTRST. Also set when an incorrect sequence is written to WDTRST. Must be cleared by software.							
WDTOVF	Watchdog Overflow Flag. Set when a WDT rest is generated by the WDT timer overflow. Also set when an incorrect sequence is written to WDTRST. Must be cleared by software.							
WDTEN	Watchdog Enable Flag. This bit is READ-ONLY and reflects the status of the WDT (whether it is running or not). The WDT is disabled after any reset and must be re-enabled by writing 1EH/E1H to WDTRST							

Table 20-3. WDTRST – 看门狗复位寄存器

WDTCON Address = A6H					(只能写)			
不可位寻址								
	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
<p>The WDT is enabled by writing the sequence 1EH/E1H to the WDTRST SFR. The current status may be checked by reading the WDTEN bit in WDTCON. To prevent the WDT from resetting the device, the same sequence 1EH/E1H must be written to WDTRST before the time-out interval expires. A software reset is generated by writing the sequence 5AH/A5H to WDTRST.</p>								

21. 指令设置总结

AT89LP216 是和 MCS-51 的指令设置完全二进制式匹配的。AT89LP216 和标准 8051 之间的区别在于循环次数需要执行指令来完成。在 AT89LP216 中，指令可能需要 1, 2, 3 甚至 4 个时钟周期完成。大多数指令的执行时间可用表 21-1 来估算出。

Table 21-1. 指令执行时间和一些特例

Generic Instruction Types	Cycle Count Formula
Most arithmetic, logical, bit and transfer instructions	# bytes
Branches and Calls	# bytes + 1

Table 21-1. 指令执行时间和一些特例

Single Byte Indirect (i.e. ADD A, @Ri, etc.)				2
RET, RETI				4
MOVC				3
MOVX				4
MUL				2
DIV				4
INC DPTR				2
Arithmetic	Bytes	Clock Cycles		Hex Code
		8051	AT89LP	
ADD A, Rn	1	12	1	28-2F
ADD A, direct	2	12	2	25
ADD A, @Ri	1	12	2	26-27
ADD A, #data	2	12	2	24
ADDC A, Rn	1	12	1	38-3F
ADDC A, direct	2	12	2	35
ADDC A, @Ri	1	12	2	36-37
ADDC A, #data	2	12	2	34
SUBB A, Rn	1	12	1	98-9F
SUBB A, direct	2	12	2	95
SUBB A, @Ri	1	12	2	96-97
SUBB A, #data	2	12	2	94
INC Rn	1	12	1	08-0F
INC direct	2	12	2	05
INC @Ri	1	12	2	06-07
INC A	2	12	2	04
DEC Rn	1	12	1	18-1F
DEC direct	2	12	2	15
DEC @Ri	1	12	2	16-17
DEC A	2	12	2	14
Arithmetic	Bytes	Clock Cycles		Hex Code
		8051	AT89LP	
INC DPTR	1	24	2	A3
MUL AB	1	48	2	A4
DIV AB	1	48	4	84
DA A	1	12	1	D4
Logical	Bytes	Clock Cycles		Hex Code
		8051	AT89LP	

Table 21-1. 指令执行时间和一些特例

CLR A	1	12	1	E4
CPL A	1	12	1	F4
ANL A, Rn	1	12	1	58-5F
ANL A, direct	2	12	2	55
ANL A, @Ri	1	12	2	56-57
ANL A, #data	2	12	2	54
ANL direct, A	2	12	2	52
ANL direct, #data	3	24	3	53
ORL A, Rn	1	12	1	48-4F
ORL A, direct	2	12	2	45
ORL A, @Ri	1	12	2	46-47
ORL A, #data	2	12	2	44
ORL direct, A	2	12	2	42
ORL direct, #data	3	24	3	43
XRL A, Rn	1	12	1	68-6F
XRL A, direct	2	12	2	65
XRL A, @Ri	1	12	2	66-67
XRL A, #data	2	12	2	64
XRL direct, A	2	12	2	62
XRL direct, #data	3	24	3	63
RL A	1	12	1	23
RLC A	1	12	1	33
RR A	1	12	1	03
RRC A	1	12	1	13
SWAP A	1	12	1	C4
		Clock Cycles		
Data Transfer	Bytes	8051	AT89LP	Hex Code
MOV A, Rn	1	12	1	E8-EF
MOV A, direct	2	12	2	E5
MOV A, @Ri	1	12	2	E6-E7
MOV A, #data	2	12	2	74
MOV Rn, A	1	12	1	F8-FF
MOV Rn, direct	2	24	2	A8-AF
MOV Rn, #data	2	12	2	78-7F
MOV direct, A	2	12	2	F5
MOV direct, Rn	2	24	2	88-8F
MOV direct, direct	3	24	3	85

Table 21-1. 指令执行时间和一些特例

MOV direct, @Ri	2	24	2	86-87
MOV direct, #data	3	24	3	75
MOV @Ri, A	1	12	1	F6-F7
MOV @Ri, direct	2	24	2	A6-A7
MOV @Ri, #data	2	12	2	76-77
MOV DPTR, #data16	3	24	3	90
MOVC A, @A+DPTR	1	24	3	93
MOVC A, @A+PC	1	24	3	83
MOVX A, @Ri	1	24	4	E2-E3
MOVX A, @DPTR	1	24	4	E0
MOVX @Ri, A	1	24	4	F2-F3
MOVX @DPTR, A	1	24	4	F0
PUSH direct	2	24	2	C0
POP direct	2	24	2	D0
XCH A, Rn	1	12	1	C8-CF
XCH A, direct	2	12	2	C5
XCH A, @Ri	1	12	2	C6-C7
XCHD A, @Ri	1	12	2	D6-D7
CLR C	1	12	1	C3
CLR bit	2	12	2	C2
SETB C	1	12	1	D3
SETB bit	2	12	2	D2
CPL C	1	12	1	B3
CPL bit	2	12	2	B2
ANL C, bit	2	24	2	82
Bit Operations	Bytes	Clock Cycles		Hex Code
		8051	AT89LP	
ANL C, bit	2	24	2	B0
ORL C, bit	2	24	2	72
ORL C, /bit	2	24	2	A0
MOV C, bit	2	12	2	A2
MOV bit, C	2	24	2	92
Branching	Bytes	Clock Cycles		Hex Code
		8051	AT89LP	
JC rel	2	24	3	40
JNC rel	2	24	3	50
JB bit, rel	3	24	4	20

Table 21-1. 指令执行时间和一些特例

JNB bit, rel	3	24	4	30
JBC bit, rel	3	24	4	10
JZ rel	2	24	3	60
JNZ rel	2	24	3	70
SJMP rel	2	24	3	80
ACALL addr11	2	24	3	11,31,51,71,91,B1,D1,F1
LCALL addr16	3	24	4	12
RET	1	24	4	22
RETI	1	24	4	32
AJMP addr11	2	24	3	01,21,41,61,81,A1,C1,E1
LJMP addr16	3	24	4	02
JMP @A+DPTR	1	24	2	73
JMP @A+PC ⁽¹⁾	2	–	3	A5 73
CJNE A, direct, rel	3	24	4	B5
CJNE A, #data, rel	3	24	4	B4
CJNE Rn, #data, rel	3	24	4	B8-BF
CJNE @Ri, #data, rel	3	24	4	B6-B7
DJNZ Rn, rel	2	24	3	D8-DF
DJNZ direct, rel	3	24	4	D5
NOP	1	12	1	00
BREAK ⁽¹⁾	2	–	2	A5 00

Note: 1. 这个逃逸指令是对指令设置的延伸应用。

22. 片上调试系统

AT89LP216 片上调试 (OCD) 系统使用两线串行接口控制程序流程; 读取、修改、加写系统状态; 编写非易失性存储器。OCD 系统有以下特点:

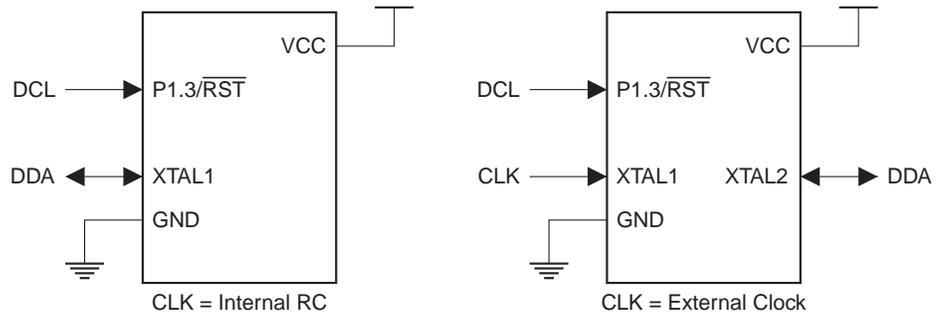
- 完全的程序流程控制
- 可对内部所有特殊功能寄存器和数据存储器进行读写和修改
- 四个硬件编程地址断点
- 通过 BREAK 指令设置软件断点
- 堆栈上溢 / 下溢时程序中止
- 看门狗溢出时程序中止
- 非入侵操作

· 非易失性存储器编程

22.1 物理接口

片上调试系统使用两线同步串行接口在目标器件和仿真系统之间建立通信。OCD接口由两个用户熔丝位控制。通过清零 Enable 熔丝位使能 OCD。当 OCD 使能时，RST 管脚被配置成调试时钟（DCL）的输入。根据被选择的时钟源，XTAL1 或 XTAL2 被配置成调试数据（DDA）的双向数据线。如果选择了外部时钟，XTAL2 就被配置成 DDA。如果选择了内部 RC 振荡器，XTAL1 就被配置成 DDA。OCD 接线见图 22-1。当 OCD 使能时，接口类型依据 OCD 接口选择用户熔丝位。这个熔丝位提供普通两线串口（TWI）和高速两线串口（FTWI）两种选择。用户必须在使能 OCD 的同时对该熔丝位进行合理配置。（见 71 页“用户配置熔丝位”）“[用户配置熔丝位](#)” on page 68).

Figure 22-1. AT89LP216 片上调试连线



当用到片上调试功能时，必须注意以下几点：

- P1.3/RST 不能和 VCC 直接相连，必须去除任何和 RST 相连的外部电容。
- 去除所有外部复位电压。
- 去除 XTAL1、XTAL2 上的石英晶体和电容。如果不想用内部 RC 振荡器，必须在 XTAL1 上驱动一个外部时钟信号，为了这个目的，有些仿真系统可能提供用户可配置的时钟。

22.2 软件断点

AT89LP216 微控制器包括一个 BREAK 指令，用来在软件内执行程序存储器断点。通过在代码中放置 BREAK 指令，可以人为插入软件断点。有些仿真系统可能允许自动添加 / 删除软件断点。每当一个软件断点改变时，闪存都要重新编程。频繁的增加 / 删除软件断点，会降低非易失性存储器的数据保持力。用于调试的器件禁止给任何客户。当 OCD 未使能时，BREAK 指令看作一个两周期的 NOP 指令。

22.3 片上调试的缺陷

AT89LP216 是一种低功耗，管脚少然而功能完善的微控制器，在它有限的 I/O 管脚上复合了一些功能。为了为片上调试提供资源一些功能必须牺牲。片上调试系统有以下一些缺陷：

- * 调试时钟管脚（DCL）和管脚 P1.3 还有外部复位管脚（RST）复用，所以当 OCD 使能时，P1.3 和外部复位都不可用。
- * 调试数据管脚（DDA）和 XTAL1/P3.2 或 XTAL2/P3.3 复用。所以在调试时，晶体振荡器是不被支持的。必须选择内部 RC 振荡器或外部时钟源来提供系统时钟。熔丝位指定晶体振荡器时，当 OCD 使能时默认外部时钟模式。

- * 当调试时使用内部 RC 振荡器时，DDA 位于 XTAL1/P3.2 管脚上，INT0 功能在此模式下失效。
- * 当调试时使用外部时钟时，DDA 位于 XTAL2/P3.3 管脚上，系统时钟驱动 XTAL1/P3.2。INT0, INT1 和 CLKOUT 功能在此模式下失效。
- * AT89LP216 不支持程序运行时编程，因此调试过程中当想改变代码时必须先复位器件，包括添加 / 删除软件断点。
- * 当使用看门狗溢出产生中断来代替复位时，看门狗的状态将不会被复位。为了保证看门狗正常工作，必须在继续工作之前向器件发送 OCD 复位命令。

23. 闪存编程

AT89LP216 微控制器有 2K 字节的片上在线可编程程序闪存。在线编程允许微控制器在终端系统中进行反复的编程。在线编程器使用简单的四线 SPI 接口和 AT89LP216 进行串行通信，对片上所有非易失性存储器重新编程。在线编程消除了从系统中移除芯片的需要，无论是实验室研发还是升级软件或参数，都可以节约时间和金钱。AT89LP216 的 ISP 接口具有以下特点：

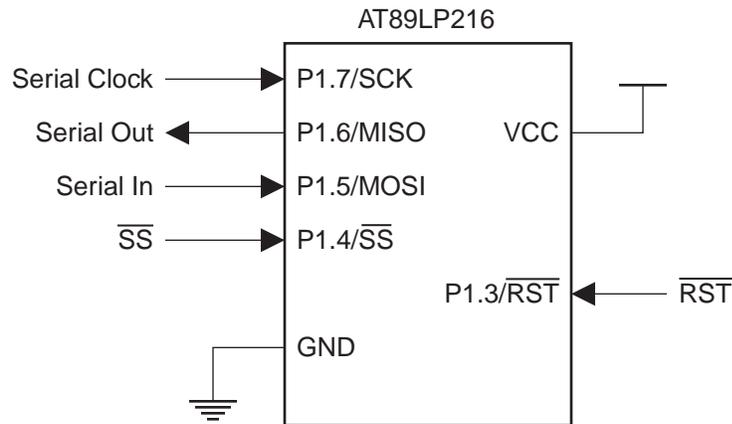
- * 四线 SPI 编程接口
- * 低有效复位编程入口
- * 从机选择允许同一个接口接多个器件
- * 用户签名队列
- * 灵活的页编程
- * 行擦除能力
- * 带自动擦除命令的页写入
- * 编程状态寄存器

要了解在线编程的更多详情，参考应用手册“AT89LP 在线编程说明”。

23.1 物理接口

在线编程使用 AT89LP216 的串行外围接口（SPI）管脚。SPI 是全双工同步串行接口，它由四根线组成：串行时钟（SCK），主进/从出（MISO），主出/从进（MOSI）和低有效的从机选择（SS）。当对 AT89LP216 编程时，编程器总是作为 SPI 主机，目标系统总是作为 SPI 从机。要进入或保持在线编程模式，器件的复位线（RST）必须保持有效（低）。除了 VCC 和 GND，AT89LP216 可以最少只用七根连线就能进行编程，如图 23-1。Figure 23-1.

Figure 23-1. 在线编程器件连线



在线编程接口是外部对 AT89LP216 进行编程的唯一方法。ISP 接口可以在线或在独立的串行编程器中对器件进行编程。ISP 接口除了 SCK 不需要任何时钟而且不受系统时钟频率的限制。在线编程时，目标器件的系统时钟源可以正常工作。

当设计一个需要使用在线编程的系统时，必须注意以下几点：

- * ISP 接口只使用 SPI 时钟模式 0（CPOL=0, CPHA=0），最高频率为 5MHZ。

- * AT89LP216 只有在复位线（RST）有效（低）时才会进入编程模式。为了简化这个步骤，建议目标复位由在线编程器控制。为了避免出现问题，在线编程器必须能够在整个编程周期内保持整个目标系统复位。目标系统禁止在复位线有效时驱动四根 SPI 线。

?RST 输入可能不能够获得额外的 I/O 引脚。这种情况下，RST 引脚在上电期间总是作为复位引脚。为了进行编程，必须在带电复位（POR）结束之前拉低 RST 引脚。POR 结束之后，除非 RST 被拉高，否则器件一直保持 ISP 模式。一旦初始化 ISP 操作结束，目标器件上的电源必须循环开关进入另一次操作。

- * 当 ISP 使能时，在复位期间，SS 引脚禁止悬空。

- * 在任何复位期间，ISP 使能熔丝位必须置位才能进行编程。如果 ISP 熔丝位被禁止，那么只有在 POR 时才能进入 ISP。

23.2 存储器结构

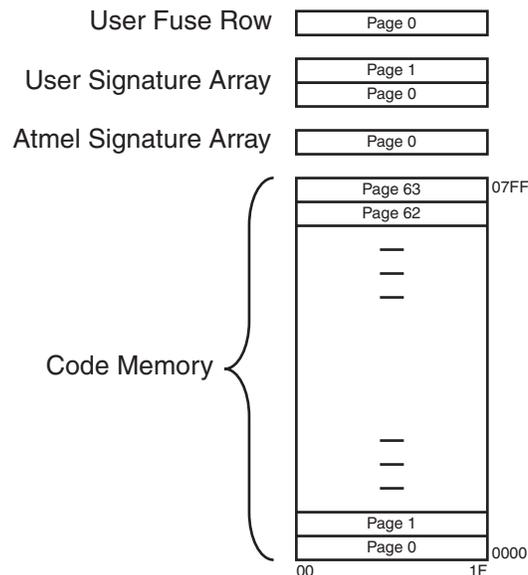
AT89LP216 提供 2K 字节的在线可编程（ISP）的非易失性闪存。除此之外，它还包含一个 64 字节的用户签名队列和一个 32 字节只读的 Atmel 签名队列。内存结构见表 23-1 和图 23-2。存储器被分成一些页，每页有 32 字节。一个读或写指令只能进入一页。每种存储器占据自己的地址空间，由专门针对该存储器的指令进行访问。然而所有类型的存储器页大小都一样。

用户配置熔丝位在存储器中映射为一行，每个字节代表一个熔丝位。从编程的角度来看，熔丝位可看成是一般的代码字节，当然它们不受芯片擦除的影响。在熔丝位那一行的适当位置写入 00h 就可以使能熔丝位。然而当想禁止熔丝位，例如写入 FFh，必须先擦除整个熔丝位所在的行然后再重新编程。编程器必须先记下所有熔丝位的状态，然后修改需要禁止的熔丝位，最后发出一个带自动擦除的写熔丝位命令，数据就是前面记下的熔丝位的状态。锁定位的操作情况和熔丝位相似，不过它们只能通过芯片擦除的方法解除锁定状态。

Table 23-1. Code Memory Size

设备 #	程序空间	页大小	页数	地址范围
AT89LP216	2K bytes	32 bytes	64	0000H - 07FFH

Figure 23-2. 程序存储器大小



23.3 命令格式

编程命令由一个操作码字节，两个地址字节和零或更多数据字节构成。除此之外，所有命令数据包必须以 AAH 和 55H 开头。这个开头能够使编程接口增加抗干扰性。68 页图 23 - 3 显示了一个简化的命令流程图。

68 页图 23 - 4 显示了命令数据包的一个样例。SS 管脚定义了包帧。命令中第一个字节发送之前 SS 必须拉低，命令中最后一个字节发送之后 SS 必须被拉回到高的状态。直到 SS 返回高命令才传输完成。命令字节在 MOSI 上串行传输。MISO 上串行接收数据输出字节。通过在最后一个字节传输完毕时 SS 返回高，就可以支持可变长的包。有时，命令字节中有些不用理睬的值，如果这些值在包的当中就必须传输，如果在结尾部分则可以忽略。

面向页的指令总是包含一个完整的 16 位地址。高位选择页数，低位选择字节。AT89LP216 分配 5 位给字节地址 6 位给页地址。当传输时要访问的页总是固定的。字节地址指定第一个数据字节的起始地址。当一个数据字节传输完毕，字节地址就增加指向下面的数据字节。这就可以让页命令

线性扫描一页中的字节。如果字节地址超过了页中的最后一个字节，字节地址将滚回到同一页的第一个字节处。当往页缓冲中载入字节时，过载数据可能引起数据冲突。

69 页表 23 - 2 显示了一些常用命令。

[Table 23-2 on page 66.](#)

Figure 23-3. 命令流程图 t

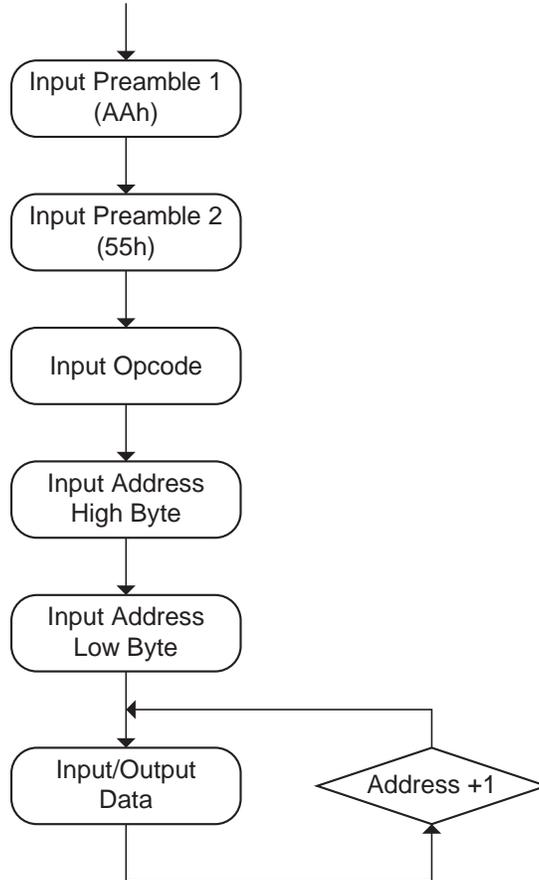


Figure 23-4. ISP 命令数据包

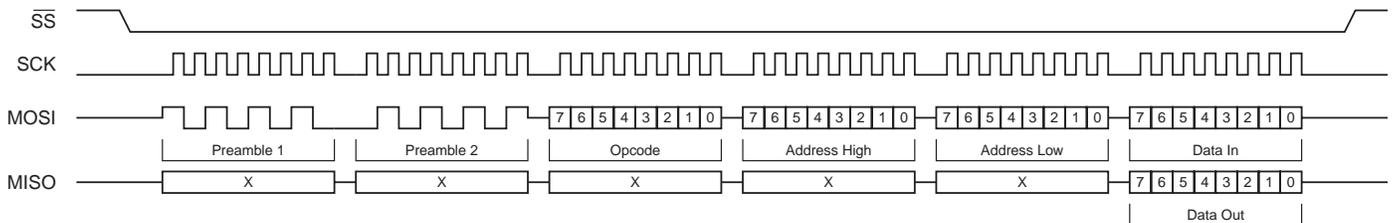


Table 23-2. 编程命令总结

命令	操作码	高地址	低地址	数据 0	数据 n
编程使能 ⁽¹⁾	1010 1100	0101 0011	–	–	–
片擦除	1000 1010	–	–	–	–
读状态	0110 0000	xxxx xxxx	xxxx xxxx	Status Out	
页下载缓冲 ⁽²⁾	0101 0001	xxxx xxxx	xxx b bbbb	DataIn 0 ... DataIn n	
写代码页 ⁽²⁾	0101 0000	xxxx xaaa	aaab bbbb	DataIn 0 ... DataIn n	
带自动擦除的写代码页 ⁽²⁾	0111 0000	xxxx xaaa	aaab bbbb	DataIn 0 ... DataIn n	
读代码页 ⁽²⁾	0011 0000	xxxx xaaa	aaab bbbb	DataOut 0 ... DataOut n	
写用户熔丝 ⁽²⁾⁽³⁾⁽⁴⁾	1110 0001	0000 0000	000b bbbb	DataIn 0 ... DataIn n	
带自动擦除功能的写用户熔丝 ⁽²⁾⁽³⁾⁽⁴⁾	1111 0001	0000 0000	000b bbbb	DataIn 0 ... DataIn n	
读用户熔丝 ⁽²⁾⁽³⁾⁽⁴⁾	0110 0001	0000 0000	000b bbbb	DataOut 0 ... DataOut n	
写锁定位 ⁽²⁾⁽³⁾⁽⁵⁾	1110 0100	0000 0000	000b bbbb	DataIn 0 ... DataIn n	
读锁定位 ⁽²⁾⁽³⁾⁽⁵⁾	0110 0100	0000 0000	000b bbbb	DataOut 0 ... DataOut n	
写用户签名页 ⁽²⁾	0101 0010	xxxx xxxx	xaab bbbb	DataIn 0 ... DataIn n	
带擦除的写用户签名页 ⁽²⁾	0111 0010	xxxx xxxx	xaab bbbb	DataIn 0 ... DataIn n	
读用户签名页 ⁽²⁾	0011 0010	xxxx xxxx	xaab bbbb	DataOut 0 ... DataOut n	
读 Atmel 签名页 ⁽²⁾⁽⁶⁾	0011 1000	xxxx xxxx	xxx b bbbb	DataOut 0 ... DataOut n	

- Notes:**
1. 使能命令必须是在进入编程模式之后首个发送的命令。
 2. 从 1 到 32 的任何数量的数据字节都可能被读写，地址每隔一个字节就增加。
 3. 每个字节地址选择一个熔丝位或锁定位。数据字节必须是 00h 或 FFh。
 4. 见 71 页表 23 – 5 熔丝位定义。
 5. 见 70 页表 23 – 4 锁定位定义。
 6. **AAtmel 签名字节**

AT89LP216: 地址 00H = 1EH
 01H = 29H
 02H = FFH

7. 关键字

- a: 页地址位
- b: 字节地址位
- x: 无关位

23.4 状态寄存器

可以通过读取状态寄存器访问存储器状态。表 23 - 3 显示状态寄存器。

Table 23-3. 状态寄存器

	-	-	-	-	LOAD	SUCCESS	WRTINH	BUSY
Bit	7	6	5	4	3	2	1	0

符号	功能
LOAD	下载标志。下载页缓冲命令清零，下次存储器写命令置高。此标志显示页缓冲之前通过下载页缓冲命令装载过数据
SUCCESS	成功标志。编程周期开始时清零，编程周期结束并且无误时置高。
WRTINH	写禁止标志。当 VCC 降到最小编程电压之下导致编程被禁止时，该位就会被警戒灯火管制检测器（BOD）拉低，此时，SUCCESS 标志在周期结束时保持低状态。WRTINH 为低时 BUSY 线也会拉低。
BUSY	忙标志。当存储器忙于编程时或写被禁止时标志清零。

23.5 DATA 轮询

AT89LP216 执行 DATA 轮询来显示编程周期结束。当器件忙时，任何试图读取最后一个字节的行为都会返回一个最高位为 1 的数据字节。一旦编程周期结束，会获得真正的值。在擦除期间，假设数据为 FFh，则 DATA 轮询会返回 7FH。当在一页中写多个字节，DATA 的值是编程开始之前最后一个载入的数据字节，而不是一页中物理地址最高的那个字节。

23.6 闪存安全

AT89LP216 为了保证程序闪存的安全提供了两个锁定位。锁定位可以不被编程（FFh）也可以被编程（00h）来获得表 23-4 所列出的保护水平。锁定位只能通过芯片擦除来擦除（FFh）。锁定位模式 2 禁止了对所有存储器空间的编程，包括用户签名队列和用户配置熔丝位。用户熔丝位必须在使能锁定位模式 2 或 3 之前编写。锁定位模式 3 执行模式 2 的操作，并且也阻塞对程序存储器的读取，然而，对用户签名队列、Atmel 签名队列和用户配置熔丝位的读取仍旧是允许的。

Table 23-4. 锁定位保护模式

程序锁定位 (按地址排列)			保护模式
模式	00h	01h	
1	FFh	FFh	没有程序锁定
2	00h	FFh	禁止对闪存的高级编程
3	00h	00h	禁止对闪存的高级编程，禁止读取闪存，禁止 OCD

23.7 用户配置熔丝位

AT89LP216 包括 19 个用户熔丝位用来对器件进行配置。每个熔丝位在用户熔丝位行中占据各自的位置，见表 23-5。熔丝位通过编写 00h 清零，对熔丝位编写 FFh 可使其保持之前的状态。要置位

熔丝位（置成 FFh），熔丝位行必须被清零然后用带自动擦除功能的熔丝位写命令来重新编写。所有熔丝位的默认状态是 FFh。

Table 23-5. 用户配置熔丝位定义

地址	熔丝名称	描述
00 – 01h	时钟源 – CS[0:1] ⁽²⁾	选择系统时钟源： CS1 CS0 时钟源 00h 00h 晶体振荡器 (XTAL) 00h FFh 保留 FFh 00h 外部时钟 XTAL1 (XCLK) FFh FFh 内部 RC 振荡器 (IRC)
02 – 03h	开始时间 – SUT[0:1]	选择 POR/BOD/PWD 唤醒的延时时间： SUT1 SUT0 Selected Time-out 00h 00h 1 ms (XTAL); 16 μs (XCLK/IRC) 00h FFh 2 ms (XTAL); 512 μs (XCLK/IRC) FFh 00h 4 ms (XTAL); 1 ms (XCLK/IRC) FFh FFh 16 ms (XTAL); 4 ms (XCLK/IRC)
04h	管脚复位使能 ⁽³⁾	FFh: RST 管脚作为复位管脚 00h: RST 管脚作为通用 I/O
05h	警戒灯火管制检测器使能	FFh: 警戒灯火管制检测器使能 00h: 警戒灯火管制检测器禁止
06h	片上调试使能	FFh: 在线编程使能 00h: 在线编程禁止（只在 POR 时使能）
07h	ISP 使能 ⁽³⁾	FFh: ISP 使能 00h: ISP 禁用 (POR)
08 – 0FH	RC 振荡器频率调整 [0:7]	调整内部 RC 振荡器频率
10H	用户签名编程	FFh: 用户签名编程禁止 00h: 用户签名编程使能
11H	三态管脚	FFh: 复位之后 I/O 管脚以只输入模式启动 00h: 复位之后 I/O 管脚以准双向模式启动
12H	OCD 接口选择	FFh: 普通两线接口 00h: 快速两线接口

- Notes:
1. 所有熔丝位的默认状态是 FFh。
 2. 对熔丝位的更改只有在器件 POR 之后才有效。
 3. 对熔丝位的更改只有在 ISP 操作结束时把 $\overline{\text{RST}}$ 拉高时才有效。

23.8 编程接口时序

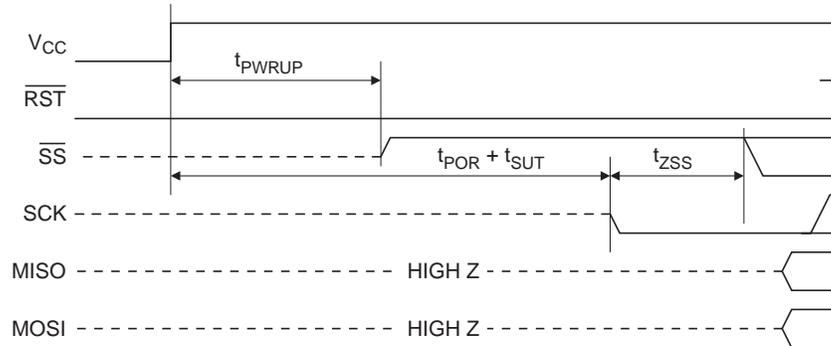
本小节详细论述系统通用时序、进入退出在线编程的约束条件和 ISP 中与串行外围接口有关的参数。以下波形图的通用时序参数见 75 页“时序参数”一节。

23.8.1 上电时序

执行这个时序之后就能在上电之后立即进入编程模式。如果 RST 管脚或 ISP 熔丝位被禁止，那么这就是唯一进入编程模式的方法。（见 14 页“外部复位”）。

1. 在 VCC 和 GND 之间供电。RST 必须保持低。
2. 等待至少 t_{PWRUP} ，然后驱动 SS 为高
3. 等待至少 t_{SUT} 到内部上电复位结束。 t_{SUT} 的值取决于器件的电路设置。
4. 开始编程操作。

Figure 23-5. 串行编程上电时序

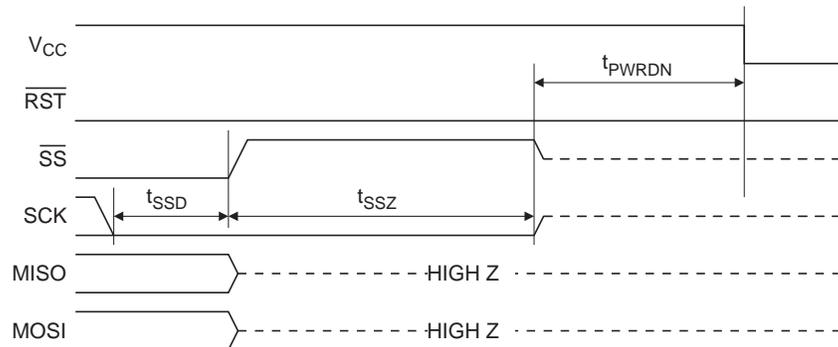


23.8.2 下电时序

执行该时序之后在编程完成后断电。

1. 驱动 SCK 低。
2. 等待至少 t_{SSD} ，然后把 SS 拉高。
3. MOSI 三态。
4. 等待至少 t_{SSZ} 然后 SS 和 SCK 变成三态。
5. 等待不超过 t_{PWRDN} 然后 VCC 断电。

Figure 23-6. 串行编程下电时序

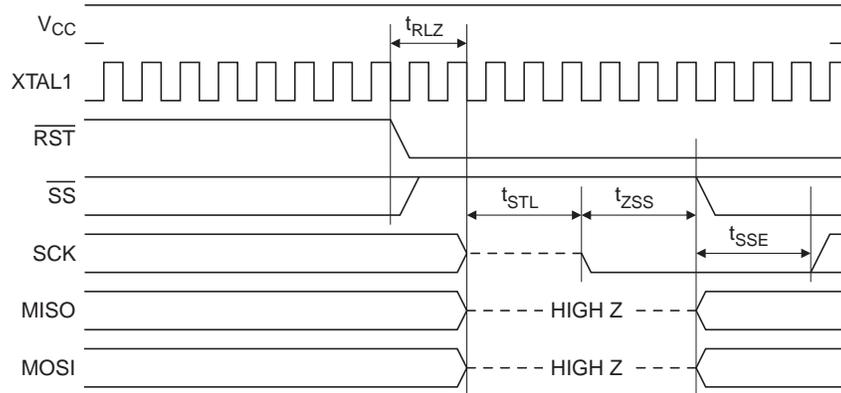


23.8.3 ISP 启动时序

当器件已经通过上电复位并且已经在操作时，执行这个时序将退出 CPU 执行模式进入 ISP 模式。

1. $\overline{\text{RST}}$ 拉低。
2. $\overline{\text{SS}}$ 拉高。
3. 等待 $t_{\text{RLZ}} + t_{\text{STL}}$ 。
4. 开始编程操作。

Figure 23-7. 在线编程 (ISP) 启动时序

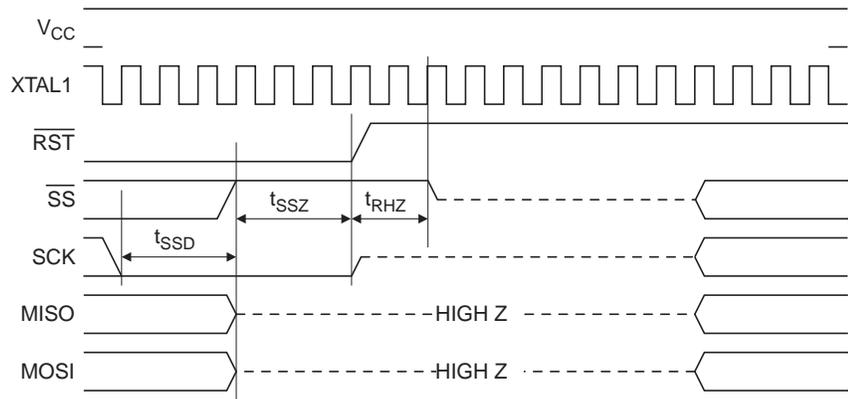


23.8.4 ISP 退出时序

执行该时序将退出 ISP 模式继续 CPU 执行模式。

1. SCK 拉低。
1. 等待至少 t_{SSD} 然后将 $\overline{\text{SS}}$ 拉高。
2. MOSI 三态。
3. 等待至少 t_{SSZ} 然后拉高 $\overline{\text{RST}}$ 。
4. SCK 三态。
5. 等待 t_{RHZ} 然后 $\overline{\text{SS}}$ 三态。

Figure 23-8. 在线编程 (ISP) 退出时序



Note: 本页各波形不按比例。

23.8.5 串行外围接口

串行外围接口 (SPI) 是面向字节的全双工同步串行通信通道。在在线编程中，编程器总是作为 SPI 主机，目标器件总是作为 SPI 从机。目标器件在 MOSI 上接收串行数据，在 MISO 上输出串行数据。编程接口作为标准的 SPI 端口，数据次序固定。对于在线编程而言，MSB 第一个传输，如图 23-9 所示。SCK 相位和极性跟随 SPI 时钟模式 0 (CPOL = 0, CPHA = 0)，在 SCK 上升沿采样，在 SCK 下降沿输出。欲知更多时序信息见图 23-10。

Figure 23-9. ISP 字节时序

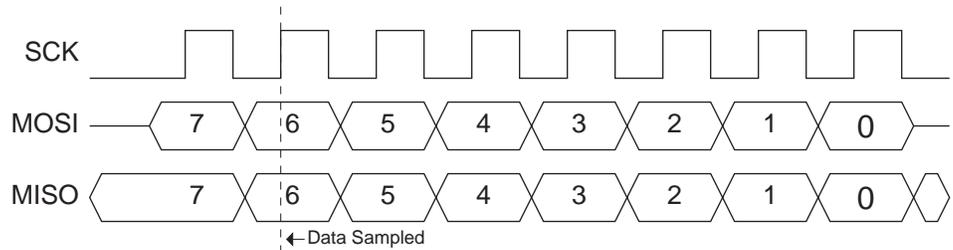
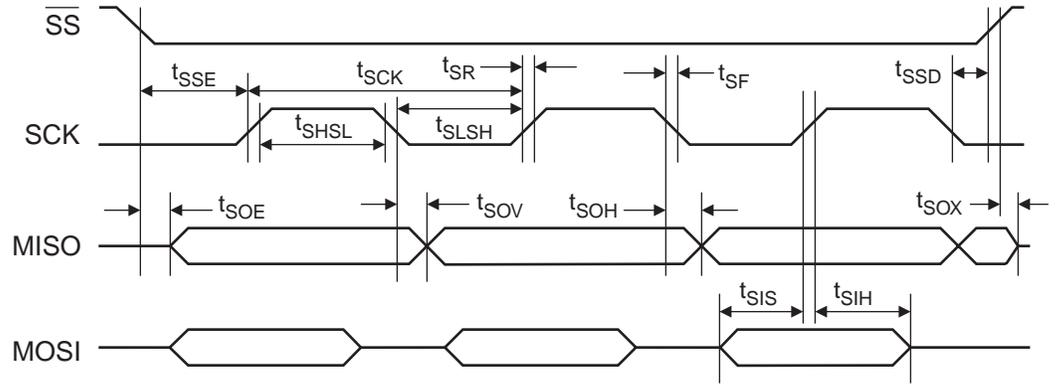


Figure 23-10. 串行编程接口时序



23.8.6 时序参数

图 23-5, 图 23-6, 图 23-7, 图 23-8 和图 23-10 的时序参数见表。

Table 23-6. 编程接口时序参数

符号	参数	最小	最大	单位
t_{CLCL}	系统时钟循环时间	0	60	ns
t_{PWRUP}	上电到 \overline{SS} 为高的时间	10		μ s
t_{POR}	上电复位时间		100	μ s
t_{PWRDN}	\overline{SS} 三态到断电		1	μ s
t_{RLZ}	\overline{RST} 低到 I/O 三态	t_{CLCL}	$2 t_{CLCL}$	ns
t_{STL}	\overline{RST} 处于低的时间	100		ns
t_{RHZ}	\overline{RST} 高到 \overline{SS} 三态	0	$2 t_{CLCL}$	ns
t_{SCK}	串行时钟循环时间	200 ⁽¹⁾		ns
t_{SHSL}	时钟高的时间	75		ns
t_{SLSH}	时钟低的时间	50		ns
t_{SR}	上升时间		25	ns
t_{SF}	下降时间		25	ns

Table 23-6. 编程接口时序参数

符号	参数	最小	最大	单位
t_{SIS}	串行输入设置时间	10		ns
t_{SIH}	串行输入保持时间	10		ns
t_{SOH}	串行输出保持时间		10	ns
t_{SOV}	串行输出有效时间		35	ns
t_{SOE}	输出使能时间		10	ns
t_{SOX}	输出禁止时间		25	ns
t_{SSE}	\overline{SS} 使能引导时间	t_{SLSH}		ns
t_{SSD}	\overline{SS} 禁止延迟时间	t_{SLSH}		ns
t_{ZSS}	SCK 设置到 \overline{SS} 低	25		ns
t_{SSZ}	\overline{SS} 高之后 SCK 保持时间	25		ns
t_{WR}	写循环时间	2.5		ms
t_{AWR}	带自动擦除写循环时间	5		ms
t_{ERS}	芯片擦除循环时间	7.5		ms

Note: 1. t_{SCK} 独立于 t_{CLCL} .

24. 电器特性

24.1 绝对极限值 *

操作温度	-40°C to +85°C
存储温度	-65°C to +150°C
任意管脚对地电压	-0.7V to +5.5V
最大工作电压	5.5V
直流输出电流	15.0 mA

*NOTICE: 如果强制芯片在超出“绝对极限值”表中所列的条件之下工作可能会造成器件的永久性损坏。这仅是工作应力的极限。并不表示器件可以工作于表中所列条件之下，或是那些超越工作范围明确规定的其他条件之下。长时间工作于绝对极限值可能会影响器件的寿命。

24.2 直流特性

$T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 2.4\text{V}$ to 5.5V (除非另行标注)

符号	参数	条件	最小	最大	单位	
V_{IL}	输入低电压		-0.5	$0.2 V_{CC} - 0.1$	V	
V_{IH}	输入高电压		$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V	
V_{OL}	输出低电压 (Ports 1, 3) ⁽¹⁾ 809	$I_{OL} = 10\text{ mA}$, $V_{CC} = 2.7\text{V}$, $T_A = 85^\circ\text{C}$		0.5	V	
V_{OH}	输出低电压 (Ports 1, 3) 带弱上拉使能	$I_{OH} = -80\ \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$	2.4		V	
		$I_{OH} = -30\ \mu\text{A}$	$0.75 V_{CC}$		V	
		$I_{OH} = -12\ \mu\text{A}$	$0.9 V_{CC}$		V	
V_{OH1}	输出高电压 (Ports 1, 3) 带强上拉使能	$I_{OH} = -10\text{ mA}$, $T_A = 85^\circ\text{C}$	$0.9 V_{CC}$			
I_{IL}	逻辑 0 输入电流 (Ports 1, 3)	$V_{IN} = 0.45\text{V}$		-50	μA	
I_{TL}	逻辑 1 到 0 转换电流 (Ports 1, 3)	$V_{IN} = 2\text{V}$, $V_{CC} = 5\text{V} \pm 10\%$		-750	μA	
I_{LI}	输入漏电流 (Port P1.0, P1.1)	$0 < V_{IN} < V_{CC}$		± 10	μA	
V_{OS}	比较输入偏移电压	$V_{CC} = 5\text{V}$		20	mV	
V_{CM}	普通模式下比较输入电压		0	V_{CC}	V	
RRST	复位下拉电阻		50	150	K Ω	
C_{IO}	复位下拉电阻	测试频率 = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF	
I_{CC}	电源供电电流	激活模式, 12 MHz, $V_{CC} = 5.5\text{V}/3\text{V}$		5.5/3.5	mA	
		空闲模式, 12 MHz, $V_{CC} = 5.5\text{V}/3\text{V}$ $P1.0 \& P1.1 = 0\text{V}$ or V_{CC}		1.2/1	mA	
	掉电模式 (2)	$V_{CC} = 5.5\text{V}$, $P1.0 \& P1.1 = 0\text{V}$ or V_{CC}			100	μA
		$V_{CC} = 3\text{V}$, $P1.0 \& P1.1 = 0\text{V}$ or V_{CC}			20	μA

Notes: 1. 在稳定状态(无瞬态产生)条件下, I_{OL} 必须严格作如下限制: 每个端口引脚的 I_{OL} 不超出 10 mA 所有输出引脚的 I_{OL} 总和不超过 15 mA 如果 I_{OL} 超过出了测试条件, V_{OL} 可能会超过相关标准规定, 此时不能保证引脚能吸收比测试条件下更大的电流。

掉电时最小 VCC 为 2V。

24.3 串行外围接口时序

Table 24-1. 主机模式下 SPI 特性

符号	参数	最小	最大	单位
t_{CLCL}	振荡器周期	41.6		ns
t_{SCK}	串行时钟周期	$4t_{CLCL}$		ns
t_{SHSL}	时钟高电平时间	$t_{SCK}/2 - 25$		ns
t_{SLSH}	时钟低电平时间	$t_{SCK}/2 - 25$		ns
t_{SR}	上升时间		25	ns
t_{SF}	下降时间		25	ns
t_{SIS}	串行输入建立时间	10		ns
t_{SIH}	串行输入保持时间	10		ns
t_{SOH}	串行输出保持时间		10	ns
t_{SOV}	串行输出有效时间		35	ns

Table 24-2. 从机模式下 SPI 特性

符号	参数	最小	最大	单位
t_{CLCL}	振荡器周期	41.6		ns
t_{SCK}	串行时钟周期	$4t_{CLCL}$		ns
t_{SHSL}	时钟高电平时间	$1.5 t_{CLCL} - 25$		ns
t_{SLSH}	时钟低电平时间	$1.5 t_{CLCL} - 25$		ns
t_{SR}	上升时间		25	ns
t_{SF}	下降时间		25	ns
t_{SIS}	串行输入建立时间	10		ns
t_{SIH}	串行输入保持时间	10		ns
t_{SOH}	串行输出保持时间		10	ns
t_{SOV}	串行输出确认时间		35	ns
t_{SOE}	输出使能时间		10	ns
t_{SOX}	输出禁止时间		25	ns
t_{SSE}	从机使能引导时间	10		ns
t_{SSD}	从机禁止滞后时间	0		ns

Figure 24-1. 主机模式下 SPI 时序 (CPHA = 0)

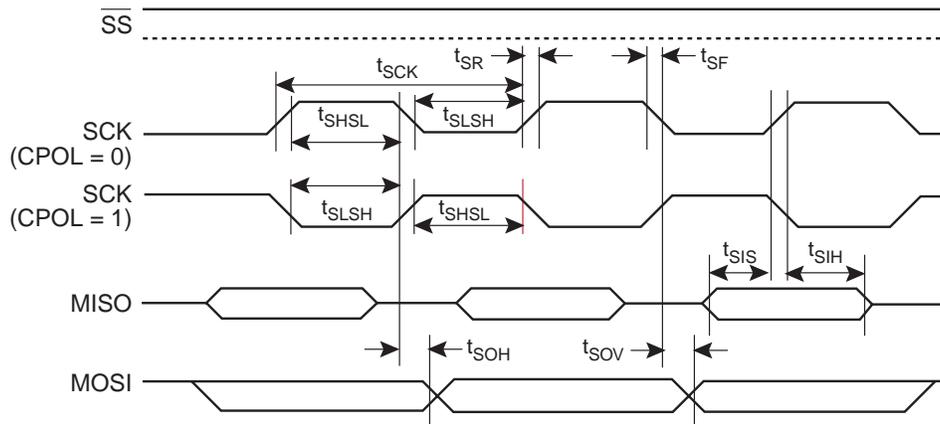


Figure 24-2. 从机模式下 SPI 时序 (CPHA = 0)

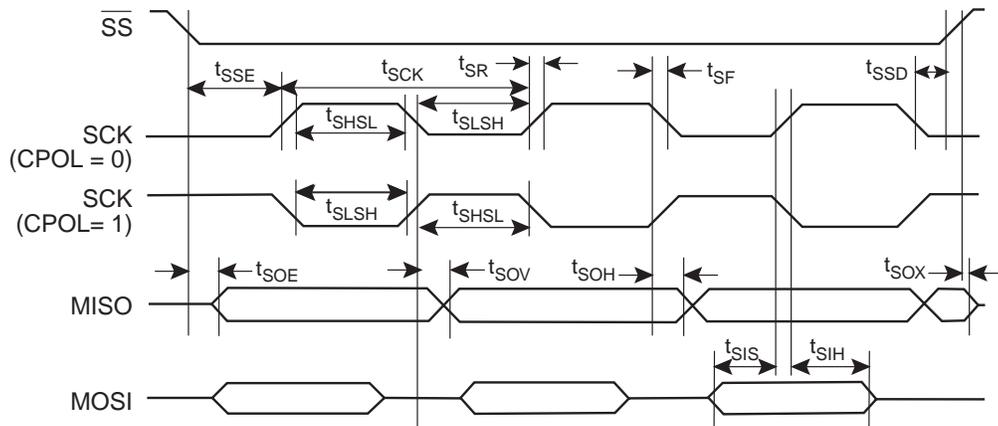


Figure 24-3. 主机模式下 SPI 时序 (CPHA = 1)

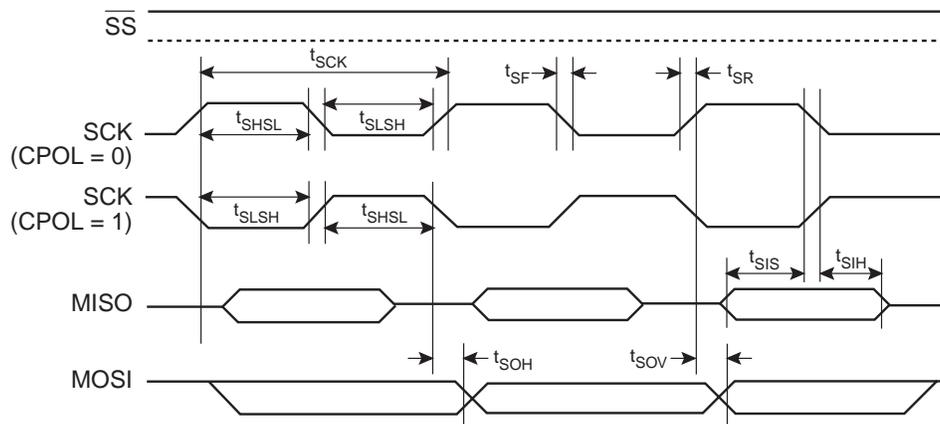
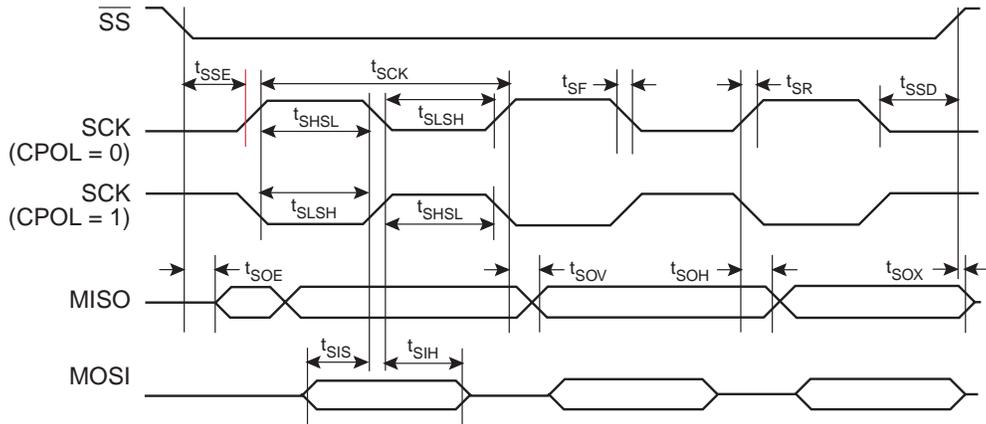
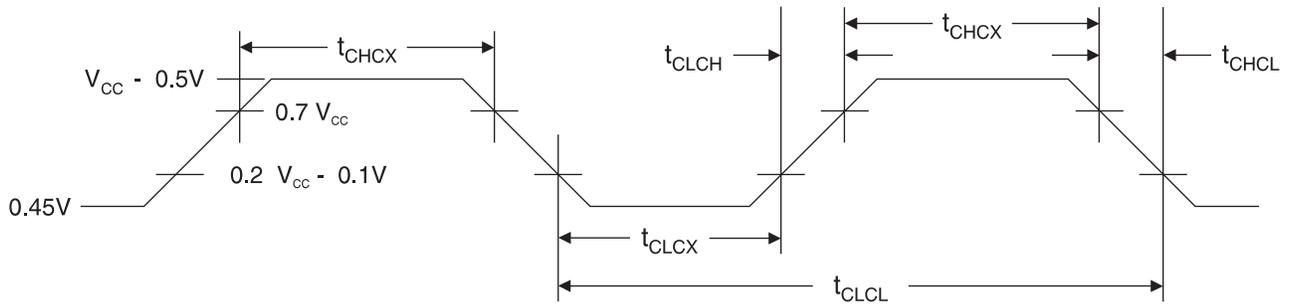


Figure 24-4. 从机模式下 SPI 时序 (CPHA = 1)



24.4 外部时钟驱动

Figure 24-5. 外部时钟驱动波形



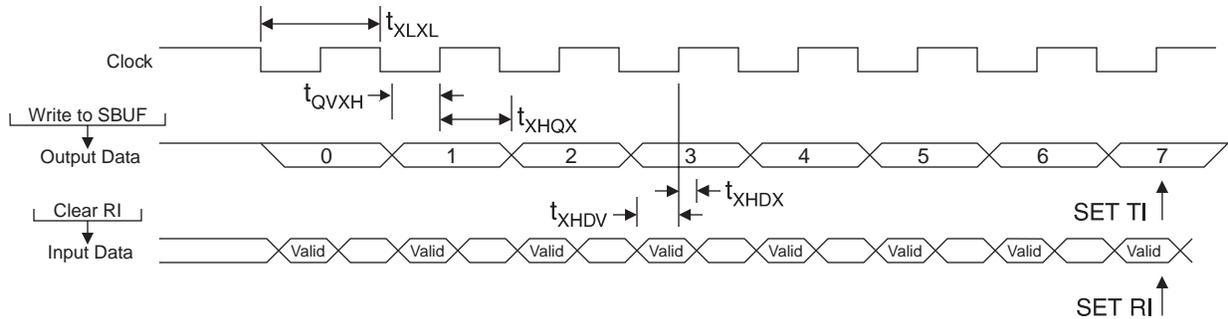
符号	参数	V _{CC} = 2.4V to 5.5V		单位
		最小	最大	
1/t _{CLCL}	Oscillator Frequency	0	20	MHz
t _{CLCL}	Clock Period	50		ns
t _{CHCX}	High Time	12		ns
t _{CLCX}	Low Time	12		ns
t _{CLCH}	Rise Time		5	ns
t _{CHCL}	Fall Time		5	ns

24.5 串口时序：移位寄存器模式

下表各值的测试条件为 $V_{CC} = 2.4V$ 到 $5.5V$ ，负载电容 = 80 pF 。

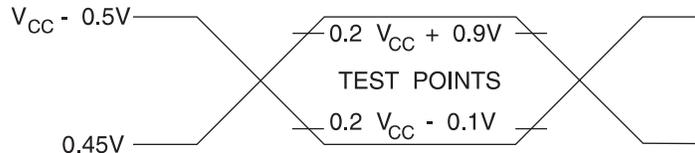
符号	参数	可变振荡器		单位
		最小	最大	
t_{XLXL}	串口时钟周期	$2t_{CLCL} - 15$		μs
t_{QVXH}	输出数据建立到时钟上升沿时间	$t_{CLCL} - 15$		ns
t_{XHGX}	时钟上升沿后输出数据保持时间	$t_{CLCL} - 15$		ns
t_{XHDX}	时钟上升沿后输入数据保持时间	0		ns
t_{XHDX}	输入数据有效到时钟上升沿时间	15		ns

Figure 24-6. 移位寄存器模式



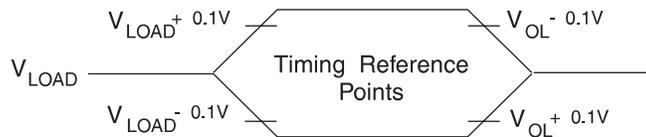
24.6 测试条件

24.6.1 交流测试输入 / 输出波形 (1)



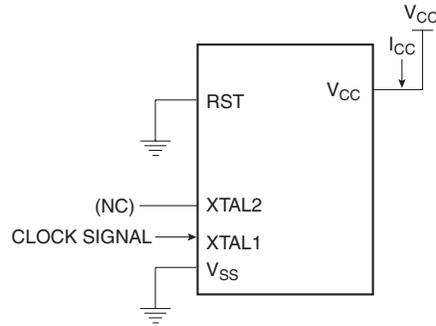
Note: 1. 交流输入测试的驱动条件 $V_{CC} - 0.5V$ 为逻辑“1”， $0.45V$ 为逻辑“0”。时序的测试条件 V_{IH} 为逻辑“1”的最小值， V_{IL} 为逻辑“0”的最大值。

24.6.2 悬浮状态的波形 (1)

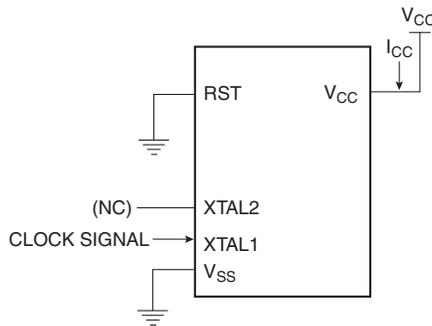


Note: 1. 1. 因为时序的要求，当负载电压有 $100mV$ 的变化时端口引脚状态不再悬浮，当有 $100mV$ 电压在 V_{OH}/V_{OL} 状态下变化时，端口的引脚状态变为悬浮。

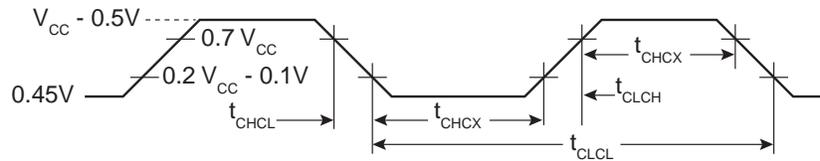
24.6.3 ICC 测试条件，运行模式，所有其他引脚断开



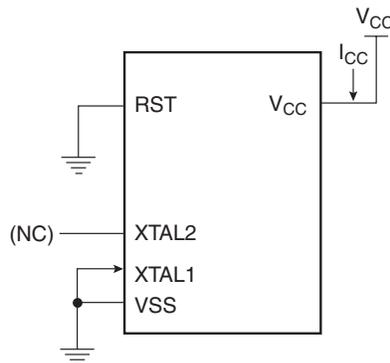
24.6.4 ICC 测试条件，空闲模式，所有其他引脚断开



24.6.5 运行和空闲模式下 ICC 时钟信号的波形， $t_{CLCH} = t_{CHCL} = 5 \text{ ns}$



24.6.6 ICC 测试条件，空闲模式，所有其他引脚断开， $V_{CC} = 2V \text{ to } 5.5V$



25. 订购信息

25.1 标准封装

速度 (MHz)	电源供应	订购号	封装	适用范围
20	2.4V to 5.5V	AT89LP216-20PI AT89LP216-20SI AT89LP216-20XI	16P3 16S2 16X	工业 (-40°C to 85°C)

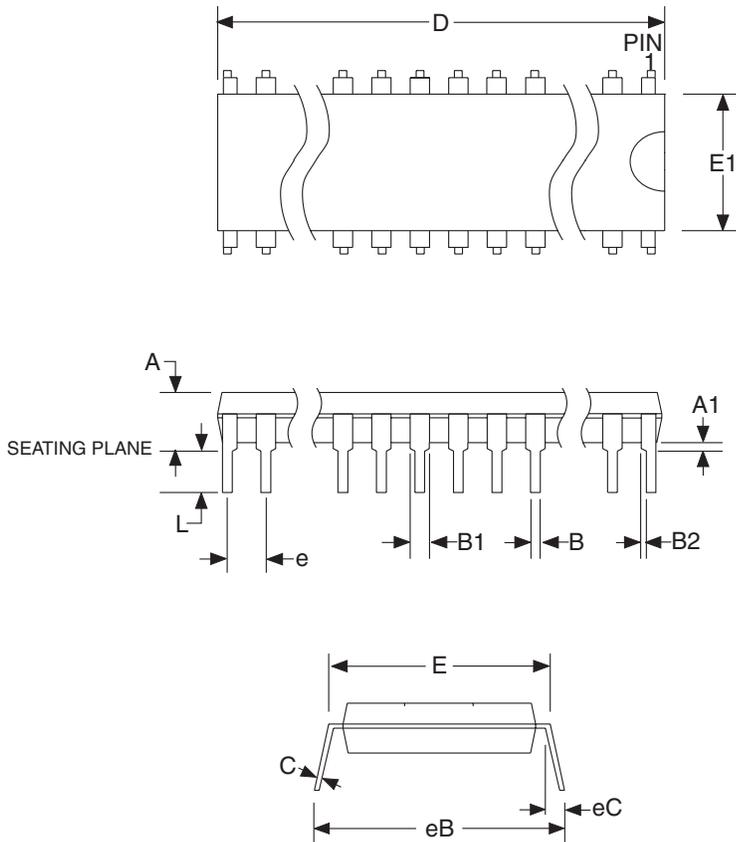
25.2 绿色封装 (无铅汞)

速度 (MHz)	电源供应	订购号	封装	适用范围
20	2.4V to 5.5V	AT89LP216-20PU AT89LP216-20SU AT89LP216-20XU	16P3 16S2 16X	工业 (-40°C to 85°C)

封装类型	
16P3	16-lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)
16S2	16-lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)
16X	16-lead, 0.173" Wide, Plastic Thin Shrink Small Outline Package (TSSOP)

26. Packaging Information

26.1 16P3 – PDIP



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	5.334	
A1	0.381	–	–	
D	19.81	–	20.32	Note 2
E	7.620	–	8.255	
E1	6.096	–	7.112	Note 2
B	0.356	–	0.559	
B1	1.143	–	1.778	
B2	0.762	–	1.143	
L	2.921	–	3.810	
C	0.203	–	0.356	
eB	–	–	10.922	
eC	0.000	–	1.524	
e	2.540 TYP			

- Notes: 1. This package conforms to JEDEC reference MS-001, Variation AB.
 2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

11/02/05



2325 Orchard Parkway
San Jose, CA 95131

TITLE

16P3, 16-lead (0.300"/7.62 mm Wide) Plastic Dual
Inline Package (PDIP)

DRAWING NO.

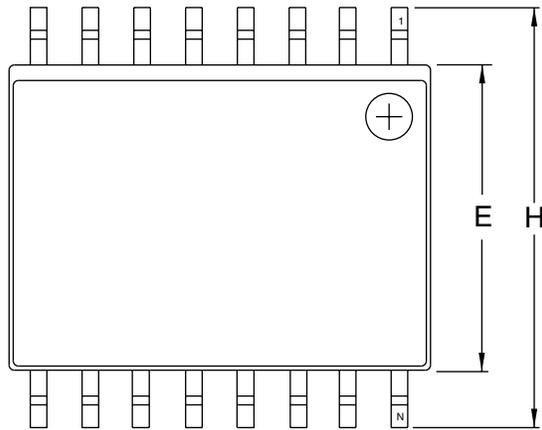
16P3

REV.

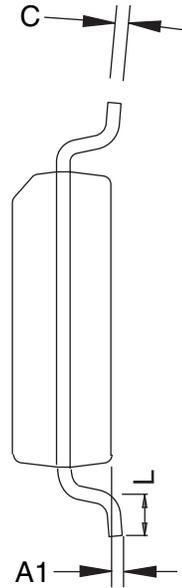
A



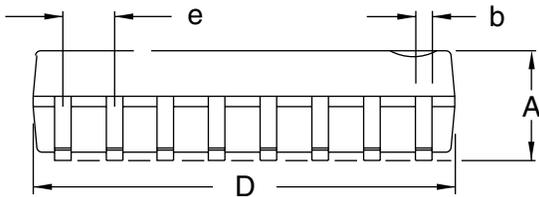
26.2 16S2 – SOIC



Top View



End View



Side View

COMMON DIMENSIONS
(Unit of Measure = inches)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.0926		0.1043	
A1	0.0040		0.0118	
b	0.0130		0.0200	5
C	0.0091		0.0125	
D	0.3977		0.4133	2
E	0.2914		0.2992	3
H	0.3940		0.4190	
L	0.0160		0.050	4
e	0.050 BSC			

- Notes: 1. This drawing is for general information only; refer to JEDEC Drawing MS-013, Variation AA for additional information.
 2. Dimension "D" does not include mold Flash, protrusions or gate burrs. Mold Flash, protrusions and gate burrs shall not exceed 0.15 mm (0.006") per side.
 3. Dimension "E" does not include inter-lead Flash or protrusion. Inter-lead Flash and protrusions shall not exceed 0.25 mm (0.010") per side.
 4. "L" is the length of the terminal for soldering to a substrate.
 5. The lead width "B", as measured 0.36 mm (0.014") or greater above the seating plane, shall not exceed a maximum value of 0.61 mm (0.024") per side.

1/9/02



2325 Orchard Parkway
San Jose, CA 95131

TITLE

16S2, 16-lead, 0.300" Wide Body, Plastic Gull Wing Small Outline Package (SOIC)

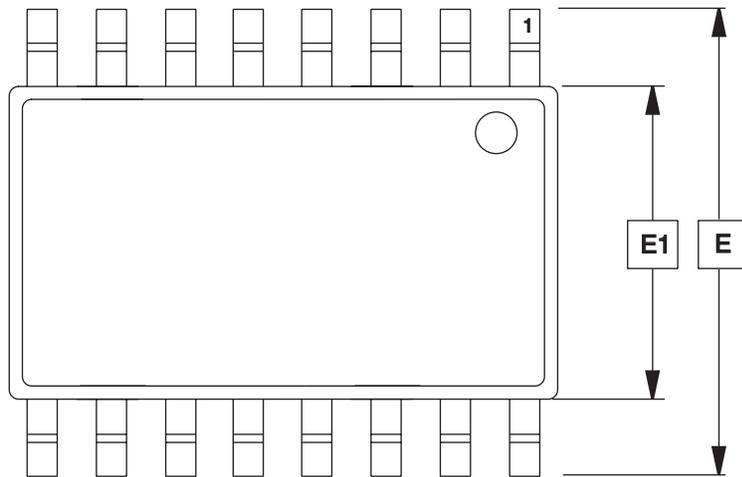
DRAWING NO.

16S2

REV.

A

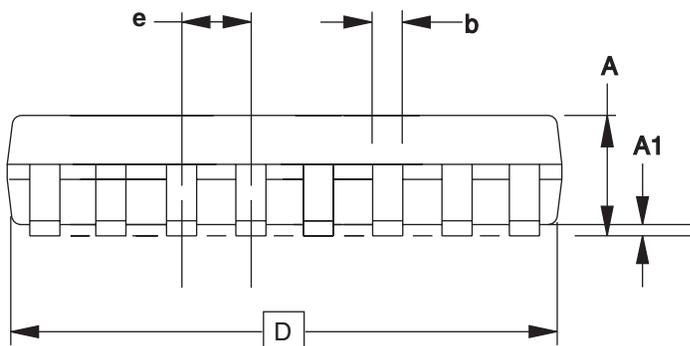
26.3 16X – TSSOP



Top View



End View



Side View

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
b	0.19	–	0.30	
C	0.09	–	0.20	
D	4.90	5.00	5.10	
E	6.40 BSC			
e	0.65 TYP			
L	0.45	0.60	0.75	
ø	0°	–	8°	

Note: These drawings are for general information only. Refer to JEDEC Drawing MO-153-AB.

11/11/05



2325 Orchard Parkway
San Jose, CA 95131

TITLE
16X, 16-lead, 4.4 mm Body Width, Plastic Thin Shrink
Small Outline Package (TSSOP)

DRAWING NO.
16X

REV.
A





27. Revision History

Revision No.	History
Revision A – June 2006	<ul style="list-style-type: none"><li data-bbox="768 342 964 365">• Initial Release

Table of Contents

1.	<i>Description</i>	1
2.	<i>Pin Configuration</i>	2
	2.1 AT89LP216: 16-lead PDIP/SOIC/TSSOP	2
3.	<i>Pin Description</i>	3
4.	<i>Block Diagram</i>	4
5.	<i>Comparison to Standard 8051</i>	4
	5.1 System Clock	4
	5.2 Instruction Execution with Single-cycle Fetch	4
	5.3 Interrupt Handling	5
	5.4 Timer/Counters	5
	5.5 Serial Port	5
	5.6 Watchdog Timer	5
	5.7 I/O Ports	5
	5.8 Reset	6
6.	<i>Memory Organization</i>	6
	6.1 Program Memory	6
	6.2 Data Memory	7
7.	<i>Special Function Registers</i>	8
8.	<i>Enhanced CPU</i>	9
	8.1 Restrictions on Certain Instructions	10
9.	<i>System Clock</i>	11
	9.1 Crystal Oscillator	11
	9.2 External Clock Source	11
	9.3 Internal RC Oscillator	11
	9.4 System Clock Out	11
10.	<i>Reset</i>	12
	10.1 Power-on Reset	12
	10.2 Brown-out Reset	14
	10.3 External Reset	14
	10.4 Watchdog Reset	15
	10.5 Software Reset	15
11.	<i>Power Saving Modes</i>	15
	11.1 Idle Mode	15
	11.2 Power-down Mode	15

Table of Contents (Continued)

12. Interrupts	18
12.1 Interrupt Response Time	19
13. I/O Ports	22
13.1 Port Configuration	22
13.2 Port 1 Analog Functions	25
13.3 Port Read-Modify-Write	25
13.4 Port Alternate Functions	26
14. Enhanced Timer/Counters	27
14.1 Mode 0 – Variable Width Timer/Counter	27
14.2 Mode 1 – 16-bit Auto-Reload Timer/Counter	28
14.3 Mode 2 – 8-bit Auto-Reload Timer/Counter	29
14.4 Mode 3 – 8-bit Split Timer	29
14.5 Pulse Width Modulation	32
15. External Interrupts	36
16. General-purpose Interrupts	36
17. Serial Interface	38
17.1 Multiprocessor Communications	38
17.2 Baud Rates	40
17.3 More About Mode 0	41
17.4 More About Mode 1	43
17.5 More About Modes 2 and 3	45
17.6 Framing Error Detection	48
17.7 Automatic Address Recognition	48
18. Serial Peripheral Interface	49
19. Analog Comparator	55
19.1 Comparator Interrupt with Debouncing	55
20. Programmable Watchdog Timer	57
20.1 Software Reset	58
21. Instruction Set Summary	59
22. On-Chip Debug System	63
22.1 Physical Interface	63
22.2 Software Breakpoints	64
22.3 Limitations of On-Chip Debug	64

Table of Contents (Continued)

23. Programming the Flash Memory	65
23.1 Physical Interface	65
23.2 Memory Organization	66
23.3 Command Format	67
23.4 Status Register	70
23.5 DATA Polling	70
23.6 Flash Security	70
23.7 User Configuration Fuses	71
23.8 Programming Interface Timing	72
24. Electrical Characteristics	76
24.1 Absolute Maximum Ratings*	76
24.2 DC Characteristics	76
24.3 Serial Peripheral Interface Timing	77
24.4 External Clock Drive	79
24.5 Serial Port Timing: Shift Register Mode	80
24.6 Test Conditions	80
25. Ordering Information	82
25.1 Standard Package	82
25.2 Green Package Option (Pb/Halide-free)	82
26. Packaging Information	83
26.1 16P3 – PDIP	83
26.2 16S2 – SOIC	84
26.3 16X – TSSOP	85
27. Revision History	86