

MA801

数据手册

版本 1.00

目 录

1. 概述	5
2. 功能	6
3. 引脚	7
3.1. 封装	7
3.2. 引脚定义	8
4. 方框图	9
5. 特殊功能寄存器SFR	10
5.1. SFR 映射表	10
5.2. SFR 位分配	11
6. 8051 CPU 功能描述	12
6.1. CPU 寄存器	12
6.2. CPU 时序	13
6.3. CPU 寻址方式	13
7. 存储器结构	14
7.1. 片上程序存储器	15
7.2. 片上数据存储器 RAM	16
7.3. 关于C51编译器的声明识别符	17
8. 双数据指针寄存器 (DPTR)	18
9. I/O 结构	19
9.1. IO 配置	19
9.1.1. 准双向口	19
9.1.2. 推挽输出	19
9.1.3. 仅输入(高阻输入)	20
9.1.4. 开漏集输出	20
9.2. I/O 端口寄存器	21
9.2.1. 端口 1	22
9.2.2. 端口 3	22
10. 中断	23
10.1. 中断结构	23
10.2. 中断寄存器	24
11. 定时器/计数器	27
11.1. 定时器 0 和定时器 1	27
11.1.1. 模式 0	27
11.1.2. 模式 1	28
11.1.3. 模式 2	28
11.1.4. 模式 3	29
11.1.5. 定时器时钟输出	29
11.1.6. 定时器 0/1 寄存器	30
12. 串行口 (UART)	32
12.1. 帧错误检测	32
12.2. 多处理器通讯	33
12.3. 自动地址识别	33

12.4. 串口寄存器	35
13. 可编程计数器阵列(PCA)	38
13.1. PCA 概述	38
13.2. PCA定时/计数器	38
13.3. 比较/捕获模块	40
13.4. PCA模式设置	40
13.4.1. 捕获模式	41
13.4.2. 16 位软件定时器模式	41
13.4.3. 高速输出模式	42
13.4.4. PWM 模式	42
14. 串行外设接口(SPI)	44
14.1. 典型 SPI 配置	46
14.1.1. 单主机和单从机	46
14.1.2. 双驱动器,可以是主机或从机	46
14.1.3. 单主机和多从机	47
14.2. SPI 配置	48
14.3. 从机注意事项	48
14.4. 主机注意事项	48
14.5. /SS引脚的模式改变	49
14.6. 数据冲突	49
14.7. SPI 时钟频率选择	49
14.8. 数据模式	50
15. 模数 (A/D) 转换器	52
15.1. ADC 控制寄存器	52
15.2. ADC 操作	53
15.3. ADC 注意事项	53
15.3.1. A/D 转换时间	53
15.3.2. I/O 口用于ADC 转换	54
15.3.3. 待机和掉电模式	54
15.3.4. VDD 供电要求	54
16. 看门狗 (WDT)	55
16.1. 看门狗结构	55
16.2. 看门狗寄存器	55
17. 电源控制	56
17.1. 节能模式	56
17.1.1. 空闲模式 (Idle)	56
17.1.2. 掉电模式(Power-down)	56
17.1.3. 中断唤醒	56
17.1.4. 复位唤醒	56
17.2. 电源控制寄存器	57
18. 在系统编程 (ISP)	58
19. 在应用程序编程 (IAP)	60
20. 辅助特殊功能寄存器SFR	61
21. 指令集	62
22. 最大绝对额定参数	66

23. 电气特性	67
23.1. 直流特性	67
24. 修订历史	68

1. 概述

MA801 是基于80C51的高效1-T结构的单芯片微处理器, 每条指令需要1~7个时钟信号 (比标准8051快6~7倍), 与8051指令集兼容。因此在与标准8051有同样的处理能力的情况下, MA801只需要非常低的运行速度, 同时由此能很大程度的减少耗电量。

MA801 拥有8K字节的内置Flash存储器用于保存代码和数据。Flash存储器可以通过在系统编程(ISP)。同时, 也提供在应用编程(IAP)的能力。ISP让使用者无需从产品中取下微控制器就可以下载新的代码; IAP意味着应用程序正在运行时, 微控制器能够在Flash中写入非易失数据。这些功能都由内建的电荷泵提供编程用的高压。

MA801 保留了标准80C52的基本特色: 256字节的随机存储器、15个I/O口、2个外部中断、及2个定时/计数器。一个PCA定时器, 一个SPI接口, 一个一次性使能的看门狗WDT, 一个8-位ADC, 一个低电平侦察器LVD, 一个为方便多处理器通讯和改进了速度的增强型通用串行口(EUART)。

MA801有两种节能模式, 以减少耗电量。在空闲模式下, CPU被冻结而外围模块和中断系统依然活动。在掉电模式下, 随机存储器RAM和特殊功能寄存器SFR的内容被保存, 而其他所有功能被终止。最重要的是, 在掉电模式下的微控制器可以被外部中断唤醒。

2. 功能

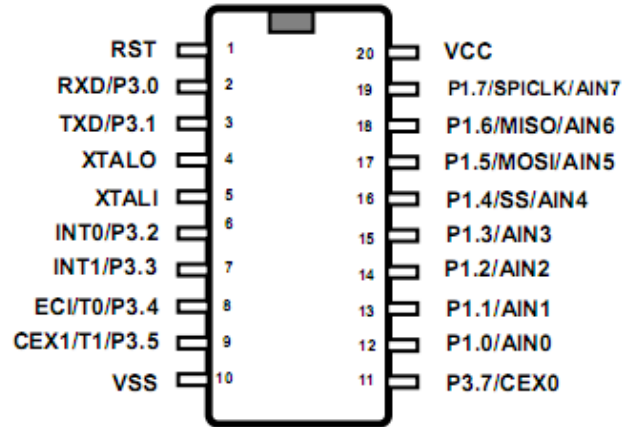
- 1-T 80C51 中央处理单元
- MA801 有 8K 字节 Flash ROM
 - 6KB 为 AP 代码程序空间
 - 1KB 为 IAP 数据 Flash 空间
 - 为 Flash 存储器存取提供代码保护
- 片上 256 字节数据 RAM
- 2 个 16 位定时/计数器: Timer 0、和 Timer 1.
- 7 个中断源, 四级优先级中断能力
- 增强型 UART,支持帧错误检测 硬件地址识别
- 15 位看门狗定时器 Watch-Dog-Timer. 8 位分频, 通过 CPU 或上电一次性使能
- SPI 接口 (主/从模式)
- 一个 PCA (可编程计数器阵列)
 - 捕捉模式
 - 16位软件定时器模式
 - 高速输出模式
 - PWM (脉冲宽度调变器) 模式
- 8 通道 8 位 ADC,
- 电源控制: 空闲模式 (idle mode) 和掉电模式 (power-down mode), 可由外部中断 INT0/INT1 唤醒
- 15 个通用输入输出
- 内建 6MHz 内部晶振
- 双数据指针 DPTR
- 2 个外部中断输入: INT0/INT1
- 内建低电压检测 LVD 以及复位电路
- 工作电压: 4.5V~5.5V
- 工作频率: 最大工作频率为 24MHz。
- 工作温度:
 - 工业级别 (-40℃ 至 +85℃)*
- 封装类型:
 - PDIP-20:MA801AE
 - SOP-20:MA801AS
 - TSSOP-20:MA801AT
- 下单型号 封装正印

型号	封装形式	正印型号
MA801AE	PDIP-20	MA801AE
MA801AS	SOP-20	MA801AS
MA801AT	TSSOP-20	MA801AT

3. 引脚

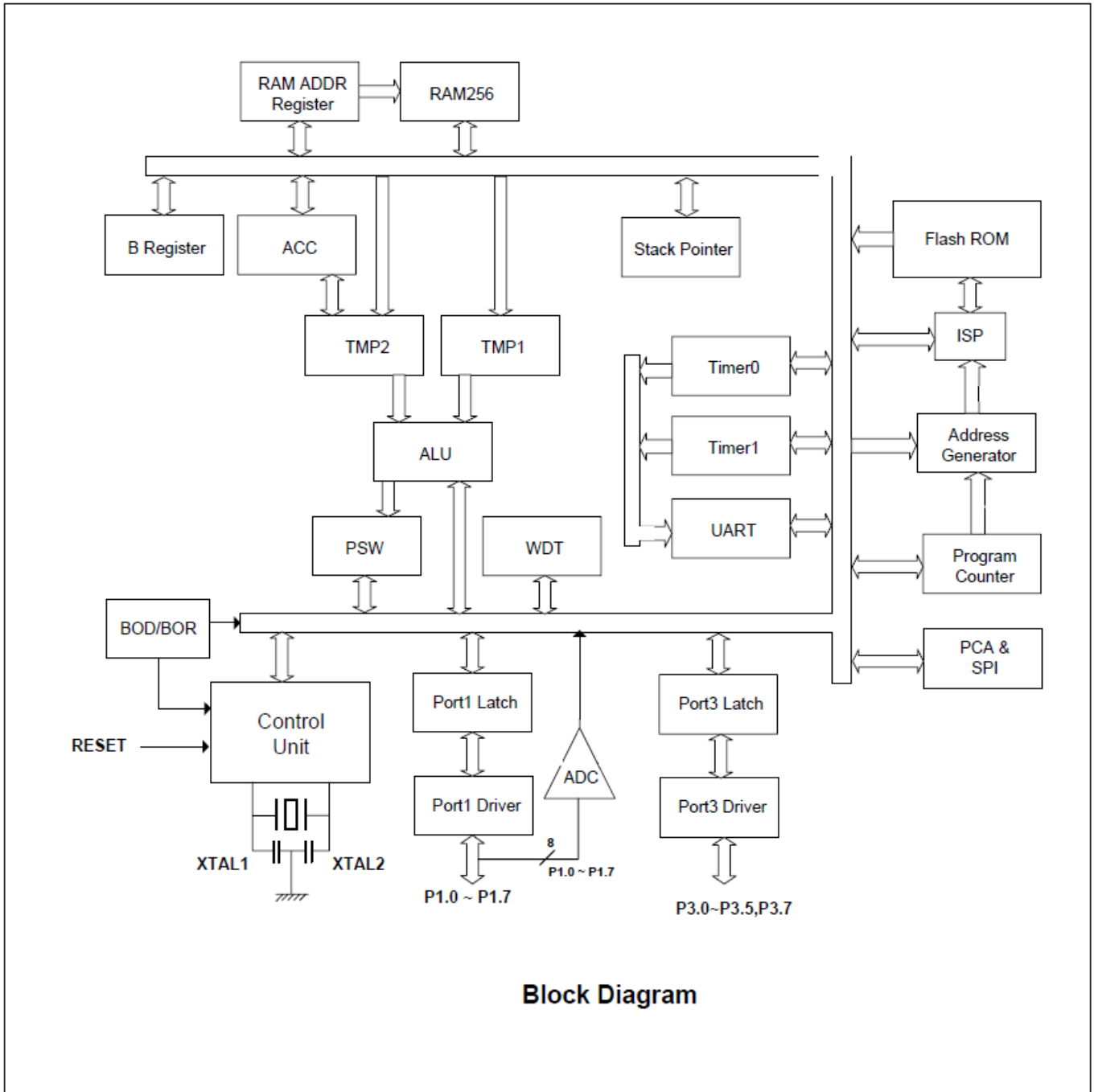
3.1. 封装

MA801 : PDIP-20/SOP-20/TSSOP-20



3.2. 引脚定义

引脚名	引脚号	类型	描述
P1.0(AIN0)	12	I/O	P1 是可编程 I/O 口 同时可作为 ADC 输入 P1 还具有以下特殊功能 SS(P1.4) SPI 接口串行模式选择 MOSI(P1.5) SPI 接口主数据输出或从数据输入 MISO(P1.6) SPI 接口主数据输入或从数据输出 SPICLK(P1.7) SPI 接口时钟线
P1.1(AIN1)	13		
P1.2(AIN2)	14		
P1.3(AIN3)	15		
P1.4(SS/AIN4)	16		
P1.5(MOSI/AIN5)	17		
P1.6(MISO/AIN6)	18		
P1.7(SPICLK/AIN7)	19		
XTALO	4	O	晶振输出管脚
XTALI	5	I	晶振输入管脚
P3.0(RXD)	2	I/O	P3 是可编程 I/O 口 P3 口还具有以下特殊功能 RxD(P3.0) 串行输入口 TxD(P3.1) 串行输出口 INT0(P3.2) 外部中断 0 INT1(P3.3) 外部中断 1 T0(P3.4) 定时器 0 外部输入 ECI(P3.4) PCA 外部时钟输入口 T1(P3.5) 定时器 1 外部输入 CEX1(P3.5) CEX0(P3.7)
P3.1(TXD)	3		
P3.2(INT0)	6		
P3.3(INT1)	7		
P3.4(ECI/T0)	8		
P3.5(CEX1/T1)	9		
P3.7(CEX0)	11		
RST	1	I	当该脚输入高电平超过 2 个机器周期时，芯片就会产生复位
VSS	10	G	电源地
VCC	20	P	电源

4. 方框图


5. 特殊功能寄存器SFR

5.1. SFR 映射表

	8	9	A	B	C	D	E	F
F8		CH	CCAP0H	CCAP1H				
F0	B		PCAPWM0	PCAPWM1				
E8		CL	CCAP0L	CCAP1L				
E0	ACC	WDTCR	IFD	IFADRH	IFADRL	IFMT	SCMD	ISPCR
D8	CCON	CMOD	CCAPM0	CCAPM1				
D0	PSW							
C8								
C0						ADCTL	ADCV	PCON2
B8	IP	SADEN						
B0	P3	P3M0	P3M1					IPH
A8	IE	SADDR						
A0								
98	SCON	SBUF						
90	P1	P1M0	P1M1					
88	TCON	TMOD	TL0	TL1	TH0	TH1	AUXR	
80		SP	DPL	DPH	SPISTAT	SPICTL	SPIDAT	PCON
	0	1	2	3	4	5	6	7

5.2. SFR 位分配

符号	描述	地址	位地址和符号								复位值
			Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	
SP	堆栈指针	81H									00000111B
DPL	数据指针低	82H									00000000B
DPH	数据指针高	83H									00000000B
SPISTAT	SPI 状态寄存器	84H	SPIF	WCOL							00xxxxxB
SPICTL	SPI 控制寄存器	85H	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	00000100B
SPIDAT	SPI 数据寄存器	86H									00000000B
PCON	电源控制	87H	SMOD1	SMOD0	GF	POF	GF1	GF0	PD	IDL	00010000B
TCON	定时器控制	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000B
TMOD	定时器模式	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00000000B
TL0	定时器 0 低字节	8AH									00000000B
TL1	定时器 1 低字节	8BH									00000000B
TH0	定时器 0 高字节	8CH									00000000B
TH1	定时器 1 高字节	8DH									00000000B
AUXR	辅助寄存器	8EH	T0x12	T1x12	URM0x6	EADCI	ESPI	ENLVFI	-	-	000000xxB
P1	端口 1	90H	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	11111111B
P1M0	端口 1 模式寄存器 0	91H	P1M0.7	P1M0.6	P1M0.5	P1M0.4	P1M0.3	P1M0.2	P1M0.1	P1M0.0	00000000B
P1M1	端口 1 模式寄存器 1	92H	P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0	00000000B
SCON	串口控制	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	Ti	RI	00000000B
SBUF	串口数据缓冲器	99H									xxxxxxxxB
IE	中断使能	A8H	EA	EPCA_LVD	ESPI_ADC	ES	ET1	EX1	ET0	EX0	00000000B
SADDR	从机地址	A9H									00000000B
P3	端口 3	B0H	P3.7		P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	11111111B
P3M0	端口 3 模式寄存器 0	B1H	P3M0.7		P3M0.5	P3M0.4	P3M0.3	P3M0.2	P3M0.1	P3M0.0	00000000B
P3M1	端口 3 模式寄存器 1	B2H	P3M1.7		P3M1.5	P3M1.4	P3M1.3	P3M1.2	P3M1.1	P3M1.0	00000000B
IPH	中断优先级 高	B7H		PPCAH_LVD	PSPIH_ADC	PSH	PT1H	PX1H	PT0H	PX0H	00000000B
IP	中断优先级 低	B8H		PPCA_LVD	PSPI_ADC	PSL	PT1L	PX1L	PT0L	PX0L	00000000B
SADEN	从机地址屏蔽	B9H									00000000B
ADCTL	ADC 控制寄存器	C5H									00000000B
ADCV	ADC 结果	C6H	ADCV.7	ADCV.6	ADCV.5	ADCV.4	ADCV.3	ADCV.2	ADCV.1	ADCV.0	00000000B
PCON2	电源控制 2	C7H	-	-	-	-	-	CKS2	CKS1	CKS0	Xxxxx000B
PSW	程序状态字	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00000000B
CCON	PCA 控制寄存器	D8H	CF	CR	-	-	-	-	CCF1	CCF0	00xx0xxxB
CMOD	PCA 模式寄存器	D9H	CIDL								00000000B
CCAPM0	PCA 通道 0 寄存器	DAH		ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECF0	00000000B
CCAPM1	PCA 通道 1 寄存器	DBH		ECOM1	CAPP1	CAPN1	MAT1	TOG1	PWM1	ECF1	x0000000B
ACC	累加器	E0H	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000B
WDTCR	看门狗定时器控制	E1H	WRF	--	ENW	CLW	WIDL	PS2	PS1	PS0	0x000000B
IFD	ISP Flash 数据	E2H									11111111B
IFADRH	ISP Flash 地址 高	E3H									00000000B
IFADRL	ISP Flash 地址 低	E4H									00000000B
IFMT	ISP 模式选择	E5H	--	--	--	--			MS1	MS0	xxxx0000B
SCMD	ISP 指令	E6H									xxxxxxxxB
ISPCR	ISP 控制寄存器	E7H	ISPEN	BS	SRST	CFAIL	--	WAIT.2	WAIT.1	WAIT.0	0000xxxxB
CL	PCA 低位计数器	E9H									00000000B
CCAP0L	PCA 通道 0 低位计数	EAH									00000000B
CCAP1L	PCA 通道 1 低位计数	EBH									00000000B
B	B 寄存器	F0H	F7H	F6H	F5H	F4H	F3H	F2H	F1H	F0H	00000000B
PCAPWM0	PCA 通道 0 PWM 计数	F2H									00000000B
PCAPWM1	PCA 通道 1 PWM 计数	F3H									00000000B
CH	PCA 高位计数器	F9H									00000000B
CCAP0H	PCA 通道 0 高位计数	FAH									00000000B
CCAP1H	PCA 通道 1 高位计数	FBH									00000000B

6.2. CPU 时序

MA801 是一个基于1-T结构80C51 CPU的单芯片微处理器，指令完全兼容8051，执行一条指令只要 1~7时钟周期(是标准8051的 6~7倍)。它通过管道结构加快了指令执行速度超过标准8051结构。指令时序不同于标准8051。

多数8051执行指令的差别是基于机器周期和时钟周期的，机器周期为2到12个时钟周期。然而，1T-80C51是基于时钟周期时序的。所有指令时序被指定为时钟周期。关于1T-80C51指令的更多详情（包括每条指令的助记符、字节数、时钟周期）请参考“指令集”。

6.3. CPU 寻址方式

直接寻址

在指令中直接给出操作数地址的就属于直接寻址。此时，指令中的操作数部分就是操作数的地址。例如指令：

```
MOV    A,4FH           ;(A)←(4FH)
```

可用于直接寻址的空间是，内部数据RAM的低128字节及特殊功能寄存器SFRs。

寄存器间接寻址

由指令中指出某一个寄存器的内容作为操作数的地址。内部RAM和外部RAM都能通过间接寻址方式进行访问。使用当前工作寄存器组中的R0或R1存放操作数单元的地址指针（8位地址），在执行PUSH（压栈）和POP（出栈）指令时采用堆栈指针SP作寄存间接寻址。而如果地址是16位时就只能使用DPTR数据指针作间接寻址了。例如指令：

```
MOV    A,@R0           ;(A) ←((R0))
MOVX   A,@R1           ;(A) ←((R1))
MOVX   A,@DPTR        ;(A) ←((DPTR))
```

寄存器寻址(REG)

寄存器寻址就是以通过寄存器的内容作为操作数。在指令的助记符号中直接以寄存器的名字来表示操作数的地址。例如指令：

```
MOV    A,R0            ;(A) ←(R0)
ADD    A,R0            ;(A) ←(Acc)+(R0)
```

能用于这种寻址方式的寄存器还有ACC、B、DPTR、AB(双字节)和CY(位累加器)。

变址寻址

以某个寄存器的内容作为基本地址，然后在这个基本地址基础上加上地址偏移量才是真正的操作数地址。例如指令：

```
MOVC   A,@A+DPTR      ;(A) ←((A)+(DPTR))
```

不论用DPTR或是PC作为基址指针，变址寻址方式都只适用于8051的程序存储器，通常用于读取数据表。

立即寻址

指令中地址码部分给出的就是操作数。即取出指令的同时立即得到了操作数。例如指令：

```
MOV    A,#4FH         ;(A) ←6FH
```

相对寻址

相对寻址时，由程序计数器PC提供的基地址与指令中提供的偏移量rel相加，得到操作数的地址。这时指出的地址是操作数与现行指令的相对位置。例如指令：

```
SJMP   rel            ;PC←(PC)+2+rel
```

位寻址

操作数是二进制数的某一位，其位地址出现在指令中，例如指令：

```
SETB   bit           ;(bit) ←1
```

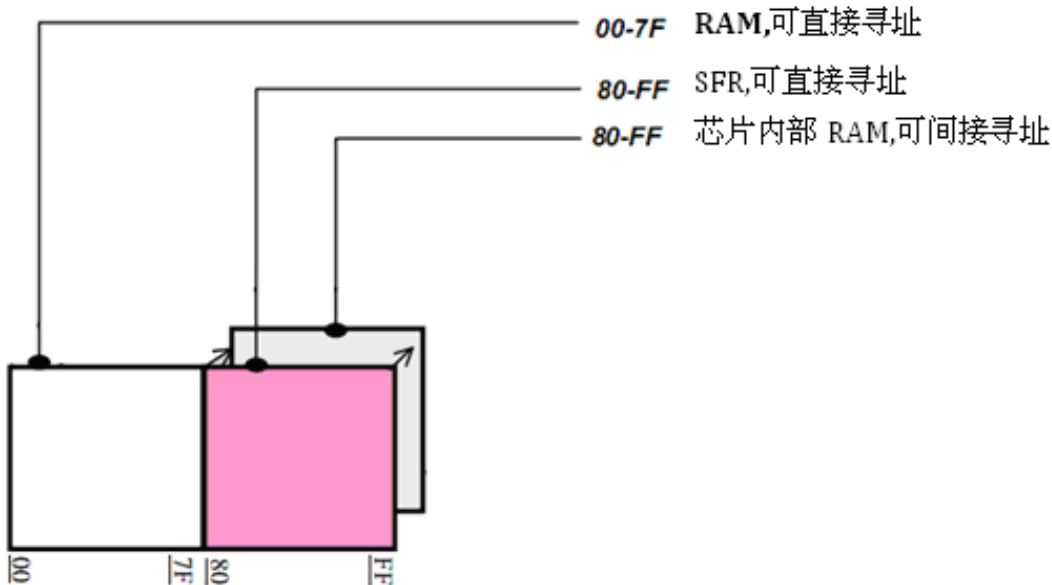
7. 存储器结构

像所有的 80C51一样，MA801 的程序存储器和数据存储器的地址空间是分开的，这样8位微处理器可以通过一个8位的地址快速而有效的访问数据存储器。

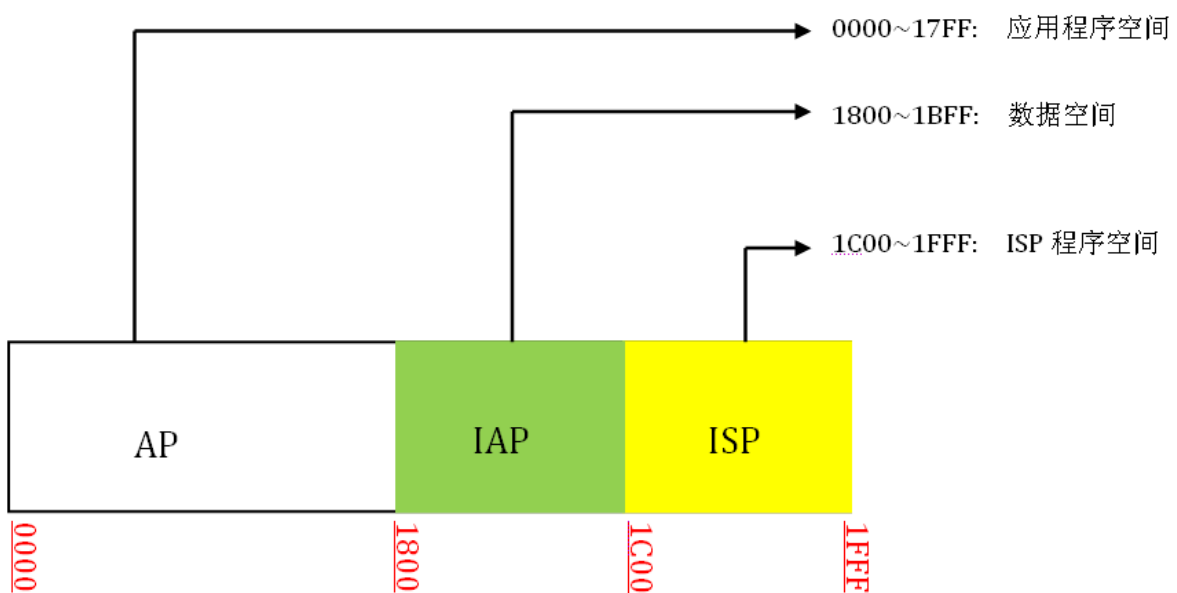
程序存储器(APROM) 在运行时只能读取，不能写入。最大8K字节。在MA801中，所有的程序存储器都是片上Flash存储器。因为没有设计外部程序使能 (/EA)和编程使能 (/PSEN) 信号，所以不允许外接程序存储器。

数据存储器使用与程序存储器不同的地址空间。MA801 有256字节的内部RAM。

RAM 空间



MA801 FLASH 空间

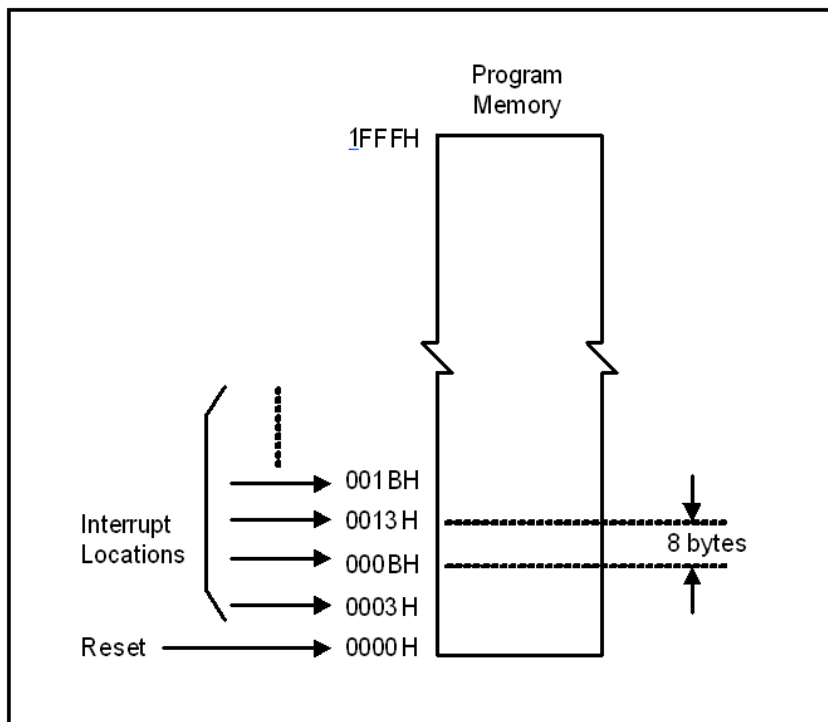


7.1. 片上程序存储器

程序存储器用来保存让CPU进行处理的程序代码，如图 8-1所示。复位后，CPU从地址为0000H的地方开始运行，用户应用代码的起始部分应该放在这里。为了响应中断，中断服务位置(被称为中断矢量)应该位于程序存储器。每个中断在程序存储器中有一个固定的起始地址，中断使CPU跳到这个地址运行中断服务程序。举例来说，外部中断0被指定到地址0003H，如果使用外部中断0，那么它的中断服务程序一定是从0003H开始的。如果中断未被使用，那么这些地址就可以被一般的程序使用。

中断服务程序的起始地址之间有8字节的地址间隔：外部中断0，0003H；定时器0，000BH；外部中断1，0013H；定时器1，001BH等等。如果中断服务程序足够短，它完全可以放在这8字节的空间中。如果其他的中断也被使用的话，较长的中断服务程序可以通过一条跳转指令越过后面的中断服务起始地址。

图 8-1 程序存储器



7.2. 片上数据存储器 RAM

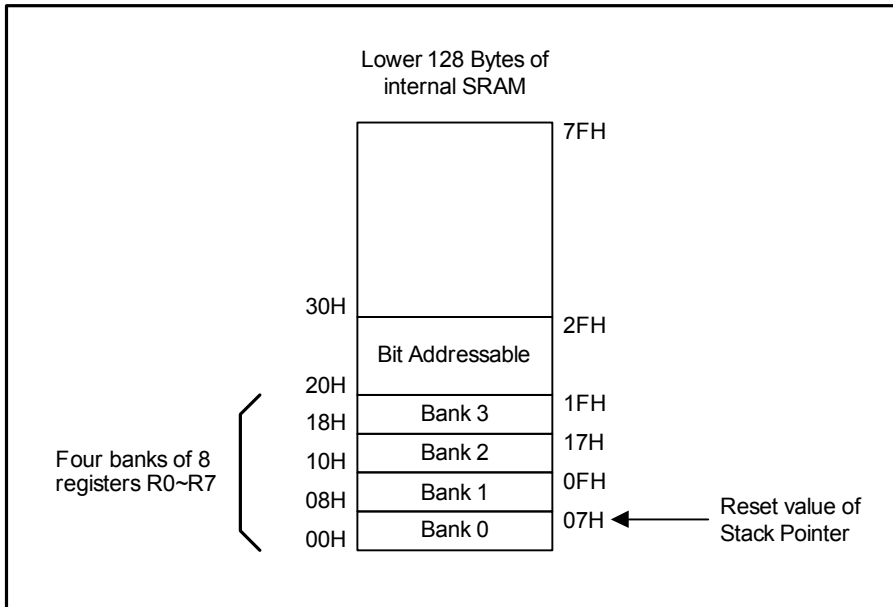
图 8-2 向MA801使用者展示了内部和外部数据存储器的空间划分。内部数据存储器被划分为三部分,通常被称为低128字节 RAM , 高128字节 RAM 和128字节 SFR 空间。内部数据存储器的地址线只有8位宽,因此地址空间只有256字节。SFR 空间的地址高于7FH,用直接地址访问;而用间接访问的方法访问高128字节的RAM。这样虽然SFR和高128字节RAM占用相同的地址空间,但他们实际上是分开的。

图 8-2 所示,低128字节RAM与所有80C51 一样。最低的32字节被划分为4组每组8字节的寄存器组。指令中称这些寄存器为R0到R7。程序状态字 (PSW) 中的两位用于选择哪组寄存器被使用。这使得程序空间能够被更有效的使用,因为对寄存器访问的指令比使用直接地址的指令短。接下来的16字节是可以位寻址的存储器空间。80C51的指令集包含一个位操作指令集,这区域中的128位可以被这些指令直接使用。位地址从00H开始到7FH结束。

所有的低128字节RAM都可以用直接或间接地址访问,而高128字节RAM只能用间接地址访问。

SFR包括端口寄存器,定时器和外围器件控制器,这些寄存器只能用直接地址访问。SFR 空间中有16个地址同时支持位寻址和直接寻址。可以位寻址的 SFR 的地址末位是0H 或8H。

图 8-2 低 128 字节内部 RAM



7.3. 关于C51 编译器的声明识别符

C51编译器的声明识别符与 MA801 存储空间的对应关系。

data

128字节的内部数据存储空间 (00h~7Fh)；使用除MOVX和MOVC以外的指令，可以直接或间接的访问。全部或部分的堆栈可能保存在此区域中。

idata

间接数据；256字节的内部数据存储空间 (00h~FFh) 使用除MOVX和 MOVC以外的指令间接访问。全部或部分的堆栈可能保存在此区域中。此区域包括 data区和data区以上的128字节。

sfr

特殊功能寄存器； CPU寄存器和外围部件控制/状态寄存器，只能通过直接地址访问。

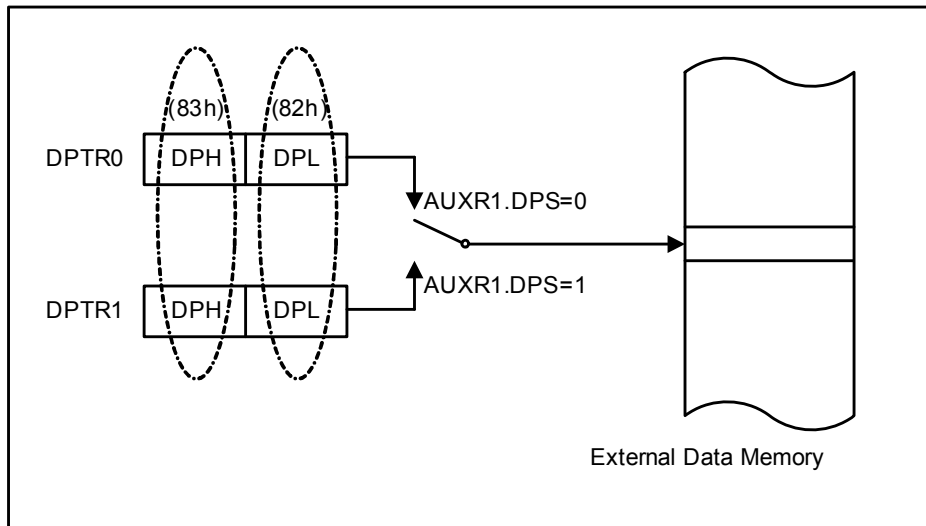
code

8K程序存储空间；通过“MOVC @A+DTPR”访问，作为程序的一部分被读取。MA801有8K字节的片上 code存储器。

8. 双数据指针寄存器 (DPTR)

通过数据指针用来加速代码执行和减少代码尺寸，双DPTR结构是一种方法。芯片将指定外部数据存储器的定位地址。外部存储器有两个16位DPTR寄存器，和一个控制位称作为DPS(AUXR1.0)，允许在程序代码和外部存储器之间的切换。

图 9-1 双 DPTR



DPTR 指令

使用DPS位的六条指令参考DPTR的当前选择，如下：

```

INC    DPTR      ; 数据指针加1
MOV    DPTR,#data16 ; DPTR加载16位常量
MOVC   A, @A+DPTR ; 将代码字节移动到ACC
MOVX   A, @DPTR   ; 移动外部RAM(16位地址)到ACC
MOVX   @DPTR, A   ; 移动ACC到外部RAM(16位地址)
JMP    @A+DPTR   ; 直接跳转到DPTR
  
```

9. I/O 结构

9.1. IO 配置

所有 I/O 端口引脚可以配置为四种模式中的其中一种。这四种类型有：准双向口(标准 8051 的 I/O 端口)、上拉输出、集电极开路输出和输入(高阻抗输入)。下面描述这四种 I/O 口模式。

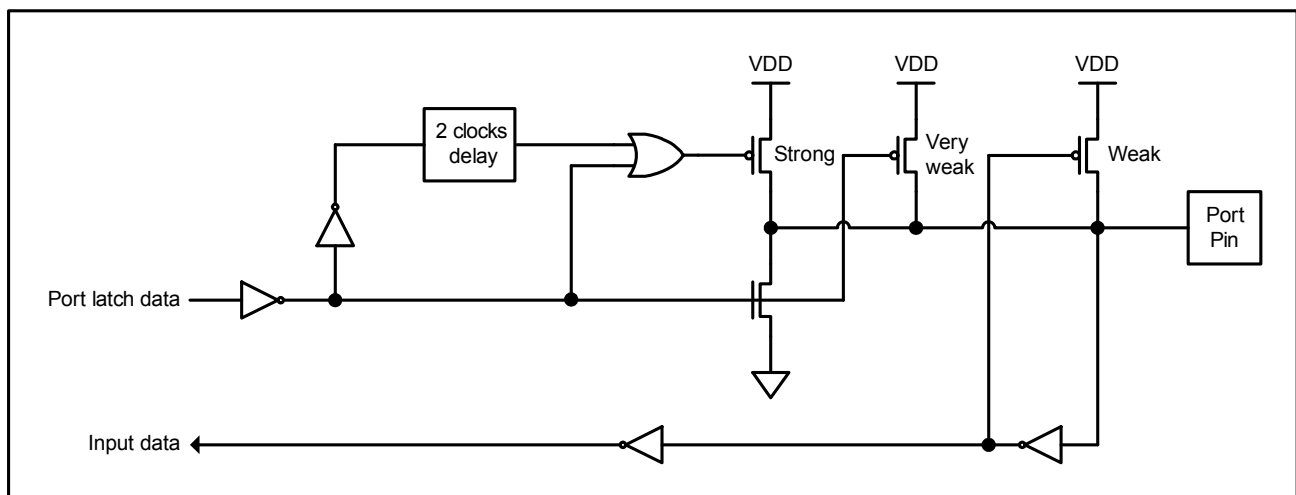
9.1.1. 准双向口

端口引脚工作在准双向模式时与标准 8051 端口引脚类似。一个准双向端口用作输入和输出时不需要对端口重新配置。这种可能是因为端口输出逻辑高时，弱上拉，允许外部器件拉低引脚。当输出低时，强的驱动能力可吸收大电流。在准双向输出时有三个上拉晶体管用于不同的目的。

其中的一种上拉，称为微上拉，只要端口寄存器的引脚包含逻辑 1 则打开。如果引脚悬空，则这种非常弱上拉提供一个非常小的电流将引脚拉高。第二种上拉称为“弱上拉”，端口寄存器的引脚包含逻辑 1 时且引脚自身也在逻辑电平时打开。这种上拉对准双向引脚提供主要的电流源输出为 1。如果引脚被外部器件拉低，这个弱上拉关闭，只剩一个微上拉。为了在这种条件下将引脚拉低，外部器件不得不吸收超过弱上拉功率的电流，且拉低引脚在输入的极限电压之下。第三种上拉称为“强”上拉。这种上拉用于加速准双向端口的上升沿跳变，当端口寄存器从逻辑 0 到逻辑 1 时。当这发生时，强上拉打开两个 CPU 时钟，快速将端口引脚拉高。

准双向端口配置如下图所示。

图 10-1 准双向 I/O 口

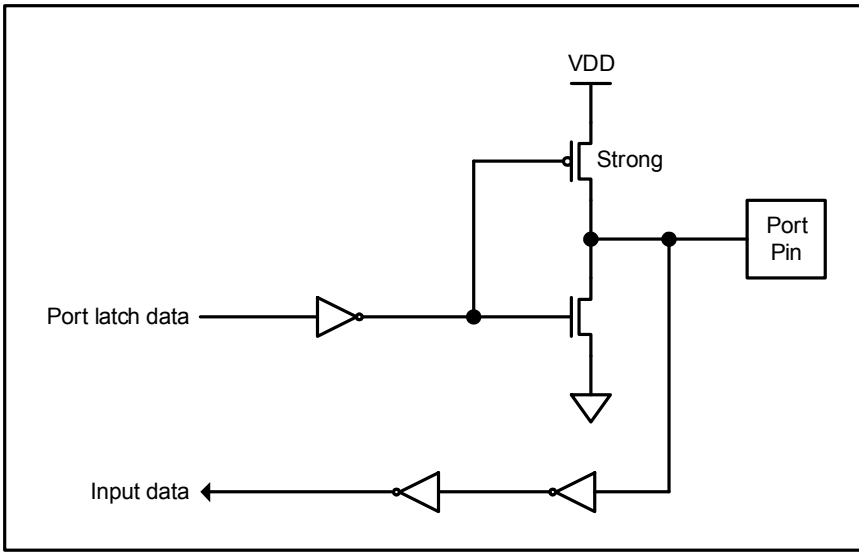


9.1.2. 推挽输出

推挽输出配置有下拉，和开漏输出、准双向输出模式有着相同的结构，当端口寄存器包含逻辑 1 时提供一个连续的强上拉。当一个端口输出需要更大的电流时可配置为推挽输出模式。另外，在这种配置下的端口输入引脚和输入路径的准双向模式的配置相同。

推挽输出配置如下图所示。

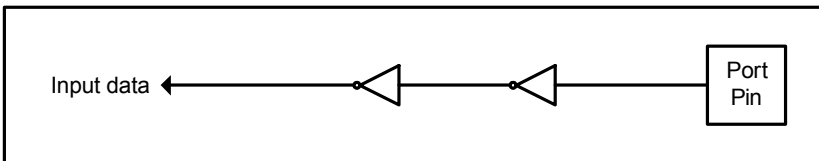
图 10-2 推挽输出



9.1.3. 仅输入(高阻输入)

仅输入模式在引脚上没有任何上拉电阻，如下图所示。

图 10-3 仅输入

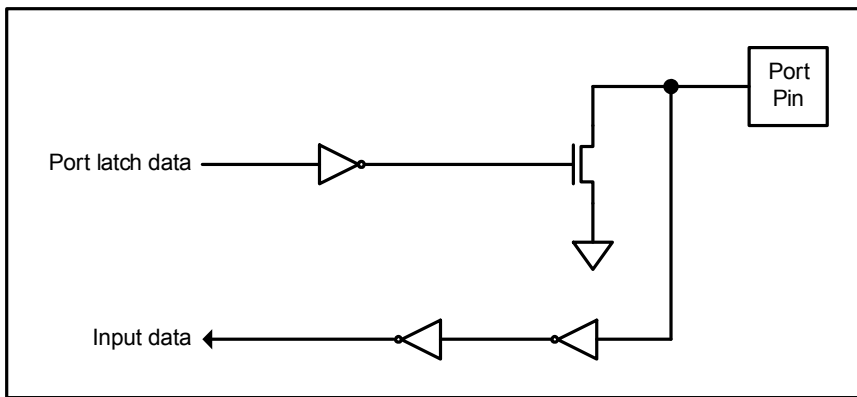


9.1.4. 开漏集输出

配置为开漏输出时，当端口寄存器包含逻辑 0 时，关闭所有上拉，只有端口引脚的下拉晶体管。使用这个功能配置应用，端口引脚必须有外部上拉，典型的将电阻接到 VDD。这个模式的下拉和准双向端口的模式相同。另外，在这种配置下的端口输入引脚的输入路径和准双向模式相同。

开漏输出端口配置如图 10-2 所示。

图 10-4 开漏集输出



9.2. I/O 端口寄存器

MA801 所有端口可通过软件个别的、独立的配置为四种中的一种类型，基于位位基础，如下表所示。这四种类型有：准双向(标准 8051 的 I/O 端口)、上拉输出、集电极开路输出和输入(高阻抗输入)。每个端口有两个模式寄存器来选择各个端口引脚的类型。

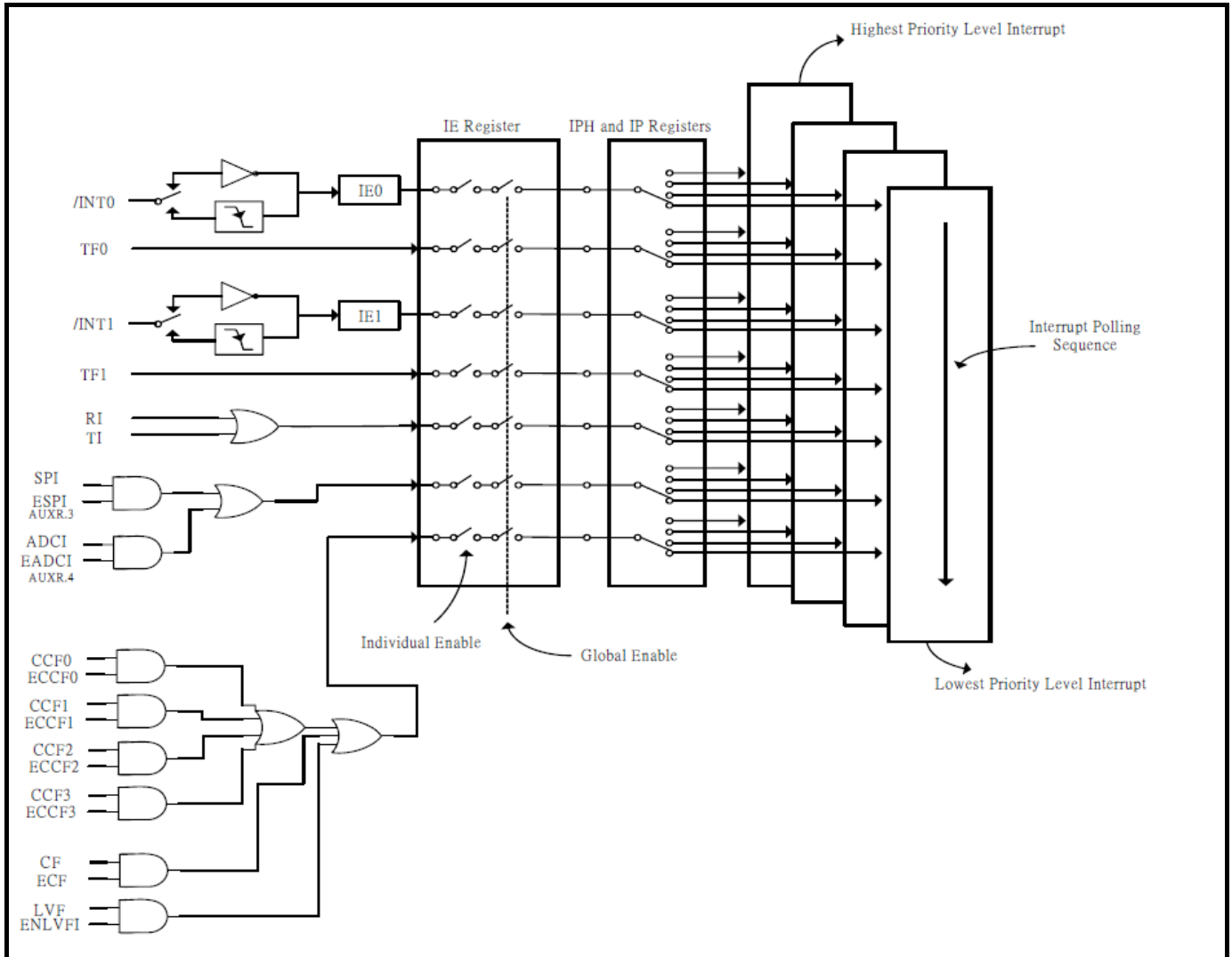
表 10-2 端口配置设定

PxM0.y	PxM1.y	端口模式
0	0	准双向端口
0	1	推挽式输出
1	0	仅输入(高阻抗输入)
1	1	开漏集输出

这里 $x=1, 3$ (端口号), $y=0\sim7$ (端口引脚)。寄存器 PxM0 和 PxM1 列表如下。

10. 中断

10.1. 中断结构



10.2. 中断寄存器

IE: 中断使能寄存器

SFR 地址 = 0xA8

SFR 页 = 全部

复位值 = 0000-0000

7	6	5	4	3	2	1	0
EA	EPCA_LVD	ESPI_ADC	ES	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: EA, 全局中断使能位

0: 禁止所有中断

1: 使能全局中断

Bit 6: EPCA_LVD: PCA 和低压中断使能位.

0: 禁止 PCA 和低压中断

1: 使能 PCA 和低压中断

Bit 5: ESPI_ADC: SPI_ADC 中断使能位

0: 禁止 SPI_ADC 中断

1: 使能 SPI_ADC 中断

Bit 4: ES, 串口中断使能位

0: 禁止串口中断

1: 使能串口中断

Bit 3: ET1, 定时器 1 中断使能位

0: 禁止定时器 1 中断

1: 使能定时器 1 中断

Bit 2: EX1, 外部中断 1 使能位

0: 禁止外部中断 1

1: 使能外部中断 1

Bit 1: ET0, 定时器 0 中断使能位

0: 禁止定时器 0 中断

1: 使能定时器 0 中断

Bit 0: EX0, 外部中断 0 使能位

0: 禁止外部中断 0

1: 使能外部中断 0

IP: 中断优先级寄存器 0 低

SFR 地址 = 0xB8

SFR 页 = 全部

复位值 = 0000-0000

7	6	5	4	3	2	1	0
	PPCA_LVD	PSPI_ADC	PS	PT1	PX1	PT0	PX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: 保留

Bit 6: PPCA_LVD, PCA_LVD 中断优先级低位

Bit 5: PSPIH_ADC, ADC 中断优先级低位

Bit 4: PS, 串行口中断优先级 低位

Bit 3: PT1, 定时器 1 中断优先级 低位

This document information is the intellectual property of Megawin Technology.

© Megawin Technology Co., Ltd. 2009 All rights reserved.

QP-7300-03D

Bit 2: PX1, 外部中断 1 优先级 低位
 Bit 1: PT0, 定时器 0 中断优先级 低位
 Bit 0: PX0, 定时器 0 中断优先级 低位

IPH: 中断优先级寄存器 高

SFR 地址 = 0xB7

SFR 页 = 全部

复位值 = 0000-0000

7	6	5	4	3	2	1	0
	PPCAH_LVD	PSPIH_ADC	PSH	PT1H	PX1H	PT0H	PX0H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: 保留
 Bit 6: PPCAH_LVD, PCA_LVD 中断优先级 高位
 Bit 5: PSPIH_ADC, SPI_ADC 中断优先级 高位
 Bit 4: PSH, 串行口中断优先级 高位
 Bit 3: PT1H, 定时器 1 中断优先级 高位
 Bit 2: PX1H, 外部中断 1 优先级 高位
 Bit 1: PT0H, 定时器 0 中断优先级 高位
 Bit 0: PX0H, 定时器 0 中断优先级 高位

IP, IPH, 组合成 4 级优先级中断, 见下表:

{IPH., IP}	优先级
11	1 (最高)
10	2
01	3
00	4

MA801 提供 7 个中断源, 4 级优先级。每个中断源都有两个对应的位来定义它的优先级。一个在 **IPH**, 另一个在 **IP**。处理高优先级中断时, 不会响应低优先级的中断请求。如果两个不同优先级的中断同时发出请求, 高优先级的中断请求将会被响应。如果相同优先级的中断同时发出请求, 则由内部优先级来决定哪个中断会被响应。下表说明了内部优先级和中断向量地址。

中断源	向量地址	优先级别
外部中断 0	0003H	1 (高)
定时器 0	000BH	2
外部中断 1	0013H	3
定时器 1	001BH	4
串行口中断	0023H	5
SPI/ADC	002BH	6
PCA/LVF	0033H	7

外部中断/INT0、/INT1、分别通过 TCON 的 IT0、IT1 触发。实际产生的中断标志位是 TCON 的 IE0、IE1。

定时器 0 和定时器 1 中断由 TF0 和 TF1(分别由各自的定时/计数寄存器控制)产生。当产生定时器中断时, 进入中断服务程序后由片内硬件清除标志位。

串口中断由 RI 和 TI 的逻辑或产生。进入中断服务程序后, 这些标志均不能被硬件清除。实际上, 中断服务程序通常需要确定是由 RI 还是 TI 产生的中断, 然后由软件清除中断标志。

所有这些产生中断的位都可通过软件置位或清零, 与通过硬件置位或清零的效果相同。简而言之, 中断可由软件产生, 推迟或取消。

硬件如何响应中断

每一个机器周期的 S5P2 时间采样中断请求标志，采样被保持到下一个 S5P2 时期。如果一个中断请求标志在第一个 S5P2 时间被置位，在第二个 S5P2 时间（采样周期）将被硬件发现，这时若没有下面任何一个阻止条件则中断系统将产生一个硬件调用 LCALL 转去执行相应的中断服务程序。

阻止条件如下：

1. CPU 正在处理同级或更高级的中断；
2. 现行机器周期不是所执行的最后一个机器周期；
3. 正在执行的是 RETI 或是访问 IE 或 IP 的指令。

上叙任何一个条件都会阻止硬件去响应发生的中断请求。条件 2 保证了在进入其它中断服务程序前执行完当前指令。条件 3 保证了执行完 RETI 指令或是访问 IE 或 IP 的指令后在进入其它中断向量前能执行至少一条或更多条的指令。

每个机器周期硬件都会重复查询中断请求标志，查询到的值都是前一个 S5P2 时间的状态。值得注意的是如果一个被触发的中断标志由于上面的阻止条件发生了而没得到响应，并且此标志没有持续到阻止条件消失，这样的中断请求将得不到响应。也就是说，中断处理本身不能锁存中断，譬如外部电平中断若在电平出现时被屏蔽，而在中断识别之前电平消失，它被完全忽略。也就是每个查询都是新状态。

11. 定时器/计数器

MA801 有 2 个定时/计数器：定时器 0、定时器 1 和 PCA 定时器。定时器 0/1 能被配置作为定时器或事件计数器。PCA 定时器能配置为定时器 或 PWM 发生器。

定时器功能，TLx 寄存器每 12 个时钟周期或 1 个时钟周期加 1，通过软件设置 AUXR.T0X12、AUXR.T1X12 来选择。每 12 个时钟周期加一，计数速率达 1/12 的晶振频率。

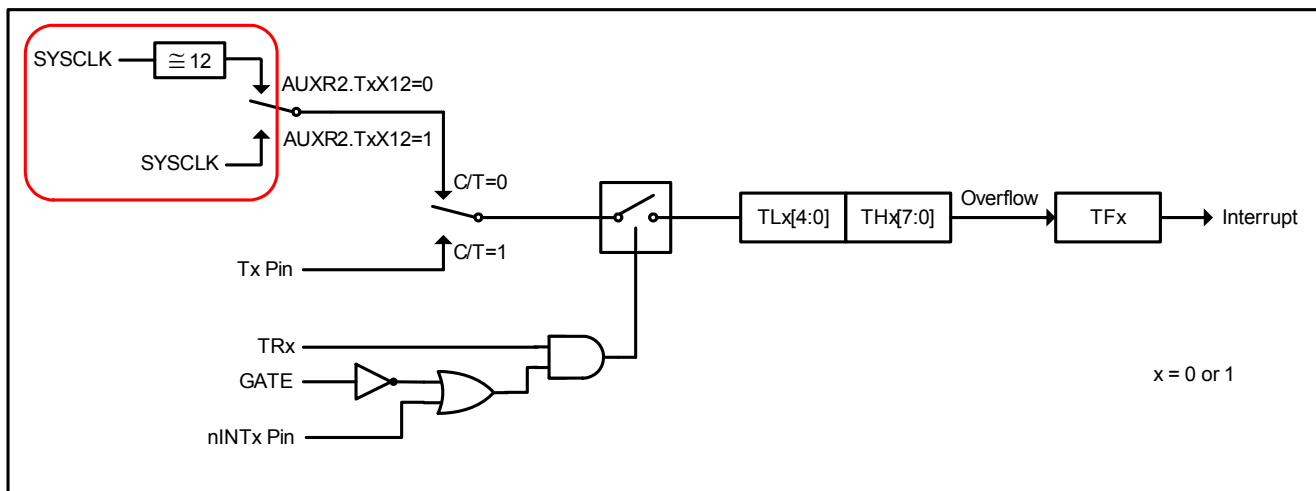
计数器功能，根据对应的外部输入引脚的下降沿 T0、T1 寄存器加 1。在这些功能中，每个时钟周期对外部输入信号(T0、T1 引脚)进行采样。当采样信号出现一个高电平接着一个低电平，计数加 1。当检测到跳变时新计数值出现在寄存器中。对定时器 0 和定时器 1 来说，需要用两个时钟周期来识别下降沿跳变，最大的计数速率为 1/2 的晶振频率。外部输入信号没有严格的周期限制，但是要确保在电平改变前至少有一次采样，对定时器 0 和定时器 1 来说信号应该至少保持一个时钟。

11.1. 定时器 0 和定时器 1

11.1.1. 模式 0

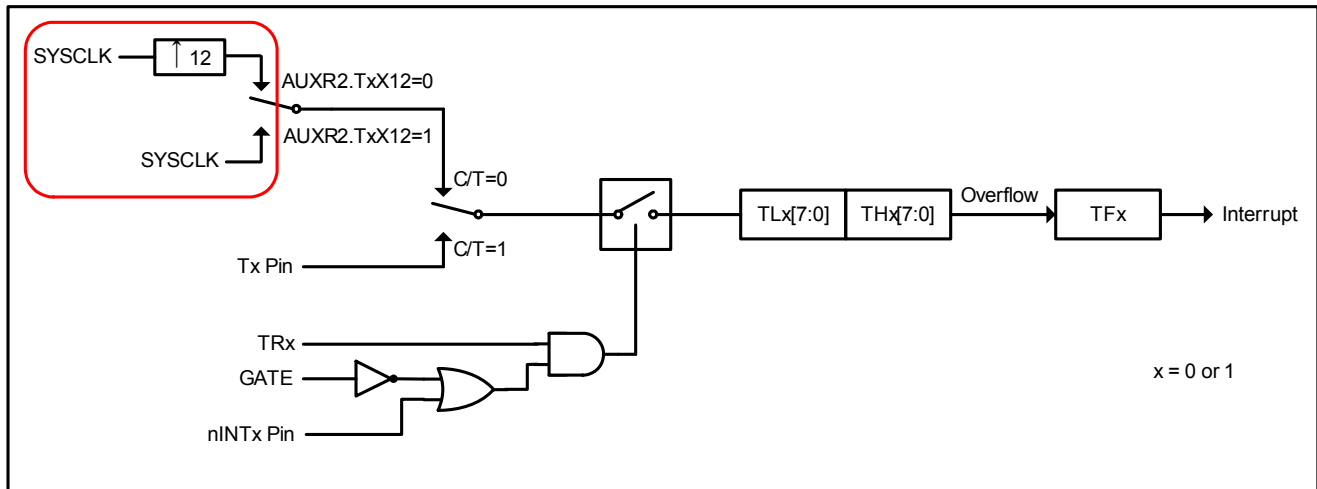
在这个模式，定时器寄存器配置为一个 13 位寄存器。计数器所有位从全 1 翻转到全 0，置位定时器中断标志位 TFx。当 TRx=1 且 GATE=0 或 /INTx=1，定时器使能输入计数。(置 GATE=1 时通过外部输入/INTx 控制定时器，以便脉冲宽度测量)。TRx 和 TFx 控制位在专用寄存器 TCON。GATE 位在 TMOD。有两个不同的 GATE 位，一个是定时器 0(TM0D.7) 另一个是定时器 1(TM1D.3)。

13 位寄存器包含 THx 的所有 8 位和 TLx 的低 5 位。TLx 的高 3 位是不确定的可以忽略。置位运行标志(TRx)不会清除寄存器。意思是说用户在开始计数前应对 THx 和 TLx 进行初始化。



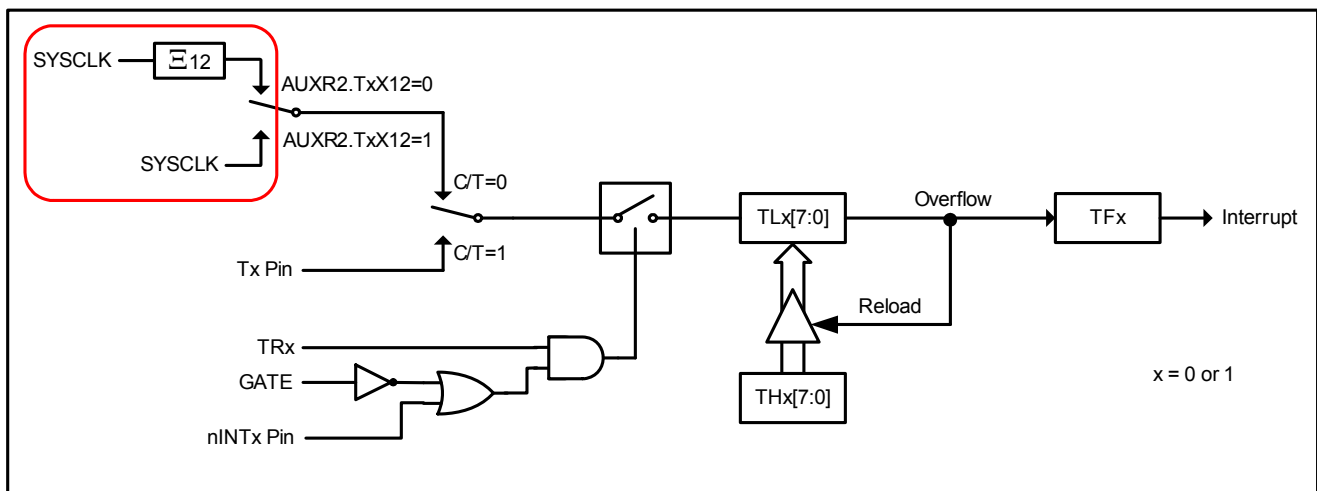
11.1.2. 模式 1

除了定时器的寄存器使用全部 16 位外，模式 1 和模式 0 是相同的。在这个模式，THx 和 TLx 串联，没有预分频。



11.1.3. 模式 2

模式 2 配置定时器寄存器为一个自动加载的 8 位计数器(TLx)。TLx 溢出不仅置位 TFx，而且也将 THx 的内容加载到 TLx，THx 内容由软件预置，加载不会改变 THx 的值。

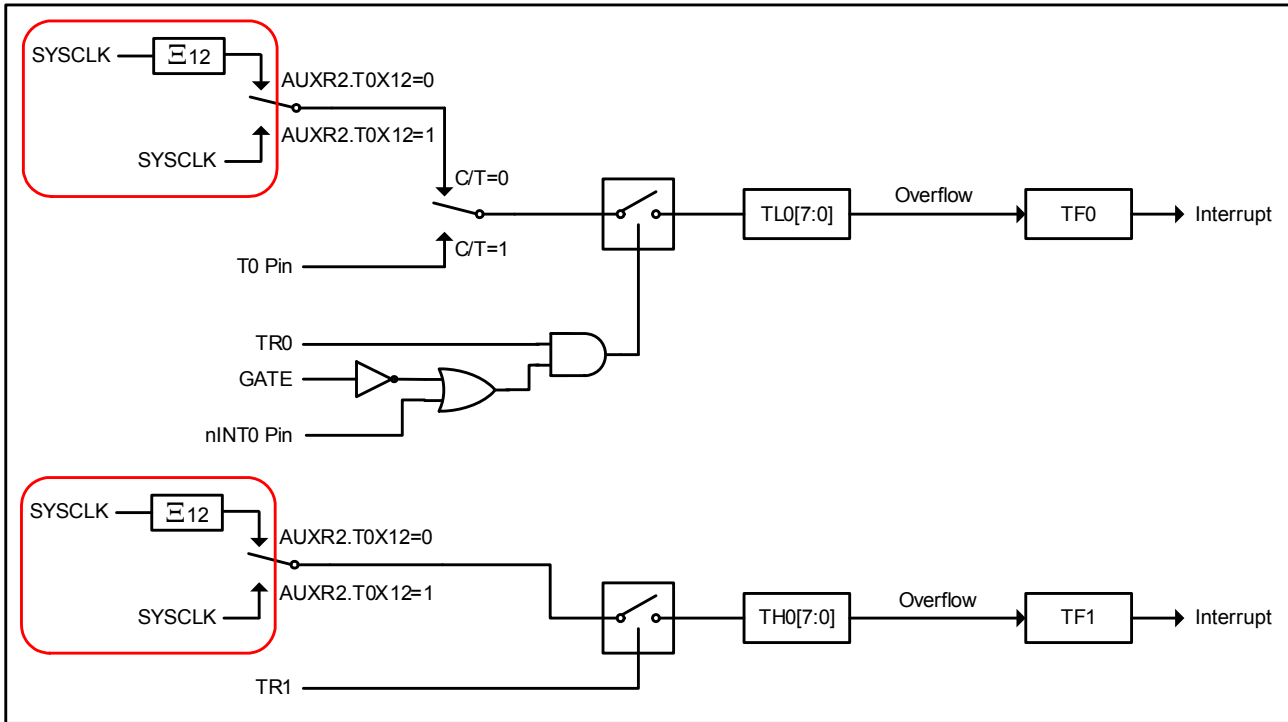


11.1.4. 模式 3

定时器 1 在模式 3 保持计数值。效果和设置 TR1=0 一样。

定时器 0 在模式 3 建立 TL0 和 TH0 两个独立的计数器。TL0 使用定时器 0 控制位：C/T、GATE、TR0、/INT0 和 TF0。TH0 锁定为定时器功能(每个机器周期计数)且接替定时器 1 来使用 TR1 和 TF1，因从 TH0 控制定时器 1 中断。

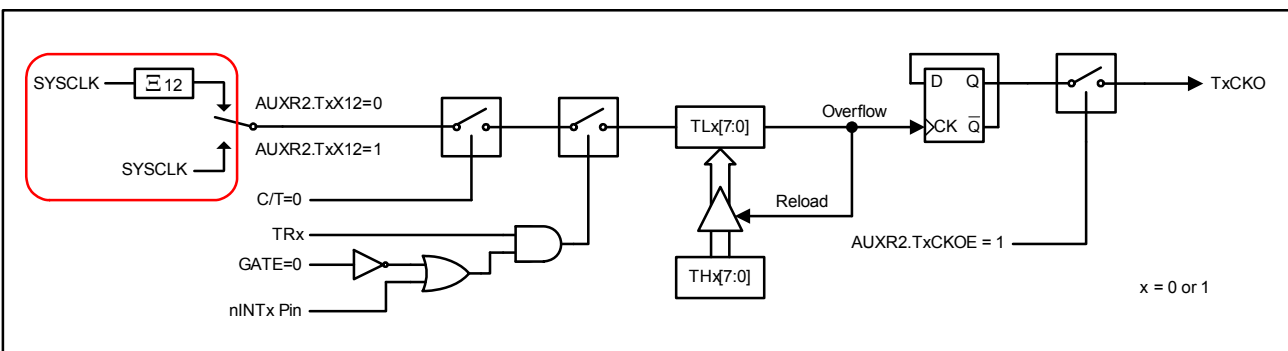
模式 3 提供当有额外的需求应用时的一个 8 位定时器或计数器时。当定时器 0 在模式 3 时，定时器 1 可打开或关闭并切换到脱离，进入到自己的模式 3，或仍然可用作为串行口的波特率发生器，或者不需要中断的其它应用。



11.1.5. 定时器时钟输出

定时器的可编程时钟输出模式，则从引脚 TxCKO 输出占空比为 50%的时钟周期。输出频率根据系统时钟频率(Fosc)和加载值到 THx 寄存器的值决定，公式如下所示。时钟输出模式编程步骤如下：

1. 在 AUSR 寄存器置位 TxCKOE。
2. 在 TMOD 寄存器清除定时器 0/1 的 C/T 位。



$$T0/T1 \text{ Clock-out Frequency} = \frac{\text{SYSCLK Frequency}}{n \times (256 - THx)} \quad \begin{array}{l} ; n=24, \text{ if } TxX12=0 \\ ; n=2, \text{ if } TxX12=1 \\ ; x = 0 \text{ or } 1 \text{ \& } C/T = 0 \end{array}$$

11.1.6. 定时器 0/1 寄存器

TMOD: 定时器/计数器模式寄存器

SFR 地址 = 0x89

SFR 页 = 全部

复位值 = 0000-0000

7	6	5	4	3	2	1	0
GATE	C/T	M1	M0	GATE	C/T	M1	M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

|←----- Timer1 ----->|←----- Timer0 ----->|

GATE: 当门控位置位时，只有在 /INT0 或 /INT1 引脚是高电平且 TR0 或 TR1 控制位置位时，定时器/计数器 0 或 1 使能。当门控位置清零时，只要 TR0 或 TR1 置 1 定时器 0 或 1 使能。

C/T: 定时器或计数器选择器。清零为定时器功能(从内部系统时钟输入)。置位为计数器功能(从 T0 或 T1 引脚输入)。

M1	M0	工作模式
0	0	13 位定时器/计数器。
0	1	16 位定时器/计数器。THx 与 TLx 串联，没有分频器
1	0	8 位自动重载定时器/计数器。THx 保持一个值，并在每次溢出时加载到 TLx
1	1	(定时器 0)TL0 是一个 8 位定时器/计数器并通过标准定时器 0 的控制位控制。TH0 仅仅是一个 8 位定时器通过定时器 1 的控制位控制
1	1	(定时器 1)定时器/计数器停止

TCON: 定时器/计数器控制寄存器

SFR 地址 = 0x88

SFR 页 = 全部

复位值 = 0000-0000

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

TF1: 定时器 1 溢出标志位。定时器/计数器溢出时由硬件置位。处理器进入中断向量程序由硬件清零。

TR1: 定时器 1 运行控制位。通过软件置位/清零开启或关闭定时器/计数器 1。

TF0: 定时器 0 溢出标志位。定时器/计数器溢出时由硬件置位。处理器进入中断向量程序由硬件清零。

TR0: 定时器 0 运行控制位。通过软件置位/清零开启或关闭定时器/计数器 0。

IE1: 外部中断 1 请求标志。外部中断 1 由边沿或电平触发（由 IT1 设置）硬件置标志。

IT1: 外部中断 1 类型控制位。软件选择下降沿/低电平触发外部中断 1。

IE0: 外部中断 0 请求标志。外部中断 0 由边沿或电平触发（由 IT0 设置）硬件置标志。

IT0: 外部中断 0 类型控制位。软件选择下降沿/低电平触发外部中断 0。

TL0: 定时器 0 低

SFR 地址 = 0x8A

SFR 页 = 全部

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TL0[7]	TL0[6]	TL0[5]	TL0[4]	TL0[3]	TL0[2]	TL0[1]	TL0[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TH0: 定时器 0 高

SFR 地址 = 0x8C

SFR 页 = 全部

复位值 = 0000-0000

7	6	5	4	3	2	1	0
TH0[7]	TH0[6]	TH0[5]	TH0[4]	TH0[3]	TH0[2]	TH0[1]	TH0[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12. 串行口 (UART)

MA801 支持一个全双工的串行口，意思是可以同时发送和接收数据。它也有一个接收缓冲，意味着在前一个接收到的字节没有从寄存器读出前，就可以开始接收第二个字节。(然而，如果第一个字节在第二个字节接收完成前仍然没有被读出，则其中的一个字节将会丢失)。串行口的接收和发送寄存器都通过特殊寄存器 SBUF 来访问。写到 SBUF 加载到发送寄存器，当从 SBUF 读时是通过一个物理上独立分离的接收寄存器。

串行口可以工作在四种模式：模式 0 提供同步通讯同时模式 1、2 和模式 3 提供异步通讯。异步通讯作为一个全双工的通用异步收发器(UART)，可以同时发送和接收并使用不同的波特率。

模式 0：8 位数据(低位先出)通过 RXD 传送和接收。TXD 总是作为输出移位时钟。波特率固定为系统时钟频率的十二分之一，也就是 $F_{osc}/12$ 。

模式 1：10 位通过 TXD 传送或通过 RXD 接收，一个起始位(0)，8 个数据位(低位优先)，和一个停止位(1)。在接收时，停止位进入到专用寄存器(SCON)的 RB8。波特率是可变的。

$$\text{Baud Rate (for Mode 1)} = \frac{2^{\text{SMOD}}}{32} \times \text{(Timer-1 overflow rate)}$$

模式 2：11 位通过 TXD 传送或通过 RXD 接收，起始位(0)，8 个数据位(低位优先)，一个可编程的第九个数据位和一个停止位(1)。在传送时，第 9 个数据位(TB8 在 SCON 寄存器)可以分配为 0 或者 1。例如，奇偶检验位(P，在 PSW 寄存器)可以移到 TB8 中。在接收时，第九个数据位到 SCON 寄存器中的 RB8，同时忽略停止位。波特率可以配置为 1/32 或 1/64 的系统时钟频率。也就是 $F_{osc}/64$ 或 $F_{osc}/32$ 。

$$\text{Baud Rate (for Mode 2)} = \frac{2^{\text{SMOD}}}{64} \times F_{osc}$$

模式 3：11 位通过 TXD 传送或通过 RXD 接收，起始位(0)，8 个数据位(低位优先)，一个可编程的第九个数据位和一个停止位(1)。实际上，模式 3 和模式 2 除了波特率不相同之外其它的都相同。模式 3 的波特率是可变的。

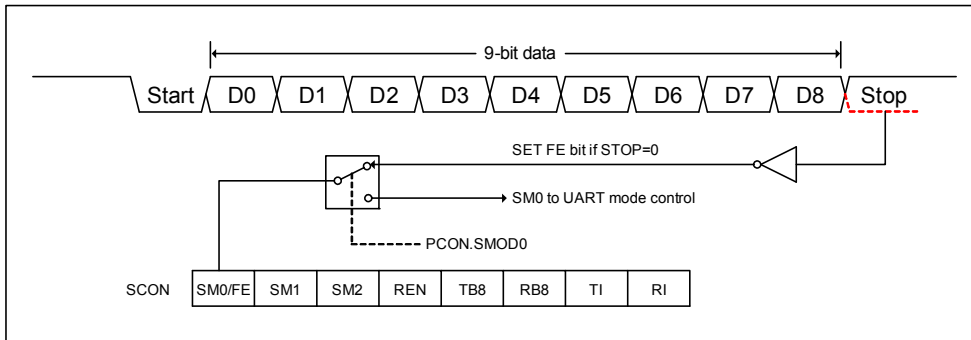
在四种模式中，使用 SBUF 作为一个目的寄存器，可以通过任何指令发起传输。在模式 0，当 RI=0 且 REN=1 时启动接收。在其它模式，在 REN=1 时，通过收到起始位启动接收。

$$\text{Baud Rate (for Mode 3)} = \frac{2^{\text{SMOD}}}{32} \times \text{(Timer-1 overflow rate)}$$

12.1. 帧错误检测

开启帧错误检测功能后，UART 会在通讯中检测是否丢失停止位，如果丢失一个停止位，就设置 SCON 寄存器的 FE 标志位。FE 标志位和 SM0 标志位共享 SCON.7，SMOD0 标志位(PCON.6)决定 SCON.7 究竟代表哪个标志，如果 SMOD0 位(PCON.6)置位则 SCON.7 就是 FE 标志，SMOD0 位清零则 SCON.7 就是 SM0 标志。当 SCON.7 代表 FE 时，只能软件清零。参考下图。

图 123-1 UART 帧错误检测



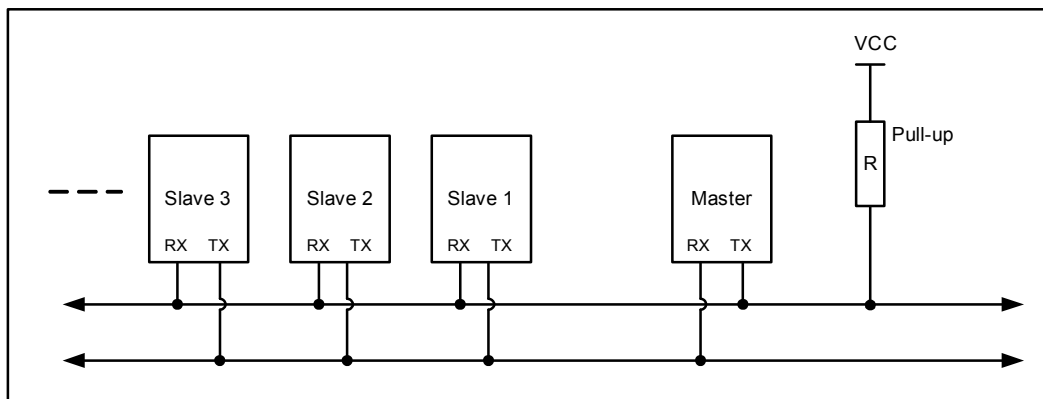
12.2. 多处理器通讯

模式 2 和 3 在用作多处理器通讯时有特殊的规定。在这两种模式，接收 9 个数据位。第 9 个数据位存入 RB8，接着进来一个停止位。端口可以编程为：在 RB8=1 时，当收到停止位后，串口中断将激活。这种特征通过设置 SM2 位(在 SCON 寄存器中)来使能。这种方式用于多处理器系统如下：

当主处理器想传送一个数据块到多个从机中的某一个时，首先传送想要传送的目标地址标识符的地址。地址字节与数据字节的区别在于，在地址字节中第 9 位为 1，数据字节中为 0。当 SM2=1 时，收到一个数据字节将不会产生中断。然而一个地址字节将引发所有从机中断。因而所有的从机可以检测收到的字节是否是自己的地址。从机地址将清除 SM2 位并准备好接收即将进来的所有数据。从机地址不匹配的将保持 SM2 置位，并继续他们的工作，忽略进来的数据字节。

SM2 在模式 0 和模式 1 没有影响，但是可以用来检测停止位的有效性。在接收模式 1 中，如果 SM2=1，除非收到一个有效的停止位否则接收中断不会被激活。

图 123-2 UART 多处理器通讯



12.3. 自动地址识别

自动地址识别通过硬件比较可以让 UART 识别串行码流中的地址部分，该功能免去了使用软件识别时需要大量代码的麻烦。该功能通过设定 SCON 的 SM2 位来开启。

在 9 位数据 UART 模式下，即模式 2 和模式 3，收到特定地址或广播地址时自动置位接收中断(RI)标志，9 位模式的第 9 位信息为 1 表明接收的是一个地址而不是数据。自动地址识别功能请参考图 12-6。在 8 位模式，即模式 1 下，如果 SM2 置位并且在 8 位地址与给定地址或广播地址核对一致后收到有效停止位则 RI 置位。模式 0 是移位寄存器模式，SM2 被忽略。

使用自动地址识别功能可以让一个主机选择性的同一个或多个从机进行通讯，所有从机可以使用广播地址接收信息。增加了 SADDR 从机地址寄存器和 SADEN 地址掩码寄存器。

SADEN 用来定义 SADDR 中的那些位是“无关紧要”的，SADEN 掩码和 SADDR 寄存器进行逻辑与来定义供主机寻址从机的“给定”地址，该地址让多个从机进行排他性的识别。

下面的实例帮助理解这个方案的通用性：

This document information is the intellectual property of Megawin Technology.

© Megawin Technology Co., Ltd. 2009 All rights reserved.

QP-7300-03D

33/68

从机 0	从机 1
SADDR = 1100 0000	SADDR = 1100 0000
SADEN = 1111 1101	SADEN = 1111 1110
地址 = 1100 00X0	地址 = 1100 000X

上面的例子中 SADDR 是相同的值，而使用 SADEN 数据来区分两个从机。

从机 0 要求第 0 位必须为 0，并忽略第 1 位的值；从机 1 要求第 1 位必须为 0，并忽略第 0 位的值。从机 0 的唯一地址是 1100 0010，而从机 1 的唯一地址是 1100 0001，地址 1100 0000 是可以同时寻找到从机 0 和从机 1 的。

下面一个更为复杂的系统可以寻址到从机 1 和从机 2，而不会寻址到从机 0：

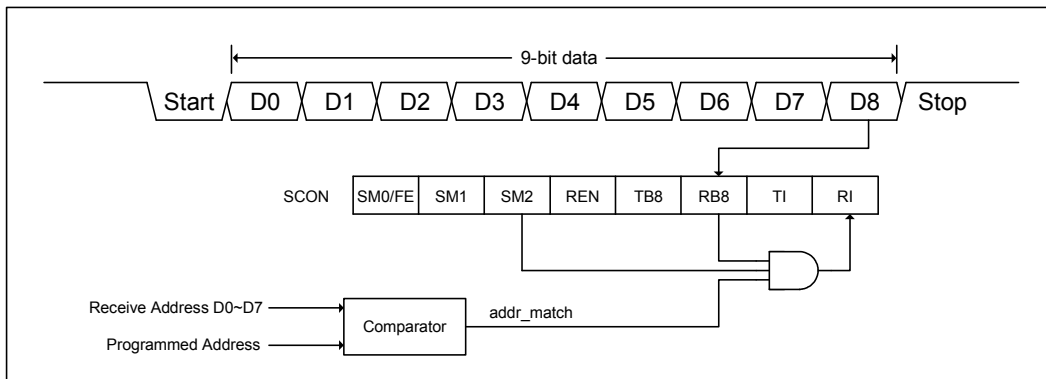
从机 0	从机 1	从机 2
SADDR = 1110 0000	SADDR = 1110 0000	SADDR = 1110 0000
SADEN = 1111 1001	SADEN = 1111 1010	SADEN = 1111 1100
地址 = 1110 0XX0	地址 = 1110 0X0X	地址 = 1110 00XX

上面的例子中，3 个从机的低 3 位地址不一样，从机 0 要求第 0 位必须为 0，1110 0110 可以唯一寻址从机 0；从机 1 要求第 1 位必须为 0，1110 0101 可以唯一寻址从机 1；从机 2 要求第 2 位必须为 0，它的唯一地址是 1110 0011。为了寻址到从机 0 和从机 1 而不会寻址到从机 2，可以使用地址 1110 0100，因为这个地址第 2 位是 1。

每个从机的广播地址 SADDR 和 SADEN 的逻辑或，0 按不需关心处理。大部分情况下，使用 FF 作为广播地址。

复位后，SADDR（SFR 地址 0A9H）和 SADEN（SFR 地址 0B9H）值均为 0，这样可以接收所有地址的信息，也就有效的禁用了自动地址识别模式，从而使该处理器运行于标准 80C51 的 UART 下。

图 123-3 自动地址识别



注：(1) 收到匹配地址后(addr_match=1), 清 SM2 以接收数据字节
 (2) 收完全部数据字节后, 置 SM2 为 1 以等待下一个地址

12.4. 串口寄存器

四个操作模式除了波特率设置不同外其它都与标准 8051 相同。PCON 和 AUXR 二个寄存器与波特率设置有关。

SCON: 串口控制寄存器

SFR 地址 = 0x98

SFR 页 = 全部

复位值 = 0000-0000

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

Bit 7: FE, 帧错误位。当接收器检测到一个无效的停止位时这位置 1。当收到有效的帧时 FE 不会自动清除，但是可以用软件清除。SMOD0 位(在 PCON 寄存器)必须置 1 来使能访问 FE 位。

Bit 7: SM0, 串行口模式位 0(SMOD0 必须为 0 来访问 SM0 位)

Bit 6: SM1, 串行口模式位 1。

SM0	SM1	模式	描述	波特率
0	0	0	移位寄存器	SYSCCLK/12 或 /2
0	1	1	8 位 UART	可变的
1	0	2	9 位 UART	SYSCCLK/64, /32, /16 或 /8
1	1	3	9 位 UART	可变的

Bit 5: SM2, 在模式 2 和 3 时使能地址自动识别，如果 SM2=1 那么 RI 将不能设置，除非接收到的第 9 位数据(RB8)为 1，指示是一个地址，并且接收到的字节是本机地址或者是一个广播地址；在模式 1，如果 SM2=1 那么 RI 将不能被激活除非收到一个有效的停止位，并且接收到的字节是本机地址或者是一个广播地址；在模式 0，SM2 可以为 0。

Bit 4: REN, 允许接收位。通过软件置 1 接收使能，软件清零将禁止接收。

Bit 3: TB8, 在模式 2 和 3 时第 9 位数据被传送, 根据需要通过软件置位或清零。

Bit 2: RB8, 在模式 2 和 3 时收到的第 9 位数据。在模式 1, 如果 SM2=0, RB8 是收到数据的停止位。在模式 0, RB8 没有使用。

Bit 1: TI, 发送中断标志。在模式 0 时, 在第 8 位个数据位时序后由硬件置位。其它模式中, 在发送停止位之初由硬件置位, 任何模式中必须由软件清除。

Bit 0: RI, 接收中断标志。在模式 0 时, 在第 8 位个数据位时序后由硬件置位。其它模式中, 在接收停止位的中间时刻由硬件置位。任何模式中必须由软件清除。

SBUF: 串口缓冲寄存器

SFR 地址 = 0x99

SFR 页 = 全部

复位值 = XXXX-XXXX

7	6	5	4	3	2	1	0
SBUF[7]	SBUF[6]	SBUF[5]	SBUF[4]	SBUF[3]	SBUF[2]	SBUF[1]	SBUF[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7~0: 在发送和接收时作缓冲寄存器

SADDR: 从机地址寄存器

SFR 地址 = 0xA9

SFR 页 = 全部

复位值 = 0000-0000

7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SADEN: 从机地址屏蔽寄存器

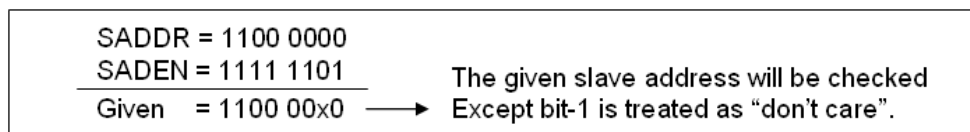
SFR 地址 = 0xB9

SFR 页 = 全部

复位值 = 0000-0000

7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

当地址自动识别功能启用后, 可用 SADDR 和 SADEN 来预置地址, 事实上, SADEN 是 SADDR 的“屏蔽”寄存器, 如下图所示



每个从对象的广播地址为 SADDR 和 SADEN 进行逻辑“或”的结果, 结果中为“0”的位将被忽略。在系统复位后, SADDR 和 SADEN 都被初始化为 0, 从而忽略“Given”地址的全部地址位和“广播”地址的全部地址位而导致自动地址识别功能无效。

PCON: 电源控制寄存器

SFR 地址 = 0x87

SFR 页 = 全部

复位值 = 00X1-0000

7	6	5	4	3	2	1	0
SMOD1	SMOD0	LVF	POF	GF1	GF0	PD	IDL
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit 7: SMOD1, 双倍波特率控制位

0: 禁止 UART 双倍波特率

1: 使能 UART 双倍波特率(模式 1, 2, 或 3.)

Bit 6: SMOD0, 帧错误选则

This document information is the intellectual property of Megawin Technology.

© Megawin Technology Co., Ltd. 2009 All rights reserved.

QP-7300-03D

0: SCON.7 作 SM0 功能
 1: SCON.7 作 FE 功能。

AUXR: 辅助寄存器

SFR 地址 = 0x8E

SFR 页 = 全部

复位值 = 0000-00XX

7	6	5	4	3	2	1	0
T0X12	T1X12	URM0X6	EADCI	ESPI	ENLVFI		
R/W	R/W	R/W	R/W	R	R/W	R	R

Bit 7: T0X12: 当 C/T=0 时, 定时器 0 的时钟源选择。

0: 选择 SYSCLK/12.

1: 选择 SYSCLK 作时钟源

Bit 6: T1X12, 当 C/T=0 时, 定时器 1 时钟源选择

0: 选择 SYSCLK/12.

1: 选择 SYSCLK 作时钟源

Bit 5: URM0X6, 串口模式 0 波特率选择

0: 选择 SYSCLK/12 作 UART 模式 0 波特率

1: 选择 SYSCLK/2 作 UART 模式 0 波特率

Bit 4: EADCI, ADC 中断使能

0: 禁止 ADC 中断使能

1: 使能 ADC 中断使能

Bit 3: ESPI, SPI 中断使能

0: 禁止 SPI 中断使能

1: 使能 SPI 中断使能

Bit 2: ENLVFI, 低电平检测中断使能

0: 禁止低电平检测中断

1: 使能低电平检测中断

Bit 1~0:保留

13. 可编程计数器阵列(PCA)

MA801带有一个可编程计数器阵列(PCA)，该功能与标准定时/计数器相比以更少的CPU占用提供了更多的定时能力。它的优点包括减少了软件复杂度并提高了精度。

13.1. PCA 概述

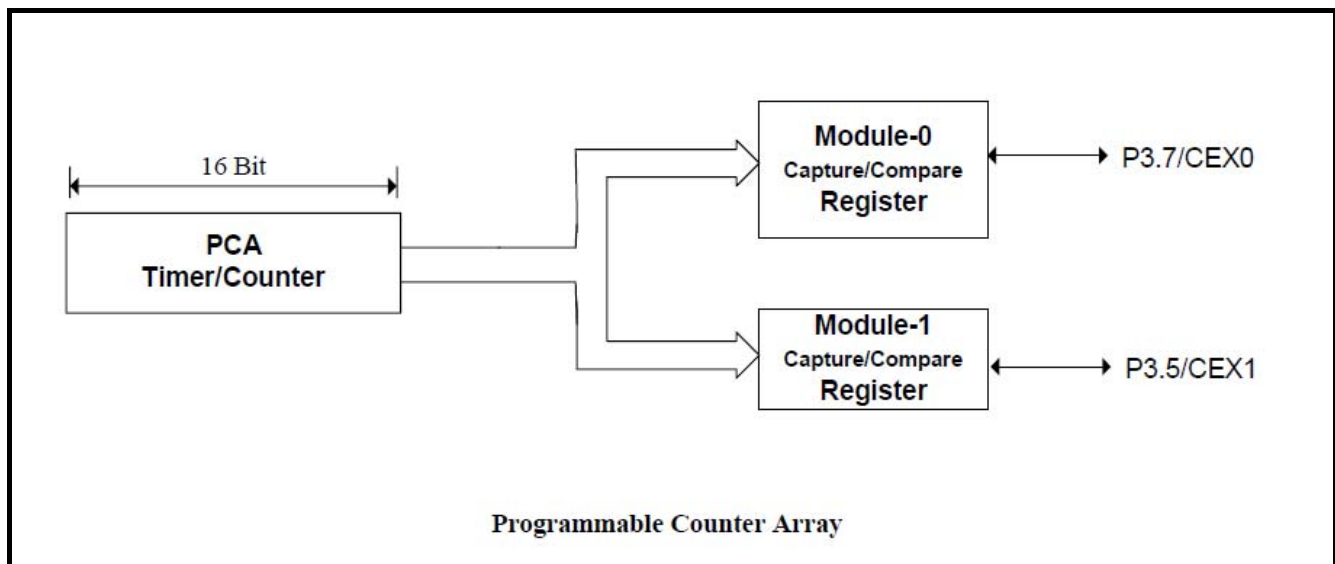
PCA由一个特定定时/计数器作为一个比较/捕获模块的时基，图 13-1 显示了PCA的功能方框图。需要注意的是PCA定时器和模块都是16位的。如果一个外部事件同一个模块关联，那末该功能就和相应的端口1引脚共享。若某模块没有使用端口引脚，这个引脚还可以用作标准I/O。

比较/捕获模块中的每一个都可以编程为如下任意模式：

- 上升和/或下降沿捕获
- 软件定时器
- 高速输出
- 脉宽调制输出

所有这些模式将在后面的章节进行详细讨论。这里，让我们先看看如何设置PCA时钟和模块。

图13-1. PCA 方框图



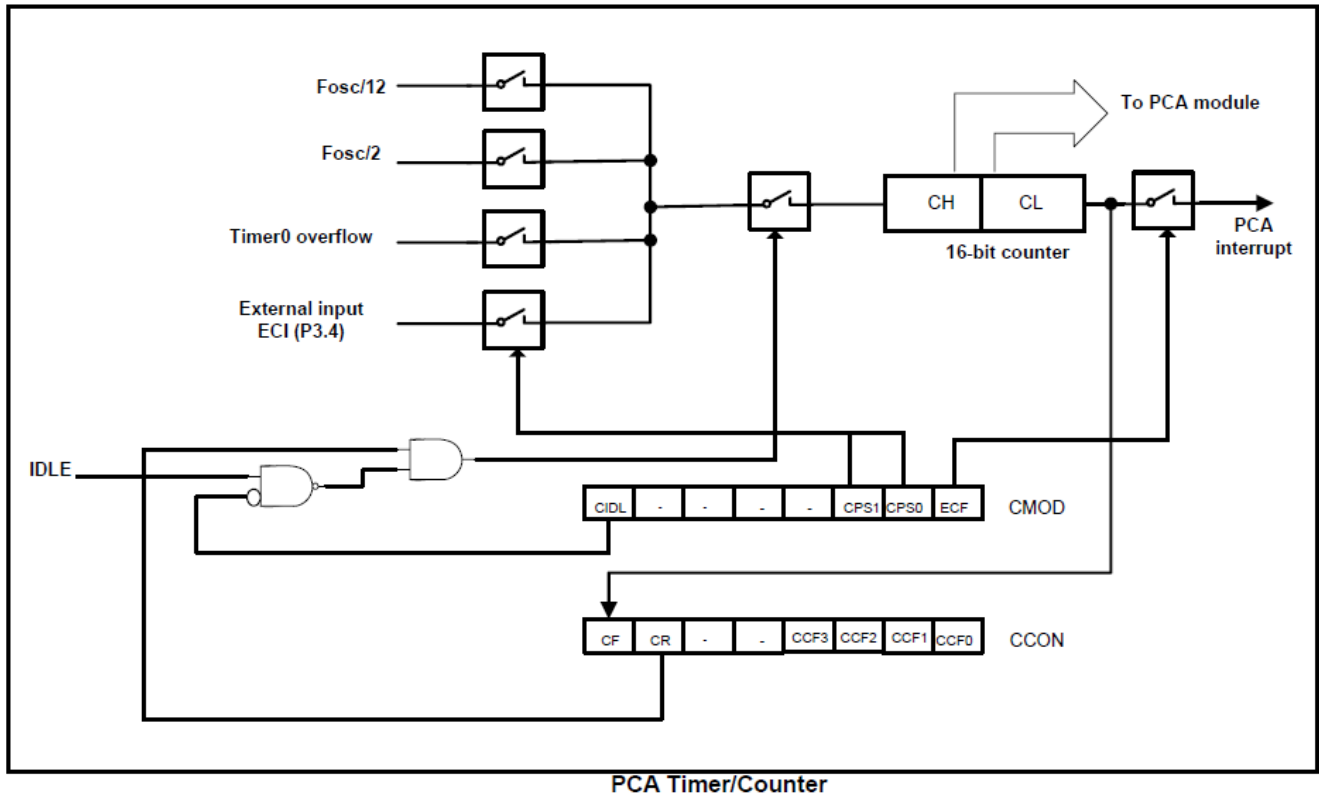
13.2. PCA定时/计数器

PCA的定时/计数器由一个可以自由运行的16位定时器组成，如图13-2所示分为CH和CL高低两部分，它是所有模块的共有时基，它的时钟输入可以从以下来源选取：

- 1/12 系统时钟频率,
- 1/2 系统时钟频率,
- 定时器0溢出，可以让低频时钟源输入到PCA定时器.
- 外部时钟输入，ECI (P1.1) 引脚的1-0反转.

特殊功能寄存器 CMOD 包含了计数脉冲选择位 (CPS1 和 CPS0) 来指定PCA定时器时钟源。这个寄存器也包括了 ECF位来使能计数器溢出中断。此外，用户可以在待机模式下设置计数器待机位 (CIDL)，来关闭PCA定时器，这样可以进一步降低待机模式下的功耗。n=0,1

图 13-2. PCA 定时/计数器



图中，Fosc是系统时钟。

CMOD (地址=D9H, PCA计数器模式寄存器)

7	6	5	4	3	2	1	0
CIDL	-	-	-	-	CPS1	CPS0	ECF

CIDL: PCA 计数器待机控制。
 CIDL=0 让PCA计数器在待机模式下继续运行。
 CIDL=1 待机模式下关闭PCA计数器。

CPS1-CPS0: PCA计数器时钟源选择位。
 0 0 内部时钟, Fosc/12 (Fosc代表系统时钟)
 0 1 内部时钟, Fosc/2
 1 0 定时器 0 溢出
 1 1 ECI引脚输入的外部时钟源。

ECF: 使能PCA计数器溢出中断。
 ECF=1 当CF位 (CCON寄存器中) 置位时使能中断。

如下所示的CCON寄存器包含PCA运行控制位和PCA定时器与每个模块的标志。要运行PCA，CR为 (CCON.6) 必须软件置位，要关闭PCA，可以清除该位。PCA计数器溢出时，CF (CCON.7)置位，并且若CMOD寄存器的ECF为置位，还会产生一个中断，CF位只能软件清零。CCF0是模块0的相应中断标志位，当发生一个匹配或捕获事件时，硬件置位，这些位也必须软件清零。PCA中断系统如图 13-2所示。

CCON (地址=D8H, PCA控制寄存器)

7	6	5	4	3	2	1	0
CF	CR					CCF1	CCF0

CF: PCA 计数溢出标志。溢出时硬件置位，CF标志在CMOD寄存器的ECF位置位时会产生一个中断，CF可以硬件或软件置位，但只能软件清零。

CR: PCA 计数控制位。

软件设置来启停PCA计数器。1：启动PCA计数器，0：关闭PCA计数器。

CCF1: PCA模块1中断标志。
发生一个匹配或捕获时硬件置位，必须软件清零。

CCF0: PCA模块0中断标志。
发生一个匹配或捕获时硬件置位，必须软件清零。

图 14-3. PCA 中断系统

13.3. 比较/捕获模块

比较/捕获模块中的一个模式寄存器，叫做CCAPMn，来选择其工作模式。ECCFn位控制当中断标志置位时模块的中断开启/关闭。

CCAPM n, n=0 (地址=DAH, PCA模块比较/捕获寄存器) n=0,1

7	6	5	4	3	2	1	0
-	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn

ECOMn: 比较器使能位。 ECOMn=1 使能比较器功能。

CAPPn: 捕获上升沿。 CAPPn=1 使能上升沿捕获

CAPNn: 捕获下降沿。 CAPNn=1 使能下降沿捕获

MATn: 匹配控制。 MATn=1, PCA计数器同相应模块的比较/捕获寄存器匹配时设置CCON寄存器的CCFn位。

TOGn: 翻转控制。 TOGn=1, PCA计数器同相应模块的比较/捕获寄存器匹配时CEXn引脚电平翻转一次。

PWMn: PWM控制。 PWMn=1使能CEXn引脚用作脉宽调制输出。

ECCFn: 使能CCFn中断。使能CCON寄存器中的比较/捕获标志CCFn中断。

注: CAPNn (CCAPMn.4)位和CAPPn (CCAPMn.5)位决定了捕获发生时信号脉冲沿,若这两位同时设置,则上下沿都会发生捕获。

每一个模块都有一对8位比较/捕获寄存器 (CCAPnH, CCAPnL) 与其相关联。这些寄存器用来保存捕获事件发生时或比较事件发生时的值。当某个模块用作PWM模式时,除了上面两个寄存器外,还有一个扩展的寄存器PCAPWMn被用来扩展输出占空比的范围,扩展的范围从0%到100%,步距是1/256。

PCAPWMn, (PWM 模式辅助寄存器, n=0,1)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	ECAPnH	ECAPnL

ECAPnH: 扩展的第9位 (最高位扩展), 用在PWM模式下, 与CCAPnH联用并成为其第9位。

ECAPnL: 扩展的第9位 (最高位扩展), 用在PWM模式下, 与CCAPnL联用并成为其第9位。

13.4. PCA模式设置

表13-1显示了不同PCA功能对应的CCAPMn寄存器设置。

表13-1. PCA模块工作模式

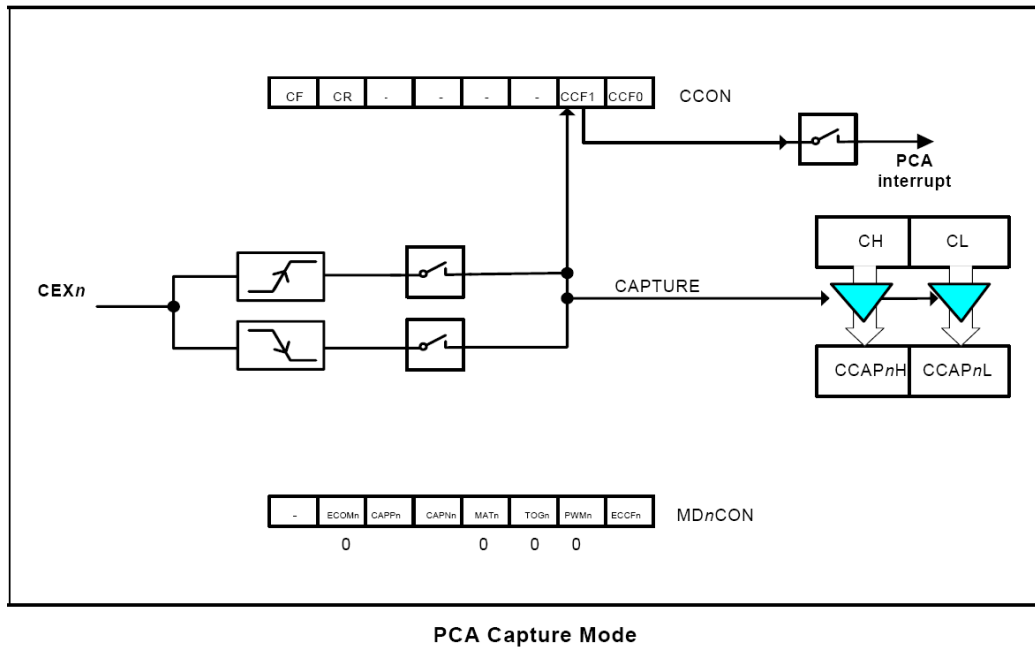
ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	模块功能
0	0	0	0	0	0	0	无操作
X	1	0	0	0	0	X	16位CEXn引脚上升沿触发捕获模式
X	0	1	0	0	0	X	16位CEXn引脚下降沿触发捕获模式
X	1	1	0	0	0	X	16位CEXn引脚跳变触发捕获模式
1	0	0	1	0	0	X	16位软件定时器

1	0	0	1	1	0	X	16位高速输出
1	0	0	0	0	1	0	8位脉宽调制器(PWM)

13.4.1. 捕获模式

要让某一PCA模块工作与捕获模式，相应的CAPN和/或CAPP位必须置位。外部CEX输入会在每次跳变是采样，当有效跳变发生时，PCA硬件会将PCA计数器寄存器值装入模块的捕获寄存器（CH放入CCAPnH，CL放入CCAPnL）。若模块的CCFn和ECCFn标志置位，会产生一个中断。

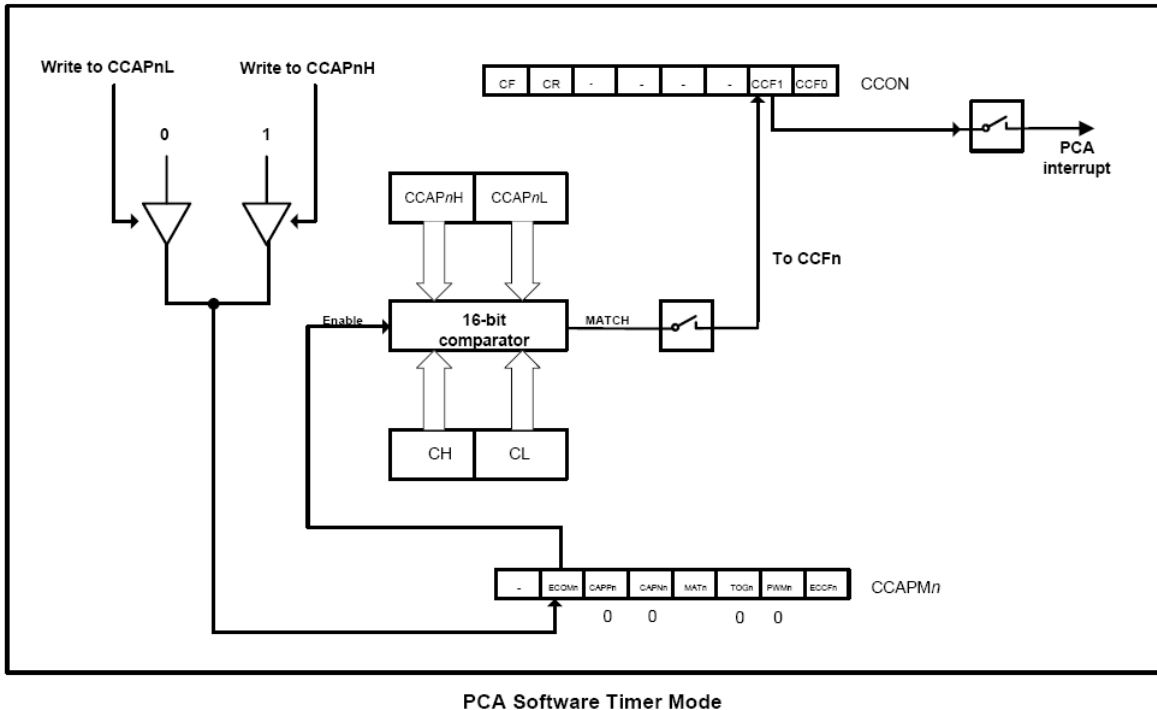
图 13-4. PCA捕获模式



13.4.2. 16 位软件定时器模式

PCA模块可以通过设置CCAPMn寄存器的ECOM位和MAT位来作为一个软件定时器使用，PCA定时器与模块的捕获寄存器值进行比较，若相等则当CCFn和ECCFn位设置时会产生一个中断信号。

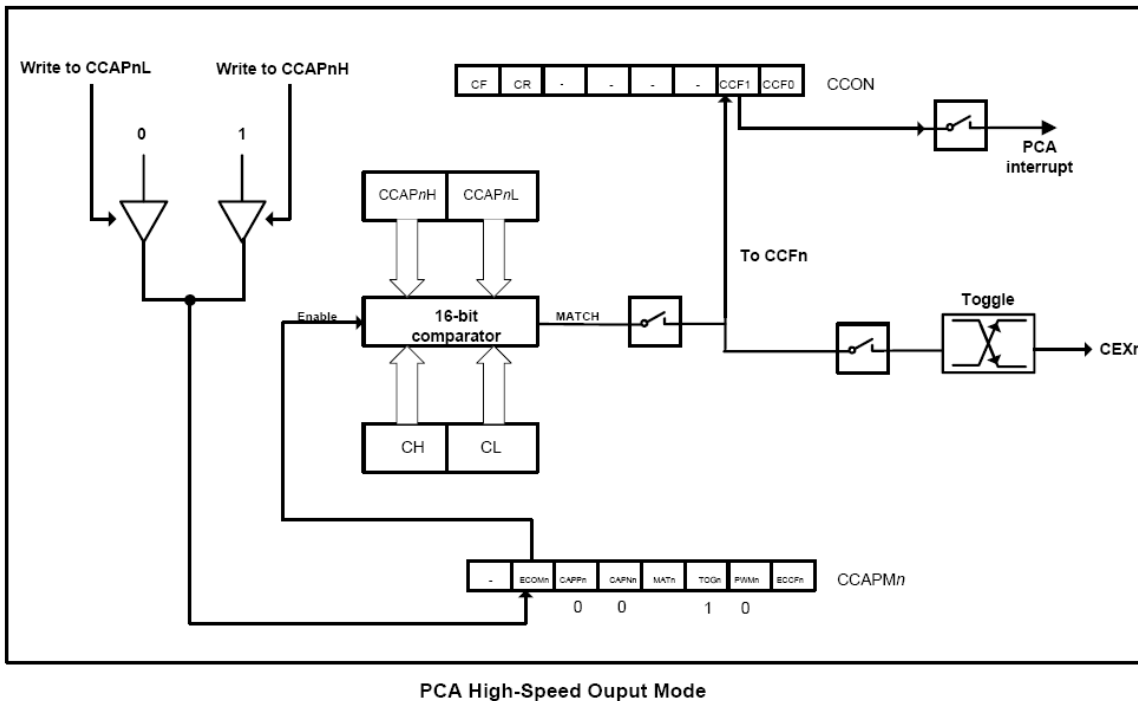
图 13-5. PCA 软件定时器模式



13.4.3. 高速输出模式

这种模式下，每当PCA计数器与模块捕获寄存器值相等时，CEX的输出就翻转一次。为激活这种模式，CCAPMn寄存器的TOG, MAT 和 ECOM 位必须都置为1。

图 13-6. PCA 高速输出模式



13.4.4. PWM 模式

所有PCA模块都可用作PWM输出，输出频率取决于PCA定时器的时钟源，所有的模块都有相同的输出频率，因为它们共享PCA定时器。

占空比取决于模块捕获寄存器CCAPnL 与扩展的第9位ECAPnL的值。当9位数据{0,[CL]}值小于{ ECAPnL,

[CCAPnL] }组成9位数据时，输出低电平，相等或大于时输出高电平。

当CL从0xFF到0x00溢出时，{ ECAPnL, [CCAPnL] } 的值使用 { ECAPnH,[CCAPnH] }的值重载，这样可以允许无异常脉冲的更新PWM。模块的CCAPMn 寄存器PWMn 和 ECOMn 位必须置位以使能PWM模式。

使用9位比较，输出的占空比可以真正实现从0%到100%可调。占空比计算公式如下：

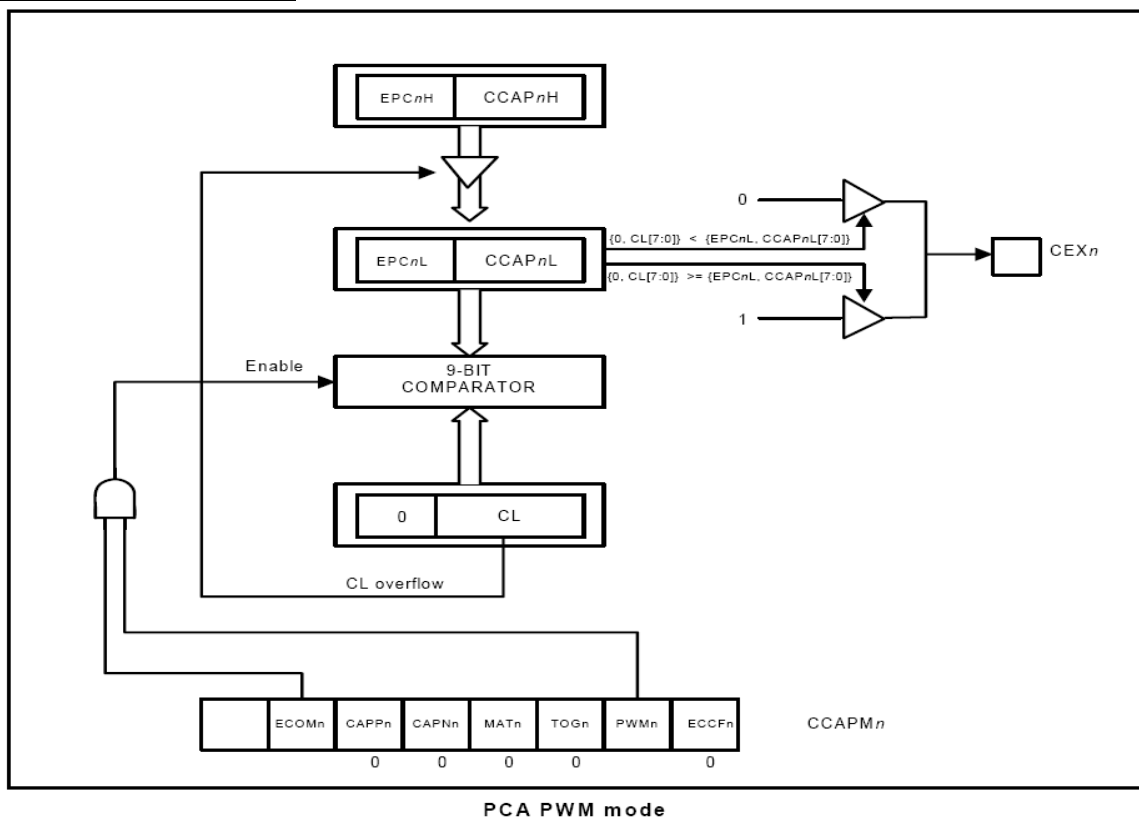
$$\text{占空比} = 1 - \{ ECAPnH, [CCAPnH] \} / 256.$$

这里, [CCAPnH] 是CCAPnH 寄存器的值， ECAPnH (PCAPWMn 寄存器的第1位) 是1位值。所以， { ECAPnH, [CCAPnH] } 组成了9位比较器用的9位值。

例如,

- 若 ECAPnH=0 且 CCAPnH=0x00 (即9位值, 0x000), 占空比是100%.
- 若 ECAPnH=0 且 CCAPnH=0x40 (即9位值, 0x040), 占空比是75%.
- 若 ECAPnH=0 且 CCAPnH=0xC0 (即9位值, 0x0C0), 占空比是25%.
- 若 ECAPnH=1 且 CCAPnH=0x00 (即9位值, 0x100), 占空比是0%.

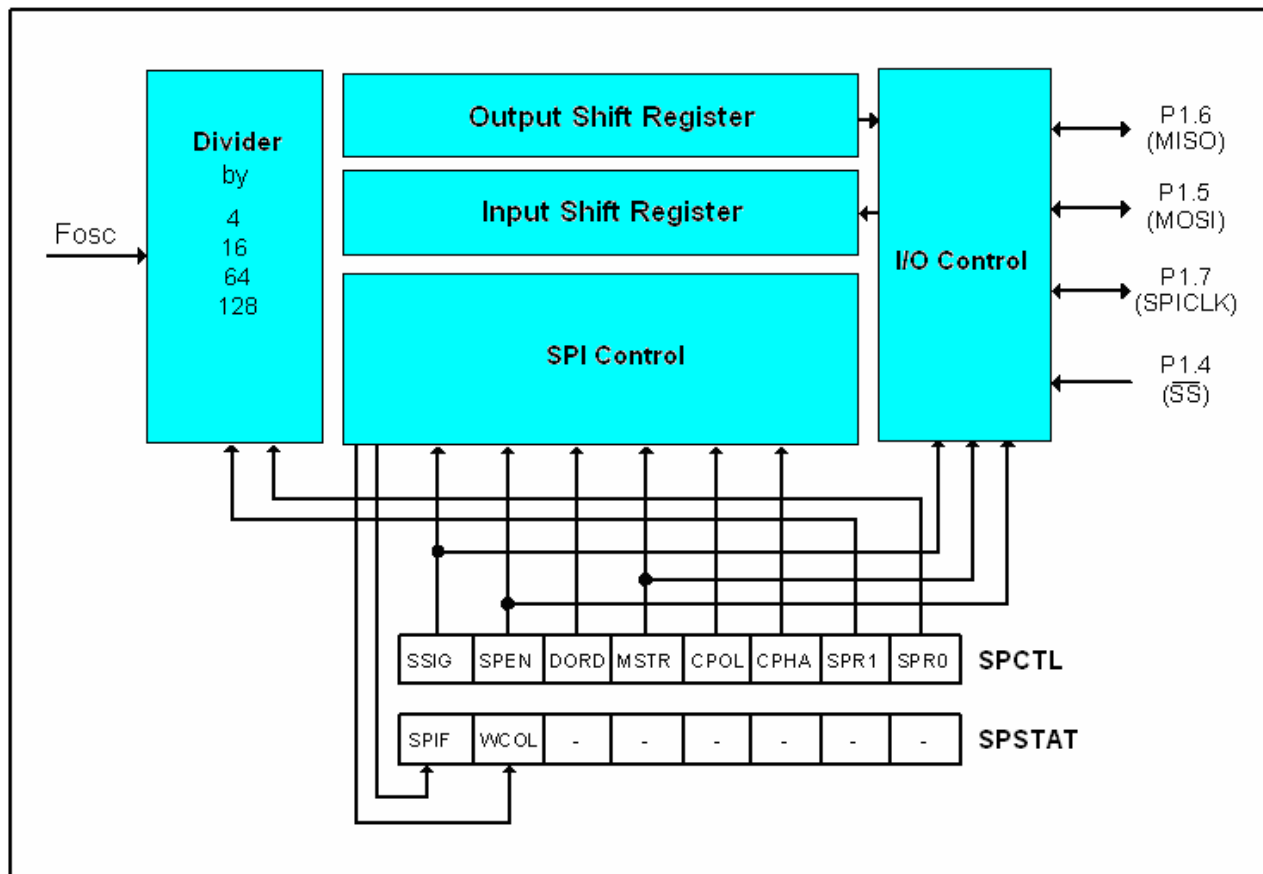
图 13-7. PCA PWM 模式



14. 串行外设接口(SPI)

MA801提供了一个高速串行通讯接口（SPI）。SPI接口是一种全双工、高速同步通讯总线，有两种操作模式：主机模式和从机模式。无论哪种模式，12MHz系统时钟时支持高达3Mbps的通讯速度。MA801的SPI状态寄存器（SPSTAT）有一个传送完成标志（SPIF）和写冲突标志（WCOL）。

图 14-1. SPI 框图



SPI接口有4个引脚：MISO (P1.6), MOSI (P1.5), SPICLK (P1.7) 和 \overline{SS} (P1.4):

- SPICLK, MOSI 和 MISO 通常将两个或多个SPI设备连接在一起。数据从主机到从机使用MOSI 引脚 (Master Out / Slave In主出从入)，从从机到主机使用MISO 引脚 (Master In / Slave Out主入从出)。SPICLK 信号在主机模式时输出，从机模式时输入。若SPI接口禁用，即 SPEN (SPCTL.6) = 0，这些引脚可以作为普通I/O口使用。

- \overline{SS} 是从机选择端。典型配置中，SPI主机可以使用其某个端口选择某一个SPI设备作为当前从机，一个SPI 从机设备使用它的 \overline{SS} 引脚确定自己是否被选中。下面条件下 \overline{SS} 被忽略:

- 若 SPI系统被禁用，即 SPEN (SPCTL.6) = 0 (复位值).
- 若SPI作为主机运行，即 MSTR (SPCTL.4) = 1, 且 P1.4 (\overline{SS}) 被配置成输出。
- 若 \overline{SS} 被设置成忽略，即 SSIG (SPCTL.7) = 1, 这个端口作为普通I/O使用。

注意，即使SPI被配置成主机运行(MSTR=1)，它仍然可以被 \overline{SS} 引脚的低电平拉成从机(若 SSIG=0)，一旦发生这种情况，SPIF 位(SPSTAT.7)置位。(参见 14.5节： \overline{SS} 引脚的模式改变)

下面是SPI操作相关寄存器:

SPCTL (地址=85H, SPI控制寄存器, 复位值=0000,0100B)

7	6	5	4	3	2	1	0
SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0

SSIG: 忽略/SS

若SSIG=1, MSTR位决定该设备是主机还是从机。

若SSIG=0, /SS引脚决定该设备是主机还是从机。

SPEN: SPI使能

若SPEN=1, SPI功能打开。

若SPEN=0, SPI接口禁用, 所有SPI引脚可作为通用I/O口使用。

DORD: SPI 数据顺序

1: 传送数据时先传数据字节最低位。

0: 传送数据时先传数据字节最高位。

MSTR: 主机/从机模式选择
CPOL: SPI时钟极性选择

1: SPICLK待机是为高电平, SPICLK时钟脉冲前沿是下降沿, 而后沿是上升沿。

0: SPICLK待机是为低电平, SPICLK时钟脉冲前沿是上升沿, 而后沿是下降沿。

CPHA: SPI时钟相位选择

1: SPICLK脉冲前沿放数据, 后沿采样。

0: /SS引脚低电平 (SSIG=0)开始放数据并在SPICLK后沿改变数据. 数据在SPICLK的前沿采样。

(注: 若 SSIG=1, CPHA必须不为1, 否则就是为定义操作.)

SPR1-SPR0: SPI 时钟速率选择 (主机模式)

00: Fosc/4

01: Fosc/16

10: Fosc/64

11: Fosc/128 (Fosc是系统时钟.)

SPSTAT (地址=84H, SPI状态寄存器, 复位值=00xx,xxxxB)

7	6	5	4	3	2	1	0
SPIF	WCOL	-	-	-	-	-	-

SPIF: SPI 传输完成标志

当一次串行传输完成时, SPIF位置位, 同时若SPI中断允许, 会产生一个中断。若 /SS 引脚在主机模式下被拉低且SSIG=0, SPIF位也会置位以表明“模式改变”。SPIF标志通过软件在该位写入“1”来清零。

WCOL: SPI 写冲突标志

SPI数据寄存器SPDAT在数据传输过程中被写入时, WCOL置位。(参见 14.6节:写冲突). WCOL标志通过软件在该位写入“1”来清零。

SPDAT (地址=86H, SPI数据寄存器, 复位值=0000,0000B)

7	6	5	4	3	2	1	0
(MSB)							(LSB)

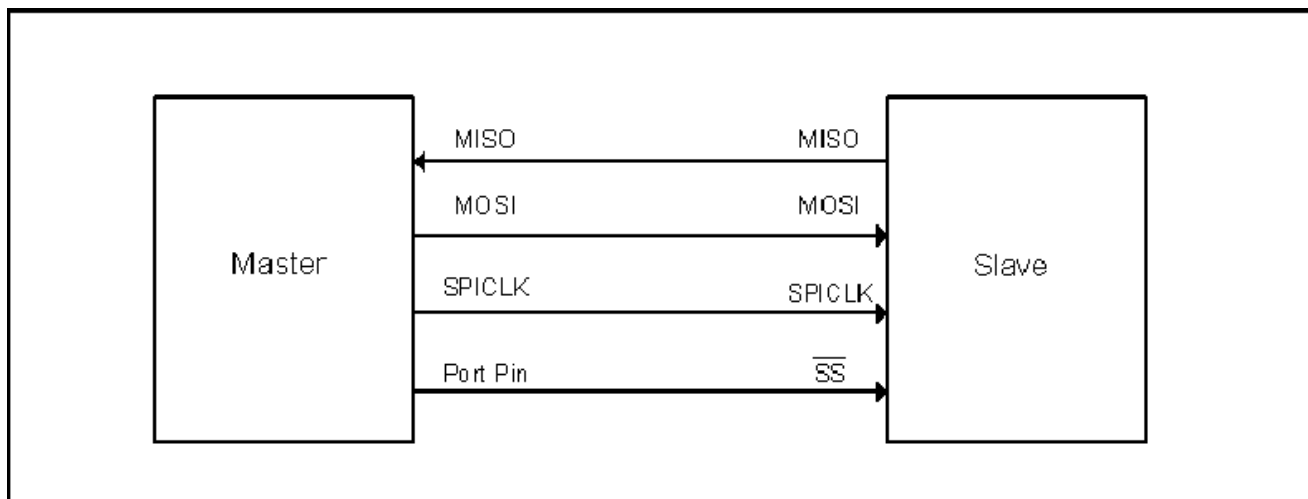
SPDAT有两个物理缓冲器供传输过程中写入和读取, 一个写入缓冲器, 一个读取缓冲器。

14.1. 典型 SPI 配置

14.1.1. 单主机和单从机

对于主机: 任何端口, 包括P1.4 (/SS), 都可以用来控制从机的/SS片选引脚。
 对于从机: SSIG 为 '0', /SS引脚决定该设备是否被选中。

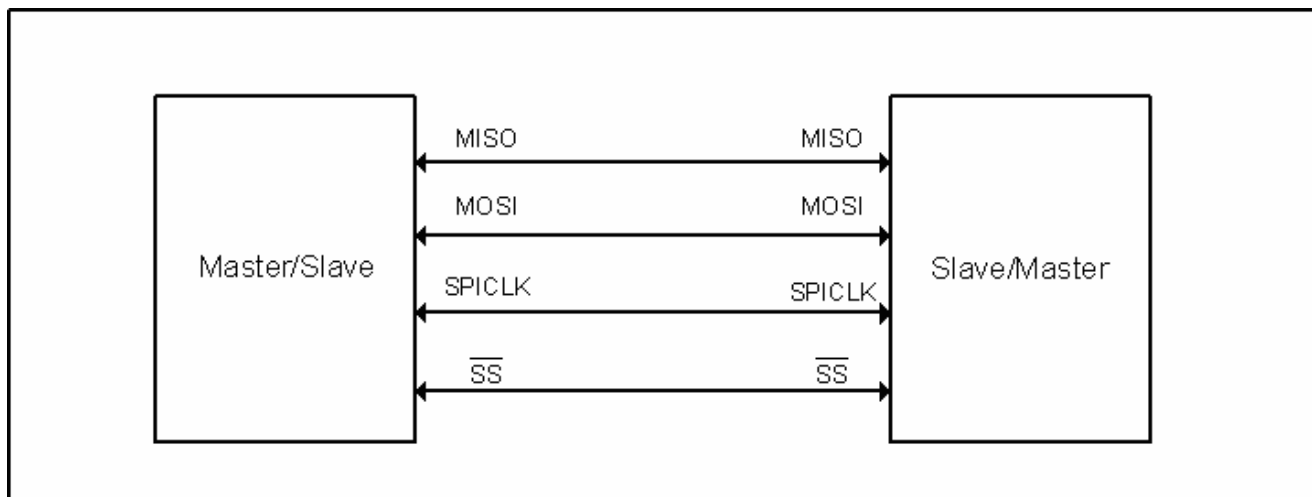
图 14-2. SPI 单主机配置



14.1.2. 双驱动器, 可以是主机或从机

两个彼此连接的设备, 均可成为主机或从机, 没有SPI操作时, 都可以被通过设置MSTR=1, SSIG=0, P1.4 (/SS)双向口配置成主机。任何一方要发起传输, 它可以配置P1.4位输出并强行拉低, 使另一个设备发生“被改成从机模式”事件。(参见 15.5节: /SS引脚模式改变)

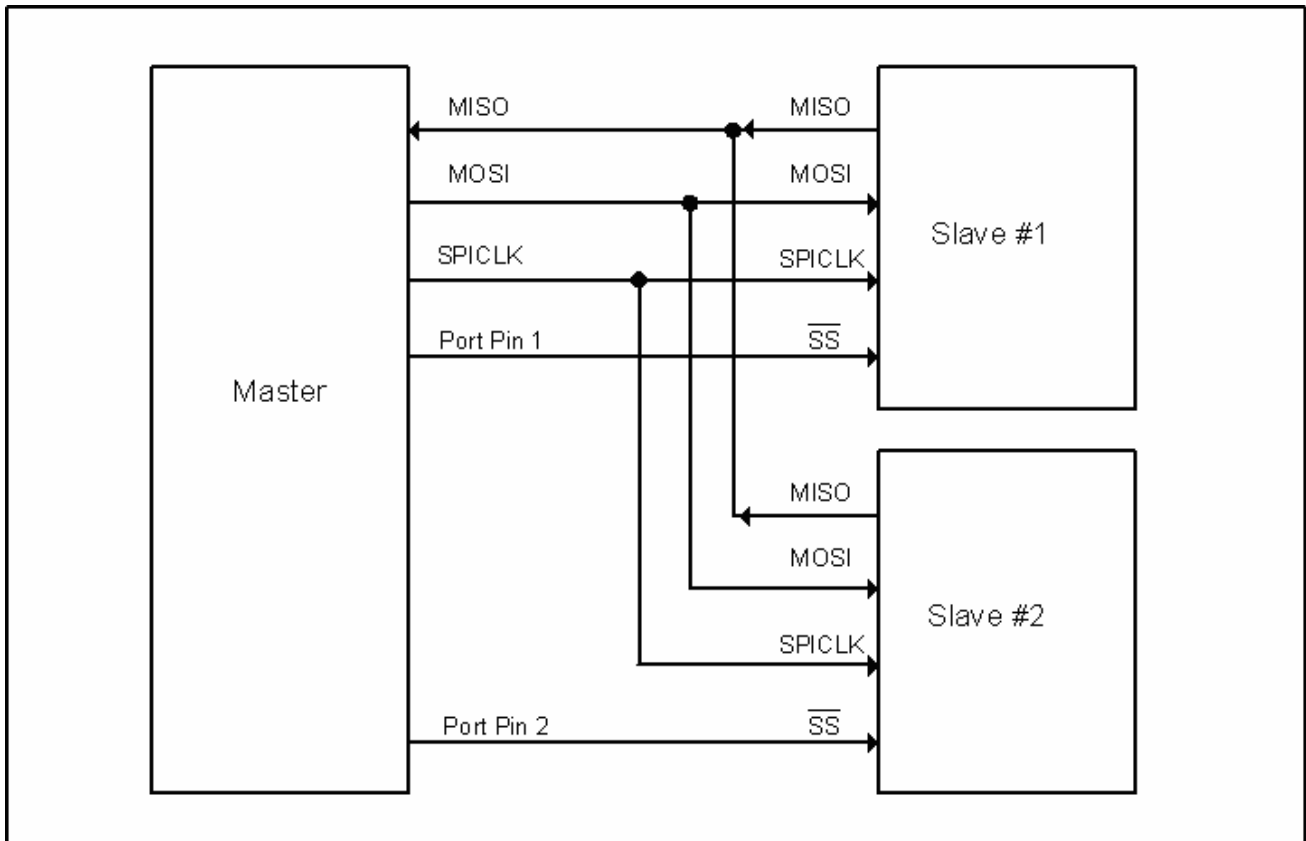
14-3. SPI 双驱动器, 可以是主机或从机配置



14.1.3. 单主机和多从机

对于主机: 任何端口, 包括P1.4 (/SS), 都可以用来控制从机的/SS片选引脚。
 对于所有从机: SSIG 为 '0', /SS引脚决定该设备是否被选中。

图 14-4. SPI 单主机和多从机配置



14.2. SPI 配置

表14-1 显示了主/从机模式配置及使用方法和传输方向。

表 14-1. SPI 主从机选择

SPEN (SPCTL.6)	SSIG (SPCTL.7)	/SS 引脚	MSTR (SPCTL.4)	模式	MISO 引脚	MOSI 引脚	SPICLK 引脚	备注
0	X	X	X	SPI 禁用	输入	输入	输入	P1.4~P1.7用作通用I/O
1	0	0	0	从机 (被选中)	输出	输入	输入	被选择为从机
1	0	1	0	从机 (未被选中)	高阻	输入	输入	未被选中
1	0	0	1 $\bar{0}$	从机 (通过模式改变)	输出	输入	输入	若/SS被拉低，MSTR被硬件自动清‘0’，模式被改为从机
1	0	1	1	主机 (待机)	输入	高阻	高阻	MOSI和SPICLK在主机待机时被置为高阻，以防止总线冲突。
				主机 (活动)		输出	输出	MOSI和SPICLK在主机活动时被上拉。
1	1	X	0	从机	输出	输入	输入	
1	1	X	1	主机	输入	输出	输出	

“X”表示“无需关心”。

14.3. 从机注意事项

当CPHA = 0时，SSIG必须为 0 且 /SS 引脚必须在每次串行字节传输前负跳变，传输结束恢复正常高电平。注意 SPDAT寄存器不能在/SS引脚低电平时写入；CPHA = 0, SSIG=1的操作是未定义的。

当CPHA =1时，SSIG可以为0或1。若SSIG=0，/SS引脚可以在每次成功传输之间保持低电平（可以一直拉低），这种格式有时非常适合单固定主从机配置应用。

14.4. 主机注意事项

SPI通讯中，传输总是有主机发起。若SPI使能(SPEN=1)并作为主机运行，写入SPI数据寄存器(SPDAT) 数据即可开始SPI时钟生成器和数据传输器，大约半个到1个SPI位时间后写入SPDAT的数据开始出现在MOSI线上。

在开始传输之前，主机通过拉低相应/SS引脚选择一个从机作为当前从机。写入SPDAT寄存器数据从主机MOSI引脚移出，同时从从机MISO移入主机MISO的数据也写入到主机的SPDAT寄存器中。

移出1字节后，SPI时钟发生器停止，置传输完成标志SPIF，若SPI中断使能则生成一个中断。主机CPU和从机CPU中的两个移位寄存器可以看成是一个分开的16位环形移位寄存器，数据从主机移到从机同时数据也从从机移到主机。这意味着，在一次传输过程中，主从机数据进行了交换。

14.5. /SS引脚的模式改变

若 SPEN=1, SSIG=0, MSTR=1 且 /SS pin=1, SPI使能在主机模式, 这种情况下, 其他主机可以将/SS引脚拉低来选择该设备为从机并开始发送数据过来。为避免总线冲突, 该SPI设备成为一个从机, MOSI 和SPICLK引脚被强制为输入端口, MISO成为输出端口, SPSTAT中SPIF标志置位, 若此时SPI中断使能, 则还会产生一个SPI中断。用户软件必须经常去检查MSTR位, 若该位被从机选择清零而用户又想要继续保持该SPI主机模式, 用户必须再次设置MSTR位, 否则, 将处于从机模式。

14.6. 数据冲突

SPI在发送方向是单缓冲的, 而在接收方向是双缓冲的。发送数据直到上一次数据发送完成后才能写入移位寄存器, 数据发送过程中写入数据寄存器就会使WCOL(SPSTAT.6) 置位来表明数据冲突。这种情况下, 正在发送的数据继续发送, 而刚写入数据寄存器造成冲突的数据就会丢失。

写冲突对于主从机都有可能发生, 对于主机, 这种现象并不多见, 因为主机控制着数据的传送; 然而对于从机, 由于没有控制权, 因此很可能发生。

对于数据接收, 接收的数据被传输到一个并行读数据缓冲器中, 以便于移位寄存器再能接收新的字节。然而, 接收的数据必须在下一个字节完全移入前从数据寄存器SPDAT读出, 否则前一个数据就会丢失。

WCOL使用软件向其位写入'1'来清零。

14.7. SPI 时钟频率选择

SPI时钟频率选择 (主机模式) 使用SPCTL寄存器的SPR1 和SPR0 位来设置, 如表14-2所示

表14-2. SPI 串行时钟速率

SPR1	SPR0	SPI时钟速率 @ Fosc=12MHz	Fosc 分频
0	0	3 MHz	4
0	1	750 KHz	16
1	0	187.5 KHz	64
1	1	93.75 KHz	128

这里, Fosc是系统时钟。

14.8. 数据模式

时钟相位(CPHA) 位可以让用户设定数据采样和改变时的时钟沿。时钟极性位,CPOL, 可以让用户设定时钟极性。下面图例显示了不同时钟相位、极性设置下SPI通讯时序。

图 14-5. SPI从机传送,CPHA=0

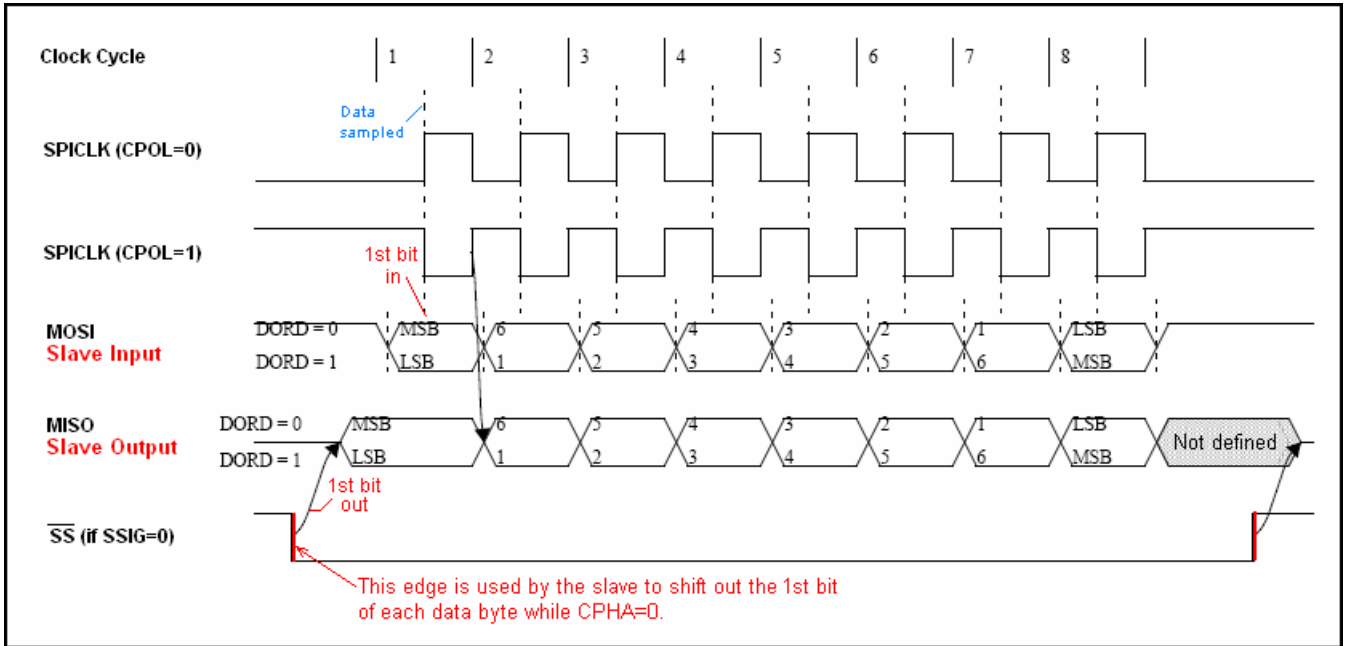
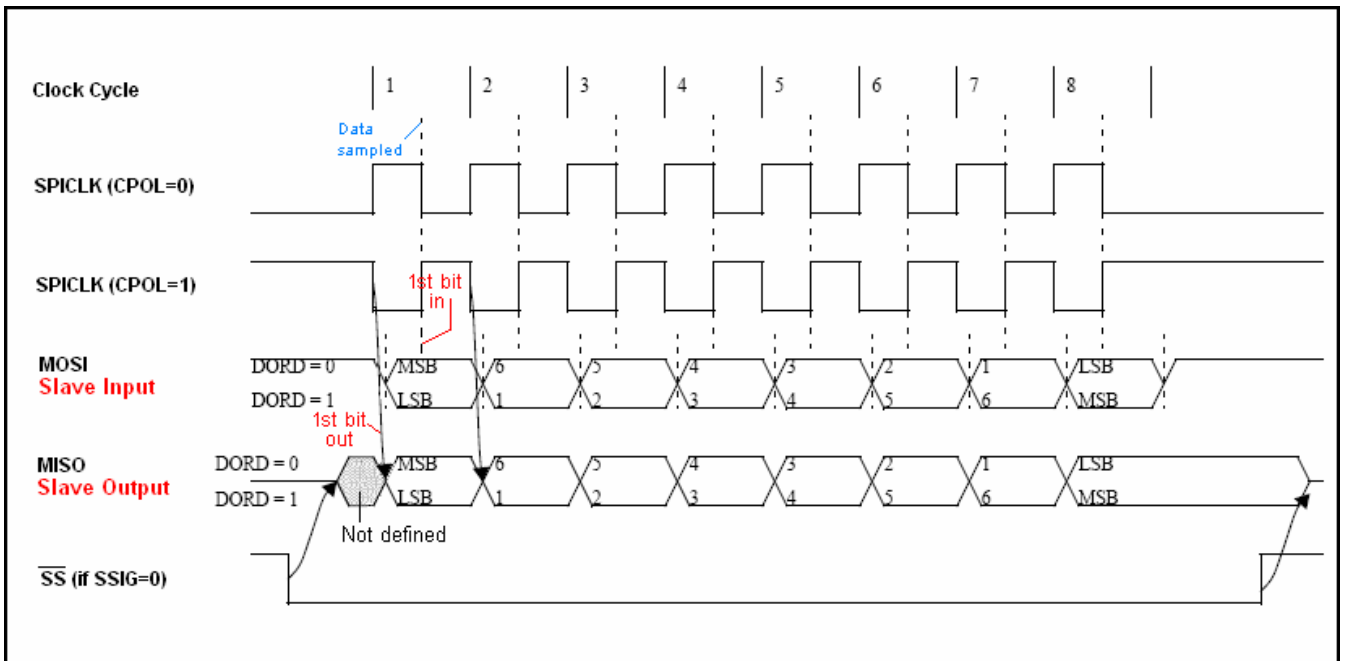


图 14-6. SPI从机传送,CPHA=1



This document information is the intellectual property of Megawin Technology.

© Megawin Technology Co., Ltd. 2009 All rights reserved.

QP-7300-03D

50/68

图 14-7. SPI主机传送, CPHA=0

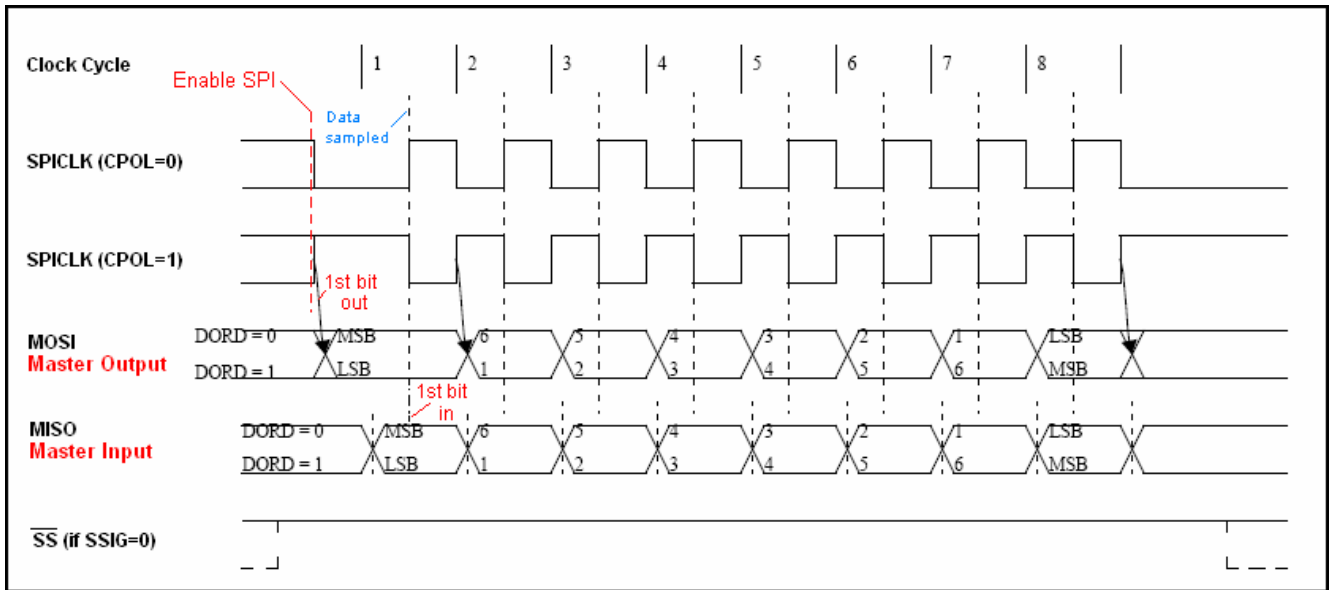
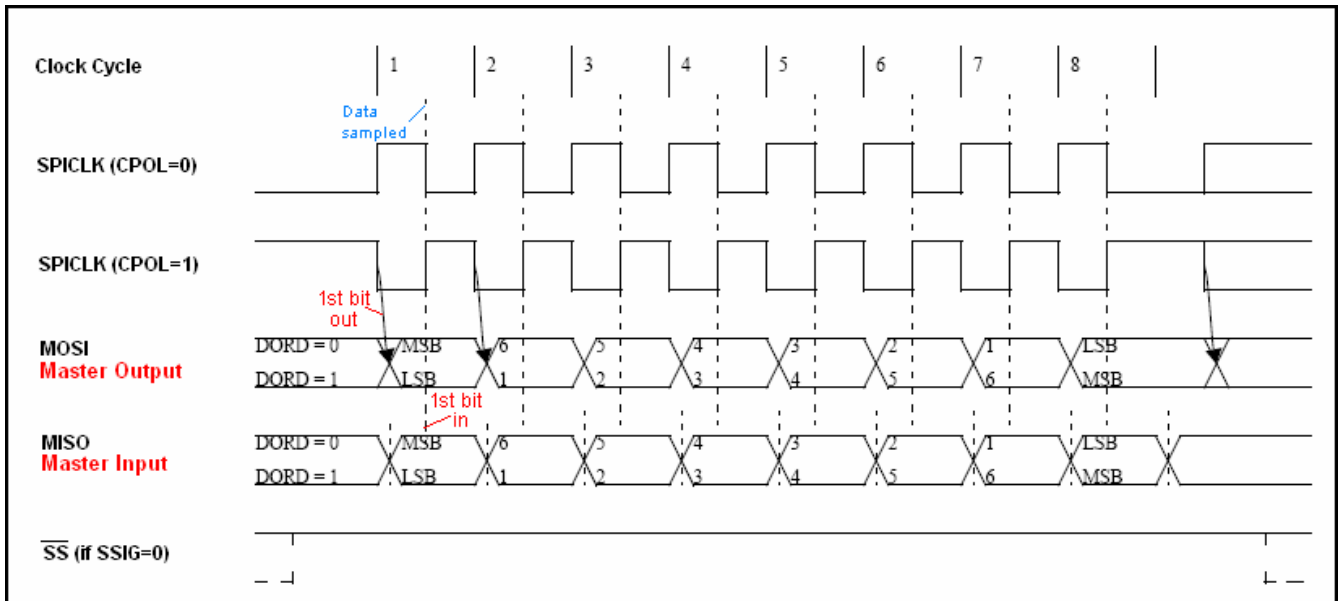


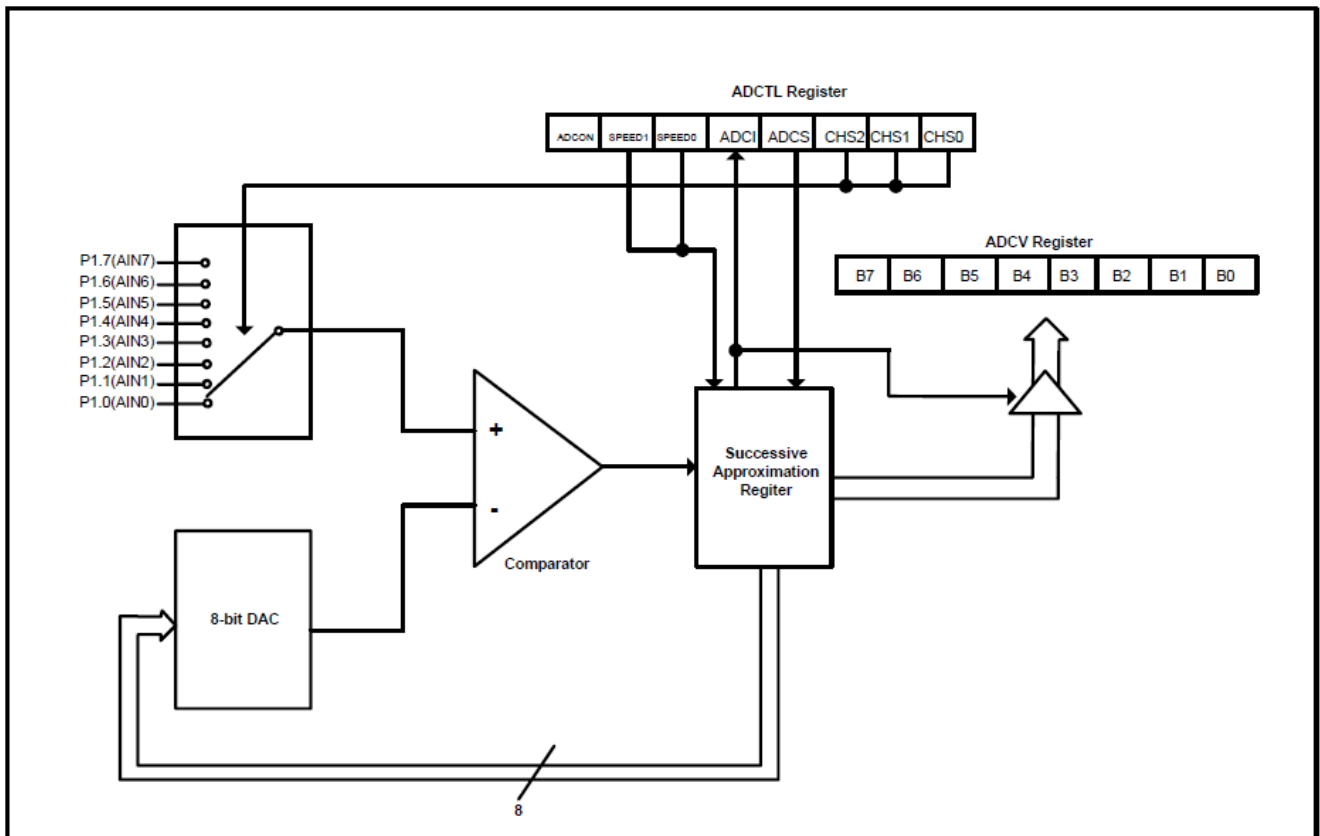
图 14-8. SPI主机传送, CPHA=1



15. 模数 (A/D) 转换器

MA801带有一个8位、8通道逐次逼近型(SAR)模数转换器。图 15-1 显示了A/D转换器部分框图。8路模拟输入同P1口共享端口，多路输入带有一个采样保持电路将模拟电压输入到比较器的输入端，比较器的输出连接到SAR进行逐次逼近操作。

图 15-1. ADC功能框图



15.1. ADC 控制寄存器

ADCTL (地址=C5H, ADC控制寄存器, 复位值=0000,0000B)

7	6	5	4	3	2	1	0
ADCON	SPEED1	SPEED0	ADCI	ADCS	CHS2	CHS1	CHS0

ADCON: 0 : 关闭ADC模块 ; 1 : 开始ADC模块.

SPEED1 and SPEED0: 模数转换速度选择位.

(0,0): 一次转换需要1080个时钟周期.

(0,1): 一次转换需要540个时钟周期.

(1,0): 一次转换需要360个时钟周期.

(1,1): 一次转换需要270个时钟周期.

注 : 1时钟周期等于 $1/F_{osc}$.

ADCS: ADC 启动位。

软件置此位启动A/D转换，转换完成，ADC硬件会自动清除ADCS并设置ADCI。ADCS不能被软件清零。ADCS或ADCI为1时将不会开始新的A/D转换。

ADCI: ADC 中断标志。

一次A/D转换完成时该标志置1，若中断允许则还会产生一个中断。该标志必须软件清零。

CHS2, CHS1 and CHS0: 输入通道选择位

(0,0,0): 选择AIN0 (P1.0) 作为模拟输入

(0,0,1): 选择AIN1 (P1.1) 作为模拟输入

(0,1,0): 选择AIN2 (P1.2) 作为模拟输入

(1,0,1): 选择AIN5 (P1.5) 作为模拟输入

(1,1,0): 选择AIN6 (P1.6) 作为模拟输入

(1,1,1): 选择AIN7 (P1.7) 作为模拟输入

ADCV (地址=C6H, ADC结果寄存器, 复位值=xxH)

7	6	5	4	3	2	1	0
(B7)	(B6)	(B5)	(B4)	(B3)	(B2)	(B1)	(B0)

15.2. ADC 操作

基于引脚兼容标准 8051 MCU的考虑, ADC硬件没有独立的内部正参考电压 (Vref+) 和负参考电压 (Vref-) 输入引脚。Vref+ 和 Vref- 输入芯片内部分别接至VDD和地。所以，满量程 Vref+ – Vref-就是VDD。

在使用ADC功能之前，用户应：

- 1) 设置ADCON 位启动ADC硬件,
- 2) 设置SPEED1 和 SPEED0位设定转换速度,
- 3) 设置CHS2、CHS1 和CHS0选择输入通道,
- 4) 设置P1M0和P1M1寄存器将所选引脚设定成只输入模式,

现在，用户就可以置位ADCS来启动AD转换了。转换时间取决于SPEED1 和 SPEED0位的设置。一旦转换结束，硬件自动清除ADCS位，设置中断标志ADCI，转换结果存入ADCV。

如上所述，中断标志ADCI，由硬件设置以表明一次转换完成。因此，有两种方法检测AD转换是否完成：

- (1) 软件检测ADCI中断标志;
- (2) 设置AUXIE寄存器EADC位和IE寄存器EA位使能ADC中断。这样，转换结束就会跳入中断服务进程。无论(1) 或 (2)，ADCI标志都必须在下次转换前用软件清零。

15.3. ADC 注意事项

15.3.1. A/D 转换时间

用户可以根据输入的模拟信号频率选择合适的转换速度。例如，若 Fosc=10MHz，转换速度设为270个时钟周期，

则输入的模拟信号频率不应超过37KHz，以保证转换精度。(转换时间 = $1/10\text{MHz} \times 270 = 27\mu\text{s}$ ，所以转换速率 = $1/27\mu\text{s} = 37\text{KHz}$.)

15.3.2. I/O 口用于ADC 转换

用作A/D转换的模拟输入引脚也可以保持其数字I/O输入输出功能。为了获得最佳转换效果，用作ADC的引脚应当禁止其数字输出和输入，可以按照引脚配置一节中的描述将引脚设为只输入模式。

15.3.3. 待机和掉电模式

在待机和掉电模式下，ADC将无法使用，若A/D功能打开，它将消耗一部分的电流。因此，为了降低待机和掉电模式下的功耗，可以在进入掉电和待机模式前关闭ADC硬件 (ADCON=0)。

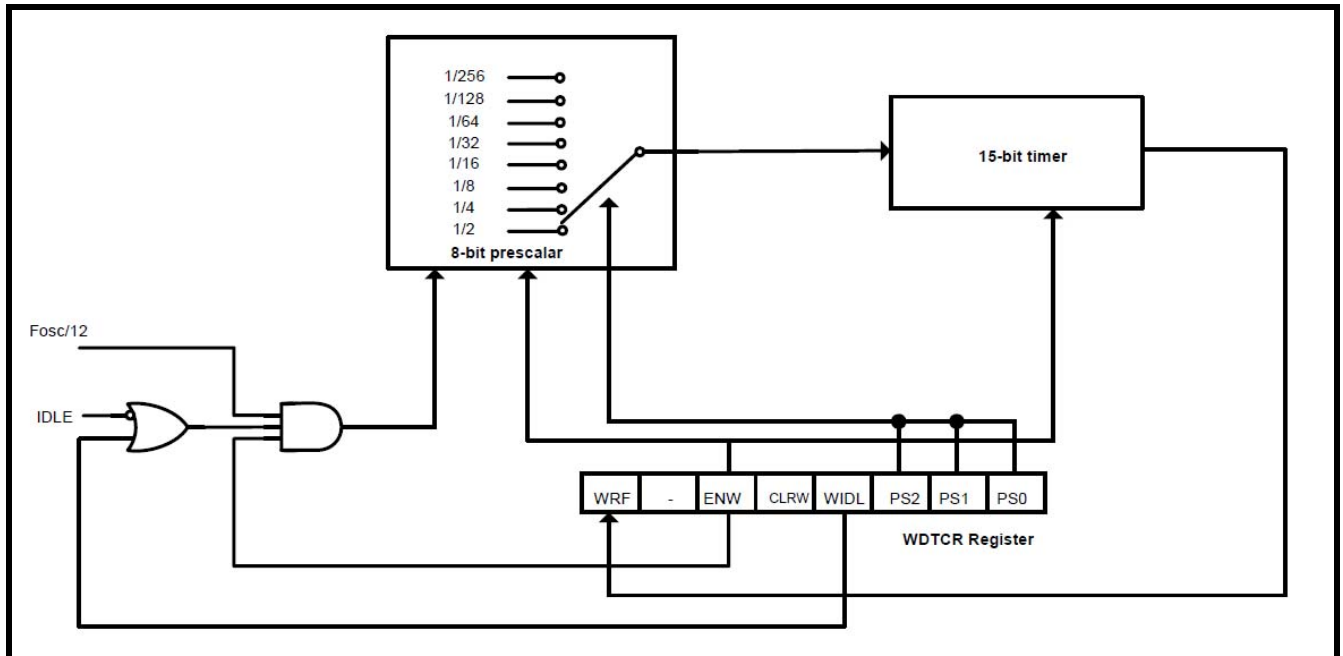
15.3.4. VDD 供电要求

如前所述，Vref+ 和 Vref- 内部分别连接到了 VDD和地，VDD可加载的电压为4.5V~5.5V (5V系统)，所以满量程电压Vref+ - Vref- = VDD 就并不固定。然而，计算公式保持不变，这造成同样的Vin却得到不同的转换结果。因此，VDD必须保持不变，这是用户必须要注意的。

既然VDD用作正参考电压Vref+，用户也必须保证VDD尽可能的纯净，以期得到最佳的ADC性能。

16. 看门狗 (WDT)

16.1. 看门狗结构



16.2. 看门狗寄存器

WDTCR: 看门狗控制寄存器

SFR 地址 = 0xE1

SFR 页 = 全部

复位值 = 0X00-XXXX

7	6	5	4	3	2	1	0
WRF	--	ENW	CLRW	WIDL	PS2	PS1	PS0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7: WRF, WDT 复位标志。WDT 溢出时，这一位被硬件置位，应当软件清零。

Bit 6: 保留。

Bit 5: ENW. WDT 使能位。设置打开 WDT。(注:一旦设置, 该位就只能上电复位清零了。)

Bit 4: CLRW. WDT 清零位。该位写“1”会清除 15 位 WDT 计数器为 0000H。注意该位本身不需写‘0’清除。

Bit 3: WIDL. WDT 在空闲模式的运行。置位该位会让 WDT 在空闲模式下继续计数。

Bit 2~0: PS2 ~ PS0, 预分频设置, 见下表:

PS[2:0]	预分频值
0 0 0	2
0 0 1	4
0 1 0	8
0 1 1	16
1 0 0	32
1 0 1	64
1 1 0	128
1 1 1	256

17. 电源控制

MA801 支持一个电源监测模块，低电平侦察器、和两个电源节能模式：空闲模式（IDLE）和掉电模式（Power-Down）。通过 PCON 寄存器来访问这些模式。

17.1. 节能模式

17.1.1. 空闲模式 (Idle)

可以通过软件的方式置 PCON.IDL 位，使设备进入空闲模式。在空闲模式下，系统不会给 CPU 提供时钟，CPU 状态、RAM、SP、PC、PSW、ACC 被保护起来。I/O 端口也保持当前逻辑。有两种方式是设备从空闲模式唤醒，首先，将“复位”脚连接到高电平来产生一个内部硬件复位可以唤醒空闲模式中的设备，其次任何处于激活状态的中断源都将会清除 PCON.0 而使设备终止空闲模式，并同时进入中断服务程序，只有在中断返回后才会开始执行进入空闲模式指令之后的程序。空闲模式下定时器 0、定时器 1、定时器 2、PWM 定时器和串口仍然处于工作状态，模拟比较器和看门狗有条件的设置在 IDLE 时是否唤醒 CPU。

另外一种使 IDLE 退出的方法是使能唤醒 GPIO 而不需要应用中断能力。

当在 IDLE 模式或掉电模式时，比较器若不用，则作为模拟比较器用的 I/O 口应该设置输出“0”或配置为准双向口。

17.1.2. 掉电模式(Power-down)

可以通过软件的方式置 PCON.PD 位，使设备进入掉电模式。在掉电模式下，振荡器被停止，Flash 存储器掉电以节约电能，只有上电电路继续刷新电源，在减少 VDD 的时候 RAM 的内容仍然会被保持，但特殊功能寄存器 SFR 的内容就不一定能保持住。外部复位、上电复位、外部中断或使能的唤醒口（通用 IO 口 GPIO）使系统退出掉电模式。

系统复位或刚退出掉电模式后至少要等 4 微秒后才能进入或再次进入掉电模式。

17.1.3. 中断唤醒

外部中断 INT0 (P3.2), INT1 (P3.3), 2 个外部中断可以使系统退出掉电模式，但这些中断必须在进入掉电模式前使能并配置成低电平敏感。

唤醒由内部时钟控制，中断口的下降沿系统退出掉电模式，振荡器开始振荡，内部时钟开始计数，但 CPU 要等到内部时钟计数满时才开始执行指令，计数溢出后，中断服务程序开始工作。为了避免中断被重复触发，中断服务程序在返回前应该被禁止，中断口低电平应保持足够长的时间以等系统稳定

17.1.4. 复位唤醒

外部复位脚 RST 唤醒有点类似于中断，复位脚 RST 有上升沿电平时系统退出掉电模式，振荡器开始振荡，内部时钟开始计数，但 CPU 要等到内部时钟计数满时才开始执行指令。复位脚 RST 必须保持足够长时间的高电平以保证系统完全复位，复位脚 RST 变成低电平时便开始执行程序。

值得指出的是当 IDLE 模式被硬件复位唤醒时，前两个机器周期（内部复位没有取得控制权）程序正常从进入 IDLE 模式的后一条指令执行。这时内部硬件是禁止访问内部 RAM 的，但访问 I/O 端口没有被禁止，为了保证不可预料的写 I/O 口，在进入 IDLE 指令后不要放置写 I/O 口或外部存储器的指令（最好加两到三个 NOP 指令）。

17.2. 电源控制寄存器

PCON: 电源控制寄存器

SFR 地址 = 0x87

SFR 页 = 全部

复位值 = 0011-0000

7	6	5	4	3	2	1	0
SMOD1	SMOD0	LVF	POF	GF1	GF0	PD	IDL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 1: PD, 掉电 (Power-Down) 控制位

0: CPU 清 0 或任何一个退出 Power-down 事件发生时自动清 0

1: 置 1 则进入掉电模式

Bit 0: IDL, 空闲模式 (IDLE 模式) 控制位

0: CPU 清 0 或任何一个退出 IDLE 事件发生时自动清 0

1: 置 1 则进入 IDLE 模式

SCMD: ISP 顺序命令寄存器 / RDID (读 DID 寄存器)

SFR 地址 = 0xE6

SFR 页 = 全部

复位值 = xxxx-xxxx

7	6	5	4	3	2	1	0
SCMD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ISP/IAP/IAPLB 的操作都需要用 SCMD 寄存器来触发，当 ISPCR.7 为“1”且 SCMD 顺序写入命令“0x46 0xB9”时，ISP 操作被触发。

ISPCR: ISP 控制寄存器

SFR 地址 = 0xE5

SFR 页 = 全部

复位值 = 0000-xxxx

7	6	5	4	3	2	1	0
ISPEN	SWBS	SWRST	CFAIL	--	WAIT2	WAIT1	WAIT0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit 7: ISPEN, ISP/IAP 使能位

0: 全局禁用 ISP/IAP 编程/擦除/读功能

1: 使能 ISP/IAP 编程/擦除/读功能

Bit 6: SWBS, 软件引导选择位

0: 复位后从 AP 区域启动

1: 复位后从 ISP 区域启动

Bit 5: SWRST, 软件复位触发控制位

0: 无操作

1: 产生软件复位，启动后硬件自动清除它

Bit 4: CFAIL, ISP/IAP 命令是否执行失败

0: ISP/IAP 操作成功

1: ISP/IAP 操作失败

Bit 3: 保留

Bit 2~0 : Wait[2:0] ISP 忙等待时间表等待周期设置

ISPCR.2:0	CPU 等待时间(时钟周期)			对应系统时钟
	页擦除	写	读	
000	672384	1760	2	30M~24M
001	504288	1320	2	24M~20M
010	420240	1100	2	20M~12M
011	252144	660	2	12M~6M
100	126072	330	2	6M~3M
101	63036	165	2	3M~2M
110	42024	110	2	2M~1M
111	21012	55	2	<1M

ISP 详细描述

MA801 不用 IDLE 模式去完成 ISP 操作，而是建立一个 CPU 等待以释放 Flash 存储器供给 ISP 控制电路使用。一旦 ISP 执行完成，CPU 将被唤醒并开始执行调用 ISP 指令的下一条指令，整个 ISP 操作期间中断服务程序被封锁。

19. 在应用程序编程 (IAP)

MA801 可编程序存储器 (AP-memory) 大小被限制为 8K。

当 MCU 从应用程序区 AP 启动时，应用程序只能存取 IAP 区域而 AP 区域及 ISP 区域因受保护而无法访问。

当 MCU 从 ISP 启动时，AP 区域与 IAP 区域都能被 ISP 完全存取。

20. 辅助特殊功能寄存器SFR

AUXR: 辅助寄存器

SFR 地址 = 0x8E

SFR 页 = 全部

复位值 = 0000-00XX

7	6	5	4	3	2	1	0
T0X12	T1X12	URM0X6	EADCI	ESPI	ENLVFI		
R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit 7: T0X12, 当 C/T=0 时, 定时器 0 的时钟源选择。

0: 选择 SYSCLK/12 作时钟源

1: 选择 SYSCLK 作时钟源

Bit 6: T1X12, 当 C/T=0 时, 定时器 1 的时钟源选择。

0: 选择 SYSCLK/12 作时钟源

1: 选择 SYSCLK 作时钟源

Bit 5: URM0X6, 串口模式 0 波特率选择

0: 选择 SYSCLK/12 作 UART 模式 0 波特率

1: 选择 SYSCLK/2 作 UART 模式 0 波特率

Bit 4: EADCI, ADC 中断使能

0: 禁止 ADC 中断使能

1: 使能 ADC 中断使能

Bit 3: ESPI, SPI 中断使能

0: 禁止 SPI 中断使能

1: 使能 SPI 中断使能

Bit 2: ENLVFI, 低电平检测中断使能

0: 禁止低电平检测中断

1: 使能低电平检测中断

Bit 1~0:保留

21. 指令集

Rn	暂存器 R0~R7
direct	8 位内部存储器，包括
1.	内部存储器(00~7F)的地址
2.	特殊功能寄存器(80~FF)的地址，如 P0,PSW,TMOD,...等
@Ri	由寄存器 R0 或 R1 所索引的内部 RAM 数据
#data	8 位常数
#data16	16 位常数
Addr16	16 位的目的地址，可使跳转指令跳转 64K
Addr11	11 位的目的地址，可使跳转指令跳转 2K
rel	有正负号的 8 位地址偏移量，用于相对地址的跳转
bit	1 个 bit: 指所有可以位寻址的的位元
A	累加器 Acc
C 或 CY	进位标志
AC	辅助进位标志
Bb	指定位元 B0~B7
D	半位元组(4bit)
F0	旗号 0
I	中断
PC	程序计数器
SP	堆栈
B	寄存器 B
DPTR	程序数据地址寄存器
@	间接寻址符号
\$	程序计数器当前的值
reg	寄存器

数据传送			
助记符	描述	字节数	指令周期
MOV A, Rn	Acc ← Rn	1	1
MOV A, direct	Acc ← direct	2	2
MOV A, @Ri	Acc ← Ri	1	2
MOV A, #data	Acc ← data	2	2
MOV Rn,A	Rn ← Acc	1	2
MOV Rn,direct	Rn ← direct	2	4
MOV Rn,#data	Rn ← data	2	2

MOV direct,A	direct \leftarrow Acc	2	3
MOV direct,Rn	direct \leftarrow Rn	2	3
MOV direct,direct	direct \leftarrow direct	3	4
MOV direct,@Ri	direct \leftarrow Ri	2	4
MOV direct,#data	direct \leftarrow data	3	3
MOV @Ri,A	Ri \leftarrow Acc	1	3
MOV @Ri,direct	Ri \leftarrow direct	2	3
MOV @Ri,#data	Ri \leftarrow data	2	3
MOV DPTR,#data16	DPTR \leftarrow 16bit data	3	3
MOVC A,@A+DPTR	Acc \leftarrow (A+DPTR)地址所指的数据	1	4
MOVC A,@A+PC	Acc \leftarrow (A+PC)地址所指的数据	1	4
PUSH direct	堆栈 \leftarrow direct	2	4
POP direct	direct \leftarrow 堆栈	2	3
XCH A,Rn	A 和 Rn 互换	1	3
XCH A,direct	A 和 direct 互换	2	4
XCH A,@Ri	A 和 Ri 互换	1	4
XCHD A,@Ri	A 和 Ri 的低四互换	1	4
算术运算			
ADD A,Rn	Acc \leftarrow Acc+Rn	1	2
ADD A,direct	Acc \leftarrow Acc+direct	2	3
ADD A,@Ri	Acc \leftarrow Acc+Ri	1	3
ADD A,#data	Acc \leftarrow Acc+data	2	2
ADDC A,Rn	Acc \leftarrow Acc+Rn+C	1	2
ADDC A,direct	Acc \leftarrow Acc+direct+C	2	3
ADDC A,@Ri	Acc \leftarrow Acc+Ri+C	1	3
ADDC A,#data	Acc \leftarrow Acc+data+C	2	2
SUBB A,Rn	Acc \leftarrow Acc-Rn-C	1	2
SUBB A,direct	Acc \leftarrow Acc-direct-C	2	3
SUBB A,@Ri	Acc \leftarrow Acc-Ri-C	1	3
SUBB A,#data	Acc \leftarrow Acc-data-C	2	2
INC A	Acc \leftarrow Acc +1	1	2
INC Rn	Rn \leftarrow Rn +1	1	3
INC direct	direct \leftarrow direct +1	2	4
INC @Ri	Ri \leftarrow Ri +1	1	4
INC DPTR	DPTR \leftarrow DPTR +1	1	1

DEC A	$Acc \leftarrow Acc - 1$	1	2
DEC Rn	$Rn \leftarrow Rn - 1$	1	3
DEC direct	$direct \leftarrow direct - 1$	2	4
DEC @Ri	$Ri \leftarrow Ri - 1$	1	4
MUL AB	两数相乘，结果高八位存入 B,低八位存入 A	1	4
DIV AB	Acc 除以 B, 商存入 Acc,余数存入 B	1	5
DA A	Acc 作十进制调整	1	4
逻辑运算			
ANL A,Rn	$Acc \leftarrow Acc \text{ and } Rn$	1	2
ANL A,direct	$Acc \leftarrow Acc \text{ and } direct$	2	3
ANL A,@Ri	$Acc \leftarrow Acc \text{ and } Ri$	1	3
ANL A,#data	$Acc \leftarrow Acc \text{ and } data$	2	2
ANL direct,A	$Direct \leftarrow direct \text{ and } Acc$	2	4
ANL direct,#data	$Direct \leftarrow direct \text{ and } data$	3	4
ORL A,Rn	$Acc \leftarrow Acc \text{ or } Rn$	1	2
ORL A,direct	$Acc \leftarrow Acc \text{ or } direct$	2	3
ORL A,@Ri	$Acc \leftarrow Acc \text{ or } Ri$	1	3
ORL A,#data	$Acc \leftarrow Acc \text{ or } data$	2	2
ORL direct,A	$Direct \leftarrow direct \text{ or } Acc$	2	4
ORL direct,#data	$Direct \leftarrow direct \text{ or } data$	3	4
XRL A,Rn	$Acc \leftarrow Acc \text{ xor } Rn$	1	2
XRL A,direct	$Acc \leftarrow Acc \text{ xor } direct$	2	3
XRL A,@Ri	$Acc \leftarrow Acc \text{ xor } Ri$	1	3
XRL A,#data	$Acc \leftarrow Acc \text{ xor } data$	2	2
XRL direct,A	$Direct \leftarrow direct \text{ xor } Acc$	2	4
XRL direct,#data	$Direct \leftarrow direct \text{ xor } data$	3	4
CLR A	清除累加器 Acc	1	1
CPL A	累加器反相	1	2
RL A	累加器向左旋转	1	1
RLC A	累加器和 C 左旋	1	1
RR A	累加器向右旋转	1	1
RRC A	累加器和 C 右旋	1	1
SWAP A	累加器的高低四位互换	1	1
位逻辑运算			
CLR C	清除进位标记	1	1

CLR bit	清除直接位元	2	4
SETB C	设定进位标记	1	1
SETB bit	设定直接位元	2	4
CPL C	进位标记取反	1	1
CPL bit	直接位元取反	2	4
ANL C,bit	$C \leftarrow C \text{ and bit}$	2	3
ANL C,/bit	$C \leftarrow C \text{ and bit(反相)}$	2	3
ORL C,bit	$C \leftarrow C \text{ or bit}$	2	3
ORL C,/bit	$C \leftarrow C \text{ or bit(反相)}$	2	3
MOV C,bit	$C \leftarrow \text{bit}$	2	3
MOV bit,C	$\text{bit} \leftarrow C$	2	4
位逻辑跳转			
JC rel	如果 C=1 跳转到 rel	2	3
JNC rel	如果 C=0 跳转到 rel	2	3
JB bit,rel	如果 bit=1 跳转到 rel	3	4
JNB bit,rel	如果 bit=0 跳转到 rel	3	4
JBC bit,rel	如果 bit=1 跳转到 rel,并且清除 bit	3	5
程序跳转			
ACALL addr11	绝对式子程序调用	2	6
LCALL addr16	远程子程序调用	3	6
RET	从子程序返回	1	4
RETI	从中断返回	1	4
AJMP addr11	绝对式跳转	2	3
LJMP addr16	远程跳转	3	4
SJMP rel	短程跳转	2	3
JMP @A+DPTR	间接跳转	1	3
JZ rel	如果 Acc=0 则跳到 rel	2	3
JNZ rel	如果 Acc≠0 则跳到 rel	2	3
CJNE A,direct,rel	如果 Acc≠direct 则跳到 rel	3	5
CJNE A,#data,rel	如果 Acc≠data 则跳到 rel	3	4
CJNE Rn,#data,rel	如果 Rn≠data 则跳到 rel	3	4
CJNE @Ri,#data,rel	如果 Ri≠data 则跳到 rel	3	5
DJNZ Rn,rel	如果(Rn-1)≠0 则跳到 rel	2	4
DJNZ direct,rel	如果(direct-1)≠0 则跳到 rel	3	5
NOP	无动作	1	1

22. 最大绝对额定参数

MA801

参数	额定值	单位
环境温度偏差	-55 ~ +125	°C
存储温度	-65 ~ + 150	°C
IO 口和复位脚的对地电压	-0.5 ~ VDD + 0.5	V
VDD 脚的对地电压	-0.5 ~ +6.0	V
芯片总电流	400	mA
IO 口的最大吸收电流	40	mA

*注意：实际参数超过上述各项“绝对最大额定值”可能会对设备造成永久性损坏。这些参数是一个设备进行正常功能操作的最大额定值，任何超过上述各项的条件都不被建议，否则可能会影响设备运行的稳定性。

23. 电气特性

23.1. 直流特性

VSS = 0V, TA = 25 °C, VDD = 5.0V

符号	参数	条件	范围			单位
			最小	标称	最大	
V _{IH1}	输入高电压 (Quasi, Input-only or Open-drain)		2.0			V
V _{IH2}	输入高电压, RST		3.5			V
V _{IL1}	输入低电压, (Quasi, Input-only or Open-drain)				0.8	V
I _{IH1}	输入高的漏电流 (Quasi, Input-only or Open-drain)	V _{PIN} = VDD		0	10	uA
I _{IL1}	输入低的电流 (Quasi-bidirectional)	V _{PIN} = 0.45V		17	50	uA
I _{IL2}	输入低的电流 (Input-only or Open-drain)	V _{PIN} = 0.45V		0	10	uA
I _{H2L}	输入下降沿的跳变电流 (Quasi-bidirectional)	V _{PIN} = 1.8V		230	500	uA
I _{OH2}	输出高的电流 (Quasi-bidirectional)	V _{PIN} = 2.4V		220		uA
I _{OH1}	输出高的电流 (Push-pull output)	V _{PIN} = 2.4V	12	20		mA
I _{OL}	输出低的电流	V _{PIN} = 0.45V	12	20		mA
I _{OP}	工作电流	F _{OSC} = 12MHz		12	30	mA
I _{IDLE}	空闲模式电流	F _{OSC} = 12MHz		6	15	mA
I _{PD}	掉电模式电流			0.1	50	uA
R _{RST}	复位脚上的内部下拉电阻			100		Kohm

24. 修订历史

版本	日期	页	描述
V1.00	2010/Feb./25		初稿