

**Nuvoton 8051内核 微控制器**

**N79E815A**  
**N79E814A**  
**N79E8132A**

中文规格书

版本: **A2.1 SC6**

## 目 录

1	概述.....	6
2	特性.....	7
3	器件信息列表 .....	9
4	结构框图 .....	10
5	管脚配置 .....	11
6	内存结构 .....	16
6.1	APROM 闪存存储器 .....	17
6.2	LDRAM 闪存存储器 .....	17
6.3	配置位.....	17
6.4	片内非易失性数据存储器.....	17
6.5	片内外部数据暂存器 XRAM .....	19
6.6	片内暂存器RAM 和 特殊功能寄存器SFR .....	19
6.7	工作寄存器 .....	20
6.8	可位寻址区 .....	21
6.9	堆栈 .....	21
7	特殊功能寄存器(SFR).....	22
8	通用80C51 系统控制 .....	27
9	通用IO (GPIO )结构及操作 .....	30
9.1	准双向输出配置 .....	30
9.1.1	读-修改-写 .....	31
9.2	开漏输出配置.....	32
9.3	推挽输出配置.....	32
9.4	输入配置 .....	33
10	定时器/计数器 .....	36
10.1	定时器/计数器0 和 1 .....	36
10.1.1	模式 0 (13位 定时器).....	39
10.1.2	模式1 (13位定时器).....	40
10.1.3	模式2 (8位自动重载定时器) .....	40
10.1.4	模式3 (两组独立8位定时器) .....	41
10.2	定时器/计数器 2.....	42
10.2.1	输入捕获模式 .....	45
10.2.2	自动重加载模式.....	50
10.2.3	比较模式.....	50
11	看门狗定时器(WDT).....	51
11.1	功能描述 .....	51
11.2	看门狗定时器复位的应用.....	53
11.3	看门狗定时器中断应用 .....	54
12	串口 (UART) .....	56
12.1	模式0 .....	58
12.2	模式1 .....	60
12.3	模式2 .....	62
12.4	模式 3 .....	64

12.5	波特率.....	66
12.6	帧错误检测.....	67
12.7	多机通讯.....	67
12.8	自动地址识别.....	68
13	串行外围设备接口(SPI).....	70
13.1	特征.....	70
13.2	功能描述.....	70
13.3	SPI控制寄存器.....	72
13.4	工作模式.....	75
13.4.1	主机模式.....	75
13.4.2	从机模式.....	75
13.5	时钟格式和数据传输.....	75
13.6	从机选择脚配置.....	78
13.7	模式故障侦测.....	78
13.8	写冲突错误.....	78
13.9	溢出错误.....	79
13.10	SPI 中断.....	79
14	键盘中断 (KBI).....	81
15	模数转换 (ADC).....	84
16	I <sup>2</sup> C总线.....	90
16.1	特征.....	90
16.2	功能描述.....	90
16.2.1	起始START 和停止STOP 信号.....	91
16.2.2	7位地址数据格式.....	92
16.2.3	应答.....	92
16.2.4	仲裁.....	93
16.3	I <sup>2</sup> C 控制寄存器.....	94
16.4	工作模式.....	97
16.4.1	主机发送模式.....	97
16.4.2	主机接收模式.....	99
16.4.3	从机接收模式.....	100
16.4.4	从机发送模式.....	101
16.4.5	广播呼叫.....	102
16.4.6	其它状态.....	103
16.5	典型I <sup>2</sup> C结构的中断服务程序.....	103
16.6	I <sup>2</sup> C 超时.....	107
16.7	I <sup>2</sup> C 中断.....	107
17	脉冲宽度调制(PWM).....	108
17.1	特征.....	108
17.2	功能描述.....	108
18	时控保护.....	116
19	中断系统.....	118
19.1	中断源.....	118
19.2	优先级结构.....	119

19.3	中断响应时间 .....	123
19.4	中断控制寄存器 .....	123
20	在线编程 (ISP) .....	127
20.1	ISP 引导程序 .....	127
20.2	ISP 命令行 .....	130
20.3	ISP 编程访问表 .....	131
20.4	ISP 用户指南 .....	131
20.5	ISP 例程 .....	132
21	电源管理 .....	134
21.1	空闲模式 .....	134
21.2	掉电模式 .....	135
22	时钟系统 .....	137
22.1	片内 RC 振荡器 .....	138
22.2	晶振 / 谐振荡器 .....	139
23	电源监控 .....	140
23.1	上电检测 .....	140
23.2	欠压检测 .....	140
24	复位条件 .....	142
24.1	上电复位 .....	142
24.2	欠压复位 .....	143
24.3	复位引脚复位 .....	144
24.4	看门狗定时器复位 .....	144
24.5	软件复位 .....	145
24.6	启动选项 .....	146
24.7	复位状态 .....	147
25	配置位 (CONFIG) .....	149
25.1	配置位 0 .....	149
25.2	配置位 1 (仅适用于 N79E815A) .....	150
25.3	配置位 2 .....	151
25.4	配置位 3 .....	152
26	指令集 .....	153
27	在电路编程 (ICP) .....	157
28	电气特性 .....	158
28.1	最大绝对额定值 .....	158
28.2	直流电气特性 .....	158
28.3	交流电气特性 .....	161
28.3.1	10 位 SAR-ADC 标准 .....	161
28.3.2	4~24MHz XTAL 外部晶振标准 .....	162
28.3.3	内部 22.1184MHz / 11.0592MHz RC 晶振标准 .....	162
28.3.4	内部 10kHz RC 晶振标准 .....	163
29	抗电测干扰典型应用电路 .....	164
30	封装尺寸 .....	165
30.1	28 脚 TSSOP – 4.4X9.7mm .....	165
30.2	20 脚 TSSOP – 4.4X6.5mm .....	166

30.3	28脚 SOP-300mil .....	167
30.4	20脚 SOP - 300 mil .....	168
30.5	16脚 SOP - 150 mil .....	169
31	版本历史 .....	170

## 1 概述

N79E815A/814A/8132A 为增强型8位51微控制器 (4T 模式)，内嵌16K<sup>[1]</sup>/8K/4K 的存储器 EPROM，可以通过通用烧写器，ICP编程器，自带ISP功能等烧录程序。N79E815A/814A/8132A 系列的指令集与标准8052兼容。N79E815A/814A/8132A 内存程序空间分别为 16K/8K/4K 字节的APROM，4K字节的数据内存及2K 字节的LDROM； 256字节的直接和间接RAM, 256 字节MOVX RAM； 25个可位寻址的 I/O口； 两个 16-位的定时器/计数器； 8-通道多路10-位 A/D 转换器； 4-通道10-位 PWM； 三个串行接口分别为一个SPI, 一个 I2C和一个增强型串口； 2-级 BOD电压检测/复位, 低压检测/复位(LVR) 和上电复位 (POR)。 N79E815A/814A/8132A系列自带内部 22.1184MHz RC 振荡器，可产生 22.1184MHz / 11.0592MHz 内部晶振，最高精度偏差仅±1%。N79E815A/814A/8132A有14个中断源，4个中断级。内部的闪存存储器可电擦写。一旦代码确定，用户可以通过安全码保护自己的代码。

N79E815A/814A/8132A 微控制器系列, 宽电压操作, 内建丰富的模拟和数字外设与非易失性闪存存储器, 广泛应用于通用控制领域, DC/BLDC 电机驱动系统和家电应用。

[1] N79E815A，数据和 APROM 共享16k字节空间。

## 2 特性

- 内核
  - 全静态8位CMOS加速51(4T) CMOS微控制器
  - 指令与MCS-51兼容
- 工作电压范围
  - $V_{DD} = 4.5V \text{ to } 5.5V @ F_{OSC}$  最高至 24MHz
  - $V_{DD} = 3.0V \text{ to } 5.5V @$  内部 RC 22.1184MHz
  - $V_{DD} = 2.4V \text{ to } 5.5V @ F_{OSC} = 12MHz$  或内部 RC 11.0592MHz
- 工作温度范围
  - $-40^{\circ}C \sim 85^{\circ}C$
- 时钟源
  - 高速外部振荡器:
    - 最高24 MHz 晶振和振荡器 (由配置位CONFIG使能)
  - 内部RC 振荡器: 22.1184MHz/11.0592MHz (由配置位CONFIG选择)
    - 在  $V_{DD} = 5V$  温度  $25^{\circ}C$  条件下精度为  $\pm 1\%$
    - 在  $V_{DD} = 2.7V \sim 5.5V$  温度  $25^{\circ}C$  条件下精度为  $\pm 3\%$
    - 在  $V_{DD} = 2.7V \sim 5.5V$  温度  $-10^{\circ}C \sim +70^{\circ}C$  条件下精度为  $\pm 5\%$
    - 在  $V_{DD} = 2.7V \sim 5.5V$  温度  $-40^{\circ}C \sim 85^{\circ}C$  条件下精度为  $\pm 8\%$
  - CPU时钟源由配置位CONFIG和软件配置
  - 8-位可编程的CPU时钟除频器 (DIVM)
- 芯片内存
  - FLASH 擦写/编程次数 100,000次
  - N79E815A : 总共 16K-字节分别用于APROM程序闪存存储器 加 数据闪存存储器 (由配置位定义各自容量)
  - N79E814A: 8K字节APROM 程序闪存存储器 和 4K-字节数据闪存存储器
  - N79E8132A: 4K字节 APROM 程序闪存存储器 和 4K-字节数据闪存存储器
  - APROM闪存存储器/ LDRAM闪存存储器 和 数据闪存存储器 都可加密保护
  - 全部闪存存储器区域128字节为一页
  - 256 字节片上直接/间接RAM
  - 256 字节MOVX-RAM, 用指令MOVX 访问.
  - 2K-字节LD 闪存存储器 用于ISP功能(LDRAM)
  - 片上闪存存储器可用以下方式编程
    - 通用 H/W 烧录器模式
    - 串行方式在电路编程模式 (ICP)
    - 软件执行ISP (在线编程)
- I/O端口
  - 根据不同封装, 最多 25个 I/O端口
  - 除P1.2和P1.3外, 所有引脚支持由软件配置的4种不同模式

- 软件可选择TTL或史密特触发输入类型
- 14 个中断源带4级中断优先级.
- P10, P11, P14, P16, P17的LED 驱动能力为38mA.
- 端口P0, P2, P3的LED驱动能力为20mA.
- 定时器/计数器
  - 2 组16-位定时器/计数器.
  - 一个16位定时器带三路输入捕获功能
- 看门狗定时器
  - 可编程看门狗定时器
  - 时钟源为内部额外10kHz 偏差率±50% RC振荡器.
- 串口(UART, SPI, I<sup>2</sup>C )
  - 一组增强型全双工UART口, 带帧错误检测和自动地址识别。
  - 一路SPI 带主/从功能. 软件可切换两组SPI引脚
  - 一个I<sup>2</sup>C 带主/从功能.
- PWM
  - 4通道 10-位 PWM 输出带一路 钳位/故障 (Brake/Fault) 输入
- KBI
  - 8-键盘中断输入(KBI) 带8个上升沿/下降沿/双边沿 检测引脚 (由软件选择)
- ADC
  - 10-位 A/D 转换器
  - 最高达150 Ksps.(每秒采样率)
  - 8个模拟输入通道
- 欠压检测
  - 2-级 (3.8V/2.7V) BOD 检测器
  - 支持低电压中断和低电压复位选项
- POR (上电复位)
  - 门限电压为2.0V
- 内建电源管理.
  - 睡眠模式
  - 掉电模式 可选使能WDT功能
- 开发工具
  - ICE(在电路仿真)工具
  - 通用型编程器
  - ICP 编程器
  - ISP 编程器 (同时支持标准串口模式 / 脱机编程模式)



### 3 器件信息列表

表3-1: 无铅封装 (RoHS) 标准选型表

型号	APROM	LDROM	RAM	数据存储器	封装
N79E815AS28	16KB	2KB	512B	与 APROM 共享	SOP-28 脚
N79E815AT28	16KB	2KB	512B	与 APROM 共享	TSSOP-28 脚
N79E814AS28	8KB	2KB	512B	4KB	SOP-28 脚
N79E814AT28	8KB	2KB	512B	4KB	TSSOP-28 脚
N79E815AS20	16KB	2KB	512B	与 APROM 共享	SOP-20 脚
N79E815AT20	16KB	2KB	512B	与 APROM 共享	TSSOP-20 脚
N79E814AS20	8KB	2KB	512B	4KB	SOP-20 脚
N79E814AT20	8KB	2KB	512B	4KB	TSSOP-20 脚
N79E8132AS16	4KB	2KB	512B	4KB	SOP-16 脚

### 4 结构框图

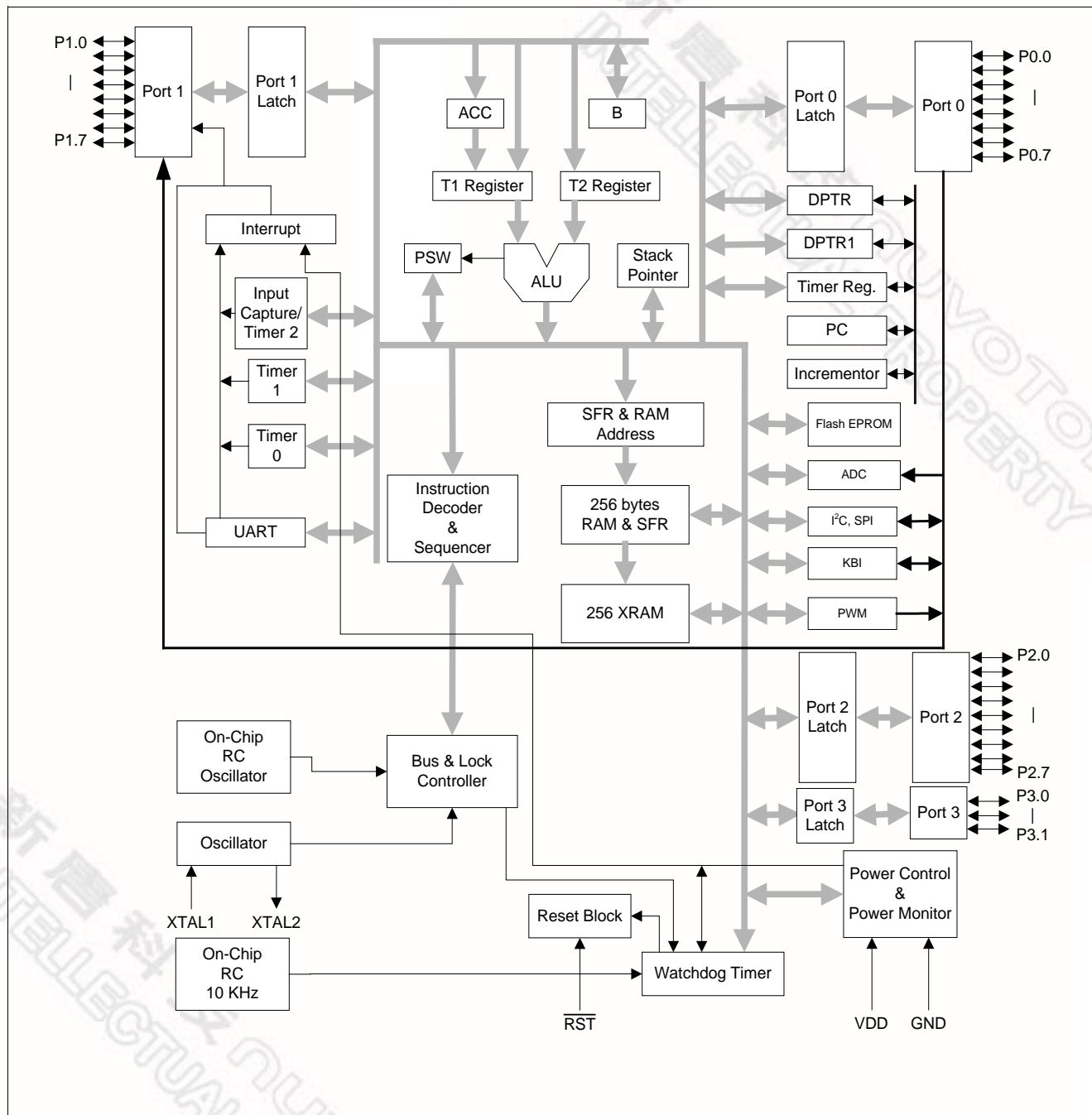


图4-1. N79E815A/814A/8132A功能模块框图

5 管脚配置

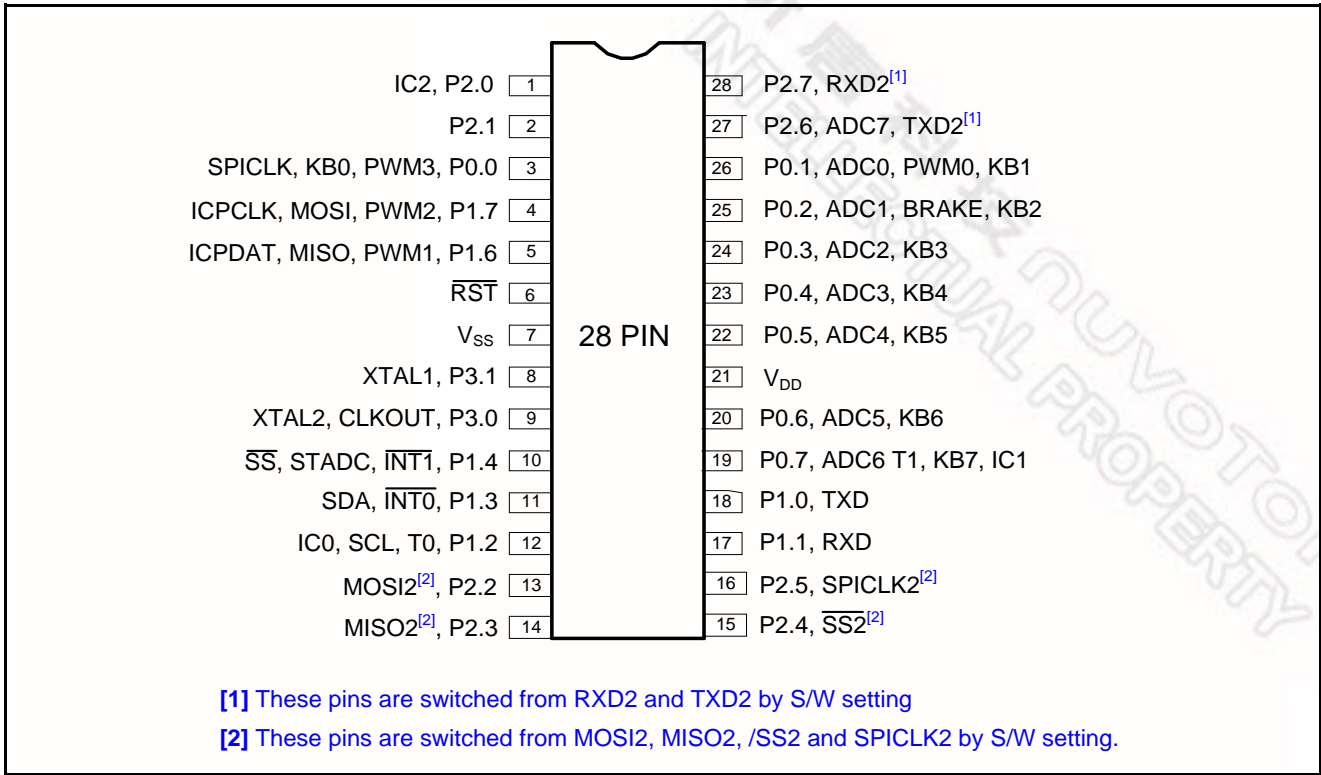


图5-1. TSSOP28 管脚分配图

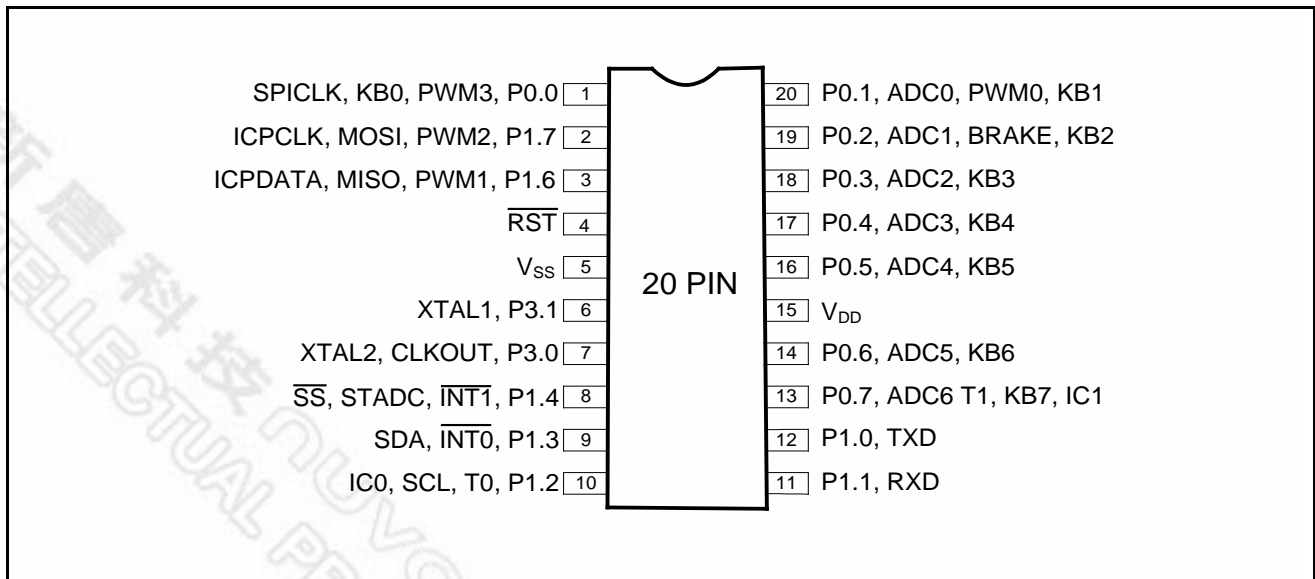


图5-2. TSSOP20 管脚分配

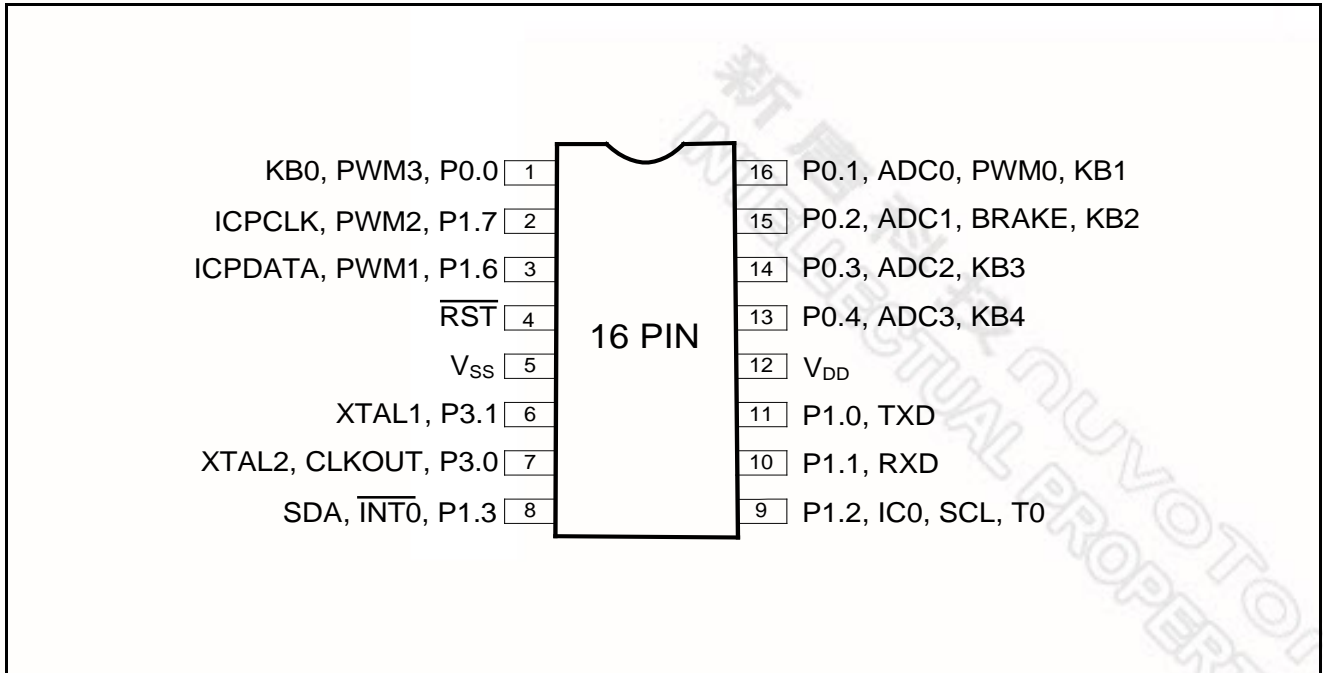


图5-3. SOP 16管脚分配

表5-1. 管脚描述

管脚号			符号	复用功能				管脚类型	描述
SOP16	SOP20 TSSOP20	SOP28 TSSOP28		1	2	3			
12	15	21	V <sub>DD</sub>					P	提供电源: 提供工作电源V <sub>DD</sub> .
5	5	7	V <sub>SS</sub>					P	GROUND:地.
4	4	6	/RST					I (ST)	RESET: 芯片复位引脚, 低电平有效. 因为芯片复位脚内部上拉电阻约 200K 欧姆, 所以该脚不可悬空, 要 求在复位脚必须接上拉 100 欧姆电 阻并 10uF 电容到地
1	1	3	P0.0	PWM3		KB0	SPICLK	I/O	端口 0 有 4 个类型的 I/O 口. 其复用 能可用于 PWM0, PWM3, T1, BRAKE, SPICLK2, ADC0-ADC6 和 KB0-KB7. ADC0 ~ADC6: ADC 通道输入. KB0 ~ KB7: 键盘输入 PWM0 和 PWM3 是 PWM 输出通道 T1: 定时器 1 外部输入 SPICLK: SPI-1 时钟输入 BRAKE: PWM 钳位输入
16	20	26	P0.1	PWM0	ADC0	KB1		I/O	
15	19	25	P0.2	BRAKE	ADC1	KB2		I/O	
14	18	24	P0.3		ADC2	KB3		I/O	
13	17	23	P0.4		ADC3	KB4		I/O	
-	16	22	P0.5		ADC4	KB5		I/O	
-	14	20	P0.6		ADC5	KB6		I/O	
-	13	19	P0.7	T1	ADC6	KB7	IC1	I/O	
11	12	18	P1.0	TXD				I/O	端口 1 有 4 个类型的 I/O 口. 其多功 能用于 TXD, RXD, T0, /INT0, /INT1, SCL, SDA, STADC, ICPDAT, ICPCLK 和 /SS2, MISO, MOSI. P1.2 (SCL) 和 P1.3 (SDA) 为开漏
10	11	17	P1.1	RXD				I/O	



表5-1. 管脚描述

管脚号			符号	复用功能				管脚类型	描述
SOP16	SOP20 TSSOP20	SOP28 TSSOP28		1	2	3			
9	10	12	P1.2	T0		SCL	IC0	D	口。 ICPDAT 和 ICPCLK 为 ICP (在电路编程)功能引脚。 /SS2, MISO, MOSI 为 SPI-1 功能引脚。 PWM1 和 PWM2 为 PWM 输出通道 T0: 定时器 0 外部输入 IC0/1: 捕获输入引脚
8	9	11	P1.3	/INT0		SDA		D	
-	8	10	P1.4	/INT1	STADC		/SS	I/O	
3	3	5	P1.6	PWM1		ICPDAT	MISO	I/O	
2	2	4	P1.7	PWM2		ICPCLK	MOSI	I/O	
-	-	1	P2.0				IC2	I/O	端口 2 有 4 个类型的 I/O 口. 其多功能用于 T2, ADC7, TXD2, RXD2 和 MOSI, MISO, /SS, SPICLK TXD2 和 RXD2 为 UART 口, MOSI2, MISO2, /SS 和 SPICLK2 为 SPI-2 功能引脚。 <b>ADC7</b> : ADC通道输入。 <b>IC2</b> : 捕获引脚输入
-	-	2	P2.1					I/O	
-	-	13	P2.2				MOSI2	I/O	
-	-	14	P2.3				MISO2	I/O	
-	-	15	P2.4				/SS2	I/O	
-	-	16	P2.5				SPICLK2	I/O	
-	-	27	P2.6	TXD2	ADC7			I/O	
-	-	28	P2.7	RXD2				I/O	
7	7	9	P3.0	XTAL2	CLKOUT			I/O	端口 3 有 4 个类型的 I/O 口. 其多功

表5-1. 管脚描述

管脚号			符号	复用功能			管脚类型	描述
SOP16	SOP20 TSSOP20	SOP28 TSSOP28		1	2	3		
6	6	8	P3.1	XTAL1				能用于 XTAL1, XTAL2 和 CLKOUT, <b>CLKOUT</b> : 内部RC OSC/4 输出引脚. <b>XTAL2</b> : 内部放大器的输出. 发送 XTAL2的反向信号XTAL2 <b>XTAL1</b> : 内部放大器的输出. 发送 XTAL1的反向.

[1] I/O 类型描述 I:输入, O:输出, I/O: 准双向, D: 开漏, P: 电源脚, ST: 史密特触发

## 6 内存结构

N79E815A/814A/8132A系列分别内嵌有16K/8K字节的闪存存储器作为应用程序空间(APROM)，除N79E815外，都配置有4K字节的数据闪存存储器（N79E815A由用户自定义通过配置位设置），内嵌2K字节的ISP引导程序存储器(LDROM)和配置位。N79E815A/814A/8132A全系列提供256字节的片上直接/间接RAM以及额外256字节由MOVX指令访问的XRAM。

对于16K字节APROM的N79E815，APROM和数据存储器共享16K空间。具体数据存储器大小由配置位CONFIG通过编程器进行设定。

N79E815A/814A/8132A的FLASH支持内建CMOS页擦除机制，每个页为128个字节。通过该机制，用户可灵活运用数据存储器。

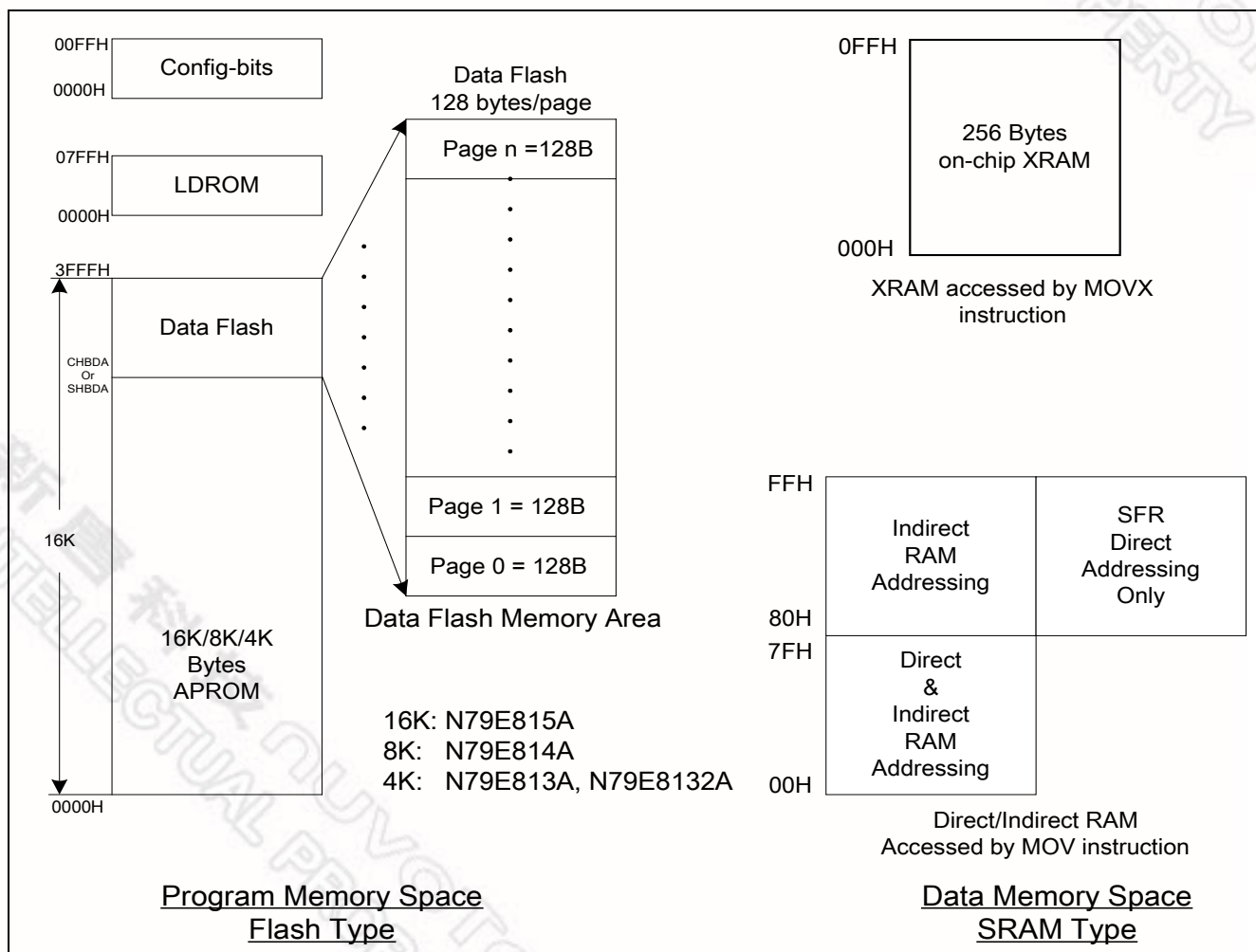


图6-1 N79E815A/814A/8132A 内存结构图



## 6.1 APROM 闪存存储器

N79E815A/814A/8132A 系列的应用程序内存分别为 16K/8K 字节。所有指令由该内存区域取出并执行。指令 MOV<sub>C</sub> 可以访问该内存区域。

用户的应用程序位于 APROM。当 CPU 由 APROM 启动 (CHPCON.BS=0)，CPU 开始从 APROM 地址 0000H 开始执行程序。如果程序计数器 (PC) 的值超过 AP 闪存存储器的空间，CPU 将执行 NOP 指令，程序计数器持续加 1 直到 PC 达到 3FFFH，然后回到 AP 闪存存储器地址 0000H，CPU 重新执行应用程序。

## 6.2 LDROM 闪存存储器

N79E815A/814A/8132A 系列配备有 2K 字节 LDROM 用于存储 ISP 引导程序。用户存储在 LDROM 内的 ISP 更新程序，用于更新应用程序或数据存储器内容。同时，APROM 也可以重新编写 LDROM 和数据存储器中的内容。LDROM 的起始地址为 0000H。当 CPU 由 LDROM 启动时，CPU 自动定位到 LDROM 的 0000H 地址，因此在 LDROM 中程序视为独立程序存储区域，意味着，在这个程序存储器中，CPU 所有的中断向量都支持。

## 6.3 配置位

在配置位区域有几个字节的配置位。上电复位后，配置位定义了 CPU 的初始设置。可通过通用型编程器 / ICP 编程器进行硬件擦除/编程。也可通过软件的 ISP 方式改写配置位。

## 6.4 片内非易失性数据存储器

N79E815A/814A/8132A 系列配备有非易失性数据存储器，数据存储器掉电时仍可保存数据。因此，用户可以在程序内执行写入参数或常量便于读出应用。数据存储器可以通过 ISP 编程程序方式擦除/编写，也可以通过硬件方式在通用编程器 / ICP 编程器更改内容。

**N79E814A** 的数据存储器空间固定为 4K 地址为 3000H 到 3FFFH。

**N79E815A** (16KB) 数据存储器大小必须通过配置位 SHBDA 内的值来设置。SHBDA[7:0] 表示 16 位数据存储器开始地址的高字节，低字节由硬件默认为 00H。例如，设定 SHBDA = 38H，则数据存储器大小为 2K 字节，其余 APROM 程序空间。复位后，SHBDA 的值根据 CONFIG1 (CHBDA) 内的值载入。应用程序可以通过重置 SHBDA 的值动态校正三局存储器容量大小。一旦数据存储器的容量大小改变，APROM 的容量大小也相应改变。为避免误操作，对 SHBDA 进行写操作时有时序保护要求。注意如果 CHBDA 值为 00H，数据存储器的大小为 16k 字节，则没有片内 APROM。CPU 将自动执行外部内存的代码。

使用数据存储器之前必须使能该功能，将配置位内的 DFEN (CONFIG0.0) 设定为0使能。未经编程默认DFEN的值为1。

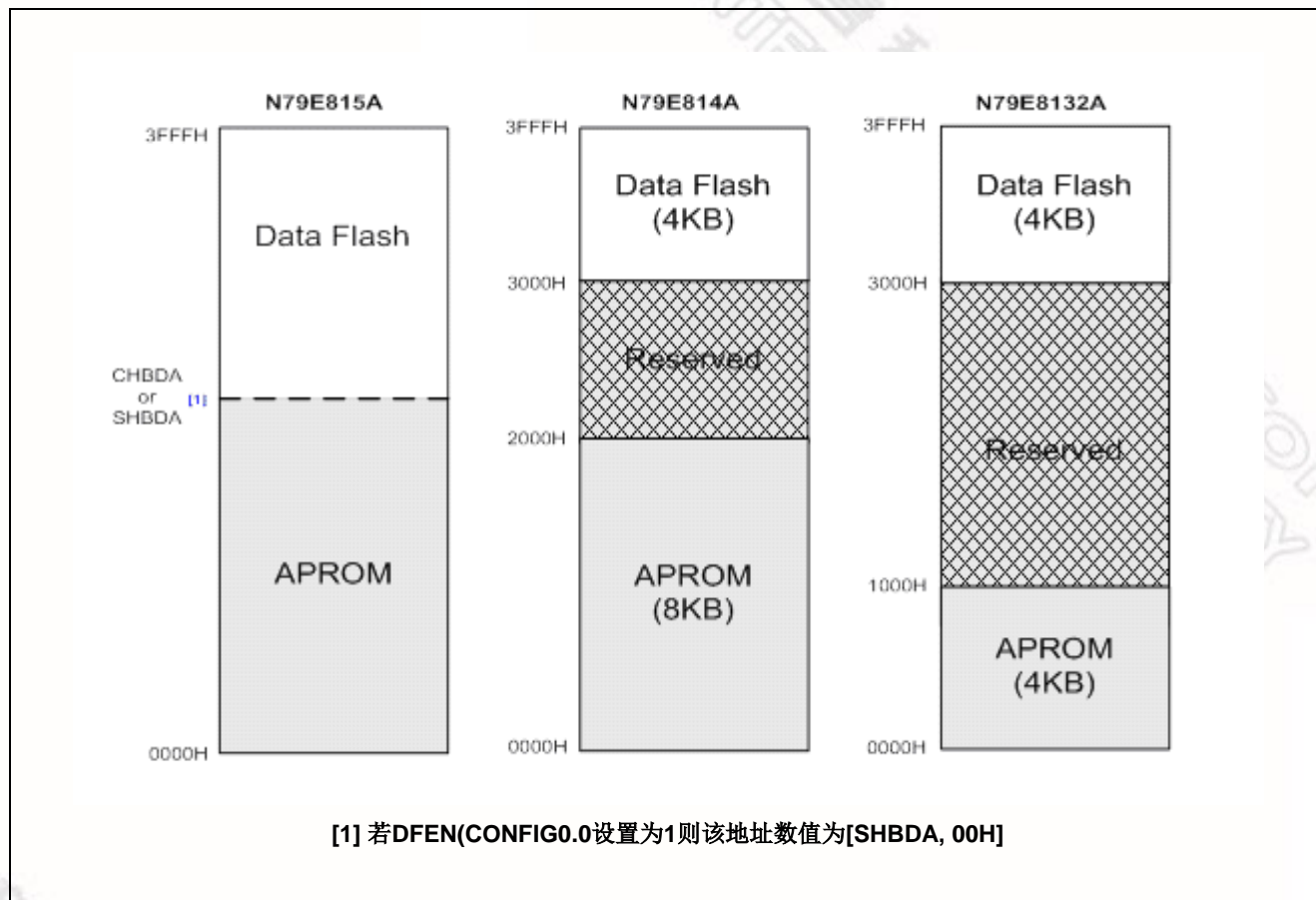


图6-2. N79E815A/814A/8132A 数据存储器

SHBDA – SFR起始地址高字节 (TA保护, 仅N79E815A使用)

7	6	5	4	3	2	1	0
SHBDA[7:0] <sup>[1]</sup>							
r/w							

地址: 9CH

位	名称	描述
7:0	SHBDA[7:0]	数据存储器起始地址高字节 字节仅在DFEN (CONFIG0.0)为0的条件下有效. 用于应用程序执行时动态校正数据存储器的开始地址.

[1] 复位后SHBDA 从CONFIG1 内载入值

## 6.5 片内外部数据暂存器 XRAM

N79E815系列提供片上256字节辅助RAM区域（XRAM）以扩展RAM空间。地址空间从00H到FFH。XRAM的256字节可以由MOVX @DPTR或MOVX @Ri间接访问。（参见下面的例程）注意堆栈指针不能在XRAM上。图6-1所示该系列产品的内存图

XRAM 例程:

```

MOV R0,#23H ; write #5AH to XRAM with address @23H
MOV A,#5AH
MOVX @R0,A

MOV R1,#23H ; read from XRAM with address @23H
MOVX A,@R1

MOV DPTR,#0023H ; write #5BH to XRAM with address @0023H
MOV A,#5BH
MOVX @DPTR,A

MOV DPTR,#0023H ; read from XRAM with address @0023H
MOVX A,@DPTR

```

## 6.6 片内暂存器RAM和特殊功能寄存器SFR

N79E815A/814A/8132A系列提供软件读写片内256 RAM和特殊功能寄存器（SFRs）。SFR仅可直接寻址，片上RAM可直接/间接寻址。

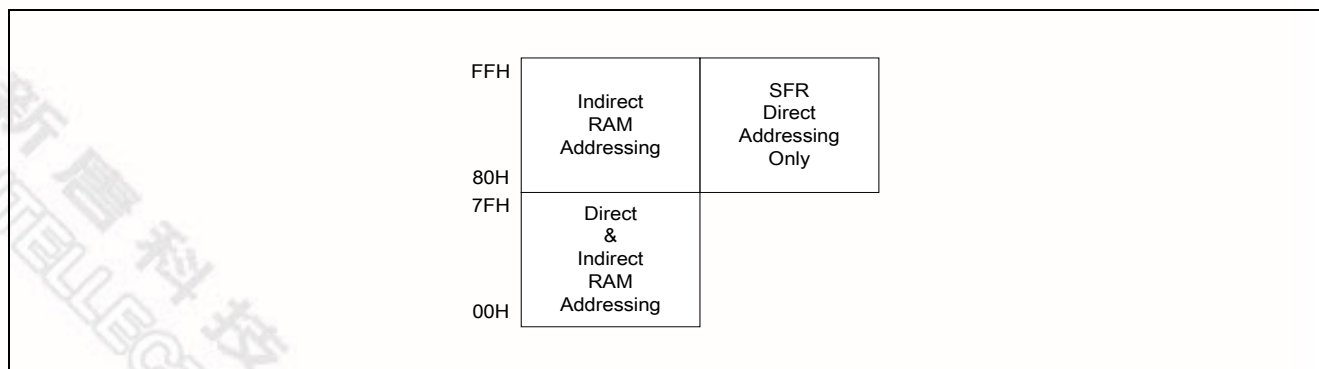


图6-3 256 字节 RAM及SFR示意图

由于暂存器只有256字节，仅在数据内容很小时才用。在暂存器里有几个其他特殊用途区域。描述如下：

FFH	Indirect Accessing RAM							
80H 7FH	Direct or Indirect Accessing RAM							
30H								
2FH	7F	7E	7D	7C	7B	7A	79	78
2EH	77	76	75	74	73	72	71	70
2DH	6F	6E	6D	6C	6B	6A	69	68
2CH	67	66	65	64	63	62	61	60
2BH	5F	5E	5D	5C	5B	5A	59	58
2AH	57	56	55	54	53	52	51	50
29H	4F	4E	4D	4C	4B	4A	49	48
28H	47	46	45	44	43	42	41	40
27H	3F	3E	3D	3C	3B	3A	39	38
26H	37	36	35	34	33	32	31	30
25H	2F	2E	2D	2C	2B	2A	29	28
24H	27	26	25	24	23	22	21	20
23H	1F	1E	1D	1C	1B	1A	19	18
22H	17	16	15	14	13	12	11	10
21H	0F	0E	0D	0C	0B	0A	09	08
20H	07	06	05	04	03	02	01	00
1FH	Register Bank 3							
18H 17H	Register Bank 2							
10H 0FH	Register Bank 1							
08H 07H	Register Bank 0							
00H								

图6-4 数据存储器和可位寻址区域

## 6.7 工作寄存器

共有四组工作寄存器，每组由8位寄存器组成，分布在bank区域0, 1, 2, 和 3。这些区域的寄存器可以由特殊指令直接寻址。这些寄存器为R0, R1, R2, R3, R4, R5, R6 和 R7。然而，N79E815A/814A/8132A系列一次只能工作在一个寄存器区域。区域配置通过设置PSW的RS1-RS0位来选择。寄存器 R0 和 R1用于间接寻址的地址存储。

## 6.8 可位寻址区

位于20h 到 2Fh 的暂存器区可通过位寻址，这也意味着该区域上的每位可以独立寻址。除此之外还有一些特殊功能寄存器也可以位寻址，如在SFR 区域, SFR的地址以0或8结尾都是可位寻址的。

## 6.9 堆栈

暂存器可被作为堆栈使用。该区域由堆指针（SP）选定，存储于堆栈顶部顶部。当发生跳转，呼叫自程序或发生中断后，自动返回堆栈地址。堆栈起始地址放置在RAM的位置并没有规定，但在默认情况下，复位时堆栈指针在07h，用户可以改变该值。通常SP指向之前使用的值。SP遵守后入先出原则。当地址存入堆栈后，SP指针值增加，相反，由堆栈内取出地址后，SP指针递减。

## 7 特殊功能寄存器(SFR)

N79E815A/814A/8132A系列特殊功能寄存器(SFR)用于控制和监测外设运行和外设模式。特殊功能寄存器位于80H-FFH的地址空间内，只能用直接寻址的方式来访问。一些特殊功能寄存器是可位寻址的，这个功能特别适用于只想修改寄存器中的某一位而不影响其它位的场合。可位寻址的特殊功能寄存器，其地址编号是以0H或8H结尾。N79E815A/814A/8132A系列中含有标准8051中所有的特殊功能寄存器，同时也加入了一些新的特殊功能寄存器。在一些应用场合，8051中未被定义的位被赋予了新的功能。下表列出了特殊功能寄存器。

表7-1.N79E815A/814A/8132A特殊功能寄存器(SFR)分配图

F8	<b>ADCCON0</b>	-	-	-	-	-	-	EIP	FF
F0	<b>B</b>	-	-	SPCR	SPSR	SPDR	P0DIDS	EIPH	F7
E8	<b>EIE</b>	KBIE	KBIF	KBLS0	KBLS1	C2L	C2H	-	EF
E0	<b>ACC</b>	ADCCON1	ADCH	-	C0L	C0H	C1L	C1H	E7
D8	<b>WDCON0</b>	PWMPL	PWM0L	PWM1L	PWMCON0	PWM2L	PWM3L	PWMCON1	DF
D0	<b>PSW</b>	PWMPH	PWM0H	PWM1H	-	PWM2H	PWM3H	PWMCON2	D7
C8	<b>T2CON</b>	T2MOD	RCOMP2L	RCOM2H	TL2	TH2	-	-	CF
C0	<b>I2CON</b>	I2ADDR	-	-	-	-	-	TA	C7
B8	<b>IP</b>	SADEN	-	-	I2DAT	I2STA	I2CLK	I2TOC	BF
B0	<b>P3</b>	P0M1	P0M2	P1M1	P1M2	P2M1	P2M2	IPH	B7
A8	<b>IE</b>	SADDR	-	<b>WDCON1</b>	-	-	ISPFD	ISPCN	AF
A0	<b>P2</b>	-	AUXR1	<b>PMCR</b>	<b>ISPTRG</b>	-	ISPAL	ISPAH	A7
98	<b>SCON</b>	SBUF	-	-	<b>SHBDA</b>	-	-	<b>CHPCON</b>	9F
90	<b>P1</b>	-	CAPCON0	CAPCON1	CAPCON2	DIVM	P3M1	P3M2	97
88	<b>TCON</b>	TMOD	TL0	TL1	TH0	TH1	CKCON	-	8F
80	<b>P0</b>	SP	DPL	DPH	-	-	-	PCON	87

<b>粗体字</b>
-

可位寻址

保留位

注:

- 保留的 SFR 地址必须保持他们自己的初始值. 用户不得自行改变其值.
- 有深色边框的 SFRs 可位寻址

\*带 TA-保护. (定时读写保护)

表7-2.N79E815A/814A/8132ASFR功能描述和复位值

符号	定义	地址	MSB								LSB		复位值 <sup>[1]</sup>
EIP	Interrupt Priority 1	FFH	PT2	PSPI	PPWM	PWDI	-	-	PKB	PI2		0000 0000B	
ADCCON0	ADC control register 0	F8H	(FF) ADC.1	(FE) ADC.0	(FD) ADCEX	(FC) ADCI	(FB) ADCS	(FA) AADR2	(F9) AADR1	(F8) AADR0		0000 0000B	
EIPH	Interrupt High Priority 1	F7H	PT2H	PSPIH	PPWMH	PWDIH	-	-	PKBH	PI2H		00000000B	
PODIDS	Port 0 Digital Input Disable	F6H	PODIDS[7:0]									0000 0000B	
SPDR	Serial Peripheral Data Register	F5H	SPDR[7:0]									00000000B	
SPSR	Serial Peripheral Status Register	F4H	SPIF	WCOL	SPIOVF	MODF	DISMODF	-	-	-		0000 0000B	
SPCR	Serial Peripheral Control Register	F3H	SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR1	SPR0		0000 0100B	
B	B register	F0H	(F7) B.7	(F6) B.6	(F5) B.5	(F4) B.4	(F3) B.3	(F2) B.2	(F1) B.1	(F0) B.0		0000 0000B	
C2H	Input Capture 2 High	EEH	C2H[7:0]									0000 0000B	
C2L	Input Capture 2 Low	EDH	C2L[7:0]									0000 0000B	
KBLS1	Keyboard level select 1	ECH	KBLS1[7:0]									0000 0000B	
KBLS0	Keyboard level select 0	EBH	KBLS0[7:0]									0000 0000B	
KBIF	KBI Interrupt Flag	EAH	KBIF[7:0]									0000 0000B	
KBIE	Keyboard Interrupt Enable	E9H	KBIE[7:0]									0000 0000B	
EIE	Interrupt enable 1	E8H	(EF) ET2	(EE) ESPI	(ED) EPWM	(EC) EWDI	(E7)	(E8) ECPTF	(E9) EKB	(E8) EI2C		0000 0000B	
C1H	Input Capture 1 High	E7H	C1H[7:0]									0000 0000B	
C1L	Input Capture 1 Low	E6H	C1L[7:0]									0000 0000B	
C0H	Input Capture 0 High	E5H	C0H[7:0]									0000 0000B	
C0L	Input Capture 0 Low	E4H	C0L[7:0]									0000 0000B	
ADCH	ADC converter result	E2H	ADC.9	ADC.8	ADC.7	ADC.6	ADC.5	ADC.4	ADC.3	ADC.2		0000 0000B	
ADCCON1	ADC control register1	E1H	ADCEN	-	-	-	-	-	RCCLK	ADC0SEL		0000 0000B	
ACC	Accumulator	E0H	(E7) ACC.7	(E6) ACC.6	(E5) ACC.5	(E4) ACC.4	(E3) ACC.3	(E2) ACC.2	(E1) ACC.1	(E0) ACC.0		0000 0000B	
PWMCON1	PWM control register 1	DFH	BKCH	BKPS	BPEN	BKEN	PWM3B	PWM2B	PWM1B	PWM0B		0000 0000B	
PWM3L	PWM 3 low bits register	DEH	PWM3.7	PWM3.6	PWM3.5	PWM3.4	PWM3.3	PWM3.2	PWM3.1	PWM3.0		0000 0000B	
PWM2L	PWM 2 low bits register	DDH	PWM2.7	PWM2.6	PWM2.5	PWM2.4	PWM2.3	PWM2.2	PWM2.1	PWM2.0		0000 0000B	
PWMCON0	PWM control register 0	DCH	PWMRUN	LOAD	CF	CLRPWM	PWM3I	PWM2I	PWM1I	PWM0I		0000 0000B	
PWM1L	PWM 1 low bits register	DBH	PWM1.7	PWM1.6	PWM1.5	PWM1.4	PWM1.3	PWM1.2	PWM1.1	PWM1.0		0000 0000B	
PWM0L	PWM 0 low bits register	DAH	PWM0.7	PWM0.6	PWM0.5	PWM0.4	PWM0.3	PWM0.2	PWM0.1	PWM0.0		0000 0000B	
PWMP0L	PWM counter low register	D9H	PWMP0.7	PWMP0.6	PWMP0.5	PWMP0.4	PWMP0.3	PWMP0.2	PWMP0.1	PWMP0.0		0000 0000B	
WDCON0 <sup>[4]</sup> <sup>[3]</sup>	Watch-Dog control 0	D8H	(DF) WDTEN	(DE) WDCLR	(DD) WDTF	(DC) WIDPD	(DB) WDTRF	(DA) WPS2	(D9) WPS1	(D8) WPS0	Power-ON C000 0000B Watch reset C0UU 1UUUB Other reset C0UU UUUUB		
PWMCON2	PWM control register 2	D7H	-	-	-	-	FP1	FP0	-	BKF		0000 0000B	
PWM3H	PWM 3 high bits register	D6H	-	-	-	-	-	-	PWM3.9	PWM3.8		0000 0000B	
PWM2H	PWM 2 high bits register	D5H	-	-	-	-	-	-	PWM2.9	PWM2.8		0000 0000B	
PWM1H	PWM 1 high bits register	D3H	-	-	-	-	-	-	PWM1.9	PWM1.8		0000 0000B	
PWM0H	PWM 0 high bits register	D2H	-	-	-	-	-	-	PWM0.9	PWM0.8		00000000B	
PWMP0H	PWM counter high register	D1H	-	-	-	-	-	-	PWMP0.9	PWMP0.8		0000 0000B	



表7-2.N79E815A/814A/8132ASFR功能描述和复位值

符号	定义	地址	MSB								LSB			复位值 <sup>[1]</sup>					
			(D7) CY	(D6) AC	(D5) F0	(D4) RS1	(D3) RS0	(D2) OV	(D1) F1	(D0) P	(D7) TH2	(D6) TL2	(D5) RCOMP2H		(D4) RCOMPL2	(D3) LDEN	(D2) T2DIV[2:0]	(D1) CAPCR	(D0) COMPCR
PSW	Program status word	D0H	(D7) CY	(D6) AC	(D5) F0	(D4) RS1	(D3) RS0	(D2) OV	(D1) F1	(D0) P									0000 0000B
TH2	Timer 2 MSB	CDH	TH2[7:0]											0000 0000B					
TL2	Timer 2 LSB	CCH	TL2[7:0]											0000 0000B					
RCOMP2H	Timer 2 Reload MSB	CBH	RCOMP2H[7:0]											0000 0000B					
RCOMP2L	Timer 2 Reload LSB	CAH	RCOMPL2[7:0]											0000 0000B					
T2MOD	Timer 2 Mode	C9H	LDEN	T2DIV[2:0]			CAPCR	COMPCR	LDTS[1:0]			0000 0000B							
T2CON	Timer 2 Control	C8H	(CF) TF2	-	-	-	-	(CA) TR2	-	(C8) CP/RL2				00000000B					
TA	Timed Access Protection	C7H												11111111B					
I2ADDR	I2C address	C1H	ADDR[7:1]								GC		00000000B						
I2CON	I2C Control register	C0H	(C7) -	(C6) I2CEN	(C5) STA	(C4) STO	(C3) SI	(C2) AA	(C1) -	(C0) -				0000 0000B					
I2TOC	I2C Time-out Counter register	BFH	-	-	-	-	-	I2TOCEN	DIV	I2TOF				0000 0000B					
I2CLK	I2C Clock Rate	BEH	I2CLK[7:0]											0000 0000B					
I2STA	I2C Status Register	BDH	I2STA[7:3]					0	0	0				1111 1000B					
I2DAT	I2C Data Register	BCH	I2DAT[7:0]											00000000B					
SADEN	Slave address mask	B9H	SADEN[7:0]											0000 0000B					
IP	Interrupt priority	B8H	(BF) PCAP	(BE) PADC	(BD) PBOD	(BC) PS	(BB) PT1	(BA) PX1	(B9) PT0	(B8) PX0				0000 0000B					
IPH	Interrupt high priority	B7H	PCAPH	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H				0000 0000B					
P2M2	Port 2 output mode 2	B6H	P2M2[7:0]											00000000B					
P2M1	Port 2 output mode 1	B5H	P2M1[7:0]											0000 0000B					
P1M2	Port 1 output mode 2	B4H	P1M2.7	P1M2.6	-	P1M2.4	P1M2.3	P1M2.2	P1M2.1	P1M2.0				0000 0000B					
P1M1	Port 1 output mode 1	B3H	P1M1.7	P1M1.6	-	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0				0000 0000B					
P0M2	Port 0 output mode 2	B2H	P0M2[7:0]											0000 0000B					
P0M1	Port 0 output mode 1	B1H	P0M1[7:0]											0000 0000b					
P3	Port3	B0H	-	-	-	-	-	-	(B1) X1	(B0) X2 CLKOUT				00000011B					
ISPCN	ISP Control Register	AFH	ISPA17	ISPA16	FOEN	FCEN	FCTRL3	FCTRL2	FCTRL1	FCTRL0				0011 0000B					
ISPFD	ISP Flash Data Register	AEH	ISPFD[7:0]											0000 0000B					
WDCON <sup>[4]</sup>	Watch-Dog control1	ABH	-	-	-	-	-	-	-	EWRST				0000 0000B					
SADDR	Slave address	A9H	SADDR[7:0]											00000000B					
IE	Interrupt enable	A8H	(AF) EA	(AE) EADC	(AD) EBOD	(AC) ES	(AB) ET1	(AA) EX1	(A9) ET0	(A8) EX0				0000 0000B					
ISPAH	ISP Flash Address High-byte	A7H	ISPAH[7:0]											0000 0000B					
ISPAL	ISP Flash Address Low-byte	A6H	ISPAL[7:0]											0000 0000B					
ISPTRG <sup>[4]</sup>	ISP Trigger Register	A4H	-	-	-	-	-	-	-	ISPGO				00000000B					
PMCR <sup>[2][4]</sup>	Power Monitor Control Register	A3H	BODEN	BOV	-	BORST	BOF	-	-	-	Power-on CCOC 100XB BOR reset UU0U 100XB Other reset UU0U 000XB								
AUXR1	AUX function register	A2H	SPI_Sel	UART_Sel	-	-	DisP26	-	0	DPS				0000 0000B					

表7-2.N79E815A/814A/8132ASFR功能描述和复位值

符号	定义	地址	MSB								LSB				复位值 <sup>[1]</sup>		
			(97) P27	(96) P26	(95) P25	(94) P24	(93) P23	(92) P22	(91) P21	(90) P20	(97) P17	(96) P16	(94) P14	(93) P13		(92) P12	(91) P11
P2	Port 2	A0H															1111 1111B
CHPCON <sup>[4]</sup>	Chip Control	9FH	SWRST	ISPF (Read only)	LDUE	-	-	-	BS <sup>[3]</sup>	ISPEN	Power-ON 0000 00C0B Other reset 000000C0B						
SHBDA <sup>[4]</sup>	High-byte Data Flash Start Address	9CH	SHBDA[7:0], SHBDA Initial by CHBDA										Power ON CCCC CCCCB Other Reset UUUU UUUUB				
SBUF	Serial buffer	99H	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0	0000 0000B						
SCON	Serial control	98H	(9F) SM0/FE	(9E) SM1	(9D) SM2	(9C) REN	(9B) TB8	(9A) RB8	(99) TI	(98) RI	0000 0000B						
P3M2	Port 3 output mode 2	97H	-	-	-	-	-	ENCLK	P3M2.1	P3M2.0	00000000B						
P3M1	Port 3 output mode 1	96H	P3S	P2S	P1S	P0S	T1OE	T0OE	P3M1.1	P3M1.0	00000000B						
DIVM	CPU Clock Divide Register	95H	DIVM[7:0]								0000 0000B						
CAPCON2	Input capture control 2	94H	-	ENF2	ENF1	ENF0	-	-	-	-	0000 0000B						
CAPCON1	Input capture control 1	93H	-	-	CAP2LS1[2:0]		CAP1LS1[2:0]		CAP1LS1[2:0]			0000 0000B					
CAPCON0	Input capture control 0	92H	-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0	0000 0000B						
P1	Port 1	90H	(97) P17	(96) P16	-	(94) P14	(93) P13	(92) P12	(91) P11	(90) P10	1111 1111B						
CKCON	Clock control	8EH	-	-	-	T1M	T0M	-	-	-	0000 0000B						
TH1	Timer high 1	8DH	TH1[7:0]								0000 0000B						
TH0	Timer high 0	8CH	TH0[7:0]								0000 0000B						
TL1	Timer low 1	8BH	TL1[7:0]								0000 0000B						
TL0	Timer low 0	8AH	TL0[7:0]								0000 0000B						
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	0000 0000B						
TCON	Timer control	88H	(8F) TF1	(8E) TR1	(8D) TF0	(8C) TR0	(8B) IE1	(8A) IT1	(89) IE0	(88) IT0	0000 0000B						
PCON	Power control	87H	SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL	Power-on 0001 0000B Other reset 000u 0000B						
DPH	Data pointer high	83H	DPH[7:0]								0000 0000B						
DPL	Data pointer low	82H	DPL[7:0]								0000 0000B						
SP	Stack pointer	81H	SP[7:0]								0000 0111B						
P0	Port 0	80H	(87) P07	(86) P06	(85) P05	(84) P04	(83) P03	(82) P02	(81) P01	(80) P00	1111 1111B						

注 标注为 "-" 的位必须保留初始状态，用户不得自行更改。

注:

[1.] ( ) 表示可位寻址特殊功能寄存器。

[2.] BODEN, BOD 和 BORST 在上电复位时由 CONFIG2 初始化，其他复位时保持不变。如果 BODEN=1, LVF 将在上电复位时自动由硬件置位，其他复位保持不变。

[3.] 由上电复位初始化。WDTEN=/CWDTEN; BS=/CBS;

[4.] 带 TA-保护。(Time Access Protection)

[5.] 注 "C" 表示该位由配置位定义; "U" 表示该位在除上电复位外的其他复位时保持不变。

[6.] 复位值符号描述。0: logic 0, 1: logic 1, U: 不改变, X, C: 由配置位初始化。

## 8 通用80C51 系统控制

### A 或 ACC –累加器 (可位寻址)

7	6	5	4	3	2	1	0
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: E0H

复位值: 0000 0000B

位	名称	描述
7:0	ACC[7:0]	<b>累加器</b> A 或 ACC寄存器是标准的8051累加器用于算术运算

### B – B 寄存器(可位寻址)

7	6	5	4	3	2	1	0
B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: F0H

复位值: 0000 0000B

位	名称	描述
7:0	B[7:0]	<b>B寄存器</b> B寄存器是标准8051的另一个累加器, 主要用于MUL 和 DIV 操作

### SP – 堆栈指针

7	6	5	4	3	2	1	0
SP[7:0]							
r/w							

地址: 81H

复位值: 0000 0111B

位	名称	描述
7:0	SP[7:0]	<b>对栈指针</b> 对栈指针存储暂存器堆开始的地址. 使用PUSH或CALL指令, 在数据存储之前地址递增. 注SP的默认值为07H. 使堆由08H开始.

### DPL – 数据指针低字节

7	6	5	4	3	2	1	0
DPL[7:0]							
r/w							

地址: 82H

复位值: 0000 0000B

位	名称	描述
7:0	DPL[7:0]	<b>数据指针低字节.</b> 标准8051的16位数据指针低字节. DPL与DPH 组成16-位数据指针DPTR 以寻址非易失内存或程序内存.



## DPH – 数据指针高字节

7	6	5	4	3	2	1	0
DPH[7:0]							
r/w							

地址: 83H

复位值: 0000 0000B

位	名称	描述
7:0	DPH[7:0]	数据指针高字节。 标准8051的16位数据指针高字节. DPL与DPH 组成16-位数据指针DPTR 以寻址非易失内存或程序内存.

## PSW – 程序状态字 (可位寻址)

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	F1	P
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r

地址: D0H

复位值: 0000 0000B

位	名称	描述																				
7	CY	<b>高位进位标志</b> 进行加法或减法操作时, 当前运算需要向高位进位或借位时, CY将置位, 否则清零。在进行MUL 或 DIV运算时, CY始终为0。 如果初始BCD数大于100, CY受DA A指令影响。在CJNE指令中, 如果第一个无符号数的值小于第二个, 则CY置1, 否则清0。																				
6	AC	<b>辅助进位标志</b> 当前运算导致从半字节的低序第4位进位或借位, 该位置位, 否则清零。																				
5	F0	<b>用户标志0.</b> 可由用户置位或清零的通用标志。																				
4	RS1	<b>寄存器页选择位.</b> 这两位用于选择不同的页。 <table border="1" style="margin-left: 20px;"> <tr> <td>RS1</td> <td>RS0</td> <td>寄存器页</td> <td>RAM 地址</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>00~07H</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>08~0FH</td> </tr> <tr> <td>1</td> <td>0</td> <td>2</td> <td>10~17H</td> </tr> <tr> <td>1</td> <td>1</td> <td>3</td> <td>18~1FH</td> </tr> </table>	RS1	RS0	寄存器页	RAM 地址	0	0	0	00~07H	0	1	1	08~0FH	1	0	2	10~17H	1	1	3	18~1FH
RS1	RS0		寄存器页	RAM 地址																		
0	0		0	00~07H																		
0	1		1	08~0FH																		
1	0	2	10~17H																			
1	1	3	18~1FH																			
3	RS0																					
2	OV	<b>溢出标志</b> OV用于标示发生溢出。对于加法指令 ADD或ADDC指令中, 如果位6有进位而位7没进位, 或者位7有进位而位6没有进位, 则溢出标志置“1”, 反之清“0”。OV 也用于标示有符号数累加结果, 当两个正数相加, 或两个负数相加结果为正数时OV为1。对于减法指令SUBB, 当位6发生借位而位7没有, 或者位7发生借位而位6没有借位, 则溢出标志置“1”, 反之清“0”。OV也用于标示两个数相减时, 当一个正数加一负数结果为负, 或两个负数相减结果为负时。 对于MUL乘法指令, 当结果大于255 (00FFH)时, OV置1。反之清0。 对于DIV除法指令, 通常情况下OV为0。除非当B设定值为00H, 则A和B的返回值为随机值, 同时OV 置1。																				
1	F1	<b>用户标志1</b> 可由用户置位或清零的通用标志。																				
0	P	<b>奇偶标志</b> 当累加结果为奇数时, 该标志置1, 偶数时清0。其执行奇偶校验。																				

表8-1. 指令对标志位的影响

指令	CY	OV	AC	指令	CY	OV	AC
ADD	X <sup>[1]</sup>	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	0	X		ANL C, /bit	X		
DIV	0	X		ORL C, bit	X		
DA A	X			ORL C, /bit	X		
RRC A	X			MOV C, bit	X		
RLC A	X			CJNE	X		
SETB C	1						

[1] X表示根据指令的结果变化

## PCON – 电源控制

7	6	5	4	3	2	1	0
SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL
r/w	r/w	-	r/w	r/w	r/w	r/w	r/w

地址：87H

复位值：见 表7-2.N79E815A/814A/8132ASFR功能描述和复位值

位	名称	描述
3	GF1	通用标志1 通用标志可由用户置位或清零。
2	GF0	通用标志0 通用标志可由用户置位或清零。

标准8051仅一组DPTR，但N79E815A/814A/8132A支持两组DPTR，两组共用DPH/DPL寄存器，通过附加功能选择寄存器1选择当前使用DPTR或DPTR1。

## AUXR1 – 附加功能选择寄存器1

7	6	5	4	3	2	1	0
SPI_Sel	UART_Sel	-	-	DisP26	-	0	DPS
r/w	r/w	-	-	r/w	-	r	r/w

地址：A2H

复位值：0000 0000B

位	名称	描述
0	DPS	双数据指针选择 0: 选择标准8051单DPTR 1: 选择启用DPTR1，此时DPH和DPL的值为DPTR1。

## 9 通用IO (GPIO) 结构及操作

N79E815A/814A/8132A 有 4组 I/O，端口 0、端口1、端口2和端口3。如果使用片内RC振荡器并配置复位引脚为普通I/O，N79E815A/814A/8132A系列可支持达25个I/O。所有的I/O口（除P1.2和P1.3外）可配置成下表四种类型。

表9-1. I/O结构设定表

PxM1.y	PxM2.y	端口输入/输出模式
0	0	准双向模式
0	1	推挽输出模式
1	0	输入模式（高阻）
1	1	开漏模式

注: 该表内设置对 P1.2 和 P1.3 不起作用

复位后，所有引脚都默认设定为准双向模式。P1.2和P1.3除外。

为配合I<sup>2</sup>C应用，P1.2 和 P1.3始终为开漏模式。

通过配置寄存器P3M1的P(n)S位，N79E815A/814A/8132A每个I/O口都可定义为TTL电平输入或史密特输入；其中n为0、1或2。当P(n)S置1，端口选为史密特触发输入。

当使用片上RC或外部振荡器作为时钟源时，P3.0 (XTAL2) 可被配置成时钟输出脚，时钟输出的频率为片上RC时钟。

### 9.1 准双向输出配置

N79E815A/814A/8132A系列默认端口与标准8051相同，都为“准双向”模式。这种模式可以直接用作输入或输出。

当端口输出逻辑高时，驱动能力较弱，同时允许外部器件将电平拉低。当引脚被拉低时有强驱动能力，会吸收大电流。在准双向I/O结构中，有三个上拉三极管，为适应不同的应用，当端口锁定在逻辑1时，打开“特弱上拉”在引脚悬空时，“特弱上拉”有很小电流就将引脚拉高。

第二种上拉为“弱上拉”，当外部端口引脚自身处于逻辑1电平时打开。这种上拉提供源电流以使准双向引脚输出1。如果引脚为逻辑1，被外部器件拉低，“弱上拉”关闭，仅有“特弱上拉”打开。此时要将引脚拉低，外部器件要有足够的灌电流（大于 $I_{TL}$ ）以克服“弱上拉”，并使端口的电压低于输入门限电压（低于 $V_{IL}$ ）。

第三种上拉为“强上拉”。这种上拉用于在准双向口引脚上，加速端口电平由逻辑0转为逻辑1的转换速度。当这种情况发生时，强上拉打开两个总线时钟的时间以快速将端口引脚拉高。然后就关闭，“弱上拉”继续保持该端口引脚为高。准双向端口结构如下所示。

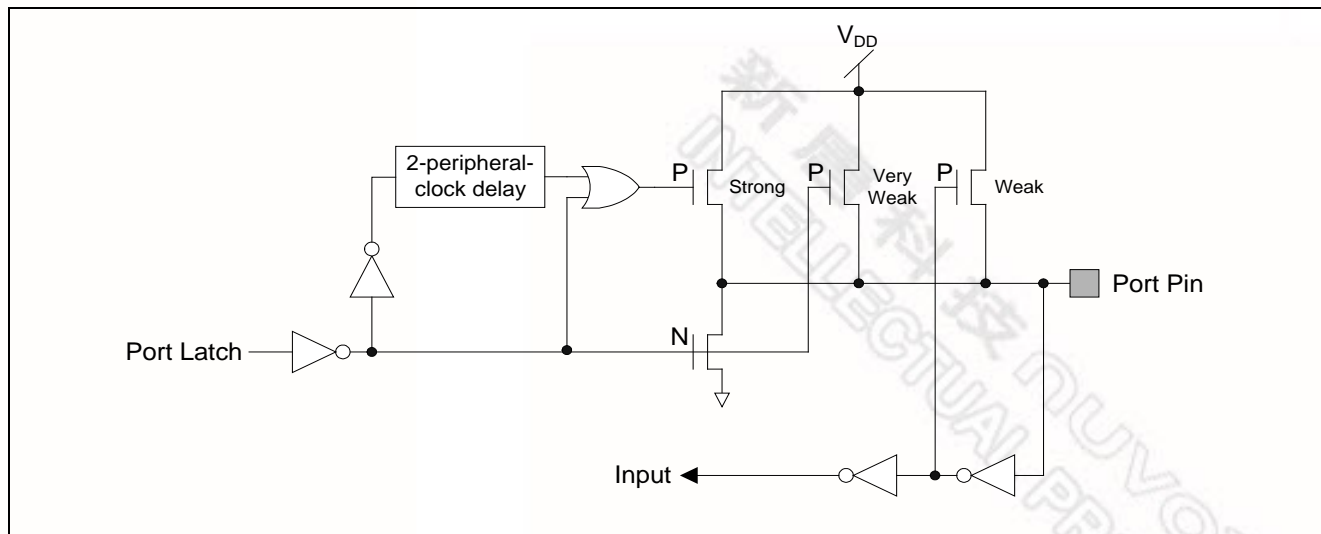


图9-1. 准双向端口结构

### 9.1.1 读-修改-写

在标准8051结构中，执行读 - 改 - 写的步骤需要特别注意。取代传统的架构，读 - 修改 - 写指令改为读取内部端口锁存器（SFR的Px）值，而不是外部端口引脚的状态。这种指令读取端口所存器内的值，修改后写回端口所存器。读 - 修改 - 写指令如下所列

指令	描述
ANL	逻辑与 (ANL Px,A 和 ANL Px,direct)
ORL	逻辑或 (ORL Px,A 和 ORL Px,direct)
XRL	逻辑异或 (XRL Px,A 和 XRL Px,direct)
JBC	为1转跳指令并清除指令 (JBC Px.y,LABEL)
CPL	位取反 (CPL Px.y)
INC	加一指令. (INC Px)
DEC	减一指令. (DEC Px)
DJNZ	减一不为零转跳指令 (DJNZ Px,LABEL)
MOV	Px.y,C Px.y进位
CLR	Px.y Px.y清位
SETB	Px.y Px.y置位

最后三条指令看似不是明显的读-修改-写指令，实际也是读-修改-写指令。可以读整个端口锁定值，修改改变位，写入新的值。

## 9.2 开漏输出配置

开漏输出配置关闭所有内部上拉，当端口锁定为逻辑0时，仅打开端口驱动程序的下拉晶体管。用作逻辑输出时，端口必须配置外部上拉，典型的作法是由电阻拉到VDD。这种模式的下拉与准双向模式相似。开漏端口配置如下所示

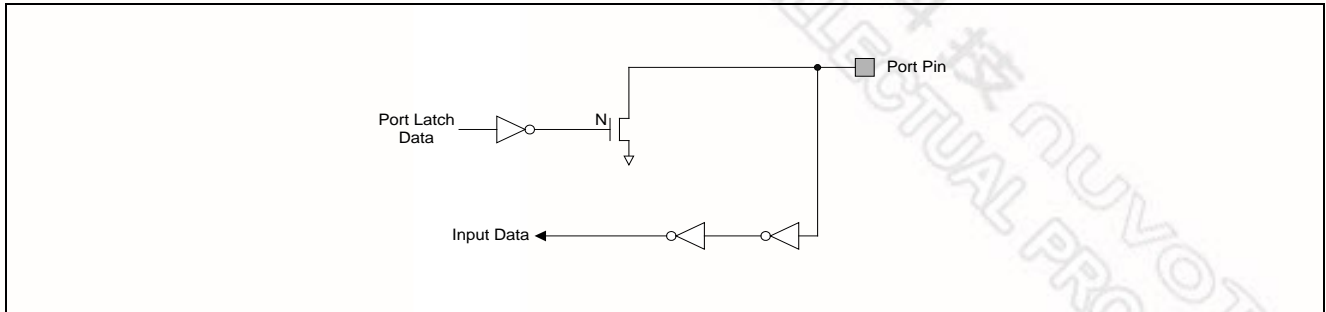


图9-2 开漏输出结构

## 9.3 推挽输出配置

推挽输出配置与开漏和准双向输出模式有相同的下拉结构。当端口锁定为1时，提供持续的强上拉。推挽输出模式用于需要从端口输出大电流从时的应用，推挽端口配置如图9-3所示。P1.2 (SCL) 和 P1.3 (SDA)不能配置成推挽结构，只能配置成开漏输出。通过写1到相应的端口寄存器可用作输出，此外，如果接外部晶振时，端口引脚P3.0 和 P3.1 无法作输入和输出，详见振荡器章节的描述。

复位后，所有端口为准双向模式，端口引脚驱动为高，为准双向模式，则不能有较大的源电流。

N79E815A/814A/8132A 系列的输出可用作驱动38 mA 的LED 输出。然而，该值为所有输出电流的最大值。

N79E815A/814A/8132A系列的所有端口引脚受电压转换速率控制输出，该功能限制因输出信号快速切换产生的噪声，电压转换速率由工厂设置大约为10 ns上升和下降一次。寄存器P3M1并非用于配置P3.1 和 P3.0而另作他用，这些位用以使能每个端口的史密特触发输入，使能定时器0和定时器1的输出，如果使用内部RC或外部时钟输入时，使能时钟输出。N79E815A/814A/8132A系列的每个I/O 端口可被选作TTL 电平输入或带迟滞功能史密特输入，包括P1.2 和 P1.3 。

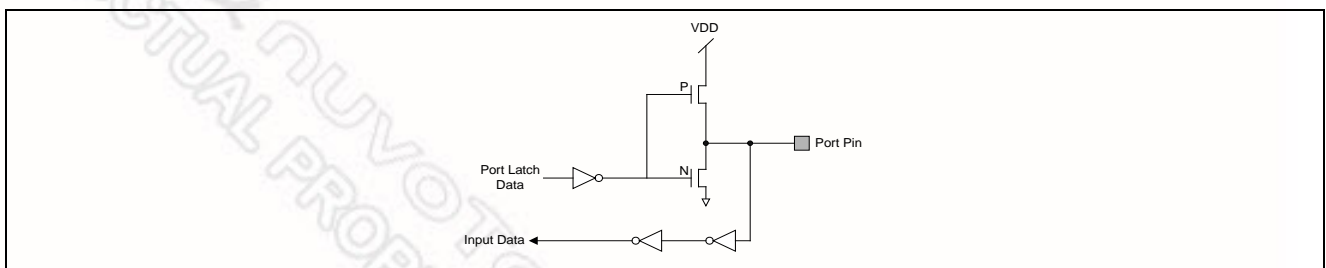


图9-3 推挽输出结构



## 9.4 输入配置

配置成该模式时，该端口只能输入。该引脚为高阻抗。

### P0 – 端口0 (可位寻址)

7	6	5	4	3	2	1	0
P07	P06	P05	P04	P03	P02	P01	P00
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址：80H

复位值：1111 1111B

位	名称	描述
7:0	P0[7:0]	端口0 端口0为8位可位寻址端口。

### P1 – 端口1 (可位寻址)

7	6	5	4	3	2	1	0
P17	P16	-	P14	P13	P12	P11	P10
r/w	r/w	-	r/w	r/w	r/w	r/w	r/w

地址：90H

复位值：1111 1111B

位	名称	描述
7:0	P1[7:0]	端口1 端口1为8位可位寻址端口。 配合I <sup>2</sup> C接口，P1.2和P1.3始终为开漏模式，其余默认为双向模式。

### P2 – 端口2 (可位寻址)

7	6	5	4	3	2	1	0
P27	P26	P25	P24	P23	P22	P21	P20
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址：A0H

复位值：1111 1111B

位	名称	描述
7:0	P2[7:0]	端口2 端口2为8位可位寻址端口。

### P3 – 端口3 (可位寻址)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	P31	P30
-	-	-	-	-	-	r/w	r/w

地址：B0H

复位值：0000 0011B

位	名称	描述
7:2	-	保留
1	P3.1	外部晶振脚1 或普通端口
0	P3.0	外部晶振脚2 或时钟输出脚 或普通端口

**P0M1 – 端口0输出模式配置1**

7	6	5	4	3	2	1	0
P0M1.7	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: B1H

复位值: 0000 0000B

**P0M2 – 端口0输出模式配置2**

7	6	5	4	3	2	1	0
P0M2.7	P0M2.6	P0M2.5	P0M2.4	P0M2.3	P0M2.2	P0M2.1	P0M2.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: B2H

复位值: 0000 0000B

**P1M1 – 端口1输出模式配置1**

7	6	5	4	3	2	1	0
P1M1.7	P1M1.6	-	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0
r/w	r/w	-	r/w	r/w	r/w	r/w	r/w

地址: B3H

复位值: 0000 0000B

**P1M2 – 端口1输出模式配置2**

7	6	5	4	3	2	1	0
P1M2.7	P1M2.6	-	P1M2.4	P1M2.3	P1M2.2	P1M2.1	P1M2.0
r/w	r/w	-	r/w	r/w	r/w	r/w	r/w

地址: B4H

复位值: 0000 0000B

**P2M1 – 端口2输出模式配置1**

7	6	5	4	3	2	1	0
P2M1.7	P2M1.6	P2M1.5	P2M1.4	P2M1.3	P2M1.2	P2M1.1	P2M1.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: B5H

复位值: 0000 0000B

**P2M2 – 端口2输出模式配置2**

7	6	5	4	3	2	1	0
P2M2.7	P2M2.6	P2M2.5	P2M2.4	P2M2.3	P2M2.2	P2M2.1	P2M2.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: B6H

复位值: 0000 0000B

**端口输出配置设置:**

PxM1.y	PxM2.y	端口输入/输出模式
0	0	准双向模式
0	1	推挽输出模式
1	0	输入(高阻)模式
1	1	开漏模式

## P3M1 – 端口3输出模式配置1

7	6	5	4	3	2	1	0
P3S	P2S	P1S	P0S	T1OE	T0OE	P3M1.1	P3M1.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: 96H

复位值: 0000 0000B

位	名称	描述
7	P3S	使能端口3的史密特触发输入
6	P2S	使能端口2的史密特触发输入
5	P1S	使能端口1的史密特触发输入
4	P0S	使能端口0的史密特触发输入
1	P3M1.1	端口3.1及端口3.0输出模式配置位1, 参考表9-1 I/O结构设定表
0	P3M1.0	

## P3M2 – 端口3输出模式配置2

7	6	5	4	3	2	1	0
-	-	-	-	-	ENCLK	P3M2.1	P3M2.0
-	-	-	-	-	r/w	r/w	r/w

地址: 97H

复位值: 0000 0000B

位	名称	描述
7:3	-	保留
0	ENCLK	使能时钟输出到XTAL2 引脚 (P3.0) 如果时钟为内部RC, P3.0的频率为内部RC/4 (22.1184MHz/4).
1	P3M2.1	端口 3.1 及端口 3.0 输出模式配置位 2,参考表 9-1 I/O 结构设定表
0	P3M2.0	

## 10 定时器/计数器

N79E815A/814A/8132A系列有3个16位可编程定时器/计数器。

### 10.1 定时器/计数器0 和 1

N79E815A/814A/8132A系列的定时器/计数器0 和 1 是两个16位定时器/计数器。每个都由两个8位的寄存器组成16计数寄存器。对于定时器/计数器0，为高8位寄存器TH0、低8位寄存器TL0。同样定时器/计数器1也有两个8位寄存器，TH1 和TL1。TCON 和 TMOD 可以配置定时器/计数器0和1的模式。

与常见的定时器/计数器相比，还有一个定时器0或定时器1的溢出端口电位翻转输出使能。当定时器发生溢出时，端口电位翻转输出可配置成根据T0 或 T1引脚自动翻转输出。

将它们设置为定时器后，定时器将对时钟周期计数。时钟源可以是系统时钟的12分频或是系统时钟的4分频。在计数器模式下，每当检测到外部计数输入脚上的负电平跳变（T0针对定时器0，T1针对定时器1），计数寄存器的内容就会加一。T0和T1上的电平在每个机器周期的C4态被采样，如果在一个机器周期采样到高电平，在下一个机器周期采样到低电平，那么就会确认一个电平由高到低的跳变，计数器寄存器指针加一。由于需要2个机器周期来确认管脚上的电平负跳变，因此外部输入信号的最大频率是主频的24分之一。无论是定时器还是计数器，计数寄存器都在机器周期的C3态加一。因此在定时器模式下，在T0 和T1 脚上检测到的电平负跳变会在紧跟着检测到该电平跳变后的那个机器周期中使计数器加1。

TMOD寄存器中的“C/T”位决定工作在定时器模式还是计数器模式。每个定时器/计数器都有它自己的模式选择位；TMOD中用第2位选择定时器/计数器0 的功能、第6位来选择定时器/计数器1的功能。此外每个定时器/计数器都可以选定4种运行方式中的一种来运行。由TMOD中的M0 和M1位来选择定时器的工作模式。

N79E815A/814A/8132A系列可以像标准8051/52家族一样，计数速率为时钟的1/12，或进行快速模式，计数速率为时钟的1/4。速率由CKCON的T0M 和 T1M 位控制，在使用标准8051/52速率时，默认值为0。

## CKCON – 时钟控制

7	6	5	4	3	2	1	0
-	-	-	T1M	T0M	-	-	-
-	-	-	r/w	r/w	-	-	-

地址：8EH

复位值：0000 0000B

位	名称	描述
7:5	-	保留-
4	T1M	定时器 1 时钟选择: 0: 定时器 1 的时钟选择为 1/12 系统时钟. 1: 定时器1的时钟选择为1/4系统时钟.
3	T0M	定时器 0 时钟选择: 0: 定时器 0 的时钟选择为 1/12 系统时钟. 1: 定时器0的时钟选择为1/4系统时钟
2:0	-	保留-

## TMOD – 定时器0 和 1 模式配置

7	6	5	4	3	2	1	0
GATE	C/T	M1	M0	GATE	C/T	M1	M0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址：89H

复位值：0000 0000B

位	名称	描述															
7	GATE	定时器 1 门控制。 0 = 当TR1=1时，定时器 1 时钟运行不管 $\overline{INT1}$ 的逻辑电平 1 = 当TR1=1和 $\overline{INT1}$ 为逻辑1，定时器1 运行															
6	C/T	定时器 1 计数器/定时器选择。 0 = 定时器1随内部时钟而递增。 1 = 定时器1 随外部引脚T1的下降沿递增															
5	M1	定时器1模式选择															
4	M0	<table border="1"> <thead> <tr> <th>M1</th> <th>M0</th> <th>定时器1 模式</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>模式0: 8位定时器/计数器带5位预分频 <math>\square</math>TL1[4:0])</td> </tr> <tr> <td>0</td> <td>1</td> <td>模式1: 16位定时器/计数器</td> </tr> <tr> <td>1</td> <td>0</td> <td>模式2: 8位定时器/计数器带自动从TH1重载模式</td> </tr> <tr> <td>1</td> <td>1</td> <td>模式3: 定时器1停止</td> </tr> </tbody> </table>	M1	M0	定时器1 模式	0	0	模式0: 8位定时器/计数器带5位预分频 $\square$ TL1[4:0])	0	1	模式1: 16位定时器/计数器	1	0	模式2: 8位定时器/计数器带自动从TH1重载模式	1	1	模式3: 定时器1停止
M1	M0	定时器1 模式															
0	0	模式0: 8位定时器/计数器带5位预分频 $\square$ TL1[4:0])															
0	1	模式1: 16位定时器/计数器															
1	0	模式2: 8位定时器/计数器带自动从TH1重载模式															
1	1	模式3: 定时器1停止															
3	GATE	定时器 0 门控制。 0 = 当TR0=1时，定时器0 时钟运行不管 $\overline{INT0}$ 的逻辑电平。 1 = 当TR0=0和 $\overline{INT0}$ 为逻辑1，定时器0 运行。															
2	C/T	定时器 0 计数器/定时器选择。 0 = 定时器0随内部时钟而递增。 1 = 定时器0 随外部引脚T1的下降沿递增															
1	M1	定时器0模式选择															
0	M0	<table border="1"> <thead> <tr> <th>M1</th> <th>M0</th> <th>定时器0 模式</th> </tr> </thead> <tbody> <tr> <td>0</td> <td><math>\square</math></td> <td>模式0: 8位定时器/计数器带5位预分频 (TL1[4:0])</td> </tr> <tr> <td>0</td> <td>1</td> <td>模式1: 16位定时器/计数器</td> </tr> <tr> <td>1</td> <td>0</td> <td>模式2: 8位定时器/计数器带自动从TH0重载模式在</td> </tr> <tr> <td>1</td> <td>1</td> <td>Mode 3: TL0 / TH1 分别单独作为一个8位定时器/计数器</td> </tr> </tbody> </table>	M1	M0	定时器0 模式	0	$\square$	模式0: 8位定时器/计数器带5位预分频 (TL1[4:0])	0	1	模式1: 16位定时器/计数器	1	0	模式2: 8位定时器/计数器带自动从TH0重载模式在	1	1	Mode 3: TL0 / TH1 分别单独作为一个8位定时器/计数器
M1	M0	定时器0 模式															
0	$\square$	模式0: 8位定时器/计数器带5位预分频 (TL1[4:0])															
0	1	模式1: 16位定时器/计数器															
1	0	模式2: 8位定时器/计数器带自动从TH0重载模式在															
1	1	Mode 3: TL0 / TH1 分别单独作为一个8位定时器/计数器															

**TCON –定时器 0 和 1 控制位 (可位寻址)**

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: 88H

复位值: 0000 0000B

位	名称	描述
7	TF1	定时器 1 溢出标志。 在定时器1溢出时该位置1。当程序响应定时器1中断执行相应的中断服务程序时, 该位自动清0。软件也可对其写1或写0
6	TR1	定时器 1 启动控制。 0 = 定时器1 中止. 清该位将中止定时器1和当前计数将保存在TH1 和 TL1。 1 = 使能定时器1.
5	TF0	定时器 0 溢出标志。 在定时器0溢出时该位置1。当程序响应定时器0中断执行相应的中断服务程序时, 该位自动清0。软件也可对其写1或写0
4	TR0	定时器 0 启动控制。 0 = 定时器0 中止. 清该位将中止定时器0和当前计数将保存在TH0和 TL0。 1 = 使能定时器0.

**TL0 –定时器0 低字节**

7	6	5	4	3	2	1	0
TL0[7:0]							
r/w							

地址: 8AH

复位值: 0000 0000B

位	名称	描述
7:0	TL0[7:0]	定时器 0 低字节 寄存器TL0是定时器0的16位数值的低字节

**TH0 –定时器0高字节**

7	6	5	4	3	2	1	0
TH0[7:0]							
r/w							

地址: 8CH

复位值: 0000 0000B

位	名称	描述
7:0	TH0[7:0]	定时器 0 高字节 寄存器TH0是定时器0的16位数值的高字节

**TL1 –定时器1低字节**

7	6	5	4	3	2	1	0
TL1[7:0]							
r/w							

地址: 8BH

复位值: 0000 0000B

位	名称	描述
7:0	TL1[7:0]	定时器 1 低字节 寄存器TL1是定时器1的16位数值的低字节



## TH1 – 定时器1高字节

7	6	5	4	3	2	1	0
TH1[7:0]							
r/w							

地址：8DH

复位值：0000 0000B

位	名称	描述
7:0	TH1[7:0]	定时器 1 高字节 寄存器TH1是定时器1的16位数值的高字节

## P3M1 – 端口3输出模式配置1

7	6	5	4	3	2	1	0
P3S	P2S	P1S	P0S	T1OE	T0OE	P3M1.1	P3M1.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址：96H

复位值：0000 0000B

位	名称	描述
3	T1OE	当定时器1溢出时，P0.7口电位翻转。 该端口方波输出频率等于定时器1溢出频率的一半。
2	T0OE	当定时器0溢出时，P1.2口电位翻转。 该端口方波输出频率等于定时器0溢出频率的一半。

## 10.1.1 模式 0 (13 位 定时器)

模式0下，定时器/计数器为由8位的THx和TLx的低5位组合，8位加5位带除以32预分频组成，TLx的高3位忽略。TLx会在时钟源的负跳变处加一，当TLx的第五位由1变0后，THx开始计数。当THx的数值由FF变为00以后，TCON中的溢出标志位TFx会置位，当TRx置位且GATE为0或 $\overline{INTx}$ 为1时，计数输入才有效。C/T=0时，定时器/计数器对时钟周期进行计数，C/T=1时对P1.2(T0)以及P0.7 (T1) 上的1到0跳变进行计数。当13位的定时器计数值变为1FFFH后，下一次计数会使其变为0000H。此时相关的溢出标志位置位如果中断打开，此时还会产生一个定时器中断。

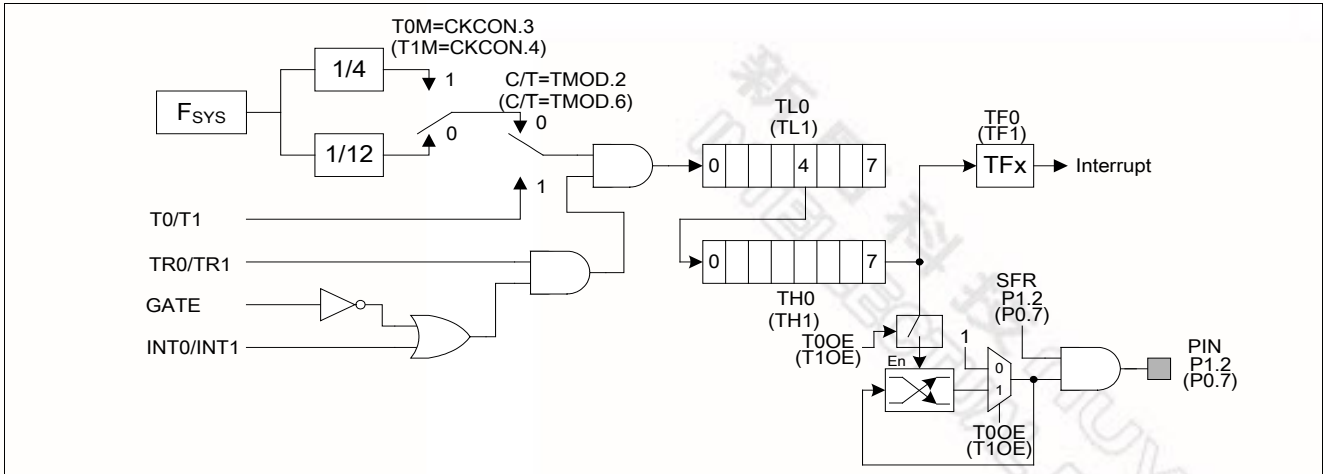


图10-1.定时器/计数器0 模式0 结构

### 10.1.2 模式 1 (13 位定时器)

模式1与模式0 非常相似，只是模式1下定时器/计数器为16位的，而非13位。就是说是用THx和TLx的全部16位来计数。当计数值由FFFFH向0000H翻转后，相应的溢出标志TFx置1，并产生中断。

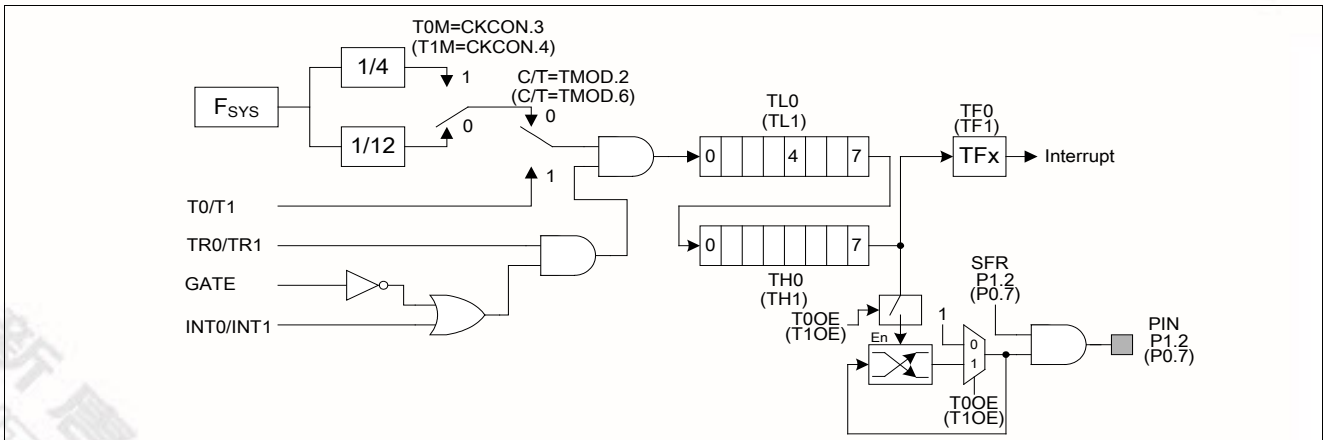


图10-2.定时器/计数器0和1 模式1 结构

### 10.1.3 模式 2 (8 位自动重载定时器)

模式2下定时器/计数器为自动重装模式。此模式下TLx是一个8位的计数器，THx保存重装计数值。当TLx由FFH向00H溢出后，TCON中的TFx标志置位THx中内容重装至TLx，继续计数过程。重装过程中THx内的值保持不变。该特征最好地适用于UART波特率发生器，不需要连续软件介入。注：仅有定时器1可以用作UART的波特率源。正确设置GATE和 $\overline{INTx}$  引脚及TRx位，使能计数。GATE 和 $\overline{INTx}$  引脚的功能与模式0和1相同。



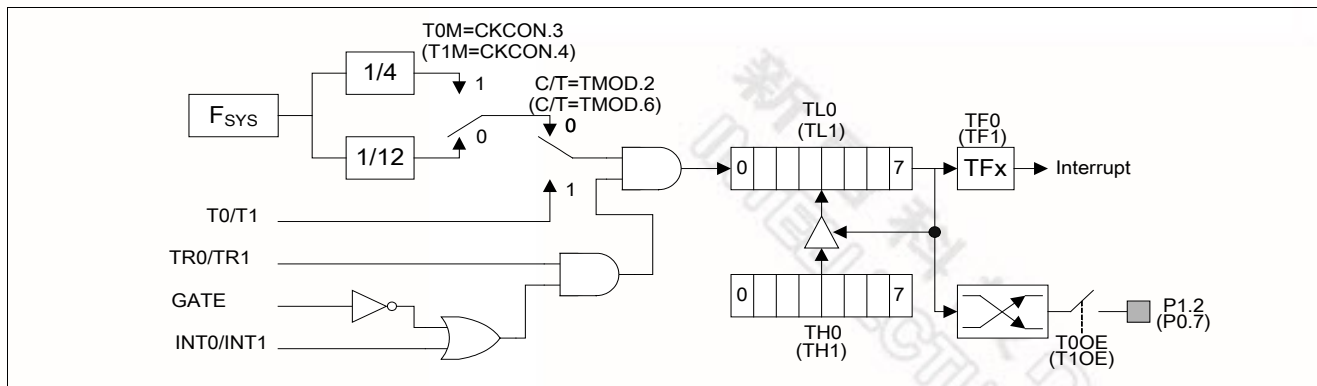


图10-3. 定时器/计数器0和1 模式2 结构

### 10.1.4 模式3 (两组独立8位定时器)

模式3 有着不同的工作方式。对定时器/计数器1 来说模式3 会将其停止；对定时器/计数器0来说模式3下  $TL0$  和  $TH0$  是 2个独立的8位计数寄存器。下图表示这种模式下的逻辑关系。模式3下  $TL0$  用定时器0的控制位：如  $C/\bar{T}$ 、 $GATE$ 、 $TR0$ 、 $\overline{INT0}$  和  $TF0$ 。  $TL0$  可以用来对时钟周期来计数（时钟源的  $1/12$  或  $1/4$ ）以及对  $T0$  脚上的1到0 跳变计数。  $TH0$  只能对内部时钟源计数，并使用定时器/计数器1的控制位（ $TR1$ 和 $TF1$ ）。当需要额外的8位定时器时可以使用模式3。当定时器0处于模式3时，定时器1依然可以工作在模式0、1、2下，但它的灵活性受到限制。虽然基本功能得以维持，但已不能对  $TF1$  和  $TR1$  进行控制。此时定时器1依然可以使用  $GATE$  及  $INT1$  脚。另外可以通过将其放入或离开模式3的方式来打开或关闭它。它同样可以用作串行口的波特率发生器。

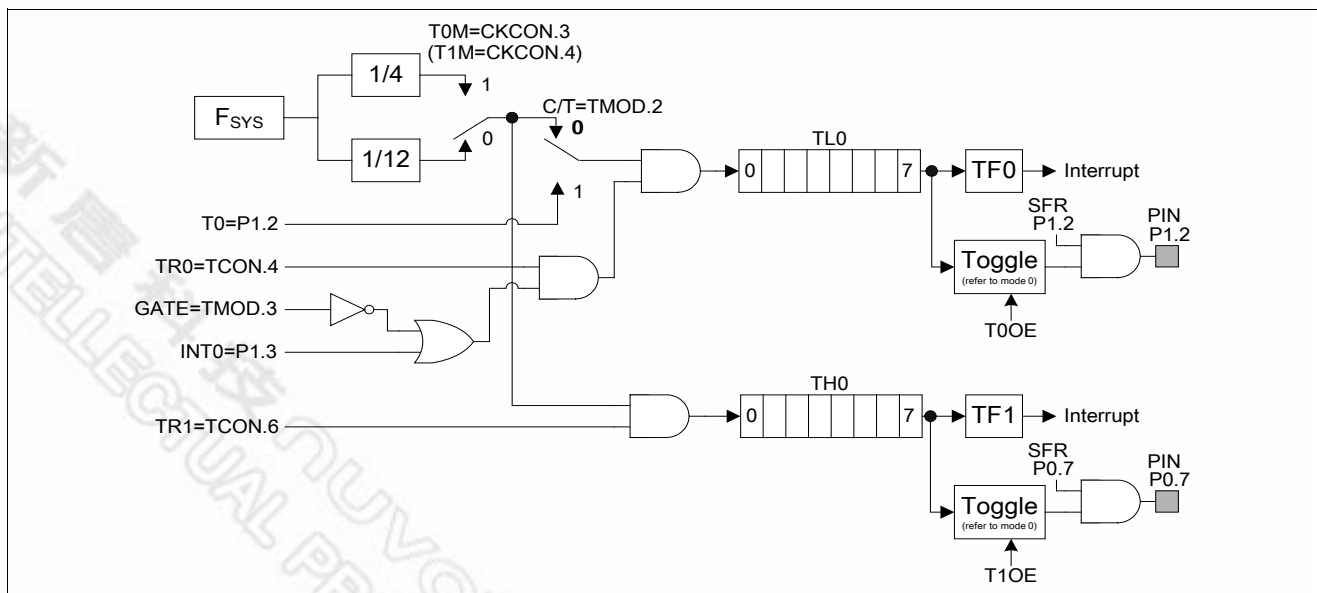


图10-4. 定时器/计数器0 模式3 结构

## 10.2 定时器/计数器 2

定时器 2 是一个16位的向上计数器，高8位寄存器为TH2，低8位寄存器为TL2。通过设置RCOMP2H和RCOMP2L，设定定时器2工作在比较模式或自动重载模式。另外3路输入捕获模块使寄存器2可用于检测输入脉冲的宽度和周期。3路输入捕获结果分别存储在C0H 和 C0L, C1H 和 C1L, C2H 和 C2L。由系统时钟提供给定定时器2的时钟源由时钟除频器预分频，使该捕获模块有8种不同捕获时间长度。当TR2 (T2CON.2)为1时时钟开始计数，当TR2为0，时钟关闭。下列寄存器实现定时器2的相关功能

### T2CON – 定时器 2 控制 (可位寻址)

7	6	5	4	3	2	1	0
TF2	-	-	-	-	TR2	-	CP/RL2
r/w	-	-	-	-	r/w	-	r/w

地址：C8H

复位值：0000 0000B

位	名称	描述
7	TF2	<b>定时器 2 溢出标志</b> 在定时器2溢出或比较匹配时该位置1。当使能定时器2中断和全局中断时，设置该位将使CPU执行定时器2中断服务子程序，该位不能由硬件自动清零，必须软件清零。
6:3	-	保留
2	TR2	<b>定时器 2 运行控制。</b> 0 = 中止定时器2。清该位将中止定时器2和当前计数将保存在TH2 和 TL2。 1 = 打开定时器2
1	-	保留。
0	CP/RL2	<b>定时器 2 捕获或重加载选择。</b> 该位选择定时器2的功能是比较模式还是自动重加载模式 0 = 定时器2为自动重加载 1 = 定时器2为比较模式

### T2MOD – 定时器2模式

7	6	5	4	3	2	1	0
LDEN	T2DIV[2:0]			CAPCR	COMPCR	LDTs[1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址：C9H

复位值：0000 0000B

位	名称	描述
7	LDEN	<b>使能自动重加载。</b> 0 = 在定时器2溢出或发生输入捕获事件时，禁止将RCOMP2H 和 RCOMP2L加载到TH2 和 TL2 1 = 在定时器2溢出或发生输入捕获事件时，使能将RCOMP2H 和 RCOMP2L加载到TH2 和 TL2

位	名称	描述
6:4	T2DIV[2:0]	<p>定时器 2 时钟除频.</p> <p>000 = 定时器 2 时钟除频1/4. 001 = 定时器 2 时钟除频1/8. 010 = 定时器 2 时钟除频1/16. 011 = 定时器 2 时钟除频1/32. 100 = 定时器 2 时钟除频1/64. 101 = 定时器 2 时钟除频1/128. 110 = 定时器 2 时钟除频1/256. 111 = 定时器 2 时钟除频1/512.</p>
3	CAPCR	<p>自动清除捕获</p> <p>当输入捕获事件发生时, 该位使能自动清定时器2在TH2与TL2的值. 0 = 发生捕获事件后, 定时器 2继续计数. 1 = 发生捕获事件后, 定时器2的自动清零为0000H</p>
2	COMP2CR	<p>比较匹配自动清零.</p> <p>在比较匹配发生时, 该位使能自动清定时器2的TH2与TL2的值. 0 = 发生比较匹配时, 定时器2继续计数. 1 = 发生比较匹配时, 定时器2自动清零为0000H.</p>
1:0	LDT2S[1:0]	<p>自动加载触发选择.</p> <p>该位选择自动重加载触发事件 00 = 定时器2溢出重加载 01 = 输入捕获0事件发生时重加载 10 = 输入捕获1事件发生时重加载 11 = 输入捕获2事件发生时重加载</p>

#### RCOMP2L – 定时器 2 重加载/比较低字节

7	6	5	4	3	2	1	0
RCOMP2L[7:0]							
r/w							

地址: CAH

复位值: 0000 0000B

位	名称	描述
7:0	RCOMP2L[7:0]	<p>定时器 2 重加载/比较低字节.</p> <p>定时器2配置成比较模式, 该寄存器存储着比较值的低字节. 当配置成自动加载模式时, 该字节保存需重加载的值</p>

#### RCOMP2H – 定时器 2 重加载/比较高字节

7	6	5	4	3	2	1	0
RCOMP2H[7:0]							
r/w							

地址: CBH

复位值: 0000 0000B

位	名称	描述
7:0	RCOMP2H[7:0]	<p>定时器 2 重加载/比较高字节.</p> <p>定时器2配置成比较模式, 该寄存器存储着比较值的高字节. 当配置成自动加载模式时, 保存着重加载的值的高字节</p>

## TL2 – 定时器 2 低字节

7	6	5	4	3	2	1	0
TL2[7:0]							
r/w							

地址: CCH

复位值: 0000 0000B

位	名称	描述
7:0	TL2[7:0]	定时器 2 低字节. 寄存器TL2 为定时器2的16位低字节

## TH2 – 定时器 2 高字节

7	6	5	4	3	2	1	0
TH2[7:0]							
r/w							

地址: CDH

复位值: 0000 0000B

位	名称	描述
7:0	TH2[7:0]	定时器 2 高字节. 寄存器TH2 为定时器2的16位高字节

定时器/计数器2提供三种操作模式，可以通过T2CON 和 T2MOD 的控制位选择，见下表。请注意TH2 和 TL2 分别读写，请用户停止定时器2后读写TH2和TL2. 运行时读写可能会引起不可预见的错误。

表10-1. 定时器 2 操作模式

定时器 2 模式	CP/RL2 (T2CON.0)	LDEN (T2MOD.7)
输入捕获模式	0	0
自动重加载模式	0	1
比较模式	1	X

### 10.2.1 输入捕获模式

定时器2的输入捕获模式. 定时器2通过 $\overline{CP/RL2}$ 和LDEN位清零进入输入捕获模式。输入捕获模块通过寄存器CAPCON0~2配置, 输入捕获模块支持3通道输入(IC0, 和IC1管脚), 和普通端口共用的P1.2, P0.7和P2.0 每个输入通道组成自己的史密特触发输入。每个通道的噪声滤波器通过设置ENF0~2 (CAPCON2[6:4])使能, 可滤除小于4个CPU时钟的输入毛刺。输入捕获0~2 有独立的边沿检测与定时器2的触发边沿共享(通过对CAPCON1配置)支持正边沿捕获, 负边沿捕获, 或双边沿捕获。每个输入捕获通道有自己的使能位CAPEN0~2 (CAPCON0[6:4])。

当使能任何输入捕获通道和所选择的边沿触发发生时, 定时器2的计数值TH2和TL2将被捕获、传输并存储到捕获寄存器CnH 和 CnL。边沿触发也可硬件方式使CAPFn (CAPCON0.n)置1。如果ECPTF (EIE.7)和EA都打开, 将产生中断。三个输入捕获标志共享中断向量, 用户应该检查CAPFn以确定哪个通道有输入捕获。这些标志必须由软件清零。

CAPCR (T2MOD.3)用于周期计算。设置CAPCR为1, 可以硬件上达到TH2与TL2的值已经被捕获后自动清定时器2为0000H功能, 可以省去软件循环清除标志的步骤。

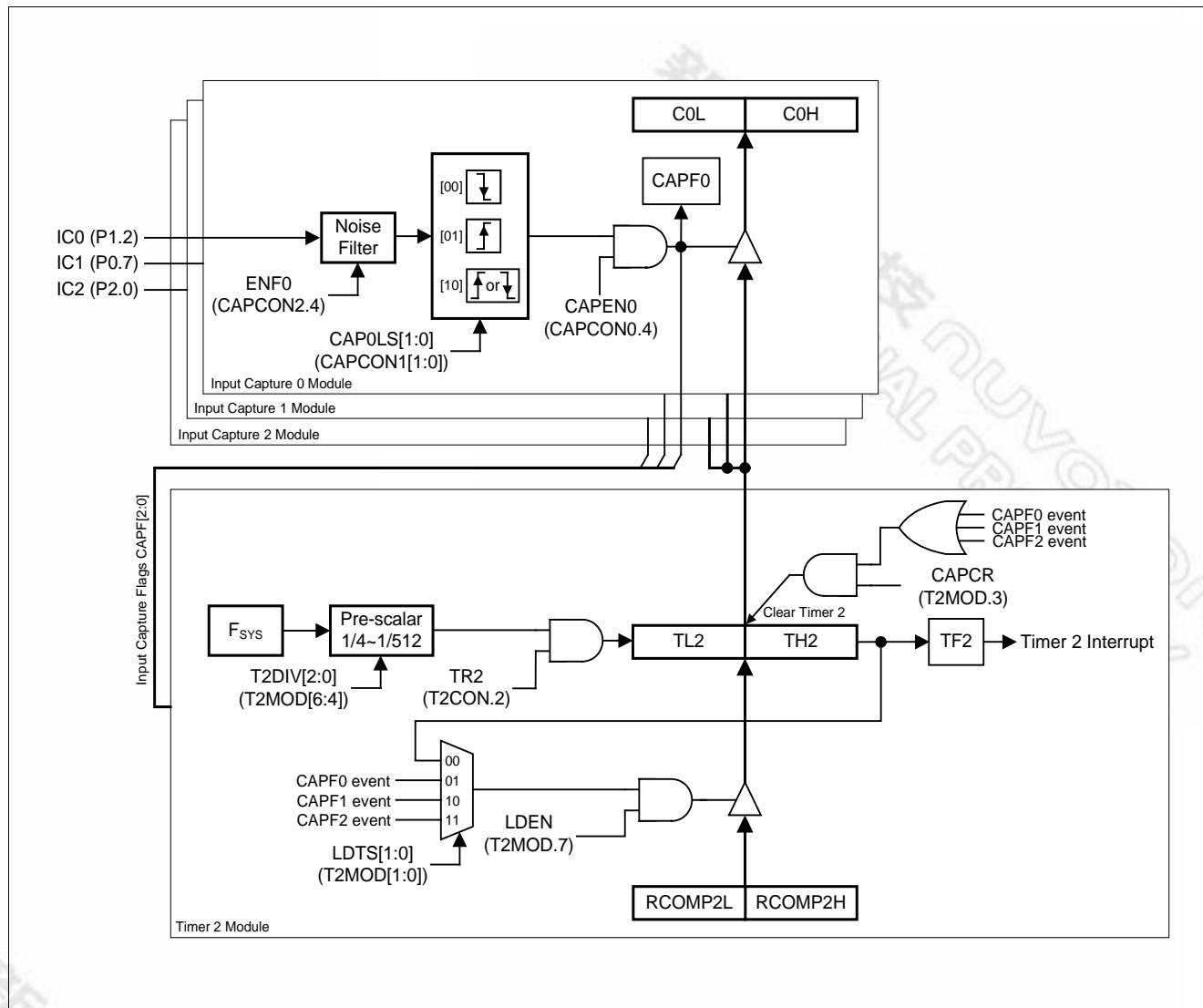


图10-5. 定时器 2 输入捕获和自动重加载模式功能模块

**CAPCON0 – 输入捕获控制位0**

7	6	5	4	3	2	1	0
-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0
-	r/w	r/w	r/w	-	r/w	r/w	r/w

地址：92H

复位值：0000 0000B

位	名称	描述
7	-	保留。
6	CAPEN2	使能输入捕获通道2使能位 0 = 关闭输入捕获通道2。 1 = 打开输入捕获通道2。

位	名称	描述
5	CAPEN1	使能输入捕获通道1使能位 0 = 关闭输入捕获通道1. 1 = 打开输入捕获通道1.
4	CAPEN0	使能输入捕获通道0使能位 0 = 关闭输入捕获通道0. 1 = 打开输入捕获通道0
3	-	保留.
2	CAPF2	输入捕获通道2标志位 如果输入捕获2边沿发生, 该位由硬件置位, 由软件清零
1	CAPF1	输入捕获通道1标志位 如果输入捕获1边沿发生, 该位由硬件置位, 由软件清零
0	CAPF0	输入捕获通道0标志位 如果输入捕获0边沿发生, 该位由硬件置位, 由软件清零

## CAPCON1 – 输入捕获控制位1

7	6	5	4	3	2	1	0
-	-	CAP2LS[1:0]		CAP1LS[1:0]		CAP0LS[1:0]	
-	-	r/w	r/w	r/w	r/w	r/w	r/w

地址: 93H

复位值: 0000 0000B

位	名称	描述
7:6	-	保留.
5:4	CAP2LS[1:0]	输入捕获通道2条件选择. 00 = 下降沿. 01 = 上升沿. 10 = 上升沿或下降沿. 11 = 保留.
3:2	CAP1LS[1:0]	输入捕获1电平条件选择 00 = 下降沿. 01 = 上升沿. 10 = 上升沿或下降沿. 11 = 保留
1:0	CAP0LS[1:0]	输入捕获通道0条件选择 00 = 下降沿. 01 = 上升沿. 10 = 上升沿或下降沿. 11 = 保留.



## CAPCON2 – 输入捕获控制位 2

7	6	5	4	3	2	1	0
-	ENF2	ENF1	ENF0	-	-	-	-
-	r/w	r/w	r/w	-	-	-	-

地址: 94H

复位值: 0000 0000B

位	名称	描述
7	-	保留.
6	ENF2	输入捕获通道2 噪声滤波使能位 0 = 关闭输入捕获通道2的噪声滤波 1 = 打开输入捕获通道2的噪声滤波
5	ENF1	输入捕获通道1 噪声滤波使能位 0 = 关闭输入捕获通道1的噪声滤波 1 = 打开输入捕获通道1的噪声滤波
4	ENF0	输入捕获通道0 噪声滤波使能位 0 = 关闭输入捕获通道0的噪声滤波 1 = 打开输入捕获通道0的噪声滤波
3:0	-	保留.

## COL – 捕获通道0 低字节

7	6	5	4	3	2	1	0
COL[7:0]							
r/w							

地址: E4H

复位值: 0000 0000B

位	名称	描述
7:0	COL[7:0]	捕获通道0 输入结果低字节 寄存器COL是16位捕获通道0输入结果的低字节值

## COH – 捕获通道0 高字节

7	6	5	4	3	2	1	0
COH[7:0]							
r/w							

地址: E5H

复位值: 0000 0000B

位	名称	描述
7:0	COH[7:0]	捕获通道0 输入结果高字节 寄存器COL是16位捕获通道0输入结果的高字节值



**C1L – 捕获通道1 低字节**

7	6	5	4	3	2	1	0
C1L[7:0]							
r/w							

地址: E6H

复位值: 0000 0000B

位	名称	描述
7:0	C1L[7:0]	捕获通道1 输入结果低字节 寄存器C1L是16位捕获通道1输入结果的低字节值

**C1H – 捕获通道1 高字节**

7	6	5	4	3	2	1	0
C1H[7:0]							
r/w							

地址: E7H

复位值: 0000 0000B

位	名称	描述
7:0	C1H[7:0]	捕获通道1 输入结果高字节 寄存器C1L是16位捕获通道1输入结果的高字节值

**C2L – 捕获通道2 低字节**

7	6	5	4	3	2	1	0
C2L[7:0]							
r/w							

地址: EDH

复位值: 0000 0000B

位	名称	描述
7:0	C2L[7:0]	捕获通道2 输入结果低字节 寄存器C2L是16位捕获通道2输入结果的低字节值

**C2H – 捕获通道2 高字节**

7	6	5	4	3	2	1	0
C2H[7:0]							
r/w							

地址: EEH

复位值: 0000 0000B

位	名称	描述
7:0	C2H[7:0]	捕获通道2 输入结果高字节 寄存器C2L是16位捕获通道2输入结果的高字节值

### 10.2.2 自动重加载模式

定时器2可以通过清  $\overline{CP/RL2}$  和设置LDEN位为1配置成自动重加载模式。此模式下RCOMP2H 和 RCOMP2L 存储生加载的值。在自动重加载事件发生时，RCOMP2H 和 RCOMP2L的内容传输到TH2 和 TL2. 该事件可以通过 LDTS[1:0] (T2MOD[1:0])来选择为定时器2的溢出或输入捕获通道的触发事件。

注：一旦 CAPCR (T2MOD.3)硬件置1，一次输入捕获完成仅会清除TH2和TL2内的值，而不会重加载 RCOMP2H 和 RCOMP2L 的内容。

### 10.2.3 比较模式

定时器2可以通过  $\overline{CP/RL2}$  置1 配置成比较模式。此模式下RCOMP2H 和 RCOMP2L 为比较值寄存器。当定时器2向上计数，TH2 和 TL2 与RCOMP2H 和 RCOMP2L值匹配，TF3 (T2CON.7) 将由硬件置位以示有比较匹配事件发生。

当COMPCR (T2MOD.2)置1，比较匹配发生后硬件会自动清定时器2计数为0000H。

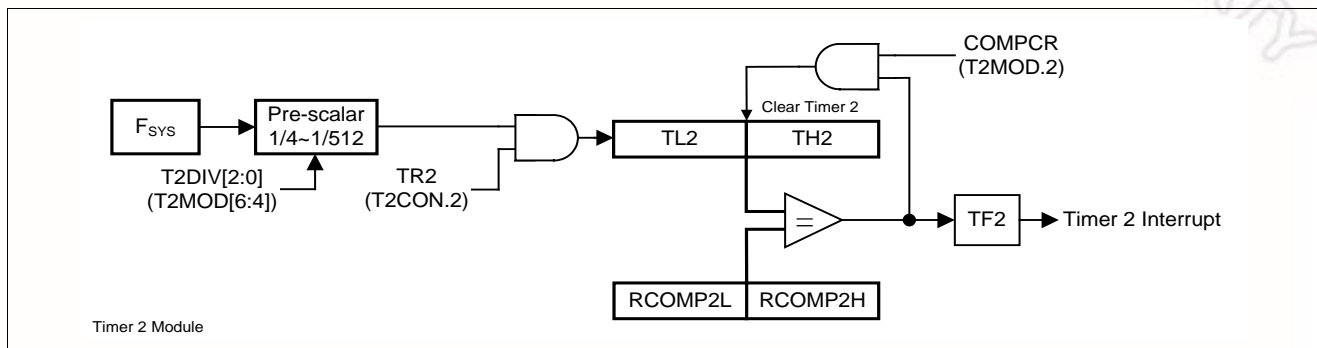


图10-6. 定时器2比较模式功能框图

## 11 看门狗定时器(WDT)

N79E815A/814A/8132A系列提供一组看门狗定时器，用于监测系统以提高系统可靠性。对于容易受到噪声，电源干扰或静电放电干扰的系统，是十分有用的工具。看门狗定时器周期中断，也可以作为事件定时或在一个周期监测制度，用于空闲模式或掉电模式下运行时长系统监控。定时器基于一组分频器，对系统时钟频率进行分割。分频器输出可选，并决定溢出时间。当定时溢出，将会把系统从空闲模式或掉电模式下唤醒，并产生中断。如果看门狗定时器复位使能，在一个延迟周期后没有软件响应，系统将复位。

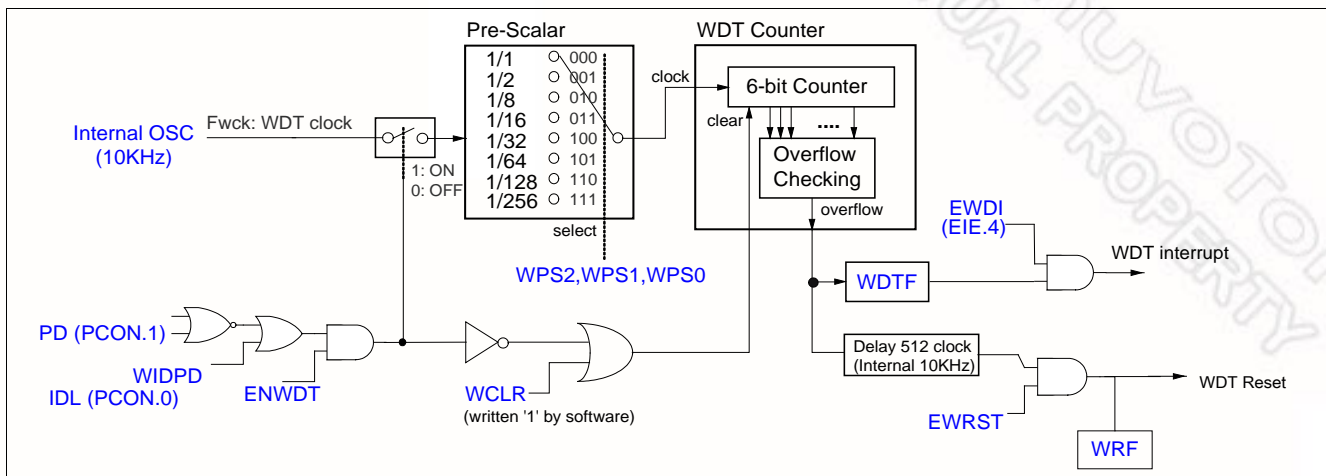


图11-1 看门狗定时器

### 11.1 功能描述

看门狗定时器 通过设定WDCLR(WDCON.6) 从00H开始执行，保证看门狗为已知的状态。通过清除WDTEN (WDCON.7)关闭看门狗后，该计数器同时清零。WDCLR 用于复位看门狗定时器，该位自动清除。WDCLR 置1看门狗定时器计数清零。WDTEN置1，看门狗定时器开始计数。通过设定WPS2、WPS1和WPS0 (WDCON[2:0]) 选择定时器溢出时间，一旦选择时间溢出，看门狗定时器中断标志WDTF (WDCON0.5)置1。看门狗中断使能在寄存器EIE.4里设置，写1到EWRST (WDCON1.0)使能看门狗定时器复位，如果同时设置WDCLR，设定的看门狗定时时间加上基于内部低速RC时钟的额外512个时钟延迟后复位发生，如果没有设置WDCLR，在512个时钟周期之内发生复位。一旦看门狗定时器复位发生，看门狗定时复位标志WDTRF (WDCON0.3)将被置1，该位只有在上电复位后才可硬件清零，用户可以通过软件清WDTRF，看门狗定时器提供WIDPD (WDCON.4)位允许看门狗定时器在系统进入空闲模式或掉电模式后继续运行。

WDT计数器需特别注意. 在进入和唤醒空闲模式或掉电模式 硬件会自动清 WDT 计数器，防止系统未知的复位。

**WDCON0 – 看门狗定时器控制 (时控保护)**

7	6	5	4	3	2	1	0
WDTEN	WDCLR	WDTF	WIDPD	WDTRF	WPS2	WPS1	WPS0
r/w	w	r/w	r/w	r/w	r/w	r/w	r/w

地址: D8H

复位值: 见表7-2.N79E815A/814A/8132ASFR功能描述和复位值

位	名称	描述
7	WDTEN	<b>WDT 使能控制位</b> 复位后该寄存器导入CWDTEN (CONFIG3, 第7位)内的值 0: 关闭WDT上电复位功能 1: 打开WDT上电复位功能
6	WDCLR	<b>WDT 计数器清零位</b> 写“1”清WDT计数器为0000H. 注: 该位只写。
5	WDTF	<b>WDT 中断标志</b> 当WDT计数器溢出时, 该位将由硬件置位
4	WIDPD	<b>看门狗在空闲模式和掉电模式下运行状态</b> 该位决定看门狗在空闲模式和掉电模式下的运行状态 0 = 当CPU在空闲模式或掉电模式下, WDT 计数器停止 1 = 当CPU在空闲模式或掉电模式下, WDT 计数器保持运行
3	WDTRF	<b>WDT 复位标志</b> 当MCU自身复位时, 该位由硬件置位. 该位应该由软件清零 如果EWRST=0, 中断标志WDTF不会被硬件置位, 同时MCU立即复位. 如果EWRST=1, 中断标志WDTF将由硬件置位, 如果WDT中断使能, MCU跳到WDT 中断服务程序, 执行中断服务程序后持续512个CPU始终, 系统才复位。也就是说, 用户必须在这512个时钟周期内软件清除WDT计数器 (对WDCLR写1), 或者512个时钟周期后系统复位。
2:0	WPS[2:0]	<b>WDT 预分频选择</b> 用于选择WDT定时溢出周期。  WDT 定时溢出周期公式 = $\frac{64}{(F_{wck} \times PreScalar)}$ ,  F <sub>wck</sub> 为其中 F <sub>wck</sub> 是内部10kHz RC频率。

[1] 任一复位后WDTEN 内的值导入CWDTEN(CONFIG3.7) 内的值。

[2] 上电复位WIDPD 和WPS[2:0] 内的数值清零, 其他复位不改变这两个寄存器内的值。

[3] 上电复位, 看门狗复位后WDTRF内的值清零, 其他复位不改变这个寄存器内的值。

**WDCON1 –看门狗定时器控制 (时控保护)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	EWRST
-	-	-	-	-	-	-	r/w

地址: ABH

复位值: 0000 0000B

位	名称	描述
0	EWRST	<b>WDT复位功能</b> 0: 关闭WDT 复位功能 1: 打开WDT复位功能

[1] 上电复位后EWRST内容清零, 其他复位不改变该寄存器值。

看门狗定时溢出间隔公式 =  $\frac{64}{(F_{wck} \times PreScalar)}$  .其中 F<sub>wck</sub> 是内部10kHz RC频率。

## EIE – 扩展中断使能

7	6	5	4	3	2	1	0
ET2	ESPI	EPWM	EWDI	-	ECPTF	EKB	EI2
r/w	r/w	r/w	r/w	-	r/w	r/w	r/w

地址: E8H

复位值: 0000 0000B

位	名称	描述
4	EWDI	看门狗中断使能位 0: 禁止看门狗定时器中断. 1: 使能看门狗定时器中断

看门狗定时溢出选择根据时钟速率会有不同定时溢出值. 当使能复位, 将会在定时溢出发生的512个时钟之后发生复位。

表11-1: 看门狗定时器定时溢出值

(WPS2,WPS1,WPS0)	预分频	WDT 中断发生时间		复位发生时间	
		时钟计数	时长	时钟计数	时长
(0,0,0)	1/1	$2^6$	6.4ms	$2^6+512$	57.6ms
(0,0,1)	1/2	$2 \times 2^6$	12.8ms	$2 \times 2^6+512$	64ms
(0,1,0)	1/8	$8 \times 2^6$	51.2ms	$8 \times 2^6+512$	102.4ms
(0,1,1)	1/16	$16 \times 2^6$	102.40ms	$16 \times 2^6+512$	153.6ms
(1,0,0)	1/32	$32 \times 2^6$	204.80ms	$32 \times 2^6+512$	256ms
(1,0,1)	1/64	$64 \times 2^6$	409.60ms	$64 \times 2^6+512$	460.8ms
(1,1,0)	1/128	$128 \times 2^6$	819.20ms	$128 \times 2^6+512$	870.4ms
(1,1,1)	1/256	$256 \times 2^6$	1.638s	$256 \times 2^6+512$	1.6892s

## 11.2 看门狗定时器复位的应用

看门狗定时器复位的主要应用是系统监测, 这对于实时控制很重要, 适用于电磁干扰等避免发生程序跑飞等场合, 或在未知状态发生时保护用户的代码. 使用看门狗定时器 用户可选择理想的看门狗复位看门狗定时时间. 设定 WCLR, 可使代码继续运行而无看门狗定时器复位. 若未开启看门狗定时器复位, 一旦代码运行在错误的状态下, 无法清除看门狗定时器, 将引起芯片复位. WDCON 有时序写入保护功能, 避免误改写。



### 11.3 看门狗定时器中断应用

看门狗定时器的另一个应用是用作简单的定时器。看门狗定时器完成所选的时间间隔后,WDTF标志置位. 软件让WDTF标志检测定时溢出, WDCLR允许软件重启定时器. 看门狗定时器可被用作较长的定时器. 任何定时溢出发生时, 在EWDT (EIE.4)和EA置1打开的条件下将发生中断。

在一些功耗的应用中, CPU常在没有处理事件时处于空闲模式或掉电模式, 需要定时唤醒察看是否需要响应, 而定时器0、1和2打开后耗电量将达到mA级, 无法达到系统需要保持在 $\mu$ A级的要求。N79E815A/814A/8132A 看门狗定时器可以起到定时唤醒的功能, 由于基于内部10kHz的RC时钟源, 看门狗定时器功耗非常低, 更是用于需要低功耗的场合。以下为应用示例

看门狗定时器在掉电模式下应用的例程

```

ORG 0000H
LJMP  START

ORG 0053H
LJMP  WDT_ISR

ORG 0100H
WDT_ISR:
CLR  EA
MOV  TA,#0AAH
MOV  TA,#55H
ORL  WDCON0,#01000000B      ; clear Watchdog Timer counter
INC  ACC
MOV  P0,ACC
SETB  EA

CLR  EA
MOV  TA,#0AAH
MOV  TA,#55H
ANL  WDCON0,#11011111B     ; clear Watchdog Timer interrupt flag
SETB  EA
RETI

START:
MOV  TA,#0AAH
MOV  TA,#55H
ORL  WDCON0,#01000000B     ; clear Watchdog Timer counter

MOV  TA,#0AAH
MOV  TA,#55H
ORL  WDCON0,#10000000B     ; enable Watchdog Timer to run

Check_clear:
MOV  A,WDCON0
JB   ACC.6,Check_clear

MOV  TA,#0AAH
MOV  TA,#55H
ORL  WDCON0,#00000111B     ; choose interval length

```



```
MOV TA,#0AAH
MOV TA,#55H
ANL WDCON1,#11111110B      ; disable Watchdog Timer reset
SETB EWDI                  ; enable Watchdog Timer interrupt
MOV TA,#0AAH
MOV TA,#55H
SETB WIDPD

SETB EA

; *****
; Enter into Power Down mode
; *****
LOOP:
    ORL PCON,#02H
    LJMP LOOP

END
```

## 12 串口 (UART)

N79E815A/814A/8132A系列提供一组，带自动地址识别和帧错误检测功能的，全双工串口 UART (全双工传输和接收)，可运行模式 1、2 和 3。该串行口提供同步及异步通信方式。在同步模式下串行口产生时钟并以半双工的方式工作。在异步模式下，能以全双工的方式工作，即可以同时收发数据。发送，接收寄存器均用SBUF来访问。对SBUF的写是发送数据，从SBUF读是读取数据。串行口能以4种不同的方式工作，使用串口功能前，P3.0和P3.1 (RXT 和 TXD) 需置 1。串口位置和软件更改UART\_Sel (AUXR1.6) 提供软件切换串口位置功能。

### SCON – 串口控制 (可位寻址)

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

地址：98H

复位值：0000 0000B

位	名称	描述
7	SM0/FE	<b>串口模式选择位</b>
6	SM1	<p><b>SMOD0 (PCON.6) = 0:</b> 详见表12-1. 串口模式描述 串口模式描述.</p> <p><b>SMOD0 (PCON.6) = 1:</b> SM0/FE 位用作帧错误 (FE) 状态标志. 0 = 没有帧错误 (FE) 1 = 检测到帧错误 (FE)</p>
5	SM2	<p><b>多处理通讯模式选择</b> 该位功能取决于串口模式</p> <p><b>模式0:</b> 该位选择波特率<math>F_{SYS}/12</math> 或 <math>F_{SYS}/4</math>. 0 = 时钟运行在<math>F_{SYS}/12</math> 波特率,与标准8051兼容. 1 = 时钟运行在<math>F_{SYS}/4</math> 波特率, 获得更高传输速度</p> <p><b>模式1:</b> 该位检查有效停止位. 0 = 接收有效不管停止位是否有逻辑电平. 1 = 接收仅在接收停止位为逻辑1,同时接收数据与GIVEN或BROADCAST地址匹配时有效</p> <p><b>模式2 或 3:</b> 对于多机通信. 0 =接收有效不管第9th位是否有逻辑电平. 1 =接收仅在第9th位为逻辑1和接收数据与GIVEN或BROADCAST地址匹配时有效.</p>
4	REN	<p><b>接收使能</b> 0 = 关闭串口接收功能. 1 = 打开串口在模式1, 2或3模式下的接收功能。接收完成后, 该位不会被硬件清除。所以用户必须在每字节接收完毕后, 软件清除该位, 并等待接收下一字节.</p>
3	TB8	<p><b>第9位传输位</b> 模式2和3中要被发送的第九位数据。在模式0和1中, 不支持该功能.</p>



位	名称	描述
2	RB8	<b>第9位接收位</b> 模式2和3中接收到的第九位数据。模式1下，若SM2=0则RB8是接收到的停止位。模式0下该位无意义。
1	TI	<b>发送中断标志位</b> 发送中断标志：模式0下该标志由硬件在发送完8位数据后置1，而在其它模式下在串行发送到停止位的开始时置位。当该位中断使能，发生中断后会转至中断子程。该位必须由软件来清除。
0	RI	<b>接收中断标志</b> 模式0下该标志由硬件置位。在模式0中，接收到第8位或第9位；模式1中接收到停止位(stop bit)；模式2和3中接收到第9位，使该位置位。当SM2被限制的情况例外。当UART中断使能，该位置位会转跳到中断子程运行。该位必须由软件来清除。

表12-1. 串口模式描述

Mode	SM0	SM1	描述	帧位数	波特率
0	0	0	同步	8	$F_{SYS}$ 除以12 或 除以4 <sup>[1]</sup>
1	0	1	异步	10	定时器1溢出时间 除以32 或 除以16 <sup>[2]</sup>
2	1	0	异步	11	$F_{SYS}$ 除以64 或 除以32 <sup>[2]</sup>
3	1	1	异步	11	定时器1溢出时间 除以32 或 除以16 <sup>[2]</sup>

[1] 当 SM2 (SCON.5) 的值为1.

[2] 当 SMOD (PCON.7) 的值为1.

#### PCON – 电源控制

7	6	5	4	3	2	1	0
SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL
r/w	r/w	-	r/w	r/w	r/w	r/w	r/w

地址： 87H

复位值： 见 表7-2.N79E815A/814A/8132ASFR功能描述和复位值

位	名称	描述
7	SMOD	<b>串口波特率加倍使能.</b> 在模式 1, 2, 或 3中串口波特率加倍使能. 仅适用于定时器1溢出作为为波特率时钟源时. 详见表12-1. 串口模式描述.
6	SMOD0	<b>帧错误检测使能.</b> 0 = 关闭帧错误检测功能. 和标准8051相同SM0/FE (SCON.7)作为SM0用 1 = 使能帧错误检测. SM0/FE位用于帧错误 (FE) 状态标志

**SBUF – 串行数据缓冲**

7	6	5	4	3	2	1	0
SBUF[7:0]							
r/w							

地址： 99H

复位值： 0000 0000B

位	名称	描述
7:0	SBUF[7:0]	<b>串口数据缓冲</b> 串行口接收或发送的数据都放在这个寄存器中。实际上该地址上有2个独立的8位寄存器。一个用于接收数据，一个用于发送数据。对它进行读操作将会接收串行数据，对它进行写操作则发送串行数据。 每次向SBUF写入一字节数据，启动一次发送。

**AUXR1 – 附加功能寄存器1**

7	6	5	4	3	2	1	0
SPL_Sel	UART_Sel	-	-	DisP26	-	0	DPS
r/w	r/w	-	-	r/w	-	r	r/w

地址： A2H

复位值： 0000 0000B

位	名称	描述
6	UART_Sel	<b>串口管脚位置选择</b> 0: 选择 P1.0, P1.1 作为串口管脚 1: 选择 P2.6, P2.7 作为串口管脚

**12.1 模式0**

模式 0 与外部设备进行同步通信的方式。在该模式下，串行数据由RXD脚进行收发，而TXD 脚用于产生移位时钟。这种方式下是以半双工的形式进行通信，每帧接收或发送8位数据。数据的最低位被最先发送或接收，波特率  $F_{SYS}/12(SM2 (SCON.5) 为 0)$  或  $F_{SYS}/4 (SM2 为 1)$ 。无论传输或接受 串行时钟将一直产生.因此串口模式 0 为主机模式. [图12-1](#) 模式0功能框图

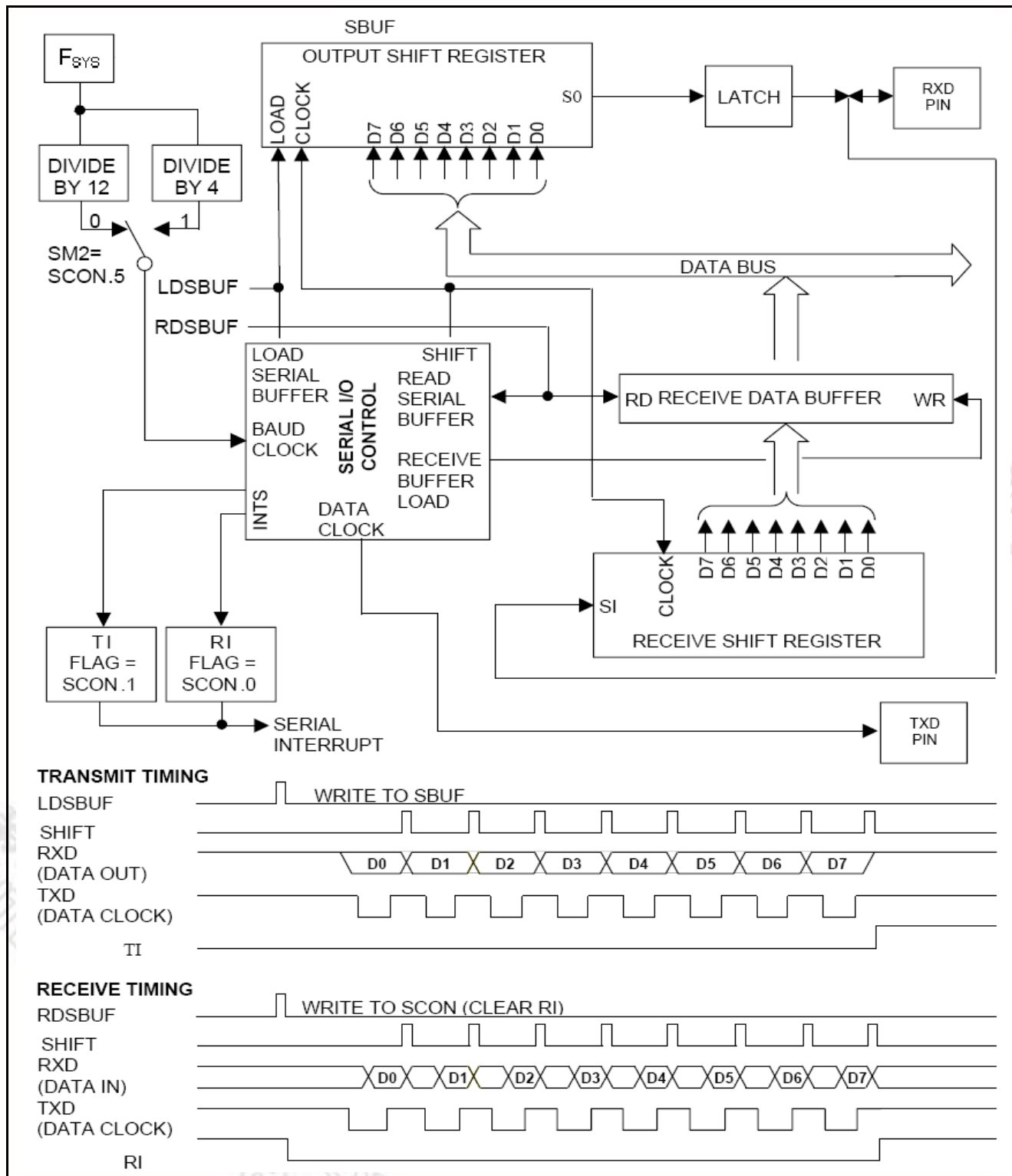


图12-1. 串口模式功能模块及时序图

如图所示，数据由双向RXD线进行收发。移位时钟TXD线用来输出移位时钟，移位时钟用来和其它设备串行接收/发送数据。数据最低位用于移入移出数据，波特率根据TXD时钟同步。

向SBUF的写入数据将会开启发送，此时移位时钟启动数据从RXD脚串行移出，直至送完8位数据传输。传输标志位TI (SCON.1) 置 1表示 1 字节传输完成。

当REN (SCON.4)=1 且 RI(SCON.0)=0 时串行口接收数据。移位时钟被激活，串行口会在移位时钟的上升沿锁定数据。外部设备要在移位时钟的下降沿处送出数据。这个过程持续到8位数据全部发送完毕。RI会在TXD的最后一个下降沿处置1，这时接收动作结束，注REN不由硬件清零，用户应该首先清零RI，清REN，并再次通过软件置位REN，以触发下一字节的接收。

## 12.2 模式1

模式1为全双工的方式工作。串行通信的数据帧由10位数据组成，在RXD和TXD脚上进行收发。10位数据组成如下：起始位（位0），8位数据（最低位在前），终止位（1）。波特率由定时器1决定，SMOD (PCON.7) 设置为1使波特率加倍(定时器1为波特率发生源).下图为模式1的功能框图.

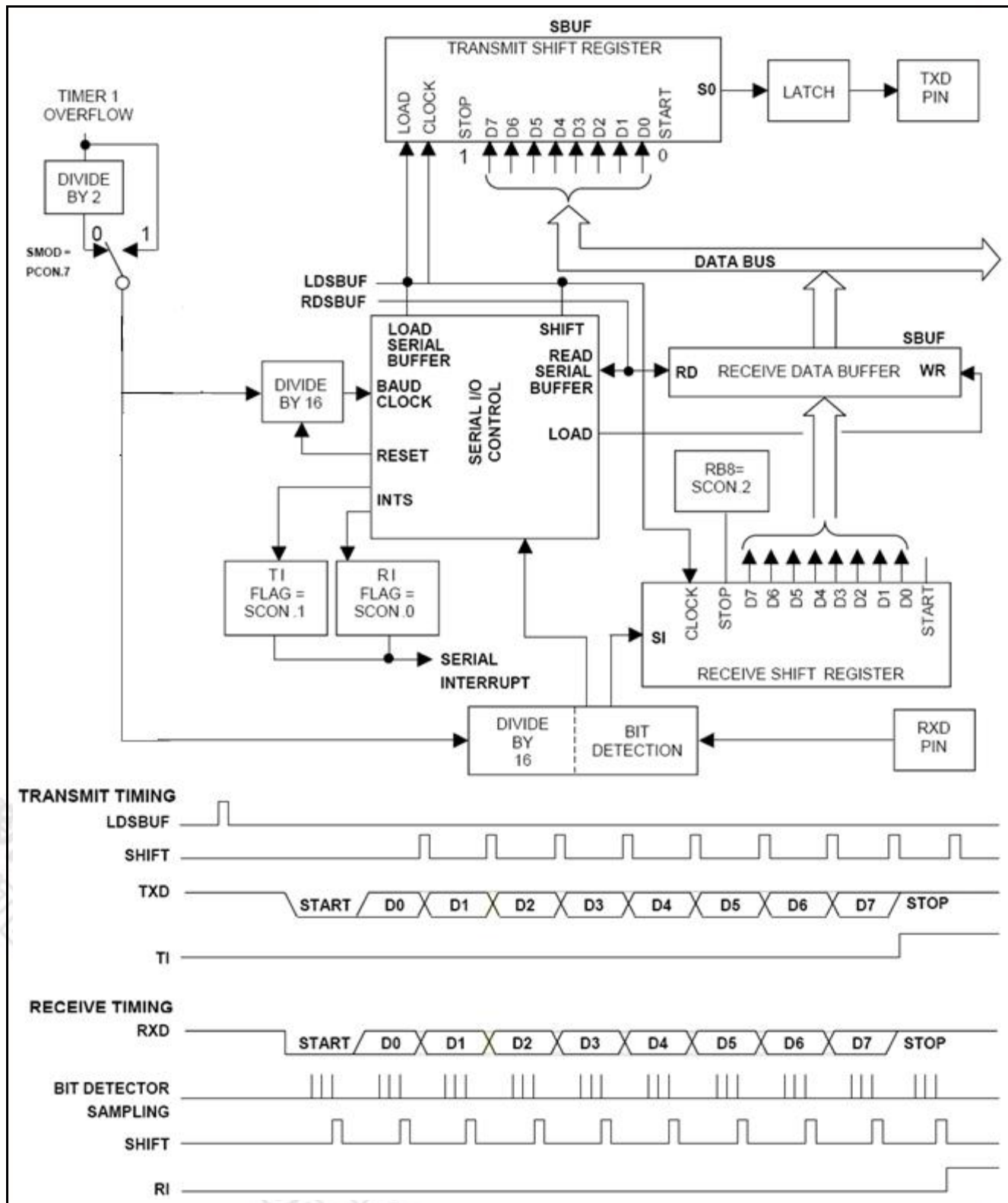


图12-2. 串口模式1功能模块及时序图

向SBUF写入指令开始传输，传输发生在TXD引脚上。首先是起始位，随后是8位数据位，最后是停止位，停止位出现后，TI (SCON.1) 置1 表示一个字节传输完成，所有位的传输速度取决于波特率。

当波特率发生器打开且REN(SCON.4) =1 时系统进行接收操作，RXD脚上接收到1-0跳变就启动接收器接收。数据根据波特率的时钟频率接收，停止位必须符合一定的条件，才能从SBUF读到的数据：

1. RI (SCON.0) = 0
2. 任何SM2 (SCON.5) = 0, 或当SM2 = 1.时接收停止位= 1

如果上述条件满足，SBUF将加载接收数据，RB8 (SCON.2) 停止位，和RI将被置1， 如果条件不满足，RI保持为0，没有数据加载。完成接收过程后，串口控制等待RXD脚上的另一个1-0传输以开始新的数据接收。

### 12.3 模式2

模式2支持全双工异步通信，数据由起始位位（逻辑0），8位数据（最低位在前），可编成的第9位数据（TB8）和停止位组成。第9位数据接收至RB8。波特率是时钟频率的1/32 或1/64，由 SMOD位来选择。[图12-3](#) 模式2的功能框图。

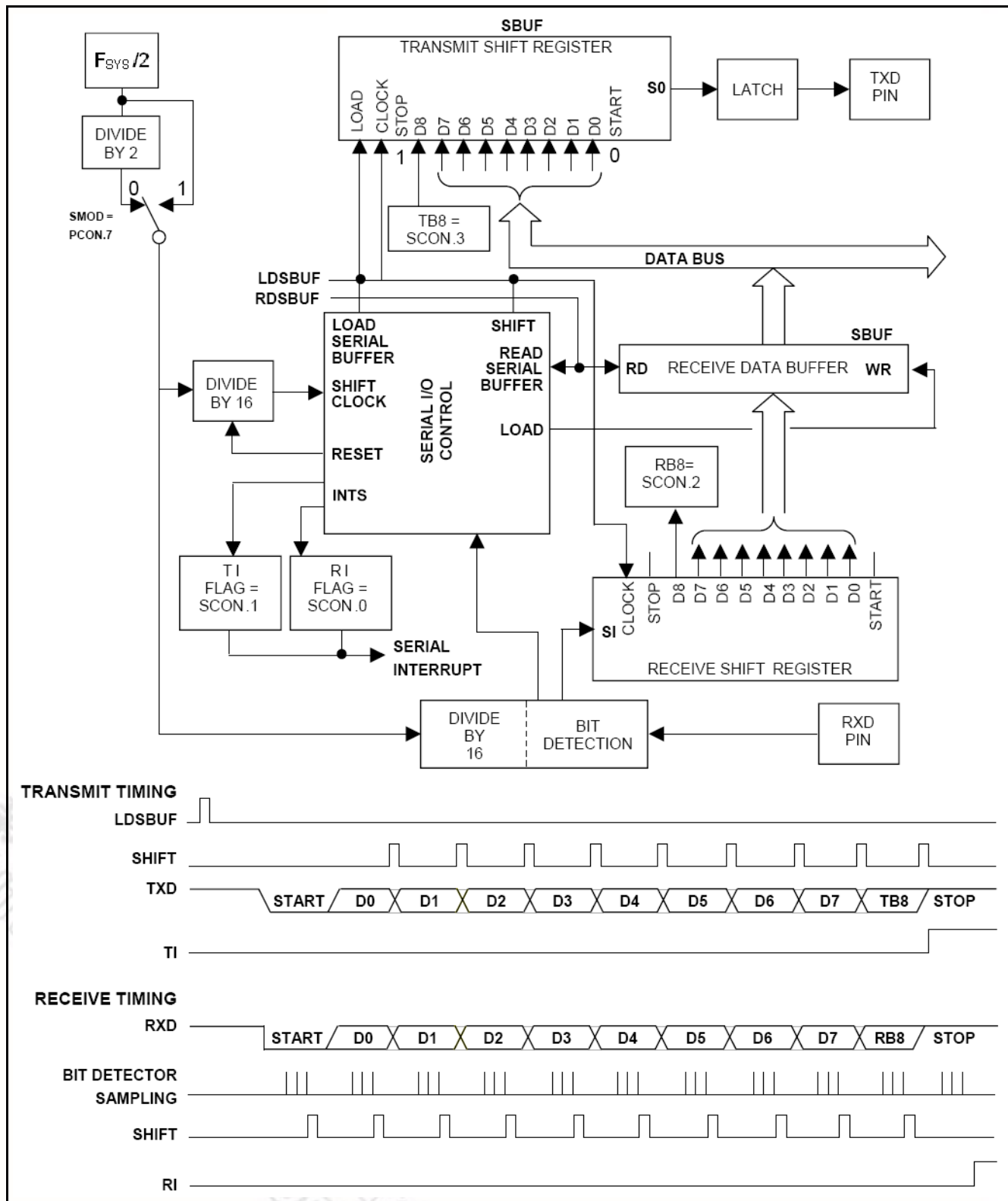


图12-3. 串口模式2结构及时序图

向SBUF中写入数据启动一次发送，首先是开始位，8位数据和TB8（SCON.3），最后是停止位，停止位出现时，TI将置位以示传输完成

当REN=1 时系统进行接收操作，RXD上的下降沿表示接收过程开始，数据根据所配置波特率进行采样，并在选定的波特率转移。在第9位的情况下，必须符合一定的条件，装载SBUF才接收到的数据：

1. RI (SCON.0) = 0,
2. 任何 SM2(SCON.5) = 0, 或接受 9<sup>th</sup> 位 = 1 当 SM2 = 1.

如果上述条件满足，则停止位进入RB8(SCON.2)，8位数据进入SBUF，RI置位，否则丢弃接收到的贞数据。在停止位的中间，接收器重启，开始新的一次接收。

### 12.4 模式 3

除波特率外 模式 3 与模式 2相同。图12-4, 模式3采用定时器1的溢出作为波特率时钟。



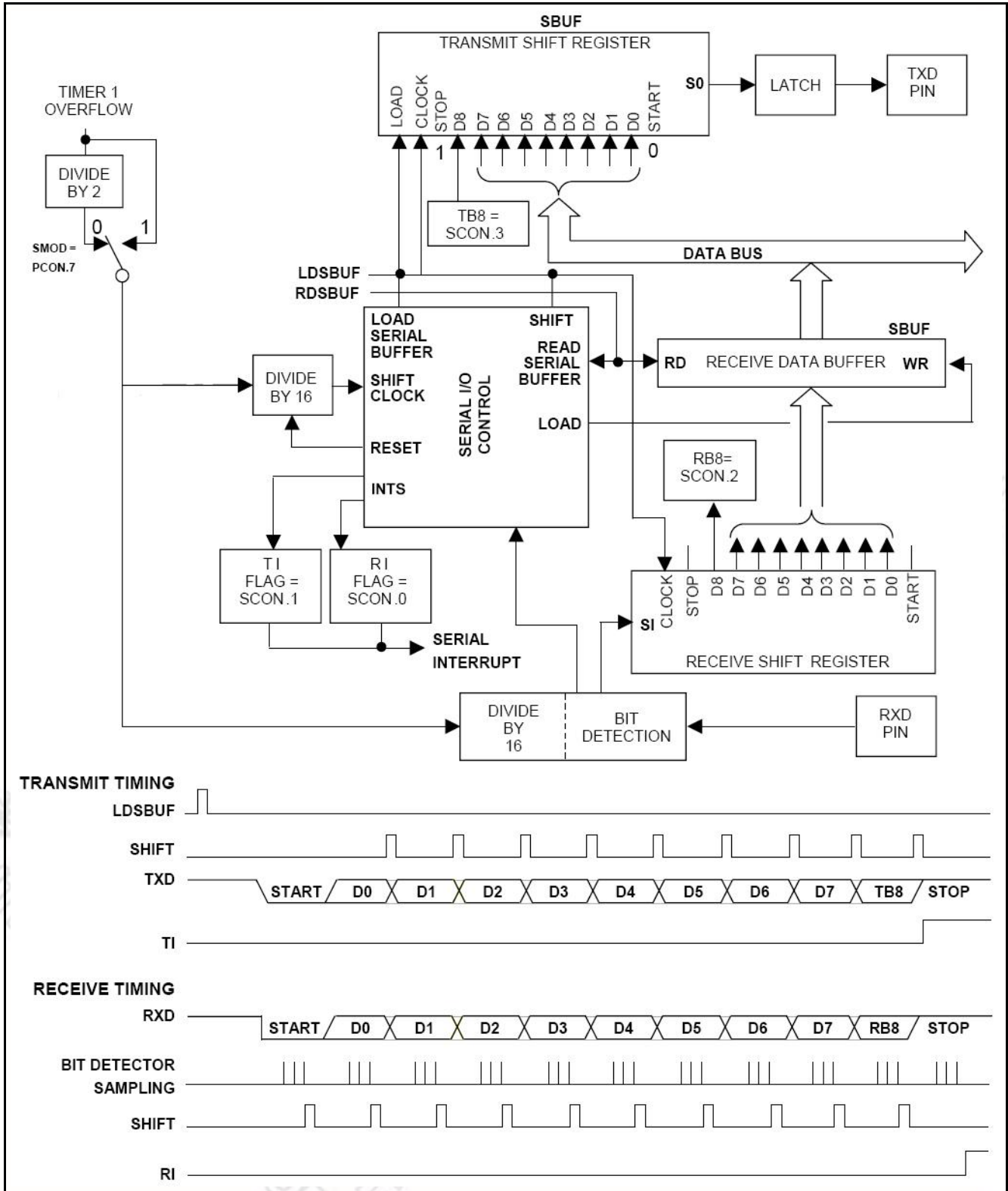


图12-4. 串口模式3结构及时序图

## 12.5 波特率

表12-2. 串口波特率计算方程

串口模式	波特率时钟源	波特率
0	芯片时钟晶振	$F_{SYS} / 12$ or $F_{SYS} / 4$ <sup>[1]</sup>
2	芯片时钟晶振	$\frac{2^{SMOD}}{64} \times F_{SYS}$
1 or 3	定时器1定时溢出 <sup>[2]</sup>	$\frac{2^{SMOD}}{32} \times \frac{F_{SYS}}{12 \times (256 - TH1)}$ or $\frac{2^{SMOD}}{32} \times \frac{F_{SYS}}{4 \times (256 - TH1)}$ <sup>[3]</sup>

[1] 当SM2 (SCON.5) 设定为1时。

[2] 定时器1定义在自动重载模式（定时器模式2）。

[3] 当 T1M (CKCON.4) 设定为1 时。

注：使用定时器 1 作为波特率发生器，定时器中断必须关闭。定时器可配置为计数器或定时器模式。定时器 1 可运行于3 种模式中任意中模式，但通常配置为定时器自动重载模式，在此模式下，如果定时器 1 作为波特率发生器。特率的值由 TH1的值决定波特率的值。

表12-3为使用定时器 1 做为波特率发生器。在此模式下，定时器 1 为自动装载 12T 模式 SMOD (PCON.7) 为 0，如果SMOD为1，波特率加倍。

表12-3. 定时器1用作波特率发生器时的频率

TH1 自动重载模式	晶振频率 (MHz)			
	11.0592	14.7456	18.432	22.1184
波特率				
57600				FFh
38400		FFh		
19200		FEh		FDh
9600	FDh	FCh	FBh	FAh
4800	FAh	F8h	F6h	F4h
2400	F4h	F0h	ECh	E8h
1200	E8h	E0h	D8h	D0h
300	A0h	80h	60h	40h

## 12.6 帧错误检测

帧错误检测用于异步模式 (Mode 1, 2 和 3.)。当有总线噪声或连接不上导致没有检测到停止位时, 发生帧错误.串口可以检测帧错误, 并通过软件标志提示出错。

SCON.7是FE标志 (帧错误标志) (FE\_1) 。在标准8051种该位是SM0 , 但当SMOD0 (PCON.6)置1, 帧错误检测功能打开, 她作为FE标志, 但在 N79E815系列中它有附加功能称为SM0/FE。他们其实是相互独立的标志位

FE标志由硬件置位且必须由软件清0。注意在对FE标志位进行读写时, SMOD0必须为1。如果FE置位, 那么下次接收到的正确数据帧不会将其清除。对该位的清除必须由软件来完成。

## 12.7 多机通讯

N79E815A/814A/8132A 串口多机通讯, 可让一个主机 (master device) 向多个从机 (slave device) 发送多帧序列信息, 在同一串行线上使用该功能过程中不需要中断其它从机设备。该功能只能在模式2或模式3下进行。当第9位数据接收到后, 第9位数据内容存入RB8 (SCON.2)。当接收到停止位stop bit后, 用户可通过设定SM2 (SCON.5) 为1使能该功能。只有RB8为1时, 才能产生中断。当SM2位 为1, 接收到的第9位数据为0, 不会引发中断。在上例中, 第9位能简单得从地址中分离。

当主机需要向多个从机中的一个发送数据时, 首先需要发送目标从机的地址。注, 地址字节与数据字节是不同的: 在地址字节中, 第9位为1。而数据字节中第9位为0。地址字节会触发所有从机, 而每台从机检查接收到的地址是否与自身匹配。未被寻址到的从机的SM2 必须保持, 从而系统会持续工作。.

配置多机通信步骤如下:

1. 设置所有设备(主机与从机) 为UART 模式 2 或 3.

2. 所有从机 SM2 位置为1

3. 主机传输协议:

- 开始位: 地址, 定义目标从机 (第9位 = 1)
- 下 1 字节: 数据, (第9位 = 0).

4. 当目标从机接受到开始位, 所以从机将中断 因为 9<sup>th</sup> 数据为 1。目标从机比较自身地址 并且清 SM2 位 接受下列数据, 其它从机继续运行。

5. 接收到所有数据后，置 SM2 为 1 等待下一地址。

SM2 在模式 0 下无效。若 SM2 置 1，模式 1 可用于检测有效的停止位，同时将无法产生中断除非有效停止位已经接收。

## 12.8 自动地址识别

自动地址识别是这样一种特性，它允许串口在数据流传输过程中，通过硬件识别比较，确认特定数据为地址。该功能可以节省软件识别地址而所占用的程序空间，仅当串口识别到自身地址时，接收器置位 RI 位并请求中断。当多机通信特征使能时（SM2 置位），就使能自动地址识别。

如果需要，用户可以在模式 1 下使能自动地址识别特征。在这种配置下，停止位取代第九位的数据位。仅当接收命令的帧地址与器件地址匹配和有效的停止位中止时，RI 置位。

使用自动地址识别，允许一个主机选择与一个或多个从机通信，通过“Given”从机地址。所有从机可以通过“广播”地址联系。有两个特殊功能寄存用于定义从机地址 SADDR 和从机地址掩码 SADEN。SADEN 用于定义 SADDR 的哪位被用，哪位不必关心。SADEN 掩码可以与 SADDR 以逻辑与得方式以创建每个从机的“Given”地址。使用“Given”地址允许多从机被识别。

### SADDR – 从机地址

7	6	5	4	3	2	1	0
SADDR[7:0]							
r/w							

地址：A9H

复位值：0000 0000B

位	名称	描述
7:0	SADDR[7:0]	从机地址。 该字节定义微控制器自身的从机地址以用于串口多机通信。

### SADEN – 从机地址掩码

7	6	5	4	3	2	1	0
SADEN[7:0]							
r/w							

地址：B9H

复位值：0000 0000B

位	名称	描述
7:0	SADEN[7:0]	从机地址掩码。 该字节为掩码存储“Given”地址的“无关”位。无关为科使更多从机得以灵活运用。



下列范例用以说明该功能的灵活应用。

范例1, 从机0:

```
SADDR = 11000000b
SADEN = 11111101b
Given = 110000X0b
```

范例2, 从机1:

```
SADDR = 11000000b
SADEN = 11111110b
Given = 110000Xb
```

在上面的例子中SADDR是相同的，SADEN的数据用于区分两个从机。从机0要求位0为“0”而忽略位1，从机1要求位1为“0”而位0被忽略。一个从机0唯一的地址11000010B，由于从机1要求位1为0。一个从机1唯一的地址将自1位011000001b将排除从机0。这两个从机可以选择在同一时间，地址位0 = 0（从机0）和第1位= 0（从机1）。因此，就可以解决同时地址位为11000000b的问题。

在更复杂的系统，可用于以下选择1和2的从机，而排除从机0:

范例1, 从机0:

```
SADDR = 11000000b
SADEN = 11111001b
Given = 11000X0b
```

范例2, 从机1:

```
SADDR = 11100000b
SADEN = 11111010b
Given = 11100X0b
```

范例3, 从机2:

```
SADDR = 11000000b
SADEN = 11111100b
Given = 110000XXb
```

在上面的例子中，3个从机的分别是在地址的低3位。从机0要求位0 = 0，它可用11100110b解决。从机1要求位1 = 0，它可用11100101b识别。从机2要求位2 = 0，其独立的地址是11100011b。要选择从机0和1，去除从机2，可使用地址11100100b，因为它是必要的第2位 = 1来排除从机2。为每个从站的地址是“广播”创建逻辑或SADDR和SADEN。使用“无关”地址功能时结果为零。在大多数情况下，解释“无关”的广播地址将是FFh。

复位后，SADDR和SADEN初始化为00H。这将对于所有所有“无关”地址产生一个“Given”地址，以及一个“广播”地址对应所有XXXXXXXXb地址（所有“无关”位）。这样有效地禁止了自动寻址模式，允许微控制器保持标准串口模式而不使用这个功能。

## 13 串行外围设备接口(SPI)

### 13.1 特征

N79E815A/814A/8132A系列 有支持高速串行通信的SPI模块。SPI 为全双工、高速、同步传输总线在微芯片外设 EEPROM, LCD 驱动, D/A 转换之间, 提供主机从机模式传输, 速度可达主机模式下 $F_{SYS}/16$ 和从机模式下 $F_{SYS}/4$ , 支持传输完成标志位 写 冲突标志位。在多主机系统中, SPI 支持主机模式故障报错用以防止主机冲突。

### 13.2 功能描述

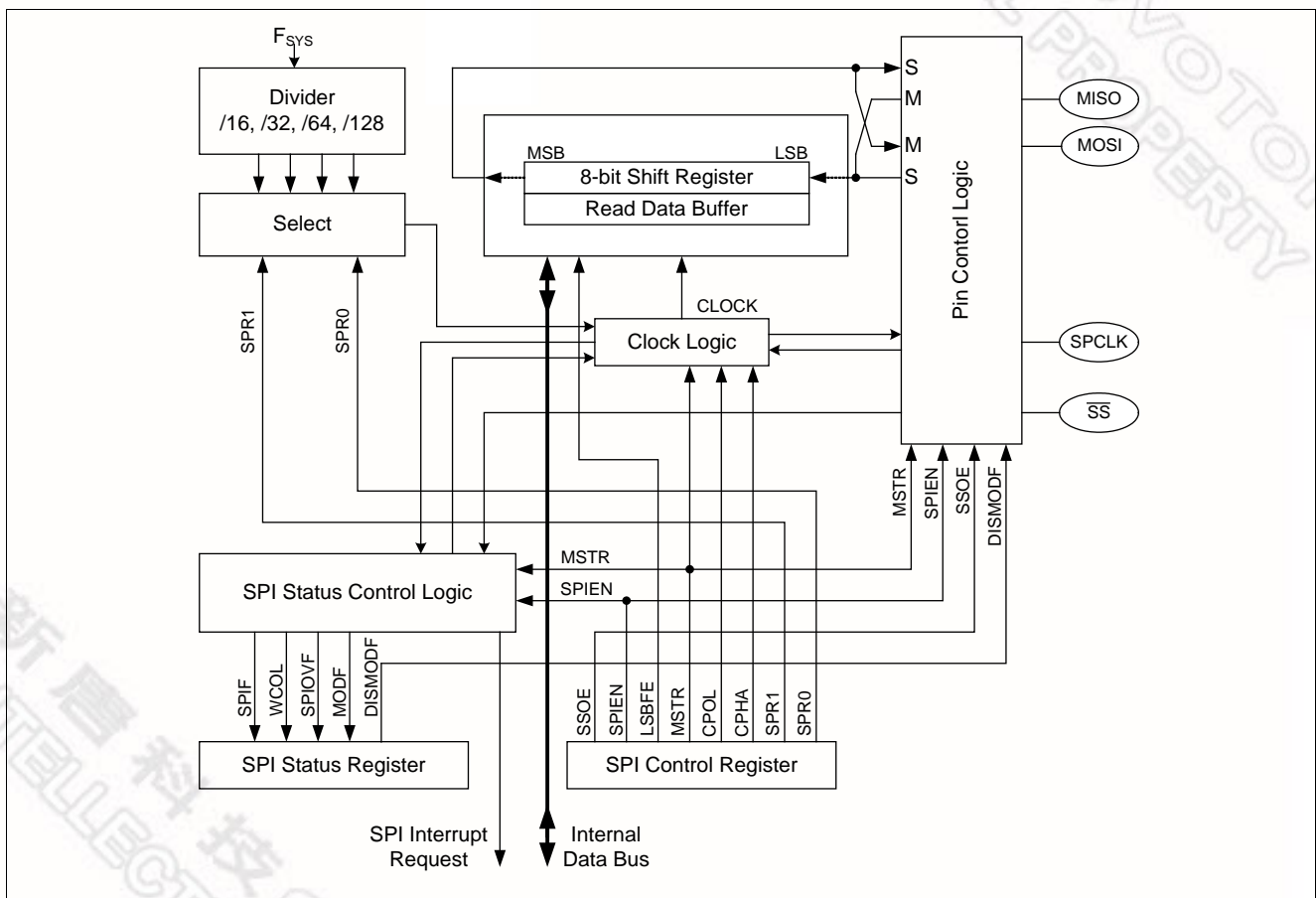


图13-1. SPI 模块结构图

图 13-1 为SPI框图, 展示了SPI的体系结构。SPI寄存器板块是SPI模块的主要组成部分, 包括逻辑控制, 波特率控制和管脚逻辑控制, SPI 包括移位寄存器和读出数据缓冲器, 传送数据是单缓冲器, 接收数据是双缓冲器。在传送完成之前传送的数据不能写入移位装置。

SPI 界面需要四个管脚 主进/从出(MISO)，主出/从进(MOSI)，移位时钟(SPCLK)，和从机选择( $\overline{SS}$ )。MOSI脚用于传输主机到从机的8位数据，因此，MOSI是一个主机设备的输出引脚，从机设备的输入引脚。相应的，MISO用于接收从机到主机的串行数据。

SPCLK引脚为主机模式下的时钟输出，从机模式的时钟输入。移位时钟用于MOSI和MISO脚之间数据传输的时钟同步。移位时钟由主机输出，一组SPI传输系统上只能有一个主机以避免设备冲突。建议该管脚设置为史密特触发输入模式。

每路从机外设通过设定从机选择脚 ( $\overline{SS}$ )使能。当需要读取任何从机时，该信号脚必须保持低。当  $\overline{SS}$  为高，从机读写将被禁止。若为多从机模式，在同一时刻必须只有一个从机保持此低电位，对于主机  $\overline{SS}$  脚不做任何用途，可配置为普通端口另做他用。 $\overline{SS}$  可用于多主机模式下错误侦测功能。N79E815A/814A/8132A也提供通过  $\overline{SS}$  所存字节传输数据功能。

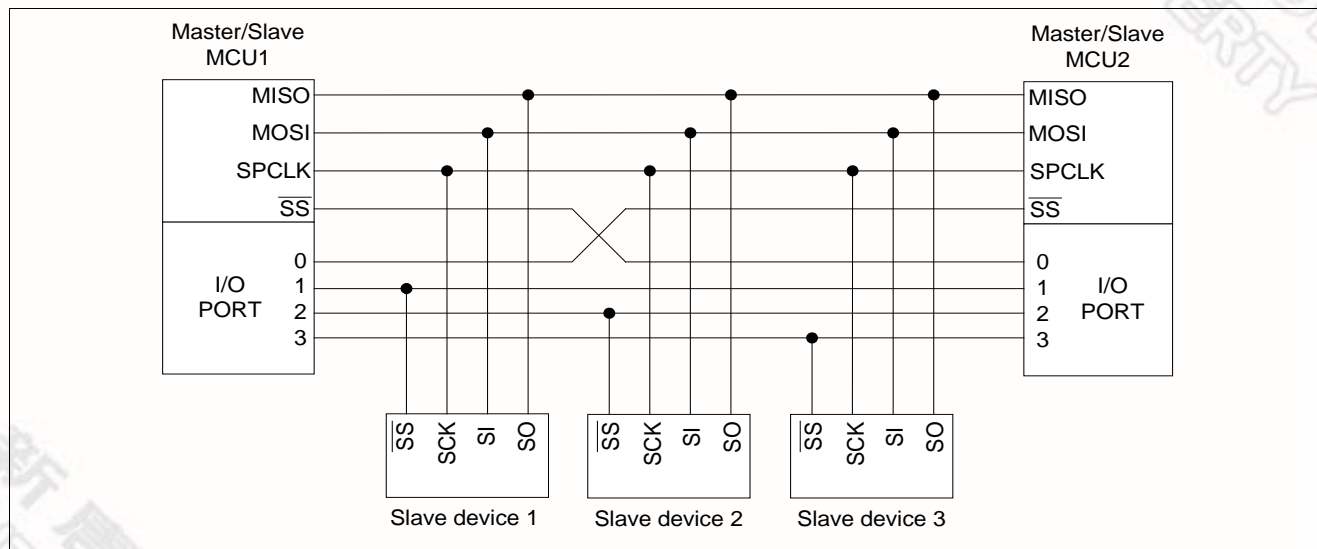


图13-2. SPI 多主机多从机传输机构示意图

图13-2 为典型的SPI设备通信总线通常为 3 信号线相连，MOSI ~ MOSI, MISO ~ MISO, 和 SPCLK ~ SPCLK. 主机通过四线并行连接的方式，每根 $\overline{SS}$ 线分别控制每个从机。MCU1 和 MCU2 可以任意定义为主/从机模式。 $\overline{SS}$ 需配置为主机模式侦测功能 避免多主机冲突。

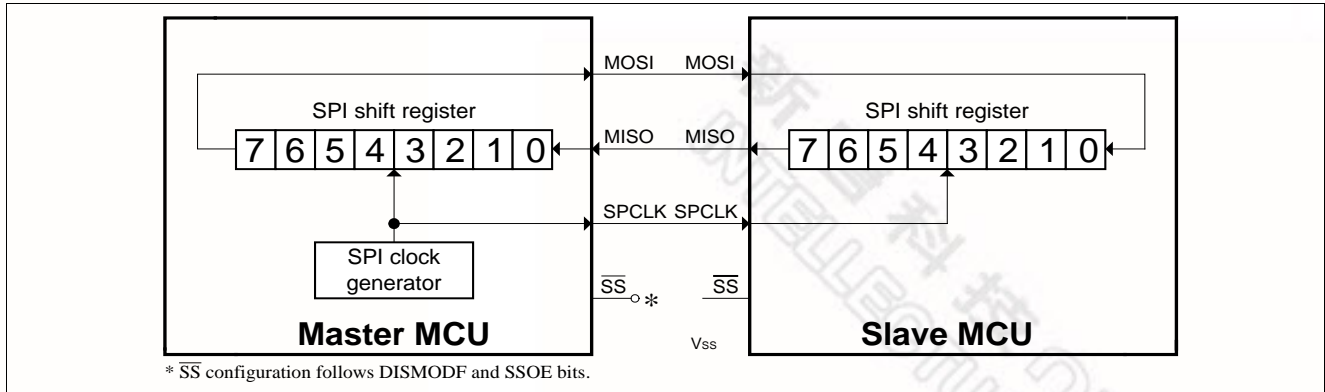


图13-3. SPI 单主机单从机连接示意图

图13-3 表示SPI 模块单主机/从机互连简图。在传输时，主机通过MOSI线向从机发送数据。同时，主机也通过MISO线由从机接收数据。此时主机和从机的两个数据寄存器可被视为一个16位的循环位移寄存器。因此，当主机向从机某地址送数据时，从机内该地址内的数据同时也由从机推向主机。传输进行了交换的动作。

默认情况下，SPI先发送MSB。当LSBFE (SPCR.5) 置1，SPI首先发送LSB，该位不会影响寄存器内MSB/LSB的排列顺序。注，下述全部基于LSBFE为0的情况，MSB 首先被发送。

### 13.3 SPI控制寄存器

控制寄存器 (SPCR), SPI 状态寄存器 (SPSR), SPI 数据寄存器 (SPDR) 这三个寄存器用于SPI传输。这些寄存器提供控制，状态检测，数据存储以及时钟发生设置。SPI\_Sel (AUXR1.7) 用于软件切换定义SPI 管脚位置。

#### AUXR1 – 附加功能寄存器1

7	6	5	4	3	2	1	0
SPI_Sel	UART_Sel	-	-	DisP26	-	0	DPS
r/w	r/w	-	-	r/w	-	r	r/w

地址：A2H

复位值：0000 0000B

位	名称	描述
7	SPI_Sel	<b>SPI管脚选择位</b> 0: 选择P1.7, P1.6, P1.4 和P0.0 作为SPI管脚 1: 选择P2.2, P2.3, P2.4, 和P2.5 作为SPI管脚



## SPCR – 外围设备串行控制寄存器

7	6	5	4	3	2	1	0
SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR1	SPR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: F3H

复位值: 0000000B

位	名称	描述																				
7	SSOE	<p><b>从机选择输出使能位</b></p> <p>该位搭配DISMODF (SPSR.3) 用于定义管脚, 该位仅在MSTR=1和DISMODF=1的条件下有效。</p> <p>注, 仅当MSTR = 1 及DISMODF = 1的条件下该位有效。</p> <p>0 = <math>\overline{SS}</math> 作为普通 I/O.</p> <p>1 = <math>\overline{SS}</math> 选择外部从机驱动 自动拉低, 总线进入空闲模式时变高</p> <p>0 = <math>\overline{SS}</math> 作为通用I/O 引脚.</p> <p>1 = 当选择外部从机器件, <math>\overline{SS}</math> 在每次传输时自动为低, 在每次idle状态重选为从机设备时为高</p>																				
6	SPIEN	<p><b>SPI 使能</b></p> <p>0 = 关闭SPI功能.</p> <p>1 = 打开SPI功能.</p>																				
5	LSBFE	<p><b>LSB 优先使能</b></p> <p>0 = SPI优先传输最高位MSB数据</p> <p>1 = SPI优先传输最低位LSB数据</p>																				
4	MSTR	<p><b>使能主机模式</b></p> <p>该位用于切换SPI工作于主机与从机模式.</p> <p>0 = SPI 配置为从机模式.</p> <p>1 = SPI配置为主机模式.</p>																				
3	CPOL	<p><b>SPI时钟极性位选择</b></p> <p>CPOL 位决定SPI时钟的idle状态电平. 见图13-4. SPI 时钟格式.</p> <p>0 = SPI时钟在idle状态为低.</p> <p>1 = SPI时钟在idle状态为高.</p>																				
2	CPHA	<p><b>SPI 时钟相位选择</b></p> <p>CPHA 位决定spi时钟的数据采样边沿. 见图13-4. SPI 时钟格式.</p> <p>0 = 在SPI时钟的第一个边沿采样.</p> <p>1 = 在SPI时钟的第二个边沿采样.</p>																				
1	SPR1	<p><b>SPI 时钟选择</b></p> <p>这两位搭配确定SPI时钟分频</p> <table border="1"> <thead> <tr> <th>SPR1</th> <th>SPR0</th> <th>分频</th> <th>SPI 时钟速率</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>16</td> <td>1.25M bit/s</td> </tr> <tr> <td>0</td> <td>1</td> <td>32</td> <td>625k bit/s</td> </tr> <tr> <td>1</td> <td>0</td> <td>64</td> <td>312k bit/s</td> </tr> <tr> <td>1</td> <td>1</td> <td>128</td> <td>156k bit/s</td> </tr> </tbody> </table> <p>上述速度基于时钟FSYS= 20MHz条件计算</p>	SPR1	SPR0	分频	SPI 时钟速率	0	0	16	1.25M bit/s	0	1	32	625k bit/s	1	0	64	312k bit/s	1	1	128	156k bit/s
SPR1	SPR0		分频	SPI 时钟速率																		
0	0	16	1.25M bit/s																			
0	1	32	625k bit/s																			
1	0	64	312k bit/s																			
1	1	128	156k bit/s																			
0	SPR0																					

表13-1. 从机选择脚定义

DISMODF	SSOE	主机模式 (MSTR = 1)	从机模式 (MSTR = 0)
0	x	$\overline{SS}$ 作为模式错误输入脚	$\overline{SS}$ 作为从机选择输入脚
1	0	普通端口	
1	1	自动 $\overline{SS}$ 输出	

## SPSR – 串行外围设备状态寄存器

7	6	5	4	3	2	1	0
SPIF	WCOL	SPIOVF	MODF	DISMODF	-	-	-
r/w	r/w	r/w	r/w	r/w	-	-	-

地址: F4H

复位值: 0000 0000B

位	名称	描述
7	SPIF	<b>SPI中断完成标志</b> 在SPI数据传输完成或接收到的数据移入到SPI读缓冲时, 该位通过硬件设置为1. 如果使能 ESPI (EIE .6) 和 EA, SPI中断请求. 该位必须由软件清零. 如果SPIF置位, 禁止向SPDR写入
6	WCOL	<b>写冲突位</b> 该位表示写冲突事件. 一旦发生写冲突事件, 该位被置位, 必须通过软件清零.
5	SPIOVF	<b>SPI 溢出标志</b> 该位表示溢出事件, 一旦发生溢出, 该位置位, 如果使能ESPI 和 EA, SPI请求中断. 该位必须由软件清零
4	MODF	<b>模式错误中断状态标志</b> 该位表示模式错误事件. 如果 $\overline{SS}$ 配置成模式错误输入(MSTR=1且DISMODF=0) 和 $\overline{SS}$ 被外部器件拉低, 产生模式错误. MODF将被置1. 如果使能 ESPI 和 EA, SPI中断请求. 该位必须由软件清零
3	DISMODF	<b>禁止模式错误检测.</b> 该位结合SSOE (SPCR.7) 位用于决定 $\overline{SS}$ 的特征. DISMODF 仅在主机模式下有效 (MSTR = 1) 0 = 不禁止模式错误检测. $\overline{SS}$ 为模式错误检测的输入脚, 不管SSOE 1 = 禁止模式错误检测. $\overline{SS}$ 的特征依赖SSOE 位
2:0	-	保留.

## SPDR – 串行外设数据寄存器

7	6	5	4	3	2	1	0
SPDR[7:0]							
r/w							

地址: F5H

复位值: 0000 0000B

位	名称	描述
7:0	SPDR[7:0]	<b>串行外设数据寄存器</b> 该字节为SPI总线上传输或接收的数据。这该字节写入执行对移位寄存器写操作。读取这个字节, 实际上是一个缓冲区读取数据读取。在主机模式, 写该寄存器同时初始化传输和一个字节接收。

## 13.4 工作模式

### 13.4.1 主机模式

对MSTR (SPCR.4)位置1, 芯片作为主机模式开始SPI传输模块开始工作。整个SPI系统中只允许一个主机启动传输。每次传输总是由主机发起, 对主机SPDR寄存器的写开始传送。在SPCLK控制下在MOSI管脚传送数据。8位数据传输完毕, SPIF (SPSR.7)由硬件自动置位以示完成一个字节数据传输, 同时由从机接收到的数据传送到SPDR。从SPDR读出数据后, 用户才可以清除SPIF。

### 13.4.2 从机模式

设定MSTR为0, SPI将工作在从机模式。当作为从机模式时, SPCLK管脚变为输入脚, 它将被另外一个主机的SPI设备控制,  $\overline{SS}$  管脚也变为输入, 同样, 在数据传输完成前保持低电平状态。如果 $\overline{SS}$ 变为高电平, SPI将被迫进入闲置状态。如果 $\overline{SS}$ 管脚在传输的过程被置高, 那么传输将被取消, 同时接受数据的缓存区也将进入闲置状态。

在从机模式下, 数据在MOSI管脚从主机向从机流动, 在MISO管脚从从机向主机流动。根据SPCLK的时钟控制数据由主机位移传入, 每次一个字节传输完成SPIF置1, 此时读取SPDR寄存器即为该字节内容。对SPDR的读实际上就是对缓冲器的读。为了防止缓冲器溢出和由于溢出导致的数据丢失, SPIF必须在数据第二次从移位寄存器向读缓冲器传送前清零。

## 13.5 时钟格式和数据传输

为了适应各种各样的同步串行外设, SPI提供时钟极性位CPOL (SPCR.3)和时钟相位位CPHA (SPCR.2)寄存器用以控制。如图13-4 SPI时钟格式所示, CPOL和CPHA组和出四种不同的时钟格式。CPOL位表示空闲状态时SPCLK脚电位。CPHA位表示是由MOSI或由MISO上那条线的边缘采样。在同一系统上的主从设备中, CPOL和CPHA的应该是相同的, 传输不同的数据格式, 将产生随机错误结果。

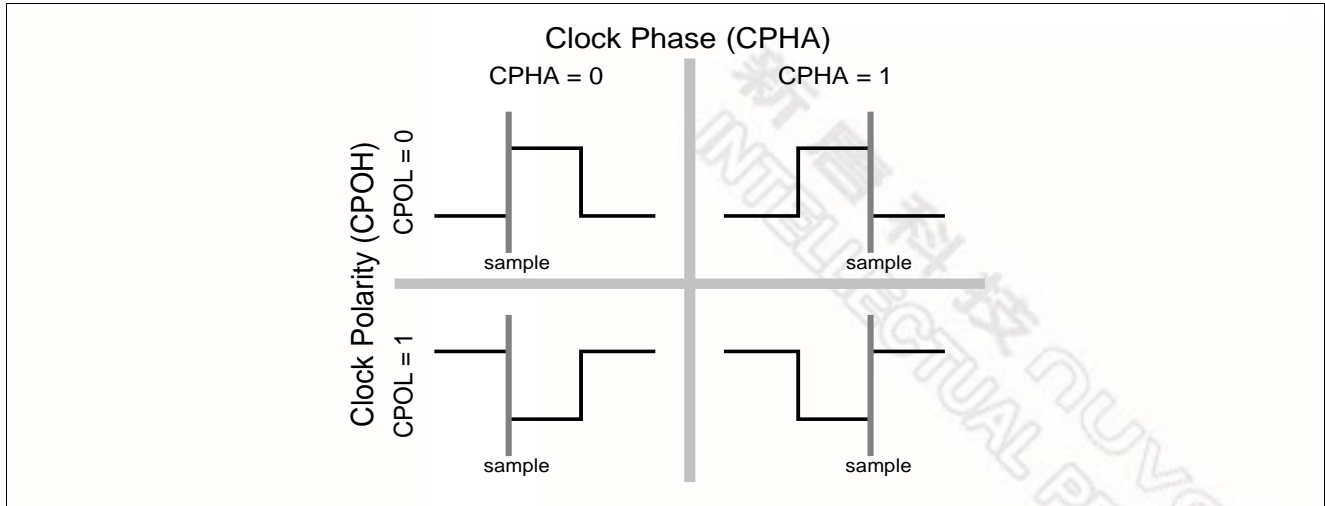


图13-4. SPI 时钟格式

在SPI传输中，总是由主机启动传输。如果SPI被选定作为主模式（MSTR = 1）并且打开传输（SPIEN = 1），对主机的SPI数据寄存器（SPDR）写入内容将启动SPI时钟和数据传输。传出一个字节的会同时接受一个字节的的内容，此后SPI时钟停止，主机和从机的SPIF（SPSR.7）同时被置1。如果SPI中断使能位ESPI（EIE.6）设置为1，全局中断使能（EA= 1），芯片将执行SPI（ISR）的中断服务程序。

关于从机模式下， $\overline{SS}$  信号需要注意。如图13-4SPI时钟格式所示，CPHA=0时，第一个SPCLK边沿为MSB的采样点（LSBFE= 0，MSB优先发送为例）。因此，从机必须在SPCLK第一个采样边沿出现之前先把MSB传出。 $\overline{SS}$  的下降沿可用于准备MISO的MSB。因此，每次成功串行传输一个字节后，该引脚必须切换先高然后低，每个成功逐次串行字节之间。此外，如果从将数据写入SPI数据寄存器（SPDR）时，如果 $\overline{SS}$  为低电位，则会发生写冲突错

当CPHA = 1，采样边沿位于SPCLK时钟的第二个边沿。从机使用的第一个SPCLK时钟转移的MSB，而不是 $\overline{SS}$  的下降沿。因此，在每次成功传输时 $\overline{SS}$  可以始终保持低电位保持低之间的转移。此格式更适合单主机单从机的结构使用。从机的 $\overline{SS}$  可以不连接在SPI系统中，直接接地。

**注意：所有寄存器配置必须在传输开始（SPIEN=1）之前完成，传输过程中，对LSBFE、MSTR、CPOL、CPHA 和 SPR[1:0]任何一个寄存器进行更改，都会使SPI传输中止并进入闲置状态。因此，在更改任何寄存器之前，先停止SPI传输（SPIEN=0）**

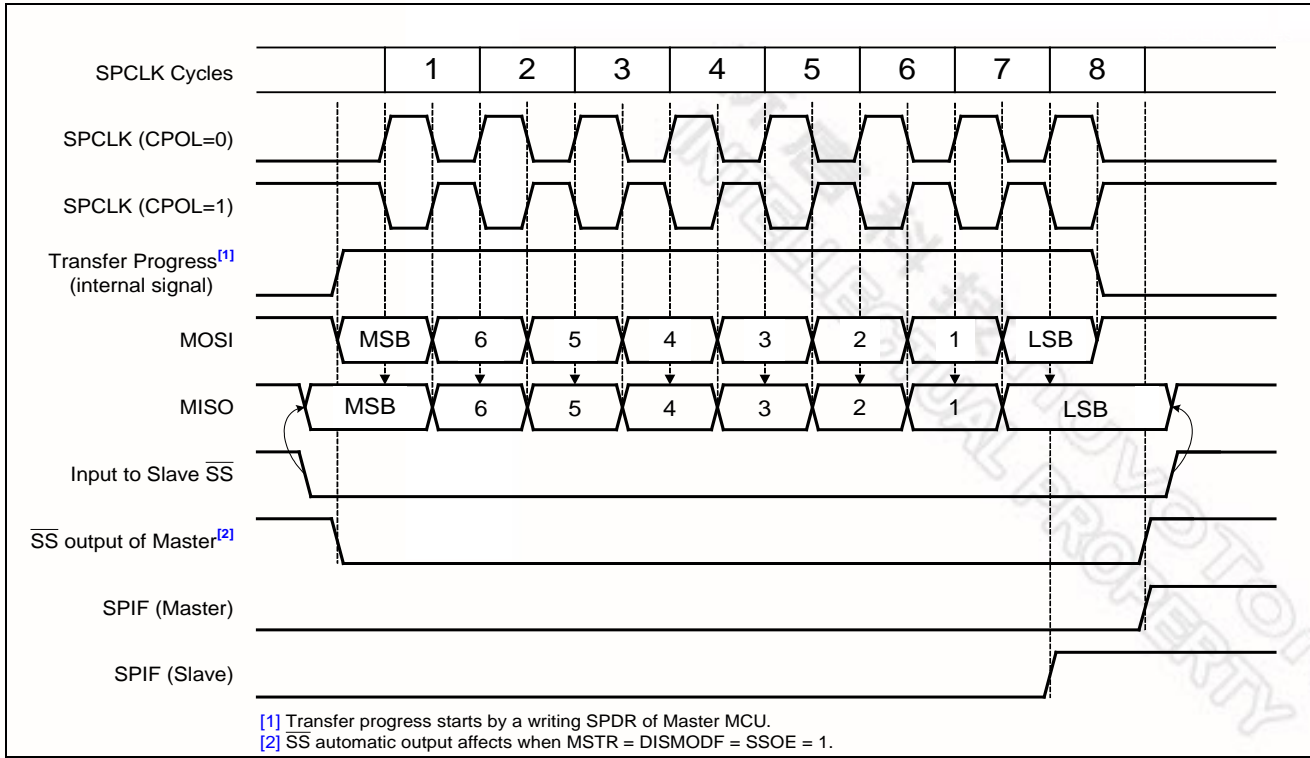


图13-5. CPHA = 0时SPI 时钟与数据格式

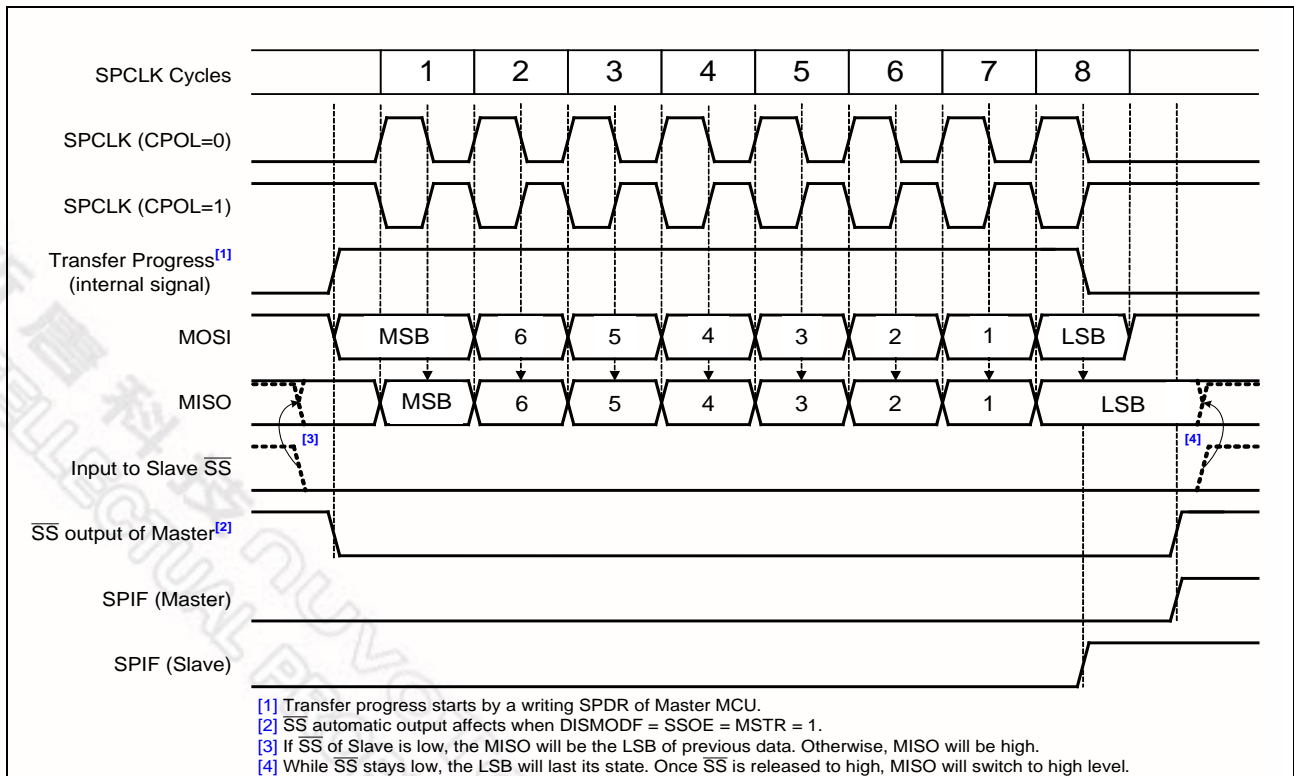


图13-6. CPHA = 1时SPI 时钟与数据格式

### 13.6 从机选择脚配置

N79E815A/814A/8132A SPI 提供灵活的  $\overline{SS}$  配置用于不同系统。当作为从机时， $\overline{SS}$  时钟定一位从机选择输入脚，当作为主机时， $\overline{SS}$  有三种不同的功能定义，可以通过DISMODF (SPSR.3) 和SSOE (SPCR.7)配置。默认情况DISMODF=0，故障侦测功能打开， $\overline{SS}$  配置为输入脚并检测是否发生故障。反之，如果DISMODF=1，故障侦测功能关闭，SSOE寄存器定义控制 $\overline{SS}$ 管脚。当SSOE=1，从机选择信号自动生成，主机的 $\overline{SS}$ 管脚直接与从机的 $\overline{SS}$ 脚连接，当选择外部从机进行传输时 $\overline{SS}$ 自动拉低，当进入闲置状态或者没有选择从机时，自动拉高。当SSOE=0且DISMODF=1时， $\overline{SS}$ 不再用作SPI管脚，而完全配置为普通端口状态。

### 13.7 模式故障侦测

在一个SPI网络中，当不止一个设备有可能成为主机时，为减少数据传输错误，模式故障侦测功能是非常有用的。模式故障侦测发现SS由其它设备拉低，说明系统上有一个从机试图寻找主机地址并把注主机认为成从机。此时，硬件回自动将SPCR的MSTR和SPIEN清除，从而SPI功能关闭，莫使错误侦测标志MODF (SPSR.4)置1，如果之前已打开中断ESPI (EIE .6) 和EA置1，则会进入中断向量。

### 13.8 写冲突错误

写冲突检测显示当正在进行一次传送时，设备正在试图写数据到SPDR。SPDR在传送方不是双缓冲器，对SPDR的写被直接写进SPI移位寄存器，如果这种写在转移过程中被误用，将发生一个写冲突错误(WCOL(SPSR.6)将被置位)。如果转移连续稳定没有受到干扰，那么导致错误的写数据是没有写进移位装置。一次写冲突通常是一个从机错误，原因是当主机开始一次传送时主机知道传送正在进行，所以主机没有理由产生写冲突错误，尽管SPI逻辑可以在主机和从机之间进行写冲突检测。WCOL标志用软件清除。

SPI是信号对于接收数据，是双向缓冲的。在前一笔数据传输完成之前不能对传输缓冲写入新的数据。否则对SPDR写入内容会引起写冲突错误。对于发送端SPDR不是双缓冲的，此时对SPDR写入数据被直接写入到SPI移位寄存器。一旦产生写冲突错误，WCOL (SPSR.6) 将通过硬件设置为1表示写冲突。在这种情况下，当前的数据传输继续其传送，而新的数据将会丢失。虽然SPI逻辑可以检测写在主机模式和从机模式的冲突，写冲突通常是从机错误引起的，因为当主机启动传输时，从机并没有指示一个从机没有指示标志。在从机接收模式，对SPDAT写入内容，也会引起写冲突错误。WCOL标志需用户软件清除。

### 13.9 溢出错误

对于接收数据，SPI是双向缓冲的。接收数据时，数据移入一个并行的数据缓冲去中，位移器同时清空以接收下一个数据。因此，在下一个数据传入之间，必须确保从SPDR中读取出当前数据。当第一个字节从缓冲区内被读取，并且SPIF被清零后，模块准备接收下一个数据，这样的传输不会产生溢出错误。反之，产生溢出错误时，第二个字节的数据没有正常传入，缓冲区内仍保留有第一个字节的数据。当发生溢出错误时，SPIOVF (SPSR.5)会被硬件置1。如果中断打开，会进入中断向量。[图13-7. SPI](#) 表示接收数据与溢出错误之间的关系。

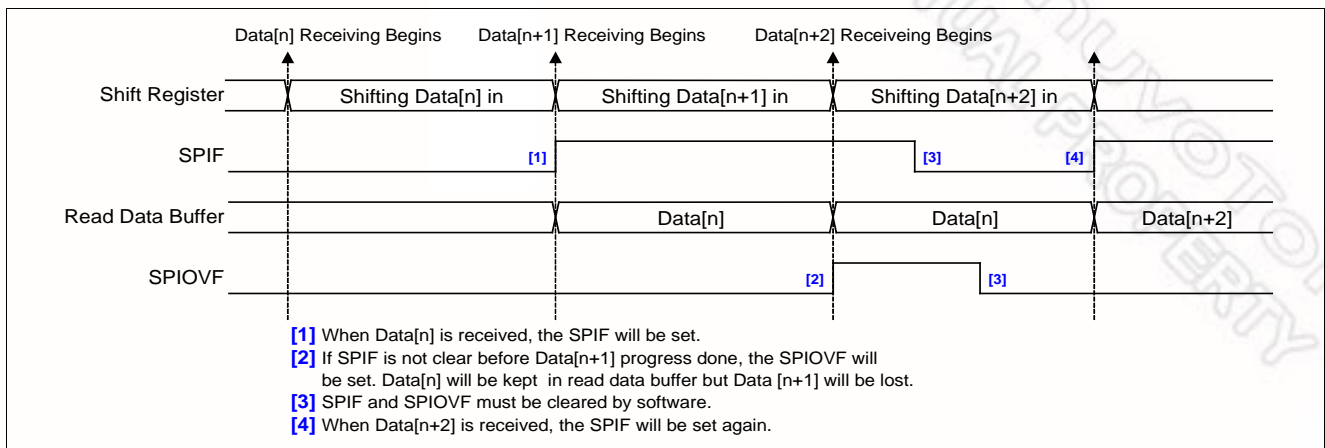


图13-7. SPI 溢出错误波形图

### 13.10 SPI 中断

SPI中断状态标志包括SPIF、MODF 和 SPIOVF 用于产生SPI中断需求。当有外部数据传入SPDR或自身完成数据传输后，这些位于SPSR和SPIF内的标志位会被置起。MODF置1时，表示 $\overline{SS}$ 进入模式错误状态，SPIOVF表示接收发生溢出错误。当SPI中断打开时（ESPI (EIE.6) 和EA置1），当这3个标志中的任意一个置1，CPU会执行SPI中断服务程序。用户若需要了解是由何种标志引起中断，必须检查相应的标志位。这三个标志必须由用户软件清除。

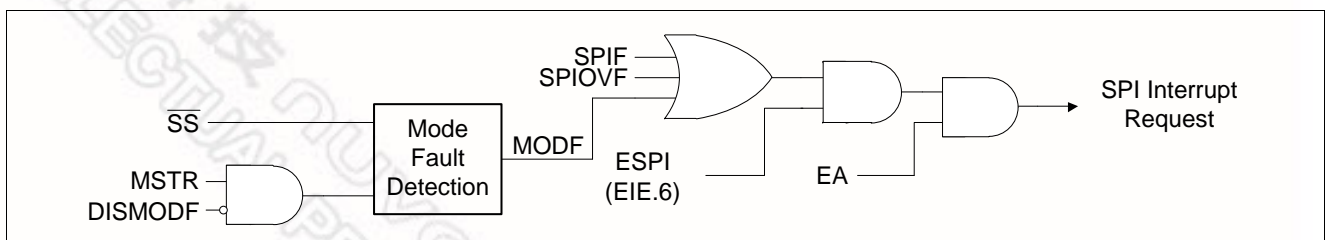


图13-8. SPI 中断请求



## 范例

```

ORG 0000H
LJMP    START

ORG 004BH
LJMP    SPI_ISR

ORG 0100H

SPI_ISR:
ANL     SPSR,#7FH
reti

START:
ANL     SPCR,#0DFH      ; MSB first
ANL     SPCR,#0F7H      ; The SPI clock is low in idle mode
ORL     SPCR,#04H       ; The data is sample on the second edge of SPI clock
ORL     SPCR,#10H       ; SPI in Master mode
ANL     SPCR,#0FCH      ; SPI clock = Fosc/16
SETB    ESPI           ; Enable SPI interrupt
SETB    EA
ORL     SPCR,#40H       ; Enable SPI function

MOV     SPDR,#90H       ; Send 0x90 to Slave
ORL     PCON,#01H      ; Enter idle mode

SJMP    $
END

```



## 14 键盘中断 (KBI)

N79E815A/814A/8132A系列有用于检测按键状态的8个键盘中断功能，如下图所示，当任何按键按下或键盘连接到N79E815A/814A/8132A系列的特定脚可以单独产生一个中断。该中断常用于把CPU从掉电模式或空闲模式唤醒。

端口P0支持键盘功能，P0的所有的管脚都可以产生中断。在KBI寄存器中可以设置端口允许KBI0 ~ KBI7功能，如下表。当任何使能的引脚受到触发时，KBI中断功能激活，则KBIF(EAH)的键盘中断标志KBIF[7:0]置位，如果中断使能，则产生中断。KBIF[7:0]位由硬件置位，必须由软件清零。并可通过该寄存器了解是端口P0的哪一位引起KBI中断。

KBI 支持四种中断条件判别，包括低电平、下降沿、上升沿或上升和下降任意边沿。触发条件根据寄存器KBLS1(ECH).x 和KBLS0(EBH).x 决定。

触发产生后经过两个机器周期，KBIF置位。

KBI 通用用于检测外设如键盘等的边沿传输。在空闲状态中，系统先进入掉电模式以减小功耗并等待事件触发。

N79E815A/814A/8132A系列支持KBI中断，将MCU从掉电模式下唤醒。注意，如果KBI选作任何边沿触发模式，必须遵循如下条件，以确保芯片能有效得从掉电模式中唤醒。对于下降沿唤醒，引脚状态在进入掉电模式的时候必须为高。相应地，上升沿唤醒时引脚状态必须为低。

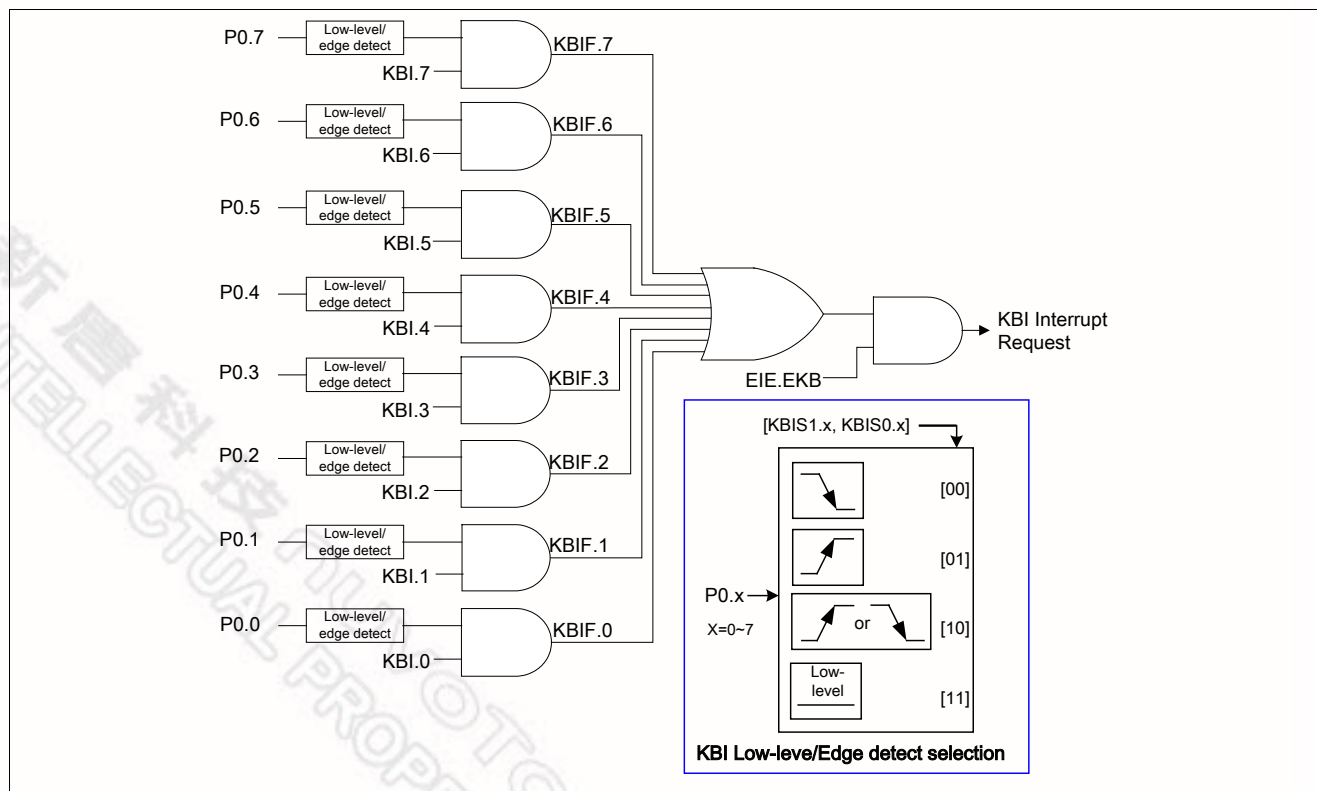


图14-1 键盘中断检测

表14-1. KBI响应条件设定表

KBLS1.n	KBLS0.n	KBI 响应条件
0	0	下降沿
0	1	上升沿
1	0	上升或下降沿都相应
1	1	低电平

## KBIE – 键盘中断使能寄存器

7	6	5	4	3	2	1	0
KBIE.7	KBIE.6	KBIE.5	KBIE.4	KBIE.3	KBIE.3	KBIE.1	KBIE.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: E9H

复位值: 0000 0000B

位	名称	描述
7:0	KBIE	键盘中断 使能P0[7:0] 引起键盘中断.

## KBIF – Keyboard Interface Flags

7	6	5	4	3	2	1	0
KBIF[7:0]							
r (level)							
r/w (edge)							

地址: EAH

复位值: 0000 0000B

位	名称	描述
7:0	KBIFn	键盘接口通道标志. KBI的任何边沿触被选择, 如果KBI通道n检测到所定义的边沿信号, 该标志由硬件置位. 该标志由软件清零. 如果KBI选择为低电平触发模式, 如果在KBI通道输入信号电平为1, 该标志取反. 软件不能控制它.

KBLS0 – 键盘电平选择位 0 <sup>[1]</sup>

7	6	5	4	3	2	1	0
KBLS0[7:0]							
r/w							

地址: EBH

复位值: 0000 0000B

位	名称	描述
7:0	KBLS0[7:0]	键盘电平选择位0.

**KBLS1 – 键盘电平选择位 1<sup>[1]</sup>**

7	6	5	4	3	2	1	0
KBLS1[7:0]							
r/w							

地址: ECH

复位值: 0000 0000B

位	名称	描述
7:0	KBLS1[7:0]	键盘电平选择位1

[1] KBLS1 与 KBLS0 搭配决定KBI的侦测类型，详见表14-1 KBI响应条件设定表。

## 15 模数转换 (ADC)

N79E815A/814A/8132A系列有一个ADC采用DAC逐次比较寄存器的内容转换成电压(VDAC)；比较器是把模拟输入电压 ( $V_{in}$ )和(VDAC)相比较并回馈到逐次比较控制器。ADCCON0寄存器中的ADCS位设为‘1’，就开始了一次转换。ADCS可以只有软件置位也可以由硬件或软件置位。

当禁止ADC 功能时, 所有 ADC的相关SFR位将无效。

ADC 模块的功耗几乎为0。

通过控制位ADCCON0.5 (ADCEX) =0, 选择仅由软件启动模式。设置ADCCON0.3 (ADCS )位开始转换, 硬件或软件开始模式由ADCCON0.5 (ADCEX) =1选择, 可通过设置ADCCON0.3或在STADC提供上升沿信号使转换开始.当转换开始是由上升沿触发的, 在至少一个机器周期的高电平之后必须有一个机器周期的低电平到STADC。

STADC上的由低到高的电平翻转被视作一个机器周期的结束,转换在下一个周期的起始处开始. 当软件初始化转换, 转换开始在指令设置ADCS之后的一个机器周期的起始处。 ADCS实际实际是一个触发器: 命令触发转换, 状态标志在读操作时访问。

下两个机器周期用于初始化转换器, 第一个周期的结束, ADCS状态标志位置位, ADCS标志返回“1”。模拟输入采样在第二个周期结束时开始。

在接下去的8个机器周期里, 当前端口0种使能的引脚电压被采样, 输入电压应该稳定以获得有效采样数据。 任何情况下, 输入电压的频票必须低于10V/ms 以防止出现不确定结果。

逐次逼近控制逻辑首先设置标志位, 并清除逐次逼近寄存器(10 0000 0000b)所有位, DAC的输出满量程的50%与输入电压 $V_{in}$ 相比较。如果输入电压大于 $V_{DAC}$ , 该位保持置位, 否则清零。

逐次逼近控制逻辑设置下一个标志(根据当前结果为11 0000 0000b 或 01 0000 0000b),  $V_{DAC}$ 再次与 $V_{in}$ 比较。如果输入电压大于 $V_{DAC}$ , 该位保持置位, 否则清零。该过程一直重复直到所有10位都测试完, 此阶段, 转换结果保存在逐次逼近寄存器中。每位转换需要4个机器周期。

10位转换结束由控制位ADCCON0.4 (ADCI)标示, 结果的高8位保存在寄存器 ADCH, 低两位保存在ADCCON0.7 (ADC.1) 和 ADCCON0.6 (ADC.0)。用户可以忽略低两位而使用ADC为8位转换器。任何情况下, 实际转换时间35个机器周期后, ADC将置位, ADCS状态标志将在ADCS置位后的35个周期后复位。

控制位 ADCCON0.0 ~ ADCCON0.2 用于控制模拟多路转换器, 以选择8个模拟通道。 ADC转换过程中不受外部或ADC开始的软件设置影响, 转换完成的结果不受ADCI=1的影响; 当系统进入空闲或掉电模式时, 一个新的正在进行的ADC转换会中止。当进入空闲模式, 转换完成的结果保持不变 (ADCI = 逻辑1)。

当 ADCCON0.5 (ADCEX)由外部引脚触发开始ADC转换，在N79E815A/814A/8132A系列进入空闲模式，P1.4 在一个机器周期后可以开始ADC转换。

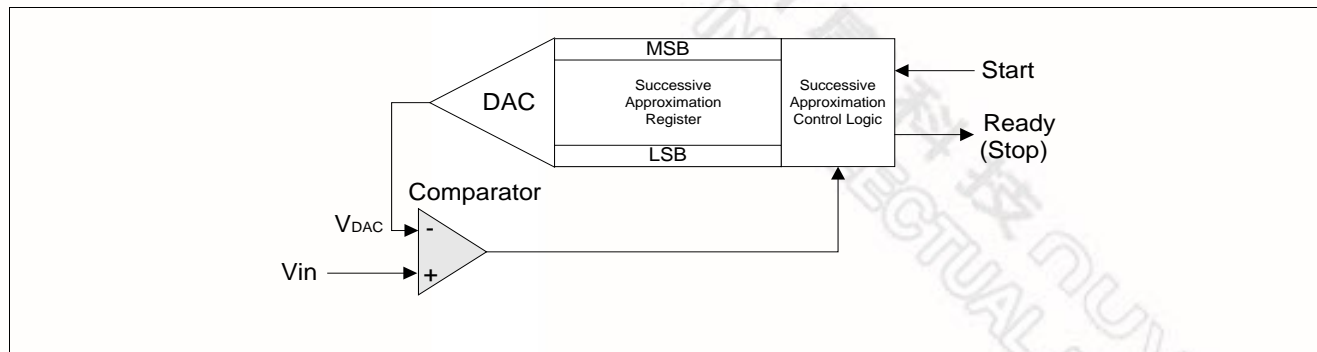


图15-1 逐次逼近型ADC结构

ADC有自己的电源输入脚 ( $AV_{DD}$  和  $AV_{SS}$ ) 和一个参考电压输入脚 ( $V_{ref+}$ ) 连接到DAC的各自电阻阶梯网络。该阶梯网络由电阻分成1023个相等的阶梯；第一个阶梯是  $AV_{SS}$  上的  $0.5XR$ ，最后一个阶梯是  $V_{ref+}$  下的  $0.5XR$ 。总共有  $1024XR$  电阻阶梯。该结构确定DAC的单调变化并均匀误差。

如果输入电压在  $AV_{SS}$  和  $[(V_{ref+}) + \frac{1}{2} \text{LSB}]$  之间，10-位ADC结果会是  $00\ 0000\ 0000\ b = 000H$ ；如果输入电压在  $[(V_{ref+}) - \frac{3}{2} \text{LSB}]$  和  $V_{ref+}$  之间，10-位ADC结果会是  $11\ 1111\ 1111\ B = 3FFH$ 。  $AV_{ref+}$  和  $AV_{SS}$  可以在  $AV_{DD} + 0.2V$  和  $AV_{SS} - 0.2V$  之间。  $AV_{ref+}$  应该是比  $AV_{SS}$  的电位高，并且输入电压 ( $V_{in}$ ) 应该在  $AV_{ref+}$  和  $AV_{SS}$  之间。

下面是计算结果方程式:

$$\text{结果} = 1024 \times \frac{V_{in}}{AV_{ref+}} \quad \text{或} \quad \text{结果} = 1024 \times \frac{V_{in}}{V_{DD}}$$

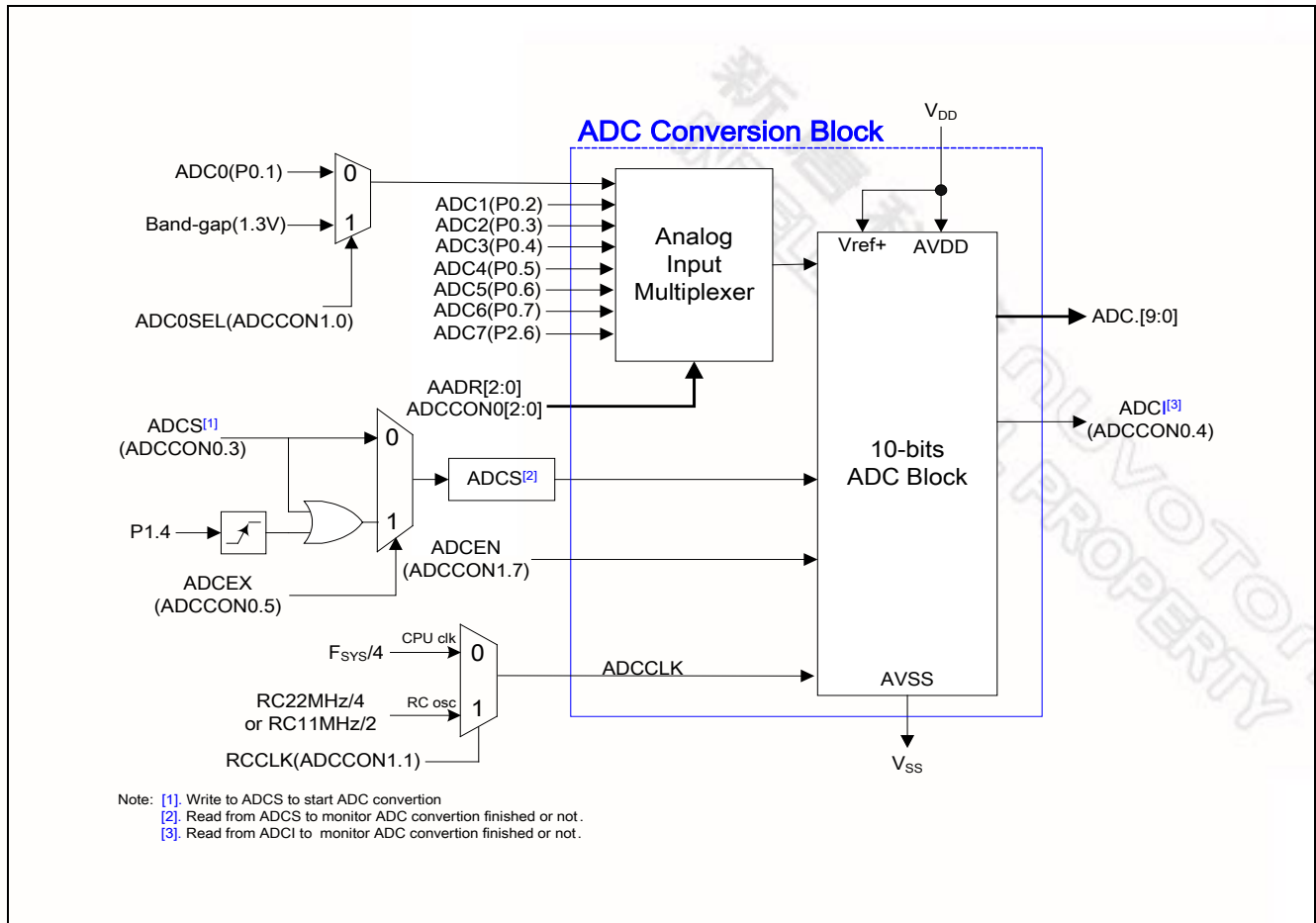


图15-2 ADC 结构框图

**ADCCON0 – ADC 控制寄存器 0**

<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
ADC.1	ADC.0	ADCEX	ADCI	ADCS	AADR2	AADR1	AADR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: F8H

复位值: 0000 0000B

位	名称	描述
7	ADC.1	ADC 转换结果.
6	ADC.0	ADC 转换结果
5	ADCEX	0: 禁止外部P1.4的开始转换 1: 使能外部P1.4的开始转换.
4	ADCI	0: ADC空闲. 1: ADC转换结果已经可以读取. 如果中断使能, 会产生一个中断. 不能由软件置位.
3	ADCS	ADC 开始和状态: 设置该位开始A/D转换. 如果ADCEX为1, 也由STADC置位. 当ADC忙时该位保持高, 在ADCI置位后立即复位. 注: 在置位ADCS之前建议先清ADCI. 然而, 如果ADCI清零与ADCS置位同时进行, 相同通道将开始一次新的A/D 转换. 软件清ADCS将中止转换 当ADCS或ADCI为高时, ADC 不能进行新的转换.
2	AADR2	选择 ADC输入.
1	AADR1	选择 ADC输入.
0	AADR0	选择 ADC输入.

ADCI	ADCS	ADC 状态
0	0	ADC 空闲; 可以开始转换.
0	1	ADC 忙; 阻止新的转换开始
1	0	转换完成; 新的转换开始要求 ADCI = 0
1	1	转换完成; 新的转换开始要求 ADCI = 0

如果ADCI清零与ADCS置位同时进行, 相同通道将开始一次新的A/D 转换. 建议在ADCS置位前先复位ADCI.

**AADR2, AADR1, AADR0: ADC 模拟输入通道选择位:**

**当 ADCI 和 ADCS 都为 0 时, 这些位才可被 改变.**

AADR2	AADR1	AADR0	ADC 输入通道选择
0	0	0	ADC0 (P0.1)
0	0	1	ADC1 (P0.2)
0	1	0	ADC2 (P0.3)
0	1	1	ADC3 (P0.4)
1	0	0	ADC4 (P0.5)

1	0	1	ADC5 (P0.6)
1	1	0	ADC6 (P0.7)
1	1	1	ADC7 (P2.6)

**ADCH – ADC 转换结果寄存器**

7	6	5	4	3	2	1	0
ADC.9	ADC.8	ADC.7	ADC.6	ADC.5	ADC.4	ADC.3	ADC.2
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: E2H

复位值: 0000 0000B

位	名称	描述
7:0	ADCH	ADC 转换结果位[9:2].

**ADCCON1 – ADC 控制寄存器**

7	6	5	4	3	2	1	0
ADCEN	-	-	-	-	-	RCCLK	ADC0SEL
r/w	-	-	-	-	-	r/w	r/w

地址: E1H

复位值: 0000 0000B

位	名称	描述
7	ADCEN	0: 关闭 ADC 电路 1: 打开 ADC 电路
6:2	-	保留
1	RCCLK	0: FSYS/4 时钟用作 ADC 时钟. 1: 内部 RC/2 时钟用作 ADC 时钟.
0	ADC0SEL	0: 选择 ADC 通道 0 为输入. 1: 选择 带隙电压 (~1.3V) 为输入.

**P0DIDS – 端口 0 数字输入关闭**

7	6	5	4	3	2	1	0
P0DIDS[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: F6H

复位值: 0000 0000B

位	名称	描述
7:0	P0DIDS.x	0: 打开端口 0 的数字逻辑输入输出功能 1: 关闭端口 0 的数字逻辑输入输出功能



## AUXR1 – 附加功能寄存器1

7	6	5	4	3	2	1	0
SPI_Sel	UART_Sel	-	-	DisP26	-	0	DPS
r/w	r/w	-	-	r/w	-	r	r/w

地址: A2H

复位值: 0000 0000B

位	名称	描述
3	DisP26	0: 打开P2.6 数字输入输出功能. 1: 关闭P2.6数字逻辑输入输出功能, 仅为ADC 通道7使用。

ADC通道0 时钟源= Fsys/4 例程如下:

```

ORG 0000H
LJMP    START

ORG 005BH           ; ADC Interrupt Service Routine
CLR     ADCI        ; Clear ADC flag
reti

START:
ORL     P0DIDS,#02H ; Disable digital function for P0.1
ORL     P0M1,#02H  ; ADC0(P0.1) is input-only mode
ANL     P0M2,#0FDH
ANL     ADCON0,#0F8H ; ADC0(P0.1) as ADC Channel
ANL     ADCCON1,#0FDH ; The FSYS/4 clock is used as ADC clock.
SETB    EADC       ; Enable ADC Interrupt
SETB    EA
ORL     ADCCON1,#80H ; Enable ADC Function

Convert_LOOP:
SETB    ADCS       ; Trigger ADC
ORL     PCON,#01H  ; Enter idle mode
MOV     P0,ADCH    ; Converted Data put in P0 and P1
MOV     P1,ADCL
SJMP   Convert_LOOP

END

```

## 16 I<sup>2</sup>C总线

### 16.1 特征

I<sup>2</sup>C 总线作为微控制器与I<sup>2</sup>C设备之间的串行接口，I<sup>2</sup>C设备如EEPROM，LCD 模组等。连接在2线制(SDA和SCL) I<sup>2</sup>C总线上的2个设备之间可以相互交换信息。

I<sup>2</sup>C 总线使主机和从机之间数据可以双向传输。没有中心主机，且通过仲裁允许同时多主机系统。同步串行时钟允许器件之间通过一条串行总线以不同位率传输。I<sup>2</sup>C总线支持4种传输模式，包括主机发送模式，主机接收模式，从机发送模式和从机接收模式。I<sup>2</sup>C 接口仅支持7位寻址模式和广播呼叫。I<sup>2</sup>C 支持标准速率(最多100kbps) 和快速 (最多400kbps)。

### 16.2 功能描述

对于双向传输操作，SDA和SCL引脚必须开漏模式。该接口最基本的操作是执行线与功能。当一个或多个I<sup>2</sup>C器件输出“0”时，I<sup>2</sup>C总线上为低电平。当所有I<sup>2</sup>C器件输出“1”时，产生高电平，允许上拉电阻将总线拉高。

对于N79E815A/814A/8132A，用户通过设置I2CEN (I2CON.6)使能I<sup>2</sup>C功能之前，应该设置P1.2和P1.3输出为逻辑“1”。一旦使能I<sup>2</sup>C功能，P1.2和P1.3就被配置成开漏I/O。P1M2 和 P1M1 将重新配置。强烈建议设置P1S使能史密特触发输入缓冲，提高毛刺抑止能力。

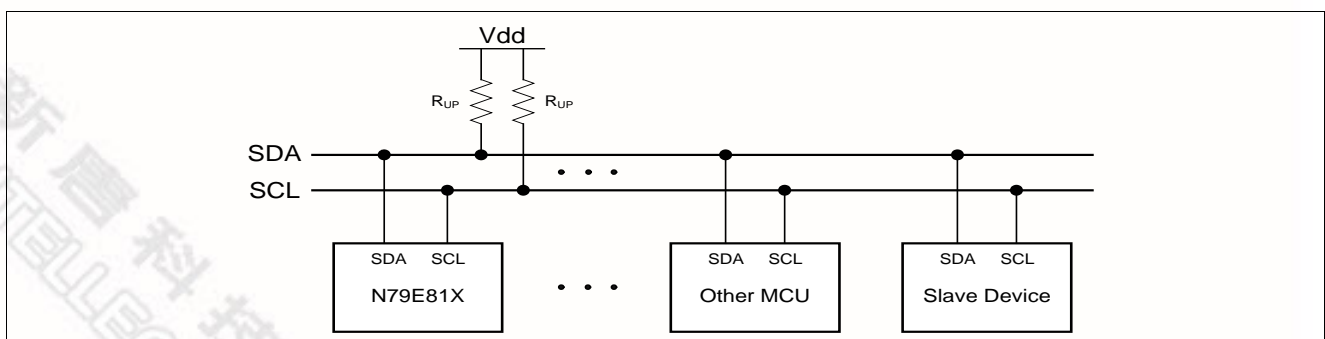
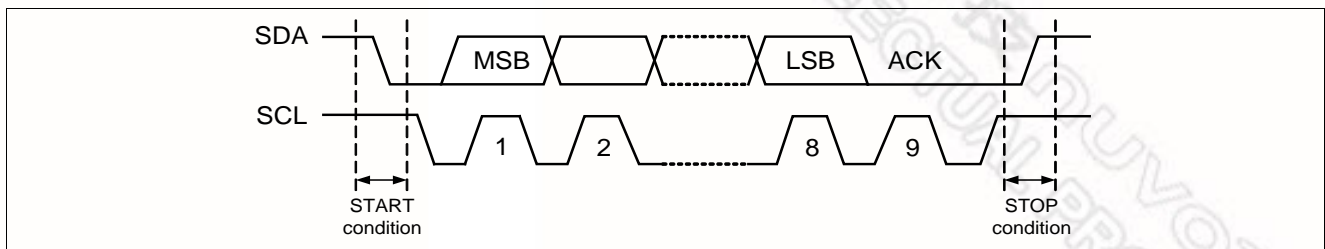


图16-1. I<sup>2</sup>C 总线连接图

当两线都为高时，I<sup>2</sup>C空闲。同时，任何器件可作为主机占用总线并在产生起始信号后传输，在发送停止信号结束传输之前，总线视为忙。主机产生所有串行时钟脉冲和起始与停止信号。然而如果总线上没有起始信号，所有器件均作为没被寻址的从机。硬件寻找自己的从机地址或广播呼叫地址。(广播呼叫地址检测由GC (I2ADDR.0)使能或禁止。)如果接收到的地址匹配时，请求中断。

I<sup>2</sup>C总线上的每个传输为9位长度，由8位数据（先MSB）和一个应答位组成。每次传输的字节数（一次有效的START与停止信号之间）不受限制，但每个字节后都跟随着一个应答位。主机产生8个时钟脉冲发送8位数据，在SCL总线上的第8个下降沿，器件将SDA改变输出为输入，并在第9个时钟脉冲读应答值。第9个时钟脉冲之后，如果下一次接收还没有准备好，数据接收器件将SCL总线拉低，从而迫使下一字节的传输暂停。当接收器释放SCL总线，数据传输继续。

图16-2. I<sup>2</sup>C 总线数据协议

### 16.2.1 起始START 和停止STOP 信号

I<sup>2</sup>C总线协议定义两个状态开始和结束传输，起始 (S) 和 停止(P) 信号。起始信号：当SCL为高时，在SDA总线上有从高到低的电平变化。停止信号：当SCL为高时，在SDA上有从低到高的电平变化。起始或停止信号常由主机产生，在起始信号产生之后I<sup>2</sup>C总线视为忙，在停止信号之后，I<sup>2</sup>C总线视为空闲。停止信号出现后，主机设备将释放控制权并返回为无寻址从机。因此，原来寻址的从机将变成未被寻址的从机，I<sup>2</sup>C总线空闲并等待下一个起始信号。

通常由主机产生停止信号中止数据传输，然而，如果主机仍希望在总线上通信，就会重复产生起始信号(Sr) 和地址。传输中可能存在着各种结合的读/写格式。

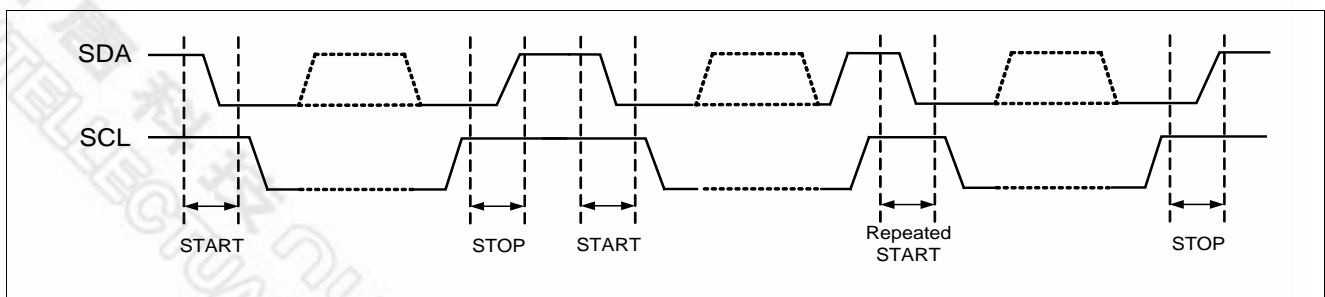


图16-3. 起始START、重复起始Repeated START和停止STOP信号

### 16.2.2 7位地址数据格式

产生起始信号之后，应该由主机传输一个字节的特殊数据，包括在第8位数据方向位 (R/W)之后的7位从机地址 (SLA)，寻址目标从机并决定数据流的方向。如果R/W位为 0，表示主机向所选从机写信息，如果该位为1，表示主机从从机读取信息。一个地址包 由从机地址和 读(R)或写(W) 位组成，分别称之为SLA+R 或 SLA+W。一次传输基本上由一个起始信号，一个 SLA+R/W，一个或多个数据包和一个停止信号组成。在SLA+R/W 指定从机地址后，第二个和之后的8位数据字节由主机或从机地址根据R/W位配置。

“广播呼叫”是个例外，它可以通过将第一个字节的数据全部赋值为0来寻址所有器件。广播呼叫用于当主机希望发送相同信息到几个从机时情况。当该地址在使用时，其他器件根据软件配置可能响应应答或忽略。如果器件响应广播呼叫，其操作就像从机接收器模式。

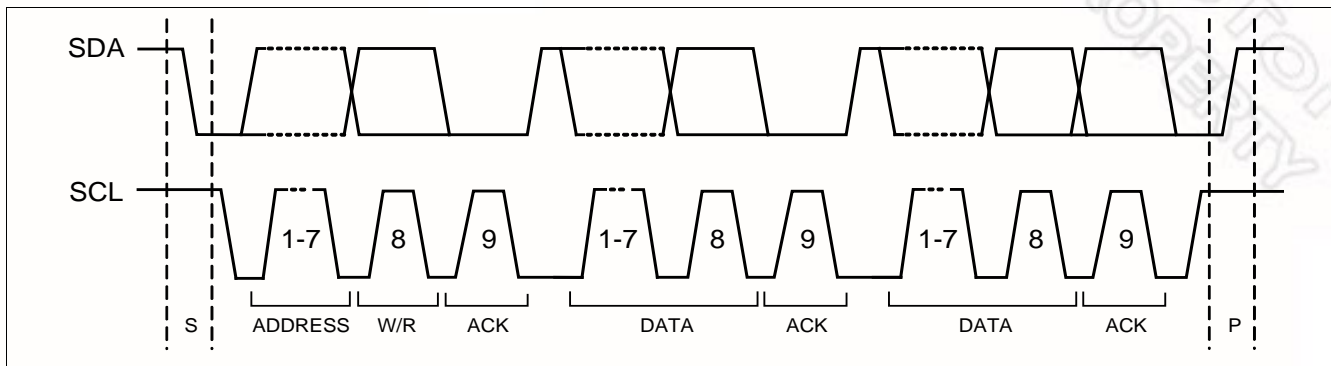


图16-4. I<sup>2</sup>C 传输数据格式

在数据传输过程中，SDA总线上的数据必须在时钟为高的期间内保持稳定，数据总线仅在SCL为低时改变。

### 16.2.3 应答

任何传输字节的第9个SCL脉冲被视作应答信号(ACK)。通过将SDA拉低，允许接收器件(可以为主机或从机)对发送器件的响应(可以为主机或从机)。应答由主机产生相关时钟脉冲，发送器件必须在应答时钟脉冲期间释放SDA总线控制，ACK为低电平有效信号，在时钟脉冲为高时将SDA总线拉低，通知发送器 接收器已经接收到数据。通常，被寻址的接收器在接收到一个字节之后要求产生一个ACK，当从机接收器没有应答 (NACK)从机地址，SDA线必须由从机拉离高电平，以让主机产生停止或重复起始信号。

如果从机接收器应答了从机地址，将切换到不寻址从机模式，不再接收任何数据，该从机保持SDA总线为高，主机应该产生停止信号或复位起始信号。

如果主机接收器传输时，由于主机在传输时控制字节数目，就必须向从机发送器标记数据的结束，而不是在最后一个字节产生应答信号。从机发送器切换到不寻址模式，并释放SDA总线，允许主机产生停止信号或复位起始信号。

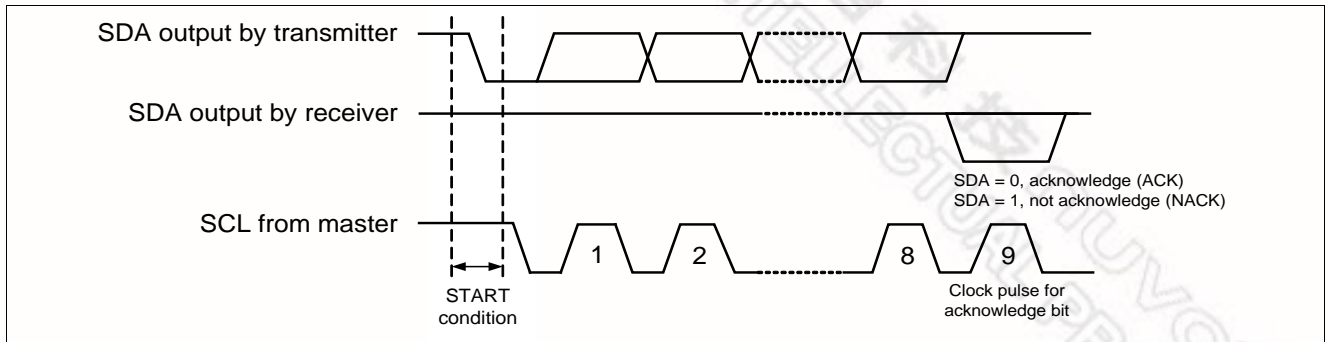


图16-5. 应答位

#### 16.2.4 仲裁

主机仅在总线空闲时开始一次传输，可能是两个或更多主机产生起始信号。在这种情况下，当SCL为高时，就要在SDA总线上有仲裁。仲裁期间，相互竞争的第一个主机设备在SDA上置‘1’(高)，其他主机发送‘0’(低)，由于其电平与自身电平不匹配而关闭其数据输出，仲裁失败的主机立即切换成不被寻址从机，并检测自身的从机地址以判断是否被仲裁胜出的主机寻址，它也释放SDA为高电平，以防影响胜出主机开始数据传输。但是，仲裁失败的主机继续在SCL上产生时钟脉冲，直到它失去仲裁的最后字节输入。如果检测的地址与仲裁失败的主机的自身从机地址相匹配，它就会切换到被寻址的从机模式。

输出数据后，所有主机进行仲裁以持续侦测SDA总线。如果从SDA总线上读取的值与主机输出的值不匹配时，就失去仲裁。注，当一个主机在其他主机输出低时，自身输出一个高的SDA值时就可能失去仲裁，只要有一个主机保存，仲裁就将继续，这可能会占用很多位。如果几个主机试图寻址相同从机时，仲裁将继续输入数据包。

仲裁可以超过几位，第一阶段是比较地址位，如果主机试图寻址相同器件，仲裁继续比较数据位或应答位。

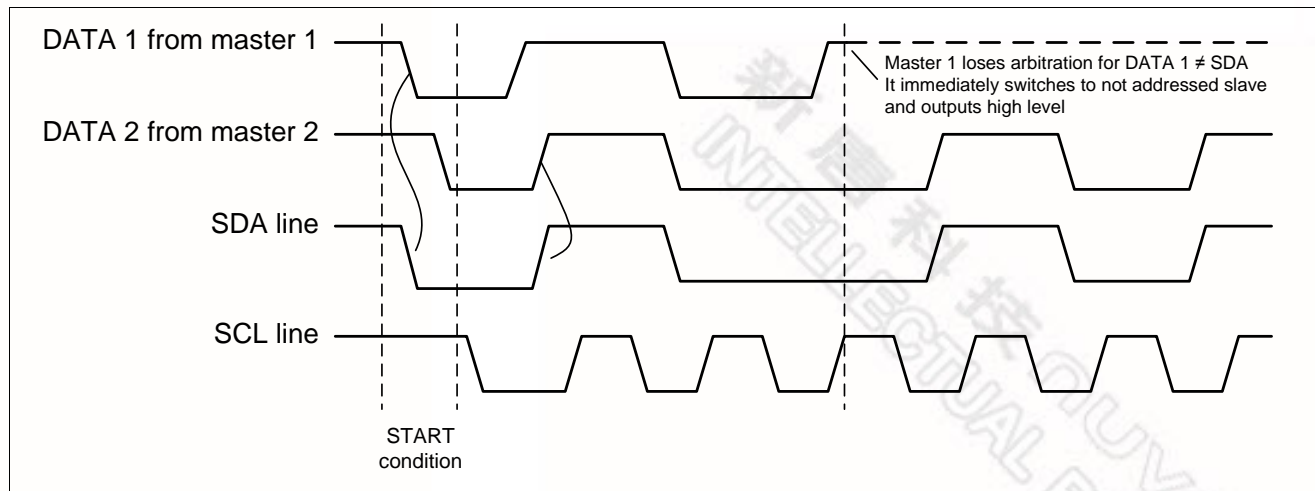


图16-6. 两个主机的仲裁机制

I<sup>2</sup>C总线的控制由地址或主机码和竞争主机发送的数据决定，没有中心主机，总线也没有优先级。

从机没有仲裁过程。

### 16.3 I<sup>2</sup>C 控制寄存器

I<sup>2</sup>C总线有5个控制寄存器，它们是I2CON, I2STA, I2DAT, I2ADDR, I2CLK和 I2TMR。这些寄存器提供协议控制，状态，数据传输和接收功能，时钟速率配置和超时提醒，以下为I<sup>2</sup>C功能相关寄存器描述。

#### I2CON – I<sup>2</sup>C 控制寄存器

7	6	5	4	3	2	1	0
-	I2CEN	STA	STO	SI	AA	-	-
-	r/w	r/w	r/w	r/w	r/w	-	-

地址：C0H

复位值：0000 0000B

位	名称	描述
7	-	保留。
6	I2CEN	<p><b>I<sup>2</sup>C 总线使能</b></p> <p>0 = 禁止 I<sup>2</sup>C 总线。</p> <p>1 = 使能 I<sup>2</sup>C 总线。</p> <p>使能 I<sup>2</sup>C 之前，Px.x 和 Px.x 口必须设置为逻辑1，一旦使能 I<sup>2</sup>C 总线，SDA 引脚 (Px.x) 和 SCL 引脚 (Px.x) 将自动切换到开漏模式，PxM2 和 PxM1 也将被重新配置。</p>

位	名称	描述
5	STA	<p><b>START标志</b></p> <p>当 STA 置位, 如果总线空闲, I<sup>2</sup>C产生起始信号。如果总线忙, I<sup>2</sup>C等待停止信号, 并产生一个起始信号。</p> <p>主机模式下, 当I<sup>2</sup>C准备好和发送或接收一个或多个字节, STA置位, I<sup>2</sup>C产生一个重复起始信号。</p> <p>注STA可在任何时候置位, 甚至在从机模式下, 但STA不能在检测到START或重复起始信号后由硬件自动清零, 用户应该手动清STA。</p>
4	STO	<p><b>停止标志</b></p> <p>I<sup>2</sup>C为主机模式时, STO位置'1', 将在总线上输出停止信号。当检测到总线上出现停止信号, I<sup>2</sup>C硬件清除STO标志。</p> <p>STO标志的设置也用于将I<sup>2</sup>C设备从错误状态(I2STA 为 00H)恢复, 此条件下, 没有停止信号发送到I<sup>2</sup>C总线上。</p> <p>如果STA和STO位都置位, 且在主机模式下设备为原始的, I<sup>2</sup>C总线将产生停止信号并立即伴随着起始信号。如果设备为从机模式, STA和STO同时置位, 应该防止I<sup>2</sup>C帧的错误。</p>
3	SI	<p><b>串行中断标志</b></p> <p>当I<sup>2</sup>C进入25种可能状态之一(除状态 F8H外)时, SI 标志由硬件置位。SI置位后, 软件应该读取寄存器 I2STAT, 以决定哪步已经通过, 下一步采取什么行动。</p> <p>SI 由软件清零, 在SI清零之前, SCL总线处于低周期, 传输暂停, 这对从机器件在准备接受下一个字节之前, 处理当前数据字节很有用。</p> <p>串行传输暂停, 直到软件清SI, 清除SI后, I<sup>2</sup>C总线继续产生START 或重复起始信号、停止信号、8-位数据, 或依据软件配置控制字节或位。</p>
2	AA	<p><b>应答标志</b></p> <p>如果AA标志置位, 当I<sup>2</sup>C设备为接收器时, 在应答时钟脉冲期间将返回ACK。这个接收器可以是主机, 被寻址的从机, 与自身地址匹配的从机, 或广播呼叫的从机。</p> <p>如果AA被清零, 当I<sup>2</sup>C设备是接收器时, 在应答时钟脉冲期间将返回NACK(SDA上为高电平), 这个接收器可以是主机, 被寻址的从机。一个设备的自身AA标志被清零, 将忽略其自身从机地址和广播呼叫, 因此, SI将不被置位, 不请求中断。</p> <p>注, 如果一个被寻址的从机在从机接收模式下不能返回ACK, 或在从机发送模式下不能接收ACK, 从机将变成不被寻址的从机, 在AA标志置位和再将被主机寻址之前不能接收任何数据。</p> <p>I2STA 的一个特殊值C8H发生在从机发送模式下。在从机设备发送最后数据字节到主机之前, AA标志可以被清为0, 然后发送最后数据字节。从机设备将有效切换到不被寻址的从机模式。主机读取的将全为FFH。</p>
1:0	-	保留。



### I2STA – I<sup>2</sup>C 状态寄存器

7	6	5	4	3	2	1	0
I2STA[7:3]						0	0
r						r	r

地址: BDH

复位值: 1111 1000B

位	名称	描述
7:3	I2STA[7:3]	<b>I<sup>2</sup>C 状态码</b> I2STA的高五位包含状态码, 总共有26个可能的状态码, 当I2STA为F8H, 没有相关状态信息且SI标志为0, 所有其他25种状态码响应I <sup>2</sup> C状态, 当进入这些状态时, SI将被置1且请求中断。
2:0	-	<b>保留.</b> I2STA 的低三位常为0。

### I2DAT – I<sup>2</sup>C 数据寄存器

7	6	5	4	3	2	1	0
I2DAT[7:0]							
r/w							

地址: BCH

复位值: 0000 0000B

位	名称	描述
7:0	I2DAT[7:0]	<b>I<sup>2</sup>C 数据</b> I2DAT 包含一个字节的将被发送或刚接收到的I <sup>2</sup> C数据。只要SI为逻辑1, I2DAT中的数据保持不变, 在I <sup>2</sup> C发送接收过程中, 读或写I2DAT的结果都是不确定的。当I2DAT的数据被移出, 总线上的数据同步被移入以更新I2DAT。I2DAT常显示当前I <sup>2</sup> C总线上的最后字节。因此失去仲裁, 在传输之后的I2DAT原始值被改变。

### I2ADDR – I<sup>2</sup>C 设备自身从机地址

7	6	5	4	3	2	1	0
I2ADDR[7:1]							GC
r/w							r/w

地址: C1H

复位值: 0000 0000B

位	名称	描述
7:1	I2ADDR[7:1]	<b>I<sup>2</sup>C 设备自身从机地址</b> <u>主机模式:</u> 这些位没有影响。 <u>从机模式:</u> 这7位由用户定义I <sup>2</sup> C设备的从机地址。主机通过在起始信号或重复START后的第一个字节数据中发送相同的地址寻址该I <sup>2</sup> C设备, 如果AA标志置位, I <sup>2</sup> C设备将在接收到自身地址和成为被寻址的从机后应答主机。否则, 主机的寻址将被忽略。



位	名称	描述
0	GC	<p>广播呼叫位</p> <p>主机模式: 该位无影响。</p> <p>从机模式: 0 = 广播呼叫常被忽略。 1 = 如果AA标志为1, 广播呼叫被识别; 否则, 如果AA为0, 忽略广播呼叫。</p>

### I2CLK – I<sup>2</sup>C 时钟控制

7	6	5	4	3	2	1	0
I2CLK[7:0]							
r/w							

地址: BEH

复位值: 0000 1110B

位	名称	描述
7:0	I2CLK[7:0]	<p>I<sup>2</sup>C 时钟设置</p> <p>主机模式: 当设备为主机模式时, 该寄存器决定I<sup>2</sup>C总线的时钟, 该时钟依据下面公式。</p> $F_{I^2C} = \frac{F_{PHERI}}{1 + I2CLK}$ <p>如果使用带DIVM 1/4的24MHz的时钟系统, 默认值将使I<sup>2</sup>C总线时钟速率为400kbps 注I2CLK 的值为v 00H和01H 都无效, 这是极限值。</p> <p>从机模式: 该字节无效, 在从机模式下, I<sup>2</sup>C设备将自动同步任何给定的时钟速率, 最高可达400kps。</p>

## 16.4 工作模式

I<sup>2</sup>C协议定义四种操作模式包括主机发送, 主机接收, 从机接收和从机发送。还有一个特殊模式称之为广播呼叫, 其操作相似于主机传输模式。

### 16.4.1 主机发送模式

在主机发送模式下, 向从机接收器发送几个数据字节。主机通过I2CLK设置期望时钟速率并向I2CEN (I2CON.6)写1使能I2C总线, 设置STA (I2CON.5)为1进入主机发送模式, 只要总线空闲, 硬件将测试总线并产生起始信号, 成功产生起始信号后, SI标志 (I2CON.3)将置位且I2STA的状态码为08H, 之后就是给I2DAT载入目标从机地址和数据方向位“写” (SLA+W), SLA+W 开始传输时SI 位必须清零。

在SLA+W 字节被发送且由被寻址的从器件返回应答(ACK)后, SI标志再次置位且I2STA读数为18H, 依据用户定义的通信协议持续发送数据, 全部数据发送完成后, 主机可以通过设置STO (I2CON.4)发送停止信号并清SI中止传输, 在没有发送停止信号下重复起始信号将立即开始另一次传输。

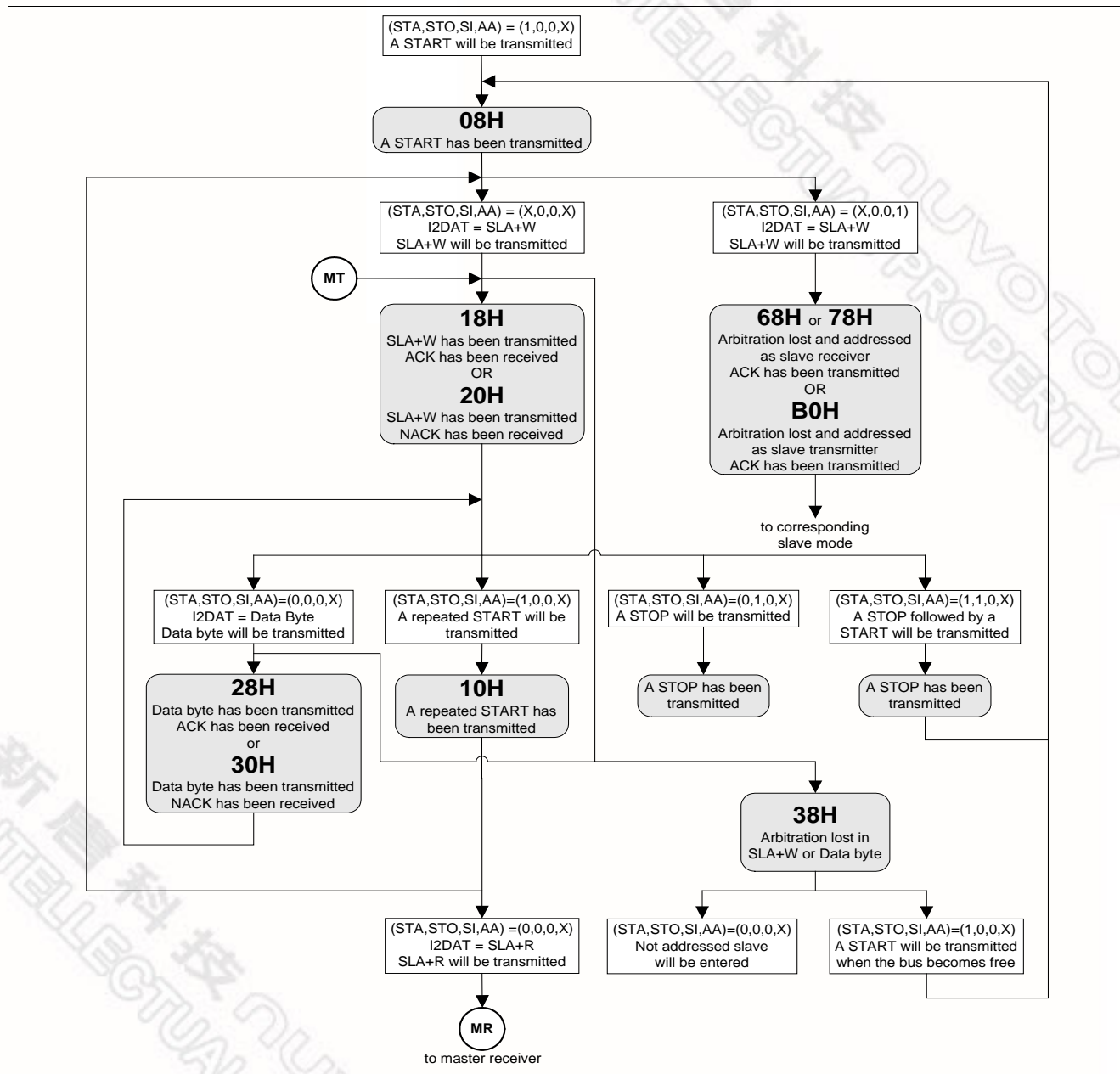


图16-7. 主机发送模式流程与状态

## 16.4.2 主机接收模式

在主机接收模式下，从从机发送器接收几个字节的数据。传输开始与主机发送模式相似，在起始信号之后，I2DAT应该加载目标从机地址和数据方向位“读”(SLA+R)，SLA+R 字节发送后，且返回应答位，重新置位 SI标志且I2STA读出为40H，SI标志应该被清零以便接收从机发送过来的数据，如果AA 标志 (I2CON.3) 置位，主机接收器将应答从机发送器，如果清零AA，主机接收器将不会应答从机，并释放从机发送器为不被寻址的从机，然后，主机产生停止信号或重复起始信号中止传输或开始另一次传输。

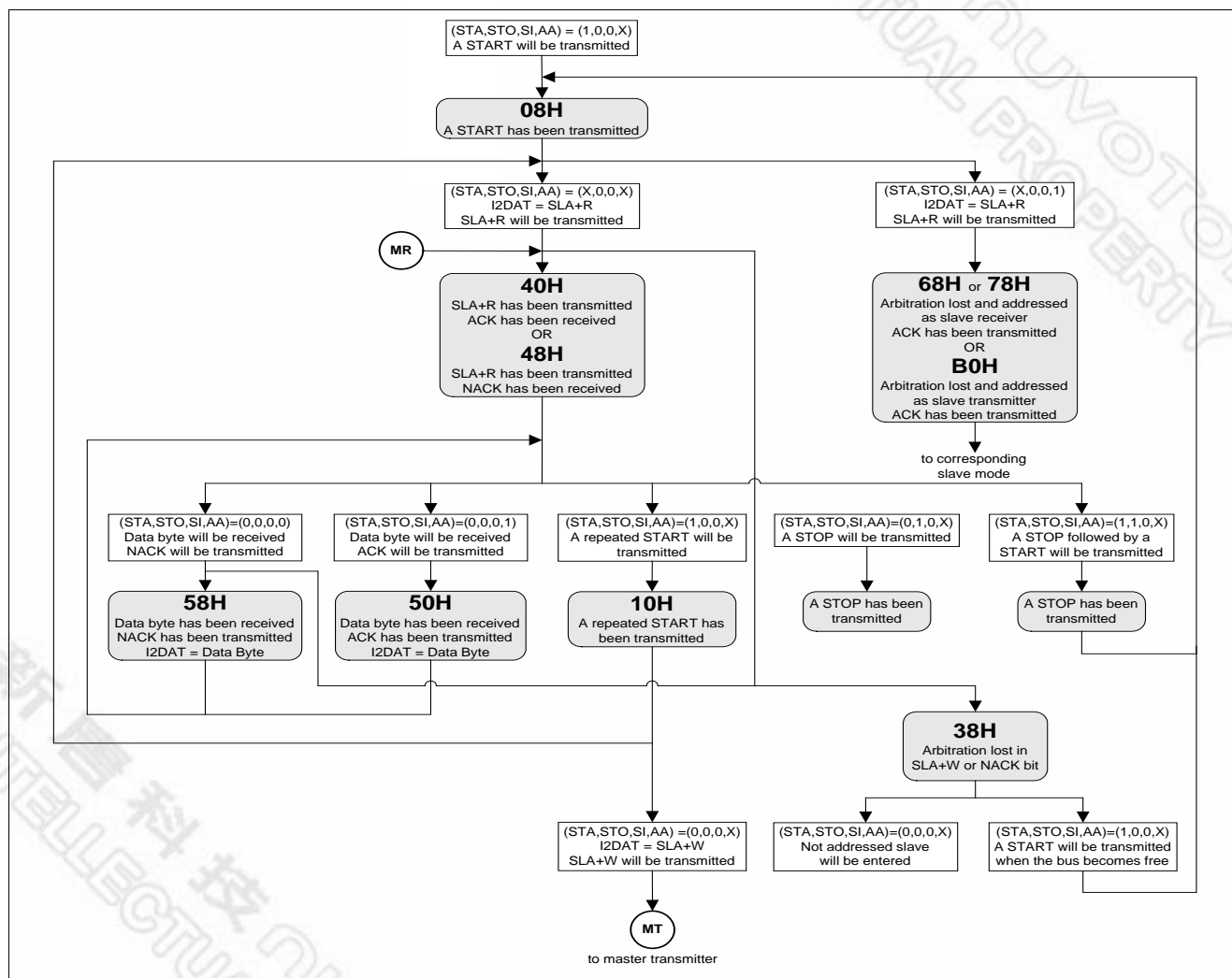


图16-8. 主机接收模式流程与状态

### 16.4.3 从机接收模式

在从机接收模式下，从主机发送器接收几个字节数据。发送开始之前，I2ADDR必须装载响应器件的地址，以让主机寻址，从机模式下I2CLK 无效，AA位必须设置使能应答自身从机地址或广播呼叫，完成以上初始过程后，I<sup>2</sup>C等待自身地址被寻址与数据方向位“写”(SLA+W)或被广播呼叫寻址。如果在仲裁失败时，也可以进入从机接收模式。

在从机被SLA+W寻址后，应该清SI标志以便接收主机发送过来的数据，传输期间，如果AA 位为0，从机将在下一次接收到的数据字节之后返回无应答（non-acknowledge），从机也不被寻址并与主机分离，不能接收I2DAT的任何字节，而保持当前接收到的数据字节。

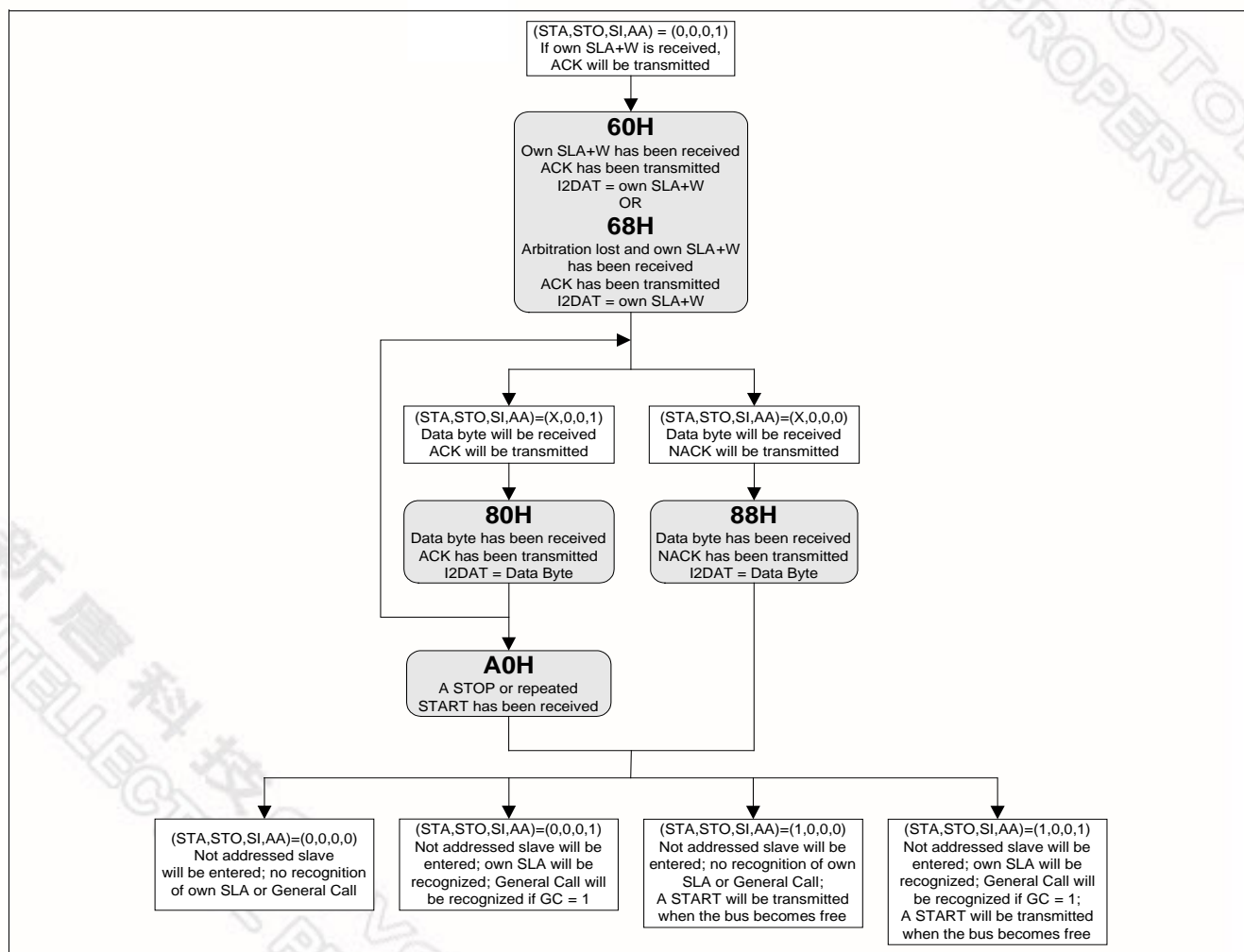


图16-9. 从机接收模式流程与状态

#### 16.4.4 从机发送模式

在从机发送模式下，发送几个字节数据到主机接收器。确定I2ADDR 和 I2CON 的值之后，I<sup>2</sup>C等待自己的地址被寻址“读”(SLA+R)。如果仲裁失败后，也可以进入从机发送模式。

在从机被SLA+W寻址后，应该清SI标志以便传输数据到主机发送器，通常主机接收器将在从机发送每个字节数据之后返回应答，如果没有接收到应答，如果继续传输将发送全“1”，就成为不被寻址的从机，如果在传输中清了AA标志，从机发送最后一个字节数据，下一次传输数据全为“1”，从机成为不被寻址。

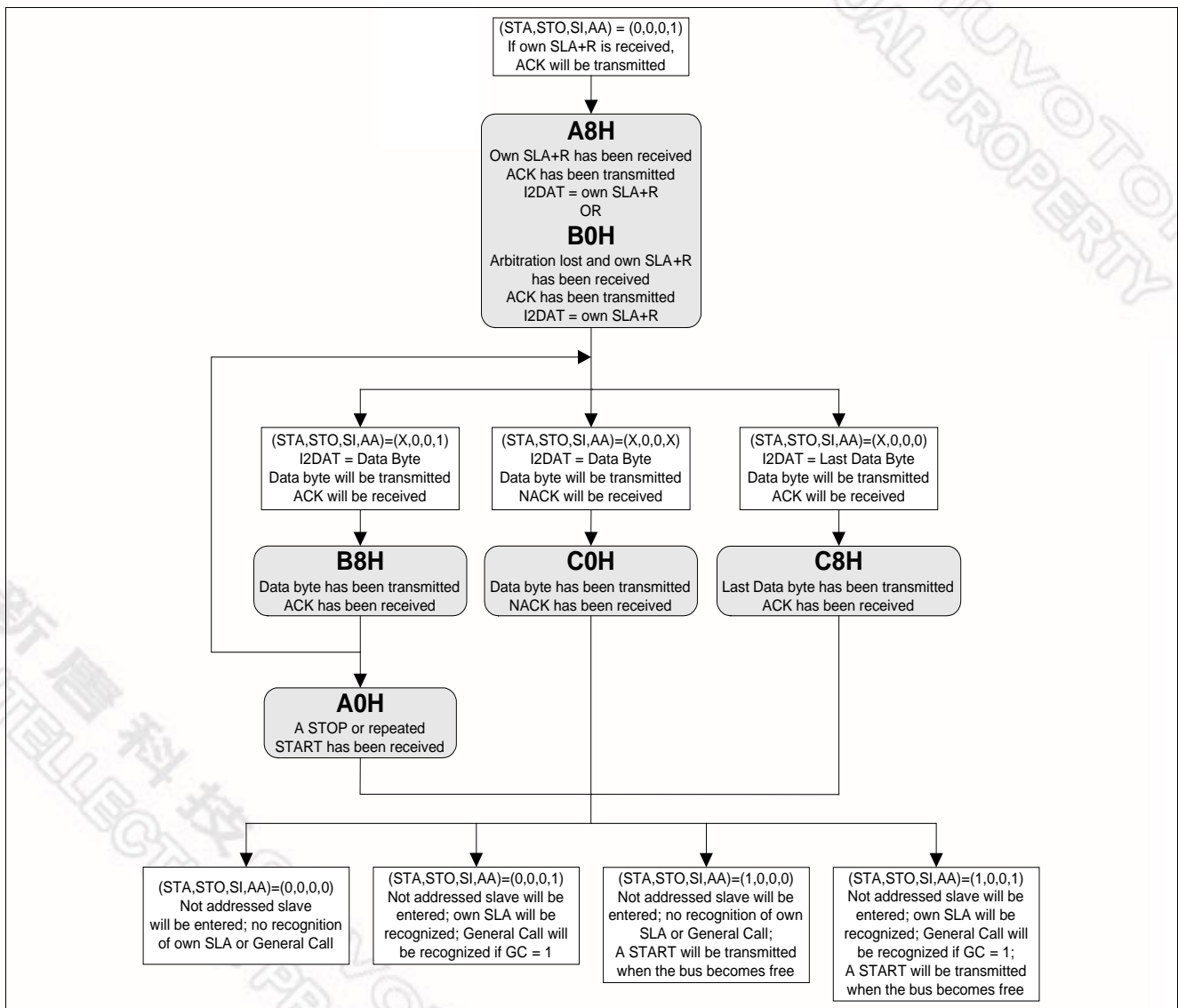


图16-10. 从机发送模式流程与状态

### 16.4.5 广播呼叫

广播呼叫是从机接收模式的一种特殊情况，即从机地址和数据方向位全为0，被广播呼叫寻址的从机在正常从机接收模式的I2STA里有不同状态码，如果仲裁失败，也可以产生广播呼叫。

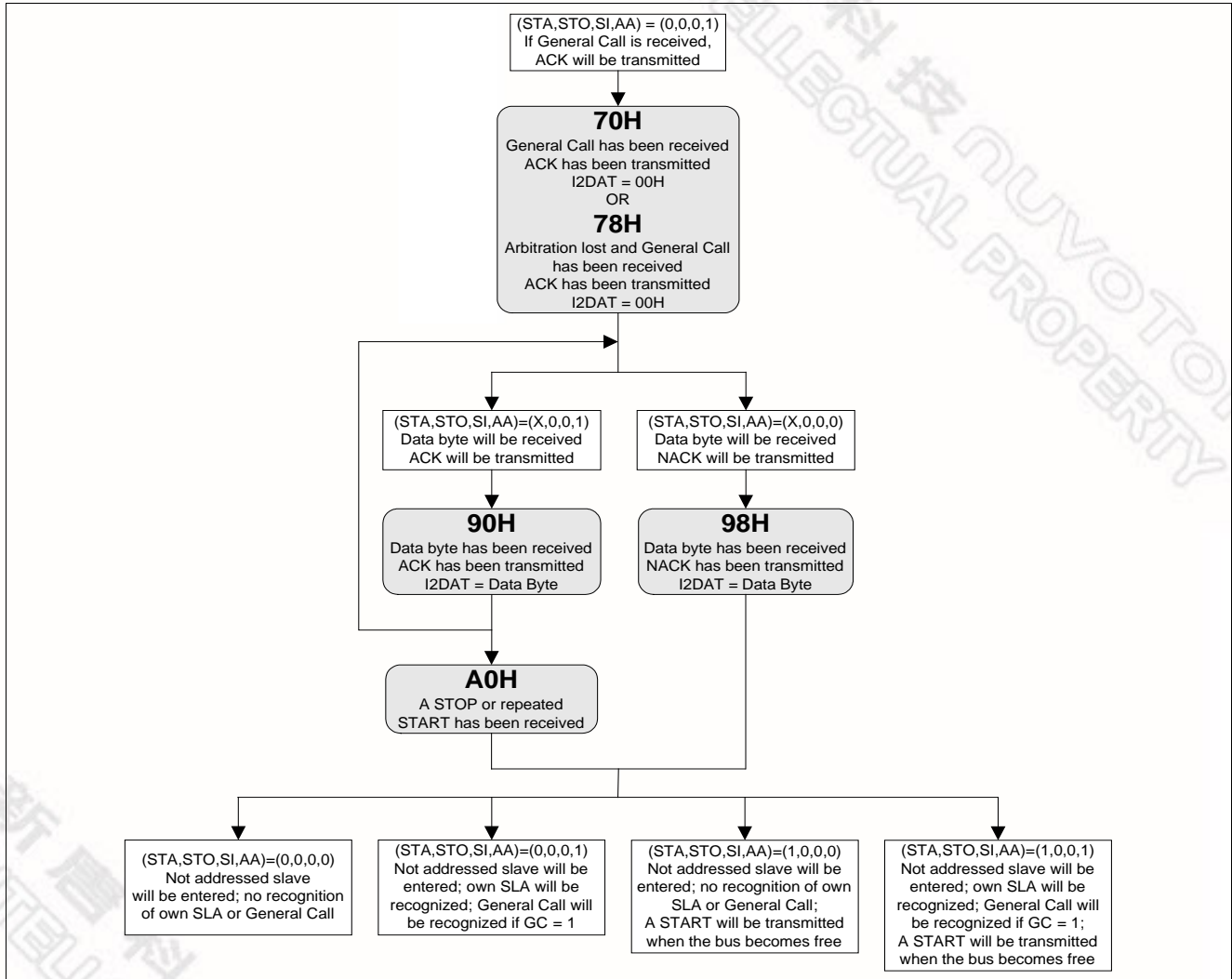


图16-11. 广播呼叫模式流程与状态

### 16.4.6 其它状态

有两个I2STA状态码与24个定义状态不一致，即前面提到的F8H和00H状态。

第一个状态码F8H表示在每次传输期间没有得到相关信息，同时，SI标志为0且没有I<sup>2</sup>C中断请求。

另一个标志码00H意味在传输过程中发生错误，总线错误是由START或停止信号暂时出现在一个非法的位置，如地址字节里第2位换到第8位，或数据字节包括应答位，当出现总线错误时，SI标志立即置位，当在I<sup>2</sup>C总线上检测到总线错误，工作器件立即切换到不被寻址从机模式，释放SDA和SCL总，置位SI标志，将00H载入I2STA。要从总线错误恢复，STO位必须设置为逻辑1且SI必须清零，然后，STO由硬件清零且在停止信号就释放I<sup>2</sup>C总线。

特例：如果没有成功产生START或重复起始信号，I<sup>2</sup>C总线被SDA的低电平阻挡，如一个从机器件没有位同步，可以通过在SCL总线上发送额外时钟脉冲解决这个问题。当STA位置位时，I<sup>2</sup>C硬件发送额外时钟脉冲，但是由于SDA被拉低，不能产生起始信号，当SDA总线最终被释放，发送一个普通的START条件，进入状态08H，继续进行串行传输。当SDA为低，如果发送重复起始信号，I<sup>2</sup>C硬件也执行以上相同的动作。此情况下，在成功发送起始信号后，进入状态08H，而不是进入10H。注软件不能解决这类总线问题。

## 16.5 典型I<sup>2</sup>C结构的中断服务程序

下面例程可以在KEIL C51编译器下运行，是比较典型的I<sup>2</sup>C中断服务程序，包括有26种状态服务子程序，可以满足用户的基本应用。用户可以直接或稍作修改后应用于自己的系统中。

```

Void I2C_ISR (void) interrupt 6
{
  switch (I2STA)
  {
    //=====
    //Bus Error, always put in ISR for noise handling
    //=====
    case 0x00: /*00H, bus error occurs*/
      STO = 1; /*recover from bus error
      break;
    //=====
    //Master Mode
    //=====
    case 0x08: /*08H, a START transmitted*/
      STA = 0; /*STA bit should be cleared by software
      I2DAT = SLA_ADDR1; /*LOAD SLA+W/R
      break;
    case 0x10: /*10H, a repeated START transmitted*/
      STA = 0;
  }

```



```

I2DAT = SLA_ADDR2;
break;
//=====
//Master Transmitter Mode
//=====
case 0x18:          /*18H, SLA+W transmitted, ACK received*/
    I2DAT = NEXT_SEND_DATA1; //LOAD DATA
    break;
case 0x20:          /*20H, SLA+W transmitted, NACK received*/
    STO = 1;        //transmit STOP
    AA = 1;         //ready for ACK own SLA+W/R
    break;
case 0x28:          /*28H, DATA transmitted, ACK received*/
    if (Conti_TX_Data) //if continuing to send DATA
        I2DAT = NEXT_SEND_DATA2;
    else //if no DATA to be sent
    {
        STO = 1;
        AA = 1;
    }
    break;
case 0x30:          /*30H, DATA transmitted, NACK received*/
    STO = 1;
    AA = 1;
    break;
//=====
//Master Mode
//=====
case 0x38:          /*38H, arbitration lost*/
    STA = 1;        //retry to transmit START if bus free
    break;
//=====
//Master Receiver Mode
//=====
case 0x40:          /*40H, SLA+R transmitted, ACK received*/
    AA = 1;         //ACK next received DATA
    break;
case 0x48:          /*48H, SLA+R transmitted, NACK received*/
    STO = 1;
    AA = 1;
    break;
case 0x50:          /*50H, DATA received, ACK transmitted*/
    DATA_RECEIVED1 = I2DAT; //store received DATA
    if (To_RX_Last_Data1) //if last DATA will be received
        AA = 0;         //not ACK next received DATA
    else //if continuing receiving DATA
        AA = 1;
    break;
case 0x58:          /*58H, DATA received, NACK transmitted*/
    DATA_RECEIVED_LAST1 = I2DAT;
    STO = 1;
    AA = 1;
    break;
//=====
//Slave Receiver and General Call Mode
//=====
case 0x60:          /*60H, own SLA+W received, ACK returned*/

```





```

AA = 1;
break;
case 0x68:          /*68H, arbitration lost in SLA+W/R
                   own SLA+W received, ACK returned */
    AA = 0;        //not ACK next received DATA after
                   //arbitration lost
    STA = 1;      //retry to transmit START if bus free
    break;
case 0x70:          //70H, General Call received, ACK returned
    AA = 1;
    break;
case 0x78:          /*78H, arbitration lost in SLA+W/R
                   General Call received, ACK returned*/
    AA = 0;
    STA = 1;
    break;
case 0x80:          /*80H, previous own SLA+W, DATA received,
                   CK returned*/
    DATA_RECEIVED2 = I2DAT;
    if (To_RX_Last_Data2)
        AA = 0;
    else
        AA = 1;
    break;
case 0x88:          /*88H, previous own SLA+W, DATA received,
                   NACK returned, not addressed SLAVE mode
                   entered*/
    DATA_RECEIVED_LAST2 = I2DAT;
    AA = 1;        //wait for ACK next Master addressing
    break;
case 0x90:          /*90H, previous General Call, DATA received,
                   ACK returned*/
    DATA_RECEIVED3 = I2DAT;
    if (To_RX_Last_Data3)
        AA = 0;
    else
        AA = 1;
    break;
case 0x98:          /*98H, previous General Call, DATA received,
                   NACK returned, not addressed SLAVE mode
                   entered*/
    DATA_RECEIVED_LAST3 = I2DAT;
    AA = 1;
    break;
//=====
//Slave Mode
//=====
case 0xA0:          /*A0H, STOP or repeated START received while
                   still addressed SLAVE mode*/
    AA = 1;
    break;
//=====
//Slave Transmitter Mode
//=====
case 0xA8:          /*A8H, own SLA+R received, ACK returned*/
    I2DAT = NEXT_SEND_DATA3;

```



```

    AA = 1;                //when AA is "1", not last data to be
                          //transmitted
    break;
case 0Xb0:                /*B0H, arbitration lost in SLA+W/R
                          own SLA+R received, ACK returned */
    I2DAT = DUMMY_DATA;
    AA = 0;                //when AA is "0", last data to be
                          //transmitted
    STA = 1;              //retry to transmit START if bus free
    break;
case 0Xb8:                /*B8H, previous own SLA+R, DATA transmitted,
                          ACK received*/
    I2DAT = NEXT_SEND_DATA4;
    if (To_TX_Last_Data) //if last DATA will be transmitted
        AA = 0;
    else
        AA = 1;
    break;
case 0Xc0:                /*C0H, previous own SLA+R, DATA transmitted,
                          NACK received, not addressed SLAVE mode
                          entered*/
    AA = 1;
    break;
case 0Xc8:                /*C8H, previous own SLA+R, last DATA trans-
                          mitted, ACK received, not addressed SLAVE
                          mode entered*/
    AA = 1;
    break;
} //end of switch (I2STA)

SI = 0;                  //SI should be the last step of I2C ISR
while(STO);              //wait for STOP transmitted or bus error
                          //free, STO is cleared by hardware
} //end of I2C_ISR

```

## 16.6 I<sup>2</sup>C 超时

有一个14-位的超时计数器，可用于处理I<sup>2</sup>C总线挂起，如果使能超时计数器，计数器开始计数直到溢出，同时TIF 由硬件置位并请求I<sup>2</sup>C中断，当使能超时计数器时，设置标志SI为高将复位计数器并在SI清零后重启计数，如果I<sup>2</sup>C总线挂起，会导致SI 标志在一段时期内不被置位，14-位超时计数器将溢出且请求中断服务。

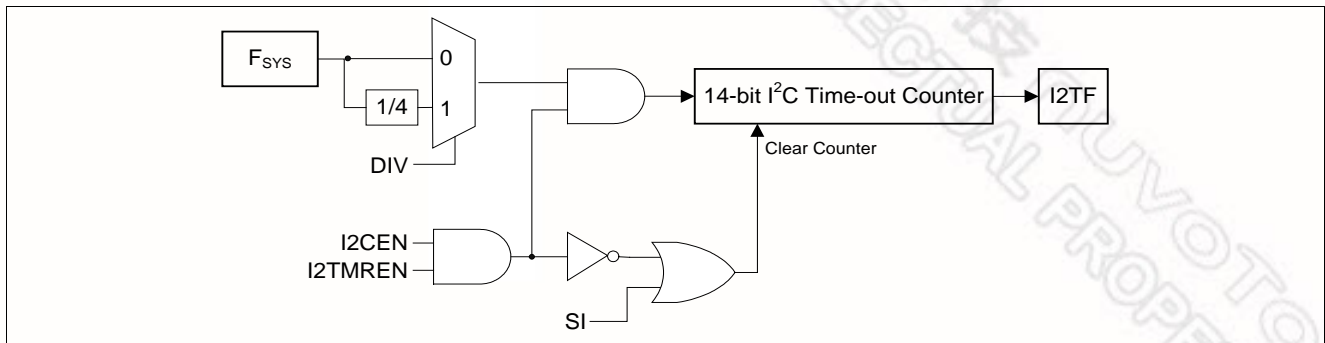


图16-12. I<sup>2</sup>C 超时计数器

### I2TOC – I<sup>2</sup>C 超时计数器

7	6	5	4	3	2	1	0
-	-	-	-	-	I2TOCEN	DIV	I2TOF
-	-	-	-	-	r/w	r/w	r/w

地址：BFH

复位值：0000 000B

位	名称	描述
7:3	-	保留。
2	I2TOCEN	I <sup>2</sup> C 超时计数器使能 0 = 禁止I <sup>2</sup> C 超时计数器 1 = 使能I <sup>2</sup> C 超时计数器
1	DIV	I <sup>2</sup> C 超时计数器时钟分频 0 = I <sup>2</sup> C 超时计数器的分频为1/1 F <sub>sys</sub> . 1 = I <sup>2</sup> C 超时计数器的分频为1/4 F <sub>sys</sub> .
0	I2TOF	I <sup>2</sup> C 超时计数器溢出标志 如果I <sup>2</sup> C的14位超时计数器溢出，硬件将I2TOF标志置位。软件清I2TOF标志。

## 16.7 I<sup>2</sup>C 中断

有两个I<sup>2</sup>C标志，SI 和 I2TOF，两者均可以产生I<sup>2</sup>C中断请求。如果设置EI2C (EIE.0) 和EA为1使能I<sup>2</sup>C 中断屏蔽后，一旦两个标志中任何一个置位，CPU将执行I2C中断服务程序，用户需要检查标志位以确定发生的是哪个中断事件，两个I<sup>2</sup>C 标志由软件清零。

## 17 脉冲宽度调制(PWM)

### 17.1 特征

PWM (脉冲宽度调制) 信号有很广泛的应用, 可以用于马达驱动, 风扇控制, 背光调制, LED 灯光, 或通过低通滤波器模拟数模转换输出, N79E815A/814A/8132A提供四个通道, 最大10位的PWM输出。

### 17.2 功能描述

N79E815A/814A/8132A包括4个脉冲宽度调制(PWM)通道, 可以产生长度和间隔可调的脉冲。PWM0在P0.1上输出, PWM1 在 P1.6上输出, PWM2 在 P1.7上输出, PWM3在P0.0上输出。芯片复位后, 每个PWM通道的输出都为“1”, 此情况下, 在引脚反映内部PWM输出状态之前, 必须向每个端口位上写1以作为PWM输出, 如图17-1所示, 连续输出的间隔由10位向下计数器控制, 用户可以配置内部时钟预分频作为10位向下计数器的输入, PWM计数器时钟频率 $F_{PWM} = F_{SYS}/Prescaler$ , 当计数器达到向下溢出时, 将重新加载用户所选的值, 该机制允许用户设置PWM频率, PWM 的频率参考以下公式:

$$\text{PWM 输出频率} = \frac{F_{SYS}}{1+PWMP}, \text{ PWM 有效电平占空比} = \frac{PWMn}{1+PWMP}.$$

PWMP包括PWMPH 和 PWMP, 如下所述:

#### PWMP, PWMPL – PWM计数器寄存器低字节

7	6	5	4	3	2	1	0
PWMP.7	PWMP.6	PWMP.5	PWMP.4	PWMP.3	PWMP.2	PWMP.1	PWMP.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: D9H

复位值: 0000 000B

位	名称	描述
7:0	PWMP, PWMPL	PWM 计数器寄存器位 [7:0]。

#### PWMPH – PWM 计数器寄存器高字节

7	6	5	4	3	2	1	0
-	-	-	-	-	-	PWMP.9	PWMP.8
-	-	-	-	-	-	r/w	r/w

地址: D1H

复位值: 0000 000B

位	名称	描述
7:2	-	保留。
1:0	PWMPH	PWM 计数器寄存器位[9:8]。

用户根据以下初始化步骤开始产生PWM信号输出，第一步，设置CLRPWM (PWMCON0.4)，确定10位向下计数器的值，在设置全部周期和占空比寄存器后，可设置PWMRUN (PWMCON0.7)为1，触发10位向下计数运行，刚开始PWM输出保持为高，直到计数器的值小于占空比寄存器PWMnH 和 PWMnL 的值，此时PWM输出为低，直到下一次向下计数溢出。当10位向下计数器向下溢出时，PWMP缓冲寄存器重新载入10位向下计数器，继续输出PWM信号，并重复这个过程。

所有周期和占空比控制寄存器的硬件都是双缓冲设计，因此任何时候都可以写寄存器PWMP 和PWMn，但是PWM的周期和占空比将不能立即更新，直到置位Load (PWMCON0.6)且前一周期完成，这允许更新PWM周期和占空比，而无毛刺产生。

#### PWM0L – PWM 0 低位寄存器

7	6	5	4	3	2	1	0
PWM0.7	PWM0.6	PWM0.5	PWM0.4	PWM0.3	PWM0.2	PWM0.1	PWM0.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: DAH

复位值: 0000 0000B

位	名称	描述
7:0	PWM0L	PWM0 低位寄存器位 [7:0]。

#### PWM0H – PWM 0 高位寄存器

7	6	5	4	3	2	1	0
-	-	-	-	-	-	PWM0.9	PWM0.8
-	-	-	-	-	-	r/w	r/w

地址: D2H

复位值: 0000 0000B

位	名称	描述
7:2	-	保留。
1:0	PWM0H	PWM0 高位寄存器位 [9:8]。

#### PWM1L– PWM 1低位寄存器

7	6	5	4	3	2	1	0
PWM1.7	PWM1.6	PWM1.5	PWM1.4	PWM1.3	PWM1.2	PWM1.1	PWM1.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: DBH

复位值: 0000 0000B

位	名称	描述
7:0	PWM1L	PWM1低位寄存器位[7:0]。

#### PWM1H – PWM 1 高位寄存器

7	6	5	4	3	2	1	0
-	-	-	-	-	-	PWM1.9	PWM1.8



-	-	-	-	-	-	r/w	r/w
---	---	---	---	---	---	-----	-----

地址: D3H 复位值: 0000 0000B

位	名称	描述
7:2	-	保留.
1:0	PWM1H	PWM1 高位寄存器位[9:8].

**PWM2L– PWM 2 低位寄存器**

7	6	5	4	3	2	1	0
PWM2.7	PWM2.6	PWM2.5	PWM2.4	PWM2.3	PWM2.2	PWM2.1	PWM2.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: DDH 复位值: 0000 0000B

位	名称	描述
7:0	PWM2L	PWM 2 低位寄存器位 [7:0].

**PWM2H – PWM 2 高位寄存器**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	PWM2.9	PWM2.8
-	-	-	-	-	-	r/w	r/w

地址: D5H 复位值: 0000 0000B

位	名称	描述
7:2	-	保留.
1:0	PWM2H	PWM 2 高位寄存器位[9:8].

**PWM3L– PWM 3 低位寄存器**

7	6	5	4	3	2	1	0
PWM3.7	PWM3.6	PWM3.5	PWM3.4	PWM3.3	PWM3.2	PWM3.1	PWM3.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: DEH 复位值: 0000 0000B

位	名称	描述
7:0	PWM3L	PWM 3低位寄存器位[7:0].

**PWM3H – PWM 3高位寄存器**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	PWM3.9	PWM3.8
-	-	-	-	-	-	r/w	r/w

地址: D6H 复位值: 0000 0000B

位	名称	描述
7:2	-	保留.
1:0	PWM3H	PWM 3高位寄存器位[9:8].

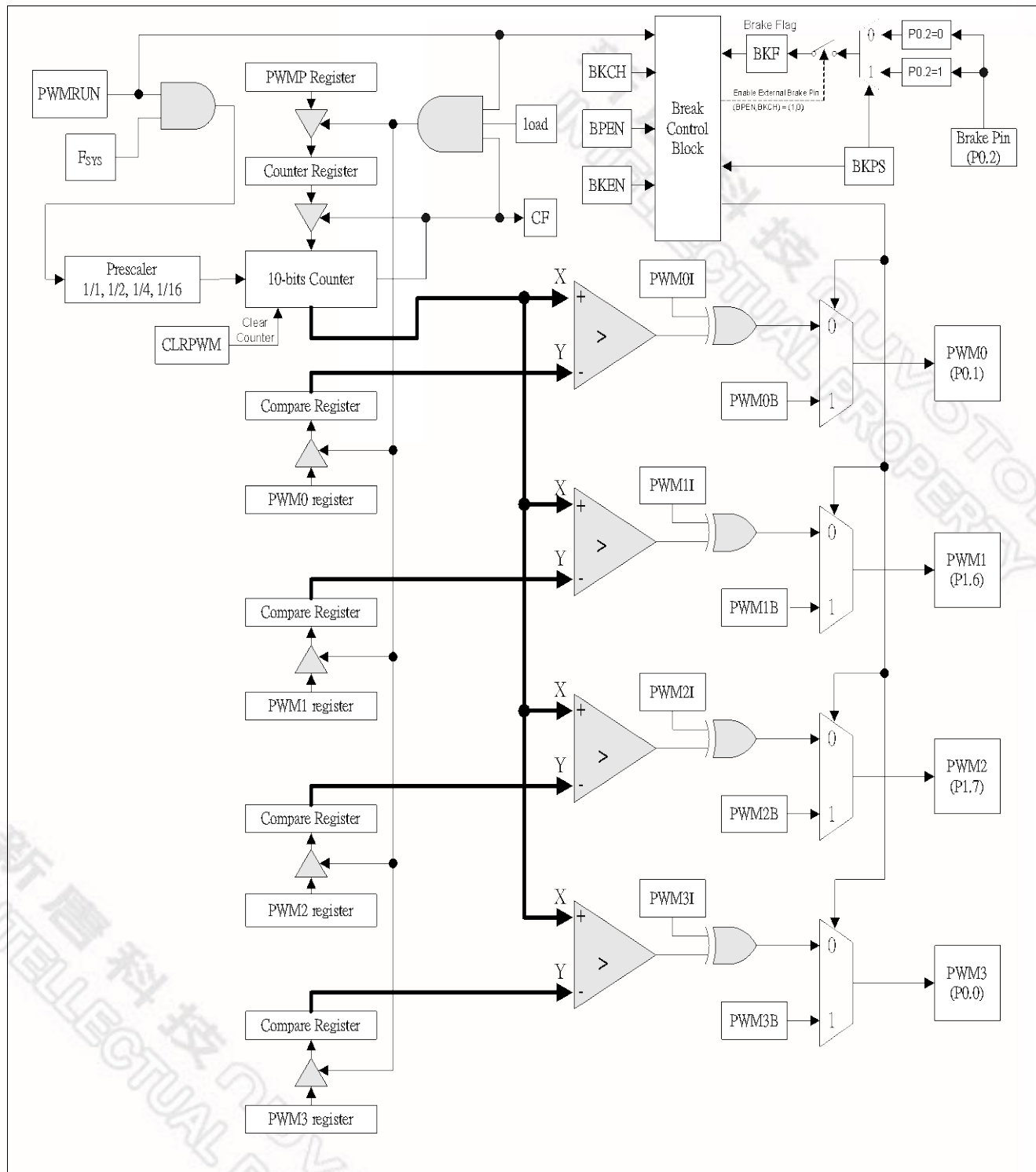


图17-1 PWM 模块框图

比较值大于PWM的计数器加载值，输出恒为低。有两个特例，比较值全为0，000H，导致输出恒为高，比较值全为1，3FFH，导致PWM输出恒为低，比较器的值再次载入比较寄存器，在程序的控制下，该寄存器传输到实际比较寄存器里，寄存器分配如下，“PWMn”的数值定义了PWM的输出，因此，PWM0 控制PWM0的宽度，PWM1 控制PWM1的宽度，以此类推。

PWM模块的功能由寄存器PWMCON0控制，这些控制位都很简单。例如，每个输出都有一个反转位，可以产生一个反相位的输出，与反转前的输出相比较。当 PWMCON0.7 (PWMRUN) 允许PWM运行或空闲时，从计数器和比较器向控制寄存器的数据传输由PWMCON0.6 (load) 控制。用户可以通过PWCON1.6(Load) 与PWMCON0.5(CF flag)位监测溢出，支持PWM中断使能和EA=1，无论CF标志是否置位都不发生PWM中断，支持PWM中断使能和EA=1，PWM中断仅发生在当PWM 钳位标志置位时，当传输发生时，PWM逻辑在下一周期自动复位这些位。

通过设置LOAD加载一个新的周期和占空比，此时应该通过监测它并等待硬件自己清LOAD位确保完成，在LOAD位为1的期间任何更新PWM控制寄存器将引起不可预见的输出。

#### PWMCON0 – PWM 控制寄存器 0

7	6	5	4	3	2	1	0
PWMRUN	Load	CF	CLRPWM	PWM3I	PWM2I	PWM1I	PWM0I
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址： DCH

复位值： 0000 0000B

位	名称	描述
7	PWMRUN	0: PWM 没有运行。 1: PWM 计数器在运行。
6	Load	0: PWMP和比较的值不加载到计数器和比较器寄存器中。 1: 在计数器溢出后，PWMP的值加载到计数器寄存器，且硬件将被下一个时钟周期清零。
5	CF	10-位计数器溢出标志： 0: 10-位计数器没有向下计数溢出。 1: 10-位计数器向下计数溢出。
4	CLRPWM	1: 清10-位PWM计数器到000H。
3	PWM3I	0: PWM3 输出不反转。 1: PWM3 输出反转。
2	PWM2I	0: PWM2 输出不反转。 1: PWM2 输出反转。
1	PWM1I	0: PWM1输出不反转。 1: PWM1输出反转。
0	PWM0I	0: PWM0输出不反转。 1: PWM0输出反转。



事实上，从计数器和PWMn寄存器传输到工作寄存器(10位计数器和比较寄存器)仅发生在计数结果满足用户程序设定溢出，如果PWMCON0写入Load设置而没有使能运行，则不会传输。如果使能运行而没有设置Load，则比较器和计数器的值不是所期望的值。如果load与运行都设置了，则优先于溢出的是加载PWMCON0，设置运行而非load，则Load不会发生。

如上图所示，Load位可以决定load发生时间。推荐在写PWMCON0时设置PWMRUN（PWMCON0.7）与load(PWMCON0.6)。

当PWMCON0.7（PWMRUN）位被清零，PWM输出的状态为刚清零时的值，通常该状态是不确定的，当PWMRUN清零时为使输出处于确定的状态，比较寄存器可以写入“恒1”或“恒0”，以使在计数器停止时输出为期望状态。在PWMCON0应该写入Load和运行位使能后，PWMCON0有Load或CF标志产生。一旦Load发生，PWMCON0的运行位可以被清零，输出保持在清运行位时的状态。如果钳位引脚（见下面关于PWMCON1的操作）不是用于控制钳位功能，“钳位不运行”功能可以用于当PWM停止时将输出保持在确定状态。这种方法仅用于时间紧急的情形。关于这点的探论可参考PWMCON1的描述。

#### PWMCON1 – PWM 控制寄存器 1

7	6	5	4	3	2	1	0
BKCH	BKPS	BPEN	BKEN	PWM3B	PWM2B	PWM1B	PWM0B
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址：DFH

复位值：0000 0000B

位	名称	描述
7	BKCH	当BKEN置位时，见下表。
6	BKPS	0: 如果P0.2为低，钳位。 1: 如果P0.2为高，钳位。
5	BPEN	当BKEN置位时，见下表。
4	BKEN	0: 不发生钳位。 1: 使能钳位，见下表。
3	PWM3B	0: 钳位时PWM3 输出为低。 1: 钳位时PWM3 输出为高。
2	PWM2B	0: 钳位时PWM2 输出为低。 1: 钳位时PWM2 输出为高。
1	PWM1B	0: 钳位时PWM1 输出为低。 1: 钳位时PWM1 输出为高。
0	PWM0B	0: 钳位时PWM0 输出为低。 1: 钳位时PWM0 输出为高。

### 钳位发生条件表

BPEN	BKCH	钳位条件
0	0	打开钳位(软件钳位和保持钳位)
0	1	打开, 当PWM不运行时 (PWMRUN=0), PWM的输出依据PWMNB设置。 关闭, 当PWM运行时 (PWMRUN=1)。
1	0	打开钳位, 当声明为钳位引脚, 没有PWM输出, PWMRUN将被清零, 且BKF标志置位。PWM输出条件依据PWMNB设置。
1	1	无效。

### PWMCON2 – PWM 控制寄存器 2

7	6	5	4	3	2	1	0
-	-	-	-	FP1	FP0	-	BKF
-	-	-	-	r/w	r/w	-	r/w

地址: D7H

复位值: 0000 0000B

位	名称	描述										
7:4	-	保留.										
3:2	FP[1:0]	PWM频率预分频选择位。预分频的时钟源, 如果PWMRUN=1, Fpwm与F <sub>sys</sub> 相位一致。 <table border="1" data-bbox="483 1136 802 1356"> <thead> <tr> <th>FP[1:0]</th> <th>Fpwm</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>F<sub>sys</sub> (默认值)</td> </tr> <tr> <td>01</td> <td>F<sub>sys</sub>/2</td> </tr> <tr> <td>10</td> <td>F<sub>sys</sub>/4</td> </tr> <tr> <td>11</td> <td>F<sub>sys</sub>/16</td> </tr> </tbody> </table>	FP[1:0]	Fpwm	00	F <sub>sys</sub> (默认值)	01	F <sub>sys</sub> /2	10	F <sub>sys</sub> /4	11	F <sub>sys</sub> /16
FP[1:0]	Fpwm											
00	F <sub>sys</sub> (默认值)											
01	F <sub>sys</sub> /2											
10	F <sub>sys</sub> /4											
11	F <sub>sys</sub> /16											
1	-	保留.										
0	BKF	<b>外部钳位引脚标志</b> 0: PWM没有钳位。 1: PWM由外部钳位引脚钳位, 可由软件清零。										

钳位功能由寄存器PWMCON1的内容控制, 通常当钳位发生时, 4个PWM的输出钳位在用户选择的状态, 也就是由PWMCON1 位0到3选择的状态, 如寄存器PWMCON1 操作描述, 如果PWMCON1.4为“1”, 在PWMCON1.7, BKCH, 和 PWMCON1.5, BPEN控制下声明钳位, 如果都为“0”, 声明钳位, 如果PWMCON1.7 为“1”当PWMCON0.7为“0”声明 brake, 如果PWMCON1.6 为“1”, 当钳位引脚P0.2, 与PWMCON1.6有相同极性, 声明钳位, 当对该引脚声明钳位时, PWMCON0.7自动被清零且 BKF(PWMCON2.0)标志置位, 不允许PWMCON1.7 和 PWMCON1.5都为“1”。

一旦钳位引脚被声明，将自动清零PWMCON0.7而且标志位 BKF(PWMCON2.0) 被置位，用户程序查询该位或使能PWM钳位中断以决定何时钳位引脚发生钳位，另一个检查钳位的方法是将钳位引脚接到外部中断引脚上。后一种方法是在钳位信号不够长而难觅被捕获的情况下所需要。声明后，当引起钳位的条件被移除后，PWM输出为钳位时的状态，这意味着从钳位声明返回运行PWM而不需要经历不确定的状态。如果钳位引脚引起钳位被声明下面协议代码，将通过软件查询BKF标志或使能PWM中断，允许PWM从钳位平稳返回运行。

注，如果在钳位引脚有狭小的脉冲导致声明钳位，则在脉冲结束之前不可能通过上述代码，此情况下，除所示代码外，也可能被请求钳位引脚上的外部锁定以确保能从钳位平稳返回运行。

#### 例程

```

ORG      0H
SJMP    START
ORG 100H
START:
MOV     PWMPH,#0           ; PWM Frequency = Fsys/(1+PWMP)
MOV     PWMPL,#0FFH       ; If Fsys=20MHz, PWM Frequency=78.1kHz
MOV     PWM0H,#0
MOV     PWM0L,#080H       ; PWM0(P0.1) duty = PWM0/(1+PWMP)
MOV     PWM1H,#0
MOV     PWM1L,#0A0H       ; PWM1(P1.6) duty = PWM1/(1+PWMP)
MOV     PWM2H,#0
MOV     PWM2L,#0C0H       ; PWM2(P1.7) duty = PWM2/(1+PWMP)
MOV     PWM3H,#0
MOV     PWM3L,#0F0H       ; PWM3(P0.0) duty = PWM3/(1+PWMP)

ORL     PWMCON0,#0D0H     ; Start PWM

MOV     PWMCON1,#30H      ; PWM will be stopped when P0.2 is low level.
                          ; PWM output condition is follow PWMNB setting.
                          ; In this case, PWM0B=PWM1B=PWM2B=PWM3B=0

END

```

## 18 时控保护

N79E815A/814A/8132A有几个特殊功能，如看门狗定时器，ISP功能等。引导选择控制等关键系统的正常运行。如果没有对这些寄存器进行写保护，无关代码可能对其写入不确定的值，结果在不正确的操作和控制损失。为了防止这种风险，N79E815A/814A/8132A的时控保护功能，可限制了关键的SFR的写访问。此保障方式，是用一个定时器控制的访问。以下寄存器相关的时控保护过程。

### TA – 时控保护开启寄存器

7	6	5	4	3	2	1	0
TA[7:0]							
W							

地址：C7H

复位值：1111 1111B

位	名称	描述
7:0	TA[7:0]	<p><b>时控保护寄存器</b></p> <p>用于定时对SFR读写，保护SFR寄存器内的值。当需要写特殊规定的寄存器时，必须先对TA寄存器写入AAH，接着是55H，当写完这两条后，才可以对具有时控保护的寄存器内写入数据。</p>

对被保护的位的访问是受时间限制的。要对他进行写操作，那么时控窗口必须打开，否则写操作无效。当条件满足时，时控窗口开放3个机器周期。在3个机器周期过后，时控窗口自动关闭。要打开时控窗口，必须先向TA寄存器写入AAH，再写入55H。下面列出对时控寄存器进行访问的推荐代码。当软件向TA写入Aah后，计数器开始计数，计数器会等待3个机器周期来接受55h;如果在3个机器周期内接收到了55h,那么时控窗口被打开。时控窗口开放3个机器周期，期间用户可以对被保护的位进行读写。一旦时控窗口关闭，那么要重复上述过程来访问被保护的位。

```

(CLR EA)           ; if any interrupt is enabled, disable temporarily
MOV TA, #0AAH
MOV TA, #55H
(Instruction that writes a TA protected register)
(SETB EA)         ; resume interrupts enabled

```

写入AAH和55H必须发生在彼此指令的3个机器周期内。本应用禁用使用中断，在此过程中，以避免这两者之间的延迟写入。如果没有中断启用CLR EA和SETB EA的指示可以排除在外。一旦定时访问窗口关闭时，程序必须重复访问被保护的位。

以下为一组时控保护寄存器的读写范例。

## 范例1, (有效写入)

```
(CLR EA) ; if any interrupt is enabled, disable temporarily
MOV TA,#0AAH ; 2 machine-cycles.
MOV TA,#55H ; 2 machine-cycles.
ORL CHPCON,#data ; 2 machine-cycles.
(SETB EA) ; resume interrupts enabled
```

## 范例2, (无效写入)

```
(CLR EA) ; if any interrupt is enabled, disable temporarily
MOV TA,#0AAH ; 2 machine-cycles.
MOV TA,#55H ; 2 machine-cycles.
NOP ; 1 machine-cycle.
NOP ; 1 machine-cycle.
ANL ISPTRG,#data ; 2 machine-cycles.
(SETB EA) ; resume interrupts enabled
```

## 范例3, (无效写入)

```
(CLR EA) ; if any interrupt is enabled, disable temporarily
MOV TA,#0AAH ; 2 machine-cycles.
NOP ; 1 machine-cycle.
MOV TA,#55H ; 2 machine-cycles.
MOV WDCON0,#data1 ; 2 machine-cycles.
ORL PMCR,#data2 ; 2 machine-cycles.
(SETB EA) ; resume interrupts enabled
```

## 范例4, (无效写入)

```
(CLR EA) ; if any interrupt is enabled, disable temporarily
MOV TA,#0AAH ; 2 machine-cycles.
NOP ; 1 machine-cycle.
NOP ; 1 machine-cycle.
MOV TA,#55H ; 2 machine-cycles.
ANL WDCON0,#data ; 2 machine-cycles.
(SETB EA) ; resume interrupts enabled
```

在第一个例子中, 写保护三个机器周期窗口关闭之前完成。然而, 在例2中, ISPTRG的写入并没有在时控保护打开时完成, 操作完这些指令后, ISPTRG的值不会有变化。示例3中, WDCON0写入成功, 但对PMCR访问超过三个机器周期窗口, 因此PMCR值不会改变。例4, 第二次写55H对应第一个AAH写入时间超过了3个机器周期, 时控保护打开失败, 所以后面的写入全部无效。

N79E815A/814A/8132A中, 被时控访问保护的特殊功能寄存器包括PMCR, CHPCON (9FH), ISPTRG (A4H), SHBDA (9CH), WDCON0 (D8H), 和 WDCON1 (ABH)。

## 19 中断系统

N79E815A/814A/8132A系列有四个中断优先级，总有14个中断源。每个中断源都有独立的优先级位、标志、中断向量和使能位。另外，中断可被全局使能或关闭。

### 19.1 中断源

外部中断 $\overline{INT0}$ 和 $\overline{INT1}$ 可边沿触发或电平触发，根据IT0和IT1。TCON寄存器的位IE0和IE1是中断标志，用以检查产生哪个中断。在边沿触发模式，在每个机器周期采样INTx输入。如果在一个周期采样为高且下一个为低，就检测到由高到低的转换，TCON的中断请求标志lex置1。因为外部中断是在每个机器周期采样的，所以必须保持为高或低至少一个完整的机器周期，当调用服务程序时lex标志自动清零。如果选择电平触发模式，请求源必须保持引脚为低直到中断被处理，进入服务程序，标志lex将不被硬件清零。如果服务程序完成后中断继续保持为低，处理器可能应答相同中断源的另一个中断请求。

当TF0、TF1标志位置位时会产生定时器0和定时器1中断。当定时器溢出时这些标志位会置位。当执行定时器中断服务程序时，这些标志位会被硬件自动清零。

看门狗定时器可作为系统监视器或简单的定时器，在超时计数器溢出时，看门狗定时器中断标志WDIF(WDCON.3)置位，若此时中断使能位EIE.4使能，将发生中断。

串口中断发生在接收或发送时，存在两个中断源，分别取自特殊功能寄存器SCON的RI和TI位，该标志必须由软件清零。

I<sup>2</sup>C中断，在EA和ES同时使能时，I2STATUS寄存器出现新的SIO状态时产生中断。

SPI中断，标志位SPIF，在完成与外部器件的数据传输，如果SPI中断使能(ESPI at EIE.6)，会产生串行外设中断，SPIF标志由软件写0清除。MODF和SPIOVF也可能产生中断，共享向量地址。

ADC转换完成后会产生ADC中断。只有一个中断源，在ADCCON SFR的ADCI位。该位不能自动由硬件清零，用户通过软件清该位。

如果P0.2(钳位)检测到高(BKPS=1)或低(BKPS=0)产生PWM钳位中断标志BKF。此时，BKF(PWMCON2.0)由硬件置位且必须由软件清零。当10位向下计数器向下溢出时，PWM周期中断标志CF由硬件置位，只能由软件清零。如果使能PWM中断，BKF置位，请求PWM中断。

当有键盘连接到P0，且检测到低电平或边沿改变，产生键盘中断。每个键盘中断可以独立使能或禁止，KBI标志(KBIF[7:0])必须由软件清零。

LVR检测可以引起POF标志，BOF标志表示如果电源电压降到BOD电压以下，如果使能EBOD(IE.5)和EA，产生中断。

所有产生中断的位可以由硬件来置位和清零，因此可以由软件来引发相应的中断。各个中断可由IE中的相应位来打开或关闭，IE中还有EA位来控制所有中断的打开或关闭。

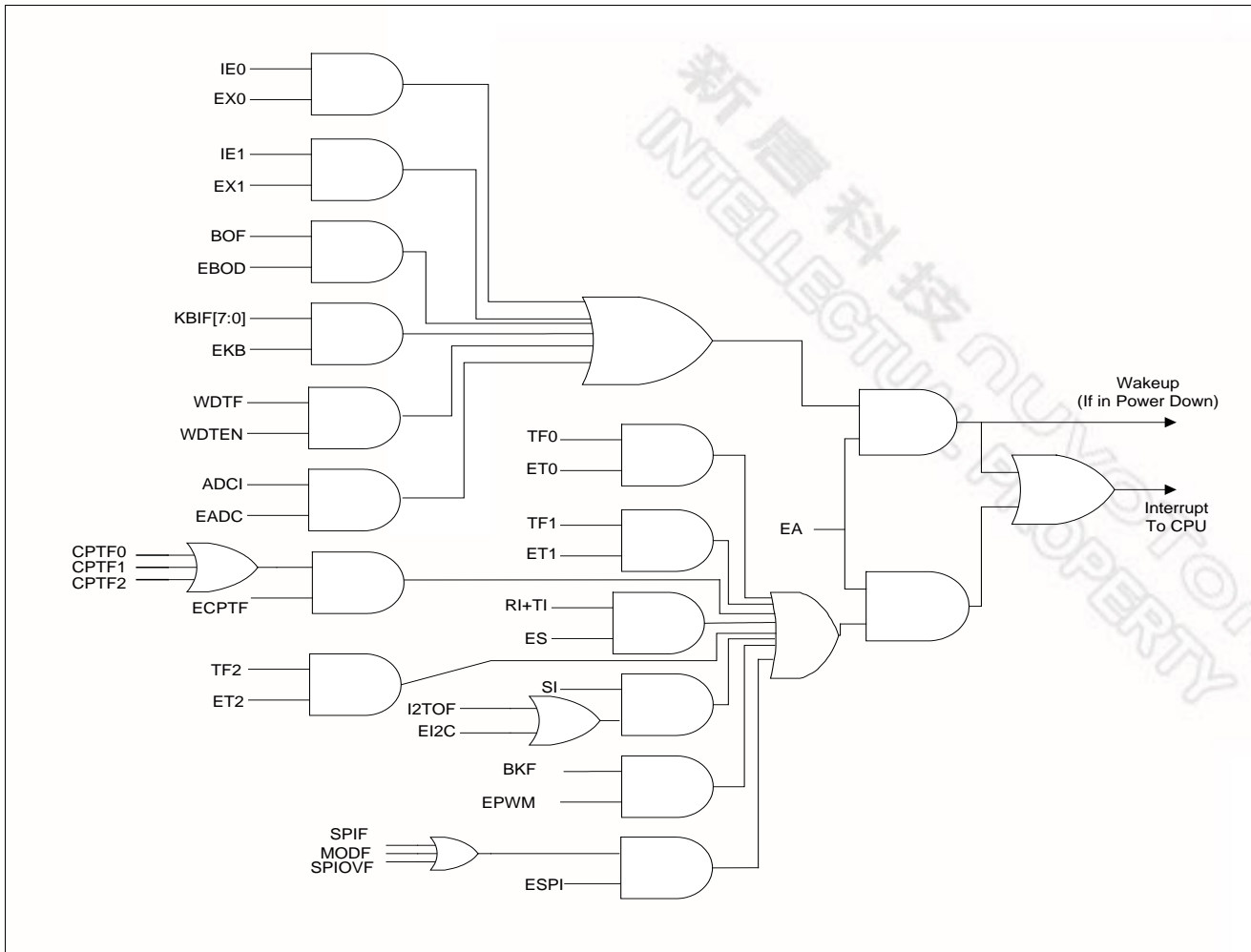


图19-1 中断标志产生框图

## 19.2 优先级结构

对中断来说，系统为其提供4种优先级：最高、高、低、最低。可以单独的将中断源设置为高低优先级，很自然较低的中断源不能中断较高的中断源。但是系统中存在一个预定义的中断处理顺序结构，用于处理同时产生且优先级又相同的中断。见表19-3，中断从最高优先级向最低优先级排序。

处理器响应一个有效的中断是通过执行一个LCALL 指令将程序转移到中断入口地址。引起中断的中断标志可能被清除也有可能不被清除，每个机器周期都检测中断标志和中断优先权。如果满足特定条件硬件将执行内部产生的LCALL 指令，目标地址是中断向量地址。产生LCALL的条件是

1. 较低优先级的中断不会打断同等优先级的中断和较高优先级的中断服务程序。
2. 在正在执行指令的最后一个周期检测中断标志。
3. 当前指令不能引起写寄存器IE, EIE, IP, IPH, EIP 或 IPH1且不是RETI。

如果上述的任何一个条件不满足，LCALL 就不会发生。在每一个指令周期都会检测中断标志。如果上述条件有一个不满足，虽然标志位置'1'，也不能响应中断。当所有的条件都满足了，中断标志已经消失，该中断也不能再被回应。

处理器响应一个有效的中断是通过执行一个LCALL 指令将程序转移到中断入口地址。引起中断的中断标志可能被清除也有可能不被清除。当进入中断服务程序定时，器中断的TF0、TF1标志会被硬件清除。外部中断INT0和INT1只有在它们的触发条件发生时他们的标志被清除。串行中断标志不能由硬件清除。定时器2中断，标志不能由硬件清零，硬件LCALL与软件LCALL指令相同，该指令保存程序计数器内容到堆栈，但是不保存程序状态字PSW。当中断发生时PC被装入中断向量地址.不同中断源的向量地址如下表

**表19-1中断源的向量位置**

中断源	向量地址	中断源	向量地址
外部中断 0	0003h	定时器 0 溢出	000Bh
外部中断1	0013h	定时器 1 溢出	001Bh
串口	0023h	定时器 2 溢出	002Bh
I <sup>2</sup> C 中断	0033h	KBI 中断	003Bh
BOD 中断	0043h	SPI 中断	004Bh
看门狗定时器	0053h	ADC 中断	005Bh
捕获	0063h		
PWM 钳位中断	0073h		

**表19-2 四级中断优先级设定**

中断优先级		中断优先级
IPXH	IPX	
0	0	级别 0 (最低)
0	1	级别 1
1	0	级别 2
1	1	级别 3 (最高)

从向量地址继续执行直到执行RETI指令，在执行RETI指令时，处理器要压栈，将PC内容保存在堆栈顶。硬件LCALL后，用户必须注意堆栈存储的状态,是否执行返回中断程序。注RET 指令与RETI指令表现相同，但它不通知中断控制器中断服务已经完成，使得控制器认为中断服务仍在进行。

N79E815A/814A/8132A系列使用4级优先级的中断结构，支持14个中断源。



每个中断源可以通过设置或清除IE 或 EIE的相关位独立使能或禁止. 寄存器 IE 也包括一个全局禁止位, EA, 可以一次性禁止全部中断.

每个中断源可以通过设置IP, IPH, EIP, 和 EIPH独立编程为四种优先级的一种. 中断服务程序可以被更高优先级别的中断中断, 但不能被同级或较低级的中断. 最高优先级中断服务程序不能被任何中断源打断. 因此, 如果两个不同优先级的中断同时产生, 先响应较高优先级的中断请求.

如果相同中断优先级的请求同时发生, 内部查询时序决定先服务哪个请求. 称之为仲裁级. 注, 仲裁级仅用于解决同时发生的相同优先级的中断请求.

下表列出中断源, 标志位, 向量地址, 使能位, 优先位, 仲裁级和哪些中断可将CPU从掉电模式唤醒.

表19-3 中断源汇总表

中断名称	中断标志位	中断向量地址	中断使能位	标志清除方式	中断级别	默认优先级	是否可从掉电模式中唤醒芯片
外部中断0	IE0	0003H	EX0 (IE0.0)	硬件 软件	IPH.0, IP.0	1 (最高)	是
BOD 检测	BOF	0043H	EBOD (IE.5)	软件	IPH.5, IP.5	2	是
看门狗定时器	WDTF	0053H	EWDI(EIE.4)	软件	EIPH.4, EIP.4	3	是
定时器0中断	TF0	000BH	ET0 (IE.1)	硬件 软件	IPH.1, IP.1	4	否
I2C中断	SI I2TOF	0033H	EI2C (EIE.0)	软件	EIPH.0, EIP.0	5	否
ADC转换器	ADC1	005BH	EADC (IE.6)	软件	IPH.6, IP.6	6	是 <sup>[1]</sup>
外部中断1	IE1	0013H	EX1 (IE.2)	硬件 软件	IPH.2, IP.2	7	是
KBI中断	KBIF[7:0]	003BH	EKB (EIE.1)	软件	EIPH.1, EIP.1	8	是
定时器1中断	TF1	001BH	ET1 (IE.3)	硬件 软件	IPH.3, IP.3	9	否
串口Tx 和 Rx	TI & RI	0023H	ES (IE.4)	软件	IPH.4, IP.4	10	否
PWM 中断	BKF	0073H	EPWM (EIE.5)	软件	EIPH.5, EIP.5	11	否
SPI	SPIF + MODF + SPIOVF	004BH	ESPI (EIE.6)	软件	EIPH.6, EIP.6	12	否
定时器 2溢出 /比较匹配	TF2	002Bh	ET2(EIE.7)	软件	EIPH.7, EIP.7	13	否
捕捉	CAPF0-2	0063H	ECPTF(EIE.2)	软件	IPH.7, IP.7	14 (最低)	否

[1]. 当ADC转换器采用的是内部RC为时钟源的方式时，可以从掉电模式中唤醒芯片。

### 19.3 中断响应时间

每一个中断源的响应时间取决于几个方面，如中断自身特点和指令的执行。外部中断  $\overline{\text{INT0}}$  和  $\text{RI+TI}$  在机器周期的  $\text{C3}$  采样并且他们相应的中断标志  $\text{lex}$  自动的置位或清除。定时器0和1溢出标志在机器周期的  $\text{C3}$  置位，在下一个机器周期检测中断标志。如果有1个中断请求满足3个条件，硬件将自动产生长跳指令，该指令需要4个机器周期。这样从中断标志置位到执行中断服务程序最少只需要5个机器周期。

很长的响应时间应该可以预知的如果三个条件有一个不满足，如果有较高或同等优先级的中断正在执行中断服务程序。很明显中断等待时间正在执行的中断服务程序的长短。如果检测机器周期正在执行指令，需等待指令执行完毕，最大的响应时间(如果不在其它中断的服务程序)发生在N79E815A/814A/8132A系列执行写  $\text{IE}$ ,  $\text{EIE}$ ,  $\text{EIE2}$ ,  $\text{IP0}$ ,  $\text{IP0H}$ ,  $\text{IP1}$ ,  $\text{IP1H}$ ,  $\text{IP2}$  或  $\text{IP2H}$  和  $\text{MUL}$ 、 $\text{DIV}$  指令。中断中断源的最长响应时间是12 机器周期，其中包括检测中断1机器周期，完成  $\text{IE}$ ,  $\text{EIE}$ ,  $\text{EIE2}$ ,  $\text{IP0}$ ,  $\text{IP0H}$ ,  $\text{IP1}$ ,  $\text{IP1H}$ ,  $\text{IP2}$  或  $\text{IP2H}$  访问2机器周期，完成  $\text{MUL}$  或  $\text{DIV}$  指令5机器周期和完成硬件  $\text{LCALL}$  中断向量位置4机器周期。

也就是说一个简单中断系统中中断响应时间总是大于5机器周期并且不大于12机器周期。最大的等待时间是12机器周期既是48时钟周期。注标准 8051 最小等待时间为8机器周期既是96时钟周期。这可以减少50%时钟周期。

### 19.4 中断控制寄存器

中断所用到的特殊功能寄存器描述如下

#### IE – 中断使能 (可位寻址)

7	6	5	4	3	2	1	0
EA	EADC	EBOD	ES	ET1	EX1	ET0	EX0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: A8H

复位值: 0000 0000B

位	名称	描述
7	EA	<b>全局中断使能位</b> 该寄存器为所有中断使能控制。 0 = 关闭所有中断 1 = 打开中断
6	EADC	<b>打开ADC 中断.</b>
5	EBOD	<b>打开 BOD 中断</b>
4	ES	<b>使能串口中断 (UART).</b> 0 = 关闭串口中断 1 = 打开串口中断. 设定 $\text{TI}(\text{SCON}.1)$ 或 $\text{RI}(\text{SCON}.0)$ .



位	名称	描述
3	ET1	使能定时器1中断。 0 = 关闭定时器1中断 1 = 打开由TF1 (TCON.7)产生的中断。
2	EX1	使能外部中断1。 0 = 关闭外部中断1。 1 = 打开由 $\overline{INT1}$ (P1.4)产生的中断。
1	ET0	使能定时器0中断。 0 = 关闭定时器0中断 1 = 打开由TF0 (TCON.5)产生的中断。
0	EX0	使能外部中断0。 0 = 关闭外部中断0。 1 = 打开由 $\overline{INT0}$ (P1.3)产生的中断。

## EIE – 扩展中断使能

7	6	5	4	3	2	1	0
ET2	ESPI	EPWM	EWDI	-	ECPTF	EKB	EI2C
r/w	r/w	r/w	r/w	-	r/w	r/w	r/w

地址: E8H

复位值: 0000 0000B

位	名称	描述
7	ET2	0: 关闭定时器2中断。 1: 打开定时器2中断。
6	ESPI	SPI 中断使能: 0: 关闭SPI中断。 1: 打开SPI中断。
5	EPWM	0: 当外部钳位引脚发生钳位时不打开PWM中断。 1: 当外部钳位引脚发生钳位时打开PWM中断。
4	EWDI	0: 关闭看门狗定时器中断。 1: 打开看门狗定时器中断。
3	-	保留。
2	ECPTF	0: 关闭捕获中断。 1: 打开捕获中断。
1	EKB	0: 关闭键盘中断。 1: 打开键盘中断。
0	EI2C	0: 关闭I <sup>2</sup> C中断。 1: 打开I <sup>2</sup> C中断。

## IP – 中断优先级寄存器

7	6	5	4	3	2	1	0
PCAP	PADC	PBOD	PS	PT1	PX1	PT0	PX0



r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
-----	-----	-----	-----	-----	-----	-----	-----

地址: B8H 复位值: 0000 0000B

位	名称	描述
7	PCAP	1: 设置捕捉0/1/2的中断优先级为最高优先级.
6	PADC	1: 设置ADC的中断优先级为较高优先级.
5	PBOD	1: 设置BOD检测器的中断优先级为较高优先级.
4	PS	1: 设置串口0的中断优先级为较高优先级.
3	PT1	1: 设置定时器1的中断优先级为较高优先级.
2	PX1	1: 设置外部中断1的中断优先级为较高优先级.
1	PT0	1: 设置定时器0的中断优先级为较高优先级.
0	PX0	1: 设置外部中断0的中断优先级为较高优先级.

#### IPH – 中断优先级寄存器高字节

7	6	5	4	3	2	1	0
PCAPH	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: B7H 复位值: 0000 0000B

位	名称	描述
7	PCAPH	1: 设置捕捉0/1/2中断高优先级为最高优先级.
6	PADCH	1: 设置ADC中断高优先级为最高优先级.
5	PBODH	1: 设置BOD检测中断高优先级为最高优先级.
4	PSH	1: 设置串口0中断高优先级为最高优先级.
3	PT1H	1: 设置定时器1中断高优先级为最高优先级.
2	PX1H	1: 设置外部中断1中断高优先级为最高优先级.
1	PT0H	1: 设置定时器0中断高优先级为最高优先级.
0	PX0H	1: 设置外部中断0中断高优先级为最高优先级.

#### EIP – 扩展中断优先级寄存器

7	6	5	4	3	2	1	0
PT2	PSPI	PPWM	PWDI	-	-	PKB	PI2
r/w	r/w	r/w	r/w	-	-	r/w	r/w

地址: FFH 复位值: 0000 0000B

位	名称	描述
7	PT2	1: 设置定时器2的中断优先级为较高优先级.
6	PSPI	1: 设置SPI的中断优先级为较高优先级.
5	PPWM	1: 设置PWM's钳位的中断优先级为较高优先级.
4	PWDI	1: 设置看门狗的中断优先级为较高优先级.
3:2	-	保留
1	PKB	1: 设置键盘的中断优先级为较高优先级.



位	名称	描述
0	PI2	1: 设置I2C的中断优先级为较高优先级

## EIPH – 扩展中断优先级寄存器高字节

7	6	5	4	3	2	1	0
PT2H	PSPIH	PPWMH	PWDIH	-	-	PKBH	PI2H
r/w	r/w	r/w	r/w	-	-	r/w	r/w

地址: F7H

复位值: 0000 0000B

位	名称	描述
7	PT2H	1: 设置定时器2的中断高优先级为最高优先级.
6	PSPIH	1: 设置SPI的中断高优先级为最高优先级.
5	PPWMH	1: 设置PWM 外部钳位中断高优先级为最高优先级.
4	PWDIH	1: 设置看门狗的中断高优先级为最高优先级.
3:2	-	保留
1	PKBH	1: 设置键盘的中断高优先级为最高优先级.
0	PI2H	1: 设置I2C的中断高优先级为最高优先级

## TCON – 定时器0和定时器1控制 (可位寻址)

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址: 88H

复位值: 0000 0000B

位	名称	描述
3	IE1	<b>外部中断1边沿标志</b> 当检测到边沿/电平类型时, 该标志由硬件置位. 如果 IT1 = 1, 该位将保持置1直到软件清零或在外部中断1服务程序中清零. 如果 IT1 = 0, 该标志是 $\overline{INT1}$ 输入信号逻辑电平的反转.
2	IT1	<b>外部中断1类型选择</b> 该位选择 $\overline{INT1}$ 的中断触发类型是下降沿还是低电平. 0 = $\overline{INT1}$ 为低电平触发. 1 = $\overline{INT1}$ 为下降沿触发.
1	IE0	<b>外部中断0边沿标志</b> 当检测到边沿/电平类型时, 该标志由硬件置位. 如果 IT0 = 1, 该位将保持置1直到软件清零或在外部中断1服务程序中清零. 如果 IT0 = 0, 该标志是 $\overline{INT0}$ 输入信号逻辑电平的反转.
0	IT0	<b>外部中断0类型选择.</b> 该位选择 $\overline{INT0}$ 的中断触发类型是下降沿还是低电平. 0 = $\overline{INT0}$ 为低电平触发. 1 = $\overline{INT0}$ 为下降沿触发.

## 20 在线编程 (ISP)

内部程序存储器和片上数据内存支持硬件编程和在线编程功能(ISP)。硬件编程是在产品进入批量生产阶段，采用编程器编程，可以减少费用和时间。然而，如果产品在研发阶段或产品需要更新软固件时，硬件编程就显得不太方便，采用ISP方式，可使这一过程变得方便。N79E815A/814A/8132A支持ISP模式，允许通过软件模式更新程序。更新应用程序所需的电压为： $V_{DD} = 3.0V \sim 5.5V$ 。

执行ISP不需要将控制器从系统板上拆下来。最常用的方法就是通过UART执行ISP，此时LDROM中需要有固件代码。也就是PC通过串口传输新的APROM代码，LDROM中的固件接受并通过ISP命令重新编程到APROM中。Nuvoton 提供ISP固件，请到如下Nuvoton 8-位微控制器网站查看。选择“Nuvoton ISP-ICP Programmer”。

### 20.1 ISP引导程序

不同于对暂存器可以实时操作，更新存储器数据需要较长时间。因此，需要复杂的时序控制擦、写、读数据。N79E815A/814A/8132A提供方便的机制帮助用户更新程序。通过设置ISPEN (CHPCON.0 受TA保护)使能ISP后，用户可以很容易写入16位目标地址ISPAH和ISPAL，给ISPFD写入数据，给ISPCN写入命令，然后通过设置触发ISPGO (ISPTRG.0)准备执行ISP。注ISPTRG也受TA保护。

设置ISPGO (ISPTRG.0)，开始执行ISP。注ISPTRG也是受TA保护。此时，CPU保持程序计数器，内建ISP自动控制内部电源升为高电压，并控制信号时序。ISP动作完成后，程序计数器继续运行之后的指令。ISPGO位将自动清零。如果需要再次执行ISP动作，用户只需要重复以上步骤，通过这个程序可以方便用户擦除，编程，校验内存。

下面是关于ISP的寄存器。

#### CHPCON – 芯片控制 (时控保护)

7	6	5	4	3	2	1	0
SWRST	ISPF	LDUEN	-	-	-	BS	ISPEN
w	r	r/w	-	-	-	r/w	r/w

地址：9FH

复位值：见表7-2.N79E815A/814A/8132ASFR功能描述和复位值

位	名称	描述
6	ISPF	<b>ISP 错误标志(只读)</b> 以下条件满足时，硬件置位该位： 1. 以下访问是不允许的，如， (a) 当APROM中代码在运行时，擦除或编程APROM自身。 (b) 当APROM代码在运行时，但LDUEN为0，擦除或编程LDROM。 I 当APROM代码在运行时，擦除，编程或读取CONFIG字节。

位	名称	描述
		(d) 当LDROM代码在运行时，擦除或编程LDROM。 1 访问超过其大小的区域。 2. ISP的操作由内部程序空间进入外部程序空间。 该位应由软件清零
5	LDUEN	<b>更新LDROM使能</b> 0 = 当APROM代码在运行时，禁止擦除或编程LDROM，LDROM保持只读。 1 = 当APROM代码在运行时，允许访问LDROM。
4:2	-	保留
1	BS	<b>启动选择</b> 对该位写或读的不同方式。 写： 定义了复位后MCU启动的区块。 0 = 下一次从APROM启动。 1 = 下一次从LDROM启动。 读： 定义了前次复位后MCU启动的区块。 0 = 前次从APROM启动。 1 = 前次从LDROM启动。
0	ISPEN	<b>ISP 使能</b> 0 = 打开ISP功能。 1 = 关闭ISP功能。 使能ISP功能将以内部的22.1184MHz RC 振荡器为时钟。清ISPEN应该是ISP操作后的最后一条指令，这样可以停止内部RC以减少功耗。

**ISPCN – ISP 控制**

7	6	5	4	3	2	1	0
ISPA17	ISPA16	FOEN	FCEN	FCTRL.3	FCTRL.2	FCTRL.1	FCTRL.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

地址：AFH

复位值：0011 0000B

位	名称	描述
7:6	ISPA[17:16]	<b>ISP 控制</b> 该字节为ISP控制命令，以决定ISP动作。
5	FOEN	
4	FCEN	
3:0	FCTRL[3:0]	





### ISPAH – ISP地址高字节

7	6	5	4	3	2	1	0
ISPA[15:8]							
r/w							

地址: A7H

复位值: 0000 0000B

位	名称	描述
7:0	ISPA[15:8]	<b>ISP 地址高字节</b> ISPAH 包含地址ISPA[15:8]。

### ISPAL – ISP 地址低字节

7	6	5	4	3	2	1	0
ISPA[7:0]							
r/w							

地址: A6H

复位值: 0000 0000B

位	名称	描述
7:0	ISPA[7:0]	<b>ISP 地址低字节</b> ISPAL 包含地址ISPA[7:0]。

### ISPF D – ISP 内存数据

7	6	5	4	3	2	1	0
ISPF D[7:0]							
r/w							

地址: AEH

复位值: 0000 0000B

位	名称	描述
7:0	ISPF D[7:0]	<b>ISP 内存数据</b> 该字节包含将要读或写进内存空间的数据。编程模式下，用户需要在触发ISP之前写数据到ISPF D里，读/校验模式下，在ISP完成后从ISPF D读出数据。

### ISPTRG – ISP 执行 (时控保护)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	ISPGO
-	-	-	-	-	-	-	w

地址: A4H

复位值: 0000 0000B

位	名称	描述
0	ISPGO	<b>执行ISP</b> 设置该位为1开始执行ISP。该指令后，CPU保持程序计数器 (PC)，ISP硬件自动管理控制该过程。ISP完成后，程序计数器继续执行。ISPGO位自动清零，保持为0。

## 20.2 ISP 命令行

ISP 命令		ISPCN				ISPAH, ISPAL A[15:0]	ISPDF D[7:0]
		A17, A16	FOEN	FCEN	FCTRL[3:0]		
APROM & 数据FLASH	FLASH 页擦除	0, 0	1	0	0010	地址写入 A[15:0]	x <sup>[1]</sup>
	FLASH 编程	0, 0	1	0	0001	地址写入 A[15:0]	数据写入 D[7:0]
	FLASH 读数据	0, 0	0	0	0000	地址写入 A[15:0]	数据读出 D[7:0]
LDROM	FLASH 页擦除	0, 1	1	0	0010	地址写入 A[15:0]	x <sup>[1]</sup>
	FLASH 编程	0, 1	1	0	0001	地址写入 A[15:0]	数据写入 D[7:0]
	FLASH 读数据	0, 1	0	0	0000	地址写入 A[15:0]	Data out D[7:0]
CONFIG <sup>[2]</sup> 页擦除		1, 1	1	0	0010	地址写入 A[15:0]=0000H	x <sup>[1]</sup>
CONFIG <sup>[2]</sup> 编程		1, 1	1	0	0001	地址写入 A[15:0]	数据写入 D[7:0]
CONFIG <sup>[2]</sup> 读数据		1, 1	0	0	0000	地址写入 A[15:0]	数据读出 D[7:0]





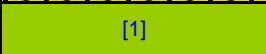

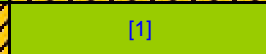





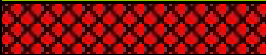

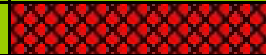







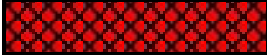
注:

[1] 'x' 表示无关项。

[2] 'CONFIG' 即芯片的配置位, 对芯片硬件特性进行定义。

[3] 每页容量为128字节。所以页擦除时, 写入的地址应为 0000H, 0080H, 0100H, 0180H, 0200H, ..., 每0080H 增加一次地址。

## 20.3 ISP 编程访问表

目标区域	不加密		加密	
	ISP 引导代码所在区域		ISP 引导代码所在区域	
	APROM	LDROM	APROM	LDROM
APROM				
LDROM	 [1]		 [1]	
Data Flash				
CONFIGs		 [2]		 [2]
ID (read)				
颜色区块说明	注释			
	全功能操作			
	只读			
	无法操作			
[1]	LDUE 应设定为 1, 否则为 只读			
[2]	重新设定的配置位 CONFIG 功能, 必须在执行过 POR、WDT、复位脚复位或软件复位后才能生效			

注:

- I. 加密时, 可以通过LDROM完全访问CONFIG。
- II. 禁止APROM跳转到LDROM或LDROM跳转到APROM中。
- III. 程序在APROM中运行, 不能读CONFIGs。

## 20.4 ISP用户指南

ISP可以方便用户更新存储器内容, 不过, 用户必须遵循一定限制以确保ISP正确执行, 否则可能引起不确定的结果, 甚至损坏器件。同时该段对于正确执行ISP非常有用。

(1) 没有ISP操作时, 用户必须清ISPEN (CHPCON.0)为0。可以防止系统意外触发ISP。此外, ISP需要使用内部22.1184MHZ RC振荡器。如果选择外部时钟源, 禁止ISP将停止内部22.1184MHz RC, 可以达到省电的目的。注ISPEN受TA保护。

(2) CONFIG 字节仅在执行LDROM中的启动代码时可以被ISP完全访问。所有复位后, 除CBS位外, 新的CONFIG字节全都激活。除软件复位外的所有复位后, 新的CBS位也被激活。

(3) 当LOCK位(CONFIG0.1) 被激活, ISP读, 写或擦除仍然有效。

(4) ISP 工作于 $V_{DD}= 3.0V \sim 5.5V$ 。

(5) APROM 和 LDROM 可以通过ISP方法读自身内容。

**注用户可以自行开始 ISP程序, 为保护数据安全, 擦除和编程CONFIG字节必须是最后步骤。**

## 20.5 ISP 例程

### ISP通用子程序

#### Enable\_ISP:

```
MOV ISPCN,#00110000b      ; select "Standby" mode
CLR   EA                  ; if any interrupt is enabled, disable temporarily
MOV TA,#0AAH              ; CHPCON is TA-Protection
MOV TA,#55H                ;
ORL CHPCON,#00000001b     ; ISPEN=1, enable ISP function
SETB  EA
CALL  Trigger_ISP ;
RET
```

#### Disable\_ISP:

```
MOV ISPCN,#00110000b      ; select "Standby" mode
CALL  Trigger_ISP         ;
CLR   EA                  ; if any interrupt is enabled, disable temporarily
MOV TA,#0AAH              ; CHPCON is TA-Protection
MOV TA,#55H ;
ANL CHPCON,#11111110b     ; ISPEN=0, disable ISP function
SETB  EA
RET
```

#### Trigger\_ISP:

```
CLR   EA                  ; if any interrupt is enabled, disable temporarily
MOV TA,#0AAH              ; ISPTRG is TA-Protection
MOV TA,#55H ;
MOV ISPTRG,#00000001b     ; write '1' to bit ISPG0 to trigger an ISP processing
SETB  EA
RET
```

#### FLASH Page Erase (target address in APROM/Data Flash/LDROM area)

```
CALL  Enable_ISP
MOV ISPCN,#00100010b      ; select "FLASH Page Erase" mode, (A17,A16)=(0,0) for
                          ; APROM/Data Flash/LDROM
MOV ISPAH,???H            ; fill page address
MOV ISPAL,???H
CALL  Trigger_ISP
CALL  Disable_ISP
```



#### FLASH Program (target address in APROM/Data Flash/LDROM area)

```
CALL Enable_ISP
MOV ISPCN,#00100001b      ; select "FLASH Program" mode, (A17,A16)=(0,0) for APROM/Data
                          ; Flash/LDROM
MOV ISPAH,###H           ; fill byte address
MOV ISPAL,###H
MOV ISPFd,###H           ; fill data to be programmed
CALL Trigger_ISP
CALL Disable_ISP
```

#### FLASH Read (target address in APROM/Data Flash/LDROM area)

```
CALL Enable_ISP
MOV ISPCN,#00000000b      ; select "FLASH Read" mode, (A17,A16)=(0,0) for APROM/Data
                          ; Flash/LDROM
MOV ISPAH,###H           ; fill byte address
MOV ISPAL,###H
CALL Trigger_ISP
MOV  A,ISPFd              ; now, ISPFd contains the Flash data, move to ACC for further
                          ; use
CALL Disable_ISP
```

#### CONFIGPage Erase (target address in CONFIGarea)

```
CALL Enable_ISP
MOV ISPCN,#11100010b      ; select "CONFIG Page Erase" mode, (A17,A16)=(1,1) for CONFIG
MOV ISPAH,#00H           ; fill page address #0000H, because there is only one page
MOV ISPAL,#00H
CALL Trigger_ISP
CALL Disable_ISP
```

#### CONFIG Program (target address in CONFIGarea)

```
CALL Enable_ISP
MOV ISPCN,#11100001b      ; select "CONFIG Program" mode, (A17,A16)=(1,1) for CONFIG
MOV ISPAH,#00H           ; fill byte address, 0000H/0001H/0002H/0003H for CONFIG0/1/2/3,
                          ; respectively
MOV ISPAL,###H           ; fill data to be programmed
MOV ISPFd,###H
CALL Trigger_ISP
CALL Disable_ISP
```

#### CONFIG Read (target address in CONFIGarea)

```
CALL Enable_ISP
MOV ISPCN,#11000000b      ; select "CONFIG Read" mode, (A17,A16)=(1,1) for CONFIG
MOV ISPAH,#00H           ; fill byte address, 0000H/0001H/0002H/0003H for CONFIG0/1/2/3,
                          ; respectively
MOV ISPAL,###H
CALL Trigger_ISP
MOV A,ISPFd              ; now, ISPFd contains the CONFIG data, move to ACC for
                          ; further use
CALL Disable_ISP
```

## 21 电源管理

N79E815A/814A/8132A有几个功能模块可以帮助用户控制设备的功耗。省电的途径有掉电模式和空闲模式。为了确保最小功耗，需要特别注意P0管脚。

在系统省电模式下，看门狗定时器需要特别注意。在进入或退出空闲或掉电模式后，硬件会自动清除WDT计数器，以避免系统复位。

### PCON – 电源控制

7	6	5	4	3	2	1	0
SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL
r/w	r/w	-	r/w	r/w	r/w	r/w	r/w

地址：87H

复位值：见表7-2.N79E815A/814A/8132ASFR功能描述和复位值

位	名称	描述
1	PD	<p><b>掉电模式</b></p> <p>设置该位使MCU进入掉电模式。在此模式下，CPU和外设时钟停止，程序计数器（PC）挂起。此时为最小功耗。CPU从掉电模式下唤醒后，该位自动由硬件清零，且在系统唤醒之前程序继续执行中断服务程序（ISR）。从ISR返回后，设备继续执行系统进入掉电模式时所处指令。</p> <p>注如果IDL位和PD位同时置位，MCU进入掉电模式。从掉电模式退出后不会进入空闲模式。</p>
0	IDL	<p><b>空闲模式</b></p> <p>设置该位使MCU进入空闲模式。在此模式下，CPU时钟停止，且程序计数器（PC）挂起。CPU从空闲模式唤醒后，该位自动由硬件清零，且在系统唤醒之前程序继续执行中断服务程序（ISR）。从ISR返回后，设备继续执行系统进入掉电模式时所处指令。</p>

### 21.1 空闲模式

空闲模式下通过保持程序计数器使CPU挂起。在空闲模式下没有程序代码的取指和运行。这迫使CPU处于待机状态。程序计数器(PC)，堆栈指针(SP)，程序状态字(PSW)，累加器(ACC)，和其他寄存器在空闲模式下保持其值不变。端口引脚保持进入空闲前的逻辑状态。通常空闲模式下的功耗约为工作状态下的一半。

既然在空闲模式下，外设电路如定时器和串口的时钟依然工作，则可以通过相应使能的中断唤醒CPU。用户可通过向IDL (PCON.0)写1进入空闲模式。这条指令是系统进入空闲模式前的最后一条指令。

有两种方法可以中止空闲模式，方法一，任何使能的中断可以使系统退出空闲模式。这将自动清零IDL位，中止空闲模式，且将执行中断服务程序 (ISR)。在使用指令RETI跳出ISR，所执行的将是进入空闲模式指令后的程序。第二种方法是除软件复位外的所有复位。

## 21.2 掉电模式

掉电模式是N79E815A/814A/8132A进入的最低功耗状态，保持功耗在“微安uA”级。此时停止内部RC时钟或外部晶振。CPU和外设如定时器或UART都待机，Flash内存也停止，所有动作完全停止，功耗降至最低。可以通过向PD (PCON.1)写1进入掉电模式。在掉电模式下，RAM保存其内容，端口引脚的值也保持不变。

N79E815A/814A/8132A有两种方法可以退出掉电模式。一，除软件复位外的所有复位。欠压检测复位将把CPU由掉电模式唤醒，在系统进入掉电模式之前要确保使能欠压检测。除最低功耗外，很少在掉电模式下使能欠压检测，也不推荐这种用法。当然RST引脚的复位或上电复位也可以唤醒掉电状态。RST引脚复位或上电复位后，CPU初始化，并开始执行程序。

N79E815A/814A/8132A可以通过外部中断脚唤醒掉电模式，并有自己的中断向量和终端标志。在中断条件设定的情况下，外部中断信号会唤醒内部同步时钟从而唤醒系统，设备执行相应的外部中断的中断服务例程 (ISR)。ISR完成后，如果程序执行掉电指令，系统会再进入掉电模式。

低电压检测中断BOD，是唤醒掉电模式的另一个来源。如前所述BOD检测电路的功耗较大，用户如果开启BOD唤醒系统在掉电模式下功耗也会增加，所以这不是一个典型的应用。

芯片可通过BOD、看门狗或KBI中断唤醒掉电模式。但需注意：如采用BOD唤醒模式，打开BOD侦测模块，会在掉电模式下，使芯片消耗更多的电流。如采用KBI中断模式，若为下降边沿唤醒，则在进入掉电模式前，端口电位必须为高；若为上升边沿唤醒，在进入掉电模式前，端口电位必须为低。

### 掉电唤醒准备时间

N79E815A/814A/8132A 在进入掉电模式时需要一些准备时间。在一些特殊应用场合，例如，系统采用INT1的下降沿唤醒掉电模式后，马上又进入掉电模式，则下图中水平距离的 $\Delta X$ 时间就是进入掉电模式的准备时间，该时间长度约为1.45ms。

注：, 黄线为: INT1, 红线为: P2.1状态指示

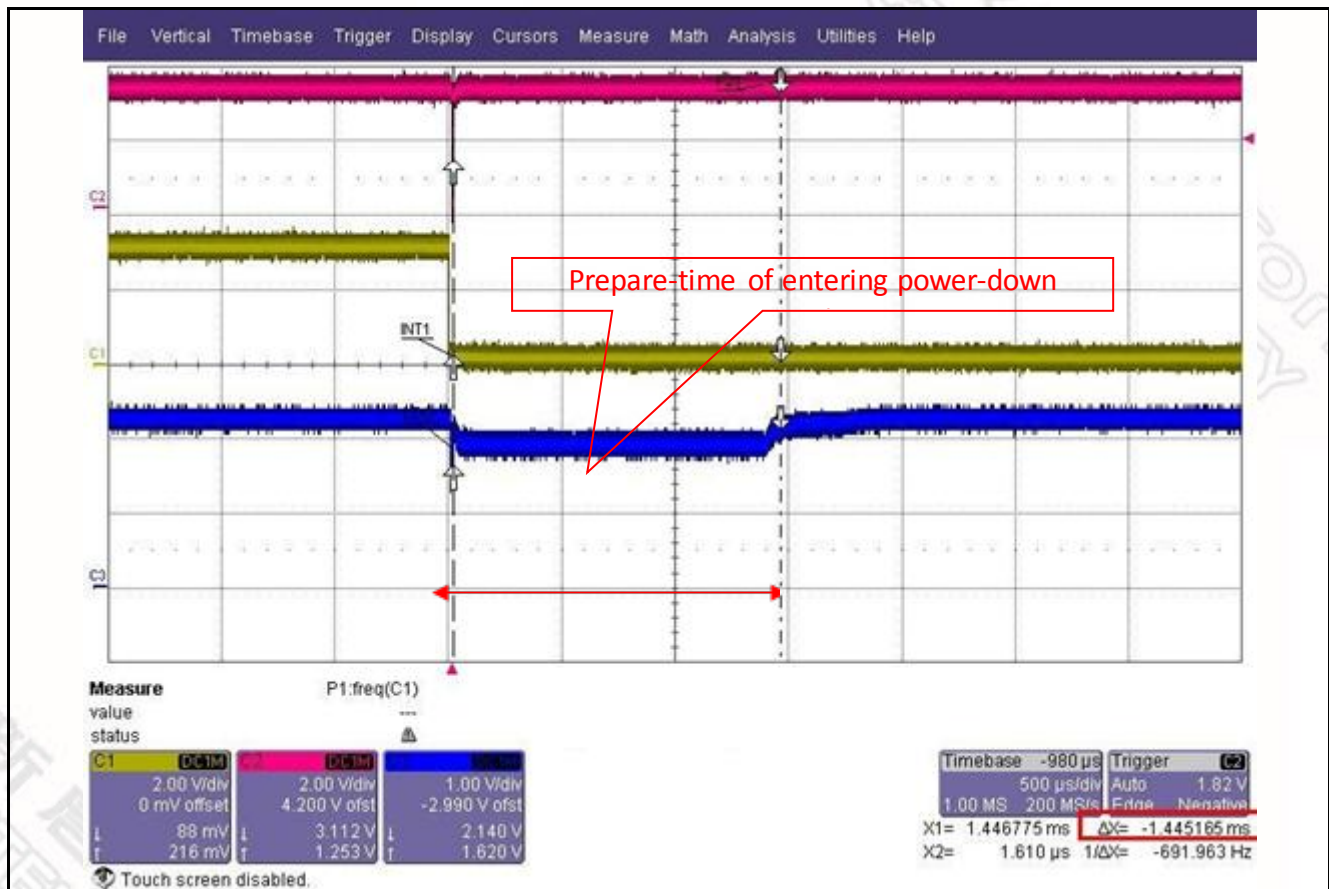


图21-1 进入掉电模式所需要的准备时间

在采用内部RC10kHz 时钟源的情况下，由于内部RC10kHz 时钟有 $\pm 50\%$ 的偏差度，所以掉电模式准备时间偏差较大，最大时长可能达到 $1.45\text{ms} * 1.5 = 2.17\text{ms}$ 。该时长由硬件决定，无法缩短。



## 22 时钟系统

N79E815A/814A/8132A提供三个可选的时钟源。可由 $F_{osc}$ (CONFIG3.1~0)配置。可从晶振/谐振器，片上RC振荡器，或XTAL1引脚的外部时钟选择时钟系统。N79E815A/814A/8132A通过CONFIG设置选择片上22.1184MHz/11.0592MHz的RC振荡器，误差为 $\pm 1\%$ （温室下，VDD=5V）。如果外部时钟源为晶振，频率支持从4MHz到24MHz。

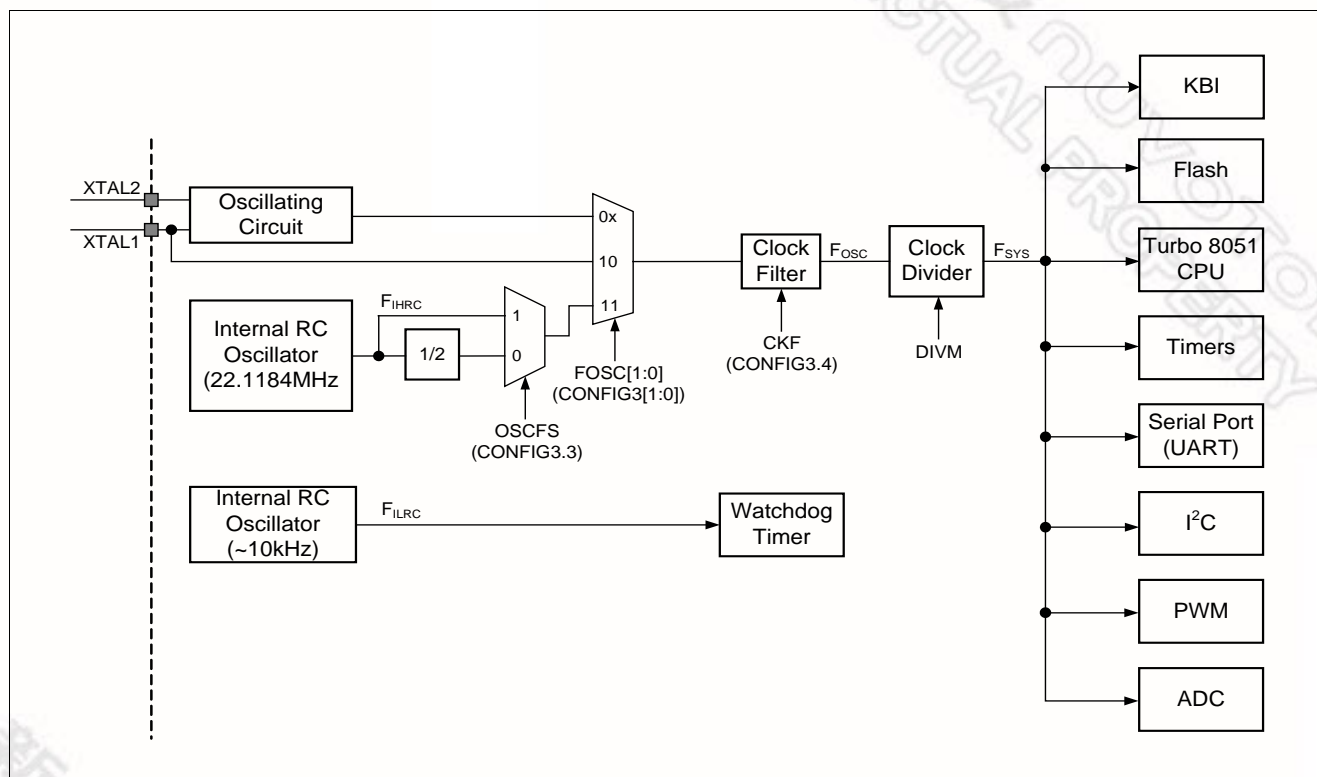


图22-1. 时钟系统模块框图

## 配置位3 (CONFIG3)

7	6	5	4	3	2	1	0
CWDTEN	-	-	CKF	OSCFS	-	FOSC1	FOSC0
r/w	-	-	r/w	r/w	-	r/w	r/w

未编程值: 1111 1111B

位	名称	描述									
4	CKF	<b>时钟滤波器使能</b> 1 = 使能时钟滤波器, 可增强抗噪声和EMC的能力。 0 = 禁止时钟滤波器。									
3	OSCFS	<b>内部RC振荡器频率选择</b> 1 = 如果内部RC振荡模式被占用, 则选择22.1184MHz 作为时钟系统。绕过2分频内部振荡器, 选择22.1184MHz输出作为时钟系统源。 0 = 如果内部RC振荡模式被占用, 选择11.0592MHz 作为时钟系统。选择2分频内部RC。内部振荡器配置为11.0592MHz 输出用途时钟系统									
2	-	保留。									
1:0	FOSC1 FOSC0	<b>振荡器选择位</b> 芯片时钟源选择。见下表: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>(FOSC1, FOSC0)</th> <th>芯片时钟源</th> </tr> </thead> <tbody> <tr> <td>(1, 1)</td> <td>内部RC振荡器</td> </tr> <tr> <td>(1, 0)</td> <td>-保留位</td> </tr> <tr> <td>(0, 1)</td> <td rowspan="2">外部晶振, 4MHz ~ 24MHz</td> </tr> <tr> <td>(0, 0)</td> </tr> </tbody> </table>	(FOSC1, FOSC0)	芯片时钟源	(1, 1)	内部RC振荡器	(1, 0)	-保留位	(0, 1)	外部晶振, 4MHz ~ 24MHz	(0, 0)
(FOSC1, FOSC0)	芯片时钟源										
(1, 1)	内部RC振荡器										
(1, 0)	-保留位										
(0, 1)	外部晶振, 4MHz ~ 24MHz										
(0, 0)											

## DIVM – 时钟分频寄存器

7	6	5	4	3	2	1	0
DIVM[7:0]							
r/w							

地址: 95H

复位值: 0000 0000B

位	名称	描述
7:0	DIVM[7:0]	<b>时钟分频</b> 系统时钟频率 $F_{SYS}$ 依据DIVM的值按下面公式计算。 当DIVM = 00H时 $F_{SYS} = F_{OSC}$ , 当 DIVM = 01H ~ FFH时, $F_{SYS} = \frac{1}{2(DIVM+1)} \times F_{OSC}$ ,

## 22.1 片内RC振荡器

当FOSC (CONFIG3.1~0)为1时, 打开片内RC振荡器。设置OSCFS (CONFIG3.3)为逻辑1将切换到2分频。

## 22.2 晶振 /谐振器

当FOSC[1:0]编程为[0:1]，系统时钟选择为晶振/谐振器。XTAL1和XTAL2分别为输入和输出，内部有反向放大器。通过连接XTAL1和XTAL2，可以使用晶振/谐振器。晶振或谐振器的频率可以从4MHz到24MHz。CKF (CONFIG3.4)是XTAL1作为输入引脚的电路时钟滤波器的控制位。

## 23 电源监控

为了防止上电和掉电时出现错误执行，N79E815A/814A/8132A提供两个电源监控功能，上电检测和欠压检测BOD。

### 23.1 上电检测

上电检测功能用于检测电源上升到系统可以工作的电压。上电检测后，POF (PCON.4) 将置1决定冷复位，上电复位完成。POF标志可由软件清零。

### 23.2 欠压检测

另一个电源监控功能是欠压检测，欠压检测电路是用于监测执行期间 $V_{DD}$ 电平。有两个可编程的欠压检测触发电平，以适用于宽电压应用。两个电平2.7V和3.8V可由CONFIG2的CBOV设置选择。当 $V_{DD}$ 下降到所选择的欠压检测触发电平( $V_{BOD}$ )，欠压检测逻辑将复位CPU或请求欠压检测中断。用户可同应用系统决定欠压复位还是使能中断。

当 $V_{DD}$ 下降到 $V_{BOD}$ 以下，且BORST (PMCR.4)为0时，欠压检测将请求中断。此情况下，BOF (PMCR.3)将置为1。用户清该标志后，当 $V_{DD}$ 保持在 $V_{BOD}$ 以下，BOF将不再置位。BOF仅通知用户有电源下降事件发生。当 $V_{DD}$ 上升到高于 $V_{BOD}$ 时，BOF将置1，以示电源恢复。 $V_{BOD}$ 有20~200mV的迟滞范围。

#### 配置位2

7	6	5	4	3	2	1	0
CBODEN	CBOV	-	CBORST	-	-	-	-
r/w	r/w	-	r/w	-	-	-	-

未编程值：1111 1111B

位	名称	描述									
7	CBODEN	<b>配置 欠压检测使能</b> 1 = 禁止欠压检测。 0 = 使能欠压检测。 BODEN 由复位后CBODEN (CONFIG2, 位-7)的取反值初始化。									
6	CBOV	<b>配置欠压检测电压选择</b> 该位选择欠压检测电压 <table border="1" data-bbox="397 1732 990 1890"> <thead> <tr> <th>CONFIG位 CBOV</th> <th>SFR BOV</th> <th>欠压检测电压</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>使能欠压检测= 2.7V</td> </tr> <tr> <td>0</td> <td>1</td> <td>使能欠压检测= 3.8V</td> </tr> </tbody> </table>	CONFIG位 CBOV	SFR BOV	欠压检测电压	1	0	使能欠压检测= 2.7V	0	1	使能欠压检测= 3.8V
CONFIG位 CBOV	SFR BOV	欠压检测电压									
1	0	使能欠压检测= 2.7V									
0	1	使能欠压检测= 3.8V									
5	-	保留									

位	名称	描述
4	CBORST	<p><b>配置欠压检测复位使能</b></p> <p>该位决定欠压检测事件后是否发生欠压检测复位。</p> <p>1 = 当VDD下降到VBOD以下或VDD上升到VBOD以上，使能欠压检测复位。</p> <p>0 = 当VDD下降到VBOD 以下，禁止欠压检测复位。当VDD下降到VBOD以下或VDD上升到VBOD以上，芯片将置位BOF。</p>

### PMCR –电源监测控制(时控保护)

7	6	5	4	3	2	1	0
BODEN	BOV	-	BORST	BOF	-	-	-
r/w	r/w	-	r/w	r/w	-	-	-

地址: A3H

复位值: 见 表7-2.N79E815A/814A/8132ASFR功能描述和复位值

位	名称	描述									
7	BODEN	<p><b>欠压检测功能控制</b></p> <p>BODEN 由复位后CBODEN (CONFIG2, 位-7)的取反值初始化。</p> <p>1 = 使能欠压检测。</p> <p>0 = 禁止欠压检测。</p>									
6	BOV	<p><b>欠压检测电压选择位</b></p> <p>复位后，欠压检测由CONFIG2的CBOV的值初始化。</p> <p>欠压检测电压选择位:</p> <table border="1"> <thead> <tr> <th>CONFIG-位 CBOV</th> <th>SFR BOV</th> <th>欠压检测电压</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>使能欠压检测= 2.7V</td> </tr> <tr> <td>0</td> <td>1</td> <td>使能欠压检测= 3.8V</td> </tr> </tbody> </table>	CONFIG-位 CBOV	SFR BOV	欠压检测电压	1	0	使能欠压检测= 2.7V	0	1	使能欠压检测= 3.8V
CONFIG-位 CBOV	SFR BOV	欠压检测电压									
1	0	使能欠压检测= 2.7V									
0	1	使能欠压检测= 3.8V									
5	-	保留									
4	BORST	<p><b>欠压检测复位使能</b></p> <p>该位决定欠压检测事件后是否发生欠压检测复位。</p> <p>0 = 当VDD下降到VBOD以下或VDD上升到VBOD以上，禁止欠压检测复位。当VDD下降到VBOD 以下，芯片将置位BOF。</p> <p>1 = 当VDD下降到VBOD以下或VDD上升到VBOD以上，使能欠压检测复位。</p>									
3	BOF	<p><b>欠压检测标志</b></p> <p>当VDD下降到VBOD以下或VDD上升到VBOD以上时，该标志由硬件设置为逻辑1。如果EBOD (IE.5) 和 EA (IE.7) 都置位，将请求欠压检测中断。该位必须由软件清零。</p>									
2	-	<b>必须设置为逻辑0。</b>									
1	-	保留									
0	-	保留									

**注意** 芯片复位后如果BOF是1，强烈建议用户在程序的初始化时清零BOF。

## 24 复位条件

N79E815A/814A/8132A的复位条件有几个选项。通常，一些寄存器复位后的值与复位条件无关，但是一些复位源的标志位的状态取决于复位源。有5种方法使芯片进入复位状态。他们是上电复位,RST 引脚复位,，软件复位,看门狗定时器复位, 以及欠压复位。

### 24.1 上电复位

N79E815A/814A/8132A包含内部参考电压。在上电过程中，当 $V_{DD}$ 低于参考电压门限值，参考电压保持CPU为复位模式。这种设计使CPU在 $V_{DD}$ 不满足执行读取存储器时，不访问程序存储器空间。如果从程序存储器读取并执行一个不确定的操作码，可能会使CPU甚至是整个系统进入错误状态。 $V_{DD}$ 上升到参考门限电压以上，系统工作，所选的振荡器起振，程序从0000H开始执行。同时，上电标志 POF (PCON.4) 置1表示冷复位，上电复位完成。注：上电后，内部RAM的内容不确定。建议用户初始化RAM。P1.6, P1.7, P1.0 和 P1.1 在芯片复位时为准双向类型。

建议通过软件清 POF 为0，以检测在下次复位是冷复位还是热复位。如果是由掉电或上电引起的冷复位，POF将再次置1。如果是由其他复位源引起的热复位，POF将保持为0。用户可以检测复位标志位，处理热复位事件。

#### PCON – 电源控制

7	6	5	4	3	2	1	0
SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL
r/w	r/w	-	r/w	r/w	r/w	r/w	r/w

地址：87H

复位值：见表7-2.N79E815A/814A/8132ASFR功能描述和复位值

位	名称	描述
4	POF	上电复位标志 上电复位后该位置1。表示一次冷复位，上电复位完成。在其他复位后该位保持不变，建议软件清该标志。

## 24.2 欠压复位

欠压检测电路于用监测系统运行时 $V_{DD}$ 电平。当 $V_{DD}$ 下降到所选的欠压触发电平 ( $V_{BOD}$ )或上升到 $V_{BOD}$ ，如果BORST (PMCR.4) 置 1，欠压检测逻辑将复位CPU。

### PMCR –电源监测控制(时控保护)

7	6	5	4	3	2	1	0
BODEN	BOV	-	BORST	BOF	-	-	-
r/w	r/w	-	r/w	r/w	-	-	-

地址： A3H

复位值： 见表7-2.N79E815A/814A/8132ASFR功能描述和复位值

位	名称	描述									
7	BODEN	<p><b>欠压检测功能控制</b></p> <p>复位后，BODEN由CBODEN (CONFIG2，位-7)取反的值初始化。</p> <p>1 = 使能欠压检测。</p> <p>0 = 禁止欠压检测。</p>									
6	BOV	<p><b>欠压检测电压选择位:</b></p> <p>复位后，欠压检测由CONFIG2的CBOVB位的值初始化。</p> <p>欠压检测电压选择位:</p> <table border="1"> <thead> <tr> <th>CONFIG-位 CBOV</th> <th>SFR BOV</th> <th>欠压检测电压</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>使能欠压检测电压= 2.7V</td> </tr> <tr> <td>0</td> <td>1</td> <td>使能欠压检测电压= 3.8V</td> </tr> </tbody> </table>	CONFIG-位 CBOV	SFR BOV	欠压检测电压	1	0	使能欠压检测电压= 2.7V	0	1	使能欠压检测电压= 3.8V
CONFIG-位 CBOV	SFR BOV	欠压检测电压									
1	0	使能欠压检测电压= 2.7V									
0	1	使能欠压检测电压= 3.8V									
5	-	保留									
4	BORST	<p><b>欠压检测复位使能</b></p> <p>该位决定欠压检测事件后是否发生欠压检测复位。</p> <p>0 = 当VDD下降到低于VBOD或VDD上升到高于VBOD，禁止欠压检测复位。当VDD下降到VBOD以下，芯片声明BOF。</p> <p>1 =当VDD下降到低于VBOD或VDD上升到高于VBOD，使能欠压检测复位。</p>									
3	BOF	<p><b>欠压检测标志</b></p> <p>当VDD下降到低于VBOD或VDD上升到高于VBOD后，该标志将由硬件置1。如果EBOD (IE.5) 和EA (IE.7) 都置位，就会产生一个欠压检测中断请求。该位必须由软件清零。</p>									
2	-	<b>必须设置为0。</b>									
1	-	保留									
0	-	保留									

### 24.3 复位引脚复位

RST复位引脚是硬件为史密特触发复位输入脚。将RST引脚保持在最少两个机器周期的低电平，以确保能检测到有效的硬件复位信号，就完成一次硬件复位动作。复位电路同步于内部复位信号。因此复位是同步动作，要求时钟在此期间运行以引起外部复位。

一旦芯片满足复位条件，只要RST引脚上电平为1就会保持。在RST撤除低电平之后，CPU将在两个机器周期内退出复位状态，并从0000H开始执行代码。对RST引脚上的复位来说，没有与之配套的标志位。但是由于其他复位源都有相应的标志位存在，那么当这些复位标志位清零时，可以将外部复位认为是默认的复位。

如果CPU在掉电模式下，有RST引脚复位时，触发硬件复位的方法略有不同。因为掉电模式停止系统时钟，复位信号将同步引起系统时钟恢复。在系统时钟稳定后，CPU将进入复位状态，然后退出并从0000H地址开始执行应用程序。

**注意：因为内部上拉电阻约 200K欧姆，所以该脚不可悬空，要求在复位脚必须接上100欧姆电阻和10Uf电容。**

### 24.4 看门狗定时器复位

看门狗定时器是一个带可编程溢出时间间隔的自由运行的定时器。用户可以在任何时候清除看门狗定时器，使它重新开始计数。当选择的溢出时间间隔发生溢出后，看门狗定时器将直接复位系统。其复位条件通过硬件保持两个机器周期。撤出复位后，芯片从0000H开始运行。

一旦由看门狗定时器引起复位，看门狗定时器复位标志WDTRF (WDCON0.3)将置位。除上电复位外该位保持不变，用户可以通过软件清WDTRF。

#### WDCON0 – 看门狗定时器控制位0 (时控保护)

7	6	5	4	3	2	1	0
WDTEN	WDCLR	WDTF	WIDPD	WDTRF	WPS2	WPS1	WPS0
r/w	w	-	r/w	r/w	r/w	r/w	r/w

地址：D8H

复位值：见表7-2.N79E815A/814A/8132ASFR功能描述和复位值

位	名称	描述
3	WDTRF	看门狗复位标志 WDT 复位标志。当MCU复位时，该位由硬件置位。该位由硬件清零。 如果EWRST =0，中断标志WDTRF不会被硬件置位，MCU将立即复位自身 如果EWRST =1,中断标志WDTRF将由硬件置位，MCU跳到WDT中断服务程序， 如果使能WDT中断，MCU不会复位自身直到512个CPU时钟之后。换句话说，在此条件下，用户也需要在这512个CPU时钟之内清WDT计数器（向WDCLR位写‘1’），或MCU也会在512个CPU时钟之后复位自身



**WDCON1 – 看门狗定时器控制位0 (时控保护)**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	EWRST
-	-	-	-	-	-	-	r/w

地址: ABH

复位值: 0000 0000B

位	名称	描述
0	EWRST	看门狗复位 0: 关闭看门狗复位功能 1: 打开看门狗复位功能

**24.5 软件复位**

N79E815A/814A/8132A 增强了软件复位功能。允许应用程序以软件方式复位整个系统。这对于ISP动作结束后非常有用。例如，如果 LDR0M更新APROM，ISP 完成且APROM 中代码已更新，软件复位可使CPU 从 APROM 中启动以检查 APROM 中代码。写 1 到 SWRST (CHPCON.7) 触发软件复位。注该位为TA 保护。见下面例程。

**CHPCON – 芯片控制(时控保护)**

7	6	5	4	3	2	1	0
SWRST	ISPF	LDUEN	-	-	-	BS	ISPEN
w	r/w	r/w	-	-	-	r/w	r/w

地址: 9FH

复位值: 见 表7-2.N79E815A/814A/8132ASFR功能描述和复位值

位	名称	描述
7	SWRST	软件复位 设置该位为逻辑1将引起软件复位。复位完成后由硬件自动清零。

软件例程如下:

```

CLR EA           ; If any interrupt is enabled, disable temporarily
MOV TA, #0Aah   ; TA protection.
MOV TA, #55h    ;
ANL CHPCON, #0FDh ; BS = 0, reset to APROM.
MOV TA, #0Aah
MOV TA, #55h
ORL CHPCON, #80h ; Software reset

```

## 24.6 启动选项

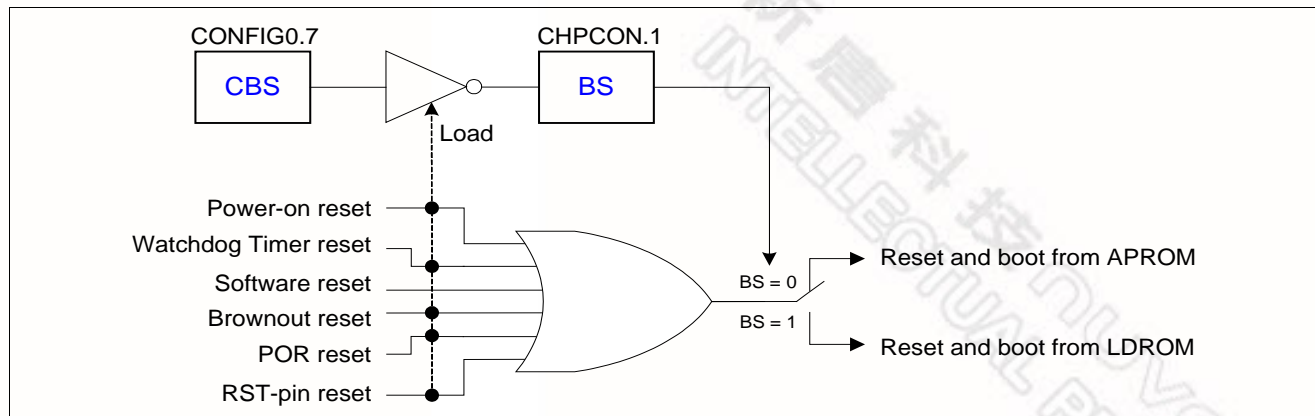


图24-1. 启动选项框图

N79E815A/814A/8132A 提供给用户灵活的启动选择以适用于不同应用。CHPCON.1的BS位用于决定复位后CPU从APROM 还是LDROM 中启动。复位后，如果BS = 0，CPU从APPROM 中启动。反之CPU 从 LDROM 中启动。

### 配置位0

7	6	5	4	3	2	1	0
CBS	-	-	-	-	-	LOCK	DFEN
r/w	-	-	-	-	-	r/w	r/w

未编程值：1111 1111B

位	名称	描述
7	CBS	<b>CONFIG 启动选择</b> 该位决定复位后MCU从哪个区块启动（除软件复位外）。 1 = 复位后MCU由APROM启动（除软件复位外）。 0 = 复位后MCU由LDROM启动（除软件复位外）。

## CHPCON – 芯片控制 (TA 保护)

7	6	5	4	3	2	1	0
SWRST	ISPF	LDUEN	-	-	-	BS <sup>[1]</sup>	ISPEN
w	r/w	r/w	-	-	-	r/w	r/w

地址：9FH

复位值：见表7-2.N79E815A/814A/8132ASFR功能描述和复位值

位	名称	描述
1	BS	<p><b>启动选择</b></p> <p>读写该位的不同方式。</p> <p>写： 定义复位后MCU由哪块启动。 0 = 下一次由APROM启动。 1 = 下一次由LDROM启动。</p> <p>读： 定义前次复位后由哪块启动。 0 = 前次由APROM启动。 1 = 前次由LDROM启动。</p>

[1] 注该位由复位后（除软件复位外）CONFIG0.7的CBS位取反值初始化。软件复位后保持不变。

**注CPU从所有复位状态释放后，硬件将检查BS位以决定是由APROM还是LDROM启动。**

## 24.7 复位状态

复位状态不会影响片上RAM。复位期间，RAM中的数据保留。注如果V<sub>DD</sub>下降到近1.2V时，RAM中的数据可能会丢失。这是RAM保存所需的最小电压。因此，在上电复位后，在电源失效的状态下，RAM中的内容将不确定。如果电源下降到数据保存的最小电压以下，RAM的内容也将丢失。

复位后，除一些受不同复位事件影响的寄存器外，大多数寄存器回到初始值。见表7-2.N79E815A/814A/8132ASFR功能描述和复位值对于所有寄存器的初始状态，一些特殊功能寄存器初始值取决于不同复位源。参见

表24-2。只要复位条件应用时，程序计数器为0000H且保持。注堆栈指针复位为07H，因此，在复位期间，即使RAM内容不被修改，堆中内容也可能丢失。

复位后，中断与定时器都禁止。如果复位源是上电复位，看门狗定时器也被禁止。I/O口寄存器写入FFH，使这些端口处于高电平状态。

表24-2不同复位引起的寄存器初始状态

	上电复位	看门狗复位	软件/外部复位	欠压检测复位	是否受编程 时控保护
WDCON0 (D8H)	C000 0000B b7(ENWDT)= /CENWDT(CONFIG3.7)	C0Uu 1UUUB	C0UU UUUUB		是
WDCON1(ABH)	0000 0000B				是
ISPTRG (A4H)	XXXX XXX0B				是
PMCR (A3H)	CXCC 10XXB b[7:4]=CONFIG2	UXUU U0XXB		UXUU 10XXB	是
CHPCON (9FH)	0000 00C0B b1(BS)=CBS	000X XU00B			是
SHBDA (9CH)	CONFIG1	不改变			是
PCON (87H)	0001 000b	00uu 0000b	00uu 0000b (软件复位/外部复位)	00uu 0000b	否

注：写入AAH和55H，必须发生在3个机器周期之内，此过程中必须禁止中断以避免延迟。

## 25 配置位(CONFIG)

N79E815A/814A/8132A具有硬件配置位，设定这些配置位可用于安全位，系统时钟位等等。这些硬件配置位可通过编程器/烧录器或ISP 来配置。N79E815A/814A/8132A具有四个配置位，配置位0~3。有些特定的配置位定义的功能也可以通过特定寄存器位重新配置。因此，需要加载这睦配置位到相应的寄存器位。这些加载发生在复位之后。(软件复位会加载除CONFIG0.7的CBS位以外的所有配置位) 这些寄存器位可以持续通过用户的软件控制。其他复位将不改变这些寄存器位的值。

**注配置位标记为“-”的位应该保证不被编程器改写。**

### 25.1 配置位0

7	6	5	4	3	2	1	0
CBS	-	-	-	-	-	LOCK	DFEN
r/w	-	-	-	-	-	r/w	r/w

未编程值： 1111 1111B

位	名称	描述
7	CBS	<b>配置启动选项</b> 该位定义除软件复位外的所有复位后MCU从哪个区块启动。 1 = 除软件复位外的所有复位后MCU从APROM启动。 0 = 除软件复位外的所有复位后MCU从LDROM启动。
6:2	-	保留。
1	LOCK	<b>芯片加密使能</b> 1 = 芯片不加密。所有APROM，LDROM和数据内存都不加密。其内容可以通过编程器/烧录器读出。 0 = 芯片加密。APROM，LDROM和数据内存都加密。其内容通过编程器/烧录器读出时将变成FFH。 注配置字节通常不加密，可以读出。一旦加密后，配置字节不能被单独擦除或编程，唯有禁止芯片加密并擦除整颗芯片。如果执行这个过程，APROM，LDROM，数据内存及其他配置位都将被擦除，如果芯片加密，不能修改ISP功能。
0	DFEN	<b>数据内存使能</b> (仅适用于N79E815A) 1 = 没有数据内存空间。APROM的大小为16k 字节。 0 = 有数据内存。数据内存的大小依据SHBDA的设置与APROM共享16k空间。

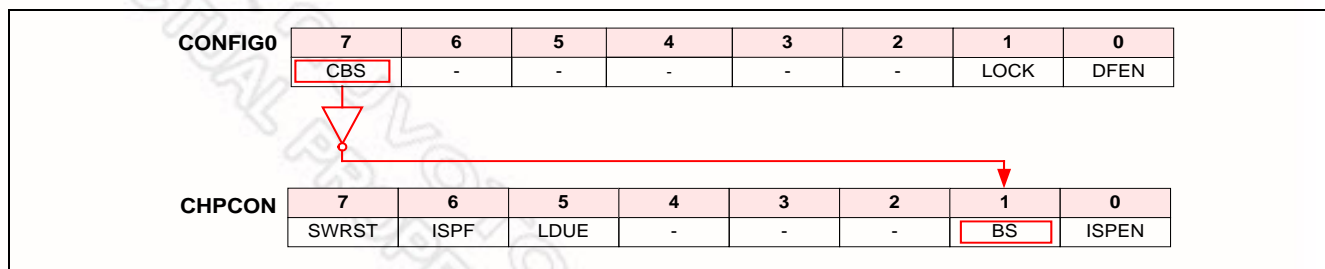


图25-1. 配置位0 复位重载（软件复位出外）

## 25.2 配置位1 (仅适用于N79E815A)

7	6	5	4	3	2	1	0
CHBDA[7:0] <sup>[1]</sup>							
r/w							

未编程值： 1111 1111B

位	名称	描述
7:0	CHBDA[7:0]	配置数据内存开始地址的高字节 该字节仅当DFEN (CONFIG0.0)为0时有效。用于决定数据内存的开始地址。

**注**如果CHBDA设置为00H，将没有APROM空间。CPU将执行内部程序最小内存的代码，值为(256字节)。

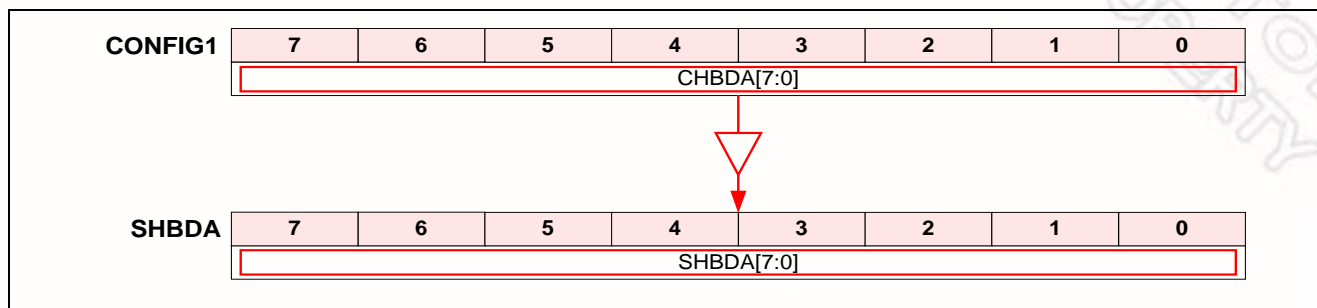


图25-2. 配置位1 复位重载

## 25.3 配置位2

7	6	5	4	3	2	1	0
CBODEN	CBOV	-	CBORST	-	-	-	-
r/w	r/w	-	r/w	-	-	-	-

未编程值: 1111 1111B

位	名称	描述									
7	CBODEN	配置欠压检测使能 1 = 禁止欠压检测 0 = 使能欠压检测 复位后, BODEN由CBODEN (CONFIG2, 位-7)的取反值初始化。									
6	CBOV	配置欠压检测电压选择 该位选择欠压检测电压 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>CONFIG-位 CBOV</th> <th>SFR BOV</th> <th>欠压检测电压</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>使能欠压检测= 2.7V</td> </tr> <tr> <td>0</td> <td>1</td> <td>使能欠压检测= 3.8V</td> </tr> </tbody> </table>	CONFIG-位 CBOV	SFR BOV	欠压检测电压	1	0	使能欠压检测= 2.7V	0	1	使能欠压检测= 3.8V
CONFIG-位 CBOV	SFR BOV	欠压检测电压									
1	0	使能欠压检测= 2.7V									
0	1	使能欠压检测= 3.8V									
5	-	保留.									
4	CBORST	配置欠压检测复位使能 该位决定欠压检测事件后是否发生欠压检测复位。 1 =当VDD下降到低于VBOD或VDD上升到高于VBOD, 使能欠压检测复位。 0 =当VDD下降到低于VBOD或VDD上升到高于VBOD, 禁止欠压检测复位。									
3:0	-	保留.									

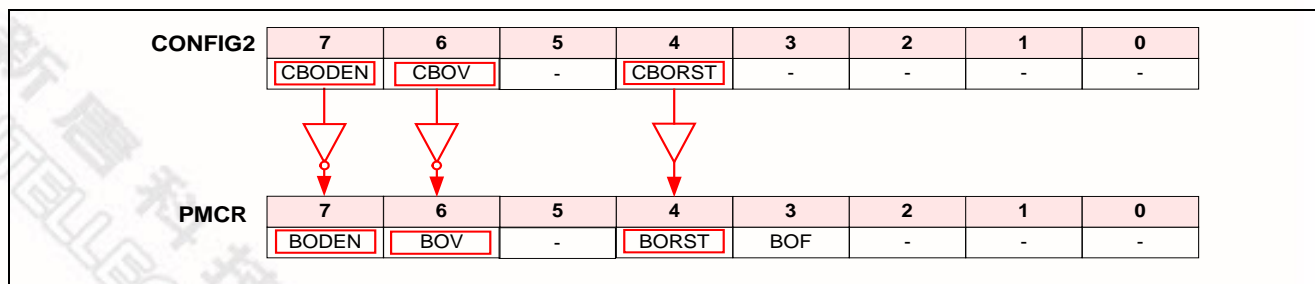


图25-3. 配置位2复位重载

## 25.4 配置位3

7	6	5	4	3	2	1	0
CWDTEN	-	-	CKF	OSCFS	-	FOSC1	FOSC0
r/w	-	-	r/w	r/w	-	r/w	r/w

未编程值： 1111 1111B

位	名称	描述									
7	CWDTEN	<b>配置看门狗定时器使能</b> 1 = 所有复位后禁止看门狗定时器。 0 = 所有复位后使能看门狗定时器。 其他复位后WDTEN由CWDTEN (CONFIG3, 位-7)的取反位初始化。									
6	-	保留.									
5	-	保留.									
4	CKF	<b>时钟滤波器使能</b> 1 = 使能时钟滤波器。可提高噪声抗干扰能力和增强EMC。 0 = 禁止时钟滤波器。									
3	OSCFS	<b>选择内部RC振荡频率</b> 1 = 如果内部RC振荡模式被占用，则选择22.1184MHz 作为时钟系统。绕过2分频内部振荡器，选择22.1184MHz输出作为时钟系统源。 0 =如果内部RC振荡模式被占用，选择11.0592MHz 作为时钟系统。选择2分频内部RC。内部振荡器配置为11.0592MHz 输出用途时钟系统。									
2	-	保留.									
1	FOSC1	<b>振荡器选择位</b>									
0	FOSC0	芯片时钟源选择。见下表： <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>(FOSC1, FOSC0)</th> <th>芯片时钟源</th> </tr> </thead> <tbody> <tr> <td>(1, 1)</td> <td>内部RC振荡器</td> </tr> <tr> <td>(1, 0)</td> <td>-保留位</td> </tr> <tr> <td>(0, 0)</td> <td rowspan="2">外部晶振, 4MHz ~ 24MHz</td> </tr> <tr> <td>(0, 1)</td> </tr> </tbody> </table>	(FOSC1, FOSC0)	芯片时钟源	(1, 1)	内部RC振荡器	(1, 0)	-保留位	(0, 0)	外部晶振, 4MHz ~ 24MHz	(0, 1)
(FOSC1, FOSC0)	芯片时钟源										
(1, 1)	内部RC振荡器										
(1, 0)	-保留位										
(0, 0)	外部晶振, 4MHz ~ 24MHz										
(0, 1)											

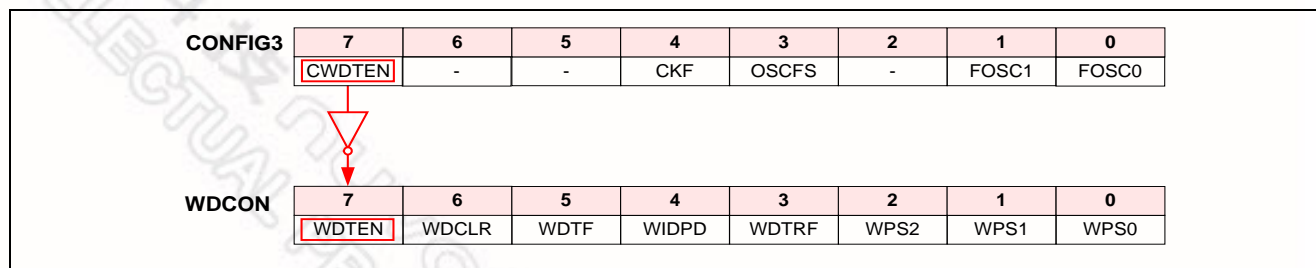


图25-4. 配置位2复位重载



## 26 指令集

N79E815A/814A/8132A 执行所有标准8051的指令集。所有指令代码为 8-位 OPCODE。单字节来源于程序存储器。OPCODE 通过CPU译码。决定系统的工作情况以及从存储器中运作数据。如果没有其它数据请求，为单字节指令。在一些应用中，需要更多的数据，使用2或3字节指令。

[表26-1](#)列出所有指令。指令设定和寻址如下。

Rn (n = 0~7) 寄存器 R0~R7 为当前选择寄存器区域。

8-位 内部数据地址。可作为内部 RAM 地址 (0~127) 或 SFR (I/O, 控制寄存器, 状态寄存器等(128~255))。

@Ri (i = 0, 1)通过R0或R1间接寻址8-位内部RAM 区域 (0~255)。

#data 指令包括8-位常量。

#data16 指令包括16-位常量。

Addr16 16-位目的地址。使用LCALL和LJMP。分支可达16k字节程序空间任何位置。

Addr11 11-位目的地址。使用ACALL和AJMP。分支达2k字节程序内存。

Rel 带符号(2's 互补) 8-位偏移字节。使用SJMP和条件分支。范围为-128到+127字节。

Bit 对内部数据RAM或寄存器直接寻址。

表26-1. N79E815A/814A/8132A指令集

指令	机器码	占用字节	时钟周期	N79E815A/814A/8132A 与标准80C51的速率对比
NOP	00	1	4	3.0
ADD A, Rn	28~2F	1	4	3.0
ADD A, @Ri	26, 27	1	4	3.0
ADD A, direct	25	2	8	1.5
ADD A, #data	24	2	8	1.5
ADDC A, Rn	38~3F	1	4	3.0
ADDC A, @Ri	36, 37	1	4	3.0
ADDC A, direct	35	2	8	1.5
ADDC A, #data	34	2	8	1.5
SUBB A, Rn	98~9F	1	4	3.0
SUBB A, @Ri	96, 97	1	4	3.0
SUBB A, direct	95	2	8	1.5
SUBB A, #data	94	2	8	1.5
INC A	04	1	4	3.0
INC Rn	08~0F	1	4	3.0
INC @Ri	06, 07	1	4	3.0

表26-1. N79E815A/814A/8132A指令集

指令	机器码	占用字节	时钟周期	N79E815A/814A/8132A 与标准80C51的速率对比
INC direct	05	2	8	1.5
INC DPTR	A3	1	8	3.0
DEC A	14	1	4	3.0
DEC Rn	18~1F	1	4	3.0
DEC @Ri	16, 17	1	4	3.0
DEC direct	15	2	8	1.5
DEC DPTR	A5	1	8	-
MUL AB	A4	1	20	2.4
DIV AB	84	1	20	2.4
DA A	D4	1	4	3.0
ANL A, Rn	58~5F	1	4	3.0
ANL A, @Ri	56, 57	1	4	3.0
ANL A, direct	55	2	8	1.5
ANL A, #data	54	2	8	1.5
ANL direct, A	52	2	8	1.5
ANL direct, #data	53	3	12	2.0
ORL A, Rn	48~4F	1	4	3.0
ORL A, @Ri	46, 47	1	4	3.0
ORL A, direct	45	2	8	1.5
ORL A, #data	44	2	8	1.5
ORL direct, A	42	2	8	1.5
ORL direct, #data	43	3	12	2.0
XRL A, Rn	68~6F	1	4	3.0
XRL A, @Ri	66, 67	1	4	3.0
XRL A, direct	65	2	8	1.5
XRL A, #data	64	2	8	1.5
XRL direct, A	62	2	8	1.5
XRL direct, #data	63	3	12	2.0
CLR A	E4	1	4	3.0
CPL A	F4	1	4	3.0
RL A	23	1	4	3.0
RLC A	33	1	4	3.0
RR A	03	1	4	3.0
RRC A	13	1	4	3.0
SWAP A	C4	1	4	3.0
MOV A, Rn	E8~EF	1	4	3.0
MOV A, @Ri	E6, E7	1	4	3.0
MOV A, direct	E5	2	8	1.5
MOV A, #data	74	2	8	1.5
MOV Rn, A	F8~FF	1	4	3.0
MOV Rn, direct	A8~AF	2	8	3.0

表26-1. N79E815A/814A/8132A指令集

指令	机器码	占用字节	时钟周期	N79E815A/814A/8132A 与标准80C51的速率对比
MOV Rn, #data	78~7F	2	8	1.5
MOV @Ri, A	F6, F7	1	4	3.0
MOV @Ri, direct	A6, A7	2	8	3.0
MOV @Ri, #data	76, 77	2	8	1.5
MOV direct, A	F5	2	8	1.5
MOV direct, Rn	88~8F	2	8	3.0
MOV direct, @Ri	86, 87	2	8	3.0
MOV direct, direct	85	3	12	2.0
MOV direct, #data	75	3	12	2.0
MOV DPTR, #data16	90	3	12	2.0
MOVC A, @A+DPTR	93	1	8	3.0
MOVC A, @A+PC	83	1	8	3.0
MOVX A, @Ri <sup>[1]</sup>	E2, E3	1	8	3.0
MOVX A, @DPTR <sup>[1]</sup>	E0	1	8	3.0
MOVX @Ri, A <sup>[1]</sup>	F2, F3	1	8	3.0
MOVX @DPTR, A <sup>[1]</sup>	F0	1	8	3.0
PUSH direct	C0	2	8	3.0
POP direct	D0	2	8	3.0
XCH A, Rn	C8~CF	1	4	3.0
XCH A, @Ri	C6, C7	1	4	3.0
XCH A, direct	C5	2	8	1.5
XCHD A, @Ri	D6, D7	1	4	3.0
CLR C	C3	1	4	3.0
CLR bit	C2	2	8	1.5
SETB C	D3	1	4	3.0
SETB bit	D2	2	8	1.5
CPL C	B3	1	4	3.0
CPL bit	B2	2	8	1.5
ANL C, bit	82	2	8	3.0
ANL C, /bit	B0	2	8	3.0
ORL C, bit	72	2	8	3.0
ORL C, /bit	A0	2	8	3.0
MOV C, bit	A2	2	8	1.5
MOV bit, C	92	2	8	3.0
ACALL addr11	11, 31, 51, 71, 91, B1, D1, F1 <sup>[2]</sup>	2	12	2.0
LCALL addr16	12	3	16	1.5
RET	22	1	8	3.0
RETI	32	1	8	3.0
AJMP addr11	01, 21, 41, 61, 81,	2	12	2.0

表26-1. N79E815A/814A/8132A指令集

指令	机器码	占用字节	时钟周期	N79E815A/814A/8132A 与标准80C51的速率对比
	A1, C1, E1			
LJMP addr16	02	3	16	1.5
JMP @A+DPTR	73	1	8	3.0
SJMP rel	80	2	12	2.0
JZ rel	60	2	12	2.0
JNZ rel	70	2	12	2.0
JC rel	40	2	12	2.0
JNC rel	50	2	12	2.0
JB bit, rel	20	3	16	1.5
JNB bit, rel	30	3	16	1.5
JBC bit, rel	10	3	16	1.5
CJNE A, direct, rel	B5	3	16	1.5
CJNE A, #data, rel	B4	3	16	1.5
CJNE @Ri, #data, rel	B6, B7	3	16	1.5
CJNE Rn, #data, rel	B8~BF	3	16	1.5
DJNZ Rn, rel	D8~DF	2	12	2.0
DJNZ direct, rel	D5	3	16	1.5

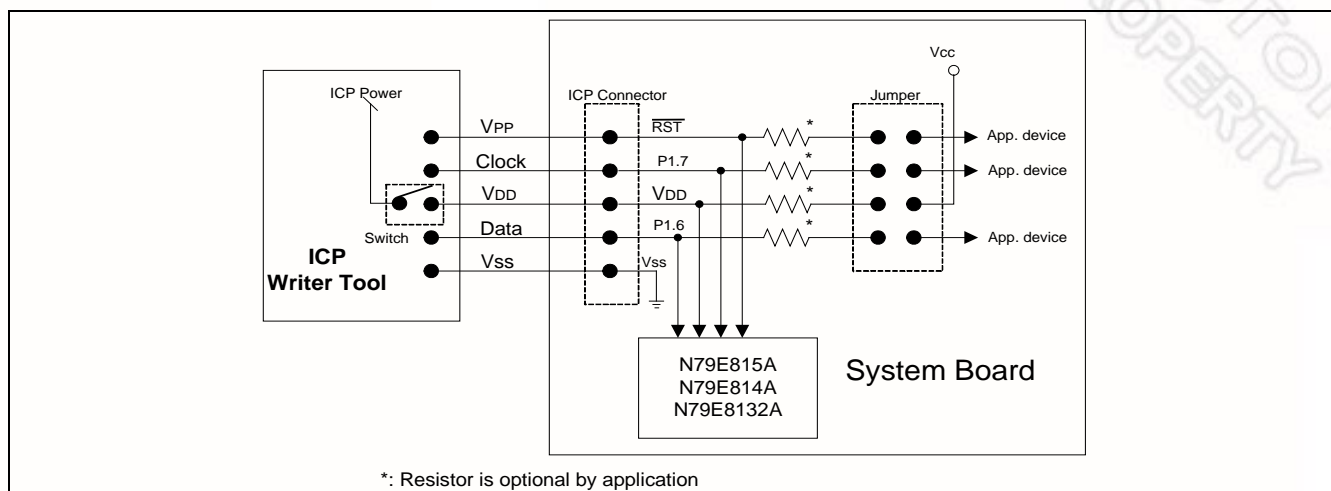
[1] 11位地址[A10:A8]的最高三位决定ACALL hex 码。代码为[A10,A9,A8,1,0,0,0,1]。

[2] 11位地址[A10:A8]的最高三位决定AJMP hex 码。代码为 [A10,A9,A8,0,0,0,0,1]。

## 27 在电路编程 (ICP)

ICP (在电路编程) 模式是另一种访问存储器EPROM的方式, 仅需要3个引脚执行ICP功能, 一个是/RST输入引脚, 在ICP工作期间必须拉到GND, 一个是时钟输入, 与P1.7复用, 接收外部器件的串行时钟。另一个是数据I/O引脚, 与P1.6复用, 外部ICP编程工作通过P1.6在P1.7脚上的同步时钟下将数据送入N79E815A/814A/8132A的存储器 EPROM。

进入ICP编程模式时, 所有引脚将设置为准双向模式, 输出为“1”。N79E815A/814A/8132A支持存储器 EPROM (16K/8K 字节APROM EPROM), 数据内存 (每页128字节) 和LDROM的编程。用户可选择编程APROM, 数据内存和LDROM。



### 注:

1. 在使用 ICP 更新代码时, /RST, P1.6 和 P1.7 必须断开与系统板负载的连接。
2. 在 ICP 编程结束后, 建议关闭系统电源移去 ICP 工具, 然后在接上电源。
3. 建议客户连续执行擦除和编辑配置位两个步骤, 不要间断。

## 28 电气特性

### 28.1 最大绝对额定值

参数	范围	单位
工作温度	-40 to +85	°C
存储温度	-55 to +150	°C
$V_{DD}$ 到 $V_{SS}$ 的电压	-0.3 to +6.5	V
其他引脚到 $V_{SS}$ 的电压	-0.3 to ( $V_{DD}+0.3$ )	V

用户于使用中达到或超过“绝对最大额定值”中所列值有可能会造成芯片永久性损坏。本规格仅采用芯片在该条件下测试完成，如超过本规范部分不做保证。芯片长期处于绝对最大额定值条件下，可能会影响器件的可靠性。

### 28.2 直流电气特性

表28-1. 工作电压

参数	符号	最小值	典型值	最大值	条件	单位
工作电压	$V_{DD}$	4.5		5.5	$F_{osc} = 4 \text{ MHz} \sim 24 \text{ MHz}$	V
		2.4		5.5	$F_{osc} = 4 \text{ MHz} \sim 12 \text{ MHz}$	
		3.0		5.5	内部 RC 22.1184MHz 晶振	
		2.4		5.5	内部 RC 11.0592MHz 晶振	
ISP 工作电压	$V_{DD}$	3.0		5.5	内部 RC 22.1184MHz 晶振 $F_{osc} = 4 \text{ MHz} \sim 12 \text{ MHz}$	V
		4.5		5.5	$F_{osc} = 4 \text{ MHz} \sim 24 \text{ MHz}$	V

表28-2. 直流电气特性表

(除非特别说明，则本表内测试条件为  $V_{DD}-V_{SS} = 2.4\sim 5.5V$ ,  $T_A = -40\sim 85^\circ C$ )

符号	参数	测试条件	最小值	典型值	最大值	单位
$S_{VDD}$	确保内部上电复位有效的 $V_{DD}$ 上升速率	细节参数请参考 上电复位章节	0.05 <sup>[7]</sup>	-	-	V/ms
$V_{IL}$	上电低电压 (标准端口 TTL 输入状态)	$2.4 < V_{DD} < 5.5V$	-0.5		$0.2V_{DD}-0.1$	V
$V_{IL1}$	输入低电压 (标准端口史密特输入状态)	$2.4 < V_{DD} < 5.5V$	-0.5		$0.3V_{DD}$	V
$V_{IL2}$	输入低电压 (/RST, XTAL1)	$2.4 < V_{DD} < 5.5V$	-0.5		$0.2V_{DD}-0.1$	V
$V_{IH}$	输入高电压 (标准端口 TTL 输入状态)	$2.4 < V_{DD} < 5.5V$	$0.2V_{DD}+0.9$		$V_{DD}+0.5$	V

表28-2. 直流电气特性表

(除非特别说明, 则本表内测试条件为  $V_{DD}-V_{SS} = 2.4\sim 5.5V$ ,  $T_A = -40\sim 85^{\circ}C$ )

符号	参数	测试条件	最小值	典型值	最大值	单位
$V_{IH1}$	输入高电压 (标准端口史密特输入状态)	$2.4 < V_{DD} < 5.5V$	$0.7V_{DD}$		$V_{DD}+0.5$	V
$V_{IH2}$	输入高电压 (/RST, XTAL1)	$2.4 < V_{DD} < 5.5V$	$0.7V_{DD}$		$V_{DD}+0.5$	V
$V_{OL}$	输出低电压 (标准端口 P0,P2, 除输入外的所有状态)	$V_{DD}=4.5V,$ $I_{OL} = 20\text{ mA}^{[3],[4]}$			0.45	V
		$V_{DD}=3.0V,$ $I_{OL} = 14\text{ mA}^{[3],[4]}$			0.45	V
		$V_{DD}=2.4V,$ $I_{OL} = 10\text{ mA}^{[3],[4]}$			0.45	V
$V_{OL1}$	输出低电压 (端口P10, P11, P14, P16, P17 除输入外的所有状态)	$V_{DD}=4.5V,$ $I_{OL} = 38\text{ mA}^{[3],[4]}$			0.45	V
		$V_{DD}=3.0V,$ $I_{OL} = 27\text{ mA}^{[3],[4]}$			0.45	V
		$V_{DD}=2.4V,$ $I_{OL} = 20\text{ mA}^{[3],[4]}$			0.45	V
$V_{OH}$	输出高电压 (标准端口, 准双向模式)	$V_{DD}=4.5V$ $I_{OH} = -380\text{ }\mu\text{A}^{[4]}$	2.4			V
		$V_{DD}=3.0V$ $I_{OH} = -90\text{ }\mu\text{A}^{[4]}$	2.4			V
		$V_{DD}=2.4V$ $I_{OH} = -48\text{ }\mu\text{A}^{[4]}$	2.0			V
$V_{OH1}$	输出高电压 (标准端口, 推挽输出模式)	$V_{DD}=4.5V$ $I_{OH} = -28.0\text{ mA}^{[3],[4]}$	2.4			V
		$V_{DD}=3.0V$ $I_{OH} = -7\text{ mA}^{[3],[4]}$	2.4			V
		$V_{DD}=2.4V$ $I_{OH} = -3.5\text{ mA}^{[3],[4]}$	2.0			V
$I_{IL}$	逻辑0电位输入电流 (标准端口, 准双向模式)	$V_{DD}=5.5V, V_{IN}=0.4V$		<b>-40 @5.5V</b>	-50	$\mu\text{A}$
$I_{TL}$	逻辑1向0转换电流 (标准端口, 准双向模式)	$V_{DD}=5.5V, V_{IN}=2.0V^{[2]}$		<b>-550 @5.5V</b>	-650	$\mu\text{A}$
$I_{LI}$	输入漏电流 (标准端口, 开漏模式或输入模式)	$0 < V_{IN} < V_{DD}$		<b>&lt;1</b>	$\pm 10$	$\mu\text{A}$
$I_{OP}$	工作电流 (正常工作模式 <sup>[5]</sup> )	XTAL 12MHz, $V_{DD}=5.0V$		<b>3.1</b>		mA
		XTAL 24MHz, $V_{DD}=5.5V$		<b>4.3</b>		mA
		XTAL 12MHz, $V_{DD}=3.3V$		<b>1.7</b>		mA
		XTAL 24MHz, $V_{DD}=3.3V$		<b>3.2</b>		mA

表28-2. 直流电气特性表

(除非特别说明, 则本表内测试条件为  $V_{DD}-V_{SS} = 2.4\sim 5.5V$ ,  $T_A = -40\sim 85^{\circ}C$ )

符号	参数	测试条件	最小值	典型值	最大值	单位
		内部晶振 22.1184MHz, $V_{DD}=5V$		2.8		mA
		内部晶振 22.1184MHz, $V_{DD}=3.3V$		2.6		mA
$I_{IDLE}$	空闲模式工作电流	XTAL 12MHz, $V_{DD}=5.0V$		2.7		mA
		XTAL 24MHz, $V_{DD}=5.5V$		3.7		mA
		XTAL 12MHz, $V_{DD}=3.3V$		1.3		mA
		XTAL 24MHz, $V_{DD}=3.3V$		2.3		mA
		内部RC晶振 22.1184MHz, $V_{DD}=5V$		2.0		mA
		内部RC晶振 22.1184MHz, $V_{DD}=3.3V$		1.8		mA
$I_{PD}$	掉电模式工作电流			<5	30	$\mu A$
	掉电模式(BOD模块打开)工作电流			100		$\mu A$
$R_{RST}$	复位脚内部上拉电阻	$2.4 < V_{DD} < 5.5V$	100		250	K $\Omega$
$V_{BOD38}$	BOD38 有效侦测电压 (温度=25 $^{\circ}C$ )		3.5	3.8	4.1	V
	BOD38 有效侦测电压 (温度=85 $^{\circ}C$ )		3.5	3.8	4.9	V
	BOD38 有效侦测电压 (温度=-40 $^{\circ}C$ )		3.0	3.8	4.1	V
$V_{BOD27}$	BOD27 有效侦测电压 (温度=25 $^{\circ}C$ )		2.5	2.7	2.9	V
	BOD27 有效侦测电压 (温度=85 $^{\circ}C$ )		2.5	2.7	3.1	V
	BOD27 有效侦测电压 (温度=-40 $^{\circ}C$ )		2.4	2.7	2.9	V

[1] 典型值不做保证。所列出的值均在室温并基于有限数量样品进行测试。

[2] 端口0至端口3在外部对其由“1”电位拉到“0”电位时会产生暂态电流, 该电流在 $V_{IN}$ 接近2V时达到最大值。

[3] 基于以下(非暂态)状态条件测试,  $I_{OL}/I_{OH}$  外部限制如下:  
P0、P2、P3 每个端口  $I_{OL}/I_{OH}$  最大值 20 mA。  
P10、P11、P14、P16、P17 端口  $I_{OL}/I_{OH}$  最大值 38mA。  
所有输出累计  $I_{OL}/I_{OH}$  最大值 100mA (即通过芯片  $V_{DD}$  的所有电流值)。  
所有输出累计  $I_{OL}/I_{OH}$  最大值 150mA (即流过芯片  $V_{SS}$  的所有电流值)。

[4] 如果  $I_{OH}$  超出表格测试条件, 则  $V_{OH}$  值将低于表格所列范围。  
如果  $I_{OL}$  超出表格测试条件, 则  $V_{OL}$  值将高于表格所列范围。



[5] 在CPU保持复位状态时测得值。

[6] 标准端口表示 做普通I/O使用，例如I/O P0, P1, P2, P3。

[7] 该参数为标称值，非直接测量取得。

其他: P1.2 和 P1.3 始终为开漏结构端口，软件无法设定为准双向或推挽输出模式。

## 28.3 交流电气特性

### 28.3.1 10位SAR-ADC标准

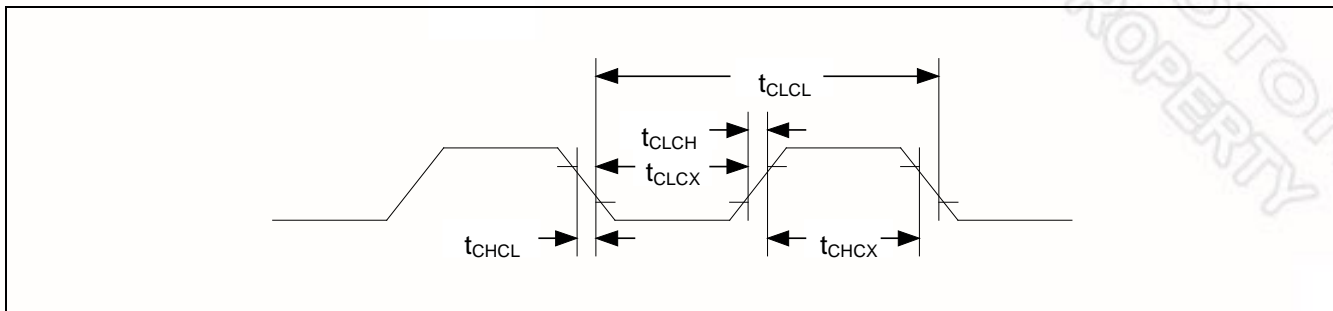
	符号	最小值	典型值	最大值	单位
工作电压	$V_{DD}$	2.7		5.5	V
精度				10	bit
转换时间			$35t_{ADC}^{[1]}$		us
采样率				150K	Hz
积分非线性误差	INL	-1		1	LSB
差分非线性误差	DNL	-1		1	LSB
增益误差	Ge	-1		1	LSB
失调误差	Ofe	-4		4	LSB
时钟频率	ADCCLK			5.25	MHz
绝对误差		-4		4	LSB
带隙基准电压	$V_{BG}$	1	1.3	1.6	V

[1]  $t_{ADC}$  ADC输入时钟周期

### 28.3.2 4~24MHz XTAL 外部晶振标准

参数	条件	最小值	典型值	最大值	单位
时钟输入频率	外部晶振源	4		24	MHz

参数	符号	最小值	典型值	最大值	单位	备注
晶振频率	$1/t_{CLCL}$	0		24	MHz	
时钟高电位持续时间	$t_{CHCX}$	20.8	-	-	Ns	
时钟低电位持续时间	$t_{CLCX}$	20.8	-	-	Ns	
时钟上升时间	$t_{CLCH}$	-	-	10	Ns	
时钟下降时间	$t_{CHCL}$	-	-	10	Ns	



注: 占空比为 50%.

### 28.3.3 内部 22.1184MHz / 11.0592MHz RC晶振标准

参数	条件	最小值	典型值	最大值	单位
中心频率			22.1184		MHz
内部晶振频率偏差	+25°C @ $V_{DD}=5V$	-1		+1	%
	+25°C @ $V_{DD}=2.7\sim 5.5V$	-3		+3	%
	-10°C~+70°C @ $V_{DD}=2.7\sim 5.5V$	-5		+5	%
	-40°C~+85°C @ $V_{DD}=2.7\sim 5.5V$	-8		+8	%

#### 28.3.4 内部 10kHz RC晶振标准

参数	条件	最小值	典型值	最大值	单位
中心频率	$V_{DD} = 2.4V \sim 5.5V$	5	10	15	kHz

## 29 抗电测干扰典型应用电路

典型应用电路如下所示。建议用户遵循封闭灰色块内的参数设计应用线路，尤其是在一个电源噪声电磁干扰较严重的环境中，可使芯片更稳定运行。如果采用内部RC振荡器作为时钟，如图增加一个0.1uF电容应增加可以获得精确的频率RC。

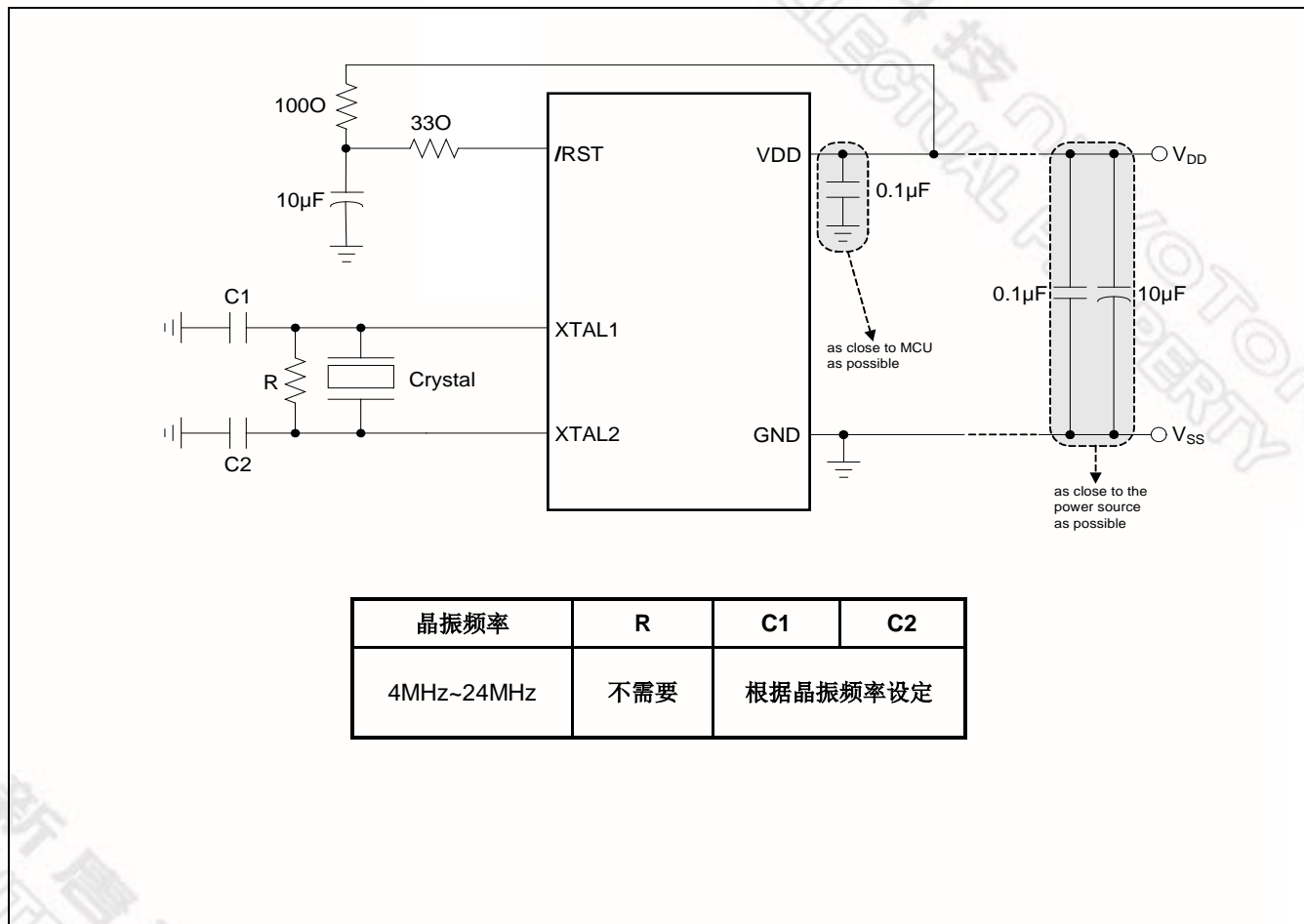
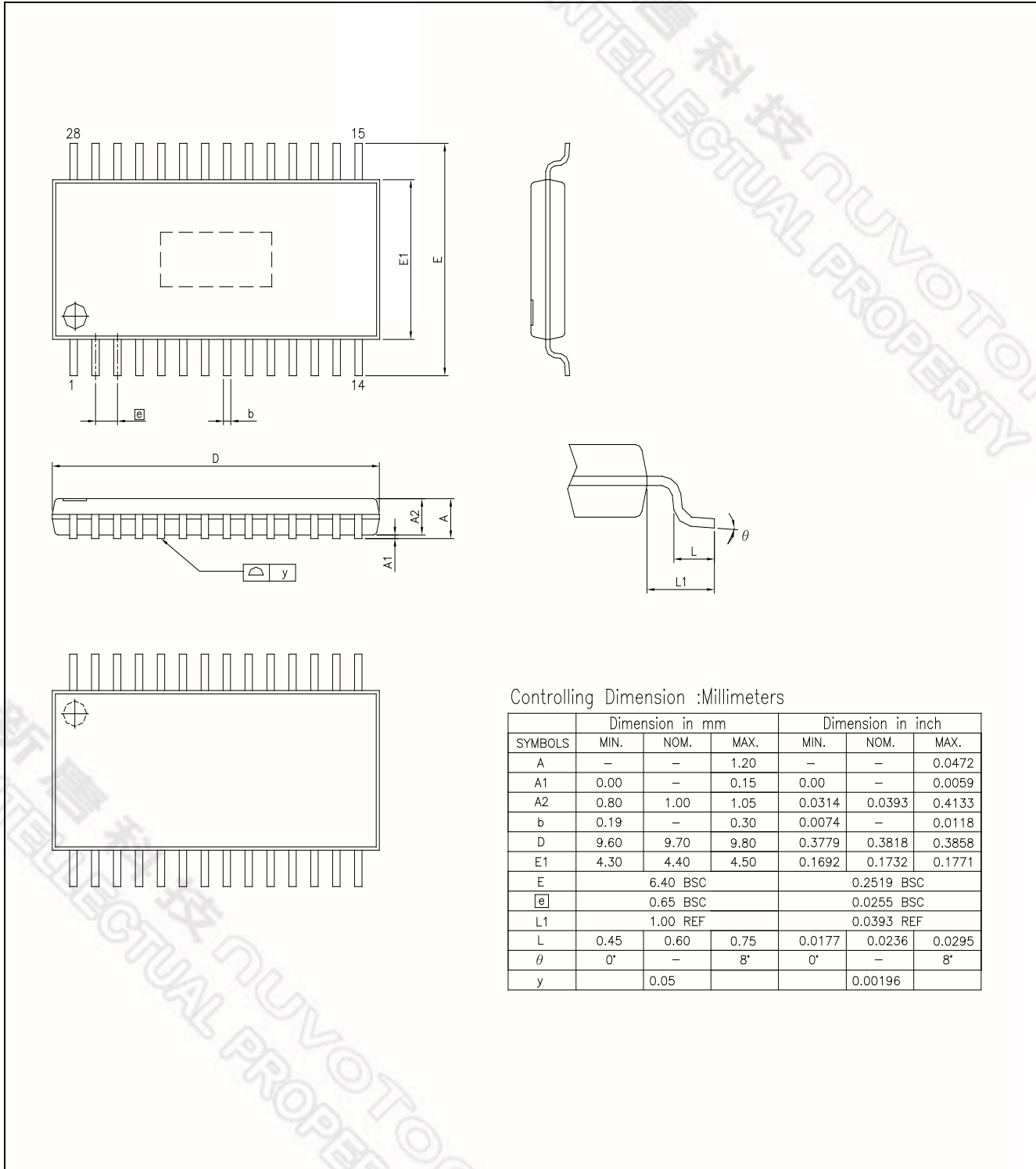


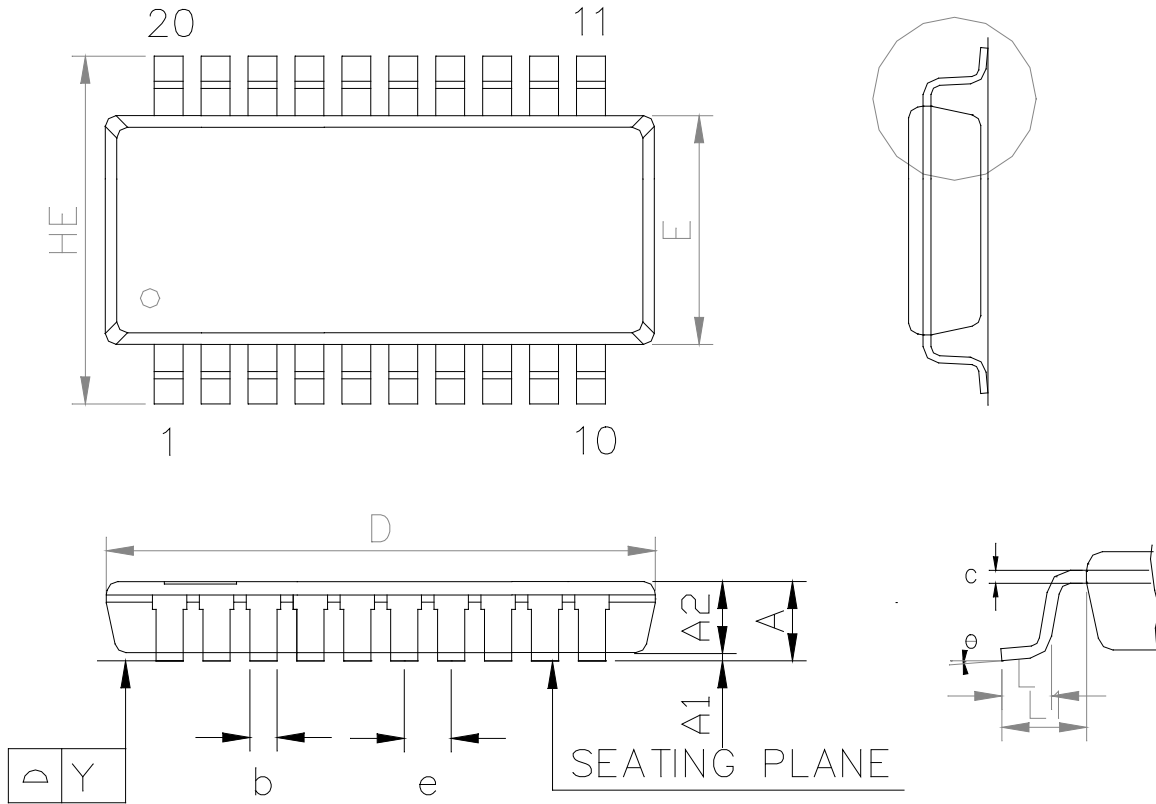
图29-1. 增加抗干扰性能建议电路

30 封装尺寸

30.1 28脚 TSSOP – 4.4X9.7mm

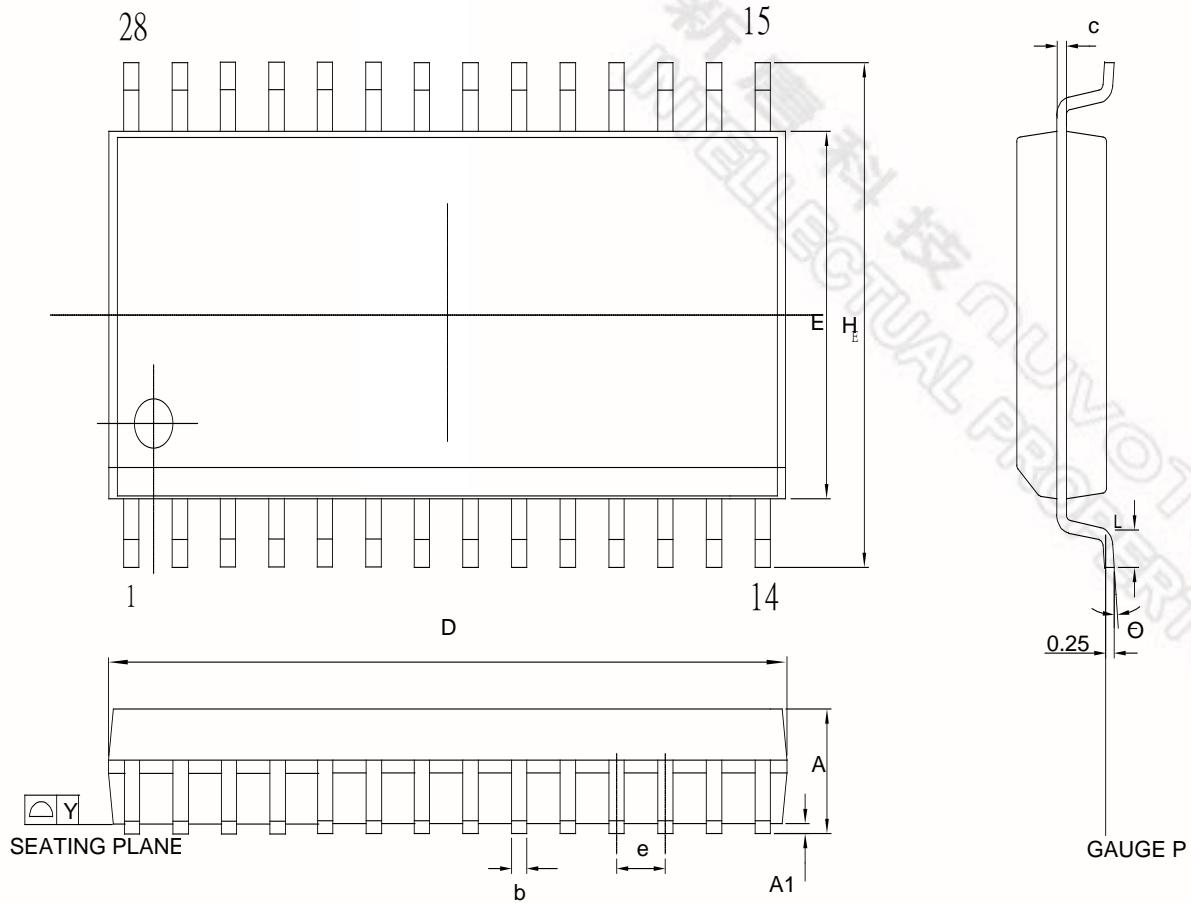


30.2 20脚 TSSOP – 4.4X6.5mm



SYMBOL	DIMENSION (MM)			DIMENSION (INCH)		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	-	-	1,20	-	-	0,047
A1	0,05	-	0,15	0,002	-	0,006
A2	0,80	0,90	1,05	0,031	0,035	0,041
E	4,30	4,40	4,50	0,169	0,173	0,177
HE	6,40 BSC			0,252 BSC		
D	6,40	6,50	6,60	0,252	0,256	0,260
L	0,50	0,60	0,75	0,020	0,024	0,030
L1	1,00 REF			0,039 REF		
b	0,19	-	0,30	0,007	-	0,012
e	0,65 BSC			0,026 BSC		
c	0,09	-	0,20	0,004	-	0,008
θ	0°	-	8°	0°	-	8°
Y	0,10 BASIC			0,004 BASIC		

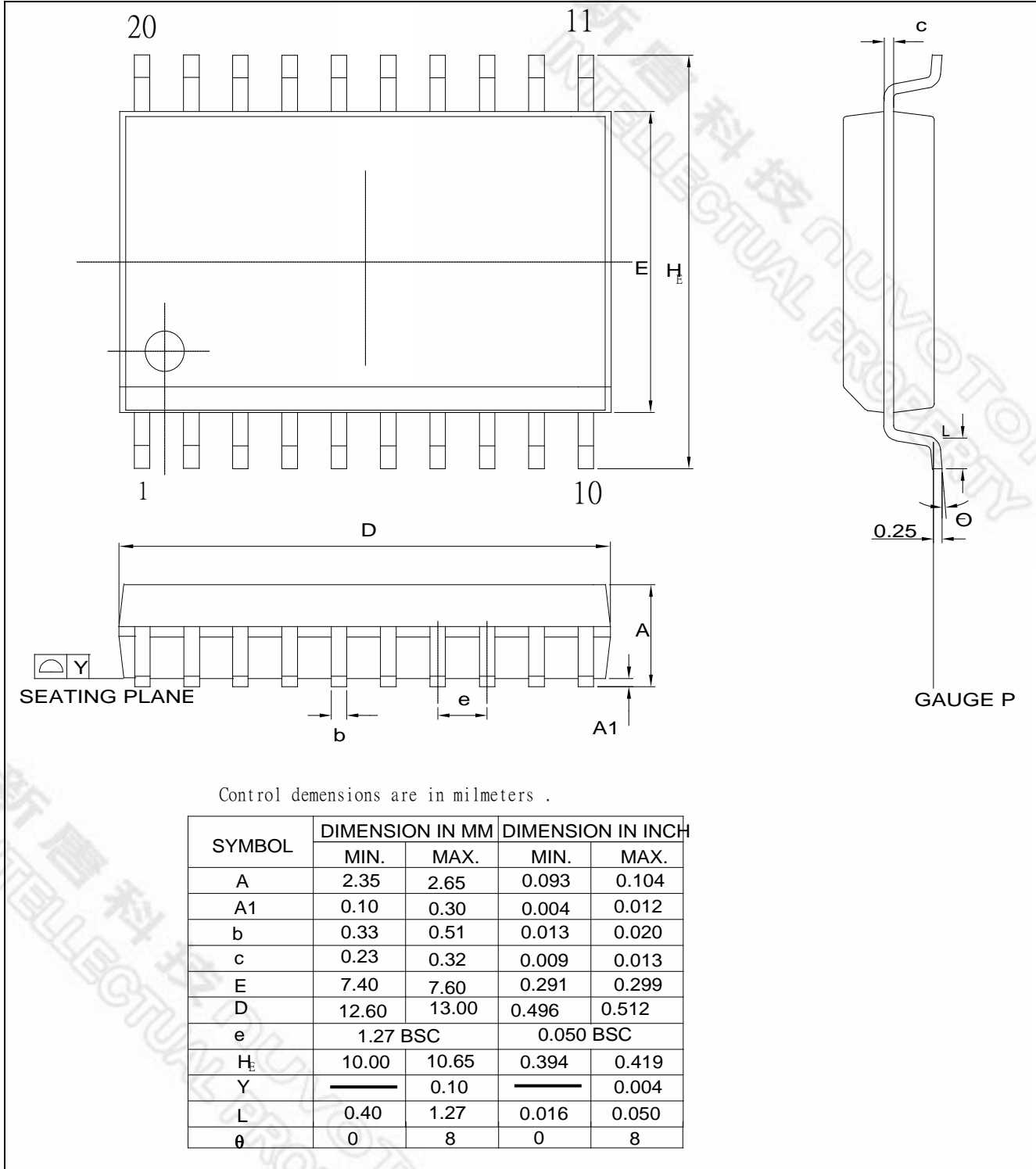
## 30.3 28脚 SOP-300mil



Control demensions are in milimeters .

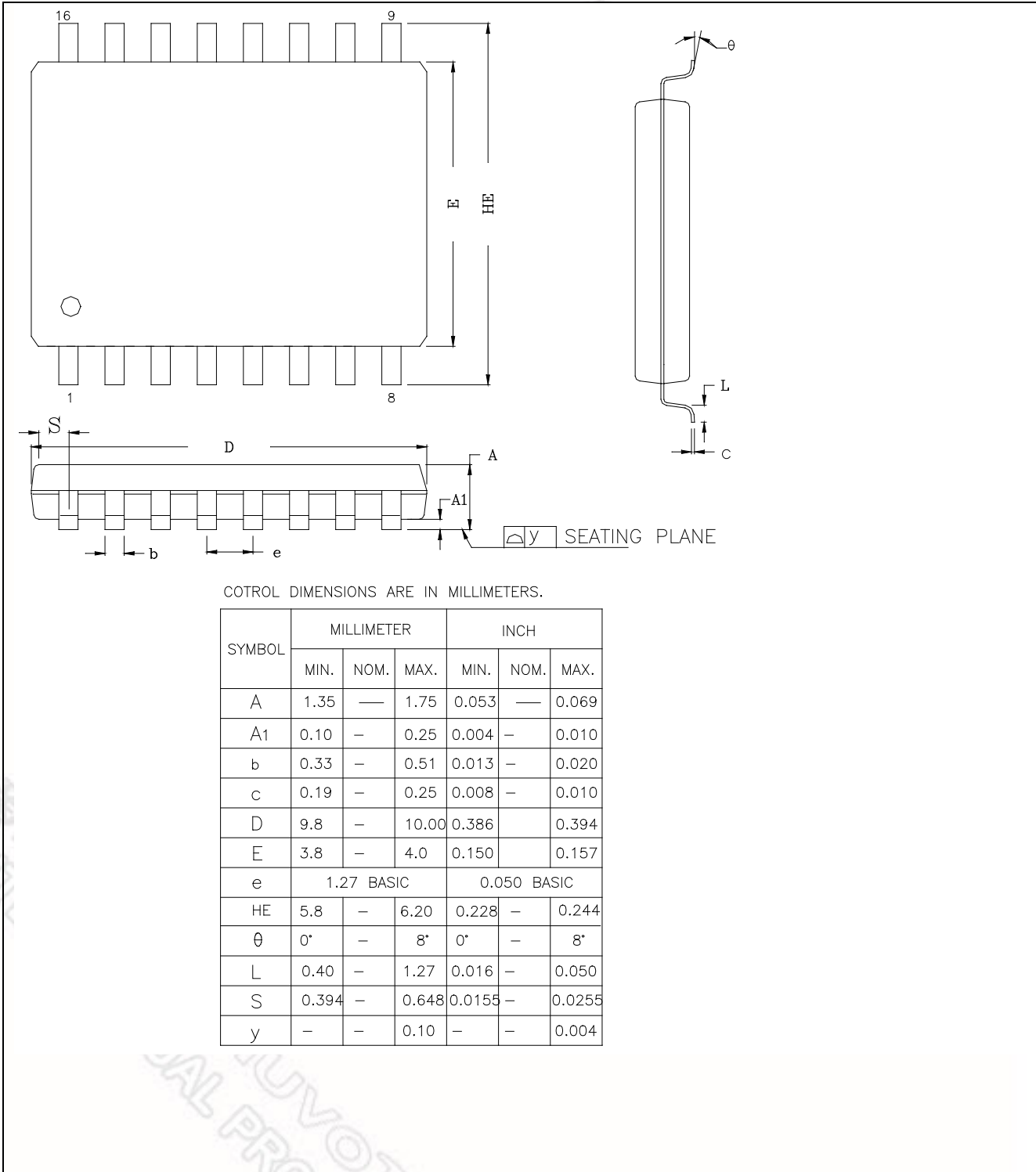
SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN.	MAX.	MIN.	MAX.
A	2.35	2.65	0.093	0.104
A1	0.10	0.30	0.004	0.012
b	0.33	0.51	0.013	0.020
c	0.23	0.32	0.009	0.013
E	7.40	7.60	0.291	0.299
D	17.70	18.10	0.697	0.713
e	1.27 BSC		0.050 BSC	
H <sub>E</sub>	10.00	10.65	0.394	0.419
Y	—	0.10	—	0.004
L	0.40	1.27	0.016	0.050
θ	0	8	0	8

30.4 20脚 SOP - 300 mil





## 30.5 16脚 SOP - 150 mil



### 31 版本历史

版本	发布日期	修改章节	描述
A1.0	-	-	初次发行
A2.0 – SC1	2011.10.25	-	更新描述错误
A2.1 – SC1	2011.11.30	-	增加复位脚描述
A2.1 – SC2	2012.3.30	-	1.修改2.4V条件下晶振范围 2.修改XTAL管脚特性 3.修改软件复位时SWDTRF 特性 4.修改XTAL 24MHz, $V_{DD}=5.5V$ 条件下电流特性 5.修改封装
A2.1 – SC3	2012.5.9	-	1.修改封装
A2.1 – SC4	2012.5.30	-	1.修改BOD侦测电压范围 2.修改ADCCON0 band-gap电压值
A2.1 – SC5	2012.9.5	-	1.修改 Flash可读写次数 2.修改封装说明 3. 修正电流单位
A2.1–SC6	2013.1	P89 P17 P35 P129 P122 P134	1. P0DIDS例程数据修正 2. CHBDA设定值修正 3. P3M1/P3M2描述修正 4. OP current值修正 5. 修正BOF清除方式. 6. 修正INT描述管脚
A2.1–SC7		P52 P158 P159	1. WDCON0/1 描述修正 2. ISP 电压 3. 修正Voh

新唐科技 NUVOTON  
INTELLECTUAL PROPERTY

### Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

---

*Please note that all data and specifications are subject to change without notice.  
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*