

微机原理与接口技术

第一至七章复习

第一章、知识要点

- 微机系统的组成：
 - 微机硬件核；系统总线；存储器；外设接口及外设。
- 三总线的概念：
 - 地址总线AB (Address Bus)
 - 数据总线DB (Data Bus)
 - 控制总线CB (Control Bus)

注意：

- 微机系统的硬件结构有哪几部分组成？
- 微机硬件核主要作用？
- 什么是三总线，它们各自的作用？
- 微机主要技术指标 (概念)？

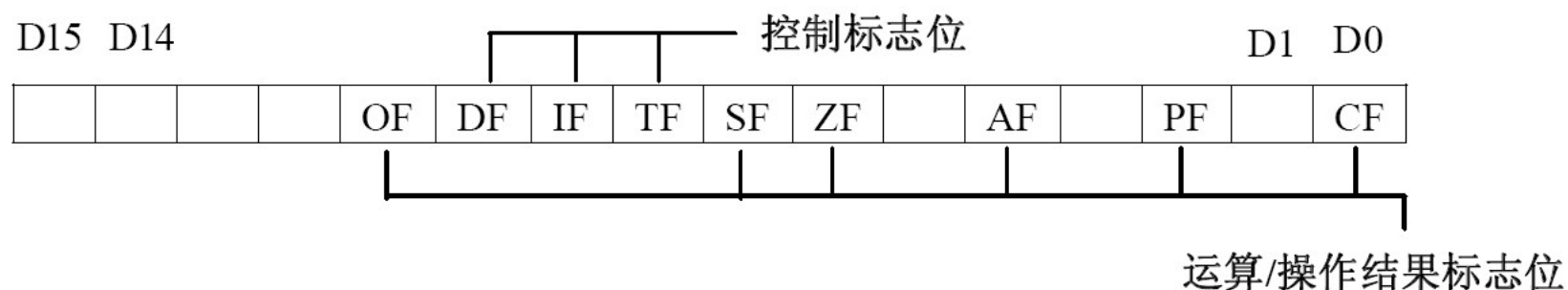
第二章、知识要点

● CPU的组成

- 由总线接口部件**BIU**(Bus Interface Unit)和指令执行部件**EU** (Execution Unit) 两大部分组成。
- **BIU**的主要作用是实现CPU对外部**三总线**的控制并与外部进行**数据交换**。具体的操作主要是根据指令的要求合成**20位**的地址信号及产生与外部总线数据传输需要的控制信号时序，最终实现与外部的数据交换。
- **EU**的作用则是**执行指令**。指令队列中的指令经EU控制系统转换成具体的操作控制信号并实现指令规定的运算或操作。
- 8088与8086在芯片内部是完全一样的，主要区别是在外部将数据线缩减成了8位（另有几根外部引脚定义不同）

寄存器及其作用（必须掌握）

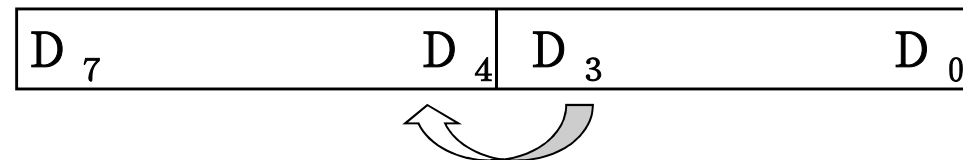
- 通用寄存器：AX, BX, CX, DX
- 指针及变址寄存器：IP, BP, SP, SI, DI
（只能以16位为单位进行操作）
- 段寄存器：CS, SS, DS, ES
- 标志寄存器（几乎每年都考）：



状态标志各位含义:

- 符号标志SF (Sign Flag): 和运算结果的最高位相同。指出前面的运算执行后的结果是正还是负, 结果为正数时SF=0, 负数时SF=1。
- 零标志ZF (Zero Flag): 当前运算结果为零时ZF=1; 当前的运算结果非零时ZF=0。
- 奇/偶标志PF (Parity Flag): 若运算结果的低8位中所含的1的个数为偶数, 则PF=1, 否则为0。
- 进位标志CF——当执行加法运算使最高位产生进位, 或者执行减法运算引起最高位产生借位时, 则CF=1。除此之外, 循环指令也会影响此标志。(无符号数)
- 溢出标志OF——当运算过程中产生溢出时, 会使OF为1。所谓溢出, 就是当字节运算的结果超出了范围-128~+127(2^7-1), 或者当字运算的结果超出了范围-32768~+32767($2^{15}-1$)时称为溢出。(有符号数)

- 辅助进位标志AF——加法运算时第3位往第4位有进位，或者减法运算时第3位从第4位有借位，则AF为1。此标志在BCD码运算中作为是否进行十进制调整的判断依据。



状态标志举例:

AF

	0101	0100	0011	1001	(21561)
+	0100	0101	0110	1010	(17770)

CF=0 **SF=1** **ZF=0** **AF=1** **PF=1** (4个“1”)

OF=1 (两个正数相加后，结果为负数，溢出)

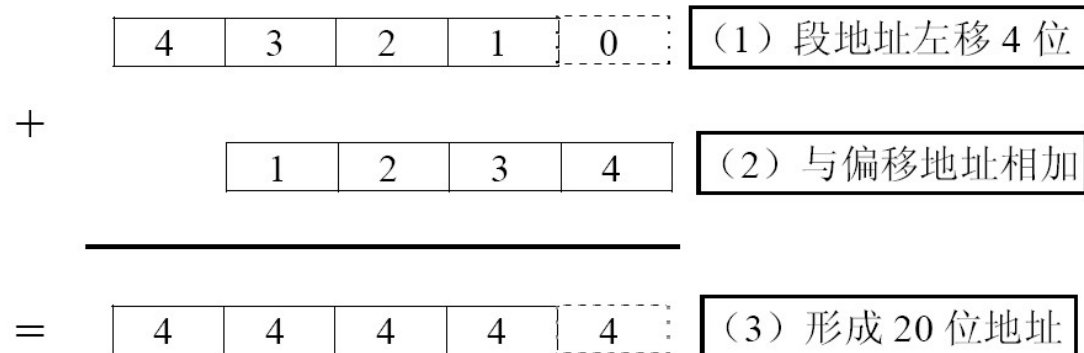
控制标志有3个，即DF、IF、TF

- 方向标志DF（Direction Flag）——控制串操作指令用的标志。如果DF为0，则串操作过程中地址会不断增值；如果DF为1，则串操作地址会不断减值。
- 中断标志IF（Interrupt Enable Flag）——可屏蔽中断允许标志。如果IF为0，则CPU不能对可屏蔽中断请求作出响应；如果IF为1，则CPU可以接受可屏蔽中断请求。
- 跟踪标志TF（Trap Flag）——如果TF为1，则CPU按跟踪方式执行指令。此方式便于进行程序的调试。

微处理器寻址内存地址的方法

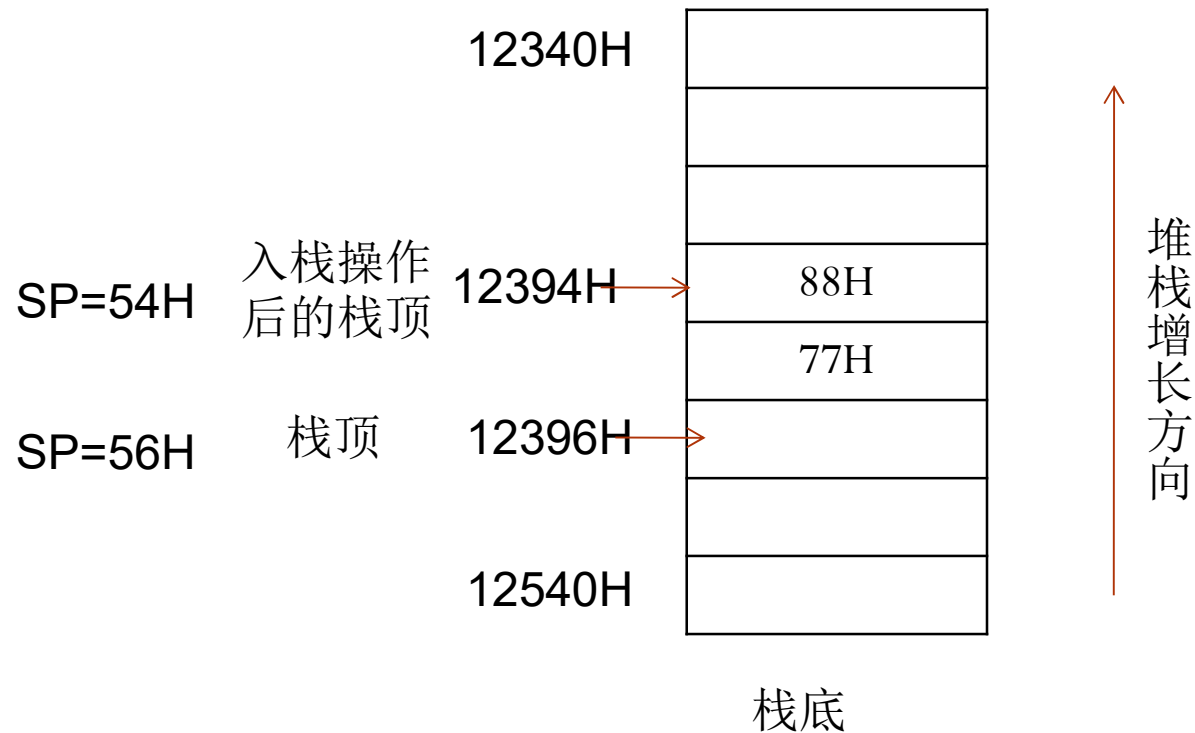
设：段地址寄存器内容
为：4321H；偏移地址
值为：1234H

最终物理地址
为：44444H



堆栈（常考）

6. 以知给定堆栈区的地址范围为：1250H: 0000H~1250H: 0100H, (SP)=0052H 问：（1）栈顶地址是什么？（2）栈底地址是什么？（3）SS中的内容是什么？（4）若在堆栈中存入数据2345H, 请画出堆栈区图, 标明数据存放位置及对应的段地址和偏移地址。



如何实现地址线与数据线分时复用的？

- 8086按16位传输数据的，有16个地址/数据复用引脚。作为地址/数据复用引脚，在总线周期的T1状态用来输出要访问的存储器或I/O端口的地址，T2~T3状态，对于读周期来说，是处于浮空状态，而对于写周期来说则传输数据。
- 在最小模式下ALE信号为地址锁存允许信号输出，在任一总线周期的T1状态，ALE输出有效电平，以表示输出的是地址信息，ALE信号不可浮空。



掌握
概念

什么是总线周期、基本总线周期和时钟周期？

- CPU每完成一次总线操作（与外部更换一次数据；一次读或一次写）称为一个“总线周期”。
- CPU与外部交换一次数据至少需要4个时钟周期，被称为基本总线周期。
- CPU主时钟周期也简称为“时钟周期”，是计算机内部的最小计时单位，是CPU主时钟频率的倒数。



掌握
概念

MOV IN OUT 的区别（必须掌握）

- MOV 寻址存储器，访问范围 $2^{20} = 1\text{M}$
- IN、OUT 寻址外设，访问范围 $2^{16} = 64\text{k}$

CPU是如何选择存储器奇、偶地址单元的？为什么存取存储单元字数据时一定要从偶数地址开始？

- 1) 8086CPU 的存储器由两个 8 位数据宽度的存储体构成，分别由奇数地址和偶数地址访问。奇地址存储体的片选信号是 BHE，偶地址存储体的片选信号是 A0。
- 2) 字的存取最好始于偶地址（A0=0）；偶地址字节读写仅 AD0~AD7 为有效数据；奇地址字节读写仅 AD8~AD15 为有效数据。
- 3) 始于偶地址的字读/写可在一个总线周期内完成，而始于奇地址的字读/写至少需要两个总线周期。所以尽量要在偶地址读/写字数据。

不考

- 最大模式下的时序
- 2.4 设计实例与实践

第五章、知识要点

- 外设端口（或I/O 端口）
- CPU 与外设的数据传送方式（查询式传送方式与中断传送方式作比较）
- 译码器
- 端口电路图。

什么是外设端口（或I/O 端口）？通常有哪些类型的外设端口？

- 接口电路内部必须具备一些不同功能的可以由CPU三总线按特定地址进行访问的单元（寄存器），通常将它们称之为外设端口或I/O 端口。
- （1）输入数据寄存器和输出数据寄存器组（数据中转型接口中用于暂存数据）
- （2）控制寄存器组（用于接受CPU 的控制命令）
- （3）状态寄存器组（用于提供CPU 可查询的状态信息）

CPU 与外设的数据传送有哪几种方式，它们各有哪些优缺点？

CPU 的运行速度通常都高于外设的运行速度，但因外设的种类繁多，所以需要 CPU 具备多种速度匹配方式针对不同响应速度的外设灵活选用。当 CPU 通过接口电路与 I/O 设备交换数据时通常采用的速度匹配方式有以下几种：

1. 直接传送方式

即无等待传送。CPU 两次 IN/OUT 操作间不延时。用于外设速度足够快的情况。

2. 程序延时传送方式

在两次 IN/OUT 指令之间插入一些无意义的指令（通常为循环指令），通过这些指令运行所耗费的时间来实现短时间延时等待的传送方式。这种方式需先计算出所用指令的执行时间（时钟周期数 \times CPU 时钟周期），然后再计算指令的执行次数，经循环执行后得到外设所需的延时时间。

3. 查询传送方式

在一次 IN/OUT 指令之后，CPU 通过不断查询（读取）接口芯片提供的就绪信号来决定是否执行下一条 IN/OUT 指令的传送方式。

4. 中断传送方式

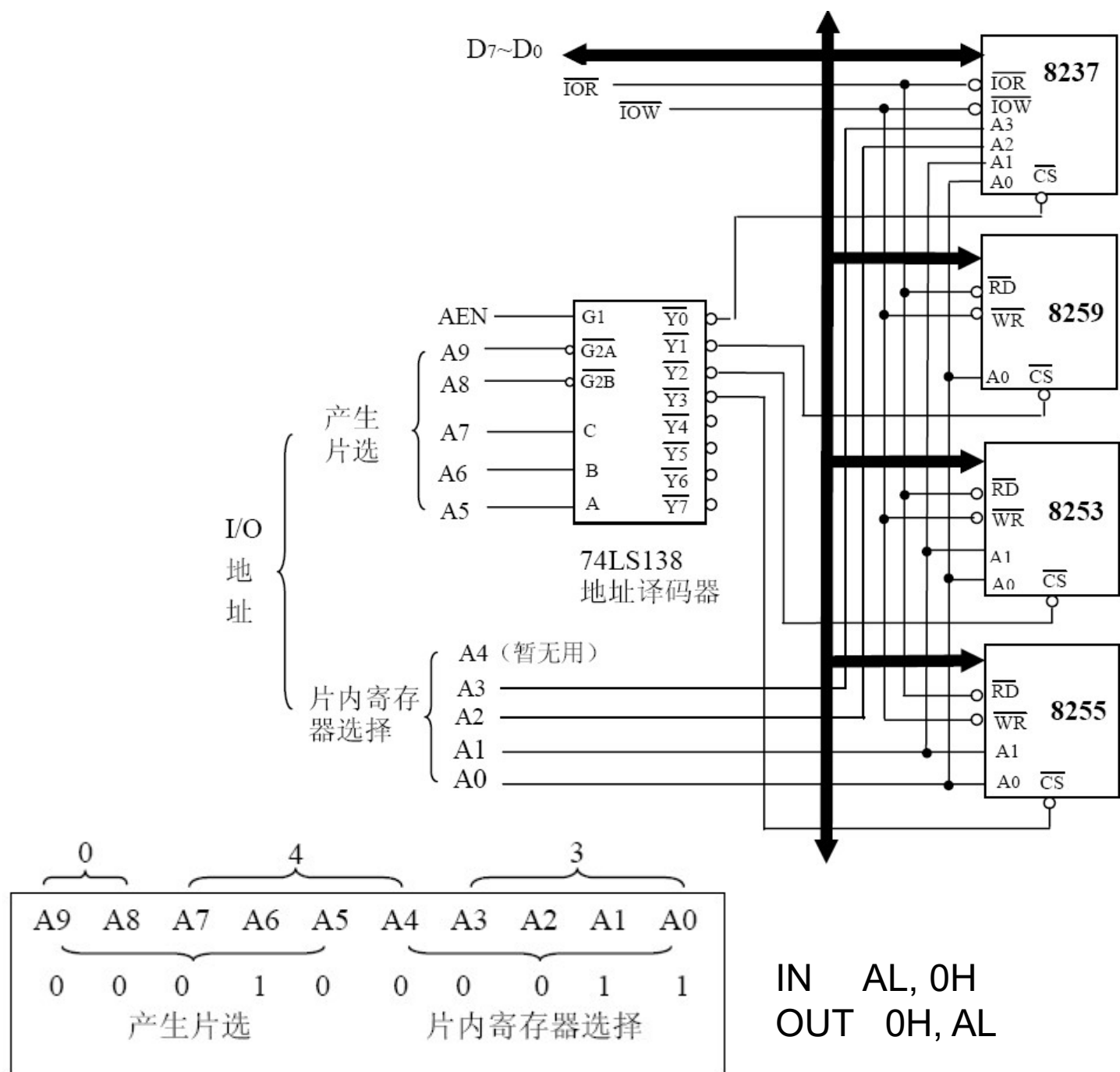
CPU 在没有中断请求的时候会运行主程序或其它程序。当外设准备好后发中断请求，CPU 响应中断并转入中断处理程序内去执行一次 IN/OUT 操作。中断处理程序结束后又返回主程序运行，直到下一次中断的产生。

5. DMA 传送方式（仅用于连续大块数据的传送）

由 DMA 控制器取代 CPU，在内存与外设之间进行的成块数据传送。

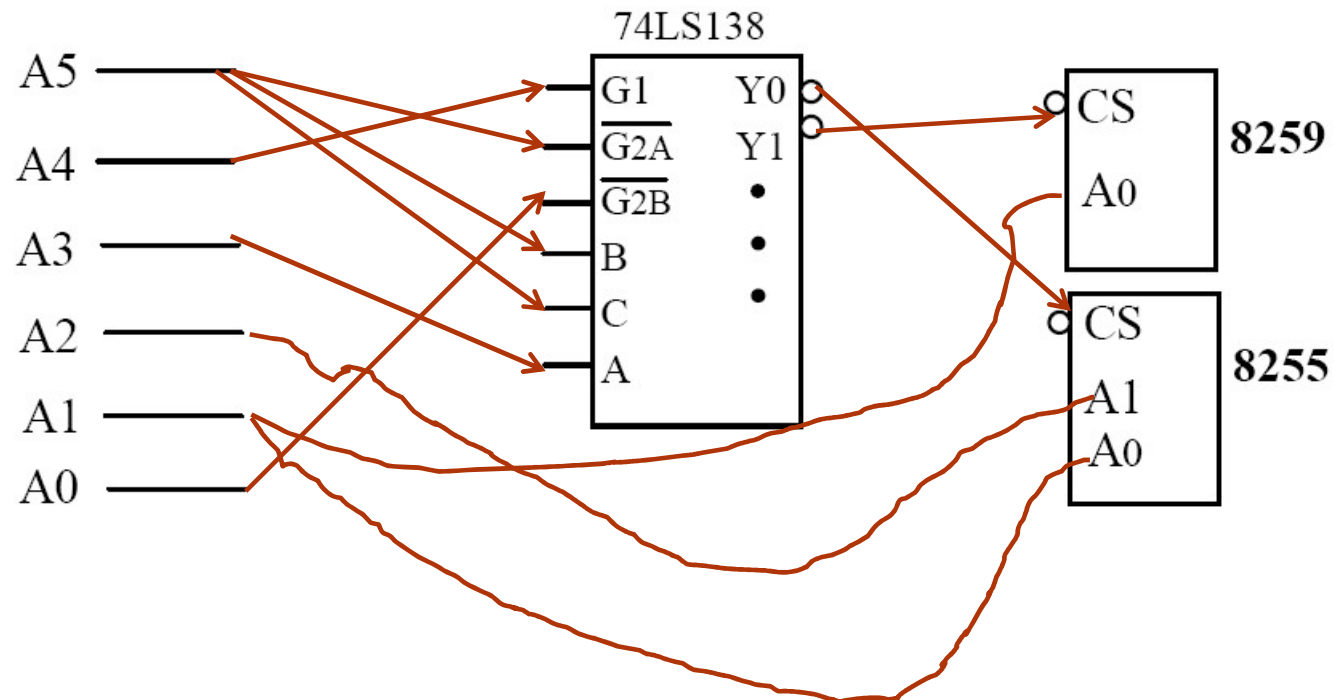
9. 80X86 CPU 最多可寻址多少个I/O 端口。如果微机内共有7 个接口芯片，其中一个接口芯片有6 个独立端口地址，CPU 应如何寻址这些端口？请画出原理性电路，并写出读写这些端口的指令。

- 80X86 系列CPU用于外设数据传送的IN/OUT指令最多可以驱使A15~A0 共16 根地址线。即I/O空间为 $2^{16} = 65536$ （I/O端口地址）。
- IN AL, 0H
- OUT 0H, AL



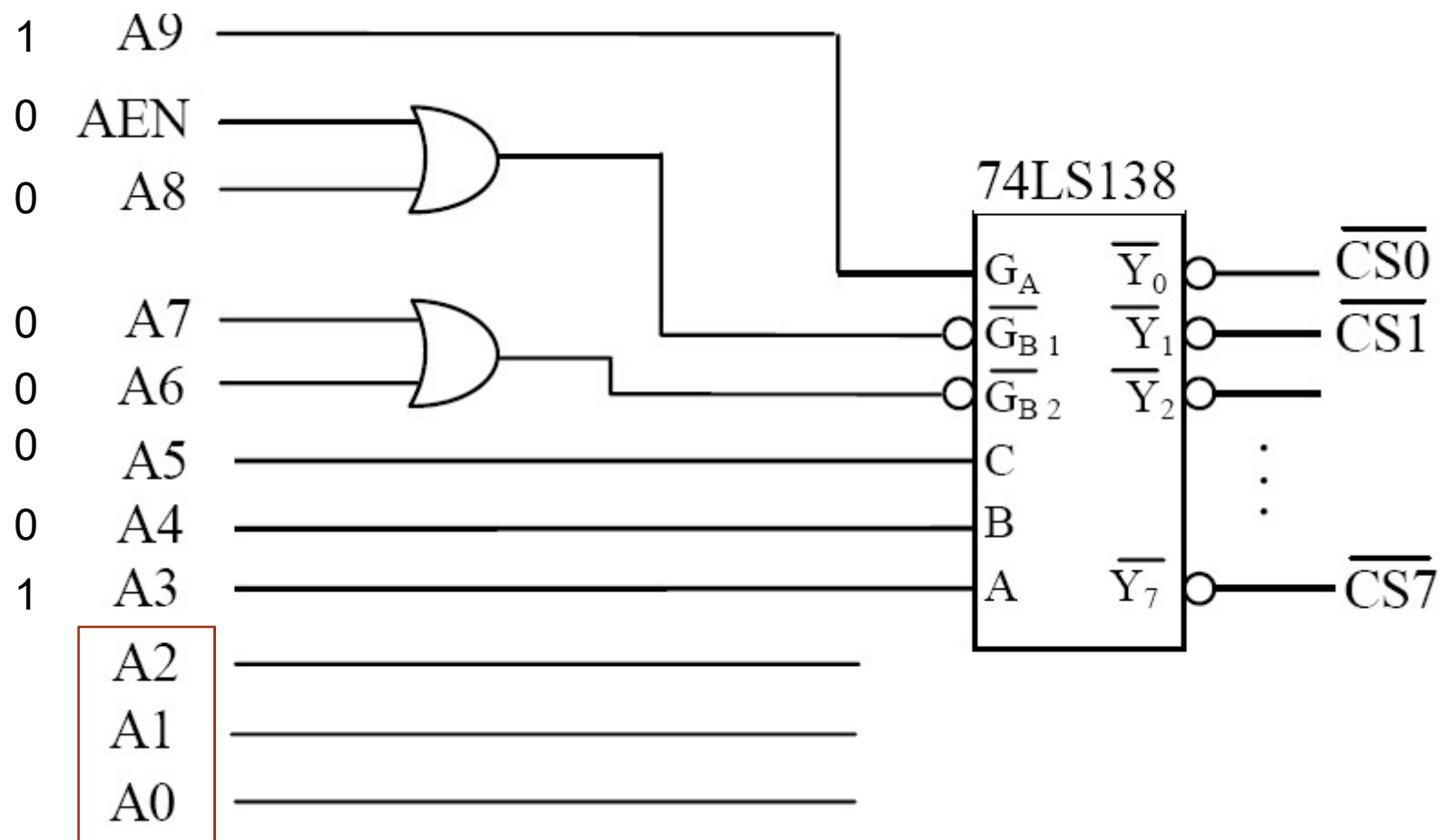
74LS138 译码器

如果只采用A5~A0 共6 根地址线对8255 和8259 进行地址译码，且要求8255 □地址为10H、12H、14H、16H；8259 的□地址为18H、1AH。请在下图画出正确连线。



12. 针对 I/O 地址译码

(1) 请指出以下地址译码电路 $\overline{CS1}$ 对应的 I/O 地址范围为 208H~20FH。

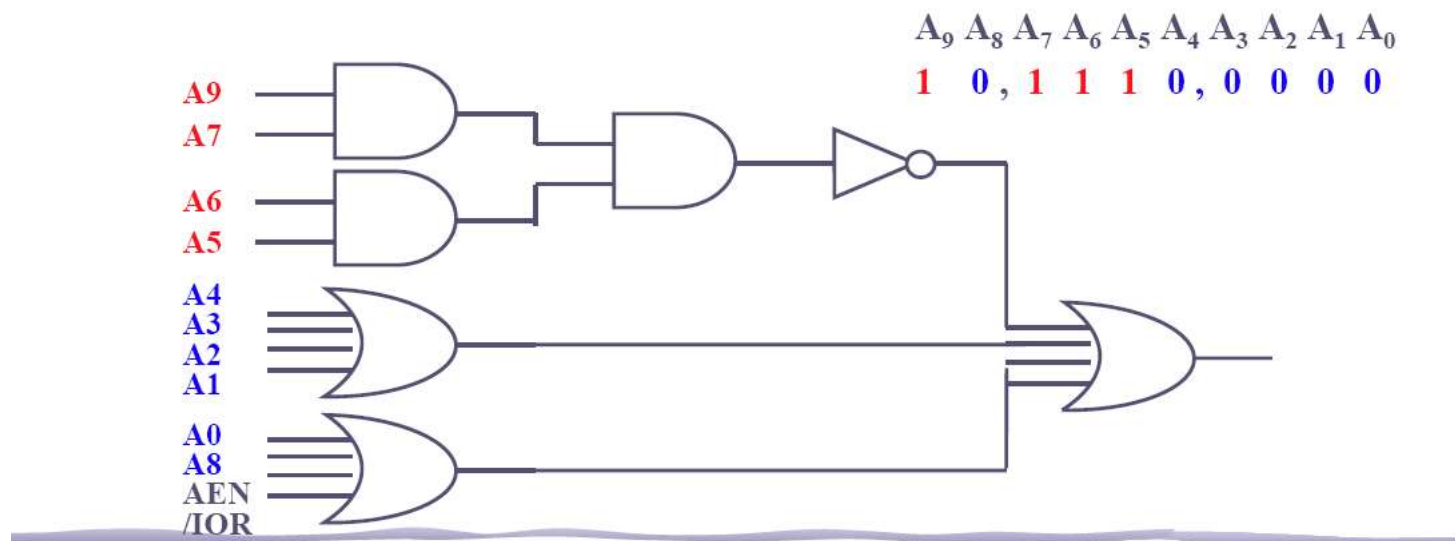


(2) 用两输入“与门”、四输入“或门”和单输入反向器，对 A9-A0、IOR（低电平有效）、AEN（低电平有效）实现 I/O 端口 2E0H 的译码（画出电路）。

解：（1）三态门用于隔离开关状态和CPU数据总线，只有在执行读**2E0H**端口时，才允许三态门打开将状态读入。

上拉电阻的作用是在开关**K**打开时使状态成为高电平，而不是悬空状态。

（2）读**I/O**端口**2E0H**的译码电路如下：



第六章、知识要点

- 8255 的三个端口。
- 端口的初始化。
- 键盘扫描。

8255 有哪几种工作方式、各适用哪些端口。在这些方式中端口C 各起什么作用？

由前面的内容可以看到，8255 有三种工作方式可供选择。但是 A、B、C 三个端口能够选择的工作方式以及在每种工作方式中所扮演的角色是不同的。其中：

- 端口 A 可选择全部三种工作方式，而且都是作为与外设的数据 I/O 线。
- 端口 B 只能选择 0、1 两种工作方式且都是作为与外设的数据 I/O 线。
- 端口 C 在工作方式 0、1、2 下都可以工作，但自己却没有工作方式选择权。只有 A、B 口都选择工作在方式 0 时，C 口才可以作为两个独立的 4 位并行端口被编程为输入或输出数据线整体使用。方式 1 下端口 C 将被分割成 PCA 和 PCB 两部分，分别从属于端口 A 和端口 B，作为它们的信号联络线。方式 2 下端口 C 将全部从属于端口 A，作为它的信号联络线。（参见前表 6-1 和 6-2）

1. 方式0（无固定信号联络线的输入/输出方式）
2. 方式1（有信号联络线的输入/输出方式）
3. 方式2：（双向数据传送方式，只适用于端口A）

3. 若8255 端口A 设定为方式0，端口B 设定为方式1，则端口C 还有哪些引脚可分别编程为输入和输出，对应的控制字是怎样的？

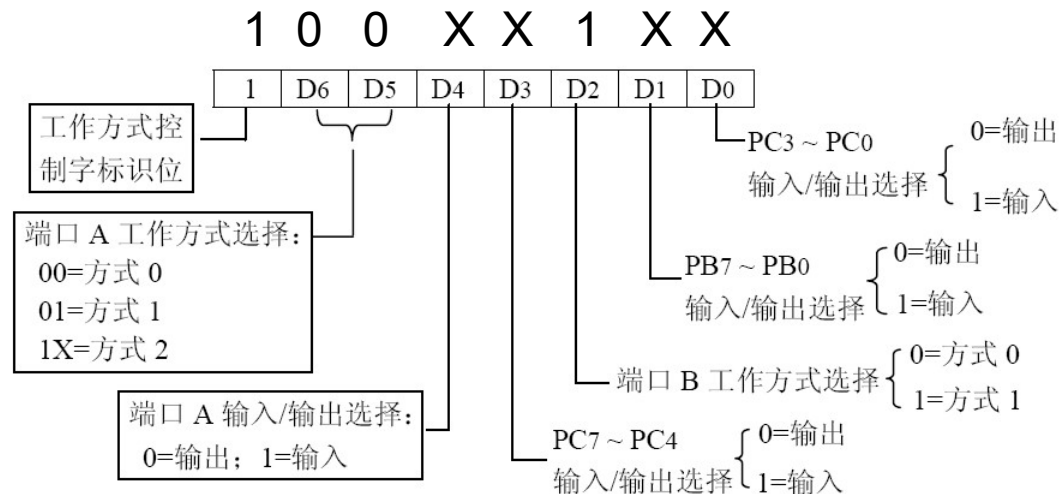
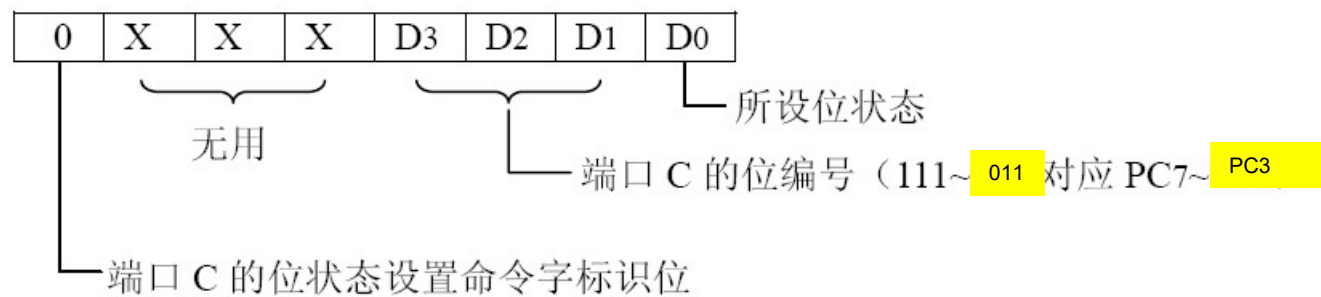


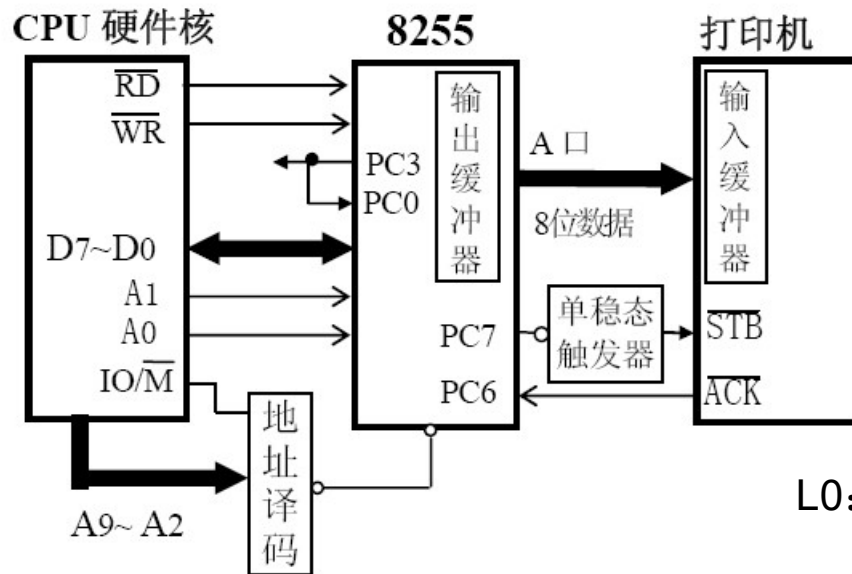
表 6.1 8255 的 A 口 B 口在方式 1 时占用 C 口引脚对应表

端口及 I/O	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
端口 A 输入	待用	待用	IBF	$\overline{\text{STB}}$; INTE	INTR	待用	待用	待用
端口 B 输入	待用	待用	待用	待用	待用	$\overline{\text{STB}}$; INTE	IBF	INTR
端口 A 输出	$\overline{\text{OBF}}$	$\overline{\text{ACK}}$; INTE	待用	待用	INTR	待用	待用	待用
端口 B 输出	待用	待用	待用	待用	待用	$\overline{\text{ACK}}$; INTE	$\overline{\text{OBF}}$	INTR

PC3-PC7



5. 如果设定8255 端口A 工作于方式1 输出数据, 且CPU 采用查询方式而非中断方式向端口A 写数据。请画出CPU 与8255 之间、8255 与外设之间的原理性电路连接图并写出有关的程序段。设端口A、B、C 及控制寄存器口地址为0D0H、0D2H、0D4H、0D6H。



MOV CX, 100 ; 设置计数值
LEA SI, BUF ; 取数据缓冲区偏移地址
MOV DX, 0D6H ; 控制寄存器口地址
MOV AL, 10101001B ; A 口方式1 输出;
PC7~PC4 输入; PC3~PC0 输入。
OUT DX, AL ;
MOV AL, 00001101 ; 位设置命令字, 将
PC6 置1
OUT DX, AL ; 用于打开A 口方式1 下8255
内部的中断允许位

L0: MOV DX, P8255 ; A 口地址
MOV AL, [SI] ; 取打印数据
INC SI ; 指向下一数据
OUT DX, AL ; 向8255 的A 口送打印数据
MOV DX, P8255+2
L1: IN AL, DX ; 读PC0 状态
AND AL, 00000001B ; 判断PC0 是否为1
(PC3=INTR=1 表示输出缓冲器空)
JZ L1 ; PC0 不为1 继续查询
LOOP L0 ; PC0 为1, 进行下一次数据输出
程序结束 ; 数据全部输出完毕

5. 如果设定8255 端口A 工作于方式1 输出数据, 且CPU 采用查询方式而非中断方式向端口A 写数据。请画出CPU 与8255 之间、8255 与外设之间的原理性电路连接图并写出有关的程序段。设端口A、B、C 及控制寄存器口地址为0D0H、0D2H、0D4H、0D6H。

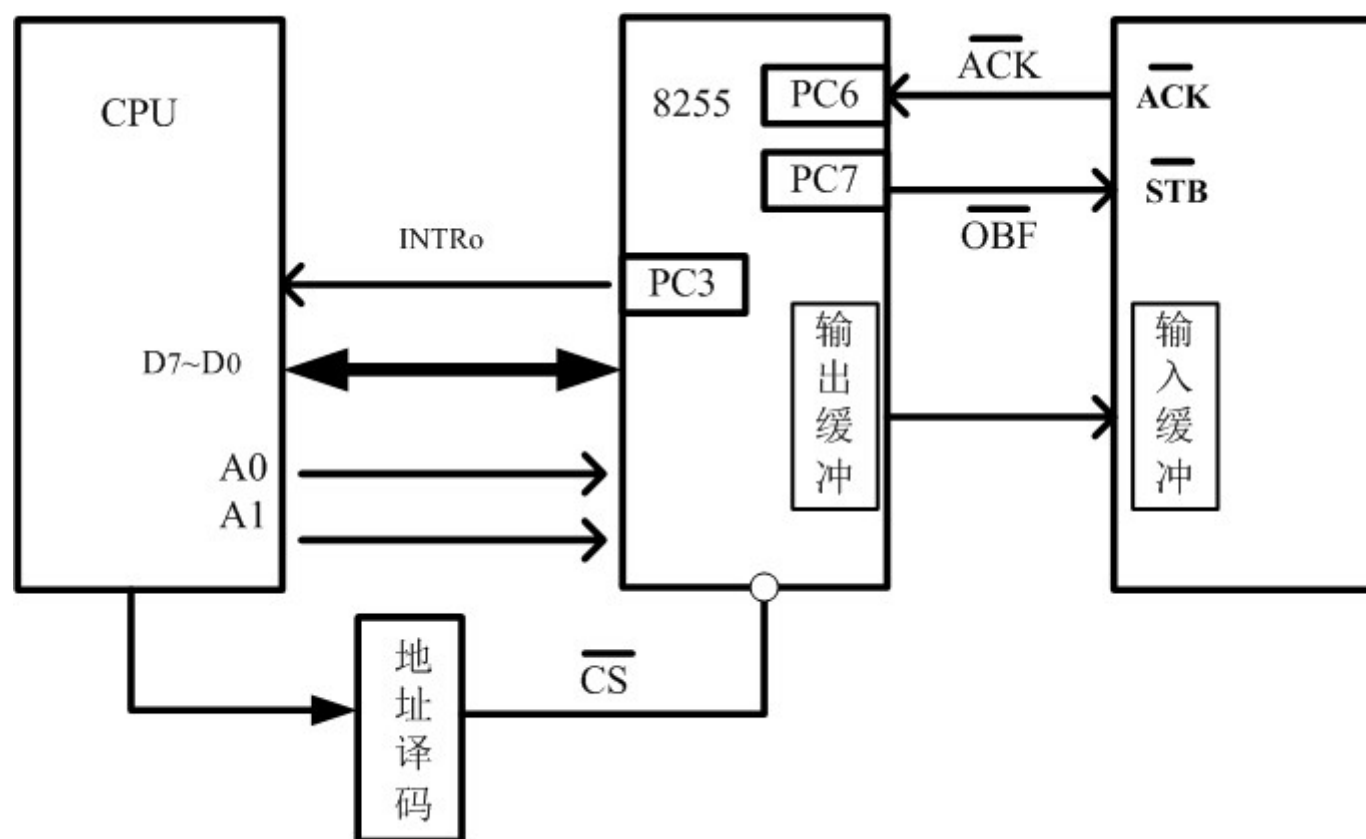
```
MOV CX, 100 ; 设置计数值
LEA SI, BUF ; 取数据缓冲区偏移地址
MOV DX, 0D6H ; 控制寄存器口地址
MOV AL, 10101001B ; A 口方式1 输出; PC7~PC4 输入; PC3~PC0 输入。
OUT DX, AL ;
MOV AL, 00001101 ; 位设置命令字, 将PC6 置1 INTE=1
OUT DX, AL ; 用于打开A 口方式1 下8255 内部的中断允许位
L0: MOV DX, 0D0H ; A 口地址
    MOV AL, [SI] ; 取数据
    INC SI ; 指向下一数据
    OUT DX, AL ; 向8255 的A 口送数据
    MOV DX, 0D4H
L1: IN AL, DX ; 读PC0 状态
    AND AL, 00000100B ; 判断PC3 是否为1 (PC3=INTR=1 表示输出缓冲器空)
    JZ L1 ; PC3不为1 继续查询
    LOOP L0 ; PC3为1, 进行下一次数据输出
程序结束 ; 数据全部输出完毕
```

6. 以下是某个采用 8255 中转 8X86 CPU 与外设数据的应用系统中对 8255 设置的有关程序段，分析该程序并：

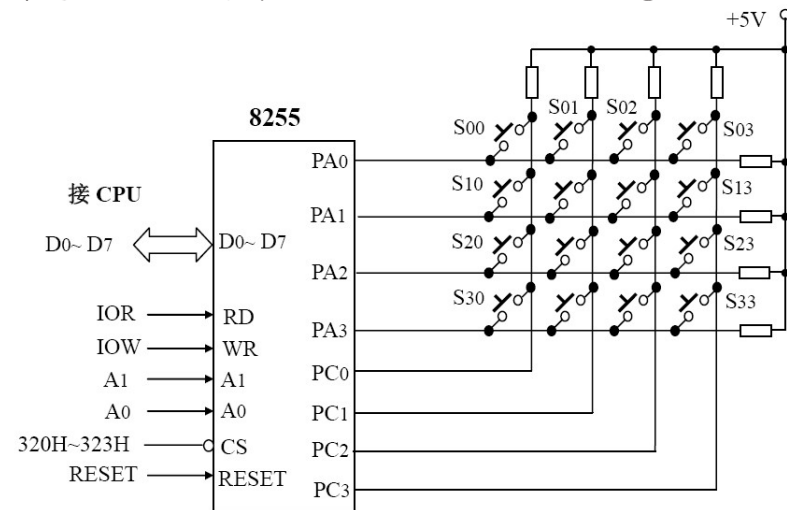
- (1) 指出 A、B、C 口各工作于什么方式
- (2) 画出该应用系统的电路连接原理图中必要的引脚及连线。

```

                                1010 0000 A口工作于方式1，输出，B口工作于方式0，输出。
MOV     AL, 0A0H                PCA从属于端口A
OUT     P8255+6, AL
MOV     AL, 1                    PC0 = 1
OUT     P8255+6, AL
.       .
.       .
                                0000 1101 PC6 = 1 C口工作于中断方式
MOV     AL, 0DH
OUT     P8255+6, AL
STI
```



7. 请画出3行×3列，行列式键盘的电路原理图并说明其工作原理。



(1) 行扫描法:

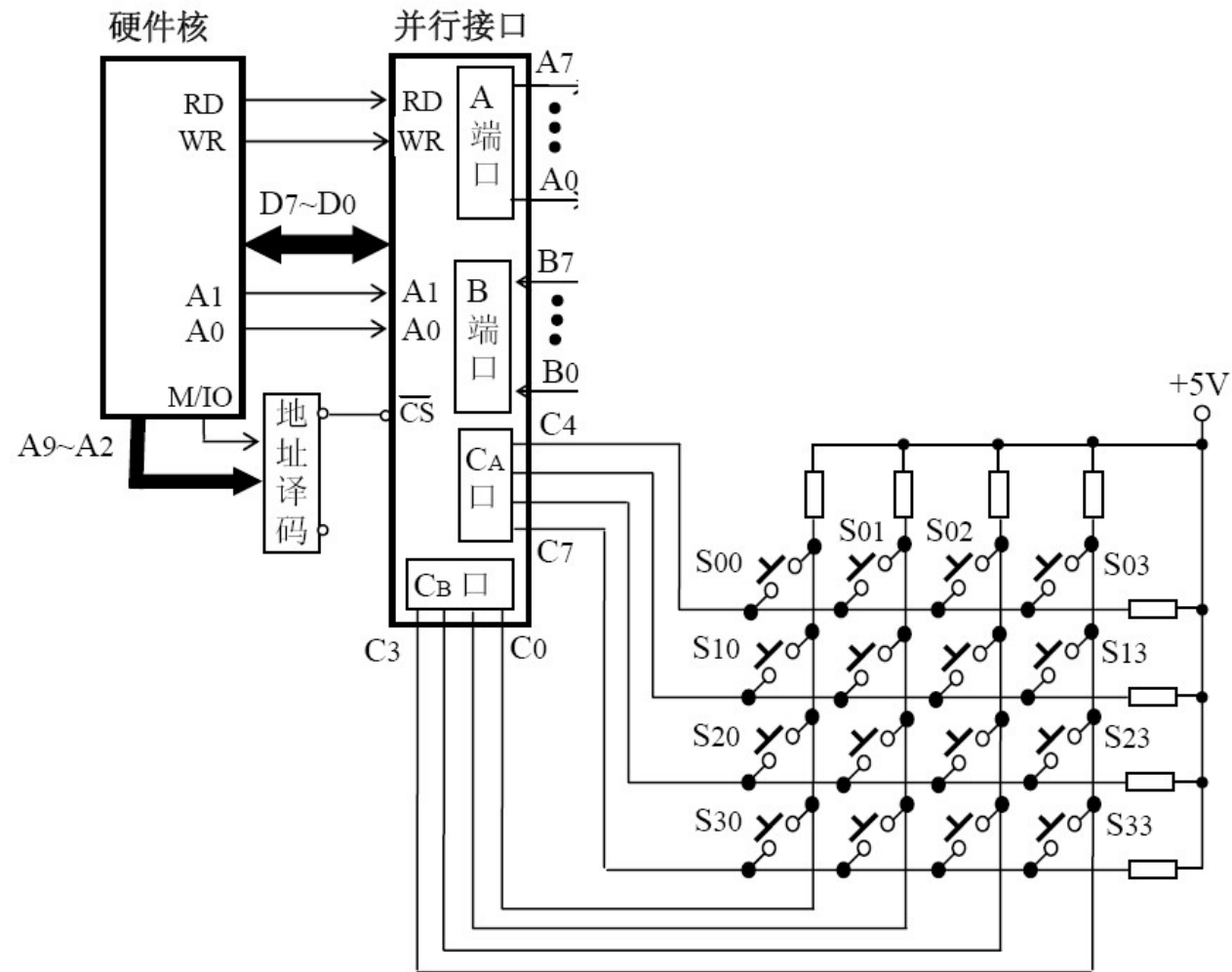
特点: 先输出一组仅有一位为“0”的行值，再读取列值并判断是否有为“0”的位。若有，则说明有键按下，且其行、列值将唯一确定按键的位置。这种方式下采用固定的行输出寄存器和列输入寄存器。4位行值和4位列值组成8位的键码。

(2) 行反转法:

特点: 先输出一组所有位为“0”的行值，再读取列值并判断是否有为“0”的位。若有，则说明有键按下。然后将刚读入的列值输出，再读取行值，从而由行、列值唯一确定按键的位置。

这种方式下行、列寄存器为双向传送数据。先是行寄存器为输出，列寄存器为输入。而后行寄存器变为输入，列寄存器变为输出。

9. 若采用8255的CA口作为行线，CB口作为列线，设计一个 4×4 的键盘电路，请画出8255与键盘矩阵的电路连接图并编写行反转法实现的按键识别程序。



8255地址为320H

1) 首先判别是否有键按下 (向行线A3~A0 输出全0):

MOV DX, 323H ; 口地址指向8255 控制寄存器

MOV AL, 81H ; 设定A、B口为方式0 输出, C 口C3~C0 为输入, C7~C4 为输出

OUT DX, AL

ABC: MOV AL, 00 ; 向CA口输出全 “0”行值

MOV DX, 322H ;

OUT DX, AL ;

IN AL, DX ; 从CB 口读入列值

AND AL, 0FH

CMP AL, 0FH ; 判断是否有键按下

JZ ABC ; 无键按下则继续查询

MOV AH, AL ; 保存CB值

CALL DELAY ; 有键按下则延迟一定时间 (一般为5~20ms) 用于消除键抖动;

2) 反向查找

MOV DX, 323H ; 口地址指向8255 控制寄存器

MOV AL, 88H ; 设定A、B口为方式0 输出, C 口C3~C0 为输出, C7~C4 为输入

OUT DX, AL;

MOV AL, AH ;

MOV DX, 322H

OUT DX, AL ; 输出列值

IN AL, DX ; 读入列值

AND AL, F0H ; 保留低4 位有效列值

ADD AL, AH ; AL 中为按键扫描码 (高4 位为行码, 低4 位为列码)

第七章、知识要点

- 几种不同的工作模式（方波发生器）
- 触发方式、GATE信号作用。
- 8253初始化。

1. 请说明 8253 分别用作定时器和计数器有哪些相同的地方和不同的地方。

1. 定时器和计数器的相同之处：都是采用倒计时方式对减一计数器预先设置的计数初始值在计数时钟 CLK 作用下不断减 1，并且在减一计数器的值减到零后引发输出电平变化，从而实现定时（分频）或计数功能。输出电平变化形式由工作方式决定，可编程选择。

2. 定时器和计数器的不同之处：

(1) 定时器方式下减一计数器的值减到零后会自动恢复其初始值进行新一轮的减一计数过程，并循环往复。而计数器方式下当减一计数器的值减到零后不再进行新的自动往复计数过程，除非有外界因素的重触发。

(2) 定时器方式下输入时钟必须是等周期的，而计数器方式下则不一定。

2. 请说明 8253 中的硬件触发和软件触发的含义是什么，起始触发和重触发的含义是什么。

- 在写入控制字后启动减一计数器开始计数工作的过程称为**起始触发**。
- 在减一计数器工作过程中使其按计数初值重新开始计数的过程称为**重触发**。
- 有软件触发和硬件触发两种方法。

软件触发：由程序中向计数初值寄存器写入初值来启动新计数过程的开始采用软件起始触发的工作方式有：方式 0；方式 2；方式 3；方式 4 能够软件重触发的工作方式有：方式 0；方式 4

硬件触发：由外设提供门控信号上升沿（由 0 变 1）来触发新计数过程的开始。

采用硬件起始触发的工作方式有：方式 1；方式 5 。

能够硬件重触发的工作方式有：方式 1；方式 2；方式 3；方式 4；方式 5 。

3. 请说明 8253 的 6 种工作模式中用于定时器和计数器各有几种工作方式，并归纳说明各种不同的定时器和计数器工作方式之间的主要区别。

8253 可编程设定为 6 种不同的工作模式，包括四种计数模式和两种定时模式。这些工作模式主要是通过以下方面加以区分：

- 输出端 OUT 状态不同

输出端的状态可分初始状态；计数过程中状态；计数结束后状态三个阶段。

注：后续提到的初始状态是指向 8253 写入控制命令字后的输出状态

- 计数过程中门控信号变化产生的作用不同

也即是否可硬件重触发。如果门控信号的变化（由 0 变 1）会导致重新开始减一计数过程，则称为硬件可重触发。

- 计数过程中向初值寄存器重新写入计数初值后对工作过程的影响不同

也即是否可软件重触发。如果在计数过程中向初值寄存器新写入计数初值后使得减一计数器按新的计数初值重新开始减一计数，则称为软件可重触发。

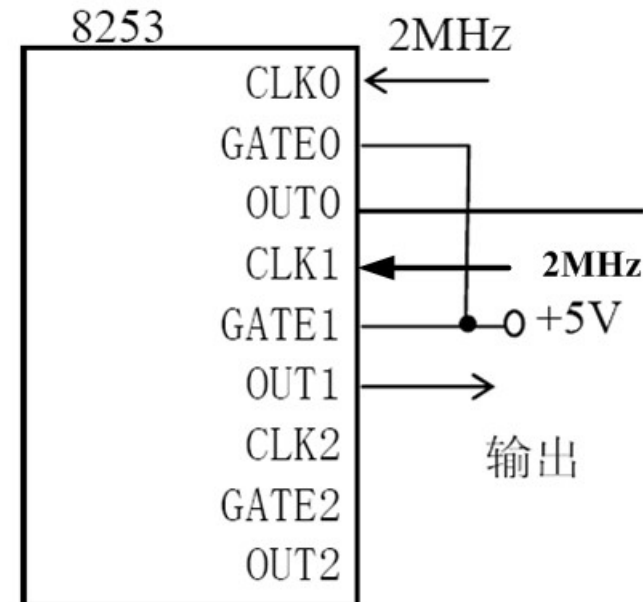
- 起始触发方式（启动减一计数器开始计数）不同

有软件起始触发和硬件起始触发两种方式。软件起始触发是指当计数初值写入初值寄存器后减一计数器就开始计数工作的方式。而硬件起始触发则是在计数初值写入初值寄存器后，还必须由 GATE 信号产生由低电平到高电平变化的上升沿才能启动减一计数器工作的方式。

4. 已知 CLK=2MHZ ，请编程使 8253 通道 0 输出产生一个 20ms 的低电平脉冲信号；使通道 1 输出产生一个周期为 20ms 的等脉宽时钟信号。

$$2000000 \times 0.02 = 40000$$

```
MOV DX, P8253+3;
MOV AL, 00110000B;
OUT DX, AL;
MOV DX, P8253;
MOV AX, 40000;
OUT DX, AL;
MOV AL, AH;
OUT DX, AL;
MOV DX, P8253+3;
MOV AL, 01110110B;
OUT DX, AL;
MOV DX, P8253+1;
MOV AX, 40000;
OUT DX, AL;
MOV AL, AH;
OUT DX, AL;
```



5. 请说明 PC 机中的 8253 定时/计数器 3 个通道各自都有哪些主要作用。

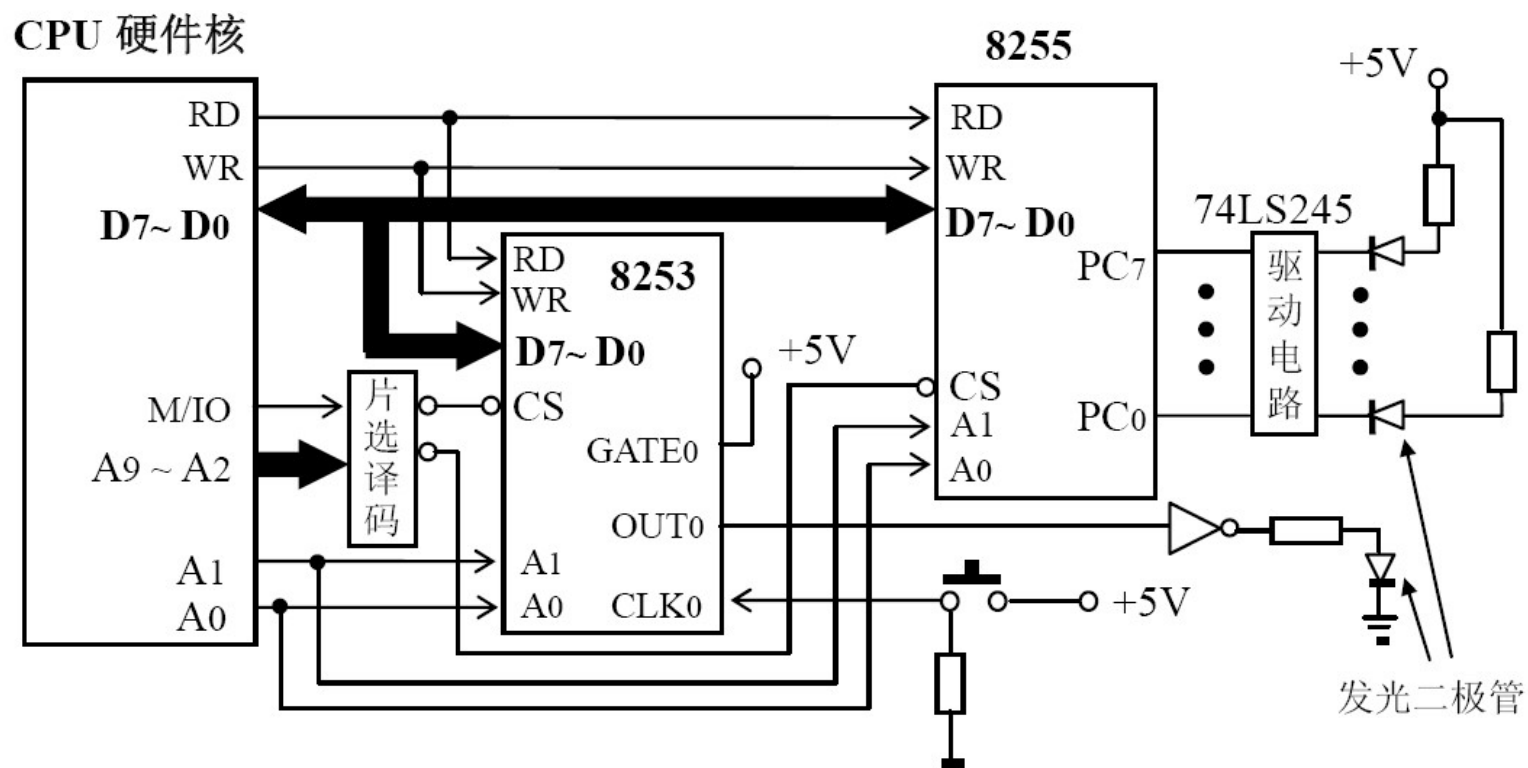
在 PC 系列微机中有一片 8253 定时/计数器，它的三个通道作用分别是：

通道 0：应用程序可编程使用的定时器，用于产生周期可编程设定的定时中断。系统在初始状态下设置为最大计数初值 0（共计数 65536 次，产生 54.9mS 的定时中断）。用户可通过截取 08H 类型号中断使用。

通道 1：只能由系统使用，用于产生动态存储器的定时刷新信号。

通道 2：用于编程产生使机器内的扬声器发出不同频率声音的定时信号串。

6. 欲自行设计一个自行车里程表并采用 8253 作为计数器，自行车每转一圈产生一个计数脉冲。请说明 8253 可采用什么工作方式，并画出实现方案图。如果在某时刻要将已骑圈数用 OUT 指令输出到口地址为 PORT 的显示设备，请写出相关的程序片断。



```
MOV DX,20BH ; 设置8255 控制字
MOV AL,10000000B ; A、B、C 口都为方式0 输出（A、B 口无用）
OUT DX,AL
MOV DX,203H ; 设置8253 控制字
MOV AL,00110000B ; 0 通道为方式0，按字节读写，二进制
OUT DX,AL
MOV DX,200H ; 向8253 通道0 写计数初值
MOV AL, FFH; 假设设置的初值为FFH
OUT DX,AL;
NOP
```

```
DISPLAY: MOV DX,203H ; 8253 控制口地址
MOV AL,00000000B ; 向8253 通道0 写“锁存减1 计数器当前值”命令
OUT DX,AL
NOP
MOV DX,200H ; 8253 通道0 口地址
IN AL,DX ; 读8253 通道0 当前值锁存器内容
MOV BL,AL
MOV AL,FFH ; 取计数初值FFH
SUB AL,BL ; 用计数初值减当前计数值
MOV DX,20AH ; 8255 的C 口地址
NOT AL ; 将计算出的按键脉冲计数值取反（因“0”才发亮）
OUT DX,AL ; 向8255 C 口输出
JMP DISPLAY ; 继续下一次循环
```