



微机原理及接口技术教程

习题解答

习题1

习题2

习题3

习题4

习题5

习题6

习题7

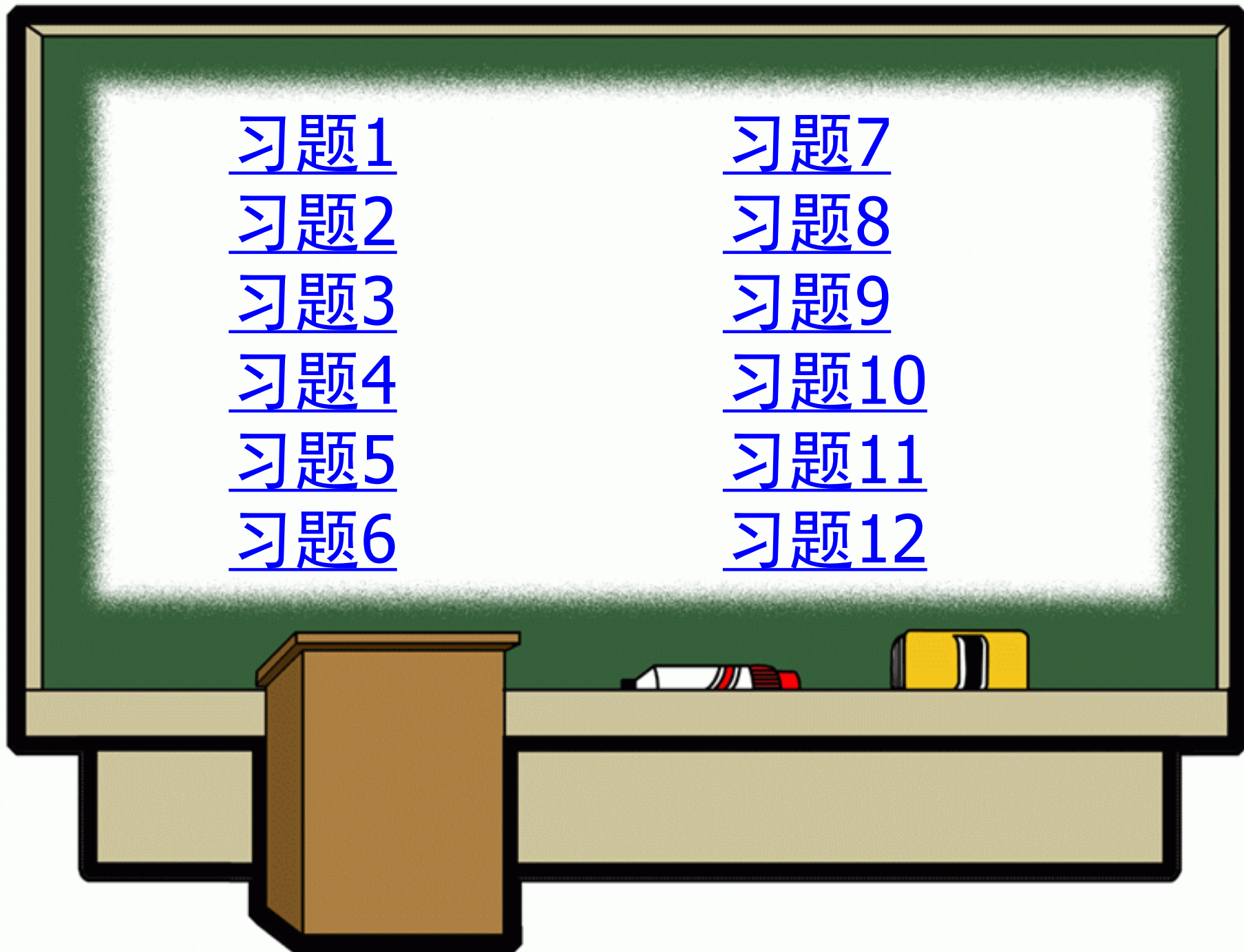
习题8

习题9

习题10

习题11

习题12



习题1——第14页

- 1.2
- 1.6
- 1.7
- 1.12



习题1.2解答



- 通用微处理器：适合较广的应用领域的微处理器，例如装在**PC**机、笔记本电脑、工作站、服务器上的微处理器
- 单片机：是指通常用于控制领域的微处理器芯片，其内部除**CPU**外还集成了计算机的其他一些主要部件，只需配上少量的外部电路和设备，就可以构成具体的应用系统
- **DSP**芯片：称数字信号处理器，也是一种微控制器，其更适合处理高速的数字信号，内部集成有高速乘法器，能够进行快速乘法和加法运算。
- 嵌入式系统：利用微控制器、数字信号处理器或通用微处理器，结合具体应用构成的控制系统，其典型的特点是把计算机直接嵌入到应用系统中

习题1.6解答



- 总线：传递信息的一组公用导线，**CPU**通过它们与存储器和**I/O**设备进行信息交换
- **3组信号线：**
 - 数据总线：传输将要访问的主存单元或**I/O**端口的地址信息
 - 地址总线：传输读写操作的数据信息
 - 控制总线：协调系统中各部件的操作

习题1.7解答1



(1) 计算机字长

- 处理器每个单位时间可以处理的二进制数据位数称计算机字长

(2) 取指—译码—执行周期

- 指令的处理过程，即指处理器从主存储器读取指令（简称取指），翻译指令代码的功能（简称译码），然后执行指令所规定的操作（简称执行）的过程

(3) ROM-BIOS

- “基本输入输出系统”，操作系统通过对**BIOS**的调用驱动各硬件设备，用户也可以在应用程序中调用**BIOS**中的许多功能



习题1.7解答2



(4) 中断

- 中断是**CPU**正常执行程序的流程被某种原因打断、并暂时停止，转向执行事先安排好的一段处理程序，待该处理程序结束后仍返回被中断的指令继续执行的过程

(5) **ISA**总线

- 指**IBM PC/AT**机上使用的系统总线

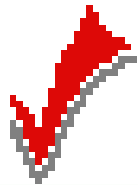
习题1.12解答



- 数码**0**~**9**对应的**ASCII**码依次是：
30H~39H
- 大写字母**A**~**Z**对应的**ASCII**码依次是：
41H~5AH
- 小写字母**a**~**z**对应的**ASCII**码依次是：
61~7AH
- **ASCII**码**0DH**和**0AH**分别对应的是
回车和换行字符

习题2——第49页

- 2.5
- 2.8
- 2.11
- 2.12
- 2.14
- 2.17
- 2.20(1)
- 2.20(2)
- 2.20(3)
- 2.20(4)
- 2.20(5)



习题2.5解答1



- 物理地址：物理存储单元具有的一个唯一的**20**位编号
- 逻辑地址：在**8088**内部和用户编程时，所采用的“段地址：偏移地址”形式
- 将逻辑地址中的段地址左移二进制**4**位（对应**16**进制是一位，即乘以**16**），加上偏移地址就得到**20**位物理地址
- **1MB**最多能分成 $1\text{MB} \div 16\text{B} = 2^{20} \div 2^4 = 2^{16}$ 个逻辑段，因为每隔**16**个字节单元就可以开始一个逻辑段



习题2.5解答2



(1) **FFFFH:0 = FFFF0H**

(2) **40H:17H = 00417H**

(3) **2000H:4500H = 24500H**

(4) **B821H:4567H = BC777H**

习题2.8解答



- (1) **mov ax,1200h** ; **AX=1200H**, 立即数寻址
- (2) **mov ax,bx** ; **AX=0100H**, 寄存器寻址
- (3) **mov ax,[1200h]** ; **AX=4C2AH**, 直接寻址
- (4) **mov ax,[bx]**
; **AX=3412H**, 寄存器间接寻址
- (5) **mov ax,[bx+1100h]**
; **AX=4C2AH**, 寄存器相对寻址
- (6) **mov ax,[bx+si]**
; **AX=7856H**, 基址变址寻址
- (7) **mov ax,[bx][si+1100h]**
; **AX=65B7H**, 相对基址变址寻址

习题2.11解答



mov al, 89h	;	AL=89H	CF	ZF	SF	OF	PF
add al, al	;	AL=12H	1	0	0	1	1
add al, 9dh	;	AL=0AFH	0	0	1	0	1
cmp al, 0bch	;	AL=0AFH	1	0	1	0	1
sub al, al	;	AL=00H	0	1	0	0	1
dec al	;	AL=0FFH	0	0	1	0	1
inc al	;	AL=00H	0	1	0	0	1

习题2.12解答



- (1) **add dx,bx**
- (2) **add al,[bx+si]**
- (3) **add [bx+0b2h],cx**
- (4) **add word ptr [0520h],3412h**
- (5) **add al,0a0h**

习题2.14解答



mov ax,1470h ; AX = 1470H CF ZF SF
OF PF

and ax,ax ; AX=1470H 0 0 0 0 0

or ax,ax ; AX=1470H 0 0 0 0 0

xor ax,ax ; AX=0000H 0 1 0 0 1

not ax ; AX=FFFFH 0 1 0 0 1

test ax,0f0f0h
; AX=FFFFH 0 0 1 0 1

习题2.17解答



(1) **xor ax,1e1eh**

je equal ; AX=1e1eh (异或后为0)

(2) **test al,10000001b**

jnz there ; AL的D₀或D₇至少有一位为1

(3) **cmp cx,64h**

jb there ; CX (无符号数) < 64h

习题2.20 (1) 解答



```
mov dl,string[1]      ; string[0]  
mov dh,string[6]      ; string[5]
```

习题2.20 (2) 解答



mov ax, word ptr buffer1

sub ax, word ptr buffer2

；先减低**16**位

mov dx, word ptr buffer1+2

sbb dx, word ptr buffer2+2

；后减高**16**位，需减低**16**位的借位

习题2.20 (3) 解答



```
test dx,0f000h ; test dh,0f0h
jz next        ; jnz next
mov ax,-1      ; mov ax,0
jmp again
next:  mov ax,0      ; mov ax,0ffffh
again: ...
```

习题2.20 (4) 解答



mov cx,4

again: shr dx,1

； 右移一位，移出的低位进入**CF**标志

rcr ax,1

； 将**CF**移进**AX**高位，同时实现**AX**右移

loop again

； 循环**4**次，实现**4**位右移

习题2.20 (5) 解答



```
mov si,offset array
```

```
mov cx,100
```

```
again: dec byte ptr [si]
```

```
inc si
```

```
loop again
```

习题3——第85页

■ 3.6

■ 3.26

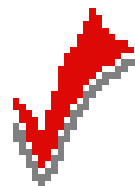
■ 3.10

■ 3.28

■ 3.18

■ 3.34

■ 3.22



习题3.6解答



```
.model small  
.stack  
.data  
string db 'Hello, Assembly !',0dh,0ah,'$'  
.code  
start: mov ax,@data  
       mov ds,ax  
       mov dx,offset string  
       mov ah,9  
       int 21h  
       mov ax,4c00h  
       int 21h  
       end start
```



习题3.10解答



	'A'	'B'	'C'	10	10h	'E'	'F'	-1	—	4	4	4	...
	41h	42h	43h	0ah	10h	45h	46h	ffh	—	04h	04h	04h	
										3个			

10h	0	FBh	FFh	—	—	—	—	—	—			
10h		—5		?		?		?				

习题3.18解答



```
mov al, bufX  
cmp al, bufY  
jge done  
mov al, bufY
```

```
done:  mov bufZ, al
```



习题3.22解答



```

; 数据段
b_data db 12h,45h,0f3h,6ah,20h
        db 0feh,90h,0c8h,57h,34h

num     equ 10                ; 数据个数
sum     db ?                  ; 预留结果单元

; 代码段
xor si, si                    ; 位移量清零
xor al, al                    ; 取第一个数
mov cx, num                    ; 累加次数
again:  add al, b_data[si]    ; 累加
        inc si                ; 指向下一个数
        loop again            ; 如未完，继续累加
        mov sum, al           ; 完成，存结果
```



习题3.26解答



```
crazy PROC  
    push ax          ; 删除  
    xor ax, ax  
    xor dx, dx  
again: add ax, [bx]  
    adc dx, 0  
    inc bx  
    inc bx  
    loop again  
    ret  
ENDP crazy          ; 修改为: crazy ENDP
```

字量求和子程序

入口参数:

BX=数据首址

CX=数据个数

出口参数:

AX=和的低字

DX=和的高字
(进位部分)



习题3.28解答



asctob proc

push cx

； 先转换十位数

and dh,0fh

； 乘以**10**

shl dh,1

mov ch,dh

shl dh,1

shl dh,1

add dh,ch

； 转换个位数

and dl,0fh

； 十位数加个位数

add dh,dl

； 设置出口参数

mov al,dh

pop cx

ret

asctob endp



习题3.34解答1



； 数据段

array db

12h,25h,0f0h,0a3h,3,68h,71h,0cah,0ffh,90h

count equ \$-array

； 数组元素个数

result db ?

； 校验和

； 代码段（主程序）

mov bx,offset array

； **BX**←数组的偏移地址

mov cx,count

； **CX**←数组的元素个数

call checksum

； 调用求和过程

mov result,al

； 处理出口参数



习题3.34解答2



- ； 代码段（子程序）
- ； 计算字节校验和的通用过程
- ； 入口参数：**DS:BX**=数组的段地址:偏移地址
CX=元素个数
- ； 出口参数：**AL**=校验和
- ； 说明：除**AX/BX/CX**外，不影响其他寄存器

```
checksum proc
    xor al,al                ; 累加器清0
sum:    add al,[bx]          ; 求和
        inc bx              ; 指向下一个字节
        loop sum
        ret
checksum endp
```



习题4——第107页

■ 4.1

■ 4.4

■ 4.5

■ 4.6

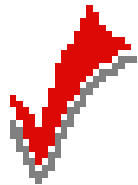
■ 4.7

■ 4.8

■ 4.9

■ 4.11

■ 4.13



习题4.1解答



- 微机总线的信号线包括 数据总线、地址总线、控制总线 以及电源和地线。微机系统可以将总线划分为三层（类），它们是 芯片总线、内总线 和 外总线。

习题4.4解答



$$\begin{aligned} & (16 \div 2) \times 8 \times 10^6 \\ &= 64 \times 10^6 \text{bps} \\ &= 64 \text{ Mb/s} \\ &= 8 \text{ MB/s} \end{aligned}$$

习题4.5解答



- 三态：引脚除正常的高、低电平外，还能输出高阻状态。
- 输出高阻状态时，表示芯片实际上已放弃了对该引脚的控制，使之“悬空”，这样他所连接的设备就可以接管对该引脚所连导线的控制。



习题4.5解答（续）



8088最小组态具有三态能力的引脚：

- **A19/S6~A16/S3、A15~A8、AD7~AD0**
- **IO/M*、RD*、WR*、ALE**
- **DEN*、DT/R***



习题4.6解答



- **RESET**: 复位请求。有效时，将使**CPU**回到其初始状态。
- **HOLD**: 总线保持。有效时，表示总线请求设备向**CPU**申请占有总线。
- **NMI**: 不可屏蔽中断请求。有效时，表示外界向**CPU**申请不可屏蔽中断。
- **INTR**: 可屏蔽中断请求。有效时，表示请求设备向**CPU**申请可屏蔽中断。

习题4.7解答



- 执行一条指令所需要的时间被称为 指令周期，而总线周期指的是 CPU通过总线操作与外界（存储器和I/O端口）进行一次数据交换的过程（时间），8088基本的总线周期由 4 个T组成。如果8088的CLK引脚接5MHz的时钟信号，那么每个T状态的持续时间为 200ns。

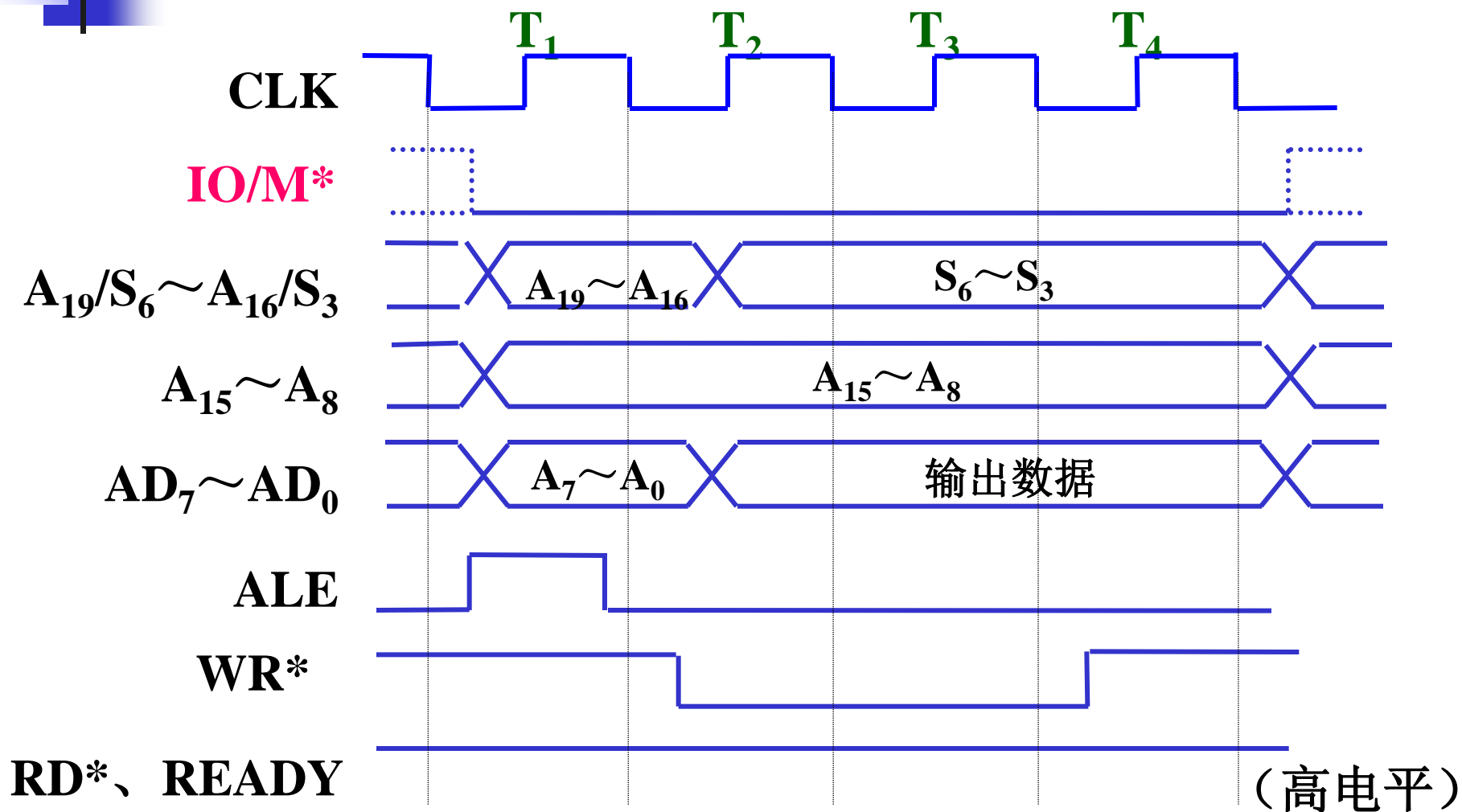
习题4.8解答



- **CLK: CPU时钟信号**
- **A19/S6~A16/S3: 地址信号A19~A16和状态信号S6~S3分时复用信号**
- **A15~A8: 地址信号A15~A8**
- **AD7~AD0: 地址信号A7~A0和数据信号D7~D0分时复用信号**
- **IO/M*: I/O接口和存储器操作区别信号**
- **RD*: 读控制信号, WR*: 写控制信号**
- **ALE: 地址锁存信号**



习题4.8图



习题4.9解答



- 当**8088**进行读写存储器或**I/O**接口时，如果存储器或**I/O**接口无法满足**CPU**的读写时序（来不及提供或读取数据时），需要**CPU**插入等待状态 **T_w** 。
- 在读写总线周期的 **T_3** 和 **T_4** 之间插入 **T_w** 。

习题4.11解答



- **8088**每个总线周期只能读写一个字节数据。所以读取指令长度为**3B**的指令“**ADD [2000H], AX**”需要 3 个时钟周期，执行时需要 2 个时钟周期。
- 读取指令是 存储器读 总线周期，执行时是 存储器写 总线周期？

习题4.13解答



D0~D7: 8位双向数据总线

A0~A19: 20位输出地址总线

ALE: 地址锁存允许, 每个CPU 总线周期有效

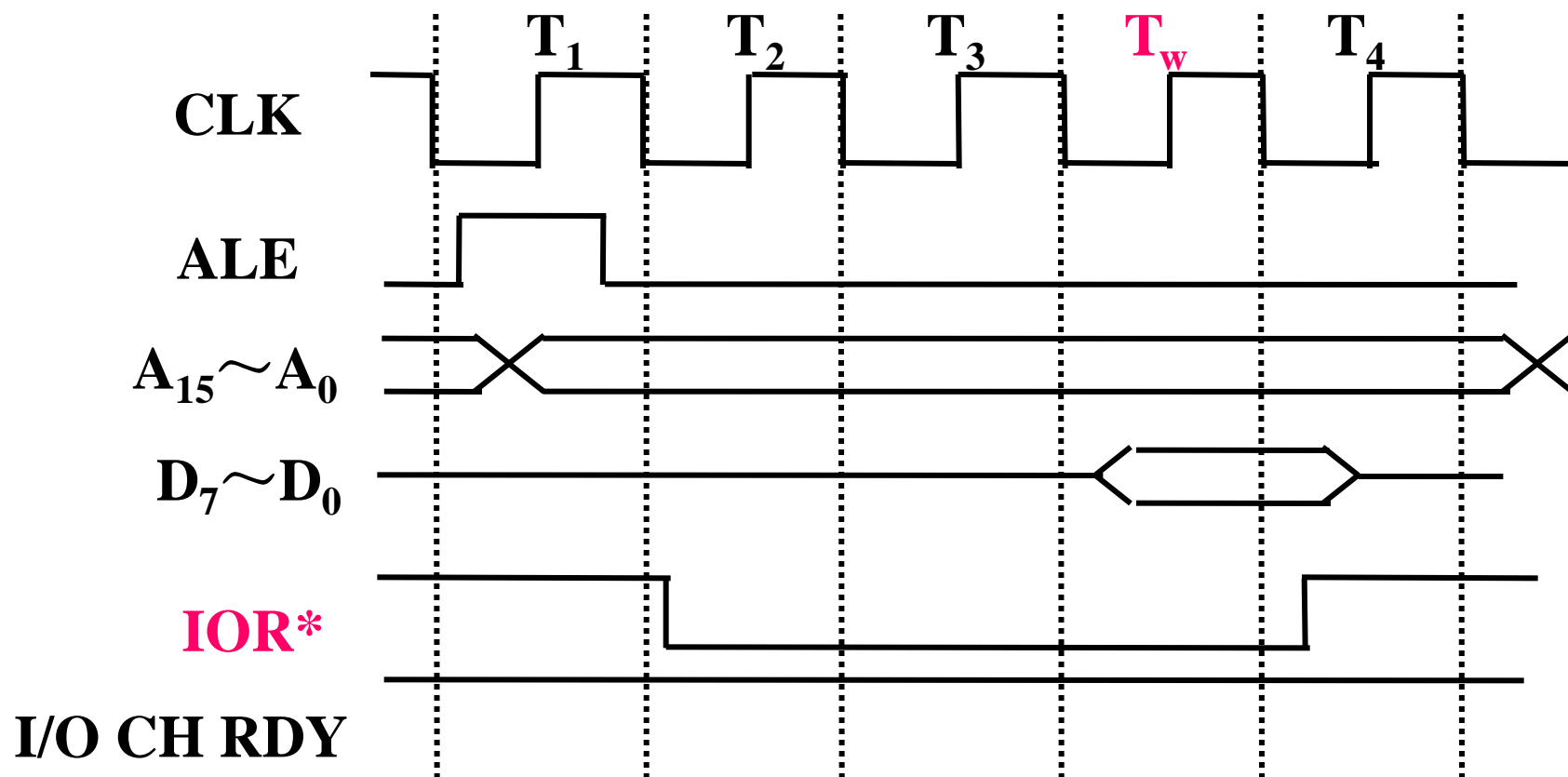
IOR*: I/O读, 输出

IOW*: I/O写, 输出

IO CH RDY: I/O通道准备好, 输入



习题4.13图



习题5——第132页

■ 5.3

■ 5.12

■ 5.8

■ 5.13

■ 5.10

■ 5.14

■ 5.11



习题5.3解答



- 类似处理器总线，存储器芯片也分成数据、地址和控制3类引脚。以存储结构为 $32\text{K} \times 8$ 的SRAM 62256为例，该芯片应有8个数据引脚、15个地址引脚，3个典型的控制引脚分别是片选、输出允许和8写入允许。

习题5.8解答



- 主存容量: $2^{15}=32\text{KB}$
 - 因低位地址信号的个数是15个
- 地址范围:
01011 0000000000000000 ~
01011 1111111111111111
■ 即 $58000\text{H} \sim 5\text{FFFFH}$

习题5.10解答



- 全译码：使用全部系统地址总线进行译码确定存储单元地址的方法
 - 特点是：地址唯一，空间连续
- 部分译码：使用部分系统地址总线进行译码确定存储单元地址的方法
 - 特点是：地址译码相对简单
- 采用部分译码会产生地址重复
- 2个高位系统地址信号没有参加译码
占用 $2^2=4$ 个地址

习题5.11解答



- 位扩充——存储器芯片数据位数小于主机数据总线数时，利用多个存储器芯片在数据“位”方向的扩充；
- 字扩充——当一个存储器芯片不能满足系统存储容量时，利用多个存储器芯片在“数据字”方向的扩充
- 组成512KB存储模块，用DRAM 21256（ $256\text{K} \times 1$ ）需要16个芯片；位扩充8个，字扩充2组

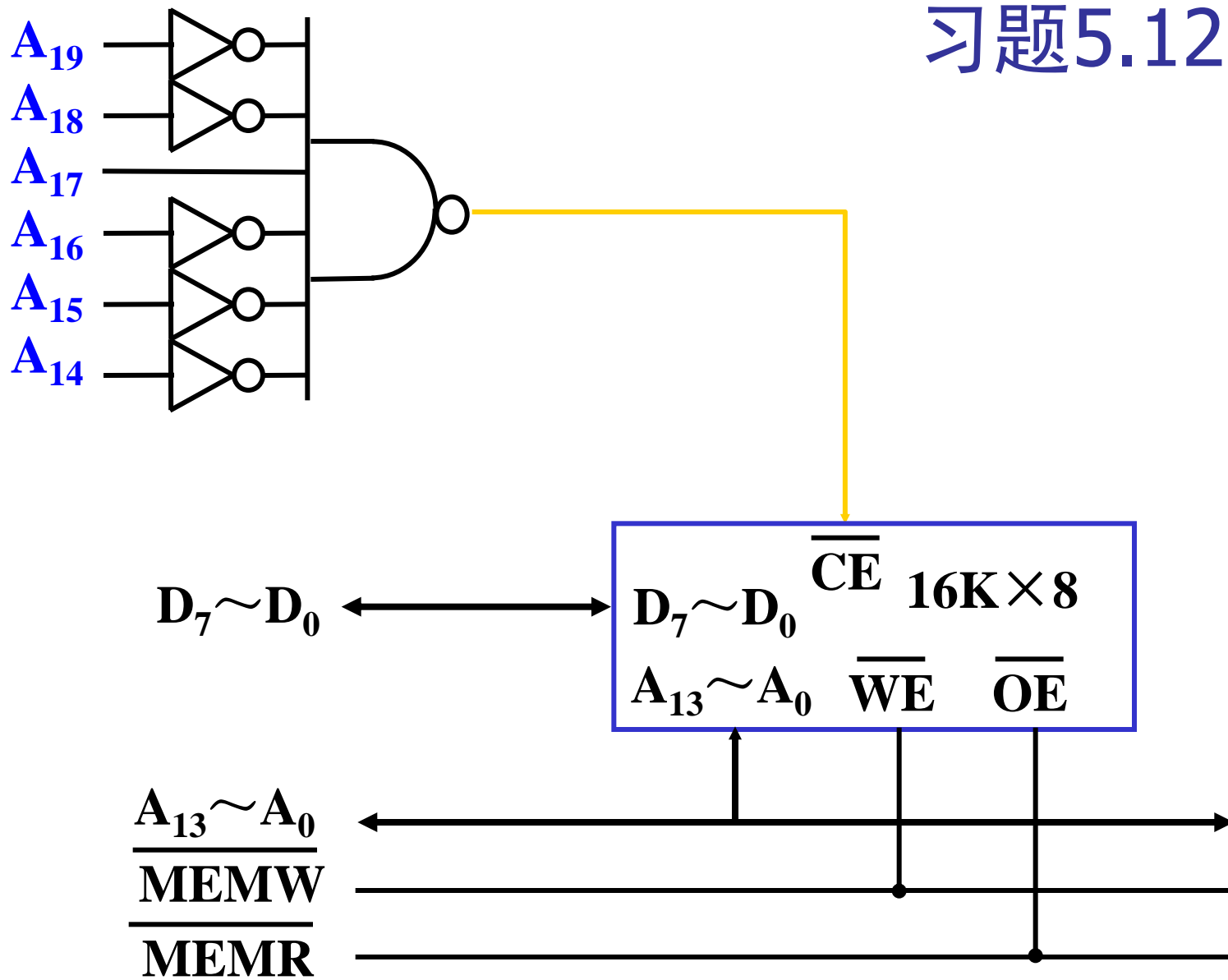
习题5.12解答



$A_{19} \sim A_{14}$	$A_{13} \sim A_0$	地址
001000	全0~全1	20000H~23FFFH



习题5.12图



习题5.13



问题:

- 请写出图5.28中4个存储芯片各自的可用地址范围，并指明其重复地址。

解答:

	$A_{19} A_{18}$	$A_{17} A_{16}$	$A_{15} \sim A_0$	一个可用地址
1	0 0	× ×	全0~全1	00000H~3FFFFH
2	0 1	× ×	全0~全1	40000H~7FFFFH
3	1 1	× 0	全0~全1	C0000H~CFFFFH
4	1 1	× 1	全0~全1	D0000H~DFFFFH



习题5.13重复地址



芯片	一个可用地址	重复地址
3	C0000H~CFFFFFFH	E0000~EFFFFFFH
4	D0000H~DFFFFFFH	F0000~FFFFFFH



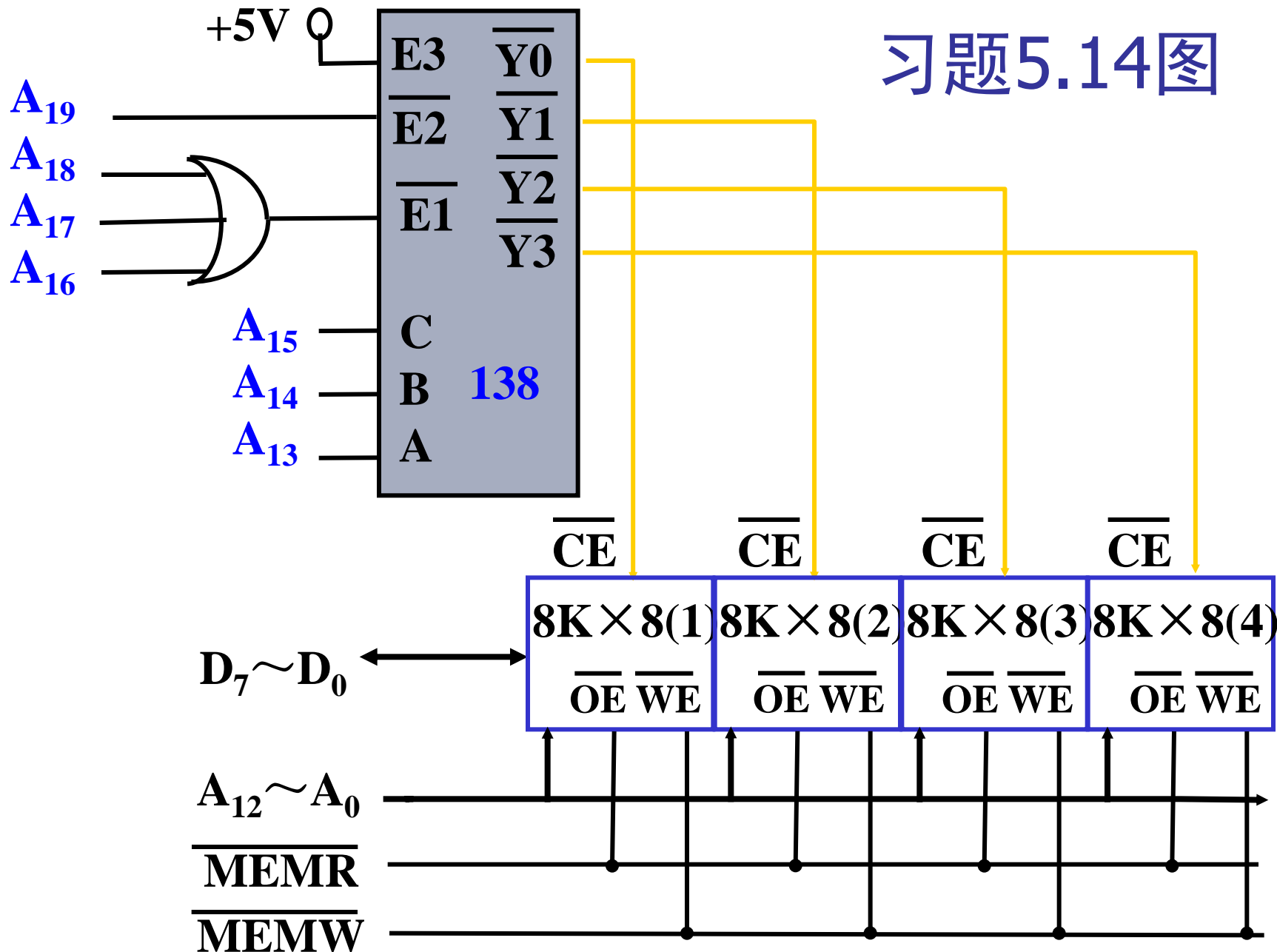
习题5.14解答



	$A_{19} \sim A_{16}$	$A_{15} \sim A_{13}$	$A_{12} \sim A_0$	地址
1	0000	000	全0~全1	00000H~01FFFH
2	0000	001	全0~全1	02000H~03FFFH
3	0000	010	全0~全1	04000H~05FFFH
4	0000	011	全0~全1	06000H~07FFFH

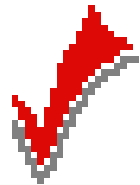


习题5.14图



习题6——第152页

- 6.1
- 6.2
- 6.4
- 6.5
- 6.8



习题6.1解答



(1) 数据寄存器

- 保存外设给**CPU**和**CPU**发往外设的数据

(2) 状态寄存器

- 保存外设或接口电路的状态

(3) 控制寄存器

- 保存**CPU**给外设或接口电路的命令

习题6.2解答



- (I/O端口与存储器地址) 统一编址
- (I/O端口与存储器地址) 独立编址
- 外设
- 处理器
- I/O读
- (I/O端口的DX) 寄存器间接
- 寄存器

习题6.4程序1



```
again:  mov dx,6000h
        mov al,0ffh
        out dx,al      ;全不亮
again1:  in al,dx
        cmp al,7fh     ;D7~D0=0111111B ?
        jz done        ;单独按下K7, 退出
        cmp al,0feh    ;=11111110B ?
        jz next1       ;单独按下K0, 转next1
        cmp al,0fdh    ;=11111101B ?
        jz next2       ;单独按下K1, 转next2
        jmp again      ;其它情况不点亮
```



习题6.4程序2



```
next1:  mov cx,8
        mov al,1      ;从K0开始
next11: out dx,al      ;某个LED电亮
        call delay    ;延时200ms
        shl al,1       ;rol al,1
        loop next11
        jmp again1
```



习题6.4程序3



```
next2:  mov cx,8
        mov al,80h    ;从K7开始
next21: out dx,al      ;某个LED电亮
        call delay    ;延时200ms
        shr al,1       ;ror al,1
        loop next21
        jmp again1
done:   mov al,0ffh
        out dx,al      ;全不亮
```



习题6.5程序



```
mov bx,offset buf
mov cx,100
again:  mov dx,8f42h
status: in al,dx          ; 查询一次
        test al,01h
        jz status
        mov dx,8f40h
        in al,dx          ; 输入一个字节
        mov [bx],al
        inc bx
        loop again        ; 循环，输入100个字节
```

习题6.8程序

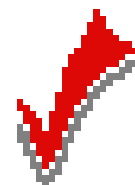


status:	sti
	push ax	service0: call proc0
	push dx	jmp done
	mov dx,4000h	service1: call proc1
	in al,dx	jmp done
	test al,01h	service2: call proc2
	jnz service0	jmp done
	test al,02h	service3: call proc3
	jnz service1	jmp done
	test al,04h
jnz service2	done: pop dx	
test al,08h	pop ax	
jnz service3	iret	

BACK

习题7——第174页

- 7.4
- 7.6
- 7.9
- 7.11
- 7.13



习题7.4解答



■ 中断请求寄存器IRR

- 保存8条外界中断请求信号IR0~IR7的请求状态
- Di位为1表示IRi引脚有中断请求；为0表示无请求

■ 中断服务寄存器ISR

- 保存正在被8259A服务着的中断状态
- Di位为1表示IRi中断正在服务中；为0表示没有被服务

■ 中断屏蔽寄存器IMR

- 保存对中断请求信号IR的屏蔽状态
- Di位为1表示IRi中断被屏蔽（禁止）；为0表示允许

习题7.6解答



- 某时刻8259A的IRR内容是08H，说明 IR3引脚有中断请求。某时刻8259A的ISR内容是08H，说明 IR3正在被中断服务，其它不被处理。在两片8259A级连的中断电路中，主片的第5级IR5作为从片的中断请求输入，则初始化主、从片时，ICW3的控制字分别是 20H 和 05H。

习题7.9解答



■ 普通全嵌套方式

- 8259A的中断**优先权顺序**固定不变，从高到低依次为IR0、IR1、IR2、.....IR7
- 中断请求后，8259A对**当前请求中断中优先权最高的中断IRi**予以**响应**，将其向量号送上数据总线
- 在ISR的Di位置位期间，**禁止再发生同级和低级优先权的中断**，但允许高级优先权中断的嵌套

■ 普通中断结束方式

- 配合全嵌套优先权方式使用
- CPU用输出指令往8259A发出**普通中断结束EOI命令**，8259A就会复位正在服务的中断中优先权最高的ISR位

习题7.11解答



不是查询字

- 读取中断服务寄存器ISR的内容
- 因为执行输入指令（ $A0=0$ ）之前，执行的输出指令，写入了OCW3（ $D4D3=01$ ），其中P RR RIS（ $D2D1D0$ ）=011指明随后读出ISR
- 不是查询字

习题7.13解答1



int08h	proc far	;远过程
	sti	;开中断，允许中断嵌套
	push ds	;现场保护
	push ax	
	push dx	
	;日时钟计时
	;控制软驱马达



习题7.13解答2



int 1ch ;调用指令中断1CH

mov al,20h ;发送EOI命令

out 20h,al

pop ax ;现场恢复

pop dx

pop ds

iret ;中断返回

int08h endp



习题8——第188页

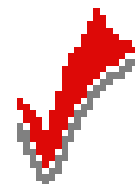
■ 8.2

■ 8.3

■ 8.4

■ 8.6

■ 8.7



习题8.2解答



- **CLK**时钟输入信号——在计数过程中，此引脚上每输入一个时钟信号（下降沿），计数器的计数值减**1**
- **GATE**门控输入信号——控制计数器工作，可分成电平控制和上升沿控制两种类型
- **OUT**计数器输出信号——当一次计数过程结束（计数值减为**0**），**OUT**引脚上将产生一个输出信号

习题8.3解答



- **8253**每个通道有 6 种工作方式可供选择。若设定某通道为方式**0**后，其输出引脚为 低 电平；当 写入计数初值（并进入减1计数器） 后通道开始计数，CLK 信号端每来一个脉冲 减1计数器 就减**1**；当 计数器减为0，则输出引脚输出 高 电平，表示计数结束。**8253**的**CLK0**接**1.5MHz**的时钟，欲使**OUT0**产生频率为**300KHz**的方波信号，则**8253**的计数值应为 **5** ($=1.5\text{MHz} \div 300\text{KHz}$)，应选用的工作方式是 3。

习题8.4解答 (1)



```
mov al,50h  
mov dx,207h  
out dx,al  
mov al,128 ; 80h  
mov dx,205h  
out dx,al
```



习题8.4解答 (2)



mov al,33h

mov dx,207h

out dx,al

mov ax,3000h ; 不是3000

mov dx,204h

out dx,al

mov al,ah

out dx,al



习题8.4解答 (3)



```
mov al,0b4h  
mov dx,207h  
out dx,al  
mov al,02f0h  
mov dx,206h  
out dx,al  
mov al,ah  
out dx,al
```



习题8.6解答



- 计数器0——

每隔**55ms**产生一个**IRQ0**中断请求

- 计数器1——

每隔**15 μ s**产生一个**DRAM**刷新请求

- 计数器2——

控制扬声器音调

习题8.7解答



- **101**个下降沿，还可以采用方式**4**
- **GATE0**接外部启动计数器的控制信号，可以选用方式**1**或方式**5**

mov dx,203h

mov al,12h ;方式**5**为**1ah**

out dx,al

mov dx,200h

mov al,64h

out dx,al

习题9——第201页

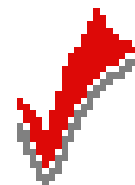
■ 9.1

■ 9.2

■ 9.3

■ 9.6

■ 9.9



习题9.1解答



- 当**8237A**的任意一个通道都没有**DMA**请求时就处于**空闲周期**
- 当**8237A**采样到外设有**DMA**请求时就脱离空闲周期进入**有效周期**

习题9.2解答1



单字节传送方式

- 每次**DMA**传送仅传送一个字节。传送一个字节之后，字节数寄存器减**1**，地址寄存器加**1**或减**1**，然后释放总线

数据块传送方式

- 由**DMA**请求信号启动，连续地传送数据，直到字节数寄存器从**0**减至**FFFFH**终止计数，最后释放总线



习题9.2解答2



单字节传送与数据块传送的根本区别

- 单字节传送方式：每次申请系统总线，仅传送一个字节，然后释放总线
- 数据块传送方式：每次申请系统总线，连续传送到规定的字节数传送完成，中间不释放总线

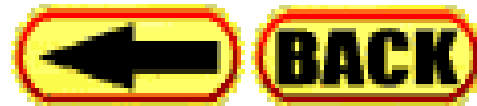


习题9.2解答3



数据块传送和请求传送对**DREQ**信号有效有什么要求

- 数据块传送方式：**DREQ**信号只需有效到外设**DMA**响应**DACK**有效
- 请求传送方式：**DREQ**信号必须在**DMA**传送过程中一直有效，否则传送暂停



习题9.3解答



- **DMA读: MEMR*和IOW*有效**
- **DMA写: MEMW*和IOR*有效**
- **DMA校验: 都无效**

习题9.6解答



```
out 0ch,al      ;清高/低触发器（可选）  
mov ax,1024-1   ;400h-1=3ffh  
out 05h,al  
mov al,ah  
out 05h,al
```


习题9.9解答



- 设置通道**3**
- 单字节传送方式
- **DMA**写传送类型
- 地址增量、禁止自动初始化

习题10——第232页

■ 10.4

■ 10.5

■ 10.6

■ 10.7

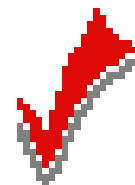
■ 10.8

■ 10.10

■ 10.11

■ 10.15

■ 10.16

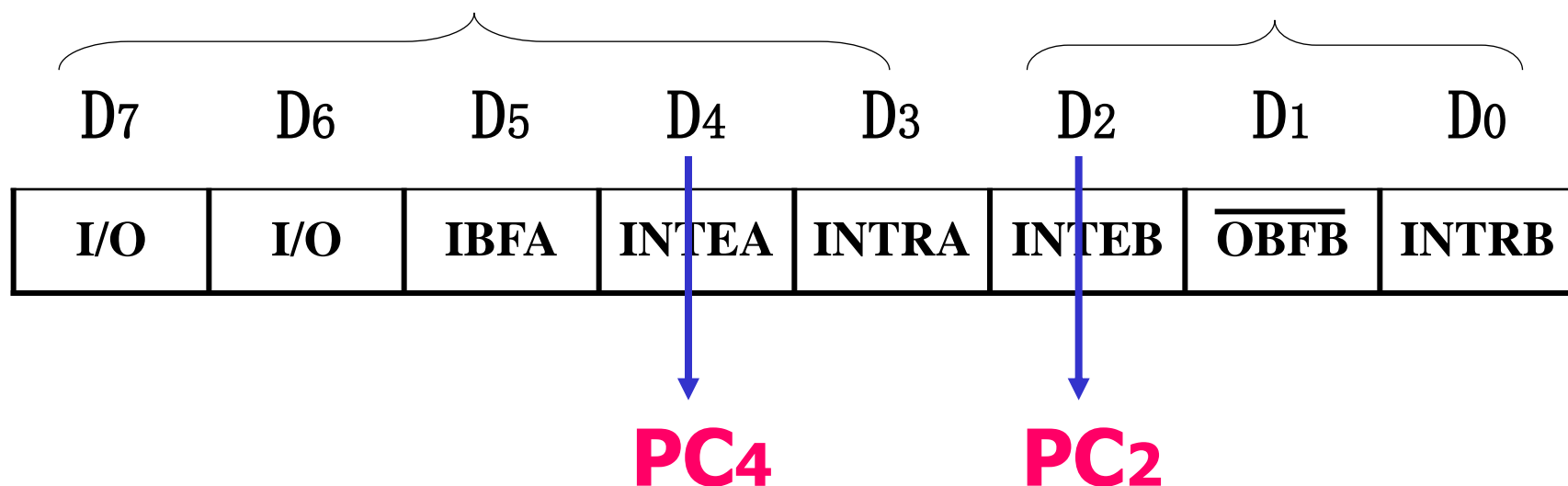


习题10.4解答



A组方式1输入

B组方式1输出



习题10.5解答



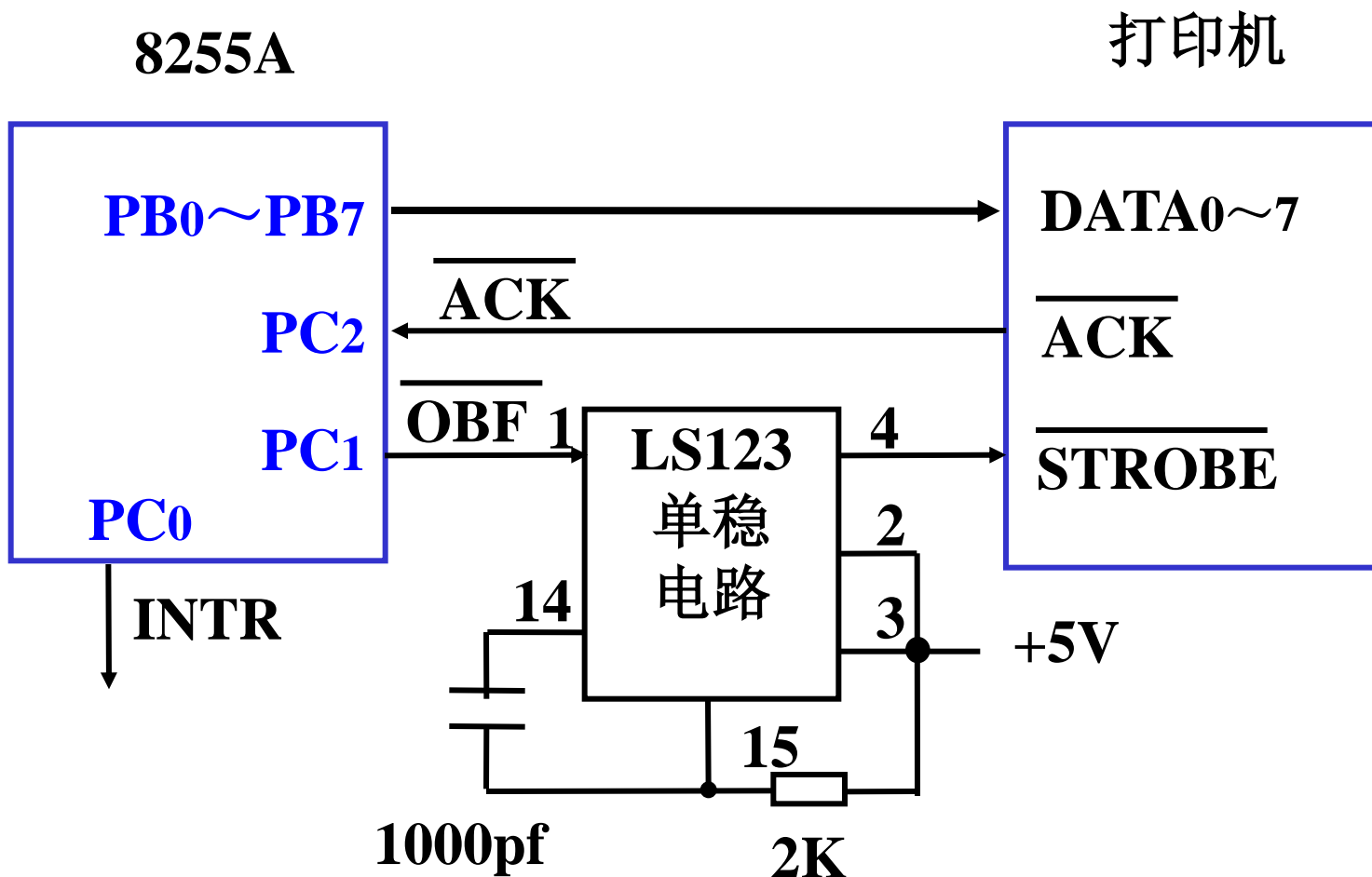
- 方式控制字为**B0H = 10110000B**，说明**A**组为方式**1**输入，它将征用**PC5**引脚作为输入缓冲器满信号**IBFA**

习题10.6解答



- 接口电路：用端口引脚**PB0~PB7**与打印机**DATA0~7**连接，其他不变
- 程序：用端口**B**的**I/O**地址**FFFAH**替换端口**A**的**FFF8H**
- 另外，应设置端口**B**为方式**0**输出。但由于原控制字已经做到，所以无需修改

习题10.7解答1



习题10.7解答2



mov dx,0ffeh

mov al,84h

out dx,al

mov al,04h

;使INTEB (PC2) 为0, 禁止中断

out dx,al

.....

mov cx,counter **;打印字节数送CX**

mov bx,offset buffer **;取字符串首地址**

call prints **;调用打印子程序**



习题10.7解答3



```
prints    proc
           push ax           ;保护寄存器
           push dx
print1:   mov al,[bx]        ;取一个数据
           mov dx,0ffhah
           out dx,al         ;从端口B输出
```



习题10.7解答4



```
mov dx,0fffh
print2:  in al,dx
         test al,02h      ;检测 (PC1) 为1否?
         jz print2
         inc bx
         loop print1
         pop dx
         pop ax
         ret
prints   endp
```



习题10.8解答1



； 写入方式字

mov al,100×00×1b ; =81h

mov dx,控制口地址 ; 0fffeh

out dx,al

； 加入下一段更好，使L0~L3全亮

mov al,0fh

mov dx,端口C地址 ; 0fffch

out dx,al



习题10.8解答2



； 控制程序段

mov dx,端口C地址 ; 0fffch

in al,dx ; 读入PC0~PC3

mov cl,4

shl al,cl ; 左移4位

out dx,al ; 控制PC4~PC7



习题10.10解答



- 当按下或释放一个键时，往往会出现按键在闭合位置和断开位置之间跳几下才稳定到闭合状态的现象就是机械按键的抖动
- 方法解决
 - 硬件：消抖电路
 - 软件：程序延时，以避开抖动的时间

习题10.11解答



- 重键指两个或多个键同时闭合
- 简单情况：不予识别，认为是错误的按键
- 通常情况：只承认先识别出来的键
 - 连锁法：直到所有键都释放后，读入下一个键
 - 巡回法：等被识别的键释放以后，就可以对其
他闭合键作识别，而不必等待全部键释放
- 正常的组合键：都识别出来

习题10.15解答



LEDtb db 3fh, 06h, 5bh, 4fh
db 66h, 6dh, 7dh, 07h
db 7fh, 6fh, 77h, 7ch
db 39h, 5eh, 79h, 71h

习题10.16解答1



； 显示代码表

```
table      db 0c0h ; 对应0（任意）  
           db 0f9h,0a4h,0b0h,99h  
           db 92h,82h,0f8h,80h  
           ; 对应1~8
```

； 8255A初始化

```
mov dx,0fffbh  
mov al,10001001b ; =89h  
out dx,al
```



习题10.16解答2



； 控制程序

```
again0:  mov dx,0ffhah    ; 输入开关状态
         in al,dx
         mov cx,8         ; 确定哪个闭合
         mov ah,01h      ; mov ah,08h
again1:  shr al,1         ; shl al,1
         jnc disp0
         inc ah           ; dec ah
         loop again1
         jmp disp1
```



习题10.16解答3



； 显示字段

disp0: mov bx,offset table

mov al,ah

xlat

mov dx,0fff8h

out dx,al ； 输出段码



习题10.16解答4



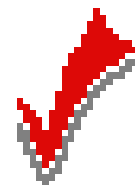
disp1: mov cx,8 ; 循环显示**8**位
 mov al,01h
 mov dx,0fff9h

disp2: out dx,al ; 输出位码
 call delay
 shl al,1
 loop disp2
 jmp again0



习题11——第249页

- 11.2
- 11.5
- 11.7
- 11.8



习题11.2解答1



发送数据 TxD

接收数据 RxD

请求发送 RTS

允许发送 CTS

数据装置准备好 DSR

信号地 GND

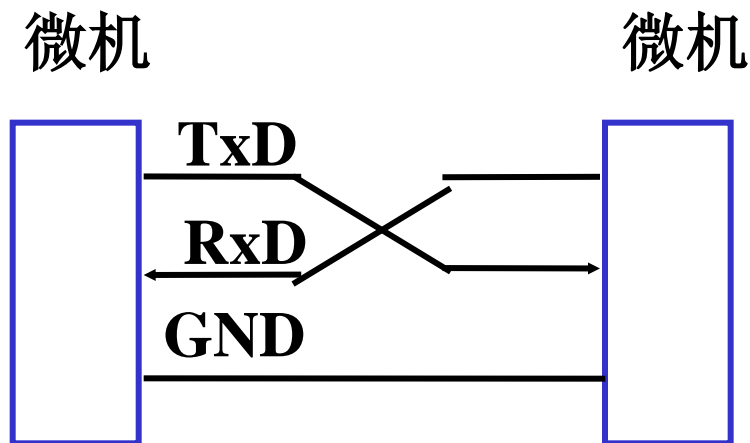
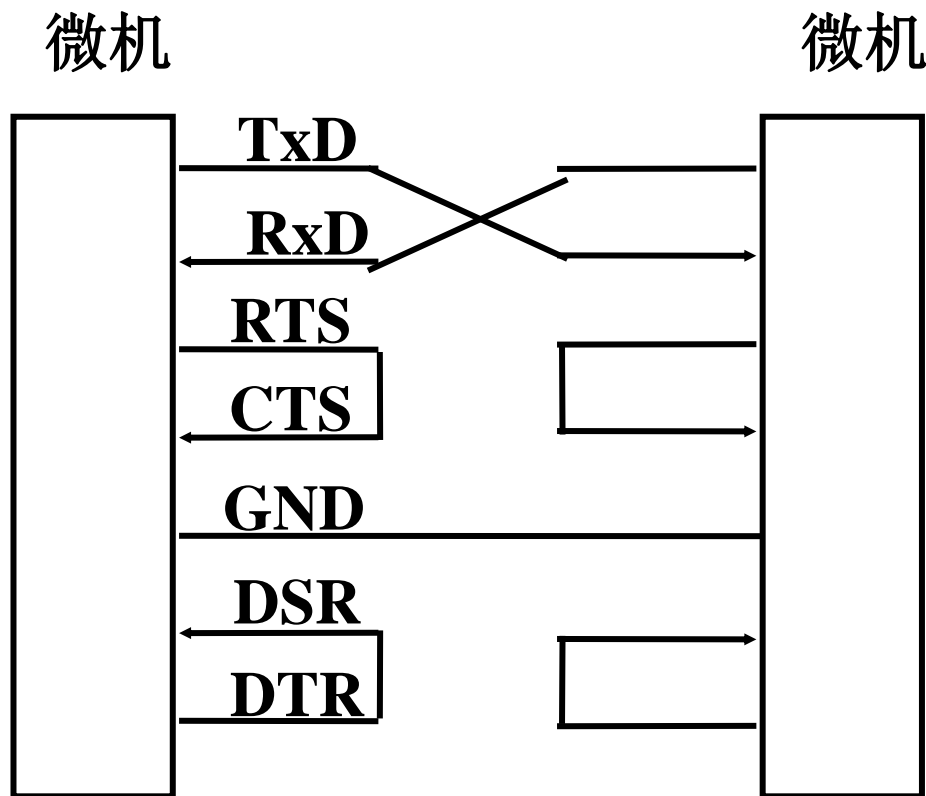
载波检测 CD

数据终端准备好 DTR

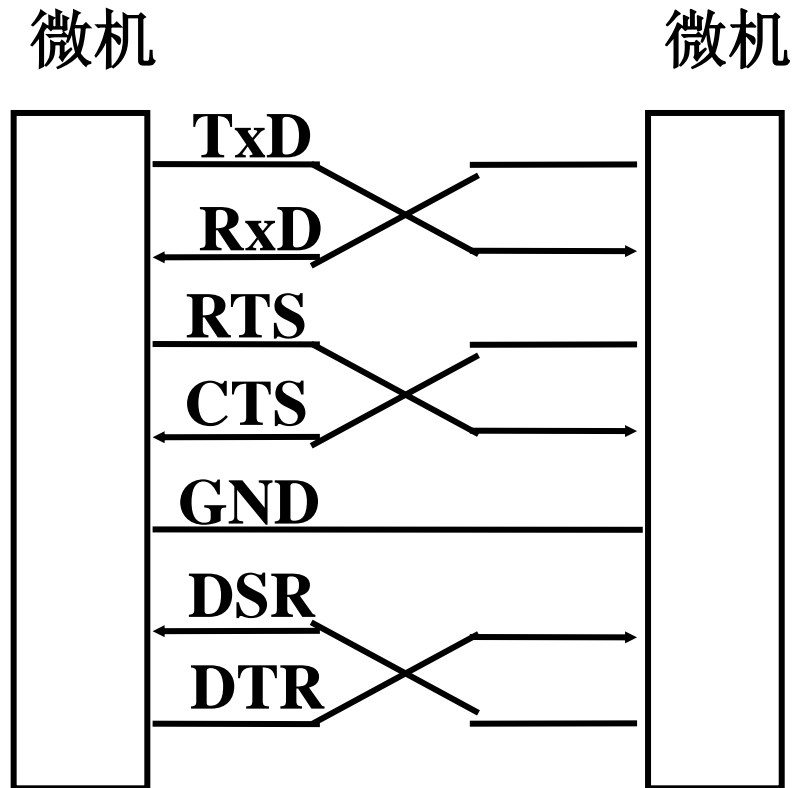
振铃指示 RI



习题11.2解答2



习题11.2解答3



习题11.5解答



- 欲使通信字符为**8**个数据位、偶校验、**2**个停止位，则应向**8250 通信线路控制**寄存器写入控制字 **00011111b (1fh)**，其在**PC**系列机上的**I/O**地址 (**COM2**) 是 **2fbh**
- **XT**机通信适配器电路上设计**J9~J12**跨接器的作用是 **选择两个I/O地址及中断请求引脚IRQ之一**

习题11.7解答



- **ROM-BIOS**中该段程序的作用是检测是否存在串行异步通信接口电路
- 如果不发生条件转移，说明存在异步通信接口电路，**RS232-BASE**字单元存放异步通信接口电路的基地址：**3f8h**和**2f8h**

习题11.8解答1



mov al,80h

mov dx,2fbh

out dx,al

mov ax,24 ; $1.8432\text{MHz} \div (4800 \times 16) = 24 = 18\text{H}$

mov dx,2f8h

out dx,al ;写入除数寄存器低8位

mov al,ah

inc dx

out dx,al ;写入除数寄存器高8位



习题11.8解答2



mov al,03h

mov dx,2fbh

out dx,al ;写入通信线路控制寄存器

mov al,0bh

mov dx,2fch

out dx,al ;写入调制解调器控制寄存器

mov al,01h ;至少有一位为1，不能禁止所有中断

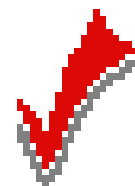
mov dx,2f9h

out dx,al

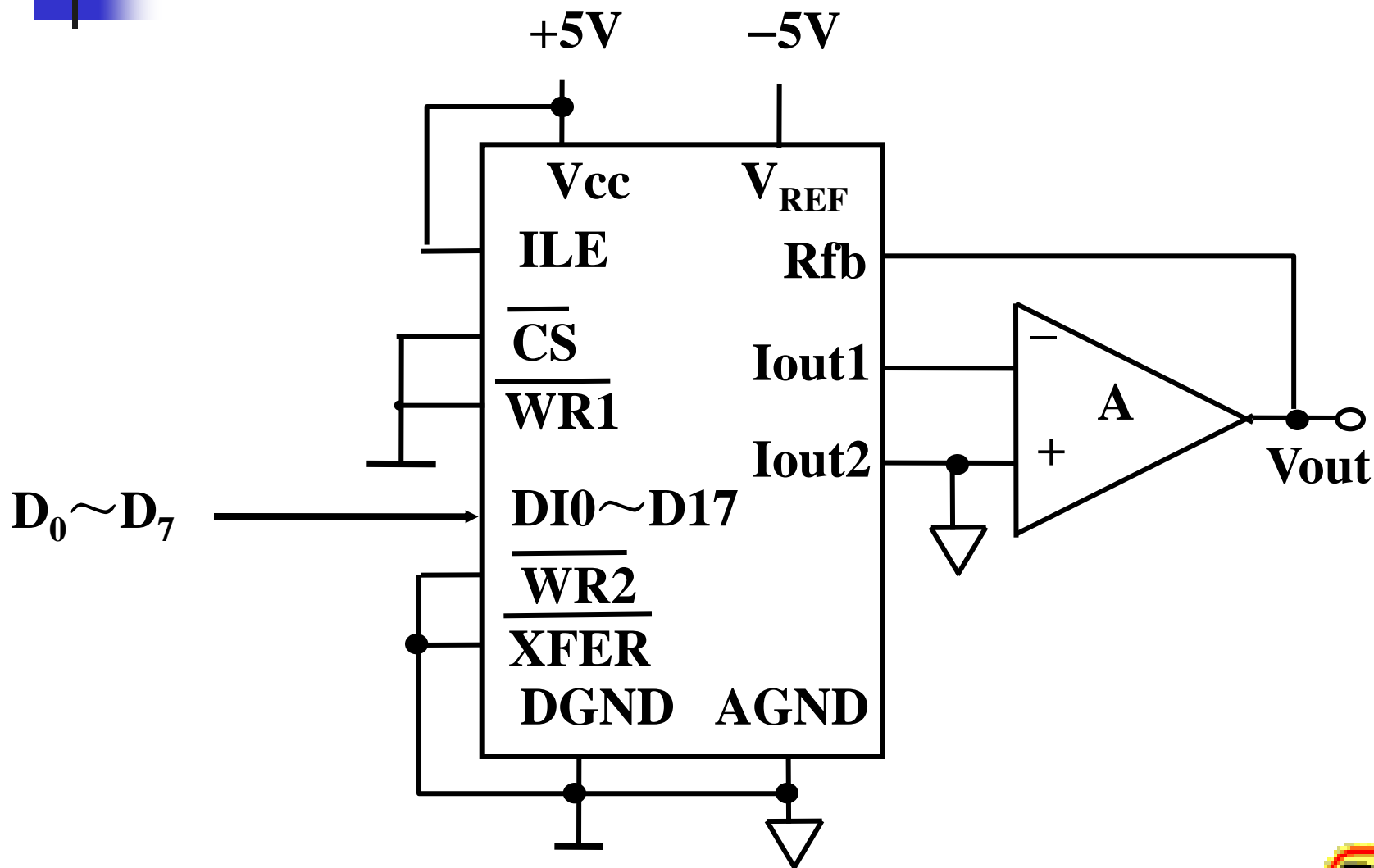


习题12——第264页

- 12.2
- 12.3
- 12.6
- 12.7



习题12.2解答

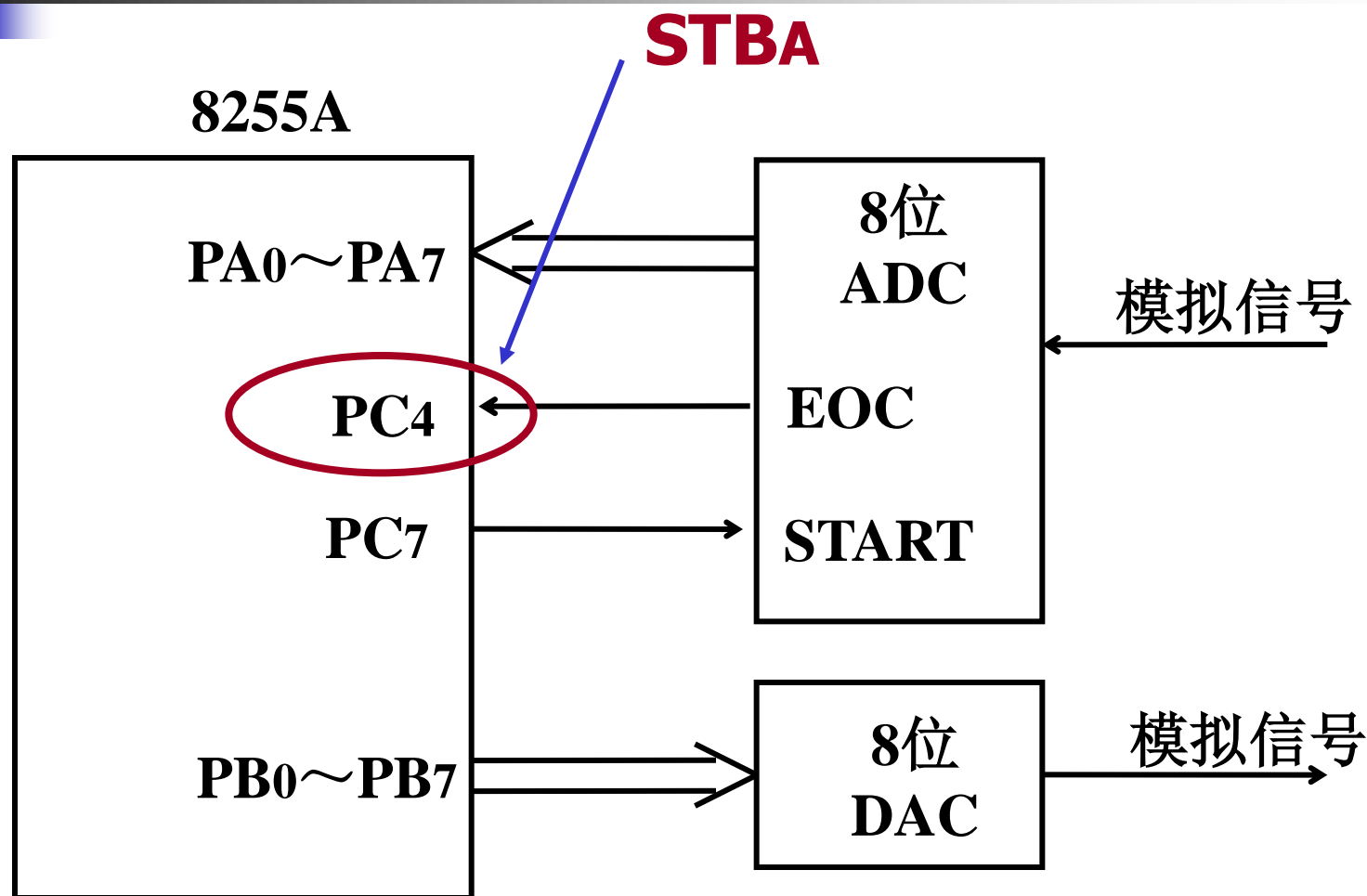


习题12.3解答



```
mov dx,port1l  
mov al,bl  
out dx,al  
mov dx,port1h  
mov al,bh  
out dx,al  
mov dx,port2  
out dx,al
```

习题12.6解答1



习题1.6解答2



； **8255A**初始化

mov al,1011000×b

mov dx,0fffbh

out dx,al

； 使**PC7=0**（**START**为低）

mov al,00001110b

mov dx,0fffbh

out dx,al



习题12.6解答3



；启动**A/D**转换

```
mov al,00001111b
```

```
mov dx,0fffbh
```

```
out dx,al; 使PC7=1 (START为高)
```

```
nop
```

```
mov al,00001110b
```

```
out dx,al; 使PC7=0 (START为低)
```



习题12.6解答4



PC5=IBF

； 查询是否转换结束

mov dx,0ffh

again: in dx,al

test al,20h

； **PC5=0**（转换未结束，继续检测）

jz again

； **PC5=1**（转换结束）



习题12.6解答5



```
mov dx,0fff8h ; 输入数据  
in al,dx  
call adprcs ; 处理数据  
mov dx,0fff9h  
out dx,al ; 输出数据
```



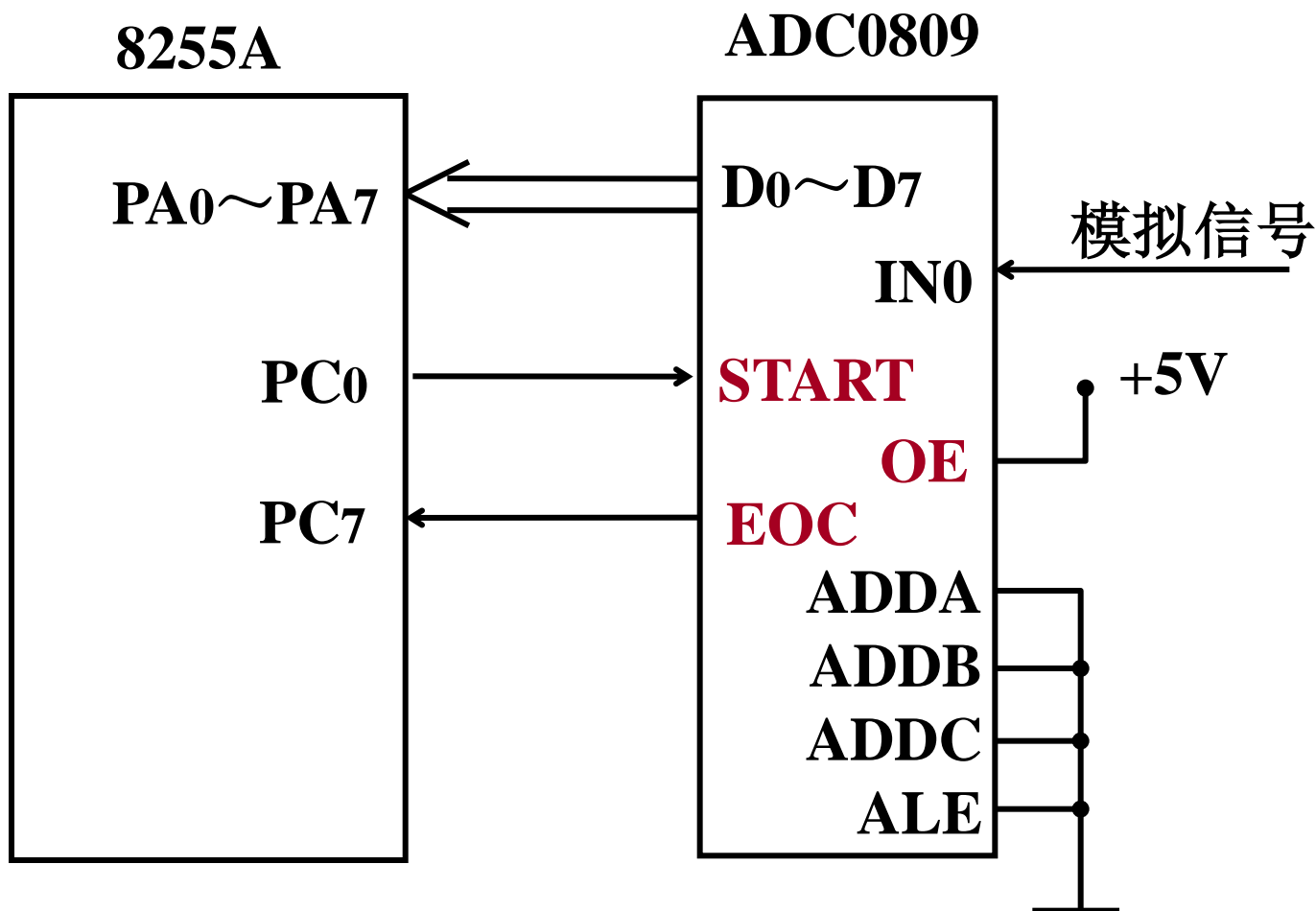
习题12.7解答1



- 采用**8255A**的**方式0**，端口**A**输入
- **PC0**接**ADC0809**的**START**，用于启动转换
- **PC7**接**ADC0809**的**EOC**，用于输入、判断**A/D**转换是否结束；采用**查询方式**
- **ADC0809**的**OE**接**+5V**，这样可以随时读取**A/D**转换的结果
- 假设**8255A**的**A**、**B**、**C**和控制口地址依次为**portA~portD**



习题12.7解答2



习题12.7解答3



； **8255A初始化**

mov al,100110×0b

mov dx,portd

out dx,al

； **使PC0=0（START为低）**

mov al,0000000b

mov dx,portd

out dx,al



习题12.7解答4



；启动**A/D**转换

```
mov al,00000001b
```

```
mov dx,portd
```

```
out dx,al; 使PC0=1 (START为高)
```

```
nop
```

```
mov al,00000000b
```

```
out dx,al; 使PC0=0 (START为低)
```



习题12.7解答5



； 查询是否转换结束

mov dx,portc

again: in dx,al

test al,80h

； **PC7=0**（转换未结束，继续检测）

jz again

； **PC7=1**（转换结束）



习题12.7解答6



； 输入数据

```
mov dx,porta
```

```
in al,dx
```



也可以采用方式**1**输入实现，
其中关键需要处理好**STB***信号，
以便将数字量送入**8255A**

