

Nuvoton 8-位微控制器

N79E815A

N79E814A

N79E813A

规格书

版本: **SC1**

目录

目录	2
1 概述	6
2 特征	7
3 器件信息列表	10
4 框图	11
5 引脚配置	12
6 内存组织	18
6.1 Flash IP的描述	19
6.1.1 APROM Flash	19
6.1.2 LDROM Flash	19
6.1.3 Data Flash	19
6.1.4 Config-bits(配置位)	19
6.2 片上XRAM	20
6.3 片上 scratch-pad RAM 和 SFR	20
6.4 工作寄存器	21
6.5 可位寻址区	22
6.6 Stack	22
6.7 片上非易失性Data Flash	22
7 特殊功能寄存器 (SFR)	24
8 通用 8051 内核	30
9 通用 IO (GPIO)	34
9.1 准双向输出配置	34
9.1.1 读-修改-写	35
9.2 开漏输出配置	36
9.3 推挽输出配置	36
9.4 输入配置	37
10 定时器/计数器 (定时器)	42
10.1 定时器/计数器0 和 1	42
10.1.1 模式 0 (13-bit 定时器)	45
10.1.2 模式 1 (16-bit Timer)	46
10.1.3 模式 2 (8-bit Auto-Reload Timer)	47
10.1.4 模式 3 (Two Separate 8-bit Timers)	47
10.2 定时器/计数器2	49
10.2.1 输入捕获模式	52
10.2.2 自动重加载模式	56
10.2.3 比较模式	57
11 看门狗定时器(WDT)	58
11.1 功能描述	58
11.2 看门狗定时器复位的应用	61
11.3 看门狗定时器中断应用	61
12 串口 (UART)	63

12.1	模式 0.....	65
12.2	模式 1.....	67
12.3	模式 2.....	68
12.4	模式 3.....	70
12.5	波特率.....	71
12.6	帧错误检测.....	72
12.7	多机通信.....	72
12.8	自动地址识别.....	73
13	串行外围设备接口 (SPI).....	75
13.1	特征.....	75
13.2	功能描述.....	75
13.3	SPI控制寄存器.....	77
13.4	操作模式.....	80
	13.4.1 主机模式.....	80
	13.4.2 从机模式.....	80
13.5	时钟格式和数据传输.....	81
13.6	从机选择脚配置.....	83
13.7	模式错误检测.....	83
13.8	写冲突检测.....	84
13.9	过载错误.....	84
13.10	SPI 中断.....	84
14	键盘中断(KBI).....	86
15	模拟到数字转换(ADC).....	90
15.1	ADC 分辨率和模拟电压.....	91
16	I ² C.....	95
16.1	特征.....	95
16.2	功能描述.....	95
	16.2.1 START 和 STOP 条件.....	96
	16.2.2 7-位 地址的数据格式.....	97
	16.2.3 应答.....	97
	16.2.4 仲裁.....	98
16.3	I ² C控制寄存器.....	99
16.4	模式操作.....	101
	16.4.1 主机发送模式.....	102
	16.4.2 主机接收器.....	103
	16.4.3 从机接收模式.....	104
	16.4.4 从机发送模式.....	105
	16.4.5 General Call.....	106
	16.4.6 其他状态.....	106
16.5	典型I ² C结构的中断服务程序.....	107
16.6	I ² C 超时.....	110
16.7	I ² C 中断.....	111
17	脉冲宽度调制(PWM).....	112
17.1	特征.....	112
17.2	功能描述.....	112

18	时控访问保护(TA).....	120
19	中断系统 (INTERRUPT).....	122
	19.1 中断源.....	122
	19.2 优先级结构.....	123
	19.3 中断响应时间.....	127
	19.4 中断的SFR.....	127
20	在系统编程(ISP).....	133
	20.1 ISP程序.....	133
	20.2 ISP 模式表.....	137
	20.3 N79E815 系列ISP程序访问表.....	138
	20.4 ISP使用指南.....	138
	20.5 ISP示例代码.....	139
21	电源管理.....	143
	21.1 Idle 模式.....	143
	21.2 掉电模式.....	144
22	时钟系统.....	145
	22.1 外部时钟源.....	147
	22.2 片上RC振荡器.....	147
23	电源监控.....	148
	23.1 上电检测.....	148
	23.2 BOD 检测.....	148
24	复位条件.....	152
	24.1 上电复位.....	153
	24.2 BOD 复位.....	153
	24.3 RST引脚复位.....	153
	24.4 看门狗定时复位.....	154
	24.5 软件复位.....	154
	24.6 启动选择.....	155
	24.7 复位状态.....	156
25	配置位 (CONFIG).....	158
	25.1 配置位0.....	158
	25.2 配置位1 (N79E815 Only).....	159
	25.3 配置位2.....	160
	25.4 配置位3.....	161
26	指令集.....	163
27	IN-CIRCUIT PROGRAM (ICP).....	167
28	电气特性.....	168
	28.1 绝对最大额定值.....	168
	28.2 DC 电气特性.....	168
29	模拟电气特性.....	172
	29.1 LDO 管理器规格.....	172
	29.2 10-位 SAR-ADC的规格.....	172
30	外部晶振和内部RC规格.....	173

30.1	4~24MHz XTAL 振荡器规格	173
30.2	内部RC Oscillator-22.1184MHz规格	173
30.3	内部 RC Oscillator-10KHz的规格	173
31	典型应用电路.....	175
32	封装尺寸	176
32.1	28-pin SOP - 300 mil	176
32.2	28-pin TSSOP - 4.4X9.7mm	177
32.3	20-pin TSSOP - 4.4X9.7mm	178
32.4	20-pin SOP - 300 mil	179
32.5	16-pin SOP – 150 mil.....	180
33	版本历史	181

1 概述

N79E815 系列是8位快速51微控制器 (4T 模式), 内嵌16K^[1]/8K/4K 的Flash EPROM , 可以通过通用烧写器, ICP编程器烧录程序, 带ISP功能. **N79E815** 系列的指令集与标准8052兼容. **N79E815** 系列有16K/8K/4K 字节的 AP Flash 内存, 4K字节的数据内存和2K 字节的 LD Flash 内存; 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). **N79E815** 系列支持内部RC振荡器, 22.1184MHz ($\pm 1\%$). 有16个中断源, 4个中断级. **N79E815** 内部的Flash EPROM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.

N79E815 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Flash 内存, 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] **N79E815**, 数据Flash和 APROM 共享16k-字节空间.

2 特征

- 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
 - $V_{DD} = 4.5V \text{ to } 5.5V @ F_{SYS} \text{ up to } 24MHz$
 - $V_{DD} = 2.7V \text{ to } 5.5V @ F_{SYS} = 12MHz$ 或内部 RC 11.0592MHz
 - $V_{DD} = 2.4 \text{ to } 5.5V @ F_{SYS} = 4MHz$
- 工作温度范围
 - $-40^{\circ}C \sim 85^{\circ}C$
- 时钟源
 - 高速外部振荡器:
 - 24 MHz 晶振和振荡器 (由config位使能).
 - 内部RC 振荡器: 22.1184MHz/11.0592MHz (由config位选择)
 - 在 $V_{DD} = 2.4V \sim 5.5V$ 和 $25^{\circ}C$ 条件下为 $\pm 1\%$.
 - 在 $V_{DD} = 2.4V \sim 5.5V$ 和 $-40^{\circ}C \sim 85^{\circ}C$ 条件下为 $\pm 5\%$.
 - 由config位和软件配置的CPU时钟源
 - 8-位可编程的CPU时钟除频器 (DIVM)
- 片上内存
 - 2K-字节LD Flash 用于ISP功能(LDROM)
 - 16K/8K/4K字节的片上Flash EPROM, 可页擦除和字节编程
 - 16K/8K/4K 应用程序Flash (APROM)
 - 数据 Flash 4K 字节(DATAFlash)
 - 在16K AP Flash模式下, 可配置APROM 和DataFlash大小.
 - N79E815A: 总共 16K-字节分别用于AP Flash 和 Data Flash (由config位定义)
 - N79E814A: 8K-字节AP Flash 和 4K-字节数据Flash
 - N79E813A: 4K-字节AP Flash 和 4K-字节数据Flash
 - APROM/LDROM 和数据Flash 安全保护.
 - Flash 页大小为128 字节
 - 256 字节片上直接/间接RAM
 - 256 字节MOVX-RAM, 用指令MOVX 访问.
 - 片上FLASH可用以下方式编程
 - 通用 H/W 烧录器模式
 - 串行在电路编程模式 (ICP)
 - 软件执行ISP/IAP (在系统编程/在应用编程)

- I/O 端口
 - 最大**25** I/O.
 - 每个引脚支持**4**种由软件配置的输出模式
 - 软件可选择**TTL**或**史密特**触发输入类型
 - **14** 个中断源带**4**个中断优先级.
 - **P10, P11, P14, P16, P17**的**LED** 驱动能力为**40mA**.
 - 端口**0, 2, 3**的**LED**驱动能力为**20mA**.
- 定时器/计数器
 - **2** 组**16**-位定时器/计数器.
 - 一个定时器带三个输入捕获能力
- 看门狗定时器
 - 可编程看门狗定时器
 - 时钟源为内部**10KHz 50%** 的**RC**振荡器.
- 串口(**UART, SPI, I2C**)
 - 一组增强型全双工**UART**口, 带帧错误检测和自动地址识别. 软件可切换两组**UART**引脚.
 - 一个**SPI**带主/从功能. 软件可切换两组**SPI**引脚
 - 一个**I2C**带主/从功能.
- **PWM**
 - **4** 通道 **10**-位 **PWM** 输出带一个**Brake/Fault**输入
- **KBI**
 - **8**-键盘中断输入(**KBI**) 带**8**个下降沿/双边沿检测引脚 (由软件选择)
- **ADC**
 - **10**-位 **A/D** 转换器
 - 达**150 Ksps**.(采样率/秒)
 - **8** 个模拟输入通道
- 欠压检测
 - **2**-级 (**3.8V/2.7V**) **BOD** 检测器
 - 支持中断和复位选项
- **LVR** (低压复位)
 - 门限电压为**2.0V**
- 内建电源管理.
 - **Idle** 模式
 - 掉电模式 (使能**BOD**或**WDT**功能)
- 开发工具
 - **ICE**(在电路仿真)工具
 - **H/W** 烧写器
 - **ICP** 编程器
 - **ISP** 编程器
- 封装



- N79E815AS28 --- SOP28
- N79E815AT28 --- TSSOP28
- N79E815AD20 --- PDIP20
- N79E815AS20 --- SOP20
- N79E815AT20 --- TSSOP20
- N79E814AS28 --- SOP28
- N79E814AT28 --- TSSOP28
- N79E814AD20 --- PDIP20
- N79E814AS20 --- SOP20
- N79E814AT20 --- TSSOP20
- N79E813AS28 --- SOP28
- N79E813AT28 --- TSSOP28
- N79E813AD20 --- PDIP20
- N79E813AS20 --- SOP20
- N79E813AT20 --- TSSOP20
- N79E8132AS16 --- SOP16

3 器件信息列表

无铅(RoHS)器件信息列表

PART NO.	APROM	LDROM	RAM	数据 FLASH	封装	备注
N79E815AS28	16KB	2KB	512B	与 APROM 共享 16KB	SOP-28 Pin	
N79E815AT28	16KB	2KB	512B	与APROM共享 16KB	TSSOP-28 Pin	
N79E815AS20	16KB	2KB	512B	与APROM共享 16KB	SOP-20 Pin	
N79E815AT20	16KB	2KB	512B	与APROM共享 16KB	TSSOP-20 Pin	
N79E815AD20	16KB	2KB	512B	与APROM共享 16KB	PDIP-20 Pin	
N79E814AS28	8KB	2KB	512B	4KB	SOP-28 Pin	
N79E814AT28	8KB	2KB	512B	4KB	TSSOP-28 Pin	
N79E814AS20	8KB	2KB	512B	4KB	SOP-20 Pin	
N79E814AT20	8KB	2KB	512B	4KB	TSSOP-20 Pin	
N79E814AD20	8KB	2KB	512B	4KB	PDIP-20 Pin	
N79E813AS28	4KB	2KB	512B	4KB	SOP-28 Pin	
N79E813AT28	4KB	2KB	512B	4KB	TSSOP-28 Pin	
N79E813AS20	4KB	2KB	512B	4KB	SOP-20 Pin	
N79E813AT20	4KB	2KB	512B	4KB	TSSOP-20 Pin	
N79E813AD20	4KB	2KB	512B	4KB	PDIP-20 Pin	
N79E8132AS16	4KB	2KB	512B	4KB	SOP-16 Pin	

表 3-1: 无铅 (RoHS) 器件信息列表

4 框图

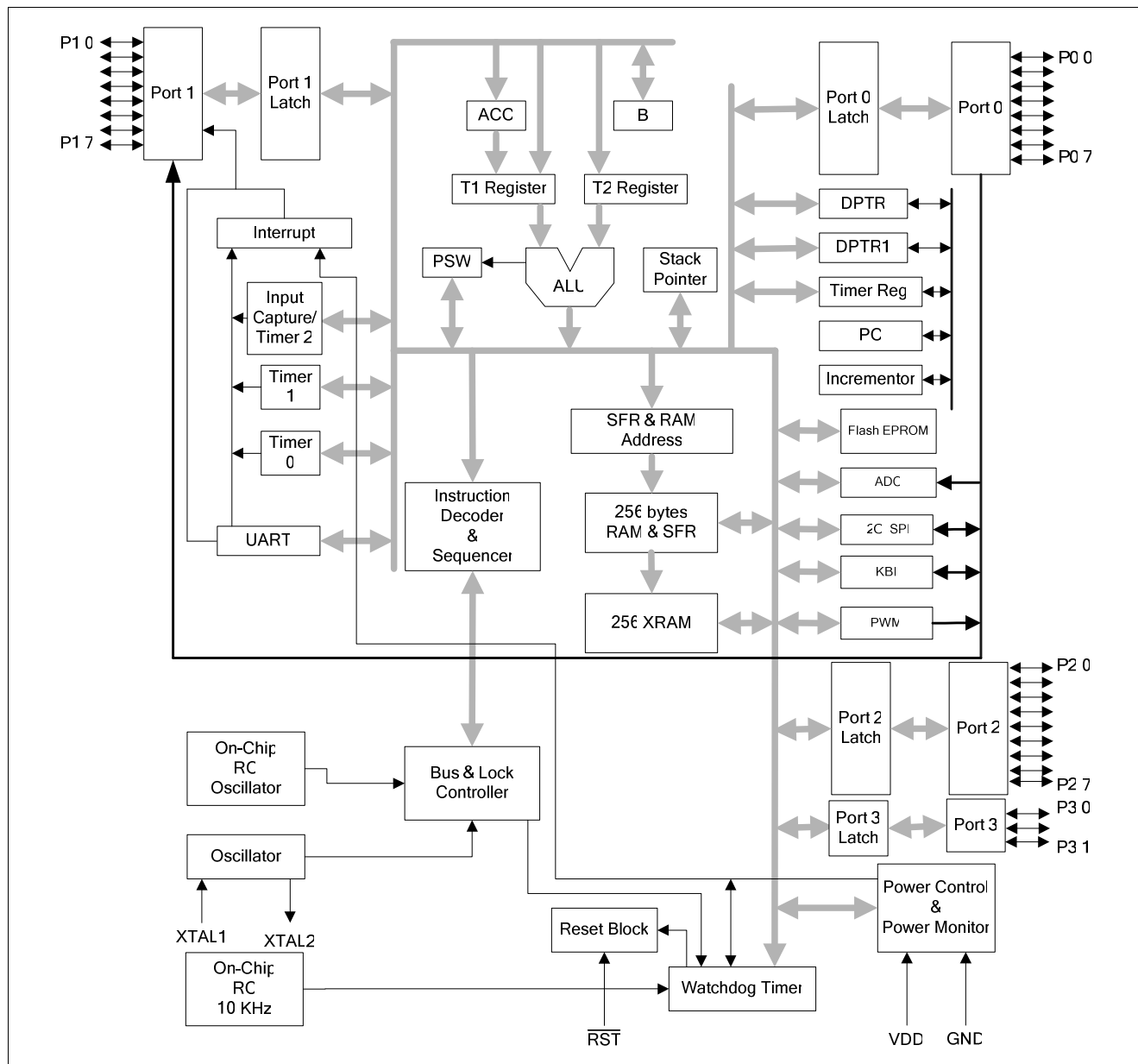


图 4-1. N79E815 系列功能框图

5 引脚配置

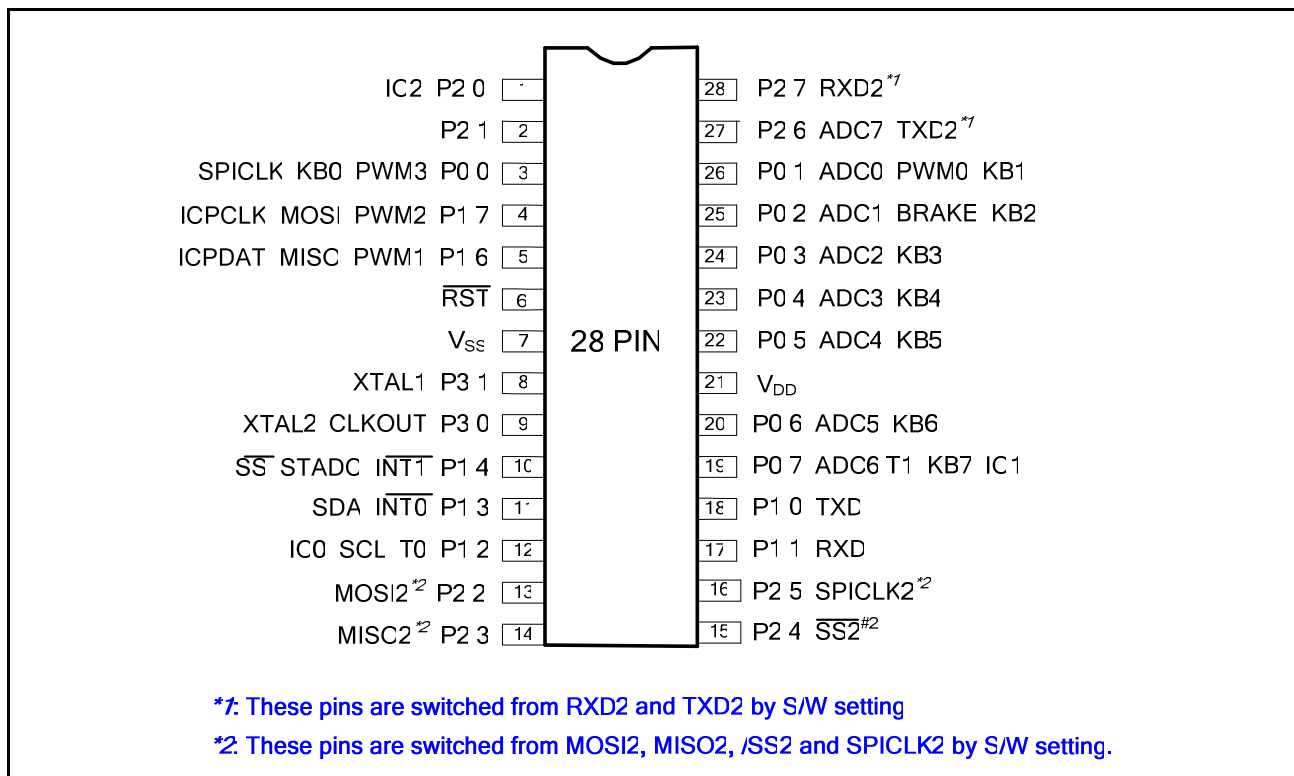


图 5-1. SOP/TSSOP28 管脚引脚分配图

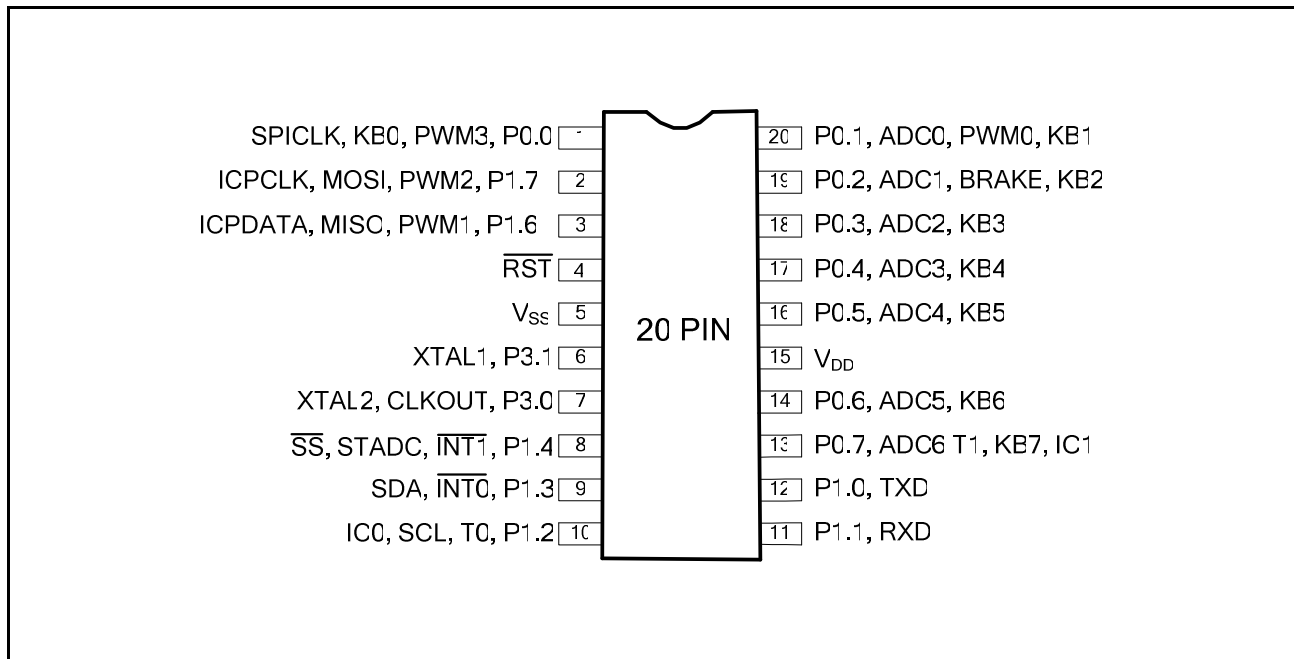


图 5-2. TSSOP/PDIP/SOP 20脚引脚分配图

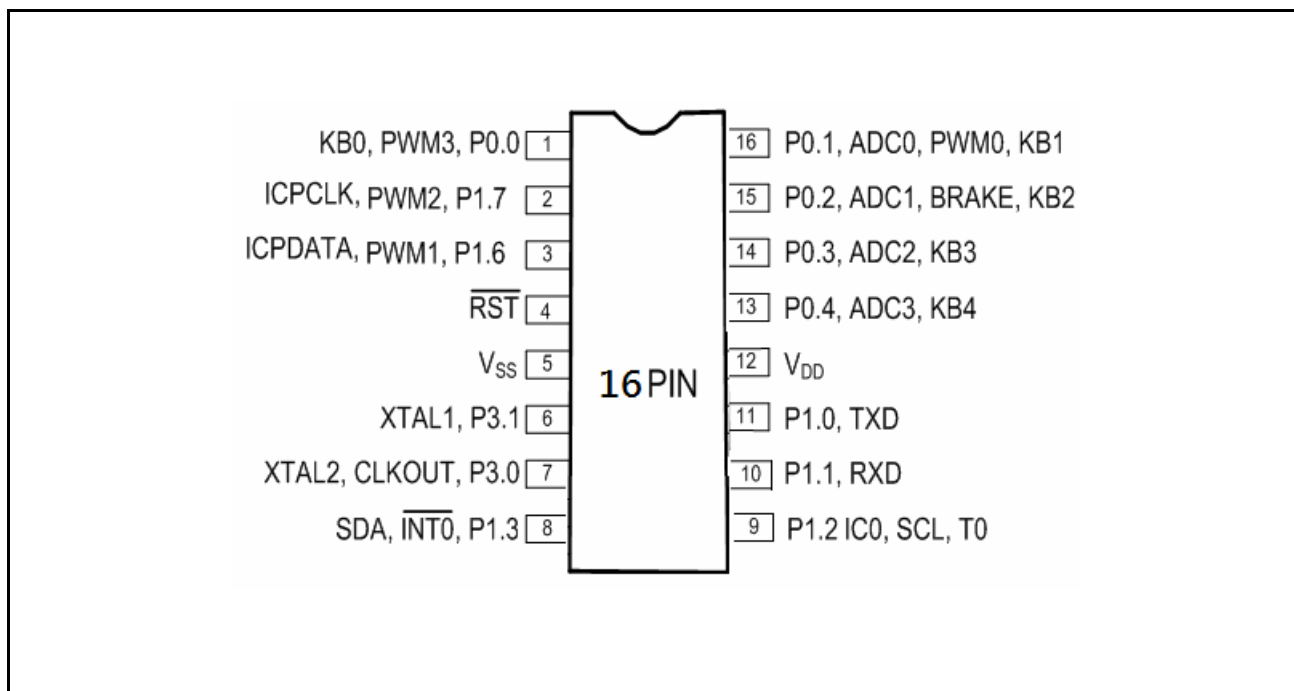


图5-3. SOP 16脚引脚分配图

表 5-1. 引脚描述 SOP28 / TSSOP28

引脚号	符号	复用功能				类型 ^[1]	描述
		1	2	3			
21	V _{DD}					P	提供电源: 提供工作电源V _{DD} .
7	V _{SS}					P	GROUND: 地.
6	/RST					I (ST)	RESET: 芯片复位引脚, 低电平有效.
3	P0.0	PWM3		KB0	SPICLK	I/O	PORT0: 端口 0 有 4 个类型的 I/O 口. 其多功能可用于 PWM0, PWM3, T1, BRAKE, SPICLK, 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
26	P0.1	PWM0	ADC0	KB1		I/O	
25	P0.2	BRAKE	ADC1	KB2		I/O	
24	P0.3		ADC2	KB3		I/O	
23	P0.4		ADC3	KB4		I/O	
22	P0.5		ADC4	KB5		I/O	
20	P0.6		ADC5	KB6		I/O	
19	P0.7	T1	ADC6	KB7	IC1	I/O	
18	P1.0	TXD				I/O	
17	P1.1	RXD				I/O	PORT1: 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 /SS, MISO, MOSI. TXD 和 RXD 为 UART 口 P1.2 (SCL) 和 P1.3 (SDA) 为开漏口. ICPDAT 和 ICPCLK 为 ICP (在电路编程)功能引脚. /SS, 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 STADC: ADC 外部触发脚
12	P1.2	T0		SCL	IC0	D	
11	P1.3	/INT0		SDA		D	
10	P1.4	/INT1	STADC		/SS	I/O	
5	P1.6	PWM1		ICPDAT	MISO	I/O	
4	P1.7	PWM2		ICPCLK	MOSI	I/O	
1	P2.0				IC2	I/O	
2	P2.1					I/O	
13	P2.2				MOSI2	I/O	
14	P2.3				MISO2	I/O	PORT2: 端口 2 有 4 个类型的 I/O 口. 其多功能用于 T2, ADC7, TXD2, RXD2 和 MOSI2, MISO2, /SS2, SPICLK2, IC2 TXD2 和 RXD2 为 UART 口, MOSI2, MISO2, /SS2 和 SPICLK2 为 SPI-2 功能引脚.可以通过软件将SPI-1口切换到SPI2 ADC7: ADC通道输入.
15	P2.4				/SS2	I/O	
16	P2.5				SPICLK2	I/O	

表 5-1. 引脚描述 SOP28 / TSSOP28

引脚号	符号	复用功能			类型 ^[1]	描述
		1	2	3		
27	P2.6	TXD2	ADC7		I/O	IC2: 捕获引脚输入
28	P2.7	RXD2			I/O	
9	P3.0	XTAL2	CLKOUT		I/O	PORT3: 端口 3 有 4 个类型的 I/O 口. 其多功能用于 XTAL1, XTAL2 和 CLKOUT, CLKOUT: 内部RC OSC/4 输出引脚. XTAL2: 内部放大器的输出. 发送XTAL2的反向信号XTAL2. XTAL1: 内部放大器的输出. 发送XTAL1的反向信.
8	P3.1	XTAL1			I/O	

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

表5-2. 引脚描述 DIP20 / SOP20 / TSSOP20

引脚号	符号	复用功能				类型 ^[1]	描述
		1	2	3			
21	V _{DD}					P	提供电源: 提供工作电源V _{DD} ..
7	V _{SS}					P	GROUND:地.
6	/RST					I (ST)	RESET: 芯片复位引脚, 低电平有效.
3	P0.0	PWM3		KB0	SPICLK	I/O	PORT0: 端口 0 有 4 个类型的 I/O 口. 其多功能可用于 PWM0, PWM3, T1, BRAKE, SPICLK, 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
26	P0.1	PWM0	ADC0	KB1		I/O	
25	P0.2	BRAKE	ADC1	KB2		I/O	
24	P0.3		ADC2	KB3		I/O	
23	P0.4		ADC3	KB4		I/O	
22	P0.5		ADC4	KB5		I/O	
20	P0.6		ADC5	KB6		I/O	
19	P0.7	T1	ADC6	KB7	IC1	I/O	
18	P1.0	TXD				I/O	
17	P1.1	RXD				I/O	PORT1: 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 /SS, MISO, MOSI. TXD 和 RXD 为 UART 口 P1.2 (SCL) 和 P1.3 (SDA) 为开漏口. ICPDAT 和 ICPCLK 为 ICP (在电路编程)功能引脚. /SS, 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 STADC: ADC 外部触发脚
12	P1.2	T0		SCL	IC0	D	
11	P1.3	/INT0		SDA		D	
10	P1.4	/INT1	STADC		/SS	I/O	
5	P1.6	PWM1		ICPDAT	MISO	I/O	
						I/O	
4	P1.7	PWM2		ICPCLK	MOSI	I/O	
						I/O	
						I/O	

9	P3.0	XTAL2	CLKOUT			I/O	PORT3: 端口 3 有 4 个类型的 I/O 口. 其多功用于 XTAL1, XTAL2 和 CLKOUT, CLKOUT: 内部RC OSC/4 输出引脚.. XTAL2: 内部放大器的输出. 发送XTAL2的反向信号XTAL2. XTAL1: 内部放大器的输出. 发送XTAL1的反向信.
8	P3.1	XTAL1				I/O	

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

6 内存组织

N79E815系列内嵌有16K/8K/4K字节的Flash EEPROM为应用程序Flash(AP Flash), 配置有4K字节的数据Flash(除16K AP Flash的器件), 配置有2K字节的Load ROM Flash (LD Flash) 和配置位Flash. N79E815系列还提供256字节的片上直接/间接RAM和256字节的MOVX-RAM(由MOVX指令访问).

对于16K-字节AP Flash的器件, AP Flash 和 Data Flash 共同组成16K字节的内嵌Flash. 块大小由配置位/软件配置.

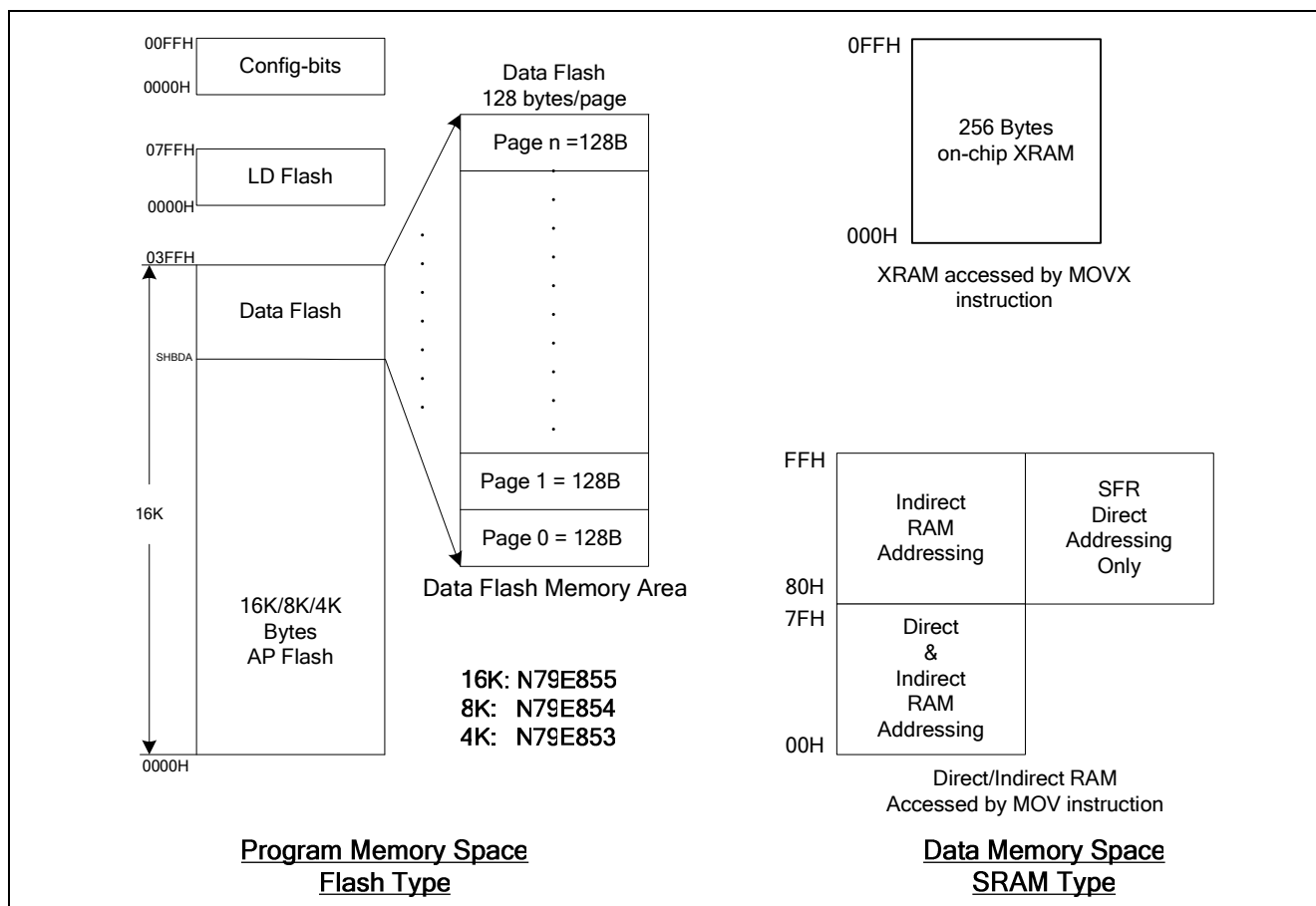


图 6-1 N79E815 系列内存图

6.1 Flash IP的描述

N79E815 系列内建CMOS页可擦写和字节编程的Flash内存，可分成两个内存块。页擦除操作可擦除一个页上的128字节。

N79E815 系列的内嵌Flash内存分为4个区块： AP Flash, Data Flash, LD Flash 和 Config-bits. 详细描述见以下章节。

6.1.1 APROM Flash

N79E815 系列的应用程序内存可达**16K/8K/4K** 字节。所有指令由该内存区域取出并执行。指令MOVC 可以访问该内存区域。

用户的应用程序位于AP Flash。CPU 由AP Flash 启动(CHPCON.BS=0), CPU 开始从地址0000H 执行程序。如果PC计数器的值超过AP Flash的空间, CPU 将执行NOP operand 和程序计数器加1直到PC达到3FFFH , 然后回到AP Flash地址 0000H, CPU 重新执行应用程序。

6.1.2 LDROM Flash

N79E815 系列配备有2K-字节 LD Flash 用于存储ISP应用程序。用户在LD Flash里开发ISP功能用于更新应用程序或数据Flash。同时, AP Flash 也可以重新编程LD Flash 和 Data Flash。LD Flash 的起始地址在 0000H。当CPU 在LD Flash中运行时, CPU 自动定位到LD Flash 的开始地址0000H, 因此用户程序把LD Flash 作为独立的程序内存, 同时, 包括CPU提供的所有中断向量。

6.1.3 Data Flash

对于16K-byte AP Flash 器件, AP 和 数据区域都在16K Flash里。Data Flash 的开始地址(DF_Start_Addr)由SFR SHBDA(9CH)决定。上电复位后SHBDA 的值在配置位CHBDA里。在写SHBDA 时SFR SHBDA[7:0] 有时序访问保。注意如果SHBDA设置成不合理的值, 如超过16K-byte 地址空间, CPU 将迫使Data Flash无法由软件访问。

例如, 如果用户想把16K的Flash配置成10K AP Flash 和 4K Data Flash。Data Flash 的起始地址为2800H, CHBDA/SHBDA 必须编程为28H。

6.1.4 Config-bits(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。只有硬件烧录器或ICP烧录器可以擦除/编程Config-bits。LD Flash 的ISP程序也可以软件擦除/编程Config-bits。

6.2 片上XRAM

N79E815 系列提供片上256 bytes 辅助 RAM (XRAM) 以扩大RAM 空间. 占用的地址空间从00H 到 FFH. XRAM 的256字节可以由MOVX @DPTR or MOVX @Ri 间接访问. (参见下面的demo code.) 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 demo code:

```

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

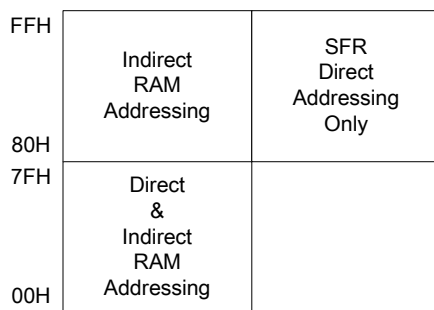
MOV    DPTR,#0023H       ;write #5BH to XRAM with address @0023H
MOV    A,#5BH
MOVX   @DPTR,A

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

```

6.3 片上 scratch-pad RAM 和 SFR

N79E815 系列提供片上256 bytes scratch pad RAM 和特殊功能寄存器 (SFRs) , 可由软件访问. SFRs 仅可直接寻址, 片上RAM 可直接/间接寻址.



RAM and SFR Data Memory Space

图 6-2 256 bytes RAM 和 SFR

scratch-pad RAM只有 256 byte , 仅在数据内容很小时才用.在scratch-pad RAM里有几个其他特殊用途区域. 描述如下.

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

6.4 工作寄存器

有四组工作寄存器, 每组由8个8位寄存器组成. 分布在Banks 0, 1, 2, 和 3. 这些banks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, **N79E815** 系列一次只能工作在一个特殊bank. bank 通过设置PSW的RS1-RS0位来选择. 寄存器 R0 和 R1用于存储地址以间接寻址.

6.5 可位寻址区

Scratch-pad RAM 位于20h 到 2Fh ，可以字节或位寻址. 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SFRs 也可以位寻址.. 在 SFR 区域, SFR 的地址以0或8结尾都是可位寻址.

6.6 Stack

scratch-pad RAM 可被用于stack. 该区域由堆指针 (SP) 选定, 存储堆顶的地址. 跳转或中断发生时, 返回地址在堆上. Stack对RAM上的起始地址没有限制, 默认情况下, 复位时Stack 指针在07h. 用户可以改变该值. SP 将指向上次使用的值. 因此, SP 将递增, 地址保存到stack. 相反, 当出堆时, 先读, SP递减.

6.7 片上非易失性Data Flash

N79E815 系列有非易失性 Data Flash. Data Flash可以在掉电时保存数据. 因此, 用户可以写或读数据 (参数或常量). 通过软件方式, Data Flash 仅由ISP擦写, 可通过MOVC指令或ISP读. 通过软件方式, Data Flash 仅可通过ISP擦除, 写, 或读. 当然, Data Flash也可以通过硬件编程器访问.

N79E815 (16KB)的Data Flash 大小可通过软件更新SHBDA的值来校正. SHBDA[7:0] 表示16位Data Flash 开始地址的高字节, 低字节由硬件设置为00H. 所有复位后, SHBDA的值由CONFIG1 (CHBDA)的值加载. 应用程序可以通过重置SHBDA的值动态校正Data Flash 的大小. 一旦Data Flash 的大小改变, APROM 的大小也相应改变. 写SHBDA 时有时序保护要求. 注意如果CHBDA 为 00H, Data Flash 的大小为16k bytes , 则没有APROM. CPU 将执行外部内存的代码.

在**N79E814/N79E813**, Data Flash 大小固定为4k bytes , 从地址F000H 到 FFFFH. SHBDA 无效.

在访问Data Flash之前, CONFIG bit DFEN (CONFIG0.0) 应该编程为0. 在**N79E815** 如果 DFEN 保持为1, APROM 将占用整个16k-byte block和在**N79E814/N79E813**里, 4k-byte block 从3000H到3FFFH保持为空.

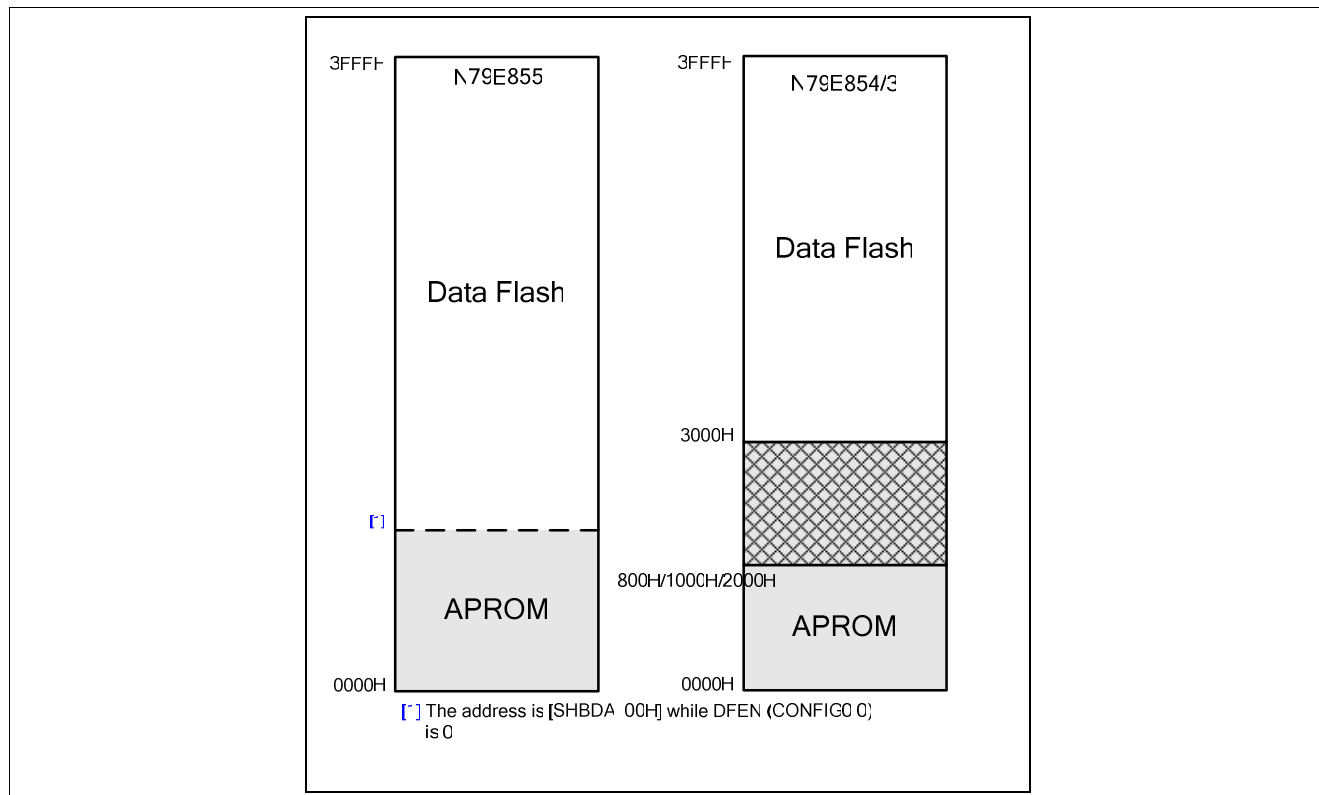


图 6-3. N79E815 Data Flash

SHBDA –Data Flash开始地址的高字节(TA protected, N79E815 Only)

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

地址: 9CH

位	名称	描述
7:0	SHBDA[7:0]	Data Flash开始地址的高字节。 该字节仅在DFEN (CONFIG0.0)为0的条件下有效. 用于应用程序执行时动态校正Data Flash的开始地址。

[1] SHBDA is loaded from CONFIG1 after all resets.

7 特殊功能寄存器 (SFR)

N79E815 系列MCU内核用特殊功能寄存器(特殊功能寄存器)来控制 and 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0或8结尾。**N79E815**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. N79E815 系列特殊功能寄存器(SFR) 图

F8	ADCCON0	-	-	-	-	-	-	EIP	FF
F0	B	-	-	SPCR	SPSR	SPDR	P0DIDS	EIPH	F7
E8	EIE	KBIE	KBIF	KBLS0	KBLS1	C2L	C2H	-	EF
E0	ACC	ADCCON1	ADCH	-	C0L	C0H	C1L	C1H	E7
D8	WDCON0^[3]	PWMPL	PWM0L	PWM1L	PWMCON0	PWM2L	PWM3L	PWMCON1	DF
D0	PSW	PWMPH	PWM0H	PWM1H	-	PWM2H	PWM3H	PWMCON2	D7
C8	T2CON	T2MOD	RCOMP2L	RCOM2H	TL2	TH2	-	-	CF
C0	I2CON	I2ADDR	-	-	-	-	-	TA	C7
B8	IP	SADEN	-	-	I2DATA	I2STAT	I2CLK	I2TOC	BF
B0	P3	P0M1	P0M2	P1M1	P1M2	P2M1	P2M2	IPH	B7
A8	IE	SADDR	-	WDCON1 ^[3]	-	-	ISPFD	ISPCN	AF
A0	P2	-	AUXR1	PMCR^[3]	ISPTRG^[3]	-	ISPAL	ISPAH	A7
98	SCON	SBUF	-	-	SHBDA^[3]	-	-	CHPCON^[3]	9F
90	P1	-	CAPCON0	CAPCON1	CAPCON2	DIVM	P3M1	P3M2	97
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	-	8F
80	P0	SP	DPL	DPH	-	-	-	PCON	87

In Bold	bit-addressable
-	reserved

- [1]. 保留的 SFR 地址必须保持他们自己的初始值. 用户不能改变其值.
- [2]. 有深色边框的 SFRs 可位寻址
- [3]. 带 TA-保护. (Time Access Protection)



Table 7–2. N79E815 Series SFR Descriptions and Reset Values

Symbol	Definition	Address	MSB							LSB		Reset Value ^[1]
EIP	Interrupt Priority 1	FFH	PT2	PSPI	PPWM	PWDI	-	-	PKB	PI2	0000 0000B	
ADCCON0	ADC control register	F8H	(FF) ADC.1	(FE) ADC.0	(FD) ADCEX	(FC) ADCI	(FB) ADCS	(FA) AADR2	(F9) AADR1	(F8) AADR0	XX00 0X00B	
EIPH	Interrupt High Priority 1	F7H	PT2H	PSPIH	PPWMH	PWDIH	-	-	PKBH	PI2H	XX00 0000B	
P0DIDS	Port 0 Digital Input Disable	F6H	P0DIDS[7:0]								0000 0000B	
SPDR	Serial Peripheral Data Register	F5H	SPDR[7:0]								XXXX XXXXB	
SPSR	Serial Peripheral Status Register	F4H	SPIF	WCOL	SPIOVF	MODF	DISMODF	-	-	-	0000 0XXXXB	
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	XXXX XXXXB	
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 ^[4]	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 C0JU 1UUUB Other reset C0JU UUUUB	
PWMCON2	PWM control register 2	D7H	-	-	-	-	FP1	FP0	-	BKF	XXXX 00X0B	
PWM3H	PWM 3 high bits register	D6H	-	-	-	-	-	-	PWM3.9	PWM3.8	XXXX XX00B	
PWM2H	PWM 2 high bits register	D5H	-	-	-	-	-	-	PWM2.9	PWM2.8	XXXX XX00B	
PWM1H	PWM 1 high bits register	D3H	-	-	-	-	-	-	PWM1.9	PWM1.8	XXXX XX00B	
PWM0H	PWM 0 high bits register	D2H	-	-	-	-	-	-	PWM0.9	PWM0.8	XXXX XX00B	



Table 7–2. N79E815 Series SFR Descriptions and Reset Values

Symbol	Definition	Address	MSB								LSB		Reset Value ^[1]
PWMPH	PWM counter high register	D1H	-	-	-	-	-	-	-	PWMP0.9	PWMP0.8	0000 0000B	
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	T2DIV2	T2DIV1	T2DIV0	CAPCR	COMPCR	LDTS1	LDTS0	0000 0000B		
T2CON	Timer 2 Control	C8H	(CF) TF2	-	-	-	-	(CA) TR2	-	(C8) CMP/RL2	0XXX X0X0B		
TA	Timed Access Protection	C7H									0000 0000B		
I2ADDR	I2C address	C1H	ADDR.6	ADDR.5	ADDR.4	ADDR.3	ADDR.2	ADDR.1	ADDR.0	GC	XXXX XXX0B		
I2CON	I2C Control register	C0H	(C7) -	(C6) I2CEN	(C5) STA	(C4) STO	(C3) SI	(C2) AA	(C1) -	(C0) -	X000 00XXB		
I2TOC	I2C Time-out Counter register	BFH	-	-	-	-	-	I2TOCEN	DIV	I2TOF	0000 0000B		
I2CLK	I2C Clock Rate	BEH	I2CLK[7:0]								0000 0000B		
I2STAT	I2C Status Register	BDH	I2STAT.4	I2STAT.3	I2STAT.2	I2STAT.1	I2STAT.0	0	0	0	1111 1000B		
I2DAT	I2C Data Register	BCH	I2DAT[7:0]								XXXX XXXXB		
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	X000 0000B		
IPH	Interrupt high priority	B7H	PCAPH	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H	X000 0000B		
P2M2	Port 2 output mode 2	B6H	P2M2[7:0]								XXXX XX00B		
P2M1	Port 2 output mode 1	B5H	P2M1[7:0]								0000 0000B		
P1M2	Port 1 output mode 2	B4H	P1M2[7:0]								0000 0000B		
P1M1	Port 1 output mode 1	B3H	P1M1[7:0]								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	XXXX XX00B		
ISPCN	ISP Control Register	AFH	ISPA17	ISPA16	FOEN	FCEN	FCTRL3	FCTRL2	FCTRL1	FCTRL0	0011 0000B		
ISPFD	ISP Flash Data Register	AEH	ISPFD[7:0], ISP Flash Data								0000 0000B		
WDCON1 ^[4]	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 ^[4]	ISP Trigger Register	A4H	-	-	-	-	-	-	-	ISPGO	XXXX XXX0B		

Table 7-2. N79E815 Series SFR Descriptions and Reset Values

Symbol	Definition	Address	MSB								LSB	Reset Value ^[1]
PMCR ^{[2][4]}	Power Monitor Control Register	A3H	BODEN	BOV	-	BORST	BOF	LPBOD	-	-	Power-on CCXC 10XXB BOR reset UUXU 10XXB Other reset UUXU 00XXB	
AUXR1	AUX function register	A2H	SPI_Sel	UART_Sel	-	-	DisP26	-	-	DPS	0000 0000B	
P2	Port 2	A0H	(97) P27	(96) P26	(95) P25	(94) P24	(93) P23	(92) P22	(91) P21	(90) P20	1111 1111B	
CHPCON ^[4]	Chip Control	9FH	SWRST	ISPF (Read only)	LDUE	-	-	-	BS ^[3]	ISPEN	Power-ON 0000 00C0B Other reset 000X XU00B	
SHBDA ^[4]	High-byte Data Flash Start Address	9CH	SHBDA[7:0], SHBDA Initial by CHBDA								Power ON CCCC CCCC 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	XXXXX000B	
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	CAP2LS0	CAP1LS1	CAP1LS0	CAP0LS1	CAP0LS0	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	-	-	-	XXX0 0XXB	
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 BOD Reset 00xu 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：标记“-”的位必须保持他们自己的初始状态。用户不能更改其值。

注：

[1] () 表示可位寻址。



- [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. T: 由 Trim bits 初始化.

8 通用 8051 内核

A or 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

Address: E0H

reset value: 0000 0000b

位	名称	描述
7:0	ACC[7:0]	累加器. 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

Address: F0H

reset value: 0000 0000b

位	名称	描述
7:0	B[7:0]	B 寄存器. 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 – Stack 指针

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

Address: 81H

reset value: 0000 0111b

位	名称	描述
7:0	SP[7:0]	Stack 指针. Stack 指针存储scratch-pad RAM 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							

Address: 82H

reset value: 0000 0000b

位	名称	描述
7:0	DPL[7:0]	数据指针低字节。 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							

Address: 83H

reset value: 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

Address: D0H

reset value: 0000 0000b

位	名称	描述
7	CY	进位标志。 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 为0..
6	AC	辅助进位。 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	用户标志0。 可由用户置位或清零的通用标志.
4	RS1	寄存器 Bank 选择位.

位	名称	描述																				
3	RS0	这两位用于选择banks. <table border="1"> <thead> <tr> <th>RS1</th> <th>RS0</th> <th>Register Bank</th> <th>RAM Address</th> </tr> </thead> <tbody> <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> </tbody> </table>	RS1	RS0	Register Bank	RAM Address	0	0	0	00~07H	0	1	1	08~0FH	1	0	2	10~17H	1	1	3	18~1FH
RS1	RS0	Register Bank	RAM Address																			
0	0	0	00~07H																			
0	1	1	08~0FH																			
1	0	2	10~17H																			
1	1	3	18~1FH																			
2	OV	溢出标志. OV 用于符号运算..																				
1	F1	用户标志1. 可由用户置位或清零的通用标志																				
0	P	奇偶标志..																				

表 8-1. 指令对标志位的影响

指令	CY	OV	AC	指令	CY	OV	AC
ADD	X ^[1]	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 表示改动依赖指令的结果.

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

Address: 87H

reset value: see [Table 7-2. N79E815 Series SFR Descriptions and Reset Values](#)

位	名称	描述
7	SMOD	<p>串口波特率加倍使能</p> <p>1: 置1时, 该位加倍串口模式1, 2, 3的波特率.</p>
6	SMOD0	<p>帧错误检测使能</p> <p>0: 禁止帧错误检测. SCON.7 用作标准8052功能.</p> <p>1: 使能帧错误检测, SCON.7 表示帧错误, 用作FE 标志.</p>
3	GF1	<p>通用标志1.</p> <p>通用标志可由用户置位或清零.</p>
2	GF0	<p>通用标志0.</p> <p>通用标志可由用户置位或清零.</p>

9 通用 IO (GPIO)

N79E815 有 4 个 I/O, 端口 0, 端口 1, 端口 2 和 端口 3. 如果使用片上 RC 振荡器和配置复位引脚, **N79E815** 系列可支持达 25 pins. 所有的 I/O 口可配置成下表四种类型之一.

表 9-1. I/O 端口结构配置表

PxM1.y	PxM2.y	Port Input/Output Mode
0	0	Quasi-bidirectional
0	1	Push-Pull
1	0	Input Only (High Impedance)
1	1	Open Drain

注: P1.2 和 P1.3 在该表中不起作用.

复位后, 除 P1.2 和 P1.3 外这些引脚都是准双向口.

复位后, P1.2 和 P1.3 为开漏用于 I2C 接口.

N79E815 每个 I/O 口被寄存器 P3M1 的 P(n)S 位选作 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 Port(n).

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, 时钟输出的频率为片上 RC 时钟或外部振荡器时钟除 4.

9.1 准双向输出配置

标准 **N79E815** 系列的端口都为“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, 有弱驱动能力, 允许外部器件拉低. 当引脚被拉低, 有强驱动能力, 可吸收大电流. 在准双向 I/O 结构中, 有三个上拉三极管. 以适应不同的应用. 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”源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. 如果引脚在逻辑 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 sink 电流 (larger than I_{TL}) 以克服“弱上拉”和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”. 当端口电平由逻辑 0 转为逻辑 1 时, 这种上拉用于在准双向口引脚上加速由低到高的传输. 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.

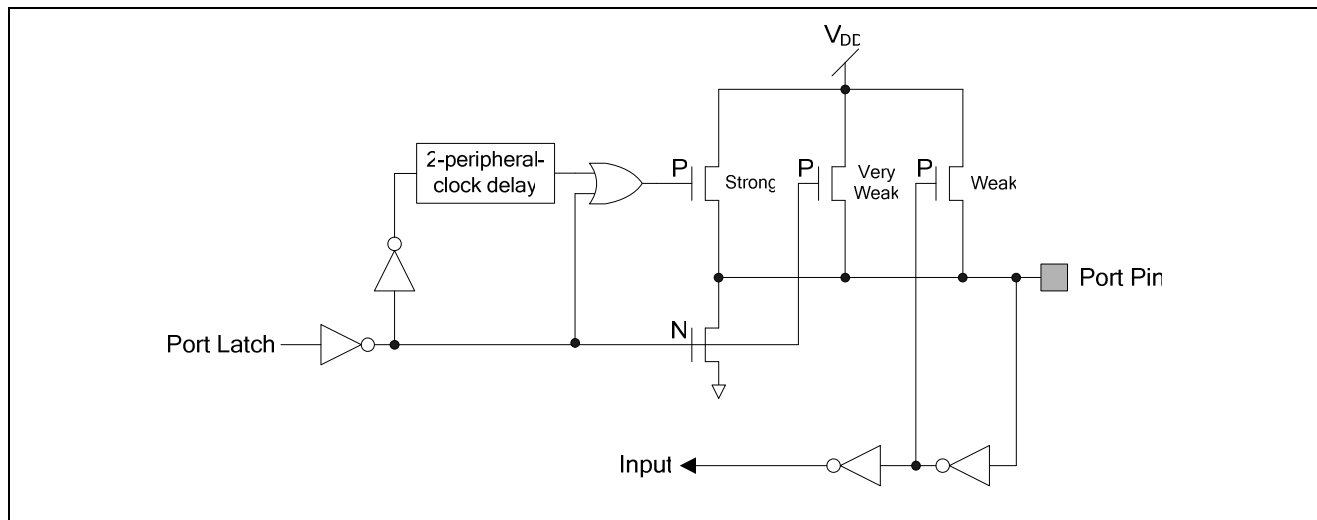


图 9-1. 准双向I/O 结构

9.1.1 读-修改-写

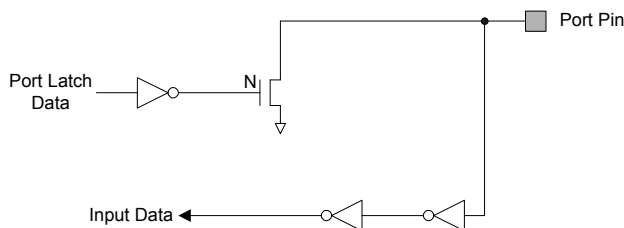
标准的8051指令集, 读-修改-写指令应该特别注意..

指令	描述
ANL	Logical AND. (ANL Px,A and ANL Px,direct)
ORL	Logical OR. (ORL Px,A and ORL Px,direct)
XRL	Logical exclusive OR. (XRL Px,A and XRL Px,direct)
JBC	Jump if bit = 1 and clear it. (JBC Px.y,LABEL)
CPL	Complement bit. (CPL Px.y)
INC	Increment. (INC Px)
DEC	Decrement. (DEC Px)
DJNZ	Decrement and jump if not zero. (DJNZ Px,LABEL)
MOV	Px.y,C Move carry bit to Px.y.
CLR	Px.y Clear bit Px.y.
SETB	Px.y Set bit Px.y.

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上拉，当端口锁定为逻辑0时，仅驱动下拉电阻。用作逻辑输出，端口以该方式配置必须有外部上拉，典型的作法是由电阻拉到 V_{DD} 。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。



Open Drain Output

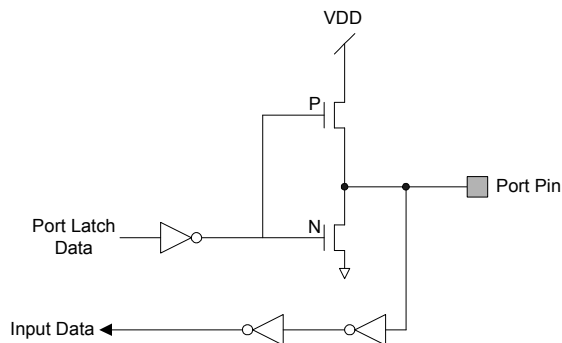
图 9-2 开漏输出

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。推挽输出模式用于当许多源电流需要从端口输出时。推挽端口配置如图0-3所示。P1.2 (SCL) 和 P1.3 (SDA)不能配置成推挽结构。P1.2 (SCL) 和 P1.3 (SDA)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，如果接晶振时，端口引脚P3.0 和 P3.1 禁止作输入和输出。见振荡器章节的描述。

复位后，端口引脚驱动为高，为准双向模式，则不能有较大的源电流。N79E815 系列的输出可用作驱动40 mA sink LED 输出。然而，所有端口的输出电流不能超过总的最大输出电流。

N79E815 系列的所有端口引脚有slew rate控制输出。这可以限制因输出信号快速切换产生的噪声。slew rate由工厂设置大概为10 ns上升和下降一次。寄存器P3M1的该位不能用于控制配置P3.1 和 P3.0作其他用途。这些位可以使能每个I/O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。N79E815 系列的每个I/O 端口可被选作TTL 电平输入或史密特输入。P1.2 和 P1.3 可被选作TTL电平输入或史密特触发输入。



Push-Pull Output

图 9-3 推挽输出

9.4 输入配置

配置成该模式时; 该端口只能输入. 该引脚就为高阻抗.

P0 – Port 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

Address: 80H

reset value: 1111 1111b

位	名称	描述
7:0	P0[7:0]	Port 0. 端口 1 是8位准双向I/O 口.

P1 – Port 1 (可位寻址)

7	6	5	4	3	2	1	0
P17	P16	P15	P14	P13	P12	P11	P10
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: 90H

reset value: 1111 1111b

位	名称	描述
7:0	P1[7:0]	Port 1. 端口 1 是8位准双向I/O 口.

**P2 – Port 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

Address: A0H

reset value: 1111 1111b

位	名称	描述
7:0	P2[7:0]	Port 2. 端口 2 是8位准双向I/O口。

P3 – Port 3 (可位寻址)

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

Address: B0H

reset value: 1111 1111b

位	名称	描述
7:2	-	保留
1	P3.1	X1 或 I/O.
0	P3.0	X2 或 CLKOUT 或 I/O.

**P0M1 – Port 0 输出模式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

Address: B1H

reset value: 0000 0000b

P0M2 – Port 0 输出模式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

Address: B2H

reset value: 0000 0000b

P1M1 – Port 1 输出模式1

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

Address: B3H

reset value: 0000 0000b

P1M2 – Port 1 输出模式2

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

Address: B4H

reset value: 0000 0000b

P2M1 – Port 2 输出模式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

Address: B5H

reset value: 0000 0000b

P2M2 – Port 2 输出模式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

Address: B6H

reset value: 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
0	1	推挽
1	0	输入(高阻)

1	1	开漏
---	---	----

P3M1 – Port3输出模式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

Address: 96H

reset value: 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.
3	T1OE	定时器1溢出, P0.7引脚toggled. 输出频率是定时器1溢出率的一半.
2	T0OE	定时器1溢出, P1.2引脚toggled. 输出频率是定时器1溢出率的一半.
1	P3M1.1	控制P3.1输出配置.
0	P3M1.0	控制P3.0输出配置.

P3M2 – Port3输出模式2

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

Address: 97H

reset value: 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	参见 P2M2 SFR.
0	P3M2.0	


PADIDS – Port0 禁止数字输入

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

Address: 97H

reset value: 0000 0000b

位	名称	描述
7:0	PADIDS.x	1: 禁止端口0的数字功能. 0: 使能端口0的数字功能.

10 定时器/计数器 (定时器)

N79E815 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

N79E815 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toggle 输出使能. 定时器溢出toggle输出可配置成自动toggle T0 或 T1 引脚输出 (当定时器发生溢出时).

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电平负跳变 (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.

由TMOD寄存器中的 C/\bar{T} 位来确定定时器/计数器以何种方式工作. 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.

N79E815 系列可以像8051/52家族一样工作, 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	-	-	-

Address: 8EH

reset value: 0000 0000b

位	名称	描述
7:5	-	保留

位	名称	描述
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

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

Address: 89H

reset value: 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/ \bar{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. <table border="1" style="margin-left: 40px;"> <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>Mode 0: 8-bit Timer/Counter with 5-bit pre-scalar (TL1[4:0])</td> </tr> <tr> <td>0</td> <td>1</td> <td>Mode 1: 16-bit Timer/Counter</td> </tr> <tr> <td>1</td> <td>0</td> <td>Mode 2: 8-bit Timer/Counter with auto-reload from TH1</td> </tr> <tr> <td>1</td> <td>1</td> <td>Mode 3: Timer 1 halted</td> </tr> </tbody> </table>	M1	M0	定时器 1 模式	0	0	Mode 0: 8-bit Timer/Counter with 5-bit pre-scalar (TL1[4:0])	0	1	Mode 1: 16-bit Timer/Counter	1	0	Mode 2: 8-bit Timer/Counter with auto-reload from TH1	1	1	Mode 3: Timer 1 halted
M1	M0		定时器 1 模式														
0	0		Mode 0: 8-bit Timer/Counter with 5-bit pre-scalar (TL1[4:0])														
0	1		Mode 1: 16-bit Timer/Counter														
1	0	Mode 2: 8-bit Timer/Counter with auto-reload from TH1															
1	1	Mode 3: Timer 1 halted															
4	M0																
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/ \bar{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T0的下降沿递增															

位	名称	描述															
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>0</td> <td>Mode 0: 8-bit Timer/Counter with 5-bit pre-scalar (TL0[4:0])</td> </tr> <tr> <td>0</td> <td>1</td> <td>Mode 1: 16-bit Timer/Counter</td> </tr> <tr> <td>1</td> <td>0</td> <td>Mode 2: 8-bit Timer/Counter with auto-reload from TH0</td> </tr> <tr> <td>1</td> <td>1</td> <td>Mode 3: TL0 as a 8-bit Timer/Counter and TH0 as a 8-bit Timer</td> </tr> </tbody> </table>	M1	M0	定时器 0 模式	0	0	Mode 0: 8-bit Timer/Counter with 5-bit pre-scalar (TL0[4:0])	0	1	Mode 1: 16-bit Timer/Counter	1	0	Mode 2: 8-bit Timer/Counter with auto-reload from TH0	1	1	Mode 3: TL0 as a 8-bit Timer/Counter and TH0 as a 8-bit Timer
M1	M0	定时器 0 模式															
0	0	Mode 0: 8-bit Timer/Counter with 5-bit pre-scalar (TL0[4:0])															
0	1	Mode 1: 16-bit Timer/Counter															
1	0	Mode 2: 8-bit Timer/Counter with auto-reload from TH0															
1	1	Mode 3: TL0 as a 8-bit Timer/Counter and TH0 as a 8-bit Timer															

TCON – 定时器 0 和 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

Address: 88H

reset value: 0000 0000b

位	名称	描述
7	TF1	<p>定时器 1 溢出标志.</p> <p>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。软件也可对该位置位或复位。</p>
6	TR1	<p>定时器 1 启动控制.</p> <p>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.</p> <p>1 = 使能定时器1.</p>
5	TF0	<p>定时器 0 溢出标志.</p> <p>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。软件也可对该位置位或复位。</p>
4	TR0	<p>定时器 0 启动控制.</p> <p>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.</p> <p>1 = 使能定时器0.</p>

TL0 – 定时器 0 低字节

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

Address: 8AH

reset value: 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位低字节.
-----	----------	-----------------------------------

TH0 – 定时器 0 高字节

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

Address: 8CH

reset value: 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位低字节.

TL1 – 定时器 1 低字节

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

Address: 8BH

reset value: 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位低字节.

TH1 – 定时器 1 高字节

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

Address: 8DH

reset value: 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位低字节.

10.1.1 模式 0 (13-bit 定时器)

模式0下，是13位的定时器/计数器，由8位的THx和TLx的低5位组成，TLx的高3位被忽略。

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/ \overline{T} =0时，定时器/

计数器对时钟周期进行计数， $C/\bar{T}=1$ 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。

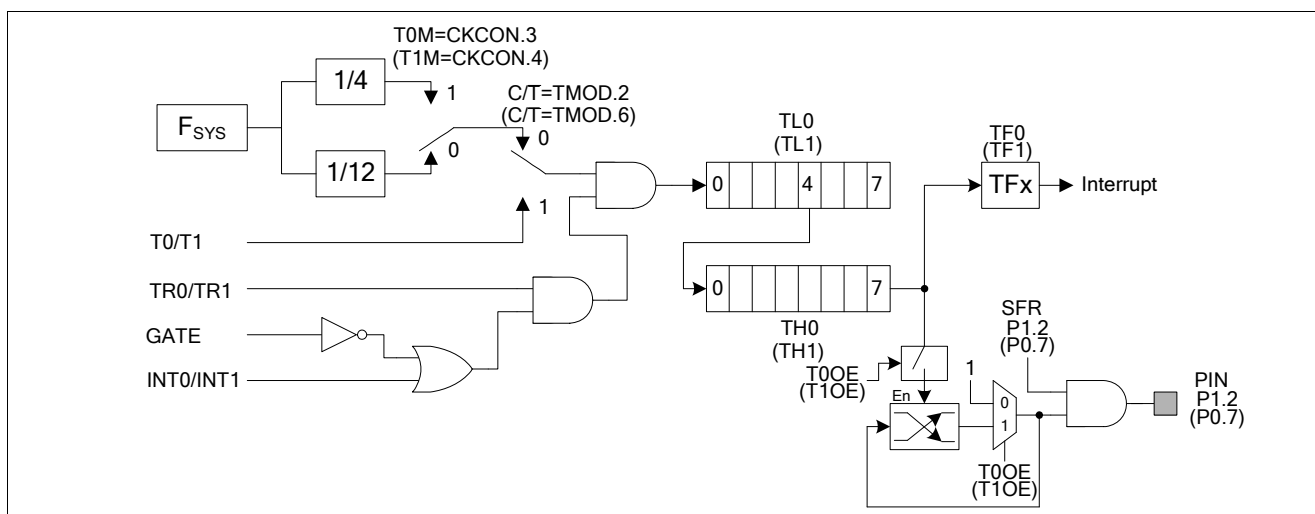


图 10-1. 定时器/计数器0 和 1 的模式 0

10.1.2 模式 1 (16-bit Timer)

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。

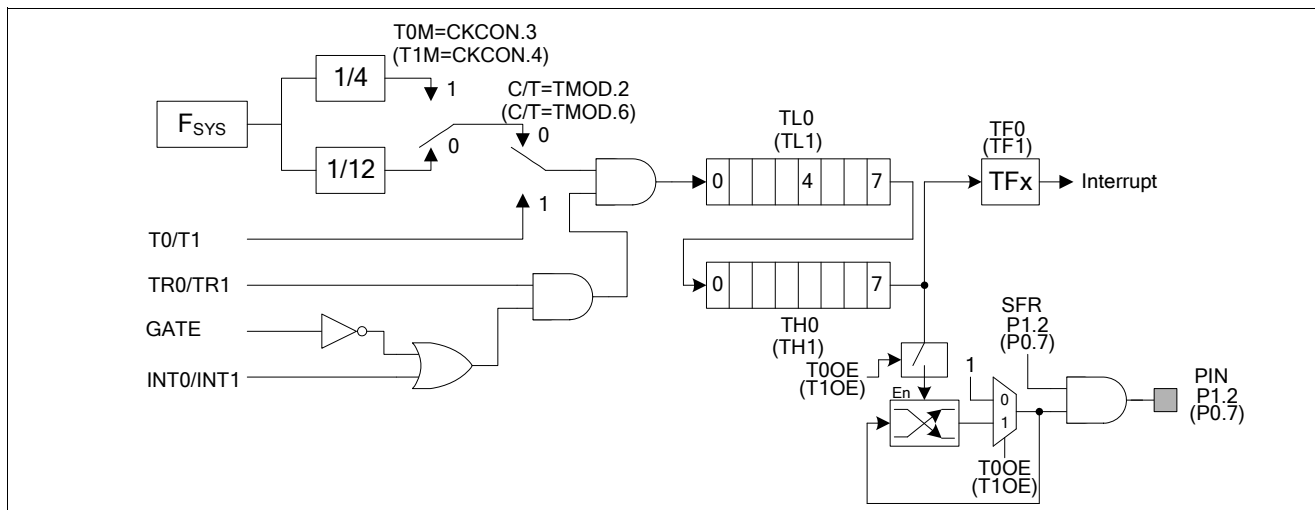


图 10-2. 定时器/计数器0 和 1 的模式 1

10.1.3 模式 2 (8-bit Auto-Reload Timer)

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 $\overline{\text{INTx}}$ 引脚及TRx位，使能计数。GATE 和 $\overline{\text{INTx}}$ 引脚的功能与模式0和1相同。

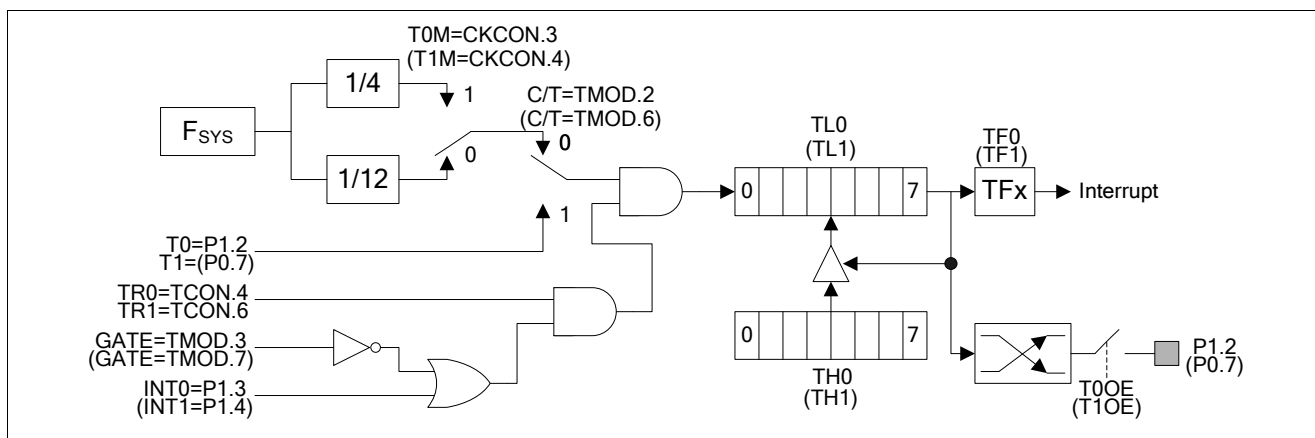


图 10-3. 定时器/计数器0和1的模式 2

10.1.4 模式 3 (Two Separate 8-bit Timers)

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/\overline{T} ，GATE, TR0, $\overline{\text{INT0}}$ 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。

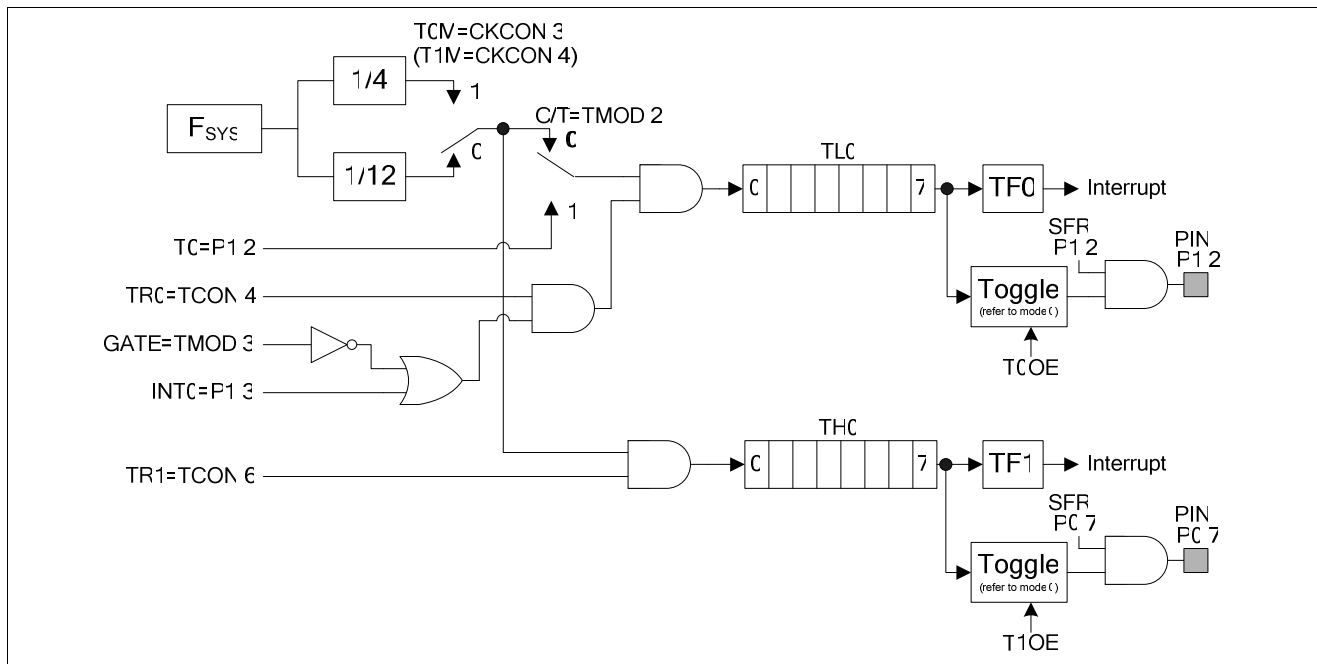


图 10-4. 定时器/计数器0的模式 3

10.2 定时器/计数器2

定时器/计数器2是一个16位的向上计数器, 配置RCOMP2H 和 RCOMP2L, 定时器2可以工作在比较模式和自动重载模式. 3个输入捕获模块使定时器2用于检测输入脉冲的周期或宽度. 3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时器2的时钟源由时钟除频器预分频, 有8种不同尺度. 当TR2 (T2CON.2)为1, 使能时钟, 当TR2为0, 禁止时钟. 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

Address: C8H

reset value: 0000 0000b

位	名称	描述
7	TF2	定时器 2 溢出标志. 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, 该位不能自动由硬件清零, 必须由软件清零.
6:3	-	保留.
2	TR2	定时器 2 运行控制. 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 = 使能定时器2.
1	-	保留.
0	CP/RL2	定时器 2 捕获或重加载选择. 该位选择定时器2的功能是比较功能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	T2DIV2	T2DIV1	T2DIV0	CAPCR	COMPCR	LDTS[1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: C9H

reset value: 0000 0000b

位	名称	描述
7	LDEN	使能自动重加载. 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]	定时器 2 时钟除频. 000 = 定时器 2 时钟除频器为 1/4. 001 = 定时器 2 时钟除频器为 1/8. 010 = 定时器 2 时钟除频器为 1/16. 011 = 定时器 2 时钟除频器为 1/32. 100 = 定时器 2 时钟除频器为 1/64. 101 = 定时器 2 时钟除频器为 1/128. 110 = 定时器 2 时钟除频器为 1/256. 111 = 定时器 2 时钟除频器为 1/512.
3	CAPCR	自动清除捕获 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.
2	COMPCR	比较匹配自动清零. 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.
1:0	LDTs[1:0]	自动加载触发选择. 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.

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							

Address: CAH

reset value: 0000 0000b

位	名称	描述
7:0	RCOMP2L[7:0]	定时器 2 重加载/比较低字节. 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, 保存着重加载的值的低字节.

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							

Address: CBH

reset value: 0000 0000b

位	名称	描述
7:0	RCOMP2H[7:0]	定时器 2 重加载/比较高字节. 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, 保存着重加载的的高字节.

TL2 – 定时器 2 低字节

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

Address: CCH

reset value: 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位低字节部分.

TH2 – 定时器 2 高字节

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

Address: CDH

reset value: 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位高字节部分.

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. 注TH2 和 TL2 分别访问. 强烈推荐用户停止定时器2后读写TH2和TL2. 运行时读写可能会引起不可预见的状况.

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	0	0
自动重加载	0	1
比较	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清 $CP/RL2$ 和 LDEN位进入输入捕获模式. 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, IC1, and IC2 pins) 共享I/O 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. 边沿触发也可引起硬件置位CAPFn (CAPCON0.n). 如果ECPTF (EIE.7)和EA都置位, 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边沿. 这些标志必须由软件清零.

CAPCR (T2MOD.3)用于周期计算. 设置CAPCR 使硬件在TH2与TL2的值已经被捕获后自动清定时器2为0000H..

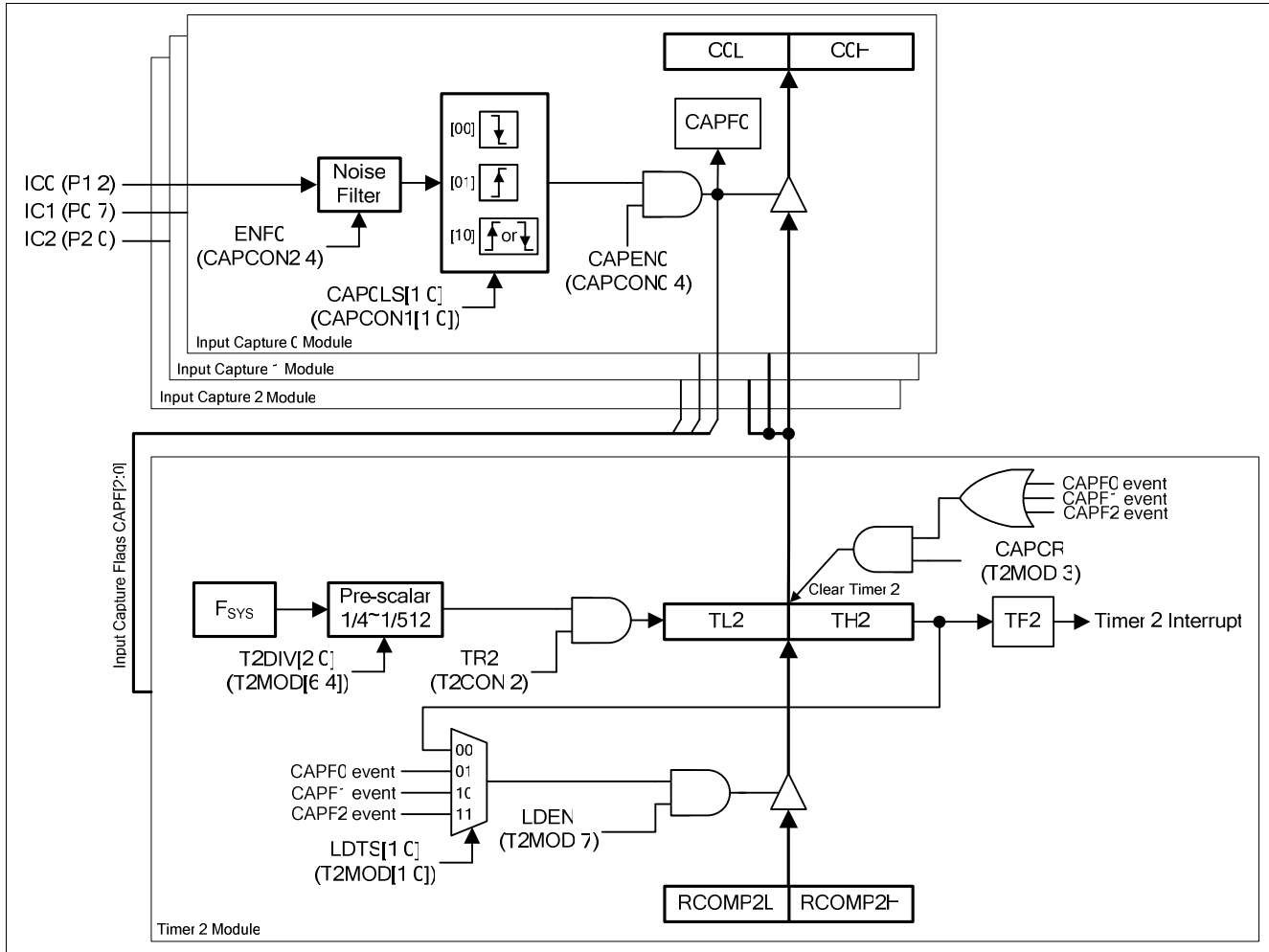


图 10-5. 定时器 2 输入捕获和自动重加载模式功能模块

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

Address: 92H

reset value: 0000 0000b

位	名称	描述
7	-	保留.
6	CAPEN2	使能输入捕获 2. 0 = 禁止输入捕获通道2. 1 = 使能输入捕获通道2.
5	CAPEN1	使能输入捕获 1. 0 = 禁止输入捕获通道1. 1 = 使能输入捕获通道1.

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

CAPCON1 –输入捕获控制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

Address: 92H

reset value: 0000 0000b

位	名称	描述
7:6	-	保留.
5:4	CAP2LS[1:0]	输入捕获2电平选择. 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0电平选择. 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 2

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

Address: 92H

reset value: 0000 0000b

位	名称	描述
7	-	保留.
6	ENF2	使能输入捕获2的噪声滤波.

位	名称	描述
		0 = 禁止输入捕获通道2的噪声滤波. 1 = 使能输入捕获通道2的噪声滤波.
5	ENF1	使能输入捕获1的噪声滤波. 0 = 禁止输入捕获通道1的噪声滤波. 1 = 使能输入捕获通道1的噪声滤波
4	ENF0	使能输入捕获0的噪声滤波. 0 = 禁止输入捕获通道0的噪声滤波. 1 = 使能输入捕获通道0的噪声滤波
3:0	-	保留.

C0L – 捕获 0 低字节

7	6	5	4	3	2	1	0
C0L[7:0]							
r/w							

Address: E4H

reset value: 0000 000B

位	名称	描述
7:0	C0L[7:0]	输入捕获 0 结果低字节. C0L 为输入捕获0的16位捕获结果的低字节寄存器.

C0H – 捕获 0 高字节

7	6	5	4	3	2	1	0
C0H[7:0]							
r/w							

Address: E5H

reset value: 0000 000B

位	名称	描述
7:0	C0H[7:0]	输入捕获 0 结果高字节. C0H为输入捕获0的16位捕获结果的高字节寄存器.

C1L – 捕获 1 低字节

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

Address: E6H

reset value: 0000 000B

位	名称	描述
7:0	C1L[7:0]	输入捕获 1 结果低字节. C1L为输入捕获1的16位捕获结果的低字节寄存器.

C1H –捕获 1 高字节

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

Address: E7H

reset value: 0000 0000B

位	名称	描述
7:0	C1H[7:0]	输入捕获 1 结果高字节。 C1H为输入捕获1的16位捕获结果的高字节寄存器。

C2L –捕获 2 低字节

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

Address: EDH

reset value: 0000 0000B

位	名称	描述
7:0	C2L[7:0]	输入捕获 2 结果低字节。 C2L为输入捕获2的16位捕获结果的低字节寄存器。

C2H –捕获 2 高字节

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

Address: EEH

reset value: 0000 0000B

位	名称	描述
7:0	C2H[7:0]	输入捕获 2 结果高字节。 C2H为输入捕获2的16位捕获结果的高字节寄存器。

10.2.2 自动重加载模式

定时器2可以通过清 $CP/RL2$ 和设置LDEN位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：一旦设置 CAPCR (T2MOD.3)，输入捕获事件仅清TH2和TL2，而不会重加载 RCOMP2H 和 RCOMP2L 的内容。

10.2.3 比较模式

定时器2可以通过设置 $CP/\overline{RL2}$ 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.

设置 COMP2CR (T2MOD.2) 使硬件在比较匹配发生后自动清定时器2计数为 0000H.

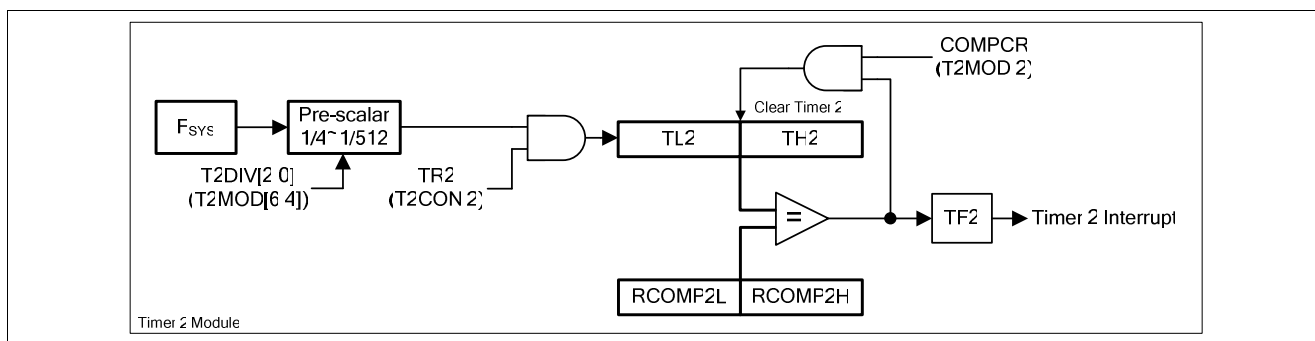


图 10-6. 定时器2比较模式功能框图

11 看门狗定时器(WDT)

N79E815 系列提供一个看门狗定时器，监测系统以提高系统可靠性。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Idle或掉电模式下唤醒，并产生中断事件。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。

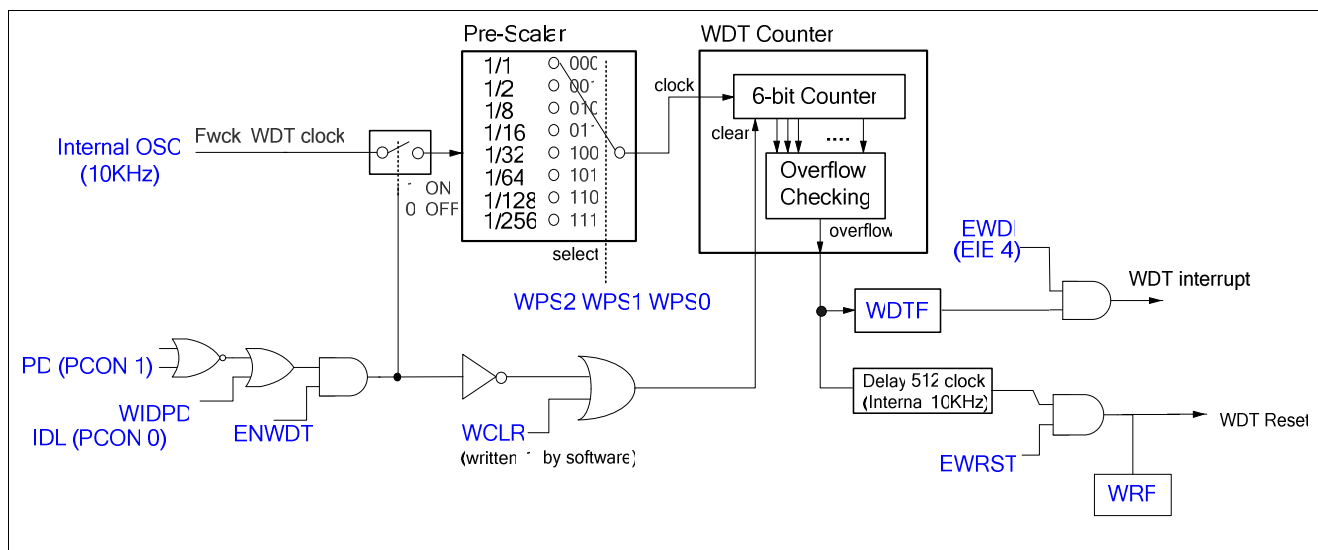


图 11-1 看门狗定时器

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WDTEN(WDCON.7)禁止后看门狗后，将清零该计数器。WDCLR用于复位看门狗定时器。该位自动清除。WDCLR置1，硬件将自动清除。WDTEN置1，看门狗定时器开始计数。通过设定WPS2, WPS1, 和 WPS0 (WDCON[2:0]) 选择溢出时间。一旦选择时间溢出，看门狗定时器将置位中断标志WDTF (WDCON0.5)，看门狗中断使能位在寄存器EIE.4里设置，如果写1到EWRST (WDCON1.0)使能看门狗定时器复位，如果设置WDCLR，在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，看门狗定时器复位标志WDTRF (WDCON0.3)将被置位，该位只有在上电复位后才被清零，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Idle或掉电模式后继续运行

WDT 计数器需特别注意。硬件会自动清 WDT 计数器 在唤醒Idle 或 掉电模式。防止系统未知的复位

WDCON0 – 看门狗定时器控制 (TA protected)

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

Address: D8H

reset value: see [Table 7–2. N79E815 Series SFR Descriptions and Reset Values](#)

位	名称	描述
7	WDTEN	WDTEN: WDT 使能位 0: 上电复位禁止WDT. 1: 上电复位使能WDT.
6	WDCLR	WDT 计数器清零位. 写“1”清WDT计数器为0000H. 注: 该位只写, 不必要通过写“0”清除.
5	WDTF	WDT 中断标志. 当WDT计数器溢出时, 该位将由硬件置位..
4	WIDPD	看门狗运行于Idle和掉电模式下. 该位决定看门狗工作于Idle 还是掉电模式下. 0 = 当CPU在Idle或掉电模式下, 中止WDT 计数器. 1 = 当CPU在Idle或掉电模式下, WDT 计数器保持运行.
3	WDTRF	WDT 复位标志. 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, MCU将立即复位自身. 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.
2:0	WPS[2:0]	WDT 预分频选择. 用于选择WDT定时溢出周期. $\text{WDT 定时溢出周期公式} = \frac{64}{(F_{wck} \times PreScalar)}$, 其中 F_{wck} 为WDT时钟源的频率. 下表为不同 F_{wck} 对应的WDT定时溢出周期示例.

[1] WDTEN is initialized by reloading the inversed value of CWDTEN (CONFIG3.7) after all resets.

[2] WIDPD and WPS[2:0] are cleared after power-on reset and keep unchanged after any other resets.

[3] WDTRF will be cleared after power-on reset, be set after Watchdog Timer reset, and remains unchanged after any other resets.

WDCON1 – 看门狗定时器控制 (TA protected)

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

Address: ABH

reset value: see Table 7-2. N79E815 Series SFR Descriptions and Reset Values

位	名称	描述
0	EWRST	0: 禁止 WDT 复位功能. 1: 使能WDT复位功能.

[1] EWRST仅在上电复位后清零.

看门狗定时溢出间隔公式 $\frac{1}{F_{wck} \times \text{clock divider scalar}} \times 64$. 其中 F_{wck} 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

Address: E8H

reset value: 0000 0000b

位	名称	描述
4	EWDI	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.

表 11-1: 看门狗定时器定时溢出值

(WPS2,WPS1,WPS0)	Pre-Scalar	WDT Interrupt time-out		Reset time-out	
		Number of Clocks	Time	Number of Clocks	Time
(0,0,0)	1/1	2^6	6.4ms	2^6+512	57.6ms
(0,0,1)	1/2	2×2^6	12.8ms	$2 \times 2^6+512$	64ms
(0,1,0)	1/8	8×2^6	51.2ms	$8 \times 2^6+512$	102.4ms
(0,1,1)	1/16	16×2^6	102.40ms	$16 \times 2^6+512$	153.6ms

(1,0,0)	1/32	32x2 ⁶	204.80ms	32x2 ⁶ +512	256ms
(1,0,1)	1/64	64x2 ⁶	409.60ms	64x2 ⁶ +512	460.8ms
(1,1,0)	1/128	128x2 ⁶	819.20ms	128x2 ⁶ +512	870.4ms
(1,1,1)	1/256	256x2 ⁶	1.638s	256x2 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. 适用于电磁干扰等发生程序跑飞等, 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器. 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, 将引起复位. WDCON 有时序写入保护.

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EWDI (EIE.4)和EA置位的情况下, 将发生中断.

在一些低功耗的应用中, CPU常在没有处理事件时处于Idle模式或掉电模式. 看门狗定时器可以起到唤醒的功能. N79E815 系列带有这样的功能. 提供低功耗的内部RC 10kHz. 以下为应用示例.

看门狗将CPU从掉电模式下唤醒的demo code.

```

ORG    0000H
LJMP   START

ORG    005BH
LJMP   WDT_ISR

ORG    0100H
WDT_ISR:
CLR    EA
MOV    TA,#0AAH
MOV    TA,#55H
ORL    WDCON0,#01000000B      ;clear Watchdog Timer counter
SETB   EA
Check_clear1:
MOV    A,WDCON0
JB     A.6,Check_clear1

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
Check_clear:
MOV    A,WDCON0
JB     A.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   EWDT                    ;enable Watchdog Timer interrupt

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

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

```

12 串口 (UART)

N79E815系列提供全双工串口 UART (全双工传输和接受) 模式 1, 2, 和 3. 为用户提供帧错误检测、自动地址识别等附加功能。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) 置 1。

SCON – 串口控制 (bit-addressable)

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

Address: 98H

reset value: 0000 0000b

位	名称	描述
7	SM0/FE	选择串口模式.
6	SM1	<p>SMOD0 (PCON.6) = 0: 见 表 12-1. 串口模式描述 .</p> <p>SMOD0 (PCON.6) = 1: 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>多处理通讯模式选择. 该位功能取决于串口模式.</p> <p>模式 0: 该位选择波特率$F_{SYS}/12$ 和 $F_{SYS}/4$. 0 = 时钟运行在$F_{SYS}/12$ 波特率. 1 = 时钟运行在$F_{SYS}/4$ 波特率.</p> <p>模式 1: 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和接收数据与GIVEN或BROADCAST地址匹配时有效.</p> <p>模式 2 或 3: 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>

位	名称	描述
4	REN	接收使能 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。
3	TB8	第9位传输位 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。
2	RB8	第9位接收位 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	发送中断标志位 发送中断标志: 模式0下该标志由硬件在发送完8位数据后置位, 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	接收中断标志 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, 该位置位会转跳置中断子程运行。该位必须由软件来清除。

表 12-1. 串口模式描述

模式	SM0	SM1	描述	帧位	波特率
0	0	0	同步	8	F _{sys} divided by 12 or by 4[1]
1	0	1	异步	10	Timer 1 overflow rate divided by 32 or divided by 16[2]
2	1	0	异步	11	F _{sys} divided by 64 or 32 [2]
3	1	1	异步	11	Timer 1 overflow rate divided by 32 or divided by 16[2]

[1] While SM2 (SCON.5) is logic 1.

[2] While SMOD (PCON.7) is logic 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

Address: 87H

reset value: see [Table 7-2. N79E815 Series SFR Descriptions and Reset Values](#)

位	名称	描述
7	SMOD	串口波特率加倍使能。 在模式 1, 2, 或 3 串口波特率加倍使能. 定时器1 为波特率时钟源. 见 表 12-1. 串口模式描述 .

位	名称	描述
6	SMOD0	帧错误检测使能。 0 = 禁止帧错误检测。 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							

Address: 99H

reset value: 0000 0000b

位	名称	描述
7:0	SBUF[7:0]	串口数据缓冲。 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。

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。

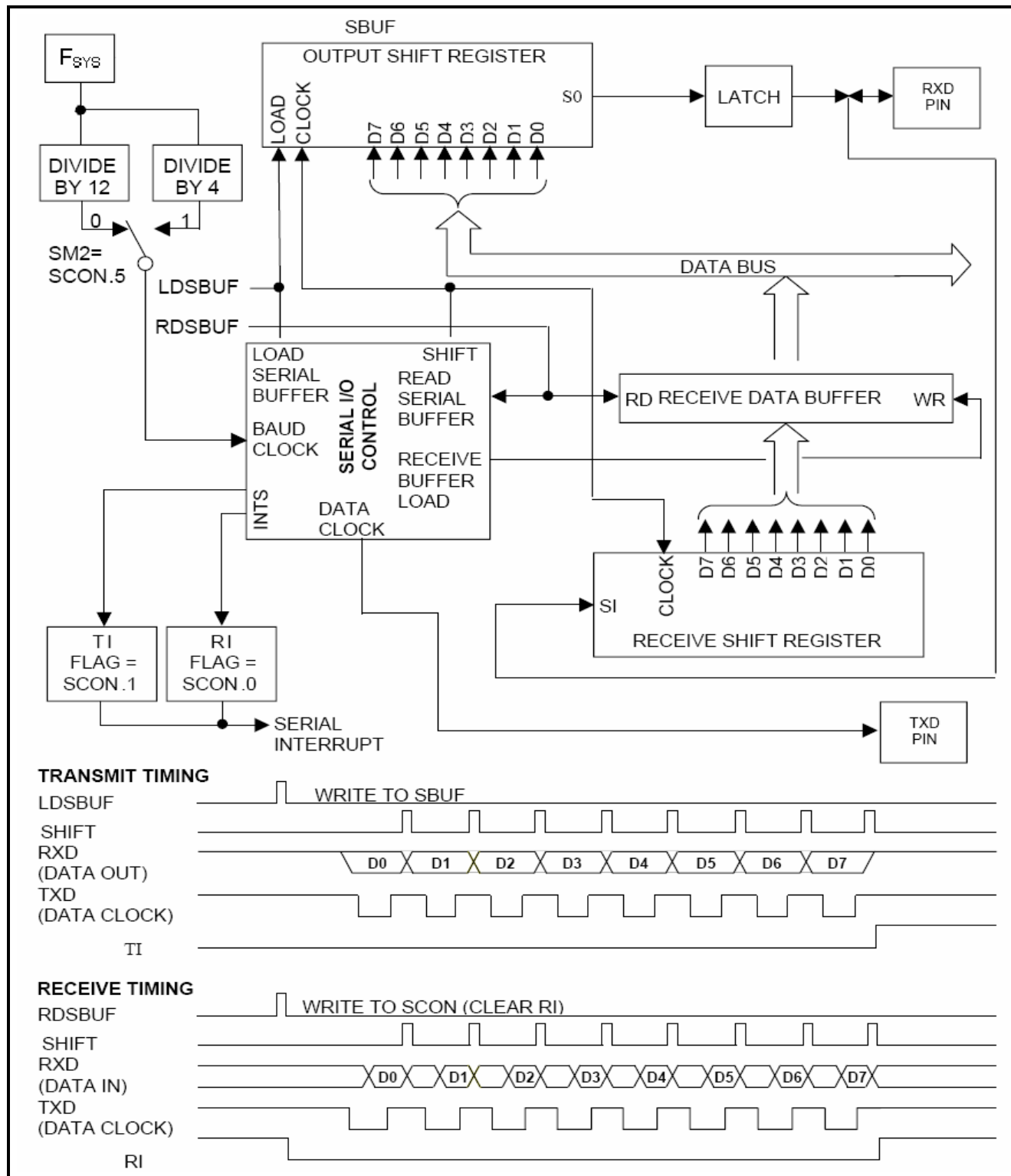


图 12-1. 串口模式0功能框图

数据由RXD线进行收发。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。

SBUF的写将会发送数据，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。传输标志位TI (SCON.1)置1表示1字节传输完成。

当REN (SCON.4)=1且RI=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，RI要由软件清零。

12.2 模式 1

模式 1 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)。图 12-2 模式1的功能框图。

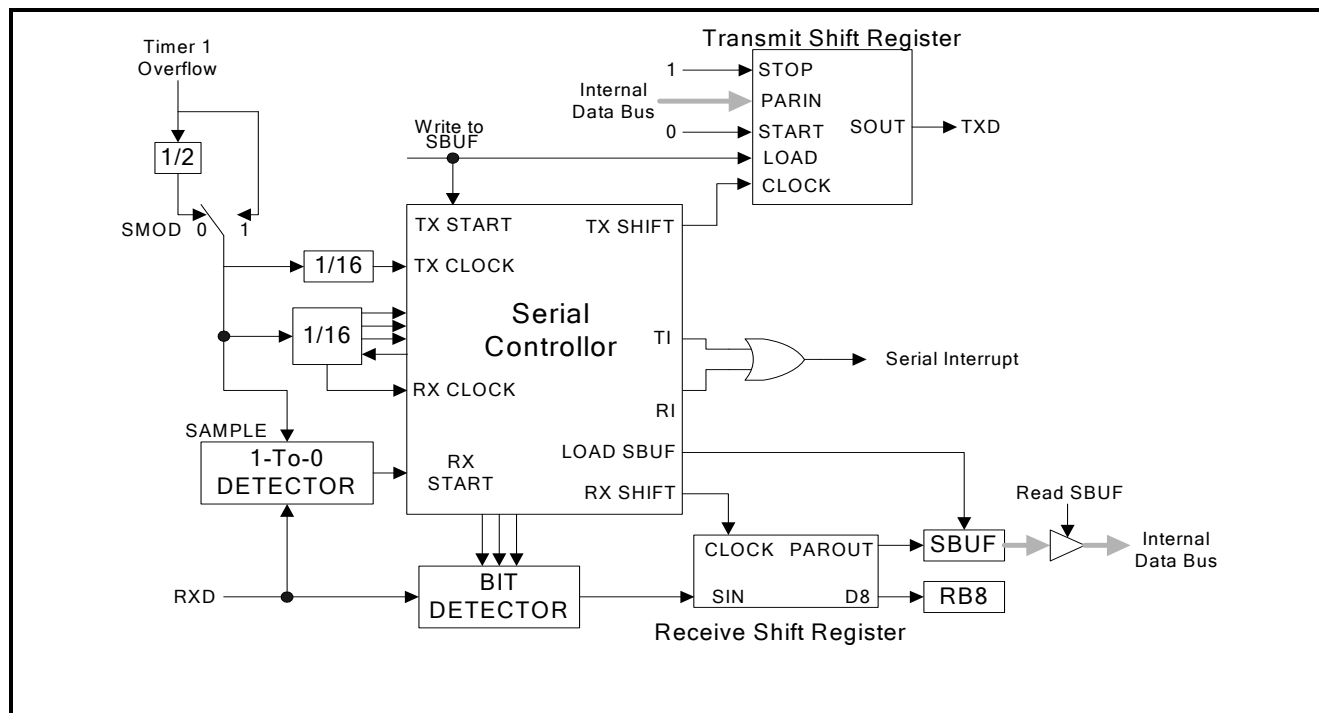


图 12-2. 串口模式1功能框图

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

当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

1. RI (SCON.0) = 0,
2. 任何 SM2 (SCON.5) = 0, 或接受停止位 = 1 当 SM2 = 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将被置位，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（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

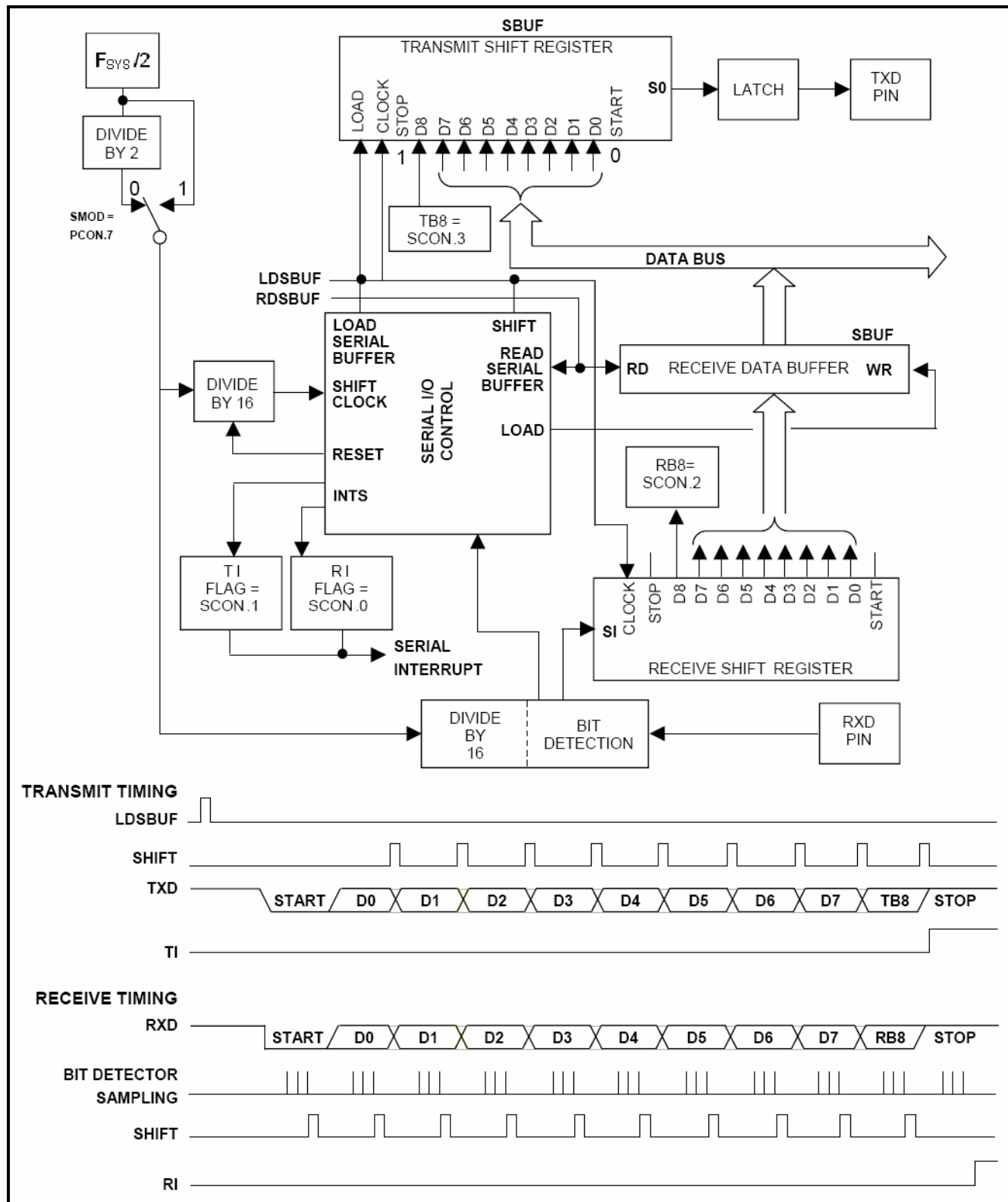


图 12-3. 串口模式2功能框图

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，数据在所选波特率下采样和移入。

1. RI (SCON.0) = 0,
2. 任何 SM2(SCON.5) = 0, 或接受 9th 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，否则丢弃接收到的页数据。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。

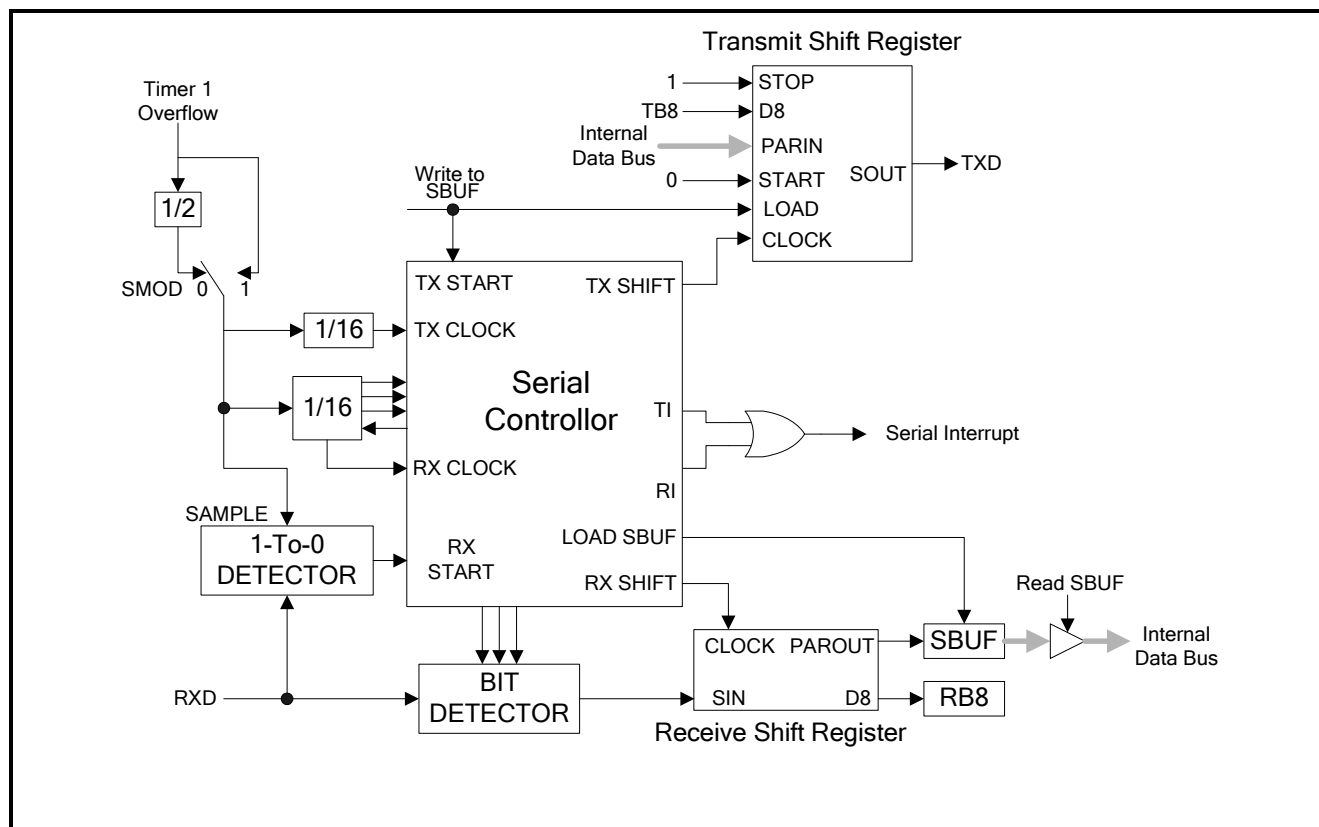


图 12-4. 串口模式3功能框图

12.5 波特率

表 12-2. UART 波特率公式

UART 模式	波特率时钟	波特率
0	振荡器	$F_{SYS} / 12$ or $F_{SYS} / 4$ [1]
2	振荡器	$\frac{2^{SMOD}}{64} \times F_{SYS}$
1 or 3	定时器/计数器1溢出[2]	$\frac{2^{SMOD}}{32} \times \frac{F_{SYS}}{12 \times (256 - TH1)}$ or $\frac{2^{SMOD}}{32} \times \frac{F_{SYS}}{4 \times (256 - TH1)}$ [3]

[1] While SM2 (SCON.5) is set as a logic 1.

[2] Timer 1 is configured as a timer in auto-reload mode (Mode 2).

[3] While T1M (CKCON.4) is set as a logic 1.

注：使用定时器 1 作为波特率发生器，中断需禁止。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，通常配置为定时器自动重载模式。如果定时器 1 作为波特率发生器。特率的值由 TH1 的值决定。

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

TH1 重载值	振荡器频率 (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. UART可以检测帧错误, 并通知软件

SCON.7是FE标志(帧错误标志)(FE_1)。在标准8051种该位是SM0, 但在 N79E875 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 多机通信

N79E815 UART 的多机通讯, 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。被呼唤的从机须清除SM2。

下列步骤配置多机通信:

1. 设置所有设备(主机与从机) 为UART 模式 2 或 3.

2. 从机置 SM2 位 为1

3. 主机传输协议:

- 开始位: 地址, 识别位, 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 9th 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, 模式 1 可用于检测有效的停止位. 若 SM2 置 1, 接受中断将无法产生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UART识别特定地址. 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. 此配置, 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“Broadcast”地址联系. 两个SFRs 用于定义从机地址, SADDR, 和从机地址mask, SADEN. SADEN 用于定义SADDR的哪位被用, 哪位不必关心. SADEN mask 可以与SADDR逻辑AND以创建“Given”地址. 使用“Given”地址允许许多从机被识别.

SADDR – 从机地址

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

Address: A9H

reset value: 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UART多机通信.

SADEN – 从机地址Mask

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

Address: B9H

reset value: 0000 0000b

位	名称	描述
7:0	SADEN[7:0]	从机地址 mask. 该字节是mask 字节.

参考下例.



例 1, slave 0:

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

例 2, slave 1:

SADDR = 11000000b
SADEN = 11111110b
 Given = 1100000Xb

在上例中，SADDR 是相同的，SADEN 数据用于区分两个从机。从机 0 请求第 0 位为 0 并忽略第 1 位。从机 1 请求第 1 位为 0，并忽略第 0 位。当从机 1 第 1 位请求 0 时，一个唯一的从机 0 的地址就是 1100 0010。当从机 1 第 0 位请求 1 时，一个唯一的地址就是 11000001b。两个从机可以同时被选中（当 bit 0 = 0 (for slave 0) 和 bit 1 = 0 (for slave 1) 时）。因此，两个的地址都是 11000000b。

下例是更复位的系统，选择从机 1 和从机 2，不选从机 0:

例 1, slave 0:

SADDR = 11000000b
SADEN = 11111001b
 Given = 11000XX0b

例 2, slave 1:

SADDR = 11100000b
SADEN = 11111010b
 Given = 11100X0Xb

例 3, slave 2:

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

上例 3 个从机的区别在低 3 位地址。Slave 0 请求 bit 0 = 0，可被唯一确定的地址为 11100110b。Slave 1 请求 bit 1 = 0，可被唯一确定的地址为 11100101b。从机 2 请求 bit 2 = 0，可被唯一确定的地址为 11100011b。要选择 Slaves 0 和 1 不选 Slave 2 则使用地址 11100100b，当 bit 2 = 1 不选 slave 2。每个从机的“Broadcast”地址通过 SADDR 与 SADEN 的 OR 得到。结果中的 0 “不关心”。多数情况下，1 解释为 “不关心”，广播地址为 FFH。

复位后，SADDR 和 SADEN 初始化为 00H。产生了所有 “不关心” 的 “Given” 地址，也产生了所有 XXXXXXXXb (all “don’t care” bits) 的 “Broadcast” 地址。这有效地禁止自动地址模式并允许微控制器使用标志 UART，而不使用该特征。

13 串行外围设备接口 (SPI)

13.1 特征

N79E815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MCUs 外设之间 EEPROM, LCD 驱动, D/A 转换. 提供主机 从机模式, 速度可达 $F_{PERIPH}/16$ 主机模式 和 $F_{PERIPH}/4$ 从机模式, 传输2完成 写 冲突标志位。在多主机系统中, SPI 支持主机模式侦测 保护主机模式冲突。

13.2 功能描述

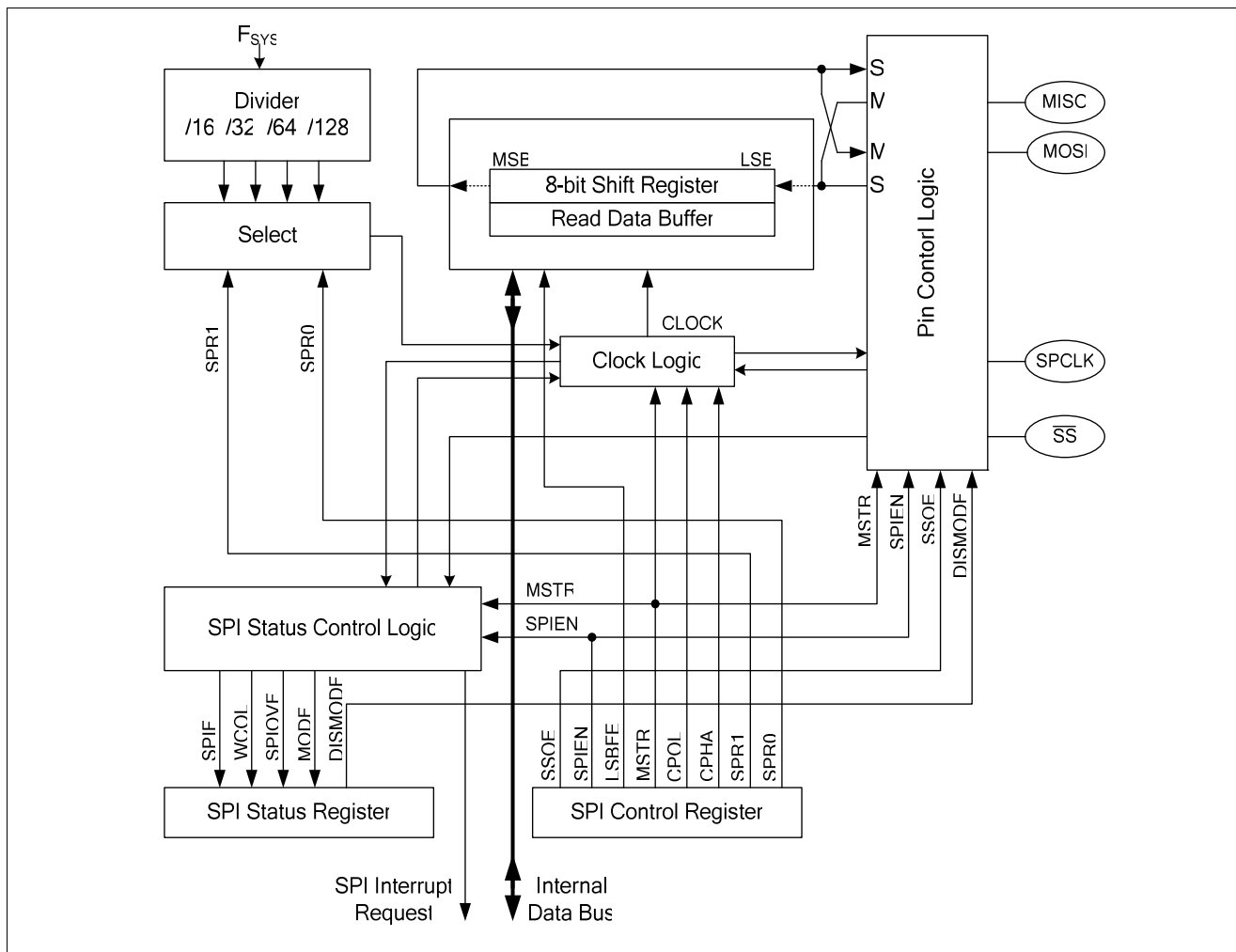


图 13-1. SPI 框图

图 13-1 为SPI框图. 是SPI模块图表, 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。

四个管脚表示 SPI 界面 Master-In/Slave-Out (MISO), Master-Out/Slave-In (MOSI), Shift Clock (SPCLK), 和 Slave Select (\overline{SS}). 主机模式。MOSI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, 移位时钟用于同步数据移动。

每路从机外设通过设定 从机选择脚 (\overline{SS}). 该信号脚需要保持低。当 \overline{SS} 为高, 从机状态将被禁止。若为多从机模式, 在同一时刻 需只有一个从机保持此模式, \overline{SS} 脚只是普通 I/O. , \overline{SS} 可用于主机模式错误侦测功能 ([Section 13.7“模式错误检测” on page 83](#)) . N79E815提供自动栓牢 \overline{SS} 在每字节传输。

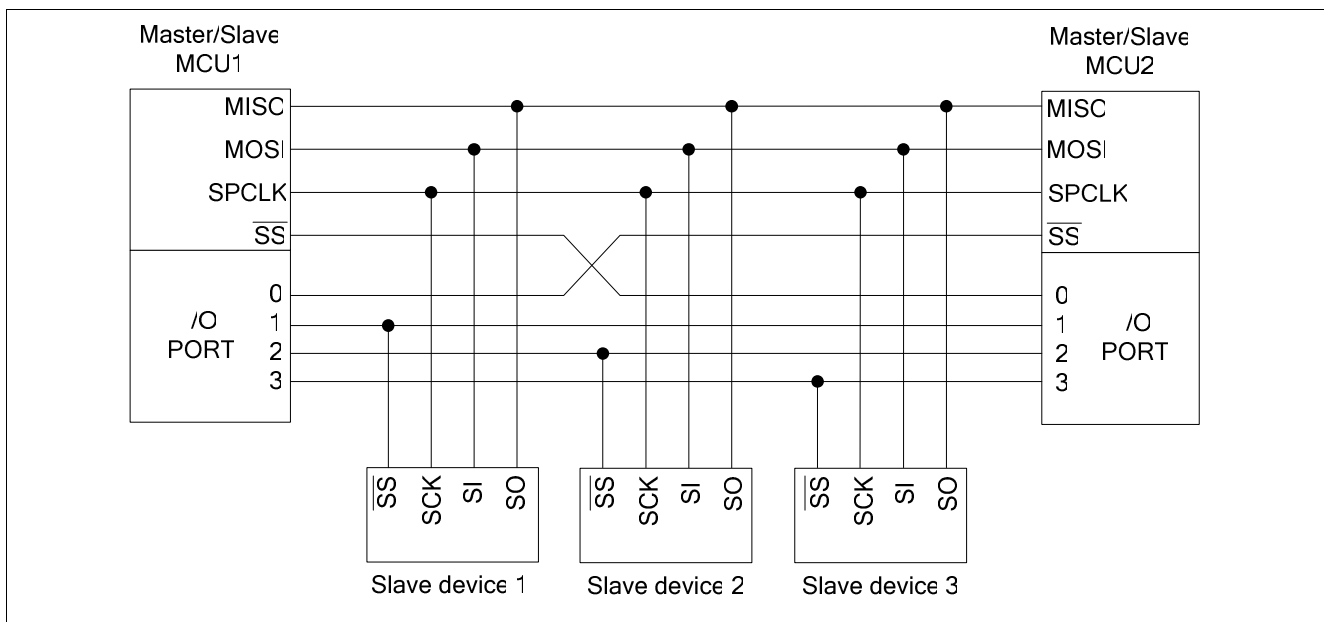


图 13-2. SPI 多主机, 多从机连接

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。

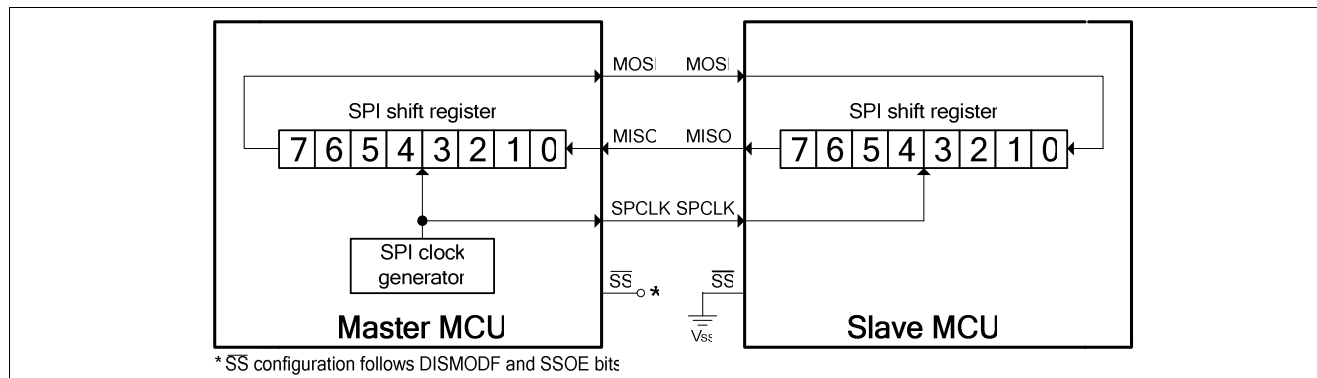


图 13-3. SPI单主机 从机互连

图 13-3 表示SPI 模块简图，单主机/从机互连。。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)..

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

Address: F3H

reset value: 0000 0000b

位	名称	说明
7	SSOE	<p>从机选择输出使能位。</p> <p>该位搭配DISMODF (SPSR.3) 用于定义\overline{SS}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 = \overline{SS} 作为普通 I/O.</p> <p>1 = \overline{SS} 选择外部从机驱动 自动拉低，总线进入空闲模式时变高。</p>

6	SPIEN	<p>SPI 使能.</p> <p>0 = 禁止SPI功能.</p> <p>1 = 使能SPI功能.</p>																				
5	LSBFE	<p>LSB – 优先使能.</p> <p>0 = SPI优先传输MSB data.</p> <p>1 = SPI优先传输LSB data.</p>																				
4	MSTR	<p>使能主机模式.</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>SPI时钟极性位选择.</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>SPI 时钟相位选择.</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	SPI 时钟速率选择.																				
0	SPR0	<p>该两位选择四级SPI时钟除频.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>SPR1</th> <th>SPR0</th> <th>Divider</th> <th>SPI clock rate</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>以上时钟频率基于$F_{PERIPH} = 20\text{MHz}$ 条件计算.</p>	SPR1	SPR0	Divider	SPI clock rate	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	Divider	SPI clock rate																			
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																			

表 13–1. 从机选择引脚配置

DISMODF	SSOE	主机模式 (MSTR = 1)	从机模式(MSTR = 0)
0	x	$\overline{\text{SS}}$ 输入 模式侦测	$\overline{\text{SS}}$ 输入 从机选择
1	0	普通 I/O	

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	-	-	-

Address: F4H

reset value: 0000 0000b

位	名称	描述
7	SPIF	SPI中断完成标志. 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	写冲突位. 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	SPI 过载标志. 该位表示过载事件, 一旦发生过载事件, 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	模式错误中断状态标志. 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 and 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	禁止模式错误检测. 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							

Address: F5H

reset value: 0000 0000b

位	名称	描述
7:0	SPDR[7:0]	串行外设数据。 该字节为SPI总线上传输或接收的数据。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 操作模式**13.4.1 主机模式**

SPI工作模式可以通过位MSTR (SPCR.4)配置为主机模式或从机模式，在主机模式，只有主机的SPI设备可以开始发送数据。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。用户可清SPIF，并从SPDR读出数据。

13.4.2 从机模式

MSTR为0，SPI工作在从机模式。当作为从机模式时，SPCLK管脚变为输入，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。

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。当在串行总线上传送或接收数据时用SPDR，只有对该寄存器的写可以开始传送或接收一位，并且只发生在主机设备。在一个数据位传送完成之后，主机和从机的SPIF位都置位。

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。

在从机模式下，数据依靠主机的MOSI脚向从机的MISO脚传输。数据受主机的移位寄存器SPCLK控制。移位寄存器接收到1组数据后，将写数据进缓存区并且SPIF位置位。SPDR读取缓存区数据。为了防止数据丢失，从机需读PDR和SPIF的第一位。在第二次传输数据前。

13.5 时钟格式和数据传输

时钟极性控制位CPOL选择为高或是低对传送格式没有太大的影响。通过采样SCK边缘上奇数数据或偶数数据的数目，时钟相位控制位CPHA选择两种不同的传送协议中的一种，因此这两位可以选择SPI系统的4种不同的时钟形式。

主机SPI设备和正在通信的从机设备的时钟的极性和相位必须保持一致。

当CPHA=0， \overline{SS} 线在每个连续的串行字节之间必须取消和重新确定。当 \overline{SS} 为低时如果从机向SPI数据寄存器SPDR写数据，结果会是写冲突错误。当CPHA=1， \overline{SS} 在连续传输之间保持低电平。

当CPHA = 0，数据是SPCLK第一个边缘上采样的样品，当CPHA=1时，数据是SPCLK第二个边缘上采样的样品。在改变CPOL设置之前，SPE必须首先禁止

SPI 有时钟极性位 CPOL (SPCR.3) 和时钟相位CPHA (SPCR.2)。图 13-4. SPI 时钟格式 为CPOL和CPHA组成的四种不同时钟格式。在SPI Idle状态，CPOL 位表示SPCLK 的电平。CPHA位定义MOSI和MISO采边沿。相同的系统，CPOL 和 CPHA 在主机和从机设备中应该一样的。

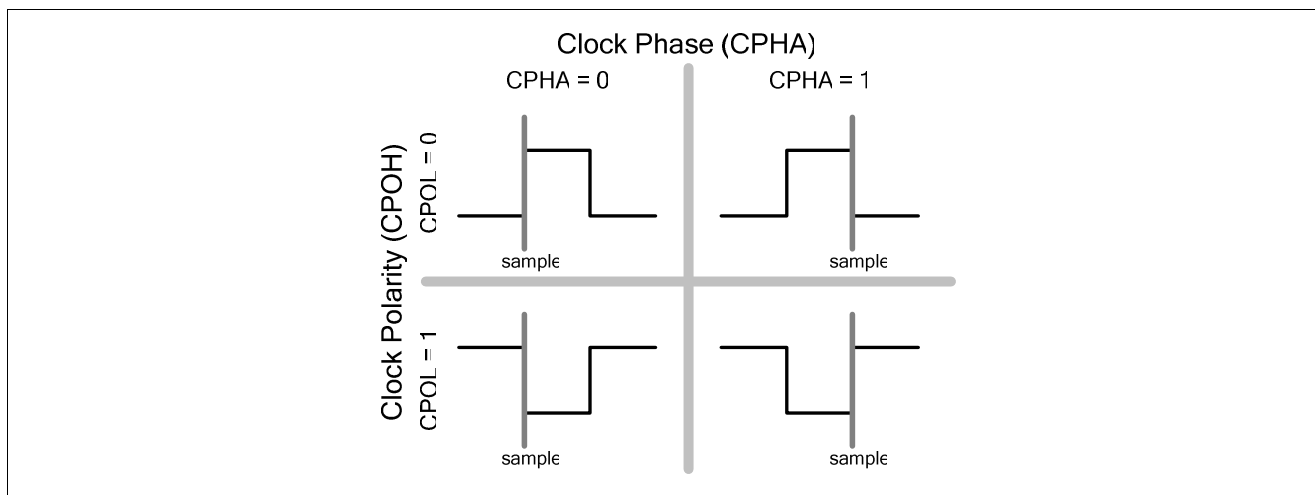


图 13-4. SPI 时钟格式

在 SPI模式下，主机发动传输.如果SPI 选择为主机模式 (MSTR= 1) 并且使能 (SPIEN= 1)，写SPI 数据寄存器 (SPDR) 通过主机驱动SPI的时钟和数据传输.在一组数据移位完成并接收到另外一组数据后.SPI 时钟将停止并且主机和从机的SPIF (SPSR.7) 位置位.如果SPI中断位ESPI(EIE.6)使能.并且总中断(EA=1),将产生SPI 中断.

在从机模式下,SS需特别注意.详见图 13-4. SPI 时钟格式. 当CHPA= 0,起始SPCLK为MSB的边缘滤波 (例如 LSBFE=0 MSB起始), 因此从机在第一个SPCLK边缘必须移出MSB数据. SS的下降沿用于准备MSB到MISO

线), 在两个连续的字节之间SS脚需先置高电平, 然后置低电平. 如果从机在SS为低时写入SPI数据寄存器 (SPDR),将产生写冲突错误.

当CPHA=1, 采样边沿为SPCLK的第二个时钟边沿. 从机采用第一个时钟用来将MSB移出, 而非SS下降沿. 因此SS可以在传输中保持低电平. 此格式最适用于单主机和单从机的传输. 只要CPHA=1的时钟模式保持, 从机可以始终保持将SS拉低到Vss.

注: SPI在(SPIEN = 1)使能前, 需要先进行初始化配置. 否则传输过程中, 对LSBFE, MSTR, CPOL, CPHA 或 SPR[1:0]修改, 将会影响传输过程, 强迫SPI进入idle状态. 所以在配置之前, 必须先禁止SPIEN.

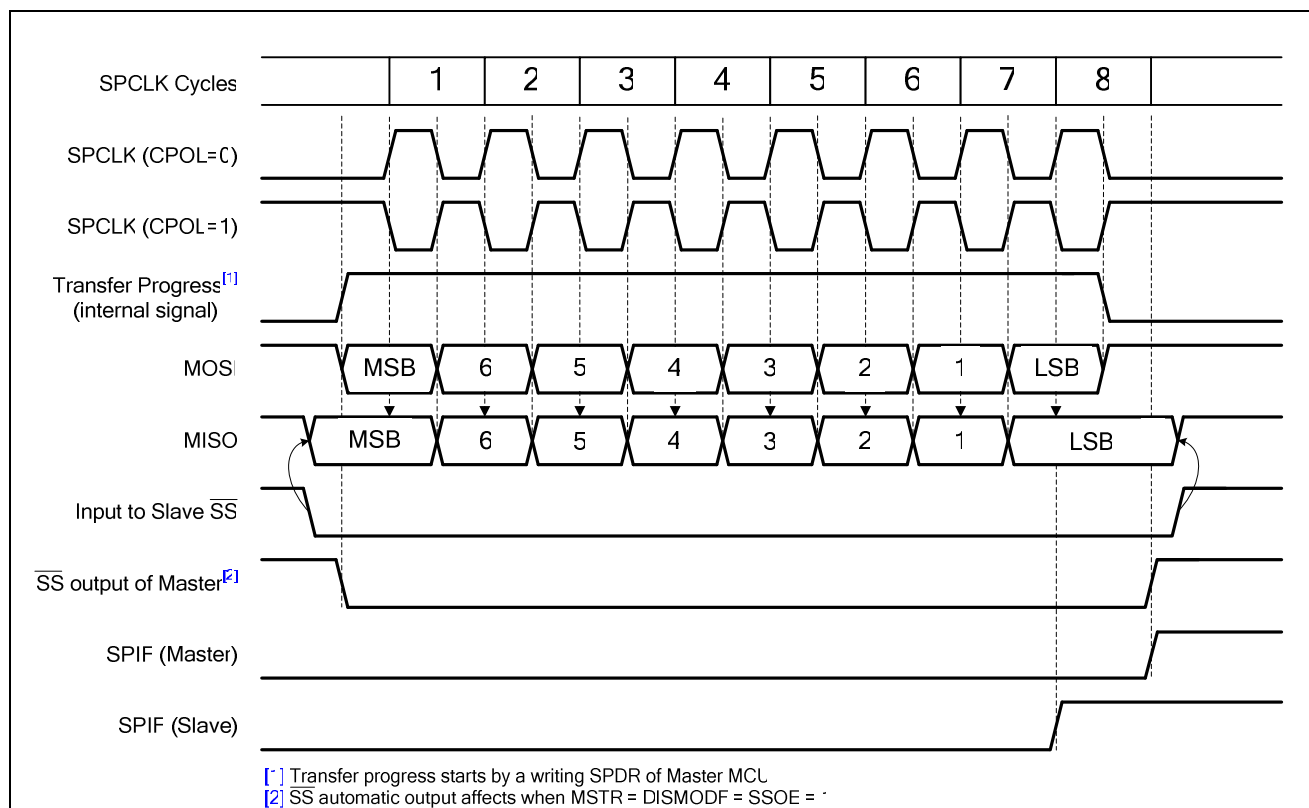


图 13-5. SPI 时钟和数据格式 (CPHA = 0)

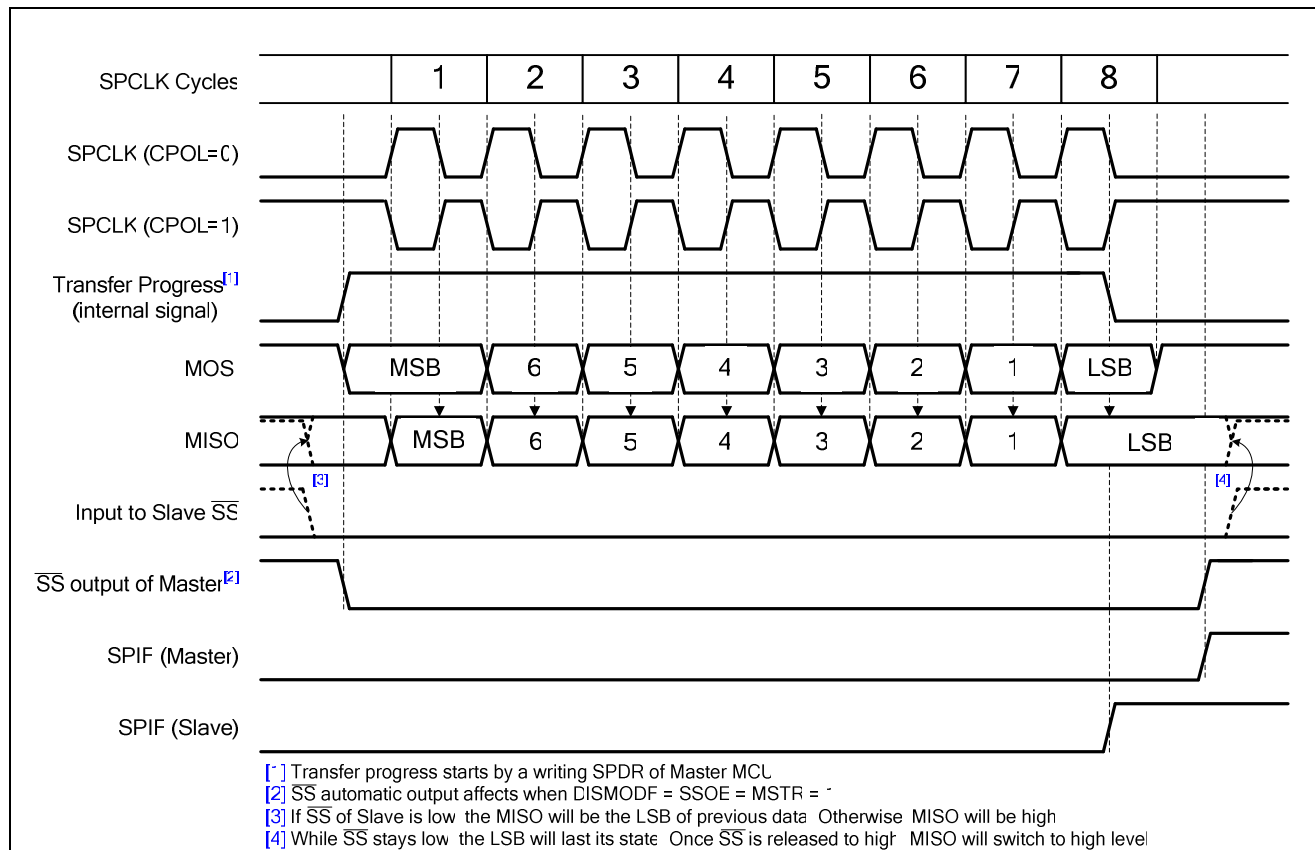


图 13-6. SPI 时钟和数据格式 (CPHA = 1)

13.6 从机选择脚配置

N79E815 SPI 提供可选择的 \overline{SS} 用于不同系统. 在主机设备能够和从机设备交换数据之前, 从机选择 (\overline{SS}) 输入设备必须确定. \overline{SS} 有三种不同功能. 设定 DISMODF (SPSR.3) 和 SSOE (SPCR.7). 默认状态下 DISMODF = 0. 在数据传输前和数据传输中必须保持低电平. 主机 \overline{SS} 必须保持高电平. 无论外围串行设备是否工作, 其它三个管脚都用于 SPI. SSOE = 1 从选择信号自动产生. 选择从机模式. SSOE = 0 DISMODF = 1, \overline{SS} 不应用于 SPI 做为普通 I/O 口.

13.7 模式错误检测

模式错误检测应用于超过一路的 SPI 设备竞争为主机. 包括数据竞争, 侦测模式下. 一旦发现 \overline{SS} 拉低. 预示 SPI 设备有地址的从机向主机竞争. MSTR 和 SPIEN 控制位. 清除硬件禁止 SPI, 模式侦测标志位 MODF (SPSR.4) ESPI (EIE .6) 和 EA 使能. 将产生中断.

13.8 写冲突检测

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。

13.9 过载错误

为了接收数据，SPI具有双数据缓冲接收。接受到的数据传输到数据缓冲区，移位接收第2位连续位。接收到的数据需从SPDR中读取在下一数据完成移位之前。只要从数据寄存器中读取第一位并且在下一数据完成传输之前，将不会产生过载状态。在此条件下，第2位数据将不能写入数据寄存器之前的数据将继续保留。一旦过载发生，SPIOVF (SPSR.5)由硬件置位。并产生中断请求。图13-7. SPI过载波形表示数据接收和过载检测的关系。

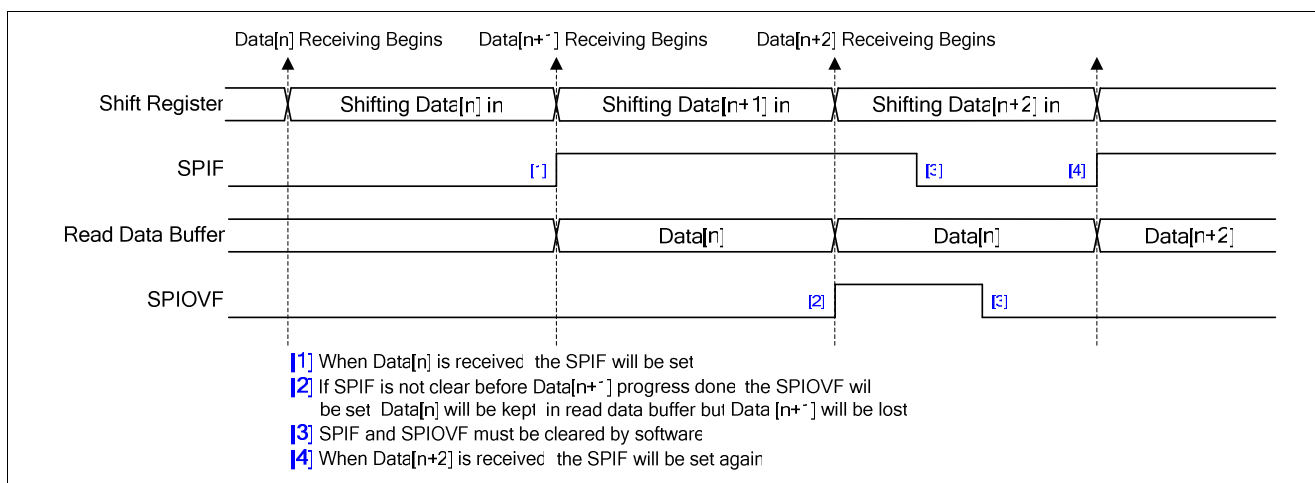


图 13-7. SPI 过载波形

13.10 SPI 中断

三个 SPI 状态标志位，SPIF，MODF，SPIOVF，可触发 SPI 中断请求。这三个标志位在寄存器SPSR中。数据传输完成后SPIF将置位。MODF 置位表示 \overline{SS} 有由模式错误状态引起的低电平。SPIOVF表示接收溢出错误。如果设置ESPI(EIE.6)和EA为1使能SPI中断屏蔽，三个标志位中任何一个置位，CPU将执行SPI中断服务程序。用户需要检测标志判断是什么事件引起中断的。这三个标志位可由软件清零。

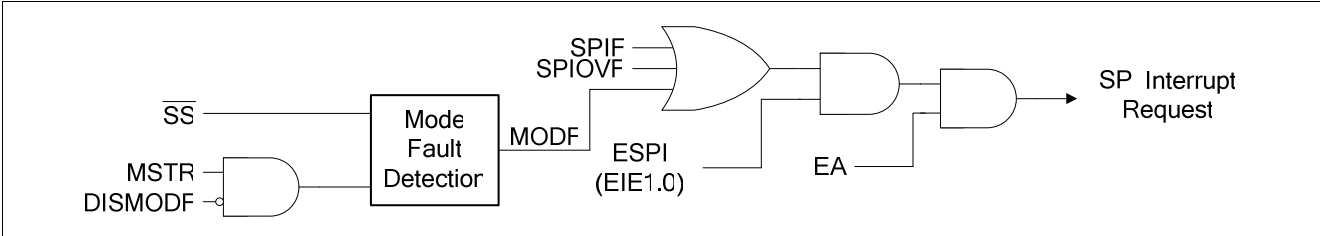


图 13-8. SPI中断请求

14 键盘中断(KBI)

N79E815系列有用于检测按键状态的8个键盘中断功能，当任何按键按下或键盘连接到**N79E815**系列的特定脚可以单独产生一个中断，如下图。该中断常用于把CPU从掉电模式或空闲模式唤醒。

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。判断哪一个按键按下KBI中断 程序里可以检测P0口

KBI 支持四种触发条件：低电平，下降沿，上升沿和边沿检测。每个端口引脚的触发条件独立由KBIS1(ECH).x 和 KBIS0(EBH).x 控制，其中x 为 0 到 7。

触发发生后，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idle状态下，系统先进入掉电模式以减小功耗并等待事件触发。**N79E815** 系列支持KBI中断，将MCU从掉电模式下唤醒。注如果KBI选作任何边沿触发模式，约定必须从掉电模式有效唤醒。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。

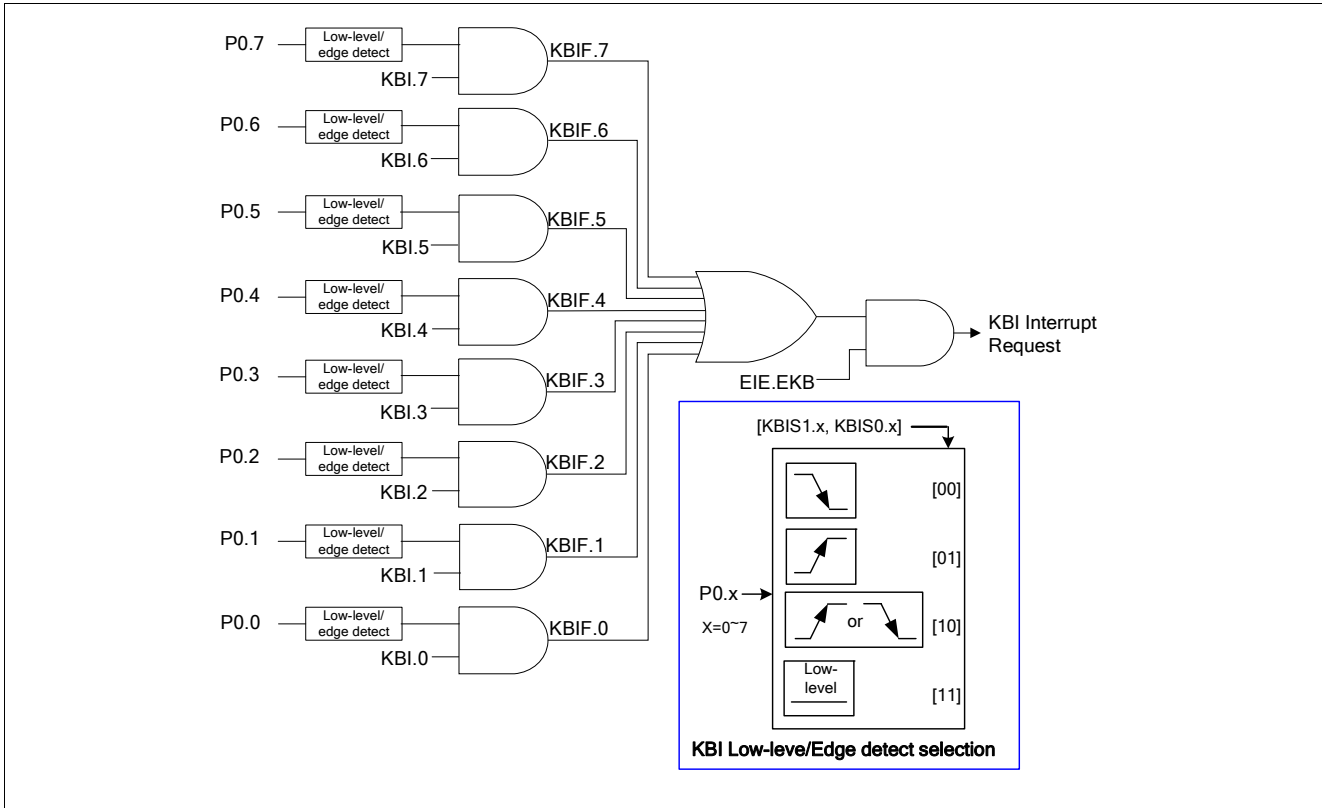


图 14-1 键盘中断检测

表 14-1. 配置不同的KBI电平选择

KBIS1.n	KBIS0.n	KBI Channel n Type
0	0	下降沿
0	1	上升沿
1	0	上升沿或下降沿
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

Address: E9H

reset value: 0000 0000B

位	名称	描述
7:0	KBIE	键盘中断 使能 P0[7:0] 为键盘中断。

KBIF – 键盘中断标志

7	6	5	4	3	2	1	0
KBIF7	KBIF6	KBIF5	KBIF4	KBIF3	KBIF2	KBIF1	KBIF0
r (level) r/w (edge)	r (level) r/w (edge)	r (level) r/w (edge)	r (level) r/w (edge)	r (level) r/w (edge)	r (level) r/w (edge)	r (level) r/w (edge)r	r (level) r/w (edge)

Address: EAH

reset value: 0000 0000B

位	名称	描述
7:0	KBIFn	键盘接口通道n标志。 选择KBI的边沿触发模式, 该标志由硬件置位 (KBI channel n (P0.n) detects a type defined edge) . 该标志由软件清零。 选择KBI的低电平触发模式, 该标志随着KBI通道n (P0.n)的输入信号的逻辑电平反转. 软件不能控制。

KBLS0 – 键盘电平选择^[1]

7	6	5	4	3	2	1	0
KBLS0.7	KBLS0.6	KBLS0.5	KBLS0.4	KBLS0.3	KBLS0.2	KBLS0.1	KBLS0.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: EBH

reset value: 0000 0000B

位	名称	描述
7:0	KBLS0[7:0]	键盘电平选择 0.

KBLS1 – 键盘电平选择1^[1]

7	6	5	4	3	2	1	0
KBLS1.7	KBLS1.6	KBLS1.5	KBLS1.4	KBLS1.3	KBLS1.2	KBLS1.1	KBLS1.0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Address: ECH

reset value: 0000 0000B

位	名称	描述
7:0	KBLS1[7:0]	键盘电平选择1.

[1] KBLS1 和 KBLS0 共同决定KBI每个通道的输入类型(on P0). See [Error! Reference source not found.](#)

15 模拟到数字转换(ADC)

N79E815 系列有一个ADC包含有DAC。DAC是把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 (Vin)和(VDAC)相比较并回馈到逐次比较控制器。把ADCCON0寄存器中的ADCS位设为‘1’，就开始了第一次转换。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实际执行tpw flip-flops: flip-flop命令受操作设置的影响, 状态标志在读操作时被访问.

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.

在以下8个机器周期里, 当前端口5选择引脚的电压被采样, 输入电压应该稳定以获得有效采样. 任何情况下, 输入电压的摆率必须低于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 (50% full scale)与 输入电压Vin相比较. 如果输入电压大于VDAC, 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(11 0000 0000b 或 01 0000 0000b, 根据当前结果), VDAC再次与Vin比较. 如果输入电压大于VDAC, 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. 每位转换花费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(8 upper bits in ADCH). 任何情况下, 实际转换时间为35个机器周期. 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idle或掉电模式时, 一个新的正在进行的ADC转换会中止. 当进入idle模式, 转换完成的结果保持不变 (ADCI = logic 1).

当 ADCCON0.5 (ADCEX)由外部引脚触发开始ADC转换, 在N79E815 系列进入idle模式, P1.4 在一个机器周期后可以开始ADC转换.

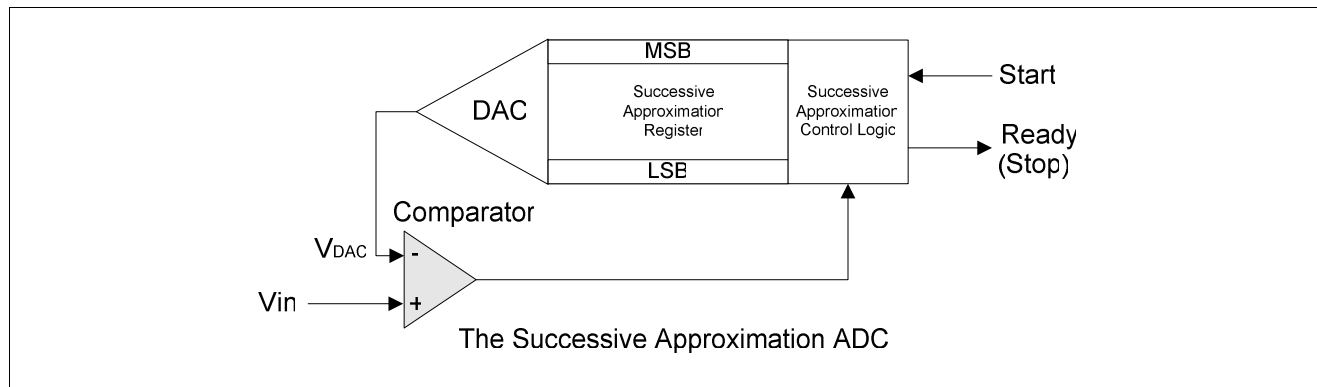


图 15-1 逐次逼近ADC

15.1 ADC 分辨率和模拟电压

ADC有自己的电源输入脚 (AVDD and AVSS)和一个参考电压输入脚(Vref+) 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Avss上的0.5XR，最后一个阶梯是Vref+下的0.5XR。总共有1024XR电阻阶梯。该结构确定DAC的单调变化并误差均匀。

输入电压在Avss和[(Vref+) + ½ LSB]之间，10-位ADC结果会是00 0000 0000 b = 000H；输入电压在 [(Vref+) – 3/2 LSB]和Vref+之间，10-位ADC结果会是11 1111 1111B = 3FFH。Avref+和AVSS可以在AVDD + 0.2V和AVSS – 0.2 V之间。Avref+应该是比AVSS的电位高，并且输入电压(Vin)应该在Avref+ 和AVSS之间。

下面是计算结果方程式：

$$\text{Result} = 1024 \times \frac{V_{in}}{AV_{ref+}} \quad \text{or} \quad \text{Result} = 1024 \times \frac{V_{in}}{VDD}$$

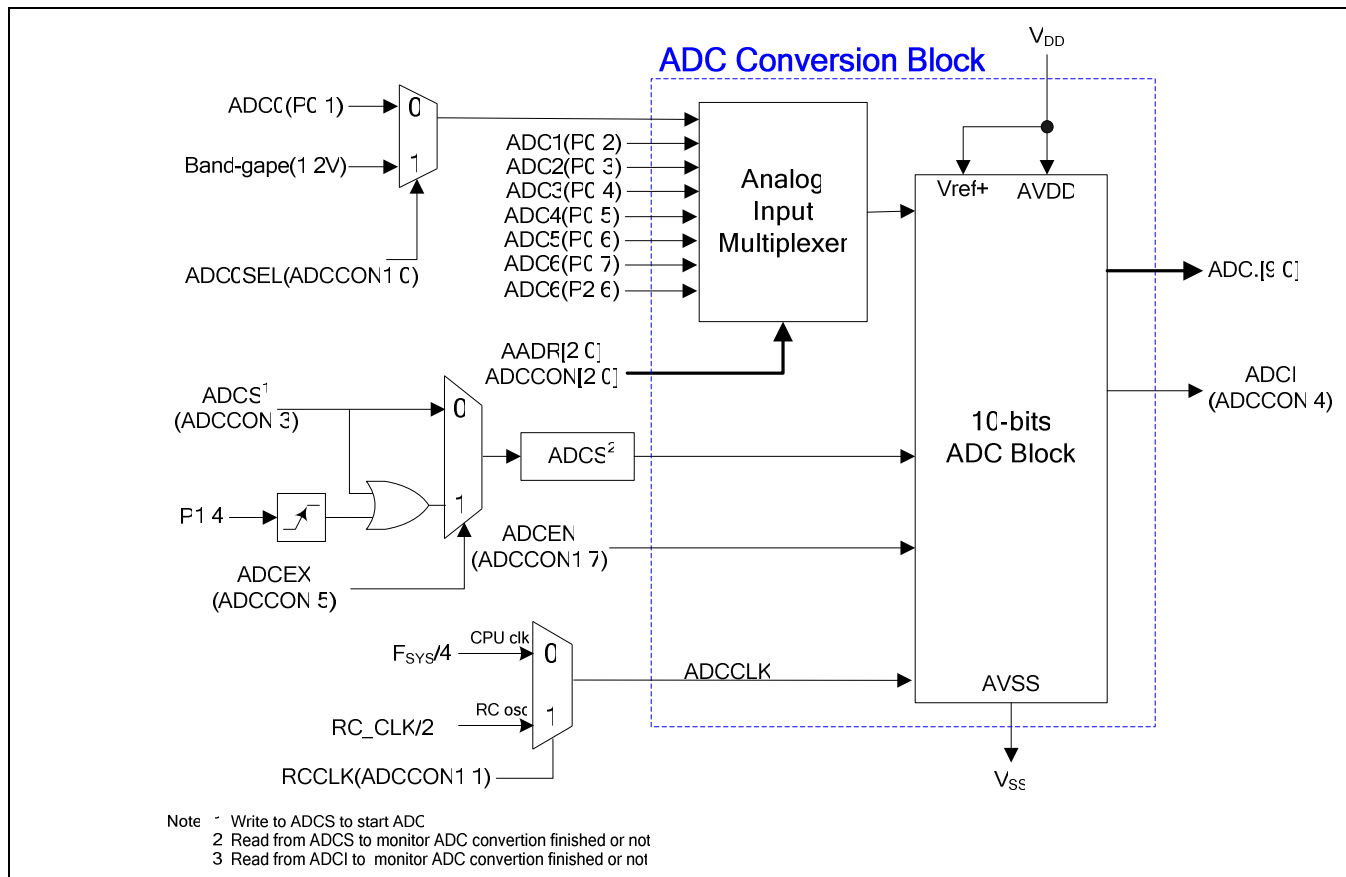


图 15-2 ADC 框图

ADCCON0 – ADC 控制寄存器

7	6	5	4	3	2	1	0
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

Address: F8H

reset value: XX00 0X00b

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, 这些位才可被 改变.

AADR2	AADR1	AADR0	选择模拟通道
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

Address: E2H

reset value: 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

Address: E1H

reset value: 0000 0000b

位	名称	描述
7	ADCEN	0: 禁止 ADC电路 1: 使能ADC电路
6:2	-	保留
1	RCCLK	0: F _{sys} /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 Band-gape (~1.2V)为输入.

16 I²C

16.1 特征

I²C 总线作为微控制器与I²C设备之间的串行接口, I²C设备如EEPROM, LCD 模组等.连接在2线制(SDA和SCL)I²C总线上的2个设备之间可以相互交换信息.

I²C 总线在主机和从机之间数据双向传输. 多总机总线(没有中心总机). 总线上出现两个主机同时传输数据的无冲突仲裁. I²C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. I²C 接口仅支持7位寻址模式和General Call. I²C 支持标准速率(up to 100kbps) 和快速 (up to 400kbps).

16.2 功能描述

对于双向传输操作, SDA和SCL必须开漏模式. 执行wired-AND 功能是该接口最基本的操作. I²C器件输出“0”时, I²C总线上为低电平. 当所有I²C器件输出“1”时, 产生高电平,允许上拉电阻将总线拉高.

N79E815 系列, 用户在使能I2CEN (I2CON.6)使能I²C功能之前, 应该设置P1.2和P1.3输出为逻辑“1”. 一理使能I²C功能, P1.2和P1.3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.

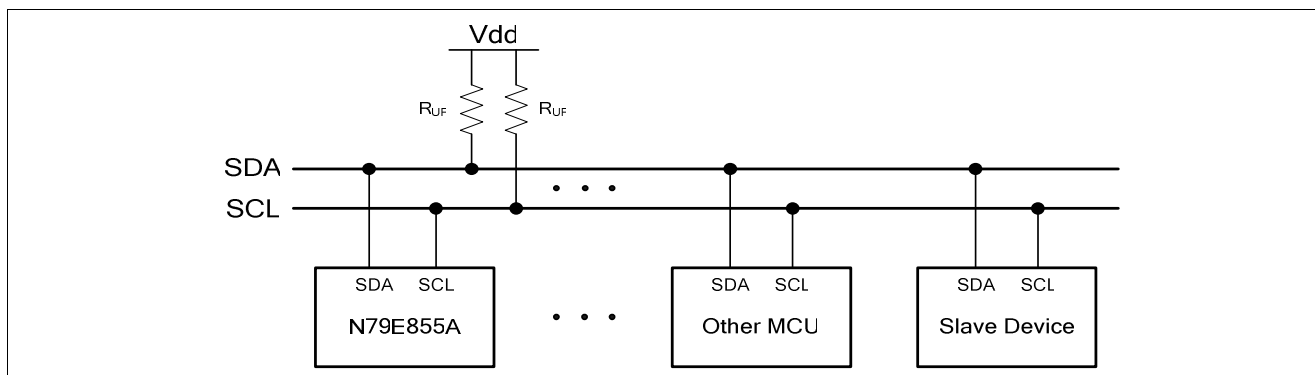


图 16-1. I²C 总线连接

两线都为高时, I²C空闲. 同时, 任何器件可作为主机占用总线并在产生START条件后传输. 在发送STOP条件结束传输之前, 总线视为忙. 主机发生所有串行时钟脉冲和START和STOP条件. 然而如果总线上没有START 条件, 所有器件作为没被寻址的从机. 硬件寻找自己的从机地址或General Call 地址. (The General Call address detection may be enabled or disabled by GC (I2ADDR.0).)如果接收到的地址匹配时,请求中断.

I²C总线上每个传输为9位长度, 由8位数据和一个应答位组成. 每次传输的字节数 (defined as the time between a valid START and STOP condition) 不受限制, 但每个字节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. 在第8个SCL总线上的下降沿, 器件将SDA改变输出为输入, 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. 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.

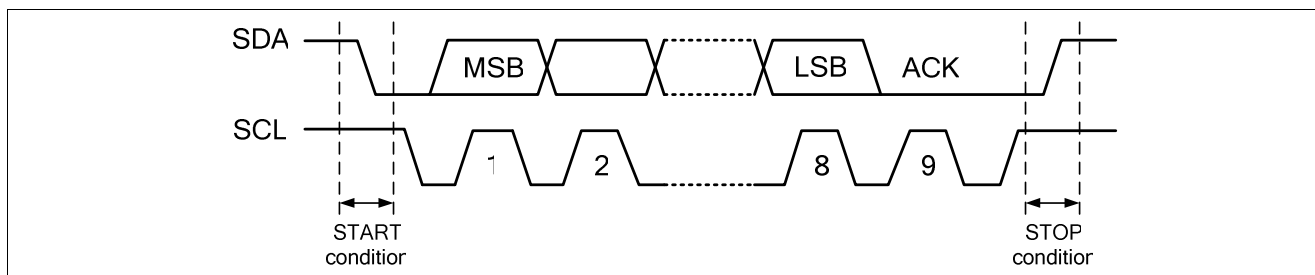


图 16-2. I²C 总线协议

16.2.1 START 和 STOP 条件

I²C总线协议定义两个状态开始和结束传输, START (S) 和 STOP (P) 条件. START条件: 当SCL为高时, 在SDA总线上有从高到低的变化. STOP条件: 当SCL为高时, 在SDA上有从低到高的变化. START 或 STOP 条件常由主机产生, 在START条件之后I²C总线视为忙, 在STOP条件之后, I²C总线视为空闲. STOP条件出现后, 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无寻址的从机. I²C 总线空闲并等待下一个START条件.

通常由主机产生STOP条件中止数据传输. 然而, 如果主机仍希望在总线上通信, 就会重复产生START (Sr) 条件和地址或另一个从机没有先产生STOP条件. 读/写格式的各种结合在传输中皆有可能.

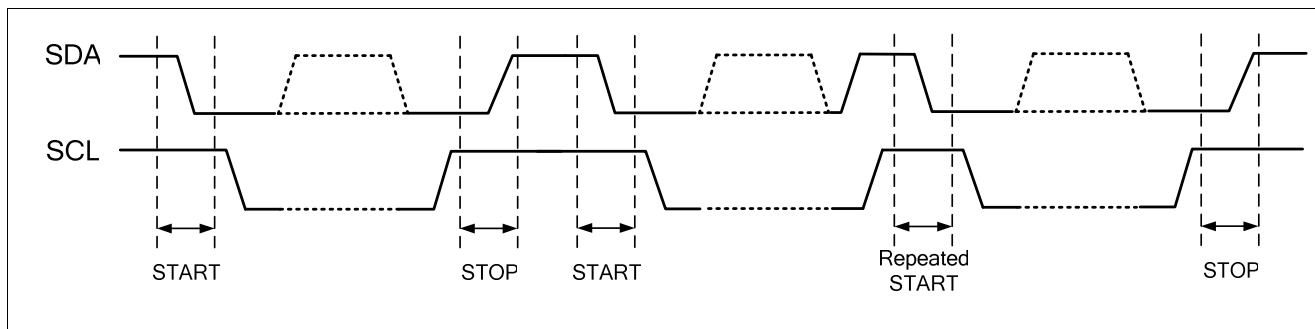


图 16-3. START, 重复 START, 和 STOP 条件

16.2.2 7-位 地址的数据格式

START条件产生之后, 一个字节的特殊数据应该由主机传输. 包括在第8位数据方向位 (R/W)之后7位长度的从机地址(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read (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START 条件, 一个 SLA+R/W, 一个或多个数据包和一个STOP条件组成. 在SLA+R/W 指定从机地址后, 第二个和之后8位数据字节由主机或从机地址根据R/W位配置.

“General Call” 地址可以寻址所有器件, 通过数据的第一个字节全为0. General Call用于当主机希望发送相同信息到几个从机时. 当地址使用时, 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General Call, 其操作就像从机接收器模式.

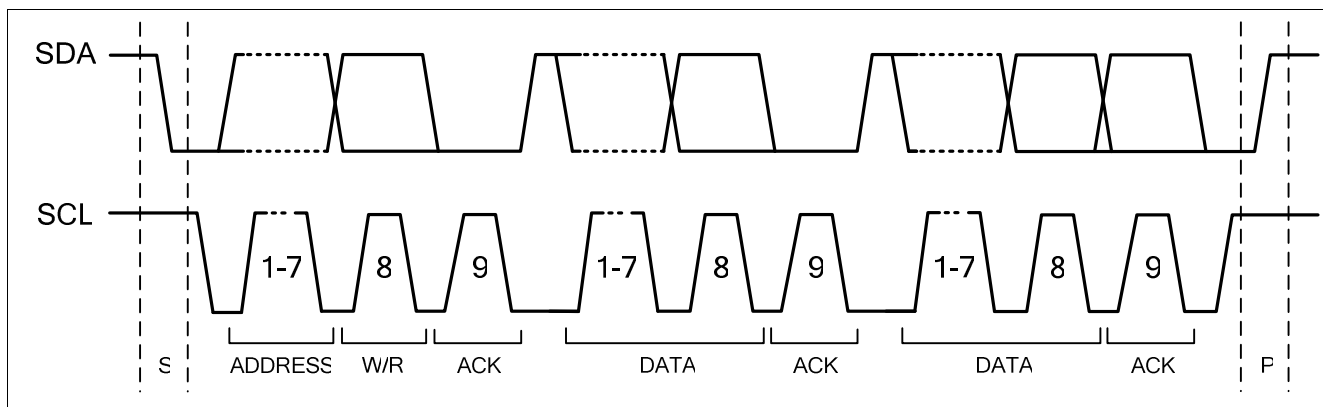


图 16-4. I2C传输的数据格式

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9th 个SCL脉冲视作应答信号(ACK). 允许接收器件(which can be the master or slave)对发送器件的响应(which also can be the master or slave),将SDA拉低. 应答相关时钟脉冲由主机产生. 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, 表示器件已经接收到数据. 通常, 接收器在接收到一个字节之后产生一个ACK. 当从机接收器没有应答 (NACK)从机地址, SDA线必须由从机拉高以让主机产生STOP或重复START条件.

如果从机接收器不应答从机地址, 将切换到不寻址从机模式, 不再接收任何数据. 从机保持SDA总线为高. 主机应该产生STOP或复位START条件.

如果主机接收器传输时，由于主机在传输时控制字节数据，就必须给从机发送器标记数据的结束，这种标记就是在最后一个字节不产生应答。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STOP或复位START条件。

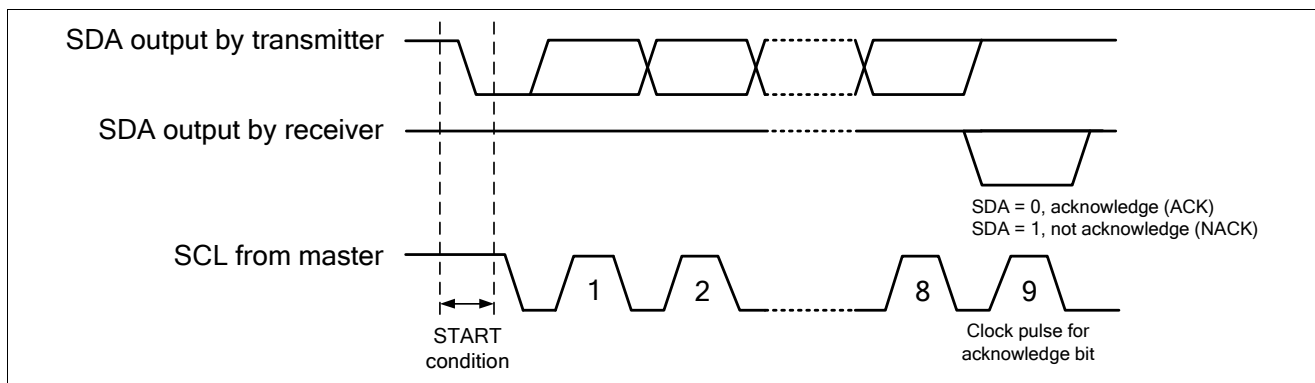


图 16-5. 应答位

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START条件。在这种情况下，当SCL为高时，就要在SDA总线上有仲裁。

输出数据后，所有主机进行仲裁以侦测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。如果几个主机试图寻址相同从机时，仲裁继续输出数据包。

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。

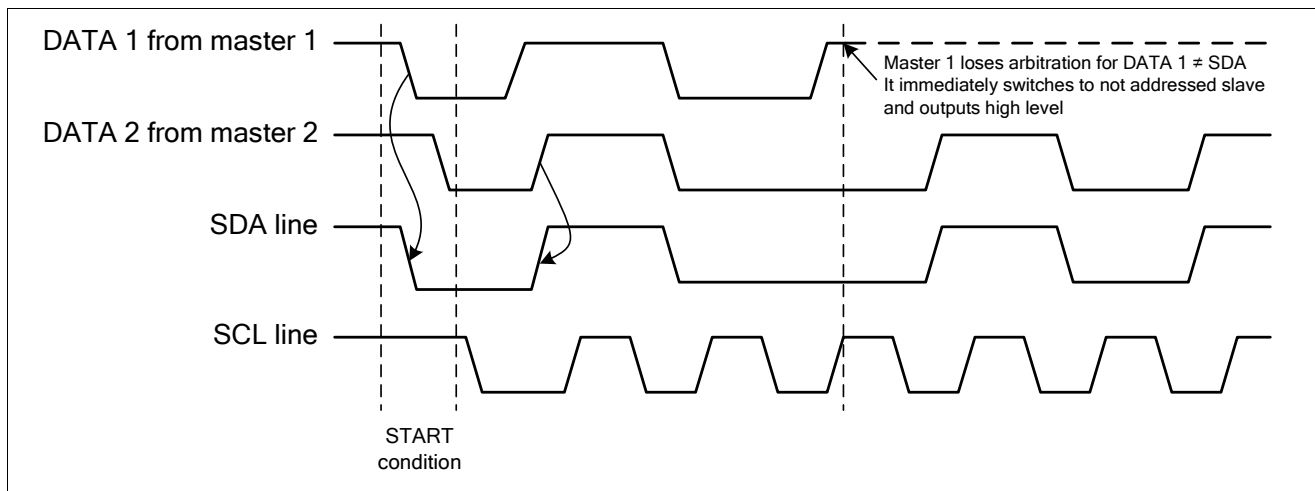


图 16-6. 仲裁两个主机的过程

I2C总线的控制由地址或主机码和主机完成发生数据决定，没有中心主机，总线也没有优先级。

从机没有仲裁过程。

16.3 I²C控制寄存器

I2C总线有5个控制寄存器。它们是I2CON, I2STAT, 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I2C功能相关寄存器描述。

I2CON – I²C 控制

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

Address: C0H

reset value: 0000 0000b

位	名称	描述
7	-	保留。
6	I2CEN	I²C 总线使能. 0 = 禁止 I ² C总线。 1 = 使能 I ² C总线。 使能 I ² C之前，Px.x 和 Px.x口必须设置为逻辑“1”。一旦使能I2C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。
5	STA	START 标志. 当 STA 置位，如果总线空闲，I ² C产生START条件。如果总线忙，I2C等待STOP条件，并产生一个START条件。 主机模式下，当I2C准备好和发送或接收一个或多个字节，STA置位，I ² C产生一个重复START条件。 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START条件后由硬件自动清零。用户应该手动清STA。
4	STO	STOP 标志. I2C为主机模式时，STO位置‘1’，将在I2C总线上输出STOP条件。当检测到总线上出现STOP条件。I2C硬件清除STO标志，在从机模式，STO标志被置‘1’恢复总线错误条件。在改模式下没有STOP条件传输到I2C总线上。然而I2C 硬件动作好像有STOP条件已经被接收并切换到不可寻址的从接收模式。STO标志由硬件自动清除。如果I2C在主机模式 (在从机模式，I2C产生一个内部的STOP条件不传输到总线上)，如果STA和STO位同时被置‘1’，STOP条件被传输到I2C总线。随后I2C传输开始条件。

位	名称	描述
3	SI	<p>串行中断标志.</p> <p>当I2C进入25种可能状态之一(besides F8H status)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总线处长低周期. 传输暂停. 这对从机器件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, I²C 总线继续产生START 或重复START 条件, STOP 条件, 8-位数据, 或依据软件配置控制字节或位.</p>
2	AA	<p>应答标志.</p> <p>如果置位AA 标志, ACK (low level on SDA) 将在SCL总线的应答时钟脉冲期间返回 (当I2C器件为接收器).</p> <p>如果AA 标志被清零, NACK (high level on SDA)将在SCL总线的应答时钟脉冲期间返回 (当I²C器件为接收器) .</p>
1:0	-	保留.

I2STAT – I²C 状态

7	6	5	4	3	2	1	0
I2STAT[4:0]						0	0
r						r	r

Address: BDH

reset value: 1111 1000b

位	名称	描述
7:3	I2STAT[4:0]	<p>I²C 状态码.</p> <p>I2STAT的高五位包含状态码. 总共有26个可能的状态码. 当 I2STAT 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I2C状态, 当进入这些状态时, SI将被置1且请求中断.</p>
2:0	-	<p>保留.</p> <p>I2STAT 的最低三位常为0.</p>

I2DAT – I²C 数据

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

Address: BCH

reset value: 0000 0000b

位	名称	描述
7:0	I2DAT[4:0]	<p>I²C 数据.</p> <p>I2DAT 包含一个字节 (发送的I2C数据字节或刚接收到的一个字节) .只要SI为逻辑1, I2DAT的数据保持.</p>

I2ADDR – I²C 自身从机地址

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

Address: C1H

reset value: 0000 0000b

位	名称	描述
7:1	I2ADDR[6:0]	I ² C 器件的自身从机地址。 <u>主机模式:</u> 该位无效。 <u>从机模式:</u> 该7位定义了I2C器件的从机地址..
6	GC	General Call 位. <u>主机模式:</u> 该位无效。 <u>从机模式:</u> 0 = 忽略General Call. 1 = 如果AA标志为1, 视为General Call; otherwise, it is ignored if AA is 0.

I2CLK – I²C 时钟

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

Address: BEH

reset value: 0000 1110b

位	名称	描述
7:0	I2CLK[7:0]	I ² C 时钟设置。 <u>主机模式:</u> 当器件为主机模式时, 该寄存器决定了I2C总线的时钟速率. 时钟速率依据下面公式. $F_{I^2C} = \frac{F_{PHER1}}{1 + I2CLK}$ 如果系统时钟为24MHZ带DIVM 1/4模式, I2C总线时钟速率为400kbps. 注I2CLK的值为00H 和 01H 无效。 <u>从机模式:</u> 该字节无效. 在从机模式下, I ² C器件将自动同步所给的时钟速率达到400kps.

16.4 模式操作

I²C协议定义, 有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General Call. 工作于简单的主机传输模式.

16.4.1 主机发送模式

主机发送模式，几个数据字节发送到从机接收器。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START条件。成功产生START条件后，SI标志 (I2CON.3)将置位且I2STAT的状态码为08H。给I2DAT载入目标从机地址和数据方向位“写”(SLA+W)，该过程继续。SLA+W开始传输时SI位必须清零。

在SLA+W字节被发送后，由被寻址的从机器件返回应答(ACK)，SI标志再次置位且I2STAT读出为18H。依据用户定义的通信协议采取相应的动作。数据发送后，主机可以通过设置STO (I2CON.4)发送STOP条件且清SI中止传输。在没有发送STOP条件时重复START条件立即开始另一次传输。

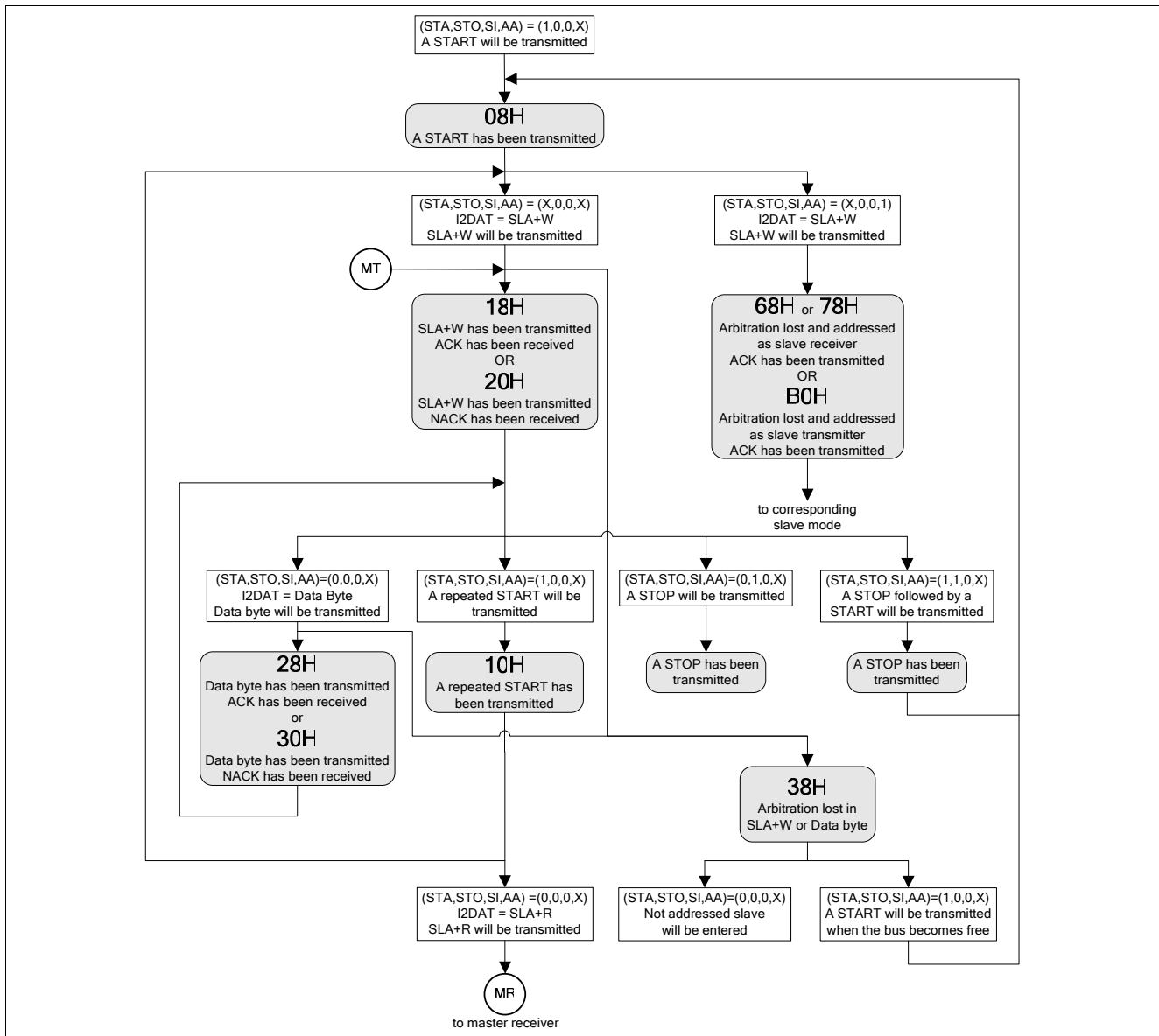


图 16-7. 主机发送模式流程和状态

16.4.2 主机接收器

主机接收模式，从从机发送器接收几个字节的数据。START条件之后，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I2STAT读作40H。SI标志应该被清零以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从机。之后，主机产生STOP条件或重复START条件中止传输或开始另一个。

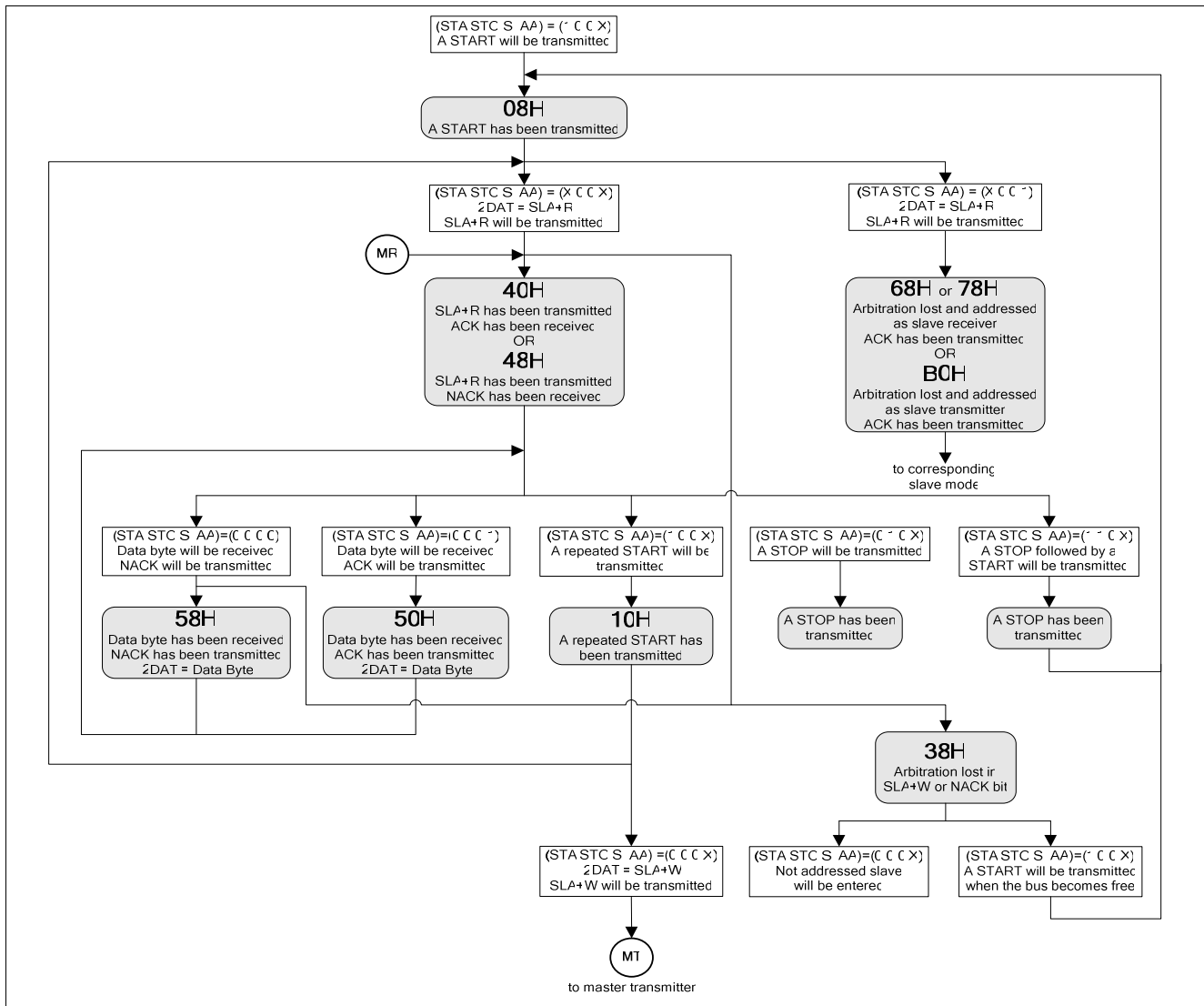


图 16-8. 主机接收模式流程和状态

16.4.3 从机接收模式

从机接收模式, 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位知心朋友设置使能应答自己从机地址或General Call.

从机被SLA+W寻址, 应该清SI标志以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.

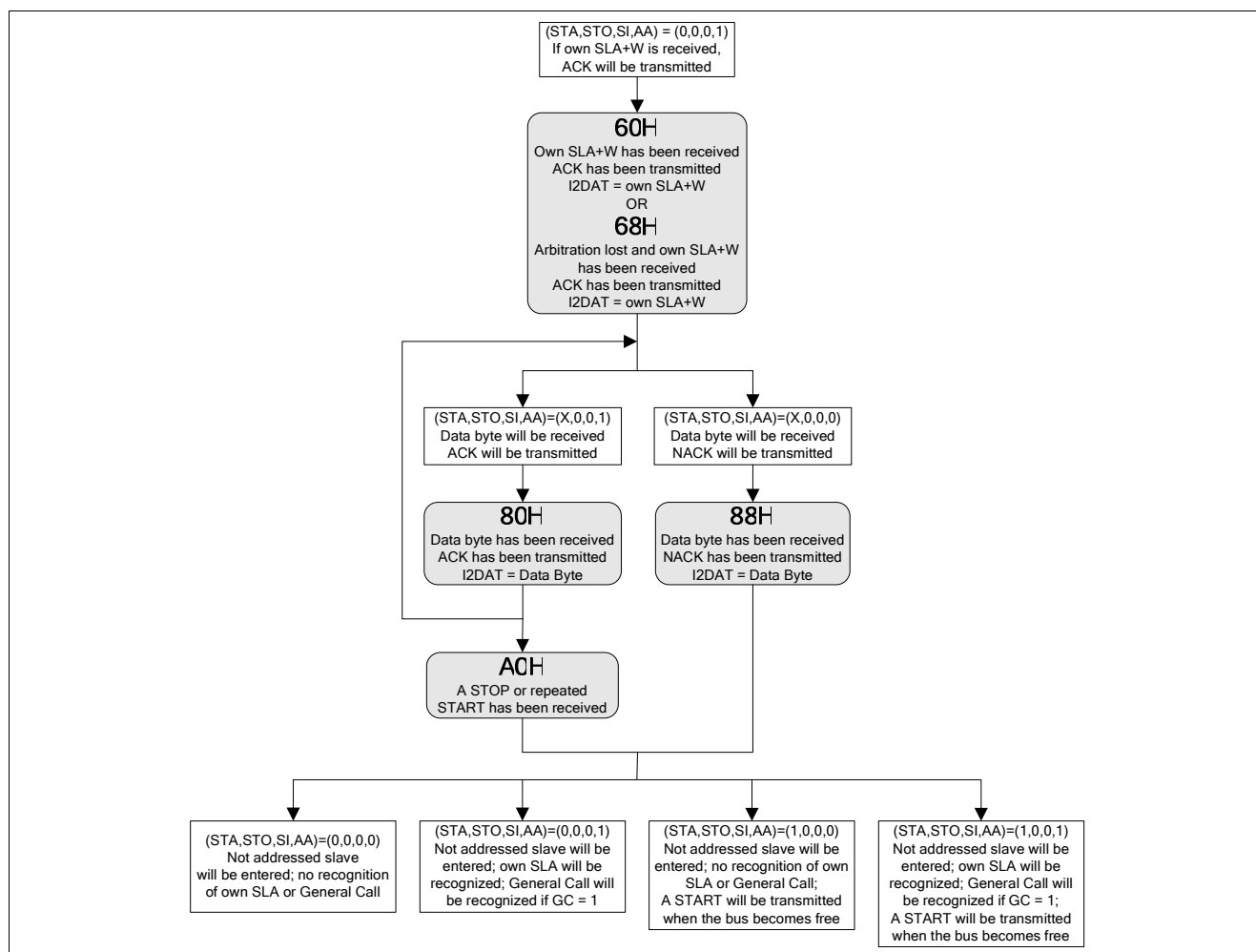


图 16-9. 从机接收器流程和状态

16.4.4 从机发送模式

从机发送模式，发送几个字节数据到主机接收器。确定 I2ADDR 和 I2CON 的值之后，I²C 等待自己的地址被寻址“读” (SLA+R)。

在从机被 SLA+W 寻址后，应该清 SI 标志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。

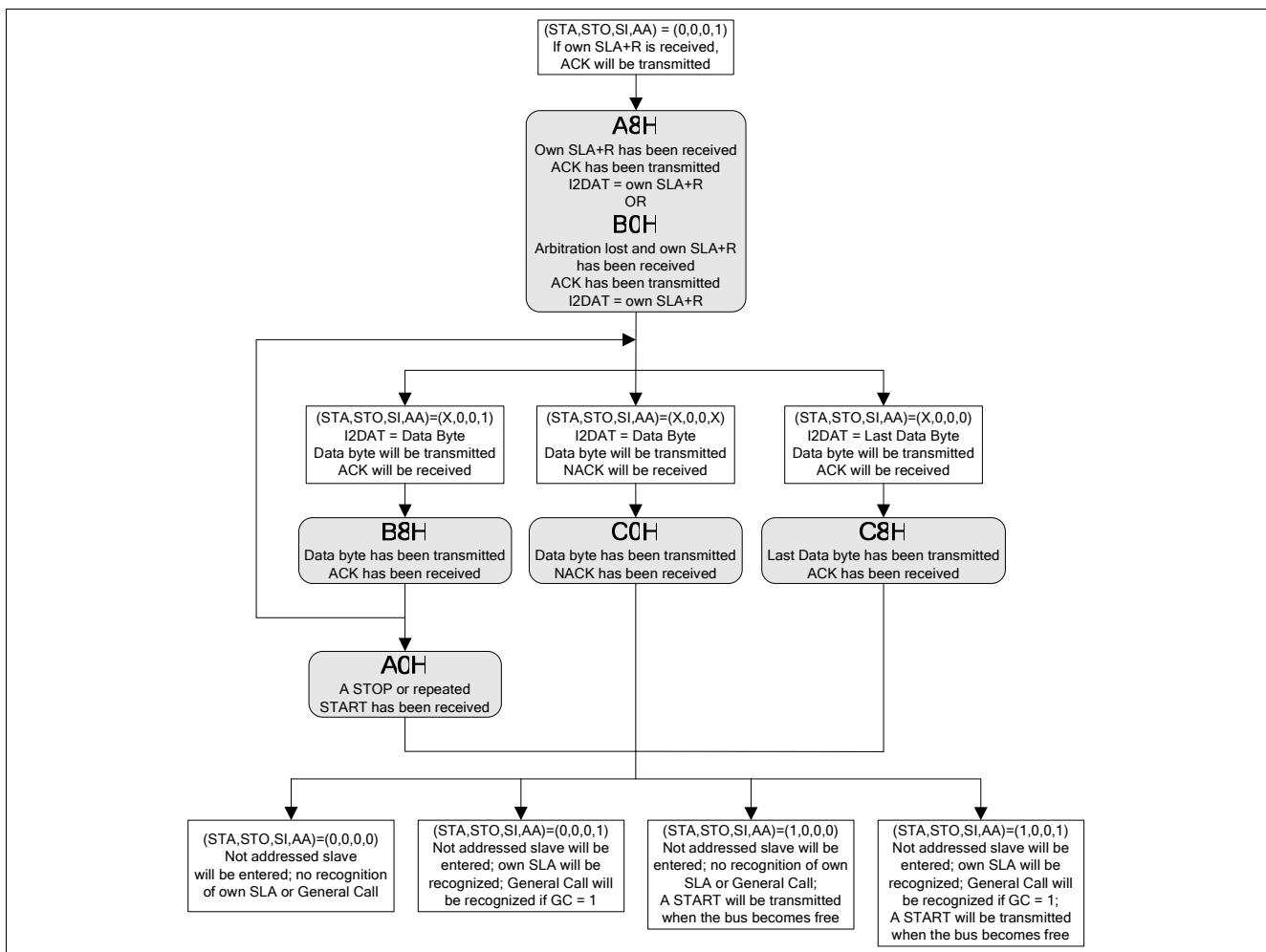


图 16-10. 从机传输模式流程和状态

16.4.5 General Call

General Call是从机接收模式的一种特殊条件(从机地址和数据方向位全为0). 从机被General Call 寻址I2STAT有不同状态码.

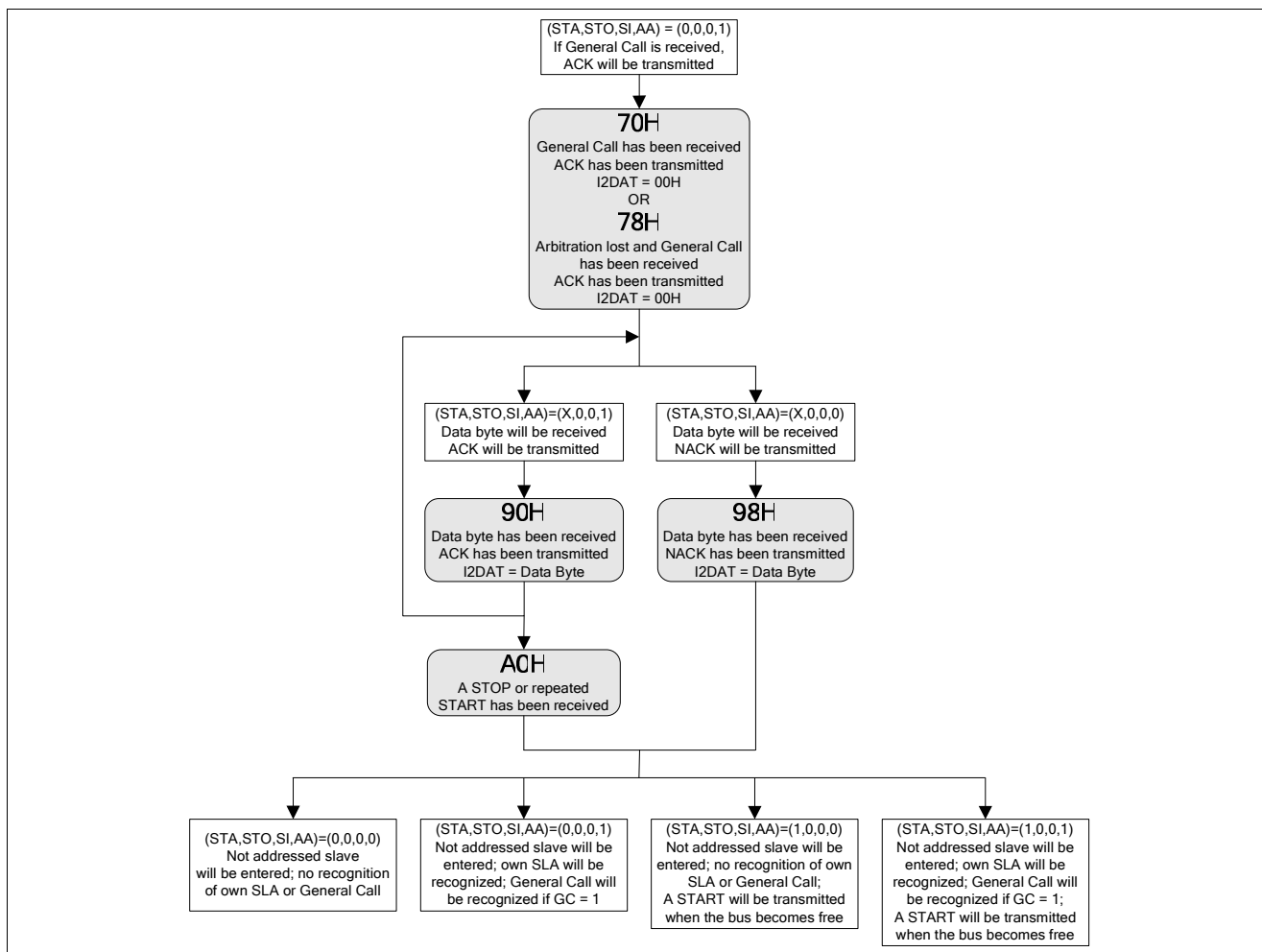


图 16-11. 模式流程和状态

16.4.6 其他状态

有两个I2STAT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I2C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STOP条件暂时出现在一个非法的位置, 如通过在第8位的第2个字节的地址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I2C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，载入I2STAT 00H。从总线错误恢复，STO位必须设置为逻辑1且SI必须清零。此后，STO 由硬件清零且在没有STOP条件就释放I2C总线。

特例：如果START或重复 START条件不成功，I2C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，I²C 硬件发送额外时钟脉冲，但是没有START条件产生，因为SDA被拉低。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START条件，I²C 硬件也执行以上相同的动作。此情况下，进入状态08H，而不是在成功发送START条件后进入10H。注软件不能解决这类总线问题。

16.5 典型I²C结构的中断服务程序

以下例子用户可以根据自己的应用进行修改..

```
void I2C_ISR (void) interrupt 6
{
    switch (I2STAT)
    {
        //=====
        //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,
                                                ACK 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 (I2STAT)

SI = 0;
while(STO);
} //end of I2C_ISR

```

//SI should be the last step of I2C ISR
//wait for STOP transmitted or bus error
//free, STO is cleared by hardware

16.6 I²C 超时

有一个14-位的超时计数器，可用于处理I²C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I²C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I²C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。

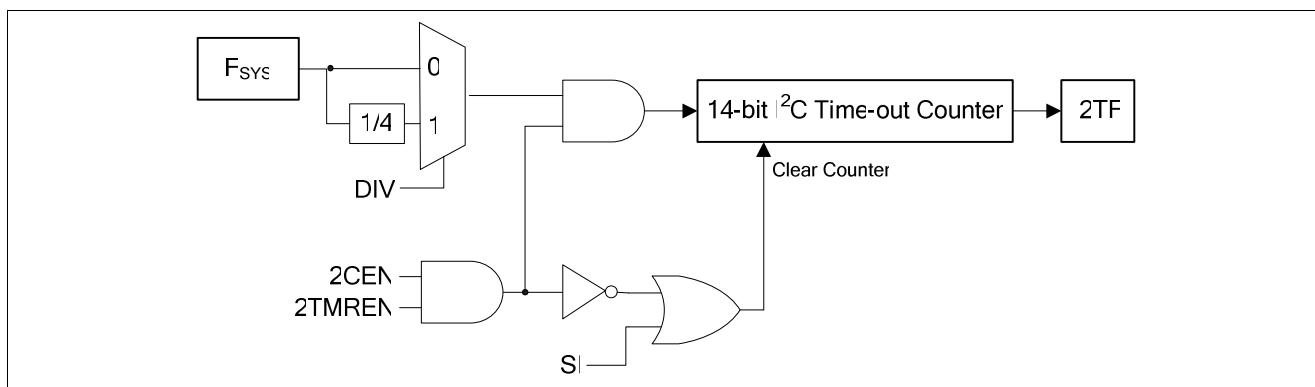


图 16-12. I²C 超时计数

I2TOC – I²C 超时计数器

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

Address: BFH

reset value: 0000 0000b

位	名称	描述
7:3	-	保留

位	名称	描述
2	I2TOCEN	使能I ² C超时计数器. 0 = 禁止I ² C超时计数器. 1 = 使能I ² C超时计数器.
1	DIV	I ² C 超时计数器时钟分频. 0 = I ² C 超时计数器分频为1/1 of F _{sys} . 1 = I ² C超时计数器分频为1/4 of F _{sys} .
0	I2TOF	I ² C 超时计数溢出标志. 如果14位I2C超时计数器溢出, I2TOF标志由硬件置位. I2TOF标志由软件清零.

16.7 I²C 中断

有两个 I²C 标志, SI 和 I2TOF. 均可以产生I2C中断请求. 如果使能 I²C 中断mask (设置EI2C (EIE.0)) 和EA为1, 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是哪个中断事件. 两个 I²C 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. **N79E815** 系列提供四个通道, 最大10位的PWM输出.

17.2 功能描述

N79E815 系列包括4个PWM (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, 在引脚作为内部PWM输出时, 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:

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

PWMP包含PWMPH和PWMP, 如下所述.

PWMP – PWM 计数器低位寄存器

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

Address: D9H

reset value: 0000 0000b

位	名称	描述
7:0	PWMP	PWM 计数器位寄存器bit[7:0].

PWMPH – PWM计数器高位寄存器

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

Address: D1H

reset value: 0000 0000b

位	名称	描述
7:2	-	保留.

位	名称	描述
1:0	PWMPH	PWM 计数器位寄存器 bit[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. 然后设置周期和占空比寄存器, PWMRUN (PWMCON0.7) 可设置为1触发10位向下计数运行. PWM输出开始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PWM信号输出, 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

Address: DAH

reset value: 0000 0000b

位	名称	描述
7:0	PWM0L	PWM 0 低位寄存器bit[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

Address: D2H

reset value: 0000 0000b

位	名称	描述
7:2	-	保留.
1:0	PWM0H	PWM 0 高位寄存器bit[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

Address: DBH

reset value: 0000 0000b

位	名称	描述
7:0	PWM1L	PWM 0低位寄存器bit[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

Address: D3H

reset value: 0000 0000b

位	名称	描述
7:2	-	保留.
1:0	PWM1H	PWM 1高位寄存器bit[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

Address: DDH

reset value: 0000 0000b

位	名称	描述
7:0	PWM2L	PWM 2低位寄存器bit[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

Address: D5H

reset value: 0000 0000b

位	名称	描述
7:2	-	保留.
1:0	PWM2H	PWM 2 高位寄存器 bit[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

Address: DEH

reset value: 0000 0000b

位	名称	描述
7:0	PWM3L	PWM 0 低位寄存器bit[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
---	---	---	---	---	---	-----	-----

Address: D6H

reset value: 0000 0000b

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

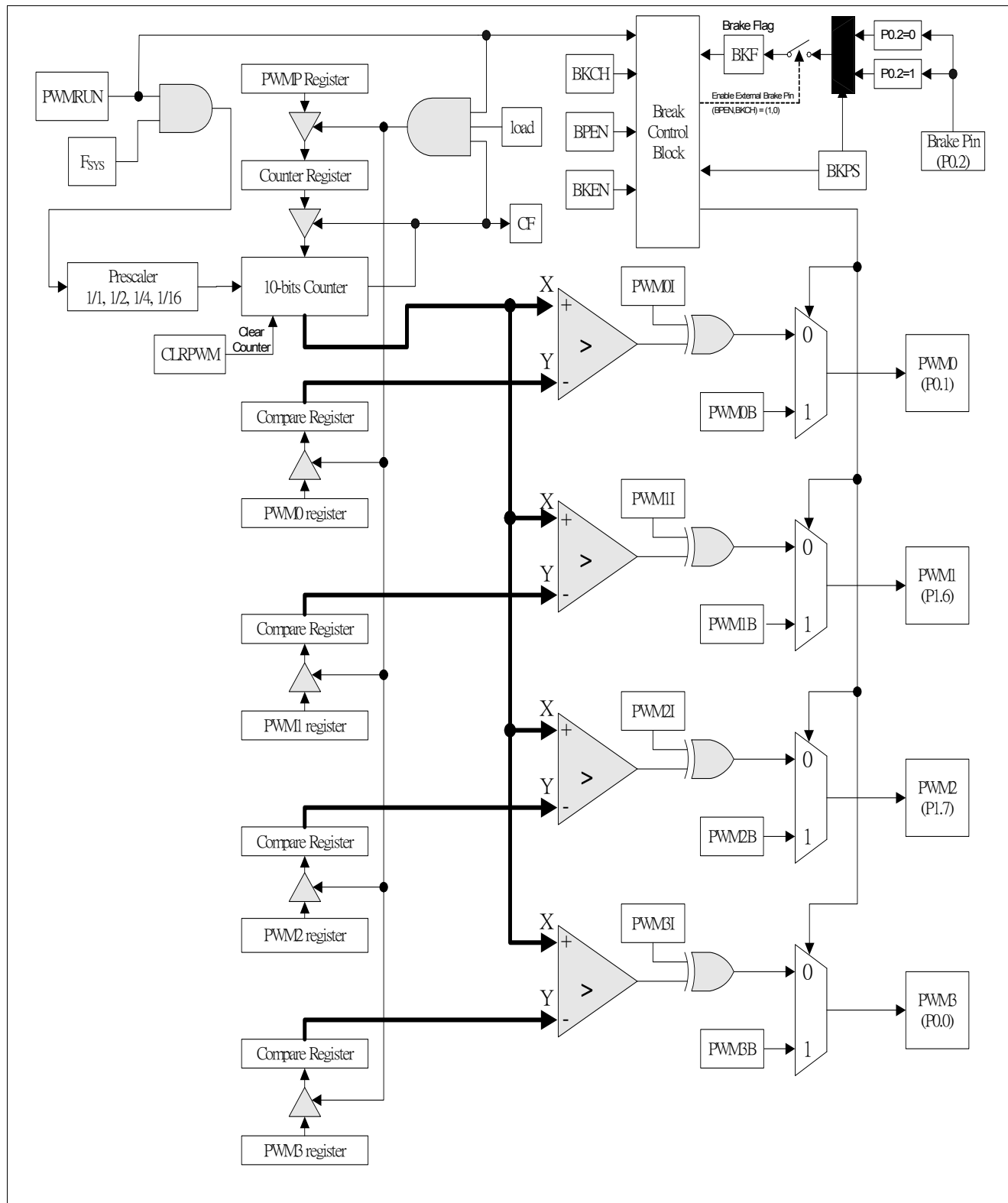


图 17-1 PWM 框图

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 etc.

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, 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.

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

Address: DCH

reset value: 0000 0000b

位	名称	描述
7	PWMRUN	0: PWM 不运行. 1: PWM 计数器运行.
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-位计数器没有向下溢出. 1: 10-位计数器向下溢出
4	CLRPWM	1: 清 10-位PWM计数器为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.

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

Address: DFH

reset value: 0000 0000b

位	名称	描述
7	BKCH	见下表, BKEN置位.
6	BKPS	0: 如果P0.2为低, Brake. 1: 如果P0.2为高, Brake
5	BPEN	见下表, BKEN 置位.
4	BKEN	0: 没有Brake. 1: 使能Brake, 见下表.
3	PWM3B	0: PWM3输出为低, Brake. 1: PWM3输出为高, Brake.
2	PWM2B	0: PWM2输出为低, Brake. 1: PWM2输出为高, Brake.
1	PWM1B	0: PWM1输出为低, Brake. 1: PWM1输出为高, Brake.
0	PWM0B	0: PWM0输出为低, Brake. 1: PWM0输出为高, Brake.

Brake 条件列表

BPEN	BKCH	BREAK 条件
0	0	打开 Brake, (软件 brake 和保持 brake)
0	1	打开, 当 PWM 不运行时 (PWMRUN=0), PWM 的输出条件依据 PWMNB 的设置. 关闭, 当 PWM 运行时(PWMRUN=1).
1	0	打开 Brake, 当声明为 break 引脚, 没有 PWM 输出, PWMRUN 将被清零, 且 BKF 标志置位.
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
Address: D7H				reset value: 0000 0000b			

位	名称	描述										
7:4	-	保留.										
3:2	FP[1:0]	选择 PWM 频率预分频. 预分频的时钟源, 如果PWMRUN=1, Fpwm 的相位F _{sys} .										
		<table border="1"> <thead> <tr> <th>FP[1:0]</th> <th>Fpwm</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>F_{sys} (Default)</td> </tr> <tr> <td>01</td> <td>F_{sys} /2</td> </tr> <tr> <td>10</td> <td>F_{sys} /4</td> </tr> <tr> <td>11</td> <td>F_{sys} /16</td> </tr> </tbody> </table>	FP[1:0]	Fpwm	00	F _{sys} (Default)	01	F _{sys} /2	10	F _{sys} /4	11	F _{sys} /16
FP[1:0]	Fpwm											
00	F _{sys} (Default)											
01	F _{sys} /2											
10	F _{sys} /4											
11	F _{sys} /16											
1	-	保留.										
0	BKF	外部 brake引脚标志. 0: PWM 没有brake. 1: PWM由外部brake引脚brake. 将由软件清零.										

Brake功能, 由寄存器PWMCON1的内容控制. 通常当Brake发生时, 4个PWM的输出钳位在用户选择的状态, 即状态由PWMCON1 bits 0 到 3位选择. 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brake. 如果都为“0”, 声明Brake. 如果 PWMCON1.7 为“1”当PWMCON0.7为“0”声明 brake. 如果PWMCON1.6 为“1”当Brake引脚, P0.2与PWMCON1.6有相同极性, 声明brake. 当声明brake响应引脚, 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”.

Brake 引脚被声明将自动清运行位PWMCON0.7且标志 BKF(PWMCON2.0) 置位, 用户程序查询该位或使能 PWM's brake 中断以决定何时Brake引脚发生brake. 另一个检查brake 的方法, 将Brake引脚接到外部中断引脚上. 后一种方法需要Brake信号足够长以确保被查询程序捕获.

详细的 PWMCON1 见下表.

18 时控访问保护(TA)

N79E815 系列有许多新的功能，如看门狗定时器，片上ROM 大小调整，等待状态控制信号，上电/掉电复位标志，这些对系统的正常运行来说非常的重要。如果不加以保护，无关代码可能会改写看门狗定时器的相应位，而使系统工作不正常或失控。为了保护这些位，**N79E815**提供了一种保护机制，来控制对SFRs 这些位的写操作。这种保护是通过时控访问来实现。

TA –时控访问

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

Address: C7H reset value: 0000 0000b

位	名称	描述
7:0	TA[7:0]	<p>时控访问。</p> <p>用于控制对保护位的访问。要访问被保护的位，用户首先要向TA寄存器写入AAH，然后立即再写入55H，之后系统将提供3个机器周期的时间以供用户访问被保护的位。该寄存器用于保护一些会影响系统正常运行的关键寄存器，防止代码误写这些寄存器。</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
```

时控访问例子用来解释正确或错误的写过程。

例 1,

```
(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

```

例 2,

```

  (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

```

例 3,

```

  (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

```

例 4,

```

  (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

```

第一个例子，向受保护的写动作在三个机器周期之前进行。例 2，对ISPTRG的写动作没有在窗口打开时完成，因此ISPTRG的值没有改变。例 3，WDCON0成功被写入数据，但是对PMCR的访问超过了三个机器周期。因此PMCR的值没有改变。例 4，向TA写55H 在第一次向TA写AAH三个机器周期之后，因此时控访问窗口没有打开，向保护位的写操作失败。

N79E815 系列，TA 保护SFRs包括PMCR, CHPCON (9FH), ISPTRG (A4H), SHBDA (9CH), WDCON0 (D8H), 和 WDCON1 (ABH)。

19 中断系统 (Interrupt)

N79E815 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禁止。

19.1 中断源

外部中断 $\overline{INT0}$ 和 $\overline{INT1}$ 可边沿触发或电平触发，取决于 $IT0$ 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IEx置位。该标志位请求中断。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 IEx 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IEx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EIE14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。

I2C 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，则共享向量地址。

ADC转换完成后会产生ADC中断。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 (Brake pin) 检测到高(BKPS=1)或低(BKPS=0)产生PWM brake 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，LVF，如果电源电压降到BOD电压以下被声明，如果使能EBO (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。

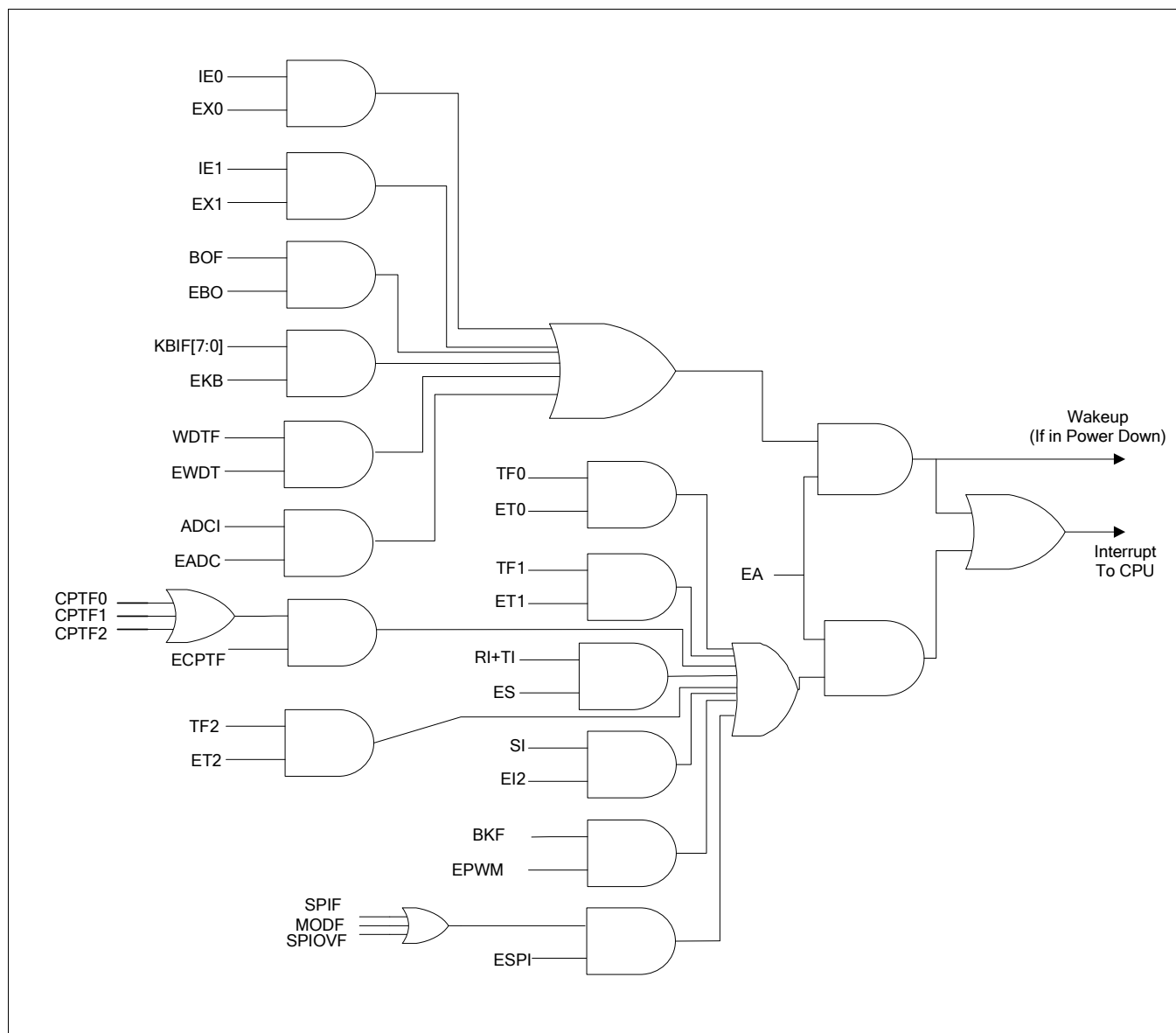


图 19-1 中断标志框图

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级。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

源	向量地址	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
I2C 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 brake 中断	0073h		

The vector table is not evenly spaced; this is to accommodate future expansions to the device family.

表 19-2 四级中断优先级

优先位		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RET指令。在执行RET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RET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。

N79E815 系列使用4级优先级的中断结构。**N79E815** 系列支持14个中断源。

每个中断源可以通过设置或清除IE or 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 中断源列表

描述	中断标志位(s)	向量地址	中断使能位(s)	清标志方式	中断优先级	仲裁级	掉电唤醒
外部中断 0	IE0	0003H	EX0 (IE0.0)	硬件软件	IPH.0, IP.0	1(highest)	Yes
BOD 检测	LVF	0043H	EBO (IE.5)	硬件	IPH.5, IP.5	2	Yes
看门狗定时器	WDTF	0053H	EWDI (EIE.4)	软件	EIPH.4, EIP.4	3	Yes
定时器0中断	TF0	000BH	ET0 (IE.1)	硬件软件	IPH.1, IP.1	4	No
I2C中断	SI	0033H	EI2 (EIE.0)	软件	EIPH.0, EIP.0	5	No
ADC转换器	ADCI	005BH	EADC (IE.6)	软件	IPH.6, IP.6	6	Yes ⁽¹⁾
外部中断1	IE1	0013H	EX1 (IE.2)	硬件软件	IPH.2, IP.2	7	Yes
KBI中断	KBIF[7:0]	003BH	EKB (EIE.1)	软件	EIPH.1, EIP.1	8	Yes
定时器1中断	TF1	001BH	ET1 (IE.3)	硬件软件	IPH.3, IP.3	9	No
串口Tx 和 Rx	TI & RI	0023H	ES (IE.4)	软件	IPH.4, IP.4	10	No
PWM中断	BKF	0073H	EPWM (EIE.5)	软件	EIPH.5, EIP.5	11	No
SPI	SPIF + MODF + SPIOVF	004BH	ESPI (EIE.6)	软件	EIPH.6, EIP.6	12	No
定时器2溢出中断	TF2	002Bh	ET2 (EIE.7)	软件	EIPH.7, EIP.7	13	No
捕获	CAPF0-2	0063H	ECPTF (EIE.2)	软件	IPH.7, IP.7	14 (lowest)	No

Note: 1. The ADC Converter can wake up “Power Down Mode” when its clock source is from internal RC.

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 $\overline{INT0}$ 和 $\overline{RI+TI}$ 在机器周期的 C3 采样并且他们相应的中断标志 IEx 自动的置位或清除。定时器 0 和 1 溢出标志在机器周期的 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 N79E815 系列执行写 IE, EIE, EIE2, IP0, IP0H, IP1, IP1H, IP2 或 IP2H 和 MUL、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 IE, EIE, EIE2, IP0, IP0H, IP1, IP1H, IP2 或 IP2H 访问 2 机器周期，完成 MUL 或 DIV 指令 5 机器周期和完成硬件 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 中断的SFR

中断的相关SFRs 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

Address: A8H

reset value: 0000 0000b

位	名称	描述
7	EA	使能所有中断。 0 = 禁止所有中断 1 = 使能中断。
6	EADC	使能ADC中断。
5	EBOD	使能BOD中断。
4	ES	使能串口中断 (UART). 0 = 禁止串口中断 1 = 使能串口中断。设定TI(SCON.1)或RI(SCON.0).

位	名称	描述
3	ET1	使能定时器1中断。 0 = 禁止定时器1中断 1 = 使能由TF1 (TCON.7)产生的中断.
2	EX1	使能外部中断1。 0 = 禁止外部中断1。 1 = 使能由 $\overline{\text{INT1}}$ pin (P3.3)产生的中断.
1	ET0	使能定时器0中断。 0 = 禁止定时器0中断 1 = 使能由TF0 (TCON.5)产生的中断.
0	EX0	使能外部中断0。 0 = 禁止外部中断0。 1 = 使能由 $\overline{\text{INT0}}$ pin (P3.2)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

Address: E8H

reset value: 0000 0000b

位	名称	描述
7	ET2	0: 禁止定时器 2 中断。 1: 使能定时器2中断.
6	ESPI	SPI 中断使能: 0: 禁止SPI中断。 1: 使能SPI中断.
5	EPWM	0: 当外部brake引脚brake时禁止PWM中断。 1: 当外部brake引脚brake时使能PWM中断.

位	名称	描述
4	EWDI	0: 禁止看门狗定时器中断. 1: 使能看门狗定时器中断.
3	-	保留.
2	ECPTF	0: 禁止捕获中断. 1: 使能捕获中断.
1	EKB	0: 禁止键盘中断. 1: 使能键盘中断.
0	EI2C	0: 禁止I2C中断. 1: 使能I2C中断.

IP – 中断优先级0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

Address: B8H

reset value: 0000 0000b

位	名称	描述
7	PCAP	1: 设置Capture 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 – 中断高优先级寄存器

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
Address: B7H						reset value: 0000 0000b	

位	名称	描述
7	PCAPH	1: 设置 Capture 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 –中断优先级-1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

Address: FFH

reset value: 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 brake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.
0	PI2	1: 设置I2C的中断优先级为较高优先级

EIPH – 中断高优先级-1寄存器

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
-----	-----	-----	-----	---	---	-----	-----

Address: F7H reset value: 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's 外部 brake的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 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

Address: 88H reset value: 0000 0000b

位	名称	描述
3	IE1	外部中断1边沿标志. 当检测到边沿/电平类型时, 该标志由硬件置位. 如果 IT1 = 1, 该位将保持置位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	外部中断1类型选择. 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.

位	名称	描述
1	IE0	<p>外部中断0边沿标志.</p> <p>当检测到边沿/电平类型时, 该标志由硬件置位. 如果 IT0 = 1, 该位将保持置位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.</p>
0	IT0	<p>外部中断0类型选择.</p> <p>该位选择 $\overline{INT0}$ 的中断触发类型是下降沿还是低电平.</p> <p>0 = $\overline{INT0}$ 为低电平触发.</p> <p>1 = $\overline{INT0}$ 为下降沿触发.</p>

20 在系统编程(ISP)

内部程序存储器支持 在系统编程功能(ISP). **N79E815**支持 ISP 模式 允许通过软件模式更新程序. 在需要不断更新应用程序的场合, ISP功能使这种工作变得简单而高效. 在某些应用场合, ISP功能使得用户不用打开机盒就能方便的进行固件升级.通常状况下 ISP 可通过 UART 执行 LDRUM, LDRUM 中驱动可更新 APROM 通过 ISP. Nuvoton 提供 ISP 驱动, USB ISP 烧写器和应用说明. 见下链接:

www.nuvoton.com/hq/enu/ProductAndSales/ProductLines/ConsumerElectronicsIC/Microcontroller/TechnicalSupportingFile.htm.

20.1 ISP程序

不同于 RAM's 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. **N79E815**. 使能 ISP 设定 ISPEN (CHPCON.0), 16-bit 地址 ISPAH ISPAL, 数据 ISPFID ISPCN. ISP 开始设定 ISPGO (ISPTRG.0). ISPTRG 为 TA 保护. CPU 保持编程数据 和 内嵌 ISP 控制电压和时序. ISP 完成后, 编程计数器 继续运行其它指令. ISPGO 将自动清除. 用户可重复下次 ISP. 用户可实现擦, 写, 读数据 通过软件.

下面为ISP的相关寄存器.

CHPCON – 芯片控制(TA protected)

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

Address: 9FH

reset value: see [Table 7-2. N79E815 Series SFR Descriptions and Reset Values](#)

位	名称	描述
6	ISPF	<p>ISP 错误标志.</p> <p>硬件置位:</p> <p>1. 出现的条件,</p> <p>(a)擦 写 APROM 当 APROM 运行.</p> <p>(b)擦 写 LDRUM 当 APROM 运行 LDUEN = 0.</p> <p>(c)擦 写 配置位 当 APROM 运行.</p> <p>(d)擦 写 LDRUM 当 LDRUM 运行.</p> <p>(e) 访问溢出.</p>

位	名称	描述
		<p>2. ISP 运行 从内部程序存储区到外部存储区.</p> <p>该位由软件清除.</p>
5	LDUEN	<p>更新 LDROM 使能.</p> <p>0 = LDROM 禁止擦写 当 APROM 运行. LDROM 只读.</p> <p>1 = LDROM 允许访问 当 APROM 运行.</p>
2	-	保留
1	BS	<p>启动选择.</p> <p>有不同方式对该位写或读.</p> <p>写:</p> <p>决定MCU在复位后从哪个区块启动.</p> <p>0 = 下一次从APROM中重启.</p> <p>1 = 下一次从LDROM中重启.</p> <p>读:</p> <p>定义MCU当前复位从哪区块启动.</p> <p>0 = 当前从APROM中重启.</p> <p>1 = 当前从LDROM中重启.</p>
0	ISPEN	<p>ISP 使能.</p> <p>0 = 禁止ISP.</p> <p>1 = 使能ISP.</p> <p>使能ISP 功能 将开始内部22.1184MHz RC振荡器为时序控制, 清ISPEN位后, 将停止内部RC 以减少电源消耗.</p>

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

Address: AFH

reset value: 0000 0000b

位	名称	描述
7:6	ISPA[17:16]	ISP 控制. 该字节用于ISP控制命令以决定ISP的目的的动作.
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							

Address: A7H

reset value: 0000 0000b

位	名称	描述
7:0	ISPA[15:8]	ISP 地址高字节. ISPAH 的地址 ISPA[15:8]用于ISP操作.

ISPAL – ISP 地址低字节

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

Address: A6H

reset value: 0000 0000b

位	名称	描述
7:0	ISPA[7:0]	ISP 地址低字节 ISPAL的地址ISPA[7:0]用于ISP操作.

ISPF D – ISP Flash数据

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

Address: AEH

reset value: 0000 0000b

位	名称	描述
7:0	ISPF D[7:0]	ISP flash数据.


ISPTRG – ISP 触发 (TA protected)

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

Address: A4H

reset value: 0000 0000b

Bit	Name	Description
0	ISPGO	<p>ISP开始.</p> <p>设定该位，启动ISP。该指令执行后，CPU会保持程序指针(PC),ISP接管控制程序运行，当ISP执行完毕后，PC指针继续运行。ISPGO自动清除。任何情况下读取ISPGO的值都为0.</p>

20.2 ISP 模式表

ISP Mode		ISPCN				ISPAH, ISPAL A[15:0]	ISPFID D[7:0]
		A17, A16	FOEN	FCEN	FCTRL[3:0]		
Read Company ID		x, x	0	0	1011	x	Data out D[7:0]=DAH
Read Device ID		x, x	0	0	1100	A[15:0]=0000H for low-byte ID A[15:0]=0001H for high-byte ID	Data out D[7:0]=Dev. ID
AP	FLASH Page Erase	0, 0	1	0	0010	Address in A[15:0] ^{Note(3)}	x
	FLASH Program	0, 0	1	0	0001	Address in A[15:0]	Data in D[7:0]
	FLASH Read	0, 0	0	0	0000	Address in A[15:0]	Data out D[7:0]
LD	FLASH Page Erase	0, 1	1	0	0010	Address in A[15:0] ^{Note(3)}	x
	FLASH Program	0, 1	1	0	0001	Address in A[15:0]	Data in D[7:0]
	FLASH Read	0, 1	0	0	0000	Address in A[15:0]	Data out D[7:0]
CONFIG Page Erase		1, 1	1	0	0010	Address in A[15:0]=0000H	x
CONFIG Program		1, 1	1	0	0001	Address in A[15:0]	Data in D[7:0]
CONFIG Read		1, 1	0	0	0000	Address in A[15:0]	Data out D[7:0]

Note:

(0) 'x' 表示 '不关心'.

(1) 'FLASH' 包括APROM, DataFlash/DataFlash2 和 LDROM. 'CONFIG'表示MCU硬件配置.

(2) A17 和 A16的功能为FLASH的物理地址 (除CONFIG). 在 CONFIG模式下, A17 和 A16 必须全为1s. 对于 N79E866, A17 和A16 应该全为 '00', 因为它的FLASH大小不超过64KB.

(3) 每页为128字节长度. 因此每页的地址擦除0000, 0080H, 0100H, 0180H, 0200H, ..., 递增0080H.

20.3 N79E815 系列ISP程序访问表

Destination	UNLOCK		LOCK	
	ISP code residence		ISP code residence	
	APROM	LDROM	APROM	LDROM
APROM				
LDROM	[1]		[1]	
Data Flash				
CONFIGs		[2]		[2]
ID (read)				
Block color	Comment			
	Fully accessing			
	Read only			
	Accessing inhibit			
[1]	LDUE should be 1, or it will be read only			
[2]	New CONFIG functions after POR, WDT, or SW reset			

注 1: CONFIG full accessing by LD while LOCK.

注 2: Inhibit AP jmp to LD or LD jump to AP.

注 3: MCU run in APROM can't read CONFIGs.

20.4 ISP使用指南

ISP 提供便捷更新Flash内容的途径，但用户必须遵循一些规则，以确保ISP正常运行。否则有可能造成不可预测后果并对芯片造成损伤。请注意下列说明，有助于稳定的运行ISP功能。

(1) 无 ISP 需求时, 用户需清 ISPEN (CHPCON.0) 为 0. ISP 需求内部 22.1184MHZ RC.选择外部时钟, 禁止 ISP 停止 22.1184MHz RC节省电源.

(2) 装载 ISP 程序, 从内部到 外部存储器 中运行, ISP 将无法工作 并且设定ISPFI 确保数据安全.

(3) 配置位 可通过 ISP 访问 当执行 LDROM. 配置位 CBS 位 复位后 有效.

(4) 使能 LOCK 位 (配置位0.1)后, ISP 读, 写, 擦 等动作仍然有效.

(5) ISP 工作电压 V_{DD} 2.7V ~ 5.5V.

(6) APROM 和 LDROM 可通过 ISP 读.

用户若自行开发ISP编程，需特别注意，对CONFIG的擦除和写动作，必须放在最后一步来执行.

20.5 ISP示例代码

Common Subroutine for ISP

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 ISPGR to trigger an ISP processing
SETB EA
RET
```

Read Company ID

```
CALL Enable_ISP

MOV  ISPCN,#00001011b    ;select "Read Company ID" mode

CALL Trigger_ISP
MOV  A,ISPFID            ;now, ISPFID contains Company ID (should be DAH), move to ACC for further
use

CALL Disable_ISP
```

Read Device ID

```
CALL Enable_ISP
```



```
MOV  ISPCN,#00001100b    ;select "Read Device ID" mode

MOV  ISPAH,#00H          ;fill address with 0000H for low-byte DID
MOV  ISPAL,#00H          ;
CALL  Trigger_ISP
MOV  A,ISPFD             ;now, ISPFD contains low-byte DID, move to ACC for further use

MOV  ISPAH,#00H          ;fill address with 0001H for high-byte DID
MOV  ISPAL,#01H          ;
CALL  Trigger_ISP
MOV  A,ISPFD             ;now, ISPFD contains high-byte DID, move to ACC for further use

CALL  Disable_ISP
```



FLASH Page Erase (target address in APROM/DataFlash/LDROM area)

```

CALL  Enable_ISP

MOV  ISPCN,#00100010b ;select "FLASH Page Erase" mode, (A17,A16)=(0,0) for
APROM/DataFlash/LDROM

MOV  ISPAH,###H ;fill page address
MOV  ISPAL,###H ;
CALL  Trigger_ISP

CALL  Disable_ISP

```

FLASH Program (target address in APROM/DataFlash/LDROM area)

```

CALL  Enable_ISP

MOV  ISPCN,#00100001b ;select "FLASH Program" mode, (A17,A16)=(0,0) for
APROM/DataFlash/LDROM

MOV  ISPAH,###H ;fill byte address
MOV  ISPAL,###H ;
MOV  ISPFH,###H ;fill data to be programmed
CALL  Trigger_ISP

CALL  Disable_ISP

```

FLASH Read (target address in APROM/DataFlash/LDROM area)

```

CALL  Enable_ISP

MOV  ISPCN,#00000000b ;select "FLASH Read" mode, (A17,A16)=(0,0) for APROM/DataFlash/LDROM

MOV  ISPAH,###H ;fill byte address
MOV  ISPAL,###H ;
CALL  Trigger_ISP
MOV  A,ISPFH ;now, ISPFH contains the Flash data, move to ACC for further use

CALL  Disable_ISP

```



Config Page Erase (target address in Config area)

```

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 Config area)

```

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, respec-
tively
MOV   ISPAL,#??H          ;
MOV   ISPDF,#??H          ;fill data to be programmed
CALL  Trigger_ISP

CALL  Disable_ISP

```

Config Read (target address in Config area)

```

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, respec-
tively
MOV   ISPAL,#??H          ;
CALL  Trigger_ISP
MOV   A,ISPDF             ;now, ISPDF contains the CONFIG data, move to ACC for further use

CALL  Disable_ISP

```

21 电源管理

N79E815系列有若干节电选项来帮助用户减少电源消耗，节电模式分为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。

在节省电源模式下，看门狗定时器需要特别注意。在进入省电模式或被唤醒退出省电模式后，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

Address: 87H

reset value: see [Table 7–2. N79E815 Series SFR Descriptions and Reset Values](#)

位	名称	描述
1	PD	掉电模式. 设定MCU进入掉电模式. 在掉电模式下, 系统所有的时钟都停止工作设备进入停止状态. 系统所有的工作都停止, 这样电源的消耗就降至最低. 在这种情况下, 端口上输出其相应SFR寄存器内的值.
0	IDL	Idle模式. 在空闲模式下, 提供给CPU的时钟被切断, 但是中断、定时器、串行口的时钟照常工作. 这样CPU就进入冻结状态; 程序计数器、堆栈指针、程序状态字、累加器及其它一些寄存器的内容保持不变.

21.1 Idle 模式

在空闲模式下, 提供给CPU的时钟被切断, 但是中断、定时器、串行口的时钟照常工作. 这样CPU就进入冻结状态; 程序计数器、堆栈指针、程序状态字、累加器及其它一些寄存器的内容保持不变.

由于一些总线功能如定时器或串口仍然运行在Idle模式, 如果中断源使能, CPU可以从Idle模式唤醒. 用户可以向IDL (PCON.0)写入1使器件进入Idle模式. 在器件进入Idle模式之前, 设置IDL位的指令就是最后的指令.

Idle模式可由两种方法中止. 一, 中断引起退出Idle模式. 将自动清IDL位, 中止Idle模式, 执行中断服务程序(ISR). 在使指令RETI就跳出ISR. 二, 除软件复位的其他复位可以中止Idle模式.

21.2 掉电模式

掉电模式是N79E815进入的最低功耗状态. 保持功耗在“ μA ”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都冻结. 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.

N79E815系列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. BOD 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BOD检测. 除最低功耗外, 很少在掉电模式下使能BOD. 也不推荐这种用法. 当然RST引脚的复位和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.

N79E815 系列可以采用外部中断唤醒掉电模式, 提供相应的中断使能和置位EA (IE.7). 如果这些条件都满足, 外部引脚触发同步重启系统时钟. 然后器件执行中断服务程序 (ISR) 响应外部中断. 完成ISR之后, 程序执行返回.

BOD 中断是另一个可以将CPU由掉电模式唤醒的中断源.

22 时钟系统

N79E815提供三种时钟源. 通过配置 FOSC (配置位3.1). 包括 外部包括外部时钟输入选项 晶振振荡器输入选项和片内部振荡器. N79E815支持片内22.1184MHz/11.0592MHz 通过 配置位 设定, 室温下 工厂 误差为 ± 1%. 选择外部晶振 范围为4MHz ~24MHz .

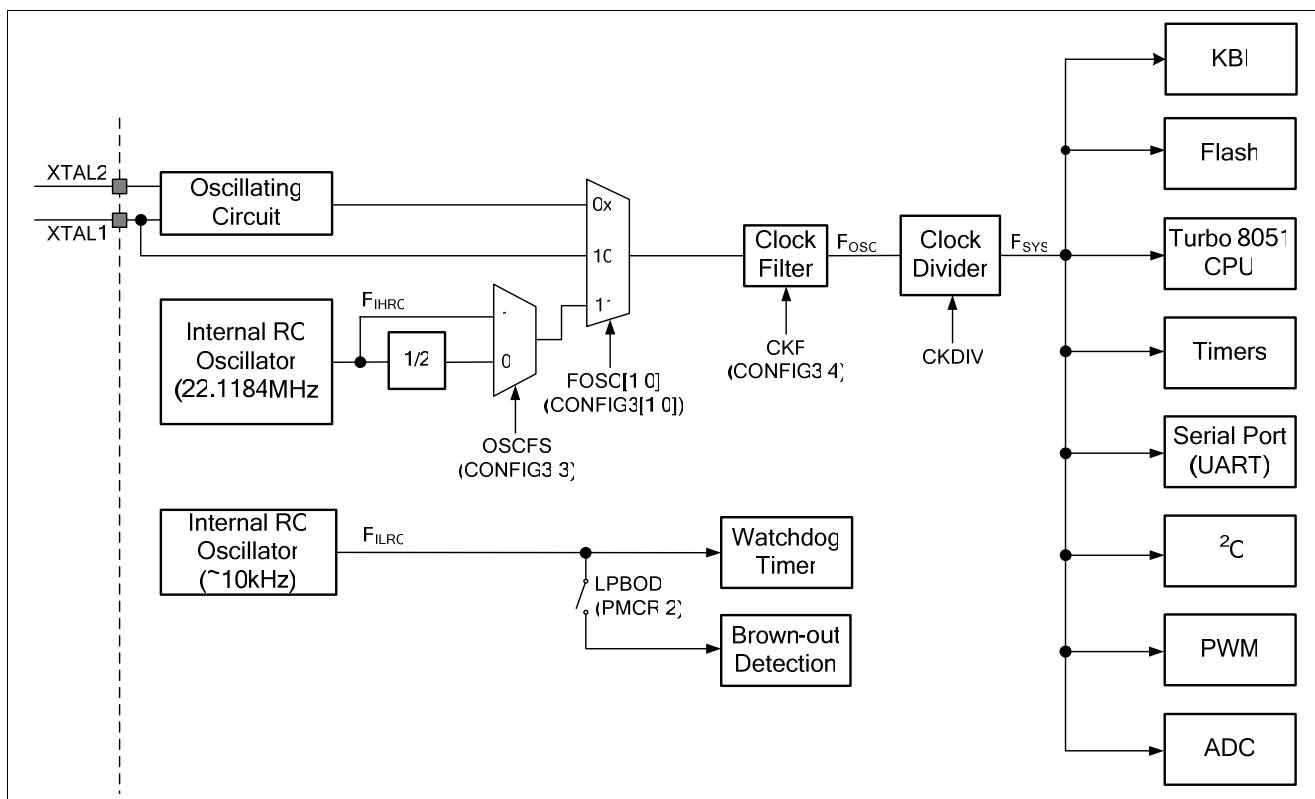


图 22-1. 系统时钟框图

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

unprogrammed value: 1111 1111b

位	名称	描述
7	CWDTEN	<p>CONFIG 看门狗定时器使能.</p> <p>1 = 所有复位之后禁止看门狗定时器</p> <p>0 = 所有复位之后使能看门狗定时.</p> <p>WDCIN.WDTEN. 由CWDTEN取位初始化.</p>

位	名称	描述								
6	-	保留.								
5	-	保留.								
4	CKF	时钟滤波使能. 1 = 使能时钟滤波器. 可以提高噪声抗干扰能力, 增强EMC. 0 = 禁止时钟滤波器.								
3	OSCFS	内部RC振荡器频率选择. 1 = 选择 22.1184MHz 做为内部RC晶振. 0 = 选择 11.0592MHz做为内部RC晶振.								
2	-	保留.								
1:0	FOSC1 FOSC0	振荡器选择位. 芯片时钟源选择, 见下表. <table border="1" data-bbox="516 1024 1390 1211"> <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>XTAL1-引脚的外部振荡</td> </tr> <tr> <td>(0, x)</td> <td>外部晶振, 4MHz ~ 24MHz</td> </tr> </tbody> </table>	(FOSC1, FOSC0)	芯片时钟源	(1, 1)	内部 RC 振荡器	(1, 0)	XTAL1-引脚的外部振荡	(0, x)	外部晶振, 4MHz ~ 24MHz
(FOSC1, FOSC0)	芯片时钟源									
(1, 1)	内部 RC 振荡器									
(1, 0)	XTAL1-引脚的外部振荡									
(0, x)	外部晶振, 4MHz ~ 24MHz									

DIVM – 时钟分频寄存器

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

Address: 95H

reset value: 0000 0000b

位	名称	描述
7:0	DIVM[7:0]	时钟分频. 系统时钟频率 F_{SYS} 公式 $F_{SYS} = F_{OSC}$, 当 $DIVM = 00H$. $F_{SYS} = \frac{1}{2(DIVM + 1)} \times F_{OSC}$, 当 $DIVM = 01H \sim FFH$.

22.1 外部时钟源

该系统的时钟源可以从外部XTAL1管脚灌入时钟信号。当采用外部时钟信号时，XTAL2应悬空。XTAL1和XTAL2分别为输入和输出，内部反相放大器。当采用外部晶振时，连接XTAL1和XTAL2引脚，允许连接晶振的频率为4MHz至24MHz。CKF(CONFIG3.4)用于对XTAL1时钟进行滤波。

22.2 片上RC振荡器

当FOSC (CONFIG3.1~0)为1时，使能片上RC振荡器。设置 OSCFS (CONFIG3.3)为1 将切换到divided-by-2 path. 注 当片上RC振荡器作为系统时钟源时，XTAL1应该连接到V_{DD}。

23 电源监控

为了防止在上电或电源不稳时出现错误，N79E815系列提供3个电源监视功能：上电检测，欠压(BOD)检测和低压检测。

23.1 上电检测

上电检测功能作用在检测电源电压上升到欠压检测可以工作的地方。POF (PCON.4)设置“1”表示初始电源上升。POF标志由软件清除。

23.2 BOD 检测

欠压检测功能是检测电源电压下降到欠压电压值，防止错误操作或提供电源报警。N78E366提供4级电压2.2V, 2.7V, 3.8V, 和 4.5V 设定，通过设定配置位2内的CBOV[1:0]可选择欠压电压。当系统电压 V_{DD} 降到选择 V_{BOD} 的值时，欠压监测器将检测出，并进行复位CPU 或请求欠压中断。

当BORST (PMC.4) =0后， V_{DD} 低于 V_{BOD} 时BOF (PMC.3) 将置1.当 V_{DD} 仍低于 V_{BOD} 时，用户亦可软件清除该位，BOF 不会再次置位。BOF仅在发生电源电压掉落时响应。当 V_{DD} 高于 V_{BOD} 时， BOF 置 1，状态标志位 BOS (PMC.0)用于指示状态，如果BORSR 位置位，将使能 欠压复位功能。欠压复位发生后，BORF (RSR.2) 置 1。该位通过软件清除。VBOD有一段20~200mV迟滞电压带。

欠压检测自备节能模式，当LPBOD设置为1，欠压检测节能模式打开，每隔12.8ms才进行一次检测。注，在欠压检测省电模式下，电压迟滞带消失

另一种电源监控功能，BOD 检测电路用作监控 V_{DD} 。有两个可编程的BOD触发电平。通过设置CONFIG2 的CBOD 选择2.7V 和4.2V。当 V_{DD} 降到所选的 BOD 触发电平(V_{BOD})，BOD检测逻辑复位CPU或请求BOD中断。用户根据不同应用决定是BOD复位还是使能中断。

当BORST (PMC.4) =0后， V_{DD} 低于 V_{BOD} 时BOF (PMC.3) 将置1.当 V_{DD} 仍低于 V_{BOD} 时，用户亦可软件清除该位，BOF 不会再次置位。BOF仅在发生电源电压掉落时响应。当 V_{DD} 高于 V_{BOD} 时， BOF 置 1，状态标志位 BOS (PMC.0)用于指示状态，如果BORSR 位置位，将使能 欠压复位功能。欠压复位发生后，BORF (RSR.2) 置 1。该位通过软件清除。VBOD有一段20~200mV迟滞电压带



当BORST (PMCR.4)为0, BOD 检测到电压下降到 V_{BOD} 时请求中断, 这种情况下, LVF (PMCR.3) 将置 1. 当 V_{DD} 仍低于 V_{BOD} 时, 用户亦可软件清除该位, LVF不会再次置位. LVF仅在发生电源电压掉落时响应. 在 V_{DD} 高于 V_{BOD} 之后LVF将置1, 表示电源恢复正常. V_{BOD} 有一段20~200mV的迟滞电压带.

BOD 检测电路提供一个低压BOD检测模式用于节电功能. 当 BODPS设置为1, BOD 大约每12.8ms 检测一次电源电压. 由于需要间隔计数, 内部 10kHz RC 振荡器将在BOD低压模式时打开. 注: 在低压BOD检测模式下没有迟滞特征.

CONFIG2

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

unprogrammed value: 1111 1111b

位	名称	描述									
7	CBODEN	<p>BOD 检测使能.</p> <p>1 = 禁止BOD检测.</p> <p>0 = 使能BOD检测.</p>									
6	CBOV	<p>BOD 电压选择.</p> <p>该位选择BOD的电压级.</p> <table border="1" data-bbox="430 1249 1461 1381"> <thead> <tr> <th>Config-bits CBOV</th> <th>SFR BOD</th> <th>BOD 电压</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>使能BOD= 2.7V</td> </tr> <tr> <td>0</td> <td>1</td> <td>使能BOD=4.2V</td> </tr> </tbody> </table>	Config-bits CBOV	SFR BOD	BOD 电压	1	0	使能BOD= 2.7V	0	1	使能BOD=4.2V
Config-bits CBOV	SFR BOD	BOD 电压									
1	0	使能BOD= 2.7V									
0	1	使能BOD=4.2V									
5	-	保留									
4	CBORST	<p>BOD 复位使能.</p> <p>该位检测是否引起BOD复位.</p> <p>1 = 当V_{DD}降到V_{BOD}以下, 使能BOD复位.</p> <p>0 = 当V_{DD}降到V_{BOD}以下, 禁止BOD复位. 当V_{DD}降到V_{BOD}以下, 芯片将声明LVF.</p>									

PORDIS – 禁止POR (TA protected)

7	6	5	4	3	2	1	0
PORDIS[7:0]							
w							

Address: FDH

reset value: 0000 0000b

Bit	Name	Description
7:0	PORDIS[7:0]	POR 禁止.(One time POR) 首先向PORDIS写入5AH，然后立即写入A5H 将禁止POR.

PMCR – 电源监控控制(TA protected)

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

Address: A3H

reset value: see [Table 7–2. N79E815 Series SFR Descriptions and Reset Values](#)

位	名称	描述						
7	BODEN	BOD-检测功能控制. 复位时，BODEN 由CBODEN (CONFIG2, bit-7)取反初始化。 1 = 使能 BOD 检测。 0 = 禁止BOD 检测。						
6	BOV	BOD 电压选择位: BOD 电压选择位: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>CBOV (Config-bits)</th> <th>BOD 电压</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>使能 BOD= 2.7V,</td> </tr> <tr> <td>1</td> <td>使能 BOD=4.2V</td> </tr> </tbody> </table>	CBOV (Config-bits)	BOD 电压	0	使能 BOD= 2.7V,	1	使能 BOD=4.2V
CBOV (Config-bits)	BOD 电压							
0	使能 BOD= 2.7V,							
1	使能 BOD=4.2V							
5	-	保留						
4	BORST	BOD 复位使能. 该位决定是否产生BOD复位。 0 = 当V _{DD} 降到V _{BOD} 以下，禁止BOD复位. 当V _{DD} 降到V _{BOD} 以下，芯片将声明LVF 1 =当V _{DD} 降到V _{BOD} 以下，使能BOD复位...						

位	名称	描述
3	BOF	BOD 标志. 在V _{DD} 降到V _{BOD} 以下或升到V _{BOD} 以上该标志将置1. 如果EBOD (IE.52) 和 EA (IE.7) 都置位, 请求BOD中断. 该位必须由软件清.
2	LPBOD	低电压BOD 检测使能. 该位切换BOD检测进入省电模式. 当CBODEN = 1, 该位有效. 0 = 禁止 BOD 省电模式. 如果使能, BOD检测工作在普通模式下.检测依然进行. 1 = 使能BOD省电模式. 如果使能, BOD 检测工作于省电模式. 使能该位将打开内部10kHz RC 用于12.8ms 检测间隔的定时器.
1	-	保留
0	-	保留

[1] BODEN and BORST will be directly loaded from CONFIG2[7:4] after all resets.

注: 如果芯片复位后LVF为1, 强烈建议通过清LVF初始化用户程序.

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

Address: 87H reset value: see [Table 7-2. N79E815 Series SFR Descriptions and Reset Values](#)

位	名称	描述
4	POF	上电复位标志. 在上电复位后该位置1. 其他复位之后该位保持其值. 该标志由软件清零.

24 复位条件

N79E815支持多种复位条件. 一般来说许多寄存器在复位后都将回到其初始值, 而不管复位的类型如何. 但有些标志位的状态取决于复位的类型. 用户可以根据这些标志位来判断复位的类型. 有6种方法可以将系统复位, 上电复位, 低压复位, RST 脚 复位, 软件复位, 看门狗复位, 欠压复位.

AUXR1 – AUX 功能寄存器-1

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

Address: A2H

reset value: 0000 0000b

位	名称	描述
7	SPI_Sel	0: 选择 P2.2, P2.3, P2.4, P2.5 为 SPI引脚. 1: 选择 P1.7, P1.6, P1.4, P0.0为 SPI引脚.
6	UART_Sel	0: 选择 P1.0, P1.1为UART 引脚. 1: 选择P2.6, P2.7为UART 引脚.
5:4	-	保留
3	DisP26	0: 使能 P2.6数据输入和输出. 1: 禁止P2.6数据输入和输出.
2:1	-	保留
0	DPS	双数据指针选择 0: 选择标准8051的DPTR. 1: 选择DPTR1

WDCON0 – 看门狗定时器控制(TA protected)

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

Address: D8H

reset value: see [Table 7–2. N79E815 Series SFR Descriptions and Reset Values.](#)

位	名称	描述
3	WDTF	<p>WDT 复位标志. 当MCU自身复位时, 该位由硬件置位, 该位由软件清零.</p> <p>如果 EWRST=0, 中断标志WDTF将不能由硬件置位, MCU将立即自身复位.</p> <p>如果 EWRST=1, 中断标志WDTF 将由硬件置位且MCU将跳到WDT's 中断服务程序 (如果使能WDT中断), MCU在512个CPU时钟之后才复位.</p>

24.1 上电复位

N79E815 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flash时, 不访问程序flash. 如果从程序flash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, 所选的振荡器运行, 程序从0000H开始执行. 同时, 上电标志 POF (PCON.4) 置1表示复位, 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, 以检测在下一次复位发生之后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.

24.2 BOD 复位

欠压复位 用于侦测 V_{DD} 电压. 当 V_{DD} 下降到欠压电压 (V_{BOD}), BORST (PMC.4)置 1 将引起欠压复位. 欠压复位后, BORF (RSR.2)置 1. 复位后该位保留 (除上电复位) 可由软件清除.

24.3 RST引脚复位

系统在每个机器周期的C4态对RST管脚进行连续的采样. 因此RST管脚上的电平至少要维持2个机器周期, 以保证系统检测到有效的RST高电平. 然后复位电路将同步发出复位信号, 因此复位是一个同步的动作, 要求时钟在此期间一直运行来实现外部复位.

系统进入复位状态以后，只要RST脚上电平一直为高，那么系统就一直处于复位状态中。在RST信号撤除后，系统仍将会在2个机器周期内保持复位状态，然后才从0000H处开始执行程序。对外部复位来说，没有与之配套的标志位。但是由于另外的2种复位模式都有相应的标志位存在，那么当其它2个标志位为零时，可以将外部复位认为是默认的复位情况..

如果CPU在掉电模式，有RST引脚复位，这种方法触发硬件复位。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。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。

24.4 看门狗定时复位

看门狗定时器是一个带可编程溢出时间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。当看门狗定时器溢出后，将会产生一个中断（如果该中断打开）如果用户允许看门狗定时器产生复位信号，复位后系统从0000H运行。

一旦看门狗复位标志位WDTRF (WDCON.3) 置位。该位只有软件清除。

24.5 软件复位

N79E815支持软件复位。可适用于ISP结束后，如果LDRAM更新APROM ISP完成 APROM 中代码已更新，软件复位可决定CPU从APROM中启动检查APROM中代码。写1到SWRST (CHPCON.7) 触发软件复位。该位为TA保护。复位后SWRF (RSR.0) 由硬件置位。复位后该位保留（除上电复位）可由软件清除，SWDTRF可通过软件清零。

CHPCON – 芯片控制(TA protected)

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

Address: 9FH

reset value: see [Table 7-2. N79E815 Series SFR Descriptions and Reset Values](#)

位	名称	描述
7	SWRST	软件复位。 该位置1产生软件复位。该位由硬件清除。

以下为软件demo code:

```
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 启动选择

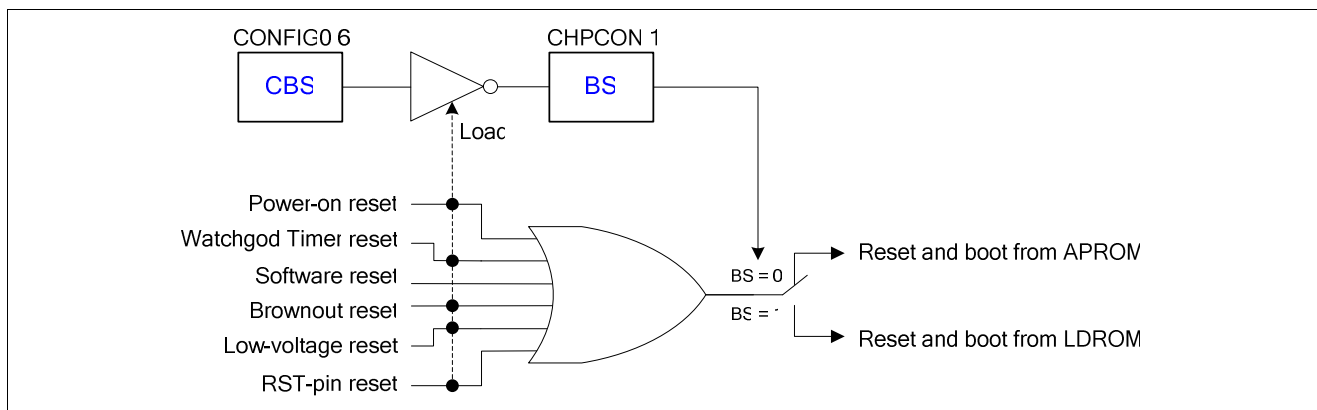


图 24-1. 启动选择框图

N79E815支持启动选择. SFR BS 位 (CHPCON.1) 决定 CPU 从 APROM 或 LDROM 中启动. BS = 0, CPU 从 APPROM 中启动. 否则, CPU 从 LDROM 中启动.

CONFIG0

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

unprogrammed value: 1111 1111b

位	名称	描述
7	CBS	<p>CONFIG 启动选择.</p> <p>1 =复位后 MCU 从 APROM 中启动 除软件复位.</p> <p>0 = 复位后 MCU 从 LDROM 中启动除软件复位.</p>

CHPCON – 芯片控制 (TA protected)

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

Address: 9FH

reset value: see [Table 7–2. N79E815 Series SFR Descriptions and Reset Values](#)

位	名称	描述
1	BS	启动选择. 写. 0 = 从 APROM 中启动. 1 = 从 LDROM 中启动. 读. 0 = 从 APROM 中启动. 1 = 从 LDROM 中启动.

[1] 除软件复位后 CBS 配置位0.7 位 重置. 软件复位后 该位值不变.

CPU 从所有状况复位后, 将硬件检查 BS 位 替换为CBS 位 , 以确定复位后程序是由 APROM 还是由 LDROM 中启动运行.

24.7 复位状态

复位状态不会影响片上RAM. RAM上的数据在复位期间保持不变. 如果V_{DD}低于1.2V, RAM上内容可能会丢失. 因此, 上电复位后RAM的内容不确定. 在电源失效条件, 如果电源降到数据保存的最小电压以下, RAM的内容将丢失.

大多数SFR在复位后回到其初始状态. 见 [Table 7–2. N79E815 Series SFR Descriptions and Reset Values](#). 一些特殊功能寄存器的初始值取决于不同复位源. 参考 表 24-, 程序计数器为0000H 并且只要有复位条件满足就保存. 注堆栈指针也复位到07H, 因此堆栈内容可能丢失.

复位后, 中断和定时器都禁止. 如果复位源是上电复位, 看门狗定时器也禁止. I/O 口的SFRs 写入FFH, 将端口引脚拉为高状态.

表 24-2 不同复位下的SFR的初始状态

SFR	上电复位 (LVR Reset)	看门狗复位	软件/外部复位	BOD复位	带时控保护
WDCON0 (D8H)	C000 0000B b7(ENWDT)= /CENWDT(config3.7)	C0Uu 1UUUB	C0UU UUUUB		Y
WDCON1 (ABH)	0000 0000B				Y
ISPTRG (A4H)	XXXX XXX0B				Y
PMCR (A3H)	CXCC 10XXB b[7:4]=config2	UXUU U0XXB		UXUU 10XXB	Y
CHPCON (9FH)	0000 00C0B b1(BS)=/CBS	000X XUJ0B			Y
SHBDA (9CH)	Config1	Unchanged			Y
PCON (87H)	0001 000b	00uu 0000b	00uu 0000b (SW/External reset)	00uu 0000b	N

Note: The writes of AAH and 55H must occur within 3 machine-cycles of each other. Interrupts should be disabled during this procedure to avoid delay between these two writes.

25 配置位 (CONFIG)

N79E815 具有硬件配置位 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. **N79E815**具有 三个 配置位 位 配置位0, 2 和 3. 这些功能由 配置位 位 进行默认配置, 同时可用 SFR 位 再次配置.因此 需要装载 配置位 位到 相应的SFR 位. (例如软件复位需要重新装载 配置位 位, 除CBS 配置位0.) SFR 位 可通过软件改写. 其它复位 SFR位 保持不变.

配置位 位 "-" 表示不能被配置 (始终保持为1).

25.1 配置位0

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

unprogrammed value: 1111 1111b

位	名称	描述
7	CBS	<p>CONFIG 启动选择.</p> <p>该位配置 MCU 除软件复位后的启动 位置.</p> <p>1 = MCU 从 APROM 中启动 (除软件复位).</p> <p>0 = MCU 从 LDROM 中 启动 (除软件复位).</p>
6:2	-	保留.
1	LOCK	<p>芯片锁定使能.</p> <p>1 = 不锁定芯片.</p> <p>0 = 锁定芯片.</p> <p>注CONFIG 字节通常不锁定, 可以读取. 否则, 一旦芯片被锁定, CONFIG 字节不能被擦除或独立编程. 这是唯一禁止芯片锁定的方法, 以用于整个芯片擦除模式. 因此, APROM, LDROM, Data Flash, 和其他 CONFIG 的数据在执行这个过程时被擦除.</p> <p>如果芯片锁定, 就不更改ISP功能.</p>

位	名称	描述
0	DFEN	<p>使能数据 Flash. (N79E815 only)</p> <p>1 = 没有数据Flash空间. APROM的大小为16k-byte.</p> <p>0 = 存在数据 Flash. 数据 Flash 和APROM 共享16k bytes (取决于SHBDA 的设置).</p>

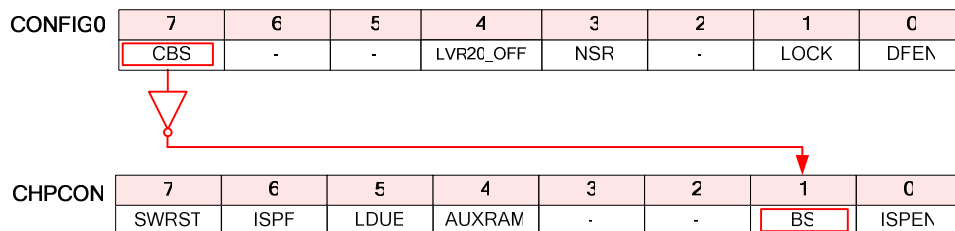


图 25-1. CONFIG0

25.2 配置位1 (N79E815 Only)

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

unprogrammed value: 1111 1111b

位	名称	描述
7:0	CHBDA[7:0]	<p>配置数据Flash开始地址的高字节.</p> <p>仅DFEN (CONFIG0.0)当为0时, 该字节有效. 用于决定数据Flash的开始地址.</p>

[1] 注如果设置CHBDA 为00H 就没有APROM. CPU 将执行内部程序内存的最小256B中的代码.

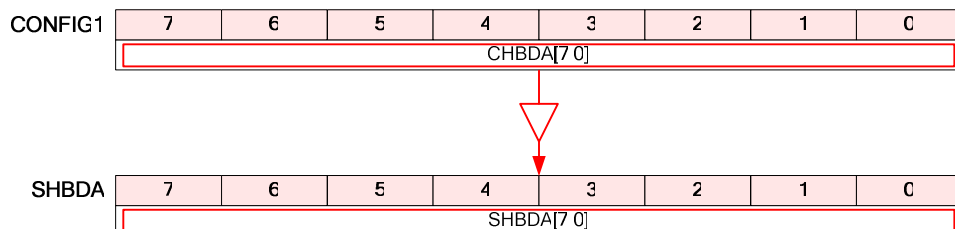


图 25-2. 配置位1

25.3 配置位2

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

unprogrammed value: 1111 1111b

位	名称	描述						
7	CBODEN	使能配置BOD检测. 1 = 使能BOD检测. 0 = 禁止BOD检测.						
6	CBOV	配置BOD电压选择. 该位选择BOD的电压级 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>CBOV</th> <th>BOD voltage</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Enable $V_{BOD} = 2.7V$</td> </tr> <tr> <td>0</td> <td>Enable $V_{BOD} = 4.2V$</td> </tr> </tbody> </table>	CBOV	BOD voltage	1	Enable $V_{BOD} = 2.7V$	0	Enable $V_{BOD} = 4.2V$
CBOV	BOD voltage							
1	Enable $V_{BOD} = 2.7V$							
0	Enable $V_{BOD} = 4.2V$							
5	-	保留.						
4	CBORST	配置使能 BOD复位. 该位决定在BOD事件后是否产生BOD复位. 1 = 当 V_{DD} 降低低于 V_{BOD} , 使能BOD复位. 0 = 当 V_{DD} 降低低于 V_{BOD} , 禁止BOD复位.						
3:0	-	保留.						

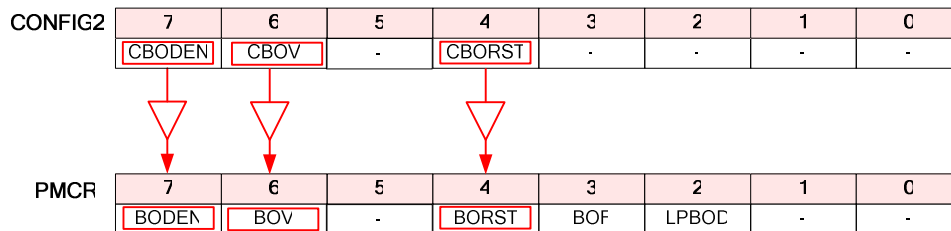


图 25-3. CONFIG2

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

unprogrammed value: 1111 1111b

位	名称	描述								
7	CWDTEN	配置看门狗定时器使能。 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。 WDCIN.WDTEN. 由CWDTEN取反初始化.								
6	-	保留.								
5	-	保留.								
4	CKF	使能时钟滤波器。 1 = 使能时钟滤波器. 可提高噪声抗干扰能力, 增强EMC. 0 = 禁止时钟滤波器.								
3	OSCFS	选择内部RC振荡频率。 1 = 选择 22.1184MHz 作为内部 RC 振荡器. 内部振荡器22.1184MHz 输出作为系统时钟源. 0 = 选择11.0592MHz作为内部 RC 振荡器. 内部振荡器11.0592MHz 输出作为系统时钟源								
2	-	保留.								
1	FOSC1	振荡器选择位.								
0	FOSC0	芯片时钟源选择,见下表. <table border="1" data-bbox="516 1612 1357 1812"> <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>XTAL1-pin的外部振荡器</td> </tr> <tr> <td>(0, x)</td> <td>外部晶振, 4MHz ~ 24MHz</td> </tr> </tbody> </table>	(FOSC1, FOSC0)	芯片时钟源	(1, 1)	内部 RC 振荡器	(1, 0)	XTAL1-pin的外部振荡器	(0, x)	外部晶振, 4MHz ~ 24MHz
(FOSC1, FOSC0)	芯片时钟源									
(1, 1)	内部 RC 振荡器									
(1, 0)	XTAL1-pin的外部振荡器									
(0, x)	外部晶振, 4MHz ~ 24MHz									

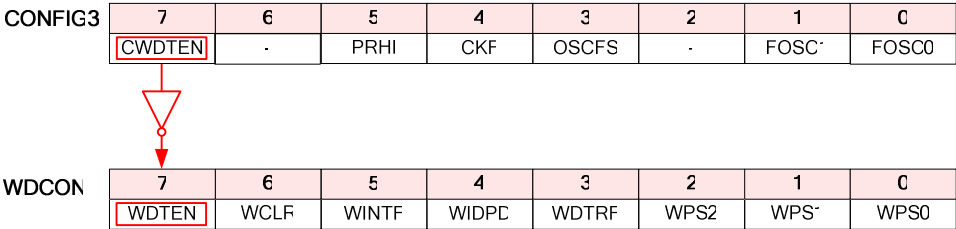


图 25-4. CONFIG3

26 指令集

N79E815执行标准8051 体系微处理器中的所有的指令. 所以指令代码为 8-为 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2~3 位 更多地数据.

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 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) 8-位内部 RAM 区域 (0~255) 地址 通过 R0或R1.
- #data 8-bit 常量指令.
- #data16 16-bit 常量指令.
- addr16 16-bit 目的地址. 使用LCALL和 LJMP 指令. 可作为64K 字节程序空间的任何地址区域.
- addr11 11-bit目的地址. 使用 ACALL 和AJMP指令.可用于2K 字节程序空间的首字节的下列指令
- rel 有符号(2's 补充) 8-位 转移字节. 使用 SJMP 和其它有效指令.长度 -128 ~ +127 字节.关联首字节的下列指令.
- bit 内部 RAM 或 SFR 地址位.

表 26-1. N79E815 系列指令集

指令	操作码	字节	时钟周期	N79E815 V.S. 传输 80C51的速率
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

表 26-1. N79E815 系列指令集

指令	操作码	字节	时钟周期	N79E815 V.S. 传输 80C51的速率
INC Rn	08~0F	1	4	3.0
INC @Ri	06, 07	1	4	3.0
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

表 26-1. N79E815 系列指令集

指令	操作码	字节	时钟周期	N79E815 V.S. 传输 80C51的速率
MOV Rn, A	F8~FF	1	4	3.0
MOV Rn, direct	A8~AF	2	8	3.0
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 ^[1]	E2, E3	1	8	3.0
MOVX A, @DPTR ^[1]	E0	1	8	3.0
MOVX @Ri, A ^[1]	F2, F3	1	8	3.0
MOVX @DPTR, A ^[1]	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 ^[2]	2	12	2.0
LCALL addr16	12	3	16	1.5
RET	22	1	8	3.0

表 26-1. N79E815 系列指令集

指令	操作码	字节	时钟周期	N79E815 V.S. 传输 80C51的速率
RETI	32	1	8	3.0
AJMP addr11	01, 21, 41, 61, 81, A1, C1, E1 ^[2]	2	12	2.0
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

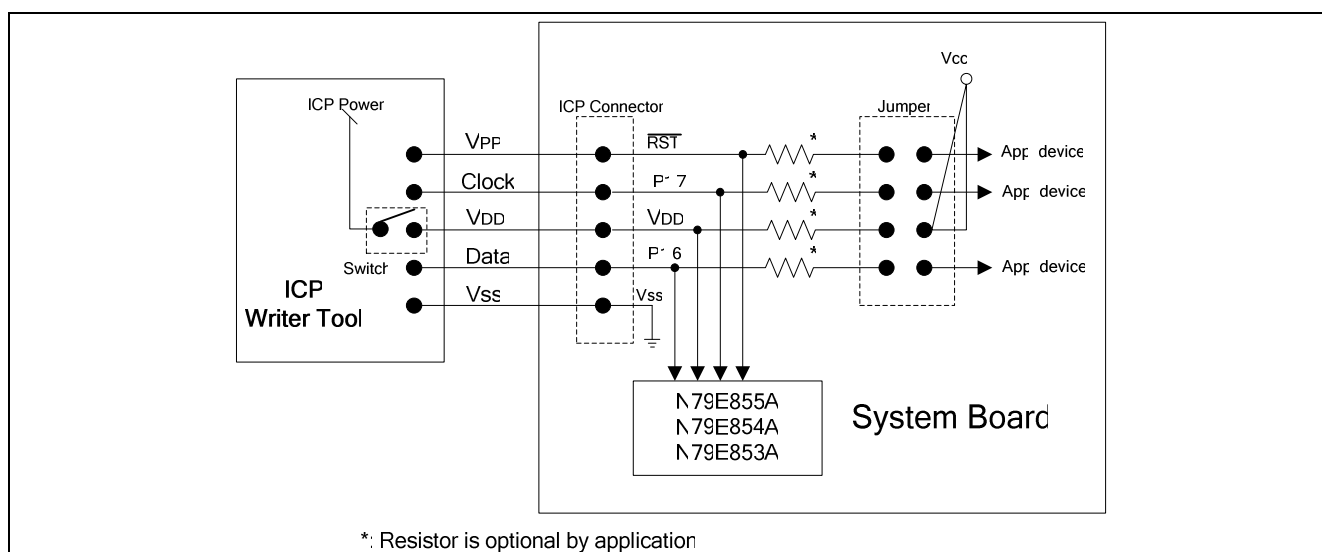
[1] 11位地址的最高三位 [A10:A8] 决定ACALL hex code. code 是 [A10,A9,A8,1,0,0,0,1].

[2] 11位地址的最高三位 [A10:A8] 决定AJMP hex code. code 是 [A10,A9,A8,0,0,0,0,1].

27 In-Circuit Program (ICP)

ICP (In-Circuit-Program) 模式是另一种访问Flash 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N79E815 系列的Flash EPROM. 用户可参考<http://www.manley.com.cn/english/index.asp> ICP 编程工具.

在进入 ICP 编程模式时, 所有引脚必须设置成准双向模式, 输出“1”. N79E815 系列支持对 Flash EPROM (16K/8K/4K bytes AP Flash EPROM) 和 DataFlash 内存 (128 bytes per page)编程. 用户可以选择编程 AP flash 和 DataFlash.



注:

- 1 更新代码时, /RST, P1.6 和 P1.7 必须断开和系统负载的联接
- 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 绝对最大额定值

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

超出表内“最大绝对额定值”的情况使用，会对器件的可靠性和寿命造成严重损害。

28.2 DC 电气特性

温度 = 25°C; V_{SS} = 0V;

V_{DD} = 2.4V to 5.5V @ F_{sys} = up to 24MHz

V_{DD} = 2.4V to 5.5V @ F_{sys} = 12MHz 或内部RC11.0592MHz

V_{DD} = 2.4V to 5.5V @ F_{sys} = 4MHz

表 25-1. DC 特性

(V_{DD}-V_{SS} = 2.4~5.5V, TA = -40~85°C, 除非特别说明.)

符号	参数	测试条件	最小值	典型值	最大值	单位
Svdd	保证内部上电复位信号的V _{dd} 上升速率	参考上电复位章节的描述	0.05 ^[7]	-	-	V/ms
V_{IL}	输入低电压 (GPIO with TTL input)	2.4 < V _{DD} < 5.5V	-0.5		0.2V _{DD} -0.1	V
V_{IL1}	输入低电压 (GPIO with Schmitt trigger input)	2.4 < V _{DD} < 5.5V	-0.5		0.3V _{DD}	V

表 25-1. DC 特性

(V_{DD}-V_{SS} = 2.4~5.5V, TA = -40~85°C, 除非特别说明.)

符号	参数	测试条件	最小值	典型值	最大值	单位
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 (GPIO with TTL input)	2.4 < V _{DD} < 5.5V	0.2V _{DD} +0.9		V _{DD} +0.5	V
V _{IH1}	输入高电压 (GPIO with Schmitt trigger input)	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 (GPIO of P0,P2,P3, all modes except input only)	V _{DD} =4.5V, I _{OL} = 20mA ^{[3], [4]}			0.45	V
		V _{DD} =3.0V, I _{OL} = 14mA ^{[3], [4]}			0.45	V
		V _{DD} =2.4V, I _{OL} = 10mA ^{[3], [4]}			0.45	V
V _{OL1}	输出低电压 (P10, P11, P14, P16, P17) (All modes except input only)	V _{DD} =4.5V, I _{OL} = 40mA ^{[3], [4]}			0.45	V
		V _{DD} =3.0V, I _{OL} = 27mA ^{[3], [4]}			0.45	V
		V _{DD} =2.4V, I _{OL} = 20mA ^{[3], [4]}			0.45	V
V _{OH}	输出高电压 (GPIO, quasi bi-directional)	V _{DD} =4.5V I _{OH} = -380μA ^[4]	2.4			V
		V _{DD} =3.0V I _{OH} = -90μA ^[4]	2.4			V
		V _{DD} =2.4V I _{OH} = -48μA ^[4]	2.0			V
V _{OH1}	输出高电压 (GPIO, push-pull)	V _{DD} =4.5V I _{OH} = -28.0mA ^{[3], [4]}	2.4			V

表 25-1. DC 特性

(V_{DD}-V_{SS} = 2.4~5.5V, TA = -40~85°C, 除非特别说明.)

符号	参数	测试条件	最小值	典型值	最大值	单位
		V _{DD} =3.0V I _{OH} = -7mA ^{[3], [4]}	2.4			V
		V _{DD} =2.4V I _{OH} = -3.5mA ^{[3], [4]}	2.0			V
I _{IL}	逻辑0输入电流 (GPIO, quasi bi-direction)	V _{DD} =5.5V, V _{IN} =0.4V		-40 @5.5V	-50	μA
I _{TL}	逻辑1向0转换电流 (GPIO, quasi bi-direction)	V _{DD} =5.5V, V _{IN} =2.0V ^[2]		-550 @5.5V	-650	μA
I _{LI}	输入漏电流 (GPIO, open-drain or input only)	0 < V _{IN} < V _{DD} +0.5		<1	±10	μA
I _{OP}	OP 电流 (Active mode ^[5])	XTAL 12MHz, V _{DD} =5.0V		3.1		mA
		XTAL 24MHz, V _{DD} =5.5V		6.4		mA
		XTAL 12MHz, V _{DD} =3.3V		1.7		mA
		XTAL 24MHz, V _{DD} =3.3V		3.2		mA
		Internal 22.1184MHz, V _{DD} =5V		2.3		mA
		Internal 22.1184MHz, V _{DD} =3.3V		2.2		mA
I _{IDLE}	IDLE 电流	XTAL 12MHz, V _{DD} =5.0V		2.7		mA
		XTAL 24MHz, V _{DD} =5.5V		4.4		mA
		XTAL 12MHz, V _{DD} =3.3V		1.3		mA

表 25-1. DC 特性

(V_{DD}-V_{SS} = 2.4~5.5V, TA = -40~85°C, 除非特别说明.)

符号	参数	测试条件	最小值	典型值	最大值	单位
		XTAL 24MHz, V _{DD} =3.3V		2.3		mA
		Internal 22.1184MHz, V _{DD} =5V		1.6		mA
		Internal 22.1184MHz, V _{DD} =3.3V		1.5		mA
I _{PD}	掉电模式			<2		μA
R _{RST}	RST-引脚内部上拉电阻	2.4 < V _{DD} < 5.5V	100		250	KΩ
V _{BOD42}	BOD42 检测电压	2.4 < V _{DD} < 5.5V		4.2		V
V _{BOD27}	BOD27 检测电压	2.4 < V _{DD} < 5.5V		2.7		V

[1]: 不一定有典型值. 所列出的值均在室温和基于有限个样品进行测试的.

[2]: 端口0~3的引脚的传输电流, 当V_{IN}接近2V时, 传输电流达到最大值.

[3]: 在稳定状态下 (non-transient), I_{OL}/I_{OH} 如下:

P0, P2, P3每个端口的最大I_{OL}/I_{OH}: 20 mA

P10, P11, P14, P16, P17的最大I_{OL}/I_{OH}: 40 mA

所有输出总的最大I_{OL}/I_{OH}: 100mA (Through V_{DD} total current)

所有输出总的最大I_{OL}/I_{OH}: 150mA (Through V_{SS} total current)

[4]: 如果 I_{OH} 超过测试条件, V_{OH}将低于列表规格.

如果 I_{OL} 超过测试条件, V_{OL} 将高于列表规格.

[5]: 在CPU保持复位状态时测得.

[6]: GPIO 表示通用I/O, 如 P0, P1, P2, P3.

[7]: 这些参数为特征, 没有测试.

其他: P1.2 和 P1.3 都是开漏结构, 没有准双向或推挽模式.

29 模拟电气特性

29.1 LDO 管理器规格

参数	单位	最小值	典型值	最大值	注
输入电压	V _{DD}	2.4	5	5.5	
输出电压	V _{LDO}	-10%	1.8	+10%	

29.2 10-位 SAR-ADC的规格

	符号	测试条件	最小值	典型值	最大值	单位
操作电压	V _{DD}	V _{DD}	2.7		5.5	V
精度					10	bit
转换时间				35t _{ADC} ^[1]		us
采样率					150K	Hz
整体非线性错误	INL		-1		1	LSB
差分非线性	DNL		-1		1	LSB
增益错误	Ge		-1		1	LSB
偏移错误	Ofe		-3		3	LSB
时钟频率	ADCCLK				5.25	MHz
绝对错误			-2		2	LSB
Band-gape	V _{BG}		1.15	1.3	1.45	V

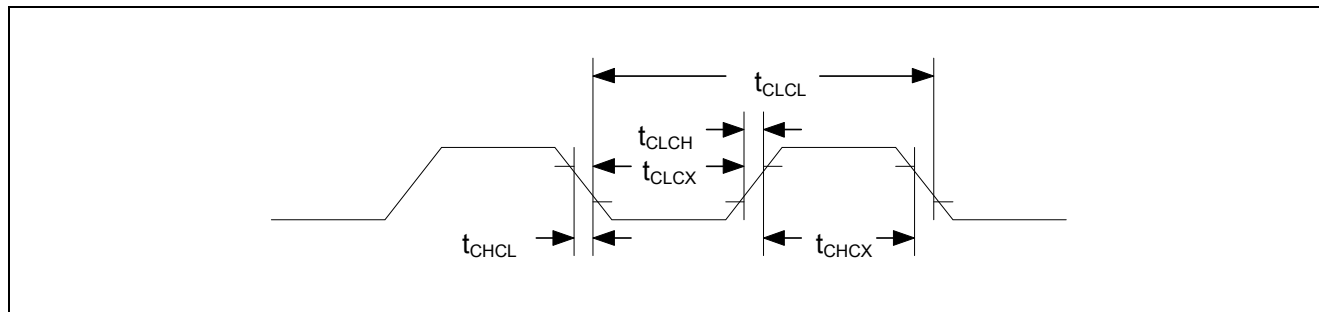
注: 1. t_{ADC}: ADCI输入时钟的周期

30 外部晶振和内部RC规格

30.1 4~24MHz XTAL 振荡器规格

参数	条件	最小值	典型值	最大最小值	单位
输入时钟频率	外部晶振	4		24	MHz

参数	符号	最小值	典型值	最大值	单位	注
振荡器频率	$1/t_{CLCL}$	0	24	MHz		
时钟高脉冲的时间	t_{CHCX}	20.8	-	-	nS	
时钟低脉冲的时间	t_{CLCX}	20.8	-	-	nS	
时钟上升沿的时间	t_{CLCH}	-	-	10	nS	
时钟下降沿的时间	t_{CHCL}	-	-	10	nS	



注: 占空比为50%.

30.2 内部RC Oscillator-22.1184MHz规格

参数	条件	最小值	典型值	最大值	单位
中心频率			22.1184		MHz
校准的内部振荡器频率	+25°C	-1		+1	%
	-40°C~+85°C	-5		+5	%

30.3 内部 RC Oscillator-10KHz的规格

参数	条件	最小值	典型值	最大值	单位
----	----	-----	-----	-----	----

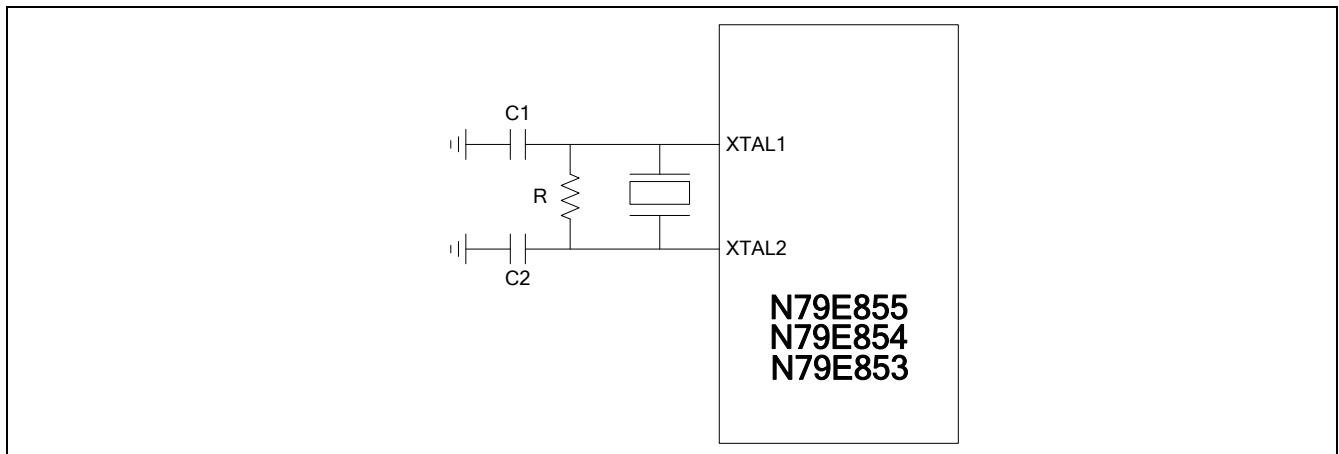


中心频率		5	10	15	KHz
------	--	---	----	----	-----

31 典型应用电路

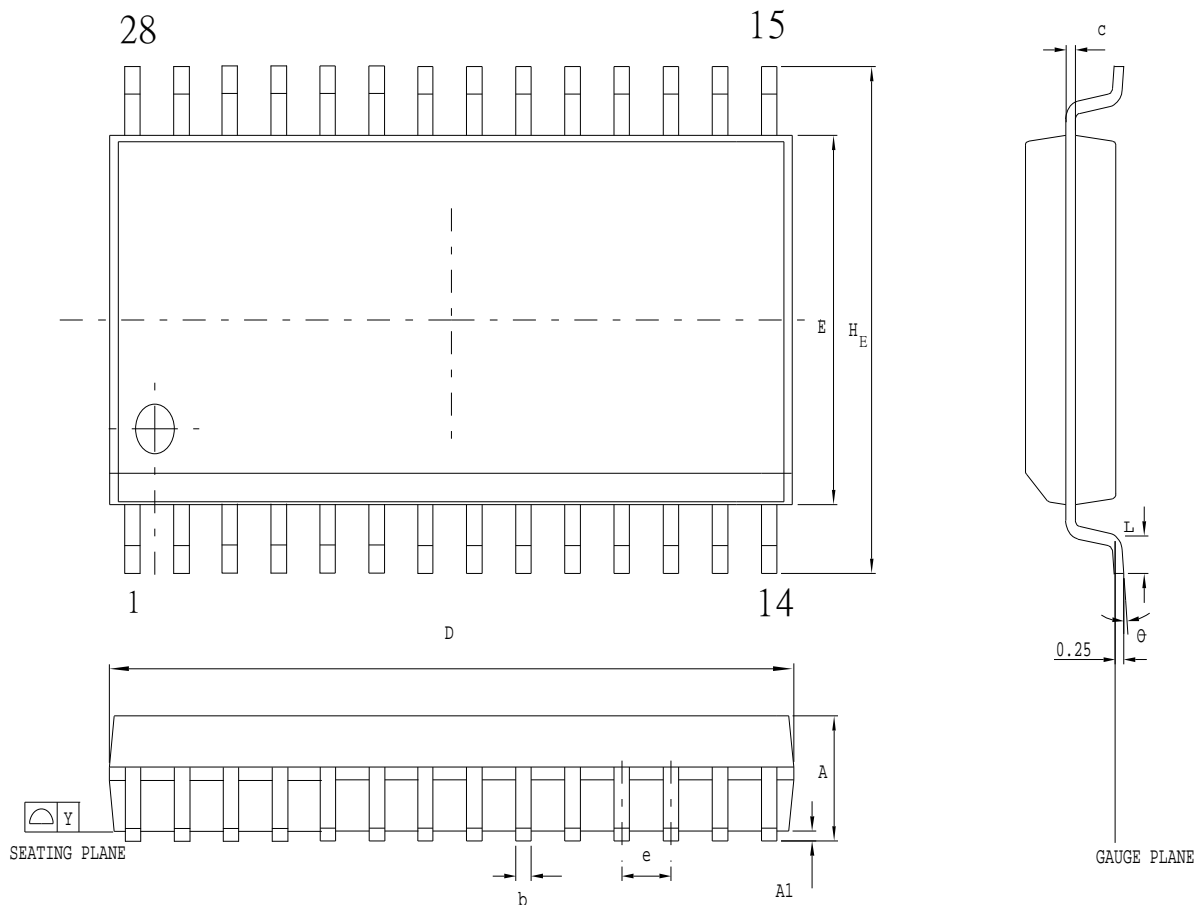
晶振	C1	C2	R
4MHz ~ 24 MHz	不需要	不需要	不需要

上表是晶振应用电路的参考值.



32 封装尺寸

32.1 28-pin SOP - 300 mil

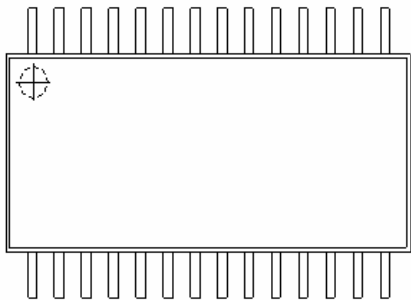
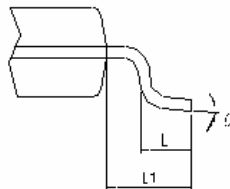
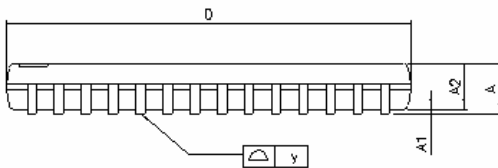
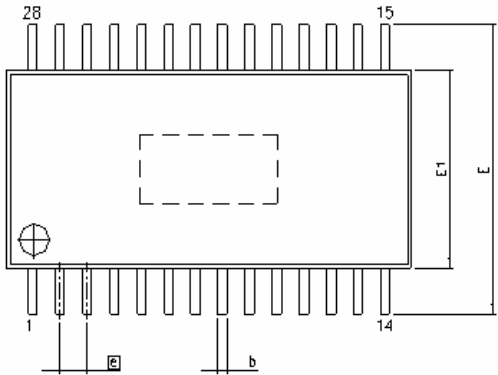


Control dimensions are in millimeter

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_E	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



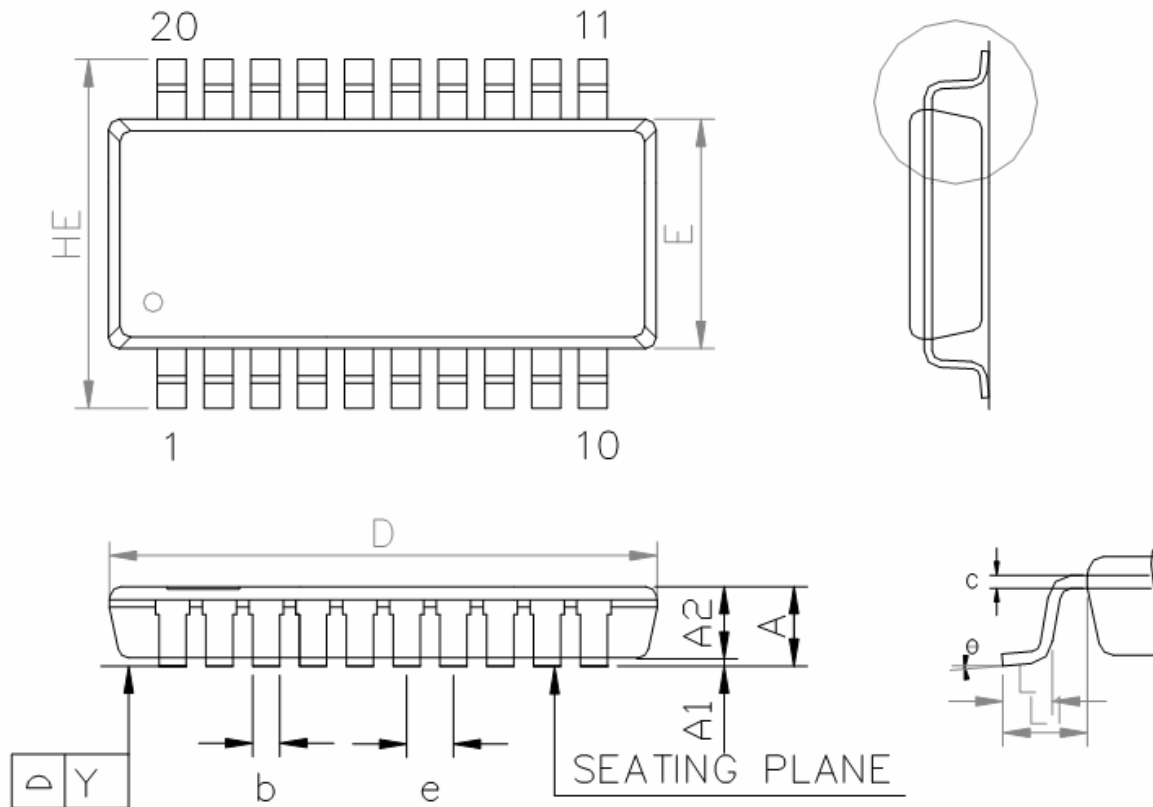
32.2 28-pin TSSOP - 4.4X9.7mm



Controlling Dimension :Millimeters

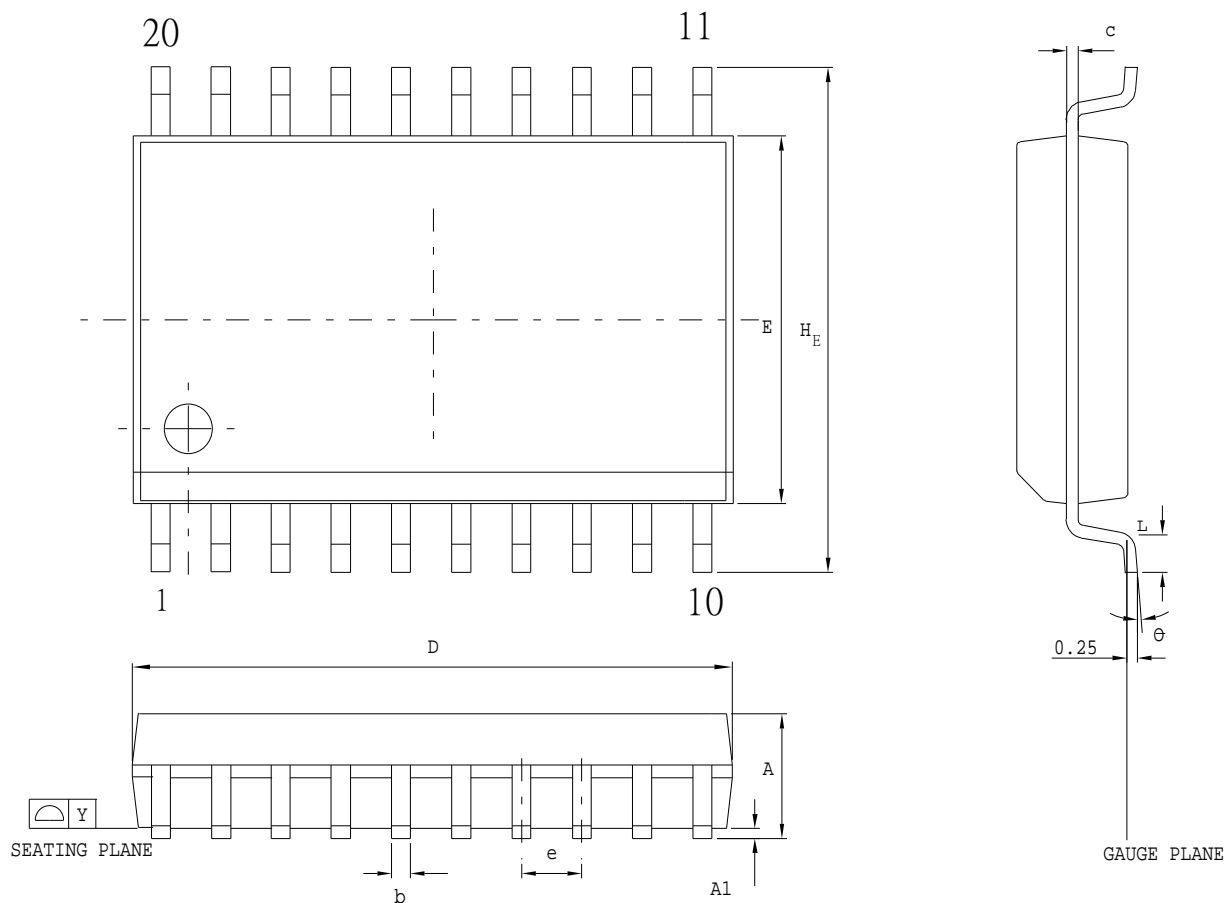
SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	-	-	1.20	-	-	0.0472
A1	0.00	-	0.15	0.00	-	0.0059
A2	0.80	1.00	1.05	0.0314	0.0393	0.4133
b	0.19	-	0.30	0.0074	-	0.0118
D	9.60	9.70	9.80	0.3779	0.3818	0.3858
E1	4.30	4.40	4.50	0.1692	0.1732	0.1771
E	6.40 BSC			0.2519 BSC		
Ⓜ	0.65 BSC			0.0255 BSC		
L1	1.00 REF			0.0393 REF		
L	0.45	0.60	0.75	0.0177	0.0236	0.0295
i/	0°	-	5°	0°	-	5°
y		0.05			0.00196	

32.3 20-pin TSSOP - 4.4X9.7mm



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
theta	0°	-	8°	0°	-	8°
Y	0.10 BASIC			0.004 BASIC		

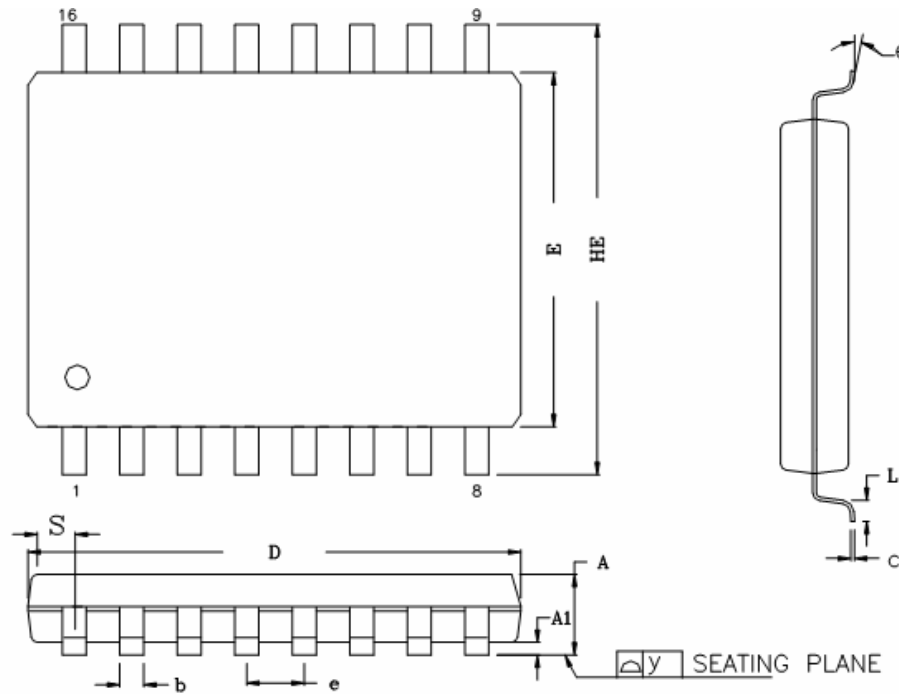
32.4 20-pin SOP - 300 mil



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	12.60	13.00	0.496	0.512
e	1.27 BSC		0.050 BSC	
H_E	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

32.5 16-pin SOP – 150 mil



CONTROL DIMENSIONS ARE IN MILLIMETERS.

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	1.35	—	1.75	0.053	—	0.069
A1	0.10	—	0.25	0.004	—	0.010
b	0.33	—	0.51	0.013	—	0.020
c	0.19	—	0.25	0.008	—	0.010
D	9.8	—	10.00	0.386	—	0.394
E	3.8	—	4.0	0.150	—	0.157
e	1.27 BASIC			0.050 BASIC		
HE	5.8	—	6.20	0.228	—	0.244
θ	0°	—	8°	0°	—	8°
L	0.40	—	1.27	0.016	—	0.050
S	0.394	—	0.648	0.0155	—	0.0255
y	—	—	0.10	—	—	0.004

33 版本历史

版本	日期	章节	描述
A1.0 SC1	2011.06.20		初次发行版本

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Further more, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur. Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.

*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.*