

## 特性

- 与 MCS<sup>®</sup>51 系列相兼容的 8 位微控制器
- 增强型 8051 体系结构
  - 单时钟周期字节取指能力
  - 20 MHz 时钟频率下吞吐量达 20 MIPS
  - 全静态工作：0 Hz 到 20 MHz
  - 片上 2 周期硬件乘法器
  - 128 x 8 内部 RAM
  - 4 个中断优先级
- 非易失性程序存储器
  - 2KB 系统内可编程 (ISP) Flash 存储器
  - 寿命：可重复擦写最少 10,000 次
  - 数据保持能力：最少 10 年
  - 程序下载串行接口
  - 32 字节快速页面编程模式
  - 64 字节用户签名数组区
  - 2 级存储锁保证软件安全
- 外围特性
  - 2 个 16-bit 增强型定时器 / 计数器
  - 2 个 8-bit PWM 输出端 (仅 AT89LP213)
  - 带自动地址识别和帧错误检测的增强型 UART (仅 AT89LP214)
  - 带双缓冲器可收发增强型主 / 从 SPI
  - 可软件复位的可编程看门狗定时器
  - 带可选中断和反跳的模拟比较器
  - 8 个通用中断引脚
- 特殊微控制器特性
  - 双线 (TW) 片上调试接口
  - 掉电检测以及用断电标志上电复位
  - 内部 RC 振荡器
  - 低功耗省电和掉电模式
  - 从掉电模式的中断恢复功能
- I/O 和封装
  - 共 12 个可编程 I/O 线
  - 可配置 I/O：准双向，输入，推挽式输出，开漏共四种模式
  - 5V 耐压能力的 I/O
  - 14 针 TSSOP 或 PDIP
- 工作条件
  - 2.4V 到 5.5V V<sub>CC</sub> 电压范围
  - -40°C 到 85°C 温度范围

## 1. 描述

AT89LP213/214 是低功耗的高性能的 CMOS 8 位微控制器，带有 2K 字节的系统内可编程 Flash 存储器。该设备的生产使用了 Atmel 的高密度非易失性存储器技术并且与工业标准的 MCS-51 指令集相兼容。AT89LP213/214 的构建基于增强型的每时钟周期可取一字节指令的 CPU 核心。在经典的 8051 体系结构中，每取一字节需要 6 个时钟周期，这使得执行指令需要 12, 24 或 48 个时钟周期。在 AT89LP213/214 CPU 中，执行指令只需要 1 到 4 个时钟周期，因而吞吐量比标准的 8051 大 6 到 12 倍。百分之七十的指令的执行只需要与其字节数相同的时钟周期。



带 2K 字节 Flash  
的 8-bit 微控制器

AT89LP213

AT89LP214

Preliminary

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。

其余的指令的执行只需要比其字节数多一个的时钟周期。增强型的 CPU 核心有 20 MIPS 的吞吐能力，而经典的 8051 CPU 在同样的工作电流下只能达到 4 MIPS。另一方面，保持与经典的 8051 一样的吞吐量时，这个新的 CPU 核心工作在低得多的速度下从而极大地降低了功耗。

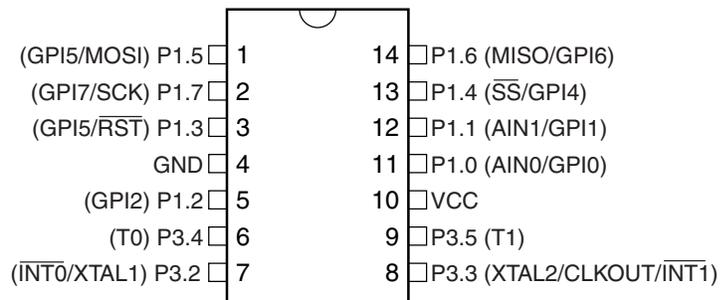
AT89LP213/214 提供下列标准特性：2K 字节系统内可编程 Flash 存储器，128 字节 RAM，12 条 I/O 线，2 个 16 位定时器 / 计数器，2 个 PWM 输出端 (仅 AT89LP213)，一个可编程看门狗定时器，一个全双工串口 (仅 AT89LP214)，一个串行外围接口，一个内部 RC 振荡器，片上晶振，一个 4 优先级的 6 向量中断系统。

AT89LP213/214 的两个 16 位定时器 / 计数器有两个新的模式。模式 0 可以配置为 9 到 16 位可变的定时器 / 计数器，模式 1 可以配置为 16 位自动重载的定时器 / 计数器。此外，AT89LP213 的两个定时器 / 计数器还可独立地驱动一个脉冲宽度调制输出端。

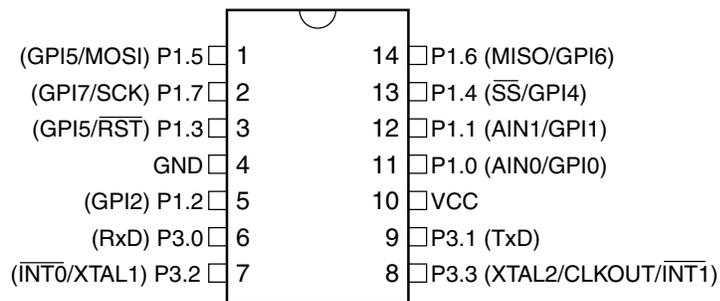
AT89LP213/214 的 I/O 口可独立地配置为四个工作模式下的一种。在准双向模式下，端口工作方式与经典 8051 相同。在输入模式下，端口是三态的。推挽式输出模式提供完全的 CMOS 驱动，开漏模式提供一个下拉电阻。此外，Port 1 的 8 个引脚都可使用通用中断接口配置为中断源。AT89LP213/214 的 I/O 的耐压值比设备自身的供电电源要高，可达 5.5V。当设备供电电压为 2.4V 而所有的 I/O 口接收为 5.5V 信号时，所有 I/O 总的回流电流少于 100  $\mu$ A。

## 2. 引脚配置

### 2.1 AT89LP213: 14 针 TSSOP/PDIP



### 2.2 AT89LP214: 14 针 TSSOP/PDIP



## 3. 引脚描述

**Table 3-1. AT89LP213 引脚描述**

引脚	符号	类型	描述
1	P1.5	I/O I/O 	<b>P1.5:</b> 用户可配置 I/O, Port 1 bit 5. <b>MOSI:</b> SPI 主 - 输出 / 从 - 输入。当配置为主控端时, 为输出引脚。当配置为从控端时, 为输入引脚。 <b>GPI5:</b> 通用中断输入端 5.
2	P1.7	I/O I/O 	<b>P1.7:</b> 用户可配置 I/O 口, Port 1 bit 7. <b>SCK:</b> SPI 时钟。当配置为主控端时, 为输出引脚。当配置为从控端时, 为输入引脚。 <b>GPI7:</b> 通用中断输入端 7.
3	P1.3	I/O     	<b>P1.3:</b> 用户可配置 I/O 口, Port 1 bit 3 (若禁止复位保险). <b>RST:</b> 外部 <b>Active-Low</b> 复位输入端 (若使能复位保险, See “外部复位” on page 15). <b>GPI3:</b> 通用中断输入端 3. <b>DCL:</b> 当使能 OCD 时作为片上调试接口的串行时钟输入。
4	GND		地
5	P1.2	I/O 	<b>P1.2:</b> 用户可配置 I/O 口, Port 1 bit 2. <b>GPI2:</b> 通用中断输入端 2.
6	P3.4	I/O I/O	<b>P3.4:</b> 用户可配置 I/O 口, Port 3 bit 4. <b>T0:</b> 定时器 / 计数器 0 外部输入或 PWM 输出。
7	P3.2	I/O   I/O	<b>P3.2:</b> 用户可配置 I/O 口, Port 3 bit 2. <b>XTAL1:</b> 输入到反向晶振放大器和内部时钟发生电路。若采用内部 RC 振荡器作为时钟源时, 它可用作一个端口引脚。 <b>DDA:</b> 当使能 OCD 并采用内部 RC 振荡器作为时钟源时, 作为片上调试接口的串行数据输入 / 输出。
8	P3.3	I/O O O I/O	<b>P3.3:</b> 用户可配置 I/O 口, Port 3 bit 3. <b>XTAL2:</b> 从反向晶振放大器输出。若采用内部 RC 振荡器作为时钟源时, 它可用作一个端口引脚。 <b>CLKOUT:</b> 若采用内部 RC 振荡器作为时钟源时, 可用于输出内部时钟的二分频。 <b>DDA:</b> 当使能 OCD 并采用外部时钟作为时钟源时, 作为片上调试接口的串行数据输入 / 输出。
9	P3.5	I/O I/O	<b>P3.5:</b> 用户可配置 I/O 口, Port 3 bit 5. <b>T1:</b> 定时器 / 计数器 1 外部输入或 PWM 输出。
10	VDD		电源电压
11	P1.0	I/O   	<b>P1.0:</b> 用户可配置 I/O 口, Port 1 bit 0. <b>AIN0:</b> 正极输入模拟比较器。 <b>GPI0:</b> 通用中断输入端 0.
12	P1.1	I/O   	<b>P1.1:</b> 用户可配置 I/O 口, Port 1 bit 1. <b>AIN1:</b> 负极输入模拟比较器。 <b>GPI1:</b> 通用中断输入端 1
13	P1.4	I/O   	<b>P1.4:</b> 用户可配置 I/O 口, Port 1 bit 4. <b>SS:</b> SPI 从控选择输入引脚。 <b>GPI4:</b> 通用中断输入端 4.
14	P1.6	I/O I/O 	<b>P1.6:</b> 用户可配置 I/O 口, Port 1 bit 6. <b>MISO:</b> SPI 主控 - 输入 / 从控 - 输出。当配置为主控端时, 为输入引脚。当配置为从控端时, 为输出引脚。 <b>GPI6:</b> 通用中断输入端 6.

**Table 3-2. AT89LP214 引脚描述**

引脚	符号	类型	描述
1	P1.5	I/O I/O I	<b>P1.5:</b> 用户可配置 I/O 口, Port 1 bit 5. <b>MOSI:</b> SPI 主 - 输出 / 从 - 输入。当配置为主控端时, 为输出引脚。当配置为从控端时, 为输入引脚。 <b>GPI5:</b> 通用中断输入端 5.
2	P1.7	I/O I/O I	<b>P1.7:</b> 用户可配置 I/O 口, Port 1 bit 7. <b>SCK:</b> SPI 时钟。当配置为主控端时, 为输出引脚。当配置为从控端时, 为输入引脚。 <b>GPI7:</b> 通用中断输入端 7.
3	P1.3	I/O I I I	<b>P1.3:</b> 用户可配置 I/O 口, Port 1 bit 3 (若禁止复位保险)。 <b>RST:</b> 外部 <b>Active-Low</b> 复位输入端 (若使能复位保险, See “外部复位” on page 15). <b>GPI3:</b> 通用中断输入端 3. <b>DCL:</b> 片上调试接口的串行时钟输入。
4	GND	I	地
5	P1.2	I/O I	<b>P1.2:</b> 用户可配置 I/O 口, Port 1 bit 2. <b>GPI2:</b> 通用中断输入端 2.
6	P3.0	I/O I	<b>P3.0:</b> 用户可配置 I/O 口, Port 3 bit 0. <b>RXD:</b> 串行端口接收端输入。
7	P3.2	I/O I I/O	<b>P3.2:</b> 用户可配置 I/O 口, Port 3 bit 2. <b>XTAL1:</b> I 输入到反向晶振放大器和内部时钟发生电路。若采用内部 RC 振荡器作为时钟源时, 它可用作一个端口引脚。 <b>DDA:</b> 当使能 OCD 并采用内部 RC 振荡器作为时钟源时, 作为片上调试接口的串行数据输入 / 输出。
8	P3.3	I/O O O I/O	<b>P3.3:</b> 用户可配置 I/O 口, Port 3 bit 3. <b>XTAL2:</b> 从反向晶振放大器输出。若采用内部 RC 振荡器作为时钟源时, 它可用作一个端口引脚。 <b>CLKOUT:</b> 若采用内部 RC 振荡器作为时钟源时, 可用于输出内部时钟的二分频。 <b>DDA:</b> 当使能 OCD 并采用外部时钟作为时钟源时, 作为片上调试接口的串行数据输入 / 输出。
9	P3.1	I/O O	<b>P3.1:</b> 用户可配置 I/O 口, Port 3 bit 1. <b>TXD:</b> 串口发送端输出。
10	VDD	I	电源电压
11	P1.0	I/O I I	<b>P1.0:</b> 用户可配置 I/O 口, Port 1 bit 0. <b>AIN0:</b> 正极输入模拟比较器。 <b>GPI0:</b> 通用中断输入端 0.
12	P1.1	I/O I I	<b>P1.1:</b> 用户可配置 I/O 口, Port 1 bit 1. <b>AIN1:</b> 负极输入模拟比较器。 <b>GPI1:</b> 通用中断输入端 1
13	P1.4	I/O I I	<b>P1.4:</b> 用户可配置 I/O 口, Port 1 bit 4. <b>SS:</b> SPI 从控选择输入引脚。 <b>GPI4:</b> 通用中断输入端 4.
14	P1.6	I/O I/O I	<b>P1.6:</b> 用户可配置 I/O 口, Port 1 bit 6. <b>MISO:</b> SPI 主控 - 输入 / 从控 - 输出。当配置为主控端时, 为输入引脚。当配置为从控端时, 为输出引脚。 <b>GPI6:</b> 通用中断输入端 6.

## 4. 方框图

Figure 4-1. AT89LP213 模块框图

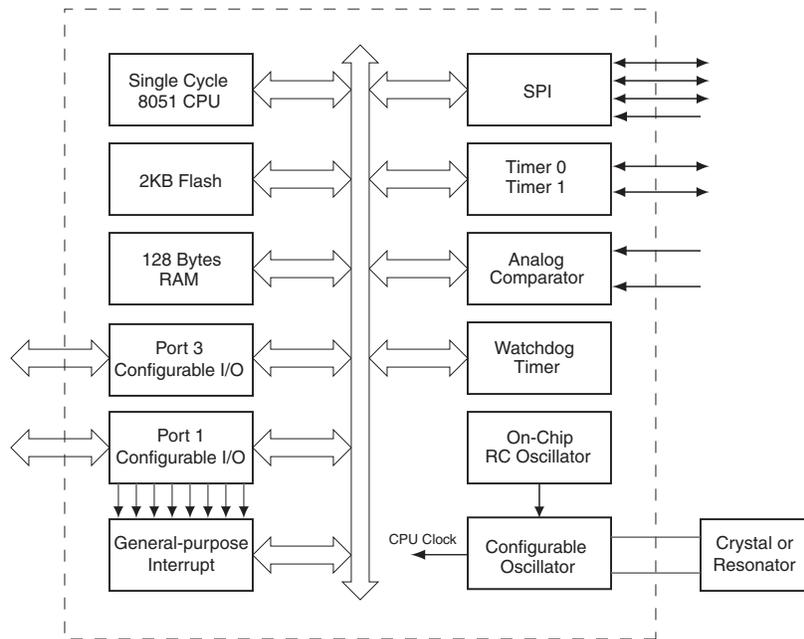
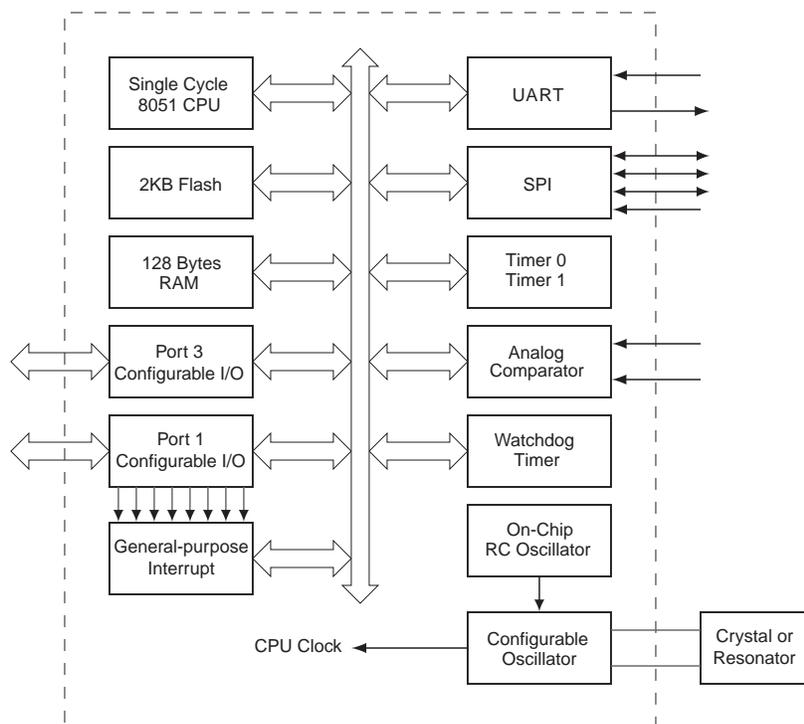


Figure 4-2. AT89LP214 模块框图



## 5. 与标准 8051 比较

AT89LP213/214 是完全与 MCS-51 指令集兼容的带增强特性的控制器系列的一部分。此外，大部分 SFR 地址，位分配和引脚转换功能与 Atmel 现有的标准 8051 系列产品相一致。不过，由于该微控制器有更高的性能，一些系统级行为则与象 AT89S52 或 AT89S2051 这样的 Atmel 的标准 8051 产品有所不同。与标准 8051 的这些不同之处罗列如下：

### 5.1 系统时钟

CPU 时钟频率与外部 XTAL1 频率相等。振荡器不再作二分频用于提供内部时钟，同时 x2 模式也不再支持。

### 5.2 指令执行时每时钟周期取一字节指令

CPU 每一个时钟周期从内存取一字节代码，而不再是 6 个时钟周期。这极大地增加了 CPU 的吞吐量。于是，CPU 执行指令不再需要 12 到 48 个时钟周期。每条指令的执行仅在 1 到 4 时钟周期内完成。详细见 [See “指令机摘要” on page 57](#) .

### 5.3 中断处理

中断控制器在每个指令的最后的时钟周期轮询中断标志。为使在最后的指令周期响应某个中断请求，其中断标志需要在该指令的最后的时钟周期到来前一直锁存为有效，如果该指令为单时钟周期执行的指令则需要在前一指令的最后一个时钟周期内一直保持有效。

外部中断引脚  $\overline{INT0}$  和  $\overline{INT1}$  每个时钟周期采样，而非每 12 个时钟周期采样一次。有了更端的指令时序和更快的中断响应，使得对外部事件的中断的最快响应速率更高。

### 5.4 定时器 / 计数器

缺省情况下，定时器 / 计数器的计数频率为一个时钟周期。相比之下，标准 8051 是每 12 个时钟周期计数一次。一个通用预定器可用于为所有的定时器划分时间基并降低计数速率。CLKREG SFR 中的 TPS 位控制预定器 ([Table 9-2 on page 13](#)). 设置 TPS = 1011B 将使定时器每 12 时钟周期计数一次。

外部定时器 / 计数器引脚 T0 和 T1 在每个时钟周期采样一次，而非 12 个时钟周期。这增加了计数器模块的最大工作速率。

### 5.5 串口

UART 在模式 0 下的波特率是时钟频率的 1/2，在标准 8051 中则是时钟频率的 1/12；输出数据仅在串行时钟的上升沿有效。还应该注意，使用定时器 1 在模式 1 或模式 3 下发生波特率时，定时器以时钟频率计数而非时钟频率的 1/12。当 AT89LP214 与标准 8051 工作在同一频率时，为保持 AT89LP214 具有相同的波特率，则延迟时间必须为 12 倍时钟周期。定时器 1 的模式 1 支持 16 位自动重载，可以用于较长的时间延迟以产生较低的波特率。

## 5.6 看门狗定时器

AT89LP213/214的看门狗定时器的计数速率与时钟周期一致。相比之下，标准8051则每12个时钟周期计数一次。一个通用预定器可用于为所有的定时器分频时间基并降低计数速率。

## 5.7 I/O 端口

AT89LP213/214的I/O端口可配置为四种模式。缺省情况下，掉电或复位时I/O端口回到只读(三态)模式。在标准8051中，所有端口在掉电或复位时仅下拉至低电平。为使能类似8051的端口，端口必须设置为准双向模式，这通过清零P1M0和P3M0 SFR来实现。用户还可以通过禁止Tristate-Port User Fuse(三态用户保险)来将端口配置为准双向模式起始方式。当禁止该保险时，P1M0和P3M0将复位到00h而非FFh，并且端口会上拉到高电位。

## 5.8 复位

AT89LP213/214的 $\overline{\text{RST}}$ 引脚是低电位触发的，而在标准8051中，则是高电位复位。此外， $\overline{\text{RST}}$ 引脚每时钟周期采样一次，并且必须最少保持低电位2个时钟周期，而非24个时钟周期，这样才能识别为有效的复位信号。

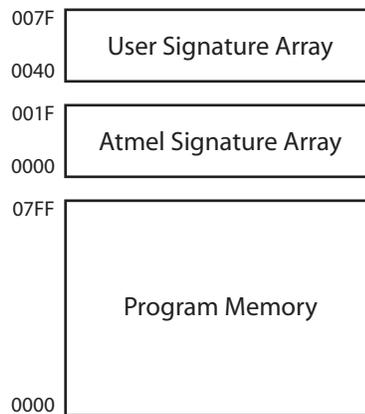
## 6. 存储器结构

AT89LP213/214使用哈佛体系结构，程序和数据存储器的地址线分离。程序存储器使用常规的线性地址空间，支持达64K字节的可直接寻址的应用代码。数据存储器有128字节的内部RAM以及128字节的特殊功能寄存器I/O空间。AT89LP213/214不支持外部数据存储器或外部程序存储器。

### 6.1 程序存储器

AT89LP213/214包含2K字节的片上系统内可编程Flash存储器用于保存程序。Flash存储器的使用寿命至少为10,000次重复擦写，并且数据保持能力可达10年。复位和中断向量放在程序存储器的开头的59字节(参考Table 13-2 on page 23)。常量表放在整个2K程序存储器地址空间，可被MOVC指令访问。AT89LP213/214不支持外部程序存储器。

Figure 6-1. 程序存储器映射



AT89LP213/214 的程序存储器的一个映射见 Figure 6-1。除了 2K 从 0000h 到 07FFh 的代码空间外，AT89LP213/214 还支持一个 64 字节的用户签名数组和一个 32 字节的 Atmel 签名数组，它们可以只读模式被 CPU 访问。为从签名数组中读取信息，必须设置 AUXR1 的 SIGEN 位。当 SIGEN 是 1 时，MOVC A,@A+DPTR 可访问签名数组。用户签名数组映射到地址 0040h 到 007Fh，Atmel 签名数组映射到地址 0000h~001Fh。在使用 MOVC 指令访问代码存储区时必须将 SIGEN 清零。

Atmel 签名数组在出厂时用设备 ID 进行初始化。用户签名数组则可用作用户 ID 或者常参数据保存。保存在签名数组中的数据是不安全的。安全位可禁止对该数组的写操作；但是，读取永远是被允许的。

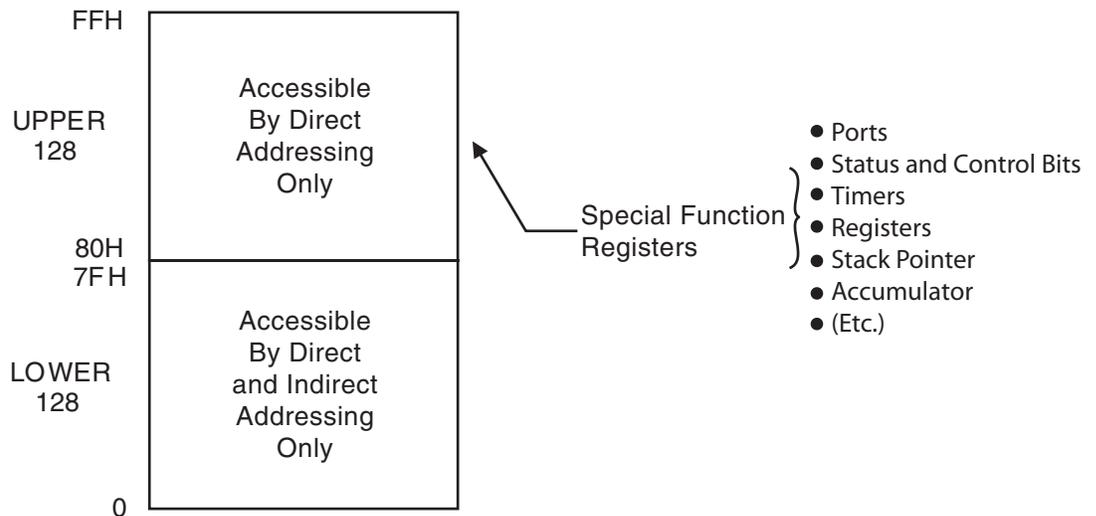
**Table 6-1.** AUXR1 – 辅助寄存器 1

AUXR1 = A2H					复位值 = XXXX 0XXXB			
不可位寻址								
	-	-	-	-	SIGEN	-	-	-
位	7	6	5	4	3	2	1	0

## 6.2 数据存储器

AT89LP213/214 包含 128 字节的通用 SRAM 数据存储器，另外还有 128 字节的 I/O 存储区，映射到一个 8 位地址空间。128 字节的数据存储器可以通过对低 128 字节地址进行直接或间接寻址来访问。128 字节的 I/O 存储区驻留在较高的 128 字节地址空间 (Figure 6-2)。I/O 存储区只能通过直接寻址方式访问并且包含特殊功能寄存器 (SFRs)。对较高的 128 字节地址的间接访问会返回无效数据。数据存储器的最低的 32 个字节被分为 4 个组，每组有 8 个寄存器。RS0 和 RS1 位 (PSW.3 和 PSW.4) 选择使用哪个寄存器组。使用寄存器寻址的指令只访问当前指定的寄存器组。AT89LP213/214 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

有一块片上存储区域的映射叫作特殊功能寄存器 (SFR) 空间，见 [Table 7-1](#)。

要注意并没有占用所有的地址，而没有使用的地址是在芯片上不可用的，读这些地址通常会返回任意数据，而写操作会发生不可预料的结果。用户程序不应该对这些没有列出的地址进行写操作，因为它们可能用于在将来的产品中引入新的特性。

**Table 7-1.** AT89LP213/214 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 <sup>(2)</sup>	P1M1 xx00 0000			P3M0 <sup>(2)</sup>	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)	WDTCN 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. 最左边一列的所有 SFR 都是位可寻址的。
  2. 当三态端口保险打开时，复位值为 xx11 1111B，否则为 xx00 0000B。

## 8. 加强型 CPU

AT89LP213/214 使用一个加强型的 8051 CPU，它的运行速度是普通 8051 单片机的 6 到 12 倍。(或者说 X2 型 8051 单片机速度的 3 到 6 倍)。速度的增加归功于下面两条因素。首先，CPU 每个时钟周期就从代码存储器中取一条指令字节。其次，CPU 使用一个二级流水线并行的取指令和执行指令。这种基本的流水线操作理念使 CPU 的速度高达 1 MIPS 每 MHz。简单示例如图 Figure 8-1。

MCS-51 指令系统允许它的指令的长度可变。变化范围是 1 到 3 字节。在一个每字节单时钟取指令的系统中，执行每一条指令至少需要和这条指令字节数一样多的机器时钟数。AT89LP213/214 中的大多数指令遵循如下规则：每一条指令的执行时间（需要的时钟周期数）等于这条指令包含的字节数（有一些特例除外）。分支和调用需要一个额外的时钟周期以计算目标地址，并且某些复杂的指令需要多个时钟周期。对于每一条指令的细节，请参见 59 页“指令集摘要”。Figures 8-2 and 8-3 显示了一字节长和两字节长指令的情况。

Figure 8-1. 取得和执行并行指令

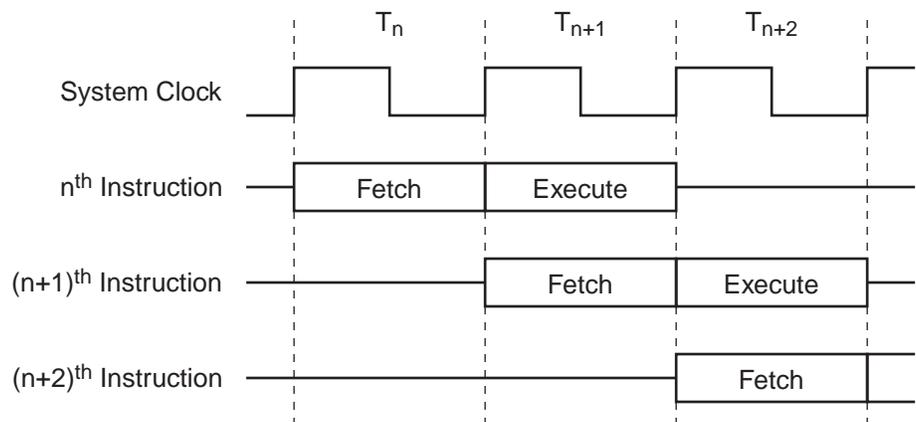
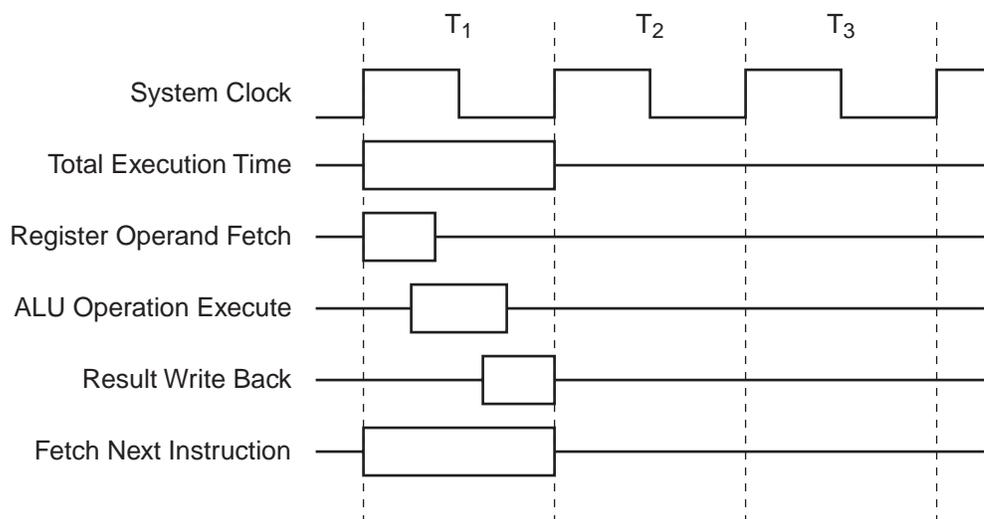
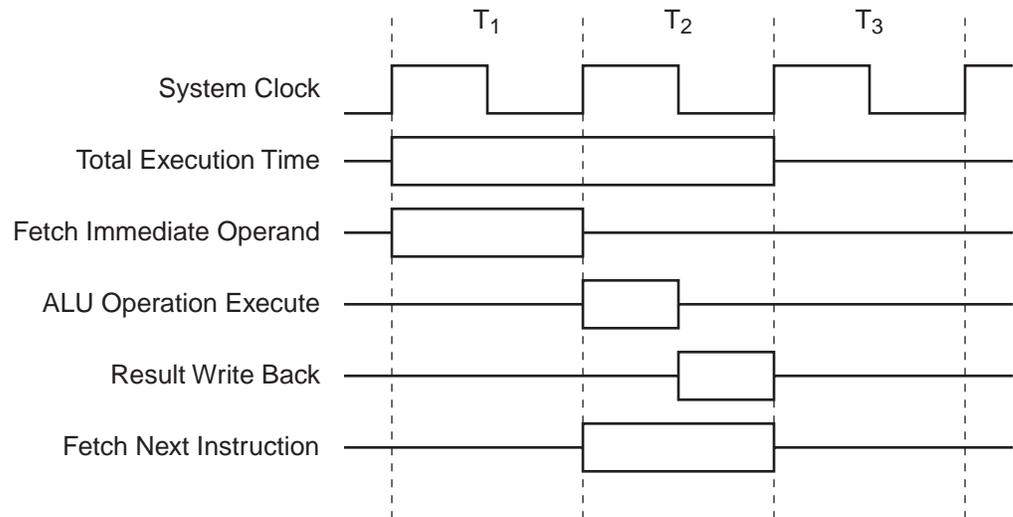


Figure 8-2. 单周期 ALU 操作 (例: INC R0)



**Figure 8-3.** 两个周期的 ALU 操作 (例如: ADD A, #data)



## 8.1 某些特殊指令的限制

AT89LP213/214 是 ATMELO 日益成长的微处理器家族的一款经济型性价比高的芯片。它包含 2K 字节的闪存程序存储器。他完全兼容于普通的 MCS-51 架构。可以使用 MCS-51 指令集编程。然而在使用某些特殊指令编程此芯片时，需要考虑某些特殊的情况。所有关系到跳转或分支的指令都需要考虑 AT89LP213/214 的闪存程序存储器的大小是 2K，所以定义地址时，一定要注意不要超过这个物理地址的极限。这就需要软件编程者多多注意。例如，LJMP 7E0H 是一条有效的指令，然而 LJMP 900H 就是一条无效的指令

### 8.1.1 分支指令

只要编程时注意定义的分支地址在程序存储器尺寸（AT89LP213/214 的地址范围从 000H 到 7FFH）范围以内，那么这些无条件分支命令 LCALL, LJMP, ACALL, AJMP, SJMP, 和 JMP @A+DPTR 就可以正确的运行。如果超出了它的物理地址的范围，将会造成无法预直的错误结果。有条件分支命令例如 CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, 和 JNZ 等，情况与上面的无条件分支命令相同。同样，如果超出了它的物理地址的范围，将会造成无法预直的错误结果。如果应用程序使用了中断，那么中断处理程序的入口地址同 8051 架构单片机的入口地址相同。

### 8.1.2 MOVX 的相关指令，数据存储器

AT89LP213/214 包含 128 字节的内部数据存储器。RAM 的存取地址如果超过 7FH 会返回无效的数据。而且，堆栈的深度限制在 128 字节，与有效的 RAM 大小相同。堆栈指针也不可以指向的地址大于 7FH。这款芯片不支持外部数据存储器存取，也不支持外部程序存储器。因此，命令 MOVX [...] 在这款芯片上不能使用。

在通常的 8051 汇编程序中，即使我们写的语句违反了上面我们所说的这些限制，指令仍然可以集合起来。所以，编程者必须知道芯片的物理特点和限制，以正确的使用命令。

## 9. 系统时钟

系统时钟直接有 3 个可选择的时钟源中的一个来产生。这 3 个时钟源分别是芯片内部的晶振，外部时钟源和一个内部的 RC 震荡器。时钟源的选择是通过 [Table 9-1](#) 中的 Clock Source User Fuses 来设定的。并没有内部的时钟分频器对系统时钟分频得到 CPU 时钟，参见“[用户配置引脚](#)” on [page 69](#)。

**Table 9-1.** 时钟源设定

Clock Source Fuse 1	Clock Source Fuse 0	选择的时钟源
0	0	晶振
0	1	保留
1	0	XTAL1 上的外部时钟
1	1	内部的 8 MHz RC 振荡器

### 9.1 晶振

当晶振被激活以后，内部的反向振荡放大器就被连接在 XTAL1 和 XTAL2 之间以连接一个外部的石英晶振或者陶瓷振荡器。当使用晶振的时候，应该使 P3.2 和 P3.3 的输入或输出无效，XTAL2 不可以在没有缓冲的情况下被用于驱动一个板级的时钟。

### 9.2 外部时钟源

选择外部时钟会使振荡放大器无效并且允许 XTAL1 直接由时钟源来驱动。XTAL2 可以不连接，被用作 P3.3 I/O 口，或者配置成为输出系统时钟的分开版本。

### 9.3 内部 RC 振荡器

AT89LP213/214 有一个内部的 RC 振荡器，频率范围在 8.0 MHz  $\pm$ 2.5%。当它被激活作为时钟源的时候，XTAL1 和 XTAL2 可分别作为 P3.2 和 P3.3。XTAL2 也可以配置成为输出系统时钟的分开版本。可以通过改变 RC Adjust Fuses 来调节振荡器的频率。(See “[用户配置引脚](#)” on [page 69](#)).

### 9.4 系统时钟输出

当 AT89LP213/214 被配置为使用外部时钟或内部 RC 振荡器的时候，XTAL2 (P3.3) 输出系统时钟的 **divided version**。可以通过设置 CLKREG 的 COE 位来激活此输出。两个 CDV 位决定了时钟的分频数。例如，当使用内部振荡器时，设定 COE = “1” and CDIV = “00” 将使 P3.3 的输出为 3.950Mhz。在使用时钟输出功能时，P3.3 必须设定为输出。

**Table 9-2.** CLKREG – 时钟控制寄存器

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

符号	Function															
TPS3 TPS2 TPS1 TPS0	预脉冲计数器。预脉冲计数器从计数器 0。计数器 1，看门狗计数器中选择一个计数器。预脉冲计数器是一个 4 位二进制的向下的计数器。当计数器计到 0 时，它会重载存储在 TPS 位中的值，从而给出分频的值。这个值的范围是从 1 到 16。在默认情况下，每一个时钟，周期计数器都会计数。(TPS = 0000B)。如果想配置计数器工作在通用的 8051 单片机情况下（即每 12 个时钟周期计数），TPS 需要设定为 1011B。															
CDV1 CDV0	时钟输出分频。决定了相对于系统时钟的始终输出频率。 <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;"><u>CDIV1</u></th> <th style="text-align: left;"><u>CDIV0</u></th> <th style="text-align: left;"><u>Clock Out Frequency</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>f/2</td> </tr> <tr> <td>0</td> <td>1</td> <td>f/4</td> </tr> <tr> <td>1</td> <td>0</td> <td>f/8</td> </tr> <tr> <td>1</td> <td>1</td> <td>f/16</td> </tr> </tbody> </table>	<u>CDIV1</u>	<u>CDIV0</u>	<u>Clock Out Frequency</u>	0	0	f/2	0	1	f/4	1	0	f/8	1	1	f/16
<u>CDIV1</u>	<u>CDIV0</u>	<u>Clock Out Frequency</u>														
0	0	f/2														
0	1	f/4														
1	0	f/8														
1	1	f/16														
COE	时钟输出激活。设定 COE 以确定在 XTAL2 (P3.3) 上输出的系统时钟的分频率。要使用此功能，必须选择内部 RC 振荡器或者外部时钟源。															

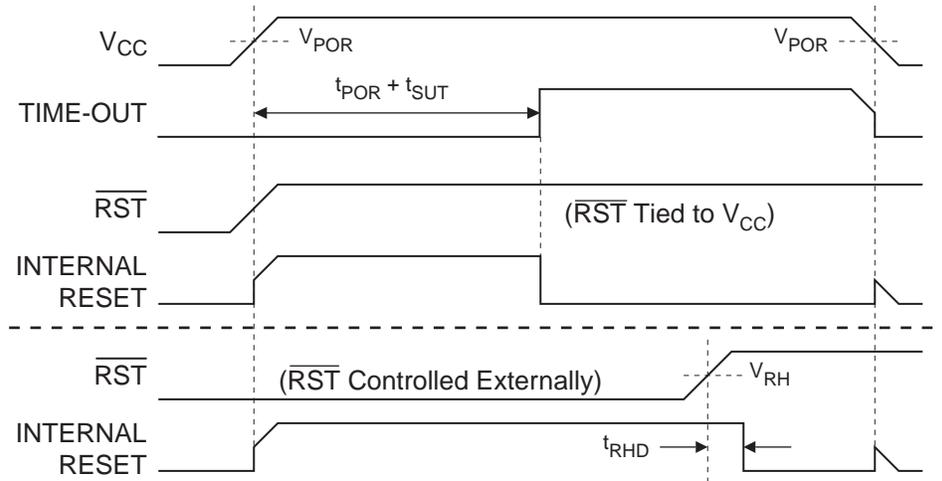
## 10. 复位

在复位时，所有的 I/O 寄存器都被设定为它们的初始值，芯片的管脚都是三态。程序从复位向量 0000H 开始执行。AT89LP213/214 有五个复位源：上电复位，掉电复位，外部复位，看门狗复位，和软件复位。

### 10.1 上电复位

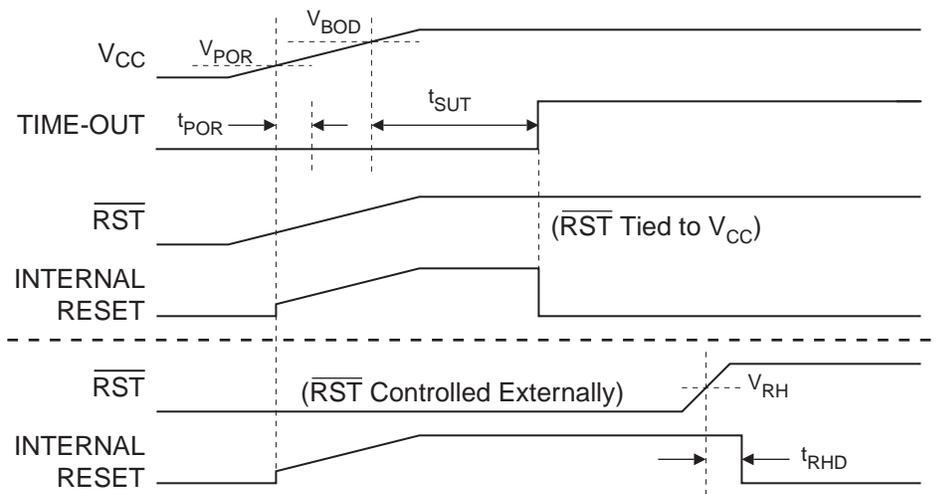
上电复位 (POR) 是由一个片上的检测电路产生的。检测级别是 1.4V。当 Vcc 低于探测级别时，产生一个上电复位。POR 电路可以被用来触发一个复位或者在没有掉电探测器的情况下检测出供电电源失败。POR 电路确保设备通过上电来复位。上电的顺序如图：[Figure 10-1 on page 14](#)。当 Vcc 达到了上电复位的阈值电压  $V_{POR}$ ，一个持续时间  $t_{POR}$  的初始化动作就开始了。初始化动作结束后，启动定时器会决定设备在 Vcc 上升沿之后要保持在 POR 多长时间。当 Vcc 低于上电复位的阈值电压时，POR 信号再一次被激活，没有任何的延迟。上电复位（例如冷复位）需要设定 PCON 中的 POF 标志。通过保持  $\overline{RST}$  脚低电平时间长过休息时间，内部产生的复位可以扩展到超过上电周期。

**Figure 10-1.** 上电复位时序 (BOD Disabled)



如果掉电探测器 (BOD) 也被激活, 那么直到 V<sub>CC</sub> 达到如图 [Figure 10-2](#) 所示的阈值电压 V<sub>BOD</sub>, 启动定时器才开始计数。然而, 如果在初始化完成之前, 就出现这种情况, 定时器必须首先等待初始化完成再开始计数。

**Figure 10-2.** 上电复位时序 (BOD Enabled)



**Note:** t<sub>POR</sub> 近似等于 92 μs ± 5%.

启动定时器的延迟依赖于时钟源 ([Table 10-1](#)), 可以通过配置启动定时引线来设置这个延时。我们设定启动延时时要注意给 V<sub>CC</sub> 和可选时钟源提供足够的时间。启动定时引线同时也控制着掉电复位后启动时间的长度, 和在内部时控模式时下电后何时唤醒。

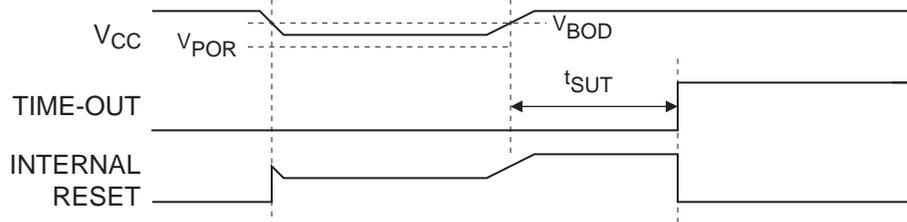
**Table 10-1.** 启动定时器设置

SUT Fuse 1	SUT Fuse 0	时钟源	$t_{SUT} (\pm 5\%)$
0	0	内部 RC/ 外部时钟	16 $\mu$ s
		晶振	1024 $\mu$ s
0	1	内部 RC/ 外部时钟	512 $\mu$ s
		晶振	2048 $\mu$ s
1	0	内部 RC/ 外部时钟	1024 $\mu$ s
		晶振	4096 $\mu$ s
1	1	内部 RC/ 外部时钟	4096 $\mu$ s
		晶振	16384 $\mu$ s

## 10.2 掉电复位

AT89LP213/214 有一个片上掉电探测电路 (BOD)，它通过同一个固定的触发电平相比较以监控  $V_{CC}$  的情况。BOD 的触发电平是 2.2V。BOD 的作用是当系统在高速工作时，如果出现  $V_{CC}$  电压错误，系统可以安全平稳的进入复位状态，而不会发生一些包括运行错误等的可能的错误。BOD 的时序如图 Figure 10-3。当  $V_{CC}$  减小到低于触发电平  $V_{BOD}$  时，立即启动内部复位。当  $V_{CC}$  增加到高于触发电平时，启动定时器在暂停周期满之后释放内部复位 (Table 10-1)。只有设置了 BOD 激活引线之后才可以激活掉电探测器。(See “用户配置引脚” on page 69).

**Figure 10-3.** 掉电探测器复位



## 10.3 外部复位

P1.3/ $\overline{RST}$  脚既可以作为一个低有效的复位输入，也可以作为一个通用的数字 I/O 口。当复位脚激活引线被设置为 1 时，P1.3 脚的外部复位输入功能就被激活了。(See “用户配置引脚” on page 69). 当清 0 时，P1.3 可以被用做一个输入或者输出脚。当配置成复位输入时，要出发内部复位，管脚必须至少保持在低电平 2 个时钟周期。

**Note:** 在上电时序中，引线选择总是无效的，因此，管脚总是执行着复位输入的功能。与此管脚相连的外部电路在上电时不可以将此管脚保持在低电平因为这将会导致芯片一直处于复位状态，除非将此管脚拉高。在上电延迟之后，这个输入脚或者执行一个外部复位输入功能或者是执行一个由引线位定义的数字输入功能。只有上电复位时会使得由复位引线位定义的选择项无效。而别的复位源不会使复位引线位无效。P1.3/ $\overline{RST}$  也可以用做系统编程 (ISP) 激活管脚。当外部复位脚被保持在低电平时，ISP 功能就被激活了。当复位脚被引线位设置无效时，可以在上电时通过将 P1.3 拉低来实现 ISP 功能。

## 10.4 看门狗复位

当看门狗计数器溢出后，它将产生一个持续 16 个时钟周期的内部复位脉冲。看门狗复位也会设置 WDTCON 寄存器的 WDTOVF 标志位。为了阻止看门狗复位，在看门狗溢出之前，必须将看门狗复位序列 1EH/E1H 写入到 WDTRST 中去。关于看门狗的操作细节请参照 [. See “ 可编程看门狗定时器 ” on page 55](#)

## 10.5 软件复位

可以通过软件写 5AH/A5H 到 WDRST 寄存器中，使 CPU 产生一个内部的 16 个时钟周期的复位脉冲。软件复位将会把 WDTCON 寄存器中的 SWRST 位置 1。关于软件复位的详细信息请参见 [. See “ 软件复位 ” on page 55](#)

## 11. 省电模式

AT89LP213/214 支持两种不同的低功耗模式：空闲模式和省电模式。这两种模式可以通过 PCON 来设定。

### 11.1 空闲模式

设定 PCON 的 IDL 位可以使芯片进入空闲模式。空闲模式下，内部 CPU 时钟停止工作。CPU 的状态保持不变。包括 RAM，堆栈指针，程序计数器，程序状态字和数字存储器。各个端口管脚的状态保持在系统进入空闲模式时候的状态。空闲模式下，外部设备仍然运转，这样，当中断产生时，它们就可以唤醒芯片。定时器，UART, SPI, 和 GPI 模块也会保持运转。在空闲模式下，比较器或者看门狗可以设定为工作状态或停止状态。任何中断或者复位都可以使系统结束空闲模式。当通过中断结束空闲模式时，中断服务程序立即被执行。之后的下一条指令 RETI 应该是使系统进入空闲模式。

### 11.2 待机模式

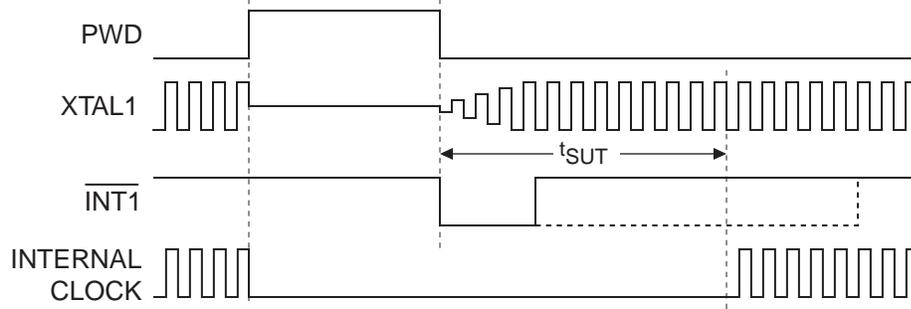
设置 PCON 中的待机位 (PD) 可以使系统进入待机模式。待机模式下振荡器停止工作，闪存掉电，这样功耗就可以降到最低。此时只有上电电路保持有效。在待机模式下，电源供给电压可以减小到保持 RAM 活动状态的电压。RAM 里面的内容会保持不变，但一旦 Vcc 降下来，我们无法保证 SFR 中的内容保持不变。可以通过外部复位，上电复位，或者某些特定的中断来结束待机状态。

#### 11.2.1 待机模式下的中断恢复

可以通过三个外部中断来结束待机模式。当 XTAL1 或 XTAL2 没有被用于晶振或外部时钟的时候 可以用来通过外部中断  $\overline{INT0}$  (P3.2) 和  $\overline{INT1}$  (P3.3) 结束系统的待机状态。为了可以通过外部中断  $\overline{INT0}$  或  $\overline{INT1}$  来唤醒系统，中断必须被激活和配置成电平敏感模式。当  $\overline{RST}$  脚无效时，通用中断 3 (GPI3) 也可以用来唤醒系统。为了外结束待机模式，GPI3 必须被激活和配置为低电平探测模式。

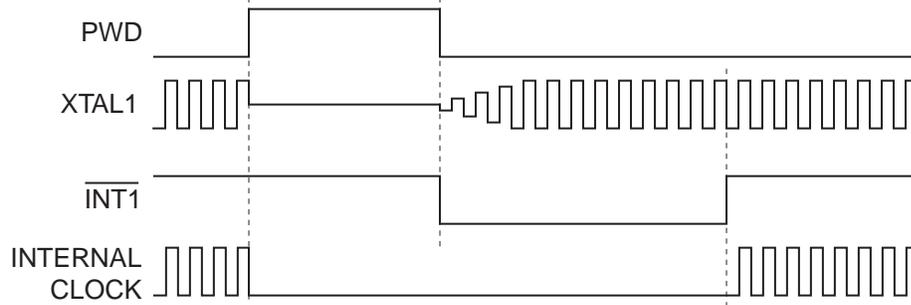
当通过中断结束待机模式时，两种不同的唤醒模式有效。当 PCON 中的 PWDEX 位是 0 时，唤醒周期由如图 [Figure 11-1](#) 所示内部时控。在中断管脚的下降沿时结束待机模式，重启晶振，内部定时器开始计数。直到定时器溢出，内部时钟才可以向 CPU 传送。溢出周期结束之后，开始调用中断服务程序。溢出周期由启动计时引线来设定。(参见 [Table 10-1 on page 15](#))。在整个溢出周期，中断引脚不需要一直保持低电平。

**Figure 11-1.** 待机模式下的中断恢复 (PWDEX = 0)



当 PWDEX = “1” 时，唤醒周期表面上看起来是由中断所控制的。在中断管脚信号的下降沿，待机模式结束并且重启振荡器。然而，直到如图 Figure 11-2 所示 的中断管脚信号的上升沿，内部时钟才开始工作。为了使选定的时钟源稳定，中断管脚的信号必须保持足够时间的低电平。中断管脚信号的上升沿之后，系统调用中断服务程序。

**Figure 11-2.** 待机模式下的中断恢复 (PWDEX = 1)



### 11.2.2 待机模式下的复位恢复

通过一个外部的复位来将系统从待机模式下唤醒和用 PWDEX = “0” 的中断差不多。在  $\overline{\text{RST}}$  信号的下降沿，结束待机状态，重启振荡器，如图 Figure 11-3 所示的内部定时器开始计数。直到这个定时器溢出，内部时钟才对 CPU 起作用。溢出周期由启动时间引线控制参见 Table 10-1 on page 15 如果在溢出之前  $\overline{\text{RST}}$  回到高电平状态，当内部时钟重启时，一个两周期的内部复位就产生了。否则，芯片将保持复位状态一直到  $\overline{\text{RST}}$  信号被拉高。

Figure 11-3. 待机模式下的复位恢复。

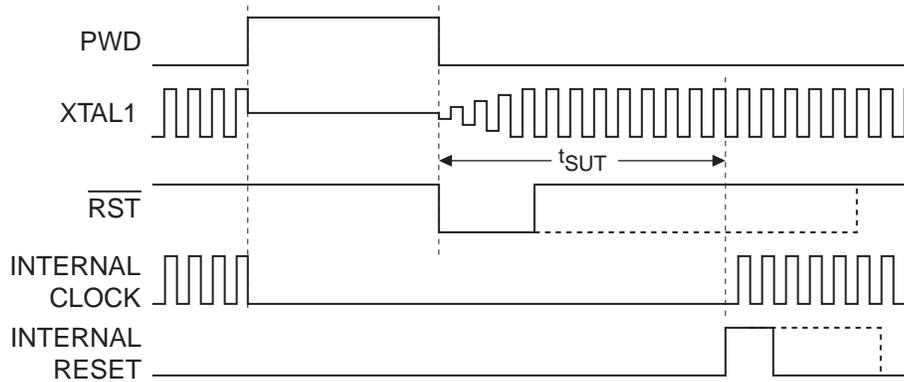


Table 11-1. PCON – 电源控制寄存器

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	双倍波特率位。在模式 1, 2, 3 下 UART 双倍波特率
SMOD0	帧错误选择。当 SMOD0 = 1, SCON.7 是 SM0。当 SMOD0 = 0, SCON.7 是 FE（此处有误？）。注意无论 SMOD0 的状态如何，一个帧错误之后都要设成 FE。
PWDEX	待机结束位。当 PWDEX = 1 时，待机模式下的唤醒由外部控制。当 PWDEX = 0 时，待机模式下的唤醒是内部时控的。
POF	关电标志。在上电时，POF 位被设置为 1（例如：冷重启）。它可以通过软件来置位或重启，而不受 RST 或 BOD 的影响。（例如热重启）。
GF1, GF0	通用标志位
PD	待机位。该位置位则系统进入待机模式。
IDL	空闲模式位。该位置位则系统进入空闲模式

## 12. 中断

AT89LP213/214 提供了 7 个中断源：两个外部中断，两个定时器中断，一个串口中断，一个通用中断，和一个模拟比较器中断。这些中断和系统复位在程序存储空间的开始处都各自有一个独立的程序向量。通过设置中断激活寄存器（IE）的一位可以使每一个中断源有效或失效。IE 寄存器也包含一个全局的中断开关位 EA，可以使所有的中断源无效。

每一个中断源（模拟比较器除外）都可以通过设置或清除中断优先级寄存器 IP 和 IPH 来独立的设定它的中断优先级，一共有四个中断优先级。模拟比较器的中断被固定在最低优先级。一个在运行的中断服务程序可以被一个比它优先级高的中断打断，但不可以被与它相同或更低的中断优先级的中断打断。最高优先级的中断不可以被任何其它的中断源打断。如果在某条指令的最后同时有两条不同中断优先级的中断请求，那么系统响应高优先级的中断。如果在某条指令的最后同时有两条相同中断优先级的中断请求，则一个内部的查询将会来决定系统响应那一条中断。

轮询序列是基于向量地址的，低向量地址的中断优先级高于高向量地址的中断。注意，轮询序列只用于解决为响应的同优先级的中断。

根据 TCON 寄存器的 IT0 和 IT1 位，外部中断  $\overline{INT0}$  和  $\overline{INT1}$  都可以是电平触发或边沿触发。实际上产生这些中断的标志是 TCON 的 IE0 和 IE1 位。当服务例行程序是矢量的，只有当中断是边沿触发时，硬件清除标志位才可以产生外部中断。如果中断是电平触发的，那么外部请求源（而不是片上硬件）控制请求标志。

定时器 0 和定时器 1 的中断由 TF0 和 TF1 产生，当各自的定时 / 计数寄存器溢出时置位（除了定时器 0 工作于方式 3）。当产生定时器中断时，如果服务例行程序是矢量的，则片上硬件清除产生中断的标志位。

串行口中断是由 SCON 的 RI 和 TI 的或逻辑加上 SPSR 的 SPIF 产生的。如果服务例行程序是矢量的，所有这些标志位不能由硬件清除。实际上，服务例行程序通常要检测 RI，TI，或 SPIF 有没有产生中断，这些位必须由软件清零。

GPIF 寄存器的八个标志位的或逻辑产生通用中断。如果服务例行程序是矢量的，所有这些标志位都不能由硬件清零。服务例行程序必须检测中断是由哪位产生的，该位必须由软件清零。如果中断是电平触发的，那么在软件清除标志位前，外部请求源必须保持中断信号。

ACSR 寄存器的 CF 位产生比较中断。如果服务例行程序是矢量的，这个标志位不能由硬件清除，而必须由软件清除。

大多数产生中断的位可以由软件置位或清零，与由硬件置位和清零的效果相同。也就是说，可以由软件产生中断和取消未响应的中断。SPI 中断标志位 SPIF 和通用中断标志位 GPIF 是两个例外。这两个标志位只能由硬件置位，软件清除。

**Table 12-1.** 中断向量地址

中断	源	向量地址
系统重启	RST or POR or BOD	0000H
外部中断 0	IE0	0003H
定时器 0 溢出	TF0	000BH
外部中断 1	IE1	0013H
定时器 1 溢出	TF1	001BH
串行口	RI or TI or SPIF	0023H
通用中断	GPIF	002BH
模拟比较器	CF	0033H

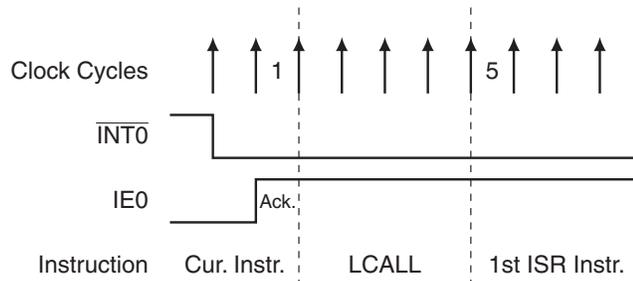
## 12.1 中断响应时间

中断标志位可以在任何时钟周期由硬件置位。在进程中指令的最后一个时钟周期，中断控制器获得标志位。如果在前述的周期中，一个标志位被置位，在轮询周期中会被发现，并且中断系统会在适当的服务例行程序中产生 LCALL，作为下一条指令。在以下情况下中断不会被响应：相同优先级或较高优先级的中断正在运行中；运行中的指令是 RETI 或者任何写入 IE，IP，或 IPH 寄存器的。这些情况中的任何一种会阻止对中断服务例行程序产生 LCALL。第二个条件确保如果在运行中

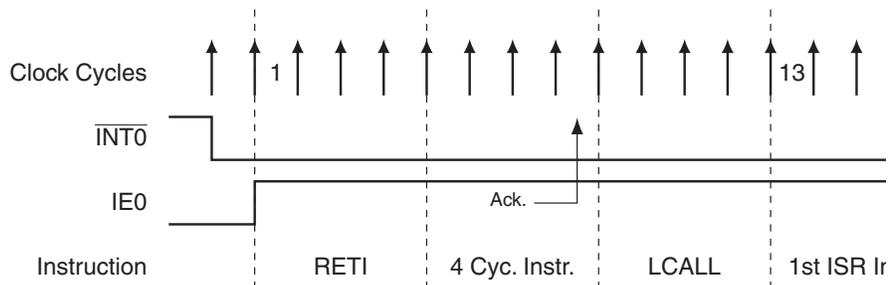
的指令是 RETI 或者任何访问 IE, IP 或 IPH 寄存器的, 那么在中断矢量化前至少还可以执行一条指令。轮询周期在每条指令的最后一个周期重复, 选中的值是前述周期中存在的值。如果因为以上原因一个激活的中断标志位没有被响应, 且当上述原因消除后中断不再是激活的, 那么这个中断将不被响应。换一句话说, 中断源一旦被激活但是没有被响应的不会被记录。每一个轮询周期是新的。

如果请求被激活且满足被响应的条件, 硬件子例行程序调用请求的服务例行程序, 使其成为下一条被执行的指令。调用程序本身需要四个周期。因此, 在激活中断请求和开始执行第一条服务例行程序的指令之间, 至少要消耗五个完整的时钟周期。如果请求被前面所列的原因阻止了, 那就需要更长的响应时间。如果同级的中断源或者更高优先级的中断源在运行中, 附加等待时间取决于其他中断的服务例行程序。如果运行中的指令不是处于最后一个时钟周期, 附加的等待时间不会超过 3 个周期, 因为最长也只有 4 个周期。如果运行中的指令是 RETI 或者访问 IE 或 IP 寄存器, 附加的等待时间不会超过 7 个周期 (结束运行中的指令最多需要 3 个周期, 加上结束下一条指令最多需要 4 个周期)。因此, 在单个中断系统中, 响应时间总是大于 5 个时钟周期而小于 13 个时钟周期。见图 12-1 和图 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 – 中断允许寄存器

IE = A8H		复位值 = 0000 0000B						
位可寻址								
	EA	EC	EGP	ES	ET1	EX1	ET0	EX0
Bit	7	6	5	4	3	2	1	0

符号	功能
EA	全部有效 / 失效 . 当 EA = 0 时, 所有的中断失效 . 当 EA = 1 时, 每个中断源通过设置 / 清除自己的使能位使中断有效 / 失效。
EC	比较器中断有效
EGP	通用中断有效
ES	串行口中断有效
ET1	定时器 1 中断有效
EX1	外部中断 1 有效
ET0	定时器 0 中断有效
EX0	外部中断 0 有效

**Table 12-3.** IP – 中断优先级寄存器

IP = B8H		复位值 = X000 0000B						
位可寻址								
	—	—	PGP	PS	PT1	PX1	PT0	PX0
位	7	6	5	4	3	2	1	0

符号	功能
PGP	通用中断优先级低
PS	串行口中断优先级低
PT1	定时器 1 中断优先级低
PX1	外部中断 1 优先级低
PT0	定时器 0 中断优先级低
PX0	外部中断 0 优先级低

**Table 12-4.** IPH – 中断优先级高寄存器

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

符号	功能
PGH	通用中断优先级高
PSH	串行口中断优先级高
PT1H	定时器 1 中断优先级高
PX1H	外部中断 1 优先级高
PT0H	定时器 0 中断优先级高
PX0H	外部中断 0 优先级高

## 13. I/O 端口

AT89LP213/214 可以设置9到12个I/O引脚。有效的I/O引脚的精确数目取决于时钟和复位选择，如表 13 - 1 所示。所有端口引脚 5V 的容限，也就是说可以被上拉或驱动到 5.5V，即使工作在较低的  $V_{CC}$ ，如 3V

**Table 13-1.** I/O 引脚配置

时钟源	复位选择	I/O 引脚数
外部晶体或振荡器	外部 $\overline{RST}$ 引脚	9
	没有外部复位	10
外部时钟	外部 $\overline{RST}$ 引脚	10
	没有外部复位	11
内部 RC 振荡器	外部 $\overline{RST}$ 引脚	11
	没有外部复位	12

**Table 13-2.** 端口 x, y 位配置模式

PxM0.y	PxM1.y	端口模式
0	0	准双向的
0	1	推挽式输出
1	0	只输入 (高阻抗)
1	1	开消耗输出

### 13.0.1 准双向输出

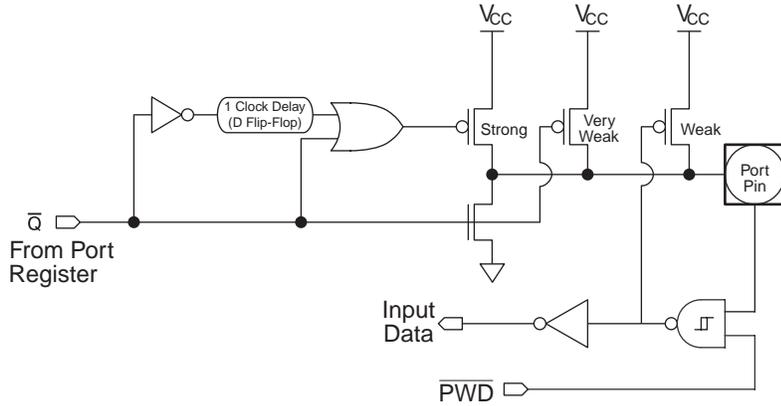
准双向输出模式的端口引脚功能与标准 8051 端口引脚相似。不需要重新配置端口，准双向的端口就可以同时用作输入和输出。这是可行的，因为当端口输出一个逻辑高电平时，它被弱驱动，允许外部设备把引脚电平拉低。当引脚低电平驱动时，强驱动使大电流减小。为了不同的目的，准双向输出有 3 个上拉电阻。

这 3 个上拉电阻中的一个叫做“极弱”上拉电阻，只要引脚的端口门限包含逻辑“1”，这个电阻就是工作的。当引脚悬空时，这个极弱的上拉电阻产生极小的电流使引脚电平拉高。

第二个上拉电阻叫做“弱”上拉电阻，当引脚的端口门限包含逻辑“1”且引本身是逻辑“1”电平时，这个电阻是工作的。这个上拉电阻提供准双向引脚输出为“1”的基本源电流。如果外部设备把该引脚下拉为低电平，则这个弱上拉电阻关闭，只有极弱上拉电阻维持工作。为了在这些条件下使引脚电平下拉，外部设备不得不减小足够的电流抑制弱上拉电阻，并使端口引脚电平位于输入门限电平以下。

第三个上拉电阻是“强”上拉电阻。这个上拉电阻的作用是，当端口门限由逻辑“0”转换为逻辑“1”时，可以加快准双向端口引脚的低电平到高电平的转换速度。当这种情况发生时，强上拉电阻工作两个 CPU 周期，使端口电平迅速上拉。准双向端口配置如 Figure 13-1 所示。在下电时，P1.3, P3.2 和 P3.3 的输入电流不是无效的 (见 Figure 13-3)。

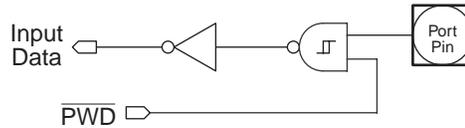
**Figure 13-1.** 准双向输出



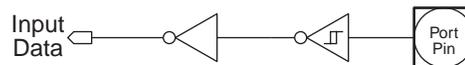
### 13.0.2 只输入模式

只输入端口配置如 Figure 13-2 所示。输出驱动是三态的。输入包括斯密特触发输入以提高对输入噪声的抑制。在下电时，P1.3, P3.2 和 P3.3 的输入电路不是无效的 (见 Figure 13-3)。输入引脚可以被安全的上拉到 5.5V，即使工作在较低的  $V_{CC}$  电平；然而，斯密特触发器的输入门限是由  $V_{CC}$  电平设置的，这一点必须考虑。

**Figure 13-2.** 只输入



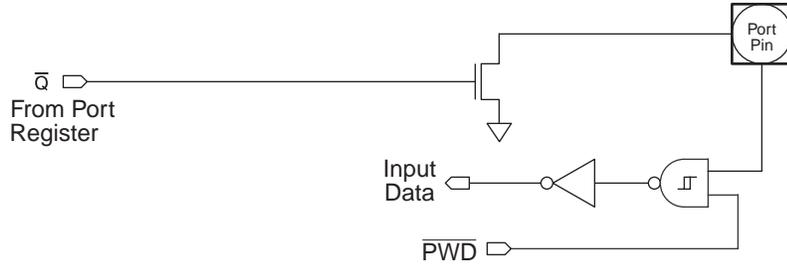
**Figure 13-3.** P1.3, P3.2 和 P3.3 只输入



### 13.0.3 开消耗输出

当端口门限包含逻辑“0”时，开消耗输出配置关闭多有的上拉，只驱动端口引脚的下拉电阻。为了用作逻辑输出，在这种方式下端口配置必须要有外部上拉电阻，典型的是电阻与  $V_{CC}$  相连。这种模式的下拉电阻与准双向模式相同。开消耗配置如 Figure 13-4 所示。在下电时，P1.3, P3.2 和 P3.3 的输入电路不是无效的 (见 Figure 13-3)。开消耗引脚可以被安全的上拉到 5.5V，即使工作在较低的  $V_{CC}$  电平；然而，斯密特触发器的输入门限是由  $V_{CC}$  电平设置的，这一点必须考虑。

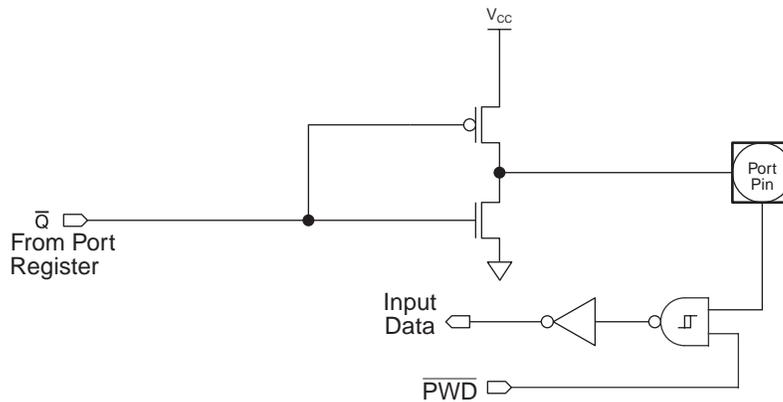
Figure 13-4. Open-Drain Output



### 13.0.4 推挽式输出

推挽式输出配置和开消耗与准双向输出模式有相同的下拉电阻结构，但是当端口门限包含逻辑“1”时，提供持续的强上拉电阻。当端口输出需要更多的源电流时，采用推挽模式。推挽端口配置如 Figure 13-5 所示。在下电时，P1.3、P3.2 和 P3.3 的输入电路不是无效的（见 Figure 13-3）。

Figure 13-5. 推挽式输出



### 13.1 端口 1 模拟功能

AT89LP213/214 结合了模拟比较器。为了给出做好的模拟效果和最小化功率损耗，作为模拟功能的引脚必须使他们的数字输出和数字输入无效。如 22 页“端口配置”描述的，使端口引脚处于只输入模式可以使端口引脚的数字输出无效。只要设置 ACSR 的 CEN 位使模拟比较器有效，那么 P1.0 和 P1.1 的数字输入是无效的。CEN 迫使 P1.0 和 P1.1 的 PWD 输入为的电平，因此使斯密特触发电路无效。在复位后不管三端口得状态如何，P1.0 和 P1.1 总是默认为只输入模式。

### 13.2 端口 读 - 修改 - 写

读引脚的状态还是读端口寄存器的状态取决于使用的指令。简单的读指令总是直接访问端口引脚读 - 修改 - 写 指令，就是读到一个值，可能修改这个值，然后重新写回去，这个指令总是访问端口寄存器。包括位写入指令，如 CLR 或 SETB，它们实际上读了整个端口，修改单独的位，然后把数据重新写回整个端口。如 Table 13-3 所示，列出了所有访问端口的读 - 修改 - 写指令。

Table 13-3. 端口读 - 修改 - 写指令

助记符	指令	举例
ANL	逻辑 AND	ANL P1, A
ORL	逻辑 OR	ORL P1, A
XRL	逻辑 EX-OR	XRL P1, A

**Table 13-3.** 端口读 - 修改 - 写指令

助记符	指令	举例
JBC	如果置位则跳转，同时清除该位	JBC P3.0, LABEL
CPL	补位	CPL P3.1
INC	递增	INC P1
DEC	递减	DEC P3
DJNZ	如果不是零则递减并跳转	DJNZ P3, LABEL
MOV PX.Y, C	把进位移动到端口 X 的 Y 位	MOV P1.0, C
CLR PX.Y	清除端口 X 的 Y 位	CLR P1.1
SETB PX.Y	设置端口 X 的 Y 位	SETB P3.2

### 13.3 端口选择功能

AT89LP213/214 的大多数通用数字输入 / 输出引脚 与各种串行单元需要的输入 / 输出共享功能 [Table 13-5](#) 列出了端口引脚的选择功能. 选择功能以逻辑与的方式与引脚相连. 为了使端口引脚的选择功能有效, 引脚相应的端口寄存器位必须为“1”, 否则输入 / 输出总是为“0”. 另外, 在使用前, 每个引脚必须按照外设的要求配置成正确的输入 / 输出模式. [Table 13-4](#) 给出了如何使用通用引脚配置为选择功能。

**Table 13-4.** 端口 x 引脚 y 选择功能配置

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

**Table 13-5.** 端口引脚选择功能

端口引脚	配置位		选择功能	注释
	PxM0.y	PxM1.y		
P1.0	P1M0.0	P1M1.0	AIN0	只输入
			GPI0	
P1.1	P1M0.1	P1M1.1	AIN1	只输入
			GPI1	
P1.2	P1M0.2	P1M1.2	GPI2	
P1.3	P1M0.3	P1M1.3	GPI3	$\overline{\text{RST}}$ 必须无效

**Table 13-5.** 端口引脚选择功能

端口引脚	配置位		选择功能	注释
	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	仅限于 AT89LP214
P3.1	P3M0.1	P3M1.1	TXD	
P3.2	P3M0.2	P3M1.2	INT0	仅限于内部 RC 振荡器
P3.3	P3M0.3	P3M1.3	INT1	仅限于内部 RC 振荡器或外部时钟源
			CLKOUT	
P3.4	P3M0.4	P3M1.4	T0	仅限于 AT89LP213
P3.5	P3M0.5	P3M1.5	T1	
P3.6	not configurable		CMPOUT	引脚与比较器输出相连

## 14. 性能增强型计时器计数器

AT89LP213/214 产品有两个 16 位的计时计数器 计时器 0 和计时器 1。作为计时器，每一个时钟周期寄存器增加一次。因此寄存器可以对时钟周期计数。因为每个时钟周期包括一个振荡周期，所以计数频率就是振荡器频率。计时器频率可以通过可变频计时器从 1 到 16 改变。（如表 9-2）。两个计时器使用同一个分频器。

作为计数器，相应引脚 T0 或 T1 处有 1 到 0 的变化时寄存器就会相应的递增。每一个时钟周期都会对外部输入进行采样。当采样值第一个周期为高电平，紧接着的那个周期为低电平，计数开始。一旦检测到这种跳变，寄存器中的值就会刷新开始递增。因为进行这样的检测需要两个时钟周期，所以最大的计数率也仅仅是振荡器频率的一半。对输入的信号没有约束，但信号必须确保检测到的电平在变化之前至少保持了一个完整的时钟周期。在 AT89LP214 中，T0 和 T1 是不能通过引脚输入的。但可以通过软件接换寄存器 3 中的 P3.4 和 P3.5 位来实现输入。

另外，计时器 0 和 1 就其功能来说有四种操作模式：可变宽度计时计数器，16 位自动重置计时计数器、8 位自动重置计时计数器和分时计时计数器。可以通过特殊功能寄存器 TMOD 中的控制位 C/T 选择计时计数器功能。TMOD 中的 M1 和 M0 位用来选择操作模式。

### 14.1 模式 0 – 可变频计时计数器

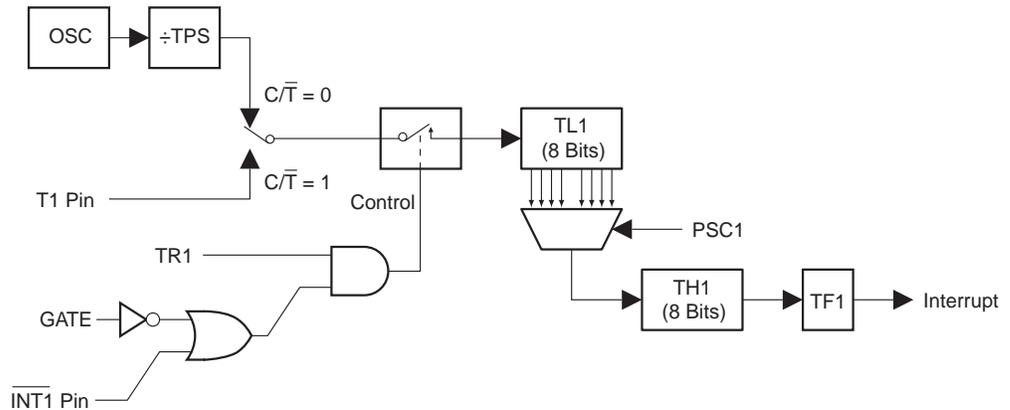
两个计时器在模式 0 下都是可变频的 8 位计数器。分频器范围从 1 到 8 位，到底能变几位这要取决于 TCONB 中的 PSC 位。这样计时器的范围就变成了 9 到 16 位。在标准 8051 模式 0 下的计时器位数缺省值位 13 位。如图 14 – 1 所示。当计数器从全 1 变为全 0 时，中断标识符 TF1 会被置位。计数输入是在 TR1=1 并且 GATE=0 或 INT1=1 有效的。给 GATE 置位使得外部输入 INT1 能够控制计

时器以便方便脉冲宽度测量。TR1 是特殊功能寄存器 TCON 中的控制位。GATE 是在 TMOD 中的。13 位寄存器是由 TH1 的 8 位和 TL1 中的低 5 位构成。TL1 的高 3 位未定忽略。对 TR1 置位不会清零寄存器。

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

Note: 在模式 0 下计时器 1 对 RH1/RL1 没有要求，可以用作临时存储寄存器。

Figure 14-1. 模式 0 下计时计数器：可变位数计数器



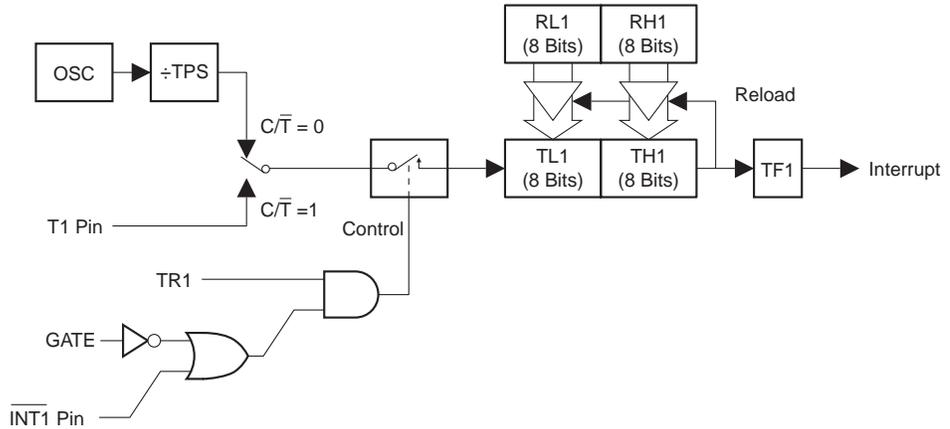
除了用 TRO、TF0 和 INTO 分别替代图 14 - 1 中相应的信号的情况，计时器 0 在模式 0 下的操作与计时器 1 相同。这里有两种不同的 GATE 位，计时器 1 的是 TOMD. 7，而计时器 0 则是 TOMD. 3。XTAL 振荡器使用  $\overline{\text{INT0}}$  和  $\overline{\text{INT1}}$  引脚。它们只有在内部 RC 振荡器作系统时钟时实现 GATE 功能。

## 14.2 模式 1 - 16 位自动重置计时计数器

计时器在模式 1 下是 16 位自动重置。计时寄存器 16 位运行。16 位重置值放在高低重置寄存器 RH1 和 RL1 中。高低计时寄存器 TH1 和 TL1 相互配合完成计时。当收到时钟信号，计时器开始计数：0000H、0001H、0002H……当 FFFFH 到 0000H 跳变时会发生溢出，这里还要注意计时寄存器是将 RH1 和 RL1 中的值拿来重置并且 TCON 中的溢出标识位已经置位。如图 14 - 2。重置缺省值为 0000H，这样的 16 位计时器可与标准 8051 相比。模式 1 操作同样适应于计时计数器 0。

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

**Figure 14-2.** 1 模式 1 下的计时计数器：16 位自动重置

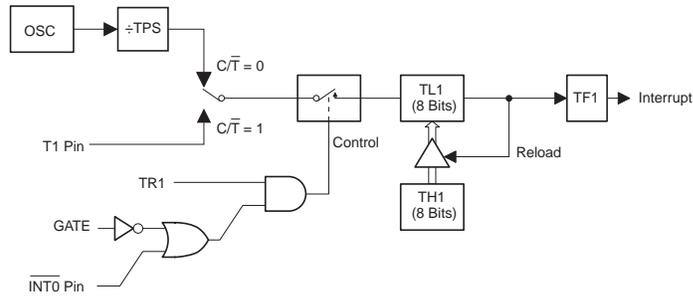


### 14.3 模式 2 – 8 位自动重置计时计数器

模式 2 下的计时寄存器是自动重置的 8 位 (TL1) 计数器。如图 14 – 3。TL1 溢出不仅要 对 TF1 置位，而且也要用 TH1 中的内容对 TL1 进行重置，这里的内容是预先用软件设置好的，而此时 TH1 内容并没有变化。模式 2 操作同样适合于计时计数器 0。

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

**Figure 14-3.** 模式 2 下的计时器计数器：8 位自动重置



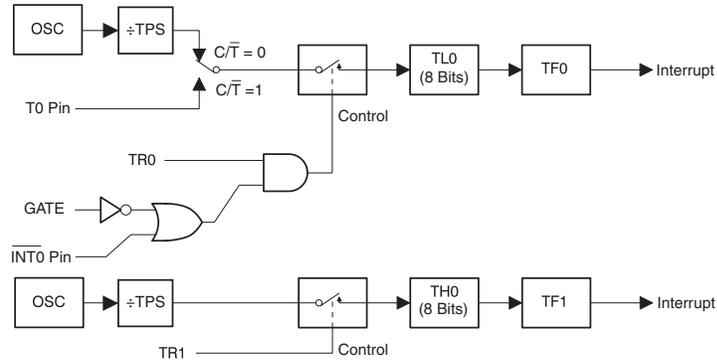
Note: 在模式 2 下计时器 1 对 RH1/RL1 没有要求，可以用作临时存储寄存器。

### 14.4 模式 3 – 8 位分时计时器

模式 3 下的计时器 1 简单的说就是保持计数，相当于对 TR1 清零。模式 3 下的计时器 0 将 TL0 和 TH0 独立，逻辑图如 14 – 4 所示。TL0 用到计时器 0 的控制位：C/T, GATE, TR0, INT0 和 TF0。TH0 用来对机器周期计数并且用到 TR1 和 TF1。因此 TH0 控制计时器 1 的中断。

模式 3 是为了满足额外 8 位计时计数器应用所需。从模式 3 下的计时器 0 看来，AT89LP213/214 产品看起来好像就拥有了三个计时计数器；这种情况下，计时器 1 可以通过切换进出模式 3 来实现关闭功能。这样看来计时器 1 仍然能作为波特率发生器的串口使用或任何不需要中断的应用。

**Figure 14-4.** 模式 3 下的计时器计数器 0: 两个 8 位计数器



**Note:** 在模式 3 下计时器 0 对 RH0/RL0 没有要求，可以用作临时存储寄存器。

**Table 14-1.** TCON – 计时器计数器控制寄存器

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

标识符	功能
TF1	计时器 1 溢出标识，有硬件设置。当操作向量指向中断时对其清零。
TR1	计时器 1 运行控制位，通过软件置位或清零来实现计时计数器通断。
TF0	计时器 0 溢出标识，有硬件设置。当操作向量指向中断时对其清零。
TR0	计时器 0 运行控制位，通过软件置位或清零来实现计时计数器通断。
IE1	中断 1 标识。当检测到外部中断时通过硬件置位，中断运行完清零。
IT1	中断 1 类型控制位。通过软件置位清零来指定下降沿 / 低电平触发中断。
IE0	中断 0 标识。当检测到外部中断时通过硬件置位，中断运行完清零。
IT0	中断 0 类型控制位。通过软件置位清零来指定下降沿 / 低电平触发中断。

**Table 14-2. TMOD: 计时计数器模式控制寄存器**

TMOD = 88H				复位值 = 0000 0000B																																																			
无位选																																																							
	GATE	$\overline{C/T}$	M1	M0	GATE	$\overline{C/T}$	M1	M0																																															
	7	$\overline{6}$	5	4	3	$\overline{2}$	1	0																																															
	计时器 1				计时器 0																																																		
Gate	门控制：当置位时，计时计数器只有在 INTx 引脚高电平和 TRx 控制引脚置位时才有效；当清零时，计时器 x 在 TRx 控制位置位时都有效。				计时器 0 门位																																																		
$\overline{C/T}$	计时器或计数器选择：若清零则为计时器操作（内部系统时钟输入）；若置位则为计数器操作（Tx 引脚作为输入）				计时器 0 计数记时选择位																																																		
M1	计时器 1 模式位 1				计时器 0 M1 位																																																		
M0	计时器 1 模式位 0				计时器 0 M0 位																																																		
	<b>M1</b>	<b>M0</b>	<b>模式</b>	<b>操作模式</b>																																																			
	0	0	0	9 到 16 位可变计时计数器 带有 1 到 8 位变长 THx 和 TLx 的 8 位计时计数器																																																			
	0	1	1	16 位自动重置记时计数器 16 位计时计数器，THx 与 TLx 连带，无变长器。																																																			
	1	0	2	8 位自动重置计时计数器 8 位自动重置计时计数器，THx 有重置值每次溢出时重置给 TLx。																																																			
	1	1	3	分时计时计数器模式 (计时器 0) TL0 时由标准计时器 0 控制位控制的 8 位计时计数器。 TH0 时标准计时器 1 控制位控制的 8 位计时计数器。																																																			
	1	1	3	(计时器 1) 计时计数器 1 停止																																																			
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">计时器 SFR</th> <th style="width: 25%;">作用</th> <th style="width: 25%;">地址</th> <th style="width: 25%;">位选</th> </tr> </thead> <tbody> <tr> <td>TCON</td> <td>控制</td> <td>88H</td> <td>是</td> </tr> <tr> <td>TMOD</td> <td>模式</td> <td>89H</td> <td>否</td> </tr> <tr> <td>TL0</td> <td>计时器 0 低字节</td> <td>8AH</td> <td>否</td> </tr> <tr> <td>TL1</td> <td>计时器 1 低字节</td> <td>8BH</td> <td>否</td> </tr> <tr> <td>TH0</td> <td>计时器 0 高字节</td> <td>8CH</td> <td>否</td> </tr> <tr> <td>TH1</td> <td>计时器 1 高字节</td> <td>8DH</td> <td>否</td> </tr> <tr> <td>TCONB</td> <td>模式</td> <td>91H</td> <td>否</td> </tr> <tr> <td>RL0</td> <td>计时器 0 重置低字节</td> <td>92H</td> <td>否</td> </tr> <tr> <td>RL1</td> <td>计时器 1 重置低字节</td> <td>93H</td> <td>否</td> </tr> <tr> <td>RH0</td> <td>计时器 0 重置高字节</td> <td>94H</td> <td>否</td> </tr> <tr> <td>RH1</td> <td>计时器 1 重置高字节</td> <td>95H</td> <td>否</td> </tr> </tbody> </table>							计时器 SFR	作用	地址	位选	TCON	控制	88H	是	TMOD	模式	89H	否	TL0	计时器 0 低字节	8AH	否	TL1	计时器 1 低字节	8BH	否	TH0	计时器 0 高字节	8CH	否	TH1	计时器 1 高字节	8DH	否	TCONB	模式	91H	否	RL0	计时器 0 重置低字节	92H	否	RL1	计时器 1 重置低字节	93H	否	RH0	计时器 0 重置高字节	94H	否	RH1	计时器 1 重置高字节	95H	否
计时器 SFR	作用	地址	位选																																																				
TCON	控制	88H	是																																																				
TMOD	模式	89H	否																																																				
TL0	计时器 0 低字节	8AH	否																																																				
TL1	计时器 1 低字节	8BH	否																																																				
TH0	计时器 0 高字节	8CH	否																																																				
TH1	计时器 1 高字节	8DH	否																																																				
TCONB	模式	91H	否																																																				
RL0	计时器 0 重置低字节	92H	否																																																				
RL1	计时器 1 重置低字节	93H	否																																																				
RH0	计时器 0 重置高字节	94H	否																																																				
RH1	计时器 1 重置高字节	95H	否																																																				

**Table 14-3.** TCONB – 定时器计数器控制寄存器 B

TCONB = 91H		复位值 = 0010 0100B						
无位选								
	PWM1EN	PWM0EN	PSC12	PSC11	PSC10	PSC02	PSC01	PSC00
位	7	6	5	4	3	2	1	0

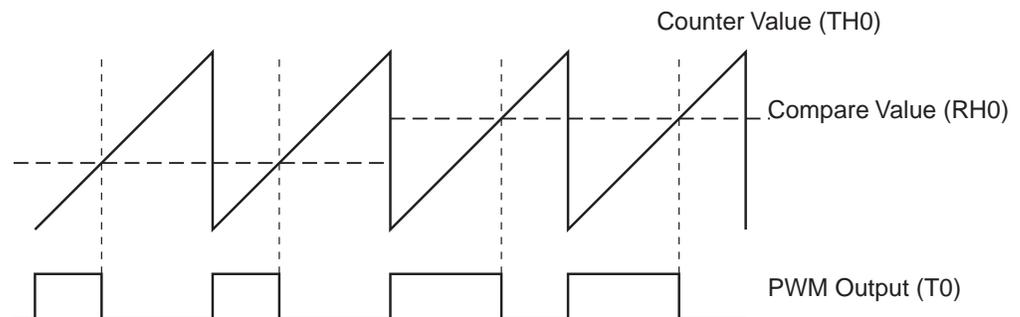
符号	功能
PWM1EN	在 T1 (P3.5) 上将定时器 1 设置成脉冲宽度调制输出。
PWM0EN	在 T0 (P3.4) 上将定时器 0 设置成脉冲宽度调制输出。
PSC12 PSC11 PSC10	模式 0 下定时器 1 的可变长器。TL1 可变位数为 PSC1+1。 在 AT89S2051 中复位后 PSC1=100B 使得 TL1 的 5 个位能够用于 13 位模式 0 的情况下。
PSC02 PSC01 PSC00	模式 0 下定时器 0 的可变长器。TL0 可变位数为 PSC0+1。 在 AT89S2052 中复位后 PSC0=100B 使得 TL0 的 5 个位能够用于 13 位模式 0 的情况下。

## 14.5 脉宽调制

AT89LP213 中，可以通过置位 TCONB 中 PWM0EN 或 PWM1EN 分别将定时器 0 和 1 配置成 8 位异步脉冲宽度调制器 (PWM)。在 PWM 模式下定时器引脚 T0 或 T1 输出产生的波形。在这种模式下 C/T 也必须置 0。T0 (P3.4) 和 T1 (P3.5) 必须设置为输出模式。定时器溢出标识和中断将在 PWM 模式下继续工作，定时器 1 仍可为 UART 口产生波特率。PWM 可以通过选择 TMOD 中的模式位来选择四种模式中的任意一种。

图 14-5 给出了在 PWM 模式下定时器 0 输出的一种波形。TH0 作为 8 位计数器使用而 RHO 存储 8 位比较值。当 TH0 为 00H 时，PWM 输出为高。当 TH0 中计数值累加到了 RHO 中的值时，PWM 输出置低。因此脉宽与 TH0 中的值一致。为了防止毛刺产生，仅在 RHO 从 FFH 到 00H 溢出时才写 RHO。将 RHO 置 00H 将会保持 PWM 输出为低。

**Figure 14-5.** 异步脉宽调制



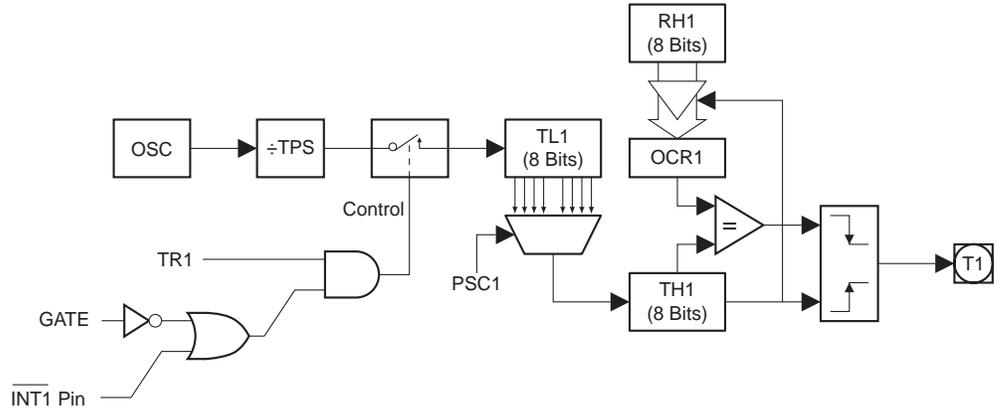
### 14.5.1 模式 0 – 带有 8 位分频器的 8 位 PWM

模式 0 下，TLx 作为算术分频器驱动 8 位计数器 THx。(如图 14-6) TCONB 中的 PSCx 位控制分频器的值。THx 溢出时，RHx 中的周期值转换给 OCRx，同时输出引脚置高。当 THx 中的计数与 OCRx 匹配时，输出清零。接下来便是给输出频率和 PWM 模式 0 下定时器 0 的周期。PWM 模式下定时器 1 与定时器 0 一致。

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

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

Figure 14-6. 计时器计数器 1PWM 模式 0



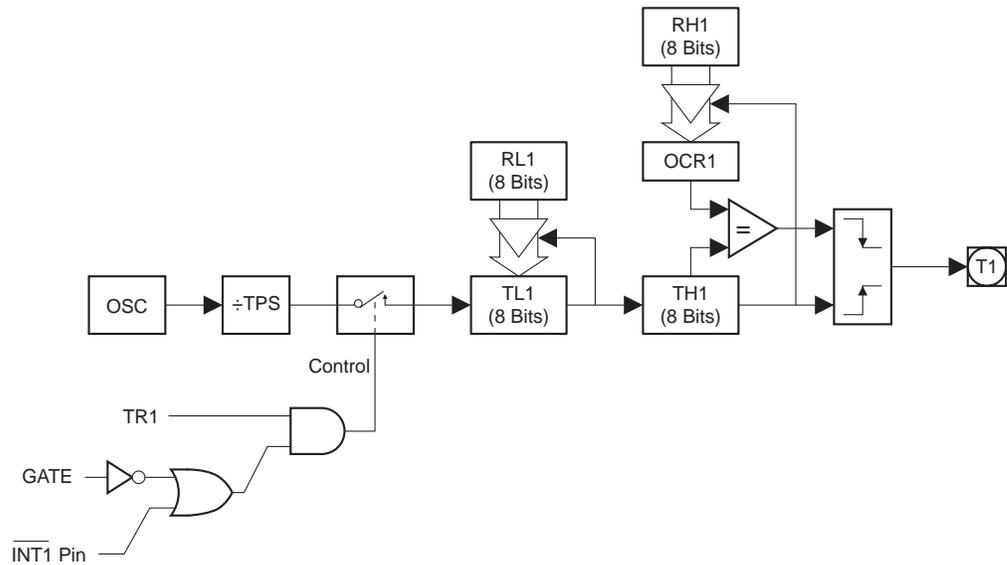
#### 14.5.2 模式 1- 带有 8 位线性分频器的 8 位 PWM

模式 1 下，TL<sub>x</sub> 利用 RL<sub>x</sub> 的 8 位自动重置实现线性分频（如图 14 - 7）。TL<sub>x</sub> 溢出时将 RL<sub>x</sub> 中的值置入 TL<sub>x</sub> 中。当 TH<sub>x</sub> 溢出时，RH<sub>x</sub> 中的周期值转换给 OCR<sub>x</sub>，同时输出引脚置高。当 TH<sub>x</sub> 中的计数与 OCR<sub>x</sub> 匹配时，输出清零。接下来便是给输出频率和 PWM 模式 0 下计时器 0 的周期。PWM 模式下计时器 1 与计时器 0 一致。

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

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

Figure 14-7. 计时器计数器 1 PWM 模式 1

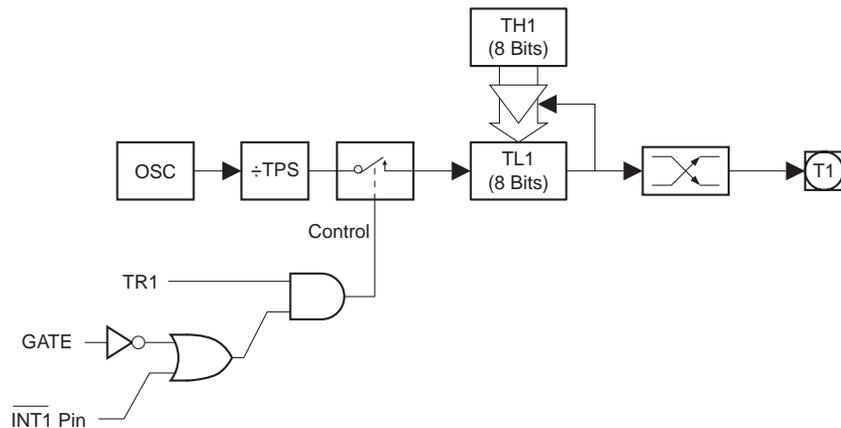


### 14.5.3 模式 2 – 8 位频率发生器

PWM 模式 2 下，计时器 0 与 8 位自动重置计时器功能类似。与通常情况下模式 2 相同，但是在每次 TL0 溢出时输出引脚 T0 接换。（如图 14 – 8）PWM 模式 2 下的计时器 1 与计时器 0 一致。模式 2 下 PWM 可以输出频率可变的方波。THx 可作 8 位计数器。接下来便是给 PWM 模式 2 下计时器 0 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. PWM 模式 2 下的计时器计数器 1



Note: {RH0 和 RL0}/{RH1 和 RL1} 在 PWM 模式 2 下计时器 0/1 不做要求，可被用作临时存储寄存器。

### 14.5.4 模式 3 – 8 位分时 PWM

PWM 模式 3 下计时器 1 保持其值，与设置 TR = 1 时的效果相同。PWM 模式 3 下计时器 0 将 TL0 和 TH0 作为两个独立的 PWM 计数器使用，其方式与通常情况下的模式 3 类似。PWM 模式 3 下的计时器 0 工作情况如图 14 – 9 所示。只有在 PWM 模式 3 下计时器可变量工作时才可以改变输出频率。TL0

可以使用计时器 0 的控制位：GATE, TR0,  $\overline{INT0}$ , PWM0EN 和 TF0。TH0 被锁存到，某个计时器功能中并且使用 TR1, PWM1EN 和 TF1。RLO 为 TLO 提供周期频率，RH0 则为 TH0 提供周期频率。

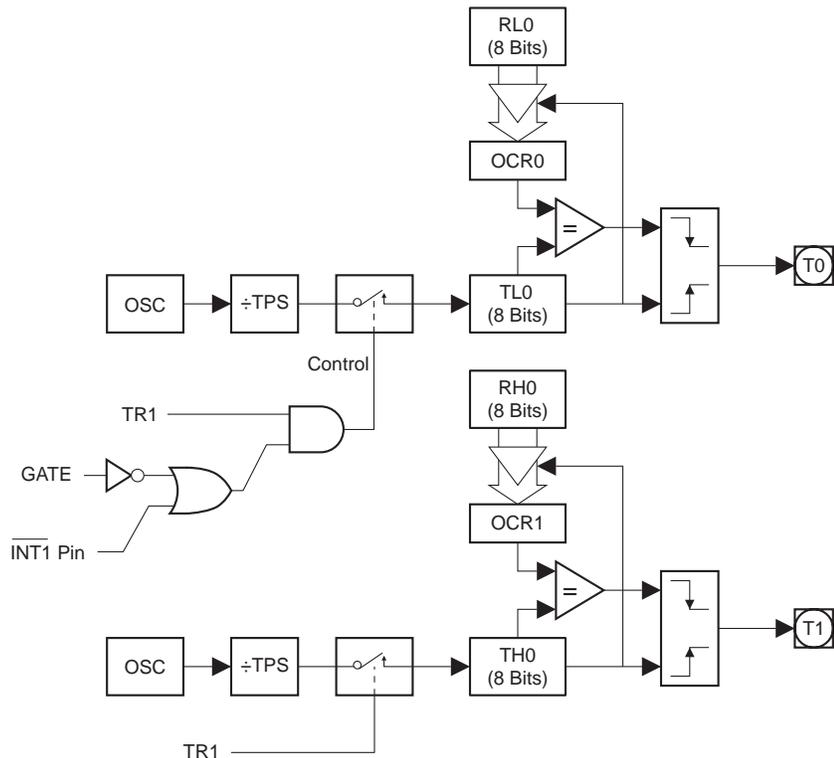
PWM 模式 3 应用时需要一个 PWM 通道和两个计时器，或者两个 PWM 通道和一个额外的计时器或计数器。对于 PWM 模式 3 下的计时器 0，AT89LP213 拥有三个计时器 / 计数器。当计时器 0 工作在 PWM 模式 3 下，计时器 1 可以通过通断控制来进入或退出模式 3。在这种情况下，计时器 1 仍可以作为串口的波特率产生器而无需任何中断。接下来便是给输出频率和 PWM 模式 3 下计时器 0 的周期。

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

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

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

Figure 14-9. PWM 模式 3 下的计时器计数器 0



## 15. 外部中断

当 AT89LP213/214 被设置为使用内部 RC 振荡器时，XTAL1 and XTAL2 可能被用作  $\overline{INT0}$  和  $\overline{INT1}$  外部中断资源。外部中断能够通过设置或清零寄存器 TCON 中 IT1 或 IT0 位达到水平活性或阶跃活性编程。如果  $ITx = 0$ ，外部中断 x 将会被  $\overline{INTx}$  引脚察觉到低电平时触发。如果  $ITx = 1$ ，外部中断将是边缘触发。在这个模式中，如果  $\overline{INTx}$  引脚中的连续样本表现为一个周期高电平而下一个周期为低电平，那么 TCON 中的中断请求标记  $IEx$  将被置位。标记位  $IEx$  开始请求中断。每当外部中断引脚取样一个时钟周期，一个输入高或低电平将被持续至少两个振荡器周期，目的是为了取样顺利。如果外部中断处于阶跃活性，那么外部资源将保持请求引脚高电平至少两个时钟周

期, 然后再将它置为低电平, 时间也是至少两个时钟周期; 这样, 中断请求标记 IEx 才能被置位。在边缘触发模式中, 如果服务周期被需求, 那么 IEx 将被 CPU 自动清除。如果外部中断处于水平活性, 那么外部资源将保持需求使能直到需求中断发生。然后, 外部资源必须在中断服务程序完全或另一个中断发生之前使请求无效。

## 16. 多用途中断

多用途中断 (GPI) 功能在端口 1 提供了 8 个可配置的外部中断。每个端口引脚都能探测到高 / 低电平或正 / 负边缘。GPIEN 寄存器挑选端口 1 中的位用于产生中断。GPMOD 和 GPLS 寄存器确定每个独立引脚的模式。GPMOD 在水平灵敏和边缘触发模式之间进行选择。GPLS 在水平模式下的高 / 低和边缘模式下的正 / 负之间进行选择。端口 1 的引脚每个时钟周期都在采样。在水平灵敏模式下, 一个正确的水平面必须在产生中断前出现在两个连续的采样之间。在边缘触发模式下, 当两个采样之间评估值产生变化时, 将会探测到一个阶跃。当一个中断条件在一个引脚被发现, 此时这个引脚使能, GPIF 寄存器中适当的标记被置位。GPIF 中的标记必须由软件清零。

**Table 16-1.** GPMOD – 多用途中断模式寄存器

GPMOD = 9AH							复位值 = 0000 0000B	
没有可以设置地址的位								
	GPMOD7	GPMOD6	GPMOD5	GPMOD4	GPMOD3	GPMOD2	GPMOD1	GPMOD0
Bit	7	6	5	4	3	2	1	0
<p>GPMOD.x    0 = level-sensitive interrupt for P1.x                         1 = edge-triggered interrupt for P1.x</p>								

**Table 16-2.** GPLS – 多用途中断水平选择寄存器

GPLS = 9BH							复位值 = 0000 0000B	
没有可以设置地址的位								
	GPLS7	GPLS6	GPLS5	GPLS4	GPLS3	GPLS2	GPLS1	GPLS0
Bit	7	6	5	4	3	2	1	0
<p>GPMOD.x    0 = detect low level or negative edge on P1.x                         1 = detect high level or positive edge on P1.x</p>								

**Table 16-3. GPIEN – 多用途中断使能寄存器**

GPIEN = 9CH							复位值 = 0000 0000B	
没有可以设置地址的位								
	GPIEN7	GPIEN6	GPIEN5	GPIEN4	GPIEN3	GPIEN2	GPIEN1	GPIEN0
Bit	7	6	5	4	3	2	1	0
GPIEN.x    0 = interrupt for P1.x disabled 1 = interrupt for P1.x enabled								

**Table 16-4. GPIF – 多用途中断标记寄存器**

GPIF = 9DH							复位值 = 0000 0000B	
没有可以设置地址的位								
	GPIF7	GPIF6	GPIF5	GPIF4	GPIF3	GPIF2	GPIF1	GPIF0
Bit	7	6	5	4	3	2	1	0
GPIF.x    0 = interrupt on P1.x inactive 1 = interrupt on P1.x active. 必须由软件清除。								

## 17. 连续分界面

AT89LP214 上的连续分界面实现了通用异步接收器 / 发送器功能 (UART)。UART 具有以下特点:

- 全双工操作
- 8 或 9 个数据位
- 错误侦查
- 自动地址识别下的多处理器传输模式
- 使用定时器 1 的波特率发生器
- 接收缓冲器或传输完全上的中断

连续分界面是全双工的，这意味着它能够同时发送和接收信号。它同时也是一个接收缓冲器，这意味着它能在前一个字节被接收寄存器读取之前接收第二个字节信息。（尽管如此，如果第二个字节接收完成而第一个字节还没有被读取时，这两个字节中的一个将会丢失。）连续端口接收和发送寄存器都能访问特殊功能寄存器 SBUF。发送寄存器负责向 SBUF 中写入数据；而接收寄存器负责从 SBUF 中读取数据。连续端口能在以下四个模式中进行操作：

- **模式 0:** 通过 RXD 实现连续数据的进入和退出。TXD 输出变换时钟。8 个数据位是发送 / 接收。波特率固定为振荡器频率的 1/2。
- **模式 1:** 10 个位是发送（通过 TXD）或接收（通过 RXD）：1 个开始位 (0)，8 个数据位，1 个停止位 (1)。接收时，停止位变为特殊功能寄存器 SCON 中的 RB8。波特率基于定时器 1 可变。
- **模式 2:** 11 个位是发送（通过 TXD）或接收（通过 RXD）：1 个开始位 (0)，8 个数据位，1 个可编程的第 9 数据位，1 个停止位 (1)。发送时，第 9 数据位（SCON 中的 TB8）可以被赋值“0”或“1”。例如，奇偶位 (P，在 PSW 中) 能够被移动至 TB8。接收时，当停止位被忽视时，第 9 数据位将去特殊功能寄存器 SCON 中的 TB8。波特率可编程设置为振荡器周期的 1/16 或 1/32
- **模式 3:** 11 个位是发送（通过 TXD）或接收（通过 RXD）：1 个开始位 (0)，8 个数据位，1 个可编程的第 9 数据位，1 个停止位 (1)。事实上，模式 3 与模式 2 除了波特率以外其余都一模一样。模式 3 的波特率基于定时器 1 可变。

在所有的 4 种模式中，发送器能被将 SBUF 定义为目的寄存器的任何指令启动。模式 0 中，当 RI = 0 且 REN = 1 时，开始接收。在其他模式中，当引入的启动位 REN=1 时，开始接收。

## 17.1 多处理器通信

模式 2 和模式 3 有一个多处理器通信模式。在这个模式中，9 个数据位用于接收，1 个停止位紧随其后。第 9 个位去 RB8。然后产生一个停止位。端口仅在停止位接收时可以被编辑；只有当 RB8=1 时，连续端口中断才具备活性。通过设置 SCON 中的 SM2 位，能实现这一特点。

接下来的例子将会说明如何在多处理器通信中使用连续中断。当主处理器必须将一块数据传输给其他处理器时，它首先发送一个地址位，用于鉴别目标处理器。这个地址位不同于数据位，因为在地址位中第 9 位是“1”，而在数据位中第 9 位是“0”。当 SM2=1 时，没有处理器能够被数据位阻止。然而，地址位可以中断所有的处理器。每个处理器都能检验接收字节并且鉴别它是否属于地址位。地址处理器清除了它的 SM2 位，并且准备接收接下来的数据字节。处理器不能设置它的 SM2 位，并且忽略数据字节。

SM2 位在模式 0 中没有任何作用，但它可以用来检查模式 1 中停止位的合法性。在模式 1 接收中，如果 SM2=1，那么除非收到一个有效的停止位，不然接收中断都具有活性。

**Table 17-1.** SCON – 连续端口控制寄存器

SCON 地址 = 98H		复位值 = 0000 0000B						
可以设置地址的位								
	SM0/FE	SM1	SM2	REN	TB8	RB8	T1	RI
Bit	7	6	5	4	3	2	1	0
(SMOD0 = 0/1) <sup>(1)</sup>								

符号	功能																									
FE	取景错误位。当发现一个无效停止位时，接收器将它置位。FE 位不能被有效取景清零而只能被软件清零。SMOD0 位必须被设置为使能 FE 位。FE 的设置与 SMOD0 的规定无关。																									
SM0	连续端口模式位 0, (SMOD0 must = 0 to access bit SM0)																									
SM1	连续端口模式位 1																									
	<table border="1"> <thead> <tr> <th>SM0</th> <th>SM1</th> <th>Mode</th> <th>Description</th> <th>Baud Rate<sup>(2)</sup></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>shift register</td> <td><math>f_{osc}/2</math></td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>8-bit UART</td> <td>variable (Timer 1)</td> </tr> <tr> <td>1</td> <td>0</td> <td>2</td> <td>9-bit UART</td> <td><math>f_{osc}/32</math> or <math>f_{osc}/16</math></td> </tr> <tr> <td>1</td> <td>1</td> <td>3</td> <td>9-bit UART</td> <td>variable (Timer 1)</td> </tr> </tbody> </table>	SM0	SM1	Mode	Description	Baud Rate <sup>(2)</sup>	0	0	0	shift register	$f_{osc}/2$	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	Description	Baud Rate <sup>(2)</sup>																					
	0	0	0	shift register	$f_{osc}/2$																					
	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	在模式 2 或 3 中，自动地址识别功能使能。当 SM2=1 时，除非接收到的第 9 位数据为 1，否则 RI 不能被置位。在模式 1 中，如果 SM2=1，除非接收到一个有效停止位，否则 RI 不能被激活，接收字节是一个赠与或广播地址。在模式 0 中，SM2 是 0。																									
REN	连续接收使能。由软件置位，由软件清零。																									
TB8	在模式 2 和模式 3 中传输第 9 数据位。由软件置位或清零。																									
RB8	在模式 2 和模式 3 中接收第 9 数据位。在模式 1 中，如果 SM2=0，RB8 将作为停止位被接收。在模式 0 中，RB8 不被使用。																									
TI	传输中断标志。无论处于哪种连续传输模式下，在模式 0 中，由 8 位时间硬件设置，在其他模式中，由地址位的开始状态设置。必须由软件清零。																									
RI	接收中断标志。在任何连续接收状态下（除了 SM2），在模式 0 中，由 8 位时间的末尾设置；或者在其他模式下由停止位时间的部分设置。必须由软件清零。																									

- Notes: 1. SMOD0 is 定位于 PCON.6.  
2.  $f_{osc}$  = 振荡器周期

## 17.2 波特率

模式 0 中，波特率是固定的，如下式所示：

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

模式 2 中，波特率由特殊功能寄存器中的 SMOD1 位的值决定。如果 SMOD1=0，那么波特率为振荡器周期的 1/32。如果 SMOD1=1，那么波特率为振荡器周期的 1/16。如下式所示：

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

### 17.2.1 利用定时器 1 产生波特率

在模式 1 和 3 中，定时器 1 的溢出率决定波特率。当定时器 1 是波特率发生器时，波特率由定时器 1 的溢出率和 SMOD1 的值决定。如下式所示：

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

在这个请求中，定时器 1 中断将被使无效。在 3 个连续模式下，定时器本身能被设置为任何定时器或计数器运行。在最典型的应用程序中，它被设置为自动再装模式下的定时器运行。既然这样，波特率由下式给出：

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

程序员可以通过设定定时器以 16 位自动再装定时器而获得非常低的波特率。既然这样，波特率由下式给出：

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

Table 17-2 给出通常使用的波特率以及它们是如何从定时器 1 中获得的。

**Table 17-2.** 定时器 1 中常使用的波特率 (TPS = 0000B)

Baud Rate	f <sub>osc</sub> (MHz)	SMOD1	Timer 1		
			C/T	Mode	Reload Value
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
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 关于模式 0 的更多内容

通过 RXD 实现连续数据进入和退出。TXD 输出变换时钟。8 个数据位用于发送。波特率固定为振荡器周期的 1/2。Figure 17-1 on page 40 显示了在模式 0 及关联定时状态下连续端口的单一化功能图表。

发送器能被将 SBUF 定义为目的寄存器的任何指令启动。“向 SBUF 写入”信号等同于将“1”置于传输变换寄存器的第 9 位，并使 TX 控制模块开始传输。内部定时在“向 SBUF 写入”和 SEND 激活中将机器周期流逝。

SEND 将交换寄存器的输出传送到 P3.0 的轮换功能输出线上，它也将变换时钟传送到 P3.1 的轮换功能输出线上。在变换时钟下降沿，传输轮换寄存器的内容将变换到右边的位置上。

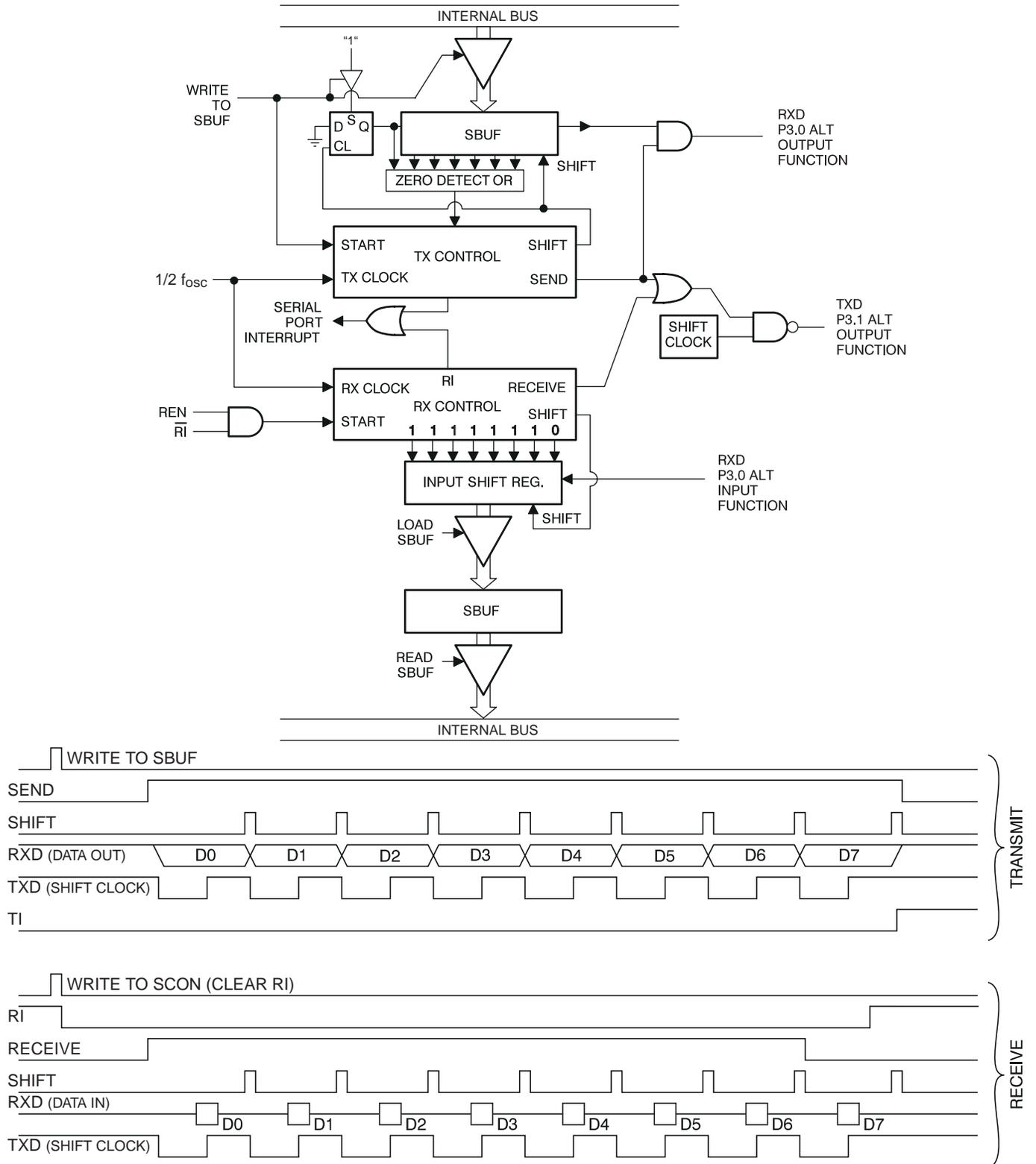
当数据位向右变换时，“0”向左边移动。当数据字节中的 MSB 在变换寄存器的输出位置时，将第 9 个位置的最初部分置 1。这个控制标志 TX 控制时钟进行最后轮换，然后解除 SEND 并将 TI 置 1。

当 REN = 1 且 R1 = 0 时，开始接收。在下一个时钟周期，RX 控制单元将 11111110 写入接收轮换寄存器并且在下一个时钟相位激活 RECEIVE i

RECEIVE 将轮换时钟设置到 P3.1 的轮换输出功能线上。在轮换时钟的下降沿，接收变换寄存器中的内容向左移动。从右侧移位过来的数值来自于 P3.0 引脚在变换时钟上升沿的取样。

当数据位向右移动时，“1”向左移动。当将“0”写入变换寄存器最右边的最初位置时，它标志RX 控制时钟去进行一个最后的轮换。然后将 RECEIVE 清零并将 RI 置 1。

Figure 17-1. 连续端口模式 0



## 17.4 关于模式 1 的更多内容

10 位传输（通过 TXD）或接收（通过 RXD）：1 个开始位（0），8 个数据位，1 个停止位。接收时，停止位进入 SCON 中的 RB8。在 AT89LP214 中，波特率由定时器 1 的溢出率决定。Figure 17-2 显示在传输和接收过程中，模式 1 和关联定时的连续端口的单一化功能图表。

发送器能被将 SBUF 定义为目的寄存器的任何指令启动。“向 SBUF 写入”信号等同于将“1”置于传输变换寄存器的第 9 位，并使 TX 控制模块开始传输。传输实际上从机器周期的 S1P1 开始。因此，因此，位时间是同步的，而不是“向 SBUF 写入”信号。

当  $\overline{\text{SEND}}$  具有活性时，开始传输。1 位时间过后，DATA 被激活，它将传输轮换寄存器使能。在此之后，第一个轮转脉搏出现 1 位时间。

当数据位向右移动，“0”向左移动。当数据位的 MSB 在轮换寄存器的输出位置时，“1”写入第 9 位置的开头，所有位置向左移。这个控制标志 TX 控制单元去进行最后的轮换，然后解除  $\overline{\text{SEND}}$  并将 TI 置 1。

当 RXD 探测到一个 1 至 0 的转变时，开始接收。为了达到这个目的，RXD 以确定波特率的 16 被速率取样。当探测到传输时，16 分频计数器立刻重启，输入轮换寄存器中写入 1FFH。16 分频计数器通过引入的位时间重新设置。

16 分频计数器被分为 16 个位时间。在每个位时间的第 7、8、9 计数器状态下，位探测器取样 RXD 的值。接收的样本最少需要 3 个样本中的两个。这样可以拒绝噪声。为了拒绝错误位，如果接收的第 1 位时间不为 0，那么接收电路重启，单元继续寻找 1 至 0 传输。如果开始位有效，它转变为输入轮换寄存器。

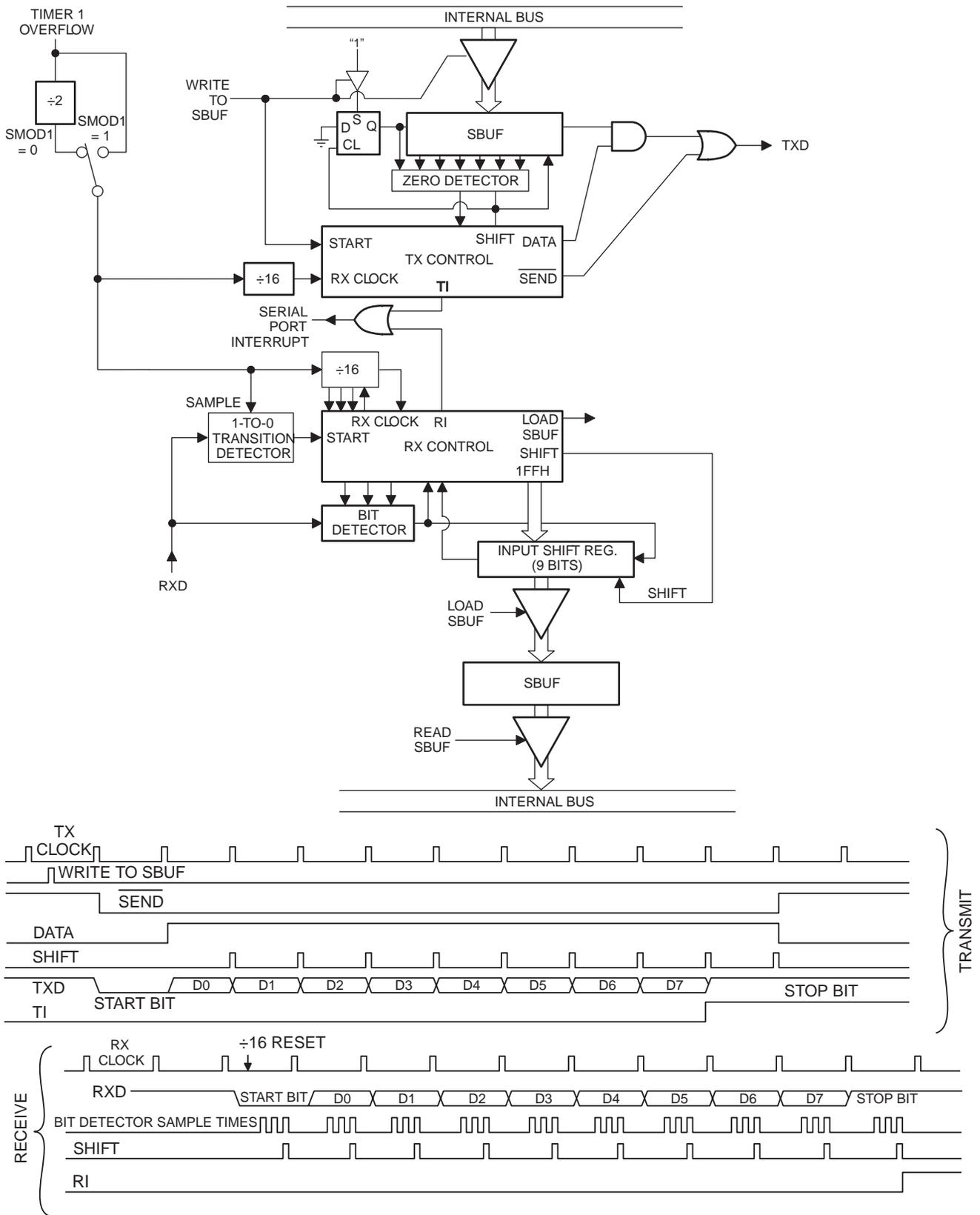
当数据位向右移动时，“1”向左移动。当开始位到达轮换寄存器最左边的位置时，它标记 RX 控制块去做最后的轮换，加载 SBUF 和 RB8，并将 RI 置 1。当且仅当满足下列条件时，产生加载 SBUF 和 RB8 并且将 RI 置 1 的信号。

$\text{RI} = 0$  且

$\text{SM2} = 0$ , 或接收停止位 = 1

如果这两个条件中的任何一个不满足，接收结构将不可挽回的丢失。如果两个条件同时满足，停止位进入 RB8，8 个数据位写入 SBUF，并且 RI 被激活。此时，不论上述条件是否满足，单元都将继续在 RXD 中寻找 1 至 0 的传输。

Figure 17-2. 连续端口模式 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）进入移位计数器第 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. 连续端口模式 2

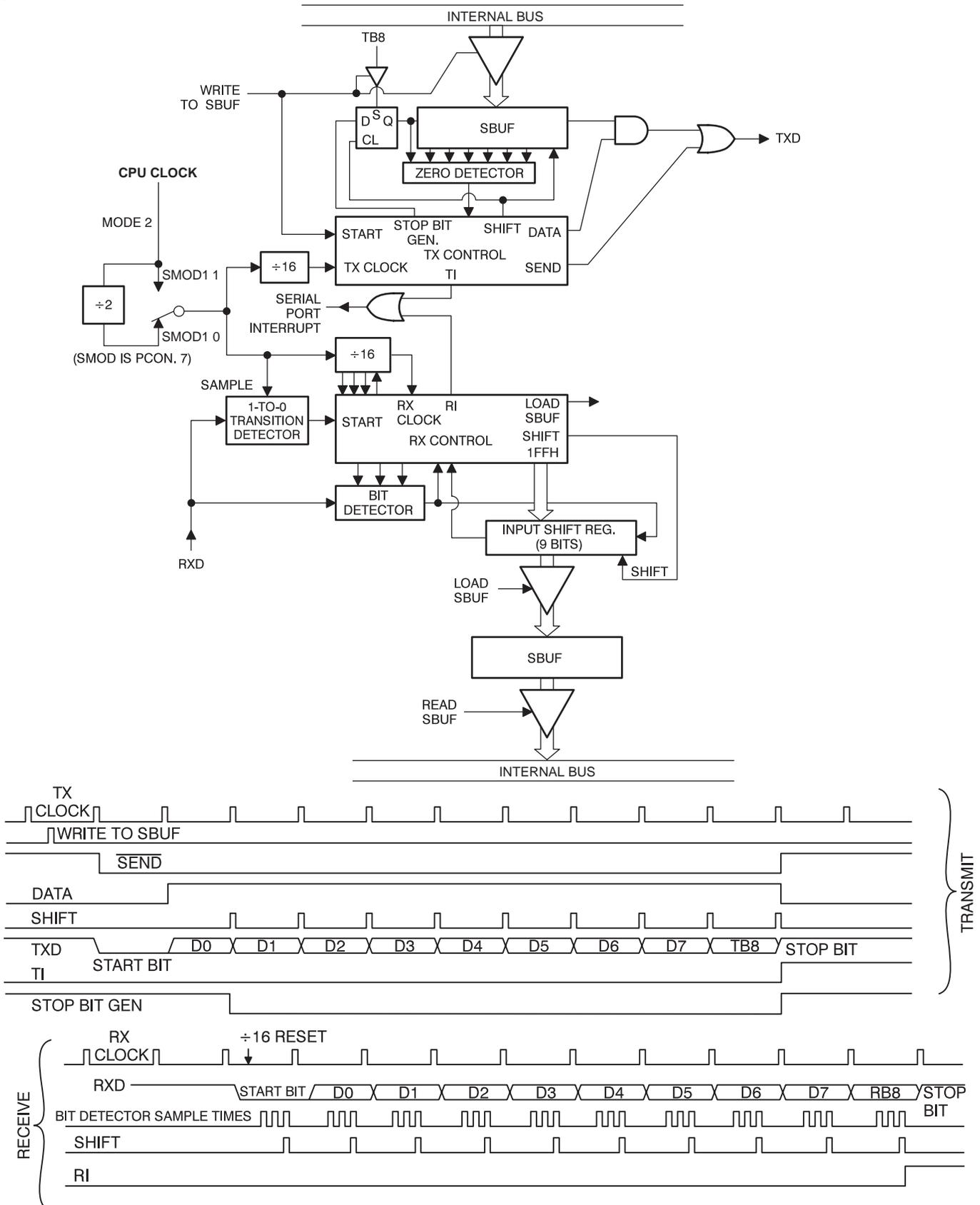
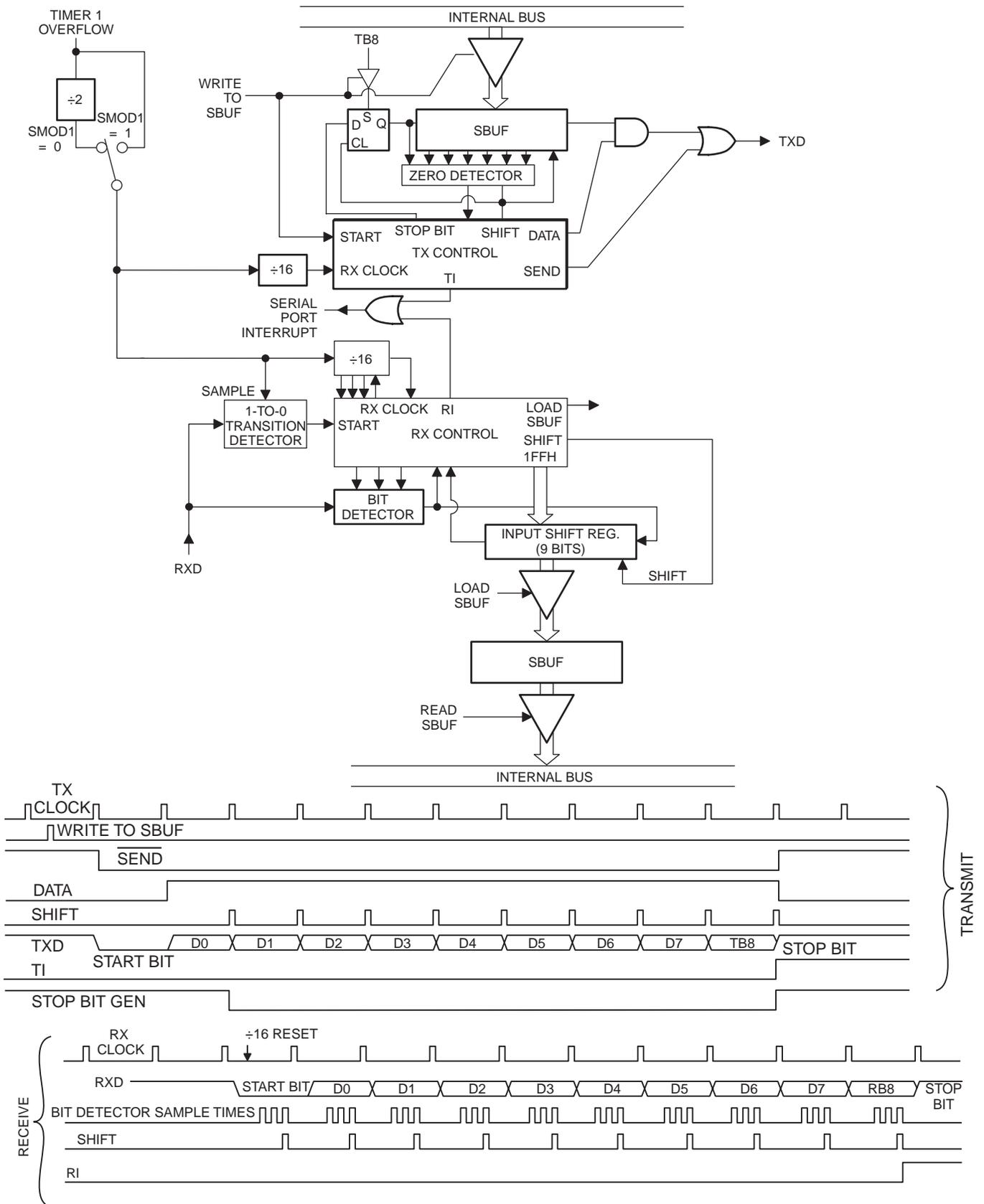


Figure 17-4. Serial Port Mode 3



## 17.6 帧错误检测

除了它所有的通用模式外，UART 可以作为通过寻找丢失的停止位进行帧错误检测，并自动的地址识别。当用于帧错误检测时，UART 在通信中寻找丢失的停止位。一个丢失的停止位会将 SCON 寄存器的 FE 置位。FE 由 PCON. 6 (SMOD0) 决定是作为 SCON. 6 还是 SM0。如果 SMOD0 置位，那么 SCON. 7 就作为 FE，此时 SCON. 7 仅能由软件清零。当 SMOD0 清零，那么 SCON. 7 作为 SM0。

## 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 0XX0
```

分端口 1	SADDR = 1110 0000
	SADEN = <u>1111 1010</u>
	Given = 1110 0X0X
分端口 2	SADDR = 1110 0000
	SADEN = <u>1111 1100</u>
	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。

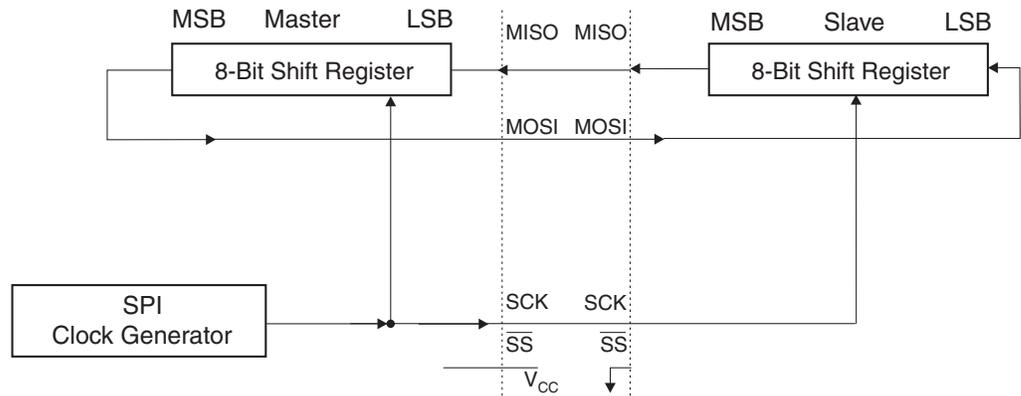
## 18. 串行外设接口

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 = f<sub>osc</sub>/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 终端使能。它与寄存器 IE 中的 ES 位同时置位使能 SPI 中断。SPIE = 0 时禁用 SPI 中断
SPE	SPI 使能。SPI = 1 时，启用串行通道，分别连接 $\overline{SS}$ , MOSI, MISO 和 SCK 至管脚 P1.4, P1.5, P1.6, 和 P1.7。SPI = 0 时禁用串行通道。
DORD	数据顺序位。DORD = 1 选择数据的 LSB。DORD = 0 选择数据的 MSB。
MSTR	主 / 从选择位。当 MSTR = 1 时为主 SPI 模式。MSTR = 0 时为从 SPI 模式。

符号	功能															
CPOL	时钟极性。当 CPOL = 1 时，SCK 在空闲时为高。 当 CPOL = 0 并且不传输数据时，主设备的 SCK 为低。同时请参考 SPI 时钟状态和极性控制。															
CPHA	时钟相位。CPHA 和 CPOL 位一起控制主从设备之间的数据和时钟的关系。同时请参考 SPI 时钟状态和极性控制。															
SPR0 SPR1	<p>SPI 时钟频率选择。在主模式下，这两个位控制主设备的 SCK 频率，在从模式下这两位无效。SCK 频率和振荡器频率的关系如下：</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><math>f_{osc}/4</math></td> </tr> <tr> <td>0</td> <td>1</td> <td><math>f_{osc}/8</math></td> </tr> <tr> <td>1</td> <td>0</td> <td><math>f_{osc}/32</math></td> </tr> <tr> <td>1</td> <td>1</td> <td><math>f_{osc}/64</math></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. 设定时钟模式再启用 SPI: 先设置 SPCR 中除了 SPE 位的其他位，再设置 SPE 位。
  2. 先使能主设备再使能从设备。
  3. 如果没有装载新的数据的话，从设备会在下一个 Tx 时刻反馈给主设备信号。

**Table 18-2. SPSR – SPI 状态寄存器**

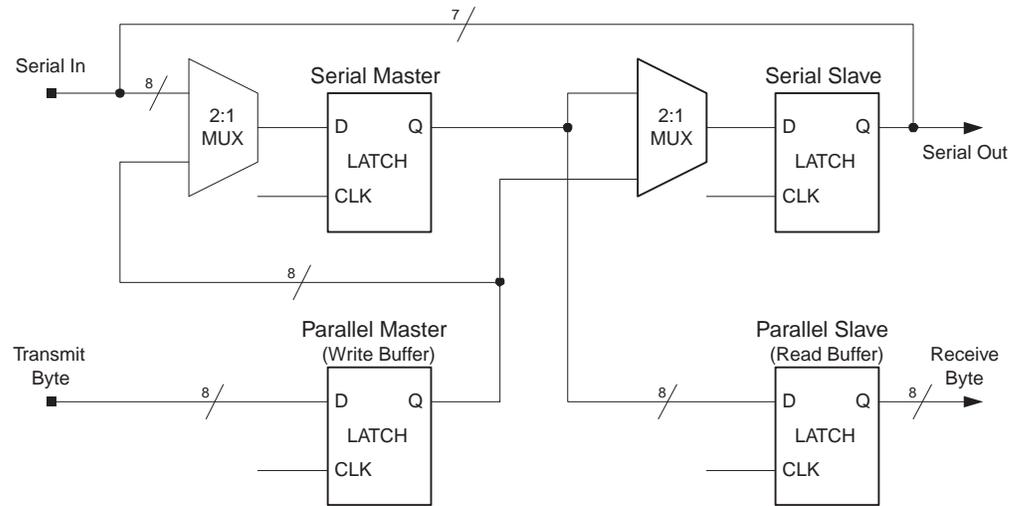
SPDR Address = EAH								复位值 = 00H (after cold reset) unchanged (after warm reset)
不可位寻址								
Bit	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	7	6	5	4	3	2	1	0

**Table 18-3. SPSR – SPI Status Register**

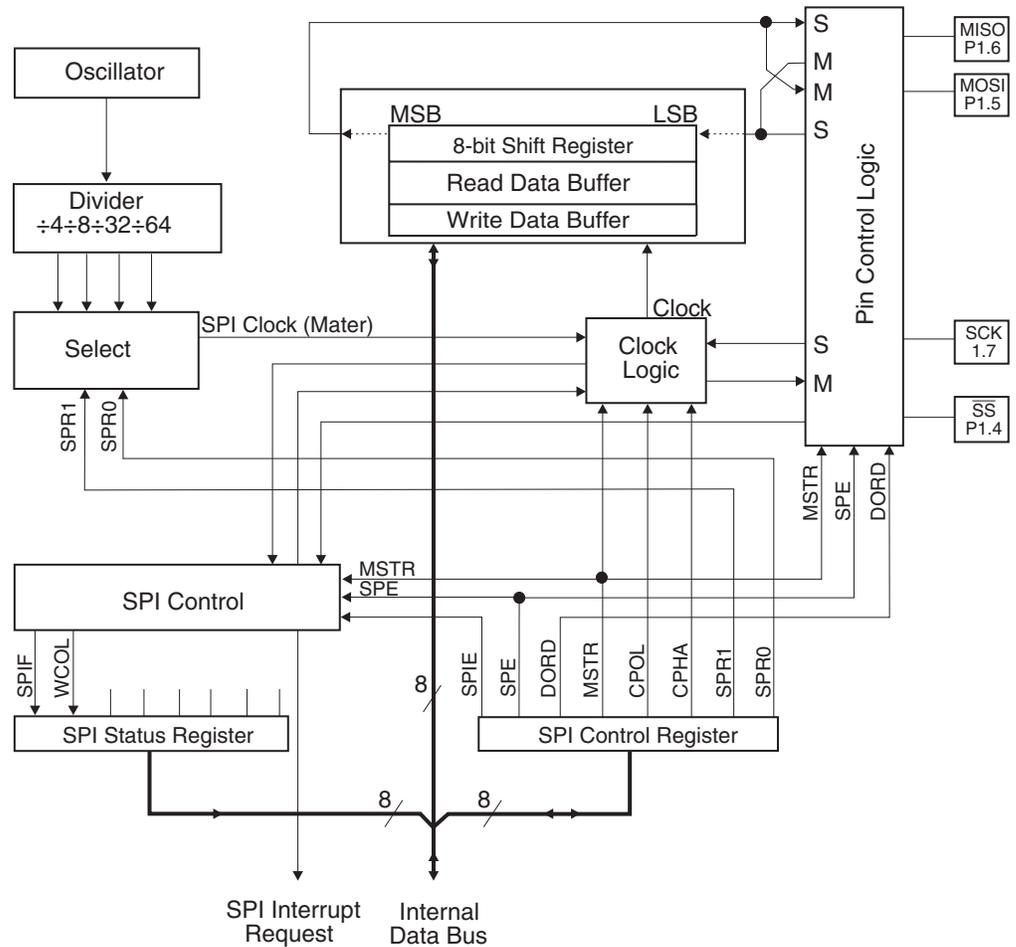
SPSR Address = E8H								复位值 = 000X X000B
不可位寻址								
Bit	SPIF	WCOL	LDEN	–	–	SSIG	DISSO	ENH
	7	6	5	4	3	2	1	0

符号	功能
SPIF	SPI 中断标志。当串行传输完成时 SPIF 会被置位，如果此时 SPIE = 1 且 ES = 1，那么就会产生中断。通过读取 SPI 状态寄存器然后再读 / 写 SPI 数据寄存器可以将 SPIF 清零。
WCOL	<p>当 ENH = 0 时：WCOL 是写入冲突标志。如果在传输数据时向 SPI 数据寄存器中写入数据，将置位 WCOL。此时若读取寄存器 SPDR 将得到错误的数值，且向 SPDR 写入数据也是无效的。通过读取 SPI 状态寄存器然后再读 / 写 SPI 数据寄存器可以将 WCOL bit ( 和 SPIF 位 ) 清零。</p> <p>当 ENH = 1 时：WCOL 在增强模式下作为 Tx 缓冲满标志。在增强模式下，当 WCOL = 1 时写入数据，将会覆盖已在 Tx 缓冲区内等待发送的数据。此时，WCOL 不再由 SPIF 的复位来清零，而是写入缓冲区的数据被装入移位寄存器时被清零。</p>
LDEN	<p>增强 SPI 模式下 Tx 缓冲区装载使能。</p> <p>在 ENH 被置位的情况下，并且 LDEN = 1，WCOL = 0，此时向 Tx 缓冲区装载数据是安全的。在串行字节传输的前 0 - 3 位时 LDEN 为高，4 - 7 位时 LDEN 为低。</p>
SSIG	从属选择忽略。如果 SSIG = 0，如果 $\overline{SS}$ (P1.4) 被拉低，SPI 将只工作于从属模式。当 SSIG = 1，SPI 忽略从属模式下的 $\overline{SS}$ ，并且无论 SPE (SPCR.6) 是否置位，它都有效。当 SSIG = 1 时，P1.4 可以用于通用 I/O 引脚。
DISSO	<p>禁止从输出位</p> <p>置位时，使得 MISO 管脚处于三态，能让多个从设备共用一个接口。通常传输的第一个字节是从设备的地址，只有被选定的从设备才把它的 DISSO 位置零。</p>
ENH	<p>增强 SPI 模式选择位。当 ENH = 0 时，SPI 工作在普通模式下，没有写入数据缓冲</p> <p>当 ENH = 1 时，SPI 工作在增强模式下，有写入数据缓冲。Tx 缓冲区和寄存器 SPDR 公用一个地址。</p>

**Figure 18-2.** SPI 移位寄存器图表

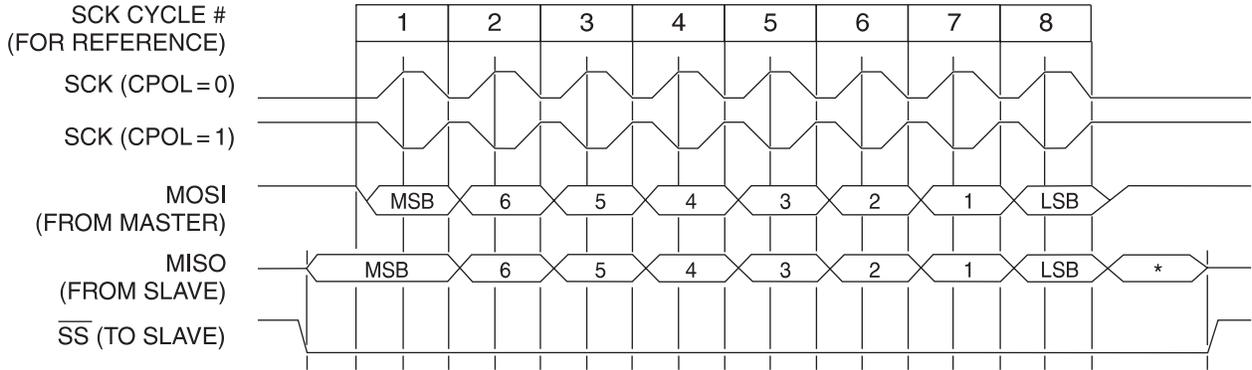


**Figure 18-3.** SPI 结构图



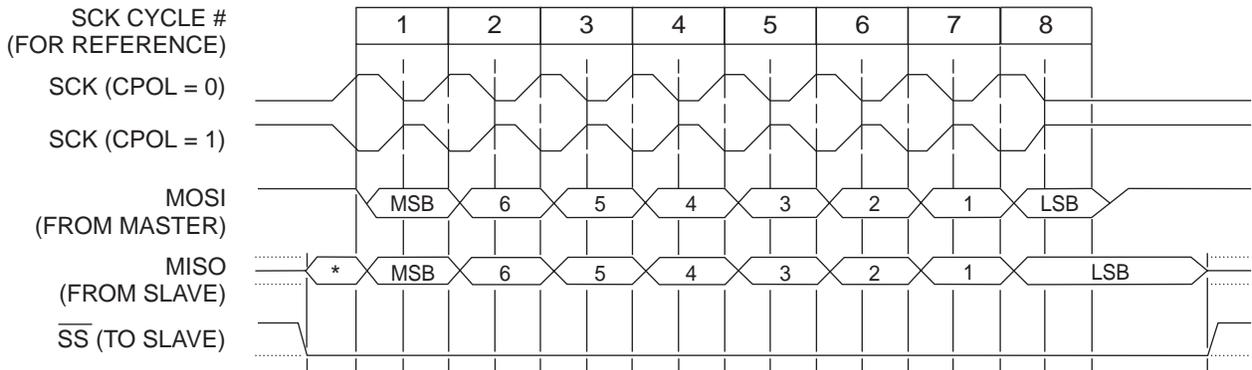
CPHA (Clock PHase时钟相位), CPOL (Clock POLarity时钟极性), 以及SPCR中的 SPR (Serial Peripheral clock Rate = baud rate 串行外围时钟率=波特率)位控制SCK的形状及速度。当SPI在主模式下时, 两个SPR位提供四个可能的时钟频率。在从模式下, 只要输入的SCK频率不超过最大的位频率, SPI将以它的频率工作。至于串行数据, 这里也有四种可能的SCK相位以及极性组合。CPHA以及CPOL决定哪一个版本用于传输。SPI数据传输的版本见Figures 18-4 and 18-5。为了避免SCK的短时脉冲干扰破坏接口, CPHA, CPOL, 以及SPR在接口被使能时不应被修正, 并且主器件应该在从器件之前被使能。

Figure 18-4. CPHA = 0 时 SPI 的转换波形图



Note: \* 非定义但是正常的 MSB 特性被接受。

Figure 18-5. CPHA = 1 时 SPI 的转换波形



Note: \* 非定义但是先前传输的正常的 LSB 特性。

## 19. 模拟比较器

AT89LP213/214 上提供一个单独的模拟比较器。模拟比较器有以下特性：

- 比较器输出标志及中断
- 可选中断条件
  - 高或低等级
  - 上升或下降边缘
  - 输出跳转
- 硬件跳转模式

比较器操作是这样的，当正输入 AIN0 (P1.0) 比负输入 AIN1 (P1.1) 大时输出逻辑 1。否则输出 0。置位 ACSR 中的 CEN 位使能比较器。当比较器第一次被使能，比较器输出并且中断标志在 10  $\mu$ s 中不一定稳定。相应的比较器在这一时间段不应该被使能，而比较器中断标志必须在中断被使能前清零以避免一个直接的中断服务。在使能比较器之前模拟输入应通过将 P1.0 以及 P1.1 置入仅输入模式来置为三态。见“[端口 1 模拟功能](#)” on page 24。

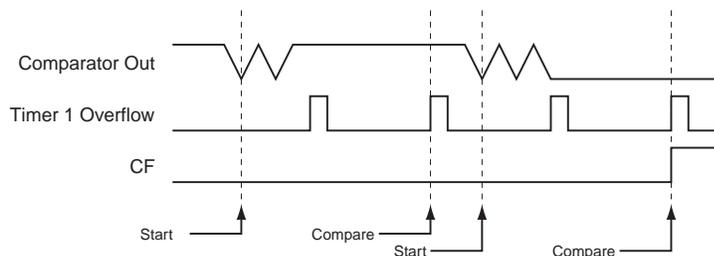
在多种输出值情况下，比较器可通过设置 ACSR 中的 CM 位来设定产生中断。无论何时只要比较器输出与 CM 说明的情况匹配，ACSR 中的比较器中断标志 CF 被置位。标志位可能被软件选取或被用于产生一个中断并且必须被软件清零。

### 19.1 含消抖的比较器中断

每个时钟周期比较器输出都被取样。模拟输入上的状态可能是这样的，比较器输出可能过度跳转。当申请慢速移动模拟输入时这一情况尤其真实。有三种消抖模式被提供用于虑除这一噪声。在消抖模式下，比较器使用 Timer 1 来调节它的采样时间。当相应的跳转发生时，比较器会等待直至两个 Timer 1 溢出值在对输出重新取样前出现。如果新的采样适合预期的值，CF 被置位。否则，事件被忽略。滤波器可以通过调整 Timer 1 的 time-out 周期来调谐。这是因为 Timer 1 是自由运转的，消抖器必须等待两个溢出来保证采样延迟至少有一个 time-out 周期。因此，在最初的边缘事件之后，中断可能在 1 到 2 个 time-out 周期之后发生见 [Figure 19-1 on page 53](#)。

在默认状态下空闲模式中比较器是被失效的。为了在空闲模式下允许比较器运行，ACSR 中的 CIDL 位必须被置位。当 CIDL 被置位，如果比较器中断被使能，比较器可以用于将 CPU 从空闲模式中唤醒。在下电模式下比较器始终被失效。

**Figure 19-1.** 消抖下的负边沿



**Table 19-1.** ACSR – 模拟比较器控制 & 状态寄存器

ACSR = 97H		复位值 = XXX0 0000B						
非位可寻址								
	–	–	CIDL	CF	CEN	CM3	CM1	CM0
Bit	7	6	5	4	3	2	1	0

符号	功能																																				
CIDL	比较器空闲使能。如果 CIDL = 1 比较器将在空闲模式下持续运作。如果 CIDL = 0 比较器在空闲模式下被下电。比较器在下电模式下始终被关闭。																																				
CF	比较器中断标志。当比较器输出遇到这样的情况，以 CM [2:0] 位说明并且 CEN 被置位，这一位置位。这一标志必须由软件清零。中断可以通过置位 / 清零 IE 的 bit6 来使能 / 失效。																																				
CEN	比较仪使能。置位这一位使能比较仪。清零这一位将迫使比较仪输出低电平并且防止置位 CF 引发进一步事件。当 CEN = 1 模拟输入输入引脚，P1.0 以及 P1.1，它们的数字输入失效。																																				
CM [2:0]	比较仪中断模式 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>CM2</th> <th>CM1</th> <th>CM0</th> <th>中断模式</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>负（低）电平</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>正跳变</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>套牢跳变<sup>(1)</sup></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>负边缘跳变<sup>(1)</sup></td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>负沿</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>套牢</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>负边缘跳变<sup>(1)</sup></td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>正（高）电平</td> </tr> </tbody> </table>	CM2	CM1	CM0	中断模式	0	0	0	负（低）电平	0	0	1	正跳变	0	1	0	套牢跳变 <sup>(1)</sup>	0	1	1	负边缘跳变 <sup>(1)</sup>	1	0	0	负沿	1	0	1	套牢	1	1	0	负边缘跳变 <sup>(1)</sup>	1	1	1	正（高）电平
CM2	CM1	CM0	中断模式																																		
0	0	0	负（低）电平																																		
0	0	1	正跳变																																		
0	1	0	套牢跳变 <sup>(1)</sup>																																		
0	1	1	负边缘跳变 <sup>(1)</sup>																																		
1	0	0	负沿																																		
1	0	1	套牢																																		
1	1	0	负边缘跳变 <sup>(1)</sup>																																		
1	1	1	正（高）电平																																		

Note: 1. Debouncing 模式要求使用定时器 1 来产生取样延迟。

## 20. 可编程看门狗定时器

在定时器溢出前，如果软件未能定时供给，将触发一个系统复位，而可编程看门狗定时器 (WDT) 可以保护系统以免因此产生误操作。默认状态下 WDT 计 CPU 时钟周期。SFR WDTCON 中的预置数分频位，PS0, PS1 以及 PS2 用来将看门狗定时器的周期设置为从 16K 到 2048K 时钟周期。定时器预置数分频器同样可用于延长 time-out 周期 (见 Table 9-2 on page 13)。在上电模式下以及复位状态下 WDT 被失效。如果 WDT 计时结束而没有被服务，一个内部 RST 脉冲将被产生来复位 CPU。可用 WDT 周期选择见 Table 20-1。

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

WDT 预置数分频位			周期 <sup>(1)</sup> (时钟周期)
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 计时结束周期由系统时钟频率决定。

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

看门狗定时器由 14 位定时器包含 7 位可编程预置数分频器组成。向 WDTRST 寄存器写队列 1EH/E1H 可以使能定时器。当 WDT 被使能，WDTCON 中的 WDTEN 位将被置 1。为了避免 WDT 在溢出时产生复位，看门狗饲养队列必须在计时周期结束之前写入 WDTRST。为了喂养看门狗，两条写指令必须被接连成功地执行。在两条写指令之间，SFR 读是被允许的，但是写不允许。指令必须把 1EH 移动到 WDTRST 寄存器然后 1EH 到 WDTCON 寄存器。不正确的喂养或使能队列将产生直接的看门狗复位。喂养或使能看门狗定时器的程序队列如下：

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

### 20.1 软件复位

通过对 WDTRST SFR 写软件复位队列 5AH/A5HA，AT89LP213/214 完成一个软件复位。The WDT 不需要被使能以产生软件复位。一个正常的软件复位会置位 WDTCON 中的 SWRST 标志。然而，任何时候如果有错误的队列写到 WDTRST (例，除了 1EH/E1H 或 5AH/A5H)，一个软件复位将立即产生，SWRST 以及 WDTOVF 标志都会被置位。在这种方式下，一个故意的软件复位和一个误操作产生的复位是有区别的。产生软件复位的程序队列如下：



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			
非位可寻址								
	PS2	PS1	PS0	WDIDLE	–	SWRST	WDTOVF	WDTEN
Bit	7	6	5	4	3	2	1	0
<b>符号</b>	<b>功能</b>							
PS2 PS1 PS0	看门狗定时器 (WDT) 的预置数分频器。当这所有三位都被清零时，看门狗定时器有一个 16K 时钟周期的常规周期。当这所有三位都被置 1 时，这一常规周期为 2048K 时钟周期。							
WDIDLE	在空闲模式下失效 / 使能看门狗定时器。当 WDIDLE = 0 时，WDT 在空闲模式下持续计数。当 WDIDLE = 1 时，WDT 在空闲模式下冻结。							
SWRST	软件复位标志。当通过对 WDTRST 写 5AH/A5H 队列产生一个软件复位时，它被置位。同样，当一个不正确的队列写到 WDTRST 中时，它必须被软件清零。							
WDTOVF	看门狗溢出标志。当 WDT 定时器溢出产生 WDT 复位时，它被置位。同样，当一个不正确的队列写到 WDTRST 中时，它必须被软件清零。							
WDTEN	看门狗使能标志位。这一位是只读位，它反映了 WDT 的状态（不管它是不是在运转）。任何复位之后 WDT 都被失效，并且必须通过写 1EH/E1H 到 WDTRST 来重新使能。							

**Table 20-3. WDTRST – 看门狗复位寄存器**

WDTCON 地址 = A6H					(只写)			
非位可寻址								
	–	–	–	–	–	–	–	
Bit	7	6	5	4	3	2	1	0
<p>WDT 通过写 1EH/E1H 队列到 WDTRST SFR 中被使能。通过读 WDTCON 中的 WDTEN 位可以核对目前的状态。为了阻止 WDT 复位器件，在 time-out 间隔期间同样的 1EH/E1H 队列必须被写到 WDTRST 中。通过写 5AH/A5H 队列到 WDTRST 中可以产生一个软件复位。</p>								

## 21. 指令机摘要

AT89LP213/214 是完全二进制的，与 MCS-51 指令集兼容。AT89LP213/214 以及标准 8051 之间的区别在于执行一条指令所需要的循环数。AT89LP213/214 中的指令可能需要 1 个，2 个，3 个或 4 个时钟周期来完成。大多数指令的执行时间可以用 Table 21-1 来估算。

**Table 21-1. 指令执行时间及例外**

一般指令类型	循环数公式			
大多数算法，逻辑，位及传输指令	# bytes			
分支及调用	# bytes + 1			
单字节间接 (i.e. ADD A, @Ri, etc.)	2			
RET, RETI	4			
MOVC	3			
MOVX	4			
MUL	2			
DIV	4			
INC DPTR	2			
算法	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

**Table 21-1. 指令执行时间及例外 (Continued)**

算法	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
逻辑	Bytes	Clock Cycles		Hex Code
		8051	AT89LP	
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

**Table 21-1.** 指令执行时间及例外 (Continued)

数据传输	Bytes	Clock Cycles		Hex Code
		8051	AT89LP	
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
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

**Table 21-1. 指令执行时间及例外 (Continued)**

位操作	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
分支	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
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 <sup>(1)</sup>	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 <sup>(1)</sup>	2	–	2	A5 00

Note: 1. 漏掉的指令是这一套指令系统的扩展部分。

## 22. 片上调试系统

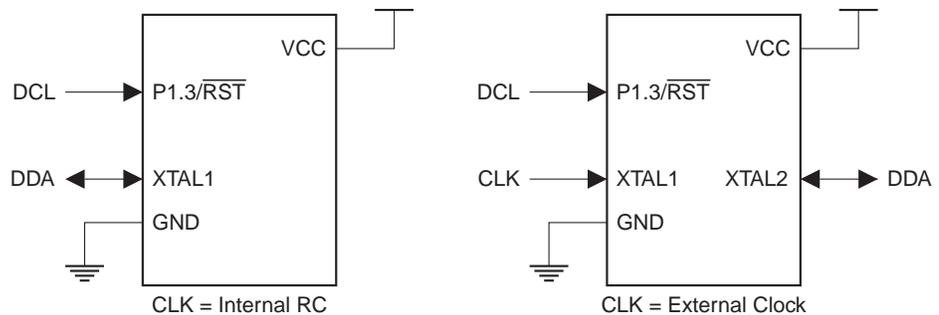
AT89LP213/214 片上调试 (OCD) 系统采用双线串行接口来控制程序流，读写和修改系统状态；并且编程控制非易失性存储器。OCD 系统有以下特性：

- 完全程序流控制
- 对所有内部 SFR 和数据存储器进行读 - 修改 - 写访问
- 四个硬件程序地址断点
- 有限个程序软件断点，在使用 BREAK 指令时
- 在堆栈溢出或下溢时 Break
- 看门狗溢出时 Break
- 无中断操作
- 编程控制非易失性存储器

### 22.1 物理接口

片上调试系统采用一个双线同步串行接口，来建立目标设备与控制仿真系统之间的通信。OCD 接口由两个用户引脚来控制。OCD 由  $\overline{\text{RST}}$  使能端清零开启。当 OCD 开启后， $\overline{\text{RST}}$  端口引脚设置成调试时钟的输入口 (DCL)。根据时钟源的选择将 XTAL1 或 XTAL2 引脚设置成一个调试数据 (DDA) 的双向数据线。如果选择外部时钟，则 XTAL2 设置成 DDA。如果选择内部 RC 振荡器，则 XTAL1 设置成 DDA。OCD 设备连接见图 Figure 22-1。当 OCD 开启时，所使用的接口类型由 OCD 接口的 Select User Fuse 来选择。有两个类型，一个是常规双线接口 (TWI)，另一个是快速双线接口 (FTWI)。这是在开启 OCD 的同时用户需要设置的，用户通过编程来控制熔丝为他们的调试系统进行正确的配置。(见“用户配置引脚” on page 69)。

Figure 22-1. AT89LP213/214 片上调试连接



当设计一个要使用片上调试的系统时，下面的资料必须认真研究才可以作出正确操作：

- P1.3/ $\overline{\text{RST}}$  不可直接连接到  $V_{CC}$ ，而且任何与  $\overline{\text{RST}}$  连接的外部电容必须被移除。
- 所有外部 reset 源必须被移除。
- XTAL1 或 XTAL2 上的石英晶体和所有电容必须被移除，并且一个外部时钟信号必须在 XTAL1 被驱动，如果用户不想使用内部 RC 振荡器。因此，一些仿真器系统提供用户可配置时钟。

## 22.2 软件断点

AT89LP213/214 微控制器有一个 BREAK 指令，用来实现软件中的程序存储器断点。一个软件断点可以通过在程序代码中写入 BREAK 指令来手动插入。一些仿真器系统允许自动插入 / 删除软件断点。每次一个软件断点改变时，闪存都必须重新编程设置。频繁的软件断点的插入 / 删除会降低非易失性存储器的数据保持能力。用于调试的设备不可以连接到终端用户。当 OCD 禁用时，BREAK 指令被处理成双周期 NOP。

## 22.3 片上调试的局限性

AT89LP213/214 是一个低成本，低引脚输出但是多功能的微控制器，它能够在它的有限的 I/O 引脚上实现多种功能。一些设备功能需要牺牲以为片上调试提供资源。片上调试系统有以下局限性：

- 调试时钟引脚 (DCL) 与端口引脚 P1.3 和外部 Reset( $\overline{\text{RST}}$ ) 位于相同的引脚。因此，OCD 启动时，P1.3 和外部 reset 源都不能被仿真。
- 调试数据引脚 (DDA) 位于 XTAL1/P3.2 或 XTAL2/P3.3 引脚。因此，在调试期间不支持晶体振荡器。用户必须选择内部 RC 振荡器或者外部时钟源来提供系统时钟。OCD 启动时，晶体振荡器设备将默认为外部时钟模式。
- 在调试期间使用内部 RC 振荡器时，DDA 位于 XTAL1/P3.2 引脚。在这种模式下  $\overline{\text{INT0}}$  功能不能被仿真。
- 在调试期间使用外部时钟时，DDA 位于 XTAL2/P3.3 引脚，而且系统时钟驱动 XTAL1/P3.2。在这种模式下  $\overline{\text{INT0}}$ ,  $\overline{\text{INT1}}$  和 CLKOUT 功能不能被仿真。
- AT89LP213/214 不支持 In-Application 编程，所以此设备必须在调试期间改变代码之前进行 reset。这包括软件断点的插入和删除。
- 当通过看门狗产生一个中断时，看门狗定时器的状态将不会被 reset。一个 OCD reset 指令被发送到设备不能优先于继续正常执行正确的看门狗定时器的指令。

## 23. 闪存编程

Atmel 的 AT89LP213/214 微控制器拥有 2KB 的片上系统可编程 Flash 程序存储器。系统内编程 (ISP) 允许位于终端系统内部的微控制器的编程和重新编程。仅通过一个 4 线的 SPI 接口，In-System 程序员就可以对所有的片上非易失性存储器进行重新编程，与 AT89LP213/214 微控制器进行串行通信。这样，在改进实验室和升级软件及相关参数时，就回节省时间和财力。AT89LP213/214 的 ISP 接口包括以下特性：

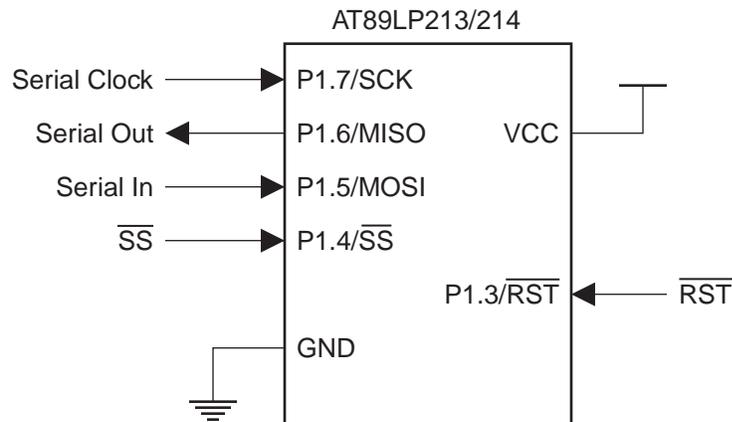
- 四线 SPI 编程接口
- 低电平复位进入编程
- 从属选择允许在一些接口并联设备
- 用户信号队列
- 可变的页编程
- 行擦除能力
- 写入页时自动擦除命令
- 程序状态寄存器

要得到更多的有关 In-System Programming 的详细信息，请查阅 应用软件注释“AT89LP In-System Programming 说明书”。

### 23.1 物理接口

In-System Programming 采用 AT89LP213/214 微控制器的 SPI 接口。SPI 是一个全双工同步串行接口，由四根线组成：串行时钟 (SCK)，主入 / 从出 (MISO)，主出从入 (MOSI) 和一个低电平选通的从属选择 ( $\overline{SS}$ )。当对一个 AT89LP213/214 设备编程时，程序员通常相当于 SPI 主控制器，而目标系统相当于 SPI 从控制器。要进入或保持 In-System Programming 模式，设备的 reset 线 ( $\overline{RST}$ ) 必须保持低电平。再加上 VCC 和 GND，一个 AT89LP213/214 微控制器能编程设置为最少有七处连接，见图 Figure 23-1。

Figure 23-1. In-System Programming 设备连接



In-System Programming 接口是对 AT89LP213/214 微控制器外部编程的唯一方式。ISP 接口可以用于 in-system 和单机串行程序器对设备的编程。ISP 接口只需要 SCK，不需要其他任何时钟，而且不受系统时钟频率的制约。在 In-System programming 期间，目标设备的系统时钟源可以正常运行。

要设计一个 In-System Programming 可用的系统时，下面的资料必须认真研究才可以作出正确操作：

- ISP 接口采用 SPI 时钟模式 0 (CPOL = 0, CPHA = 0)，专用最高频率为 5 MHz。
- 只有当 AT89LP213/214 的 reset 线 ( $\overline{\text{RST}}$ ) 处于低电平时，它才能进入编程模式。要简化这一操作，就要求 In-System programmer 来控制目标 reset。要避免错误发生，In-System programmer 就要保持整个目标系统在编程周期内复位。在复位期间，目标系统不能试图驱动四根 SPI 线。
- $\overline{\text{RST}}$  输入可以被禁用以获得额外的 I/O 引脚。在这些情况下， $\overline{\text{RST}}$  引脚通常将在 power up 期间作用为一个 reset。要进入编程  $\overline{\text{RST}}$  引脚必须在上电复位 (POR) 结束之前被驱动。在 POR 完成之后，设备将保持在 ISP 模式，直到  $\overline{\text{RST}}$  跳至高电平。一旦 ISP 的初始化结束，目标设备就会周期性的断电，然后在通电进入下一阶段。
- 如果 ISP 开启，则在 reset 期间  $\overline{\text{SS}}$  引脚不能悬空。
- ISP 使能引脚必须置位，允许在任何 reset 期间编程。如果 ISP 引脚禁用，则只能在 POR 访问 ISP。

## 23.2 存储器机构

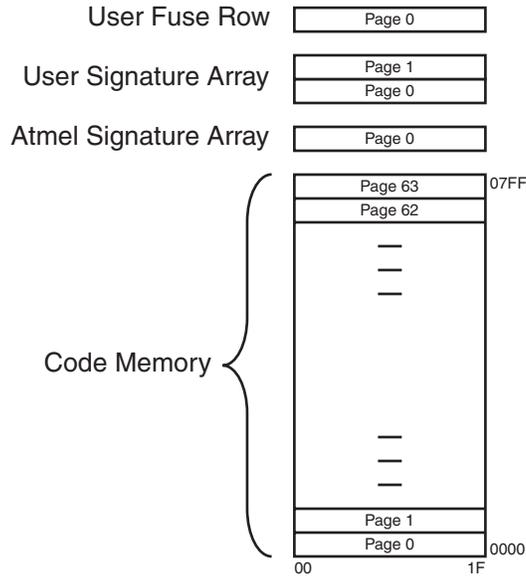
AT89LP213/214 提供 2K 字节的 ISP 非易失性闪存代码存储器。此外，该设备还包含一个 64-byte 的用户信号序列和一个 32-byte 的只读 Atmel 信号序列。存储器机构详见图表 Table 23-1 和 Figure 23-2。存储器被拆分成页，每页 32 bytes。一个信号的读和写指令只访问存储器中的一个信号页。每个存储器类型位于它自己的地址空间，并且由指定对某个存储器进行访问的指令对其进行访问。然而，所有的存储器类型享有同样大小的页。

用户配置引脚在存储器中映射成一行，每种类型代表一个引脚。从一个程序断点，引脚被设置成与常规代码字节相同，除非它们不受到芯片擦除的影响。在任何时候向引脚行的正确位置写入 00h，引脚就可以起用。然而，要禁用引脚，也就是将它置为 FFh，整个引脚行就必须被擦除，然后重新写入程序。程序员需要将所有引脚的状态读入一个暂时的位置，如果有需要禁用的引脚就修改该引脚的值，然后通过自动擦除指令将临时数据写入一个引脚。对锁定位的处理与引脚采取同种方式，除非他们只能通过 Chip Erase 来擦除。

Table 23-1. 代码存储器大小

Device #	Code Size	Page Size	# Pages	Address Range
AT89LP213	2K bytes	32 bytes	64	0000H - 07FFH
AT89LP214	2K bytes	32 bytes	64	0000H - 07FFH

**Figure 23-2.** AT89LP213/214 存储器机构



## 23.3 指令格式

编程指令包括一个 **opcode** 字节，两个地址字节，还有 0，或者更多数据字节。除此之外，所有指令信息包必须开始于一个两个字节的同步码设置为 **AAH** 和 **55H**。同步码使程序接口很难发出意外的错误指令，从而增加了抗噪能力。Figure 23-3 on page 66 是一个指令序列的单一流程图。

一个样本指令信息包见图 Figure 23-4 on page 66。SS 引脚定义信息帧。在指令的第一个字节发送之前 SS 必须置低电平，而在最后一个字节发送之后，它将返回高电平。直到 SS 返回高电平，该指令才完成。指令字节在 MOSI 串行发送。数据输出字节在 MISO 被串行接收。当最后字节发送以后将 SS 返回高电平，就可以支持不同长度的信息包。有时指令字节有一个 **don't care** 值。在一个信息包中间的 **Don't care** 字节必须发送。在一个信息包末端的 **Don't care** 字节可以被忽略。

页导向指令通常包括一个完全 16 位的地址。高指令位选页，低指令位在该页中选择字节。AT89LP213/214 分配 5 位为字节地址，6 位为页地址。将被访问的页通常由发送来的页地址来确定。字节地址指定第一个数据字节的起始地址。每发送一个数据字节之后，字节地址增加，指向下一个数据字节。这样，一个页指令就可以在该页中线性的访问每一个字节。如果字节地址增加到超过此页中的最后一个字节，字节地址就会再指向这一页中的第一个字节。当载入字节到页缓冲器中时，之前写溢出的字节就会导致数据出错。

可用指令摘要见 Table 23-2 on page 67.

Figure 23-3. 指令序列流程图

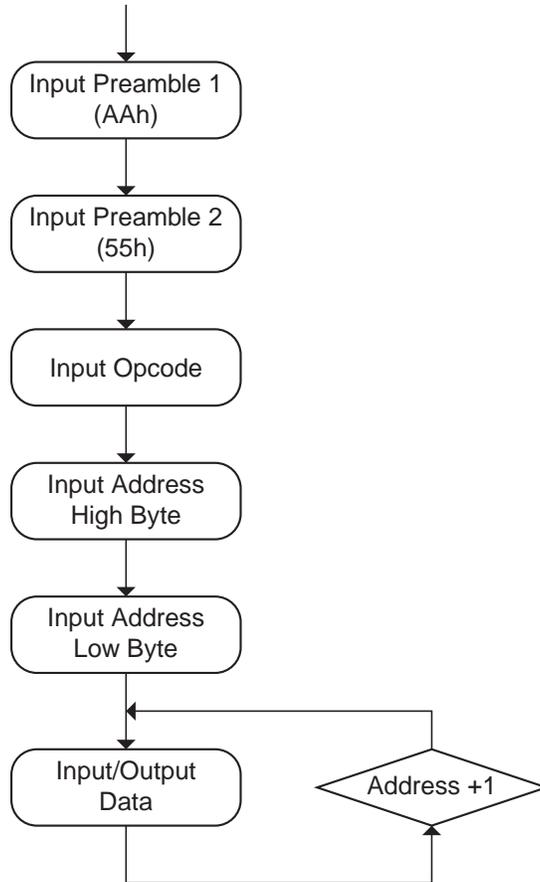
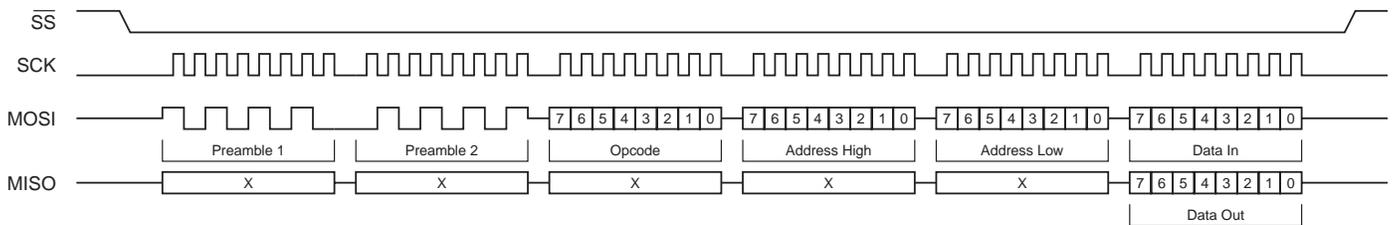


Figure 23-4. ISP 指令信息包



**Table 23-2.** 程序指令摘要

Command	Opcode	Addr High	Addr Low	Data 0	Data n
Program Enable <sup>(1)</sup>	1010 1100	0101 0011	–	–	–
Chip Erase	1000 1010	–	–	–	–
Read Status	0110 0000	xxxx xxxx	xxxx xxxx	Status Out	
Load Page Buffer <sup>(2)</sup>	0101 0001	xxxx xxxx	xxxb bbbb	DataIn 0 ... DataIn n	
Write Code Page <sup>(2)</sup>	0101 0000	xxxx xaaa	aaab bbbb	DataIn 0 ... DataIn n	
Write Code Page with Auto-Erase <sup>(2)</sup>	0111 0000	xxxx xaaa	aaab bbbb	DataIn 0 ... DataIn n	
Read Code Page <sup>(2)</sup>	0011 0000	xxxx xaaa	aaab bbbb	DataOut 0 ... DataOut n	
Write User Fuses <sup>(2)(3)(4)</sup>	1110 0001	0000 0000	000b bbbb	DataIn 0 ... DataIn n	
Write User Fuses with Auto-Erase <sup>(2)(3)(4)</sup>	1111 0001	0000 0000	000b bbbb	DataIn 0 ... DataIn n	
Read User Fuses <sup>(2)(3)(4)</sup>	0110 0001	0000 0000	000b bbbb	DataOut 0 ... DataOut n	
Write Lock Bits <sup>(2)(3)(5)</sup>	1110 0100	0000 0000	000b bbbb	DataIn 0 ... DataIn n	
Read Lock Bits <sup>(2)(3)(5)</sup>	0110 0100	0000 0000	000b bbbb	DataOut 0 ... DataOut n	
Write User Signature Page <sup>(2)</sup>	0101 0010	xxxx xxxx	xaab bbbb	DataIn 0 ... DataIn n	
Write User Signature Page with Auto-Erase <sup>(2)</sup>	0111 0010	xxxx xxxx	xaab bbbb	DataIn 0 ... DataIn n	
Read User Signature Page <sup>(2)</sup>	0011 0010	xxxx xxxx	xaab bbbb	DataOut 0 ... DataOut n	
Read Atmel Signature Page <sup>(2)(6)</sup>	0011 1000	xxxx xxxx	xxxb bbbb	DataOut 0 ... DataOut n	

- Notes:**
1. 程序使能时必须时进入编程模式之后发出的第一条指令。
  2. 1 到 32 之间的数据字节的任何数字可能被读 / 写。内部地址在每个字节之间增加。
  3. 每一字节地址选择一个引脚或锁定位。数据字节必须为 00h 或 FFh。
  4. 见 [Table 23-5 on page 69](#) : 引脚定义。
  5. 见 [Table 23-4 on page 68](#) : 锁定位定义。
  6. **Atmel 信号字节**

```

AT89LP213:  Address  00H = 1EH
              01H = 27H
              02H = FFH
AT89LP214:  Address  00H = 1EH
              01H = 28H
              02H = FFH
    
```

**7. 信号密码 :**

- a: Page Address Bit
- b: Byte Address Bit
- x: Don't Care Bit



## 23.4 状态寄存器

存储器的当前状态可能通过读状态寄存器来访问。状态寄存器见 Table 23-3.

**Table 23-3.** 状态寄存器

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	$\overline{\text{LOAD}}$	SUCCESS	$\overline{\text{WRTINH}}$	$\overline{\text{BUSY}}$

标志	功能
$\overline{\text{LOAD}}$	装载标志。由装载页缓冲器指令将此位清零，由下一个存储器写操作将此位置 1。这一标志表示页缓冲器通过页缓冲器指令已经载入数据。
SUCCESS	完成标志。在一个编程周期的开始清零，只有在没有收到节能检波器的中断时完成一个编程周期的情况下，此位才能置位。
$\overline{\text{WRTINH}}$	写抑制标志。如果 $V_{CC}$ 降低到要求的编程电压以下，编程将为内向的，在这种情况下，此位由节能检波器清零。如果一个 BOD 时间发生在编程期间，则 SUCCESS 标志将在这个周期结束之后仍保持低电平。 $\overline{\text{WRTINH}}$ 为低电平也会将 $\overline{\text{BUSY}}$ 拉为低电平。
$\overline{\text{BUSY}}$	繁忙标志。当存储器处于编程期间或当前执行写操作时，此位清零。

## 23.5 $\overline{\text{DATA}}$ 查询

The AT89LP213/214 执行  $\overline{\text{DATA}}$  查询，表示编程周期结束。当设备繁忙时，任何对于最后一个写入字节的读操作将会返回 MSB 补偿的数据字节。一旦编程周期结束，就可获得一个真正的值。在擦除阶段，数据被认为是 FFH，并且  $\overline{\text{DATA}}$  查询将返回 7FH。当在一页中写入很多字节时， $\overline{\text{DATA}}$  值将为编程开始前最后一个被写入的字节，而不是在此页中最高物理地址中被写入的字节。

## 23.6 Flash 安全性

AT89LP213/214 为 Flash 代码存储器的安全性提供两个锁定位 (Lock bits)。锁定位可设置为不被编程 (FFh) 或者被编程 (00h)，获得的保护标准见 Table 23-4。锁定位只能由 Chip Erase 来擦除 (置位为 FFh)。锁定位模式 2 禁止所有存储器空间的编程，包括用户信号序列 (User Signature Array) 和用户信号引脚 (User Configuration Fuses)。用户引脚必须在启动锁定位模式 2 或 3 之前编程设置。锁定位模式 3 执行模式 2，并且还阻止从代码存储器进行读操作，然而对于 User Signature Array, Atmel Signature Array 和 User Configuration Fuses 的读操作仍然允许。

**Table 23-4.** 锁定位保护模式

程序锁定位 (by 地址)			保护模式
模式	00h	01h	
1	FFh	FFh	无程序锁定特性
2	00h	FFh	进一步的对 Flash 编程被禁止
3	00h	00h	进一步的对 Flash 编程被禁止，并且校验 (读) 也被禁止；OCD 禁用

## 23.7 用户配置引脚

AT89LP213/214 有 19 个用于设备配置的用户引脚。每个引脚在用户引脚行的不同地址分别被访问，见 Table 23-5。可通过对引脚的位置编程输入 00h 来对此引脚清零。对引脚的位置编程输入 FFh 会使该引脚保持之前的状态。要想将一引脚置位（置成 FFh），此引脚行必须被擦除，然后通过自动擦除指令，使用 Fuse Write 来对其进行重新编程。所有引脚的默认状态为 FFh。

**Table 23-5.** 用户配置引脚定义

地址	Fuse 名称	描述
00 – 01h	时钟源 – CS[0:1] <sup>(2)</sup>	为系统时钟选择源。
		CS1 CS0 已选源
		00h 00h 晶体震荡器 (XTAL)
		00h FFh 保留
		FFh 00h XTAL1 上的外部时钟 (XCLK)
FFh FFh 内部 RC 震荡器 (IRC)		
02 – 03h	启动时间 – SUT[0:1]	为 POR/BOD/PWD wake-up 时期选择时间溢出延时：
		SUT1 SUT0 选择时间溢出
		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	Reset 引脚使能端 <sup>(3)</sup>	FFh: $\overline{\text{RST}}$ 引脚功能 reset 00h: RST 引脚功能：通用 I/O 口
05h	节能检波器使能端	FFh: 节能检波器启用 00h: 节能检波器禁用
06h	片上调试使能端 $\overline{\text{Enable}}$	FFh: 片上调试禁用 00h: 片上调试启用
07h	ISP 使能端 <sup>(3)</sup>	FFh: In-System Programming 开启 00h: In-System Programming 禁用 (只可用于 POR)
08 – 0FH	RC 震荡器频率调节器 [0:7]	调节内部 RC 震荡器的频率。
10H	用户信号编程	FFh: 用户信号编程禁用 00h: 用户信号编程启用
11H	三态端口	FFh: I/O 端口起始于 input-only 模式 (三态)，在 reset 之后 00h: I/O 端口起始于准双向模式，在 reset 之后
12H	OCD 接口选择	FFh: 标准双线接口 00h: 快速双线接口

- Notes:
1. 所有的引脚的默认状态为 FFh。
  2. 在一个设备 POR 之后，对于这些引脚的改变才会起作用。
  3.  $\overline{\text{RST}}$  跳为高电平 ISP 阶段终止之后，对于这些引脚的改变才会起作用。

## 23.8 编程接口定时

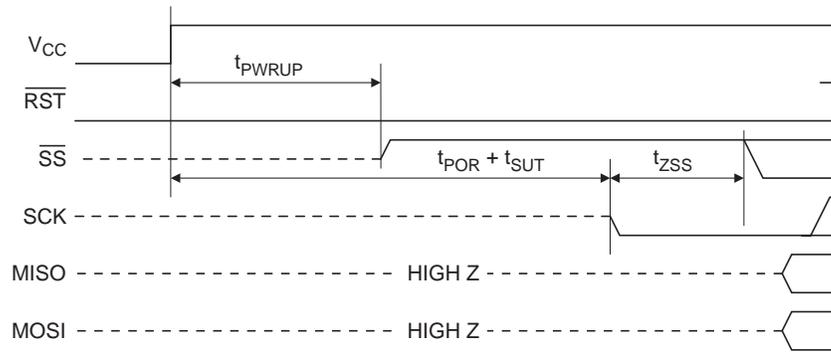
这一部分详细说明了通用系统的定时序列和对于插入或退出的约束，其中的插入和退出是指在 ISP 期间插入或推出 In-System Programming 中，以及与串行外部接口有关的参数。以下波形的通用定时序列在 [23.8.6 "Timing Parameters" on page 74](#) 部分列出。

### 23.8.1 上电序列

执行此序列，在上电之后立即进入编程模式。在  $\overline{\text{RST}}$  引脚禁用或者 ISP 引脚禁用时，这是唯一进入编程模式的方法。（见 [Section 10.3 "外部复位" on page 15](#)）。

1. VCC 和 GND 引脚之间的供应功率。 $\overline{\text{RST}}$  要保持低电平。
2. 等待至少  $t_{\text{PWRUP}}$  时间，并将  $\overline{\text{SS}}$  拉高。
3. 等待至少  $t_{\text{SUT}}$  时间，等待内部上电复位完成。 $t_{\text{SUT}}$  的值将取决于当前设备的配置。
4. 开始编程。

**Figure 23-5.** 串行编程上电序列

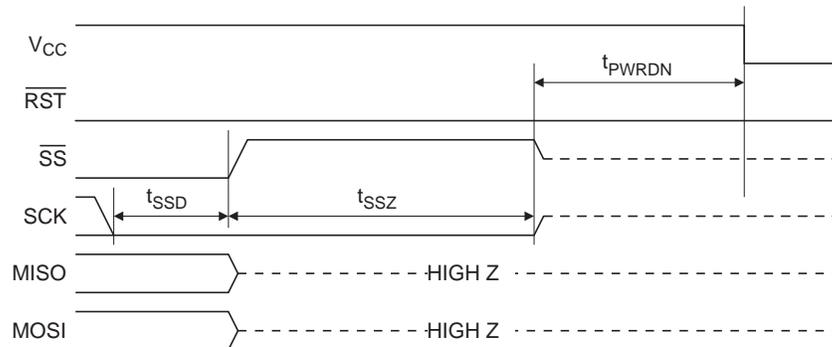


### 23.8.2 下电序列

编程后执行此序列使设备处于下电状态。

1. 将 SCK 置电平。
2. 等待至少  $t_{\text{SSD}}$  时间，并且将  $\overline{\text{SS}}$  拉高。
3. 三态 MOSI。
4. 等待至少  $t_{\text{SSZ}}$  时间，然后三态  $\overline{\text{SS}}$  和 SCK。
5. 等待时间不超过  $t_{\text{PWRDN}}$  并且关掉 VCC。

**Figure 23-6.** 串行编程下电序列

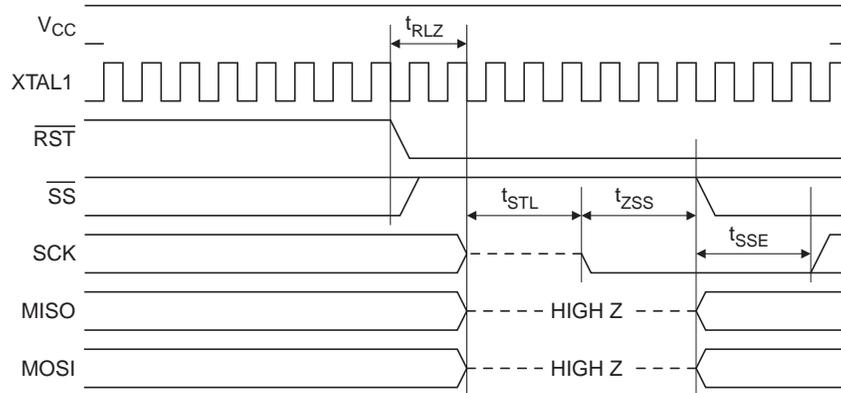


## 23.8.3 ISP 开始序列

当器件已经通过上电复位且已经可以工作时，执行这一序列来退出 CPU 执行模式，且进入 ISP 模式。

1. 驱动  $\overline{\text{RST}}$  为低。
2. 驱动  $\overline{\text{SS}}$  为高。
3. 等待  $t_{\text{RLZ}} + t_{\text{STL}}$ 。
4. 开始可编程部分。

Figure 23-7. 在线可编程 (ISP) 开始序列

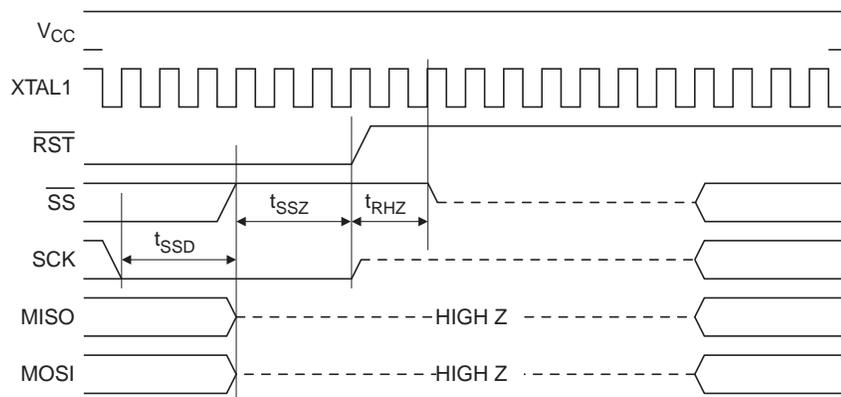


## 23.8.4 ISP 退出时序

执行这一序列来退出 ISP 模式，回复 CPU 执行模式。

1. 驱动 SCK 低。
1. 至少等到  $t_{\text{SSD}}$  并驱动  $\overline{\text{SS}}$  为高。
2. 三态 MOSI。
3. 至少等到  $t_{\text{SSZ}}$  并将  $\overline{\text{RST}}$  拉高。
4. 三态 SCK。
5. 等待  $t_{\text{RHZ}}$  并三态  $\overline{\text{SS}}$ 。

Figure 23-8. 在线可编程 (ISP) 退出时序



Note: 这一页上的波形不是按比例。

### 23.8.5 串行外围设备接口

串行外围设备接口 (SPI) 是一个字节型的全双工同步串行通信通道。在在线编程过程中编程器总是作为 SPI 主机，且目标设备总是作为 SPI 从机。目标设备从 MOSI 接收串行数据并且从 MISO 发送串行数据。编程接口采用了一个标准的有固定数据顺序的 SPI 端口。对在系统编程来说，先发送 MSB，如图 23 - 9 所示。SCK 相位和极性遵循 SPI 时钟模式 0 (CPOL = 0, CPHA = 0)，在 SCK 的上升沿采样位且在 SCK 下降沿输出。详细的时序信息请参考图 23 - 10。

Figure 23-9. ISP 字节序列

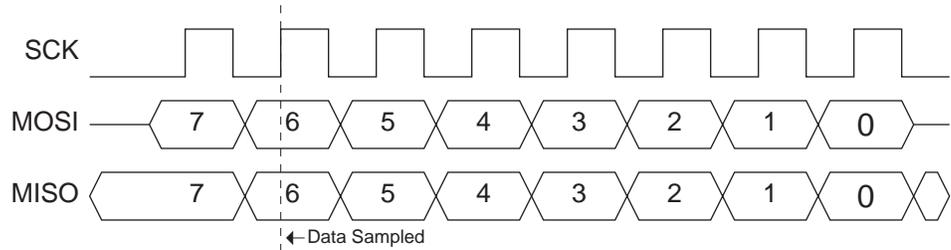
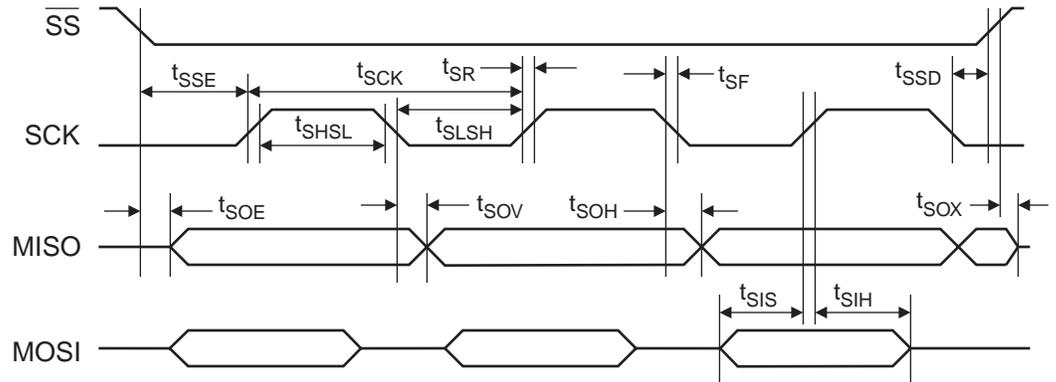


Figure 23-10. S 串行编程接口时序



### 23.8.6 时序参数

表 23 - 6 中列出了图 23 - 5, 图 23 - 6, 图 23 - 7, 图 23 - 8, 图 23 - 10 的时序参数。

**Table 23-6.** P 编程接口时序参数

Symbol	Parameter	Min	Max	Units
$t_{CLCL}$	系统时钟周期	0	60	ns
$t_{PWRUP}$	上电到 $\overline{SS}$ 高电平时间	10		$\mu$ s
$t_{POR}$	上电复位时间		100	$\mu$ s
$t_{PWRDN}$	$\overline{SS}$ 三态下电时间 f		1	$\mu$ s
$t_{RLZ}$	$\overline{RST}$ 低到 I/O 三态	$t_{CLCL}$	$2 t_{CLCL}$	ns
$t_{STL}$	$\overline{RST}$ 低设置时间	100		ns
$t_{RHZ}$	$\overline{RST}$ 高到 $\overline{SS}$ 三态	0	$2 t_{CLCL}$	ns
$t_{SCK}$	串行时钟周期时间	200 <sup>(1)</sup>		ns
$t_{SHSL}$	时钟高时间	75		ns
$t_{SLSH}$	时钟低时间	50		ns
$t_{SR}$	上升时间		25	ns
$t_{SF}$	下降时间		25	ns
$t_{SIS}$	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
$t_{SSE}$	$\overline{SS}$ 使能延迟时间	$t_{SLSH}$		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
$t_{SSZ}$	$\overline{SS}$ 变高后 SCK 保持时间	25		ns
$t_{WR}$	写周期时间	2.5		ms
$t_{AWR}$	带自动擦除时间的写周期	5		ms
$t_{ERS}$	芯片擦除周期	7.5		ms

Note: 1.  $t_{SCK}$  取决于  $t_{CLCL}$ .

## 24. 电器特性

### 24.1 绝对最大额定值 \*

工作温度 .....	-40°C to +85°C
存储温度 .....	-65°C to +150°C
任何引脚的对地电压 .....	-0.7V to +5.5V
最大工作电压 .....	5.5V
DC 输出直流 .....	15.0 mA

\*NOTICE: 超出“绝对最大额定值”列出的设置可能对器件造成永久性的破坏。这只是额定值，在这些情况或者任何别的情况下对器件进行超出操作部分所指出的功能性操作都是不允许的。长期使器件工作在绝对最大额定条件下会影响器件的稳定性。

### 24.2 直流特性

$T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$ ,  $V_{CC} = 2.4\text{V}$  to  $5.5\text{V}$  (除非另有说明)

符号	参数	条件	Min	Max	Units
$V_{IL}$	I 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 <sup>(1)</sup> (Ports 1, 3)	$I_{OL} = 10\text{ mA}$ , $V_{CC} = 2.7\text{V}$ , $T_A = 85^\circ\text{C}$		0.5	V
$V_{OH}$	带弱上拉使能的输出高电压 (Ports 1, 3)	$I_{OH} = -80\ \mu\text{A}$ , $V_{CC} = 5\text{V} \pm 10\%$	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
$V_{OH1}$	带强上拉使能的输出高电压 (Ports 1, 3)	$I_{OH} = -10\text{ mA}$ , $T_A = 85^\circ\text{C}$	$0.9 V_{CC}$		
$I_{IL}$	逻辑 0 输入电流 t (Ports 1, 3)	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$
$I_{TL}$	逻辑 1 过渡电流 (Ports 1, 3)	$V_{IN} = 2\text{V}$ , $V_{CC} = 5\text{V} \pm 10\%$		-750	$\mu\text{A}$
$I_{LI}$	输入漏电流 (Port P1.0, P1.1)	$0 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
$V_{OS}$	比较器输入偏移电压	$V_{CC} = 5\text{V}$		20	mV
$V_{CM}$	比较器输入		0	$V_{CC}$	V
RRST	复位下拉电阻		50	150	K $\Omega$
$C_{IO}$	引脚电容	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
$I_{CC}$	电压供给电流	Active Mode, 12 MHz, $V_{CC} = 5.5\text{V}/3\text{V}$		5.5/3.5	mA
		Idle Mode, 12 MHz, $V_{CC} = 5.5\text{V}/3\text{V}$ P1.0 & P1.1 = 0V or $V_{CC}$		1.2/1	mA
	下电模式 <sup>(2)</sup>	$V_{CC} = 5.5\text{V}$ , P1.0 & P1.1 = 0V or $V_{CC}$		100	$\mu\text{A}$
		$V_{CC} = 3\text{V}$ , P1.0 & P1.1 = 0V or $V_{CC}$		20	$\mu\text{A}$

- Notes: 1. 在稳定的状态下 (无三态),  $I_{OL}$  必须被外部限制如下:  
 每个端口引脚上的最大  $I_{OL}$ : 10 mA  
 所有输出引脚的最大总  $I_{OL}$ : 15 mA  
 如果  $I_{OL}$  超过了测试条件,  $V_{OL}$  可能会超过相关的指标。不能确保引脚漏电流大于所列出的测试条件。  
 2. 最小的下电  $V_{CC}$  2V.

## 24.3 串行外围设备接口时序

**Table 24-1. SPI Master Characteristics**

Symbol	Parameter	Min	Max	Units
$t_{CLCL}$	晶振周期	41.6		ns
$t_{SCK}$	串行时钟周期	$4t_{CLCL}$		ns
$t_{SHSL}$	时钟高时间	$t_{SCK}/2 - 25$		ns
$t_{SLSH}$	时钟低时间	$t_{SCK}/2 - 25$		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

**Table 24-2. 从机 SPI 特性**

符号	参数	Min	Max	Units
$t_{CLCL}$	晶振周期	41.6		ns
$t_{SCK}$	串行时钟周期	$4t_{CLCL}$		ns
$t_{SHSL}$	时钟高时间	$1.5 t_{CLCL} - 25$		ns
$t_{SLSH}$	时钟低时间	$1.5 t_{CLCL} - 25$		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
$t_{SSE}$	从机使能延迟时间	10		ns
$t_{SSD}$	从机禁止延迟时间	0		ns

Figure 24-1. SPI 主机时序 (CPHA = 0)

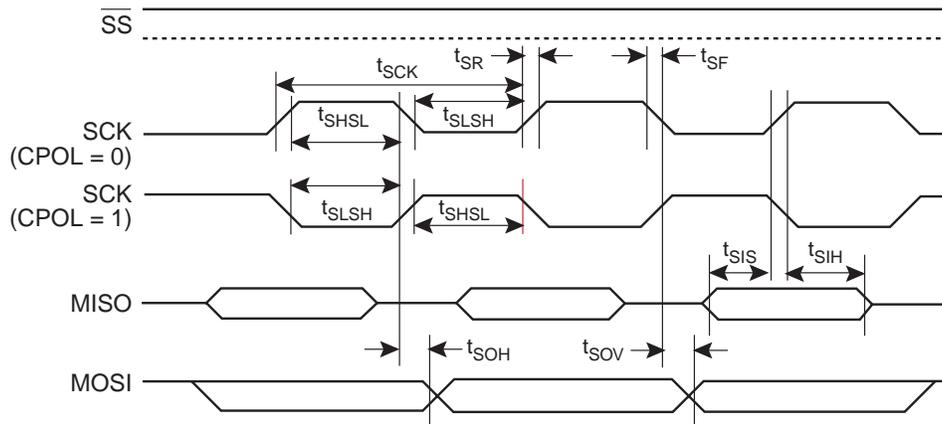


Figure 24-2. SPI 从机时序 (CPHA = 0)

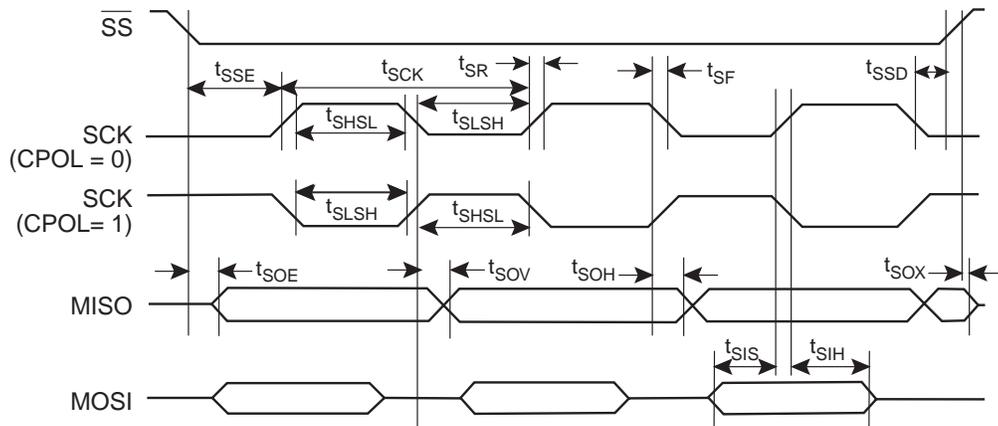


Figure 24-3. SPI 主机时序 (CPHA = 1)

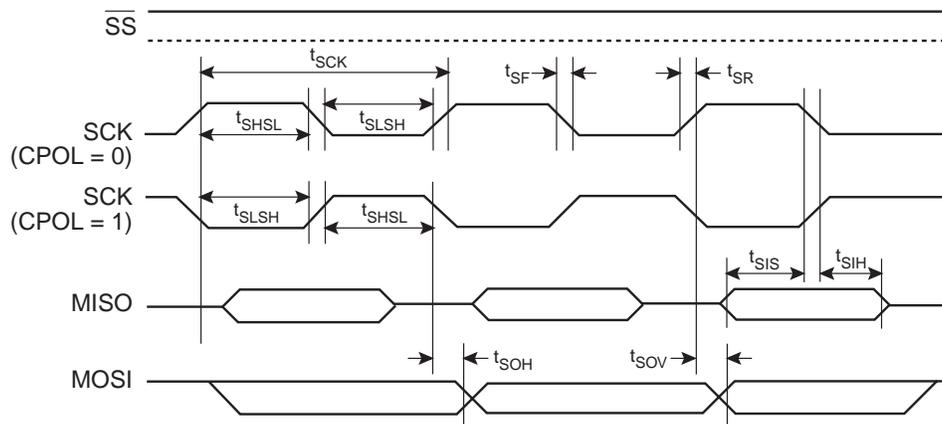
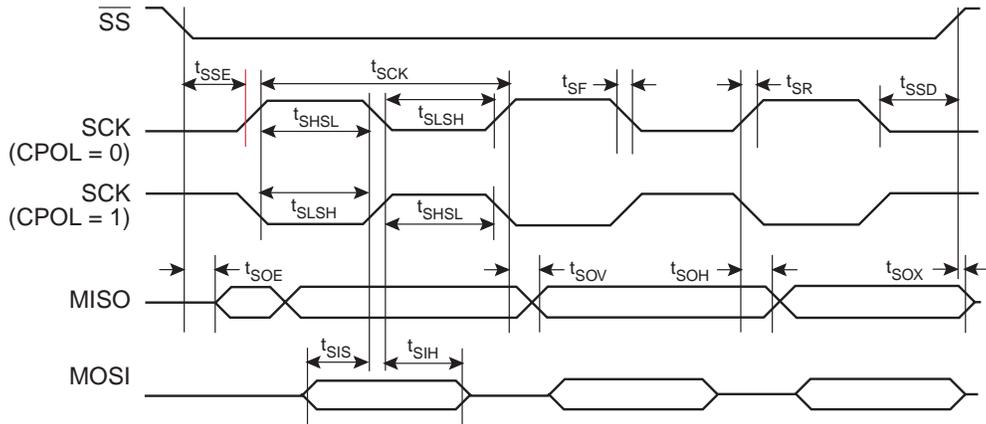
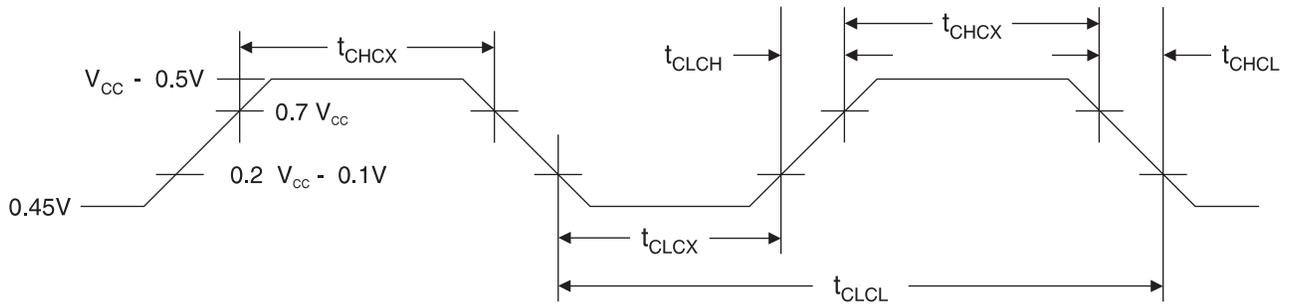


Figure 24-4. SPI S 从机时序 (CPHA = 1)



## 24.4 外部时钟驱动

Figure 24-5. 外部时钟驱动波形



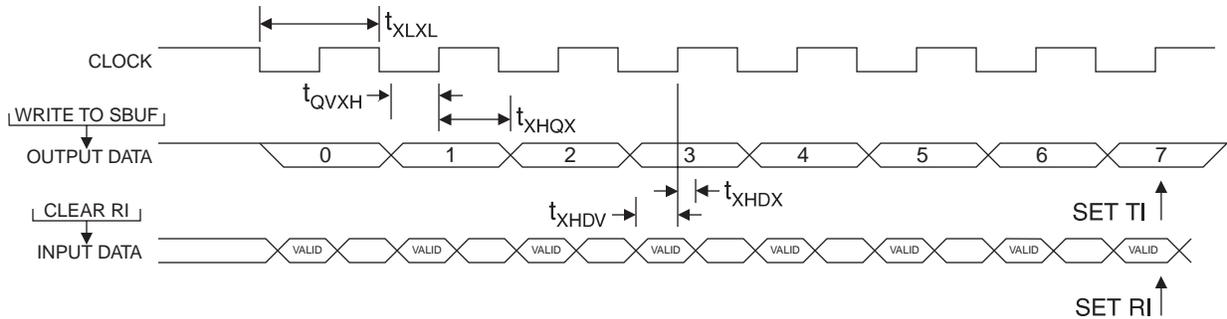
符号	参数	$V_{CC} = 2.4V \text{ to } 5.5V$		Units
		Min	Max	
$1/t_{CLCL}$	晶振频率	0	20	MHz
$t_{CLCL}$	时钟周期	50		ns
$t_{CHCX}$	高电平时间	12		ns
$t_{CLCX}$	低电平时间	12		ns
$t_{CLCH}$	上升时间		5	ns
$t_{CHCL}$	下降时间		5	ns

## 24.5 串行端口时序：移位寄存器测试模式

T 此表中的值对  $V_{CC} = 2.4V$  到  $5.5V$  有效且负载电容 =  $80\text{ pF}$ 。

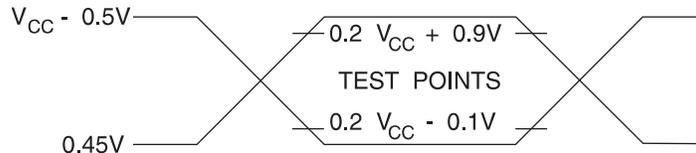
符号	参数	可变晶振		Units
		Min	Max	
$t_{XLXL}$	串行端口时钟周期	$2t_{CLCL} - 15$		$\mu\text{s}$
$t_{QVXH}$	输出数据设置到时钟上升沿	$t_{CLCL} - 15$		ns
$t_{XHQX}$	时钟上升沿后输出数据保持	$t_{CLCL} - 15$		ns
$t_{XHDX}$	时钟上升沿后输入数据保持	0		ns
$t_{XHDX}$	I 输入数据有效到时钟上升沿	15		ns

Figure 24-6. 移位寄存器模式时序波形



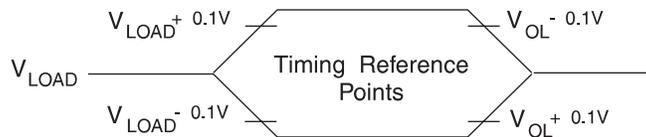
## 24.6 测试条件

### 24.6.1 交流测试输入 / 输出波形 <sup>(1)</sup>



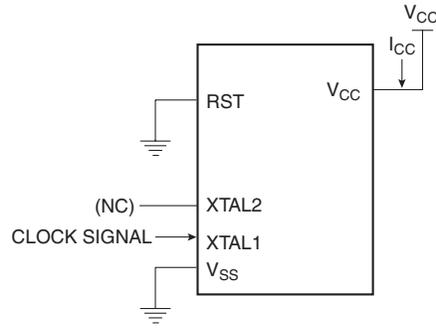
Note: 1. 测试中的交流输入由  $V_{CC} - 0.5V$  表示逻辑“1”和  $0.45V$  表示逻辑“0”来驱动。时序测量是  $V_{IH\ min}$  表示逻辑“1”， $V_{IL\ max}$  表示逻辑“0”。

### 24.6.2 漂移波形 <sup>(1)</sup>

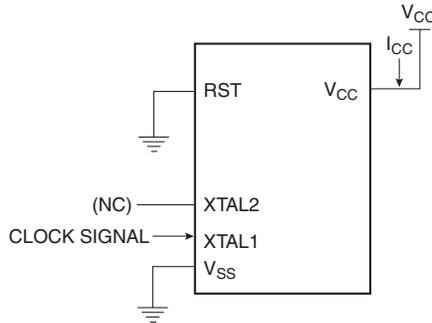


Note: 1. 为了时序，当负载电压发生了  $100\text{ mV}$  的变化时端口引脚不再漂移。当负载  $V_{OH}/V_{OL}$  电平发生了  $100\text{ mV}$  的变化时端口引脚开始发生漂移。

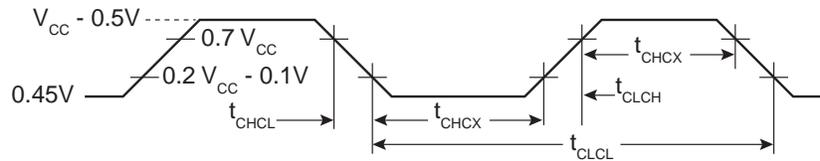
### 24.6.3 $I_{CC}$ 测试条件，活动模式，所有别的引脚都悬空



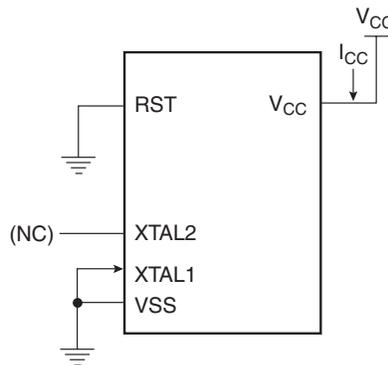
### 24.6.4 $I_{CC}$ 测试条件，空闲模式，所有别的引脚都悬空



### 24.6.5 活动和空闲模式下的 $I_{CC}$ 测试时钟信号波形 $t_{CLCH} = t_{CHCL} = 5 \text{ ns}$



### 24.6.6 $I_{CC}$ 测试条件，下电模式，所有别的引脚都悬空， $V_{CC} = 2\text{V to } 5.5\text{V}$





## 25. 分类信息

### 25.1 标准封装

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
20	2.4V to 5.5V	AT89LP213-20PI AT89LP213-20XI	14P3 14X	Industrial (-40°C to 85°C)
		AT89LP214-20PI AT89LP214-20XI	14P3 14X	

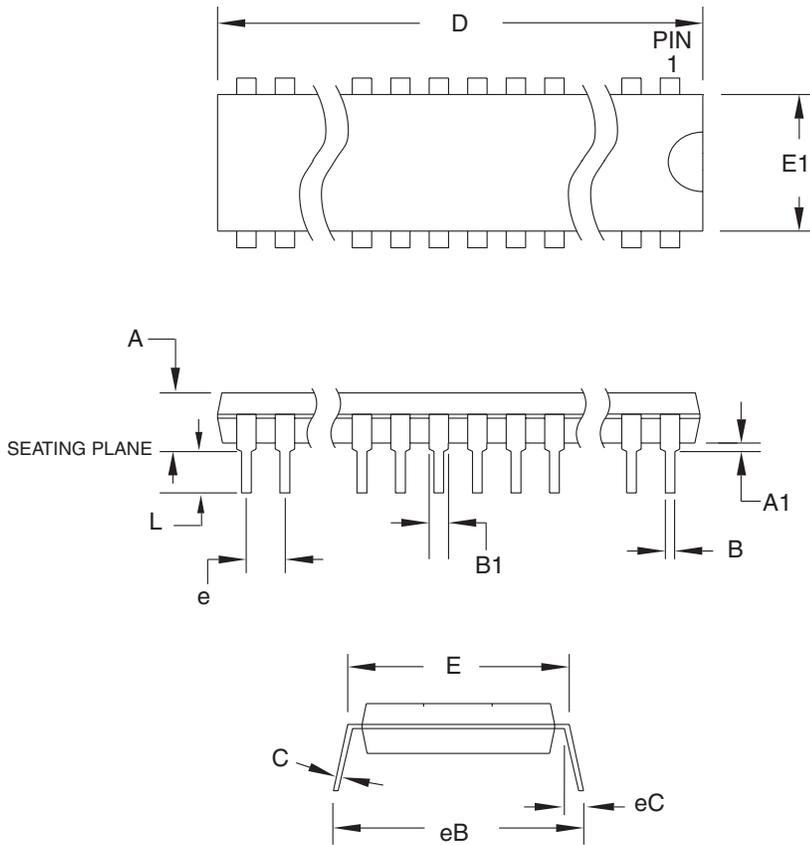
### 25.2 绿色封装 (Pb/Halide-free)

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
20	2.4V to 5.5V	AT89LP213-20PU AT89LP213-20XU	14P3 14X	Industrial (-40°C to 85°C)
		AT89LP214-20PU AT89LP214-20XU	14P3 14X	

Package Type	
<b>14P3</b>	14-lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)
<b>14X</b>	14-lead, 0.173" Wide, Plastic Thin Shrink Small Outline Package (TSSOP)

## 26. Packaging Information

### 26.1 14P3 – PDIP



**COMMON DIMENSIONS**  
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	5.334	
A1	0.381	–	–	
D	18.669	–	19.685	Note 2
E	7.620	–	8.255	
E1	6.096	–	7.112	Note 2
B	0.356	–	0.559	
B1	1.143	–	1.778	
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 AA.  
 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

**AMEL** 2325 Orchard Parkway  
San Jose, CA 95131

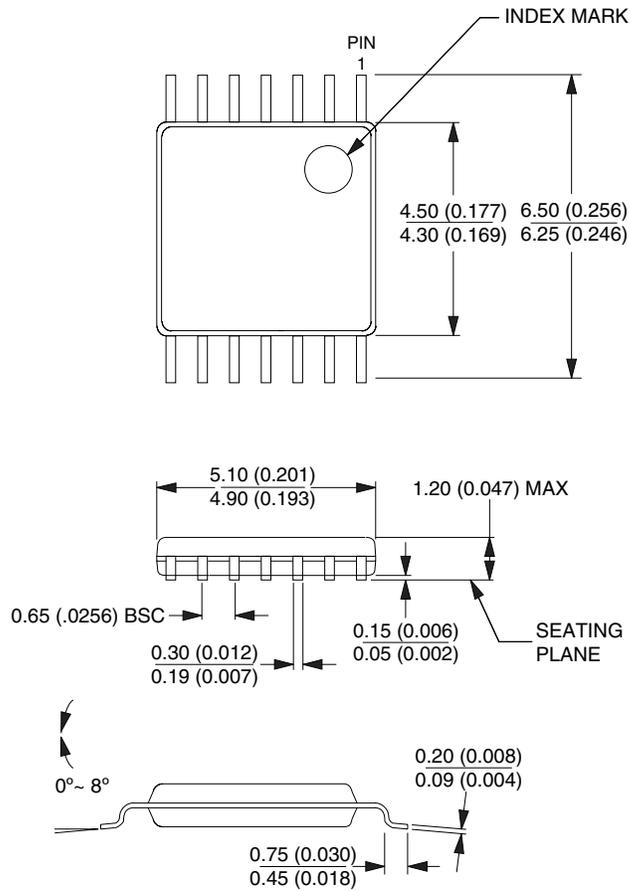
**TITLE**  
**14P3**, 14-lead (0.300"/7.62 mm Wide) Plastic Dual  
 Inline Package (PDIP)

**DRAWING NO.**  
14P3

**REV.**  
A

## 26.2 14X – TSSOP

Dimensions in Millimeters and (Inches).  
 Controlling dimension: Millimeters.  
 JEDEC Standard MO-153 AB-1.



05/16/01



2325 Orchard Parkway  
 San Jose, CA 95131

**TITLE**

**14X (Formerly "14T")**, 14-lead (4.4 mm Body) Thin Shrink  
 Small Outline Package (TSSOP)

**DRAWING NO.**

14X

**REV.**

B

## 27. Revision History

Revision No.	History
Revision A – June 2006	<ul style="list-style-type: none"><li data-bbox="766 342 966 367">• Initial Release</li></ul>

## Table of Contents

<b>1.</b>	<b>Description</b>	<b>1</b>
<b>2.</b>	<b>Pin Configuration</b>	<b>2</b>
2.1	AT89LP213: 14-lead TSSOP/PDIP	2
2.2	AT89LP214: 14-lead TSSOP/PDIP	2
<b>3.</b>	<b>Pin Description</b>	<b>3</b>
<b>4.</b>	<b>Block Diagram</b>	<b>5</b>
<b>5.</b>	<b>Comparison to Standard 8051</b>	<b>6</b>
5.1	System Clock	6
5.2	Instruction Execution with Single-Cycle Fetch	6
5.3	Interrupt Handling	6
5.4	Timer/Counters	6
5.5	Serial Port	6
5.6	Watchdog Timer	7
5.7	I/O Ports	7
5.8	Reset	7
<b>6.</b>	<b>Memory Organization</b>	<b>7</b>
6.1	Program Memory	7
6.2	Data Memory	8
<b>7.</b>	<b>Special Function Registers</b>	<b>9</b>
<b>8.</b>	<b>Enhanced CPU</b>	<b>10</b>
8.1	Restrictions on Certain Instructions	11
<b>9.</b>	<b>System Clock</b>	<b>12</b>
9.1	Crystal Oscillator	12
9.2	External Clock Source	12
9.3	Internal RC Oscillator	12
9.4	System Clock Out	12
<b>10.</b>	<b>Reset</b>	<b>13</b>
10.1	Power-on Reset	13
10.2	Brown-out Reset	15
10.3	External Reset	15
10.4	Watchdog Reset	16
10.5	Software Reset	16

<b>11. Power Saving Modes .....</b>	<b>16</b>
11.1 Idle Mode .....	16
11.2 Power-down Mode .....	16
<b>12. Interrupts .....</b>	<b>18</b>
12.1 Interrupt Response Time .....	20
<b>13. I/O Ports .....</b>	<b>22</b>
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
<b>14. Enhanced Timer/Counters .....</b>	<b>27</b>
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
<b>15. External Interrupts .....</b>	<b>36</b>
<b>16. General-purpose Interrupts .....</b>	<b>36</b>
<b>17. Serial Interface .....</b>	<b>38</b>
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
<b>18. Serial Peripheral Interface .....</b>	<b>49</b>
<b>19. Analog Comparator .....</b>	<b>55</b>
19.1 Comparator Interrupt with Debouncing .....	55
<b>20. Programmable Watchdog Timer .....</b>	<b>57</b>
20.1 Software Reset .....	58
<b>21. Instruction Set Summary .....</b>	<b>59</b>
<b>22. On-Chip Debug System .....</b>	<b>63</b>
22.1 Physical Interface .....	63
22.2 Software Breakpoints .....	64
22.3 Limitations of On-Chip Debug .....	64

<b>23. Programming the Flash Memory .....</b>	<b>65</b>
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
<b>24. Electrical Characteristics .....</b>	<b>76</b>
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 Test Conditions .....	80
24.6 Test Conditions .....	80
<b>25. Ordering Information .....</b>	<b>82</b>
25.1 Standard Package .....	82
25.2 Green Package Option (Pb/Halide-free) .....	82
<b>26. Packaging Information .....</b>	<b>83</b>
26.1 14P3 – PDIP .....	83
26.2 14X – TSSOP .....	84
<b>27. Revision History .....</b>	<b>85</b>