

N79A82X1规格书 *Rev.1.20*

选型指南

产品型号	存储器	NVM FLASH 容量	I/O (MIN/MAX)	接口	ADC	比较 器	PWM	封装
N79A8251D20	16KB/256	256B	18/16	UART/I ² C	4x10Bit	4CH	4x10Bit	DIP-20
N79A8251S20	16KB/256	256B	18/16	UART/I ² C	4x10Bit	4CH	4x10Bit	SOP-20
N79A8241D20	8KB/256	256B	18/16	UART/I ² C	4x10Bit	4CH	4x10Bit	DIP-20
N79A8241S20	8KB/256	256B	18/16	UART/I ² C	4x10Bit	4CH	4x10Bit	SOP-20
N79A8231D20	4KB/128	128B	18/16	UART/I ² C	4x10Bit	4CH	4x10Bit	DIP-20
N79A8231S20	4KB/128	128B	18/16	UART/I ² C	4x10Bit	4CH	4x10Bit	SOP-20
N79A8231D16	4KB/128	128B	14/12	UART/I ² C	3x10Bit	3CH	4x10Bit	DIP-16
N79A8231D14	4KB/128	128B	12/10	UART/I ² C	1x10Bit	2CH	4x10Bit	DIP-14
N79A8221D20	2KB/128	128B	18/16	UART/I ² C	4x10Bit	4CH	4x10Bit	DIP-20
N79A8221S20	2KB/128	128B	18/16	UART/I ² C	4x10Bit	4CH	4x10Bit	SOP-20

技术支持

立超电子科技有限公司

中国南京市和燕路251号金港大厦A幢2406室

ZIP:210028

Tel: 0086-25-83306839/83310926

Fax: 0086-25-83737785

Website: <http://www.dycmcu.com/>

目 录

1	概述	5
2	特性	5
3	产品型号信息	6
3.1	无铅(RoHS)产品型号信息表列	6
4	管脚配置	7
5	管脚描述	8
6	功能描述	9
6.1	片内 Flash EPROM	9
6.2	I/O端口	9
6.3	串行口(UART)	9
6.4	定时器	9
6.5	中断	9
6.6	数据指针	10
6.7	电源管理	10
6.8	CPU结构	10
6.8.1	ALU	10
6.8.2	累加器(ACC)	10
6.8.3	寄存器B	10
6.8.4	程序状态字寄存器(PSW)	11
6.8.5	片内便签RAM	11
6.8.6	堆栈指针	11
7	内存组织	12
7.1	程序内存	12
7.2	数据存储器	12
7.3	寄存器的映射	13
7.4	工作寄存器	16
7.5	位寻址区	16
7.6	堆栈	16
8	特殊功能寄存器	17
9	电源管理	18
9.1	空闲模式	18
9.2	掉电模式	18
10	复位条件	19
10.1	外部复位	19
10.2	上电复位 (POR)	19
10.3	看门狗定时器复位	19
10.4	复位状态	19

11	中断	22
11.1	中断源	22
11.2	中断响应时间	25
11.3	中断输入	25
12	可编程定时器/计数器	28
12.1	定时器/计数器0&1	28
12.2	时基选择	29
12.3	模式0	29
12.4	模式1	30
12.5	模式2	30
12.6	模式3	32
13	NVM数据存储器	33
14	看门狗定时器	36
14.1	看门狗控制	37
14.2	时钟控制看门狗	37
15	串行口 (UART)	40
15.1	模式0	40
15.2	模式1	41
15.3	模式2	42
15.4	模式3	44
15.5	帧错误检测	44
15.6	多机通信	45
16	时控访问保护	47
17	键盘中断 (KBI)	49
18	模拟比较器	50
19	I/O 端口配置	51
19.1	准双向端口模式配置	51
19.2	开漏端口模式配置	52
19.3	推挽端口模式配置	53
19.4	输入配置	53
20	振荡器	54
20.1	外部时钟输入选项	54
20.2	CPU时钟速度选择	54
21	电源监视功能	54
21.1	上电检测	54
21.2	欠压检测	54
22	脉宽调制(PWM)	56
23	模拟数字转换器	62
23.1	ADC的分辨率和仿真电源	64
24	I2C串行控制	68

24.1	SIO1端口.....	68
24.2	I2C控制寄存器:	68
24.2.1	地址寄存器, I2ADDR	68
24.2.2	资料寄存器, I2DAT	69
24.2.3	控制寄存器, I2CON	69
24.2.4	I2C时钟波特率位, I2CLK	70
24.2.5	状态寄存器, I2STATUS	71
24.3	I2C的操作模式	71
24.3.1	主传输模式	75
24.3.2	主传接收模式	76
24.3.3	从传输模式	77
24.3.4	从接收模式	78
24.3.5	GC模式	79
25	ICP(在电路编程) FLASH 编程	80
26	配置位	81
26.1	CONFIG1	81
26.2	CONFIG2	82
27	特殊功能寄存器细部列表描述	82
28	指令	111
28.1	指令时序	111
29	DC/ADC转换/模拟比较器电路电气特性	115
29.1	极限参数	115
29.2	DC 电气特性	115
29.3	ADC转换DC电气特性	116
29.4	模拟比较器电气特性	117
29.5	AC电气特性	117
29.6	外部时钟特性	117
29.7	AC规格	118
29.8	典型应用电路	118
30	封装尺寸	119
30.1	20-pin SOP	119
30.2	20-pin DIP	120
30.3	16-pin PDIP	121
30.4	14-pin PDIP	122
31	业务联络	124
32	免责声明	125

1 概述

N79A82X1系列是一个快速51微控制器，它有可以在系统编程的(ICP)应用程序Flash EPROM，可以使用烧写器在系统中编程。N79A82X1系列的指令系统完全与标准的8052指令系统兼容。16K/8K/4K/2K字节主Flash EPROM；256/128字节的NVM 数据Flash EPROM；256/128字节RAM；2个8位双向可位寻址的I/O端口；2个16-位定时器/计数器；4路10位AD转换器；4路10位PWM；2个串口包括1个I2C和1个增强型全双工串口。支持13个中断源4级中断；容易编程和校验，N79A82X1系列内部的FLASH EPROM程序内存支持电编程读取。一旦程序确定后，用户可以对代码进行保护。N79A82X1系列支持在电路仿真(ICE)功能，有一个JTAG接口可以连接到开发系统用于调试。

2 特性

- 全静态8位CMOS加速51微控制器，时钟频率可达20MHZ；VDD=4.5V~5.5V时钟频率可达20MHZ，VDD=2.7V~ 5.5V 时钟频率可达 12MHz。
- 16K/8K/4K/2K字节可以在系统编程的(ICP)应用程序Flash EPROM (AP Flash EPROM)
- 256/128字节片内RAM
- 256/128字节 NVM 数据FLASH EPROM，可以存储数据
- 指令与MCS-51兼容
- 集成JTAG接口，在芯片调试功能
- 2个8位双向可位寻址的I/O端口
- 2个16-位定时器/计数器
- 支持13个中断源，4级中断
- 一个全双工串口，并具有帧错误检测和地址自动识别功能
- 4个I/O口，可以选择为输出模式和TTL/史密斯触发器输入模式
- 可编程看门狗定时器
- 4路10位PWM
- 4路10位ADC
- 1个I2C通信接口(主/从)
- 8个按键中断输入口
- 两个模拟比较器
- 所有的端口引脚都有直接LED的能力 (20mA)
- 低电压检测，可以产生中断和复位
- 开发工具：
 - JTAG ICE(在线仿真)
 - ICP烧写(在线编程)
- 包装：
 - Lead Free (RoHS) DIP 20: N79A8251D20
 - Lead Free (RoHS) SOP 20: N79A8251S20
 - Lead Free (RoHS) DIP 20: N79A8241D20
 - Lead Free (RoHS) SOP 20: N79A8241S20
 - Lead Free (RoHS) DIP 20: N79A8231D20
 - Lead Free (RoHS) SOP 20: N79A8231S20
 - Lead Free (RoHS) DIP 20: N79A8221D20

— Lead Free (RoHS) SOP 20: N79A8221S20

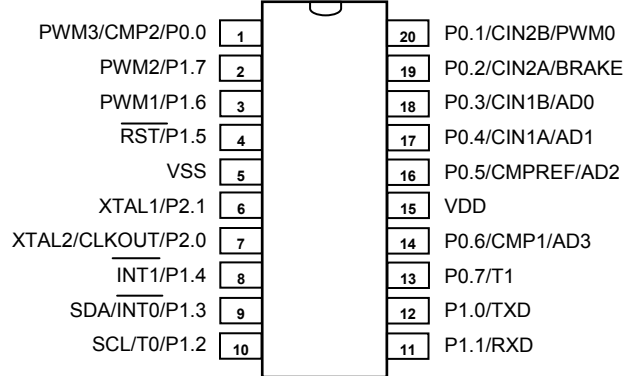
3 产品型号信息

3.1 无铅(RoHS)产品型号信息表列

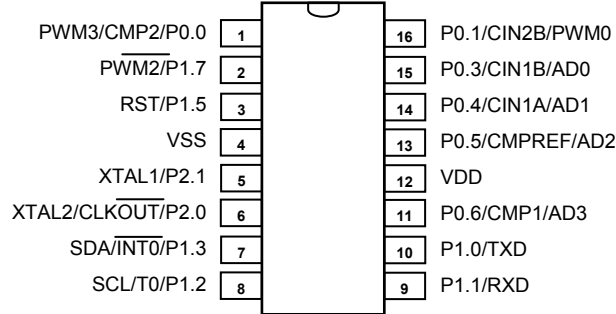
产品型号	EPROM FLASH 容量	RAM	NVM FLASH 容量	ADC	PWM	封装	REMARK
N79A8251D20	16KB	256B	256B	4x10Bit	4x10Bit	DIP-20 Pin	
N79A8251S20	16KB	256B	256B	4x10Bit	4x10Bit	SOP-20 Pin	
N79A8241D20	8KB	256B	256B	4x10Bit	4x10Bit	DIP-20 Pin	
N79A8241S20	8KB	256B	256B	4x10Bit	4x10Bit	SOP-20 Pin	
N79A8231D20	4KB	128B	128B	4x10Bit	4x10Bit	DIP-20 Pin	
N79A8231S20	4KB	128B	128B	4x10Bit	4x10Bit	SOP-20 Pin	
N79A8231D16	4KB	128B	128B	4x10Bit	3x10Bit	DIP-16 Pin	
N79A8231D14	4KB	128B	128B	4x10Bit	1x10Bit	DIP-14 Pin	
N79A8221D20	2KB	128B	128B	4x10Bit	4x10Bit	DIP-20 Pin	
N79A8221S20	2KB	128B	128B	4x10Bit	4x10Bit	SOP-20 Pin	

4 管脚配置

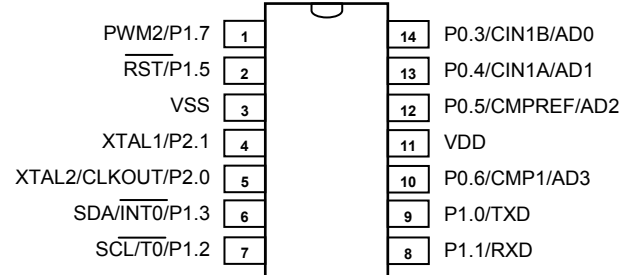
20PIN-DIP20/SOP20



16PIN-DIP16



14PIN-DIP14



备注：仅N79A8231提供DIP16/DIP14封装。

5 管脚描述

标识符	类型	描 述
$\overline{\text{RST}}$ (P1.5)	I	复位：当振荡器工作的情况下，2个机器周期的低电平复位系统。
XTAL1	I	晶振1: 晶体振荡器输入，可以由外部时钟驱动。
XTAL2	O	晶振2: 晶体振荡器输出，XTAL1的反相。
VSS	P	地
VDD	P	电源
P0.0–P0.7	I/O	端口0：4种输出模式、2种输入模式，P0.3~P0.6 可以作为ADC模拟信号输入脚。
P1.0–P1.7	I/O	端口1：4种输出模式、2种输入模式，P1.2(SCL)和P1.3(SDA)只能是开漏脚，P1.5只能作为输入脚。

注：类型 P: 电源，I: 输入，O: 输出，I/O: 双向输入输出

6 功能描述

N79A82X1系列是由4分频8051内核和寄存器、16K/8K/4K/2K字节Flash EPROM, 256/128字节RAM, 256/128字节NVM 数据FLASH EPROM, 2个8位双向可位寻址的I/O端口; 2个16-位定时器/计数器; 4路10位AD转换器; 4路10位PWM; 1个I²C和1个全双工串口。内部的FLASH EPROM程序内存可以由通用烧写器或ICP烧写器烧写。

6.1 片内 Flash EPROM

N79A82X1系列内部有16K/8K/4K/2K字节的应用程序存储空间, 当使用在电路编程时需要专门的ICP烧写器烧写内部的16K/8K/4K/2K字节Flash EPROM。ICP特性使经常更换/升级软件变得非常容易和高效, 同时也是终端客户在不需要拿掉IC甚至不需要打开机壳就可以方便的更新应用程序。

6.2 I/O端口

N79A82X1系列有2个8位双向I/O口, 复位脚也可用为I/O, 最多可达16个I/O引脚。通过设置PxM1.y和PxM2.寄存器, 所有端口可以备用为4种输出模式, 推挽模式和高阻输入这两种模式下不需要任何外部的上拉, 另外也可以作为普通的I/O口或开漏I/O口。所有的端口都可以作为双向I/O端。这些端口是强下拉弱上拉。

6.3 串行口(UART)

N79A82X1系列有一个增强型串行口, 它的功能与标准8052串行口相似。N79A82X1系列的串行口能以不同的方式运行, 以获得时序相似。串行口有自动地址识别和帧错误检测的增强功能

6.4 定时器

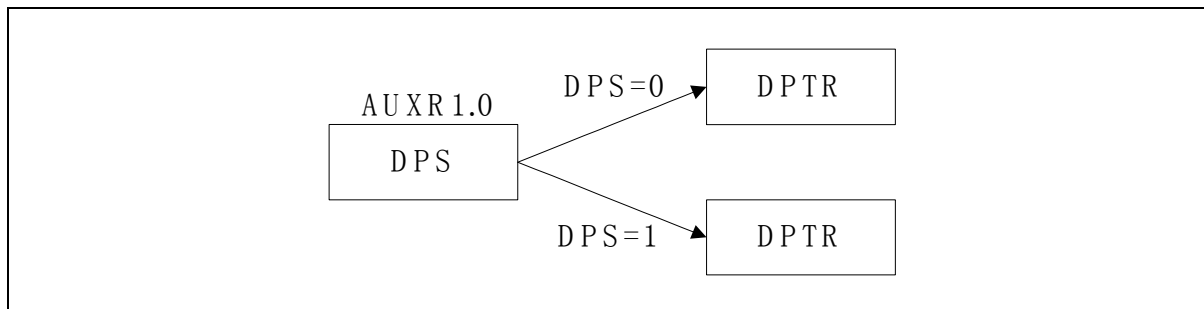
N79A82X1系列有2个16位定时器, 其功能与8052体系中的定时器类似。当作为定时器使用时, 可将它们设置为每4个时钟周期进行一次计数, 或者每12个时钟周期进行一次计数。这位用户提供了模拟8052时钟运行的一种方式。

6.5 中断

N79A82X1系列的中断系统与标准8052之中断系统有细微的差别。由于存在新增功能和外设, 中断源的数量和中断向量都相应得增加。

6.6 数据指针

和标准8052一样N79A82X1系列MCU有两个16位的数据指针(DPTR)。有AUXR1的DPS位设置。如下表所示：



6.7 电源管理

电源管理与标准的8052相似，N79A82X1系列有空闲和掉电模式。在空闲模式下，CPU时钟停止但定时器，串行口和中断时钟不会停止。在掉电模式下，所有的时钟都停止，此时功耗最低。

6.8 CPU结构

N79A82X1系列是基于标准的8052内核，在8-位的ALU周围集成了用于临时存储数据和控制外设的内部寄存器。N79A82X1系列可以执行标准8052的指令集。

6.8.1 ALU

ALU N79A82X1系列MCU的核心，它有算术运算和逻辑运算功能，它还具有判断和程序转移功能。但客户不可以直接使用ALU，指令经过译码器译码后经过ALU和它的辅助寄存器产生的运算结果。ALU主要通过特殊寄存器ACC和特殊寄存器B实现乘除法运算。ALU产生几种状态标志，这些标志存放于状态字寄存器(PSW)中。

6.8.2 累加器(ACC)

在N79A82X1系列的MCU中算术运算、逻辑运算、数据传送的操作中，累加器 (ACC)是一个非常重要的寄存器。CPU直接访问累加器，所以高速指令会使用累加器作为第一参数。

6.8.3 寄存器B

通用寄存器B是一个8位寄存器，在乘/除法运算中存放第二参数。在其它指令中通用寄存器B可以作为通用寄存器使用

6.8.4 程序状态字寄存器(PSW)

PSW是一个8位标志寄存器，存放ALU的计算结果。包含：进位标志位、辅助进位标志位、用户标志位、寄存器工作组选择位、溢出标志和奇偶标志。

6.8.5 片内便签RAM

N79A82X1系列有256/128字节片内便签RAM。在程序的执行中可以临时存放数据，有一个可位寻址区域，可以直接使用。

6.8.6 堆栈指针

N79A82X1系列有一个8-位堆栈指针，它指向堆栈的顶端。堆栈在便签RAM区，因此堆栈的大小由此部分RAM大小决定的。

7 内存组织

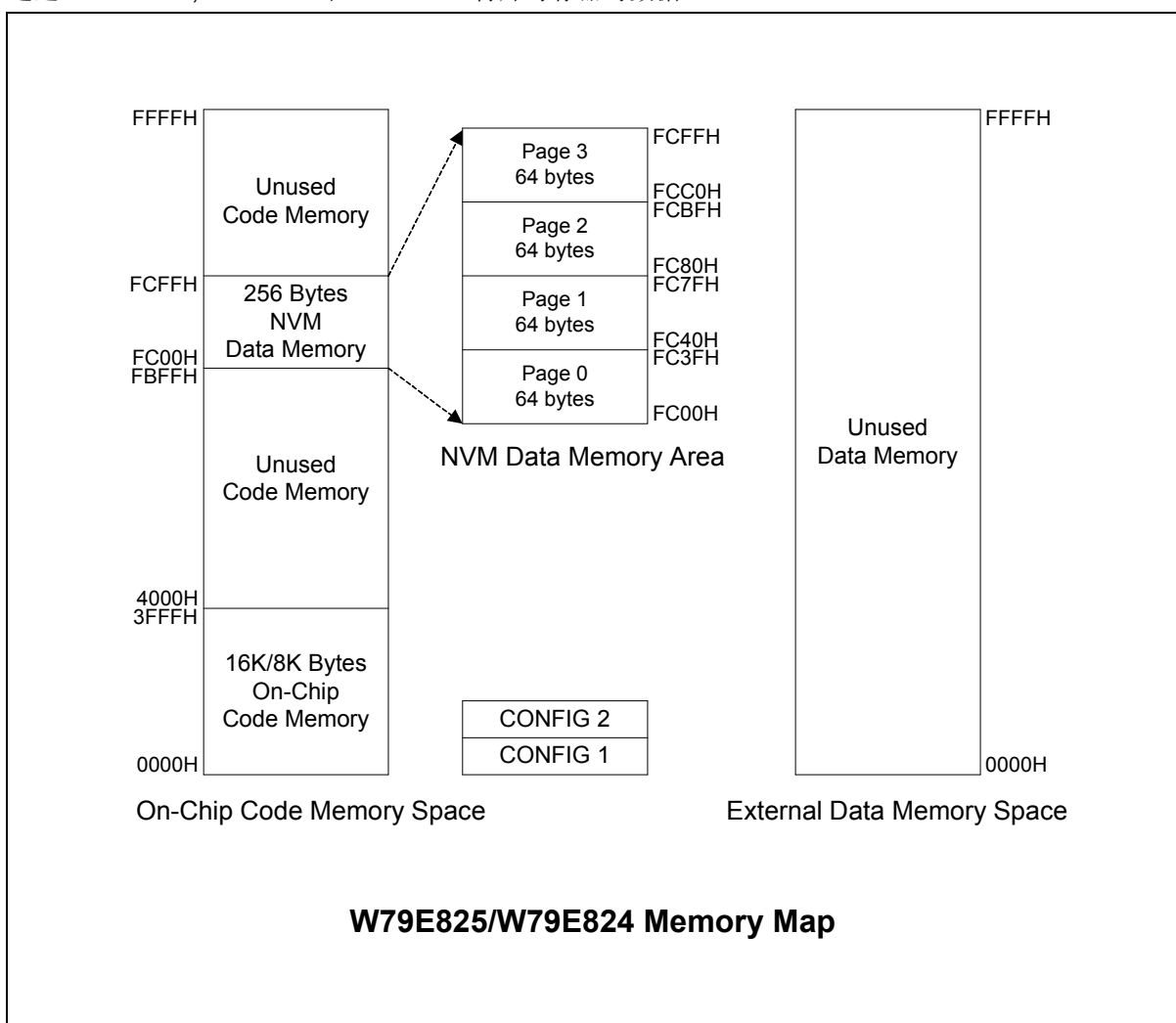
N79A82X1系列将内存分为2个独立的区域：程序内存区和数据存储器区。程序内存区用来存放程序代码，数据存储器区用来存放数据及内存映像的设备

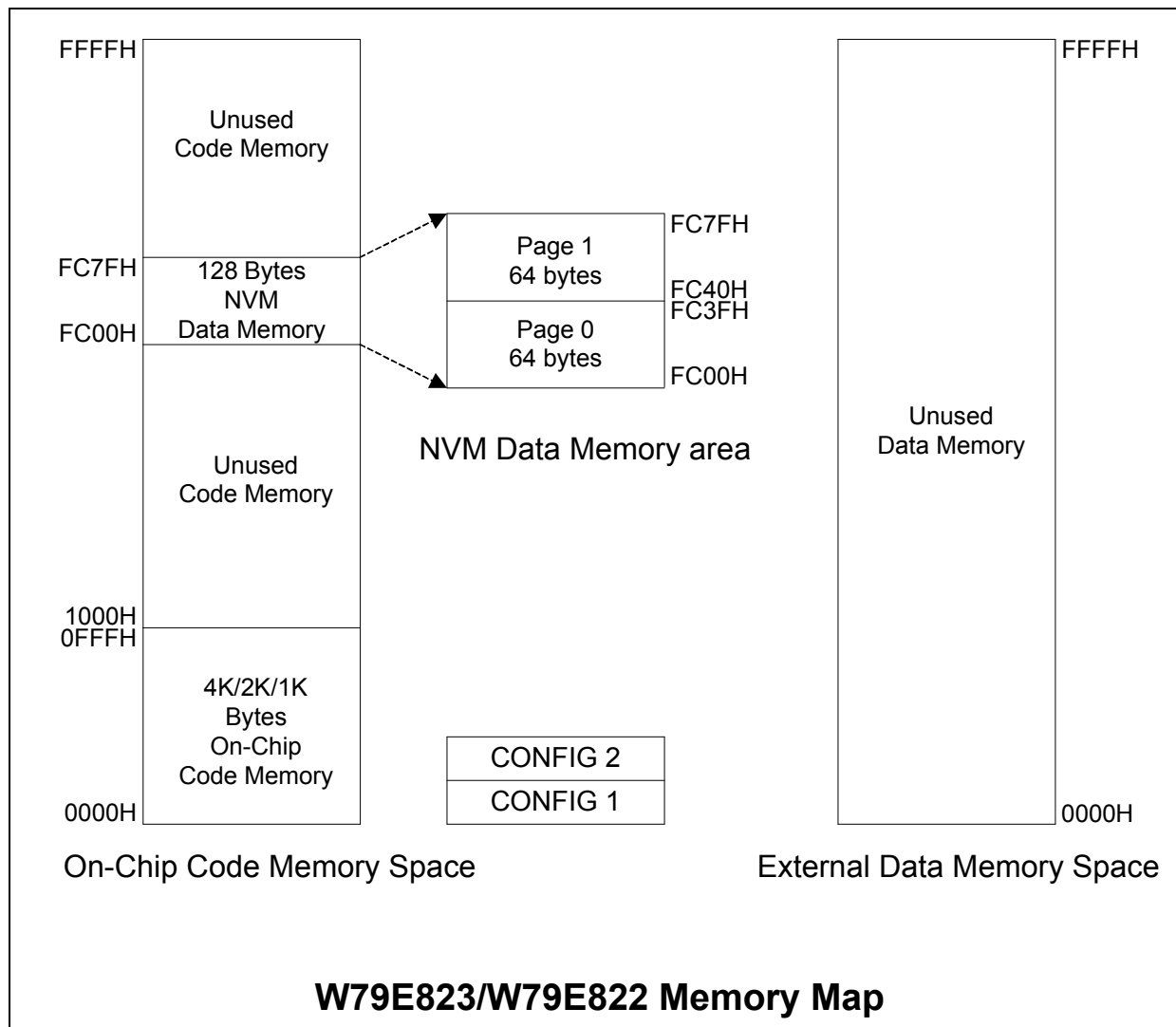
7.1 程序内存

N79A82X1系列最大有16K/8K/4K/2K字节的程序内存，所有指令都从这些区域中取出执行。MOVC指令同样也访问这些区域。

7.2 数据存储器

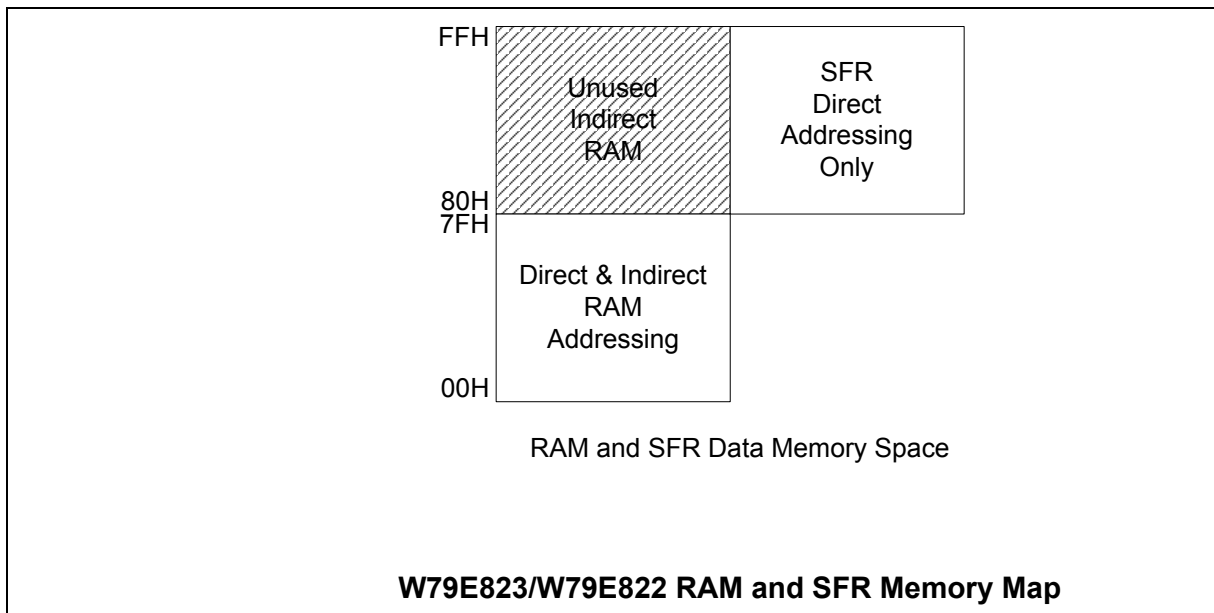
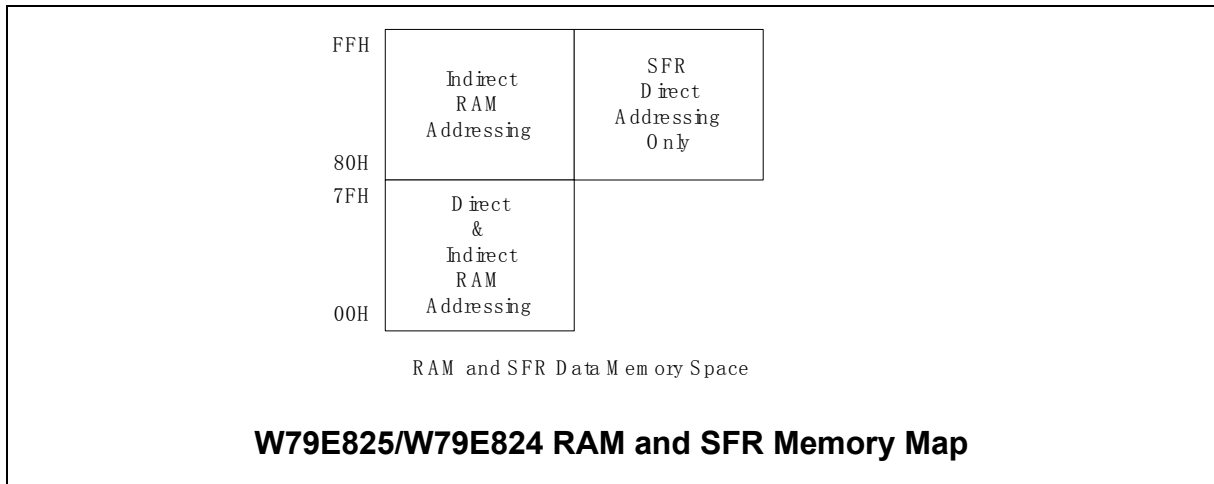
N79A82X1系列中有256/128位组长度的NVM数据存储器。读该部分的内容使用“MOVC A,@A+DPTR”；通过NVMADDR, NVMDAT 和NVMCON 特殊寄存器写数据。





7.3 寄存器的映射

N79A82X1系列有独立的程序存储空间和数据存储空间。片内256/128字节便签RAM不属于外部内存，它包含有特殊功能寄存器(特殊功能寄存器)。SFR只能用直接寻址方式访问其它的片内RAM可以直接寻址也可以间接寻址访问。



便签RAM只有256/128字节，因此仅适用于数据量较小的场合；在使用的时候，注意不要超出范围。描述如下：

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

7.4 工作寄存器

工作寄存器有四组，每组有8个8-位寄存器。组号标识为：第1组、第2组、第3组、第4组，在组中寄存器可以直接访问。寄存器名称分别为：R0、R1、R2、R3、R4、R5、R6和R7，他们可以指向任何一组，有PSW寄存器中的RS0、RS1的状态决定。R0和R1寄存器被用作间接寻址的地址。

7.5 位寻址区

便签RAM区从20h到2Fh的区域可以字节寻址也可以位寻址，也就是说在这个区域可以按位寻址，指令译码器会自动分辨位指令还是字节指令。在特殊功能寄存器中地址是以0或8结尾的都可以位寻址。

7.6 堆栈

便签RAM可以用作堆栈，该区域由堆栈指针(SP)指定，SP是堆栈的顶端地址。当跳转、调用或中断调用时返回地址放在栈顶，在RAM中堆栈的起始地址是没有限定的，复位后堆栈指针默认是07h，使用者可以根据需求改变堆栈的起始地址。SP指向堆栈里最后的那个值，进站后SP加1，出栈是读出栈定的值然后SP会减1。

8 特殊功能寄存器

N79A82X1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。N79A82X1系列中含有标准8052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，8052中未被定义的位被赋予了新的功能。下表列出了N79A82X1系列中的特殊功能寄存器。

表1：特殊功能寄存器列表

F8	IP1							
F0	B						P0IDS	IP1H
E8	IE1							
E0	ACC	ADCCON	ADCH					
D8	WDCON	PWMPL	PWM0L	PWM1L	PWMCON1	PWM2L	PWM3L	PWMCON2
D0	PSW	PWMPH	PWM0H	PWM1H		PWM2H	PWM3H	PWMCON3
C8							NVMCON	NVMDAT
C0	I2CON	I2ADDR					NVMADDR	TA
B8	IP0	SADEN			I2DATA	I2STATUS	I2CLK	I2TIMER
B0		P0M1	P0M2	P1M1	P1M2	P2M1	-	IP0H
A8	IE	SADDR			CMP1	CMP2		
A0	-	KBI	AUXR1					
98	SCON	SBUF						
90	P1					DIVM		
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	
80	P0	SP	DPL	DPH				PCON

注释：

1、有加粗边框的那一列为可位寻址的特殊功能寄存器

2、上表列出了N79A82X1系列中的特殊功能寄存器，每行分了8列。空白项表示该地址空间没有特殊功能寄存器存在，对这些空间的访问将会得到全1的结果。

9 电源管理

N79A82X1系列有若干节电选项来帮助用户减少电源消耗，节电模式为掉电模式、空闲模式。

9.1 空闲模式

用户通过将1写入PCON.0，使系统进入空闲模式。把系统放入空闲模式的指令是系统在进入空闲模式前执行的最后一条指令。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。ALE和PSEN在空闲模式下处于高电平状态。各个端口维持进入空闲模式前的逻辑状态。有2种方式可以让系统从空闲模式中退出。由于中断控制器依旧在工作，因此任何使能的中断都可以让系统退出空闲模式。当这样的中断发生时，系统将自动清除空闲位，退出空闲模式并转向相应的中断服务程序。在中断服务程序完成后，系统将在使系统进入空闲模式的那条指令之后继续程序的运行。复位同样可以使系统退出空闲模式。实现复位的方式有在RST脚上输入高电平，上电复位以及看门狗定时器复位。外部复位时，高电平至少要维持2个机器周期（8个时钟周期），以便系统识别外部复位信号。复位后程序指针数值为0000H，所有SFR都回到初始状态。由于时钟并没有停止工作因此程序会被立即执行。在空闲模式下，看门狗定时器依旧工作，因此如果看门狗定时器中断打开，看门狗定时器溢出后会产生中断使系统退出空闲模式。软件必须复位看门狗定时器，以便在看门狗定时器溢出并经过512个时钟周期后将系统复位。当N79A82X1系列以复位的方式从空闲模式中退出后，系统将从头开始执行指令。

电源控制寄存器PCON

地址: 87h

Bit:	7	6	5	4	3	2	1	0
	SMOD	SMOD0	BOF	POR	GF1	GF0	PD	IDL
位	名称	功能						
1	PD	1: 系统进入掉电模式；该模式下，所有时钟停止工作，程序也不再执行。						
0	IDL	1: 系统进入空闲模式；该模式下，CPU的时钟停止工作，程序停止运行；但串口、定时器、中断的时钟没有停止，这些功能模块仍正常运行。						

9.2 掉电模式

用户通过将1写入PCON.1，使系统进入掉电模式。把系统放入掉电模式的指令是系统在进入掉电模式前最后执行的一条指令。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。

复位以及电平跳变出发的中断可以使系统退出掉电模式。外部复位可让系统退出中断，RST脚上的高电平将终止掉电模式，然后重新开启时钟。程序将从0000H处开始执行，由于在掉电模式中时钟停止工作，因此看门狗定时器不能提供复位功能让系统退出掉电模式。

如果EA=1，外部中断被设置为电平触发方式而且相应的外部中断开放，那么外部中断输入脚上的低电平将迫使系统退出掉电模式。如果上面所述的条件满足，当外部中断输入脚上有低电平信号时，该信号将重新启动时钟。设备转向相应的中断服务程序，在ISR服务完成后，系统将从使系统进入掉电模式的那条指令之后继续程序的运行。

10 复位条件

用户有很多与硬件相关的选项来将N79A82X1系列复位。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2种方法可以将系统复位：1.外部复位信号；2.看门狗定时器复位。

10.1 外部复位

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保持2个机器周期，以保证系统检测到有效的RST低电平。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脚上电平一直为低，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种复位状态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。

10.2 上电复位 (POR)

软件在读取POR位以后必须将其清除，否则将会影响到将来对复位状态的判断。如果发生掉电的情况（VDD低于Vrst），那么系统将会回到复位状态。当电源恢复正常，系统会再进行一次上电复位延迟并设置POR标志位。

10.3 看门狗定时器复位

看门狗定时器是一个带可编程溢出时间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，那么在其溢出（未被清零）且经过512个时钟后看门狗定时器会产生一个复位信号。这样会使系统进入复位状态。这个状态由硬件维持2个机器周期。一旦退出复位状态，系统将从0000H处执行代码。

10.4 复位状态

大多数SFR在复位后回到其初始状态。程序计数器被设为0000H，而且只要复位状态一直保持，它也将维持0000H的数值不变。但是复位不影响片上RAM的状态。RAM中的资料在复位期间维持不变。但是堆栈指针变为07H，因此堆栈的数据会丢失。如果VDD低于2V（维持RAM中数据所需的最小电压），那么RAM中的数据就会丢失。因此第一次上电复位后RAM中的数据不确定，而当电源电压跌至2V以下后，RAM中资料丢失。

复位后大多数SFR被清除，中断和定时器被关闭。如果复位源是上电复位，那么看门狗定时器也被关闭。端口特殊寄存器中的值是FF，所以端口上将输出全高电平。由于没有片内上拉，P0口的状态是浮空的。

特殊功能寄存器复位状态

特殊功能寄存器名称	复位 VALUE	特殊功能寄存器名称	复位 VALUE
P0	11111111B	I2DAT	xxxxxxxB
SP	0000111B	I2STATUS	00000xxxB
DPL	00000000B	I2TIMER	00000000B
DPH	00000000B	I2CLK	00000000B
PCON	00xx0000B	I2CON	00000000B
TCON	00000000B	I2ADDR	xxxxxxxB
TMOD	00000000B	TA	00000000B
TL0	00000000B	PSW	00000000B
TL1	00000000B	PWMPH	xxxxxx00B
TH0	00000000B	PWM0H	xxxxxx00B
TH1	00000000B	PWM1H	xxxxxx00B
CKCON	00000000B	PWM2H	xxxxxx00B
P1	1111xx11B	PWM3H	xxxxxx00B
DIVM	00000000B	WDCON	0x000000B
SCON	00000000B	PWMPL	00000000B
SBUF	xxxxxxxB	PWM0L	00000000B
-	-	PWM1L	00000000B
KBI	00000000B	PWMCON1	00000000B
AUXR1	00000000B	PWM2L	00000000B
IE	00000000B	PWM3L	00000000B
SADDR	00000000B	PWMCON2	00000000B
CMP1	00000000B	PWMCON3	xxxxxxx0B
CMP2	00000000B	ACC	00000000B
P0M1	00000000B	ADCCON	xx000x00B
P0M2	00000000B	ADCH	xxxxxxxB
P1M1	00000000B	EIE	xx000000B
P1M2	00000000B	B	00000000B
P2M1	00000000B	P0IDS	00000000B
-	-	IPH	xx000000B
IP0H	x0000000B	IP1	xx000000B
IP0	x0000000B	NVMADDR	0000 0000B
SADEN	00000000B	NVMDAT	0000 0000B
		NVMCON	00xx xxxxB

表10-1: SFR 复位值

WDCON 中的位按照不同的复位类型进行置位/清0

	外部复位	看门狗定时器复位	上电复位
WDCON	0x0x0xx0b	0x0x01x0b	01000000b

POR (WDCON.6) 在上电复位后置位。当电源跌落时, PFI位WDCON.4被置'1', 上电复位后被清'0'; WTRF (WDCON.2) 在看门狗定时器复位后置'1', 上电复位后被清'0'。 EWT (WDCON.1) 也在上电复位时清除, 这样就将看门狗定时器复位关闭, 看门狗定时器或外部复位不会影响该位。

11 中断

N79A82X1系列的中断分4个优先级13个中断源。每个中断源都有相应的优先级设置位，标志位中断向量及使能位。另外系统可以关闭或打开所有中断。

11.1 中断源

外部中断 $\overline{\text{INT0}}$ 和 $\overline{\text{INT1}}$ 按照IT0和IT1的设置可以是边沿触发或是电平触发。TCON中的IE0和IE1位是外部中断的标志位，检测这2位的状况可以知道是否产生了外部中断。在边沿触发模式中，系统在每个机器周期都要采样INTx脚。如果在一个周期里采样到高电平在下一个周期里采样到低电平，那么系统就检测到了一个高电平到低电平的跳变，此时相应的IEx位置位，同时向系统申请中断服务。由于系统在每个机器周期都要对外部中断进行采样，因此外部中断输入脚上的高电平或低电平至少要维持一个机器周期。当系统响应中断执行中断服务程序时，lex位被自动清除。如果选择电平触发方式，那么中断请求源的低电平信号必须保持到系统响应该中断。在进入中断服务程序时，lex位不会被硬件清零。如果外部中断输入脚上的电平在中断服务程序完成后依然保持，系统会立即识别该中断再次进入同样的中断服务程序。

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。

看门狗定时器可以用作系统监控器或是一个简单的定时器。无论以何种方式工作，当定时器超时时。看门狗定时器中断标志WDIF（WDCON.3）会置位，如果EIE.4=1，那么这时会产生一个中断。

当串口的两个中断源接受或发送，发生中断时特殊功能寄存器SCON的RI和TI被值'1'；该位不能自动清'0'，用户必须软件清'0'。

所有中断产生标志均可由硬件置位/复位，同样若软件将这些位置位也可以引发中断。各个中断可以由IE寄存器中的相应位来打开或关闭。IE中有一个中断总控制位，可以打开或关闭所有的中断。

ADC有一个中断源，当ADC转换结束后会产生ADC中断，标志位是特殊功能寄存器ADCCON中的ADCI位；该位不能自动清'0'，用户必须软件清'0'。

两个比较器可以产生中断，标志位是CMF1和CMF2。该位不能自动清'0'，用户必须软件清'0'。

I2C在完成一个动作后可以产生中断，标志位是SI；SI由硬件置'1'，如果I2C中断允许，将产生I2C中断。该位不能自动清'0'，用户必须软件清'0'。

PWM功能可以产生中断，标志位是BKF，在外部钳制脚发生钳制时会产生中断请求。该位不能自动清'0'，用户必须软件清'0'。优先级结构

对中断来说，系统为其提供3种优先级：最高、高、低和最低。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。结构的具体方式见下表。各个中断按照其中断优先级顺序编号。

中断优先权结构

中断源	标志	优先级
外部中断0	IE0	1(最高)
欠压检测	BOF	2
看门狗定时器	WDIF	3
定时器0溢出	TF0	4
I2C中断	SI	5
ADC中断	ADCI	6
外部中断1	IE1	7
KBI中断	KBF	8
比较器1中断	CMF1	9
定时器1溢出	TF1	10
比较器2中断	CMF2	11
串行口	RI + TI	12
PWM	BKF	13 (最低)

每个机器周期都检测中断标志和中断优先权。如果满足特定条件硬件将执行内部产生的LCALL 指令，目标地址是中断向量地址。产生LCALL的条件是：

1. 较低优先级的中断不会打断同等优先级的中断和较高优先级的中断服务程序
2. 在正在执行指令的最后一个周期检测中断标志
3. 正在执行的指令不包括写IE、IE1、IP0、IP0H、IP1或IPH1寄存器的指令并且不是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。看门狗定时器中断标志 WDIF必须有软件清除。硬件执行一个长调指令。该指令保存程序计数器的内容到堆栈，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。中断源的向量地址分配如下：

中断源向量地址

中断源	向量地址	中断源	向量地址
外部中断0	0003h	定时器0溢出	000Bh
外部中断1	0013h	定时器1溢出	001Bh
串行口	0023h	欠压中断	002Bh
I2C中断	0033h	KBI中断	003Bh
比较器2中断	0043h	-	004Bh
看门狗定时器	0053h	ADC 中断	005Bh
比较器1中断	0063h	-	006Bh
PWM钳制中断	0073h	-	007Bh

四级中断优先权

优先权位		中断优先级
IPXH	IPX	
0	0	第0级(最低优先权)
0	1	第1级
1	0	第2级
1	1	第3级(最高优先权)

中断向量表不是连续的空间，保留的中断向量可用于今后产品的扩充时使用。

程序从向量地址连续执行到**RETI** 指令。执行**RETI**指令处理器将从栈顶弹出数据并装载到**PC**指针。用户必须注意在进入中断后堆栈存放的内容，如果执行中断返回操作，栈顶的内容已经改变**CPU**不会知道堆栈的内容已经改变掉，而是按正常情况将栈顶的数据装入**PC**指针，这样将会引起错误发生。

N79A82X1系列有4个优先中断级结构。这样使**N79A82X1**系列控制更多的中断源有极大的灵活性，**N79A82X1**系列支持多达13个中断源。

每个中断源可以透过**IEN0**或**IEN1**单独的允许或禁止设置。**IEN0**寄存器包含一个全局的中断禁止位**EA**，该位可以一次性禁止所有中断。

每一个中断源都可以单独的设为4个中断级别之一，设置方法是通过设置或清除**IP0**, **IP0H**, **IP1**和**IP1H**寄存器的相应位。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中；最高优先权中断不会被任何中断打断；故若同时有2个中断请求，较高优先级的中断先执行服务程序。

若具有同等优先级中断同时请求中断，由内部有一个监测顺序来决定执行中断服务程序的顺序。

如下表所示的内容是：中断源、标志位、向量地址、允许位、优先权位、仲裁序列，并且描述了那一个唤醒**CPU** 的掉电模式。

描述	中断标志位	向量地址	中断允许位	中断优先权	仲裁序列	唤醒掉电
外部中断 0	IE0	0003H	EX0 (IE0.0)	IP0H.0, IP0.0	1(最高)	是
欠压检测	BOF	002BH	EBO (IE.5)	IP0H.5, IP0.5	2	是
看门狗定时器	WDIF	0053H	EWDI (IE1.4)	IP1H.4, IP1.4	3	是
定时器0中断	TF0	000BH	ET0 (IE.1)	IP0H.1, IP0.1	4	否
I2C中断	SI	0033H	EI2 (IE1.0)	IP1H.0, IP1.0	5	否
ADC转换器	ADCI	005BH	EAD (IE.6)	IP0H.6, IP0.6	6	是
外部中断1	IE1	0013H	EX1 (IE.2)	IP0H.2, IP0.2	7	是
KBI中断	KBF	003BH	EKB (IE1.1)	IP1H.1, IP1.1	8	是
比较器1中断	CMF1	0063H	EC1 (IE1.2)	IP1H.2, IP1.2	9	是
定时器1中断	TF1	001BH	ET1 (IE.3)	IP0H.3, IP0.3	10	否
比较器2中断	CMF2	0043H	EC2 (IE1.3)	IP1H.3, IP1.3	11	是
串行口Tx和Rx	TI & RI	0023H	ES (IE.4)	IP0H.4, IP0.4	12	否
PWM中断	BKF	0073H	EPWM (IE1.5)	IP1H.5, IP1.5	13 (最低)	否

注：当使用内部RC作为主振时，看门狗定时器和ADC转换器可以将芯片由掉电模式唤醒。

11.2 中断响应时间

每一个中断源的响应时间取决于几个方面，如中断自身特点和指令的执行。外部中断 $\overline{\text{INT0}}$ 和RI+TI在机器周期的C3采样并且他们相应的中断标志lex自动的置位或清除。定时器0和1溢出标志在机器周期的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N79A82X1系列执行写IE, IE1, IP0, IP0H, IP1或IP1H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, IE1, IP0, IP0H, IP1或IP1H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。

11.3 中断输入

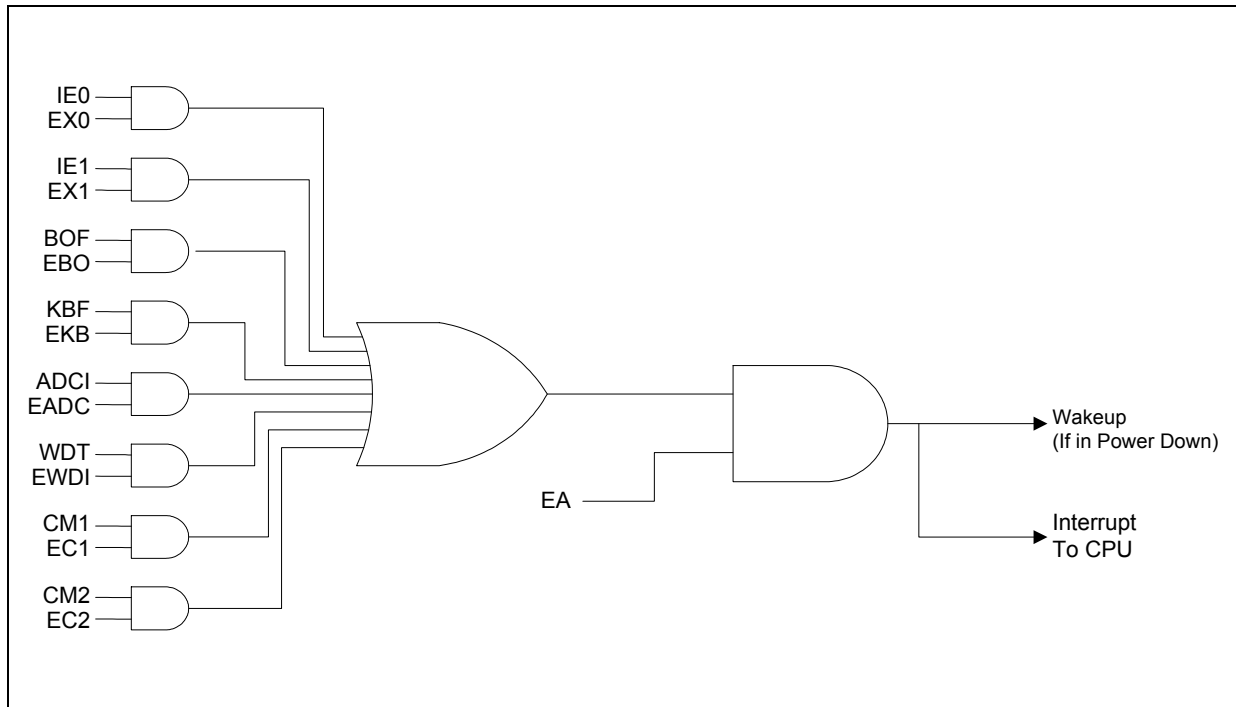
N79A82X1系列有13个中断源和两个独立的中断源输入，一个是：IE0,IE1, BOF, KBF, WDT, ADC,

N79E8251/8241/8231/8221

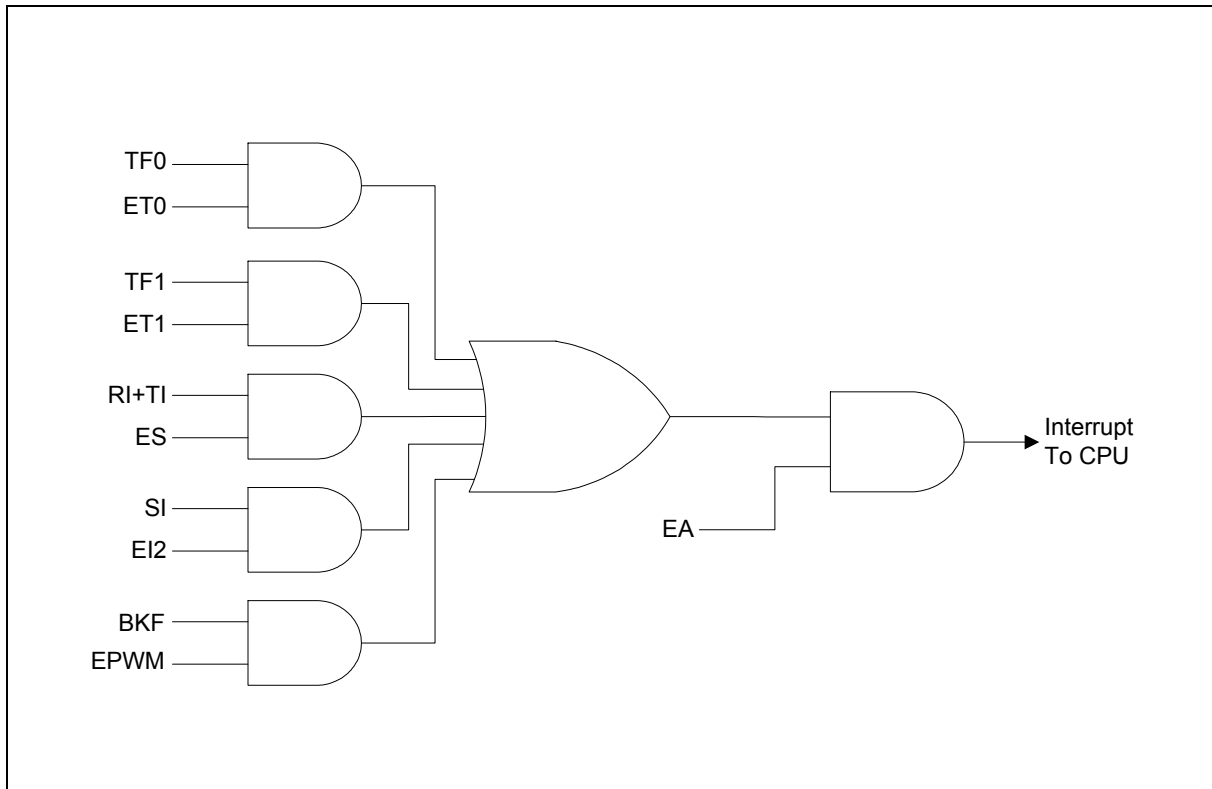
www.dycmcu.com

CMF1, CMF2; 另一个是: IF0, IF1, RI+TI, SI和BKF.。2个中断输入是为了尽量与标准80C51一致, 如下图。

把N79A82X1系列设置为掉电或空闲模式, 如果外部中断为允许, 中断产生将唤醒CPU, 并继续执行程序。



下图所示，这些中断不能唤醒掉电模式



12 可编程定时器/计数器

N79A82X1系列有2个16位可编程定时器/计数器和一个可编程看门狗定时器。看门狗定时器的运行方式不同于其它3个定时器。

12.1 定时器/计数器 0&1

N79A82X1系列有2个16位定时器/计数器，这些定时器中都有2个8位寄存器以构成16位的计数寄存器。对于定时器0它们是TH0（高8位的计数寄存器）和TL0（低8位的计数寄存器）。定时器1也有类似的计数寄存器TH1和TL1。可以将它们设置为定时器（对机器周期进行计数）和外部事件计数器。

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

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。

定时器0低字节 TL0

地址: 8Ah

Bit:	7	6	5	4	3	2	1	0
	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0

定时器1低字节 TL1

地址: 8Bh

Bit:	7	6	5	4	3	2	1	0
	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0

定时器0高字节 TH0

地址: 8Ch

Bit:	7	6	5	4	3	2	1	0
	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0

定时器1高字节 TH1

地址: 8Dh

Bit:	7	6	5	4	3	2	1	0
	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0

定时器模式控制寄存器 TMOD

地址: 89h

Bit:	7	6	5	4	3	2	1	0
	GATE	C/ $\bar{T}$	M1	M0	GATE	C/ $\bar{T}$	M1	M0
	TIMER1				TIMER0			
位	名称	功能						
7	GATE	门控位为1时, 定时器/计数器的运行除受TRx控制外还受 $\overline{\text{int } n}$ 控制, 当TRx和 $\overline{\text{int } n}$ 均为1时定时器/计数器开始运行。该位为0时, 定时器的运行只受TRx的控制						
6	C/ $\bar{T}$	定时器/计数器工作方式选择: 为0时以定时器的方式运行; 为1时对TX脚上的高到低电平变化进行计数						
5	M1	模式选择位						

4	M0	模式选择位
3	GATE	门控位为1时, 定时器/计数器的运行除受TRx控制外还受 $\overline{\text{int } n}$ 控制, 当TRx和 $\overline{\text{int } n}$ 均为1时定时器/计数器开始运行。该位为0时, 定时器的运行只受TRx的控制
2	$\overline{\text{C/T}}$	定时器/计数器工作方式选择: 为0时以定时器的方式运行; 为1时对TX脚上的高到低电平变化进行计数
1	M1	模式选择位 如下表
0	M0	模式选择位 如下表

模式选择位:

M1	M0	模式
0	0	模式 0: 8-位定时器, 有5位的预分频。
0	1	模式 1: 16-位定时器, 没有5位的预分频。
1	0	模式 2: 8位从THx中自动重装定时器
1	1	模式3: (仅适用于T0) TL0是受定时器0控制的8位定时器/计数器。TH0 是受定时器1控制的8位定时器/计数器。定时器1在此方式下不工作。

12.2 时基选择

N79A82X1系列为定时器提供2种时钟源, 一种是标准8051时钟源, 即系统工作频率的1/12为计数时钟源。这种运行方式保证了时间循环与标准的8051一致, 这也是N79A82X1系列默认的定时器时钟来源。用户也可以选择让时钟以加速的方式来运行, 这时的计数时钟源是系统工作频率的1/4, 这样就将计数速度加快了3倍。由CKCON中的T0M和T1M 位来选择加速计数模式。复位后这些为变为0, 定时器工作在标准8051模式下。如果用户要将计数器设为加速模式。

时钟控制寄存器 CKCON

地址: 8Eh

Bit:	7	6	5	4	3	2	1	0
	-保留	-保留	-保留	T1M	T0M	-保留	-保留	-保留
位	名称	功能						
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。						

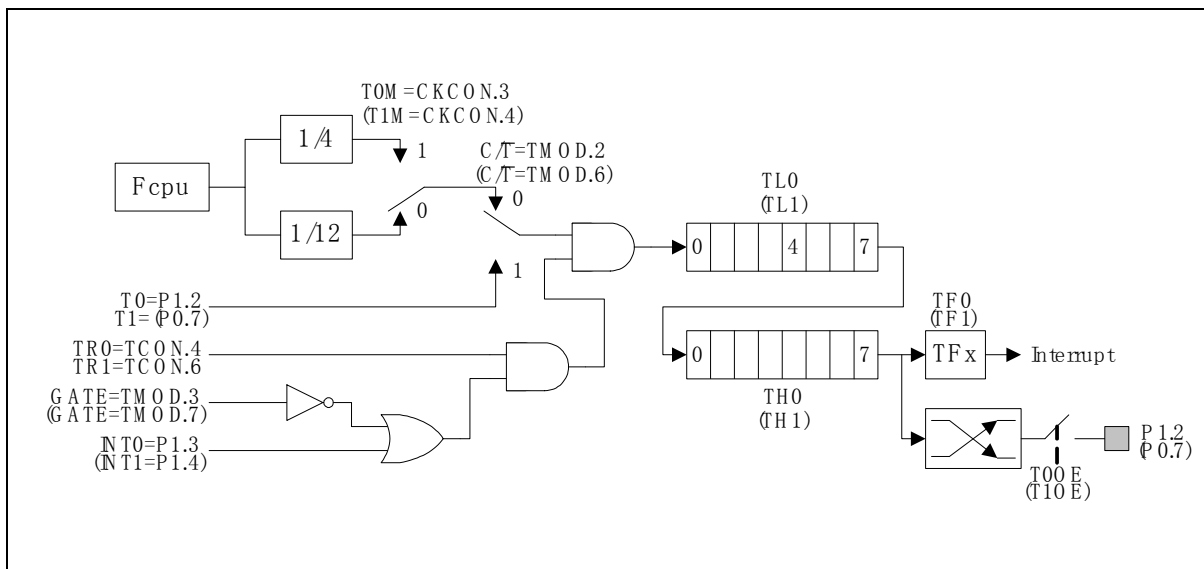
12.3 模式 0

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 $\overline{\text{INTx}}$ 为1时, 计数输入才有效。C/ $\overline{\text{T}}$ =0时, 定时器/计数器对时钟周期进行计数, C/ $\overline{\text{T}}$ =1时对P1.2(T0)以及P0.7 (T1) 上的1到0跳变进行计数。当13位的定时器计数值变为1FFFH后, 下一次计数会使其变为0000H。此时相关的溢出标志位置位如果中断打开, 此时还会产生一个定时

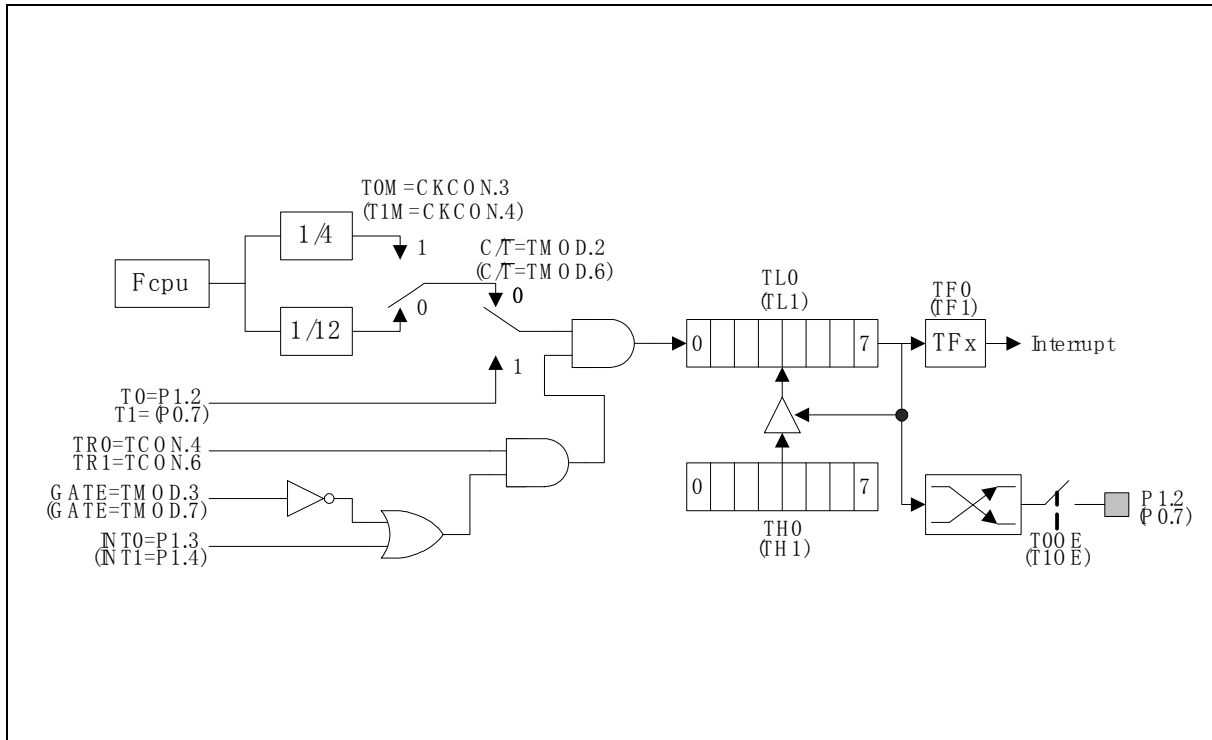
Timing diagram for the 8051 microcontroller showing the sequence of events for the timer overflow interrupt. The diagram includes the Fcpu signal, timer control signals (TOM, TIM, C/T, T), and the timer registers (TL0, TH0). The sequence starts with the Fcpu signal, followed by the timer control signals, and then the timer registers. The timer registers are shown as 8-bit counters. The TL0 register is shown with a value of 4, and the TH0 register is shown with a value of 7. The timer overflow flag (TF0) is set, and the interrupt is generated. The timer is then reset to 0.

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，相应的溢出标志置1，并产生中断。对时钟源的选择与模式0 下的方式一致，门控方式也同模式0 相同。



2007.8.16

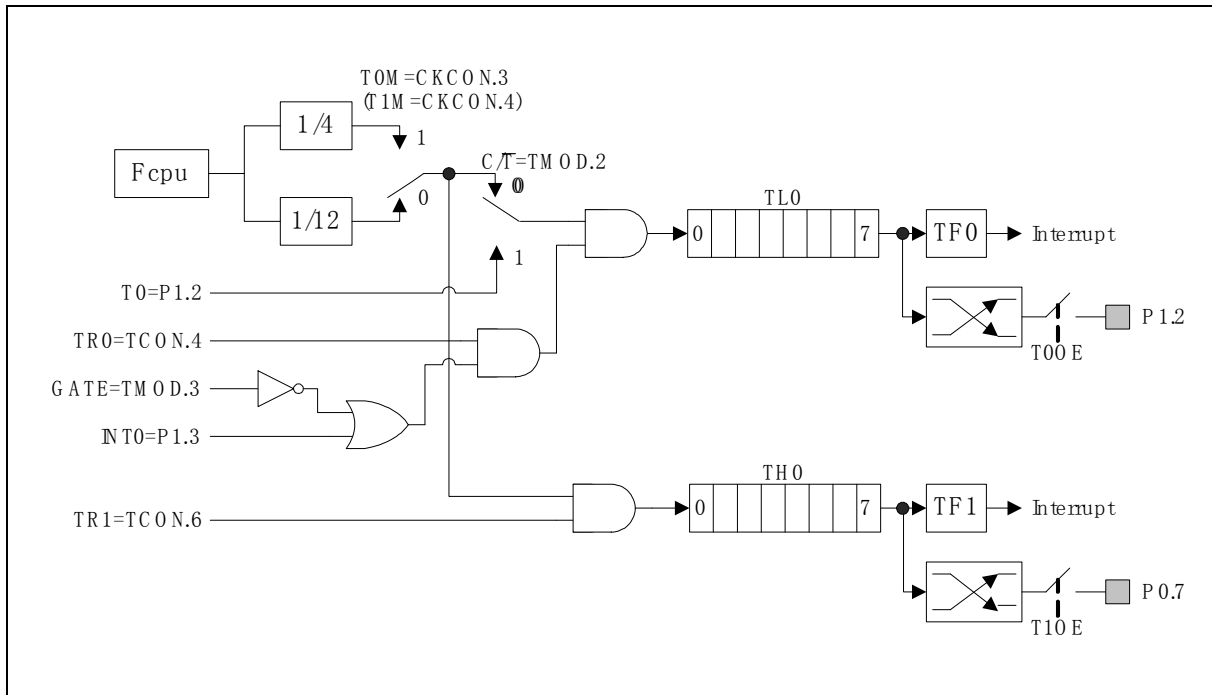
模式2下定时器/计数器为自动重装模式。此模式下TLx是一个8位的计数器，THx保存重装计数值。当TLx由FFH向00H溢出后，TCON中的TFx标志置位THx中内容重装至TLx，继续计数过程。重装过程中THx内的值保持不变。当TRx置位且GATE为0或 $\overline{INTx}$ 为1时，计数器才真正开始工作。同其它2种方式一样，模式2的时钟源可以是系统时钟周期的1/12或1/4。也可对Tn脚上的脉冲输入计数



定时器/计数器 模式 2.

12.6 模式 3

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



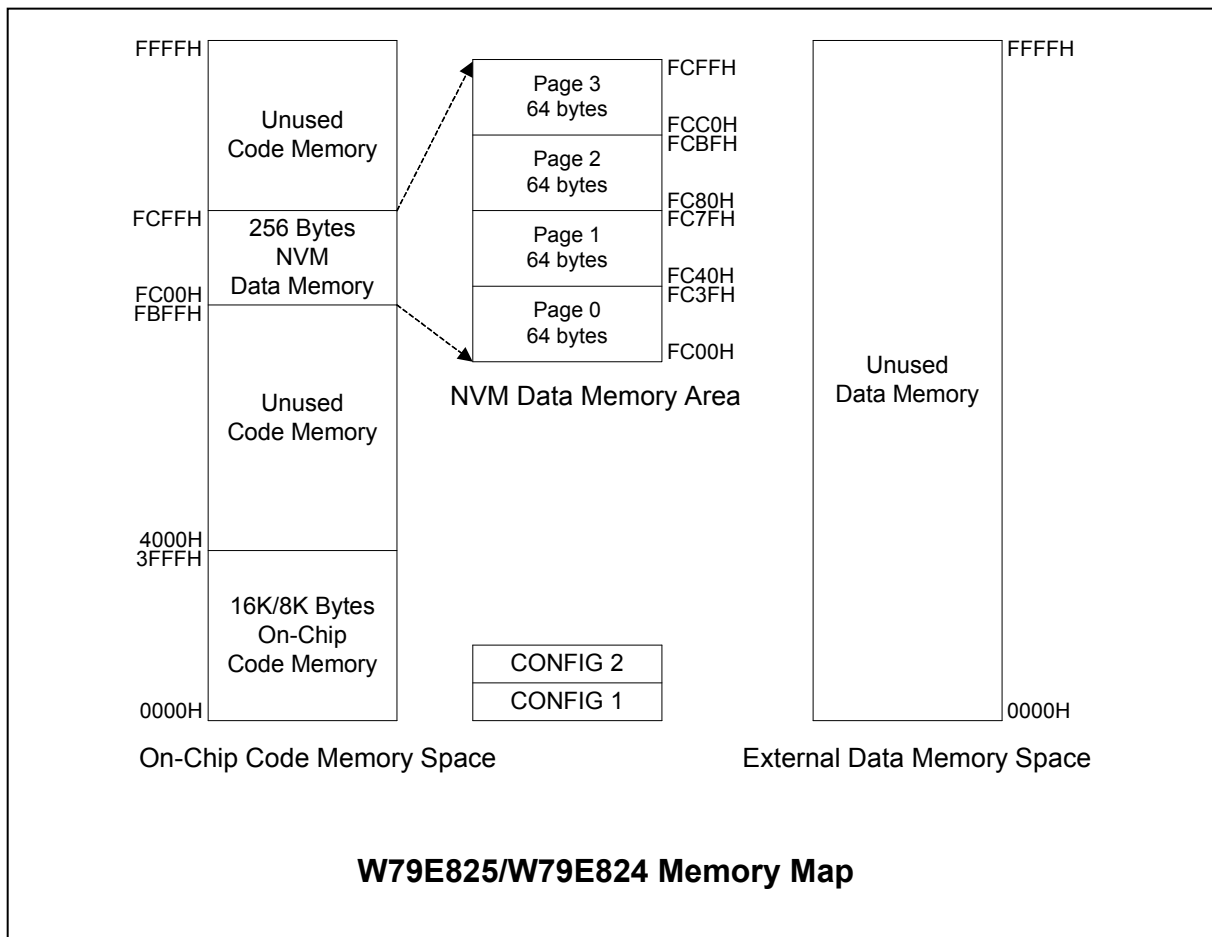
定时器/计数器 模式 3.

13 NVM数据存储器

N79A82X1系列有256/128字节的NVM 数据存储器。这256/128字节NVM数据存储器分为4/2个页面，每页大小是64字节如下图。第0页地址为FC00h ~ FC3Fh，第1页地址为FC40h ~ FC7Fh，第2页地址为FC80h ~ FCBFh，第3页地址为FCC0h ~ FCFFh。

客户程序可以读写NVM数据存储器。读NVM 数据用 `MOVC A,@A+DPTR` 指令，写数据由特殊功能寄存器NVMADDR, NVMDAT和NVMCON访问。在向NVM 内存写数据之前，必须先擦除相应的页。设置页地址 可以译码出片内代码内存空间低字节地址并允许NVMADDR的页，再设置NVMCON.7的EER，在执行页擦除操作期间处理器将自动的控制住(暂停)取程序代码和PC等待页擦除结束，在页擦除结束后这一位由硬件位清除。擦除时间大约为5ms。

向NVM 内存写数据，必须先设置地址和数据到NVMADDR 和NVMDAT，再设置EWR(NVMCON.6)写数据，uC将等待操作结束，数据被写到映像地址，操作完成后该位由硬件清除，然后继续执行程序。写数据的时间大约是50us。



N79E8251/8241/8231/8221

www.dycmcu.com

位	名称	功能
7~0	NVMADDR.7 ~ NVMADDR.0	NVM 地址: 寄存器标识为片内代码内存空间低字节地址NVM数据存储器

助记符: NVMADDR

地

址: C6h

位	名称	功能
7	EER	NVM 页擦除位 0: 不擦除NVM页 1: 该位设为'1', 把NVM 数据页中的内容擦成'FFH'。NVM数据存储器有4个页, 每页大小为64字节。通过NVMADDR 寄存器选择页后, 在设定该位后, 该页将会被清除, 程序指针将会等待该指令的结束。在本指令结束之后程序指针将执行下一条指令。NVM页地址定义如下页
6	EWR	NVM 数据 写 位 0: 不写NVM 数据。 1: 把改位设置为'1'向NVM写一个字节的数据。程序指针将会等待该指令的结束。在本指令结束之后程序指针将执行下一条指令。
5~0	-	保留

助记符: NVMCON

地

址: Ceh

NVM页面地址表:

PAGE	起始地址	结束地址
0	00H	3FH
1	40H	7FH
2	80H	BFH
3	C0H	FFH

注: W79E823, W79E822和W79E821没有页2和页3.

位	名称	功能
7~0	NVMDAT.7 ~NVMDAT.0	写NVM数据寄存器, 读NVM 数据使用MOVC 指令.

助记符: NVM数据

地

址: CFh

```

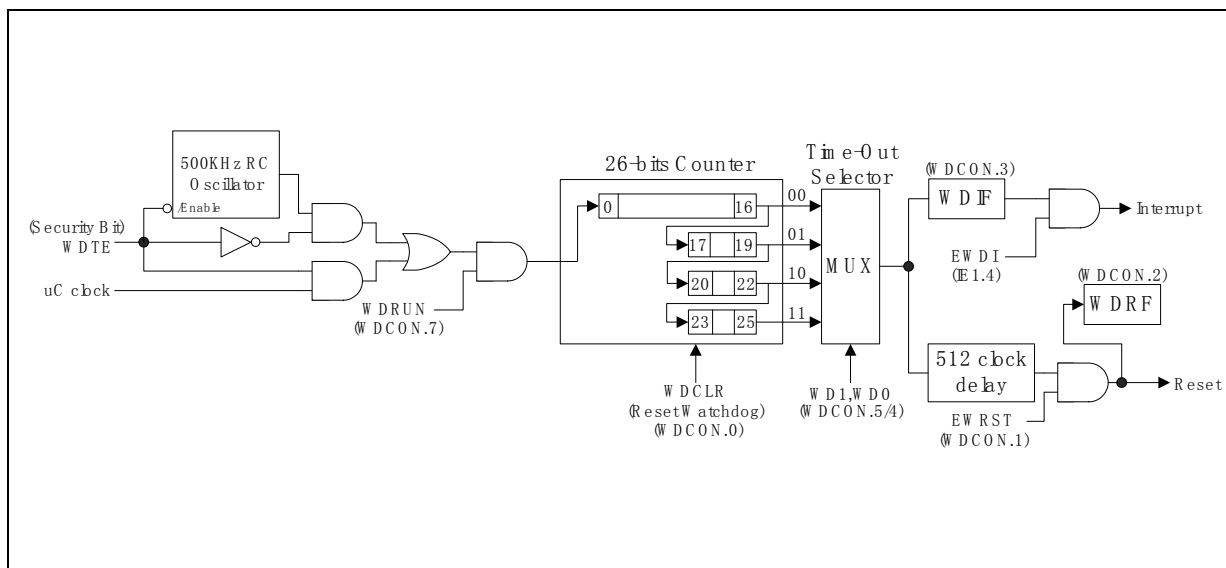
void EraseNVMPage(unsigned char ErasePage_Num)
{
    switch(ErasePage_Num)
    {
        case 0:
            NVMADDR=0x00;    // NVM第0页
            break;
        case 1:
            NVMADDR=0x40;    // NVM第1页
            break;
        case 2:
            NVMADDR=0x80;    // NVM第2页
            break;
        case 3:
            NVMADDR=0xc0;    // NVM第3页
            break;
        default:
            break;
    }
    NVMCON|=0x80;    // EER = 1时进行擦除工作
}

void WriteNVM(unsigned char NVM_WAddr,unsigned char NVM_WData)
{
    NVMDAT = NVM_WData;    // 数据
    NVMADDR = NVM_WAddr;    // 地址
    NVMCON |= 0x40;    // EWR = 1写NVM
}

```

14 看门狗定时器

看门狗定时器是一个自行运行定时器，用户可通过编程将其设置为系统监控器，时基发生器或事件定时器。该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。溢出时，如果看门狗有效（且看门狗计时器重定打开），将引起系统复位。看门狗溢出中断以及看门狗复位功能可由软件设置，将2者的功能合并或分离（即看门狗定时器溢出并使系统复位以及看门狗定时器仅溢出而不引发系统复位）。



看门狗定时器

看门狗定时器应先用WDCLR来重新启动，这保证看门狗定时器从一个确定状态开始运行。WDCLR位用来复位看门狗定时器。该位会自动清0，就是在软件向该位写入1后，系统会自动把该位清为0。将RWT位设为1后，看门狗定时器会对时钟周期进行计数。超时时间由WD1和WD0位来决定（WDCON.5和WDCON.4）。超时时间到以后，WDIF(WDCON.3)位置位；之后看门狗定时器将等待512个时钟周期，如果EWRST (WDCON.1) =1且在等待期间没有对WDCLR 进行操作，那么512个时钟周期以后会产生看门狗定时器复位。这个复位会持续2个机器周期同时WTRF(WDCON.2)标志位置位，软件可以用此位来判别是否是看门狗定时器复位。

看门狗定时器可以用作一个简单的定时器，此时中断和复位功能被关闭。每次超时时间到以后WDIF位会置位。可以对WDIF位进行轮询来检测看门狗定时器的溢出与否，并用RWT位来复位看门狗定时器。看门狗定时器也可用作一个能超长计时的定时器，在这种模式下看门狗定时器中断有效，每次溢出后并在EA=1 时会产生看门狗定时器中断。

看门狗定时器主要用作一个系统监控器，在实时控制的应用中尤为重要。如果出现电源脉冲干扰或电磁干扰，处理器将会运行不确定的代码。如果不及时检查，整个系统可能会崩溃。用户可以在软件中使用看门狗定时器来防止程序运行的错误；用户在软件中适当的地方安排看门狗计时器重定程序，每当运行到看门狗计时器重定程序时就将看门狗定时器复位防止看门狗定时器复位的产生。如果系统受到干扰，程序运行发生异常，系统就可能不会运行看门狗定时器的复位代码，此时系统就会被看门狗定时器复位。

对于不同的时钟速率，看门狗定时器将会产生不同的溢出时间。当使能看门狗定时器复位后，这个复位会在其溢出并经过512个时钟周期后结束。

看门狗定时器溢出值

WD1	WD0	看门狗 INTERVAL	NUMBER OF 时钟	时间 @ 10 MHZ
0	0	217	131072	13.11 mS
0	1	220	1048576	104.86 mS
1	0	223	8388608	838.86 mS
1	1	226	67108864	6710.89 mS

看门狗定时器在上电或掉电复位后无效，看门狗定时器复位不会关闭看门狗定时器，但会将它重新启动，软件应重新启动看门狗定时器把它放入一个确定的状态

看门狗定时器的控制位描述如下。

14.1 看门狗控制

WDIF: WDCON.3 – 看门狗 定时器中断标志。当看门狗定时器定时溢出，该位为置‘1’。如果看门狗中断允许 (IE1.4)=‘1’，就会产生中断(如果全局中断允许位置‘1’且符合其它的中断需求)。软件复位或其它复位使该位清‘0’。

WDRF: WDCON.2 – 看门狗计时器重定标志。当看门狗定时器复位后置‘1’。该位可用来判别复位的类型。软件可以读取该位，但必须手动清除。掉电复位会将此位清除。如果EWDRST = 0，该位不会受看门狗定时器的影响。

EWRST: WDCON.1 – 看门狗定时器复位使能位。为1时使能看门狗定时器复位功能为0 关闭该功能，此时看门狗定时器自由运行

WDCLR: WDCON.0 – 将看门狗定时器复位。该位用于清除看门狗定时器并将它复位。该位会自动清零，在软件向该位写入1后，系统会自动将它置0。如果看门狗计时器重定使能，那么软件必须在看门狗定时器溢出后512个时钟周期内将看门狗定时器清零，否则将会产生一个看门狗定时器复位

14.2 时钟控制看门狗

WD1, WD0: CKCON.5, CKCON.4 – 看门狗定时器模式选择位。这2位用来选择看门狗定时器的溢出时间。复位在定时器溢出并经过512个时钟周期后发生。

默认的看门狗溢出时间是217 个时钟，是最短的溢出时间。**EWT, WDIF**和**RWT**是受时控访问限制的位。这种机制可以防止软件意外读写这些寄存器位。更为重要的是，它将防止无关代码关闭，启动看门狗定时器。

WDTE是**CONFIG**寄存器的第7位，此位是用来配置看门狗定时器的时钟源不论是来自内部RC振荡器还是外部晶振。

当**WDTE**位清0关闭用来运行看门狗时钟**500KHz**时钟频率。由于计数器不自增有可能使看门狗定时器中止。当看门狗使能有效，**WDCLR**位清0看门狗定时器而且下一指令是设置寄存器**PCON**让**CPU**工作在空闲或者掉电模式这个时候就有可能使问题产生。发生这种情况的原因是置1/清0，当**WDCLR**位设置为1，要重新设置它使计数器必须清零。因为当**CPU**时钟停止进入空闲或者掉电模式时计数器没有时钟让它自增。导致的结果就是**WDCLR**一直置位而且看门狗计数器一直停留在0。解决这一问题的方法就是监控**WDCLR**位，保证在有指令让**CPU**进入到空闲或者掉电模式之前清零**WDCLR**。

看门狗控制寄存器 WDCON

地址: D8h

N79E8251/8241/8231/8221

www. dycmcu. com

Bit:	7	6	5	4	3	2	1	0
	WDRUN	-保留	WD1	WD0	WDIF	WTRF	EWRST	WDCLR
位	名称	功能						
7	WDRUN	0: 看门狗停止 1: 看门狗运行.						
5~4	WD1~WD0	看门狗 定时器选择.						
3	WDIF	看门狗定时器中断标志: 如果看门狗中断使能, 硬件会将该位置1 表示看门狗定时器中断产生。如果看门狗定时器中断关闭, 那么该位的置位表示看门狗定时器已经超时。该位必须由软件来清零						
2	WTRF	看门狗计时器重定标志: 当看门狗定时器复位后置位。该位可用来判别复位的类型。软件可以读取该位, 但必须手动清除。掉电复位会将此位清除。如果EWT=0, 该位不会受看门狗定时器的影响						
1	EWRST	0: 禁止看门狗定时器复位. 1: 允许看门狗定时器复位.						
0	WDCLR	看门狗定时器清'0': WDCON.0 – 将看门狗定时器复位。该位用于清除看门狗定时器并将它复位。该位会自动清零, 在软件向该位写入1后, 系统会自动将它置0。如果看门狗计时器重定使能, 那么软件必须在看门狗定时器溢出后512个时钟周期内将看门狗定时器清零, 否则将会产生一个看门狗定时器复位						

void WDT_ISR(void) interrupt 10

```
{
    // 中断防止看门狗定时器使系统复位
    TA = 0xAA;
    TA = 0x55;
    WDCON = 0x00;      // 看门狗从确定的状态开始运行

    TA = 0xAA;
    TA = 0x55;
    WDCON |= 0xE3;     // 再次运行
}
```

void main(void)

```
{
    TA = 0xAA;
    /* 通过向TA寄存器写入AAH, 再写入55H来开启对被保护位的访问*/
    TA = 0x55;
    WDCON = 0x00;      // 使得看门狗从确定的状态开始运行

    TA = 0xAA;
    TA = 0x55;
    WDCON |= 0xE3;     // 看门狗定时器运行, 2^23个时钟周期后溢出

    EWDI = 1;          // 允许看门狗中断
    EA = 1;            // 总中断允许
}
```

}

15 串行口 (UART)

N79A82X1系列有一个全双工串行口。N79A82X1系列还提供附加的功能如，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。

串行数据缓冲器 SBUF

地址: 99h

Bit:	7	6	5	4	3	2	1	0
	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0
位	名称	功能						
7~0	SBUF	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						

串行口控制寄存器 SCON

地址: 98h

Bit:	7	6	5	4	3	2	1	0
	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI
位	名称	功能						
7	SM0/FE	串行口0，模式0控制位或贞错误标志位。PCON特殊功能寄存器中的SMOD0位决定该位的功能。下面会描述SM0的运行功能。当用作贞错误标志时，该位的置位表示一个无效的停止位。该位必须由软件来清除						
6	SM1	串行口模式位1						
		SM0	SM1	模式	说明	数据长	波特率	
		0	0	0	同步	8	时钟的4或12分之一	
		0	1	1	异步	10	可变	
		1	0	2	异步	11	时钟的64或32分之一	
1	1	3	异步	11	可变			
5	SM2	多机通信控制。将该位置1，则使能模式2及模式3下的多机通信功能。在模式2或3下，如果SM2置1，那么收到的第九位数据RB8是0的话，RI将不会置位。在模式1下如果SM2置1，那么在没有收到有效的停止位前RI是不会置位的。在模式0下，SM2位控制着串行口的时钟。如果清0，那么串行口的时钟是系统时钟的12分频。这样系统就与标准8052兼容。如果该位置1，那么串行口的时钟是系统时钟的4分频，这样就加快了同步通信的速度						
4	REN	接收使能，置1时打开串行口接收功能，否则关闭该功能						
3	TB8	模式2和3中要被发送的第九位数据。软件可以根据需求将该位置1或清0						
2	RB8	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：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。该位必须由软件来清除						

15.1 模式 0

该模式提供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脚用于产生移位元时钟。在发送或接收时TXD上的时钟由N79A82X1系列提供。这种方式下是以半双工的形式

停止位输出到TXD脚以后，TI会在C1态置位。这发生在向SBUF写入数据后16状态计数器的第11次翻转以后。当REN=1时系统进行接收操作，接收器以所选波特率的16倍速度采样RXD脚状态

当REN='1'时接收使能，当RXD脚上接收到1-0跳变就启动接收器接收。监视器一直监视RXD，进行16倍于波特率的速率采样，检测到下降沿时，16状态定时器清'0'；该定时器用于分辨字符界限。

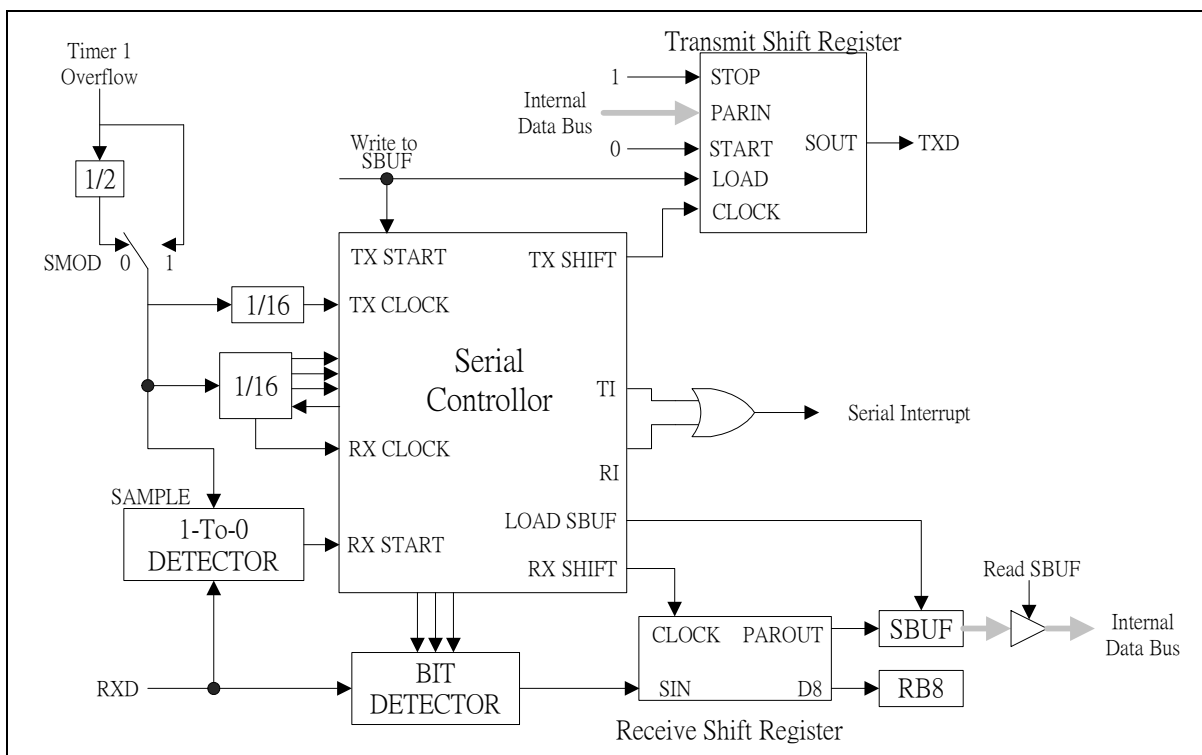
16状态寄存器把1个位有效时间分成16个时间单元，位检测最好有3个时间点，分别在第8、第9、第10个时间3个单元时间上检测RXD脚的状态，3次采样中至少2次相同的值，以保证接收准确。在检测到RXD上的下降沿后，RXD不为0，则起始位无效，复位接收电路，当再次接收到一个由1-0的跳变时重新启动接收器。如果接收值为0起始位有效，接收器开始接收本帧的其余信息。这样做是为了提高串口的噪声抑制特性。

在接收了8位数据以后，还将接收一个停止位，进入RB8，之后RI置位。然而在RI置'1'之前必须设置相应的条件。

RI=0，且

SM2=0或接收到的停止位为1时才有效

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



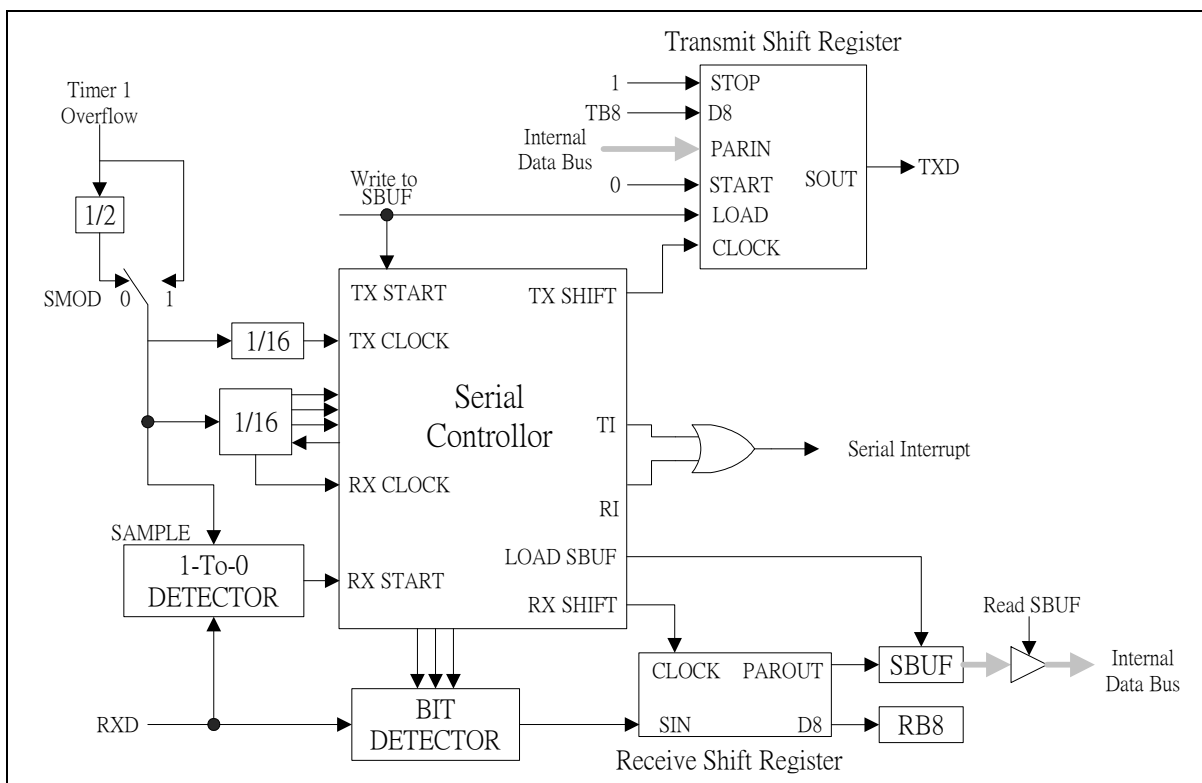
串行口 模式 1

15.3 模式 2

该模式用11位数据来进行全双工异步通信。下图是对他的功能描述。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(0)组成。第9位数据接收至RB8。波特率是时钟

15.4 模式 3

模式3中除了波特率可编程外，其它方面都与模式2相同。用户必须在进行串行通信前初始化SFR寄存器。初始化动作包括模式和波特率的选择。如果是用模式1 或模式3，那么定时器1也要被初始化。在所有的模式中对SBUF写入数据将启动一次发送。在模式0中当RI=0和REN=1时启动一次接收。这时TXD脚上会出现同步时钟，并在RXD脚上传送8位数据。在其它模式下，接收动作在REN=1且接收到数据后就启动。外部设备以发送起始位的方式来开始串行通信。



串行口模式3

串行口的模式

SM1	SM0	模式	类型	串口时钟	数据帧大小	起始位	停止位	第9位功能
0	0	0	同步	4或12 TCLKS	8 位	无	无	无
0	1	1	异步	定时器1	10 位	1	1	无
1	0	2	异步	32或64TCLKS	11 位	1	1	0, 1
1	1	3	异步	定时器1	11 位	1	1	0, 1

15.5 帧错误检测

当没有检测到一个有效的停止位时，可能就出现了一个帧错误。这表示一个无效的串行资料接收。通常错误是由串行通信在线的干扰造成的。N79A82X1系列可以检测这种错误，并将标志位置位，以供软件进行检测。

SCON.7是FE标志（帧错误标志）（FE_1）。在标准8051中该位是SM0，但在N79A82X1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。一个是SM0，一个是FE。具体访问哪一个位是由SMOD0（PCON.6）决定的。当SMOD0=1时访问FE标志位，当SMOD0=0时访问SM0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贞不会将其清除。对该位的清除必须由软件来完成。

电源控制寄存器 PCON

地址: 87h

Bit:	7	6	5	4	3	2	1	0
	SMOD	SMOD0	-保留	POR	GF1	GF0	PD	IDL
位	名称	功能						
7	SMOD	1: 使串行口在模式1, 2, 3下的波特率加倍.						
6	SMOD 0	0: SCON.7表示一个贞错误它是FE（贞错误）标志 1: SCON.7的功能与标准8052中SCON.7相同.						

15.6 多机通信

多机通信利用模式2和模式3下的第九位数据，在N79A82X1系列中RI仅在接收的数据贞的地址符合本机地址或系统进行广播通信时置位。硬件所具有的特性，免除了要软件进行地址识别的麻烦。

在多机通信模式下，当第9位置1时，发送的数据是地址贞。当主机想对从机发送数据块，它首先发送从机的地址贞，当从机在接收地址贞时，他们的SM2位必须为高。这保证他们能在接收到地址贞时产生中断。自动地址识别功能保证只有在接收到的地址和本机地址符合时才产生中断。地址比较由硬件来完成。

被寻址的从设备将SM2位清零，然后准备开始接收数据。SM2=0后，每当接收到一个有效数据贞从机就会产生一个中断。未被寻址的从设备不会受到影响，因为他们在等待自身地址的到来。在模式1中，第九位是停止位，1是有效的停止数据。如果SM2=1那么只有在接收到有效数据且自身被寻址后RI才会置1。

从机地址寄存器 SADDR

地址: A9h

Bit:	7	6	5	4	3	2	1	0
	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0
位	名称	功能						
7~0	SADDR	SADDR 中应当写入多机通信时的广播地址或是从机地址						

从机地址屏蔽允许寄存器 SADEN

地址: B9h

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-
位	名称	功能						
7~0	SADEN	该寄存器使能串口0的自动地址识别功能，当SADEN中的某位被置为1，那么SADDR寄存器中的相应位会同接收到到的数据进行比较。如果SADEN.n被设为0，那么系统会忽略对该位的比较。如果SADEN为全0，那么对于所有的地址帧系统都会产生中断。						

主机可以用从机地址来选择性的访问从机。可以用广播的方式来寻址所有的从机。从机的地址由SADDR和SADEN寄存器来定义，从机地址是由SADDR设定的8位数据，如果SADEN中相应的位置0则SADDR中对应的位就无效。只有当SADEN中的相应位为1，SADDR中的数据才有效。

下面的代码说明如何定义从机地址，以及寻址不同的从机

从机 1:

```
SADDR 1010 0100
SADEN 1111 1010
Given  1010 0x0x
```

从机 2:

```
SADDR 1010 0111
SADEN 1111 1001
Given  1010 0xx1
```

从机1和2的地址在最低位处不同，在从机1中该位被忽略，而在从机2中该位有效。因此要与从机2通信的话，那么他地址数据的位1应该为1。如果主机要与所有从机通信，那么地址数据的位0=1且位1=0。位3被忽略。这样就形成了广播地址。

主机能用广播的方式来和所有从机通信，地址是SADDR和SADEN中数据的逻辑与。相应得位如果为0，那么该位就被忽略。在大多数应用场合，广播地址是FFh，而在上面的例子中从机1的广播地址是（1111111X），从机2的广播地址是（11111111）。

SADDR 和SADEN的地址分别是 A9h和B9h。复位后，2个寄存器的值均为0；这样广播地址和给定的地址都无效，这样多机通信功能就被关闭。

串行口控制初始化示例：

```
TMOD&=0x00;
TMOD|=0x20;    //定时器1选择模式2
TH1=0xFD;
TL1=0xFD;      //11.0592M晶振的时候波特率为9600
SCON = 0x50;   //串行口工作在模式1，接收使能
PCON|=0x80;    //SMOD置位，串行口波特率加倍
TR1=1;         //启动定时器1
ES=1;          //打开串行口中断
```

16 时控访问保护

N79A82X1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，N79A82X1系列提供了一种保护机制，来控制对这些位的写操作。这种保护是通过时控访问来实现的。

在这种方式下，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。TA寄存器的地址是C7H，下面列出对时控寄存器进行访问的推荐代码：

```
TA    REG    0C7h           ;定义位于C7H处的新寄存器TA
MOV   TA, #0Aah
MOV   TA, #055h
```

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。

时控访问的例子：

例1：有效访问

```
MOV  TA, #0Aah      3 M/C      注: M/C =机器周期
MOV  TA, #055h      3 M/C
MOV  WDCON, #00h    3 M/C
```

例2：有效访问

```
MOV  TA, #0Aah      3 M/C
MOV  TA, #055h      3 M/C
NOP                      1M/C
SETB EWT            2 M/C
```

例3：有效访问

```
MOV  TA, #0Aah      3 M/C
MOV  TA, #055h      3 M/C
ORL  WDCON, #00000010B  3M/C
```

例4：有效访问

MOV	TA, #0Aah	3 M/C
MOV	TA, #055h	3 M/C
NOP		1 M/C
NOP		1 M/C
CLR	POR	2 M/C

例5：无效访问

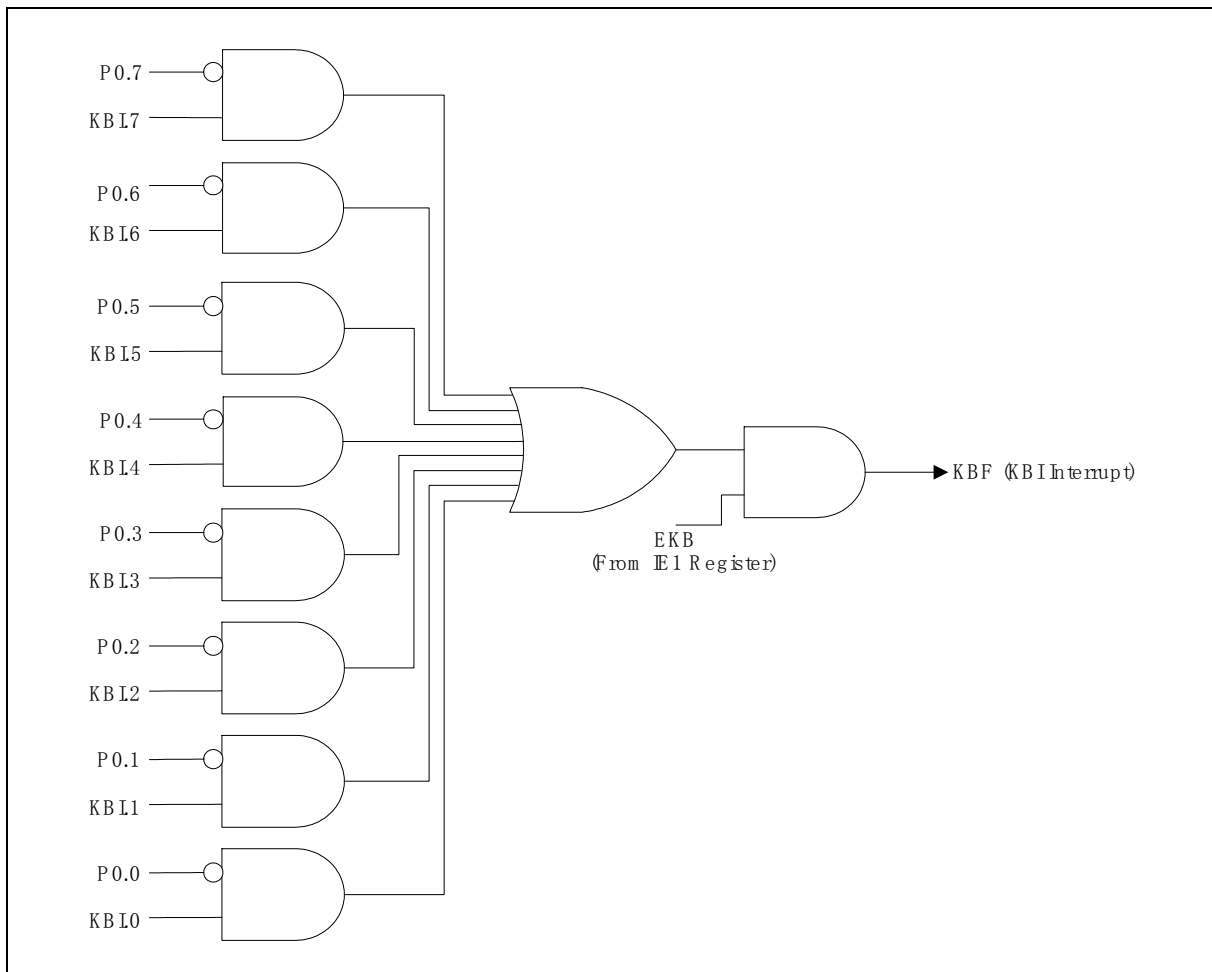
MOV	TA, #0Aah	3 M/C
NOP		1 M/C
MOV	TA, #055h	3 M/C
SETB	EWT	2 M/C

在前2个例子中，对被保护位的写是在3个机器周期以内完成的。例3中对保护位的写操作是在时控窗口关闭后进行的，此时不会对被保护的位产生效果。例4中是在第4个机器周期对被保护位进行写操作，因此写操作根本无效。

17 键盘中断 (KBI)

N79A82X1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N79A82X1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。

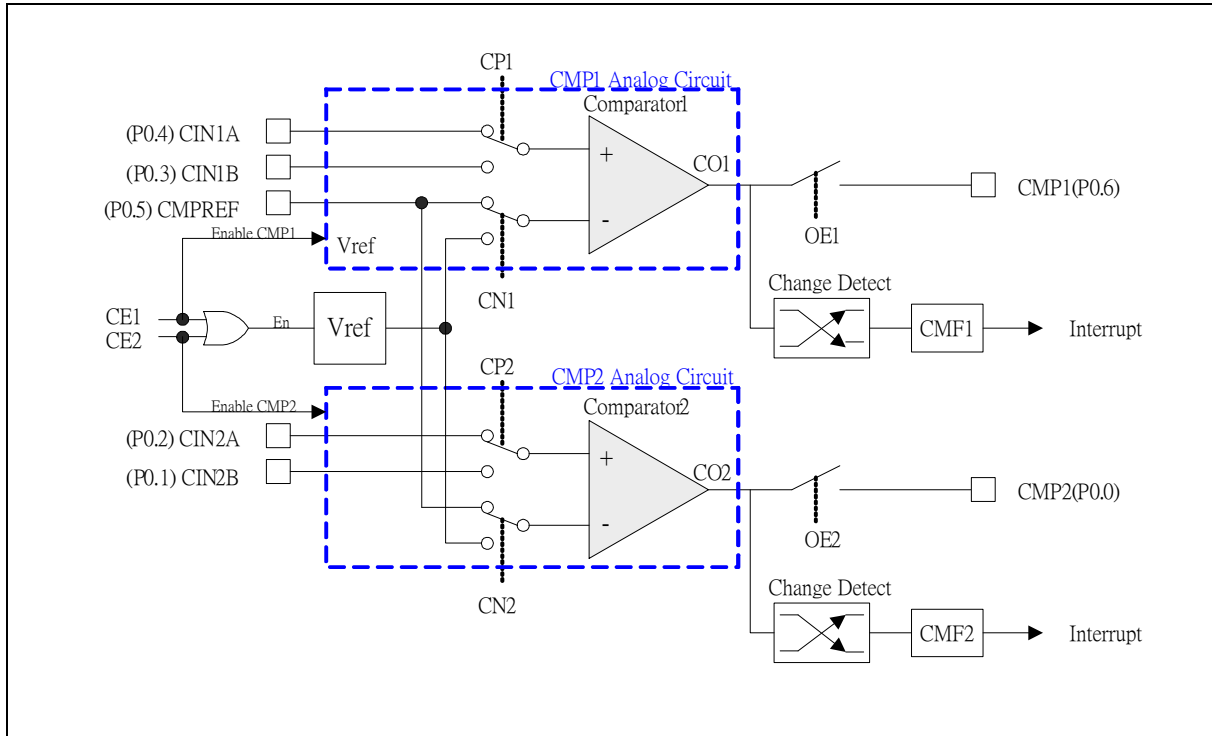
P0支持键盘功能，P0的所有的管脚都可以产生中断。在KBI寄存器中可以设置端口允许KBI0 ~ KBI7功能，如下表。当设置允许KBI管脚拉低产生中断，且中断有效，管脚拉低(大于1个机器周期)，AUXR1寄存器中的键盘中断标志(KBF)会置'1'，中断将会产生。KB有硬件置位，必须由软件清'0'。判断哪一个按键按下KBI中断 程序里可以检测P0口。



18 模拟比较器

N79A82X1系列MCU提供2个比较器。比较器的输入、输出和允许选项用户可以根据需要配置。当正极电平高于负极时，比较器的输出'1'，反之输出'0'。每一个比较器都可以设置成，当输出改变时产生比较器中断，框图如下：

每一个比较器都有一个控制寄存器 (CMP1和CMP2)，每一个比较器的输入有：CINnA, CINnB, CMPREF和内部参考电压，把Oen位置'1'，输出是CMP1和CMP2。在允许比较器后，比较器需要等待一定的时间确保比较器稳定。如果使用内部的参考电压，必须把Oen位置“1”。内部参考电压值 (Vref)为 1.19V +/- 10%。



19 I/O 端口配置

N79A82X1系列MCU有2个I/O端口，P0，P1。所有的I/O端口配置都可以通过软件配置成4种类型，比如P1.5脚，当配置成为复位引脚时，只是输入，设置方法是配置1寄存器RPD=0；P1.5配置为输入脚，N79A82X1系列MCU可以支持16个I/O。I/O端口设置如下表：

I/O 端口配置表

PXM1.Y	PXM2.Y	端口模式配置
0	0	准双向
0	1	推挽模式
1	0	输入(高阻)
1	1	开漏

所有的端口引脚可以通过CONFIG1寄存器的PRHI位配置成在复位后管脚输出高或低，复位后这些引脚是准双向模式，端口P1.5只能是施密特触发器型输入。P2M1寄存器的ENT0 和 ENT1位可以允许定时器0和定时器1输出，输出频率由定时器0或定时器1的定时器溢出决定。

N79A82X1系列的每一个I/O端口可以通过P2M1寄存器的P(n)S位选择TTL电平输入或施密特输入，n=0, 1, 2。当P(n)S置1，相应的端口选择为施密特触发器输入。

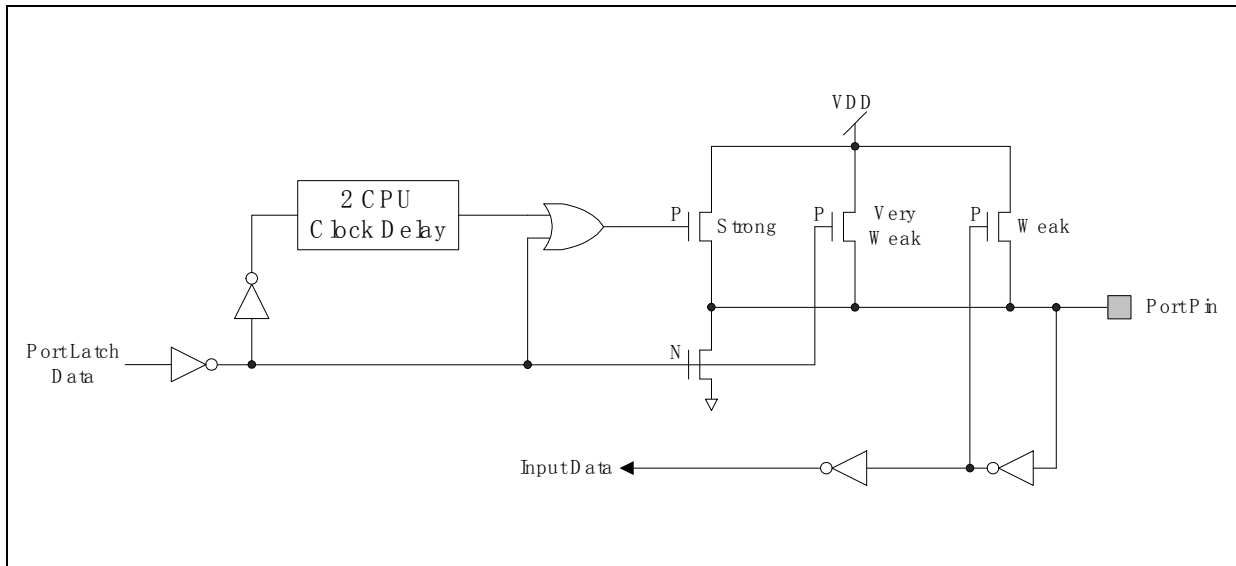
19.1 准双向端口模式配置

当MCU上电或复位后，所有的端口都是这种模式，输出与标准的8051一致，这种模式可以直接用作输入或输出，不需要另行配置。当管脚输出为低有很强的驱动能力可以提供很强的灌电流，这个功能有一点象开漏输出。不同的是在准双向口有3个上拉电阻，以适应不同的应用。

这个模式有3个上拉分别是“强上拉”，“弱上拉”和“特弱上拉”。“强上拉”使从“0”跳变到“1”跳变很快，适应于快速收发应用。当端口引脚上出现逻辑“0”到“1”的跳变，强上拉迅速打开2个CPU时钟后关闭强上拉。

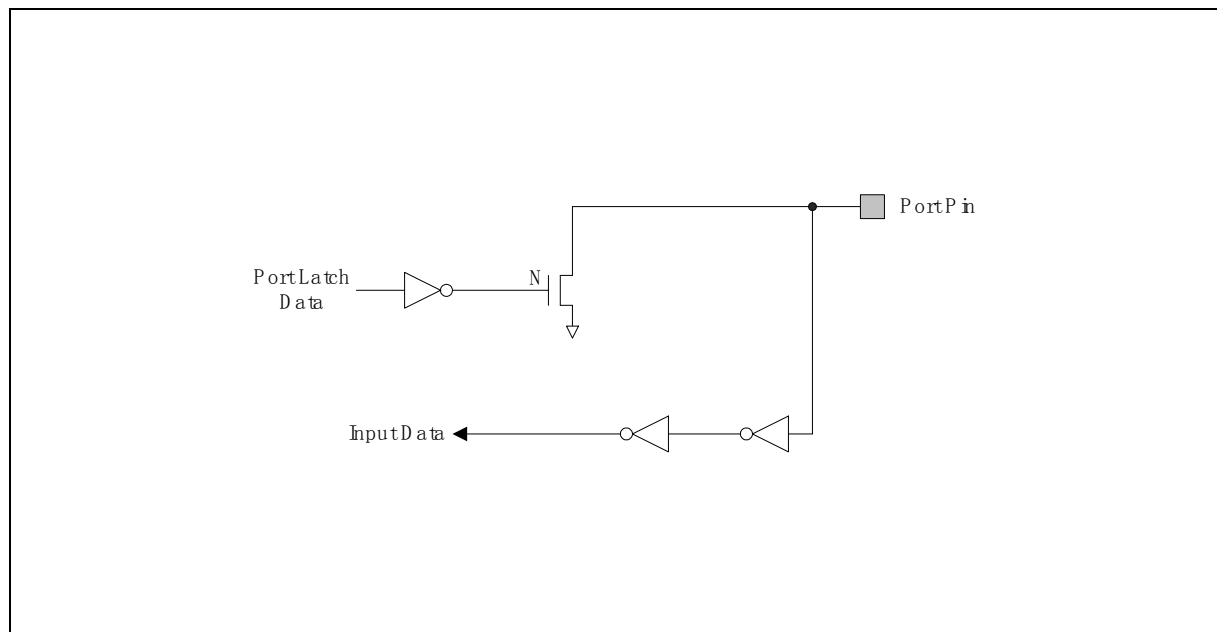
当输入端口为高电平或输出高电平“弱上拉”打开，给准双向口提供电流维持输出“1”或保持端口“0”。

输入端口保持“1”时，“特弱上拉”打开；输入端口保持“0”时，“特弱上拉”关闭。特弱上拉提供非常小的电流保持引脚不在悬浮状态。准双向端口的配置如下表。



19.2 开漏端口模式配置

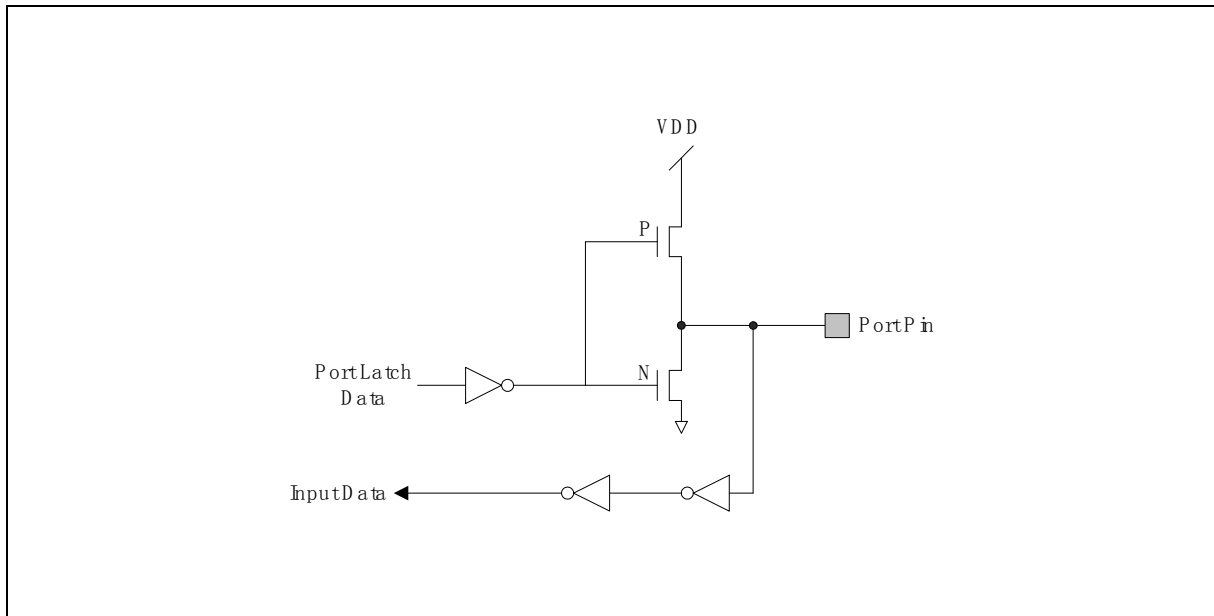
该配置关闭所有的上拉。如果作为逻辑输出必须外加上拉电阻，开漏端口配置如下：



19.3 推挽端口模式配置

推挽输出模式由2个强上拉和一个下拉构成，可以提供正向和反向的电流输出。如果在准双向输出模式中移去“弱上拉”和“特弱上拉”保留“强上拉”，当端口保持为高时，“强上拉”一直打开提供电流。推挽端口配置如下图：

N79A82X1系列MCU有3个端口引脚不可以：P1.2, P1.3,和P1.5。端口引脚P1.2和P1.3可以配置成开漏输出。可以分别设置他们的端口缓冲，把他们设为输入模式。

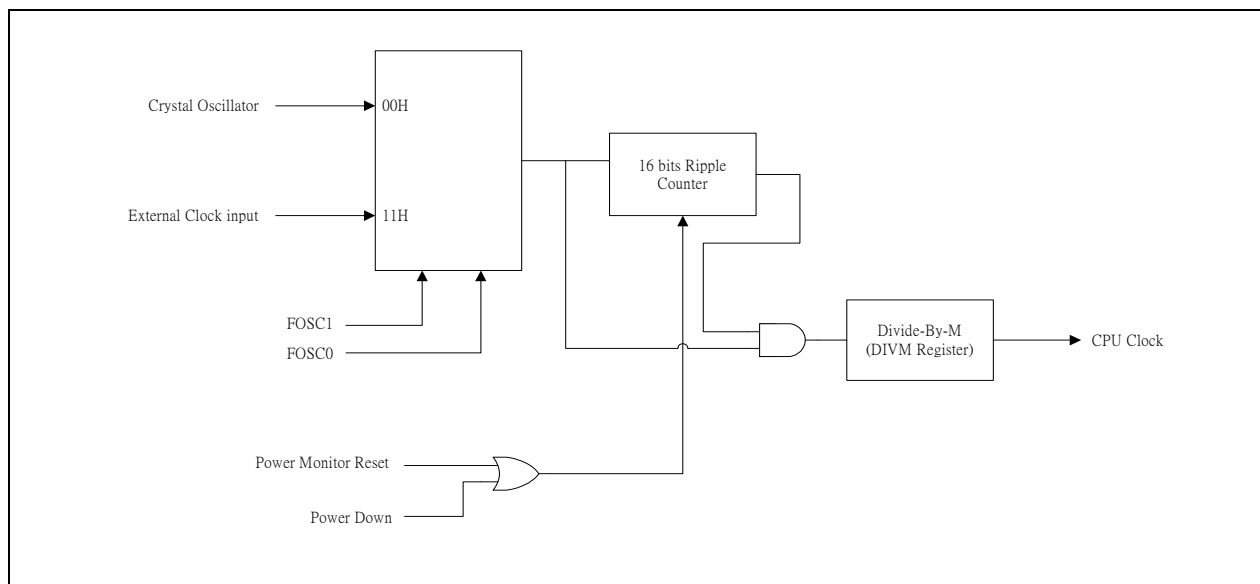


19.4 输入配置

端口配置成这种模式，只能作为数字输入并禁止数字输出。N79A82X1系列可以通过PxM1.y和PxM2.y寄存器选择输入是施密特触发器或TTL电平输入。

20 振荡器

N79A82X1系列提供3个振荡器输入选项。在CONFIG寄存器 (CONFIG1) 中设置，包括：外部时钟输入选项和晶振振荡器输入选项。晶振振荡器输入频率可以支持从4MHz到20MHz，不需要电阻或电容



20.1 外部时钟输入选项

当FOSC1, FOSC0 = 11H时，时钟源脚(XTAL1)可以从外部输入时钟，频率范围是0Hz到20MHz。时钟可以从 (XTAL2) 脚输出。

N79A82X1系列MCU支持时钟输出功能，当选择外部时钟输入选项时，为了使N79A82X1系列和外部设备的同步，通过P2M1寄存器的ENCLK位使XTAL2/CLK输出脚上输出时钟，任何只要振荡器振荡模式下包括空闲模式时钟输出频率为1/4 CPU时钟。如果为了节省电流在空闲模式不需要时钟输出，可以在进入空闲模式前关掉时钟输出。外部时钟输入时也可以输出时钟

20.2 CPU时钟速度选择

N79A82X1系列的CPU时钟可以由DIVM寄存器选择。DIVM = 00H，CPU时钟运行在4CPU时钟/ 机器周期，源时钟(Fosc)没有分频。当DIVM寄存器设置为N时，CPU时钟被分为2(DIVM+1)，所以CPU时钟是从4到512分频。用户可以使用这个特性来降低CPU的速度节省功耗，原理与CPU进入空闲模式相似。另外该分频功能启用后将影响所有外设的时钟，因为外设的时钟为CPU (Fcpu)时钟。

21 电源监视功能

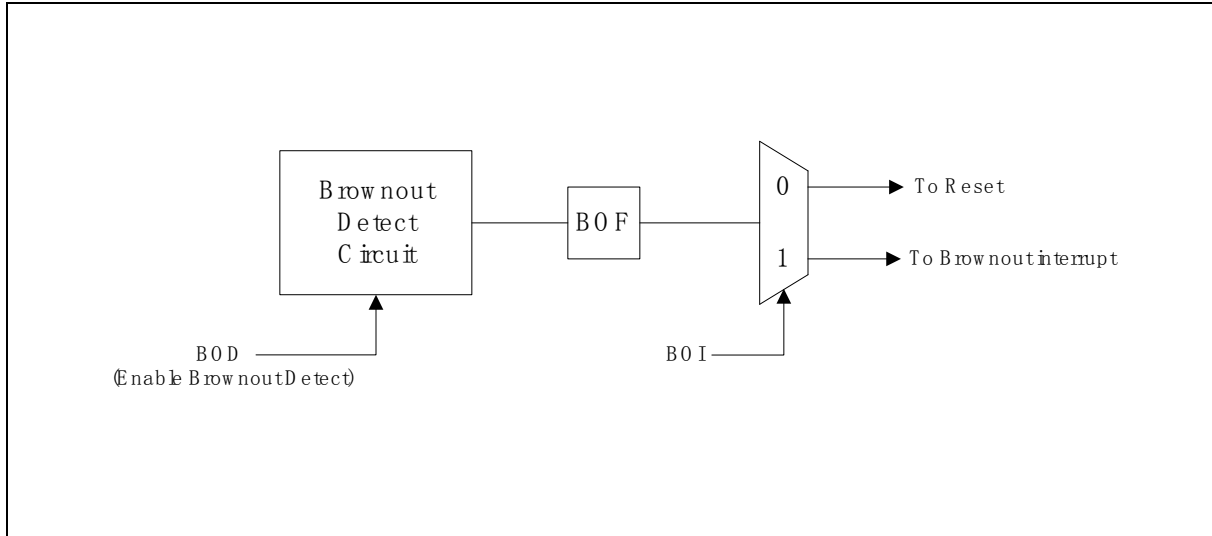
为了防止在上电或电源不稳时出现错误，N79A82X1系列提供2个电源监视功能：上电检测和欠压检测。

21.1 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。

21.2 欠压检测

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。N79A82X1系列有2个欠压检测值，由BOV (配置1.4)选择。如果BOV =0欠压值为3.8V，如果BOV = 1欠压值为2.5V。当电压降到选择的值，欠压监测器将检测并保持直到VDD回到欠压检测电压以上。欠压检测框图如下：



欠压检测启动BOD (AUXR.6)，BOF (PCON.5) 标志置位可以触发欠压复位或中断，BOF将由软件清除，如果BOI (AUXR1.5)置“1”，欠压检测触发中断EA (IE.7)并EBO (IE.5) 位置‘1’.为了确保正确的检测欠压，VDD下降时间必须慢于50mV/us，上升时间慢于2mV/us。

22 脉宽调制(PWM)

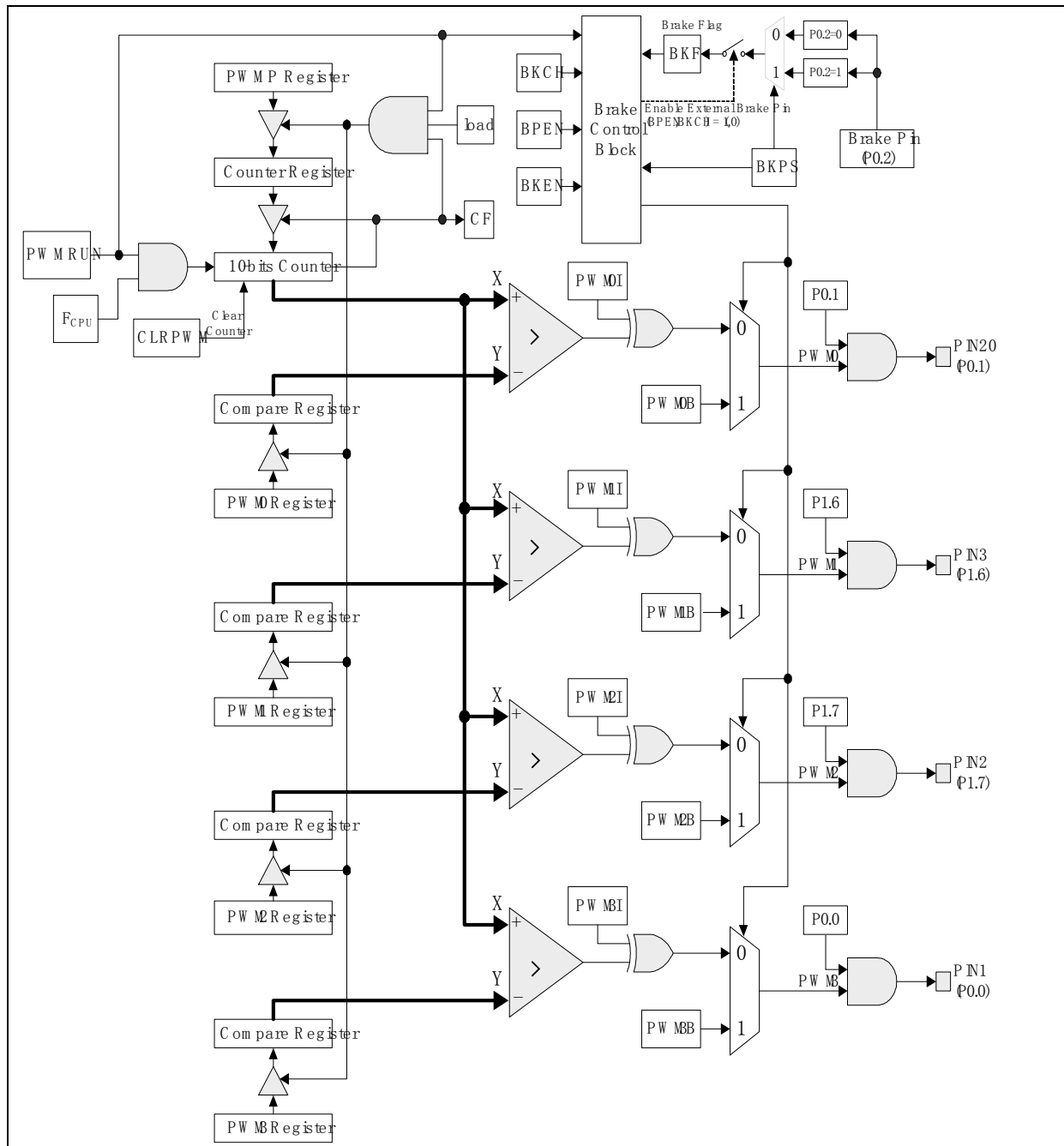
N79A82X1系列中有4个PWM输出通道，PWM分别由PWM0(P0.1)、PWM1(P1.6)、PWM2(P1.7)和PWM3(P0.0)输出。当把PRHI置“1”，复位后PWM个输出通道为‘1’；当把PRHI置“0”，复位后PWM个输出通道为‘0’。因此如果PWM引脚输出“1”，必须向各PWM引脚写“1”，结构框图如下。

N79A82X1系列支持10-位向下PWM计数器，它的时钟源为控制器时钟。PWM计数器时钟与CPU频率一致 $FCPU = FOSC$ 。当计数器向下溢出后计数器的内容将自动从计数器寄存器重装。PWM的频率为： $f_{PWM} = FCPU / (PWMP+1)$ ，PWMP的10-位寄存器由PWMPH.1、PWMPH.0和PWML.7~PWML.0组成。

PWMP寄存器写入的数据在会自动的装载到PWMRUN，CF标志为10-位向下计数器向下溢出，CF标志在下一个周期自动清除，当PWMP寄存器被装载到计数器寄存器，装载位将在下个周期被自动清除。如果第一PWM输出周期由PWMP设定，CLRPWM把10-位计数器设为000H，CF也会被清除；然后设置PWMRUN和装载位来启动PWM。

每一个PWM输出脉冲的宽度由比较寄存器PWM0L~PWM3L和PWM0H~PWM3H决定的，当PWM比较寄存器的值大于10-位计数器寄存器时，PWM输出低。如果要输出可变的PWM脉宽，在写PWMn寄存器后，必须把装载位置“1”，向下溢出时把PWMn寄存器的值装载到比较寄存器。PWM输出高脉冲宽度是：

$t_{HI} = (PWMP - PWMn + 1)$ 。注：如果把比较寄存器设置成000H，PWMn输出高；如果比较寄存器置成3FFH，PWMn输出低。



有关PWM控制寄存器有PWMCON1, PWMCON2,和PWMCON3寄存器，功能描述如下：

PWM 计数器低位寄存器

PWMPL(D9

H)

位	名称	功能
7~0	PWMP.7 ~PWMP.0	PWM 计数器 低位 7~0 寄存器

PWM 计数器高位寄存器

PWMPH(D1

H)

位	名称	功能
7~2	-	保留
1~0	PWMP.9 ~PWMP.8	PWM 计数器 低位 9~8 寄存器

PWM 0 低位寄存器

PWM0L(DA

H)

位	名称	功能
7~0	PWM0.7 ~PWM0.0	PWM 0 低位 7~0 寄存器

PWM 1 低位寄存器

PWM1L(DB

H)

位	名称	功能
7~0	PWM1.7 ~PWM1.0	PWM 1 低位 7~0 寄存器

PWM 2 低位寄存器

PWM2L(DD

H)

位	名称	功能
7~0	PWM2.7 ~PWM2.0	PWM 2 低位 7~0 寄存器

PWM 3 低位寄存器

PWM3L(DE

H)

位	名称	功能
7~0	PWM3.7 ~PWM3.0	PWM3低位7~0寄存器

PWM 0 高位寄存器

PWM0H(D2

H)

位	名称	功能
7~2	-	保留
1~0	PWM0.9 ~PWM0.8	PWM0低位9~8寄存器

PWM1高位寄存器

PWM1H(D3

H)

位	名称	功能
7~2	-	保留
1~0	PWM1.9 ~PWM1.8	PWM 1低位9~8寄存器

PWM2高位寄存器

PWM2H(D5

H)

位	名称	功能
7~2	-	保留
1~0	PWM2.9 ~PWM2.8	PWM 2低位9~8寄存器

PWM3高位寄存器

PWM3H(D6

H)

位	名称	功能
7~2	-	保留
1~0	PWM3.9 ~PWM3.8	PWM3低位9~8寄存器

PWM 控制寄存器 1

PWMCON1(

DCH)

位	名称	功能
7	PWMRUN	0: PWM没有运行. 1: PWM计数器运行.
6	Load	0: PWMP和PWMn 寄存器的值不会被装载到计数器和比较器寄存器中; 1: 在计数器向下溢出后, PWMP和PWMn寄存器的值将会装入计数器和比较器寄存器, 在PWMP和PWMn被发送到计数器和比较器寄存器后, 该位由硬件自动清'0'.
5	CF	0: 10-位向下计数计数器没有溢出. 1: 10-位向下计数器向下溢出。由软件清'0'.

4	CLRPWM	1: 把10-位 PWM 计数器设置为 000H. 置位后硬件自动清零。
3	PWM3I	0: PWM3正相输出 1: PWM3反相输出
2	PWM2I	0: PWM2正相输出 1: PWM2反相输出
1	PWM1I	0: PWM1正相输出 1: PWM1反相输出
0	PWM0I	0: PWM0正相输出 1: PWM0反相输出

PWM控制寄存器2

PWMCON2(

DFH)

位	名称	功能
7	BKCH	见下表
6	BKPS	0: P0.2=0, 钳制PWM 1: P0.2=1, 钳制PWM
5	BPEN	见下表
4	BKEN	0: 钳制禁止. 1: 钳制允许, 详见下表;
3	PWM3B	0: 当钳制有效PWM3输出为低 1: 当钳制有效 PWM3输出为高
2	PWM2B	0: 当钳制有效 PWM2输出为低 1: 当钳制有效 PWM2输出为高
1	PWM1B	0: 当钳制有效 PWM1输出为低 1: 当钳制有效 PWM1输出为高
0	PWM0B	0: 当钳制有效 PWM0输出为低 1: 当钳制有效 PWM0输出为高

钳制条件表

BPEN	BKCH	钳制条件
0	0	钳制打开, 软件钳制有BKEN控制
0	1	打开: 当PWM不运行(PWMRUN=0)时, 由PWMnB设定PWM的输出条件
1	0	钳制打开, 当钳制脚指定后, 没有PWM输出, PWMRUN位被清除并把BKF标志位置'0'。.
1	1	保留

PWM控制寄存器 3

PWMCON3

(D7H)

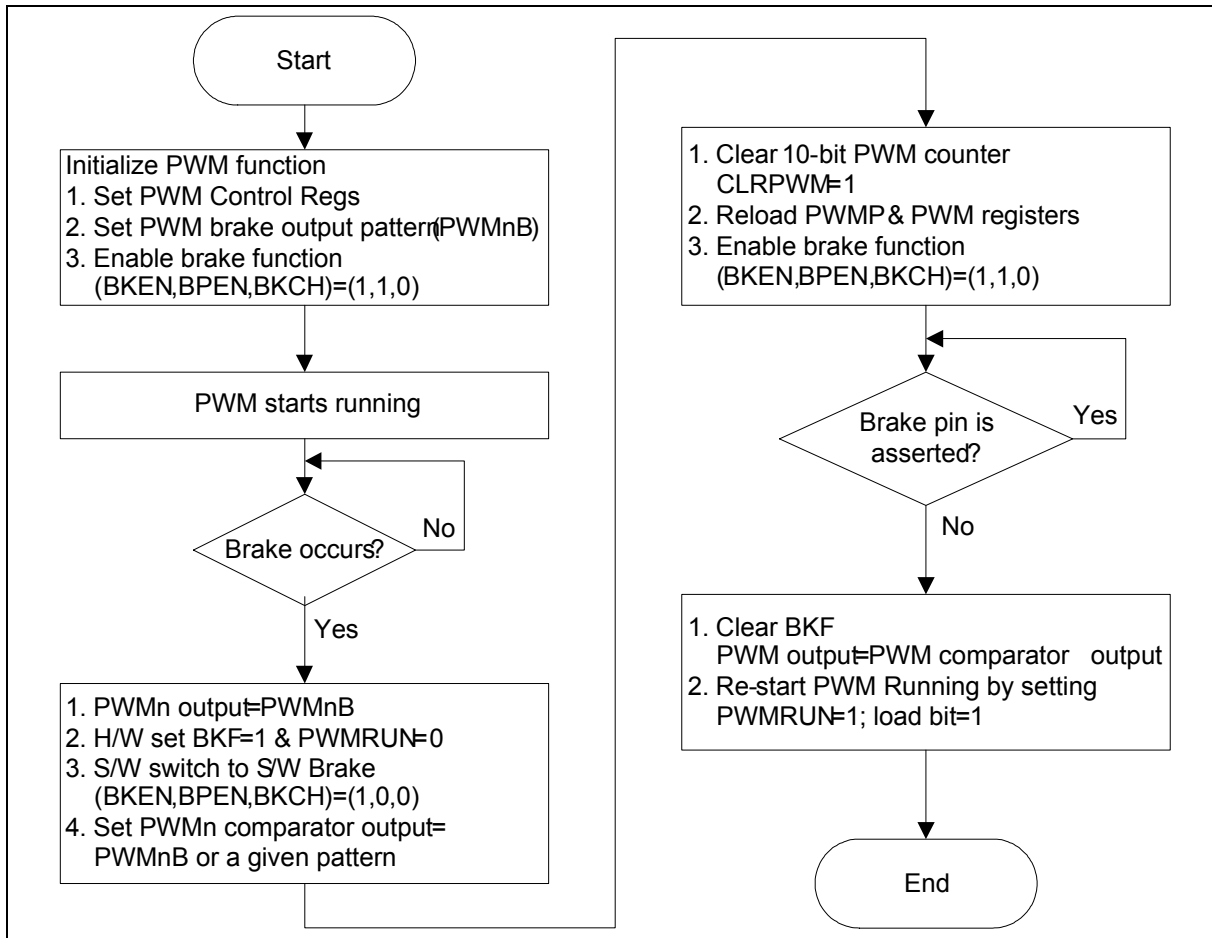
位	名称	功能
7~1	-	保留
0	BKF	外部钳制脚标志. 0: PWM不钳制. 1: PWM被外部钳制脚钳制。由软件清'0'。

N79A82X1系列支持软件或外部引脚(P0.2)钳制功能，钳制控制通过PWMCON2寄存器设定。软件钳制和外部引脚钳制设定请参考钳制条件表。钳制确定时，PWM输出由PWMnB设定，软件钳制，把BKEN置“1”将允许钳制功能，由BPEN和BKCH位设定，(BPEN, BKCH) = (0,0)，钳制确定；(BPEN, BKCH) = (0,1)，PWM输出与PWMRUN位一致；当PWM不运行时也就是说PWMRUN=0，PWM输出由PWMnB设定；当PWM正在运行时，PWMRUN = 1并保持PWM输出。

N79A82X1系列外部钳制引脚(P0.2)钳制PWM可以产生钳制中断，可以在中断服务程序或查询钳制标志(BKF)确定外部引脚是否钳制PWM。如果P0.2拉低，BKPS = 0，BKF(PWMCON3.0)将会被置“1”和PWNRUN将会被清除，PWM停止运行；在钳制脚释放后，PWM输出条件由PWMnB设定。

钳制引脚确定后，运行位将会自动清除，BKF(PWMCON3.0)标志将会自动置'1'。在用户程序里，可以通过检测PWMCON1.7或允许PWM的钳制中断来判断是不是由钳制引脚引起的钳制发生。另外的检测方法是通过钳制由钳制引脚和钳制引脚的外部中断引脚。最后的方法是，如果检测程序发现钳制的时间不够长，在引起钳制的钳制条件消失时，无论PWM处于什么状态，PWM输出在本周期内不会立即取消钳制；这样防止钳制解除后，PWM会进入混乱状态。

为了平缓的解除外部钳制引脚的钳制，PWM继续运行。步骤参见下图。



```

void PWM(void)                                // PWM输出设置，以PWM0为例
{
    PWMCON1 |= 0x40;                          // 设定自动装载计数值
    PWMPH = 0x03;                            // 设定计数器寄存器初值为0X3FF
    PWMPL = 0xff;
    PWM0H = 0x01;                            // 设定比较寄存器初值
    PWM0L = 0xff;
    PWMCON1 |= 0x80;                          // PWM运行
    while(1);
}
    
```

23 模拟数字转换器

N79A82X1系列有一个ADC包含有DAC、比较器、逐次比较寄存器和逐次比较控制器。DAC是把逐次比

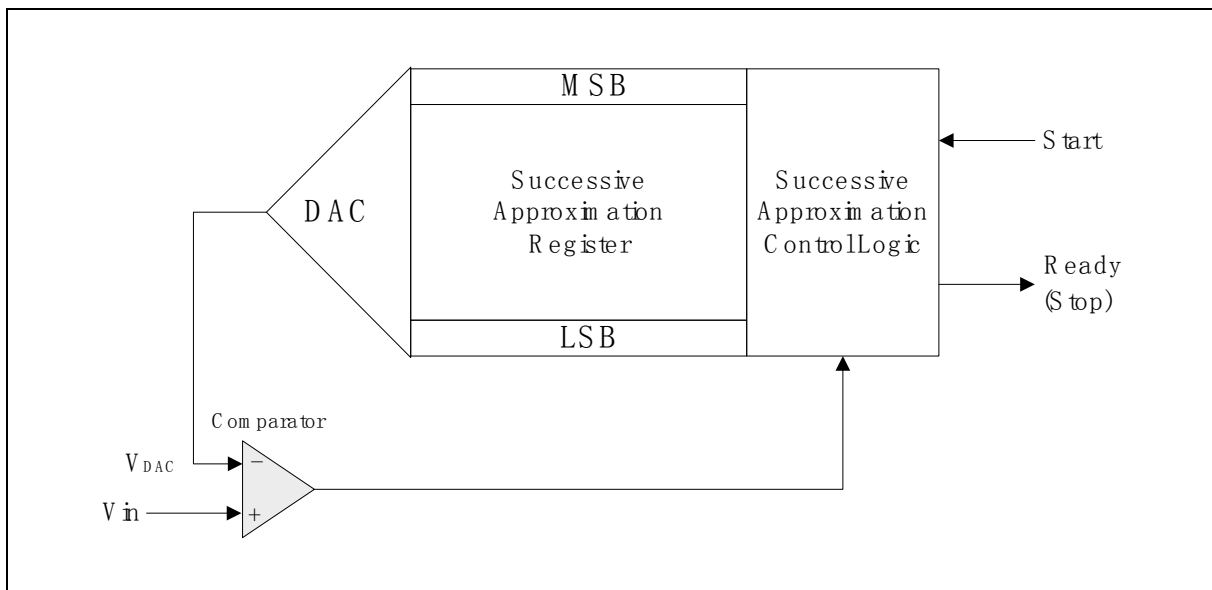
较寄存器的内容转换成电压(VDAC)；比较器是把模拟输入电压(Vin)和(VDAC)相比较并回馈到逐次比较控制器。把ADCCON寄存器中的ADCS位设为‘1’，就开始了一次转换。ADCS可以由软件置位也可以由硬件或软件置位

在使用ADC电路前，必须把ADCCON.5 (ADCEX)置‘1’打开ADC电路。当ADCCON.5 (ADCEX) = 0时，是软件开始模式，把控制位ADCCON.3 (ADCS)置‘1’即可以开始ADC转换。当ADCCON.5 = 1时，是软件或硬件开始模式，ADCCON.3 (ADCS)置‘1’或STADC上的有效上升沿开始ADC转换。当使用STADC上的有效上升沿开始ADC转换时低电平至少保持一个机器周期，之后的高电平也至少保持一个机器周期。

STADC由低电平到高电平转换是在机器周期末采样的，转换开始是在下一个周期初开始的。

控制位ADCCON.4 (ADCI)置‘1’标志10-位转换的结束。转换结果的高8位存放在特殊功能寄存器ADCH中，剩下的两位存放在ADCCON.7 (ADC.1)和ADCCON.6 (ADC.0)中。用户可以忽略ADCCON中的最低两位，作为8-位的ADC使用 (高8位存放在ADCH中)。在任何情况下，总的转换时间共有50个机器周期。ADC 时把ADCS状态标志置‘1’，并在50时钟周期后清‘0’。

控制位ADCCON.0、ADCCON.1被用作4选1多路模拟开关控制。ADC转换步骤不受外部或软件开始转换的影响。转换结束的结果也不会影响ADCI = 逻辑1；ADC在转换过程中系统将不能进入空闲或掉电模式。转换结束结果 (ADCI = 逻辑 1) 仍然不影响空闲模式



逐次逼近式ADC

23.1 ADC的分辨率和仿真电源

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。

辅助功能寄存器1

位:	7	6	5	4	3	2	1	0
	KBF	BOD	BOI	LPBOV	SRST	ADCEN	0	DPS

助记符: AUXR1

地址: A2h

2	ADCEN	0: 禁止 ADC 电路. 1: 允许ADC 电路.
---	-------	-------------------------------

中断允许

位:	7	6	5	4	3	2	1	0
	EA	EADC	EBO	ES	ET1	EX1	ET0	EX0

助记符: IE

地址: A8h

6	EADC	允许ADC中断.
---	------	----------

ADC 控制寄存器

位:	7	6	5	4	3	2	1	0
	ADC.1	ADC.0	ADCEX	ADCI	ADCS	RCCLK	AADR1	AADR0

助记符: ADCCON

地址: E1h

位	名 称	功 能
7	ADC.1	ADC 结果的第1位。
6	ADC.0	ADC 结果的第0位。
5	ADCEX	=‘0’，只有软件把ADCS 位置‘1’才开始ADC转换。 =‘1’，软件把ADCS 位置‘1’或外部STADC（1.4）上的下降沿开始ADC转换。
4	ADCI	ADC 中断标志位。当ADC转换结束，转换结果可以读时， ADCI标志位置‘1’。若ADC中断使能，就可以进入ADC中断，进入中断后该标志位清‘0’，也可以软件清‘0’但不能软件置‘1’。若该标志位为‘1’时，就无法开始新的ADC转换。

3	ADCS	ADC 开始和状态标志位。把ADCS置‘1’开始一次ADC转换，可以由软件或外部的STADC信号置‘1’，当ADC忙时ADCS=‘1’。转换结束，ADCI置位后ADCS=‘0’。如果ADCS=‘1’或ADCI=‘1’时无法开始新的ADC转换		
		ADCI	ADCS	ADC 状态
		0	0	ADC 空闲，可以开始一个新的转换
		0	1	ADC 忙，转换中
		1	0	转换结束，开始新的转换时，要求ADCI=0
		1	1	转换结束，开始新的转换时，要求ADCI=0
		如果软件在设ADCI=0的时候，ADCS=1，那么在同一通道上的新的A/D转换马上开始。但推荐先ADCI=0，再ADCS=1。		
2	RCCLK	0: CPU 时钟作为 ADC 时钟. 1: 内部 RC 时钟作为 ADC 时钟.		
1	AADR1	见下表。		
0	AADR0	见下表。		

AADR1, AADR0: ADC仿真输入通道选择位:

ADC仿真输入通道选择位。只有当ADCI=0和ADCS=0时才可以改变这些位。

AADR1	AADR0	选择仿真输入通道
0	0	AD0 (P0.3)
0	1	AD1 (P0.4)
1	0	AD2 (P0.5)
1	1	AD3 (P0.6)

ADC 转换结果寄存器

位:	7	6	5	4	3	2	1	0
	ADC.9	ADC.8	ADC.7	ADC.6	ADC.5	ADC.4	ADC.3	ADC.2

助记符: ADCH

地址: E2h

位	名称	功能
7~0	ADC.9 ~ADC.2	ADC转换结果

AUXR1.2 (ADCEN) =1打开ADC 的电路，AUXR1.2 (ADCEN) =0关闭ADC 的电路。

```
void EnableADC()
{
    AUXR1|=0x04; //打开ADC 的电路，ADC 的电路开始工作
}
void DisableADC()
{
    AUXR1&=0xFB; // 关闭ADC 的电路，ADC 的电路停止工作
}
```

```

}
void ADC_Select(unsigned char ADC_C)
{
switch(ADC_C)
{
case 0:P0M1|=0x08;P0M2&=0xF7;ADCCON=0;break;
case 1:P0M1|=0x10;P0M2&=0xef;ADCCON=1; break;
case 2:P0M1|=0x20;P0M2&=0xdf;ADCCON=2; break;
case 3:P0M1|=0x40;P0M2&=0xbf;ADCCON=4; break;
default: break;
}
}

```

A-D转换

(1)、若ADCCON.5(ADCEX)=0；把ADCCON.3(ADCS)置'1'，就开始一次新的A-D转换。实例程序如下：

```

void StartADC()
{
ADCCON|=0x08;
}

```

(2)、若ADCCON.5(ADCEX)=0，STADC(P1.4)上的上升沿,就开始一次新的A-D转换。

ADC中断

(1)、打开ADC中断。

```

void EnableADC_INT()
{
EADC=1;
EA=1;
}

```

(2)、编写ADC中断服务程序。

```

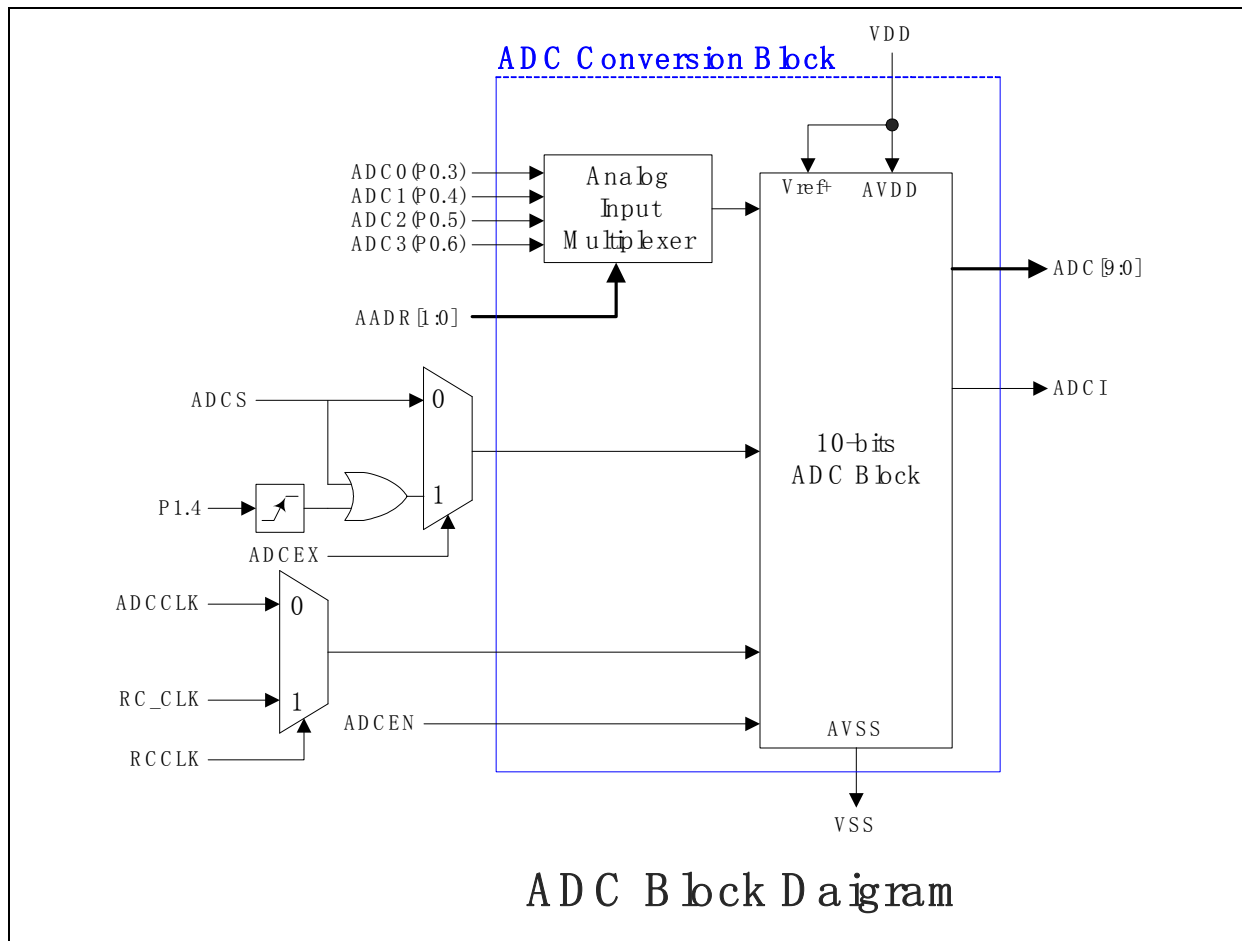
void ADC_ISR() interrupt 11
{
. . . . . //
}

```

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

下面是计算结果方程式：

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



ADC框图

24 I2C串行控制

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。主要特性如下:

- 2007. 在主机和从机之间数据双向传输
- 2007. 多总机总线(没有中心总机)
- 2007. 总线上出现两个主机同时传输数据的无冲突仲裁
- 2007. 通过串行同步时钟在同一条总线上不同的设备可以使用不同的速率通信。
- 2007. 串行同步时钟可以作为握手协议可以控制数据传输
- 2007. I2C 总线可以用作测试和侦测等目的

N79A82X1系列片内I2C逻辑提供一个串行接口,符合I2C总线规格和支持所有的传输模式(其它低速模式)。I2C逻辑实现自动字节传输。串行传输和状态寄存器(I2STA)反映I2C总线的状态。

CPU与I2C逻辑接口透过4个特殊功能寄存器: I2CON (SIO1控制寄存器), I2STA (SIO1状态寄存器), I2DAT (SIO1资料寄存器),和I2ADR (SIO1 从机地址寄存器)。SIO1逻辑与外部I2C总线的接口是透过P1引脚: P1.2/SCL (串行时钟线)和P1.3/SDA (串行数据线)。

24.1 SIO1 端口

SIO1端口是串行I/O端口,支持所有的I2C总线收发传输模式。SIO1端口是自动字节传输操作。I2CON的ENS1位置'1'允许该端口,CPU到SIO1端口的接口是通过下面6个特殊功能寄存器: I2CON (I2C控制寄存器, C0H), I2STATUS (状态寄存器, BDH), I2DAT (资料寄存器, BCH), I2ADDR (地址寄存器, C1H), I2CLK (时钟寄存器 BEH)和I2定时器 (地址寄存器, BFH)。SIO1到I2C总线硬件接口是透过2个引脚: SDA (P1.3,串行数据线)和SCL (P1.2,串行时钟线)。允许SIO1,必须把P1.2和P1.3输出锁存成逻辑1。

24.2 I2C控制寄存器:

N79A82X1系列需要设置几个控制寄存器来控制I2C串行口。详细描述如下:

24.2.1 地址寄存器, I2ADDR

SIO1设备地址寄存器: I2ADDR, CPU可以从中读取8-位数据或写8-位数据,它是直接寻址地址特殊功能寄存器。这些寄存器的内容在SIO1为主机模式时,不起作用。在从机模式这七位将会被作为本机地址。SIO1硬件将检测地址是否匹配。

位:	7	6	5	4	3	2	1	0
	I2ADDR.7	I2ADDR.6	I2ADDR.5	I2ADDR.4	I2ADDR.3	I2ADDR.2	I2ADDR.1	GC

助记符: I2ADDR

地址: C1h

24.2.2 资料寄存器, I2DAT

该寄存器的内容是准备发送的或刚接收的串行数据一个字节的的数据。只要不在移位处理的过程，CPU可以读写访问8-位可直接寻址的特殊功能寄存器。当SIO1的状态设定后和串行中断标志(SI)置'1'；只要SI='1'，I2DAT中的数据一直是稳定的。在数据移出的过程中，总线上的数据同时也在移动；I2DAT的内容一直是总线上出现的最后一个字节。在主机发送数据从机接收数据的模式中，不需要仲裁来保证I2DAT中的数据正确。

位:	7	6	5	4	3	2	1	0
	I2DAT.7	I2DAT.6	I2DAT.5	I2DAT.4	I2DAT.3	I2DAT.2	I2DAT.1	I2DAT.0

助记符: I2DAT

地址: BCh

移位寄存器包含I2DAT和应答位9-位，应答位由SIO1的硬件控制，CPU不能访问。I2DAT中的串行数据和应答位在在串行时钟SCL线的上升沿移出。当一个字节被移位元到I2DAT后，I2DAT中的串行资料是可以使用的，应答位(ACK或NACK)在第9个时钟返回。串行数据在每一个下降沿(SCL时钟)从I2DAT移出输出，在每一个上升沿(SCL时钟)资料移进I2DAT

24.2.3 控制寄存器, I2CON

CPU可以直接读写操作访问8-位可位寻址的特殊功能寄存器。SI位受SIO1硬件的影响；当串行中断请求和STO位被清除即条件显示在总线上，SI位被硬件置'1'。当ENS1 = "0"时，STO位被清除。

位:	7	6	5	4	3	2	1	0
	-	ENS1	STA	STO	SI	AA	-	-

助记符: I2CON

地址: C0

位	名称	功能
7	-	保留.
6	ENS1	0: 禁止I2C串行功能。SDA和SCL为高阻状态，SDA和SCL输入信号被忽视，I2C不在可寻址的从机状态；I2CON中的STO位被强行的设为“0”，其它位不受影响。P1.0(SCL)和P1.1(SDA)可以作为开漏I/O。 1: 允许I2C串行功能。P1.0和P1.1口必须保持逻辑1状态。
5	STA	开始标志。 0: STA位被清除，没有开始条件或重新开始条件的产生。 1: STA位被置'1'。I2C进入为主机模式；如果总线为空闲，I2C硬件检测I2C的总线状态和产生开始条件；如果总线不空闲，I2C等待STOP条件后产生开始条件。当I2C已经是主机模式并且已经传输或接收一个或多个字节，STA置'1'，I2C传输一个重新开始条件。任何时间都可以把STA置'1'。STA也可以在I2C是可寻址的从机模式被置'1'。

(续)

位	名称	功能
4	STO	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。
3	SI	0: 当SI标志被清除，没有串行中断请求，并且不展宽SCL在线的串行时钟。 1: SI标志由硬件置'1'，必须由软件清'0'；如果EA和ES位(在IE寄存器)被置'1'，串行中断请求，可以产生中断。SI被置'1' SCL在线的串行时钟为低的时间延长并且串行传输数据保持，在线的串行时钟为高的时间不受串行中断标志SI的影响。
2	AA	接收应答标志 0: 在下面情况下，在应答时钟脉冲下，SCL上没有应答(SDA上高电平): 1) SIO1为主机接收模式，已经接收一个数据。2) SIO1为可寻址的从机接收模式，已经接收一个数据。 1: 在下面情况下，在应答时钟脉冲下，SCL上没有应答(SDA上高电平): 1) 接收到自己的地址; 2) SIO1为主机接收模式，已经接收一个数据。3) SIO1为可寻址的从机接收模式，已经接收一个数据。
1	-	保留.
0	-	保留.

24.2.4 I2C时钟波特率位, I2CLK

当SIO1在主机模式下，I2C数据的波特率由I2CLK寄存器设定。在从机模式下是很重要的；在从机模式下，SIO1将自动与主机I2C设备时钟频率同步，可高达400 KHz。

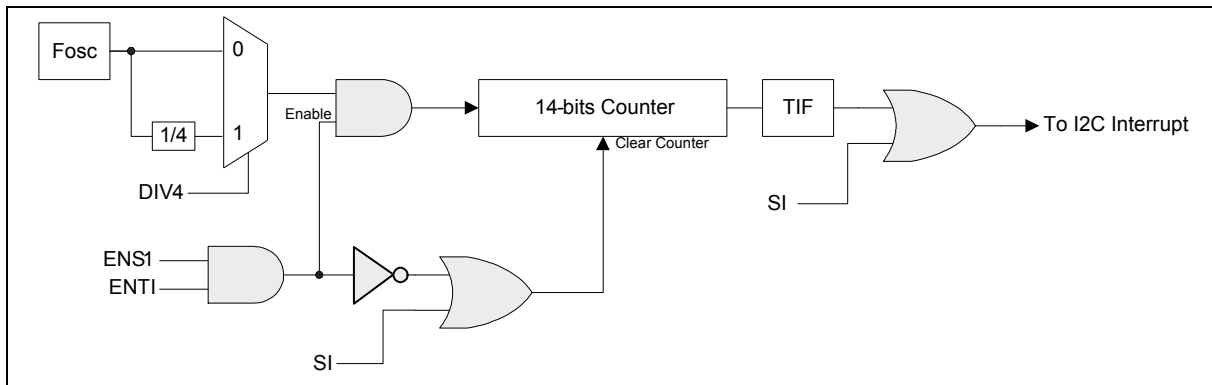
I2C数据波特率设定是：I2C的数据波特率 = $F_{cpu} / (I2CLK + 1)$ 。 $F_{cpu} = F_{osc} / 4$ ，如果 $F_{osc} = 16\text{MHz}$ ， $I2CLK = 40(28H)$ ，I2C的数据波特率 $I2C = 16\text{MHz} / (4 \times (40 + 1)) = 97.56\text{K位/秒}$ 。结构框图如下：

位:	7	6	5	4	3	2	1	0
	I2CLK.7	I2CLK.6	I2CLK.5	I2CLK.4	I2CLK.3	I2CLK.2	I2CLK.1	I2CLK.0

助记符: I2CLK

地址: Beh

位	名称	功能
7 ~ 0	I2CLK	时钟波特率设置位.



24.2.5 状态寄存器, I2STATUS

I2STATUS是一个8-位只读寄存器。低3位一直为0。其余是状态码。有23个可能的状态码，当I2STATUS的内容是F8H，没有串行中断请求。所有的其它I2STATUS值对应一个SIO1状态。当每一个进入状态，就会产生状态中断请求(SI = 1)。在SI被硬件置'1' 1个机器周期后或在SI被软件清除之后，有效状态码出现在I2STATUS中。

另外，00H状态表示总线错误。总线错误发生在START或STOP条件出现在帧结构不正确的位置。不正确的位置比如是在串行传输地址字节、数据字节或应答位期间。

24.3 I2C的操作模式

4种操作模式：主机/传输，主机/接收，从机/传输和从机/接收。I2CON中的STA，STO和AA置位后，SI被清除后，SIO1 硬件才开始下一个动作，当一个动作结束，I2STATUS将会被更新并且同时SI被硬件置'1'。之后会进入中断服务程序(如果SI中断允许)，新状态码可以被用为软件判断哪一个中断进入中断服务程序。

主机/传输模式：

- 主机产生开始信号，即置位STA。
- 主机向总线传输从机地址（一般为7位）+W（1），等待接收应答（ACK）信号。
- 接收到应答信号后，向被寻址到的从机发送数据，等待ACK信号。
- 数据发送完毕后发送结束信号。即置位STO。

补充：对Flash的固定页码或存储单元进行写操作时要在B步骤的后面再次发送存储单元的地址，然后再发送数据。

代码示例如下：

```
I2CON = 0xE4;           //置位STA,置位ACK
while(!SI) _nop_();     // 等待 ACK
I2DATA = Slave_Address+1; // 从机地址+写
I2CON = 0xC4;           //置位ACK
while(!SI) _nop_();
I2DATA = Send_Data;     // 给从机发送数据
while(!SI) _nop_();
I2CON = 0xD4;           // 结束,置位 STO
```

主机/接收模式：

从机为微控制器

- A. 主机产生开始信号，即置位STA。
- B. 向总线传输从机地址（一般为7位）+R（0），等待接收应答（ACK）信号。
- C. 接收到应答信号后，主机接收从机发送的数据，等待ACK信号。
- D. 数据发送完毕后发送结束信号。即置位STO。

代码示例如下：

```
I2CON = 0xE4;           //置位STA, 置位ACK
while(!SI) _nop();      //等待 ACK
I2DATA = Slave_Address+0; //从机地址 + 读
I2CON = 0xC4;           //置位ACK
while(!SI) _nop();
Receive_Data = I2DATA;   // 接收到来自从机的数据
while(!SI) _nop();
I2CON = 0xD4;           //结束,置位 STO
```

从机为存储器等对固定存储单元的读操作:

- A. 主机产生开始信号，即置位STA。
- B. 主机向总线传输从机地址（一般为7位）+W（1），等待接收应答（ACK）信号。
- C. 主机向总线发送所要读取的存储单元地址，等待接收应答（ACK）信号。
- D. 主机再次启动总线，发送开始信号，置位STA。
- E. 主机向总线传输从机地址（一般为7位）+R（0），等待接收应答（ACK）信号。
- F. 接收到应答信号后，主机接收从机发送的数据，等待ACK信号。
- G. 数据发送完毕后发送结束信号。即置位STO。

代码示例如下：

```
I2CON = 0xE4;           //置位STA, 置位ACK
while(!SI) _nop();      //等待 ACK
I2DATA = Slave_Address+1; // 从机地址 + 写
I2CON = 0xC4;           // 置位 ACK
while(!SI) _nop();      //等待ACK
I2DATA = Memory_Address; // 发送存储器地址
I2CON = 0xC4;           // 置位 ACK
while(!SI) _nop();      //等待 ACK
I2CON = 0xE4;           //重新发送
I2DATA = Slave_Address+0; // 从机地址 + 读
I2CON = 0xC4;           //置位 ACK
while(!SI) _nop();      //等待 ACK
Receive_Data = I2DATA;   //接收到来自从机的数据
while(!SI) _nop();
I2CON = 0xD4;           //结束,置位 STO
```

从机/传输模式:

- A. 设定自身的寻址地址，即从机地址，对 I2ADDR 寄存器进行设定。
- B. 被主机寻址到，即接收到总线传输的地址寻址信号与从机自身地址相符，此时发送应答信号 ACK。
- C. 向总线发送数据+ACK。
- D. 发送结束，由主机停止总线。

代码示例如下：

```
I2ADDR = SlaveOwn_Address;// 设定自身的寻址地址
I2CON = 0xC4;           //置位 ACK
while(!SI) _nop_();      //等待主机信号
I2DATA = Send_Data;      // 给主机发送数据
I2CON = 0xC4;           //置位 ACK
while(!SI) _nop_();
```

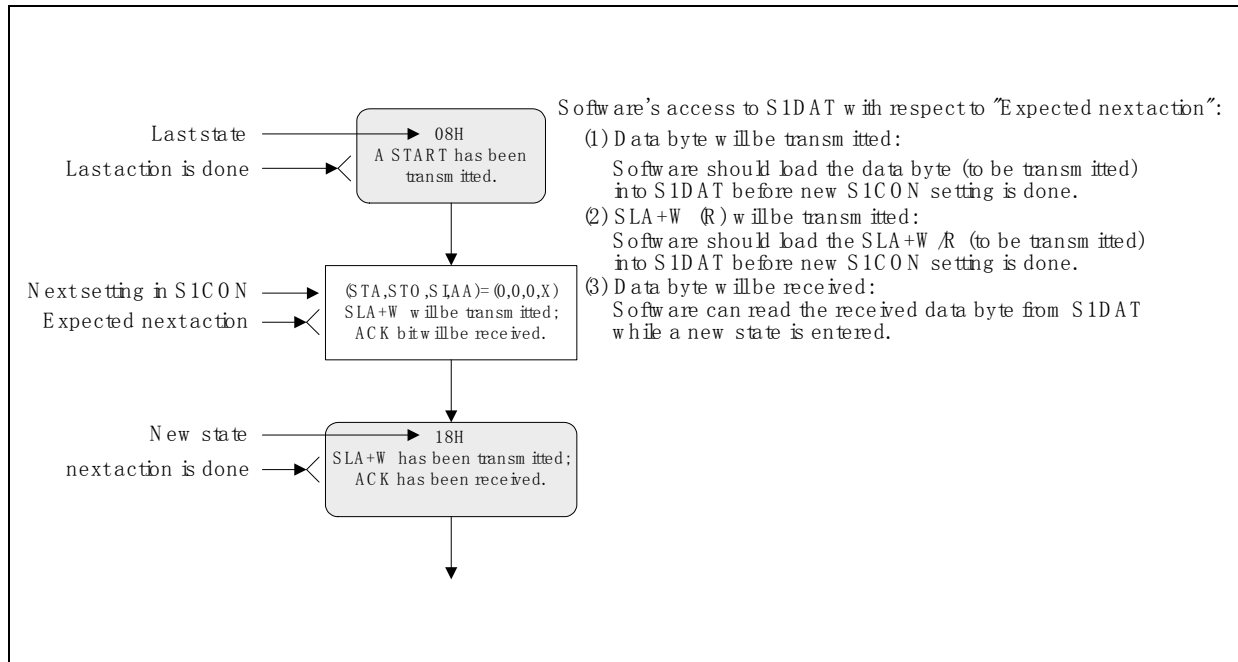
从机/接收模式：

- A. 设定自身的寻址地址，即从机地址，对 I2ADDR 寄存器进行设定。
- B. 被主机寻址到，即接收到总线传输的地址寻址信号与从机自身地址相符，此时发送应答信号 ACK。
- C. 向总线发送自身从机地址+R（0）。
- D. 接收主机发送的数据，并给出应答信号 ACK。
- E. 接收结束，由主机停止总线。

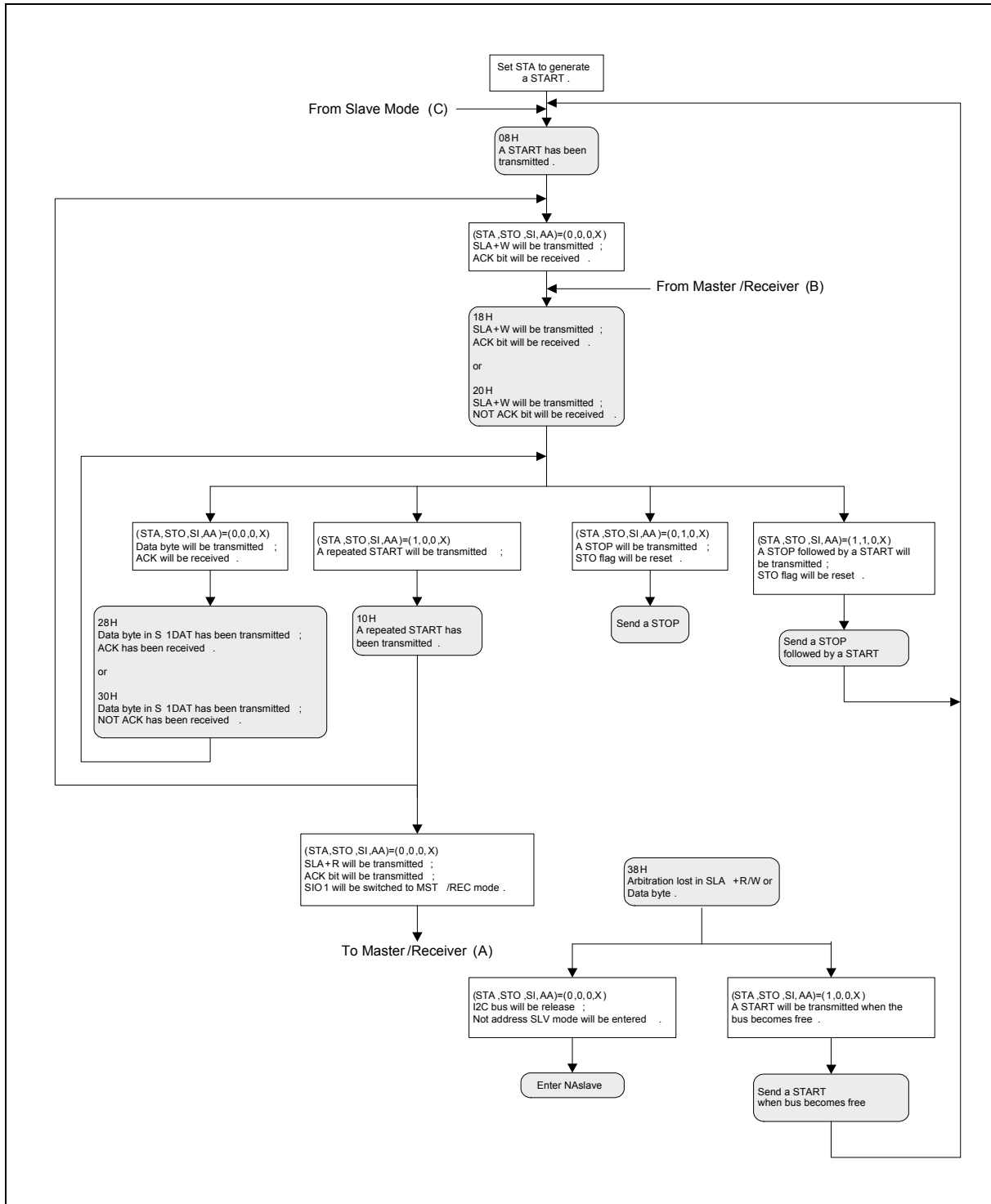
代码示例如下：

```
I2ADDR = SlaveOwn_Address;// 设定自身的寻址地址
I2CON = 0xC4;           //置位 ACK
while(!SI) _nop_();      //等待主机信号
I2DATA = SlaOwn_Address+0;// 向总线发送自身从机地址+读
I2CON = 0xC4;           //置位 ACK
while(!SI) _nop_();      // 等待ACK
Receive_Data = I2DATA;    //接收到来自主机的数据
I2CON = 0xC4;           //置位 ACK
while(!SI) _nop_();
```

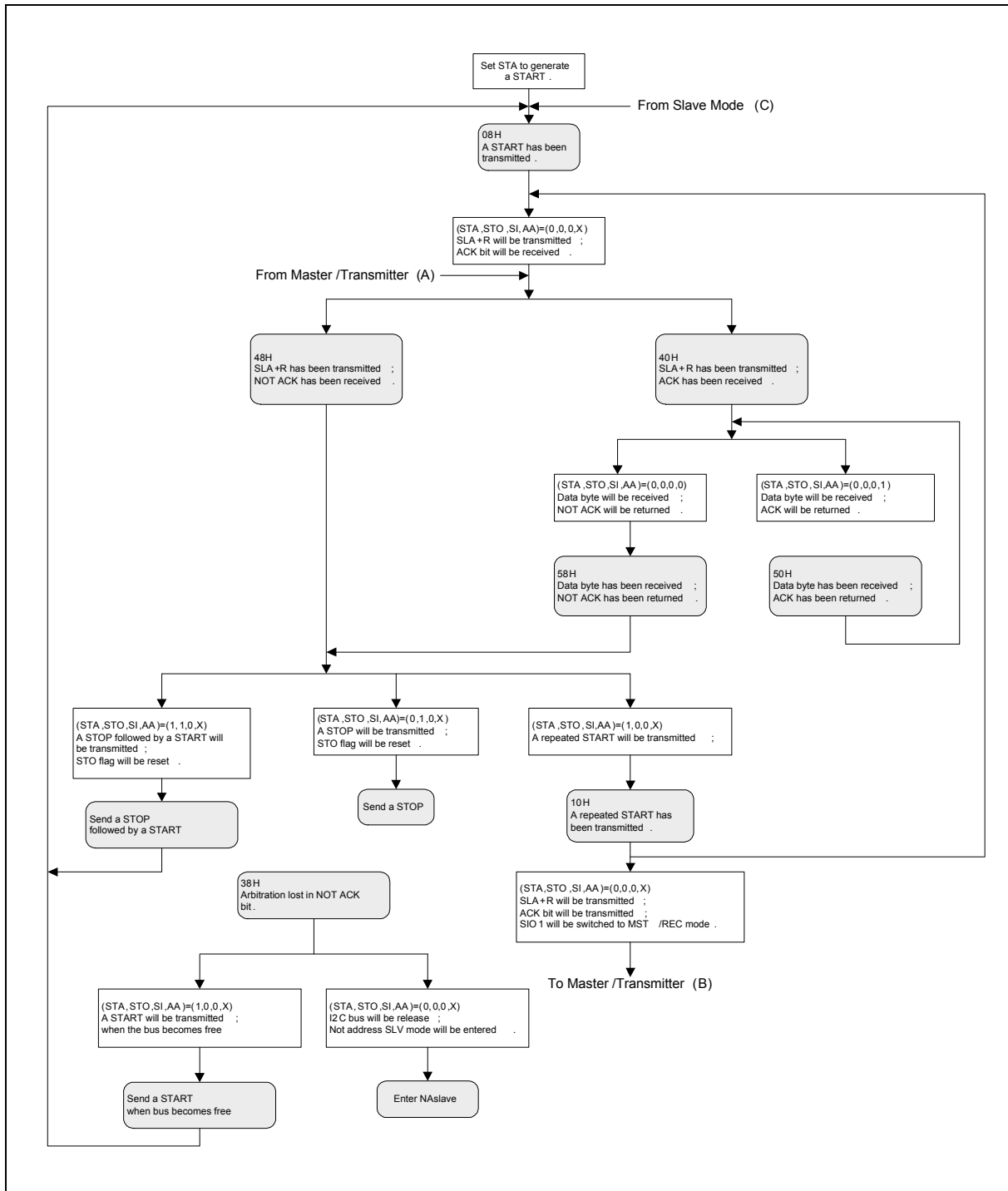
在各种模式下数据传输结构如下图：



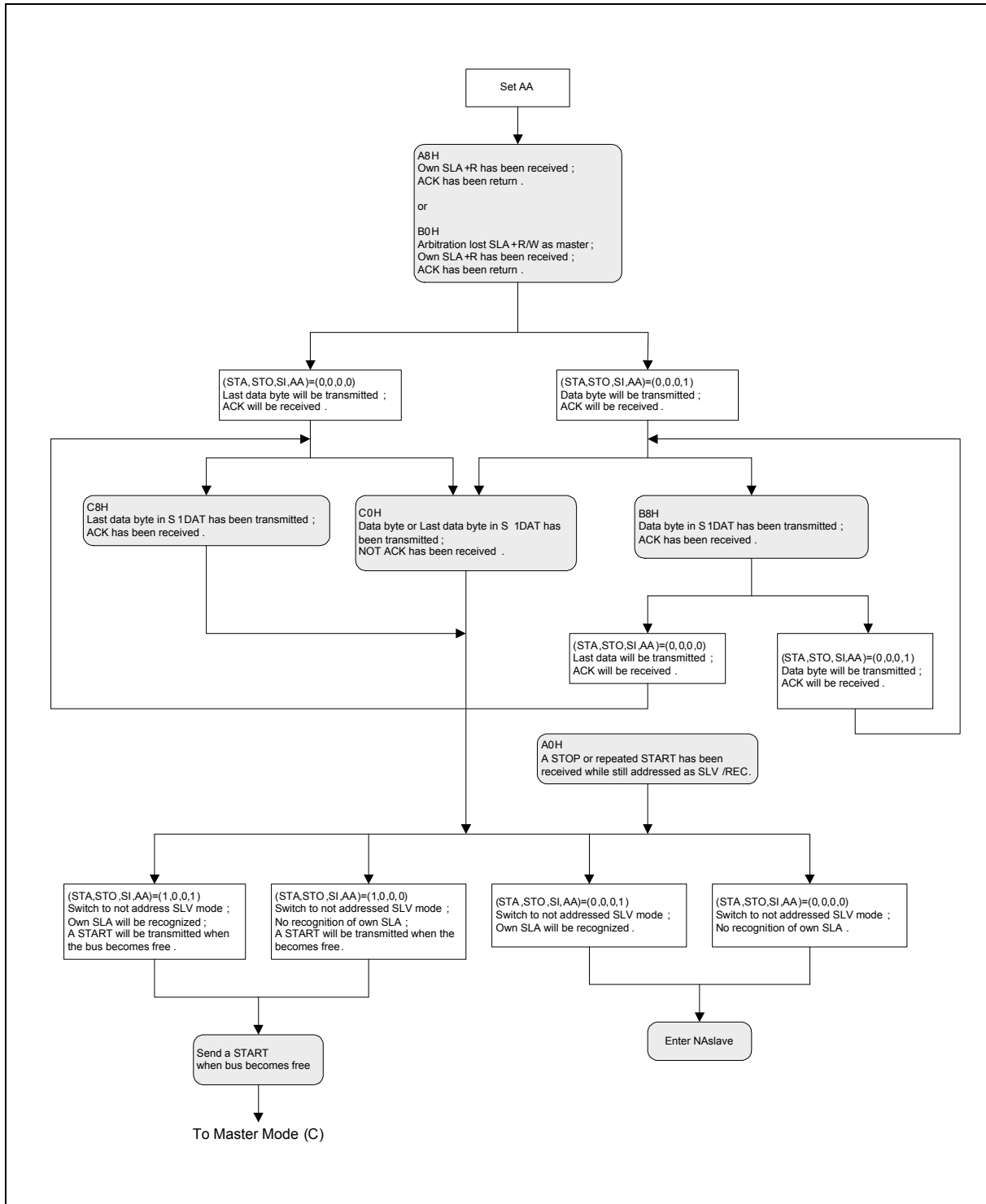
24.3.1 主传输模式



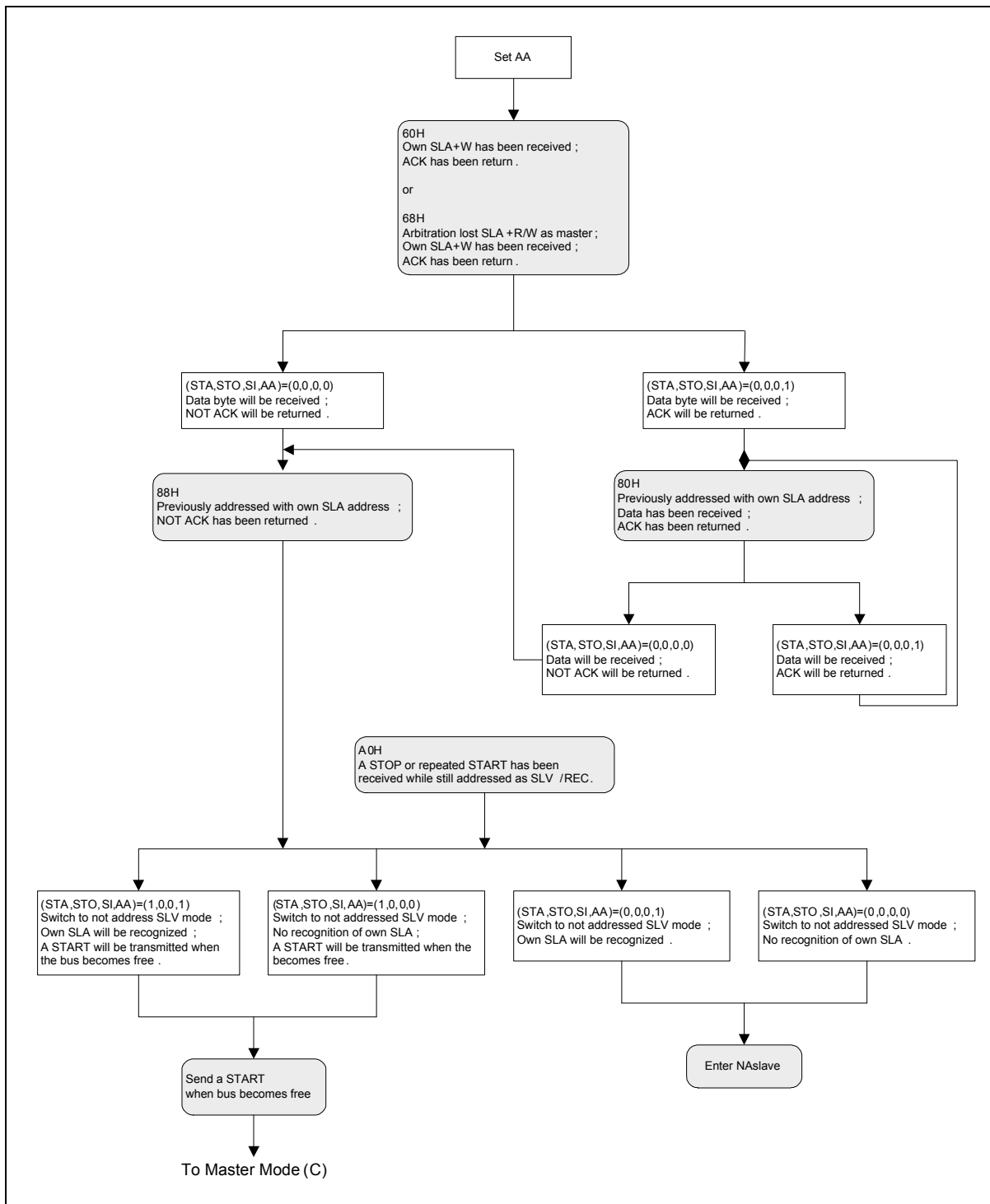
24.3.2 主传接收模式



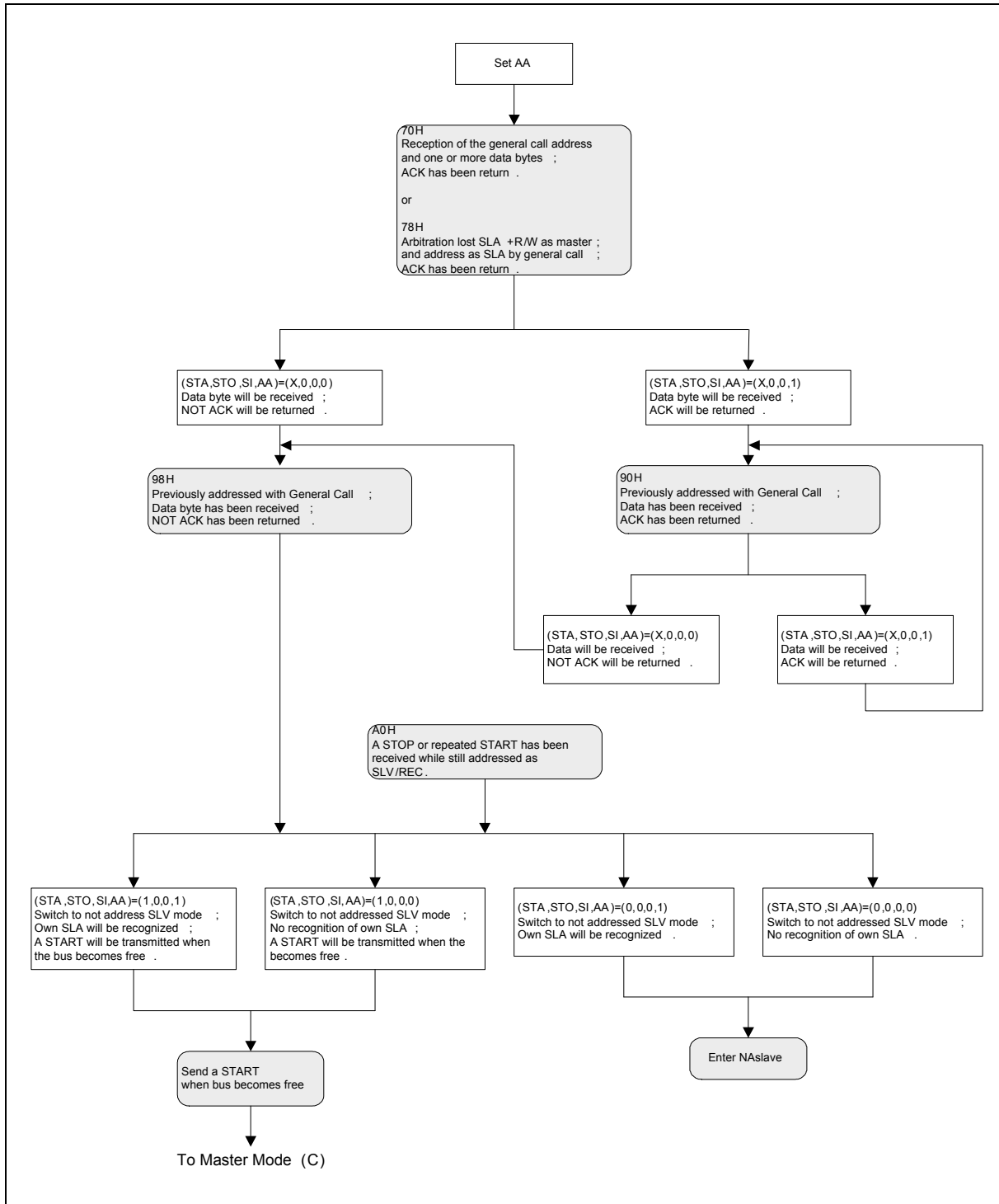
24.3.3 从传输模式



24.3.4 从接收模式



24.3.5 GC模式



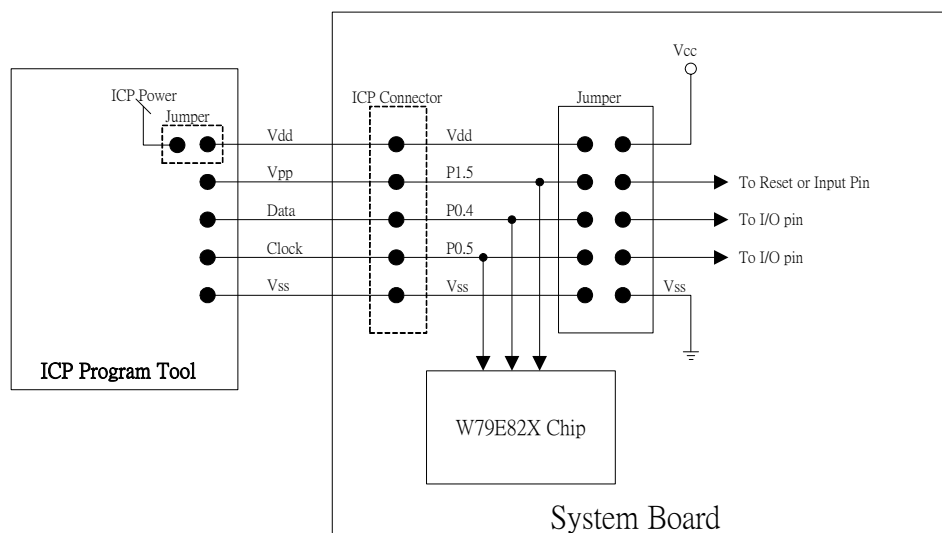
25 ICP(在电路编程) FLASH 编程

出厂时N79A82X1系列中的程序存储空间是空的，在使用之前必须通过普通烧写器或ICP(在电路编程)工具把程序写入内存

使用ICP工具，用户必须在设计应用电路的时候注意P1.5，P0.4和P0.5以供ICP工具使用，如下图。在使用ICP工具烧录程序时，会被拉高(电压~10.5V)，并且保持这样高的电压来更新代码、数据和配置，在编程结束后P1.5上的高电压解除。因此使用ICP工具烧写程序时，建议关闭系统板上的电源，在烧录结束时再上电。

在进入ICP编程模式后，所有的引脚都被设置成准双向模式，并且输出“1”电平。

N79A82X1系列支持两个Flash EPROM区域，分别为：16K/8K/4K/2K字节的AP Flash EPROM和256/128字节NVM数据存储器。在ICP模式下可以分别更新AP Flash EPROM中的代码或256/128字节的NVM 数据。



注：1更新代码时，P1.5, P0.4和P0.5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

26 配置位

N79A82X1系列有2个配置位(CONFIG1, CONFIG2), 这2个配置位必须在上电之前设置, 一旦程序开始执行就不能改了。这些特性是通过2个flash EPROM 字节来完成配置, 这2个flash EPROM可以编程和校验。在代码编程结束后, 代码可以加密, 详细描述如下, 这2个字节的数据使用MOVC指令从它所在的地址读取

26.1 CONFIG1

配置 1 :

7	6	5	4	3	2	1	0	
W DTE	RPD	PRHI	BOV	-	-	Fosc1	Fosc0	配置位

W DTE:W DT时钟源选择位

RPD:复位脚禁止位

PRHI:端口复位时高低位电平

BOV:欠电压选择位

Fosc1:CPU 振荡器类型选择位1

Fosc0:CPU 振荡器类型选择位0

配置寄存器1

位	名称	功能
7	WDTE	看门狗定时器的时钟源选择位: 0: 保留 1: CPU时钟作为看门狗定时器时钟。
6	RPD	复位脚禁止位: 0: 允许P1.5脚的复位功能. 1: 禁止P1.5脚的复位功能, 作为输入引脚
5	PRHI	端口复位时高低位电平: 0: 端口复位时在低状态. 1: 端口复位时在高状态.
4	BOV	欠压电压选择位: 0: 欠压检测电压为3.8V. 1: 欠压检测电压为2.5V.
3	-	保留.
2	-	保留.
1	Fosc1	CPU 振荡器类型选择位1
0	Fosc0	CPU 振荡器类型选择位0

振荡器配置位:

FOSC1	FOSC0	振荡器
0	0	4MHz ~ 20MHz 晶振
0	1	保留
1	0	保留
1	1	外部振荡器时钟输入

26.2 CONFIG2

配置 2 :

7	6	5	4	3	2	1	0	
C7	C6	-	-	-	-	-	-	配置位

C7: 8K/4K/2K Flash EPROM 加密位

C6: 256 字节数据加密位

配置寄存器2

C7: 16K/8K/4K/2K字节Flash EPROM 加密位

此位是用来保护用户在N79A82X1系列中的程序代码。在完成编程和校验操作后，设置此位。一旦该位设置为0，就无法再对Flash EPROM的数据和特殊设置寄存器进行访问。

C6: 256/128字节数据 Flash EPROM 加密位

此位是用来保护用户在N79A82X1系列中的256/128字节的数据代码。在完成编程和校验操作后，设置此位。一旦该位设置为0，使用烧写工具就无法再对Flash EPROM的数据和特殊设置寄存器进行访问。

位 7	位 6	功能描述
1	1	16K/8K/4K/2K字节程序代码和256/128字节数据区域都不加密，可以由烧写器、ICP读写。
0	1	16K/8K/4K/2K字节程序代码加密，由烧写器、ICP也都不可以，256/128字节数据区域不加密可以读写。
1	0	不支持
0	0	16K/8K/4K/2K字节程序代码和256/128字节数据区域都加密，烧写器、ICP都不可擦写。

27 特殊功能寄存器细部列表描述

端口0

位:	7	6	5	4	3	2	1	0
	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0

助记符: P0

地

址: 80h

端口0是一个开漏双向I/O口。

堆栈指针

位:	7	6	5	4	3	2	1	0
	SP.7	SP.6	SP.5	SP.4	SP.3	SP.2	SP.1	SP.0

助记符: SP

地

址: 81h

堆栈指针存储暂存RAM中堆栈的起始地址，就是说他总指向栈顶。

数据指针低字节

位:	7	6	5	4	3	2	1	0
	DPL.7	DPL.6	DPL.5	DPL.4	DPL.3	DPL.2	DPL.1	DPL.0

助记符: DPL

地

址: 82h

16位数据指针的低字节。

数据指针高字节

位:	7	6	5	4	3	2	1	0
	DPH.7	DPH.6	DPH.5	DPH.4	DPH.3	DPH.2	DPH.1	DPH.0

助记符: DPH

地

址: 83h

16位数据指针的高字节

电源控制

位:	7	6	5	4	3	2	1	0
	SMOD	SMOD0	BOF	POR	GF1	GF0	PD	IDL

助记符: PCON

地址: 87h

位	名称	功能
7	SMOD	1: 使串行口在模式1, 2, 3下的波特率加倍
6	SMOD0	0: SCON.7表示一个贞错误它是FE（贞错误）标志 1: SCON.7的功能与标准8052中SCON.7相同.
5	BOF	0: 软件清零 1: 当发生上电复位、降压复位、降压中断时硬件置位。
4	POR	0: 软件清零。 1: 当发生上电复位时硬件置位。
3	GF1	通用的标志位
2	GF0	通用的标志位
1	PD	1: 系统进入掉电模式；该模式下，所有时钟停止工作，程序也不再执行
0	IDL	1: 系统进入空闲模式；该模式下，CPU的时钟停止工作，程序停止运行；但串口、定时器、中断的时钟没有停止，这些功能模块仍正常运行

定时器控制

位:	7	6	5	4	3	2	1	0
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

助记符: TCON

地址: 88h

位	名称	功能
7	TF1	定时器1溢出标志；在定时器1溢出时该位置1。当程序响应定时器1中断执行相应的中断服务程序时，该位自动清0。软件也可对该位置位或复位
6	TR1	定时器1启动控制：该位由软件来置位或清零来启动或关闭定时器
5	TF0	定时器0溢出标志；在定时器1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
4	TR0	定时器0启动控制：该位由软件来置位或清零来启动或关闭定时器
3	IE1	外部中断1标志；当 $\overline{\text{int1}}$ 上出现电平跳变时由硬件置1；若被设置为下沿触发中断，进入中断服务程序IE1会自动清除为0。.
2	IT1	1触发方式控制；1：低电平边沿触发； 0：低电平触发
1	IE0	外部中断0标志；当 $\overline{\text{INT0}}$ 上出现电平跳变时由硬件置1；若被设置为下沿触发中断，进入中断服务程序IE0会自动清除为0
0	IT0	外部中断0触发方式控制；1：低电平边沿触发； 0：低电平触发

定时器模式控制

位:	7	6	5	4	3	2	1	0
	GATE	C/T	M1	M0	GATE	C/T	M1	M0

助记符: TMOD

地址: 89h

位	名称	功能
7	GATE	门控位为1时, 定时器/计数器的运行除受TRx控制外还受 $\overline{\text{int } n}$ 控制, 当TRx和 $\overline{\text{int } n}$ 均为1时定时器/计数器开始运行。该位为0时, 定时器的运行只受TRx的控制
6	$\overline{\text{C/T}}$	定时器/计数器工作方式选择: 为0时以定时器的方式运行; 为1时对TX脚上的高到低电平变化进行计数
5	M1	模式选择位
4	M0	模式选择位
3	GATE	门控位为1时, 定时器/计数器的运行除受TRx控制外还受 $\overline{\text{int } n}$ 控制, 当TRx和 $\overline{\text{int } n}$ 均为1时定时器/计数器开始运行。该位为0时, 定时器的运行只受TRx的控制
2	$\overline{\text{C/T}}$	定时器/计数器工作方式选择: 为0时以定时器的方式运行; 为1时对TX脚上的高到低电平变化进行计数
1	M1	模式选择位
0	M0	模式选择位

M1, M0: 模式选择位:

M1	M0	模式
0	0	模式 0: 8-位定时器, 有5位的预分频。
0	1	模式 1: 16-位定时器, 没有5位的预分频。
1	0	模式 2: 8位从THx中自动重装定时器
1	1	模式3: (仅适用于T0) TL0是受定时器0控制的8位定时器/计数器。TH0是受定时器1控制的8位定时器/计数器。定时器1在此方式下不工作。

定时器0低字节

位:	7	6	5	4	3	2	1	0
	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0

助记符: TL0

地

址: 8Ah

TL0.7-0: 定时器 0低字节

定时器1低字节

位:	7	6	5	4	3	2	1	0
	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0

助记符: TL1

地

址: 8Bh

TL1.7-0: 定时器 1低字节

定时器0高字节

位:	7	6	5	4	3	2	1	0
	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0

助记符: TH0

地

址: 8Ch

TH0.7-0: 定时器 0高字节

定时器1高字节

位:	7	6	5	4	3	2	1	0
	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0

助记符: TH1

地

址: 8Dh

TH1.7-0: 定时器1高字节

时钟控制

位:	7	6	5	4	3	2	1	0
	-	-	-	T1M	T0M	-	-	-

助记符: CKCON

地

址: 8Eh

位	名称	功能
7~5		保留
4	T1M	定时器1时钟选择: 0: 定时器1的时钟选择为1/12系统时钟。 1: 定时器1的时钟选择为1/4系统时钟。
3	T0M	定时器 0 时钟选择: 0: 定时器 0的时钟选择为1/12系统时钟。 1: 定时器 0的时钟选择为1/4系统时钟。

2~0	保留
-----	----

端口1

位:	7	6	5	4	3	2	1	0
	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0

助记符: P1

地

址: 90h

P1.7–0：通用数字输入/输出端口。大多数指令可以对这个端口进行读操作，在读-修改-写的指令中可以读出端口信息，管脚复用功能如下：

位	名称	功能
7	P1.7	PWM2输出
6	P1.6	PWM 1输出
5	P1.5	RST引脚和输入引脚复用
4	P1.4	INT1中断
3	P1.3	INT0中断和I ² C的SDA 复用
2	P1.2	定时器0和I ² C的SCL复用
1	P1.1	串行口的RXD
0	P1.0	串行口的TXD

分频器时钟

位:	7	6	5	4	3	2	1	0
	DIVM.7	DIVM.6	DIVM.5	DIVM.4	DIVM.3	DIVM.2	DIVM.1	DIVM.0

助记符: DIVM

地

址: 95h

DIVM 寄存器uC的时钟分频器，具体描述参见振荡器章节。

串行口控制

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

助记符: SCON

地址: 98h

位	名称	功能																														
7	SM0/FE	串行口0，模式0控制位或贞错误标志位。PCON特殊功能寄存器中的SMOD0位决定该位的功能。下面会描述SM0的运行功能。当用作贞错误标志时，该位的置位表示一个无效的停止位。该位必须由软件来清除																														
6	SM1	串行口模式位1 <table><tr><th>SM0</th><th>SM1</th><th>模式</th><th>说明</th><th>数据长</th><th>波特率</th></tr><tr><td>0</td><td>0</td><td>0</td><td>同步</td><td>8</td><td>时钟的4或12分之一</td></tr><tr><td>0</td><td>1</td><td>1</td><td>异步</td><td>10</td><td>可变</td></tr><tr><td>1</td><td>0</td><td>2</td><td>异步</td><td>11</td><td>时钟的64或32分之一</td></tr><tr><td>1</td><td>1</td><td>3</td><td>异步</td><td>11</td><td>可变</td></tr></table>	SM0	SM1	模式	说明	数据长	波特率	0	0	0	同步	8	时钟的4或12分之一	0	1	1	异步	10	可变	1	0	2	异步	11	时钟的64或32分之一	1	1	3	异步	11	可变
SM0	SM1	模式	说明	数据长	波特率																											
0	0	0	同步	8	时钟的4或12分之一																											
0	1	1	异步	10	可变																											
1	0	2	异步	11	时钟的64或32分之一																											
1	1	3	异步	11	可变																											
5	SM2	多机通信控制。将该位置1，则使能模式2及模式3下的多机通信功能。在模式2或3下，如果SM2置1，那么收到的第九位数据RB8是0的话，RI将不会置位。在模式1下如果SM2置1，那么在没有收到有效的停止位前RI是不会置位的。在模式0下，SM2位控制着串行口的时钟。如果清0，那么串行口的时钟是系统时钟的12分频。这样系统就与标准8052兼容。如果该位置1，那么串行口的时钟是系统时钟的4分频，这样就加快了同步通信的速度																														
4	REN	接收使能，置1时打开串行口接收功能，否则关闭该功能																														
3	TB8	模式2和3中要被发送的第九位数据。软件可以根据需求将该位置1或清0																														
2	RB8	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：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。该位必须由软件来清除																														
0	RI	接收中断标志：模式0下该标志由硬件在接收到8位数据后置位，而在其它模式下在串行接收到停止位的中间时置位。该位必须由软件来清除																														

串行数据缓冲器

位:	7	6	5	4	3	2	1	0
	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0

助记符: SBUF

地址: 99h

位	名称	功能
7~0	SBUF	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

键盘中断

位:	7	6	5	4	3	2	1	0
	KBI.7	KBI.6	KBI.5	KBI.4	KBI.3	KBI.2	KBI.1	KBI.0

助记符: KBI

地

址: A1h

键盘中断允许.

位	名称	功能
7	KBI.7	1: 允许P0.7触发键盘中断.
6	KBI.6	1: 允许P0.6触发键盘中断.
5	KBI.5	1: 允许P0.5触发键盘中断.
4	KBI.4	1: 允许P0.4触发键盘中断.
3	KBI.3	1: 允许P0.3触发键盘中断.
2	KBI.2	1: 允许P0.2触发键盘中断.
1	KBI.1	1: 允许P0.1触发键盘中断.
0	KBI.0	1: 允许P0.0触发键盘中断.

辅助功能寄存器1

位:	7	6	5	4	3	2	1	0
	KBF	BOD	BOI	LPBOV	SRST	ADCEN	0	DPS

助记符: AUXR1

地

址: A2h

位	名称	功能
7	KBF	键盘中断标志: 1: 当引脚变低键盘中断功能被允许时, 相应的管脚变低后。必须由软件清'0'.
6	BOD	欠压检测: 0: 允许欠压检测功能. 1: 禁止欠压检测功能, 节省电源.
5	BOI	欠压中断: 0: 禁止欠压检测中断功能. 1: 禁止欠压检测引起复位, 允许欠压检测功能中断.
4	LPBOV	电源欠压检测控制: 0: 当BOD被允许, 无论在正常模式还是在掉电模式, 欠压检测功能一直处于打开状态。 1: 当BOD被允许, 当MCU进入掉电模式, BOD允许内部RC振荡器 (2MHz~0.5MHZ), 在1/16的定时时间内关闭欠压检测电路。
3	SRST	软件复位: 1:硬件复位芯片。
2	ADCEN	0: 禁止 ADC 电路. 1: 允许ADC 电路.
1	0	保留
0	DPS	Dual 数据 指针 选择 0:选择标准8051的DPTR. 1:选择DPTR1

中断允许

位:	7	6	5	4	3	2	1	0
	EA	EADC	EBO	ES	ET1	EX1	ET0	EX0

助记符: IE

址: A8h

地

位	名称	功能
7	EA	全局中断允许.允许/禁止所有的中断.
6	EADC	允许ADC中断.
5	EBO	允许欠压中断.
4	ES	允许串行端口中断.
3	ET1	允许定时器1中断.
2	EX1	允许外部中断1.
1	ET0	允许定时器0中断.
0	EX0	允许外部中断0.

从机地址

位:	7	6	5	4	3	2	1	0
	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0
	助记符: SADDR							
	地址: A9h							

位	名称	功能
7	SADDR	SADDR 中应当写入多机通信时的广播地址或是从机地址

比较器1控制寄存器

位:	7	6	5	4	3	2	1	0
	-	-	CE1	CP1	CN1	OE1	CO1	CMF1
	助记符: CMP1							
	地址: Ach							

位	名称	功能
7	-	保留
6	-	保留
5	CE1	比较器允许: 0:禁止比较器。 1:允许比较器。在CE1设为1 稳定10uS后, 比较器输出。
4	CP1	比较器正极输入选择: 0: 选择CIN1A作为比较器的正极输入; 1: 选择CIN1B 作为比较器的正极输入。
3	CN1	比较器负极输入选择: 0:选择CMPREF比较器的参考端作为比较器的负极输入; 1:选择Vref比较器的参考端作为比较器的负极输入。
2	OE1	输出允许: 1: 如果比较器允许 (CE1 = 1), 比较器的输出端连接到CMP1脚; 该输出不与CPU时钟同步。
1	CO1	比较器输出: 输出与CPU时钟同步, 允许软件读取, 当比较器被禁止(CE1 = 0)该位清零。
0	CMF1	比较器中断标志: 在比较器输出CO1的状态改变时, 该位是由硬件设置。如果比较器中断被允许且优先级有效, MCU将产生硬件中断。可以软件清零或当比较器被禁止(CE1 = 0)该位被清零。

比较器2控制寄存器

位:	7	6	5	4	3	2	1	0
	-	-	CE2	CP2	CN2	OE2	CO2	CMF2

助记符: CMP2

址: ADh

地

位	名称	功能
7	-	保留
6	-	保留
5	CE2	比较器允许: 0:禁止比较器。 1:允许比较器。在CE2设为1 稳定10uS后, 比较器输出。

4	CP2	比较器 正极输入 选择: 0: 选择CIN2A作为比较器的正极输入; 1: 选择CIN2B 作为比较器的正极输入。
3	CN2	比较器 负极输入 选择: 0:选择CMPREF比较器的参考端作为比较器的负极输入; 1:选择Vref比较器的参考端作为比较器的负极输入。
2	OE2	输出允许: 1: 如果比较器允许 (CE2 = 1), 比较器的输出端连接到CMP2脚; 该输出不与CPU时钟同步。
1	CO2	比较器 输出: 输出与CPU时钟同步, 允许软件读取, 当比较器被禁止(CE2 = 0)该位清零。
0	CMF2	比较器 中断 标志: 在比较器输出CO2的状态改变时, 该位是由硬件设置。如果比较器中断被允许且优先级有效, MCU将产生硬件中断。可以软件清零或当比较器被禁止(CE2 = 0)该位被清零。

端口0配置模式1

位:	7	6	5	4	3	2	1	0
	P0M1.7	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0
	助记符: P0M1							地
	址: B1h							

端口0配置模式2

位:	7	6	5	4	3	2	1	0
	P0M2.7	P0M2.6	P0M2.5	P0M2.4	P0M2.3	P0M2.2	P0M2.1	P0M2.0
	助记符: P0M2							地
	址: B2h							

端口1配置模式1

位:	7	6	5	4	3	2	1	0
	P1M1.7	P1M1.6	-	P1M1.4	-	-	P1M1.1	P1M1.0
	助记符: P1M1							地
	址: B3h							

端口1配置模式2

位:	7	6	5	4	3	2	1	0
	P1M2.7	P1M2.6	-	P1M2.4	-	-	P1M2.1	P1M2.0
	助记符: P1M2							地
	址: B4h							

端口-配置模式3

位: 7 6 5 4 3 2 1 0

-	P1S	P0S	-	T1OE	T0OE	-	-
---	-----	-----	---	------	------	---	---

助记符: P2M1

地

址: B5h

位	名称	功能
7	-	保留
6	P1S	=1:允许P1口作为带施密特触发器的输入。
5	P0S	=1:允许P0口作为带施密特触发器的输入。
4	ENCLK	=1;使能时钟从XTAL2 管脚输出.
3	T1OE	=1:当定时器1溢出时P0.7脚翻转。P0.7脚的输出频率是定时器1溢出频率的一半。
2	T0OE	=1:当定时器0溢出时P1.2脚翻转。P1.2脚的输出频率是定时器0溢出频率的一半。
1	-	保留
0	-	保留

中断高级优先

位: 7 6 5 4 3 2 1 0

-	PADCH	PBOH	PSH	PT1H	PX1H	PT0H	PX0H
---	-------	------	-----	------	------	------	------

助记符: IP0H

地

址: B7h

位	名称	功能
7	-	保留位，如果读它结果是高。
6	PADCH	1:设置ADC中断为高级优先。
5	PBOH	1:设置欠压监测器中断为高级优先。
4	PSH	1:设置串行端口0中断为高级优先。
3	PT1H	1:设置定时器1中断为高级优先。
2	PX1H	1:设置外部中断1中断为高级优先
1	PT0H	1:设置定时器0中断为高级优先
0	PX0H	1:设置外部中断0中断为高级优先

中断优先权0

N79E8251/8241/8231/8221

www. dycmcu. com

位:	7	6	5	4	3	2	1	0
	-	PADC	PBO	PS	PT1	PX1	PT0	PX0

助记符: IP0

地

址: B8h

位	名称	功能
7	-	该位是没有使用，读的结果是‘1’.
6	PADC	1:设置中断ADC的优先权是较高优先级.
5	PBO	1:设置中断欠压监测器优先权是较高优先级.
4	PS	1:设置中断串行口的优先权是较高优先级.
3	PT1	1:设置中断定时器1的优先权是较高优先级.
2	PX1	1:设置中断外部中断1的优先权是较高优先级.
1	PT0	1:设置中断定时器0的优先权是较高优先级.
0	PX0	1:设置中断外部中断0的优先权是较高优先级.

从机 地址屏蔽允许

位:	7	6	5	4	3	2	1	0

助记符: SADEN

地址: B9h

位	名称	功能
7~0	SADEN	该寄存器使能串口0的自动地址识别功能，当SADEN中的某位被置为1，那么SADDR寄存器中的相应位会同接收到到的数据进行比较。如果SADEN.n被设为0，那么系统会忽略对该位的比较。如果SADEN为全0，那么对于所有的地址帧系统都会产生中断。

I2C 资料寄存器

位:	7	6	5	4	3	2	1	0
	I2DAT.7	I2DAT.6	I2DAT.5	I2DAT.4	I2DAT.3	I2DAT.2	I2DAT.1	I2DAT.0

助记符: I2DAT

地址: BCh

位	名称	功能
0~7	I2DAT	I2C的资料寄存器。

I2C 状态寄存器

位:	7	6	5	4	3	2	1	0
----	---	---	---	---	---	---	---	---

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

助记符: I2STATUS

地址: BDh

位	名称	功能
0~7	I2STATUS	该I2C状态寄存器: 低三位始终是0; 高5位包含状态码。含状态码有23可能; 当I2STATUS的值是F8H, 表示没有串行断请求; 其它的所有的I2STATUS值可以反映I2C的状态。当进入这些状态时会产生一个状态中断请求(SI=1)。一个有效的状态码在SI被硬件设为'1'后一个周期内反映到I2STATUS中; 在SI被软件清'0'后一个周期内反映到I2STATUS中。另外, 状态码是00H时表示总线错误; 当'起始'或'结束'时出现帧结构错误时会产生总线错误。

I2C 波特率控制寄存器

位: 7 6 5 4 3 2 1 0

I2CLK.7	I2CLK.6	I2CLK.5	I2CLK.4	I2CLK.3	I2CLK.2	I2CLK.1	I2CLK.0
---------	---------	---------	---------	---------	---------	---------	---------

助记符: I2CLK

地址: BEh

位	名称	功能
7~0	I2CLK	I2C的时钟速度位。

I2C 定时器/计数器寄存器

位: 7 6 5 4 3 2 1 0

-	-	-	-	-	ENTI	DIV4	TIF
---	---	---	---	---	------	------	-----

助记符: I2TIMER

地址: BFh

位	名称	功能
7~3	-	保留。
2	ENTI	允许I2C14-位定时器/计数器: 0:禁止14-位定时器/计数器计数。 1:允许14-位定时器/计数器计数, 14-位计数器被允许计数后将被清'0'。如果I2C的SI标志是'1'。该计数器不能向上计数。
1	DIV4	I2C 定时器/计数器时钟源分频功能: 0: 14-位定时器/计数器源时钟是Fosc。 1: 14-位定时器/计数器源时钟是Fosc/4。

N79E8251/8241/8231/8221

www. dycmcu. com

0	TIF	I2C 定时器/计数器计数标志: 0: 14-位定时器/计数器没有溢出. 1: 14-位定时器/计数器溢出, 在允许I2C定时器(ENT1)之前, SI必须清'0'。如果I2C中断被允许将执行中断服务程序, 该位有软件清'0'。
---	-----	--

I2C 控制 寄存器

位:	7	6	5	4	3	2	1	0
	-	ENS1	STA	STO	SI	AA	-	-

助记符: I2CON

地址: C0h

位	名称	功能
7	-	保留.
6	ENS1	允许I2C功能。
5	STA	I2C的START标志。
4	STO	I2C的STOP标志。
3	SI	I2C的中断标志。
2	AA	I2C的应答标志。
1	-	保留.
0	-	保留.

I2C地址寄存器

位:	7	6	5	4	3	2	1	0
	I2ADDR.7	I2ADDR.6	I2ADDR.5	I2ADDR.4	I2ADDR.3	I2ADDR.2	I2ADDR.1	I2ADDR.0

助记符: I2ADDR

地址: C1h

位	名称	功能
7~1	I2ADDR.7 ~ I2ADDR.1	I2C地址寄存器: 8051 uC可以直接对该8-位寄存器进行读/写操作。在主模式下寄存器的值没有意义; 在从机模式下, 高7位作为MCU本身的地址。如果地址符合硬件会自动应答。
0	GC	全呼功能. 0: 禁止全呼功能. 1: 允许全呼功能.

NVM地址

位:	7	6	5	4	3	2	1	0
----	---	---	---	---	---	---	---	---

NVMAD DR.7	NVMAD DR.6	NVMAD DR.5	NVMAD DR.4	NVMAD DR.3	NVMAD DR.2	NVMAD DR.1	NVMAD DR.0
---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------

助记符: NVMADDR

地

址: C6h

位	名称	功能
7~0	NVMADDR.7 ~ NVMADDR.0	NVM 地址: 寄存器标识为片内代码内存空间低字节地址NVM数据存储器

访问时控寄存器

位: 7 6 5 4 3 2 1 0

TA.7	TA.6	TA.5	TA.4	TA.3	TA.2	TA.1	TA.0
------	------	------	------	------	------	------	------

助记符: TA

地

址: C7h

位	名称	功能
7~0	TA	访问时控寄存器: 访问时控寄存器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。

NVM 控制

位: 7 6 5 4 3 2 1 0

EER	EWR	-	-	-	-	-	-
-----	-----	---	---	---	---	---	---

助记符: NVMCON

地

址: CEh

位	名称	功能
7	EER	NVM 页擦除位 0: 不擦除NVM页 1: 该位设为'1', 把NVM 数据页中的内容擦成'FFH'。NVM数据存储器有4个页, 每页大小为64字节。通过NVMADDR 寄存器选择页后, 在设定该位后, 该页将会被清除, 程序指针将会等待该指令的结束。在本指令结束之后程序指针将执行下一条指令。NVM 页地址定义如下页
6	EWR	NVM 数据 写 位 0: 不写NVM 数据。 1: 把改位设置为'1'向NVM写一个字节的数据。程序指针将会等待该指令的结束。在本指令结束之后程序指针将执行下一条指令。
5~0	-	保留

NVM 页地址定义表

页号	起始地址	结束地址
0	00H	3FH
1	40H	7FH
2	80H	BFH
3	C0H	FFH

注：W79E823，W79E822和W79E821没有页2和页3。

NVM 数据

位:	7	6	5	4	3	2	1	0
	NVMDA T.7	NVMDA T.6	NVMDA T.5	NVMDA T.4	NVMDA T.3	NVMDA T.2	NVMDA T.1	NVMDA T.0

助记符: NVM数据

地

址: CFh

位	名称	功能
7~0	NVMDAT.7 ~ NVMDAT.0	写NVM数据寄存器，读NVM 数据使用MOVC 指令。

程序状态字

位:	7	6	5	4	3	2	1	0
	CY	AC	F0	RS1	RS0	OV	F1	P

助记符: PSW

地址: D0h

位	名 称	功 能
7	CY	进位标志：当ALU进行算术运算产生进位或借位时置位
6	AC	辅助进位标志：高半字节运算产生进位或借位时置位
5	F0	用户标志0：用户可以使用的通用标志位
4	RS1	寄存器区选择位
3	RS0	寄存器区选择位
2	OV	溢出标志: 作为一个预先操作，当第七位而不是第八位产生进位时该标志被设置
1	F1	用户标志1: 用户可以使用的通用标志位
0	P	奇、偶标志位。由硬件控制其置位与复位。用于表示累加器中“1”的数目奇数还是偶数

RS.1-0: 寄存器分组选择位:

RS1	RS0	寄存器分组号	地址
0	0	0	00-07h
0	1	1	08-0Fh
1	0	2	10-17h
1	1	3	18-1Fh

PWM计数器高位寄存器

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PWMP.9	PWMP.8

助记符: PWMPH

地

址: D1h

位	名称	功能
7~2	-	保留
1~0	PWMP.9 ~PWMP.8	PWM计数器寄存器的9~8位.

PWM 0高位寄存器

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PWM0.9	PWM0.8

助记符: PWM0H

地

址: D2h

位	名称	功能
7~2	-	保留
1~0	PWM0.9 ~PWM0.8	PWM 0寄存器9~8位.

PWM 1 高位寄存器

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

助记符: PWM1H

地

址: D3h

位	名称	功能
7~2	-	保留
1~0	PWM1.9 ~PWM1.8	PWM1寄存器9~8位.

PWM 2高位寄存器

N79E8251/8241/8231/8221

www.dycmcu.com

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PWM2.9	PWM2.8

助记符: PWM2H

地

址: D5h

位	名称	功能
7~2	-	保留
1~0	PWM2.9 ~PWM2.8	PWM2 寄存器9~8位.

PWM 3 高位寄存器

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PWM3.9	PWM3.8

助记符: PWM3H

地

址: D6h

位	名称	功能
7~2	-	保留
1~0	PWM3.9 ~PWM3.8	PWM3 寄存器9~8位.

PWM 控制寄存器3

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	BKF

助记符: PWMCON3

地

址: D7h

位	名称	功能
7~1	-	保留
0	BKF	外部钳制脚标志. 0: PWM不钳制. 1: WM被外部钳制脚钳制。有软件清'0'。

看门狗控制

位:	7	6	5	4	3	2	1	0
	WDRUN	-	WD1	WD0	WDIF	WTRF	EWRST	WDCLR

助记符: WDCON

地址: D8h

位	名称	功能
---	----	----

7	WDRUN	0: 看门狗停止 1: 看门狗运行.
6	-	预留
5	WDI	看门狗 定时器选择.
4	WD0	看门狗 定时器选择.
3	WDIF	看门狗定时器中断标志 如果看门狗中断使能, 硬件会将该位置1 表示看门狗定时器中断产生。如果看门狗定时器中断关闭, 那么该位的置位表示看门狗定时器已经超时。该位必须由软件来清零

(续)

位	名称	功能
2	WTRF	看门狗计时器重定标志 当看门狗定时器复位后置位。该位可用来判别复位的类型。软件可以读取该位, 但必须手动清除。掉电复位会将此位清除。如果EWT=0, 该位不会受看门狗定时器的影响
1	EWRST	0: 禁止看门狗定时器复位. 1: 允许看门狗定时器复位.
0	WDCLR	狗定时器清'0' WDCON.0 – 将看门狗定时器复位。该位用于清除看门狗定时器并将它复位。该位会自动清零, 在软件向该位写入1后, 系统会自动将它置0。如果看门狗计时器重定使能, 那么软件必须在看门狗定时器溢出后512个时钟周期内将看门狗定时器清零, 否则将会产生一个看门狗定时器复位

特殊功能寄存器WDCON在复位后的初值是0x0000x0B。WTRF (WDCON.2) 被设置'1'是看门狗定时器复位, 被设置'0'是上电复位。WDIF (WDCON.3)外部复位不改变。上电复位时POR被设置'1'。上电复位时EWRST (WDCON.1)被设置'0', 其它复位不受影响。特殊功能寄存器WDCON的EWRST, WDIF和WDCLR位的写操作受TA寄存器控制; 其它位不受控制, 任何读操作不受控制。具体请参阅TA 寄存器的描述。

TA	REG	C7H
WDCON	REG	D8H
CKCON	REG	8EH

```
MOV    TA, #AAH
MOV    TA, #55H
SETB   WDCON.0           ;复位看门狗定时器
ORL     CKCON, #00110000B ;选择26位看门狗定时器
```

```
MOV    TA, #AAH
MOV    TA, #55H
ORL     WDCON, #00000010B ;允许看门狗
```

PWM 计数器低位寄存器

位:	7	6	5	4	3	2	1	0
----	---	---	---	---	---	---	---	---

N79E8251/8241/8231/8221

www.dycmcu.com

PWMP.7	PWMP.6	PWMP.5	PWMP.4	PWMP.3	PWMP.2	PWMP.1	PWMP.0
--------	--------	--------	--------	--------	--------	--------	--------

助记符: PWMPL

地

址: D9h

位	名称	功能
7~0	PWMP.7 ~PWMP.0	PWM计数器的低位寄存器.

PWM0位寄存器

位: 7 6 5 4 3 2 1 0

PWM0.7	PWM0.6	PWM0.5	PWM0.4	PWM0.3	PWM0.2	PWM0.1	PWM0.0
--------	--------	--------	--------	--------	--------	--------	--------

助记符: PWM0L

地

址: DAh

位	名称	功能
7~0	PWM0.7 ~PWM0.0	PWM0的低位寄存器.

PWM1低位寄存器

位: 7 6 5 4 3 2 1 0

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

助记符: PWM1l

地

址: DBh

位	名称	功能
7~0	PWM1.7 ~PWM1.0	PWM 1的低位寄存器.

PWM控制寄存器1

位: 7 6 5 4 3 2 1 0

PWMRUN	Load	CF	CLRPWM	PWM3I	PWM2I	PWM1I	PWM0I
--------	------	----	--------	-------	-------	-------	-------

助记符: PWMCON1

地址: DCh

位	名称	功能
7	PWMRUN	0: PWM没有运行. 1: PWM计数器运行.
6	Load	0: PWMP寄存器的值比较器不会被装载到计数器和比较器寄存器中; 1: 在计数器向下溢出后, PWMP寄存器的值将会装入计数器寄存器, 下一个周期硬件清'0'.

5	CF	0: 10-位向下计数计数器没有溢出。 1: 10-位向下计数器向下溢出。由软件清'0'。
4	CLRPWM	1: 把10-位 PWM 计数器设置为 000H. 置位后由硬件自动清除。
3	PWM3I	0: PWM3正相输出 1: PWM3反相输出
2	PWM2I	0: PWM2正相输出 1: PWM2反相输出

位	名称	功能
1	PWM1I	0: PWM1正相输出 1: PWM1反相输出
0	PWM0I	0: PWM0正相输出 1: PWM0反相输出

PWM2的低位寄存器

位: 7 6 5 4 3 2 1 0

PWM2.7	PWM2.6	PWM2.5	PWM2.4	PWM2.3	PWM2.2	PWM2.1	PWM2.0
--------	--------	--------	--------	--------	--------	--------	--------

助记符: PWM2L

地

址: DDh

位	名称	功能
7~0	PWM2.7 ~PWM2.0	PWM2的低位寄存器.

PWM3低位寄存器

位: 7 6 5 4 3 2 1 0

PWM3.7	PWM3.6	PWM3.5	PWM3.4	PWM3.3	PWM3.2	PWM3.1	PWM3.0
--------	--------	--------	--------	--------	--------	--------	--------

助记符: PWM3L

地

址: DEh

位	名称	功能
7~0	PWM3.7 ~PWM3.0	PWM 3的低位寄存器

PWM控制寄存器 2

位: 7 6 5 4 3 2 1 0

BKCH	BKPS	BPEN	BKEN	PWM3B	PWM2B	PWM1B	PWM0B
------	------	------	------	-------	-------	-------	-------

助记符: PWMCON2

地址: DFh

位	名称	功能
7	BKCH	见下表
6	BKPS	0: P0.2=0, 钳制PWM 1: P0.2=1, 钳制PWM
5	BPEN	见下表
4	BKEN	0: 钳制禁止. 1: 钳制允许, 详见下表;
3	PWM3B	0: 当钳制有效PWM3输出为低 1: 当钳制有效 PWM3输出为高
2	PWM2B	0: 当钳制有效 PWM2输出为低 1: 当钳制有效 PWM2输出为高
1	PWM1B	0: 当钳制有效 PWM1输出为低 1: 当钳制有效 PWM1输出为高
0	PWM0B	0: 当钳制有效 PWM0输出为低 1: 当钳制有效 PWM0输出为高

钳制条件表

BPEN	BKCH	钳制条件
0	0	钳制打开, 软件钳制有BKEN控制
0	1	打开: 当PWM不运行(PWMRUN=0)时, 由PWMnB设定PWM的输出条件 关闭: 当PWM为运行(PWMRUN=1)时,
1	0	钳制打开, 当钳制脚指定后, 没有PWM输出, PWMRUN位被清除并把BKF标志只 为置'0'。.
1	1	保留

累加器

位:	7	6	5	4	3	2	1	0
	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0

助记符: ACC

地址: E0h

ACC.7-0: The A (或ACC)寄存器是标准8052的累加器。

ADC 控制寄存器

位:	7	6	5	4	3	2	1	0
	ADC.1	ADC.0	ADCEX	ADCI	ADCS	RCCLK	AADR1	AADR0

助记符: ADCCON

地址: E1h

位	名 称	功 能														
7	ADC.1	ADC 结果的第1位。														
6	ADC.0	ADC 结果的第0位。														
5	ADCEX	=‘0’，只有软件把ADCS 位置‘1’才开始ADC转换。 =‘1’，软件把ADCS 位置‘1’或外部STADC（1.4）上的下降沿开始ADC转换。														
4	ADCI	ADC 中断标志位。当ADC转换结束，转换结果可以读时，ADCI标志位置‘1’。若ADC中断使能，就可以进入ADC中断，进入中断后该标志位清‘0’，也可以软件清‘0’但不能软件置‘1’。若该标志位为‘1’时，就无法开始新的ADC转换。														
3	ADCS	ADC 开始和状态标志位。把ADCS置‘1’开始一次ADC转换，可以由软件或外部的STADC信号置‘1’，当ADC忙时ADCS=‘1’。转换结束，ADCI置位后ADCS=‘0’。如果ADCS=‘1’或ADCI =‘1’ 时无法开始新的ADC转换														
		ADCI	ADCS	ADC 状态	0	0	ADC 空闲，可以开始一个新的转换	0	1	ADC 忙，转换中	1	0	转换结束，开始新的转换时，要求ADCI=0	1	1	转换结束，开始新的转换时，要求ADCI=0
		ADCI	ADCS	ADC 状态												
		0	0	ADC 空闲，可以开始一个新的转换												
0	1	ADC 忙，转换中														
1	0	转换结束，开始新的转换时，要求ADCI=0														
1	1	转换结束，开始新的转换时，要求ADCI=0														
如果软件在设ADCI=0的时候，ADCS=1，那么在同一通道上的新的A/D转换马上开始。但推荐先ADCI=0，再ADCS=1。																
2	RCCLK	0: CPU 时钟作为 ADC 时钟. 1: 内部 RC 时钟作为 ADC 时钟.														
1	AADR1	见下表。														
0	AADR0	见下表。														

AADR1, AADR0: ADC仿真输入通道选择位:

ADC仿真输入通道选择位。只有当ADCI=0和ADCS=0时才可以改变这些位。

AADR1	AADR0	选择仿真输入通道
0	0	AD0 (P0.3)
0	1	AD1 (P0.4)
1	0	AD2 (P0.5)
1	1	AD3 (P0.6)

ADC 转换结果寄存器

位:	7	6	5	4	3	2	1	0
	ADC.9	ADC.8	ADC.7	ADC.6	ADC.5	ADC.4	ADC.3	ADC.2

助记符: ADCH

地址: E2h

N79E8251/8241/8231/8221

www.dycmcu.com

位	名称	功能
7~0	ADC.9 ~ADC.2	ADC转换结果

中断允许寄存器 1

位:	7	6	5	4	3	2	1	0
	-	-	EPWM	EWDI	EC2	EC1	EKB	EI2

助记符: IE1

地址: E8h

位	名称	功能
7	-	保留.
6	-	保留.
5	EPWM	0: 当外部钳制时, 禁止PWM中断。 1: 当外部钳制时, 允许PWM中断。
4	EWDI	0: 禁止看门狗定时器中断。 1: 允许看门狗定时器中断。
3	EC2	0: 禁止比较器2中断。 1: 允许比较器2中断。
2	EC1	0: 禁止比较器1中断。 1: 允许比较器1中断。
1	EKB	0: 禁止键盘中断。 1: 允许键盘中断。
0	EI2	0: 禁止I2C中断。 1: 允许I2C中断。

B 寄存器

位:	7	6	5	4	3	2	1	0
	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0

助记符: B

地

址: F0h

B.7-0: B寄存器是标准8052中的辅助累加器

端口0数字输入禁止

位:	7	6	5	4	3	2	1	0
	P0ID.7	P0ID.6	P0ID.5	P0ID.4	P0ID.3	P0ID.2	P0ID.1	P0ID.0

助记符: P0ID

地址: F6h

位	名称	功能
7~0	P0ID.7 ~P0ID.0	允许/禁止端口0数字 输入. 0:允许端口0数字输入 1:禁止端口0数字输入

中断优先级 1

位:	7	6	5	4	3	2	1	0
	-	-	PPWMH	PWDIH	PC2H	PC1H	PKBH	PI2H

助记符: IP1H

地址: F7h

位	名称	功能
7	-	保留.
6	-	保留.
5	PPWMH	1: 设置PWM钳制中断级优先级最高
4	PWDIH	1: 设置看门狗中断优先级最高
3	PC2H	1: 设置比较器 2中断优先级最高
2	PC1H	1: 设置比较器 1中断优先级最高
1	PKBH	1: 设置键盘中断优先级最高
0	PI2H	1: 设置I2C中断优先级最高

中断优先权1

位:	7	6	5	4	3	2	1	0
	-	-	PPWM	PWDI	PC2	PC1	PKB	PI2

助记符: IP1

地址: F8h

位	名称	功能
7	-	保留.
6	-	保留.
5	PPWM	1: 设置PWM外部钳制中断优先权较高
4	PWDI	1: 设置看门狗中断优先权较高
3	PC2	1: 设置比较器2中断优先权较高
2	PC1	1: 设置比较器1中断优先权较高
1	PKB	1: 设置键盘中断优先权较高

N79E8251/8241/8231/8221

www. dycmcu. com

0	PI2	1: 设置I2C中断优先权较高
---	-----	-----------------

28 指令

N79A82X1系列执行8032 体系微处理器中的所有的指令。指令的功能，对标志位及状态位的影响完全与标准8032 处理器的指令相同。但是指令的时序存在差别；主要是有2个原因，第一N79A82X1系列每4个时钟周期为一个机器周期，而标准8032每12个时钟周期为一个极其周期。另外N79A82X1系列 每个机器周期只有一个取动作，而标准8032 每个机器周期有2个取动作。

N79A82X1系列 的优势在于由于每个机器周期只有一个取动作，因此对大多数指令来说其机器周期数和它的操作数数目相同。而对于跳转和调用指令，会增加一个指令周期用以计算新的程序地址。从整体上来说，N79A82X1系列 减少了空取和等待的周期，因而提高了系统的效率。

表: 受指令影响的标志位

指令	CARRY	溢出	AUXILIARY CARRY	指令	CARRY	溢出	AUXILIARY CARRY
ADD	X	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, 位	X		
MUL	0	X		ANL C, 位	X		
DIV	0	X		ORL C, 位	X		
DA A	X			ORL C, 位	X		
RRC A	X			MOV C, 位	X		
RLC A	X			CJNE	X		
SETB C	1						

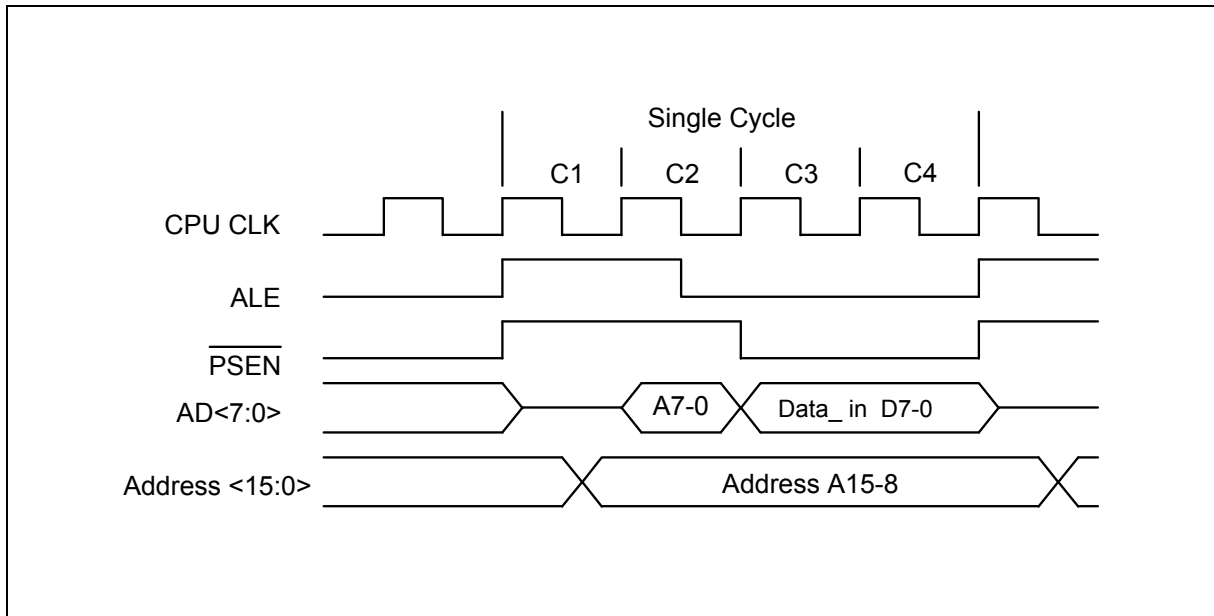
"X" 表示会受到指令的影响。

28.1 指令时序

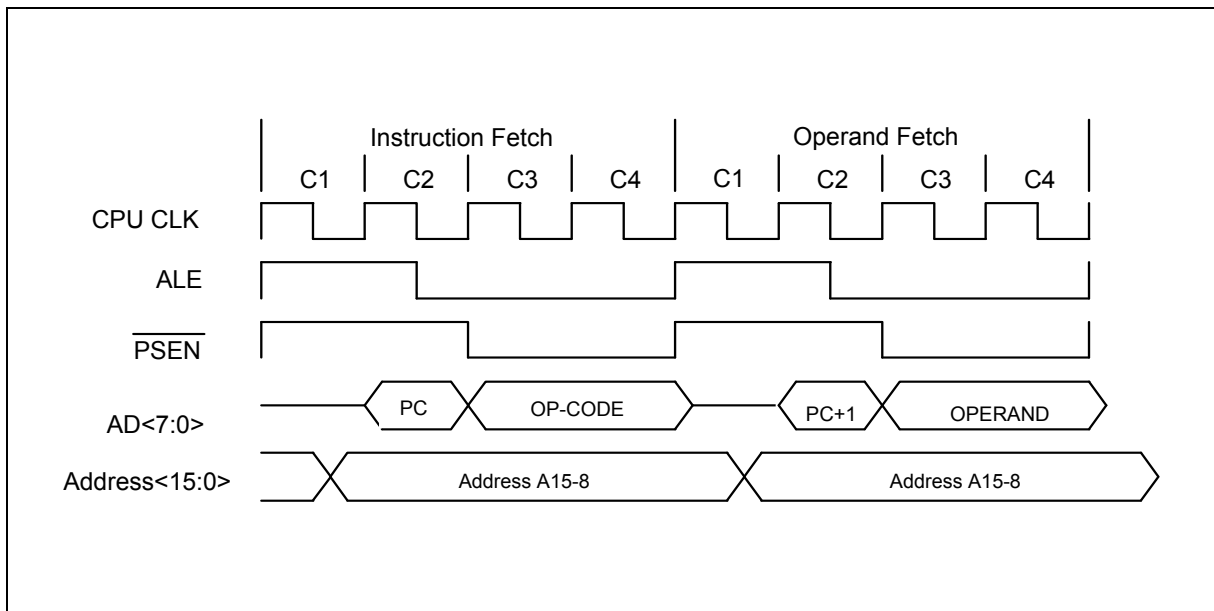
指令时序对N79A82X1系列来说是一个很重要的特性，对于用软件的方式来产生定时的用户更为重要。它也向用户说明N79A82X1系列与标准8032在时序上的差别。在N79A82X1系列中每个机器周期是4个时钟周期，每个时钟周期都是一个确定的状态。因此一个机器周期由4个确定的状态C1、C2、C3、C4组成。由于每条 指令的执行速度都加快了，所以时钟的2个跳变边沿都用于内部时序。因此时钟的占空比接近于50%，以避免时间上发生冲突。

前面已经说到N79A82X1系列每一个机器周期进行一次代码读取操作，因此对大多数指令来说，执行指令的机器周期与操作码中的字节数相同。系统总共有256 个操作码，其中有128个是单周期指令。因此在N79A82X1系列中有一半的指令会在4个时钟周期内执行完毕。对多数双字节指令来说，指令的执行周期是2个机器周期。但也有指令为一个字节但周期是2个时钟周期的情况；一个需要特别注意的指令是MOVX指令，在标准8032中他的指令周期固定为2个机器周期. 但在N79A82X1系列中他的指令周期可变为2-9 个机器周期。RD 和 WR 信号也有相应的变化。这为用户访问快速或慢速设备就带来了方便，不需使用额外的外围电路，也减少了软件负担。剩下的指令的机器周期数目可以是3个，4个，5个。注意在N79A82X1系列中基于指令字节数目的不同，共有5种类型的指令，而标准8032中只有3种指令类型。但是N79A82X1系列中每4个时钟周期为一个机器周期，而不是标准8032中每12个时钟周期为一个机器周期。因此尽管指令种类增多，N79A82X1系列中的指令执行速度要比标准8032快1.5-3倍。（以时钟周期计算）

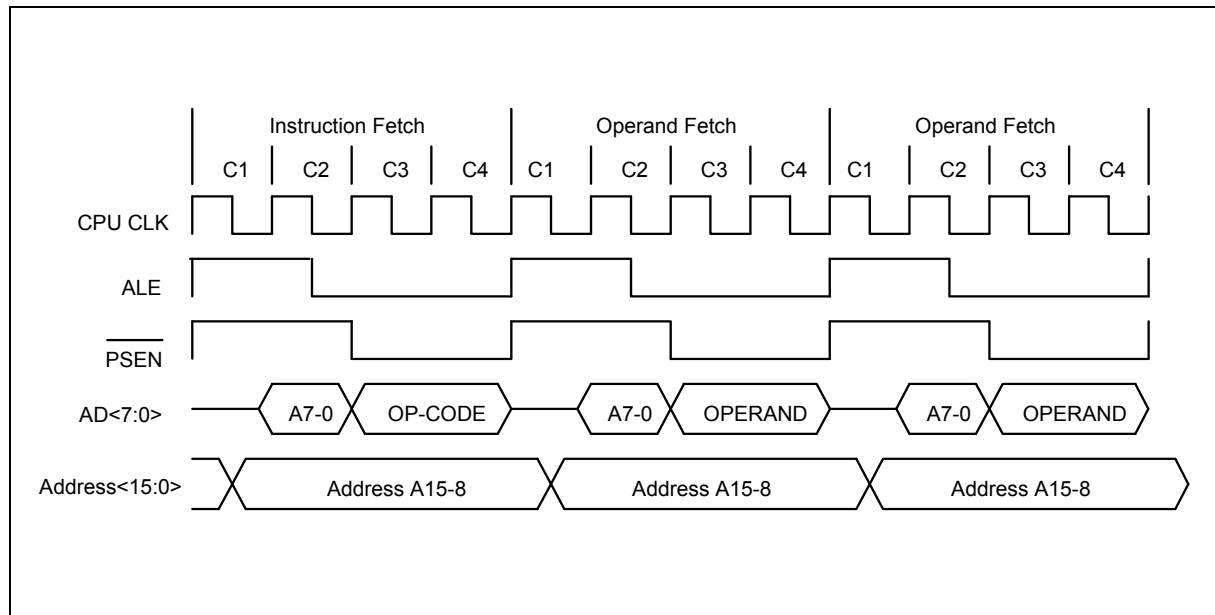
（注：以下指令时序为内部控制时序信号）



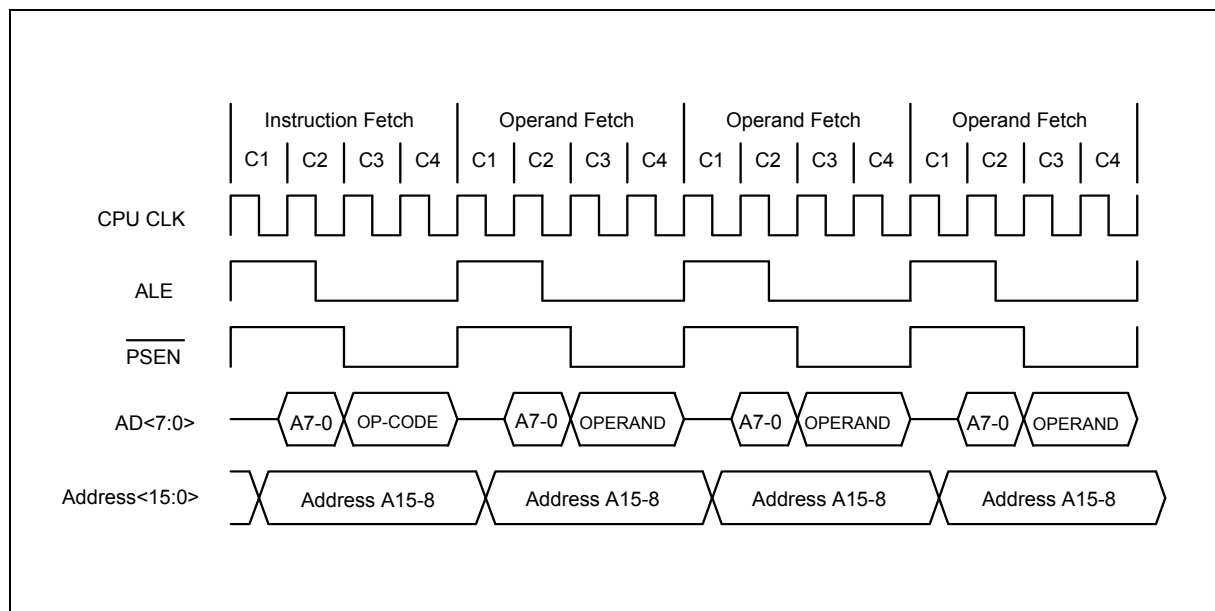
单周期指令时序



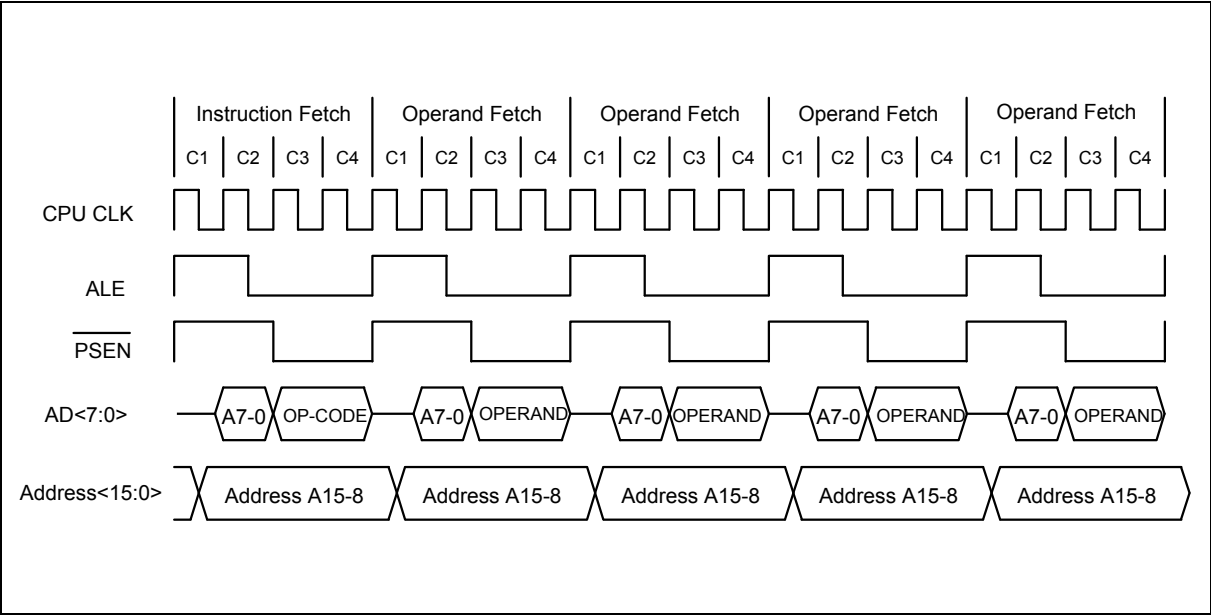
双周期指令时序



3周期指令指令时序



4周期指令指令时序



5周期指令时序

29 DC/ADC转换/模拟比较器电路电气特性

29.1 极限参数

参数	符号	最小值	最大值	单位
DC电源	$V_{DD} - V_{SS}$	-0.3	+7.0	V
输入电压	V_{IN}	$V_{SS} - 0.3$	$V_{DD} + 0.3$	V
操作温度	TA	-40	+85	°C
存储温度	Tst	-55	+150	°C

注释：超出最大绝对额定值表所列的情况使用，会对器件的可靠性和寿命造成严重损害。

29.2 DC 电气特性

(TA = -40~85°C, 非特别注明皆为该条件)

参数	符号	规格				测试条件
		最小值	典型值	最大值	单位	
操作电压	V_{DD}	2.7		5.5	V	$V_{DD} = 4.5V \sim 5.5V @ 20MHz$ $V_{DD} = 2.7V \sim 5.5V @ 12MHz$
操作电流	I_{DD}		18	25	mA	$V_{DD} = 5.0V @ 20MHz$, 无负载, RST = V_{SS}
			6	8	mA	$V_{DD} = 3.0V @ 12MHz$, 无负载, RST = V_{SS}
空闲电流	I_{IDLE}		11.5	15	mA	$V_{DD} = 5.5V, 20MHz$, 无负载
			5	6.5	mA	$V_{DD} = 3.0V, 12MHz$, 无负载
掉电电流	I_{PWDN}		1	10	μA	$V_{DD} = 5.5V$, 无负载 @禁止BOV 功能
			1	10	μA	$V_{DD} = 3.0V$, 无负载 @禁止BOV 功能
P0, P1输入电流	I_{IN1}	-50	-	+10	μA	$V_{DD}=5.5V$, $V_{IN}=0 < V_{IN} < V_{DD}$
P1.5(RST pin)[1] 输入电流	I_{IN2}	-55	-45	-30	μA	$V_{DD} = 5.5V$, $V_{IN} = 0.45V$
P0,P1 (开漏) 输入漏电流	I_{LK}	-10	-	+10	μA	$V_{DD} = 5.5V$, $0 < V_{IN} < V_{DD}$
P0,P1逻辑1到0转换电流	$I_{TL}^{[3]}$	-500	-	-200	μA	$V_{DD} = 5.5V$, $V_{IN} < 2.0V$
P0,P1输入低电压 (TTL输入)	V_{IL1}	0	-	1.0	V	$V_{DD} = 4.5V$
		0	-	0.6		$V_{DD} = 2.7V$
P0, P1输入高电压 (TTL输入)	V_{IH1}	2.4	-	$V_{DD} + 0.2$	V	$V_{DD} = 5.5V$
		2.0	-	$V_{DD} + 0.2$		$V_{DD} = 3.0V$

N79E8251/8241/8231/8221

www. dycmcu. com

XTAL1[*2]输入低电压	V_{IL3}	0	-	0.8	V	$V_{DD} = 4.5V$
		0	-	0.4		$V_{DD} = 3.0V$
XTAL1[*2]输入高电压	V_{IH3}	3.5	-	$V_{DD} + 0.2$	V	$V_{DD} = 5.5V$
		2.4	-	$V_{DD} + 0.2$		$V_{DD} = 3.0V$
负极区间(施密特输入)	V_{ILS}	-0.5	-	$0.3 V_{DD}$	V	
正极区间(施密特输入)	V_{IHS}	$0.7 V_{DD}$	-	$V_{DD} + 0.5$	V	
滞后电压	V_{HY}		$0.2 V_{DD}$		V	
P0, P1源电流 (准双向和弱上拉模式)	I_{sr1}	-150	-210	-360	μA	$V_{DD} = 4.5V, V_S = 2.4V$
P0, P1反向电流 (准双向和弱上拉模式)	I_{SK2}	13	18.5	24	mA	$V_{DD} = 4.5V, V_S = 0.45V$
P0, P1输出低电压 (推挽模式)	V_{OL1}	-	0.5	0.9	V	$V_{DD} = 4.5V, I_{OL} = 20 \text{ mA}$
		-	0.1	0.4	V	$V_{DD} = 2.7V, I_{OL} = 3.2 \text{ mA}$
P0, P1输出高电压 (推挽模式)	V_{OH}	2.4	3.4	-	V	$V_{DD} = 4.5V, I_{OH} = -16mA$
		1.9	2.4	-		$V_{DD} = 2.7V, I_{OH} = -3.2mA$
欠压电压 BOV=1	$V_{BO2.5}$	2.4	-	2.7	V	
欠压电压 BOV=0	$V_{BO3.8}$	3.5	-	4	V	
比较器参考电压	V_{ref}	1.02	1.20	1.31	V	

注: *1. RST 脚是一个施密特触发输入脚.

*2. XTAL1是一个CMOS输入脚.

29.3 ADC转换DC电气特性

($V_{DD} - V_{SS} = 3.0 \sim 5V$, $T_A = -40 \sim 85^\circ C$, $F_{osc} = 20MHz$, 除非另有说明.)

参数	符号	规格				测试条件
		最小值.	典型值.	最大值.	单位	
模拟输入	A_{vin}	$V_{SS} - 0.2$		$V_{DD} + 0.2$	V	
ADC 时钟	ADCCLK	200KHz	-	5MHz	Hz	ADC输入时钟
转换时间	t_C		$52t_{ADC}^{[1]}$		μs	
微分非线性误差	DNL	-1	-	+1	LSB	
积分非线性误差	INL	-2	-	+2	LSB	
冲销误差	Ofe	-1	-	+1	LSB	
增益误差	Ge	-1	-	+1	%	
绝对电压误差	Ae	-3	-	+3	LSB	

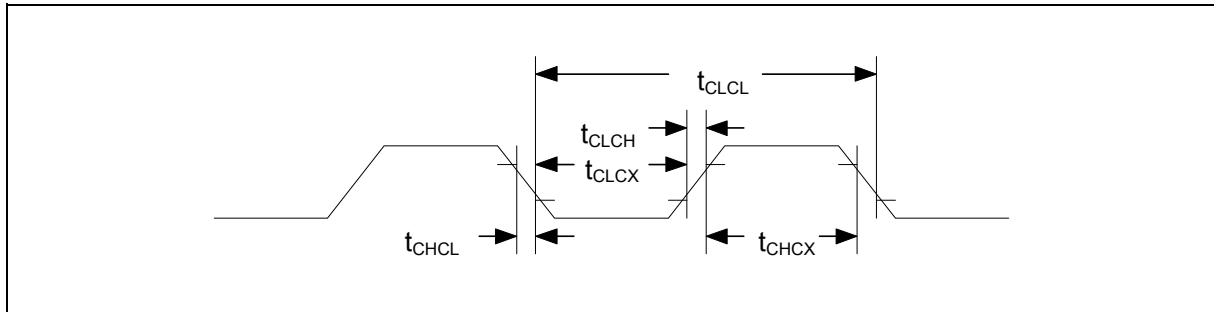
注: t_{ADC} : ADC输入时钟周期

29.4 模拟比较器电气特性

(VDD-VSS = 3.0~5V, TA = -40~85°C, Fosc = 20MHz, 除非另加说明)

参数	符号	规格				测试条件
		最小值.	典型值.	最大值.	单位	
电压比较器输入偏移	V _{CR}	0		V _{DD} -0.3	V	
共模抑制比	CMRR			-50	dB	
响应时间	t _{RS}	-	30	100	ns	
比较器允许到输出有效时间	t _{EN}	-	1	5	us	
比较器输入漏电流	I _{IL}	-10	0	10	uA	0 < V _{IN} < V _{DD}

29.5 AC电气特性



注: 占空比为50%.

29.6 外部时钟特性

参数	符号	最小值	典型值.	最大值	单位	注
时钟高时间	t _{CHCX}	12.5	-	-	nS	
时钟低时间	t _{CLCX}	12.5	-	-	nS	
时钟Rise 时间	t _{CLCH}	-	-	10	nS	
时钟Fall 时间	t _{CHCL}	-	-	10	nS	

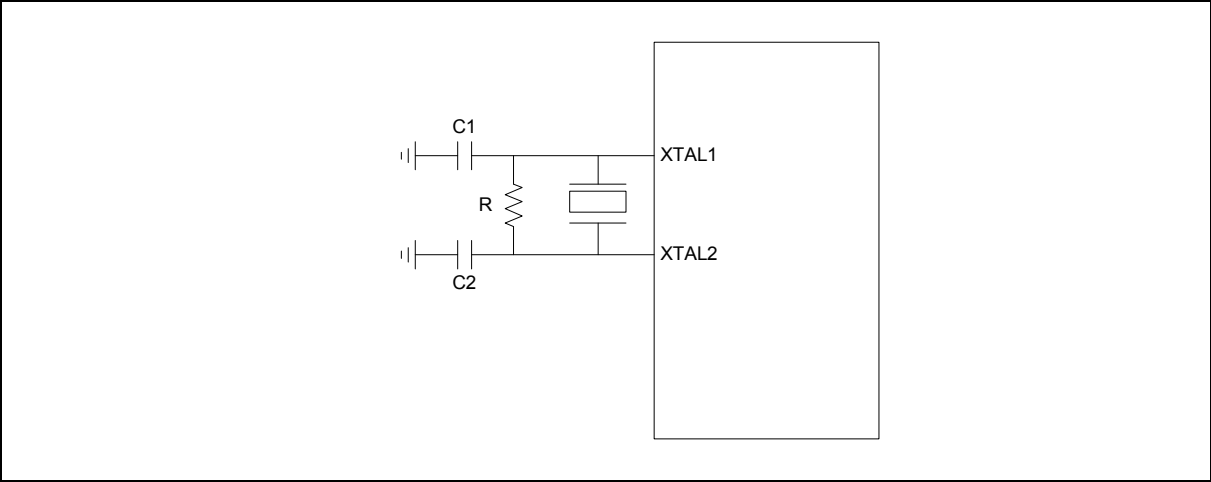
29.7 AC规格

参数	符号	时钟最小值	时钟最大值	单位
振荡器频率	1/ t _{CLCL}	0	20	MHz

29.8 典型应用电路

晶振	C1	C2	R
4MHz ~ 20 MHz	无	无	无

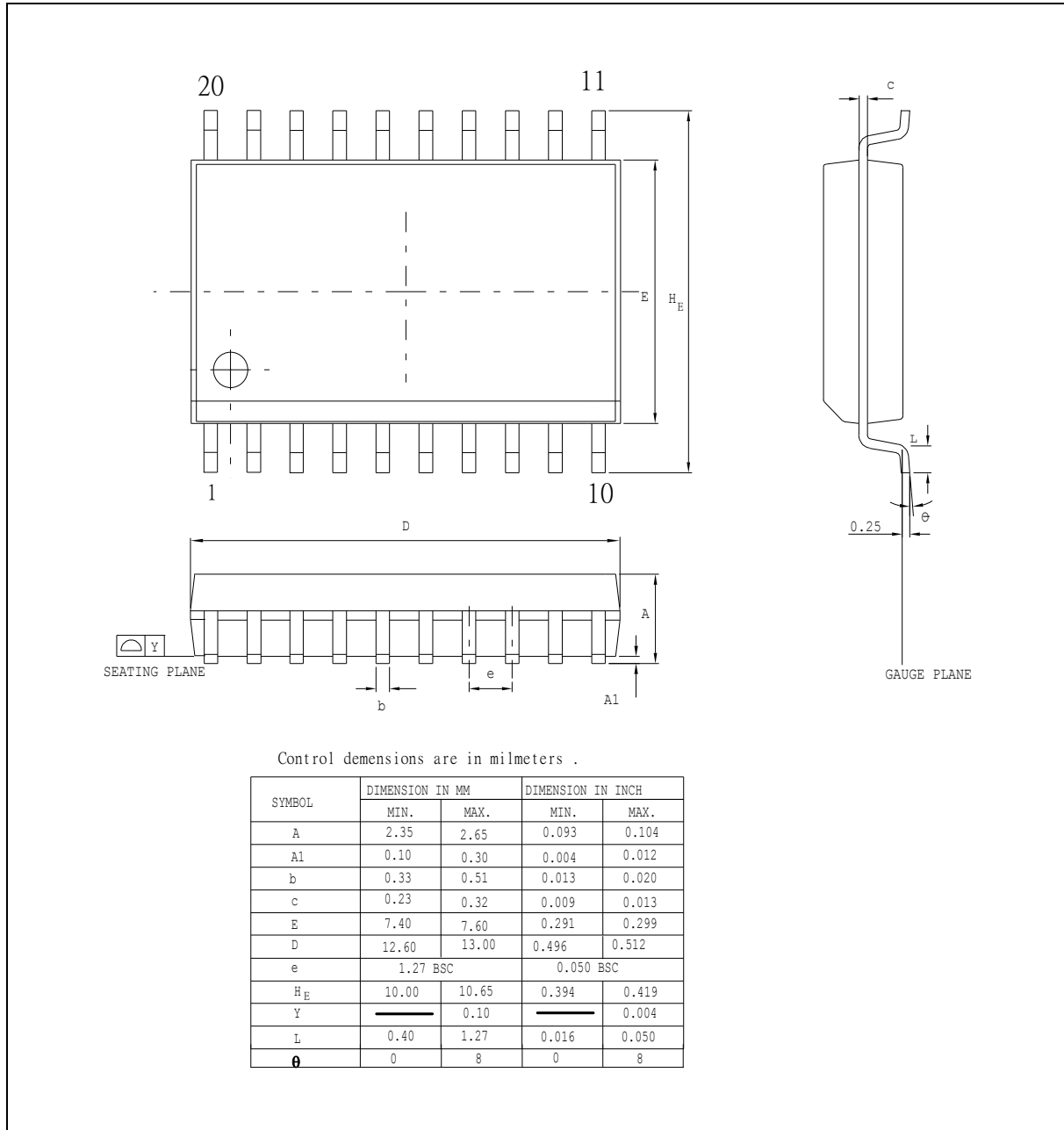
上表所示是晶振应用参考值



30 封装尺寸

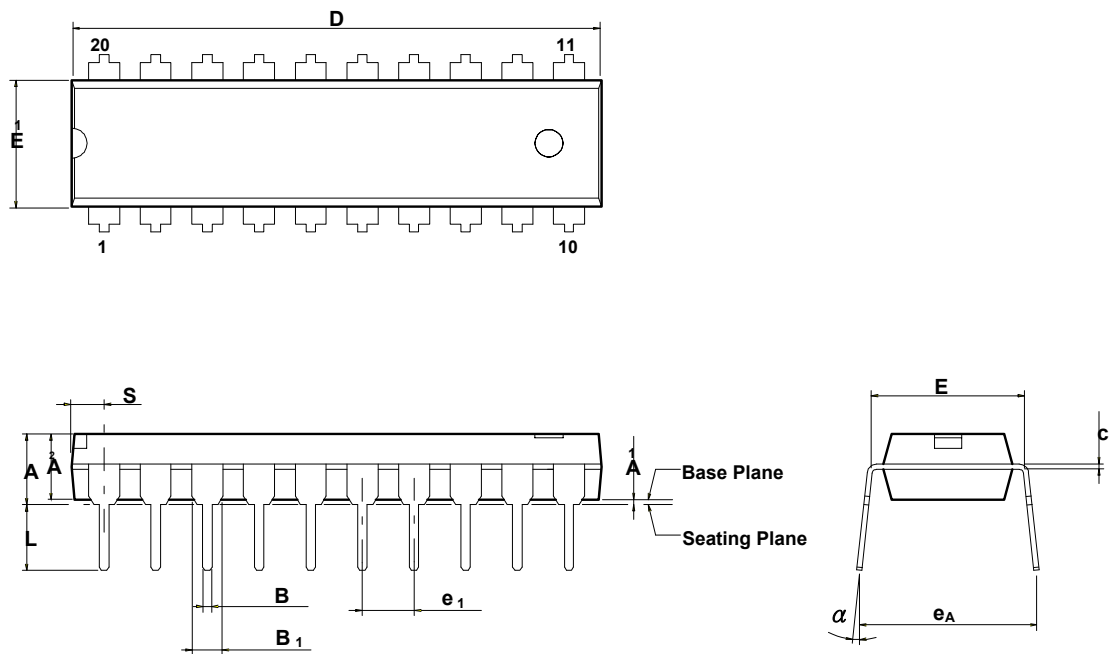
30.1 20-pin SOP

20L SOP-300mil



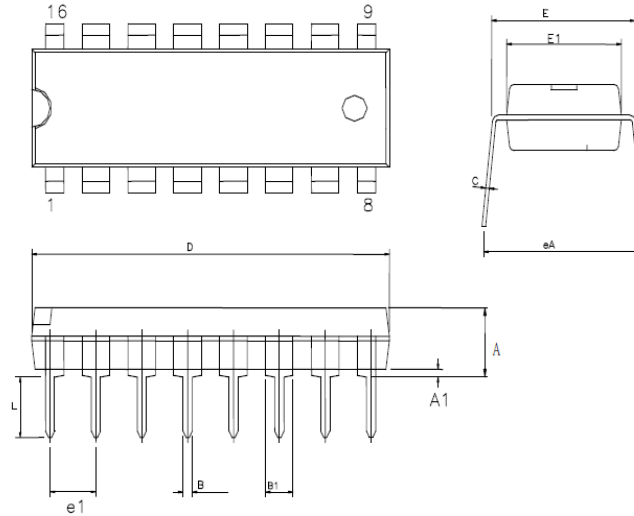
30.2 20-pin DIP

20L PDIP 300mil



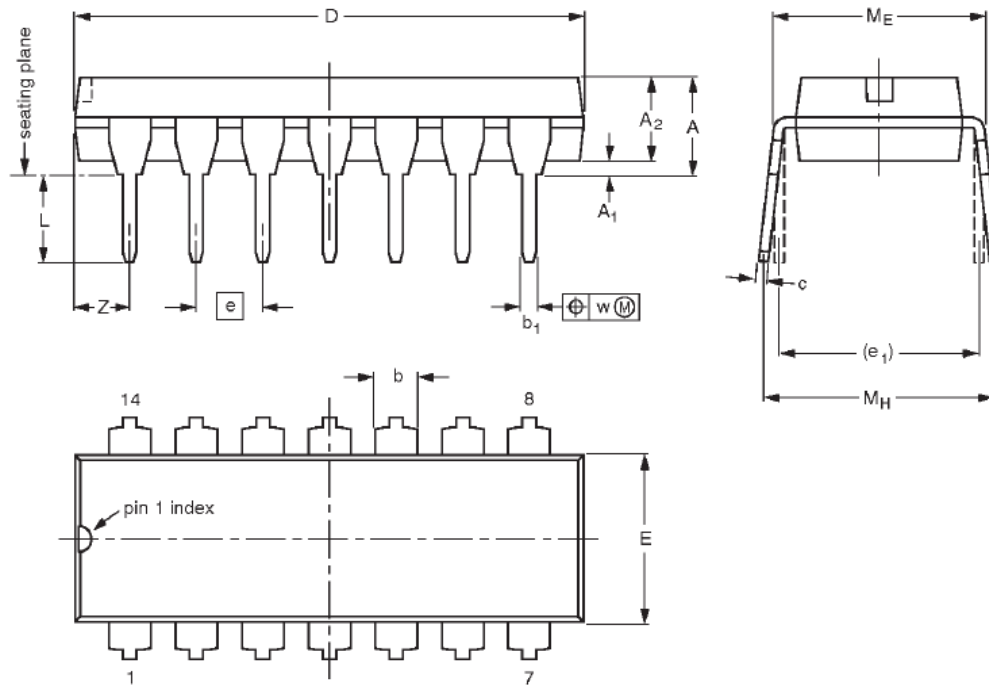
Symbol	Dimension in inch			Dimension in mm		
	Min	Nom	Max	Min	Nom	Max
A	—	—	0.175	—	—	4.45
A₁	0.010	—	—	0.25	—	—
A₂	0.125	0.130	0.135	3.18	3.30	3.43
B	0.016	0.018	0.022	0.41	0.46	0.56
B₁	0.058	0.060	0.064	1.47	1.52	1.63
c	0.008	0.010	0.014	0.20	0.25	0.36
D	—	1.026	1.040	—	20.06	26.42
E	0.290	0.300	0.310	7.37	7.62	7.87
E₁	0.245	0.250	0.255	6.22	6.35	6.48
e₁	0.090	0.100	0.110	2.29	2.54	2.79
L	0.120	0.130	0.140	3.05	3.30	3.56
α	0	—	15	0	—	15
e_A	0.335	0.355	0.375	8.51	9.02	9.53
S	—	—	0.075	—	—	1.91

30.3 16-pin PDIP



Symbol	Dimension in inch			Dimension in mm		
	Min	Nom	Max	Min	Nom	Max
A	—	—	0.175	—	—	4.45
A ₁	0.010	—	—	0.25	—	—
A ₂	0.125	0.130	0.135	3.18	3.30	3.43
B	0.016	0.018	0.022	0.41	0.46	0.56
B ₁	0.058	0.060	0.064	1.47	1.52	1.63
c	0.008	0.010	0.014	0.20	0.25	0.36
D	—	1.026	1.040	—	26.06	26.42
E	0.290	0.300	0.310	7.37	7.62	7.87
E ₁	0.245	0.250	0.255	6.22	6.35	6.48
e ₁	0.090	0.100	0.110	2.29	2.54	2.79
L	0.120	0.130	0.140	3.05	3.30	3.56
α	0	—	15	0	—	15
e _A	0.335	0.355	0.375	8.51	9.02	9.53
S	—	—	0.075	—	—	1.91

30.4 14-pin PDIP



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A ₁ min.	A ₂ max.	b	b ₁	c	D ⁽¹⁾	E ⁽¹⁾	e	e ₁	L	M _E	M _H	w	Z ⁽¹⁾ max.
mm	4.2	0.51	3.2	1.73 1.13	0.53 0.38	0.36 0.23	19.50 18.55	6.48 6.20	2.54	7.62	3.60 3.05	8.25 7.80	10.0 8.3	0.254	2.2
inches	0.17	0.020	0.13	0.068 0.044	0.021 0.015	0.014 0.009	0.77 0.73	0.26 0.24	0.10	0.30	0.14 0.12	0.32 0.31	0.39 0.33	0.01	0.087

Symbol	Dimensions in inches	Dimensions in mm
A	0.175 Max.	4.45 Max.
A ₁	0.010 Min.	0.25 Min.
A ₂	0.130 ± 0.010	3.30 ± 0.25
B	0.018 +0.004 -0.002	0.46 +0.10 -0.05
B ₁	0.060 +0.004 -0.002	1.52 +0.10 -0.05
C	0.010 +0.004 -0.002	0.25 +0.10 -0.05
D	0.360 Typ. (0.380 Max.)	9.14 Typ. (9.65 Max.)
E	0.300 ± 0.010	7.62 ± 0.25
E ₁	0.250 Typ. (0.262 Max.)	6.35 Typ. (6.65 Max.)
e ₁	0.100 ± 0.010	2.54 ± 0.25
L	0.130 ± 0.010	3.30 ± 0.25
α	0° ~ 15°	0° ~ 15°
ea	0.345 ± 0.035	8.76 ± 0.89
S	0.045 Max.	1.14 Max.

Notes:

- (1) The maximum value of dimension D includes end flash.
- (2) Dimension E₁ does not include resin fins.
- (3) Dimension S includes end flash.

31 业务联络

立超电子科技有限公司

中国南京市和燕路251号金港大厦A幢2406室

ZIP:210028

Tel: 0086-25-83306839/83310926

Fax: 0086-25-83737785

Email: Yunchao.Ding@sykee.net

Website:Http://www.dycmcu.com

32 免责声明

规格书中所出现的信息在出版当时相信是正确的，然而本公司对于说明书的使用不负任何责任。文中提到的应用目的仅仅是用来做说明，本公司不保证或表示这些没有进一步修改的应用将是适当的，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。本产品不授权使用于救生、维生器件或系统中做为关键器件。本公司拥有不事先通知而修改产品的权利，对于最新的信息，请参考我们的网址 <http://www.dycmcu.com>