特点概述

- μPSD3251器件包含一个带8032微控制器的FlashPSD结构，带两块Flash存储器，SRAM，通用I/O口，可编程逻辑管理监控功能，并可实现I^2C, ADC and PWM功能，片内8032微控中心, 带两个标准异步通讯口，三个16位定时/计数器和一个外部中断。和其他FlashPSD系列一样。μPSD32××系列同样可以通过JTAG ISP 接口进行在系统编程。
- 具有可选择后备电池的2K字节大容量SRAM
- 具有两片Flash存储器
  - 64K字节主Flash存储器
  - 16K字节次Flash存储器
- 内容加密
  - 阻址访问 Flash 存储器
- 可编程逻辑可为所有存储器提供灵活的地址映射
- 标准8032核心的高速时钟（12周期）
- 为外围电路提供I2C接口
- 五个脉宽调制(PWM)通道（仅80管脚封装）
- 模数转换器（ADC）
- 6个I/O口提供46个I/O引脚
- 带有16个宏单元的3000门可编程逻辑电路（PLD）
- 看门狗定时器监控功能
- 通过 JTAG 口在系统编程
- 零功耗技术
- 单电源供电
  - 4.5 至5.5 V
  - 3.0 至3.6 V

Figure 1. 52-lead, Thin, Quad, Flat Package

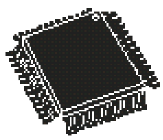
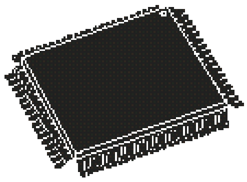


Figure 2. 80-lead, Thin, Quad, Flat Package



目　录

简介

■ 两片Flash存储器

- 当擦除和写一块存储器时读操作只能在另一存储器内进行为了远程更新的在应用编程
- 为应用程序操作系统或图形用户界面位图提供64K字节主Flash存储器
- 16k字节次Flash存储器分为几个扇区. 便于用软件来模拟EEPROM以代替外部EEPROM
- 次Flash存储器容量足以应付实现复杂的通讯协议的在应用编程(IAP)

■ 可选后备电池的大容量SRAM

- 2K字节SRAM给RTOS,高级语言通讯缓冲器和堆栈提供足够的空间

■ 可编程的地址译码PLD为所有存储器提供灵活的地址映射

- 每个Flash和SRAM扇区都可设置为任意地址范围
- 内置页寄存器打破了8032只有64k字节地址空间的限制
- 特殊寄存器(VM)能灵活的转换不同的Flash存储器块为8032微控制器的程序空间和数据空间实现在应用编程IAP

■ 高速的标准8032核心12时钟周期

- 5V操作在40MHz,3.3V操作在24MHz
- 两个异步通讯口三个16位定时/计数器和两个外部中断

■ 为外围器件的连接提供I2C接口

- 可以工作在主从两种方式

■ 五个脉宽调制器PWM 通道

- 四个8位脉宽调制单元
- 一个可编程周期8位脉宽调制单元

■ 4通道，8位带模拟电源的模数转换器（ADC）

■ 超过46个I/O引脚的6个I/O口

- 多功能I/O:GPIO,DDC,I2C,PWM,PLD I/O,监控和JTAG
- 不需要外部锁存和逻辑电路

■ 带有16个宏单元的3000门PLD

- 创建粘合glue 逻辑状态机(state machines) 延时等等
- 不需要外部接PALs,PLDs,和74HCXX电路
- 简单免费的PSDsoft EXPRESS 软件…非常方便

■ 监控功能

- 低电压或看门狗时钟输出产生复位去掉外部监控芯片
- 通过引脚复位，通过PLD复位输出

■ 通过JTAG在系统编程(ISP)

- 在10-25秒内编程整块芯片不需要8032参与过程
- 提高生产效率产品测试更容易有效控制库存
- 去掉芯片插座和预编程部分
- 用FASHLINKTM 电缆和PC机编程

■ 加密安全技术

- 可编程的加密位防止非法访问器件

■ 零功耗技术

- 在输入发生改变的间隙存储器和PLD自动降到静态电流

■ 封装

- 52-脚TQFP封装
- 80-脚TQFP封装允许通过8032地址/数据/控制信号连接外围设备

表1. μPSD32××系列产品矩阵

Part No.	Main Flash (bit)	Sec. Flash (bit)	SRAM (bit)	Macro -Cells	I/O Pins	PWM Ch.	Timer / Ctr	UART Ch.	I^2C	ADC Ch.	DDC	USB	V_{CC}	MHz	Pins
uPSD 3234 A-40	2M	256K	64K	16	50	5	3	2	1	4	yes	yes	5V	40	80
uPSD 3234 BV-24	2M	256K	64K	16	50	5	3	2	1	4	yes		3V	24	80
uPSD 3233 B-40	1M	256K	64K	16	41 or 50	5	3	2	1	4	yes		5V	40	52 or 80
uPSD 3233 BV-24	1M	256K	64K	16	41 or 50	5	3	2	1	4	yes		3V	24	52 or 80

图2. TQFP52 接线

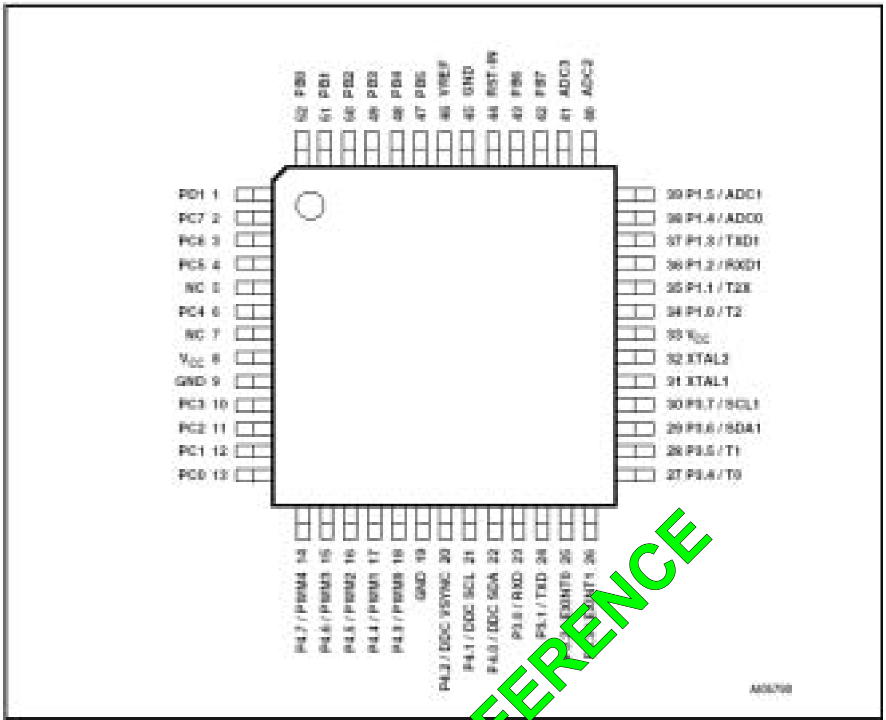


图3. TQFP80 接线

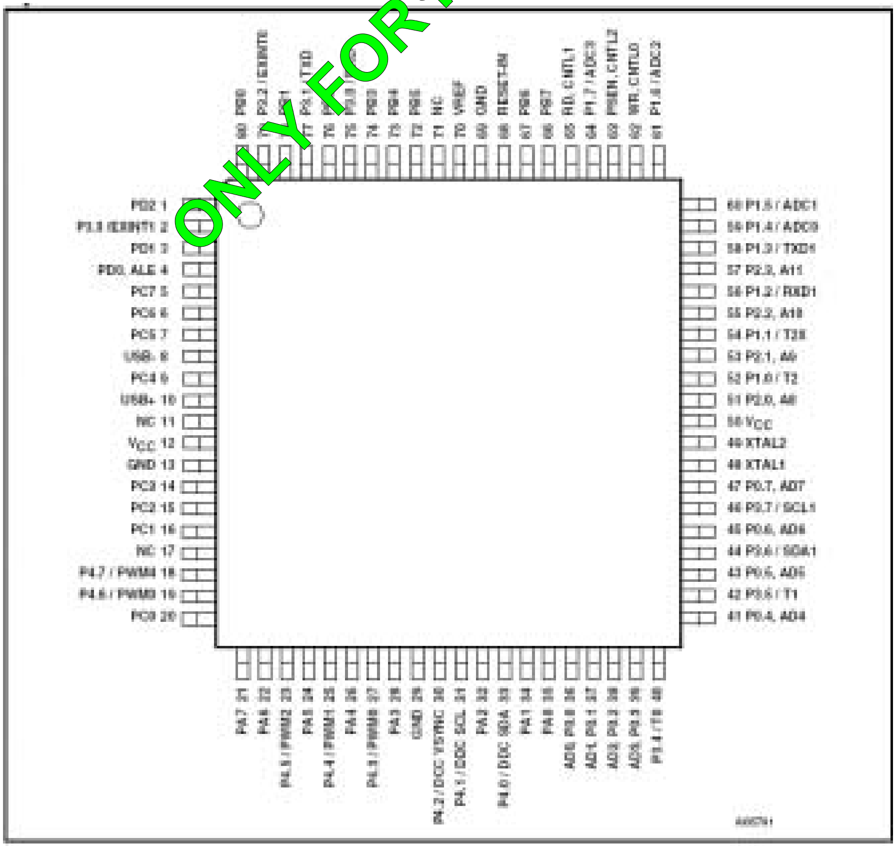


表2. 80脚封装引脚描述

Port Pin	Signal Name	Pin No.	In/Out	Function	
				Basic	Alternate
P0.0	AD0	36	I/O	External Bus Multiplexed Address/Data bus A1/D1	
P0.1	AD1	37	I/O	Multiplexed Address/Data bus A0/D0	
P0.2	AD2	38	I/O	Multiplexed Address/Data bus A2/D2	
P0.3	AD3	39	I/O	Multiplexed Address/Data bus A3/D3	
P0.4	AD4	41	I/O	Multiplexed Address/Data bus A4/D4	
P0.5	AD5	43	I/O	Multiplexed Address/Data bus A5/D5	
P0.6	AD6	45	I/O	Multiplexed Address/Data bus A6/D6	
P0.7	AD7	47	I/O	Multiplexed Address/Data bus A7/D7	
P1.0	T2	52	I/O	General I/O port pin	Timer 2 Count input
P1.1	T2EX	54	I/O	General I/O port pin	Timer 2 Trigger input
P1.2	RxD2	56	I/O	General I/O port pin	2nd UART Receive
P1.3	TxD2	58	I/O	General I/O port pin	2nd UART Transmit
P1.4	ADC0	59	I/O	General I/O port pin	ADC Channel 0 input
P1.5	ADC1	60	I/O	General I/O port pin	ADC Channel 1 input
P1.6	ADC2	61	I/O	General I/O port pin	ADC Channel 2 input
P1.7	ADC3	64	I/O	General I/O port pin	ADC Channel 3 input
P2.0	A8	51	I/O	External Bus, Address A8	
P2.1	A9	53	I/O	External Bus, Address A9	
P2.2	A10	55	I/O	External Bus, Address A10	
P2.3	A11	57	I/O	External Bus, Address A11	
P3.0	RxD1	75	I/O	General I/O port pin	UART Receive
P3.1	TxD1	77	I/O	General I/O port pin	UART Transmit
P3.2	INT0	79	I/O	General I/O port pin	Interrupt 0 input / timer0 gate control
P3.3	INT1	2	I/O	General I/O port pin	Interrupt 1 input / timer1 gate control
P3.4	T0	40	I/O	General I/O port pin	Counter 0 input
P3.5	T1	42	I/O	General I/O port pin	Counter 1 input
P3.6	SDA1	44	I/O	General I/O port pin	I^2C Bus serial data I/O
P3.7	SCL1	46	I/O	General I/O port pin	I^2C Bus clock I/O
P4.0	SDA2	33	I/O	General I/O port pin	I^2C serial data I/O for DDC interface
P4.1	SCL2	31	I/O	General I/O port pin	I^2C clock I/O for DDC interface
P4.2	VSYNC	30	I/O	General I/O port pin	VSYNC input for DDC interface

Port Pin	Signal Name	Pin No.	In/Out	Function	
				Basic	Alternate
P4.3	PWM0	27	I/O	General I/O port pin	8-bit Pulse Width Modulation output 0
P4.4	PWM1	25	I/O	General I/O port pin	8-bit Pulse Width Modulation output 1
P4.5	PWM2	23	I/O	General I/O port pin	8-bit Pulse Width Modulation output 2
P4.6	PWM3	19	I/O	General I/O port pin	8-bit Pulse Width Modulation output 3
P4.7	PWM4	18	I/O	General I/O port pin	16-bit Pulse Width Modulation output
	USB-	8	I/O	USB I/O	
	USB+	10	I/O	USB I/O	
	AVREF	70	O	Reference Voltage input for ADC	
	RD_	65	O	Read signal, external bus	
	WR_	62	O	Write signal, external bus	
	PSEN_	63	O	PSEN signal, external bus	
	ALE	4	O	Address Latch signal, external bus	
	RESET_	68	I	Active low reset input	
	XTAL1	48	I	Oscillator input pin for system clock	
	XTAL2	49	O	Oscillator output pin for system clock	
PA0		35	I/O	General I/O port pin	1. PLD Macro-cell outputs 2. PLD inputs 3. Latched Address Out (A0-A7) 4. Peripheral I/O mode
PA1		34	I/O	General I/O port pin	
PA2		32	I/O	General I/O port pin	
PA3		28	I/O	General I/O port pin	
PA4		26	I/O	General I/O port pin	
PA5		24	I/O	General I/O port pin	
PA6		22	I/O	General I/O port pin	
PA7		21	I/O	General I/O port pin	
PB0		80	I/O	General I/O port pin	1. PLD Macro-cell outputs 2. PLD inputs 3. Latched Address Out (A0-A7)
PB1		78	I/O	General I/O port pin	
PB2		76	I/O	General I/O port pin	
PB3		74	I/O	General I/O port pin	
PB4		73	I/O	General I/O port pin	
PB5		72	I/O	General I/O port pin	
PB6		67	I/O	General I/O port pin	
PB7		66	I/O	General I/O port pin	

Port Pin	Signal Name	Pin No.	In/Out	Function	
				Basic	Alternate
PC0	TMS	20	I/O	General I/O port pin	1. PLD Macro-cell outputs 2. PLD inputs 3. SRAM stand by voltage input (VSTBY) 4. JTAG Interface (TDI, TDO, TMS, TCK, TSTAT, TERR) 5. SRAM battery-on indicator (PC4)
PC1	TCK	16	I/O	General I/O port pin	
PC2	VSTBY	15	I/O	General I/O port pin	
PC3	TSTAT	14	I/O	General I/O port pin	
PC4	TERR	9	I/O	General I/O port pin	
PC5	TDI	7	I/O	General I/O port pin	
PC6	TDO	6	I/O	General I/O port pin	
PC7		5	I/O	General I/O port pin	
PD1	CLKIN	3	I/O	General I/O port pin	1. PLD I/O 2. Clock input to PLD and APD
PD2	CSI	1	I/O	General I/O port pin	1. PLD I/O 2. Chip select to PSD Module
Vcc		12			
Vcc		50			
GND		13			
GND		29			
GND		69			
NC		11			
NC		17			
NC		71			

52 脚封装I/O口

μPSD32××系列中52脚封装和80脚封装的器件除去以下端口外有相同的端口引脚

- 端口0 P0.0-P0.7,外部地址/数据总线AD0-AD7
- 端口2 P2.0-P2.3,外部地址总线A8-A11
- 端口A(PA0-PA7)
- 端口D(PD2)
- 总线控制信号（RD,WR,PSEN,ALE）
  管脚5要求上拉电阻（3V器件2K，5V器件7.5K）
- 管脚4无PWM支持

结构概图

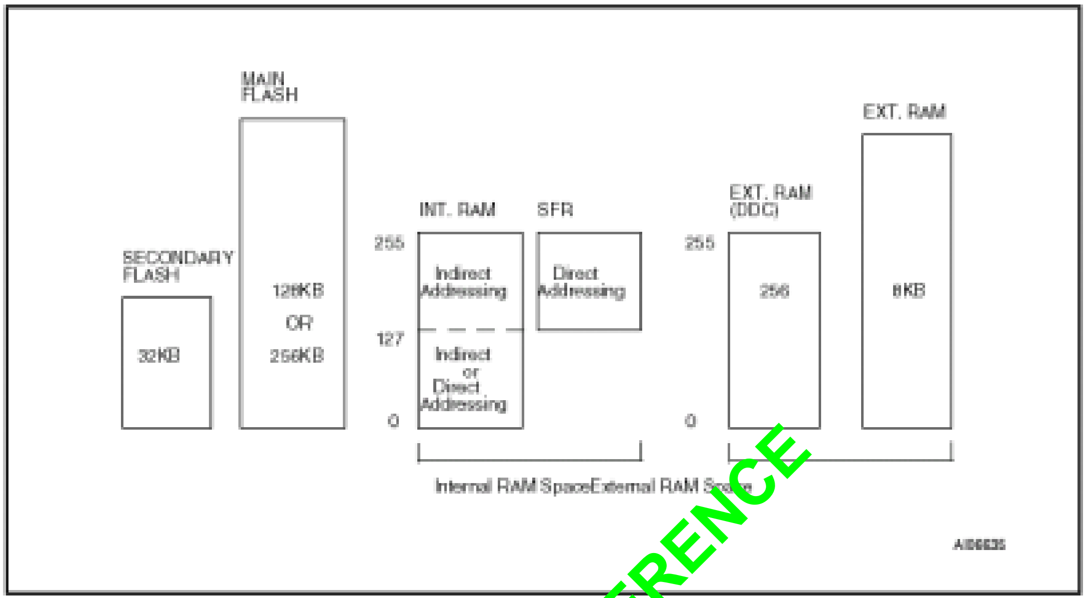
存储器组成

μPSD32××系列的标准8032核心有独立的64K程序空间和64K数据空间程序存储器。程序存储器用于8032执行指令，数据存储器用于保存数据变量。FLASH存储器可以映射到程序和数据 空间。FLASH存储器由两块组成：主FLASH存储器（512K）和第二FLASH（128k）存储器。

除了在Flash存储器编程和更新期间外，程序存储器只能被读不能被写。超出64k字节地址空间通过页寄存器来访问。详细请参考PSD模块Flash存储器映射图。8032微控制器核有两种存储器（内部和外部）。内部有256字节RAM 包括堆栈区。SFR 特殊功能寄存器占用256字节内部RAM的128个字节寄存器仅能

被直接地址访问。PSD模块另外2K字节可以根据拥护定义映射到任何地址空间。

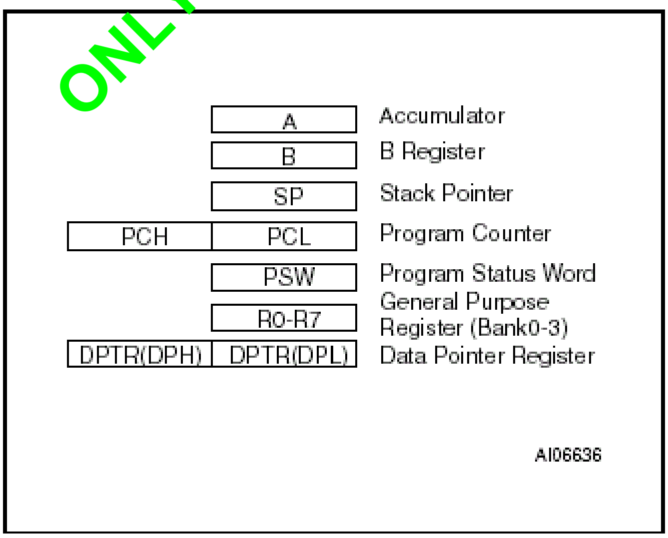
图4. 存储器示意图和地址空间



寄存器

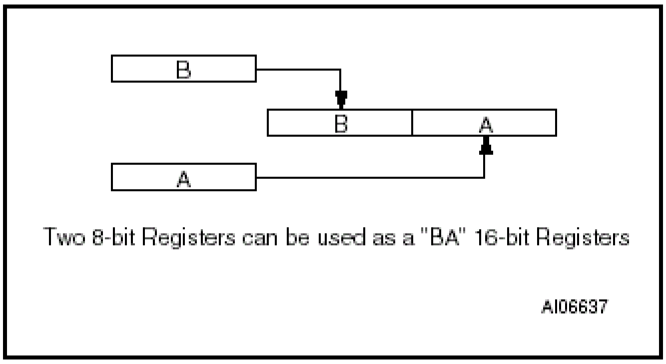
该器件寄存器包括程序计数器（PC），累加器（A），B寄存器（B），堆栈指针（SP），程序状态字（PSW），通用寄存器（R0-R7）和DPTR（数据指针）。

图5. 寄存器结构



累加器. 累加器是8位通用寄存器，用于数据传递，临时存储，条件判断等数据操作。累加器接在B寄存器后面可以构成一个16位的寄存器。

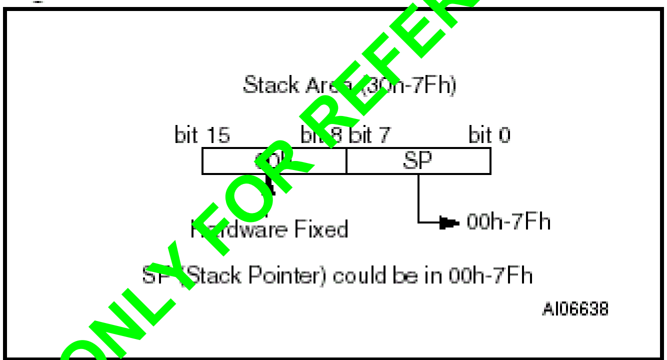
图6. BA 16位寄存器的构成



B 寄存器. B寄存器是8位通用寄存器，用于和累加器一起，进行如乘、除的数学运算。

堆栈指针. 堆栈指针是一个8位寄存器，用于中断事件和子程序调用。堆栈指针确定堆栈内被访问的位置（保存或再存放）。堆栈可位于内部数据存储器0000h至007Fh间的任意位置。SP不能被硬件初始化，是在初始化程序写入初始值（堆栈的起始位置），通常，初始值用07h时,堆栈区域是07h至7Fh。

图7. 堆栈指针



程序计数器. 程序计数器是一个16位，包含两个8位寄存器，PCH和PCL。计数器的值指出执行的下一个指令的地址。在复位状态，程序计数器被复位到固定地址（PCH:0FFh, PCLL:0FEh）。

程序状态字. 程序状态字（PSW）包含了几个位，这几位，反映了当前CPU的状态和当前选择的内部RAM（00h至1Fh: 0组至3组）。PSW各位，在图8中描述。它包含进位标志、辅助进位标志、半进位（BCD操作）、用户标志位、寄存器组选择位、溢出标志、未定义的标志和奇偶标志。

[进位标志CY]. 用于保存数学运算后CPU的ALU所有的进位或没借位，同时它也可以被转移或循环指令改变。

[辅助进位标志AC]. 操作后，当ALU的第3位有进位，或ALU的第4位无借位时，该位被设置

[寄存器组选择标志RS0,RS1]. 该标志选择内部四组RAM中的一组（00-07H: 0组，08-0Fh:1组，10-17h:2组，18-1Fh:3组）。

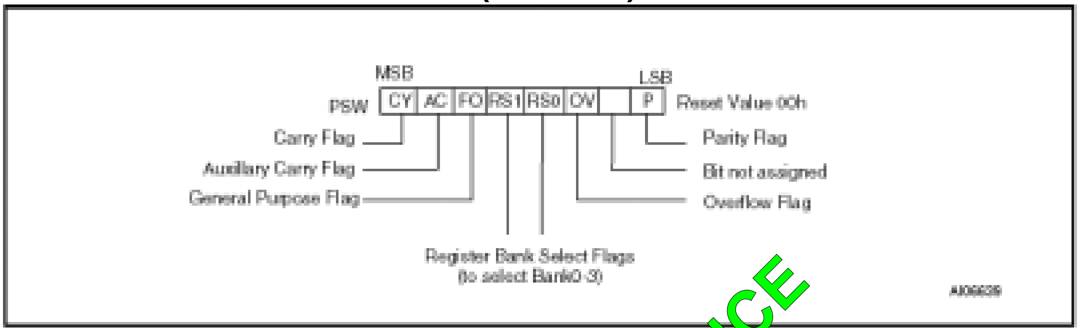
[溢出标志OV]. 作为一个数学运算操作所包含的符号，当有溢出出现时，这个标志被置1。当加法或减法的运算结果超出+127（7Fh）或-128（80h）时，将出现溢出。CLRV指令可以清除溢出标志。没有设置溢出标志的指令。当执行位指令(BIT)时，存储器的第6位被复制到这个标志。

[奇偶标志P]. 这个标志用于指示累加器的1的个数。如果累加器中1的个数是奇数，则P=0，是偶数，则P=1.

R0-R7. 通用寄存器

数据指针寄存器. 数据指针寄存器是包含两个8位寄存器DPH和DPL的16位寄存器。它作为外部数据存储器数据传输的数据指针。

图8. PSW(程序状态字)寄存器



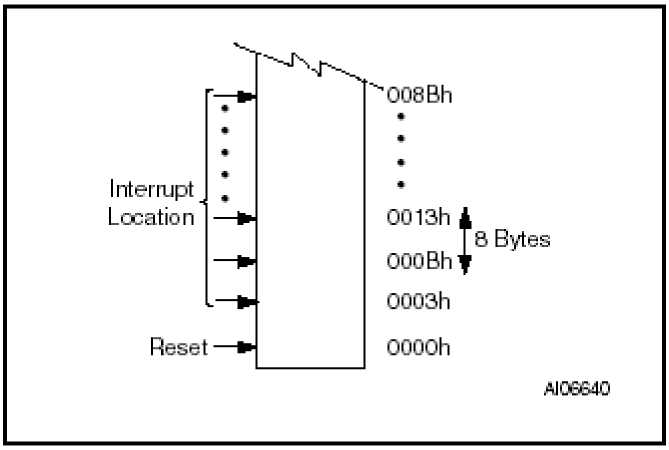
程序存储器

程序存储器包含两片Flash存储器：128K字节（或256K字节）主Flash存储器和32K字节的次Flash存储器。Flash存储器能被用户通过PSD软件工具定义映射为任意地址空间。它也可以在Flash存储器更新和编程时被映射成数据存储器空间。

复位后，CPU从0000h开始执行。如图9表示，每个中断在程序存储器中被分配一个固定的位置。中断使CPU跳转到该位置，开始执行中断服务程序。例如外部中断0，被指定为地址0003h。如果不使用外部中断0，它的服务地址可作为通用程序存储器。

中断服务地址以8字节为间隔：外部中断0是0003h，定时器0是000Bh，外部中断1是0013h，定时器1是001Bh等等。如果中断服务程序足够短，可以把它放在这8个字节的间隙中。如果其他中断也在使用，那么，长的服务程序可以使用跳转指令，跳到中断服务子程序。

图9. 程序存储器中断地址



数据存储器

内部数据存储器分为4个物理上分离的块：256字节内部RAM，128字节特殊功能寄存器（SFRs）区域，256字节外部RAM（SRAM-DDC）和在PSD模块内部的8K字节（XRAM-PSD）。

RAM

四个寄存器组，每组8个寄存器，占用低位RAM区域0到31个寄存器。在某一时刻只能选定一组。地址32至47的16个字节，包含128个可直接寻址的位地址。堆栈深度仅受限于有效的256字节的内部RAM空间。

XRAM-PSD

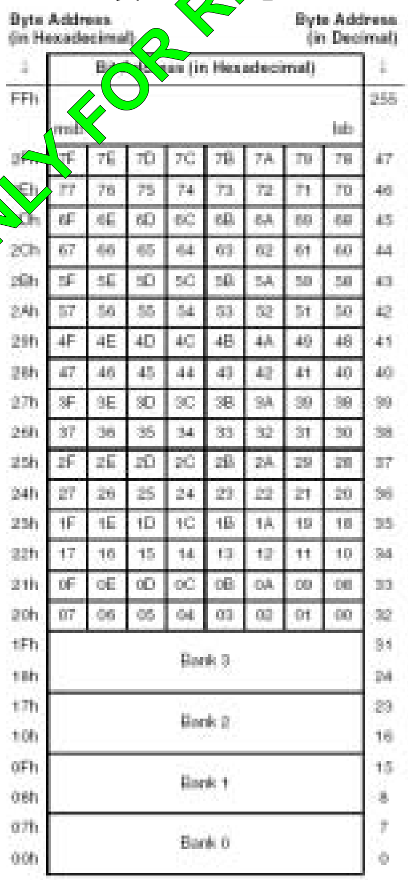
PSD模块内的8K字节的XRAM-PSD，用户可用PSDsoft开发工具定义，通过DPLD（PLD解码）映射成任意地址空间。XRAMPSD有一个后备电池特征，允许掉电后数据仍然保留。电池连接到端口C的PC2引脚。

特殊功能寄存器

特殊功能寄存器可以在128到255的地址范围内被直接寻址。表15给出了特殊功能寄存器的概图。

特殊功能寄存器空间，有16个地址是双字节和可以设定位地址的。可以设定位地址的特殊功能寄存器是低位地址为0h和8h的。这个区域的位地址是80h至FFh。

表3. RAM地址



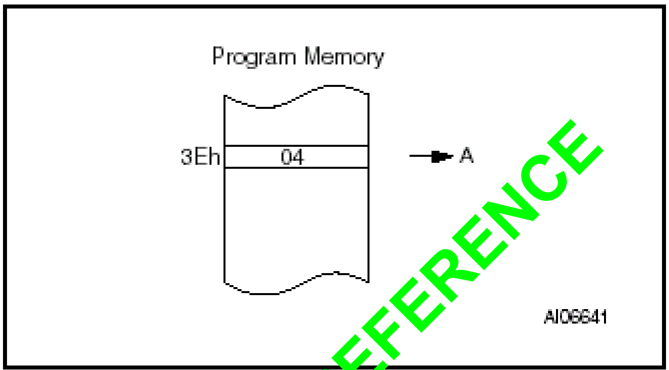
Byte Address (in Hexadecimal)	Bit Address (in Hexadecimal)								Byte Address (in Decimal)
FFh									255
	msb							lsb	
2Fh	7F	7E	7D	7C	7B	7A	79	78	47
2Eh	77	76	75	74	73	72	71	70	46
2Dh	6F	6E	6D	6C	6B	6A	69	68	45
2Ch	67	66	65	64	63	62	61	60	44
2Bh	5F	5E	5D	5C	5B	5A	59	58	43
2Ah	57	56	55	54	53	52	51	50	42
29h	4F	4E	4D	4C	4B	4A	49	48	41
28h	47	46	45	44	43	42	41	40	40
27h	3F	3E	3D	3C	3B	3A	39	38	39
26h	37	36	35	34	33	32	31	30	38
25h	2F	2E	2D	2C	2B	2A	29	28	37
24h	27	26	25	24	23	22	21	20	36
23h	1F	1E	1D	1C	1B	1A	19	18	35
22h	17	16	15	14	13	12	11	10	34
21h	0F	0E	0D	0C	0B	0A	09	08	33
20h	07	06	05	04	03	02	01	00	32
1Fh – 18h	Bank 3								31 – 24
17h – 10h	Bank 2								23 – 16
0Fh – 08h	Bank 1								15 – 8
07h – 00h	Bank 0								7 – 0

寻址方式

μPSD3200系列的指令寻址模式如下：

- 直接寻址
- 间接寻址
- 寄存器寻址
- 特殊寄存器寻址
- 立即寻址
- 变址寻址

图10. 直接寻址



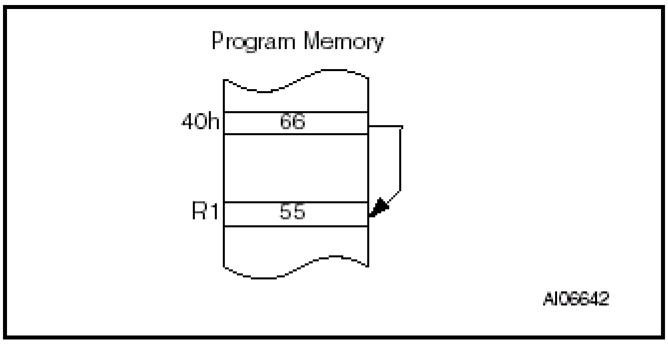
（1）直接寻址.

直接寻址时，指令中，操作数被一个8位地址单元指定。只有内部数据RAM和特殊功能寄存器（80-FFh RAM）可以直接寻址。

例如：

mov A,3EH;A←--RAM(3E)

图11. 间接寻址



（2）间接寻址.

间接寻址时，指令指定一个包含操作数地址的寄存器。两个内部和外部寄存器可以间接寻址。8位地址的地址寄存器，可以是选定寄存器组的R0或R1,也可以是堆栈指针。16位地址寄存器只能是16位的数据指针寄存器DPTR。

例如：

mov @R1,40H;[R1]←--[40H]

（3）寄存器寻址.

寄存器组，包含寄存器R0到R7,可以被某些指令（指令操作码包含3位寄存器标志，以指定所访问的寄存器R0---R7）。因为这种寻址模式去掉了地址字节，所以用这种方法访问寄存器的指令是代码有效。

例如：

mov PSW,#0001000B;选择0组

mov A,#30h

mov R1,A

（4）特殊寄存器寻址.

一些指令针对于一些确定的寄存器。例如：一些指令总是在累加器或数据指针上操作等等，所以，不需要给它指定地址字节。只需要操作数。

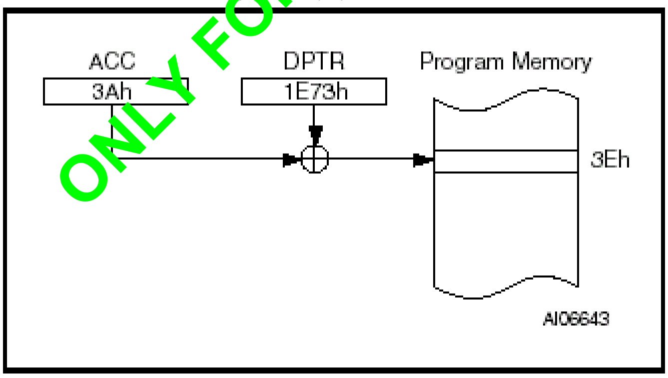
（5）立即寻址.

程序存储器中，操作马后紧跟一个常数。

例如：

mov A,#100H

图12. 变址寻址



（6）变址寻址.

变址寻址只能访问程序存储器，而且只能被读。这种模式特别为在程序存储器中查表读取数据而设置。一个16位基寄存器（DPTR或PC）指向表的基地址，累加器里置的是表的入口数据，程序存储器中表的入口地址是累加器里的数据和基地址之和。

例如：

mov c A,@A+DPTR

表4. 算术指令

Mnemonic	Operation	Addressing Modes			
		Dir.	Ind.	Reg.	Imm
ADD A,<byte>	A = A + <byte>	X	X	X	X
ADDC A,<byte>	A = A + <byte> + C	X	X	X	X
SUBB A,<byte>	A = A − <byte> − C	X	X	X	X
INC	A = A + 1	Accumulator only			
INC <byte>	<byte> = <byte> + 1	X	X	X	
INC DPTR	DPTR = DPTR + 1	Data Pointer only			
DEC	A = A − 1	Accumulator only			
DEC <byte>	<byte> = <byte> − 1	X	X	X	
MUL AB	B:A = B x A	Accumulator and B only			
DIV AB	A = Int[A / B] B = Mod[A / B]	Accumulator and B only			
DA A	Decimal Adjust	Accumulator only			

算术指令

算术指令列在表4里。表中指出了寻址模式，和指令能够访问的<byte>操作数。例如：

ADD A, <byte> 指令能够写成：

ADD a,7FH(直接寻址)

ADD A,@R0(间接寻址)

ADD a,R7(寄存器寻址)

ADD A,#127(立即寻址)

注意：内部数据存储器空间的任意字节的增加，都可以不通过累加器A。

INC指令，可以操作16位的数据指针。数据指针可用于为外部存储器产生16位地址，所以，在一个16位操作中能够加一是个很有用的特点。

MUL AB指令，是累加器A的值与B寄存器的值相乘，16位的乘积连续放入B和A寄存器。

DIV AB指令，是累加器A的值被B寄存器的值除，结果，8位的商放入A，8位的余数放入B。有趣的是，DIV AB在算术运算中，作“除法”应用的情况比在根的变换和可编程移位操作中的应用要少些。

后面将给出在根变换中使用DIV AB的示例。在移位操作中，一个数除以2n等于该数右移了n位用DIV AB指令来执行除法操作完成A值的移位同时B寄存器保留移出去的位。DA A指令用于BCD码的数学运算操作。在BCD数学运算操作中，ADD和ADDC指令后面总是跟着DA A操作。确保结果为BCD码。注意：DA A指令不是将二进制数变成BCD码。DA A操作只有紧跟在BCD码相加后才有意义

表5. 逻辑指令

Mnemonic	Operation	Addressing Modes			
		Dir.	Ind.	Reg.	Imm
ANL A,<byte>	A = A .AND. <byte>	X	X	X	X
ANL <byte>,A	A = <byte> .AND. A	X			
ANL <byte>,#data	A = <byte> .AND. #data	X			
ORL A,<byte>	A = A .OR. <byte>	X	X	X	X
ORL <byte>,A	A = <byte> .OR. A	X			
ORL <byte>,#data	A = <byte> .OR. #data	X			
XRL A,<byte>	A = A .XOR. <byte>	X	X	X	X
XRL <byte>,A	A = <byte> .XOR. A	X			
XRL <byte>,#data	A = <byte> .XOR. #data	X			
CRL A	A = 00h	Accumulator only			
CPL A	A = .NOT. A	Accumulator only			
RL A	Rotate A Left 1 bit	Accumulator only			
RLC A	Rotate A Left through Carry	Accumulator only			
RR A	Rotate A Right 1 bit	Accumulator only			
RRC A	Rotate A Right through Carry	Accumulator only			
SWAP A	Swap Nibbles in A	Accumulator only			

逻辑指令

表5列出了μPSD3200系列的逻辑指令。这些指令完成布尔操作（AND, OR, 异OR, NOT）,在位与位的基础上由字节来完成操作。就是说，如果A的值是00110101B和值为01010011B的字节作AND(与)运算，那么，ANL A,<byte>,将使A的值为00010001B。这种寻址模式可以访问的操作数<byte>，列在表5里面了。ANL A,<byte> 指令可以是下面任意形式：

ANL A,7FH（直接寻址）

ANL A,@R1（间接寻址）

ANL A,R6（寄存器寻址）

ANL A,#53H（立即寻址）

注意：对内部数据存储器空间作布尔操作，不需要A寄存器。例如：XRL <byte>，#data指令，提供了一种方便的端口位的反相方法，如：XRL P1,#0FFH。如果操作在响应一个中断，在中断服务程序中，不使用A累加器可节约时间，并且在服务程序中尽量把它压入堆栈。环移指令（RL A, RLC A,等等）将A里的一位左移或右移。对于左环移，最高位移入最低位位置。对于右环移，最低位移入最高位位置。SWAP A指令是累加器A的高4位和低4位互换，这个指令在BCD码处理中很有用。例如：如果A里有一个小于100的二进制数，它可以由下面的代码快速转换成BCD码：

```
MOV B,#10
DIV AB
SWAP A
ADD A,B
```

除以10，十位留在A累加器的低4位。个位在B寄存器里，SWAP和ADD指令把十位移到A累加器的高

4位，个位加到低4位。

表6. 访问内部数据存储空间的数据传输指令

Mnemonic	Operation	Addressing Modes			
		Dir.	Ind.	Reg.	Imm
MOV A,<src>	A = <src>	X	X	X	X
MOV <dest>,A	<dest> = A	X	X	X	
MOV <dest>,<src>	<dest> = <src>	X	X	X	X
MOV DPTR,#data16	DPTR = 16-bit immediate constant				X
PUSH <src>	INC SP; MOV "@SP",<src>	X			
POP <dest>	MOV <dest>,"@SP"; DEC SP	X			
XCH A,<byte>	Exchange contents of A and <byte>	X	X	X	
XCHD A,@Ri	Exchange low nibbles of A and @Ri		X		

数据传送

内部RAM. 表6给出了内部存储器空间内可以移动数据的指令菜单，和可以被各指令使用的寻址模式。MOV <dest>,<src>指令允许数据在不通过A的情况下，在任意两个内部RAM和特殊寄存器之间传送数据。记住，高128字节的数据RAM只能间接寻址，特殊寄存器只能直接寻址。注意，μPSD3200系列器件，堆栈位于芯片上的RAM,并且向上增长。PUSH指令首先使堆栈指针增加，然后将字节拷贝到堆栈。PUSH和POP只能直接寻址，确定字节是被存储还是被恢复，但堆栈自己却是通过SP寄存器间接寻址。这就是说，堆栈可以进入RAM的高128字节，如果这么做，堆栈的数据就不会进入特殊功能寄存器空间。数据传送指令包含一个16位的MOV指令，它用于查询程序存储器里的数据表时，初始化数据指针（DPTR）。

XCH A,.<byte>指令是使A与给定地址的字节交换数据。XCHD A, @Ri指令功能也类似，不过只交换低4字节。看看XCH和XCHD如何帮助处理数据，首先考虑移位的问题和8位数的BCD数向右移两位。

表8给出了如何使用XCH指令。为了理解代码如何工作，存放BCD数的寄存器和A寄存器的内容，在每个指令的旁边给出，指示出指令执行后各寄存器的状态。

这段程序执行完后，A包含了右移出去的两个数。完成相同的功能，直接用MOV指令，需要用14个字节的代码，而用XCH指令只需9字节，速度几乎快了一倍。右移奇数位，首先必须移动一位。

表9给出了用XCHD 指令将BCD数右移一位的示例程序。同样，存放数据的寄存器和A寄存器的内容，也在每条指令的旁边给出。

表7. 将一个BCD数向右移两位（用指令MOVs: 14字节）

	2A	2B	2C	2D	2E	ACC
MOV A,2Eh	00	12	34	56	78	78
MOV 2Eh,2Dh	00	12	34	56	56	78
MOV 2Dh,2Ch	00	12	34	34	56	78
MOV 2Ch,2Bh	00	12	12	34	56	78
MOV 2Bh,#0	00	00	12	34	56	78

首先，设置指针R1和R0，指向包含最后 4 位BCD数的两个字节单元。然后，执行一个循环，让最后

一个字节，即2EH，存放移动后的最后两个数。指针减，指针指向2DH，循环执行。CJNE指令（比较，不相等则跳转）是一个循环控制指令，后面将会具体描述。循环从LOOP执行到CJNE，指针R1分别指向2EH、2DH、2CH、2BH。循环完，最初向右移出的那个数，现在已经传到了2AH。最后，2AH被置成 0，丢掉的数移入A。

表8. 将一个BCD数向右移两位（用指令XCHs: 9字节）

		2A	2B	2C	2D	2E	ACC
CLR	A	00	12	34	56	78	00
XCH	A,2Bh	00	00	34	56	78	12
XCH	A,2Ch	00	00	12	56	78	34
XCH	A,2Dh	00	00	12	34	78	56
XCH	A,2Eh	00	00	12	34	56	78

表9. 将一个BCD数向右移一位

		2A	2B	2C	2D	2E	ACC
MOV	R1,#2Eh	00	12	34	56	78	xx
MOV	R0,#2Dh	00	12	34	56	78	xx
; loop for R1 = 2Eh							
MOV	A,@R1	00	12	34	56	78	78
XCHD	A,@R0	00	12	34	58	78	76
SWAP	A	00	12	34	58	78	67
MOV	@R1,A	00	12	34	58	67	67
DEC	R1	00	12	34	58	67	67
DEC	R0	00	12	34	58	67	67
CNJE	R1,#2Ah,LOOP	00	12	34	58	67	67
; loop for R1 = 2Dh		00	12	38	45	67	45
; loop for R1 = 2Ch		00	18	23	45	67	23
; loop for R1 = 2Bh		08	01	23	45	67	01
CLR	A	08	01	23	45	67	00
XCH	A,2Ah	00	01	23	45	67	08

表10. 访问外部数据存储器空间的数据传送指令

Address Width	Mnemonic	Operation
8 bits	MOVX A,@Ri	Read external RAM @Ri
8 bits	MOVX @Ri,A	Read external RAM @Ri
8 bits	MOVX A,@DPTR	Read external RAM @DPTR
16 bits	MOVX @DPTR,a	Read external RAM @DPTR

外部RAM. 表10列出了访问外部数据存储器的数据传送指令。只能使用间接寻址。要么选择使用一字节地址，@Ri，Ri是选定寄存器组的R0或R1；要么使用两字节地址，@DPTR。如果仅有少量外部RAM，而使用16位地址访问，需要使用端口2的8位数据总线，这样就不太好，此时，使用8位地址模式，便可以不用牺牲掉端口2的所有引脚。

注意：访问所有的外部数据RAM，A要么是数据源，要么是数据目的地址。

表11. 访问内部（??）数据存储器空间的数据传送指令

Mnemonic	Operation
MOVC A,@A+DPTR	Read program memory at (A+DPTR)
MOVC A,@A+PC	Read program memory at (A+PC)

查表. 表11给出了查找读取程序存储器的数据表的两个有效指令。因为这两个指令只能访问程序存储器，所以，查表只能是读取数据，而不能更改数据。

助记符MOVC的意思是“移动常数”。表11中的第一个MOVC指令适用于256个数的表，0到255。希望访问的数的位置放入A，数据指针指向表的起始位置。那么，

MOVC A,@A+DPTR

将表中希望访问的数据复制到A。另外一个MOVC指令除了是用程序计数器（PC）作为表的起始位置，而表是通过一个子程序来访问以外，其他的执行方法相同。首先，希望访问的数的位置放入A，然后调用子程序

MOV A , ENTRY NUMBER

CALL TABLE

子程序“TABLE”像这样：

TABLE: MOVC A,@A+PC

RET

紧接着RET（返回）指令的表，是程序存储器。这个表有255个数，1到255。不能使用0，因为MOVC指令的执行，PC值包含了RET指令地址。位置0是RET操作码自身。

表12. 布尔指令

Mnemonic	Operation
ANL C,bit	C = A .AND. bit
ANL C,/bit	C = C .AND. .NOT. bit
ORL C,bit	C = A .OR. bit
ORL C,/bit	C = C .OR. .NOT. bit
MOV C,bit	C = bit
MOV bit,C	bit = C
CLR C	C = 0
CLR bit	bit = 0
SETB C	C = 1
SETB bit	bit = 1
CPL C	C = .NOT. C
CPL bit	bit = .NOT. bit
JC rel	Jump if C =1
JNC rel	Jump if C = 0
JB bit,rel	Jump if bit =1
JNB bit,rel	Jump if bit = 0
JBC bit,rel	Jump if bit = 1; CLR bit

布尔指令

μPSD3200系列器件包含一个完整的布尔（位信号）处理器。内部RAM一页就包含了128个位地址，SFR空间也能提供128个位地址。所有的口线都是位地址，每个都可以作为单独的信号位端口来处理。访问这些位的指令不仅仅包含条件转移指令，也包含移动，置1，清0，还附加有“或”和“与”指令。这种位操作在其他面向字节的器件中都是不容易实现的。基于布尔处理器的指令在表12中列出了。所有的位都是直接寻址。位地址00h到7Fh在低128字节中，80h到FFh在SFR空间。

注意：一个内部标志被移到端口引脚是非常容易的：

MOV C,FLAG

MOV P1.0,C

在这个例子中，FLAG是低128字节或SFR空间中的任意位地址。I/O线（在这里，是端口1的最低位）置1或清0，完全依靠标志位FAGL是1或0。PSW中的进位位被用作布尔处理器的A的单独的位。把进位位表示成C的位指令，构成特殊进位位指令（如CLR C等等）。因为进位位在PSW寄存器里，而PSW寄存器可以位寻址，所以进位位也有一个直接地址，

注意：布尔指令包含ANL和ORL操作，但是没有XRL（异或）操作。在软件里实现异或操作很简单。设想一下，如，必须设计两位异或的软件：

C =bit 1 XRL bit2

软件可以如下设计：

MOV C, bit1

JNB bit2, OVER

CPL C

OVER:（继续）

首先，bit1移入C，如果bit2=0，那么此时C包含了正确的结果。这就是说，如果bit2=0，bit1 XRL bit2 =bit1。相反，如果bit2 =1，现在C里的值和正确的结果相反。仅仅需要一个取反，就完成了操作。

这儿，代码用了JNB指令，位测试指令中的一个，如果被寻址的位是1（JC,JB,JBC）则跳转，或者是如果被寻址的位是0（JNC,JNB）则跳转。对于上面的程序，bit2被测试，如果bit2=0，则CPL C指令被跳过。

如果被寻址的位是1，JBC指令在跳转的同时将C清零。因此在一个操作中可同时对位进行测试和清零。PSW的所有位都可直接设置地址，所以奇偶位或用户位在位测试指令中都是有效的。

相对偏移

这些跳转的目的地址，对汇编程序来说，都是被程序存储器里的一个表或一个实际地址来指定的。然而，目的地址汇编成一个相对偏移字节。这是一个有符号的偏移字节，如果跳转执行，它从正负两个方向添加到PC值里。因此,跳转范围是：相对于指令的下一个字节，-128至+127个字节。

表13. 无条件跳转指令

Mnemonic	Operation
JMP addr	Jump to addr
JMP @A+DPTR	Jump to A+DPTR
CALL addr	Call subroutine at addr
RET	Return from subroutine
RETI	Return from interrupt
NOP	No operation

跳转指令

表13列出了无条件跳转指令。表列出了一个单独的“JMP add”指令，事实上，这儿应该是有三个跳转指令SJMP,LJMP和AJMP,它们在目的地址的安排上是不同的。JMP是通常意义的助记符，如果程序员不考虑跳转语句的编码（即代码占用几个字节）时，可以使用它。SJMP指令编译时，把目的地址作为一个相对偏移，如前面所描述的那样。指令长为2个字节，由操作码和相对偏移量组成。跳转被限制在相对紧跟在SJMP后面的指令-128至+127个字节范围内。LJMP指令按16位常数编译目的地址。指令长3字节，由一个操作码和两字节的地址组成。目的地址可以是64K程序存储空间内的任意地址。AJMP指令按11位常数编译目的地址。指令长2字节，由一个操作码和另一个字节组成，操作码包含了11位地址的前3位，另一个字节则是11位地址的低8位。指令执行时，11 位地址直接放入PC寄存器的低11位，高5位仍保持不变。因此，跳转范围是：与AJMP指令的下一条指令在同一个2K字节的块里。

在所有的情况下，程序员都是以相同的方法对汇编程序指定目的地址，一个表或一个16位的常数。汇编程序将按正确的格式把目的地址送给指令。如果指令要求的格式不支持指定的目的地址的距离，在列表文件内将给出错误信息“目的地址跳出范围”。JMP @A+DPTR指令支持这些出错的跳转。执行时，目的地址是16位DPTR寄存器和A累加器的和。典型用法是，DPTR设置为跳转表的地址。例如，一个有5个分支的程序里，整数0到4放入A。执行代码可以如下：

MOV DPTR,#JUMP TABLE

MOV A,INDEX_NUMBER
RL A
JMP @A+DPTR

指令RL A使索引数（0～4）变成0～8的偶数，因为跳转表的每一个入口都是2字节长。

JUMP TABLE:
AJMP CASE 0
AJMP CASE 1
AJMP CASE 2
AJMP CASE 3
AJMP CASE 4

表13给出了一个单独的“CALL addr”指令，其实应该是两个指令LCALL和ACALL，它们的区别是送给CPU的子程序地址的格式不同。CALL是通常意义的助记符，如果程序员不考虑跳转语句的编码（即代码占用几个字节）时，可以使用它。LCALL指令使用的是16位数据格式，子程序可以在64K程序存储器空间的任何地方。ACALL指令使用11位格式，子程序必须与ACALL指令的下一条指令在同一个2K字节的块里。

在所有的情况下，程序员都是以相同的方法对汇编程序指定子程序地址，一个表或一个16位的常数。汇编程序将按正确的格式把地址送给指令。

子程序以RET指令结束，返回CALL的下一条指令。

RETI是中断服务程序的返回指令。RET和RETI的区别是：RETI告诉中断控制系统，程序中的中断已经执行了。如果程序中没有中断而却执行了RETI指令，此时的RETI指令的作用等同于RET的作用。

表14. 条件跳转指令

Mnemonic	Operation	Addressing Modes			
		Dir.	Ind.	Reg.	Imm
JZ rel	Jump if A = 0	Accumulator only			
JNZ rel	Jump if A ≠ 0	Accumulator only			
DJNZ <byte>,rel	Decrement and jump if not zero	X		X	
CJNE A,<byte>,rel	Jump if A ≠ <byte>	X			X
CJNE <byte>,#data,rel	Jump if <byte> ≠ #data		X	X	

表14给出了μPSD3200系列使用的有效的条件跳转。所有的这些跳转都是通过相对偏移模式指定目的地址的，所以跳转距离限制为相对于条件跳转指令的下一条指令-128到+127个字节。然而，特别注意：用户对汇编程序指定实际的目的地址方法与其他跳转相同：一个表或一个16位的常数。PSW里没有“零”位，在需要测试“零”的情况下，可用JZ和JNZ指令测试A里的数。

DJNZ指令（减，如果不为零则跳转）用于循环控制。执行一个N次循环，N载入计数器，中止这个已经开始循环的DJNZ循环，如下面给出的，N=10：

MOV COUNTER,#10
LOOP:(begin loop)
.

.

.

(end loop)

DJNZ COUNTER,LOOP

(continue)

CJNE指令（比较，如果不等就跳转）也能够用于循环控制，象表9那样。两个字节被指定在指令的操作数单元。如果两个字节不相等，就执行跳转。在表 9的例子里，把BCD数向右移一位，两个字节就是R1里的数和常数2AH。R1的初值是2EH。循环每执行一次，R1就减一，继续循环，直到R1的值为2AH。

这个指令的其他应用是“大于，小于”比较。操作数单元里的两个字节被作为无符号整数，如果第一个数小于第二个数，则进位位被置为1。如果第一个数大于或等于第二个数，则进位位被清0。

机器周期

一个机器周期包括6个状态时序，S1到S6。一个机器周期的变化是依照SC-MOD寄存器的值。参照图13。

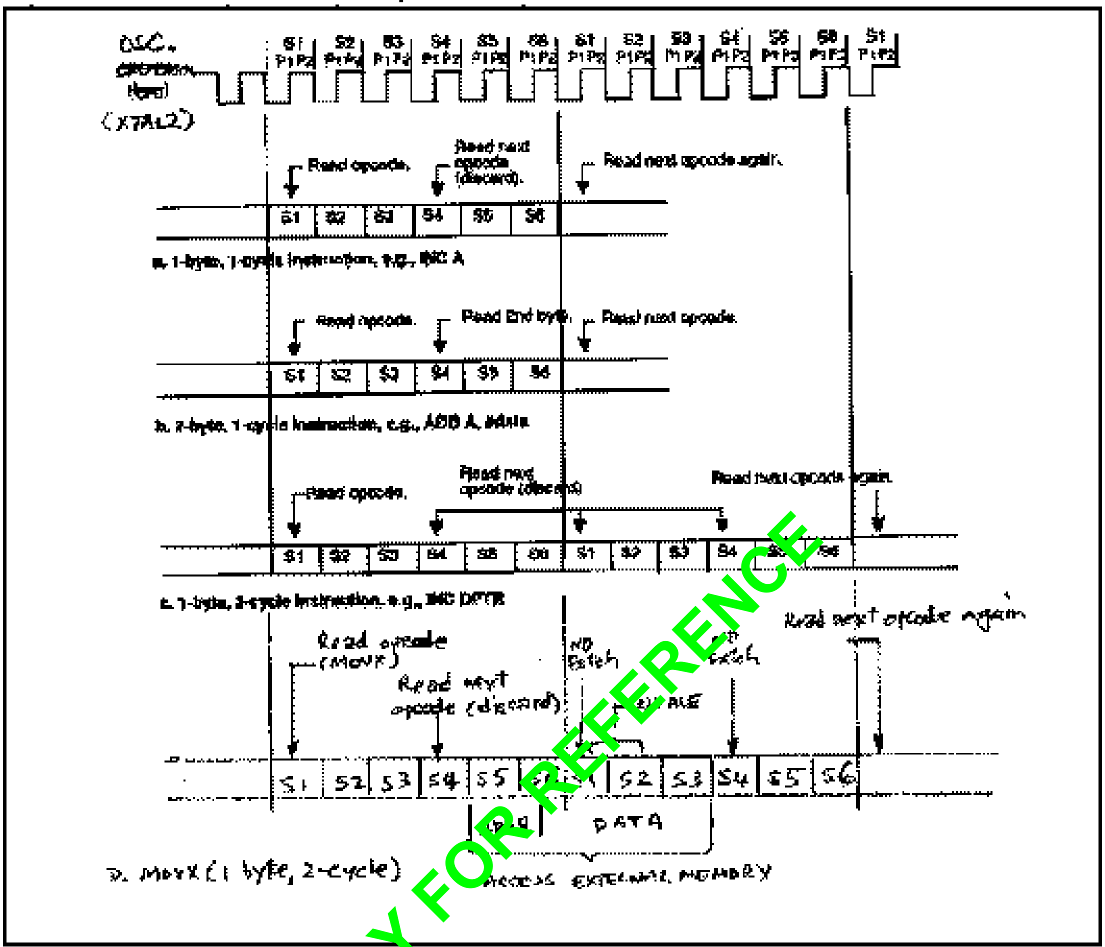
每个状态又分为P1和P2两个拍。μPSD3200系列器件的状态时序给出了不同种类指令取址和执行时的状态和拍的时序。

通常，每个机器周期要取两次数据，即便指令执行不要求取两个数。如果指令执行不需要更多的代码字节，CPU会完全忽视多余的取数，程序指针不增加。

指令周期（图13）的执行在机器周期的状态1（S1）开始。此时操作码送入指令寄存器。第二步取数在同一个机器周期的状态4（S4）。指令在机器周期的状态6（S6）执行完毕。

MOVX指令的执行需要两个机器周期。程序取址没有产生在MOVX指令的第二个机器周期。只有这个指令程序取址被跳过。图13给出了MOVX指令的取址和执行时序。

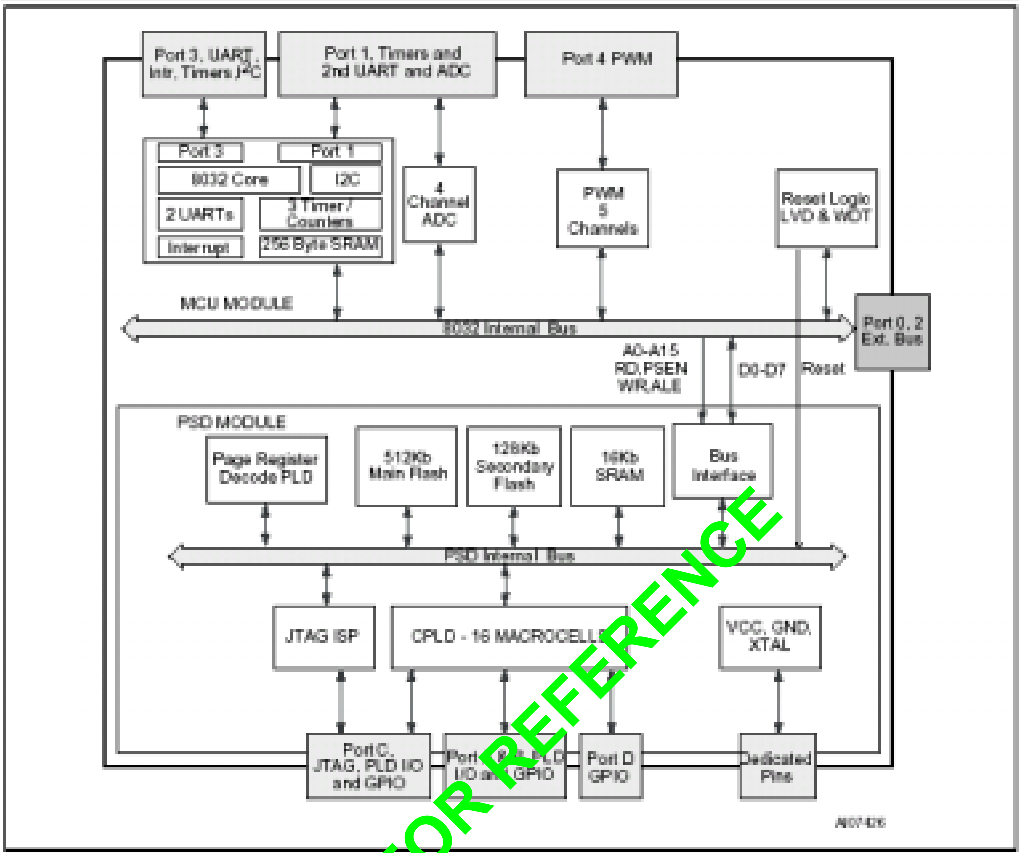
图13. μPSD3200系列器件的状态时序



μPSD32× 硬件描述

模块化结构的μPSD3200系列由两个主要的功能模块组成： MCU模块和PSD模块。MCU模块包含了标准的8032内核，外围电路和其他系统支持功能。PSD模块为8032CPU提供可配置的程序和数据存储器。另外，它有自己的I/O端口设置，为执行常规逻辑而设计的：拥有16个宏单元的PLD。端口A，B，C，D是通用I/O口，它的口结构与MCU模块的端口0~4是不同的。PSD模块与CPU通讯是通过内部地址数据总线（A0~A15，D0~D7）和控制信号（RD_，WR_，PSEN_，ALE，RESET_）。用户通过PSDsoft开发工具定义地址译码PLD，将PSD模块内的存储器块其映射为任意程序或数据地址空间。

图14. μPSD32××系列功能模块



MCU模块描述

这一节详细描述了MCU模块的系统功能和外围电路，包括：

- 特殊功能寄存器
- 定时／计数器
- 中断
- PWM
- 监控功能（LVD和看门狗）
- 异步串口
- 节电模式
- I2C总线
- 振荡器
- 模数转换器
- I/O端口

特殊功能寄存器

表15给出了位于内部存储器的特殊功能寄存器（SFR）的空间结构图。注意，特殊功能寄存器里不是所有的地址都被占用。芯片内没有占用的地址是没有用的。读取这些地址，将返回随机数，写操作也

是无效的。用户软件将这些无效地址写入0。

表15. SFR存储器图

F8									FF
F0	B[1]								F7
E8	UISTA[1]	UIEN	UCON0	UCON1	UCON2	USTA	UADR	UDR0	EF
E0	ACC[1]	USCL					UDT1	UDT0	E7
D8	S1CON[1]	S1STA	S1DAT	S1ADR	S2CON	S2STA	S2DAT	S2ADR	DF
D0	PSW[1]	S1SETUP	S2SETUP		RAMBUF	DDCDAT	DDCADR	DDCCON	D7
C8	T2CON[1]	T2MOD	RC2L	RC2H	TL2	TH2			CF
C0	P4[1]								C7
B8	IP[1]								BF
B0	P3[1]	PSCL0L	PSCL0H	PSCL1L	PSCL1H			IPA	B7
A8	IE[1]		PWM4H	PWM4L			WDKEY		AF
A0	P2[1]	PWMCON	PWM0	PWM1	PWM2	PWM3	WDRST	IEA	A7
98	SCON	SBUF	SCON2	SBUF2					9F
90	P1[1]	P1SFS		P3SFS	P4SFS	ASCL	ADAT	ACON	97
88	TCON[1]	TMOD	TL0	TL1	TH0	TH1			8F
80	P0[1]	SP	DPL	DPH				PCON	87

注意：1. 寄存器可以被位寻址

表16. 所有SFR的列表

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
80	P0									FF	Port 0
81	SP									07	Stack Ptr
82	DPL									00	Data Ptr Low
83	DPH									00	Data Ptr High
87	PCON	SMOD	SMOD1	LVREN		GF1	GF0	PD	IDLE	00	Power Ctrl
88	TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00	Timer / Cntr Control
89	TMOD	Gate	C/T	M1	M0	Gate	C/T	M1	M0	00	Timer / Cntr Mode Control
8A	TL0									00	Timer 0 Low
8B	TL1									00	Timer 1 Low
8C	TH0									00	Timer 0 High
8D	TH1									00	Timer 1 High
90	P1									FF	Port 1
91	P1SFS	P1S7	P1S6	P1S5	P1S4					00	Port 1 Select Register
93	P3SFS	SCL	SDA							00	Port 3 Select Register
94	P4SFS	P4S7	P4S6	P4S5	P4S4	P4S3	P4S2	P4S1	P4S0	00	Port 4 Select Register
95	ASCL									00	8-bit Prescaler for ADC clock
96	ADAT	ADAT7	ADAT6	ADAT5	ADAT4	ADAT3	ADAT2	ADAT1	ADAT0	00	ADC Data Register
97	ACON			ADEN		ADS1	ADS0	ADST	ADSF	00	ADC Control Register
98	SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00	Serial Control Register
99	SBUF									00	Serial Buffer
9A	SCON2	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00	2nd UART Ctrl Register
9B	SBUF2									00	2nd UART Serial Buffer
A0	P2									FF	Port 2
A1	PWMCON	PWML	PWMP	PWME	CFG4	CFG3	CFG2	CFG1	CFG0	00	PWM Control Polarity

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
A2	PWM0									00	PWM0 Output Duty Cycle
A3	PWM1									00	PWM1 Output Duty Cycle
A4	PWM2									00	PWM2 Output Duty Cycle
A5	PWM3									00	PWM3 Output Duty Cycle
A6	WDRST									00	Watch Dog Reset
A7	IEA	EDDC			ES2			EI^2C	EUSB	00	Interrupt Enable (2nd)
A8	IE	EA	-	ET2	ES	ET1	EX1	ET0	EX0	00	Interrupt Enable
A9											
AA	PWM4H									00	PWM4 Output Duty (High)
AB	PWM4L									00	PWM4 Output Duty (Low)
AC											
AD											
AE	WDKEY									00	Watch Dog Key Register
AF											
B0	P3									FF	Port 3
B1	PSCL0L									00	Prescaler 0 Low (8-bit)
B2	PSCL0H									00	Prescaler 0 High (8-bit)
B3	PSCL1L									00	Prescaler 1 Low (16-bit)
B4	PSCL1H									00	Prescaler 1 High (16-bit)
B7	IPA	PDDC			PS2			PI^2C	PUSB	00	Interrupt Priority (2nd)
B8	IP			PT2	PS	PT1	PX1	PT0	PX0	00	Interrupt Priority
C0	P4									FF	New Port 4

SFR Addr	Reg Name	Bit Register Name 7	6	5	4	3	2	1	0	Reset Value	Comments
CC	TL2									00	Timer 2 Low byte
CD	TH2									00	Timer 2 High byte
D0	PSW	CY	AC	FO	RS1	RS0	OV		P	00	Program Status Word
D1											
D2	S2SETUP									00	I^2C (S2) Setup
D4											
D5											
D6											
D7											
D8											
D9											
DA											
DB											
DC	S2CON	CR2	EN1	STA	STO	ADDR	AA	CR1	CR0	00	I^2C Bus Control Reg
DD	S2STA	GC	Stop	Intr	TXMd	Bbusy	Blost	ACK_R	SLV	00	I^2C Bus Status
DE	S2DAT									00	Data Hold Register
DF	S2ADR									00	I^2C address
E0	ACC									00	Accumulator
F0	B									00	B Register

表17. PSD模块寄存器地址偏移

CSIO PAddr Offset	Register Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
00	Data In (Port A)	Reads Port pins as input									
02	Control (Port A)	Configure pin between I/O or Address Out mode. Bit = 0 selects I/O								00	
04	Data Out (Port A)	Latched data for output to Port pins, I/O output mode								00	
06	Direction (Port A)	Configures Port pin as input or output. Bit = 0 selects input								00	
08	Drive (Port A)	Configures Port pin between CMOS, Open Drain or Slew rate. Bit = 0 selects CMOS								00	
0A	Input Macrocell (Port A)	Reads latched value on Input Macrocells									
0C	Enable Out (Port A)	Reads the status of the output enable control to the Port pin driver. Bit = 0 indicates pin is in input mode.									
01	Data In (Port B)										
03	Control (Port B)									00	
05	Data Out (Port B)									00	
07	Direction (Port B)									00	
09	Drive (Port B)									00	
0B	Input Macrocell (Port B)										
0D	Enable Out (Port B)										
10	Data In (Port C)										
12	Data Out (Port C)									00	
14	Direction (Port C)									00	
16	Drive (Port C)									00	
18	Input Macrocell (Port C)										
1A	Enable Out (Port C)										
11	Data In (Port D)	*	*	*	*	*		*			Only Bit 1 and 2 are used
13	Data Out (Port D)	*	*	*	*	*		*		00	Only Bit 1 and 2 are used
15	Direction (Port D)	*	*	*	*	*		*		00	Only Bit 1 and 2 are used
17	Drive (Port D)	*	*	*	*	*		*		00	Only Bit 1 and 2 are used

CSIO PAddr Offset	Register Name	Bit Register Name 7	6	5	4	3	2	1	0	Reset Value	Comments
1B	Enable Out (Port D)	*	*	*	*	*		*			Only Bit 1 and 2 are used
20	Output Macrocells AB										
21	Output Macrocells BC										
22	Mask Macrocells AB										
23	Mask Macrocells BC										
C0	Primary Flash Protection	Sec7_Prot	Sec6_Prot	Sec5_Prot	Sec4_Prot	Sec3_Prot	Sec2_Prot	Sec1_Prot	Sec0_Prot		Bit = 1 sector is protected
C2	Secondary Flash Protection	Security_Bit	*	*	*	Sec3_Prot	Sec2_Prot	Sec1_Prot	Sec0_Prot		Security Bit = 1 device is secured
C7	JTAG Enable	*	*	*	*	*	*	*	JTAG SWEN	00	JTAG_SWEN =1 JTAG is enabled
B0	PMMR0	*	*	PLD Mcells clk	PLD array-clk	PLD Turbo		APD enable	*	00	Control PLD power consumption
B4	PMMR2	*	PLD array WRh	PLD array Ale	PLD array Cntl2	PLD array Cntl1	PLD array Cntl0	*	*	00	Blocking inputs to PLD array
E0	Page									00	Page Register
E2	VM	Periph-mode	*	*	FL_data	Boot_data	FL_code	Boot_code	SR_code		Configure 8032 Program and Data Space

注意：（寄存器地址=csiop地址 + 地址偏移，这儿，csiop地址是用户在PSDsoft里定义的）*所指示的位不能使用，需设为0。

中断系统

有10个中断源的中断请求，如下：

- INT0外部中断
- 第二异步串口中断
- 定时器0中断
- I2C中断
- INT1外部中断
- DDC中断
- 定时器1中断
- 异步串口中断
- 定时器2中断

外部中断 Int0

- INT0可以通过设置寄存器TCON的IT0位，将中断设置为电平触发或边沿触发。中断产生的标志位是TCON的IE0。
- 当外部中断产生时，如果中断是边沿触发，当进入相应的中断服务程序时，相应的中断请求标志被硬件自动清零。
- 如果中断是电平触发，中断请求标志将一直保持，直到被请求的中断被响应。那么，在中断服务程序执行完以前，中断标志是无效的，否则将会产生其他的中断。

定时器0和1中断

- 定时器0和定时器1中断被TF0和TF1产生，TF0和TF1各自的定时／计数寄存器溢出时，TF0和TF1被设置（除了定时器0的模式3）。
- 当中断响应时，这些标志被内部硬件清零。

定时器2中断

- 定时器2中断被TF2产生，TF2被定时器2的溢出置 1 。这个标志不能被硬件清零，必须被软件清零。
- 它也可以通过T2EX信号（定时器2外部中断引脚P1.1）产生，T2EX被T2CON寄存器的EXEN2和EXF2两位控制。定时器2的这个功能定义同90C320的该功能定义一样。

I2C中断

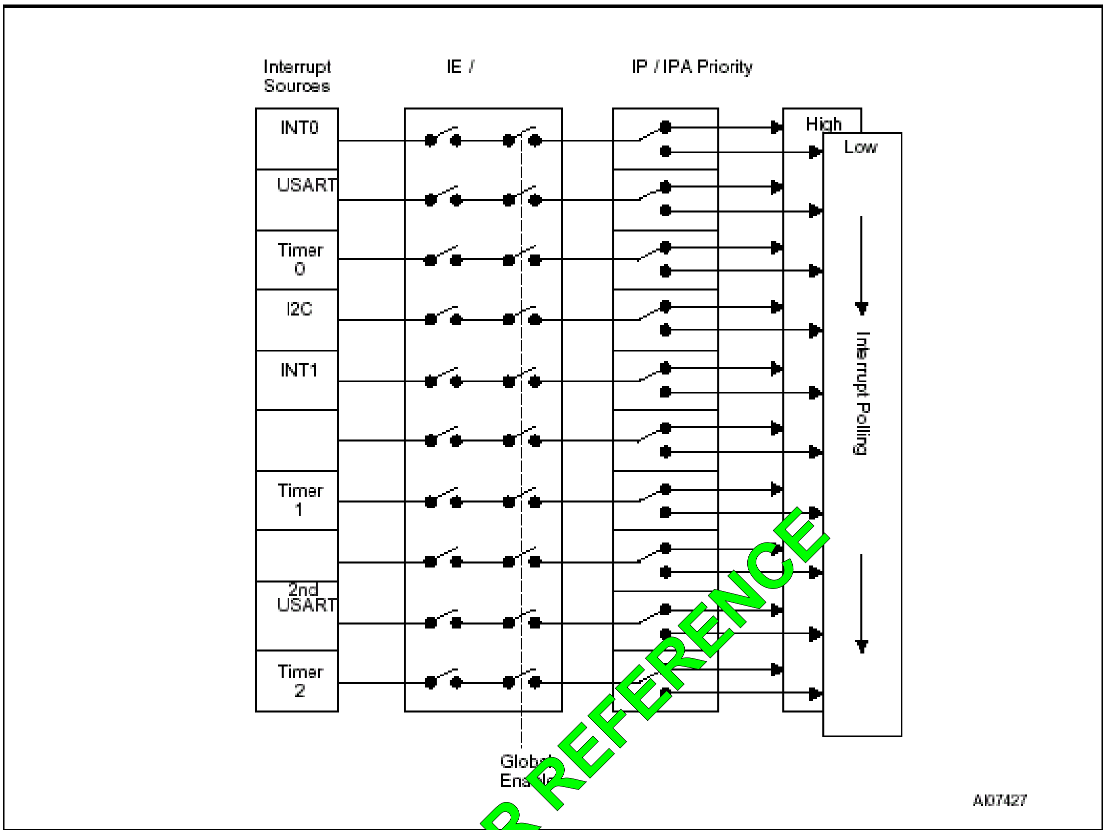
- I2C中断被寄存器S2STA的INTR位产生。
- 该标志被硬件清零。

外部中断Int1

- INT1可以通过设置寄存器TCON的IT1位，设置为电平触发或边沿触发。中断产生的标志位是TCON的IE1。
- 当外部中断产生时，如果中断是边沿触发，当进入相应的中断服务程序时，相应的中断请求标志被硬件自动清零。
- 如果中断是电平触发，中断请求标志将一直保持，直到被请求的中断被响应。那么，在中断服务程序执行完以前，中断标志是无效的，否则其他中断将产生。

- 异步串行中断被RI（接收中断）或TI（发送中断）产生。
- 当USART中断产生时，相应的请求标志必须被软件清零。中断服务程序将不得不检测各种USART寄存器，来确定数据源和清除相应的标志。
- 除了附加中断控制寄存器（A7H,B7H）的位4的附加中断控制外，两个异步串行口其他都相同。

图15. 中断系统



表：特殊功能寄存器

表18. 特殊功能寄存器

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
A7	IEA	EDDC	—	—	ES2	—	—	EI^2C	EUSB	00	Interrupt Enable (2nd)
A8	IE	EA	—	ET2	ES	ET1	EX1	ET0	EX0	00	Interrupt Enable
B7	IPA	PDDC	—	—	PS2	—	—	PI^2C	PUSB	00	Interrupt Priority (2nd)
B8	IP	—	—	PT2	PS	PT1	PX1	PT0	PX0	00	Interrupt Priority

表19. 优先级

Source	Priority with Level
Int0	0 (highest)
2nd USART	1
Timer 0	2
I²C	3
Int1	4
reserved	5
Timer 1	6
reserved	7
1st USART	8
Timer 2+EXF2	9 (lowest)

中断优先级的结构

每个中断源都可由软件设置为两种优先级中的一种。中断优先级被中断优先级特殊功能寄存器IP和IPA定义。

0 = 低优先级

1 = 高优先级

一个优先级低的中断，能被一个优先级高的中断中止。而一个优先级高的中断，不能被其他任何中断源中止。如果不同优先级别的中断同时出现，优先级高的中断被响应。如果同时接收到同一中断优先级的多个请求，有内部对优先级有一个统一的规定。因此，每级优先级，都有第二个优先级检测机构。

中断使能结构

通过设置和清除IE和IEA中断允许寄存器的相应位，每一个中断源都可以被使能或屏蔽。IE寄存器的EA位清零，将屏蔽掉所有的中断源。

表20. IE寄存器各位的定义

Bit	Symbol	Function
7	EA	Disable all interrupts: 0: no interrupt with be acknowledged 1: each interrupt source is individually enabled or disabled by setting or clearing its enable bit
6	—	Reserved
5	ET2	Enable Timer2 interrupt
4	ES	Enable USART interrupt
3	ET1	Enable Timer1 interrupt
2	EX1	Enable external interrupt (Int1)
1	ET0	Enable Timer0 interrupt
0	EX0	Enable external interrupt (Int0)

表21. IEA寄存器各位的定义

Bit	Symbol	Function
7	—	Not used
6	—	Not used
5	—	Not used
4	ES2	Enable 2nd USART Interrupt
3	—	Not used
2	—	Not used
1	EI2C	Enable I²C Interrupt
0	—	Not used

表22. IP寄存器各位的定义

Bit	Symbol	Function
7	—	Reserved
6	—	Reserved
5	PT2	Timer2 interrupt priority level
4	PS	USART interrupt priority level
3	PT1	Timer1 interrupt priority level
2	PX1	External interrupt (Int1) priority level
1	PT0	Timer0 interrupt priority level
0	PX0	External interrupt (Int0) priority level

表23. IPA寄存器各位的定义

Bit	Symbol	Function
7	—	Not used
6	—	Not used
5	—	Not used
4	PS2	2nd USART Interrupt priority level
3	—	Not used
2	—	Not used
1	PI2C	I²C Interrupt priority level
0	—	Not used

如何处理中断

中断标志在每个机器周期的S5P2（状态5的第2拍）被读取。读取的值在下一个机器周期被选中。如果在第一个机器周期的S5P2读取了一个中断标志，第二个机器周期找到这个标志，中断系统将产生一个对应的中断服务子程序调用。但是，如有下面情况，将不会调用中断程序：

- 一个同优先级或优先级更高的中断已经响应。
- 当前机器周期不是正在执行指令的最后一个机器周期，
- 正在执行的指令是RETI，或正在访问中断优先级寄存器和中断允许寄存器。

轮询在各自的机器周期里重复执行，被轮询的值是上一个机器周期的S5P2时读取的值。

注意，如果一个中断标志被激活，但因为上面提到的原因没有响应，即使上述原因消失，中断标志也不会一直保持，被取消的中断不会被响应了。
换句话说，事实上，中断标志被激活但是却没有响应，不会被记得。每次轮询确定是否中断的周期都是新的。

处理器使硬件产生一个恰当的中断服务子程序调用，来响应一个中断请求。硬件产生子程序调用，将程序计数器的值放入堆栈（但不保存PSW），将中断源的中断入口地址放入程序计数器（PC）。

中断入口地址见表24。

程序按中断入口地址开始执行，一直执行到RETI指令。RETI指令告诉处理器，中断子程序已经执行完，弹出堆栈最上面的两个字节，恢复程序计数器。程序又从中止的地方恢复，继续执行。

注意：如果用RET指令，也能返回中止的程序，但它会让中断控制系统认为中断服务程序一直在执行，不再响应任何其他中断。

表24. 中断入口地址

Source	Vector Address
Int0	0003h
2nd USART	004Bh
Timer 0	000Bh
I²C	0043h
Int1	0013h
Timer 1	001Bh
1st USART	0023h
Timer 2+EXF2	002Bh

节电模式

可以执行两种节电模式。

空闲模式

关闭下列功能。

- CPU（暂停工作）

空闲模式下下列功能保持：

- 外部中断
- 定时器0，定时器1，定时器2
- PWM单元
- 串口
- 8位模数转换器
- I2C接口

中断或复位将退出空闲模式。

掉电模式

- 系统时钟暂停工作
- LVD逻辑继续工作
- SRAM内容保持
- 特殊功能寄存器的值一直保持，直到复位

只有复位能退出掉电模式。

表25.

Mode	Addr/Data	Ports1,3,4	PWM	I^2C	DDC	USB
Idle	Maintain Data	Maintain Data	Active	Active	Active	Active
Powerdown	Maintain Data	Maintain Data	Disable	Disable	Disable	Disable

功耗控制寄存器

空闲模式和掉电模式通过软件设置PCON寄存器来实现。

表26. 功率控制寄存器（PCON）

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
87	PCON	SMOD	SMOD1	LVREN	—	GF1	GF0	PD	IDLE	00	Power Ctrl

表27. PCON寄存器各位的定义

Bit	Symbol	Function
7	SMOD	Double baud data rate bit USART
6	SMOD1	Double baud data rate bit 2nd USART
5	LVREN	LVR disable bit (active High)
4	—	Not used
3	—	Not used
2	—	Not used
1	PD	Activate Power-down mode (High enable)
0	IDL	Activate Idle mode (High enable)

空闲模式

在进入空闲模式工作之前，将PCON.0位设置为1的指令，是常规操作模式下执行的最后一条指令。一旦进入空闲模式，CPU的所有状态被保护：堆栈指针、程序计数器、程序状态字、A累加器、RAM和其它所有的寄存器在空闲模式期间，都将保持它们的数据。

有三种方法可中止空闲模式

- 任何中断将使硬件清除PCON.0，以中止空闲模式。中断被响应，接着从中断指令RETI返回，下一条执行的指令是将PCON.0置1指令的下一条指令。
- 外部硬件复位：要求硬件复位信号保持两个机器周期，这样才能完成复位操作。
- 内部复位：在所有内部复位3个机器周期后，微控器重新启动，程序从头开始执行。

掉电模式

在进入掉电模式前置PCON.1为1的指令是最后一条指令。一旦进入掉电模式，晶振停止振荡。芯片内的RAM和特殊功能寄存器被保护。掉电模式将被外部复位中止。

I/O端口（MCU模块）

MCU模块有五个端口：端口0、端口1、端口2、端口3、端口4（参照PSD模块一节I/O端口A、B、C和D）。只有80脚封装的芯片才有的端口0和端口2 是设置的外部数据地址总线。端口0和端口2只能专用于外部数据地址总线扩展，对于芯片本身无法用作其他用途。

除了附加的特殊的外围功能，端口1～端口3与标准8032微控器的端口1～3时相同的。所有的口都是双向口。引脚的附加的特殊的外围功能不可以像普通双向I/O口那样使用。将μPSD3200系列提供的相关的SFR位设置成高电平，那么端口1～4将自动设置成其他附加的特殊功能。

表28.

Port Name	Main Function	Alternate
Port 1	GPIO	Timer 2 - Bits 0,1 2nd UART - Bits 2,3 ADC - Bits 4..7
Port 3	GPIO	UART - Bits 0,1 Interrupt - Bits 2,3 Timers - Bits 4,5 I^2C - Bits 6,7
Port 4	GPIO	DDC - Bits 0..2 PWM - Bits 3..7
USB +/-	USB +/- Only	

下面是SFR寄存器用来控制I/O口改变功能的位地址映射图。端口1使用P1SFS寄存器来改变端口功能，除了定时器2和第二异步通讯口靠它们的配置寄存器允许生效外。复位后P1.0～P1.3缺省为GPIO。

端口3的6脚和7脚已经更改，不同于标准8032的脚。这两只脚过去是读写控制信号，现在是GPIO或I2C总线引脚。读写信号定义有专门的引脚。

端口3和端口4的附加功能被特殊功能选择寄存器P3SFS和P4SFS控制，复位后缺省为GPIO，在相应的PXSFS寄存器位中设置1，将改变其功能。

表29. P1SFS（91H）

7	6	5	4	3	2	1	0
0=Port 1.7 1=ACH3	0=Port 1.6 1=ACH2	0=Port 1.5 1=ACH1	0=Port 1.4 1=ACH0	Bits Reserved		Bits Reserved	

表30. P3SFS（93H）

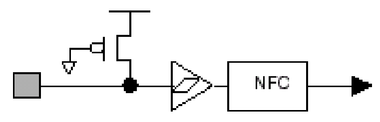
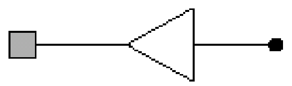
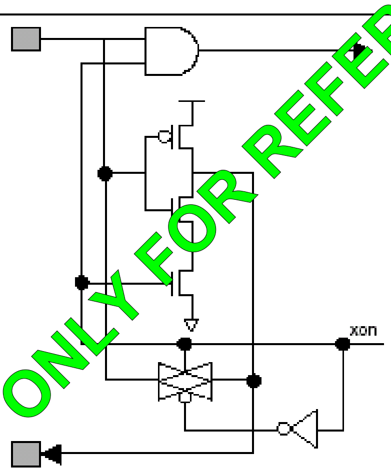
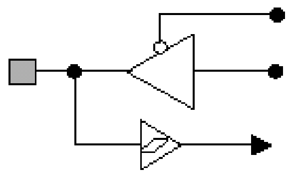
7	6	5	4	3	2	1	0
0 = Port 1.7 1 = SCL from I^2C unit	0 = Port 1.6 1 = SDA from I^2C unit	Bits are reserved.					

表31. P4SFS（94H）

7	6	5	4	3	2	1	0
0=Port 4.7 1=PWM 4 (16-bit)	0=Port 4.6 1=PWM 3	0=Port 4.5 1=PWM 2	0=Port 4.4 1=PWM 1	0=Port 4.3 1=PWM 0	0=Port 4.2 1=Vsync	0=Port 4.1 1=DDC - SCL	0=Port 4.0 1=DDC - SDA

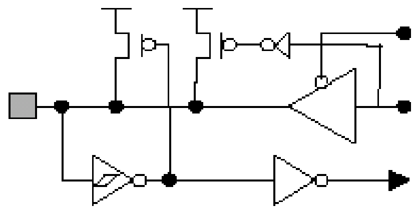
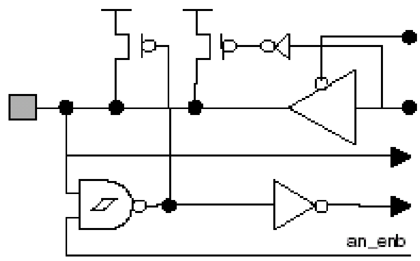
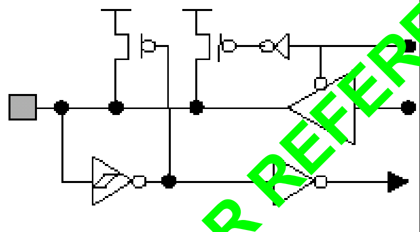
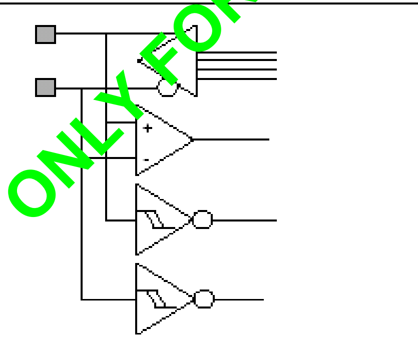
端口类型和描述

图16. 端口类型和定义（第一部分）

Symbol	In / Out	Circuit	Description
RESET	I		• Schmitt input with internal pull-up CMOS compatible interface NFC : 400ns
WR, RD,ALE, PSEN	O		Output only Sink current : 5mA
XTAL1, XTAL2	I O		On-chip oscillator On-chip feedback resistor Stop in the power down mode External clock input available CMOS compatible interface
PORT0	I/O		Bidirectional I/O port Schmitt input Open-drain output(5V) Address Output (Push-Pull) Sink current : 5mA CMOS compatible interface Source current: 5mA

AI06653

图17. 端口类型和定义（第二部分）

Symbol	In/ Out	Circuit	Function
PORT1 <3:0>, PORT3, PORT4<7:3,1:0> PORT2	I/O		Bidirectional I/O port with internal pull-ups Schmitt input Sink current : 5mA CMOS compatible interface Source current =5mA when push-pull output mode.
PORT1 < 7:4 >	I/O		Bidirectional I/O port with internal pull-ups Schmitt input Sink current : 5mA CMOS compatible interface Analog input option Source current =5mA
PORT4.2	I/O		Bidirectional I/O port with internal pull-ups Schmitt input. Sink current : 5mA TTL compatible interface Pull-up when reset Address Latch Enable Program Strobe Enable Source current =5mA
USB -, USB +	I/O		Bidirectional I/O port Schmitt input TTL compatible interface

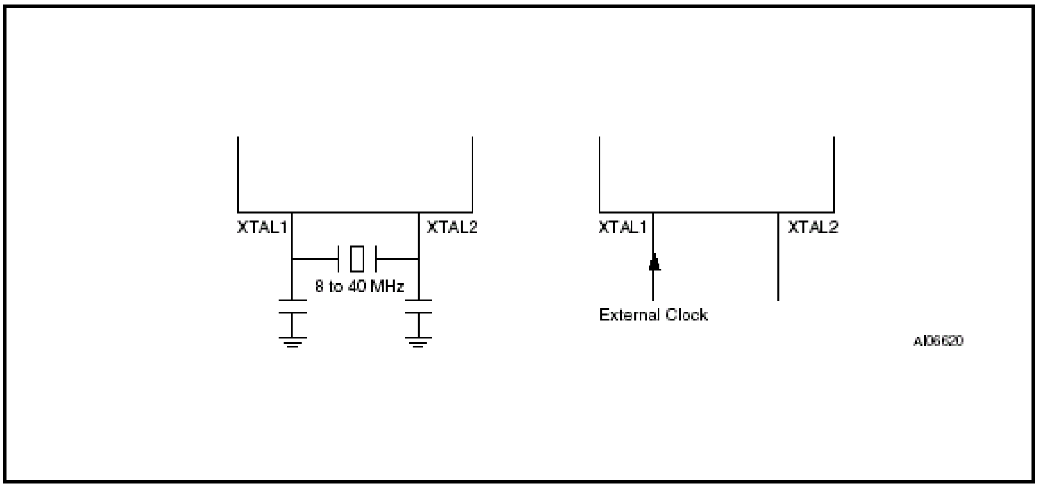
AI06654

振荡器

μPSD3200系列的振荡电路是一个单稳反向放大的自激振荡器。XTAL1和XTAL2之间的电路，基本上是反向器，晶体和陶瓷振荡器都可以用来作为反馈元件，组成振荡电路。两种都产生相同的谐振。

XTAL1是高增益放大器的输入端，XTAL2是输出端。μPSD3200系列的外部振荡驱动，是XTAL1由一个外部时钟源驱动，XTAL2开路。

图18. 振荡器



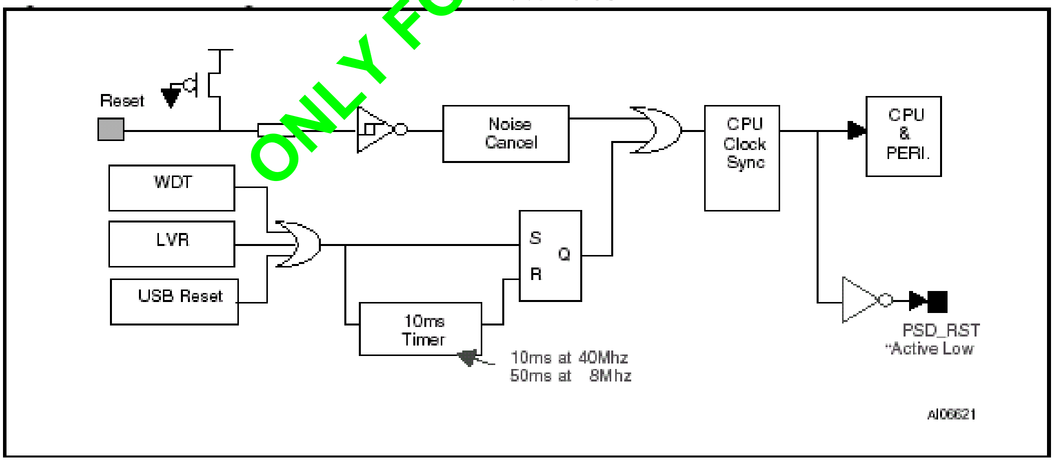
监控

μPSD3200系列有四种方法产生复位。

- 通过外部RESET脚
- 通过内部LVR部件
- 通过看门狗定时器

复位机制见图19。

图19. 复位结构



每个复位源都会激活一个内部复位信号。CPU响应一个内部复位，并将预定的状态放入内部寄存器。内部复位也发送一个有效的低电平复位信号给PSD模块。

外部复位

复位引脚RESET连接到一个为了减少噪声的斯密特触发器。上电，振荡器工作后，使RESET脚为低电平，并保持1ms，这样一个复位就完成了。参照AC说明中的其他复位时间要求。

低电压复位

当VDD低于复位极限时，LVR电路产生一个内部复位。VDD返回复位极限后，复位信号保持10ms。最初上电，LVR工作（缺省）。以后上电，LVR可以通过PCON寄存器的LVREN位使它停止工作。

注意：在空闲模式和掉电模式，LVR逻辑一直有效。

复位极限：

- 5V操作：4V±0.25V
- 3.3V操作：2.5V±0.2V

这个逻辑支持大约0.1V的滞后和1μs的噪声抑制延迟。

看门狗定时器溢出

当看门狗的22位计数器溢出时，将产生一个内部复位。细节请看“看门狗定时器”一节。

看门狗定时器

当硬件看门狗定时器（WDT）溢出，将复位μPSD3200。看门狗是专门设计的，当CPU陷入软件混乱时的一种恢复方式。防止系统复位的定时器必须被应用软件及时重载。如果处理器遇到硬件或软件故障，定时器的重载将会失败。这个失败将会导致定时器溢出，产生复位信号，从而防止处理器失去控制。

在空闲模式，看门狗定时器和复位电路都处于工作状态。看门狗包含22位计数器、看门狗定时器复位特殊寄存器（WDRST）和看门狗键寄存器（WDKEY）。

处理器一工作，WDT就自动激活，用户只需管理它。

当定时器计数到4194304（3FFFFFH）时，就溢出。WDT每个机器周期都加1。

这就意味着：用户必须至少每4194304个机器周期（40MHZ时为1.258秒）复位一次WDT。要复位WDT，用户必须写一个00~7EH之间的值到WDRST寄存器。写到WDRST寄存器的值放入22位计数器的高7位。这就允许用户预置一个初始值到计数器产生一个灵活的看门狗时间输出周期。WDRST写入“00”清计数器。

看门狗定时器被看门狗键寄存器WDKEY控制。只有当置入01010101（=55H）时，看门狗定时器被禁止。置入其它数值，看门狗定时器开启。当需要看门狗定时器功能时，安全键将防止看门狗定时器异常中止。

在空闲模式，振荡器继续工作，防止空闲模式下WDT复位处理器，用户必须定期退出空闲模式，设置定时器，处理WDT，再进入空闲模式。

表32. 看门狗定时器键寄存器（WDKEY: 0AEH）

7	6	5	4	3	2	1	0
WDKEY7	WDKEY6	WDKEY5	WDKEY4	WDKEY3	WDKEY2	WDKEY1	WDKEY0

表33. WDKEY寄存器各位定义

Bit	Symbol	Function
7 to 0	WDKEY7 to WDKEY0	Enable or disable watchdog timer. 01010101 (=55h): disable watchdog timer. Others: enable watchdog timer

表34. 看门狗定时器清除寄存器（WDRST: 0AEH）

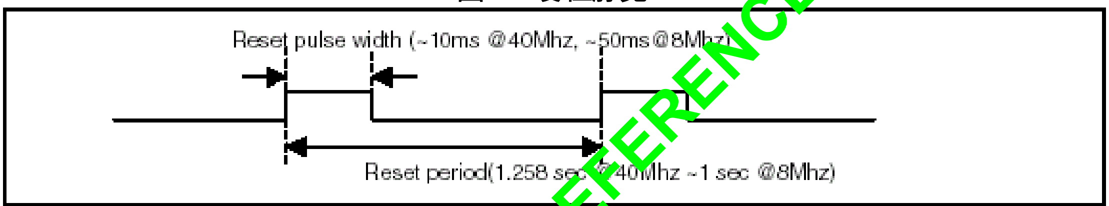
7	6	5	4	3	2	1	0
Reserved	WDRST6	WDRST5	WDRST4	WDRST3	WDRST2	WDRST1	WDRST0

表35. WDRST寄存器各位定义

Bit	Symbol	Function
7	—	Reserved
6 to 0	WDRST6 to WDKEY0	To reset watchdog timer, write any value beteen 00h and 7Eh to this register. This value is loaded to the 7 most significant bits of the 22-bit counter. For example: MOV WDRST,#1EH

看门狗复位脉冲的脉冲宽度取决于时钟频率。复位周期是：Tfosc × 12 ×222
复位脉宽是：Tfosc × 12 × 215

图20. 复位脉宽



定时器/计数器（定时器0、定时器1和定时器2）

μPSD3200系列有三个16位定时／计数寄存器：Timer0、Timer1和Timer2。它们都可以配置成定时器或计数器使用，功能同标准8032结构一致。在“定时器”功能里，寄存器一个机器周期加1。因为一个机器周期包括6个时钟周期，所以计数频率为1/6CPU时钟频率。在“计数”功能里，相应的外部输入引脚由1变0时，计数器加1。在这种功能下，外部输入在每个机器周期的S5P2被采样。当第一个周期采样值为高电平，第二个周期采样值为低电平，计数器就加1。在检测到电平变化的下一个机器周期的S2P1时刻，新的计数值出现在寄存器里。因为它需要2个机器周期（即12个CPU时钟周期）去识别1到0的变化，所以最大的计数率为1/12的CPU时钟频率。对于外部输入信号没有任务周期的限制，但是必须保证在它改变之前至少要被采样一次。它必须保持至少一个完整的周期。另外，“定时”或“计数”选择，Timer0和Timer1由四种操作模式可选。

定时器0和定时器1

“定时”或“计数”功能的选择靠特殊功能寄存器TMOD的C/T位来控制。这些定时/计数器有四种操作模式，它们通过TMOD的两位（M1和M0）来选择。模式0、1、2对于两个定时／计数器是相同的，只有模式3不同。四种操作在下文被描述。

表36. 控制寄存器（TCON）

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

表37. TCON寄存器各位的定义

Bit	Symbol	Function
7	TF1	Timer 1 overflow flag. Set by hardare on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine
6	TR1	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter on or off
5	TF0	Timer 0 overflow flag. Set by hardare on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine
4	TR0	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter on or off
3	IE1	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed
2	IT1	Interrupt 1 Type control bit. Set/cleared by software to specify falling-edge/low-level triggered external interrupt
1	IE0	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed
0	IT0	Interrupt 0 Type control bit. Set/cleared by software to specify falling-edge/low-level triggered external interrupt

模式0.

将任意一个定时器置成模式0，使它像一个8048定时器，是一个带有可被32除的预标定器的8位计数器。图21给出了定时器1在模式0的操作。在这个模式，定时器寄存器作为一个13位寄存器来配置。当计数从全1变成全0时，将设置定时器中断标志TF1。当TR1=1、GATE=0或TR=1、/INT1=1时，被计数的输入对定时器才有效。（设置GATE=1允许定时器被外部输入／INT1控制，很容易测量脉冲宽度。）TR1是特殊功能寄存器TCON中的一个控制位，GATE位在TMOD寄存器里。

这个13位的寄存器包含了8位的TH1和低5位的TL1。TL1的高3位是不确定的，被忽略。设置运行标志不会清除该寄存器。

定时器0的模式0操作同定时器1相同。图21中只需将TR1、TF1、/INT1替换成TR0、TF0、/INT0。有两个不同的GATE位，一个给定时期1，一个给定时期0。

模式1.

模式1和模式0，除了定时寄存器是16位以外，其他都相同。

表38. TMOD寄存器（TMOD）

7	6	5	4	3	2	1	0
Gate	C/$\overline{T}$	M1	M0	Gate	C/$\overline{T}$	M1	M0

表39. TMOD寄存器各位定义

Bit	Symbol	Timer	Function
7	Gate	Timer1	Gating control when set. Timer/Counter 1 is enabled only while INT1 pin is High and TR1 control pin is set. When cleared, Timer 1 is enabled whenever TR1 control bit is set
6	C/T		Timer or Counter selector, cleared for timer operation (input from internal system clock); set for counter operation (input from T1 input pin)
5	M1		(M1,M0)=(0,0): 13-bit Timer/Counter, TH1, with TL1 as 5-bit prescaler (M1,M0)=(0,1): 16-bit Timer/Counter. TH1 and TL1 are cascaded. There is no prescaler. (M1,M0)=(1,0): 8-bit auto-reload Timer/Counter. TH1 holds a value which is to be reloaded into TL1 each time it overflows (M1,M0)=(1,1): Timer/Counter 1 stopped
4	M0		
3	Gate	Timer0	Gating control when set. Timer/Counter 0 is enabled only while INT0 pin is High and TR0 control pin is set. When cleared, Timer 0 is enabled whenever TR0 control bit is set
2	C/T		Timer or Counter selector, cleared for timer operation (input from internal system clock); set for counter operation (input from T0 input pin)
1	M1		(M1,M0)=(0,0): 13-bit Timer/Counter, TH0, with TL0 as 5-bit prescaler (M1,M0)=(0,1): 16-bit Timer/Counter. TH0 and TL0 are cascaded. There is no prescaler. (M1,M0)=(1,0): 8-bit auto-reload Timer/Counter. TH0 holds a value which is to be reloaded into TL0 each time it overflows (M1,M0)=(1,1): TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits
0	M0		

图21. 定时／计数器模式0：13位计数器

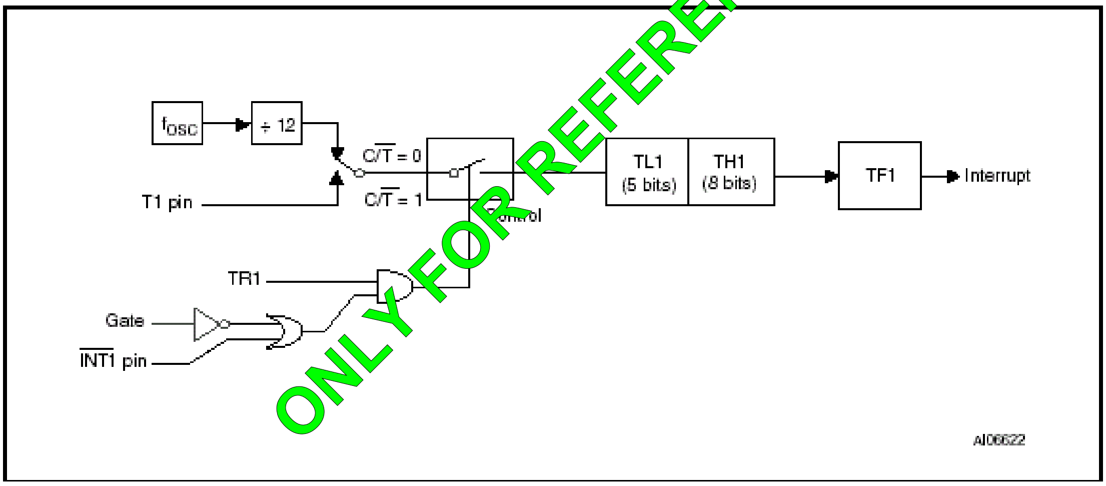


图22. 定时／计数器模式2：8位自动重载

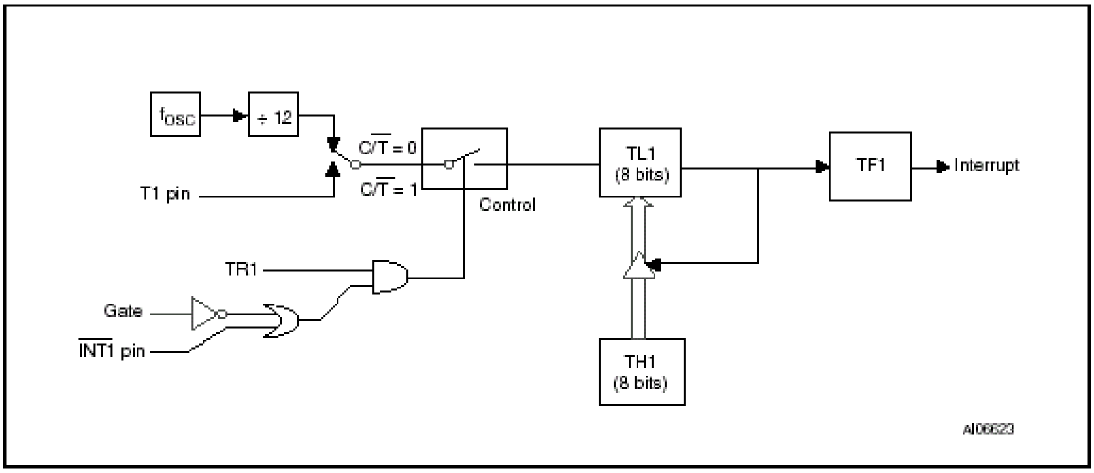
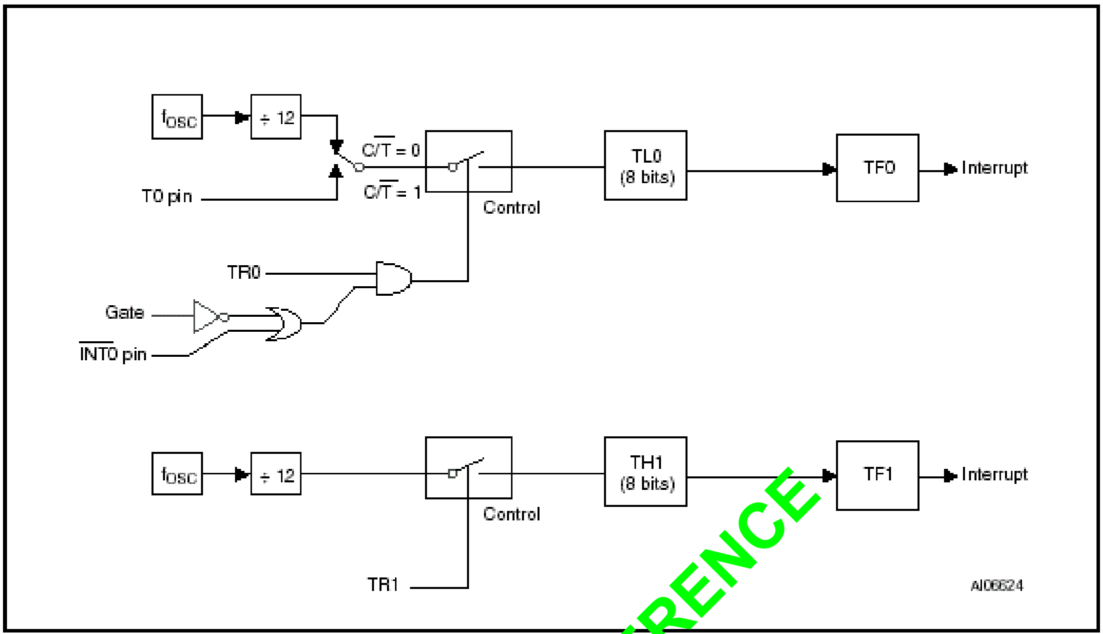


图23. 定时／计数器模式3：双8位计数器



模式2.

模式2将定时器寄存器配置成一个自动重载的8位计数器（TL1），见图22。TL1溢出时不仅设置TF1，同时将软件预先置好的TH1的内容重新载入TL1。重载不会改变THL的值。模式2操作对定时器0也是相同的。

模式3.

定时器1的模式3仅仅是保持它的计数值，停止计数。相当于TR1=0。定时器0的模式3使TL0和TH0作为两个分开的计数器。图23是定时器0的模式3的逻辑图。TL0使用定时器 0 的控制位：C/T、GATE、TR0、INT0和TF0。TH0被锁定在定时器功能（计算机器周期）和使用来自定时器1的状态控制位TR1和TF1。因此，TH0现在控制“定时器1”的中断。模式3提供了：要求在计数器上增加一个附加的8位定时器的应用。定时器0在模式3的情况下，μPSD3200系列看起来就好像有3个定时／计数器。当定时器0在模式3的时候，定时器1依靠退出和进入模式3来开启和关闭定时器1，或被串口当波特率发生器来使用，或任何不要求中断的应用。

定时器2

和定时器0、定时器1一样，定时器也可以作为定时器或计数器使用。这个选择通过特殊功能寄存器T2CON的C/T2位来设置。它有3种操作模式：捕获、自动载入和波特率发生器，这些模式通过T2CON的某些位来选择，见表41。在捕获模式，有两个选项，用T2CON的EXEN2位来选择，EXEN2=0时，定时器2是一个16位的定时或计数器，当它溢出时将TF2置1，从而产生一个中断。如果EXEN2=1，定时器2不仅有上述功能，而且当外部输入T2EX脚由1到0变化时，将定时器2寄存器TL2和TH2当前的值取出，分别送入寄存器RC2L和RC2H。另外，T2EX脚上的这个跳变，会把寄存器T2CON的EXF2位置1，而EXF2象TF2一样，可以产生一个中断。图24画出了捕获模式。

在自动重载模式，又有两个选项，被T2CON的EXEN2位设置。如果EXEN2=0，当定时器2溢出时，不仅将TF2置1，同时，将软件预置到寄存器RC2L和RC2H的16位值，重载入定时器2寄存器。如果EXEN2=1，定时器2在保持上述功能的同时，当外部输入T2EX脚由1到0变化时，也将引发这个16位的重载操作，同时将EXF2置1。自动重载模式在图25的标准串口图中说明了。当(RCLK，RCLK1)=1同时／或者（TCLK，TCLK1）=1时，选中波特率发生器模式。它将和串口一起详细说明。

表40. 定时／计数器2控制寄存器（T2CON）

7	6	5	4	3	2	1	0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T}$2	CP/$\overline{RL}$2

表41. T2CON寄存器的各位的定义

Bit	Symbol	Function
7	TF2	Timer 2 overflow flag. Set by a Timer 2 overflow, and must be cleared by software. TF2 will not be set when either RCLK or TCLK = 1
6	EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2=1. When Timer 2 interrupt is enabled, EXF2=1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software
5	RCLK	Receive clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in modes 1 and 3. RCLK=0 causes Timer 1 overflow to be used for the receive clock
4	TCLK	Transmit clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3. TCLK=0 causes Timer 1 overflow to be used for the transmit clock
3	EXEN2	Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2=0 causes Time 2 to ignore events at T2EX
2	TR2	Start/stop control for Timer 2. A logic 1 starts the timer
1	C/$\overline{T}$2	Timer or counter select for Timer 2. Cleared for timer operation (input from internal system clock, t_{CPU}); set for external event counter operation (negative edge triggered)
0	CP/$\overline{RL}$2	Capture/reload flag. When set, capture will occur on negative transition of T2EX if EXEN2=1. When cleared, auto-reload will occur either with Timer 2 overflows, or negative transitions of T2EX when EXEN2=1. When either RCLK or TCLK = 1, this bit is ignored, and timer is forced to auto-reload on Timer 2 overflow

表42. 定时 / 计数器2操作模式

Mode	T2CON			T2MOD	T2CON	P1.1	Remarks	Input Clock	
	RxCLK or TxCLK	CP/$\overline{RL2}$	TR2	DECN	EXEN	T2EX		Internal	External (P1.0/T2)
16-bit Auto-reload	0	0	1	0	0	x	reload upon overflow	fosc/12	MAX fosc/24
	0	0	1	0	1	↓	reload trigger (falling edge)		
	0	0	1	1	x	0	Down counting		
	0	0	1	1	x	1	Up counting		
16-bit Capture	0	1	1	x	0	x	16-bit Timer/Counter (only up counting)	fosc/12	MAX fosc/24
	0	1	1	x	1	↓	Capture (TH1,TL2) → (RC2H,RC2L)		
Baud Rate Generator	1	x	1	x	0	x	No overflow interrupt request (TF2)	fosc/12	MAX fosc/24
	1	x	1	x	1	↓	Extra external interrupt (Timer 2)		
Off	x	x	0	x	x	x	Timer 2 stops	—	—

注意：↓ = 下降沿

图24. 定时器2的捕获模式

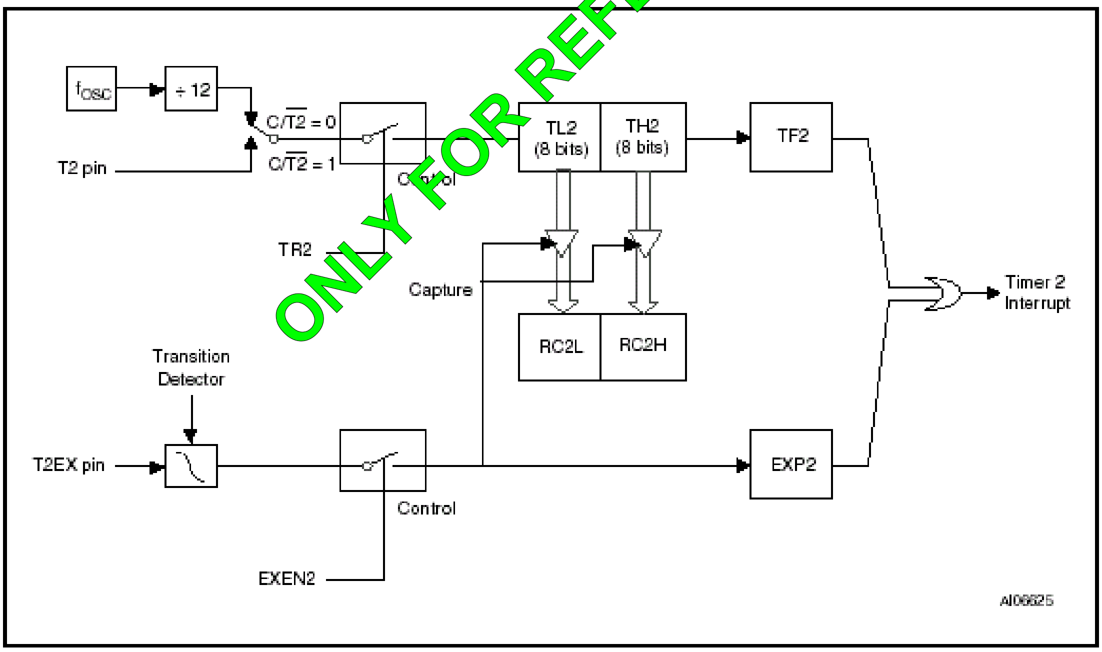
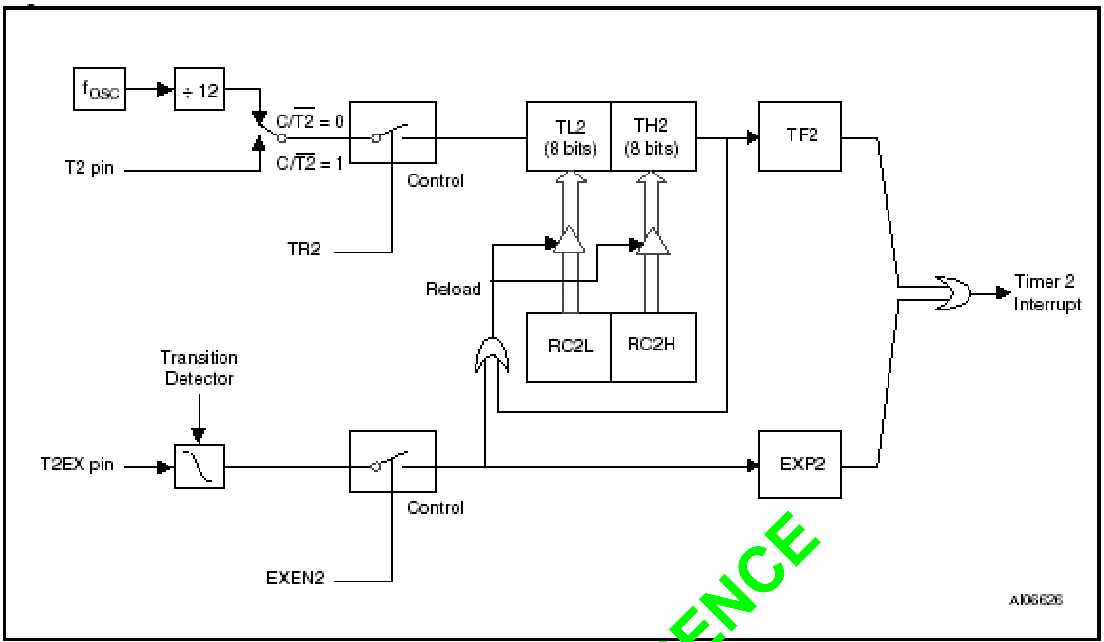


图25. 定时器2的自动重载模式



标准串口（USART）

μPSD3200系列提供了两个标准8032异步通讯串口。第一个口连接到引脚P3.0（RX）和P3.1（TX）。第二个口连接到引脚P1.2（RX）和P1.3（TX）。两个串口的操作相同，都是被寄存器SCON和SCON2控制。串口是全双工的，就是说，它可以同时发送和接收。它也是缓冲器接收，就是说，前一个接收的字节从寄存器读取前，就可以开始着手接受第二个字节。（然而，如果第二个字节接收完了，第一个字节还没有读取，那么其中一个字节就会丢失）串口接收和发送寄存器都是通过特殊功能寄存器SBUF来访问（串口2通过SBUF2来访问）。写SBUF就是载入发送寄存器，读SBUF就是访问物理上分开的接收寄存器。

串口可操作在4种模式下：

模式0.

串行数据进出通过RXD，TXD输出移位时钟信号。发送接收数据为8位（低字节在前）。波特率固定为1/6CPU时钟频率。

模式1.

10位数据被发送（通过TXD）或接收（通过RXD）：1位起始位（0），8位数据位（低字节在前）和1位停止位（1）。接收时，停止位进入特殊功能寄存器SCON的RB8位。波特率可变。

模式2.

11位数据被发送（通过TXD）或接收（通过RXD）：1位起始位（0），8位数据位（低字节在前），1位可编程的第9位数据位和1位停止位（1）。发送时，第9位数据位（SCON的TB8 位）可以被设置为0或1，例如，奇偶位（PSW的P位）移入TB8。接收时，第9位数据位进入特殊功能寄存器SCON的RB8位。停止位被忽略。波特率可编程为1/16或1/32的CPU时钟频率。

模式3.

11位数据被发送（通过TXD）或接收（通过RXD）：1位起始位（0），8位数据位（低字节在前），1位可编程的第9位数据位和1位停止位（1）。模式3和模式2除波特率外其他都是相同的。波特率是可变的。

四种模式中，任何使用SBUF做目的寄存器的指令，都将使发送初始化。模式 0 时，RI=0和REN=1将使接收初始化。其他模式时，如果REN=1，接收将被进来的起始位所初始化。

多机通讯系统

模式2和3为多机通讯预备了一个特殊功能。在这些模式里，9位数据位被接收。第9位数进入RB8。接着是一个停止位。当接收到停止位时，端口可以这样编程，仅仅当RB8=1，串口中断被激活。这个功能可以通过设置SCON的SM2位来启动。如下在多处理器系统中使用这个功能：

当主机想发送一个数据块到几个从机中的一个时，它首先发送一个地址，确定目标从机。地址字节和数据字节的第9位不同，地址字节的第9位为1，数据字节的第9位为0。对SM2=1，没有从机被数据字节中断。然而，地址字节会使所有的从机发生中断，所以每个从机可以检查接收到的字节，看它是否是被指定的从机。被指定地址的从机清除它的SM2位，准备接收即将到来的数据字节。没有被指定的从机保持它们的SM2位为1，继续忽略到来的数据字节。

SM2在模式0时无效，在模式1时用于检测停止位的有效性。模式1接收时，如果SM2=1，除非接到有效的停止位，否则接收中断不会激活。

串口控制寄存器

串口控制和状态寄存器是特殊功能寄存器SCON（第二串口为SCON2），如图26。这个寄存器不仅包含模式选择位、发送和接收的第9位数据（TB8和RB8）和串口中断标志位（TI和RI）。

表43. 串口控制寄存器（SCON）

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	T1	R1

表44. SCON寄存器各位的定义

Bit	Symbol	Function
7	SM0	(SM1,SM0)=(0,0): Shift Register. Baud rate = $f_{OSC}/12$ (SM1,SM0)=(1,0): 8-bit UART. Baud rate = variable (SM1,SM0)=(0,1): 8-bit UART. Baud rate = $f_{OSC}/64$ or $f_{OSC}/32$ (SM1,SM0)=(1,1): 8-bit UART. Baud rate = variable
6	SM1	
5	SM2	Enables the multiprocessor communication features in mode 2 and 3. In mode 2 or 3, if SM2 is set ot 1, R1 will not be activated if its received 8th data bit (RB8) is 0. In mode 1, if SM2=1, R1 will not be activated if a valid stop bit was not recieved. In mode 0, SM2 should be 0
4	REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception
3	TB8	The 8th data bit that will be transmitted in modes 2 and 3. Set or clear by software as desired
2	RB8	In modes 2 and 3, this bit contains the 8th data bit that was received. In mode 1, if SM2=0, RB8 is the snap bit that was received. In mode 0, RB8 is not used
1	T1	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software
0	R1	Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit in the other modes, in any serial reception (except for SM2). Must be cleared by software

波特率.

模式0时波特率是固定的：模式 0 波特率＝fosc/12

模式2的波特率依靠SMOD位的值，SMOD=0（复位时的值）波特率是1/64CPU时钟频率。如果SMOD=1，波特率是1/32CPU时钟频率。

模式2波特率=$(2^{SMOD}/64)\times fosc$

模式1和3，波特率由定时器1的溢出率决定。

用定时器 1 产生波特率.

当使用定时器1作波特率发生器时，模式1和3下，波特率被定时器1的溢出率和SMOD的值所决定，如下：

模式1、3波特率＝$(2^{SMOD}/32)\times$(Timer1溢出率)

这种情况下，定时器 1 中断被禁止。

定时器自身在3种运行模式中的任意模式下，可以按“定时器”或“计数器”操作来配置。最典型的应用是，在自动重载模式下（TMOD寄存器的高四位=0010B），按“定时器”操作来配置。在那种情况下，波特率公式为：

模式1、3波特率=（$2^{SMOD}/32$）×（fosc/12×[256 –(TH1)]）

通过保留定时器1中断，将定时器配置成16位定时器（TMOD的高四位=0001B），使用定时器1中断来产生16位软件重载，便可产生非常低的波特率。图21列出了多种通常使用的波特率和如何从定时器1获得这些波特率。

用定时器/计数器2产生波特率.

μPSD3200系列，T2CON的TCLK和/或RCLK位置1，可将定时器2选择为波特率发生器（看图21定时/计数器2控制寄存器T2CON）。注意，发送和接收的波特率在同一时间可以不同。

T2CON寄存器的TCLK和RCLK位用来配置UART1，PCON寄存器中的TCLK1和RCLK1位用来配置UART2。

波特率发生器模式类似于自动重载模式，TH2滚动溢出将使软件预先设置的RC2H和RC2L寄存器里的16位值，重载到定时器2的寄存器。

模式1和3的波特率决定于定时器2的溢出率，公式如下：

模式1、3波特率=定时器2溢出率/16.

定时器可以配置成“定时器”或“计数器”操作。最典型的应用是，配置为“定时器”操作（C/T2=0）。当用作波特率发生器时，定时器2的“定时器”操作有一点不同。通常，作为定时器时，每个机器周期（1/6CPU时钟频率）定时器寄存器都要加1。而作为波特率发生器时，波特率由下面的公式给出：

模式1、3波特率=fosc/(32×[65536-(RC2H,RC2L)])

这儿，（RC2H,RC2L）是16位无符号数RC2H和RC2L的内容。 定时器2被用在波特率发生模式。仅仅当T2CON寄存器或PCON寄存器的RCLK+TCLK=1时，波特率发生器模式才有效。注意，TH2的滚动溢出不会将TF2置1，不会产生中断。因此，定时器2在波特率发生器模式下，定时器中断不需要被禁止。注意另外一点，如果EXEN2为1，T2EX上的一个1到0的变化将会使EXF2置1，但是不会产生（RC2H,RC2L）对（TH2,TL2）的重载。因此，当定时器2用作波特率发生器时，T2EX可以被用作附加的外部中断。

当定时器2运行在波特率发生器模式下的“定时器”功能时，不要去尝试读写TH2或TL2。在这种情况下，每个状态时间，定时器都要被加1，读写的结果不会正确。可以读RC寄存器，但不要写它，因为，写操作可能和重载从迭，产生写与重载的“与/或”错误。在这种情况下，在访问定时器2或RC寄存器之前，关闭定时器（清TR2）。

模式0.

串行数据输入和输出是通过RXD，TXD输出移位时钟。接收/发送8位数据：8数据位（低字节在前）。波特率固定为1/6的CPU时钟频率。图26给出了串口在模式0时的简单功能图解，和相关的时序。执行任何用SBUF作目的寄存器的指令，都将使发送初始化。S6P2时“写SBUF”信号会将1写入发送移位寄存器的第9位，并告诉TX控制部件，开始发送。内部时序是这样：在“写SBUF”和发送激活之间， 将用掉一个完整的机器周期。

发送使移位寄存器数据输出到RXD脚，移位时钟从TXD脚输出。发送时，移位时钟在每个机器周期的S3、S4和S5为低电平，在每个机器周期的S6、S1和S2为高电平，发送移位的内容向右移一位。数据位从右边移出，0从左边移入。当数据字节的最高位移到了输出位置，最初装入第9位的1刚好在最高位的左边，左边的其他所有位现在都是0。这种条件标志着TX控制部件作最后一次移位、停止发送、将T1置1。这些动作都发生在S1P1。这些动作都发生在“写SBUF”后的第10个机器周期的S1P1。

当REN=1且RI=0时接收被初始化。在下一个机器周期的S6P2，RX控制单元将11111110写入接收移位寄存器，并在下一个时钟拍激活接收。

接收使TXD脚变成输出功能线，输出移位时钟。接收开始后，移位时钟在每个机器周期的S3P1和S6P1变化，接收移位寄存器的内容被左移1位。从右边移入的值是在同一个机器周期的S5P2从RXD脚采样的值。

当数据位从右边移入时，1从左边移出。当最初装入最右边的0到达移位寄存器的最左边时，就标志着RX控制部件做最后一次移位，并装载SBUF。在写SCON清RI位后的第10个机器周期的S1P1，RI被置1停止接收。

模式1.

十位数据被发送（通过TXD）或接收（通过RXD）：一位起始位（0）、8位数据位（低字节在前）、一位停止位（1）。接收时，停止位进入SCON的RB8位。波特率由定时器1溢出率决定。

图27给出了串口在模式1时发送接收的相关时序图。

执行任何以SBUF作目的寄存器的指令，都会将发送初始化。“写SBUF”信号也将1载入发送移位寄存器的第9位，通知TX控制单元，要求发送。其实在可16分频的计数器滚动溢出后的下一个机器周期的S1P1才开始发送。（因此，位的发送与16分频的计数器同步，而不是和“写SBUF”信号同步。）

把起始位送到TXD上激活发送，这样发送就开始了。一位起始位发送后，开始发送数据，使移位寄存器的输出位送到TXD。之后每发送1位的时间出现一个移位脉冲。

因为数据位向右移出，所以0从左边移入。当数据字节的最高位处在移位寄存器的输出位置上，且最初载入第9位的1刚好在最高位的左边，左边的所有位都是0。这个情况标志着TX控制单元作最后一次移位，同时停止发送，置TI为1。这出现在“写SBUF”后16分频定时器的第10次滚动溢出。

发现RXD上1到0的变化将使接收初始化。无论波特率怎样设置，RXD都以16倍的速率采样。当一个发送被检测到，16分频的计数器被立即复位，1FFH被写入输入移位寄存器。复位16分频的计数器将以输入位时间（bit times）为界从新调整计数器的滚动溢出。

计数器的16个状态把1位时间等分成16份。在每个位时间的第7、第8、第9个计数器状态，位监测器采样RXD上的值。每一位的数值采样三次，至少两次相同的值才被确认。这是为了过滤噪声。如果起始位接收到的值不是0，接收电路将被复位，控制单元将返回检测另一个1到0的变化。这就排除了错误的起始位。如果提供的起始位有效，它将被移入输入移位寄存器，剩下的数据接收将会重新开始继续进行。

当数据位从右边进来时，1从左边移出。当起始位到达移位寄存器的最左边的位置（模式1时，是9位寄存器），通知RX控制单元，作最后一次移位，装载SBUF和RB8，同时置RI为1。

如果在最后一个移位脉冲产生时，满足下列条件，数据载入SBUF和RB8，设置RI的信号被产生：

1．RI=0

2．要么SM2=0或接收的停止位=1

如果这些条件没有一个满足，接收状态就被丢失。如果两个条件都满足，停止位进入RB8，8位数据进入SBUF，RI被置位。此时，无论是否满足上面的条件，控制单元都会返回，检测RXD输入线上1到0的变化。

模式2和3.

11位数据被发送（通过TXD）被接收（通过RXD）：一位起始位（0）、8位数据位（低字节在前）、一位可编程的第9位数据位、一位停止位。发送时，第9位数据位（TB8）可以赋值0或1。接收时，第9位进入SCON的RB8。模式2时，波特率可设置为1/16和1/32的CPU时钟频率。模式3可以通过定时器1产生一个可变的波特率。

图28和29给出了模式2和3时，串口的功能图。接收部分和模式1完全相同。发送部分与模式1，仅仅在移位寄存器的第9位有所不同。执行任何以SBUF为目的寄存器的指令，都将使发送初始化。“写SBUF”信号也将TB8载入发送移位寄存器的第9位，同时通知TXD控制单元，请求发送。紧跟着16分频的定时器的下一个滚动溢出的机器周期的S1P1时刻，发送开始。（因此，位时间与16分频的定时器同步，而不与“写SBUF”信号同步）随着发送的激活，发送过程开始，它将起始位送到TXD上。一位起始位发送后，开始发送数据，使移位寄存器的输出位送到TXD。之后每发送1位的时间出现一个移位脉冲。第一个移位时钟移入一个1（停止位）被移入移位寄存器的第9位。这以后，只有0被移入。因此，当数据位从右边移

出，0从左边移入。当TB8处在移位寄存器的输出位置时，停止位刚好在TB8的左边，左边的所有位都是0。这个情况标志着TX控制单元作最后一次移位，同时停止发送，置TI为1。这出现在“写SBUF”后16分频寄存器的第11次滚动溢出结束时。

检测到RXD上1到0的变化，接收便被初始化了。无论波特率如何设置，RXD都以16倍的速率采样。当检测到一个变化，16分频的计数器立即复位，同时将1FFH写入输入移位寄存器。

在每个位时间的第7、第8、第9个计数状态，位检测器采样RXD上的值。一个被认可的值是被采3次至少有2次采到的一样。如果被认可的值在第一个位时间不是0，接收电路将被复位，控制单元将返回检测另一个1到0的变化。如果提供的起始位有效，它将被移入输入移位寄存器，剩下的数据接收将会继续进行。

当数据位从右边进来时，1从左边移出。当起始位到达移位寄存器的最左边的位置（模式2和3时，是9位寄存器），通知RX控制单元，作最后一次移位，装载SBUF和RB8，同时置RI为1。

如果在最后一个移位脉冲产生时，满足下列条件，数据载入SBUF和RB8，设置RI的信号被产生：

3. RI=0
4. SM2=0或接收的第9位数据位=1

如果这些条件没有一个满足，接收状态就被丢失，RI不会被置1。如果两个条件都满足，接收的第9位数据进入RB8，前面的8位数据进入SBUF。一个位时间后，无论是否满足上面的条件，控制单元都会返回，检测RXD输入线上1到0的变化。

图26. 串口模式0

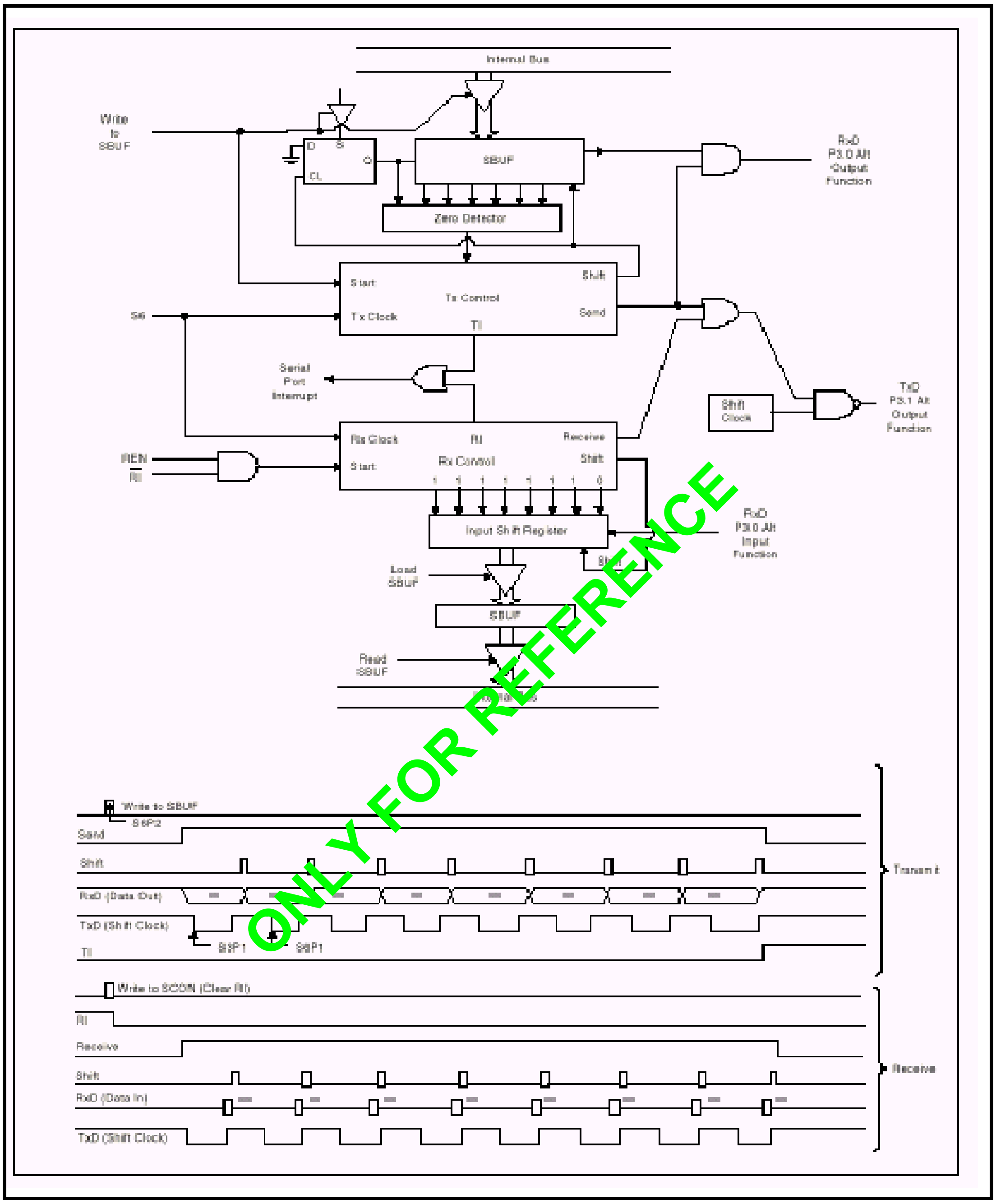

Internal Bus
Write to SBUF
D
S
Q
CL
SBUF
Zero Detector
Start
Shift
Tx Control
Tx Clock
Send
TI
S6
Serial Port Interrupt
RxD P3.0 Alt Output Function
Shift Clock
TxD P3.1 Alt Output Function
Rx Clock
RI
Receive
REN
RI
Start
Rx Control
Shift
Input Shift Register
RxD P3.0 Alt Input Function
Shift
Load SBUF
SBUF
Read SBUF
Write to SBUF
S6P2
Send
Shift
RxD (Data Out)
TxD (Shift Clock)
TI
S3P1
S6P1
Transmit
Write to SCON (Clear RI)
RI
Receive
Shift
RxD (Data In)
TxD (Shift Clock)
Receive

图27. 串口模式1

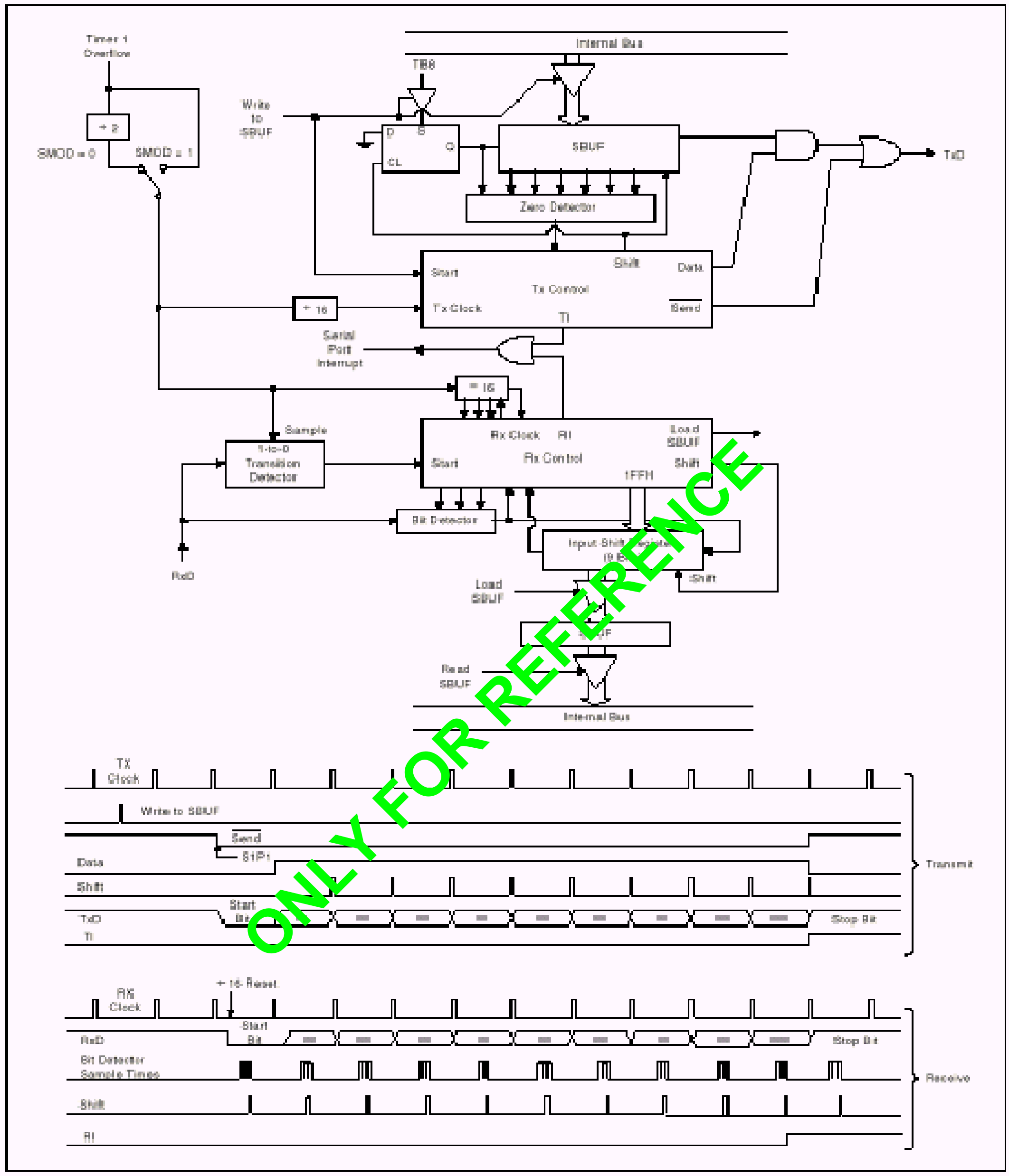

Timer 1 Overflow
Internal Bus
TB8
Write to SBUF
SBUF
SMOD = 0
SMOD = 1
Zero Detector
TxD
Start
Shift
Data
Tx Control
Tx Clock
Send
TI
Serial Port Interrupt
Sample
1-to-0 Transition Detector
Rx Clock
RI
Load SBUF
Rx Control
1FFH
Bit Detector
RxD
Load SBUF
Read SBUF
Internal Bus
TX Clock
Write to SBUF
Data
Shift
Start Bit
Stop Bit
Transmit
RX Clock
Bit Detector Sample Times
Receive

图28. 串口模式2

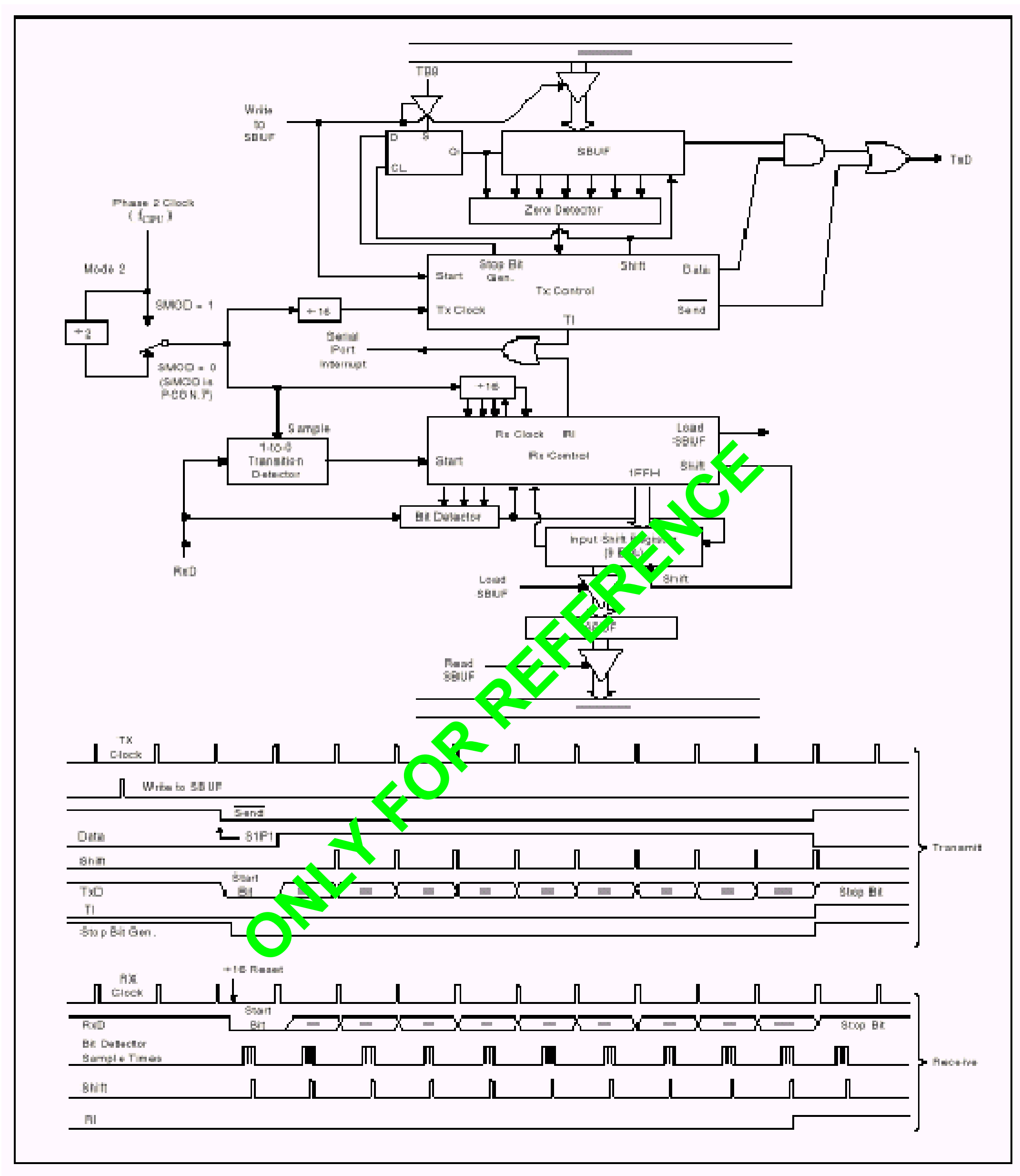
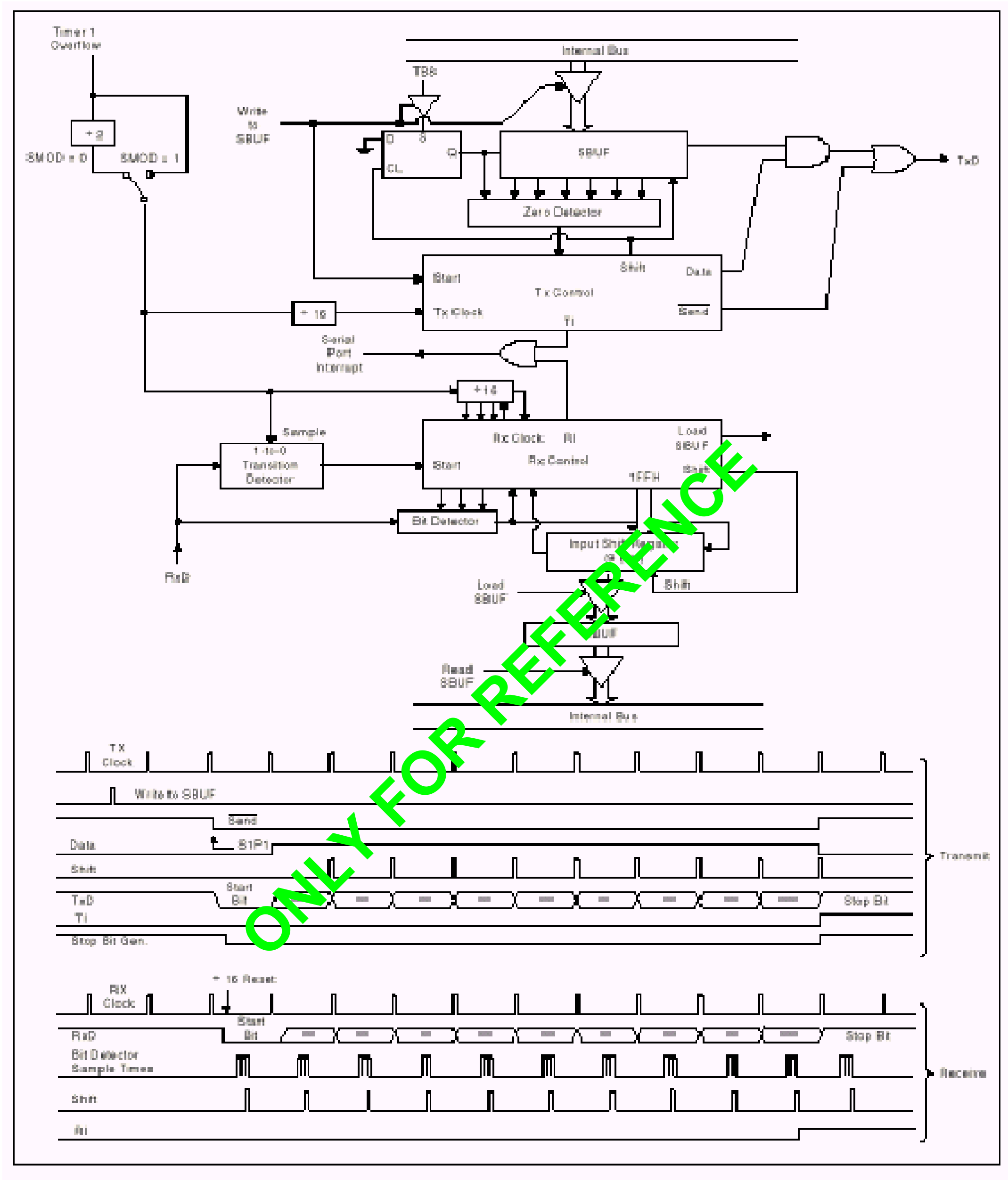

TB8
Write to SBUF
SBUF
Zero Detector
Phase 2 Clock
Mode 2
SMOD = 1
SMOD = 0
Stop Bit Gen.
Start
Shift
Data
Tx Control
Tx Clock
TI
Send
TxD
Serial Port Interrupt
Sample
1-to-0 Transition Detector
Rx Clock
RI
Load SBUF
Rx Control
Start
1FFH
Shift
Bit Detector
Input Shift Register (9 Bits)
RxD
Load SBUF
Read SBUF
TX Clock
Write to SBUF
Send
Data
S1P1
Shift
TxD
Start Bit
Stop Bit
TI
Stop Bit Gen.
Transmit
÷16 Reset
RX Clock
RxD
Start Bit
Stop Bit
Bit Detector Sample Times
Shift
RI
Receive

图29. 串口模式3


Timer 1
Overflow
Internal Bus
TB8
Write
to
SBUF
SMOD = 0
SMOD = 1
SBUF
Zero Detector
Start
Tx Control
Shift
Data
Tx Clock
TI
Send
TxD
Serial
Port
Interrupt
Sample
1-to-0
Transition
Detector
Rx Clock
RI
Load
SBUF
Rx Control
1FFH
Bit Detector
Input Shift Register
Shift
RxD
Load
SBUF
Read
SBUF
Internal Bus
TX
Clock
Write to SBUF
Send
Data
S1P1
Shift
TxD
Start
Bit
Stop Bit
TI
Stop Bit Gen.
Transmit
RX
Clock
÷ 16 Reset
RxD
Start
Bit
Stop Bit
Bit Detector
Sample Times
Shift
RI
Receive

模数转换（ADC）

模数转换器将一个模拟输入转换成相应的8位数字值。A/D模块有四路模拟输入，四路模拟输入是多路复用，进入同一采样和保持电路。采样保持电路的值输入一个转换器，通过连续近似，产生结果。模拟供电电压连接到A/D模块的梯形电阻网络引脚AVREF。

A/D模块有两个寄存器，控制寄存器ACON和A/D结果寄存器ADAT。表46给出的ACON寄存器控制A/D转换器模块的操作。寄存器P1SFS用于选择使用的模拟输入I/O引脚。一个8位预分频ASCL将主系统时钟分割为近似6MHZ时钟，提供给ADC逻辑。需要载入基于主MCU时钟频率的恰当值。

当开始位ADST置1，转换过程开始。一个周期后，它被硬件清零。ADAT寄存器的值为A/D转换的结果。当转换完成，结果载入ADAT，同时A/D转换状态位ADSF被置为1。图30为A/D模块的部件框图。当A/D转换完成，A/D状态位ADSF被自动置1，当处于A/D转换过程中，A/D状态位ADSF又被自动清零。

图30. A/D部件框图

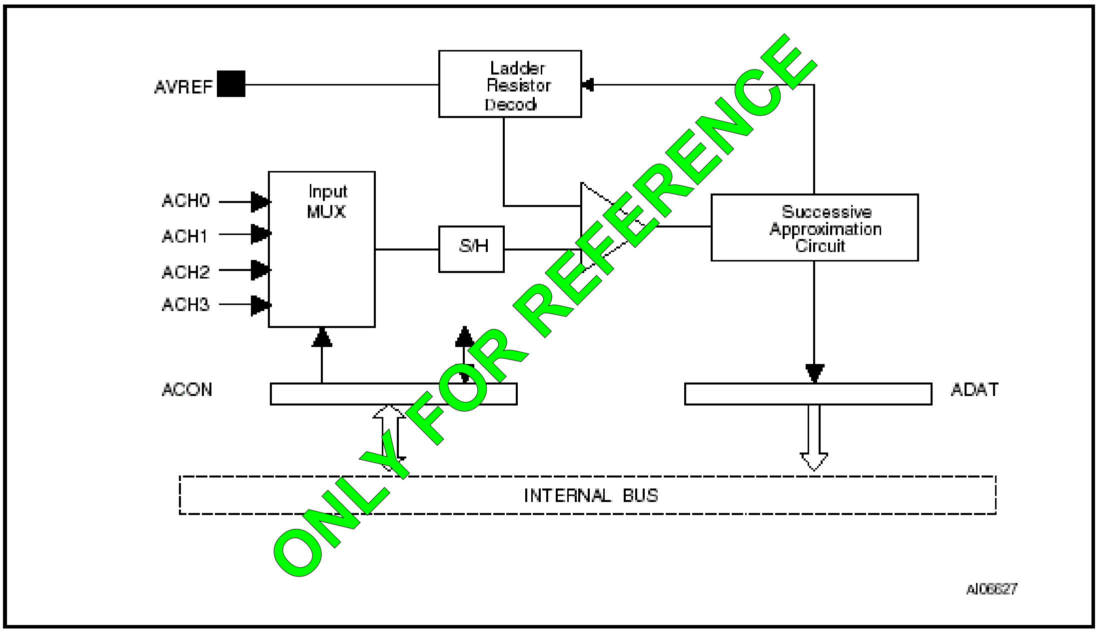


表45. ADC特殊功能寄存器内存印象图

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
95	ASCL									00	8-bit Prescaler for ADC clock
96	ADAT	ADAT7	ADAT6	ADAT5	ADAT4	ADAT3	ADAT2	ADAT1	ADAT0	00	ADC Data Register
97	ACON			ADEN		ADS1	ADS0	ADST	ADSF	00	ADC Control Register

表46. ACON寄存器各位的定义

Bit	Symbol	Function	
7 to 6	—	Reserved	
5	ADEN	ADC enable bit:	0 : ADC shut off and consumes no operating current
			1 : enable ADC
4	—	Reserved	
3 to 2	ADS1, ADS0	Analog channel select	
	0, 0		Channel0 (ACH0)
	0, 1		Channel1 (ACH1)
	1, 0		Channel2 (ACH2)
	1, 1		Channel3 (ACH3)
1	ADST	ADC start bit:	0: force to zero
			1 : start an ADC; after one cycle, bit is cleared to "0"
0	ADSF	ADC status bit:	0 : A/D conversion is in process
			1 : A/D conversion is completed, not in process

为了提供近似6MHZ时钟频率，ASCL为必需载入的一个8位分频数值，数值的计算公式如下：
ADC时钟输入=（Fosc/2）/（分频寄存器的值+1）
这里，Fosc是MCU时钟输入频率
ADC转换时间可以如下计算：
ADC转换时间=8时钟×8位×（ADC时钟）约等于10.67us(在 6 MHZ)

数模转换中断

当转换完成，结果载入ADAT，同时A/D转换状态位ADSF被置为1。该位可以被处理器控制，也能将其配置成一个下降沿触发的中断。

ADSF中断允许由PCON寄存器中的ADSFINT位设置。一旦该位被设置，外部INT1中断将被ADSF中断取代而失效。但外部中断1必须被设置为边沿触发中断输入。此时，EXINT1脚（P3.3）还可以用作通用I/O口，或者定时器1的门控制。

脉宽调制（PWM）

PWM单元有如下特点：

■ 带有16位预分频的4通道8位PWM
■ 带有可编程频率和脉宽的1通道8位PWM
■ 极性可编程

PWM的8位计数器计算的数从 0 到255，共256个值。保存在8位计数器里的数值相应PWM的特殊功能寄存器（PWMn）的内容比较。

PWMCON寄存器的PWMLVL位是可编程的，可以选择PWM的输出有无极性。倘若PWMn寄存器的内容等于或大于计数器的值，相应的PWM输出被设置为高（当PWMLVL=“0”）。当这个寄存器的内容小于等于计数器的值，相应的PWM输出被设置为低（当PWMLVL=“0”）。占空比是通过PWM的相应的特殊功能寄存器（PWMn）来设置的。通过给相应的特殊功能寄存器置入00H或FFH，PWM输出可以

分别保持在常高或常低电平（当PWMLVL=“0”）。

对于每个PWM单元，都有一个16位的预分频，用于分频主系统时钟，为相应的PWM单元产生输入时钟。这个预分频为PWM单元定义一个期望的循环率。SFR寄存器的B1H到B2H用来装载这个16位的除数。PWM输出的循环频率如下：

Fpwm8=(Fosc / 预分频0)/(2×256)或

同时，PWM计数器的输入时钟频率＝Fosc/2/(预分频数据+1)。

表47. PWM特殊功能寄存器内存映射图

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
A1	PWMCON	PWML	PWMP	PWME	CFG4	CFG3	CFG2	CFG1	CFG0	00	PWM Control Polarity
A2	PWM0									00	PWM0 Output Duty Cycle
A3	PWM1									00	PWM1 Output Duty Cycle
A4	PWM2									00	PWM2 Output Duty Cycle
A5	PWM3									00	PWM3 Output Duty Cycle
AA	PWM4H									00	PWM4 Output Duty (High)
AB	PWM4L									00	PWM4 Output Duty (Low)
B1	PSCL0L									00	Prescaler 0 Low (8-bit)
B2	PSCL0H									00	Prescaler 0 High (8-bit)
B3	PSCL1L									00	Prescaler 1 Low (16-bit)
B4	PSCL1H									00	Prescaler 1 High (16-bit)

注意：PWMCON寄存器位定义：

PWML = PWM0～3极性控制

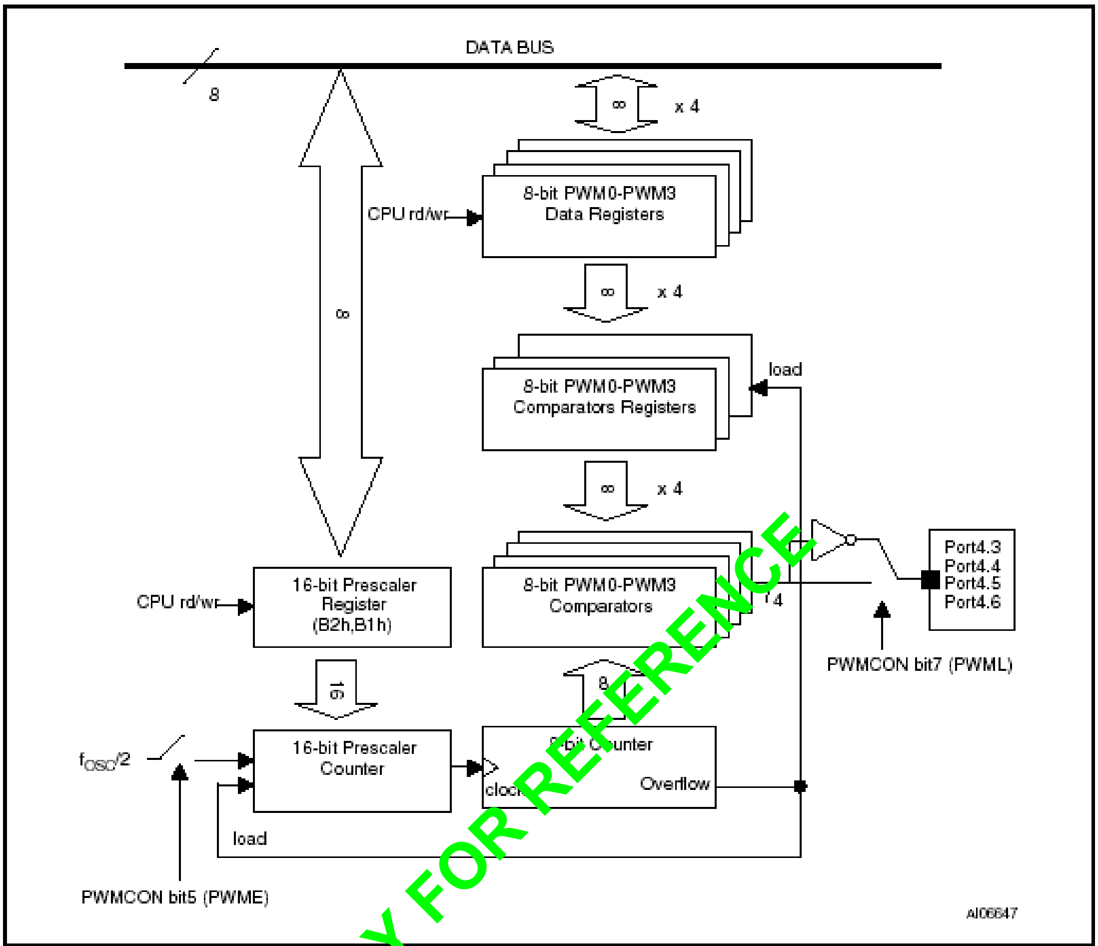
PWMP = PWM4极性控制

PWME = PWM使能（0=禁止，1=使能）

CFG4 = 16位PWM输出（0=开漏，1=推挽）

CFG3,CFG0 = 8位PWM输出（0=开漏，1=推挽）

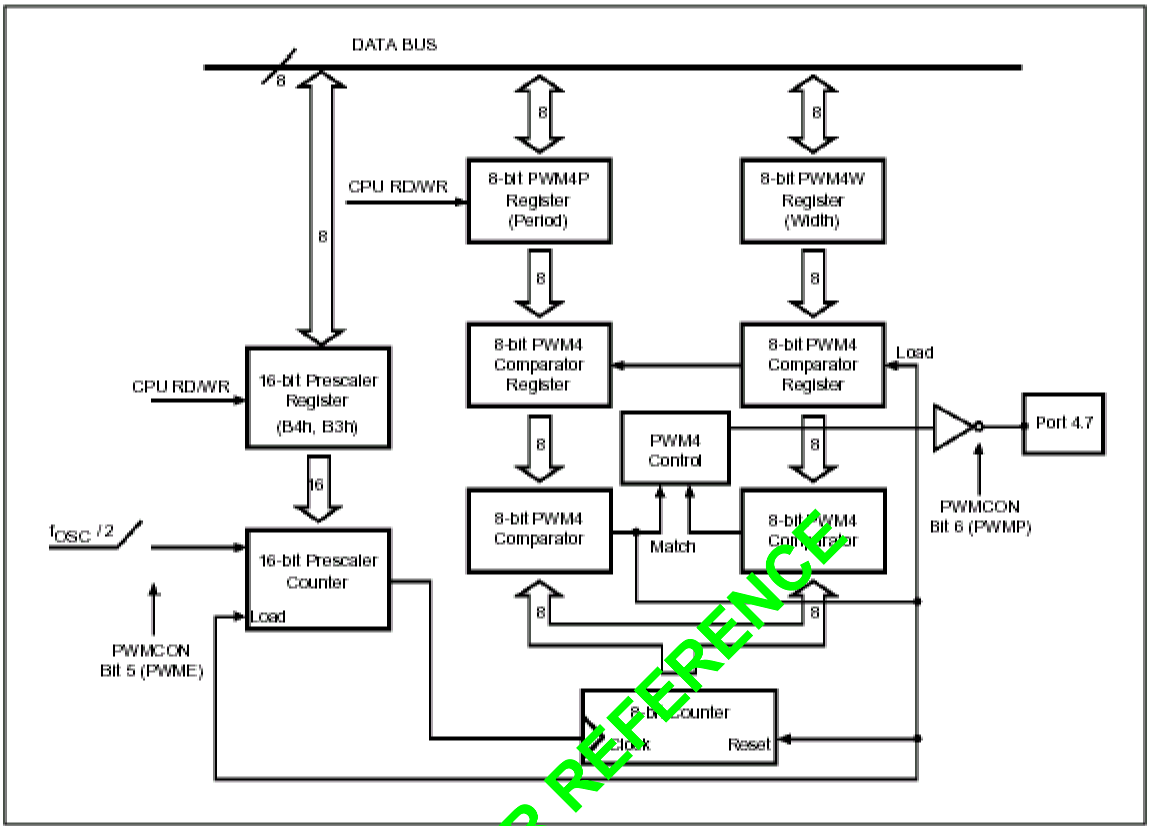
图31. 四通到8位PWM部件框图



可编程周期的8位PWM

PWM4输出的脉宽和周期是可编程的。它有一个16位的预分频器，一个8位计数器，一个脉宽寄存器，一个周期寄存器组成。脉宽寄存器定义PWM脉冲宽度时间，同时周期寄存器定义PWM脉冲输出的周期。输入到预分频器的时钟是系统时钟的一半（$f_{osc}/2$）。PWM4输出在P4.7端口。

图32. 可编程的PWM4通道框图



PWM4通道操作

16位的预分频器将输入时钟（$f_{osc}/2$）16分频给PWM4通道的8位计数器。PWM4计数器的输入时钟频率为：

PWM4计数器的输入时钟频率＝Fosc/2/(预分频数据+1)

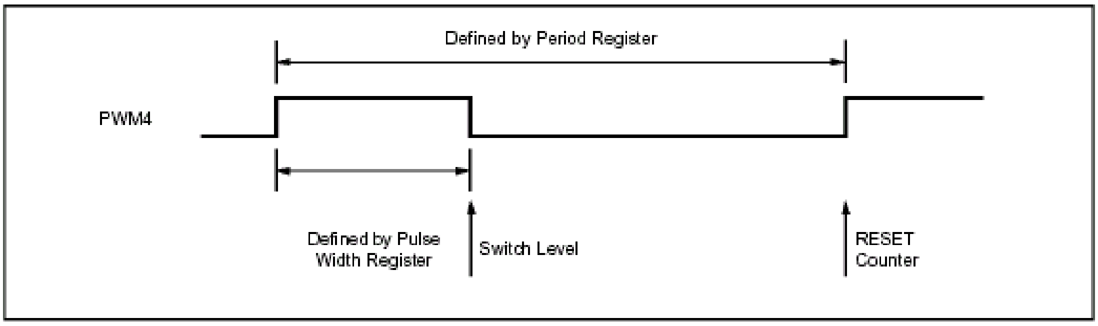
当预分频寄存器（B4H，B3H）的值为0时，PWM4计数器最大的输入时钟频率可高达20MHz。

PWM4的计数器是一个自由运行的8位计数器，它的输出与比较寄存器进行比较，比较寄存器的值由脉宽寄存器（PWM4W，ABH）和周期寄存器（PWM4P，AAH）装入。脉宽寄存器定义PWM脉冲宽度时间，同时周期寄存器定义PWM脉冲输出的周期。当PWM4被使能，它们的被装入比较寄存器，然后同PWM4的计数器进行比较。当计数器的值大于或等于脉宽寄存器的值时，PWM4通道的输出为“低”（当PWMP位＝0）。当周期寄存器的值等于PWM4的计数器的值时，计数器被清零，且PWM4通道的输出为“高”（下一个脉冲开始时）。

预分频器，脉宽寄存器和周期寄存器能在PWM4通道工作的时候修改。这些寄存器的值能在当前PWM4脉冲输出周期结束时自动装入预分频器和比较寄存器。

PWMCON寄存器的第5位和第6位控制PWM4通道的使能和极性。

图32-A. 可编程脉宽和频率的PWM4通道



I2C接口

μPSD3200系列有两个串行I2C接口。

这个串口支持两线的I2C总线，数据线（SDAx）和时钟线（SCLx）。

根据配置，SDA和SCL线上可能需要上拉电阻。

- SDA1，SCL1：为DDC协议提供的串口线
- SDA2，SCL2：为普通I2C总线连接提供的串口线

I2C接口的这些连线同时也可作为普通I/O口，如下：

- SDA1/P4.0，SCL1/P4.1，SDA2/P3.6，SCL2/P3.7

因为系统的特点，所以数据传输、时钟发生、地址识别和总线仲裁控制结构都被硬件控制。I2C串行I/O端口的字节处理和操作都是完全自主完成四种工作方式。四种方式如下：

- 主发送
- 主接收
- 从发送
- 从接收

这些功能都被以下的特殊功能寄存器所控制：

- SxCON：字节处理和四种方式的选择控制字
- SxSTA：寄存器的各位可以作为各种服务程序的向量标志
- SxDAT：数据移位寄存器
- SxADR：从地址寄存器。从地址的识别由芯片内的硬件完成。

图33. I2C总线串行I/O的功能框图

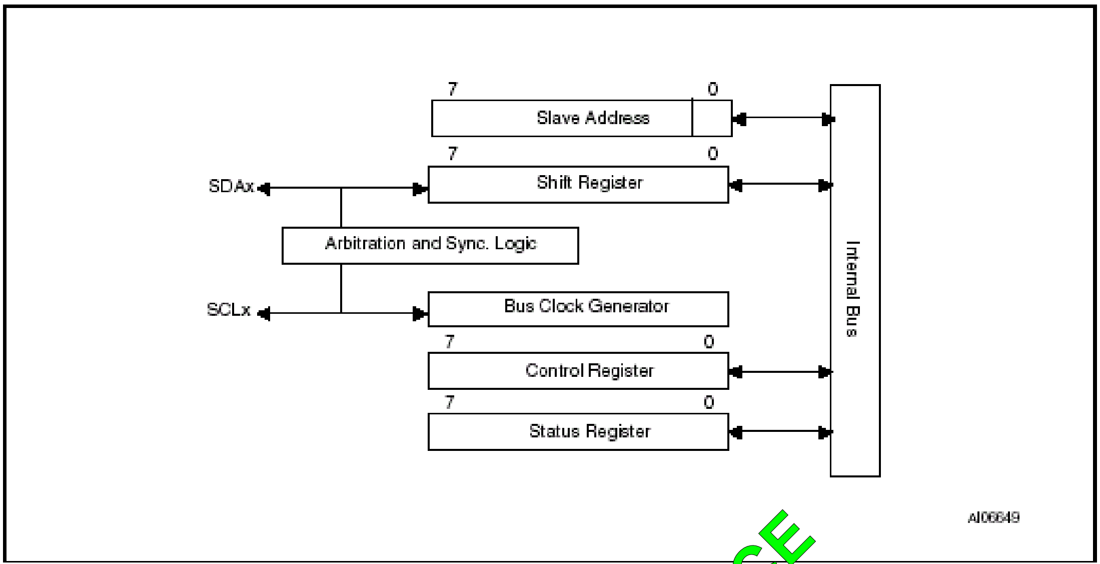


表48. 串行控制寄存器（SxCON：S1CON，S2CON）

7	6	5	4	3	2	1	0
CR2	ENII	STA	STO	ADDR	AA	CR1	CR0

表49. SxCON寄存器各位的定义

Bit	Symbol	Function
7	CR2	This bit along with bits CR1and CR0 determines the serial clock frequency when SIO is in the Master mode.
6	ENII	Enable IIC. When ENI1 = 0, the IIC is disabled. SDA and SCL outputs are in the high impedance state.
5	STA	START flag. When this bit is set, the SIO H/W checks the status of the I2C-bus and generates a START condition if the bus free. If the bus is busy, the SIO will generate a repeated START condition when this bit is set.
4	STO	STOP flag. With this bit set while in Master mode a STOP condition is generated. When a STOP condition is detected on the I2C-bus, the I2C hardware clears the STO flag. Note: This bit have to be set before 1 cycle interrupt period of STOP. That is, if this bit is set, STOP condition in master mode is generated after 1 cycle interrupt period.
3	ADDR	This bit is set when address byte was received. Must be cleared by software.
2	AA	Acknowledge enable signal. If this bit is set, an acknowledge (low level to SDA) is returned during the acknowledge clock pulse on the SCL line when: • Own slave address is received • A data byte is received while the device is programmed to be a Master Receiver • A data byte is received while the device is a selected Slave Receiver. When this bit is reset, no acknowledge is returned. SIO release SDA line as high during the acknowledge clock pulse.
1	CR1	These two bits along with the CR2 bit determine the serial clock frequency when SIO is in the Master mode.
0	CR0	

表50. 在主模式下串行时钟频率SCL的选择

CR2	CR1	CR0	Fosc Divisor	Bit Rate (kHz) at Fosc			
				12 MHz	24 MHz	36 MHz	40 MHz
0	0	0	16	375	750	X	X
0	0	1	24	250	500	750	833
0	1	0	30	200	400	600	666
0	1	1	60	100	200	300	333
1	0	0	120	50	100	150	166
1	0	1	240	25	50	75	83
1	1	0	480	12.5	25	37.5	41
1	1	1	960	6.25	12.5	18.75	20

串行状态寄存器（SxSTA：S1STA，S2STA）

SxSTA是只读寄存器。这个寄存器的内容作服务子程序的入口。这就优化了软件响应时间和I2C总线响应时间。对所有I2C总线接口可能的模式，状态代码在表52里给出。

表51. 串行状态寄存器（SxSTA）

7	6	5	4	3	2	1	0
GC	STOP	INTR	TX_MODE	BBUSY	BLOST	/ACK_REP	SLV

表52. SxSTA寄存器各位的定义

Bit	Symbol	Function
7	GC	General Call flag
6	STOP	Stop flag. This bit is set when a STOP condition is received
5	INTR	Interrupt flag. This bit is set when an I²C Interrupt condition is requested
4	TX_MODE	Transmission mode flag. This bit is set when the I²C is a transmitter; otherwise this bit is reset
3	BBUSY	Bus Busy flag. This bit is set when the bus is being used by another master; otherwise, this bit is reset
2	BLOST	Bus Lost flag. This bit is set when the master loses the bus contention; otherwise this bit is reset
1	/ACK_REP	Acknowledge Responseflag. This bit is set when the receiver transmits the not acknowledge signal This bit is reset when the receiver transmits the acknowledge signal
0	SLV	Slave mode flag. This bit is set when the I²C plays role in the slave mode; otherwise this bit is reset

注意：1. 中断标志位（INTR, SxSTA位5）当读SxSTA寄存器时被硬件清零。

2. I2C中断标志（INTR）可以出现在下列情况(除了DDC2B模式在SWENB=0)

在下列任意情况出现时，这个标志被置1，同时发生中断：

1．在AA=1期间，接收到自己的从地址：ack_int

2．当GC(SxADR.0)=1同时AA=1时，检测到全体呼叫地址：ack_int

3．主机模式接收或发送一个数据字节（即便判断数据丢失）：ack_int

4．作为被选择的从机，接收或发送一个数据字节：ack_int
5．作为被选择的从机，接收或发送一个停止状态：stop_int

数据移位寄存器（SXDAT: S1DAT，S2DAT）

SxDAT包含了发送或被接收的数据。数据从右到左移动，首先接收或发送的是最高位（bit7）。

表53. 数据移位寄存器（SXDAT：S1DAT，S2DAT）

7	6	5	4	3	2	1	0
SxDAT7	SxDAT6	SxDAT5	SxDAT4	SxDAT3	SxDAT2	SxDAT1	SxDAT0

地址寄存器（SxADR：S1ADR，S2ADR）

当编程是从机接收 / 发送时，8位寄存器可以载入控制器将响应的7位从机地址。

表54. 地址寄存器（SxADR）

7	6	5	4	3	2	1	0
SLA6	SLA5	SLA4	SLA3	SLA2	SLA1	SLA0	—

下面是和I2C单元相关的寄存器，用于配合高MCU系统频率值，指定工作的开始 / 停止检测时间。例如，在40MHZ的系统时钟一起。

表55. 开始 / 停止有效时间检测寄存器（S1SETUP,S2SETUP）

	Address	Register Name	Reset Value	Note
SFR	D1h	S1SETUP	00h	To control the start/stop hold time detection for the DDC module in slave mode
	D2h	S2SETUP	00h	To control the start/stop hold time detection for the multi-master I²C module in slave mode

表56. 40MHZ系统时钟

S1SETUP, S2SETUP Register Value	Number of Sample Clock (fosc/2 -> 50 ns)	Required Start/ Stop Hold Time	Note
00h	1 EA	50 ns	When bit 7 (enable bit) = 0, the number of sample clock is 1 EA (ignore bit 6 to bit 0)
80h	1 EA	50 ns	
81h	2EA	100 ns	
82h	3EA	150 ns	
...	...	...	
8Bh	12EA	600 ns	Fast Mode I²C Start/Stop hold time specification
...	...	...	
FFh	128EA	6000 ns	

表57. 示例

System Clock	S1SETUP, S2SETUP Register Value	Number of Sample Clock	Required Start/Stop Hold Time
40 MHz (f_{OSC}/2 -> 50 ns)	8Bh	12 EA	600 ns
30 MHz (f_{OSC}/2 -> 66.6 ns)	89h	9 EA	600 ns
20 MHz (f_{OSC}/2 -> 100 ns)	86h	6 EA	600 ns
8 MHz (f_{OSC}/2 -> 250 ns)	83h	3 EA	750 ns

PSD模块

■ PSD模块为8032CPU核心提供可配置的程序和数据存储器。另外，为产生通用逻辑，它有自己的I/O端口设置和16个宏单元的PLD。

■ A、B、C和D是通用可编程I/O端口，它的端口结构与MCU模块的I/O端口是不同的。

■ PSD模块通过内部地址、数据总线（A0～A15，D0～D7）和控制信号（／RD，／WR，／PSEN，ALE，／RESET）与MCU模块通讯。用户通过PSDsoft开发工具定义地址译码PLD，可将PSD模块的资源映射到任意程序或地址空间。图44给出了PSD模块的功能部件。

功能概述

■ 512KFlash存储器。这是主Flash存储器。它分为4个相等大小的存储器块，可以按用户指定的地址来访问。

■ 次128K位Flash引导存储器。它分为2个大小相等的存储器块，可以按用户指定的地址来访问。此次存储器可执行代码，同时也可更新主Flash的内容。

■ 16K位SRAM。在掉电的情况下，SRAM的内容可以通过外接备份电池来保护。

■ 有16个输出宏单元和20个输入宏单元的CPLD。CPLD可用于有效地实现多种用于内部和外部控制的逻辑功能。例如状态机，可装载移位寄存器和可装载计数器。

■ 地址译码PLD（DPLD）用于选择PSD模块中存储器块的地址译码。

■ 可配置的I/O端口（端口A、B、C和D），可用于以下功能

－ MCU I/O

－ PLD I/O

－ 锁存的MCU地址输出

－ 特殊功能I/O

－ 漏极开路I/O

－ 符合JTAG标准的串行口可对芯片进行在系统编程（ISP）。你可以在工厂或现场对一个空白器件进行编程或者对一个已编程的芯片重新编程。

－ 内部页寄存器可以扩展MCU模块寻址空间的256倍

－ 支持低功耗模式的内部可编程功率管理单元（PMU）可以自动检测8032 CPU工作空隙，并将PSD模块置入掉电模式

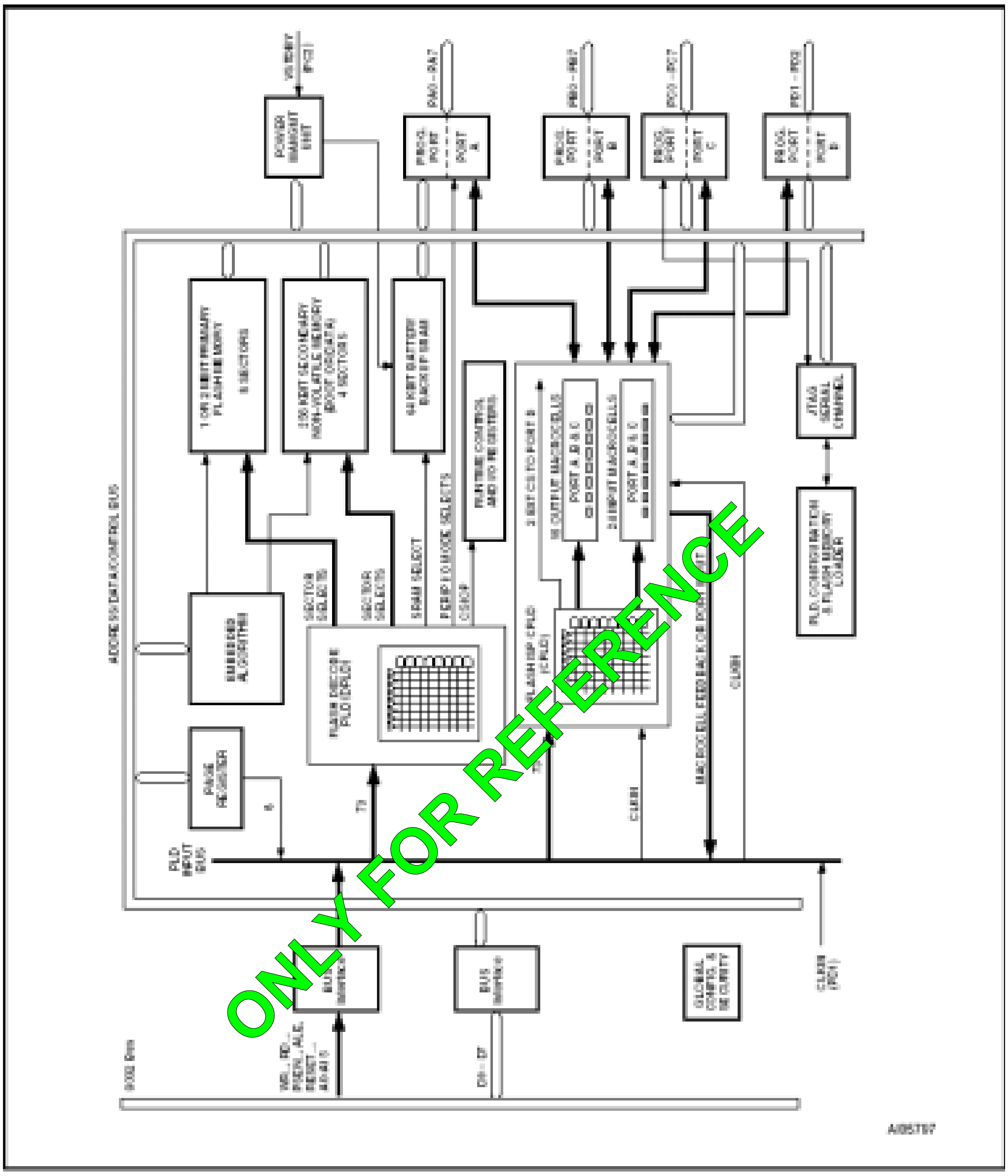
■ 擦／写次数

－ Flash存储器——至少100,000 次

－ PLD——至少1,000 次

－ 数据保存：至少15年（主Flash存储器，次Flash存储器，PLD和配置位）

PLD模块结构图



在系统编程（ISP）

使用端口C上的JTAG信号可以不需要MCU就可以对整个PSD模块进行编程和擦除。通过MCU执行来自第二存储器或SRAM的编程算法，主Flash存储器也能被在应用中编程。第二存储器也可以通过执行来自主Flash存储器的编程算法进行应用中 编程。PLD或其他PSD模块配置可通过JTAG口或器件编程器进行编程。表81表示那个编程方法可对PSD模块的不同功能块进行编程。

表81. PSD模块不同功能块的编程方法

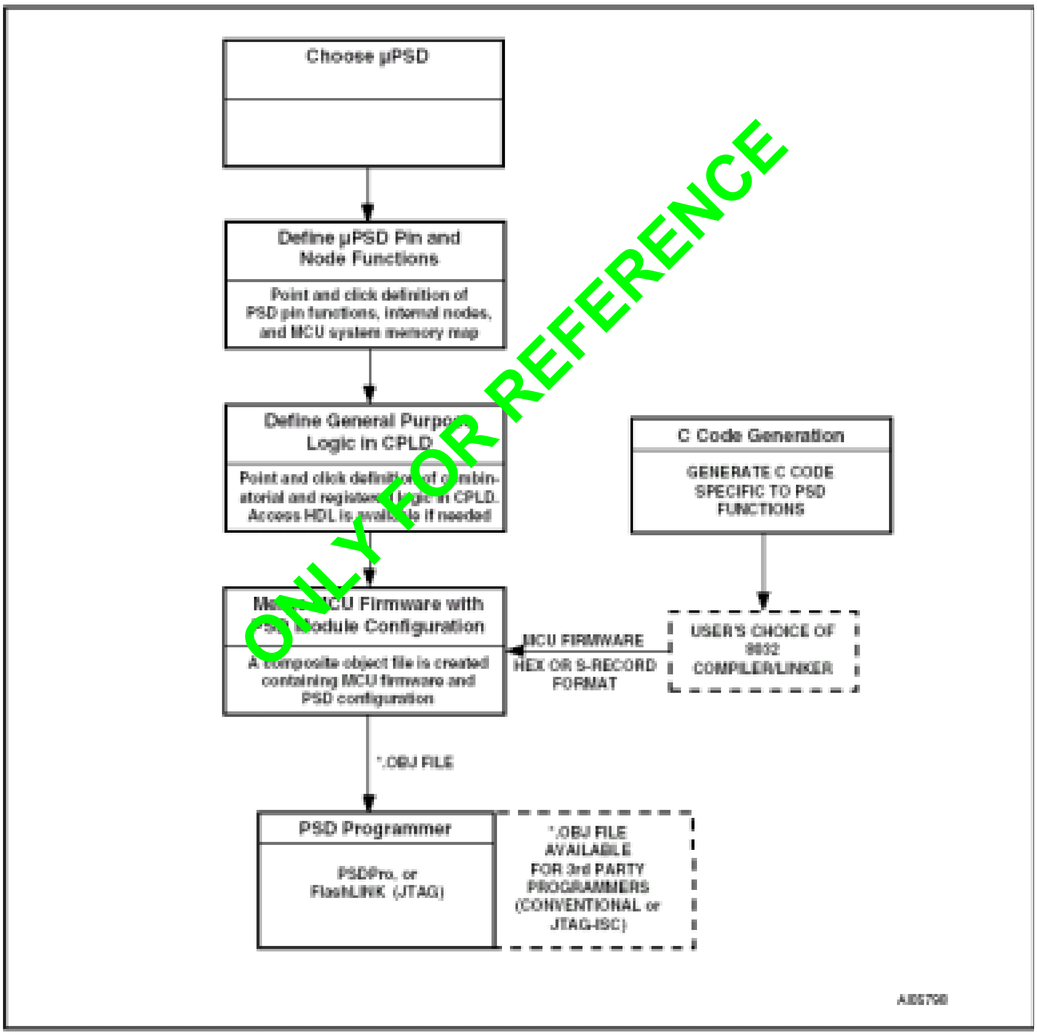
Functional Block	JTAG Programming	Device Programmer	IAP
Primary Flash Memory	Yes	Yes	Yes
Secondary Flash Memory	Yes	Yes	Yes
PLD Array (DPLD and CPLD)	Yes	Yes	No
PSD Module Configuration	Yes	Yes	No

开发系统

基于WINDOWS的PSDsoft开发系统工具（Windows95，Windows98，Windows-NT）支持μPSD3200。只需要点击鼠标，便可快速简单地完成PSD模块的设计。设计人员不需要输入硬件描述语言（HDL）方程式，除非要定义PSD模块的引脚功能和存储器地址映射信息。图45给出了一般的设计流程。PSDsoft开发软件可以从我们的网站免费下载。

PSDsoft直接支持便宜的ST公司的编程工具：FlashLINK（JTAG）。这两个编程工具可以从当地发行商或代理商处购买，也可以使用信用卡从我们的网站购买。μPSD3200也支持第三方器件编程软件。查询当前网站上列出的信息。

图45. PSDsoft快速开发工具



PSD 模块寄存器描述和偏移地址

表82表示PSD模块寄存器相对于CSIOP基地址的偏移地址。CSIOP空间是256个字节空间，它由用户分配给内部PSD模块寄存器。表82给出了CSIOP空间寄存器的简要描述。下一节将更详细地描述。

表82. 寄存器地址偏移

Register Name	Port A	Port B	Port C	Port D	Other[1]	Description
Data In	00	01	10	11		Reads Port pin as input, MCU I/O input mode
Control	02	03				Selects mode between MCU I/O or Address Out
Data Out	04	05	12	13		Stores data for output to Port pins, MCU I/O output mode
Direction	06	07	14	15		Configures Port pin as input or output
Drive Select	08	09	16	17		Configures Port pins as either CMOS or Open Drain on some pins, while selecting high slew rate on other pins.
Input Macrocell	0A	0B	18			Reads Input Macrocells
Enable Out	0C	0D	1A	1B		Reads the status of the output enable to the I/O Port driver
Output Macrocells AB	20	20				Read – reads output of macrocells AB Write – loads macrocell flip-flops
Output Macrocells BC		21	21			Read – reads output of macrocells BC Write – loads macrocell flip-flops
Mask Macrocells AB	22	22				Blocks writing to the Output Macrocells AB
Mask Macrocells BC		23	23			Blocks writing to the Output Macrocells BC
Primary Flash Protection					C0	Read only – Primary Flash Sector Protection
Secondary Flash memory Protection					C2	Read only – PSD Module Security and Secondary Flash memory Sector Protection
JTAG Enable					C7	Enables JTAG Port
PMMR0					B0	Power Management Register 0
PMMR2					B4	Power Management Register 2
Page					E0	Page Register
VM					E2	Places PSD Module memory areas in Program and/or Data space on an individual basis.

注意：1. 不是I/O端口部分的其他寄存器

PSD模块详细操作

如图14给出的，PSD模块包括5个主要的功能块：

- 存储器块
- PLD块
- I/O端口
- 电源管理单元（PMU）
- JTAG接口

每个功能块的功能在下面各节中详细描述。这些块完成多种功能，且可由用户自行配置。

存储器块

PSD模块有下面这些存储器块：

- 主Flash存储器
- 第二Flash存储器

■ SRAM
存储器选择信号来自地址译码PLD（DPLD）并可在PSDsoft软件中由用户定义。

主Flash存储器和第二Flash存储器描述

主Flash存储器被分为4个大小相等的扇区。第二Flash存储器被分为2个大小相等的扇区。每个扇区可单独地保护，防止被编程和擦除。

Flash存储器可以一个扇区一个扇区地被擦除。当块的其他扇区读数据时，Flash扇区擦除会被暂停，数据读取完毕后，再恢复。

在Flash存储器编程和擦除期间，状态输出到Ready/Busy（PC3）。这个脚可以被PSDsoft配置为Ready/Busy方式

存储器块选择信号

DPLD为所有的内部存储器块产生选择信号（见110页，被定义的“PLDs”）。主存储器的4个扇区，每个都有一个选择信号（FS0～FS3），每一个选择信号包含3个乘积项。第二Flash存储器的2个扇区，每个都有一个选择信号（CSBOOT0～CSBOOT1），每个选择信号包含3个乘积项。每个扇区选择信号具有3个乘积项，允许被给定的段可被映射到不同的程序或数据空间。

Ready/Busy（PC3）。这个信号用于输出Flash存储器的准备就绪／忙信号。当Flash存储器正在被写或擦除时，Ready/Busy（PC3）输出0（表示“忙”）。当程序没有写或擦除时，输出为1（表示“准备就绪”）

存储器操作。主Flash存储器和第二Flash存储器是通过MCU总线寻址。MCU可以通过下面两种方式访问这些存储器：

■ MCU执行标准总线读写操作。

■ MCU执行包含几个读写操作的特殊Flash存储器指令。这包含写特殊数据格式到Flash存储器特定的地址，从而调用一个内置的算法。表83中概述了这些指令。

MCU可以象读ROM器件那样用读操作读取Flash存储器。然而，Flash存储器只能使用特殊的擦除和编程指令来更改。例如，MCU不能像写一个字节到RAM那样，直接写一个字节到Flash存储器。编程一个字节到Flash存储器，MCU必须执行一个编程指令，然后检测编程周期的状态。通过读操作或检测Ready/Busy（PC3） 来完成状态检测。

表83. 指令

Instruction	FS0-FS7 or CSBOOT0-CSBOOT3	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7
Read[5]	1	"Read" RD @ RA						
Read Main Flash ID[6]	1	AAh@ X555h	55h@ XAAAh	90h@ X555h	Read identifier (A6,A1,A0 = 0,0,1)			
Read Sector Protection[6,8,13]	1	AAh@ X555h	55h@ XAAAh	90h@ X555h	Read identifier (A6,A1,A0 = 0,1,0)			
Program a Flash Byte[13]	1	AAh@ X555h	55h@ XAAAh	A0h@ X555h	PD@ PA			
Flash Sector Erase[7,13]	1	AAh@ X555h	55h@ XAAAh	80h@ X555h	AAh@ XAAAh	55h@ XAAAh	30h@ SA	30h[7]@ next SA
Flash Bulk Erase[13]	1	AAh@ X555h	55h@ XAAAh	80h@ X555h	AAh@ XAAAh	55h@ XAAAh	10h@ X555h	
Suspend Sector Erase[11]	1	B0h@ XXXXh						
Resume Sector Erase[12]	1	30h@ XXXXh						
Reset[6]	1	F0h@ XXXXh						
Unlock Bypass	1	AAh@ X555h	55h@ XAAAh	20h@ X555h				
Unlock Bypass Program[9]	1	A0h@ XXXXh	PD@ PA					
Unlock Bypass Reset[10]	1	90h@ XXXXh	00h@ XXXXh					

注意：1. 所有的总线周期都是写总线周期，带有"Read"标签的除外。

2. 所有的值都是十六进制

X =不用管。表中的XXXXh地址必须是确切地址

RA =被读取位置存储器的地址

RD =读周期期间从位置RA读取数据

PA =被编程的存储器的地址，地址在写选通（WR,CNTL0）的下降沿被锁存。在字编程模式下，PA是PSD的一个确切地址。

PD =在地址PA被编程的数据字。数据在写选通（WR,CNTL0）的上升沿被锁存。

SA =被擦除或校验的扇区的地址。被擦除或校验的扇区的扇区选择（FS0~FS7或CSBOOT0~CSBOOT3）必须激活（高）。

3. 扇区选择（FS0~FS7或CSBOOT0~CSBOOT3）信号被激活为高，同时在PSDsoft内定义

4. 在指令解码中只有地址A11~A0被使用

5. 器件在读模式下不要求启动或指令周期。

6. 在读取Flash的ID或读取扇区保护状态后，或者如果错误标志（DQ5/DQ13）位变高，复位指令要求返回读模式。

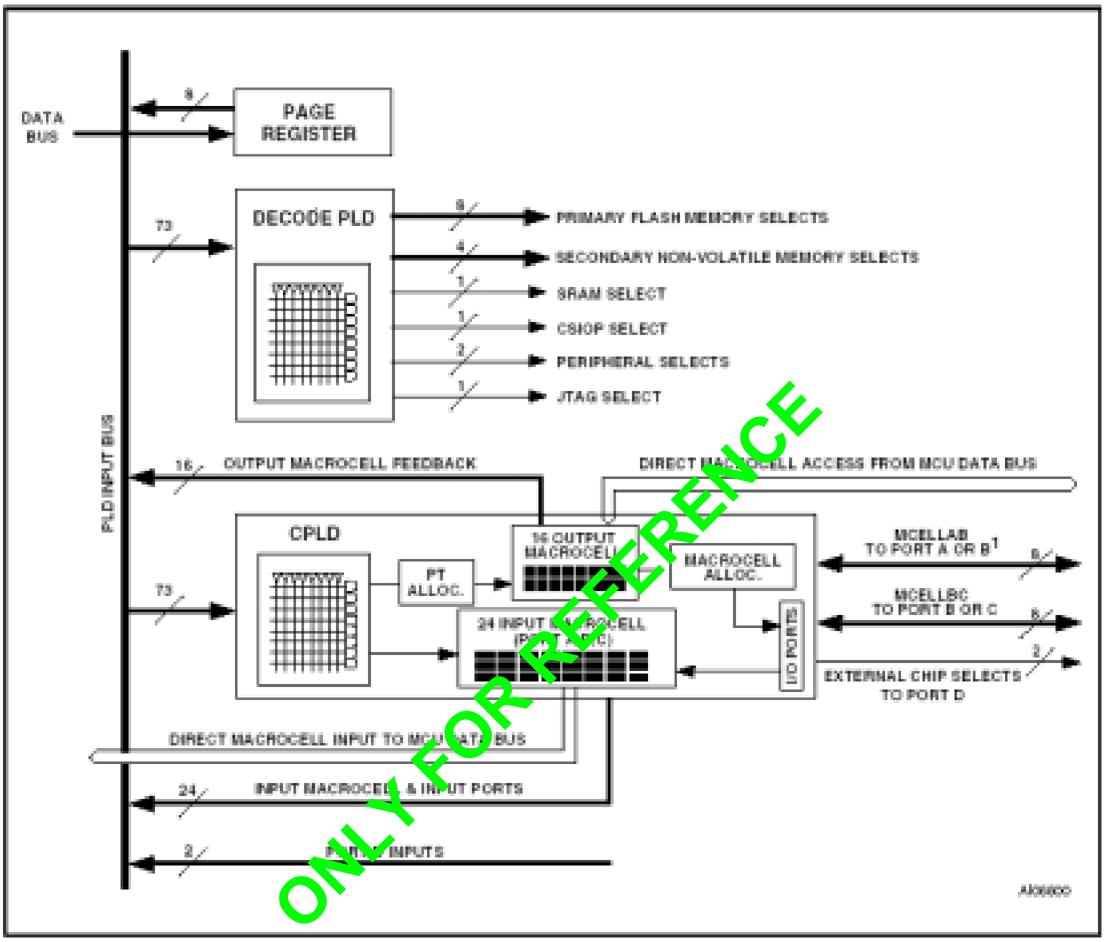
7. 在扇区擦除指令结束后80us内，另外被擦除的扇区必须被写。

8. 对于一个没被保护的扇区，数据是00H，对于一个被保护的扇区，数据是01H，在第四个周期，扇区选择被激活，同时（A1,A0）=（1,0）。

9. 当扇区擦除暂停模式时，在没有正在被擦除的扇区，系统可以完成读和编程周期，读取Flash ID读扇区保护状态。暂停扇区擦除指令仅仅在扇区擦除周期内有效。

没有在PLD逻辑公式中使用时，才可以被使用。每个PLD都有适合自身的唯一的特征。将在下面的章节里描述。

图52. PLD框图



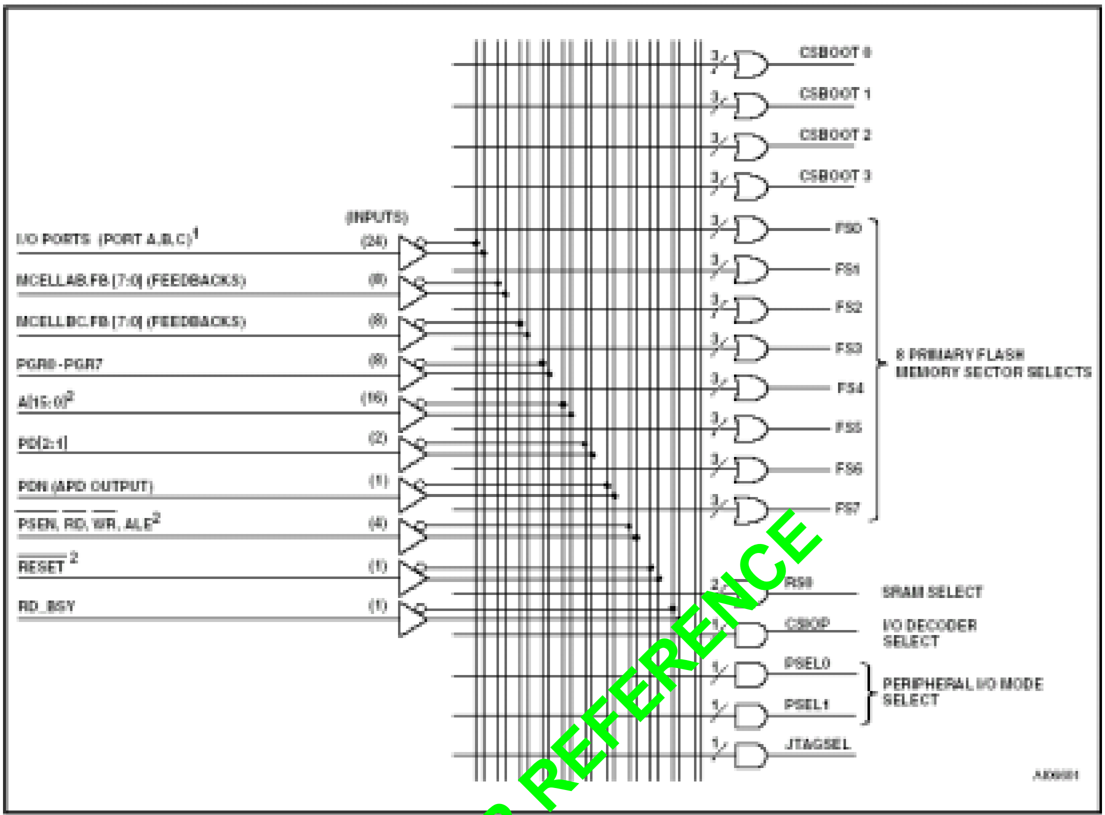
注意：1. 端口A在52脚封装器件上是无效的。

可编程逻辑地址译码（DPLD）

图53给出的DPLD用于为PSD模块和外部部件进行地址译码。DPLD可用于产生如下译码信号：

- 主Flash存储器的4个扇区选择信号（FS0～FS3）（每个信号有3个乘积项）
- 次Flash存储器的2个扇区选择信号（CSBOOT0～CSBOOT1）（每个信号有3个乘积项）
- 1个内部SRAM选择信号（RS0）（每个信号有2个乘积项）
- 1个内部CSIOP选择信号（选择PSD模块寄存器）
- 1个JTAG选择信号（激活端口C上的JTAG）
- 2个内部的外设选择信号（外部I/O模式）

图53. DPLD逻辑阵列



注意：1. 端口A输入在52脚封装芯片中是无效的。

2. 来自MCU模块的输入。

复杂可编程逻辑（CPLD）

CPLD用于执行系统逻辑功能，如：可装载计数器、移位寄存器、系统邮箱、握手协议、状态机和随机逻辑。CPLD可通过端口D，用于产生外部芯片选择（ECS1～ECS2）。尽管外部芯片选择（ECS1～ECS2）可以通过任何输出宏单元（OMC）产生，端口D上的这些外部芯片选择（ECS1～ECS2）不会占用任何输出宏单元。如图52表示的那样，CPLD有如下的功能块：

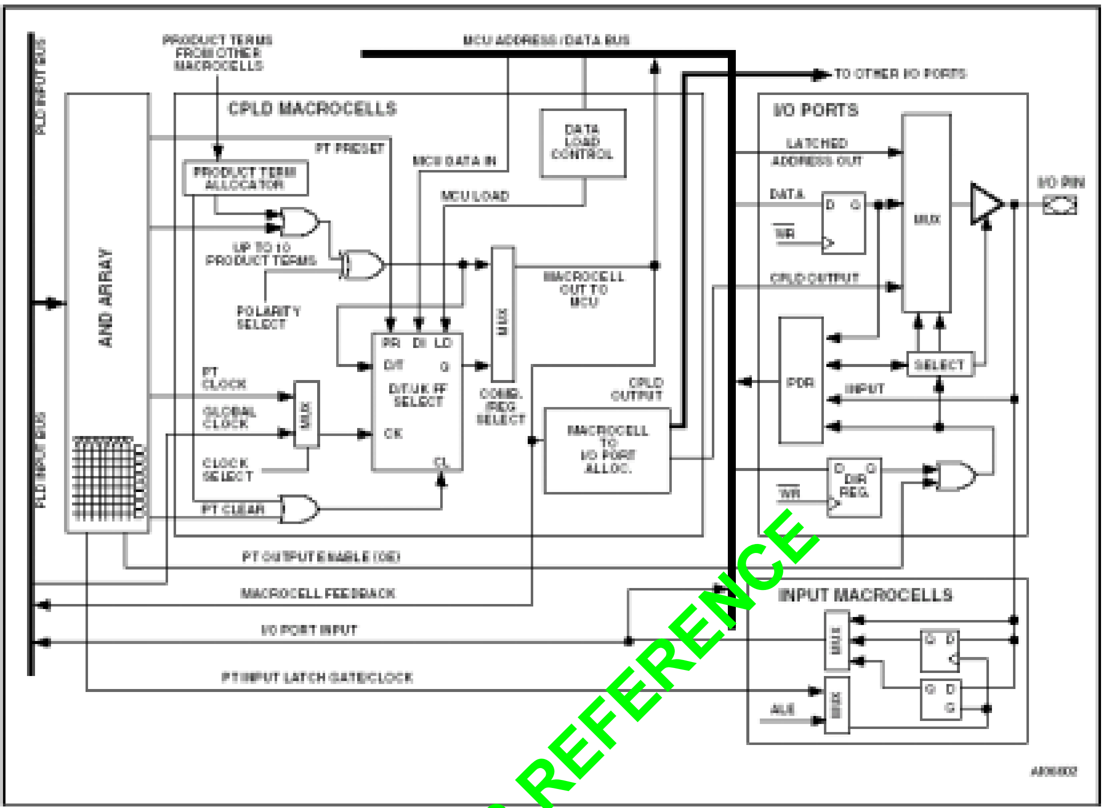
- 24个输入宏单元（IMC）
- 16个输出宏单元（OMC）
- 宏单元分配符
- 乘积项分配符
- AND阵列可产生137个乘积项
- 四个I/O端口

下面描述了每一个功能块。

输入宏单元（IMC）和输出宏单元（OMC）连接到PSD模块的内部数据总线，可被MCU直接寻址。MCU软件可将数据载入输出宏单元（OMC），可从输入和输出宏单元读取数据。

这个特点允许系统逻辑的更有效地执行，而不需要象标准PLD宏单元体系要求的那样，将数据总线连接到AND阵列。

图54. 宏单元和I/O端口



输出宏单元（OMC）

有8个输出宏单元连接到端口A和端口B，取名为McellAB0～McellAB7。另外8个宏单元连接到端口B和端口C，取名为McellBC0～McellBC7。如果McellAB输出没有被PSDsoft指定为特殊功能脚，那么宏单元分配符功能块将指定它为端口A或端口B。端口B或C上的McellBC输出也是同样的。表89表示了宏单元和端口分配。

图55给出了输出宏单元（OMC）的结构图。图中表示了来自AND阵列的有用的原乘积项和来自其他输出宏单元（OMC）的有用的乘积项。乘积项的极性被XOR门控制。输出宏单元（OMC）可以用触发器实现时序逻辑或组合逻辑。多路转换器选择时序逻辑或组合逻辑作为宏单元的输出。多路转换器输出可驱动端口引脚和具有反馈通路AND阵列输入端。

输出宏单元（OMC）功能块的触发器可以通过PSDsoft配置成D、T、JK或SR类型。触发器的时钟、预置、清除输入可由AND阵列的乘积项来驱动。另外，CLKIN（PD1）也可用于触发器的时钟输入。触发器时钟同步在CLKIN（PD1）的上升沿。预置和清除高电平输入有效。每个清除输入能使用到两个的乘积项。

表89. 输出宏单元口和数据位分配

Output Macrocell	Port Assignment[1]	Native Product Terms	Maximum Borrowed Product Terms	Data Bit for Loading or Reading
McellAB0	Port A0, B0	3	6	D0
McellAB1	Port A1, B1	3	6	D1
McellAB2	Port A2, B2	3	6	D2
McellAB3	Port A3, B3	3	6	D3
McellAB4	Port A4, B4	3	6	D4
McellAB5	Port A5, B5	3	6	D5
McellAB6	Port A6, B6	3	6	D6
McellAB7	Port A7, B7	3	6	D7
McellBC0	Port B0, C0	4	5	D0
McellBC1	Port B1, C1	4	5	D1
McellBC2	Port B2, C2	4	5	D2
McellBC3	Port B3, C3	4	5	D3
McellBC4	Port B4, C4	4	6	D4
McellBC5	Port B5, C5	4	6	D5
McellBC6	Port B6, C6	4	6	D6
McellBC7	Port B7, C7	4	6	D7

注意：1. McellAB0-McellAB7在52脚封装的器件中仅仅能够被分配到端口B。

乘积项分配算符

CPLD有一个乘积项分配符。PSDsoft用乘积项分配符从一个宏单元借用乘积项放置到另一个宏单元。下面列出了乘积项如何分配：

- McellAB0～McellAB7都有3个原乘积项，并可借用其他宏单元6个以上的乘积项。
- McellBC0～McellBC3都有4个原乘积项，并可借用其他宏单元5个以上的乘积项。
- McellBC4～McellBC7都有4个原乘积项，并可借用其他宏单元6个以上的乘积项。

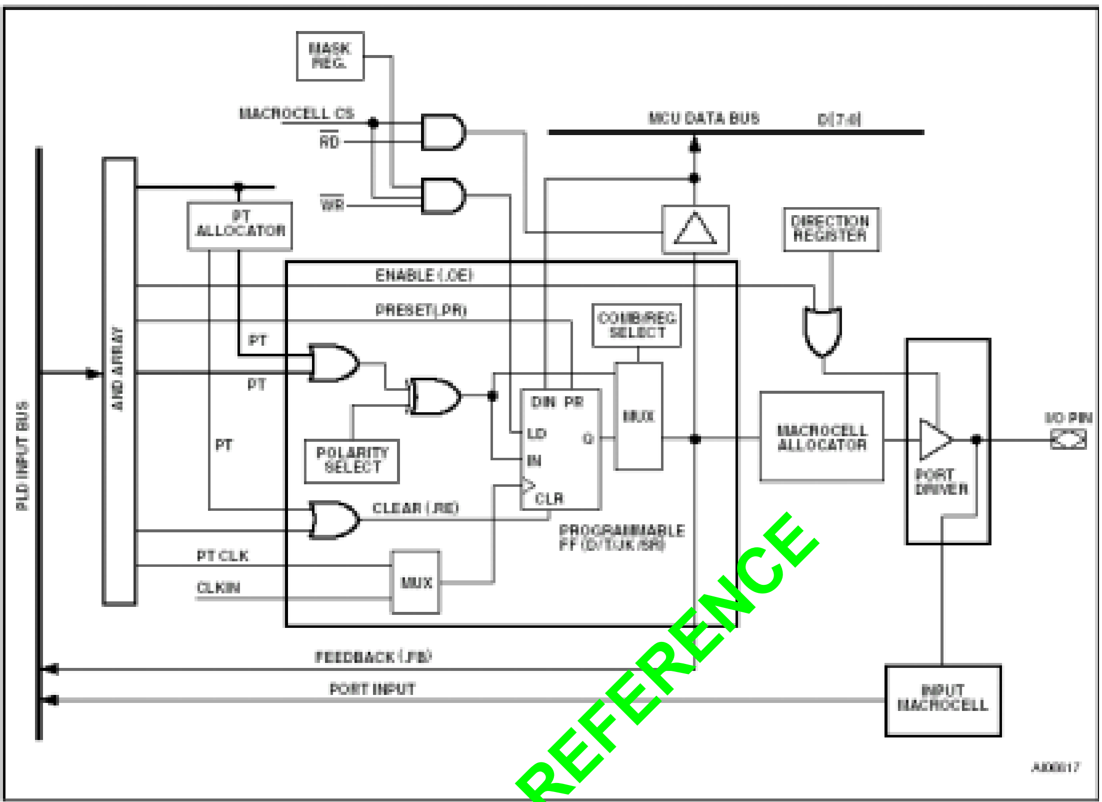
每个宏单元仅仅只能从其他确定的宏单元中借用乘积项。如果一个乘积项已被一个宏单元使用，那么对其他宏单元，该乘积项就是无效的。

如果一个方程式所要求的乘积项的数目大于可用的乘积项，那么就要求“外部”乘积项，但是，这将占用其他的输出宏单元（OMC）。如果使用“外部”乘积项，那么该方程式将会增加额外的延时。

这就叫乘积项扩展。PSDsoft按需要执行这种扩展。

装载和读输出宏单元（OMC）。输出宏单元（OMC）功能块占用被CSIOP功能块定义的MCU地址空间的一个存储器地址。（见117页，“I/O端口（PSD 模块）”一节）16个输出宏单元中每个触发器都可以通过数据总线被MCU装载。从MCU装载数据到输出宏单元优先于内部功能。同样，MCU不考虑触发器的预置，清除和时钟输入。触发器的可装载和可读取用于可装载寄存器、移位寄存器、邮箱和握手协议等应用中。输出宏单元的数据可以在写门控信号的边缘装载或写门控信号有效期内装载。装载模式在PSDsoft内指定。

图55. CPLD输出宏单元



输出宏单元（OMC）屏蔽寄存器。每组8输出宏单元（OMC）都有一个屏蔽寄存器。屏蔽寄存器可用于防止对某一输出宏单元装载数据。屏蔽寄存器的缺省值为00h，允许装载输出宏单元。假定屏蔽寄存器的某一位置1，则MCU将不能写数到相关的输出宏单元（OMC）。例如，假设McellAB0～McellAB3用于状态机，此时，你就不希望MCU写数据到McellAB覆盖状态机寄存器。因此，你就会为McellAB的屏蔽寄存器载入数值0Fh。

OMC的输出允许。输出宏单元功能块可以连接到一个I/O端口，作PLD输出。每一个端口引脚输出驱动器的输出使能被来自AND阵列的单个乘积项控制，并和方向寄存器输出相或（OR）。如果没有定义输出使能方程式，同时PSDsoft也没有将该引脚声明为PLD输出脚，那么引脚在上电时即被使能。

如果输出宏单元在PSDsoft文件内被声明为一个内部节点而不是一个端口引脚输出，那么端口引脚可被用作其他I/O功能。内部节点反馈可以作为输入发送给AND阵列。

输入宏单元（IMC）

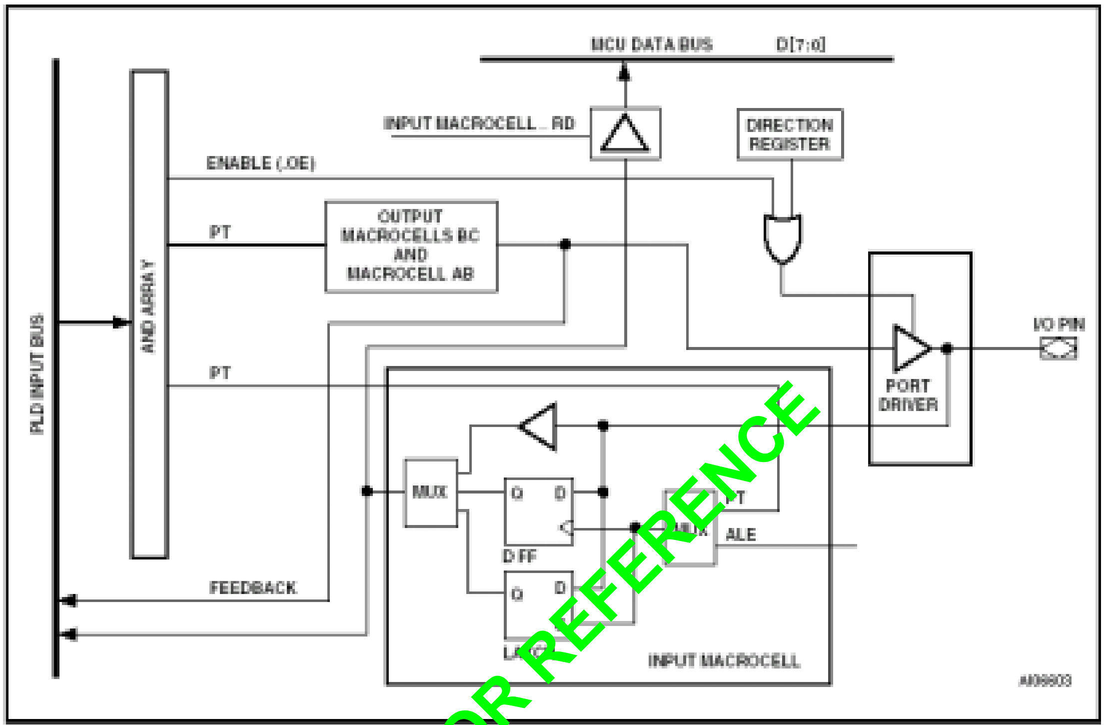
CPLD有20个输入宏单元，每一个对应端口A、B、C上的一个引脚。图56给出了输入宏单元的结构图。输入宏单元被单独配置，在把输入宏单元的信号驱动至PLD输入总线之前，先将输入宏单元作为锁存器、寄存器或传送输入端口的信号。

MCU可以通过内部数据总线读取输入宏单元的输出信号。

锁存的使能和寄存器的时钟被多路转换器驱动，多路转换器的输入是一个来自CPLD AND阵列的乘积项或者是MCU地址选通（ALE）。每个乘积项输出用于锁存／定时4个输入宏单元（IMC）。端口输入3～0可被一个乘积项控制，7～4被另一个控制。输入宏单元通过PSDsoft写入方程式来指定（见应用笔记

AN1171）。MCU可以通过IMC缓冲器来读取输入宏单元的输出。详见117页，“I/O端口（PSD模块）”一节。

图56. 输入宏单元



I/O端口（PSD模块）

PSD模块有4个可编程I/O端口：A、B、C、D。除了端口D是3位以外，其他端口都是8位。每个端口引脚都可以单独配置，因此每个端口都允许多个功能。端口可以用PSDsoft来配置，也可以由MCU写CSIOP空间的片内寄存器来配置。52脚封装器件没有端口A。这一节要讨论：

- 通用端口的结构
- 端口操作模式
- 端口配置寄存器（PCR）
- 端口数据寄存器
- 各个端口功能

通用端口结构

图57给出了通用I/O端口功能块的结构。图59到62给出了各个端口的结构。一旦一个端口引脚被定一为某种用途就不能再用作其他用途。除非被著名可作他用。

如图57所示，端口包括一个输出多路转换器，多路转换器的选择信号由控制寄存器的配置位（仅仅端口A和B如此）和PSDsoft配置来驱动。多路转换器的输入信号包括：

- 数据输出寄存器的输出数据
- 被锁存的地址输出

- CPLD宏单元输出
- 来自CPLD的外部芯片选择（ECS1～ECS2）

端口数据缓冲器（PDB）是一个三态缓冲器，在一次只能读取一个数据源。端口数据缓冲器（PDB）被连接到内部数据总线可以被MCU读取。数据输出、宏单元输出、方向和控制寄存器、以及端口引脚输入都连接到端口数据缓冲器（PDB）。

图57. 通用I/O端口结构

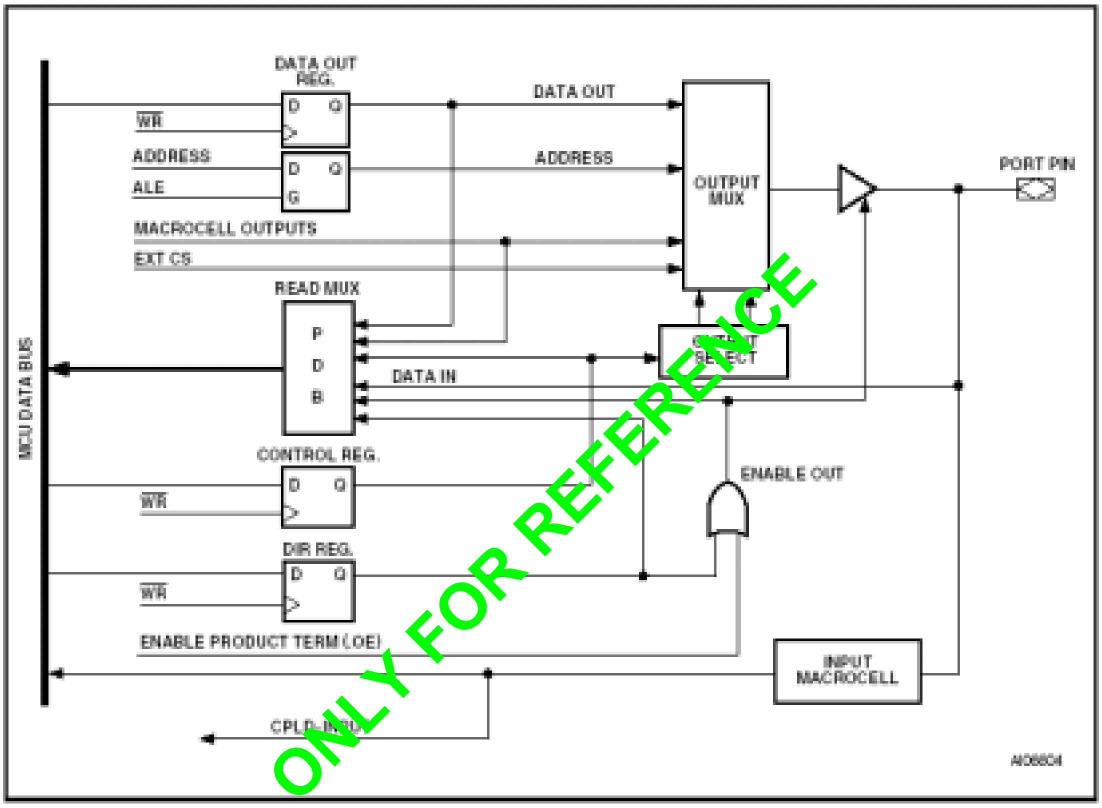


表90. 端口操作模式

Port Mode	Port A [2]	Port B	Port C	Port D
MCU I/O	Yes	Yes	Yes	Yes
PLD I/O McellAB Outputs McellBC Outputs Additional Ext. CS Outputs PLD Inputs	 Yes No No Yes	 Yes Yes No Yes	 No Yes No Yes	 No No Yes Yes
Address Out	Yes (A7 – 0)	Yes (A7 – 0)	No	No
Peripheral I/O	Yes	No	No	No
JTAG ISP	No	No	Yes[1]	No

注意：1. 和其他I/O功能可以被复用。

2. 端口A在52脚封装的芯片中是无效的。

表91. 端口操作模式设置

Mode	Defined in PSDsoft	Control Register Setting	Direction Register Setting	VM Register Setting	JTAG Enable
MCU I/O	Declare pins only	0	1 = output, 0 = input (Note 2)	N/A	N/A
PLD I/O	Logic equations	N/A	(Note 2)	N/A	N/A
Address Out (Port A,B)	Declare pins only	1	1 (Note 2)	N/A	N/A
Peripheral I/O (Port A)	Logic equations (PSEL0 & 1)	N/A	N/A	PIO bit = 1	N/A
JTAG ISP (Note 3)	JTAGSEL	N/A	N/A	N/A	JTAG_Enable

注意：1. N/A = 无用。
2. 端口A,B,C,D引脚的方向被方向寄存器与各自的来自CPLD AND阵列的输出使能乘积项相或来控制。
3. 三个方法中的任意一种都可以使能端口C的JTAG引脚。

表92. I/O端口被锁存的地址输出分配

Port A (PA3-PA0)	Port A (PA7-PA4)	Port B (PB3-PB0)	Port B (PB7-PB4)
Address a3-a0	Address a7-a4	Address a3-a0	Address a7-a4

端口引脚的三态输出驱动器使能受两个输入OR门控制，两个输入OR门的输入来自CPLD AND阵列使能乘积项和方向寄存器。如果任何阵列输出的使能乘积项没有定义，而且PSDsoft里，端口引脚没有定义为CPLD输出，那么方向寄存器单独控制驱动端口引脚的缓冲器。这些寄存器的内容可以被MCU改变。端口数据缓冲器（PDB）的反馈回路允许MCU检测这些寄存器的内容。

端口A、B、C有输入宏单元。输入宏单元可以配置成锁存器、寄存器、或对PLD直接输入。锁存器和寄存器用地址选通（ALE）或一个来自PLD AND阵列的乘积项来做时钟。

输入宏单元的输出驱动PLD输入总线，同时也能被MCU读取。见116页“输入宏单元”一节。

端口操作模式

I/O端口有几种操作模式。一些模式可以用PSDsoft来定义，一些模式可以通过MCU写CSIOP空间的控制寄存器来定义，一些模式可以用两者来定义。只能用PSDsoft定义的模式必须编程到器件内，除非器件被再次编程，否则模式不能改变。能够被MCU修改的模式可以在运行时动态地改变。PLD I/O、数据端口、地址输入、和外围I/O模式只能在编程器件以前定义。而所有的其他模式都可以在运行时由MCU来改变。更多细节见应用笔记AN1171。

表90概述了每个端口上哪些模式有效。表93指出了如何、在什么地方配置不同的模式。下面将描述每种端口操作模式。

MCU I/O 模式

在MCU I/O模式，MCU用I/O端口功能块去扩展自己的I/O端口。通过设置CSIOP空间，PSD模块的端口被映射到MCU地址空间。端口地址列在表82中。

通过将0写入控制寄存器的相应位，端口引脚可以被设置成MCU I/O模式。MCU I/O方向可以通过写

方向寄存器的相应位来改变，也可以通过输出使能乘积项来改变。见119页“外围I/O模式”一节。

当引脚被配置成输出脚，数据输出寄存器的内容驱动该脚。当配置为输入脚，MCU可以通过缓冲器里的数据来读取端口的输入。见图57。

端口C和D没有控制寄存器，缺省为MCU I/O模式。如果在PSDsoft内将配置PLD I/O方程式写入器件，那么端口可以用作PLD I/O。

PLD I/O模式

PLD I/O模式用一个端口作CPLD的输入宏单元的一个输入，同时／或者作CPLD的输出宏单元的一个输出。通过控制信号可以使输出为三态。输出使能控制信号可以被来自PLD的一个乘积项或将相应的方向寄存器复位到0来定义。如果引脚在PSDsoft中定义为PLD输入信号，那么方向寄存器相应的位必须设置为1。PLD I/O模式在PSDsoft中通过端口引脚声明同时写一个分配PLD I/O到一个端口的方程式来定义。

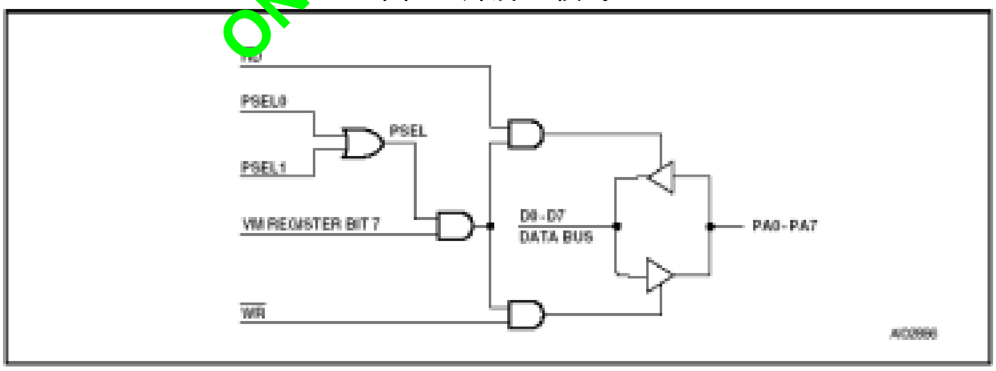
地址输出模式

地址输出模式用于将锁存的MCU地址驱动到端口引脚。依次，这些端口引脚可以驱动外部设备。要么输出使能，要么方向寄存器和控制寄存器的相应位必须设置为1。这样引脚可以用作地址输出模式。这些须MCU在运行时操作。见表92，端口A和B上，不同MCU的地址输出引脚的分配。

外设I/O模式

外设I/O模式用于和外设接口。在这个模式下，端口A每个引脚都用作MCU的三态双向数据缓冲器。通过将VM寄存器的第7位置1，外设I/O模式被使能。图58给出了，如果外设I/O模式被使能，端口A如何作为MCU数据总线的双向缓冲器。在PSDsoft中，一个PSEL0和／或PSEL1的方程式必须写入。当PSEL0或PSEL1没被激活，缓冲器是三态。

图58. 外设I/O模式



JTAG在系统编程（ISP）

端口C可出让为JTAG口，用于在系统编程（ISP）。在端口C上，你可以使JTAG操作和其它功能复用。因为在一般的操作模式下不能进行在系统编程（ISP）操作。要了解JTAG端口的更多信息，请看133页“使用JTAG串口在线编程”一节

端口配置寄存器（PCR）

每个端口都有一个端口配置寄存器，用来对端口进行配置。寄存器的内容可以由MCU按通常的读／写总线周期来访问，表82给出了访问地址。表82给出的地址是CSIOP寄存器基地址的16进制偏移量。

端口引脚是单独配置的，寄存器的每一位控制各自对应的引脚。例如，寄存器的第0位对应端口的第0位。表93给出的3个端口配置寄存器（PCR）用于端口的配置。表93中的每个寄存器的缺省上电状态都是00h。

表93. 端口配置寄存器（PCR）

Register Name	Port	MCU Access
Control	A,B	Write/Read
Direction	A,B,C,D	Write/Read
Drive Select[1]	A,B,C,D	Write/Read

注意：1. 驱动寄存器的位定义见表97。

控制寄存器。控制寄存器的任何位复位到0都将设置相应的端口引脚为MCU I/O模式，设置成1将为地址输出模式。缺省模式是MCU I/O模式。只有端口A和B有相应的控制寄存器。

方向寄存器。方向寄存器协同输出使能寄存器（端口D除外），控制I/O端口数据流的方向。寄存器的任何一位置为1都将使对应的引脚为输出脚，置为0将使对应的引脚为输入脚。所有端口的缺省模式为输入。

表94. 端口引脚方向控制，输出使能P.T没有定义

Direction Register Bit	Port Pin Mode
0	Input
1	Output

表95. 端口引脚方向控制，输出使能P.T已经被定义

Direction Register Bit	Output Enable P.T.	Port Pin Mode
0	0	Input
0	1	Output
1	0	Output
1	1	Output

表96. 端口方向分配示例

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

图59和图60给出了端口A/B和C各自的端口结构框图。端口A、B、C数据流的方向不仅被方向寄存器

控制，也被来自PLD AND阵列的输出使能乘积项所控制。如果输出使能乘积项无效，那么方向寄存器单独控制引脚的方向。

表96给出了一个配置示例：端口低三位设置为输出其余设置为输入。因为端口D只有两个引脚（如图62给出的），所以它的方向寄存器也只有两位有效。

驱动选择寄存器。驱动选择寄存器可将一些端口引脚配置成漏极开路或CMOS方式，另一些引脚配置成可控制斜升率方式。配置为漏极开路的引脚，须外接上拉电阻。如果引脚相应的驱动选择寄存器位被设置为1，那么该脚就被配置为了漏极开路方式。引脚驱动缺省为CMOS。

注意斜升率指的是一个输出的上升和下降时间。高斜升率意味着更快的输出相应，但同时也会产生更多的电噪声。当驱动寄存器相应的位被置1，一个引脚就将操作在高斜升率方式下。缺省为慢斜升率。

表97给出了端口A、B、C、D的驱动寄存器。它总结了哪些引脚可以配置成漏极开路输出，那些引脚可以配置成斜升率可控方式。

端口数据寄存器

表98给出的端口数据寄存器，被MCU用于写数据到端口或从端口读取数据。表98给出了寄存器名、各端口拥有的各类寄存器、和MCU如何访问各类寄存器。寄存器在下面描述。

表97. 驱动寄存器引脚分配

Drive Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Port A	Open Drain	Open Drain	Open Drain	Open Drain	Slew Rate	Slew Rate	Slew Rate	Slew Rate
Port B	Open Drain	Open Drain	Open Drain	Open Drain	Slew Rate	Slew Rate	Slew Rate	Slew Rate
Port C	Open Drain	Open Drain	Open Drain	Open Drain	Open Drain	Open Drain	Open Drain	Open Drain
Port D	NA[1]	NA[1]	NA[1]	NA[1]	NA[1]	Slew Rate	Slew Rate	NA[1]

注意：1. N/A = 无用。

表98. 端口数据寄存器

Register Name	Port	MCU Access
Data In	A,B,C,D	Read – input on pin
Data Out	A,B,C,D	Write/Read
Output Macrocell	A,B,C	Read – outputs of macrocells Write – loading macrocells flip-flop
Mask Macrocell	A,B,C	Write/Read – prevents loading into a given macrocell
Input Macrocell	A,B,C	Read – outputs of the Input Macrocells
Enable Out	A,B,C	Read – the output enable control of the port driver

数据输入。端口引脚直接和缓冲器里的数据连接。在MCU I/O输入模式，通过缓冲器里的数据来读取引脚的输入。

数据输出寄存器。在MCU I/O输出模式下存储被MCU所写的输出数据。如果方向寄存器或输出使能乘积项被设置为1，寄存器的内容将驱动到引脚。寄存器的内容也可以被MCU读回来。

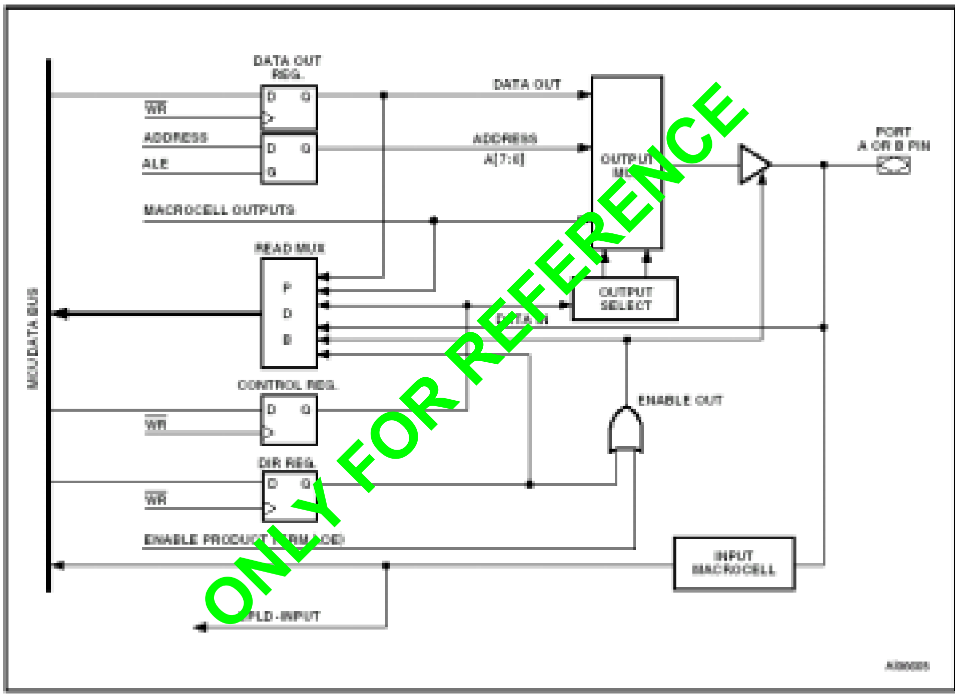
输出宏单元。CPLD输出宏单元占用MCU地址空间的一个地址。MCU可以读取输出宏单元的输出。如果OMC屏蔽寄存器的位没有设置，对宏单元写的数据将被装载进宏单元触发器。见110页，“PLD”一节。

OMC屏蔽寄存器。OMC屏蔽寄存器的每一位都对应一个输出宏单元触发器。当OMC屏蔽寄存器的位被置为1，数据将不能载入输出宏单元的触发器。缺省值为0，数据可以载入触发器。

输入宏单元（IMC）。输入宏单元（IMC）用于锁存或存储外部输入。输入宏单元的输出被引至PLD输入总线，同时可被MCU读取。见110页“PLD”一节。

使能输出。使能输出寄存器可以被MCU读取。它包含给定端口的输出使能值。1表示驱动器在输出模式，0表示驱动器在三态引脚在输入模式。

图59. 端口A和B结构

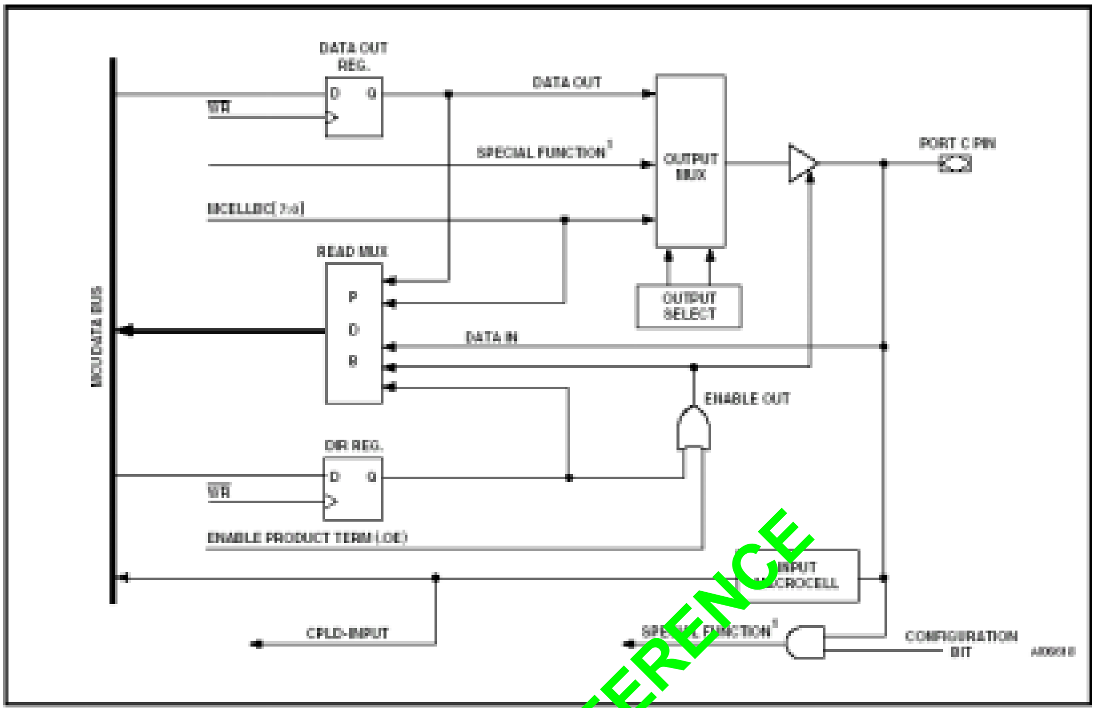


端口A和B——功能和结构

端口A和B有相似的功能和结构，如图59给出的。两个端口可以被配置，完成以下的一个或多个功能：

- MCU I/O模式
- CPLD输出——宏单元McellAB7～McellAB0可以被连到端口A或B。McellBC7～McellBC0可以被连到端口B或C。
- CPLD输入——通过输入宏单元（IMC）
- 锁存地址输出——如表92的每一项，提供锁存地址输出。
- 漏极开路／斜升率——引脚PA3~PA0和PB3~PB0可以配置成快斜升率。引脚PA7~PA4和PB7~PB4可以配置成漏极开路模式。
- 外设模式——只有端口A可以（80脚封装）

图60. 端口C的结构



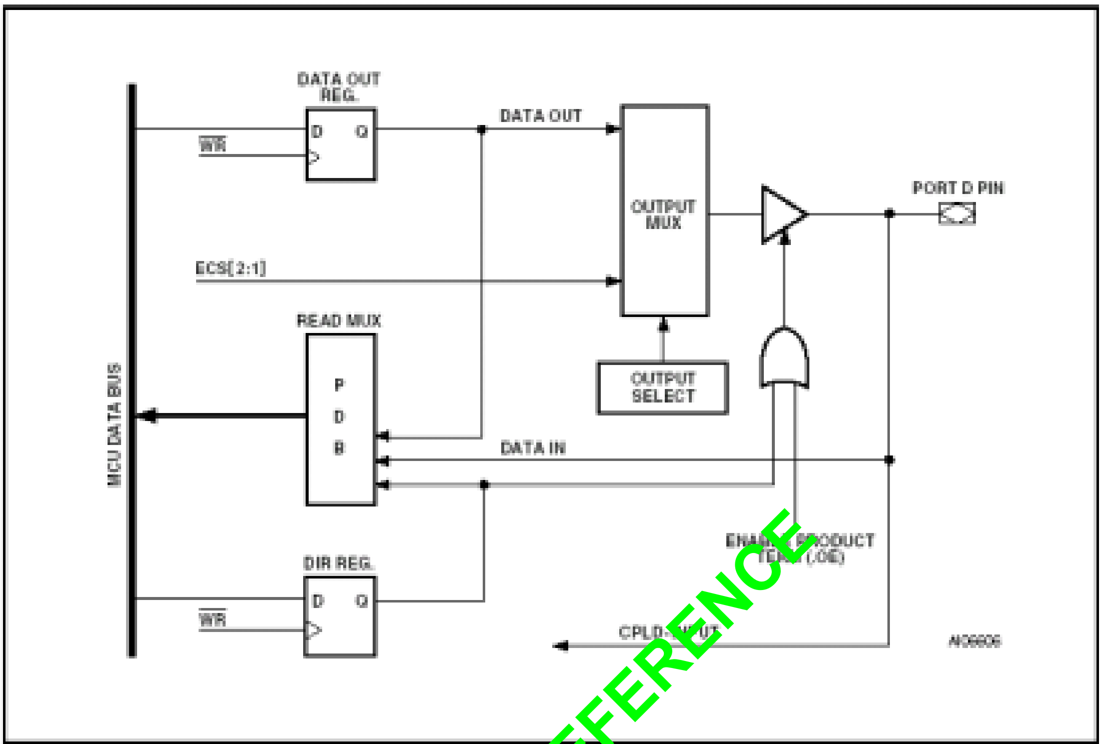
端口C——功能和结构

端口可以被配置，完成以下的一个或多个功能（见图60）：

- MCU I/O模式
- CPLD输出——McellBC7～McellBC0输出可以被连到端口B或C。
- CPLD输入——通过输入宏单元（IMC）
- 在系统编程（ISP）——为了编程／擦除PSD模块器件，JTAG口可以被使能（需要更多的JTAG编程信息，请参考133页“使用JTAG串行接口在线编程”一节）
- 漏极开路——端口C引脚可以被配置成漏极开路模式
- 电池备用特点——PC2可以被配置为电池输入通电，保持电压（VSTBY）。
  PC4可以配置为电池接通指示（VBATON），指示何时VCC低于VBAT。

端口C不支持地址输出模式，因此不需要控制寄存器。

图61. 端口D的结构



端口D——功能和结构

端口D有两个I/O引脚（52脚封装的器件端口D只有一个引脚PD1）。见图61和图62。这个端口不支持地址输出模式，因此不需要控制寄存器。端口D寄存器的8位只有1和2位用于配置引脚PD1和PD2。

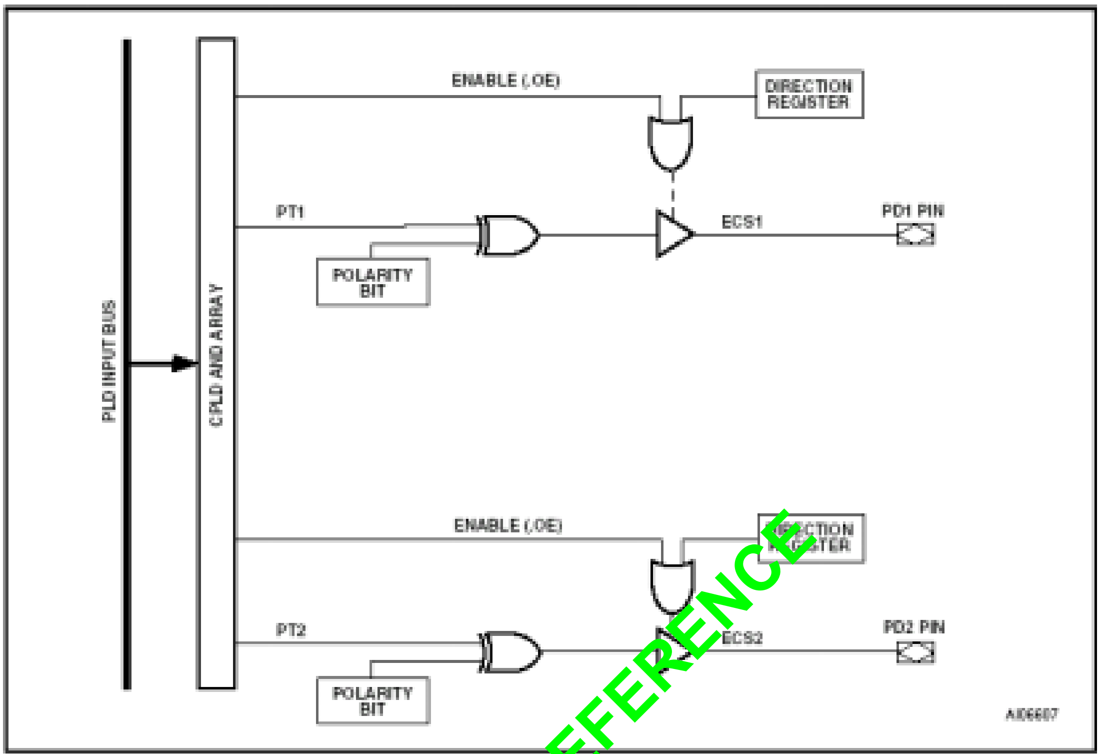
端口D可以被配置，完成以下的一个或多个功能（见图60）：

- MCU I/O模式
- CPLD输出——外部芯片选择（ECS1～ECS2）
- CPLD输入——直接输入到CPLD，不是输入宏单元（IMC）
- 斜升率——引脚可以被设置为快斜升率。端口D引脚可以由PSDsoft配置为输入脚，用于其它专有功能：
- CLKIN（PD1）作为宏单元触发器和APD计数器的输入
- PSD芯片选择输入（/CS1，CS2）。高电平禁止Flash存储器、SRAM和CSIOP。

外部芯片选择

CPLD在端口D上，也提供两个外部芯片选择（ECS1～ECS2）输出，用于选择外部器件。每一个外部芯片选择（ECS1～ECS2）包含一个乘积项，乘积项可以被配置成高（激活）或低。引脚的输出使能被输出使能乘积项或方向寄存器控制（见图62）。

图62. 端口D外部芯片选择信号



功率管理

所有的PSD模块都提供可配置的节电选项，这些选项可以单独使用，也可以组合使用，如：

■ 主、次Flash存储器和SRAM部件在构造时就考虑了功耗管理技术。另外，使用特殊的硅片设计方法，当地址／数据输入没有变化时（0直流电流），功率管理技术可以使存储器进入静态模式。一旦一个输入有变化时，就会唤醒存储器，改变和锁存它的输出，然后再返回静态模式。当输入没有改变时，设计者不必作任何工作使存储器进入静态模式，一些都是自动的。当输入没有改变时，PLD部件也能进入静态模式，如功率管理模式寄存器（PMMR）一节描述的那样。

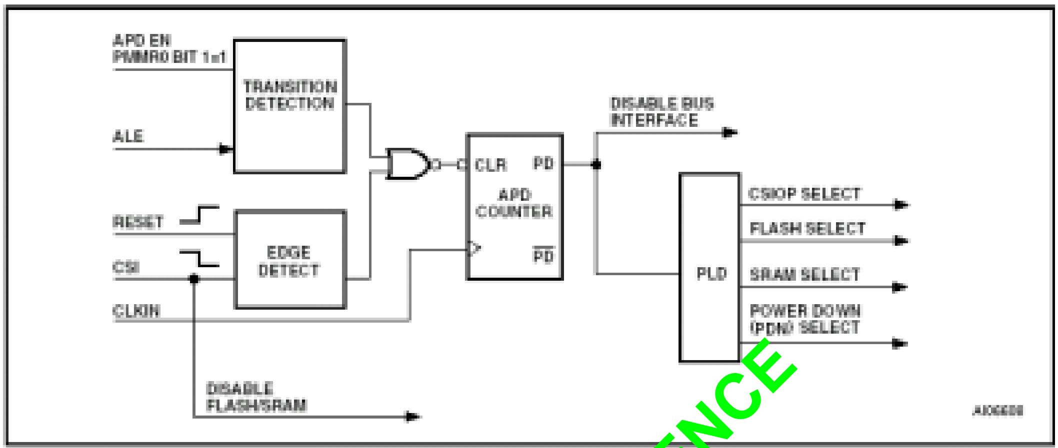
■ 在功率管理模式下，自动掉电（APD）功能块允许PSD模块自动减小静态电流。APD单元也可以阻止与存储器和PLD联系的地址／数据信号。APD单元更详细地描述在128页的“PSD模块在PMMR0中有一个加速位。设置这一位可以打开或关闭加速模式（缺省为打开加速模式）。当加速模式关闭，当PLD输入信号没有变化时（0直流电流），PLD能达到静态电流。即便输入有变化，和加速打开相比，在低频（交流电流）有效功率被节省。当加速模式打开，有了一个有效的直流成分，且交流成分要更高一些”一节。

嵌入式的逻辑监控MCU运行的地址选通。如果某一时期内没有运行（MCU 处在睡眠）。APD单元启动掉电模式（如果被使能）。一旦进入掉电模式，所有与存储器和PLD相关的地址／数据信号都会被阻止，存储器被内部取消选定。这就允许存储器和PLD保持在静态模式，即便是地址／数据信号表面上正在改变状态（噪声、MCU总线上的其他器件等等）。记住，任何PLD输入信号发生改变时，将使保持静态模式的PLD模块进入工作状态，但存储器不能这样。

■ 即便输入改变，PSD芯片选择输入（CS1，PD2）可以用来禁止内部存储器，使它们进入静态模式。这个特点不能阻止任何内部信号或禁止PLD。这是使用APD单元的一个很好的选择。当PSD芯片选择输入（CS1，PD2）最初从禁止到选通时，存储器访问时间略有延时。

■ 运行时MCU可以写PMMR寄存器来管理功率。PSD模块支持这些寄存器里的“禁止位”，这些寄存器的设置用来禁止被指定从两个PLD传来~~间~~的信号。PLD输入复杂程度直接关系到他们的电流消耗（见图66和图67）。可以通过禁止DPLD和CPLD逻辑方程式里没用的信号，来减少功耗。

图63. APD单元



PSD模块在PMMR0中有一个加速位。设置这一位可以打开或关闭加速模式（缺省为打开加速模式）。当加速模式关闭，当PLD输入信号没有变化时（0直流电流），PLD能达到静态电流。即便输入有变化，和加速打开相比，在低频（交流电流）有效功率被节省。当加速模式打开，有了一个有效的直流成分，且交流成分要更高一些。

自动掉电单元和掉电模式。APD单元通过检测地址选通（ALE）的活动使PSD模块进入掉电模式，如图63。如果APD单元被使能，当地址选通（ALD）停止活动，一个四位计数器就开始计数。如果地址选通（ALE/AS，PD0）保持无效达15个CLK IN（PD1）时钟周期，那么掉电（PDN）信号变成高电平，PSD模块进入掉电模式。

掉电模式。如果你使能APD单元，掉电模式被自动使能。如果地址选通（ALE）保持无效达15个CLK IN（PD1）时钟周期，那么器件进入掉电模式。当PSD模块在掉电模式下，记住：

■ 如果如果地址选通（ALE）脉冲又开始有了，PSD模块将返回正常操作模式。如果PSD芯片选择输入（CS1，PD2）信号为低或复位（RESET）信号输入为高，PSD模块也将返回正常操作模式。

■ 来自所有存储器和PLD的MCU地址／数据总线被阻止

■ 通过设置PMMR寄存器的适当位，可以断开送入PLD的各种信号。这些被断开的信号包括MCU控制信号和共同的CLKIN（PD1）信号。注意，断开PLD的CLKIN（PD1）不会使APD单元的CLKIN（PD1）断开。

■ 所有存储器进入静态模式后，电流都会降为静态电流。然而，PLD和I/O端口功能块不会进入静态模式，因为在逻辑和I/O输出改变之前，你不得不等待逻辑和I/O来唤醒。见表99，掉电模式对PSD模块端口的影响。

■ 典型的静态电流是微安级的。这些静态电流值是假定任何PLD输入都没有变化时。

其它节电选项。PSD模块提供其他不依靠掉电模式的节电选项。除了使SRAM进入静态模式和PSD芯片选择输入（CS1，PD2）特点外，它们可以通过设置PMMR0和PMMR2中的位来使能。

表99. 端口上掉电模式的结果

Port Function	Pin Level
MCU I/O	No Change
PLD Out	No Change
Address Out	Undefined
Peripheral I/O	Tri-State

图64. 掉电使能流程图

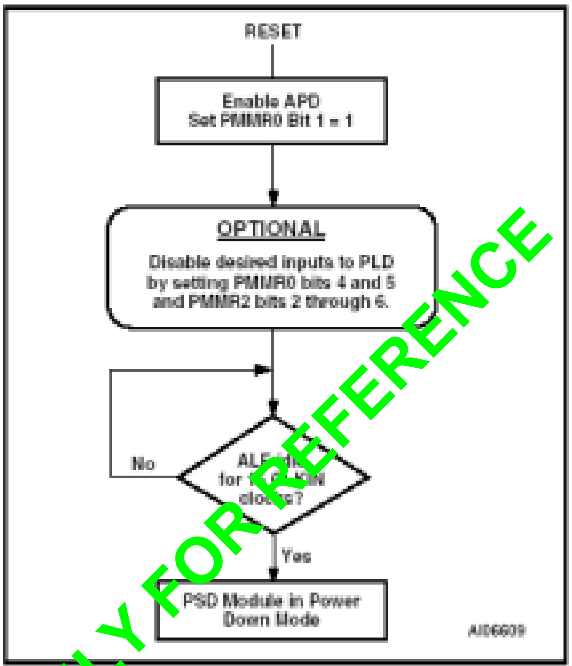


表100. 功率管理模式寄存器 PMMR01

Bit 0	X	0	Not used, and should be set to zero.
Bit 1	APD Enable	0 = off	Automatic Power-down (APD) is disabled.
		1 = on	Automatic Power-down (APD) is enabled.
Bit 2	X	0	Not used, and should be set to zero.
Bit 3	PLD Turbo	0 = on	PLD Turbo mode is on
		1 = off	PLD Turbo mode is off, saving power. μPSD3000 operates at 5 MHz below the maximum rated clock frequency
Bit 4	PLD Array clk	0 = on	CLKIN (PD1) input to the PLD AND Array is connected. Every change of CLKIN (PD1) Powers-up the PLD when Turbo bit is 0.
		1 = off	CLKIN (PD1) input to PLD AND Array is disconnected, saving power.
Bit 5	PLD MCell clk	0 = on	CLKIN (PD1) input to the PLD macrocells is connected.
		1 = off	CLKIN (PD1) input to PLD macrocells is disconnected, saving power.
Bit 6	X	0	Not used, and should be set to zero.
Bit 7	X	0	Not used, and should be set to zero.

PLD功率管理

PLD的功率和速度被PMMR0的加速位（第3位）来控制的。通过将该位置1，关闭加速模式，当输入在70ns扩展时间内不变化，PLD将消耗指定的静态电流。当输入以小于15MHZ的频率变化时，在加速位被置到1（关闭）之后，传输延时将增加10ns。当加速位被设置到0（打开），PLD以满功率全速运行。加速位影响PLD的直流功率、交流功率和传输延时。当加速模式关闭，μPSD3200输入时钟频率将从最大频率减少5MHZ。通过PMMR2寄存器的位断开MCU控制信号，可以更多地减少PLD的交流功耗。

表101. 功率管理模式寄存器PMMR21

Bit 0	X	0	Not used, and should be set to zero.
Bit 1	X	0	Not used, and should be set to zero.
Bit 2	PLD Array $\overline{WR}$	0 = on	$\overline{WR}$ input to the PLD AND Array is connected.
		1 = off	$\overline{WR}$ input to PLD AND Array is disconnected, saving power.
Bit 3	PLD Array $\overline{RD}$	0 = on	$\overline{RD}$ input to the PLD AND Array is connected.
		1 = off	$\overline{RD}$ input to PLD AND Array is disconnected, saving power.
Bit 4	PLD Array $\overline{PSEN}$	0 = on	$\overline{PSEN}$ input to the PLD AND Array is connected.
		1 = off	$\overline{PSEN}$ input to PLD AND Array is disconnected, saving power.
Bit 5	PLD Array ALE	0 = on	ALE input to the PLD AND Array is connected.
		1 = off	ALE input to PLD AND Array is disconnected, saving power.
Bit 6	X	0	Not used, and should be set to zero.
Bit 7	X	0	Not used, and should be set to zero.

注意：1. 上电后这个寄存器的位被清为0。以后的复位（Reset）脉冲不会清这个寄存器。

表102. APD计数器操作

APD Enable Bit	ALE Level	APD Counter
0	X	Not Counting
1	Pulsing	Not Counting
1	0 or 1	Counting (Generates PDN after 15 Clocks)

SRAM 备用模式（电池备用）。PSD模块支持电池备用模式，在掉电发生时保持SRAM的内容。SRAM有电压保持引脚（VSTBY，PC2）可以连接外部电池。当VCC低于VSTBY时，SRAM自动连接到电压保持脚（VSTBY，PC2），把它作为SRAM的电源。SRAM备用电流（ISTBY）典型值是0.5μA。 SRAM数据保持电压最小为2V。电池接通指示脚（VBATON）被接至PC4。此信号指示何时VCC降至VSTBY电压以下。

PSD芯片选择输入（CS1，PD2）

PSDsoft中端口D的PD2可以配置为PSD芯片选择输入（CS1）。当它为低的时候，信号选择，为了读写操作使能PSD模块的Flash存储器、SRAM、和I/O功能块。PSD芯片选择输入（CS1，PD2）为高的时候，禁止访问Flash存储器和SRAM，同时减小功耗。然而，当PSD芯片选择输入（CS1，PD2）为高时，PLD和I/O信号仍保持工作。

输入时钟

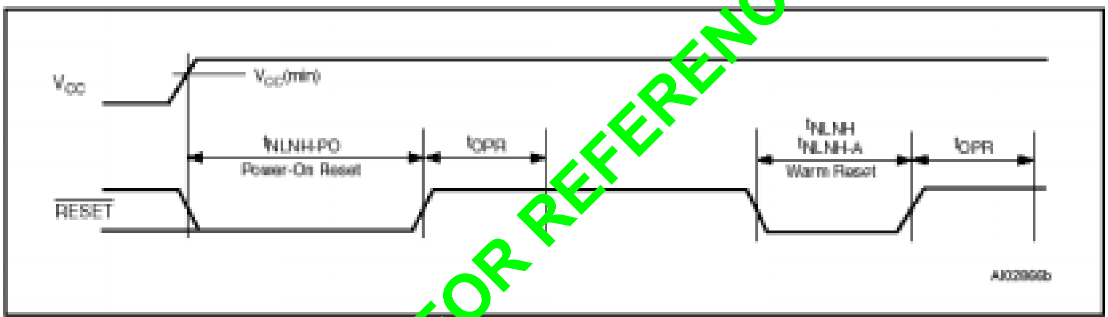
断开CLKIN（PD1）到PLD的连接可以节省交流功耗。CLKIN（PD1）是PLD AND阵列和输出宏单元（OMC） 的一个输入信号。

在掉电模式期间，或者，如果CLKIN（PD1）没有用作PLD逻辑方程式的一部分，这个时钟将被禁止，以节省交流功率。通过将PMMR0寄存器的第4或第5位置1，可以断开CLKIN（PD1）和PLD AND阵列或宏单元功能块的连接。

输入控制信号

PSD模块提供断开MCU信号（WR，RD，PSEN和地址选通（ALE））至PLD连接的选项，以节省交流功耗。这些控制信号输入到PLD AND阵列。在掉电模式期间，或者，如果它们中的任何一个没有用作PLD逻辑方程式的一部分时，这些控制信号将被禁止，以节省交流功率。通过将PMMR2寄存器的第2、3、4、5和6位置1，可以断开CLKIN（PD1）和PLD AND阵列的连接。

图65. 复位（RESET）时序



复位时序和复位时器件的状态

上电，当VCC稳定后，PSD模块要求一个持续时间为Tnlnh-po的复位脉冲。在复位期间，器件装载内部配置，清一些寄存器，设置Flash存储器为工作模式。在复位脉冲（/RESET）的上升沿后，PSD模块在第一个存储器的访问被允许之前，另外还要保持一段复位时间，Topr。

上电时Flash存储器被复位到读模式。在上电复位期间，为了最大限度地保证数据安全，消除在写选通（WR）信号的第一个脉冲沿写一个字节的可能性，扇区选择（FS0～FS3和CSBOOT0～CSBOOT1）必须全部为低，写选通（WR，CNTL0）必须为高。当VCC低于VLKO时，任何Flash存储器的写周期都被自动禁止。

热复位

如果器件已经上电并正在运行，PSD模块可以被一个窄的多的脉冲复位，脉宽Tnlnh。热复位后，在器件被操作之前，同样需要延时topr时间。图65给出了上电和热复位的时序图。

复位时I/O脚，寄存器和PLD的状态

表103给出了上电复位，热复位和掉电模式下I/O引脚，寄存器和PLD的状态。在热复位期间，PLD输出总是有效，上电复位时一旦内部配置位被装载PLD输出也开始有效。这个装载早在VCC上升到操作电平前就已经完成。一旦PLD被激活，输出状态被PLD方程式决定。

表103. 在电源上电复位，热复位和掉电模式期间的状态

Port Configuration	Power-On Reset	Warm Reset	Power-down Mode
MCU I/O	Input mode	Input mode	Unchanged
PLD Output	Valid after internal PSD configuration bits are loaded	Valid	Depends on inputs to PLD (addresses are blocked in PD mode)
Address Out	Tri-stated	Tri-stated	Not defined
Peripheral I/O	Tri-stated	Tri-stated	Tri-stated

Register	Power-On Reset	Warm Reset	Power-down Mode
PMMR0 and PMMR2	Cleared to 0	Unchanged	Unchanged
Macrocells flip-flop status	Cleared to 0 by internal Power-On Reset	Depends on .re and .pr equations	Depends on .re and .pr equations
VM Register[1]	Initialized, based on the selection in PSDsoft Configuration menu	Initialized, based on the selection in PSDsoft Configuration menu	Unchanged
All other registers	Cleared to 0	Cleared to 0	Unchanged

注意：1. VM寄存器的SR_cod和PeriphMode位总是被上电复位和热复位清0.

使用JTAG串行接口在线编程

JTAG端口映射在端口C上(见表104).所有的存储器块(主和次Flash存储器),PLD逻辑,和PSD模块配置寄存器位都可以通过JTAG串口编程.一片空白器件可以先安装在电路板上然后再通过JTAG口编程.标准的JTAG信号(IEEE 1149.1)包括TMS,TCK,TDI和TDO信号.两个额外信号TSTAR和TERR用于选择JTAG口扩展使用快速编程和快速擦除周期。在一片空白PSD模块上(出厂或擦除后)，端口C上的四个引脚作为默认的基本JTAG信号TMS,TCK,TDI和TDO.

见应用笔记AN1153,为JTAG在系统编程(ISP)提供更多细节.

标准JTAG信号

上电后，JTAG口为输入方式，等待来自外部JTAG控制器(如FlashLINK或自动测试设备)的JTAG串行命令。当接收到使能命令，TDO变成输出，JTAG通道准备好，具备全部功能。相同的使能命令可以使两个额外的JTAG信号,TSTAT和TERR使能。

／RESET输入能在JTAG编程时被激活，当PSD模块正在被编程时，／RESET信号输入到MCU，使MCU进入复位模式。见应用笔记AN1153,为JTAG在系统编程(ISP)提供更多细节。

μPSD3200器件支持JTAG在系统配置(ISC)命令,但不支持边界扫描技术。PSDsoft软件工具和FlashLINK JTAG编程电缆执行JTAG在系统配置(ISC)命令。这些JTAG在系统配置(ISC)命令的定义和顺序在ST的有效附加文档中有定义.这个文档对用FlashLINK编程μPSD3200系列的设计者来说仅仅作为参考.

表104. JTAG端口信号

Port C Pin	JTAG Signals	Description
PC0	TMS	Mode Select
PC1	TCK	Clock
PC3	$\overline{\text{TSTAT}}$	Status
PC4	$\overline{\text{TERR}}$	Error Flag
PC5	TDI	Serial Data In
PC6	TDO	Serial Data Out

JTAG扩展

在四个标准JTAG信号(TMS,TCK,TDI和TDO)之外,TSTAT和TERR是两个被标准"ISC_ENABLE"命令使能的扩展信号.通过μPDS指示信号上的状态来替代使用标准JTAG通道扫描状态可以加快编程和擦除周期.见应用笔记AN1153.

当擦除一个扇区或编程一个Flash存储器字节时,如果有错误出现,TERR会给出提示.当出错时,信号变低(激活),而且一直保持直到执行了"ISC_CLEAR"命令或在"ISC_DISABLE"命令后接收到一个芯片复位脉冲.

TSTAT操作与98页"就绪/忙(PC3)"一节中就绪/忙描述的一样.当PSD模块器件处于读模式时(主,次Flash存储器的内容都可以被读取),TSTAT为高.当Flash存储器处于编程或擦除周期时,TSTAT为低,当数据正被写入次Flash存储器时,TSTAT也为低.

TSTAT 和TERR在"ISC_ENABLE"命令期间,可以被配置为漏极开路类型.

加密和Flash存储器擦除保护

当加密位置1时,无论在器件编程器上或通过JTAG口,器件均不能被读取.当使用JTAG口时,仅允许全芯片擦除命令.所有的其他编程,擦除和校验命令都被阻止.全芯片擦除可以使器件返回没有保护的空白状态.加密位可以在PSDsoft配置里被设置.

所有的主和次Flash存储器扇区都可以单独地被扇区保护以防止擦除。扇区保护位可以在PSDsoft配置内被设置.

初始出厂状态

当器件从ST出厂,μPSD3200系列器件的存储器和PLD的所有位都被置1。可以使用编程器烧写代码、配置和PLD逻辑。

表105. JTAG使能寄存器

Bit 0	JTAG_Enable	0 = off	JTAG port is disabled.
		1 = on	JTAG port is enabled.
Bit 1	X	0	Not used, and should be set to zero.
Bit 2	X	0	Not used, and should be set to zero.
Bit 3	X	0	Not used, and should be set to zero.
Bit 4	X	0	Not used, and should be set to zero.
Bit 5	X	0	Not used, and should be set to zero.
Bit 6	X	0	Not used, and should be set to zero.
Bit 7	X	0	Not used, and should be set to zero.

注意：1. 如果JTAG信号被NVM配置位保留专用（通过PSDsotf），复位状态不会终止（或防止）JTAG操作。但是，如果JTAG使能寄存器用来使能JTAG信号，复位将防止或终止JTAG操作。

AC/DC参数

这些表描述了μPSD32××系列的AD和DC参数：

□ DC 电特性

□ AC 时序特性

■ PLD 时序

– 组合时序

– 同步时钟模式

– 异步时钟模式

– 输入宏单元时序

■ MCU 模块时序

– 读时序

– 写时序

– 掉电和复位时序

下面的内容是参数介绍：

■ 在DC特性中不同的操作模式供电电流不同.

■ 交流功率成分使PLD,Flash存储器和SRAM产生了电流（MA）与工作频率(MHZ)的关系。图66和图67示出了PLD的电流与工作频率（MA/MHZ）关系被用作乘积项（PT）的数值函数。

■ 在PLD时序中,当Turbo位为0时,参数要加必须的延时.

图66. PLD Icc与频率的关系（5V系列）

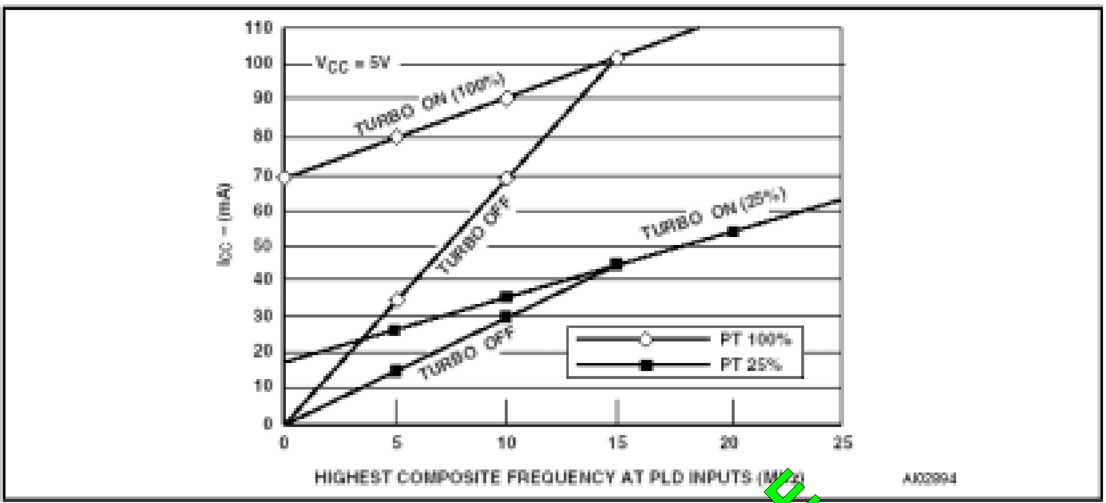


图67. PLD Icc与频率的关系（3V系列）

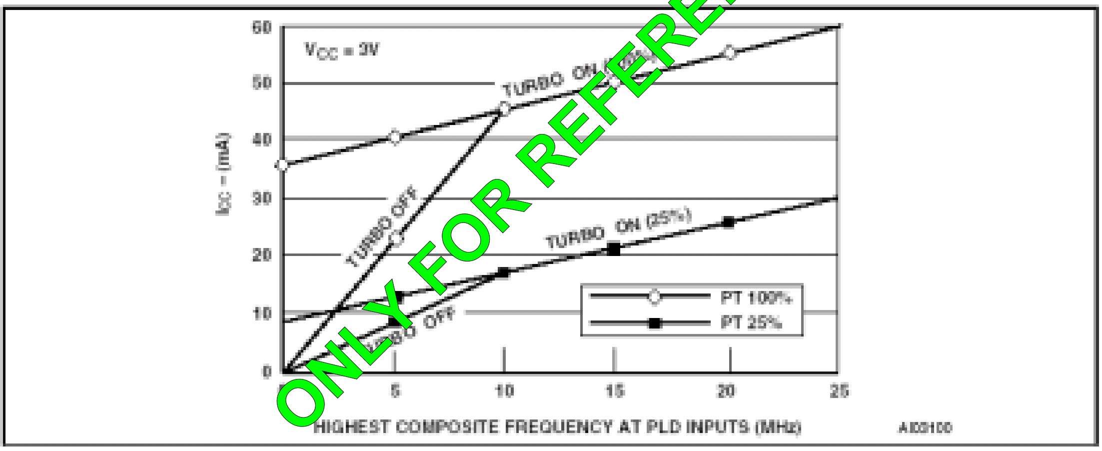


表106. Vcc=5.0V时，PSD模块典型功率计算示例（加速模式打开）

Conditions	
Highest Composite PLD input frequency	
(Freq PLD)	= 8 MHz
MCU ALE frequency (Freq ALE)	= 4 MHz
% Flash memory Access	= 80%
% SRAM access	= 15%
% I/O access	= 5% (no additional power above base)
Operational Modes	
% Normal	= 10%
% Power-down Mode	= 90%
Number of product terms used	
(from fitter report)	= 45 PT
% of total product terms	= 45/182 = 24.7%
Turbo Mode	= ON
Calculation (using typical values)	
I_{CC} total	= Ipwrdown x %pwrdown + %normal x (I_{CC} (ac) + I_{CC} (dc))
	= Ipwrdown x %pwrdown + %normal x (%flash x 2.5 mA/MHz x Freq ALE
	+ %SRAM x 1.5 mA/MHz x Freq ALE
	+ % PLD x 2 mA/MHz x Freq PLD
	+ #PT x 400 µA/PT)
	= 50 µA x 0.90 + 0.1 x (0.8 x 2.5 mA/MHz x 4 MHz
	+ 0.15 x 1.5 mA/MHz x 4 MHz
	+ 2 mA/MHz x 8 MHz
	+ 45 x 0.4 mA/PT)
	= 45 µA + 0.1 x (8 + 0.9 + 16 + 18 mA)
	= 45 µA + 0.1 x 42.9
	= 45 µA + 4.29 mA
	= 4.34 mA
This is the operating power with no EEPROM Write or Flash memory Erase cycles in progress. Calculation is based on I_{OUT} = 0 mA.	

表107. Vcc=5.0V时，PSD模块典型功率计算示例（加速模式关闭）

Conditions	
Highest Composite PLD input frequency	
(Freq PLD)	= 8 MHz
MCU ALE frequency (Freq ALE)	= 4 MHz
% Flash memory Access	= 80%
% SRAM access	= 15%
% I/O access	= 5% (no additional power above base)
Operational Modes	
% Normal	= 10%
% Power-down Mode	= 90%
Number of product terms used	
(from fitter report)	= 45 PT
% of total product terms	= 45/182 = 24.7%
Turbo Mode	= Off
Calculation (using typical values)	
I_{CC} total	= Ipwrdown x %pwrdown + %normal x (I_{CC} (ac) + I_{CC} (dc))
	= Ipwrdown x %pwrdown + % normal x (%flash x 2.5 mA/MHz x Freq ALE
	+ %SRAM x 1.5 mA/MHz x Freq ALE
	+ % PLD x (from graph using Freq PLD))
	= 50 µA x 0.90 + 0.1 x (0.8 x 2.5 mA/MHz x 4 MHz
	+ 0.15 x 1.5 mA/MHz x 4 MHz
	+ 24 mA)
	= 45 µA + 0.1 x (8 + 0.9 + 24)
	= 45 µA + 0.1 x 32.9
	= 45 µA + 3.29 mA
	= 3.34 mA
This is the operating power with no EEPROM Write or Flash memory Erase cycles in progress. Calculation is based on I_{OUT} = 0 mA.	

极限参数

强度超出所列的极限参数可能导致器件的永久性损坏。这些仅仅是极限参数，并不意味着在极限参数条件下或在任何其他超出本说明书正常运用各节所列参数的情况下，器件还能有效工作。延长在极限参数条件下的工作时间可影响器件的可靠性。也可参考STMicroelectronics SURE编程和其他相关的质量文档。

表：极限参数

表108. 极限参数

Symbol	Parameter	Min.	Max.	Unit
T_{STG}	Storage Temperature	–65	125	°C
T_{LEAD}	Lead Temperature during Soldering (20 seconds max.)[1]		235	°C
V_{IO}	Input and Output Voltage (Q = V_{OH} or Hi-Z)	–0.5	6.5	V
V_{CC}	Supply Voltage	–0.5	6.5	V
V_{PP}	Device Programmer Supply Voltage	–0.5	14.0	V
V_{ESD}	Electrostatic Discharge Voltage (Human Body model) [2]	–2000	2000	V

注意：1. IPC/JEDEC J-STD-020A

2. JEDEC Std JESD22-A114A (C1=100 pF, R1=1500, R2=500)

DC和AC参数

这一节概述了操作和测试条件,以及器件的DC和AC特性.DC和AC特性表中的参数是在相应的表中概述的测试条件下测试的结果.当要依赖被提供的参数时,设计者可以检查他们电路的操作条件是否和测试条件匹配.

表：操作条件（5V器件）

表109. 操作条件(5V器件)

Symbol	Parameter	Min.	Max.	Unit
V_{CC}	Supply Voltage	4.5	5.5	V
T_A	Ambient Operating Temperature (industrial)	–40	85	°C
	Ambient Operating Temperature (commercial)	0	70	°C

表：操作条件（3V器件）

表110. 操作条件(3V器件)

Symbol	Parameter	Min.	Max.	Unit
V_{CC}	Supply Voltage	3.0	3.6	V
T_A	Ambient Operating Temperature (industrial)	–40	85	°C
	Ambient Operating Temperature (commercial)	0	70	°C

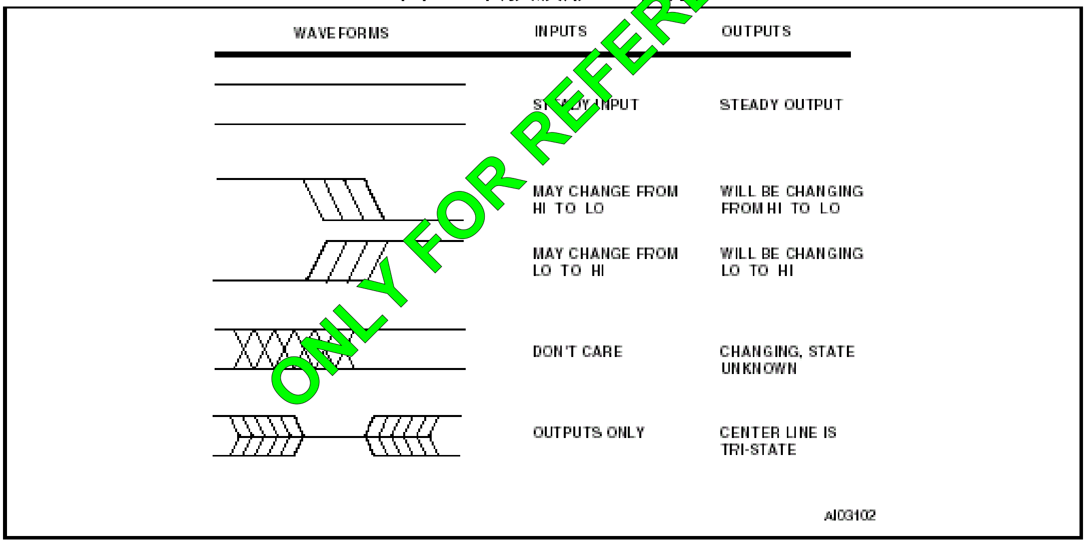
表111. 时序图中的交流符号

Signal Letters	
A	Address
C	Clock
D	Input Data
I	Instruction
L	ALE
N	Reset Input or Output
P	PSEN signal
Q	Output Data
R	RD signal
W	WR signal
B	V_{STBY} Output
M	Output Macrocell

Signal Behavior	
t	Time
L	Logic Level Low or ALE
H	Logic Level High
V	Valid
X	No Longer a Valid Logic Level
Z	Float
PW	Pulse Width

例如：tAVLX — 从地址有效到ALE无效的时间。

图68. 转换波形 – 说明



表：直流特征（5V器件）

表112. DC特性(5V器件)

Symbol	Parameter	Test Condition (in addition to those in Table 109)	Min.	Typ.	Max.	Unit
V_{IH1}	Input High Voltage (XTAL1, $\overline{RESET}$)	4.5 V < V_{CC} < 5.5 V	0.7V_{CC}		V_{CC} + 0.5	V
V_{IH}	Input High Voltage (all other inputs)	4.5 V < V_{CC} < 5.5 V	2		V_{CC} + 0.5	V
V_{IL1}	Input Low Voltage (XTAL1, $\overline{RESET}$)	4.5 V < V_{CC} < 5.5 V	–0.5		0.3V_{CC}	V
V_{IL}	Input Low Voltage (all other inputs)	4.5 V < V_{CC} < 5.5 V	–0.5		0.8	V
V_{OL}	Output Low Voltage (Ports A,B,C,D)	I_{OL} = 20 µA, V_{CC} = 4.5 V		0.01	0.1	V
		I_{OL} = 8 mA, V_{CC} = 4.5 V		0.25	0.45	V
V_{OL1}	Output Low Voltage (Ports 1,2,3,4)[1]	I_{OL} = 100 µA			0.3	V
		I_{OL} = 1.6 mA			0.45	V
		I_{OL} = 3.5 mA			1.0	V
V_{OL2}	Output Low Voltage (Port 0, ALE, PSEN, RD, WR)[1]	I_{OL} = 200 µA			0.3	V
		I_{OL} = 3.2 mA			0.45	V
		I_{OL} = 7.0 mA			1.0	V
V_{OH}	Output High Voltage (Ports A,B,C,D)	I_{OH} = –20 µA, V_{CC} = 4.5 V	4.4	4.49		V
		I_{OH} = –2 mA, V_{CC} = 4.5 V	2.4	3.9		V
V_{OH1}	Output High Voltage (Ports 1,2,3,4)	I_{OH} = –80 µA	2.4			V
		I_{OH} = –10 µA	0.9V_{CC}			V
V_{OH2}	Output High Voltage (Port 0 in ext. bus mode, ALE, PSEN, RD, WR)[2]	I_{OH} = –800 µA	2.4			V
		I_{OH} = –80 µA	0.9V_{CC}			V
V_{OH3}	Output High Voltage V_{STBYON}	I_{OH} = –1 µA	V_{STBY} – 0.8			V
I_{IL}	Logic 0 Input Current (Ports 1,2,3,4)	V_{IN} = 0.45 V	–10		–50	µA
I_{TL}	Logic 1-to-0 Transition Current (Ports 1,2,3,4)	V_{IN} = 2.0 V	–65		–650	µA
V_{LKO}	V_{CC} (min) for Flash Erase and Program		2.5		4.2	V
V_{STBY}	SRAM (PSD) Stand-by Voltage		2.0		V_{CC}	V
I_{STBY}	SRAM (PSD) Stand-by Current (VSTBY input)	V_{CC} = 0 V		0.5	1	µA
I_{IDLE}	SRAM (PSD) Idle Current (VSTBY input)	V_{CC} > V_{STBY}	–0.1		0.1	µA

Symbol	Parameter		Test Condition (in addition to those in Table 109)	Min.	Typ.	Max.	Unit
V_{DF}	SRAM (PSD) Data Retention Voltage		Only on V_{STBY}	2			V
I_{LI}	Input Leakage Current		$V_{SS} < V_{IN} < V_{CC}$	–1		1	µA
I_{LO}	Output Leakage Current		$0.45 < V_{OUT} < V_{CC}$	–10		10	µA
I_{PD}[6]	Power-down Mode		V_{CC} = 5.5 V LVD logic disabled			250	µA
			LVD logic enabled			350	µA
I_{CC_CPU}[3,4,5]	Active (12 MHz)		V_{CC} = 5 V			16	mA
	Idle (12 MHz)					7.5	mA
	Active (24 MHz)		V_{CC} = 5 V			26	mA
	Idle (24 MHz)					13.5	mA
	Active (40 MHz)		V_{CC} = 5 V			44	mA
	Idle (40 MHz)					18	mA
I_{CC_PSD} (DC)[9]	Operating Supply Current	PLD Only	PLD_TURBO = Off, f = 0 MHz [7]		0		µA/PT[8]
			PLD_TURBO = On, f = 0 MHz		400	700	µA/PT
		Flash memory	During Flash memory Write/Erase Only		15	30	mA
			Read Only, f = 0 MHz		0	0	mA
		SRAM	f = 0 MHz		0	0	mA
I_{CC_PSD} (AC)[9]	PLD AC Base			note [8]			
	Flash memory AC Adder				2.5	3.5	mA/MHz
	SRAM AC Adder				1.5	3.0	mA/MHz

注意：

1. 测试CPU的Icc（激活模式），测试条件
   XTAL1在tclch,tchcl=5ns驱动下，VIL=Vss+0.5V,VIH=Vcc-0.5V,XTAL2不接，端口0=RESET=Vcc,其他脚都不接。如果使用晶振，Icc将要稍微高一点（大约1mA）。
2. 测试CPU的Icc（空闲模式），测试条件：
   XTAL1在tclch,tchcl=5ns驱动下，VIL=Vss+0.5V,VIH=Vcc-0.5V,XTAL2不接，端口0=Vcc,RESET=Vss,其他脚都不接。
3. IPD(电源掉电模式)在EA脚被拉高的情况下测试，在XTAL1=Vss,XTAL2不接，RESET=Vss,端口0=Vcc,其他所有脚都不接的情况下测得。
4. PLD电流计算请看图66
5. I/O电流=0mA

表：直流特征（3V器件）

表113. DC特性(3V器件)

Symbol	Parameter	Test Condition (in addition to those in Table 110)	Min.	Typ.	Max.	Unit
V_{IH1}	Input High Voltage (XTAL1, RESET)	$3.0\ V < V_{CC} < 3.6\ V$	$0.7V_{CC}$		$V_{CC} + 0.5$	V
V_{IH}	Input High Voltage (all other inputs)	$3.0\ V < V_{CC} < 3.6\ V$	$0.7V_{CC}$		$V_{CC} + 0.5$	V
V_{IL1}	Input Low Voltage (XTAL1, RESET)	$3.0\ V < V_{CC} < 3.6\ V$	–0.5		$0.3V_{CC}$	V
V_{IL}	Input Low Voltage (all other inputs)	$3.0\ V < V_{CC} < 3.6\ V$	–0.5		0.8	V
V_{OL}	Output Low Voltage (Ports A,B,C,D)	$I_{OL} = 20\ \mu A$, $V_{CC} = 3.0\ V$		0.01	0.1	V
		$I_{OL} = 4\ mA$, $V_{CC} = 3.0\ V$		0.15	0.45	V
V_{OL1}	Output Low Voltage (Ports 1,2,3,4)[1]	$I_{OL} = 100\ \mu A$			0.3	V
		$I_{OL} = 1.6\ mA$			0.45	V
V_{OL2}	Output Low Voltage (Port 0, ALE, PSEN, RD, WR)[1]	$I_{OL} = 200\ \mu A$			0.3	V
		$I_{OL} = 3.2\ mA$			0.45	V
V_{OH}	Output High Voltage (Ports A,B,C,D)	$I_{OH} = -20\ \mu A$, $V_{CC} = 3.0\ V$	2.9	2.99		V
		$I_{OH} = -1\ mA$, $V_{CC} = 3.0\ V$	2.7	2.8		V
V_{OH1}	Output High Voltage (Ports 1,2,3,4)	$I_{OH} = -20\ \mu A$	2.0			V
		$I_{OH} = -10\ \mu A$	$0.9V_{CC}$			V
V_{OH2}	Output High Voltage (Port 0 in ext. bus mode, ALE, PSEN, RD, WR)[2]	$I_{OH} = -800\ \mu A$	2.0			V
		$I_{OH} = -80\ \mu A$	$0.9V_{CC}$			V
V_{OH3}	Output High Voltage VSTBYON	$I_{OH} = -1\ \mu A$	$V_{STBY} - 0.8$			V
I_{IL}	Logic 0 Input Current (Ports 1,2,3,4)	$V_{IN} = 0.45\ V$	–1		–50	µA
I_{TL}	Logic 1-to-0 Transition Current (Ports 1,2,3,4)	$V_{IN} = 2.0\ V$	–25		–250	µA
V_{LKO}	V_{CC} (min) for Flash Erase and Program		1.5		2.2	V
V_{STBY}	SRAM (PSD) Stand-by Voltage		2.0		V_{CC}	V
I_{STBY}	SRAM (PSD) Stand-by Current (VSTBY input)	$V_{CC} = 0\ V$		0.5	1	µA
I_{IDLE}	SRAM (PSD) Idle Current (VSTBY input)	$V_{CC} > V_{STBY}$	–0.1		0.1	µA
V_{DF}	SRAM (PSD) Data Retention Voltage	Only on V_{STBY}	2			V
I_{LI}	Input Leakage Current	$V_{SS} < V_{IN} < V_{CC}$	–1		1	µA

Symbol	Parameter		Test Condition (in addition to those in Table 110)	Min.	Typ.	Max.	Unit
I_{LO}	Output Leakage Current		$0.45 < V_{OUT} < V_{CC}$	–10		10	µA
I_{PD}[6]	Power-down Mode		V_{CC} = 3.6 V LVD logic disabled			110	µA
			LVD logic enabled			160	µA
I_{CC_CPU}[3,4,5]	Active (16 MHz)		V_{CC} = 3.3 V			10	mA
	Idle (16 MHz)					5.25	mA
	Active (24 MHz)		V_{CC} = 3.3 V			16	mA
	Idle (24 MHz)					8.25	mA
I_{CC_PSD} (DC)[9]	Operating Supply Current	PLD Only	PLD_TURBO = Off, f = 0 MHz [7]		0		µA/PT[8]
			PLD_TURBO = On, f = 0 MHz		200	400	µA/PT
		Flash memory	During Flash memory Write/Erase Only		10	25	mA
			Read Only, f = 0 MHz		0	0	mA
		SRAM	f = 0 MHz		0	0	mA
I_{CC_PSD} (AC)[9]	PLD AC Base		note [8]				
	Flash memory AC Adder				1.5	2.0	mA/MHz
	SRAM AC Adder				0.8	1.5	mA/MHz

注意：

1. 测试CPU的Icc（激活模式），测试条件：
   XTAL1在tclch,tchcl=5ns驱动下，VIL=Vss+0.5V,VIH=Vcc-0.5V,XTAL2不接，端口0=RESET=Vcc,其他脚都不接。如果使用晶振，Icc将要稍微高一点（大约1mA）。
2. EA脚拉低，测试CPU的Icc（空闲模式），测试条件：
   XTAL1在tclch,tchcl=5ns驱动下，VIL=Vss+0.5V,VIH=Vcc-0.5V,XTAL2不接，端口0=Vcc,RESET=Vss,其他脚都不接。
3. IPD(电源掉电模式)在EA脚被拉高的情况下测试，在XTAL1=Vss,XTAL2不接，RESET=Vss,端口0=Vcc,其他所有脚都不接的情况下测得。
4. PLD电流计算请看图66
5. I/O电流=0mA

表：外部程序存储器交流特征（带40MHz MCU模块）

表114. 外部程序存储器交流特性（带40MHZ MCU的模块）

Symbol	Parameter[1]	40 MHz Oscillator		Variable Oscillator $1/t_{CLCL}$ = 8 to 40 MHz		Unit
		Min	Max	Min	Max	
t_{LHLL}	ALE pulse width	35		$2t_{CLCL} - 15$		ns
t_{AVLL}	Address set-up to ALE	10		$t_{CLCL} - 15$		ns
t_{LLAX}	Address hold after ALE	10		$t_{CLCL} - 15$		ns
t_{LLIV}	ALE Low to valid instruction in		55		$4t_{CLCL} - 45$	ns
t_{LLPL}	ALE to $\overline{PSEN}$	10		$t_{CLCL} - 15$		ns
t_{PLPH}	$\overline{PSEN}$ pulse width	60		$3t_{CLCL} - 15$		ns
t_{PLIV}	$\overline{PSEN}$ to valid instruction in		25		$3t_{CLCL} - 50$	ns
t_{PXIX}	Input instruction hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}[2]	Input instruction float after $\overline{PSEN}$		15		$t_{CLCL} - 10$	ns
t_{PXAV}[2]	Address valid after $\overline{PSEN}$	20		$t_{CLCL} - 5$		ns
t_{AVIV}	Address to valid instruction in		65		$5t_{CLCL} - 60$	ns
t_{AZPL}	Address float to $\overline{PSEN}$	–5		–5		ns

注意：1. 条件（另外表109里的特性，Vcc=4.5到5.5V）: VSS = 0 V; CL 对Port 0, ALE 和PSEN 输出是100 pF; CL 对其他输出脚是80 pF

2. μPSD3200系列与器件的接口允许有20ns以上的浮动期，这个被限制的总线竟争不会对端口0产生任何损害。

表：外部数据存储器交流特征（带40MHz MCU模块）

表115. 外部数据存储器交流特性（带40MHZ MCU的模块）

Symbol	Parameter[1]	40 MHz Oscillator		Variable Oscillator $1/t_{CLCL}$ = 8 to 40 MHz		Unit
		Min	Max	Min	Max	
t_{RLRH}	$\overline{RD}$ pulse width	120		$6t_{CLCL} - 30$		ns
t_{WLWH}	$\overline{WR}$ pulse width	120		$6t_{CLCL} - 30$		ns
t_{LLAX2}	Address hold after ALE	10		$t_{CLCL} - 15$		ns
t_{RHDX}	$\overline{RD}$ to valid data in		75		$5t_{CLCL} - 50$	ns
t_{RHDX}	Data hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	Data float after $\overline{RD}$		38		$2t_{CLCL} - 12$	ns
t_{LLDV}	ALE to valid data in		150		$8t_{CLCL} - 50$	ns
t_{AVDV}	Address to valid data in		150		$9t_{CLCL} - 75$	ns
t_{LLWL}	ALE to $\overline{WR}$ or $\overline{RD}$	60	90	$3t_{CLCL} - 15$	$t_{CLCL} + 15$	ns
t_{AVWL}	Address valid to $\overline{WR}$ or $\overline{RD}$	70		$4t_{CLCL} - 30$		ns
t_{WHLH}	$\overline{WR}$ or $\overline{RD}$ High to ALE High	10	40	$t_{CLCL} - 15$	$t_{CLCL} + 15$	ns
t_{QVWX}	Data valid to $\overline{WR}$ transition	5		$t_{CLCL} - 20$		ns
t_{QVWH}	Data set-up before $\overline{WR}$	125		$7t_{CLCL} - 50$		ns
t_{WHQX}	Data hold after $\overline{WR}$	5		$t_{CLCL} - 20$		ns
t_{RLAZ}	Address float after $\overline{RD}$		0		0	ns

注意：1. 条件（另外表109里的特性，Vcc=4.5到5.5V）: VSS = 0 V; CL 对Port 0, ALE 和PSEN 输出是100 pF; CL 对其他输出脚是80 pF

表：外部时钟驱动（带40MHz MCU模块）

表116. 外部时钟驱动（带40MHZ MCU的模块）

Symbol	Parameter[1]	40 MHz Oscillator		Variable Oscillator $1/t_{CLCL}$ = 8 to 40 MHz		Unit
		Min	Max	Min	Max	
t_{RLRH}	Oscillator period			25	125	ns
t_{WLWH}	High time			10	$t_{CLCL} - t_{CLCX}$	ns
t_{LLAX2}	Low time			10	$t_{CLCL} - t_{CLCX}$	ns
t_{RHDX}	Rise time				10	ns
t_{RHDX}	Fall time				10	ns

注意：1. 条件（另外表109里的特性，Vcc=4.5到5.5V）: VSS = 0 V; CL 对Port 0, ALE 和PSEN 输出是100 pF; CL 对其他输出脚是80 Pf

表：外部程序存储器交流特征（带24MHz MCU模块）

表117. 外部程序存储器交流特性（带24MHZ MCU的模块）

Symbol	Parameter[1]	24 MHz Oscillator		Variable Oscillator $1/t_{CLCL}$ = 8 to 24 MHz		Unit
		Min	Max	Min	Max	
t_{LHLL}	ALE pulse width	43		$2t_{CLCL} - 40$		ns
t_{AVLL}	Address set-up to ALE	17		$t_{CLCL} - 25$		ns
t_{LLAX}	Address hold after ALE	17		$t_{CLCL} - 25$		ns
t_{LLIV}	ALE Low to valid instruction in		80		$4t_{CLCL} - 87$	ns
t_{LLPL}	ALE to $\overline{PSEN}$	22		$t_{CLCL} - 20$		ns
t_{PLPH}	$\overline{PSEN}$ pulse width	95		$3t_{CLCL} - 30$		ns
t_{PLIV}	$\overline{PSEN}$ to valid instruction in		60		$3t_{CLCL} - 65$	ns
t_{PXIX}	Input instruction hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}[2]	Input instruction float after $\overline{PSEN}$		32		$t_{CLCL} - 10$	ns
t_{PXAV}[2]	Address valid after $\overline{PSEN}$	37		$t_{CLCL} - 5$		ns
t_{AVIV}	Address to valid instruction in		148		$5t_{CLCL} - 60$	ns
t_{AZPL}	Address float to $\overline{PSEN}$	–10		–10		ns

注意：1. 条件（另外表110里的特性，Vcc=3.0到3.6V）: VSS = 0 V; 5V器件的CL 对Port 0, ALE 和PSEN 输出是100 pF;，3V器件是50PF; 5V器件的CL 对其他输出脚是80 Pf，3V器件是50PF。

2. μPSD3200系列与器件的接口允许有35ns以上的浮动期，这个被限制的总线竟争不会对端口0产生任何损害。

表：外部数据存储器交流特征（带20MHz MCU模块）

表118. 外部数据存储器交流特性（带24MHZ MCU的模块）

Symbol	Parameter[1]	24 MHz Oscillator		Variable Oscillator $1/t_{CLCL}$ = 8 to 24 MHz		Unit
		Min	Max	Min	Max	
t_{RLRH}	$\overline{RD}$ pulse width	180		$6t_{CLCL} - 70$		ns
t_{WLWH}	$\overline{WR}$ pulse width	180		$6t_{CLCL} - 70$		ns
t_{LLAX2}	Address hold after ALE	56		$2t_{CLCL} - 27$		ns
t_{RHDX}	$\overline{RD}$ to valid data in		118		$5t_{CLCL} - 90$	ns
t_{RHDX}	Data hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	Data float after $\overline{RD}$		63		$2t_{CLCL} - 20$	ns
t_{LLDV}	ALE to valid data in		200		$8t_{CLCL} - 133$	ns
t_{AVDV}	Address to valid data in		220		$9t_{CLCL} - 155$	ns
t_{LLWL}	ALE to $\overline{WR}$ or $\overline{RD}$	75	175	$3t_{CLCL} - 50$	$t_{CLCL} + 50$	ns
t_{AVWL}	Address valid to $\overline{WR}$ or $\overline{RD}$	67		$4t_{CLCL} - 97$		ns
t_{WHLH}	$\overline{WR}$ or $\overline{RD}$ High to ALE High	17	67	$t_{CLCL} - 25$	$t_{CLCL} + 25$	ns
t_{QVWX}	Data valid to $\overline{WR}$ transition	5		$t_{CLCL} - 37$		ns
t_{QVWH}	Data set-up before $\overline{WR}$	170		$7t_{CLCL} - 122$		ns
t_{WHQX}	Data hold after $\overline{WR}$	15		$t_{CLCL} - 27$		ns
t_{RLAZ}	Address float after $\overline{RD}$		0		0	ns

注意：1. 条件（另外表110里的特性，Vcc=3.0到3.6V）: VSS = 0 V; 5V器件的CL 对Port 0, ALE 和PSEN 输出是100 pF;，3V器件是50PF; 5V器件的CL 对其他输出脚是80 Pf，3V器件是50PF。

表：外部时钟驱动（带24MHz MCU模块）

表119. 外部时钟驱动（带24MHZ MCU的模块）

Symbol	Parameter[1]	24 MHz Oscillator		Variable Oscillator $1/t_{CLCL}$ = 8 to 24 MHz		Unit
		Min	Max	Min	Max	
t_{RLRH}	Oscillator period			41.7	125	ns
t_{WLWH}	High time			12	$t_{CLCL} - t_{CLCX}$	ns
t_{LLAX2}	Low time			12	$t_{CLCL} - t_{CLCX}$	ns
t_{RHDX}	Rise time				12	ns
t_{RHDX}	Fall time				12	ns

注意：1. 条件（另外表110里的特性，Vcc=3.0到3.6V）: VSS = 0 V; 5V器件的CL 对Port 0, ALE 和PSEN 输出是100 pF;，3V器件是50PF; 5V器件的CL 对其他输出脚是80 Pf，3V器件是50PF。

图69. 外部程序存储器读周期

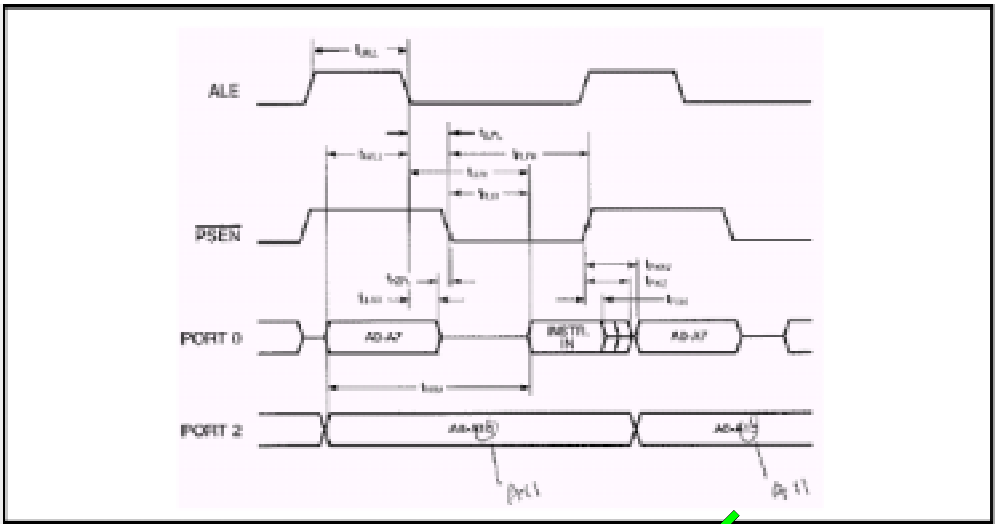


图70. 外部数据存储器读周期

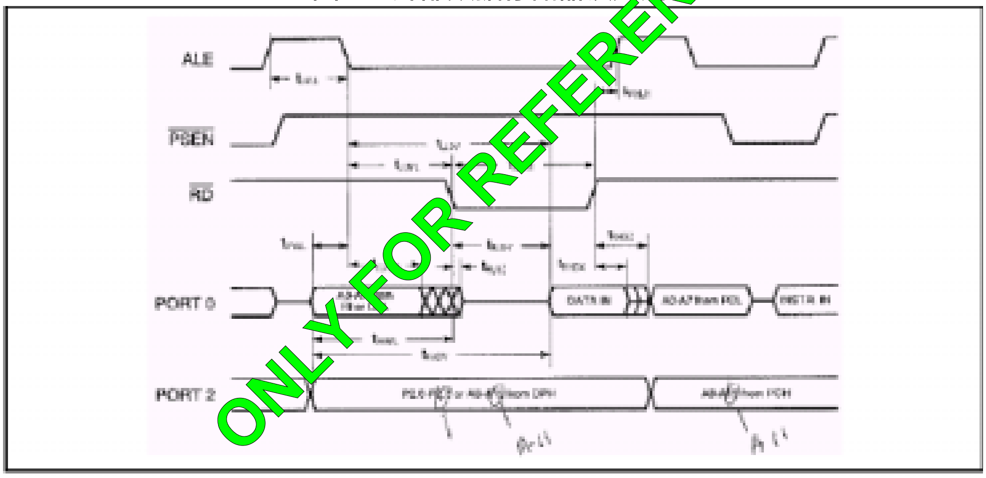
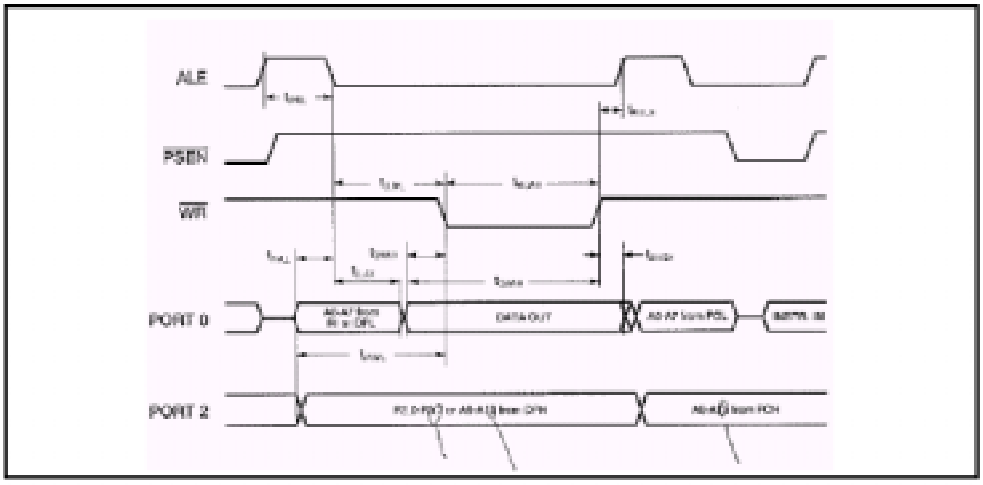


图71. 外部数据存储器写周期



表：CPLD组合时序（5V器件）

表120. CPLD组合时序（5V器件）

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Slew rate[1]	Unit
t_{PD}[2]	CPLD Input Pin/Feedback to CPLD Combinatorial Output			20	+ 2	+ 10	– 2	ns
t_{EA}	CPLD Input to CPLD Output Enable			21		+ 10	– 2	ns
t_{ER}	CPLD Input to CPLD Output Disable			21		+ 10	– 2	ns
t_{ARP}	CPLD Register Clear or Preset Delay			21		+ 10	– 2	ns
t_{ARPW}	CPLD Register Clear or Preset Pulse Width		10			+ 10		ns
t_{ARD}	CPLD Array Delay	Any macrocell		11	+ 2			ns

注意：1. 在PA3~PA0,PB3~PB0,和PD2~PD1上，都可以用快速斜升率输出。给出了所耗时间值。
2. MCU地址和控制信号的tPD指的是来自端口0，端口2， RD,WR,PSEN和ALE引脚的延时，这些引脚输出到CPLD组件的。（仅对80脚封装的器件）

表：CPLD组合时序（3V器件）

表121. CPLD组合时序（3V器件）

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Slew rate[1]	Unit
t_{PD}[2]	CPLD Input Pin/Feedback to CPLD Combinatorial Output			40	+ 4	+ 20	– 6	ns
t_{EA}	CPLD Input to CPLD Output Enable			43		+ 20	– 6	ns
t_{ER}	CPLD Input to CPLD Output Disable			43		+ 20	– 6	ns
t_{ARP}	CPLD Register Clear or Preset Delay			40		+ 20	– 6	ns
t_{ARPW}	CPLD Register Clear or Preset Pulse Width		25			+ 20		ns
t_{ARD}	CPLD Array Delay	Any macrocell		25	+ 4			ns

注意：1. 在PA3~PA0,PB3~PB0,和PD2~PD1上，都可以用快速斜升率输出。给出了所耗时间值。
2. MCU地址和控制信号的tPD指的是来自端口0，端口2， RD,WR,PSEN和ALE引脚的延时，这些引脚输出到CPLD组件的。（仅对80脚封装的器件）

表：CPLD宏单元同步时钟模式时序（5V器件）

表122. CPLD宏单元同步时钟模式时序（5V器件）

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Slew rate[1]	Unit
f_{MAX}	Maximum Frequency External Feedback	$1/(t_S+t_{CO})$		40.0				MHz
	Maximum Frequency Internal Feedback (f_{CNT})	$1/(t_S+t_{CO}-10)$		66.6				MHz
	Maximum Frequency Pipelined Data	$1/(t_{CH}+t_{CL})$		83.3				MHz
t_S	Input Setup Time		12		+ 2	+ 10		ns
t_H	Input Hold Time		0					ns
t_{CH}	Clock High Time	Clock Input	6					ns
t_{CL}	Clock Low Time	Clock Input	6					ns
t_{CO}	Clock to Output Delay	Clock Input		13			– 2	ns
t_{ARD}	CPLD Array Delay	Any macrocell		11	+ 2			ns
t_{MIN}	Minimum Clock Period[2]	$t_{CH}+t_{CL}$	12					ns

注意：1. 在PA3~PA0,PB3~PB0,和PD2~PD1上，都可以用快速斜升率输出。给出了所耗时间值。

2. CLKIN (PD1) tCLCL = tCH + tCL .

表：CPLD宏单元同步时钟模式时序（3V器件）

表123. CPLD宏单元同步时钟模式时序（3V器件）

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Slew rate[1]	Unit
f_{MAX}	Maximum Frequency External Feedback	$1/(t_S+t_{CO})$		22.2				MHz
	Maximum Frequency Internal Feedback (f_{CNT})	$1/(t_S+t_{CO}-10)$		28.5				MHz
	Maximum Frequency Pipelined Data	$1/(t_{CH}+t_{CL})$		40.0				MHz
t_S	Input Setup Time		20		+ 4	+ 20		ns
t_H	Input Hold Time		0					ns
t_{CH}	Clock High Time	Clock Input	15					ns
t_{CL}	Clock Low Time	Clock Input	10					ns
t_{CO}	Clock to Output Delay	Clock Input		25			– 6	ns
t_{ARD}	CPLD Array Delay	Any macrocell		25	+ 4			ns
t_{MIN}	Minimum Clock Period[2]	$t_{CH}+t_{CL}$	25					ns

注意：1. 在PA3~PA0,PB3~PB0,和PD2~PD1上，都可以用快速斜升率输出。给出了所耗时间值。

2. CLKIN (PD1) tCLCL = tCH + tCL .

表：CPLD宏单元异步时钟模式时序（5V器件）

表124. CPLD宏单元异步时钟模式时序（5V器件）

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Slew Rate	Unit
f_{MAXA}	Maximum Frequency External Feedback	$1/(t_{SA}+t_{COA})$		38.4				MHz
	Maximum Frequency Internal Feedback (f_{CNTA})	$1/(t_{SA}+t_{COA}-10)$		62.5				MHz
	Maximum Frequency Pipelined Data	$1/(t_{CHA}+t_{CLA})$		71.4				MHz
t_{SA}	Input Setup Time		7		+ 2	+ 10		ns
t_{HA}	Input Hold Time		8					ns
t_{CHA}	Clock Input High Time		9			+ 10		ns
t_{CLA}	Clock Input Low Time		9			+ 10		ns
t_{COA}	Clock to Output Delay			21		+ 10	– 2	ns
t_{ARDA}	CPLD Array Delay	Any macrocell		11	+ 2			ns
t_{MINA}	Minimum Clock Period	$1/f_{CNTA}$	16					ns

表：CPLD宏单元异步时钟模式时序（5V器件）

表125. CPLD宏单元异步时钟模式时序（3V器件）

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Slew Rate	Unit
f_{MAXA}	Maximum Frequency External Feedback	$1/(t_{SA}+t_{COA})$		21.7				MHz
	Maximum Frequency Internal Feedback (f_{CNTA})	$1/(t_{SA}+t_{COA}-10)$		27.8				MHz
	Maximum Frequency Pipelined Data	$1/(t_{CHA}+t_{CLA})$		33.3				MHz
t_{SA}	Input Setup Time		10		+ 4	+ 20		ns
t_{HA}	Input Hold Time		12					ns
t_{CHA}	Clock High Time		17			+ 20		ns
t_{CLA}	Clock Low Time		13			+ 20		ns
t_{COA}	Clock to Output Delay			36		+ 20	– 6	ns
t_{ARD}	CPLD Array Delay	Any macrocell		25	+ 4			ns
t_{MINA}	Minimum Clock Period	$1/f_{CNTA}$	36					ns

图72. 输入到输出禁止/使能

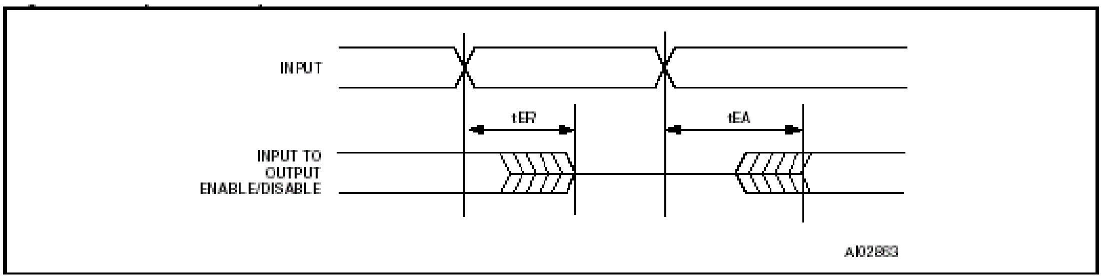


图73. 异步复位/预置

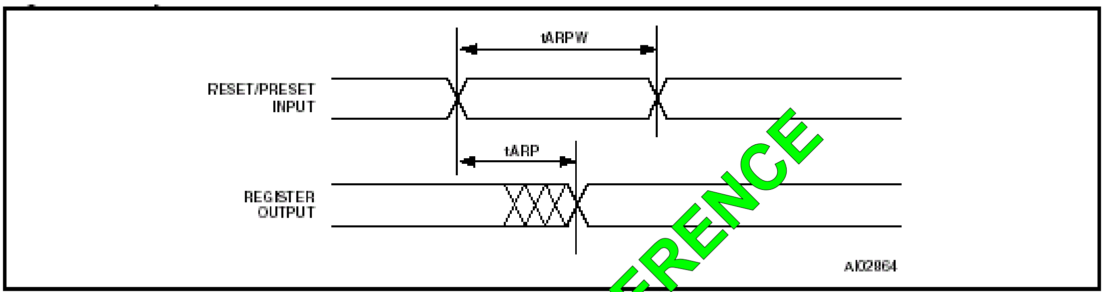


图74. 同步时钟模式时序-PLD

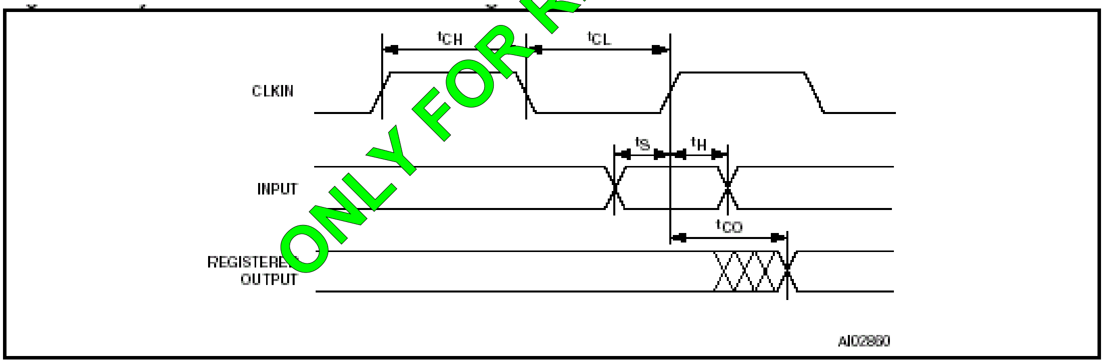
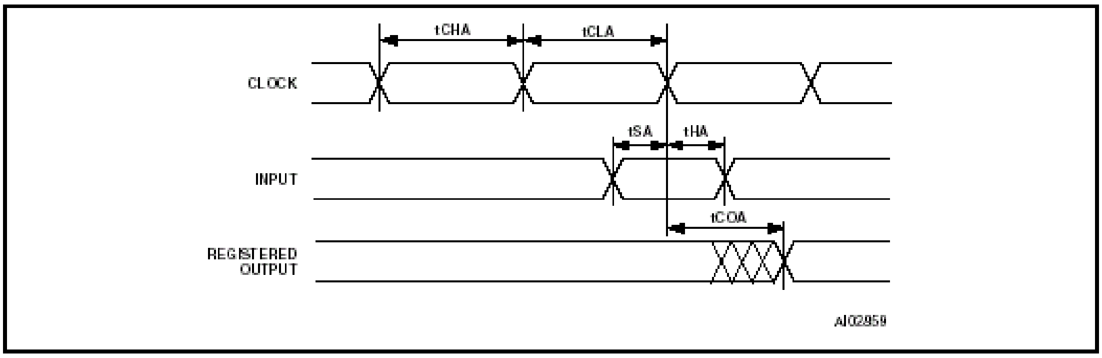


图75. 异步时钟模式时序（乘积项时钟）



表：输入宏单元时序（5V器件）

表126. 输入宏单元时序（5V器件）

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Unit
t_{IS}	Input Setup Time	(Note [1])	0				ns
t_{IH}	Input Hold Time	(Note [1])	15			+ 10	ns
t_{INH}	NIB Input High Time	(Note [1])	9				ns
t_{INL}	NIB Input Low Time	(Note [1])	9				ns
t_{INO}	NIB Input to Combinatorial Delay	(Note [1])		34	+ 2	+ 10	ns

注意：1. 关于PLD的寄存器/锁存时钟的端口A,B,C的输入。ALE/AS锁存时序参考tAVLX 和 tLXAX .

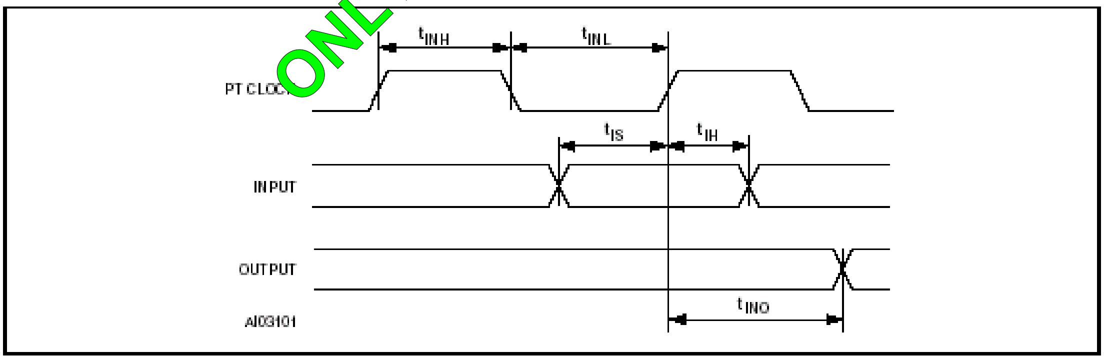
表：输入宏单元时序（3V器件）

表127. 输入宏单元时序（3V器件）

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Unit
t_{IS}	Input Setup Time	(Note [1])	0				ns
t_{IH}	Input Hold Time	(Note [1])	25			+ 20	ns
t_{INH}	NIB Input High Time	(Note [1])	12				ns
t_{INL}	NIB Input Low Time	(Note [1])	12				ns
t_{INO}	NIB Input to Combinatorial Delay	(Note [1])		46	+ 4	+ 20	ns

注意：1. 关于PLD的寄存器/锁存时钟的端口A,B,C的输入。ALE锁存时序参考tAVLX 和 tLXAX .

图76. 输入宏单元时序（乘积项时钟）



表：编程，写和擦除时序（5V器件）

表128. 编程，写和擦除时序（5V器件）

Symbol	Parameter	Min.	Typ.	Max.	Unit
	Flash Program		8.5		s
	Flash Bulk Erase[1] (pre-programmed)		3	30	s
	Flash Bulk Erase (not pre-programmed)		5		s
t_{WHQV3}	Sector Erase (pre-programmed)		1	30	s
t_{WHQV2}	Sector Erase (not pre-programmed)		2.2		s
t_{WHQV1}	Byte Program		14	1200	µs
	Program / Erase Cycles (per Sector)	100,000			cycles
t_{WHWLO}	Sector Erase Time-Out		100		µs
t_{Q7VQV}	DQ7 Valid to Output (DQ7-DQ0) Valid (Data Polling)[2]			30	ns

注意：1. 在擦除前先编程为全零

2. 在数据字节DQ0-DQ7读有效之前，轮训状态位DQ7是有效的tQ7VQV时间单元。

表：编程，写和擦除时序（3V器件）

表129. 编程，写和擦除时序（3V器件）

Symbol	Parameter	Min.	Typ.	Max.	Unit
	Flash Program		8.5		s
	Flash Bulk Erase[1] (pre-programmed)		3	30	s
	Flash Bulk Erase (not pre-programmed)		5		s
t_{WHQV3}	Sector Erase (pre-programmed)		1	30	s
t_{WHQV2}	Sector Erase (not pre-programmed)		2.2		s
t_{WHQV1}	Byte Program		14	1200	µs
	Program / Erase Cycles (per Sector)	100,000			cycles
t_{WHWLO}	Sector Erase Time-Out		100		µs
t_{Q7VQV}	DQ7 Valid to Output (DQ7-DQ0) Valid (Data Polling)[2]			30	ns

注意：1. 在擦除前先编程为全零

2. 在数据字节DQ0-DQ7读有效之前，轮训状态位DQ7是有效的tQ7VQV时间单元。

表：端口A外围设备数据模式读时序（5V器件）

表130. 端口A外围数据模式读时序（5V器件）

Symbol	Parameter	Conditions	Min	Max	Turbo Off	Unit
$t_{AVQV-PA}$	Address Valid to Data Valid	(Note [1])		37	+10	ns
$t_{SLQV-PA}$	$\overline{CSI}$ Valid to Data Valid			27	+10	ns
$t_{RLQV-PA}$	$\overline{RD}$ to Data Valid	(Note [2])		32		ns
$t_{DVQV-PA}$	Data In to Data Out Valid			22		ns
$t_{RHQZ-PA}$	$\overline{RD}$ to Data High-Z			23		ns

注意：1. 所有输入习惯于选择端口A数据外围模式。
2. 数据在端口A已经稳定

表：端口A外围设备数据模式读时序（3V器件）

表131. 端口A外围数据模式读时序（3V器件）

Symbol	Parameter	Conditions	Min	Max	Turbo Off	Unit
$t_{AVQV-PA}$	Address Valid to Data Valid	(Note [1])		50	+20	ns
$t_{SLQV-PA}$	$\overline{CSI}$ Valid to Data Valid			37	+20	ns
$t_{RLQV-PA}$	$\overline{RD}$ to Data Valid	(Note [2])		45		ns
$t_{DVQV-PA}$	Data In to Data Out Valid			38		ns
$t_{RHQZ-PA}$	$\overline{RD}$ to Data High-Z			36		ns

注意：1. 所有输入习惯于选择端口A数据外围模式。
2. 数据在端口A已经稳定

表：端口A外围设备数据模式写时序（5V器件）

表132. 端口A外围数据模式写时序（5V器件）

Symbol	Parameter	Conditions	Min	Max	Unit
$t_{WLQV-PA}$	$\overline{WR}$ to Data Propagation Delay			25	ns
$t_{DVQV-PA}$	Data to Port A Data Propagation Delay	(Note [1])		22	ns
$t_{WHQZ-PA}$	$\overline{WR}$ Invalid to Port A Tri-state			20	ns

注意：1. 数据在端口A已经稳定

表：端口A外围设备数据模式写时序（3V器件）

表133. 端口A外围数据模式写时序（3V器件）

Symbol	Parameter	Conditions	Min	Max	Unit
$t_{WLQV-PA}$	$\overline{WR}$ to Data Propagation Delay			42	ns
$t_{DVQV-PA}$	Data to Port A Data Propagation Delay	(Note [1])		38	ns
$t_{WHQZ-PA}$	$\overline{WR}$ Invalid to Port A Tri-state			33	ns

注意：1. 端口0上稳定的数据固定端口A上的数据。

图77. 外围I/O读时序

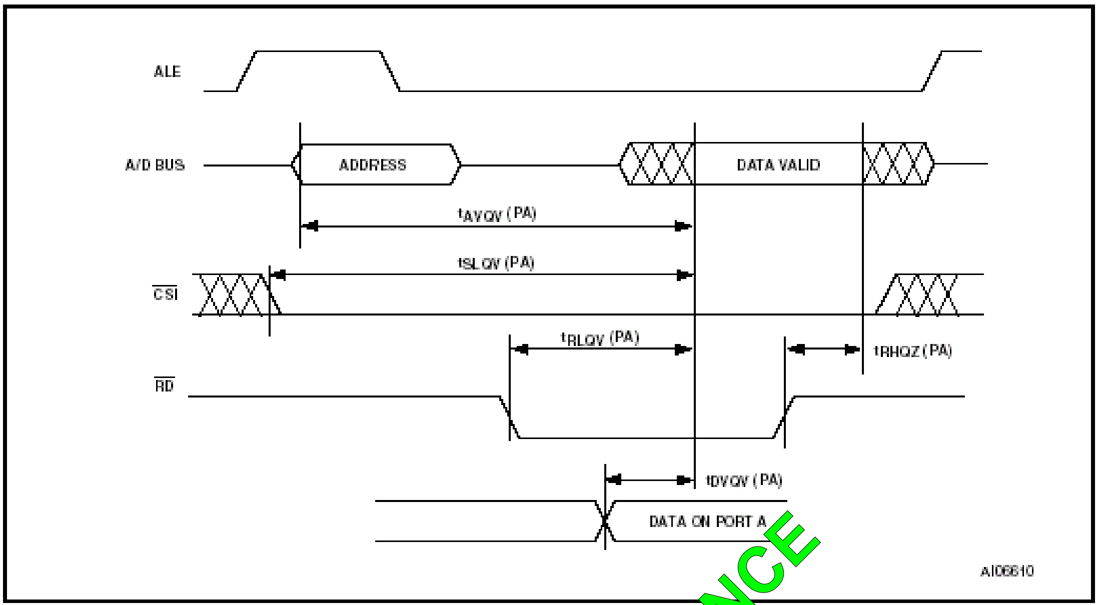


图78. 外围I/O写时序

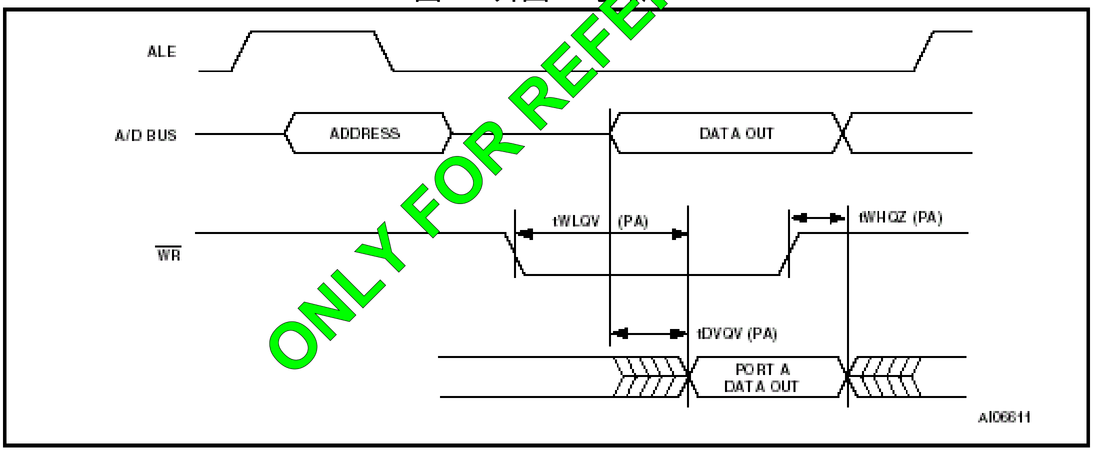
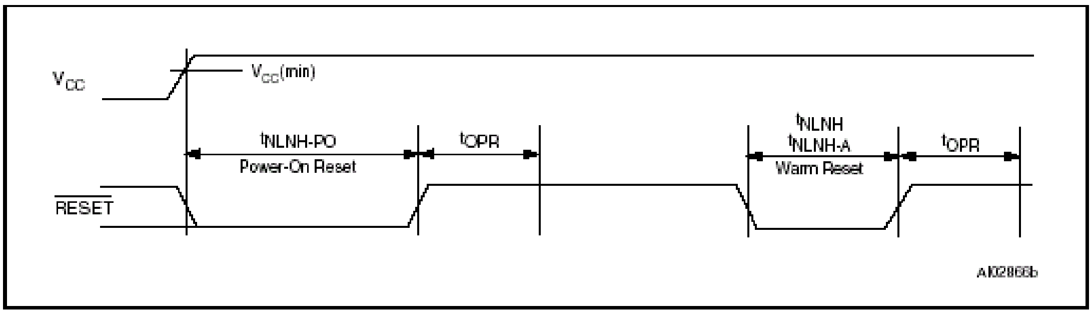


图79. 复位（RESET）时序



表：复位（Reset）时序（5V器件）

表134. 复位（Reset）时序（5V器件）

Symbol	Parameter	Conditions	Min	Max	Unit
t_{NLNH}	RESET Active Low Time [1]		150		ns
$t_{NLNH-PO}$	Power On Reset Active Low Time		1		ms
t_{NLNH-A}	Warm Reset (on the PSD834Fx) [2]		25		µs
t_{OPR}	RESET High to Operational Device			120	ns

注意：1. 复位（RESET）不会使Flash存储器的编程或擦除周期复位。
2. 热复位将中止Flash存储器编程或擦除周期，并使器件返回读模式。

表：复位（Reset）时序（3V器件）

表135. 复位（Reset）时序（3V器件）

Symbol	Parameter	Conditions	Min	Max	Unit
t_{NLNH}	RESET Active Low Time [1]		300		ns
$t_{NLNH-PO}$	Power On Reset Active Low Time		1		ms
t_{NLNH-A}	Warm Reset (on the PSD834Fx) [2]		25		µs
t_{OPR}	RESET High to Operational Device			300	ns

注意：1. 复位（RESET）不会使Flash存储器的编程或擦除周期复位。
2. 热复位将中止Flash存储器编程或擦除周期，并使器件返回读模式。

表：VSTBYON时序（5V器件）

表136. VSTBYON时序（5V器件）

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_{BVBH}	V_{STBY} Detection to V_{STBYON} Output High	(Note [1])		20		µs
t_{BXBL}	V_{STBY} Off Detection to V_{STBYON} Output Low	(Note [1])		20		µs

注意：1. VSTBYON 时序是在Vcc斜率为2ms测得的。

表：VSTBYON时序（3V器件）

表137. VSTBYON 时序（3V器件）

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_{BVBH}	V_{STBY} Detection to V_{STBYON} Output High	(Note [1])		20		µs
t_{BXBL}	V_{STBY} Off Detection to V_{STBYON} Output Low	(Note [1])		20		µs

注意：1. VSTBYON 时序是在Vcc斜率为2ms测得的。

表：ISC时序（5V器件）

表138. ISC时序（5V器件）

Symbol	Parameter	Conditions	Min	Max	Unit
t_{ISCCF}	Clock (TCK, PC1) Frequency (except for PLD)	(Note 1)		20	MHz
t_{ISCCH}	Clock (TCK, PC1) High Time (except for PLD)	(Note 1)	23		ns
t_{ISCCL}	Clock (TCK, PC1) Low Time (except for PLD)	(Note 1)	23		ns
t_{ISCCFP}	Clock (TCK, PC1) Frequency (PLD only)	(Note 2)		2	MHz
t_{ISCCHP}	Clock (TCK, PC1) High Time (PLD only)	(Note 2)	240		ns
t_{ISCCLP}	Clock (TCK, PC1) Low Time (PLD only)	(Note 2)	240		ns
t_{ISCPSU}	ISC Port Set Up Time		7		ns
t_{ISCPH}	ISC Port Hold Up Time		5		ns
t_{ISCPCO}	ISC Port Clock to Output			21	ns
t_{ISCPZV}	ISC Port High-Impedance to Valid Output			21	ns
t_{ISCPVZ}	ISC Port Valid Output to High-Impedance			21	ns

注意：1. 提供给非PLD编程，擦除或在ISC旁路模式
2. 仅仅提供给PLD编程或擦除

表：ISC时序（3V器件）

表139. ISC时序（3V器件）

Symbol	Parameter	Conditions	Min	Max	Unit
t_{ISCCF}	Clock (TCK, PC1) Frequency (except for PLD)	(Note 1)		12	MHz
t_{ISCCH}	Clock (TCK, PC1) High Time (except for PLD)	(Note 1)	40		ns
t_{ISCCL}	Clock (TCK, PC1) Low Time (except for PLD)	(Note 1)	40		ns
t_{ISCCFP}	Clock (TCK, PC1) Frequency (PLD only)	(Note 2)		2	MHz
t_{ISCCHP}	Clock (TCK, PC1) High Time (PLD only)	(Note 2)	240		ns
t_{ISCCLP}	Clock (TCK, PC1) Low Time (PLD only)	(Note 2)	240		ns
t_{ISCPSU}	ISC Port Set Up Time		12		ns
t_{ISCPH}	ISC Port Hold Up Time		5		ns
t_{ISCPCO}	ISC Port Clock to Output			30	ns
t_{ISCPZV}	ISC Port High-Impedance to Valid Output			30	ns
t_{ISCPVZ}	ISC Port Valid Output to High-Impedance			30	ns

注意：1. 提供给非PLD编程，擦除或在ISC旁路模式
2. 仅仅提供给PLD编程或擦除

图80. ISC时序

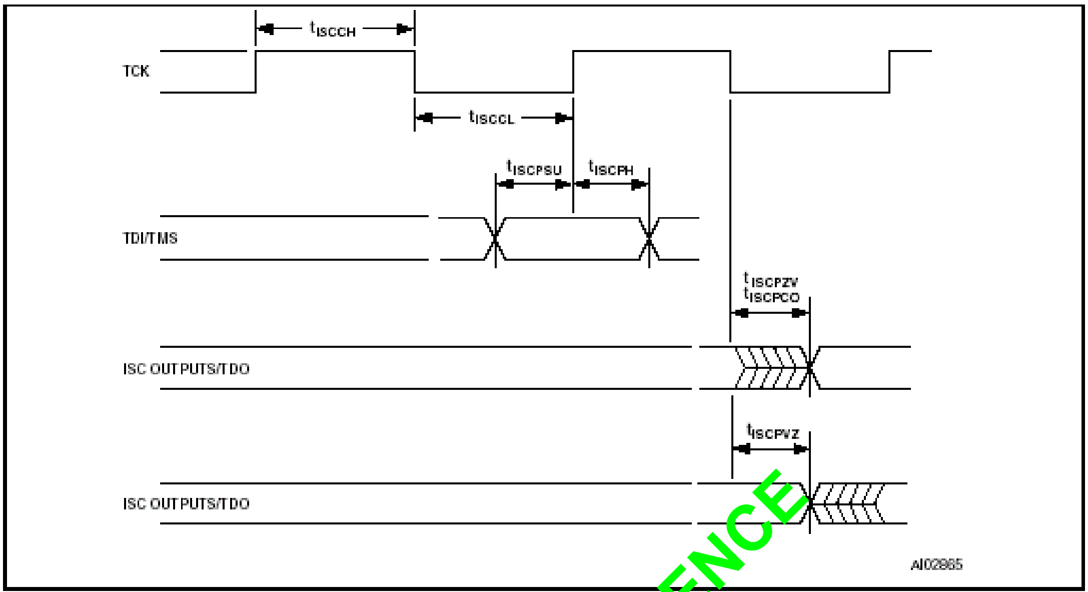
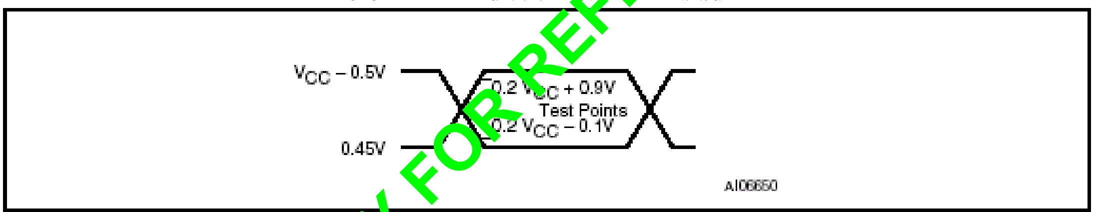
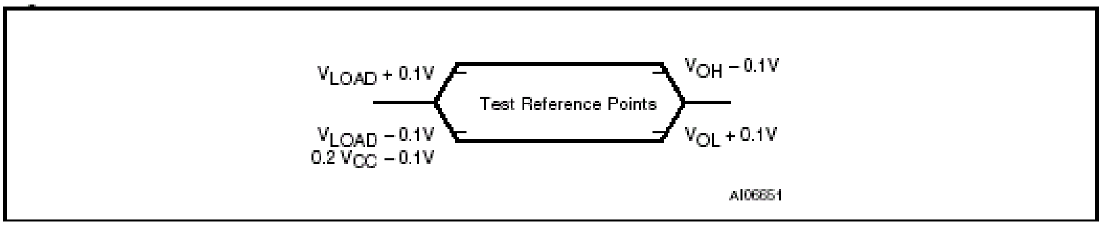


图81. MCU模块AC测量I/O波形



注意：在测试期间AC输入驱动信号Vcc-0.5V为逻辑1，0.45V为逻辑0。
时序测量VIH（最小）为逻辑1，VIL（最大）为逻辑0。

图82. PSD模块AC浮动I/O波形



注意：为了时序，当负载电压出现一个100Mv的变化，端口引脚被认为不再浮动。IOL和IL20Ma, 当被载的VOH或VOL电平出现100Mv 变化时，开始浮动。

图83. 外部时钟周期

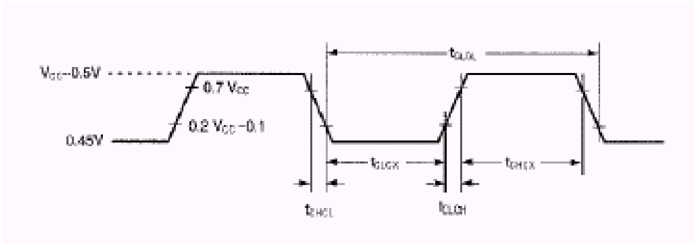
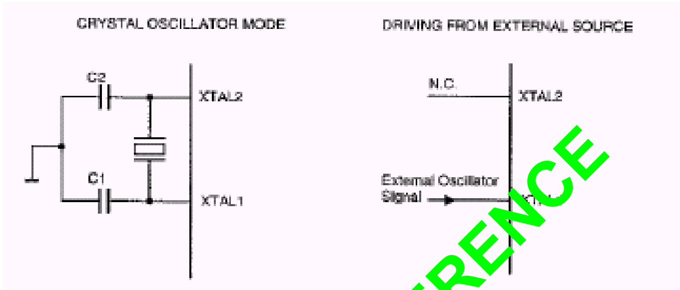


图84. 推荐的振荡电路



注意：对于晶振，C1,C2 = 30pF±10Pf。对于陶瓷谐振器，C1,C2的值请询问生产厂商。振荡电路可设计为使用陶瓷或晶体振荡器。因为每个陶瓷或晶体振荡器都有自己的特性，所以，外部器件的近似值，用户可询问厂商。

图85. PSD模块AC测量I/O波形

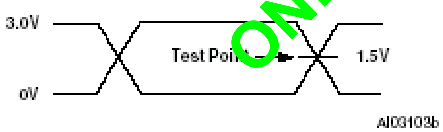
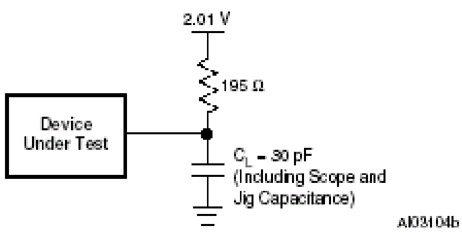


图86. PSD模块AC测量负载电路



表：电容

表140. 电容

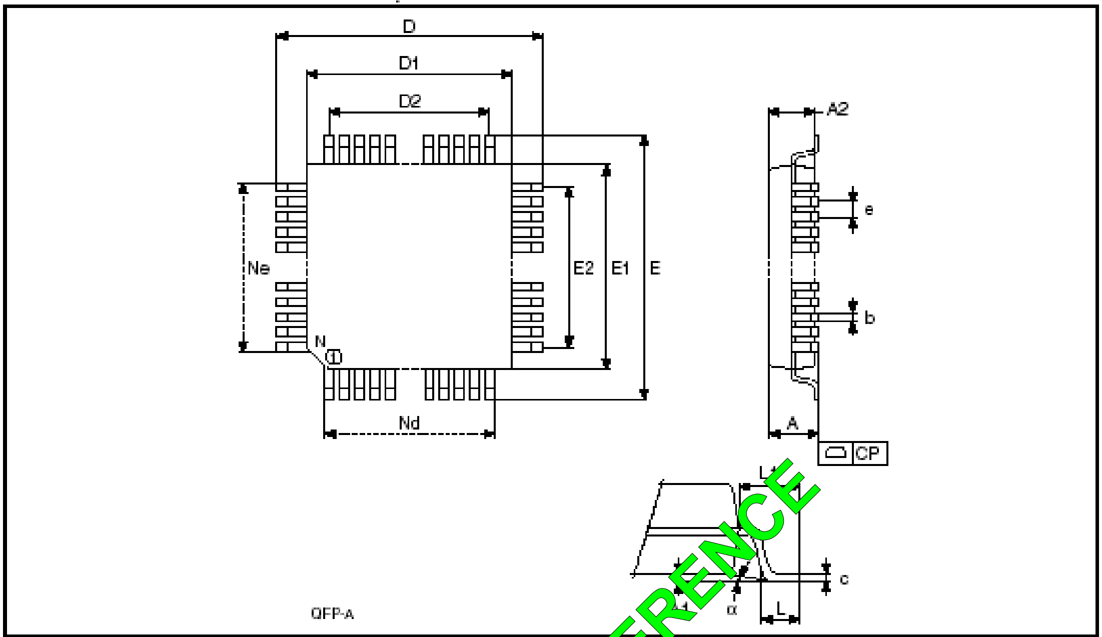
Symbol	Parameter	Test Condition	Typ.[2]	Max.	Unit
C_{IN}	Input Capacitance (for input pins)	V_{IN} = 0V	4	6	pF
C_{OUT}	Output Capacitance (for input/ output pins)	V_{OUT} = 0V	8	12	pF

注意：1. 仅仅是取样，而不是100%的测试。

2. 典型值在T_A = 25 和极小的供电电压。

机械封装

TQFP80 – 80脚PQFP封装



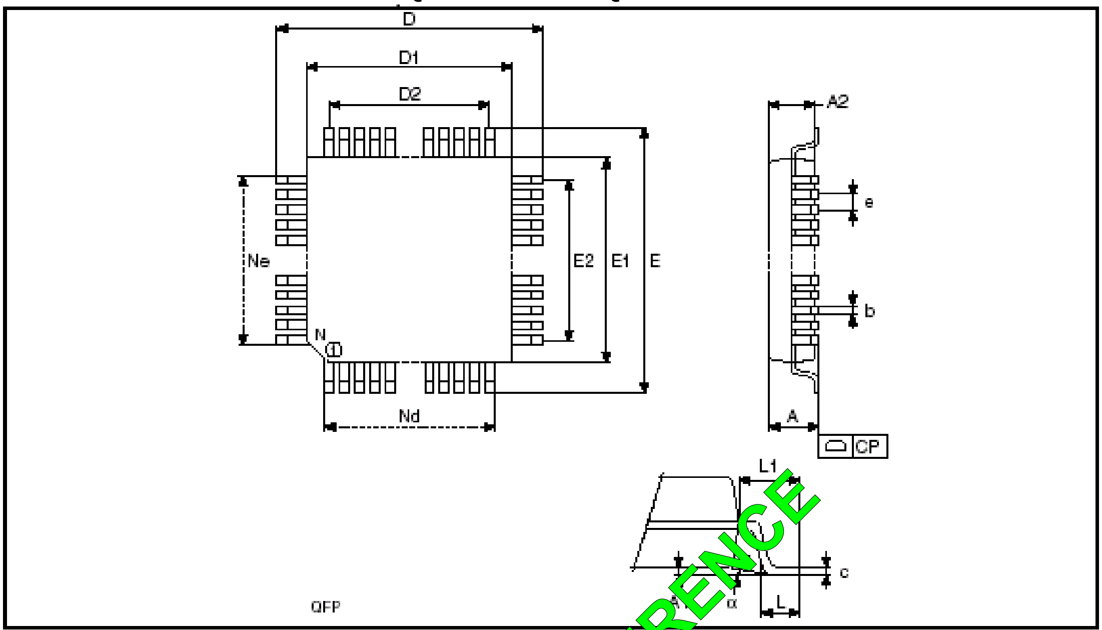
注意：上图不是按比例绘制。（下面是初步的数据）

表：TQFP80-80脚PQFP（方形塑封）

TQFP80 – 80脚PQFP封装

Symb.	mm			inches		
	Typ.	Min.	Max.	Typ.	Min.	Max.
A			1.600			0.0630
A1		0.050	0.150		0.0020	0.0059
A2	1.400	1.350	1.450	0.0551	0.0531	0.0571
α	3.5	0.0	7.0	3.5	0.0	7.0
b	0.220	0.170	0.270	0.0087	0.0067	0.0106
c		0.090	0.200		0.0035	0.0079
D	14.000	–	–	0.5512	–	–
D1	12.000	–	–	0.4724	–	–
D2	9.500	–	–	0.3740	–	–
E	14.000	–	–	0.5512	–	–
E1	12.000	–	–	0.4724	–	–
E2	9.500	–	–	0.3740	–	–
e	0.500	–	–	0.0197	–	–
L	0.600	0.450	0.750	0.0236	0.0177	0.0295
L1	1.000	–	–	0.0394	–	–
N	80			80		
Nd	20			20		
Ne	20			20		
CP			0.080			0.0031

PQFP52 – 52脚PQFP封装



注意：上图不是按比例绘制。

表：TQFP52-52脚PQFP（方形塑封）

PQFP52 – 52脚PQFP封装

Symb.	mm			inches		
	Typ.	Min.	Max.	Typ.	Min.	Max.
A			1.75			0.069
A1		0.05	0.2		0.002	0.008
A2		1.25	1.55		0.049	0.061
b		0.2	0.4		0.007	0.016
c		0.07	0.23		0.002	0.009
D	12			0.473		
D1	10			0.394		
D2						
E	12			0.473		
E1	10			0.394		
E2						
e	0.65	–	–	0.026		
L		0.45	0.75		0.018	0.03
L1	1.0	–	–	0.039		
α		0°	7°		0°	7°
N	52			52		
Nd	13			13		
Ne	13			13		
CP			0.10			0.004

表：命名规则信息表

表141. 排序信息图

Example: uPSD 3 2 3 4 B V – 24 U 6 T

Device Type
uPSD = Microcontroller PSD

Family
3 = 8032 core

PLD Size
2 = 16 Macrocells
3 = 32 Macrocells

SRAM Size
1 = 16 Kbit
3 = 64 Kbit
5 = 256 Kbit

Main Flash Memory Size
3 = 1 Mbit
4 = 2 Mbit
5 = 4 Mbit

IP Mix
A = USB, I^2C, PWM, DDC, ADC, (2) UARTs
Supervisor (Reset Out, Reset In, LVD, WD)
B = I^2C, PWM, DDC, ADC, (2) UARTs
Supervisor (Reset Out, Reset In, LVD, WD)

Operating Voltage
blank = V_{CC} = 4.5 to 5.5V
V = V_{CC} = 3.0 to 3.6V

Speed
24 = 24 MHz
40 = 40 MHz

Package
T = 52-pin TQFP
U = 80-pin TQFP

Temperature Range
1 = 0 to 70 °C (commercial)
6 = –40 to 85 °C (industrial)

Option
T = Tape & Reel Packing